

# ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

## ΘΕΜΑ

Αναμόρφωση των εργαστηριακών ασκήσεων για  
τα μαθήματα Βάσεις Δεδομένων & Ειδικά Θέματα  
Βάσεων Δεδομένων



**Σπουδάστριες:**

Σαριανίδου Χαρίκλεια 2758

Καρατζοπούλου Μιχαέλα 2793

**Επιβλέπων Καθηγητής:**

Δρ. Τσιμπίρης Αλκιβιάδης

Σέρρες, Σεπτέμβριος 2014



## Περιγραφή πτυχιακής εργασίας

Σκοπός της πτυχιακής εργασίας είναι η αναμόρφωση των εργαστηριακών ασκήσεων των μαθημάτων Βάσεις Δεδομένων και Ειδικά θέματα Βάσεων Δεδομένων. Θα ξαναγραφούν οι υφιστάμενες ασκήσεις και θα αναπτυχθούν και νέες. Στις εργαστηριακές αυτές ασκήσεις θα δίνεται έμφαση στη γλώσσα SQL αλλά και σε γνωστά RDBMS (Access και SQL Server ) στα οποία θα εργασθούν οι σπουδαστές. Τα RDBMS αυτά μπορούν να επικοινωνούν με γλώσσες προγραμματισμού (MS Visual Studio, C#, PHP). Στις ασκήσεις αυτές θα δοθεί προσοχή στη λεπτομερή και στη βήμα – βήμα περιγραφή του κάθε προβλήματος καθώς και της λύσης του. Θα αναπτυχθεί ένας ιστότοπος όπου θα αναρτηθούν οι ασκήσεις και οι λύσεις τους, με επιπρόσθετες ερωτήσεις πολλαπλής επιλογής.

## Εισαγωγή

Για να επιτευχθεί μια ενιαία δομή στην πτυχιακή αυτή εξαιτίας του μεγάλου όγκου ασκήσεων και ύλης που έπρεπε να οργανωθεί, ακολουθήσαμε την διαμέριση της σε επτά ενότητες.

Συγκεκριμένα στην πρώτη ενότητα περιγράφουμε τον τρόπο εγκατάστασης των απαραίτητων προγραμμάτων για τα εργαστήρια των μαθημάτων, καθώς και τα εγχειρίδια χρήσης της ACCESS 2010, του SQL Server 2008 και του Visual Studio 2010 στα θέματα που αφορούν τις εργαστηριακές ασκήσεις.

Στην Ενότητα 2 παρουσιάζουμε τις διαφάνειες του θεωρητικού μέρους του μαθήματος «Βάσεις Δεδομένων» οι οποίες έχουν διαμορφωθεί με συγκεκριμένο template ώστε να είναι όλες ομοιόμορφες.

Στην Ενότητα 3 παραθέτουμε τις 13 εργαστηριακές ασκήσεις των μαθήματος «Βάσεις Δεδομένων» οι οποίες έχουν αναπτυχθεί με ACCESS και SQL Server.

Στην Ενότητα 4 παρουσιάζουμε τις διαφάνειες του θεωρητικού μέρους του μαθήματος «Ειδικά θέματα Βάσεων Δεδομένων» οι οποίες έχουν διαμορφωθεί με συγκεκριμένο template ώστε να είναι όλες ομοιόμορφες και με πλούσιο οπτικό υλικό ( διαγράμματα και παραδείγματα).

Στην Ενότητα 5 παραθέτουμε τις 9 εργαστηριακές ασκήσεις των μαθήματος «Ειδικά θέματα Βάσεων Δεδομένων» οι οποίες έχουν αναπτυχθεί με SQL Server και Visual Studio.

Στην Ενότητα 6 καταγράφονται τα ερωτήματα πολλαπλής επιλογής καθώς και οι απαντήσεις τους σύμφωνα με παλαιότερα θέματα εξεταστικών περιόδων. Επίσης γίνεται μια περιγραφή του προγράμματος υλοποίησης τους (Hot Potatoes).

Τέλος στην Ενότητα 7 γίνεται μια μικρή περιγραφή της δημιουργίας της ιστοσελίδας μέσω του συστήματος διαχείρισης περιεχομένου (Wordpress).

Η Πτυχιακή Εργασία είναι αυτοδύναμη επιστημονικό-πρακτική εργασία δημιουργικού χαρακτήρα, η εκπόνηση της οποίας αποτελεί αποτέλεσμα της αναλυτικής και συνθετικής ικανότητας του φοιτητή, της δυνατότητάς του να χρησιμοποιεί όλο το φάσμα των γνώσεων και δεξιοτήτων που έλαβε κατά τη διάρκεια των σπουδών του για την ανάπτυξη ενός επίκαιρου θέματος.

Η αντιστοιχία του περιεχομένου και της ποιότητας της πτυχιακής εργασίας στις απαιτήσεις των προδιαγραφών του Τμήματος και η επιτυχής παρουσίασή της πιστοποιούν την υψηλή προετοιμασία του πτυχιούχου στην εφαρμογή της επιστήμης. Αποτελεί μια αναλυτική και σύνθετη εργασία που σχετίζεται με όλο το φάσμα της Επιστήμης της Πληροφορικής.



## Περιεχόμενα

Περιγραφή πτυχιακής εργασίας .....	3
Εισαγωγή .....	4
Ενότητα1-Εγχειρίδια εγκατάστασης και χρήσης.....	8
Manual of Microsoft SQL Server 2008 .....	8
Microsoft Visual Studio .....	13
Οδηγός εγκατάστασης εργαλείων για το εργαστήριο «Ειδικά θέματα Βάσεων Δεδομένων» .....	15
Εγκατάσταση Visual Studio 2010 Professional.....	23
Εγκατάσταση Microsoft SQL Server 2008 .....	26
Εγχειρίδιο Microsoft Access 2010.....	33
Βάση δεδομένων.....	33
Περιβάλλον της Access 2010.....	33
Κοινές χρήσεις της Microsoft Access 2010.....	39
Δημιουργία Βάσης Δεδομένων .....	39
Πίνακας.....	40
Ερωτήματα .....	43
Φόρμα .....	49
Έκθεση .....	50
Ενότητα2 – Διαφάνειες του θεωρητικού μέρους του μαθήματος «Βάσεις Δεδομένων» .....	51
Παρουσίαση 1 – Συστήματα Βάσεων Δεδομένων .....	51
Παρουσίαση 2 – Σχεδιασμός βάσης δεδομένων, E – R .....	61
Παρουσίαση 3 – Σχεσιακό Μοντέλο .....	76
Παρουσίαση 4 – Μετατροπή σχήματος Ο/Σ σε σχεσιακό .....	87
Παρουσίαση 5 – Σχεσιακή άλγεβρα.....	95
Παρουσίαση 6 – ΒΔ διαχείρισης προσωπικού.....	100
Παρουσίαση 7 – Εισαγωγή στη γλώσσα SQL.....	106
Παρουσίαση 8 – Γλώσσα χειρισμού δεδομένων .....	113
Παρουσίαση 9 – SQL – Σύζευξη πινάκων .....	124
Παρουσίαση 10 – Συναθροιστικές συναρτήσεις .....	130
Παρουσίαση 11 – Ερωτήματα Ομαδοποίησης με Συνένωση Πινάκων .....	137
Παρουσίαση 12 – Κανονικοποίηση.....	143
Ενότητα3 – Εργαστηριακές ασκήσεις του μαθήματος «Βάσεις Δεδομένων» .....	150
Άσκηση 1- Φοιτητές, ΒΔ σε Access .....	150

Άσκηση 2 – Φοιτητές – Μαθήματα - Καθηγητές σε Access.....	155
Άσκηση 3 – Δημιουργία φορμών – Query by example σε Access .....	165
Άσκηση 4 – Εργαλεία σχεδιασμού φορμών και εκθέσεων σε Access .....	177
Άσκηση 5 – Γλώσσα χειρισμού δεδομένων SQL σε Access.....	184
Άσκηση 6 –Δημιουργία ερωτημάτων SQL σε Access .....	188
Άσκηση 7 – Σύζευξη πινάκων .....	191
Άσκηση 8 – Συναθροιστικές συναρτήσεις σε Access .....	197
Άσκηση 9 – Εργαζόμενοι – Τμήματα - Έργα, ΒΔ σε SQL Server .....	202
Άσκηση 10 – Δημιουργία συσχετίσεων πινάκων σε SQL Server.....	210
Άσκηση 11 - INSERT INTO, UPDATE, DELETE σε SQL Server.....	214
Άσκηση 12 – Δημιουργία ερωτημάτων QBE στον SQL SERVER .....	216
Άσκηση 13 – Σύνθετα ερωτήματα και συναθροιστικές συναρτήσεις σε SQL Server .....	223
Ενότητα 4 – Διαφάνειες του θεωρητικού μέρους του μαθήματος «Ειδικά θέματα Βάσεων Δεδομένων» .....	230
Παρουσίαση 1 – Εισαγωγή και αρχιτεκτονική των βάσεων δεδομένων.....	230
Παρουσίαση 2 – Ενοιολογικά – Λογικά μοντέλα .....	244
Παρουσίαση 3 – Εισαγωγή και αρχιτεκτονική των βάσεων δεδομένων.....	262
Παρουσίαση 4 – Οι όψεις στην SQL.....	279
Παρουσίαση 5 – Επεκτάσεις SQL .....	291
Παρουσίαση 6 – Ανάκαμψη και συναλλαγές .....	313
Παρουσίαση 7 – Ταυτοχρονισμός.....	331
Παρουσίαση 8 – Ασφάλεια ΒΔ.....	344
Παρουσίαση 9 – Κρυπτογράφηση δεδομένων .....	361
Παρουσίαση 10 – Μεθοδολογία συσχέτισης ΒΔ με εφαρμογές οπτικού προγραμματισμού .....	375
Παρουσίαση 11 – Αντικειμενοστραφής και αντικειμενο-σχεσιακές βάσεις δεδομένων. ....	381
Παρουσίαση 12 – Ευρετήρια .....	388
Παρουσίαση 13 – Σύγχρονα θέματα βάσεων δεδομένων.....	408
Παρουσίαση 14 – XML .....	417
Παρουσίαση 15 – Εξόρυξη Γνώσης (Data Mining) .....	435
Ενότητα 5 – Εργαστηριακές ασκήσεις του μαθήματος «Ειδικά θέματα Βάσεων Δεδομένων» .....	443
Άσκηση 1- Εισαγωγή στον SQL Server 2008.....	443
Άσκηση 2 – Δημιουργία σύνδεσης σε SQL Server.....	461
Άσκηση 3 – Ενοικίαση Αυτοκινήτων .....	471

Άσκηση 4 – Δημιουργία Εφαρμογής.....	490
Άσκηση 5 – Εκτέλεση ερωτημάτων σε Visual Studio 2010 .....	499
Άσκηση 6 – Εκτέλεση διαδοχικών ερωτημάτων .....	502
Άσκηση 7 – Προβολή εικόνας σε φόρμα .....	509
Άσκηση 8 – Δημιουργία αναφορών .....	515
Άσκηση 9 – Δημιουργία αποθηκευμένων διαδικασιών και σκανδάλης .....	519
Ενότητα 6 –Ερωτήματα πολλαπλής επιλογής.....	526
Χρήση του προγράμματος Hot Potatoes .....	526
Ερωτήσεις μαθήματος «Βάσεις Δεδομένων» .....	528
Ερωτήσεις του μαθήματος «Ειδικά θέματα Βάσεων Δεδομένων» .....	542
Ενότητα 7 – Ιστοσελίδα μαθημάτων Βάσεις Δεδομένων & Ειδικά θέματα Βάσεων Δεδομένων .....	556
Εισαγωγή .....	556
Τι είναι το WordPress.....	556
Τι είναι Xampp.....	556
Περιγραφή Site.....	556
Βιβλιογραφία .....	559
Βιβλία .....	559
Άρθρα .....	561

## Ενότητα1-Εγχειρίδια εγκατάστασης και χρήσης

### Manual of Microsoft SQL Server 2008

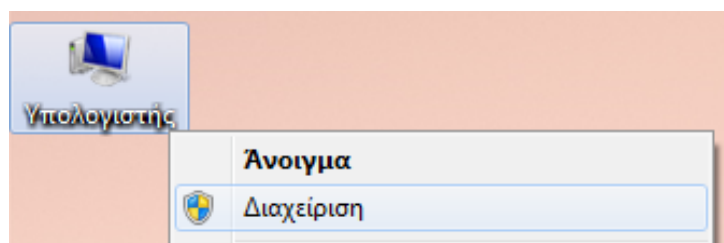


#### Τι είναι ο Microsoft SQL Server

Ο SQL Server είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων, το οποίο αναπτύσσεται από τη Microsoft. Ο SQL Server βγήκε για πρώτη φορά στην αγορά το 1989 σε συνεργασία με την Sybase. SQL είναι τα αρχικά της αγγλικής ονομασίας (Structured Query Language) η οποία στην ελληνική ορολογία μεταφράζεται ως δομημένη γλώσσα ερωτημάτων.

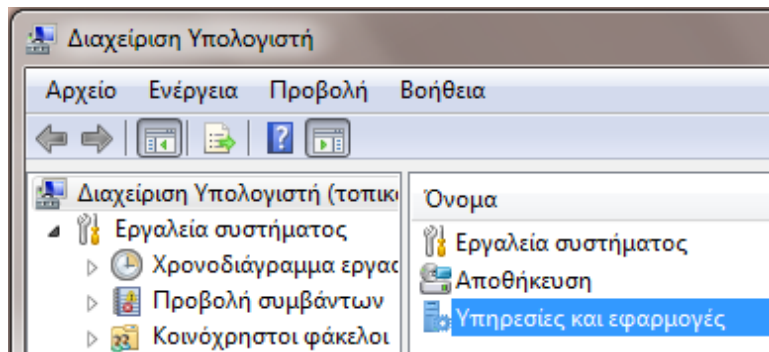
Με την SQL μπορούμε να έχουμε πρόσβαση σε πληθώρα βάσεων δεδομένων, να εκτελούμε αναζητήσεις με κριτήρια, να εισάγουμε νέα δεδομένα, να διαγράφουμε υπάρχοντα δεδομένα να κάνουμε αλλαγές σε υπάρχοντα δεδομένα. Αξίζει να σημειωθεί ότι η SQL είναι τυποποιημένη από τον Αμερικανικό Ινστιτούτο Τυποποιήσεων γνωστή και ως ANSI (American National Standards Institute) για την πρόσβαση και διαχείριση βάσεων δεδομένων. Εντολές SQL χρησιμοποιούνται για την εξαγωγή και ανανέωση δεδομένων από μια βάση δεδομένων. Οι βασικές εντολές συναντώνται σε όλες τις βάσεις δεδομένων όπως SELECT, UPDATE, DELETE, INSERT, WHERE, και άλλες).

Αφού έχουμε εγκαταστήσει τον MS SQL Server ξεκινάμε ελέγχοντας αν όλες οι υπηρεσίες του τρέχουν σωστά στον υπολογιστή μας πατώντας δεξί κλικ στον «Υπολογιστή» → Διαχείριση .



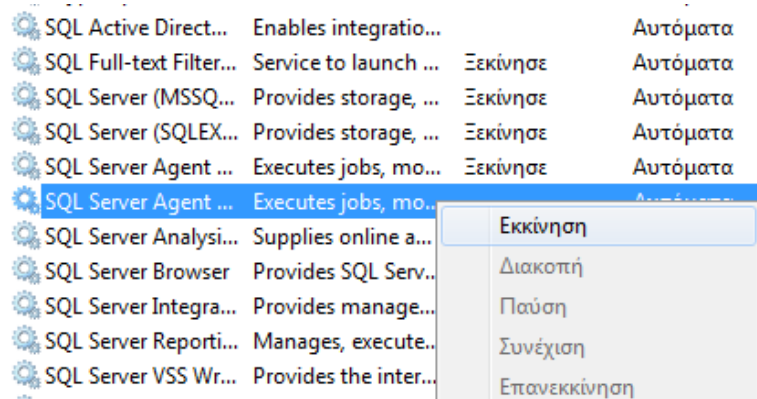
Εικόνα 1

Στο παράθυρο που ανοίγει πατάμε διπλό κλικ στις **Υπηρεσίες και εφαρμογές** → **Υπηρεσίες**.



Εικόνα 2

Αναζητούμε όλες τις εφαρμογές που υπάρχουν για την SQL, ελέγχουμε αν τρέχουν, αν όχι κάνουμε **δεξί κλικ → Εκκίνηση**.



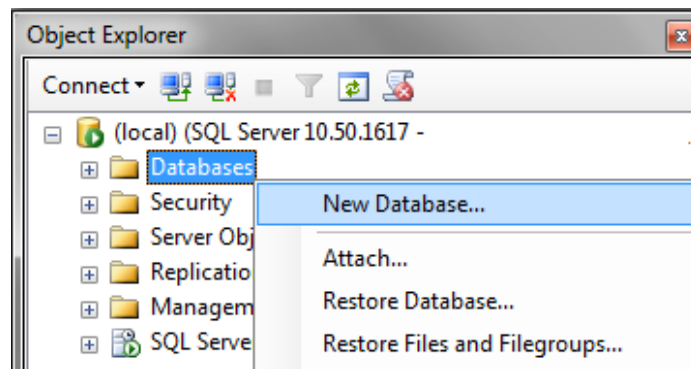
Εικόνα 3

Μόλις ολοκληρώσουμε τις παραπάνω διαδικασίες ανοίγουμε τον MS SQL Server που μας ζητάει τύπο και όνομα του Server για να ξεκινήσει η επικοινωνία με τον Server. Συμπληρώνουμε τα στοιχεία και πατάμε Connect.



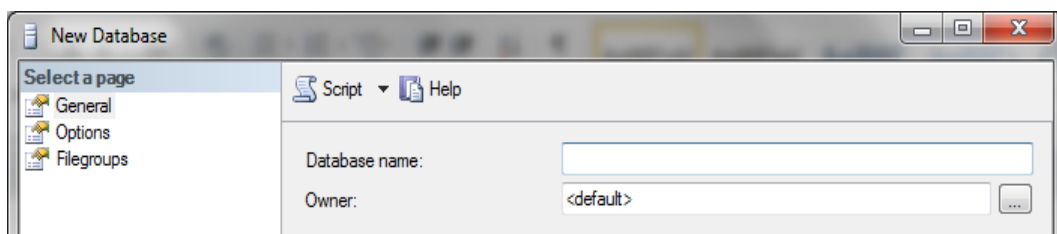
Εικόνα 4

Για να δημιουργήσουμε μια βάση δεδομένων πηγαίνουμε στο παράθυρο **Object Explorer** (που ανοίγει αυτόματα αν όχι, πάμε **View** → **Object Explorer** ή πατάμε F8 από το πληκτρολόγιο μας). Έπειτα κάνουμε **δεξί κλικ** → **Databases** → **New Database...**



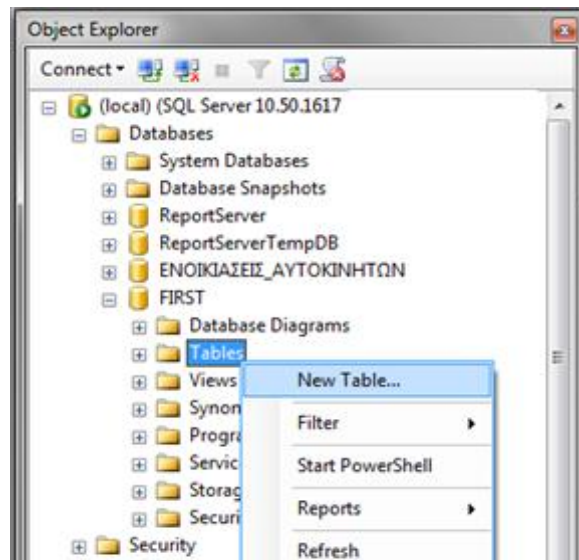
Εικόνα 5

Ονομάζουμε την νέα βάση δεδομένων και πατάμε **OK**.



Εικόνα 6

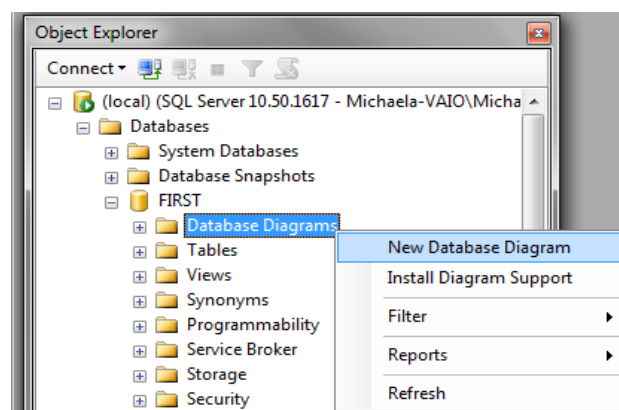
Αν επεκτείνουμε την διακλάδωση του Databases θα δούμε την νέα μας βάση, επεκτείνουμε και την βάση μας κάνουμε **δεξί κλικ** → **New Table...** για να δημιουργήσουμε πίνακες. Ακολουθούμε την ίδια διαδικασία για όσους πίνακες θέλουμε να δημιουργήσουμε.



Εικόνα 7

### Δημιουργία διαγραμμάτων βάσης δεδομένων

Μπορείτε να χρησιμοποιήσετε **Object Explorer** για να δημιουργήσετε νέα διαγράμματα βάσης δεδομένων (**Database Diagrams**). Διαγράμματα βάσης δεδομένων δείχνουν παραστατικά την δομή της βάσης δεδομένων. Χρησιμοποιώντας τα διαγράμματα βάσης δεδομένων, μπορείτε να δημιουργήσετε και να τροποποιήσετε πίνακες, στήλες, τις σχέσεις, και τα κλειδιά. Επιπλέον, μπορείτε να τροποποιήσετε τα ευρετήρια και περιορισμούς.

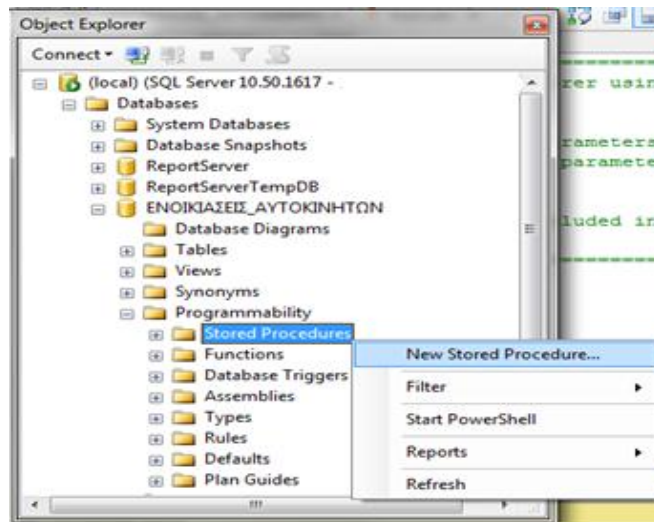


Εικόνα 8

### Αποθηκευμένες διαδικασίες (Stored Procedures)

Μια αποθηκευμένη διαδικασία είναι ένα πρόγραμμα που αποτελείται από SQL εντολές, που αποθηκεύεται και εκτελείται στον database server χωρίς ενδιάμεση επικοινωνία με το πρόγραμμα εφαρμογής.

Για να δημιουργήσετε μια Αποθηκευμένη Διαδικασία ακολουθείτε τα παρακάτω βήματα: **Programmability** → **Stored Procedures** → **Δεξί κλικ** → **New Stored Procedure**.

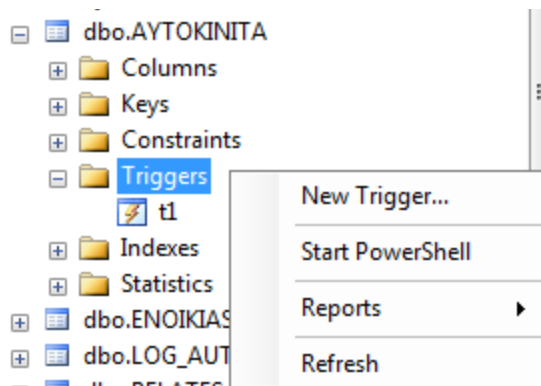


Εικόνα 9

### Σκανδάλες ( Triggers )

Trigger (Σκανδάλη), ονομάζεται ένα αντικείμενο Βάσης Δεδομένων, το οποίο απευθύνεται σε συγκεκριμένο πίνακα και εκτελείται όταν συμβεί ένα συγκεκριμένο γεγονός σε αυτόν τον πίνακα.

Για να δημιουργήσουμε ένα **Trigger** ή συντάσσουμε ένα νέο **Query** ή επιλέγουμε **Την βάση μας → Triggers Δεξί κλικ → New Trigger .**

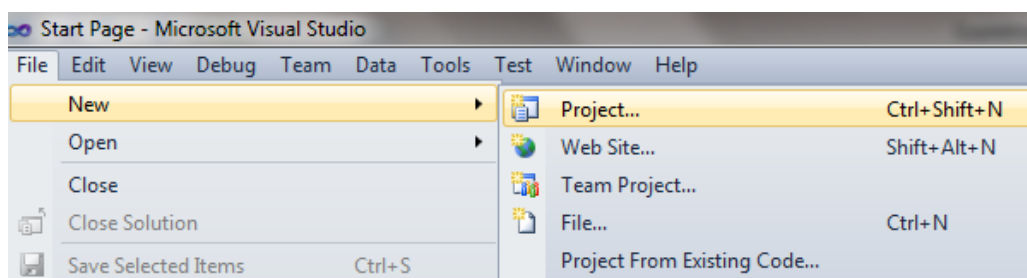


Εικόνα 10



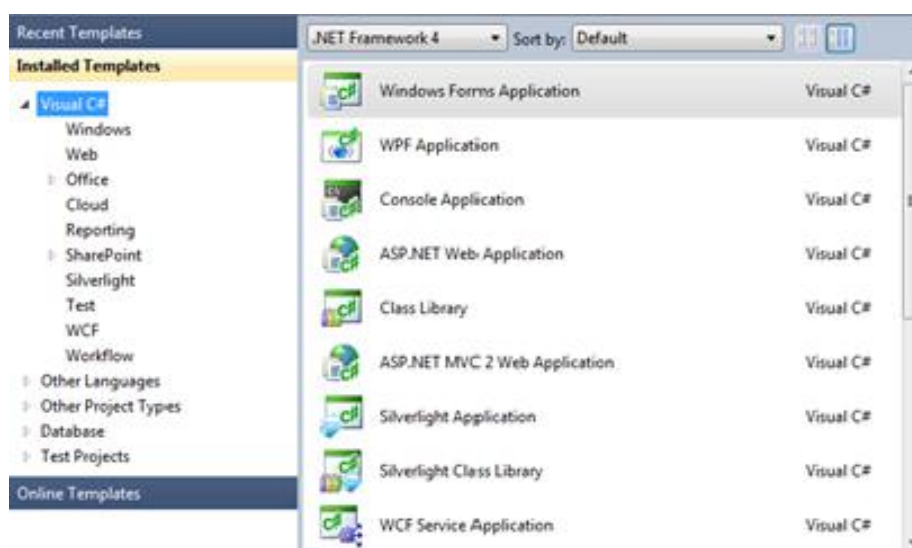
## Microsoft Visual Studio

Τέλος χρησιμοποιούμε το **Microsoft Visual Studio** για να δημιουργήσουμε ένα νέο Project (Solution) στο οποίο θα εργαστούμε. Πάμε λοιπόν File → New → Project.



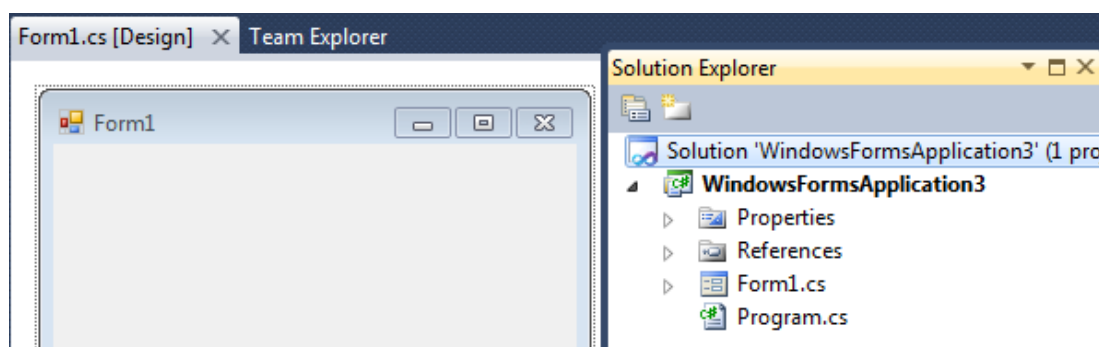
Εικόνα 11

Η Γλώσσα Προγραμματισμού με την οποία θα ασχοληθούμε είναι η Visual C#. Έτσι επιλέγουμε **Visual C# → Windows Forms Application → OK**.



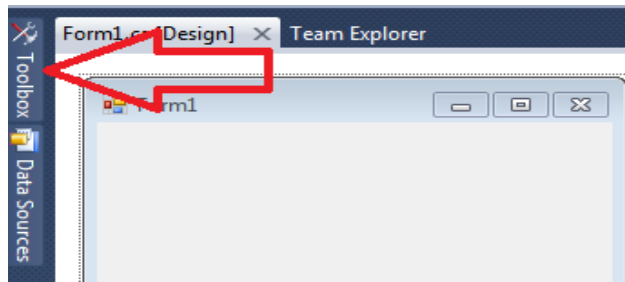
Εικόνα 12

Με τα παραπάνω βήματα έχει δημιουργηθεί μια φόρμα (Form1).



Εικόνα 13

Τέλος προσθέσουμε τα components που χρειαζόμαστε και διαμορφώνουμε την φόρμα. Όλα τα Components βρίσκονται στο εσωτερικό του **Toolbox**.



Εικόνα 14

Περισσότερα και λεπτομερέστερα παραδείγματα θα παρουσιαστούν στις Ασκήσεις του εργαστηρίου.

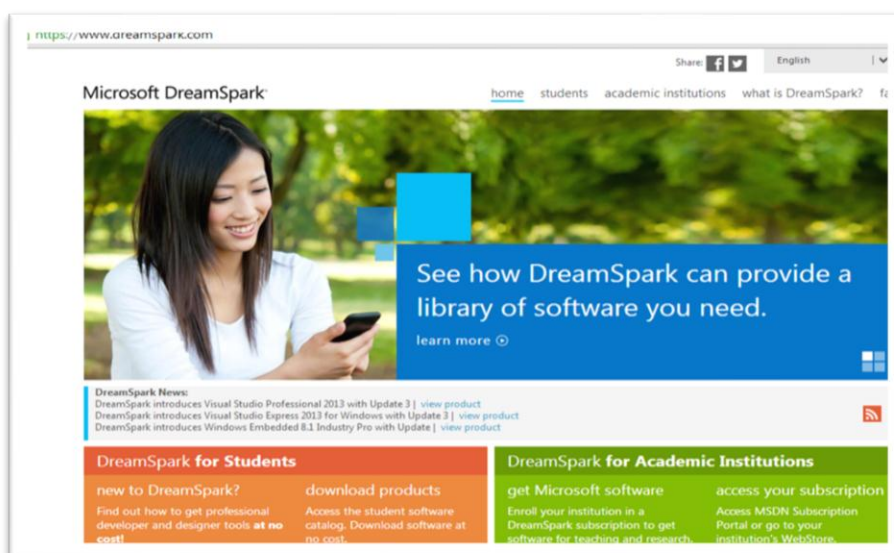
## Οδηγός εγκατάστασης εργαλείων για το εργαστήριο «Ειδικά θέματα Βάσεων Δεδομένων»

Τα εργαλεία που θα χρησιμοποιηθούν στο εργαστήριο είναι :

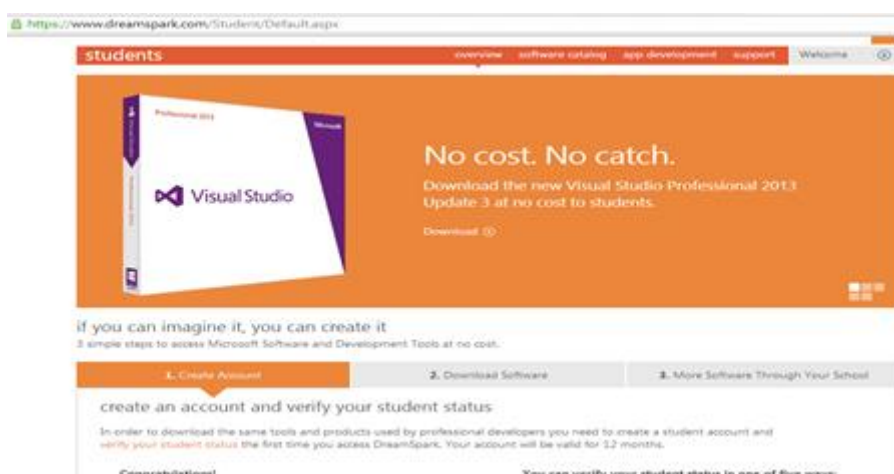
1. Microsoft SQL Server 2008
2. Microsoft Visual Studio 2010

Τα προγράμματα αυτά μπορείτε να τα βρείτε και να τα κατεβάσετε δωρεάν από την σελίδα <https://www.dreamspark.com/> η διαδικασία είναι η εξής:

- Ανοίγεται τον Mozilla Firefox μπαίνετε στην σελίδα και πατάτε στο **student** που βρίσκεται πάνω δεξιά.

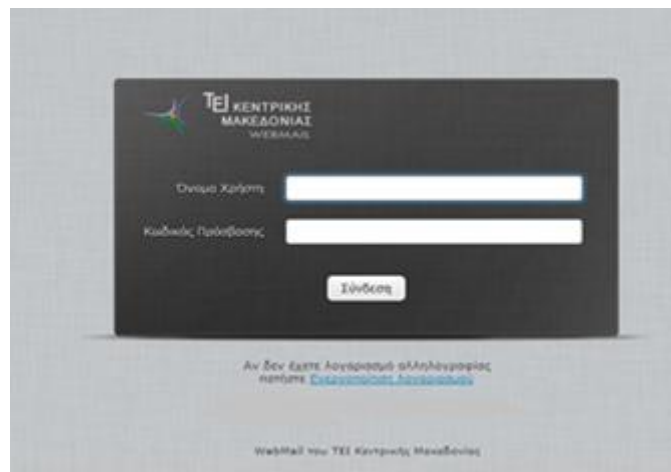


Εικόνα 1



Εικόνα 2

- Αρχικά πρέπει να δημιουργήσετε λογαριασμό στο **DreamSpark**. Για να γίνει αυτό θα πρέπει να διατηρείτε ή να δημιουργήσετε e-mail με επέκταση **@outlook.com**.
- Μόλις ολοκληρώσετε το παραπάνω βήμα εισάγουμε απλά το username και το password του outlook.
- Στην σελίδα που μας εμφανίζει πατάμε **Get Verified**.
- Παρατηρήστε ότι σας ζητάει να δημιουργήσετε έναν λογαριασμό email της μορφής \_\_\_\_@teicm.gr στο <http://mail.teicm.gr/> και επιλέγετε την δημιουργία ενός email καθώς και την ενεργοποίηση του με τα στοιχεία σας από το E-gram.



Εικόνα 3

### Ενεργοποίηση EMAIL TEI Σερρών

**ΦΟΡΜΑ ΕΝΕΡΓΟΠΟΙΗΣΗΣ EMAIL**

Εισάγετε το όνομα χρήστη και τον κωδικό σας: \*

---

Όνομα χρήστη:

Κωδικός:

---

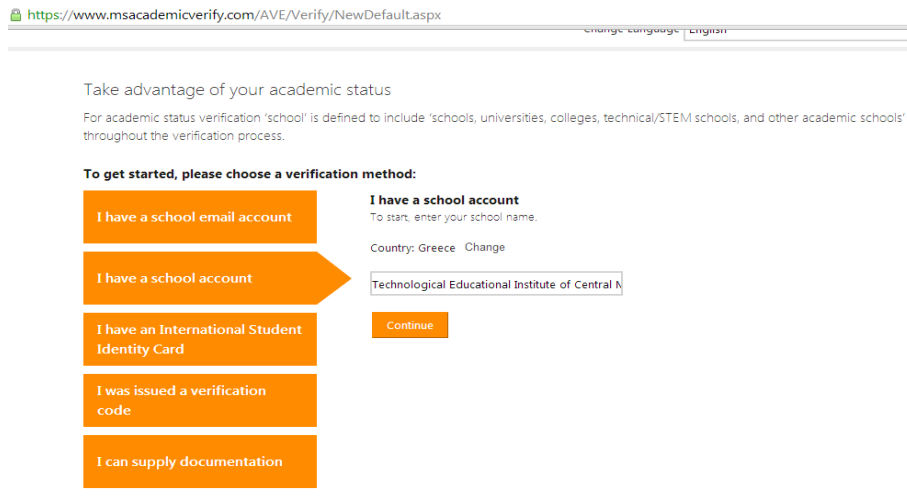
\* στοιχεία λογαριασμού Ηλεκτρονικής Γραμματείας

**ΟΔΗΓΙΕΣ:**

- Μέσω της σελίδας αυτής μπορούν να ενεργοποιηθούν αυτόματα το EMAIL τους όσοι φοιτητές ή εκπαιδευτικοί δεν έχουν ήδη λογαριασμό στο σύστημα Ηλεκτρονικής Αλληλογραφίας του Ιδρύματος.
- Απαραίτητη προϋπόθεση για την επιτυχή ενεργοποίηση είναι να υπάρχουν καταχωρημένα τα στοιχεία του ενδιαφερόμενου χρήστη στην Υπηρεσία Καταλόγου LDAP του TEI Σερρών. Η βάση δεδομένων του εξυπηρετητή LDAP συγχρονίζεται αυτόματα καθημερινά με τα στοιχεία ταυτότητας των χρηστών που είναι εγγεγραμμένοι στο σύστημα της Ηλεκτρονικής Γραμματείας.
- Έαν είστε φοιτητής και δεν έχετε λογαριασμό στην Ηλεκτρονική Γραμματεία παρακαλούμε πρώτα **ενεργοποιήστε τον**.
- Αν για οποιοδήποτε άλλο λόγο (είτε έχετε λογαριασμό στην Ηλεκτρονική Γραμματεία, είτε όχι) το σύστημα δε σας βρίσκει καταχωρημένο στον Κατάλογο LDAP, παρακαλούμε επικοινωνήστε με το Κέντρο Διαχείρισης και Λειτουργίας Δικτύου του TEI Σερρών για την υποβολή αίτησης ενεργοποίησης του λογαριασμού αλληλογραφίας σας (Τηλ. 2321049160, Email: [noc@teiser.gr](mailto:noc@teiser.gr)).

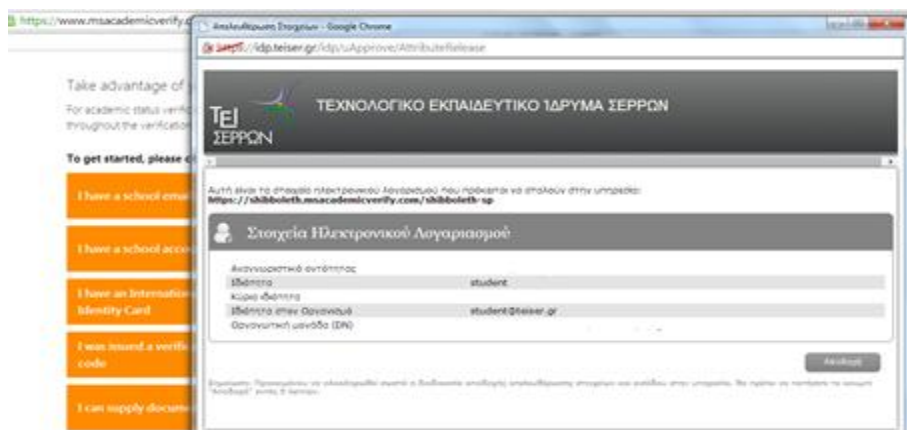
Εικόνα 4

- Τέλος με το καινούργιο σας email και τον κωδικό σας από το egram εισέρχεστε στο dreamspark το οποίο επιβεβαιώνει τα στοιχεία σας καθώς και το εκπαιδευτικό ίδρυμα στο οποίο εντάσσεστε αφού πληκτρολογήσετε αυτά τα στοιχεία στην επιλογή **I have a school account** και έπειτα συνέχεια.



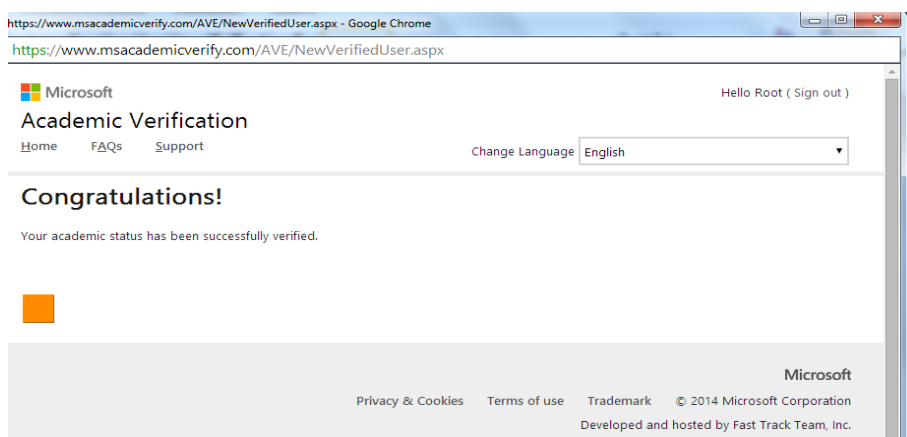
Εικόνα 5

- Σε αυτό το σημείο σας βγάζει σφάλμα επικίνδυνου ιστότοπου πατάτε «καταλαβαίνω τους κινδύνους» και εισάγουμε τα στοιχεία του e-gram.



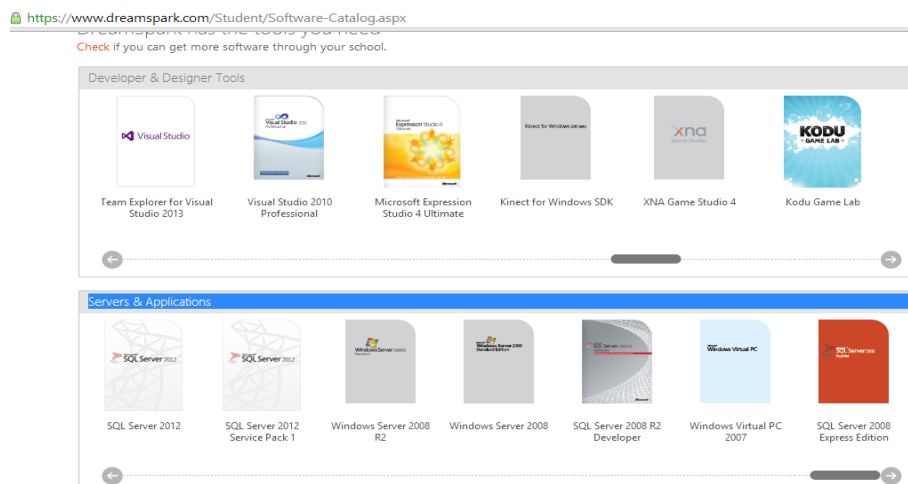
Εικόνα 6

- Ελέγχεται τα στοιχεία σας και έπειτα πατάτε Αποδοχή.



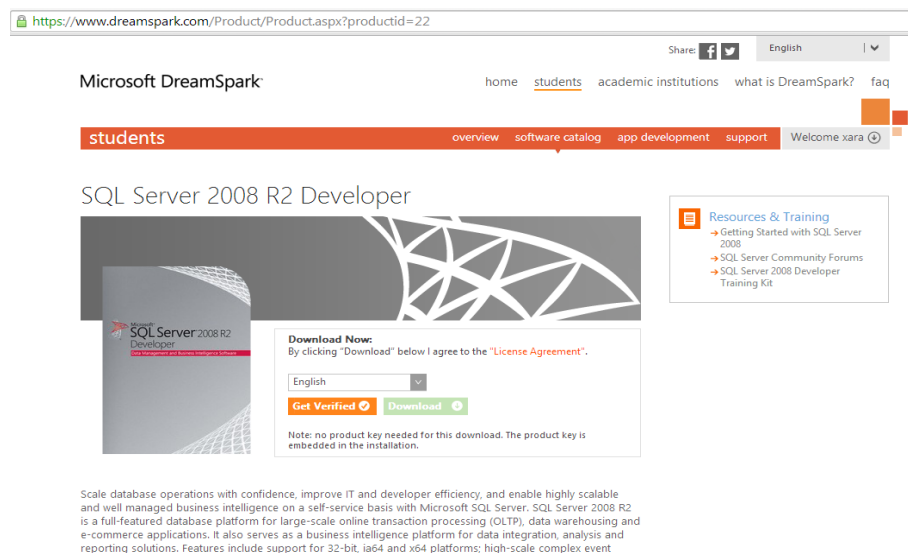
Εικόνα 7

- Εδώ τελειώνει η διαδικασία εγγραφής στο dreamspark. Είστε έτοιμοι να κατεβάσετε τα προγράμματα, επιστρέφοντας στην αρχική σελίδα στο υπομενού student του dreamspark και πατώντας πάνω δεξιά στην επιλογή software catalog.
- Αφού ανοίξει ο κατάλογος με τα διαθέσιμα προγράμματα είστε έτοιμοι να επιλέξετε τα προγράμματα που σας ενδιαφέρουν. Για το συγκεκριμένο μάθημα θα χρειαστείτε την **Microsoft SQL server 2008**.



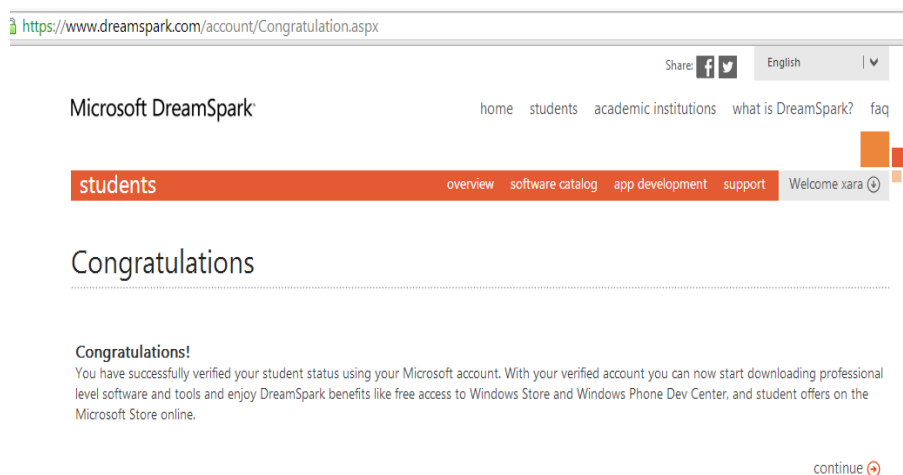
Εικόνα 8

- Μετά την επιλογή του προγράμματος ανοίγει αυτή η σελίδα πατάτε κάτω αριστερά στο get verified.



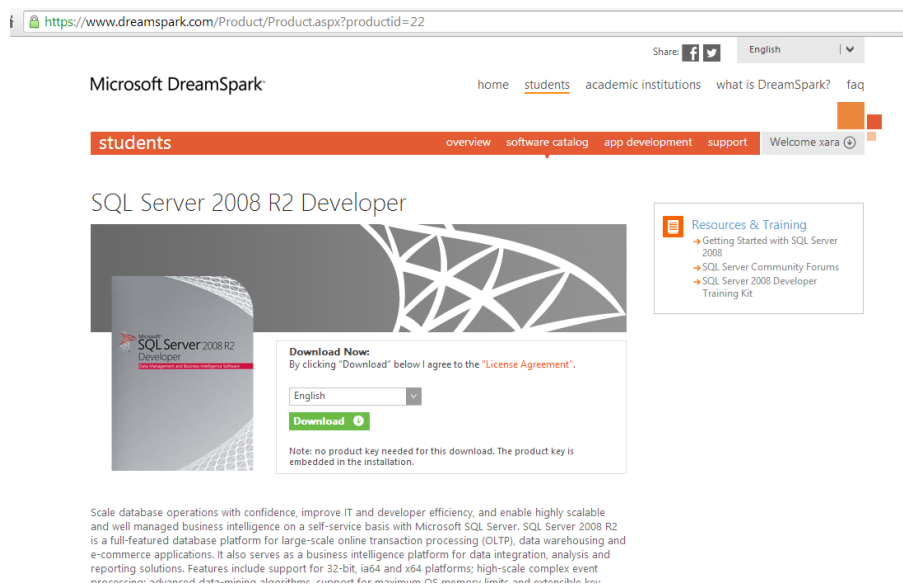
Εικόνα 9

- Σας ενημερώνει για την σωστή εισαγωγή σας στην διαδικασία και πατάτε συνέχεια.



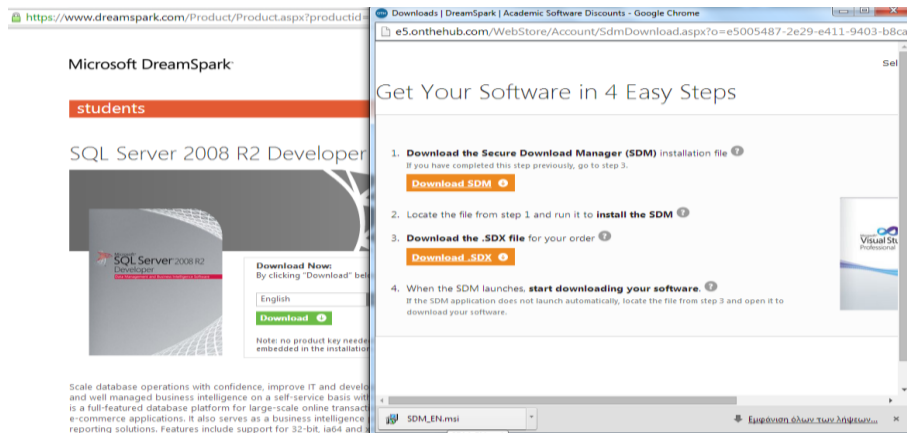
Εικόνα 10

- Το επόμενο βήμα είναι να πατήσετε στο download ώστε να ξεκινήσει η διαδικασία.



Εικόνα 11

- Αφού πατήσετε στο Download ανοίγει ένα παράθυρο το οποίο σας βοηθάει να κατέβει το πρόγραμμα που επιθυμείτε γρηγορότερα και ασφαλέστερα. Το μόνο που έχετε να κάνετε είναι να κατεβάσετε και να εγκαταστήσετε το **SDM (Secure Download Manager)** και έπειτα με την επιλογή Download SDX ανοίγει ένα νέο παράθυρο το οποίο είναι και το τελευταίο για το κατέβασμα του προγράμματος.



Εικόνα 12

- Πατήστε Start Download και αναμένετε μέχρις ότου να ολοκληρωθεί.



Εικόνα 13

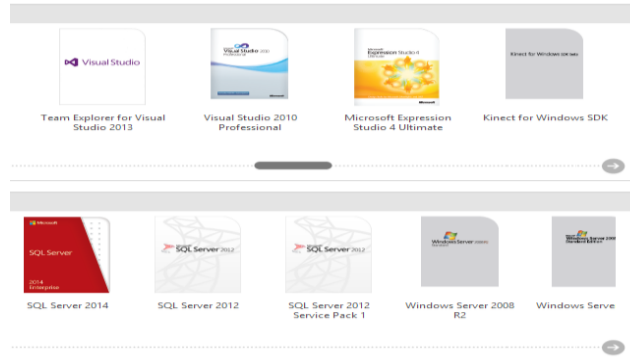
- Η διαδικασία λήψης έχει ολοκληρωθεί, με την επιλογή Launch σας πηγαίνει στην τοποθεσία όπου έχει αποθηκευτεί το πρόγραμμά σας. Η τοποθεσία στην οποία θα αποθηκευτεί είναι της επιλογής σας και μπορείτε να την τροποποιήσετε πατώντας στο change location.



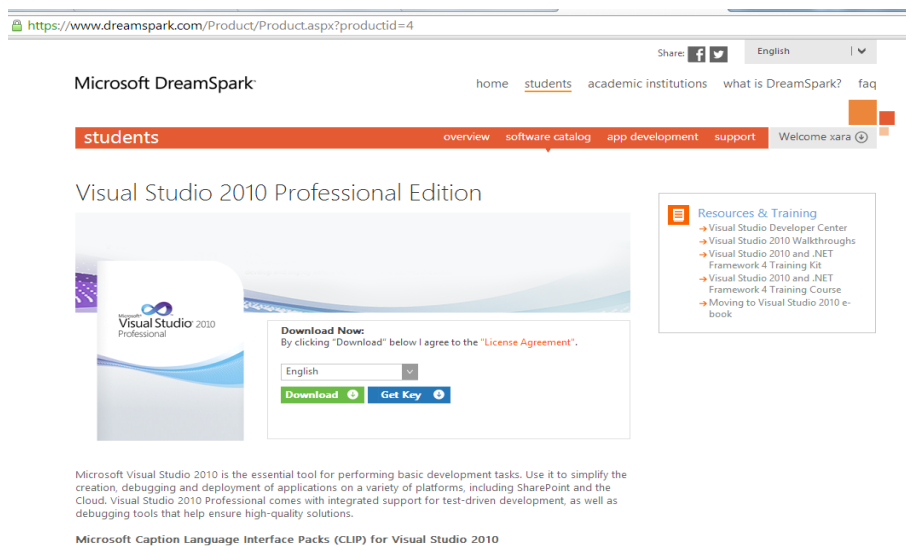
Εικόνα 14



- Έπειτα για να κατεβάσετε την **Visual Studio 2010** η διαδικασία είναι πιο απλή. Πηγαίνετε πίσω στο υπο-μενού student επιλέξτε το software catalog και ακολουθείτε την ίδια διαδικασία με πριν **χωρίς** όμως να εγκαταστήσετε το SDM (Secure Download Manager) εφ' όσον το εγκαταστήσατε μια φορά.

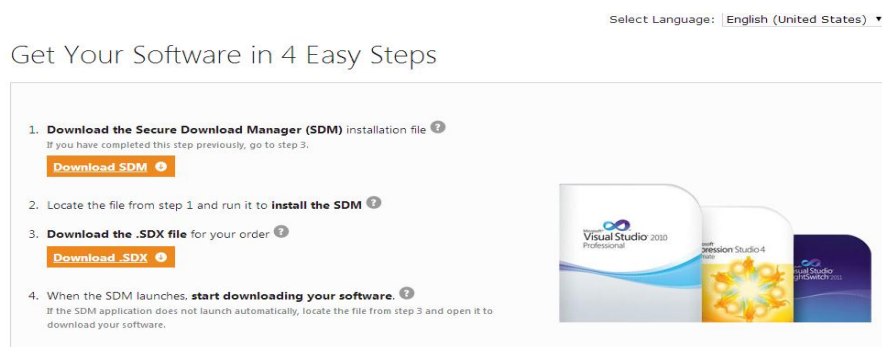


Εικόνα 15



Εικόνα 16

8



Εικόνα 179

**Download Your Software** Select Language: [English \(United States\)](#)


---

**Order Summary**

**Order Number:** 100312413783  
**Order Date:** 2014-08-21  
**Download Location:** C:\Users [Change Location](#)

---

**Items** ? **Need Help?**



**Microsoft Visual Studio 2010 Professional 32-bit (English)**

Download Ready **Start Download** ↻

Size: 2 GB

Εικόνα 18

**Download Your Software**


---

**Order Summary**

**Order Number:** 100312413783  
**Order Date:** 2014-08-21  
**Download Location:** C:\Users [Change Location](#)

---

**Items** ? **Need Help?**



**Microsoft Visual Studio 2010 Professional 32-bit (English)**

Done **Launch** ↻

Size: 2 GB C:\Users\Xara\Downloads

Εικόνα 10

## Εγκατάσταση Visual Studio 2010 Professional

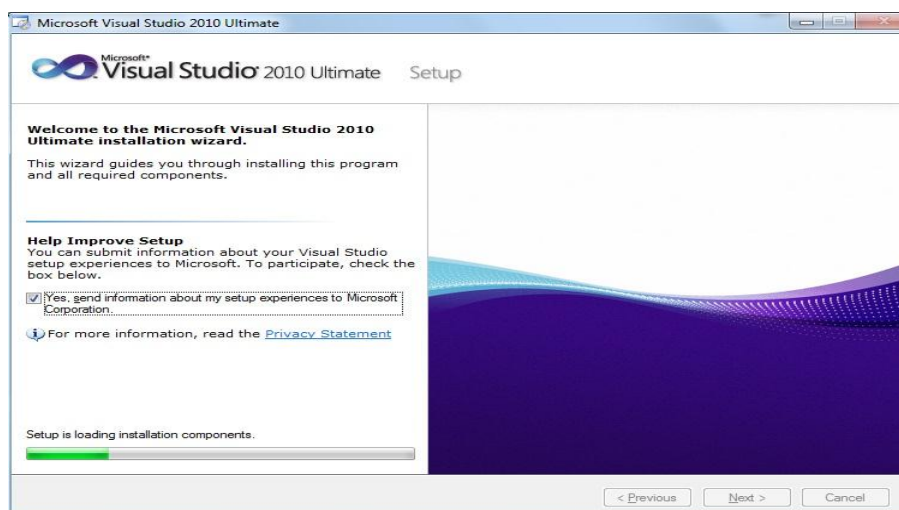
Αφού έχετε κατεβάσει το Visual Studio είστε έτοιμοι για την εγκατάσταση. Η διαδικασία που πρέπει να ακολουθήσετε είναι η εξής:

- Ανοίξτε το φάκελο και εκτελέστε το αρχείο setup.exe. Πατήστε στο **Install Microsoft Visual Studio 2010**.



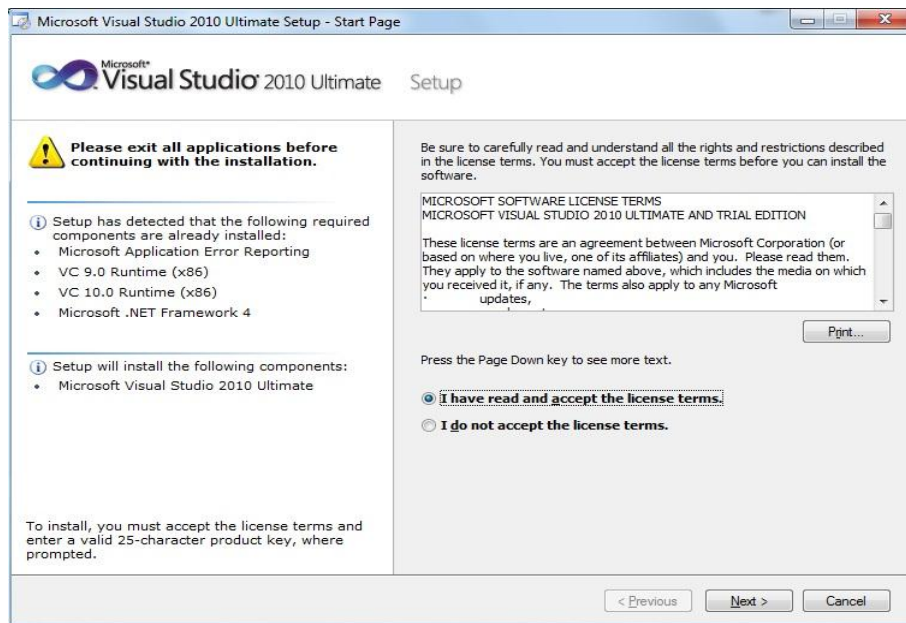
Εικόνα 20

- Περιμένετε να γίνουν οι ρυθμίσεις και πατάτε **Next**.



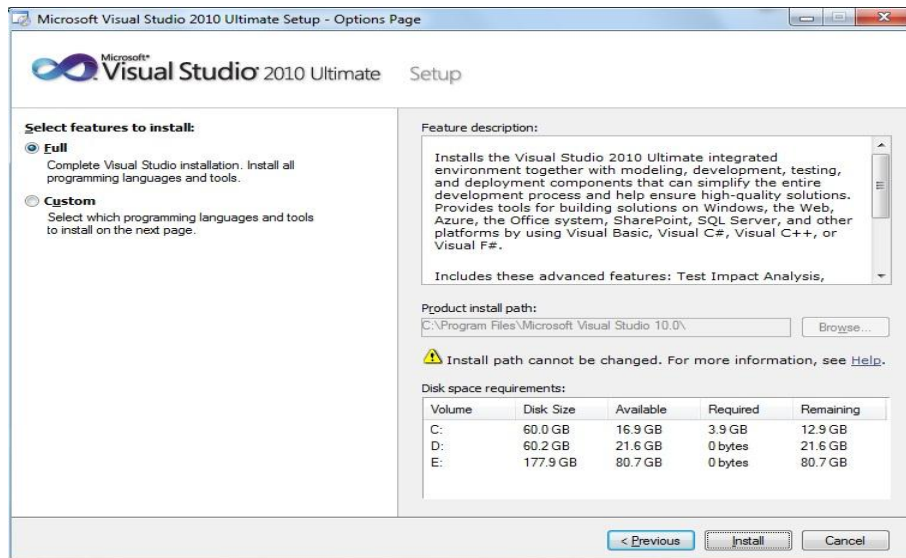
Εικόνα 21

- Επιλέγετε **I have read and accept the license terms** και πατάτε **Next**.



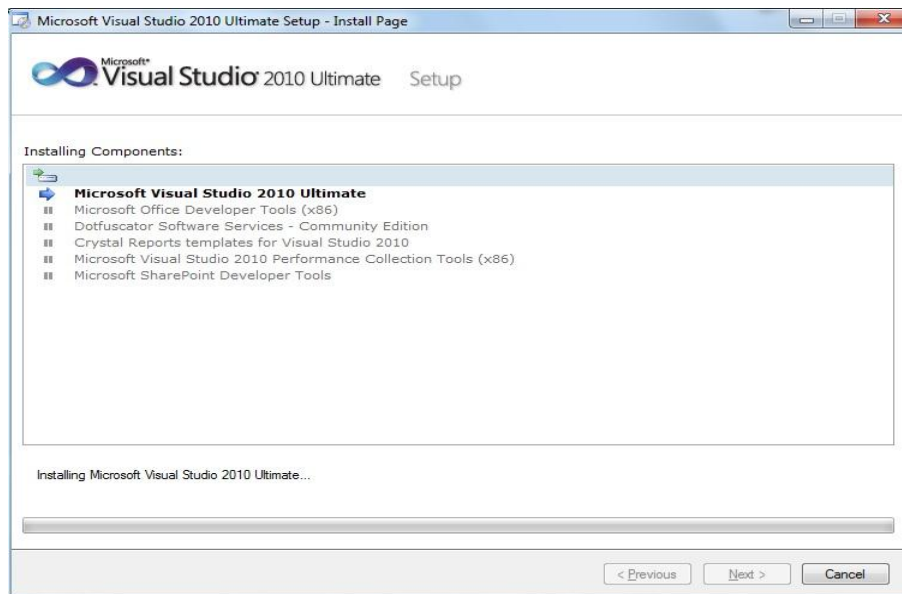
Εικόνα 22

- Επιλέγετε **Full** και πατάτε **Install**. Πρέπει να υπάρχουν 4GB ελεύθερα στον δίσκο.



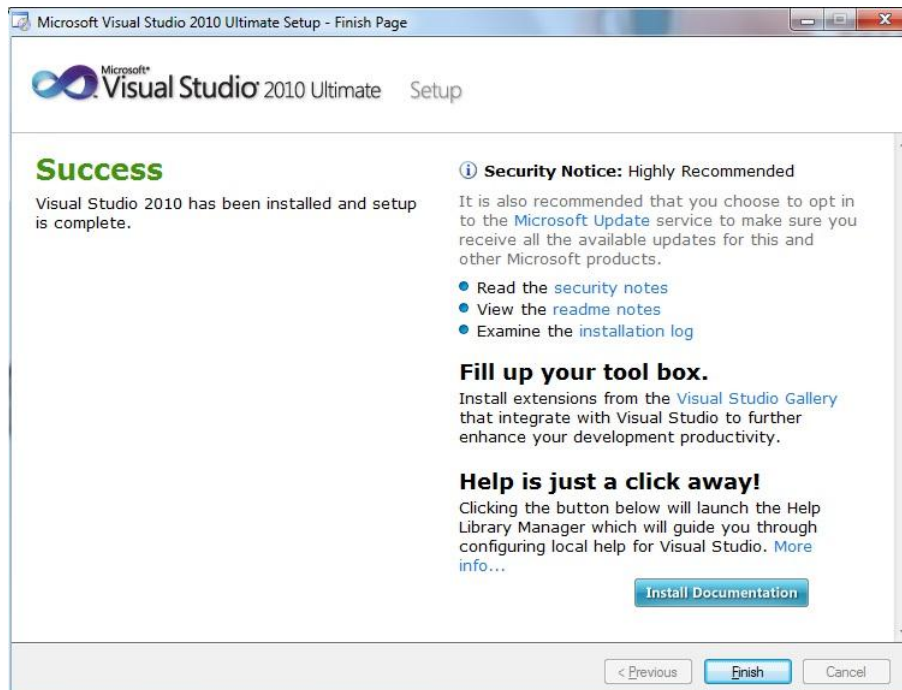
Εικόνα 23

- Περιμένετε ώσπου να ολοκληρωθεί η εγκατάσταση.



Εικόνα 24

- Πατάτε **Finish** και η εγκατάσταση έχει ολοκληρωθεί. Το πρόγραμμα είναι έτοιμο για χρήση.

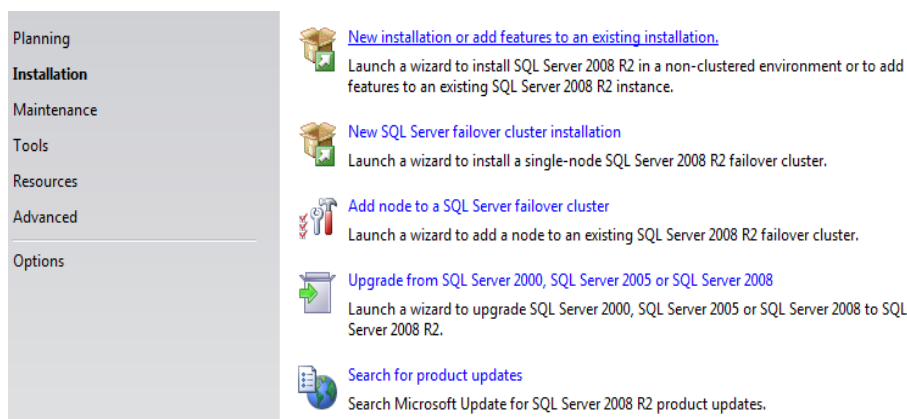


Εικόνα 25

## Εγκατάσταση Microsoft SQL Server 2008

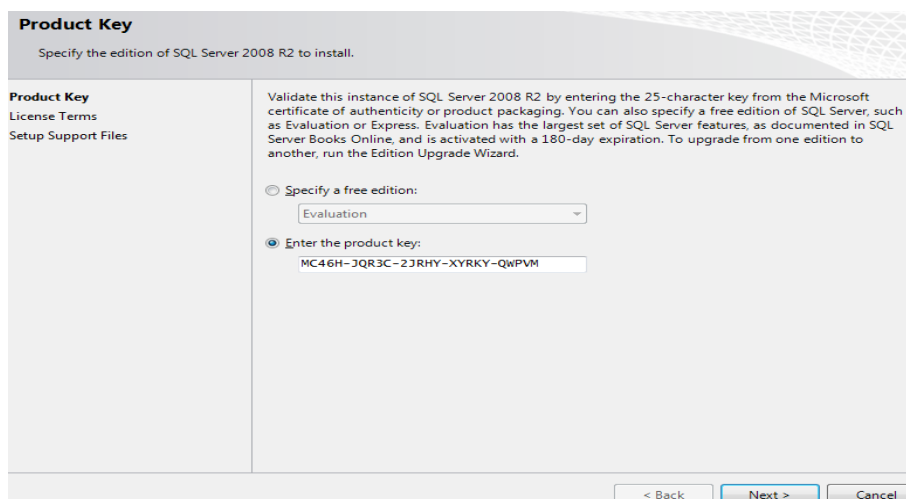
Έχετε κατεβάσει την SQL αλλά δεν την έχετε εγκαταστήσει. Η διαδικασία για την εγκατάσταση είναι η εξής:

- Εκτελούμε το αρχείο setup.
- Installation-> **New installation or add features to an existing installation.**



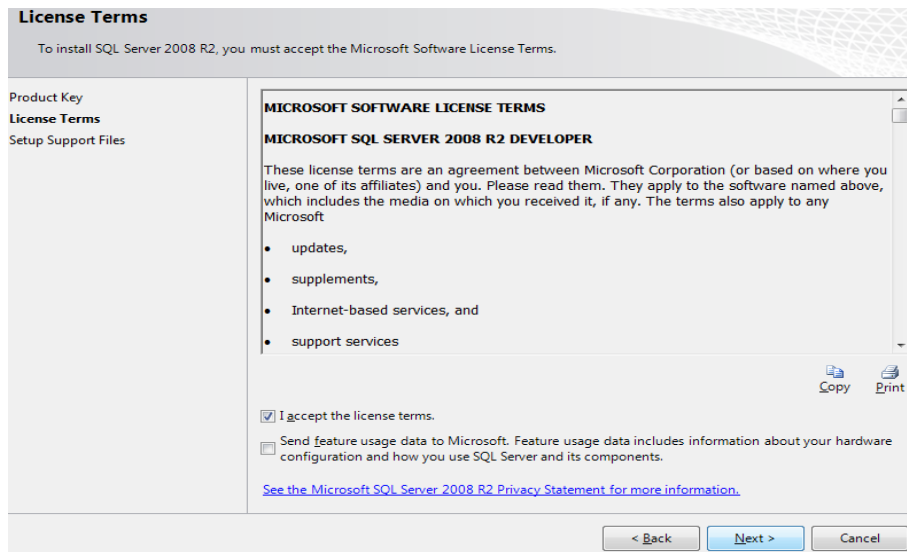
Εικόνα 26

- setup support rules -> **OK.**
- Product Key -> **Next.**



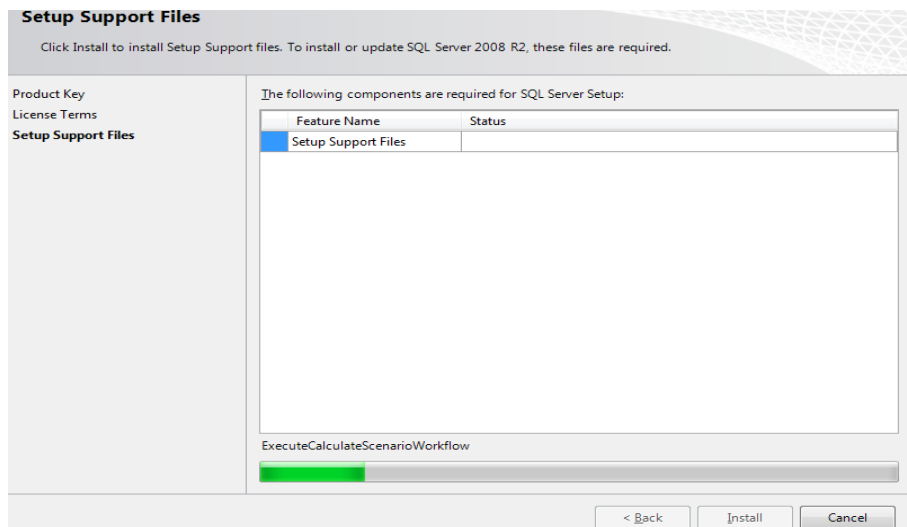
Εικόνα 27

➤ License Terms->I Accept->Next.



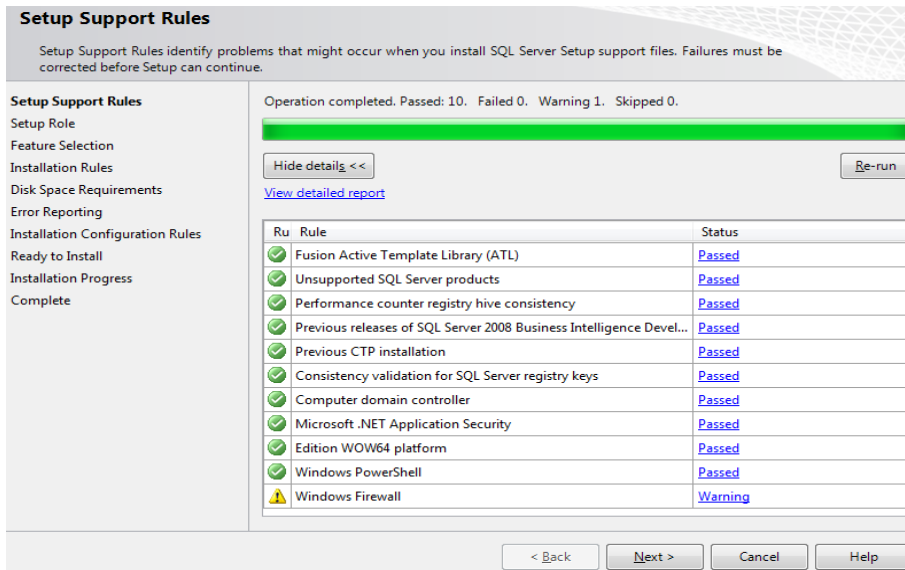
Εικόνα 28

➤ Setup support files-> Install.



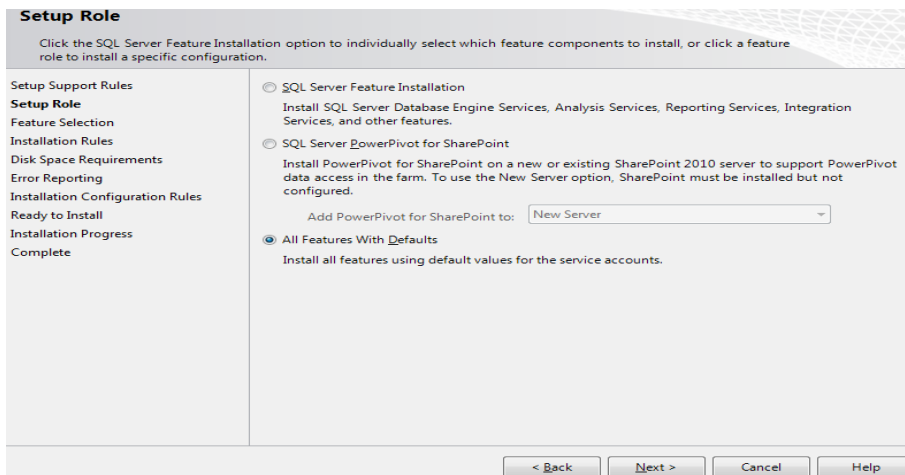
Εικόνα 29

➤ Setup support rules-> **Next.**



Εικόνα 30

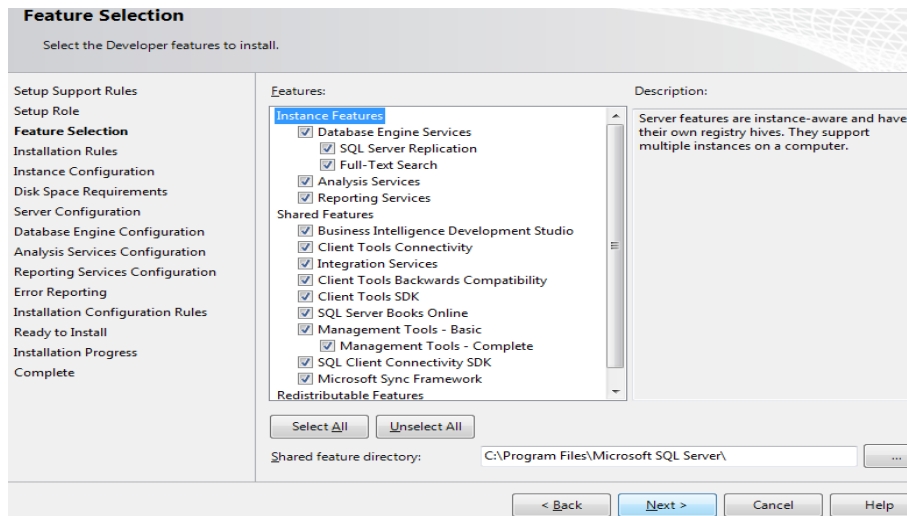
➤ Setup Role->**All Features with defaults->Next.**



Εικόνα 3111

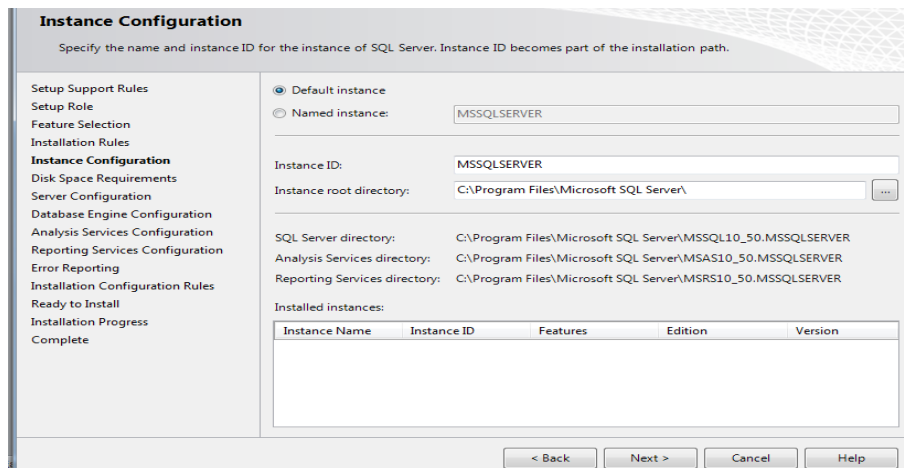


➤ Feature Selection->**Select all**->**Next**->**Next**.



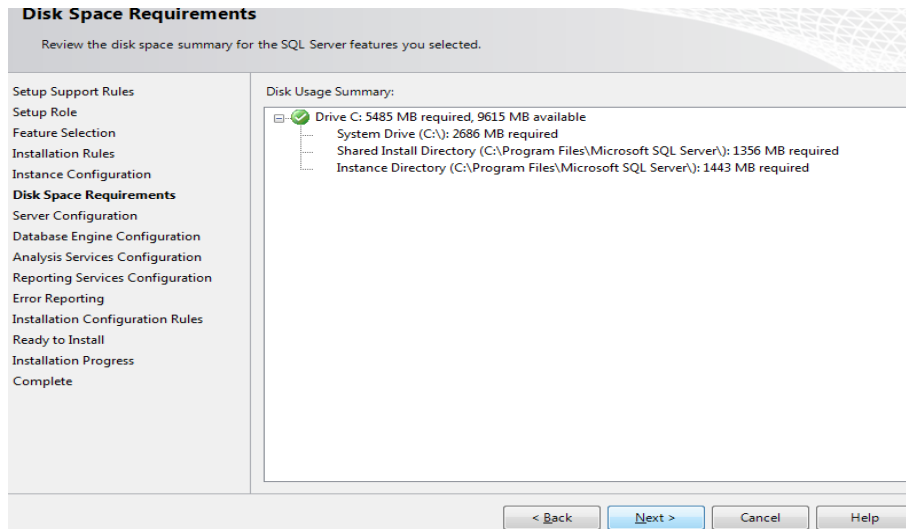
Εικόνα 32

➤ Instance Configuration ->Default instance -> **Next** .



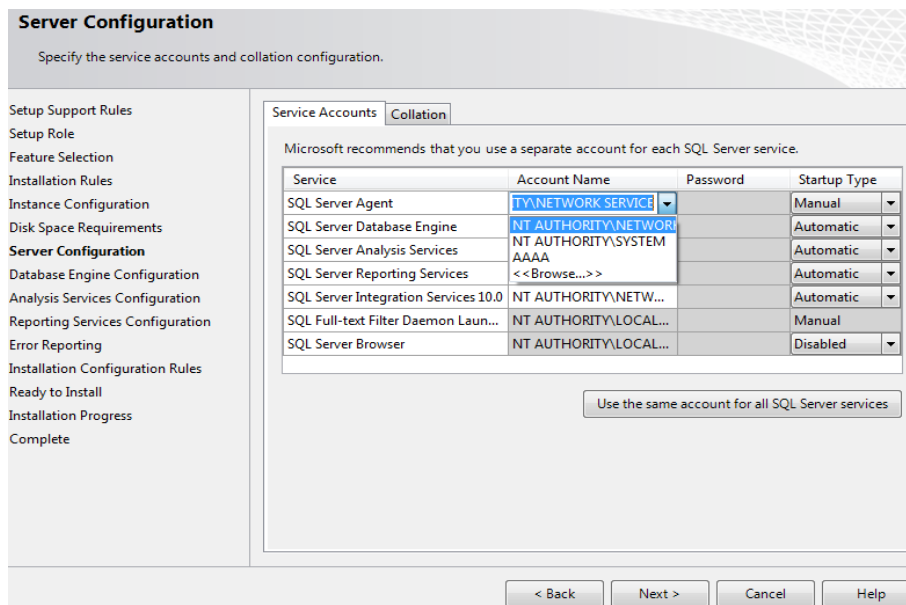
Εικόνα 33

- Disk Space requirements->Next.



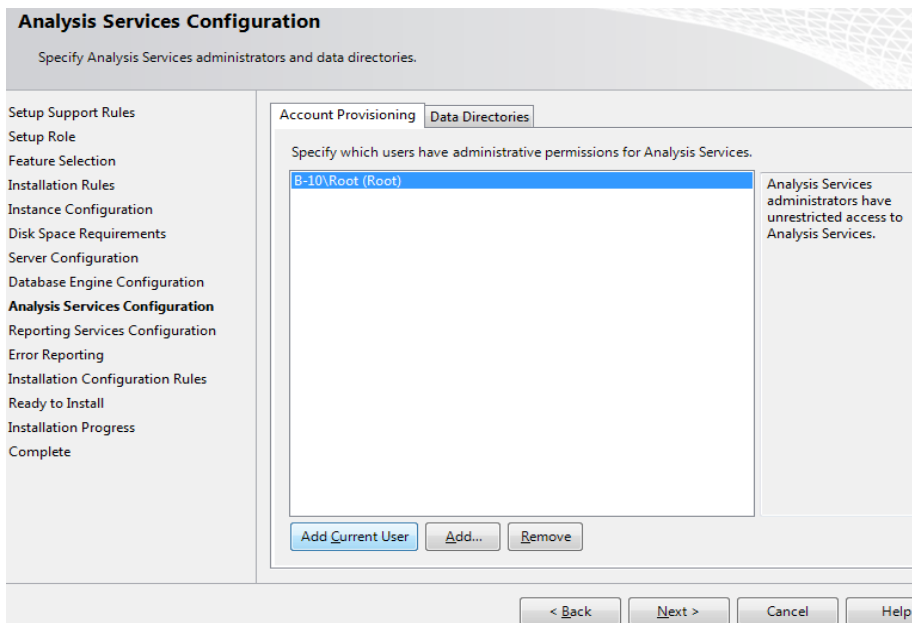
Εικόνα 34

- Server Configuration->Use the same account for all sql server services-> Account name-> NT AUTHORITY\SYSTEM-> OK
- Έπειτα πρέπει όλες οι επιλογές να έχουν NT AUTHORITY\SYSTEM-> Next-> Next.



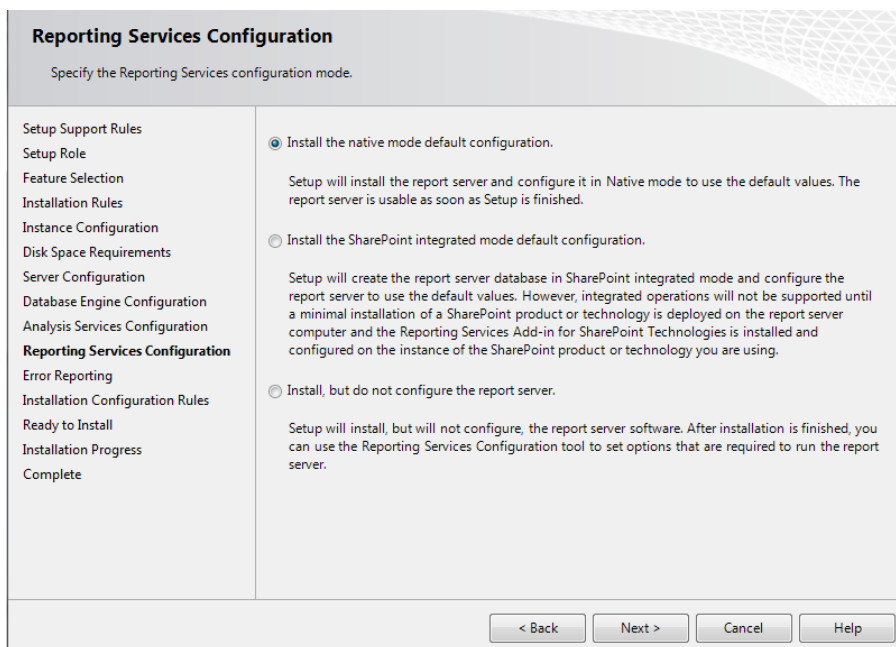
Εικόνα 35

- Analysis Services Configuration->**Add Current User->Next.**



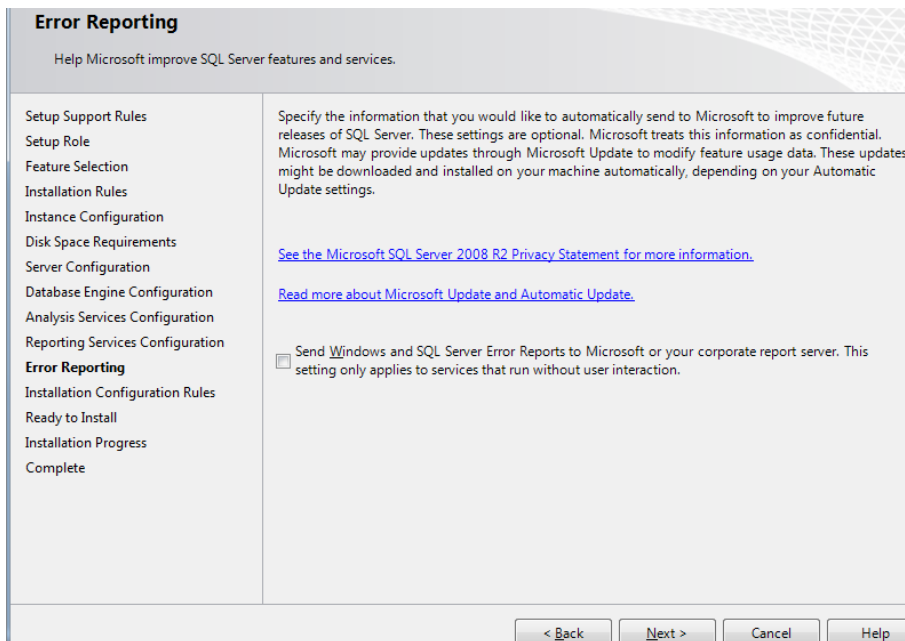
Εικόνα 36

- Install the native mode default configuration-> Next.



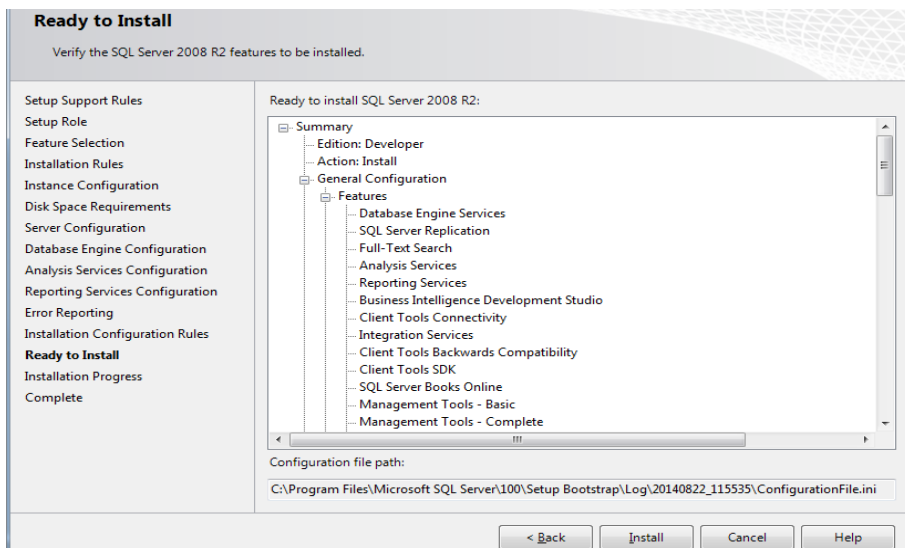
Εικόνα 37

- Error Reporting-> Next-> Next.



Εικόνα 38

- Ready to Install-> Install.



Εικόνα 39

- Περιμένετε μέχρι να ολοκληρωθεί η εγκατάσταση-> Close. Με την ολοκλήρωση το πρόγραμμα είναι έτοιμο για χρήση.

# Εγχειρίδιο Microsoft Access 2010

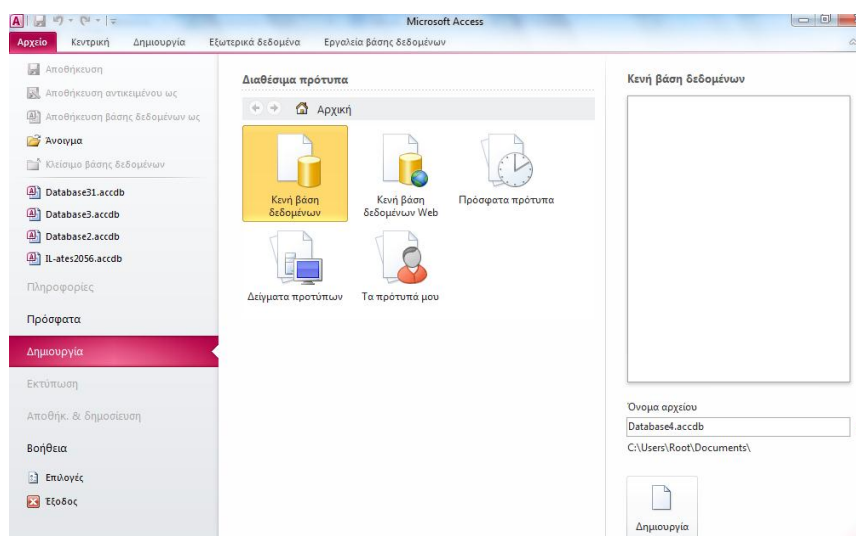
## Βάση δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από συστηματικά μορφοποιημένα σχετιζόμενα δεδομένα στα οποία είναι δυνατή η ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση. Ειδικότερα, στην επιστήμη της πληροφορικής και στην καθημερινή χρήση των ηλεκτρονικών υπολογιστών, με τον όρο βάσεις δεδομένων αναφερόμαστε σε οργανωμένες, διακριτές συλλογές σχετιζόμενων δεδομένων ηλεκτρονικά και ψηφιακά στο λογισμικό που χειρίζεται τέτοιες συλλογές (Σύστημα Διαχείρισης Βάσεων Δεδομένων, ή DBMS) και στο γνωστικό πεδίο που το μελετά.

Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει μέσω του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, δηλαδή τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων.

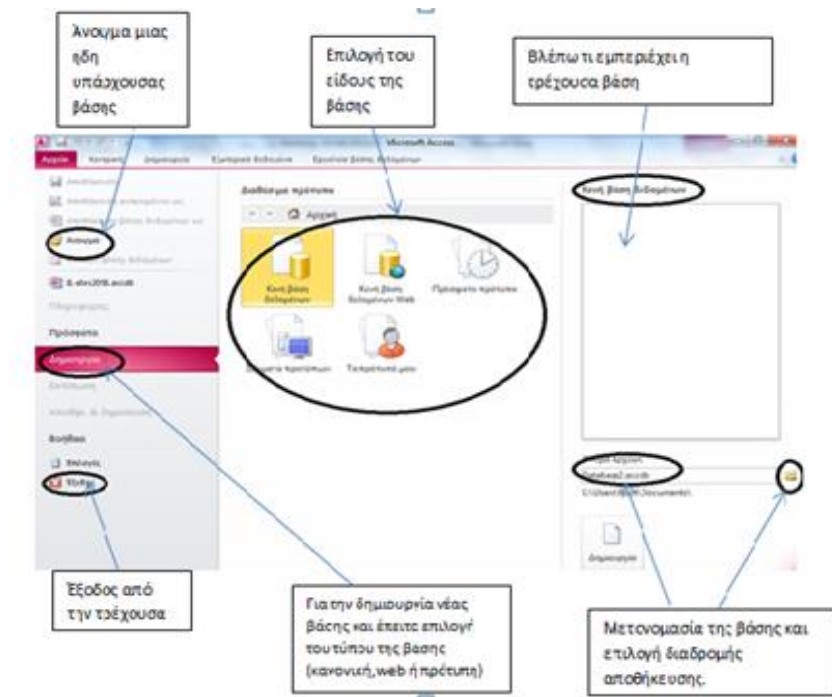
## Περιβάλλον της Access 2010

Όταν ξεκινάτε την Microsoft Access 2010, μπορείτε να δείτε την προβολή Microsoft Office Backstage, όπου μπορείτε να πάρετε πληροφορίες σχετικά με την τρέχουσα βάση δεδομένων, να δημιουργήσετε μια νέα βάση δεδομένων και να την μετονομάσετε καθώς και να ανοίξετε μια υπάρχουσα βάση δεδομένων.



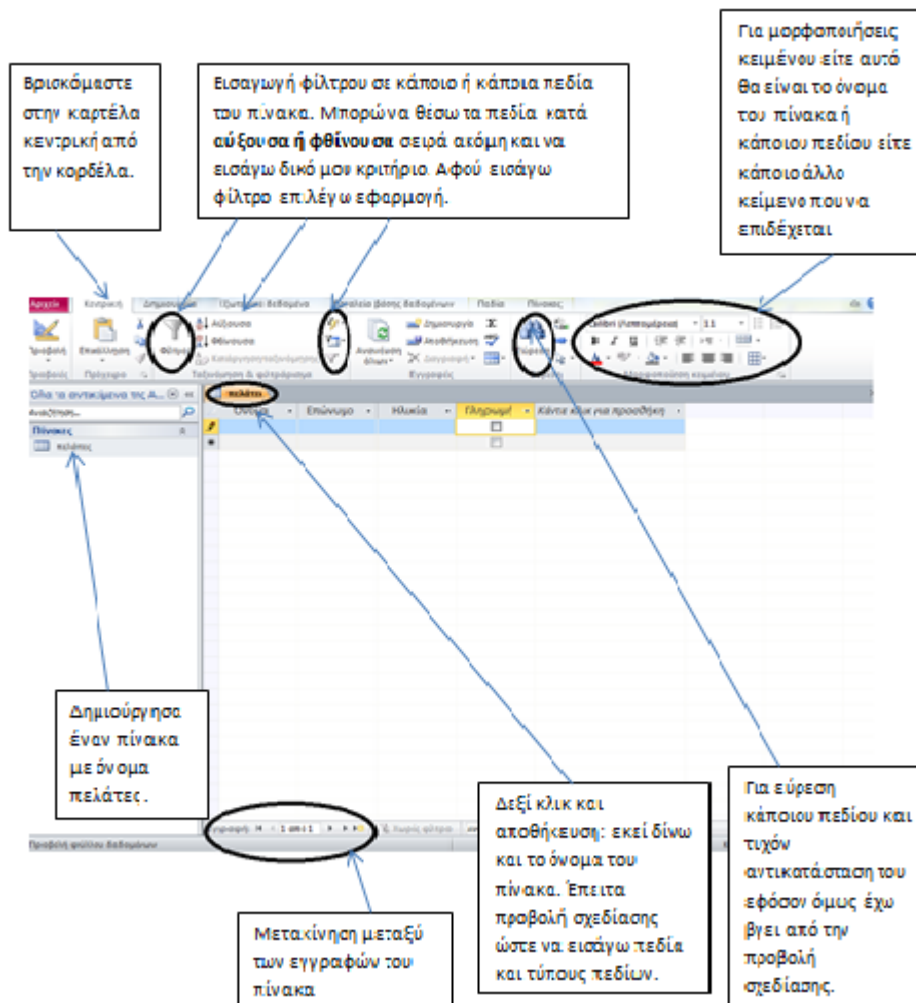
Εικόνα 1.1 Αρχικό περιβάλλον της Access

Βρίσκεστε σε ένα πρωταρχικό στάδιο για την δημιουργία μιας βάσης δεδομένων τη δήλωση του ονόματος της και της διαδρομής στην οποία θα είναι αποθηκευμένη.



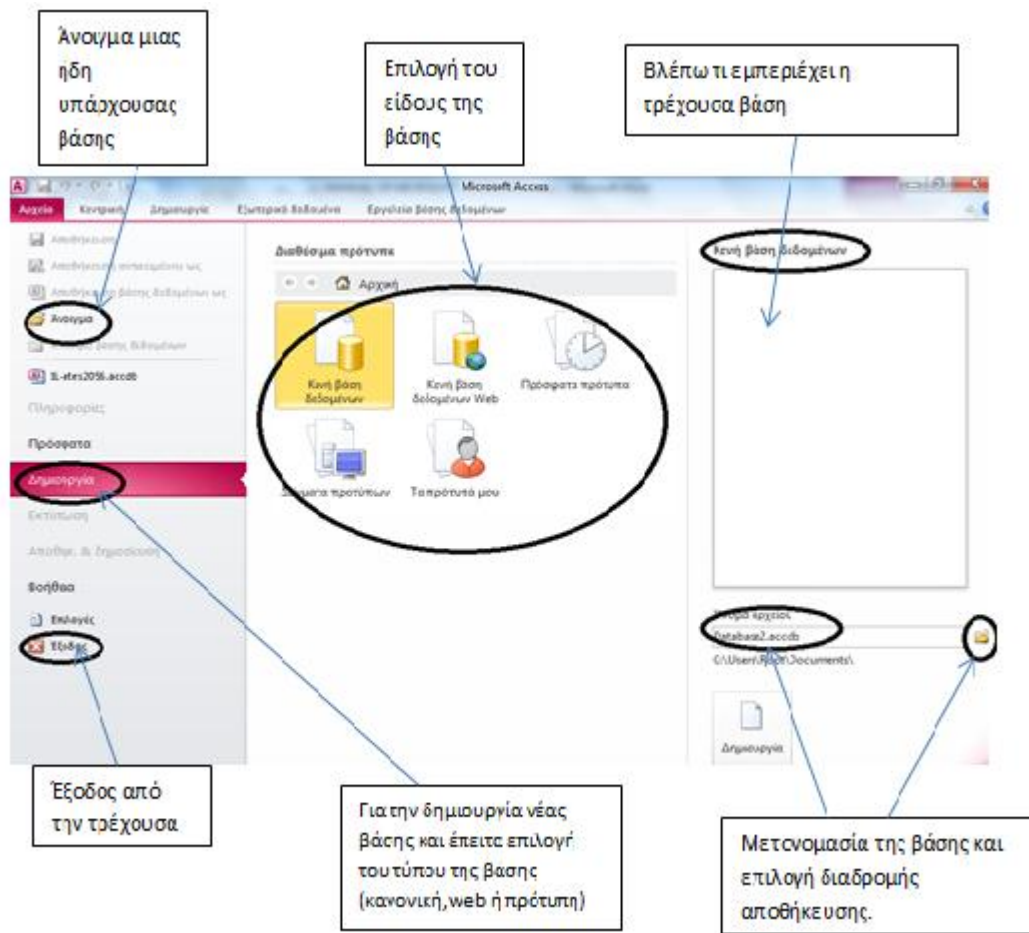
Εικόνα 1.2

Αφού ακολουθήσετε τα πρώτα βήματα για την δημιουργία μιας βάσης όπως είδατε στην εικόνα 1.2 συνεχίζετε με την δημιουργία των πινάκων που χρειάζεστε και εφαρμογή σε αυτούς διαφόρων χαρακτηριστικών.



Εικόνα 1.3

Στην καρτέλα δημιουργία θα βρείτε τους τρόπους δημιουργίας και σχεδίασης του πίνακα , των ερωτημάτων, της φόρμας ,της έκθεσης καθώς και της μακροεντολής.



Εικόνα 1.4



**Πίνακας-συλλογή των σχετικών δεδομένων:** Ένας πίνακας είναι μια συλλογή από στατικά δεδομένα για ένα συγκεκριμένο θέμα , όπως τα προϊόντα ή οι προμηθευτές . Μια ενιαία βάση δεδομένων μπορεί να περιέχει πολλούς αλληλένδετους πίνακες . Τα δεδομένα οργανώνονται σε ένα φύλλο δεδομένων πίνακα χρησιμοποιώντας στήλες ως πεδία και σειρές , όπως τα αρχεία . Ένα πεδίο είναι μια στήλη σε έναν πίνακα που περιέχει μια ενιαία κατηγορία των πληροφοριών . Η εγγραφή είναι σειρά του πίνακα που περιέχει όλα τα πεδία για ένα στοιχείο σε μια βάση δεδομένων.

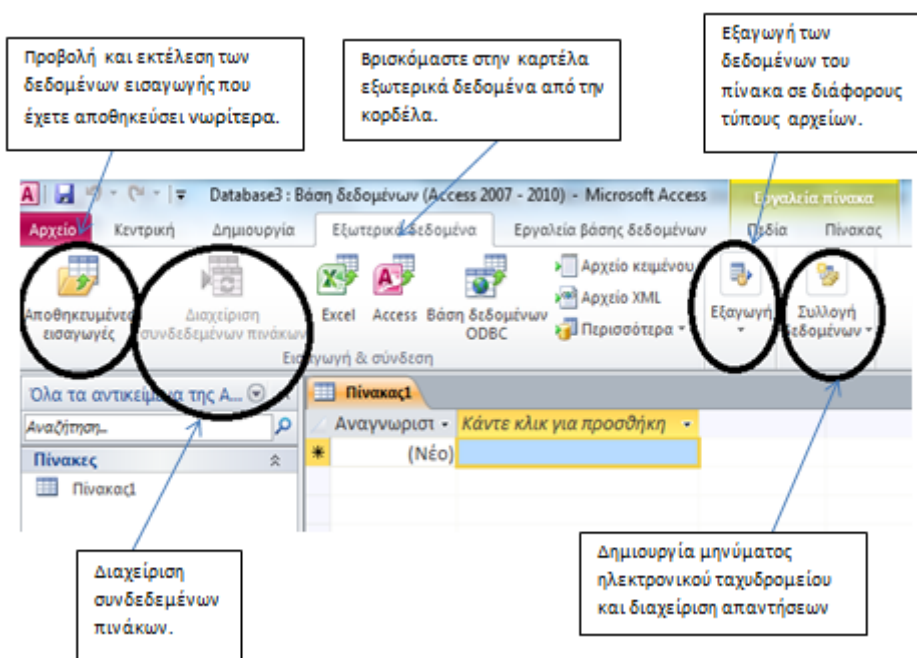
**Ερωτήματα- χειρίζονται τα δεδομένα εντός της βάσης δεδομένων:** Σχεδόν όλοι είναι εξοικειωμένοι με τους πίνακες, τις φόρμες και τις εκθέσεις, αλλά για την δημιουργία ενός ερωτήματος στην Access απαιτείται και τεχνική. Ένα ερώτημα μπορεί να θεωρηθεί ως σύνδεσμος μεταξύ του πίνακα και της φόρμας ή της έκθεσης. Εξάγει πληροφορίες και την καθιστά διαθέσιμη για μια φόρμα ή για μια έκθεση. Τα ερωτήματα χρησιμοποιούνται για την ανάκτηση δεδομένων, για εισαγωγή φίλτρου, για προβολή, για επεξεργασία καθώς και για ανάλυση των δεδομένων με διαφορετικούς τρόπους. Επιπλέον, τα ερωτήματα μπορούν να χειριστούν τα δεδομένα μέσω υπολογισμών. Μόλις ένα ερώτημα αποθηκευτεί και ανοίξει εκ νέου, μπορεί να ενημερώνεται αυτόματα όπως επίσης να τρέχει εκ νέου το ερώτημα για να εμφανίσει τις τελευταίες πληροφορίες.

**Φόρμα- Απλοποιεί την είσοδο των δεδομένων:** Έχει σχεδιαστεί κυρίως για την απλοποίηση της εισόδου , την επεξεργασία δεδομένων καθώς και την προβολή τους σε έναν πίνακα. Οι φόρμες περιέχουν στοιχεία ελέγχου , όπως πλαίσια ελέγχου , πλαίσια λίστας και κουμπιά επιλογής για να διευκολύνουν την απλούστευση των δεδομένων εισόδου . Τα πεδία σε μια φόρμα αντιστοιχούν στα πεδία ενός υποκείμενου πίνακα. Μια φόρμα μπορεί επίσης να περιέχει υπολογιζόμενα πεδία - κατηγορίες πληροφοριών που υπολογίζονται σε άλλους τομείς . Αυτά τα υπολογιζόμενα πεδία δεν είναι μέρος οποιουδήποτε πίνακα και εμφανίζονται μόνο στη φόρμα . Όταν τα δεδομένα προστεθούν, για να επεξεργαστούν ή διαγραφούν σε μια φόρμα , τότε γίνεται αυτόματη αλλαγή στον πίνακα.

**Έκθεση-Παρουσιάζει τα δεδομένα σε μια ποιο επαγγελματική μορφή εντύπων:** Μία έκθεση εμφανίζει πληροφορίες από τους πίνακες και τα ερωτήματα και χρησιμοποιείται κατά κύριο λόγο για επαγγελματικές πληροφορίες που εξάγει από τη βάση δεδομένων σε έντυπη μορφή . Αναφορές μπορούν να δημιουργηθούν από έναν πίνακα, ένα ερώτημα ή από οποιονδήποτε συνδυασμό αυτών για να παρουσιάσει τις σχέσεις μεταξύ των διαφόρων συνόλων δεδομένων. Υπολογισμοί μπορούν να γίνουν μέσα σε εκθέσεις όπως και με τη μορφή που εμφανίζονται μόνο στην έκθεση .

**Οι μακροεντολές και λειτουργικές μονάδες - Αυτοματοποίηση της βάσης δεδομένων:** Οι μακροεντολές και λειτουργικές μονάδες είναι για προχωρημένους χρήστες. Οι μακροεντολές επιτρέπουν σε έναν χρήστη να αυτοματοποιήσει μια σειρά από εργασίες βάσης δεδομένων με την έκδοση των εντολών. Μια μονάδα αυτοματοποιεί πολύπλοκες εργασίες και απαιτεί γνώσεις προγραμματισμού ηλεκτρονικών υπολογιστών . Κανένα από αυτά τα θέματα δεν καλύπτονται στο εν λόγω εγχειρίδιο.

Στην καρτέλα εξωτερικά δεδομένα θα βρείτε τον τρόπο εισαγωγής εξωτερικών αρχείων, την εξαγωγή τους σε άλλους τύπους αρχείων καθώς και δημιουργία μηνυμάτων ηλεκτρονικού ταχυδρομείου.



Εικόνα 1.5

## Κοινές χρήσεις της Microsoft Access 2010

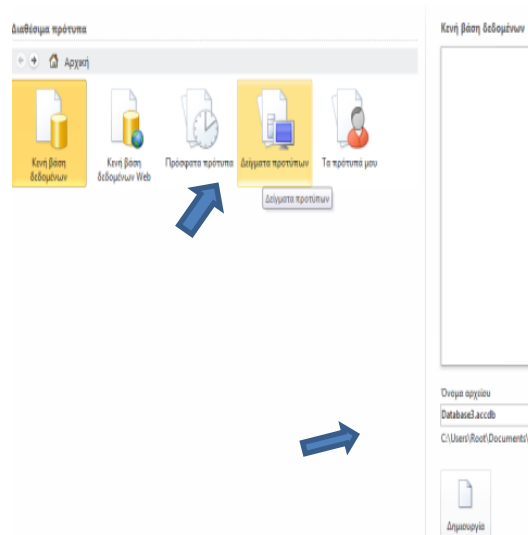
Οι χρήσεις της Access είναι πάρα πολλές και διευκολύνουν σε διάφορους τομείς της καθημερινότητας μας και όχι μόνο. Κάποιες από τις χρήσεις της είναι οι εξής:

- Διαχείριση επαφών
- Διαχείριση προσωπικού
- Προγραμματισμός πόρων
- Καταχωρήσεις ταινιών
- Λογαριασμοί πληρωτέοι/απαιτήσεις
- Πληροφόρηση πελατών
- Απογραφή έλεγχου
- Ώρα και χρεώσεις
- Για ιατρικούς καταλόγους
- Λίστες
- Σειρά προτεραιότητας
- Βιβλία τηλεφώνων
- Βιβλία συνταγών

## Δημιουργία Βάσης Δεδομένων

### Δημιουργία Βάσης Δεδομένων με χρήση προτύπου

1. Ξεκινήστε την Access
2. Κάντε κλικ στην καρτέλα Δείγματα Προτύπων στο παράθυρο εργασιών
3. Επιλέξτε το δείγμα που θέλετε με μονό κλικ
4. Πληκτρολογήστε το όνομα της βάσης δεδομένων στα δεξιά όπως βλέπουμε στην εικόνα και κάντε κλικ στο κουμπί Δημιουργία.
5. Έπειτα με διπλό κλικ ανοίξτε το πρότυπο σας
6. Τέλος ενεργοποιήστε το περιεχόμενο.

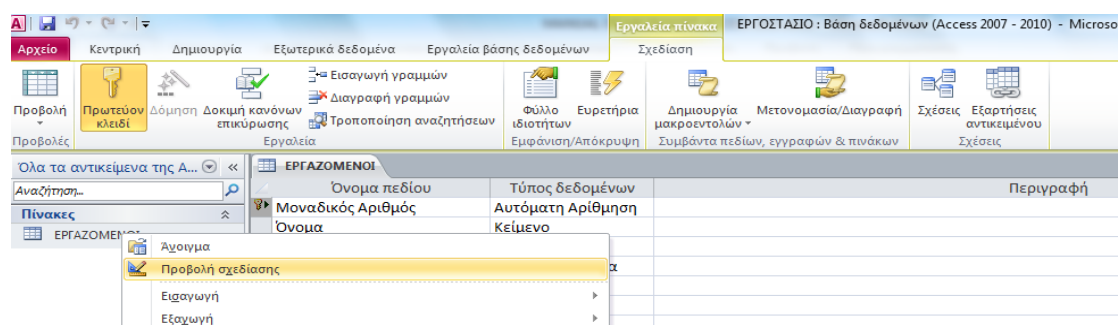


## Αποθήκευση μίας Βάσης Δεδομένων

Η Access αποθηκεύει αυτόματα είτε μη αποθηκευμένες αλλαγές στα δεδομένα ή ζητά από το χρήστη εάν οποιοδήποτε αντικείμενο πρέπει να σωθεί κατά το κλείσιμο της βάσης δεδομένων. Οι περισσότερες αλλαγές στη βάση δεδομένων αποθηκεύονται αυτόματα, ένα χαρακτηριστικό που μπορεί να είναι καλό και κακό. Είναι καλό για προφανείς λόγους, αλλά και κακό, διότι αν κάποιος διαγράψει κάτι κατά λάθος καταγράφει τα σχετικά στοιχεία χάνοντας. Τρόποι για την πρόληψη των απωλειών αυτών είναι να κάνετε backup εργασία, την εργασία διπλό έλεγχο και χρησιμοποιήστε το κουμπί Αναίρεση (Ctrl + Z) αμέσως μετά από ένα λάθος.

## Πίνακας

### Δημιουργία πίνακα σε προβολή σχεδίασης



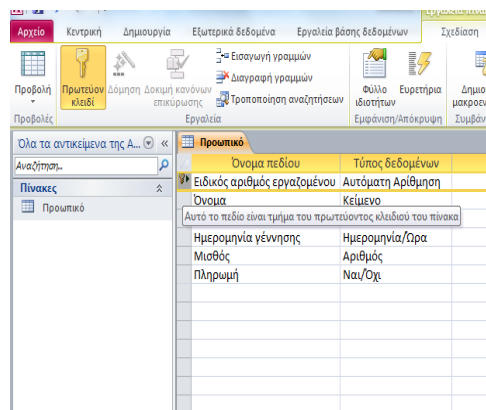
### Επιλογή ιδανικού κλειδιού

Μια βέλτιστη πρακτική στο σχεδιασμό βάσεων δεδομένων είναι η χρήση ενός εσωτερικά δημιουργημένου πρωτεύοντος κλειδιού. Το πρωτεύον κλειδί είναι ένα πεδίο ενός πίνακα της Access, που χαρακτηρίζει μοναδικά μία εγγραφή μέσα σ' ολόκληρο τον πίνακα. Το πεδίο αυτό πρέπει να είναι ένα μοναδικό αναγνωριστικό για κάθε εγγραφή. Μια καλή επιλογή για πιθανό πρωτεύον κλειδί πρέπει να διαθέτει κάποια χαρακτηριστικά:

1. Να αναγνωρίζει κάθε γραμμή μοναδικά.
2. Δεν πρέπει να είναι ποτέ κενό ή με μηδενική τιμή — να περιέχει πάντα μια τιμή.
3. Οι τιμές που περιέχει να αλλάζουν σπάνια (ιδανικά, ποτέ).

## Εισαγωγή πρωτεύοντος κλειδιού

1. Ανοίξτε τη βάση δεδομένων που θέλετε να τροποποιήσετε.
2. Στο Παράθυρο περιήγησης, κάντε δεξί κλικ στον πίνακα όπου θέλετε να ορίσετε το πρωτεύον κλειδί και κάντε κλικ στην επιλογή Προβολή σχεδίασης, από το μενού συντόμευσης. (Εάν δεν βλέπετε το Παράθυρο περιήγησης, πατήστε το πλήκτρο F11 για να το εμφανίσετε).
3. Επιλέξτε το πεδίο ή τα πεδία που θέλετε να χρησιμοποιήσετε ως πρωτεύοντα κλειδιά. Για να επιλέξετε ένα πεδίο, κάντε κλικ στο δείκτη επιλογής γραμμής του επιθυμητού πεδίου. Για να επιλέξετε περισσότερα από ένα πεδία, κρατήστε πατημένο το πλήκτρο Ctrl και στη συνέχεια κάντε κλικ στον επιλογέα γραμμής κάθε πεδίου.
4. Στην καρτέλα Σχεδίαση, στην ομάδα Εργαλεία, κάντε κλικ στην επιλογή Πρωτεύον κλειδί.



Στα αριστερά του πεδίου ή των πεδίων που ορίσατε ως πρωτεύον κλειδί θα εμφανιστεί ένας δείκτης κλειδιού.

## Κατάργηση πρωτεύοντος κλειδιού

Όταν καταργείτε το πρωτεύον κλειδί, το πεδίο ή τα πεδία που προηγουμένως λειτουργούσε/-αν ως πρωτεύον κλειδί δεν παρέχουν πλέον τα κύρια μέσα αναγνώρισης μιας εγγραφής. Με την κατάργηση του πρωτεύοντος κλειδιού, απλώς καταργείται η αντιστοίχιση πρωτεύοντος κλειδιού από αυτό το πεδίο/αυτά τα πεδία καθώς και το ευρετήριο που δημιουργήθηκε για το πρωτεύον κλειδί. Αλλά δεν πραγματοποιείτε διαγραφή των πεδίων αυτών. Η διαδικασία γίνεται ως εξής:

1. Επιβεβαίωση της μη ύπαρξης σχέσεως με άλλο πίνακα γιατί σε μία τέτοια περίπτωση η Access θα βγάλει μήνυμα σφάλματος και θα προειδοποιεί ώστε πριν την διαγραφή του πρωτεύοντος κλειδιού να γίνει και η διαγραφή της σχέσης.

Για την διαγραφή των σχέσεων μεταξύ πινάκων ακολουθήστε τα εξής βήματα:

- i. Εάν οι πίνακες που συμμετέχουν στη σχέση πίνακα είναι ανοικτοί, κλείστε τους. Δεν μπορείτε να διαγράψετε μια σχέση πίνακα μεταξύ ανοικτών πινάκων.
- ii. Στην καρτέλα Εργαλεία βάσης δεδομένων, στην ομάδα Σχέσεις, επιλέξτε Σχέσεις.
- iii. Εάν οι πίνακες που συμμετέχουν στη σχέση πινάκων δεν εμφανίζονται, στην καρτέλα Σχεδίαση, στην ομάδα Σχέσεις, κάντε κλικ στην επιλογή Εμφάνιση πίνακα.
- iv. Επιλέξτε τους πίνακες που θα προσθέσετε στο παράθυρο διαλόγου Εμφάνιση πίνακα, κάντε κλικ στην εντολή Προσθήκη και στη συνέχεια, κάντε κλικ στο κουμπί Κλείσιμο.
- v. Κάντε κλικ στη γραμμή της σχέσης πίνακα που θέλετε να διαγράψετε (η γραμμή θα γίνει έντονη, όταν την επιλέξετε) και στη συνέχεια πατήστε το πλήκτρο Delete.
- vi. Έπειτα κλείστε την καρτέλα Σχεδίαση καθώς και στην ομάδα Σχέσεις, κάντε κλικ στην επιλογή Κλείσιμο.

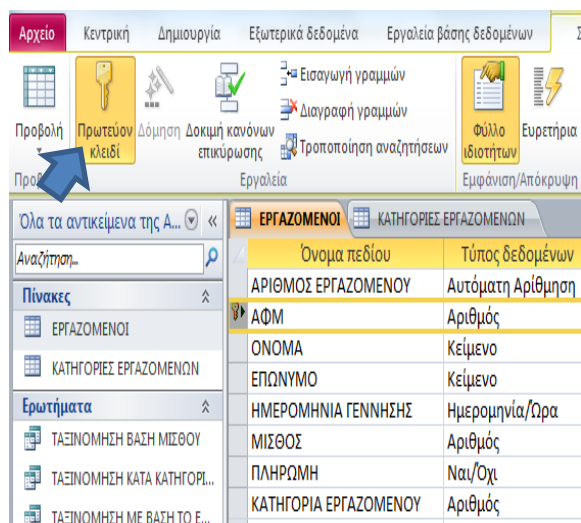
Αφού έχετε βεβαιωθεί για την μη ύπαρξη σχέσης μεταξύ πινάκων είστε έτοιμοι να συνεχίσετε για την απαλοιφή του πρωτεύοντος κλειδιού.

2. Στο Παράθυρο περιήγησης, κάντε δεξί κλικ στον πίνακα όπου θέλετε να καταργήσετε το πρωτεύον κλειδί και κάντε κλικ στην επιλογή Προβολή σχεδίασης, στο μενού συντόμευσης.
3. Κάντε κλικ στο δείκτη επιλογής γραμμής για το τρέχον πρωτεύον κλειδί.
4. Εάν το πρωτεύον κλειδί αποτελείται μόνο από ένα πεδίο, κάντε κλικ στο κουμπί επιλογής γραμμών του συγκεκριμένου πεδίου.
5. Εάν το πρωτεύον κλειδί αποτελείται από πολλά πεδία, κάντε κλικ στο κουμπί επιλογής γραμμών όλων των πεδίων του πρωτεύοντος κλειδιού.
6. Στην καρτέλα Σχεδίαση, στην ομάδα Εργαλεία, κάντε κλικ στην επιλογή Πρωτεύον κλειδί.
7. Ο δείκτης κλειδιού καταργείται από το πεδίο ή τα πεδία που είχατε ορίσει προηγουμένως ως πρωτεύον κλειδί.

## Αλλαγή πρωτεύοντος κλειδιού

Για να αλλάξετε το ήδη υπάρχον πρωτεύον κλειδί και να ορίσετε κάποιο άλλο πεδίο ως πρωτεύον κάνετε τα εξής :

1. Μπαίνετε σε προβολή σχεδίασης
2. Πηγαίνετε στο πεδίο που υπάρχει το πρωτεύον κλειδί το επιλέγετε
3. Έπειτα πατάτε στην κορδέλα στο εικονίδιο Πρωτεύον κλειδί για να από-επιλεχθεί
4. Αφού εκτελέσετε την παραπάνω διαδικασία επιλέξετε το νέο πεδίο στο οποίο θέλετε να ορίσετε τώρα το πρωτεύον κλειδί και εκτελέστε ξανά το βήμα 3 ώστε αυτήν την φορά να επιλεχθεί το Πρωτεύον κλειδί.

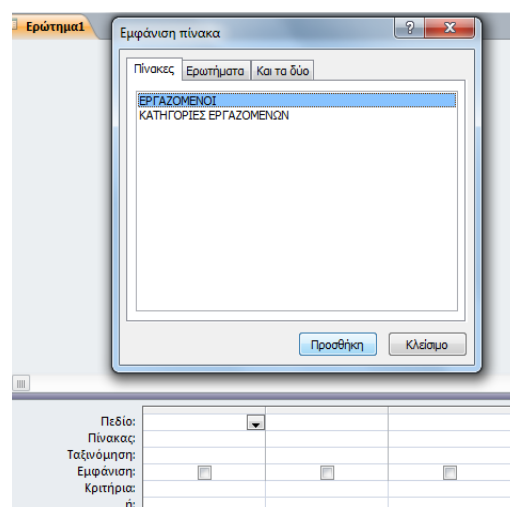


## Ερωτήματα

### Δημιουργία ερωτήματος σε προβολή σχεδίασης

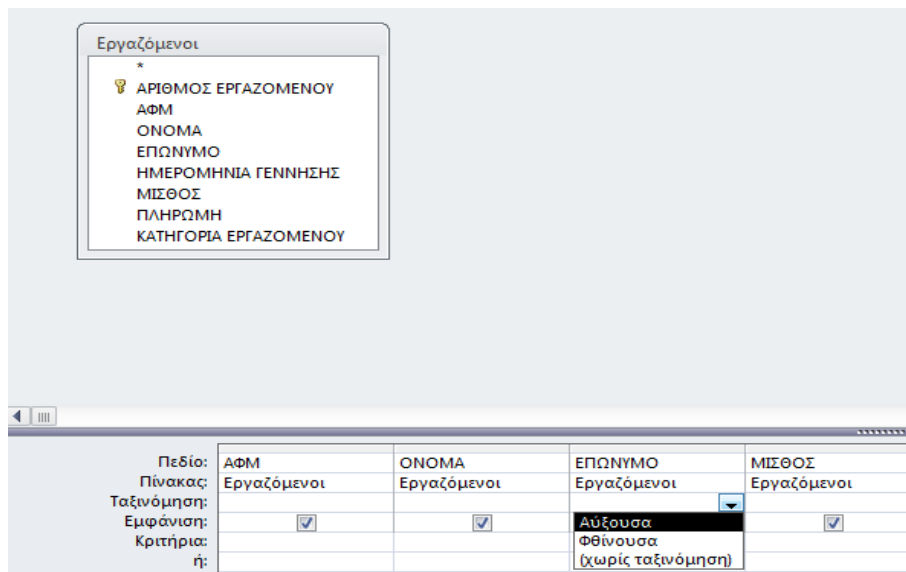
Τα ερωτήματα εξάγουν δεδομένα από τον πίνακα βασιζόμενα σε κριτήρια τα οποία παρέχει ο χρήστης .

1. Επιλέγετε από την κορδέλα την καρτέλα δημιουργία -> σχεδίαση ερωτήματος.
2. Επιλέγετε τον πίνακα ή τους πίνακες που θέλετε πατάτε προσθήκη και κλείσιμο.
3. Έπειτα επιλέγετε τα αντίστοιχα πεδία που θέλετε να εμφανιστούν με διπλό κλικ επάνω τους .
4. Τέλος στο κάτω μέρος του παραθύρου εισάγετε κάποιο κριτήριο ή κάποια ταξινόμηση που επιθυμείτε.



Παρακάτω παρουσιάζονται κάποια παραδείγματα ερωτημάτων.

### Παράδειγμα 1<sup>ο</sup> : Ταξινόμηση με βάση το επώνυμο

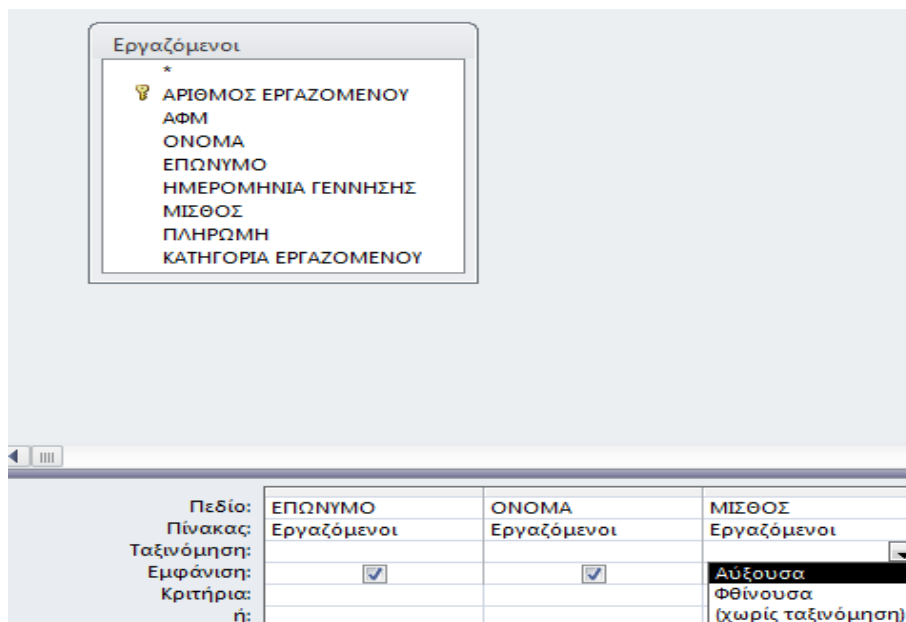


The screenshot shows a window titled "Εργαζόμενοι" with a list of fields: ΑΡΙΘΜΟΣ ΕΡΓΑΖΟΜΕΝΟΥ, ΑΦΜ, ΟΝΟΜΑ, ΕΠΩΝΥΜΟ, ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ, ΜΙΣΘΟΣ, ΠΛΗΡΩΜΗ, and ΚΑΤΗΓΟΡΙΑ ΕΡΓΑΖΟΜΕΝΟΥ. Below the window is a table with the following structure:

Πεδίο:	ΑΦΜ	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΜΙΣΘΟΣ
Πίνακας:	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι
Ταξινόμηση:			▼	
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Αύξουσα	<input checked="" type="checkbox"/>
Κριτήριο:			Φθίνουσα	
ή:			(χωρίς ταξινόμηση)	

Παράδειγμα 1<sup>ο</sup>

### Παράδειγμα 2<sup>ο</sup> : Ταξινόμηση με βάση τον μισθό



The screenshot shows the same window and table as in the first example, but the sorting is now based on salary. The table structure is as follows:

Πεδίο:	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΜΙΣΘΟΣ
Πίνακας:	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι
Ταξινόμηση:			▼
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Αύξουσα
Κριτήριο:			Φθίνουσα
ή:			(χωρίς ταξινόμηση)

Παράδειγμα 2<sup>ο</sup>



**Παράδειγμα 3<sup>ο</sup>** : Ταξινόμηση με κριτήρια η ημερομηνία γέννησης να είναι μεγαλύτερη από 01/05/1987.

Εργαζόμενοι \*

- 🔑 ΑΡΙΘΜΟΣ ΕΡΓΑΖΟΜΕΝΟΥ
- ΑΦΜ
- ΟΝΟΜΑ
- ΕΠΩΝΥΜΟ
- ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ
- ΜΙΣΘΟΣ
- ΠΛΗΡΩΜΗ
- ΚΑΤΗΓΟΡΙΑ ΕΡΓΑΖΟΜΕΝΟΥ

Πεδίο:	ΟΝΟΜΑ	ΕΠΩΝΥΜΟ	ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ
Πίνακας:	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι
Ταξινόμηση:			
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Κριτήρια:			>#1/5/1987#
ή:			

**Παράδειγμα 3<sup>ο</sup>**

**Παράδειγμα 4<sup>ο</sup>** : Ταξινόμηση με κριτήρια η ηλικία να είναι μεγαλύτερη από 1978 **ΚΑΙ** ο μισθός μεγαλύτερος από 1500. (Κριτήριο ταξινόμησης **AND**).

Εργαζόμενοι \*

- 🔑 ΑΡΙΘΜΟΣ ΕΡΓΑΖΟΜΕΝΟΥ
- ΑΦΜ
- ΟΝΟΜΑ
- ΕΠΩΝΥΜΟ
- ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ
- ΜΙΣΘΟΣ
- ΠΛΗΡΩΜΗ
- ΚΑΤΗΓΟΡΙΑ ΕΡΓΑΖΟΜΕΝΟΥ

Πεδίο:	ΕΠΩΝΥΜΟ	ΟΝΟΜΑ	ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ	ΜΙΣΘΟΣ
Πίνακας:	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι
Ταξινόμηση:				
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Κριτήρια:			>1978	<1500
ή:				

**Παράδειγμα 4<sup>ο</sup>**

**Παράδειγμα 5<sup>ο</sup>** : Ταξινόμηση με κριτήρια η ηλικία να είναι μεγαλύτερη από 1978 **Ή** ο μισθός μεγαλύτερος από 1500. (Κριτήριο ταξινόμησης **OR**).

Πεδίο:	ONOMA	ΕΠΩΝΥΜΟ	ΚΑΤΗΓΟΡΙΑ ΕΡΓΑΖΟΜΕΝΟΥ
Πίνακας:	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι
Ταξινόμηση:			
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Κριτήρια:			=1
ή:		Like 'ΠΑΠΑ*'	

**Παράδειγμα 5<sup>ο</sup>**

**Παράδειγμα 6<sup>ο</sup>** : Ταξινόμηση με κριτήρια το **ΟΝΟΜΑ** να περιέχει το γράμμα 'Α' **ΚΑΙ** το πεδίο πληρωμή να είναι Αληθές **Ή** το **ΕΠΩΝΥΜΟ** να ξεκινάει με το γράμμα 'Π'. (Κριτήρια ταξινόμησης **AND** και **OR**).

Πεδίο:	ONOMA	ΕΠΩΝΥΜΟ	ΠΛΗΡΩΜΗ
Πίνακας:	Εργαζόμενοι	Εργαζόμενοι	Εργαζόμενοι
Ταξινόμηση:			
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Κριτήρια:	Like '*A*'		Αληθές
ή:		Like 'Π*'	

**Παράδειγμα 6<sup>ο</sup>**

Το παραπάνω ερώτημα μπορεί να γραφεί σε SQL. Στην κορδέλα πατάτε το βελάκι κάτω από την προβολή και μετά Προβολή SQL.

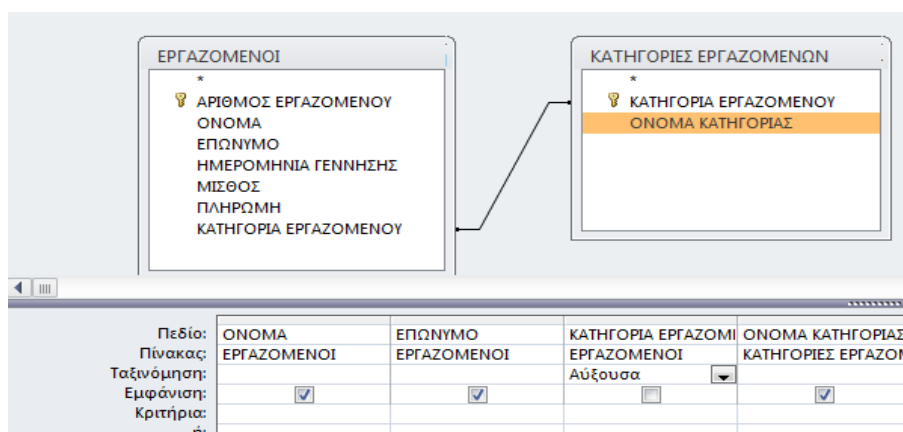
Με αυτόν τον τρόπο εμφανίζει τα παρακάτω:

```
SELECT Εργαζόμενοι.ΟΝΟΜΑ, Εργαζόμενοι.ΕΠΩΝΥΜΟ,
Εργαζόμενοι.ΠΛΗΡΩΜΗ

FROM Εργαζόμενοι

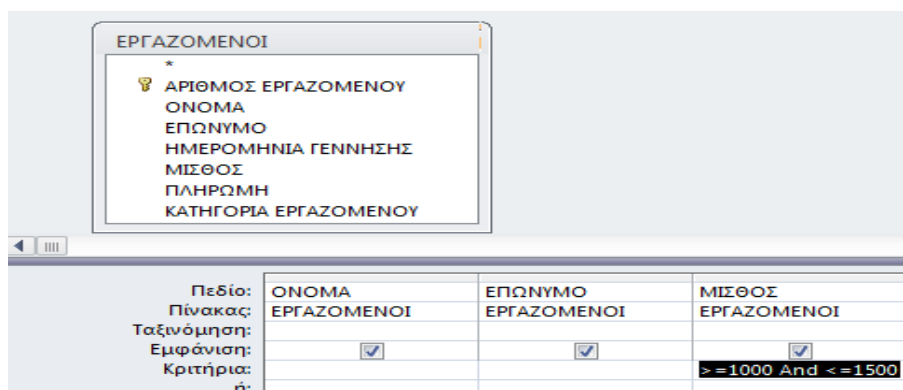
WHERE (((Εργαζόμενοι.ΟΝΟΜΑ) Like '*A*') AND
((Εργαζόμενοι.ΠΛΗΡΩΜΗ)=True)) OR
(((Εργαζόμενοι.ΕΠΩΝΥΜΟ) Like 'Π*'));
```

**Παράδειγμα 7<sup>ο</sup>** : Ταξινόμηση με βάση τη κατηγορία εργαζομένου. Εδώ βλέπετε και μία σχέση πινάκων το οποίο θα παρουσιαστεί στο επόμενο κεφάλαιο.



**Παράδειγμα 7<sup>ο</sup>**

**Παράδειγμα 8<sup>ο</sup>** : Ταξινόμηση με κριτήριο τον μισθό ο οποίος να κυμαίνεται μεταξύ των τιμών 1000 και 1500.

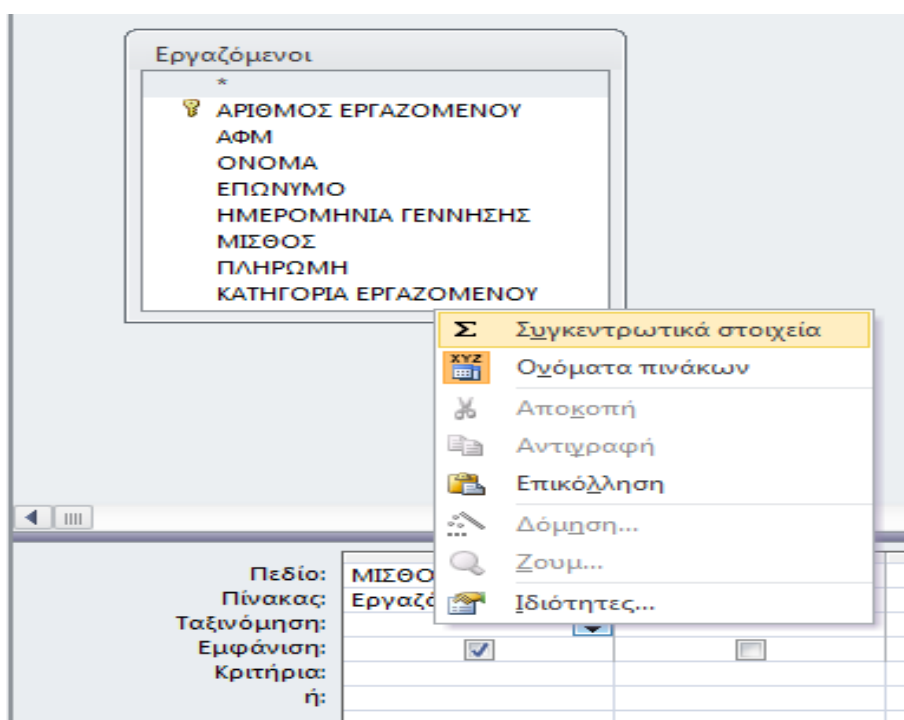


**Παράδειγμα 8<sup>ο</sup>**

**Παράδειγμα 9<sup>ο</sup>** : Εμφάνιση του μέσου όρου των μισθών των εργαζομένων.

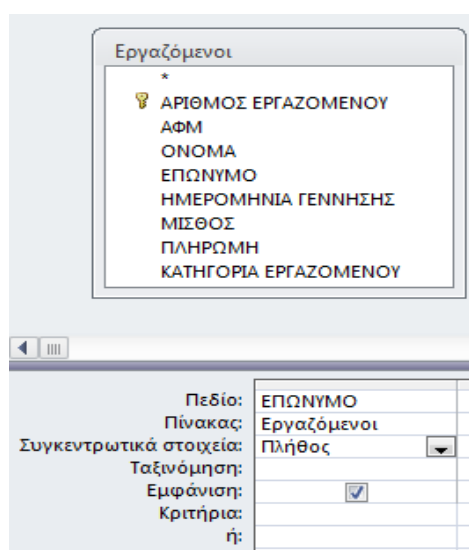
Για να γίνει αυτό πρέπει να επιλέξουμε τον πίνακα που επιθυμούμε, το πεδίο ΜΙΣΘΟΣ, να πατήσουμε δεξί κλικ στο πεδίο Ταξινόμηση και να επιλέγουμε Συγκεντρωτικά στοιχεία.

