

# **1.ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ** **ΔΙΑΧΕΙΡΙΣΗΣ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ**

## **1.1 ΣΧΕΣΙΑΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ**

Σχεσιακές βάσεις δεδομένων ( relational databases) ονομάζονται οι βάσεις δεδομένων στις οποίες τα δεδομένα απεικονίζονται σε δισδιάστατους πίνακες (Tables), οι οποίοι αποτελούνται από γραμμές (Rows) και στήλες(Columns). Στη σχετική βιβλιογραφία οι όροι εγγραφές (records) και πεδία (fields) αντιστοιχούν στις γραμμές και στις στήλες αντίστοιχα, ενώ ένας πίνακας αντιστοιχεί μ' ένα αρχείο σε μη σχεσιακές βάσεις

Υπάρχουν πολλά συστήματα που διαχειρίζονται σχεσιακές βάσεις δεδομένων όπως Access,SQL Server,Btrieve,Dbase,Paradox,MimerSQL,MiniSQL,Oracle,Mysql και άλλα.

Τα παραπάνω συστήματα βάσεων δεδομένων χωρίζονται σε τρεις κατηγορίες οι οποίες είναι:

- Οι βάσεις δεδομένων που είναι αρχεία και έχουν την ίδια μορφή με αυτά που χρησιμοποιεί η Access, που τα δημιουργεί και διαχειρίζεται άμεσα η μηχανή Jet παρέχοντας έτσι μέγιστη ευελιξία και ταχύτητα.
- Οι εξωτερικές βάσεις δεδομένων οι οποίες είναι ISAM (Indexed Sequential Access Method) βάσεις δεδομένων που συμπεριλαμβάνουν πολλές γνωστές μορφές όπως Btrieve, dBASE III, dBASE IV, Microsoft FoxPro 2.0 και 2.5, και Paradox 3.X και 4.0
- Οι βάσεις δεδομένων ODBC που είναι βάσεις δεδομένων client-server που υποστηρίζουν το πρότυπο ODBC, όπως ο SQL Server. ΤοODBCDirect mode του DAO επιτρέπει να χρησιμοποιηθούν δεδομένα ODBC χρησιμοποιώντας τα ίδια αντικείμενα , υπερπηδώντας όμως τη γεννήτρα Jet.

Ο ανταγωνισμός των εταιριών που παράγουν και προωθούν αυτά τα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων είναι πολύ μεγάλος ,για το ποιο τελικά από αυτά θα επικρατήσει στην αγορά.Οι παράγοντες που τον επηρεάζουν είναι διάφοροι όπως:

- Ποιο σύστημα παρέχει την μεγαλύτερη σταθερότητα στην αποθήκευση των δεδομένων.Η ποιο σημαντική απαίτηση ενός τέτοιου συστήματος είναι η σωστή αποθήκευση των δεδομένων χωρίς να υπάρχει πιθανότητα αλλοίωσης τους ή περίπτωση αυτά να μην αποθηκευτούν καθόλου.
- Ποιο σύστημα παρέχει το καλύτερο interface για τον χρήστη με σκοπό την γρήγορη και εύκολη αποθήκευση των δεδομένων
- Ποιο σύστημα παρέχει τη δυνατότητα ανταλλαγής πληροφοριών και σε δικτυακό επίπεδο
- Ποιο σύστημα επιτρέπει την ανταλλαγή πληροφοριών και δεδομένων με άλλα ξένα συστήματα μέσω διαφόρων μεθόδων όπως import και export.

## **1.2 ΜΟΝΤΕΛΟ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ**

Όλα τα συστήματα σχεσιακών βάσεων δεδομένων,ακολουθούν την ίδια θεωρία, **την θεωρία των σχεσιακών βάσεων δεδομένων.**

Το σχεσιακό μοντέλο αναπτύχθηκε με το σκεπτικό της αποθήκευσης των δεδομένων σε πολλαπλούς πίνακες. Οι πληροφορίες είναι χωρισμένες σε ενότητες οι οποίες συνδέονται μεταξύ τους με σχέσεις που συντηρούνται και ελέγχονται από το σύστημα. Κάθε πίνακας περιέχει πληροφορίες για μία μοναδική ενότητα π.χ. αποθήκη, πελάτες, υπάλληλοι κ.λ.π.

Δεν πρέπει να επιτρέπονται διπλές εγγραφές σύμφωνα με την θεωρία των σχεσιακών βάσεων. Ας υποθέσουμε τώρα ότι σε ένα αρχείο υπαλλήλων υπάρχουν δύο πρόσωπα πατέρα και υιός που μοιράζονται το ίδιο όνομα, διεύθυνση, τηλέφωνο. Είναι μια πιθανή περίπτωση που δημιουργεί την ανάγκη να ξεχωρίζουμε με μοναδικό τρόπο τις εγγραφές μεταξύ τους.

Η μοναδικότητα των εγγραφών επιτυγχάνεται με τη χρήση πεδίων που ορίζονται ως Κλειδιά (keys). Αν ένα πεδίο το ορίσουμε ως πρωταρχικό και βασικό Κλειδί (primary key) τότε το ίδιο το σύστημα μας προφυλάσσει από την εισαγωγή διπλών εγγραφών. Ένα primary key ορίζεται αυτόματα ως δείκτης που δεν επιτρέπει διπλές εγγραφές και παράλληλα τις ταξινομεί με την σειρά τους κατά την εμφάνιση του πίνακα.

Συνήθως το primary key ενός αρχείου πρέπει να αναφέρεται σε όποιο νούμερο ή όνομα που δεν θα χρειάζεται να αλλάξει με το πέρασμα του χρόνου. Τα Κλειδιά που περιέχουν νούμερα είναι πιο εύκολα στη διαχείριση τους από τον υπολογιστή, κρατούν λιγότερα bytes κατά την αποθήκευση και δεν δημιουργούν προβλήματα στην ταξινόμηση των δεδομένων.

Όταν ορίζουμε σχέσεις ανάμεσα στα πεδία πρέπει πάντα να φροντίζουμε να έχουν τον ίδιο τύπο, για παράδειγμα αν ένα primary key είναι αλφαριθμητικό τριών θέσεων τότε το foreign key στον πίνακα πρέπει να είναι του ιδίου τύπου τριών θέσεων. Αν κατά την δημιουργία ενός πίνακα δεν ορίσουμε primary key, τότε το σύστημα μας το υπενθυμίζει και προτείνει την δημιουργία του. Εδώ πρέπει να σημειώσουμε ότι δεν επιτρέπεται σε πεδία που έχουν ορισθεί σαν κλειδιά να έχουν μηδενικές τιμές (Null values), αντίθετα πρέπει να είναι πάντα συμπληρωμένα.

## **1.3 ΠΡΟΒΛΗΜΑΤΑ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ**

Παρά το γεγονός όμως ότι αυτά τα συστήματα ακολουθούν το ίδιο σχεσιακό μοντέλο, η επικοινωνία και ανταλλαγή δεδομένων μεταξύ τους καθίσταται πολλές φορές σχεδόν αδύνατη ή καταστρεπτική για τα δεδομένα. Η εισαγωγή και η εξαγωγή πινάκων δεδομένων, αν και υποστηρίζεται από ορισμένα σύστημα διαχείρισης σχεσιακών βάσεων δεν είναι πάντα επιτυχής και ο κύριος λόγος είναι η ασυμφωνία των τύπων δεδομένων μεταξύ των διαφορετικών συστημάτων.

Κάποια από αυτά υποστηρίζουν τύπους για δυαδικά δεδομένα όπως Blob,OLE και Binary, ενώ κάποια άλλα όχι. Επίσης για τύπους ημερομηνίας και χρόνου ,το format που υποστηρίζεται διαφέρει από σύστημα σε σύστημα με

αποτέλεσμα κάποια τιμή ημερομηνίας που είναι δεχτή σε κάποιο, να μην είναι δεχτή σε κάποιο άλλο. Μία καλή λύση για ένα τέτοιο πρόβλημα θα ήταν η αντιστοίχιση των τύπων δεδομένων όλων των συστημάτων με ένα πρότυπο σύνολο τύπων δεδομένων.

Ένα άλλο πρόβλημα που εμφανίζεται στις σχεσιακές βάσεις δεδομένων, είναι η πολύ μεγάλη χωρητικότητα πολλές φορές των πινάκων των δεδομένων, με αποτέλεσμα να είναι πολύ αργή ή και σχεδόν αδύνατη η μεταφορά τους μέσω του διαδικτύου. Σε κάποια πάλι συστήματα, μία βάση ή ένας πίνακας αποτελείται από περισσότερα του ενός αρχεία τα οποία πολλές φορές ίσως να μην μπορεί να ξεχωρίσει ο χρήστης και αυτό συνεπάγεται ακόμη μεγαλύτερη δυσκολία.

Σε τέτοιου είδους προβλήματα έρχεται να δώσει τη λύση η XML.

## 2. XML

### 2.1 Η ΣΗΜΕΙΑΚΗ ΓΛΩΣΣΑ XML

Η XML (eXtensible Markup Language) δεν είναι μία σημειακή γλώσσα όπως η HTML, είναι μία γλώσσα που χρησιμοποιείται για την περιγραφή μίας σημειακής γλώσσας. Ο τεχνικός όρος μιας τέτοιας γλώσσας είναι μετα-γλώσσα.

Η XML προήλθε λόγω μιας σειράς προβλημάτων της HTML:

- Ήταν σχεδόν αδύνατο να διακριθεί οποιαδήποτε σημαντική πληροφορία μέσα σε μια σελίδα HTML.
- Το γεγονός ότι ο ίδιος κώδικας HTML εμφανιζόταν με διάφορες μορφές, ανάλογα με τον browser που χρησιμοποιούνταν.

Η XML, όπως και η Hyper-Text Markup Language (HTML), έχει τις ρίζες της στην διαχείριση εγγράφων και παράγεται από την Standard Generalized Markup Language (SGML). Ωστόσο, αντίθετα από την SGML και την HTML, η XML μπορεί να αναπαραστήσει δεδομένα βάσεων δεδομένων, όπως επίσης πολλά άλλα είδη δομημένων δεδομένων που χρησιμοποιούνται σε επαγγελματικές εφαρμογές. Είναι πολύ χρήσιμη για την ανταλλαγή μορφοποιημένων δεδομένων ως XML αρχεία όταν μια εφαρμογή πρέπει να επικοινωνεί με μια άλλη εφαρμογή, ή για την ενοποίηση πληροφοριών από διάφορες εφαρμογές.

Ο όρος σήμανση (markup) αναφέρεται σε οτιδήποτε υπάρχει σε ένα έγγραφο που δεν πρόκειται να τυπωθεί. Όπως τα συστήματα βάσεων δεδομένων αναπτύχθηκαν από την φυσική επεξεργασία των αρχείων για να παρέχουν μια διαφορετική λογική προβολή, έτσι και οι γλώσσες σήμανσης αναπτύχθηκαν για να ορίζουν εντολές για τον τρόπο που θα τυπώνονται μέρη ενός εγγράφου, για να καθορίζεται η λειτουργία των περιεχομένων. Για παράδειγμα, με την λειτουργική σήμανση, το κείμενο που αντιπροσωπεύει επικεφαλίδες ενοτήτων (για αυτή την ενότητα η φράση «Η γλώσσα XML»), θα πρέπει να σημειώνονται ότι είναι επικεφαλίδες ενότητας.

Για την οικογένεια των γλωσσών σήμανσης που περιλαμβάνουν τις HTML, SGML και XML, η σήμανση παίρνει την μορφή **ετικετών (tag)** που περικλείονται σε

γωνιώδεις αγκύλες, <>. Οι ετικέτες χρησιμοποιούνται σε ζευγάρια, με την <tag> και την </tag> για να χωρίζουν την αρχή και το τέλος του τμήματος του εγγράφου στο οποίο αναφέρεται η ετικέτα. Για παράδειγμα, ο τίτλος:  
<title>ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ</title>

Αντίθετα με την HTML, η XML δεν περιγράφει το σύνολο των ετικετών που επιτρέπονται και το σύνολο μπορεί να εξειδικεύεται ανάλογα με την ανάγκη. Αυτή η λειτουργία είναι το κλειδί για το βασικό ρόλο της XML στην αναπαράσταση και ανταλλαγή δεδομένων, όπου η HTML χρησιμοποιείται βασικά για μορφοποίηση εγγράφων

## 2.2 ΔΟΜΗ ΤΩΝ XML ΔΕΔΟΜΕΝΩΝ

Η βασική δομή σε ένα XML έγγραφο είναι το **στοιχείο (element)**. Ένα στοιχείο είναι ένα ζεύγος από ετικέτες αρχής και τέλους και όλο το κείμενο μεταξύ τους. Πρέπει επίσης να υπάρχει ένα στοιχείο **ρίζας** που περιέχει όλα τα άλλα στοιχεία.

Για παράδειγμα, στην εφαρμογή μιας τράπεζας (bank), όπου οι πληροφορίες του πίνακα λογαριασμών (account) μπορούν να αντιπροσωπεύονται από στοιχεία ενός XML εγγράφου, το στοιχείο **<bank>** σχηματίζει το στοιχείο ρίζας όπως φαίνεται στην παρακάτω εικόνα

```
<bank>
  <account>
    <account_number> A101 </account_number>
    <branch_name> Belizelou </branch_name>
    < balance > 500 </balance >
  </account>
  <account >
    <account_number> A102 </account_number >
    <branch_name> Perryridge </branch_name>
    < balance> 400 </balance>
  </account>
  <account>
    <account_number> A201 </account_number>
    <branch_name> Brighton </branch_name>
    < balance > 900 </balance >
  </account>
</bank>
```

Εκτός από στοιχεία η XML καθορίζει την έννοια και μιας **ιδιότητας (attribute)**. Οι ιδιότητες ενός στοιχείου εμφανίζονται σε **ζεύγη ονόματος-τιμής** πριν από το ">" μιας ετικέτας. Για παράδειγμα ο τύπος ενός λογαριασμού (αν είναι όψεως ή ταμιευτηρίου) μπορεί να αντιπροσωπεύεται από μία ιδιότητα όπως παρακάτω

```
<bank>
  <account account-type="tamieutiriou">
    <account_number> A101 </account_number>
    <branch_name> Belizelou </branch_name>
```

```
< balance > 500 </balance >
</account>
<bank>
```

Οι ιδιότητες είναι συμβολοσειρές και δεν περιέχουν σήμανση. Επιπλέον οι ιδιότητες μπορεί να εμφανίζονται μόνο μία φορά σε μία ετικέτα, αντίθετα από τα υποστοιχεία που μπορεί να επαναλαμβάνονται. Επίσης μία ιδιότητα είναι έμμεσο κείμενο που δεν φαίνεται στο έγγραφο που τυπώνεται ή εμφανίζεται. Επιπλέον ένα στοιχείο της μορφής <element></element>, που δεν περιέχει κείμενο ή άλλα υποστοιχεία, αλλά μόνο ιδιότητες, μπορεί να συντομευτεί ως <element/>.

Αφού τα XML έγγραφα έχουν σχεδιαστεί να ανταλλάσσονται μεταξύ εφαρμογών, έχει παρουσιαστεί ένας μηχανισμός **χώρου ονομάτων (namespace – ns)** που επιτρέπει σε οργανισμούς να καθορίζουν καθολικά μοναδικά ονόματα ως ετικέτες στοιχείων σε έγγραφα. Η ιδέα του χώρου ονομάτων είναι να προτάσσει σε κάθε ετικέτα ή ιδιότητα ένα καθολικό αναγνωριστικό πόρων (για παράδειγμα μία Web διεύθυνση). Ένας προκαθορισμένος **χώρος ονομάτων** μπορεί να οριστεί χρησιμοποιώντας την ιδιότητα **xmlns** στο στοιχείο ρίζας.

Σε σύγκριση με την αποθήκευση των δεδομένων σε μια βάση δεδομένων, η αναπαράσταση XML μπορεί να είναι αναποτελεσματική, αφού τα ονόματα ετικετών επαναλαμβάνονται σε όλο το έγγραφο. Ωστόσο, ανεξάρτητα από αυτό το μειονέκτημα, μια αναπαράσταση XML έχει πολλά πλεονεκτήματα όταν χρησιμοποιείται για ανταλλαγή δεδομένων, επειδή:

- Πρώτον, η παρουσία των ετικετών κάνει το αρχείο να τεκμηριώνεται μόνο του, δηλαδή, δεν χρειάζεται να συμβουλευτεί κανείς ένα σχήμα για να καταλάβει τη σημασία του
- Δεύτερον, η μορφή του εγγράφου δεν είναι αυστηρή. Για παράδειγμα, αν κάποιος αποστολέας προσθέσει επιπλέον πληροφορίες, μέσα σε μία ετικέτα, ο παραλήπτης των δεδομένων XML μπορεί απλώς να αγνοήσει την ετικέτα.
- Τέλος, η μορφή XML χρησιμοποιείται ευρύτατα.

Όπως η SQL είναι η κυρίαρχη γλώσσα για ερωτήματα σχεσιακών δεδομένων, έτσι και η XML γίνεται η βασική μορφή για ανταλλαγή δεδομένων.

## **2.3 ΣΧΗΜΑ ΕΝΟΣ ΕΓΓΡΑΦΟΥ XML**

Οι βάσεις δεδομένων έχουν σχήματα (schema), που χρησιμοποιούνται για να περιγράψουν τις πληροφορίες που μπορούν να αποθηκευθούν στην βάση δεδομένων και να περιορίσουν τους τύπους των δεδομένων των αποθηκευμένων πληροφοριών. Αντίθετα, εξ ορισμού, τα XML έγγραφα μπορούν να δημιουργηθούν χωρίς κάποιο σχετικό σχήμα. Ένα στοιχείο μπορεί μετά να έχει οποιοδήποτε υποστοιχείο ή ιδιότητα.

Ο ορισμός τύπου εγγράφου (**document type definition - DTD**), είναι ένα προαιρετικό μέρος ενός XML εγγράφου. Ο κύριος σκοπός του DTD είναι όπως αυτό του σχήματος: να περιορίζει τον τύπο των πληροφοριών που παρουσιάζονται στο

έγγραφο. Ωστόσο, το DTD δεν περιορίζει τους τύπους, σε σχέση με τους βασικούς τύπους δεδομένων, όπως ακέραιους ή συμβολοσειρές. Αντίθετα, περιορίζει μόνο την εμφάνιση υποστοιχείων και ιδιοτήτων μέσα σε ένα στοιχείο.

Έτσι, στο DTD της παρακάτω εικόνας, ένα στοιχείο τράπεζας αποτελείται από έναν ή περισσότερους λογαριασμούς. Ο τελεστής | καθορίζει το "ή", ενώ ο τελεστής + καθορίζει το "ένα ή περισσότερα". Αν και δεν φαίνεται εδώ, ο τελεστής \* χρησιμοποιείται για να καθορίζει "μηδέν ή περισσότερα", ενώ ο τελεστής ? χρησιμοποιείται για να ορίσει ένα προαιρετικό στοιχείο (δηλαδή "μηδέν ή ένα").

```
<!DOCTYPE bank [  
  <!ELEMENT bank ( (account)+)>  
  <!ELEMENT account ( account_number branch_name balance )>  
  <!ELEMENT account_number ( #PCDATA )>  
  <!ELEMENT branch_name ( #PCDATA )>  
  <!ELEMENT balance ( #PCDATA )>  
>
```

Το στοιχείο **account** ορίζεται να περιέχει στο σχήμα του τα υποστοιχεία **account\_number**, **branch\_name**, και **balance** με αυτή τη σειρά. Τα στοιχεία **account\_number**, **branch\_name**, και **balance** δηλώνονται όλα **#PCDATA** που σημαίνει δεδομένα κειμένου. Το όνομα της λέξης προέρχεται από το "parsed character data" δηλαδή δεδομένα χαρακτήρων που έχουν αναλυθεί.

Υπάρχουν μερικοί περιορισμοί των DTD ως μηχανισμοί σχήματος. Τα περισσότερα μεμονωμένα στοιχεία κειμένου και οι ιδιότητες δεν μπορούν να έχουν τύπο. Για παράδειγμα, το στοιχείο **balance** δεν μπορεί να περιοριστεί να είναι ένας θετικός αριθμός. Είναι επίσης δύσκολο να χρησιμοποιήσουμε το μηχανισμό DTD για να καθορίσουμε μη διατεταγμένα σύνολα από υποστοιχεία. Η διάταξη είναι πάντα σημαντική για ανταλλαγή δεδομένων.

Μια προσπάθεια να αποκατασταθούν πολλά από τα ελαττώματα του DTD κατέληξε σε μια πιο εξειδικευμένη γλώσσα σχήματος, το **XMLSchema**

## 2.4 ΣΧΗΜΑ XML

Η παρακάτω εικόνα δείχνει πώς το DTD της προηγούμενης εικόνας μπορεί να αναπαρασταθεί από το XMLSchema.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">  
<xsd:element name="bank" type="BankType" />  
<xsd:element name="account">  
  <xsd:complexType>  
    <xsd:sequence>  
      <xsd:element name="account_number" type="xsd:string"/>  
      <xsd:element name="branch_name" type="xsd:string"/>  
      <xsd:element name="balance" type="xsd:decimal"/>  
    </xsd:sequence>  
  </xsd:complexType>  
</xsd:element>  
</xsd:schema>
```

```

        </xsd:complexType>
</xsd:element>
<xsd:complexType name="BankType" >
    <xsd:sequence>
        <xsd:element          ref="account"          minOccurs="0"
maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

Το πρώτο στοιχείο είναι το στοιχείο ρίζα **bank**, του οποίου ο τύπος δηλώνεται έπειτα. Το παράδειγμα μετά ορίζει τον τύπο του στοιχείου **account**. Παρατηρήστε την χρήση των τύπων `xsd:string` και `xsd:decimal` για περιορισμό των τύπων των στοιχείων δεδομένων. Τέλος, το παράδειγμα ορίζει τον τύπο `BankType` ότι περιέχει μηδέν ή περισσότερες επαναλήψεις του `account`. Το `XMLSchema` μετά ορίζει τον ελάχιστο και μέγιστο αριθμό επαναλήψεων υποστοιχείων χρησιμοποιώντας τα `minOccurs` και `maxOccurs`. Το προκαθορισμένο για ελάχιστες και μέγιστες επαναλήψεις είναι το 1, έτσι αυτά θα πρέπει άμεσα να καθοριστούν για να επιτρέπουν μηδέν ή περισσότερους λογαριασμούς.

## 2.5 XML PERSISTENCE FORMAT

Παρακάτω παρουσιάζεται ένα παράδειγμα ενός αρχείου XML, το οποίο αναπαριστά ένα πίνακα σε μία βάση δεδομένων. Ο τύπος κωδικοποίησης χαρακτήρων είναι "UTF-8".

```

_ <xml xmlns:s="uuid:BDC6E3F0-6DA3-11d1-A2A3-00AA00C14882"
xmlns:dt="uuid:C2F41010-65B3-11d1-A29F-00AA00C14882"
xmlns:rs="urn:schemas-microsoft-com:rowset"
xmlns:z="#RowsetSchema">
_ <s:Schema id="RowsetSchema">
_ <s:ElementType name="row" content="eltOnly">
_ <s:AttributeType name="CategoryID" rs:number="1" rs:nullable="true"
rs:maydefer="true" rs:writeunknown="true">
    <s:datatype dt:type="float" dt:maxLength="8" rs:precision="15"
rs:fixedlength="true" />
    </s:AttributeType>
_ <s:AttributeType name="CategoryName" rs:number="2" rs:nullable="true"
rs:maydefer="true" rs:writeunknown="true">
    <s:datatype dt:type="string" dt:maxLength="254" />
    </s:AttributeType>
_ <s:AttributeType name="Description" rs:number="3" rs:nullable="true"
rs:maydefer="true" rs:writeunknown="true">
    <s:datatype dt:type="string" dt:maxLength="254" />
    </s:AttributeType>
    <s:extends type="rs:rowbase" />
    </s:ElementType>
    </s:Schema>
_ <rs:data>

```

```

<z:row CategoryID="1" CategoryName="Beverages" Description="Soft
drinks, coffees, teas, beers, and ales" />
<z:row CategoryID="2" CategoryName="Condiments" Description="Sweet
and savory sauces, relishes, spreads, and seasonings" />
<z:row CategoryID="3" CategoryName="Confections"
Description="Desserts, candies, and sweet breads" />
<z:row CategoryID="4" CategoryName="Dairy Products"
Description="Cheeses" />
<z:row CategoryID="5" CategoryName="Grains/Cereals"
Description="Breads, crackers, pasta, and cereal" />
<z:row CategoryID="6" CategoryName="Meat/Poultry"
Description="Prepared meats" />
<z:row CategoryID="7" CategoryName="Produce" Description="Dried fruit
and bean curd" />
<z:row CategoryID="8" CategoryName="Seafood" Description="Seaweed
and fish" />
</rs:data>
</xml>

```

Ο πίνακας που περιγράφεται είναι ο πίνακας “Customers” της βάσεως δεδομένων “Northwind” η οποία εισάγεται με την εγκατάσταση του SQL Server στο σύστημα του υπολογιστή. Το XML αρχείο χωρίζεται σε δύο μέρη εκ των οποίων στο πρώτο είναι το σχήμα (η δομή του πίνακα) και το δεύτερο τα δεδομένα (το σύνολο των εγγραφών) του πίνακα. Το παραπάνω αρχείο είναι ένα αρχείο XML τύπου Attribute, περιέχει δηλαδή και τη δομή και τα δεδομένα του πίνακα.

Η XML persistence μορφή στο ADO χρησιμοποιεί τα τέσσερα παρακάτω namespaces:

Πρόθεμα	Περιγραφή
s	Αναφέρεται στο "xml-namespace" namespace περιλαμβάνοντας τα στοιχεία και τις ιδιότητες που καθορίζουν το σχήμα του τρέχοντος Recordset
dt	Αναφέρεται στον ορισμό των τύπων δεδομένων
rs	Αναφέρεται στο namespace που περιλαμβάνει τα data και τις ιδιότητες για τις ιδιότητες του ADO Recordset
z	Αναφέρεται στο σχήμα του τρέχοντος rowset (εγγραφής).

Να σημειωθεί επίσης ότι το πρόθεμα του namespace μπορεί να το αλλάξει ο χρήστης φτάνει να το χρησιμοποιήσει έτσι από εκεί και πέρα.

