

**ΚΑΤΑΣΚΕΥΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΝΑΛΥΣΗΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ ΨΗΦΙΑΚΩΝ
ΒΙΝΤΕΟ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕ ΓΡΗΓΟΡΗ ΑΝΑΚΤΗΣΗ ΒΑΣΗ
ΠΕΡΙΕΧΟΜΕΝΟΥ ΚΑΙ ΣΥΡΡΑΦΗ ΑΚΟΛΟΥΘΙΩΝ ΒΙΝΤΕΟ ΒΑΣΗ ΚΡΙΤΗΡΙΩΝ.**

ΤΟΥ

Λιάκου Χαράλαμπου

Πτυχιακή εργασία που υποβάλλεται προς
μερική εκπλήρωση των απαιτήσεων για την
απόκτηση του πτυχίου

Τεχνολογικό Εκπαιδευτικό Ίδρυμα Σερρών

ΜΑΡΤΙΟΣ 2005

Εγκρίθηκε από τον _____

Ημερομηνία _____

ΠΕΡΙΟΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	ΣΕΛ 3
1. ΕΙΣΑΓΩΓΗ. ΣΚΟΠΟΣ ΚΑΙ ΠΟΙΕΣ ΑΝΑΓΚΕΣ ΚΑΛΥΠΤΕΙ.	ΣΕΛ 4
2. ΠΟΛΥΜΕΣΑ ΚΑΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.....	ΣΕΛ 6
3. VIDEO DATABASES.	ΣΕΛ 7
3. 1. ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ VIDEO ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ.	ΣΕΛ 8
3. 2. 1. VIDEO ΚΑΤΑΤΜΗΣΗ.....	ΣΕΛ 9
3. 2. 2. VIDEO ΠΕΡΙΛΗΨΗ.....	ΣΕΛ 10
3. 3. ΕΥΡΕΤΗΡΙΑΣΗ – ΕΡΩΤΗΜΑ - ΑΝΑΚΤΗΣΗ.....	ΣΕΛ 13
3. 4. ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΚΤΥΣΗΣ ΣΤΟΙΧΕΙΩΝ VIDEO	ΣΕΛ 14
3. 5. Η ΑΝΑΔΥΣΗ ΠΡΟΤΥΠΩΝ ΣΥΜΠΙΕΣΗΣ ΚΑΙ Ο ΡΟΛΟΣ ΤΟΥΣ ΣΤΙΣ VIDEO ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	ΣΕΛ 15
3. 5. 1. ΑΛΓΟΡΙΘΜΟΙ ΣΥΜΠΙΕΣΗΣ	ΣΕΛ 15
3. 6. MPEG	ΣΕΛ 16
4. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ	ΣΕΛ 17
4. 1. ΜΟΝΤΕΛΟ E-R (ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΗΣΕΩΝ)	ΣΕΛ 19
4. 2. ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ	ΣΕΛ 22
4.3. ΜΕΤΑΤΡΟΠΗ ΔΙΑΓΡΑΜΜΑΤΟΣ E–R ΣΤΟ ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ	ΣΕΛ 23
4. 4. Η ΓΛΩΣΣΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ SQL.....	ΣΕΛ 27
5. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ C++ BUILDER.....	ΣΕΛ 32
5. 1. ΓΕΝΙΚΑ.....	ΣΕΛ 32
3. 2. ΟΠΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	ΣΕΛ 32
3. 3. ΣΥΝΑΡΤΗΣΕΙΣ - ΑΝΤΙΚΕΙΜΕΝΑ.....	ΣΕΛ 33
6. ΣΥΝΔΕΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.....	ΣΕΛ 35
7. ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΡΓΑΣΙΑΣ ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ...ΣΕΛ 45	
7. 1. ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΑΘΕ ΚΟΥΜΠΙΟΥ ΚΑΙ ΦΟΡΜΑΣ	ΣΕΛ 50
8. ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ – ΠΑΡΑΔΕΙΓΜΑΤΑ	ΣΕΛ 74
9. ΣΥΜΠΕΡΑΣΜΑΤΑ. ΧΡΗΣΕΙΣ.....	ΣΕΛ 87
10. ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ.....	ΣΕΛ 89
11. ΛΙΣΤΑ ΕΙΚΟΝΩΝ	ΣΕΛ 90

ΨΗΦΙΑΚΕΣ ΒΙΝΤΕΟ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

ΚΑΤΑΣΚΕΥΗ ΠΡΟΓΡΑΜΜΑΤΟΣ ΑΝΑΛΥΣΗΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗΣ ΨΗΦΙΑΚΩΝ ΒΙΝΤΕΟ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΜΕ ΓΡΗΓΟΡΗ ΑΝΑΚΤΗΣΗ ΒΑΣΗ ΠΕΡΙΕΧΟΜΕΝΟΥ ΚΑΙ ΣΥΡΡΑΦΗ ΑΚΟΛΟΥΘΙΩΝ ΒΙΝΤΕΟ ΒΑΣΗ ΚΡΙΤΗΡΙΩΝ.

ΠΕΡΙΛΗΨΗ

Σκοπός της πτυχιακής εργασίας είναι καταρχήν η κατασκευή περιβάλλοντος διαχείρισης πολλαπλών ψηφιακών αρχείων video (mpeg, avi) με κοινά στοιχεία και η δυνατότητα γρήγορης ανάλυσης και καταχώρισης των κοινών χαρακτηριστικών σε κανονικοποιημένες βάσεις δεδομένων.

Έτσι, σε πρώτη αυτή φάση, σκοπός μας είναι να σχολιάζουμε ακολουθίες βίντεο με τεχνικές κατάτμησης βίντεο, και στη συνέχεια να τα καταχωρούμε στη βάση δεδομένων ανάλογα με το περιεχόμενό τους.

Σε δεύτερη φάση αφού χαρακτηριστούν τα αρχικά βίντεο με δυνατότητα χειρονακτικού προσδιορισμού περιεχομένων και αφού καταχωρηθούν, τότε με χρήση ερωτημάτων SQL γίνεται αναζήτηση και ανάκτηση βάση περιεχομένου, συρραφή και προβολή περίληψης του βίντεο δηλ. των σκηνών του video που θα πληρούν τα κριτήρια των σύνθετων ερωτημάτων.

1. ΕΙΣΑΓΩΓΗ. ΣΚΟΠΟΣ ΚΑΙ ΠΟΙΕΣ ΑΝΑΓΚΕΣ ΚΑΛΥΠΤΕΙ.

Σύστημα Βάσης Δεδομένων (database system), είναι ένα ηλεκτρονικό σύστημα τήρησης εγγράφων, δηλαδή ένα σύστημα για υπολογιστές, που σκοπός του είναι να τηρεί και να δίνει αυτές τις πληροφορίες όταν του ζητούνται.

Ως Βάση Δεδομένων μπορούμε να θεωρήσουμε ένα είδος ηλεκτρονικής αρχειοθέτησης, ένα χώρο δηλαδή για την αποθήκευση μιας συλλογής ηλεκτρονικών αρχείων δεδομένων. Βάση Δεδομένων λοιπόν, είναι μία ολοκληρωμένη συλλογή συσχετιζόμενων δεδομένων, όπου με τον όρο δεδομένα εννοούμε γνωστά γεγονότα που μπορούν να καταγραφούν και που έχουν κάποια υπονοούμενη σημασία.

Ο τομέας των Ψηφιακών Βίντεο Βάσεων Δεδομένων, Digital Video Databases, είναι ένας χώρος όπου τελευταία συγκεντρώνει το ενδιαφέρον μελετών για εφαρμογές ανάλυσης και διαχείρισης αυτού του τεράστιου όγκου πληροφοριών (ειδήσεις, ταινίες, σπορ, κλπ) που συγκεντρώνεται καθημερινά σε ψηφιακή μορφή.

Οι ψηφιακές βίντεο βάσεις δεδομένων είναι μια σύγχρονη μορφή των απλών βάσεων δεδομένων όπου οι τελευταίες βρίσκουν εφαρμογή σε πολλά σημεία της σύγχρονης κοινωνίας. Μια μορφή Βάσεων Δεδομένων θεωρείται και ένας απλός οικιακός τηλεφωνικός κατάλογος που περιέχει εγγραφές ονομάτων, τηλεφώνων, διευθύνσεις κλπ.

Αργότερα με την εισαγωγή των Η/Υ στην ζωή μας, οι Βάσεις Δεδομένων έγιναν αναπόσπαστο κομμάτι αφού ακόμα και οι Η/Υ βασίζουν την λειτουργία τους και αποσπούν στοιχεία μέσα από τεράστιες βάσεις δεδομένων. Τέλος, τράπεζες, εμπορικά καταστήματα, μικρές και μεγάλες επιχειρήσεις και κάθε είδους εμπορική συναλλαγή βασίζονται πλέον σε ηλεκτρονικές και ψηφιακές βάσεις δεδομένων.

Για παράδειγμα, σε έναν ποδοσφαιρικό αγώνα υπάρχουν οι φάσεις με τα goal, φάσεις με out, πέναλτι, corner, τραυματισμού κλπ. Η επανάληψη των παραπάνω φάσεων μπορεί να είναι από μηδενικές μέχρι πάρα πολλές. Αφού λοιπόν, χαρακτηρίσουμε τα παιδιά κάνοντας την αλλαγή χειρονακτικά, όπως

θα δούμε παρακάτω, αρχίζουμε και καταχωρούμε σύμφωνα και με το χρόνο τις όμοιες φάσεις στο ανάλογο πεδίο.

Δηλαδή, έστω πως έχουμε το προηγούμενο παράδειγμα με τον ποδοσφαιρικό αγώνα και ότι έστω έχουμε τρία goal στον αγώνα, επτά Out, δύο πέναλτι, δέκα corner και τρεις τραυματισμούς. Με τη χρήση ερωτημάτων έχουμε τη δυνατότητα να δούμε για παράδειγμα μόνο τα γκολ του αγώνα χωρίς βέβαια να ψάχνουμε από την αρχή όλο τον αγώνα, παρά μόνο καλώντας μέσα από τη βάση δεδομένων τις σκηνές που αφορούν μόνο τα γκολ και απλά προβάλλοντας τα το ένα μετά το άλλο.

Η παρούσα εργασία μπορεί να βρει εφαρμογή σε πάρα πολλούς τομείς. Σε τηλεοπτικούς σταθμούς για τα δελτία ειδήσεων χωρίζοντας τα θέματα τους σε ενότητες, σε εκπομπές οποιουδήποτε περιεχομένου καταχωρώντας π. χ. τις συνεντεύξεις των καλεσμένων κάθε εκπομπής. Επίσης, μπορεί να χρησιμοποιηθεί από αθλητικούς συλλόγους και ομάδες. Χρησιμοποιώντας το πρόγραμμα μπορεί να βελτιώσει την εκπαίδευση των αθλητών τους παρουσιάζοντας τις κάθε φάσεις ξεχωριστά. Βέβαια, μπορεί να χρησιμοποιηθεί και από φοιτητές, οικογένειες, μαθητές καταχωρώντας οικογενειακά videos κλπ.

Παρακάτω δίνεται λεπτομερής περιγραφή της όλης εργασίας και παρουσίαση των δυνατοτήτων της.

Θα παρουσιασθεί ο σχεδιασμός των Βάσεων Δεδομένων με μοντέλα οντοτήτων και συσχετίσεων καθώς και η μετατροπή του διαγράμματος αυτού στο σχεσιακό μοντέλο.

Για την εύκολη διαχείριση των Βάσεων Δεδομένων χρησιμοποιήσαμε προγραμματισμό με C++ Builder ώστε να δημιουργηθεί ένα εύχρηστο και δημιουργικό περιβάλλον.

Επιπλέον αναφέραμε τα χαρακτηριστικά των Video Data Bases που αποτελούν την πηγή ανάπτυξης της εφαρμογής μας.

2. ΠΟΛΥΜΕΣΑ ΚΑΙ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.

Τα συστήματα διαχείρισης Βάσεων Δεδομένων χρησιμοποιούνται για την αποθήκευση, αναζήτηση, ταξινόμηση και οργάνωση κάθε τύπου ψηφιακής πληροφορίας από Βάσεις Δεδομένων που συντίθεται από πίνακες. Στα πεδία των πινάκων των Βάσεων Δεδομένων μπορούν να αποθηκευθούν σε ψηφιακή μορφή κείμενα, ήχοι εικόνες και στη παρούσα φάση videos.

Τα συστήματα διαχείρισης Βάσεων Δεδομένων χρησιμοποιούνται, κατά κανόνα, ως εξυπηρετητές σε εφαρμογές πελάτη / εξυπηρετητή, στις οποίες μία εφαρμογή πολυμέσων παίζει το ρόλο του πελάτη. Η εφαρμογή πολυμέσων μπορεί να έχει δημιουργηθεί με οποιαδήποτε εργαλείο, αλλά πρέπει να έχει την δυνατότητα να αντλεί τα δεδομένα τα οποία είναι πολυμεσικά.

Τα συστήματα διαχείρισης Βάσεων Δεδομένων συνοδεύονται συνήθως από πρόσθετα εργαλεία δημιουργίας εφαρμογών, τα οποία είναι κατάλληλα για την δημιουργία διοικητικοοικονομικών πληροφοριών. Τα εργαλεία αυτά είναι εμπλουτισμένα με δυνατότητες παρουσίασης πολυμεσικών στοιχείων σε φόρμες ή αναφορές. Έτσι, οι εφαρμογές παροχής πληροφοριών μπορούν να γίνουν πιο εργονομικές.

Η εφαρμογή που αναπτύχθηκε ανήκει στην κατηγορία των πολυμεσικών βάσεων δεδομένων και μπορεί να αποθηκεύει τις πληροφορίες σε βάσεις δεδομένων και να διαχειρίζεται πολυμεσικές πληροφορίες, στην περίπτωση αυτή βίντεο.

3. VIDEO DATABASES.

Το κίνητρο για την χρησιμοποίηση των video Βάσεων Δεδομένων έγκειται στο γεγονός ότι το τεράστιο μέγεθος των τηλεοπτικών αυτών αρχείων είναι διαθέσιμο στις επιχειρήσεις και τα άτομα αφού ικανοποιεί βασικές προϋποθέσεις, είναι φθηνότερη λύση, όλο το διαθέσιμο υλικό που έχει αποκτηθεί, επεξεργαστεί και αποθηκευθεί είναι πιο ευέλικτο πλέον, δημιουργεί κυρίαρχα δίκτυα, εκλαϊκεύει το Διαδίκτυο και τον Ιστό και βοηθάει στη ανάπτυξη των νέων προϊόντων, υπηρεσιών και τύπων αποθήκευσης (TiVo, VoD, DVD).

Συνδυάζοντας τις εικόνες, τον ήχο, το κείμενο, και τη κίνηση, το video είναι αναμφισβήτητα ο πλουσιότερος τύπος πολυμέσων. Διάφορες προκλήσεις γεννιούνται κατά την οικοδόμηση των τηλεοπτικών Βάσεων Δεδομένων:

- Δεδομένου ότι το ακατέργαστο τηλεοπτικό στοιχείο μόνο του έχει περιορίσει τη χρησιμότητα του, η ανάγκη για το σχολιασμό είναι αποφασιστική. Ο χειρωνακτικός σχολιασμός είναι κουραστική, χρονοβόρα αλλά υποκειμενική διαδικασία.
- Έχουν βρεθεί εκείνοι οι αλγόριθμοι και συστήματα που θα επιτρέπουν τους (ημι-) αυτόματους τρόπους περιγραφής και οργάνωσης με σκοπό να ρυθμίζουν τα τηλεοπτικά στοιχεία με τη καλύτερη δυνατή έννοια περιεχομένου.
- Τεράστια ποσότητα δεδομένων: video, ήχος, κείμενο.
- Διανεμημένη, όχι πάντα δομημένη, ευρεία ποικιλία αποθηκών.
- Κάθε τηλεοπτικό πρόγραμμα έχει τους δικούς του “κανόνες” και “αρχές”.
- Η κατανόηση του video είναι εξαρτημένη με το γύρω περιβάλλον.
- Υπάρχουν διαφορετικοί χρήστες, διαφορετικές πλατφόρμες, διαφορετικές ανάγκες.

Λαμβάνοντας υπόψη τις παραπάνω ανάγκες, οδηγηθήκαμε στην ανάπτυξη της εφαρμογής μας η οποία επιτρέπει την οργάνωση τεράστιου όγκου πληροφοριών και χειρωνακτική καταγραφή του σχολιασμού.

3. 1. ΒΑΣΙΚΑ ΣΤΟΙΧΕΙΑ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ VIDEO ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ.