Με αυτόν τον τρόπο μπορείτε να υπολογίσετε γρήγορα τιμές, όπως μέσους όρους, να βρείτε την ελάχιστη ή τη μέγιστη τιμή και το πλήθος σε μια στήλη δεδομένων.



**Παράδειγμα 9<sup>ο</sup>**

**Παράδειγμα 10<sup>ο</sup>** : Εμφάνιση του πλήθους των εργαζομένων.



## Φόρμα

### Δημιουργία πίνακα σε προβολή σχεδίασης

Μια φόρμα είναι ένα αντικείμενο βάσης δεδομένων το οποίο μπορείτε να χρησιμοποιήσετε για να δημιουργήσετε ένα περιβάλλον χρήστη για μια εφαρμογή βάσης δεδομένων. Μια "δεσμευμένη" φόρμα συνδέεται απευθείας με μια προέλευση δεδομένων όπως πίνακα ή ερώτημα, και μπορεί να χρησιμοποιηθεί για εισαγωγή, επεξεργασία ή εμφάνιση δεδομένων από αυτήν την προέλευση δεδομένων. Εναλλακτικά, μπορείτε να δημιουργήσετε μια "μη δεσμευμένη" φόρμα η οποία δεν συνδέεται απευθείας σε μια προέλευση δεδομένων, αλλά η οποία εξακολουθεί να περιέχει κουμπιά εντολών, ετικέτες ή άλλα στοιχεία ελέγχου τα οποία χρειάζεστε για τη λειτουργία της εφαρμογής.

1. Επιλέγετε από την κορδέλα την καρτέλα δημιουργία -> σχεδίαση φόρμας.

Κάθε ένα από τα εικονίδια είναι αντικείμενα που μπορείτε να εισάγετε στην φόρμα.

Βρισκόμαστε στην καρτέλα σχεδίαση από την κορδέλα.

Επιλέγοντας κάποιο από τα αντικείμενα μπορείτε να εισάγετε λογότυπο, τίτλο ή την τρέχουσα ημερομηνία και ώρα στην φόρμα.

Τα περιθώρια τις φόρμας μπορείτε είτε να τα μεγαλώσετε είτε να τα μικρύνετε.

The image shows the Microsoft Access interface in design view. The ribbon is set to 'Design' (Σχεδίαση). A text box on the left lists various controls like 'Logotipo', 'Titlos', 'Hmeromhnia kai ora', and 'Listes pidiwn'. A text box on the right explains that these controls can be added to the form. A text box at the bottom left notes that the form's borders can be resized. The main area shows a grid representing the form design.

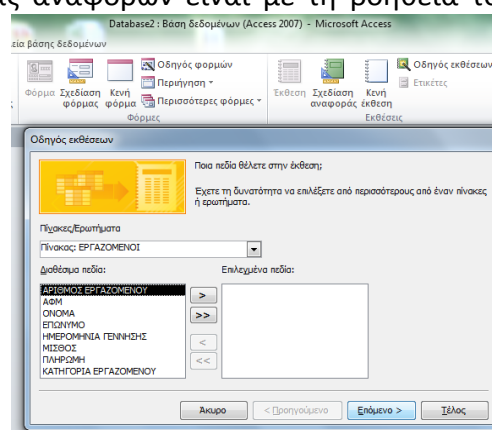
## Έκθεση

Συνήθως χρησιμοποιείτε εκθέσεις για την προβολή, μορφοποίηση και σύνοψη δεδομένων όταν χρησιμοποιείτε μια βάση δεδομένων.

### Δημιουργία έκθεσης με οδηγό σχεδίασης

Οι αναφορές ή εκθέσεις (reports) χρησιμοποιούνται για την εξαγωγή δεδομένων από τη βάση κυρίως σε εκτυπώσεις στο χαρτί. Τα δεδομένα τους μπορούν να είναι απλά στοιχεία από πεδία, υπολογισμοί μεταξύ διαφόρων πεδίων καθώς και μερικά ή γενικά σύνολα. Επίσης τα δεδομένα μπορούν να εμφανίζονται ταξινομημένα ή ομαδοποιημένα. Ο τρόπος σχεδίασης μιας έκθεσης μοιάζει πολύ με αυτόν της φόρμας. Ο ευκολότερος τρόπος δημιουργίας αναφορών είναι με τη βοήθεια του οδηγού.

1. Από την καρτέλα δημιουργία επιλέγετε την ομάδα εκθέσεις.
2. Στη συνέχεια τον οδηγό των εκθέσεων.
3. Έπειτα στο παράθυρο του οδηγού επιλέγετε και συμπληρώνετε τα ζητούμενα πεδία.



Στις Εκθέσεις οι προβολές που χρησιμοποιείτε είναι οι γνωστές η προβολή έκθεσης και προβολή σχεδίασης.

## Ενότητα2 - Διαφάνειες του θεωρητικού μέρους του μαθήματος «Βάσεις Δεδομένων»

### Παρουσίαση 1 - Συστήματα Βάσεων Δεδομένων

# Ενότητα 1

---

## Συστήματα Βάσεων Δεδομένων

Δρ. Τιμπίρης Αλκιβιάδης



---

## Περιεχόμενα ενότητας

---

1. Συστήματα βάσεων δεδομένων
2. Συστατικά στοιχεία του συστήματος βάσεων δεδομένων
3. Οντότητες και συσχετίσεις
4. Πλεονεκτήματα της χρήσης βάσεων δεδομένων
5. Μοντέλα συστημάτων διαχείρισης βάσεων δεδομένων
6. Αρχιτεκτονική συστημάτων διαχείρισης βάσεων δεδομένων
7. Σχισιακές βάσεις δεδομένων
8. Σχισιακά μοντέλα δεδομένων-Πίνακες



# Σκοποί ενότητας

---

- Σκοπός της ενότητας αυτής είναι η εισαγωγή των φοιτητών σε θέματα Βάσεων Δεδομένων (ΒΔ) ώστε να μπορούν να αντιλαμβάνονται τις εισαγωγικές έννοιες που αφορούν τις ΒΔ. Αφού περιηγηθούν μέσα από μια ιστορική αναδρομή στην πορεία και στην και την εξέλιξη των ΒΔ να μπορούν να ξεχωρίσουν τα γνωστά συστήματα ΒΔ και με ποια από αυτά θα ασχοληθούν στην πορεία των σπουδών του. Γνωρίζουν τα πλεονεκτήματα και μειονεκτήματα της χρήσης των ΒΔ και την αρχιτεκτονική client/server σύμφωνα με την οποία θα αναπτύξουν τις εφαρμογές τους.



# Συστήματα βάσεων δεδομένων

---

**DataBase System**= Σύστημα τήρησης εγγραφών με υπολογιστή.  
**Βάση δεδομένων**=χώρος για αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων.  
**SQL**=Structured Query Language (Δομημένη Γλώσσα Ερωτημάτων)

Εισαγωγή → insert  
Ανάκληση → retrieval  
Ενημέρωση → update  
Διαγραφή → delete





# Data - Information

**Δεδομένα** = τιμές

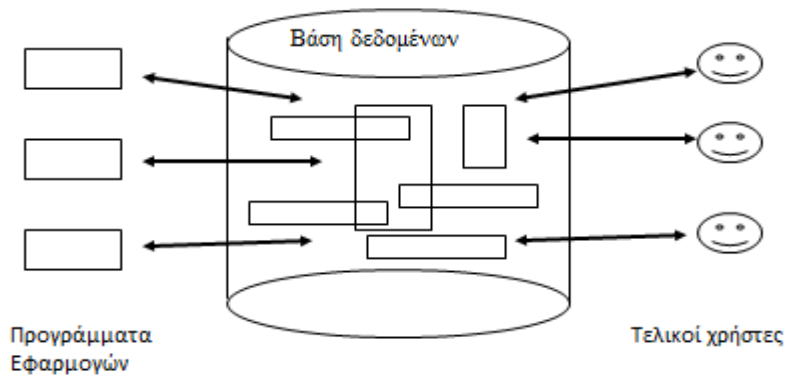
**Πληροφορίες** = σημασία τιμών

Γνωρίζετε κάποιο σύστημα ΒΔ?

Χρησιμοποιήσατε ποτέ κάποιο σύστημα ΒΔ?

Χειρόγραφες ή ηλεκτρονικές ΒΔ?

## Σύστημα βάσης δεδομένων



## Συστατικά στοιχεία του συστήματος βάσης δεδομένων

---

### ΤΑ ΔΕΔΟΜΕΝΑ (DATA)

- Σύστημα ενός χρήστη
- Σύστημα πολλών χρηστών
- Ενοποιημένα integrated
- Μεριζόμενα shared, κοινόχρηστα

### ΤΟ ΥΛΙΚΟ (HARDWARE)

- Μέσα αποθήκευσης, backup

### ΤΟ ΛΟΓΙΣΜΙΚΟ (SOFTWARE)

- Μεταξύ δεδομένων και χρηστών υπάρχει το DBMS (Database management system ή σύστημα διαχείρισης βάσεων δεδομένων)

### ΟΙ ΧΡΗΣΤΕΣ (USERS)

- προγραμματιστές εφαρμογών
- Τελικοί χρήστες
- Υπεύθυνος διαχείρισης βάσεων δεδομένων (database administrator DBA)



## Πλεονεκτήματα της χρήσης βάσεων δεδομένων

---

- Μείωση του πλεονασμού (επανάληψη) των αποθηκευμένων δεδομένων (redundancy)
- Αποφυγή ασυνεπειών
- Κοινόχρηστα δεδομένα
- Επιβολή προτύπων για την αναπαράσταση δεδομένων, που διευκολύνει την ανταλλαγή δεδομένων
- Εφαρμογή περιορισμών ασφαλείας
- Διατήρηση της ακεραιότητας

- Οικονομία χώρου
- Ταχύτητα
- Λιγότερος κόπος
- Άμεση πληροφόρηση



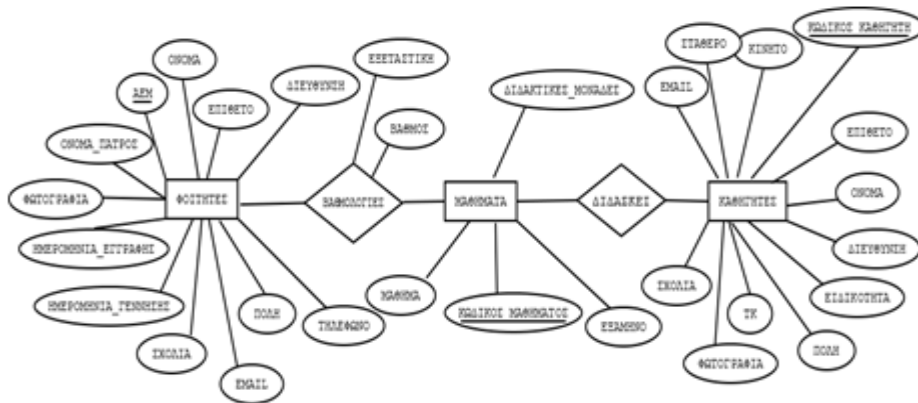
## Μοντέλα συστημάτων διαχείρισης δεδομένων

Συστήματα αντιστραμμένης λίστας	CA-DATACOM/DB της Computer Associates International Inc.
Ιεραρχικά συστήματα	IMS της IBM Corporation
Δικτυωτά συστήματα	CA-IDMS/DB της Computer Associates International Inc.
Σχεσιακά συστήματα (αρχές του 1980) RDBMS	DB2 της IBM Corporation INGRES της Ingres Inc. ORACLE της Oracle Corporation SYBASE της Sybase Inc. SQL Server MySQL

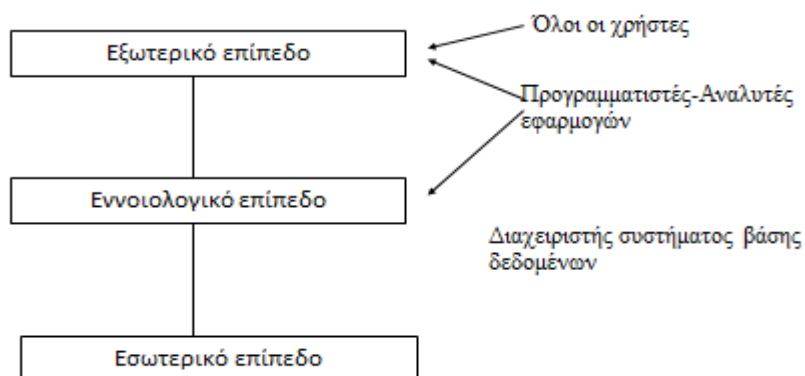
## Οντότητες και συσχετίσεις

ΕΝΝΟΙΑ	ΑΤΥΠΟΣ ΟΡΙΣΜΟΣ	ΠΑΡΑΔΕΙΓΜΑΤΑ
ΟΝΤΟΤΗΤΑ	Ένα διακεκριμένο αντικείμενο	Προμηθευτής, εξάρτημα, υπάλληλος, τμήμα, πρόσωπο
ΙΔΙΟΤΗΤΑ	Μια πληροφορία που περιγράφει μια οντότητα (λέγεται και γνώρισμα)	Κωδικός προμηθευτή, ποσότητα αποστολής, τμήμα υπαλλήλου, τύπος εξαρτήματος, ύψος προσώπου
ΣΥΣΧΕΤΙΣΗ	Μια οντότητα που εξυπηρετεί για να συνδέει δύο ή περισσότερες άλλες οντότητες μεταξύ τους	Αποστολή (προμηθευτής-εξάρτημα) Πρόσληψη (υπάλληλος-τμήμα)

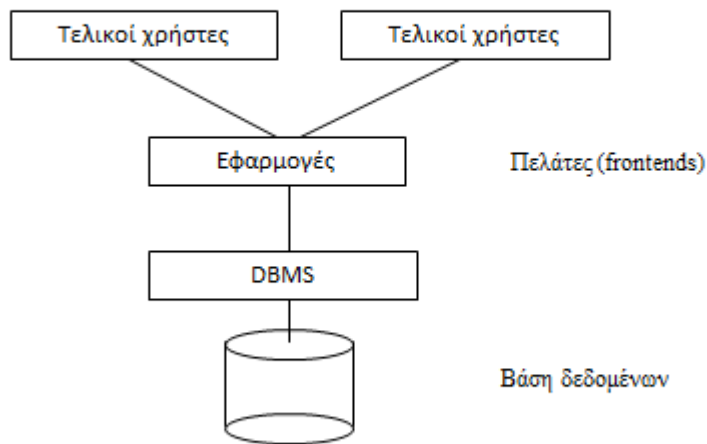
## Διάγραμμα βασικών οντοτήτων και συσχετίσεων



## Αρχιτεκτονική συστημάτων διαχείρισης βάσης δεδομένων



## Αρχιτεκτονική πελάτη/διακομιστή



## Αρχιτεκτονική πελάτη/διακομιστή



## Σχεσιακές βάσεις δεδομένων

---

Σχεσιακό μοντέλο ( Codd '68)

Εγγραφή = συστοιχία

σχέση = πίνακας (γραμμές στήλες)

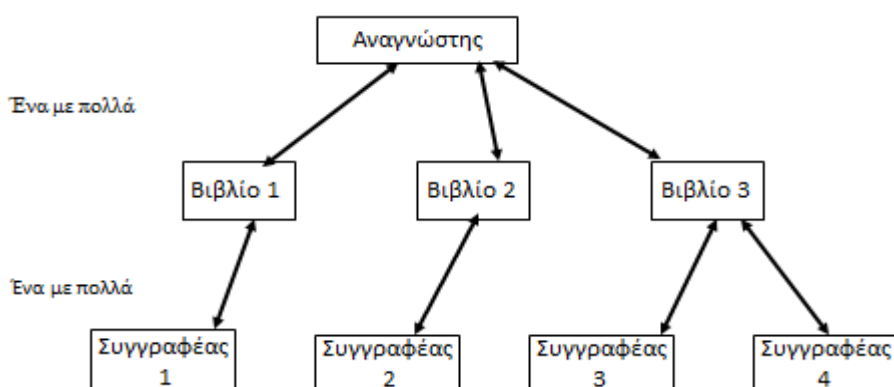
Τιμές βαθμωτές= σε κάθε θέση γραμμής στήλης υπάρχει μια τιμή δεδομένων όχι πολλές τιμές

## Σχεσιακές βάσεις δεδομένων

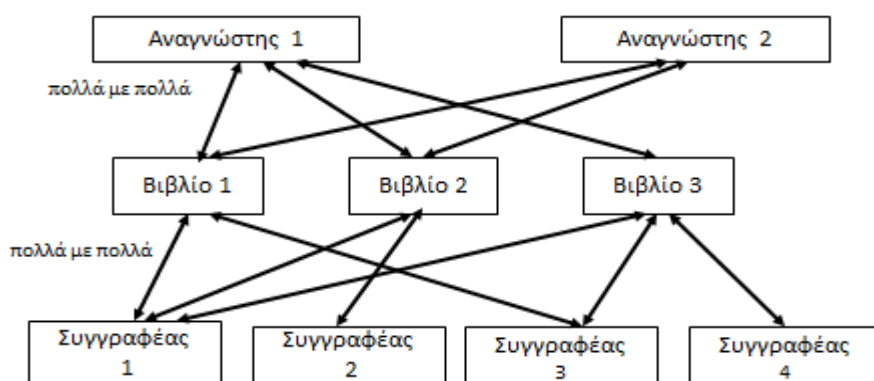
---

- Βασικά αντικείμενα δεδομένων σχεσιακού μοντέλου = πεδία ορισμού & σχέσεις
- Πεδίο ορισμού (domain) = τύπος δεδομένων , παρέχει σύνολο βαθμωτών τιμών από το οποίο διάφορα γνωρίσματα από διάφορες σχέσεις αντλούν συγκεκριμένες τιμές.
- Τα πεδία ορισμού θέτουν δεσμεύσεις στις συγκρίσεις από τις διάφορες σχεσιακές πράξεις (ένωση, τομή, κ.ά)
- Σχέση έχει δύο μέρη: Επικεφαλίδα και κορμό
- Επικεφαλίδα = σύνολο από γνωρίσματα (ζεύγη) . Όνομα γνωρίσματος/ όνομα πεδίου ορισμού.
- Κορμός = σύνολο από συστοιχίες
- Βαθμός = πλήθος γνωρισμάτων
- Πληθικότητα= πλήθος συστοιχιών
- Σχέση = πίνακας όπου στήλες = γνωρίσματα και γραμμές = συστοιχίες

## Ιεραρχικά μοντέλα βάσεων δεδομένων



## Δικτυακά μοντέλα δεδομένων



# Σχεσιακά μοντέλα δεδομένων με πίνακες

ΑΕΜ	ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	ΤΟΠΗ	ΔΙΕΥΘΥΝΣΗ
1	ΜΑΡΙΑ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΑΡΧΗΣ	1/9/2009	1/1/1985	ΣΕΡΡΕΣ	ΣΤΑΜΟΝΙΑ 19
2	ΚΩΣΤΑΣ	ΣΤΑΜΟΣ	ΠΑΝΟΣ	2/9/2009	10/1/1989	ΘΕΣΣΑΛΟΝΙΚΗ (ΟΥΤΙΝΙΑΝΟΥ)	3
3	ΕΛΕΝΗ	ΓΕΩΡΓΙΟΥ	ΑΝΕΣΤΑΣ	1/2/2009	9/2/1989	ΑΘΗΝΑ	ΠΑΙΣΤΟΥ 26

ΑΕΜ	ΕΠΙΘΕΤΟ	ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ	ΒΑΘΜΟΣ	
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2010	ΜΑΘΗΜΑΤΙΚΑ	3
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2011	ΑΥΤΑΥΤΑ	3
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2011	ΒΑΛΕΣ_ΔΕΔΟΜΕΝΩΝ	4
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2009	ΣΤΑΤΙΣΤΙΚΑ	5
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2011	ΦΥΣΙΚΗ	7
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2010	ΜΑΘΗΜΑΤΙΚΑ	3
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2011	ΑΥΤΑΥΤΑ	3
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2011	ΒΑΛΕΣ_ΔΕΔΟΜΕΝΩΝ	4
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2009	ΣΤΑΤΙΣΤΙΚΗ	5
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2010	ΦΥΣΙΚΗ	6
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2010	ΜΑΘΗΜΑΤΙΚΑ	7
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2010	ΑΥΤΑΥΤΑ	4
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2010	ΒΑΛΕΣ_ΔΕΔΟΜΕΝΩΝ	3
1	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΚΩΔ12009-2011	ΣΤΑΤΙΣΤΙΚΗ	3

```

    erDiagram
        ΦΟΙΤΗΣ ||--}| ΕΓΓΡΑΦΕΣ : "1"
        ΕΓΓΡΑΦΕΣ ||--}| ΜΑΘΗΜΑΤΑ : "1"
        ΜΑΘΗΜΑΤΑ ||--}| ΕΛΕΓΧΟΣ : "1"
    
```





## Ενότητα 2

---

### Σχεδιασμός βάσης δεδομένων, E-R

Δρ. Τσιμπήρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

1. Σχεδιασμός βάσης δεδομένων
2. Βήματα σχεδιασμού βάσης δεδομένων
3. Μοντέλο οντοτήτων και συσχετίσεων
4. Οντότητες
5. Τύποι γνωρισμάτων
6. Η τιμή null
7. Η έννοια του κλειδιού
8. Πεδίο τιμών
9. Συσχετίσεις
10. Βαθμός τύπου συσχέτισης
11. Λόγος πληθικότητας
12. Γνωρίσματα τύπων συσχετίσεων
13. Ολική συμμετοχή
14. Ασθενείς τύποι οντοτήτων
15. Περιορισμοί
16. Τύποι με βαθμό μεγαλύτερο του δύο
17. Επεκτάσεις
18. Κριτήρια σχεδιασμού



# Σκοποί ενότητας

---

Ο σκοπός της ενότητας αυτής είναι ο σωστός Σχεδιασμός μιας Βάσης δεδομένων μέσα από συγκεκριμένα βήματα σχεδιασμού. Η χρήση μοντέλων σχεδίασης όπως το Μοντέλο οντοτήτων και συσχετίσεων (E-R) οδηγεί στο σωστό σχεδιασμό. Αυτά που πρέπει να γνωρίζει κάποιος για να χρησιμοποιήσει το μοντέλο E-R είναι: οι οντότητες και οι συσχετίσεις καθώς και οι τύποι γνωρισμάτων, η τιμή null, τα κλειδιά που πρέπει να χρησιμοποιηθούν και τα πεδία τιμών, ο βαθμός και λόγος πληθικότητας οι συμμετοχές (ολική και μερική) καθώς και κάποια κριτήρια και περιορισμοί.

## Σχεδιασμός βάσης δεδομένων

---

- Ανάλυση της πληροφορίας και της σχέσης ανάμεσα στα στοιχεία της βάσης.
- Περιγραφή της δομής με σχήματα σε διάφορους συμβολισμούς ή μοντέλα
- Μοντέλο Οντοτήτων - Συσχετίσεων
- Γραφικό μοντέλο (εννοιολογικό)
- Μετατροπή σε σχεσιακό -> είσοδο σε ένα ΣΔΒΔ
- **Ιδέες → Ο/Σ → Σχέσεις → Σχεσιακό ΣΔΒΔ**

## Βήματα σχεδιασμού βάσης δεδομένων

---

- Ανάλυση απαιτήσεων
- Εννοιολογικός Σχεδιασμός (υψηλού επιπέδου περιγραφή των δεδομένων και των περιορισμών)
- Λογικός Σχεδιασμός (μετατροπή του εννοιολογικού σχεδιασμού σε ένα σχήμα στο μοντέλο που υποστηρίζει το ΣΔΒΔ που επιλέξαμε)
- Φυσικός Σχεδιασμός (επιλογή ΣΔΒΔ και μετατροπή του λογικού σχεδιασμού σε πραγματικούς πίνακες του ΣΔΒΔ)

## Μοντέλο οντοτήτων συσχετίσεων

---



# Οντότητες

## Οντότητα

- Είναι ένα αντικείμενο με φυσική ύπαρξη
- Κάθε οντότητα έχει συγκεκριμένες ιδιότητες - γνωρίσματα
- Μια συγκεκριμένη οντότητα θα έχει μια τιμή για καθένα από τα γνωρίσματα

## Τύπος οντοτήτων (πολλές οντότητες ίδιες με ίδια γνωρίσματα)






- Ορίζει ένα σύνολο από οντότητες που έχουν τα ίδια γνωρίσματα
- Περιγράφεται από ένα όνομα και μια λίστα γνωρισμάτων



# Τύποι γνωρισμάτων

Γνωρίσματα: είναι η ιδιότητα της οντότητας

## Τύποι Γνωρισμάτων:

- Απλά ή ατομικά 
- Σύνθετα 
- Μονότιμα 
- Πλειότιμα: σύνολο από τιμές (κάτω-πάνω όριο) 
- Παραγομμένα: μπορεί να υπολογιστεί από σχετιζόμενες οντότητες ή γνωρίσματα 
- Αποθηκευμένα



# Η τιμή Null

---

**Τιμή null:** Όταν μια οντότητα δεν έχει τιμή για ένα γνώρισμα.

Υπάρχουν δύο περιπτώσεις που η τιμή είναι null

- Όταν δεν υπάρχει δυνατή τιμή (not applicable)
- Όταν υπάρχει δυνατή τιμή αλλά δεν είναι γνωστή
  - ξέρουμε ότι υπάρχει (missing)
  - δεν ξέρουμε αν υπάρχει (not known)

**ΠΡΟΣΟΧΗ:** Όταν μία βάση έχει πολλά Null τότε δεν είναι καλά σχεδιασμένη



# Η έννοια του κλειδιού

---

**Η έννοια του κλειδιού:** Οι τιμές κάποιου γνωρίσματος (ή γνωρισμάτων) προσδιορίζουν μία οντότητα μοναδικά (δηλαδή, δεν μπορεί να υπάρχουν δυο οντότητες με τις ίδιες τιμές στα γνωρίσματα κλειδιά).

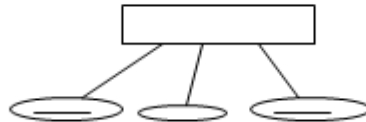
- **Υπερκλειδί:** σύνολο από ένα ή περισσότερα γνωρίσματα που προσδιορίζουν μοναδικά μια οντότητα (superkey)
- **Υποψήφιο κλειδί:** ελάχιστο (μικρότερο αριθμό γνωρισμάτων) υπερκλειδί (candidate key)
- **Πρωτεύον κλειδί:** το υποψήφιο κλειδί που επιλέγουμε (primary key)

**ΠΡΟΣΟΧΗ:** το κλειδί είναι σύνολο γνωρισμάτων



# Η έννοια του κλειδιού

Ισχύει: υπερκλειδί  $\supseteq$  κάθε υποψήφιο κλειδί



*Προσοχή: ο περιορισμός κλειδιού είναι μέρος του σχήματος*

## Παράδειγμα για την κατανόηση της έννοιας του κλειδιού



**Υπερκλειδί:** ΑΤ, ΑΕΜ, ΑΜΚΑ, ΑΦΜ (ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΗΜ\_ΓΕΝΝΗΣΗΣ, ΟΝ\_ΠΑΤΡΟΣ, ΟΝ\_ΜΗΤΡΟΣ)

**Υποψήφιο κλειδί:** ΑΤ, ΑΕΜ, ΑΜΚΑ, ΑΦΜ (μικρότερο αριθμό γνωρισμάτων)

**Πρωτεύον κλειδί:** ΑΕΜ

# Πεδίο τιμών

**Πεδίο τιμών** : σύνολο τιμών ή πεδίο ορισμού των γνωρισμάτων

Ένα απλό γνώρισμα  $A$  με σύνολο τιμών  $V$  ενός τύπου οντοτήτων  $E$  μπορεί να οριστεί ως μια συνάρτηση από το  $E$  στο δυναμοσύνολο ( $P$ ) του  $V$

$$A : E \rightarrow P(V)$$

μονότιμα - σύνολο από ένα στοιχείο

σύνθετα - καρτεσιανό γινόμενο  $P(V_1) \times P(V_2) \times \dots \times P(V_n)$

**Συμβολισμός**

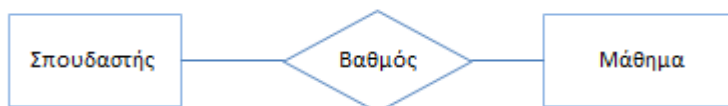
- Σύνθετα:  $()$
- Πλειότιμα:  $\{\}$



# Συσχετίσεις

**Τύπος συσχέτισης R**: ορίζεται μια σύνδεση (σχέση) μεταξύ  $n$  τύπων οντοτήτων

**Τύπος - Στιγμιότυπο**



Συχνά αναπαράσταση του στιγμιότυπου ως ένα πίνακα (σχέση) όπου κάθε γραμμή αντιστοιχεί στα ζεύγη των οντοτήτων που συμμετέχουν στη συσχέτιση

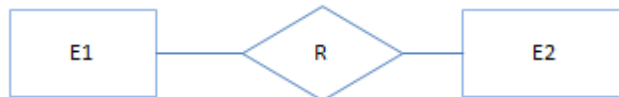


# Συσχετίσεις

---

Μαθηματικά το  $R$  είναι ένα σύνολο από στιγμιότυπα συσχετίσεων  $r_i$  όπου κάθε  $r_i$  συνδέει η οντότητες

$R$  υποσύνολο καρτεσιανού γινομένου



$$R \subseteq E1 \times E2$$

# Αναδρομικές Συσχετίσεις

---

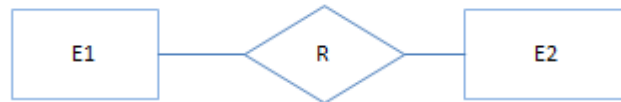
Μια οντότητα συσχετίζεται με τον εαυτό της





## Βαθμός τύπου συσχέτισης

Βαθμός ενός τύπου συσχέτισης (degree): το πλήθος των τύπων οντοτήτων που συμμετέχουν.



Βαθμός=2

## Λόγος πληθικότητας

Για δυαδικές συσχετίσεις :

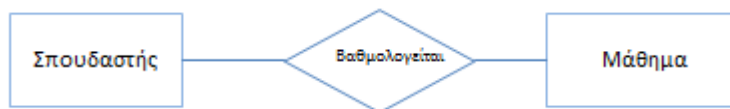
- ένα-προς-ένα 1:1
- ένα-προς-πολλά 1:N
- πολλά-προς-ένα N:1
- πολλά-προς-πολλά N:M

Για ένα τύπο συσχετίσεων ισχύει:

σε πόσες συσχετίσεις (στιγμιότυπα συσχετίσεων) μια οντότητα μπορεί να συμμετέχει?

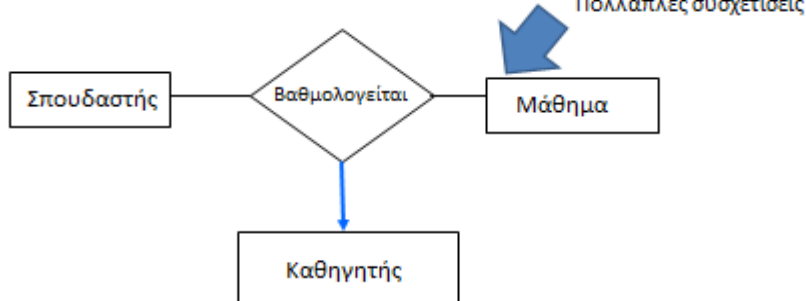
Δυαδικές συσχετίσεις

Παράδειγμα - Συμβολισμός



# Λόγος πληθικότητας

Παράδειγμα - Συμβολισμός



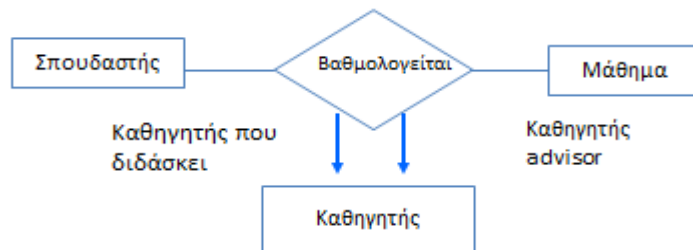
Αν το βέλος δείχνει στο E, αυτό σημαίνει ότι αν επιλέξουμε μια οντότητα από καθένα από τα άλλα σύνολα οντοτήτων, αυτές συσχετίζονται με μια μοναδική οντότητα του E. Περιορισμός; (συναρτησιακές εξαρτήσεις!)



# Λόγος πληθικότητας

Αναδρομικές συσχετίσεις: Όταν ο τύπος οντοτήτων συμμετέχει παραπάνω από μία φορά

Ένας τύπος που συμμετέχει σε μια σχέση παίζει ένα συγκεκριμένο ρόλο



(Καθηγητής που διδάσκει, Καθηγητής advisor, Σπουδαστής, Μάθημα)



## Γνωρίσματα τύπων συσχετίσεων

---

Οι τύποι συσχετίσεων μπορεί να έχουν και **γνωρίσματα**.

Πότε είναι αυτό είναι καλή επιλογή αντί της δημιουργίας νέου τύπου οντοτήτων;

Εάν είναι 1:N ή N:1 μπορώ να βάλω το γνώρισμα σε άλλη οντότητα

Εάν είναι N:M δεν μπορείτε να το βάλετε σε άλλη οντότητα.

## Ολική συμμετοχή

---

Η συμμετοχή ενός συνόλου οντοτήτων E σε ένα σύνολο συσχετίσεων R είναι **ολική** αν κάθε οντότητα του E συμμετέχει τουλάχιστον σε μια συσχέτιση στο R.



Αν κάποιες οντότητες του E δεν συμμετέχουν στο R τότε είναι **μερική**.

## Ασθενείς τύποι οντοτήτων

Μη ισχυροί ή ασθενείς τύποι οντοτήτων: Όταν μια οντότητα δεν έχει αρκετά γνωρίσματα για να σχηματίσει πρωτεύον κλειδί.

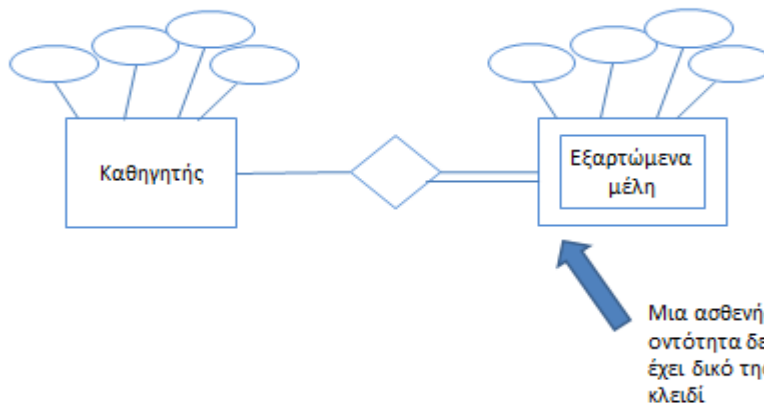
Μια ασθενής οντότητα E πρέπει να συμμετέχει με *ολική συμμετοχή* σε μια *ένα-προς-πολλά* συσχέτιση R με έναν τύπο οντοτήτων F.

R: προσδιορίζουσα συσχέτιση, F: προσδιορίζοντα ιδιοκτήτη

Προσδιορίζεται μοναδικά από:

μερικό κλειδί (γνωρίσματα της E) + κλειδί της F

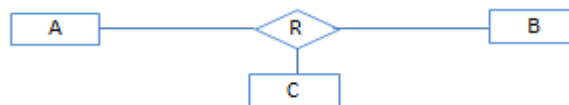
## Ασθενείς τύποι οντοτήτων



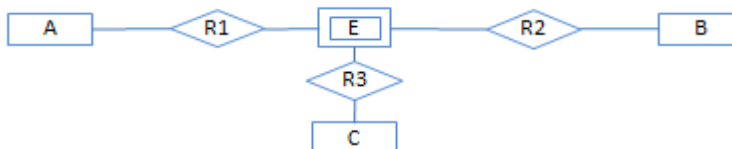
# Περιορισμοί

- Κλειδιού
- Μοναδικής Τιμής (Πληθικότητα, Μονότιμα γνωρίσματα)
- Συμμετοχής
- Εξάρτησης (Ασθενής Οντότητας)
- Πληθικότητα

## Τύποι με βαθμό μεγαλύτερο του δύο



Πρέπει να μετατραπεί σε δυαδική:

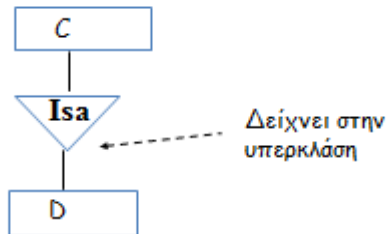


Βαθμός μεγαλύτερος του δύο :

- πρόβλημα στην αποθήκευση
- Πολυπλοκότητα
- περιορισμοί συμμετοχής

# Επεκτάσεις

Επεκτάσεις του Μοντέλου



Το **Isa** βοηθάει στο να επεκταθεί το πρόβλημα.

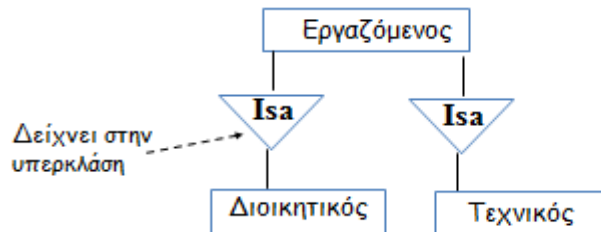


Τμήμα Μηχανικών Πληροφορικής ΤΕ

27

# Επεκτάσεις

Παράδειγμα



Τμήμα Μηχανικών Πληροφορικής ΤΕ

28

## Κριτήρια σχεδιασμού

---

Τα κριτήρια πλεονασμού πρέπει:

- να ακολουθούν πιστά τους περιορισμούς (specifications)
- να αποφεύγουν τον πλεονασμό (χώρος, συνέπεια)
- να έχουν απλότητα

## Κριτήρια σχεδιασμού

---

Επιλογή των κατάλληλων στοιχείων σχεδιασμού:

- Γνώρισμα ή Τύπο Οντοτήτων?
- Πολλές δυαδικές συσχετίσεις ή μία συσχέτιση μεγαλύτερου βαθμού?
- Οντότητα ή Συσχέτιση?
- Γνωρίσματα συσχετίσεων (πότε μπορεί να μεταφερθούν στις συμμετέχουσες οντότητες)?
- Χρήση ασθενούς οντότητας?

## Ενότητα 3

---

### Σχεσιακό Μοντέλο

Δρ. Τιμπίρης Αλκιβιάδης



---

## Περιεχόμενα ενότητας

---

- Λογικός Σχεδιασμός (Σχεσιακό Μοντέλο)
- Εισαγωγικά
- Τυπικός ορισμός Σχεσιακού Μοντέλου
- Κριτήρια Σχεδιασμού





# Σκοποί ενότητας

Ο σκοπός της άσκησης αυτής είναι η παρουσίαση και η κατανόηση του σχεσιακού μοντέλου, το μοντέλο που έχει επικρατήσει για το σχεδιασμό των βάσεων δεδομένων. Παρουσιάζονται τα δομικά στοιχεία, οι δομικοί περιορισμοί του μοντέλου αυτού

## Διαδικασία Ανάπτυξης ΒΔ



## Δομές - Πράξεις

---

- ΔΟΜΕΣ: **RELATIONS** (ΣΧΕΣΕΙΣ) – μία μόνο Δομή
- ΠΡΑΞΕΙΣ: **Επεξεργασία των Σχέσεων**
- Αντιστοιχίες με το E-R Μοντέλο
  - Τύπος Οντοτήτων <-> Τύπος Σχέσης (Relation)
  - Οντότητα <-> n-πλειάδα (Tuple)
  - Γνώρισμα <-> Γνώρισμα (Attribute)
  - Συσχέτιση <-> Τύπος Σχέσης (Relation)

## Πίνακας=Σχέση?

---

Ένα Σχεσιακό Σχήμα είναι ένα Σύνολο Σχέσεων ή Πινάκων

Τα ονόματα των στηλών είναι τα ονόματα των Γνωρισμάτων

AAA	345	78	ΦΦΦΦΦ
123	ΚΚΚΚΚ	ΒΒΒΒ	ΤΤΤΤΤ
1-12-1990	1234	YES	2-3-2000

Είναι αυτός ο πίνακας σχέση?

## Στιγμιότυπο ΒΔ

The image displays four screenshots of a database management system (DBMS) interface, likely Oracle SQL Developer, showing different tables and their data. The top-left screenshot shows a table with columns for course ID, course name, credits, and instructor ID. The top-right screenshot shows a table with columns for course ID, course name, credits, and instructor ID, with a list of rows. The bottom-left screenshot shows a table with columns for course ID, course name, credits, and instructor ID, with a list of rows. The bottom-right screenshot shows a table with columns for course ID, course name, credits, and instructor ID, with a list of rows.

## Βασικές Πράξεις

### 3 ΒΑΣΙΚΕΣ ΠΡΑΞΕΙΣ

- ΕΠΙΛΟΓΗ (*SELECT*)
- ΠΡΟΒΟΛΗ (*PROJECT*)
- ΚΑΡΤΕΣΙΑΝΟ ΓΙΝΟΜΕΝΟ (*PRODUCT*)

### ΜΗ ΒΑΣΙΚΗ ΑΛΛΑ ΧΡΗΣΙΜΗ ΠΡΑΞΗ

- ΣΥΝΔΕΣΗ (*JOIN*). Είναι συνδυασμός καρτεσιανού γινομένου και επιλογής

## Ενημερώσεις/τροποποιήσεις

---

INSERT: *Εισαγωγή γραμμών*

DELETE: *Διαγραφή γραμμών*

UPDATE: *Τροποποίηση*

Η συντριπτική πλειοψηφία των DBMS είναι Σχεσιακά και διατίθενται σε ΟΛΕΣ τις Υπολογιστικές Πλατφόρμες.

E.F. Codd ("A relational model for large shared data banks", CACM), σαν Θεωρία για Μοντέλα Δεδομένων

## Ορισμός

---

- Σχεσιακή Βάση Δεδομένων είναι ένα σύνολο από ΣΧΕΣΕΙΣ
- **ΣΧΕΣΗ (RELATION)**: Ένας Πίνακας τιμών.
  - Κάθε **στήλη** στον πίνακα έχει όνομα, ονομάζεται **γνώρισμα, πεδίο (attribute –field-column)**.
  - Κάθε **γραμμή** ονομάζεται **πλειάδα, εγγραφή (Tuple, record)** και παριστά τα χαρακτηριστικά μιας οντότητας στο μοντέλο.

## Τυπικός Ορισμός

---

- **ΔΟΜΕΣ**
  - Μόνο μία δομή: *relations (σχέσεις)* (ΜΕ ΜΟΝΑΔΙΚΟ ΟΝΟΜΑ)
- Ένα **Σχεσιακό Σχήμα**  $R$  είναι το όνομα και τα γνωρίσματα σε μια Σχέση, μαζί με τα αντίστοιχα πεδία τιμών για τα γνωρίσματα. Όταν είναι προφανές, τα πεδία τιμών παραλείπονται.  
Συμβολισμός:  $R(A_1, A_2, \dots, A_n)$   
π.χ., ΦΟΙΤΗΤΗΣ (ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ)
- Ένα **Σχήμα Βάσης Δεδομένων**  $S$  είναι ένα σύνολο Σχέσεων.  
Συμβολισμός:  $S = \{R_1, R_2, \dots, R_m\}$   
π.χ., ΤΕΙ = { ΦΟΙΤΗΤΕΣ, ΚΑΘΗΓΗΤΕΣ, ... }

## Τυπικός Ορισμός

---

Μια **πλειάδα (Tuple)**  $t$  μιας Σχέσης  $R(A_1, A_2, \dots, A_n)$  είναι μια (διατεταγμένη) λίστα τιμών  $t = \langle v_1, v_2, \dots, v_n \rangle$ , όπου κάθε τιμή  $v_i$  είναι ένα στοιχείο του πεδίου  $D(A_i)$ .

Ένα **στιγμιότυπο σχέσης (relation instance)**  $r(R)$ , πιο απλά, **σχέση (relation)**, είναι ένα σύνολο πλειάδων

$$r(R) = \{ t_1, t_2, \dots, t_k \}$$

Η **πληθικότητα (cardinality)** της  $R$  είναι ο αριθμός των πλειάδων στην  $r(R)$ , και συμβολίζεται με  $CARD_R$

## Χαρακτηριστικά των Σχέσεων

---

- Η ΔΙΑΤΑΞΗ των στηλών σε μια σχέση δεν είναι σημαντική
- Η ΔΙΑΤΑΞΗ των γραμμών σε μια σχέση δεν είναι σημαντική
- Κάθε εγγραφή αποθηκεύεται ΜΙΑ ΦΟΡΑ σε μια σχέση
- Μια τιμή μπορεί να παρουσιάζεται ΠΟΛΛΕΣ ΦΟΡΕΣ σε μια στήλη και είναι ΑΤΟΜΙΚΗ
- Τα ονόματα των στηλών είναι μοναδικά
- Σε κάθε στήλη οι τιμές είναι του ίδιου τύπου
  
- Μια ειδική τιμή, ονομαζόμενη ΚΕΝΟ (NULL), χρησιμοποιείται για την να παραστήσει στη βάση μιας τιμή που είναι *μη εφαρμόσιμος* (non-applicable) ή *άγνωστη* (unknown).  
Π.χ, τηλέφωνο, Διεύθυνση δεν τα γνωρίζουμε
- Συμβολισμός: *τιμή γνωρίσματος για μια πλειάδα*  $t$ ,  $t[A_i] = v_i$

## Δομικοί Περιορισμοί

---

- Υπάρχουν **τρία** είδη περιορισμών *που είναι έμφυτοι στο μοντέλο*:
  - ΚΛΕΙΔΙ (KEY),
  - ΑΚΕΡΑΙΟΤΗΤΑ ΟΝΤΟΤΗΤΑΣ (ENTITY INTEGRITY), και
  - ΑΝΑΦΟΡΙΚΗ ΑΚΕΡΑΙΟΤΗΤΑ (REFERENTIAL INTEGRITY.)
- Υπάρχουν **τρία** είδη *ρητών περιορισμών*:
  - ΠΕΔΙΟ ΤΙΜΩΝ (DOMAIN),
  - ΣΤΗΛΩΝ (COLUMN) και
  - ΟΡΙΖΟΜΕΝΩΝ από τον ΧΡΗΣΤΗ (USER-DEFINED)

## Έμφυτοι Δομικοί Περιορισμοί

---

- **Περιορισμοί Κλειδιών:** Τα διαφορετικά κλειδιά, όπως ορίστηκαν στο μοντέλο E-R, ισχύουν και στο Σχεσιακό Μοντέλο.  
Ένα σύνολο γνωρισμάτων **A** Βαθμιασμού μιας σχέσης **R** για το οποίο κάθε πλειάδα σε στιγμιότυπο  $r(R)$  πρέπει να έχει *μοναδική τιμή* (είναι ένα
  - **υπέρ-κλειδί (superkey.)** Δηλαδή, για διαφορετικά  $t_1$  και  $t_2$ , ισχύει  $t_1[AB] \neq t_2[AB]$
  - Ένα **υποψήφιο κλειδί (candidate key)** **K** είναι ένα *ελάχιστο* υπέρ-κλειδί (δηλαδή, δεν υπάρχει υποσύνολο του **K** που να είναι και αυτό υπέρ-κλειδί). Το **K** ονομάζεται συνήθως **ΚΛΕΙΔΙ (key)**.
  - Ένα **πρωτεύον κλειδί (primary key)** **PK** είναι ένα από τα υποψήφια κλειδιά που συμφωνείται να παίξει το ρόλο του του προσδιοριστή για τις πλειάδες της σχέσης (τα πρωτεύοντα κλειδιά **υπογραμμίζονται**)

## Έμφυτοι Δομικοί Περιορισμοί

---

- **ΑΚΕΡΑΙΟΤΗΤΑ ΟΝΤΟΤΗΤΑΣ (ENTITY INTEGRITY):**  
Το κύριο κλειδί PK στο σχήμα της σχέσης **R** **ΔΕΝ** **ΜΠΟΡΕΙ ΝΑ ΕΧΕΙ** ΚΕΝΕΣ (NULL) τιμές σε πλειάδες μιας σχέσης  $r(R)$ .  
 $t[PK] \neq \text{NULL}$ , για κάθε  $t$  στην  $r(R)$ 
  - Ο λόγος πίσω από αυτό τον περιορισμό είναι ότι το κύριο κλειδί χρησιμοποιείται για τον προσδιορισμό μιας εγγραφής σε μια Σχέση.
  - Σημειώνεται ότι και άλλα γνωρίσματα στην **R** μπορεί να περιορίζονται στο να μην έχουν NULL από ΡΗΤΟΥΣ περιορισμούς.

## Έμφυτοι Δομικοί Περιορισμοί

---

- **ΑΝΑΦΟΡΙΚΗ ΑΚΕΡΑΙΟΤΗΤΑ (REFERENTIAL INTEGRITY):**  
Αυτός ο δομικός περιορισμός εμπλέκει ΔΥΟ σχέσεις και χρησιμοποιείται για να καταγράψει τη συνέπεια σε μια συσχέτιση μεταξύ εγγραφών των δυο σχέσεων.
- Η συνήθης μορφή είναι αυτή των **ξένων κλειδιών**.
  - Ένα **ξένο κλειδί (foreign key) FK** είναι ένα σύνολο γνωρισμάτων σε μια σχέση R1 που αποτελεί **κύριο κλειδί** σε μια άλλη σχέση R2.  
Μια πλειάδα  $t_1$  στην  $r(R_1)$  λέγεται ότι **αναφέρεται** σε μια άλλη πλειάδα  $t_2$  στην  $r(R_2)$ , εάν:  $t_1[FK] = t_2[FK]$

## Ρητοί Περιορισμοί

---

- **Περιορισμοί Πεδίου Τιμών:** Είναι οι κανόνες που ορίζονται για το πεδίο τιμών και κληρονομούνται από τις στήλες (γνωρίσματα) **που παίρνουν τιμές από το πεδίο**.
  - Το πεδίο μπορεί να οριστεί μαζί με κανόνες ακεραιότητας (π.χ., το πεδίο των *integers* με όλους τους κανόνες για ακέραιους). Αυτοί είναι (κυρίως) οι βασικοί **τύποι δεδομένων (data types)**.
- **Περιορισμοί Στηλών:** Είναι **επιπρόσθετοι** των περιορισμών πεδίου τιμών και αναφέρονται στις τιμές για τα γνωρίσματα.
  - Για παράδειγμα., η στήλη των *small integers* ή *integers between 1 and 10*, κλπ. είναι επιπλέον περιορισμοί των ακεραίων



---

## USER-DEFINED

---

- ΠΕΡΙΟΡΙΣΜΟΙ ΟΡΙΖΟΜΕΝΟΙ από τον ΧΡΗΣΤΗ (USER-DEFINED)
  - επίσης* stored procedures, triggers, methods (για object-oriented systems)
- Γενικά, τα DBMS είναι **αδύνατα σε υποστήριξη περιορισμών**

---

## Πράξεις

---

- Διαχωρίζονται σε (α) ΕΝΗΜΕΡΩΣΕΙΣ, (β) ΑΝΑΚΤΗΣΕΙΣ
- Το σύνολο των πράξεων στο Σχεσιακό Μοντέλο είναι **ΚΛΕΙΣΤΟ** δηλαδή οι πράξεις ορίζονται σε Σχέσεις και έχουν αποτέλεσμα νέες Σχέσεις
- **Ενημερώσεις (UPDATE) σε Σχέσεις**
  - Εισαγωγή (INSERT) πλειάδας
  - Διαγραφή (DELETE) πλειάδας
  - Τροποποίηση (MODIFY) πλειάδας
- Οι περιορισμοί ακεραιότητας δεν πρέπει να παραβιάζονται με την εκτέλεση μιας πράξης ενημέρωσης. Για αυτό, ενημερώσεις μπορεί είτε να **απορρίπτονται** ή να **διορθώνουν** με την εκτέλεση **νέων (επιπλέον) ενημερώσεων**.
  - Π.χ., όταν μια πλειάδα του EMPLOYEE διαγράφεται, όλες οι πλειάδες στην WORKING\_ON που έχουν την ίδια τιμή στο SSN διαγράφονται (ανύπαρκτοι υπάλληλοι δεν δουλεύουν σε έργα!)

## Άτυπα Κριτήρια Σχεδιασμού

---

- **Σχεδιασμός Σχέσεων:** Ομαδοποίηση Γνωρισμάτων έτσι ώστε να επιτευχθούν “καλά” σχήματα Σχέσεων (Σχέσεις Βάσης)
- **ΑΤΥΠΑ ΚΡΙΤΗΡΙΑ:**
  - Προσπάθεια για εννοιολογική καθαρότητα
  - Προσπάθεια για αποδοτικότητα χώρου (αποφυγή πλεονασμού)
  - Προσπάθεια για ακεραιότητα (αποφυγή ανωμαλιών ενημέρωσης)
  - Προσπάθεια για πληρότητα (αποφυγή NULL τιμών σε πλειάδες)
  - Προσπάθεια για γλωσσολογική αποδοτικότητα
  - Προσπάθεια για καλές Επιδόσεις (*performance*)

## Σχεδιασμός Σχεσιακού Μοντέλου

---

- Α τρόπος: από εννοιολογικό σχεδιασμό (π.χ. με διάγραμμα E-R) στο λογικό σχεδιασμό (π.χ. Σχεσιακό μοντέλο)
  - Βήματα μετάβασης από το E-R σε πίνακες του σχεσιακού μοντέλου
- Β τρόπος: από το «μηδέν»
  - Ορισμός «καθολικής» (universal) σχέσης που τα περιέχει ΟΛΑ
  - Αποσύνθεση σε περισσότερες σχέσεις με βάση τη **Θεωρία Κανονικοποίησης**

## Ενότητα 4

---

### Μετατροπή σχήματος Ο/Σ σε σχεσιακό

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

1. Μετατροπή σχήματος Ο/Σ σε σχεσιακό
2. Ισχυροί τύποι οντοτήτων
3. Ασθενείς τύποι οντοτήτων
4. Συσχετίσεις 1-1
5. Συσχετίσεις 1-N
6. Συσχετίσεις N-M
7. Γνωρίσματα



# Σκοποί ενότητας

---

Σκοπός της ενότητας αυτής είναι η κατανόηση του τρόπου με τον οποίο γίνεται η μετατροπή ενός μοντέλου οντοτήτων – συσχετίσεων στο σχεσιακό μοντέλο. Η σωστή μετατροπή των οντοτήτων με τα γνωρίσματά τους (απλά, σύνθετα, πλειότιμα κλπ) αλλά και των συσχετίσεων (1:1, 1:N και N:M) εγγυάται το σωστό σχεδιασμό της ΒΔ που θα προκύψει. Υπάρχουν σαφείς κανόνες που αν δεν τηρηθούν θα αλλάξει όλο το νόημα της ΒΔ με επακόλουθα προβλήματα συνέπειας και πλεονασμού.



## Μετατροπή σχήματος Ο/Σ σε σχεσιακό

---

Για κάθε τύπο οντοτήτων και για κάθε τύπο συσχετίσεων N:M δημιουργούμε ένα σχήμα σχέσης που παίρνει το όνομα του αντίστοιχου τύπου.



# Ισχυροί τύποι οντοτήτων

---

## Ισχυροί τύποι οντοτήτων με μονότιμα γνωρίσματα

Για κάθε (ισχυρό) τύπο οντοτήτων  $E$  δημιουργούμε ένα σχήμα σχέσης  $R$  με τα ίδια γνωρίσματα - ένα για κάθε απλό γνώρισμα του  $E$ .

Αν το  $E$  έχει **σύνθετα γνωρίσματα**, στο σχεσιακό σχήμα  $R$  έχουμε ένα γνώρισμα για κάθε απλό γνώρισμα που απαρτίζει το σύνθετο.

# Ασθενείς τύποι οντοτήτων


---

## Ασθενείς τύποι οντοτήτων με (μονότιμα) γνωρίσματα

Για κάθε ασθενή τύπο οντοτήτων  $A$  που εξαρτάται από τον ισχυρό τύπο οντοτήτων  $B$  (προσδιορίζον ιδιοκτήτης) δημιουργούμε ένα σχήμα σχέσης  $R$  με γνωρίσματα:

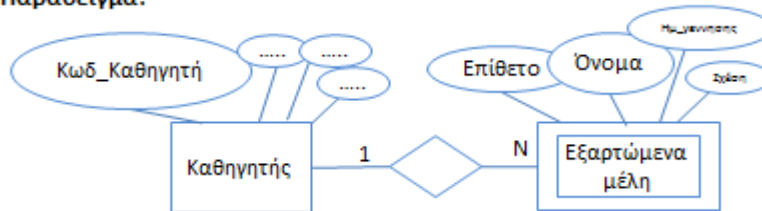
- τα γνωρίσματα του μερικού κλειδιού του  $A$
- τα γνωρίσματα του πρωτεύοντος κλειδιού του  $B$

Ξένο κλειδί



# Ασθενείς τύποι οντοτήτων

Παράδειγμα:



Ο πίνακας εξαρτώμενα μέλη είναι ασθενής οντότητα. Κατά τη μετατροπή στο σχεσιακό μοντέλο δημιουργείται ένας νέος πίνακας που περιέχει τα πεδία της ασθενούς οντότητας και ένα ξένο κλειδί (το πρωτεύον κλειδί του πίνακα καθηγητές). Ο συνδυασμός όλων των πεδίων αποτελεί το πρωτεύον κλειδί του πίνακα.



Εξαρτώμενα μέλη
Επίθετο
Όνομα
Ημ_γεννησης
Σχέση
Κωδ_Καθηγητή

## Συσχετίσεις

Γενικά, για κάθε συσχέτιση  $R$  μεταξύ  $n$  τύπων οντοτήτων που αντιστοιχούν στις σχέσεις  $S_1, S_2, \dots, S_n$  δημιουργούμε μια νέα σχέση  $R$  με γνωρίσματα:

- τα γνωρίσματα (ξένα κλειδιά) του πρωτεύοντος κλειδιού κάθε συμμετέχουσας σχέσης  $S_i$
- τα γνωρίσματα της  $R$  (αν υπάρχουν)

## Συσχετίσεις 1:1

---

### 1:1 δυαδική (μη ασθενής) συσχέτιση

Για κάθε 1-1 δυαδική συσχέτιση  $R$  μεταξύ δύο τύπων οντοτήτων του διαγράμματος  $O/S$  που αντιστοιχούν στις σχέσεις  $T$  και  $S$

1. επιλογή μιας εκ των  $T$  και  $S$ , έστω της  $S$
2. το πρωτεύον κλειδί  $S$  γίνεται ξένο κλειδί της  $T$

## Συσχετίσεις 1-N

---

### 1:N δυαδική (μη ασθενής) συσχέτιση

Για κάθε 1:N δυαδική συσχέτιση  $R$  μεταξύ δύο τύπων οντοτήτων του διαγράμματος  $O/S$  που αντιστοιχούν στις σχέσεις  $T$  και  $S$

1. έστω  $T$  από την **πλευρά 1**
2. το πρωτεύον κλειδί της  $T$  γίνεται ξένο κλειδί της  $S$

# Γνωρίσματα

## Σύνθετα:

Ένα γνώρισμα για κάθε απλό γνώρισμα που απαρτίζει το σύνθετο.

## Πλειότιμα:

Για κάθε πλειότιμο γνώρισμα A, κατασκευάζουμε μια σχέση R με γνωρίσματα:

- το A (ή τα γνωρίσματα του A αν το A είναι σύνθετο) και
- τα γνωρίσματα (ξένο κλειδί) του πρωτεύοντος κλειδιού της σχέσης που παριστάνει τον τύπο οντοτήτων η συσχετίσεων του οποίου γνώρισμα είναι το A

# Μεθοδολογία δημιουργίας ΒΔ

- Εννοιολογικό μοντέλο, E-R
- Λογικό μοντέλο, Σχεσιακό μοντέλο
- Φυσικός Σχεδιασμός (RDBMS)

Σχέση= Πίνακας

Γραμμές

Στήλη 1	Στήλη 2	Στήλη 3
Τιμή	Τιμή	Τιμή
....	....	....



# Σχεσιακό μοντέλο

Για να είναι ένας πίνακας μια σχέση (relation) θα πρέπει να έχει τα εξής κριτήρια:

- Τα ονόματα των στηλών να είναι μοναδικά.
- Σε κάθε στήλη οι τιμές των πεδίων πρέπει να είναι του ίδιου τύπου.
- Οι τιμές του κάθε πεδίου είναι ατομικές.
- Δεν παίζει ρόλο η διάταξη των πεδίων από δεξιά προς τα αριστερά.
- Δεν παίζει ρόλο η διάταξη των γραμμών από πάνω προς τα κάτω.
- Δεν πρέπει να υπάρχουν οι διπλό-εγγραφές (ονομάζεται πλεονασμός).

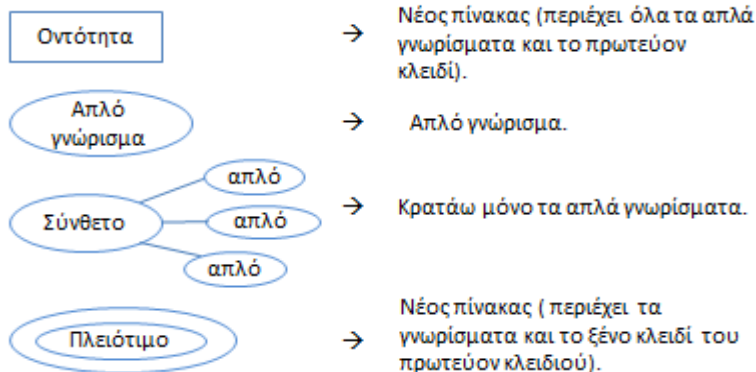
## Σχέση= Πίνακας



Τμήμα Μηχανικών Πληροφορικής ΤΕ

13

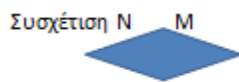
# Μετατροπές



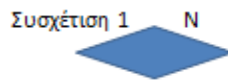
Τμήμα Μηχανικών Πληροφορικής ΤΕ

14

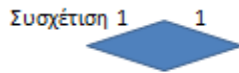
## Μετατροπές συσχετίσεων



→ Νέος πίνακας (περιέχει όλα τα απλά γνωρίσματα της συσχέτισης και δύο ξένα κλειδιά που αναφέρονται στα πρωτεύοντα κλειδιά των συσχετιζόμενων πινάκων .



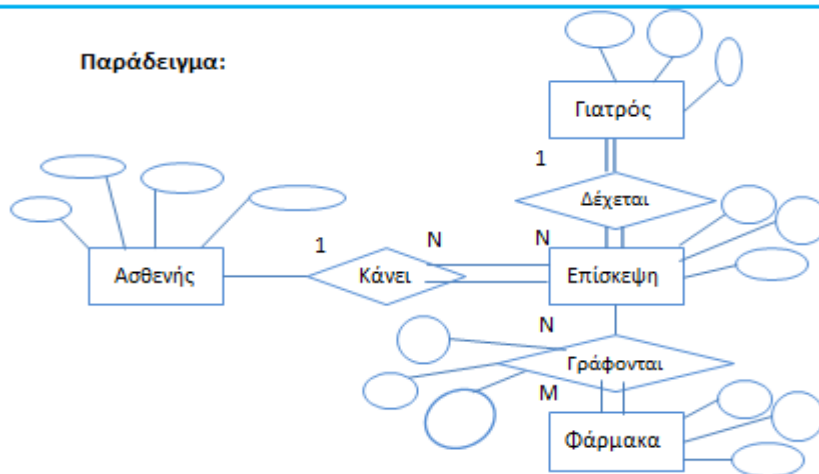
→ Ξένο κλειδί στην πλευρά του N  
Θα είναι ένα απλό γνώρισμα.



→ Ξένο κλειδί στην πλευρά της ολικής συμμετοχής  
Θα είναι ένα απλό γνώρισμα.

## Μετατροπή σχήματος Ο/Σ σε σχεσιακό

Παράδειγμα:



## Ενότητα 5

---

### Σχεσιακή άλγεβρα

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Σύμβολα πράξεων σχεσιακής άλγεβρας
- Προβολή στηλών
- Περιορισμός γραμμών
- Καρτεσιανό γινόμενο
- Εσωτερική, φυσική σύζευξη
- Ένωση
- Τομή
- Διαφορά
- Παραδείγματα



# Σκοποί ενότητας

---

Στην ενότητα αυτή παρουσιάζονται οι πράξεις της Σχεσιακής Άλγεβρας (Προβολή, Επιλογή, Καρτεσιανό Γινόμενο, μετονομασία, Ένωση, Τομή, Διαφορά, Επιπρόσθετοι Τελεστές) πάνω στις οποίες βασίζεται το σχεσιακό μοντέλο.

Η κατανόηση της σχεσιακής άλγεβρας βοηθάει στην καλύτερη αντίληψη εννοιών που θα παρουσιαστούν στη συνέχεια κυρίως στη γλώσσα SQL



## Σύμβολα πράξεων σχεσιακής άλγεβρας

---

Τα σύμβολα πράξεων σχεσιακής άλγεβρας είναι τα εξής:

- $\Pi$  → Προβολή στηλών
- $\sigma$  → Περιορισμός γραμμών
- $\times$  → Καρτεσιανό γινόμενο
- $\bowtie$  → εσωτερική, φυσική σύζευξη
- $\cup$  → ένωση
- $\cap$  → τομή
- $-$  → διαφορά



# Παράδειγμα με τη σχεσιακή άλγεβρα

Πίνακας r

A1	A2	A3	AK
1	AA	A1	AA
2	BB	B1	BB
3	CC	C1	CC

Δείξτε την στήλη A2 που το σ είναι ίσο με CC του πίνακα r.

Απάντηση:  $\Pi_{A2}(\sigma_{A2=CC}(r))$



## Τελεστής τομής

Ορισμός:  $r \cap s = \{t \mid t \in r \text{ και } t \in s\}$



Τελεστής διαφοράς:



# Ένωση-τομή-διαφορά

Έστω ο πίνακας R περιέχει  $N_1$  εγγραφές και ο πίνακας S που περιέχει  $N_2$  εγγραφές με  $N_1 > N_2$ . Ποιο είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος εγγραφών που θα προκύψει από τις πράξεις  $R \cup S$ ,  $R \cap S$ ,  $R - S$ ,  $S - R$ .

	MAX	MIN
$R \cup S$	$N_1 + N_2$	$N_1$
$R \cap S$	$N_2$	0
$R - S$	$N_1$	$N_1 - N_2$
$S - R$	$N_2$	0



# Καρτεσιανό γινόμενο

R → Πίνακας R

A1	A2	A3
1	AA	A1
2	BB	B1
3	CC	C1

S → Πίνακας S

A1	F1	F2
1	KK	KK
1	LL	LL
2	FF	FF
2	KK	KK

$R \times S$

Δημιουργείται ένας νέος πίνακας με 12 γραμμές.

R			S		
A1	A2	A3	A1	F1	F2
1	AA	A1	1	KK	KK
1	AA	A1	1	LL	LL
1	AA	A1	2	FF	FF
1	AA	A1	2	KK	KK
2	BB	B1	1	KK	KK
2	BB	B1	1	LL	LL
2	BB	B1	2	FF	FF
2	BB	B1	2	KK	KK
3	CC	C1	1	KK	KK
3	CC	C1	1	LL	LL
3	CC	C1	2	FF	FF
3	CC	C1	2	KK	KK

Πλειάδες = Γραμμές



## Εσωτερική σύζευξη

Έστω οι πίνακες  $R(A1,A2,A3)$  και  $S(A1,F1,F2)$  πως παρουσιάζεται ο τελεστής εσωτερικής σύζευξης με προβολής και περιορισμού?

$$R \bowtie S = \Pi_{r.A1, r.A1, r.A2, r.A3, s.A1, s.F1, s.F2} (\sigma_{r.A1=s.A1} (R \times S))$$

## Σύνθετο πρωτεύον κλειδί

A	B	C	D
1	7	7	2
1	7	5	1
2	8	2	3
3	9	2	2

Ποιοι συνδυασμοί των παραπάνω πεδίων αποτελούν πρωτεύον κλειδί?

## Ενότητα 6

---

### Άσκηση: ΒΔ διαχείρισης προσωπικού

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

1. Μετατροπή σχήματος Ο/Σ σε σχεσιακό
2. Ισχυροί τύποι οντοτήτων
3. Ασθενείς τύποι οντοτήτων
4. Συσχετίσεις
5. 1-1 Δυαδική (μη ασθενείς) συσχέτιση
6. 1-N Δυαδική (μη ασθενείς) συσχέτιση
7. Γνωρίσματα





# Σκοποί ενότητας

---

Ο σκοπός της άσκησης αυτής είναι η παρουσίαση ενός ολοκληρωμένου παραδείγματος που ακολουθεί τα βήματα σχεδιασμού μιας ΒΔ. Ανάλυση απαιτήσεων, μοντέλο E-R, μετατροπή σε σχεσιακό μοντέλο

## Διαχείριση προσωπικού

---

Σχεδιασμός μιας ΒΔ που θα διαχειρίζεται το προσωπικό μιας επιχείρησης. Η βάση αυτή θα κρατά στοιχεία σε τους Υπαλλήλους, τα τμήματα που έχει, τα προστατευόμενα μέλη του κάθε υπαλλήλου, καθώς και τα έργα που εργάζεται ο κάθε υπάλληλος.

### Απαιτήσεις:

- Κάθε υπάλληλος μπορεί να ανήκει σε ένα μόνο τμήμα, αλλά κάθε τμήμα μπορεί να έχει πολλούς υπαλλήλους.
- Ένας εργαζόμενος μπορεί να εργάζεται σε πολλά έργα ή σε κανένα και σε κάθε έργο μπορεί να εργάζονται τουλάχιστον δύο υπάλληλοι.
- Ένας υπάλληλος μπορεί να έχει προστατευόμενα μέλη, ώστε να καθοριστεί ο μισθός του.

## Διαχείριση προσωπικού

---

### Πίνακες:

**Υπάλληλοι** → ΚωδικόςΥπαλλήλου, Όνομα, Επίθετο, Ημ\_Γέννησης, Φύλο, Α.Τ., Περιοχή, Διεύθυνση, Τηλέφωνο.

**Προστατευόμενα μέλη** → Όνομα, Επίθετο, Ημ\_Γέννησης, Φύλο, Σχέση

**Τμήματα** → ΚωδικόςΤμήματος

**Έργα** → ΚωδικόςΈργου, Τίτλος, Περιοχή



## Διαχείριση προσωπικού

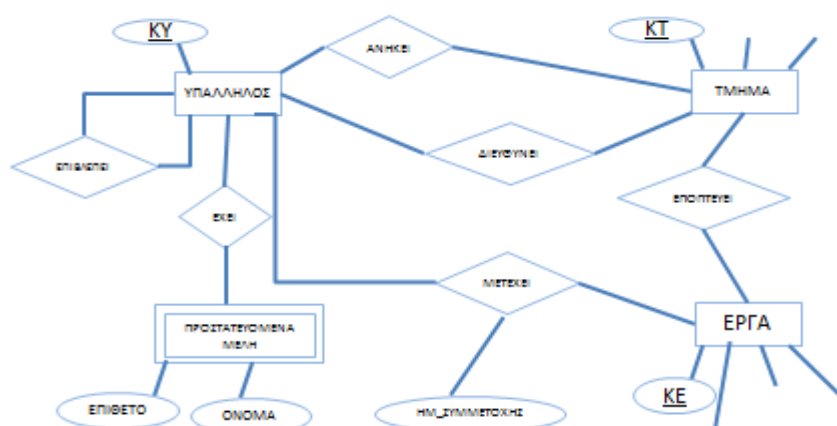
---

Για να γίνουν οι συσχετίσεις μεταξύ των πινάκων κάνετε τις εξής ερωτήσεις :

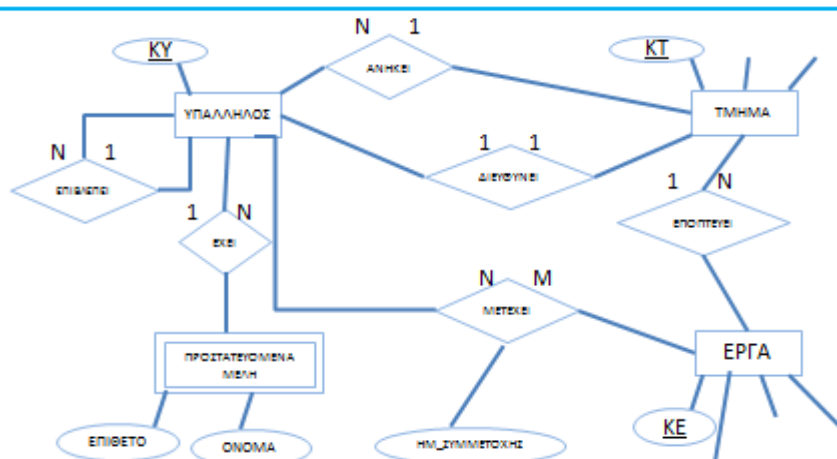
- Κάθε υπάλληλος ανήκει σε ένα τμήμα;
- Κάθε τμήμα έχει τουλάχιστον δύο υπαλλήλους;
- Κάθε τμήμα διευθύνεται από έναν υπάλληλο;
- Ένας υπάλληλος μπορεί να μετέχει σε πολλά έργα ;
- Σε κάθε έργο μπορεί να μετέχουν πολλοί υπάλληλοι;
- Το κάθε έργο εποπτεύεται από ένα τμήμα ;
- Ένα τμήμα μπορεί να εποπτεύει πολλά έργα;
- Ένας υπάλληλος μπορεί να επιβλέπει πολλούς άλλους υπαλλήλους ή κανέναν;
- Οι υπάλληλοι εποπτεύονται από έναν επόπτη;



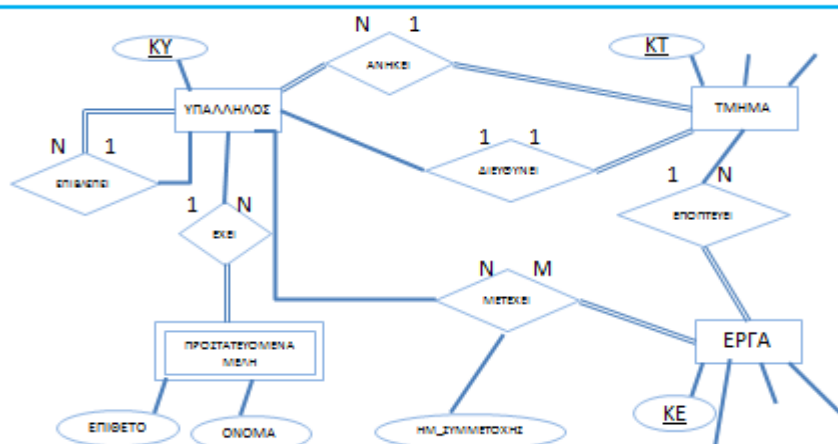
## Σχεδίαση



## Λόγος Πληθυκότητας



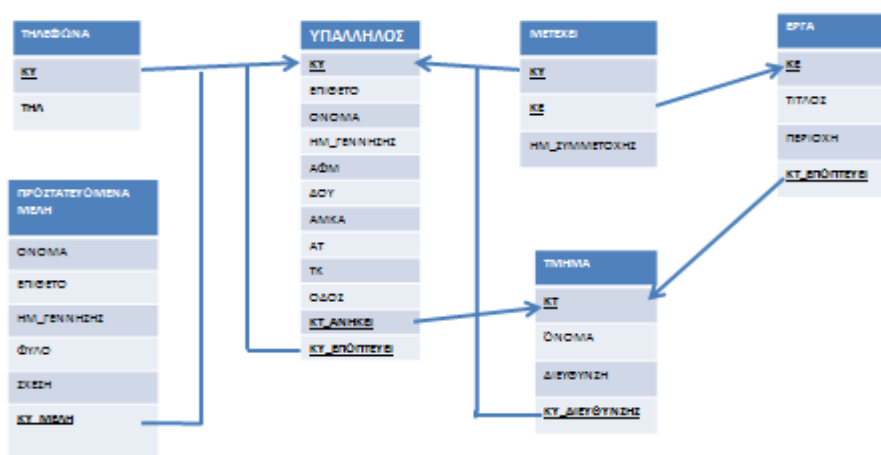
# Ολική-Μερική Συμμετοχή



# Σχεσιακό μοντέλο



# Σχεσιακό μοντέλο



## Ενότητα 7

---

### Εισαγωγή στη γλώσσα SQL

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Η γλώσσα SQL
- Εντολές Γλώσσας Ορισμού δεδομένων
- Δημιουργία, Τροποποίηση, Διαγραφή Πίνακα
- Περιορισμοί Κλειδιού
- Περιορισμοί οντότητας
- Αναφορική ακεραιότητα



# Σκοποί ενότητας

---

Στην ενότητα αυτή παρουσιάζονται οι βασικές εντολές της γλώσσας SQL. Γίνεται διαχωρισμός των εντολών σε Εντολές Γλώσσας Ορισμού δεδομένων που περιλαμβάνουν τη Δημιουργία, Τροποποίηση, Διαγραφή Πίνακα. Οι περιορισμοί κλειδιού οντότητας καθώς και η αναφορική ακεραιότητα απαιτούν ιδιαίτερη προσοχή ώστε να δημιουργηθούν σωστά οι πίνακες της ΒΔ.

# Η γλώσσα SQL

---

- Η **standard** γλώσσα για σχεσιακές βάσεις δεδομένων.
- αρχικά ονομάζονταν **Sequel** στην IBM ως μέρος του **System R**
- Σήμερα ονομάζεται **SQL** από τα αρχικά των λέξεων ( **Stuctured Query Language** )
- **SQL-89, SQL-92, SQL-99**

# Τμήματα της SQL

---

- Γλώσσα Ορισμού Δεδομένων (ΓΟΔ)
- Γλώσσα Χειρισμού Δεδομένων (ΓΧΔ)
- Ενσωματωμένη Γλώσσα Χειρισμού Δεδομένων
- Ορισμό Όψεων
- Εξουσιοδότηση (authentication)
- Ακεραιότητα
- Έλεγχο Συναλλαγών



# Η γλώσσα SQL

---

Η γλώσσα SQL αποτελείται από:

**DDL (Data Definition language)** → Γλώσσα ορισμού δεδομένων

- Create table → Δημιουργία νέου πίνακα
- Alter table → Τροποποίηση δομής πίνακα
- Drop table → Διαγραφή πίνακα

**DML (Data Manipulation language)** → Γλώσσα χειρισμού δεδομένων

- Insert into → Εισάγει νέες γραμμές στον πίνακα
- Update → Ενημερώνει/Διορθώνει τιμές των πεδίων του πίνακα
- Delete → Διαγράφει γραμμές από τον πίνακα
- Select → Επιλέγει και προβάλλει δεδομένα του πίνακα





# Τύποι Πεδίου Ορισμού

Για τον ορισμό του πεδίου ορισμού, οι διαθέσιμοι ενσωματωμένοι τύποι δεδομένων της SQL είναι:

**char(n)** (σταθερού μήκους)  
**varchar(n)**  
**int**  
**smallint**  
**numeric(p, d)** (d από τα p ψηφία είναι στα δεξιά της υποδιαστολής)  
**real, double precision**  
**float(n)**  
**date** (ημερομηνία)  
**time** (ώρα)

## Δημιουργία πινάκων στην SQL

Σπουδαστής			
AEM	Επίθετο	Πόλη	Ημ_γέννησης
1	AA	ΣΕΡΡΕΣ	1/1/1980
2	BB	ΑΘΗΝΑ	30/10/1990
3	CC	ΑΘΗΝΑ	6/1/1993

Δημιουργία του πίνακα με SQL:

```
Create table Σπουδαστής (AEM (int) NOTNULL,  
Επίθετο(varchar(30)),  
Πόλη(varchar(30)),  
Ημ_γέννησης(date),  
Constraint PK_AEM primary key (AEM) )
```

Περιορισμός ΚΛΕΙΔΙΟΥ

## Εντολές γλώσσας χειρισμού δεδομένων

---

Άλλες εντολές γλώσσας χειρισμού είναι οι εξής:

**DROP TABLE** → Διαγραφή πίνακα

**ALTER TABLE** → Τροποποίηση πίνακα

**ADD** → Πρόσθεση καινούργιας στήλης

**MODIFY** → Τροποποίηση μιας στήλης

**DROP** → Τροποποίηση μιας στήλης

**ΣΧΟΛΙΟ!** Όταν υπάρχει περιορισμός ακεραιότητας αναφοράς δεν μπορούμε να κάνουμε αλλαγή δεδομένων.



## Συνθήκες την εντολής Where

---

**Συγκριτικοί τελεστές:**

=, >, <, >=, <=, <>, Between, Not Between, In, Not in, like, not like  
(ανάμεσα σε αριθμητικές εκφράσεις, συμβολοσειρές ( strings ) και ειδικούς τύπους).

**Λογικοί τελεστές:**

AND, OR, NOT

**Αριθμητικοί τελεστές:**

+, -, \*, /



## Μεταβολή δομής πινάκων στην SQL

Σπουδαστής			
AEM	Επίθετο	Πόλη	Ημ_ γέννησης
1	AA	ΣΕΡΡΕΣ	1/1/1980
2	BB	ΑΘΗΝΑ	30/10/1990
3	CC	ΑΘΗΝΑ	6/1/1993

Δημιουργία του πίνακα με SQL:

```
ALTER TABLE Σπουδαστής ADD COLUMN ΤΗΛΕΦΩΝΟ VARCHAR(20)
```

```
ALTER TABLE Σπουδαστής MODIFY ΤΗΛΕΦΩΝΟ VARCHAR(10)
```

```
ALTER TABLE Σπουδαστής DROP ΤΗΛΕΦΩΝΟ
```



## Αναφορική Ακεραιότητα

ΜΑΘΗΜΑ			
<u>ΚΜ</u>	ΤΙΤΛΟΣ	ΔΙΑ_ΜΟ ΝΑΔΕΣ	ΚΚ
1	ΒΑΣΕΙΣ Ι	3	1
2	ΛΟΓΙΣΜΟΣ	2	1
3	ΛΕΙΤΟΥΡΓΙΚΑ	3	2

ΚΑΘΗΓΗΤΗΣ			
<u>ΚΚ</u>	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΠΟΛΗ
1	BBB	BBB	ΣΕΡΡΕΣ
2	AAA	AAA	ΑΘΗΝΑ
3	CCC	CCC	ΑΘΗΝΑ

Δημιουργία του πίνακα με SQL:

```
ALTER TABLE ΜΑΘΗΜΑ ADD CONSTRAINT FK1 FOREIGN KEY (ΚΚ)  
REFERENCES ΚΑΘΗΓΗΤΗΣ(ΚΚ)
```



## Τιμή Null

---

Η λέξης κλειδί **is null** (**is not null**) σε μια συνθήκη υπάρχει για να διαπιστωθεί αν μια τιμή είναι null (κενή). Εάν υπάρχει τιμή τότε είναι **not null**.

## Ενότητα 8

---

### SQL-Γλώσσα χειρισμού δεδομένων

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Προβολή δεδομένων πίνακα
- Πράξεις με Τελεστές (Αριθμητικοί, Συγκριτικοί, Λογικοί)
- Πράξεις Συνόλων



# Σκοποί ενότητας

---

Στην ενότητα αυτή παρουσιάζονται οι εντολές της Γλώσσας Χειρισμού δεδομένων για Προβολή δεδομένων πίνακα με χρήση τελεστών (Αριθμητικοί, Συγκριτικοί, Λογικοί). Επίσης παρουσιάζονται και εντολές για Πράξεις Συνόλων.



# Η γλώσσα SQL

---

- Η **standard** γλώσσα για σχεσιακές βάσεις δεδομένων.
- αρχικά ονομάζονταν Sequel στην IBM ως μέρος του System R
- Σήμερα ονομάζεται SQL από τα αρχικά των λέξεων ( Structured Query Language )
- SQL--89, SQL--92, SQL-99



## Εντολές SQL χειρισμού δεδομένων

---

- Insert εισαγωγή εγγραφών
- Update ενημέρωση τιμών πεδίων
- Delete Διαγραφή εγγραφών πίνακα (drop?)
- Select



## Εισαγωγή δεδομένων

---

- Insert into Πίνακας Values (τιμή1, τιμή2.....) → Εισαγωγή των τιμών στα αντίστοιχα πεδία του πίνακα.
- Insert into Πίνακας(field2, field1) Values (τιμή2, τιμή1) → Εισαγωγή των τιμών στα συγκεκριμένα πεδία του πίνακα με άλλη σειρά.
- Insert into Πίνακας1 Select F1,F2.. From Πίνακας2 → Εισαγωγή τιμών στα αντίστοιχα πεδία του πίνακα1 που προέρχονται από τον πίνακα2.
- Insert into Σπουδαστής Values (3, '22', 'ΣΕΡΡΕΣ', '1/1/1980') → Εισαγωγή των τιμών στα αντίστοιχα πεδία του πίνακα.
- 'κείμενο'
- 'ημερομηνία' ή #ημερομηνία#



## Ενημέρωση δεδομένων

- Update ΠΙΝΑΚΑΣ Set ΠΕΔΙΟ= ΤΙΜΗ → Θα αλλάξει όλες τις τιμές του συγκεκριμένου πεδίου σε όλο τον πίνακα

Σπουδαστής			
AEM	Επίθετο	Πόλη	Ημ_γέννησης
1	AA	ΣΕΡΡΕΣ	1/1/1980
2	BB	ΑΘΗΝΑ	30/10/1990
3	CC	ΑΘΗΝΑ	6/1/1993

- Χρήση περιορισμών με την εντολή WHERE
- Update Σπουδαστής Set Πόλη = 'Αθήνα' → Θα αλλάξει για όλους τους σπουδαστές την πόλη.
- Update Σπουδαστής Set Πόλη = 'Βόλος' Where AEM=3 → Θα αλλάξει για τον σπουδαστή με AEM=3 την πόλη.

## Συνθήκες την εντολής Where

### Συγκριτικοί τελεστές:

=, >, <, >=, <=, <>, Between, Not Between, In, Not in  
(ανάμεσα σε αριθμητικές εκφράσεις, συμβολοσειρές ( strings ) και ειδικούς τύπους).

### Λογικοί τελεστές:

AND, OR, NOT



## Ενημέρωση δεδομένων

---

**Παράδειγμα:** Αύξηση όλων των μισθών των εργαζομένων που είναι μεγαλύτεροι ή ίσοι των € 1000 κατά 5%

```
update ΕΡΓΑΖΟΜΕΝΟΙ
set Μισθός= Μισθός* 1.05
where Μισθός >= 100
```

## Ενημέρωση δεδομένων

---

*Παράδειγμα:*

*Αύξηση στους υπαλλήλους που έχουν μισθό < 1.000€ η εταιρεία δίνει 5% και στους υπαλλήλους που έχουν μισθό > 1.000 δίνει 9%:*

```
update ΕΡΓΑΖΟΜΕΝΟΙ
set Μισθός = Μισθός * 1.05
where Μισθός < 1.000

update ΕΡΓΑΖΟΜΕΝΟΙ
set Μισθός = Μισθός * 1.09
where Μισθός >= 1.000
```

Ποιο update πρέπει να τρέξουμε πρώτα;

## Ενημέρωση δεδομένων

---

Παράδειγμα: Αύξηση όλων των μισθών που είναι μεγαλύτεροι από τον μέσο όρο κατά 5%

```
update ΕΡΓΑΖΟΜΕΝΟΙ
set Μισθός = Μισθός * 1.05
where Μισθός > select avg(Μισθός )
from ΕΡΓΑΖΟΜΕΝΟΙ
```

## Βασική σύνταξη γλώσσας χειρισμού δεδομένων

---

Βασική Σύνταξη Γλώσσας Χειρισμού Δεδομένων (ΓΧΔ):

**select**  $A_1, A_2, \dots, A_n$  ονόματα γνωρισμάτων  
**from**  $R_1, R_2, \dots, R_m$  ονόματα σχέσεων  
**where**  $P$  συνθήκη

Ισοδύναμο του:  $\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$

## Βασική σύνταξη γλώσσας χειρισμού δεδομένων

---

**select** A<sub>1</sub>, A<sub>2</sub>, ..., A<sub>n</sub>  
**from** R<sub>1</sub>, R<sub>2</sub>, ... R<sub>m</sub>  
**where** P

**select** αντιστοιχεί στην πράξη της προβολής της σχεσιακής άλγεβρας. Ποια γνωρίσματα θέλουμε να υπάρχουν στο αποτέλεσμα της ερώτησης.

$\pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_m))$



## Βασική σύνταξη γλώσσας χειρισμού δεδομένων

---

**select** A<sub>1</sub>, A<sub>2</sub>, ..., A<sub>n</sub>  
**from** R<sub>1</sub>, R<sub>2</sub>, ... R<sub>m</sub>  
**where** P

**from** αντιστοιχεί στην πράξη του καρτεσιανού γινομένου της σχεσιακής άλγεβρας. Ποιες σχέσεις θα χρησιμοποιηθούν για τον υπολογισμό του αποτελέσματος.

$\pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_m))$



## Βασική σύνταξη γλώσσας χειρισμού δεδομένων

---

```
select A1, A2, ..., An  
from R1, R2, ..., Rm  
where P
```

**where** αντιστοιχεί στη συνθήκη της πράξης της επιλογής στη σχεσιακή άλγεβρα. Το κατηγορήμα **P** έχει γνωρίσματα των σχέσεων που εμφανίζονται στο **from**.

$$\pi_{A_1, A_2, \dots, A_n}(\sigma_P(R_1 \times R_2 \times \dots \times R_m))$$


## Βασική σύνταξη γλώσσας χειρισμού δεδομένων

---

### Συμβουλές

- Είτε γράφετε με μικρά είτε με κεφαλαία στην είναι το ίδιο.
- Στην SQL δεν είναι έτοιμοι οι πίνακες εσείς τους δημιουργείτε με εντολές.
- Όταν δεν υπάρχει το **where**, το **P** θεωρείται ότι ισχύει.
- **Προσοχή** δεν γίνεται απαλοιφή των διπλών εμφανίσεων.
- Για απαλοιφή διπλών εμφανίσεων:

```
Select distinct Όνομα_ Πεδίου  
From Πίνακας
```



## Τιμή Null

---

Η λέξη κλειδί `is null` (`is not null`) σε μια συνθήκη υπάρχει για να διαπιστωθεί αν μια τιμή είναι null (κενή). Εάν υπάρχει τιμή τότε είναι **not null**.



## Πράξεις με συμβολοσειρές

---

Η πιο συνηθισμένη πράξη είναι το ταίριασμα προτύπων:

- **Επί τις εκατό %** ταιριάζει οποιαδήποτε συμβολοσειρά (String)
- **Κάτω παύλα \_** ταιριάζει οποιοδήποτε χαρακτήρα
- **Ερωτηματικό αγγλικό ?**
- **Αστεράκι \***
- **Δίεση #**

Σύγκριση χρησιμοποιώντας το `like`, `not like`

**ΣΧΟΛΙΟ!** Γίνεται διάκριση ανάμεσα σε κεφαλαία και μικρά.



## Παραδείγματα πράξεων με συμβολοσειρές

Εμφανίστε όλους τους σπουδαστές που στο επίθετό τους εμπεριέχεται το γράμμα «Σ».

```
Select *  
From Σπουδαστές  
Where Πόλη Like '%Σ%'
```

Εμφανίστε όλους τους σπουδαστές που στο επίθετό τους αρχίζει με το γράμμα «Σ».

```
Select *  
From Σπουδαστές  
Where Πόλη Like 'Σ%'
```



Όταν τελειώνει: '% Σ'

Εμφανίστε όλους τους σπουδαστές που στο επίθετό τους αρχίζει και τελειώνει με το γράμμα «Σ».

```
Select *  
From Σπουδαστές  
Where Πόλη Like 'Σ%Σ'
```

## Διάταξη πεδίων

Η διάταξη των πεδίων σε:

- αύξουσα σειρά → ASC
- φθίνουσα σειρά → DESC

Παράδειγμα: Ταξινομήστε τους σπουδαστές κατά αύξουσα σειρά βάση επιθέτου.

```
Select *  
From Σπουδαστές  
Order By Επίθετο ASC
```

## Πράξεις συνόλων

---

Πράξεις:

- union
- intersection
- except

εφαρμόζονται σε συμβατές σχέσεις.

## Ενότητα 9

---

### SQL-Σύζευξη πινάκων

Δρ. Τιμπίρης Αλκιβιάδης



## Περιεχόμενα ενότητας

---

- Γενικός τύπος εντολής SELECT
- Καρτεσιανό γινόμενο πινάκων
- Εσωτερική Σύζευξη
- Ερωτήματα εσωτερικής σύζευξης
- WERE και INNER JOIN





## Σκοποί ενότητας

---

Σκοπός της ενότητας αυτής είναι κατανόηση της έννοιας της σύζευξης πινάκων. Είναι ένας από τους λόγους που επικράτησε το σχεσιακό μοντέλο. Όταν απαιτούνται δεδομένα από πολλούς πίνακες, οι οποίοι συσχετίζονται, θα πρέπει να δίνεται προσοχή στα ξένα κλειδιά που υπάρχουν στους πίνακες και στα πρωτεύοντα κλειδιά με τα οποία συσχετίζονται.



## Γενικός τύπος

---

```
SELECT A1 ,A2, ....., An  
FROM R1, R2,....  
WHERE P1 AND P2 OR .... NOT ....  
GROUP BY A1, A2,...  
HAVING H1 AND H2 OR .... NOT ....  
ORDER BY A1, A2, ... (ASC/DESC)
```



# Άσκηση-Σπουδαστής

Βαθμολογίες			Σπουδαστής		
ΑΕΜ	ΚΜ	ΒΑΘΜΟΣ	ΑΕΜ	Επίθετο	Πόλη
2	1	5	1	ΑΑ	ΣΕΡΡΕΣ
2	2	8	2	ΒΒ	ΑΘΗΝΑ
1	1	8	3	CC	ΑΘΗΝΑ
1	3	6			
1	2	4			

Μάθημα		
ΚΜ	Τίτλος	Εξάμηνο
1	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	ΣΕΡΡΕΣ
2	ΠΛΗΡΟΦΟΡΙΚΗ	ΑΘΗΝΑ
3	ΜΑΘΗΜΑΤΙΚΑ	ΑΘΗΝΑ



## Καρτεσιανό γινόμενο πινάκων Βαθμολογίες-Σπουδαστής

Βαθμολογίες			Σπουδαστής		
ΑΕΜ	ΚΜ	ΒΑΘΜΟΣ	ΑΕΜ	Επίθετο	Πόλη
2	1	5	1	ΑΑ	ΣΕΡΡΕΣ
2	1	5	2	ΒΒ	ΑΘΗΝΑ
2	1	5	3	CC	ΣΕΡΡΕΣ
2	2	8	1	ΑΑ	ΣΕΡΡΕΣ
2	2	8	2	ΒΒ	ΑΘΗΝΑ
2	2	8	3	CC	ΣΕΡΡΕΣ
1	1	8	1	ΑΑ	ΣΕΡΡΕΣ
1	1	8	2	ΒΒ	ΑΘΗΝΑ
1	1	8	3	CC	ΣΕΡΡΕΣ
1	3	6	1	ΑΑ	ΣΕΡΡΕΣ
1	3	6	2	ΒΒ	ΑΘΗΝΑ
1	3	6	3	CC	ΣΕΡΡΕΣ
1	2	4	1	ΑΑ	ΣΕΡΡΕΣ
1	2	4	2	ΒΒ	ΑΘΗΝΑ
1	2	4	3	CC	ΣΕΡΡΕΣ

Ποιο το ανάλογο στη σχεσιακή άλγεβρα?



## Ερωτήματα Σύζευξης

---

Να εμφανίσετε τον τίτλο του μαθήματος και τον βαθμό.

```
Select Τίτλος, Βαθμός  
From Βαθμολογίες, Μάθημα  
Where Μάθημα.KM= Βαθμολογίες.KM
```

Ποιο το ανάλογο  
στη σχεσιακή άλγεβρα?

Να εμφανιστούν τα επίθετα των φοιτητών που γράφουν κάτω από τη βάση.

```
Select Επίθετο  
From Βαθμολογίες, Σπουδαστής  
Where Σπουδαστής.AEM= Βαθμολογίες.AEM And Βαθμός>5
```



## Ερωτήματα Σύζευξης

---

Να εμφανίσετε τις βαθμολογίες με τα μαθήματα που πέρασαν οι φοιτητές από Σέρρες.

```
Select Επίθετο, Όνομα, Τίτλος, Βαθμός  
From Βαθμολογίες, Μάθημα, Βαθμολογίες  
Where Μάθημα.KM= Βαθμολογίες.KM And Σπουδαστής.AEM=Βαθμολογίες .AEM  
And Βαθμός>=5 And Πόλη= 'Σέρρες'
```

Να εμφανιστούν ποια μαθήματα πέρασαν οι φοιτητές.

```
Select Distinct Τίτλος  
From Βαθμολογίες, Μάθημα, Βαθμολογίες  
Where Μάθημα.KM= Βαθμολογίες.KM And  
Σπουδαστής.AEM=Βαθμολογίες .AEM And  
Βαθμός>=5
```

Distinct= Έχει την ιδιότητα όποια εγγραφή διπλό-εμφανίζεται να μην την εμφανίζει



## Ερωτήματα Σύζευξης

---

Η SQL--92 υποστηρίζει διάφορους τύπους σύζευξης που συνήθως χρησιμοποιούνται στο from, αλλά μπορούν να χρησιμοποιηθούν οπουδήποτε μπορεί να χρησιμοποιηθεί μια σχέση.

Τύποι σύζευξης:

**inner join**: εσωτερική (θήτα) συνένωση

**left outer join**: αριστερή εξωτερική συνένωση

**right outer join**

**full outer join**



## Ερωτήματα Σύζευξης

---

**Να εμφανίσετε τον τίτλο του μαθήματος και τον βαθμό.**

```
Select Τίτλος, Βαθμός  
From Βαθμολογίες inner join Μάθημα  
On Μάθημα.KM= Βαθμολογίες.KM
```

**Να εμφανιστούν τα επίθετα των φοιτητών που γράφουν κάτω από τη βάση.**

```
Select Επίθετο  
From Βαθμολογίες inner join Σπουδαστής  
On Σπουδαστής.AEM= Βαθμολογίες.AEM And Βαθμός>5
```



## Ερωτήματα Σύζευξης

---

Να εμφανίσετε τις βαθμολογίες με τα μαθήματα που πέρασαν οι φοιτητές από Σέρρες.

```
Select  Επίθετο, Όνομα, Τίτλος, Βαθμός
From    Βαθμολογίες inner join (Μάθημα inner join Βαθμολογίες
        on Σπουδαστής.AEM=Βαθμολογίες .AEM )
        on Μάθημα.KM= Βαθμολογίες.KM
Where   Βαθμός>=5 And Πόλη= 'Σέρρες'
```



## Ενότητα 10

---

### Συναθροιστικές συναρτήσεις

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Ποιες είναι οι συναθροιστικές συναρτήσεις
- Max-Min
- AVG
- SUM
- COUNT
- Ερωτήματα Ομαδοποίησης
- Εμφωλευμένα Ερωτήματα



# Σκοποί ενότητας

---

Σκοπός της ενότητας αυτής είναι η εισαγωγή στις συναθροιστικές συναρτήσεις. Με απλά ερωτήματα και με τη χρήση αυτών των συναρτήσεων τα αποτελέσματα είναι εντυπωσιακά. Σε ελάχιστο χρόνο και με λίγες γραμμές κώδικα μπορούν να υπολογιστούν στατιστικά από πίνακες μεγάλου όγκου δεδομένων.



## Συναθροιστικές συναρτήσεις

---

Η SqI έχει πέντε συναθροιστικές συναρτήσεις:

- Μέσος όρος: **avg(A)**
- Ελάχιστο: **min(A)**
- Μέγιστο: **max(A)**
- Άθροισμα: **sum(A)** (μόνο σε αριθμούς)
- Πλήθος: **count(A)**

**Σχόλιο!** Το A είναι γνώρισμα



## Άσκηση-Εργαζόμενος

Εργαζόμενος				
ΚΕ	Επίθετο	Μισθός	Πόλη	Τμήμα
1	AA	1.000	Σέρρες	Αποθήκη
2	BB	2.000	Αθήνα	Πωλήσεις
3	CC	1.000	Σέρρες	Λογιστήριο
4	DD	1.500	Θεσσαλονίκη	Πωλήσεις



## Συνάρτηση Sum

Γενική σύνταξη της συνάρτησης Sum: `Select Sum(Πεδίο ) As Σύνολο  
From Πίνακας  
Where Περιορισμός`

Να βρεθεί το σύνολο των εργαζομένων κάθε μήνα.

```
Select Sum(Μισθός ) As Σύνολο  
From Εργαζόμενος
```





## Συνάρτηση Avg

---

Γενική σύνταξη της συνάρτησης Avg: `Select Avg(Πεδίο ) As Μέσος όρος  
From Πίνακας  
Where Περιορισμός`

Να βρεθεί ο μέσος όρος του μισθού κάθε μήνα

```
Select Avg(Μισθός ) As MO  
From Εργαζόμενος
```

Να βρεθεί ο μέσος όρος των μισθών κάθε μήνα που παίρνουν οι Σερραίοι.

```
Select Avg(Μισθός ) As MO  
From Εργαζόμενος  
Where Πόλη= 'Σέρρες'
```



## Συνάρτηση Count

---

Γενική σύνταξη της συνάρτησης Avg: `Select Count(Πεδίο ) As Πλήθος  
From Πίνακας  
Where Περιορισμός`

### Παραδείγματα

Να βρεθεί το πλήθος των εργαζομένων.

```
Select Count(ΚΕ ) As Πλήθος  
From Εργαζόμενος
```

Να βρεθεί το πλήθος των μοναδικών μισθών των εργαζομένων.

```
Select Count(Μισθός Distinct ) As πλήθος  
From Εργαζόμενος
```

**ΣΧΟΛΙΟ!** Το Count(\*) μετράει το πλήθος γραμμών.



# Συναρτήσεις Min-Max

---

Γενική σύνταξη των Min-Max:

```
Select Min(Πεδίο ) As Ελάχιστος  
From Πίνακας  
Where Περιορισμός
```

```
Select Max(Πεδίο ) As Μέγιστος  
From Πίνακας  
Where Περιορισμός
```

## Παράδειγμα

Να βρεθεί ο ελάχιστος μισθός των εργαζομένων.

```
Select Min(Μισθός ) As Ελάχιστος  
From Εργαζόμενος
```

Ομοίως και για την  
συνάρτηση Max



# Εμφωλευμένα ερωτήματα

---

Ένα ερώτημα είναι εμφωλευμένο όταν περικλείεται μέσα σε ένα άλλο.

## Παράδειγμα

Να εμφανιστεί το επίθετο αυτού που έχει τον μεγαλύτερο μισθό.

```
Select Επίθετο, Μισθός  
From Εργαζόμενος  
Where Μισθός=(Select Max(Μισθός) As Μέγιστος  
From(Εργαζόμενος)
```



# Ομαδοποίηση με συναθροιστικές συναρτήσεις

---

## Παραδείγματα:

Να υπολογιστεί για κάθε τμήμα ποία είναι η μισθοδοσία που δίνεται.

```
Select Τμήμα ,Sum(Μισθός) As Σύνολο  
From Εργαζόμενος  
Group by Τμήμα
```

Για κάθε πόλη πόσοι εργαζόμενοι παίρνουν 1500 ευρώ και πάνω.

```
Select Πόλη ,Count (*) As Αριθμός  
From Εργαζόμενος  
Where Μισθός >= 1.500  
Group by Πόλη
```



# Εμφωλευμένα ερωτήματα

---

## Παραδείγματα:

Να εμφανιστεί ο μέγιστος μισθός.

```
Select Max(Μισθός)  
From Εργαζόμενος
```

Βρείτε τα στοιχεία των εργαζομένων που έχουν το μέγιστο μισθό.

```
Select *  
From Εργαζόμενος  
Where Μισθός=(Select Max(Μισθός)  
From Εργαζόμενος)  
Order by ΚΕ
```



# Εμφωλευμένα ερωτήματα

---

## Παραδείγματα:

Ποιοι είναι οι εργαζόμενοι που είναι από την ίδια πόλη που είναι και ο εργαζόμενος ΑΑ.

```
Select *  
From Εργαζόμενος  
Where Πόλη=(Select Πόλη  
From Εργαζόμενος  
Where Επίθετο = ' ΑΑ ')
```

## Ενότητα 11

---

### SQL-Ερωτήματα Ομαδοποίησης με Συνένωση Πινάκων

Δρ. Τσιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Τύποι σύζευξης
- Εφαρμογή σε σύνθετα προβλήματα
- Πολλαπλοί πίνακες
- Στατιστικά με σύνθετα ερωτήματα
- Ομαδοποίηση και σύζευξη



# Σκοποί ενότητας

---

Σε αυτή την ενότητα παρουσιάζονται τρόποι αντιμετώπισης σύνθετων ερωτημάτων όπου υπάρχουν πολλοί πίνακες σε σύζευξη και πρέπει να γίνει υπολογισμός συναθροιστικών συναρτήσεων με πολλούς περιορισμούς. Παρουσιάζονται παραδείγματα ανάλογα με την πραγματικά προβλήματα όπου ζητούνται απλά στατιστικά στοιχεία με σύνθετα ερωτήματα ομαδοποίησης και σύζευξης



# Τύποι Σύζευξης

---

**Υπάρχουν οι εξής τύποι σύζευξης:**

INNER JOIN → Εσωτερική σύζευξη

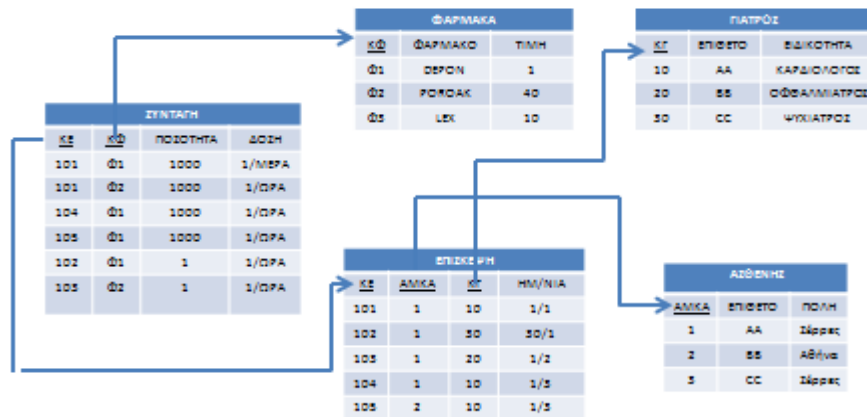
LEFT OUTER JOIN → Αριστερή εξωτερική σύζευξη

RIGHT OUTER JOIN → Δεξιά εξωτερική σύζευξη

FULL OUTER JOIN → Γενική εξωτερική σύζευξη



# Άσκηση-Νοσοκομείο



## Σύζευξη

### Παραδείγματα:

Εμφανίστε το επίθετο, το φάρμακο, την ποσότητα, και την ημερομηνία που πήρε την συνταγή ο ασθενής.

```
Select ΕΠΙΘΕΤΟ, ΦΑΡΜΑΚΟ, ΠΟΣΟΤΗΤΑ, ΗΜ/ΝΙΑ
From ΑΣΘΕΝΗΣ INNER JOIN (ΕΠΙΣΚΕΨΗ,
INNER JOIN (ΣΥΝΤΑΓΗ,
INNER JOIN (ΦΑΡΜΑΚΟ
ON ΦΑΡΜΑΚΟ.ΚΦ=ΣΥΝΤΑΓΗ .ΚΦ)
ON ΣΥΝΤΑΓΗ.ΚΕ=ΕΠΙΣΚΕΨΗ.ΚΕ)
ON ΑΣΘΕΝΗΣ.ΑΜΚΑ=ΕΠΙΣΚΕΨΗ.ΑΜΚΑ
ORDER BY ΕΠΙΘΕΤΟ, ΗΜ/ΝΑΙ DESC
```



## Ομαδοποίηση και σύζευξη

### Παραδείγματα:

Εμφανίστε το ΑΜΚΑ, το επίθετο, το πλήθος των επισκέψεων του κάθε ασθενή και το πλήθος των ασθενών ταξινομημένο σε φθίνουσα σειρά.

```
Select ΑΣΘΕΝΗΣ.ΑΜΚΑ, ΕΠΙΘΕΤΟ, Count( ΚΕ) AS ΠΛΗΘΟΣ  
From ΑΣΘΕΝΗΣ, ΕΠΙΣΚΕΨΗ  
Where ΑΣΘΕΝΗΣ.ΑΜΚΑ=ΕΠΙΣΚΕΨΗ.ΑΜΚΑ  
Group by ΑΣΘΕΝΗΣ.ΑΜΚΑ, ΕΠΙΘΕΤΟ  
Order BY ΠΛΗΘΟΣ Desc
```

Εμφανίζει:

ΑΜΚΑ	ΕΠΙΘΕΤΟ	ΠΛΗΘΟΣ
1	ΑΑ	4
2	ΒΒ	1



## Ομαδοποίηση και σύζευξη

### Παραδείγματα:

Εμφανίστε το ΑΜΚΑ του κάθε ασθενή και το κόστος των φαρμάκων που του συνταγογράφησαν οι γιατροί και το πλήθος των ασθενών ταξινομημένο σε φθίνουσα σειρά.

```
Select ΑΜΚΑ, Sum(ΤΙΜΗ*ΠΟΣΟΤΗΤΑ) AS ΚΟΣΤΟΣ  
From ΕΠΙΣΚΕΨΗ, ΦΑΡΜΑΚΟ, ΣΥΝΤΑΓΗ  
Where ΕΠΙΣΚΕΨΗ.ΚΕ=ΣΥΝΤΑΓΗ.ΚΕ  
And ΦΑΡΜΑΚΟ.ΚΦ= ΣΥΝΤΑΓΗ.ΚΦ  
Group by ΑΜΚΑ  
Order BY ΚΟΣΤΟΣ Desc
```

Εμφανίζει:

ΑΜΚΑ	ΚΟΣΤΟΣ
1	42.011
2	1.000





## Ομαδοποίηση και σύζευξη

### Παραδείγματα:

Εμφανίστε το ΚΓ του κάθε γιατρού, το κόστος των φαρμάκων που συνταγογράφησαν και ταξινομήστε το σε φθίνουσα σειρά.

```
Select ΚΓ, Sum(ΤΙΜΗ*ΠΟΣΟΤΗΤΑ) AS ΚΟΣΤΟΣ
From ΕΠΙΣΚΕΨΗ, ΦΑΡΜΑΚΟ, ΣΥΝΤΑΓΗ
Where ΕΠΙΣΚΕΨΗ.ΚΕ=ΣΥΝΤΑΓΗ.ΚΕ
And ΦΑΡΜΑΚΟ.ΚΦ= ΣΥΝΑΤΓΗ.ΚΦ
Group by ΚΓ
Order BY ΚΟΣΤΟΣ Desc
```

Εμφανίζει:

ΚΓ	ΚΟΣΤΟΣ
10	43.000
20	10
30	1



## Ομαδοποίηση και σύζευξη

### Παραδείγματα:

Εμφανίστε το ΚΦ για το κάθε φάρμακο που συνταγογράφησαν και ταξινομήστε το σε φθίνουσα σειρά.

```
Select ΚΦ, Sum(ΤΙΜΗ*ΠΟΣΟΤΗΤΑ) AS ΚΟΣΤΟΣ
From ΦΑΡΜΑΚΟ, ΣΥΝΤΑΓΗ
Where ΦΑΡΜΑΚΟ.ΚΦ= ΣΥΝΑΤΓΗ.ΚΦ
Group by ΚΦ
Order BY ΚΟΣΤΟΣ Desc
```

Εμφανίζει:

ΦΑΡΜΑΚΑ	ΚΟΣΤΟΣ
Φ2	40.000
Φ1	3.000
Φ3	10





# Ενότητα 12

---

## Κανονικοποίηση

Δρ. Τσιμπήρης Αλκιβιάδης



## Περιεχόμενα ενότητας

---

- Θεωρία Κανονικοποίησης
- Κανονικές Μορφές (normal forms)
- Πρώτη Κανονική Μορφή 1ΚΜ
- Δεύτερη Κανονική Μορφή 2ΚΜ
- Τρίτη Κανονική Μορφή 3ΚΜ



## Σκοποί ενότητας

---

Στην ενότητα αυτή παρουσιάζεται ένας ακόμη τρόπος σχεδιασμού βάσεων δεδομένων ο οποίος βασίζεται στην Θεωρία Κανονικοποίησης σύμφωνα με την οποία δημιουργείται ένα μεγάλος πίνακας που περιέχει όλα τα γνωρίσματα που απατούνται και στη συνέχεια απόδομείται. Αυτό επιτυγχάνεται ορθά με την εφαρμογή των Κανονικών Μορφών (normal forms). Η Πρώτη Κανονική Μορφή (1KM) είναι αυτή στην οποία πρέπει να βρίσκονται όλοι οι πίνακες μιας ΒΔ. Επίσης η Δεύτερη Κανονική Μορφή (2KM) και η Τρίτη Κανονική Μορφή (3KM) αντιμετωπίζουν το μεγαλύτερο μέρος των πιθανών προβλημάτων κατά τον σχεδιασμό μιας ΒΔ.



## Θεωρία Κανονικοποίησης

---

- Μία σχέση είναι ένα σύνολο από γνωρίσματα με τιμές για κάθε γνώρισμα τέτοιες ώστε να ισχύουν οι παρακάτω ιδιότητες
  1. Κάθε όνομα γνωρίσματος είναι μοναδικό.
  2. Όλες οι τιμές κάθε γνωρίσματος είναι ίδιου τύπου (ή πεδίου ορισμού).
  3. Κάθε τιμή γνωρίσματος είναι ατομική (μία τιμή και όχι ομάδα πολλών τιμών).
  4. Τα γνωρίσματα δεν έχουν διάταξη από τα αριστερά προς τα δεξιά..
  5. Οι συστοιχίες (σειρές) δεν έχουν διάταξη από επάνω προς τα κάτω.
  6. Δεν υπάρχουν δύο ίδιες σειρές (συστοιχίες) σε μία σχέση.

## Διαδικασία σωστού σχεδιασμού ΒΔ

---

Η διαδικασία που ακολουθούμε είναι :

- Συγκεντρώνουμε τις απαιτήσεις της επιχείρησης και των χρηστών.
- Σχεδιάζουμε το μοντέλο οντοτήτων-συσχετίσεων
- Μετατρέπουμε το διάγραμμα οντοτήτων-συσχετίσεων της επιχείρησης σε ένα σύνολο από σχέσεις (πίνακες) με το σχεσιακό μοντέλο.
- Κανονικοποιούμε τις σχέσεις για να απομακρύνουμε τυχόν ανωμαλίες ενημέρωσης-διαγραφής-εισαγωγής στοιχείων.
- Υλοποιούμε τη βάση δεδομένων δημιουργώντας ένα πίνακα για κάθε κανονικοποιημένη σχέση.

## Κανονικοποίηση-Κανονικές μορφές

---

- Κατηγορίες (ή κλάσεις) **Κανονικές Μορφές** (normal forms).
  - **Κανονική Μορφή**: Μία κλάση σχέσεων απαλλαγμένων από συγκεκριμένα προβλήματα τροποποιήσεων.
1. Πρώτη κανονική μορφή (1NF – 1KM)
  2. Δεύτερη κανονική μορφή (2NF – 2KM)
  3. Τρίτη κανονική μορφή (3NF – 3KM)
  4. Boyce-Codd κανονική μορφή (BCNF – KM BC)
  5. Τέταρτη κανονική μορφή (4NF – 4KM)
  6. Πέμπτη κανονική μορφή (5NF – 5KM)
  7. Κανονική μορφή πεδίου ορισμού κλειδιού (Domain-Key/NF)

## Κανονικοποίηση-Κανονικές μορφές

---

- Αυτές οι κανονικές μορφές είναι αθροιστικές. Μία σχέση που βρίσκεται σε Τρίτη κανονική μορφή είναι επίσης και σε δεύτερη και σε πρώτη.
- Οι τρεις πρώτες κανονικές μορφές (1KM, 2KM, 3KM) ορίστηκαν από τον Codd. Όλες οι κανονικοποιημένες σχέσεις είναι σε 1KM.
- Με άλλα λόγια, "κανονικοποιημένη" και "σε 1KM" σημαίνει ακριβώς το ίδιο πράγμα.
- Μερικές σχέσεις 1KM είναι επίσης σε 2KM, και μερικές σχέσεις 2KM είναι επίσης σε 3KM.

## Κανονικοποίηση-Κανονικές μορφές

---

- Ο Fagin όρισε την τέταρτη κανονική μορφή.
- Μετέπειτα και πάλι ο Fagin όρισε άλλη μία κανονική μορφή, την κανονική μορφή προβολής-σύζευξης (projection join), που αργότερα έγινε γνωστή και ως πέμπτη κανονική μορφή (5KM).
- Μερικές σχέσεις που είναι σε KM-BC είναι επίσης σε 4KM, και μερικές σχέσεις που είναι σε 4KM είναι επίσης σε 5KM.

## Πρώτη Κανονική Μορφή 1KM

---

- Μία σχέση βρίσκεται σε πρώτη κανονική μορφή αν ικανοποιεί όλες τις 6 ιδιότητες του ορισμού της σχέσης
- Εάν υπάρχει καθορισμένο **κλειδί** για τη σχέση τότε ικανοποιείται η απαίτηση της μοναδικότητας των συστοιχιών (σειρών).
- Ένας πίνακας σε πρώτη κανονική μορφή λέγεται κανονικοποιημένος πίνακας και τότε και μόνο τότε αντιστοιχεί σε μία σχέση (οι σχέσεις του σχεσιακού μοντέλου είναι στην 1η κανονική μορφή).

## Δεύτερη κανονική μορφή (2NF)

---

- Μία σχέση βρίσκεται σε δεύτερη κανονική μορφή εάν κάθε ένα από τα γνωρίσματά της που δεν είναι κλειδιά εξαρτώνται συναρτησιακά από ολόκληρο το πρωτεύων κλειδί και όχι μόνο από ένα τμήμα του.
- Οι σχέσεις που έχουν μόνο **ένα γνώρισμα σαν πρωτεύων κλειδί** βρίσκονται αυτόματα και στη **δεύτερη κανονική μορφή**.
- Αυτός είναι ένας λόγος για τον οποίο χρησιμοποιούμε συχνά τεχνητά αναγνωριστικά σαν κλειδιά.

## Τρίτη κανονική μορφή (3NF)

---

- Μία σχέση βρίσκεται σε Τρίτη κανονική μορφή εάν είναι σε δεύτερη και δεν περιέχει **μεταβατικές εξαρτήσεις**.
- Θεωρήστε για παράδειγμα τη σχέση R που έχει γνωρίσματα τα A, B και Γ. Εάν  $A \rightarrow B$  και  $B \rightarrow \Gamma$  τότε θα ισχύει και  $A \rightarrow \Gamma$ .

## Κανονικές μορφές

---

Για τις κανονικές μορφές ισχύουν οι παρακάτω κανόνες:

- Δεν πρέπει να υπάρχουν επαναλαμβανόμενα ονόματα πεδίων.
- Όσα πεδία δεν είναι κλειδιά θα πρέπει να εξαρτώνται συναρτησιακά από ολόκληρο το πρωτεύον κλειδί και όχι από μέρος του.
- Δεν πρέπει να υπάρχει συναρτησιακή μεταβατική εξάρτηση.

$$\begin{array}{ccc} A \rightarrow B & \longrightarrow & A \rightarrow C \\ B \rightarrow C & & \end{array}$$



## Κανονικές μορφές

1KM

ΑΕΜ	ΕΠΙΘΕΤΟ	<del>ΤΗΛ</del>	<del>ΤΗΛ</del>



Τα ονόματα των πεδίων δεν είναι μοναδικά.

2KM

ΚΜ	ΑΕΜ	<del>ΟΝΟΜΑ</del>	<del>ΤΗΛ</del>



Το ΟΝΟΜΑ και το ΤΗΛ εξαρτώνται από το ΑΕΜ.

3KM

ΑΕΜ	ΕΠΙΘΕΤΟ	<del>ΟΝΟΜΑ</del>	<del>ΤΗΛ</del>
		<del>ΠΑΤΡΟΣ</del>	<del>ΠΑΤΡΟΣ</del>



Μεταβατική Εξάρτηση



## Από-κανονικοποίηση

1KM

ΑΕΜ	ΕΠΙΘΕΤΟ	ΤΗΛ1	ΤΗΛ2



Δεν τηρεί τους κανόνες κανονικοποίησης.

Υπάρχουν περιπτώσεις που εάν δεν ακολουθήσουμε τους κανόνες κανονικοποίησης επιτυγχάνουμε καλύτερα αποτελέσματα.

Αυτό καλείται Από-Κανονικοποίηση (De-normalization)



## Ενότητα3 – Εργαστηριακές ασκήσεις του μαθήματος «Βάσεις Δεδομένων»

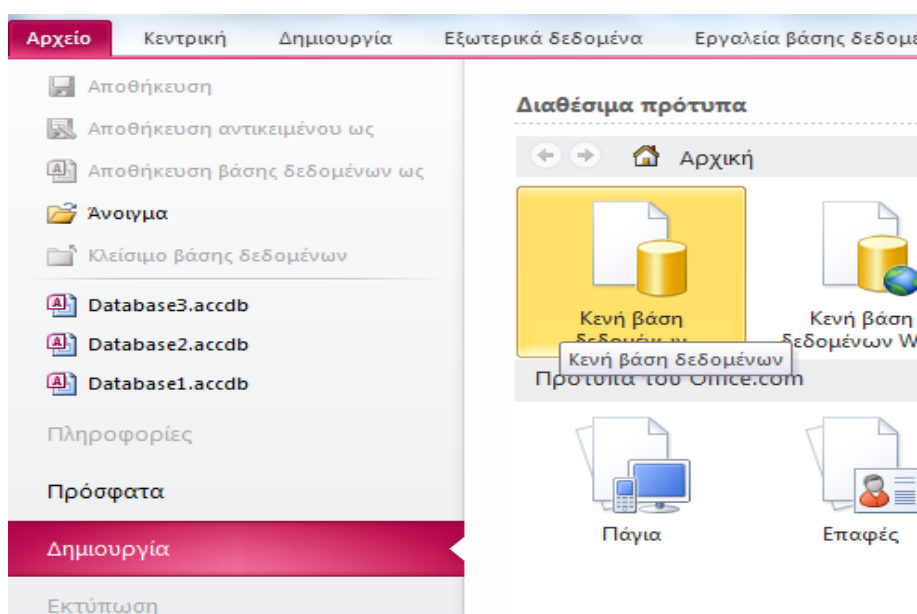
### Άσκηση 1- Φοιτητές, ΒΔ σε Access

Σκοπός αυτής της άσκησης είναι η εισαγωγή σας στο περιβάλλον της Microsoft Access και η δημιουργία του πρώτου σας πίνακα. Πληρέστερη παρουσίαση των δυνατοτήτων της Access θα βρείτε στο εγχειρίδιο το οποίο επισυνάπτεται μαζί τις ασκήσεις του εργαστηρίου.

Υπάρχουν δύο μέθοδοι δημιουργίας πίνακα (στην Access, MySQL, SQL Server, Oracle κλπ). Μέσω ενός γραφικού περιβάλλοντος το οποίο είναι γνωστό και ως QBE (Query by Example) ή χρησιμοποιώντας κώδικα SQL. Η πρώτη μέθοδος είναι πιο εύχρηστη αλλά η δεύτερη είναι πιο πρακτική επειδή θα γράψετε μόνο μία φορά τον κώδικα SQL και θα μπορείτε να τον εκτελέσετε ξαναδημιουργώντας τη βάση όσες φορές θέλετε. Σε αυτήν την άσκηση θα εργαστείτε με την πρώτη μέθοδο.

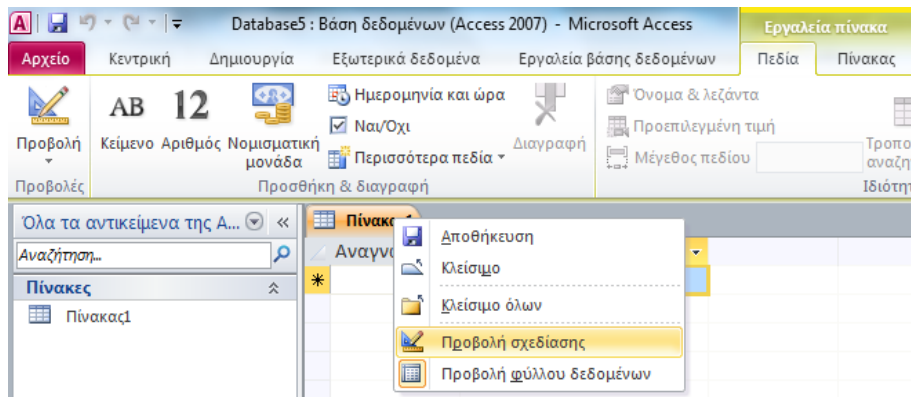
Αφού ανοίξετε την Microsoft Access 2010 για την δημιουργία του πίνακα **ΦΟΙΤΗΤΕΣ** πρέπει να κάνετε τα εξής:

- Δημιουργήστε μια κενή βάση πατώντας διπλό κλικ στο εικονίδιο **Κενή Βάση Δεδομένων**.



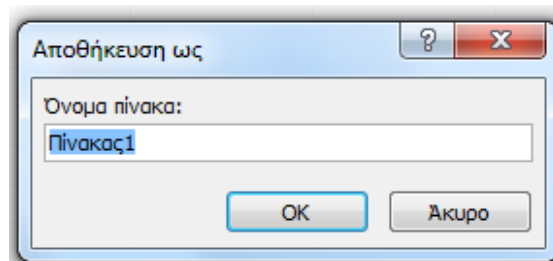
Εικόνα 2

- Βλέπετε ένα περιβάλλον όμοιο με την Εικόνα 2, πατήστε δεξί κλικ στον πίνακα → **Προβολή σχεδίασης**.



Εικόνα 3

- Πληκτρολογήστε το όνομα του πίνακα (ΦΟΙΤΗΤΕΣ).



Εικόνα 4

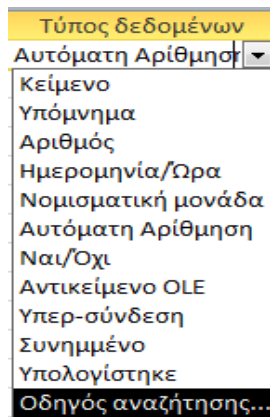
- Εισάγετε τα πεδία στον πίνακα σας ομοίως όπως στην Εικόνα 4 καθώς και τους αντίστοιχους τύπους δεδομένων αυτών των πεδίων.
- **ΠΡΟΣΟΧΗ!** Υπάρχουν κάποιες συμβάσεις για τα ονόματα των αντικειμένων των βάσεων δεδομένων και αυτά είναι τα εξής: Το όνομα κάθε αντικειμένου μπορεί να έχει μέχρι 64 χαρακτήρες. Δεν μπορεί να εμπεριέχει σύμβολα όπως, ; ( ) [ ] { }.' ή να αρχίζει με κενό. Επίσης για λόγους συμβατότητας, κατά την ονομασία: καλό είναι να αποφεύγετε τα σύμβολα πράξεων : + - \* / \ % ^, να ξεκινάτε ένα όνομα πάντα με χαρακτήρα και όχι με αριθμό τέλος να αποφεύγετε τα ονόματα και συνώνυμα των τύπων δεδομένων π.χ. Time, Date.

ΦΟΙΤΗΤΕΣ	
Όνομα πεδίου	Τύπος δεδομένων
ΑΕΜ	Αυτόματη Αρίθμηση
ΟΝΟΜΑ	Κείμενο
ΕΠΙΘΕΤΟ	Κείμενο
ΟΝΟΜΑ_ΠΑΤΡΟΣ	Κείμενο
ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ	Ημερομηνία/Ωρα
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	Ημερομηνία/Ωρα
ΠΟΛΗ	Κείμενο
ΔΙΕΥΘΥΝΣΗ	Κείμενο
ΤΗΛΕΦΩΝΟ	Αριθμός
ΣΧΟΛΙΑ	Υπόμνημα
E_MAIL	Υπερ-σύνδεση
ΦΩΤΟΓΡΑΦΙΑ	Αντικείμενο OLE

Εικόνα 5

Για κάθε πεδίο επιλέξτε τον σωστό τύπο δεδομένων ανάλογα με το τί θέλετε να εμπεριέχει. Οι τύποι δεδομένων φαίνονται στην Εικόνα 5. Ποιο αναλυτικά:

- Αυτόματη αρίθμηση → Ο τύπος δεδομένων του πρωτεύον κλειδιού.
- Κείμενο → Εισαγωγή κειμένου έως 255 χαρακτήρες.
- Υπόμνημα → Κείμενο μέχρι 64Kb.
- Αριθμός → Ακέραιος ή πραγματικός αριθμός διπλής ακρίβειας.
- Ημερομηνία/ώρα → Τιμές "Ημερομηνία" και "Ωρα" για τις χρονιές 100 μέχρι 9999.
- Νομισματική μονάδα → Νομισματικές τιμές.
- Ναι/Όχι → Λογικό διαβάζει μηδέν και ένα.
- Αντικείμενο OLE → Για εισαγωγή φωτογραφίας.
- Υπερ-σύνδεση → Για εισαγωγή κάποιου συνδέσμου email, είτε κάποιας σελίδας.
- Συνημμένο → Συνημμένες εικόνες, αρχεία υπολογιστικών φύλλων, έγγραφα, γραφήματα και άλλοι τύποι υποστηριζόμενων αρχείων στις εγγραφές της βάσης δεδομένων σας, παρομοίως με την επισύναψη αρχείων σε μηνύματα ηλεκτρονικού ταχυδρομείου.
- Υπολογίστηκε → Αποτελέσματα ενός υπολογισμού. Ο υπολογισμός πρέπει να αναφέρεται σε άλλα πεδία του ίδιου πίνακα. Χρησιμοποιήστε τη "Δόμηση παραστάσεων" για τη δημιουργία του υπολογισμού.
- Οδηγός Αναζήτησης → Χρησιμοποιείται κυρίως για συσχέτιση πινάκων παρακάτω θα εξεταστεί αναλυτικότερα.



Εικόνα 6

- Στη συνέχεια δεξί κλικ στον πίνακα ΦΟΙΤΗΤΕΣ → **Προβολή Φύλλου Δεδομένων**.
- Βλέπετε τα πεδία του πίνακα σας άδεια στην προβολή φύλλου δεδομένων έχετε την δυνατότητα να τα συμπληρώσετε.

ΑΕΜ	ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ	ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ	ΠΟΛΗ	ΔΙΕΥΘΥΝΣΗ
1	MARIA	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΧΡΗΣΤΟΣ	1/9/2009	1/1/1985	ΣΕΡΡΕΣ	ΣΤΑΜΟΥΛΗ 19
2	ΚΩΣΤΑΣ	ΣΤΑΜΟΣ	ΓΙΑΝΝΗΣ	2/9/2009	10/8/1989	ΘΕΣΣΑΛΟΝΙΚΗ	ΙΟΥΣΤΙΝΙΑΝΟΥ 3
3	ΕΛΕΝΗ	ΓΕΩΡΓΙΟΥ	ΑΛΕΞΙΟΣ	1/2/2009	9/3/1989	ΑΘΗΝΑ	ΡΑΙΔΕΣΤΟΥ 16
*	(Νέο)						

Εικόνα 7

- Εισάγετε στον πίνακα τουλάχιστον τρεις εγγραφές που επιθυμείτε. Στο τελευταίο πεδίο εισάγεται μία εικόνα της αρεσκείας σας με αντιγραφή – επικόλληση ή δεξί κλικ εισαγωγή αντικειμένου.


ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ	ΠΟΛΗ	ΔΙΕΥΘΥΝΣΗ	ΤΗΛΕΦΩΝΟ	ΣΧΟΛΙΑ	Ε_MAIL	ΦΩΤΟΓΡΑΦΙΑ
1/1/1985	ΣΕΡΡΕΣ	ΣΤΑΜΟΥΛΗ 19	698990122	ΚΑΛΟΣ	<a href="mailto:papad.maria@gmail">papad.maria@gmail</a>	Package
10/8/1989	ΘΕΣΣΑΛΟΝΙΚΗ	ΙΟΥΣΤΙΝΙΑΝΟΥ 3	698098657	ΑΡΙΣΤΟΣ	<a href="mailto:stam.gian@gmail.co">stam.gian@gmail.co</a>	Bitmap Image
9/3/1989	ΑΘΗΝΑ	ΡΑΙΔΕΣΤΟΥ 16	695690986	ΚΑΛΟΣ	<a href="mailto:geo.alex@gmail.com">geo.alex@gmail.com</a>	Bitmap Image
*						

Εικόνα 8

## Δημιουργία φόρμας του πίνακα ΦΟΙΤΗΤΕΣ

Η φόρμα είναι ένα εύχρηστο κομμάτι της Access δίνει την δυνατότητα στον χρήστη να εμφανίσει τα δεδομένα μορφοποιημένα ώστε να μπορεί εύκολα να τα διαχειρίζεται και να τα ενημερώνει. Η διαδικασία δημιουργίας μιας φόρμας έχει ως εξής:

Ενώ βρίσκεστε στον πίνακα φοιτητές σε προβολή φύλλου δεδομένων επιλέξτε από την κορδέλα **Δημιουργία** → **Φόρμα** και δώστε ένα όνομα.