Στην περίπτωση τώρα που το αρχείο xml περιείχε μόνο τις εγγραφές και τα ονόματα των πεδίων, χωρίς να αναλύεται η δομή του πίνακα, το αρχείο θα ήταν τύπου Element. Παρακάτω παρουσιάζεται ο ίδιος πίνακας σαν Element-centric XML.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <rootElement xmlns:z="#RowsetSchema">
- <categories>
  <CategoryID>1</CategoryID>
  <CategoryName>Beverages</CategoryName>

```



```

    <Description>Soft drinks, coffees, teas, beers, and ales</Description>
  </categories>
- <categories>
  <CategoryID>2</CategoryID>
  <CategoryName>Condiments</CategoryName>
  <Description>Sweet and savory sauces, relishes, spreads, and
seasonings</Description>
</categories>
- <categories>
  <CategoryID>3</CategoryID>
  <CategoryName>Confections</CategoryName>
  <Description>Desserts, candies, and sweet breads</Description>
</categories>
- <categories>
  <CategoryID>4</CategoryID>
  <CategoryName>Dairy Products</CategoryName>
  <Description>Cheeses</Description>
</categories>
- <categories>
  <CategoryID>5</CategoryID>
  <CategoryName>Grains/Cereals</CategoryName>
  <Description>Breads, crackers, pasta, and cereal</Description>
</categories>
- <categories>
  <CategoryID>6</CategoryID>
  <CategoryName>Meat/Poultry</CategoryName>
  <Description>Prepared meats</Description>
</categories>
- <categories>
  <CategoryID>7</CategoryID>
  <CategoryName>Produce</CategoryName>
  <Description>Dried fruit and bean curd</Description>
</categories>
- <categories>
  <CategoryID>8</CategoryID>
  <CategoryName>Seafood</CategoryName>
  <Description>Seaweed and fish</Description>
</categories>
</rootElement>

```

Ένα αρχείο XML, άσχετα με τον τύπο του, είναι ένα αρχείο κειμένου το οποίο καταλαμβάνει σίγουρα πολύ μικρότερη χωρητικότητα από τη χωρητικότητα του ίδιου του πίνακα ή της βάσης στην οποία ανήκει αυτός, ανάλογα βέβαια με το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που χρησιμοποιεί ο χρήστης. Επομένως η μεταφορά των δεδομένων μέσω ενός αρχείου XML δικτυακά (ακόμη και μεταξύ διαφορετικών συστημάτων βάσεων), μπορεί να γίνει πολύ εύκολα και αρκετά γρήγορα.

Ακόμη, λόγω της φύσης ενός αρχείου XML (το οποίο είναι ένα αρχείο κειμένου) ο μετασχηματισμός αυτού καθίσταται επίσης εύκολος και τρομερά χρήσιμος για τους χρήστες των βάσεων δεδομένων. Ο μετασχηματισμός αυτός μπορεί να είναι δύο ειδών, «μορφής» ή «δεδομένων». Ο μετασχηματισμός μορφής

έχει να κάνει με τη μορφοποίηση του κειμένου του αρχείου , όπως τη γραμματοσειρά του κειμένου αυτού ή το χρώμα π.χ. με το οποίο εμφανίζονται οι εγγραφές (στα παραπάνω αρχεία αυτό είναι το μαύρο, ενώ η γραμματοσειρά που χρησιμοποιείται είναι η Arial). Ο μετασχηματισμός δεδομένων έχει να κάνει με τη δομή του κειμένου του αρχείου XML. Ένα χαρακτηριστικό παράδειγμα τέτοιου μετασχηματισμού είναι ο μετασχηματισμός ενός Attribute-centric XML αρχείου σε Element-centric XML.

Η χρησιμοποίηση της γλώσσας και των αρχείων XML όπως εύκολα μπορεί να καταλάβει κανείς, είναι ένα τρομερό εργαλείο για τους χρήστες βάσεων δεδομένων και γενικότερα για την ανάπτυξη εφαρμογών. Όλο και περισσότερα προγράμματα, στις νέες εκδόσεις τους υποστηρίζουν αρχεία τέτοιου τύπου και διαθέτουν μηχανές εξαγωγής των δεδομένων που αυτά επεξεργάζονται σε XML μορφή.

### **ΑΝΑΦΟΡΕΣ-ΒΙΒΛΙΟΓΡΑΦΙΑ**

- ✚ [Kanne 2000] C.C. Kanne and G. Moerkotte, “Efficient Storage of XML Data”, Proc. Of the International Conf on Data Engineering (2000-) page 198.
- ✚ [Draper 2001] D. Draper, A. Y. Haley and D.S. Weld “The Nimble XML Data Integration system”, Proc. Of the International Conf on Data Engineering (2001) pages 155-160.
- ✚ [Baru 1999] C. K. Baru, A. Gypta, B. Ludascher, R. Marciano, Y. Papakonstantinou, P. Velikhov, and V. Chu, “XML-Based Information Mediation with MIX”, Proc of the ACM SIGMOD Conf. On Management of Data (1999) pages 597-599.
- ✚ [www.w3.org/TR/1998/NOTE-XML-data](http://www.w3.org/TR/1998/NOTE-XML-data)
- ✚ MSDN-Visual Basic 6.0

## **3. ΣΚΟΠΟΣ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ**

### **3.1 ΑΝΑΠΤΥΞΗ ΚΑΙ ΑΝΑΛΥΣΗ ΘΕΜΑΤΟΣ ΠΤΥΧΙΑΚΗΣ**

Στα πλαίσια της πτυχιακής γίνεται μια προσπάθεια ανάπτυξης ενός τέτοιου λογισμικού, που θα σχετίζεται με τα συστήματα σχεσιακών βάσεων δεδομένων που αναφέρθηκαν παραπάνω και θα έχει σαν σκοπό την ανταλλαγή δεδομένων μεταξύ τους μέσω εισαγωγής και εξαγωγής αρχείων XML, από και προς τα συστήματα αυτά.

Πιο συγκεκριμένα , το θέμα της πτυχιακής εργασίας που παρουσιάζεται και αναλύεται είναι :

**“Ανάπτυξη λογισμικού ανταλλαγής δεδομένων μέσω μετατροπής σε XML, από και προς αρχεία σχεσιακών βάσεων δεδομένων τύπου Paradox, dBase, MS Access, σχεσιακές βάσεις client/server τύπου SQL Server, Oracle, MySQL, MimerSQL, ενσωματωμένες και εικονικές βάσεις τύπου SQLite”.**

Κατά το πρώτο βήμα της εργασίας έγινε η εγκατάσταση των συστημάτων βάσεων δεδομένων με αντιπροσωπευτικά δείγματα βάσεων. Στο δεύτερο βήμα δοκιμάστηκαν οι γνωστότερες μέθοδοι σύνδεσης (ODBC, OLE DB) στο κάθε

σύστημα βάσης δεδομένων. Στο τρίτο βήμα έγινε η μελέτη μεθόδων πρόσβασης στα μετά-δεδομένα του καταλόγου της κάθε βάσης δεδομένων. Στο τέταρτο βήμα υλοποιήθηκαν οι βέλτιστοι μέθοδοι ανάκτησης (DAO, ADO κλπ) των δεδομένων της κάθε βάσης σε σύνολα εγγραφών. Στο πέμπτο βήμα έγινε η μετατροπή των συνόλων στο πρότυπο XML (element ή attribute type). Στο έκτο βήμα τέλος έγινε η εισαγωγή των αρχείων XML στη βάση επιλογής του χρήστη.

Η εφαρμογή παρέχει το κατάλληλο περιβάλλον, για την εισαγωγή των παραμέτρων σύνδεσης, την επιλογή των server και των βάσεων, την επιλογή των πινάκων και την επιλογή των αρχείων XML. Επίσης παρέχεται η δυνατότητα ενοποίησης δύο αρχείων από διαφορετικές βάσεις σε καθολικό σύνολο εγγραφών, καθώς και περιβάλλον εμφάνισης των εγγραφών συγκεκριμένου πίνακα που επιλέγει ο χρήστης .

### **3.2 ΑΝΑΛΥΣΗ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΑΝΑΠΤΥΞΗΣ ΠΤΥΧΙΚΗΣ**

Το λογισμικό αναπτύχθηκε στο γραφικό περιβάλλον της **Visual Basic 6.0**. Ο λόγος που χρησιμοποιήθηκε η συγκεκριμένη γλώσσα προγραμματισμού είναι η δυνατότητα της να χειρίζεται και να επεξεργάζεται συστήματα σχεσιακών βάσεων δεδομένων μέσω μιας σειράς μηχανών όπως το DAO ( Data Access Objects ) το οποίο είναι ένα προγραμματιστικό περιβάλλον με πλήρη έλεγχο των σχεσιακών βάσεων. Επίσης η μεγάλη συνεισφορά της **Visual Basic 6.0** είναι το νέο μοντέλο πρόσβασης δεδομένων ADO ( ActiveX Data Objects ) που είναι ειδικά σχεδιασμένο για να εξυπηρετεί επαρκέστατα τον προγραμματισμό στο Internet.

Κατά την ανάπτυξη της εφαρμογής ,σε συνδυασμό με το ADO και το DAO χρησιμοποιήθηκαν οι παροχές OLE DB και ODBC για την σύνδεση του λογισμικού με τα συστήματα των σχεσιακών βάσεων .

Το τελικό αποτέλεσμα της εφαρμογής είναι ένα εκτελέσιμο αρχείο “.exe” με το όνομα “**Perfect XML**” και το μέγεθος αυτής αρκείται σε μόλις λίγα **Kbytes**. Η λειτουργία και η εκτέλεση του υποστηρίζεται μόνο από λειτουργικό σύστημα Windows. Επίσης είναι απαραίτητοι οι **MyOLEDB v3.0** και **MySQL ODBC v3.51 Providers** (ή οποιεσδήποτε νεότερες εκδόσεις τους) οι οποίοι δεν παρέχονται μαζί με το εκτελέσιμο αρχείο ,εξαιτίας του κόστους αυτών και της μη δωρεάν διάθεσής τους από τις εταιρίες.

Να αναφερθεί ότι το πρόγραμμα κατά την εισαγωγή των αρχείων XML στις βάσεις , δεν υποστηρίζει τύπους δεδομένων δυαδικών αρχείων όπως BLOBS, OLE Objects και Binary Data. Ο λόγος αυτής της παράλειψης είναι ευνόητος αφού:

- Δεν υποστηρίζουν όλα τα συστήματα διαχείρισης σχεσιακών βάσεων τους παραπάνω τύπους δεδομένων
- Τέτοιοι τύποι δεδομένων αυξάνουν υπερβολικά το μέγεθος του αρχείου και κατά την εισαγωγή του στη βάση πρέπει να γίνει αποδιαμόρφωση των δυαδικών δεδομένων και επανάκτηση των OLE ή Binary αντικειμένων στην αρχική τους μορφή.

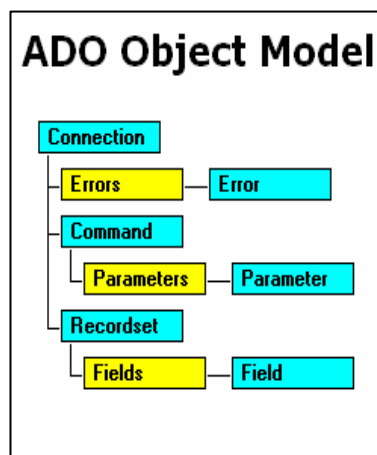
Ακολουθεί περιγραφή και ανάλυση των μηχανών ADO,DAO και των παροχών OLE DB και ODBC που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής από τη Visual Basic.

Επίσης αναλύονται οι έννοιες σχήματος και καταλόγου σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων και το πως μπορεί ο χρήστης να αντλήσει πληροφορίες μέσα απ' αυτές.

## 4. ΜΕΤΑΦΟΡΑ ΚΑΙ ΑΝΤΑΛΛΑΓΗ ΔΕΔΟΜΕΝΩΝ

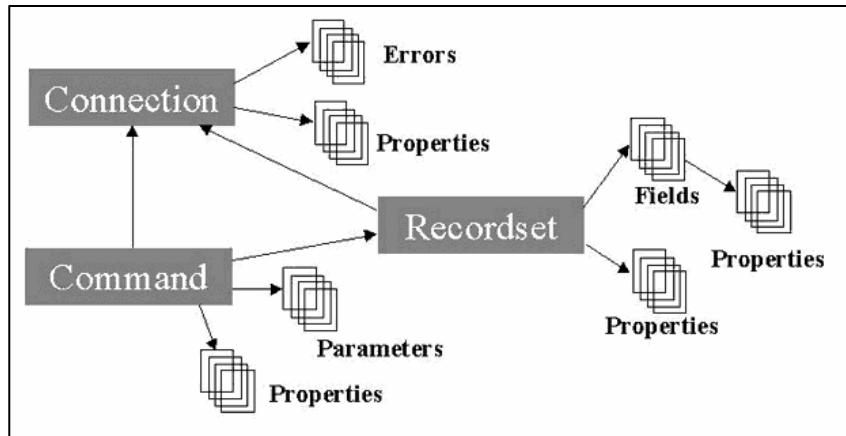
### 4.1 ΤΟ ΣΤΟΙΧΕΙΟ ΕΛΕΓΧΟΥ ΔΕΔΟΜΕΝΩΝ ADO

Η Visual Basic 6.0 συμπεριλαμβάνει ένα προσαρμοσμένο στοιχείο ελέγχου που επιτρέπει στον χρήστη να προβάλλει, να επεξεργάζεται και να ενημερώνει τιμές σε μια βάση δεδομένων χωρίς σύνταξη κώδικα. Αυτό το στοιχείο ελέγχου ονομάζεται στοιχείο ελέγχου δεδομένων ADO. Η απλή προσθήκη αυτού του στοιχείου ελέγχου στη φόρμα επιτρέπει στο πλαίσιο κειμένου, σε ετικέτες και σε άλλα στοιχεία ελέγχου να έχουν "αντίληψη" της παρουσίας δεδομένων. Αυτό σημαίνει ότι με λίγα βήματα και καθόλου προγραμματισμό, ο χρήστης μπορεί να προβάλλει και να επεξεργαστεί τιμές σε μια βάση δεδομένων. Υπάρχουν στην πραγματικότητα τρεις διαφορετικές όψεις του στοιχείου ελέγχου δεδομένων: **Jet**, **ADO** και **Remote Data Objects (RDO)**.



Η ADO διαθέτει διάφορα χαρακτηριστικά που συγκροτούν ένα ιδανικό μοντέλο αντικειμένων για τη δημιουργία πελάτη/διακομιστή και εφαρμογές που βασίζονται στο Web. Παρακάτω παρουσιάζονται μερικά απ' αυτά τα χαρακτηριστικά:

- Μικρό μοντέλο αντικειμένων, το μέγεθος του οποίου το καθιστά εύκολο στη χρήση.
- Καταλαμβάνει λίγο χώρο στη μνήμη, άρα περισσότερη μνήμη απελευθερώνεται στη μνήμη για την εφαρμογή.
- Γρήγορη πρόσβαση σε μια πηγή δεδομένων.
- Οι συνδέσεις μπορούν να δημιουργηθούν on-the-fly, κάτι που εξυπηρετεί αφάνταστα για εφαρμογές που βασίζονται στο Web.
- Υποστηρίζονται διαφορετικοί τύποι δείκτη (δρομέα), συμπεριλαμβανομένων δεικτών διακομιστών και πελατών.
- Υποστηρίζονται αποθηκευμένες διαδικασίες, συμπεριλαμβανομένων παραμέτρων εισόδου και εξόδου και επιστρεφόμενων τιμών.
- Μπορούν να επιστραφούν πολλαπλά recordsets
- Διαχείριση cache για recordsets, για προχωρημένους, η οποία περιλαμβάνει έναν αριθμό γραμμών για επιστροφή καθώς και σε πόσες γραμμές πρέπει να γίνει caching στον πελάτη.
- Τα ασύγχρονα ερωτήματα μπορούν να χρησιμοποιηθούν για τα ερωτήματα εκείνα που θέλουν πολύ χρόνο για να ολοκληρωθούν. Αυτό επιτρέπει στους χρήστες να συνεχίσουν να εργάζονται στην εφαρμογή, ενώ ο διακομιστής βάσης δεδομένων επεξεργάζεται τα δεδομένα.
- Τα συμβάντα θα κληθούν για να υποδείξουν ότι η επεξεργασία έχει ολοκληρωθεί ή ότι η επεξεργασία πρόκειται να λάβει χώρα.



Μοντέλο αντικειμένων ADO

Αν και η ADO υποστηρίζει αυτά τα καταγραμμένα χαρακτηριστικά, είναι απαραίτητη η διαβεβαίωση ο υποκείμενος παροχέας OLE DB τα υποστηρίζει. Δεν έχουν δημιουργηθεί όλοι οι παροχείς το ίδιο. Αυτό που υποστηρίζει ο ένας μπορεί να μην υποστηρίζει ο άλλος.

Το ADO λοιπόν είναι το interface οποιουδήποτε OLE DB παροχέα, συμπεριλαμβανομένων σχεσιακών ή μη βάσεων δεδομένων, e-mail, συστήματα αρχείων, κείμενο, γραφικά, επαγγελματικά αντικείμενα που έχουν δημιουργηθεί από τους χρήστες και οι υπάρχουσες πηγές δεδομένων ODBC.

Το ADO είναι εύκολο στην χρήση του, δεν διαθέτει πολύπλοκη ιεραρχία αντικειμένων όπως το DAO, είναι ανεξάρτητο της γλώσσας και δεν επιβαρύνει ιδιαίτερα το δίκτυο.

Όπως ήδη αναφέραμε το μοντέλο αντικειμένων του ADO είναι ιδιαίτερα απλό. Τα κυριότερα αντικείμενα είναι τα **Connection**, **Command** και **Recordset**. Για να χρησιμοποιηθεί το ADO μέσα από την Visual Basic, πρέπει πρώτα να κληθεί το πλαίσιο διαλόγου References και να επιλεγούν μία από τις δύο ADO type βιβλιοθήκες. Η πρώτη καλείται ADODB κι εμφανίζεται σαν Microsoft ActiveX Data Objects 2.0 Library. Η δεύτερη καλείται ADOR κι εμφανίζεται σαν Microsoft ActiveX Data Objects Recordset 2.0 Library.

Η πρώτη (ADODB) είναι μεγαλύτερη, περιέχει όλα τα αντικείμενα ADO και είναι αυτή που συνήθως που χρησιμοποιείται περισσότερο. Η δεύτερη είναι πιο 'ελαφριά' μορφή της πρώτης και υποστηρίζει μόνο recordsets.

## 4.2 ΤΥΠΟΠΟΙΗΣΕΙΣ ΣΥΝΔΕΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Η τυποποίηση ODBC είναι μια ευρέως χρησιμοποιούμενη τυποποίηση για επικοινωνία μεταξύ εφαρμογών πελατών και βάσεων δεδομένων. Το ODBC βασίζεται στις τυποποιήσεις SQL Call-Level Interface (CLI), που αναπτύχθηκαν από την ομάδα της βιομηχανίας X/Open και το SQL Access Group, αλλά έχουν διάφορες επεκτάσεις. Το ODBC API ορίζει ένα CLI, έναν ορισμό σύνταξης SQL και κανόνες για τις επιτρεπόμενες σειρές κλήσεων CLI. Η τυποποίηση ορίζει επίσης επίπεδα

συμβατότητας για τη σύνταξη CLI και SQL. Για παράδειγμα, ο πυρήνας της CLI έχει εντολές για σύνδεση με μια βάση δεδομένων, προετοιμασία και εκτέλεση SQL εντολών για λήψη αποτελεσμάτων ή τιμές καταστάσεων και για διαχείριση συναλλαγών. Το επόμενο επίπεδο συμβατότητας (επίπεδο 1) απαιτεί υποστήριξη για ανάκληση πληροφοριών καταλόγων και παλιές άλλες λειτουργίες, πέρα και πάνω από το CLI. Το επίπεδο 2 απαιτεί επιπλέον λειτουργίες, όπως τη δυνατότητα να στέλνονται και να ανακαλούνται πίνακες με τιμές παραμέτρων και να ανακαλούνται πιο λεπτομερείς πληροφορίες καταλόγων.

Το ODBC επιτρέπει σε ένα πελάτη να συνδέεται ταυτόχρονα σε πολλές πηγές δεδομένων και να εναλλάσσεται μεταξύ τους, αλλά οι συναλλαγές σε κάθε πηγή γίνονται ανεξάρτητα. Το ODBC API περιγράφεται στις αναφορές [Microsoft 1997] και [Sanders 1998].

Υπάρχουν πολλές πηγές δεδομένων που δεν είναι σχεσιακές βάσεις δεδομένων και μπορεί να μην είναι καθόλου βάσεις δεδομένων. Παραδείγματα τέτοια είναι επίπεδα αρχεία και ηλεκτρονικό ταχυδρομείο. Το OLE-DB της Microsoft είναι ένα C++ API, με στόχους παρόμοιους με το ODBC, αλλά υποστηρίζει και δεδομένα που δεν προέρχονται από βάσεις δεδομένων στα οποία μπορούν να παρέχονται μόνο περιορισμένες δυνατότητες ερωτημάτων και ενημέρωσης. Όπως και το ODBC έτσι και το OLE-DB παρέχει δομές σύνδεσης με μια πηγή δεδομένων, για έναρξη μιας συνόδου, εκτέλεση εντολών και λήψη αποτελεσμάτων με την μορφή ενός συνόλου γραμμής, που είναι ένα σύνολο από γραμμές με τα αποτελέσματα.

Ωστόσο, το OLE DB διαφέρει από το ODBC, με διάφορους τρόπους. Στο ODBC, οι εντολές είναι πάντα στην SQL. Στο OLE DB, οι εντολές μπορεί να είναι σε οποιαδήποτε γλώσσα που υποστηρίζεται από την πηγή δεδομένων. Ενώ κάποιες πηγές μπορεί να υποστηρίζουν SQL, ή περιορισμένο υποσύνολο της SQL, άλλες πηγές μπορεί να παρέχουν μόνο απλές δυνατότητες, όπως πρόσβαση σε δεδομένα ενός επιπέδου αρχείου, χωρίς δυνατότητα ερωτημάτων. Μια άλλη βασική διαφορά του OLE DB από το ODBC είναι ότι ένα σύνολο γραμμής είναι ένα αντικείμενο που μπορεί να είναι κοινόχρηστο από πολλές εφαρμογές, μέσω κοινόχρηστης μνήμης. Ένα σύνολο γραμμής μπορεί να ενημερωθεί από μια εφαρμογή και οι άλλες εφαρμογές που μοιράζονται αυτό το αντικείμενο θα πρέπει να ειδοποιούνται για την αλλαγή.

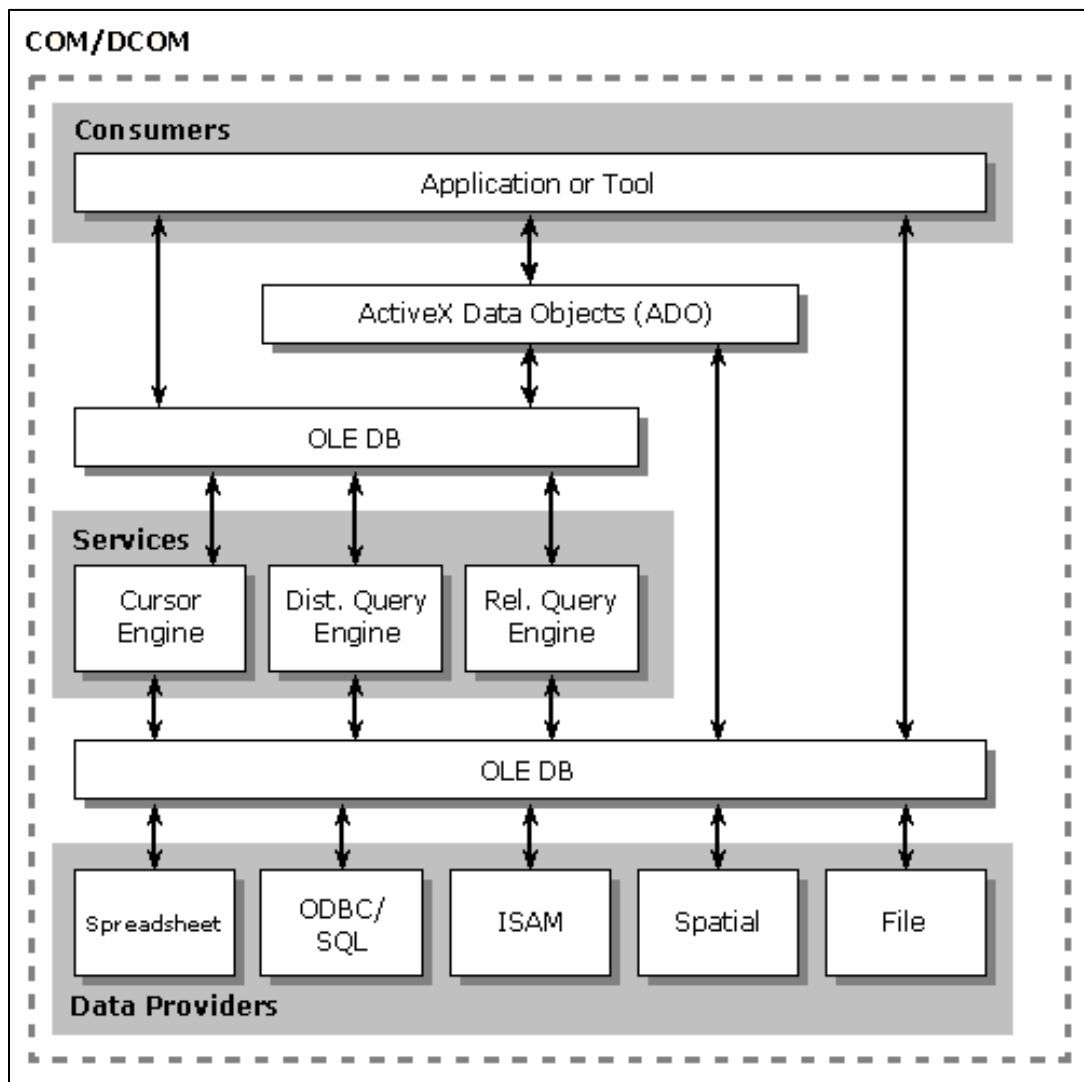
Το Active Data Objects (ADO) API, που επίσης δημιουργήθηκε από τη Microsoft, παρέχει μια εύκολη διασύνδεση για τη λειτουργικότητα OLE DB, που μπορεί να κληθεί ακόμη και από γλώσσες script, όπως τις Visual Basic Script και Java Script. Αναλυτικές πληροφορίες για τα ODBC, OLE-DB και ADO μπορούν να βρεθούν στην [Microsoft/data].

### **4.3 Ο ΠΑΡΟΧΕΑΣ OLE DB**

Ο OLE DB αποτελείται από ένα σύνολο από περιβάλλοντα OLE που εξασφαλίζουν στις εφαρμογές μια κοινή μέθοδο πρόσβασης σε δεδομένα. Δεν έχει σημασία με ποια μορφή αποθηκεύονται αυτά τα δεδομένα όπως κείμενα, XLS, email, MDB ή SQL Server. Αυτά τα περιβάλλοντα που ορίζονται στον OLE DB θα

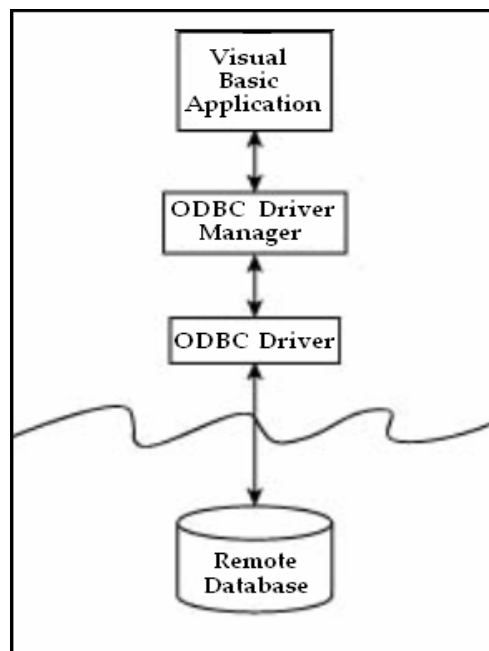
υποστηρίζουν τις δυνατότητες του Database Management System (DBMS) για την ανάκτηση ή και τροποποίηση δεδομένων.

Ο OLE DB χρησιμοποιεί το Component Object Model (COM) ως υποκείμενη τεχνολογία. Μ' αυτές τις υπηρεσίες, μπορούν να εκλείψουν οι αντιγραφές μεταξύ των διαφορετικών εργαλείων πρόσβασης δεδομένων. Επίσης, επειδή το COM είναι ένα πρότυπο του οποίου δικαιώματα έχουν οργανισμοί προτύπων, μπορεί να γίνει μια κοινή πλατφόρμα προτύπων καθώς συμμετέχουν κι άλλες εταιρίες. Αυτό μπορεί να καταστήσει τον κώδικα πιο συμβατό και με περισσότερες δυνατότητες μεταφοράς ανάμεσα στα διάφορα περιβάλλοντα προγραμματισμού και εργαλεία.



## 4.4 Ο ΠΑΡΟΧΕΑΣ ODBC

Ο παροχέας ODBC είναι απλώς το περιβλήμα του προτύπου Open Database Connectivity (ODBC). Ο ODBC είναι ένα πρότυπο που προτείνει η Microsoft για τη σύνδεση με διάφορους διακομιστές βάσεων δεδομένων μέσα από ένα κοινό περιβάλλον προγραμματισμού εφαρμογών (API, application programming interface). Πρόκειται για ένα πρότυπο σύνολο κλήσεων συναρτήσεων που μπορούν να αξιοποιηθούν από τη C, C++, Visual Basic, Delphi, PowerBuilder κ.ο.κ.. Αυτό το API σας επιτρέπει να προγραμματίσετε τον πηγαίο σας κώδικα μια φορά και εύκολα να μεταφέρετε αυτή την εφαρμογή σε έναν άλλο διακομιστή βάσης δεδομένων με λίγες ή και καθόλου αλλαγές κώδικα. Ο ODBC εγκαθίσταται με τη Visual Basic, με



οποιαδήποτε από τα προϊόντα της Microsoft Office και με τη Visual Studio. Το πιθανότερο είναι ο ODBC να είναι ήδη εγκατεστημένος στο σύστημά σας. Πρέπει να μάθετε για τον ODBC γιατί δεν υπάρχουν ακόμη τόσοι πολλοί εγγενείς παροχείς για το OLE DB, οπότε πρέπει να συνεχίζεται να χρησιμοποιείτε προγράμματα οδήγησης ODBC για να έχετε πρόσβαση σε μερικούς από τους διακομιστές βάσεων δεδομένων

### **ΑΝΑΦΟΡΕΣ-ΒΙΒΛΙΟΓΡΑΦΙΑ**

- ✚ [http://MSDN/Microsoft/Frequently Asked Questions About OLE DB](http://MSDN/Microsoft/Frequently%20Asked%20Questions%20About%20OLE%20DB)
- ✚ ISO/IEC Document 9075: 1989(1989).
- ✚ [Sanders 1998] R.E. Sanders, ODBC 3.5 Developer's Guide, McGraw Hill (1998).
- ✚ [Microsoft/data] web τοποθεσία [www.microsoft.com/data](http://www.microsoft.com/data).
- ✚ <http://www.microsoft.com/data/>
- ✚ Geiger, K., *Inside ODBC*, Microsoft Press, Redmond, WA, 1995

## 5. ΤΟ ΠΡΟΤΥΠΟ SQL2

### 5.1 Η ΓΛΩΣΣΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ SQL

Η προέλευση της σχεσιακής γλώσσας SQL, που είναι μια συντομογραφία του Structured Query Language (Δομημένη Γλώσσα Ερωτημάτων) αναπτύχθηκε και υλοποιήθηκε από την IBM στα πλαίσια του ερευνητικού της προγράμματος System R που απέβλεπε στην ανάπτυξη ενός πρότυπου σχεσιακού συστήματος βάσεων δεδομένων (RDBMS) χρησιμοποιώντας τις θεωρητικές βάσεις του E.F. Codd για το σχεσιακό μοντέλο και τη σχεσιακή άλγεβρα, που περιγράφονται στις αναφορές [Codd 1970] και [Codd 1990]. Μετά από την επιτυχία του System R, αρκετές νέες εταιρίες



δημιούργησαν τα δικά τους συστήματα RDBMS χρησιμοποιώντας σαν γλώσσα την SQL. Τα δύο πρώτα RDBMSs που διατέθηκαν στο εμπόριο και βασίζονταν στην SQL ήταν το ORACLE από την Relational Software Inc. (σήμερα ORACLE Corporation) το 1979 και το SQL/DS από την IBM το 1982. Όλες αυτές οι υλοποιήσεις ήταν επεκταμένες διάλεκτοι της γλώσσας, επειδή κάθε εταιρία υλοποίησε τις δικές τις επεκτάσεις. Για αυτό τον λόγο, το Αμερικανικό Ινστιτούτο Εθνικών Προτύπων (American National Standards Institute, ANSI) και ο Διεθνής Οργανισμός Τυποποίησης (International Standards Organization, ISO) ίδρυσαν μια επιτροπή το 1982 με σκοπό να σχεδιάσει μια πρότυπη έκδοση της SQL. Το πρώτο πρότυπο της γλώσσας, που βασικά στηρίχθηκε στην διάλεκτό της από την IBM εκδόθηκε το 1986. Μετά από την έκδοση ενός ενδιάμεσου προτύπου το 1989, ένα μεγαλύτερο πρότυπο το SQL2, ή αλλιώς SQL92, εκδόθηκε τον Δεκέμβριο του 1992. Τα τελευταία χρόνια, και οι δύο οργανισμοί τυποποίησης ανέπτυξαν ένα εκτεταμένο πρότυπο το SQL3 (ή SQL99) που περιλαμβάνει αρκετές νέες αρχές βάσεων δεδομένων, περιλαμβανομένων των ερεθισμών, αποθηκευμένων διαδικασιών και αρκετών αρχών αντικειμενοστραφούς προγραμματισμού.

Σήμερα, υπάρχουν δεκάδες εμπορικά RDBMS που λειτουργούν σε υπολογιστές και λειτουργικά συστήματα διαφόρων δυναμικοτήτων και χαρακτηριστικών, όπως το DB2 από την IBM, το ORACLE από την ORACLE Corporation, το INGRES από την INGRES Corporation, το SYBASE από την SYBASE Inc., το INFORMIX από την INFORMIX Software, το MS-SQL Server μία συμπαραγωγή από την SYBASE Inc. και την Microsoft, το dBASE IV/SQL από την Ashton-Tate, η Access (Microsoft), το Paradox (Borland), Q&A (Symantec), FileMaker Pro (Claris Corp.), Approach (Lotus). Επίσης αρκετά διατίθενται δωρεάν σαν σχεσιακά συστήματα ανοιχτού-κώδικα (open-source) όπως τα PostgreSQL, MySQL, BerkeleyDB κλπ.

Σε αντίθεση με τις παραδοσιακές γλώσσες, σαν τις C, C++ και Java, η SQL είναι μια γλώσσα προσανατολισμένη προς σύνολα. (Οι προηγούμενες ονομάζονται γλώσσες προσανατολισμένες σε εγγραφές ή γλώσσες μιας εγγραφής την φορά.) Αυτό σημαίνει ότι η SQL μπορεί να υποβάλει ερωτήματα σε πολλές γραμμές από ένα ή περισσότερους πίνακες, χρησιμοποιώντας μια μόνο πρόταση. Αυτό το χαρακτηριστικό είναι ένα από τα σημαντικότερα πλεονεκτήματα της SQL, επειδή επιτρέπει την χρήση της γλώσσας σε ένα λογικά υψηλότερο επίπεδο από τις διαδικαστικές γλώσσες. Μια άλλη σημαντική ιδιότητα της SQL είναι ότι δεν είναι διαδικαστική (η σχεσιακή άλγεβρα είναι διαδικαστική). Η SQL υλοποιεί όλες τις πράξεις της σχεσιακής άλγεβρας.

Η SQL, έχει διάφορα τμήματα. Δύο από αυτά περιέχουν αντίστοιχα δύο δευτερεύουσες γλώσσες: μια γλώσσα ορισμού δεδομένων (data definition language, DDL) και μια γλώσσα χειρισμού δεδομένων (data manipulation language, DML). Οι προτάσεις DDL χρησιμοποιούνται για να περιγράψουν το σχήμα των πινάκων της βάσης δεδομένων. Η DDL περιέχει τρεις πρωτογενείς προτάσεις SQL: CREATE αντικείμενο, ALTER αντικείμενο και DROP αντικείμενο. Αυτές παράγουν, τροποποιούν και καταργούν αντικείμενα βάσης δεδομένων όπως βάσεις δεδομένων, πίνακες, στήλες και ευρετήρια. Αντίθετα, η DML περιλαμβάνει όλες τις λειτουργίες που χειρίζονται τα δεδομένα. Υπάρχουν πάντα τέσσερις πρωτογενείς λειτουργίες για χειρισμό της βάσης δεδομένων: ανάκτηση, εισαγωγή, διαγραφή και τροποποίηση (SELECT, INSERT, DELETE και UPDATE). Η γλώσσα SQL-2 περιγράφεται στις

αναφορές καθώς και σε μια πληθώρα άλλων εγχειριδίων περισσότερο προσανατολισμένων προς συγκεκριμένα εμπορικά συστήματα.

Γλώσσα Ορισμού Δεδομένων (DDL)	CREATE   DROP   ALTER TABLE (base table) CREATE   DROP   ALTER VIEW (virtual table) CREATE   DROP   ALTER INDEX (index table) CREATE TABLE SUPPLIERS ( S_No NUMBER(4) NOT NULL, S_Name CHAR(25), S_City CHAR(20) ); CREATE INDEX Supplier_Name ON SUPPLIERS(S_Name);
Γλώσσα Χειρισμού Δεδομένων (DML)	SELECT αναζήτηση SELECT * FROM SUPPLIERS WHERE S_City='Athens'; INSERT εισαγωγή εγγραφής INSERT INTO SUPPLIERS VALUES ( 043, 'John', 'Athens'); DELETE διαγραφή εγγραφής DELETE FROM SUPPLIERS WHERE S_No=043; UPDATE τροποποίηση εγγραφής UPDATE SUPPLIERS SET S_City='Patra' WHERE S_No=043;

## **5.2 ΕΝΝΟΙΕΣ ΣΧΗΜΑΤΟΣ ΚΑΙ ΚΑΤΑΛΟΓΟΥ ΣΤΗΝ SQL2**

Ένα θεμελιώδες χαρακτηριστικό της προσέγγισης βάσεων δεδομένων είναι το ότι ένα σύστημα βάσης δεδομένων δεν περιέχει μόνο την ίδια τη βάση αλλά και τον πλήρη ορισμό ή την περιγραφή αυτής που ονομάζεται σχήμα. Όταν ορίζουμε μια νέα βάση δεδομένων, προσδιορίζουμε μόνο το σχήμα της βάσης δεδομένων στο DBMS. Στο σημείο αυτό, η αντίστοιχη κατάσταση της βάσης δεδομένων είναι η "κενή κατάσταση" χωρίς δεδομένα. Παίρνουμε την "αρχική κατάσταση" της βάσης δεδομένων όταν η βάση φορτωθεί με αρχικά δεδομένα [Date 1995].