Ο αρχικός στόχος ενός συστήματος διαχείρισης Βίντεο Βάσεων Δεδομένων είναι να εξασφαλίσει τη ψευδό-τυχαία πρόσβαση στα διαδοχικά τηλεοπτικά στοιχεία. Αυτό επιτυγχάνεται ως εξής:

- Διαιρείται ένα video σε τμήματα
- Θέτονται σαν δείκτες αυτά τα κομμάτια
- Αντιπροσώπευση αυτών των δεικτών με τέτοιο τρόπο που να επιτρέπει το εύκολο “ξεφύλλισμα” και την ανάκτηση.

Ένα τέτοιο σύστημα ουσιαστικά είναι μία Βάση Δεδομένων των δεικτών σε μία καταγραφή των ακολουθιών βίντεο.

Το ακατέργαστο τηλεοπτικό στοιχείο πρέπει να διαμορφωθεί, να συνταχθεί και να γίνει δομημένο.

Τα ερωτήματα και η επόμενη ανάκτηση εκτελούν συνήθως όμοιους αλγορίθμους βασισμένους σε αλγορίθμους ανάκτησης.

Το σχέδιο GUI είναι εξαιρετικά σημαντικό.

Ένα τηλεοπτικό πρότυπο πρέπει να υποστηρίξει το σχολιασμό (προσθέτοντας τα μεταδεδομένα) του τηλεοπτικού προγράμματος. Τρεις κατηγορίες μεταδεδομένων υπάρχουν:

- Εξαρτημένου περιεχομένου μεταδεδομένα
- Περιγραφικού περιεχομένου μεταδεδομένα
- Ανεξαρτήτου περιεχομένου μεταδεδομένα

Η εφαρμογή μας ανήκει στην κατηγορία των Βίντεο Βάσεων Δεδομένων με περιγραφικού περιεχομένου μεταδεδομένα. Μπορούμε επίσης να τονίσουμε το γεγονός ότι η εφαρμογή μπορεί να λειτουργήσει και σαν φορητή βιντεοβάση δεδομένων, να εγκατασταθεί σε οποιοδήποτε υπολογιστικό σύστημα με λειτουργικό σύστημα Windows, διαθέτει φιλικό περιβάλλον GUI για τους χρήστες οι οποίοι μπορούν να έχουν όλα τα αρχεία βίντεο που επιθυμούν σε φορητά συστήματα Η/Υ. Το γεγονός ότι για να αποθηκευτούν τα δεδομένα στους πίνακες της βάσης δεν απαιτεί την λειτουργία συστήματος διαχείρισης βάσεων δεδομένων διακομιστή (server)

επιτρέπει την εύκολη εγκατάσταση και λειτουργία της εφαρμογής σε φορητά συστήματα.

3. 2. VIDEO ΚΑΤΑΤΜΗΣΗ ΚΑΙ ΠΕΡΙΛΗΨΗ.

VIDEO ΚΑΤΑΤΜΗΣΗ είναι η χρονική κατάτμηση του τηλεοπτικού περιεχομένου σε μικρότερες μονάδες.

Οι τηλεοπτικές τεχνικές ανάλυσης εξάγουν τις δομικές πληροφορίες από το τηλεοπτικό πρόγραμμα με την ανίχνευση των χρονικών ορίων, είτε υποκειμενικά από τον χρήστη, είτε από αλγόριθμους αναγνώρισης προτύπων στην εικόνα.

VIDEO ΠΕΡΙΛΗΨΗ είναι η τηλεοπτική περιληπτική παρουσίαση της πληροφορίας η οποία προσπαθεί να παρουσιάσει μια εικονογραφική περίληψη μιας τηλεοπτικής ακολουθίας, σε μια συμπαγέστερη μορφή, που αποβάλλει τον πλεονασμό.

3. 2. 1. VIDEO ΚΑΤΑΤΜΗΣΗ.

Η τηλεοπτική κατάτμηση είναι η διαδικασία της τμηματοποίησης των τηλεοπτικών ακολουθιών, στις σκηνές που μπορούν να υποδιαιρεθούν περαιτέρω στα μεμονωμένα πλάνα.

Τα πλάνα είναι συνήθως το μικρότερο αντικείμενο ενδιαφέροντος και ανιχνεύονται είτε αυτόματα είτε χειρονακτικά αφού αντιπροσωπεύονται από key - frames.

Η τηλεοπτική κατάτμηση μπορεί να εμφανιστεί είτε σε επίπεδο “πλάνου” είτε σε επίπεδο “σκηνής”.

Η ανίχνευση στιγμιότυπου ή σε επίπεδο “πλάνου” είναι η διαδικασία της μετάβασης μεταξύ δύο διαδοχικών πλάνων, έτσι ώστε να συγκεντρώνεται μια ακολουθία πλαισίων που ανήκουν σε ένα πλάνο. Υπάρχουν δύο τύποι τέτοιων μεταβάσεων, οι απότομες βαθμιαίες μεταβάσεις και οι βαθμιαίες αλλαγές (εμφάνιση, εξαφάνιση, διάλυση).

Η ανίχνευση σε επίπεδο “σκηνής” στη τηλεοπτική κατάτμηση είναι: η αυτόματη ανίχνευση των σημασιολογικών ορίων (σε αντιδιαστολή με τα φυσικά όρια) στα πλαίσια ενός τηλεοπτικού προγράμματος αποτελεί ένα πιο προκλητικό στόχο και το αντικείμενο της τρέχουσας έρευνας. Η λύση της απαιτεί ένα υψηλότερο επίπεδο ανάλυσης περιεχομένου. Τρεις κύριες στρατηγικές έχουν υιοθετηθεί στη διαδικασία αυτή της τηλεοπτικής κατάτμησης:

- Βασισμένο στους κανόνες παραγωγής ταινιών για να ανιχνευθούν οι τοπικές (χρονικές) ενδείξεις της μακροσκοπικής αλλαγής.
- Χρονικά περιορισμένη συγκέντρωση: οι εργασίες κάτω από τη λογική που αφορούσε σημασιολογικά το περιεχόμενο τείνουν να εντοπιστούν εγκαίρως.
- Πρότυποι αλγόριθμοι: στηριζόμενοι στα συγκεκριμένα δομικά πρότυπα για τα προγράμματα των οποίων χρονικές δομές είναι συνήθως πολύ άκαμπτες και προβλέψιμες, όπως οι ειδήσεις και ο αθλητισμός.

3. 2. 2. VIDEO ΠΕΡΙΛΗΨΗ.

Η τηλεοπτική περίληψη εστιάζει στη περιληπτική παρουσίαση της πληροφορίας με σκοπό την εύρεση ενός μικρότερου συνόλου εικόνων για να αντιπροσωπεύσει το οπτικό περιεχόμενο και την παρουσίαση των key-frames στο χρήστη.

Μια τέτοια διαδικασία, επίσης γνωστή και ως στατικό storyboard (στατικό πίνακα με διάταξη σκηνών έργου), είναι μια συλλογή των εμφανή ακόμα εικόνων ή key-frames που παράγονται από το video.

Το μεγαλύτερο μέρος της έρευνας της περιληπτικής παρουσίασης της πληροφορίας, περιλαμβάνει την εξαγωγή key-frames και την ανάπτυξη ενός “φυλλομετρήματος” στην κοινή επιφάνεια που αντιπροσωπεύει καλύτερα το αρχικό video.

Τα πλεονεκτήματα μιας τέτοιας αντιπροσώπησης είναι τα εξής:

- Οι περιλήψεις αυτού του είδους μπορούν να δημιουργηθούν πολύ γρηγορότερα από τις κινούμενες περιλήψεις εικόνας, δεδομένου ότι δεν είναι απαραίτητο κανένας χειρισμός του ήχου ή των πληροφοριών κειμένων.
- Η χρονική διάταξη των αντιπροσωπευτικών πλαισίων μπορεί να επιδειχθεί έτσι ώστε οι χρήστες να μπορούν να συλλάβουν τα αποσπάσματα ακόμα πιο γρήγορα.
- Οι εξαγόμενες εικόνες είναι διαθέσιμες για εκτύπωση, εφόσον επιδιώκονται.

Μία εναλλακτική λύση της αντιπροσώπευσης της παραπάνω μεθόδου είναι τα “Video Skims”:

Καθορισμός: γειτονικά video clips που αποτελούνται από μια συλλογή των ακολουθιών εικόνας και του αντίστοιχου ήχου, εξάγονται από την αρχική πιο μακροχρόνια τηλεοπτική ακολουθία.

Οι Video Skims αντιπροσωπεύουν μια χρονική αφαίρεση πολυμέσων που παίζεται παρά αντιμετωπίζεται στατικά. Αποτελούνται από τις πιο σχετικές φράσεις, τις προτάσεις, και τις ακολουθίες εικόνας.

Στόχος του είναι να παρουσιάσει την αρχική τηλεοπτική ακολουθία σε ένα μικρότερο χρονικό μέγεθος. Δύο βασικοί τύποι Video Skims υπάρχουν:

- Συνοπτικές ακολουθίες οι οποίες χρησιμοποιούνται για να παρέχουν στον χρήστη μια ιδέα των κυριοτέρων τηλεοπτικών σημείων.
- Οι ακολουθίες των σημαντικότερων σημείων όπου περιέχετε μόνο τα πιο ενδιαφέροντα μέρη μιας τηλεοπτικής ακολουθίας.

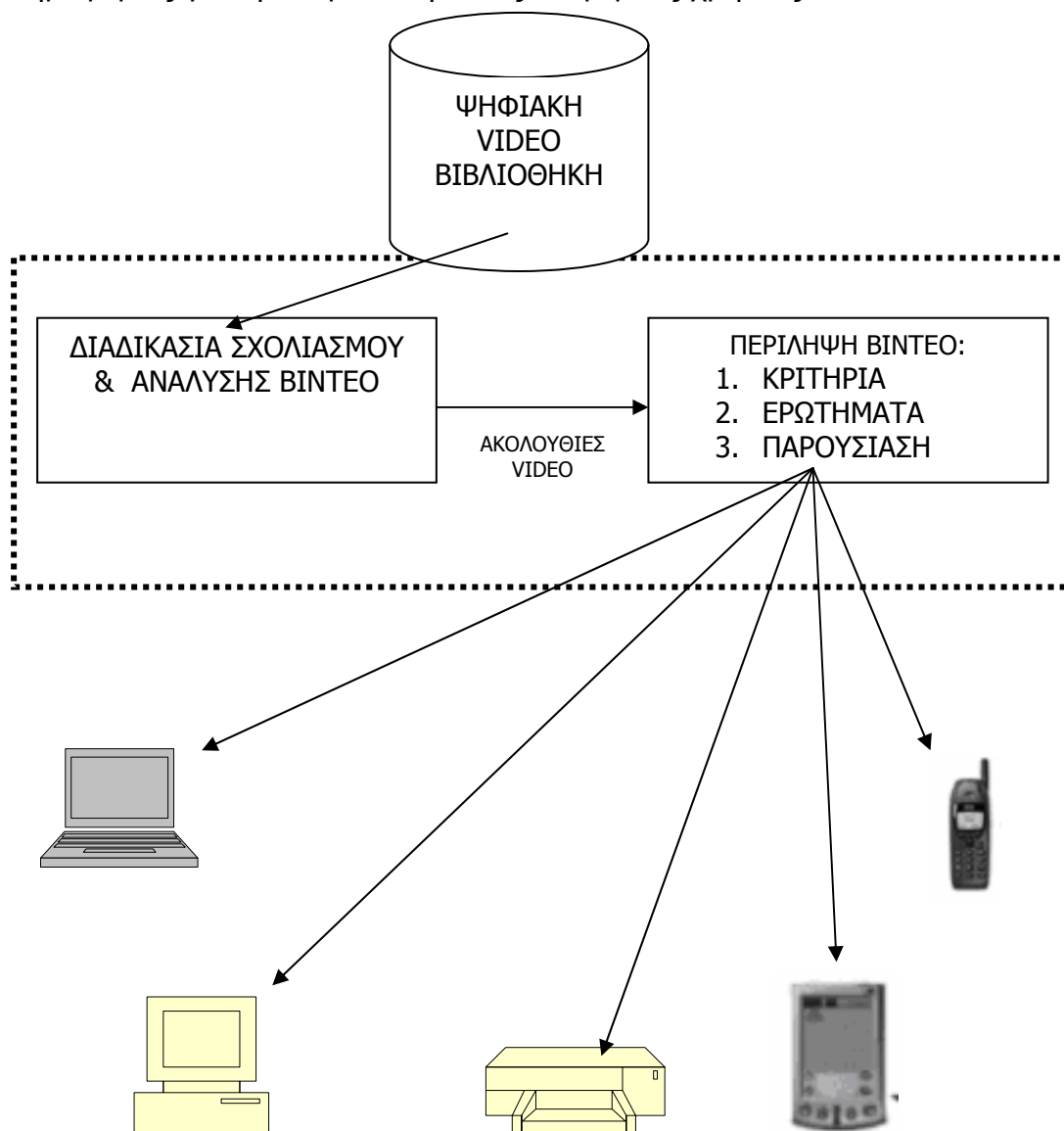
Η επιλογή των κυριοτέρων σημείων από μια τηλεοπτική ακολουθία είναι μια υποκειμενική διαδικασία. Η μεγαλύτερη υπάρχουσα σε βαθμό διαδικασία εστίασης στην εργασία βρίσκεται στην παραγωγή των συνοπτικών ακολουθιών.

Μια πολύ σημαντική πτυχή της τηλεοπτικής περιληπτικής παρουσίασης της πληροφορίας είναι η ανάπτυξη του περιβάλλοντος μεταξύ του χρήστη και των ενδιάμεσων στοιχείων που αντιπροσωπεύουν καλύτερα την αρχική τηλεοπτική ακολουθία.

Οι εναλλαγές μεταξύ των διαφορετικών επιπέδων και των τύπων αφαιρέσεων που παρουσιάζονται στο χρήστη είναι οι εξής:

- Οι περισσότεροι συμπυκνώνουν την αφαίρεση, όπου μελετούν ελάχιστα την όλη ακολουθία αλλά η ποσότητα της πληροφορίας που λαμβάνει δεν είναι αρκετή και έτσι βγάζει βιαστικά και ανακριβή συμπεράσματα για την ακολουθία.
- Η περισσότερο λεπτομερής διαδικασία αφαίρεσης μπορεί να εμφανίσει στο χρήστη αρκετές πληροφορίες για να κατανοήσει το νόημα της τηλεοπτικής ακολουθίας.

Παρακάτω, στην Εικόνα 01, φαίνεται η περιληπτική παρουσίαση της πληροφορίας για την παράδοση στους διάφορους χρήστες.



Εικόνα 01

3. 3. ΕΥΡΕΤΗΡΙΑΣΗ – ΕΡΩΤΗΜΑ - ΑΝΑΚΤΗΣΗ.

Συγκρίνοντας το παραδοσιακό κείμενο DBMS με αυτό της τηλεοπτικής ευρετηρίασης θα συναντήσει κανείς ότι είναι πιο δύσκολο και σύνθετο.

- Ενώ στο παραδοσιακό DBMS, στοιχεία επιλέγονται συνήθως βασισμένα σε μια ή περισσότερες μοναδικές ιδιότητες (βασικοί τομείς), δεν είναι ούτε σαφές ούτε εύκολο να αποφασιστεί όσον αφορά ένα τηλεοπτικό στοιχείο τι να επιλέξει για να συντάξει ένα ευρετήριο.
- Αντίθετα από τα στοιχεία κειμένου, τα τηλεοπτικά στοιχεία που παράγουν τους δείκτες τηλεοπτικών στοιχείων είναι αυτόματα πολύ πιο δύσκολα.

Το παρόν κομμάτι που αναφέρεται στην τηλεοπτική ευρετηρίαση μπορεί να ταξινομηθεί σε δύο κατηγορίες: Σχολιασμός και Γνωρίσματα.

ΣΧΟΛΙΑΣΜΟΣ.

Ο σχολιασμός είναι συνήθως μια χειρωνακτική διαδικασία που εκτελείται από έναν πεπειραμένο χρήστη. Η κοινή τεχνική απαιτεί τον ορισμό λέξης - κλειδί στις τηλεοπτικές τεχνικές ευρετηρίασης τμημάτων (πλάνα).

Στο σχολιασμό ο χρήστης ενδιαφέρεται πρώτιστα για την επιλογή των λέξεων κλειδιών, των δομών δεδομένων και των διεπαφών για να διευκολύνει την προσπάθεια του ίδιου.

Οι λέξεις κλειδιά ενώ είναι πολύ χρήσιμες στον σχολιασμό, παρουσιάζουν κάποια μειονεκτήματα όπως:

1. Οι λέξεις κλειδιά δεν εκφράζουν τις χωρικές και χρονικές σχέσεις.
2. Οι λέξεις κλειδιά δεν μπορούν να αντιπροσωπεύσουν πλήρως τις σημασιολογικές πληροφορίες και δεν υποστηρίζουν την κληρονομιά, ομοιότητα ή το συμπέρασμα μεταξύ των περιγραφών.
3. Οι λέξεις κλειδιά δεν περιγράφουν τις σχέσεις μεταξύ των περιγραφών.
4. Οι λέξεις κλειδιά δεν κλιμακώνονται.