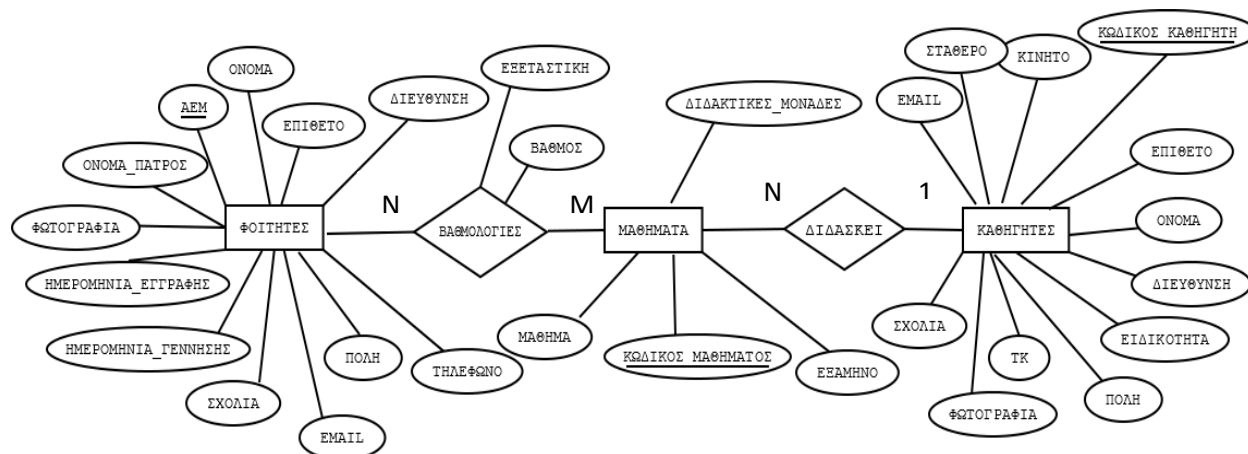
ΦΟΙΤΗΤΕΣ			
ΑΕΜ	1	ΠΟΛΗ	ΣΕΡΡΕΣ
ΟΝΟΜΑ	ΜΑΡΙΑ	ΔΙΕΥΘΥΝΣΗ	ΣΤΑΜΟΥΛΗ 19
ΕΠΙΘΕΤΟ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΤΗΛΕΦΩΝΟ	698990122
ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΧΡΗΣΤΟΣ	ΣΧΟΛΙΑ	ΚΑΛΟΣ
ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ	1/9/2009	E_MAIL	<a href="mailto:papad.maria@gmail.com">papad.maria@gmail.com</a>
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	1/1/1985	ΦΩΤΟΓΡΑΦΙΑ	 Chrysanthemum.jpg

Εικόνα 9

## Άσκηση 2 - Φοιτητές - Μαθήματα - Καθηγητές σε Access

Σκοπός αυτής της άσκησης είναι η εκμάθηση και εμπέδωση της δημιουργίας πινάκων, συσχετίσεων, της ενεργοποίησης περιορισμού ακεραιότητας αναφορών και εισαγωγής δεδομένων μέσα από το γραφικό περιβάλλον της Microsoft Access.

**Μοντέλο οντοτήτων συσχετίσεων:** Τετράγωνα→Οντότητες, Ρόμβοι→Συσχετίσεις



Σχήμα 1

Στο Σχήμα 1 φαίνεται η συσχέτιση των φοιτητών με τα μαθήματα που είναι πολλά προς πολλά ενώ των μαθημάτων και των καθηγητών είναι πολλά προς ένα. Έτσι θα δημιουργήσετε τέσσερις πίνακες για αυτή τη βάση δεδομένων όπου θα καταχωρείτε πληροφορίες για τους φοιτητές, τα μαθήματα τους καθηγητές και τις βαθμολογίες.

Ο τρόπος κατά τον οποίο προκύπτουν αυτές οι συσχετίσεις είναι απλός. Έστω σε μία γενικότερη περίπτωση φοιτητές, μαθήματα και βαθμολογίες. Για να βρείτε τις συσχετίσεις κάνετε τις εξής δύο ερωτήσεις:

- 1) Ένας φοιτητής συσχετίζεται (βαθμολογείται) με πολλά μαθήματα; (στην περίπτωση μας ΝΑΙ ένας φοιτητής συσχετίζεται με πολλά μαθήματα).
- 2) Ένα μάθημα συσχετίζεται με πολλούς φοιτητές; (στην περίπτωση μας ΝΑΙ ένα μάθημα συσχετίζεται με πολλούς φοιτητές).

Μπορούν να προκύψουν οι εξής συνδυασμοί :

- πολλά προς πολλά δηλαδή N:M
- ένα προς πολλά δηλαδή 1:N
- πολλά προς ένα δηλαδή N:1
- ένα προς ένα δηλαδή 1:1

Όταν η συσχέτιση είναι πολλά προς πολλά τότε δημιουργείται νέος πίνακας ενώ στην περίπτωση ένα προς πολλά δημιουργείται ένα ξένο κλειδί στην πλευρά του N.

## Δημιουργία του πίνακα φοιτητές

Ανοίξτε την Microsoft Access, και ανατρέξτε στην άσκηση 1. Σε περίπτωση που για κάποιο λόγο έχει χαθεί ξαναδημιουργείστε τον πίνακα φοιτητές σε προβολή σχεδίασης σύμφωνα με της οδηγίες της άσκησης 1 με τα εξής πεδία:

ΦΟΙΤΗΤΕΣ(ΑΕΜ,ΟΝΟΜΑ,ΕΠΙΘΕΤΟ,ΟΝΟΜΑ\_ΠΑΤΡΟΣ,ΗΜΕΡΟΜΗΝΙΑ\_ΕΓΓΡΑΦΗΣ, ΗΜΕΡΟΜΗΝΙΑ\_ΓΕΝΝΗΣΗΣ, ΠΟΛΗ, ΔΙΕΥΘΥΝΣΗ, ΤΗΛΕΦΩΝΟ, ΣΧΟΛΙΑ EMAIL, ΦΩΤΟΓΡΑΦΙΑ).



Όνομα πεδίου	Τύπος δεδομένων
<b>ΑΕΜ</b>	Αυτόματη Αρίθμηση
ΟΝΟΜΑ	Κείμενο
ΕΠΙΘΕΤΟ	Κείμενο
ΟΝΟΜΑ_ΠΑΤΡΟΣ	Κείμενο
ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ	Ημερομηνία/Ωρα
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	Ημερομηνία/Ωρα
ΠΟΛΗ	Κείμενο
ΔΙΕΥΘΥΝΣΗ	Κείμενο
ΤΗΛΕΦΩΝΟ	Αριθμός
ΣΧΟΛΙΑ	Υπόμνημα
E_MAIL	Υπερ-σύνδεση
ΦΩΤΟΓΡΑΦΙΑ	Αντικείμενο OLE

Εικόνα 10

(Εισάγετε τουλάχιστον τρεις εγγραφές)

Για κάθε πίνακα απαραίτητο είναι να ορίσουμε το πρωτεύον κλειδί. Ορίστε λοιπόν ως πρωτεύον κλειδί το πεδίο **ΑΕΜ** (επιλέγουμε το πεδίο **ΑΕΜ** και έπειτα δεξί κλικ→πρωτεύον κλειδί ) και αυτόματα δημιουργείται και ένα ευρετήριο (index).

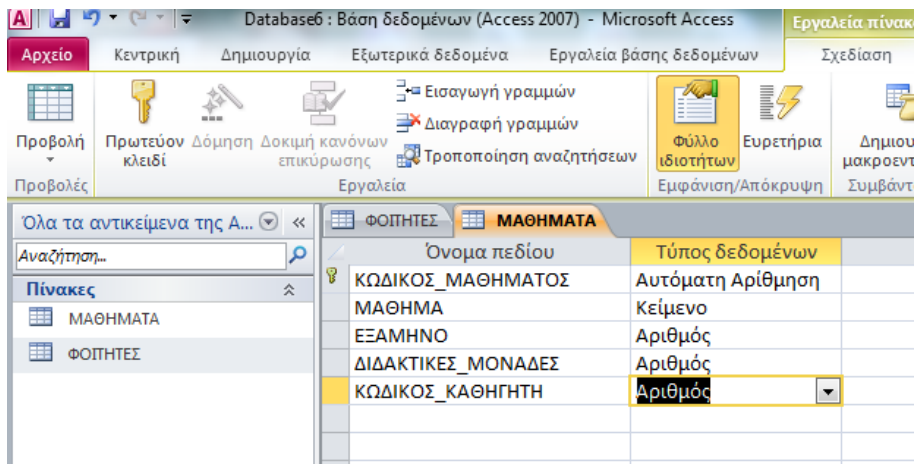
## Δημιουργία του πίνακα μαθήματα

Με όμοιο τρόπο όπως δημιουργήσατε τον πίνακα φοιτητές θα εργαστείτε και για την δημιουργία του πίνακα μαθήματα ο οποίος θα εμπεριέχει όμως τα εξής πεδία και τους αντίστοιχους τύπους δεδομένων όπως ορίζονται στη Εικόνα 1:

ΜΑΘΗΜΑΤΑ(ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ, ΜΑΘΗΜΑ, ΕΞΑΜΗΝΟ, ΔΙΔΑΚΤΙΚΕΣ\_ΜΟΝΑΔΕΣ, ΚΩΔΙΚΟΣ\_ΚΑΘΗΓΗΤΗ). Ο ΚΩΔΙΚΟΣ\_ΚΑΘΗΓΗΤΗ είναι **Ξένο κλειδί**.

(Εισάγετε τουλάχιστον τρεις εγγραφές).





Εικόνα 11

- Οι εγγραφές που εμπεριέχονται είναι οι εξής:

ΚΩΔΙΚΟΣ_Μ	ΜΑΘΗΜΑ	ΕΞΑΜΗΝΟ	ΔΙΔΑΚΤΙΚΕΣ	ΚΩΔΙΚΟΣ_Κ	Κ
3	ΦΥΣΙΚΗ	3	6	ΠΑΠΑΔΟΠΟΥΛ	
4	ΜΑΘΗΜΑΤΙΚΑ	5	5	ΠΑΠΑΣ	
5	ΑΓΓΛΙΚΑ	1	3	ΔΗΜΗΤΡΙΟΥ	
6	ΒΑΣΕΙΣ_ΔΕΔΟΜΕΝΩΝ	4	7	ΔΗΜΗΤΡΙΟΥ	
7	ΣΤΑΤΙΣΤΙΚΗ	5	7	ΠΑΠΑΔΟΠΟΥΛ	
8	ΒΙΟΜΗΧΑΝΙΚΗ	5	4	ΔΗΜΗΤΡΙΟΥ	

Εικόνα 12

## Δημιουργία του πίνακα καθηγητές

Ομοίως για τον πίνακα καθηγητές με πεδία: ΚΩΔΙΚΟΣ ΚΑΘΗΓΗΤΗ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΔΙΕΥΘΥΝΣΗ, ΕΙΔΙΚΟΤΗΤΑ, ΠΟΛΗ, ΤΚ, ΚΙΝΗΤΟ, ΣΤΑΘΕΡΟ, Ε\_MAIL, ΦΩΤΟΓΡΑΦΙΑ, ΣΧΟΛΙΑ.

(Εισάγεται τουλάχιστον τρεις εγγραφές).

Εικόνα 13

- Οι εγγραφές που εμπεριέχονται είναι οι εξής:

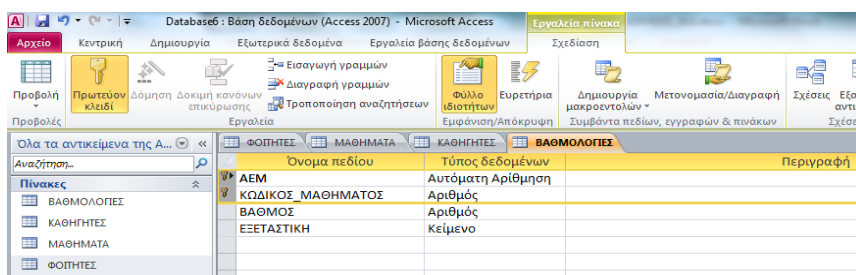
ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ	ΕΙΔΙΚΟΤΗΤΑ	ΠΟΛΗ	ΤΚ	ΚΙΝΗΤΟ	ΣΤΑΘΕΡΟ	E_MAIL	ΦΩΤΟΓΡΑΦ	ΣΧΟΛΙΑ
ΠΑΠΑΣ	ΗΛΙΑΣ	Δ.ΘΡΑΚΗΣ 1	ΜΑΘΗΜΑΤΙΚΟΣ	ΣΕΡΡΕΣ	65789	687266268	232107788	kastanh@gmail.com	Bitmap Image	
ΠΑΠΑΔΟΠΟΥΛΟΣ	ΠΑΝΝΗΣ	ΗΡΩΔΟΤΟΥ 3	ΦΥΣΙΚΟΣ	ΑΘΗΝΑ	78973	678976278	758373938	kapoukranhs@gmail.co	Bitmap Image	
ΔΗΜΗΤΡΙΟΥ	ΕΥΑΓΓΕΛΙΑ	ΚΑΡΑΪΣΚΑΚΗ 1	ΧΗΜΙΚΟΣ	ΘΕΣΣΑΛΟΝΙΚΗ	98987	698025780	838388389	soyltanh@gmail.com	Bitmap Image	

Εικόνα 14

## Δημιουργία του πίνακα βαθμολογίες

Ο πίνακας βαθμολογίες είναι ο πίνακας που δημιουργήθηκε για την ένωση των πινάκων ΦΟΙΤΗΤΕΣ και ΜΑΘΗΜΑΤΑ είναι στην ουσία ο πίνακας συσχέτισης τους ο οποίος έχει σαν πεδία τα πρωτεύον κλειδιά των άλλων δυο και δημιουργήθηκαν σε αυτόν δύο ξένα κλειδιά και σε συνδυασμό αυτών των δύο ένα πρωτεύον κλειδί. Τα πεδία που εμπεριέχει είναι τα εξής: ΑΕΜ, ΚΩΔΙΚΟΣ ΜΑΘΗΜΑΤΟΣ, ΒΑΘΜΟΣ, ΕΞΕΤΑΣΤΙΚΗ.

(Εισάγεται τουλάχιστον δέκα εγγραφές αφού πρώτα κάνετε τις συσχετίσεις στη συνέχεια).



Εικόνα 15

**ΠΡΟΣΟΧΗ!** Το πρωτεύον κλειδί του πίνακα ΒΑΘΜΟΛΟΓΙΕΣ δημιουργείται με την επιλογή των δύο πεδίων που είναι ξένα κλειδιά αλλά με δεξί κλικ στον συνδυασμό τους γίνεται πρωτεύον κλειδί.

- Οι εγγραφές που εμπεριέχονται είναι οι εξής:

ΔΕΜ	ΕΞΕΤΑΣΤΙΚΗ	ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ	ΒΑΘΜΟΣ
ΠΑΠΑΔΟΠΟΥΛΟΣ	ΕΑΡ2008-2009	ΦΥΣΙΚΗ	5
ΠΑΠΑΔΟΠΟΥΛΟΥ	ΧΕΙΜ2009-2010	ΜΑΘΗΜΑΤΙΚΑ	5
ΠΑΠΑΔΟΠΟΥΛΟΥ	ΧΕΙΜ2010-2011	ΑΓΓΛΙΚΑ	1
ΠΑΠΑΔΟΠΟΥΛΟΥ	ΧΕΙΜ2010-2011	ΒΑΣΕΙΣ_ΔΕΔΟΜΕΝΩΝ	4
ΠΑΠΑΔΟΠΟΥΛΟΥ	ΕΑΡ2008-2009	ΣΤΑΤΙΣΤΙΚΗ	5
ΣΤΑΜΟΣ	ΧΕΙΜ2010-2011	ΦΥΣΙΚΗ	7
ΣΤΑΜΟΣ	ΧΕΙΜ2009-2010	ΜΑΘΗΜΑΤΙΚΑ	3
ΣΤΑΜΟΣ	ΧΕΙΜ2011-2012	ΑΓΓΛΙΚΑ	8
ΣΤΑΜΟΣ	ΧΕΙΜ2010-2011	ΒΑΣΕΙΣ_ΔΕΔΟΜΕΝΩΝ	6
ΣΤΑΜΟΣ	ΕΑΡ2008-2009	ΣΤΑΤΙΣΤΙΚΗ	10
ΓΕΩΡΓΙΟΥ	ΕΑΡ2008-2009	ΦΥΣΙΚΗ	6
ΓΕΩΡΓΙΟΥ	ΕΑΡ2009-2010	ΜΑΘΗΜΑΤΙΚΑ	7
ΓΕΩΡΓΙΟΥ	ΕΑΡ2008-2010	ΑΓΓΛΙΚΑ	6
ΓΕΩΡΓΙΟΥ	ΕΑΡ2009-2010	ΒΑΣΕΙΣ_ΔΕΔΟΜΕΝΩΝ	5
ΓΕΩΡΓΙΟΥ	ΧΕΙΜ2009-2010	ΣΤΑΤΙΣΤΙΚΗ	3

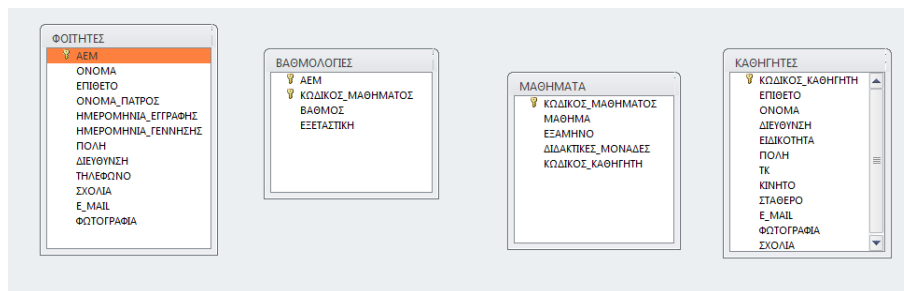
Εικόνα 16

## Τρόποι συσχετίσεων πινάκων

Υπάρχουν δύο αποδεκτοί τρόποι για την δημιουργία συσχετίσεων μεταξύ των πινάκων.

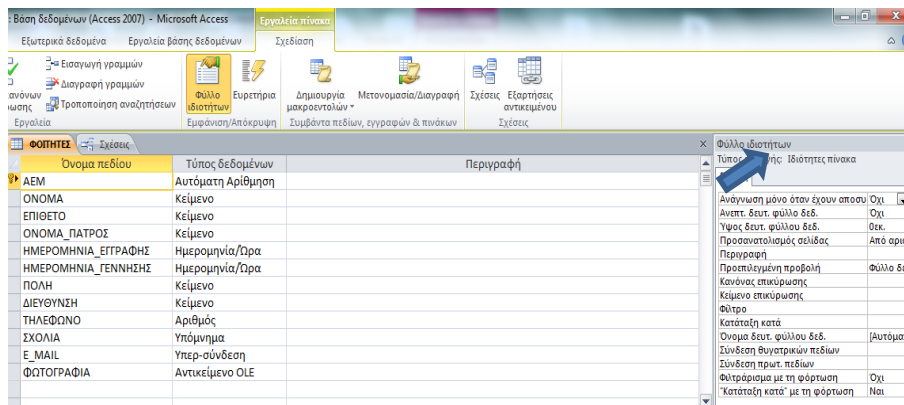
### Α τρόπος :

Ενώ βρίσκεστε σε προβολή σχεδίασης σε έναν από τους τέσσερις πίνακες που δημιουργήσατε πηγαίνετε επάνω στα **Εργαλεία Βάσης Δεδομένων** και επιλέγετε την καρτέλα **Σχέσεις**.



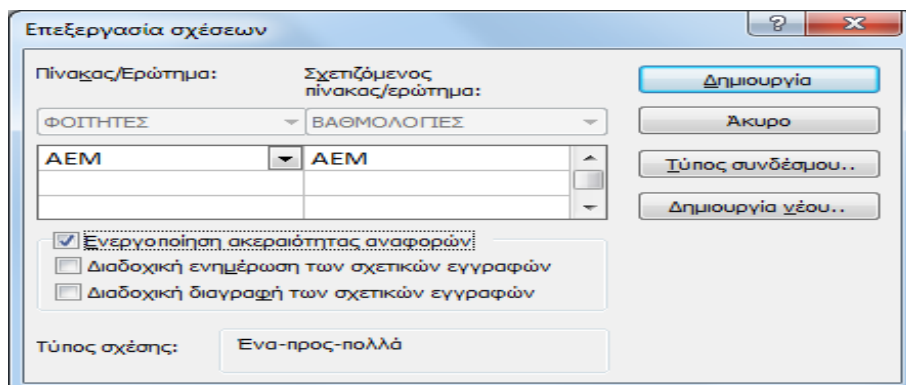
Εικόνα 17

Για να συνδέσετε τους πίνακες μεταξύ τους όμως πρέπει να κλείσετε του πίνακες που έχετε ανοιχτούς πρώτα.



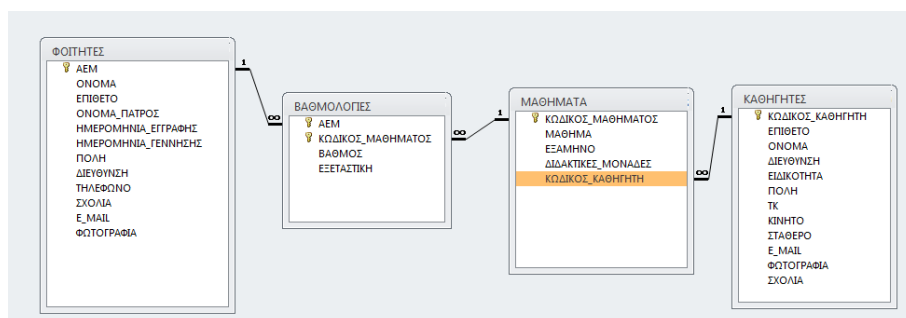
Εικόνα 18

Η σύνδεση στη συνέχεια γίνεται εύκολα απλά συνδέοντας το ξένο με το αντίστοιχο πρωτεύων κλειδί με το σύρσιμο του κέρσορα. Μόλις το κάνετε αυτό εμφανίζει ένα νέο παράθυρο τα οποίο σας δείχνει τους πίνακες και το πεδίο που τους συνδέει  
 Εικόνα 17. Τέλος τσεκάρετε την **Ενεργοποίηση ακεραιότητας αναφορών** και **Δημιουργία**.



Εικόνα 19

Ομοίως γίνεται και δημιουργία σχέσεων των άλλων πινάκων όπως βλέπετε στην Εικόνα 19.



Εικόνα 20


Τέλος **Αποθηκεύετε** τη σχέση σας ζητάει αυτόματα όνομα με την έξοδο σας από το σχεσιακό διάγραμμα.

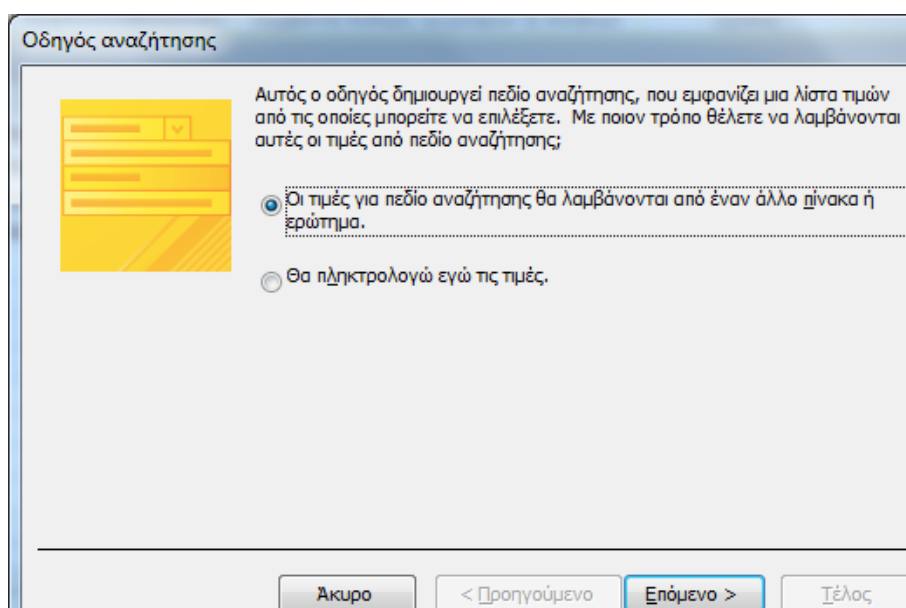
Ο δεύτερος τρόπος ο οποίος είναι ποίο περίπλοκος αλλά παρόλα αυτά είναι ποιο αποδεκτός είναι ο εξής:

**ΠΡΟΣΟΧΗ!** Πρώτα πρέπει να γίνουν οι συνδέσεις και έπειτα οι καταχωρήσεις.

**ΣΗΜΕΙΩΣΗ:** Ο συμβολισμός  σημαίνει (1:N) ένα προς πολλά.

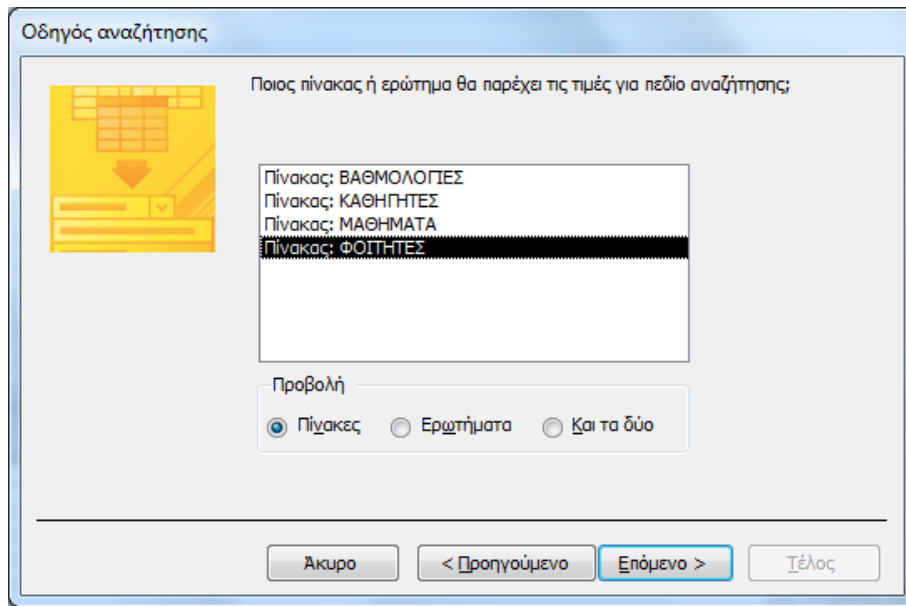
## Β τρόπος

Μεταβείτε στον πίνακα ΒΑΘΜΟΛΟΓΙΕΣ σε προβολή σχεδίασης  όπου φαίνονται τα ξένα κλειδιά, στη συνέχεια αλλάζουμε τον τύπο δεδομένων του πεδίου **ΑΕΜ** από αριθμό σε οδηγό αναζήτησης. Με την εναλλαγή αυτή εμφανίζεται το παράθυρο της Εικόνας 19. Επιλέξτε **Οι τιμές για πεδίο αναζήτησης θα λαμβάνονται από έναν άλλο πίνακα ή ερώτημα** και στη συνέχεια πιέστε **Επόμενο**.



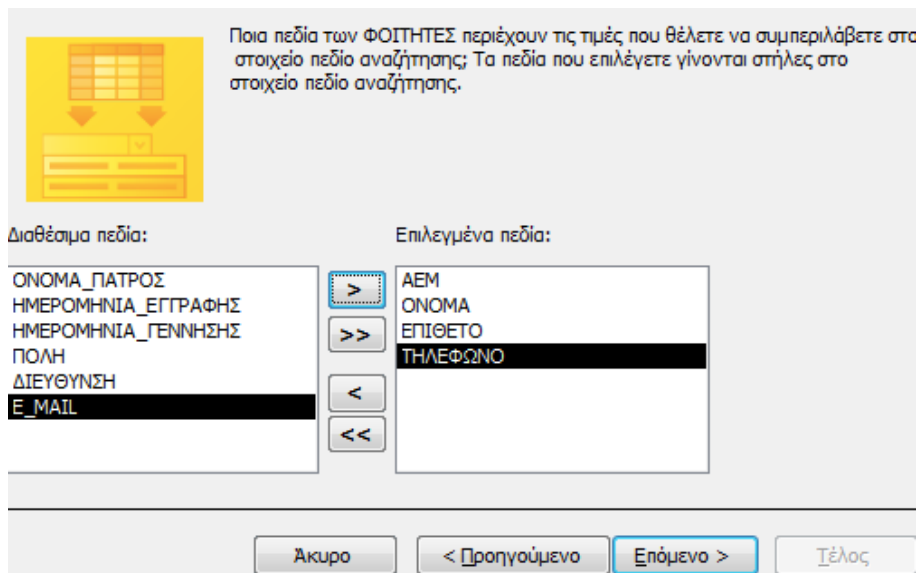
Εικόνα 21

Στην Εικόνα 20 εμφανίζεται ένα νέο παράθυρο το οποίο σας δίνει τη δυνατότητα να επιλέξετε τον πίνακα το ερώτημα ή και τα δύο που θα εμπεριέχουν τις τιμές για το πεδίο αναζήτησης. Στην άσκηση σας θα χρειαστείτε τον πίνακα **ΦΟΙΤΗΤΕΣ** αυτός εμπεριέχει το πρωτεύον κλειδί με τις τιμές για το πεδίο αναζήτησης.



Εικόνα 22

Παρακάτω στην Εικόνα 21 εμφανίζονται τα διαθέσιμα πεδία του πίνακα ΦΟΙΤΗΤΕΣ για να επιλέξετε ποια από αυτά θα περιλαμβάνονται σαν στοιχεία στο πεδίο αναζήτησης. Για την άσκηση θα χρησιμοποιηθούν το ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛΕΦΩΝΟ. Όποιο από αυτά θα εισαχθεί στο πεδίο αναζήτησης αυτόματα θα αναγνωρίζει το ΑΕΜ. Επειτα πιάστε Επόμενο.



Εικόνα 23

Σας δίνει την δυνατότητα να τοποθετήσετε τα στοιχεία στο πλαίσιο της λίστας κατά αύξουσα ή φθίνουσα σειρά. Στην προκειμένη περίπτωση θα βάλουμε το πεδίο ΕΠΙΘΕΤΟ σε αύξουσα σειρά και πιάστε Επόμενο.

**Οδηγός αναζήτησης**

Ποια σειρά ταξινόμησης θέλετε για τα στοιχεία στο πλαίσιο λίστας;

Μπορείτε να ταξινομήσετε τις εγγραφές έως και κατά τέσσερα πεδία, σε αύξουσα ή φθίνουσα σειρά.

1 ΕΠΙΘΕΤΟ ▾ Αύξουσα

2 ▾ Αύξουσα

3 ▾ Αύξουσα

4 ▾ Αύξουσα

Ακυρο < Προηγούμενο Επόμενο > Τέλος

Εικόνα 24

Τσεκάρετε την απόκρυψη στήλης κλειδιού.

**Οδηγός αναζήτησης**

Ποιο θα είναι το πλάτος των στηλών στο στοιχείο πεδίο αναζήτησης;

Για να προσαρμόσετε το πλάτος των στηλών, σύρετε τη δεξιά τους πλευρά στο επιθυμητό πλάτος ή κάντε διπλό κλικ στη δεξιά πλευρά της επικεφαλίδας της στήλης για βέλτιστη προσαρμογή.

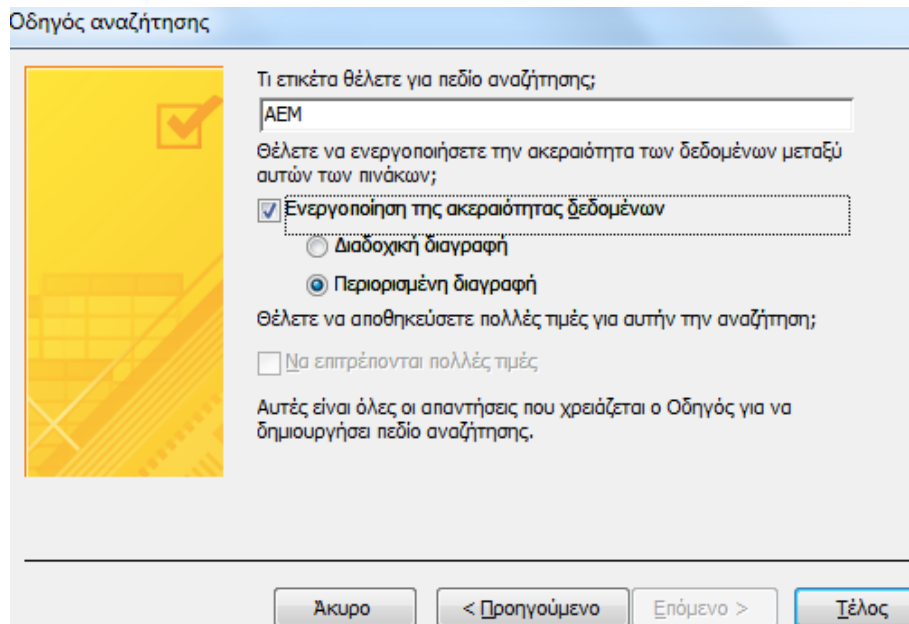
Απόκρυψη στήλης κλειδιού (συνιστάται)

ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΤΗΛΕΦΩΝΟ			
ΓΕΩΡΓΙΟΥ	ΕΛΕΝΗ	695690986			
ΠΑΠΑΔΟΠΥΛΟΥ	ΜΑΡΙΑ	698990122			
ΣΤΑΜΟΣ	ΚΩΣΤΑΣ	698098657			

Ακυρο < Προηγούμενο Επόμενο > Τέλος

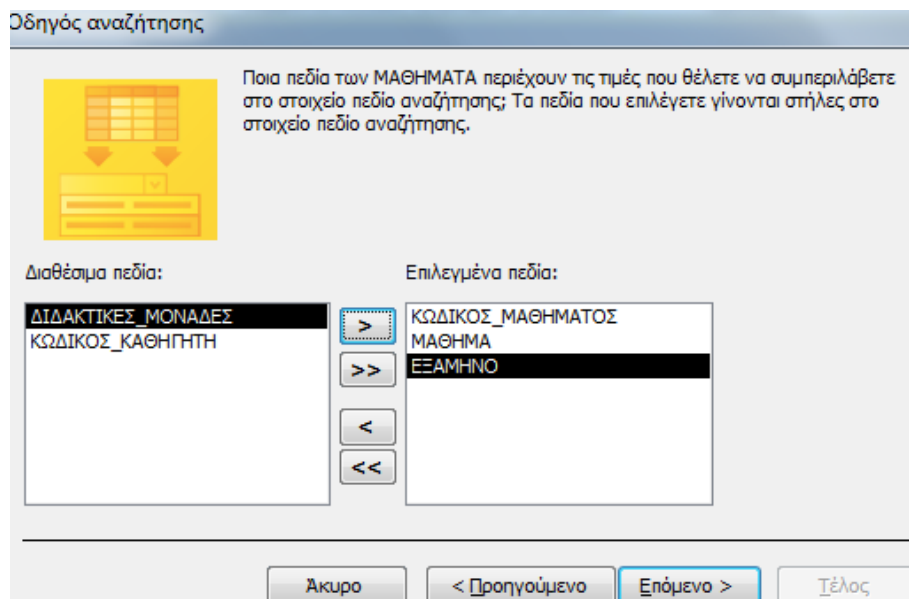
Εικόνα 25

Στην Εικόνα 24 φαίνεται πως θα εισάγετε το όνομα του πεδίου αναζήτησης και **Ενεργοποιείτε την ακεραιότητα των δεδομένων**. Πιέστε Τέλος.



Εικόνα 26

Ομοίως θα εργαστείτε και για το πεδίο **ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ** του πίνακα **ΒΑΘΜΟΛΟΓΙΕΣ** το οποίο θα πρέπει να συνδεθεί με το πεδίο **ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ** του πίνακα **ΜΑΘΗΜΑΤΑ**.



Εικόνα 27



### Άσκηση 3 – Δημιουργία φορμών – Query by example σε Access

Σκοπός αυτής της άσκησης είναι η εξοικείωση σας με την δημιουργία φορμών καθώς και την εισαγωγή στη δημιουργία ερωτημάτων SQL με την μέθοδο Query by Example (QBE).

Η δημιουργία των φορμών όπως είδαμε και σε προηγούμενη άσκηση γίνεται με τα ακόλουθα βήματα: **Μενού→ Δημιουργία→Φόρμα. Δημιουργήστε μία φόρμα για κάθε πίνακα.**

ΚΑΘΗΓΗΤΕΣ

ΚΩΔΙΚΟΣ\_ΚΑΘΗΓΗΤΗ: Neo TK

ΕΠΙΘΕΤΟ: ΚΙΝΗΤΟ

ΟΝΟΜΑ: ΣΤΑΘΕΡΟ

ΔΙΕΥΘΥΝΣΗ: E\_MAIL

ΕΙΔΙΚΟΤΗΤΑ: ΦΩΤΟΓΡΑΦΙΑ

ΠΟΛΗ: ΣΧΟΛΙΑ

Εικόνα 28.

ΜΑΘΗΜΑΤΑ

ΚΑΡΤΕ

Αποθήκευση

Κλείσιμο

Εξέλιξη όλων

Προβολή φόρμας

Προβολή διάρθρωσης

Προβολή σχεδίασης

ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑ

ΜΑΘΗΜΑ

ΕΞΑΜΗΝΟ

ΔΙΔΑΚΤΙΚΕΣ\_ΜΟΝΑΔΕΣ

ΚΩΔΙΚΟΣ\_ΚΑΘΗΓΗΤΗ

ΑΕΜ	ΒΑΘΜΟΣ	ΕΞΕΤΑΣΤΙΚΗ
*		

Εγγραφή: 1 από 1

Εικόνα 29

ΒΑΘΜΟΛΟΓΙΕΣ

Αποθήκευση ως

Όνομα φόρμας: ΚΑΡΤΕΛΑ\_ΒΑΘΜΟΛΟΓΙΕΣ

OK

Άκυρο

ΑΕΜ

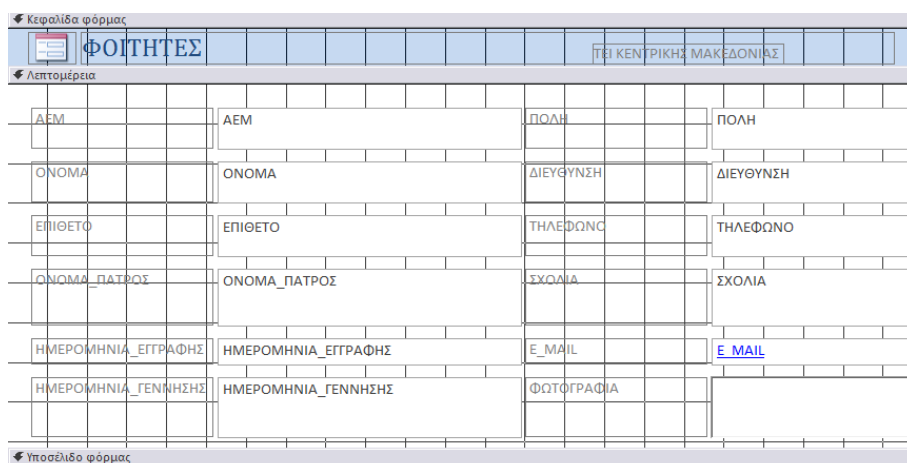
ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ

ΒΑΘΜΟΣ

ΕΞΕΤΑΣΤΙΚΗ

Εικόνα 30

Με την επιλογή της φόρμας και προβολή σχεδίασης μπορείτε να επεξεργαστείτε τα πεδία της, βλέπετε την αντίστοιχη Εικόνα 29.



Εικόνα 31

## Query by example

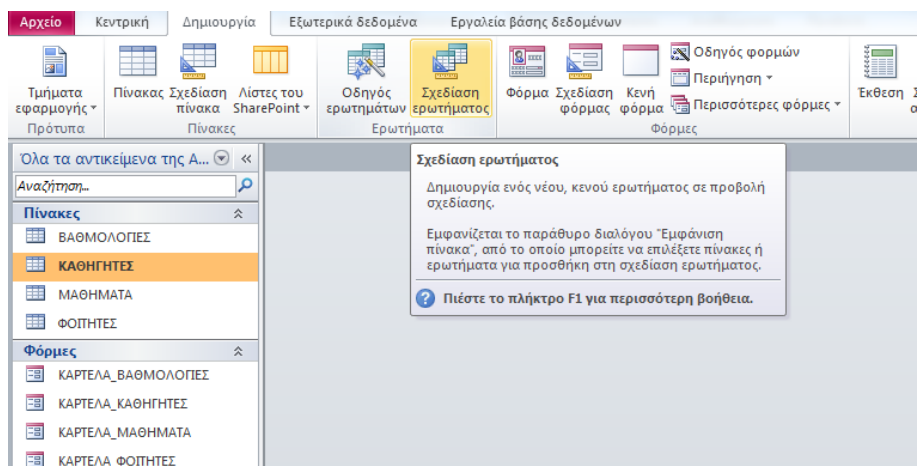
Πριν προχωρήσετε παρακάτω αναφέρουμε τους τελεστές που μπορείτε να χρησιμοποιήσετε στα ερωτήματα.

### ΤΕΛΕΣΤΕΣ:

Αριθμητικοί τελεστές	+, -, *, /, MOD
Λογικοί τελεστές	AND, OR, NOT
Συγκριτικοί Τελεστές	<, >, >=, <=, <>, =, IN, NOT IN, LIKE NOT LIKE, BETWEEN, (... AND ....)

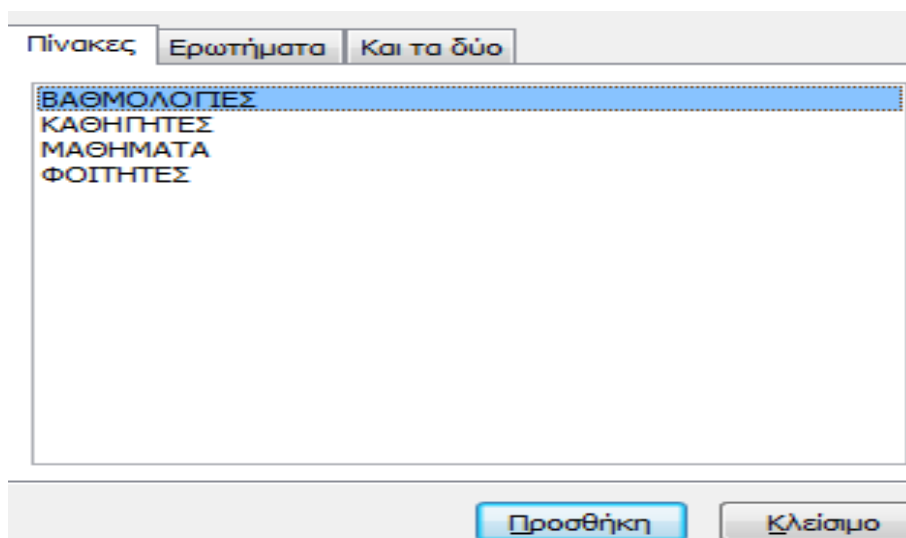
Ακολουθούν ερωτήματα με την μέθοδο (QBE):

Για την δημιουργία ενός ερωτήματος πηγαίνετε στην κορδέλα, έπειτα **Δημιουργία** → **Σχεδίαση ερωτήματος**



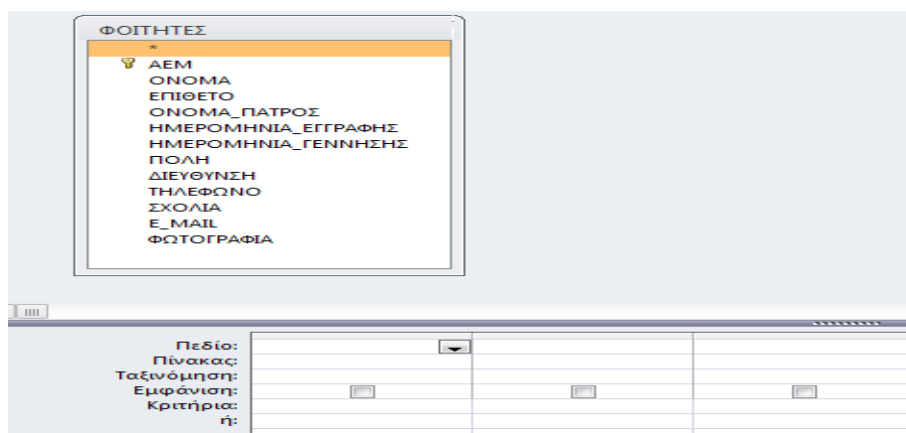
Εικόνα 32

Με το που επιλέξετε την σχεδίαση ερωτήματος εμφανίζεται το παράθυρο της Εικόνας 31 και από εκεί επιλέγετε ποιόν πίνακα αφορά το ερώτημά σας.



Εικόνα 33

Αφού επιλέξετε πίνακα δημιουργείτε τα ερωτήματα στο κάτω μέρος της εικόνας παρακάτω υπάρχουν αναλυτικά παραδείγματα.

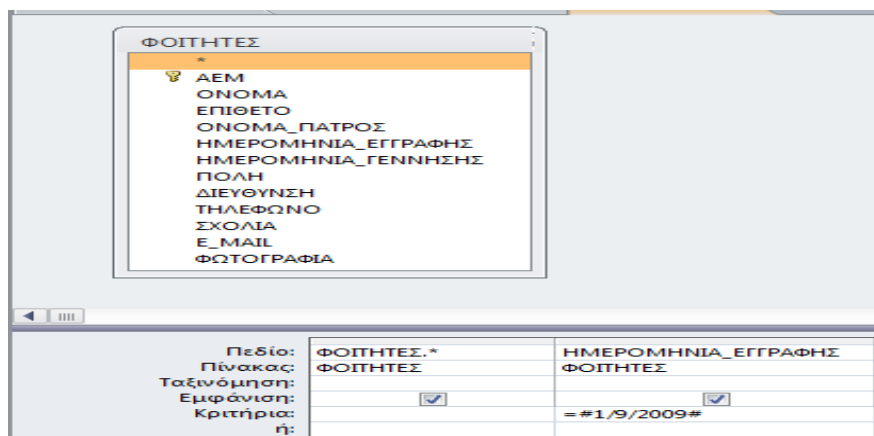


Εικόνα 34


## Ερώτημα 1

Βρείτε όλους τους φοιτητές που η ημερομηνία εγγραφής τους είναι 1/9/2009.

**Λύση:**



Εικόνα 35

Εάν πατήσετε εκτέλεση  θα δείτε ότι εμφανίζει μόνο αυτούς τους φοιτητές που η ημερομηνία εγγραφής τους είναι 1/9/2009.

The screenshot shows the query results table with the following data:

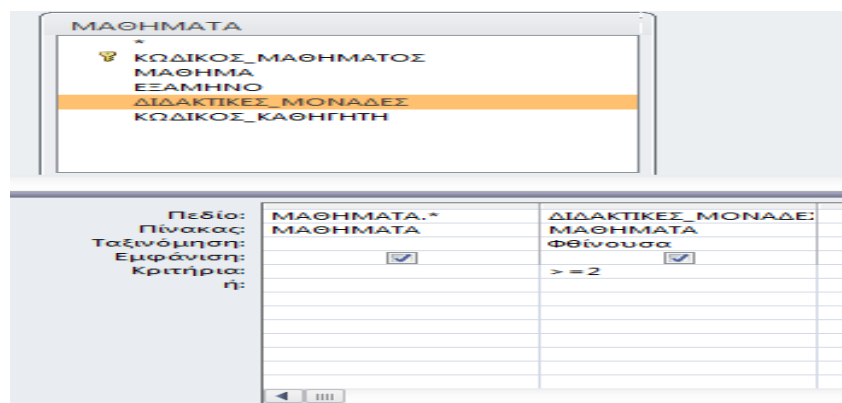
ΑΕΜ	ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΦΟΙΤΗΤΕΣ.ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	ΠΟΛΗ	ΔΙΕΥΘΥΝΣΗ	ΤΗΛΕΦΩΝΟ
1	ΜΑΡΙΑ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΧΡΗΣΤΟΣ	1/9/2009	1/1/1985	ΣΕΡΡΕΣ	ΣΤΑΜΟΥΛΗ 19	698

Εικόνα 36

## Ερώτημα 2

Εμφανίστε από τον πίνακα ΜΑΘΗΜΑΤΑ όλα τα μαθήματα που έχουν διδακτικές μονάδες μεγαλύτερες από 2 κατά φθίνουσα σειρά.

**Λύση:**

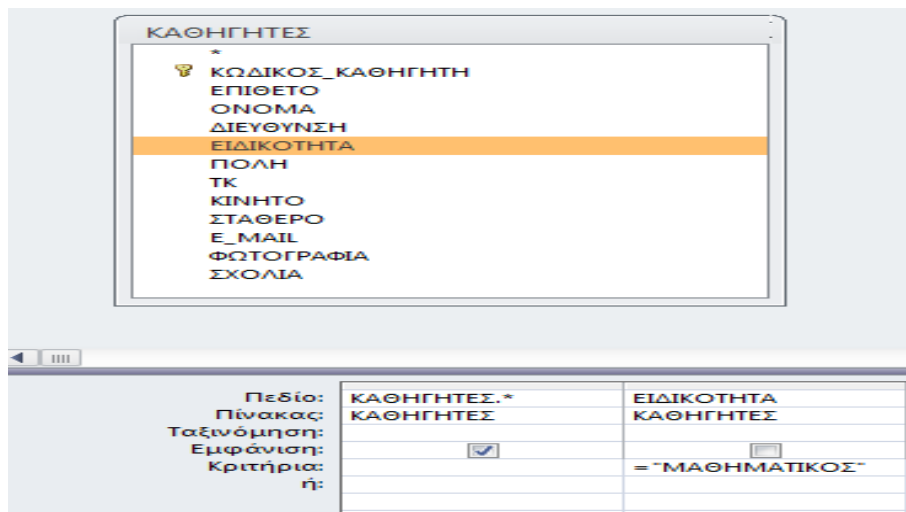


Εικόνα 37

### Ερώτημα 3

Εμφανίστε όλους τους καθηγητές που η ειδικότητα του είναι ΜΑΘΗΜΑΤΙΚΟΣ.

### Λύση:

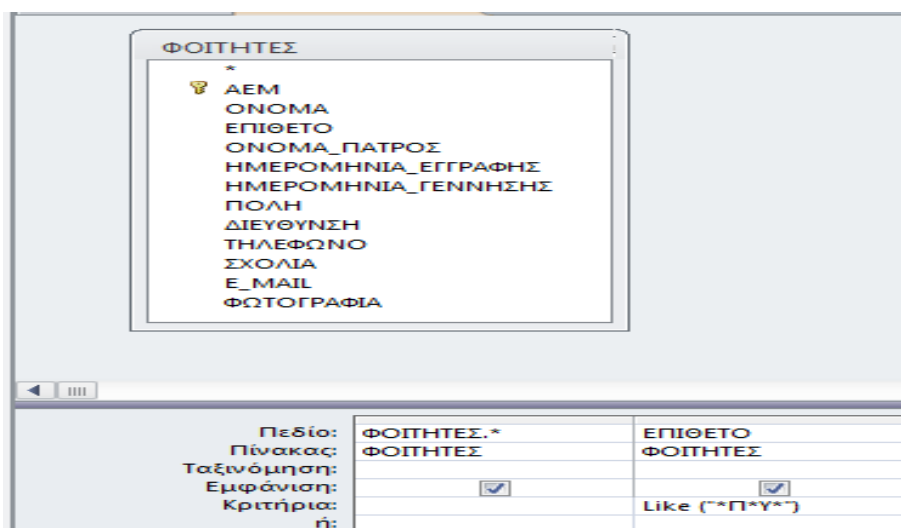


Εικόνα 38

### Ερώτημα 4

Εμφανίστε όλους τους φοιτητές που το επίθετο τους να ξεκινάει από Π και τελειώνει σε Υ.

### Λύση

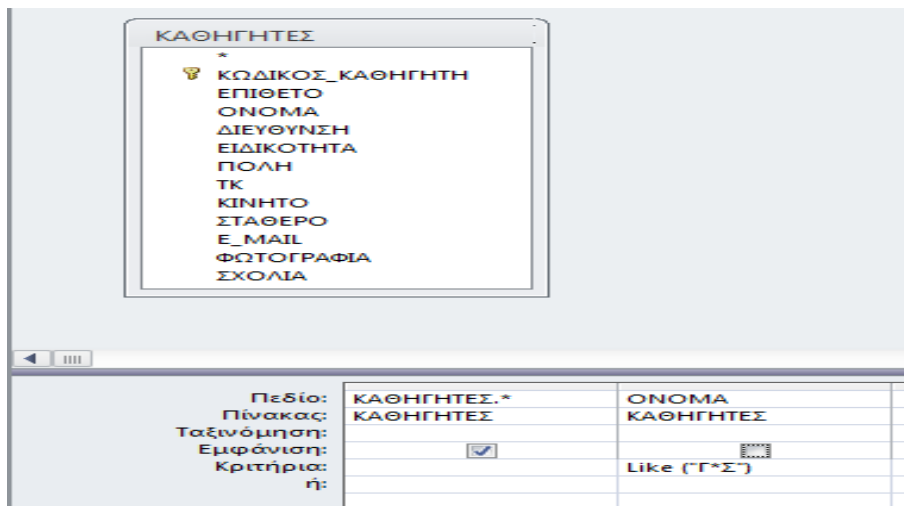


Εικόνα 39

## Ερώτημα 5

Εμφανίστε τους καθηγητές που το όνομα τους ξεκινάει από Γ και τελειώνει σε Σ.

## Λύση

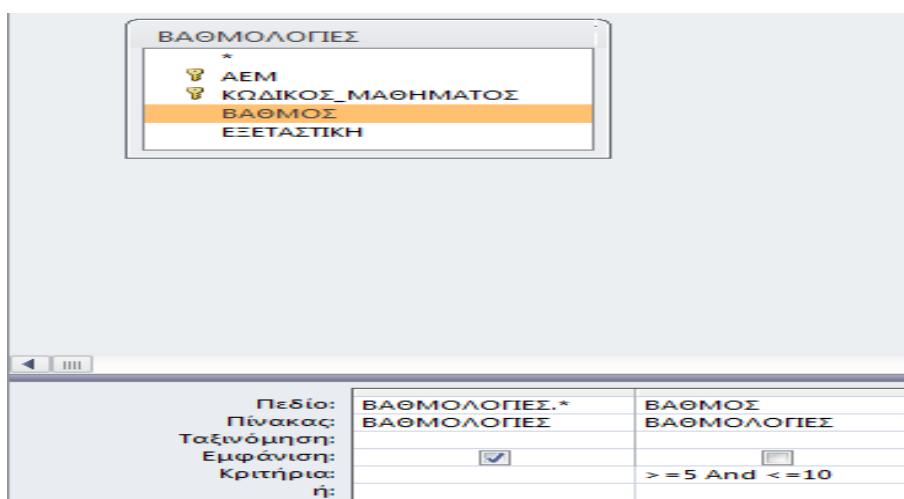


Εικόνα 40

## Ερώτημα 6

Εμφανίστε όλες τους βαθμούς του πίνακα ΒΑΘΜΟΛΟΓΙΕΣ που είναι μεγαλύτεροι-ίσοι του πέντε και μικρότεροι-ίσοι του 10 με φθίνουσα ταξινόμηση.

## Λύση

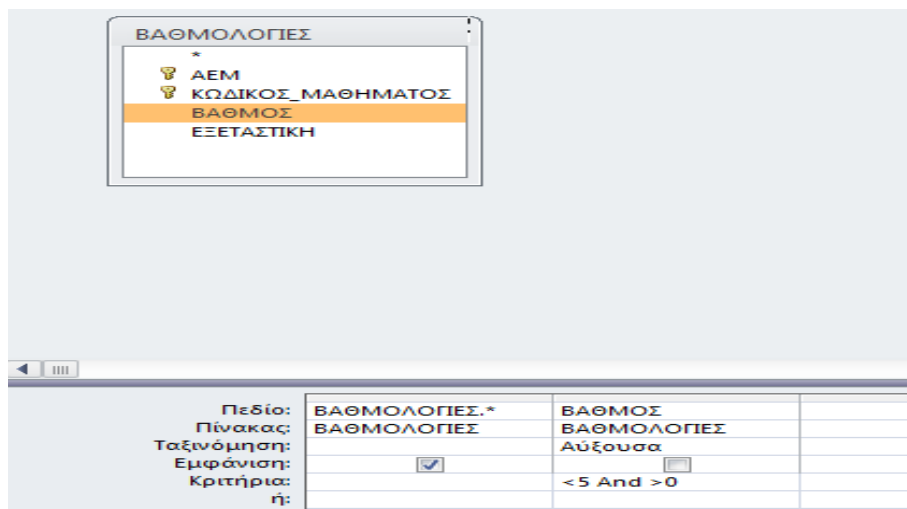


Εικόνα 41

## Ερώτημα 7

Εμφανίστε όλες τους βαθμούς του πίνακα ΒΑΘΜΟΛΟΓΙΕΣ που είναι μικρότεροι του πέντε και μεγαλύτεροι του 0 με αύξουσα ταξινόμηση.

## Λύση

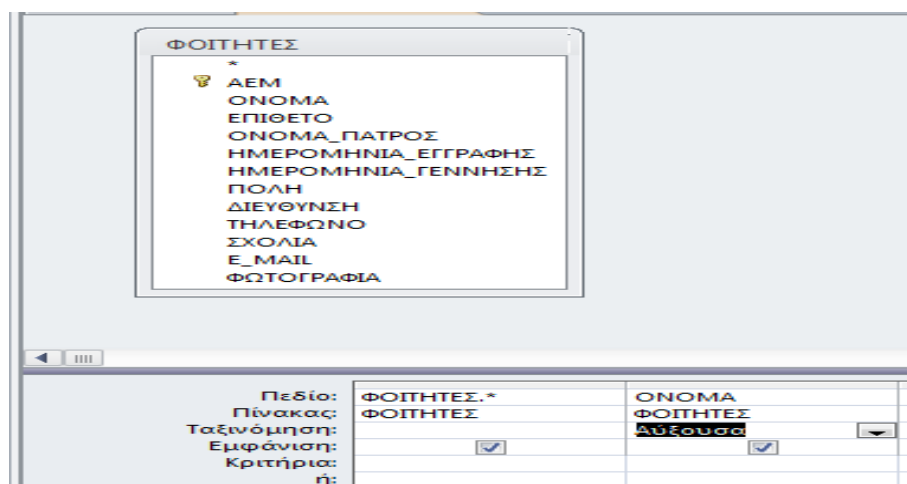


Εικόνα 42

## Ερώτημα 8

Εμφανίστε όλους τους φοιτητές του πίνακα ΦΟΙΤΗΤΕΣ ταξινομημένους αλφαβητικά.

## Λύση

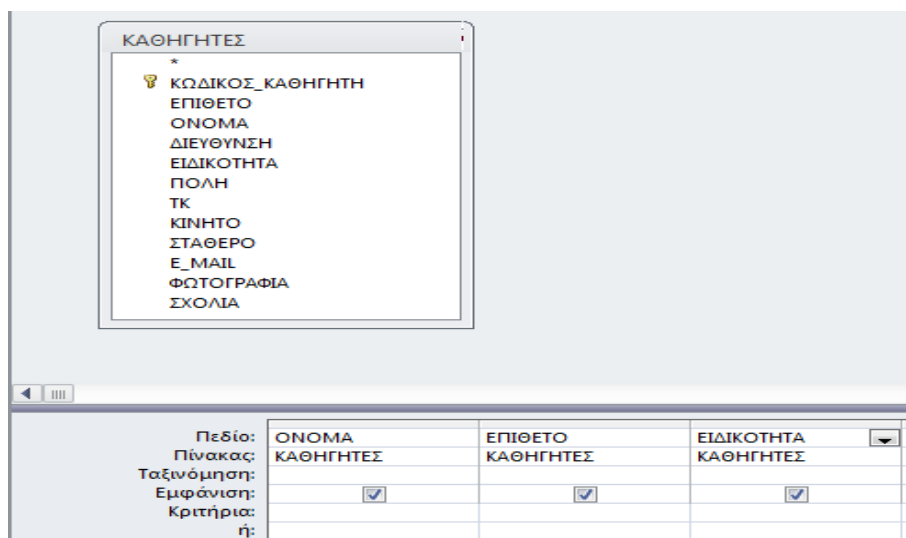


Εικόνα 43

## Ερώτημα 9

Εμφανίστε σε ένα νέο πεδίο το όνομα το επίθετο και την ειδικότητα από τον πίνακα ΚΑΘΗΓΗΤΕΣ.

## Λύση

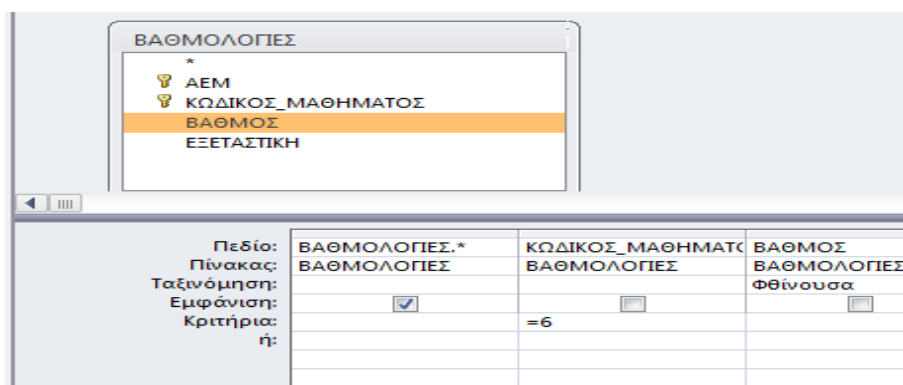


Εικόνα 44

## Ερώτημα 10

Εμφανίστε όλες τις βαθμολογίες του μαθήματος Βάσεις δεδομένων σε φθίνουσα σειρά.

## Λύση



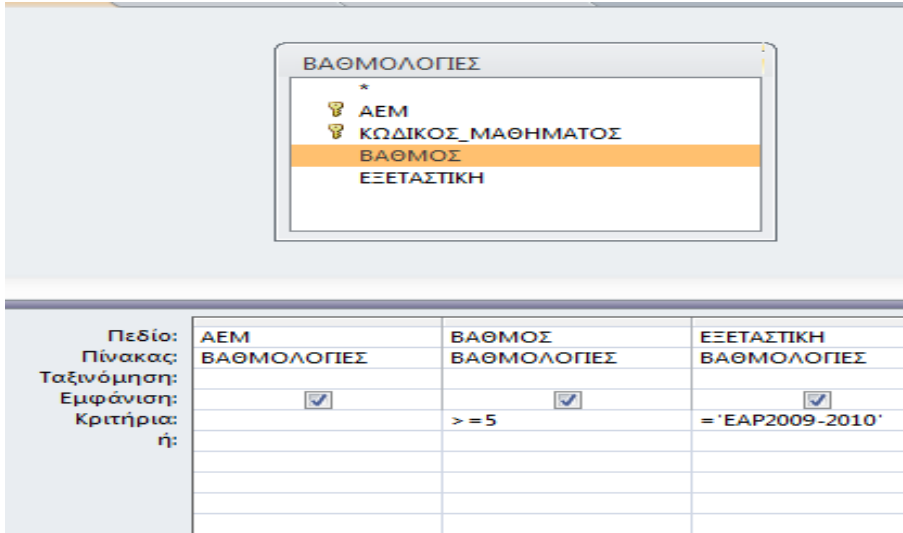
Εικόνα 45



## Ερώτημα 11

Εμφανίστε όλους τους φοιτητές που ο βαθμός τους είναι μεγαλύτερος ή ίσος του 5 σε κάποιο μάθημα της ΕΑΡ2009-2010 εξεταστικής.

## Λύση

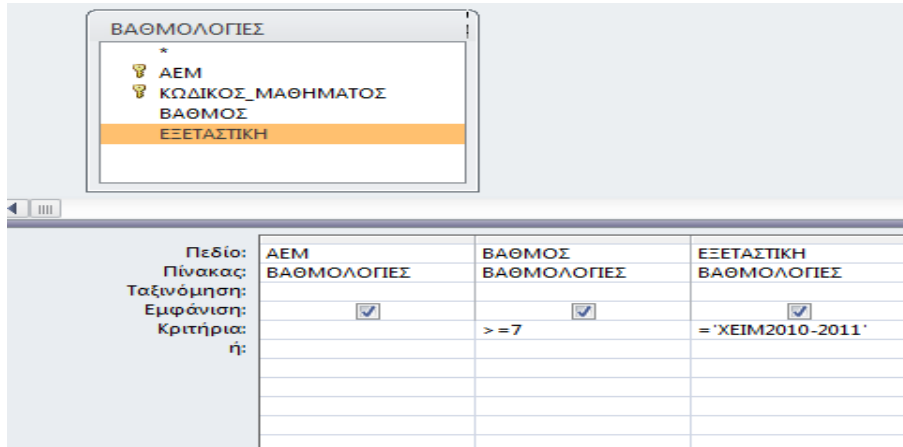


The screenshot shows a database query interface. At the top, a window titled "ΒΑΘΜΟΛΟΓΙΕΣ" contains a list of fields: ΑΕΜ, ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ, ΒΑΘΜΟΣ (highlighted), and ΕΞΕΤΑΣΤΙΚΗ. Below this, a table defines the query parameters:

Πεδίο:	ΑΕΜ	ΒΑΘΜΟΣ	ΕΞΕΤΑΣΤΙΚΗ
Πίνακας:	ΒΑΘΜΟΛΟΓΙΕΣ	ΒΑΘΜΟΛΟΓΙΕΣ	ΒΑΘΜΟΛΟΓΙΕΣ
Ταξινόμηση:			
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Κριτήρια:		> = 5	= 'ΕΑΡ2009-2010'
ή:			

Εικόνα 46

Εμφανίστε όλους τους φοιτητές που ο βαθμός τους είναι μεγαλύτερος ή ίσος του 7 σε κάποιο μάθημα της ΧΕΙΜ2010-2011 εξεταστικής.



The screenshot shows a database query interface. At the top, a window titled "ΒΑΘΜΟΛΟΓΙΕΣ" contains a list of fields: ΑΕΜ, ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ, ΒΑΘΜΟΣ (highlighted), and ΕΞΕΤΑΣΤΙΚΗ. Below this, a table defines the query parameters:

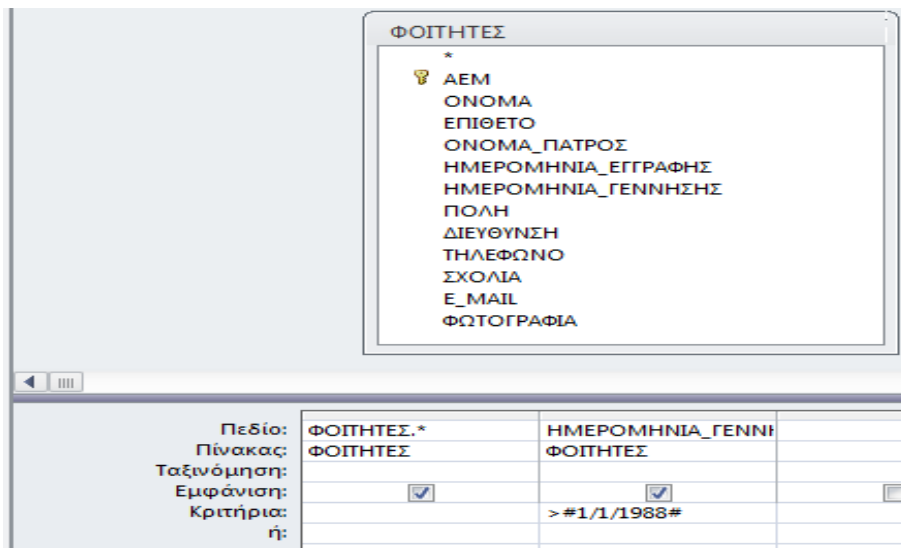
Πεδίο:	ΑΕΜ	ΒΑΘΜΟΣ	ΕΞΕΤΑΣΤΙΚΗ
Πίνακας:	ΒΑΘΜΟΛΟΓΙΕΣ	ΒΑΘΜΟΛΟΓΙΕΣ	ΒΑΘΜΟΛΟΓΙΕΣ
Ταξινόμηση:			
Εμφάνιση:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Κριτήρια:		> = 7	= 'ΧΕΙΜ2010-2011'
ή:			

Εικόνα 47

## Ερώτημα 12

Εμφανίστε όλους τους φοιτητές που γεννήθηκαν από το 1988 και μετά.

### Λύση



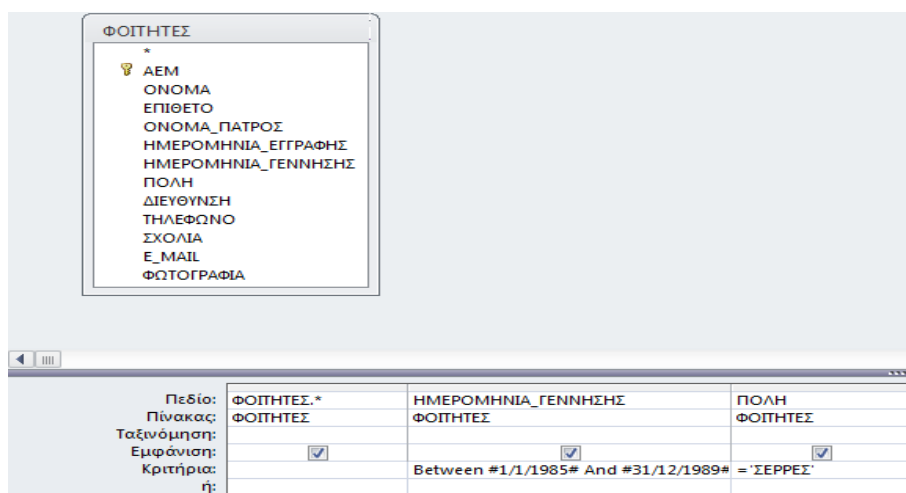
Εικόνα 48

## Ερώτημα 13

Εμφανίστε όλους τους φοιτητές που γεννήθηκαν από το 1985 έως το 1988 και μένουν Σέρρες.

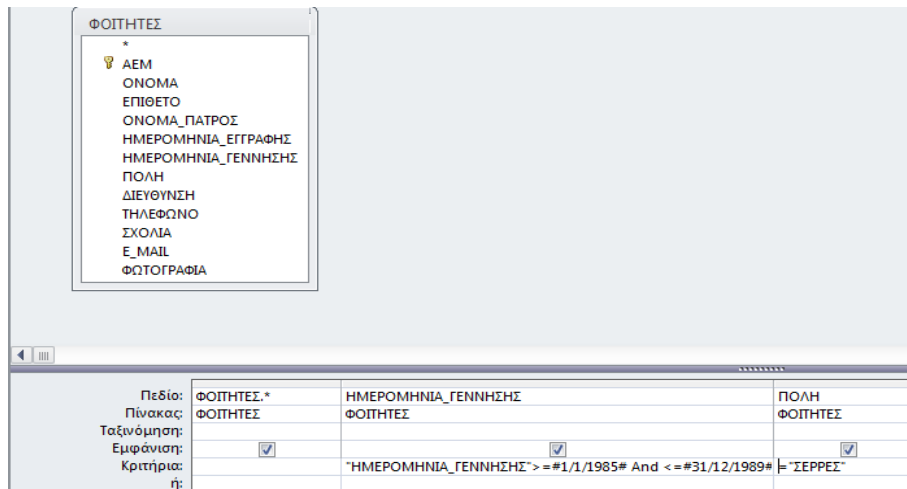
### Λύση

#### 1<sup>ος</sup> Τρόπος



Εικόνα 49

## 2<sup>ος</sup> Τρόπος

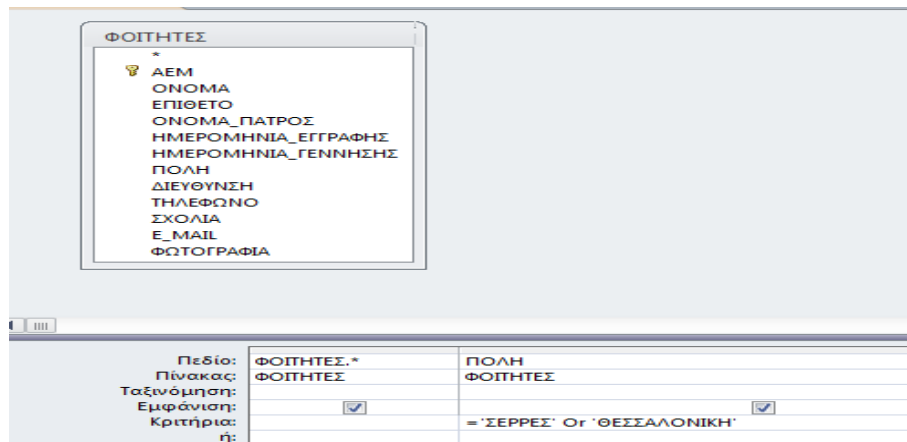


Εικόνα 50

## Ερώτημα 14

Εμφανίστε όλους τους φοιτητές που είναι από Σέρρες ή Θεσσαλονίκη.

## Λύση

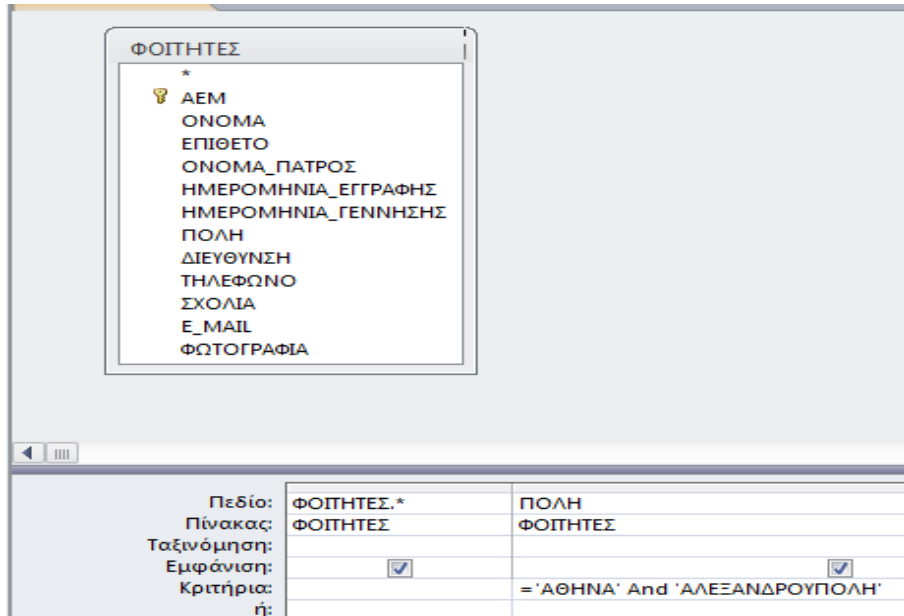


Εικόνα 51

## Ερώτημα 15

Εμφανίστε όλους τους φοιτητές που είναι από Αθήνα και Αλεξανδρούπολη.

## Λύση



Εικόνα 52

## Άσκηση 4 – Εργαλεία σχεδιασμού φόρμών και εκθέσεων σε Access










Σκοπός αυτής της άσκησης είναι η εξοικείωση σας με τα εργαλεία φόρμών και εκθέσεων όπως τα σύνθετα πλαίσια, τα πλαίσια κειμένου, οι ετικέτες και τα πλαίσια έλεγχου.













Μια φόρμα είναι ένα αντικείμενο βάσης δεδομένων το οποίο μπορείτε να χρησιμοποιήσετε για να δημιουργήσετε ένα περιβάλλον χρήστη για μια εφαρμογή βάσης δεδομένων. Για το πώς μπορεί να δημιουργηθεί μία φόρμα από το μηδέν έχει γίνει αναφορά σε προηγούμενο κεφάλαιο σε αυτό θα ασχοληθείτε κυρίως με τα εργαλεία που μπορούν να χρησιμοποιηθούν σε μία φόρμα.

Μια φόρμα αποτελείται από τα εξής μέρη:

- Λεπτομέρειες→Εμφανίζουντε τα δεδομένα, τα πεδία των πινάκων , των ερωτημάτων τα οποία αλλάζουν σε κάθε εγγραφή.
- Κεφαλίδα-Υποσέλιδο→Οι πληροφορίες εμφανίζονται σε όλες τις εγγραφές.
- Στοιχεία έλεγχου→Εμπεριέχει όλα τα δεδομένα της φόρμας. Μπορούν αν είναι δεσμευμένα ή μη δεσμευμένα.

Περιγραφή εργαλείων

	Επιλογή αντικειμένων. Μπορείτε εκτός του να το επιλέξουμε να το μετακινήσουμε και να αλλάξουμε το μέγεθός του.
	Πλαίσιο κειμένου. Για δημιουργία στοιχείων έλεγχου πλαισίων κειμένου με σκοπό την εμφάνιση κειμένων, αριθμών ή ημερομηνιών. Επίσης μπορείτε να εμφανίσετε το αποτέλεσμα ενός υπολογισμού.
	Ετικέτα. Δημιουργεί στοιχεία έλεγχου ετικετών.
	Εισαγωγή κουμπιού.
	Εισαγωγή υπερσύνδεσης.
	Στοιχείο έλεγχου προγράμματος περιήγησης Web.
	Έλεγχος περιήγησης.
	Ομάδα επιλογών. Δημιουργεί στοιχεία έλεγχου τα οποία μπορούν να περιέχουν κουμπιά εναλλαγής, πλαίσια έλεγχου ή κουμπιά επιλογών.
	Αλλαγή σελίδας χρησιμοποιείται για την διακοπή μιας σελίδας σε μία φόρμα.

	Σύνθετο πλαίσιο καταλόγου. Εμφανίζοντε όλες οι πιθανές τιμές όπως επίσης και το απλό πλαίσιο καταλόγου.
	Για εισαγωγή γραφήματος.
	Για εισαγωγή γραμμών σε μία φόρμα.
	Κουμπί εναλλαγής.
	Πλαίσιο καταλόγου.
	Ορθογώνιο. Χρησιμοποιείτε για την εισαγωγή ορθογωνίων σε μία φόρμα.
	Πλαίσιο έλεγχου χρησιμοποιείται για να καταχωρούμε τιμές όπως Ενεργό/Ανενεργό, Ναι/Όχι. Όταν είναι επιλεγμένο παίρνει την τιμή -1 ενώ όταν δεν είναι παίρνει την τιμή 0. Εάν υπάρχουν πολλά πλαίσια μπορεί να επιλέξει πάνω από ένα.
	Πλαίσιο μη δεσμευμένου αντικειμένου. Χρησιμοποιείται για να εισάγετε ένα αντικείμενο στην φόρμα το οποίο έχει δημιουργηθεί από μία άλλη εφαρμογή αρκεί βέβαια αυτό να υποστηρίζει την τεχνική OLE.
	Για εισαγωγή συνημμένου.
	Κουμπί επιλογής ή ραδιόπληκτρο. Χρησιμοποιείται συνήθως σε ομάδες επιλογών. Ο χρήστης μπορεί αν επιλέξει μόνο ένα πλήκτρο κάθε φορά.
	Δευτερεύουσα φόρμα/έκθεση. Για την ενσωμάτωση μιας δευτερεύουσας μέσα στην τρέχουσα.
	Πλαίσιο δεσμευμένου αντικειμένου. Χρησιμοποιείται για τη σύνδεση ενός αντικειμένου OLE μέσα στη φόρμα.

Για να εισάγεται κάποιο από αυτά τα εργαλεία σε μία φόρμα απαραίτητο είναι να βρίσκεστε σε προβολή σχεδίασης της φόρμας. Έπειτα η διαδικασία είναι απλή διαλέγεται το εργαλείο που θέλετε να εισάγεται στην φόρμα με διπλό κλικ καθώς επίσης και το μέρος της φόρμας (κεφαλίδα, υποσέλιδο, λεπτομέρειες) που επιθυμείτε να το τοποθετήσετε. Έπειτα το ενσωματώνετε στην φόρμα σας. Παρακάτω ακολουθεί παράδειγμα.

### Παράδειγμα:

Δημιουργήστε μία φόρμα με όνομα καρτέλα φοιτητή ή ανοίξτε την ήδη υπάρχουσα στην οποία θα εισάγετε μία ετικέτα στην κεφαλίδα με όνομα ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ και ένα σύνθετο πλαίσιο κειμένου.

- Ενώ βρίσκεστε σε προβολή σχεδίασης της φόρμας Καρτέλα\_ Φοιτητή από την καρτέλα Σχεδίαση της κορδέλας εισάγετε ετικέτα **Aa** στην κεφαλίδα και πληκτρολογείτε το όνομα της .

Κεφαλίδα φόρμας			
ΦΟΙΤΗΤΕΣ			
ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ			
Λεπτομέρεια			
ΑΕΜ	ΑΕΜ	ΠΟΛΗ	ΠΟΛΗ
ΟΝΟΜΑ	ΟΝΟΜΑ	ΔΙΕΥΘΥΝΣΗ	ΔΙΕΥΘΥΝΣΗ
ΕΠΙΘΕΤΟ	ΕΠΙΘΕΤΟ	ΤΗΛΕΦΩΝΟ	ΤΗΛΕΦΩΝΟ
ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΣΧΟΛΙΑ	ΣΧΟΛΙΑ
ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ	Ε_ΜΑΙΛ	Ε_ΜΑΙΛ
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	ΦΟΤΟΓΡΑΦΙΑ	

Εικόνα 53

**ΣΧΟΛΙΟ!** Για την μετακίνηση των στοιχείων, τα επιλέγετε και μετακινείτε σε περίπτωση που είναι ομαδοποιημένα επιλέγετε τα στοιχεία, **δεξί κλικ** → **Διάταξη** → **Κατάργηση διάταξης**.

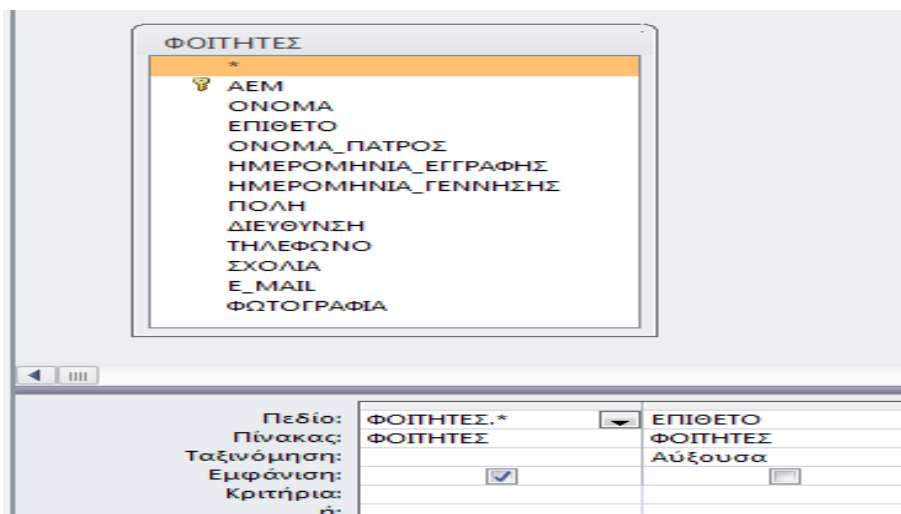
Ομοίως και για της άλλες τρεις καρτέλες ( καρτέλα\_ Βαθμολογίες, καρτέλα\_ μαθήματα, καρτέλα καθηγητές).

Η σχεδίαση μίας φόρμας και μίας έκθεσης δεν απέχουν πολύ μεταξύ τους τα εργαλεία που χρησιμοποιούν είναι τα ίδια. Η διαφορά τους είναι στη χρήση. Μία έκθεση συνήθως χρησιμοποιείται για την προβολή, την μορφοποίηση και την σύνοψη δεδομένων όταν χρησιμοποιείται σε βάση δεδομένων.

### Παράδειγμα:

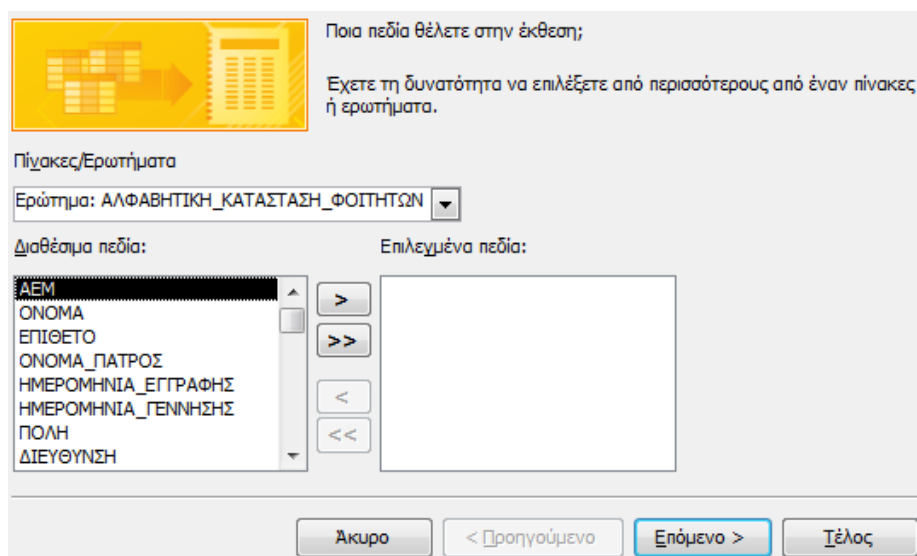
Δημιουργήστε ένα ερώτημα όπου θα εμφανίζονται με αλφαβητική σειρά οι φοιτητές σύμφωνα με το επίθετο τους και τοποθετήστε το στην έκθεση έπειτα εισάγετε ένα σύνθετο πλαίσιο κειμένου και μία ετικέτα.

- Δημιουργία ερωτήματος με όνομα Αλφαβητική\_ Κατάσταση.



Εικόνα 54

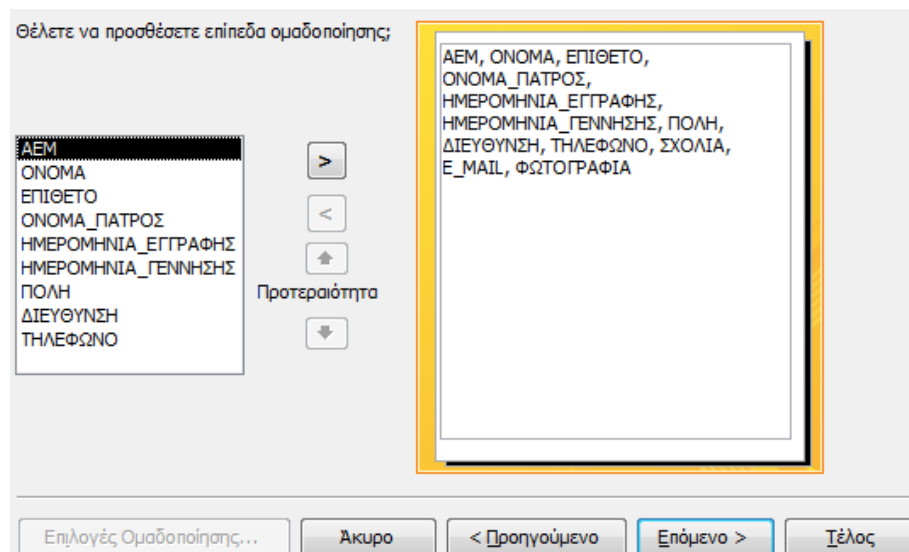
- Για την τοποθέτησή του στην έκθεση κάντε το εξής: ενώ βρίσκεστε στο ερώτημα πηγαίνετε **Δημιουργία→Οδηγός εκθέσεων** και ανοίγει το παράθυρο της Εικόνας 53, επιλέγετε τα πεδία που θέλετε να εμφανιστούν στην έκθεση και επόμενο.



Εικόνα 55

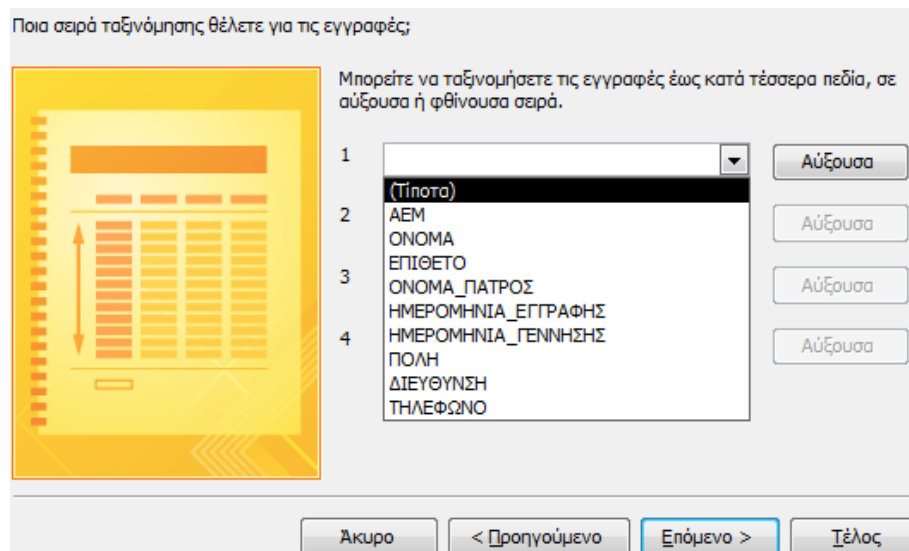


- Έπειτα εισάγετε κάποια άλλα πεδία για ομαδοποίηση εάν επιθυμείτε και επόμενο. Στο παράδειγμα δεν χρειάζεται η εισαγωγή άλλων πεδίων οπότε **Επόμενο**.



Εικόνα 56

- Επίσης στο τρέχον παράδειγμα δεν χρειάζεται κάποια ταξινόμηση μιας και είναι ενσωματωμένη στο ερώτημα. Επιλέξτε **Τίποτα** και **Επόμενο**.



Εικόνα 57

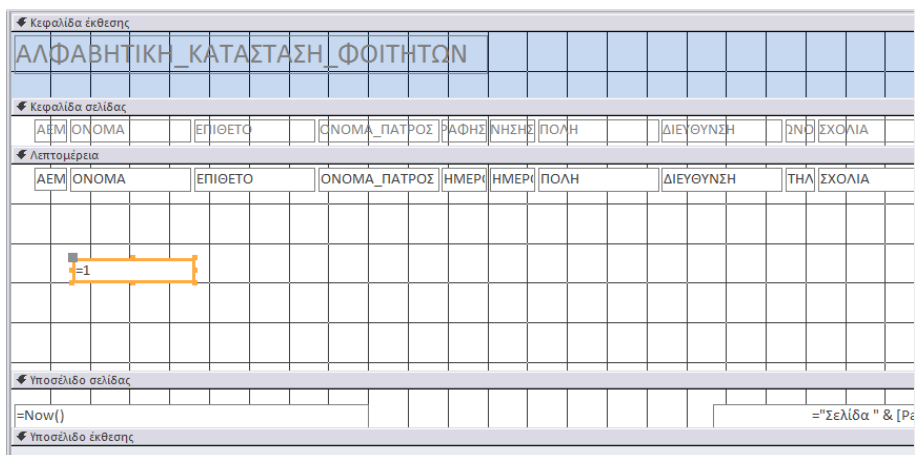
- Επιλέξτε την διάταξη πίνακα ώστε το αποτέλεσμα σε **Προβολή έκθεσης** να είναι πίνακας και επιπλέον τον **Οριζόντιο προσανατολισμό** ώστε να είναι εμφανή όλα τα πεδία σας. Τσεκάρετε την **Προσαρμογή του πλάτους των πεδίων** και **Επόμενο**.

Εικόνα 58

- Μετά την επιλογή επόμενο πατάτε **Προεπισκόπηση Φόρμας** και **Τέλος**, τότε εμφανίζεται η έκθεση σε προβολή σχεδίασης.

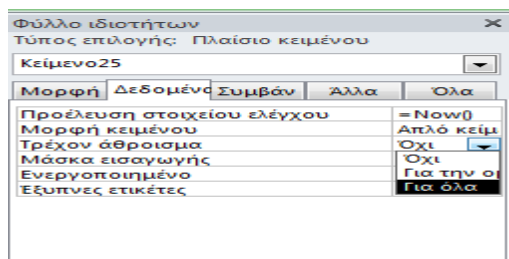
Εικόνα 59

- Έπειτα με διπλό κλικ επιλέγετε ένα πλαίσιο κειμένου **ab** και το τοποθετήτε στην έκθεση στο μέρος λεπτομέρια. Όταν εμφανιστεί το πλαίσιο κειμένου ταυτόχρονα εμφανίζεται και μία ετικέτα δίπλα του την οποία την σβήνετε. Μέσα στο πλαίσιο κειμένου γράφετε =1.



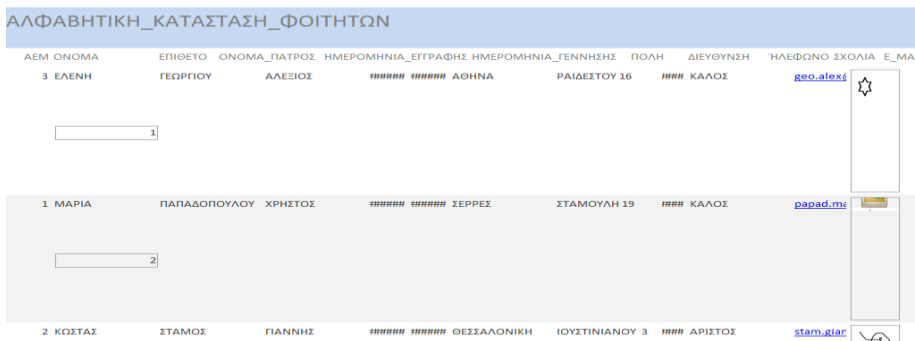
Εικόνα 60

- Έπειτα το επιλέγετε και με **δεξί κλικ** μπαίνετε στις **ιδιότητες** του στην καρτέλα δεδομένα και κάντε **Το τρέχον άθροισμα** → Για όλα όπως βλέπετε και στην Εικόνα 59.



Εικόνα 61

- Στη συνέχεια σβήστε από την κεφαλίδα και από την λεπτομέρεια το ΑΕΜ και στην θέση της λεπτομέρειας τοποθετήστε το πλαίσιο κειμένου **ab|** ενώ στην κεφαλίδα τοποθετήστε μία ετικέτα **Aa** με όνομα AA. Τέλος με την προβολή της έκθεσης βλέπετε το αποτέλεσμα της Εικόνας 60.



Εικόνα 62

## Άσκηση 5 – Γλώσσα χειρισμού δεδομένων SQL σε Access

Σκοπός αυτής της άσκησης είναι να σας εισάγει στην γλώσσα χειρισμού δεδομένων SQL με εντολές τροποποίησης πινάκων , την διαγραφή τους, την εισαγωγή δεδομένων σε αυτούς, την ενημέρωσή τους και τέλος την προβολή τους.

Το σχεσιακό μοντέλο μίας βάσης δεδομένων αποτελείται από σχέσεις , γνωρίσματα, πρωτεύον κλειδιά, αναφορικές ακεραιότητες ξένων κλειδίων, ακεραιότητες οντότητας και σχεσιακές πράξεις. Βάση όλων αυτών συντάσσονται και οι προτάσεις της γλώσσας SQL.

Για την δημιουργία ενός ερωτήματος **SQL** πρέπει να κάνετε τα εξής: πηγαίνετε στην καρτέλα **Δημιουργία** της κορδέλας, **Σχεδίαση ερωτήματος**, και τέλος **Προβολή SQL**.

### Εισαγωγή δεδομένων σε πίνακα

Η εντολή εισαγωγής μίας εγγραφής σε ένα πίνακα είναι η:

```
INSERT INTO ΠΙΝΑΚΑ (ΠΕΔΙΑ_ΠΙΝΑΚΑ) VALUES (ΤΙΜΗ1 , ΤΙΜΗ2... ) ;
```

**ΣΧΟΛΙΑ!** Οι τιμές που μπορούν να εισαχθούν στο VALUES είναι είτε συμβολοσειρές είτε αριθμοί. Οι συμβολοσειρές περικλείονται σε εισαγωγικά ενώ οι αριθμοί όχι. Επίσης οι ημερομηνίες περικλείονται σε διέσεις #.

#### Παράδειγμα 1

Δημιουργήστε ένα ερώτημα το οποίο θα εισάγει δεδομένα στον πίνακα ΜΑΘΗΜΑΤΑ το μάθημα ΠΙΘΑΝΟΤΗΤΕΣ που είναι πέμπτου εξαμήνου και έχει επτά διδακτικές μονάδες.

```
INSERT INTO ΜΑΘΗΜΑΤΑ (ΜΑΘΗΜΑ, ΕΞΑΜΗΝΟ, ΔΙΔΑΚΤΙΚΕΣ_ΜΟΝΑΔΕΣ) VALUES ('ΣΤΑΤΙΣΤΙΚΗ', 5, 7);
```

### Τροποποίηση δεδομένων πίνακα

Η εντολή για την τροποποίηση σε ένα πίνακα είναι η:

```
UPDATE ΠΙΝΑΚΑ SET ΠΕΔΙΟ= ΝΕΑ_ΕΓΓΡΑΦΗ [WHERE ΚΛΕΔΙ=ΤΙΜΗ]
```

#### Παράδειγμα 2

Αλλάξτε το όνομα του φοιτητή που έχει επίθετο ΣΤΑΜΟΣ από ΚΩΣΤΑΣ σε ΧΡΗΣΤΟΣ.

```
UPDATE ΦΟΙΤΗΤΕΣ SET ΟΝΟΜΑ='ΧΡΗΣΤΟΣ' WHERE ΕΠΙΘΕΤΟ='ΣΤΑΜΟΣ' AND ΟΝΟΜΑ='ΚΩΣΤΑΣ';
```

## Διαγραφή εγγραφής από πίνακα

Η εντολή για την διαγραφή ενός πεδίου από ένα πίνακα είναι:

```
DELETE FROM ΠΙΝΑΚΑ WHERE ΠΕΔΙΟ=ΕΓΓΡΑΦΗ;
```

### Παράδειγμα 3

Διαγράψτε το μάθημα ΜΑΘΗΜΑΤΙΚΑ του πίνακα ΜΑΘΗΜΑΤΑ.

```
DELETE FROM ΜΑΘΗΜΑΤΑ WHERE ΜΑΘΗΜΑ='ΜΑΘΗΜΑΤΙΚΑ' ;
```

## Προβολή στοιχείων από πίνακα

```
SELECT ΠΕΔΙΑ_ΓΙΑ_ΠΡΟΒΟΛΗ FROM ΠΙΝΑΚΑ;
```

### Παράδειγμα 4

Εμφανίστε τα πεδία ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΑΕΜ του πίνακα ΦΟΙΤΗΤΕΣ.

```
SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΑΕΜ FROM ΦΟΙΤΗΤΕΣ;
```

## Επιλογή εγγραφών και όλων των πεδίων ενός πίνακα

```
SELECT ΠΕΔΙΑ_ΓΙΑ_ΠΡΟΒΟΛΗ ή * (Με το αστεράκι εμφανίζονται  
όλα τα πεδία του πίνακα)
```

```
FROM ΠΙΝΑΚΑ
```

```
WHERE ΠΕΔΙΟ=ΕΓΓΡΑΦΗ;
```

Με το **WHERE** γίνεται η επιλογή των εγγραφών που ικανοποιούν μία ή περισσότερες λογικές συνθήκες. Την θέση του = μπορούν να πάρουν οι τελεστές σύγκρισης του ακόλουθου πίνακα

=	ίσον
<	μικρότερο
>	Μεγαλύτερο
<>	διάφορο
>=	Μεγαλύτερο ή ίσο
<=	Μικρότερο ή ίσο

καθώς και να γίνει συνδυασμός των τελεστών αυτών με τη βοήθεια των λογικών τελεστών (not, and, or) και των αριθμητικών τελεστών (+, -, \*, /). Τέλος υπάρχουν και οι τελεστές σε συνθήκες του ακόλουθου πίνακα.

Τελεστές	Περιγραφή.
<b>between ... and ...</b>	Περιλαμβάνει εγγραφές με τιμή πεδίου εντός της ορισμένης κλίμακας.
<b>not between ... and ...</b>	Αποκλείει εγγραφές με τιμή πεδίου εντός της ορισμένης κλίμακας.
<b>like</b>	Περιλαμβάνει εγγραφές με πεδίο συμβολοσειράς που να ταιριάζει στον τύπο.
<b>not like</b>	Αποκλείει εγγραφές με πεδίο συμβολοσειράς που να ταιριάζει στον τύπο.
<b>in</b>	Περιλαμβάνει εγγραφές με τιμή πεδίου που περιέχετε σε ορισμένη λίστα.
<b>not in</b>	Αποκλείει εγγραφές με τιμή πεδίου που περιέχετε σε ορισμένη λίστα.
<b>is null</b>	Περιλαμβάνει εγγραφές με τιμή κενή (null) στο πεδίο σύγκρισης.
<b>is not null</b>	Αποκλείει εγγραφές με τιμή κενή (null) στο πεδίο σύγκρισης.

### Παράδειγμα 5

Εμφανίστε τα μαθήματα που έχουν διδακτικές μονάδες μεγαλύτερες ή ίσες του τέσσερα και το εξάμηνο τους είναι μικρότερο ή ίσο του έκτου.

```
SELECT * FROM ΜΑΘΗΜΑΤΑ WHERE ΔΙΔΑΚΤΙΚΕΣ_ΜΟΝΑΔΕΣ>=4 AND
ΕΞΑΜΗΝΟ<=6;
```

Επιπλέον μπορείτε να χρησιμοποιήσετε την εντολή **ORDER BY** για ταξινόμηση των πεδίων που επιθυμείτε. Στο παράδειγμα 5 έστω ότι θέλετε έκτος από την εμφάνιση των πεδίων του πίνακα μαθήματα σύμφωνα με τι αντίστοιχες διδακτικές που έχετε επιλέξει να τα ταξινομήσετε και ως προς το εξάμηνο. Τότε το παράδειγμα θα γίνει ως εξής:

```
SELECT * FROM ΜΑΘΗΜΑΤΑ WHERE ΔΙΔΑΚΤΙΚΕΣ_ΜΟΝΑΔΕΣ>=4 AND
ΕΞΑΜΗΝΟ<=6 ORDER BY ΕΞΑΜΗΝΟ;
```

Για φθίνουσα ταξινόμηση χρησιμοποιείτε την εντολή **ORDER BY ΠΕΔΙΟ DESC**.

Τέλος υπάρχουν ακόμη δύο εντολές δημιουργίας ερωτημάτων η **GROUP BY** η οποία κάνει ομαδοποίηση εγγραφών και η **HAVING** η οποία δημιουργεί συνθήκες περιορισμού των γραμμών που επιστρέφονται από μια ομαδοποίηση. Η σύνταξη τους είναι ως εξής:

**SELECT** ΠΕΔΙΑ\_ΓΙΑ\_ΠΡΟΒΟΛΗ ή \* (Με το αστεράκι εμφανίζονται όλα τα πεδία του πίνακα)

**FROM** ΠΙΝΑΚΑ

**WHERE** ΠΕΔΙΟ=ΕΓΓΡΑΦΗ

**GROUP BY** ΠΕΔΙΑ\_ΟΜΑΔΟΠΟΙΗΣΗΣ

**HAVING** ΣΥΝΘΗΚΗ\_ΟΜΑΔΟΠΟΙΗΣΗΣ

### Ερωτήσεις εξάσκησης

1. Εμφανίστε όλα τα στοιχεία των φοιτητών που η απόδοση τους είναι καλή (δηλαδή ψάξτε στο πεδίο σχόλια τον χαρακτηρισμό καλός).
2. Διαγράψτε όλους τους φοιτητές που το εξάμηνο τους είναι μεγαλύτερο του όγδοου.
3. Αλλάξτε το όνομα του καθηγητή που έχει επίθετο ΠΑΠΑΣ από ΗΛΙΑΣ σε ΓΙΑΝΝΗΣ.
4. Εμφανίστε όλες τις εγγραφές μαθημάτων με κωδικό μαθήματος μεγαλύτερο του 3.
5. Εμφανίστε όλους τους φοιτητές που μπήκαν στην σχολή πριν της 2/9/2009.
6. Αλλάξτε τις διδακτικές μονάδες του μαθήματος ΒΙΟΜΗΧΑΝΙΚΗ από τέσσερις σε επτά.
7. Εμφανίστε όλα τα μαθήματα που το εξάμηνο τους είναι το τρίτο και ο καθηγητής του μαθήματος έχει κωδικό καθηγητή ίσον με 14.
8. Αφού διαγράψετε τον καθηγητή που κάνει το μάθημα της φυσικής ταξινομήστε τους υπόλοιπους βάση του επιθέτου.
9. Ταξινομήστε τους φοιτητές με φθίνουσα σειρά βάση της ημερομηνίας εγγραφής τους.
10. Εμφανίστε όλα τα ονόματα των καθηγητών με φθίνουσα σειρά και όλα τα επίθετα με αύξουσα σειρά.
11. Εμφανίστε όλους τους φοιτητές που το όνομα τους ξεκινάει από Ε και τελειώνει σε Η.
12. Εμφανίστε τους καθηγητές που έχουν γεννηθεί μεταξύ 1/3/1989 με 1/10/1989.
13. Εμφανίστε όλους τους καθηγητές που τον όνομα τους δεν ξεκινάει με το γράμμα Π.
14. Ομαδοποιήστε τα όνομα και το επίθετο του φοιτητή εφόσον αυτός .....

## Άσκηση 6 – Δημιουργία ερωτημάτων SQL σε Access

### ΤΥΠΟΙ ΔΕΔΟΜΕΝΩΝ

ΤΥΠΟΣ	ΠΕΡΙΓΡΑΦΗ	ΜΕΓΕΘΟΣ
VARCHAR(n)	Αλφαριθμητικό μεταβλητού μήκους (από 0 έως n). Το μέγιστο μήκος ορίζεται από τον χρήστη.	2 byte ανά χαρακτήρα
CHAR(n)	Αλφαριθμητικό σταθερού μήκους n που ορίζεται από τον χρήστη. Το n μπορεί να πάρει τιμές έως 255 χαρακτήρες	2 byte ανά χαρακτήρα
INTEGER	Ακέραιος (Στη σχεδίαση πίνακα της MS Access λέγεται Ακέραιος μεγάλου μήκους)	4 byte
SMALLINT	Μικρός ακέραιος, με εύρος τιμών από -32,768 έως 32,767. Στη σχεδίαση πίνακα της MS Access λέγεται ακέραιος	2 byte
TINYINT	Πολύ μικρός ακέραιος (0 έως 255). Στη σχεδίαση πίνακα της MS Access λέγεται Byte	1 byte
REAL	Πραγματικός αριθμός, απλής ακρίβειας	4 byte
FLOAT	Πραγματικός αριθμός, διπλής ακρίβειας	8 byte
DATETIME	Ημερομηνία και ώρα της μορφής Π.χ. '2002-10-22 09:05:32.75'	8 byte

**ΣΧΟΛΙΟ!** Μετά την δήλωση του τύπου με το **NOT NULL δηλώνετε** ότι το πεδίο δεν μπορεί να είναι κενό δεν επιτρέπεται. Αυτήν την τιμή την παίρνουν συνήθως πεδία που η αρίθμηση τους είναι αυτόματη αλλά όχι μόνο.

### Δημιουργία πίνακα

Για την δημιουργία ενός πίνακα η εντολή είναι η εξής:

```
CREATE TABLE ΠΙΝΑΚΑΣ (ΠΕΔΙΟ 1 ΤΥΠΟΣ-ΔΕΔΟΜΕΝΩΝ 1, ΠΕΔΙΟ 2 ΤΥΠΟΣ -ΔΕΔΟΜΕΝΩΝ 2, ΠΕΔΙΟ 3 ΤΥΠΟΣ-ΔΕΔΟΜΕΝΩΝ 3, ΠΕΡΙΟΡΙΣΜΟΣ 1, ΠΕΡΙΟΡΙΣΜΟΣ 2)
```

Οι περιορισμοί αναφέρονται στους ορισμούς των:

- **PRIMARY KEY** (ΠΕΔΙΟ 1, ΠΕΔΙΟ 2) → Ορίζει τα πεδία που θα αποτελούν το πρωτεύον κλειδί.
- **FOREIGN KEY** (ΠΕΔΙΟ 1) **REFERENCES** ΠΙΝΑΚΑΣ (ΠΕΔΙΟ 1) → Ορίζει σε ποιο πρωτεύον κλειδί θα αναφέρονται οι τιμές του ξένου κλειδιού.



## Παράδειγμα 5

```
CREATE TABLE ΦΟΙΤΗΤΕΣ (ΚΩΔΙΚΟΣ_ΦΟΙΤΗΤΗ INTEGER (30) NOT
NULL, ΟΝΟΜΑ VARCHAR (30) NOT NULL, ΕΠΩΝΥΜΟ VARCHAR (30) NOT
NULL, ΔΙΕΥΘΗΝΣΗ VARCHAR (40), ΠΟΛΗ VARCHAR (20), ΤΗΛΕΦΩΝΟ
VARCHAR (40), ΕΞΑΜΗΝΟ VARCHAR (5),
CONSTRAINT ΚΩΔΙΚΟΣ_ΦΟΙΤΗΤΗ PRIMARY KEY (ΚΩΔΙΚΟΣ_ΦΟΙΤΗΤΗ) );
```

Άλλες χρήσιμες εντολές εκτός από την δημιουργία ενός πίνακα είναι οι εξής:

ΕΝΤΟΛΗ	ΧΡΗΣΗ
DROP TABLE <ΠΙΝΑΚΑΣ>	Διαγραφή Πίνακα
ALTER TABLE <ΠΙΝΑΚΑΣ> ADD COLUMN <ΠΕΔΙΟ> ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ	Προσθήκη πεδίου
ALTER TABLE < ΠΙΝΑΚΑΣ > DROP COLUMN < ΠΕΔΙΟ >	Διαγραφή πεδίου
ALTER TABLE < ΠΙΝΑΚΑΣ > ALTER COLUMN <ΠΕΔΙΟ> νέος τύπος δεδομένων	Αλλαγή τύπου δεδομένων πεδίου
ALTER TABLE < ΠΙΝΑΚΑΣ > ADD CONSTRAINT <ΟΝΟΜΑ ΠΕΡΙΟΡΙΣΜΟΥ> ΟΡΙΣΜΟΣ ΠΕΡΙΟΡΙΣΜΟΥ	Προσθήκη περιορισμού
ALTER TABLE < ΠΙΝΑΚΑΣ > DROP CONSTRAINT <ΟΝΟΜΑ ΠΕΡΙΟΡΙΣΜΟΥ>	Διαγραφή περιορισμού

## ΔΗΜΙΟΥΡΓΙΑ ΠΕΡΙΟΡΙΣΜΩΝ ΑΝΑΦΟΡΙΚΗΣ ΑΚΕΡΑΙΟΤΗΤΑΣ

Ο πίνακας ΒΑΘΜΟΛΟΓΙΕΣ έχει δύο ξένα κλειδιά το ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ του πίνακα ΜΑΘΗΜΑΤΑ και το ΑΕΜ από τον πίνακα ΦΟΙΤΗΤΕΣ. Ο πίνακας ΜΑΘΗΜΑΤΑ έχει ένα ξένο κλειδί ΚΩΔΙΚΟΣ\_ΚΑΘΗΓΗΤΗ το οποίο θα πρέπει να συσχετιστεί με το πρωτεύον κλειδί ΚΩΔΙΚΟΣ\_ΚΑΘΗΓΗΤΗ από τον πίνακα ΚΑΘΗΓΗΤΕΣ. Οι εντολές με τις οποίες δηλώνεται ο περιορισμός της αναφορικής ακεραιότητας για τις παραπάνω συσχετίσεις είναι οι παρακάτω:

```
ALTER TABLE ΒΑΘΜΟΛΟΓΙΕΣ ADD CONSTRAINT FK1 FOREIGN KEY ΑΕΜ
REFERENCES ΦΟΙΤΗΤΕΣ (ΑΕΜ)
```

```
ALTER TABLE ΒΑΘΜΟΛΟΓΙΕΣ ADD CONSTRAINT FK2 FOREIGN KEY
ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ REFERENCES ΜΑΘΗΜΑΤΑ (ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ)
```

```
ALTER TABLE ΜΑΘΗΜΑΤΑ ADD CONSTRAINT FK3 FOREIGN KEY
ΚΩΔΙΚΟΣ_ΚΑΘΗΓΗΤΗ REFERENCES ΚΑΘΗΓΗΤΗΣ (ΚΩΔΙΚΟΣ_ΚΑΘΗΓΗΤΗ) ;
```

## ΕΡΩΤΗΜΑΤΑ

### Ερώτημα 1

Προσθέστε ένα πεδίο στον πίνακα ΜΑΘΗΜΑΤΑ με όνομα ΒΑΘΜΟΣ\_ΔΥΣΚΟΛΙΑΣ.

### Απάντηση 1

```
ALTER TABLE ΜΑΘΗΜΑΤΑ ADD COLUMN ΒΑΘΜΟΣ_ΔΥΣΚΟΛΙΑΣ  
INTEGER (2) ;
```

### Ερώτημα 2

Αλλάξτε τον τύπο δεδομένων του πεδίου ΒΑΘΜΟΣ\_ΔΥΣΚΟΛΙΑΣ του πίνακα ΜΑΘΗΜΑΤΑ από INTEGER σε VARCHAR().

### Απάντηση 2

```
ALTER TABLE ΜΑΘΗΜΑΤΑ ALTER COLUMN ΒΑΘΜΟΣ_ΔΥΣΚΟΛΙΑΣ  
VARCHAR (20) ;
```

### Ερώτημα 3

Διαγράψτε το πεδίο του πίνακα ΜΑΘΗΜΑΤΑ με όνομα ΒΑΘΜΟΣ\_ΔΥΣΚΟΛΙΑΣ.

### Απάντηση 3

```
ALTER TABLE ΜΑΘΗΜΑΤΑ DROP COLUMN ΒΑΘΜΟΣ_ΔΥΣΚΟΛΙΑΣ;
```

### Ερώτημα 4

Διαγράψτε τον περιορισμό FK1 του πίνακα ΒΑΘΜΟΛΟΓΙΕΣ.

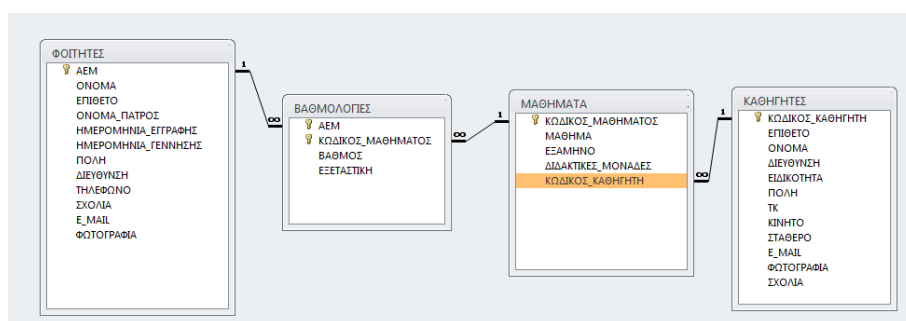
### Απάντηση 4

```
ALTER TABLE ΒΑΘΜΟΛΟΓΙΕΣ DROP CONSTRAINT FK1;
```

Ομοίως δοκιμάστε μόνοι σας να εκτελέσετε τα ίδια ερωτήματα με πρόσθετο πεδίο στον πίνακα ΒΑΘΜΟΛΟΓΙΕΣ το ΑΠΟΤΕΛΕΣΜΑ(ΕΠΙΤΥΧΙΑ-ΑΠΟΤΥΧΙΑ) με τύπο δεδομένων Ναι/Όχι

## Άσκηση 7 – Σύζευξη πινάκων

Έχετε μάθει τον τρόπο χειρισμού ερωτημάτων όταν αυτά αφορούν έναν μόνο πίνακα τι θα γινόταν όμως αν αφορούσαν παραπάνω από έναν. Την λύση σε αυτό το ζήτημα έδωσε η σύζευξη των πινάκων που είναι μία σύνθετη πράξη. Στην ουσία η λογική της σύζευξης είναι η εξής τα κοινά πεδία ίσα εφόσον οι δύο πίνακες συνδέονται μεταξύ τους με κοινά πεδία τιμών. Σαν αποτέλεσμα επιστρέφει τα πεδία και των δύο αρχικών πινάκων και τις εγγραφές στις οποίες οι τιμές του κοινού τους πεδίου σύμφωνα με το οποίο γίνεται η σύζευξη είναι ίσες. Αυτή είναι η έννοια της εσωτερικής σύζευξης **INNER JOIN** δύο πινάκων. Η σύζευξη έχει έτσι και δικό της τελεστή **JOIN**. Υπάρχουν τέσσερις τύποι σύζευξης: εσωτερική (όπως η παρακάτω), αριστερή, δεξιά και πλήρης στην ύλη σας θα είναι μόνο η εσωτερική σύζευξη. Με την Εικόνα 62 θα σας είναι ευκολότερη η σύνταξη των ερωτημάτων σύζευξης.



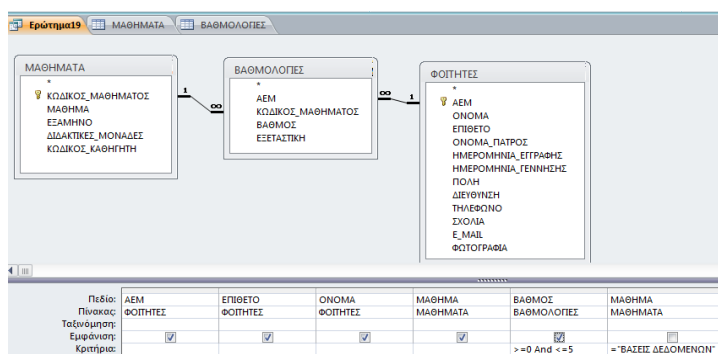
Εικόνα 63

### Παράδειγμα 1

Εμφανίστε με SQL το ΑΕΜ, το ΕΠΙΘΕΤΟ, το ΟΝΟΜΑ, το ΜΑΘΗΜΑ και τον ΒΑΘΜΟ αυτών που ΔΕΝ πέρασαν το μάθημα ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ και ο βαθμός τους είναι μεγαλύτερος ή ίσος του μηδενός και μικρότερος ή ίσος του πέντε.

### Απάντηση 1

**α. Με QBE(Query by Example): Προσθέστε** τους πίνακες, σύρετε τα πεδία που σας ενδιαφέρουν, βάζετε τους περιορισμούς στην γραμμή κριτήρια και τέλος εκτελείτε το ερώτημα πατώντας το κουμπί .



Εικόνα 64



**β. Με κώδικα SQL:** πιάστε το κουμπί προβολή SQL και γράψτε τον παρακάτω κώδικα τέλος πιάστε το κουμπί εκτέλεση .

```
SELECT ΦΟΙΤΗΤΕΣ.ΑΕΜ, ΦΟΙΤΗΤΕΣ.ΕΠΙΘΕΤΟ, ΦΟΙΤΗΤΕΣ.ΟΝΟΜΑ,  
ΜΑΘΗΜΑΤΑ.ΜΑΘΗΜΑ, ΒΑΘΜΟΛΟΓΙΕΣ.ΒΑΘΜΟΣ  
  
FROM ΦΟΙΤΗΤΕΣ, ΜΑΘΗΜΑΤΑ, ΒΑΘΜΟΛΟΓΙΕΣ  
  
WHERE ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ =  
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ AND ΦΟΙΤΗΤΕΣ.ΑΕΜ =  
ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND ΒΑΘΜΟΛΟΓΙΕΣ.ΒΑΘΜΟΣ>=0 AND  
ΒΑΘΜΟΛΟΓΙΕΣ.ΒΑΘΜΟΣ<=5 AND ΜΑΘΗΜΑΤΑ.ΜΑΘΗΜΑ="ΒΑΣΕΙΣ  
ΔΕΔΟΜΕΝΩΝ";
```

## Παράδειγμα 2

Εμφανίστε με SQL το ΟΝΟΜΑ το ΕΠΙΘΕΤΟ το ΕΞΑΜΗΝΟ και τον ΒΑΘΜΟ των φοιτητών.

**ΣΧΟΛΙΑ!** Παρατηρήστε από την Εικόνα 61 ότι το ΟΝΟΜΑ και το ΕΠΙΘΕΤΟ βρίσκονται στον πίνακα ΦΟΙΤΗΤΕΣ ενώ ο ΒΑΘΜΟΣ στον πίνακα ΒΑΘΜΟΛΟΓΙΕΣ. Παρόλα αυτά όμως ο πίνακας ΦΟΙΤΗΤΕΣ συνδέεται με τον πίνακα ΒΑΘΜΟΛΟΓΙΕΣ μέσω του πρωτεύον κλειδιού ΑΕΜ.

## Απάντηση 2

```
SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ  
  
FROM ΦΟΙΤΗΤΕΣ, ΒΑΘΜΟΛΟΓΙΕΣ, ΜΑΘΗΜΑΤΑ  
  
WHERE ΦΟΙΤΗΤΕΣ.ΑΕΜ=ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND  
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ=ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ;
```

ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΜΑΘΗΜΑ	ΒΑΘΜΟΣ
ΕΛΕΝΗ	ΓΕΩΡΓΙΟΥ	ΦΥΣΙΚΗ	6
ΕΛΕΝΗ	ΓΕΩΡΓΙΟΥ	ΜΑΘΗΜΑΤΙΚΑ	7
ΜΑΡΙΑ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΑΓΓΛΙΚΑ	1
ΜΑΡΙΑ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΦΥΣΙΚΗ	5
ΚΩΣΤΑΣ	ΣΤΑΜΟΣ	ΜΑΘΗΜΑΤΙΚΑ	3
ΜΑΡΙΑ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΜΑΘΗΜΑΤΙΚΑ	5
ΕΛΕΝΗ	ΓΕΩΡΓΙΟΥ	ΑΓΓΛΙΚΑ	6
ΚΩΣΤΑΣ	ΣΤΑΜΟΣ	ΦΥΣΙΚΗ	7
ΚΩΣΤΑΣ	ΣΤΑΜΟΣ	ΑΓΓΛΙΚΑ	8
ΚΩΣΤΑΣ	ΣΤΑΜΟΣ	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	6
ΕΛΕΝΗ	ΓΕΩΡΓΙΟΥ	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	5
ΜΑΡΙΑ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	4
ΚΩΣΤΑΣ	ΣΤΑΜΟΣ	ΣΤΑΤΙΣΤΙΚΗ	10
ΜΑΡΙΑ	ΠΑΠΑΔΟΠΟΥΛΟΥ	ΣΤΑΤΙΣΤΙΚΗ	5
ΕΛΕΝΗ	ΓΕΩΡΓΙΟΥ	ΣΤΑΤΙΣΤΙΚΗ	3

Εικόνα 65

### **Παράδειγμα 3**

Ποιο μάθημα διδάσκει ο καθηγητής ΠΑΠΑΔΟΠΟΥΛΟΣ;

### **Απάντηση 3**

```
SELECT ΜΑΘΗΜΑ, ΕΠΙΘΕΤΟ
FROM ΜΑΘΗΜΑΤΑ, ΚΑΘΗΓΗΤΕΣ
WHERE ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ= ΚΑΘΗΓΗΤΕΣ.ΚΩΔΙΚΟΣ_
ΜΑΘΗΜΑΤΟΣ AND ΕΠΙΘΕΤΟ = 'ΠΑΠΑΔΟΠΟΥΛΟΣ';
```

### **Παράδειγμα 4**

Εμφανίστε με SQL το ΜΑΘΗΜΑ, τις ΔΙΔΑΚΤΙΚΕΣ ΜΟΝΑΔΕΣ, το ΕΞΑΜΗΝΟ, το ΕΠΙΘΕΤΟ και το ΟΝΟΜΑ του καθηγητή που το κάνει.

### **Απάντηση 4**

```
SELECT ΜΑΘΗΜΑ, ΔΙΔΑΚΤΙΚΕΣ_ΜΟΝΑΔΕΣ, ΕΞΑΜΗΝΟ, ΕΠΙΘΕΤΟ,
ΟΝΟΜΑ
FROM ΜΑΘΗΜΑΤΑ, ΚΑΘΗΓΗΤΕΣ
WHERE
ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΚΑΘΗΓΗΤΗ=ΚΑΘΗΓΗΤΕΣ.ΚΩΔΙΚΟΣ_ΚΑΘΗΓΗΤΗ;
```

### **Παράδειγμα 5**

Εμφανίστε με SQL το ΕΠΙΘΕΤΟ και το ΟΝΟΜΑ των καθηγητών που διδάσκουν το μάθημα ΜΑΘΗΜΑΤΙΚΑ και τους βαθμούς του μαθήματος.

### **Απάντηση 5**

```
SELECT ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΒΑΘΜΟΣ
FROM ΚΑΘΗΓΗΤΕΣ, ΒΑΘΜΟΛΟΓΙΕΣ, ΜΑΘΗΜΑΤΑ
WHERE ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ=
ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ AND
ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΚΑΘΗΓΗΤΗ=ΚΑΘΗΓΗΤΕΣ.ΚΩΔΙΚΟΣ_ΚΑΘΗΓΗΤΗ AND
ΜΑΘΗΜΑ='ΜΑΘΗΜΑΤΙΚΑ';
```

### **Παράδειγμα 6**

Εμφανίστε με SQL αναλυτικά τις βαθμολογίες του κάθε φοιτητή δηλαδή το ΑΕΜ,ΕΠΙΘΕΤΟ, ΟΝΟΜΑ\_ ΠΑΤΡΟΣ, ΗΜΕΡΟΜΗΝΙΑ ΕΓΓΡΑΦΗΣ , ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ, ΠΟΛΗ, ΤΗΛΕΦΩΝΟ, ΔΙΕΥΘΥΝΣΗ, ΣΧΟΛΙΑ, EMAIL και την ΦΩΤΟΓΡΑΦΙΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ ταξινομημένα κατά ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΜΑΘΗΜΑ.

### **Απάντηση 6**

```
SELECT ΕΠΙΘΕΤΟ, ΟΝΟΜΑ_ΠΑΤΡΟΣ, ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ,  
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ, ΠΟΛΗ, ΤΗΛΕΦΩΝΟ, ΔΙΕΥΘΥΝΣΗ, ΣΧΟΛΙΑ,  
EMAIL, ΦΩΤΟΓΡΑΦΙΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ  
  
FROM ΦΟΙΤΗΤΕΣ, ΜΑΘΗΜΑΤΑ, ΒΑΘΜΟΛΟΓΙΕΣ  
  
WHERE ΦΟΙΤΗΤΕΣ.ΑΕΜ=ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND  
  
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ=ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜ  
ΑΤΟΣ ORDER BY ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΜΑΘΗΜΑ;
```

### **Παράδειγμα 7**

Εμφανίστε με SQL αναλυτικά τις βαθμολογίες του κάθε φοιτητή δηλαδή το ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ\_ ΠΑΤΡΟΣ, ΗΜΕΡΟΜΗΝΙΑ\_ΕΓΓΡΑΦΗΣ, ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ, ΠΟΛΗ, ΤΗΛΕΦΩΝΟ, ΔΙΕΥΘΥΝΣΗ, ΣΧΟΛΙΑ, EMAIL και την ΦΩΤΟΓΡΑΦΙΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ για αυτούς που έχουν βαθμό μεγαλύτερο ή ίσο του πέντε.

### **Απάντηση 7**

```
SELECT ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΟΝΟΜΑ_ΠΑΤΡΟΣ, ΗΜΕΡΟΜΗΝΙΑ_ΕΓΓΡΑΦΗΣ,  
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ, ΠΟΛΗ, ΤΗΛΕΦΩΝΟ, ΔΙΕΥΘΥΝΣΗ, ΣΧΟΛΙΑ,  
EMAIL, ΦΩΤΟΓΡΑΦΙΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ  
  
FROM ΦΟΙΤΗΤΕΣ, ΜΑΘΗΜΑΤΑ, ΒΑΘΜΟΛΟΓΙΕΣ  
  
WHERE ΦΟΙΤΗΤΕΣ.ΑΕΜ=ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND  
  
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ=ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜ  
ΑΤΟΣ AND ΒΑΘΜΟΣ>=5;
```

### **Παράδειγμα 8**

Εμφανίστε με SQL το ΑΕΜ, το ΟΝΟΜΑ και το ΕΠΙΘΕΤΟ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ των φοιτητών που πέρασαν τα μαθήματα ΜΑΘΗΜΑΤΙΚΑ, ΑΓΓΛΙΚΑ, ΣΤΑΤΙΣΤΙΚΗ.

#### **α. Απάντηση 8**

```
SELECT ΦΟΙΤΗΤΕΣ.ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ
FROM ΦΟΙΤΗΤΕΣ, ΜΑΘΗΜΑΤΑ, ΒΑΘΜΟΛΟΓΙΕΣ
WHERE          ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ          =
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ          AND          ΦΟΙΤΗΤΕΣ.ΑΕΜ          =
ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND ΒΑΘΜΟΛΟΓΙΕΣ.ΒΑΘΜΟΣ>=5 AND ΜΑΘΗΜΑ IN
("ΜΑΘΗΜΑΤΙΚΑ", "ΑΓΓΛΙΚΑ", "ΣΤΑΤΙΣΤΙΚΗ");
```

#### **β. Απάντηση 8**

```
SELECT ΦΟΙΤΗΤΕΣ.ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ
FROM ΦΟΙΤΗΤΕΣ, ΜΑΘΗΜΑΤΑ, ΒΑΘΜΟΛΟΓΙΕΣ
WHERE          ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ          =
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ          AND          ΦΟΙΤΗΤΕΣ.ΑΕΜ          =
ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND ΒΑΘΜΟΛΟΓΙΕΣ.ΒΑΘΜΟΣ>=5 AND ΜΑΘΗΜΑ
="ΜΑΘΗΜΑΤΙΚΑ" OR ΜΑΘΗΜΑ ="ΑΓΓΛΙΚΑ" OR ΜΑΘΗΜΑ
="ΣΤΑΤΙΣΤΙΚΗ";
```

Απαντήστε με κώδικα SQL στα παρακάτω ερωτήματα:

### **Παράδειγμα 9**

Εμφανίστε με SQL το ΑΕΜ, το ΟΝΟΜΑ και το ΕΠΙΘΕΤΟ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ των φοιτητών που πέρασαν τα μαθήματα και είναι από τις πόλεις Σέρρες και Αθήνα .

### **Παράδειγμα 10**

Εμφανίστε με SQL το ΚΩΔΙΚΟ\_ΚΑΘΗΓΗΤΗ, το ΕΠΙΘΕΤΟ και το ΟΝΟΜΑ, ΜΑΘΗΜΑ, ΕΙΔΙΚΟΤΗΤΑ των καθηγητών που είναι από Σέρρες και Αθήνα.

### **Παράδειγμα 11**

Εμφανίστε με SQL το ΟΝΟΜΑ, το ΕΠΙΘΕΤΟ, ΜΑΘΗΜΑ και τον ΒΑΘΜΟ των φοιτητών που πέρασαν τα μαθήματα ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ, ΦΥΣΙΚΗ, ΒΙΟΜΗΧΑΝΙΚΗ.

### **Παράδειγμα 12**

Ποιο μάθημα διδάσκει ο καθηγητής ΠΑΠΑΣ και ποιοι μαθητές το έχουν περάσει;

### **Παράδειγμα 13**

Εμφανίστε με SQL το ΑΕΜ, το ΕΠΙΘΕΤΟ, το ΟΝΟΜΑ, το ΜΑΘΗΜΑ και τον ΒΑΘΜΟ αυτών που πέρασαν το μάθημα ΑΓΓΛΙΚΑ και ο βαθμός τους είναι μεγαλύτερος ή ίσος του πέντε και μεγαλύτερος ή ίσος του δέκα.

### **Παράδειγμα 14**

Εμφανίστε με SQL αναλυτικά τις βαθμολογίες του κάθε φοιτητή δηλαδή το ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ\_ ΠΑΤΡΟΣ, ΗΜΕΡΟΜΗΝΙΑ\_ΕΓΓΡΑΦΗΣ, ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ, ΠΟΛΗ, ΤΗΛΕΦΩΝΟ, ΔΙΕΥΘΥΝΣΗ, ΣΧΟΛΙΑ, EMAIL και την ΦΩΤΟΓΡΑΦΙΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ για αυτούς που ΔΕΝ έχουν βαθμό μεγαλύτερο ή ίσο του πέντε.

### **Παράδειγμα 15**

Εμφανίστε με SQL το ΟΝΟΜΑ το ΕΠΙΘΕΤΟ, την ΠΟΛΗ και την ΦΩΤΟΓΡΑΦΙΑ των καθηγητών που το επίθετο τους ξεκινάει από Π.



## Άσκηση 8 – Συναθροιστικές συναρτήσεις σε Access

Στο εργαστήριο αυτό θα ασχοληθείτε με τις συναθροιστικές συναρτήσεις δηλαδή με τις συναρτήσεις **max** (εύρεση μεγαλύτερης τιμής), **min** ( εύρεση μικρότερης τιμής), **avg** (εύρεση μέσου όρου) , **count** (εύρεση πλήθους εγγραφών), **sum** (εύρεση αθροίσματος). Παρακάτω θα δείτε παραδείγματα αυτών των συναρτήσεων.

Η γενική σύνταξη των εντολών για συναθροιστικές συναρτήσεις είναι:

```
SELECT ΠΕΔΙΑ  
FROM ΠΙΝΑΚΕΣ  
WHERE ΠΕΡΙΟΡΙΣΜΟΣ  
GROUP BY ΟΜΑΔΟΠΟΙΗΣΗ  
HAVING ΣΥΝΑΘΡΟΙΣΤΙΚΟΣ_ΠΕΡΙΟΡΙΣΜΟΣ;
```

### Παράδειγμα 1

Εμφανίστε με SQL το πλήθος των φοιτητών που κατάγονται από τη πόλη ΣΕΡΡΕΣ.

### Απάντηση 1

```
SELECT COUNT (ΑΕΜ) AS ΠΛΗΘΟΣ  
FROM ΦΟΙΤΗΤΕΣ  
WHERE ΠΟΛΗ='ΣΕΡΡΕΣ' ;
```

### Παράδειγμα 2

Εμφανίστε με SQL το πλήθος των φοιτητών για κάθε πόλη.

### Απάντηση 2

```
SELECT ΠΟΛΗ, COUNT (ΑΕΜ) AS ΠΛΗΘΟΣ  
FROM ΦΟΙΤΗΤΕΣ  
GROUP BY ΠΟΛΗ;
```

### **Παράδειγμα 3**

Εμφανίστε με SQL το πλήθος των φοιτητών για κάθε πόλη που το πλήθος τους υπερβαίνει του δύο.

### **Απάντηση 3**

```
SELECT ΠΟΛΗ, COUNT (ΑΕΜ) AS ΠΛΗΘΟΣ  
FROM ΦΟΙΤΗΤΕΣ  
GROUP BY ΠΟΛΗ;  
HAVING ΠΛΗΘΟΣ>=2;
```

### **Παράδειγμα 4**

Εμφανίστε με SQL το πλήθος των μαθημάτων που πέρασε ο κάθε φοιτητής.

### **Απάντηση 4**

```
SELECT ΑΕΜ, COUNT (ΚΜ) AS ΠΛΗΘΟΣ  
FROM ΒΑΘΜΟΛΟΓΙΕΣ  
WHERE ΒΑΘΜΟΣ >=5  
GROYP BY ΑΕΜ;
```

### **Παράδειγμα 5**

Εμφανίστε με SQL το μέσο όρο της βαθμολογίας του κάθε φοιτητή.

### **Απάντηση 5**

```
SELECT ΑΕΜ, AVG (ΒΑΘΜΟΣ) AS ΜΟ  
FROM ΒΑΘΜΟΛΟΓΙΕΣ  
GROYP BY ΑΕΜ;
```

### **Παράδειγμα 6**

Εμφανίστε με SQL το μέσο όρο της βαθμολογίας του κάθε φοιτητή για τα μαθήματα που πέρασε με μέσο όρο μεγαλύτερο από οχτώ και βαθμό μεγαλύτερο ή ίσο από πέντε.