Η έννοια του σχήματος εισήχθη στην SQL2 για να ομαδοποιηθούν πίνακες και άλλα δομικά στοιχεία που ανήκουν στην ίδια εφαρμογή βάσης δεδομένων. Ένα σχήμα SQL (SQL schema) αναγνωρίζεται από το όνομα σχήματος (schema name) και περιλαμβάνει κάποια ταυτότητα εξουσιοδότησης για την αναγνώριση του χρήστη ή του λογαριασμού που κατέχει το σχήμα, καθώς και περιγραφείς (descriptors) για κάθε στοιχείο του σχήματος.

Ένα σχήμα δημιουργείται με την εντολή CREATE SCHEMA, η οποία μπορεί να περιέχει τους ορισμούς όλων των στοιχείων του σχήματος. Εναλλακτικά μπορεί να δοθεί στο σχήμα ένα όνομα και μια ταυτότητα εξουσιοδότησης και τα στοιχεία του να οριστούν αργότερα.

Η περιγραφή της βάσης δεδομένων (database schema) ορίζεται κατά το σχεδιασμό της βάσης δεδομένων και δεν αναμένεται να αλλάζει συχνά. Αυτός ο ορισμός αποθηκεύεται στον κατάλογο (catalog) του συστήματος ή λεξικό δεδομένων. Ο κατάλογος περιέχει πάντοτε ένα ειδικό σχήμα που λέγεται **INFORMATION\_SCHEMA**, που περιέχει πληροφορίες όπως η δομή κάθε αρχείου, εγγραφές, ονόματα των αρχείων, τύποι δεδομένων, ο τύπος και η μορφή

αποθήκευσης κάθε στοιχειώδους δεδομένου, καθώς και διάφοροι περιορισμοί επί των δεδομένων.

Το σχήμα πληροφοριών εισήχθη αρχικά στο πρότυπο SQL92. Παρέχει τις προβολές σχήματος πληροφοριών έτσι ώστε οι εφαρμογές που αναπτύχθηκαν για άλλα συστήματα βάσεων δεδομένων να μπορούν να κάνουν χρήση του καταλόγου συστήματος του συστήματος διαχείρισης βάσεων, χωρίς να χρειάζεται να τον χρησιμοποιήσουν απευθείας. Αυτές οι πρότυπες προβολές χρησιμοποιούν διαφορετική ορολογία από το σύστημα, οπότε όταν κάποιος ερμηνεύει ονόματα στηλών, πρέπει να γνωρίζει ότι κατάλογος = βάση δεδομένων, σχήμα = ιδιοκτήτης, και πεδίο τιμών = τύπος δεδομένων που ορίζεται από τον χρήστη (δηλαδή catalog = database, schema = owner, domain = user-defined data type).

Οι παρακάτω ενότητες παρέχουν μια περιγραφή των τριών σημαντικότερων προβολών σχημάτων πληροφοριών.

**Information\_schema.tables:** Η προβολή Information\_schema.tables περιέχει μια γραμμή για κάθε πίνακα στην τρέχουσα βάση δεδομένων, στον οποίο ο χρήστης έχει πρόσβαση. Η προβολή επαναφέρει τις πληροφορίες από τον πίνακα συστήματος sysobjects. Ο παρακάτω πίνακας δείχνει τις τέσσερις στήλες αυτής της προβολής.

Στήλη	Περιγραφή
table_catalog	Το όνομα του καταλόγου (βάσης δεδομένων) στον οποίο ανήκει η προβολή
table_schema	Το όνομα του σχήματος (ιδιοκτήτη) στο οποίο ανήκει η προβολή
table_name	Το όνομα του πίνακα
table_type	Ο τύπος του πίνακα (μπορεί να είναι "BASE TABLE" ή "VIEW")

**Information\_schema.views:** Η προβολή Information\_schema.views περιέχει μια γραμμή για κάθε προβολή στην τρέχουσα βάση δεδομένων που είναι προσπελάσιμη από τον χρήστη. Η προβολή επαναφέρει πληροφορίες για τους πίνακες συστήματος sysobjects και syscomments. Ο παρακάτω πίνακας δείχνει τις πέντε στήλες αυτής της προβολής.

Στήλη	Περιγραφή
table_catalog	Το όνομα του καταλόγου (βάσης δεδομένων) στον οποίο ανήκει η προβολή
table_schema	Το όνομα του σχήματος (ιδιοκτήτη) στον οποίο ανήκει η προβολή
table_name	Το όνομα του βασικού πίνακα
view_definition	Το κείμενο του ορισμού προβολής
check_option	Καθορίζει αν η επιλογή check έχει τεθεί σε ON. Αν ναι, τότε επιστρέφεται "CASCADE". Αλλιώς, επιστρέφεται "NONE".

**Information\_schema.columns:** Η προβολή Information\_schema.columns περιέχει μια γραμμή για κάθε στήλη στην τρέχουσα βάση δεδομένων, που είναι προσπελάσιμη από τον χρήστη. Η προβολή επαναφέρει τις πληροφορίες από τους πίνακες συστήματος sysobjects, systypes και syscomments. Ο παρακάτω πίνακας δείχνει τις έξι επιλεγμένες στήλες αυτής της προβολής.

Στήλη	Περιγραφή
table_catalog	Το όνομα του καταλόγου (βάσης δεδομένων) στον οποίο ανήκει η στήλη
table_schema	Το όνομα του σχήματος (ιδιοκτήτη) στον οποίο ανήκει η στήλη
table_name	Το όνομα του πίνακα στον οποίο ανήκει η στήλη
column_name	Το όνομα της στήλης
ordinal_position	Ο αριθμός αναγνώρισης της στήλης
Data_type	Ο τύπος δεδομένων της στήλης

Ο κατάλογος δηλαδή είναι η θέση όπου είναι αποθηκευμένα τα σχήματα όλων των ειδών (εξωτερικό, εννοιολογικό, εσωτερικό), καθώς και όλες οι αντίστοιχες απεικονίσεις (εξωτερική/εννοιολογική, εννοιολογική/εσωτερική). Οι πληροφορίες που περιέχονται σ' αυτόν τον κατάλογο λέγονται μετα-δεδομένα (meta-data) δηλ. "δεδομένα για τα δεδομένα" και περιγράφουν τη δομή της κυρίως βάσης. Τα μετα-δεδομένα δηλαδή είναι ειδικές πληροφορίες για τα δεδομένα που υπάρχουν στη βάση.

Ο κατάλογος χρησιμοποιείται από το DBMS και περιστασιακά από τους χρήστες της βάσης δεδομένων που χρειάζονται πληροφορίες για τη δομή της βάσης. Το DBMS δεν γράφεται για κάποια συγκεκριμένη εφαρμογή και, επομένως, πρέπει να αναφέρεται στον κατάλογο για να γνωρίζει τη δομή των αρχείων σε μια συγκεκριμένη βάση δεδομένων, δηλαδή πληροφορίες όπως ο τύπος και η μορφή των δεδομένων που θα προσπελάσει. Το DBMS πρέπει να εργάζεται εξ ίσου καλά με οποιοδήποτε αριθμό εφαρμογών βάσεων δεδομένων - για παράδειγμα μια πανεπιστημιακή βάση δεδομένων, μια τραπεζική βάση δεδομένων ή τη βάση δεδομένων μιας επιχείρησης- εφόσον ο ορισμός των βάσεων δεδομένων αποθηκεύεται στον κατάλογο. Δηλαδή ο κατάλογος του συστήματος αποτελεί ο ίδιος μια «μικρο-βάση δεδομένων».

Ένα από τα πιο βολικά χαρακτηριστικά των σχεσιακών συστημάτων είναι ότι, σ' ένα τέτοιο σύστημα, ο ίδιος ο κατάλογος αποτελείται από πίνακες (για την ακρίβεια, από **πίνακες συστήματος**, που ονομάζονται έτσι για να τους ξεχωρίζουμε από τους συνηθισμένους πίνακες του χρήστη) [Date 1995]. Έτσι, οι χρήστες μπορούν να ερευνούν τον κατάλογο ακριβώς με τον ίδιο τρόπο που ερευνούν τα δικά τους δεδομένα. Για παράδειγμα, ο κατάλογος κατά κανόνα περιέχει δύο πίνακες συστήματος, που ονομάζονται TABLES (πίνακες) και COLUMNS (στήλες), των οποίων σκοπός είναι να περιγράφουν τους πίνακες που είναι γνωστοί στο σύστημα και τις στήλες αυτών των πινάκων. Για τη βάση δεδομένων τμημάτων (DEPT) και υπαλλήλων (EMP), οι πίνακες TABLES και COLUMNS θα μπορούσαν να είναι σε γενικές γραμμές όπως στην παρακάτω εικόνα.

TABLES	TABNAME	COLCOUNT	ROWCOUNT	.....
	DEPT	3	3	.....
	EMP	4	4	.....
	.....	....	....	.....

COLUMNS	TABNAME	COLNAME	.....
	DEPT	DEPT#	.....
	DEPT	DNAME	.....
	DEPT	BUDGET	.....
	EMP	EMP#	.....
	EMP	ENAME	.....
	EMP	DEPT#	.....
	EMP	SALARY	.....
	.....	.....	.....

Από τον πίνακα TABLES φαίνεται ότι ο πίνακας DEPT έχει τρεις στήλες και τρεις εγγραφές, και ο πίνακας EMP έχει 4 στήλες και 4 εγγραφές. Στον πίνακα COLUMNS φαίνονται επίσης και τα ονόματα των γνωρισμάτων (στηλών) και τα πρωτεύοντα και ξένα κλειδιά.

Οι πληροφορίες που αποθηκεύονται γενικά σε ένα κατάλογο ενός Σχεσιακού σχήματος DBMS μπορούν να περιλαμβάνουν περιγραφές των ονομάτων των σχέσεων, των ονομάτων των χαρακτηριστικών, των πεδίων ορισμού (τύπων δεδομένων) των γνωρισμάτων, των πρωτευόντων κλειδιών, των γνωρισμάτων που είναι δευτερεύοντα κλειδιά, των ξένων κλειδιών και των άλλων τύπων περιορισμών καθώς και του εξωτερικού επιπέδου για τις όψεις και εσωτερικού περιγραφές για τις δομές αποθήκευσης και τα ευρετήρια. Συμπεριλαμβάνονται πληροφορίες σχετικά με την ασφάλεια και τις διαδικασίες, πληροφορίες που προσδιορίζουν τη δικαιοδοσία προσπέλασης των χρηστών στις σχέσεις και τις όψεις, καθώς και τους δημιουργούς ή ιδιοκτήτες κάθε σχέσης.

## **6. ΑΝΑΛΥΣΗ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΥΠΟΣΤΗΡΙΖΟΜΕΝΩΝ ΑΠΟ ΤΗΝ ΕΦΑΡΜΟΓΗ ΣΥΣΤΗΜΑΤΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΣΧΕΣΙΑΚΩΝ ΒΑΣΕΩΝ**

Πριν γίνει περιγραφή και ανάλυση της εφαρμογής, είναι αναγκαία μια σύντομη αναφορά στα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων στα οποία ο χρήστης θα έχει τη δυνατότητα να συνδεθεί και να εξάγει δεδομένα μέσω της εφαρμογής. Η ανάγκη αυτή είναι επιτακτική καθώς για να κατανοηθεί ο τρόπος ανάπτυξης του προγράμματος, πρέπει να απαντηθούν ερωτήματα σχετικά με το τι είδους συστήματα βάσεων είναι αυτά, ποιους τύπους δεδομένων υποστηρίζουν, ποιοι είναι οι κανόνες που πρέπει να προσέξει ο χρήστης για τη σωστή δημιουργία και ονοματολογία των πινάκων, ποιες είναι οι ομοιότητες και οι διαφορές αυτών μεταξύ τους και άλλα τέτοια ενδιαφέροντα ερωτήματα.

Στις ενότητες που ακολουθούν γίνεται μία προσπάθεια για να απαντηθούν τα περισσότερα από τα παραπάνω ερωτήματα.

## 6.1 Paradox 5.X

Ο Paradox είναι ένα σύστημα διαχείρισης βάσεων δεδομένων που και αυτός όπως και η Dbase είναι διαθέσιμος και στο DOS και στα Windows

Ένας πίνακας Paradox είναι ένα αρχείο με επέκταση “.db” του οποίου τα δεδομένα μπορούμε να δούμε και να αλλάξουμε και μέσω της Access. Δεν μπορούμε όμως να μεταβάλλουμε τα πεδία του.

Για την ονοματολογία των πινάκων ισχύει ότι και στη Dbase, δηλαδή το όνομα πίνακα πρέπει να αποτελείται από οκτώ το πολύ χαρακτήρες, ο πρώτος των οποίων πρέπει να είναι αλφαβητικός. Κάθε πεδίο του πίνακα πρέπει επίσης να έχει όνομα με μήκος μέχρι οκτώ αλφαριθμητικών, ο πρώτος των οποίων πρέπει και πάλι να είναι αλφαβητικός. Το μέγιστο πλήθος πεδίων που υποστηρίζεται σε ένα πίνακα είναι 255.

Ένας πίνακας Paradox 5.X πρέπει να έχει primarykey για να μπορεί να τροποποιηθεί. Διαφορετικά δε γίνεται να προσθέσουμε ή να διαγράψουμε δεδομένα απ' αυτόν.

Πεδία δεν μπορούν να προστεθούν μόλις καθοριστεί ένα index σε έναν πίνακα..

Οι τύποι δεδομένων που υποστηρίζονται σε ένα πίνακα Paradox 5.X είναι:

Τύπος Δεδομένων	Περιγραφή
<b>Alphanumeric</b>	Κείμενο μέχρι 255 χαρακτήρες
<b>Number</b>	Οποιαδήποτε αριθμητική τιμή
<b>Short Number</b>	Ακέρεος αριθμός
<b>Currency</b>	Αριθμητική τιμή. Το μέγεθος δεν διευκρινίζεται. Η ακρίβεια είναι μέχρι 6 δεκαδικές θέσεις.
<b>Date</b>	1 Ιανουαρίου ..9999 Π.Χ. έως τις 31 Δεκεμβρίου ΑΓΓΕΛΙΑ ..9999
<b>Memo</b>	Σετ χαρακτήρων απεριόριστου μήκους
<b>OLE</b>	Word document, Spreadsheet, Picture, Sound, κλπ
<b>Graphic</b>	.BMP, .PCX, .TIF, .GIF, or .EPS, αλλά αποθηκευμένα στα χωριστά αρχεία από τον πίνακα
<b>Binary</b>	Ακατέργαστα στοιχεία οποιουδήποτε μεγέθους.
<b>Formatted Memo</b>	Σετ χαρακτήρων απεριόριστου μήκους

Στον παρακάτω πίνακα οι τύποι δεδομένων του ADO αντιστοιχίζονται με τους τύπους δεδομένων της Paradox. Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για τη δημιουργία πίνακα Paradox από αρχείο xml.

Τύπος δεδομένων ADO	Τιμή τύπου ADO	Τύπος δεδομένων Paradox 5.X
AdArray	<b>8192</b>	alphanumeric
adBigInt	<b>20</b>	number
adBoolean	<b>11</b>	number
adChar	<b>129</b>	alphanumeric
adCurrency	<b>6</b>	currency

adDate	7	date
adDBDate	133	date
adDBTime	134	date
adDBTimeStamp	135	date
adDecimal	14	number
adDouble	5	number
adDUID	72	number
adInteger	3	number
adLongVarBinary	205	ole
adLongVarChar	201	memo/alphanumeric
adLongVarWChar	203	memo/alphanumeric
adNumeric	131	number
adSingle	4	number
adSmallInt	2	number
adTinyInt	16	number
adUnsignedBigInt	21	number
adUnsignedInt	19	number
adUnsignedSmallInt	18	number
adUnsignedTinyInt	17	number
adVarBinary	204	ole
adVarChar	200	alphanumeric
adVarWChar	202	memo/alphanumeric
adWChar	130	memo/alphanumeric
adVarNumeric	139	number
adBinary	128	binary

## **6.2 dBASE 5**

Η dBASE 5 είναι προϊόν της Borland και είναι διαθέσιμη και στο DOS και στα Windows. Είναι dBASE δεν είναι αποτελεί πλέον δύναμη στην αγορά λογισμικού βάσεων δεδομένων, και δεν ανταγωνίζεται τα προϊόντα που χρησιμοποιούν το πρότυπο SQL.

Για κάθε βάση δεδομένων dBASE δημιουργείται ξεχωριστός κατάλογος αρχείων. Ακολουθούνται οι κανόνες ονοματολογίας του DOS. Το όνομα κάθε καταλόγου πρέπει να αποτελείται από οκτώ το πολύ χαρακτήρες, ο πρώτος των οποίων πρέπει να είναι αλφαβητικός.

Κάθε πίνακας μιας βάσης δεδομένων dBASE αντιστοιχεί σε ένα αρχείο (αρχείο DOS). Ένα χαρακτηριστικό της dBASE είναι η επέκταση “.dbf” των πινάκων της, η οποία έχει υιοθετηθεί σε διάφορες άλλες εφαρμογές. Το όνομα πίνακα πρέπει να αποτελείται από οκτώ το πολύ χαρακτήρες, ο πρώτος των οποίων πρέπει να είναι αλφαβητικός. Κάθε πεδίο του πίνακα πρέπει να έχει όνομα με μήκος μέχρι δέκα αλφαριθμητικών, ο πρώτος των οποίων πρέπει να είναι αλφαβητικός.

Μορφή	Μέγιστος Αριθμός πεδίων ενός πίνακα
DBF 2 (dBASE II)	32
DBF 3 (dBASE III)	128
DBF 4 (dBASE IV)	256

Για τα ευρετήρια σχύουν όσα και για τα ονόματα πινάκων. Δημιουργείται όμως μόνο ένα ευρετήριο (ένα αρχείο στο DOS) για κάθε πίνακα. Σ' αυτό μπορούν να συμπεριληφθούν μέχρι 47 επιμέρους ευρετήρια (ένα για το πρωτεύον και για κάθε δευτερεύον κλειδί του πίνακα).

Παρακάτω παρουσιάζονται οι τύποι δεδομένων των πεδίων που υποστηρίζονται σε μία βάση Dbase 5.

Τύπος	Περιγραφή
<b>char</b>	Συμβολοσειρά σταθερού μήκους 'n' χαρακτήρων με μέγιστη τιμή n το 254
<b>date</b>	Ημερομηνία με πιθανές μορφές: EEEE.MM.HH MM-HH-EEEE MM/HH/EEEE HH/MM/EEEE HH.MM.EEEE
<b>Float(x,y)</b>	Προσημασμένος δεκαδικός αριθμός κινητής υποδιαστολής 'n' συνολικών ψηφίων και 'm' δεκαδικών. Η ακρίβεια κυμαίνεται μεταξύ 1 και 20 ψηφίων ενώ οι δεκαδικές θέσεις μεταξύ 0 και 18.
<b>Logical</b>	Τιμές 'Ναι'/'Όχι', 'Σωστός'/'Λάθος', 'On'/'Off', '1'/'0'
<b>Memo</b>	Κείμενο ή συνδυασμοί κειμένων και αριθμών
<b>numeric(x,y)</b>	Προσημασμένος δεκαδικός αριθμός σταθερής υποδιαστολής 'n' συνολικών ψηφίων και 'm' δεκαδικών. Η ακρίβεια κυμαίνεται μεταξύ 1 και 20 ψηφίων ενώ οι δεκαδικές θέσεις μεταξύ 0 και 18.
<b>oleobject</b>	Ψηφιακά αρχεία

Στον παρακάτω πίνακα οι τύποι δεδομένων του ADO αντιστοιχίζονται με τους τύπους δεδομένων της Dbase5. Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για τη δημιουργία πίνακα Dbase από αρχείο xml.

Τύπος δεδομένων ADO	Τιμή τύπου ADO	Τύπος δεδομένων dBASE
AdArray	<b>8192</b>	memo
adBigInt	<b>20</b>	numeric
adBoolean	<b>11</b>	logical
adChar	<b>129</b>	char
adCurrency	<b>6</b>	memo
adDate	<b>7</b>	date
adDBDate	<b>133</b>	date
adDBTime	<b>134</b>	date
adDBTimeStamp	<b>135</b>	date
adDecimal	<b>14</b>	numeric



adDouble	<b>5</b>	float
adDUID	<b>72</b>	numeric
adInteger	<b>3</b>	numeric
adLongVarBinary	<b>205</b>	oleobject
adLongVarChar	<b>201</b>	memo
adLongVarWChar	<b>203</b>	memo
adNumeric	<b>131</b>	numeric
adSingle	<b>4</b>	numeric
adSmallInt	<b>2</b>	numeric
adTinyInt	<b>16</b>	numeric
adUnsignedBigInt	<b>21</b>	numeric
adUnsignedInt	<b>19</b>	numeric
adUnsignedSmallInt	<b>18</b>	numeric
adUnsignedTinyInt	<b>17</b>	numeric
adVarBinary	<b>204</b>	oleobject
adVarChar	<b>200</b>	char
adVarWChar	<b>202</b>	memo
adWChar	<b>130</b>	memo
adVarNumeric	<b>139</b>	numeric
adBinary	<b>128</b>	oleobject

### **6.3 Η Microsoft Access**

Η Microsoft Access είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων της εταιρίας Microsoft. Συνδυάζει τη μηχανή σχεσιακών βάσεων Jet με ένα γραφικό interface τα οποία επιτρέπουν τη γρήγορη ανάπτυξη των εφαρμογών από το χρήστη.

Μία βάση δεδομένων της Access είναι ένα αρχείο με επέκταση “.mdb” το οποίο μπορεί να αποθηκευτεί σε οποιοδήποτε κατάλογο στον σκληρό ενός υπολογιστή και να ανοιχτεί ανά πάσα στιγμή από το ίδιο σημείο. Το μέγιστο μέγεθος μίας βάσης δεδομένων της Access είναι 2 gigabytes ,ενώ το μέγιστο πλήθος αντικειμένων που μπορεί να έχει είναι 32.768. Το όνομα κάθε αντικειμένου της βάσης μπορεί να έχει μέγιστο πλήθος χαρακτήρων το 64.

Ο πίνακας μίας βάσης αποτελεί το κυριότερο αντικείμενό της. Το μέγιστο μέγεθος χωρητικότητας που καταλαμβάνει ένας πίνακας είναι 2 gigabytes και το όνομα του μπορεί να έχει το πολύ 64 χαρακτήρες μήκος. Ο κενός χαρακτήρας επιτρέπεται μέσα σ’ αυτό. Σε μία βάση Access μπορούν να είναι ανοιχτοί την ίδια χρονική περίοδο μέχρι 2048 πίνακες.

Ο μέγιστος αριθμός πεδίων ενός πίνακα είναι 255 και των ευρετηρίων του 32. Το μέγιστο πλήθος χαρακτήρων του ονόματος του πεδίου είναι 64 και το μέγιστο πλήθος σε χαρακτήρες των εγγραφών του είναι 2000.

Οι τύποι δεδομένων που υποστηρίζει η Access για τα πεδία των πινάκων παρουσιάζονται και περιγράφονται παρακάτω.

<b>Τύπος</b>	<b>Περιγραφή</b>	<b>Μέγεθος</b>
<b>Text</b>	Κείμενο ή συνδυασμοί χαρακτήρων και αριθμών, καθώς και αριθμοί που δεν απαιτούν υπολογισμούς, όπως οι τηλεφωνικοί αριθμοί.	Μέχρι 255 χαρακτήρες ή το μήκος του ορίζεται στην FieldSize ιδιότητα του πεδίου.
<b>Memo</b>	Μεγάλο κείμενο ή συνδυασμοί κειμένων και αριθμών.	Μέχρι 65.535 χαρακτήρες.
<b>Number</b>	Αριθμητικά στοιχεία που χρησιμοποιούνται στους μαθηματικούς υπολογισμούς.	1, 2, 4, or 8 bytes ανάλογα τον αριθμό
<b>Date/Time</b>	Τιμές ημερομηνίας και χρόνου για τα έτη 100 μέχρι 9999,	8 bytes.
<b>Currency</b>	Νομισματικές τιμές και αριθμοί που χρησιμοποιούνται στους μαθηματικούς υπολογισμούς και που έχουν μία έως τέσσερις δεκαδικές θέσεις.	8 bytes.
<b>AutoNumber</b>	Αυτόματος Αριθμός αυξημένος κατά 1 όσο αυξάνεται και το πλήθος των εγγραφών.	4 bytes
<b>Yes/No</b>	Τιμές 'Ναι'/'Όχι', 'Σωστός'/'Λάθος', 'On'/'Off'	1 bit.
<b>OLE Object</b>	Ένα αντικείμενο (όπως ένα λογιστικό φύλλο (spreadsheet) του Microsoft Excel, ένα έγγραφο Microsoft Word, μια γραφική παράσταση, ήχοι, ή άλλα ψηφιακά στοιχεία) συνδεδεμένο σε έναν πίνακα της Microsoft Access.	Μέχρι 1 gigabyte (που περιορίζεται από τη χωρητικότητα δίσκου)
<b>Hyperlink</b>	Κείμενο ή συνδυασμοί κειμένου και αριθμών που αποθηκεύονται ως κείμενα και χρησιμοποιούνται ως διευθύνσεις Internet.	

Στον παρακάτω πίνακα οι τύποι δεδομένων του ADO αντιστοιχίζονται με τους τύπους δεδομένων της Access. Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για την εισαγωγή αρχείου xml σε βάση Access.

<b>Τύπος δεδομένων ADO</b>	<b>Τιμή τύπου ADO</b>	<b>Τύπος δεδομένων MS Access</b>
AdArray	<b>8192</b>	Text
adBigInt	<b>20</b>	Number
adBoolean	<b>11</b>	Yes/No
adChar	<b>129</b>	Text
adCurrency	<b>6</b>	Currency
adDate	<b>7</b>	Date/Time
adDBDate	<b>133</b>	Date/Time
adDBTime	<b>134</b>	Date/Time
adDBTimeStamp	<b>135</b>	Date/Time

adDecimal	14	Number
adDouble	5	Number
adDUID	72	AutoNumber
adInteger	3	Number
adLongVarBinary	205	OLE
adLongVarChar	201	Memo
adLongVarWChar	203	Memo
adNumeric	131	Number
adSingle	4	Number
adSmallInt	2	Number
adTinyInt	16	Number
adUnsignedBigInt	21	Number
adUnsignedInt	19	Number
adUnsignedSmallInt	18	Number
adUnsignedTinyInt	17	Number
adVarBinary	204	OLE
adVarChar	200	Text
adVarWChar	202	Text
adWChar	130	Text
adVarNumeric	139	Number
adBinary	128	OLE

## 6.4 Microsoft SQL Server

Ο **Microsoft SQL Server** είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων της Microsoft. Υποστηρίζει τη διάλεκτο SQL, την πιο κοινή γλώσσα βάσεων δεδομένων. Χρησιμοποιείται συνήθως για τις μικρομεσαίες σε μέγεθος βάσεις και είναι από τα ανταγωνιστικότερα συστήματα βάσεων δεδομένων SQL για στον τομέα αγοράς. Ο Microsoft SQL Server είναι μία βάση δεδομένων τύπου Client/Server (που χρησιμοποιείται δηλαδή δικτυακά) και υποστηρίζει το πρότυπο ODBC.

Για κάθε βάση δεδομένων δημιουργούνται δύο ξεχωριστά αρχεία ,το αρχείο δεδομένων (**dbfile** με επέκταση “.MDF”) και το αρχείο ημερολογίου συναλλαγών (**LogFile** με επέκταση “.LDF”). Η μέγιστη χωρητικότητα μίας βάσης δεδομένων είναι απεριόριστη (και για τα δύο αρχεία) και προκαθορίζεται κατά τη δημιουργία της βάσης , ενώ το μέγιστο πλήθος βάσεων που επιτρέπεται να υπάρχουν ταυτόχρονα στον διακομιστή είναι 32,767. Όσο αναφορά τον πλήθος των αντικειμένων (όπως πίνακες,ερωτήματα και άλλα) που μπορούν να υπάρχουν σε μία βάση δεν πρέπει να ξεπερνά τα 2.147.483.647

Το όνομα ενός πίνακα είναι μοναδικό σε μία βάση και η παρουσία του κενού χαρακτήρα μέσα σ’ αυτό επιτρέπεται. Επίσης δεν επιτρέπονται σ’ αυτό κόμματα, τελείες και γενικά τα σύμβολα. Το μέγιστο πλήθος σε χαρακτήρες του ονόματος του πίνακα καθώς και των ονομάτων των πεδίων του είναι 128. Ένας πίνακας μπορεί να έχει το πολύ 1024 πεδία στα οποία αντιστοιχίζονται ο εγγραφές ανάλογα με το άνομα αυτών και των τύπο δεδομένων τους. Συγκεκριμένος περιορισμός στο πλήθος των εγγραφών δεν υπάρχει.

Παρακάτω παρουσιάζονται οι τύποι δεδομένων των πεδίων που υποστηρίζονται σε μία βάση του Microsoft SQL Server.

Τύπος	Περιγραφή	Μέγεθος
<b>Binary</b>	Σταθερά δυαδικά στοιχεία μήκους	8000 bytes το μέγιστο
<b>Bit</b>	Ακέραιος αριθμός με τιμή 1 ή 0	Ελάχιστο το 1 byte
<b>Char</b>	Καθορισμένου μήκους χαρακτήρας non-unicode με ένα ανώτατο μήκος 8.000 χαρακτήρων	8000 bytes το μέγιστο
<b>Datetime</b>	Στοιχεία ημερομηνίας και χρόνου από τις 1 Ιανουαρίου ..1753, μέχρι τις 31 Δεκεμβρίου ..9999, με μια ακρίβεια 3,33 χιλιοστά του δευτερολέπτου	8 bytes
<b>Decimal</b>	Σταθερής ακρίβειας αριθμητικά στοιχεία με κλίμακα από το $-10^{38} + 1$ μέχρι $10^{38} - 1$ .	5-17 bytes ανάλογα με την ακρίβεια
<b>Float</b>	Αριθμητικά στοιχεία κινητής υποδιαστολής και κλίμακας από $-3.40E + 38$ μέχρι $3.40E + 38$	4-8 bytes
<b>Int</b>	Ακέραιος αριθμός από $-2^{31}$ ( $-2.147.483.648$ ) μέσω $2^{31} - 1$ ( $2.147.483.647$ ).	4 bytes
<b>Image</b>	Δυαδικά στοιχεία μεταβλητού-μήκους με μέγιστο μήκος $2^{31} - (2.147.483.647)$ bytes	$2^{31} - 1$ bytes το μέγιστο
<b>Money</b>	Νομισματικές τιμές στοιχείων μεταξύ $-2^{63}$ ( $-922,337,203,685,477.5808$ ) και $2^{63} - 1$ ( $+922,337,203,685,477.5807$ ), με ακρίβεια $1/10000$ της νομισματικής μονάδας.	8 bytes
<b>nChar</b>	Καθορισμένου μήκους στοιχεία Unicode με μέγιστο μήκος 4.000 χαρακτήρων.	4000 bytes το μέγιστο
<b>nVarchar</b>	Στοιχεία Unicode μεταβλητού-μήκους με μέγιστο μήκος 4.000 χαρακτήρων. Ο Sysname είναι ένας καθορισμένος από το χρήστη τύπος δεδομένων που είναι λειτουργικά ισοδύναμος με τον nvarchar (128) και χρησιμοποιείται ονόματα αντικειμένων βάσεων δεδομένων.	4000 bytes το μέγιστο
<b>nText</b>	Στοιχεία Unicode μεταβλητού-μήκους με μέγιστο μήκος $2^{30} - (1.073.741.823)$ χαρακτήρες.	$2^{30} - 1$ bytes το μέγιστο
<b>Real</b>	Αριθμητικά στοιχεία κινητής	4 bytes

	υποδιαστολής και κλίμακας από - 3.40E + 38 μέχρι 3.40E + 38,	
<b>Smalldatetime</b>	Στοιχεία ημερομηνίας και χρόνου από τις 1 Ιανουαρίου ..1900, μέχρι τις 6 Ιουνίου ..2079, με ακρίβεια του ενός λεπτού.	4 bytes
<b>Smallint</b>	Ακέραιος αριθμός από $2^{15}$ (- 32.768) μέχρι $2^{15} - 1$ (32.767).	2 bytes
<b>SmallMoney</b>	Νομισματικές μονάδες κλίμακας από -214,748.3648 μέχρι + 214,748.3647, με ακρίβεια 1/10000 της νομισματικής μονάδας.	4 bytes
<b>Text</b>	Στοιχεία non-unicode μεταβλητού-μήκους με μέγιστο μήκος $2^{31} - 1$ (2.147.483.647) χαρακτήρες	$2^{31} - 1$ bytes το μέγιστο
<b>Timestamp</b>	Ένας μοναδικός αριθμός που ενημερώνεται κάθε φορά που ενημερώνεται και μία εγγραφή	8 bytes
<b>Tinyint</b>	Ακέραιος αριθμός από 0 μέχρι 255,	1 byte
<b>Uniqueidentifier</b>	Ένας συνολικά μοναδικός αριθμός (GUID).	
<b>Varbinary</b>	Δυαδικά στοιχεία μεταβλητού-μήκους με μέγιστο μήκος 8.000 bytes.	8000 bytes το μέγιστο
<b>Varchar</b>	Στοιχεία non-unicode μεταβλητού-μήκους με μέγιστο τα 8.000 bytes.	8000 bytes το μέγιστο

Στον παρακάτω πίνακα οι τύποι δεδομένων του **ADO** αντιστοιχίζονται με τους τύπους δεδομένων του Microsoft SQL Server. Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για την εισαγωγή αρχείου xml σε βάση SQL Server.

<b>Τύπος δεδομένων ADO</b>	<b>Τιμή τύπου ADO</b>	<b>Τύπος δεδομένων MS_SQL Server</b>
AdArray	<b>8192</b>	text
adBigInt	<b>20</b>	bigint
adBoolean	<b>11</b>	bit
adChar	<b>129</b>	char
adCurrency	<b>6</b>	money
adDate	<b>7</b>	datetime
adDBDate	<b>133</b>	datetime
adDBTime	<b>134</b>	datetime