Παρ' όλα αυτά όμως παραμένουν ο κλασσικός και σημαντικότερος τρόπος σχολιασμού των video.

Εναλλακτικές λύσεις των λέξεων κλειδιά όσο αφορά τον σχολιασμό είναι οι παρακάτω:

- Πολυστρωματικός, εικονικός σχολιασμός
- Τμηματικός σχολιασμός.
- Χωρική χρονική λογική

Στη παρούσα εργασία ο σχολιασμός είναι και εδώ χειρονακτική διαδικασία αφού ο χρήστης με δική του επέμβαση κάνει τις αλλαγές και ουσιαστικά ο ίδιος χαρακτηρίζει το video.

ΓΝΩΡΙΣΜΑΤΑ.

Η δεύτερη κατηγορία επιτρέπει την πλήρως αυτοματοποιημένη ευρετηρίαση ενός τηλεοπτικού προγράμματος βασισμένου στο περιεχόμενό της.

Στηρίζεται στις τεχνικές επεξεργασίας εικόνας για να εξαχθούν τα βασικά οπτικά χαρακτηριστικά γνωρίσματα (χρώμα, σύσταση, κίνηση αντικειμένου, κ. λ. π...) από τα τηλεοπτικά στοιχεία και χρησιμοποιεί αυτά τα χαρακτηριστικά γνωρίσματα για να χτίσει τους δείκτες. Κύριο πρόβλημα, το σημασιολογικό χάσμα. Οι τεχνικές αυτές αποτελούν μία πρόκληση για την εξέλιξη της εφαρμογής μας σε συγκεκριμένες κατηγορίες video όπως για παράδειγμα ειδήσεις.

3. 4. ΔΙΑΔΙΚΑΣΙΑ ΑΝΑΚΤΗΣΗΣ ΣΤΟΙΧΕΙΩΝ VIDEO.

1. Ο χρήστης διευκρινίζει μια ερώτηση χρησιμοποιώντας τους πόρους GUI.
2. Η ερώτηση υποβάλλεται σε επεξεργασία και αξιολογείται.
3. Η αξία ή το χαρακτηριστικό γνώρισμα που έχει χρησιμοποιείται για να ταιριάξει και να ανακτήσει τα στοιχεία του βίντεο που αποθηκεύονται στη Βάση Δεδομένων.

4. Η ακολουθία βίντεο που προκύπτει παρουσιάζεται στην οθόνη του χρήστη για μελέτη, εξέταση και (προαιρετικά) τον καθαρισμό ερώτησης (ανατροφοδότηση σχετικότητας).

3. 5. Η ΑΝΑΔΥΣΗ ΠΡΟΤΥΠΩΝ ΣΥΜΠΙΕΣΗΣ ΚΑΙ Ο ΡΟΛΟΣ ΤΟΥΣ ΣΤΙΣ VIDEO ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.

Η θέσπιση τηλεοπτικών προτύπων συμπίεσης και κωδικοποίησης έχει βοηθήσει την πρόοδο σε αυτόν τον τομέα. Τα ευρύτερα χρησιμοποιημένα τηλεοπτικά πρότυπα πηγάζουν από:

- Interactive video communications:
- Entertainment and digital TV: MPEG-1, 2, κλπ.

3. 5. 1. ΑΛΓΟΡΙΘΜΟΙ ΣΥΜΠΙΕΣΗΣ VIDEO.

Οι αλγόριθμοι συμπίεσης χρησιμοποιούνται για να συμπίεσουν το ψηφιακό video για μια ευρεία χρήση των εφαρμογών όπως:

1. Τηλεοπτική μετάδοση στο Διαδίκτυο.
2. Ανάπτυξη τηλεοπτικών δελτίων ειδήσεων.
3. Τηλεοπτικές συσκευές.
4. Κατανομές video.
5. Αποθήκευση και έκδοση video.

Η απόδοση των σύγχρονων προτύπων συμπίεσης όπως Mpeg-1, 2, είναι αρκετά εντυπωσιακή. Τα ακατέργαστα τηλεοπτικά στοιχεία μπορούν να μειωθούν σε χωρητικότητα χωρίς ιδιαίτερη απώλεια στην αναδημιουργημένη τηλεοπτική ποιότητα.

Αμέσως μετά παρουσιάζονται ορισμένα στοιχεία για το μέγεθος την αξία και την χωρητικότητα των συμπιεσμένων video:

VERY LOW BIT RATE

- Frame size: 144 χ 176 (QCIF)
- Frame rate: 5 - 15 fps
- Target bit rates: 4.8 - 64 Kbps

MEDIUM BIT RATE

- Frame size: 288 χ 352 (GIF) to 576 χ 720
- Frame rate: 25 - 30 fps
- Target bit rates: 200 Kbps - 1.5 Mbps

PROFESSIONAL AND HIGH END APPLICATIONS

- Frame size: 576 χ 720 and larger
- Frame rate: 25 - 30 fps and larger
- Target bit rates: 1.5 - 35 Mbps

3. 6. MPEG

Τα Moving Picture Expert Group (MPEG) είναι τα ευρύτερα αποδεκτά διεθνή πρότυπα για την ψηφιακή τηλεοπτική συμπίεση. Χρησιμοποιούν δύο βασικές τεχνικές.

- MPEG-1, συμβατή τηλεοπτική αναπαραγωγή ήχου και ευρέως χρησιμοποιημένος τύπος για το video και το H/Y.
- MPEG-2, στα DVD και ευρέως χρησιμοποιημένος τύπος στα ηλεκτρονικά είδη ευρείας κατανάλωσης

Στην εργασία αυτή χρησιμοποιούνται μόνο οι δύο παραπάνω τύποι αρχείων video καθώς και .avi διότι η χρησιμοποίηση αρχείων video τύπων mpeg-4 και mpeg-7 απαιτούν την χρησιμοποίηση ειδικών συναρτήσεων του Builder και decoders για την αποσυμπίεση των παραπάνω video.

4. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.

Όπως αναφέρθηκε και πιο πάνω, μια Βάση Δεδομένων είναι η συλλογή των δεδομένων που εμφανίζονται σ' ένα μικρόκοσμο. Για να υλοποιηθεί μία Βάση Δεδομένων πρέπει πρώτα να μοντελοποιηθούν τα δεδομένα αυτά. Η μοντελοποίηση των δεδομένων δεν είναι κάτι που υπάρχει μόνο στις Βάσεις Δεδομένων αλλά είναι μία γενικότερη διαδικασία καταγραφής της πληροφορίας, εξήγησης και πρόβλεψης της συμπεριφοράς, απόκρυψης λεπτομερειών, μίας αφαιρετικής διαδικασίας.

Ένα μοντέλο δεδομένων είναι ένα εργαλείο που επιτρέπει την αφαίρεση από το σύνολο των πραγματικών δεδομένων, με ορισμένους κανόνες και διαδικασίες καταγραφής και συμπεριφοράς τους.

Ένα μοντέλο δεδομένων επιτρέπει την απόκρυψη της φυσικής περιγραφής των δεδομένων. Ένα μοντέλο δεδομένων είναι ο φορμαλισμός που καθορίζει τον τρόπο περιγραφής των δεδομένων (τις δομές) σε ένα αφαιρετικό επίπεδο μαζί μ' ένα σύνολο από περιορισμούς και πράξεις για τον χειρισμό αυτών των δομών.

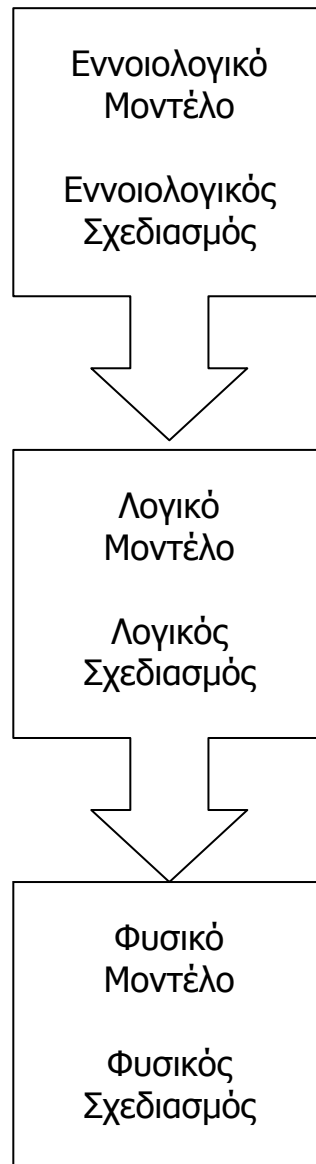
Ένα μοντέλο δεδομένων έχει τρεις βασικές συνιστώσες:

1. **Δομές (Structures):** Είναι τα αντικείμενα του μικρόκοσμου.
2. **Πράξεις (operations):** Είναι οι ενέργειες που γίνονται στις δομές
3. **Περιορισμοί (constraints):** Είναι οι περιορισμοί που μπαίνουν πάνω στις δομές.

Άρα τα μοντέλα δεδομένων είναι μία συλλογή από εργαλεία για την περιγραφή δομών δεδομένων, τη σημασιολογία (semantics) δεδομένων και τους περιορισμούς δεδομένων. Υπάρχουν τρεις διαφορετικές ομάδες που αντιστοιχούν στις αντίστοιχες φάσεις σχεδιασμού:

- **Εννοιολογικά μοντέλα** βασισμένα σε αντικείμενα
- **Λογικά μοντέλα** βασισμένα σε εγγραφές
- **Φυσικά μοντέλα** δεδομένων.

Παρακάτω, Εικόνα 02, φαίνονται σχηματικά οι τρεις αυτές ομάδες.



Εικόνα 02

Ακολουθήσαμε την παραπάνω διαδικασία για την ανάπτυξη της εφαρμογής.

Καταρχήν συλλέξαμε στοιχεία για τις απαιτήσεις της εφαρμογής μας, ρωτώντας, συζητώντας με ομάδες ανθρώπων που η κατάτμηση και περίληψη πληροφοριών από βίντεο του ενδιαφέρει είτε ερασιτεχνικά είτε επαγγελματικά (όπως πχ προπονητές, αθλητές, δημοσιογράφοι κλπ)

Στη συνέχεια αποδώσαμε τις πληροφορίες αυτές στο εννοιολογικό μοντέλο των Οντοτήτων και Συσχετίσεων (Entities and Relationship /E-R)

Για το Λογικό σχεδιασμό χρησιμοποιήσαμε το Σχεσιακό μοντέλο και αναπτύξαμε του πίνακες που επρόκειτο να χρησιμοποιηθούν.

Και για την φυσική καταγραφή των πινάκων επιλέξαμε την βάση δεδομένων της Paradox διότι αποτελεί το standard της Borland DataBase Engine. Φυσικά θα μπορούσαμε να χρησιμοποιήσουμε οποιοδήποτε από τα υπάρχοντα RDBMS για την αποθήκευση των πινάκων μας.

4. 1. ΜΟΝΤΕΛΟ E-R (ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ).

Οι βασικές συνιστώσες του μοντέλου ER είναι δύο: οι οντότητες και οι συσχετίσεις. Οντότητες (entities) είναι συγκεκριμένα αντικείμενα που υπάρχουν και μπορούν να αναπαρασταθούν στη Βάση Δεδομένων και οι συσχετίσεις (relationships) που είναι επίσης ειδικά αντικείμενα συνδέουν δύο ή περισσότερες οντότητες με συγκεκριμένο νόημα.

Δηλαδή το μοντέλο οντοτήτων – συσχετίσεων είναι βασισμένο στην αντίληψη ότι ο κάθε μικρόκοσμος που θα παρασταθεί με τη Βάση Δεδομένων αποτελείται από μία συλλογή από βασικά αντικείμενα τις οντότητες και συσχετίσεις μεταξύ αυτών των αντικειμένων.

Τόσο οι οντότητες όσο και οι συσχετίσεις μπορούν να έχουν γνωρίσματα (attributes), τα οποία είναι ιδιότητες που τις χαρακτηρίζουν. Τα είδη των γνωρισμάτων είναι τα απλά όπου η οντότητα έχει ατομική τιμή για αυτό, τα σύνθετα όπου το γνώρισμα αποτελείται από περισσότερες τιμές και τα πλειότιμα όπου οι πολλαπλές τιμές είναι το χαρακτηριστικό τους.

Βασικό χαρακτηριστικό ενός τύπου είναι το κλειδί (key). Ένα γνώρισμα ή (σύνολο γνωρισμάτων) που χαρακτηρίζει μοναδικά κάθε οντότητα (αντίστοιχα συσχέτιση) μέσα στον τύπο οντοτήτων (συσχετίσεων), δεν μπορούν δηλαδή δύο διαφορετικές οντότητες (συσχετίσεις) μέσα στον ίδιο τύπο να έχουν ίδια τιμή, ονομάζεται υπέρ – κλειδί.

Οι οντότητες που χρησιμοποιήσαμε στην εργασία είναι τα **Video** που θα σχολιαστούν και θα καταταμηθούν σε επι μέρους ακολουθίες και οι ετικέτες που θα χαρακτηρίζουν τις καταταμημένες ακολουθίες βίντεο.

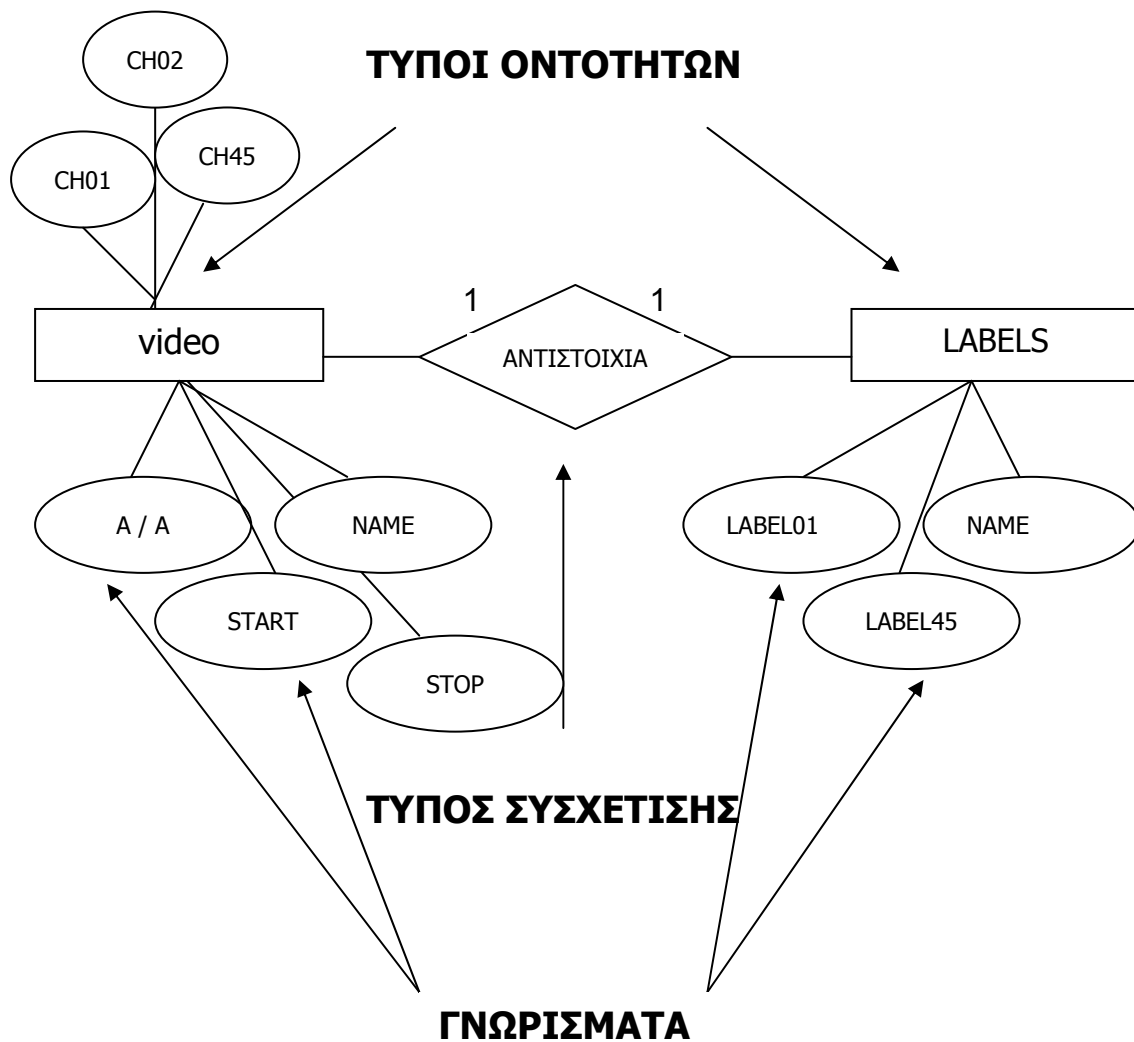
Ο τύπος οντοτήτων “video” περιλαμβάνει ως οντότητες όλα τα video clips και τα χαρακτηριστικά τους καθώς και τον χρόνο έναρξης και λήξης του κάθε video clip.