### **Απάντηση 6**

```
SELECT AEM, AVG (ΒΑΘΜΟΣ) AS ΜΟ
FROM ΒΑΘΜΟΛΟΓΙΕΣ
WHERE ΒΑΘΜΟΣ >=5
HAVING ΜΟ>=8;
```

**ΣΧΟΛΙΟ!** Ο περιορισμός για το αποτέλεσμα στις συναθροίστηκες συναρτήσεις μπαίνει πάντα στο HAVING ποτέ στο WHERE.

### **Παράδειγμα 7**

Εμφανίστε με SQL το μέγιστο βαθμό που γράψανε για το μάθημα ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.

### **Απάντηση 7**

```
SELECT MAX (ΒΑΘΜΟΣ) AS ΜΕΓΙΣΤΟΣ
FROM ΒΑΘΜΟΛΟΓΙΕΣ, ΜΑΘΗΜΑΤΑ
WHERE
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ=ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ
AND ΜΑΘΗΜΑ='ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ' ;
```

### **Παράδειγμα 8**

Εμφανίστε με SQL πόσοι φοιτητές δώσανε σε κάθε εξεταστική.

### **Απάντηση 8**

```
SELECT ΕΞΕΤΑΣΤΙΚΗ, COUNT (ΑΕΜ) AS ΠΛΗΘΟΣ
FROM ΒΑΘΜΟΛΟΓΙΕΣ
GROUP BY ΕΞΕΤΑΣΤΙΚΗ;
```

### **Παράδειγμα 9**

Εμφανίστε με SQL πόσοι φοιτητές γράψανε δέκα σε κάθε εξεταστική.

### **Απάντηση 9**

```
SELECT ΕΞΕΤΑΣΤΙΚΗ, COUNT (ΑΕΜ) AS ΠΛΗΘΟΣ  
FROM ΒΑΘΜΟΛΟΓΙΕΣ  
WHERE ΒΑΘΜΟΣ=10  
GROUP BY ΕΞΕΤΑΣΤΙΚΗ;
```

### **Παράδειγμα 10(Με εμφωλευμένα ερωτήματα)**

Εμφανίστε με SQL τους φοιτητές που δεν δώσανε κανένα μάθημα στην εξεταστική ΧΕΙΜ2010-2011.

### **Απάντηση 10**

```
SELECT *  
FROM ΦΟΙΤΗΤΕΣ  
WHERE ΑΕΜ NOT IN (SELECT ΑΕΜ FROM ΒΑΘΜΟΛΟΓΙΕΣ  
WHERE ΕΞΕΤΑΣΤΙΚΗ=' ΧΕΙΜ2010-2011' );
```

### **Παράδειγμα 11(Με εμφωλευμένα ερωτήματα)**

Εμφανίστε με SQL τους φοιτητές που δώσανε μαθήματα στην εξεταστική ΧΕΙΜ2010-2011.

### **α. Απάντηση 11**

```
SELECT *  
FROM ΦΟΙΤΗΤΕΣ  
WHERE ΑΕΜ IN (SELECT ΑΕΜ FROM ΒΑΘΜΟΛΟΓΙΕΣ  
WHERE ΕΞΕΤΑΣΤΙΚΗ=' ΧΕΙΜ2010-2011' );
```

### **β. Απάντηση 11**

```
SELECT ΦΟΙΤΗΤΕΣ . *  
FROM ΦΟΙΤΗΤΕΣ, ΒΑΘΜΟΛΟΓΙΕΣ  
WHERE ΦΟΙΤΗΤΕΣ.ΑΕΜ=ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ  
AND ΕΞΕΤΑΣΤΙΚΗ=ΧΕΙΜ2010-2011);
```

### **Παράδειγμα 12**

Εμφανίστε με SQL το μέσο όρο της βαθμολογίας του κάθε φοιτητή για τα μαθήματα που πέρασε με μέσο όρο μικρότερο από οχτώ και βαθμό μεγαλύτερο ή ίσο από πέντε.

### **Παράδειγμα 13**

Εμφανίστε με SQL τον ελάχιστο βαθμό που γράψανε για το μάθημα ΜΑΘΗΜΑΤΙΚΑ.

### **Παράδειγμα 14**

Εμφανίστε με SQL πόσοι φοιτητές γράψανε πέντε σε κάθε εξεταστική

### **Παράδειγμα 15**

Εμφανίστε με SQL και με του δύο τρόπους τους φοιτητές που δώσανε μαθήματα στην εξεταστική ΕΑΡ2008-2009.

## Άσκηση 9 – Εργαζόμενοι – Τμήματα - Έργα, ΒΔ σε SQL Server

Σκοπός της εργαστηριακής άσκησης αυτής είναι η εισαγωγή και η εξοικείωση σας στον **SQL Server 2008** με την άσκηση εργαζόμενοι- τμήματα- έργα.

Το περιβάλλον θα χρησιμοποιήσετε για τη δημιουργία και την διαχείριση της βάσης λέγεται SQL Server Management Studio. Για να το ανοίξετε πάτε Έναρξη → Όλα τα προγράμματα → Microsoft SQL Server 2008 → SQL Server Management Studio.

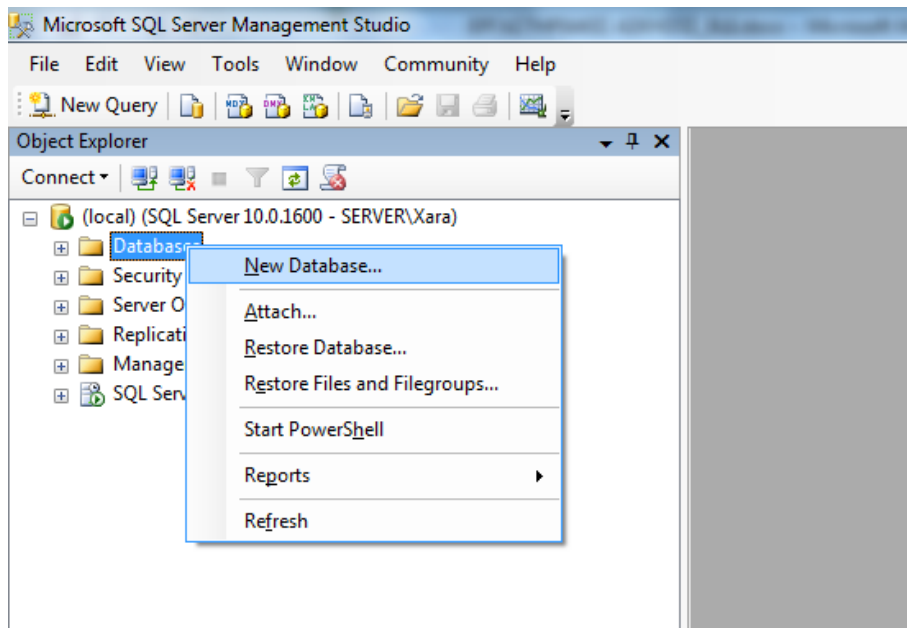


Αφού μπείτε στο περιβάλλον θα σας ανοίξει η παρακάτω Εικόνα στην οποία πρέπει να γίνει η σύνδεση με την υπηρεσία του SQL Server. Πατώντας **Connect** αρχίζει η υπηρεσία του SQL Server και μπορείτε τώρα να δημιουργήσετε την δική σας βάση.



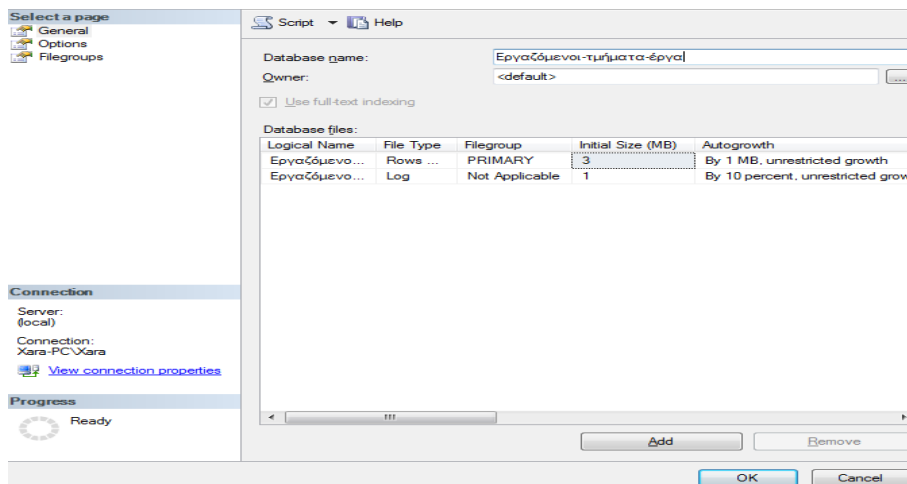
Εικόνα 66

- Για την δημιουργία μίας νέας βάσης πηγαίνετε στον **Object Explorer** → **Databases** → δεξί κλικ → **New Database**.



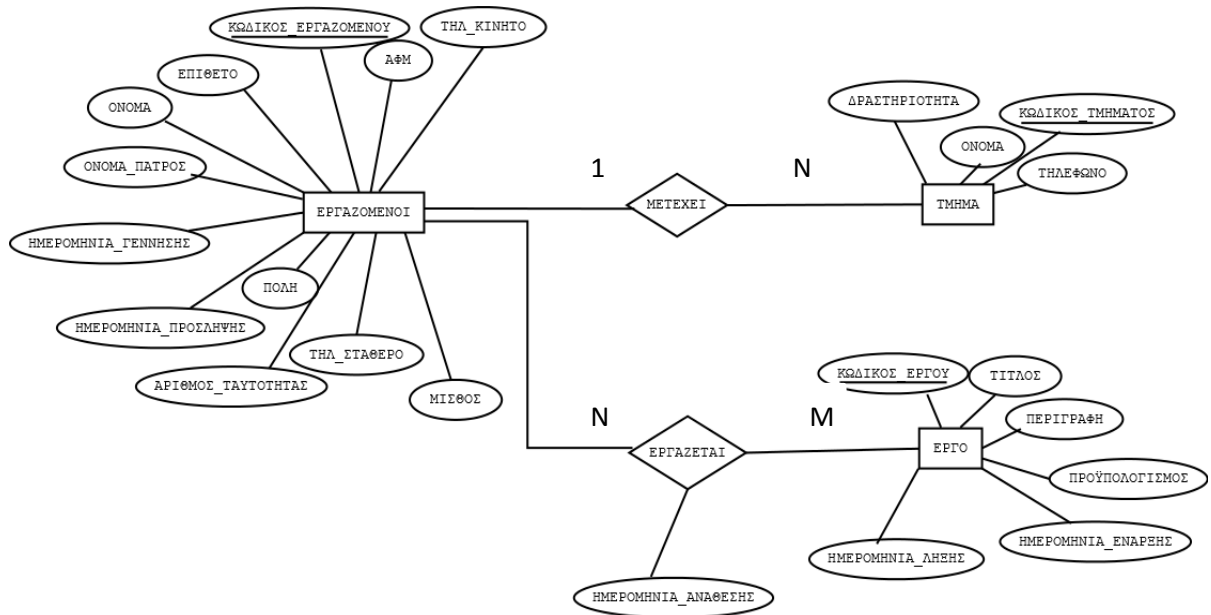
Εικόνα 67

- Δώστε όνομα στην βάση (πχ ΕργαζόμενοιΒΔ) και έπειτα **OK**.



Εικόνα 68

➤ Εισάγετε τους πίνακες και τα πεδία σύμφωνα με το σχήμα :



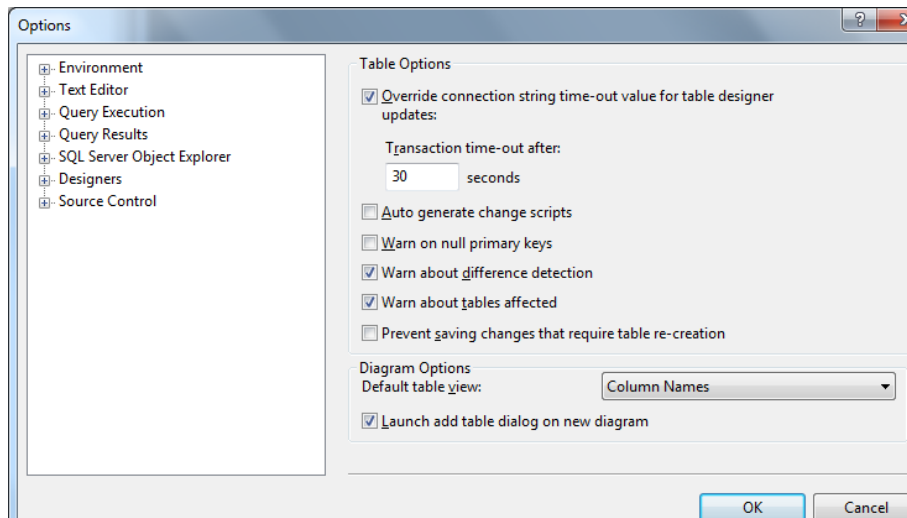
Σχήμα 2

Παρατηρήστε στο Σχήμα 2 τις συσχετίσεις των πινάκων. Συμφώνα λοιπόν με τις συσχετίσεις αυτές προκύπτουν τέσσερις πίνακες (ΕΡΓΑΖΟΜΕΝΟΙ, ΤΜΗΜΑ, ΕΡΓΑΖΕΤΑΙ, ΕΡΓΟ). Ο πίνακας ΕΡΓΑΖΕΤΑΙ δημιουργείται λόγω της συσχέτισης πολλά-προς-πολλά του πίνακα ΕΡΓΑΖΟΜΕΝΟΙ. Ενώ η συσχέτιση του πίνακα ΕΡΓΑΖΟΜΕΝΟΙ με τον πίνακα ΤΜΗΜΑ είναι ένα-προς-πολλά και λόγω αυτής της συσχέτισης δημιουργείται ένα ξένο κλειδί στην μεριά των εργαζομένων το ΚΩΔΙΚΟΣ\_ΤΜΗΜΑΤΟΣ.

\*\*\*\*\*Πριν ξεκινήσετε την εισαγωγή των πεδίων και των αντίστοιχων τύπων δεδομένων θα πρέπει να απενεργοποιήσετε μία επιλογή η οποία θα σας δυσκολέψει στην πορεία σε περίπτωση που θελήσετε να κάνετε κάποια αλλαγή στη δομή των πινάκων. Για να απενεργοποιήσετε αυτή την επιλογή ακολουθήστε τα παρακάτω βήματα.\*\*\*\*\*

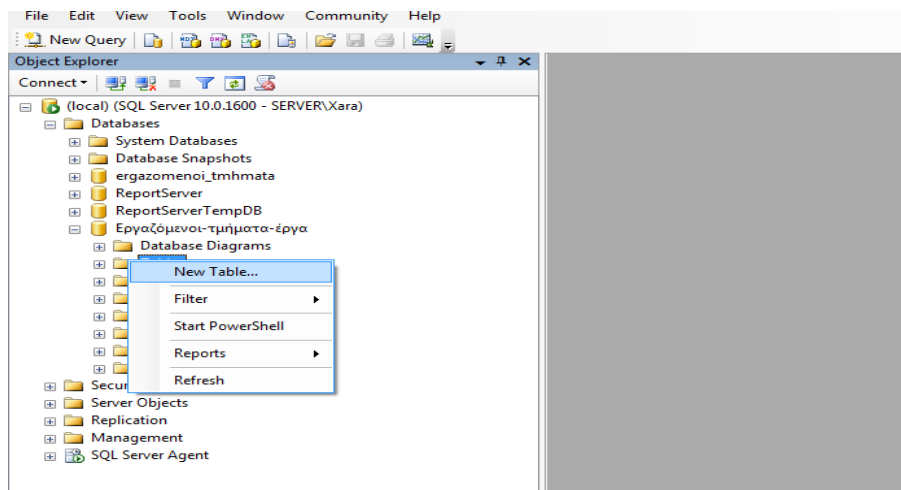
Πηγαίνετε στην κορδέλα **Tools→Options** και από-επιλέγετε το **Prevent saving changes that require table re-creation**.





Εικόνα 69

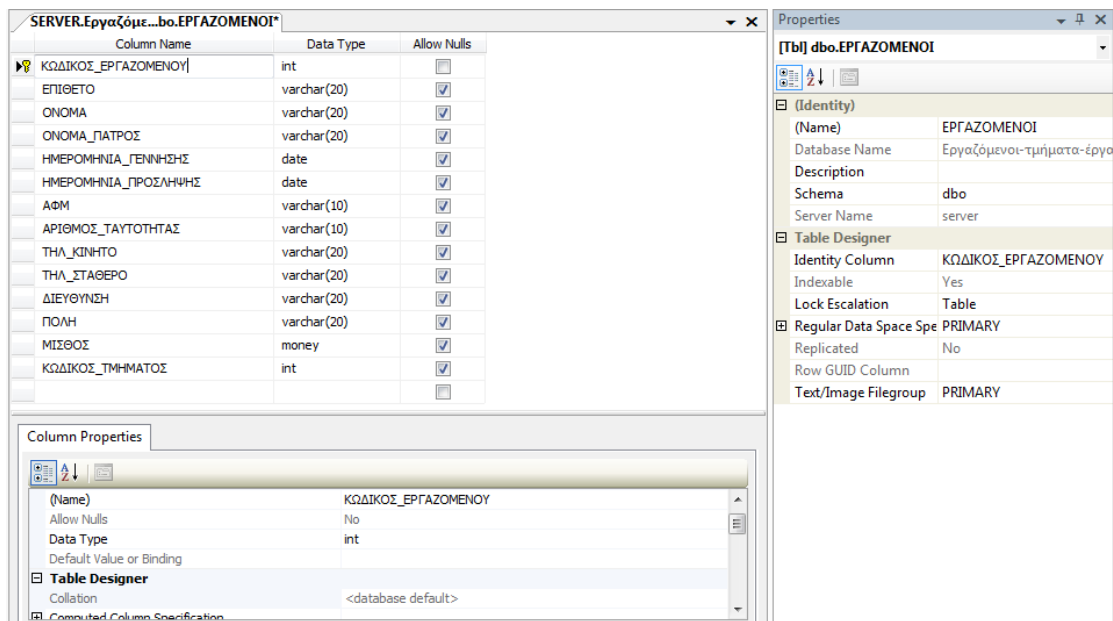
Για την δημιουργία ενός πίνακα πηγαίνετε **Object Explorer** → ανοίξτε την δενδρική δομή του **Databases** → ομοίως για την βάση Εργαζόμενοι-τμήματα-έργα → Τέλος **New Table**.



Εικόνα 70

- Έπειτα δίνετε ένα όνομα στο Table από τον πίνακα δεξιά στο Properties (ΕΡΓΑΖΟΜΕΝΟΙ). Στη συνέχεια εισάγετε τα πεδία με τον αντίστοιχο τύπο δεδομένων όπως βλέπετε στην Εικόνα 69.

Τα πεδία που θα εισάγεται είναι τα εξής: ΚΩΔΙΚΟΣ ΕΡΓΑΖΟΜΕΝΟΥ(INT), ΕΠΙΘΕΤΟ(VARCHAR (30)), ΟΝΟΜΑ(VARCHAR (30)), ΟΝΟΜΑ\_ΠΑΤΡΟΣ(VARCHAR (30)), ΗΜΕΡΟΜΗΝΙΑ\_ΓΕΝΝΗΣΗΣ(DATE), ΗΜΕΡΟΜΗΝΙΑ\_ΠΡΟΣΛΗΨΗΣ(DATE), ΑΦΜ(VARCHAR (30)), ΑΡΙΘΜΟΣ\_ΤΑΥΤΟΤΗΤΑΣ(VARCHAR (30)), ΤΗΛ\_ΚΙΝΗΤΟ(VARCHAR (30)), ΤΗΛ\_ΣΤΑΘΕΡΟ(VARCHAR (30)), ΔΙΕΥΘΥΝΣΗ(VARCHAR (30)), ΠΟΛΗ(VARCHAR (30)), ΜΙΣΘΟΣ(MONEY) και **ξένο κλειδί** το ΚΩΔΙΚΟΣ\_ΤΜΗΜΑΤΟΣ(INT).

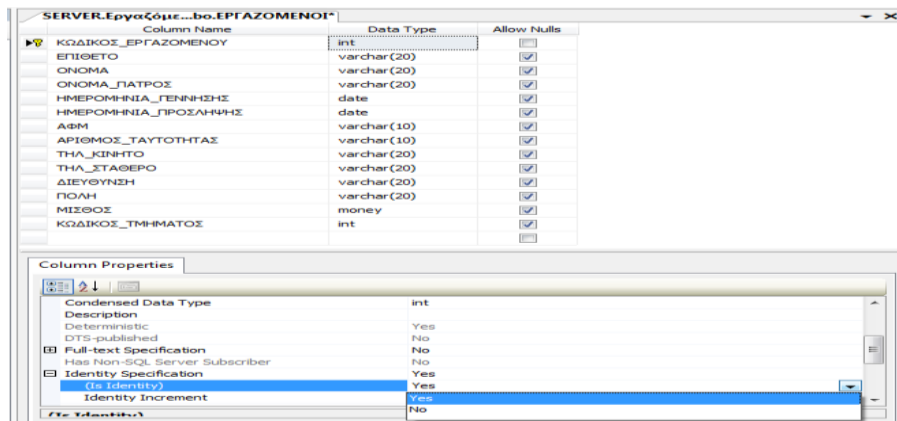


Εικόνα 71

**ΣΧΟΛΙΟ!** Παρατηρήστε ότι στα πεδία ΤΗΛ\_ΚΙΝΗΤΟ και ΤΗΛ\_ΣΤΑΘΕΡΟ έχει τύπο δεδομένων Varchar() παρόλο που τα πεδία αυτά νομίζετε ότι δέχονται ακεραίους ως τιμές και θα έπρεπε να έχουν τύπο δεδομένων int. Αυτό συμβαίνει γιατί τα πεδία αυτά δεν θα τα χρησιμοποιήσετε για εύρεση μέσου όρου, αθροίσματος και γενικά για αριθμητικές πράξεις οπότε ως int δεν θα σας χρειαστούν και επειδή ίσως χρειαστεί τα τηλέφωνα να ξεκινάνε με 00 ή +49 κλπ.

Για την εισαγωγή του πρωτεύοντος κλειδιού το οποίο στον πίνακα εργαζόμενοι είναι ο κωδικός εργαζομένου κάνετε το εξής:

- Δεξιά του πεδίου του οποίου θέλετε να γίνει πρωτεύον κλειδί πατάτε **δεξί κλικ** → **Πρωτεύον κλειδί**. Έπειτα πηγαίνετε **κάτω στο Column Properties** και ανοίγετε την δενδρική δομή του **Identity Specification** και στο **(Is identity)** κάνετε την τιμή από **No** σε **Yes** και το **Identity Increment=1** για να έχετε αυτόματη αρίθμηση. Στον πίνακα εργαζόμενοι έχετε ένα πρωτεύον κλειδί τον ΚΩΔΙΚΟ\_ΕΡΓΑΖΟΜΕΝΟΥ και ένα ξένο το ΚΩΔΙΚΟ\_ΤΜΗΜΑΤΟΣ. Το ξένο κλειδί θα το μαρκάρετε μαζί με το πρωτεύον ώστε να τοποθετήσετε την ένδειξη κλειδί αλλά μόνο το ΚΩΔΙΚΟΣ\_ΕΡΓΑΖΟΜΕΝΟΥ θα έχει αυτόματη αρίθμηση.



Εικόνα 72

Για να εισάγετε εγγραφές στον πίνακα εργαζόμενοι πηγαίνετε στο **Object explorer** στον πίνακα ΕΡΓΑΖΟΜΕΝΟΙ **δεξί κλικ**→**Edit top 200 rows**, έπειτα ανοίγει το παράθυρο για να εισάγετε τις εγγραφές.

Οι εγγραφές που θα περιέχει ο πίνακας ΕΡΓΑΖΟΜΕΝΟΙ θα είναι οι παρακάτω:

ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΠΡΟΣΛΗΨΗΣ	ΑΦΜ	ΑΡΙΘΜΟΣ_ΤΑΥΤΟΤΗΤΑΣ
ΠΑΠΑΔΟΠΟΥΛΟΣ	ΓΙΑΝΝΗΣ	ΧΡΗΣΤΟΣ	1970-01-01	2010-08-25	34564381	X87658
ΠΑΠΑΜΙΧΑΗΛ	ΜΙΧΑΗΛΗΣ	ΝΙΚΟΛΑΟΣ	1980-09-08	2009-07-07	78736368	P63176
ΔΗΜΗΤΡΙΟΥ	ΕΛΕΝΗ	ΓΙΑΝΝΗΣ	1988-09-08	2011-01-01	76236261	M72827
ΑΠΟΣΤΟΛΙΔΗΣ	ΓΡΗΓΟΡΗΣ	ΑΓΓΕΛΟΣ	1981-06-05	2012-03-02	64788383	N37728
ΠΑΠΑ	ΜΑΡΙΑ	ΝΙΚΟΛΑΟΣ	1982-09-07	2013-07-06	69803333	K27272

ΤΗΛ_ΚΙΝΗΤΟ	ΤΗΛ_ΣΤΑΘΕΡΟ	ΔΙΕΥΘΥΝΣΗ	ΠΟΛΗ	ΜΙΣΘΟΣ
6980864680	23210-56836	ΒΕΝΙΖΕΛΟΥ 10	ΘΕΣΣΑΛΟΝΙΚΗ	700,0000
6978286253	23210-36284	ΡΕΔΑΙΣΤΟΥ 50	ΣΕΡΡΕΣ	900,0000
6982527227	23210-27282	Γ.ΡΑΚΙΤΝΖΗ 20	ΣΕΡΡΕΣ	800,0000
6993646646	23210-37737	ΑΪΔΙΝΙΟΥ 10	ΣΕΡΡΕΣ	400,0000
6983738374	23210-37373	Κ.ΣΤΑΜΟΥΛΗ 14	ΣΕΡΡΕΣ	400,0000

**ΣΧΟΛΙΟ!** Το πεδίο ΚΩΔΙΚΟΣ\_ΤΜΗΜΑΤΟΣ το αφήνετε NULL (κενό) γιατί όταν κάνετε τις συσχετίσεις εάν έχει πάρει τιμές χωρίς να υπάρχουν στον πίνακα ΤΜΗΜΑ θα βγάλει σφάλμα.

Οι τύποι δεδομένων **Sql Server** είναι πολλοί και θα μπορούσαν να χωριστούν και σε κατηγορίες όπως θα δείτε παρακάτω:

#### Τύποι για ημερομηνίες

Date	Datetimeoffset
Datetime2	Smalldatetime
Datetime	Time

#### Τύποι για νούμερα

Bigint	Numeric
Bit	Smallint
Decimal	Smallmoney
Int	Tinyint
Money	

#### Τύποι προσεγγιστικών τιμών

Float	Real
-------	------

#### Τύποι για χαρακτήρες

Char	Varchar
Text	

#### Τύποι για χαρακτήρες με διεθνή κωδικοποίηση

Nchar	Nvarchar
Ntext	

#### Τύποι για δυαδικές συμβολοσειρές

Binary	Varbinary
Image	

#### Άλλοι τύποι δεδομένων

Cursor	Table
Hierarchyid	Timestamp
Sql_variant	Uniqueidentifier
Xml	Spatial types

Ένας άλλος τρόπος δημιουργίας του πίνακα ΕΡΓΑΖΟΜΕΝΟΙ είναι η σχεδίαση του με **SQL**. Για την δημιουργία του πίνακα με τον εξής τρόπο η διαδικασία είναι η εξής:

➤ Δεξί κλικ στη βάση με όνομα Εργαζόμενοι-τμήματα-έργα → New Query.

```
CREATE TABLE [ΕΡΓΑΖΟΜΕΝΟΙ] (  
[ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ] [int] IDENTITY(1,1) NOT NULL,  
[ΕΠΙΘΕΤΟ] [varchar](20) NULL,  
[ΟΝΟΜΑ] [varchar](20) NULL,  
[ΟΝΟΜΑ_ΠΑΤΡΟΣ] [varchar](20) NULL,  
[ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ] [date] NULL,  
[ΗΜΕΡΟΜΗΝΙΑ_ΠΡΟΣΛΗΨΗΣ] [date] NULL,  
[ΑΦΜ] [varchar](10) NULL,  
[ΑΡΙΘΜΟΣ_ΤΑΥΤΟΤΗΤΑΣ] [varchar](10) NULL,  
[ΤΗΛ_ΚΙΝΗΤΟ] [varchar](20) NULL,  
[ΤΗΛ_ΣΤΑΘΕΡΟ] [varchar](20) NULL,  
[ΔΙΕΥΘΥΝΣΗ] [varchar](20) NULL,  
[ΠΟΛΗ] [varchar](20) NULL,  
[ΜΙΣΘΟΣ] [money] NULL,  
[ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ] [int] NULL,  
CONSTRAINT [PK_ΕΡΓΑΖΟΜΕΝΟΙ] PRIMARY KEY (ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ)
```

## Άσκηση 10 – Δημιουργία συσχετίσεων πινάκων σε SQL Server

Σκοπός αυτής της εργαστηριακής άσκησης είναι η δημιουργία των πινάκων ΤΜΗΜΑ, ΕΡΓΑΖΕΤΑΙ ΚΑΙ ΕΡΓΟ και η δημιουργία συσχετίσεων μεταξύ αυτών και του πίνακα ΕΡΓΑΖΟΜΕΝΟΙ της προηγούμενης άσκησης.

### ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΙΝΑΚΑ ΤΜΗΜΑ

Με πεδία: ΚΩΔΙΚΟΣ ΤΜΗΜΑΤΟΣ(INT), ΟΝΟΜΑ(VARCHAR(30)),

ΔΡΑΣΤΗΡΙΟΤΗΤΑ (VARCHAR(30)), ΤΗΛΕΦΩΝΟ(VARCHAR(30)).

	Column Name	Data Type	Allow Nulls
▶	ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ	int	<input type="checkbox"/>
	ΟΝΟΜΑ	varchar(30)	<input checked="" type="checkbox"/>
	ΔΡΑΣΤΗΡΙΟΤΗΤΑ	varchar(30)	<input checked="" type="checkbox"/>
	ΤΗΛΕΦΩΝΟ	varchar(30)	<input checked="" type="checkbox"/>

Εικόνα 73

### Με κώδικα SQL

```
CREATE TABLE [ΤΜΗΜΑ] (  
    [ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ] [int] IDENTITY(1,1) NOT NULL,  
    [ΟΝΟΜΑ] [varchar](30) NULL,  
    [ΔΡΑΣΤΗΡΙΟΤΗΤΑ] [varchar](30) NULL,  
    [ΤΗΛΕΦΩΝΟ] [varchar](30) NULL,  
    CONSTRAINT [PK_ΤΜΗΜΑ] PRIMARY KEY [ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ]);
```


Οι εγγραφές που θα περιέχει ο πίνακας ΤΜΗΜΑ θα είναι **(Αλλά θα εισαχθούν αφού πρώτα γίνουν οι συσχετίσεις)**:

ΟΝΟΜΑ	ΔΡΑΣΤΗΡΙΟΤΗΤΑ	ΤΗΛΕΦΩΝΟ
ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	ΑΠΟΣΦΑΛΜΑΤΩΣΗ ΚΩΔΙΚΑ	2310-68554
ΣΧΕΔΙΑΣΜΟΥ	ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ	23210-68555
ΛΟΓΙΣΤΗΡΙΟΥ	ΟΙΚΟΝΟΜΙΚΗ ΔΙΑΧΕΙΡΙΣΗ	23210-68557

Εικόνα 74

## ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΙΝΑΚΑ ΕΡΓΟ

Με πεδία: ΚΩΔΙΚΟΣ ΕΡΓΟΥ (INT), ΤΙΤΛΟΣ(VARCHAR(30)),  
ΠΕΡΙΓΡΑΦΗ(VARCHAR(50)),  
ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ(MONEY), ΗΜΕΡΟΜΗΝΙΑ\_ΕΝΑΡΞΗΣ(DATE),  
ΗΜΕΡΟΜΗΝΙΑ\_ΛΗΞΗΣ(DATE).

	Column Name	Data Type	Allow Null
	ΚΩΔΙΚΟΣ_ΕΡΓΟΥ	int	<input type="checkbox"/>
	ΤΙΤΛΟΣ	varchar(30)	<input checked="" type="checkbox"/>
	ΠΕΡΙΓΡΑΦΗ	varchar(50)	<input checked="" type="checkbox"/>
	ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ	money	<input checked="" type="checkbox"/>
	ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ	date	<input checked="" type="checkbox"/>
	ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ	date	<input checked="" type="checkbox"/>

## Με κώδικα SQL

```
CREATE TABLE [ΕΡΓΟ] (  
    [ΚΩΔΙΚΟΣ_ΕΡΓΟΥ] [int] IDENTITY(1,1) NOT NULL,  
    [ΤΙΤΛΟΣ] [varchar](30) NULL,  
    [ΠΕΡΙΓΡΑΦΗ] [varchar](50) NULL,  
    [ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ] [money] NULL,  
    [ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ] [date] NULL,  
    [ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ] [date] NULL,  
    CONSTRAINT [PK_ΕΡΓΟ] PRIMARY [ΚΩΔΙΚΟΣ_ΕΡΓΟΥ]);
```

Οι εγγραφές που θα περιέχει ο πίνακας ΕΡΓΟ θα είναι οι παρακάτω (**Αλλά θα εισαχθούν αφού πρώτα γίνουν οι συσχετίσεις**):

ΤΙΤΛΟΣ	ΠΕΡΙΓΡΑΦΗ	ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ	ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ
ΑΠΟΣΦΑΛΜΑΤΩΣΗ	ΒΡΙΣΚΕΙ ΤΑ ΣΦΑΛΜΑΤΑ ΣΤΗΝ ΒΔ ΚΑΙ ΣΤΟΝ ΚΩΔΙΚΑ	5000,0000	2014-05-15	2014-05-25
ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ	ΔΙΑΓΡΑΜΜΑΤΑ ΓΙΑ ΤΙΣ ΑΛΛΑΓΕΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ	3000,0000	2014-01-01	2014-01-30
ΕΝΗΜΕΡΩΣΗ ΠΑΚΕΤΩΝ ΛΟΓΙΣΜΙΚΟΥ	ΑΝΑΒΑΘΜΙΣΗ ΤΩΝ ΠΑΚΕΤΩΝ ΓΙΑ ΤΗΝ ΤΕΛΕΥΤΑΙΑ ΕΚΔΟΣΗ	10000,0000	2014-11-10	2014-11-15

Εικόνα 75

## ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΙΝΑΚΑ ΕΡΓΑΖΕΤΑΙ

Με πεδία: ΚΩΔΙΚΟΣ ΕΡΓΟΥ (INT), ΚΩΔΙΚΟΣ ΕΡΓΑΖΟΜΕΝΟΥ (INT),  
ΗΜΕΡΟΜΗΝΙΑ\_ΑΝΑΘΕΣΗΣ (DATE).

	Column Name	Data Type	Allow Nulls
▶	ΚΩΔΙΚΟΣ_ΕΡΓΟΥ	int	<input type="checkbox"/>
▶	ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕ...	int	<input type="checkbox"/>
	ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΘ...	date	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

### Με κώδικα SQL

```
CREATE TABLE [ΕΡΓΑΖΕΤΑΙ] (
    [ΚΩΔΙΚΟΣ_ΕΡΓΟΥ] [int] IDENTITY(1,1) NOT NULL,
    [ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ] [int] NOT NULL,
    [ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΘΕΣΗΣ] [date] NULL,
    CONSTRAINT [PK_ΕΡΓΑΖΕΤΑΙ] PRIMARY KEY [ΚΩΔΙΚΟΣ_ΕΡΓΟΥ];
```

Οι εγγραφές που θα περιέχει ο πίνακας ΕΡΓΑΖΕΤΑΙ θα είναι οι παρακάτω **(Αλλά θα εισαχθούν αφού πρώτα γίνουν οι συσχετίσεις)**:

ΚΩΔΙΚΟΣ_ΕΡΓΟΥ	ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ	ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΘΕΣΗΣ
1	2	2014-05-15
2	3	2014-01-01
3	5	2014-11-10

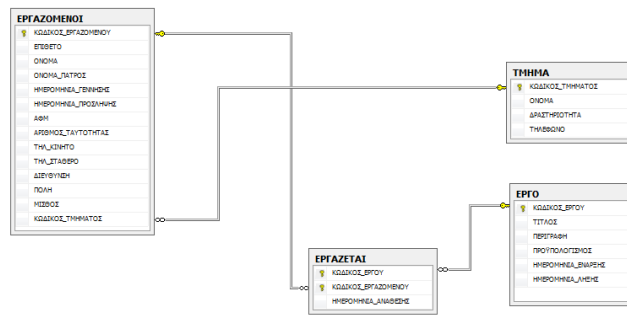
### ΔΗΜΙΟΥΡΓΙΑ ΣΥΣΧΕΤΙΣΕΩΝ ΜΕΤΑΞΥ ΤΩΝ ΠΙΝΑΚΩΝ

Για την δημιουργία συσχετίσεων μεταξύ των πινάκων υπάρχουν δύο τρόποι:

#### 1ος Τρόπος με Database Diagrams

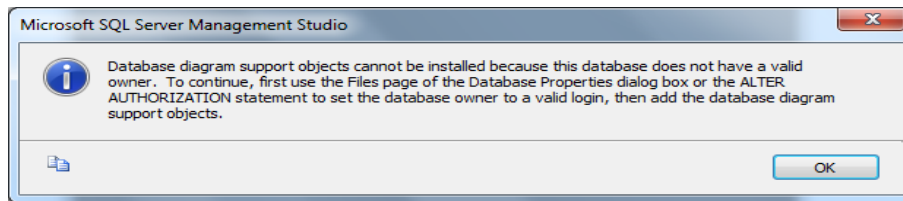
- Αφού κλείσετε όλους τους πίνακες πηγαίνετε στο Object Explorer → Databases → Ανοίγετε την δενδρική δομή της βάσης που επιθυμείτε (Εργαζόμενοι-τμήματα-έργα) → Δεξί κλικ Database Diagram → New Database Diagram και ανοίγει ένα παράθυρο με τους πίνακες που θέλετε να κάνετε τις συσχετίσεις. Επιλέγεται τους πίνακες που επιθυμείτε και ADD. Αν δεν φαίνονται όλοι οι πίνακες αλλάζετε το zoom σε μικρότερη τιμή. Στη συνέχεια σέρνετε το ποντίκι από το πεδίο που επιθυμείτε μέχρι το πεδίο που θα ενώσετε και εκεί το αφήνετε ώστε να γίνει η συσχέτιση και να ενωθούν τα πεδία. Πατήστε OK στα επόμενα δύο παράθυρα ώστε να ολοκληρωθεί η συσχέτιση. Το αποτέλεσμα που θα δείτε είναι η Εικόνα 72





Εικόνα 76

**ΣΧΟΛΙΟ!** Σε περίπτωση που ακολουθήσετε την παραπάνω διαδικασία και σας βγάλει το ακόλουθο μήνυμα το διορθώνετε πηγαίνοντας στη βάση σας στο **Object Explorer**→**Δεξί κλικ**→**Properties**→**Ανοίγει ένα νέο παράθυρο από το οποίο επιλέγετε File στο Select Pages και αλλάζετε τον Owner σε** [NT AUTHORITY\SYSTEM] .



## 2ος Τρόπος Με κώδικα SQL

Επιλέγετε από την κορδέλα και εισάγετε τον παρακάτω κώδικα:

### **ΕΡΓΑΖΟΜΕΝΟΙ-ΤΜΗΜΑ**

```
ALTER TABLE ΕΡΓΑΖΟΜΕΝΟΙ ADD CONSTRAINT
```

```
FK_KT FOREIGN KEY ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ REFERENCES ΤΜΗΜΑ  
(ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ);
```

### **ΕΡΓΑΖΕΤΑΙ-ΕΡΓΑΖΟΜΕΝΟΙ**

```
ALTER TABLE ΕΡΓΑΖΕΤΑΙ ADD CONSTRAINT
```

```
FK_KE FOREIGN KEY ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ REFERENCES ΕΡΓΑΖΟΜΕΝΟΙ  
(ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ);
```

### **ΕΡΓΑΖΕΤΑΙ-ΕΡΓΟ**

```
ALTER TABLE ΕΡΓΑΖΕΤΑΙ ADD CONSTRAINT
```



```
FK KEP FOREIGN KEY ΚΩΔΙΚΟΣ_ΕΡΓΟΥ REFERENCES ΕΡΓΟ (ΚΩΔΙΚΟΣ_ΕΡΓΟΥ);
```

## Άσκηση 11 - INSERT INTO, UPDATE, DELETE σε SQL Server

Σκοπός αυτής της εργαστηριακής άσκησης είναι η εξοικείωση σας σε έναν διαφορετικό τρόπο εισαγωγής εγγραφών, διαγραφής αυτών καθώς και τροποποίηση τους. Σε προηγούμενες ασκήσεις μάθατε την εισαγωγή εγγραφών πηγαίνοντας στον Object Explorer → Δεξί κλικ στον πίνακα → Edit top 200 rows. Σε αυτήν την εργαστηριακή άσκηση οι εγγραφές και όλες οι τροποποιήσεις τους θα γίνονται μέσω των Queries.

Έστω ο πίνακας ΤΜΗΜΑ ο οποίος περιέχει τις παρακάτω εγγραφές :

ΟΝΟΜΑ	ΔΡΑΣΤΗΡΙΟΤΗΤΑ	ΤΗΛΕΦΩΝΟ
ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	ΑΠΟΣΦΑΛΜΑΤΩΣΗ ΚΩΔΙΚΑ	2310-68554
ΣΧΕΔΙΑΣΜΟΥ	ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ	23210-68555
ΛΟΓΙΣΤΗΡΙΟΥ	ΟΙΚΟΝΟΜΙΚΗ ΔΙΑΧΕΙΡΙΣΗ	23210-68557

Για να τις εισάγετε πηγαίνετε επάνω αριστερά στην κορδέλα και  New Query . Εκεί εισάγετε τις εγγραφές ως εξής:

```
INSERT ΤΜΗΜΑ
(ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ, ΟΝΟΜΑ, ΔΡΑΣΤΗΡΙΟΤΗΤΑ, ΤΗΛΕΦΩΝΟ)
VALUES (1, 'ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ', 'ΑΠΟΣΦΑΛΜΑΤΩΣΗΣ ΚΩΔΙΚΑ', '2310-
68554');
```

```
INSERT ΤΜΗΜΑ
(ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ, ΟΝΟΜΑ, ΔΡΑΣΤΗΡΙΟΤΗΤΑ, ΤΗΛΕΦΩΝΟ)
VALUES (2, 'ΣΧΕΔΙΑΣΜΟΥ', 'ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ', '23210-68555');
```

```
INSERT ΤΜΗΜΑ
(ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ, ΟΝΟΜΑ, ΔΡΑΣΤΗΡΙΟΤΗΤΑ, ΤΗΛΕΦΩΝΟ)
VALUES (3, 'ΛΟΓΙΣΤΗΡΙΟΥ', 'ΟΙΚΟΝΟΜΙΚΗ ΔΙΑΧΕΙΡΙΣΗ', '23210-68557');
```

Ομοίως και για τους άλλους τρεις πίνακες παρακάτω ακολουθούμε ενδεικτικά παραδείγματα:

Για τον πίνακα **ΕΡΓΑΖΟΜΕΝΟΙ**:

```
INSERT ΕΡΓΑΖΟΜΕΝΟΙ
(ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΟΝΟΜΑ_ΠΑΤΡΟΣ,
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ, ΗΜΕΡΟΜΗΝΙΑ_ΠΡΟΣΛΗΨΗΣ, ΑΦΜ, ΑΡΙΘΜΟΣ_ΤΑΥΤΟΤΗΤΑΣ,
ΤΗΛ_ΚΙΝΗΤΟ, ΤΗΛ_ΣΤΑΘΕΡΟ, ΔΙΕΥΘΥΝΣΗ, ΠΟΛΗ, ΜΙΣΘΟΣ, ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ)
VALUES (1, 'ΠΑΠΑΔΟΠΟΥΛΟΣ', 'ΓΙΑΝΝΗΣ', 'ΧΡΗΣΤΟΣ', '01/01/1970,
25/08/2010, '34564381', 'X87658', '6980864680', '2310-56836',
'BENIZELOU 10', 'ΘΕΣΣΑΛΟΝΙΚΗ', 700, 2);
```

Για τον πίνακα **ΕΡΓΑΖΕΤΑΙ**:

```
INSERT ΕΡΓΑΖΕΤΑΙ
(ΚΩΔΙΚΟΣ_ΕΡΓΟΥ, ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ, ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΘΕΣΗΣ) VALUES (1,
2, 15/5/2014);
```

Για τον πίνακα **ΕΡΓΟ**:

```
INSERT ΕΡΓΟ
(ΚΩΔΙΚΟΣ_ΕΡΓΟΥ, ΤΙΤΛΟΣ, ΠΕΡΙΓΡΑΦΗ, ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ,
ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ, ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ) VALUES (1, 'ΑΠΟΣΦΑΛΜΑΤΩΣΗ',
'ΒΡΙΣΚΕΙ ΤΑ ΣΦΑΛΜΑΤΑ ΣΤΗ ΒΔ ΚΑΙ ΣΤΟΝ ΚΩΔΙΚΑ', 15/5/2014, 25/5/2014);
```

Μόλις ολοκληρώσετε τις εγγραφές εκτελέστε το Query  και αποθηκεύστε.

Εκτός από την εισαγωγή εγγραφών μπορεί να γίνει και διαγραφή και τροποποίηση εγγραφών ως εξής:

### Παράδειγμα 1

Αλλάξτε το όνομα του εργαζομένου που το επίθετο του είναι ΠΑΠΑΔΟΠΟΥΛΟΣ από ΓΙΑΝΝΗΣ σε ΚΩΣΤΑΣ.

### Απάντηση 1

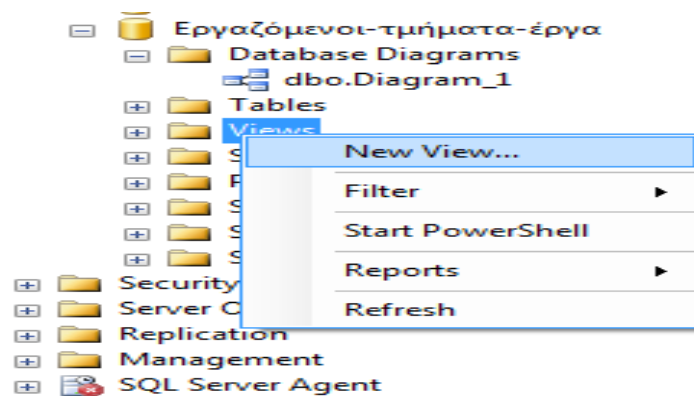
```
UPDATE ΕΡΓΑΖΟΜΕΝΟΙ SET ΟΝΟΜΑ='ΚΩΣΤΑΣ' WHERE
ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ=2;
```

## Άσκηση 12 – Δημιουργία ερωτημάτων QBE στον SQL SERVER

Σκοπός της τρέχουσας εργαστηριακής άσκησης είναι η δημιουργία ερωτημάτων με Query by Example στον SQL Server.

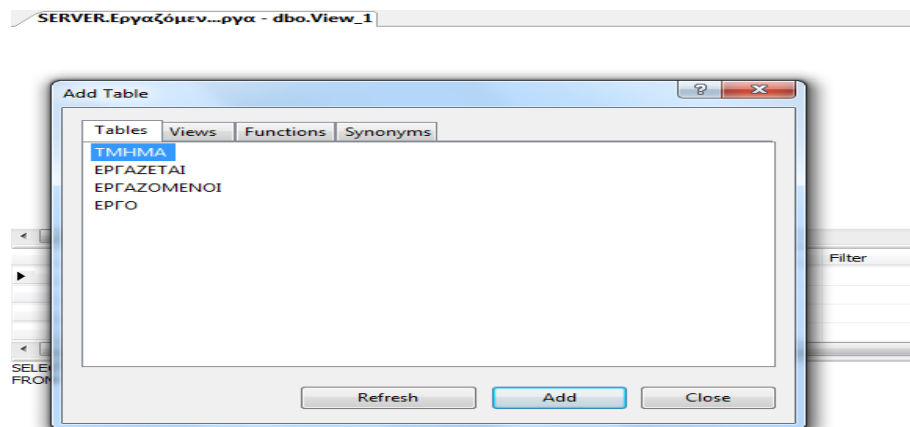
Για να δημιουργήσετε ένα ερώτημα με Query by Example η διαδικασία έχει ως εξής:

- Πηγαίνετε στον Object Explorer→Ανοίγετε την δενδρική δομή του Databases και επιλέγετε την βάση που επιθυμείτε. Ανοίγετε επίσης την δενδρική δομή της βάσης→ Views → Δεξί κλικ → New View.



Εικόνα 77

- Στη συνέχεια ανοίγει το παράθυρο της Εικόνας 76 από το οποίο επιλέγεται τους πίνακες που συσχετίζονται με το ερώτημα σας.



Εικόνα 78

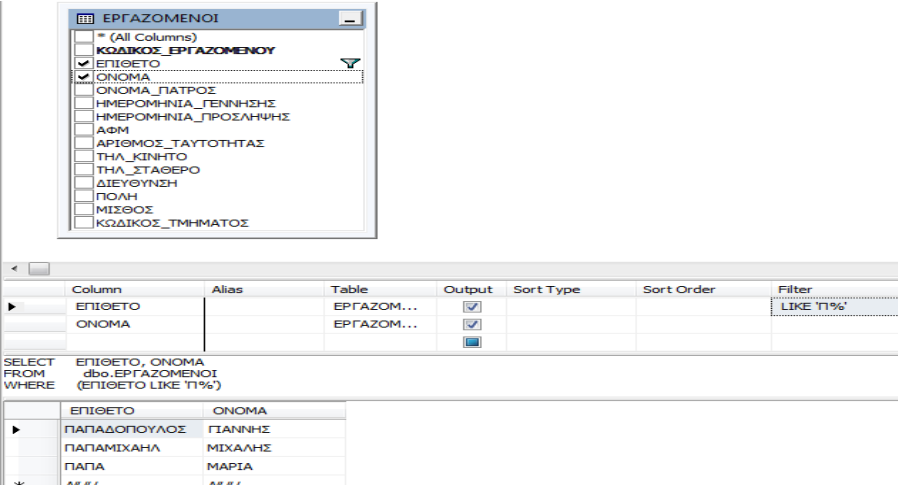
## Παρακάτω ακολουθούν παραδείγματα ερωτημάτων με QBE:

Για την δημιουργία ερωτήματος με QBE μόλις ανοίξει η όψη (view) εισάγετε τους πίνακες που θα χρειαστούν και τσεκάρτε τα πεδία που θα εμφανίζονται και θα συμμετέχουν στο ερώτημα. Μόλις τα τσεκάρτε αυτόματα τοποθετούνται στην στήλη column. Στην συνέχεια στο Filter εισάγετε την συνθήκη όμως στην γραμμή του πεδίου που θα συμμετέχει στο ερώτημα. Τέλος σε περίπτωση που θέλετε να εμφανιστεί κάποιο πεδίο σε αύξουσα σειρά, στο Sort Type του πεδίου αυτού κάνετε την τιμή 1. Παρακάτω ακολουθούν αναλυτικά παραδείγματα.

### Παράδειγμα 1

Εμφανίστε το ΟΝΟΜΑ και το ΕΠΙΘΕΤΟ από τους εργαζομένους που το επίθετο τους ξεκινάει από 'Π'.

### Απάντηση 1



The screenshot shows a QBE query editor for a table named 'ΕΡΓΑΖΟΜΕΝΟΙ'. The 'Column' section has 'ΕΠΙΘΕΤΟ' and 'ΟΝΟΜΑ' selected. The 'Filter' section contains the condition 'LIKE Π%'. The SQL statement below the editor is:

```
SELECT ΕΠΙΘΕΤΟ, ΟΝΟΜΑ
FROM dbo.ΕΡΓΑΖΟΜΕΝΟΙ
WHERE (ΕΠΙΘΕΤΟ LIKE Π%)
```

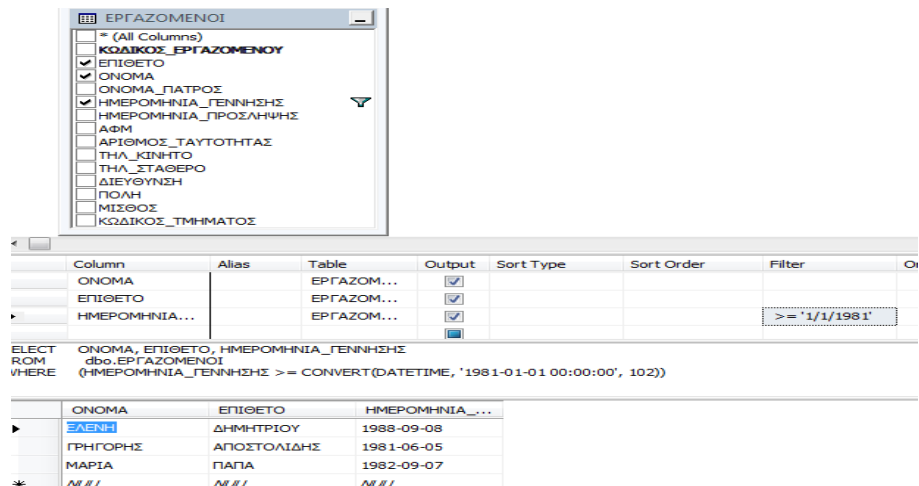
The resulting query results are shown in a table below:

ΕΠΙΘΕΤΟ	ΟΝΟΜΑ
ΠΑΠΑΔΟΠΟΥΛΟΣ	ΓΙΑΝΝΗΣ
ΠΑΠΑΜΙΧΑΗΛ	ΜΙΧΑΗΛΗΣ
ΠΑΠΑ	ΜΑΡΙΑ
NULL	NULL

## Παράδειγμα 2

Εμφανίστε το ΟΝΟΜΑ, ΕΠΙΘΕΤΟ και την ΗΜΕΡΟΜΗΝΙΑ\_ΓΕΝΝΗΣΗΣ των εργαζομένων που γεννήθηκαν από το 1981 και μετά.

## Απάντηση 2



The screenshot shows a query window titled 'ΕΡΓΑΖΟΜΕΝΟΙ'. The query is as follows:

```
SELECT ONΟΜΑ, ΕΠΙΘΕΤΟ, ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ
FROM dbo.ΕΡΓΑΖΟΜΕΝΟΙ
WHERE (ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ >= CONVERT(DATETIME, '1981-01-01 00:00:00', 102))
```

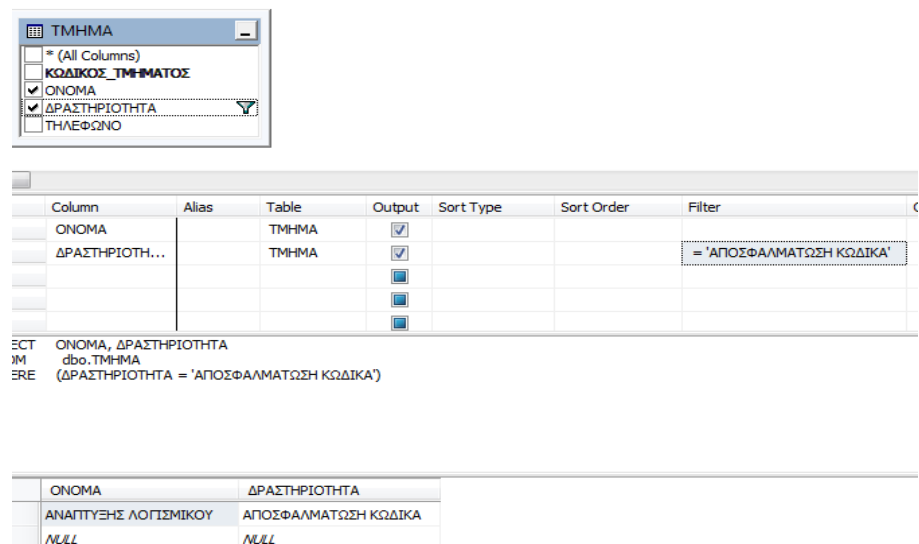
The results table is as follows:

ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΗΜΕΡΟΜΗΝΙΑ_...
ΞΑΝΘΗ	ΔΗΜΗΤΡΙΟΥ	1988-09-08
ΓΡΗΓΟΡΗΣ	ΑΠΟΣΤΟΛΙΔΗΣ	1981-06-05
ΜΑΡΙΑ	ΠΑΠΑ	1982-09-07
NULL	NULL	NULL

## Παράδειγμα 3

Εμφανίστε το όνομα του τμήματος που η δραστηριότητα του είναι ΑΠΟΣΦΑΛΜΑΤΩΣΗ ΚΩΔΙΚΑ.

## Απάντηση 3



The screenshot shows a query window titled 'ΤΜΗΜΑ'. The query is as follows:

```
SELECT ΟΝΟΜΑ, ΔΡΑΣΤΗΡΙΟΤΗΤΑ
FROM dbo.ΤΜΗΜΑ
WHERE (ΔΡΑΣΤΗΡΙΟΤΗΤΑ = 'ΑΠΟΣΦΑΛΜΑΤΩΣΗ ΚΩΔΙΚΑ')
```

The results table is as follows:

ΟΝΟΜΑ	ΔΡΑΣΤΗΡΙΟΤΗΤΑ
ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ	ΑΠΟΣΦΑΛΜΑΤΩΣΗ ΚΩΔΙΚΑ
NULL	NULL

#### Παράδειγμα 4

Εμφανίστε όλα τα έργα που ξεκίνησαν μετά από τη 1/1/2014.

#### Απάντηση 4

The screenshot shows a query window titled 'ΕΡΓΟ' with the following columns selected: ΚΩΔΙΚΟΣ\_ΕΡΓΟΥ, ΤΙΤΛΟΣ, ΠΕΡΙΓΡΑΦΗ, ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ, ΗΜΕΡΟΜΗΝΙΑ\_ΕΝΑΡΞΗΣ, and ΗΜΕΡΟΜΗΝΙΑ\_ΛΗΞΗΣ. The query is:

```
SELECT ΤΙΤΛΟΣ, ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ
FROM dbo.ΕΡΓΟ
WHERE (ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ > CONVERT(DATETIME, '2014-01-01 00:00:00', 102))
```

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
ΤΙΤΛΟΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>			
ΗΜΕΡΟΜΗΝΙΑ...		ΕΡΓΟ	<input checked="" type="checkbox"/>			> '1/1/2014'

ΤΙΤΛΟΣ	ΗΜΕΡΟΜΗΝΙΑ_...
ΑΠΟΣΦΑΛΜΑΤΩΣ...	2014-05-15
ΕΝΗΜΕΡΩΣΗ ΠΑ...	2014-11-10
* NULL	NULL

#### Παράδειγμα 5

Εμφανίστε όλα τα έργα που ο προϋπολογισμός τους είναι μεγαλύτερος των 3000 και μικρότερος από 5000.

#### Απάντηση 5

The screenshot shows a query window titled 'ΕΡΓΟ' with the following columns selected: ΚΩΔΙΚΟΣ\_ΕΡΓΟΥ, ΤΙΤΛΟΣ, ΠΕΡΙΓΡΑΦΗ, ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ, ΗΜΕΡΟΜΗΝΙΑ\_ΕΝΑΡΞΗΣ, and ΗΜΕΡΟΜΗΝΙΑ\_ΛΗΞΗΣ. The query is:

```
SELECT ΤΙΤΛΟΣ, ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ
FROM dbo.ΕΡΓΟ
WHERE (ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ >= 3000 AND ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ <= 5000)
```

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
ΤΙΤΛΟΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>			
ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>			>= 3000 AND <= 5000

ΤΙΤΛΟΣ	ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ
ΑΠΟΣΦΑΛΜΑΤΩΣΗ	5000,0000
ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ	3000,0000

## Παράδειγμα 6

Εμφανίστε το έργο που η ημερομηνία έναρξης ήταν 15/5/2014 και η ημερομηνία λήξης ήταν 25/5/2014.

## Απάντηση 6

The screenshot shows the SQL Server Enterprise Manager interface. A query window is open, displaying a query for the 'ΕΡΓΟ' table. The query filters for records where the start date is '15/5/2014' and the end date is '25/5/2014'. The results grid shows one record with the title 'ΠΙΣΩΔΑΜΑΤΟΣΗ'.

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...
ΤΙΤΛΟΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>			= '15/5/2014'		
ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>			= '25/5/2014'		
ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>					

```
SELECT ΤΙΤΛΟΣ, ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ, ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ
FROM   dbo.ΕΡΓΟ
WHERE  (ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ = CONVERT(DATETIME, '2014-05-15 00:00:00', 102)) AND (ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ = CONVERT(DATETIME, '2014-05-25 00:00:00', 102))
```

ΤΙΤΛΟΣ	ΗΜΕΡΟΜΗΝΙΑ_ΕΝΑΡΞΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ
ΠΙΣΩΔΑΜΑΤΟΣΗ	2014-05-15	2014-05-25
*	NULL	NULL

## Παράδειγμα 7

Εμφανίστε το ΟΝΟΜΑ, το ΕΠΙΘΕΤΟ και το ΚΙΝΗΤΟ των εργαζομένων που το κινητό τους ξεκινάει από 698.....

## Απάντηση 7

The screenshot shows the SQL Server Enterprise Manager interface. A query window is open, displaying a query for the 'ΕΡΓΑΖΟΜΕΝΟΙ' table. The query filters for records where the mobile phone number starts with '698'. The results grid shows three records with names ΠΑΝΝΗΣ, ΕΛΕΝΗ, and ΜΑΡΙΑ.

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...
ΟΝΟΜΑ		ΕΡΓΑΖΟΜ...	<input checked="" type="checkbox"/>					
ΕΠΙΘΕΤΟ		ΕΡΓΑΖΟΜ...	<input checked="" type="checkbox"/>					
ΤΗΛ_ΚΙΝΗΤΟ		ΕΡΓΑΖΟΜ...	<input checked="" type="checkbox"/>			LIKE '698%'		

```
SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΤΗΛ_ΚΙΝΗΤΟ
FROM   dbo.ΕΡΓΑΖΟΜΕΝΟΙ
WHERE  (ΤΗΛ_ΚΙΝΗΤΟ LIKE '698%')
```

ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΤΗΛ_ΚΙΝΗΤΟ
ΠΑΝΝΗΣ	ΠΑΠΑΔΟΠΟΥΛΟΣ	6980864680
ΕΛΕΝΗ	ΔΗΜΗΤΡΙΟΥ	6982527227
ΜΑΡΙΑ	ΠΑΠΑ	6983738374
*	NULL	NULL



## Παράδειγμα 8

Εμφανίστε αναλυτικά τους εργαζομένους που είναι από Σέρρες και η διεύθυνσή τους είναι η Βενιζέλου.

### Απάντηση 8

The screenshot shows a SQL query window with the following SQL code:

```
SELECT *
FROM dbo.ΕΡΓΑΖΟΜΕΝΟΙ
WHERE (ΔΙΕΥΘΥΝΣΗ = 'ΒΕΝΙΖΕΛΟΥ 10') AND (ΠΟΛΗ = 'ΘΕΣΣΑΛΟΝΙΚΗ')
```

The result set contains one row:

ΚΩΔΙΚΟΣ_ΕΡΓ...	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ_...	ΗΜΕΡΟΜΗΝΙΑ_...	ΑΦΜ	ΑΡΙΘΜΟΣ_ΤΑΥ...	ΤΗΛ_ΚΙΝΗΤΟ
2	ΠΑΠΑΔΟΠΟΥΛΟΣ	ΓΙΑΝΝΗΣ	ΧΡΗΣΤΟΣ	1970-01-01	2010-08-25	34564381	X87658	6980864680

## Παράδειγμα 9

Εμφανίστε το ΟΝΟΜΑ, το ΕΠΙΘΕΤΟ και το ΟΝΟΜΑ\_ΠΑΤΡΟΣ των εργαζομένων που το όνομα πατρός τους είναι Νικόλαος.

### Απάντηση 9

The screenshot shows a SQL query window with the following SQL code:

```
SELECT ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ_ΠΑΤΡΟΣ
FROM dbo.ΕΡΓΑΖΟΜΕΝΟΙ
WHERE (ΟΝΟΜΑ_ΠΑΤΡΟΣ = 'ΝΙΚΟΛΑΟΣ')
```

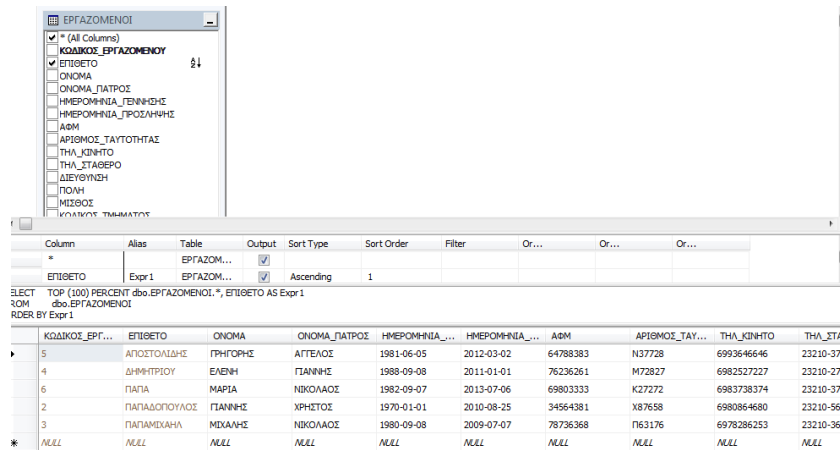
The result set contains two rows:

ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ_ΠΑΤΡΟΣ
ΜΙΧΑΗΛΣ	ΠΑΠΑΜΕΧΑΗΛ	ΝΙΚΟΛΑΟΣ
ΜΑΡΙΑ	ΠΑΠΑ	ΝΙΚΟΛΑΟΣ

## Παράδειγμα 10

Εμφανίστε τους εργαζομένους αναλυτικά και ταξινομήστε τους βάση ΕΠΙΘΕΤΟΥ κατά αύξουσα σειρά.

## Απάντηση 10



The screenshot shows a SQL query editor window titled "ΕΡΓΑΖΟΜΕΝΟΙ". The query is: `SELECT TOP (100) PERCENT dbo.ΕΡΓΑΖΟΜΕΝΟΙ.*, ΕΠΙΘΕΤΟ AS Expr1 ORDER BY Expr1`. The results table displays columns: ΚΩΔΙΚΟΣ\_ΕΡΓ... (ID), ΕΠΙΘΕΤΟ (Last Name), ΟΝΟΜΑ (Name), ΟΝΟΜΑ\_ΠΑΤΡΟΣ (Paternal Name), ΗΜΕΡΟΜΗΝΙΑ\_... (Date of Birth), ΗΜΕΡΟΜΗΝΙΑ\_... (Date of Hire), ΑΦΜ (Tax ID), ΑΡΙΘΜΟΣ\_ΤΑΥ... (ID Number), ΤΗΛ\_ΚΙΝΗΤΟ (Mobile Phone), and ΤΗΛ\_ΣΤΑ... (Fixed Phone). The results are sorted by last name in ascending order.

ΚΩΔΙΚΟΣ_ΕΡΓ...	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ_...	ΗΜΕΡΟΜΗΝΙΑ_...	ΑΦΜ	ΑΡΙΘΜΟΣ_ΤΑΥ...	ΤΗΛ_ΚΙΝΗΤΟ	ΤΗΛ_ΣΤΑ...
5	ΑΠΟΣΤΟΛΙΔΗΣ	ΓΡΗΓΟΡΗΣ	ΑΓΓΕΛΟΣ	1981-06-05	2012-03-02	64788383	N37728	699364666	23210-377
4	ΔΗΜΗΤΡΙΟΥ	ΕΛΕΝΗ	ΠΑΝΗΛΗΣ	1988-09-08	2011-01-01	76236261	M72827	698252722	23210-272
6	ΠΑΠΑ	ΜΑΡΙΑ	ΝΙΚΟΛΑΟΣ	1982-09-07	2013-07-06	69803333	K27272	6983738374	23210-373
2	ΠΑΠΑΔΟΠΟΥΛΟΣ	ΠΑΝΗΛΗΣ	ΧΡΗΣΤΟΣ	1970-01-01	2010-08-25	34564381	X87658	6980864680	23210-568
3	ΠΑΠΑΜΙΧΑΗΛ	ΜΙΧΑΗΛΗΣ	ΝΙΚΟΛΑΟΣ	1980-09-08	2009-07-07	78736368	P63176	6978286253	23210-362
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

## Άσκηση 13 – Σύνθετα ερωτήματα και συναθροιστικές συναρτήσεις σε SQL Server

Σκοπός αυτής της εργαστηριακής άσκησης είναι η εξοικείωση σας με τα σύνθετα ερωτήματα και συναθροιστικές συναρτήσεις στο περιβάλλον της Microsoft SQL Server.

### Παραδείγματα σύνθετων ερωτημάτων

#### Ερώτηση 1

Εμφανίστε το ΟΝΟΜΑ, το ΕΠΙΘΕΤΟ, την ΗΜΕΡΟΜΗΝΙΑ ΑΝΑΘΕΣΗΣ και το ΕΡΓΟ των εργαζομένων.

#### Απάντηση 1

Με QBE:

The screenshot shows the Query Designer interface with three tables: ΕΡΓΑΖΟΜΕΝΟΙ, ΕΡΓΑΖΕΤΑΙ, and ΕΡΓΟ. The columns selected are ΟΝΟΜΑ, ΕΠΙΘΕΤΟ, ΗΜΕΡΟΜΗΝΙΑ\_ΑΝΑΘΕΣΗΣ, and ΤΙΤΛΟΣ. The SQL query is as follows:

```
SELECT
FROM
  dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ,
  dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ,
  dbo.ΕΡΓΑΖΕΤΑΙ.ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΘΕΣΗΣ,
  dbo.ΕΡΓΟ.ΤΙΤΛΟΣ
FROM
  dbo.ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN
  dbo.ΕΡΓΑΖΕΤΑΙ ON
  (dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ =
  dbo.ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ)
  INNER JOIN
  dbo.ΕΡΓΟ ON
  (dbo.ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ =
  dbo.ΕΡΓΟ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ)
```

ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΗΜΕΡΟΜΗΝΙΑ_...	ΤΙΤΛΟΣ
ΓΙΑΝΝΗΣ	ΠΑΠΑΔΟΠΟΥΛΟΣ	2014-05-15	ΑΠΟΣΦΑΛΜΑΤΩ...
ΜΙΧΑΛΗΣ	ΠΑΠΑΜΙΧΑΗΛ	2014-01-01	ΔΗΜΙΟΥΡΓΙΑ ΔΙ...
ΓΡΗΓΟΡΗΣ	ΑΠΟΣΤΟΛΙΔΗΣ	2014-11-10	ΕΝΗΜΕΡΩΣΗ ΠΑ...

Εικόνα 79

Με κώδικα SQL:

```
SELECT ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ, ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ,
ΕΡΓΑΖΕΤΑΙ.ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΘΕΣΗΣ, ΕΡΓΟ.ΤΙΤΛΟΣ

FROM ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN

ΕΡΓΑΖΕΤΑΙ ON ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ =
ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ INNER JOIN

ΕΡΓΟ ON ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ = ΕΡΓΟ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ
```

## Ερώτηση 2

Ποιό είναι το έργο του εργαζομένου με επίθετο ΠΑΠΑΜΙΧΑΗΛ.

## Απάντηση 2

Με QBE:

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...
ΤΙΤΛΟΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>					
ΕΠΙΘΕΤΟ		ΕΡΓΑΖΟΜ...	<input checked="" type="checkbox"/>			= 'ΠΑΠΑΜΙΧΑΗΛ'		

```
SELECT
FROM
WHERE
dbo.ΕΡΓΟ.ΤΙΤΛΟΣ,
dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ
dbo.ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN
dbo.ΕΡΓΑΖΕΤΑΙ ON
dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ =
dbo.ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ INNER JOIN
dbo.ΕΡΓΟ ON
dbo.ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ =
dbo.ΕΡΓΟ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ
(dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ = 'ΠΑΠΑΜΙΧΑΗΛ')
```

ΤΙΤΛΟΣ	ΕΠΙΘΕΤΟ
ΔΗΜΙΟΥΡΓΙΑ ΔΙΑΓΡΑΜΜΑΤΩΝ ΡΟΗΣ	ΠΑΠΑΜΙΧΑΗΛ

Εικόνα 80

Με κώδικα SQL:

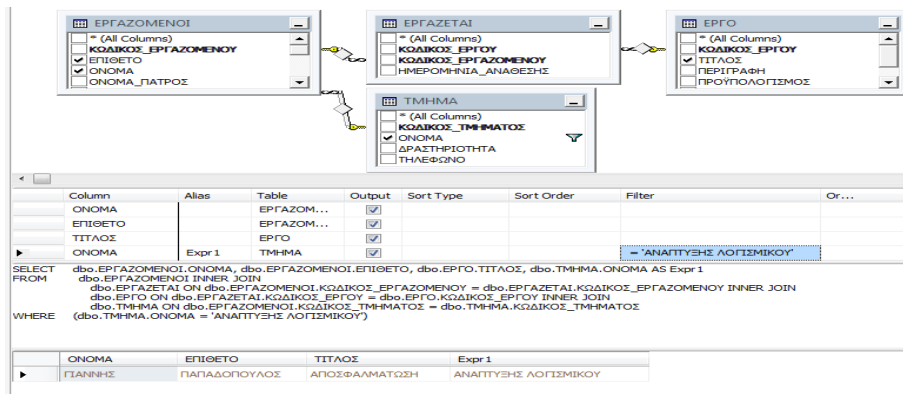
```
SELECT ΕΡΓΟ.ΤΙΤΛΟΣ, ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ
FROM ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN
ΕΡΓΑΖΕΤΑΙ ON ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ =
ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ INNER JOIN
ΕΡΓΟ ON ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ = ΕΡΓΟ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ
WHERE (ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ = 'ΠΑΠΑΜΙΧΑΗΛ')
```

## Ερώτηση 3

Εμφανίστε το ΟΝΟΜΑ, το ΕΠΙΘΕΤΟ και το ΕΡΓΟ των εργαζομένων που το τμήμα τους είναι ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ.

## Απάντηση 3

Με QBE:



Εικόνα 81

Με κώδικα SQL:

```
SELECT ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ, ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ, ΕΡΓΟ.ΤΙΤΛΟΣ,
ΤΜΗΜΑ.ΟΝΟΜΑ AS Expr1
```

```
FROM ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN
```

```
ΕΡΓΑΖΕΤΑΙ ON ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ =
ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ INNER JOIN
```

```
ΕΡΓΟ ON ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ = ΕΡΓΟ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ INNER
JOIN
```

```
ΤΜΗΜΑ ON ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ = ΤΜΗΜΑ.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ
```

```
WHERE (ΤΜΗΜΑ.ΟΝΟΜΑ = 'ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ')
```

#### Ερώτηση 4

Εμφανίστε αναλυτικά τους εργαζομένους δηλαδή τον ΚΩΔΙΚΟ\_ΕΡΓΑΖΟΜΕΝΟΥ, το ΕΠΙΘΕΤΟ, το ΟΝΟΜΑ, το ΟΝΟΜΑ\_ΠΑΤΡΟΣ, την ΗΜΕΡΟΜΗΝΙΑ\_ΓΕΝΝΗΣΗΣ, την ΗΜΕΡΟΜΗΝΙΑ\_ΠΡΟΣΛΗΨΗΣ, το ΑΦΜ, τον ΑΡΙΘΜΟΣ\_ΤΑΥΤΟΤΗΤΑΣ, το ΤΗΛ\_ΚΙΝΗΤΟ, το ΤΗΛ\_ΣΤΑΘΕΡΟ, την ΔΙΕΥΘΥΝΣΗ, την ΠΟΛΗ, τον ΜΙΣΘΟ τον ΚΩΔΙΚΟ\_ΤΜΗΜΑΤΟΣ και το ΕΡΓΟ ταξινομημένα κατά αύξουσα σειρά βάση του ΕΠΙΘΕΤΟΥ.

## Απάντηση 4

Με QBE:

Column Alias Table Output Sort Type Sort Order Filter Or... Or... Or...

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
*		ΕΡΓΑΖΟΜΕΝΟΙ	<input checked="" type="checkbox"/>						
ONOMA	Expr 1	ΕΡΓΑΖΟΜΕΝΟΙ	<input checked="" type="checkbox"/>						
ΕΠΙΘΕΤΟ	Expr 2	ΕΡΓΑΖΟΜΕΝΟΙ	<input checked="" type="checkbox"/>	Ascending	1				
ΤΙΤΛΟΣ		ΕΡΓΟ	<input checked="" type="checkbox"/>						

```
SELECT TOP (100) PERCENT dbo.ΕΡΓΑΖΟΜΕΝΟΙ.*, dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ AS Expr1, dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ AS Expr2, dbo.ΕΡΓΟ.ΤΙΤΛΟΣ
FROM
  dbo.ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN
  dbo.ΕΡΓΑΖΕΤΑΙ ON dbo.ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ = dbo.ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ INNER JOIN
  dbo.ΕΡΓΟ ON dbo.ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ = dbo.ΕΡΓΟ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ
ORDER BY Expr2
```

ΚΩΔΙΚΟΣ_ΕΡΓ...	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ_...	ΗΜΕΡΟΜΗΝΙΑ_...	ΑΦΜ	ΑΡΙΘΜΟΣ_ΤΑΥ...	ΤΗΛ_ΚΙΝΗΤΟ	ΤΗΛ_ΣΤΑΘΕΡΟ
5	ΑΠΟΣΤΟΛΙΔΗΣ	ΓΡΗΓΟΡΗΣ	ΑΓΓΕΛΟΣ	1981-06-05	2012-03-02	64788383	N37728	6993646646	23210-3773
2	ΠΑΠΑΔΟΠΟΥΛΟΣ	ΠΑΝΗΣ	ΧΡΗΣΤΟΣ	1970-01-01	2010-08-25	34564381	X87658	6980864680	23210-5683
3	ΠΑΠΑΜΑΧΑΙΑ	ΜΕΛΑΧΗΣ	ΝΙΚΟΛΑΟΣ	1980-09-08	2009-07-07	78736368	Π63176	6978286253	23210-3628

Με κώδικα SQL:

```
SELECT TOP (100) PERCENT ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ, ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ_ΠΑΤΡΟΣ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΗΜΕΡΟΜΗΝΙΑ_ΠΡΟΣΛΗΨΗΣ, ΕΡΓΑΖΟΜΕΝΟΙ.ΑΦΜ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΑΡΙΘΜΟΣ_ΤΑΥΤΟΤΗΤΑΣ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΤΗΛ_ΚΙΝΗΤΟ, ΕΡΓΑΖΟΜΕΝΟΙ.ΤΗΛ_ΣΤΑΘΕΡΟ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΔΙΕΥΘΥΝΣΗ, ΕΡΓΑΖΟΜΕΝΟΙ.ΠΟΛΗ, ΕΡΓΑΖΟΜΕΝΟΙ.ΜΙΣΘΟΣ,
ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ, ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ AS Expr1,
ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ AS Expr2, ΕΡΓΟ.ΤΙΤΛΟΣ
FROM ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN
ΕΡΓΑΖΕΤΑΙ ON ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ =
ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ INNER JOIN
ΕΡΓΟ ON ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ = ΕΡΓΟ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ
ORDER BY Expr2
```

## Ερώτηση 5

Εμφανίστε αναλυτικά τους εργαζομένους δηλαδή τον ΚΩΔΙΚΟ\_ΕΡΓΑΖΟΜΕΝΟΥ, το ΕΠΙΘΕΤΟ, το ΟΝΟΜΑ, το ΟΝΟΜΑ\_ΠΑΤΡΟΣ, την ΗΜΕΡΟΜΗΝΙΑ\_ΓΕΝΝΗΣΗΣ, την ΗΜΕΡΟΜΗΝΙΑ\_ΠΡΟΣΛΗΨΗΣ, το ΑΦΜ, τον ΑΡΙΘΜΟΣ\_ΤΑΥΤΟΤΗΤΑΣ, το ΤΗΛ\_ΚΙΝΗΤΟ, το ΤΗΛ\_ΣΤΑΘΕΡΟ, την ΔΙΕΥΘΥΝΣΗ, την ΠΟΛΗ, τον ΜΙΣΘΟ και τον ΚΩΔΙΚΟ\_ΤΜΗΜΑΤΟΣ και το ΤΜΗΜΑ που ο μισθός τους είναι μικρότερος από 600 και το τμήμα τους είναι το ΛΟΓΙΣΤΗΡΙΟ.

## Απάντηση 5

Με QBE:

Column Table Output Filter

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
*	ERGAZOM...	ERGAZOM...	<input checked="" type="checkbox"/>			
ΜΙΣΘΟΣ	Expr1	ERGAZOM...	<input checked="" type="checkbox"/>			< 600
ΟΝΟΜΑ	Expr2	TMHMA	<input checked="" type="checkbox"/>			= 'ΛΟΓΙΣΤΗΡΙΟΥ'

```

SELECT dbo.ERGAZOMENOI.*, dbo.ERGAZOMENOI.ΜΙΣΘΟΣ AS Expr1, dbo.TMHMA.ΟΝΟΜΑ AS Expr2
FROM
  dbo.ERGAZOMENOI INNER JOIN
  dbo.TMHMA ON (dbo.ERGAZOMENOI.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ = dbo.TMHMA.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ)
WHERE
  (dbo.ERGAZOMENOI.ΜΙΣΘΟΣ < 600) AND (dbo.TMHMA.ΟΝΟΜΑ = 'ΛΟΓΙΣΤΗΡΙΟΥ')
  
```

ΚΩΔΙΚΟΣ_ΕΡΓ...	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ_...	ΗΜΕΡΟΜΗΝΙΑ_...	ΑΦΜ	ΑΡΙΘΜΟΣ_ΤΑΥ...	ΤΗΛ_ΚΙΝΗΤΟ	ΤΗΛ_ΣΤΑΒΕ
5	ΑΓΙΟΣΤΟΛΑΗΣ	ΓΡΗΓΟΡΗΣ	ΑΓΓΕΛΟΣ	1981-06-05	2012-03-02	64788383	N37728	6993646646	23210-3773
6	ΠΑΠΑ	ΜΑΡΙΑ	ΝΙΚΟΛΑΟΣ	1982-09-07	2013-07-06	69803333	K27272	6983738374	23210-3737

Με κώδικα SQL:

```

SELECT ERGAZOMENOI.*, ERGAZOMENOI.ΜΙΣΘΟΣ AS Expr1, TMHMA.ΟΝΟΜΑ
AS Expr2
FROM ERGAZOMENOI INNER JOIN
TMHMA ON ERGAZOMENOI.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ = TMHMA.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ
WHERE (ERGAZOMENOI.ΜΙΣΘΟΣ < 600) AND (TMHMA.ΟΝΟΜΑ =
'ΛΟΓΙΣΤΗΡΙΟΥ')
  
```

### Ερώτηση 6

Εμφανίστε το ΟΝΟΜΑ και το ΕΠΙΘΕΤΟ των εργαζομένων που είναι από Σέρρες και η δραστηριότητα τους είναι η ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ ταξινομημένους κατά φθίνουσα σειρά.

### Απάντηση 6

Με QBE:

Column Table Output Filter

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
ΟΝΟΜΑ		ERGAZOM...	<input checked="" type="checkbox"/>			
ΕΠΙΘΕΤΟ		ERGAZOM...	<input checked="" type="checkbox"/>			
ΠΟΛΗ		ERGAZOM...	<input checked="" type="checkbox"/>			= 'ΣΕΡΡΕΣ'
ΔΡΑΣΤΗΡΙΟΤΗΤΑ		TMHMA	<input checked="" type="checkbox"/>			= 'ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ'

```

SELECT dbo.ERGAZOMENOI.ΟΝΟΜΑ, dbo.ERGAZOMENOI.ΕΠΙΘΕΤΟ, dbo.ERGAZOMENOI.ΠΟΛΗ, dbo.TMHMA.ΔΡΑΣΤΗΡΙΟΤΗΤΑ
FROM
  dbo.ERGAZOMENOI INNER JOIN
  dbo.TMHMA ON (dbo.ERGAZOMENOI.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ = dbo.TMHMA.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ)
WHERE
  (dbo.TMHMA.ΔΡΑΣΤΗΡΙΟΤΗΤΑ = 'ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ') AND (dbo.ERGAZOMENOI.ΠΟΛΗ = 'ΣΕΡΡΕΣ')
  
```

ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΠΟΛΗ	ΔΡΑΣΤΗΡΙΟΤΗΤΑ
ΜΙΧΑΛΗΣ	ΠΑΠΑΜΙΧΑΗΛ	ΣΕΡΡΕΣ	ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ
ΕΛΕΝΗ	ΔΗΜΗΤΡΙΟΥ	ΣΕΡΡΕΣ	ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ

Με κώδικα SQL:

```
SELECT ΕΡΓΑΖΟΜΕΝΟΙ.ΟΝΟΜΑ, ΕΡΓΑΖΟΜΕΝΟΙ.ΕΠΙΘΕΤΟ,  
ΕΡΓΑΖΟΜΕΝΟΙ.ΠΟΛΗ, ΤΜΗΜΑ.ΔΡΑΣΤΗΡΙΟΤΗΤΑ  
FROM ΕΡΓΑΖΟΜΕΝΟΙ INNER JOIN  
ΤΜΗΜΑ ON ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ = ΤΜΗΜΑ.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ  
WHERE (ΤΜΗΜΑ.ΔΡΑΣΤΗΡΙΟΤΗΤΑ = 'ΒΕΛΤΙΩΣΗ ΕΜΠΕΙΡΙΑΣ ΧΡΗΣΤΗ') AND  
(ΕΡΓΑΖΟΜΕΝΟΙ.ΠΟΛΗ = 'ΣΕΡΡΕΣ')
```

## Παραδείγματα συναρτήσεων με συνάθροιση

### Ερώτηση 1

Εμφανίστε το πλήθος των εργαζομένων που κατάγονται από Σέρρες.

### Απάντηση 1

```
SELECT COUNT (ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ) AS ΠΛΗΘΟΣ  
FROM ΕΡΓΑΖΟΜΕΝΟΙ  
WHERE ΠΟΛΗ='ΣΕΡΡΕΣ' ;
```

### Ερώτηση 2

Εμφανίστε το πλήθος των εργαζομένων που είναι από Σέρρες και το τμήμα τους είναι το τμήμα ΣΧΕΔΙΑΣΜΟΥ.

### Απάντηση 2

```
SELECT COUNT( ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ) AS ΠΛΗΘΟΣ  
FROM ΕΡΓΑΖΟΜΕΝΟΙ, ΤΜΗΜΑ  
WHERE  
ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ=ΤΜΗΜΑ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ AND  
ΕΡΓΑΖΟΜΕΝΟΙ.ΠΟΛΗ='ΣΕΡΡΕΣ' AND ΤΜΗΜΑ.ΟΝΟΜΑ='ΣΧΕΔΙΑΣΜΟΥ' ;
```

### Ερώτηση 3

Εμφανίστε τον μέσο όρο του μισθού των εργαζομένων.

### Απάντηση 3

```
SELECT AVG (ΜΙΣΘΟΣ) AS ΜΟ  
FROM ΕΡΓΑΖΟΜΕΝΟΙ;
```



#### **Ερώτηση 4**

Εμφανίστε τον μέγιστο μισθό των εργαζομένων.

#### **Απάντηση 4**

```
SELECT MAX (ΜΙΣΘΟΣ) AS ΜΕΓΙΣΤΟΣ  
FROM ΕΡΓΑΖΟΜΕΝΟΙ;
```

#### **Ερώτηση 5**

Εμφανίστε τους εργαζομένους που δουλεύουν στο τμήμα ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ.

#### **Απάντηση 5**

```
SELECT ΕΡΓΑΖΟΜΕΝΟΙ.*  
FROM ΕΡΓΑΖΟΜΕΝΟΙ, ΤΜΗΜΑ  
WHERE ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ=ΤΜΗΜΑ.ΚΩΔΙΚΟΣ_ΤΜΗΜΑΤΟΣ AND  
ΤΜΗΜΑ.ΟΝΟΜΑ LIKE 'ΑΝΑΠΤΥΞΗ%';
```

#### **Ερώτηση 6**

Εμφανίστε τους εργαζομένους που το έργο τους ξεκίνησε στη 1/1/2014.

#### **Απάντηση 6**

```
SELECT ΕΡΓΑΖΟΜΕΝΟΙ.*, ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΘΕΣΗΣ  
FROM ΕΡΓΑΖΟΜΕΝΟΙ, ΕΡΓΑΖΕΤΑΙ, ΕΡΓΟ  
WHERE ΕΡΓΑΖΟΜΕΝΟΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ=ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΑΖΟΜΕΝΟΥ  
AND ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ=ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ AND  
ΕΡΓΑΖΕΤΑΙ.ΚΩΔΙΚΟΣ_ΕΡΓΟΥ='1/1/2014';
```

#### **Ερώτηση 7**

Εμφανίστε τον μέσο όρο του μισθού των εργαζομένων όπου ο μισθός είναι μεγαλύτερος ή ίσος με 400.

#### **Απάντηση 7**

```
SELECT AVG (ΜΙΣΘΟΣ) AS ΜΟ  
FROM ΕΡΓΑΖΟΜΕΝΟΙ;  
WHERE ΜΙΣΘΟΣ >= 400;
```

## Ενότητα 4 – Διαφάνειες του θεωρητικού μέρους του μαθήματος «Ειδικά θέματα Βάσεων Δεδομένων»

### Παρουσίαση 1 – Εισαγωγή και αρχιτεκτονική των βάσεων δεδομένων

# Ενότητα 1

---

## Εισαγωγή και αρχιτεκτονική των βάσεων δεδομένων

Δρ. Τιμπίρης Αλκιβιάδης



## Περιεχόμενα ενότητας

---

- Σκοποί ενότητας
- Σκοπός των συστημάτων βάσεων δεδομένων
- Προβλήματα κατά την επεξεργασία των αρχείων
- Αφαιρετική άποψη δεδομένων
- Μοντέλα δεδομένων
- Εννοιολογικά μοντέλα βασισμένα σε αντικείμενα
- Μοντέλο οντοτήτων συσχετίσεων
- Διάγραμμα μοντέλου οντοτήτων-συσχετίσεων
- Αντικειμενοστραφές μοντέλο
- Σχεσιακό μοντέλο
- Φυσικό μοντέλο
- Μετά-δεδομένα
- Στιγμιότυπα κ σχήματα
- Εσωτερικό σχήμα



## Περιεχόμενα ενότητας

---

- Εννοιολογικό σχήμα
- Εξωτερικά σχήματα
- Παραδείγματα εντολών SQL στα τρία επίπεδα
- Ανεξαρτησία δεδομένων στην αρχιτεκτονική τριών σχημάτων
- Ευρετήρια
- Γλώσσες βάσεων δεδομένων
- Συστατικά στοιχεία ενός συστήματος διαχείρισης βάσεων δεδομένων
- Έννοιες βάσης δεδομένων

## Σκοποί ενότητας

---

Στο κεφάλαιο αυτό γίνεται μια εισαγωγή στην πολλαπλού-επιπέδου αρχιτεκτονική των βάσεων δεδομένων όπου κάθε επίπεδο αντιπροσωπεύει ένα επίπεδο αφαίρεσης, εξηγώντας το εσωτερικό – εννοιολογικό και εξωτερικό επίπεδο. Παρουσιάζονται επίσης τα συστατικά μέρη ενός συστήματος διαχείρισης βάσης δεδομένων (DBMS).

## Σκοπός των συστημάτων βάσεων δεδομένων

---

Ένα σύστημα βάσεων δεδομένων έχει ως στόχο να παράσχει ένα περιβάλλον που είναι και βολικό και αποδοτικό για την:

- ανάκτηση πληροφοριών από την βάση δεδομένων
- την αποθήκευση πληροφοριών στη βάση δεδομένων

## Προβλήματα κατά την επεξεργασία των αρχείων

---

- δυσκολία στην πρόσβαση των δεδομένων
- πλεονασμός και ασυνέπεια δεδομένων
- πολλαπλοί χρήστες
- απομόνωση δεδομένων
- προβλήματα ασφάλειας
- προβλήματα ακεραιότητας

# Αφαιρετική άποψη δεδομένων

---

Ο σημαντικότερος σκοπός ενός συστήματος βάσεων δεδομένων είναι να παράσχει στους χρήστες μια **αφαιρετική άποψη** του συστήματος. Το σύστημα κρύβει ορισμένες λεπτομέρειες για το πώς τα δεδομένα αποθηκεύονται και διατηρούνται .

# Αφαιρετική άποψη δεδομένων

---

Η πολυπλοκότητα των βάσεων δεδομένων πρέπει να κρυφτεί και για αυτόν το λόγο υπάρχουν διάφορα επίπεδα αφαίρεσης:

**Φυσικό επίπεδο:** είναι το χαμηλότερο επίπεδο αφαίρεσης.

- περιγράφει πώς αποθηκεύονται τα δεδομένα
- εδώ περιγράφονται λεπτομερώς όλες οι σύνθετες δομές χαμηλού επιπέδου

**Εννοιολογικό επίπεδο:** το αμέσως υψηλότερο επίπεδο αφαίρεσης.

- περιγράφει τι είδους δεδομένα αποθηκεύονται
- περιγράφει τις συσχετίσεις μεταξύ αυτών των δεδομένων

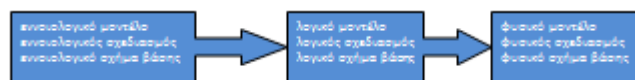
**Επίπεδο άποψης:** το πιο υψηλό επίπεδο.

- περιγράφει τμήματα της βάσης δεδομένων για κάθε συγκεκριμένη ομάδα χρηστών
- μπορεί να είναι πολλές διαφορετικές απόψεις μιας βάσης δεδομένων

# Μοντέλα δεδομένων

Τα **μοντέλα δεδομένων** είναι μια συλλογή από εργαλεία για την περιγραφή δομών δεδομένων, συσχετίσεων δεδομένων, τη σημασιολογία δεδομένων και τους περιορισμούς δεδομένων. Υπάρχουν τρεις διαφορετικές ομάδες που αντιστοιχούν στις τρεις φάσεις σχεδιασμού:

- εννοιολογικά μοντέλα βασισμένα σε αντικείμενα.
- λογικά μοντέλα βασισμένα σε εγγραφές.
- φυσικά μοντέλα δεδομένων.



## Εννοιολογικά μοντέλα βασισμένα σε αντικείμενα

Τα μοντέλα αυτά περιγράφουν τα δεδομένα στο εννοιολογικό επίπεδο και στα επίπεδα όψεων. Επίσης παρέχουν αρκετά εύκαμπτες δυνατότητες δόμησης και επιτρέπουν ρητά τον καθορισμό περιορισμών στα δεδομένα. Υπάρχουν πάνω από 30 τέτοια μοντέλα όπως τα:

- Μοντέλο οντοτήτων-συσχετίσεων (Entity-relationship model).
- Εκτεταμένο μοντέλο οντοτήτων-συσχετίσεων (EER model).
- Αντικειμενοστρεφές μοντέλο (Object-oriented model).
- Διαδικό μοντέλο (Binary model).
- Σημασιολογικό μοντέλο δεδομένων (Semantic data model).
- Μοντέλο Infological (Infological model).
- Συναρτησιακό μοντέλο δεδομένων (Functional data model)

## Εννοιολογικά μοντέλα βασισμένα σε αντικείμενα

---

Τα μοντέλα αυτά περιγράφουν τα δεδομένα στο εννοιολογικό επίπεδο και στα επίπεδα όψεων. Επίσης παρέχουν αρκετά εύκαμπτες δυνατότητες δόμησης και επιτρέπουν ρητά τον καθορισμό περιορισμών στα δεδομένα. Υπάρχουν πάνω από 30 τέτοια μοντέλα όπως τα:

- Μοντέλο οντοτήτων-συσχετίσεων (Entity-relationship model).
- Εκτεταμένο μοντέλο οντοτήτων-συσχετίσεων (EER model).
- Αντικειμενοστραφές μοντέλο (Object-oriented model).
- Διαδικό μοντέλο (Binary model).
- Σημασιολογικό μοντέλο δεδομένων (Semantic data model).
- Μοντέλο Infological (Infological model).
- Συναρτησιακό μοντέλο δεδομένων (Functional data model)

## Μοντέλο οντοτήτων συσχετίσεων

---

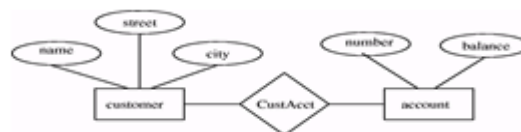
Το μοντέλο οντοτήτων-συσχετίσεων είναι βασισμένο στην αντίληψη ότι ο κάθε μικρόκοσμος που θα παρασταθεί με τη βάση δεδομένων αποτελείται από μια συλλογή από βασικά αντικείμενα τις οντότητες και συσχετίσεις μεταξύ αυτών των αντικειμένων.

- **μια οντότητα** είναι ένα διακριτό αντικείμενο
- **μια συσχέτιση** είναι μία σύνδεση μεταξύ διάφορων οντοτήτων

## Διάγραμμα μοντέλου οντοτήτων-συσχετίσεων

Η εννοιολογική δομή μιας βάσης δεδομένων μπορεί να εκφραστεί γραφικά από ένα **διάγραμμα οντοτήτων-συσχετίσεων** :

- **ορθογώνια**: αντιπροσωπεύουν τους τύπους οντοτήτων.
- **ελλείψεις**: αντιπροσωπεύουν τις ιδιότητες.
- **ρόμβοι**: αντιπροσωπεύουν τους τύπους συσχετίσεων μεταξύ των τύπων οντοτήτων.
- **γραμμές**: συνδέουν τους τύπους οντοτήτων με τα γνωρίσματά τους και με τους τύπους συσχετίσεών τους.



## Αντικειμενοστραφές μοντέλο

Το αντικειμενοστραφές μοντέλο είναι βασισμένο σε μια συλλογή αντικειμένων, όπως και το μοντέλο οντοτήτων-συσχετίσεων.

- Οι τιμές που αποθηκεύονται στις **μεταβλητές στιγμιότυπων** μέσα στο αντικείμενο είναι επίσης αντικείμενα.
- Ένα αντικείμενο περιέχει τμήματα κώδικα που ενεργούν στο ίδιο το αντικείμενο. Τα αντικείμενα αυτά περιέχουν αντικείμενα σε ένα αυθαίρετα βαθύ επίπεδο εμφωλίωσης. Αυτά τα τμήματα κώδικα καλούνται **μέθοδοι**.
- Τα αντικείμενα που περιέχουν ίδιους τύπους τιμών και ίδιες μεθόδους ομαδοποιούνται **σε κλάσεις**.



## Αντικειμενοστραφές μοντέλο

---

- Μια κλάση μπορεί να αντιμετωπισθεί ως καθορισμός τύπου του αντικειμένου.
- Αναλογία: η έννοια ενός αφηρημένου τύπου δεδομένων στις γλώσσες προγραμματισμού.
- Ο μόνος τρόπος ένα αντικείμενο μπορεί να έχει πρόσβαση στα δεδομένα ενός άλλου αντικειμένου είναι με επίκληση της μεθόδου του άλλου αντικειμένου
- Τα εσωτερικά μέρη του αντικειμένου, οι μεταβλητές στιγμιότυπων και ο κώδικας της μεθόδου, δεν είναι ορατά εξωτερικά.
- Το αποτέλεσμα είναι δύο επίπεδα αφαίρεσης δεδομένων.

## Σχεσιακό μοντέλο

---

Στο σχεσιακό μοντέλο τα δεδομένα και οι συσχετίσεις αντιπροσωπεύονται από μια συλλογή από **κανονικοποιημένους πίνακες (σχέσεις)**. Κάθε κανονικοποιημένος πίνακας έχει διάφορες στήλες με μοναδικά ονόματα.

## Μετά-δεδομένα

---

Τα μετά-δεδομένα είναι ειδικές πληροφορίες για τα δεδομένα που υπάρχουν στη βάση.

Πώς και που αποθηκεύονται τα δεδομένα

- δομές αρχείων
- εγγραφές
- ονόματα των αρχείων, τύποι δεδομένων
- μορφή (format) των τύπων δεδομένων

Πληροφορίες σχετικά με την αντιστοιχία μεταξύ των διαφορετικών σχημάτων.

Τα μετά-δεδομένα αποθηκεύονται στον **κατάλογο συστήματος** ή λεξικό δεδομένων.

Οι κατάλογοι σε ένα σχεσιακό σύστημα αποθηκεύονται και οι ίδιοι σαν σχέσεις.



## Στιγμιότυπα κ σχήματα

---

**Σχήμα βάσης δεδομένων:**

- Η γενική δομή της βάσης καλείται σχήμα και ορίζεται κατά τη δημιουργία της.
- Ένα σχήμα βάσης δεδομένων δεν αλλάζει συχνά.

**Στιγμιότυπο βάσης δεδομένων:**

- Οι πληροφορίες σε μια βάση δεδομένων σε ένα συγκεκριμένο χρονικό σημείο καλούνται στιγμιότυπα της βάσης δεδομένων.
- Κάθε αλλαγή των δεδομένων δημιουργεί καινούργια στιγμιότυπα της βάσης δεδομένων

Ορισμοί τύπων δεδομένων → σχήμα

Τιμή μιας μεταβλητής → στιγμιότυπο



## Εσωτερικό σχήμα

---

Το εσωτερικό σχήμα περιγράφει τις δομές αποθήκευσης και τα μονοπάτια πρόσβασης προς τη φυσική βάση δεδομένων και καθορίζεται συνήθως μέσω της γλώσσας ορισμού δεδομένων (DDL) του συστήματος διαχείρισης βάσεων δεδομένων.

## Εννοιολογικό σχήμα

---

Ένα εννοιολογικό σχήμα αποτελεί:

- Ένα κοινό βασικό μοντέλο για όλους τους χρήστες της λογικής περιγραφής της βάσης δεδομένων.
- Το επίπεδο αφαίρεσης που αντιστοιχεί "στον πραγματικό κόσμο".
- Το σχήμα που δημιουργείται με τη γλώσσα ορισμού δεδομένων σύμφωνα με ένα συγκεκριμένο μοντέλο δεδομένων.

# Εξωτερικά σχήματα

Τα εξωτερικά σχήματα περιγράφουν τις διαφορετικές απόψεις της εννοιολογικής βάσης δεδομένων σε σχέση με αυτά τα δεδομένα που οι διαφορετικές ομάδες χρηστών θα ήθελαν ή έχουν την άδεια να δουν.

## Παραδείγματα εντολών SQL στα τρία επίπεδα

<u>Εξωτερικό επίπεδο</u>	<pre>CREATE VIEW ALL_SUPPLIERS AS SELECT * FROM FOIHTES</pre>
<u>Εννοιολογικό επίπεδο</u>	<pre>CREATE TABLE FOIHTES ( AEM NUMBER(4) NOT NULL, EPITHETO VARCHAR(25), POLY VARCHAR(20) );</pre>
<u>Εσωτερικό επίπεδο</u>	<pre>CREATE INDEX FOIHTES_EPITHETO ON FOIHTES (EPITHETO) ; CREATE UNIQUE INDEX AEM_No ON FOIHTES (AEM) ;</pre>

## Ανεξαρτησία δεδομένων στην αρχιτεκτονική τριών σχημάτων

---

Η δυνατότητα τροποποίησης ενός σχήματος της βάσης από ένα επίπεδο χωρίς επιρροή ενός σχήματος σε ένα πιο υψηλό επίπεδο καλείται **ανεξαρτησία δεδομένων**. Υπάρχουν δύο τύπο ανεξαρτησίας δεδομένων:

- **λογική ανεξαρτησία δεδομένων** η δυνατότητα να τροποποιηθεί το εννοιολογικό σχήμα χωρίς να αναγκάσει τα προγράμματα εφαρμογών να ξαναγραφούν. Οι τροποποιήσεις σε αυτό το επίπεδο γίνονται συνήθως όταν αλλάζει η λογική δομή της βάσης δεδομένων.
- **φυσική ανεξαρτησία δεδομένων** η δυνατότητα να τροποποιηθεί το φυσικό σχήμα χωρίς να αναγκάσει τα προγράμματα εφαρμογών να ξαναγραφούν. Οι τροποποιήσεις γίνονται συνήθως για να βελτιωθεί η απόδοση.

Η λογική ανεξαρτησία δεδομένων είναι πιο δύσκολο να επιτύχει.

## Ευρετήρια

---

Ένα **ευρετήριο (index)** είναι μια βοηθητική δομή αρχείου που κάνει πιο αποδοτική την αναζήτηση μιας εγγραφής σε ένα αρχείο.

- Το ευρετήριο καθορίζεται σε ένα **γνώρισμα** του αρχείου.
- Συχνά αποκαλείται **access path** (μονοπάτι πρόσβασης) στο γνώρισμα.

# Γλώσσες βάσεων δεδομένων

---

Εκτός της γλώσσας SQL στις βάσεις δεδομένων χρησιμοποιούνται και οι εξής γλώσσες:

- Γλώσσα ορισμού αποθήκευσης -Storage Definition Language (SDL)
- Γλώσσα ορισμού δεδομένων -Data Definition Language (DDL)
- Γλώσσα ορισμού απόψεων -View Definition Language (VDL)
- Γλώσσα χειρισμού δεδομένων -Data Manipulation Language (DML)



## Συστατικά στοιχεία ενός συστήματος διαχείρισης βάσεων δεδομένων

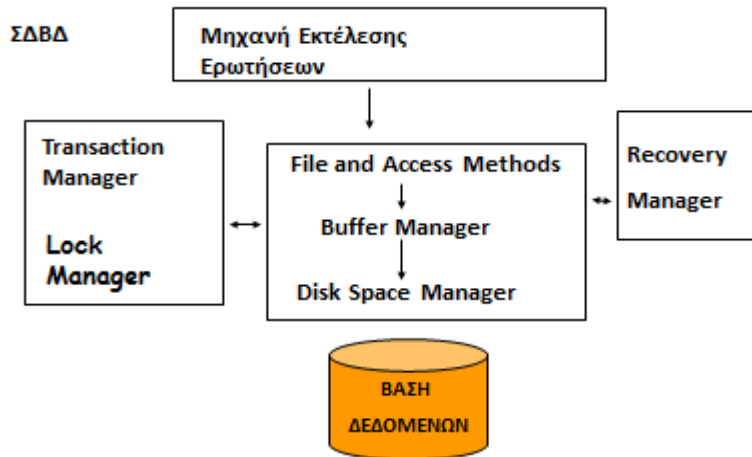
---

Query processor

- DML compiler
- Embedded DML precompiler
- DDL interpreter
- Query processing unit
- Storage manager
- Authorization and integrity control
- Transactions management
- File management
- Buffer management
- Physical storage
- data files, meta-data (data dictionary), index, statistics



# Έννοιες βάσης δεδομένων



## Ενότητα 2

---

### Εννοιολογικά – Λογικά μοντέλα

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Ανάπτυξη βάσεων δεδομένων
- Πλήρης διαδικασία ανάπτυξης DBMS
- Εννοιολογικός σχεδιασμός
- Λογικός σχεδιασμός
- Φυσικός σχεδιασμός
- E-R Μοντέλο-Δομές- Δομικοί περιορισμοί
- Γραμμικός συμβολισμός E-R μοντέλου
- Παράδειγμα E-R μοντέλου
- Γραφική αναπαράσταση τύπων οντοτήτων και συσχετίσεων
- Ρόλοι και αυτοσυσχετίσεις
- Λογικά Μοντέλα
- Λογικός σχεδιασμός
- Σχεδιασμός σχεσιακού μοντέλου
- Μετατροπή E-R σε σχεσιακό μοντέλο





# Σκοποί ενότητας

---

Στο κεφάλαιο αυτό αναλύουμε τα εννοιολογικά μοντέλα που χρησιμοποιούνται κατά τον σχεδιασμό των βάσεων δεδομένων και ειδικότερα το μοντέλο οντοτήτων – συσχετίσεων (Entity-relationship model), το εκτεταμένο μοντέλο οντοτήτων-συσχετίσεων (EER model) και το αντικειμενοστραφές (Object-oriented model).

Επίσης θα αναλυθεί η χρήση των λογικών μοντέλων κατά την υλοποίηση των βάσεων δεδομένων και ειδικότερο το σχεσιακό μοντέλο (Relational Model).



## Ανάπτυξη βάσεων δεδομένων

---

Η διαδικασία μετατροπής της γνώσης ενός πραγματικού μικρόκοσμου σε μια βάση που θα την διαχειρίζεται ένα DBMS διακρίνεται σε συνεχόμενες αυτοτελείς διαδικασίες, κάθε μια εκ των οποίων δημιουργεί ενδιάμεσες περιγραφές. Αυτές οι φάσεις είναι οι εξής:

- **Συλλογή Απαιτήσεων και Ανάλυση** (*Requirements Collection and Analysis*)
- **Εννοιολογικός Σχεδιασμός Βάσης** (*Conceptual Database Design*)
- **Λογικός Σχεδιασμός Βάσης** (*Logical Database Design*)
- **Φυσικός Σχεδιασμός Βάσης** (*Physical Database Design*)
- **Πλήρωση της Βάσης με Δεδομένα** (*Database Loading*)



## Διαδικασία Ανάπτυξης ΒΔ



## Διαδικασία ανάπτυξης βάσης δεδομένων

Το αρχικό στάδιο για την ανάπτυξη μίας βάσης δεδομένων είναι η εξόρυξη των απαιτήσεων της κυρίως από περιγραφές του μικρόκοσμου. Η διαδικασία είναι δύσκολη εννοιολογικά και απαιτεί εμπειρία και επιχειρηματική γνώση.

Περιγραφές  
Μικρόκοσμου



Απαιτήσεις  
της βάσης

## Διαδικασία ανάπτυξης βάσης δεδομένων

---

Η διαδικασία συνεχίζεται με τον σχεδιασμό της βάσης. Σχεδιασμός είναι η διαδικασία δημιουργίας του σχήματος χρησιμοποιώντας ένα επιλεγμένο μοντέλο. Υπάρχουν τρεις αυτοτελείς σχεδιασμοί που δημιουργούν σχήματα:

- Εννοιολογικός σχεδιασμός
- Λογικός σχεδιασμός
- Φυσικός σχεδιασμός

## Διαδικασία ανάπτυξης βάσης δεδομένων

---

Η τελική διαδικασία που πρέπει να ακολουθηθεί για την ανάπτυξη της βάσης δεδομένων είναι :

- Η πλήρωση της ΒΔ με δεδομένα (data loading)



**ΣΧΟΛΙΟ:** Σε όλα τα μεγάλα DBMS υπάρχουν προγράμματα για την μαζική πλήρωση.

# Εννοιολογικός σχεδιασμός

---

Ο εννοιολογικός σχεδιασμός είναι :

- η προσπάθεια για την διαχώριση των εννοιών.
- η εύρεση και η καταγραφή των οντοτήτων της βάσης των μεταξύ τους συσχετίσεων όπως και τον εννοιολογικών κανόνων.

# Λογικός σχεδιασμός

---

Ο Λογικός σχεδιασμός είναι :

- Η διαδικασία μετατροπής ενός εννοιολογικού μοντέλου σε τυπικά σχήματα εκφρασμένα στο επιλεγόμενο μοντέλο δεδομένων.

# Φυσικός σχεδιασμός

---

Ο Φυσικός σχεδιασμός είναι :

- Η διαδικασία προδιαγραφών των δομών μνήμης που υλοποιούν τον λογικό σχεδιασμό και τον οργανώσεων αυτών καθώς επίσης και των δομών ευρετηρίων που αποτελούν το εσωτερικό σχήμα.

## Μοντέλο οντοτήτων συσχετίσεων

---

Το μοντέλο οντοτήτων-συσχετίσεων E-R έγινε γενικά αποδεκτό (στην δεκαετία του 80) από τους ερευνητές για την:

- Απλότητα του
- Την σαφήνεια του
- Τον γραφικό συμβολισμό του

Το E-R μοντέλο είναι ένα **διαισθητικό** μοντέλο που επιδιώκει να προσδιορίσει **αφαιρετικά** τις πληροφορίες που αποθηκεύει και οργανώνει μία βάση δεδομένων.

# E-R Μοντέλο-Δομές

---

Υπάρχουν δύο βασικές έννοιες:

- **Οντότητες** → Συγκεκριμένα αντικείμενα που υπάρχουν και μπορούν να αναπαρασταθούν στις Βάσεις δεδομένων.
- **Συσχετίσεις** → Ειδικά αντικείμενα που αντιστοιχούν σε δύο ή περισσότερες ξεχωριστές οντότητες με ένα συγκεκριμένο νόημα.



# E-R Μοντέλο-Δομές

---

Οι οντότητες και οι συσχετίσεις μπορούν να έχουν **ΓΝΩΡΙΣΜΑΤΑ / ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ (ATTRIBUTES)**, που είναι οι ιδιότητες που τα χαρακτηρίζουν.

**Παράδειγμα:** Ένας εργαζόμενος έχει γνωρίσματα( Όνομα, Επίθετο, Ηλικία).

Συνήθως τα γνωρίσματα των συσχετίσεων προσδιορίζουν τρία πράγματα:

- Ποιος έκανε την συσχέτιση
- Πότε την έκανε την συσχέτιση
- Πότε παύει να ισχύει



# E-R Μοντέλο-Δομές

---

**Πεδία τιμών** είναι σύνολα τιμών (DOMAINS) για τα γνωρίσματα.

**Παράδειγμα:** Domain of name= το σύνολο των ονομάτων

**Είδη γνωρισμάτων:**

**ΑΠΛΑ:** μια οντότητα έχει ατομική τιμή

**ΣΥΝΘΕΤΑ:** το γνώρισμα αποτελείται από περισσότερες τιμές

**ΠΛΕΙΟΤΙΜΑ (MULTI-VALUED):** το γνώρισμα έχει πολλαπλές τιμές

## Δομικοί περιορισμοί-Συσχετίσεις

---

**Ιδιότητες των τύπων συσχετίσεων:**

- Περισσότεροι του ενός τύποι συσχετίσεων μπορεί να υπάρχουν μεταξύ των ιδίων τύπων οντοτήτων (multiple relationships).
- Μια συσχέτιση μπορεί να συνδέει δύο οντότητες που ανήκουν στον ίδιο τύπο οντοτήτων (αποκαλείται, ένας αναδρομικός τύπος συσχετίσεων - recursive relationship type).
- Ένας τύπος Συσχετίσεων έχει ΒΑΘΜΟ (relationship degree):
  - 2 τύποι οντοτήτων ---> binary relationship
  - 3 τύποι οντοτήτων ---> ternary relationship
  - N τύποι οντοτήτων ---> N-ary relationship
- **Περιορισμός Ύπαρξης** (existence dependency) ορίζει αν η συμμετοχή μιας οντότητας στον τύπο συσχέτισης είναι ΟΛΙΚΗ ή ΜΕΡΙΚΗ (total or partial).

## Δομικοί περιορισμοί-Γνωρίσματα

---

Ιδιότητες των τύπων γνωρισμάτων:

- Ένα **γνώρισμα** (ή σύνολο Γνωρισμάτων) ενός τύπου **οντοτήτων/συσχετίσεων** για το οποίο **κάθε οντότητα / συσχέτιση** στο σύνολο πρέπει να έχει **μοναδική τιμή** (*unique value(s)*) είναι **κλειδί (key)** (ή *superkey*).
- Ένα **υποψήφιο κλειδί (candidate key)** είναι ένα **ελάχιστο (minimal) κλειδί** (δηλαδή, κανένα υποσύνολο των γνωρισμάτων του δεν είναι και αυτό κλειδί).
- Το **κύριο κλειδί (primary key)** είναι ένα από τα υποψήφια κλειδιά που ορίζεται σαν αναγνωριστής (*identifier*) για τον τύπο οντοτήτων / συσχετίσεων.
- Ένα **ξένο κλειδί (foreign key)** είναι ένα σύνολο ενός ή περισσότερων γνωρισμάτων ενός τύπου οντοτήτων (ή συσχετίσεων) που αντιστοιχεί σε ένα **κύριο κλειδί** για κάποιον άλλο τύπο οντοτήτων (ή συσχετίσεων).

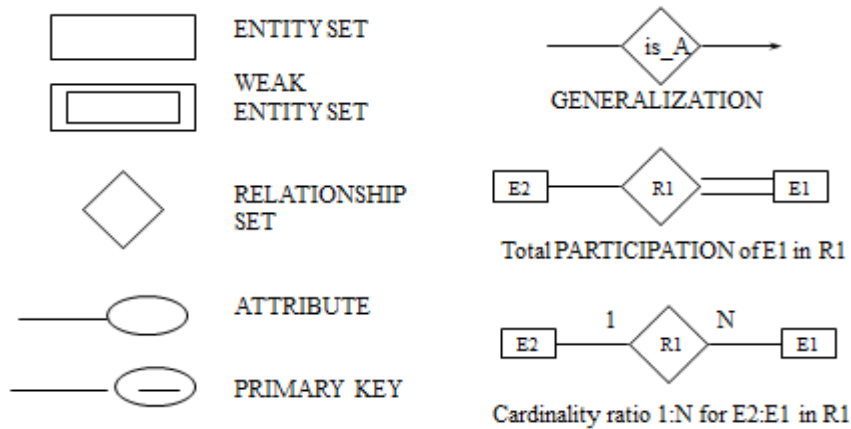
## Άλλες έννοιες για το E-R μοντέλο

---

- **Αδύναμοι τύποι οντοτήτων (ή εξαρτώμενοι):** είναι ένας τύπος χωρίς κλειδί.
- **Η συσχέτιση γενίκευση:** η θεώρηση ενός συνόλου τύπων οντοτήτων ως ένα γενικευμένο τύπο.
  - Το **βασικό πλεονέκτημα** αυτού είναι η ιδιότητα της **ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑΣ (Inheritance)** των Γνωρισμάτων.
  - Τα Γνωρίσματα που είναι κοινά για όλα τα αντικείμενα και δεν χρειάζεται να τα επαναλάβουμε για να κληρονομούνται παίρνουν αυτήν την ιδιότητα.



## Συμβολισμός E-R μοντέλου

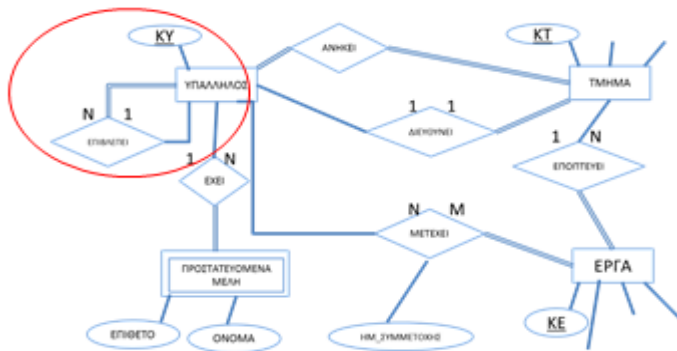


## Παράδειγμα E-R μοντέλου



- Παραλληλόγραμμα για τύπους οντοτήτων
- Ελλείψεις για γνωρίσματα
  - Διπλές ελλείψεις για πλειότιμα γνωρίσματα.
  - Διακεκομμένες ελλείψεις για παραγόμενα γνωρίσματα.
  - Υπογράμμιση για πρωτεύοντα κλειδιά
  - Ρόμβοι για τύπους συσχετίσεων
- Ακμές για σύνδεση τύπων οντοτήτων με τα γνωρίσματα και τύπους συσχετίσεων

## Ρολί και αυτοσυσχετίσεις



Με τον όρο **ρολί** εννοούνται οι ταμπέλες που τοποθετούνται στις ακμές ενός τύπου συσχέτισης για να δειχθεί η σημασία της συμμετοχής ενός τύπου οντοτήτων στον εν λόγω τύπο συσχέτισεων.

## Σχεδιασμός σχεσιακού μοντέλου

Για τον σχεδιασμό ενός σχεσιακού-λογικού μοντέλου υπάρχουν οι εξής προσεγγίσεις:

- Από εννοιολογικό σχεδιασμό (E-R) σε σχεσιακό σχεδιασμό.
- Δημιουργία του σχεσιακού μοντέλου από το μηδέν.
  - Καθολική σχέση που τα περιέχει όλα
  - Αποσύνθεση σε περισσότερες σχέσεις με βάση τη θεωρία κανονικοποίησης.

# Λογικός σχεδιασμός

---

**Εξάρτηση από το DBMS:** Το βασικό μοντέλο για την μετατροπή ενός σχήματος E-R είναι το σχεσιακό, μία δεύτερη επιλογή είναι η αντικειμενοστραφής. Παλαιότερα τα μοντέλα που χρησιμοποιούνταν ήταν το μοντέλο δικτύου και το ιεραρχικό.



## Μετατροπή E-R σε σχεσιακό μοντέλο

---

Για την **μετατροπή** του E-R σε σχεσιακό μοντέλο υπάρχει ένας **γενικός κανόνας**: Για τον κάθε τύπο συσχετίσεων και για τον κάθε τύπο οντοτήτων δημιουργείται ένα σχήμα σχέσης που παίρνει το όνομα του αντίστοιχου τύπου.

## Μετατροπή E-R σε σχεσιακό μοντέλο

### Οντότητες:

Για κάθε τύπο οντοτήτων E δημιουργείται ένα σχήμα R με τα ίδια γνωρίσματα. Ένα για κάθε απλό γνώρισμα του E.

### Παράδειγμα:



**Σχόλιο!** Οι οντότητες είναι οι εργαζόμενοι, τα τμήματα και τα έργα.

## Μετατροπή E-R σε σχεσιακό μοντέλο

### Οντότητες:

Για κάθε (ασθενές) τύπο οντοτήτων A που εξαρτάται από τον ισχυρό τύπο οντοτήτων B δημιουργείται ένα σχήμα R με τα **εξής γνωρίσματα**:

- Τα γνωρίσματα του A
- Τα γνωρίσματα του πρωτεύον κλειδιού του B

Στην ουσία δημιουργείται ξένο κλειδί.

### Παράδειγμα:



**Σχόλιο!** Το ξένο κλειδί που δημιουργείται είναι το εξαρτώμενοι.

# Μετατροπή E-R σε σχεσιακό μοντέλο

## Συσχετίσεις:

Για μία συσχέτιση R μεταξύ n τύπων οντοτήτων που αντιστοιχούν στις σχέσεις K1...Kn δημιουργείται μία νέα σχέση R με τα εξής γνωρίσματα:

- Τα γνωρίσματα του πρωτεύοντος κλειδιού που συμμετέχουν σε κάθε σχέση K.
- Τα γνωρίσματα της συσχέτισης R (αν υπάρχουν).

## Παράδειγμα:



Σχόλιο! Παράδειγμα συσχέτισης εργαζονται.



# Μετατροπή E-R σε σχεσιακό μοντέλο

## Συσχετίσεις-ειδικές περιπτώσεις:

- Δυαδική (μη ασθενής) συσχέτιση πληθικότητας 1-1:  
Για κάθε δυαδική συσχέτιση R πληθικότητας 1-1 μεταξύ δύο τύπων οντοτήτων του E-R που αντιστοιχούν στις σχέσεις T και S υπάρχουν τα εξής γνωρίσματα:
  - Επιλέξτε μία από τις σχέσεις ή την T ή την S. Έστω της S.
  - Αφού επιλέξετε το πρωτεύον κλειδί της S αυτό γίνεται το ξένο κλειδί της T.

## Παράδειγμα:



## Μετατροπή E-R σε σχεσιακό μοντέλο

### Συσχετίσεις-ειδικές περιπτώσεις:

- Δυαδική συσχέτιση πληθικότητας 1-N:

Για κάθε δυαδική συσχέτιση R πληθικότητας 1-N μεταξύ δύο τύπων οντοτήτων του E-R που αντιστοιχούν στις σχέσεις T και S υπάρχουν τα εξής γνωρίσματα:

- Επιλέξτε μία από τις σχέσεις ή την T ή την S. Έστω της S από την μία πλευρά.
- Αφού επιλέξετε το πρωτεύον κλειδί της S αυτό γίνεται το ξένο κλειδί της T.

### Παράδειγμα:



## Μετατροπή E-R σε σχεσιακό μοντέλο

### Είδη γνωρισμάτων:

- Απλά: Ένα γνώρισμα τύπου οντοτήτων ή συσχετίσεων γίνεται γνώρισμα πίνακα στο σχεσιακό μοντέλο.
- Σύνθετα: Ένα γνώρισμα για κάθε απλό γνώρισμα που απαρτίζει το σύνθετο
- Πλειότιμα: Για κάθε γνώρισμα κατασκευάζεται μία σχέση R με τα εξής γνωρίσματα
  - Το γνώρισμα A (ή τα γνωρίσματα του A αν τον A είναι σύνθετο).
  - Τα γνωρίσματα (ξένο κλειδί) του πρωτεύοντος κλειδιού της σχέσης που παριστάνει τον τύπο οντοτήτων ή συσχετίσεων του οποίου γνώρισμα είναι το A.

# Μετατροπή E-R σε σχεσιακό μοντέλο

Άλλες περιπτώσεις:

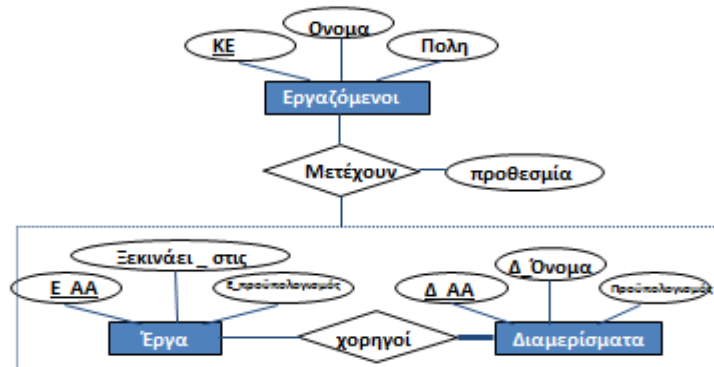
- Δεν είναι πλήρης ή έχει επικαλύψεις: Δεν υπάρχει δηλαδή οντότητα που να ανήκει σε κάποια γενίκευση αλλά ούτε ανήκει σε κάποια εξειδίκευση. Επίσης υπάρχει οντότητα που συμμετέχει ταυτόχρονα σε πάνω από δύο εξειδικεύσεις τις ίδιες γενίκευσης
- Πλήρης και χωρίς επικαλύψεις

Σε αυτήν την περίπτωση προκύπτουν N σχέσεις για κάθε υπό-τύπο οντοτήτων οι οποίες κληρονομούν και τα γνωρίσματα της γενίκευσης

Σε αυτήν την περίπτωση προκύπτουν N+1 σχέσεις για την γενίκευση και για κάθε υπό-τύπο οντοτήτων (με τα γνωρίσματα του και το πρωτεύον κλειδί της σχέσης το οποίο αντιπροσωπεύει το ξένο κλειδί

## Συσώρευση

Η συσώρευση επιτρέπει να χρησιμοποιηθεί ένας τύπος συσχετίσεων σαν τύπος οντοτήτων για λόγους συμμετοχής σε άλλες συσχετίσεις.



# Συσσώρευση

## Άλλες περιπτώσεις:

- **Συσσώρευση:** Είναι μία συσχέτιση μεταξύ ενός τύπου οντοτήτων και ενός τύπου συσχετίσεων όμως με διαφορετική αντιμετώπιση γιατί δεν υποστηρίζεται από το μοντέλο E-R

- Νέα σχέση (εάν η συσχέτιση είναι M:N)
- Προσκόλληση στη μία από τις δύο σχέσεις που αντιπροσωπεύουν τις δύο πλευρές της συσχέτισης (εάν η συσχέτιση είναι 1:1 ή N:1 ή 1:N)

# Επανάληψη

- Τύπος οντοτήτων ➡ Σχέση (οντοτήτων)
- Τύπος συσχέτισης 1:1 ή 1:N ➡ Ξένο κλειδί ή Σχέση (συσχέτισης)
- Τύπος συσχέτισης M:N ➡ Σχέση (συσχέτισης) με 2 (γενικά, n) ξένα κλειδιά
  
- Απλό γνώρισμα ➡ Γνώρισμα
- Σύνθετο γνώρισμα ➡ Σύνολο από γνωρίσματα
- Πλειότιμο γνώρισμα ➡ Σχέση και ξένο κλειδί



# Επανάληψη

---

Γενίκευση  $is\_A$  (πλήρης και χωρίς επικαλύψεις)  $\Rightarrow$   $N$  Σχέσεις που «κληρονομούν» και τα γνωρίσματα της γενίκευσης

Γενίκευση  $is\_A$  (αλλιώς)  $\Rightarrow$   $N+1$  Σχέσεις, οι  $N$  Σχέσεις με ξένο κλειδί

Συσώρευση  $\Rightarrow$  Ό,τι ισχύει και για τον Τύπο συσχέτισης

## Ενότητα 3

---

### Συναρτησιακές εξαρτήσεις και κανονικοποιήσεις

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Συναρτησιακές εξαρτήσεις
- Μεταβατική συναρτησιακή εξάρτηση
- Κλειδιά και μοναδικότητα
- Κανόνες συναγωγής- Αξιώματα του Armstrong
- Κανονικές μορφές 1KM,2KM,3KM
- BC-KM, 4KM, 5KM
- Άλλες KM
- Από-κανονικοποίηση



# Σκοποί ενότητας

Στο κεφάλαιο αυτό θα οριστούν οι συναρτησιακές εξαρτήσεις (τετριμμένες και μη) στις σχεσιακές βάσεις δεδομένων, καθώς και οι κανόνες συναγωγής ή αξιώματα του Armstrong. Η θεωρία της κανονικοποίησης αποτελεί σημαντική ενότητα για την δημιουργία συμπαγών και ακέραιων βάσεων για αυτό θα αναλυθούν όλες οι κανονικές μορφές που μπορούν να εφαρμοσθούν ξεκινώντας από την πρώτη κανονική μορφή και φτάνοντας μέχρι και την κανονική μορφή πεδίου ορισμού/κλειδιού.



## Συναρτησιακές εξαρτήσεις

Υποθετικά υπάρχει μία σχέση A και τα B και Δ είναι τυχαία υποσύνολα του συνόλου των γνωρισμάτων A, σε μία τέτοια περίπτωση το Δ είναι συναρτησιακά εξαρτημένο από το B ( $B \rightarrow \Delta$ ) όμως αυτό ισχύει εάν η κάθε τιμή του B αντιστοιχεί σε μία ακριβώς τιμή του Δ. Ποιά κατανοητά δύο συστοιχίες του συνόλου όταν συμφωνούν στην μία τιμή συμφωνούν και στην άλλη.

Πίνακας A

B		Δ		
Υποσύνολο 1	Υποσύνολο 1	Υποσύνολο 3	Υποσύνολο 4	Υποσύνολο 5



## Συναρτησιακές εξαρτήσεις

---

- Στις συναρτησιακές εξαρτήσεις εμπεριέχεται ο όρος εξάρτηση στον οποίο επίσης αποδίδονται δύο όροι **dependence (εξάρτηση)** και **dependency (εξαρτημένο αντικείμενο)**.
- οι συναρτησιακές εξαρτήσεις εμπεριέχουν ενδιαφέρουσες τυπικές ιδιότητες και αντιμετώπιση των προβλημάτων με τυπικό και αυστηρό τρόπο.
- Μια συναρτησιακή εξάρτηση είναι μία συσχέτιση πολλά προς πολλά.
- Στις συναρτησιακές σχέσεις ισχύει ότι εάν το B είναι υποψήφιο κλειδί μίας σχέσης A (ή το πρωτεύον κλειδί) τότε όλα τα γνωρίσματα της Δ πρέπει να είναι συναρτησιακά εξαρτημένα από το B. Επίσης εάν ισχύει η συναρτησιακή εξάρτηση ( $B \rightarrow \Delta$ ) και το B δεν είναι υποψήφιο κλειδί τότε η σχέση A έχει κάποιο πλεονασμό.



## Συναρτησιακές εξαρτήσεις

---

### Παράδειγμα:

Στη σχέση Φοιτητές το όνομα του φοιτητή και το ΑΕΜ του είναι συναρτησιακά εξαρτώμενο από το ΑΕΜ. Αυτό γίνεται γιατί Το ΑΕΜ προσδιορίζει μοναδικά τον φοιτητή δεν μπορούν να έχουν δύο φοιτητές το ίδιο ΑΕΜ. Για τον λόγο αυτό το ΑΕΜ ονομάζεται προσδιοριστικό (ή ορίζουσα).

Το σύμβολο  $\rightarrow$  σημαίνει προσδιορίζεται συναρτησιακά.

$AEM \rightarrow \text{Όνομα\_Φοιτητή}$  = Το ΑΕΜ προσδιορίζει μοναδικά το όνομα του φοιτητή.



# Συναρτησιακές εξαρτήσεις

---

Τετριμμένες και μη τετριμμένες εξαρτήσεις:

- **Τετριμμένη** λέγεται μία συναρτησιακή εξάρτηση εάν και μόνο εάν το δεξιό μέλος είναι υποσύνολο του αριστερού. Επίσης λέγεται τετριμμένη όταν δεν ικανοποιείται .
- **μη τετριμμένες** (nontrivial) εξαρτήσεις είναι "γνήσιες" δεσμεύσεις ακεραιότητας. Στην τυπική θεωρία των εξαρτήσεων, είναι αδύνατο να θεωρηθεί ότι οι εξαρτήσεις είναι μη τετριμμένες.



## Μεταβατική συναρτησιακή εξάρτηση

---

Οι μεταβατικές συναρτησιακές εξαρτήσεις έχουν τις εξής ιδιότητες:

- Έστω μια σχέση  $A$  που περιέχει κάποια γνωρίσματα  $(B, \Gamma, \Delta)$  τέτοια ώστε να ισχύουν οι παρακάτω συναρτησιακές εξαρτήσεις:  $B \rightarrow \Gamma, \Gamma \rightarrow \Delta$  και αφού ισχύουν αυτές να ισχύει και η  $B \rightarrow \Delta$ .
- Στο παραπάνω παράδειγμα η  $B \rightarrow \Delta$  είναι ένα παράδειγμα μεταβατικής συναρτησιακής εξάρτησης και σημαίνει ότι το  $\Delta$  εξαρτάται από το  $B$  μεταβατικά μέσω του  $\Gamma$ .



# Κλειδιά και μοναδικότητα

---

Το **κλειδί** είναι μία συστοιχία που προσδιορίζει μοναδικά ένα ή περισσότερα γνωρίσματα:

- Προσδιορίζει συναρτησιακά μια συνάρτηση.
- Δεν είναι όλα προσδιοριστικά κλειδιά.

## Κανόνες συναγωγής- Αξιώματα του Armstrong

---

Υποθετικά θεωρήστε ότι τα  $B$ ,  $\Gamma$ ,  $\Delta$  είναι τυχαία υποσύνολα ενός συνόλου με γνωρίσματα της σχέσης  $A$  και ότι το  $B\Gamma$  συμβολίζει την ένωση αυτών των δύο τότε:

- Το  $B$  είναι υποσύνολο του  $\Gamma$  ( $B \rightarrow \Gamma$ ). **Ανελαστικότητα.**
- Εάν  $B \rightarrow \Gamma$  τότε ισχύει  $B\Delta \rightarrow \Gamma\Delta$ . **Επαύξηση.**
- Εάν  $B \rightarrow \Gamma$  και  $\Gamma \rightarrow \Delta$  τότε  $B \rightarrow \Delta$ . **Μεταβατικότητα.**

## Κανόνες συναγωγής- Αξιώματα του Armstrong

---

Από τους κανόνες ανελαστικότητα, επαύξηση και μεταβατικότητα προκύπτουν και οι εξής:

- Αυτοκαθορισμός
- Ανάλυση
- Ένωση
- Σύνθεση



## Θεωρία Κανονικοποίησης

---

Μία σχέση είναι ένα σύνολο από γνωρίσματα με τιμές για κάθε γνώρισμα τέτοιες ώστε να ισχύουν οι παρακάτω ιδιότητες

1. Κάθε όνομα γνωρίσματος είναι μοναδικό.
2. Όλες οι τιμές κάθε γνωρίσματος είναι ίδιου τύπου (ή πεδίου ορισμού).
3. Κάθε τιμή γνωρίσματος είναι ατομική (μία τιμή και όχι ομάδα πολλών τιμών).
4. Τα γνωρίσματα δεν έχουν διάταξη από τα αριστερά προς τα δεξιά..
5. Οι συστοιχίες (σειρές) δεν έχουν διάταξη από επάνω προς τα κάτω.
6. Δεν υπάρχουν δύο ίδιες σειρές (συστοιχίες) σε μία σχέση.