adDBTimeStamp	<b>135</b>	timestamp
adDecimal	<b>14</b>	decimal
adDouble	<b>5</b>	float
adDUID	<b>72</b>	uniqueidentifier
adInteger	<b>3</b>	int

adLongVarBinary	<b>205</b>	image
adLongVarChar	<b>201</b>	text
adLongVarWChar	<b>203</b>	ntext
adNumeric	<b>131</b>	decimal
adSingle	<b>4</b>	real
adSmallInt	<b>2</b>	smallint
adTinyInt	<b>16</b>	bit
adUnsignedBigInt	<b>21</b>	bigint
adUnsignedInt	<b>19</b>	int
adUnsignedSmallInt	<b>18</b>	smallint
adUnsignedTinyInt	<b>17</b>	tinyint
adVarBinary	<b>204</b>	varbinary
adVarChar	<b>200</b>	varchar
adVarWChar	<b>202</b>	nvarchar
adWChar	<b>130</b>	nchar
adVarNumeric	<b>139</b>	numeric
adBinary	<b>128</b>	Binary

## 6.5 Ο MySQL

Ο **MySQL** είναι ένας πολύ γρήγορος, πολύπλοκος, πολλήχρηστος, και γερός διακομιστής σχεσιακών βάσεων δεδομένων SQL (δομημένη γλώσσα διατύπωσης ερωτήσεων). Είναι ένα ελεύθερο λογισμικό. Είναι χορηγημένο με την **άδεια ευρέως κοινού**.

Κάθε πίνακας δεδομένων αποθηκεύεται στο σύστημα σαν ξεχωριστό σύνολο τριών αρχείων, ενός αρχείου με επέκταση “.frm”, ενός δευτέρου με επέκταση “.MYD” και ενός τρίτου με επέκταση “.MYI”

Στον MySQL ο χρήστης μπορεί να επιλέξει μεταξύ τριών βασικών τύπων πινάκων (ISAM, HEAP και MyISAM). Ο προεπιλεγμένος τύπος για το σύστημα είναι MyISAM. Εάν ο χρήστης προσπαθήσει να δημιουργήσει ένα πίνακα χωρίς να επιλέξει τύπο, ο **MySQL** θα δημιουργήσει έναν πίνακα τύπου MyISAM

Ο MyISAM τύπος τώρα, υποδιαιρείται σε ακόμη τρεις τύπους πινάκων (Static, Dynamic και Compressed). Δύο από αυτούς επιλέγονται αυτόματα από το σύστημα ανάλογα με τον τύπο δεδομένων που χρησιμοποιούνται. Ο τρίτος, compressed ή συμπιεσμένος πίνακας, δημιουργείται μόνο με το εργαλείο myisampack

Ο τύπος Static δημιουργείται αυτόματα όταν ο πίνακας δεν περιέχει τους τύπους δεδομένων VARCHAR, BLOB ή TEXT. Ο τύπος Dynamic δημιουργείται σε αντίθεση με τον Static όταν περιέχονται οι παραπάνω τύποι δεδομένων.

Η βάση δεδομένων, οι πίνακες και τα πεδία ακολουθούν τους ίδιους κανόνες ονοματολογίας

Αντικείμενο	Μέγιστο πλήθος χαρακτήρων	Επιτρεπόμενοι χαρακτήρες
Βάση Δεδομένων	64	Οποιοσδήποτε χαρακτήρας που επιτρέπεται σε ένα όνομα καταλόγου εκτός από ` / "`.
Πίνακας	64	Οποιοσδήποτε χαρακτήρας που επιτρέπεται σε ένα όνομα αρχείων, εκτός από ` / " ή `.`.
Πεδίο	64	Όλοι οι χαρακτήρες.

Παρακάτω παρουσιάζονται οι τύποι δεδομένων των πεδίων που υποστηρίζονται σε μία βάση MySQL.

Τύπος	Περιγραφή	Μέγεθος
<b>TINYINT</b>	Ένας πολύ μικρός ακέραιος αριθμός. Η προσημασμένη κλίμακα είναι -128 έως 127. Η μη προσημασμένη είναι 0 έως 255	1 byte
<b>BIT</b>	‘1’ ή ‘0’	
<b>BOOL</b>	‘1’ ή ‘0’	
<b>SMALLINT</b>	Ένας μικρός ακέραιος αριθμός. Η προσημασμένη κλίμακα είναι -32768 έως 32767. Η μηπροσημασμένη κλίμακα είναι 0 έως 65535	2 bytes
<b>MEDIUMINT</b>	Ένας μεσαίου μεγέθους ακέραιος αριθμός. Η προσημασμένη κλίμακα είναι -8388608 έως 8388607. Η μηπροσημασμένη κλίμακα είναι 0 έως 16777215	3 bytes
<b>INT</b>	Ένας ακέραιος αριθμός κανονικός-μεγέθους. Η προσημασμένη κλίμακα είναι -2147483648 έως 2147483647. Η μηπροσημασμένη κλίμακα είναι 0 έως 4294967295	4 bytes
<b>INTEGER</b>	Όπως ο INT	4 bytes
<b>BIGINT</b>	Ένας μεγάλος ακέραιος αριθμός. Η προσημασμένη κλίμακα είναι -9223372036854775808 έως 9223372036854775807. Η μηπροσημασμένη κλίμακα είναι 0 έως 18446744073709551615	8 bytes
<b>REAL or DOUBLE</b>	Ένας floating-point κανονικού μεγέθους και διπλής ακρίβειας αριθμός. Δεν μπορεί να είναι μηπροσημασμένος. Οι επιτρεπόμενες τιμές είναι -1.7976931348623157E+308 μέχρι -2.225073858507201K-308, 0, και 2.225073858507201K-308 σε 1.7976931348623157E+308..	8 bytes
<b>FLOAT</b>	Ένας floating-point αριθμός. Δεν μπορεί να είναι μηπροσημασμένος. Το μέγεθος ακρίβειας είναι <=24 δεκαδικά ψηφία για έναν floating-point αριθμό μονής ακρίβειας	4 ή 8 bytes

	και μεταξύ 25 και 53 για έναν floating-point διπλής ακρίβειας αριθμό	
<b>DECIMAL</b>	Ένας unpacked floating-point αριθμός. Δεν μπορεί να είναι μηπροσημασμένος. Συμπεριφέρεται όπως ένας char . Το "unpacked" σημαίνει ότι ο αριθμός αποθηκεύεται ως string, χρησιμοποιώντας έναν χαρακτήρα για κάθε ψηφίο της τιμής	
<b>NUMERIC</b>	Ίδιο με τον DECIMAL	
<b>CHAR</b>	Μια καθορισμένου μήκους συμβολοσειρά με μέγιστο πλήθος 255 χαρακτήρες	1-255 bytes
<b>VARCHAR</b>	Μια συμβολοσειρά μεταβλητού-μήκους με μέγιστο πλήθος 255 χαρακτήρες	1-255 bytes
<b>DATE</b>	Ημερομηνία από "1000-01-01" μέχρι "9999-12-31". Ο MySQL την αποθηκεύει με τη μορφή 'YYYY-MM-DD HH:MM:SS'	3 bytes
<b>TIME</b>	Χρόνος με κλίμακα "- 838:59:59" έως "838:59:59". Ο MySQL τον αποθηκεύει με τη 'HH:MM:SS'.	3 bytes
<b>YEAR</b>	Έτος με 2 ή 4 ψηφία. Οι επιτρεπόμενες τιμές είναι από 1901 μέχρι 2155, 0000 με το τετραψήφιο σχήμα έτους, και το 1970-2069 εάν χρησιμοποιείτε το σχήμα 2-ψηφίων	1 byte
<b>TIMESTAMP</b>	Η κλίμακα τιμών είναι από το "1970-01-01 00:00:00" μέχρι κάποια στιγμή του έτους 2037. Ο MySQL την αποθηκεύει με τη μορφή YYYYMMDDHHMMSS ,YYMMDDHHMMSS ,YYYYMMDD, ή YYMMDD	4 bytes
<b>DATETIME</b>	Συνδυασμός ημερομηνίας και χρόνου. Η κλίμακα τιμών είναι από "1000-01-01 00:00:00" μέχρι " 9999-12-31 23:59:59". Ο MySQL τον αποθηκεύει με τη μορφή 'YYYY-MM-DD HH:MM:SS'	8 bytes
<b>TINYBLOB</b>	Κείμενο με μήκος 255 (2 <sup>8</sup> - 1) χαρακτήρες	L+1 bytes, όπου L < 2 <sup>8</sup>
<b>BLOB</b>	Κείμενο με μέγιστο μήκος 255 (2 <sup>8</sup> - 1) χαρακτήρες	L+2 bytes, όπου L < 2 <sup>16</sup>
<b>MEDIUMBLOB</b>	Κείμενο με μέγιστο μήκος 16777215 (2 <sup>24</sup> - 1) χαρακτήρες	L+3 bytes, όπου L < 2 <sup>24</sup>
<b>LOB</b>	Κείμενο με μέγιστο μήκος 4294967295 (2 <sup>32</sup> - 1) χαρακτήρες	L+4 bytes, όπου L < 2 <sup>32</sup>
<b>TINYTEXT</b>	Κείμενο με μέγιστο μήκος 255 (2 <sup>8</sup> - 1) χαρακτήρες	L+1 bytes, όπου L < 2 <sup>8</sup>
<b>TEXT</b>	Κείμενο με μέγιστο μήκος 255 (2 <sup>8</sup> - 1) χαρακτήρες	L+2 bytes, όπου L < 2 <sup>16</sup>
<b>MEDIUMTEXT</b>	Κείμενο με μέγιστο μήκος 16777215 (2 <sup>24</sup> - 1) χαρακτήρες	L+3 bytes, όπου L < 2 <sup>24</sup>
<b>LONGTEXT</b>	Κείμενο με μέγιστο μήκος 4294967295 (2 <sup>32</sup>	L+4 bytes, όπου



	- 1) χαρακτήρες	$L < 2^{32}$
<b>ENUM</b>	Μια απαρίθμηση. Μία σειρά τιμών από την οποία μπορεί να επιλεγεί μόνο μία τιμή. Ένα ENUM μπορεί να έχει ένα μέγιστο πλήθος 65535 τιμών.	1 or 2 bytes ανάλογα με τον αριθμό τιμών της απαρίθμησης
<b>SET</b>	Μία σειρά τιμών ,από τις οποίες κάθε μια πρέπει να επιλεγεί από τον κατάλογο τιμών Ένα SET μπορεί να έχει μέγιστο πλήθος 64 τιμών	1, 2, 3, 4 or 8 bytes ανάλογα με τον αριθμό τιμών του SET

Στον παρακάτω πίνακα οι τύποι δεδομένων του ADO αντιστοιχίζονται με τους τύπους του MySQL .Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για την εισαγωγή αρχείου xml σε βάση MySQL.

<b>Τύπος δεδομένων ADO</b>	<b>Τιμή τύπου ADO</b>	<b>Τύπος δεδομένων MySQL</b>
AdArray	<b>8192</b>	text
adBigInt	<b>20</b>	integer
adBoolean	<b>11</b>	text
adChar	<b>129</b>	text
adCurrency	<b>6</b>	text
adDate	<b>7</b>	text
adDBDate	<b>133</b>	text
adDBTime	<b>134</b>	text
adDBTimeStamp	<b>135</b>	text
adDecimal	<b>14</b>	real
adDouble	<b>5</b>	real
adDUID	<b>72</b>	integer
adInteger	<b>3</b>	integer
adLongVarBinary	<b>205</b>	blob
adLongVarChar	<b>201</b>	text
adLongVarWChar	<b>203</b>	text
adNumeric	<b>131</b>	integer
adSingle	<b>4</b>	real
adSmallInt	<b>2</b>	integer
adTinyInt	<b>16</b>	integer
adUnsignedBigInt	<b>21</b>	integer
adUnsignedInt	<b>19</b>	integer
adUnsignedSmallInt	<b>18</b>	integer
adUnsignedTinyInt	<b>17</b>	integer
adVarBinary	<b>204</b>	blob
adVarChar	<b>200</b>	text
adVarWChar	<b>202</b>	text
adWChar	<b>130</b>	text
adVarNumeric	<b>139</b>	numeric
adBinary	<b>128</b>	blob

## **6.6 MimerSQL**

Η Mimer SQL είναι ένα προηγμένο σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) που αναπτύχθηκε από τη Mimer Information Technology AB.

Η Mimer SQL διαθέτει μοναδικές τεχνικές λύσεις για να χειριστεί μερικές από τις πιο περίπλοκες λειτουργίες που ένα σύστημα διαχείρισης βάσεων δεδομένων πρέπει να παρέχει, όπως το να παρέχει λύση στο πρόβλημα της ταυτόχρονης πρόσβασης σε μία βάση δεδομένων χωρίς τον κίνδυνο αστοχίας. Το παραπάνω διευκολύνει πολύ τη διαχείριση των βάσεων δεδομένων και επιτρέπει καλή απόδοση, ακόμη και κατά τη διάρκεια που το σύστημα είναι βαριά φορτωμένο.

Η Mimer SQL είναι μία καλή πρόταση συστήματος διαχείρισης βάσεων δεδομένων που διαθέτει υποστήριξη πολυεπεξεργαστών και είναι συμβατή με το Unix (συμπεριλαμβανομένου και του Linux), τα Windows NT/2000/XP και τις πλατφόρμες 98/ME

Μια βάση δεδομένων είναι το φυσικό αρχείο όπου μια συλλογή των πινάκων αποθηκεύεται. Μια βάση δεδομένων Mimer SQL μπορεί να περιέχει οποιοδήποτε αριθμό πινάκων δεδομένων.

Υπάρχουν δύο τύποι βάσεων δεδομένων, συστημάτων και χρηστών:

### **Βάσεις δεδομένων συστημάτων**

Οι βάσεις δεδομένων συστημάτων περιέχουν τις πληροφορίες συστημάτων που χρησιμοποιούνται από το διαχειριστή των βάσεων δεδομένων. Αυτές οι βάσεις δεδομένων καθορίζονται όταν δημιουργείται το σύστημα.

Οι βάσεις δεδομένων συστημάτων είναι:

- SYSDb, που περιέχει τους πίνακες ευρετηρίων των στοιχείων
- TRANSDb, που χρησιμοποιείται για τη συναλλαγή των στοιχείων
- LOGDb, που χρησιμοποιείται για την αναγραφή της συναλλαγής
- SQLDb, που χρησιμοποιείται στο χειρισμό της συναλλαγής και για την προσωρινή αποθήκευση των εσωτερικών πινάκων εργασίας.

### **Βάσεις δεδομένων χρηστών**

Οι βάσεις δεδομένων χρηστών περιέχουν τους πίνακες χρηστών. Αυτές οι βάσεις δεδομένων καθορίζονται από το χρήστη ο οποίος είναι αρμόδιος για τη βάση δεδομένων. Εκτός από ειδικές καταστάσεις (όπως κατά τη δημιουργία των πινάκων), οι βάσεις δεδομένων είναι απολύτως αόρατες στο χρήστη.

Η ονοματολογία των πινάκων και των πεδίων ακολουθεί το σχεσιακό μοντέλο που αναφέρθηκε παραπάνω.

Παρακάτω παρουσιάζονται οι τύποι δεδομένων των πεδίων που υποστηρίζονται σε μία βάση για τη MimerSQL

Τύπος δεδομένων	Συνοτμεύσεις	Περιγραφή	Μέγεθος
CHARACTER(n)	CHAR(n)	Σειρά χαρακτήρων σταθερού μήκους $v$ .	$1 \leq v \leq 15.000$
CHARACTER VARYING(n)	CHAR VARYING(n) VARCHAR(n)	Σειρά χαρακτήρων μεταβλητού μήκους, με μέγιστο μήκος $v$ .	$1 \leq v \leq 15.000$
CHARACTER LARGE OBJECT(n[K M G])	CHAR LARGE OBJECT(n[K M G]) CLOB(n[K M G])	Σειρά χαρακτήρων μεταβλητού μήκους που μετριέται σε χαρακτήρες.	
NATIONAL CHARACTER(n)	NATIONAL CHAR(n) NCHAR(n)	Σειρά χαρακτήρων σταθερού μήκους $v$ .	$1 \leq n \leq 5\ 000$
NATIONAL CHARACTER VARYING(n)	NATIONAL CHAR VARYING(n) NCHAR VARYING(n) NVARCHAR(n)	Σειρά χαρακτήρων μεταβλητού μήκους, με μέγιστο μήκος $v$ .	$1 \leq n \leq 5\ 000$
NATIONAL CHARACTER LARGE OBJECT(n[K M G])	NATIONAL CHAR LARGE OBJECT(n[K M G]) NCHAR LARGE OBJECT(n[K M G]) NCLOB(n[K M G])	Σειρά χαρακτήρων μεταβλητού μήκους που μετριέται σε χαρακτήρες	
BINARY(n)	N/A	Δυαδική σειρά σταθερού μήκους με μέγιστο μήκος $v$ .	$1 \leq n \leq 15\ 000$
BINARY VARYING(n)	VARBINARY(n)	Δυαδική σειρά μεταβλητού μήκους, μέγιστο μήκος $v$ .	$1 \leq n \leq 15\ 000$
BINARY LARGE OBJECT(n[K M G])	BLOB(n[K M G])	Δυαδική σειρά μεταβλητού μήκους που μετριέται σε octets.	.
INTEGER(p)	INT(p)	Ακέραιος αριθμός με ακρίβεια.	$1 \leq p \leq 45$
SMALLINT	N/A	Ακέραιος με αριθμητική ακρίβεια 5	-32 768 μέχρι 32 767
INTEGER	INT	Ακέραιος με αριθμητική ακρίβεια 10.	- 2 147 483 648 μέχρι 2 147 483 648

			7
BIGINT	N/A	Ακέραιος με αριθμητική ακρίβεια 19	- 9 223 372 03 6 854 775 80 8 μέχρι 9 223 372 03 6 854 775 80 7
DECIMAL(p,s)	DEC(p,s)	Exact numerical, precision p, scale s.	$1 \leq p \leq 45$ $0 \leq s \leq p$
NUMERIC(p,s)	N/A	Αριθμός με ακρίβεια π, και κλίμακα s. (Όπως ο δεκαδικός).	$1 \leq p \leq 45$ $0 \leq s \leq p$
FLOAT(p)	N/A	Κατά προσέγγιση αριθμητικός.	$1 \leq p \leq 45$ Μηδέν ή απόλυτο $10^{-999}$ to $10^{+999}$
REAL	N/A	Κατά προσέγγιση αριθμητικός με ακρίβεια 7.	Μηδέν ή απόλυτο $10^{-38}$ μέχρι $10^{+38}$
FLOAT	N/A	Κατά προσέγγιση αριθμητικός με ακρίβεια 16.	Μηδέν ή απόλυτο $10^{-38}$ μέχρι $10^{+38}$
DOUBLE PRECISION	N/A	Κατά προσέγγιση αριθμητικός με ακρίβεια 16.	Μηδέν ή απόλυτο $10^{-38}$ μέχρι $10^{+38}$
DATE TIME(s) TIMESTAMP(s)	Αποτελούνται από ακέραιες τιμές. Αντιπροσωπεύουν ημέρα ή ώρα ανάλογα		

Στον παρακάτω πίνακα οι τύποι δεδομένων του ADO αντιστοιχίζονται με τους τύπους της MimerSQL. Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για την εισαγωγή αρχείου xml σε βάση MimerSQL.

Τύπος δεδομένων ADO	Τιμή τύπου ADO	Τύπος δεδομένων MimerSQL
AdArray	8192	clob(n[k,m,g])
adBigInt	20	bigint
adBoolean	11	odbc.bit
adChar	129	character(n)

adCurrency	6	decimal(19,4)
adDate	7	date
adDBDate	133	date
adDBTime	134	time(s)
adDBTimeStamp	135	timestamp(s)
adDecimal	14	decimal(p,c)
adDouble	5	double precision or float
adDUID	72	odbc.guid
adInteger	3	integer or int
adLongVarBinary	205	binary varying(n)
adLongVarChar	201	clob(n[k,m,g])
adLongVarWChar	203	nclob(n[k,m,g])
adNumeric	131	numeric(p,s)
adSingle	4	real
adSmallInt	2	odbc.tinyint
adTinyInt	16	odbc.tinyint
adUnsignedBigInt	21	bigint
adUnsignedInt	19	integer or int
adUnsignedSmallInt	18	smallint
adUnsignedTinyInt	17	odbc.tinyint
adVarBinary	204	varbinary(n)
adVarChar	200	character varying
adVarWChar	202	national character(n)
adWChar	130	national character varying(n)
adVarNumeric	139	numeric(p,s)
adBinary	128	binary(n)

## **6.7 Η SQLite3**

Η SQLite είναι εικονικό σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (δεν χρειάζεται να εκτελείται ξεχωριστή διεργασία του Server)(δηλαδή λειτουργεί με κλήσεις API σε ένα αρχείο DLL) που είναι αποθηκευμένο σε μια σχετικά μικρή βιβλιοθήκη C. Δημιουργήθηκε από τον D. Richard Hipp.

Στην SQLite2 οι εγγραφές αποθηκεύονται ως κείμενο ASCII. Η έκδοση 3 ενισχύει τα παραπάνω παρέχοντας τη δυνατότητα να αποθηκεύονται οι ακέραιοι και πραγματικοί αριθμοί με ένα συμπαγέστερο σχήμα ,ενώ επιτρέπει και την αποθήκευση τύπων Blob.

Μια βάση της SQLite3 είναι ένα αρχείο με επέκταση “.db”.Περιορισμοί σχετικά με το πλήθος χαρακτήρων στα ονόματα του πίνακα και των πεδίων του δεν υπάρχουν.

Οι εγγραφές σε έναν πίνακα της SQLite3 αποθηκεύονται σύμφωνα με έναν από τους παρακάτω τύπους:

<b>Τύπος</b>	<b>Περιγραφή</b>
NULL	Η εγγραφή είναι κενή

<b>INTEGER</b>	Η εγγραφή είναι ακέραιος αριθμός, που αποθηκεύεται σε 1 ..2 ..3 ..4 ..6, ή 8 bytes ανάλογα με το μέγεθος της εγγραφής
<b>REAL</b>	Η εγγραφή είναι κινητής υποδιαστολής, που αποθηκεύεται ως 8-byte αριθμός κινητής υποδιαστολής
<b>TEXT</b>	Η εγγραφή είναι κείμενο, που αποθηκεύεται χρησιμοποιώντας την κωδικοποίηση βάσεων δεδομένων (UTF-8, UTF-16BE ή UTF-16-LE)
<b>BLOB</b>	Η εγγραφή αποθηκεύεται ακριβώς όπως εισήχθη

Όπως στην SQLite2, και στην SQLite3 οποιοδήποτε πεδίο στη βάση δεδομένων ,εκτός από έναν INTEGER σαν πρωτεύον κλειδί ,μπορεί να αποθηκεύσει οποιοδήποτε εγγραφή.

Οι κατηγορίες αποθήκευσης ορίζονται αρχικά ως εξής:  
 Εγγραφές που είναι τμήμα εντολών SQL ,θεωρούνται **ΚΕΙΜΕΝΟ** εάν εσωκλείονται από τα ενιαία ή διπλά quotes, **ΑΚΕΡΑΙΟΣ** αν είναι αριθμός χωρίς το δεκαδικό μέρος ή τον εκθέτη, **ΠΡΑΓΜΑΤΙΚΟΣ** εάν είναι αριθμός με ένα δεκαδικό μέρος ή έναν εκθέτη και **ΜΗΔΕΝΙΚΟΣ** εάν η εγγραφή είναι κενή.

Στην SQLite3, ο τύπος μιας εγγραφής συνδέεται με η ίδια την εγγραφή και όχι με το πεδίο ή τη μεταβλητή στις οποίες η εγγραφή αποθηκεύεται (αυτό καλείται manifest typing.) σε αντίθεση με άλλες μηχανές βάσεων δεδομένων SQL που γνωρίζουμε. Δηλαδή σε μία στήλη ενός πίνακα ,μπορούμε να αποθηκεύσουμε δεδομένα από όλους τους τύπους δεδομένων.

Προκειμένου να μεγιστοποιηθεί η συμβατότητα μεταξύ της SQLite και των άλλων μηχανών βάσεων δεδομένων, υποστηρίζεται η έννοια της "συγγένειας τύπων" στα πεδία. Με τη "συγγένεια τύπων" ενός πεδίου ,εννοούμε τον καταλληλότερο τύπο δεδομένων για τα στοιχεία που αποθηκεύονται στο πεδίο. Το κλειδί στην SQLite είναι ότι ο τύπος προτείνεται, δεν απαιτείται. Θεωρητικά όμως πάντα το πεδίο μπορεί να αποθηκεύσει οποιοδήποτε τύπο δεδομένων. Λαμβάνοντας υπόψη όμως αυτή τη δυνατότητα επιλογής ,τα πεδία θα προτιμήσουν να χρησιμοποιήσουν κάποιον τύπο δεδομένων έναντι κάποιου άλλου. Ο προτιμότερος τύπος αποθήκευσης για ένα πεδίο είναι αυτό που καλείται "συγγένεια".

Σε κάθε πεδίο βάσης δεδομένων της SQLite3, ορίζεται μια από τις ακόλουθες συγγένειες τύπων:

- ΚΕΙΜΕΝΟ
- ΑΡΙΘΜΟΣ
- ΑΚΕΡΑΙΟΣ ΑΡΙΘΜΟΣ
- NONE

Ένα πεδίο με τη συγγένεια ΚΕΙΜΕΝΩΝ αποθηκεύει όλα τα στοιχεία χρησιμοποιώντας τους τύπους NULL, ΚΕΙΜΕΝΟ ή BLOB. Εάν το αριθμητικό στοιχείο παρεμβάλλεται σε ένα πεδίο με τη συγγένεια ΚΕΙΜΕΝΩΝ μετατρέπεται σε κείμενο πριν αποθηκευτεί.

Ένα πεδίο με ΑΡΙΘΜΗΤΙΚΗ συγγένεια ,μπορεί να περιέχει εγγραφές χρησιμοποιώντας και τους 5 τύπους αποθήκευσης. Όταν κείμενο παρεμβάλλεται σε

πεδίο ΑΡΙΘΜΟΥ, γίνεται μια προσπάθεια να το μετατρέψει σε ακέραιο ή έναν πραγματικό αριθμό προτού να το αποθηκευτεί. Εάν η μετατροπή είναι επιτυχής, η εγγραφή αποθηκεύεται σαν ΑΚΕΡΑΙΟΣ ή ΠΡΑΓΜΑΤΙΚΟΣ. Εάν η μετατροπή δεν μπορεί να εκτελεσθεί, η εγγραφή αποθηκεύεται σαν ΚΕΙΜΕΝΟ. Καμία προσπάθεια δεν γίνεται για μετατροπή των τιμών NULL ή τιμών BLOB.

Ένα πεδίο που χρησιμοποιεί τη συγγένεια ΑΚΕΡΑΙΩΝ ΑΡΙΘΜΩΝ συμπεριφέρεται με τον ίδιο τρόπο όπως ένα πεδίο με την ΑΡΙΘΜΗΤΙΚΗ συγγένεια

Ένα πεδίο με τη συγγένεια NONE δεν προτιμά καμία κατηγορία αποθήκευσης. Δεν επιχειρείται καμία προσπάθεια μετατροπής των εγγραφών προτού αυτές εισαχθούν.

Στον παρακάτω πίνακα οι τύποι δεδομένων του ADO αντιστοιχίζονται με τους τύπους της SQLite3. Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για τη δημιουργία πίνακα της SQLite3 από αρχείο xml.

Τύπος δεδομένων ADO	Τιμή τύπου ADO	Τύπος δεδομένων SQLite3
AdArray	8192	text
adBigInt	20	integer
adBoolean	11	text
adChar	129	text
adCurrency	6	text
adDate	7	text
adDBDate	133	text
adDBTime	134	text
adDBTimeStamp	135	text
adDecimal	14	real
adDouble	5	real
adDUID	72	integer
adInteger	3	integer
adLongVarBinary	205	blob
adLongVarChar	201	text
adLongVarWChar	203	text
adNumeric	131	integer
adSingle	4	real
adSmallInt	2	integer
adTinyInt	16	integer
adUnsignedBigInt	21	integer
adUnsignedInt	19	integer
adUnsignedSmallInt	18	integer
adUnsignedTinyInt	17	integer
adVarBinary	204	blob
adVarChar	200	text
adVarWChar	202	text
adWChar	130	text
adVarNumeric	139	real
adBinary	128	blob

## 6.8 Η Oracle

Το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων Oracle σχεδιάστηκε για να επιτρέψει την ταυτόχρονη πρόσβαση μεγάλες βάσεις δεδομένων.

Η Oracle είναι από τα πιο σταθερά συστήματα διαχείρισης βάσεων δεδομένων και προτιμάται για τις περιπτώσεις που αποθηκεύονται δυαδικά δεδομένα στη βάση

Η βάση δεδομένων διαιρείται σε ένα ή περισσότερα λογικά κομμάτια που είναι γνωστά ως **tablespaces**. Ένα tablespace χρησιμοποιείται για να συγκεντρώσει τα δεδομένα.

- Το μέγιστο μέγεθος ενός datafile είναι 32GB (gigabytes).
- Ο μέγιστος αριθμός datafiles ανά tablespace είναι 1,022.
- Το μέγιστο μέγεθος ενός tablespace είναι 32TB (terabyte).

Η ονοματολογία των πινάκων και των πεδίων αυτών ακολουθεί τους κανόνες του σχεσιακού μοντέλου που αναφέρθηκε παραπάνω.

Οι τύποι δεδομένων που υποστηρίζονται από την Oracle για την δημιουργία των πινάκων είναι:

<b>Τύπος</b>	<b>Περιγραφή</b>	<b>Μέγεθος</b>
<b>Char</b>	Καθορισμένου μήκους χαρακτήρας	Οποιοδήποτε μήκος μεταξύ 1 και 255 bytes
<b>Date</b>	Αποθηκεύει έτη, μήνες, ημέρες, ώρες, λεπτά και δευτερόλεπτα	
<b>Long</b>	Σειρά χαρακτήρων μεταβλητού-μήκους μέχρι 2GB στο μέγεθος.	
<b>LongRaw</b>	Παρόμοιος με το Long	
<b>Mlslabel</b>	Για τη χρήση εμπιστευτικής Oracle, αυτός ο τύπος στοιχείων αποθηκεύει το δυαδικό σχήμα της ετικέτας συστημάτων.	
<b>Number</b>	Αποθηκεύει τους σταθερούς και floating-point αριθμούς. Έχει ακρίβεια και κλίμακα.	
<b>RawID</b>	Αποθηκεύει μια τριπλέτα αριθμών	
<b>Varchar</b>	Σύνολο χαρακτήρων μεταβλητού-μήκους	2 <sup>31</sup> - 1 bytes το μέγιστο
<b>Varchar2</b>	Σύνολο χαρακτήρων μεταβλητού-μήκους που αποθηκεύει 1-2.000 χαρακτήρες.	

Στον παρακάτω πίνακα οι τύποι δεδομένων του **ADO** αντιστοιχίζονται με τους τύπους δεδομένων Oracle. Η αντιστοίχιση αυτή θα χρησιμοποιηθεί αργότερα στην ανάπτυξη του προγράμματος για την εισαγωγή αρχείου xml σε βάση Oracle



<b>Τύπος δεδομένων ADO</b>	<b>Τιμή τύπου ADO</b>	<b>Τύπος δεδομένων Oracle</b>
AdArray	<b>8192</b>	varchar2(4000)
adBigInt	<b>20</b>	number
adBoolean	<b>11</b>	number
adChar	<b>129</b>	char
adCurrency	<b>6</b>	number
adDate	<b>7</b>	date
adDBDate	<b>133</b>	date
adDBTime	<b>134</b>	date
adDBTimeStamp	<b>135</b>	date
adDecimal	<b>14</b>	number
adDouble	<b>5</b>	float
adDUID	<b>72</b>	urawid
adInteger	<b>3</b>	number(10)
adLongVarBinary	<b>205</b>	long raw
adLongVarChar	<b>201</b>	varchar(4000)/long clob/long
adLongVarWChar	<b>203</b>	nclob
adNumeric	<b>131</b>	number
adSingle	<b>4</b>	float
adSmallInt	<b>2</b>	number
adTinyInt	<b>16</b>	number
adUnsignedBigInt	<b>21</b>	number
adUnsignedInt	<b>19</b>	number
adUnsignedSmallInt	<b>18</b>	number
adUnsignedTinyInt	<b>17</b>	number
adVarBinary	<b>204</b>	blob
adVarChar	<b>200</b>	varchar2
adVarWChar	<b>202</b>	varchar2
adWChar	<b>130</b>	nchar
adVarNumeric	<b>139</b>	number
adBinary	<b>128</b>	raw

## 7. ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ – ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΑΝΑΛΥΣΗ

Στις παραπάνω ενότητες έγινε μία περιήγηση στα συστήματα σχεσιακών βάσεων δεδομένων. Αναλύθηκαν οι δομές τους και το πως αυτά δουλεύουν για την σωστή διαχείριση πληροφοριών. Ακόμη έγινε αναφορά στη σημειακή γλώσσα XML (eXtensible Markup Language) και στη μεγάλη χρησιμότητά της για τα συστήματα διαχείρισης σχεσιακών βάσεων. Περιγράφηκαν τα σημαντικότερα εργαλεία της Visual Basic 6.0 με τα οποία μπορεί κανείς να διαχειριστεί τις βάσεις δεδομένων και να εξάγει αρχεία XML από τους πίνακες των βάσεων αυτών.

Παρακάτω θα γίνει ανάλυση του λογισμικού που αναπτύχθηκε με σκοπό την ανταλλαγή δεδομένων μέσω μετατροπής σε XML, από και προς αρχεία σχεσιακών βάσεων δεδομένων. Θα περιγραφούν οι φόρμες του Project και θα γίνει εκτενή αναφορά στα αντικείμενα αυτών. Θα παρουσιαστεί ο κώδικας των αντικειμένων αυτών και θα γίνει ανάλυση αυτού.

Να σημειωθεί επίσης ότι η αναφορά στα αντικείμενα αυτά θα γίνει όχι με το πραγματικό τους όνομα (π.χ. CommandButton “**CommandButton1**”), αλλά με την ετικέτα που έχουν αυτά όταν εμφανίζονται στις φόρμες (π.χ. CommandButton “**Select Table**”).

### 7.1 Η ΦΟΡΜΑ1

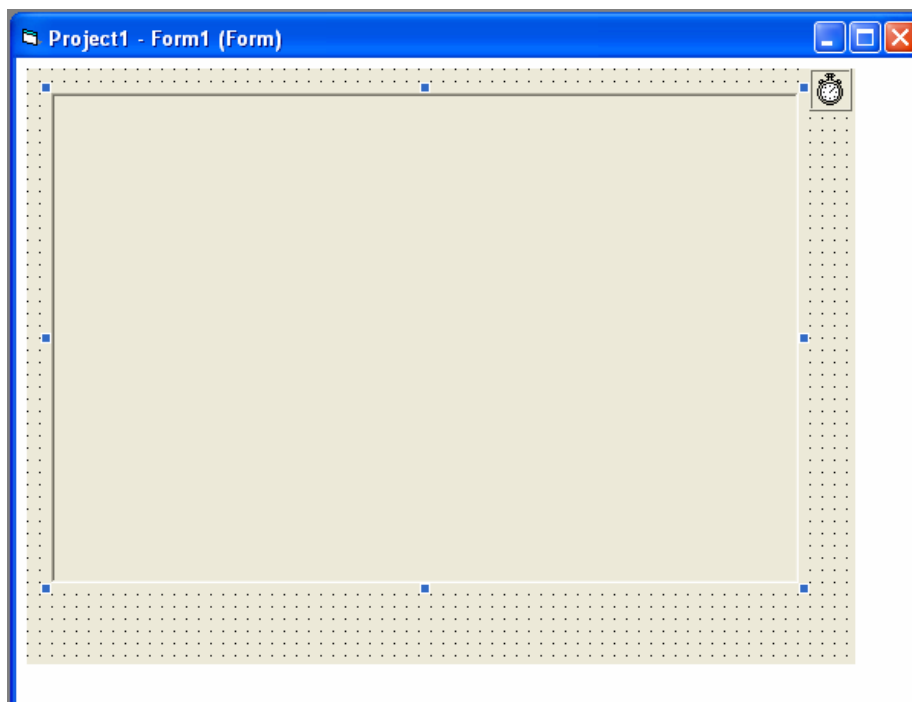
Η φόρμα1, είναι η φόρμα εισαγωγής στο πρόγραμμα. Περιέχει μόνο δύο αντικείμενα, ένα PictureBox και έναν Timer. Στο PictureBox, εισάγεται η εικόνα που εμφανίζεται με την εκτέλεση του αρχείου “PerfectXML.exe”. Η διάρκεια της εμφάνισης της εικόνας και γενικά της φόρμας1 ρυθμίζεται από το αντικείμενο Timer το οποίο δεν είναι ορατό στον χρήστη κατά την εκτέλεση του προγράμματος.

Ρυθμίζοντας την ιδιότητα Interval του Timer σε 2800(mseconds), προγραμματίζουμε την εξαφάνιση της φόρμας1 και παράλληλα την εμφάνιση της φόρμας2 (“**Menu Window**”) σε 2,8 δευτερόλεπτα από την στιγμή εκκίνησης του προγράμματος.

Ο αντίστοιχος κώδικας φαίνεται παρακάτω ενώ ακολουθούν οι εικόνες της φόρμας1 που την περιγράφουν πριν(εικόνα Form1a) και κατά την εκτέλεση της εφαρμογής (εικόνα Form1b).

#### “Timer” Code