Ο τύπος οντοτήτων “ετικέτες” περιλαμβάνει τις ονομασίες που θα παίρνουν κάθε φορά οι ετικέτες των χαρακτηριστικών των video clips.

Ορίσαμε ότι 45 χαρακτηρισμοί είναι αρκετοί για ένα video clip για να το περιγράψουν. Ο περιορισμός αυτός είναι υποκειμενικός και προέκυψε μετά από μελέτη και συσκέψεις από άτομα που χρειάζονται μία τέτοια εφαρμογή όπως για παράδειγμα προπονητές και διαπιστώθηκε πως κανένας δεν χρησιμοποίησε παραπάνω από 30 χαρακτηριστικά ταυτόχρονα για μία καταγραφή μίας ενδιαφέρουσας σκηνής video.

Εξάλλου και τα 45 διαφορετικά κριτήρια είναι ήδη αρκετά για κάποιον που θέλει να ομαδοποιήσει σκηνές video και όσο περισσότερους χαρακτηρισμούς κάνει, τόσο περισσότερο χρονοβόρα γίνεται η διαδικασία ομαδοποίησης.

Στη συνέχεια φαίνεται ένα διάγραμμα του μοντέλου οντοτήτων συσχετίσεων ER, Εικόνα 03, όπου τα ορθογώνια αντιπροσωπεύουν τους τύπους οντοτήτων, οι ελλείψεις τις ιδιότητες, οι ρόμβοι τους τύπους συσχετίσεων μεταξύ των τύπων οντοτήτων και οι γραμμές συνδέουν τους τύπους οντοτήτων με τα γνωρίσματα τους και με τους τύπους συσχετίσεων τους.



Εικόνα 03

Σκοπός αυτού του Διαγράμματος είναι να διαχωρίσει τις οντότητες των video clips μεταξύ τους, να καταχωρούμε ταυτόχρονα 45 διαφορετικούς χαρακτηρισμούς αλλά σε επόμενο video clip να έχουμε την δυνατότητα να αλλάζουμε τις ετικέτες των χαρακτηρισμών.

Με αυτόν τον τρόπο οι εφαρμογή μας αποκτά μία δύναμη και ευελιξία ώστε να μπορεί να χρησιμοποιηθεί από διαφορετικούς τύπους χρηστών, που ο καθένας θα μπορεί να δώσει τους δικούς του τίτλους στους χαρακτηρισμούς που θα κάνει στις video ακολουθίες.

Ο λόγος πληθυκότητας του τύπου συσχετίσεων που τα συνδέει είναι 1:1 γιατί για κάθε video ακολουθία αντιστοιχεί ένα σετ ετικετών. Η συμμετοχή των οντοτήτων στις συσχετίσεις είναι ολική μιας και όλα τα video έχουν ετικέτες στα χαρακτηριστικά τους καθώς και όλες οι ετικέτες αντιστοιχούν σε video.

4. 2. ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Τα λογικά μοντέλα βασισμένα σε εγγραφές χρησιμοποιούνται για να περιγράψουν τα δεδομένα στο εννοιολογικό επίπεδο και το σχήμα αρχιτεκτονικής ANSI – SPARK δείχνει τα τρία επίπεδα.

Αντίθετα από τα εννοιολογικά μοντέλα, τα λογικά μοντέλα καθορίζουν τη γενική λογική δομή της Βάσης Δεδομένων και παρέχουν μία υψηλότερου επιπέδου περιγραφή της υλοποίησης της.

Ονομάζονται έτσι επειδή η Βάση Δεδομένων είναι δομημένη σε εγγραφές διάφορων τύπων. Κάθε τύπος εγγραφής καθορίζει έναν σταθερό αριθμό πεδίων (γνωρίσματα - ιδιότητες). Κάθε πεδίο είναι συνήθως σταθερού μήκους (αυτό απλοποιεί την εφαρμογή).

Το σχεσιακό μοντέλο (για τον λογικό σχεδιασμό) χωρίζεται σε τρία μέρη που αφορούν τις δομές (structure) των δεδομένων (οι πίνακες), την ακεραιότητα (integrity) των δεδομένων (δομικοί περιορισμοί), και το χειρισμό των δεδομένων (Πράξεις πάνω στις δομές).

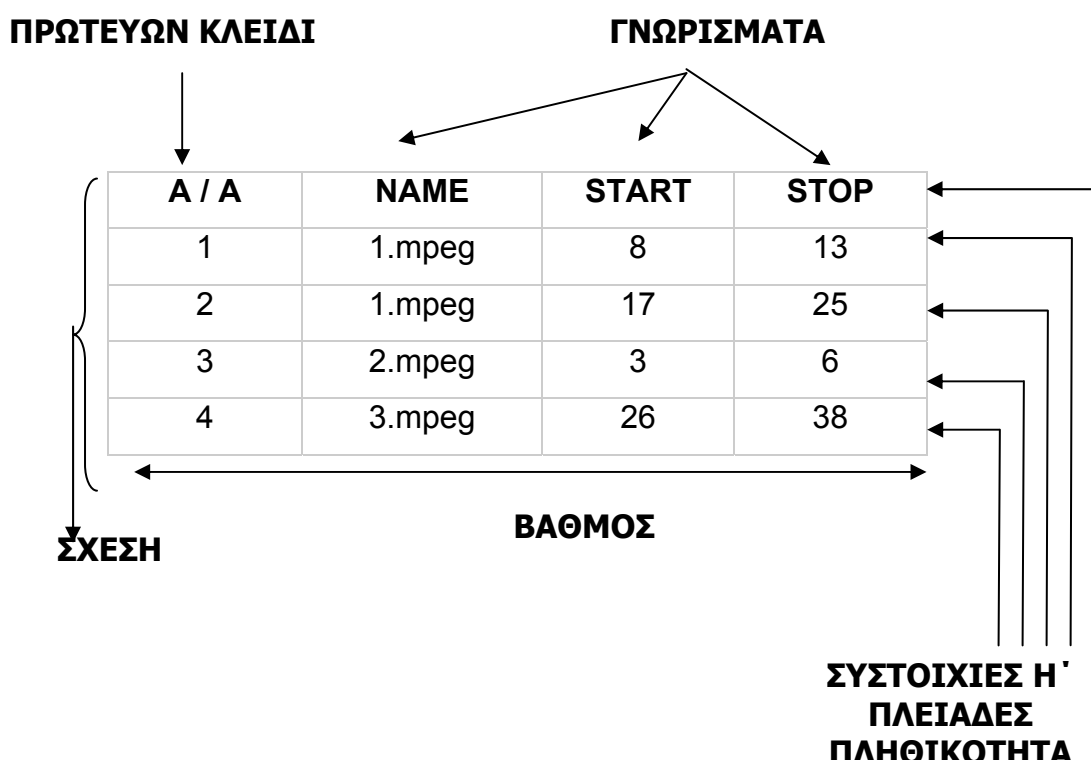
Στο σχεσιακό μοντέλο τα δεδομένα και οι συσχετίσεις αντιπροσωπεύονται από μία συλλογή από κανονικοποιημένους πίνακες (σχέσεις). Κάθε κανονικοποιημένος πίνακας έχει διάφορες στήλες με μοναδικά ονόματα.

Οι σημαντικότεροι από τους όρους που χρησιμοποιούνται με τα “αντικείμενα του σχεσιακού μοντέλου” είναι η σχέση (relation) που αντιστοιχεί σε εκείνο που μέχρι τώρα ονομαζόταν πίνακας, μία συστοιχία ή πλειάδα (tuple) αντιστοιχεί σε μία γραμμή ενός τέτοιου πίνακα και ένα γνώρισμα (attribute) αντιστοιχεί σε μία στήλη. Το πλήθος των συστοιχιών ονομάζεται πληθικότητα (cardinality) και το πλήθος των γνωρισμάτων ονομάζεται βαθμός (degree).

Το πρωτεύον κλειδί (primary key) είναι ένα μοναδικό αναγνωριστικό για τον πίνακα. Δηλαδή, μία στήλη ή συνδυασμός στηλών που έχει την ιδιότητα ότι οποιαδήποτε δεδομένη στιγμή δεν υπάρχουν δύο γραμμές του πίνακα που να περιέχουν την ίδια τιμή σε αυτή τη στήλη ή συνδυασμό στηλών.

Τέλος, ένα πεδίο ορισμού (domain) είναι μία δεξαμενή τιμών από την οποία τα συγκεκριμένα γνωρίσματα των συγκεκριμένων σχέσεων αντλούν τις συγκεκριμένες τιμές τους.

Στην Εικόνα 04, υπάρχει ένας πίνακας όπου φαίνονται όλα τα παραπάνω.



Εικόνα 04

4. 3. ΜΕΤΑΡΟΠΗ ΔΙΑΓΡΑΜΜΑΤΟΣ E-R ΣΤΟ ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Από τη μία πλευρά, το μοντέλο ER διακρίνει τύπους οντοτήτων και τύπους συσχετίσεων (με τα γνωρίσματα τους) ενώ, από την άλλη, το Σχεσιακό μοντέλο υποστηρίζει μία δομή μόνο, τις σχέσεις (που έχουν και αυτές γνωρίσματα). Άρα η διαδικασία μετατροπής ενός διαγράμματος ER σε σχεσιακό σχήμα αφορά στη μετατροπή των τύπων οντοτήτων και συσχετίσεων (και των γνωρισμάτων τους).

Υπάρχει ο γενικός κανόνας ότι για κάθε τύπο οντοτήτων και για κάθε τύπο συσχετίσεων δημιουργείτε μία σχέση που παίρνει το όνομα του αντίστοιχου τύπου. Πέρα από αυτόν το γενικό κανόνα, παρακάτω αναλύονται οι περιπτώσεις για τους τύπους οντοτήτων, τους τύπους συσχετίσεων και τα γνωρίσματα.

➤ ΜΕΤΑΤΡΟΠΗ ΤΥΠΩΝ ΟΝΤΟΤΗΤΩΝ.

Διακρίνουμε τους ισχυρούς από τους αδύναμους τύπους οντοτήτων.

Για κάθε ισχυρό τύπο οντοτήτων δημιουργεί ο χρήστης μία σχέση R με τα ίδια γνωρίσματα, ένα για κάθε απλό γνώρισμα της οντότητας. Αν η οντότητα έχει σύνθετα γνωρίσματα, στη σχέση R υπάρχει ένα γνώρισμα για κάθε απλό γνώρισμα που απαρτίζει το σύνθετο.

Για κάθε αδύνατο τύπο οντοτήτων A που εξαρτάται από τον ισχυρό τύπο οντοτήτων B δημιουργείτε ένα σχήμα σχέσης R με τα γνωρίσματα (α) τα γνωρίσματα του A, και (β) τα γνωρίσματα του πρωτεύοντος κλειδιού του B.

➤ ΜΕΤΑΤΡΟΠΗ ΤΥΠΩΝ ΣΥΣΧΕΤΙΣΕΩΝ.

Διακρίνονται τρεις περιπτώσεις με βάση το λόγο πληθικότητας: M:N συσχετίσεις, 1:1 δυαδικές και 1:N δυαδικές συσχετίσεις. Σχετικά με τις 1:1 και 1:N συσχετίσεις, δεν ξανά σχολιάζονται οι αδύναμες περιπτώσεις, δηλαδή τις συσχετίσεις μεταξύ ενός ισχυρού και ενός αδύναμου τύπου οντοτήτων, αφού ήδη έχουν αναφερθεί παραπάνω.

Για κάθε M:N συσχέτιση R μεταξύ τύπων οντοτήτων που αντιστοιχούν στις σχέσεις S1, S2, ..., Sn δημιουργεί ο χρήστης μία νέα σχέση R με γνωρίσματα: (α) τα γνωρίσματα της R, αν υπάρχουν, και (β) τα γνωρίσματα του πρωτεύοντος κλειδιού κάθε συμμετέχουσας σχέσης Si, τα οποία συμμετέχουν ως ξένα κλειδιά στη σχέση R.

Για κάθε 1-1 δυαδική (μη αδύναμη) συσχέτιση R μεταξύ δύο τύπων οντοτήτων που αντιστοιχούν στις σχέσεις T και S, δεν δημιουργείται νέα σχέση R αλλά επιλέγεται μία εκ των T και S, έστω την T. Στα ήδη υπάρχοντα γνωρίσματα της T προσθέτουμε: (α) τα γνωρίσματα της R, αν υπάρχουν, και (β) το πρωτεύον κλειδί της S, το οποίο συμμετέχει ως ξένο κλειδί στη σχέση T. Για την επιλογή T και S, κριτήριο είναι η ολική συμμετοχή του ενός ή του

άλλου τύπου οντοτήτων στη συσχέτιση: η σχέση που αντιστοιχεί στον τύπο οντοτήτων που συμμετέχει ολικά στη συσχέτιση, είναι αυτή (η σχέση Τα παραπάνω) που επιλέγεται να “φιλοξενήσει” ως ξένο κλειδί το πρωτεύον κλειδί της άλλης (S). Αν δε υπάρχει ολική συμμετοχή από μία πλευρά, τότε προσπαθούμε να μην έχουμε πολλές null τιμές, οπότε επιλέγουμε να προσθέσουμε ξένο κλειδί στη σχέση που αντιστοιχεί στον τύπο οντοτήτων με τη μεγαλύτερη συμμετοχή στη συσχέτιση.

Τέλος, για κάθε 1-N δυαδική (μη αδύναμη) συσχέτιση R μεταξύ δύο τύπων οντοτήτων που αντιστοιχούν στις σχέσεις T και S, δεν δημιουργούμε νέα σχέση R αλλά στα ήδη υπάρχοντα γνωρίσματα της T προστίθενται: (α) τα γνωρίσματα της R, αν υπάρχουν, και (β) το πρωτεύον κλειδί της S, το οποίο συμμετέχει ως ξένο κλειδί στη σχέση T.

Για τις 1:1, 1:N και N:1 δυαδικές (μη αδύναμες) μπορούμε, εναλλακτικά, να ακολουθήσουμε το γενικό κανόνα (δημιουργία νέας σχέσης με ξένα κλειδιά, τα πρωτεύοντα κλειδιά των εμπλεκόμενων σχέσεων), ειδικά όταν δε υπάρχει ολική συμμετοχή από καμία πλευρά. Στην περίπτωση αυτή, κερδίζουμε ως προς την εννοιολογική καθαρότητα και πληρότητα αλλά χάνουμε ως προς την αποδοτικότητα χώρου και τη επίδοση. Συνήθως δε επιλέγεται αυτός ο τρόπος αλλά η ειδική αντιμετώπιση τους, όπως παρουσιάστηκε παραπάνω.

➤ ΓΝΩΡΙΣΜΑΤΑ.