## Θεωρία Κανονικοποίησης

---

- Η διαδικασία που ακολουθούμε είναι :
- 1. Συγκεντρώνουμε τις απαιτήσεις της επιχείρησης και των χρηστών.
- 2. Σχεδιάζουμε το μοντέλο οντοτήτων-συσχετίσεων
- 3. Μετατρέπουμε το διάγραμμα οντοτήτων-συσχετίσεων της επιχείρησης σε ένα σύνολο από σχέσεις (πίνακες) με το σχεσιακό μοντέλο.
- 4. Κανονικοποιούμε τις σχέσεις για να απομακρύνουμε τυχόν ανωμαλίες ενημέρωσης-διαγραφής-εισαγωγής στοιχείων.
- 5. Υλοποιούμε τη βάση δεδομένων δημιουργώντας ένα πίνακα για κάθε κανονικοποιημένη σχέση.

## Κανονικοποίηση-Κανονικές μορφές

---

- Κατηγορίες (ή κλάσεις) **Κανονικές Μορφές** (normal forms).
  - **Κανονική Μορφή**: Μία κλάση σχέσεων απαλλαγμένων από συγκεκριμένα προβλήματα τροποποιήσεων.
1. Πρώτη κανονική μορφή (1NF – 1KM)
  2. Δεύτερη κανονική μορφή (2NF – 2KM)
  3. Τρίτη κανονική μορφή (3NF – 3KM)
  4. Boyce-Codd κανονική μορφή (BCNF – KM BC)
  5. Τέταρτη κανονική μορφή (4NF – 4KM)
  6. Πέμπτη κανονική μορφή (5NF – 5KM)
  7. Κανονική μορφή πεδίου ορισμού κλειδιού (Domain-Key/NF)



## Κανονικοποίηση-Κανονικές μορφές

---

- Αυτές οι κανονικές μορφές είναι αθροιστικές. Μία σχέση που βρίσκεται σε Τρίτη κανονική μορφή είναι επίσης και σε δεύτερη και σε πρώτη.
- Οι τρεις πρώτες κανονικές μορφές (1KM, 2KM, 3KM) ορίστηκαν από τον Codd. Όλες οι κανονικοποιημένες σχέσεις είναι σε 1KM.
- Με άλλα λόγια, "κανονικοποιημένη" και "σε 1KM" σημαίνει ακριβώς το ίδιο πράγμα.
- Μερικές σχέσεις 1KM είναι επίσης σε 2KM, και μερικές σχέσεις 2KM είναι επίσης σε 3KM.

## Κανονικοποίηση-Κανονικές μορφές

---

- Ο Fagin όρισε την τέταρτη κανονική μορφή.
- Μετέπειτα και πάλι ο Fagin όρισε άλλη μία κανονική μορφή, την κανονική μορφή προβολής-σύζευξης (projection join), που αργότερα έγινε γνωστή και ως πέμπτη κανονική μορφή (5KM).
- Μερικές σχέσεις που είναι σε 4KM-BC είναι επίσης σε 4KM, και μερικές σχέσεις που είναι σε 4KM είναι επίσης σε 5KM.

## Πρώτη Κανονική Μορφή 1KM

---

- Μία σχέση βρίσκεται σε πρώτη κανονική μορφή αν ικανοποιεί όλες τις 6 ιδιότητες του ορισμού της σχέσης
- Εάν υπάρχει καθορισμένο **κλειδί** για τη σχέση τότε ικανοποιείται η απαίτηση της μοναδικότητας των συστοιχιών (σειρών).
- Ένας πίνακας σε πρώτη κανονική μορφή λέγεται κανονικοποιημένος πίνακας και τότε και μόνο τότε αντιστοιχεί σε μία σχέση (οι σχέσεις του σχεσιακού μοντέλου είναι στην 1η κανονική μορφή).

## Δεύτερη κανονική μορφή (2NF)

---

- Μία σχέση βρίσκεται σε δεύτερη κανονική μορφή εάν κάθε ένα από τα γνωρίσματά της που δεν είναι κλειδιά εξαρτώνται συναρτησιακά από ολόκληρο το πρωτεύων κλειδί και όχι μόνο από ένα τμήμα του.
- Οι σχέσεις που έχουν μόνο **ένα γνώρισμα σαν πρωτεύων κλειδί** βρίσκονται αυτόματα και στη **δεύτερη κανονική μορφή**.
- Αυτός είναι ένας λόγος για τον οποίο χρησιμοποιούμε συχνά τεχνητά αναγνωριστικά σαν κλειδιά.

## Τρίτη κανονική μορφή (3NF)

---

- Μία σχέση βρίσκεται σε Τρίτη κανονική μορφή εάν είναι σε δεύτερη και δεν περιέχει **μεταβατικές εξαρτήσεις**.
- Θεωρήστε για παράδειγμα τη σχέση R που έχει γνωρίσματα τα A, B και Γ. Εάν  $A \rightarrow B$  και  $B \rightarrow \Gamma$  τότε θα ισχύει και  $A \rightarrow \Gamma$ .

## Κανονική Μορφή Boyce/Codd

---

- Ο 3KM δεν αντιμετωπίζει περιπτώσεις:
  1. Έχει δύο ή περισσότερα κλειδιά
  2. Τα δύο υποψήφια κλειδιά να είναι σύνθετα
  3. Να επικαλύπτονται (να έχουν τουλάχιστον ένα γνώρισμα κοινό)
- κάθε σχέση στην **BCNF** είναι επίσης στην 3NF, αλλά δεν ισχύει πάντα το αντίστροφο.

## Κανονική Μορφή Boyce/Codd

---

- Μία σχέση βρίσκεται σε κανονική μορφή Boyce-Codd εάν κάθε προσδιοριστικό της σχέσης είναι ένα υποψήφιο κλειδί.
- ΚΜ-BC εάν και μόνο εάν τα μόνα ορίζοντα μέλη είναι υποψήφια κλειδιά

## Κανονική Μορφή Boyce/Codd

---

- Οποιαδήποτε σχέση που έχει μόνο δύο γνωρίσματα είναι σε BCNF.
- Η κανονική μορφή Boyce-Codd είναι η μεγαλύτερη που μπορούμε να φτάσουμε μέσω των συναρτησιακών εξαρτήσεων.

## Κανονική Μορφή Boyce/Codd

- Πχ.
- Χρηματιστηριακές Συναλλαγές γίνονται σε πολλούς Τύπους Μετοχών
- Οι Συναλλαγές διαχειρίζονται από έναν ή περισσότερους Χρηματιστές
- Οι Τύποι Μετοχών μπορούν έχουν έναν ή πολλούς Χρηματιστές
- Οι Χρηματιστές μπορούν να συναλλάσσονται σε έναν Τύπο Μετοχών

- ΚωδΣυναλλαγής, ΤύποςΜετοχής -> Χρηματιστής
- ΚωδΣυναλλαγής, Χρηματιστής -> ΤύποςΜετοχής
- Χρηματιστής -> ΤύποςΜετοχής

## Κανονική Μορφή Boyce/Codd

ΚωδΣυναλλαγής	ΤύποςΜετοχής	Χρηματιστής
15-03	Κοινές Μετοχές	Παπάς
15-04	Ομόλογα Δημοσίου	Γιώτης
3-103	Κοινές Μετοχές	Δημητρίου
2-234	Προνομιακές Μετοχές	Δήμου
56-117	Κοινές Μετοχές	Παπάς

## Κανονική Μορφή Boyce/Codd

---

Τι θα συνέβαινε εάν διαγραφόταν η εγγραφή με ΚωδΣυναλλαγής 2-234. Θα χανόταν και το γεγονός ότι ο Δήμος διαχειρίζεται τις Προνομιακές Μετοχές

### Βήματα για ΚΜ BC:

- ✓Λίστα με όλα τα γνώρισμα
- ✓Έλεγχος εάν κάθε γνώριμα μπορεί να είναι υποψήφιο κλειδί
- ✓Για τα γνώρισμα που δεν είναι κλειδιά δημιουργήστε μία σχέση για την κάθε συναρτησιακή εξάρτηση.
- ✓Συσχετίστε το γνώριμα με την αρχική σχέση

### Λύση

ΠίνακαςΑ(Χρηματιστής, Τύπος Μετοχής)

ΠίνακαςΒ(ΚωδΣυναλλαγής, Χρηματιστής)



## Κανονική Μορφή Boyce/Codd

---

- Επίπεδα κανονικοποίησης πλήρη διάταξη με την έννοια ότι σε κάθε σχέση που είναι σε  $n+1$  κανονική μορφή είναι αυτόματα και σε  $n$  ΚΜ ενώ το αντίστροφο δεν ισχύει
- Η αναγωγή σε ΚΜ-BC είναι πάντα δυνατή. Οποιαδήποτε δεδομένη σχέση μπορεί πάντα να αντικατασταθεί με ένα ισοδύναμο σύνολο σχέσεων σε ΚΜ-BC
- Σκοπός της αναγωγής είναι να αποφευχθεί ο πλεονασμός. Και ανωμαλίες ενημέρωσης



## Εξάρτηση Πολλαπλών Τιμών

---

- R σχέση, A,B,C, τυχαία υποσύνολα του συνόλου των γνωρισμάτων της R.
- Τότε το B είναι πολλαπλά εξαρτημένο με A ή  $B \twoheadrightarrow A$

ή αλλιώς το A καθορίζει πολλαπλά το B εάν σύνολο τιμών του B που αντιστοιχούν σε δεδομένο ζεύγος (A,C) στο R, εξαρτάται μόνο από την τιμή A και ανεξάρτητο από την τιμή-C

## Εξάρτηση Σύζευξης

---

- R σχέση, A,B,..Z, τυχαία υποσύνολα του συνόλου των γνωρισμάτων της R. τότε R ικανοποιεί εξάρτηση σύζευξης  $*(A,B,...Z)$  εάν και μόνο εάν η R είναι ίση με τη σύζευξη των προβολών της πάνω στα A,B...Z
- Η τέταρτη κανονική μορφή εξετάζει άλλου είδους εξαρτήσεις, τις εξαρτήσεις πολλαπλών τιμών. Επίσης η πέμπτη κανονική μορφή απομακρύνει εξαρτήσεις σύζευξης.

## ΤΕΤΑΡΤΗ Κανονική Μορφή

---

- Μία σχέση είναι σε 4KM εάν είναι σε BC-NF και δεν περιέχει εξαρτήσεις πολλαπλών τιμών (*multivalued dependencies*)
- 4KM εάν και μόνο εάν οι εξαρτήσεις πολλαπλών τιμών που ικανοποιεί είναι στην πραγματικότητα συναρτησιακές εξαρτήσεις που ξεκινούν από υποψήφια κλειδιά



## ΤΕΤΑΡΤΗ Κανονική Μορφή

---

- Ο εργαζόμενος συμμετέχει σε πολλά έργα.  
Ο εργαζόμενος επιβλέπει περισσότερους του ενός εργαζομένους.

ΠΡΟΪΣΤΑΜΕΝΟΣ	ΕΡΓΟ	ΕΠΙΒΛ-ΕΡΓΑΖΟΜ
ΠΑΠΑΣ	A	ΓΙΑΝΝΗΣ
ΠΑΠΑΣ	B	ΣΟΦΙΑ
ΠΑΠΑΣ	A	ΣΟΦΙΑ
ΠΑΠΑΣ	B	ΓΙΑΝΝΗΣ
ΔΗΜΟΥ	A	ΚΩΣΤΑΣ





## ΤΕΤΑΡΤΗ Κανονική Μορφή

### •ΛΥΣΗ

ΠΡΟΪΣΤΑΜΕΝΟΣ	ΕΡΓΟ
ΠΑΠΑΣ	A
ΠΑΠΑΣ	B
ΔΗΜΟΥ	Δ
ΔΗΜΟΥ	A
ΔΗΜΟΥ	B

ΠΡΟΪΣΤΑΜΕΝΟΣ	ΕΞΑΡΤΩΜΕΝΟΣ
ΠΑΠΑΣ	ΓΙΑΝΝΗΣ
ΠΑΠΑΣ	ΣΟΦΙΑ
ΔΗΜΟΥ	ΚΩΣΤΑΣ
ΔΗΜΟΥ	ΝΙΚΗ
ΔΗΜΟΥ	ΣΤΕΛΛΑ

## Πέμπτη Κανονική Μορφή

- 5KM εάν και μόνο εάν οι μόνες εξαρτήσεις σύζευξης που ικανοποιεί είναι στην πραγματικότητα συναρτησιακές εξαρτήσεις που ξεκινούν από υποψήφια κλειδιά

## Άλλες ΚΜ

---

### ΚΜ ΠΟΚ(ΠΕΔΙΟΥ ΟΡΙΣΜΟΥ-ΚΛΕΙΔΙΩΝ)

- Μια δέσμευση που λέει ότι οι τιμές ενός δεδομένου γνωρίσματος παίρνονται από ένα καθορισμένο πεδίο ορισμού
- Μια δέσμευση που ορίζει ότι κάποιο γνώρισμα ή συνδυασμός γνωρισμάτων είναι υποψήφιο κλειδί

### ΚΜ ΠΕΡΙΟΡΙΣΜΟΥ- ΕΝΩΣΗΣ

Δίνει απαντήσεις σε ερωτήσεις που η θεωρία της κανονικοποίησης δεν δίνει απαντήσεις. Κακοί σχεδιασμοί.



## Απο-κανονικοποίηση

---

- Υπάρχουν περιπτώσεις όπου μπορεί γίνει να από-κανονικοποίηση των σχέσεων ώστε να επιτευχθεί καλύτερη απόδοση της βάσης( De-Normalization)



## Ενότητα 4

---

### Οι όψεις στην SQL

Δρ. Τιμπίρης Αλκιβιάδης



---

## Περιεχόμενα ενότητας

---

- Βασική Δομή
- Συμβολοσειρές – διάταξη
- Πράξεις συνόλων
- Συναθροίστηκες Συναρτήσεις
- Φωλιασμένες Υπο-ερωτήσεις
- Συνενώσεις συνόλων
- Παραγόμενες Σχέσεις
- Κανόνας ερωτημάτων 80-20
- Όψεις (Views)



## Σκοποί ενότητας

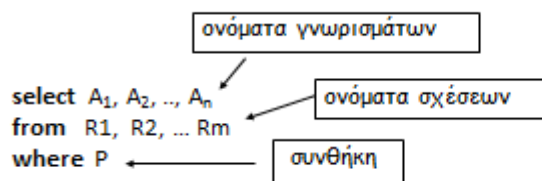
---

Στο κεφάλαιο θα γίνει αναφορά στην γλώσσα SQL και συγκεκριμένα στα εμφωλευμένα ερωτήματα, περιορισμούς ακεραιότητας, στη δημιουργία όψεων, στις συνενώσεις συνόλων και τέλος στις παραγόμενες σχέσεις.

## Βασική Δομή

---

Μια χαρακτηριστική ερώτηση σε SQL έχει την εξής μορφή:



Ισοδύναμο του:  $\pi_{A_1, A_2, \dots, A_n} (\sigma_P (R_1 \times R_2 \times \dots \times R_m))$

# Βασική Δομή

---

## Select

Αριθμητικές πράξεις (+, -, \*, /) ανάμεσα σε σταθερές ή γνωρίσματα πλειάδων

Διαγραφή διπλότιμων: **select distinct**

## Συνθήκη του where

Λογικοί τελεστές: **and, or, not**

Τελεστές σύγκρισης: <, <=, >, >=, =, <>, **between, not between**

ανάμεσα σε αριθμητικές εκφράσεις, συμβολοσειρές (strings), και ειδικούς τύπους.

# Βασική Δομή

---

- Όταν το ίδιο γνώρισμα εμφανίζεται στο σχήμα περισσότερων από μια σχέσεων, τότε γίνεται διάκριση βάση του συμβολισμού:

<όνομα-σχέσης>.<όνομα-γνωρίματος>

- Δυνατότητα **αλλαγής του ονόματος** τόσο μιας σχέσης όσο και ενός γνωρίματος:

<παλιό-όνομα> **as** <νέο-όνομα>

Το **as** μπορεί να εμφανίζεται στο **select** ή στο **from**.

Οι **μεταβλητές πλειάδων** είναι ιδιαίτερα χρήσιμες όταν θέλουμε να συγκρίνουμε δυο πλειάδες τις ίδιας σχέσης.

# Συμβολοσειρές - διάταξη

---

## Πράξεις με Συμβολοσειρές

Η πιο συνηθισμένη πράξη είναι ταίριασμα προτύπων:

- % (\* στην access) ταιριάζει οποιαδήποτε συμβολοσειρά
- \_ ταιριάζει οποιοδήποτε χαρακτήρα
- Σύγκριση χρησιμοποιώντας το **like**, **not like**

## Διάταξη των Πλειάδων

- Χρήση του **order by** ώστε οι πλειάδες στο αποτέλεσμα να είναι ταξινομημένες με βάση το αντίστοιχο γνώρισμα.
- Default: αύξουσα διάταξη, αλλά και άμεσα χρησιμοποιώντας το **asc** (αύξουσα) ή το **desc** (φθίνουσα).



# Πράξεις Συνόλων

---

## **Πράξεις:**

- **union**
- **intersection**
- **except**

## Σύνταξη:

(select-from-where) **union** (select-from-where)

Απαλοιφή διπλών εμφανίσεων, εκτός αν χρησιμοποιηθεί το **union all**.

## Τιμή null

Χρήση της λέξης κλειδί **is null** (**is not null**) σε μια συνθήκη για να ελέξουμε αν μια τιμή είναι null.



## Συναθροίστερες Συναρτήρες

Μέρος όρος: **avg(A)** (μόνο σε αριθμούς)

Ελάχιστο: **min(A)**

Μέγιστο: **max(A)**

Άθροισμα: **sum(A)** (μόνο σε αριθμούς)

Πλήθος: **count(A)**, όπου **A** γνώρισμα.

Αν θέλουμε να απαλείψουμε  
διπλές εμφανίσεις  
χρησιμοποιούμε τη λέξη-κλειδί  
**distinct** στην αντίστοιχη έκφραση

**group by**: για να εφαρμόσουμε τις συναρτήρες όχι μόνο σε ένα σύνολο από πλειάδες, αλλά σε ομάδες από σύνολα πλειάδων.

**having**: για να εφαρμόσουμε μια συνθήκη σε μια συγκεκριμένη ομάδα από πλειάδες. Η συνθήκη του **having** εφαρμόζεται αφού σχηματιστούν οι ομάδες και υπολογιστούν οι συναθροιστικές συναρτήρες.



## Φωλιασμένες Υπο-ερωτήσεις

Γενική δομή:

**select ...**

**from ...**

**where**

? (select ...

from ...

where ... );

υποερώτηση

Υπολογισμός της υποερώτησης για κάθε γραμμή (πλειάδα) της εξωτερικής ερώτησης.



## Φωλιασμένες Υπο-ερωτήσεις

---

- in/not in (συμμετοχή σε σύνολο)
- (>, =, κλπ) some/any/all (σύγκριση συνόλων)
- exists/not exists (έλεγχος για κενά σύνολα)
- unique/not unique (έλεγχος για διπλότιμα)

## Συενώσεις Συνόλων

---

Η SQL υποστηρίζει διάφορους τύπους συενώσεων που συνήθως χρησιμοποιούνται στο **from**, αλλά μπορούν να χρησιμοποιηθούν οπουδήποτε μπορεί να χρησιμοποιηθεί μια σχέση.

### Γενική σύνταξη:

<όνομα-σχέσης1><τύπος-συνένωσης><όνομα-σχέσης2><συνθήκη-συνένωσης>

ή

<όνομα-σχέσης1> **natural** <τύπος-συνένωσης> <όνομα-σχέσης2>



# ΣΥΝΕΝΩΣΕΙΣ ΣΥΝΟΛΩΝ

---

## Τύποι Συνένωσης:

**inner join:** εσωτερική (θήτα) συνένωση

**left outer join:** αριστερή εξωτερική συνένωση

**right outer join :** δεξιά εξωτερική συνένωση

**full outer join :** ολική εξωτερική συνένωση



# ΣΥΝΕΝΩΣΕΙΣ ΣΥΝΟΛΩΝ

---

## Συνθήκες Συνένωσης:

on P

**using** ( $A_1, A_2, \dots, A_n$ ): γνωρίσματα που πρέπει να ταιριάζουν στη συνένωση είναι τα  $A_i$ . Είναι γνωρίσματα κοινά και στις δυο σχέσεις και εμφανίζονται στο αποτέλεσμα μόνο μια φορά.

- Για την εσωτερική συνένωση η συνθήκη δεν είναι υποχρεωτική, όταν λείπει ισοδυναμεί με καρτεσιανό γινόμενο.



## Συνενώσεις Συνόλων

---

Οι λέξεις κλειδιά **inner** και **outer** είναι προαιρετικές.

**natural**: φυσική συνένωση, τα γνωρίσματα εμφανίζονται στο αποτέλεσμα με την εξής διάταξη: πρώτα αυτά με τα οποία έγινε η συνένωση (δηλ., αυτά που είναι κοινά και στις δύο σχέσεις), μετά τα υπόλοιπα της πρώτης σχέσης, και τέλος τα υπόλοιπα της δεύτερης σχέσης.

*Παράδειγμα: Τα ονόματα των πελατών που είτε έχουν καταθέσεις είτε έχουν πάρει δάνεια (αλλά όχι και τα δυο).*

```
select ONOMA_ΠΕΛΑΤΗ
from ΚΑΤΑΘΕΤΗΣ inner join ΔΑΝΕΙΖΟΜΕΝΟΣ
where AP_ΛΟΓΑΡΙΑΣΜΟΥ is null or AP_ΔΑΝΕΙΟΥ is null
```

## Παραγόμενες Σχέσεις

---

- Η SQL δίνει τη δυνατότητα μια υποερώτηση να χρησιμοποιηθεί στο **from**.
- Τότε πρέπει να τις δοθεί ένα όνομα και τα γνωρίσματα της να μετανομαστούν
- Αυτό γίνεται χρησιμοποιώντας το **as**
- Η SQL δίνει τη δυνατότητα χρησιμοποιώντας το **as** να δοθεί ένα προσωρινό όνομα σε μία προσωρινή σχέση που προκύπτει από μια υποερώτηση.

## Παραγόμενες Σχέσεις

---

*Παράδειγμα: Το μέσο υπόλοιπο για όλα τα υποκαταστήματα για τα οποία το μέσο ποσό είναι μεγαλύτερο των \$1200.*

```
select ΟΝΟΜΑ_ΥΠΟΚΑΤΑΣΤΗΜΑΤΟΣ, ΜΕΣΟ_ΥΠΟΛΟΙΠΟ
from (select ΟΝΟΜΑ_ΥΠΟΚΑΤΑΣΤΗΜΑΤΟΣ, avg(ΠΟΣΟ)
from ΚΑΤΑΘΕΤΗΣ
group by ΟΝΟΜΑ_ΥΠΟΚΑΤΑΣΤΗΜΑΤΟΣ)
as ΑΠΟΤΕΛΕΣΜΑ (ΟΝΟΜΑ_ΥΠΟΚΑΤΑΣΤΗΜΑΤΟΣ, ΜΕΣΟ_ΥΠΟΛΟΙΠΟ)
where ΜΕΣΟ_ΥΠΟΛΟΙΠΟ > 1200
```



## Κανόνας ερωτημάτων 80-20

---

- Σε ένα σύστημα στο οποίο χρησιμοποιείται βάση δεδομένων (δηλαδή σε όλα τα συστήματα) εκτελούνται πολλά ερωτήματα
- Έχει παρατηρηθεί ένας εμπειρικός κανόνας ότι ορισμένα ερωτήματα εκτελούνται πιο συχνά

### Παράδειγμα:

- `Select * from customers where id=X`

Όπου X ένας ακέραιος αριθμός.

- Ο κανόνας ορίζει ότι το 20% των ερωτημάτων εκτελούνται το 80% των φορών.
- Και το 80% των ερωτημάτων συνολικά καταλαμβάνουν το υπόλοιπο 20%.



## Κανόνας ερωτημάτων 80-20

---

Συνεπώς ορισμένα ερωτήματα είναι πιο σημαντικά από τα υπόλοιπα, υπό την έννοια ότι εκτελούνται συχνότερα.

Αυτό σημαίνει:

- Γρηγορότερη εκτέλεση.
- Απλοποίηση. Αντί δηλαδή να χρειάζεται να γράψουμε κάθε φορά πολλούς χαρακτήρες.
- Επαναχρησιμοποίηση.

## Όψεις (Views)

---

Μπορούμε να ορίσουμε μια όψη χρησιμοποιώντας την εντολή:

```
create view <όνομα--όψης> as <select-from-where ερώτηση>
```

- Το αποτέλεσμα ενός ερωτήματος SELECT έχει τη μορφή πίνακα.
- Μπορούμε ένα ερώτημα να το ονοματίσουμε και έπειτα να το μεταχειριστούμε ως πίνακα.

Επίσης, μπορούν να προσδιοριστούν τα ονόματα των γνωρισμάτων άμεσα

```
create view <όνομα--όψης> (<λίστα ονομάτων-γνωρισμάτων>)  
as <select-from-where ερώτηση>
```

## Όψεις (Views)

---

- Κάθε φορά που εκτελούμε ένα ερώτημα σε μία όψη, αυτή επαναυπολογίζεται.
- Δηλαδή, τρέχει το ερώτημα το οποίο τη δημιουργεί.
- Ανάλογα με τα πεδία τα οποία περιέχουν και τους πίνακες, μπορεί να επιτρέπονται και λειτουργίες ενημέρωσης, εισαγωγής και διαγραφής.
- Τέτοιες όψεις ονομάζονται **updatable views** έναντι των **Read-only vs.**
- Η βάση δεδομένων αποφασίζει σε ποια κατηγορία μία όψη ανήκει.

## Όψεις (Views)

---

### Παράδειγμα :

*Μια όψη που περιλαμβάνει τα ονόματα όλων των υποκαταστημάτων και το άθροισμα του ποσού των δανείων που έχουν γίνει από αυτά*

```
create view ΥΠΟΚΑΤΑΣΤΗΜΑ_ΣΥΝΟΛΟ_ΔΑΝΕΙΩΝ(ΣΥΝΟΛΟ_ΔΑΝΕΙΩΝ,  
ΟΝΟΜΑ_ΥΠΟΚΑΤΑΣΤΗΜΑ) as  
select ΟΝΟΜΑ_ΥΠΟΚΑΤΑΣΤΗΜΑ, sum(ΠΟΣΟ)  
from ΔΑΝΕΙΑ  
group by ΟΝΟΜΑ_ΥΠΟΚΑΤΑΣΤΗΜΑ
```

# Ώψεις (Views)

---

- Τα ονόματα όψεων μπορεί να χρησιμοποιηθούν οπουδήποτε μπορεί να χρησιμοποιηθεί το όνομα μιας σχέσης.
- Ο ορισμός της όψης παραμένει στην βάση δεδομένων, εκτός αν σβηστεί:

**drop view** <όνομα-όψης>

## Ενότητα 5

---

### Επεκτάσεις SQL

Δρ. Τιμπίρης Αλκιβιάδης



## Περιεχόμενα ενότητας

---

- Επεκτάσεις της γλώσσας SQL
- Μπλοκ εντολών
- Εντολές IF, WHILE, FOR
- Δηλώσεις μεταβλητών
- Αποθηκευμένες διαδικασίες (Σύνταξη- Εκτέλεση)
- Συναρτήσεις (Σύνταξη- Εκτέλεση)
- Σκανδάλες (Σύνταξη- Εκτέλεση)



## Σκοποί ενότητας

---

Η ενότητα αυτή αποτελεί ίσως την κυριότερη ενότητα αυτού του συνόλου διαλέξεων. Θα γίνει αναφορά στις επεκτάσεις της γλώσσας SQL και συγκεκριμένα στις εντολές προγραμματισμού που της δίνουν ευελιξία στο αναπτύσσονται σύνθετα προγράμματα. Οι εντολές IF, WHILE, FOR μέσα με μπλοκ εντολών και με συνδυασμό τοπικών και γενικευμένων μεταβλητών μπορούν να χρησιμοποιηθούν σε Αποθηκευμένες διαδικασίες, σε συναρτήσεις οριζόμενες από τον χρήστη αλλά και σε σκανδάλες που ενεργοποιούνται με οποιαδήποτε αλλαγή γίνει σε στοχευόμενους πίνακες.



## Επεκτάσεις SQL

---

Πρόσθετες προτάσεις Transact-SQL, που αναφέρονται σαν "επεκτάσεις SQL"

- Αποθηκευμένες διαδικασίες ή σκριπτ που αποθηκεύονται στον διακομιστή και μπορούν να ξαναχρησιμοποιηθούν
- Μια δέσμη είναι μια αλληλουχία προτάσεων SQL και επεκτάσεων SQL που στέλνονται στο SQL Server για να εκτελεστούν μαζί.
- Οι προτάσεις ορισμού δεδομένων CREATE VIEW, CREATE PROCEDURE, CREATE RULE, CREATE TRIGGER και CREATE DEFAULT πρέπει να είναι η καθεμία η μόνη πρόταση μέσα σε μια δέσμη.



## Μπλοκ Προτάσεων

---

Ενα μπλοκ επιτρέπει την δημιουργία μονάδων με μια ή περισσότερες προτάσεις SQL. Κάθε μπλοκ αρχίζει με την πρόταση BEGIN και τελειώνει με την πρόταση END:

```
BEGIN
    Πρόταση 1
    πρόταση_2
END
```

## Πρόταση IF

---

Ενα μπλοκ μπορεί να χρησιμοποιηθεί μέσα στην πρόταση IF για να επιτρέψει την εκτέλεση περισσότερων της μιας προτάσεων, ανάλογα με μια ορισμένη συνθήκη

```
IF Συνθήκη Αληθής
    BEGIN
        πρόταση 1
        πρόταση_2
    END
ELSE BEGIN
        πρόταση 3
        πρόταση_4
    END
```

## Πρόταση IF

---

```
IF (SELECT COUNT(*)
     FROM ΒΑΘΜΟΛΟΓΙΑ
     WHERE ΚΜ = '403' AND ΒΑΘΜΟΣ >= 5
     GROUP BY ΑΜ ) > 100
    PRINT 'Ο αριθμός των σπουδαστών που πέρασαν το
μάθημα 403 είναι πάνω από 100'
ELSE BEGIN
    PRINT 'Οι σπουδαστές που πέρασαν στο μάθημα 403
είναι:'
        SELECT ΣΠΟΥΔΑΣΤΕΣ.ΑΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ
        FROM ΣΠΟΥΔΑΣΤΕΣ, ΒΑΘΜΟΛΟΓΙΑ
        WHERE ΣΠΟΥΔΑΣΤΕΣ.ΑΜ = ΒΑΘΜΟΛΟΓΙΑ.ΑΜ AND
ΚΜ='403'
    END
```

Η πρόταση PRINT είναι μια άλλη επέκταση SQL. Επιστρέφει ένα μήνυμα που ορίζεται από τον χρήστη

## Πρόταση WHILE

---

Η πρόταση WHILE εκτελεί επαναληπτικά προτάσεις SQL που περικλείονται μέσα σε ένα μπλοκ, όσο η συνθήκη της while επιστρέφει τιμή true.

Ένα μπλοκ μέσα στην πρόταση WHILE μπορεί να περιέχει

BREAK για να σταματά την εκτέλεση των προτάσεων μέσα στο μπλοκ και αρχίζει την εκτέλεση της πρότασης που ακολουθεί αυτό το μπλοκ.

Η CONTINUE για να σταματά μόνο την τρέχουσα εκτέλεση των προτάσεων μέσα στο μπλοκ και αρχίζει την εκτέλεση του μπλοκ από την αρχή.

## Πρόταση WHILE

---

```
WHILE (SELECT SUM(ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ)
      FROM ΕΡΓΟ) < 100.000
BEGIN
    UPDATE ΕΡΓΟ SET ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ = ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ
*1.3
    IF (SELECT ΠΡΟΫΠΟΛΟΓΙΣΜΟΣ)
      FROM ΕΡΓΟ) > 50.000 BREAK
    ELSE CONTINUE
END
```

## Τοπικές Μεταβλητές

---

Επέκταση στην γλώσσα Transact-SQL.

- Αποθηκεύουν τιμές
- Είναι "τοπικές" επειδή χρησιμοποιούνται μόνο μέσα στην ίδια δέσμη, όπου δηλώνονται.
- Το SQL Server υποστηρίζει επίσης καθολικές μεταβλητές(@@)
- Τοπική μεταβλητή ορίζεται με **DECLARE**.
- Ο ορισμός περιέχει το όνομα και τον αντίστοιχο τύπο δεδομένων
- Οι μεταβλητές χρησιμοποιούν το πρόθεμα @
- Η εκχώρηση μιας τιμής σε μια τοπική μεταβλητή γίνεται χρησιμοποιώντας τα παρακάτω:
  1. Με τη **SELECT**
  2. Με την πρόταση **SET**

## Τοπικές Μεταβλητές- ΠΑΡΑΔΕΙΓΜΑ

---

```
DECLARE @ΜΕΣΗ_ΤΙΜΗ_ΑΓ float, @ΑΥΞ_ΤΙΜΗΣ float
SET @ΑΥΞ_ΤΙΜΗΣ = 10
SELECT @ΜΕΣΗ_ΤΙΜΗ_ΑΓ = AVG(ΤΙΜΗ_ΑΓ) FROM ΑΠΟΘΗΚΗ
IF (SELECT ΤΙΜΗ_ΑΓ FROM ΑΠΟΘΗΚΗ
    WHERE ΚΩΔ_ΕΙΔΟΥΣ='ΑΑ05') < @ΜΕΣΗ_ΤΙΜΗ_ΑΓ
BEGIN
    UPDATE ΑΠΟΘΗΚΗ
    SET ΤΙΜΗ_ΑΓ = ΤΙΜΗ_ΑΓ + @ΑΥΞ_ΤΙΜΗΣ
    WHERE ΚΩΔ_ΕΙΔΟΥΣ = 'ΑΑ05'
    PRINT 'Η ΤΙΜΗ ΑΓΟΡΑΣ ΑΥΞΗΘΗΚΕ ΚΑΤΑ
    @ΑΥΞ_ΤΙΜΗΣ'
END
ELSE PRINT 'Η ΤΙΜΗ ΑΓΟΡΑΣ ΔΕΝ ΑΛΑΞΕ'
```

## Προτάσεις Διαδικασιών

---

- **RETURN**
- **GOTO**
- **RAISEERROR**
- **WAITFOR**

## RETURN , GOTO

---

**RETURN** Έχει την ίδια λειτουργικότητα μέσα σε μια δέσμη εντολών με την πρόταση BREAK μέσα στην WHILE.

Αυτό σημαίνει ότι η RETURN κάνει την εκτέλεση της δέσμης να τερματίσει και να αρχίσει να εκτελείται η πρώτη πρόταση που ακολουθεί την δέσμη.

**GOTO:** Διακλαδώνει σε μια ετικέτα, που βρίσκεται μπροστά από μια πρόταση Transact-SQL μέσα σε μια δέσμη

## RAISEERROR

---

Παράγει ένα σφάλμα που ορίζεται από τον χρήστη και θέτει μια ένδειξη σφάλματος συστήματος.

Ενα μήνυμα σφάλματος που ορίζεται από τον χρήστη πρέπει να είναι μεγαλύτερο από 50000.

Όλοι οι αριθμοί σφάλματος που είναι  $\leq 50000$  είναι αριθμοί που ορίζονται από το σύστημα και δεσμεύονται από το SQL Server.

Οι τιμές σφάλματος αποθηκεύονται στην καθολική μεταβλητή @@ERROR

## WAITFOR

---

Ορίζει το χρονικό διάστημα που πρέπει να περιμένει το σύστημα, πριν να εκτελέσει την επόμενη πρόταση στην δέσμη.

Η σύνταξη αυτής της πρότασης είναι

**WAITFOR {DELAY 'χρόνος' | TIME 'ώρα'}**

Η **DELAY** λέει στο SQL Server να περιμένει μέχρι να περάσει ο καθορισμένος χρόνος.

Η **TIME** καθορίζει μια ώρα σε μια αποδεκτή μορφή δεδομένων ημερομηνίας/ώρας

---

---

## Αποθηκευμένες Διαδικασίες

## Αποθηκευμένες Διαδικασίες

---

**Τι είναι οι Αποθηκευμένες Διαδικασίες?**

Είναι ένα ειδικό είδος δέσμης γραμμένη σε Transact-SQL, χρησιμοποιώντας την γλώσσα SQL και επεκτάσεις SQL.

Αποθηκεύεται στον διακομιστή της βάσης δεδομένων για να βελτιώσει την απόδοση και την συνέπεια επαναληπτικών εργασιών

Το SQL Server υποστηρίζει αποθηκευμένες διαδικασίες

Δημιουργούνται χρησιμοποιώντας την γλώσσα ορισμού δεδομένων

## Αποθηκευμένες Διαδικασίες

---

Η διαδικασία δέχεται τα αντίστοιχα ορίσματα κάθε φορά που καλείται.

Οι αποθηκευμένες διαδικασίες μπορούν προαιρετικά να επιστρέψουν μια τιμή, που εμφανίζει τις πληροφορίες που ορίζονται από τον χρήστη

ή, στην περίπτωση ενός σφάλματος, το αντίστοιχο μήνυμα σφάλματος

## Αποθηκευμένες Διαδικασίες

---

Μια αποθηκευμένη διαδικασία μεταγλωττίζεται εκ των προτέρων, πριν να αποθηκευτεί σαν αντικείμενο μέσα στην βάση δεδομένων

Πλεονέκτημα: η επαναλαμβανόμενη μεταγλώττιση μιας διαδικασίας σχεδόν πάντα εξαλείφεται και η απόδοση της εκτέλεσης αυξάνεται

Πλεονέκτημα που αφορά τον όγκο των δεδομένων :  
χρειάζονται λιγότερα από 50 bytes για να κληθεί μια αποθηκευμένη διαδικασία που περιέχει αρκετές χιλιάδες bytes προτάσεων

## Αποθηκευμένες Διαδικασίες

---

Γιατί αποθηκευμένες διαδικασίες?

Για έλεγχο της εξουσιοδότησης πρόσβασης

Για δημιουργία μιας διαδρομής ελέγχου δραστηριοτήτων σε πίνακες βάσης δεδομένων

Για διαχωρισμό προτάσεων ορισμού δεδομένων και χειρισμού δεδομένων, που αφορούν μια βάση δεδομένων και όλες τις αντίστοιχες εφαρμογές



# Αποθηκευμένες Διαδικασίες

---

## Δημιουργία και Εκτέλεση Αποθηκευμένων Διαδικασιών

### Σύνταξη

```
CREATE PROC [EDURE] [ιδιοκτήτης.]όνομα_διαδικασίας  
    [;αριθμός]  
    [( {@παράμετρος1 } τύπος1 [VARYING] [=προεπιλογή1]  
    [OUTPUT]])]  
    [( {@παράμετρος2 } τύπος2 [VARYING] [=προεπιλογή2]  
    [OUTPUT]])]...  
    [WITH {RECOMPILE | ENCRYPTION | RECOMPILE,  
    ENCRYPTION}] [FOR REPLICATION]  
AS Δέση ΕΝΤΟΛΩΝ
```

# Αποθηκευμένες Διαδικασίες

---

## Δημιουργία Αποθηκευμένων Διαδικασιών

### Παράδειγμα1

```
CREATE PROCEDURE ΑΥΞΗΣΗ_ΛΙΑΝ_ΤΙΜΗΣ  
    (@ΠΟΣΟΣΤΟ FLOAT=5)  
AS UPDATE ΑΠΟΘΗΚΗ  
    SET ΤΙΜΗ_ΛΙΑΝ=ΤΙΜΗ_ΛΙΑΝ*(1+@ΠΟΣΟΣΤΟ/100)
```

### Παράδειγμα2

```
CREATE PROCEDURE ΑΥΞΗΣΗ_bathmon ( @ΠΟΣΟ FLOAT=1)  
AS UPDATE [ΒΑΘΜΟΙ ΣΠΟΥΔΑΣΤΩΝ]  
    SET ΒΑΘΜΟΣ=ΒΑΘΜΟΣ+@ΠΟΣΟ
```

# Αποθηκευμένες Διαδικασίες

---

## Εκτέλεση Αποθηκευμένων Διαδικασιών

### ΣΥΝΤΑΞΗ

```
EXEC[UTE] [@επιστρ_κατάσταση =] όνομα διαδικασίας  
                                     [;αριθμός]  
{[[@παράμετρος1 =] τιμή |  
  @παράμετρος1 = ]@μεταβλητή [ή [OUTPUT]]}..  
[WITH RECOMPILE]
```

# Αποθηκευμένες Διαδικασίες

---

## Εκτέλεση Αποθηκευμένων Διαδικασιών

### Παράδειγμα 1

```
EXECUTE ΑΥΞΗΣΗ_ΛΙΑΝ_ΤΙΜΗΣ 15
```

```
EXECUTE ΑΥΞΗΣΗ_bathmon 3
```

## Αποθηκευμένες Διαδικασίες

---

### Εκτέλεση Αποθηκευμένων Διαδικασιών

`EXECUTE Αποθ_Διαδικασία Παράμετρος_Εισόδου`

Η εντολή αυτή εκτελείται είτε μέσα από μια γλώσσα προγραμματισμού προγραμματίζοντας ο `TEXT` ενός `dataSet`

ή

από την επιλογή `Query` του `SQL Server`

---

## Συναρτήσεις που Ορίζονται από τον Χρήστη

### User Defined Functions

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

Στις γλώσσες προγραμματισμού υπάρχουν γενικά δύο τύποι ρουτινών: **Διαδικασίες & Συναρτήσεις**

Οι διαδικασίες αποτελούνται από αρκετές προτάσεις που έχουν μηδέν ή περισσότερες παραμέτρους εισόδου, αλλά δεν επιστρέφουν παραμέτρους εξόδου. Σε αντίθεση με αυτό, οι συναρτήσεις γενικά επιστρέφουν μια ή περισσότερες παραμέτρους.

Σημείωση Όπως θα δείτε παρακάτω, οι συναρτήσεις SQL Server δεν υποστηρίζουν παραμέτρους εξόδου, αλλά επιστρέφουν μια μόνο τιμή δεδομένων.

Τμήμα Μηχανικών Πληροφορικής ΤΕ

27

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

### Δημιουργία και Εκτέλεση των Συναρτήσεων

```
CREATE FUNCTION [ owner_name. ] function_name
    ( [ { @parameter_name [AS] scalar_parameter_data_type [= default ] } [ , ...n ] ]
    )
RETURNS @return_variable TABLE < table_type_definition >
[ WITH < function_option > [ [ , ...n ] ]
[ AS ]
BEGIN
    function_body
    RETURN
END
< function_option > ::=
    { ENCRYPTION | SCHEMABINDING }
< table_type_definition > ::=
    ( { column_definition | table_constraint } [ , ...n ] )
```

Τμήμα Μηχανικών Πληροφορικής ΤΕ

28

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

```
CREATE FUNCTION [ιδιοκτήτης.] όνομα συνάρτησης  
[[{@παράμετρος1 } τύπος1 [= προεπιλογή1]]  
  (({@παράμετρος2 } τύπος2 [= προεπιλογή2] )}]...  
  
RETURNS [βαθμωτόςτύπος | ©μεταβλητή] TABLE}  
  
[WITH {ENCRYPTION | SCHEMABINDING} [AS] {μπλοκ |  
  RETURN (πρόταση)}]
```

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

**Ιδιοκτήτης** είναι το όνομα του χρήστη στον οποίο έχει εκχωρηθεί η ιδιοκτησία της συνάρτησης που έχει οριστεί από τον χρήστη.

**Όνομα\_συνάρτησης** είναι το όνομα της νέας συνάρτησης.

**@παράμετρος1, @παράμετρος2,...** παράμετροι εισόδου

**τύπος1, τυπος2, ...** καθορίζουν τους τύπους δεδομένων.

Οι παράμετροι είναι τιμές που περνούν από τον καλούντα την συνάρτηση που δημιουργείται από τον χρήστη και χρησιμοποιούνται μέσα στην συνάρτηση.

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

**προεπιλογή1, προεπιλογή2** καθορίζουν την προαιρετική προεπιλεγμένη τιμή της αντίστοιχης παραμέτρου, μπορεί επίσης να είναι NULL

**RETURNS** ορίζει ένα τύπο δεδομένων της τιμής που επιστρέφεται από την συνάρτηση που ορίζεται από τον χρήστη.

Αυτός ο τύπος δεδομένων μπορεί να είναι οποιοσδήποτε από τους πρότυπους τύπους δεδομένων που υποστηρίζονται από το SQL Server, περιλαμβανομένου και του τύπου δεδομένων TABLE.

(Οι μόνοι πρότυποι τύποι δεδομένων που δεν μπορείτε να χρησιμοποιήσετε είναι ο τύπος δεδομένων TIMESTAMP και τα δεδομένα τύπου text/image.)

---

Τμήμα Μηχανικών Πληροφορικής ΤΕ

31

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

**WITH ENCRYPTION** κρυπτογραφεί τις στήλες του πίνακα συστήματος που περιέχουν το κείμενο της πρότασης CREATE FUNCTION

**WITH SCHEMABINDING**, δεσμεύει την συνάρτηση που ορίζεται από τον χρήστη στα αντικείμενα βάσης δεδομένων στα οποία αναφέρεται.

Κάθε προσπάθεια τροποποίησης της δομής του αντικειμένου βάσης δεδομένων στο οποίο αναφέρεται η συνάρτηση αποτυγχάνει.

---

Τμήμα Μηχανικών Πληροφορικής ΤΕ

32

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

**μπλοκ** είναι το μπλοκ **BEGIN-END** που περιέχει την υλοποίηση της συνάρτησης.

Η τελική πρόταση του μπλοκ πρέπει να είναι μια πρόταση **RETURN** με ένα όρισμα.

Η τιμή του ορίσματος είναι η τιμή που επιστρέφεται από την συνάρτηση.

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

Στο σώμα ενός **μπλοκ BEGIN-END**, μόνο οι παρακάτω προτάσεις επιτρέπονται:

- Προτάσεις εκχώρησης σαν την **SET**
- Προτάσεις ελέγχου ροής σαν τις **WHILE** και **IF**
- Προτάσεις **DECLARE** που ορίζουν τοπικές μεταβλητές δεδομένων
- Προτάσεις **SELECT** που περιέχουν λίστες **SELECT** με εκφράσεις που εκχωρούν σε μεταβλητές που είναι τοπικές στην συνάρτηση
- Προτάσεις **INSERT**, **UPDATE** και **DELETE**, που τροποποιούν μεταβλητές τύπου δεδομένων **TABLE**, που είναι τοπικές στην συνάρτηση

## Συναρτήσεις που Ορίζονται από τον Χρήστη

Αυτή η συνάρτηση υπολογίζει πρόσθετα συνολικά κόστη που προκύπτουν αν αυξηθούν οι προϋπολογισμοί των έργων

```
CREATE FUNCTION computecosts (@percent REAL =10)
RETURNS DECIMAL(14,2)
AS
BEGIN
    DECLARE @additional_costs DEC (16,2),
            @sum_budget DEC(6,2)
    SELECT @sum_budget = (select SUM (budget) FROM
                                project)
    SET @additional_costs = @sum_budget *
                            @percent/100
    RETURN @additional_costs
END
```

Τμήμα Μηχανικών Πληροφορικής ΤΕ

35

## Συναρτήσεις που Ορίζονται από τον Χρήστη

**ΠΑΡΑΔΕΙΓΜΑ** συνάρτησης που επιστρέφει μια μεταβλητή τύπου TABLE.

```
CREATE FUNCTION employees_in_project (@pr_number CHAR(4))
RETURNS TABLE AS
Begin
RETURN
    (SELECT empfname, empLname
     FROM workson, employee
     WHERE employee.empno = workson.empno
     AND projectno = @pr_number)
End
```

- Η συνάρτηση `employees_in_project` χρησιμοποιείται για να εμφανίσει ονόματα όλων των υπαλλήλων που ανήκουν σε ένα συγκεκριμένο έργο.
- Η παράμετρος εισόδου `@pr_number` καθορίζει ένα αριθμό έργου.
- Ενώ η συνάρτηση γενικά επιστρέφει πολλές γραμμές,
- Η φράση `RETURNS` περιέχει τον τύπο δεδομένων `TABLE`.

Τμήμα Μηχανικών Πληροφορικής ΤΕ

36



## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

**ΠΑΡΑΔΕΙΓΜΑ** συνάρτησης που επιστρέφει μια μεταβλητή τύπου TABLE όπου

Φαίνεται η χρήση της συνάρτησης `employees_in_project`.

```
SELECT * FROM employees_in_project('p3')
```

Το αποτέλεσμα είναι

<u>fname</u>	<u>Lname</u>
aaa	aaaa
bbbb	bbbb
cccc	cccc

---

Τμήμα Μηχανικών Πληροφορικής ΤΕ

37

## Συναρτήσεις που Ορίζονται από τον Χρήστη

---

**ALTER FUNCTION**, τροποποιεί την δομή μιας συνάρτησης που ορίζεται από τον χρήστη. Η πρόταση `ALTER FUNCTION` χρησιμοποιείται συνήθως για να καταργήσει μια δέσμευση σχήματος. Όλες οι επιλογές της πρότασης `ALTER FUNCTION` αντιστοιχούν στις επιλογές με το ίδιο όνομα στην πρόταση `CREATE FUNCTION`.

**DROP FUNCTION** καταργείται μια συνάρτηση που ορίζεται από τον χρήστη. Μόνο ο ιδιοκτήτης της συνάρτησης (ή τα μέλη των σταθερών ρόλων βάσης δεδομένων `db_owner` και `sysadmin`) μπορούν να καταργήσουν την συνάρτηση.

Ο πίνακας συστήματος **sysobjects** εμπλουτίζεται στο SQL Server έτσι ώστε να μπορεί να εμφανίσει τις πληροφορίες για υπάρχουσες συναρτήσεις που ορίζονται από τον χρήστη. Η διαδικασία συστήματος **sp\_helptext** παρέχει επίσης σχετικές πληροφορίες για συναρτήσεις που ορίζονται από τον χρήστη.

---

Τμήμα Μηχανικών Πληροφορικής ΤΕ

38

## ΣΚΑΝΔΑΛΕΣ (Ερεθισμοί)- Triggers

---

Μία σκανδάλη ή αλλιώς ένας ερεθισμός είναι ένας μηχανισμός που καλείται όταν συμβαίνει μια συγκεκριμένη ενέργεια σε ένα συγκεκριμένο πίνακα και εκτελεί δέσμες εντολών που αφορούν το συγκεκριμένο πίνακα ή άλλους πίνακες. Κάθε ερεθισμός έχει τρία γενικά μέρη:

**Ένα όνομα - Μια ενέργεια - Την εκτέλεση**

Η ενέργεια ενός ερεθισμού μπορεί να είναι μια πρόταση INSERT, UPDATE ή DELETE.

Το τμήμα εκτέλεσης μιας σκανδάλης περιέχει συνήθως μια αποθηκευμένη διαδικασία ή μια δέσμη εντολών.

Μία σκανδάλη δημιουργείται χρησιμοποιώντας την πρόταση CREATE TRIGGER, που έχει την παρακάτω μορφή:

## ΣΚΑΝΔΑΛΕΣ (Ερεθισμοί)- Triggers

---

```
CREATE TRIGGER trigger name ON tablename | viewname  
{FOR | AFTER | INSTEAD OF} { [INSERT] [,] [UPDATE] [,]  
[DELETE] }  
[WITH ENCRYPTION] AS {batch | IF UPDATE (column)  
[ {AND|OR} UPDATE (column) ] batch}
```

**trigger\_name** είναι το όνομα του ερεθισμού. **table\_name** είναι το όνομα του πίνακα για τον οποίο καθορίζεται ο ερεθισμός. Στον SQL Server 2000, μπορείτε επίσης να ορίσετε ερεθισμούς για όψεις/views.

**ON tablename | viewname** μια σκανδάλη μπορεί να εφαρμόζεται μόνο σε ένα πίνακα ή όψη.

**AFTER** και **INSTEAD OF** είναι δύο πρόσθετες επιλογές, που μπορείτε να ορίσετε σε μια σκανδάλη.

**FOR** είναι συνώνυμη της AFTER. Η AFTER ξεκινά την εκτέλεση της δέσμης εντολών μετά την εκκίνηση της σκανδάλης.

**INSTEAD OF** εκτελούνται αντί του αντίστοιχου συμβάντος INSERT, UPDATE ή DELETE της σκανδάλης.

## ΣΚΑΝΔΑΛΕΣ (Ερεθισμοί)- Triggers

---

- Σε κάθε εκτέλεση μιας σκανδάλης δημιουργούνται δύο εικονικοί πίνακες με ειδικά ονόματα. Ο πίνακας **deleted** και ο πίνακας **inserted**.
- Η δομή αυτών των πινάκων είναι ισοδύναμη με την δομή του πίνακα στον οποίο αναφέρεται η σκανδάλη. Ο πίνακας **deleted** περιέχει αντίγραφα από τις γραμμές που διαγράφονται από τον πίνακα που έχει εφαρμοσθεί η σκανδάλη.
- Το ίδιο ισχύει και για τον πίνακα **inserted** όπου δημιουργούνται αντίγραφα των γραμμών που εισάγονται στον πίνακα που έχει εφαρμοσθεί η σκανδάλη.

## ΣΚΑΝΔΑΛΕΣ (Ερεθισμοί)- Triggers

---

- Αν στην σκανδάλη υπάρχει μια εντολή UPDATE, τότε στον πίνακα **deleted** μπαίνουν τα δεδομένα πριν από την τροποποίηση και στον πίνακα **inserted** μπαίνουν δεδομένα μετά την τροποποίηση.
- Ο πίνακας **deleted** χρησιμοποιείται αν υπάρχει η εντολή **DELETE** ή **UPDATE** μέσα στην CREATE TRIGGER. Ο πίνακας **inserted** χρησιμοποιείται υπάρχει η εντολή **INSERT** ή **UPDATE** μέσα στην CREATE TRIGGER.

## ΠΑΡΑΔΕΙΓΜΑ Create trigger

```
*
* Δημιουργήστε έναν πίνακα με όνομα LOG_AUTO στον οποίο με την εφαρμογή μιας αναίδηλης (TRIGGER) που θα εδραζόταν
* πάνω στον βασικό πίνακα ΑΥΤΟΚΙΝΗΤΑ, θα αποθηκεύονται οι τροποποιήσεις που θα γίνουν στο πεδίο ΤΙΜΗ_ΕΝΟΙΚΙΑΣ του
* πίνακα ΑΥΤΟΚΙΝΗΤΑ.
*
* CREATE TABLE LOG_AUTO( AUTO_NUM VARCHAR(10),
*                          USER_ID VARCHAR(20),
*                          ΗΜΕΡΟΜΗΝΙΑ DATETIME,
*                          OLD FLOAT,
*                          NEW FLOAT)
*
*
* CREATE TRIGGER TRIG_LOG ON ΑΥΤΟΚΙΝΗΤΑ
* AFTER UPDATE AS
* IF UPDATE(ΤΙΜΗ_ΕΝΟΙΚΙΑΣ)
* BEGIN
* DECLARE @TIMH_olD FLOAT
* DECLARE @TIMH_new FLOAT
* DECLARE @auto_number VARCHAR(10)
* SELECT @TIMH_olD = (SELECT ΤΙΜΗ_ΕΝΟΙΚΙΑΣ FROM deleted)
* SELECT @TIMH_new = (SELECT ΤΙΜΗ_ΕΝΟΙΚΙΑΣ FROM inserted)
* SELECT @auto_number = (SELECT ΚΑ FROM deleted)
* INSERT INTO LOG_AUTO VALUES
* (@auto_number,USER_NAME(),GETDATE(),@TIMH_olD, @TIMH_new)
* END
*
* UPDATE ΑΥΤΟΚΙΝΗΤΑ
* SET ΤΙΜΗ_ΕΝΟΙΚΙΑΣ=200 WHERE ΚΑ=1
*
* Το περιεχόμενο του πίνακα LOG_AUTO θα είναι ως εξής:
```

AUTO_NUM	USER_ID	ΗΜΕΡΟΜΗΝΙΑ	OLD	NEW
1	200,00	000	2005-05-05 11:51	300,00

## Ενότητα 6

---

### Ανάκαμψη και συναλλαγές

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Συναλλαγές
- Ταυτοχρονισμός σε ΣΔΒΔ
- Πράξεις μιας συναλλαγής
- Ημερολόγιο συστήματος (log)
- Οι ιδιότητες μιας ACID συναλλαγής
- Ορισμός συναλλαγής
- Ορισμός χρονοπρογράμματος
- Σύγκρουση πράξεων σε χρονοπρογραμματισμό
- Σειριοποιησιμότητα
- Ισοδυναμία Χρονοπρογραμμάτων βάσει Συγκρούσεων
- Σειριοποιησιμότητα βάσει Συγκρούσεων
- Ανάκαμψη από Αποτυχίες
- Ανάκαμψη μέσων
- Επικύρωση δύο φάσεων



# Σκοποί ενότητας

Στο κεφάλαιο αυτό περιγράφονται οι συναλλαγές, δηλαδή η εκτέλεση ενός ή περισσότερων προγραμμάτων που προσπελούν ή τροποποιούν το περιεχόμενο της βάσης δεδομένων. Εξηγούνται οι ιδιότητες ACID μιας συναλλαγής καθώς και τις πράξεις στα χρονοπρογράμματα. Παρουσιάζονται τα είδη των αποτυχιών καθώς και τρόπους ανάκαμψης ενός ΣΔΒΔ από πιθανή αστοχία συστήματος ή μέσων .

## Συναλλαγές

- Η ταυτόχρονη εκτέλεση προγραμμάτων χρηστών είναι απαραίτητη για την καλή απόδοση ενός ΣΔΒΔ.
- Επειδή οι προσπελάσεις στο δίσκο είναι συχνές και σχετικά αργές, είναι σημαντικό να κρατείται η CPU απασχολημένη με πολλά προγράμματα χρηστών.

⇒ Πολυχρηστικά ΣΔΒΔ

Διαπλεγμένο μοντέλο ταυτόχρονης εκτέλεσης



# Συναλλαγές

---

## Συναλλαγή (transaction) :

- η εκτέλεση ενός προγράμματος που προσπελαίνει ή τροποποιεί το περιεχόμενο της βάσης δεδομένων.
- το πώς βλέπει το **ΣΔΒΔ** τα προγράμματα των χρηστών

# Συναλλαγές

---

## Συναλλαγή (transaction) :

Ένα πρόγραμμα χρήστη μπορεί να εκτελεί πολλές λειτουργίες στα δεδομένα που ανακτά από τη ΒΔ, αλλά το ΣΔΒΔ ενδιαφέρεται μόνο για τα δεδομένα που διαβάζονται/γράφονται στη ΒΔ.

- Ανάγνωση (X) - R(X)
- Εγγραφή (X) - W(X)

## Ταυτοχρονισμός σε ΣΔΒΔ

---

- Οι χρήστες υποβάλουν συναλλαγές, πρέπει να μπορούν να θεωρούν ότι κάθε συναλλαγή εκτελείται μόνη της.
- Ο Ταυτοχρονισμός ή συνδρομικότητα (concurrency) επιτυγχάνεται από το ΣΔΒΔ που διαπλέκει τις πράξεις (αναγνώσεις/εγγραφές) των συναλλαγών αυτών.

## Συναλλαγές

---

- Κάθε Συναλλαγή πρέπει να αφήνει τη ΒΔ στην κατάσταση που ήταν όταν άρχισε η Συναλλαγή.
- Το ΣΔΒΔ επιβάλλει κάποιους ΠΑ (Περιορισμούς Ακεραιότητας) με βάση τους ΠΑ που έχουν δηλωθεί στις εντολές CREATE TABLE.
- Πέρα από αυτό, το ΣΔΒΔ δεν καταλαβαίνει τη σημασιολογία των δεδομένων (π.χ., δεν καταλαβαίνει πώς να υπολογίσει το επιτόκιο).



# Πράξεις μιας συναλλαγής

---

## Πράξεις συναλλαγών

- **BEGIN**  
R(X)  
W(X)  
END
- **COMMIT (επικύρωση)** - επιτυχία - όλες οι τροποποιήσεις επικυρώνονται και δεν μπορούν να αναιρεθούν.
- **ABORT ή ROLLBACK (ακύρωση ή ανάκληση)** - αποτυχία - όλες οι τροποποιήσεις πρέπει να αναιρεθούν.

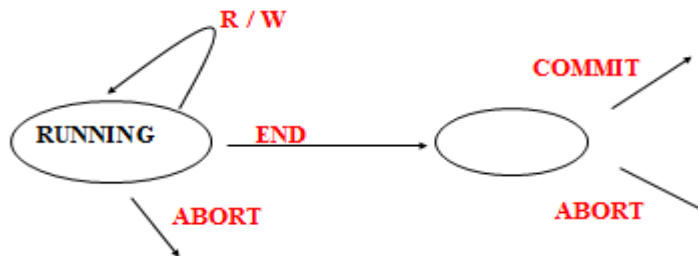
# Πράξεις μιας συναλλαγής

---

- Μια Συναλλαγή μπορεί να επικυρωθεί (commit) αφού ολοκληρώσει όλες τις πράξεις της. Μπορεί να ακυρωθεί (abort- ROLLBACK) όμως, αφού εκτελέσει κάποιες από τις πράξεις της.
- Το ΣΔΒΔ καταγράφει (logs) όλες τις πράξεις έτσι ώστε να μπορεί να αναιρέσει (undo) τις πράξεις μιας ακυρωμένης (aborted) Συναλλαγής.

## Πράξεις μιας συναλλαγής

---



## Ημερολόγιο Συστήματος (log)

---

- Για να είναι δυνατή η ανάκαμψη από αποτυχίες, καταχωρούνται πληροφορίες για τις πράξεις των Συναλλαγών.
- Αποθηκεύονται στο δίσκο.
- Τύποι πληροφορίας: έναρξη Συναλλαγής  
εγγραφή στοιχείου (παλιά, νέα)  
ανάγνωση στοιχείου  
επικύρωση/ακύρωση
- Είναι δυνατή η αναίρεση (undo) ή η επανάληψη (redo) μιας Συναλλαγής.

# Οι ιδιότητες μιας ACID συναλλαγής

---

- **Atomicity (ατομικότητα)** - είτε όλες οι πράξεις εκτελούνται, είτε καμία.
- **Consistency (συνέπεια)** - διατήρηση συνέπειας της ΒΔ δηλαδή μετασχηματίζει μια συνεπή κατάσταση της βάσης δεδομένων σε μια άλλη συνεπή κατάσταση, χωρίς κατανάγκη να διατηρεί τη συνέπεια σε όλα τα ενδιάμεσα σημεία.
- **Isolation (απομόνωση)** - δεν αποκαλύπτει ενδιάμεσα αποτελέσματα.
- **Durability (μονιμότητα ή διάρκεια)** - μετά την επικύρωση μιας Συναλλαγής οι αλλαγές δεν είναι δυνατόν να χαθούν.

## Ορισμός συναλλαγής

---

Μια **Συναλλαγή** είναι μια ακολουθία από πράξεις εγγραφής και ανάγνωσης που τελειώνει με μια πράξη επικύρωσης (commit) ή με μια πράξη ακύρωσης (abort-rollback).

Θεωρείστε τις δύο συναλλαγές (*Acts*) του παραδείγματος:

```
T1: BEGIN R(X), X=X-N, W(X), R(Y), Y=Y+N, W(Y) END
T2: BEGIN R(X) X=X+M, W(X) END
```

```
T1: R(X) W(X) R(Y) W(Y) C
```

```
T2: R(X) W(X) C
```

## Ορισμός χρονοπρογράμματος

- Ένα **χρονοπρόγραμμα (schedule)**  $S$  των Συναλλαγών  $T_1, T_2, \dots, T_n$  είναι μια διάταξη των πράξεων τους με τον περιορισμό ότι για κάθε Συναλλαγή  $T_i$  που συμμετέχει στο  $S$  οι πράξεις της  $T_i$  στο  $S$  πρέπει να εμφανίζονται με την ίδια σειρά που εμφανίζονται στην  $T_i$ .
- Οι πράξεις των Συναλλαγών εμφανίζονται στο χρονοπρόγραμμα με τη σειρά που εκτελούνται.

## Ορισμός χρονοπρογράμματος

Θα χρησιμοποιούμε δείκτη στις πράξεις που να δείχνει σε ποια Συναλλαγή.

T1	T2
R1(X) W1(X) R1(Y) W1(Y) C1	R2(X) W2(X) C2

**S: R<sub>1</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>1</sub>(Y) C<sub>1</sub> R<sub>2</sub>(X) W<sub>2</sub>(X) C<sub>2</sub>**

# Ορισμός χρονοπρογράμματος

T1	T2
R1(X)	R2(X)
W1(X) R1(Y)	W2(X) C2
W1(Y) C1	

**S: R<sub>1</sub>(X) R<sub>2</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>2</sub>(X) C<sub>2</sub> W<sub>1</sub>(Y) C<sub>1</sub>**

## Σύγκρουση πράξεων σε χρονοπρογραμματισμό

Δύο πράξεις σε ένα χρονοπρόγραμμα συγκρούονται αν:

- (α) ανήκουν σε διαφορετικές Συναλλαγές
- (β) προσπελαίνουν το ίδιο στοιχείο
- (γ) μια από αυτές είναι πράξη εγγραφής (W)

**Ερώτηση:**

Ποια είναι η σχέση των χρονοπρογραμμάτων S1 και S2;

**S1: R<sub>1</sub>(X) R<sub>2</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>2</sub>(X) C<sub>2</sub> W<sub>1</sub>(Y) C<sub>1</sub>**

**S2: R<sub>2</sub>(X) R<sub>1</sub>(X) W<sub>1</sub>(X) R<sub>1</sub>(Y) W<sub>2</sub>(X) C<sub>2</sub> W<sub>1</sub>(Y) C<sub>1</sub>**

Τα **S1** και **S2** είναι ισοδύναμα, διαφέρουν μόνο στη διάταξη πράξεων που δε συγκρούονται άρα έχουμε μερική διάταξη.

# Ορισμός χρονοπρογραμματισμού

---

Ένα πλήρες χρονοπρόγραμμα (schedule)  $S$  των Συναλλαγών  $T_1, T_2, \dots, T_n$  είναι ένα σύνολο από πράξεις και μια μερική διάταξη των πράξεων αυτών με τους ακόλουθους περιορισμούς:

(α) οι πράξεις του  $S$  είναι ακριβώς οι πράξεις των  $T_1, T_2, \dots, T_n$  συμπεριλαμβανομένης μιας πράξης ακύρωσης ή επικύρωσης ως τελευταίας πράξης σε κάθε Συναλλαγή στο χρονοπρόγραμμα.

(β) για κάθε Συναλλαγή  $T_i$  που συμμετέχει στο  $S$  οι πράξεις της  $T_i$  στο  $S$  πρέπει να εμφανίζονται με την ίδια σειρά που εμφανίζονται στην  $T_i$

(γ) Για κάθε ζεύγος συγκρουόμενων πράξεων, μια από τις δύο πρέπει να προηγείται της άλλης στο χρονοπρόγραμμα.

# Σειριοποιησιμότητα

---

## Σειριακά Χρονοπρογράμματα:

Χρονοπρογράμματα που δεν διαπλέκουν πράξεις διαφορετικών Συναλλαγών δηλαδή οι πράξεις κάθε Συναλλαγής εκτελούνται διαδοχικά, χωρίς παρεμβολή πράξεων από άλλη Συναλλαγή.

**Παρατήρηση:** Αν κάθε Συναλλαγή διατηρεί τη συνέπεια, τότε κάθε σειριακό χρονοπρόγραμμα διατηρεί τη συνέπεια.

Ένα σειριακό χρονοπρόγραμμα, όπως το παρακάτω, είναι σωστό

**S:**  $R_1(X) W_1(X) R_1(Y) W_1(Y) C_1 R_2(X) W_2(X) C_2$

# Σειριοποιησιμότητα

---

## Ισοδύναμα Χρονοπρογράμματα :

- Για κάθε κατάσταση της ΒΔ, το αποτέλεσμα της εκτέλεσης του πρώτου χρονοπρογράμματος είναι το ίδιο με το αποτέλεσμα του δεύτερου χρονοπρογράμματος. Ένα χρονοπρόγραμμα ισοδύναμο με ένα σειριακό.

# Σειριοποιησιμότητα

---

## Σειριοποιήσιμο Χρονοπρόγραμμα :

Ένα χρονοπρόγραμμα που είναι *ισοδύναμο* με κάποιο σειριακό

## Τι σημαίνει ισοδύναμο;

- Ισοδυναμία βάσει συγκρούσεων
- Ισοδυναμία όψεων

## Ισοδυναμία Χρονοπρογραμμάτων βάσει Συγκρούσεων

---

Δυο χρονοπρογράμματα είναι ισοδύναμα βάσει συγκρούσεων αν η διάταξη κάθε ζεύγους συγκρουόμενων πράξεων είναι ίδια και στα δυο χρονοπρογράμματα.

### Παραδείγματα:

S1:  $R_1(X) R_2(X) W_1(X) R_1(Y) W_2(X) C_2 W_1(Y) C_1$

S2:  $R_2(X) R_1(X) W_1(X) R_1(Y) W_2(X) C_2 W_1(Y) C_1$

S3:  $R_2(X) W_2(X) C_2 R_1(X) W_1(X) R_1(Y) W_1(Y) C_1$

S4:  $R_2(X) R_1(X) W_1(X) R_1(Y) W_1(Y) C_1 W_2(X) C_2$

## Σειριοποιησιμότητα βάσει Συγκρούσεων

---

### • Σειριοποιησιμότητα βάσει Συγκρούσεων:

Ένα χρονοπρόγραμμα  $S$  είναι σειριοποιήσιμο βάσει συγκρούσεων αν είναι ισοδύναμο βάσει συγκρούσεων με κάποιο σειριακό χρονοπρόγραμμα  $S'$ .

Σε αυτήν την περίπτωση μπορούμε να αναδιατάξουμε τις μη συγκρουόμενες πράξεις στο  $S$  μέχρι να σχηματίσουμε ένα ισοδύναμο σειριακό χρονοπρόγραμμα.



## Σειριοποιησιμότητα βάσει Συγκρούσεων

---

S1:  $R_1(X) R_2(X) W_1(X) R_1(Y) W_2(X) C_2 W_1(Y) C_1$

S2:  $R_2(X) R_1(X) W_1(X) R_1(Y) W_2(X) C_2 W_1(Y) C_1$

### Σειριοποίησιμα:

Sα:  $R_1(X) W_1(X) R_1(Y) W_1(Y) C_1 R_2(X) W_2(X) C_2$

Sβ:  $R_2(X) W_2(X) C_2 R_1(X) W_1(X) R_1(Y) W_1(Y) C_1$

## Ανάκαμψη από Αποτυχίες

---

- Όταν μια Συναλλαγή υποβάλλεται στο ΣΔΒΔ το σύστημα πρέπει να εξασφαλίσει ότι :
  - (α) είτε όλες οι πράξεις της θα ολοκληρωθούν είτε .
  - (β) καμία δε θα εκτελεστεί- δηλαδή δε θα έχει καμία επίδραση στη ΒΔ -- ακόμα και αν συμβούν αποτυχίες.
- Αυτή είναι μια σημαντική ιδιότητα που πρέπει να εξασφαλίσει το ΣΔΒΔ - Ο χρήστης πρέπει να μπορεί να θεωρεί ότι όλο το πρόγραμμα (πράξεις) εκτελούνται σε ένα βήμα είτε καμία πράξη δεν εκτελείται (ατομικότητα των Συναλλαγών)

# Ανάκαμψη από Αποτυχίες

---

## Είδη Αποτυχιών :

Δυο κατηγορίες: καταστροφή ή όχι της μόνιμης αποθήκευσης (δίσκου).

### **Παραδείγματα αποτυχιών:**

Αστοχίες συστήματος (διακοπή ρεύματος).

Αστοχίες μέσων (καταστροφή δίσκου).



# Ανάκαμψη από Αποτυχίες

---

## Ερώτηση:

Πώς θα ξέρει το σύστημα κατά την επανεκκίνηση ποιες συναλλαγές πρέπει να αναιρέσει και ποιες να επανεκτελέσει;

## Απάντηση:

Σε κάποια προκαθορισμένα χρονικά διαστήματα — κατά κανόνα, όποτε έχει γραφεί ένας προκαθορισμένος αριθμός καταχωρίσεων στο αρχείο πεπραγμένων log— το σύστημα αυτόματα θέτει ένα σημείο ελέγχου (checkpoint).



## Ανάκαμψη από Αποτυχίες

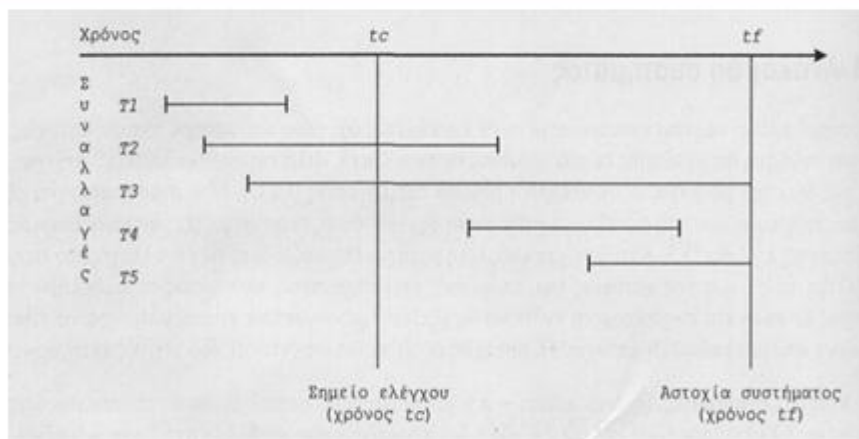
Η τοποθέτηση ενός σημείου ελέγχου σημαίνει:

**(α)** φυσική καταγραφή "καταναγκαστική καταγραφή" των περιεχομένων των περιοχών προσωρινής αποθήκευσης στη φυσική βάση δεδομένων, και

**(β)** φυσική καταγραφή μιας ειδικής εγγραφής σημείου ελέγχου (checkpoint record) στο φυσικό αρχείο πεπραγμένων.

Η εγγραφή του σημείου ελέγχου δίνει μια λίστα όλων των συναλλαγών που ήταν σε εξέλιξη τη στιγμή που τέθηκε το σημείο ελέγχου.

## Ανάκαμψη από Αποτυχίες



## Ανάκαμψη από Αποτυχίες

---

Κατά την επανεκκίνηση λουπόν, το σύστημα πρώτα ακολουθεί την παρακάτω διαδικασία, για να αναγνωρίσει όλες τις συναλλαγές των τύπων T2-T5:

1. Η διαδικασία ξεκινάει με δύο λίστες συναλλαγών, τη λίστα αναιρέσεων (UNDO) και τη λίστα επανεκτελέσεων (REDO). Αρχικά, η λίστα UNDO είναι εξ ορισμού ίση με τη λίστα όλων των συναλλαγών στην πιο πρόσφατη εγγραφή σημείου ελέγχου' η λίστα REDO είναι εξ ορισμού κενή.
2. Γίνεται αναζήτηση προς τα εμπρός μέσα στο αρχείο πεπραγμένων, ξεκινώντας από την εγγραφή του σημείου ελέγχου.

## Ανάκαμψη από Αποτυχίες

---

3. Αν βρεθεί στο log μια καταχώριση BEGIN TRANSACTION για τη συναλλαγή T, προστίθεται η T στη λίστα UNDO.
4. Αν βρεθεί στο αρχείο πεπραγμένων μια καταχώριση COMMIT για τη συναλλαγή T, μεταφέρεται η T από τη λίστα UNDO στη λίστα REDO.
5. Όταν η αναζήτηση φτάσει στο τέλος του αρχείου πεπραγμένων, οι λίστες UNDO και REDO προσδιορίζουν, αντίστοιχα, τις συναλλαγές των τύπων T3 και T5 και τις συναλλαγές των τύπων T2 και T4.

## Ανάκαμψη μέσων

---

Για την ανάκαμψη από μια τέτοια αστοχία, απαιτείται να γίνει επαναφόρτωση (ή επαναφορά - restore) της βάσης δεδομένων από ένα εφεδρικό αντίγραφο (database backup).

Έπειτα χρησιμοποιείτε το αρχείο πεπραγμένων - τόσο το ενεργό όσο και το αρχειακό μέρος του, για να επανεκτελεστούν - όλες οι συναλλαγές που ολοκληρώθηκαν μετά από τη στιγμή που πάρθηκε το εφεδρικό αντίγραφο.

Οι συναλλαγές που ήταν ακόμα σε εξέλιξη τη στιγμή της αστοχίας, έχουν "ανααιρεθεί" (για την ακρίβεια, έχουν χαθεί), έτσι και αλλιώς.

## Ανάκαμψη μέσων

---

Προκύπτει η ανάγκη να έχουμε ένα βοηθητικό πρόγραμμα :

- αποτύπωσης/επαναφοράς (dump/restore),  
ή εκφόρτωσης/επαναφόρτωσης (unload/reload),
  - εφεδρικών αντιγράφων.
- ✓ Το μέρος αυτού του βοηθητικού προγράμματος που ασχολείται με την αποτύπωση (dump) χρησιμοποιείται για τη δημιουργία εφεδρικών αντιγράφων της βάσης δεδομένων
- ✓ Αντίγραφα μπορεί να τηρούνται σε ταινία ή σε κάποιο άλλο αρχειακό μέσο αποθήκευσης, δεν είναι απαραίτητο να βρίσκονται σε ένα μέσο άμεσης προσπέλασης)
- ✓ Μετά από μια αστοχία μέσου, γίνεται επαναφορά (restore) και ξανα-δημιουργείται η βάση δεδομένων από ένα καθορισμένο εφεδρικό αντίγραφο.

## Επικύρωση δύο φάσεων

---

### A.

Καταναγκαστική καταγραφή όλων των καταχωρίσεων πεπραγμένων της συναλλαγής σε μόνιμο μέσο αποθήκευσης.

ότι και αν συμβεί μετά, ο διαχειριστής πόρων θα έχει μια μόνιμη Εγγραφή της εργασίας που έκανε για λογαριασμό της συναλλαγής, και έτσι θα μπορεί να επικυρώσει τις ενημερώσεις της ή να τις ανασκευάσει, ανάλογα με την περίπτωση.

### B.

Επικύρωση ή ανασκευή τη συναλλαγής τοπικά, ανάλογα με την τελική υπόδειξη (commit, rollback).

## Ενότητα 7

---

### Ταυτοχρονισμός

Δρ. Τσιμπίρης Αλκιβιάδης



---

### Περιεχόμενα ενότητας

---

- Ταυτοχρονισμός
- Προβλήματα ταυτοχρονισμού
- Χαμένη ενημέρωση
- Εξάρτηση από αναπικύρωτη μεταβολή
- Ασυνεπής Ανάλυση
- Κλειδώμα (Αποκλειστικό-Μεριζόμενο)
- Αδιέξοδο
- Λύση των προβλημάτων ταυτοχρονισμού.
- Προτιθέμενα κλειδώματα



# Σκοποί ενότητας

Εξηγούνται τα προβλήματα που προκύπτουν από ταυτόχρονη εκτέλεση συναλλαγών καθώς και τις τεχνικές που χρησιμοποιούνται για τον έλεγχο της ταυτόχρονης εκτέλεσης συναλλαγών που βασίζονται στην έννοια του κλειδώματος των δεδομένων. Τέλος παρουσιάζονται τα προβλήματα αδιεξόδου καθώς και οι υπάρχουσες λύσεις.



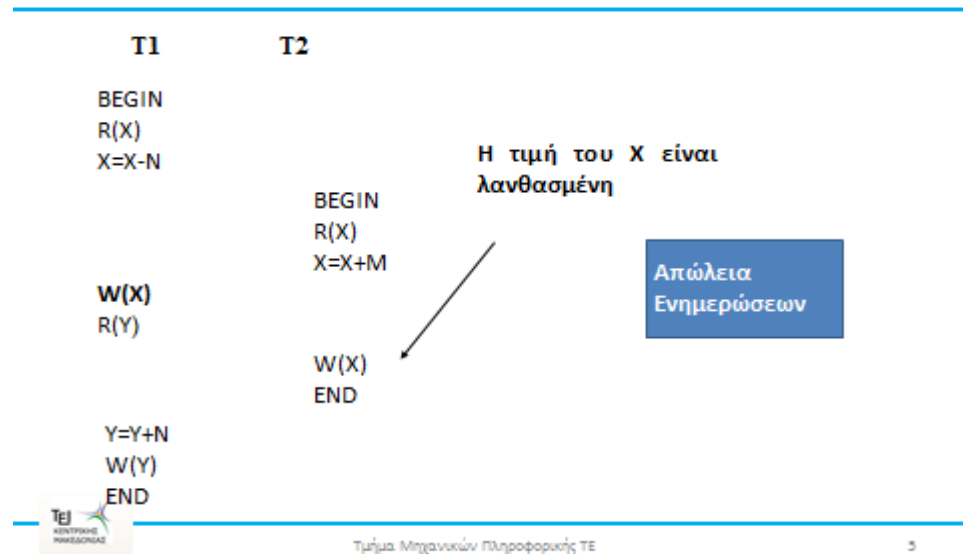
# Ταυτοχρονισμός

T1	T2	T1	T2
BEGIN R(X) X=X-N W(X) R(Y) Y=Y+N W(Y) END	<b>σειριακή εκτέλεση</b> T1 → T2	BEGIN R(X) X=X-N W(X) R(Y) Y=Y+N W(Y) END	BEGIN R(X) X=X+M W(X) END <b>σειριακή εκτέλεση</b> T2 → T1

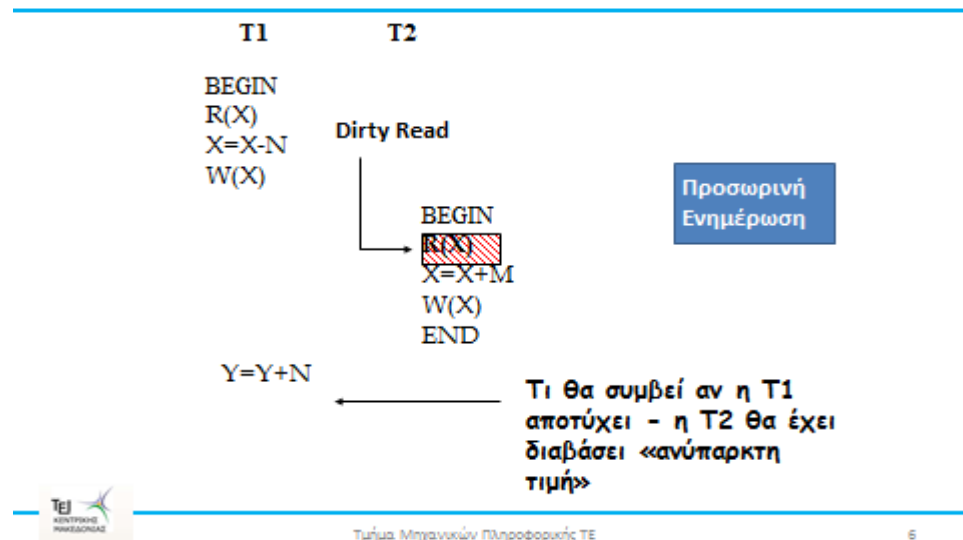




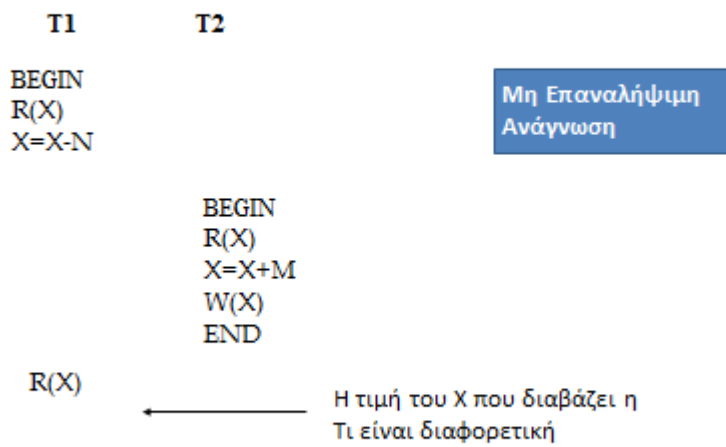
# Ταυτοχρονισμός



# Ταυτοχρονισμός



# Ταυτοχρονισμός



## Προβλήματα ταυτοχρονισμού

Τα προβλήματα του ταυτοχρονισμού είναι:

- Το πρόβλημα της **χαμένης ενημέρωσης** (lost update)
- Το πρόβλημα της **εξάρτησης από ανεπικύρωτη μεταβολή** (uncommitted dependency)
- Το πρόβλημα της **ασυνεπούς ανάλυσης** (inconsistent analysis)



## Χαμένη ενημέρωση ταυτοχρονισμού

---

### Χαμένη Ενημέρωση

A			B
-----			
Retrieve p	t1	-	
-		t2	Retrive p
Update p	t3	-	
-		t4	Update p



## Εξάρτηση από ανεπικύρωτη μεταβολή ταυτοχρονισμού

---

### Εξάρτηση από ανεπικύρωτη μεταβολή

A		B
-----		
----	t1	Update p
Retrieve p	t2	-
Update p	t3	
----	t4	Rollback



## Ασυνεπής ανάλυση ταυτοχρονισμού

### Ασυνεπής ανάλυση

A	[ACC1=40, ACC2=50, ACC3=30]	B	
Retrieve ACC1 sum=40	t1	-	
Retrieve ACC2 sum=90	t2	-	
-		t3	Retrieve ACC3
-		t4	Update ACC3 30--
>20			
-		t5	Retrieve ACC1
-		t6	Update ACC1 40--
>50			
-		t7	COMMIT
Reatrive ACC3 sum=110 no 120	t8	-	

## Τεχνική έλεγχου ταυτοχρονισμού

Μία τεχνική ελέγχου που χρησιμοποιεί ο ταυτοχρονισμός είναι το κλείδωμα:

**Κλείδωμα (locking)** η λειτουργία του έχει σαν αποτέλεσμα τον αποκλεισμό των συναλλαγών από την συστοιχία ώστε αν μην υπάρχει δυνατότητα μεταβολής της . Υπάρχουν δύο είδη κλειδώματος:

- **Αποκλειστικά κλειδώματα X (exclusive locks)** κλειδωμα πράξεων εγγραφής (write lock)
- **Μεριζόμενα κλειδώματα S (shared locks)** κλειδωμα πράξεων ανάγνωσης (read lock)

# Κλείδωμα

---

Έστω ότι η συναλλαγή A κατέχει ένα αποκλειστικό κλείδωμα (X) πάνω στη συστοιχία  $\rho$ , τότε μια αίτηση από κάποια άλλη συναλλαγή B για ένα κλείδωμα οποιουδήποτε από τους δύο τύπους στην  $\rho$  θα **απορριφθεί**.

# Κλείδωμα

---

Έστω το A κατέχει ένα μεριζόμενο (S) κλείδωμα πάνω στη συστοιχία  $\rho$ , τότε:

- Μια αίτηση από κάποια άλλη συναλλαγή B για ένα κλείδωμα X στην  $\rho$  θα **απορριφθεί**.
- Μια αίτηση από κάποια άλλη συναλλαγή B για ένα κλείδωμα S στην  $\rho$  θα γίνει **δεκτή** (δηλαδή, και η B θα διατηρεί τώρα ένα κλείδωμα S στην  $\rho$ ).

## Μητρώο συμβατότητας ταυτοχρονισμού

**ΜΗΤΡΩΟ ΣΥΜΒΑΤΟΤΗΤΑΣ** (Compatibility Matrix): για τους τύπους κλειδωμάτων X και S.

	X	S	-
X	Όχι	Όχι	Ναι
S	Όχι	Ναι	Ναι
-	Ναι	Ναι	Ναι

**Σχόλιο!** Η παύλα = χωρίς κλείδωμα.



## Πρωτόκολλο προσπέλασης δεδομένων ταυτοχρονισμού

**Πρωτόκολλο προσπέλασης δεδομένων** (data access protocol):

- Μια συναλλαγή που θέλει να ανακαλέσει μια συστοιχία πρέπει πρώτα να αποκτήσει ένα κλείδωμα S σε αυτή τη συστοιχία.
- Μια συναλλαγή που θέλει να ενημερώσει μια συστοιχία πρέπει πρώτα να αποκτήσει ένα κλείδωμα X σε αυτή τη συστοιχία. Αν κατέχει ήδη ένα κλείδωμα S στη συστοιχία, όπως συμβαίνει σε μια ακολουθία ανάκλησης και ενημέρωσης (RETRIEVE-UPDATE), τότε θα πρέπει να προαγάγει το κλείδωμα S σε κλείδωμα επιπέδου X.
- Αν μια αίτηση κλειδώματος από τη συναλλαγή B απορριφθεί επειδή έρχεται σε σύγκρουση με ένα κλείδωμα που κατέχει ήδη η συναλλαγή A, τότε η συναλλαγή B περνάει σε κατάσταση αναμονής.
- Τα κλειδώματα X διατηρούνται μέχρι το τέλος της συναλλαγής (COMMIT ή ROLLBACK).



# Αδιέξοδο

---

Με την λέξη αδιέξοδο (Deadlock) εννοείται μία κατάσταση κατά την οποία δύο ή περισσότερες συναλλαγές βρίσκονται ταυτόχρονα σε κατάσταση αναμονής και η κάθε μία από αυτές περιμένει την άλλη ή τις άλλες να απελευθερώσουν ένα κλειδωμα πριν μπορέσει να συνεχίσει.

## Αδιέξοδο-Το πρόβλημα της χαμένης ενημέρωσης

---

Το πρόβλημα της χαμένης ενημέρωσης

A			B
retrieve p S	t1	-	
-		t2	retrieve p S
update p X	t3	-	
αναμονή	t4		update p X
αναμονή		αναμονή	

## Αδιέξοδο-Πρόβλημα εξάρτησης ανεπικύρωτης μεταβολής

---

Το πρόβλημα Εξάρτησης από Ανεπικύρωτη Μεταβολή

A		B
----	t1	Update p X
Retrieve p αναμονή	t2	-
αναμονή	t3	Rollback/ Commit
Συνέχεια Retrieve p S	t4	



## Αδιέξοδο-Πρόβλημα ασυνεπούς ανάλυσης

---

Το πρόβλημα Ασυνεπούς Ανάλυσης

A	<i>[ACC1=40, ACC2=50, ACC3=30]</i>	B
Retrieve S ACC1 sum=40	t1	-
Retrieve S ACC2 sum=90	t2	-
-		t3      Retrieve S ACC3
-		t4      Update X ACC3
30-->20		
-		t5      Retrieve S ACC1
-		t6      Update X ACC1
40-->50		
Retrieve S ACC3	t7	Αναμονή
Αναμονή	t8	Αναμονή





## Παράδειγμα ταυτοχρονισμού

---

### Παράδειγμα

A			B
Lock p1 X	t1	-	
-		t2	Lock p2 X
Lock p2 X	t3	-	
αναμονή	t4		Lock p1 X
αναμονή			αναμονή

## Γενικό συμπέρασμα αδιεξόδου

---

Δεν είναι δυνατό όλα τα συστήματα να εντοπίσουν ταυτόχρονα όλα τα αδιέξοδα. Μερικά χρησιμοποιούν απλώς έναν περιορισμό χρονικού ορίου και θεωρούν ότι μία συναλλαγή που δεν εκπλήρωσε την δουλειά της σε συγκεκριμένο χρονικό διάστημα βρίσκεται σε αδιέξοδο.

## Πρωτόκολλο κλειδώματος δύο φάσεων

---

1. Πριν επενεργήσει σε οποιοδήποτε αντικείμενο (π.χ., σε μια συστοιχία βάσης δεδομένων), μια συναλλαγή πρέπει να αποκτήσει ένα κλείδωμα σε αυτό το αντικείμενο.
2. Αφού απελευθερώσει ένα κλείδωμα, μια συναλλαγή δεν πρέπει ποτέ να προχωρήσει στην απόκτηση και άλλων κλειδωμάτων.

Μια συναλλαγή που υπακούει σε αυτό το πρωτόκολλο έχει λοιπόν δύο φάσεις, μια φάση απόκτησης κλειδώματος και μια φάση απελευθέρωσης κλειδώματος

## Προτιθέμενο κλείδωμα

---

Αναφέραμε ήδη ότι τα κλειδώματα X και S έχουν νόημα και για ολόκληρες σχέσεις και για μεμονωμένες συστοιχίες.

Εισάγουμε τρία νέα είδη κλειδωμάτων, που ονομάζονται προτιθέμενα κλειδώματα (intent locks), τα οποία έχουν επίσης νόημα για τις σχέσεις αλλά όχι για τις μεμονωμένες συστοιχίες.

**προτιθέμενο μεριζόμενο κλείδωμα** (intent shared — IS),

**προτιθέμενο αποκλειστικό κλείδωμα** (intent exclusive — IX)

**μεριζόμενο και προτιθέμενο αποκλειστικό κλείδωμα** (shared intent exclusive — SIX).

## Προτιθέμενο κλειδίωμα

- **IS** : Η συναλλαγή T σκοπεύει να θέσει κλειδιάματα S σε μεμονωμένες συστοιχίες της R, για να εξασφαλιστεί η σταθερότητα αυτών των συστοιχιών ενώ γίνεται η επεξεργασία τους.
- **IX** : Το ίδιο με το IS, και επιπλέον η T μπορεί να χρειαστεί να ενημερώνει μεμονωμένες συστοιχίες της R, γι' αυτό θέτει κλειδιάματα X σε αυτές τις συστοιχίες.
- **S** : Η T ανέχεται ταυτόχρονες αναγνώσεις, αλλά όχι ταυτόχρονες ενημερώσεις της R. Η ίδια η T δεν ενημερώνει συστοιχίες της R.
- **SIX** : Συνδυάζει τα S και IX, δηλαδή, η T ανέχεται ταυτόχρονες αναγνώσεις, αλλά όχι ταυτόχρονες ενημερώσεις της R, και επιπλέον η T μπορεί να χρειαστεί να ενημερώνει μεμονωμένες συστοιχίες της R, γι' αυτό θέτει κλειδιάματα X σε αυτές τις συστοιχίες.
- **X** : Η T δεν ανέχεται καμία απολύτως ταυτόχρονη προσπέλαση της R, η ίδια η T μπορεί να ενημερώνει ή να μην ενημερώνει μεμονωμένες συστοιχίες της R.

## Μητρώο συμβατότητας

Μητρώο συμβατότητας, διευρυμένο ώστε να περιλαμβάνει τα προτιθέμενα κλειδιάματα

	X	SIX	IX	S	IS	-
X	0	0	0	0	0	N
SIX	0	0	0	0	N	N
IX	0	0	N	0	N	N
S	0	0	0	N	N	N
IS	0	N	N	N	N	N
-	N	N	N	N	N	N

## Ενότητα 8

---

### Ασφάλεια ΒΔ

Δρ. Τιμπίρης Αλκιβιάδης



## Περιεχόμενα ενότητας

---

- Ασφάλεια(Security)
- Ακεραιότητα (integrity)
- Εξέταση προβλημάτων ασφαλείας
- Πολιτική Ασφάλειας
- Μηχανισμοί Ασφάλειας
- Έλεγχος Προσπέλασης
- Επιλεκτικός Έλεγχος Προσπέλασης
- Εντολές GRANT και REVOKE
- Υποχρεωτικός ή Κανονιστικός Έλεγχος Προσπέλασης
- Κλάσεις Ασφαλείας(Security Classes)



# Σκοποί ενότητας

---

Σε αυτήν την ενότητα γίνεται αναφορά στην ασφάλεια και στην προστασία των δεδομένων από την γνωστοποίηση την αλλοίωση ή την καταστροφή τους από μη εξουσιοδοτημένα άτομα. Επίσης παρουσιάζονται πολιτικές και μηχανισμοί ασφαλείας καθώς και η υλοποίηση τους με εντολές SQL δίνοντας δικαιώματα, προνόμια ή ρόλους ασφαλείας στους χρήστες.

# Ασφάλεια(Security)

---

Οι τεχνικές ασφαλείας μας εξασφαλίζουν προστασία από μη εξουσιοδοτημένα άτομα.

Προστασία των δεδομένων από ανεπιθύμητες ενέργειες όπως:

- Διαγραφή
- Μετατροπή
- Υποκλοπή

# Ακεραιότητα (integrity)

---

- Η ακεραιότητα αφορά την διατήρηση της **εγκυρότητας** των δεδομένων
- Οι τεχνικές ακεραιότητας έχουν ως στόχο να **αποτρέψουν** τις ακούσιες αλλαγές στα δεδομένα και να τα κρατήσουν με ακρίβεια όπως καταχωρήθηκαν

## Εξέταση προβλημάτων ασφαλείας

---

- **Νομικά, Κοινωνικά, Ηθικά**
- **Φυσικά:** (κλείδωμα, συναγερμός, βιομετρικός έλεγχος)
- **Πολιτικά:** σε ποιόν και τι επιτρέπεται
- **Λειτουργικά:** (κωδικοί, πόσο συχνά αλλάζουν)
- **Έλεγχοι μέσω υλικού:** (κλειδιά προστασίας μνήμης, προνόμια)
- **Ασφάλεια μέσω λειτουργικού συστήματος:** (άδειες χρήσης, σβήσιμο περιεχομένων μνήμης, προσωρινά αρχεία)

# Ορολογία

---

## Ακεραιότητα(Integrity)

- Οι χρήστες δεν θα έπρεπε να τροποποιούν τα δεδομένα που δεν έχουν εξουσιοδότηση
- **π.χ.**, Ένας υπάλληλος δεν μπορεί να αλλάξει τον μισθό του

## Διαθεσιμότητα(Availability)

- Οι χρήστες θα πρέπει να δουν και να αλλάξουν μόνο πράγματα που τους επιτρέπεται
- **π.χ.**, Ένας υπάλληλος θα μπορεί να αλλάξει την λίστα καταγραφής εμπορευμάτων

## Μυστικότητα(Secrecy)

- Οι χρήστες θα πρέπει να δουν μόνο ότι πρέπει
- **π.χ.**, Ένας υπάλληλος δεν μπορεί να δει τους μισθούς άλλων υπαλλήλων

# Ορολογία

---

## Πολιτική ασφαλείας(Security Policy)

- Μια δήλωση που καθορίζει ποιος έχει τη δικαιοδοσία να κάνει τι

## Μηχανισμός ασφαλείας(Security Mechanism)

- Προστατεύει αποτελεσματικά τα δεδομένα
- Μας επιτρέπει την εφαρμογή μιας συγκεκριμένης πολιτικής

## Πολιτική Ασφάλειας (Security Policy)

---

Τα αποτελέσματα αυτών των αποφάσεων πολιτικής

- Πρέπει να **γνωστοποιηθούν** στο σύστημα (εντολές γραμμένες σε κάποια κατάλληλη γλώσσα ορισμών)
- Το σύστημα πρέπει να τα **θυμάται** (αποθήκευση στον κατάλογο με τη μορφή κανόνων ασφάλειας (security rule), που είναι γνωστοί και ως κανόνες εξουσιοδότησης (authorization)).

## Πολιτική Ασφάλειας (Security Policy)

---

Για τα αποτελέσματα αυτών των αποφάσεων πολιτικής:

- Πρέπει να υπάρχει ένα μέσο για τον έλεγχο μιας δεδομένης αίτησης πρόσβασης με βάση τους ισχύοντες κανόνες ασφάλειας. (Με τον όρο "αίτηση πρόσβασης" εδώ εννοούμε γενικά το συνδυασμό αιτούμενη πράξη συν αιτούμενο αντικείμενο συν αιτών χρήστης.)
- Ο έλεγχος αυτός γίνεται από το υποσύστημα ασφάλειας του DBMS, που είναι γνωστό και ως **υποσύστημα εξουσιοδοτήσεων**



## Πολιτική Ασφάλειας (Security Policy)

---

Το **DBMS** για να μπορεί να αποφασίζει ποιοι κανόνες ασφάλειας ισχύουν για μια δεδομένη αίτηση πρόσβασης:

- Πρέπει να έχει τη δυνατότητα να **αναγνωρίζει** την προέλευση αυτής της αίτησης — δηλαδή, τον αιτούντα χρήστη. Γι' αυτόν το λόγο, όταν οι χρήστες εισέρχονται (sign in) στο σύστημα, συνήθως απαιτείται να δίνουν όχι μόνο την ταυτότητα τους (user ID — για να δηλώσουν ποιοι είναι) αλλά και ένα συνθηματικό (password — για να αποδείξουν ότι είναι αυτοί που ισχυρίζονται ότι είναι). Το συνθηματικό υποτίθεται ότι είναι γνωστό μόνο στο **σύστημα** και στους **νόμιμους χρήστες** που έχουν τη συγκεκριμένη ταυτότητα χρήστη.

## Μηχανισμοί Ασφαλείας ΣΔΒΔ

---

**Επιλεκτικοί μηχανισμοί ασφάλειας ή Περιπτωσιακός έλεγχος ( Discretionary Access Control)**

- Κάθε χρήστης έχει διαφορετικά δικαιώματα πρόσβασης
  - **Προνόμια (Privileges)**
  - **Εξουσίες (Authorities)**

**Υποχρεωτικοί μηχανισμοί ασφαλείας ή Κανονιστικός έλεγχος( Mandatory Access Control - MAC)**

- Επιβολή πολλών επιπέδων - διαχωρισμός των χρηστών σε διάφορα επίπεδα (κλάσεις) ασφαλείας
- Κάθε αντικείμενο χαρακτηρίζεται με επίπεδο βαθμού ασφαλείας (classification) και κάθε χρήστης έχει ορισμένο επίπεδο δικαιοδοσίας (clearance). Μηχανισμοί ελέγχου άκαμπτοι.

## Πολιτική Ασφάλειας (Security Policy)

---

Το **DBMS** για να μπορεί να αποφασίζει ποιοι κανόνες ασφάλειας ισχύουν για μια δεδομένη αίτηση πρόσβασης:

- Πρέπει να έχει τη δυνατότητα να **αναγνωρίζει** την προέλευση αυτής της αίτησης — δηλαδή, τον αιτούντα χρήστη. Γι' αυτόν το λόγο, όταν οι χρήστες εισέρχονται (sign in) στο σύστημα, συνήθως απαιτείται να δίνουν όχι μόνο την ταυτότητα τους (user ID — για να δηλώσουν ποιοι είναι) αλλά και ένα συνθηματικό (password — για να αποδείξουν ότι είναι αυτοί που ισχυρίζονται ότι είναι). Το συνθηματικό υποτίθεται ότι είναι γνωστό μόνο στο **σύστημα** και στους **νόμιμους χρήστες** που έχουν τη συγκεκριμένη ταυτότητα χρήστη.

## Διαχειριστής Συστημάτων

---

Ο διαχειριστής συστημάτων (system administrator) έχει λογαριασμό με αυξημένες αρμοδιότητες όπως:

- **Δημιουργία** λογαριασμού (έλεγχος προσπέλασης)
- **Εκχώρηση** προνομίων (επιλεκτική ασφάλεια)
- **Αφαίρεση** προνομίων (επιλεκτική ασφάλεια)
- **Καθορισμός** επιπέδου ασφαλείας (υποχρεωτική ασφάλεια)
- **Γνωστοποίηση** πολιτικής στο σύστημα με κανόνες ασφάλειας και εξουσιοδότησης (authorization)

## Έλεγχος Προσπέλασης

---

- Ίχνη ελέγχου. (audittrail)
  - Εξακρίβωση αν τα πράγματα είναι υπό έλεγχο
  - Εντοπισμός ενόχου
- Ίχνος ελέγχου → Αρχείο ή Βάση Δεδομένων στην οποία καταγράφονται όλες οι πράξεις που γίνονται από τους χρήστες.
  - Αίτηση
  - Τερματικό
  - Χρήστης
  - Ημν/νία, ώρα
  - Σχέσεις, συστοιχίες, γνωρίσματα που επηρεάσθηκαν
  - Παλιές τιμές
  - Νέες τιμές

## Επιλεκτικός ή Περιπτωσιακός έλεγχος προσπέλασης

---

Βασίζεται στην έννοια των δικαιωμάτων προσπέλασης ή προνομίων πάνω σε αντικείμενα όπως πίνακες(TABLES) ή όψεις(VIEWS)) και μηχανισμούς εκχώρησης και αφαίρεσης προνομίων στους χρήστες.

## Επιλεκτικός Έλεγχος Προσπέλασης

---

### Μοντέλο Πίνακα Προσπέλασης(Access Matrix Model)

- $M(i, j)$  :
  - $i$ : υποκείμενο (π.χ, χρήστης, λογαριασμός, πρόγραμμα)
  - $j$ : αντικείμενο (σχέση, εγγραφή (πλειάδα), στήλη (γνώρισμα), όψη, πράξη)
- Αναπαριστά τον τύπο των προνομίων (εγγραφή, ανάγνωση, τροποποίηση) που έχει το υποκείμενο  $i$  στο αντικείμενο  $j$

## Επιλεκτικός Έλεγχος Προσπέλασης στην SQL

---

### Μοντέλο Πίνακα Προσπέλασης(Access Matrix Model)

- $M(i, j)$  :
  - $i$ : υποκείμενο (π.χ, χρήστης, λογαριασμός, πρόγραμμα)
  - $j$ : Στην SQL μόνο σχέση, όψη στήλη(γνώρισμα)

## Επιλεκτικά επίπεδα δικαιωμάτων βάσει όψεων

---

### Επίπεδο λογαριασμού

- Ορίζεται από αυτούς που υλοποιούν τις βάσεις δεδομένων
- Παραδείγματα:
  - CREATE SCHEMA
  - CREATE TABLE
  - CREATE VIEW
  - ALTER
  - DROP
  - MODIFY SELECT

### Επίπεδο σχέσης

- Ορίζεται ως τμήμα της SQL
- Προσδιορίζουν για κάθε χρήστη τις σχέσεις στις οποίες μπορεί να εφαρμοστεί κάθε τύπος εντολής

## Επιλεκτικός Έλεγχος Προσπέλασης

---

- Ο δημιουργός ενός πίνακα ή μιας όψης παίρνει αυτόματα όλα τα προνόμια σε αυτόν -- ιδιοκτήτης
- Το ΣΔΒΔ διατηρεί πληροφορία σχετικά με το ποιος παίρνει ή χάνει προνόμια και επιτρέπει μόνο αιτήσεις από χρήστες που έχουν τα απαραίτητα προνόμια όταν γίνεται η αίτηση

## Η εντολή GRANT

---

### GRANT Privileges ON Object TO Users [WITH GRANT OPTION]

Επιτρέπονται τα παρακάτω Privileges :

- **SELECT**: Μπορεί να διαβάσει όλες τις στήλες (συμπεριλαμβανομένων αυτών που μπορεί να προστεθούν αργότερα με την εντολή ALTER TABLE).
- **INSERT(column-name)**: Μπορεί να εισάγει πλειάδες με non-NULL ή non-default τιμές στη στήλη column-name.
  - **INSERT** σημαίνει το ίδιο για όλες τις στήλες.
- **DELETE**: Μπορεί να διαγράφει πλειάδες.
- **UPDATE(column-name)**: Μπορεί να ενημερώνει πλειάδες.
- **REFERENCES (column-name)**: Μπορεί να ορίσει ξένα κλειδιά (σε άλλους πίνακες) που αναφέρονται στην column-name.

## Η εντολή GRANT

---

### GRANT Privileges ON Object TO Users [WITH GRANT OPTION]

- Αν ένας χρήστης έχει ένα προνόμιο με το GRANT OPTION μπορεί να δώσει αυτό το προνόμιο σε άλλους χρήστες (μπορεί να δώσει ή να μη δώσει το GRANT OPTION).
- Αν και δεν έχουν ακόμα υλοποιηθεί υπάρχουν μηχανισμοί για τον περιορισμό της διάδοσης προνομίων
- Τα **CREATE**, **ALTER**, και **DROP** μπορεί να εκτελεστούν μόνο από τον ιδιοκτήτη.

## Η εντολή GRANT : Παραδείγματα

---

- Ο χρήστης **Chara** να μπορεί να κάνει ερωτήσεις και να εισάγει πλειάδες στη σχέση **Vathmoi**.
  - GRANT INSERT, SELECT ON **Vathmoi** TO **maria**
- Ο χρήστης **Grammateia** να μπορεί να διαγράφει πλειάδες απο τη σχέση **Student** και μπορεί να εξουσιοδοτήσει και άλλους γιαυτό.
  - GRANT DELETE ON **Student** TO **Grammateia** WITH GRANT OPTION
- Ο χρήστης **Michaela** μπορεί να τροποποιεί μόνο το γνώρισμα **DIEYTHINSI** της σχέσης **Student**.
  - GRANT UPDATE(DIEYTHINSI) ON **Student** TO **elenh**

## Η εντολή REVOKE

---

### REVOKE *Privilege* ON *Object* FROM *Users*

- **Αφαίρεση** προνομίου από έναν χρήστη
- Παρόμοια με την εντολή **GRANT**
- Όταν **αφαιρεθεί** ένα προνόμιο από το χρήστη **X** **αφαιρείται και από όλους** τους χρήστες που πήραν αυτό το προνόμιο αποκλειστικά από αυτόν

## Η εντολή REVOKE : Παραδείγματα

---

- Ο χρήστης **maria** να μην μπορεί να κάνει ερωτήσεις στη σχέση **Misthoi**.  
→ REVOKE SELECT ON **Misthoi** TO **xara**
- Ο χρήστης **elenh** να μην μπορεί να διαγράψει πλειάδες απο τη σχέση **Misthoi**.  
→ REVOKE DELETE ON **Misthoi** TO **elenh**

## Όψεις(VIEWS)

---

- Οι όψεις μπορεί να χρησιμοποιηθούν για να δώσουν **μόνο** την απαραίτητη πληροφορία
- Οι όψεις μαζί με τις εντολές GRANT/REVOKE αποτελούν ένα πολύ ισχυρό εργαλείο για τον έλεγχο προσπέλασης.
- Ο δημιουργός μιας όψης έχει ένα προνόμιο πάνω στην όψη μόνο αν έχει το προνόμιο σε όλες τις σχέσεις που περιλαμβάνονται στον ορισμό της όψης
- Για την δημιουργία της όψης απαιτείται το προνόμιο **SELECT** σε όλες τις σχέσεις που περιλαμβάνονται στον ορισμό της όψης



## Επιλεκτικός Έλεγχος Προσπέλασης

---

Πως μπορείτε να ορίσετε προνόμια στο επίπεδο του ενός γνωρίσματος μιας πλειάδαςQ:

- **SQL-92:** προνόμια σε authorization ids, που μπορεί να αναφέρονται σε ένα χρήστη ή σε μια ομάδα χρηστών
- **SQL-1999:** προνόμια σε ρόλους.
  - Στη συνέχεια μπορεί να ανατεθούν ρόλοι σε χρήστες ή σε άλλους ρόλους

## Υποχρεωτικός ή Κανονιστικός Έλεγχος Προσπέλασης

---

Βασίζονται σε πολιτικές που ισχύουν για όλο το σύστημα και δεν μπορούν να τροποποιηθούν από συγκεκριμένους χρήστες.

- Σε κάθε αντικείμενο της ΒΔ ανατίθεται και μια **κλάση ασφάλειας**.
- Σε κάθε υποκείμενο (χρήστη ή πρόγραμμα) ανατίθεται και ένα δικαίωμα για μια κλάση ασφάλειας
- Κανόνες βασιζόμενοι στις κλάσεις ασφάλειας και στα δικαιώματα καθορίζουν ποιος μπορεί να διαβάσει/γράψει ποια αντικείμενα

Τα περισσότερα εμπορικά συστήματα δεν υποστηρίζουν τον υποχρεωτικό έλεγχο πρόσβασης.

## Υποχρεωτικός ή Κανονιστικός Έλεγχος Προσπέλασης

---

- Χρήστης  $i$  μπορεί να δει το αντικείμενο  $j$  εάν το επίπεδο δικαιοδοσίας του  $i$  είναι μεγαλύτερο ή ίσο του επιπέδου ασφαλείας του  $j$ .
- Χρήστης  $i$  μπορεί να τροποποιήσει το αντικείμενο  $j$  εάν το επίπεδο δικαιοδοσίας του  $i$  είναι μεγαλύτερο ή ίσο του επιπέδου ασφαλείας του  $j$ .
- Κάθε τι που γράφει ο χρήστης  $i$  έχει επίπεδο ασφαλείας ίσο με το επίπεδο δικαιοδοσίας του  $i$

## Υποχρεωτικός Έλεγχος Ασφαλείας Κλάσεις Ασφαλείας (Security Classes)

---

### A -- Βεβαιωμένη προστασία.

- Μαθηματική απόδειξη ότι μηχανισμός ασφαλείας είναι συνεπής και επαρκής

### B -- Δομημένη προστασία

- **B1** προστασία με ετικέτες (απόρρητο - εμπιστευτικό ..)
- **B2** επιπλέον τυπική δήλωση πολιτικής, **B3** λογιστικός έλεγχος και υπεύθυνος διαχείρισης ασφαλείας.

## Υποχρεωτικός Έλεγχος Ασφαλείας Κλάσεις Ασφαλείας (Security Classes)

---

C -- Περιπτώσιακή προστασία

- C1 -- Διαχωρισμός δεδομένων - χρηστών
- C2 -- Διαδικασίες ελέγχου χρήστη, λογιστικού ελέγχου, απομόνωσης πόρων

D -- Λιγότερο ασφαλής

## Υποχρεωτικός Έλεγχος Προσπέλασης

---

- Αντικείμενα (π.χ., πίνακες, όψεις, πλειάδες)
- Υποκείμενα (π.χ., χρήστες, προγράμματα)
- Κλάσεις ασφαλείας:
  - Άκρως Απόρρητη (TS)
  - Απόρρητη (S)
  - Εμπιστευτική (C)
  - Αδιαβάθμητη (U):
  - $TS > S > C > U$

## Υποχρεωτικός Έλεγχος Προσπέλασης

---

- Χρήστες με S και TS μπορούν να δουν και τις δυο γραμμές
- Χρήστες με C βλέπουν μόνο 2<sup>η</sup> γραμμή
- Χρήστες με U δε βλέπουν καμία
  
- Αν ένας χρήστης με C προσπαθήσει να εισάγει <101,Pasta,Blue,C>:
  - Αν επιτρέψουμε την εισαγωγή, παραβίαση του περιορισμού κλειδιού
  - Αν δεν τον επιτρέψουμε αποκαλύπτουμε ότι υπάρχει ένα άλλο αντικείμενο με κλειδί 101 και κλάση > C!
  - Η κλάση μέρος του κλειδιού.
  
- **Πρόβλημα: κανάλι διαρροής (covert channel)**
- **ΛΥΣΗ: Δημιουργία Πολλαπλών Στιγμιότυπων**

## Ενότητα 9

---

### Κρυπτογράφηση δεδομένων

Δρ. Τσιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Ασφάλεια δεδομένων -Αρχές αυθεντικοποίησης
- Ταυτοποίηση
- Αυθεντικοποίηση
- Ηλεκτρονικές υπογραφές
- Ψηφιακές υπογραφές
- Κρυπτογράφηση δεδομένων
- Παραδείγματα κρυπτογράφησης δεδομένων
- Πρότυπο Des
- Κρυπτογράφηση με δημόσιο κλειδί
- Μηχανισμός κρυπτογράφησης RSA
- Παράδειγμα με RSA
- Παράδειγμα κρυπτογράφησης με δημόσιο κλειδί



# Σκοποί ενότητας

---

Σε αυτήν την ενότητα θα γίνει ανάλυση της κρυπτογράφησης των δεδομένων η οποία ορίζεται ως η αποθήκευση ή η μετάδοση των εμπιστευτικών δεδομένων σε κωδικοποιημένη μορφή. Τέλος παρουσιάζονται οι μηχανισμοί κρυπτογράφησης, υποκατάστασης μετάθεσης, δημόσιου κλειδιού και γίνεται μια εξοικείωση με τις ψηφιακές υπογραφές.

## Ασφάλεια δεδομένων -Αρχές αυθεντικοποίησης

---

Στις ηλεκτρονικές συναλλαγές πρώτιστο ρόλο παίζει η προστασία των προσωπικών δεδομένων και της ασφάλειας. Για επιτυχημένες λύσεις ΗΔ απαιτείται να υπάρχει η **αυθεντικοποίηση** των συναλλασσόμενων μερών εξασφαλίζοντας:

- Εμπιστευτικότητα
- Ακεραιότητα
- Ταυτοποίηση
- Αυθεντικότητα
- Μη αποποίηση ευθύνης



## Ασφάλεια δεδομένων-Ταυτοποίηση

---

Ως **ταυτοποίηση** ορίζεται ο προσδιορισμός των πληροφοριών/δεδομένων που μπορούν να ταυτοποιήσουν μια οντότητα (πολίτες/ επιχειρήσεις). Στο πλαίσιο αυτό εντάσσονται οι εξής παράμετροι:

- Η προστασία της ιδιωτικότητας και η προστασία των προσωπικών δεδομένων εφαρμόζοντας το Ν. 2742/1997 και την Κοινοτική Οδηγία 95/46/ΕΚ
- Η εγκυρότητα μητρώων που αφορά τις ιδιότητες με τις οποίες ο πολίτης συναλλάσσεται με τη δημόσια διοίκηση
- Ο μοναδικός αριθμός ταυτοποίησης

## Ασφάλεια δεδομένων- Αυθεντικοποίηση

---

Ως **αυθεντικοποίηση** ορίζεται η δυνατότητα παροχής αδιαμφισβήτητων στοιχείων για την επαλήθευση της ταυτότητας μιας οντότητας. Συγκεκριμένα αφορά:

- Έλεγχος αυθεντικότητας κι επίπεδα εμπιστοσύνης, δηλαδή ότι ο χρήστης είναι αυτός που ισχυρίζεται ότι είναι
- Αναγνωριστικά & ψηφιακά πιστοποιητικά, δηλαδή η ηλεκτρονική επιβεβαίωση της ταυτότητας μιας οντότητας

## Ηλεκτρονικές υπογραφές

---

Έρχονται να καλύψουν το θέμα της ασφάλειας των συναλλαγών επιτρέποντας στον λήπτη των δεδομένων να διαπιστώνει την προέλευσή τους (αυθεντικότητα), να ελέγχει κατά πόσο τα πρωτότυπα δεδομένα είναι πλήρη και αναλώσιμα (ακεραιότητα), αλλά και μέσω των «παρεχόντων υπηρεσιών πιστοποίησης» να είναι βέβαιος για την ταυτότητα του υπογράφοντος (ταυτοποίηση). Χωρίζονται σε 2 κατηγορίες:

- τις **ψηφιακές υπογραφές** που χρησιμοποιούν την κρυπτογραφία με τη χρήση αλγοριθμικών κλειδιών
- τις **ηλεκτρονικές υπογραφές** που υποστηρίζονται από μεθόδους όπως η βιομετρική, η αναγνώριση φωνής κ.ά.

## Ψηφιακές υπογραφές

---

Υποστηρίζονται από την ασύμμετρη κρυπτογραφία και πρέπει να ικανοποιούν τις ακόλουθες απαιτήσεις για να θεωρούνται αξιόπιστες και έγκυρες:

- Να συνδέεται μονοσήμαντα με τον υπογράφοντα
- Να ταυτοποιεί τον υπογράφοντα
- Να δημιουργείται με μέσα τα οποία ο υπογράφων διατηρεί υπό αποκλειστικό του έλεγχο
- Να εντοπίζεται οποιαδήποτε αλλοίωση των δεδομένων με τα οποία συνδέεται



## Κρυπτογράφηση δεδομένων

---

Η διαδικασία της **Κρυπτογράφησης δεδομένων(Data encryption)** είναι η εξής:

- Απλό κείμενο → Κρυπτογράφηση κειμένου (αλγόριθμος κρυπτογράφησης + κλειδί κρυπτογράφησης)
- Κρυπτογράφηση κειμένου → Κρυπτογραφικό κείμενο
  
- Κρυπτογράφηση του ΚΑΙΣΑΡΑ

**Σχόλιο!** Οι λεπτομέρειες του αλγορίθμου γνωστοποιούνται το κλειδί μυστικό



## Κρυπτογράφηση δεδομένων

---

Η κρυπτογράφηση αποτελεί μια μέθοδο διασφάλισης των ηλεκτρονικών συναλλαγών, προστασίας των διακινούμενων δεδομένων και των οντοτήτων που συναλλάσσονται.

Η **κρυπτογράφηση δημόσιου κλειδιού** εξασφαλίζει την ιδιωτικότητα, την ακεραιότητα και την εμπιστευτικότητα. Υπάρχουν 2 τεχνικές κρυπτογράφησης η **συμμετρική κρυπτογραφία** (κοινό κλειδί για κρυπτογράφηση κι αποκρυπτογράφηση) και η **ασύμμετρη κρυπτογραφία** (δημόσιο κλειδί για την κρυπτογράφηση και ιδιωτικό κλειδί για την αποκρυπτογράφηση)

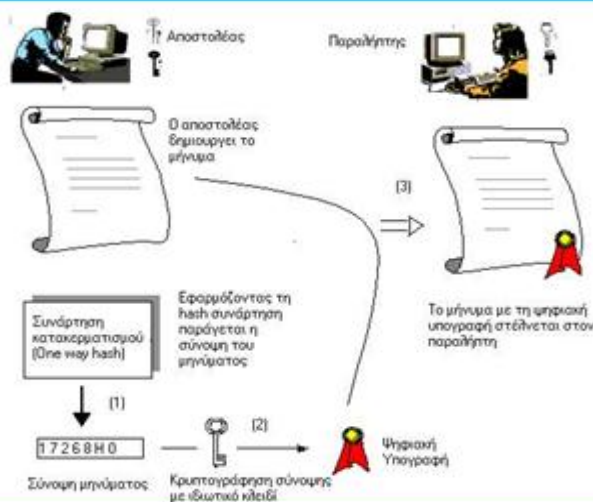
## Υποδομή Δημόσιου Κλειδιού (Public Key Infrastructure – PKI)

Είναι ο μηχανισμός επαλήθευσης των ταυτοτήτων των προσώπων που διαθέτουν δημόσιο κλειδί. Πρόκειται για ένα συνδυασμό λογισμικού, τεχνολογιών κρυπτογραφίας και υπηρεσιών που πιστοποιεί την εγκυρότητα του κάθε φυσικού προσώπου που εμπλέκεται σε μια συναλλαγή στο Διαδίκτυο και παράλληλα προστατεύει την ασφάλεια της συναλλαγής.

Ο οργανισμός που εκδίδει πιστοποιητικά ονομάζεται Πάροχος Υπηρεσιών Πιστοποίησης (ΠΥΠ) ή Αρχή Πιστοποίησης.

Στην Ελλάδα οι ΠΥΠ ελέγχονται από την Εθνική Επιτροπή Τηλεπικοινωνιών & Ταχυδρομείων (Ε.Ε.Τ.Τ)

## Χρήση κρυπτογράφησης και ψηφιακής υπογραφής



## Παράδειγμα κρυπτογράφησης δεδομένων

PlainText = ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ

Κλειδί = Test

Για την κρυπτογράφηση του κειμένου η διαδικασία είναι η εξής:

- Χωρίστε το σε ίσα μέρη σύμφωνα με το κλειδί (ΤΕΙ+ ΚΕΝΤ ΡΙΚΗ Σ+ΜΑΚ ΕΔΟΝ ΙΑΣ+)
- Αντικαταστήστε τους χαρακτήρες με αριθμούς από το 00-26 (κενό=00 Α=01....).
- Αποτέλεσμα

**Σχόλιο!** Στα κενά τοποθετείται +

A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9
J	10
K	11
L	12
M	13
N	14
O	15
P	16
Q	17
R	18
S	19
T	20
U	21
V	22
W	23
X	24
Y	25
Z	26



## Παράδειγμα κρυπτογράφησης δεδομένων

- Άθροιση των προηγούμενων αποτελεσμάτων και διαίρεση τους με το 27
- Κρατήστε το υπόλοιπο της διαίρεσης
- Αντικατάστασή κάθε αριθμού του υπολοίπου με τον χαρακτήρα που του αντιστοιχεί



## Παράδειγμα2 κρυπτογράφησης δεδομένων

PlainText = ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

Κλειδί= Test

Για την κρυπτογράφηση του κειμένου η διαδικασία είναι η εξής:

- Χωρίστε το σε ίσα μέρη σύμφωνα με το κλειδί (ΤΜΗΜΑ+ΜΗΧΑΝΙΚΩΝ+ΠΛΗΡΟΦΟΡΙΚΗΣ+ΤΕ+)
- Αντικαταστήστε τους χαρακτήρες με αριθμούς από το 00-26 (κενό=00 A=01....).
- Αποτέλεσμα 201308010013082401140911.....

**Σχόλιο!** Στα κενά τοποθετείται +

A	1
B	2
C	3
D	4
E	5
F	6
G	7
H	8
I	9
J	10
K	11
L	12
M	13
N	14
O	15
P	16
Q	17
R	18
S	19
T	20
U	21
V	22
W	23
X	24
Y	25
Z	26



## Πρότυπο Des

Πρότυπο DES:

- **Υποκατάσταση** (SUBSTITUTION) :χρησιμοποιεί ένα κλειδί κρυπτογράφησης έτσι ώστε να προσδιοριστεί για τον κάθε χαρακτήρα απλού κειμένου ένας χαρακτήρας κρυπτογραφικού κειμένου με τον οποίο θα υποκατασταθεί αυτός ο χαρακτήρας
- **Μετάθεση** (PERMUTATION) οι χαρακτήρες απλού κειμένου απλώς αναδιατάσσονται με κάποια διαφορετική σειρά.

Καμία από τις δύο αυτές προσεγγίσεις δεν είναι ιδιαίτερα ασφαλής. Ο συνδυασμός όμως αυτών των δύο παρέχουν έναν πολύ υψηλό βαθμό ασφάλειας. Ο συνδυασμός αυτών των δύο δημιούργησε το πρότυπο DES (Data Encryption Standard — Πρότυπο Κρυπτογράφησης Δεδομένων), που υιοθετήθηκε για πρώτη φορά ως ομοσπονδιακό πρότυπο των ΗΠΑ το 1977.



# Πρότυπο Des

---

Χαρακτηριστικά του προτύπου DES:

- Το απλό κείμενο υποδιαιρείται σε τμήματα των 64 bit και το κάθε τμήμα κρυπτογραφείται με ένα κλειδί των 64 bit
- Το κλειδί αποτελείται από 56 bit δεδομένων + 8 bit ισοτιμίας, και επομένως δεν υπάρχουν  $2^{64}$  αλλά μόνο  $2^{56}$  πιθανά κλειδιά).
- Για την κρυπτογράφηση ενός τμήματος, πρώτα εκτελείται μια αρχική μετάθεση πάνω σε αυτό, έπειτα το τμήμα που προκύπτει από τη μετάθεση υποβάλλεται σε μια ακολουθία 16 βημάτων σύνθετης υποκατάστασης, και τέλος εφαρμόζεται άλλη μία μετάθεση, η αντίστροφη της αρχικής στο αποτέλεσμα του προηγούμενου βήματος.



# Πρότυπο Des

---

- Η υποκατάσταση στο βήμα  $i$  δεν ελέγχεται άμεσα από το αρχικό κλειδί κρυπτογράφησης  $K$ , αλλά από ένα κλειδί  $K_i$  που υπολογίζεται από τις τιμές  $K$  και  $i$ .
- Το DES έχει την ιδιότητα-χαρακτηριστικό ότι ο αλγόριθμος αποκρυπτογράφησης είναι ο ίδιος με τον αλγόριθμο κρυπτογράφησης, με μία διαφορά όμως τα κλειδιά  $K_i$  εφαρμόζονται με αντίστροφη σειρά.



## Κρυπτογράφηση με δημόσιο κλειδί

---

- Σε ένα μηχανισμό με δημόσιο κλειδί (public-key), τόσο ο αλγόριθμος κρυπτογράφησης όσο και 'το κλειδί κρυπτογράφησης είναι διαθέσιμα σε όλους ώστε να μπορεί ο καθένας να μετατρέψει κάποιο απλό κείμενο σε κρυπτογραφικό κείμενο.
- Το αντίστοιχο κλειδί αποκρυπτογράφησης διατηρείται μυστικό (οι μηχανισμοί με δημόσιο κλειδί έχουν δύο κλειδιά, ένα για την κρυπτογράφηση και ένα για την αποκρυπτογράφηση).
- Το κλειδί αποκρυπτογράφησης δεν είναι δυνατό να προκύψει από το κλειδί κρυπτογράφησης έτσι, ακόμα και το άτομο που κάνει την αρχική κρυπτογράφηση δεν μπορεί να κάνει την αντίστοιχη αποκρυπτογράφηση αν δεν έχει την εξουσιοδότηση να το κάνει.

## Μηχανισμός κρυπτογράφησης RSA

---

Ο μηχανισμός κρυπτογράφησης με δημόσιο κλειδί RSA βασίζεται στα εξής δύο γεγονότα:

- Υπάρχει ένας γνωστός γρήγορος αλγόριθμος με τον οποίο μπορεί να προσδιοριστεί αν ένας δεδομένος αριθμός είναι πρώτος αριθμός.
- Δεν υπάρχει κανένας γνωστός γρήγορος αλγόριθμος για την εύρεση των πρώτων παραγόντων ενός δεδομένου παραγώγου (δηλαδή, όχι πρώτου) αριθμού.

**Σχόλιο!** Το όνομα RSA του μηχανισμού κρυπτογράφησης προέκυψε από τα αρχικά του ονόματος των εμπνευστών του (Rivest, Shamir, Adleman).

# Μηχανισμός κρυπτογράφησης RSA

Η λειτουργία του μηχανισμού κρυπτογράφησης RSA έχει ως εξής:

- Επιλέγονται τυχαία δύο διαφορετικοί μεγάλοι πρώτοι αριθμοί,  $p$  και  $q$ , και υπολογίζεται το γινόμενο  $r = p * q$ .
- Επιλέγεται τυχαία ένας μεγάλος ακέραιος  $e$  που είναι σχετικά πρώτος (relatively prime) ως προς το γινόμενο  $(p - 1) * (q - 1)$ . Ο ακέραιος  $e$  είναι το κλειδί της κρυπτογράφησης. Έστω ότι παίρνετε ως κλειδί αποκρυπτογράφησης  $d$  το μοναδικό "πολλαπλασιαστικό αντίστροφο" του ακεραίου υπολοίπου της διαίρεσης του  $e$  με το  $(p - 1) * (q - 1)$ , δηλαδή:
  - $d * e = 1 \text{ modulo } (p - 1) * (q - 1)$
  - Ο αλγόριθμος για τον υπολογισμό του  $d$  με δεδομένα τα  $e$ ,  $p$ , και  $q$  είναι απλός
  - Γνωστοποιούνται οι ακέραιοι  $r$  και  $e$ , όχι όμως ο  $d$ .

**Σχόλιο!** Η επιλογή του  $e$  είναι εύκολη. Οποιοσδήποτε πρώτος αριθμός, μεγαλύτερος και από τον  $p$  και από τον  $q$ , είναι κατάλληλος.



# Μηχανισμός κρυπτογράφησης RSA

Η λειτουργία του μηχανισμού κρυπτογράφησης RSA έχει ως εξής (Συνέχεια):

- Για να κρυπτογραφηθεί ένα απόσπασμα απλού κειμένου  $P$  (που θεωρούμε για λόγους απλότητας ότι είναι ένας ακέραιος μικρότερος από τον  $r$ ), αντικαθίσταται από το κρυπτογραφικό κείμενο  $C$  που υπολογίζεται με τον εξής τρόπο:  $C = P^e \text{ modulo } r$
- Για να αποκρυπτογραφηθεί ένα απόσπασμα κρυπτογραφικού κειμένου  $C$ , αντικαθίσταται από το απλό κείμενο  $P$  που υπολογίζεται με τον εξής τρόπο:  $P = C^d \text{ modulo } r$

Η αποκρυπτογράφηση του  $C$  με χρήση του  $d$  **αποκαθιστά** το αρχικό  $P$ . Όμως, ο υπολογισμός του  $d$  με γνωστά μόνο τα  $r$  και  $e$  (και όχι τα  $p$  και  $q$ ) είναι **ανέφικτος**.



## Παράδειγμα με RSA

---

Έστω  $p=3$  και  $q=5$ , τότε  $r=15$ , και το γινόμενο  $(p-1) * (q-1) =8$  και το  $e=11$ . Επίσης ένας πρώτος αριθμός μεγαλύτερος και από το  $p$  και από το  $q$ . Για να υπολογιστεί το  $d$  κάνετε τα εξής:

- $d * 11 = 1 \text{ modulo } 8$  από το οποίο προκύπτει  $d=3$ .

Έστω τώρα ότι το απλό κείμενο  $P$  αποτελείται από τον ακέραιο 13. Τότε, το κρυπτογραφικό κείμενο  $C$  προκύπτει από τις πράξεις:

- $C = P^e \text{ modulo } r = 13^{11} \text{ modulo } 15 = 1.792.160.394.037 \text{ modulo } 15 = 7$



## Παράδειγμα με RSA

---

Το αρχικό απλό κείμενο  $P$  προκύπτει από τις πράξεις :

$$\begin{aligned} P &= C^d \text{ modulo } r \\ &= 7^3 \text{ modulo } 15 \\ &= 343 \text{ modulo } 15 \\ &= 13 \end{aligned}$$





## Παράδειγμα κρυπτογράφησης με δημόσιο κλειδί

**Μηχανισμοί κρυπτογράφησης με Δημόσιο Κλειδί** :Επιτρέπουν τα κρυπτογραφημένα μηνύματα να είναι "υπογεγραμμένα", ώστε ο παραλήπτης να μπορεί να είναι βέβαιος ότι το μήνυμα προέρχεται από το άτομο που υποτίθεται ότι προέρχεται (δηλαδή, οι "υπογραφές" δεν μπορούν να πλαστογραφηθούν).

### Παράδειγμα

- Έστω ότι οι αλγόριθμοι κρυπτογράφησης είναι οι ECA και ECB (για την κρυπτογράφηση των μηνυμάτων που θα στέλνονται στον A και στον B, αντίστοιχα).
- Έστω ότι οι αντίστοιχοι αλγόριθμοι αποκρυπτογράφησης είναι οι DCA και DCB, αντίστοιχα.
- Οι αλγόριθμοι ECA και DCA είναι αντίστροφοι μεταξύ τους, όπως και οι ECB και DCB.



## Παράδειγμα κρυπτογράφησης με δημόσιο κλειδί

### Παράδειγμα συνέχεια

- ο A εφαρμόζει πρώτα τον αλγόριθμο αποκρυπτογράφησης DCA στο P, και στη συνέχεια κρυπτογραφεί το αποτέλεσμα και το μεταδίδει ως κρυπτογραφικό κείμενο : $C = ECB ( DCA ( P ) )$ .
- Μόλις πάρει το C, ο χρήστης B εφαρμόζει τον αλγόριθμο αποκρυπτογράφησης DCB και στη συνέχεια τον αλγόριθμο κρυπτογράφησης ECA, ώστε να προκύψει το τελικό αποτέλεσμα P:  
$$ECA ( DCB ( C ) )$$
$$= ECA ( DCB ( ECB ( DCA ( P ) ) ) )$$
$$= ECA ( DCA ( P ) ) \quad \text{επειδή τα DCB και ECB αναιρούνται.}$$
$$= P \quad \text{επειδή τα ECA και DCA αναιρούνται.}$$



## Παράδειγμα κρυπτογράφησης με δημόσιο κλειδί

---

Μετά την διαδικασία αυτή ο Β ξέρει ότι το μήνυμα προέρχεται από τον Α, επειδή ο αλγόριθμος ECA θα δώσει το Ρ μόνο αν χρησιμοποιήθηκε ο αλγόριθμος DCA στη διαδικασία κρυπτογράφησης και αυτός ο αλγόριθμος είναι γνωστός μόνο στον Α. Κανένας, ούτε ακόμα και ο Β δεν μπορεί να πλαστογραφήσει την υπογραφή του Α.