```
Private Sub Timer1_Timer()  
'Εξφόρτωση και εξαφάνιση της φόρμας1  
Unload Form1  
'Φόρτωση και εμφάνιση της φόρμας2  
Load Form2  
Form2.Show  
End Sub
```



(εικόνα Form1a)



(εικόνα Form1b)

## 7.2 Η ΦΟΡΜΑ “Menu Window”

Η φόρμα2(“Menu Window”),είναι φόρμα επιλογών ενεργειών ή φόρμα μενού για το πρόγραμμα.Στη φόρμα αυτή ο χρήστης επιλέγει τι θέλει να κάνει. Η φόρμα περιέχει πέντε CommandButtons, όσες και οι δυνατές επιλογές.Η μορφή της φόρμας2 φαίνεται στην εικόνα Form2.



(εικόνα Form2)

Τα πρώτα τέσσερα CommandButtons “Export XML From Table”, “Export XML From Two Tables” , “Import XML” και “View Data” εμφανίζουν τις ομώνυμες φόρμες ,ενώ το CommandButton “Exit” εκτελεί έξοδο από το πρόγραμμα .

Οι κώδικες των CommandButtons διατίθενται παρακάτω:

### “Exit” CommandButton

```
Private Sub Exit_Click()  
'Ξεφόρτωση και εξαφάνιση της φόρμας2  
Unload Form2  
End Sub
```

### “Export XML From Table” CommandButton

```
Private Sub ExportXML_Click()  
'Ξεφόρτωση και εξαφάνιση της φόρμας2  
Unload Form2  
'Φόρτωση και εμφάνιση της φόρμας3  
Load Form3  
Form3.Show
```

```
End Sub
```

### **“Export XML From Two Tables” CommandButton**

```
Private Sub ExportXMLTables_Click()  
'Ξεφόρτωση και εξαφάνιση της φόρμας2  
Unload Form2  
'Φόρτωση και εμφάνιση της φόρμας5  
Load Form5  
Form5.Show  
End Sub
```

### **“Import XML” CommandButton**

```
Private Sub ImportXML_Click()  
'Ξεφόρτωση και εξαφάνιση της φόρμας2  
Unload Form2  
'Φόρτωση και εμφάνιση της φόρμας4  
Load Form4  
Form4.Show  
End Sub
```

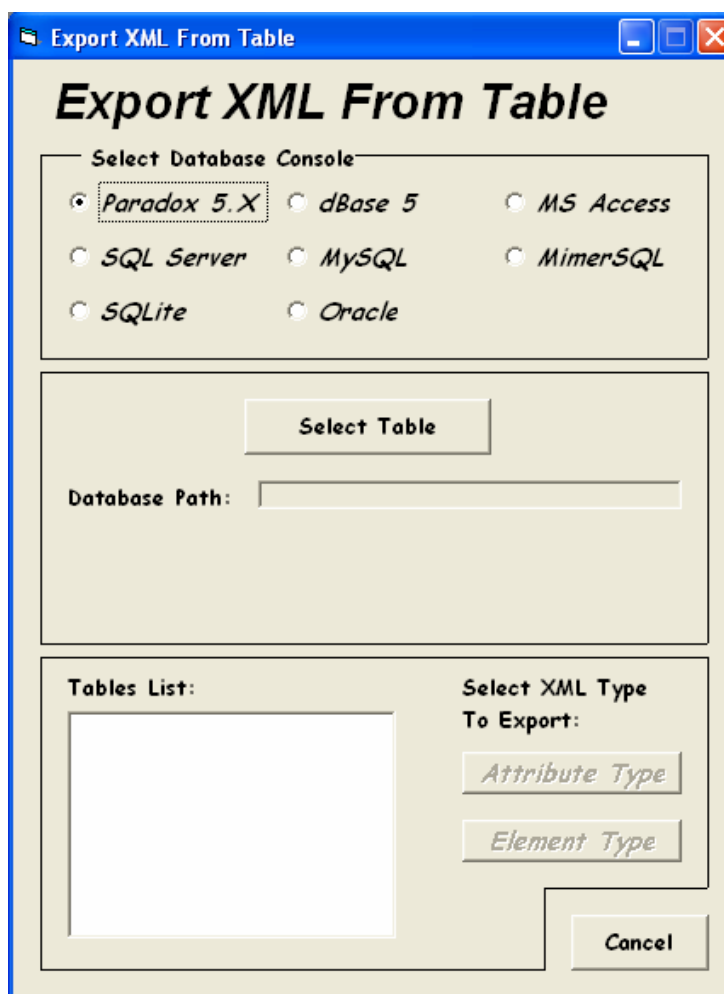
### **“View Data” CommandButton**

```
Private Sub ViewData_Click()  
'Ξεφόρτωση και εξαφάνιση της φόρμας2  
Unload Form2  
'Φόρτωση και εμφάνιση της φόρμας7  
  
Load Form7  
Form7.Show  
End Sub
```

### 7.3 Η ΦΟΡΜΑ “Export XML From Table”

Στη φόρμα (“Export XML From Table”) ο χρήστης εξάγει τον επιθυμητό πίνακα δεδομένων σε αρχείο τύπου “.xml”. Το πρόγραμμα δίνει τη δυνατότητα σύνδεσης με οποιονδήποτε απ’τους οχτώ τύπους βάσεων δεδομένων, για τους οποίους έγινε αναφορά παραπάνω, καθώς και δυνατότητα επιλογής του πίνακα εξαγωγής μέσα από μία λίστα πινάκων.

Πιο συγκεκριμένα, η φόρμα για τις περιπτώσεις της **MS\_Access** , **Paradox\_5.X** , **Dbase\_5** και **SQLite** έχει τη μορφή της εικόνας **Form3a**.

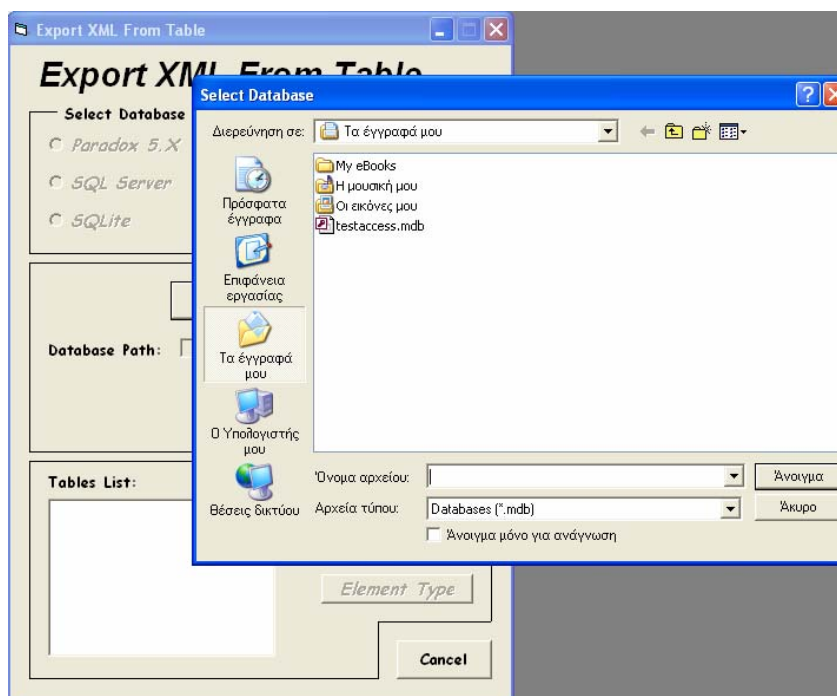


(εικόνα **Form3a**)

Στο πρώτο κομμάτι της φόρμας υπάρχουν οχτώ **OptionButtonControls** εκ των οποίων καθένα αντιστοιχεί και σε ένα τύπο βάσεων δεδομένων. Ανάλογα με το ποιο θα επιλέξει ο χρήστης κατά, την εκτέλεση του προγράμματος, θα δουλέψει και με τον αντίστοιχο τύπο βάσης. Τα πρώτα τρία έχουν αντιστοιχία ως εξής:

Option1 → Paradox 5.X
Option2 → Dbase 5
Option3 → MS Access
Option7 → SQLite

Στο μεσαίο κομμάτι της φόρμας<sup>3</sup>, γίνεται η επιλογή του επιθυμητού πίνακα για τις περιπτώσεις **Paradox\_5.X** και **Dbase\_5**, όπου μπορούμε να συνδεθούμε απευθείας στον πίνακα μέσω ενός `CommonDialogControl` που εμφανίζεται με το πάτημα του `CommandButton` “**Select Table**” (εικόνα **Form3c**). Το path του πίνακα εμφανίζεται σε αντίστοιχο `Label` με ετικέτα “**Table Path:**” και το όνομα του πίνακα φορτώνεται αυτομάτως στο `Listbox` για να μπορέσει ο χρήστης να το χρησιμοποιήσει στη συνέχεια.



(εικόνα **Form3c**)

Στην περίπτωση των **MS\_Access** και **SQLite** ο χρήστης συνδέεται πρώτα στη βάση δεδομένων “.mdb” ή “.db”, την οποία επιλέγει και πάλι μέσω του `CommonDialogControl` το οποίο εμφανίζεται ξανά με το πάτημα του ίδιου κουμπιού, του οποίου η ετικέτα τώρα είναι “**Select Database**”, ενώ του αντίστοιχου `Label` για το Path της βάσης είναι “**Database Path:**”. Αυτομάτως και πάλι, οι πίνακες της συνδεδεμένης βάσης δεδομένων φορτώνονται στο `Listbox` της φόρμας<sup>3</sup> για να επιλέξει ο χρήστης, ποιον θα εξάγει.

Τα υπόλοιπα τέσσερα `OptionButtonControls` αντιστοιχούν σε σχεσιακές βάσεις δεδομένων τύπου Client/Server. Πιο συγκεκριμένα αντιστοιχούν ως εξής:

Option4 → MS SQL Server
Option5 → MySQL
Option6 → MimerSQL
Option8 → Oracle 9i

Η φόρμα<sup>3</sup> για τις περιπτώσεις των **MS\_SQL\_Server**, **MySQL** και **MimerSQL** και **Oracle\_9i** έχει τη μορφή της εικόνας **Form3b**.

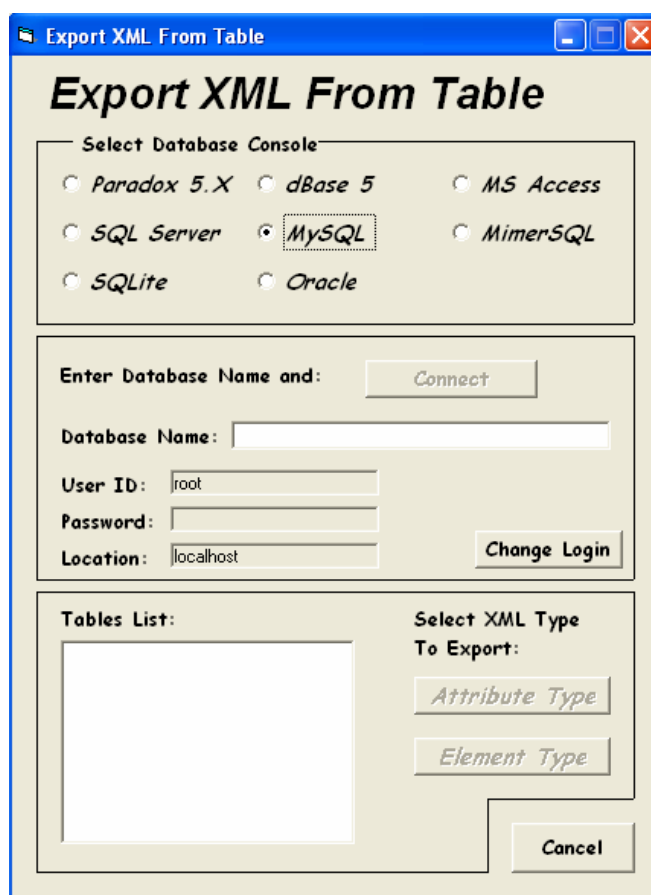
Το `CommandButton` “**Select Table**” ή “**Select Database**” δεν είναι ενεργό πλέον, όπως και το αντίστοιχο `Label`. Το `CommandButton` “**Connect**” εμφανίζεται για

να πραγματοποιηθεί η σύνδεση στο διακομιστή της βάσης. Το όνομα της βάσης για την σύνδεση εισάγεται από το χρήστη στο αντίστοιχο TextBox ,ενώ τα υπόλοιπα στοιχεία (**UserID,Password,Server**) παρουσιάζονται στα αντίστοιχα Labels.

Η δυνατότητα αλλαγής τους δίνεται στο χρήστη μέσω του CommandButton “**Change Login**” ,το οποίο εμφανίζει την φόρμα για εισαγωγή των νέων στοιχείων. Αμέσως μόλις πραγματοποιηθεί η σύνδεση τα ονόματα των πινάκων της βάσης φορτώνονται στο ListBox της φόρμας για να επιλέξει ο χρήστης ,ποιον τελικά θα εξάγει.

Να αναφερθεί ότι, με την ολοκλήρωση οποιασδήποτε συνδέσεως και για τους οχτώ τύπους βάσεων, τα OptionButtonsControls ,όπως και τα “**Select Table**” και “**Connect**” CommandButtons γίνονται ανενεργά, για να αποφευχθούν αστοχίες κατά την εκτέλεση του προγράμματος, εξαιτίας λανθασμένων χειρισμών του χρήστη. Επίσης, τα CommandButtons “**Attribute Type**” και “**Element Type**” γίνονται ενεργά μόνο μετά την επιλογή κάποιου από τους πίνακες της λίστας.

Με αυτό τον τρόπο, επιτυγχάνεται μια σταθερή σειρά ενεργειών που πρέπει να εκτελεστούν ,για να πάρει ο χρήστης το επιθυμητό αποτέλεσμα. Τα βήματα που πρέπει να κάνει είναι ξεκάθαρα, αφού μόνο τα απαραίτητα Buttons και Controls είναι ενεργά.



(εικόνα Form3b)

Όπως αναφέρθηκε παραπάνω, τα CommandButtons “**Attribute Type**” και “**Element Type**” γίνονται ενεργά μόνο μετά την επιλογή κάποιου από τους πίνακες του ListBox. Ο χρήστης έχει τη δυνατότητα να επιλέξει τον τύπο εξαγωγής του αρχείου “.xml”, αν θα είναι δηλαδή Attribute Type (structure and data) ή Element



Type (only data) XML File. Ακόμη, μέσω του CommonDialogControl καθορίζει το σημείο αποθήκευσης του αρχείου στο σκληρό δίσκο.

Τέλος, το “Cancel” CommandButton επιτρέπει την έξοδο κάθε στιγμή από την φόρμα3 και την επιστροφή στο μενού της φόρμας2.

Παρακάτω παρουσιάζεται ο κώδικας των CommandButtons, που αναφέρθηκαν παραπάνω.

### **“Select Table” ή “Select Database” CommandButton**

```
Private Sub SelectDatabase_Click()  
  
'Απενεργοποίηση των OptionButtons για αποφυγή λαθών  
Option1.Enabled = False  
Option2.Enabled = False  
Option3.Enabled = False  
Option4.Enabled = False  
Option5.Enabled = False  
Option6.Enabled = False  
Option7.Enabled = False  
Option8.Enabled = False  
  
'Σε περίπτωση λάθους μεταπήδηση στον διαχειριστή λάθους  
On Error GoTo NoDatabase  
  
'Το μήνυμα λάθους, σε περίπτωση που θα πατηθεί το 'Cancel' Button του  
'CommonDialogControl ,δεν εμφανίζεται  
CommonDialog1.CancelError = True  
  