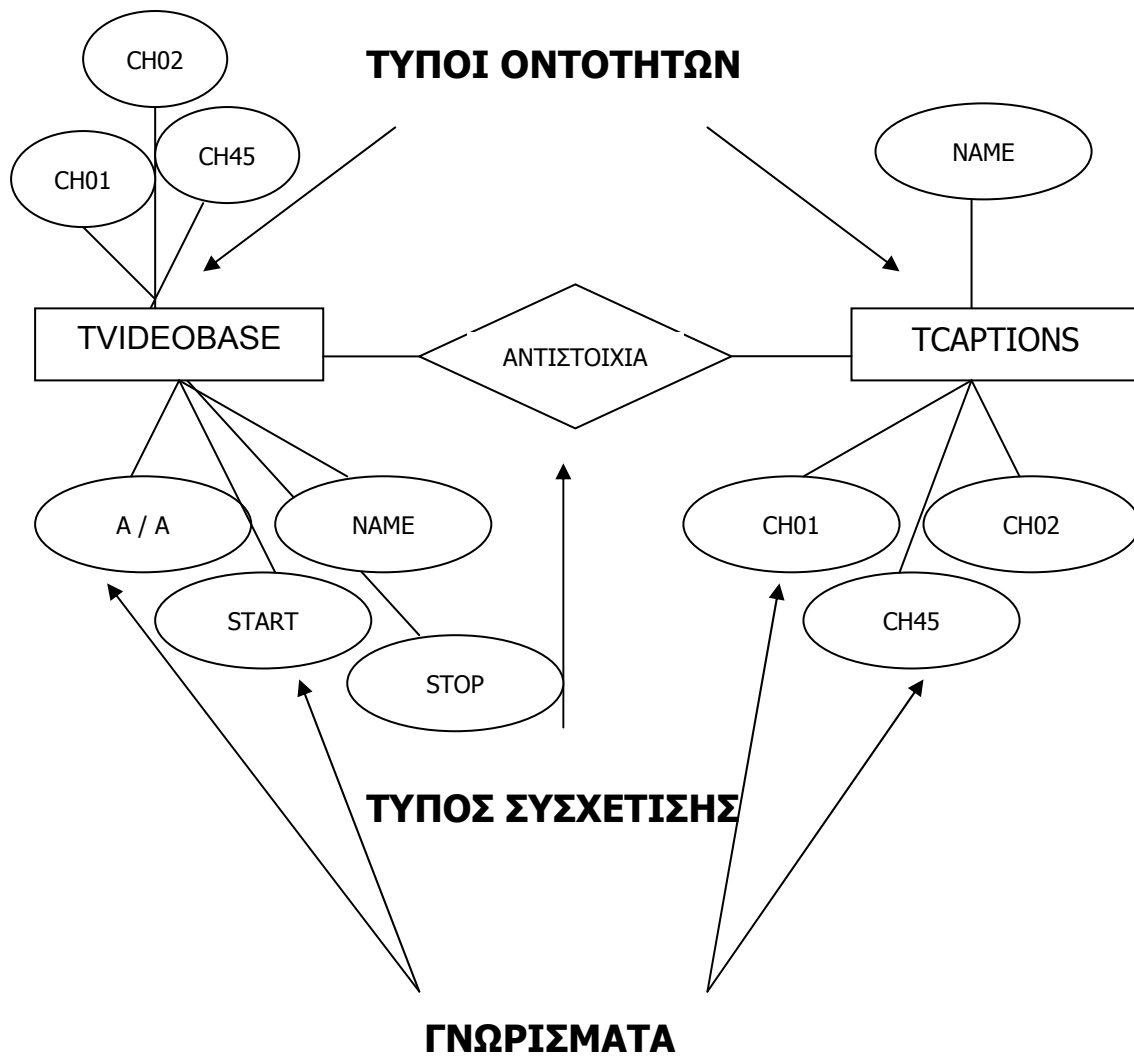
Διακρίνονται μεταξύ απλών γνωρισμάτων όπου ένα απλό γνώρισμα A γίνεται γνώρισμα της αντίστοιχης σχέσης, στα σύνθετα γνωρίσματα όπου ένα σύνθετο γνώρισμα μετατρέπεται σε μία ομάδα γνωρισμάτων για την καινούρια σχέση και στα πλειότιμα γνωρίσματα όπου κατασκευάζεται μία σχέση R με γνωρίσματα το A και τα γνωρίσματα του πρωτεύοντος κλειδιού της σχέσης που παριστάνει τον τύπο οντοτήτων ή συσχετίσεων του οποίου γνώρισμα είναι το A.

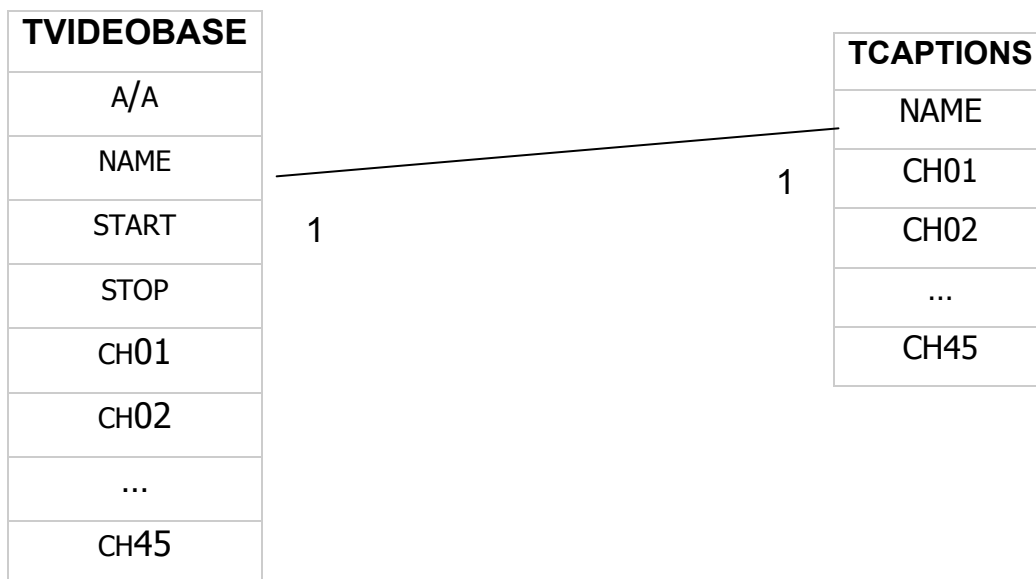
Παρακάτω αναφέρονται επιγραμματικά οι κανόνες και τα βήματα μετατροπής :

1. Αντιστοιχία των οντοτήτων στο σχεσιακό μοντέλο.
2. Αντιστοιχία των ασθενών οντοτήτων στο σχεσιακό μοντέλο.

3. Αντιστοιχία των συσχετίσεων στο σχεσιακό μοντέλο.
 - a) Αντιστοιχία δυαδικής συσχέτισης 1:1.
 - b) Αντιστοιχία δυαδικής συσχέτισης 1:N
 - c) Αντιστοιχία δυαδικής συσχέτισης M:N
4. Αντιστοιχία των αναδρομικών συσχετίσεων στο σχεσιακό μοντέλο
5. Εξάλειψη των πλειώτιμων χαρακτηριστικών.

Στη συνέχεια φαίνεται, Εικόνα 05, σχηματικά η μετατροπή ενός διαγράμματος ER σε σχεσιακό μοντέλο.





Εικόνα 05

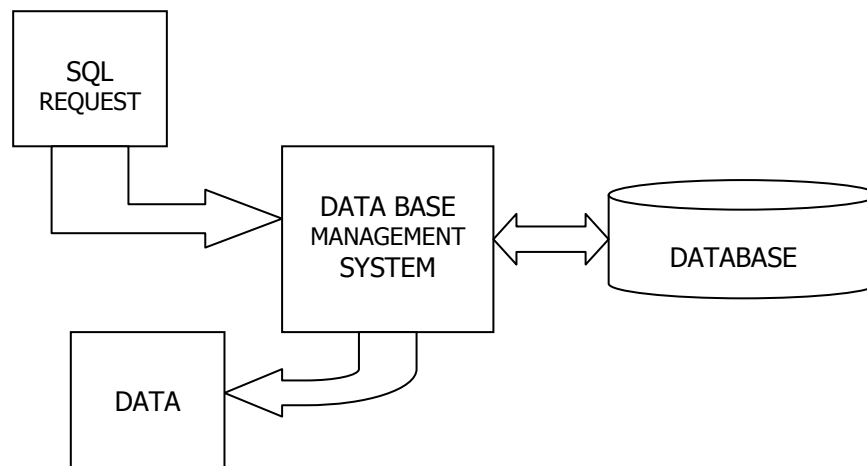
Το πεδίο NAME είναι το path που βρίσκεται το αρχείο του video που πρόκειται να χαρακτηριστεί. Όπως παρατηρείται η ασθενής οντότητα CAPTIONS έγινε ένας πίνακας με αντίστοιχο όνομα, όμως για πρωτεύον κλειδί έχει το πρωτεύον κλειδί της ισχυρής οντότητας video με το οποίο συνδέεται με μία συσχέτιση 1:1

4. 4. Η ΓΛΩΣΣΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ SQL.

Η SQL (Structured Query Language – Δομημένη Γλώσσα Ερωτημάτων) είναι μία σχεσιακή γλώσσα Βάσεων Δεδομένων η οποία σχεδιάστηκε και υλοποιήθηκε από την IBM στα πλαίσια του ερευνητικού της προγράμματος System R που απέβλεπε στην ανάπτυξη ενός πρότυπου σχεσιακού συστήματος Βάσεων Δεδομένων (ΣΣΔΒΔ). Τα δύο πρώτα ΣΣΔΒΔ που διατέθηκαν στο εμπόριο και βασίζονταν στην SQL ήταν το ORACLE και το SQL/DS.

Αν και το όνομα της δηλώνει ότι η SQL χρησιμοποιείται για την υποβολή ερωτημάτων (σχεσιακών πράξεων) προς μία σχεσιακή Βάση Δεδομένων, η SQL είναι κάτι πολύ ευρύτερο. Η SQL έχει σαν συστατικά στοιχεία της μία γλώσσα ορισμού δεδομένων (Data Definition Language, DDL) και όψεων και μία γλώσσα χειρισμού δεδομένων (Data Manipulation

Language, DML) καθώς και εντολών για καθορισμό περιορισμών και εξέλιξης του σχήματος.



Εικόνα 06

Η SQL είναι ανεξάρτητη από την δομή της Βάσης Δεδομένων. Η ροή των πληροφοριών μεταξύ του χρήστη και της Βάσης Δεδομένων είναι όμοια με την παραπάνω εικόνα, Εικόνα 06.

Η SQL μπορεί να ελέγχει όλες τις λειτουργίες για:

1. Ορισμό δεδομένων
2. Ανάκτηση δεδομένων
3. Χειρισμό δεδομένων
4. Έλεγχο πρόσβασης (εξουσιοδότηση)
5. Ακεραιότητα δεδομένων
6. Κοινή χρήση δεδομένων και έλεγχο συναλλαγών

Τέλος, τα κύρια πλεονεκτήματα της SQL είναι τα εξής:

➤ **ΑΝΕΞΑΡΤΗΣΙΑ ΚΑΤΑΣΚΕΥΑΣΤΗ.**

Αλλάζοντας ΣΣΔΒΔ συνήθως δεν χρειάζεται να ξαναγραφτεί ο κώδικας SQL.

➤ **ΥΨΗΛΟΥ ΕΠΙΠΕΔΟΥ ΓΛΩΣΣΙΚΗ ΔΟΜΗ.**

Που σημαίνει πως κάνει εύκολη την της χρήση της παρούσας γλώσσας.

➤ ΤΥΠΟΠΟΙΗΣΗ.

Η τυποποίηση επικυρώθηκε από το ANSI και τον ISO

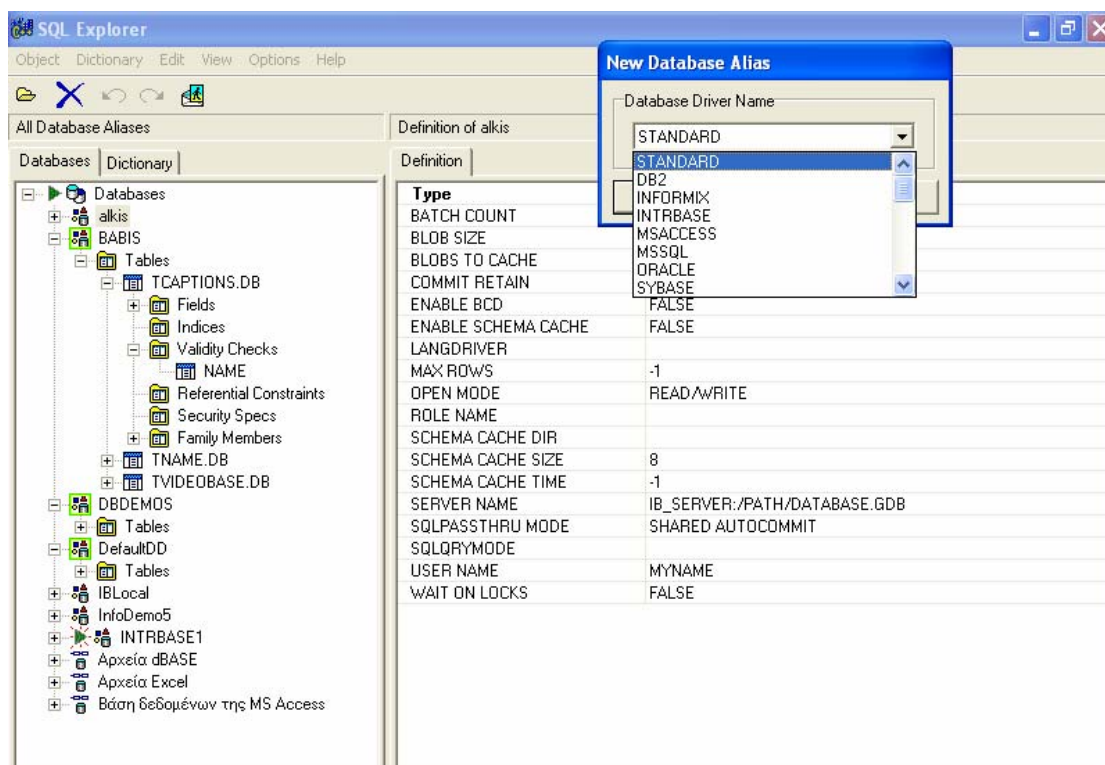
➤ ΔΥΝΑΜΙΚΗ ΔΙΑΧΕΙΡΙΣΗ ΔΕΔΟΜΕΝΩΝ.

Προσφέρει προχωρημένες εντολές επεξεργασίας που επιτρέπουν την ενσωμάτωση της καθώς και τα ερωτήματα πολλαπλών επιπέδων

Ορισμός Δεδομένων

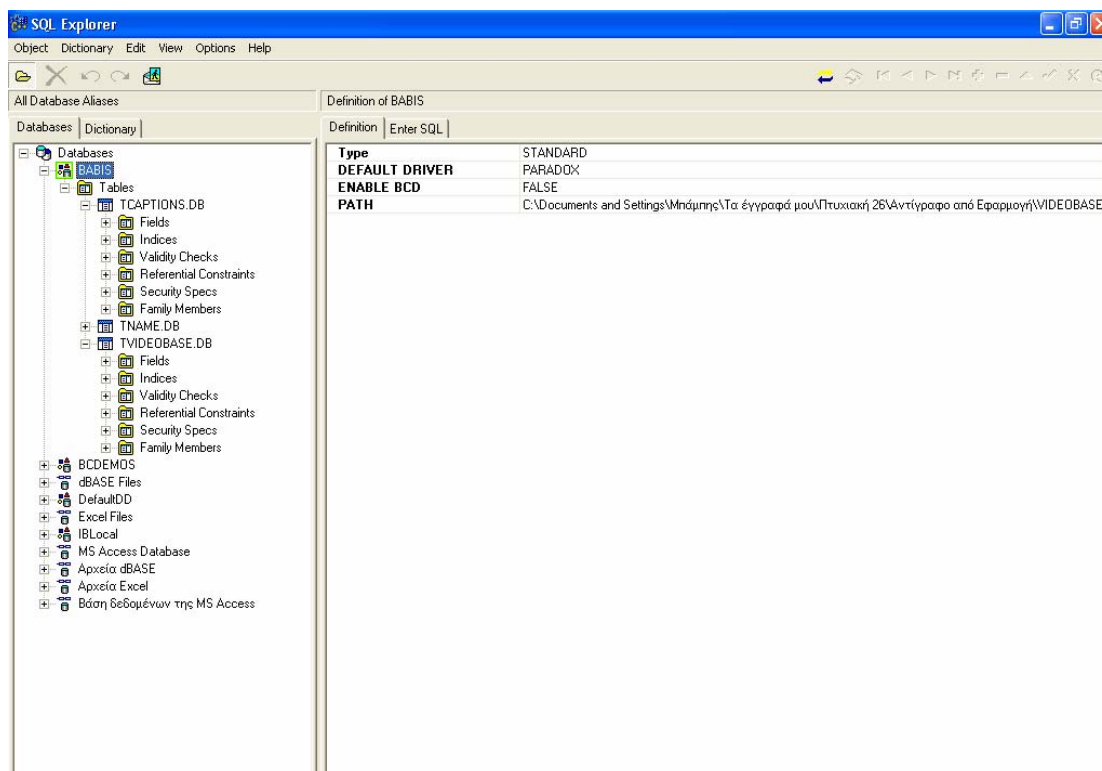
Η βάση που χρησιμοποιήσαμε στην παρούσα εργασία είναι η Paradox. Τα βήματα δημιουργία του πίνακα και των τύπων μεταβλητών είναι τα εξής, ορισμός Δεδομένων, Εισαγωγή – Ανάκτηση δεδομένων τα οποία γίνονται με δυναμικά ερωτήματα μέσα από το πρόγραμμα και Χειρισμός δεδομένων.

Τα βήματα για τον ορισμό Δεδομένων και η κατασκευή της Βάσης Δεδομένων φαίνονται στην Εικόνα 07. Μέσα από την C++ Builder και τον DataBase Explorer, δημιουργούμε μία Βάση Δεδομένων με DataBase Driver, STADARD.



Εικόνα 07

Στη συνέχεια αφού δοθεί το σχετικό όνομα στη βάση, εδώ το όνομα είναι “BABIS”, πηγαίνουμε και βρίσκουμε το PATH όπου και θα αποθηκεύονται τα στοιχεία της Βάσης Δεδομένων. Εικόνα 08.



Εικόνα 08

Ο έλεγχος μπορεί να γίνει αλλά στην εφαρμογή μας δεν χρησιμοποιήθηκε επειδή αναφέρεται σε έναν χρήστη και αφήνεται ο έλεγχος στο λειτουργικό σύστημα. Κατ' επέκταση, δεν έχουμε κοινή χρήση των δεδομένων από άλλους χρήστες.