## Ενότητα 10

---

### Μεθοδολογία συσχέτισης ΒΔ με εφαρμογές οπτικού προγραμματισμού

Δρ. Τιμπίρης Αλκιβιάδης



## Περιεχόμενα ενότητας

---

- Σύνδεση βάσης δεδομένων με Visual Studio
- Σύνδεση βάσης δεδομένων με ODBC Drivers



# Σκοποί ενότητας

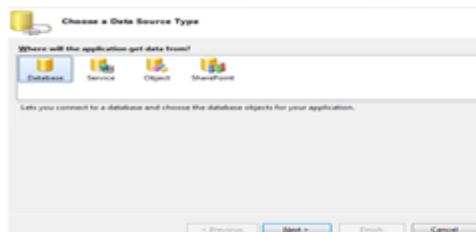
---

Στην ενότητα αυτή παρουσιάζεται ένα παράδειγμα ανάπτυξης μίας σχεσιακής βάσης και η σύνδεσή της μέσω ODBC drivers με εφαρμογή που θα αναπτυχθεί με οπτικό προγραμματισμό.

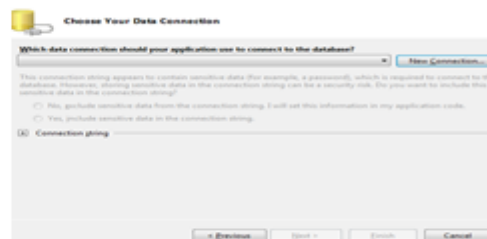


## Σύνδεση βάσης δεδομένων με Visual Studio

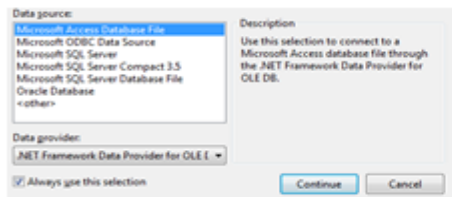
---



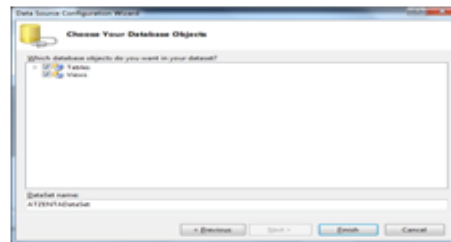
Βήμα 1



## Σύνδεση βάσης δεδομένων με Visual Studio



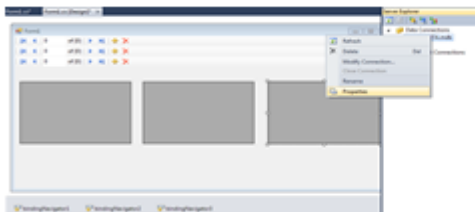
Βήμα 2



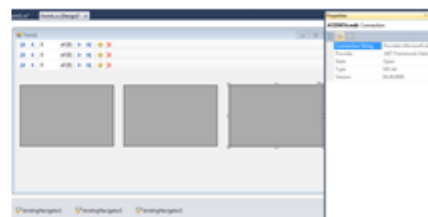
Τμήμα Μηχανικών Πληροφορικής ΤΕ

5

## Σύνδεση βάσης δεδομένων με Visual Studio



Βήμα 3

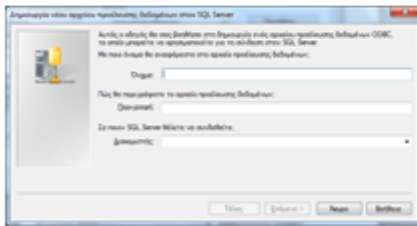


Τμήμα Μηχανικών Πληροφορικής ΤΕ

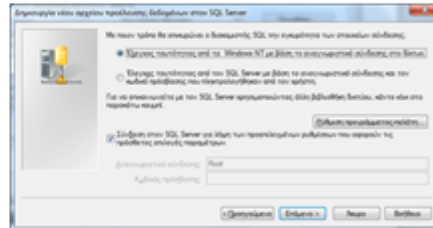
6



# Σύνδεση βάσης δεδομένων με ODBC Drivers



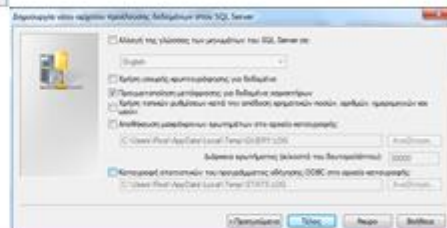
Βήμα 2



# Σύνδεση βάσης δεδομένων με ODBC Drivers

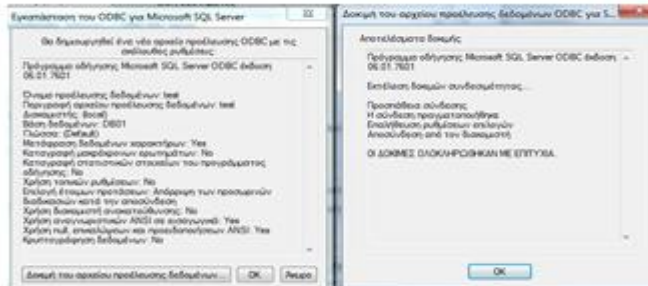


Βήμα 3



# Σύνδεση βάσης δεδομένων με ODBC Drivers

## Βήμα 4





## Ενότητα 11

---

### Αντικειμενοστραφής και αντικειμενο-σχεσιακές βάσεις δεδομένων

Δρ. Τιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Μοντέλα κατασκευής ΣΔΒΔ
- Αντικειμενοστραφής και αντικειμενο-σχεσιακές ΒΔ
- Δυαδικά Μεγάλα Αντικείμενα (Binary Large Objects BLOBs)
- Προσθήκη αντικειμενοστραφών χαρακτηριστικών στην SQL
- Reference types and deref
- Μέθοδοι ορισμένοι από τους χρήστες
- Κληρονομικότητα
- Επεκτάσεις



# Σκοποί ενότητας

---

Στην ενότητα αυτή θα γίνει ενασχόληση με τα συστήματα διαχείρισης βάσης δεδομένων βασισμένα στο αντικειμενοστραφές μοντέλο. Εισαγωγή στα αντικείμενο-σχεσιακά μοντέλα και αντικειμενοστραφή μοντέλα. Τέλος δυαδικά μεγάλα αντικείμενα τα οποία μπορούν να αποθηκευτούν και να προσπελαστούν και προσθήκη χαρακτηριστικών στο σύστημα τύπων της SQL.

## Μοντέλα κατασκευής συστημάτων διαχείρισης ΒΔ

---

Υπάρχουν δύο τάσεις κατασκευής συστημάτων διαχείρισης βάσεων δεδομένων βασισμένα στο αντικειμενοστραφές μοντέλο τα:

- **αντικειμενο-σχεσιακά** μοντέλα (επέκταση σχεσιακού μοντέλου)
- **αντικειμενο-στραφή** μοντέλα (επέκταση αντικειμενοστραφών γλωσσών προγραμματισμού)

## Εισαγωγή στις αντικειμενοστραφής και αντικειμενο-σχεσιακές βάσεις

---

Το πρώτο απλό αντικειμενο-σχεσιακό μοντέλο εμφανίστηκε στη δεκαετία του 70 :

- Ήταν κατάλληλο για δεδομένα διαχείρισης
- Δεν ήταν κατάλληλο για άλλα είδη όπως πολυμέσων ,CAD και άλλων.

Την δεκαετία του 80 εμφανίστηκαν και τα αντικειμενο-στραφή μοντέλα:

- πολύπλοκοι τύποι δεδομένων
- ταυτότητα αντικειμένου (object identity)/ αναφορές (references)
- ADTs (encapsulation, behavior goes with data)
- κληρονομικότητα (inheritance)



## Παράδειγμα ενός μη σχεσιακού μοντέλου ΒΔ

---

### Παράδειγμα

```
create table foithtes (AEM integer, image BLOB,  
                    category integer)
```

### Τα BLOBs είναι:

- Δυαδικά Μεγάλα Αντικείμενα (Binary Large Objects BLOBs) μπορούν να αποθηκευτούν και να προσπελαστούν
- Κώδικας χρήστη για την επεξεργασία των BLOBs

**Σχόλιο!** Για την επεξεργασία των BLOBs ο χρήστης πρέπει να διαθέτει κώδικα χρήσης του.



## Προσθήκη αντικειμενοστραφών χαρακτηριστικών στην SQL

---

- στήλες (γνωρίσματα) ως νέοι τύποι δεδομένων (ADTs)
- μέθοδοι ορισμένοι από τους χρήστες για τους τύπους δεδομένων
- οι στήλες μπορεί να ανήκουν σε σύνθετους τύπους
- τύποι αναφοράς (reference and "deref")

## Τύποι ορισμένοι από τους χρήστες (ADTS)

---

### ADTS

- Μπορεί οι τύποι των δεδομένων στην SQL να είναι περιορισμένοι οι μέθοδοι όμως είναι πολλοί απλοί.
- Στα αντικειμενο-σχεσιακά συστήματα βάσεων δεδομένων επιτρέπεται ο ορισμός νέων τύπων και μεθόδων.

## Reference types and deref

---

- Κάθε αντικείμενο στις σχεσιακές βάσεις έχει ένα OID
- Στις σχεσιακές βάσεις μπορείτε να έχετε έναν τύπο αναφοράς  
→ `ref(lesson_n)`
- Υπάρχει διαφορά μεταξύ σύνθετων τύπων και της αναφοράς παρόλο που το αποτέλεσμα τους μπορεί να φαίνεται το ίδιο.

## SQL-99

---

- Η SQL χρησιμοποιεί built-in πράξεις για σύνθετους τύπους δεδομένων όπως για σύνολα sets και για συμβολισμού τύπων πλειάδων
- Υποστηρίζει τελεστές για τύπους αναφοράς
- Υποστηρίζει μεθόδους που η σύνταξή τους ορίζεται από τους χρήστες.
- Η σύνταξή της εξαρτάται από το προϊόν

## Μέθοδοι ορισμένοι από τους χρήστες

---

- Για τον χειρισμό των ADTS χρειάζεται η εισαγωγή μεθόδου ορισμού ( `jpeg: thumbnail, crop, rotate, smooth`)
- Τις μεθόδους τις εισάγει σε κάποια γλώσσα και τις μεταγλωττίζει το σύστημα
- Το σύστημα δυναμικά κάνει έναν σύνδεσμο με συναρτήσεις για τον διαχειριστή

## Κληρονομικότητα

---

- Είναι χρήσιμο να δημιουργήσετε εξειδικευμένους τύπους όπως και στην C++:  
→ `create type lessons_dioikisis_t under lessons (menu text);`

**Σχόλιο!** Οι μέθοδοι του `lessons_t` ισχύουν επίσης και στους υπό-τύπους. Το `lessons_dioikisis_t` είναι ο κύριος τύπος.

- Κληρονομικότητα σε πίνακες:

→ `create table foitites_aem under aem (gpa int);`  
οι ερωτήσεις στο `aem` επιστρέφουν πλειάδες και της σχέσης `foitites_aem` (εκτός αν προσδιοριστεί "aem only")

## Επεκτάσεις

---

Όλα τα αντικείμενα ενός συγκεκριμένου τύπου μπορούν να επιλεγούν από μία μόνο όψη (π.χ. `select * from lesson_t`).

## Ενότητα 12

---

### Ευρετήρια

Δρ. Τσιμπήρης Αλκιβιάδης



## Περιεχόμενα ενότητας

---

- Ευρετήρια
- Πρωτεύον Ευρετήριο
- Ευρετήριο Συστάδων
- Ευρετήρια: Επανάληψη
- Δευτερεύον Ευρετήριο
- Data blocks
- Τεχνικά ζητήματα στα ευρετήρια
- Βελτίωση ταχύτητας εύρεσης
- Αναγκαιότητα χρήσης ευρετηρίων
- Αποφυγή δημιουργίας ευρετηρίων
- Τρόπος δημιουργίας ευρετηρίων





# Σκοποί ενότητας

---

Στην ενότητα αυτή δίνεται μεγάλη σημασία στο να κατανοηθούν τα είδη των ευρετηρίων, τα δέντρα αναζήτησης, τα Β-δέντρα και τα ευρετήρια κατακερματισμού.

## Ευρετήρια

---

Η δομή προσπέλασης (access path) συνήθως ορίζεται σε ένα πεδίο του αρχείου που λέγεται **πεδίο ευρετηριοποίησης** (indexing field)

- **Πρωτεύον ευρετήριο** (primary index): ορισμένο στο κλειδί διάταξης του αρχείου.
- **Ευρετήριο συστάδων** (clustering index): ορισμένο στο πεδίο διάταξης το οποίο όμως δεν είναι κλειδί.
- **Δευτερεύον ευρετήριο** (secondary index): ορισμένο σε πεδία διαφορετικά του κλειδιού διάταξης.

*Πόσα ευρετήρια σε ένα αρχείο δεδομένων μπορεί να έχουμε;*

# Πρωτεύον Ευρετήριο

---

**Πρωτεύον ευρετήριο (primary index):** ορισμένο στο κλειδί διάταξης του αρχείου.

**Για κάθε block του αρχείου** (μη πυκνό ευρετήριο) η εγγραφή  $i$  του ευρετηρίου είναι της μορφής  $\langle K(i), P(i) \rangle$  όπου:

- **$K(i)$ :** η τιμή του πρωτεύοντος κλειδιού της πρώτης εγγραφής του block (άγκυρα του block).
- **$P(i)$ :** δείκτης προς το block.

# Πρωτεύον Ευρετήριο

---

- Το ευρετήριο αρχείου είναι ένα *διατεταγμένο* αρχείο με σταθερού μήκους εγγραφές
- Το πρωτεύον ευρετήριο είναι ένα *μη πυκνό* ευρετήριο
- Το μέγεθος του αρχείου ευρετηρίου είναι μικρότερο από του αρχείου δεδομένων.

# Πρωτεύον Ευρετήριο

---

## Παράδειγμα:

Έστω διατεταγμένο αρχείο με  $r = 30.000$  εγγραφές, μέγεθος block  $B = 1024$  bytes, σταθερού μεγέθους εγγραφές μεγέθους  $R = 100$  bytes, όπου το πεδίο κλειδιού διάταξης έχει μέγεθος  $V = 9$  bytes, μη εκτεινόμενη καταχώρηση. Κατασκευάζουμε πρωτεύον ευρετήριο, μέγεθος δείκτη block  $P = 6$  bytes.

Μέγεθος αρχείου δεδομένων: 3.000 blocks

Μέγεθος αρχείου ευρετηρίου: 45 blocks



# Πρωτεύον Ευρετήριο

---

## ΑΝΑΖΗΤΗΣ

- Διαδική αναζήτηση στο πρωτεύον ευρετήριο
- Ανάγνωση του block από το αρχείο δεδομένων



# Πρωτεύον Ευρετήριο

---

## Παράδειγμα:

Εστω διατεταγμένο αρχείο με  $r = 30.000$  εγγραφές, μέγεθος block  $B = 1024$  bytes, σταθερού μεγέθους εγγραφές μεγέθους  $R = 100$  bytes, όπου το πεδίο κλειδιού διάταξης έχει μέγεθος  $V = 9$  bytes, μη εκτεινόμενη καταχώρηση. Κατασκευάζουμε πρωτεύον ευρετήριο, μέγεθος δείκτη block  $P = 6$  bytes

Μέγεθος αρχείου δεδομένων: 3.000 blocks - Μέγεθος αρχείου ευρετηρίου: 45 blocks

Αναζήτηση χωρίς ευρετήριο:  $\lceil \log 3.000 \rceil = 12$  blocks

Αναζήτηση με ευρετήριο:  $\lceil \log 45 \rceil + 1 = 7$  blocks



# Πρωτεύον Ευρετήριο

---

## • Εισαγωγή εγγραφής

1. αλλαγές και στο πρωτεύον ευρετήριο
2. μη διατεταγμένο αρχείο υπερχείλισης
3. συνδεδεμένη λίστα εγγραφών υπερχείλισης

## • Διαγραφή εγγραφής

1. αλλαγές και στο πρωτεύον ευρετήριο
2. χρήση σημαδιών διαγραφής



## Ευρετήριο Συστάδων

---

Ευρετήριο συστάδων (clustering index): ορισμένο στο πεδίο διάταξης το οποίο όμως **δεν** είναι κλειδί.

Το μέγεθος του αρχείου ευρετηρίου είναι μικρότερο από του αρχείου δεδομένων.

Υπάρχει μια εγγραφή για κάθε διακεκριμένη τιμή του πεδίου διάταξης (συστάδας) του αρχείου που περιέχει:

- την τιμή αυτή.
- ένα δείκτη προς το πρώτο block του αρχείου δεδομένων που περιέχει μια εγγραφή με την τιμή αυτή στο πεδίο συστάδας.



## Ευρετήριο Συστάδων

---

### Παράδειγμα:

Έστω διατεταγμένο αρχείο με  $r = 30.000$  εγγραφές, μέγεθος block  $B = 1024$  bytes, σταθερού μεγέθους εγγραφές μεγέθους  $R = 100$  bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο διάταξης έχει μέγεθος  $V = 9$  bytes και υπάρχουν 1000 διαφορετικές τιμές και οι εγγραφές είναι ομοιόμορφα κατανεμημένες ως προς τις τιμές αυτές. Υποθέτουμε ότι χρησιμοποιούνται άγκυρες block, κάθε νέα τιμή του πεδίου διάταξης αρχίζει στην αρχή ενός νέου block. Κατασκευάζουμε ευρετήριο συστάδων, μέγεθος δείκτη block  $P = 6$  bytes.

Μέγεθος αρχείου δεδομένων: 3.000 blocks

Μέγεθος ευρετηρίου συστάδων: 15 blocks



# Ευρετήριο Συστάδων

---

## ΑΝΑΖΗΤΗΣ

- Διαδική αναζήτηση στο ευρετήριο
- Ανάγνωση του block από το αρχείο δεδομένων



# Ευρετήριο Συστάδων

---

## Παράδειγμα:

Έστω διατεταγμένο αρχείο με  $r = 30.000$  εγγραφές, μέγεθος block  $B = 1024$  bytes, σταθερού μεγέθους εγγραφές μεγέθους  $R = 100$  bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο διάταξης έχει μέγεθος  $V = 9$  bytes και υπάρχουν 1000 διαφορετικές τιμές και οι εγγραφές είναι ομοιόμορφα κατανεμημένες ως προς τις τιμές αυτές. Υποθέτουμε ότι χρησιμοποιούνται άγκυρες block, κάθε νέα τιμή του πεδίου διάταξης αρχίζει στην αρχή ενός νέου block. Κατασκευάζουμε ευρετήριο συστάδων, μέγεθος δείκτη block  $P = 6$  bytes.

Μέγεθος αρχείου δεδομένων: 3.000 blocks - Μέγεθος αρχείου ευρετηρίου: 15 blocks

Αναζήτηση χωρίς ευρετήριο:  $\lceil \log 3.000 \rceil + \text{ταιριάσματα} (= 3) \approx 15$  blocks

Αναζήτηση με ευρετήριο:  $\lceil \log 15 \rceil + 3 = 7$  blocks



## Ευρετήρια: Επανάληψη

---

- Ένα ευρετήριο (index) είναι μια βοηθητική δομή αρχείου που κάνει πιο αποδοτική την αναζήτηση μιας εγγραφής σε ένα αρχείο
- Το ευρετήριο καθορίζεται (συνήθως) σε ένα γνώρισμα του αρχείου
- Συχνά αποκαλείται access path (μονοπάτι πρόσβασης) στο γνώρισμα
- Μια καταχώρηση / εγγραφή στο ευρετήριο έχει την μορφή:

Τιμή Πεδίου Ευρετηριοποίησης	Δείκτης στο block της εγγραφής
------------------------------	--------------------------------



## Ευρετήρια: Επανάληψη

---

- Το αρχείο ευρετηρίου καταλαμβάνει μικρότερο χώρο από το ίδιο το αρχείο δεδομένων (οι καταχωρήσεις είναι μικρότερες και λιγότερες).
- Κάνοντας δυαδική αναζήτηση στο ευρετήριο βρίσκουμε τον δείκτη στο block όπου αποθηκεύεται η εγγραφή που θέλουμε.



## Ευρετήρια: Επανάληψη

---

- **Πυκνό Ευρετήριο** μια καταχώρηση για κάθε εγγραφή του δίσκου
- **Μη Πυκνό Ευρετήριο**
- **Ευρετήριο συστάδων (clustered index)** στο πεδίο διάταξης το οποίο όμως *δεν είναι κλειδί* - γενικότερα, όταν η διάταξη των καταχωρήσεων στο ευρετήριο ακολουθεί τη διάταξη των εγγραφών στο αρχείο.
- **Ευρετήριο χωρίς συστάδες (unclustered index)**
- **Πρωτεύον**
- **Δευτερεύον (πλήρως αντεστραμμένο ευρετήριο)**



## Ευρετήρια: Επανάληψη

---

Τα ευρετήρια (ενός επιπέδου) χωρίζονται σε:

- **Πρωτεύον Ευρετήριο:** ορίζεται σε ένα αρχείο που είναι **διατεταγμένο στο (κύριο) κλειδί**. Περιλαμβάνει μια καταχώρηση για κάθε block. Η καταχώρηση έχει την τιμή του κλειδιού της πρώτης εγγραφής στο block. (συντά ονομάζεται, μη-πυκνό ευρετήριο --- sparse index ή non-dense index.
- **Ευρετήριο Συστάδων (Clustering Index):** ορίζεται σε ένα αρχείο που είναι **διατεταγμένο σε γνώρισμα που δεν είναι κλειδί**. Περιλαμβάνει μια καταχώρηση για κάθε ξεχωριστή τιμή του γνωρίσματος. Η καταχώρηση "δείχνει" το πρώτο block που περιέχει εγγραφές με αυτή την τιμή γνωρίσματος





# Δευτερεύον Ευρετήριο

---

## Δευτερεύον Ευρετήριο (Secondary Index):

- ορίζεται σε ένα αρχείο που είναι **μη – διατεταγμένο** στο γνώρισμα. Περιλαμβάνει μια καταχώρηση για κάθε Εγγραφή (συχνά ονομάζεται, πυκνό ευρετήριο-dense index).
- ορισμένο σε πεδία διαφορετικά του κλειδιού διάταξης.

# Δευτερεύον Ευρετήριο

---

**Περίπτωση 1:** Το πεδίο ευρετηριοποίησης είναι κλειδί (καλείται και δευτερεύον κλειδί).

Υπάρχει μια εγγραφή για κάθε εγγραφή του αρχείου που περιέχει:

- την τιμή του δευτερεύοντος κλειδιού για αυτήν την εγγραφή.
- ένα δείκτη προς το block (ή την εγγραφή) του αρχείου δεδομένων που περιέχει την εγγραφή με την τιμή αυτή

## Δευτερεύον Ευρετήριο

---

- Το ευρετήριο αρχείου είναι ένα διατεταγμένο αρχείο με σταθερού μήκους εγγραφές.
- Το δευτερεύον ευρετήριο είναι ένα πυκνό ευρετήριο.
- Το μέγεθος του δευτερεύοντος ευρετηρίου είναι μικρότερο από του αρχείου δεδομένων (αν και μεγαλύτερο από το πρωτεύον).

## Δευτερεύον Ευρετήριο

---

### Παράδειγμα:

Έστω αρχείο με  $r = 30.000$  εγγραφές, μέγεθος block  $B = 1024$  bytes, σταθερού μεγέθους εγγραφές μεγέθους  $R = 100$  bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο κλειδιού έχει μέγεθος  $V = 9$  bytes αλλά δεν είναι πεδίο διάταξης. Κατασκευάζουμε δευτερεύον ευρετήριο, μέγεθος δείκτη block  $P = 6$  bytes.

Μέγεθος αρχείου δεδομένων: 3.000 blocks

Μέγεθος αρχείου ευρετηρίου: 442 blocks

**45 για πρωτεύον**

# Δευτερεύον Ευρετήριο

---

## ΑΝΑΖΗΤΗΣ

- Δυαδική αναζήτηση στο δευτερεύον ευρετήριο
- Ανάγνωση του block από το αρχείο δεδομένων



# Δευτερεύον Ευρετήριο

---

## Παράδειγμα:

Έστω αρχείο με  $r = 30.000$  εγγραφές, μέγεθος block  $B = 1024$  bytes, σταθερού μεγέθους εγγραφές μεγέθους  $R = 100$  bytes, μη εκτεινόμενη καταχώρηση, όπου το πεδίο κλειδιού έχει μέγεθος  $V = 9$  bytes αλλά δεν είναι πεδίο διάταξης. Κατασκευάζουμε δευτερεύον ευρετήριο, μέγεθος δείκτη block  $P = 6$  bytes.

Μέγεθος αρχείου δεδομένων: 3.000 blocks - Μέγεθος αρχείου ευρετηρίου: 442 blocks

Αναζήτηση χωρίς ευρετήριο:  $3.000/2 = 1500$  blocks

Αναζήτηση με ευρετήριο:  $\lceil \log 442 \rceil + 1 = 10$  blocks

Για πρωτεύον ήταν 45 και 7 blocks αντίστοιχα
--



## Δευτερεύον Ευρετήριο

---

### Περίπτωση 2:

Το πεδίο ευρετηριοποίησης δεν είναι κλειδί

1. Πυκνό ευρετήριο: μία καταχώρηση για κάθε εγγραφή.
2. Μεταβλητού μήκους εγγραφές με ένα επαναλαμβανόμενο πεδίο για το δείκτη.
3. Μία εγγραφή ευρετηρίου για κάθε τιμή του πεδίου ευρετηριοποίησης + ένα ενδιάμεσο επίπεδο για την διαχείριση των πολλαπλών δεικτών.

## Δευτερεύον Ευρετήριο

---

### Αναζήτηση:

- Διαδική αναζήτηση στο δευτερεύον ευρετήριο.
- Ανάγνωση του block (ή των blocks) από το ενδιάμεσο επίπεδο.
- Ανάγνωση των blocks (συνήθως τόσα όσες οι εγγραφές που ταιριάζουν) από το αρχείο δεδομένων.

## Δευτερεύον Ευρετήριο

---

### Εισαγωγή:

- Πολύ απλή αν δεν αφορά εισαγωγή νέας τιμής στο ευρετήριο.
- Εύκολη η λογική διάταξη των εγγραφών με βάση το πεδίο ευρετηριοποίησης.
- Ανακτήσεις με σύνθετες συνθήκες.



## Data blocks

---

Η βάση δεδομένων αποθηκεύει τα δεδομένα στο σκληρό δίσκο σε ένα ή πολλά αρχεία. Συνήθως σε πολλά εκτός από την MS Access που αποθηκεύει σε ένα. Η μονάδα ανάκτησης σειρών ενός πίνακα ονομάζεται data block. Η ανάκτηση από το σκληρό δίσκο είναι το πιο αργό κομμάτι εκτέλεσης ενός ερωτήματος. Εσείς μπορείτε να ρυθμίσετε το μέγεθος ενός data block το οποίο εξαρτάται από:

- το μέγεθος το πίνακα
- το είδος των ερωτημάτων.



## Τεχνικά ζητήματα στα ευρετήρια

---

- Υπάρχει ένα κρυφό το οποίο δηλώνει τον κωδικό της κάθε γραμμής Rowid στην Oracle
- Τα ευρετήρια συσχετίζουν το rowid με ένα πεδίο (ή πεδία)
- Το κάθε rowid ξέρουμε σε ποιο data block είναι αποθηκευμένο με απώτερο σκοπό να προσφέρουμε ταξινόμηση με βάση αυτό το πεδίο

## Παράδειγμα

---

Πίνακας students με 10 000 εγγραφές

→ Students (id, name, AEM, ...)

Εμφάνισε τα στοιχεία του φοιτητή με AEM 3543453



Χωρίς  
ευρετήριο

→ Select \* from students where AEM= 3543453

**Σχόλιο!** Στην καλύτερη περίπτωση μία σύγκριση, στη χειρότερη 10 000 μέσος όρος 5000 συγκρίσεις.

## Παράδειγμα με ευρετήριο

---

Εισάγουμε ένα ευρετήριο στο πεδίο AEM. Αυτό παρέχει ταξινομημένες τις τιμές του AEM με δείκτη στο rowid (σκοπός μας είναι να βρούμε το rowid το οποίο περιέχει τα στοιχεία για το AEM που επιζητούμε). Εφόσον όμως στο ευρετήριο οι τιμές είναι ταξινομημένες, η αναζήτηση γίνεται δυαδικά. Γι αυτό η εύρεση θα γίνει πολύ πιο γρήγορα.

## Βελτίωση ταχύτητας εύρεσης

---

Με την χρήση ευρετηρίων υπάρχει μία μείωση των αναζητήσεων και βελτίωση της ταχύτητας ταυτόχρονα της τάξεως:

- αν  $N$  είναι το πλήθος των εγγραφών ο μέσος όρος αναζητήσεων μειώνεται από  $N/2$  σε  $\log_2 N - 1$

## Αναγκαιότητα χρήσης ευρετηρίων

---

**Μόνο στα πεδία τα οποία χρησιμοποιούνται σε**

- κριτήρια (WHERE)
- Συναρτήσεις (MAX, MIN...)
- Ομαδοποίηση (GROUP BY)
- Ταξινόμηση (ORDER BY)
- (και όχι στο SELECT)

**Σε ερωτήματα**

- Θυμηθείτε τον κανόνα 80/20

**Σε πεδία που παρουσιάζουν ποικιλία διαφορετικών τιμών**



## Αποφυγή δημιουργίας ευρετηρίων

---

Η εισαγωγή ευρετηρίων δεν είναι πολύ χρήσιμη πάντοτε καλό είναι να αποφεύγεται η δημιουργία τους όταν:

- Έχετε ένα πεδίο τύπου Bool
- Οι πίνακες είναι μικροί

**Σχόλιο!** Δεν είναι καλό να τοποθετούνται ευρετήρια σε μικρούς πίνακες γιατί αποθηκεύονται σε ελάχιστα data blocks.





## Τρόπος δημιουργίας ευρετηρίων

---

Δημιουργούνται με εντολή CREATE

```
→CREATE INDEX student_AEM_idx  
ON Students (AEM) ;
```

Δημιουργεί ένα ευρετήριο στο πεδίο AEM του πίνακα Students

Καταργούνται τα ευρετήρια με την εντολή DROP

```
→ DROP INDEX student_AEM_idx
```

**Σχόλιο!** Μπορούμε να έχουμε παραπάνω από ένα ευρετήριο σε ένα πεδίο αλλά εκτός των εξαιρέσεων γιατί δεν έχει νόημα.

## Παρατηρήσεις στα ευρετήρια

---

Δεν μπορούμε να δημιουργήσουμε ευρετήρια σε όψεις γιατί συνήθως τα ευρετήρια εισάγονται σε ξένα κλειδιά τα οποία χρησιμοποιούνται συχνά σε JOIN ερωτήματα.

## Παράδειγμα

Έστω ένας πίνακας **ΦΟΙΤΗΤΕΣ-ΜΑΘΗΜΑΤΑ-ΒΑΘΜΟΛΟΓΙΕΣ**

- ΜΑΘΗΜΑΤΑ (ΚΩΔ.ΜΑΘΗ, ΜΑΘΗΜΑ) 10 εγγραφές
- ΦΟΙΤΗΤΕΣ(ΑΕΜ ,ΟΝΟΜΑ, ΕΠΙΘΕΤΟ)10 000 εγγραφές
- ΒΑΘΜΟΛΟΓΙΕΣ(ΚΩΔ\_ΒΑΘΜ,ΜΑΘΗΜΑ,ΒΑΘΜΟΣ) 1000 εγγραφές

Ποιοι φοιτητές πέρασαν το μάθημα μαθηματικά;

```
→SELECT ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ
FROM ΦΟΙΤΗΤΕΣ, ΜΑΘΗΜΑΤΑ, ΒΑΘΜΟΛΟΓΙΕΣ
WHERE ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ =
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ AND ΦΟΙΤΗΤΕΣ.ΑΕΜ =
ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND ΒΑΘΜΟΛΟΓΙΕΣ.ΒΑΘΜΟΣ>=5 AND
ΜΑΘΗΜΑ= "ΜΑΘΗΜΑΤΙΚΑ"
```



## Παράδειγμα

```
→SELECT ΦΟΙΤΗΤΕΣ.ΑΕΜ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΜΑΘΗΜΑ, ΒΑΘΜΟΣ
FROM ΦΟΙΤΗΤΕΣ, ΜΑΘΗΜΑΤΑ, ΒΑΘΜΟΛΟΓΙΕΣ
WHERE ΦΟΙΤΗΤΕΣ.ΜΑΘΗΜΑΤΑ=ΜΑΘΗΜΑΤΑ.ΚΩΔ_ΜΑΘΗΜΑΤΟΣ AND
ΜΑΘΗΜΑΤΑ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ =
ΒΑΘΜΟΛΟΓΙΕΣ.ΚΩΔΙΚΟΣ_ΜΑΘΗΜΑΤΟΣ AND ΦΟΙΤΗΤΕΣ.ΑΕΜ =
ΒΑΘΜΟΛΟΓΙΕΣ.ΑΕΜ AND ΒΑΘΜΟΛΟΓΙΕΣ.ΒΑΘΜΟΣ>=5 AND ΜΑΘΗΜΑ=
"ΜΑΘΗΜΑΤΙΚΑ"
```

- Εισάγουμε τα παρακάτω ευρητήρια
- ΦΟΙΤΗΤΕΣ(ΜΑΘΗΜΑΤΑ)
- ΚΩΔΙΚΟΣ\_ΜΑΘΗΜΑΤΟΣ(ΜΑΘΗΜΑΤΑ)
  - Ακόμη καλύτερα θα ήταν  
(Modules (semesterΜΑΘΗΜΑΤΑ )



## Συμπεράσματα

---

Με την δημιουργία της βάσης δεδομένων η δουλειά δε σταματά στη δημιουργία πινάκων, απαιτείται και η δημιουργία κατάλληλων ερωτηρίων.

Πριν την εισαγωγή αυτών ελέγξτε:

- Το μέγεθος πίνακα
- Την ποικιλία των τιμών
- Την χρήση τους σε WHERE, GROUP BY, ORDER BY

## Ενότητα 13

---

### Σύγχρονα θέματα βάσεων δεδομένων

Δρ. Τσιμπίρης Αλκιβιάδης



### Περιεχόμενα ενότητας

---

- Εισαγωγή στις έννοιες ΒΔ πολυμέσων
- Χαρακτηριστικά των τύπων πηγής πολυμέσων
- Πηγή πολυμέσου-Εικόνα
- Πηγή πολυμέσου-Βίντεο
- Πηγή πολυμέσου-Κειμένου
- Πηγή πολυμέσου-Ήχου
- Χωρικές βάσεις δεδομένων
- Κατηγορίες τυπικών χωρικών επερωτήσεων
- Γεωγραφικά συστήματα πληροφοριών
- Απαιτήσεις διαχείρισης δεδομένων για τα ΓΣΠ
- ARC-INFO



# Σκοποί ενότητας

---

Στην ενότητα αυτή γίνεται μία γενική εισαγωγή στις έννοιες των χωρικών και πολυμεσικών βάσεων δεδομένων που αποτελούν έναν τομέα σύγχρονο με μεγάλο ενδιαφέρον και προοπτική.



## Εισαγωγή στις έννοιες ΒΔ πολυμέσων

---

Οι βάσεις δεδομένων πολυμέσων παρέχουν χαρακτηριστικά που επιτρέπουν στους χρήστες να αποθηκεύουν και να ζητούν διευκρινήσεις για διαφορετικούς τύπους πληροφοριών πολυμέσων που περιλαμβάνουν **εικόνες** (όπως φωτογραφίες ή σχέδια), **ακολουθίες βίντεο** (ταινίες ειδήσεις κλπ), **ακολουθίες ήχου** (όπως τραγούδια, τηλεφωνικά μηνύματα, ή αγορεύσεις) και **έγγραφα** (όπως βιβλία και άρθρα).



## Εισαγωγή στις έννοιες ΒΔ πολυμέσων

---

Οι βασικοί τύποι ικανοποιητικών ερωτήσεων βάσης δεδομένων που απαιτούνται περιλαμβάνουν τον εντοπισμό των πηγών πολυμέσων οι οποίες περιέχουν κάποια αντικείμενα που σας ενδιαφέρουν. Οι τύποι αυτοί αναφέρονται σαν **ανάκτηση με βάση το περιεχόμενο**.

Επομένως μια ΒΔ πολυμέσων **πρέπει** να χρησιμοποιεί κάποιο μοντέλο για την οργάνωση και τα ευρετήρια των πηγών πολυμέσων που θα βασίζεται στα περιεχόμενα τους. Ο προσδιορισμός των περιεχομένων των πηγών πολυμέσων είναι μια δύσκολη και χρονοβόρα προσπάθεια.



## Εισαγωγή στις έννοιες ΒΔ πολυμέσων

---

Υπάρχουν δύο βασικές προσεγγίσεις:

- Βασίζεται στην **αυτόματη ανάλυση των πηγών πολυμέσων** για τον προσδιορισμό κάποιων μαθηματικών χαρακτηριστικών του περιεχομένου τους. Η προσέγγιση αυτή χρησιμοποιεί διαφορετικές τεχνικές ανάλογα με τον τύπο της πηγής του πολυμέσου (εικόνα, κείμενο, βίντεο ή ήχος).
- Η δεύτερη προσέγγιση εξαρτάται από τον **χειρονακτικό προσδιορισμό των αντικειμένων** και των δραστηριοτήτων που ενδιαφέρουν σε κάθε πηγή πολυμέσων και στην χρησιμοποίηση αυτών των πληροφοριών για τη δημιουργία ευρετηρίων. Η προσέγγιση αυτή μπορεί να εφαρμοσθεί σε όλες τις διαφορετικές πηγές πολυμέσων, αλλά απαιτεί μια φάση χειρονακτικής προεπεξεργασίας όπου κάποιο άτομο πρέπει να σαρώσει κάθε πηγή πολυμέσου για να προσδιορίσει και να καταγράψει τα αντικείμενα και τις δραστηριότητες που περιέχει ώστε να μπορούν να χρησιμοποιηθούν για εύρεση αυτών των πηγών



## Χαρακτηριστικά των τύπων πηγής πολυμέσων

---

Χαρακτηριστικά κάθε τύπου πηγής πολυμέσου (εικόνες, βίντεο, ήχος και πηγές κειμένων)

**Εικόνα** αποθηκεύεται σε μη επεξεργασμένη μορφή σαν ένα σύνολο από τιμές πλέγματος ή κελιών, ή σε συμπιεσμένη μορφή για εξοικονόμηση χώρου. Η περιγραφή σχήματος της είναι ένα ορθογώνιο από κελιά συγκεκριμένου πλάτους και ύψους. Κάθε εικόνα μπορεί να αναπαρασταθεί από ένα  $m$  επί  $n$  πλέγμα από κελιά. Κάθε κελί περιέχει την τιμή ενός εικονοστοιχείου που περιγράφει το περιεχόμενο του κελιού.

Μαυρόασπρες εικόνες  $\rightarrow$  1bit

Γκρίζα κλίμακα ή στις έγχρωμες  $\rightarrow$  πολλά bits

**Σχόλιο!** Οι εικόνες απαιτούν μεγάλο χώρο και αποθηκεύονται σε συμπιεσμένη μορφή.



## Πηγή πολυμέσου-Εικόνα

---

**Δύο βασικές τεχνικές τύπων αναζήτησης εικόνας:**

- **Συνάρτηση απόστασης** συγκρίνει τη δοθείσα εικόνα με τις αποθηκευμένες εικόνες και τα τμήματά τους. Αν η τιμή της απόστασης που επιστρέφεται είναι μικρή, τότε η πιθανότητα εύρεσης του προτύπου είναι μεγάλη. Μπορούν να δημιουργηθούν ευρετήρια που να ομαδοποιούν τις αποθηκευμένες εικόνες με περίπου ίδια μετρική απόσταση έτσι που να περιορίζεται ο χώρος της αναζήτησης.
- **Προσέγγιση μετασχηματισμού** μετράει την ομοιότητα των εικόνων εκτελώντας ένα μικρό αριθμό μετασχηματισμών οι οποίοι μπορούν να μετασχηματίσουν τα κελιά μιας εικόνας ώστε να καταφέρουν να τα ταιριάξουν με την άλλη εικόνα. Οι μετασχηματισμοί περιλαμβάνουν στροφές, μεταφράσεις και κλιμάκωση.



## Πηγή πολυμέσου-Βίντεο

---

Ένα βίντεο αναπαρίσταται από μια ακολουθία από καρτέ, όπου κάθε καρτέ είναι μια παγωμένη εικόνα. Το βίντεο:

- Τοποθετεί τα αντικείμενα και τις δραστηριότητες σε κάθε ξεχωριστό καρτέ
- Χωρίζεται σε τμήματα, όπου κάθε τμήμα αποτελείται από μια ακολουθία διαδοχικών καρτέ που περιέχουν τα ίδια αντικείμενα και τις ίδιες δραστηριότητες
- Κάθε τμήμα προσδιορίζεται από το αρχικό και το τελικό καρτέ.
- Τα αντικείμενα και οι δραστηριότητες που προσδιορίζονται σε κάθε τμήμα βίντεο μπορούν να χρησιμοποιηθούν για την ευρετηριοποίηση των τμημάτων.

## Πηγή πολυμέσου-Βίντεο

---

Για την ευρετηριοποίηση των βίντεο έχει προταθεί μια τεχνική ευρετηριοποίησης που ονομάζεται **δένδρα τμημάτων καρτέ**. Το ευρετήριο περιλαμβάνει τόσο αντικείμενα, όπως άτομα, σπίτια, αυτοκίνητα, όσο και δραστηριότητες, όπως ένα άτομο που κάνει μια ομιλία ή δύο άτομα που συζητούν.



## Πηγή πολυμέσου-Πηγή κειμένου

---

**Πηγή κειμένου:** είναι το πλήρες κείμενο κάποιου άρθρου, βιβλίου ή περιοδικού. Για την ευρετηριοποίηση των πηγών χρησιμοποιείται ο προσδιορισμός των λέξεων-κλειδίων που εμφανίζονται στο κείμενο καθώς και των σχετικών συχνοτήτων τους. Για τον σκοπό αυτό χρησιμοποιείται μια τεχνική που βασίζεται σε μετασχηματισμούς πινάκων και ονομάζεται **διασπάσεις μοναδικών τιμών (singular value decompositions (SVD))**. Στη συνέχεια μπορεί να χρησιμοποιηθεί μια τεχνική ευρετηριοποίησης που ονομάζεται **δένδρα τηλεσκοπικών διανυσμάτων ή TV-δένδρα** για την ομαδοποίηση κειμένων.

## Πηγή πολυμέσου-Ήχου

---

**Πηγή ήχου:**

- Περιέχουν ηχογραφημένα μηνύματα όπως ομιλίες, παρουσιάσεις διαλέξεων ή ακόμη καταγραφές παρακολούθησης τηλεφωνικών μηνυμάτων ή συνδιαλέξεων που έχουν επιβληθεί από τον νόμο

**Σχόλιο!** Για τον εντοπισμό των κύριων χαρακτηριστικών της φωνής κάποιου ατόμου μπορούν να χρησιμοποιηθούν διακριτοί μετασχηματισμοί, ώστε να έχουμε ευρετηριοποίηση και ανάκτηση βασισμένες στην ομοιότητα. Τα χαρακτηριστικά στοιχεία του ήχου περιλαμβάνουν ένταση, τόνο και καθαρότητα.

# Χωρικές βάσεις δεδομένων

---

**Οι χωρικές βάσεις δεδομένων:**

- Υποστηρίζουν έννοιες για βάσεις δεδομένων που καταγράφουν αντικείμενα σ' έναν πολυδιάστατο χώρο
- Αποθηκεύουν αντικείμενα που περιγράφονται με χωρικά χαρακτηριστικά
- Χρειάζονται χωρικές πράξεις για να εφαρμοσθούν στα χωρικά χαρακτηριστικά των αντικειμένων
- Χρειάζονται μια περιγραφή των χωρικών θέσεων πολλών τύπων αντικειμένων



## Κατηγορίες τυπικών χωρικών επερωτήσεων

---

**Επερώτηση διαστήματος:** Βρίσκει τα αντικείμενα ενός συγκεκριμένου τύπου που βρίσκονται μέσα σε δεδομένη χωρική περιοχή ή σε συγκεκριμένη απόσταση από δοθείσα θέση.

**Επερώτηση για τον κοντινότερο γείτονα:** Βρίσκει ένα αντικείμενο συγκεκριμένου τύπου που βρίσκεται πιο κοντά σε δεδομένη θέση.

**Χωρικές συνενώσεις ή επικαλύψεις:** Συνενώνει αντικείμενα δύο τύπων με βάση κάποια χωρική συνθήκη, όπως αντικείμενα που τέμνονται ή επικαλύπτονται χωρικά ή που βρίσκονται σε κάποια απόσταση το ένα από το άλλο.



# Γεωγραφικά συστήματα πληροφοριών

---

Τα γεωγραφικά συστήματα πληροφοριών (ΓΣΠ-GIS) χρησιμοποιούνται για συλλογή, μοντελοποίηση, αποθήκευση και ανάλυση πληροφοριών που περιγράφουν φυσικές ιδιότητες του γεωγραφικού κόσμου.

**Τύποι δεδομένων:**

- Χωρικά δεδομένα, προερχόμενα από χάρτες, ψηφιακές εικόνες, διοικητικά και πολιτικά όρια, δρόμους, δίκτυα μεταφορών, φυσικά δεδομένα, ποτάμια, εδαφολογικά χαρακτηριστικά, κλιματολογικές περιοχές, λόφους.
- Μη χωρικά δεδομένα, όπως μετρήσεις απογραφής οικονομικά δεδομένα και πληροφορίες πωλήσεων και προώθησης προϊόντων.

Το ΓΣΠ αποτελούν ένα γρήγορα αναπτυσσόμενο πεδίο που προσφέρει πολύ πρωτοποριακές προσεγγίσεις για την αντιμετώπιση μερικών τεχνικών απαιτήσεων.



## Απαιτήσεις διαχείρισης δεδομένων για τα ΓΣΠ

---

- Μοντελοποίηση Δεδομένων και Αναπαράσταση
- Ανάλυση Δεδομένων
- Ενοποίηση Δεδομένων
- Συλλογή Δεδομένων



# ARC-INFO

---

Είναι απαραίτητο να σχεδιασθούν συστήματα που να υποστηρίζουν τη διανυσματική και την ψηφιοποιημένη αναπαράσταση, τη χωρική λειτουργικότητα καθώς και τα απαραίτητα χαρακτηριστικά ΣΔΒΔ γιατί τα ΣΣΔΒΔ ή τα ΑΣΔΒΔ **δεν καλύπτουν** τις ειδικές ανάγκες των γεωγραφικό σύστημα πληροφοριών. Ένα δημοφιλές γεωγραφικό σύστημα πληροφοριών ονομάζεται **ARC-INFO** και το οποίο δεν είναι ΣΔΒΔ αλλά ενοποιεί τη λειτουργικότητα των ΣΣΔΒΔ στο INFO τμήμα του συστήματος.

# Ενότητα 14

---

## XML

Δρ. Τιμπίρης Αλκιβιάδης



---

## Περιεχόμενα ενότητας

- Εισαγωγή στην XML
- Τι είναι XML
- Ιστορική Αναδρομή
- Πλεονεκτήματα
- Καινοτομίες
- Δομή της XML
- Στοιχεία XML
- Χρήση της XML
- Χαρακτηριστικά
- DTD Παράδειγμα
- Παρουσίαση XML Εγγραφών
- Λειτουργίες XML
- XML QUERY Γλώσσες
- XML Native ΒΔ Βασισμένη σε πίνακα χαρτογράφησης
- Αντικειμενοστραφής χαρτογράφηση
- Ερωτήσεις XML Σε Σχεσιακές ΒΔ
- Παραδείγματα Ευρετηρίου
- Σύστημα Αρίθμησης



## Σκοποί ενότητας

---

Ο σκοπός της ενότητας αυτής είναι η καταρχήν η εισαγωγή στην XML και μέσα από μια ιστορική αναδρομή να παρουσιαστεί η ανάγκη που οδήγησε στον ορισμό της. Παρουσιάζονται τα πλεονεκτήματα και οι καινοτομίες που παρέχει. Ο ορισμός της δομή της XML με τα απαραίτητα στοιχεία σχήματα (DTD) XML μπορούν να την κάνουν εργαλείο χρήσιμο καταρχήν για τη μεταφορά δεδομένων μεταξύ ομοίων ή και διαφορετικών RDBMS.

## Εισαγωγή στην XML

---

- Με την τεράστια εξάπλωση του παγκόσμιου ιστού, η ανάγκη για ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων και πλατφορμών έγινε επιτακτική
- Το κύριο πρόβλημα είναι ότι η μορφή και ο τύπος των δεδομένων ποικίλλει. Μπορεί να είναι αρχεία κειμένου, δεδομένα βάσεων δεδομένων, μεταδεδομένα(metadata) κτλ.
- Επιτακτική ανάγκη για ένα κοινό πρότυπο αναπαράστασης και ανταλλαγής δεδομένων, κοινό για όλες τις πλατφόρμες

# Τι είναι XML

---

- Το XML αποτελεί συντομογραφία του **eXtensive Markup Language**
- Σύνολο απο κανόνες για την δημιουργία ετικετών (tags) που **περιγράφουν** τα δεδομένα ενός εγγράφου καθώς και προσδιορίζουν και τα διάφορα μέρη απο τα οποία αποτελείται ένα έγγραφο
- Οι ετικέτες δεν είναι προκαθορισμένες, τις ορίζουμε εμείς
- Μεταγλώσσα με την οποία μπορούμε να ορίσουμε άλλες γλώσσες σήμανσης

## Ιστορική Αναδρομή

---

- Η ανάπτυξη του **XML** ξεκίνησε το 1996
- Εντάχθηκε στο **W3C** το 1998
- Βασίζεται στην **SGML (Standard Generalized Markup Language)** και είναι ένα υποσύνολό της. Διατήρησε τα λειτουργικά χαρακτηριστικά της, αλλά απέβαλε τα στοιχεία εκείνα που την έκαναν **δύσχρηστη** στο προγραμματισμό
- Ευρέως γνωστή στον τομέα του διαδικτύου

# Πλεονεκτήματα

---

- Εύκολο στην κατανόηση
- Κοινό πρότυπο μεταξύ διαφορετικών πλατφορμών
- Αυτο-περιγραφική γλώσσα
- Αποθήκευση σε ASCII κείμενο
- Ευελιξία στη δομή καθώς ο καθένας δημιουργεί όσες και όποιες ετικέτες θέλει
- Ευανάγνωστα αρχεία
- Οι περισσότερες εφαρμογές υποστηρίζουν την εξαγωγή και εισαγωγή στοιχείων από έγγραφα XML
- Υποστήριξη από πληθώρα τεχνολογιών και εργαλείων



# Καινοτομίες

---

Σημαντικότερες αλλαγές που έφερε το XML:

- Δεδομένα
- Αρχιτεκτονική
- Λογισμικό





# Καινοτομίες

---

## Δεδομένα

- Αποδέσμευση των δεδομένων από τις εφαρμογές
- Δημιουργία επιχειρηματικών λεξικών
- Ανάπτυξη της B2B επικοινωνίας

## Αρχιτεκτονική

- Ανάπτυξη κατακευμαμένων υπολογιστικών συστημάτων
- Χρήση της XML και του διαδικτύου (TCP/IP) για επικοινωνία και ανταλλαγή δεδομένων και πληροφοριών



# Καινοτομίες

---

## Λογισμικό

- Με τη χρήση της XML επεκτείνεται η δυνατότητα επικοινωνίας διαφόρων υποπρογραμμάτων και εφαρμογών μεταξύ τους
- Προγράμματα βασισμένα σε modules για τη λύση συγκεκριμένων προβλημάτων
- Απλότητα και ευελιξία



# Δομή του XML

---

- Ένα XML έγγραφο μοιάζει με ένα HTML έγγραφο
  - Το **XML** σχεδιάστηκε για να περιγράφει δεδομένα
  - Η **HTML** σχεδιάστηκε για να εμφανίζει δεδομένα
- Αποτελείται από **tags** τα οποία είναι υποχρεωτικό να κλείνονται (σε αντίθεση με την HTML)
  - π.χ., `<author>George R. R. Martin</author>`
- Επιτρέπονται **άπειρα** επίπεδα εμφωλευμένων **tag**
- **Απαγορεύεται** οι ετικέτες να ξεκινούν με 'XML' είτε σε πεζά είτε σε κεφαλαία

# Στοιχεία XML

---

- **Element:** Το βασικό δομικό στοιχείο
  - `<book> Game of Thrones </book>`
- **Attribute:** Ιδιότητα ενός Element
  - `<book author="George R. R. Martin"> Game of Thrones </book>`
- Τα attributes μπορούν να γραφούν και με τη μορφή εμφωλευμένων Element
  - π.χ.,
  - `<book>`
  - `<title>Game of Thrones </title>`
  - `<author>George R. R. Martin</author>`
- `</book>`

# Στοιχεία XML

---

- **Entity:** Οι οντότητες είναι αλφαριθμητικά που χρησιμοποιούνται ως συντομογραφίες άλλων αλφαριθμητικών  
π.χ., `<!Entity message "Hello World">`
- Με αυτόν τον τρόπο, όταν γράφουμε `&message` ισοδυναμεί με `"Hello World"`

# Χρήση της XML

---

- **Επιχειρηματικά δεδομένα** (πωλήσεις, τιμολόγια, παραγγελίες)
- **Στοιχεία απο βάσεις δεδομένων**
  - Όλα τα σύγχρονα RDBMS υποστηρίζουν εξαγωγή και εισαγωγή από XML
- **Οικονομικά δεδομένα**
- **Human Resources XML (HR-XML)**

# Χαρακτηριστικά

---

- Δεν κάνει υπόθεση για τον τρόπο παρουσίασης των εγγράφων
- Δεν ορίζονται τύποι στα δεδομένα της
- Δεν υπάρχει περιορισμός στον τρόπο μετάδοσης των εγγράφων
- Περιγράφει μόνο τη δομή και το περιεχόμενο των δεδομένων και όχι τον τρόπο παρουσίασης τους(Αντίθετα η HTML περιγράφει τον τρόπο παρουσίασης κάποιων δεδομένων)
- Οι ετικέτες του εγγράφου συνήθως σχετίζονται με την οντότητα στην οποία αναφέρονται

# DTD Παράδειγμα

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE movie[
<!ELEMENT movie(title, director)>
<!ELEMENT title(#PCDATA)>
<!ELEMENT author(#PCDATA)>
]>
<movie>
  <title>The Lord of the Rings: The Two Towers</title>
  <director>Peter Jackson</director>
</movie>
```

# Παρουσίαση XML Εγγραφών

---

Η XML δεν ορίζει τον τρόπο με τον οποίο θα παρουσιαστούν τα δεδομένα που περιγράφει

- Για την παρουσίαση, χρησιμοποιούνται άλλες γλώσσες ή εργαλεία
  - **CSS** (Cascading Style Sheets)
  - **XSL** (eXtensive Style sheets Language)
  - **XForms**
- Μειονεκτήματα CSS
  - Απαιτείται DTD
  - Εξαρτάται από τον browser



# Παρουσίαση XML Εγγραφών

---

- Πλεονεκτήματα XSL
  - Αλλαγή στη σειρά εμφάνισης των στοιχείων
  - Ανεξάρτητο browser
  - Περισσότερες λειτουργίες και δυνατότητες
- XForms
  - Αντικατάσταση των HTML φόρμών
  - Διαχωρισμός εμφάνισης και λειτουργικότητας
  - Αποτέλεσμα σε XML



# Επεξεργασία

---

- Κάθε XML έγγραφο μπορεί να θεωρηθεί ως μια πηγή πληροφορίας δεδομένων
- Πρέπει να υποστηρίζονται πράξεις αναζήτησης πληροφορίας
- Κάθε XML έγγραφο μπορεί να θεωρηθεί ως ένα μη κατευθυνόμενο δέντρο

# Λειτουργίες XML

---

Οι λειτουργίες που θα πρέπει να υποστηρίζονται σε ένα XML έγγραφο είναι:

- Εύρεση σχέσης πατέρα - γιου
- Εύρεση σχέσης προγόνου - απόγονου
- Έλεγχος τιμής χαρακτηριστικού
- Καμία, μία ή περισσότερες εμφανίσεις ενός κόμβου
- Twig patter matching
- Ταίριασμα υποδέντρου

# XML QUERY Γλώσσες

---

Γλώσσες αναζήτησης

- XPath
- XQuery
- XPointer

XPath

- Δημιουργεί μια δενδρική δομή από το XML έγγραφο
- Εκφράσεις
- Πατέρας-γιος: '/'
- Πρόγονος - Απόγονος: '//'
- Attribute: '@'
- Book/author[@name="John"]

Μία ή περισσότερες εμφανίσεις book/chapter+/title



# XML Native ΒΔ

---

Πολλές επιχειρήσεις χρησιμοποιούν την XML για αποθήκευση των επιχειρηματικών τους δεδομένων:

- Ημιδομημένα δεδομένα (semistructured data)
- Ανάπτυξη ολοκληρωμένων συστημάτων διαχείρισης
- XML εγγράφων (XML native databases)
- XML native βάσεις δεδομένων
- Συστήματα διαχείρισης XML εγγράφων, τα οποία
- Συστατικά: χρησιμοποιούν δικούς τους τρόπους αποθήκευσης, επεξεργασίας και απεικόνισης XML εγγράφων και δεδομένων
- Τρόπος αποθήκευσης
- Συλλογές XML εγγράφων

Τρόπος εκτέλεσης ερωτημάτων



# Αποθήκευση XML εγγράφων

---

## Φυσικά αρχεία

- Αργή πρόσβαση
- Απουσία δομών δεικτοδότησης

## Σχεσιακές βάσεις δεδομένων (RDBMS)

- Ανάγκη μετατροπής σε πίνακες και στήλες
- Υπαρξη δομών δεικτοδότησης
- Χρήση SQL για ερωτήσεις
- Ανάγκη DTD Schema



# XML Native ΒΔ

---

- Για να αποθηκευτεί ένα XML έγγραφο σε μια βάση δεδομένων, πρέπει το schema του να μετατραπεί σε ένα ανάλογο schema βάσης δεδομένων
- Διαχωρισμός elements, attributes και κειμένου.
- Δεν λαμβάνουμε υπόψιν την σειρά των elements και attributes
- Είδη χαρτογράφησης
  - Βασισμένη σε πίνακα
  - Αντικειμενοστραφής





## Βασισμένη σε πίνακα χαρτογράφησης

---

- Ένα XML έγγραφο μπορεί να αποθηκευτεί είτε ως ένας πίνακας είτε ως πολλοί
- Ανάλογα με το λογισμικό που χρησιμοποιείται, τα στοιχεία αποθηκεύονται
- Απαιτείται το XML έγγραφο να έχει την εξής μορφή:

```
<database>  
  
    <table>  
  
        <row>Όνομα</row>  
  
        <row>Ημερομηνία</row>  
  
    </table>  
  
</database>
```



## Βασισμένη σε πίνακα χαρτογράφησης

---

Τα attributes αποθηκεύονται αυτόματα ως στήλες στους αντίστοιχους πίνακες:

### Πλεονεκτήματα

- Εύκολη υλοποίηση
- Δεν απαιτείται DTD

### Μειονεκτήματα

- Δεν υποστηρίζονται όλα τα XML έγγραφα( πολλαπλές εμφωλεύσεις)



## Αντικειμενοστραφής χαρτογράφηση

---

- Χρησιμοποιείται σχεδόν σε όλες τις σύγχρονες σχεσιακές βάσεις δεδομένων
- Απαιτείται DTD σχήμα
- Οι τύποι στοιχείων και οι ιδιότητες τους δημιουργούνται ως κλάσεις
- Τα απλά στοιχεία (που έχουν μόνο τιμή κειμένου) διαμορφώνονται ως scalar ιδιότητες
- Οι κλάσεις αποθηκεύονται σε πίνακες και οι scalar ιδιότητες σε στήλες των αντίστοιχων πινάκων
- Οι ιδιότητες χαρτογραφούνται σε ζευγάρια πρωτεύοντος/ξένου κλειδιού
- Πλεονεκτήματα
  - Υποστηρίζονται όλα τα XML έγγραφα
  - Διατηρούνται οι ιδιότητες των πεδίων
  - Μπορούμε να ανακτήσουμε το DTD σχήμα από το σχήμα δεδομένων



## Αντικειμενοστραφής χαρτογράφηση: παράδειγμα

---

```
<!ELEMENT Order (OrderNum, Date, CustNum, Item*)>
  <!ELEMENT OrderNum (#PCDATA)>
  <!ELEMENT CustNum (#PCDATA)>

  <!ELEMENT Item (ItemNum, Quantity, Part)>
    <!ELEMENT ItemNum (#PCDATA)>
    <!ELEMENT Date (#PCDATA)>
    <!ELEMENT Quantity (#PCDATA)>

    <!ELEMENT Part (PartNum, Price)>
      <!ELEMENT PartNum (#PCDATA)>
      <!ELEMENT Price (#PCDATA)>
```



## Αντικειμενοστραφής χαρτογράφηση: παράδειγμα 2

---

### Βήμα 1

Για κάθε σύνθετο στοιχείο, δημιουργείται ένας πίνακας και ένα αυτόματο πρωτεύον κλειδί πίνακα

Π.χ Δημιουργία Πινάκων Order, Item, Part και των αντίστοιχων PK: *OrderPK*, *ItemPK*, *PartPK*

### Βήμα 2

Για κάθε απλό στοιχείο δημιουργείται μια στήλη στον αντίστοιχο πίνακα

Π.χ Δημιουργία των στηλών

OrderNum, Date, CustNum (πίνακας Order)

ItemNum,Quantity (πίνακας Item)

PartNum, Price (πίνακας Part)



## Αντικειμενοστραφής χαρτογράφηση: παράδειγμα 2

---

### Βήμα 3

Παραγωγή ξένων κλειδιών για αναφορά στα σύνθετα στοιχεία

Π.χ Δημιουργία ξένου κλειδιού OrderFK (πίνακας Item),  
ItemFK(πίνακας Part)



## Ερωτήσεις XML Σε Σχεσιακές ΒΔ

---

- Για την επεξεργασία αποθηκευμένων XML εγγράφων υπάρχουν δύο τεχνικές
- Χρήση SQL και μετατροπή των αποτελεσμάτων σε XML
- Χρήση XML Query γλώσσας και μετατροπής της σε SQL statements
- Ευρετήρια:
  - Εγγενή ευρετήρια της βάσης
  - Πρόσθετα ευρετήρια βελτιστοποιημένα για XML

## Παράδειγμα Ευρετηρίου 1

---

Ένα απλό παράδειγμα ευρετηρίου είναι το 1 - index  
Στο index αυτό όλοι οι κόμβοι ενός XML δέντρου ομαδοποιούνται ανάλογα με το path τους απο το root

Παράδειγμα  
<βάλε εικόνα>

## Παράδειγμα Ευρετηρίου 2

---

- Στα φύλλα του ευρετηρίου αποθηκεύονται δείκτες οι οποίοι δείχνουν σε όλους τους κόμβους με κοινό path. Ο τύπος των δεικτών αυτών εξαρτάται με τον τρόπο αποθήκευσης του XML εγγράφου
- Το ευρετήριο αυτό είναι συνήθως πολύ μικρό σε μέγεθος και φορτώνεται στην μνήμη. Έτσι όταν έχουμε ένα query, κάνουμε αναζήτηση στο ευρετήριο, βρίσκουμε τους επιθυμητούς κόμβους (μέσω των δεικτών) και τους επιστρέφουμε

## Σύστημα Αρίθμησης

---

- Πολλά συστήματα επεξεργασίας XML εγγράφων, χρησιμοποιούν (σε συνδυασμό με ευρετήρια) κάποια συστήματα αρίθμησης των κόμβων ενός XML εγγράφου
- Γρήγορη εύρεση των δοκιμών σχέσεων μεταξύ δύο κόμβων
- Δύο κύριες κατηγορίες
  - Αριθμητικά συστήματα
  - Prefix-based συστήματα
- Αριθμητικά συστήματα
  - Βασίζονται κυρίως στις preorder και postorder labels του κάθε κόμβου(με διάφορες παραλλαγές)
- Prefix-based συστήματα
  - Βασίζονται στην ιδέα ότι η label ενός κόμβου είναι η label του πατέρα του + <κάτι άλλο>

## Σύστημα Αρίθμησης: Παράδειγμα

---

- Σε κάθε κόμβο αναθέτουμε ένα ζεύγος τιμών  $\langle \text{order}, \text{size} \rangle$
- Το  $\text{order}$  είναι ανάλογο της preorder σειρά του κόμβου, ενώ το  $\text{size}$  είναι τέτοιο ώστε ο κόμβος να περιέχει όλους τους απογόνους του
- Έστω δύο κόμβοι  $x, y$ . Τότε ο  $x$  είναι πρόγονος του  $y$ , αν και μόνο αν

$$\text{order}(x) < \text{order}(y) + \text{size}(y) < \text{order}(x) + \text{size}(x)$$

<Βάλε εικόνα>

## Ενότητα 15

---

### Εξόρυξη Γνώσης (Data Mining)

Δρ. Τιμπίρης Αλκιβιάδης



---

## Περιεχόμενα ενότητας

---

- Εξόρυξη Δεδομένων (DATA MINING)
- KDD ΚΑΙ DATA MINING
- Μέθοδοι DATA MINING
- Συσταδοποίηση (CLUSTERING)
- Ιεραρχικό, Διαμεριστικό ΚΑΙ Ασαφές CLUSTERING
- Διαμεριστικοί Αλγόριθμοι
- K-means clustering



## Σκοποί ενότητας

---

Ο όρος εξόρυξη δεδομένων αναφέρεται στην διαδικασία ανάλυσης μεγάλων βάσεων δεδομένων για εύρεση χρήσιμων μοτίβων. Σχετίζεται με την ανακάλυψη της γνώσης σε βάσεις δεδομένων. Στην ενότητα αυτή γίνεται μια αναφορά σε βασικές τεχνικές εξόρυξης δεδομένων όπως συσταδοποίηση, κατηγοριοποίηση, δέντρα αποφάσεων και κανόνες συσχέτισης. Παρουσιάζονται στον σπουδαστή νέα επιστημονικά πεδία ερευνητικής και επαγγελματικής δράσης και γίνεται επεξήγηση αντιπροσωπευτικών αλγορίθμων.



## Εισαγωγή στην KDD

---

- Στις μέρες μας και σε ένα πολύ μεγάλο εύρος πεδίων η συλλογή, καταγραφή και επεξεργασία δεδομένων γίνεται με δραματικούς ρυθμούς. Δημιουργούνται συνεχώς θεωρίες και εργαλεία τα οποία βοηθούν στην συλλογή χρήσιμων πληροφοριών από τα ολοένα αυξανόμενα δεδομένα. Αυτές οι θεωρίες και τα εργαλεία είναι το αντικείμενο της Ανακάλυψης Γνώσης σε Βάσεις Δεδομένων γνωστή και ως KDD (Knowledge Discovery in Databases). Η KDD διαδικασία ασχολείται με την δημιουργία και μελέτη μεθόδων και τεχνικών που εφαρμόζονται στα δεδομένα με σκοπό την εξαγωγή χρήσιμων πληροφοριών.





## Χρήση των τεχνικών KDD

---

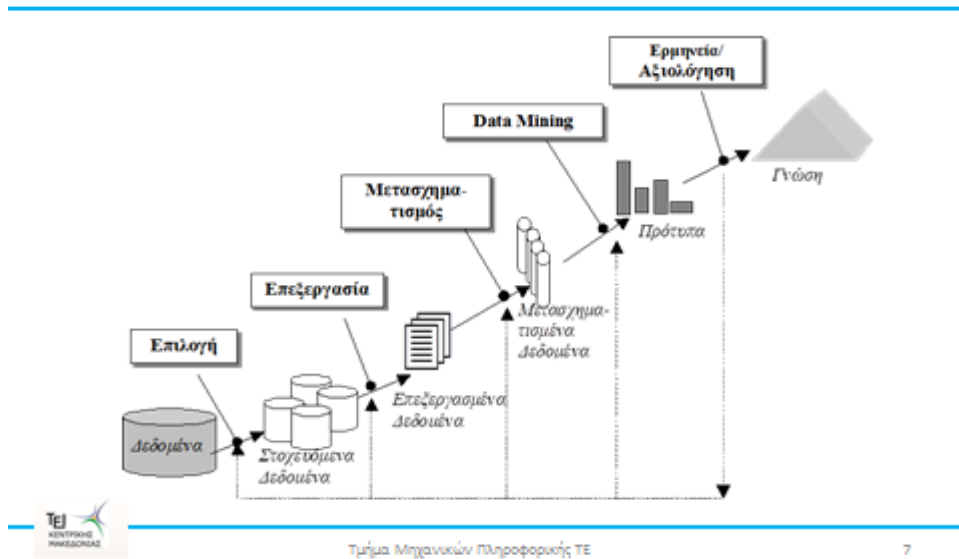
- Marketing
- Επενδύσεις (Investment)
- Fraud Detection
- Επικοινωνίες και Data Cleaning
- Ιατρική

## KDD - Data Mining

---

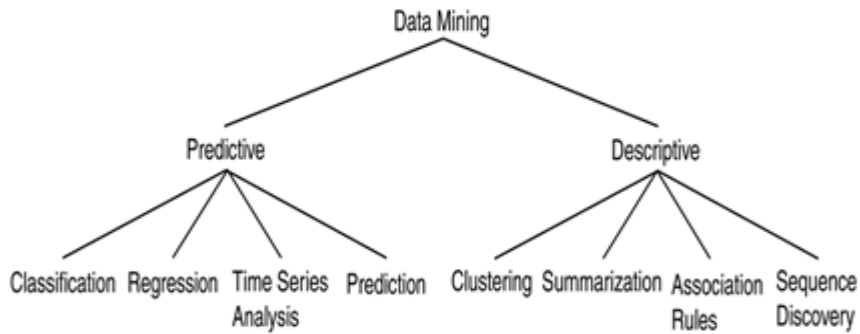
- Γενικά ο όρος KDD αναφέρεται στη συνολική διαδικασία την εύρεσης γνώσης από σύνολα δεδομένων και ο όρος Data Mining αναφέρεται σε ένα μέρος αυτής της διαδικασίας. Data Mining είναι η εφαρμογή συγκεκριμένου αλγορίθμου για την εξαγωγή μοντέλου από τα δεδομένα.
- Επιπλέον βήματα στην KDD διαδικασία είναι: προετοιμασία των δεδομένων, επιλογή δεδομένων, καθαρισμός δεδομένων (data cleaning), ενσωμάτωση κατάλληλης γνώσης και σωστή μετάφραση και μελέτη των αποτελεσμάτων

# KDD - Data Mining



7

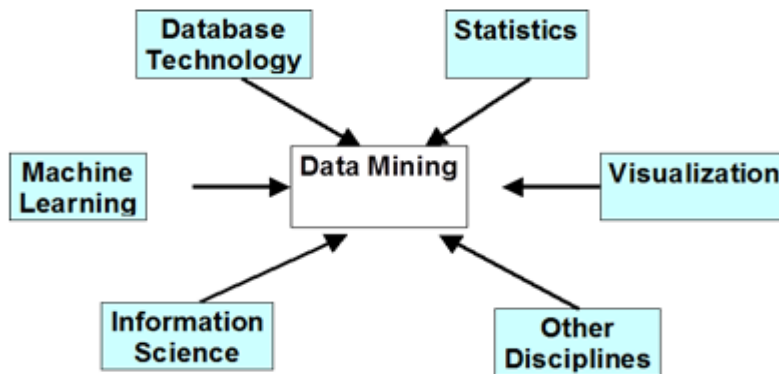
# Data Mining



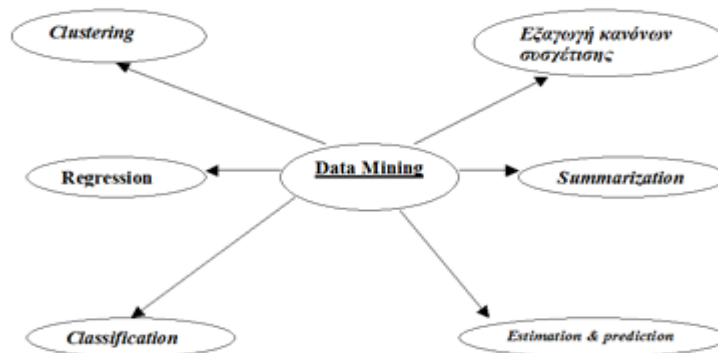
Τμήμα Μηχανικών Πληροφορικής ΤΕ

8

# Data Mining



# Data Mining



## Συσταδοποίηση- κατηγοριοποίηση

---

### *Clustering (συσταδοποίηση)*

Το clustering είναι η εργασία του μερισμού ενός συνόλου δεδομένων σε ομάδες ομοίων στοιχείων, clusters. Τα δεδομένα ομαδοποιούνται σε σύνολα με βάση κάποιο κριτήριο ομοιότητας. Το clustering δεν βασίζεται σε προκαθορισμένες κλάσεις.

### *Classification (κατηγοριοποίηση)*

Η διαδικασία κατηγοριοποίησης των δεδομένων σε κάποια από τις προκαθορισμένες κλάσεις. Συχνά η διαδικασία του classification περιγράφεται σαν μία συνάρτηση μάθησης (learning function), η οποία ταξινομεί (classifies) κάθε αντικείμενο του συνόλου δεδομένων σε μία από τις προκαθορισμένες κατηγορίες. Η διαδικασία του classification χαρακτηρίζεται από: Ένα σύνολο καλά ορισμένων κατηγοριών, ένα training set . **Στόχος:** Ο ορισμός ενός μοντέλου το οποίο μπορεί να κατηγοριοποιεί νέα δεδομένα από ένα test set



## Κανόνες συσχέτισης - εκτίμηση και πρόβλεψη

---

### *Εξαγωγή κανόνων συσχέτισης (association rules extraction)*

Προσδιορισμός και εξαγωγή των συσχετίσεων ή προτύπων τα οποία υπάρχουν σε μία συλλογή αντικειμένων. Τα πρότυπα μπορούν να εκφραστούν με κανόνες, των οποίων η γενική μορφή είναι *"If X then Y"*. Κριτήρια εγκυρότητας και σημαντικότητας κανόνων: *support factor, confidence factor*

### *Estimation & prediction (εκτίμηση και πρόβλεψη).*

Περιλαμβάνει τεχνικές εκτίμησης και πρόβλεψης μελλοντικών τάσεων ή τιμών. Ο στόχος εδώ είναι να κατασκευάσουμε ένα μοντέλο που θα επιτρέπει την τιμή μιας μεταβλητής να προβλεφθεί από τις γνωστές τιμές άλλων μεταβλητών



## Παλινδρόμηση - Συνάθροιση

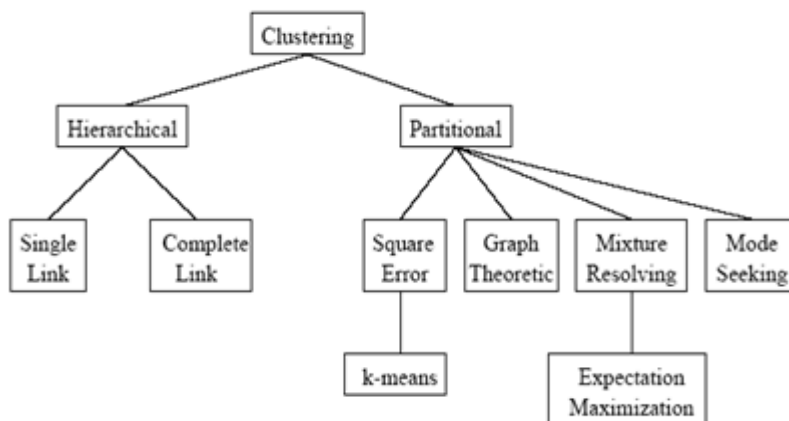
### *Regression (παλινδρόμηση).*

Αντιστοιχεί τα αντικείμενα από ένα σύνολο δεδομένων στην τιμή μίας μεταβλητής πρόβλεψης

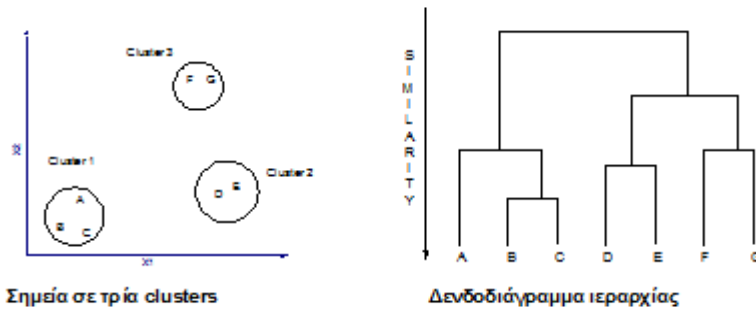
### *Summarization (Συνάθροιση)*

Περιλαμβάνει μεθόδους για την περιγραφή ενός υποσυνόλου δεδομένων. Π.χ. η εκτίμηση της μέσης και της τυπικής απόκλισης για όλα τα πεδία, geronts, τεχνικές παρουσίασης, την παραγωγή συνοπτικών κανόνων.

## Clustering

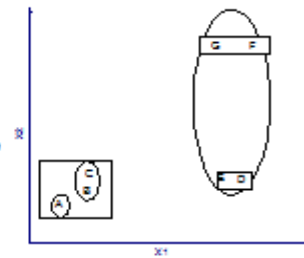


# Clustering



## *K-means clustering*

1. Επιλογή  $k$  κεντροειδών cluster τα οποία αποτελούν και τα μόνα στοιχεία των  $k$  επελεγμένων clusters.
2. Τοποθέτησε κάθε στοιχείο στο πιο κοντινό cluster μετά από υπολογισμό της απόστασης του σημείου από το κεντροειδές του cluster.
3. Υπολόγισε το νέο κεντροειδές.
4. Αν το κριτήριο τερματισμού δεν ικανοποιείται πήγαινε στο βήμα 2.



Ευαισθησία του αλγορίθμου  $k$ -means στην αρχική επιλογή clusters.



## Ενότητα 5 – Εργαστηριακές ασκήσεις του μαθήματος «Ειδικά Θέματα Βάσεων Δεδομένων»

### Άσκηση 1- Εισαγωγή στον SQL Server 2008

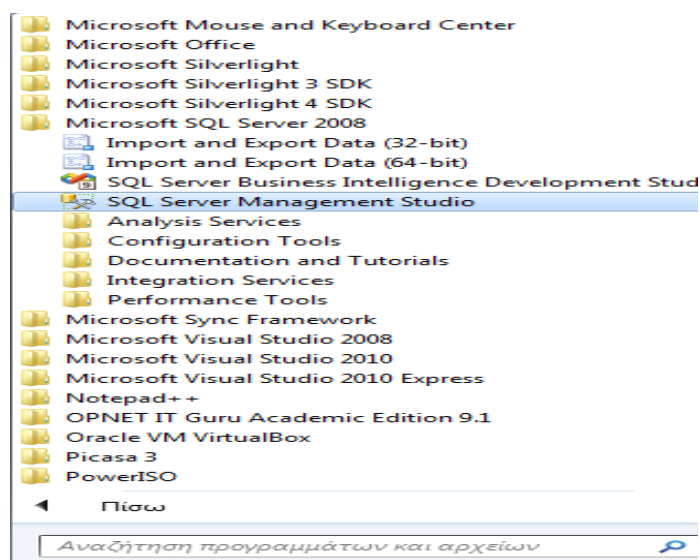
Σκοπός αυτής της άσκησης είναι η εξοικείωση σας με το περιβάλλον εργασίας του Microsoft SQL Server 2008. Σε αυτήν την άσκηση θα ασχοληθείτε κυρίως με τη δημιουργία μιας Βάσης Δεδομένων, τη δημιουργία ερωτημάτων SQL, την εξαγωγή και εισαγωγή αντιγράφων ασφαλείας της βάσης καθώς επίσης και με τη μετατροπή και εξαγωγή των δεδομένων σε άλλο RDBMS\* (π.χ. Microsoft Access).

Το περιβάλλον που θα χρησιμοποιήσετε για την δημιουργία και την διαχείριση της βάσης το **SQL Server Management Studio**.



Εικόνα 82

Το περιβάλλον αυτό βρίσκεται μέσα στον φάκελο του **SQL Server 2008** τον οποίο θα τον βρείτε εάν πάτε στην **Έναρξη**→**Όλα τα προγράμματα**→**SQL Server 2008**.



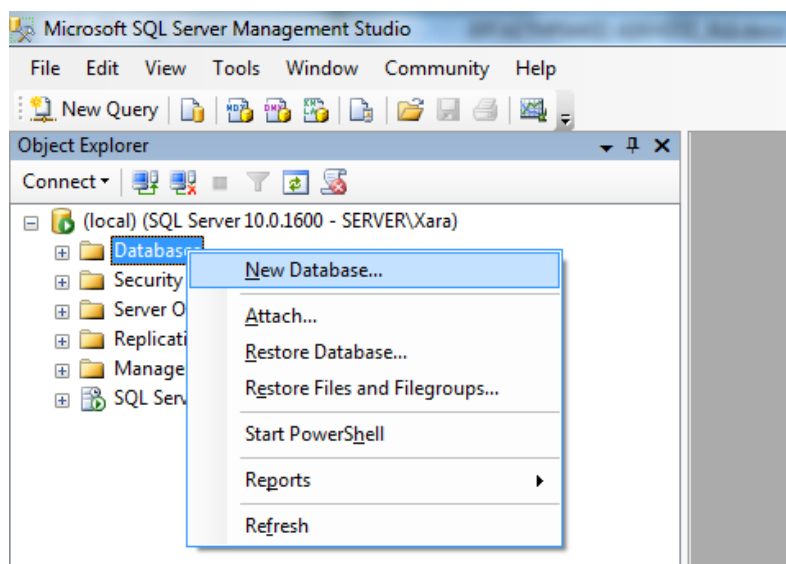
Εικόνα 83

Αφού μπειτε στο περιβάλλον θα σας ανοίξει η Εικόνα 64 στην οποία πρέπει να γίνει η σύνδεση. Πατώντας **Connect** αρχίζει η υπηρεσία του SQL Server και μπορείτε τώρα να δημιουργήσετε την δική σας βάση.



Εικόνα 84

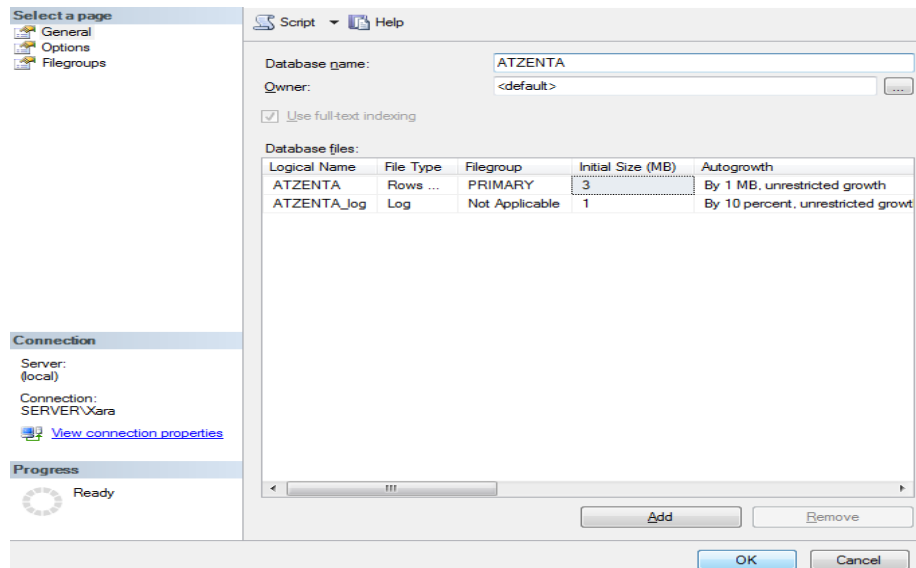
- Για την δημιουργία μίας νέας βάσης πηγαίνετε στον **Object Explorer**  
→ **Databases** → **δεξί κλικ** → **New Database**.



Εικόνα 85

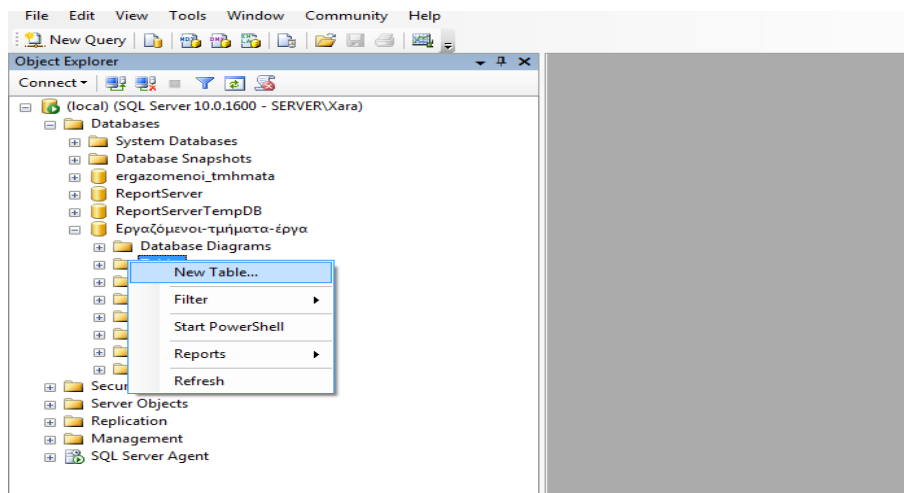
- Δώστε όνομα στην βάση (ATZENTA) και έπειτα **OK**.





Εικόνα 86

- Για την δημιουργία ενός πίνακα πηγαίνετε **Object Explorer** → ανοίξτε την δενδρική δομή του **Databases** → ομοίως για την βάση ANTZENTA → Τέλος **New Table**.



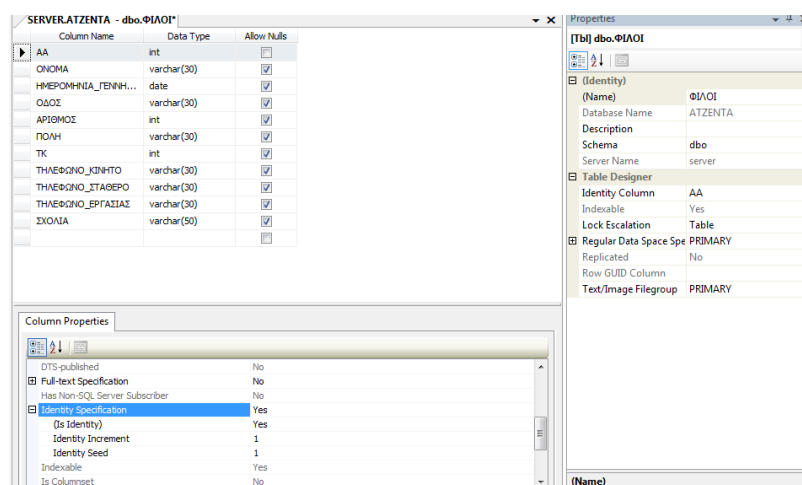
Εικόνα 87

- Όταν ανοίξει ο πίνακας δώστε του όνομα ΦΙΛΟΙ πηγαίνοντας δεξιά στο Properties.

Τα πεδία που θα εμπεριέχει θα είναι τα εξής: ΑΑ (INT), ΟΝΟΜΑ (VARCHAR(30)), ΕΠΙΘΕΤΟ (VARCHAR(30)), ΗΜΕΡΟΜΗΝΙΑ\_ΓΕΝΝΗΣΗΣ (DATE), ΟΔΟΣ (VARCHAR(30)), ΑΡΙΘΜΟΣ(INT), ΠΟΛΗ(VARCHAR(30)), ΤΚ(INT), ΤΗΛΕΦΩΝΟ\_ΚΙΝΗΤΟ (VARCHAR(30)), ΤΗΛΕΦΩΝΟ\_ΣΤΑΘΕΡΟ (VARCHAR(30)), ΤΗΛΕΦΩΝΟ\_ΕΡΓΑΣΙΑΣ (VARCHAR(30)), ΣΧΟΛΙΑ (VARCHAR(50)).

**ΣΧΟΛΙΟ!** Παρατηρήστε ότι στα πεδία ΤΗΛ\_ΚΙΝΗΤΟ και ΤΗΛ\_ΣΤΑΘΕΡΟ, ΤΗΛ\_ΕΡΓΑΣΙΑΣ έχει τύπο δεδομένων Varchar() παρόλο που τα πεδία αυτά αντικειμενικά δέχονται ακεραίους ως τιμές και θα έπρεπε να έχουν τύπο δεδομένων int. Αυτό συμβαίνει γιατί τα πεδία αυτά δεν θα τα χρησιμοποιήσετε για εύρεση μέσου όρου, αθροίσματος και γενικά για αριθμητικές πράξεις οπότε ως int δεν θα σας χρειαστούν.

- Για την εισαγωγή του πρωτεύοντος κλειδιού στο πεδίο ΑΑ πηγαίνετε δεξιά του και δεξί κλικ → Πρωτεύον κλειδί. Έπειτα πηγαίνετε κάτω στο Column Properties και ανοίγετε την δενδρική δομή του Identity Specification και στο (Is identity) κάνετε την τιμή από No σε Yes και το Identity Increment=1 για να έχετε αυτόματη αρίθμηση.



Εικόνα 88

**ΣΧΟΛΙΟ! Αυτόματη Αρίθμηση πεδίου.** Το πρώτο πεδίο του πίνακα ΦΙΛΟΙ το ΑΑ, θέλουμε να έχει αυτόματη αρίθμηση. Να ξεκινάει με τη τιμή 1 από την πρώτη εγγραφή και να συνεχίζει με βήμα 1 στις υπόλοιπες. Αυτό επιτυγχάνεται αλλάζοντας τη ρύθμιση “Identity” όπως φαίνεται και στην Εικόνα 7 σε “ Yes (Not for Replication)”. Από ποια εγγραφή θα ξεκινάει η αρίθμηση εξαρτάται από το πεδίο “Identity Seed”, στην περίπτωση μας 1 και με τι βήμα θα αυξάνεται η αρίθμηση ρυθμίζεται στο πεδίο “Identity Increment”, στο οποίο για τον συγκεκριμένο πίνακα δίνετε τιμή 1.

Οι τύποι δεδομένων που μπορεί να πάρει ένα πεδίο στον **Sql Server** είναι πολλοί και θα μπορούσαν να χωριστούν και σε κατηγορίες όπως θα δείτε παρακάτω:

#### Τύποι για ημερομηνίες

Date	Datetimeoffset
Datetime2	Smalldatetime
Datetime	Time

#### Τύποι για νούμερα

Bigint	Numeric
Bit	Smallint
Decimal	Smallmoney
Int	Tinyint
Money	

#### Τύποι προσεγγιστικών τιμών

Float	Real
-------	------

#### Τύποι για χαρακτήρες

Char	Varchar
Text	

#### Τύποι για χαρακτήρες με διεθνή κωδικοποίηση

Nchar	Nvarchar
Ntext	

#### Τύποι για δυαδικές συμβολοσειρές

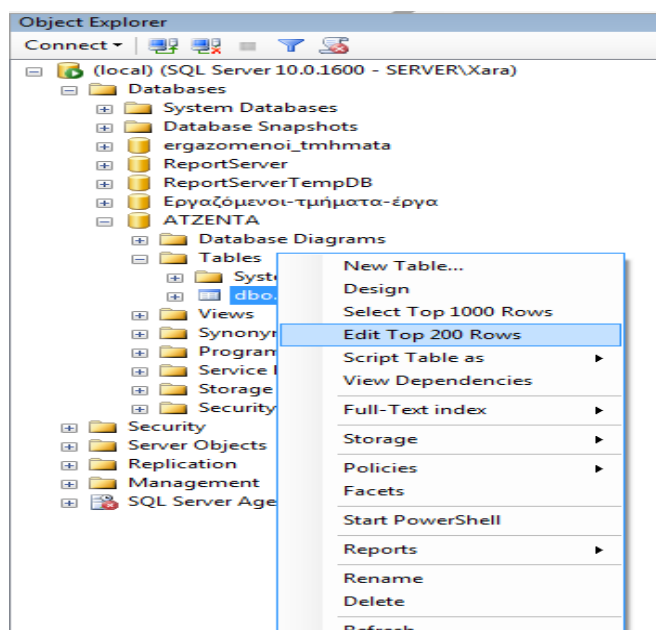
Binary	Varbinary
Image	

#### Άλλοι τύποι δεδομένων

Cursor	Table
Hierarchyid	Timestamp
Sql_variant	Uniqueidentifier
Xml	Spatial types

Αφού δημιουργηθεί ο πίνακας για την **καταχώρηση τιμών** σε αυτών κάνετε το εξής:

**Object Explorer**→**Databases**→**Tables**→**δεξί κλικ στον πίνακα ΦΙΛΟΙ**→**Edit top 200 rows**.



Εικόνα 89

Με την ολοκλήρωση αυτής της διαδικασίας ανοίγει ο πίνακας καταχώρησης τιμών. Οι τιμές που θα καταχωρήσετε στα πεδία ακολουθούν στους παρακάτω πίνακες.

ΑΑ	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	ΟΔΟΣ	ΑΡΙΘΜΟΣ
1	Γεωργίου	Απόστολος	7/4/1997	Βενιζέλου	12
2	Παπαδόπουλος	Γεώργιος	4/5/1999	Μεραρχίας	7
3	Μιχαηλίδης	Αντώνιος	4/5/1997	Δελφών	34
4	Καραλής	Ανδρέας	3/3/1984	Κηφισίας	67
5	Τριανταφύλλου	Γρηγόριος	7/8/1997	Δωδώνης	90

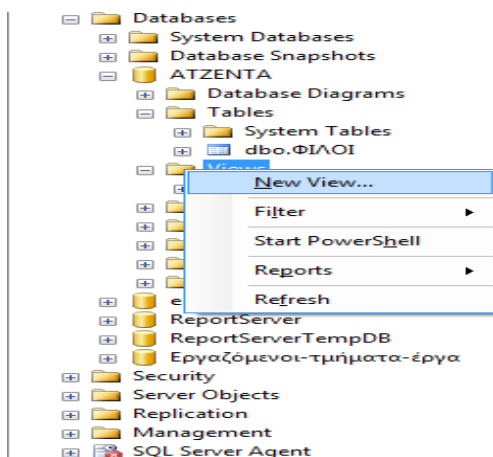
ΤΚ	ΠΟΛΗ	ΤΗΛ_ΚΙΝΗΤΟ	ΤΗΛ_ΣΤΑΘΕΡΟ	ΤΗΛ_ΕΡΓΑΣΙΑΣ	ΣΧΟΛΙΑ
62100	ΣΕΡΡΕΣ	23210-73234	23210-74321	6973434343	ΦΙΛΟΣ
62100	ΣΕΡΡΕΣ	23210-45667	23210-56789	6989032455	ΦΙΛΟΣ
45678	ΘΕΣ/ΝΙΚΗ	2310-554566	2310-567890	6945342312	ΓΝΩΣΤΟΣ
12345	ΑΘΗΝΑ	210-3456790	210-2345678	6967345673	ΓΝΩΣΤΟΣ
65100	ΙΩΑΝΝΙΝΑ	26510-32145	26510-89076	6978987654	ΦΙΛΟΣ

**ΣΧΟΛΙΟ!** Το πεδίο **ΑΑ** συμπληρώνετε αυτόματα.

## Δημιουργία Όψεων

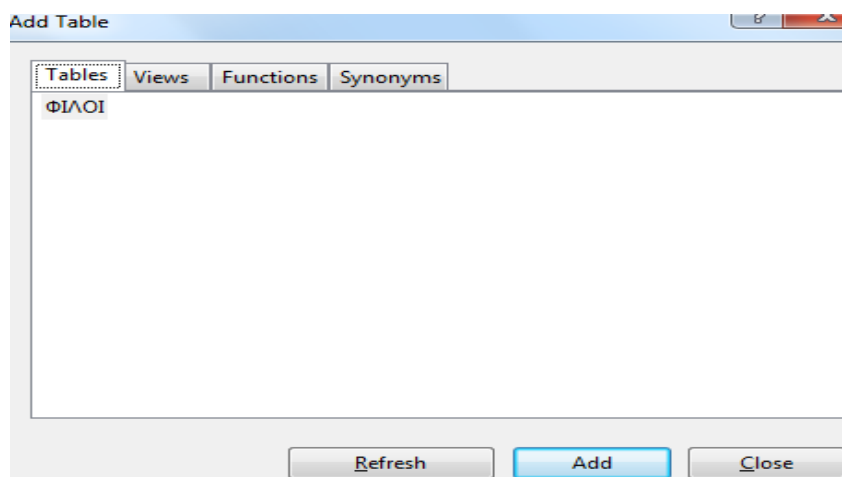
Για την δημιουργία όψεων η διαδικασία έχει ως εξής:

- **Object Explorer**→**Databases**→**ATZENTA** (ανοίξτε την δενδρική δομή)→**Views**→δεξί κλικ →**New view**.




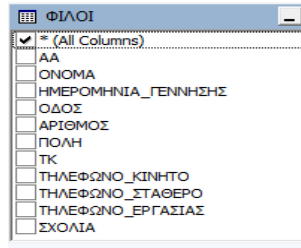
Εικόνα 90

- Έπειτα ανοίγει το παράθυρο της Εικόνας 10 στο οποίο πρέπει να επιλέξετε τον πίνακα για τον οποίο θα δημιουργηθεί η όψη και **ADD**.



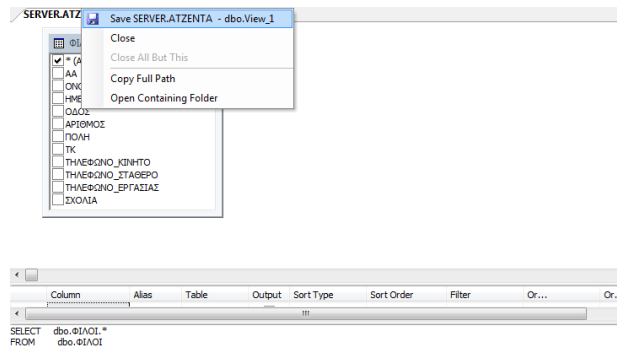
Εικόνα 91

- Επιλέγετε τα πεδία που θέλετε να εμφανίζοντε από τον πίνακα ΦΙΛΟΙ και εκτέλεση . Στην άσκηση ATZENTA επιλέξτε όλα τα πεδία.

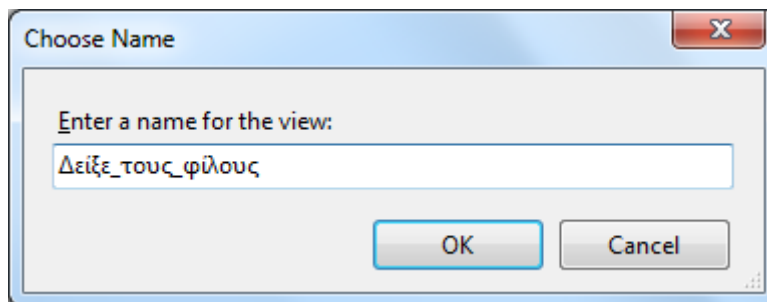


Εικόνα 92

- Έπειτα αποθηκεύστε την όψη με όνομα **Δείξε\_τους\_φίλους** πηγαίνοντας στο **Server.ATZENTA.dbo.View\_1**→**Δεξί κλικ**→**Save**.

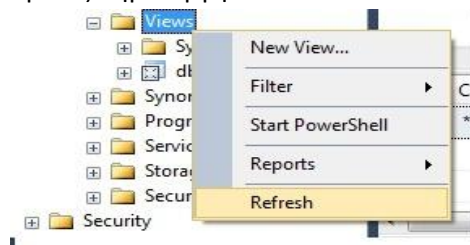


Εικόνα 93



Εικόνα 94

- Αφού το αποθηκεύσετε πηγαίνετε στα **Views**→**Δεξί κλικ**→**Refresh** για να σας εμφανίσει το view που μόλις δημιουργήσατε.



Εικόνα 95

- Τέλος για να το δείτε δεξί κλικ στο **dbo .Δείξε\_τους\_φίλους**→**Select top 1000 rows**.

Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...	Or...
AA		ΦΙΛΟΙ	<input checked="" type="checkbox"/>						
ΟΝΟΜΑ		ΦΙΛΟΙ	<input checked="" type="checkbox"/>						
ΕΠΙΘΕΤΟ		ΦΙΛΟΙ	<input checked="" type="checkbox"/>						

AA	ΟΝΟΜΑ	ΕΠΙΘΕΤΟ	ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ	ΟΔΟΣ	ΑΡΙΘΜΟΣ	ΠΟΛΗ	ΤΚ
1	Απόστολος	Γεωργίου	1997-04-07	Βενιζέλου	12	ΣΕΡΡΕΣ	62100
2	Γεώργιος	Παπαδόπουλος	1999-05-04	Μεραρχός	7	ΣΕΡΡΕΣ	62100
3	Αντώνιος	Μιχαηλίδης	1997-05-04	Δελφών	34	ΘΕΣΣΑΛΟΝΙΚΗ	45678
4	Ανδρέας	Καραλής	1984-03-03	Κηφισίας	67	ΑΘΗΝΑ	12345
5	Γρηγόριος	Τριανταφύλλου	1997-08-07	Δωδώνης	90	ΙΩΑΝΝΙΝΑ	65100

Εικόνα 96

Για την άσκηση ATZENTA να εκτελεστούν και να αποθηκευτούν οι παρακάτω όψεις:

### Ερώτηση 1

Να εμφανιστούν όλα τα στοιχεία των φίλων, που το επίθετό τους ξεκινάει από "Κ".

### Απάντηση 1

```
CREATE VIEW VIEW_ONOMA AS (SELECT *
FROM ΦΙΛΟΙ
WHERE ΕΠΙΘΕΤΟ LIKE ' Κ% ');
```

### Ερώτηση 2

Να εμφανιστούν το Επίθετο, το Όνομα και το Κινητό των φίλων που γεννήθηκαν μετά το 1997 και είναι από την πόλη Σέρρες.

### Απάντηση 2

```
CREATE VIEW VIEW_ONOMA AS (SELECT ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛΕΦΩΝΟ_ΚΙΝΗΤΟ
FROM ΦΙΛΟΙ
WHERE (ΠΟΛΗ = 'ΣΕΡΡΕΣ') AND (ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ >= '1/1/1998'));
```

### Ερώτηση 3

Να γίνει ομαδοποίηση των φίλων ανά Πόλη και να καταμετρηθεί πόσοι είναι από κάθε Πόλη.

### Απάντηση 3

```
CREATE VIEW VIEW_ONOMA AS (SELECT ΠΟΛΗ, Count(AA) As ΠΛΗΘΟΣ
FROM ΦΙΛΟΙ
Group By ΠΟΛΗ);
```

#### Ερώτηση 4

Να γίνει ομαδοποίηση των φίλων ανά Σχόλιο και να καταμετρηθεί πόσοι είναι φίλοι και πόσοι γνωστοί.

#### Απάντηση 4

```
CREATE VIEW VIEW_ONOMA AS (SELECT ΣΧΟΛΙΑ, Count(AA) As ΠΛΗΘΟΣ  
From ΦΙΛΟΙ  
Group By ΣΧΟΛΙΑ)
```

#### Ερώτηση 5

Να εμφανιστούν τα στοιχεία των φίλων που κατάγονται από την ίδια πόλη που κατάγεται ο Γεωργίου. (ΕΜΦΩΛΕΥΜΕΝΟ).

**ΣΧΟΛΙΟ!** Η SQL Επιτρέπει υπό-ερωτήματα μέσα σε κάποιο ερώτημα. Αυτά αναφέρονται ως εμφωλευμένα.

#### Απάντηση 5

```
CREATE VIEW VIEW_ONOMA AS (SELECT *  
FROM ΦΙΛΟΙ  
WHERE (ΠΟΛΗ =  
(SELECT ΠΟΛΗ  
FROM ΦΙΛΟΙ  
WHERE ΕΠΙΘΕΤΟ = 'Γεωργίου')));
```

#### Ερώτηση 6

Να εμφανιστούν τα στοιχεία των φίλων που κατάγονται από Αθήνα και Θεσσαλονίκη.

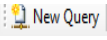
#### Απάντηση 6

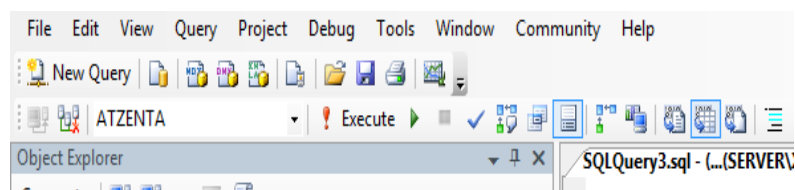
```
CREATE VIEW VIEW_ONOMA AS (SELECT *  
FROM ΦΙΛΟΙ  
WHERE (ΠΟΛΗ = 'ΑΘΗΝΑ') OR (ΠΟΛΗ = 'ΘΕΣΣΑΛΟΝΙΚΗ'));
```



## Εκτέλεση Ερωτημάτων (Queries)

Για την δημιουργία ενός ερωτήματος κάνετε τα εξής:

- Πηγαίνετε στην κορδέλα και πατάτε **New Query** .
- Στο παράθυρο που ανοίγει γράφετε τον κώδικα **SQL** που θέλετε να εκτελέσετε και πατάτε **EXECUTE**. Προσέξτε να επιλέξετε την **Βάση που σας αφορά**.

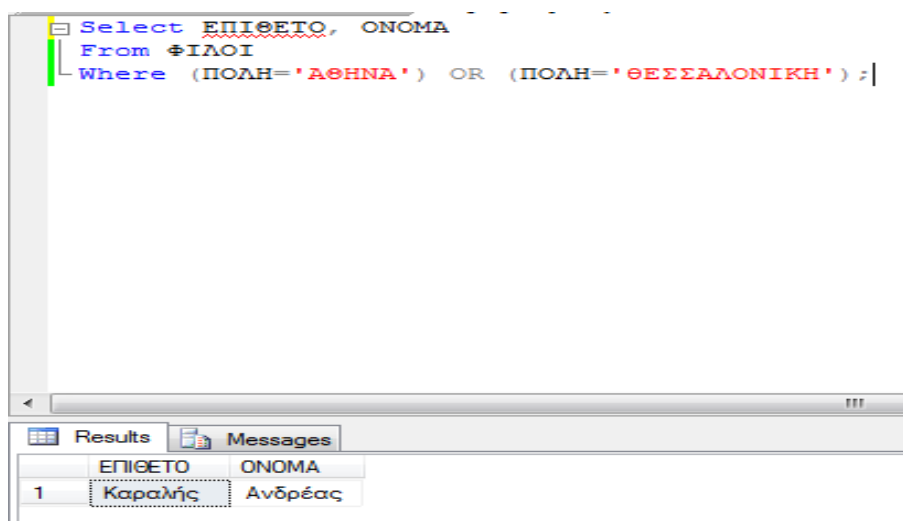


Εικόνα 97

Για εξάσκηση εκτελέστε το παρακάτω ερώτημα:

### Ερώτημα 1

Εμφανίστε το ονοματεπώνυμο των φίλων που κατάγονται από **ΑΘΗΝΑ** και των φίλων που κατάγονται από **ΘΕΣΣΑΛΟΝΙΚΗ**.

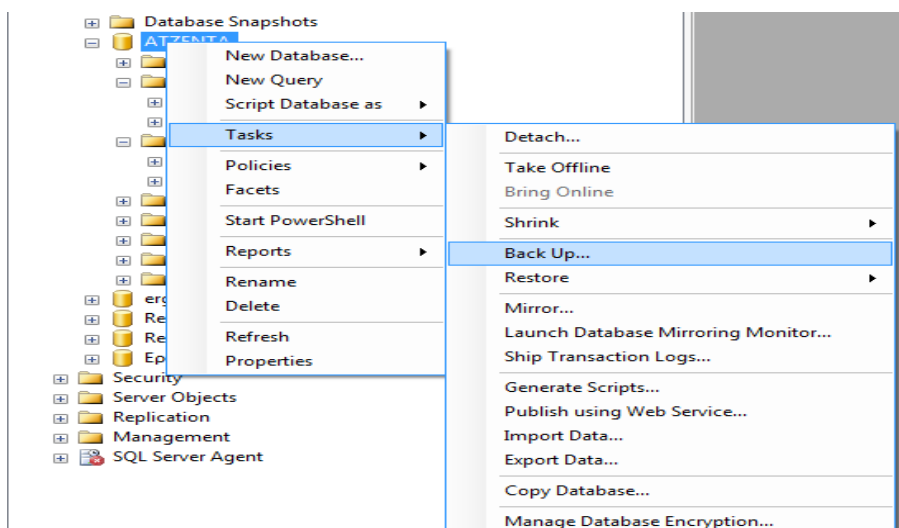


Εικόνα 98

## Δημιουργία αντιγράφου ασφαλείας και επαναφορά

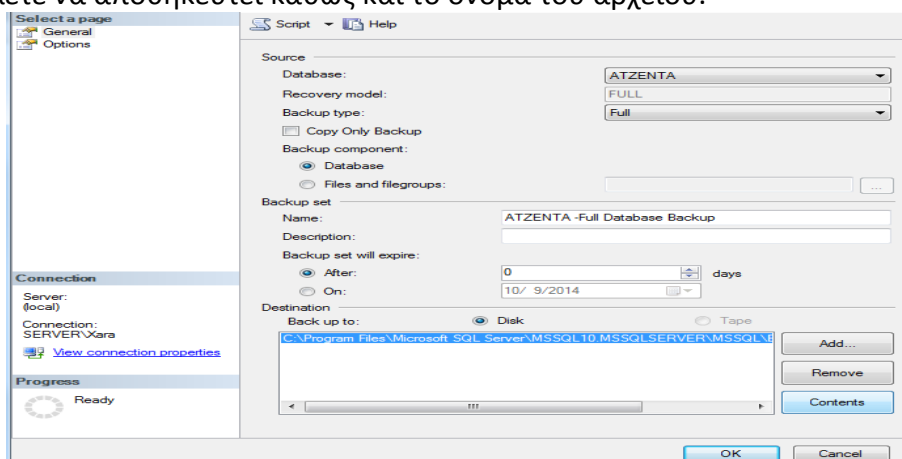
Ένα εργαλείο που θα σας φανεί ιδιαίτερα χρήσιμο στον SQL SERVER είναι η δημιουργία αντιγράφων ασφαλείας της βάσης καθώς και η επαναφορά της. Για την δημιουργία ενός αντιγράφου η διαδικασία έχει ως εξής:

**Δεξί κλικ** στην βάση που θέλετε να κρατήσετε αντίγραφο (Εστω άσκηση ATZENTA)  
→Tasks→Back up.



Εικόνα 99

- Στη συνέχεια εμφανίζεται φόρμα στην οποία μπορείτε να καθορίσετε το όνομα του αντιγράφου ασφαλείας και τη θέση στην οποία θα αποθηκευτεί, πατώντας το πλήκτρο **Add** . Έπειτα επιλέγετε τη διαδρομή του δίσκου που θέλετε να αποθηκευτεί καθώς και το όνομα του αρχείου.

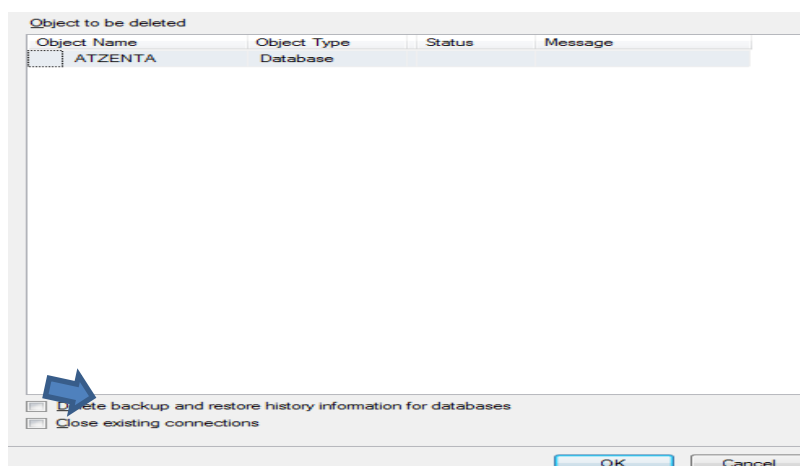


Εικόνα 100

## Επαναφορά αντιγράφου ασφαλείας

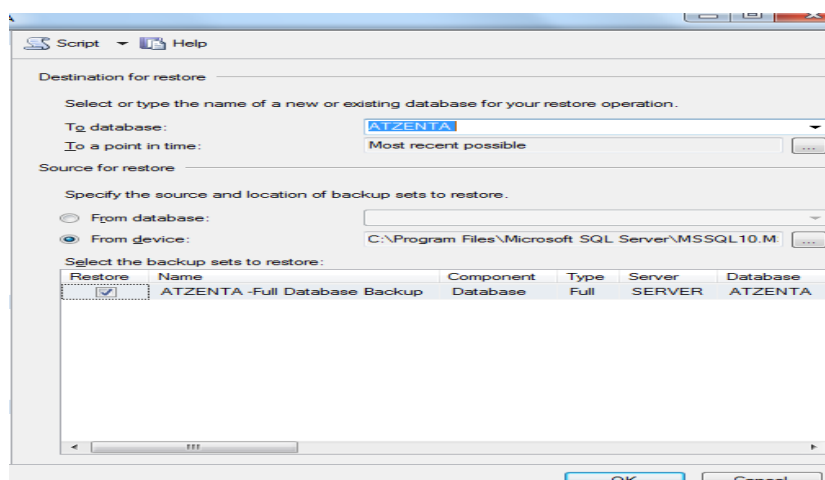
Ένα εξίσου σημαντικό κομμάτι που θα πρέπει να γνωρίζετε για τον SQL SERVER εκτός του να κρατήσετε ένα αντίγραφο ασφαλείας είναι το να ξέρετε και να το επαναφέρετε. Η επαναφορά γίνεται ως εξής:

- Εφόσον είχατε διαγράψει την βάση σας κρατώντας βέβαια το αντίγραφο ασφαλείας σας. Για να μην διαγραφεί το αντιγραφό σας κατά την διαδικασία της διαγραφής της βάσης πρέπει να ξε-τικάρετε την επιλογή που βλέπετε στο κάτω μέρος της Εικόνας



Εικόνα 101

- Στην συνέχεια πηγαίνετε στο Object Explorer Databases → Restore Database. Στο παράθυρο βρίσκουμε την τοποθεσία αποθήκευσης του αντιγράφου ασφαλείας και το προθέτουμε (From Device) και τσεκάρετε την επιλογή όπως στην Εικόνα 21, έπειτα επιλέγουμε στο Database την βάση ATZENTA και **OK**.

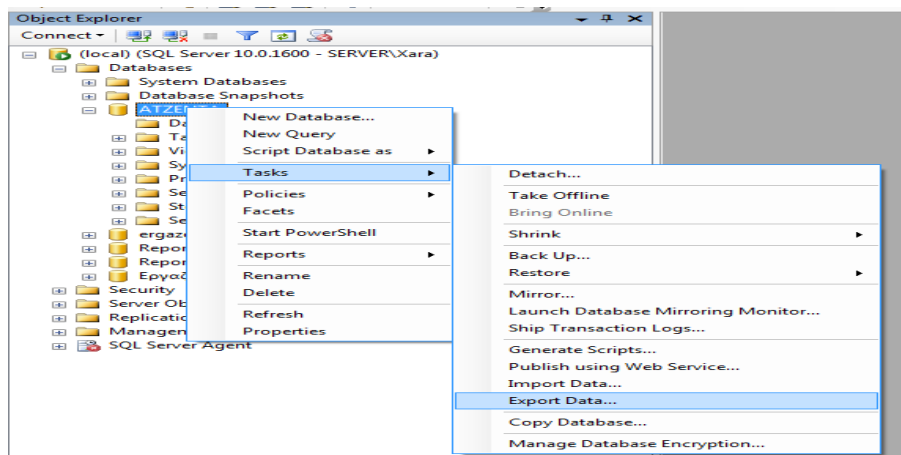


Εικόνα 102

## Εξαγωγή της Βάσης από Microsoft SQL Server σε Microsoft Access

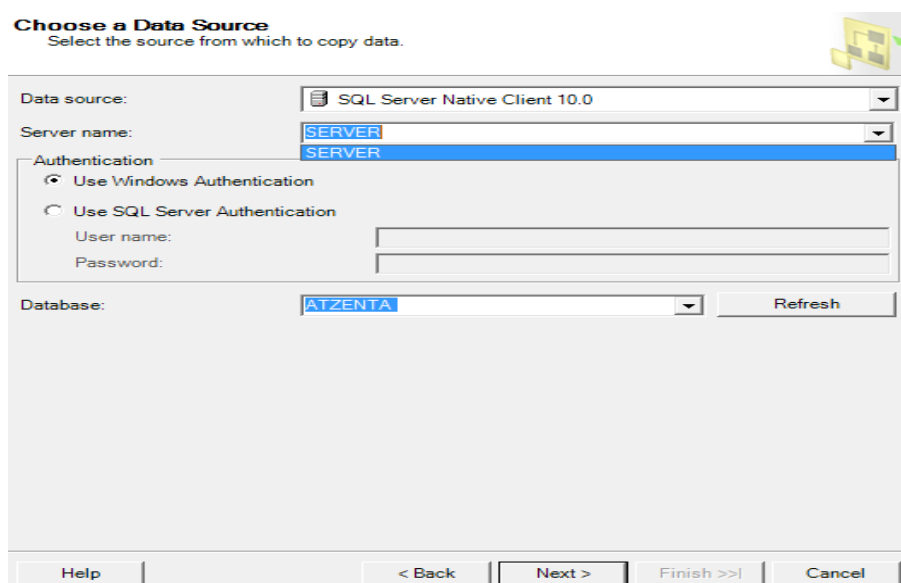
Για να εξαχθεί η βάση σε Microsoft Access, αρκεί να δημιουργήσετε μια κενή βάση σε Access δίνοντας το ίδιο όνομα (ATZENTA). Στη συνέχεια επιστρέφετε στον SQL Server και ακολουθείτε την εξής διαδικασία :

- **Object explorer**→**ATZENTA**→**δεξί κλικ**→**Tasks**→**Export Data**.



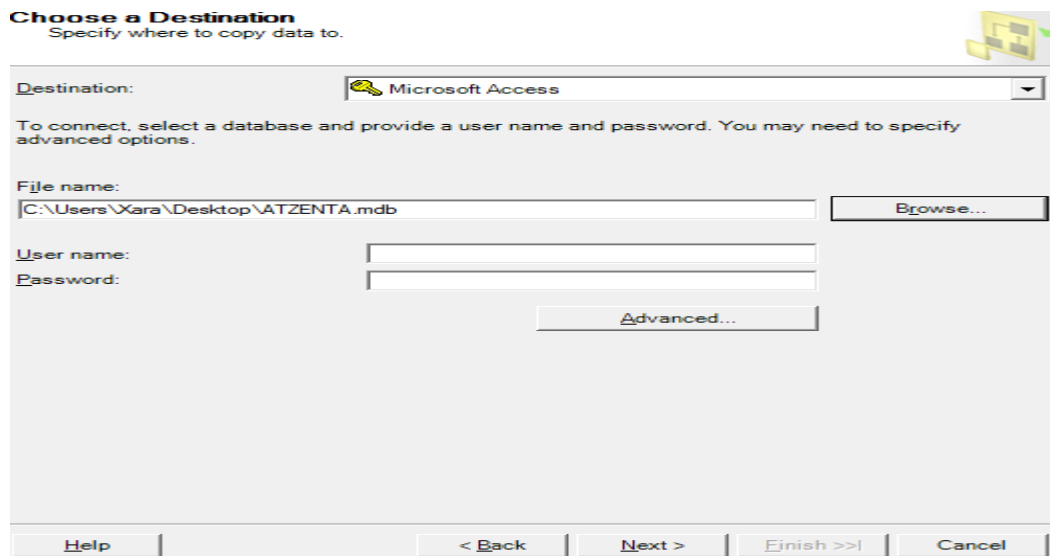
Εικόνα 103

- Στη συνέχεια εμφανίζεται το παράθυρο της Εικόνας 23.Επιλέγετε από πού θα λαμβάνει τα δεδομένα και την βάση που θέλει να κάνει εξαγωγή.



Εικόνα 104

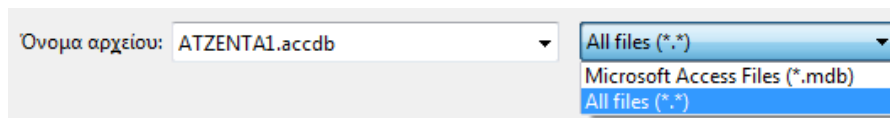
- Αφού γίνει η επιλογή των δεδομένων επιλέγετε τον προορισμό όπου θα γίνεται η εξαγωγή των δεδομένων. Στην τρέχουσα άσκηση θέλετε να κάνετε εξαγωγή των δεδομένων της βάσης ATZENTA στην ACCESS.



Εικόνα 105

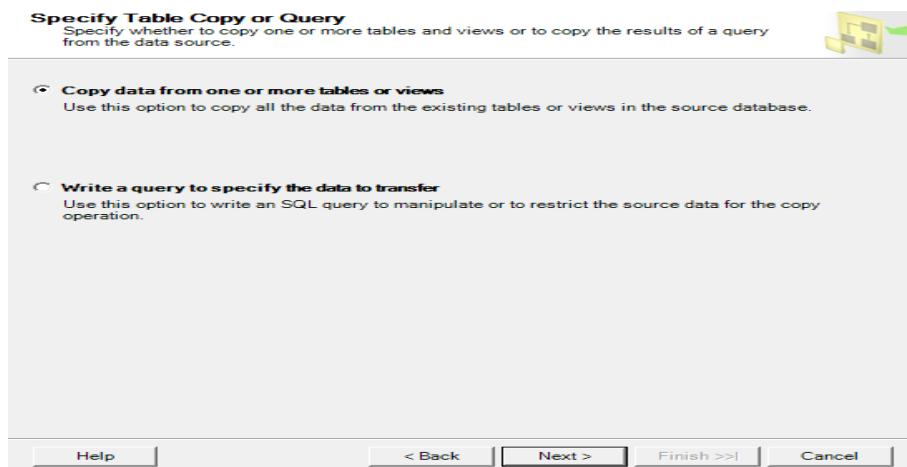
**ΣΧΟΛΙΟ!** Σε περίπτωση που ο υπολογιστής σας είναι 64-bits και δεν έχει αντίστοιχο οδηγό στον SQL SERVER για νέα έκδοση της Microsoft Access, αποθηκεύστε την βάση ATZENTA της Access σε προηγούμενη έκδοση με τον εξής τρόπο: Αρχείο→Αποθήκευση και δημοσίευση→Αποθήκευση Βάσης Δεδομένων Ως →Επιλογή έκδοσης που είναι συμβατός με τον SQL SERVER.

- Πατήστε Browse στο File Name. Αν η βάση έχει επέκταση .accdb επιλέξτε All files στο παράθυρο επιλογής. Αφού επιλέξετε την βάση ATZENTA πατήστε OK και έπειτα Next.



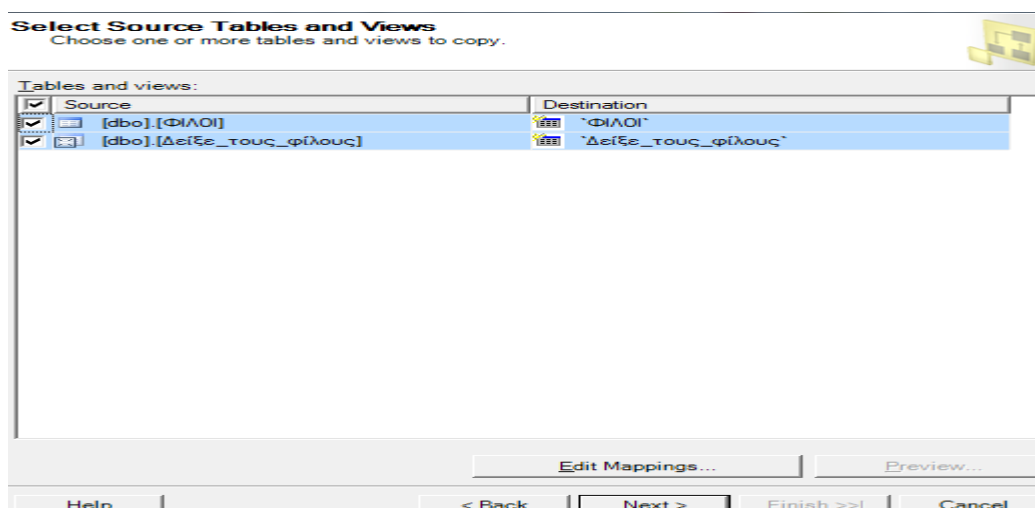
Εικόνα 106

- Αφήστε επιλεγμένο το **Copy data from one or more tables or views**.



Εικόνα 107

- Επιλέξτε τους πίνακες και τα ερωτήματα της βάσης που επιθυμείτε να κάνετε εξαγωγή στην Access. Στην τρέχουσα άσκηση κάντε εξαγωγή και τον πίνακα ΦΙΛΟΙ και το ερώτημα Δείξε\_τους\_φίλους.



Εικόνα 108

➤ Next.

### Review Data Type Mapping

Select a table to review how its data types map to those in the destination. Also, select how the wizard handles conversion issues.

Source	Destination
[dbo].[ΦΙΛΟΙ]	'ΦΙΛΟΙ'
[dbo].[Δείξε_τους_φίλους]	'Δείξε_τους_φίλους'

Source Column	Source Type	Destination Co...	Destination Ty...	Convert	On Error
AA	int	AA	Long		
ΟΝΟΜΑ	varchar	ΟΝΟΜΑ	LongText	<input checked="" type="checkbox"/>	Use Global
ΕΠΙΘΕΤΟ	varchar	ΕΠΙΘΕΤΟ	LongText	<input checked="" type="checkbox"/>	Use Global
ΗΜΕΡΟΜΗΝΙΑ...	date	ΗΜΕΡΟΜΗΝΙΑ...	VarChar	<input checked="" type="checkbox"/>	Use Global
ΟΔΟΣ	varchar	ΟΔΟΣ	LongText	<input checked="" type="checkbox"/>	Use Global
ΑΡΙΘΜΟΣ	int	ΑΡΙΘΜΟΣ	Long		
ΠΟΛΗ	varchar	ΠΟΛΗ	LongText	<input checked="" type="checkbox"/>	Use Global

To view conversion details, double-click the row that contains the column source type to be converted.

On Error (global):

On Truncation (global):

Εικόνα 109

➤ Next.

SQL Server Import and Export Wizard

### Save and Run Package

Indicate whether to save the SSIS package.

Run immediately

Save SSIS Package

SQL Server

File system

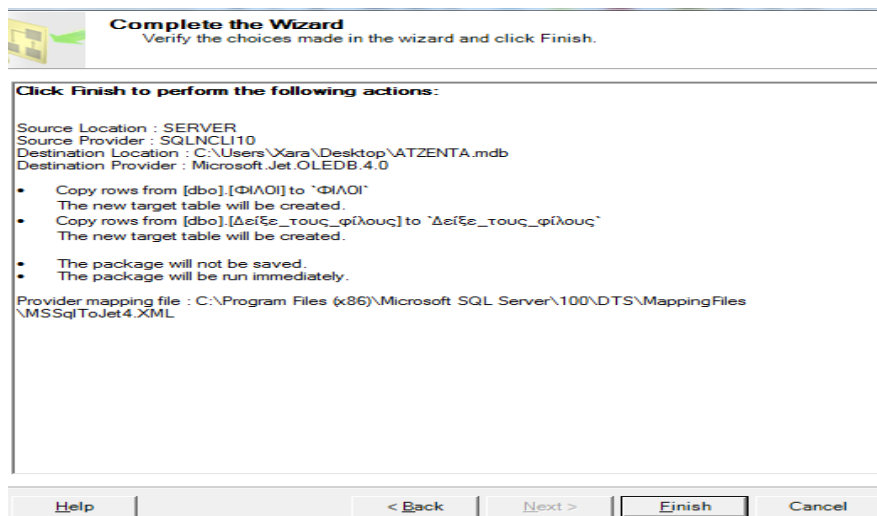
Package protection level:

Password:

Retype password:

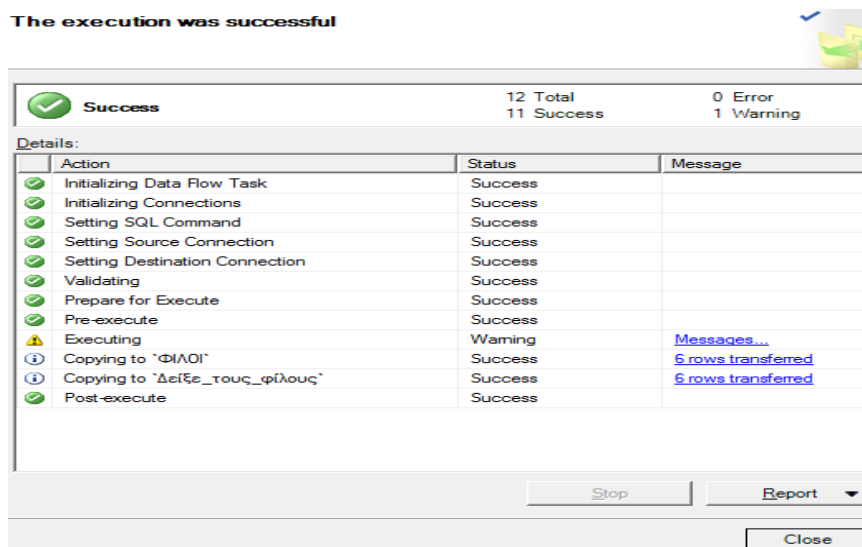
Εικόνα 110

➤ **Finish.**



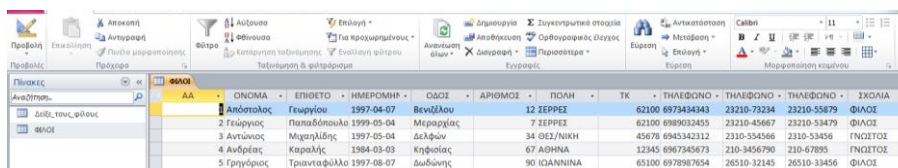
Εικόνα 111

➤ **Close.**



Εικόνα 112

➤ Τέλος εάν ανοίξετε την βάση με το όνομα ATZENTA της Microsoft Access Θα δείτε ότι όλοι οι πίνακες με τα αντίστοιχα πεδία τους και τις εγγραφές τους καθώς και τα ερωτήματα έχουν εξαχθεί.



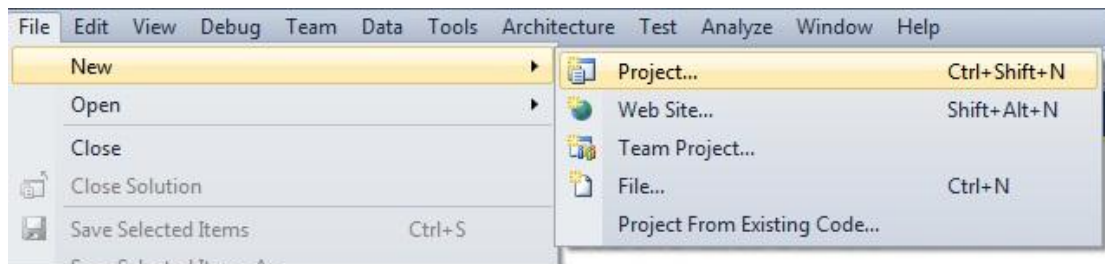
Εικόνα 113



## Άσκηση 2 - Δημιουργία σύνδεσης σε SQL Server

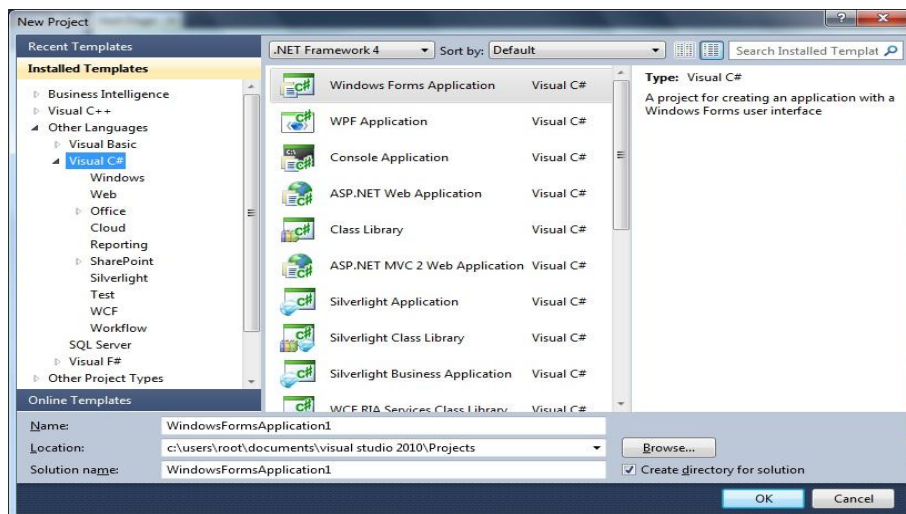
Σκοπός της παρούσας εργαστηριακής άσκησης είναι η δημιουργία μιας σύνδεσης σε MSSQL Server Βάση Δεδομένων, η δημιουργία και εκτέλεση ερωτήματος, καθώς και η εμφάνιση των αποτελεσμάτων του.

- Ανοίξτε το Microsoft Visual Studio και δημιουργήστε ένα νέο Project (Solution). Για να το κάνετε αυτό πάτε **File -> New -> Project**.



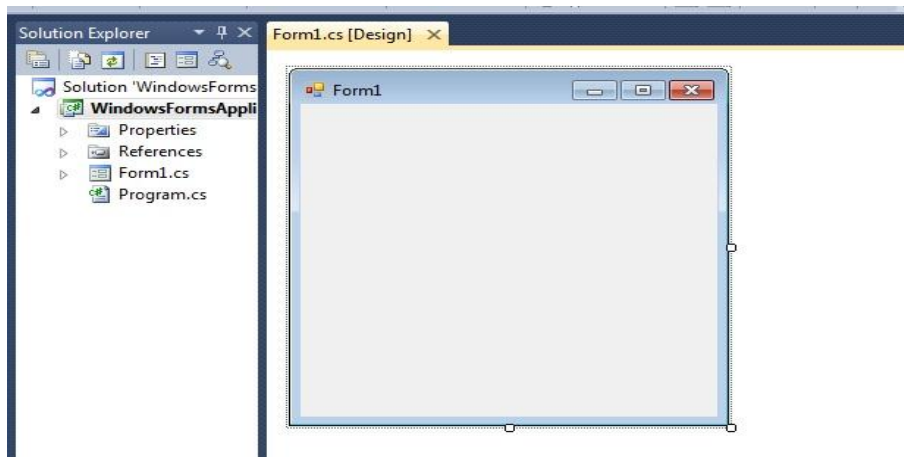
Εικόνα 114

- Η Γλώσσα Προγραμματισμού με την οποία θα ασχοληθείτε είναι η **C#**. Έτσι επιλέγετε **Other Languages -> Visual C# -> Windows Forms Application** και **OK**.



Εικόνα 115


- Η πρώτη φόρμα έχει δημιουργηθεί Εικόνα 35.

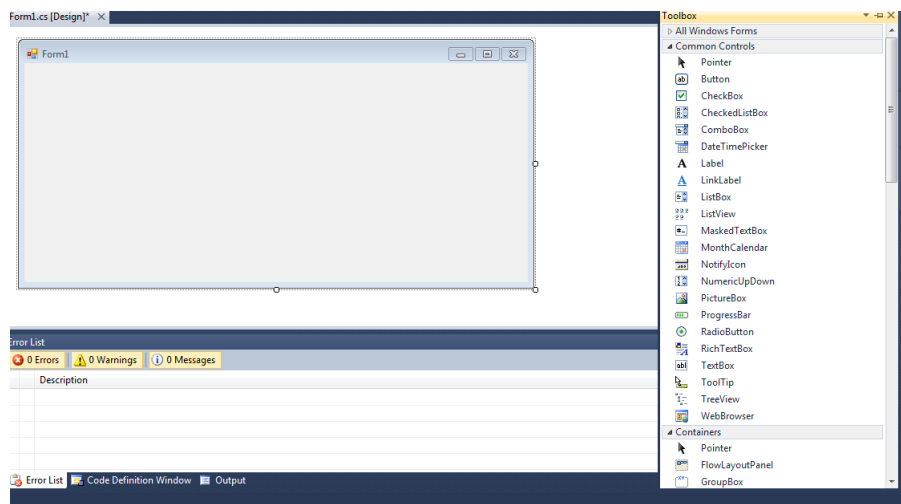


Εικόνα 116

Σε αυτή την φόρμα έχετε την δυνατότητα να προσθέσετε Components τα οποία με την σωστή ρύθμιση κάνουν επιτελούν μία λειτουργία. Στην τρέχουσα εργαστηριακή άσκηση τα Components που θα χρησιμοποιήσετε είναι τα εξής:

- τρία BindingNavigator
- τρία DataGridView

Τα Components αυτά θα τα βρείτε στο Toolbox αριστερά του παραθύρου. Σε περίπτωση που δεν φαίνεται εμφανίστε το με  Toolbox Ctrl+Alt+X.



Εικόνα 117

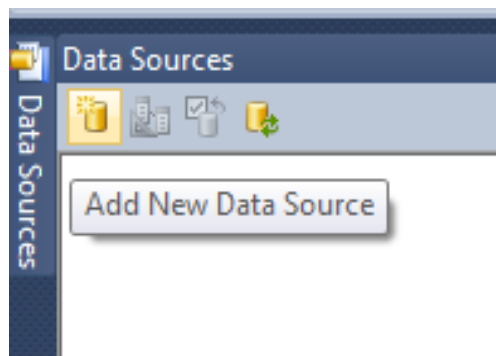
- Ανοίξτε το Toolbox και με Drag and Drop συρτέ στη φόρμα τα Components που και τοποθετήστε τα όπως παρακάτω.



Εικόνα 118

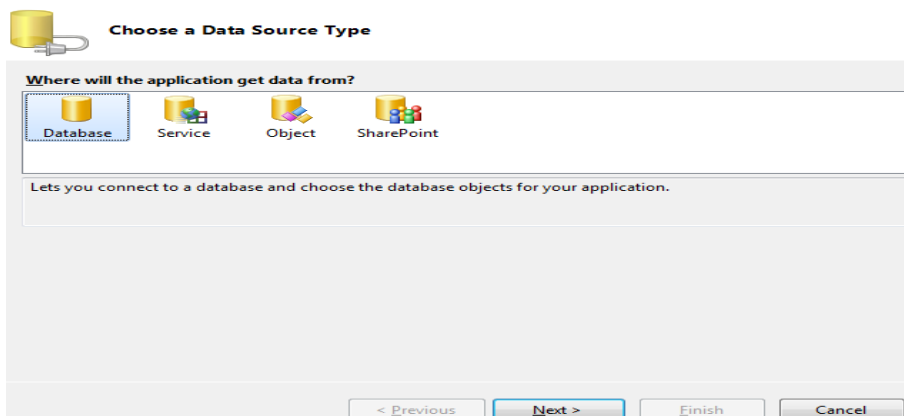
Αφού έχετε δώσει μορφή στο πρόγραμμα τώρα θα συνδέσετε τη Βάση με αυτό. Ακολουθήστε τα εξής Βήματα :

- Πηγαίνετε δεξιά στο **Data Source** και πατήστε **Add New Data Source**.



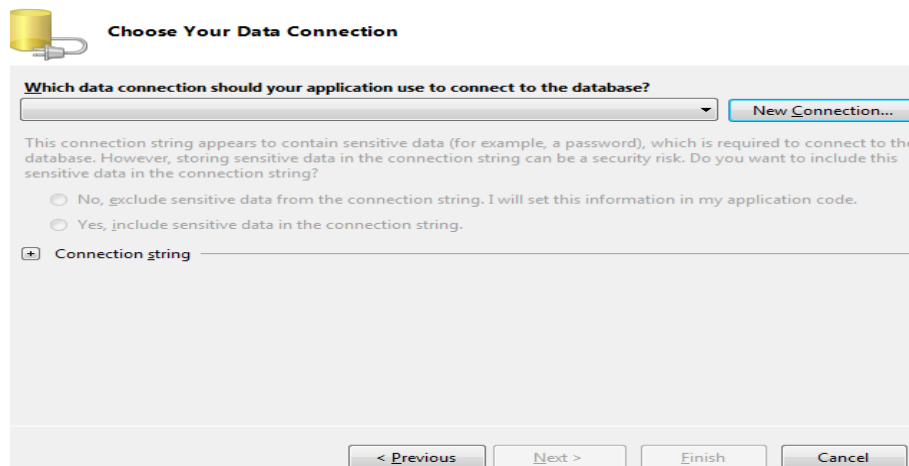
Εικόνα 119

- Στη συνέχεια ανοίγει το παράθυρο της Εικόνας και επιλέγετε **Database** → **Next** → **Dataset** → **Next**.



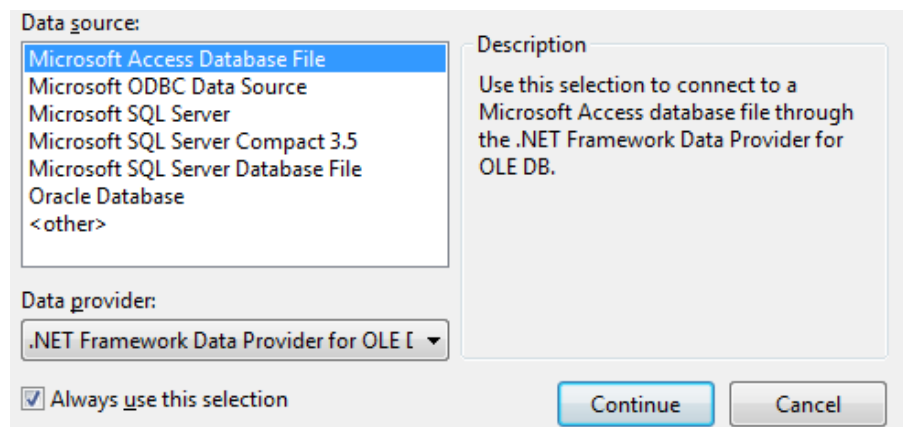
Εικόνα 120

- Έπειτα **New Connection**.



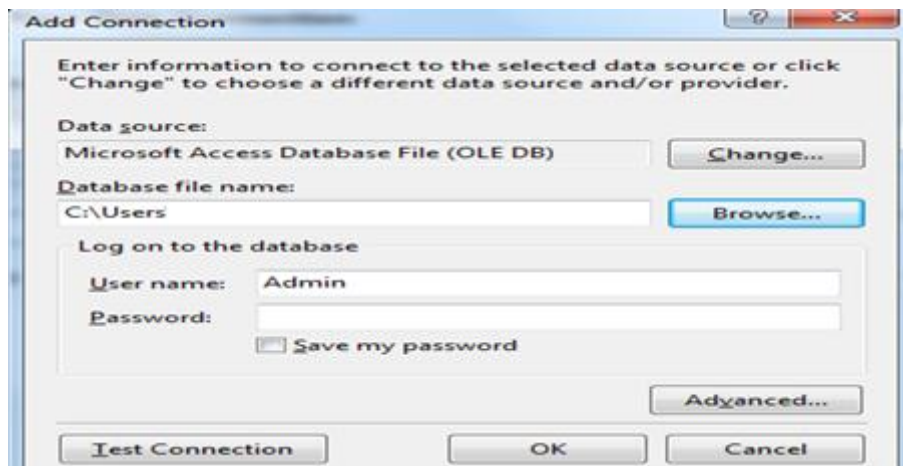
Εικόνα 121

- Επιλέγετε την πηγή από όπου θα πάρετε την βάση. Στην τρέχουσα άσκηση θα τραβήξετε την βάση από την **Microsoft Access Database File**.



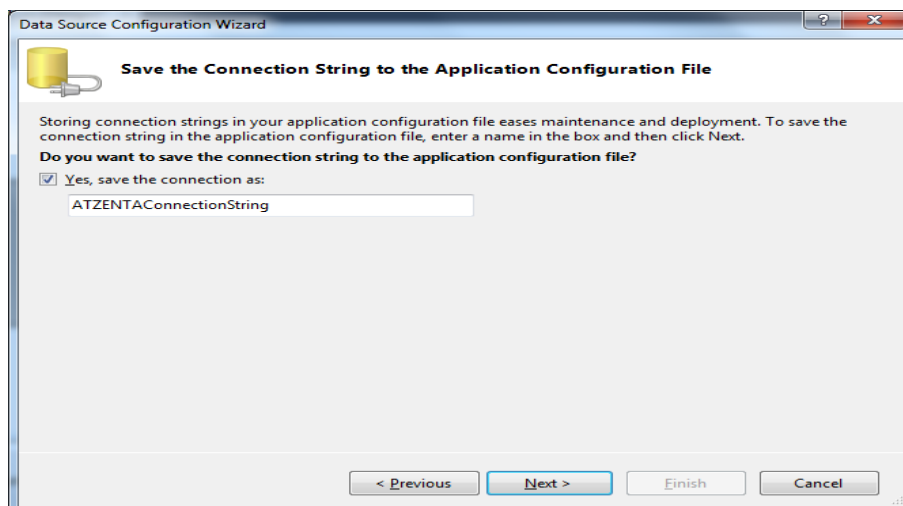
Εικόνα 122

- Επιλέξτε την βάση που θέλετε να συνδέσετε.



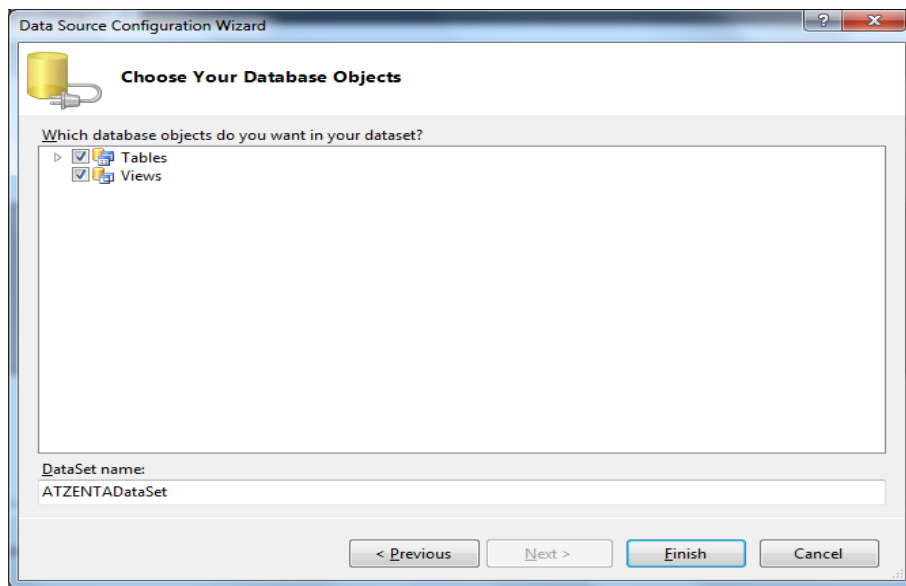
Εικόνα 123

- Next.

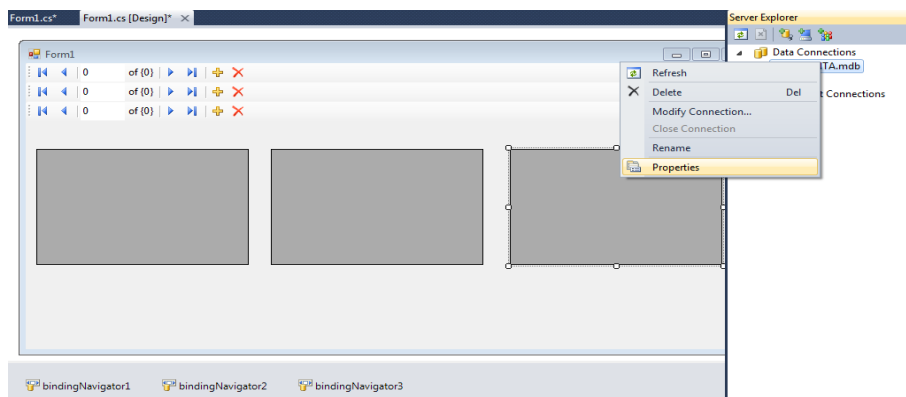


- Επιλέγετε ποια αντικείμενα της βάσης θέλετε να συνδέσετε με το Visual Studio και **Finish**.

- Στην τρέχουσα άσκηση επιλέξτε και τον πίνακα και τις όψεις.

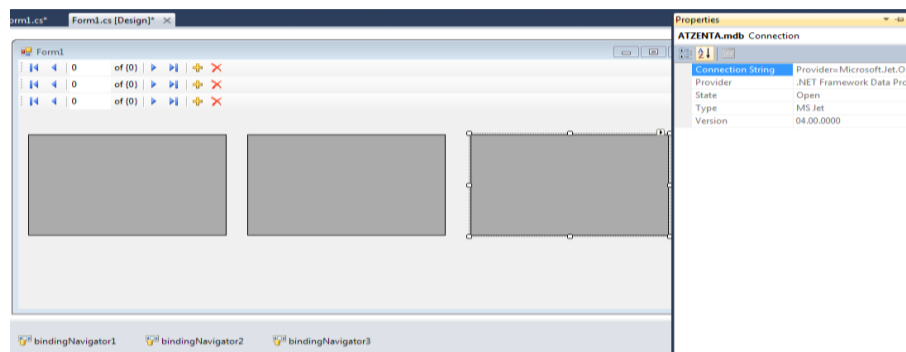


- Στη συνέχεια πηγαίνετε στον Server Explorer **δεξί κλικ στη βάση** → **Properties**



Εικόνα 124

- Από το **Connection String** παίρνετε το **Source** το οποίο θα το επικολλήσετε στον κώδικα στο **new SqlConnection**.



- Τέλος με διπλό κλικ στην φόρμα μπαίνετε στο περιβάλλον του κώδικα και γράφετε τα εξής :

```
//Εισάγουμε την κατάλληλη βιβλιοθήκη για SQL
using System.Data.SqlClient;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        /*Δημιουργούμε μεταβλητές
        * Μια τυπου SqlConnection
        * Τρεις τυπου SqlDataAdapter
        * Τρεις τυπου DataSet
        * Τρεις τυπου Binding Source
        */
        SqlConnection connection;
        SqlDataAdapter DataAdapter1, DataAdapter2, DataAdapter3;
        DataSet DataSet1, DataSet2, DataSet3;
        BindingSource BindingSource1, BindingSource2, BindingSource3;

        public Form1()
        {
            InitializeComponent();
            /* Στη μεταβλητή Connection τοποθετούμε
            * ένα νέο αντικείμενο με την σύνδεση της βάσης
            */
            connection = new SqlConnection("Data Source=.....");
            //Ανοίγουμε μια νέα σύνδεση
            connection.Open();
        } //Τέλος Form1()
    }
}
```

Αφού ολοκληρώσετε την διαδικασία εμφανίστε στο DataGridView τα εξής ερωτήματα:

- select \* from ΦΙΛΟΙ
- select \* from ΦΙΛΟΙ where ΠΟΛΗ = 'ΣΕΡΡΕΣ'
- select ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΚΙΝΗΤΟ from ΦΙΛΟΙ where ΚΙΝΗΤΟ like '69%'

Αυτό γίνεται προγραμματιστικά με τον παρακάτω τρόπο:

```
private void Form1_Load(object sender, EventArgs e)
{
    /*Γενική Χρήση :
    * Φορτώνω δεδομένα στο 1ο DataGridView και το ελέγχω με το Navigator
    */

    //ο DataAdapter1 δέχεται το αντικείμενο με το ερώτημα SQL που θέλουμε να εκτελεστεί.
    DataAdapter1 = new SqlDataAdapter("select * from ΦΙΛΟΙ", connection);
    //Δημιουργούμε ένα νέο αντικείμενο DataSet και το βάζουμε στην μεταβλητή DataSet1
    DataSet1 = new DataSet();
    //Γεμίζουμε το DataSet1 με τα αποτελέσματα που πήραμε από τον DataAdapter1
    DataAdapter1.Fill(DataSet1);
    //Δημιουργούμε ένα νέο αντικείμενο BindingSource και το βάζουμε στην μεταβλητή BindingSource1
    BindingSource1 = new BindingSource();
    /* Ορίζουμε DataSource του BindingSource1 τον 1ο
    * πίνακα (και μοναδικό) που περιέχει το DataSet1
    */
    BindingSource1.DataSource = DataSet1.Tables[0];
    /* Ορίζουμε DataSource του bindingNavigator1 το BindingSource1
    * έτσι ώστε να περιηγούμαστε στο συγκεκριμένο πίνακα
    */
    bindingNavigator1.BindingSource = BindingSource1;
    /* Ορίζουμε DataSource του dataGridView1 το BindingSource1
    * έτσι ώστε να εμφανίζονται τα περιεχόμενα του BindingSource1 στο dataGridView1
    */
    dataGridView1.DataSource = BindingSource1;

    /*Γενική Χρήση :
    * Φορτώνω δεδομένα στο 2ο DataGridView και το ελέγχω με το Navigator
    */

    DataAdapter2 = new SqlDataAdapter("select * from ΦΙΛΟΙ where ΠΟΛΗ='ΣΕΡΡΕΣ'", connection);
    DataSet2 = new DataSet();
    DataAdapter2.Fill(DataSet2);
    BindingSource2 = new BindingSource();
    BindingSource2.DataSource = DataSet2.Tables[0].DefaultView;
    bindingNavigator2.BindingSource = BindingSource2;
    dataGridView2.DataSource = BindingSource2;

    /*Γενική Χρήση :
    * Φορτώνω δεδομένα στο 3ο DataGridView και το ελέγχω με το Navigator
    */
    DataAdapter3 = new SqlDataAdapter("select ΕΠΙΘΕΤΟ, ΟΝΟΜΑ,ΤΗΛΕΦΩΝΟ_ΚΙΝΗΤΟ from ΦΙΛΟΙ|where ΚΙΝΗΤΟ like '69%'", connection);
    DataSet3 = new DataSet();
    DataAdapter3.Fill(DataSet3);
    BindingSource3 = new BindingSource();
    BindingSource3.DataSource = DataSet3.Tables[0].DefaultView;
    bindingNavigator3.BindingSource = BindingSource3;
    dataGridView3.DataSource = BindingSource3;

}

private void bindingSource4_CurrentChanged(object sender, EventArgs e)
{
}
}
```



## Επεξήγηση Κώδικα:

- Δήλωση Μεταβλητών και βιβλιοθηκών

```
//Εισάγουμε την κατάλληλη βιβλιοθήκη για SQL
using System.Data.SqlClient;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        /*Δημιουργούμε μεταβλητές
        * Μια τυπου SqlConnection
        * Τρεις τυπου SqlDataAdapter
        * Τρεις τυπου DataSet
        * Τρεις τυπου Binding Source
        */
        SqlConnection connection;
        SqlDataAdapter DataAdapter1, DataAdapter2, DataAdapter3;
        DataSet DataSet1, DataSet2, DataSet3;
        BindingSource BindingSource1, BindingSource2, BindingSource3;
    }
}
```

- Δημιουργία αντικειμένου και άνοιγμα σύνδεσης στη Βάση

```
public Form1()
{
    InitializeComponent();
    /* Στη μεταβλητή Connection τοποθετούμε
    * ένα νέο αντικείμενο με την σύνδεση της βάσης
    */
    connection = new SqlConnection("Data Source=.....");
    //Ανοίγουμε μια νέα σύνδεση
    connection.Open();
} //Τέλος Form1()
```

- Αποτύπωση των αποτελεσμάτων των ερωτημάτων SQL στο πρόγραμμα.