'Ορίζονται φίλτρα επεκτάσεων για τους διαφορετικούς τύπους βάσεων του  
'CommonDialogControl  
If Option1.Value = True or Option7.Value = True Then  
    CommonDialog1.Filter = "(*.db)|*.db" 'dbase ή sqlite  
    CommonDialog1.DialogTitle = "Select Table"  
Elseif Option2.Value = True Then 'dbase  
    CommonDialog1.Filter = "Tables (*.dbf)|*.dbf"  
    CommonDialog1.DialogTitle = "Select Table"  
Elseif Option3.Value = True Then 'ms_access  
    CommonDialog1.Filter = "Databases (*.mdb)|*.mdb"  
    CommonDialog1.DialogTitle = "Select Database"  
End If  
  
'Άνοιγμα και εμφάνιση στη φόρμα3 του CommonDialogControl  
CommonDialog1.ShowOpen  
  
'Αν επιλεγεί βάση  
If CommonDialog1.FileName <> "" Then  
  
Dim MyLen1, MyLen2 As Integer
```

```

Dim MyStr1, MyStr2 As String

'Mήκος σε χαρακτήρες του επιλεγμένου πίνακα-βάσης
MyLen1 = Len(CommonDialog1.FileTitle)
'Mήκος σε χαρακτήρες του Path του επιλεγμένου πίνακα-βάσης
MyLen2 = Len(CommonDialog1.FileName)

'Αν είναι επιλεγμένο το Option2 (dBASE 5)
If Option2.Value = True Then
    'Επιστροφή του ονόματος του πίνακα χωρίς “.dbf”
    MyStr1 = Left(CommonDialog1.FileTitle, MyLen1 - 4)
    'Επιστροφή του Path του πίνακα χωρίς το όνομα αυτού
    MyStr2 = Left(CommonDialog1.FileName, MyLen2 - MyLen1)
    'Φόρτωση στη λίστα του ονόματος του πίνακα
    List1.AddItem MyStr1
    'Κλήση συνάρτησης ConnectDB
    Call ConnectDB
    'Σε περίπτωση λάθους στη συνάρτηση ,έξοδος από τον κώδικα του
    'CommandButton και επιστροφή στη φόρμα2
    If conf = 1 Then
        Unload Form3
        Form3.Hide
        Form2.Show
        Exit Sub
    End If

'Αν είναι επιλεγμένο το Option1 (Paradox 5.X)
'Επανάληψη διαδικασιών όπως και για dBASE 5
ElseIf Option1.Value = True Or Option2.Value = True Then
    MyStr1 = Left(CommonDialog1.FileTitle, MyLen1 - 3) 'the same as above
    MyStr2 = Left(CommonDialog1.FileName, MyLen2 - MyLen1) 'the same as above
    List1.AddItem MyStr1
    Call ConnectDB
    If conf = 1 Then
        Unload Form3
        Form3.Hide
        Form2.Show
        Exit Sub
    End If

'Αν είναι επιλεγμένο το Option3 (MS_Access) ή το Option3 (SQLite) κλήση
'sυνάρτησης ConnectDB
ElseIf Option3.Value = True Or Option7.Value = True Then
    Call ConnectDB
    'Σε περίπτωση λάθους στη συνάρτηση ,έξοδος από τον κώδικα του
    'CommandButton και επιστροφή στη φόρμα2
    If conf = 1 Then
        Unload Form3
        Form3.Hide
        Form2.Show
    End If

```

```

Exit Sub
End If
Call loadList1

End If

'Εμφάνιση Path του επιλεγμένου πίνακα-βάσης
Label1.Caption = CommonDialog1.FileName
'Απενεργοποίηση του 'Select Table' CommandButton
SelectDatabase.Enabled = False

End If

'Διαχειριστής λάθους για την περίπτωση που δεν επιλεγεί βάση ή πίνακας
NoDatabase:
If Err.Number = 32755 Then 'if cancel error occur
If Option3.Value = True Then
MsgBox "Please select (*.mdb) database!"
ElseIf Option7.Value = True Then
MsgBox "Please select (*.db) database!"
ElseIf Option1.Value = True Then
MsgBox "Please select (*.db) table!"
ElseIf Option2.Value = True Then
MsgBox "Please select (*.dbf) table!"
End If
Exit Sub
End If
End Sub

```

### **“Connect” CommandButton**

```

Private Sub Connect_Click()

'κλήση συνάρτησης ConnectDB
Call ConnectDB

'Σε περίπτωση λάθους στη συνάρτηση ,έξοδος από τον κώδικα του
'CommandButton και επιστροφή στη φόρμα2
If conf = 1 Then
Unload Form3
Form3.Hide
Form2.Show
Exit Sub
Else

'κλήση συνάρτησης loadList1
Call loadList1

```

'Απενεργοποίηση των OptionButtons, του TextBox και των "Connect", "Change Login" CommandButtons για αποφυγή λαθών

```
Text1.Enabled = False  
Connect.Enabled = False  
Command2.Enabled = False  
Option1.Enabled = False  
Option2.Enabled = False  
Option3.Enabled = False  
Option4.Enabled = False  
Option5.Enabled = False  
Option6.Enabled = False  
Option7.Enabled = False  
Option8.Enabled = False
```

```
End If  
End Sub
```

### 'ConnectDB' function

Η συνάρτηση ConnectDB ανοίγει τις συνδέσεις μεταξύ του προγράμματος και των διακομιστών των βάσεων ,που αυτό υποστηρίζει. Τα στοιχεία της κάθε σύνδεσης (**DatabaseName, UserID, Password, ServerName**) εμφανίζονται στη φόρμα στα αντίστοιχά Labels. Ο χρήστης μπορεί να τα αλλάξει, μέσω του CommandButton "Change Log".

Η συνάρτηση καλείται στους κώδικες των CommandButtons "Select Table" και "Connect" για να γίνουν οι απαραίτητες συνδέσεις. Σε περίπτωση αποτυχημένης συνδέσεως εμφανίζεται μήνυμα λάθους και ακολουθεί έξοδος από τη συνάρτηση. Αυτομάτως ,ο δείκτης "conf" παίρνει την τιμή '1' για να σταματήσει η εκτέλεση του κώδικα των CommandButtons "Select Table" και "Connect" και να γίνει επιστροφή στην φόρμα2.

Function ConnectDB()

'Σε περίπτωση λάθους στη συνάρτηση μεταπήδηση στην επόμενη γραμμή κώδικα  
On Error Resume Next

Set conn = New ADODB.Connection

If Option1.Value = True Then 'paradox

'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα 'στοιχεία σύνδεσης

```
conn.Open "Driver={Microsoft Paradox Driver (*.db)};" & _  
    "DriverID=538;" & _  
    "Fil=Paradox 5.X;" & _  
    "DefaultDir=" & MyStr2 & ";" & _  
    "Dbq=" & MyStr2 & ";" & _  
    "CollatingSequence=ASCII"
```

'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη

```

'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table"
'και "Connect"
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection with table failed!Table '" &
CommonDialog1.FileTitle & "' doesn't exist,or is not supported.Operation now will
stop!Start again from the beginning!"
    Exit Function
End If

ElseIf Option2.Value = True Then 'dbase
'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα
'στοιχεία σύνδεσης
conn.Open "Driver={Microsoft dBASE Driver (*.dbf)};DriverID=277;Dbq=" &
MyStr2 & ""
'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη
'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table"
'και "Connect"
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection with table failed!Table '" &
CommonDialog1.FileTitle & "' doesn't exist,or is not supported.Operation now will
stop!Start again from the beginning!"
    Exit Function
End If

ElseIf Option3.Value = True Then 'ms_access
'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα
'στοιχεία σύνδεσης
conn.Open "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False;Data
Source=" & CommonDialog1.FileName
'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη
'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table"
'και "Connect"
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection with database failed!Database '" &
CommonDialog1.FileTitle & "' doesn't exist,or is not supported.Operation now will
stop!Start again from the beginning!"
    Exit Function
End If

ElseIf Option4.Value = True Then 'sql_server
'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα
'στοιχεία σύνδεσης
'Ανάλογα με το αν πατήθηκε το "Change Login" CommandButton το
connection_string της σύνδεσης αλλάζει
If chlog = 0 Then 'change_login clicked

```

```

conn.Open "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=" & Text1.Text
ElseIf chlog = 1 Then
    conn.Open "Provider=sqloledb;Data Source=" & Label13.Caption & ";Initial
Catalog=" & Text1.Text & ";User Id=" & Label9.Caption & ";Password=" &
Label11.Caption & ""
End If
'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη
'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table"
'και "Connect"
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection failed because of non valid information!Operation
now will stop!Start again from the beginning!"
    Exit Function
End If

ElseIf Option5.Value = True Then 'mysql
'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα
'στοιχεία σύνδεσης
conn.Open "Provider=MySqlProv;Location=" & Label13.Caption & ";User ID=" &
Label9.Caption & ";Password=" & Label11.Caption & ";Data Source=" & Text1.Text
'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη
'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table"
'και "Connect"
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection failed because of non valid information!Operation
now will stop!Start again from the beginning!"
    Exit Function
End If

ElseIf Option6.Value = True Then 'mimer
'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα
'στοιχεία σύνδεσης
conn.Open "Provider=MSDASQL.1;Password=" & Label11.Caption & " ;Persist
Security Info=True;User ID=" & Label9.Caption & ";Data Source=" & Text1.Text &
""
'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη
'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table"
'και "Connect"
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection failed because of non valid information!Operation
now will stop!Start again from the beginning!"
    Exit Function
End If
ElseIf Option7.Value = True Then 'sqlite

```

'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα 'στοιχεία σύνδεσης  
conn.Open "DSN=SQLite Datasource;Database=" & CommonDialog1.FileName & ";StepAPI=0;Timeout=1000;NoWCHAR=0""

'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη 'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table" και "Connect"

```
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection with table failed!Table "" &
CommonDialog1.FileTitle & "" doesn't exist,or is not supported.Operation now will
stop!Start again from the beginning!"
    Exit Function
End If
```

Elseif Option8.Value = True Then 'oracle

'Άνοιγμα σύνδεσης χρησιμοποιώντας τον κατάλληλο Provider και τα απαραίτητα 'στοιχεία σύνδεσης  
conn.Open "Provider=OraOLEDB.Oracle.1;Persist Security Info=False;User ID=" & Label9.Caption & ";Password=" & Label11.Caption & ";Data Source=" & Text1.Text & ""

'Σε περίπτωση λάθους στη σύνδεση έξοδος από συνάρτηση και αλλαγή δείκτη 'conf από '0' σε '1' για να χρησιμοποιηθεί στα CommandButtons "Select Table" και "Connect"

```
If conn.State <> 1 Then
    conf = 1
    MsgBox "Error:Connection failed because of non valid information!Operation
now will stop!Start again from the beginning!"
    Exit Function
End If
```

End If

End Function

### **'loadList1' function**

Η συνάρτηση loadList1 φορτώνει στο ListBox τα ονόματα των πινάκων της συνδεδεμένης βάσης δεδομένων.Στον κώδικα της συνάρτησης υπάρχουν ρουτίνες για απομάκρυνση των πινάκων του συστήματος από τη λίστα.

Η loadList1 καλείται στους κώδικες των CommandButtons "Select Table" και "Connect" (με εξαίρεση για Option1 και Option2 που δε χρησιμοποιείται) αμέσως μετά την κλήση της ConnectDB.

```
Function loadList1()
```

'Το rstSchema που είναι recordset ανοίγει, για να δώσει πληροφορίες για την βάση 'από τον Provider

```

'Αν είναι επιλεγμένο το Option6 (Mimer) ή το Option9 (Oracle) ,το
'TABLE_TYPE="Table" κι έτσι απομονώνονται αυτόματα οι πίνακες του
'συστήματος από το ListBox
If Option6.Value = True Or Option9.Value = True Then 'mimer/oracle
  Set rstSchema = conn.OpenSchema(adSchemaTables, Array(TABLE_CATALOG,
TABLE_SCHEMA, TABLE_NAME, "TABLE"))
  'Αν είναι επιλεγμένο το οποιοδήποτε άλλο από τα Options,το TABLE_TYPE
  'δεν ορίζεται και οι πίνακες του συστήματος απομακρύνονται με ρουτίνα
  'παρακάτω
Else:      Set      rstSchema      =      conn.OpenSchema(adSchemaTables,
Array(TABLE_CATALOG, TABLE_SCHEMA, TABLE_NAME, TABLE_TYPE))
End If

'Μέχρι να φτάσει το rstSchema στο τέλος,φόρτωση στη λίστα των πινάκων της βάσης
Do Until rstSchema.EOF
List1.AddItem rstSchema.Fields(2)
'Απομάκρυνση των πινάκων συστήματος του SQL Server.Οι πίνακες ,που αρχίζουν
'από "sys" ή ο πίνακας "dtproperties", διαγράφονται από τη λίστα
If Option4.Value = True Then
  If Left(List1.List(List1.ListCount - 1), 3) = "sys" Or List1.List(List1.ListCount - 1)
= "dtproperties" Then
    List1.RemoveItem (List1.ListCount - 1)
  End If
'Απομάκρυνση των πινάκων συστήματος της MS Acces.Οι πίνακες που αρχίζουν από
"Msys" διαγράφονται από τη λίστα
Elseif Option3.Value = True Then 'ms_access
  If Left(List1.List(List1.ListCount - 1), 4) = "MSys" Then
    List1.RemoveItem (List1.ListCount - 1)
  End If
End If

rstSchema.MoveNext
Loop
rstSchema.Close

End Function

```

### **'Attribute Type' CommandButton**

Το CommandButton "Attribute Type" εξάγει τον επιλεγμένο πίνακα σε αρχείο xml.Το αρχείο είναι τύπου Attribute.Περιέχει, δηλαδή, τη δομή του πίνακα(το σχήμα) και τις εγγραφές.

```

Private Sub CreateAttrXML_Click()

'Σε περίπτωση λάθους μεταπήδηση στο διαχειριστή λάθους
On Error GoTo ErrorHandler

'To recordset 'rstSchema' ανοίγει και διαλέγει τα πεδία του πίνακα και τις 'εγγραφές
τους
If Option1.Value = True Then 'paradox (5.x)

```



```
Set rstSchema = New ADODB.Recordset
rstSchema.Open "SELECT * from " & CommonDialog1.FileName & " ", conn, ,
adCmdText
```

'To recordset 'rstSchema' ανοίγει και θέτεται ίσο με όλα τα πεδία του πίνακα και τις 'εγγραφές τους

```
ElseIf Option2.Value = True Then 'dbase (5)
```

```
Set rstSchema = New ADODB.Recordset
```

```
rstSchema.Open "SELECT * from " & CommonDialog1.FileName & " ", conn, ,
adCmdText
```

'To recordset 'rstSchema' ανοίγει και θέτεται ίσο με όλα τα πεδία του πίνακα και τις 'εγγραφές τους

```
ElseIf Option3.Value = True Or Option4.Value = True Then 'ms_access / sqlserver
```

```
rstSchema.Open "SELECT * from [" & List1.Text & "]" , conn, adOpenStatic,
adLockReadOnly
```

'To recordset 'rstSchema' ανοίγει και θέτεται ίσο με όλα τα πεδία του πίνακα και τις 'εγγραφές τους

```
ElseIf Option5.Value = True Then 'mysql
```

```
rstSchema.Open "SELECT * from `" & List1.Text & "` " , conn, adOpenStatic,
adLockReadOnly
```

'To recordset 'rstSchema' ανοίγει και θέτεται ίσο με όλα τα πεδία του πίνακα και τις 'εγγραφές τους

```
ElseIf Option6.Value = True Then 'mimer
```

```
rstSchema.Open "SELECT * from " & List1.Text & " ", conn, adOpenStatic,
adLockReadOnly
```

'To recordset 'rstSchema' ανοίγει και θέτεται ίσο με όλα τα πεδία του πίνακα και τις 'εγγραφές τους

```
ElseIf Option7.Value = True Then 'sqlite
```

```
rstSchema.Open "SELECT * from " & List1.Text & " ", conn, adOpenStatic,
adLockReadOnly
```

'To recordset 'rstSchema' ανοίγει και θέτεται ίσο με όλα τα πεδία του πίνακα και τις 'εγγραφές τους

```
ElseIf Option8.Value = True Then 'oracle
```

```
rstSchema.Open "SELECT * from " & List1.Text & " ", conn, adOpenStatic,
adLockReadOnly
```

```
End If
```

'Ανοιγμα του CommonDialogControl και καθορισμός θέσης αποθήκευσης του XML 'αρχείου στο σκληρό δίσκο.

```
CommonDialog1.CancelError = True
```

```
CommonDialog1.FileName = List1.Text
```

```
CommonDialog1.InitDir = App.Path
```

```
CommonDialog1.Filter = "XML Files (*.xml)|*.xml"
```

```
CommonDialog1.ShowSave
```

```
'Το xmlfinal είναι ίσο με το CommonDialog1.FileTitle χωρίς το “.xml”
```

```
xmlfinlen = Len(CommonDialog1.FileTitle)
```

```
xmlfinal = Left(CommonDialog1.FileTitle, xmlfinlen - 4)
```

```
'Αποθήκευση του σχήματος σαν xml αρχείο μέσω του ADO Persist
```

```
rstSchema.Save xmlfinal & ".xml", adPersistXML
```

```
'Κλείσιμο του σχήματος και της σύνδεσης
```

```
rstSchema.Close
```

```
conn.Close
```

```
Set conn = Nothing
```

```
MsgBox "XML file exported to selected destination!"
```

```
Option1.Value = True
```

```
Text1.Text = ""
```

```
Label1.Caption = ""
```

```
List1.Clear
```

```
'Επιστροφή στη φόρμα2
```

```
Unload Form3
```

```
Form3.Hide
```

```
Form2.Show
```

```
'Διαχειριστής λάθους
```

```
ErrorHandler:
```

```
'Αν πατηθεί το “Cancel” του CommonDialogControl ,εμφάνιση μηνύματος για επα-  
'νάκληση του “Attribute Type” CommandButton
```

```
    If Err.Number = 32755 Then
```

```
        MsgBox "Operation canceled!Table "" & List1.Text & "" has not been  
exported.Click 'Export XML' button again! "
```

```
        rstSchema.Close
```

```
        Exit Sub
```

```
'Αν το αρχείο xml υπάρχει ήδη στο σκληρό, εμφάνιση μηνύματος αναφοράς και  
'επιστροφή στη φόρμα2
```

```
    ElseIf Err.Number = 58 Then
```

```
        MsgBox "" & CommonDialog1.FileTitle & "" allready exists!No operation  
done.Delete or rename XML_file and run again!"
```

```
        Unload Form3
```

```
        Form3.Hide
```

```
        Form2.Show
```

```
        Exit Sub
```

```
    End If
```

```
End Sub
```

### **‘Element Type’ CommandButton**

Το CommandButton “Element Type” εξάγει τον επιλεγμένο πίνακα σε αρχείο xml.Το αρχείο είναι τύπου Element.Περιέχει ,δηλαδή, μόνο τις εγγραφές του πίνακα.

Η εξαγωγή γίνεται σε τρία στάδια. Στο πρώτο στάδιο, έχουμε την εξαγωγή και προσωρινή αποθήκευση του πίνακα σαν Attribute-centric XML file. Στο δεύτερο στάδιο, ακολουθεί ο μετασχηματισμός του Attribute-centric XML file σε Element-centric XML file μέσω αρχείου αρχείου XSL. Τέλος, ακολουθεί η διαγραφή του Attribute-centric XML file και του αρχείου XSL.

```
Private Sub CreateElemXML_Click()
```

```
'Σε περίπτωση λάθους, μεταπήδηση στο διαχειριστή λάθους
```

```
On Error GoTo ErrorHandler
```

```
'To recordset 'rstSchema' ανοίγει και διαλέγει τα πεδία του πίνακα και τις
```

```
'εγγραφές τους
```

```
If Option1.Value = True Then 'paradox (5.x)
```

```
Set rstSchema = New ADODB.Recordset
```

```
rstSchema.Open "SELECT * from " & CommonDialog1.FileName & " ", conn, ,  
adCmdText
```

```
Elseif Option2.Value = True Then 'dbase (5)
```

```
Set rstSchema = New ADODB.Recordset
```

```
rstSchema.Open "SELECT * from " & CommonDialog1.FileName & " ", conn, ,  
adCmdText
```

```
Elseif Option3.Value = True Or Option4.Value = True Then 'ms_access / sqlserver
```

```
rstSchema.Open "SELECT * from [" & List1.Text & "]", conn, adOpenStatic,  
adLockReadOnly
```

```
Elseif Option5.Value = True Then 'mysql
```

```
rstSchema.Open "SELECT * from `" & List1.Text & "`", conn, adOpenStatic,  
adLockReadOnly
```

```
Elseif Option6.Value = True Then 'mimer
```

```
rstSchema.Open "SELECT * from " & List1.Text & " ", conn, adOpenStatic,  
adLockReadOnly
```

```
Elseif Option7.Value = True Then 'sqlite
```

```
rstSchema.Open "SELECT * from " & List1.Text & " ", conn, adOpenStatic,  
adLockReadOnly
```

```
Elseif Option8.Value = True Then 'oracle
```

```
rstSchema.Open "SELECT * from " & List1.Text & " ", conn, adOpenStatic,  
adLockReadOnly
```

```
End If
```

```
'Προσωρινή αποθήκευση του αρχείου 'xmldoc.xml', το οποίο είναι Attribute Type
```

```
rstSchema.Save App.Path & "xmldoc.xml", adPersistXML
```

```
Dim MyString
```

'Το RichTextBox περιέχει το “.xml” αρχείο, για το μετασχηματισμό του ‘xmldoc.xml’ σε 'Element Type xml αρχείο.Αλλαγή του “TableName” μέσα στο κείμενο του RichTextBox με το ‘όνομα\_πίνακα’.

```
MyString = Replace(RichTextBox1.Text, "<TableName>", "<" & List1.Text & ">")
```

```
RichTextBox1.Text = MyString
```

```
MyString = Replace(RichTextBox1.Text, "</TableName>", "</" & List1.Text & ">")
```

```
RichTextBox1.Text = MyString
```

'Προσωρινή αποθήκευση του αρχείου ‘xsl doc.xml’ για χρησιμοποίηση του στο 'μετασχηματισμό

```
RichTextBox1.SaveFile App.Path & "xsl doc.xml", 1
```

```
rstSchema.Close
```

```
conn.Close
```

```
Set conn = Nothing
```

'Χρησιμοποίηση του XML DOM για τον μετασχηματισμό σε Element-centric XML

```
Dim domIn As DOMDocument26
```

```
Dim domOut As DOMDocument26
```

```
Dim domStylesheet As DOMDocument26
```

```
Set domIn = New DOMDocument26
```

```
domIn.async = False
```

'Φόρτωση του ‘xmldoc.xml’

```
domIn.Load (App.Path & "xmldoc.xml")
```

'Φόρτωση του αρχείου μετασχηματισμού

```
Set domStylesheet = New DOMDocument26
```

```
domStylesheet.Load App.Path & "xsl doc.xml"
```

```
If Not domStylesheet Is Nothing Then
```

```
Set domOut = New DOMDocument26
```

'Μετασχηματισμός του domIn

```
domIn.transformNodeToObject domStylesheet, domOut
```

'Άνοιγμα του CommonDialogControl και καθορισμός θέσης αποθήκευσης του 'domOut σαν αρχείο xml στο σκληρό δίσκο.

```
CommonDialog1.CancelError = True
```

```
CommonDialog1.FileName = List1.Text
```

```
CommonDialog1.InitDir = App.Path
```

```
CommonDialog1.Filter = "XML Files (*.xml)|*.xml"
```

```
CommonDialog1.ShowSave
```

```
xmlfinlen = Len(CommonDialog1.FileTitle)
```

```
xmlfinal = Left(CommonDialog1.FileTitle, xmlfinlen - 4)
```

```
domOut.Save xmlfinal & ".xml"
```

```
End If
```

'Διαγραφή των προσωρινών αρχείων ‘xmldoc.xml’ και ‘xsl doc.xml’

```

Dim fso As New Scripting.FileSystemObject
fso.DeleteFile App.Path & "xmldoc.xml"
fso.DeleteFile App.Path & "xsl doc.xsl"

MsgBox "XML file exported to selected destination!"

Option1.Value = True
Text1.Text = ""
Label1.Caption = ""
List1.Clear

'Επιστροφή στη φόρμα2
Unload Form3
Form3.Hide
Form2.Show

'Διαχειριστής λάθους
ErrorHandler:
'Αν πατηθεί το "Cancel" του CommonDialogControl ,εμφάνιση μηνύματος για επα-
'νάκληση του "Attribute Type" CommandButton
    If Err.Number = 32755 Then 'if cancel error occur
        MsgBox "Operation canceled!Table '" & List1.Text & "' has not been
exported.Click 'Export XML' button again! "
        fso.DeleteFile App.Path & "xmldoc.xml"
        fso.DeleteFile App.Path & "xsl doc.xsl"
        Exit Sub
'Αν το αρχείο xml υπάρχει ήδη στο σκληρό, εμφάνιση μηνύματος αναφοράς και
'επιστροφή στη φόρμα2
    ElseIf Err.Number = 58 Then 'if xml file allready exists
        MsgBox "" & CommonDialog1.FileTitle & "' allready exists!No operation
done.Delete or rename XML_file and run again!"
        fso.DeleteFile App.Path & "xmldoc.xml"
        fso.DeleteFile App.Path & "xsl doc.xsl"
        Unload Form3
        Form3.Hide
        Form2.Show
        Exit Sub
    End If

End Sub

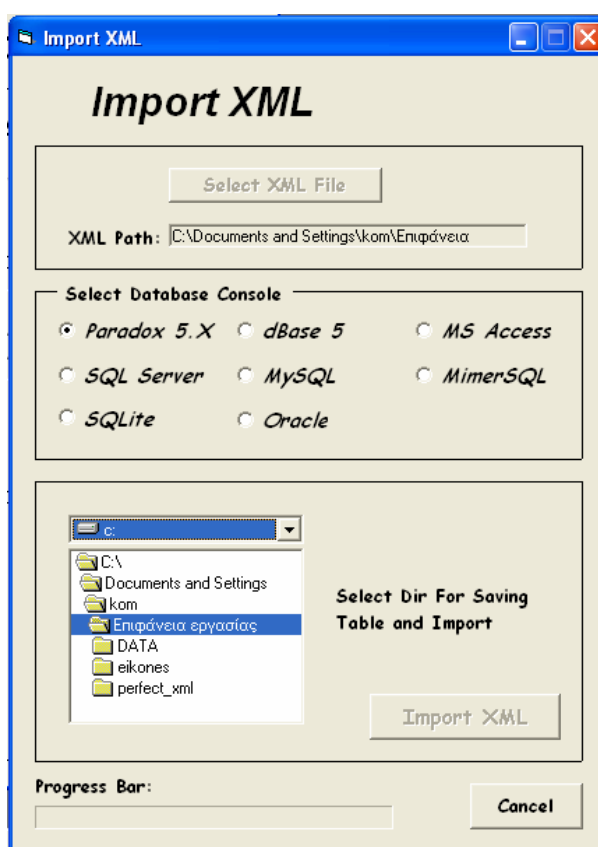
```

## 7.4 Η ΦΟΡΜΑ “Import XML”

Στη φόρμα4(“**Import XML**”),ο χρήστης κάνει εισαγωγή ενός αρχείου XML σε κάποια από τις βάσεις δεδομένων, που υποστηρίζει το πρόγραμμα.Για τις περιπτώσεις Paradox 5.X και dBASE 5, ειδικά,το XML αρχείο εξάγεται κατευθείαν σε πίνακα “.dbf” ή “.db” αντίστοιχα.Η εισαγωγή του xml στη βάση γίνεται σε δύο στάδια:

1. Δημιουργία του πίνακα της βάσης.
2. Εισαγωγή των δεδομένων στον πίνακα.

Η φόρμα4 παρουσιάζεται στην εικόνα Form4a και μπορεί να εμφανιστεί με άλλες δύο παραλλαγές,ανάλογα με τον τύπο της βάσης δεδομένων,που χρησιμοποιείται κατά την εκτέλεση του προγράμματος(εικόνες Form4b και Form4c).



(εικόνα **Form4a**)

Αρχικά, στη φόρμα4 ενεργό είναι μόνο το CommandButton “**Select XML File**”. Πατώντας το, ο χρήστης επιλέγει το αρχείο XML που θα εισάγει στη βάση. Το CommonDialogControl εμφανίζεται και ο χρήστης επιλέγει μέσω αυτού.Ο κώδικας του CommandButton περιέχει και ρουτίνα για την περίπτωση που ο χρήστης δεν επιλέξει XML αρχείο.Παρακάτω φαίνεται και αναλύεται ο αντίστοιχος κώδικας

### “Select XML File” CommandButton

```
Private Sub SelectXML_Click()
```

'Σε περίπτωση λάθους μεταπήδηση στον διαχειριστή λάθους

```
On Error GoTo ErrorHandler
```

```
'Ορισμός φίλτρου (*.xml) για το CommonDialogControl
```

```
CommonDialog1.Filter = "XML Files|*.xml"
```

```
CommonDialog1.DialogTitle = "Select XML File"
```

```
'Άνοιγμα του CommonDialogControl
```

```
CommonDialog1.ShowOpen
```

```
'Αν επιλεγεί XML αρχείο στο CommonDialogControl, το Label του CommandButton
```

```
'εμφανίζει το Path του αρχείου
```

```
If CommonDialog1.FileName <> "" Then
```

```
Label2.Caption = CommonDialog1.FileName
```

```
End If
```

```
'MyLen = πλήθος σε χαρακτήρες του ονόματος του αρχείου
```

```
MyLen = Len(CommonDialog1.FileTitle)
```

```
'Συναρτήσεις για αντικατάσταση του κενού χαρακτήρα (" ") στο όνομα του αρχείου  
από το underscore (" _")
```

```
Dim xmlfle, MyPos
```

```
MyPos = InStr(CommonDialog1.FileTitle, " ")
```

```
If MyPos > 0 Then
```

```
xmlfle = Replace(CommonDialog1.FileTitle, " ", "_")
```

```
MyStr1 = xmlfle
```

```
Else
```

```
MyStr1 = CommonDialog1.FileTitle
```

```
End If
```

```
'Το όνομα του αρχείου χωρίς ".xml" θα χρησιμοποιηθεί στη δημιουργία του πίνακα
```

```
Label5.Caption = Left(MyStr1, MyLen - 4)
```

```
'Διαχειριστής Λάθους
```

```
'Εμφάνιση μηνύματος για επανάκληση στην περίπτωση που δεν επιλεγεί XML file
```

```
'στο CommonDialogControl
```

```
ErrorHandler:
```

```
If Err.Number = 5 Then
```

```
MsgBox "Please select xml file!"
```

```
Exit Sub
```

```
End If
```

```
'Το CommandButton "Select XML File" και τα Drive/Dir ListBoxes γίνονται  
ανενεργά για αποφυγή λαθών. Αντίθετα τα OptionButtons γίνονται ενεργά
```

```
Dir1.Enabled = True
```

```
Drive1.Enabled = True
```

```
Option1.Enabled = True
```

```
Option2.Enabled = True
```

```
Option3.Enabled = True
```

```
Option4.Enabled = True
```

```
Option5.Enabled = True
```

```
Option6.Enabled = True
```

```
Option7.Enabled = True
```

```
Option8.Enabled = True
SelectXML.Enabled = False

End Sub
```

Στη συνέχεια όπως είδαμε και παραπάνω, τα OptionButtons γίνονται ενεργά, για να επιλέξει ο χρήστης σε ποια βάση θα κάνει εισαγωγή το xml αρχείο. Η αντιστοιχία με τους οχτώ τύπους βάσεων δεδομένων είναι ίδια με αυτήν της φόρμας3.