Για κάθε βάση δεδομένων Paradox δημιουργείται ξεχωριστός κατάλογος αρχείων όπου 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.

Οι πίνακες δημιουργούνται σαν αρχεία μέσα στον κατάλογο της Βάσης Δεδομένων Paradox. Ένας πίνακας Paradox είναι ένα αρχείο με επέκταση ".db". Το όνομα ενός πίνακα Paradox πρέπει να αποτελείται από οκτώ το πολύ χαρακτήρες, ο πρώτος των οποίων πρέπει να είναι αλφαβητικός.

Κάθε πεδίο του πίνακα Paradox 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.

Χειρισμός Δεδομένων

Για τον χειρισμό των δεδομένων που αποθηκεύονταν στους παραπάνω πίνακες, έγινε χρήση δυναμικών ερωτημάτων μέσα από την Hosted γλώσσα προγραμματισμού δηλ. μέσα από την C++ Builder για τα οποία αναφερόμαστε εκτενέστερα και αναλυτικά στον κώδικά του προγράμματος που παρουσιάζεται παρακάτω.

5. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ C++ BUILDER.

5. 1. ΓΕΝΙΚΑ.

Στα εργαστήρια BELL το 1972, ενώ σχεδιάζοταν τον λειτουργικό σύστημα UNIX δημιούργησαν παράλληλα και την γλώσσα προγραμματισμού C. Το σημαντικότερο στοιχείο είναι ότι η C σχεδιάστηκε για να γίνει ένα εργαλείο προγραμματισμού. Ο πρωταρχικός σκοπός ήταν να γίνει μία χρήσιμη γλώσσα.

Οι περισσότερες γλώσσες που σχεδιάζονται βασικά έχουν αυτό ως πρωταρχικό σκοπό. Να γίνουν χρήσιμες γλώσσες. Η C σε αντίθεση με πολλές άλλες γλώσσες συνδυάζει πολλά στοιχεία μαζί όπως να δώσει μία γερή βάση για την διδασκαλία των αρχών του προγραμματισμού αλλά να είναι παράλληλα κατανοητή και σε αυτούς που θα την χειρίζονται. Η C είναι ένα δυναμικό εργαλείο προγραμματισμού και παράλληλα παραμένει φιλική με τον προγραμματιστή.

5. 2. ΟΠΤΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.

Ο οπτικός προγραμματισμός με C++ Builder είναι ένα εργαλείο προγραμματισμού, το οποίο μοιάζει πολύ στη Visual Basic αλλά και στη Delphi.

Η ανάπτυξη εφαρμογών γίνεται μέσω του ολοκληρωμένου περιβάλλοντος (IDE) που διαθέτει. Βασίζεται στη αντικειμενοστραφή γλώσσα προγραμματισμού C++. Ο C++ Builder είναι ευέλικτος και έχει πολλά πλεονεκτήματα σε όλα τα στάδια ανάπτυξης εφαρμογών, καθώς και στην εγκατάσταση και εκτέλεση των εφαρμογών από τους χρήστες, αφού η εφαρμογή μπορεί να αποτελείτε από ένα αυτόνομο εκτελέσιμο αρχείο (.exe), χωρίς να βασίζεται στην ύπαρξη άλλων αρχείων. Με τον C++ Builder μπορούμε να ξαναχρησιμοποιήσουμε τα στοιχεία και τον κώδικα καθώς και να

ενσωματώσουμε βιβλιοθήκες. Έτσι ο προγραμματιστής συντομεύει κατά πολύ τη δουλειά του και γίνεται πιο εύχρηστη.

Γενικά, ο οπτικός προγραμματισμός με C++ Builder βασίζεται κυρίως σε φόρμες. Οι φόρμες φιλοξενούν μηχανισμούς – αντικείμενα. Οι πληροφορίες για τα στοιχεία μίας φόρμας καθώς και των μηχανισμών που περιέχει, αποθηκεύονται σε ένα αρχείο φόρμας (.dfm) και σε ένα αρχείο μονάδας (.cpp) με το ίδιο όνομα.

Ο C++ Builder περιλαμβάνει δικούς του μηχανισμούς που αποκαλούνται οπτικά συστατικά (Visual Components) και περιέχονται στη βιβλιοθήκη VCL (Visual Components Library).

Κάθε συστατικό – αντικείμενο του C++ Builder έχει τις ιδιότητες του, στις οποίες βασίζεται η συμπεριφορά του, όταν χρησιμοποιείται σε ένα έργο ή μία φόρμα.

Στην δική μας εφαρμογή τους πίνακες που δημιουργήσαμε στην InterBase, θέλουμε να τους διαχειριστούμε με έναν εύκολο και αποδοτικό τρόπο. Για τον λόγο αυτό χρησιμοποιήσαμε το οπτικό περιβάλλον που προσφέρει ο C++ Builder.

5. 3. ΣΥΝΑΡΤΗΣΕΙΣ - ΑΝΤΙΚΕΙΜΕΝΑ.

Στη παρούσα εργασία έχουν χρησιμοποιηθεί διάφορες πάρα πολύ σημαντικές συναρτήσεις και αντικείμενα. Αναφέρονται ορισμένες από αυτές:

- Open, Save, Save As, Close, Exit είναι πολύ συνηθισμένες και η λειτουργία τους είναι φανερή.
- Έχει χρησιμοποιηθεί ένας Timer στην φόρμα όπου γίνεται η αρχική προβολή του video ο οποίος καθορίζει την έναρξη και τον τερματισμό του χρόνου και καταγράφεται σε Edit κατά την διάρκεια της προβολής του video.
- CheckBoxes τα οποία το caption παραμένει κενό απενεργοποιούνται.
- Έχει χρησιμοποιηθεί ένα animate στο οποίο προβάλλεται το video.
- Επίσης πολύ σημαντικό, ένας media player καθορίζει την προβολή των video είτε κατά τον διαχωρισμό τους είτε κατά την προβολή των “κομματιασμένων” video.

- Ένας DBNavigator ο οποίος δίνει τη δυνατότητα στο χρήστη να καθορίζει και οργανώνει ο ίδιος τη σειρά με την οποία θα προβληθούν τα video
- Ένα ComboBox το οποίο πληροφορεί το χρήστη από πού “πηγάζει” το αρχείο video, δηλαδή δείχνει το path.

Υπάρχουν βέβαια και άλλα σημαντικά στοιχεία τα οποία τα παρουσιασθούν στη συνέχεια αναλυτικότερα. Στη συνέχεια επίσης, θα δείξουμε πως ενώνονται όλα τα παραπάνω, δηλαδή οι Βάσεις Δεδομένων με τον οπτικό προγραμματισμό.

6. ΣΥΝΔΕΣΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ.

Το C++ Builder και η βιβλιοθήκη VCL σχεδιάστηκαν ειδικά ώστε να διευκολύνουν την δημιουργία κομψών και λειτουργικών συστημάτων Βάσεων Δεδομένων οποιασδήποτε κλίμακας τα οποία και θα μπορούν να συντηρούνται εύκολα. Παρακάτω, παρουσιάζονται τα διάφορα συστατικά που παρέχει η Borland για την ανάπτυξη εφαρμογών Βάσεων Δεδομένων:

1. BDE (Borland Database Engine).

Τα συστατικά BDE είναι το αρχικό σύνολο συστατικών που παρέχονταν για προϊόντα της Borland. Χρησιμοποιούνται για την σύνδεση εφαρμογών με το Borland Database Engine και τα προγράμματα οδήγησης Βάσεων Δεδομένων (Database Drivers) που διαθέτει. Το BDE παρέχει ένα κοινό επίπεδο αφαίρεσης για μία ποικιλία συστημάτων διαχείρισης Βάσεων Δεδομένων (DBMS).

Απ' όλους τους τρόπους που υπάρχουν για την προσπέλαση δεδομένων μέσω του C++Builder, το Borland Database Engine (BDE) είναι ένας από τους πιο κοινά χρησιμοποιούμενους. Επιπρόσθετα, η αρχιτεκτονική που χρησιμοποιείται για τα συστατικά του συνεχίζει να είναι η βάση για σχεδόν κάθε σύνολο συστατικών.

Το BDE είναι ένα ανεξάρτητο από τη γλώσσα επίπεδο ακριβώς κάτω από την βιβλιοθήκη VCL. Η Borland το δημιούργησε με στόχο να επιτρέψει την απρόσκοπτη συνεργασία διαφόρων μορφών Βάσεων Δεδομένων με τα εργαλεία προγραμματισμού της.

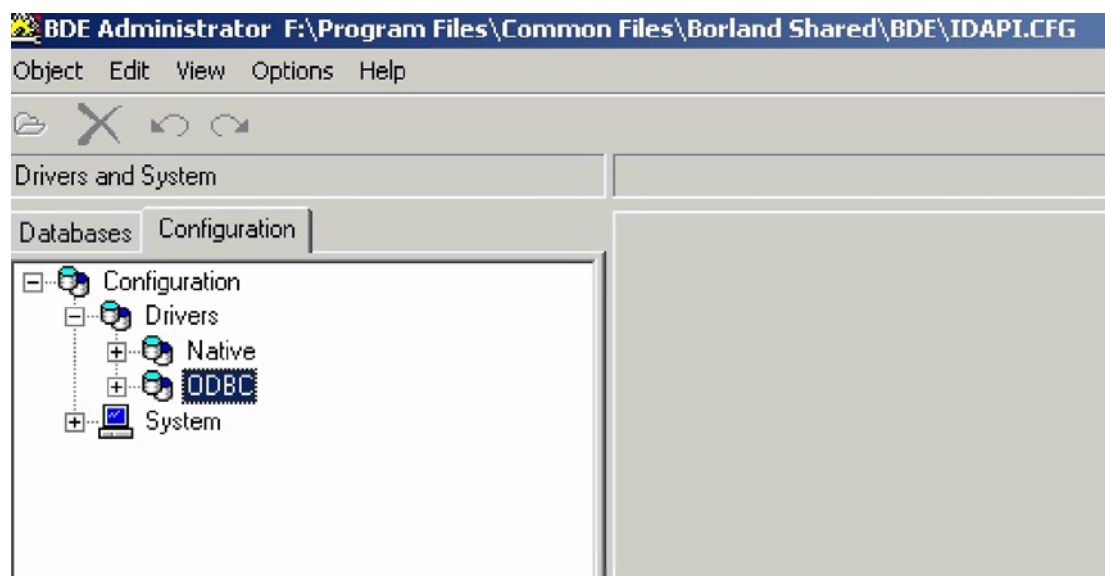
Το BDE παρέχει τρομακτική ισχύ και αξία για κάθε δημιουργό εφαρμογών και χρήστη. Αν και σε ορισμένες περιπτώσεις η εγκατάσταση και αναβάθμιση του BDE θεωρείται αχρείαστος φόρτος, οι δυνατότητες που παρέχει είναι συχνά ζωτικής σημασίας για την επιτυχία των εφαρμογών.

Στις υποστηριζόμενες μορφές περιλαμβάνονται οι αυτόνομες Βάσεις Δεδομένων, οι Βάσεις Δεδομένων Client/Server και η Open Database Connetivity (ODBC).

Δημιουργία user interface με τον Borland Builder.

Ο BDE Administrator της Borland είναι ένα πλήρως λειτουργικό εργαλείο το οποίο μας δίνει την δυνατότητα παρουσίασης ή τροποποίησης πληροφοριών που αφορούν τη δομή μίας βάσης δεδομένων καθώς και τις πληροφορίες που βρίσκονται αποθηκευμένες στους πίνακες της.

Ο BDE Administrator έχει δύο τμήματα. Το αριστερό τμήμα είναι μία γραφική απεικόνιση των ονομάτων ALIAS που αντιστοιχούν οι Βάσεις Δεδομένων για τις οποίες είναι ενήμερη η Μηχανή Βάσεων Δεδομένων της Borland. Στο δεξιό τμήμα παρουσιάζονται οι πληροφορίες που αφορούν το όνομα ALIAS που έχουμε επιλέξει στο αριστερό μέρος. Εικόνα 09.



Εικόνα 09

Εξαρτήματα πρόσβασης βάσεων δεδομένων

Η C++ Builder περιέχει μη οπτικά εξαρτήματα (non - visual) εξαρτήματα πρόσβασης δεδομένων μιας βάσης (data - access components) τα οποία συνδέουν τα αντικείμενα της εφαρμογής με τη βάση. Τα εξαρτήματα τοποθετούνται στις φόρμες της εφαρμογής (Data Modules). Περιέχει ακόμη τα οπτικά εξαρτήματα (visual components) τα οποία περιέχουν τα δεδομένα (data - aware) της βάσης. Τα αντικείμενα αυτά είναι τα οπτικά σημεία ελέγχου (visual controls) και αποτελούν τα σημεία της επικοινωνίας του χρήστη της

εφαρμογής με τη Βάση Δεδομένων. Είναι μια φόρμα επικοινωνίας ανάμεσα στο χρήστη και την εφαρμογή (user interface). Κάθε σημείο ελέγχου είναι συνδεδεμένο με ένα ή περισσότερα πεδία.

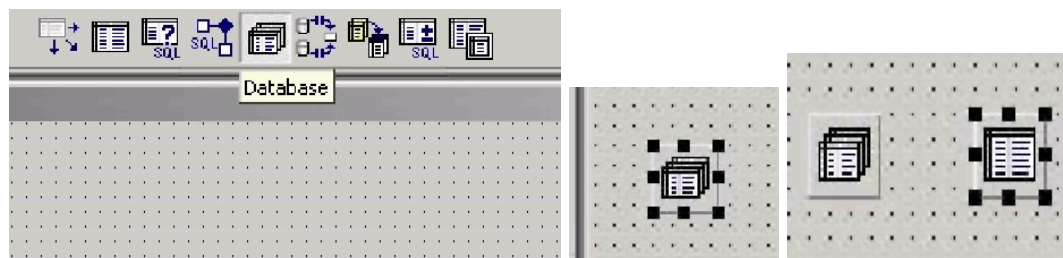
Δημιουργία πρόσβασης βάσεων δεδομένων

Ανοίγουμε τον C++ Builder. Στην φόρμα που εμφανίζετε τοποθετούμε και προγραμματίζουμε τα Εξαρτήματα πρόσβασης Βάσεων Δεδομένων. Στην καρτέλα data Access περιέχονται μη οπτικά εξαρτήματα πρόσβασης δεδομένων μιας βάσης

Το εξάρτημα Tdatabase που αντιπροσωπεύει μια σύνδεση με τη βάση δεδομένων έχει τις παρακάτω ιδιότητες:

- **Alias Name:** το όνομα του ALIAS του BDE με το οποίο είναι συνδεδεμένο το εξάρτημα.
- **DatabaseName:** Το όνομα της βάσεως με την οποία θα γίνει η σύνδεση.
- **Connected:** True, το εξάρτημα συνδέεται κατά τη έναρξη της εφαρμογής με τη βάση που ορίζεται στην ιδιότητα AliasName. Όταν γίνεται η σύνδεση ζητείται το User Name και το Password του Server με τον οποίο θα γίνει η σύνδεση.

Στη συνέχεια τοποθετείται το εξάρτημα TTable που αντιπροσωπεύει ένα σεντ πληροφοριών (Dataset) το οποίο αποτελείται από όλες τις εγγραφές και τα πεδία που βρίσκονται αποθηκευμένα σε ένα πίνακα. Τα παραπάνω φαίνονται στην Εικόνα 10.



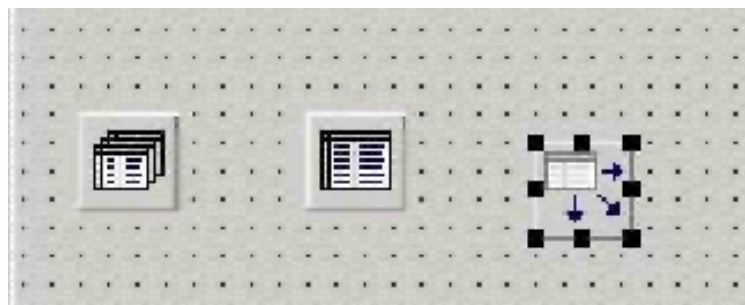
Εικόνα 10

Το εξάρτημα TTable στη φόρμα Data Modules έχει τις παρακάτω ιδιότητες:

- **DatabaseName:** Το όνομα της βάσης με την οποία θα γίνει η σύνδεση και από την οποία προέρχεται ο πίνακας που αντιπροσωπεύει ο TTable.
- **TableName:** Το όνομα του πίνακα τον οποίο αντιπροσωπεύει ο TTable και από τον οποίο αντλεί τα δεδομένα.
- **Active:** True, τα δεδομένα του πίνακα της βάσης είναι διαθέσιμα προς τα εξαρτήματα που λαμβάνουν δεδομένα από τον πίνακα.
- **Name:** Το όνομα που ορίζεται το εξάρτημα και είναι γνωστό στα υπόλοιπα εξαρτήματα.