```
private void Form1_Load(object sender, EventArgs e)
{
    /*Γενική Χρήση :
    * Φορτώνω δεδομένα στο 1ο DataGridView και το ελέγχω με το Navigator
    */

    //ο SqlDataAdapter1 δέχεται το αντικείμενο με το ερώτημα SQL που θέλουμε να εκτελεστεί.
    SqlDataAdapter1 = new SqlDataAdapter("select * from ΦΙΑΟΙ", connection);
    //Δημιουργούμε ένα νέο αντικείμενο DataSet και το βάζουμε στην μεταβλητή DataSet1
    DataSet1 = new DataSet();
    //Γεμίζουμε το DataSet1 με τα αποτελέσματα που πήραμε από τον SqlDataAdapter1
    SqlDataAdapter1.Fill(DataSet1);
    //Δημιουργούμε ένα νέο αντικείμενο BindingSource και το βάζουμε στην μεταβλητή BindingSource1
    BindingSource1 = new BindingSource();
    /* Ορίζουμε DataSource του BindingSource1 τον 1ο
    * πίνακα (και μοναδικό) που περιέχει το DataSet1
    */
    BindingSource1.DataSource = DataSet1.Tables[0];
    /* Ορίζουμε BindingSource του bindingNavigator1 το BindingSource1
    * έτσι ώστε να περιηγηθούμε στο συγκεκριμένο πίνακα
    */
    bindingNavigator1.BindingSource = BindingSource1;
    /* Ορίζουμε DataSource του dataGridView1 το BindingSource1
    * έτσι ώστε να εμφανίζονται τα περιεχόμενα του BindingSource1 στο dataGridView1
    */
    dataGridView1.DataSource = BindingSource1;

    /*Γενική Χρήση :
    * Φορτώνω δεδομένα στο 2ο DataGridView και το ελέγχω με το Navigator
    */
}
```

```

DataAdapter2 = new SqlDataAdapter( select * from ΨΙΛΟΙ where ΠΥΛΗ= ΣΕΡΡΕΣ , connection);
DataSet2 = new DataSet();
DataAdapter2.Fill(DataSet2);
BindingSource2 = new BindingSource();
BindingSource2.DataSource = DataSet2.Tables[0].DefaultView;
bindingNavigator2.BindingSource = BindingSource2;
dataGridView2.DataSource = BindingSource2;

/*Γενική Χρήση :
 * Φορτώνω δεδομένα στο 3ο DataGridView και το ελέγχω με το Navigator
 */
DataAdapter3 = new SqlDataAdapter("select ΕΠΙΘΕΤΟ, ΟΝΟΜΑ,ΘΛΑΕΦΩΝΟ_KINHTO from ΦΙΛΟΙ| where KINHTO like '69%", connection);
DataSet3 = new DataSet();
DataAdapter3.Fill(DataSet3);
BindingSource3 = new BindingSource();
BindingSource3.DataSource = DataSet3.Tables[0].DefaultView;
bindingNavigator3.BindingSource = BindingSource3;
dataGridView3.DataSource = BindingSource3;

}

private void bindingSource4_CurrentChanged(object sender, EventArgs e)
{
}

}
}

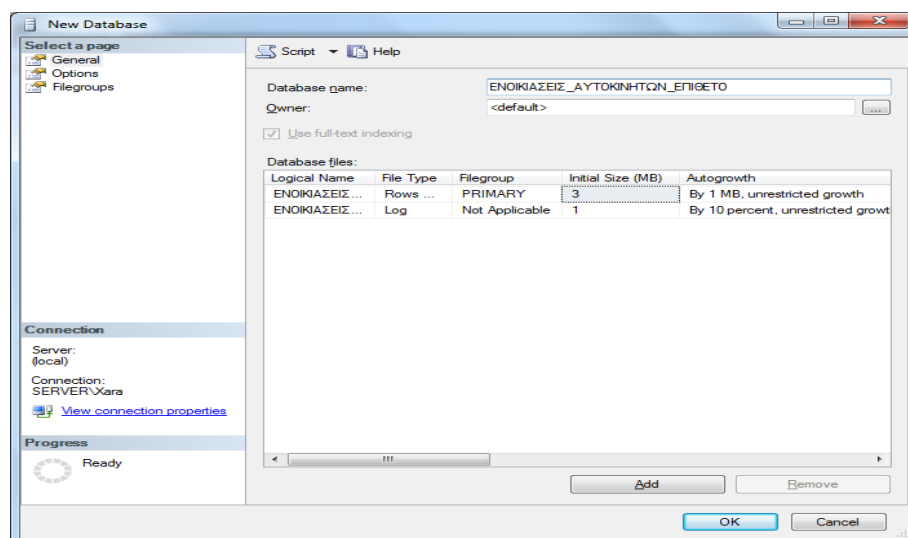
```

### Άσκηση 3 – Ενοικίαση Αυτοκινήτων

Σκοπός της εργαστηριακής άσκησης αυτής είναι η δημιουργία μιας Βάσης Δεδομένων σε SQL Server για μια επιχείρηση ενοικίασης αυτοκινήτων, ώστε να δημιουργηθούν εφαρμογές σε

C# (C Sharp) που να μπορούν να διαχειριστούν αυτήν τη βάση σε επόμενες εργαστηριακές ασκήσεις. Η βάση, οι πίνακες και οι όψεις θα δημιουργηθούν ακολουθώντας τα βήματα της 1ης Εργαστηριακής Άσκησης. Η παρούσα βάση θα περιέχει τα σταθερά στοιχεία των πελατών και των αυτοκινήτων, καθώς και πληροφορίες για την κάθε ενοικίαση.

Αρχικά δημιουργήστε την βάση ΕΝΟΙΚΙΑΣΗ\_ΑΥΤΟΚΙΝΗΤΩΝ\_ΕΠΙΘΕΤΟ όπου στο επίθετο κάθε σπουδαστής θα βάλει το δικό του.



Εικόνα 125

### Δημιουργία πινάκων

Σε αυτήν την άσκηση θα χρειαστούνε τρεις πίνακες ο πίνακας **ΠΕΛΑΤΕΣ**, ο **ΕΝΟΙΚΙΑΣΗ** και ο πίνακας **ΑΥΤΟΚΙΝΗΤΑ**.

Ο πίνακας **ΠΕΛΑΤΕΣ** θα περιέχει τα εξής πεδία : ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ (INT), ΟΝΟΜΑ(VARCHAR(20)), ΕΠΙΘΕΤΟ(VARCHAR(20)), ΟΝΟΜΑ\_ΠΑΤΡΟΣ(VARCHAR(20)), ΗΜΕΡΟΜΗΝΙΑ\_ΓΕΝΝΗΣΗΣ(VARCHAR()), ΟΔΟΣ(VARCHAR(30)), ΑΡΙΘΜΟΣ(VARCHAR(10)), ΤΚ(VARCHAR(10)), ΠΟΛΗ(VARCHAR(20)), ΑΦΜ(VARCHAR(20)), ΔΟΥ(VARCHAR(20)), ΤΗΛ\_ΚΙΝΗΤΟ(VARCHAR(15)), ΤΗΛ\_ΣΤΘΕΡΟ(VARCHAR(15)), ΤΗΛ\_ΕΡΓΑΣΙΑΣ(VARCHAR(15)), ΣΧΟΛΙΑ(VARCHAR(50)).

Column Name	Data Type	Allow Nulls
ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ	int	<input type="checkbox"/>
ΕΠΙΘΕΤΟ	varchar(20)	<input checked="" type="checkbox"/>
ΟΝΟΜΑ	varchar(20)	<input checked="" type="checkbox"/>
ΟΝΟΜΑ_ΠΑΤΡΟΣ	varchar(20)	<input checked="" type="checkbox"/>
ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗ...	date	<input checked="" type="checkbox"/>
ΟΔΟΣ	varchar(30)	<input checked="" type="checkbox"/>
ΑΡΙΘΜΟΣ	varchar(10)	<input checked="" type="checkbox"/>
ΤΚ	varchar(10)	<input checked="" type="checkbox"/>
ΠΟΛΗ	varchar(20)	<input checked="" type="checkbox"/>
ΑΦΜ	varchar(20)	<input checked="" type="checkbox"/>
ΔΟΥ	varchar(20)	<input checked="" type="checkbox"/>
ΤΗΛ_ΚΙΝΗΤΟ	varchar(15)	<input checked="" type="checkbox"/>
ΤΗΛ_ΣΤΑΘΕΡΟ	varchar(15)	<input checked="" type="checkbox"/>
ΤΗΛ_ΕΡΓΑΣΙΑΣ	varchar(15)	<input checked="" type="checkbox"/>
ΣΧΟΛΙΑ	varchar(50)	<input checked="" type="checkbox"/>

Column Properties	
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification (Is Identity)	Yes
Identity Increment	Yes
Identity Seed	1

Εικόνα 126

**ΣΧΟΛΙΟ!** Ο ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ είναι πρωτεύον κλειδί δηλαδή πεδίο αυτόματης αρίθμησης . Για να γίνει πρωτεύον κλειδί το ορίζετε ως INT και στο Column Properties ορίστε το Identity Specification από NO σε YES.

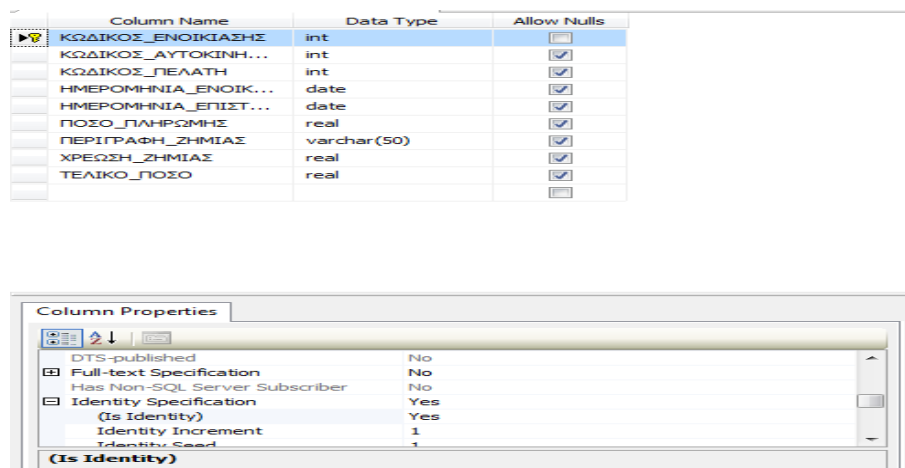
Ο πίνακας **ΑΥΤΟΚΙΝΗΤΑ** περιέχει τα εξής πεδία: ΚΩΔΙΚΟΣ ΑΥΤΟΚΙΝΗΤΟΥ(INT), ΜΑΡΚΑ(VARCHAR(30)), ΜΟΝΤΕΛΟ(VARCHAR(30)), ΑΡΙΘΜΟΣ\_ΚΥΚΛΟΦΟΡΙΑΣ(VARCHAR(30)), ΘΕΣΕΙΣ(INT) ,ΠΟΡΤΕΣ(INT), ΧΡΩΜΑ (INT), ΤΙΜΗ\_ΕΝΟΙΚΙΑΣΗΣ(REAL), ΤΥΠΟΣ\_ΚΑΥΣΙΜΟΥ(VARCHAR(30)), ΗΜΕΡΟΜΗΝΙΑ\_ΚΥΚΛΟΦΟΡΙΑΣ(DATE), ΧΙΛΙΟΜΕΤΡΑ(INT).

Column Name	Data Type	Allow Nulls
ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ	int	<input type="checkbox"/>
ΜΑΡΚΑ	varchar(30)	<input checked="" type="checkbox"/>
ΜΟΝΤΕΛΟ	varchar(30)	<input checked="" type="checkbox"/>
ΑΡΙΘΜΟΣ_ΚΥΚΛΟΦΟΡΙΑΣ	varchar(40)	<input checked="" type="checkbox"/>
ΘΕΣΕΙΣ	int	<input checked="" type="checkbox"/>
ΠΟΡΤΕΣ	int	<input checked="" type="checkbox"/>
ΧΡΩΜΑ	varchar(30)	<input checked="" type="checkbox"/>
ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ	real	<input checked="" type="checkbox"/>
ΤΥΠΟΣ_ΚΑΥΣΙΜΟΥ	varchar(30)	<input checked="" type="checkbox"/>
ΗΜΕΡΟΜΗΝΙΑ_ΚΥΚΛΟΦΟΡΙΑΣ	date	<input checked="" type="checkbox"/>
ΧΙΛΙΟΜΕΤΡΑ	int	<input checked="" type="checkbox"/>

Column Properties	
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification (Is Identity)	Yes
Identity Increment	Yes
Identity Seed	1

Εικόνα 127

Ο πίνακας **ΕΝΟΙΚΙΑΣΗ** περιέχει τα εξής πεδία: ΚΩΔΙΚΟΣ\_ΕΝΟΙΚΙΑΣΗΣ(INT), ΚΩΔΙΚΟΣ\_ΑΥΤΟΚΙΝΗΤΟΥ(INΤ), ΚΩΔΙΚΟΣ\_ΠΕΛΑΤΗ(INΤ), ΗΜΕΡΟΜΗΝΙΑ\_ΕΝΟΙΚΙΑΣΗΣ(DATE), ΗΜΕΡΟΜΗΝΙΑ\_ΕΠΙΣΤΡΟΦΗΣ(DATE), ΠΟΣΟ\_ΠΛΗΡΩΜΗΣ(REAL), ΠΕΡΙΓΡΑΦΗ\_ΖΗΜΙΑΣ(VARCHAR(50)), ΧΡΕΩΣΗ\_ΖΗΜΙΑΣ(REAL), ΤΕΛΙΚΟ\_ΠΟΣΟ(REAL).



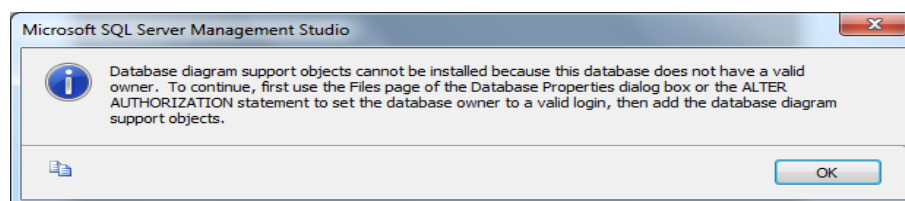
Εικόνα 128

## Δημιουργία συσχετίσεων

Για να δημιουργήσετε συσχετίσεις μεταξύ των πινάκων η διαδικασία είναι ως εξής:

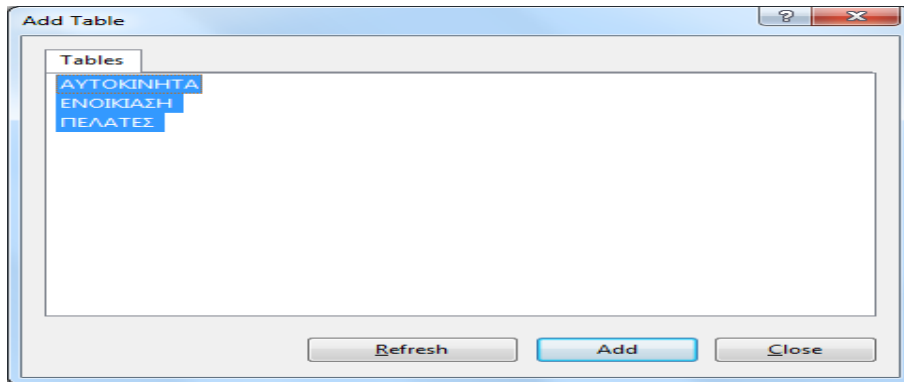
- Object Explorer→Databases→Ανοίγετε την δενδρική δομή της βάσης που επιθυμείτε (ΕΝΟΙΚΙΑΣΕΙΣ\_ΑΥΤΟΚΙΝΗΤΩΝ)→Δεξί κλικ Database Diagram→New Database Diagram και ανοίγει το παράθυρο της εικόνας με τους πίνακες που θέλετε να κάνετε τις συσχετίσεις.

**ΣΧΟΛΙΟ!** Σε περίπτωση που ακολουθήσετε την παραπάνω διαδικασία και σας βγάλει το ακόλουθο μήνυμα το διορθώνετε πηγαίνοντας στη βάση σας στο **Object Explorer**→**Δεξί κλικ**→**Properties**→Ανοίγει ένα νέο παράθυρο από το οποίο επιλέγετε **File** στο **Select Pages** και αλλάζετε τον **Owner** σε **[NT AUTHORITY\SYSTEM]**.



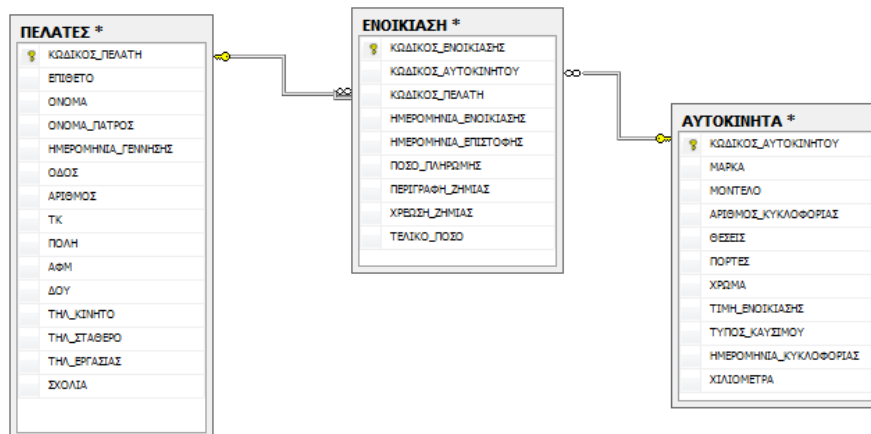
Εικόνα 129

- Έπειτα επιλέγετε τους πίνακες και Add.



Εικόνα 130

- Αν δεν φαίνονται όλοι οι πίνακες αλλάζετε το zoom σε μικρότερη τιμή. Στη συνέχεια σέρνετε το ποντίκι από το πεδίο που επιθυμείτε μέχρι το πεδίο που θα ενώσετε και εκεί το αφήνετε ώστε να γίνει η συσχέτιση και να ενωθούν τα πεδία που θέλετε να συσχετιστούν. Πατήστε OK στα επόμενα δύο παράθυρα ώστε να ολοκληρωθεί η συσχέτιση.



Εικόνα 131

**ΣΧΟΛΙΟ!** Παρόλο που θα ενώσετε σωστά τα πεδία σας τα βελάκια που κάνουν την συσχέτιση δεν τοποθετούνται σε σωστό σημείο όμως η συσχέτιση είναι επιτυχείς.

- Αποθηκεύστε το διάγραμμα.

Αφού κάνετε την συσχέτιση των πινάκων μπορείτε να εισάγετε εγγραφές στους πίνακές σας. Ο πίνακας **ΠΕΛΑΤΕΣ** θα εμπεριέχει τις εξής εγγραφές:

ΚΩΔΙΚΟΣ_ΠΕΛ...	ΕΠΙΘΕΤΟ	ΟΝΟΜΑ	ΟΝΟΜΑ_ΠΑΤΡΟΣ	ΗΜΕΡΟΜΗΝΙΑ_...	ΟΔΟΣ	ΑΡΙΘΜΟΣ
1	ΓΕΩΡΓΙΟΥ	ΑΠΟΣΤΟΛΟΣ	ΙΩΑΝΝΗΣ	1991-01-01	ΔΕΛΦΩΝ	23
2	ΠΑΠΑΔΟΠΟΥΛΟΣ	ΓΕΩΡΓΙΟΣ	ΔΗΜΗΤΡΙΟΣ	1988-03-05	ΒΕΝΙΖΕΛΟΥ	45
3	ΜΙΧΑΗΛΙΔΗΣ	ΑΝΤΩΝΙΟΣ	ΒΑΣΙΛΕΙΟΣ	1987-09-01	ΕΡΜΟΥ	12
4	ΚΑΡΑΛΗΣ	ΑΝΔΡΕΑΣ	ΙΩΑΝΝΗΣ	1966-02-08	ΤΣΙΜΙΣΚΗ	67
5	ΤΡΙΑΝΤΑΦΥΛΛΟΥ	ΑΝΤΩΝΗΣ	ΝΙΚΟΛΑΟΣ	1977-02-07	ΔΩΔΩΝΗΣ	123

Εικόνα 132

ΑΡΙΘΜΟΣ	ΤΚ	ΠΟΛΗ	ΑΦΜ	ΔΟΥ	ΤΗΛ_ΚΙΝΗΤΟ	ΤΗΛ_ΣΤΑΘΕΡΟ	ΤΗΛ_ΕΡΓΑΣΙΑΣ	ΣΧΟΛΙΑ
23	62100	ΣΕΡΡΕΣ	3251662	Α ΣΕΡΡΩΝ	6987686826	2321026828	2321072762	ΠΡΟΣΕΚΤΙΚΟΣ
45	62100	ΣΕΡΡΕΣ	2687992	Α ΣΕΡΡΩΝ	6976528528	2321064282	2321035286	ΑΣΤΑΤΟΣ
12	23456	ΑΘΗΝΑ	8765869	ΣΤ ΑΘΗΝΩΝ	6946636638	2528169170	2686282892	ΕΠΙΚΙΝΔΥΝΟΣ
67	67658	ΘΕΣΣΑΛΟΝΙΚΗ	7696936	Α ΘΕΣΣΑΛΟΝΙΚΗΣ	6934567623	2347268203	2347972387	ΠΡΟΣΕΚΤΙΚΟΣ
123	81681	ΙΩΑΝΝΙΝΑ	9761561	Γ ΙΩΑΝΝΙΝΩΝ	6992527525	2462446529	5524427275	ΑΠΑΡΑΔΕΚΤΟΣ

Εικόνα 133

Ο πίνακας **ΑΥΤΟΚΙΝΗΤΑ** περιέχει:

ΚΩΔΙΚΟΣ_ΑΥΤ...	ΜΑΡΚΑ	ΜΟΝΤΕΛΟ	ΑΡΙΘΜΟΣ_ΚΥΚ...	ΘΕΣΕΙΣ	ΠΟΡΤΕΣ	ΧΡΩΜΑ	ΤΙΜΗ_ΕΝΟΙΚΙΑ...	ΤΥΠΟΣ_ΚΑΥΣΙΜΟΥ
1	FIAT	STILO	4356	5	5	ΚΟΚΚΙΝΟ	20	ΒΕΝΖΙΝΗ
2	OPEL	CORSA	2625	5	5	ΛΕΥΚΟ	18	ΠΕΤΡΕΛΑΙΟ
3	MERCEDES	SLK	7262	2	5	ΑΣΗΜΙ	30	ΠΕΤΡΕΛΑΙΟ
4	CITROEN	SAXO	8722	4	3	ΜΑΥΡΟ	18	ΒΕΝΖΙΝΗ
5	ALFA ROMEO	GOLF	6262	4	5	ΜΑΥΡΟ	30	ΒΕΝΖΙΝΗ

Εικόνα 134

ΗΜΕΡΟΜΗΝΙΑ_ΚΥΚΛΟ...	ΧΙΛΙΟΜΕΤΡΑ
2004-01-01	56000
2000-01-01	87000
2003-03-12	90000
2000-06-05	50000
2004-02-02	60000

Εικόνα 135

Τέλος ο πίνακας **ΕΝΟΙΚΙΑΣΗ** περιέχει:

ΚΩΔΙΚΟΣ_ΕΝΟΙΚΙΑΣΗΣ	ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ	ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ	ΗΜΕΡΟΜΗΝΙΑ_ΕΝΟΙΚΙΑΣΗΣ	ΗΜΕΡΟΜΗΝΙΑ_ΕΠΙΣΤΟΦΗΣ
1	1	1	2007-07-07	2007-07-17
2	2	5	2007-07-08	2007-07-12
3	2	3	2007-09-09	2007-09-13
4	3	1	2007-03-10	2007-04-18
5	4	1	2007-11-09	2007-11-18
6	5	2	2007-11-19	2007-11-25

Εικόνα 136

ΠΟΣΟ ΠΛΗΡΩΜΗΣ	ΠΕΡΙΓΡΑΦΗ ΖΗΜΙΑΣ	ΧΡΕΩΣΗ ΖΗΜΙΑΣ	ΤΕΛΙΚΟ ΠΟΣΟ
200	NULL	NULL	200
225	NULL	NULL	225
200	NULL	NULL	200
400	ΛΑΣΤΙΧΟ ΕΜΠΡΟΣ ΔΕΞΙΑ	150	550
160	NULL	NULL	160
300	NULL	NULL	300

Εικόνα 137

## Δημιουργία πινάκων με ερωτήματα

- Δημιουργία και εισαγωγή δεδομένων στον πίνακα **ΑΥΤΟΚΙΝΗΤΑ**.

```
CREATE TABLE [ΑΥΤΟΚΙΝΗΤΑ] (
    [ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ] [int] IDENTITY(1,1) NOT NULL,
    [ΜΑΡΚΑ] [varchar](30) NULL,
    [ΜΟΝΤΕΛΟ] [varchar](30) NULL,
    [ΑΡΙΘΜΟΣ_ΚΥΚΛΟΦΟΡΙΑΣ] [varchar](40) NULL,
    [ΘΕΣΕΙΣ] [int] NULL,
    [ΠΟΡΤΕΣ] [int] NULL,
    [ΧΡΩΜΑ] [varchar](30) NULL,
    [ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ] [real] NULL,
    [ΤΥΠΟΣ_ΚΑΥΣΙΜΟΥ] [varchar](30) NULL,
    [ΗΜΕΡΟΜΗΝΙΑ_ΚΥΚΛΟΦΟΡΙΑΣ] [date] NULL,
    [ΧΙΛΙΟΜΕΤΡΑ] [int] NULL,
    CONSTRAINT [PK_ΑΥΤΟΚΙΝΗΤΑ] PRIMARY KEY CLUSTERED;

INSERT ΑΥΤΟΚΙΝΗΤΑ
(ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ, ΜΑΡΚΑ, ΜΟΝΤΕΛΟ, ΑΡΙΘΜΟΣ_ΚΥΚΛΟΦΟΡΙΑΣ, ΘΕΣΕΙΣ,
ΠΟΡΤΕΣ, ΧΡΩΜΑ, ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ, ΤΥΠΟΣ_ΚΑΥΣΙΜΟΥ,
ΗΜΕΡΟΜΗΝΙΑ_ΚΥΚΛΟΦΟΡΙΑΣ, ΧΙΛΙΟΜΕΤΡΑ)
VALUES (1, 'FIAT', 'STILO', '4356', 5, 5, 'ΚΟΚΚΙΝΟ', 20, 'BENZINΗ',
'01/01/2014', 56000);

INSERT ΑΥΤΟΚΙΝΗΤΑ
(ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ, ΜΑΡΚΑ, ΜΟΝΤΕΛΟ, ΑΡΙΘΜΟΣ_ΚΥΚΛΟΦΟΡΙΑΣ, ΘΕΣΕΙΣ,
ΠΟΡΤΕΣ, ΧΡΩΜΑ, ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ, ΤΥΠΟΣ_ΚΑΥΣΙΜΟΥ,
ΗΜΕΡΟΜΗΝΙΑ_ΚΥΚΛΟΦΟΡΙΑΣ, ΧΙΛΙΟΜΕΤΡΑ)
VALUES (1004, 'OPEL', 'CORSA', '2625', 5, 5, 'ΛΕΥΚΟ', 18,
'ΠΕΤΡΕΛΑΙΟ', '01/01/2000', 87000);

INSERT ΑΥΤΟΚΙΝΗΤΑ
(ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ, ΜΑΡΚΑ, ΜΟΝΤΕΛΟ, ΑΡΙΘΜΟΣ_ΚΥΚΛΟΦΟΡΙΑΣ, ΘΕΣΕΙΣ,
ΠΟΡΤΕΣ, ΧΡΩΜΑ, ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ, ΤΥΠΟΣ_ΚΑΥΣΙΜΟΥ,
ΗΜΕΡΟΜΗΝΙΑ_ΚΥΚΛΟΦΟΡΙΑΣ, ΧΙΛΙΟΜΕΤΡΑ)
VALUES (1005, 'MERCEDES', 'SLK', '7262', 2, 5, 'ΑΣΗΜΙ', 30,
'ΠΕΤΡΕΛΑΙΟ', '12/3/2014', 90000);
```

Ομοίως εισάγεται και τις επόμενες δύο εγγραφές.

- Δημιουργία και εισαγωγή δεδομένων στον πίνακα **ΠΕΛΑΤΕΣ**.



```

CREATE TABLE ΠΕΛΑΤΕΣ] (
    [ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ] [int] IDENTITY(1,1) NOT NULL,
    [ΕΠΙΘΕΤΟ] [varchar] (20) NULL,
    [ΟΝΟΜΑ] [varchar] (20) NULL,
    [ΟΝΟΜΑ_ΠΑΤΡΟΣ] [varchar] (20) NULL,
    [ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ] [date] NULL,
    [ΟΔΟΣ] [varchar] (30) NULL,
    [ΑΡΙΘΜΟΣ] [varchar] (10) NULL,
    [ΤΚ] [varchar] (10) NULL,
    [ΠΟΛΗ] [varchar] (20) NULL,
    [ΑΦΜ] [varchar] (20) NULL,
    [ΔΟΥ] [varchar] (20) NULL,
    [ΤΗΛ_ΚΙΝΗΤΟ] [varchar] (15) NULL,
    [ΤΗΛ_ΣΤΑΘΕΡΟ] [varchar] (15) NULL,
    [ΤΗΛ_ΕΡΓΑΣΙΑΣ] [varchar] (15) NULL,
    [ΣΧΟΛΙΑ] [varchar] (50) NULL,
    CONSTRAINT [PK_ΠΕΛΑΤΕΣ] PRIMARY KEY CLUSTERED

INSERT ΠΕΛΑΤΕΣ
(ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΟΝΟΜΑ_ΠΑΤΡΟΣ, ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ,
ΟΔΟΣ, ΑΡΙΘΜΟΣ, ΤΚ, ΠΟΛΗ, ΑΦΜ, ΔΟΥ, ΤΗΛΕΦΩΝΟ_ΚΙΝΗΤΟ, ΤΗΛΕΦΩΝΟ_ΣΤΑΘΕΡΟ,
ΤΗΛΕΦΩΝΟ_ΕΡΓΑΣΙΑΣ, ΣΧΟΛΙΑ)
VALUES (1, 'ΓΕΩΡΓΙΟΥ', 'ΑΠΟΣΤΟΛΟΣ', 'ΙΩΑΝΝΗΣ', '01/01/1991',
'ΔΕΛΦΩΝ', '23', '62100', 'ΣΕΡΡΕΣ', '3251662', 'Α ΣΕΡΡΩΝ',
'6987686826', '2321026828', '2321072762', 'ΠΡΟΣΕΚΤΙΚΟΣ')

```

Ομοίως εισάγεται και τις επόμενες τέσσερις εγγραφές με τελευταία εντολή την παρακάτω που κλείνει τον πίνακα.

```
SET IDENTITY_INSERT ΠΕΛΑΤΕΣ OFF.....
```

➤ Δημιουργία και εισαγωγή δεδομένων στον πίνακα **ΕΝΟΙΚΙΑΣΗ**.

```

CREATE TABLE [ΕΝΟΙΚΙΑΣΗ] (
    [ΚΩΔΙΚΟΣ_ΕΝΟΙΚΙΑΣΗΣ] [int] IDENTITY(1,1) NOT NULL,
    [ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ] [int] NULL,
    [ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ] [int] NULL,
    [ΗΜΕΡΟΜΗΝΙΑ_ΕΝΟΙΚΙΑΣΗΣ] [date] NULL,
    [ΗΜΕΡΟΜΗΝΙΑ_ΕΠΙΣΤΡΟΦΗΣ] [date] NULL,
    [ΠΟΣΟ_ΠΛΗΡΩΜΗΣ] [real] NULL,
    [ΠΕΡΙΓΡΑΦΗ_ΖΗΜΙΑΣ] [varchar] (50) NULL,
    [ΧΡΕΩΣΗ_ΖΗΜΙΑΣ] [real] NULL,
    [ΤΕΛΙΚΟ_ΠΟΣΟ] [real] NULL,
    CONSTRAINT [PK_ΕΝΟΙΚΙΑΣΗ] PRIMARY KEY CLUSTERED

INSERT ΕΝΟΙΚΙΑΣΗ
(ΚΩΔΙΚΟΣ_ΕΝΟΙΚΙΑΣΗΣ, ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ, ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ,
ΗΜΕΡΟΜΗΝΙΑ_ΕΝΟΙΚΙΑΣΗΣ, ΗΜΕΡΟΜΗΝΙΑ_ΕΠΙΣΤΡΟΦΗΣ, ΠΟΣΟ_ΠΛΗΡΩΜΗΣ,
ΠΕΡΙΓΡΑΦΗ_ΖΗΜΙΑΣ, ΧΡΕΩΣΗ_ΖΗΜΙΑΣ, ΤΕΛΙΚΟ_ΠΟΣΟ)
VALUES (1, 1, 1, '7/7/2007', '17/7/2007', 200, NULL, 0, 200)

```

Ομοίως και για τις επόμενες τέσσερις εγγραφές και στη συνέχεια δημιουργήστε τις συσχετίσεις.

```
SET IDENTITY_INSERT ΕΝΟΙΚΙΑΣΗ OFF
```

```

ALTER TABLE ΕΝΟΙΚΙΑΣΗ WITH CHECK ADD CONSTRAINT
FK_ΕΝΟΙΚΙΑΣΗ_ΑΥΤΟΚΙΝΗΤΑ FOREIGN KEY (ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ)
REFERENCES ΑΥΤΟΚΙΝΗΤΑ (ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ)
GO
ALTER TABLE ΕΝΟΙΚΙΑΣΗ CHECK CONSTRAINT FK_ΕΝΟΙΚΙΑΣΗ_ΑΥΤΟΚΙΝΗΤΑ
GO
ALTER TABLE ΕΝΟΙΚΙΑΣΗ WITH CHECK ADD CONSTRAINT FK_ΕΝΟΙΚΙΑΣΗ_ΠΕΛΑΤΕΣ
FOREIGN KEY (ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ)
REFERENCES ΠΕΛΑΤΕΣ (ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ)
GO
ALTER TABLE ΕΝΟΙΚΙΑΣΗ CHECK CONSTRAINT FK_ΕΝΟΙΚΙΑΣΗ_ΠΕΛΑΤΕΣ
GO

```

## Δημιουργία Όψεων

### Ερώτηση 1

Να εμφανιστεί αναλυτικά το ιστορικό των ενοικιάσεων των πελατών.

### Απάντηση 1

```

CREATE VIEW ΙΣΤΟΡΙΚΟ_ΕΝΟΙΚΙΑΣΕΩΝ AS (SELECT
ΑΥΤΟΚΙΝΗΤΑ.ΜΑΡΚΑ,
ΑΥΤΟΚΙΝΗΤΑ.ΜΟΝΤΕΛΟ,
ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ,ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ,ΕΝ
ΟΙΚΙΑΣΗ.ΗΜΕΡΟΜΗΝΙΑ_ΓΕΝΝΗΣΗΣ,
ΕΝΟΙΚΙΑΣΗ.ΗΜΕΡΟΜΗΝΙΑ_ΕΠΙΣΤΡΟΦΗΣ, ΕΝΟΙΚΙΑΣΗ.ΠΟΣΟ_ΠΛΗΡΩΜΗΣ,
ΕΝΟΙΚΙΑΣΗ.ΠΕΡΙΓΡΑΦΗ_ΖΗΜΙΑΣ,ΕΝΟΙΚΙΑΣΗ.ΧΡΕΩΣΗ_ΖΗΜΙΑΣ,
ΕΝΟΙΚΙΑΣΗ.ΤΕΛΙΚΟ_ΠΟΣΟ
FROM ΑΥΤΟΚΙΝΗΤΑ INNER JOIN ΕΝΟΙΚΙΑΣΗ ON
ΑΥΤΟΚΙΝΗΤΑ.ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ =
ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ INNER JOIN ΠΕΛΑΤΕΣ ON
ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ = ΠΕΛΑΤΕΣ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ)

```

### Ερώτηση 2

Να εμφανιστούν τα αυτοκίνητα που δεν έχουν ενοικιαστεί ποτέ.

### Απάντηση 2

```

CREATE VIEW ΞΕΝΟΙΚΙΑΣΤΑ AS (SELECT *
FROM ΑΥΤΟΚΙΝΗΤΑ
WHERE (ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ NOT IN
(SELECT ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ
FROM ΕΝΟΙΚΙΑΣΗ))

```

### Ερώτηση 3

Να προβληθούν τα στοιχεία του κάθε πελάτη με το συνολικό ποσό που πλήρωσε.

### Απάντηση 3

```

CREATE VIEW ΣΥΝΟΛΙΚΟ_ΠΟΣΟ AS (SELECT
ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ,
SUM (dbo.ΕΝΟΙΚΙΑΣΗ.ΤΕΛΙΚΟ_ΠΟΣΟ) AS ΤΖΙΡΟΣ
FROM ΕΝΟΙΚΙΑΣΗ INNER JOIN ΠΕΛΑΤΕΣ ON
ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ = ΠΕΛΑΤΕΣ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ
GROUP BY ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ)

```

#### Ερώτηση 4

Να προβληθούν τα στοιχεία πελάτη με το μέγιστο συνολικό ποσό.

#### Απάντηση 4

```
CREATE VIEW ΜΕΓΙΣΤΟ_ΠΟΣΟ_ΠΕΛΑΤΗ AS (SELECT
ΠΕΛΑΤΕΣ.ΕΠΙΘΕΤΟ,
ΠΕΛΑΤΕΣ.ΟΝΟΜΑ, ΠΕΛΑΤΕΣ.ΚΙΝΗΤΟ, ΠΕΛΑΤΕΣ.ΠΟΛΗ,
ΣΥΝΟΛΙΚΟ_ΠΟΣΟ.ΤΖΙΡΟΣ
FROM ΣΥΝΟΛΙΚΟ_ΠΟΣΟ INNER JOIN
ΠΕΛΑΤΕΣ ON ΣΥΝΟΛΙΚΟ_ΠΟΣΟ.ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ =
ΠΕΛΑΤΕΣ_ΚΩΔΙΚΟΣ.ΠΕΛΑΤΗ
WHERE (ΣΥΝΟΛΙΚΟ_ΠΟΣΟ.ΤΖΙΡΟΣ =
(SELECT MAX (ΤΖΙΡΟΣ)
FROM ΣΥΝΟΛΙΚΟ_ΠΟΣΟ)))
```

#### Ερώτηση 5

Να εμφανιστεί το ποσοστό των ενοικιάσεων κατά το οποίο συνέβη ζημιά σε ενοικιαζόμενο όχημα.

#### Απάντηση 5

```
CREATE VIEW ΠΟΣΟΣΤΟ AS (
SELECT CAST (COUNT (ΠΕΡΙΓΡΑΦΗ_ΖΗΜΙΑΣ) * 100 / COUNT
(ΚΩΔΙΚΟΣ_ΕΝΟΙΚΙΑΣΗΣ) AS
varchar(5)) + ' %' AS ΠΟΣΟΣΤΟ
FROM ΕΝΟΙΚΙΑΣΗ)
```

## Ερωτήματα Εμπέδωσης

### Ερώτηση 1

Δημιουργήστε μια όψη με τίτλο <<V6>> όπου θα εμφανίζονται η ΜΑΡΚΑ, το ΜΟΝΤΕΛΟ, το ΕΠΩΝΥΜΟ και το ΟΝΟΜΑ και την ΗΜΕΡΟΜΗΝΙΑ ΓΕΝΝΗΣΗΣ του πελάτη ταξινομημένα αλφαβητικά κατά επώνυμο.

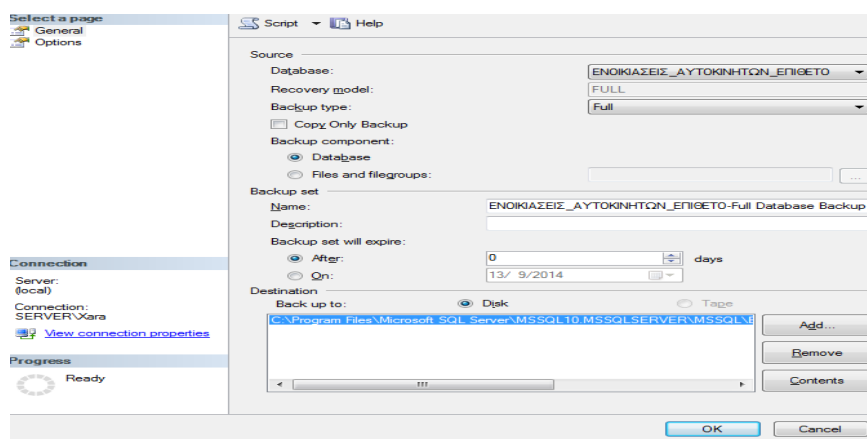
### Ερώτηση 2

Δημιουργήστε μια όψη με τίτλο <<V7>> όπου θα εμφανίζονται η μάρκα του αυτοκινήτου και το σύνολο των αυτοκινήτων που υπάρχουν στην βάση για την κάθε μάρκα. Χρησιμοποιήστε το γραφικό περιβάλλον δημιουργίας ερωτημάτων που παρέχει ο SQL Server.

## Δημιουργία Back up

Δημιουργήστε αντίγραφο ασφαλείας με όνομα ΕΝΟΙΚΙΑΣΕΙΣ\_ΑΥΤΟΚΙΝΗΤΩΝ\_BACK\_UP:

- Πηγαίνετε στην βάση **ΕΝΟΙΚΙΑΣΕΙΣ\_ΑΥΤΟΚΙΝΗΤΩΝ\_ΕΠΙΘΕΤΟ** → **Tasks** → **Back up** ανοίγει η παρακάτω εικόνα επιλέγετε την τοποθεσία αποθήκευσης του Back up και **OK**.



Εικόνα 138

## Δημιουργία Back up με χρονοπρογραμματισμό

Έστω ότι θέλετε να προγραμματίσετε αυτόματα τη δημιουργία αντιγράφου ασφαλείας κάθε Τετάρτη στις 22:00. Αρκεί να ακολουθήσετε την εξής διαδικασία:

- Εκτελέστε τα τρία Queries:

### Query 1

```
EXEC SP_CONFIGURE 'Agent XPs'
```

### Query 2

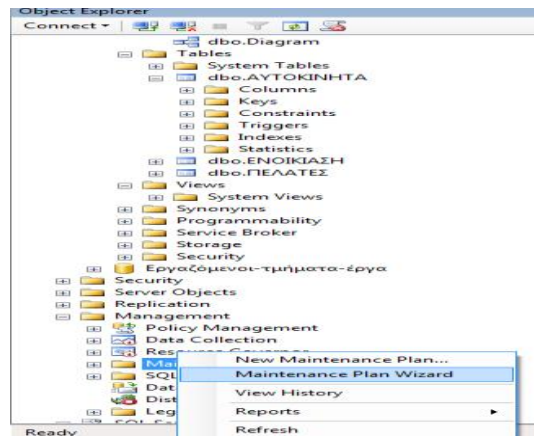
```
EXEC SP_CONFIGURE 'show advanced options', 1
```

```
GO
RECONFIGURE
GO
EXEC SP_CONFIGURE 'show advanced options'
```

### Query 3

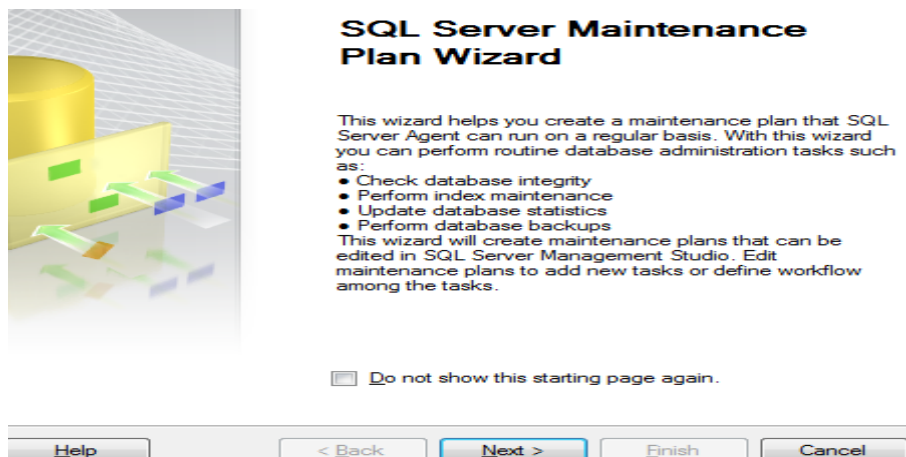
```
EXEC SP_CONFIGURE 'Agent XPs', 1
GO
RECONFIGURE
```

- Στη συνέχεια Object Explorer → Management → Δεξί Κλικ → Maintenance Plans → Maintenance Plan Wizard.



Εικόνα 139

- Next.



Εικόνα 140

- **Next.** Έχει έχει το Default όνομα εάν θέλετε να βάλετε κάποιο άλλο αλλάξτε το από το Name και εφόσον θέλετε κάποιο σχέδιο (όπως στο παράδειγμα μας το Back up να εκτελείτε κάθε Τετάρτη στις 10) η ρύθμιση του γίνεται στο Change.

**Select Plan Properties**  
How do you want to schedule your maintenance tasks?

Name: MaintenancePlan

Description:

Separate schedules for each task  
 Single schedule for the entire plan or no schedule

Schedule:  
Not scheduled (On Demand) Change...

Help < Back Next > Finish Cancel

Εικόνα 141

- Για να εκτελείτε κάθε Τετάρτη βράδυ το Back up πρέπει να κάνετε τις παρακάτω ρυθμίσεις. Στο Frequency επιλέγετε την ημέρα και στο Daily Frequency την ώρα και OK.

Name: MaintenancePlan Jobs in Schedule

Schedule type: Recuring  Enabled

One-time occurrence  
Date: 13/ 9/2014 Time: 2:41:03 πμ

Frequency  
Occurs: Weekly  
Recurs every: 1 week(s) on  
 Monday  Wednesday  Friday  Saturday  
 Tuesday  Thursday  Sunday

Daily frequency  
 Occurs once at: 10:00:00 πμ  
 Occurs every: 1 hour(s) Starting at: 12:00:00 πμ Ending at: 11:59:59 πμ

Duration  
Start date: 13/ 9/2014  End date: 13/ 9/2014  No end date

Summary  
Description: Occurs every week on Wednesday, Sunday at 10:00:00 πμ. Schedule will be used starting on

OK Cancel Help

Εικόνα 142

- Next.

**Select Plan Properties**  
How do you want to schedule your maintenance tasks?

Name: MaintenancePlan

Description:

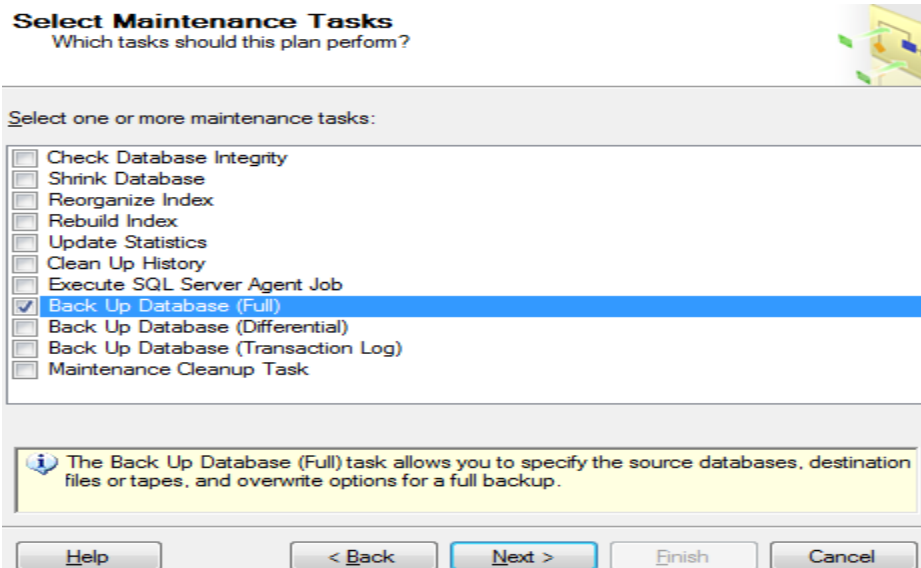
Separate schedules for each task  
 Single schedule for the entire plan or no schedule

Schedule:  
Occurs every week on Wednesday, Sunday at 10:00:00 πμ. Sche Change...

Help < Back Next > Finish Cancel

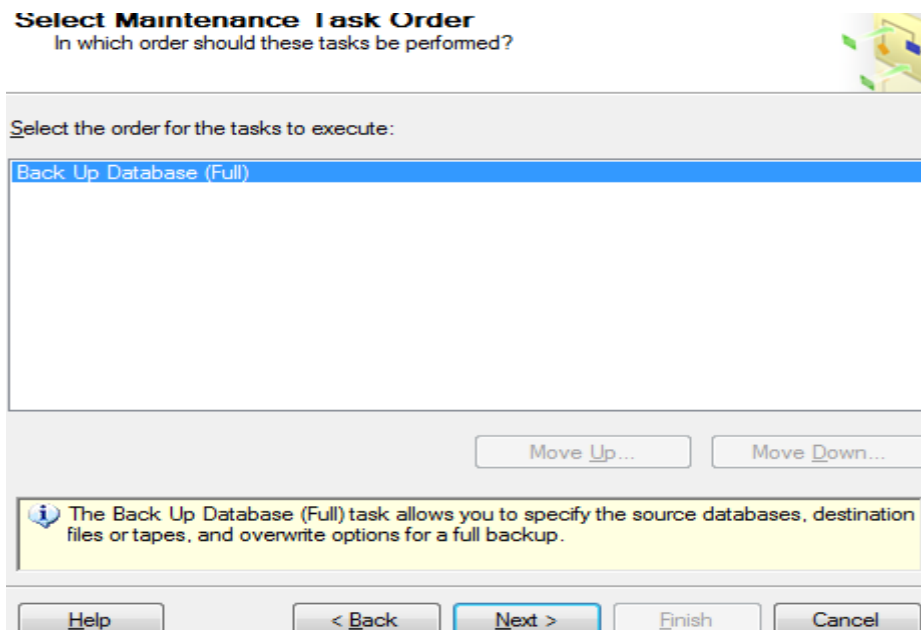
Εικόνα 143

- Επιλέγετε τον τύπο του Back up που επιθυμείτε και **Next**.



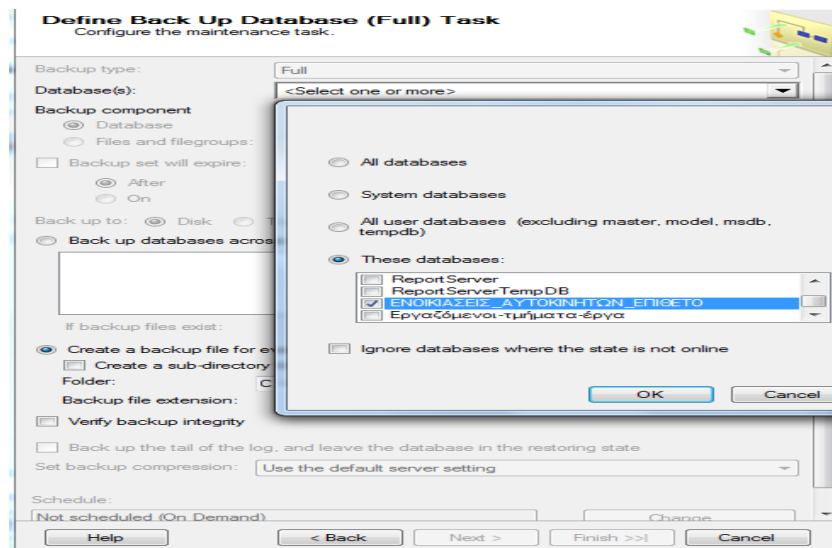
Εικόνα 144

- **Next.**



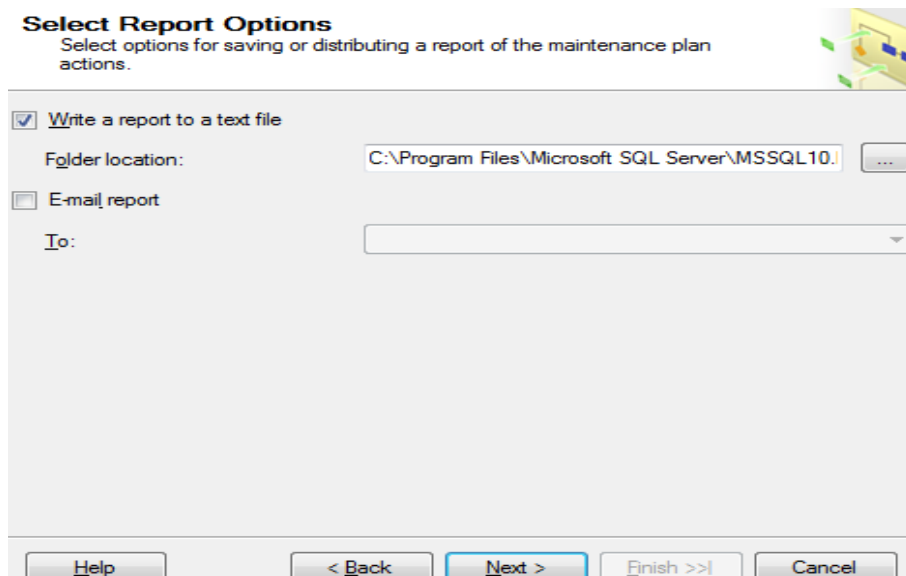
Εικόνα 145

- Επιλέγετε την βάση για την οποία θα κάνει το Back up.



Εικόνα 146

- Δώστε την ακριβή τοποθεσία όπου θα αποθηκεύεται το Back up και **Next**.



Εικόνα 147



➤ **Finish.**

**Complete the Wizard**

Verify the choices made in the wizard, and then click Finish.



Click Finish to perform the following actions:

- [-] Maintenance Plan Wizard
  - [+] Create Maintenance Plan 'MaintenancePlan'
  - [+] Define Back Up Database (Full) Task
  - [+] Selected reporting options




Εικόνα 148

➤ **Close.**






**Maintenance Plan Wizard Progress**

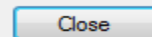
Click Stop to interrupt the operation.



 <b>Success</b>	5 Total	0 Error
	5 Success	0 Warning

Details:

Action	Status	Message
 Creating maintenance plan "Maintenanc...	Success	
 Adding tasks to the maintenance plan	Success	
 Adding scheduling options	Success	
 Adding reporting options	Success	
 Saving maintenance plan "Maintenance..."	Success	

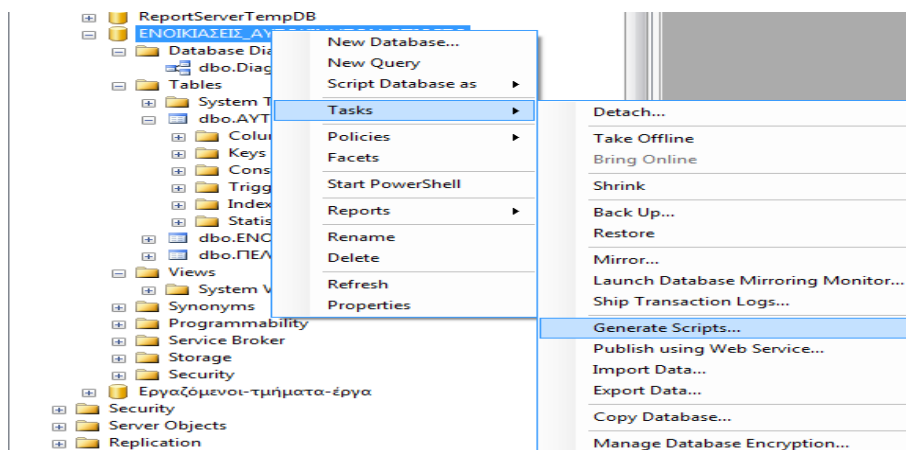


Εικόνα 149

## Εξαγωγή Βάσης Δεδομένων σε Scripts

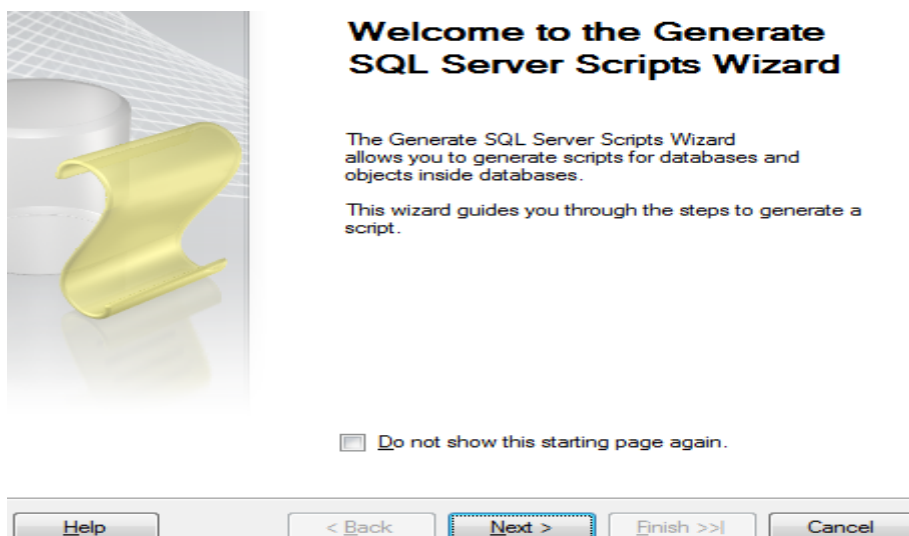
Για την εξαγωγή των δεδομένων σε **scripts** η διαδικασία που ακολουθείτε είναι η εξής:

**Object Explorer**→**Δεξί κλικ στη βάση (ΕΝΟΙΚΙΑΣΕΙΣ\_ΑΥΤΟΚΙΝΗΤΩΝ)**  
→**Tasks**→**Generate Scripts.**



Εικόνα 150

➤ **Next.**

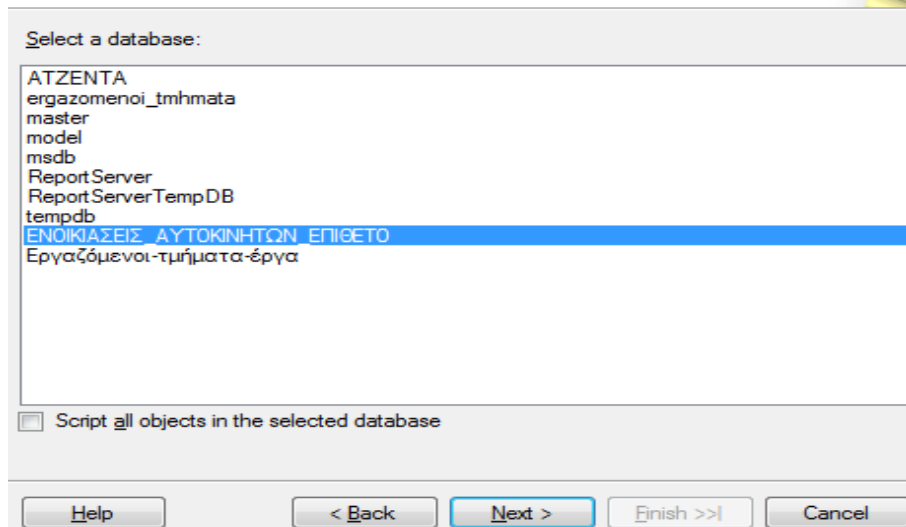


Εικόνα 151

- Επιλέξτε βάση και **Next**.

### Select Database

Select the database you want to script.

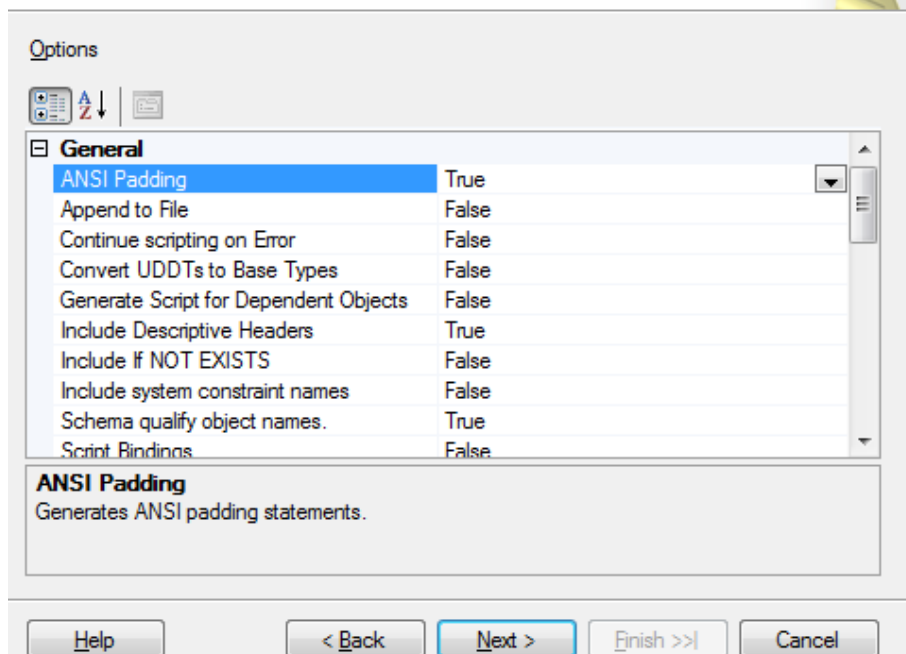


Εικόνα 152

- **Next**.

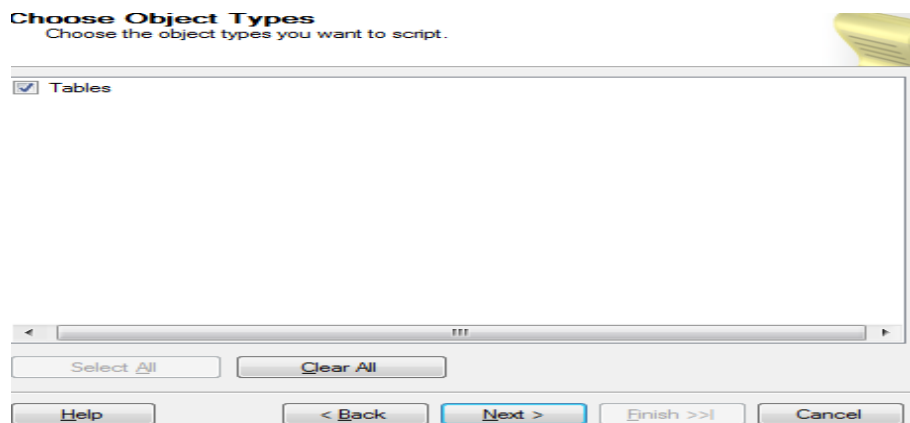
### Choose Script Options

Choose the options for the objects you want to script.



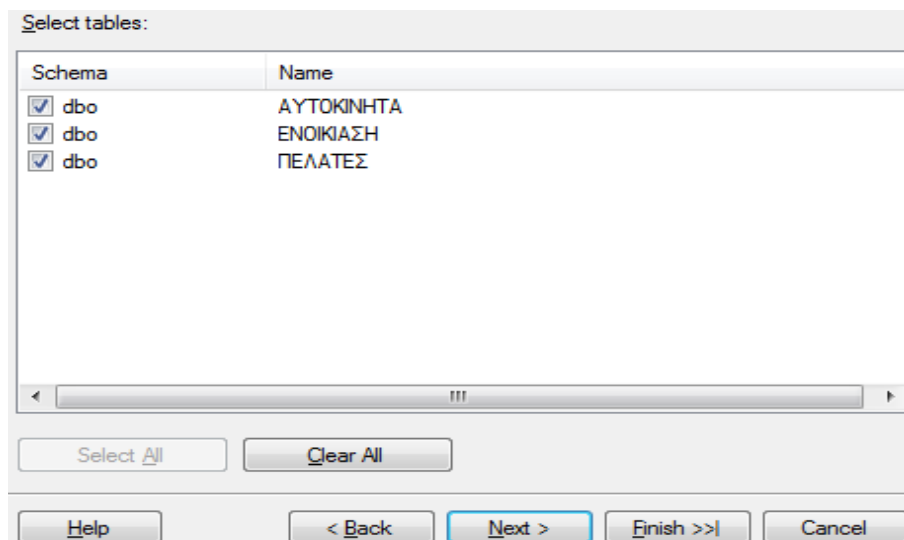
Εικόνα 153

- Επιλέξτε πίνακες ή οτιδήποτε περιέχει η βάση και θέλετε να το εξαγάγετε.



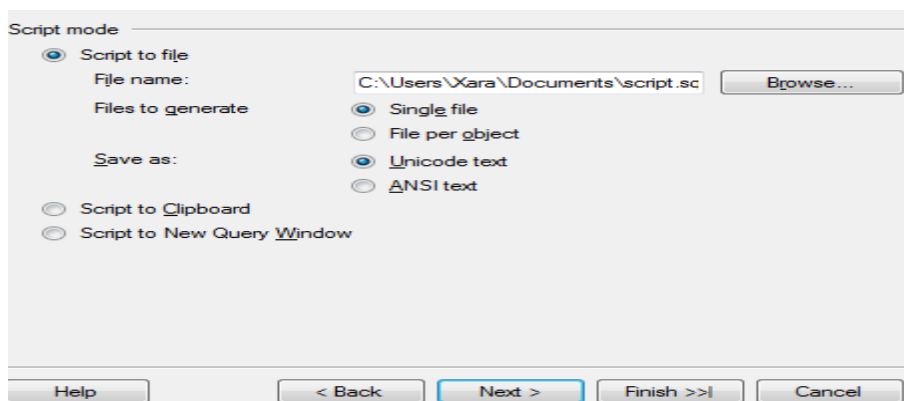
Εικόνα 154

- Επιλέγετε τους τρεις πίνακες και **Next**.



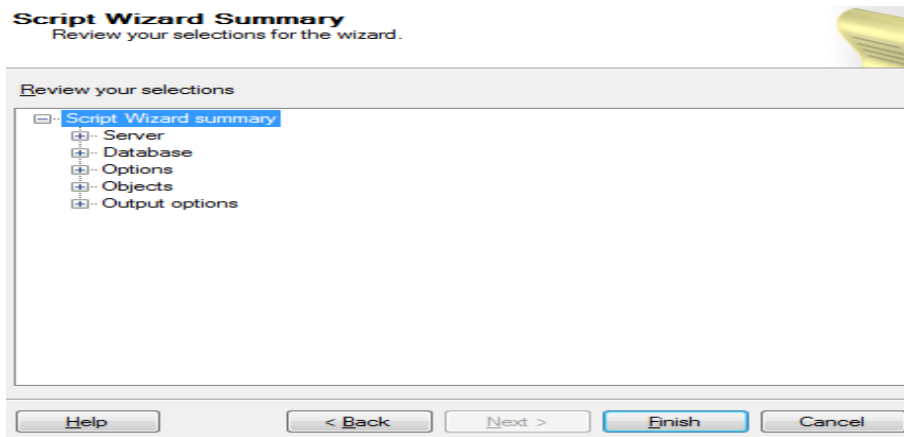
Εικόνα 155

- Δώστε την ακριβή τοποθεσία αποθήκευσης της εξαγωγής και **Next**.



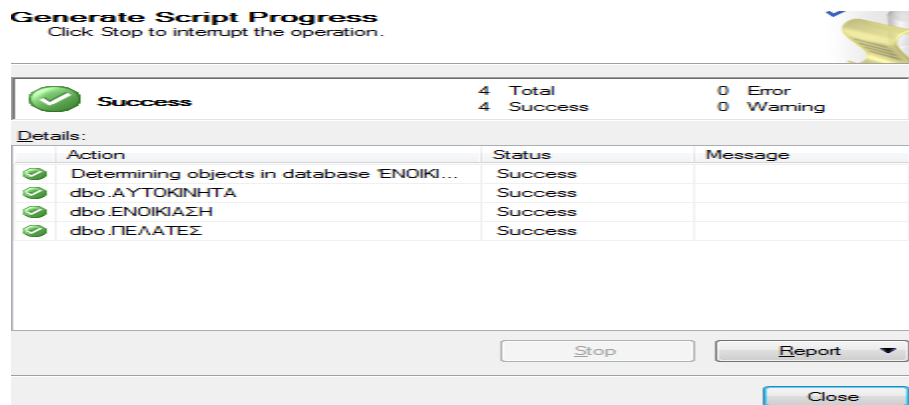
Εικόνα 156

➤ **Finish.**



Εικόνα 157

➤ **Close.**



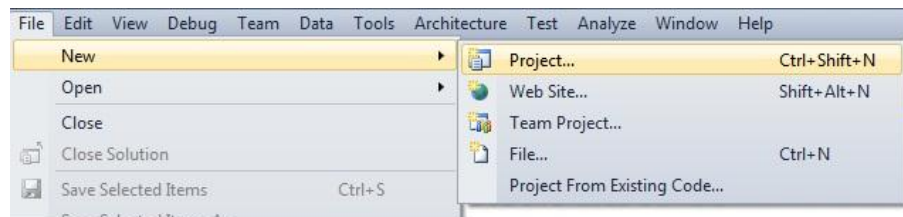
Εικόνα 158

## Άσκηση 4 - Δημιουργία Εφαρμογής

Σκοπός της εργαστηριακής άσκησης είναι η δημιουργία μιας εφαρμογής πολλαπλών καρτελών με κοινή σύνδεση σε Βάση Δεδομένων και την προβολή των δεδομένων από διαφορετικούς πίνακες της ίδιας Βάσης.

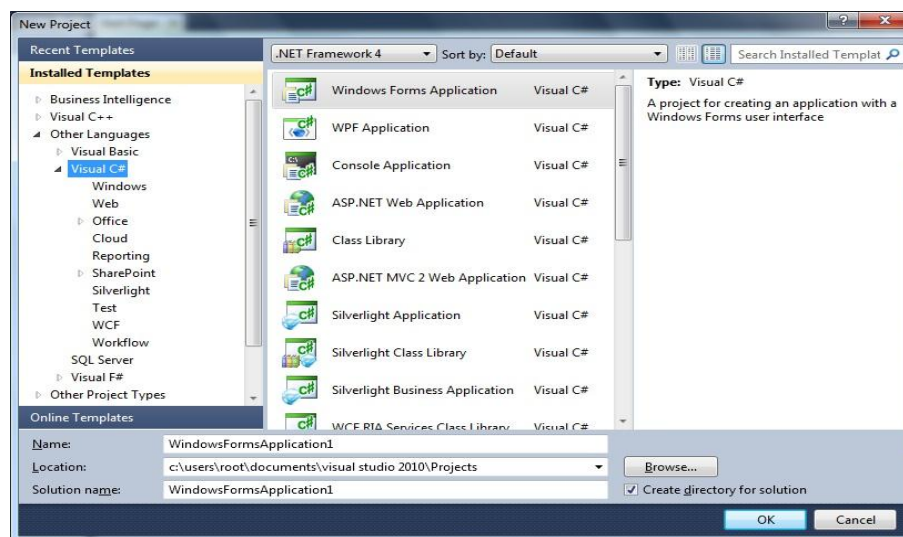
Ανοίξτε το Microsoft Visual Studio και δημιουργήστε ένα νέο project.

- **File→New→Project**



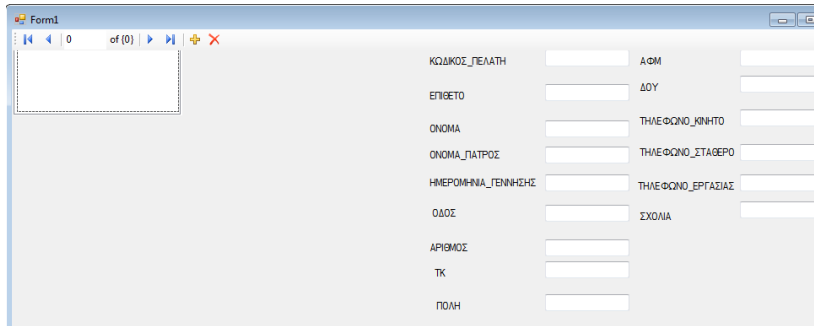
Εικόνα 159

- Έπειτα επιλέξτε την γλώσσα προγραμματισμού C# με την οποία θα ασχοληθείτε. **Other Languages -> Visual C# -> Windows Forms Application** και **OK**.



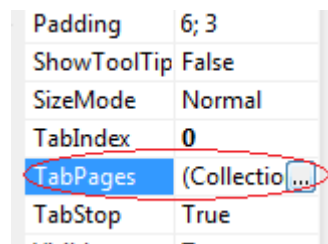
Εικόνα 160

- Στην φόρμα που δημιουργήθηκε εισάγετε από το Toolbox δεξιά με Drag and Drop τα εξής Components:
  - 3x BindingNavigator
  - 35x Labels
  - 35x TextBox
  - 1x TabControl



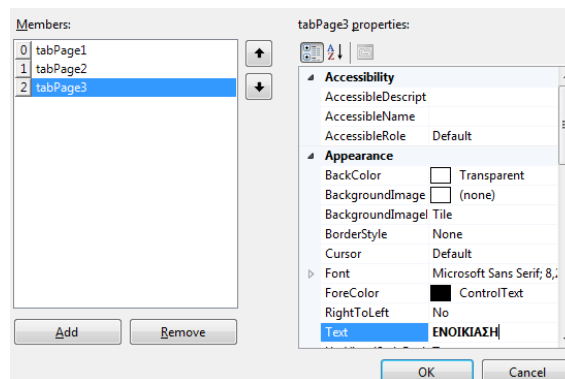
Εικόνα 161

- Στη συνέχεια επιλέξτε το tabcontrol1 και στα properties στο Tabpages κάνετε κλικ στις τελείες.



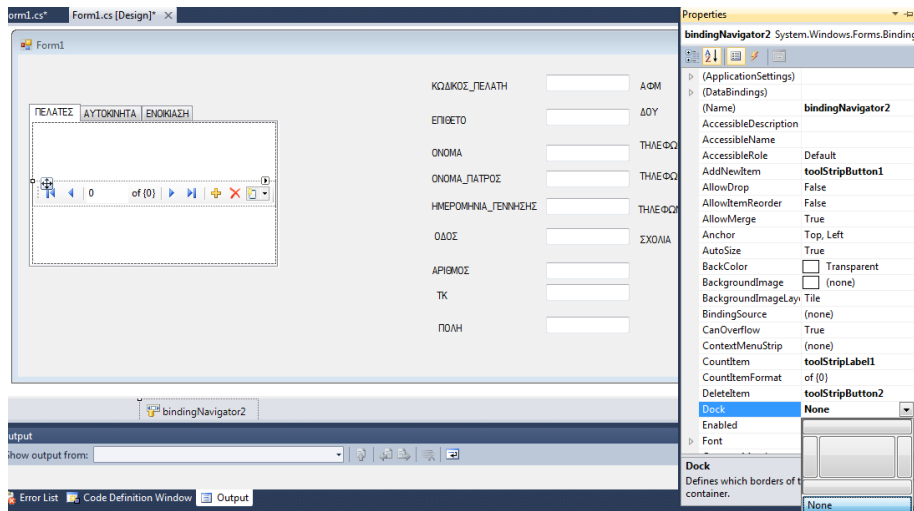
Εικόνα 162

- Αφού γίνει αυτό ανοίγει ένα νέο παράθυρο στο οποίο μπορείτε να ονομάσετε τα Tabpages καθώς και να εισάγετε και άλλα. Στην τρέχουσα εργαστηριακή άσκηση θα χρειαστούν τρία Tabpages με ονόματα ΠΕΛΑΤΕΣ, ΕΝΟΙΚΙΑΣΗ, ΑΥΤΟΚΙΝΗΤΑ.



Εικόνα 163

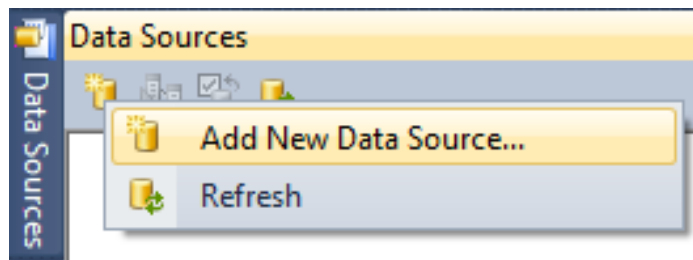
- Στη συνέχεια παίρνετε το BindingNavigator και το βάζετε μέσα στην καρτέλα ΠΕΛΑΤΕΣ, με το που θα το τοποθετήσετε θα μπει επάνω, γιατί αυτή είναι η default θέση του. Για να την αλλάξετε πηγαίνετε στα properties του BindingNavigator και στο Dock κάντε την τιμή του **None**.



Εικόνα 164

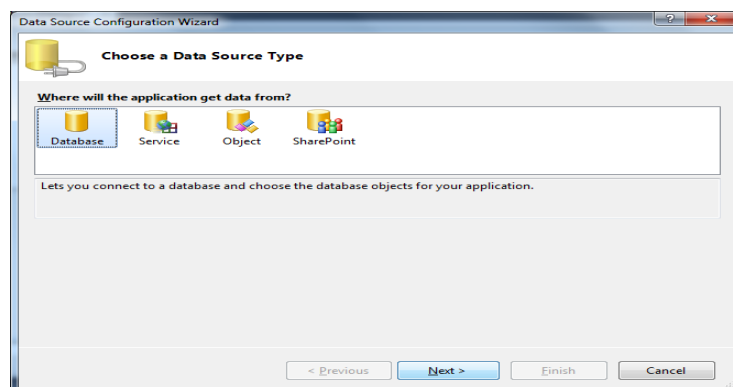
Όπως βλέπετε το πρόγραμμα σας έχει πάρει κάποια μορφή, ήρθε η ώρα λοιπόν να κάνετε την σύνδεση με την βάση η οποία γίνεται με τον εξής τρόπο:

- Πηγαίνετε δεξιά στο **Data Source** δεξί κλικ → **Add New Data Source**.



Εικόνα 165

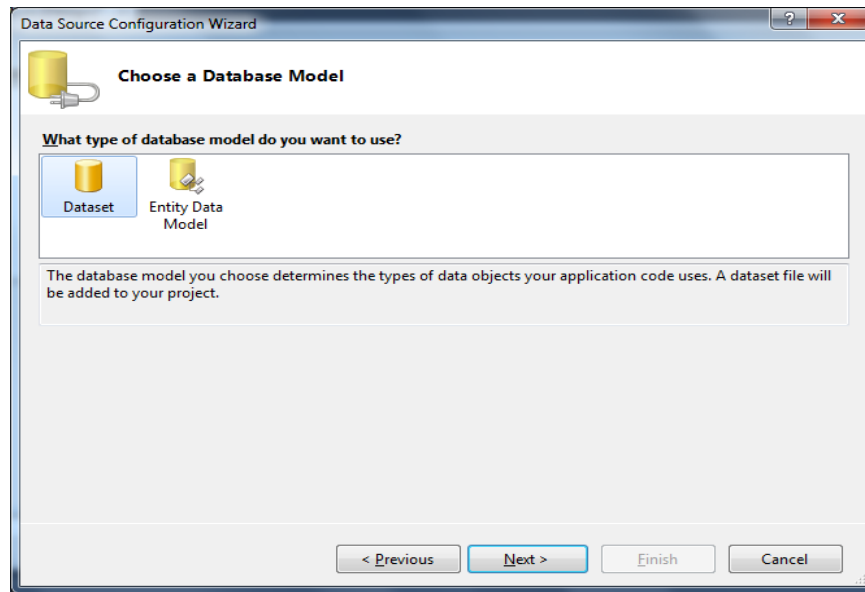
- Επιλέγετε **Database** και **Next**.



Εικόνα 166

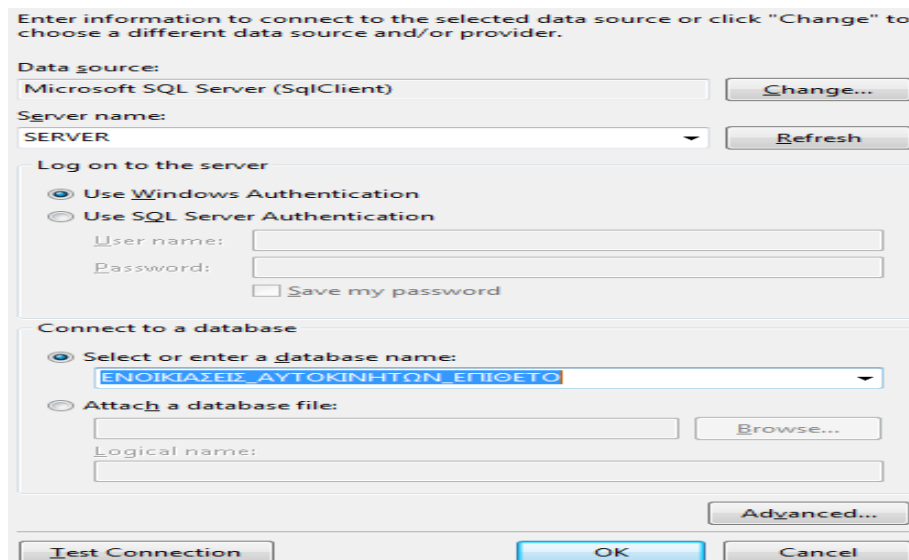


- Επιλέγετε **Dataset** και **Next**.



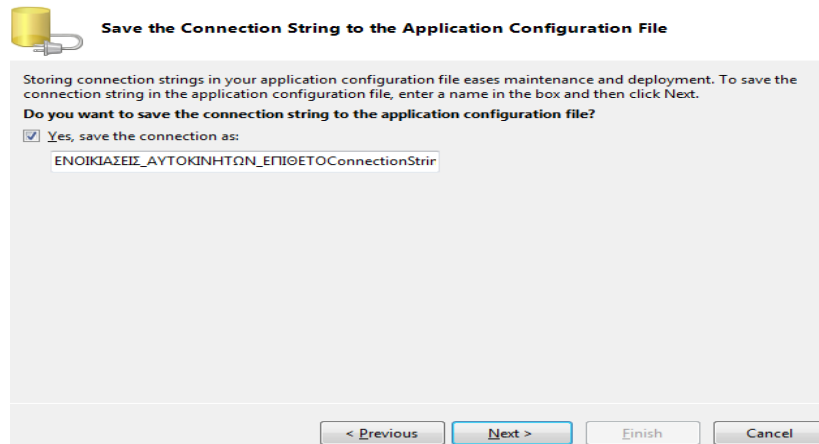
Εικόνα 167

- Ορίζετε την πηγή από την οποία θα καλέσει την βάση στο Data Source, το όνομα του Server στο Server Name, τσεκάρετε στο Log On to the server το Use windows authentication και τέλος ορίζετε την βάση που θα συνδέσει στο Connect a Database και **OK**.



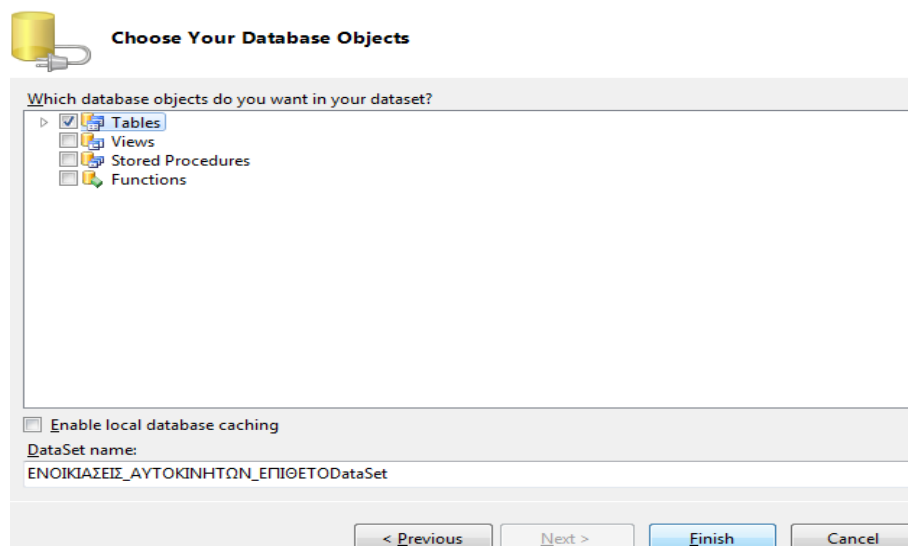
Εικόνα 168

➤ **Next**



Εικόνα 169

- Επιλέξτε τι θέλετε να πάρετε από την βάση. Στην συγκεκριμένη πάρτε μόνο τους πίνακες και Finish.



Εικόνα 170

- Τέλος πηγαίνετε στον Server Explorer ανοίξτε το Data Connections, κάντε **δεξί κλικ στην βάση → Properties**. Ψάξτε στα properties για το **connection String** και αντιγράψτε το.  
Στη συνέχεια τοποθετήστε το μέσα στην φόρμα κάνοντας διπλό κλικ πάνω της ώστε να ανοίξει ο κώδικας και γράψτε τα παρακάτω ώστε να ολοκληρωθεί η σύνδεση.

```

using System.Windows.Forms;
using System.Data.SqlClient;

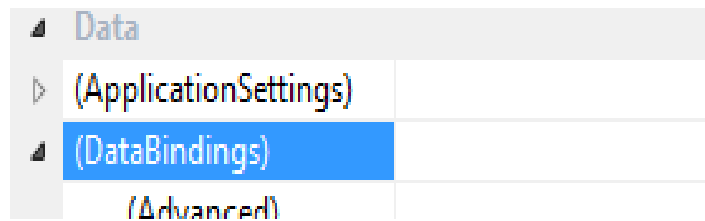
namespace askisi4
{
    public partial class Form1 : Form
    {
        SqlConnection connection;
        SqlDataAdapter DataAdapter1, DataAdapter2, DataAdapter3;
        DataSet DataSet1, DataSet2, DataSet3;
        BindingSource BindingSource1, BindingSource2, BindingSource3;

        public Form1()
        {
            InitializeComponent();
            connection = new SqlConnection("Data Source=SERVER;Initial Catalog=ΕΘΝΙΚΙΑΣΕΙΣ_ΑΥΤΟΚΙΝΗΤΩΝ_ΕΠΙΘΕΤΟ;Integrated Security=True");
            connection.Open();
        }
    }
}

```

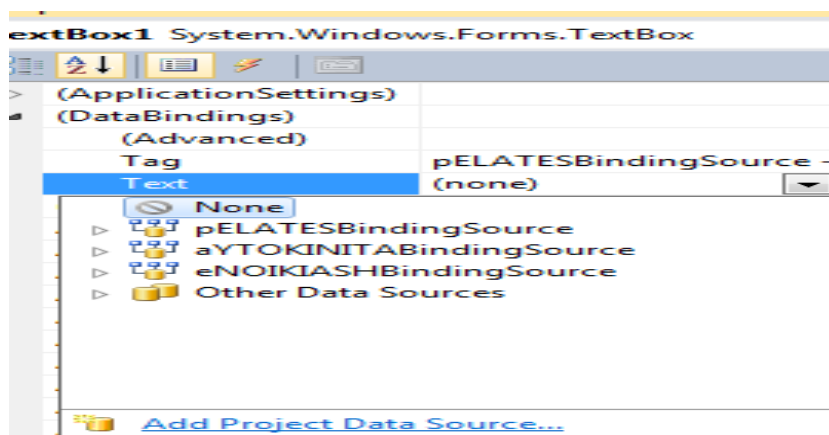
Εικόνα 171

- Στη συνέχεια επιλέξτε το textbox1 δεξί κλικ→properties→ DataBindings και ανοίγετε την δενδρική δομή.



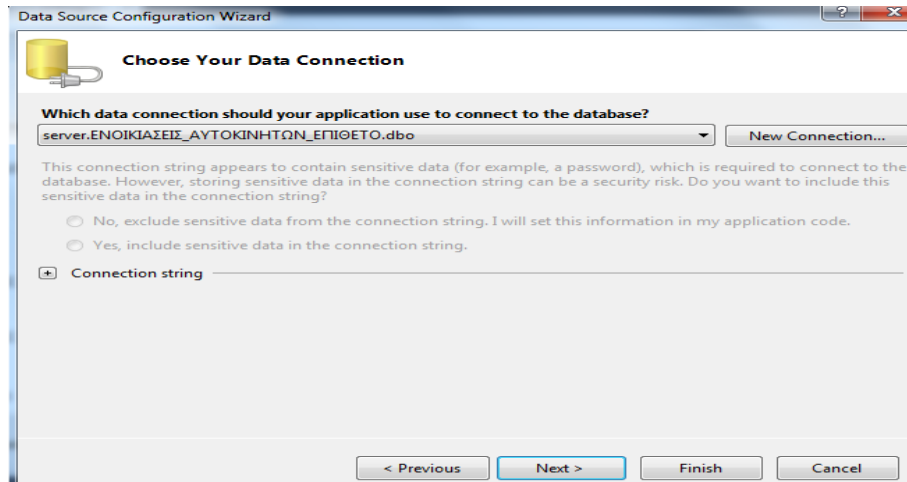
Εικόνα 172

- Αλλάξτε την τιμή του Text που είναι None και αντί αυτής εισάγετε στην βάση Add Project Data Source.



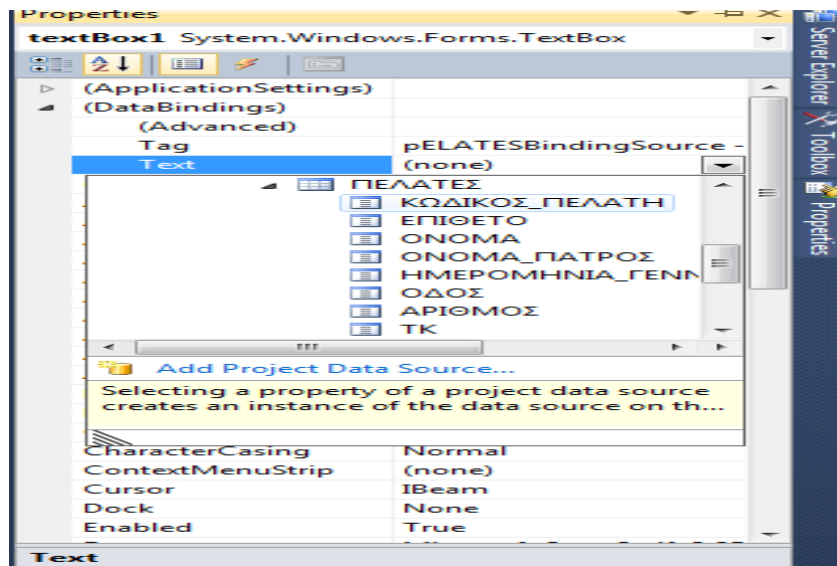
Εικόνα 173

- Ανοίγει ένα νέο παράθυρο πατήστε **Next** επιλέξτε την βάση ΕΝΟΙΚΙΑΣΕΙΣ\_ΑΥΤΟΚΙΝΗΤΩΝ\_ΕΠΙΘΕΤΟ στην οποία θέλετε να συνδεθείτε και πριν να πατήσετε Finish μαρκάρετε τους τρις πίνακες της βάσης.



Εικόνα 174

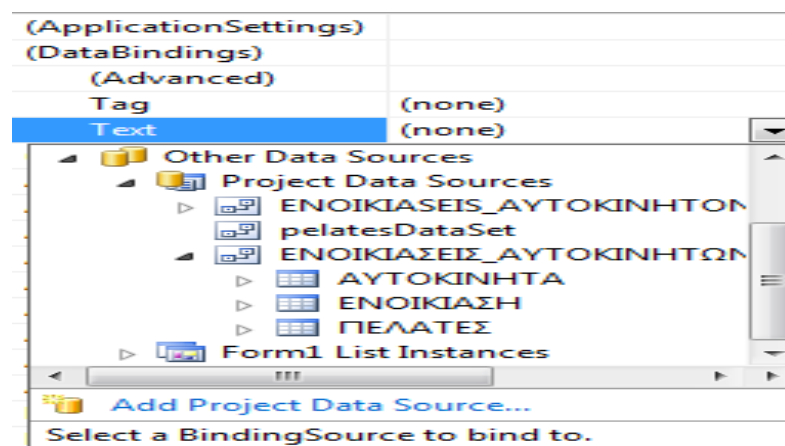
- Αφού γίνει η σύνδεση πηγαίνετε στον πίνακα ΠΕΛΑΤΕΣ επιλέξτε το TextBox και ακλουθώντας την ίδια διαδικασία με πριν **δεξί κλικ**→**properties**→**DataBindings** αλλάζετε την τιμή None και επιλέγετε τι θέλετε να σας εμφανίσει το textbox1.



Εικόνα 175

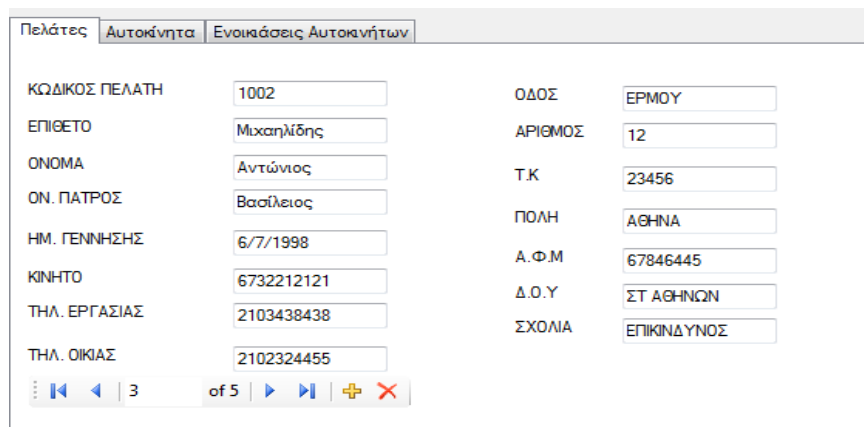
Συνεχίζετε ομοίως με τον ίδιο τρόπο για όλα τα textbox της καρτέλας αντιστοιχώντας τα με τα κατάλληλα πεδία της φόρμας. Αφού γίνει και η ρύθμιση των TextBox σειρά έχει τώρα το BindingNavigator.

- Επιλέξτε το BindingNavigator και στα Properties ανοίξτε την δενδρική δομή του DataSource. Στο Text ορίστε με ποιον πίνακα θα είναι συνδεδεμένο. Η διαδικασία είναι όμοια με εκείνη που κάνατε για τα TextBox με την μόνη διαφορά ότι τώρα δεν χρειάζεται να κάνετε **Add Project Data Source** αφού το είχατε κάνει για τα TextBox.



Εικόνα 176

Μόλις ολοκληρώσετε όλες τις παραπάνω διαδικασίες το πρόγραμμα σας θα πρέπει να τρέχει κάπως έτσι.



ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	1002	ΟΔΟΣ	ΕΡΜΟΥ
ΕΠΙΘΕΤΟ	Μιχαηλίδης	ΑΡΙΘΜΟΣ	12
ΟΝΟΜΑ	Αντώνιος	Τ.Κ	23456
ΟΝ. ΠΑΤΡΟΣ	Βασίλειος	ΠΟΛΗ	ΑΘΗΝΑ
ΗΜ. ΓΕΝΝΗΣΗΣ	6/7/1998	Α.Φ.Μ	67846445
ΚΙΝΗΤΟ	6732212121	Δ.Ο.Υ	ΣΤ ΑΘΗΝΩΝ
ΤΗΛ. ΕΡΓΑΣΙΑΣ	2103438438	ΣΧΟΛΙΑ	ΕΠΙΚΙΝΔΥΝΟΣ
ΤΗΛ. ΟΙΚΙΑΣ	2102324455		

Εικόνα 177

Πελάτες	Αυτοκίνητα	Ενοικιάσεις Αυτοκινήτων	
ΚΩΔΙΚΟΣ ΑΥΤΟΚΙΝΗΤΟΥ	1005	ΤΥΠΟΣ ΚΑΥΣΙΜΟΥ	BENZINH
ΜΑΡΚΑ	MERCEDES	ΗΜ. ΚΥΚΛΟΦΟΡΙΑΣ	1/6/2006
ΜΟΝΤΕΛΟ	SLK	ΧΙΛΙΟΜΕΤΡΑ	12000
ΑΡ. ΚΥΚΛΟΦΟΡΙΑΣ	45677		
ΘΕΣΕΙΣ	2		
ΠΟΡΤΕΣ	2		
ΧΡΩΜΑ	ΑΣΗΜΙ		
ΤΙΜΗ ΕΝΟΙΚΙΑΣΗΣ	70		

Εικόνα 178

Πελάτες	Αυτοκίνητα	Ενοικιάσεις Αυτοκινήτων	
ΚΩΔΙΚΟΣ ΕΝΟΙΚΙΑΣΗΣ	3	ΠΟΣΟ ΠΛΗΡΩΜΗΣ	420
ΚΩΔΙΚΟΣ ΠΕΛΑΤΗ	2	ΠΕΡ. ΖΗΜΙΑΣ	
ΚΩΔΙΚΟΣ ΑΥΤΟΚΙΝΗΤΟΥ	1005	ΧΡΕΩΣΗ ΖΗΜΙΑΣ	0
ΗΜ. ΕΝΟΙΚΙΑΣΗΣ	20/9/2007	ΤΕΛΙΚΟ ΠΟΣΟ	420
ΗΜ. ΕΠΙΣΤΡΟΦΗΣ	25/9/2007		

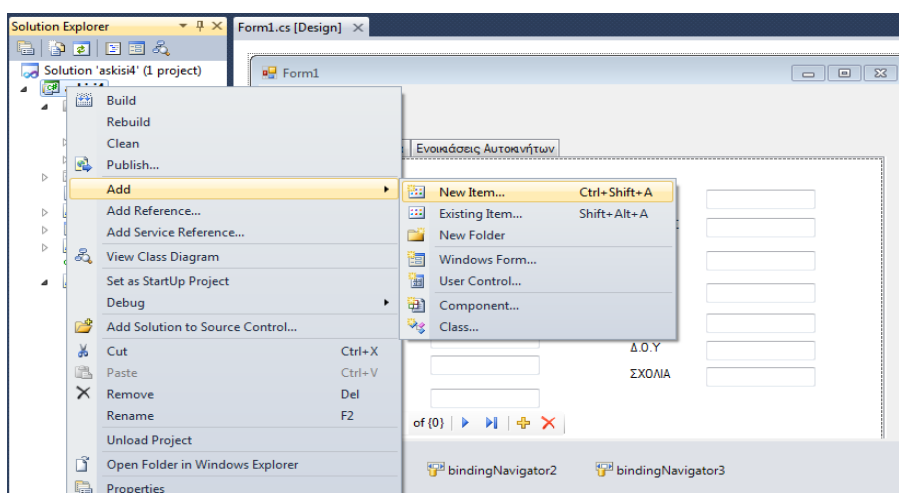
Εικόνα 179

## Άσκηση 5 - Εκτέλεση ερωτημάτων σε Visual Studio 2010

Σκοπός αυτής της εργαστηριακής άσκησης είναι η επίδειξη και η εκτέλεση δυναμικών ερωτημάτων μέσα από το προγραμματιστικό περιβάλλον του Microsoft Visual Studio 2010. Η εφαρμογή αυτή είναι η συνέχεια της προηγούμενης εργαστηριακής άσκησης.

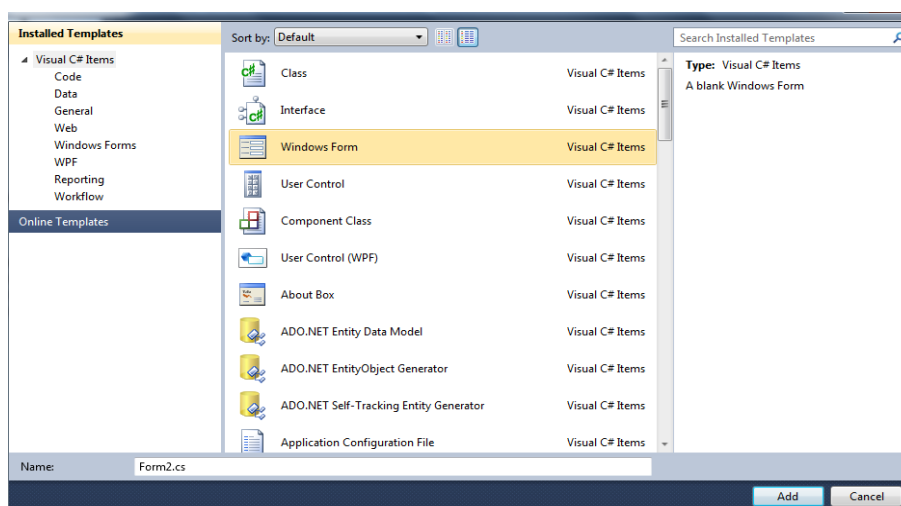
Ανοίξτε τον κώδικα της εργαστηριακής άσκησης 4 και εισάγετε δύο επιπλέον φόρμες με την εξής διαδικασία:

- Δεξί κλικ στην άσκηση →Add →New Item.



Εικόνα 180

- Visual C# Items→Windows Form.



Εικόνα 181

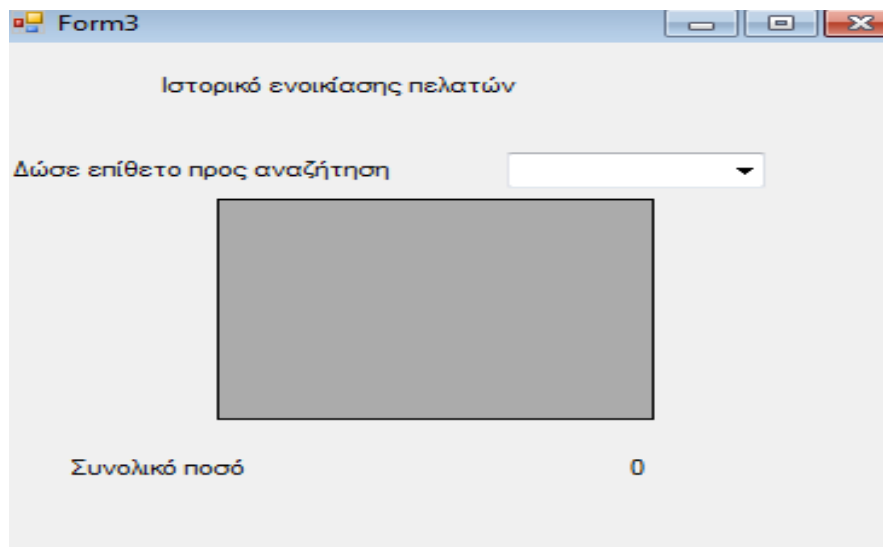
Ομοίως εισάγετε και την δεύτερη φόρμα. Η πρώτη φόρμα θα εμπεριέχει:

- 1x Button

Ενώ η στη δεύτερη θα εμπεριέχει:

- 4x Label
- 1x ComboBox
- 1x DataGridView

➤ Οι τοποθέτηση των Components γίνεται σύμφωνα με την Εικόνα 101.



Εικόνα 182

➤ Αρχικά μεταφερθείτε στην φόρμα που εμπεριέχει το Button. Με διπλό κλικ στο Button ανοίγει ο κώδικας όπου θα γράψετε τα παρακάτω ώστε να μπορεί να ανοίξει την φόρμα 3.

```
using System.Windows.Forms;

namespace askisi4
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Form3 frm = new Form3();
            frm.Show();
        }
    }
}
```

Εικόνα 183



- Στη συνέχεια ανοίξτε τον κώδικα της φόρμας 3 με διπλό κλικ και εισάγετε τα παρακάτω ώστε να μπορέσει να εκτελέσει το ερώτημα για την εύρεση των πελατών.

```

using System.Windows.Forms;
using System.Data.SqlClient;

namespace askisi5
{
    public partial class Form2 : Form
    {
        SqlConnection connection;
        SqlDataAdapter DataAdapter1, DataAdapter2;
        DataSet DataSet1, DataSet2;
        BindingSource BindingSource1, BindingSource2;

        public Form2()
        {
            InitializeComponent();
            connection = new SqlConnection("Data Source=.....");
            connection.Open();

            DataAdapter1 = new SqlDataAdapter("Select * from ΠΕΛΑΤΕΣ", connection);
            DataTable dt1 = new DataTable();
            DataAdapter1.Fill(dt1);
            comboBox1.DataSource = dt1;
            comboBox1.DisplayMember = "ΕΠΙΘΕΤΟ";
        }
    }
}

```

Εικόνα 184

**ΣΧΟΛΙΟ!** Στο Data Source Θα εισάγετε το Source του Connection String το οποίο το παίρνετε ως εξής: **Server Explorer** → **Δεξί κλικ στη βάση** → **Properties** → **Connection String** → Αντιγράψτε το και επικολλήστε το στο **new SqlConnection** του κώδικα.

```

private void Form2_Load(object sender, EventArgs e)
{
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    fillDataSet();
    // ΣΥΝΑΡΤΗΣΗ Η ΟΠΟΙΑ ΓΕΜΙΖΕΙ ΤΟ combobox1
}

public void fillDataSet() // Ο ΚΩΔΙΚΑΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ
{
    DataAdapter2 = new SqlDataAdapter("SELECT ΜΑΡΚΑ,ΜΟΝΤΕΛΟ,ΧΡΩΜΑ,ΗΜΕΡΟΜΗΝΙΑ_ΕΝΟΙΚΙΑΣΗΣ,ΗΜΕΡΟΜΗΝΙΑ_ΕΠΙΣΤΡΟΦΗΣ,ΠΕΡΙΓΡΑΦΗ_ΖΗΤΙΑΣ,
    ΤΕΛΙΚΟ_ΠΟΣΟ FROM ΠΕΛΑΤΕΣ INNER JOIN (ΕΝΟΙΚΙΑΣΗ INNER JOIN ΑΥΤΟΚΙΝΗΤΑ ON ΕΝΟΙΚΙΑΣΗ.ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ=ΑΥΤΟΚΙΝΗΤΑ_ΚΩΔΙΚΟΣ_ΑΥΤΟΚΙΝΗΤΟΥ)
    ON ΠΕΛΑΤΕΣ_ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ=ΕΝΟΙΚΙΑΣΗ_ΚΩΔΙΚΟΣ_ΠΕΛΑΤΗ WHERE ΠΕΛΑΤΕΣ_ΕΠΙΘΕΤΟ='"+comboBox1.Text.ToString()+"'.....connection);");
    DataSet2 = new DataSet();
    DataAdapter2.Fill(DataSet2);
    BindingSource2 = new BindingSource();
    DataTable dt = new DataTable();
    BindingSource2.DataSource = DataSet2.Tables[0].DefaultView;
    dataGridView1.DataSource = BindingSource2;
    int sum = 0; // ΟΡΙΖΟΥΜΕ ΤΟ SUM=0
}

```

**ΣΧΟΛΙΟ!** Ο κώδικας στο DataAdapter2 πρέπει να είναι στην ίδια γραμμή αλλιώς θα είναι εσφαλμένος και το πρόγραμμά σας δεν θα τρέξει. Αν παρατηρήσετε τα γράμματα είναι κόκκινα στην γραμμή του DataAdapter2 ενώ στην ακριβώς από κάτω είναι μαύρα αυτό γίνεται γιατί δεν τα διαβάζει σωστά το πρόγραμμα.

```

for (int i = 0; i < dataGridView1.Rows.Count; i++)
{
    //object o = this.dataGridView1.Rows[i].Cells[6].Value;

    sum += Convert.ToInt32(dataGridView1.Rows[i].Cells[6].Value);
}
label4.Text = sum.ToString();

```

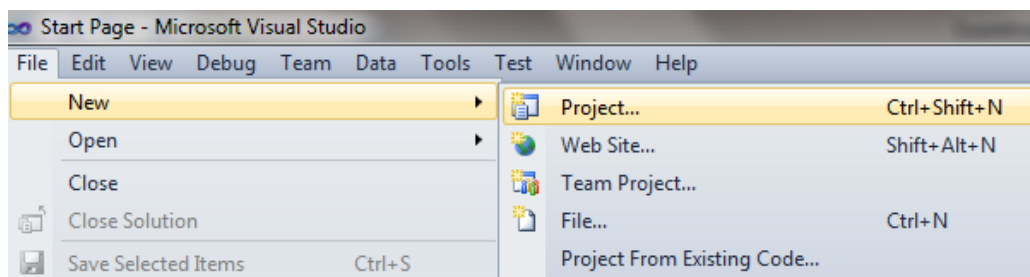
Εικόνα 185

Τέλος εκτελέστε το πρόγραμμα σας . Αφού γίνει η εκτέλεση η φόρμα 3 θα κάνει αναζήτηση πελατών.

## Άσκηση 6 – Εκτέλεση διαδοχικών ερωτημάτων

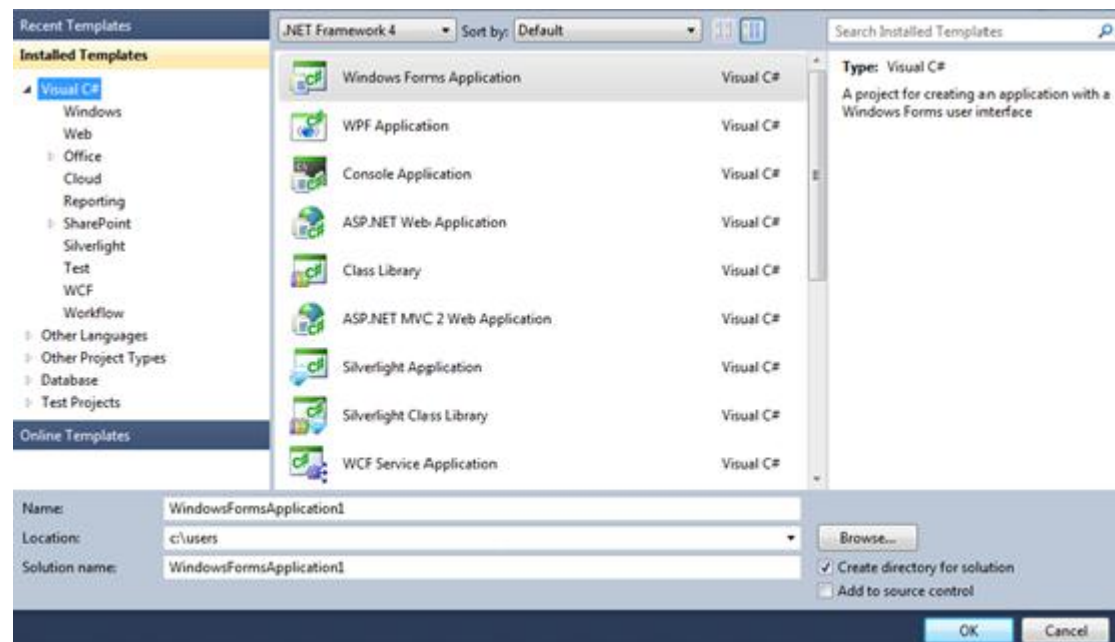
**Σκοπός** της εργαστηριακής άσκησης είναι η εξοκείωση του σπουδαστή με την εκτέλεση διαδοχικών δυναμικών ερωτημάτων που η έξοδος του ενός γίνεται είσοδος για το επόμενο ερώτημα. Συγκεκριμένα, ο χρήστης θα επιλέγει διαδοχικά ΜΑΡΚΑ, ΜΟΝΤΕΛΟ, ΧΡΩΜΑ και ΤΙΜΗ και θα εμφανίζονται τα διαθέσιμα αυτοκίνητα με αυτά τα κριτήρια.

Εκκινώντας το **Microsoft Visual Studio** πρέπει να δημιουργήσουμε ένα νέο Project (Solution) στο οποίο θα εργαστούμε. Πάμε λοιπόν **File → New → Project**.



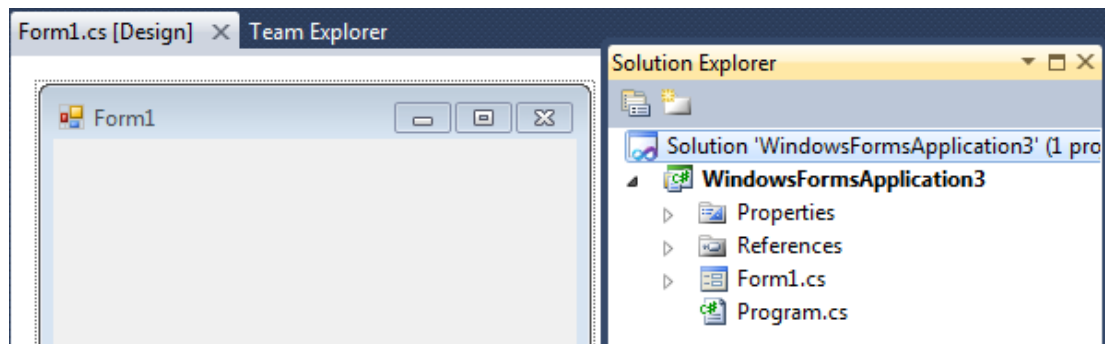
Εικόνα 1

Η Γλώσσα Προγραμματισμού με την οποία θα ασχοληθούμε είναι η Visual C#. Έτσι επιλέγουμε **Visual C# → Windows Forms Application → OK**.



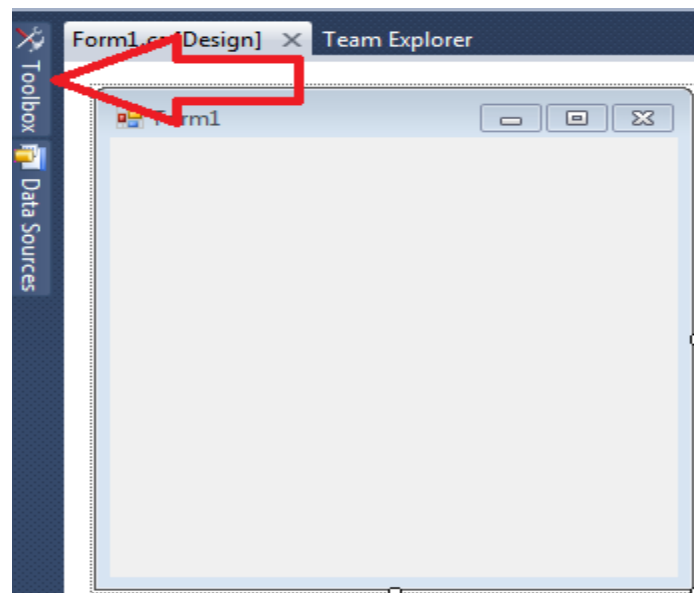
Εικόνα 2

Με τα παραπάνω βήματα έχει δημιουργηθεί η φόρμα μας (Form1).



Εικόνα 3

Προσθέτουμε τα components που χρειαζόμαστε για την άσκηση. Όλα τα Components βρίσκονται στο εσωτερικό του **Toolbox**.

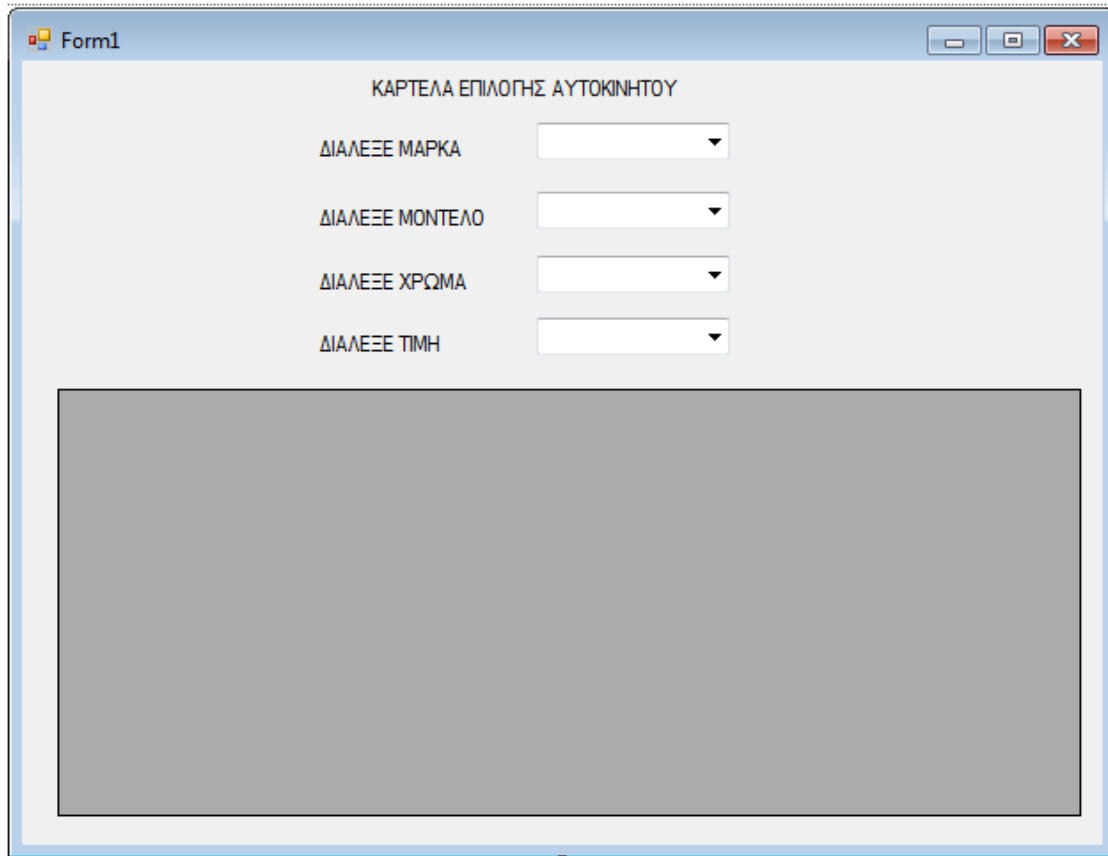


Εικόνα 4

Τα Components που θα χρειαστούν είναι :

- 5x Labels
- 4x ComboBox
- 1x DataGridView

Ανοίγοντας το Toolbox και κάνοντας Drag and Drop στη φόρμα μας τα Components που χρειαζόμαστε ώστε να εμφανιστεί το παρακάτω αποτέλεσμα.

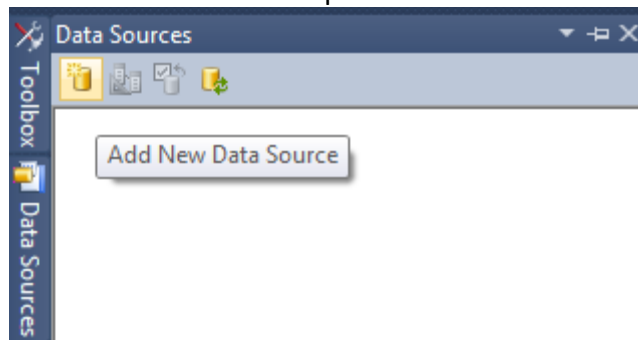


Εικόνα 5

Αφού έχουμε δώσει μορφή στο πρόγραμμα τώρα θα συνδέσουμε τη Βάση Δεδομένων με αυτό.

Ακολουθούμε τα εξής βήματα :

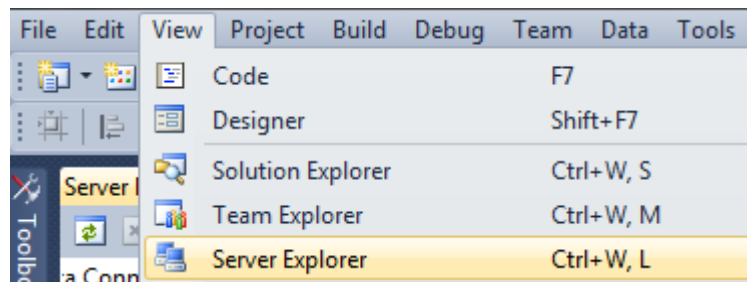
- ✓ Πηγαίνουμε στο **Data Source** και πατάμε **Add New Data Source**.



Εικόνα 6

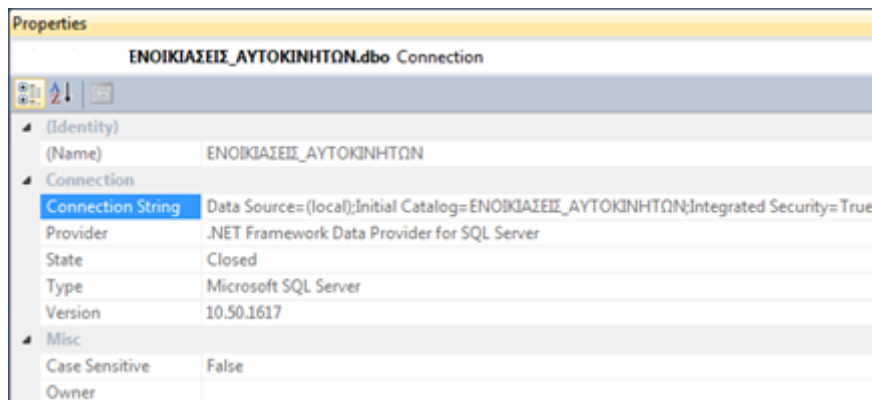
- ✓ Επιλέγουμε **Database → Next, Dataset → Next.**
- ✓ Επιλέγουμε **New Connection**
- ✓ **Data Source → Microsoft SQL Server → Server Name** το όνομα του Server που κάνουμε login στο Management Studio, επιλέγουμε την Βάση μας και πατάμε ok.
- ✓ Server name= Το όνομα που έχετε δώσει στον server σας.
- ✓ Database name= ΕΝΟΙΚΙΑΣΕΙΣ\_ΑΥΤΟΚΙΝΗΤΩΝ→**OK.**
- ✓ Στην συνέχεια επιλέγουμε **Next → Next → Finish**

Επιλέγετε από την κορδέλα **View → Server Explorer**



Εικόνα 7

Δεξί κλικ στη βάση μας → **Properties** και αντιγράψτε ότι υπάρχει στο πεδίο **Connection String**.



Εικόνα 8

Κάνοντας διπλό κλικ στην φόρμα εμφανίζει το περιβάλλον όπου θα γραφεί ο κώδικας. Προσέχουμε να μη παραλείψουμε κάτι.

Προσθέτουμε την εξής βιβλιοθήκη:

```
using System.Data.SqlClient;
```

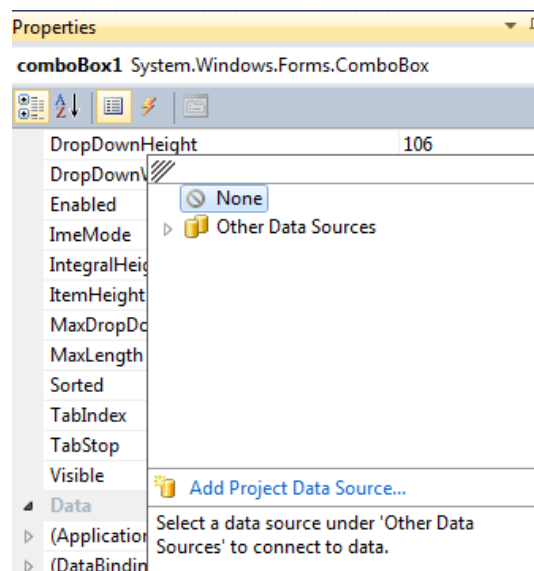
Στον constructor ορίζουμε τα αντικείμενα που θα μας χρειαστούν στη φόρμα μας.