```
Option1 → Paradox 5.X
Option2 → Dbase 5
Option3 → MS Access
Option4 → MS SQL Server
Option5 → MySQL
Option6 → MimerSQL
Option7 → SQLite
Option8 → Oracle 9i
```

Για Paradox 5.X, Dbase 5, η φόρμα4 έχει τη μορφή της εικόνας Form4a. Ο χρήστης, μέσω των Drive και Dir ListBoxes, τα οποία είναι κατάλληλα συνδεδεμένα με κώδικα μεταξύ τους, επιλέγει τον κατάλογο, στον οποίο θα αποθηκευτεί ο “.dbf” ή “.db” πίνακας, που θα έχει τη δομή και τα δεδομένα που περιέχει το xml αρχείο. Αμέσως μόλις, επιλεγεί κατάλογος, το CommandButton “**Import XML**” γίνεται ενεργό.

### DriveListBox Change

```
Private Sub Drive1_Change()
'Συγχρονισμός των Drive και Dir ListBoxes
Dir1.Path = Drive1.Drive
End Sub
```

### DirListBox Click

```
Private Sub Dir1_Click()
'To CommandButton “Import XML” γίνεται ενεργό
ImportXML.Enabled = True
End Sub
```

Για MS Access και SQLite, η φόρμα4 έχει τη μορφή της εικόνας Form4b. Τα Drive και Dir ListBoxes εξαφανίζονται και αντικαθίστανται από το CommandButton “**Select Database**”, καθώς και το αντίστοιχο Label. Ο χρήστης επιλέγει τη βάση “.mdb” ή “.db” αντίστοιχα, στην οποία θα εισαχθεί το xml αρχείο, και κατόπιν το CommandButton “**Import XML**” γίνεται ενεργό.

### “Select Database” CommandButton

```
Private Sub SelectDatabase_Click()
```



```

'Σε περίπτωση λάθους μεταπήδηση στον διαχειριστή λάθους
On Error GoTo ErrorHandler
'Ενεργοποίηση εμφάνισης μηνύματος λάθους, σε περίπτωση που πατηθεί το Cancel
'του CommonDialogControl
CommonDialog1.CancelError = True
CommonDialog1.FileName = " "
ElseIf Option3.Value = True Then
    CommonDialog1.Filter = "Databases (*.mdb)|*.mdb"
ElseIf Option7.Value = True Then
    CommonDialog1.Filter = "Databases (*.db)|*.db"
EndIf
'Άνοιγμα και εμφάνιση στη φόρμα4 του CommonDialogControl
CommonDialog1.ShowOpen

'Αν επιλεγεί βάση
If CommonDialog1.FileName <> "" Then
    'Εμφάνιση του Path της βάσεως στο αντίστοιχο Label
    Label8.Caption = CommonDialog1.FileName
End If

'To CommandButton “Select XML File” και τα OptionButtons γίνονται ανενεργά για
αποφυγή λαθών.Αντίθετα τ ο CommandButton “Import XML” γίνεται ενεργό
ImportXML.Enabled = True
Option1.Enabled = False
Option2.Enabled = False
Option3.Enabled = False
Option4.Enabled = False
Option5.Enabled = False
Option6.Enabled = False
Option7.Enabled = False
Option8.Enabled = False
Option9.Enabled = False
SelectDatabase.Enabled = False
SelectXML.Enabled = False

'Διαχειριστής Λάθους
'Εμφάνιση μηνύματος για επανάκληση του “Select Database” CommandButton στην
'περίπτωση που δεν επιλεγεί βάση στο CommonDialogControl
ErrorHandler:
    If Err.Number = 32755 Then 'if cancel error occur
        If Option3.Value = True Then
            MsgBox "Please select (*.mdb) database!"
        ElseIf Option7.Value = True Then
            MsgBox "Please select (*.db) database!"
        EndIf
        Exit Sub
    End If
End Sub

```

**Import XML**

Select XML File

XML Path: C:\Documents and Settings\kom\Επιφάνεια

Select Database Console

Paradox 5.X   
  dBase 5   
  MS Access  
 SQL Server   
  MySQL   
  MimerSQL  
 SQLite   
  Oracle

Select Database

Database Path:

Import XML

Progress Bar: \_\_\_\_\_

Cancel

(εικόνα Form4b)

**Import XML**

Select XML File

XML Path: C:\Documents and Settings\kom\Επιφάνεια

Select Database Console

Paradox 5.X   
  dBase 5   
  MS Access  
 SQL Server   
  MySQL   
  MimerSQL  
 SQLite   
  Oracle

Change Login

Enter Database Name and Import

Database Name: \_\_\_\_\_

Use ID: root \_\_\_\_\_

Password: \_\_\_\_\_

Locati: localhost \_\_\_\_\_

Import XML

Progress Bar: \_\_\_\_\_

Cancel

(εικόνα Form4c)

Για οποιονδήποτε άλλο διακομιστή, τέλος, η φόρμα4 έχει τη μορφή της εικόνας Form4c. Το CommandButton “**Select Database**” και το αντίστοιχο Label αντικαθίστανται από ένα TextBox για την εισαγωγή του ονόματος της βάσης, τρία Labels για τα υπόλοιπα στοιχεία της σύνδεσης (**UserID, Password, ServerName**) και ένα CommandButton, το “Change Login”, για αλλαγή των στοιχείων, αν ο χρήστης θελήσει. Με το που εισαχθεί όνομα για τη βάση δεδομένων, το CommandButton “**Import XML**” γίνεται ενεργό.

Στη συνέχεια, αφού επιλεγθεί είτε η βάση, στην οποία θα εισαχθεί το xml αρχείο, είτε οριστούν τα στοιχεία για την σύνδεση μ’ αυτήν, η διαδικασία συνεχίζει με το CommandButton “**ImportXML**”. Με το CommandButton αυτό, το πρόγραμμα δημιουργεί μία σύνδεση με το διακομιστή της βάσης, δημιουργεί το νέο πίνακα με πληροφορίες που συλλέγει από το xml αρχείο και στη συνέχεια εισάγει τα δεδομένα (εγγραφές). Τέλος, ακολουθεί επιστροφή στη φόρμα2.

Πριν αναλυθεί όμως ο τρόπος με τον οποίο γίνεται η εισαγωγή, πρέπει να γίνει μία αναφορά στον όρο «μετασχηματισμός δεδομένων», ο οποίος είναι και το κλειδί στην όλη διαδικασία εισαγωγής του αρχείου xml στην βάση δεδομένων.

Ο σκοπός του μηχανισμού «μετασχηματισμού δεδομένων», είναι να αντιγράψει σχήματα βάσεων δεδομένων ανάμεσα σε ομοιογενή και ετερογενή DBMS και να παρέχει υπηρεσίες εισαγωγής, εξαγωγής και μετασχηματισμού δεδομένων ανάμεσα σε πολλαπλές πηγές ομοιογενών και ετερογενών βάσεων δεδομένων. Οι ομοιογενείς βάσεις δεδομένων είναι εκείνες στις οποίες τόσο η πηγή προέλευσης, όσο και ο στόχος είναι ίδιου τύπου συστήματα (π.χ. SQL Server). Οι ετερογενείς βάσεις δεδομένων είναι όμως διαφορετικού τύπου DBMS όπου για παράδειγμα τα δεδομένα μεταφέρονται ανάμεσα στην DBase και στην MySQL. Σε αυτή την περίπτωση, είναι δυνατό να χρησιμοποιηθεί η αρχιτεκτονική που παρέχει το OLE DB ανάμεσα σε διαφορετικές προελεύσεις δεδομένων.

Η εξαγωγή δεδομένων και ο μετασχηματισμός δεδομένων μπορούν να χρησιμοποιηθούν και στην διαδικασία δημιουργίας αποθηκών δεδομένων και μικρών αποθηκών δεδομένων και ανήκουν στην φάση ενοποίησης συνόλων εγγραφών από πολλές διαφορετικές πηγές (π.χ. Oracle και MS Access). Πριν να γίνει ο μετασχηματισμός, τα δεδομένα πρέπει να εξαχθούν με ένα κατάλληλο τρόπο, από ετερογενείς πλατφόρμες, και μετά την διαδικασία μετασχηματισμού, τα προκύπτοντα δεδομένα πρέπει να φορτωθούν. Επειδή αυτή η διαδικασία θα γίνεται ξανά και ξανά (για να κρατείται η αποθήκη δεδομένων πάντα ενημερωμένη), είναι καλό ο μηχανισμός μετασχηματισμού δεδομένων να κάνει αυτό το βήμα αυτόματα.

Στη συγκεκριμένη εφαρμογή το ρόλο του πρώτου συστήματος παίζει το αρχείο XML το οποίο στην ουσία περιέχει το σχήμα του πίνακα της πρώτης βάσης, το οποίο θέλουμε να εξάγουμε και να μετασχηματίσουμε στο σχήμα του πίνακα της δεύτερης βάσης.

Ο μηχανισμός μετασχηματισμού δεδομένων είναι γενικά ένα εργαλείο για μετασχηματισμό δεδομένων από μια ή περισσότερες βάσεις δεδομένων προέλευσης σε μια ή περισσότερες βάσεις δεδομένων στόχου. Σε αυτή την "φάση ενοποίησης δεδομένων", τα δεδομένα εξάγονται από διαφορετικές προελεύσεις και μετατρέπονται σε ένα ενδιάμεσο σχήμα (σχήμα ADO). Μετά από αυτό, τα δεδομένα

καθορίζονται - δηλαδή, όλα τα εξαγόμενα δεδομένα ελέγχονται, και ενοποιούνται διαφορετικές μορφές και τύποι των ίδιων δεδομένων.

Μηχανισμοί μετασχηματισμού δεδομένων μπορούν να δημιουργηθούν είτε χειρωνακτικά είτε χρησιμοποιώντας μια γλώσσα προγραμματισμού που υποστηρίζει αυτοματισμό OLE, σαν την Visual Basic.

Έτσι, η εισαγωγή τώρα ενός αρχείου XML σε μία βάση γίνεται ως εξής:

1. Ανοίγουμε το xml αρχείο και μαζί μ'αυτό ένα source\_recordset ,το οποίο αναπαριστά όλα τα δεδομένα του xml αρχείου.Χρησιμοποιώντας το recordset αυτό, θα δημιουργήσουμε τη δομή του νέου πίνακα και θα εισάγουμε τα δεδομένα σ'αυτόν.
2. Ανατρέχουμε σε όλα τα πεδία του source\_recordset,εντοπίζουμε τα ονόματα αυτών και ελέγχουμε αν υπάρχει ο κενός χαρακτήρας (' ') ανάμεσά τους.Αν υπάρχει, τον αντικαθιστούμε με το underscore ('\_') κι έτσι αποφεύγουμε το ενδεχόμενο παρουσίας λάθους κατά τη δημιουργία των πεδίων του νέου πίνακα.Επίσης, ελέγχουμε τον τύπο δεδομένων του κάθε πεδίου του source\_recordset και τον αντιστοιχίζουμε με τον ανάλογο που υποστηρίζει το ADO.Αυτός ,με τη συνέχεια, αντιστοιχίζεται και πάλι στον τύπο δεδομένων της βάσης στην οποία θα δημιουργηθεί ο πίνακας.
3. Φορτώνουμε σε μία λίστα (**ListBox2**), που δεν είναι φανερή στο χρήστη τον αύξοντα αριθμό του πεδίου.Ακόμη, σε μία άλλη λίστα (**ListBox1**),φορτώνουμε το όνομα του πεδίου και τον τύπο δεδομένων της βάσης.Σε όλες τις παραπάνω διαδικασίες ,εξαιρούνται τα πεδία με τύπους που αντιπροσωπεύουν εικόνες ή OLE αντικείμενα.Αυτό γίνεται ,γιατί τους συγκεκριμένους τύπους δεδομένων δεν τους υποστηρίζουν όλες οι βάσεις κι έτσι, θα υπάρξουν αστοχίες κατά την εκτέλεση του προγράμματος.
4. Ενωποιούμε όλες τις γραμμές της λίστας1 σε μία και χρησιμοποιούμε τη γραμμή αυτή με εντολές SQL για να δημιουργήσουμε το νέο πίνακα.
5. Δημιουργούμε ένα destination\_recordset ,το οποίο έχει τη δομή του νέου πίνακα.Αποτελείται ,δηλαδή, μόνο από τα πεδία που πληρούν τις παραπάνω προδιαγραφές γι'αυτά.
6. Τέλος,θέτουμε τις εγγραφές των πεδίων του destination\_recordset ίσες με τις εγγραφές των πεδίων του source\_recordset που αναγράφονται στη λίστα2.

Για Paradox 5.X ,ο νέος πίνακας επιβαρύνεται με ένα ακόμη πεδίο τύπου 'number' και ονόματος 'record'. Οι τιμές αυτού του πεδίου, είναι ίσες με τον αύξοντα αριθμό της αντίστοιχης εγγραφής.Το πεδίο αυτό προστίθεται επειδή οποιοσδήποτε πίνακας Paradox απαιτεί ένα πρωτεύον κλειδί για να δημιουργηθεί.Θέτοντας σαν πρωτεύον ,οποιοδήποτε από τα πεδία του destination\_recordset , περιορίζουμε τις τιμές αυτού οι οποίες θα πρέπει να είναι διάφορες του NULL.Επιπλέον ,οι διπλοτιμίες θα απαγορεύονται.Δεν θα μπορεί ,δηλαδή ,να επαναληφθεί κάποια τιμή σε κάποια άλλη εγγραφή του ίδιου πεδίου.Με το πεδίο 'record' ξεπερνιούνται οι δυσκολίες αυτές,καθώς και θα υπάρχει πρωτεύον κλειδί και οι διπλοτιμίες θα είναι ανύπαρκτες.

Παρακάτω παρουσιάζεται και αναλύεται ο κώδικας του CommandButton "Import XML"

## CommandButton “Import XML”

```
Private Sub ImportXML_Click()

'Σε περίπτωση λάθους μεταπήδηση στην επόμενη γραμμή κώδικα
On Error Resume Next

Dim objADODConnDest As New ADODB.Connection
Dim objADORsDest As New ADODB.Recordset

Dim objADORsSrc As ADODB.Recordset
Dim srcFields As ADODB.Fields

'Άνοιγμα του xml αρχείου και του source_recordset
Set objADORsSrc = New ADODB.Recordset
objADORsSrc.Open Label2.Caption, "provider=MSpersist"

'Ορισμός των max και min του ProgressBar, το οποίο παρακολουθεί και ενημερώνει
'για την εισαγωγή των εγγραφών στο νέο πίνακα
Dim bar As Integer
bar = 0
Label10.Enabled = True
ProgressBar1.Max = 100
ProgressBar1.Value = 0

'Δήλωση μεταβλητών προς χρησιμοποίησή τους σε βρόγχους και συναρτήσεις
Dim g As Integer 'used for loop to define the right datatype of each field
Dim dttpe 'the data type of each record
Dim c As Integer 'used for loop to insert data
Dim all As Integer
Dim refi, posrefi
Dim finm As String

'MS Access
If Option3.Value = True Then

Dim xx As String 'used for loop
Dim k As Integer 'used for loop

List1.AddItem "("
For g = 0 To objADORsSrc.Fields.count - 1

'Έλεγχος για κενό χαρακτήρα ( ' ') στο όνομα του κάθε πεδίου και αντικατάσταση
'με το underscore ( '_')
finm = objADORsSrc.Fields(g).Name
posrefi = InStr(finm, " ")
If posrefi <> 0 Then
refi = Replace(finm, " ", "_")
Else: refi = finm
End If
```

'Έλεγχος κάθε πεδίου του πίνακα για τον τύπο δεδομένων ADO και αντιστοίχιση με τους τύπους δεδομένων της MS Access. Πρόσθεση στη λίστα1 του ονόματος και του τύπου δεδομένων. Πρόσθεση στη λίστα2 του αύξοντα αριθμού του πεδίου.

```
If objADORSrc.Fields(g).Type = adArray Then
    dttype = "Text"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adBigInt Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adBoolean Then
    dttype = "Yes/No"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adChar Then
    dttype = "Text"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adCurrency Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adDate Then
    dttype = "Date/Time"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adDBDate Then
    dttype = "Date/Time"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adDBTime Then
    dttype = "Date/Time"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adDBTimeStamp Then
    dttype = "Date/Time"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adDecimal Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adDouble Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORSrc.Fields(g).Type = adInteger Then
    dttype = "Number"
```

```

List1.AddItem (refi & " " & dttype & ", ")
List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adLongVarChar Then
    dttype = "Text"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adLongVarWChar Then
    dttype = "Memo"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adNumeric Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adSingle Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adSmallInt Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adTinyInt Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adUnsignedBigInt Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adUnsignedInt Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adUnsignedSmallInt Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adUnsignedTinyInt Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adVarChar Then
    dttype = "Text"
    List1.AddItem (refi & " " & dttype & "(254)" & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adVarNumeric Then
    dttype = "Number"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)

```

```

ElseIf objADORsSrc.Fields(g).Type = adVarChar Then
    dttype = "Text"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adWChar Then
    dttype = "Text"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
ElseIf objADORsSrc.Fields(g).Type = adGUID Then
    dttype = "AutoNumber"
    List1.AddItem (refi & " " & dttype & ", ")
    List2.AddItem (g)
End If
Next

```

'Το TextBox2 περιέχει τις γραμμές τις λίστας1 ενωποιημένες σε μία  
'Κατόπιν η λίστα1 διαγράφεται

```

Text2.Text = ""
For k = 0 To List1.ListCount
    If k = List1.ListCount - 1 Then
        all = Len(List1.List(k)) - 2
        xx = Left(List1.List(k), all) + ")"
    Else: xx = List1.List(k)
    End If
    Text2.Text = Text2.Text + xx
Next
List1.Clear

```

'Άνοιγμα της σύνδεσης με MS Access

```

objADOCConnDest.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &
Label8.Caption & ";Persist Security Info=False"

```

'Σε περίπτωση λάθους στη σύνδεση ακολουθεί μήνυμα αναφοράς και επιστροφή στη  
'φόρμα2

```

If objADOCConnDest.State <> 1 Then
    MsgBox "Error:Connection failed because of non valid information!Operation
now will stop!Start again from the beginning!"
    Unload Form4
    Form4.Hide
    Form2.Show
    Exit Sub
End If

```

'Δημιουργία του πίνακα με εντολές SQL.H Label5 περιέχει το όνομα του πίνακα ,ενώ  
'το TextBox2 το όνομα και τον τύπο δεδομένων του κάθε πεδίου

```

objADOCConnDest.Execute "Create table " & Label5.Caption & "" & Text2.Text & ""
'Άνοιγμα του destination_recordset
objADORsDest.Open "SELECT * FROM [" & Label5.Caption & "] WHERE 1 = 2",
objADOCConnDest, adOpenKeyset, adLockOptimistic

```



```

'end of MS_Access *****
'Η διαδικασία επαναλαμβάνεται και για τους υπόλοιπους τύπους βάσεων
'δεδομένων με μόνη διαφορά τους τύπους δεδομένων για κάθε διακομιστή.
'Ακολουθεί η εισαγωγή των δεδομένων.

Dim lst As String
Dim lst1, pdox As Integer
lst1 = 0

While Not objADORsSrc.EOF 'while not last record of source recordset
  objADORsDest.AddNew 'add records to destination recordset
  Set srcFields = objADORsSrc.Fields

  'Για Paradox 5.X
  If Option1.Value = True Then 'paradox
    For c = 0 To List2.ListCount - 1
      lst = List2.List(c)
      lst1 = CInt(lst)
      'Η τιμή του πεδίου του νέου πίνακα είναι ίση με την τιμή του αντίστοιχου πεδίου
      'του source_recordset
      objADORsDest.Fields(c + 1).Value = srcFields(lst1).Value
      'Η τιμή του πεδίου "record" του νέου πίνακα είναι ίση με τον αύξοντα αριθμό της
      'εγγραφής
      objADORsDest.Fields("record").Value = pdox
    Next
  'Για όλους τους υπόλοιπους
  Else
    For c = 0 To List2.ListCount - 1
      lst = List2.List(c)
      lst1 = CInt(lst)
      'Η τιμή του πεδίου του νέου πίνακα είναι ίση με την τιμή του αντίστοιχου πεδίου
      'του source_recordset
      objADORsDest.Fields(c).Value = srcFields(lst1).Value
    Next
  End If

  pdox = pdox + 1
  'Ανανέωση του destination_recordset
  objADORsDest.Update
  objADORsSrc.MoveNext
  'Η τιμή του ProgressBar αυξάνεται καθώς προστίθενται εγγραφές
  bar = bar + 1
  ProgressBar1.Value = (bar / objADORsSrc.RecordCount) * 100
Wend

ProgressBar1.Value = 0
bar = 0

```

'Κλείσιμο συνδέσεων

objADORsSrc.Close

Set objADORsSrc = Nothing

objADORsDest.Close

Set objADORsDest = Nothing

objADOCConnDest.Close

Set objADOCConnDest = Nothing

MsgBox "Table Created and XML data imported!"

Label8.Caption = ""

Text1.Text = ""

Text2.Text = ""

Label2.Caption = ""

List2.Clear

'Επιστροφή στη φόρμα2

Unload Form4

Form4.Hide

Form2.Show

ProgressBar1.Enabled = False

Label10.Enabled = False

End Sub

## 7.5 Η ΦΟΡΜΑ “Export XML From Two Tables”

Στη φόρμα5 (“Export XML From Two Tables”), ο χρήστης εξάγει δύο επιθυμητούς πίνακες δεδομένων σε ένα αρχείο τύπου “.xml”. Οι πίνακες αυτοί μπορούν να ανήκουν σε δύο διαφορετικές βάσεις , οι οποίες με τη σειρά τους μπορούν να ανήκουν σε δύο διαφορετικούς διακομιστές. Η φόρμα5, στην ουσία είναι μία διπλή επανάληψη της φόρμας3 , με τη διαφορά ότι, τελικά, αντί να δημιουργηθούν δύο ξεχωριστά xml αρχεία , ένα για κάθε πίνακα, δημιουργείται ένα, το οποίο περιέχει τις εγγραφές και των δύο μαζί.

Από τα παραπάνω συνεπάγεται ως απαραίτητη προϋπόθεση, το να έχουν οι δύο πίνακες ίδιο όνομα, ίδιο πλήθος πεδίων και φυσικά τα ονόματα των πεδίων τους να συμφωνούν και αυτά μεταξύ τους.

Η μορφή της φόρμας5 , φαίνεται παρακάτω στην εικόνα Form5a.

The screenshot shows a Windows-style dialog box titled "Export XML From Two Tables". The dialog has a light beige background and a blue title bar. It is divided into two main sections for configuring two different data sources. The left section, labeled "Select Table1 Source:", includes a dropdown menu currently set to "MS Access", a text box for the "Database Path" containing "D:\Documents and Settings\Kom\Τα", and a "Select Database" button. Below this is a list box titled "Connection1 Tables:" containing the following items: categories, customers, fait, faithths, parex, parole, and products. The right section, labeled "Select Table2 Source:", includes a dropdown menu set to "MySQL", a text box for "Enter Database Name" containing "test", and a "Connect Database" button. Below this are text boxes for "User ID" (containing "root"), "Password", and "Location" (containing "localhost"), along with a "Change Login" button. At the bottom of the dialog, there are two buttons: "Combine Tables and Export XML" and "Cancel".

(εικόνα Form5a)

Όπως είναι εμφανές, η φόρμα3 επαναλαμβάνεται εις διπλούν, με τη μόνη διαφορά ότι τα OptionButtonControls για κάθε πίνακα έχουν αντικατασταθεί από ένα ComboBox, το οποίο περιέχει τους οχτώ τύπους βάσεων δεδομένων, που υποστηρίζει το πρόγραμμα. Ο χρήστης πατώντας επάνω σε κάθε ComboBox , έχει τη δυνατότητα να επιλέξει την πηγή δεδομένων για τον αντίστοιχο πίνακα.

Ανάλογα με το διακομιστή βάσης που επιλέγεται για κάθε πίνακα ,τα CommandButtons “Select Table1”,“Select Table2” και “Connect1”,“Connect2” εμφανίζονται και εξαφανίζονται.Μαζί με αυτά φυσικά και τα ακόλουθα Label τους, τα οποία περιέχουν τα στοιχεία για κάθε σύνδεση. Οι κώδικες των παραπάνω CommandButtons, είναι οι ίδιοι με τους κώδικες των ομώνυμων CommandButton της φόρμας3.

Αφού πραγματοποιηθούν και οι δύο συνδέσεις, τα ονόματα των πινάκων των δύο βάσεων φορτώνονται στις δύο λίστες της φόρμας5 (ListBox1 και ListBox2). Το CommandButton, το οποίο είναι αυτό που δημιουργεί το αρχείο xml, παραμένει ανενεργό έως ότου επιλεγεί πίνακας και από τις δύο λίστες.

Για την συνένωση των δύο πινάκων και την εξαγωγή τους σε ένα αρχείο xml, χρησιμοποιούμε το πολύ σημαντικό εργαλείο της VisualBasic, τον DataEnvironment. Σ’ αυτόν έχουμε την δυνατότητα να δημιουργήσουμε αντικείμενα DEConnections και DECommands που επιτρέπουν την σύνδεση στον επιθυμητό πίνακα και την εξαγωγή πληροφοριών για τη δομή και τα δεδομένα του.Για καθένα από τους δύο πίνακες δημιουργούμε δύο αντικείμενα DECommand, τα “Comm1” και “Comm2” αντίστοιχα. Ορίζοντας τις ιδιότητες CommandType(το οποίο θέτουμε σαν “Table”),το ActiveConnection (ConnectionString με τα στοιχεία της τρέχουσας σύνδεσης) και CommandText (το όνομα του πίνακα) του αντικειμένου Command, δημιουργούμε μία σύνδεση με τον πίνακα και εξάγουμε πληροφορίες για τη δομή και τις εγγραφές μέσω ενός recordset.

Θέτουμε, λοιπόν, τρία recordsets , ένα για κάθε πίνακα και ένα προσωρινό μέσω του οποίου θα εξαχθεί το τελικό αρχείο xml.Ένα εκ των δύο πρώτων αντιγράφεται πλήρως στο προσωρινό και ακολουθεί η αντιγραφή των εγγραφών του δευτέρου και πάλι στο προσωρινό.Εν κατακλείδει, έχουμε σαν αποτέλεσμα ένα τελικό recordset ,το οποίο έχει τη δομή του ενός από τους δύο πίνακες,αλλά τις εγγραφές και τον δύο.Τέλος,το αντικείμενο αυτό αποθηκεύεται σαν αρχείο xml σε κατάλογο του συστήματος που επιλέγει ο χρήστης μέσω ενός CommonDialogControl.

Παρακάτω παρουσιάζεται και αναλύεται ο κώδικας του CommandButton “Combine Tables and Export XML”

#### **“Combine Tables and Export” CommandButton**

```
Private Sub Command6_Click()

Dim fid, j, count, sam, nosames As Integer
Dim rsd As ADODB.Recordset
nosames = 0

'Ορισμός του προσωρινού recordset
Set rsd = New ADODB.Recordset

'Ανάλογα με τον διακομιστή στον οποίο ανήκει ο πίνακας1,ορίζουμε τις ιδιότητες του
'αντικειμένου “Comm1”
DataEnvironment1.Commands("Comm1").CommandType = adCmdTable
```

```

DataEnvironment1.Commands("Comm1").CommandTimeout = 10
If Combo1.Text = "Paradox 5.X" Then 'paradox
    DataEnvironment1.Commands("Comm1").ActiveConnection =
"Driver={Microsoft Paradox Driver (*.db)};" & _
    "DriverID=538;" & _
    "Fil=Paradox 5.X;" & _
    "DefaultDir=" & MyStr2 & ";" & _
    "Dbq=" & MyStr2 & ";" & _
    "CollatingSequence=ASCII"
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
Elseif Combo1.Text = "dBASE 5" Then 'dbase
    DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=MSDASQL.1;Persist Security Info=False;Data Source=dBASE
Files;Initial Catalog=" & MyStr2 & ""
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
Elseif Combo1.Text = "MS Access" Then 'ms_access
    DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False;Data Source= " &
CommonDialog1.FileName
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
Elseif Combo1.Text = "SQL Server" Then 'sql_server
    DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=" & Text1.Text
    If chlog = 0 Then 'Αν δεν αλλάξουν τα στοιχεία της σύνδεσης
        DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;Initial Catalog=" & Text1.Text
    Elseif chlog = 1 Then Αλλιώς
        DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=sqloledb;Data Source=" & Label17.Caption & ";Initial Catalog=" &
Text1.Text & ";User Id=" & Label13.Caption & ";Password=" & Label15.Caption &
""
    End If
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
Elseif Combo1.Text = "MySQL" Then 'mysql
    DataEnvironment1.Commands("Comm1").ActiveConnection =
"DRIVER={MySQL ODBC 3.51 Driver};" & _
    "SERVER=localhost;" & _
    "DATABASE=" & Text1.Text & ";" & _
    "USER=root;" & _
    "PASSWORD=;"
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
Elseif Combo1.Text = "MimerSQL" Then 'mimer
    DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=MSDASQL.1;Password=kom;Persist Security Info=True;User
ID=kom;Data Source=" & Text1.Text & ""
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
Elseif Combo1.Text = "SQLite" Then 'SQLite

```

```

DataEnvironment1.Commands("Comm1").ActiveConnection = "DSN=SQLite
Datasource;Database=      &      CommonDialog1.FileName      &      ";
StepAPI=0;Timeout=1000;NoWCHAR=0"
DataEnvironment1.Commands("Comm1").CommandText = List1.Text