Τέλος τοποθετείται το εξάρτημα TDataSource που λειτουργεί ως αγωγός μεταξύ των εξαρτημάτων δεδομένων και οπτικών σημείων ελέγχου, Εικόνα 11, και το οποίο έχει τις παρακάτω ιδιότητες:

- **DataSet:** Το όνομα του πίνακα ή του ερωτήματος από την οποία προέρχονται τα δεδομένα.
- **Name:** Το όνομα που ορίζεται το εξάρτημα και είναι γνωστό στα υπόλοιπα εξαρτήματα.



Εικόνα 11

2. ODBC (Open Database Connetivity).

Το Open Database Connetivity (ODBC) είναι μία πρωτοβουλία της Microsoft η οποία κατέληξε να γίνει ένα καθολικά αποδεκτό πρότυπο. Το ODBC θεωρείτε σαν έναν μηχανισμό μετάφρασης για τα συστήματα προγραμματισμού εφαρμογών (API) διαφόρων DBMS. Οποιοσδήποτε κατασκευαστής μπορεί να παρέχει ένα πρόγραμμα οδήγησης (driver) για την μορφή του συστήματος DBMS που κατασκευάζει έτσι ώστε να μπορεί να χρησιμοποιηθεί σαν μία πηγή δεδομένων ODBC (ODBC data source). Ουσιαστικά, σήμερα όλα τα συστήματα DBMS έχουν ένα πρόγραμμα οδήγησης ODBC.

Το πρότυπο ODBC επιτυγχάνει την ανεξαρτησία από το υποκείμενο DBMS παρεμβάλλοντας δύο επίπεδα πριν από το επίπεδο που αντιστοιχεί στο λογισμικό του DBMS. Το πρώτο επίπεδο παρέχει ένα standar API για το οποίο οι εφαρμογές μπορούν να είναι σίγουρες ότι θα είναι συνεπείς. Αυτό το client πρόγραμμα ODBC. Το δεύτερο επίπεδο είναι το πρόγραμμα οδήγησης που εκτελεί την μετάφραση.

Να σημειωθεί ότι το ODBC αφήνει ανοικτές τις δυνατότητες υποστήριξης πολλών μορφών Βάσεων Δεδομένων όπως π.χ. η M. Access επειδή είναι πρότυπο της Microsoft και κατά συνέπεια έχει δημιουργηθεί μία μεγάλη ποικιλία προγραμμάτων οδήγησης για αυτό.

Τα πλεονεκτήματα και τα μειονεκτήματα του ODBC είναι τα εξής:

ΠΛΕΟΝΕΚΤΗΜΑΤΑ.

- Το ODBC είναι ένα ευρέως αποδεκτό πρότυπο.
- Μπορεί να συνεργάζεται με το BDE και δεν απαιτεί αλλαγές στο πρόγραμμα κατά την εναλλαγή μεταξύ BDE ή προγραμμάτων οδήγησης ODBC, γεγονός το οποίο παρέχει σημαντική ευελιξία.
- Επιτρέπει στην ίδια εφαρμογή να δουλεύει ουσιαστικά με οποιαδήποτε Βάση Δεδομένων, ανεξάρτητα από την μορφή της (υπό τον όρο ότι υπάρχει ένα πρόγραμμα οδήγησης ODBC για αυτήν.)

- Μπορεί να μετατρέψει μία εφαρμογή η οποία κατά τα' άλλα θεωρείται "μίας βαθμίδας" σε εφαρμογή client/servers.
- Δεν υπάρχει ανάγκη αγοράς και εκμάθησης νέων συστατικών και βιβλιοθηκών.

ΜΕΙΟΝΕΚΤΗΜΑΤΑ.

- Όταν χρησιμοποιούνται σε συνδυασμό τα BDE και ODBC, αυξάνονται σημαντικά οι απαιτήσεις σε χώρο δίσκου και μνήμης, πράγμα το οποίο μπορεί να μειώσει την απόδοση. Επιπλέον, σ' αυτή την περίπτωση υπάρχουν τρία επίπεδα ανάμεσα στο πρόγραμμα και στο DBMS.
- Ορισμένα προγράμματα οδήγησης ODBC δεν είναι όσο εύρωστα θα έπρεπε. Είναι πολύ σημαντικό να ελέγχετε το πρόγραμμα οδήγησης που υπάρχει περίπτωση να χρησιμοποιηθεί πριν δεσμευθεί ο χρήστης σ' αυτή την αρχιτεκτονική.
- Η αναβάθμιση δεν είναι εύκολη υπόθεση. Η εφαρμογή, η βιβλιοθήκη VCL, το BDE, οι βασιζόμενες στο ODBC εφαρμογές, το πρόγραμμα οδήγησης ODBC και το DBMS πρέπει να ενημερώνονται ανεξάρτητα.

3. Interbase Express (IBExpress).

Συστατικά παρόμοια με το σύνολο συστατικών BDE, τα οποία είναι ειδικά βελτιστοποιημένα για το ανοικτού κώδικα συστήματα διαχείρισης Βάσεων Δεδομένων της Borland ανήκουν στην Interbase.

Τα συστατικά δηλαδή IBExpress αποτελούν ουσιαστικά μία επέκταση των συστατικών BDE και υποστηρίζουν μία ειδική, βελτιστοποιημένη σύνδεση με την Interbase, ένα σύστημα διαχείρισης Βάσεων Δεδομένων.

Στην Interbase, οι γεννήτριες (generators), μπορούν να χρησιμοποιούνται για την δημιουργία κανόνων στην πλευρά του server. Χρησιμοποιούνται σε πολλές λειτουργίες, από την απλή υλοποίηση ενός πεδίου με αυτόματη αύξηση τιμής έως πολύπλοκες διεργασίες SQL οι οποίες εμπλέκουν πολλαπλούς πίνακες και χιλιάδες εγγραφές.. Επειδή βρίσκονται στη Βάση Δεδομένων, συνήθως μπορούν να εκτελούν μία λειτουργία στη Βάση Δεδομένων πολύ γρηγορότερα από έναν αντίστοιχο κανόνα, ο οποίος βρίσκεται στη πλευρά του client.

GENERATORS.

Οι Γεννήτριες (generators) μπορούν να θεωρηθούν σαν γενικές ακέραιες μεταβλητές για την Βάση Δεδομένων. Δεν συνδέονται με οποιοδήποτε πίνακα ή πεδίο, αλλά παράγουν απλώς ακολουθίες αριθμών μέσω της συνάρτησης GEN_ID(). Αν και από μόνες τους οι γεννήτριες δεν χρησιμοποιούνται συχνά, χρησιμοποιούνται εκτενώς για την εκχώρηση τιμών σε μοναδικά κλειδιά, τόσο σε triggers όσο και σε αποθηκευμένες διαδικασίες (spprocs).

Τα παραπάνω χρησιμοποιήθηκαν στη εφαρμογή μας για τις τιμές του μοναδικού κλειδιού A / A του πίνακα TVIDEOBASE.

4. ADO Express - dbGo.

Ένα σύνολο συστατικών το οποίο αρχικά ονομαζόταν ADO Express σε παλαιότερες εκδόσεις, πριν από το C++ Builder 6.0. Αυτά τα συστατικά παρέχουν επικοινωνία με τα ActiveX Data Objects της Microsoft (αναφέρονται συνήθως σαν παροχείς δεδομένων [data providers]). Το dbGo δεν απαιτεί το DBE, αλλά δεν μπορεί να τρέξει σε συστήματα Linux και για το λόγο αυτό δεν θα πρέπει να χρησιμοποιείται σε εφαρμογές CLX.

Ο όρος ADO είναι ακρωνύμιο του όρου ActiveX Database Objects. Η Microsoft αντικατέστησε το πρότυπο ODBC (Open Database Connectivity) και προγενέστερες τεχνολογίες προσπέλασης δεδομένων, όπως οι DAO, κι RDO, με το ADO.

Το ADO είναι ένα βασιζόμενο στο μοντέλο COM (Component Object Model) σύστημα επικοινωνίας εφαρμογών (Application Programming Interface, API) το οποίο αντικαθιστά το βασιζόμενο στη γλώσσα C API του ODBC και διευκολύνει τη χρήση αντικειμενοστραφών τεχνικών με Βάσεις Δεδομένων.

Τα συστατικά ADO κρύβουν μεγάλο μέρος της πολυπλοκότητας που συνεπάγεται η ενασχόληση με τις βασιζόμενες στο μοντέλο COM απόψεις του ADO πίσω από μία διεπιφάνεια παρόμοια με αυτήν των κανονικών συστατικών Βάσεων Δεδομένων της βιβλιοθήκης VCL της Borland.

Επιπρόσθετα, επιτρέπουν τη χρησιμοποίηση δεδομένων από το ADO σε κανονικούς, ενήμερους ως προς τα δεδομένα μηχανισμούς της βιβλιοθήκης VCL, όπως π.χ. πλέγματα εμφάνισης δεδομένων, πεδία επεξεργασίας κειμένου και γραφήματα. Αυτό είναι εφικτό επειδή τα συστατικά ADO είναι απόγονοι της κλάσης TDataSet, και επειδή το συστατικό TDataSource μπορεί να δουλεύει οποιοδήποτε συστατικό – απόγονο του TDataSet.

ΠΛΕΟΝΕΚΤΗΜΑΤΑ.

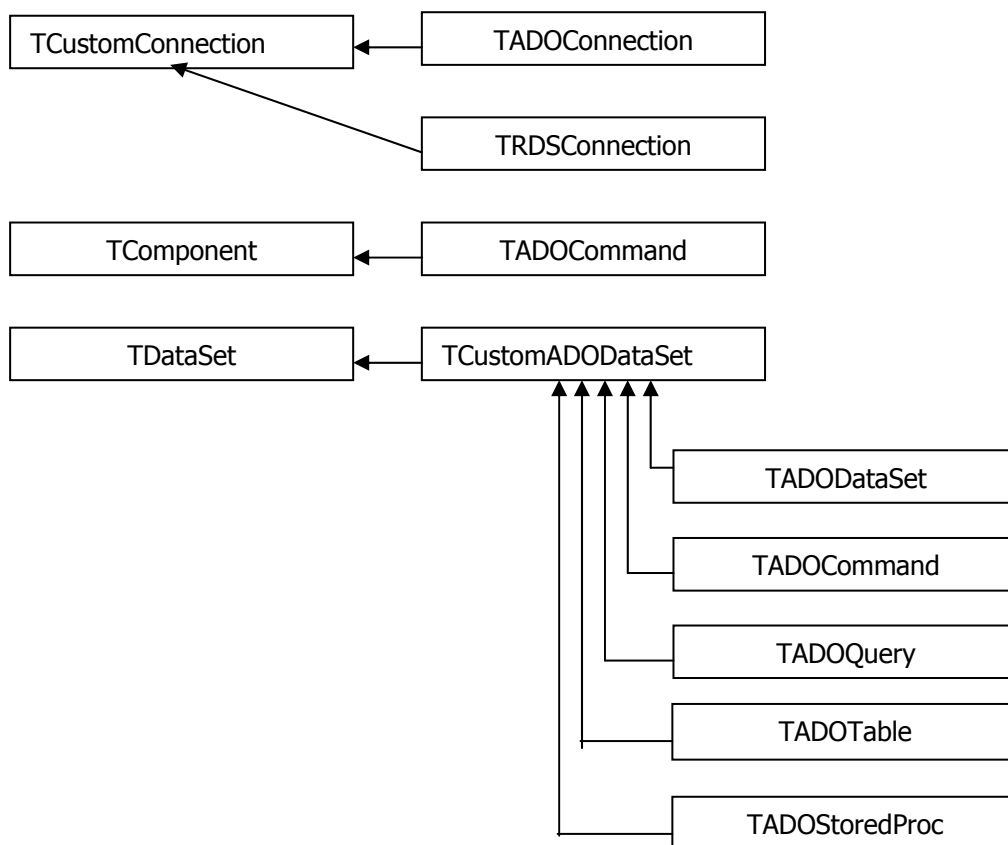
- Μεγάλο μέρος του λογισμικού που απαιτείται για την υποστήριξη τους παρέχεται μαζί με το λειτουργικό σύστημα, οπότε δε χρειάζεται να κάνετε κάτι ειδικό για να το εγκαταστήσετε.
- Λόγω του ότι η εξέλιξη των προγραμμάτων οδήγησης για το BDE είναι βραδύτερη απ' ότι στις προηγούμενες εκδόσεις, το ADO μπορεί να είναι το μοναδικό σας εισιτήριο για την προσπέλαση ασυνήθιστων ή προηγμένων τεχνολογιών δεδομένων, όπως η XML.
- Τα συστατικά ADO μπορούν να διευκολύνουν την μετάβαση από εργαλεία της Microsoft, όπως η Visual C++ στο C++ Builder.
- Τα συστατικά TADQuery είναι πάντα επεξεργάσιμα, χωρίς να καταφεύγουν σε αποθηκευμένες ενημερώσεις, UpdateSQL, ή στις πολύπλοκες συνθήκες που καθιστούν επιτυχή μία αίτηση για ανάκτηση δεδομένων.
- Τα συστατικά ADO επιτρέπουν την ασύγχρονη εκτέλεση κώδικα SQL και την παρακολούθηση της προόδου εκτέλεσης των εντολών μέσω χειριστών συμβάντων. Αυτό μπορεί επίσης να χρησιμοποιηθεί για να παρέχει τον εξαιρετικά χρήσιμο μηχανισμό εμφάνισης προόδου, ο οποίος δείχνει το ποσοστό ολοκλήρωσης της εκτέλεσης ενός ερωτήματος ή μίας εντολής.
- Ανόμοια με τα συστατικά BDE, είναι σχετικά ασφαλές να τερματίζεται ένα πρόγραμμα Βάσεων Δεδομένων χρησιμοποιώντας την Program Reset. Ωστόσο όταν δεν είναι ασφαλές, συνήθως ο τερματισμός του προγράμματος προκαλεί την κατάρρευση του συστήματος.
- Τέλος, αν και θα πρέπει να κρύβονται όσο το δυνατό περισσότερο τα χαρακτηριστικά των συστατικών ADO από τα προγράμματα, η εξοικείωση

με τις βασικές έννοιες και αρχές του ADO μπορεί να αποδειχτεί χρήσιμη μελλοντικά, π.χ. σε έργα των οποίων η διαχείριση υπαγορεύει την χρήση του ADO.

ΜΕΙΟΝΕΚΤΗΜΑΤΑ.

- Απαιτεί ένα συγκεκριμένο αριθμητικό σύνδεσης το οποίο θα κατονομάζει ένα παροχέα, έναν server και μία Βάση Δεδομένων. Αυτό καθιστά πολύ πιο εύκολη την αλλαγή της εφαρμογής ώστε να υποστηρίζει άλλες Βάσεις Δεδομένων ή συστήματα DBMS χωρίς να απαιτούνται αλλαγές στον κώδικα.
- Ανόμοια με το BDE, η SQL στα συστατικά ADO δεν μπορεί να χρησιμοποιείται για ετερογενείς ενώσεις πινάκων από διαφορετικές Βάσεις Δεδομένων του ίδιου ή διαφορετικών DBMS.

Παρακάτω εμφανίζεται η ιεραρχία κλάσεων για τα συστατικά ADO, Εικόνα 12.



Εικόνα 12

5. dbExpress.

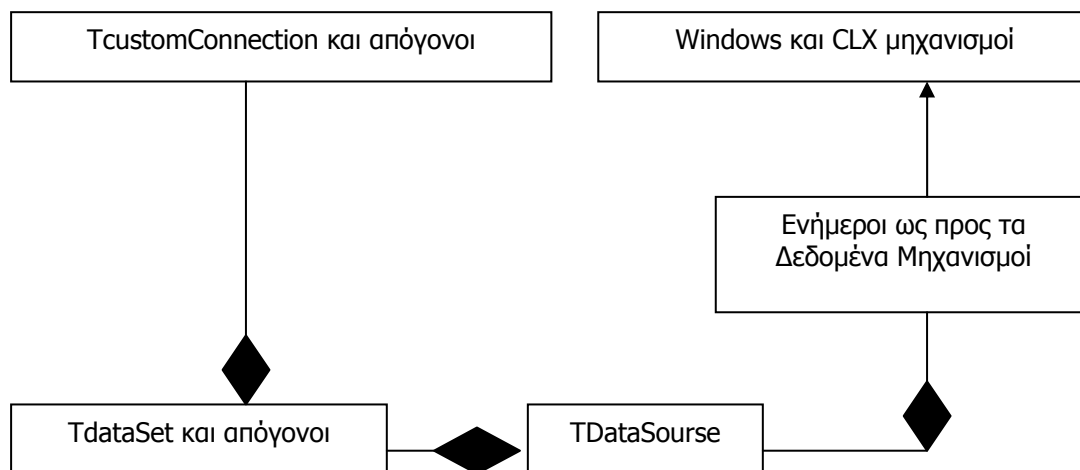
Ένα σύνολο συστατικών “μειωμένου φόρτου” (lightweight) με ανάλογα προγράμματα οδήγησης για ευρέως χρησιμοποιούμενες βάσεις δεδομένων client-server, τα οποία δεν απαιτούν το BDE.