```
public partial class Form1 : Form
{
    SqlConnection connection;
    SqlDataAdapter DataAdapter1, DataAdapter2, DataAdapter3, DataAdapter4, DataAdapter5;
    DataSet DataSet1, DataSet2;
    BindingSource BindingSource1, BindingSource2;
}
```

Κάνουμε επικύλιση το **Connection String** στο **new SqlConnection**.

```
public Form1()
{
    InitializeComponent();
    connection = new SqlConnection("Data Source=(local);Initial Catalog=ΕΜΟΙΚΙΑΣΕΙΣ_ΑΥΤΟΚΙΝΗΤΩΝ;Integrated Security=True");
    connection.Open();
}
```

Στη συνέχεια επιλέγουμε το **ComboBox1** και πηγαίνουμε στα **Properties** → **DataSource** → **Add Project Data Source**.

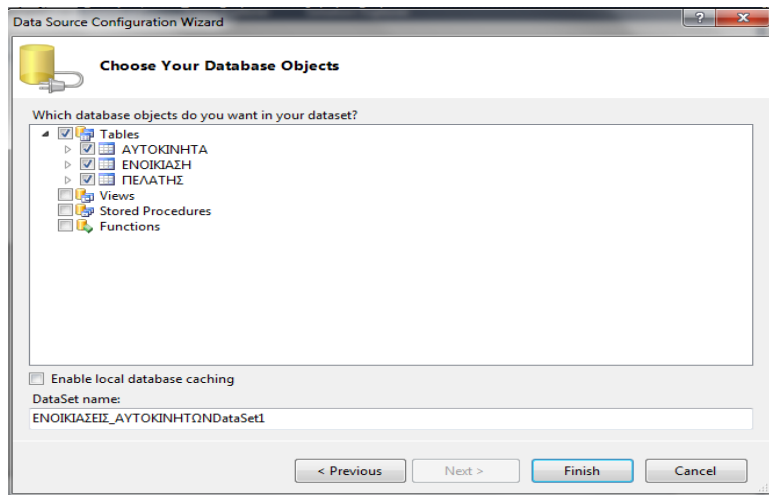


Εικόνα 9

✓ Επιλέγουμε **Database** → **Next**, **Dataset** → **Next**.

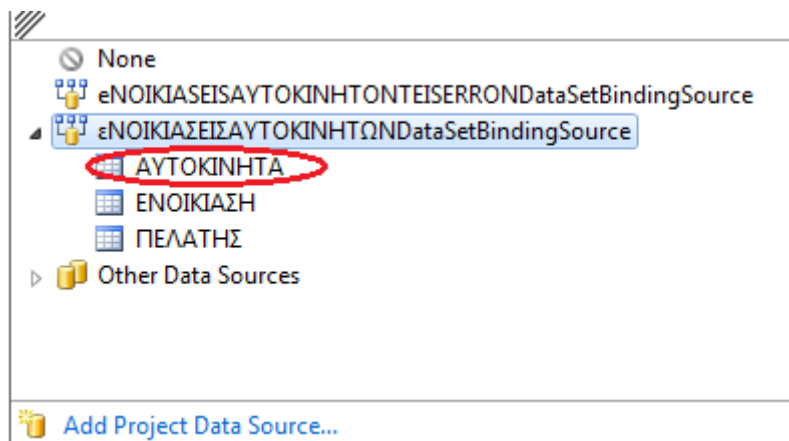
Ακολουθούμε τα ίδια βήματα όπως παραπάνω για να συνδέσουμε την Βάση που επιθυμούμε.

Πριν πατήσουμε **Finish** μαρκάρουμε **Tables**, βλέπουμε ότι έχουν επιλεχθεί και οι 3 πίνακές μας. Τέλος πατάμε **Finish**.



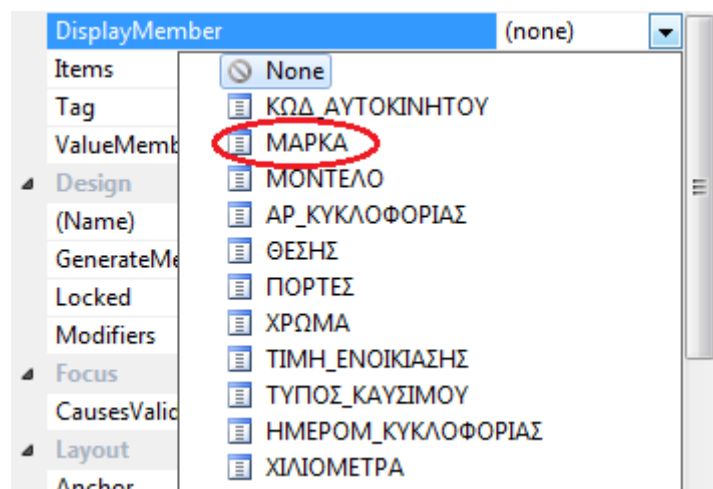
Εικόνα 10

Επιλέγουμε από το **DataSource** τον πίνακα **ΑΥΤΟΚΙΝΗΤΑ**.



Εικόνα 11

Στη συνέχεια από τα **Properties** στο **DisplayMember** επιλέγουμε το πεδίο **ΜΑΡΚΑ**.



Εικόνα 12

**Σημείωση:** Αυτό μπορούμε να το κάνουμε μόνο για το combobox1.  
Μετά μπαίνουμε μέσα στη φόρμα και γράφουμε τον εξής κώδικα:

```
private void Form1_Load(object sender, EventArgs e)
{
    this.αΥΤΟΚΙΝΗΤΑDataAdapter.Fill(this.εΝΟΙΚΙΑΣΕΙΣ_ΑΥΤΟΚΙΝΗΤΩΝDataSet.ΑΥΤΟΚΙΝΗΤΑ);

    BindingSource2 = new BindingSource();
    DataSet2 = new DataSet();

    DataAdapter1 = new SqlDataAdapter("Select MAPKA from ΑΥΤΟΚΙΝΗΤΑ", connection);
    DataSet1 = new DataSet();
    DataAdapter1.Fill(DataSet1);
    BindingSource1 =new BindingSource();
    BindingSource1.DataSource = DataSet1.Tables[0].DefaultView;

}
public void fillDataSet()
{
    String value = comboBox1.Text.ToString();
    DataAdapter2 = new SqlDataAdapter("Select * from ΑΥΤΟΚΙΝΗΤΑ where MAPKA=" + comboBox1.Text.ToString()+"" , connection);

    DataTable dt = new DataTable();
    DataAdapter2.Fill(dt);
    comboBox2.DataSource = dt;
    comboBox2.DisplayMember = "ΜΟΝΤΕΛΟ";

    DataAdapter3 = new SqlDataAdapter("Select * from ΑΥΤΟΚΙΝΗΤΑ where ΜΟΝΤΕΛΟ=" + comboBox2.Text.ToString()+ "" , connection);

    DataTable dt3 = new DataTable();
    DataAdapter3.Fill(dt3);
    comboBox3.DataSource = dt3;
    comboBox3.DisplayMember = "ΧΡΩΜΑ";

    DataAdapter4 = new SqlDataAdapter("Select * from ΑΥΤΟΚΙΝΗΤΑ where ΧΡΩΜΑ=" + comboBox3.Text.ToString()+"" , connection);

    DataTable dt4 = new DataTable();
    DataAdapter4.Fill(dt4);
    comboBox4.DataSource = dt4;
    comboBox4.DisplayMember = "ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ";


    DataAdapter5 = new SqlDataAdapter("Select * from ΑΥΤΟΚΙΝΗΤΑ where MAPKA=" + comboBox1.Text.ToString() + ""
and ΜΟΝΤΕΛΟ=' ' + comboBox2.Text.ToString() + ""
and ΧΡΩΜΑ= ' ' +comboBox3.Text.ToString() + ""
and ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ=' ' + comboBox4.Text.ToString() +"" , connection);

    DataSet2 = new DataSet();
    DataAdapter2.Fill(DataSet2);
    BindingSource2 =new BindingSource();
    BindingSource2.DataSource = DataSet2.Tables[0].DefaultView;
    dataGridView1.DataSource = BindingSource2;

}
}
```

**Σημείωση:** Στο **DataAdapter5** όλα θα πρέπει να είναι στην ίδια σειρά για να τρέξει ο κώδικας ή για να αλλάξετε γραμμή πρέπει στο τέλος να βάζετε τους χαρακτήρες +"\r\n"+. Δηλαδή για να σπάσει ένα string s σε δύο γραμμές γίνεται ως εξής:

```
string s = "One line" +
"\r\n" +
"Another line";
```

Τέλος πιάστε το πλήκτρο (run)  για να τρέξει η εφαρμογή και να εμφανιστούν τα τελικά μας αποτελέσματα.

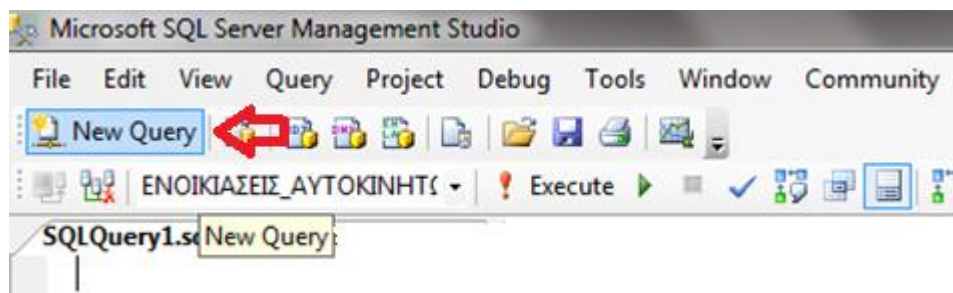


## Άσκηση 7 - Προβολή εικόνας σε φόρμα

**Σκοπός** της εργαστηριακής άσκησης είναι να διδαχθεί ο σπουδαστής να δημιουργεί πεδία σε πίνακα που να περιέχουν κάποιες διαδρομές αρχείων σε εικόνες και να προβάλει αυτές τις εικόνες μέσω μιας εφαρμογής C#.

Αρχικά θα προσθέσουμε ένα πεδία στη Βάση, του πίνακα «ΑΥΤΟΚΙΝΗΤΑ» μέσω ερωτήματος :


Ανοίγοντας το SQL Server Management Studio του Microsoft SQL Server 2008 R2 , επιλέγουμε τον πίνακα **ΑΥΤΟΚΙΝΗΤΑ** → **Desing** → **New Query** .



Εικόνα 1

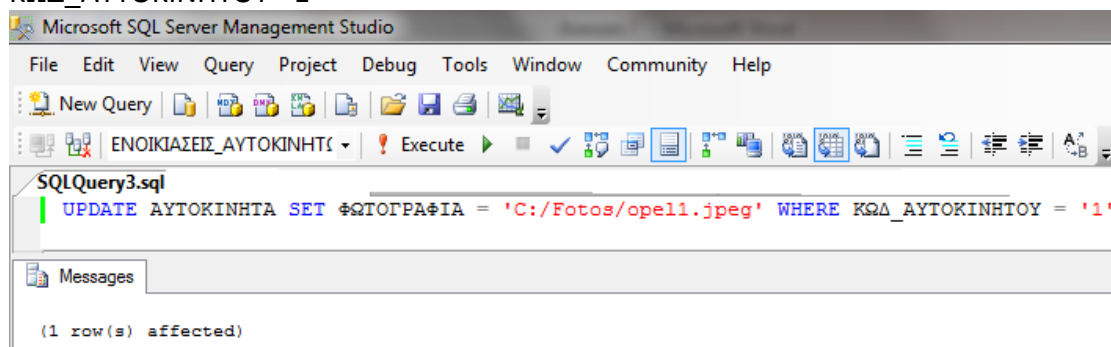
Γράφουμε το εξής ερώτημα :

```
ALTER TABLE ΑΥΤΟΚΙΝΗΤΑ ADD ΦΩΤΟΓΡΑΦΙΑ VARCHAR(100)
```

Το εκτελούμε πατώντας το πλήκτρο  **Execute** . Έστω ότι έχουμε έναν φάκελο με φωτογραφίες αυτοκινήτων (“C:/Photos”). Θέλουμε να αποθηκεύσουμε το path μιας εικόνας στην πρώτη εγγραφή εκτελώντας ένα ερώτημα.

Ακολουθούμε την προηγούμενη διαδικασία και γράφουμε το εξής:

```
UPDATE ΑΥΤΟΚΙΝΗΤΑ SET ΦΩΤΟΓΡΑΦΙΑ = 'C:/Fotos/opel1.jpeg' WHERE ΚΩΔ_ΑΥΤΟΚΙΝΗΤΟΥ='1'
```



Εικόνα 2

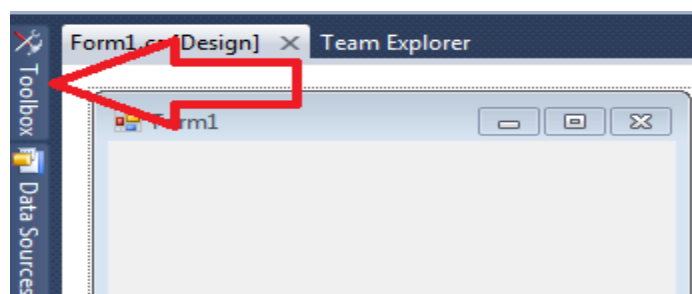
## Προγραμματισμός Εφαρμογής

Ανοίγουμε το **Visual Studio 2010** και δημιουργούμε ένα νέο solution (Windows Form Application) .Το ονομάζουμε **“Ask7”**.

Name:	Ask7
Location:	c:\users\michaela\documents\visual studio 2010\Projects
Solution name:	Ask7

Εικόνα 3

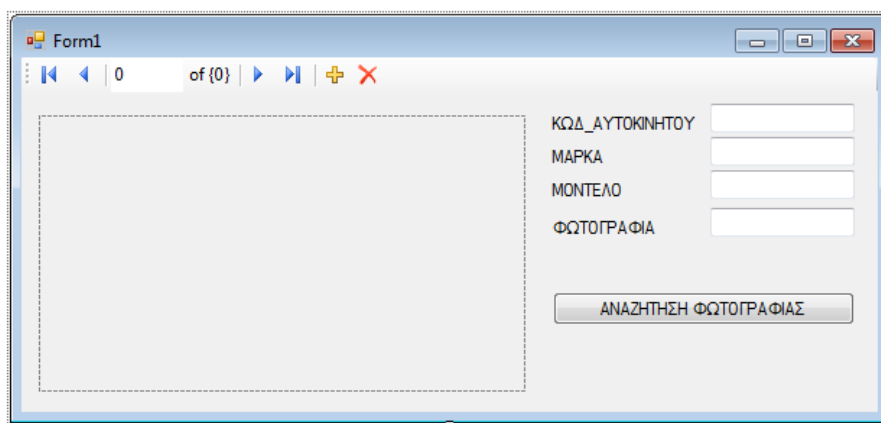
Προσθέσουμε τα components που χρειαζόμαστε για την άσκηση. Όλα τα Components βρίσκονται στο εσωτερικό του **Toolbox**.



Εικόνα 4

Τα Components που θα χρειαστούν είναι :

- 1x PictureBox
- 4x TextBox
- 4x Label
- 1x Button
- 1x OpenFileDialog
- 1x BindingNavigator
- 1x BindingSource
- 1x DataSet

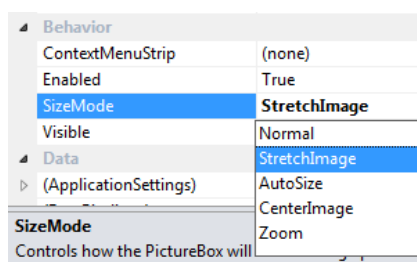


Εικόνα 5

Ορίζουμε από το παράθυρο των Properties για κάθε ένα από τα παρακάτω components την αντίστοιχη ιδιότητα **text** :

- Label1 → ΚΩΔ\_ΑΥΤΟΚΙΝΗΤΟΥ
- Label2 → ΜΑΡΚΑ
- Label3 → ΜΟΝΤΕΛΟ
- Label4 → ΦΩΤΟΓΡΑΦΙΑ
- Button1 → ΑΝΑΖΗΤΗΣΗ ΦΩΤΟΓΡΑΦΙΑΣ

Από το **pictureBox1** αλλάζουμε την ιδιότητα **SizeMode** → **StretchImage**.



Εικόνα 6

Πατάμε **δεξί κλικ** πάνω στην φόρμα και επιλέγουμε **View Code**.

Αρχικά εισάγουμε στις προκαθορισμένες βιβλιοθήκες, άλλες δύο που θα μας χρειαστούν για το project μας :

```
using System.Data.SqlClient;  
using System.IO;
```

Ακριβώς πριν τον constructor της **Form1** ορίζουμε τα **public** αντικείμενα με τα οποία θα δουλέψουμε:

```
SqlConnection connection;  
SqlDataAdapter sqlAdapter;  
String database, host, query;  
SqlCommand command;
```

**Ο Constructor της Form1 θα πρέπει να είναι όπως και παρακάτω:**

```
public Form1()  
{  
    InitializeComponent(); // Αρχικοποίηση των components  
  
    bindingNavigator1.BindingSource = bindingSource1; //Ορισμός πηγής  
    δεδομένων για το navigator μας  
  
    database = "ΕΝΟΙΚΙΑΣΕΙΣ_ΑΥΤΟΚΙΝΗΤΩΝ"; // Το όνομα της βάσης μας  
    host = "(local)"; // Το όνομα του υπολογιστή που φιλοξενεί την βάση  
  
}
```

### **Μέσα στην μέθοδο Form1 Load πληκτρολογούμε τον παρακάτω κώδικα:**

```
private void Form1_Load(object sender, EventArgs e)
{
    connection = new SqlConnection("Data Source=" + host + ";Initial
Catalog=" + database + ";Integrated Security=True"); //Σύνδεση με την βάση
    connection.Open(); //Ανοιγμα Βάσης

    if (connection.State == ConnectionState.Open) //Ελεγχος για σωστό
άνοιγμα
    {
        MessageBox.Show("Connection Established!");
    }
    else
    {
        MessageBox.Show("Connection Error!");
        Application.Exit();
    }

    query = "Select * from ΑΥΤΟΚΙΝΗΤΑ"; //Το ερώτημα που θα εκτελεστεί
για να πάρουμε δεδομένα

    sqlAdapter = new SqlDataAdapter(query, connection); //Εκτέλεση του
παραπάνω ερωτήματος

    sqlAdapter.Fill(dataSet1); //Γέμισμα του dataset που θα περιέχει
τοπικά τα δεδομένα μας για να τα επεξεργαστούμε και να τα εμφανίσουμε

    bindingSource1.DataSource = dataSet1.Tables[0]; //Ορισμός του
πρώτου πίνακα του dataSet1 ως πηγή δεδομένων του bindingSource ,
χρησιμοποιούμε το tables[0] γιατί η βάση μας έχει ως πρώτο πίνακα τον πίνακα
ΑΥΤΟΚΙΝΗΤΑ.

    bindingNavigator1.Refresh(); //Με τον παρακάτω τρόπο ορίζουμε
ποια πεδία της βάσης μας θα γεμίζουν το κάθε textbox

    textBox1.DataBindings.Add(new System.Windows.Forms.Binding("Text",
bindingSource1, "KA", true));
    textBox2.DataBindings.Add(new System.Windows.Forms.Binding("Text",
bindingSource1, "MARKA", true));
    textBox3.DataBindings.Add(new System.Windows.Forms.Binding("Text",
bindingSource1, "MONTELO", true));
    textBox4.DataBindings.Add(new System.Windows.Forms.Binding("Text",
bindingSource1, "FOTO", true));

    refreshImage(); //Αφού τα πεδία μας έχουν δεδομένα καλούμε την
παρακάτω custom συνάρτηση για να εμφανίσουμε την εικόνα μας στο pictureBox1
refreshImage();
}
```

**Στο τέλος της κλάσης μας δημιουργούμε μια custom συνάρτηση που συναντήσαμε παραπάνω , την refreshImage() με τον παρακάτω κώδικα:**

```
public void refreshImage()
{
    String photoPath = textBox4.Text.Trim();
    if (photoPath != null && File.Exists(photoPath))
    {
        pictureBox1.Image = Image.FromFile(photoPath);
    }
    else
    {
        pictureBox1.Image = Image.FromFile("C:/error.png");
    }
}
```

Ο κώδικας είναι προφανής, παίρνει την τιμή του **textBox1**, ελέγχει αν η διαδρομή αυτή περιέχει κάποια φωτογραφία που είναι έγκυρη και την εμφανίζει , αλλιώς εμφανίζει μια προκαθορισμένη φωτογραφία λάθους.

Για να εναλλάσσονται οι φωτογραφίες ανάλογα με την εγγραφή της βάσης στην οποία βρισκόμαστε πληκτρολογούμε τον αντίστοιχο κώδικα στο **event RefreshItems** του **navigator** μας :


```
private void bindingNavigator1_RefreshItems(object sender, EventArgs e)
{
    refreshImage();
}
```

Τέλος , στο **Click event** του **Button** γράφουμε κώδικα ώστε να ανοίγει ένα αρχείο από τον δίσκο, να ελέγχεται αν αυτό είναι έγκυρο, να φορτώνεται η εικόνα στο **PictureBox** μας και να αποθηκεύεται η αλλαγή αυτή στην βάση με την εκτέλεση ενός ερωτήματος.

```
private void button1_Click(object sender, EventArgs e)
{
    String openPath;
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        openPath = openFileDialog1.InitialDirectory +
openFileDialog1.FileName;
        textBox4.Text = openPath;
        pictureBox1.Image = Image.FromFile(openPath);
        command = new SqlCommand("update AYTOKINHHTA set" + "\r\n" +
"ΦΩΤΟΓΡΑΦΙΕΣ='"+openPath+"' where" + "\r\n" +
"ΚΩΔ_ΑΥΤΟΚΙΝΗΤΟΥ='"+textBox1.Text+"'", connection);
        command.ExecuteNonQuery();
    }
}
```

**Προληπτικά κλείνουμε την σύνδεση με την Βάση χειροκίνητα γράφοντας τον παρακάτω κώδικα στο Close Event της Form1.**

```
Private void Form1_FormClosed (object sender, FormClosedEventArgs e)
{
    connection.Close();
}
```

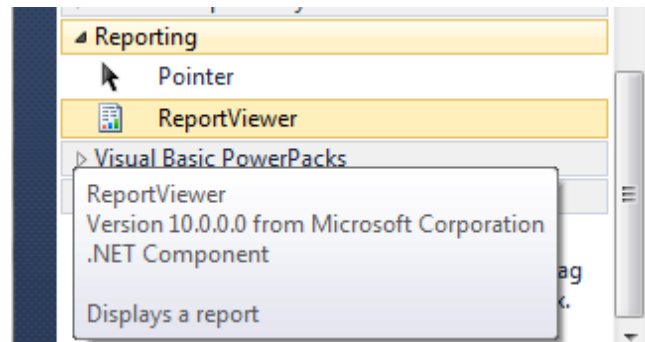
Εκτελούμε το πρόγραμμα πατώντας το run .

## Άσκηση 8 – Δημιουργία αναφορών

**Σκοπός** της παρούσας εργαστηριακής άσκησης, είναι η εξοικείωση του σπουδαστή με τη δημιουργία αναφορών. Οι αναφορές βοηθούν στην απεικόνιση των αποτελεσμάτων ερωτημάτων με τη δυνατότητα εκτύπωσή τους.

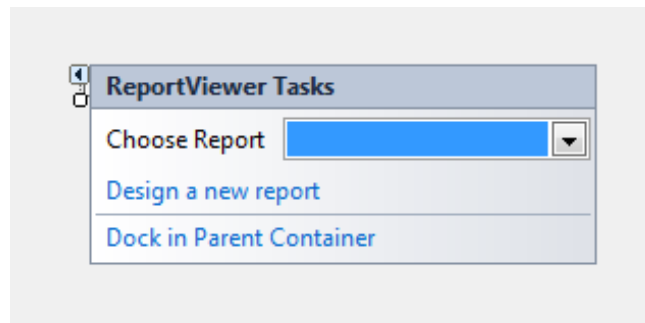
Ανοίγουμε το **Visual Studio 2010** και δημιουργούμε ένα νέο **Windows Form Application** σε C#.

Από το **toolbox** προσθέτουμε στην φόρμα μας ένα **Report Viewer**.



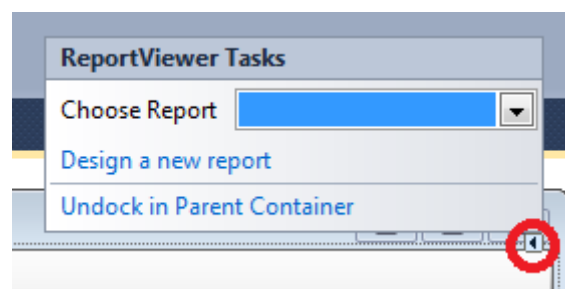
Εικόνα 1

Από το **Report Viewer Tasks** επιλέγουμε **Dock in Parent Container** για να καλύψει όλη την φόρμα μας.



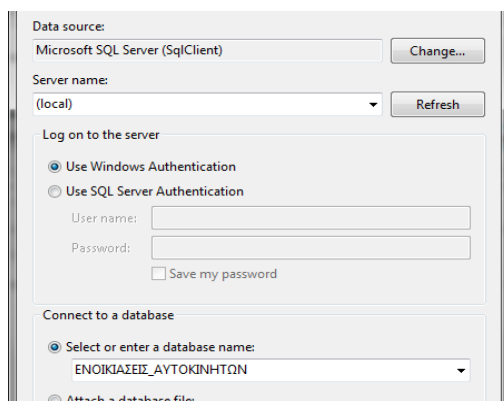
Εικόνα 2

Αμέσως μετά κάνουμε κλικ στο **Design a new report** για να σχεδιάσουμε το πώς θα εμφανίζεται στην αναφορά μας.



Εικόνα 3

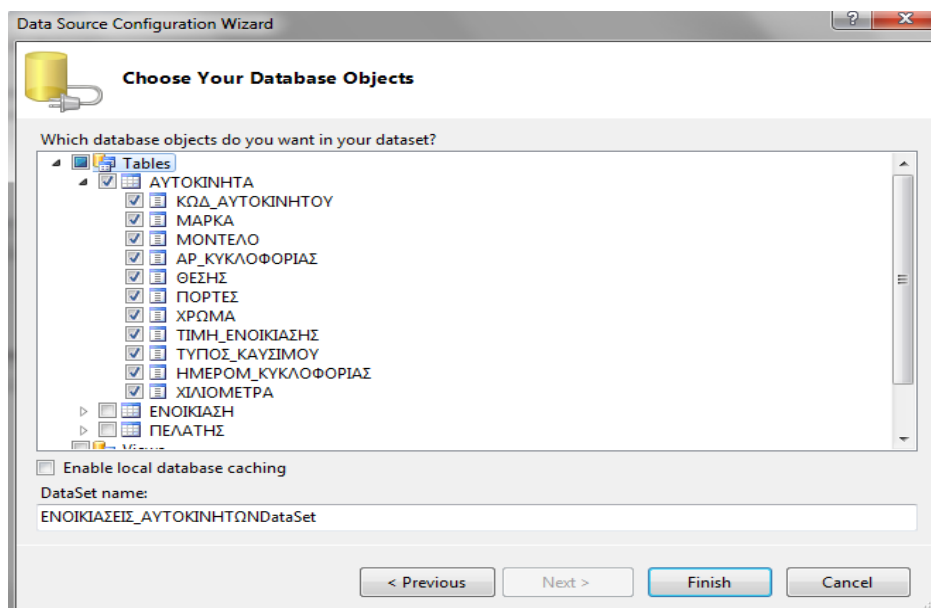
Επιλέγουμε κατά σειρά **Database** → **Next** → **Dataset** → **Next** και συμπληρώνουμε τις ιδιότητες για σύνδεση με την βάση όπως και παρακάτω :



Εικόνα 4

Πατάμε **OK** και προχωράμε στα επόμενα δύο παράθυρα πατώντας **Next** .

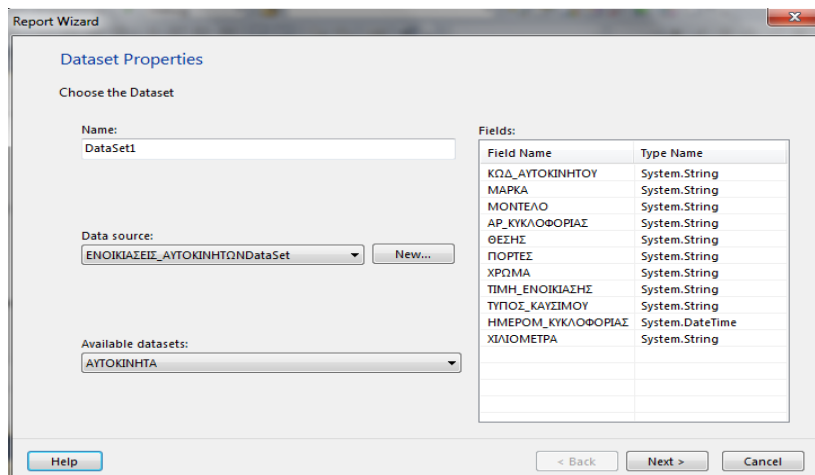
Από το παρακάτω παράθυρο επιλέγουμε τους πίνακες και τα πεδία τα οποία θα προβάλλουμε στην αναφορά μας όπως φαίνεται παρακάτω.



Εικόνα 5

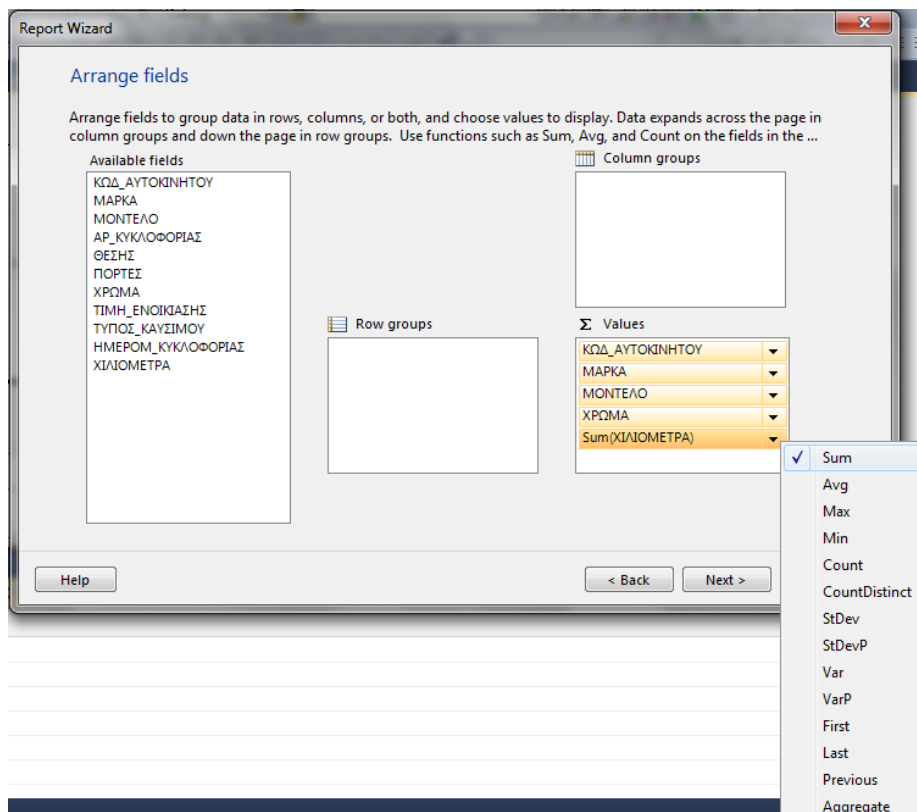


Επιλέγουμε **Next** στην σύνοψη των επιλογών μας όπως φαίνεται παρακάτω.



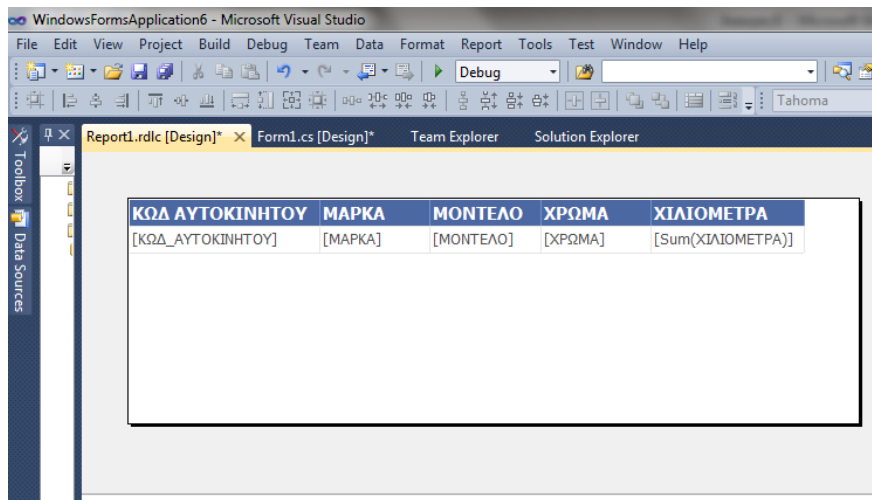
Εικόνα 6

Μετακινούμε (**Drag & Drop**) τα πεδία που θέλουμε να βάλουμε στην αναφορά μας, έχοντας την δυνατότητα να τα ταξινομήσουμε κατά στήλη ή κατά σειρά . Στο παράδειγμά μας θα χρησιμοποιήσουμε τα παρακάτω πεδία



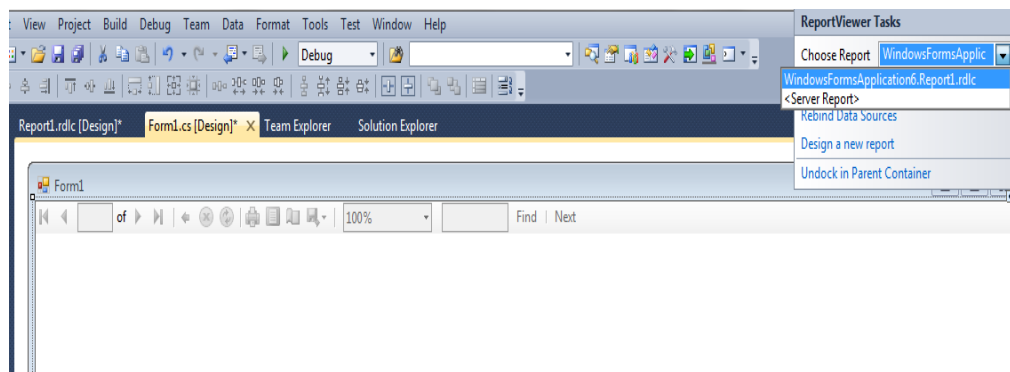
Εικόνα 7

Στα επόμενα δύο βήματα κάνουμε ρυθμίσεις μορφοποίησης αν θέλουμε και πατάμε Finish.




Εικόνα 8

Επιστρέφουμε σε **Design Mode** και από το **Report Viewer Tasks** επιλέγουμε το ask8.Report1.rdlc



Εικόνα 9

Έπειτα πατάμε το πλήκτρο (run) . Το τελικό μας αποτέλεσμα μοιάζει με την παρακάτω εικόνα.

ΚΩΔ ΑΥΤΟΚΙΝΗΤΟΥ	ΜΑΡΚΑ	ΜΟΝΤΕΛΟ	ΧΡΩΜΑ	ΧΙΛΙΟΜΕΤΡΑ
122	OPEL	ASTRA	ΑΣΗΜΙ	150000
125	OPEL	CORSA	ΜΑΥΡΟ	1230000
135	BMW	316	ΚΟΚΚΙΝΟ	1398000

Εικόνα 10

Από τα προκαθορισμένα κουμπιά του Report Viewer βλέπουμε ότι μπορούμε να κάνουμε εκτύπωση, εξαγωγή σε Word, Excel και PDF αρχεία , όπως και αρκετές ακόμα λειτουργίες.

## Άσκηση 9 – Δημιουργία αποθηκευμένων διαδικασιών και σκανδάλης

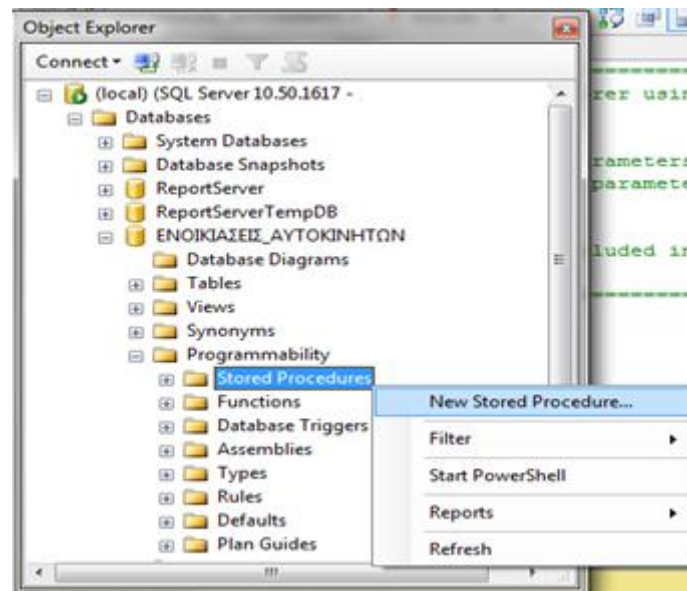
**Σκοπός** της παρούσας εργαστηριακής άσκησης, είναι η εξοικείωση του σπουδαστή με τη δημιουργία αποθηκευμένων διαδικασιών (Stored Procedures) και σκανδάλης (trigger) στον MICROSOFT SQL SERVER 2008 R2.

### Αποθηκευμένες Διαδικασίες (Stored Procedures):

Μια αποθηκευμένη διαδικασία είναι ένα πρόγραμμα που αποτελείται από SQL εντολές, που αποθηκεύεται και εκτελείται στον database server χωρίς ενδιάμεση επικοινωνία με το πρόγραμμα εφαρμογής.

#### Παράδειγμα 1 :

Μεταβαίνουμε στον **MICROSOFT SQL SERVER**, επιλέγουμε τη βάση μας → **Programmability** → **Stored Procedures** → Δεξί κλικ → **New Stored Procedure**.



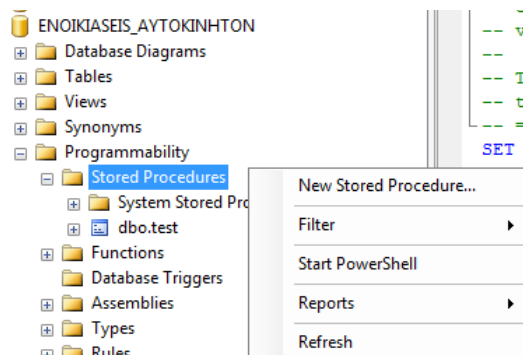
Εικόνα 1

Στο παράθυρο που ανοίγει γράφουμε τον παρακάτω κώδικα της διαδικασίας

```
-- =====  
CREATE PROCEDURE test AS  
BEGIN  
select 'Hello world'  
END  
-- =====
```

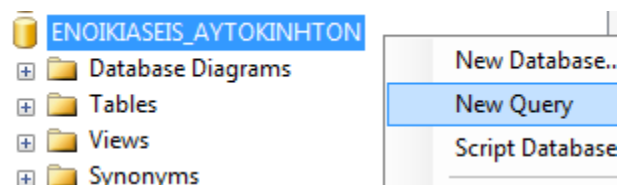
Για να αποθηκευτεί η διαδικασία αυτή πιέζετε το κουμπί **! Execute** από την γραμμή

εντολών και στη συνέχεια κάντε **refresh** στην επιλογή **Stored Procedures**.



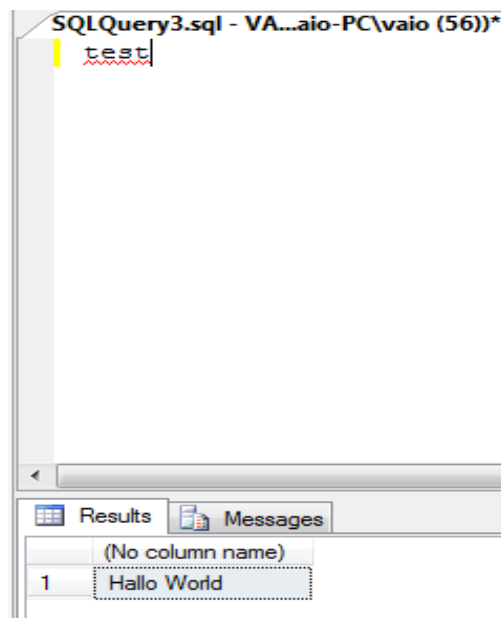
Εικόνα 2

Για να εκτελεστεί η Αποθηκευμένη Διαδικασία που μόλις δημιουργήθηκε πιέζουμε Δεξί κλικ πάνω στη Βάση Δεδομένων → **New Query**



Εικόνα 3

Γράφουμε το όνομα της διαδικασίας, την εκτελούμε πατώντας το **!Execute** από την γραμμή εντολών και βλέπουμε τα αποτελέσματα. Το συγκεκριμένο παράδειγμα τυπώνει "hello world".



Εικόνα 4

## Λόγος χρήσης των αποθηκευμένων διαδικασιών (Stored Procedures) :

Κάθε διαδικασία εκτελείται στον Database Server, επομένως είναι ανεξάρτητη από την πλατφόρμα εφαρμογής. Επιπλέον λαμβάνουμε ταχύτητα τα αποτελέσματα για τον λόγο ότι η εκτέλεση γίνεται στον Database Server και δεν χρειάζεται η επικοινωνία μέσω δικτύου με την πλατφόρμα εφαρμογής. Στο παράδειγμα 1, εκτελέσαμε Procedure μιας εντολής η οποία θα μπορούσε να γραφεί και έξω από το block εντολών. Για να εκτελέσουμε πολλές εντολές σε μια διαδικασία πρέπει υποχρεωτικά να βρίσκονται μέσα σε ένα μπλοκ begin - end. Με τον ίδιο τρόπο που περιγράψαμε πριν δημιουργήστε τις παρακάτω διαδικασίες.

### Παράδειγμα 2 :

```
-- =====  
CREATE PROCEDURE test2 AS  
BEGIN  
select 'ASKISI 9 VASEIS DEDOMENON II'  
select 'Δημιουργία Αποθηκευμένων Διαδικασιών'  
END  
-- =====
```

### Χρήση μεταβλητών σε διαδικασία :

Οι μεταβλητές δηλώνονται με το σύμβολο “@” πριν τον ορισμό τους.

### Παράδειγμα 3 :

```
-- =====  
CREATE PROCEDURE test3(@var1 INT, @var2 INT)AS  
BEGIN  
select @var1+@var2;  
END  
GO  
-- =====
```

Την καλούμε με 2 ορίσματα : **test3** 4,6

Αποτέλεσμα : 10

## Παράδειγμα 4 :

Παράδειγμα διαδικασίας που επηρεάζει τις τιμές του πίνακα :

```
-- =====  
CREATE PROCEDURE AFXISI_TIMIS (@POSOSTO INT) AS  
BEGIN  
UPDATE ΑΥΤΟΚΙΝΗΤΑ  
SET ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ=ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ*(1+@POSOSTO/100)  
select @var1+@var2;  
END  
GO  
-- =====
```

Έστω ότι θέλουμε να γίνει αύξηση τιμών κατά 10%. Παρατηρούμε τις τιμές του πίνακα πριν την αύξηση. Εφόσον γράψουμε τον παραπάνω κώδικα και πατήσουμε **!Execute** και μετά **Refresh** μεταβαίνουμε σε νέο **Query** για να καλέσουμε και να εκτελέσουμε τη διαδικασία γράφοντας «AFXISI\_TIMIS 10». Πατώντας **!Execute**, εκτελείται η διαδικασία. Παρατηρούμε τις νέες τιμές ενοικίασης στον πίνακα οι οποίες θα είναι αυξημένες κατά 10%.

## Σκανδάλες (Triggers):

Trigger (Σκανδάλη), ονομάζεται ένα αντικείμενο Βάσης Δεδομένων, το οποίο απευθύνεται σε συγκεκριμένο πίνακα και εκτελείται όταν συμβεί ένα συγκεκριμένο γεγονός σε αυτόν τον πίνακα.

Η δήλωση μιας σκανδάλης ακολουθεί το εξής πρότυπο:

```
-- =====  
CREATE TRIGGER trigger_name ON tbl_name  
AFTER INSERT, DELETE, UPDATE  
BEGIN  
. . .  
END  
-- =====
```

όπου:

- **trigger\_name** : Το όνομα μιας νέας σκανδάλης
- **tbl\_name** : Το όνομα του πίνακα που αφορά η σκανδάλη
- **AFTER INSERT, DELETE, UPDATE**: Το πότε θα εκτελεστούν οι εντολές της σκανδάλης.

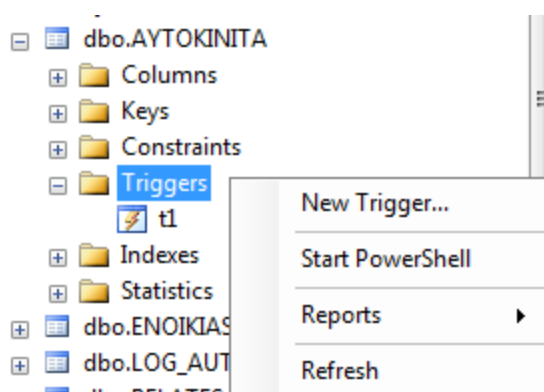
Για εκτέλεση πολλαπλών ενεργειών, όπως και στις Stored Procedures, χρησιμοποιείται η δομή **BEGIN {εντολές} END**.

## Παράδειγμα 5 :

Έστω ότι θέλουμε να υπολογίζεται αυτόματα το πεδίο του πίνακα ΕΝΟΙΚΙΑΣΗΣ, ΤΕΛΙΚΟ\_ΠΟΣΟ να ισούται με το άθροισμα του ΤΕΛΙΚΟ\_ΠΟΣΟ συν ΧΡΕΩΣΗ\_ΖΗΜΙΑΣ σε κάθε ενέργεια INSERT, UPDATE, DELETE στον πίνακα.

### Δημιουργία σκανδάλης :

Αφορά τον πίνακα ΕΝΟΙΚΙΑΣΗ, οπότε ή δημιουργούμε ένα νέο **Query** ή επιλέγουμε **ΑΥΤΟΚΙΝΗΤΑ** → **Triggers** **Δεξί κλικ** → **New Trigger** .



Εικόνα 5

Στο παράθυρο που εμφανίζεται γράφουμε τον παρακάτω κώδικα:

```
-- =====  
CREATE TRIGGER ΤΕΛ_ΠΟΣΟ ON ΕΝΟΙΚΙΑΣΗ  
AFTER INSERT, UPDATE, DELETE  
AS  
BEGIN  
UPDATE ΕΝΟΙΚΙΑΣΗ SET ΤΕΛΙΚΟ_ΠΟΣΟ=ΠΟΣΟ_ΠΛΗΡΩΜΗΣ+ΧΡΕΩΣΗ_ΖΗΜΙΑΣ  
END  
-- =====
```

Εφόσον γράψαμε τον κώδικα του Trigger, ελέγχουμε αν είναι σωστή η σύνταξη πατώντας **!Execute** και έχει δημιουργηθεί επιτυχώς η σκανδάλη. Για να δούμε τα αποτελέσματα θα δημιουργήσουμε μια νέα εγγραφή στον πίνακα ΕΝΟΙΚΙΑΣΗ. Αφού εισάγουμε τα απαραίτητα πεδία, φτάνοντας στο πεδίο “ΤΕΛΙΚΟ\_ΠΟΣΟ” πατώντας το πλήκτρο “**TAB**”, παρατηρούμε ότι το πεδίο υπολογίζεται αυτόματα.

## Παράδειγμα 6 :

Εισάγουμε ένα νέο πεδίο στον πίνακα ΠΕΛΑΤΕΣ, με όνομα ΗΛΙΚΙΑ τύπου INT, στο οποίο θέλουμε να υπολογίζεται αυτόματα η ηλικία του κάθε πελάτη, από την ημερομηνία γέννησης που υπάρχει ήδη στον πίνακα. Για να δημιουργηθεί αυτή η σκανδάλη πρέπει να γνωρίζουμε δύο συναρτήσεις :

- **DATEDIFF ( YEAR, ΗΜΕΡΟΜΗΝΙΑ\_ΑΡΧΗΣ , ΗΜΕΡΟΜΗΝΙΑ\_ΤΕΛΟΥΣ )**

η οποία υπολογίζει τη διαφορά δυο ημερομηνιών (ΗΜΕΡΟΜΗΝΙΑ\_ΑΡΧΗΣ , ΗΜΕΡΟΜΗΝΙΑ\_ΤΕΛΟΥΣ) σε χρόνια, εβδομάδες ή μέρες (YEAR, WEEK, DAY).

- **GETDATE ()** η οποία επιστρέφει την τρέχουσα ημερομηνία.

Οπότε ο κώδικας της σκανδάλης για τον πίνακα ΠΕΛΑΤΕΣ, διαμορφώνεται όπως παρακάτω :

```
-- =====  
CREATE TRIGGER ΗΛΙΚΙΑ ON [dbo].[ΠΕΛΑΤΕΣ]  
FOR INSERT, UPDATE, DELETE  
AS  
BEGIN  
UPDATE ΠΕΛΑΤΕΣ SET ΗΛΙΚΙΑ=DATEDIFF (YEAR, DAY, GETDATE())  
END  
-- =====
```

Καταχωρώντας μια νέα εγγραφή στον πίνακα πελάτες παρατηρούμε ότι υπολογίζεται η ηλικία του πελάτη αυτόματα.

## Παράδειγμα 7 :

Δημιουργούμε έναν πίνακα με όνομα LOG\_AUTO στον οποίο με την ενεργοποίηση μιας σκανδάλης (TRIGGER) που θα εφαρμόζεται πάνω στον βασικό πίνακα ΑΥΤΟΚΙΝΗΤΑ, θα αποθηκεύονται οι τροποποιήσεις που θα γίνονται στο πεδίο ΤΙΜΗ\_ΕΝΟΙΚΙΑΣΗΣ του πίνακα ΑΥΤΟΚΙΝΗΤΑ.

Ο πίνακας LOG\_AUTO περιλαμβάνει :

**ΚΩΔ\_ΑΥΤΟΚΙΝΗΤΟΥ** τον αριθμό πινακίδας του αυτοκινήτου,

**USER\_ID** τον κωδικό του χρήστη που κάνει αλλαγές στον πίνακα,

**ΗΜΕΡΟΜΗΝΙΑ** που έχει η αλλαγή στον πίνακα,

**ΠΑΛΙΑ** η παλιά τιμή ενοικίασης,

**ΝΕΑ** η νέα τιμή ενοικίασης



Ο πίνακας μπορεί να δημιουργηθεί είτε χειροκίνητα, όπως δημιουργήσαμε πίνακες στις πρώτες εργαστηριακές ασκήσεις, είτε εκτελώντας το παρακάτω Query :

```
-- =====  
CREATE TABLE LOG_ΑΥΤΟ (ΚΩΔ_ΑΥΤΟΚΙΝΗΤΟΥ varchar(10), USER_ID varchar(20), ΗΜΕΡΟΜΗΝΙΑ  
datetime, ΠΑΛΙΑ float, ΝΕΑ float)  
-- =====
```

Δημιουργούμε λοιπόν ένα **Trigger** για τον πίνακα **ΑΥΤΟΚΙΝΗΤΑ** με τον παρακάτω κώδικα :

```
-- =====  
CREATE trigger t1 ON ΑΥΤΟΚΙΝΗΤΟΥ  
AFTER UPDATE AS  
BEGIN  
DECLARE @TIME_ΠΑΛΙΑ REAL , @ΤΙΜΗ_ΝΕΑ REAL , @ΚΩΔ_ΑΥΤΟΚΙΝΗΤΟΥ INT  
SELECT @ΤΙΜΗ_ΝΕΑ= (SELECT ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ FROM INSERTED)  
SELECT @ΤΙΜΗ_ΠΑΛΙΑ= (SELECT ΤΙΜΗ_ΕΝΟΙΚΙΑΣΗΣ FROM DELETED)  
SELECT @ΚΩΔ_ΑΥΤΟΚΙΝΗΤΟΥ= (SELECT ΚΩΔ_ΑΥΤΟΚΙΝΗΤΟΥ FROM DELETED)  
INSERT INTO LOG_ΑΥΤΟ  
VALUES (@ΚΩΔ_ΑΥΤΟΚΙΝΗΤΟΥ, USER_ID(), GETDATE(), @ΤΙΜΗ_ΠΑΛΙΑ, @ΤΙΜΗ_ΝΕΑ)  
END  
-- =====
```

Αλλάζουμε τη τιμή ενοικίασης του FIAT STILO από 220 σε 300 και του MERCEDES SLK από 77 σε 100 αφού πρώτα πατήσουμε κλικ πάνω στον πίνακα ΑΥΤΟΚΙΝΗΤΑ Edit Top 200 Rows. Ανοίγουμε τον πίνακα LOG\_AUTO και παρατηρούμε τις καταχωρήσεις που έκανε αυτόματα.

Το παράδειγμα αυτό δείχνει το πώς μπορούν να χρησιμοποιηθούν οι σκανδάλες για να υλοποιήσουν ένα ημερολόγιο παρακολούθησης της κίνησης των συναλλαγών των χρηστών πάνω σε έναν πίνακα.

Βασική ιδέα των **TRIGGER** είναι η διαχείριση των προσωρινών πινάκων **INSERTED** και **DELETED** όπου καταχωρούνται προσωρινά οι κάθε αλλαγές που επιχειρεί να κάνει κάποιος χρήστης.

Σε κάθε τροποποίηση της στήλης ΤΙΜΗ\_ΕΝΟΙΚΙΑΣΗΣ χρησιμοποιώντας την πρόταση UPDATE ενεργοποιείται η σκανδάλη. Έτσι, οι τιμές των γραμμών των εικονικών πινάκων deleted και inserted εκχωρούνται στις αντίστοιχες μεταβλητές @ΤΙΜΗ\_ΠΑΛΙΑ, @ΤΙΜΗ\_ΝΕΑ και @ΚΩΔ\_ΑΥΤΟΚΙΝΗΤΟΥ. Οι εκχωρημένες τιμές, μαζί με το όνομα χρήστη και την τρέχουσα ημερομηνία και ώρα, εισάγονται κατόπιν στον πίνακα LOG\_AUTO.

## Ενότητα 6 –Ερωτήματα πολλαπλής επιλογής

### Χρήση του προγράμματος Hot Potatoes

Για την δημιουργία της ενότητας με της ερωτήσεις πολλαπλών ερωτήσεων χρησιμοποιήθηκε το λογισμικό ανοιχτού κώδικα Hot Potatoes.

Το Hot Potatoes είναι ένα πρόγραμμα ανοιχτού λογισμικού με το οποίο μπορείτε να δημιουργήσετε ασκήσεις διαφόρων μορφών για χρήση είτε μέσω internet είτε μέσω υπολογιστή. Στην πραγματικότητα είναι java scripts που δημιουργούνται με έναν ιδιαίτερα φιλικό τρόπο.

Το πρόγραμμα διατίθεται ελεύθερα για μη εμπορικούς εκπαιδευτικούς σκοπούς και με την προϋπόθεση οι ασκήσεις που δημιουργούνται με αυτό να είναι ελεύθερα διαθέσιμες στο διαδίκτυο (περισσότερα γι' αυτό το ζήτημα στην παρακάτω διεύθυνση).

Η διαδικασία η οποία ακολουθήθηκε για την δημιουργία των ερωτήσεων είναι η εξής :

1. Λήψη της εφαρμογής από το hotpot.uvic.ca

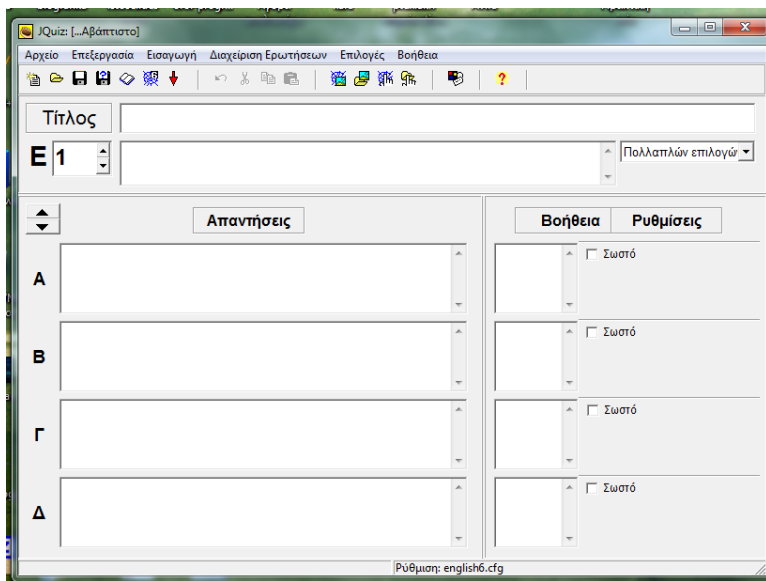


[Home](#) [News](#) [Support](#) [Downloads](#) [HotPot Sites](#) [Tutorials](#) [Bugs/Updates](#) [FAQ](#) [Extras](#)

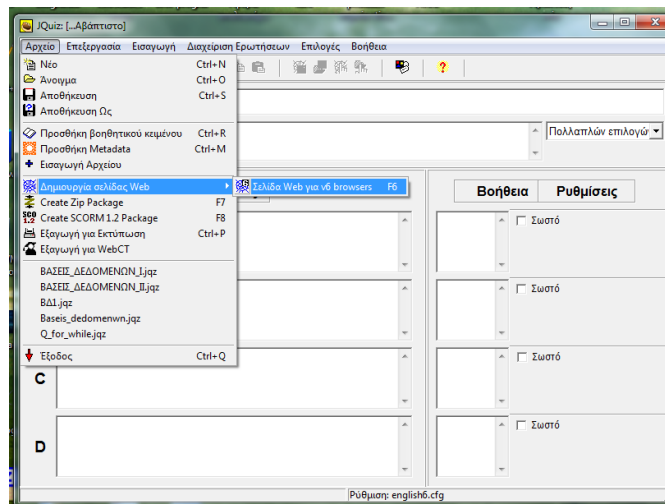
2. Εγκατάσταση το εκτελέσιμο πρόγραμμα
3. Εκτέλεση του προγράμματος



#### 4. Συμπληρώνουμε τα πεδία των ερωτήσεων - απαντήσεων



#### 5. Εξαγωγή των ερωτήσεων σε μορφή .htm



#### 6. Ενσωμάτωση των ερωτήσεων στην Ιστοσελίδα

### Ερωτήσεις μαθήματος «Βάσεις Δεδομένων»

1. Έστω ο πίνακας R που περιέχει N1 εγγραφές και ο Πίνακας S που περιέχει N2 εγγραφές με  $N1 > N2$ . Ποιό είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος εγγραφών που θα προκύψει από την πράξη  $R \cap S$ ;

- a) N1, N1+N2
- b) N1-N2, N1
- c) N2, N1+N2
- d) 0, N2

2. Ποια είναι τα βήματα σχεδιασμού μίας Βάσης Δεδομένων;

- a) 1. Εννοιολογικός σχεδιασμός, 2. Ιεραρχικός σχεδιασμός, 3. Σχεσιακός Σχεδιασμός
- b) 1. Ιεραρχικός σχεδιασμός, 2. Λογικός σχεδιασμός 3. Φυσικός Σχεδιασμός
- c) 1. Εννοιολογικός σχεδιασμός, 2. Λογικός Σχεδιασμός, 3. Φυσικός Σχεδιασμός
- d) 1. Λογικός Σχεδιασμός, 2. Εννοιολογικός σχεδιασμός, 3. Σχεσιακός Σχεδιασμός

3. Βαθμό μίας συσχέτισης είναι:

- a) Το πλήθος των γνωρισμάτων της συσχέτισης
- b) Το πλήθος των κλειδιών της συσχέτισης
- c) Το πλήθος των οντοτήτων που συμμετέχουν στη συσχέτιση
- d) Το πλήθος των εγγραφών

4. Μία σχέση είναι 2KM όταν:

- a) Τα πεδία που δεν είναι κλειδιά συμμετέχουν σε ένα υποψήφιο κλειδί
- b) Τα πεδία που δεν είναι κλειδιά εξαρτώνται συναρτησιακά από ένα ξένο κλειδί
- c) Τα πεδία που δεν είναι κλειδιά εξαρτώνται συναρτησιακά από όλο το κλειδί και όχι από μέρος του κλειδιού.
- d) Τα πεδία που δεν είναι κλειδιά εξαρτώνται συναρτησιακά από μέρος του κλειδιού και όχι από όλο το κλειδί.

**5. Η συντομογραφία του RDBMS προκύπτει από:**

- a) Relative Data Base Management System
- b) Relational Data Base Manipulative System
- c) Relational Data Base Management System
- d) Relationship Data Base Management System

**6. Το ξένο κλειδί (Foreign Key):**

- a) Είναι πεδίο ή συνδυασμός περισσότερων πεδίων μίας εγγραφής
- b) έχει το λιγότερο αριθμό πεδίων που χαρακτηρίζουν μοναδικά μία εγγραφή.
- c) Είναι ένα πεδίο ή συνδυασμός πεδίων που χαρακτηρίζει μοναδικά μία εγγραφή και το επιλέγουμε από τα διαθέσιμα υποψήφια κλειδιά.
- d) είναι ένα πεδίο που έχει το ίδιο σύνολο τιμών με το πρωτεύον κλειδί ενός άλλου πίνακα.

**7. Έστω ο πίνακας R που περιέχει N1 εγγραφές και ο Πίνακας S που περιέχει N2 εγγραφές με  $N1 < N2$ . Ποιο είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος εγγραφών που θα προκύψει από την πράξη  $R \cup S$**

- a) 0, N2
- b) N1, N1+N2
- c) N1-N2, N1
- d) N2, N1+N2

**8. Ποιό από τα παρακάτω ερωτήματα δεν έχει λάθος και τα αποτελέσματα του θα προβάλλονται σωστά;**

a) Select ΠΟΛΗ, count (ΚΩΔ\_ΠΕΛΑΤΗ)

From ΠΕΛΑΤΕΣ

Group by ΠΟΛΗ

Having count (ΚΩΔ\_ΠΕΛΑΤΗ) >100;

b) Select 'ΚΩΔ\_ΠΕΛΑΤΗ'

From ΠΕΛΑΤΕΣ

Having count [\*] >100

Grouping by 'ΟΝΟΜΑ\_ΠΕΛΑΤΗ';

c) Select ΠΟΛΗ, count (ΚΩΔ\_ΠΕΛΑΤΗ) as TOTAL

From ΠΕΛΑΤΕΣ

Group by ΠΟΛΗ

where TOTAL >100;

d) Select ΚΩΔ\_ΠΕΛΑΤΗ, count(\*)

From ΠΕΛΑΤΕΣ

Where count (\*) >100;

**9. Ο πίνακας αποτελείται από τα παρακάτω πεδία: ΚΑΤΕΧΟΥΝ : ΚΙ , ΚΑ , ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ.**

Ποία από τις παρακάτω εντολές SQL αλλάζει στον πίνακα ΚΑΤΕΧΟΥΝ το ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ σε 50 για τον ιδιοκτήτη με ΚΙ ίσο με 500 που κατέχει το ακίνητο με ΚΑ ίσο με 50;

a) Alter table ΚΑΤΕΧΟΥΝ Set ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ =50

Where ΚΙ=500 AND ΚΑ=50;

b) Select ΚΑΤΕΧΟΥΝ.ΚΙ, ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ =50

From ΚΑΤΕΧΟΥΝ

Where ΚΙ=500 AND ΚΑ=50;

c) Insert into ΚΑΤΕΧΟΥΝ (ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ) Values (50)

Where KI=500 AND ΚΑ=50;

d) Update ΚΑΤΕΧΟΥΝ Set ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ =50

Where KI=500 AND ΚΑ=50;

**10. Ο παρακάτω πίνακας αποτελείται από τα εξής πεδία: ΙΔΙΟΚΤΗΤΕΣ (ΚΙ , ΕΠΙΘΕΤΟ , ΟΝΟΜΑ, ΤΗΛ, ΟΔΟΣ, ΠΟΛΗ). Ποιό ερώτημα προβάλλει ανά το πλήθος των ιδιοκτητών;**

a) Select ΠΟΛΗ, Sum (ΚΙ) AS ΠΛΗΘΟΣ

From ΙΔΙΟΚΤΗΤΕΣ

Group by ΙΔΙΟΚΤΗΤΕΣ.ΚΙ

Having ΠΛΗΘΟΣ>1;

b) Select ΠΟΛΗ, Count (ΕΠΙΘΕΤΟ) AS ΠΛΗΘΟΣ\_ΙΔ

From ΙΔΙΟΚΤΗΤΕΣ

Group by ΠΟΛΗ;

c) Select ΠΟΛΗ, Count (ΚΙ) AS ΠΛΗΘΟΣ\_ΙΔ

From ΙΔΙΟΚΤΗΤΕΣ

Group by ΙΔΙΟΚΤΗΤΕΣ.ΚΙ;

d) Select ΙΔΙΟΚΤΗΤΕΣ.ΚΙ, Count (ΠΟΛΗ) AS ΠΛΗΘΟΣ\_ΙΔ

From ΙΔΙΟΚΤΗΤΕΣ

Group by ΙΔΙΟΚΤΗΤΕΣ.ΚΙ;

**11. Αλλάξτε με SQL στον πίνακα ΚΑΤΕΧΟΥΝ το ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ σε 30% για τον ιδιοκτήτη με ΚΙ ίσο με 567 που κατέχει το ακίνητο με ΚΑ ίσο με 123.**

ΚΑΤΕΧΟΥΝ : ΚΙ , ΚΑ , ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ.

a) Select ΚΑΤΕΧΟΥΝ.ΚΙ, ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ =30%

From ΚΑΤΕΧΟΥΝ

Group by ΚΑΤΕΧΟΥΝ.ΚΙ

Where ΚΑΤΕΧΟΥΝ.ΚΙ= ΙΔΙΟΚΤΗΤΗΣ.ΚΙ AND ΚΙ=567;

b) Select ΚΑΤΕΧΟΥΝ.ΚΙ, ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ =30%

From ΚΑΤΕΧΟΥΝ

Where ΚΑΤΕΧΟΥΝ.ΚΙ=567;

c) Update ΚΑΤΕΧΟΥΝ Set ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ =0.3

Where ΚΑΤΕΧΟΥΝ.ΚΙ=567;

d) Insert into ΚΑΤΕΧΟΥΝ (ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ) Values (0.3)

Where ΚΑΤΕΧΟΥΝ.ΚΙ=567;

**12. Ποιό από τα παρακάτω ερωτήματα δεν έχει λάθος και τα αποτελέσματα του θα προβάλλονται σωστά ;**

a) Select ΟΝΟΜΑ\_ΠΕΛΑΤΗ distinct

From ΔΑΝΕΙΖΟΜΕΝΟΣ

having ΟΝΟΜΑ\_ΠΕΛΑΤΗ not in ([Παπαδόπουλος] , [Πέτρου]);

b) Select distinct , ΟΝΟΜΑ\_ΠΕΛΑΤΗ

From ΔΑΝΕΙΖΟΜΕΝΟΣ

where ΟΝΟΜΑ\_ΠΕΛΑΤΗ not in ([Παπαδόπουλος] , [Πέτρου]);

c) Select distinct ΟΝΟΜΑ\_ΠΕΛΑΤΗ

From ΔΑΝΕΙΖΟΜΕΝΟΣ

where ΟΝΟΜΑ\_ΠΕΛΑΤΗ not in ('Παπαδόπουλος' , 'Πέτρου');



d) Select ΟΝΟΜΑ\_ΠΕΛΑΤΗ

From ΔΑΝΕΙΖΟΜΕΝΟΣ

where ΟΝΟΜΑ\_ΠΕΛΑΤΗ = ['Παπαδόπουλος', 'Πέτρου'];

**13. Μία σχέση είναι 3 ΚΜ:**

a) Όταν περιέχει τετριμμένες συναρτήσεις σύζευξης.

b) Όταν περιέχει τετριμμένες εξαρτήσεις σύζευξης.

c) Όταν δεν περιέχει μεταβατικές εξαρτήσεις.

d) Όταν περιέχει μεταβατικές εξαρτήσεις.

**14. Το υπέρ-κλειδί (Super Key):**

a) είναι ένα πεδίο ή συνδυασμός περισσότερων πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή.

b) είναι ένα πεδίο ή συνδυασμός περισσότερων πεδίων που χαρακτηρίζει μοναδικά μια εγγραφή και το επιλέγουμε από τα διαθέσιμα υποψήφια κλειδιά ως πρωτεύον κλειδί.

c) έχει το λιγότερο αριθμό πεδίων που χαρακτηρίζουν μοναδικά μία εγγραφή

d) είναι με το πρωτεύον κλειδί ενός άλλου πίνακα.

**15. Ποιό από τα παρακάτω δεν είναι Σύστημα Διαχείρισης Σχεσιακών Βάσεων Δεδομένων (RDBMS):**

a) DB/2

b) Oracle

c) SQL Server 2000

d) MS Manager

**16. Με τη λέξη Ακεραιότητα (integrity) των βάσεων δεδομένων εννοούμε :**

- a) την προστασία των δεδομένων από τη γνωστοποίηση, την αλλοίωση, ή την καταστροφή από μη εξουσιοδοτημένα άτομα.
- b) την αποφυγή πλεονασμού.
- c) την ασφάλεια μέσω του λειτουργικού συστήματος.
- d) την εξασφάλιση ότι αυτά που επιχειρούν να κάνουν οι χρήστες είναι σωστά.

**17. Μία σχέση είναι σε 1 KM όταν:**

- a) Τα πεδία που δεν είναι κλειδιά εξαρτώνται συναρτησιακά από μέρος του κλειδιού και όχι από όλο το κλειδί.
- b) Τα πεδία που δεν είναι κλειδιά εξαρτώνται συναρτησιακά από όλο το κλειδί και όχι από μέρος του κλειδιού.
- c) Τα ονόματα των πεδίων είναι μοναδικά και υπάρχει πρωτεύον κλειδί.
- d) Δεν υπάρχει πλεονασμός στις εγγραφές και υπάρχει πρωτεύον κλειδί.

**18. Ποιές είναι οι τρεις βασικές συνιστώσες ενός μοντέλου δεδομένων;**

- a) Εγγραφές (records), Πεδία (fields), Περιορισμοί (constraints).
- b) Δομές (Structures), Εγγραφές (records), Πεδία (fields).
- c) Δομές (Structures), Πράξεις (operations), Περιορισμοί (constraints).
- d) Εγγραφές (records), Πράξεις (operations), Πεδία (fields).

**19. Ποιο από τα παρακάτω ερωτήματα δεν έχει λάθος και τα αποτελέσματα του θα προβάλλονται σωστά;**

a) Select ΠΟΛΗ, count(ΚΩΔ\_ΠΕΛΑΤΗ)

From ΠΕΛΑΤΕΣ

Group by ΠΟΛΗ

Having count(ΚΩΔ\_ΠΕΛΑΤΗ) >1000;

b) Select 'ΚΩΔ\_ΠΕΛΑΤΗ'

From ΠΕΛΑΤΕΣ

Having count[\*] >100

Grouping by 'ΟΝΟΜΑ\_ΠΕΛΑΤΗ';

c) Select ΠΟΛΗ, count(ΚΩΔ\_ΠΕΛΑΤΗ)as TOTAL

From ΠΕΛΑΤΕΣ

Group by ΠΟΛΗ

where TOTAL >1000 ;

d) Select ΚΩΔ\_ΠΕΛΑΤΗ, count(\*)

From ΠΕΛΑΤΕΣ

Where count(\*) >1000;

**20. Οι παρακάτω πίνακες αποτελούνται από τα εξής πεδία:**

**ΑΝΑΓΝΩΣΤΕΣ: ΚΩΔΑΝΑΓ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ**

**ΔΑΝΕΙΣΜΟΙ: ΚΩΔΑΝΑΓ, ΚΩΔΒΙΒΛΙΟΥ, ΗΜΕΡΟΜΗΝΙΑ**

**ΒΙΒΛΙΑ: ΚΩΔΒΙΒΛΙΟΥ, ΤΙΤΛΟΣ, ΣΥΓΓΡΦΕΑΣ, ΚΑΤΗΓΟΡΙΑ, ΕΤΟΣ\_ΕΚΔΟΣΗΣ**

**Ποιο ερώτημα SQL προβάλλει τον ΚΩΔΑΝΑΓΝ και το πλήθος δανεισμών για τους αναγνώστες που έχουν κάνει συνολικά πάνω από 100 δανεισμούς;**

a) Select ΚΩΔΑΝΑΓ, count(ΚΩΔΔΑΝ)as PLHTHOS

From ΔΑΝΕΙΣΜΟΙ

Group by ΚΩΔΑΝΑΓ

Having count(ΚΩΔΔΑΝ) >100;

b) Select ΚΩΔΑΝΑΓ, sum(ΚΩΔΔΑΝ)as PLHTHOS

From ΔΑΝΕΙΣΜΟΙ

Group by ΚΩΔΑΝΑΓ

Having sum(ΚΩΔΔΑΝ) >100;

c) Select ΚΩΔΑΝΑΓ, count(ΚΩΔΔΑΝ)as PLHTHOS

From ΔΑΝΕΙΣΜΟΙ

Where PLHTHOS >100;

d) Select ΚΩΔΑΝΑΓ, sum(ΚΩΔΔΑΝ)as PLHTHOS

From ΔΑΝΕΙΣΜΟΙ

Having PLHTHOS >100

Group by ΚΩΔΑΝΑΓ;

**21. Έστω πίνακας R που περιέχει N1 εγγραφές και ο πίνακας S που περιέχει N2 έγγραφες με  $N1 < N2$ . Ποιο είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος εγγραφών που θα προκύψει από την πράξη S-R;**

a) 0, N2

b) 0, N1

c) N2-N1, N2

d) N1-N2, N1

**22. Σύμφωνα με τους παραπάνω πίνακες (ΕΡΓΑΖΟΜΕΝΟΣ, ΤΜΗΜΑ, ΕΡΓΑΖΕΤΑΙ, ΕΡΓΑ). Ποιο ερώτημα SQL προβάλλει το ΑΦΜ και το πλήθος έργων για τους εργαζομένους που εργάζονται για συνολικά πάνω από 5 έργα.**

a) Select ΑΦΜ, count(\*)as PLHTHOS

From ΕΡΓΑΖΕΤΑΙ

Group by ΑΦΜ

Having count(ΑΦΜ) >5

b) Select ΑΦΜ, sum(ΚΕΡ)as PLHTHOS

From ΕΡΓΑΖΕΤΑΙ

Group by ΑΦΜ

Having sum(ΑΦΜ) >5

c) Select ΑΦΜ, count(ΑΦΜ)as PLHTHOS

From ΕΡΓΑΖΕΤΑΙ

Where PLHTHOS >5

Group by ΑΦΜ

d) Select ΑΦΜ, sum(ΑΦΜ)as PLHTHOS

From ΕΡΓΑΖΕΤΑΙ

Having PLHTHOS >5

Group by ΑΦΜ

**23. Κατά τη μετατροπή του μοντέλου E-R σε σχεσιακό, τι κάνουμε με ένα σύνθετο γνώρισμα;**

a) Το μετατρέπουμε σε απλό γνώρισμα

b) Το μετατρέπουμε σε πλειότιμο γνώρισμα

c) Κρατάμε μόνο τα επί μέρους γνωρίσματα του

d) Δημιουργούμε νέο πίνακα με τα επί μέρους γνωρίσματα του σύνθετου γνωρίσματος και ξένα κλειδιά τα πρωτεύοντα κλειδιά των συσχετιζόμενων πινάκων

**24. Οι παρακάτω πίνακες αποτελούνται από τα εξής πεδία :**

**ΙΔΙΟΚΤΗΤΕΣ: ΚΙ, ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΤΗΛ, ΟΔΟΣ, ΠΟΛΗ**

**ΚΑΤΕΧΟΥΝ: ΚΙ, ΚΑ, ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ**

**ΑΚΙΝΗΤΑ: ΚΑ , ΘΕΣΗ , ΕΚΤΑΣΗ, ΠΟΣΟΣΤΟ\_ΚΑΛΥΨΗΣ**

**Ποιο ερώτημα SQL προβάλλει την συνολική έκταση που κατέχουν οι ιδιοκτήτες στη θέση των Σερρών και υπερβαίνει τα 500 τ. μ;**

a) Select   ΚΑΤΕΧΟΥΝ.ΚΙ, Sum(ΕΚΤΑΣΗ \* ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ)  
AS ΣΥΝΟΛ\_ΙΔ\_ΕΚΤΑΣΗ  
From    ΚΑΤΕΧΟΥΝ INNER JOIN  ΑΚΙΝΗΤΑ ON ΚΑΤΕΧΟΥΝ.ΚΑ= ΑΚΙΝΗΤΑ.ΚΑ  
Where    ΣΥΝΟΛ\_ΙΔ\_ΕΚΤΑΣΗ > 500 and ΠΟΛΗ='ΣΕΡΡΕΣ'

Group by  ΚΑΤΕΧΟΥΝ.ΚΙ

b) Select   ΚΑΤΕΧΟΥΝ.ΚΙ, Sum(ΕΚΤΑΣΗ \* ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ)  
AS ΣΥΝΟΛ\_ΙΔ\_ΕΚΤΑΣΗ  
From    ΚΑΤΕΧΟΥΝ INNER JOIN  ΑΚΙΝΗΤΑ ON ΚΑΤΕΧΟΥΝ.ΚΑ= ΑΚΙΝΗΤΑ.ΚΑ  
Where    ΘΕΣΗ ='ΣΕΡΡΕΣ'

Group by  ΚΑΤΕΧΟΥΝ.ΚΙ

Having   Sum(ΕΚΤΑΣΗ \* ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ) >500

c) Select   ΚΑΤΕΧΟΥΝ.ΚΙ, Sum(ΕΚΤΑΣΗ \* ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ)  
AS ΣΥΝΟΛ\_ΙΔ\_ΕΚΤΑΣΗ  
From    ΙΔΙΟΚΤΗΤΕΣ INNER JOIN  ΑΚΙΝΗΤΑ ON ΚΑΤΕΧΟΥΝ.ΚΑ= ΑΚΙΝΗΤΑ.ΚΑ  
Where    ΣΥΝΟΛ\_ΙΔ\_ΕΚΤΑΣΗ > 500 and ΘΕΣΗ ='ΣΕΡΡΕΣ'

Group by  ΚΑΤΕΧΟΥΝ.ΚΙ

d) Select   ΚΑΤΕΧΟΥΝ.ΚΙ, Sum(ΕΚΤΑΣΗ \* ΠΟΣΟΣΤΟ\_ΣΥΝΙΔΙΟΚΤΗΣΙΑΣ)  
AS ΣΥΝΟΛ\_ΙΔ\_ΕΚΤΑΣΗ  
From    ΙΔΙΟΚΤΗΤΕΣ,  ΚΑΤΕΧΟΥΝ, ΑΚΙΝΗΤΑ  
Where    ΚΑΤΕΧΟΥΝ.ΚΑ= ΑΚΙΝΗΤΑ.ΚΑ AND ΘΕΣΗ='ΣΕΡΡΕΣ'

Group by  ΚΑΤΕΧΟΥΝ.ΚΙ

Having   ΣΥΝΟΛ\_ΙΔ\_ΕΚΤΑΣΗ >500

25. Έστω ο πίνακας R που περιέχει N1 εγγραφές και ο πίνακας S που περιέχει N2 εγγραφές με  $N1 > N2$ . Ποιο είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος των εγγράφων που θα προκύψει από την πράξη RUS;

- a) 0, N2
- b) 0, N1
- c)  $N1 - N2$ , N1
- d) N1,  $N1 + N2$

26. Σύμφωνα με τους παραπάνω πίνακες (ΠΕΛΑΤΕΣ, ΕΝΟΙΚΙΑΖΕΙ, ΤΑΙΝΙΕΣ, ΚΑΤΗΓΟΡΙΕΣ). Ποιο ερώτημα SQL προβάλλει τον κωδικό πελάτη ΚΠ και το πλήθος ενοικιάσεων για τους πελάτες που έχουν συνολικά πάνω από 10 ενοικιάσεις;

a) Select ΚΠ, sum(ΚΤ)as PLHTHOS

From ΕΝΟΙΚΙΑΖΕΙ

Group by ΚΠ

Having sum(ΚΤ) >10

b) Select ΚΠ, count(\*)as PLHTHOS

From ΕΝΟΙΚΙΑΖΕΙ

Group by ΚΠ

Having count(ΚΠ) >10

c) Select ΚΠ, count(ΚΠ)as PLHTHOS

From ΕΝΟΙΚΙΑΖΕΙ

Where PLHTHOS >10

Group by ΚΠ

d) Select ΚΠ, sum(ΚΠ)as PLHTHOS

From ΕΝΟΙΚΙΑΖΕΙ

Having PLHTHOS >100

Group by ΚΠ

**27. Πότε επιτρέπεται οι τύποι των συσχετίσεων να έχουν γνωρίσματα;**

- a) Ποτέ
- b) Πάντα
- c) Μόνο στην περίπτωση που ο τύπος συσχέτισης είναι N: M
- d) Μόνο στην περίπτωση που ο τύπος συσχέτισης είναι 1: N

**28. Ο παρακάτω πίνακας αποτελείται από τα εξής πεδία : (ΤΑΙΝΙΕΣ : ΚΤ, ΤΙΤΛΟΣ, ΔΙΑΡΚΕΙΑ,ΚΑΤΗΓΟΡΙΑ). Ποιο ερώτημα προβάλλει το πλήθος των ΤΑΙΝΙΩΝ ανά ΚΑΤΗΓΟΡΙΑ;**

a) Select ΚΑΤΗΓΟΡΙΑ, Sum(ΚΤ) AS ΠΛΗΘΟΣ

From ΤΑΙΝΙΕΣ

Group by ΚΑΤΗΓΟΡΙΑ

b) Select ΚΑΤΗΓΟΡΙΑ, Count(ΤΙΤΛΟΣ) AS ΠΛΗΘΟΣ

From ΤΑΙΝΙΕΣ

Group by ΚΑΤΗΓΟΡΙΑ

c) Select ΚΑΤΗΓΟΡΙΑ, Count(ΚΤ) AS ΠΛΗΘΟΣ

From ΤΑΙΝΙΕΣ

Group by ΤΑΙΝΙΕΣ.ΚΤ

d) Select ΤΑΙΝΙΕΣ.ΚΤ, Count(ΚΑΤΗΓΟΡΙΑ) AS ΠΛΗΘΟΣ

From ΤΑΙΝΙΕΣ

Group by ΤΑΙΝΙΕΣ.ΚΤ



**29. Πότε ένα κλειδί θεωρείται σωστή επιλογή;**

- a) Όταν μπορεί να αναγνωρίζει κάθε γραμμή μοναδικά.
- b) Δεν πρέπει να είναι ποτέ κενό ή με μηδενική τιμή - να περιέχει πάντα μία τιμή.
- c) Οι τιμές που περιέχει να αλλάζουν σπάνια (ιδανικά ποτέ).
- d) Οι τιμές που περιέχει να αλλάζουν συνέχεια για ασφάλεια

**30. Ποια από τις παρακάτω προτάσεις είναι λανθασμένη; (πλεονεκτήματα ΒΔ)**

- a) Μείωση του πλεονασμού (επανάληψη) των αποθηκευμένων δεδομένων
- b) Αποφυγή ασυνεπειών
- c) Κοινόχρηστα δεδομένα
- d) Μη διατήρηση της ακεραιότητας

**Λύσεις**

1. (a), 2. (c), 3. (c), 4. (c) ,5. (c) , 6. (d),7. (d), 8. (a), 9. (d),10. (b),11. (c),12. (b), 13. (c) ,14. (a) ,15. (d), 16. (d),17. (c), 18. (c), 19. (a), 20. (c) , 21. (c), 22. (a) , 23. (c), 24. (b), 25. (d), 26. (b) , 27. (b) , 28. (b), 29. (d), 30. (d)

## Ερωτήσεις του μαθήματος «Ειδικά θέματα Βάσεων Δεδομένων»

**1. Ποια από τις παρακάτω διαδικασίες είναι σωστή ώστε να ενημερώνει σε πίνακα ένα πίνακα ΔΩΜΑΤΙΟ την ΤΙΜΗ του συγκεκριμένου δωματίου.**

a) CREATE PROCEDURE PROC1 (@Kd, @Timi) as

UPDATE ΔΩΜΑΤΙΟ VALUES ΤΙΜΗ= @Timi AND ΚΔ=@Kd

b) CREATE PROCEDURE PROC1 (@Kd INT, @Timi float) as

UPDATE ΔΩΜΑΤΙΟ VALUES (ΤΙΜΗ = @Timi, ΚΔ=@Kd)

c) CREATE PROCEDURE PROC1 (@Kd INT, @Timi float) as

UPDATE ΔΩΜΑΤΙΟ SET ΤΙΜΗ= @Timi WHERE ΚΔ=@Kd

d) CREATE PROCEDURE PROC1 (@Kd, @Timi) as

UPDATE ΔΩΜΑΤΙΟ SET ΤΙΜΗ = @Timi WHERE ΚΔ=@Kd

**2. Ποιά είναι η σύνταξη της εντολής με την οποία ο χρήστης User1 δεν μπορεί να τροποποιεί, να διαγράψει αλλά μόνο να εισάγει δεδομένα στον πίνακα ΠΕΛΑΤΕΣ.**

a) Grant Insert (ΠΕΛΑΤΕΣ) TO User1

b) Grant Insert, not Delete, not Update ON ΠΕΛΑΤΕΣ To User1

c) Grant only Insert on ΠΕΛΑΤΕΣ TO User1

d) Grant Insert ON ΠΕΛΑΤΕΣ TO User1

**3. Κατά την Φυσική αποθήκευση (Physical storage) μιας ΒΔ αποθηκεύονται στον δίσκο τα:**

a) Λεξικό δεδομένων (data dictionary), Ευρετήρια – Δείκτες (index), Όψεις (Views), Στιγμιότυπα (Instances), Αρχεία δεδομένων (data files)

b) Αρχεία δεδομένων (data files), Λεξικό δεδομένων (data dictionary), Ευρετήρια – Δείκτες (index), Όψεις (Views), Κλειδιά (keys)

c) Ευρετήρια – Δείκτες (index), Λεξικό δεδομένων (data dictionary), Αρχεία δεδομένων (data files)

d) Στιγμιότυπα (Instances), Λεξικό δεδομένων (data dictionary), Ευρετήρια – Δείκτες (index)

**4. Ποιό είναι το λάθος στο παρακάτω XML έγγραφο:**

```
<?xml version="1.0"?>
```

```
< ! DOCTYPE ΠΕΛΑΤΕΣ [
```

```
    < ! ELEMENT ΠΕΛΑΤΕΣ (ΕΠΙΘΕΤΟ, ΟΝΟΜΑ, ΠΟΛΗ)>
```

```
    < ! ELEMENT ΕΠΙΘΕΤΟ (#PCDATA)>
```

```
    < ! ELEMENT ΟΝΟΜΑ (#PCDATA)>
```

```
    < ! ELEMANT ΠΟΛΗ (#PCDATA)>
```

```
]>
```

a) Η πρώτη γράμμα έπρεπε να είναι ως εξής: <?xml version="1.0" >

b) Η Τρίτη γραμμή: < ! ELEMENT ΠΕΛΑΤΕΣ (ΕΠΙΘΕΤΟ, ΟΝΟΜΑ ΠΟΛΗ) > δεν χρειάζεται.

c) Όλα τα πεδία γράφουν ως εξής: < ELEMENT> ΕΠΙΘΕΤΟ </ ELEMENT> κτλ.

d) Δεν υπάρχει λάθος

**5. Έστω ο παρακάτω πίνακας: ΠΡΟΪΟΝΤΑ (ΚΕ, ΕΙΔΟΣ, ΤΙΜΗ\_ΑΓΟΡΑΣ, ΑΠΟΘΕΜΑ, ΚΑΤΗΓΟΡΙΑ, ΠΡΟΜΗΘΕΥΤΗΣ) .Εάν γνωρίζεται ότι κάθε προμηθευτής μπορεί να προμηθεύει προϊόντα μιας μόνο κατηγορίας και υπάρχει και η συναρτησιακή εξάρτηση ΠΡΟΜΗΘΕΥΤΗΣ->ΚΑΤΗΓΟΡΙΑ τότε ο πίνακας ΠΡΟΪΟΝΤΑ ποια κανονική μορφή δεν υπακούει;**

a) Την 1KM

b) Την 4KM

c) Την 5KM

d) Την Boyce-Codd KM

**6. Δίνεται το αρχικό χρονοπρόγραμμα**

S1: R1(X) W1(X) R1(Y) W1(Y) C1 R2(X) W2(X) C2

a) S2: R1(X) W1(X) R1(Y) R2(X) W2(X) C2 W1(Y) C1

b) S3: W1(X) R1(X) W2(X) R2(X) R1(Y) C2 W1(Y) C1

c) S4: R1(X) R2(X) W2(X) C2 W1(X) R1(Y) W1(Y) C1

d) S5: R1(X) R2(X) W1(X) R1(Y) W2(X) C2 W1(Y) C1

**7. Μια σχέση είναι σε 5 KM όταν:**

a) Όταν δεν περιέχει εξαρτήσεις σύζευξης

b) Όταν περιέχει εξαρτήσεις σύζευξης

c) Όταν δεν περιέχει εξαρτήσεις πολλαπλών τιμών

d) Όταν περιέχει εξαρτήσεις πολλαπλών τιμών

**8. Ποιά είναι η σύνταξη της εντολής με την οποία ο χρήστης bbUser μπορεί να τροποποιεί (μόνο) τι γνώρισμα e-mail της σχέσης Customer.**

a) Grant Update (e-mail) ON Customer TO bbUser.

b) Grant Update(e-mail) TO Customer ON bbUser.

c) Grant Update (e-mail) FOR Customer TO bbUser.

d) Grant Update FOR e-mail TO Customer TO bbUser.

**9. Ποιό από τα παρακάτω είναι αληθές:**

a) Το OLE DB αποτέλεσε την βάση για την άφιξη του ADO.

b) Το ADO αποτέλεσε την βάση για την άφιξη του OLD DB.

c) Το OLE DB αποτέλεσε την βάση για την άφιξη του ODBC.

d) Το BDE αποτέλεσε την βάση για την άφιξη του ODBC.

10. Σύμφωνα με ένα απλό μηχανισμό κρυπτογράφησης δεδομένων που έχετε διδαχθεί κρυπτογραφήστε την φράση ΚΑΛΗ ΕΡΓΙΑ με λατινικούς χαρακτήρες όταν γνωρίζεται ότι το κλειδί είναι η λέξη LAST (οι αριθμοί που αντιστοιχούν σε κάθε γράμμα είναι: κενό=00, A=01...έως το Z=26). Η φράση που θα προκύψει είναι:

- a) WBDALFHBEZPB
- b) WBDALFHBE PBF
- c) WBDALFHBEZPBMAS
- d) WBDALFHBEYQBNATU

11. Ο επεξεργαστής ερωτημάτων (Query processor) είναι ένα από τα συστατικά μέρη ενός συστήματος Βάσεων Δεδομένων που αποτελείται από:

- a) DML compiler, DDL interpreter, File management, Buffer management.
- b) DML compiler, File management, Buffer management, Query processing unit.
- c) DML compiler, Embedded DML precompiler, File management, Buffer management.
- d) DML compiler, Embedded DML precompiler, DDL interpreter, Query processing unit.

12. Έστω οι παρακάτω πίνακες με τα πεδία τους

$R=(A, B, \Gamma, \Delta)$  και  $S=(E, B, \Delta)$

Πώς ορίζεται ο τελεστής σύζευξης στη σχεσιακή άλγεβρα;

- a)  $\Pi r.A, r.B, r.\Gamma, r.\Delta, s.E (\sigma r.B=s.B \wedge r.\Delta = s.\Delta(r, s))$
- b)  $\Pi r.A, s.B, r.\Gamma, r.\Delta, s.E (\sigma r.A=s.A \wedge r.B = s.B(r, s))$
- c)  $\Pi r.A, r.B, r.\Gamma, r.\Delta, s.E (\sigma r.B=s.B \wedge r.\Delta = s.\Delta(r \times s))$
- d)  $\Pi r.A, s.B, r.\Gamma, r.\Delta, s.E (\sigma r.B=s.B \wedge r.B = s.B(r \times s))$

**13. Ποίες είναι οι βασικές συνιστώσες ενός μοντέλου δεδομένων;**

- a) Δομές (structures), Εγγραφές (records), Πεδία (fields).
- b) Εγγραφές (records), Πεδία (fields), Περιορισμοί (constraints).
- c) Εγγραφές (records), Πράξεις (operation), Πεδία (fields).
- d) Δομές (structures), Πράξεις (operation), Περιορισμοί (constraints).

**14. Έχουμε δημιουργήσει την παρακάτω διαδικασία η οποία διαγράφει τις εγγραφές από τον πίνακα ΠΕΛΑΤΗΣ.**

**CREATE PROCEDURE PROC1 (@KP int) as**

**Delete from ΠΕΛΑΤΗΣ where ΚΠ=@ΚΡ**

**Τι θα συμβεί εάν εκτελούσατε την διαδικασία αυτή με την εντολή: EXEC PROC1 2**

- a) Θα διαγράψει από τον πίνακα ΠΕΛΑΤΗΣ την εγγραφή με ΚΠ=2
- b) Υπάρχει λάθος στην εντολή εκτέλεσης. Το σωστό είναι RUN PROC1(2)
- c) Δεν επιτρέπεται να διαγράψει να την εγγραφή αυτή γιατί παραβιάζεται η αρχή της ακεραιότητας αναφοράς.
- d) Δεν επιτρέπεται να διαγράψει να την εγγραφή αυτή γιατί παραβιάζεται η αρχή της οντότητας του πίνακα.

15. Έχουμε δημιουργήσει την παρακάτω συνάρτηση η οποία παρουσιάζει τα στοιχεία των πελατών που έκαναν κρατήσεις για ένα συγκεκριμένο ΕΙΔΟΣ δωματίου.

```
CREATE FUNCTION ΠΕΛ_ΔΩΜ(@ΕΙΔΟΣ VARCHAR(10))
```

```
RETURNS TABLE AS
```

```
Begin
```

```
RETURN
```

```
(SELECT ΕΠΩΝΥΜΙΑ, ΑΦΜ, ΤΗΛ
```

```
FROM ΠΕΛΑΤΗΣ, ΚΡΑΤΗΣΗ, ΔΕΣΜΕΥΣΗ, ΔΩΜΑΤΙΟ
```

```
WHERE ΠΕΛΑΤΗΣ.ΚΠ = ΚΡΑΤΗΣΗ.ΚΠ AND  
ΚΡΑΤΗΣΗ.ΚΚΡ=ΔΕΣΜΕΥΣΗ.ΚΚΡ
```

```
AND ΔΕΣΜΕΥΣΗ.ΚΔ=ΔΩΜΑΤΙΟ.ΚΔ AND ΕΙΔΟΣ = @ΕΙΔΟΣ)
```

```
End
```

Με ποια από τις παρακάτω εντολές θα εκτελεστεί η συνάρτηση και θα εμφανιστούν οι πελάτες που κράτησαν το δωμάτιο είδους ΣΟΥΙΤΑ;

a) SELECT \* FROM ΔΩΜΑΤΙΟ WHERE ΕΙΔΟΣ='ΣΟΥΙΤΑ'

b) SELECT \* FROM ΠΕΛ\_ΔΩΜ('ΣΟΥΙΤΑ')

c) EXEC ΠΕΛ\_ΔΩΜ 'ΣΟΥΙΤΑ'

d) EXEC FUNCTION ΠΕΛ\_ΔΩΜ('ΣΟΥΙΤΑ')

16. Έχει δημιουργηθεί ο παρακάτω trigger ο οποίος κάθε φορά που γίνεται εισαγωγή στον πίνακα ΔΕΣΜΕΥΣΗ, υπολογίζει τον τζίρο του κάθε δωματίου και τον καταχωρεί στο πεδίο ΤΖΙΡΟΣ του πίνακα ΔΩΜΑΤΙΟ.

```
CREATE TRIGGER T1 ON ΔΕΣΜΕΥΣΗ
FOR INSERT AS
BEGIN
DECLARE @ΚΔ INT, @SYNOLO INT, @ΤΖΙΡΟΣ FLOAT
SELECT @ΚΔ =SELECT ΚΔ FROM INSERTED
SELECT @SYNOLO =SELECT DATEDIFF (DAY, ΗΜΕΡΟΜΗΝΙΑ_ΑΦΙΞΗΣ,
ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΧΩΡΗΣΗΣ)
FROM INSERTED
IF @SYNOLO=0 THEN @SYNOLO=1
SELECT @ΤΖΙΡΟΣ=SELECT @SYNOLO*ΤΙΜΗ FROM ΔΩΜΑΤΙΟ WHERE ΚΔ=@ΚΔ
UPDATE ΔΩΜΑΤΙΟ SET ΤΖΙΡΟΣ = ΤΖΙΡΟΣ+@ΤΖΙΡΟΣ WHERE ΚΔ=@ΚΔ
END
```

Εάν γίνει μία καταχώρηση στον πίνακα ΔΕΣΜΕΥΣΗ με τιμές (2,3,'1-7-2014','2-7-2014') ποίο το ποσό του τζίρου που θα εμφανιστεί τελικά;

- a) 300€
- b) 100€
- c) 50€
- d) 150€

17. Έστω ο παρακάτω πίνακας ο οποίος δεν υπακούει την Boyce-Codd Κανονική μορφή

ΠΡΟΪΟΝΤΑ: ΚΕ , ΕΙΔΟΣ , ΤΙΜΗ\_ΑΓΟΡΑΣ, ΑΠΟΘΕΜΑ, ΚΑΤΗΓΟΡΙΑ, ΠΡΟΜΗΘΕΥΤΗΣ

Εάν γνωρίζετε ότι ο κάθε προμηθευτής μπορεί να προμηθεύει προϊόντα μιας μόνο κατηγορίας και κάθε είδος μπορεί να ανήκει σε μία κατηγορία και να έχει έναν προμηθευτή. Ποία συναρτησιακή εξάρτηση δεν μπορεί να υπάρξει στον πίνακα



- a) ΠΡΟΜΗΘΕΥΤΗΣ -->ΚΑΤΗΓΟΡΙΑ
- b) ΚΕ --> ΚΑΤΗΓΟΡΙΑ
- c) ΚΕ --> ΠΡΟΜΗΘΕΥΤΗΣ
- d) ΚΕ -->(ΠΡΟΜΗΘΕΥΤΗΣ, ΚΑΤΗΓΟΡΙΑ)

**18. Ποία είναι η σύνταξη της εντολής με την οποία από τον χρήστη GRAM αφι το δικαίωμα της εισαγωγής από τον πίνακα ΠΕΛΑΤΕΣ.**

- a) Grant (insert into) on ΠΕΛΑΤΕΣ TO GRAM
- b) Grant insert ON ΠΕΛΑΤΕΣ from GRAM
- c) Revoke insert ON ΠΕΛΑΤΕΣ from GRAM
- d) Remove insert ON ΠΕΛΑΤΕΣ from GRAM

**19. Ποια από τις παρακάτω εκφράσεις δεν είναι σωστή;**

- a) Η γενική μορφή της βάσης καλείται σχήμα και ορίζεται κατά τη δημιουργία της.
- b) Ένα σχήμα βάσης δεδομένων αλλάζει συχνά
- c) Οι πληροφορίες σε μία βάση δεδομένων σε ένα συγκεκριμένο χρονικό σημείο καλούνται στιγμιότυπα (instance) της βάσης δεδομένων.
- d) Κάθε αλλαγή των δεδομένων δημιουργεί καινούργια στιγμιότυπα της βάσης δεδομένων.

**20. Έχουμε δημιουργήσει την παρακάτω διαδικασία η οποία διαγράφει εγγραφές από τον πίνακα ΠΕΛΑΤΗΣ.**

**CREATE PROCEDURE PROC1 (@Kp int) as**

**Delete from ΠΕΛΑΤΗΣ where ΚΠ=@Kp**

**Τι θα συμβεί εάν εκτελούσατε την διαδικασία αυτή με την εντολή: EXEC PROC1 2**

- a) Δεν επιτρέπεται να διαγράψει την εγγραφή αυτή γιατί παραβιάζεται η αρχή της ακεραιότητας αναφοράς.
- b) Δεν επιτρέπεται να διαγράψει την εγγραφή αυτή γιατί παραβιάζεται η αρχή της οντότητας του πίνακα
- c) Θα διαγράψει από τον πίνακα ΠΕΛΑΤΗΣ την εγγραφή με ΚΠ=2
- d) Υπάρχει λάθος στην εντολή εκτέλεσης. Το σωστό είναι είναι RUN PROC1(2)

21. Έχουμε δημιουργήσει την παρακάτω συνάρτηση η οποία παρουσιάζει τα στοιχεία των ΔΩΜΑΤΙΩΝ που έκαναν κρατήσεις πελάτες από συγκεκριμένη ΠΟΛΗ.

```
CREATE FUNCTION ΠΕΛ_ΔΩΜ(@ΠΟΛΗ VARCHAR(10))
```

```
RETURNS TABLE AS
```

```
Begin
```

```
RETURN
```

```
(SELECT ΠΟΛΗ, ΕΙΔΟΣ, ΚΛΙΝΕΣ, ΤΙΜΗ
```

```
FROM ΠΕΛΑΤΗΣ, ΚΡΑΤΗΣΗ, ΔΕΣΜΕΥΣΗ, ΔΩΜΑΤΙΟ
```

```
WHERE ΠΕΛΑΤΗΣ.ΚΠ = ΚΡΑΤΗΣΗ.ΚΠ AND  
ΚΡΑΤΗΣΗ.ΚΚΡ=ΔΕΣΜΕΥΣΗ.ΚΚΡ
```

```
AND ΔΕΣΜΕΥΣΗ.ΚΔ=ΔΩΜΑΤΙΟ.ΚΔ AND ΠΟΛΗ = @ΠΟΛΗ)
```

```
End
```

Με ποια από τις παρακάτω εντολές θα εκτελεστεί η συνάρτηση και θα εμφανιστούν τα στοιχεία των ΔΩΜΑΤΙΩΝ που έκαναν κρατήσεις οι πελάτες από την ΠΟΛΗ ΣΕΡΡΕΣ;

- a) SELECT \* FROM ΠΕΛ\_ΔΩΜ('ΣΕΡΡΕΣ')
- b) RUN ΠΕΛ\_ΔΩΜ('ΣΕΡΡΕΣ')
- c) EXEC ΠΕΛ\_ΔΩΜ('ΣΕΡΡΕΣ')
- d) SELECT \* FROM ΠΕΛΑΤΗΣ WHERE ΠΟΛΗ='ΣΕΡΡΕΣ'

22. Έχει δημιουργηθεί ο παρακάτω trigger ο οποίος κάθε φορά που γίνεται εισαγωγή στον πίνακα ΔΕΣΜΕΥΣΗ, υπολογίζει τον τζίρο του κάθε πελάτη και τον καταχωρεί στο πεδίο ΤΖΙΡΟΣ του πίνακα ΠΕΛΑΤΗΣ.

```
CREATE TRIGGER T1 ON ΔΕΣΜΕΥΣΗ
FOR INSERT AS
BEGIN
DECLARE @ΚΔ INT, @SYNOLO INT, @ΤΖΙΡΟΣ FLOAT, @ΚΚΡ INT, @ΚΠ INT
SELECT @ΚΔ =SELECT ΚΔ FROM INSERTED
SELECT @ΚΚΡ =SELECT ΚΚΡ FROM INSERTED
SELECT @SYNOLO =SELECT DATEDIFF(DAY, ΗΜΕΡΟΜΗΝΙΑ_ΑΦΙΞΗΣ,
ΗΜΕΡΟΜΗΝΙΑ_ΑΝΑΧΩΡΗΣΗΣ)
FROM INSERTED
IF @SYNOLO=0 THEN @SYNOLO=1
SELECT @ΤΖΙΡΟΣ=SELECT @SYNOLO*ΤΙΜΗ FROM ΔΩΜΑΤΙΟ WHERE ΚΔ=@ΚΔ
SELECT @ΚΠ =SELECT ΚΠ FROM ΚΡΑΤΗΣΗ WHERE ΚΚΡ=@ΚΚΡ
UPDATE ΠΕΛΑΤΗΣ SET ΤΖΙΡΟΣ = ΤΖΙΡΟΣ+@ΤΖΙΡΟΣ WHERE ΚΠ=@ΚΠ
END
```

Εάν γίνει μία καταχώρηση στον πίνακα (2,3,'1-7-2014','2-7-2014') ποιο το ποσό του τζίρου που θα εμφανιστεί τελικά;

- a) ΤΖΙΡΟΣ =258€
- b) ΤΖΙΡΟΣ =308€
- c) ΤΖΙΡΟΣ =50€
- d) ΤΖΙΡΟΣ =150€

**23. Ποια είναι η σύνταξη της εντολής με την οποία ο χρήστης TOM μπορεί να εισάγει, να διαγράψει αλλά όχι να τροποποιεί τα δεδομένα στον πίνακα ΠΕΛΑΤΕΣ.**

- a) Select Insert, Delete FROM ΠΕΛΑΤΕΣ TO TOM
- b) Grant Insert, Delete ON ΠΕΛΑΤΕΣ TO TOM
- c) REVOKE Insert, Delete ON ΠΕΛΑΤΕΣ FROM TOM
- d) Grant Insert, Delete, no Update ON ΠΕΛΑΤΕΣ TO TOM

**24. Έστω ο παρακάτω πίνακας Εστιατόριο-Διανομή με πεδία (ΕΙΔΟΣ, ΤΥΠΟΣ, ΠΕΡΙΟΧΗ). Το κάθε εστιατόριο μπορεί να διανέμει το κάθε είδος των προϊόντων του σε συγκεκριμένες περιοχές, και ο παραπάνω πίνακας δεν υπακούει την 4 κανονική μορφή γιατί:**

- a) Τιμές του πεδίου ΠΕΡΙΟΧΗ που αντιστοιχούν στο ζεύγος (ΕΙΔΟΣ, ΤΥΠΟΣ) εξαρτώνται μόνο από το ΕΙΔΟΣ και όχι από τον ΤΥΠΟ
- b) Οι τιμές του πεδίου ΤΥΠΟΣ δεν εξαρτώνται συναρτησιακά το πεδίο ΠΕΡΙΟΧΗ
- c) Υπάρχει μεταβατική συναρτησιακή εξάρτηση μεταξύ των πεδίων  
ΕΙΔΟΣ -->ΤΥΠΟΣ -->ΠΕΡΙΟΧΗ
- d) Υπάρχουν εξαρτήσεις σύζευξης μεταξύ των πεδίων (ΕΙΔΟΣ, ΤΥΠΟΣ) και (ΕΙΔΟΣ, ΠΕΡΙΟΧΗ)

**25. Η συντομογραφία DML προκύπτει από:**

- a) Define Markup Language
- b) Data Markup Language
- c) Data Management Language
- d) Data Manipulation Language

**26. Ποίο είναι το λάθος στον παρακάτω κώδικα XML:**

```
< ΠΕΛΑΤΕΣ >  
  
    <!ELEMENT ΚΩΔΙΚΟΣ 1234 >  
  
    <!ELEMENT ΟΝΟΜΑ ΔΕΡΟΝ >  
  
    <!ELEMENT ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ 12-5-2020</ ΗΜΕΡΟΜΗΝΙΑ_ΛΗΞΗΣ >  
  
    <!ELEMENT ΤΙΜΗ 1.5 </ ΤΙΜΗ >  
  
</ ΠΕΛΑΤΕΣ >
```

- a) Δεν υπάρχει λάθος
- b) Η πρώτη γραμμή: <ΠΕΛΑΤΕΣ> έπρεπε να είναι < ΠΕΛΑΤΕΣ >
- c) Όλα τα πεδία να γράφουν το εξής: <ΚΩΔΙΚΟΣ> ... </ ΚΩΔΙΚΟΣ>
- d) Όλα τα πεδία να γράφουν το εξής: <!ELEMENT ... />

**27. Ποία από τις παρακάτω διαδικασίες είναι σωστή ώστε να ενημερώνει έναν πίνακα ΚΑΤΑΘΕΣΕΙΣ δίνοντας σε μία μεταβλητή εισόδου ένα πραγματικό αριθμό που θα αυξάνει τις τιμές του πεδίου ΠΟΣΟ\_ΚΑΤΑΘΕΣΗΣ κατά 5%.**

- a) CREATE PROCEDURE PROC1 (\$POSOSTO FLOAT) LIKE  
UPDATE ΚΑΤΑΘΕΣΕΙΣ VALUE POSO\_ΚΑΤΑΘΕΣΗΣ /(1+\$POSOSTO:100)
- b) CREATE PROCEDURE PROC1 (@POSOSTO=5 REAL) AS  
UPDATE ΚΑΤΑΘΕΣΕΙΣ SET POSO\_ΚΑΤΑΘΕΣΗΣ =  
POSO\_ΚΑΤΑΘΕΣΗΣ\*(1+@POSOSTO/100)
- c) CREATE PROCEDURE PROC1 (@POSOSTO=5 REAL) AS  
UPDATE ΚΑΤΑΘΕΣΕΙΣ SET POSO\_ΚΑΤΑΘΕΣΗΣ = POSO\_ΚΑΤΑΘΕΣΗΣ \* @POSOSTO
- d) CREATE PROCEDURE PROC1 (\$POSOSTO REAL) LIKE  
UPDATE ΚΑΤΑΘΕΣΕΙΣ VALUE POSO\_ΚΑΤΑΘΕΣΗΣ \* \$POSOSTO:100

**28. Οι παρακάτω πίνακες αποτελούνται από τα εξής πεδία:**

**ΠΕΛΑΤΕΣ : ΚΩΔΠΕΛ , ΕΠΙΘΕΤΟ , ΟΝΟΜΑ, ΤΗΛ**

**ΠΑΡΑΓΓΕΛΙΕΣ : ΚΩΔΠΑΡ , ΚΩΔΠΕΛ , ΗΜΕΡΟΜΗΝΙΑ**

**Το πεδίο ΚΩΔΠΑΡ είναι ο κωδικός παραγγελιών και ΚΩΔΠΕΛ ο κωδικός πελατών.**

**Ποιό από τα παρακάτω ερωτήματα SQL προβάλλει ΚΩΔΠΕΛ τον συνολικό αριθμό παραγγελιών για τους πελάτες που έχουν κάνει συνολικά πάνω από 10 παραγγελίες.**

a) select ΚΩΔΠΕΛ, count(ΚΩΔΠΑΡ)as PLHTHOS

from ΠΑΡΑΓΓΕΛΙΕΣ

group by ΚΩΔΠΕΛ

having count(ΚΩΔΠΑΡ) >10

b) select ΚΩΔΠΕΛ, sum(ΚΩΔΠΑΡ)as PLHTHOS

from ΠΑΡΑΓΓΕΛΙΕΣ

group by ΚΩΔΠΕΛ

having sum(ΚΩΔΠΑΡ) >10

c) select ΚΩΔΠΕΛ, count(ΚΩΔΠΑΡ)as PLHTHOS

from ΠΑΡΑΓΓΕΛΙΕΣ

group by ΚΩΔΠΑΡ

having PLHTHOS >10

d) select ΚΩΔΠΕΛ, sum(ΚΩΔΠΑΡ)as PLHTHOS

from ΠΑΡΑΓΓΕΛΙΕΣ

having PLHTHOS >10

group by ΚΩΔΠΕΛ

**29. Κατά τη δημιουργία μιας συνάρτησης μιας συνάρτησης οριζόμενης από τον χρήστη, η χρήση της έκφρασης**

**WITH SCHEMABINDING** σημαίνει ότι:

- a) Δημιουργείται ένας τύπος δεδομένων της τιμής που επιστρέφεται από την συνάρτηση που ορίζεται από τον χρήστη.
- b) Η προσπάθεια τροποποίησης της δομής ενός αντικειμένου της Βάσης Δεδομένων στο οποίο αναφέρεται η συνάρτηση θα αποτύχει.
- c) Η προσπάθεια τροποποίησης της δομής ενός αντικειμένου τροποποίησης της δομής ενός αντικειμένου της Βάσης Δεδομένων στο οποίο αναφέρεται η συνάρτηση θα ολοκληρωθεί.
- d) Κρυπτογραφεί τις στήλες του πίνακα συστήματος που περιέχουν το κείμενο της πρότασης CREATE FUNCTION

**30. Ποια από τις παρακάτω διαδικασίες είναι σωστή ώστε να ενημερώνει έναν πίνακα ΚΑΤΑΘΕΣΕΙΣ δίνοντας σε μία μεταβλητή εισόδου ένα πραγματικό αριθμό που θα αυξάνει τις τιμές του πεδίου ΠΟΣΟ\_ΚΑΤΑΘΕΣΗΣ κατά 5%.**

a) CREATE PROCEDURE PROC1 (\$POSOSTO FLOAT) LIKE

UPDATE ΚΑΤΑΘΕΣΕΙΣ VALUE POSO\_ΚΑΤΑΘΕΣΗΣ /(1+\$POSOSTO)

b) CREATE PROCEDURE PROC1 (@POSOSTO=5 REAL) AS

UPDATE ΚΑΤΑΘΕΣΕΙΣ SET POSO\_ΚΑΤΑΘΕΣΗΣ =

POSO\_ΚΑΤΑΘΕΣΗΣ \*(1+@POSOSTO/100)

c) CREATE PROCEDURE PROC1 (@POSOSTO=5 REAL) AS

UPDATE ΚΑΤΑΘΕΣΕΙΣ SET POSO\_ΚΑΤΑΘΕΣΗΣ = POSO\_ΚΑΤΑΘΕΣΗΣ \* @POSOSTO

d) CREATE PROCEDURE PROC1 (\$POSOSTO REAL) LIKE

UPDATE ΚΑΤΑΘΕΣΕΙΣ VALUE POSO\_ΚΑΤΑΘΕΣΗΣ \* \$POSOSTO/100

**Λύσης:**

1. (c) ,2. (d) ,3. (c) , 4. (d) ,5. (d) ,6. (a) ,7. (a) ,8. (a) ,9. (a) ,10. (c) ,11. (d) ,12. (c) ,13. (d) ,14. (a) ,15. (b) , 16. (b) , 17. (a) , 18. (c) , 19. (b) , 20. (c) ,21. (a) , 22. (b) ,23. (b) , 24. (a) ,25. (d) ,26. (c),27. (b) , 28. (a) , 29. (b) ,30. (b)

## Ενότητα 7 – Ιστοσελίδα μαθημάτων Βάσεις Δεδομένων & Ειδικά θέματα Βάσεων Δεδομένων

### Εισαγωγή

Για την δημιουργία της ιστοσελίδας χρησιμοποιήθηκε το Wordpress και για την τοπική ανάπτυξη της το πακέτο εφαρμογών Xampp.

Η ιστοσελίδα στηρίχτηκε σε μια standard έκδοση του Wordpress με ένα προϋπάρχον θέμα και ελάχιστα πρόσθετα (plug-ins) καθώς οι απαιτήσεις της ιστοσελίδας δεν ήταν μεγάλες.

### Τι είναι το WordPress

Το WordPress είναι ανοικτού κώδικα λογισμικό ιστού και πλατφόρμα δημοσιεύσεων, γραμμένο σε PHP και MySQL. Συχνά τροποποιείται για χρήση ως Σύστημα διαχείρισης περιεχομένου. Έχει αρκετές δυνατότητες, όπως αρχιτεκτονικής για πρόσθετες λειτουργίες, και συστήματος προτύπων.

### Τι είναι Xampp

Το XAMPP είναι ένα πακέτο εφαρμογών λογισμικού ανοικτού κώδικα και cross-platform το οποίο περιέχει το server ιστοσελίδων http Apache, την βάση δεδομένων MySQL και ένα interpreter για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl.

### Περιγραφή Site



### Καλώς ήρθατε στον διαδικτυακό μας τόπο!!!

Posted on Τετάρτη, Σεπτεμβρίου 3rd, 2014 at 6:49 μμ.  
Written by [admin](#)

Σκοπός της δημιουργίας αυτής της ιστοσελίδας είναι να μπορεί ο κάθε ενδιαφερόμενος για το αντικείμενο των Βάσεων Δεδομένων και για το λογισμικό ελεύθερου και ανοιχτού κώδικα που τις διαχειρίζεται, να μπορεί να εμπλουτίσει τις γνώσεις του με τις διάφορες τεχνικές που υπάρχουν στα κεφάλαια του εκπαιδευτικού υλικού. Να μπορεί επίσης να κατεβάσει τον κώδικα των Project που υπάρχει σε κάθε κεφάλαιο και να ενημερώνεται συνεχώς για νέες τεχνολογίες και μεθόδους που αφορούν τις Βάσεις Δεδομένων και λογισμικά ελεύθερου και ανοιχτού κώδικα.

Εικόνα 186 - Home Page





Εργαστήρια  
Θεωρία

## Καλώς στον διαδικτυακό μας τόπο!!!

Posted on Τετάρτη, Σεπτεμβρίου 3rd, 2014 at 6:49 μμ.  
Written by [admin](#)

Σκοπός της δημιουργίας αυτής της ιστοσελίδας είναι να μπορεί ο κάθε ενδιαφερόμενος για το αντικείμενο των Βάσεων Δεδομένων και για το λογισμικό ελεύθερου και ανοιχτού κώδικα που τις διαχειρίζεται, να μπορεί να εμπλουτίσει τις γνώσεις του με τις διάφορες τεχνικές που υπάρχουν στα κεφάλαια του εκπαιδευτικού υλικού. Να μπορεί επίσης να κατεβάσει των κώδικα των Project που υπάρχει σε κάθε κεφάλαιο και να ενημερώνεται συνεχώς για νέες τεχνολογίες και μεθόδους που αφορούν τις Βάσεις Δεδομένων και λογισμικά ελεύθερου και ανοιχτού κώδικα.

localhost8079/bases\_dedomenwn/?page\_id=210

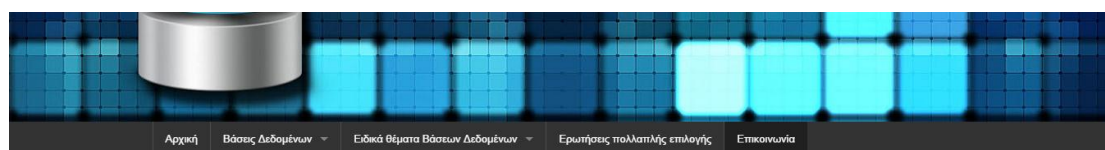
Εικόνα 187 - Χώρος εκπαιδευτικού υλικού



## Εργαστηριακή άσκηση 2



Εικόνα 116 - Χώρος εκπαιδευτικού υλικού PDF



## Επικοινωνία

<b>Δρ. Τσιμπήρης Αλκιβιάδης</b>	<b>Επιστημονικός Συνεργάτης</b>
- Φυσικός – Πανεπιστήμιο Ιωαννίνων	τηλ. 23210-55532
- Δρ. Βάσεων Δεδομένων και Εξόρυξης Γνώσης -ΑΠΘ	<b>e-mail:</b> <a href="mailto:alkisser@gmail.com">alkisser@gmail.com</a>
- Δρ. Φυσικοχημείας, Μοριακής Δυναμικής / Μηχανικής και Κβαντικής Χημείας- Πανεπιστήμιο Ιωαννίνων	<b>Ώρες Επικοινωνίας:</b>
	Τρίτη 9:00-14:00 Γραφείο Εκτάκτων
	ή στείλτε e-mail για άλλη ώρα

Εικόνα 188 - Στοιχεία επικοινωνίας

Index =>

**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

Quiz

---

Show all questions

1 / 30 =>

Έστω ο πίνακας R που περιέχει N1 εγγραφές και ο Πίνακας S που περιέχει N2 εγγραφές με  $N1 > N2$ . Ποιο είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος εγγραφών που θα προκύψει από την πράξη  $R \cap S$ ;

A.  N1, N1+N2

B.  N1-N2, N1

C.  N2, N1+N2

D.  0, N2

---

Index =>

Index =>

**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

Quiz

---

Show all questions

Your score is 66%.  
Questions completed so far: 1/30.

Correct!  
Your score is 66%.  
Questions completed so far: 1/30.

**OK**

Έστω ο πίνακας R που περιέχει N1 εγγραφές και ο Πίνακας S που περιέχει N2 εγγραφές με  $N1 > N2$ . Ποιο είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος εγγραφών που θα προκύψει από την πράξη  $R \cap S$ ;

A.  N1, N1+N2

B.  N1-N2, N1

C.  N2, N1+N2

D.  0, N2

---

Index =>

Index =>

**ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

Quiz

---

Show all questions

Your score is 66%.  
Questions completed so far: 1/30.

Correct!  
Your score is 66%.  
Questions completed so far: 1/30.

**OK**

Έστω ο πίνακας R που περιέχει N1 εγγραφές και ο Πίνακας S που περιέχει N2 εγγραφές με  $N1 > N2$ . Ποιο είναι το μικρότερο και ποιο το μεγαλύτερο πλήθος εγγραφών που θα προκύψει από την πράξη  $R \cap S$ ;

A.  N1, N1+N2

B.  N1-N2, N1

C.  N2, N1+N2

D.  0, N2

---

Index =>

**Εικόνα 189 - Σελίδα ερωτήσεων πολλαπλής επιλογής**

Για την μορφοποίηση της εμφάνισης της σελίδας των ερωτήσεων πολλαπλής επιλογής χρησιμοποιήθηκε κώδικας CSS.

## Βιβλιογραφία

### Βιβλία

Hank Korth, Avi Silberschatz, and S. Sudarshan, *Συστήματα Βάσεων Δεδομένων*, 4η Έκδοση, Εκδόσεις Μ. Γκιούρδας

ISBN 960-512-384-3

C. J. Date, *Εισαγωγή στα Συστήματα Βάσεων Δεδομένων*, (1<sup>ος</sup> κ 2<sup>ος</sup> τόμος), έκτη αμερικάνικη έκδοση 1995, (Εκδόσεις Κλειδάριθμος),

ISBN 960-332-109-5

Ramez Elmasri και Sham B. Navathe, *Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων* (1ος τόμος), 3η Έκδοση αναθεωρημένη, (Εκδόσεις ΔΙΑΥΛΟΣ),

ISBN 960-531-110-0

Ramez Elmasri και Sham B. Navathe, *Θεμελιώδεις Αρχές Συστημάτων Βάσεων Δεδομένων* (2ος τόμος), 3η Έκδοση αναθεωρημένη, (Εκδόσεις ΔΙΑΥΛΟΣ),

ISBN 960-531-119-4

Ξένος Μ. και Χριστοδουλάκης Δ., *Εισαγωγή στις Βάσεις Δεδομένων, Αναπτυξιακή Προσέγγιση*, Εκδόσεις ΠΑΠΑΣΩΤΗΡΙΟΥ

Βασιλακόπουλος Γ., *Σχεδιασμός Βάσεων Δεδομένων & dBase IV/SQL*,

Εκδόσεις Α. ΣΤΑΜΟΥΛΗΣ

R.M Riordan, *Προγραμματισμός στο MS SQL SERVER 2000 Βήμα Βήμα*, 2002, (Εκδόσεις Κλειδάριθμος),

R.Jacobson, *MS SQL SERVER 2000 Analysis Services Βήμα Βήμα*, 2002, (Εκδόσεις Κλειδάριθμος),

Jeff Ullman and Jennifer Widom, *A First Course in Database Systems*, Prentice Hall

Raghu Ramakrishnan, *Database Management Systems*, McGraw-Hill

Aho A. V., - Hopcroft J.E, - Ullman J.D, "*Data Structures and algorithms*", Addison-Wesley, 1983

Dusan Petkovic ,*Οδηγός Sql Server 2000*, Εκδόσεις Μ. Γκιούρδας

Ξένος, Χριστοδουλάκης, *Εισαγωγή στις Βάσεις Δεδομένων, αναπτυξιακή προσέγγιση*, Εκδόσεις Παπασωτηρίου, 2002

*Introduction to DataBase Systems and Client / Server Computing*, Harvard University

Riordan, *Προγραμματισμός στον MS SQL SERVER 2000 Βήμα προς Βήμα*, Εκδόσεις Κλειδάριθμος

Jacobson, *MS SQL SERVER 2000 analysis Services Βήμα Βήμα*, Εκδόσεις Κλειδάριθμος

Peter Rechenberg, *Εισαγωγή στον Πληροφορική*, Εκδόσεις Κλειδάριθμος

Jarrod Hollingworth, Bob Swart, Mark Cashman, Paul Gustavson, *Borland C++ Builder 6 Πλήρες Εγχειρίδιο*, Εκδόσεις Μ. Γκιούρδας, 1<sup>η</sup> Αμερικάνικη Έκδοση 2003

Sartaj Sahni, *Δομές Δεδομένων Αλγόριθμοι, και εφαρμογές στη C++*, Εκδόσεις ΤΖΙΟΛΑ

Stephen Prata, *Η βίβλος της C++ Primer Plus*, Εκδόσεις Μ. Γκιούρδας, 3<sup>η</sup> Αμερικάνικη Έκδοση

Scott Urman, *Oracle 9i Εγχειρίδιο Προγραμματισμού με PL/SQL*, Εκδόσεις Μ. Γκιούρδας

Rick Greenwald, Robert Stackowiak, Jonathan Stern, *Εισαγωγή στην Oracle*, Εκδόσεις Κλειδάριθμος

Michael J. Young, *Βήμα Βήμα XML*, Εκδόσεις Κλειδάριθμος

Margaret H. Dunham, *Data Mining Εισαγωγικά και Προηγμένα Θέματα Εξόρυξης Γνώσεις από Δεδομένα*, Εκδόσεις Νέων Τεχνολογιών

## Άρθρα

D. Knuth, *The Art of Computer Programming, vol. 3: Sorting and Searching*, Addison-Wesley, Reading, MA, 1973.

Raphael A. Finkel, Jon Louis Bentley: Quad Trees: A Data Structure for Retrieval on Composite Keys. *Acta Inf.* 4: 1-9(1974)

J.L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching", *Communications of the ACM*, vol. 18, pp. 509-517, 1975.

J.L. Bentley, and Stanat, D F. "Analysis of range searches in quad trees", *Inf Process Lett* 3, 6 (July 1975), 170-173

Hanan Samet: The Quad tree and Related Hierarchical Data Structures. *ACM Comput. Surv.* 16(2): 187-260(1984)

A. Guttman, "R-trees: a dynamic index structure for spatial searching", *Proceedings of ACM SIGMOD Conference on Management of Data*, June 1984.

[D. Comer, "The Ubiquitous B-Tree", *ACM Computing Surveys*, vol. 11(2), pp.121-137, June 1979.

T. Sellis, N. Roussopoulos, C. Faloutsos, "The R+-tree: a dynamic index for multidimensional objects", *Proceedings of the 13<sup>th</sup> International Conference on Very Large Data Bases (VLDB)*, September 1987.

N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger, "The R\*-tree: An Efficient and Robust Access Method for Points and Rectangles", *Proceedings of ACM SIGMOD Conference on Management of Data*, May 1990.

H. Samet, *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, Reading, MA, 1990.

I. Kamel, C. Faloutsos, "On Packing R-trees", *Proceedings of the 2nd International Conference on Information and Knowledge Management (CIKM)*, November 1993.

I. Kamel, C. Faloutsos, "Hilbert R-tree: An Improved R-tree Using Fractals", *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB)*, September 1994.

N. Roussopoulos, S. Kelley, F. Vincent, "Nearest Neighbor Queries", *Proceedings of ACM SIGMOD Conference on Management of Data*, May 1995.

## Ηλεκτρονικές Πηγές

---

<http://www.indyproject.org>

<http://www.microsoft.com/data>

<http://www.microsoft.com/data/oledb/>

<http://www.idealibrary.com/>

<http://us.imdb.com>

<http://www.film.com/admin/about.htm>

<http://www.elsevier.com/locate/infosys>

<http://www.nist.gov>

<http://www.oracle.com>

<http://www.rtreeportal.org>

<http://www.gamedev.net>

<http://en.wikipedia.org>

<http://dide.flo.sch.gr/Plinet/plinet.html>

<http://www.mysql.com/>

[http://www.intelligentedu.com/blogs/post/free\\_computer\\_books/380/free-books-for-databases-networking-and-operating-systems/el/](http://www.intelligentedu.com/blogs/post/free_computer_books/380/free-books-for-databases-networking-and-operating-systems/el/)