Elseif Combo1.Text = "Oracle" Then 'Oracle
DataEnvironment1.Commands("Comm1").ActiveConnection      =
"Provider=OraOLEDB.Oracle.1;Persist      Security      Info=False;User
ID=scott;Password=tiger;Data Source=" & Text1.Text & ""
DataEnvironment1.Commands("Comm1").CommandText = List1.Text
End If
'Ανοιγμα του recordset1
DataEnvironment1.rsComm1.Open      DataEnvironment1.Commands("Comm1"),      ,
adOpenStatic, adLockReadOnly

'Ανάλογα με τον διακομιστή στον οποίο ανήκει ο πίνακας2,ορίζουμε τις ιδιότητες του
'αντικειμένου "Comm2"
DataEnvironment1.Commands("Comm2").CommandType = adCmdTable
DataEnvironment1.Commands("Comm2").CommandTimeout = 10
If Combo2.Text = "Paradox 5.X" Then 'paradox
DataEnvironment1.Commands("Comm2").ActiveConnection      =
"Driver={Microsoft Paradox Driver (*.db )};" & _
"DriverID=538;" & _
"Fil=Paradox 5.X;" & _
"DefaultDir=" & MyStr4 & ";" & _
"Dbq=" & MyStr4 & ";" & _
"CollatingSequence=ASCII"
DataEnvironment1.Commands("Comm2").CommandText = List2.Text
Elseif Combo2.Text = "dBASE 5" Then 'dbase
DataEnvironment1.Commands("Comm2").ActiveConnection      =
"Provider=MSDASQL.1;Persist      Security      Info=False;Data      Source=dBASE
Files;Initial Catalog=" & MyStr4 & ""
DataEnvironment1.Commands("Comm2").CommandText = List2.Text
Elseif Combo2.Text = "MS Access" Then 'ms_access
DataEnvironment1.Commands("Comm2").ActiveConnection      =
"Provider=Microsoft.Jet.OLEDB.4.0;Persist      Security      Info=False;Data Source=" &
CommonDialog2.FileName
DataEnvironment1.Commands("Comm2").CommandText = List2.Text
Elseif Combo2.Text = "SQL Server" Then 'sql_server
If chlog2 = 0 Then 'Αν δεν αλλάξουν τα στοιχεία της σύνδεσης
DataEnvironment1.Commands("Comm2").ActiveConnection      =
"Provider=SQLOLEDB.1;Integrated      Security=SSPI;Persist      Security
Info=False;Initial Catalog=" & Text1.Text
Elseif chlog2 = 1 Then 'Αλλιώς
DataEnvironment1.Commands("Comm2").ActiveConnection      =
"Provider=sqloledb;Data Source=" & Label23.Caption & ";Initial Catalog=" &
Text1.Text & ";User Id=" & Label19.Caption & ";Password=" & Label21.Caption &
""
End If
DataEnvironment1.Commands("Comm2").CommandText = List2.Text

```

```

Elseif Combo2.Text = "MySQL" Then 'mysql
    DataEnvironment1.Commands("Comm1").ActiveConnection =
"DRIVER={MySQL ODBC 3.51 Driver};" & _
    "SERVER=localhost;" & _
    "DATABASE=" & Text2.Text & ";" & _
    "USER=root;" & _
    "PASSWORD=;"
    DataEnvironment1.Commands("Comm2").CommandText = List2.Text
Elseif Combo2.Text = "MimerSQL" Then 'mimer
    DataEnvironment1.Commands("Comm2").ActiveConnection =
"Provider=MSDASQL.1;Password=kom;Persist Security Info=True;User
ID=kom;Data Source=" & Text1.Text & ""
    DataEnvironment1.Commands("Comm2").CommandText = List2.Text
Elseif Combo2.Text = "SQLite" Then 'SQLite
    DataEnvironment1.Commands("Comm2").ActiveConnection =
"DSN=SQLite
Datasource;Database= " & CommonDialog2.FileName & ";
StepAPI=0;Timeout=1000;NoWCHAR=0"
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text

Elseif Combo2.Text = "Oracle" Then 'Oracle
    DataEnvironment1.Commands("Comm2").ActiveConnection =
"Provider=OraOLEDB.Oracle.1;Persist Security Info=False;User
ID=scott;Password=tiger;Data Source=" & Text2.Text & ""
    DataEnvironment1.Commands("Comm2").CommandText = List2.Text
End If
'Ανοιγμα του recordset2
DataEnvironment1.rsComm2.Open DataEnvironment1.Commands("Comm2"), ,
adOpenStatic, adLockReadOnly

' Έλεγχος για το αν τηρούνται οι απαραίτητες προϋποθέσεις για την συνένωση των
'δύο πινάκων

'Σε περίπτωση ασυμφωνίας των ονομάτων των πινάκων ακολουθεί μήνυμα λάθους,
'έξοδος από τον κώδικα του CommandButton και επιστροφή στη φόρμα2
If List1.Text <> List2.Text Then
    MsgBox "Table names do not match!Operation now will stop!Run again procedure
from the begining!"
    Unload Form5
    Form5.Hide
    Form2.Show
    Exit Sub 'stop procedure
End If
'Σε περίπτωση ασυμφωνίας πλήθους πεδίων των πινάκων ακολουθεί μήνυμα λάθους,
'έξοδος από τον κώδικα του CommandButton και επιστροφή στη φόρμα2
If DataEnvironment1.rsComm1.Fields.count <>
DataEnvironment1.rsComm2.Fields.count Then
    MsgBox "Table_fields counts do not match!Operation now will stop!Run again
procedure from the begining!"
    Form5.Hide

```

```

Unload Form5
Form5.Hide
Form2.Show
Exit Sub 'stop procedure
Set rsd = Nothing
End If
'Σε περίπτωση ασυμφωνίας ονομάτων των πεδίων ακολουθεί μήνυμα λάθους,
'έξοδος από τον κώδικα του CommandButton και επιστροφή στη φόρμα2
For j = 0 To DataEnvironment1.rsComm1.Fields.count - 1
    If DataEnvironment1.rsComm1.Fields(j).Name =
DataEnvironment1.rsComm2.Fields(j).Name Then
        sam = 0
    Else: sam = 1
        nosames = nosames + sam
    End If
Next
If nosames <> 0 Then
    MsgBox "Table_fields names do not match!Operation now will stop!Run again
procedure from the begining!"
    Unload Form5
    Form5.Hide
    Form2.Show
    Exit Sub 'stop procedure
End If

'Αν τηρηθούν όλες οι προϋποθέσεις ακολουθεί δημιουργία πεδίων του προσωρινού
'recordset προς σχηματισμό δομής του νέου πίνακα
For fid = 0 To DataEnvironment1.rsComm1.Fields.count - 1
    rsd.Fields.Append DataEnvironment1.rsComm1.Fields(fid).Name,
DataEnvironment1.rsComm1.Fields(fid).Type,
DataEnvironment1.rsComm1.Fields(fid).DefinedSize, adFldIsNullable
Next
rsd.Open

'Αντιγραφή των εγγραφών του recordset1 στο προσωρινό recordset
While Not DataEnvironment1.rsComm1.EOF
    rsd.AddNew
    For j = 0 To DataEnvironment1.rsComm1.Fields.count - 1
        rsd.Fields(j).Value = DataEnvironment1.rsComm1.Fields(j).Value
    Next
    rsd.Update
    DataEnvironment1.rsComm1.MoveNext
Wend

'Αντιγραφή των εγγραφών του recordset2 στο προσωρινό recordset
While Not DataEnvironment1.rsComm2.EOF
    rsd.AddNew
    For j = 0 To DataEnvironment1.rsComm2.Fields.count - 1
        rsd.Fields(j).Value = DataEnvironment1.rsComm2.Fields(j).Value
    Next

```



```

rsd.Update
DataEnvironment1.rsComm2.MoveNext
Wend

'Άνοιγμα του CommonDialogControl προς επιλογή καταλόγου αποθήκευσης του
'αρχείου xml
xmltable = List1.Text
CommonDialog1.InitDir = App.Path
CommonDialog1.Filter = "XML Files (*.xml)|*.xml"
CommonDialog1.ShowSave

'Αποθήκευση του προσωρινού recordset σαν xml
rsd.Save xmltable & ".xml", adPersistXML

'Κλείσιμο των recordsets
DataEnvironment1.rsComm1.Close
DataEnvironment1.rsComm2.Close
rsd.Close
Set rsd = Nothing

'Έμφάνιση μηνύματος επιτυχούς αποθήκευσης και επιστροφή στη φόρμα2
MsgBox "XML file exported to selected destination!"
Unload Form5
Form5.Hide
Form2.Show

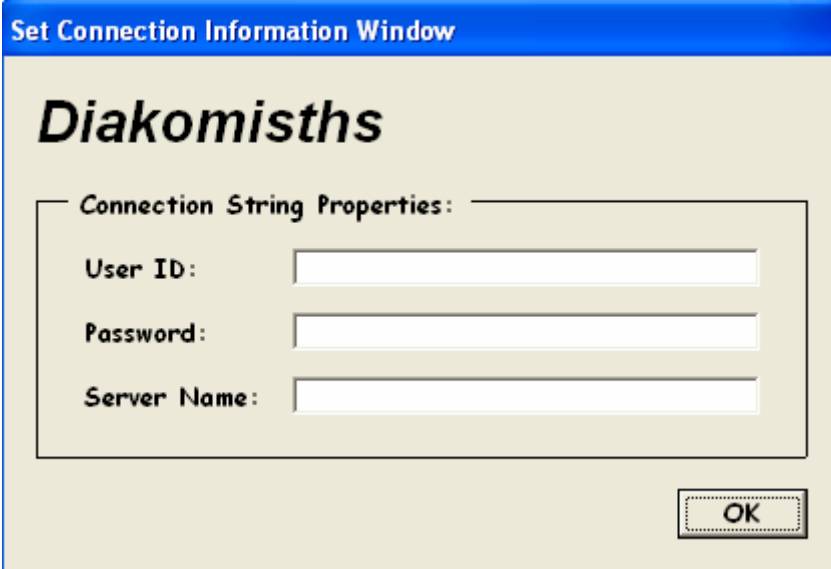
End Sub

```

## 7.6 Η ΦΟΡΜΑ “Set Connection Information Window”

Η φόρμα6(“Set Connection Information Window”) εμφανίζεται με το πάτημα του CommandButton “Change Login” σε κάποια από τις αρχικές φόρμες “Export XML From Table”, “Export XML From Two Tables”, “Import XML” ή “View Data”. Στη φόρμα6 ο χρήστης αλλάζει τα στοιχεία της σύνδεσης της εφαρμογής με τα συστήματα SQL Server , MySQL , MimerSQL ή Oracle.

Η μορφή της φόρμας φαίνεται στην εικόνα form6.



(εικόνα Form6)

Αποτελείται από τρία TextBoxes στα οποία ο χρήστης εισάγει τα στοιχεία (User ID, Password, Server Name) για την σύνδεση στο διακομιστή. Με το πάτημα του CommandButton “OK”, τα δεδομένα που εισήχθησαν στα TextBoxes αντιγράφονται στα ομώνυμα Labels της αρχικής φόρμας και ακολουθεί επιστροφή σ’ αυτή.

Η αρχική φόρμα κατά τη διάρκεια που εμφανίζεται η φόρμα6, παραμένει ορατή αλλά είναι ανενεργή. Να σημειωθεί επίσης ότι η ετικέτα του Label3 (Server Name) ανάλογα με το σύστημα διαχείρισης βάσεων στο οποίο προσπαθούμε να συνδεθούμε αλλάζει π.χ. για τον MySQL είναι “Location:”

Ακολουθεί ο κώδικας του CommandButton “OK”

### “OK” CommandButton

**Για την φόρμα "EXPORT TABLE FROM TABLE"**

If Form3.Visible = True Then

**Τα OptionButtons γίνονται ανενεργά**

Form3.Option1.Enabled = False

Form3.Option2.Enabled = False

Form3.Option3.Enabled = False

Form3.Option4.Enabled = False

Form3.Option5.Enabled = False

Form3.Option6.Enabled = False

```
Form3.Option7.Enabled = False
Form3.Option8.Enabled = False
```

'Ανάλογα με το διακομιστή ορίζονται οι ετικέτες των Labels της φόρμας6 και 'τα νέα δεδομένα της σύνδεσης από τα TextBoxes εισάγονται στα ομώνυμα Labels 'της φόρμας3

```
If Form3.Option4.Value = True Then 'sqlserver
  Form3.Label8.Visible = True
  Form3.Label9.Visible = True
  Form3.Label10.Visible = True
  Form3.Label11.Visible = True
  Form3.Label12.Visible = True
  Form3.Label13.Visible = True
  Form3.Label9.Caption = Text1.Text
  Form3.Label11.Caption = Text2.Text
  Form3.Label13.Caption = Text3.Text
ElseIf Form3.Option5.Value = True Then 'mysql
  Label5.Caption = "Location:"
  Form3.Label9.Caption = Text1.Text
  Form3.Label11.Caption = Text2.Text
  Form3.Label13.Caption = Text3.Text
ElseIf Form3.Option6.Value = True Then 'mimer
  Label5.Visible = False
  Text3.Visible = False
  Form3.Label9.Caption = Text1.Text
  Form3.Label11.Caption = Text2.Text
ElseIf Form3.Option8.Value = True Then 'oracle
  Label5.Visible = False
  Text3.Visible = False
  Form3.Label9.Caption = Text1.Text
  Form3.Label11.Caption = Text2.Text
End If
Form3.Enabled = True
```

'Για την φόρμα "IMPORT XML"

```
ElseIf Form4.Visible = True Then
  Form4.Option1.Enabled = False
  Form4.Option2.Enabled = False
  Form4.Option3.Enabled = False
  Form4.Option4.Enabled = False
  Form4.Option5.Enabled = False
  Form4.Option6.Enabled = False
  Form4.Option7.Enabled = False
  Form4.Option8.Enabled = False
```

'Ανάλογα με το διακομιστή ορίζονται οι ετικέτες των Labels της φόρμας6 και 'τα νέα δεδομένα της σύνδεσης από τα TextBoxes εισάγονται στα ομώνυμα Labels 'της φόρμας4

```

If Form4.Option4.Value = True Then 'sqlserver
    Form4.Label12.Visible = True
    Form4.Label13.Visible = True
    Form4.Label14.Visible = True
    Form4.Label15.Visible = True
    Form4.Label16.Visible = True
    Form4.Label17.Visible = True
    Form4.Label13.Caption = Text1.Text
    Form4.Label15.Caption = Text2.Text
    Form4.Label17.Caption = Text3.Text
ElseIf Form4.Option5.Value = True Then 'mysql
    Label5.Caption = "Location:"
    Form4.Label13.Caption = Text1.Text
    Form4.Label15.Caption = Text2.Text
    Form4.Label17.Caption = Text3.Text
ElseIf Form4.Option6.Value = True Then 'mimer
    Label5.Visible = False
    Text3.Visible = False
    Form4.Label13.Caption = Text1.Text
    Form4.Label15.Caption = Text2.Text
ElseIf Form4.Option8.Value = True Then 'oracle
    Label5.Visible = False
    Text3.Visible = False
    Form4.Label13.Caption = Text1.Text
    Form4.Label15.Caption = Text2.Text
End If
Form4.Enabled = True

```

#### Για την φόρμα "VIEW DATA

```

ElseIf Form7.Visible = True Then
    Form7.Combo1.Enabled = False
    'Ανάλογα με το διακομιστή ορίζονται οι ετικέτες των Labels της φόρμας6 και
    'τα νέα δεδομένα της σύνδεσης από τα TextBoxes εισάγονται στα ομώνυμα Labels
    'της φόρμας7

```

```

If Form7.Combo1.Text = "SQL Server" Then 'sqlserver
    Form7.Label6.Visible = True
    Form7.Label7.Visible = True
    Form7.Label8.Visible = True
    Form7.Label9.Visible = True
    Form7.Label10.Visible = True
    Form7.Label11.Visible = True
    Form7.Label7.Caption = Text1.Text
    Form7.Label9.Caption = Text2.Text
    Form7.Label11.Caption = Text3.Text
ElseIf Form7.Combo1.Text = "MySQL" Then 'mysql
    Label5.Caption = "Location:"
    Form7.Label7.Caption = Text1.Text
    Form7.Label9.Caption = Text2.Text
    Form7.Label11.Caption = Text3.Text

```

```

ElseIf Form7.Combo1.Text = "MimerSQL" Then 'mimer
  Label5.Visible = False
  Text3.Visible = False
  Form7.Label7.Caption = Text1.Text
  Form7.Label9.Caption = Text2.Text
ElseIf Form7.Combo1.Text = "Oracle" Then 'oracle
  Label5.Visible = False
  Text3.Visible = False
  Form7.Label7.Caption = Text1.Text
  Form7.Label9.Caption = Text2.Text
End If
Form7.Enabled = True

```

'Για την φόρμα "EXPORT XML FROM TWO TABLES"

```

ElseIf Form5.Visible = True Then
  'Εάν πατηθεί το CommandButton "Change Login_1"
  'Ανάλογα με το διακομιστή ορίζονται οι ετικέτες των Labels της φόρμας6 και
  'τα νέα δεδομένα της σύνδεσης από τα TextBoxes εισάγονται στα ομώνυμα Labels
  'της φόρμας7

```

'Το Label24 της φόρμας5 παίρνει την τιμή 1 αν πατηθεί το CommandButton "Change Login\_1", ενώ την τιμή 2 αν πατηθεί το CommandButton "Change Login\_2"

```

If Form5.Label24.Caption = "1" Then "Chandge Login_1"
  Form5.Combo1.Enabled = False
  If Form5.Combo1.Text = "SQL Server" Then 'sqlserver
    Form5.Label12.Visible = True
    Form5.Label13.Visible = True
    Form5.Label14.Visible = True
    Form5.Label15.Visible = True
    Form5.Label16.Visible = True
    Form5.Label17.Visible = True
    Form5.Label13.Caption = Text1.Text
    Form5.Label15.Caption = Text2.Text
    Form5.Label17.Caption = Text3.Text
  ElseIf Form5.Combo1.Text = "MySQL" Then 'mysql
    Label5.Caption = "Location:"
    Form5.Label13.Caption = Text1.Text
    Form5.Label15.Caption = Text2.Text
    Form5.Label17.Caption = Text3.Text
  ElseIf Form5.Combo1.Text = "MimerSQL" Then 'mimer
    Label5.Visible = False
    Text3.Visible = False
    Form5.Label13.Caption = Text1.Text
    Form5.Label15.Caption = Text2.Text
  ElseIf Form5.Combo1.Text = "Oracle" Then 'oracle
    Label5.Visible = False
    Text3.Visible = False
    Form5.Label13.Caption = Text1.Text
    Form5.Label15.Caption = Text2.Text

```

```

End If

'Εάν πατηθεί το CommandButton "Change Login_2"
'Ανάλογα με το διακομιστή ορίζονται οι ετικέτες των Labels της φόρμας6 και
'τα νέα δεδομένα της σύνδεσης από τα TextBoxes εισάγονται στα ομώνυμα Labels
'της φόρμας7

ElseIf Form5.Label24.Caption = "2" Then "'Chandge Login_2"
Form5.Combo2.Enabled = False
If Form5.Combo2.Text = "SQL Server" Then 'sqlserver
    Form5.Label18.Visible = True
    Form5.Label19.Visible = True
    Form5.Label20.Visible = True
    Form5.Label21.Visible = True
    Form5.Label22.Visible = True
    Form5.Label23.Visible = True
    Form5.Label19.Caption = Text1.Text
    Form5.Label21.Caption = Text2.Text
    Form5.Label23.Caption = Text3.Text
ElseIf Form5.Combo2.Text = "MySQL" Then 'mysql
    Label5.Caption = "Location:"
    Form5.Label19.Caption = Text1.Text
    Form5.Label21.Caption = Text2.Text
    Form5.Label23.Caption = Text3.Text
ElseIf Form5.Combo2.Text = "MimerSQL" Then 'mimer
    Label5.Visible = False
    Text3.Visible = False
    Form5.Label19.Caption = Text1.Text
    Form5.Label21.Caption = Text2.Text
ElseIf Form5.Combo2.Text = "Oracle" Then 'oracle
    Label5.Visible = False
    Text3.Visible = False
    Form5.Label19.Caption = Text1.Text
    Form5.Label21.Caption = Text2.Text
End If
End If
Form5.Enabled = True

End If
Form6.Hide

End Sub

```

## 7.7 Η ΦΟΡΜΑ “View Data”

Στη φόρμα7(“View Data”), ο χρήστης έχει τη δυνατότητα να εμφανίσει και να δει τις εγγραφές οποιουδήποτε πίνακα ,είτε αυτός ανήκει σε κάποια βάση ,είτε όχι όπως οι πίνακες Paradox και dBASE.Η φόρμα7, φαίνεται παρακάτω στην εικόνα Form7a.

CategoryID	CategoryName	Description
1	Beverages	Soft drinks, coffees, tea
2	Condiments	Sweet and savory sauci
3	Confections	Desserts, candies, and
4	Dairy Products	Cheeses
5	Grains/Cereals	Breads, crackers, pasta
6	Meat/Poultry	Prepared meats
7	Produce	Dried fruit and bean cur
8	Seafood	Seaweed and fish

(εικόνα Form7a)

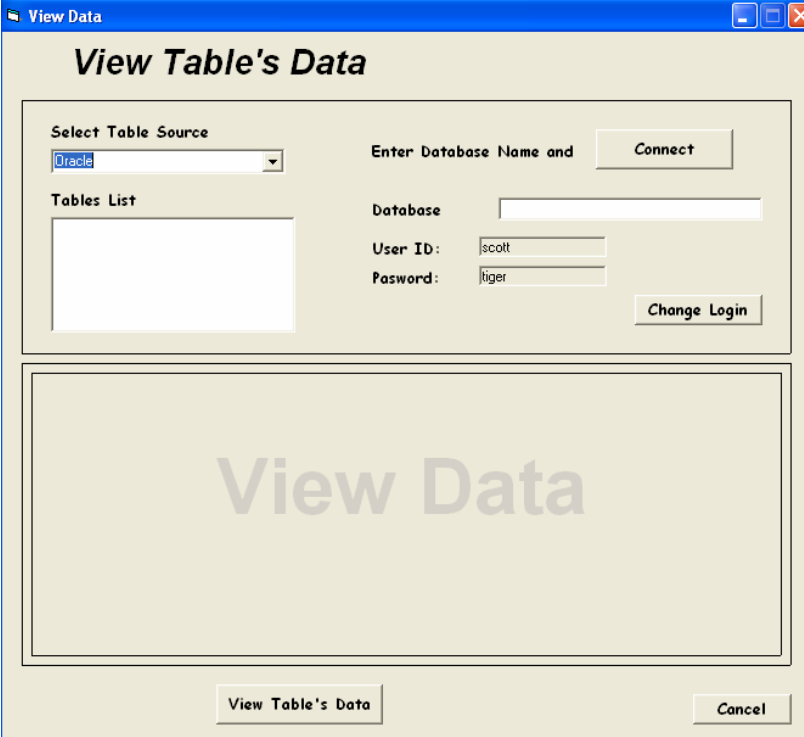
Τα OptionButtonControls έχουν αντικατασταθεί από ένα ComboBox, το οποίο περιέχει τους οχτώ τύπους βάσεων δεδομένων, που υποστηρίζει το πρόγραμμα.Ο χρήστης, πατώντας επάνω του ,έχει τη δυνατότητα να επιλέξει την πηγή δεδομένων του πίνακα.

Για τις τιμές “MS Access”,“Paradox 5.X” ,“Dbase 5” και “SQLite” του ComboBox, η φόρμα4, έχει τη μορφή της εικόνας Form7a.Το CommandButton “Select Table” ή “Select Database”, ανοίγει το CommonDialogControl για την επιλογή της βάσης “.mdb” ή των πινάκων “.dbf” και “.db” αντίστοιχα. Ο κώδικάς του είναι ο ίδιος με τον κώδικα του ομώνυμου CommandButton της φόρμας3.

Για τις βάσεις τύπου Client/Server, όπως “SQL Server”, ”MySQL” ”MimerSQL” και “Oracle Server”,η φόρμα7 έχει τη μορφή της εικόνας Form7b. Το CommandButton “Select Table” ή “Select Database” δεν είναι ενεργό πλέον,όπως και το αντίστοιχο Label.Το CommandButon “Connect” εμφανίζεται για να πραγματοποιηθεί η σύνδεση στο διακομιστή της βάσης.Το όνομα της βάσης για την σύνδεση εισάγεται από το χρήστη στο αντίστοιχο TextBox ,ενώ τα υπόλοιπα στοιχεία (UserID,Password,Server) παρουσιάζονται στα ομώνυμα Labels.

Η δυνατότητα αλλαγής τους ,δίνεται στο χρήστη μέσω του CommandButton “**Change Login**”, το οποίο εμφανίζει την φόρμαβ για εισαγωγή των νέων στοιχείων.

Με την πραγματοποίηση, τέλος, της συνδέσεως, τα ονόματα των πινάκων της βάσης φορτώνονται στο ListBox της φόρμας7, για να επιλέξει ο χρήστης ποιου πίνακα τις εγγραφές θα εμφανίσει.



(εικόνα **Form7b**)

Οι εγγραφές του πίνακα εμφανίζονται σε ένα DataGridViewControl, το οποίο δεν είναι ορατό έως ότου ο χρήστης πατήσει το CommandButton “**View Table’s Data**”. Οι ιδιότητες του “DataSource” και “DataMember” πρέπει να οριστούν ,για να εμφανιστούν οι εγγραφές στο DataGridView.Για το λόγο αυτό, χρησιμοποιούμε ξανά τον DataEnvironment και το αντικείμενό του “Comm1” που είναι ένα DECommand ADO. Στον κώδικα του CommandButton ,ρυθμίζουμε το “Comm1” ανάλογα με τον τύπο βάσης δεδομένων που χρησιμοποιούμε .Συγκεκριμένα ,ορίζουμε τις ιδιότητες ActiveConnection (το ConnectionString ,δηλαδή, της τρέχουσας σύνδεσης) και CommandText (το όνομα του πίνακα) του αντικειμένου DECommand.

Αφού ορίσουμε το αντικείμενο DECommand, θέτουμε το όνομά του (“Comm1”) σαν τιμή για την ιδιότητα “DataMember” του DataGridViewControl, και στην ιδιότητα “DataSource” εισάγουμε το όνομα του DataEnvironment (“DataEnvironment1”).Αυτόματα, οι εγγραφές του επιλεγμένου πίνακα εμφανίζονται στην οθόνη του υπολογιστή.

Παρακάτω παρουσιάζεται και αναλύεται ο κώδικας του CommandButton “**View Table’s Data**”



## “View Table’s Data” CommandButton

```
Private Sub Command5_Click()
```

```
Εμφάνιση του DataGridViewControl
```

```
DataGrid1.Visible = True
```

```
'Ορισμός των ιδιοτήτων CommandText και CommandType του αντικειμένου  
'DECommand
```

```
DataEnvironment1.Commands("Comm1").CommandType = adCmdTable
```

```
DataEnvironment1.Commands("Comm1").CommandTimeout = 10
```

```
'Ορισμός των ιδιοτήτων ActiveConnection και CommandText του αντικειμένου  
'DECommand ανάλογα πάντα με τη βάση και τον επιλεγμένο πίνακα
```

```
If Combo1.Text = "Paradox 5.X" Then 'paradox
```

```
    DataEnvironment1.Commands("Comm1").ActiveConnection =  
    "Driver={Microsoft Paradox Driver (*.db)};" & _  
        "DriverID=538;" & _  
        "Fil=Paradox 5.X;" & _  
        "DefaultDir=" & MyStr2 & ";" & _  
        "Dbq=" & MyStr2 & ";" & _  
        "CollatingSequence=ASCII"
```

```
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
```

```
Elseif Combo1.Text = "dBASE 5" Then 'dbase
```

```
    DataEnvironment1.Commands("Comm1").ActiveConnection =  
    "Provider=MSDASQL.1;Persist Security Info=False;Data Source=dBASE  
Files;Initial Catalog=" & MyStr2 & ""
```

```
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
```

```
Elseif Combo1.Text = "MS Access" Then 'ms_access
```

```
    DataEnvironment1.Commands("Comm1").ActiveConnection =  
    "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False;Data Source=" &  
CommonDialog1.FileName
```

```
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
```

```
Elseif Combo1.Text = "SQL Server" Then 'sql_server
```

```
    DataEnvironment1.Commands("Comm1").ActiveConnection =  
    "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security  
Info=False;Initial Catalog=" & Text1.Text & ""
```

```
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
```

```
Elseif Combo1.Text = "MySQL" Then 'mysql
```

```
    DataEnvironment1.Commands("Comm1").ActiveConnection =  
    "DRIVER={MySQL ODBC 3.51 Driver};" & _  
        "SERVER=" & Label11.Caption & ";" & _  
        "DATABASE=" & Text1.Text & ";" & _  
        "USER=" & Label7.Caption & ";" & _  
        "PASSWORD=" & Label9.Caption & ";"
```

```
    DataEnvironment1.Commands("Comm1").CommandText = List1.Text
```

```
Elseif Combo1.Text = "MimerSQL" Then 'mimer
```

```

DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=MSDASQL.1;Password=" & Label9.Caption & ";Persist Security
Info=True;User ID=" & Label7.Caption & ";Data Source=" & Text1.Text & ""
DataEnvironment1.Commands("Comm1").CommandText = List1.Text
Elseif Combo1.Text = "SQLite" Then 'SQLite
DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=MSDASQL.1;Persist Security Info=False;Data
Source=SQLiteDatasource;Extended Properties="DSN=SQLite
Datasource;Database=" & CommonDialog1.FileName & ";
StepAPI=0;Timeout=1000;NoWCHAR=0"
DataEnvironment1.Commands("Comm1").CommandText = List1.Text

Elseif Combo1.Text = "Oracle" Then 'Oracle
DataEnvironment1.Commands("Comm1").ActiveConnection =
"Provider=OraOLEDB.Oracle.1;Persist Security Info=False;User ID=" &
Label7.Caption & ";Password=" & Label9.Caption & ";Data Source=" & Text1.Text
& ""
DataEnvironment1.Commands("Comm1").CommandText = List1.Text
End If

'Ανοιγμα του αντικειμένου rsComm1 που είναι αντικείμενο recordset
'Ορισμός των ιδιοτήτων DataSource και DataMember του DataGridView
DataEnvironment1.rsComm1.Open DataEnvironment1.Commands("Comm1"),
adOpenStatic, adLockReadOnly
DataGridView1.DataMember = "Comm1"
Set DataGridView1.DataSource = DataEnvironment1

End Sub

```