Το dbExpress δηλαδή είναι μία ανεξάρτητη πλατφόρμα, μειωμένου φόρτου, όπως έχει προαναφερθεί, γρήγορη και ανοιχτή αρχιτεκτονική για την προσπέλαση Βάσεων Δεδομένων. Ένα πρόγραμμα οδήγησης dbExpress πρέπει να υλοποιεί πολλά συστήματα επικοινωνίας(interfaces) για την ανάκτηση meta – δεδομένων, την εκτέλεση ερωτημάτων SQL ή αποθηκευμένων διαδικασιών και την επιστροφή ενός cursor μονής κατεύθυνσης.

Το dbExpress αν και χαρακτηρίζεται σαν “ανεξάρτητο από την πλατφόρμα”, μπορεί να χρησιμοποιείται σε εφαρμογές VCL και CLX.

Η αξία των συστατικών βάσεων δεδομένων είναι ανάλογη με την αξία των αντίστοιχων συστατικών που χρησιμοποιούνται για την δημιουργία διεπιφανειών χρήστη. Σχεδόν κάθε συμβατικό συστατικό που χρησιμοποιείται για την δημιουργία μιας διεπιφάνειας χρήστη έχει ένα “ενήμερο για δεδομένα” (data aware) ισοδύναμο, το οποίο μπορεί να προσαρτηθεί σε ένα συστατικό TdataSource (μία πηγή δεδομένων). Τα συστατικά TdataSource μπορούν να συνδέονται σε συστατικά – απογόνους της κλάσης Tdataset, τα οποία είναι τα βασικά συστατικά προσπέλασης δεδομένων για όλες τις ομάδες συστατικών.

Παρακάτω, Εικόνα 13, φαίνονται οι σχέσεις μεταξύ αυτών των σημαντικών ομάδων συστατικών σε διαγραμματική μορφή.



Εικόνα 13

7. ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΕΡΓΑΣΙΑΣ ΚΑΙ ΕΠΕΞΗΓΗΣΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.

Το μεγαλύτερο μέρος της παρούσας εργασίας έχει πραγματοποιηθεί μέσω της γλώσσας προγραμματισμού C++ σε περιβάλλον C++ Builder σε συνεργασία με ένα RDBMS και της SQL. Το λογισμικό που θα παραχθεί θα επικοινωνεί με τον χρήστη με MS – Windows.

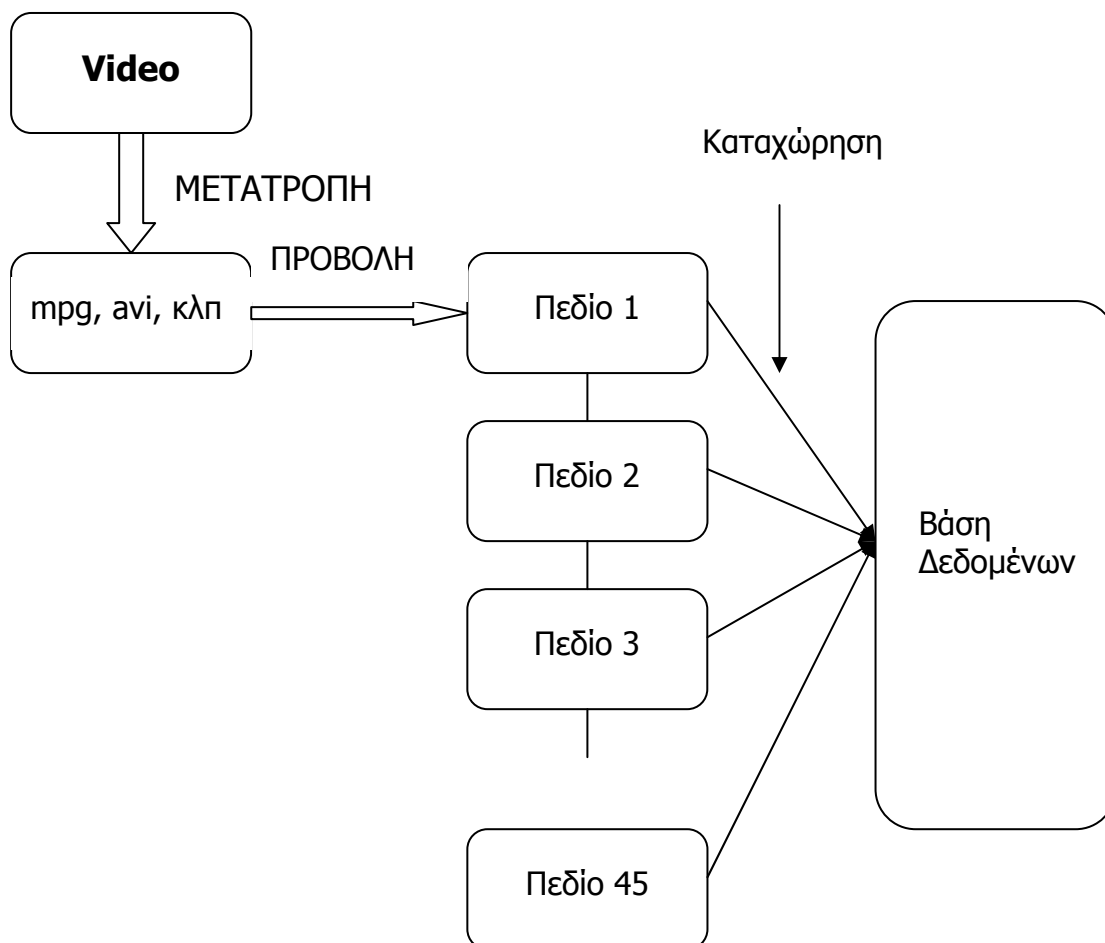
Η εργασία καλύπτει ένα μεγάλο μέρος των μαθημάτων του Τμήματος Πληροφορικής & Επικοινωνιών του Τ.Ε.Ι. Σερρών όπως Εισαγωγή στον Προγραμματισμό, Δομημένος Προγραμματισμός, Αντικειμενοστραφής Προγραμματισμός, Οπτικός Προγραμματισμός, Ψηφιακή Επεξεργασία Εικόνας, Βάσεις Δεδομένων I και Βάσεις Δεδομένων II.

Αρχικά θα δείξουμε πως λειτουργεί η όλη εργασία σχηματικά η οποία χωρίζεται σε δύο Φάσεις:

Α΄ ΦΑΣΗ ΚΑΤΑΓΡΑΦΗΣ

Στη πρώτη φάση, όπως φαίνεται και στην Εικόνα 14, υπάρχει ένα video, το οποίο μετατρέπεται σε μια από τις κατάλληλες μορφές video και ξεκινάει η προβολή του. Κατά τη διάρκεια της προβολής του παραπάνω καταχωρούνται στα διάφορα πεδία, ανάλογα με το περιεχόμενο τους, και όλα αυτά μαζί αποθηκεύονται σε μία βάση δεδομένων που είναι ήδη κατασκευασμένη και συνδεδεμένη με την όλη εφαρμογή.

Α' Φάση:

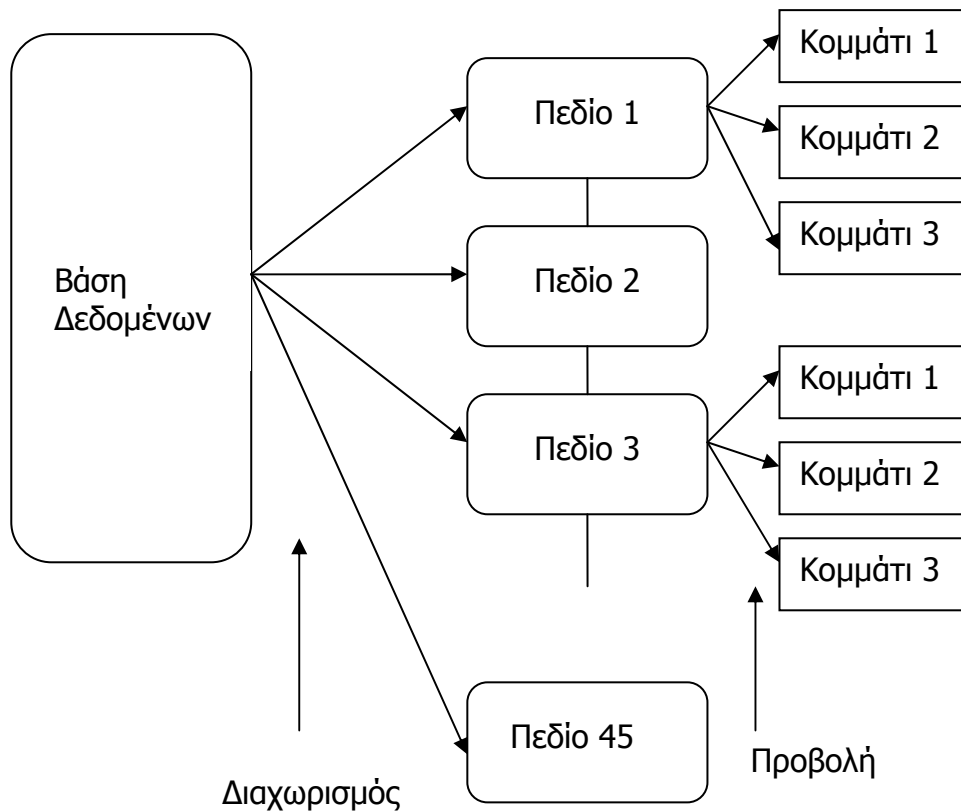


Εικόνα 14

Β' ΦΑΣΗ ΠΡΟΒΟΛΗΣ

Η δεύτερη Φάση, είναι ουσιαστικά το αντίστροφο της πρώτης αφού ξεκινάει από την βάση δεδομένων στην οποία είναι αποθηκευμένα τα “κομματιασμένα” πλέον video, segmentation , χωρίζονται αυτά στα πεδία τους και γίνεται η προβολή των διαφόρων κομματιών ανάλογα με το όμοιο περιεχόμενό τους. Εικόνα 15.

Β' Φάση:



Εικόνα 15

Η εφαρμογή έχει ως εξής:

Αρχικά, με το άνοιγμα του προγράμματος και την φόρτωση της εφαρμογής, εμφανίζεται μια φόρμα η οποία ονομάζεται FormStart με τα εξής κουμπιά, όπως φαίνεται και στη Εικόνα 16:



Εικόνα 16

Η παραπάνω φόρμα, όπως φαίνεται και στην εικόνα, περιέχει πέντε (5) κουμπιά με τις εξής λειτουργίες:

1. **ΑΝΟΙΓΜΑ ΝΕΑΣ ΦΟΡΜΑΣ**: Που οδηγεί στην φόρμα με ονομασία FormEditor.
2. **ΑΝΟΙΓΜΑ ΦΟΡΜΑΣ ΠΡΟΒΟΛΗΣ**: Που οδηγεί στην φόρμα με ονομασία FormScreening.
3. **ABOUT**: Που οδηγεί στην φόρμα με ονομασία About.
4. **CREATORS**: Που οδηγεί στην φόρμα με ονομασία Creators.
5. **EXIT**: Που οδηγεί στην έξοδο από την εφαρμογή.

Ο κώδικας της C++ Builder που αφορά την λειτουργία των παραπάνω κουμπιών είναι ο εξής:

```
// ΚΟΥΜΠΙ "ΑΝΟΙΓΜΑ ΝΕΑΣ ΦΟΡΜΑΣ"
```

```
void __fastcall TFormStart::Button1Click(TObject *Sender)
{
    FormEditor->Show();
}
```

```
//-----
```

```
// ΚΟΥΜΠΙ "ΑΝΟΙΓΜΑ ΦΟΡΜΑΣ ABOUT"
```

```
void __fastcall TFormStart::Button3Click(TObject *Sender)
{
    About->Show();
}
```

```
//-----
```

```
// ΚΟΥΜΠΙ "EXIT"
```

```
void __fastcall TFormStart::Button5Click(TObject *Sender)
{
    FormStart->Close();
}
```

```
//-----
```

```
// ΚΟΥΜΠΙ "CREATORS"
```

```
void __fastcall TFormStart::Button4Click(TObject *Sender)
{
    Creators->Show();
}
```

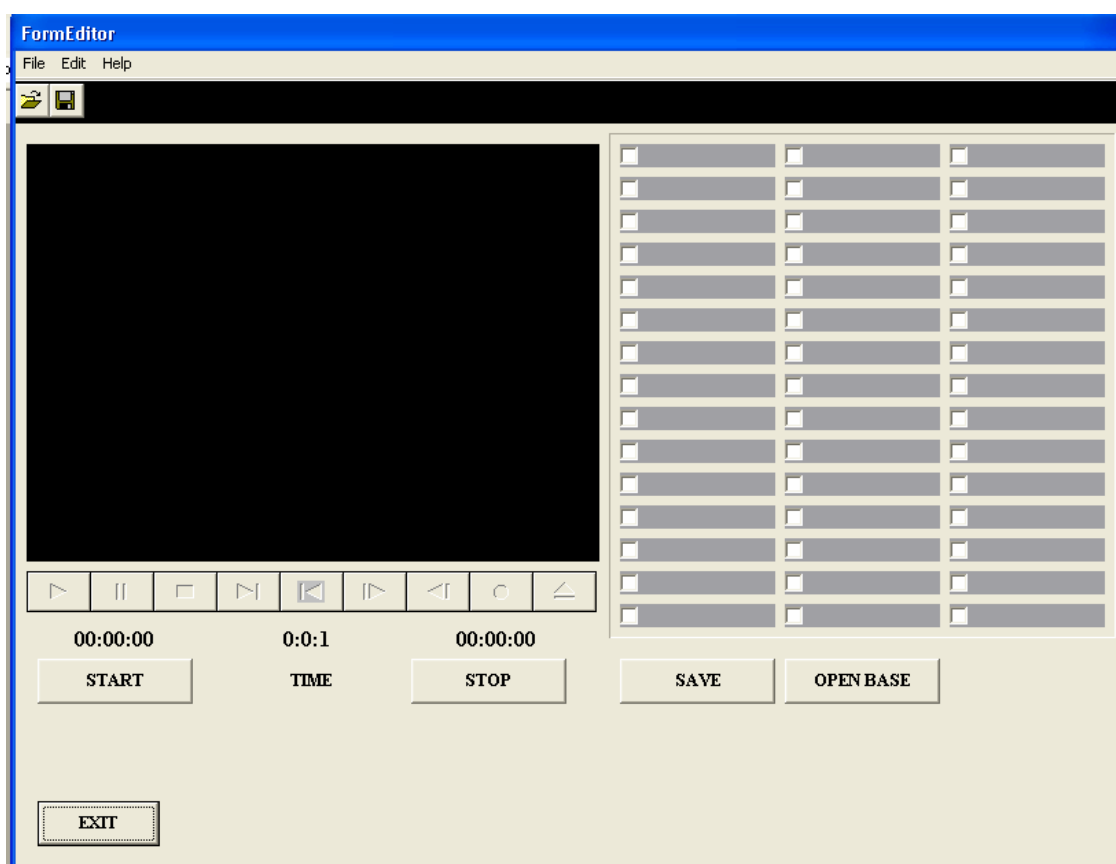
```
//-----
```

```
// ΚΟΥΜΠΙ "ΑΝΟΙΓΜΑ ΦΟΡΜΑΣ ΠΡΟΒΟΛΗΣ"
```

```
void __fastcall TFormStart::Button2Click(TObject *Sender)
{
    FormScreening->Show();
}
```

7. 1. ΑΝΑΛΥΤΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΑΘΕ ΚΟΥΜΠΙΟΥ ΚΑΙ ΦΟΡΜΑΣ.

ΑΝΟΙΓΜΑ ΝΕΑΣ ΦΟΡΜΑΣ: Πατώντας το συγκεκριμένο κουμπί, ανοίγει μια νέα φόρμα εργασίας ή πιο συγκεκριμένα, η FormEditor. Στην FormEditor πραγματοποιείται η πρώτη φάση της εργασίας, εκεί δηλαδή που γίνεται η προβολή του video, ο διαχωρισμός των κομματιών και η ταξινόμησή τους μέσα στη βάση δεδομένων. Η φόρμα καταχώρισης, όπως λέγεται, φαίνεται στην Εικόνα 17:



Εικόνα 17

Όπως φαίνεται και από την παραπάνω εικόνα στη φόρμα αυτή υπάρχει ένα menu που περιέχει τα file, edit, help. Επίσης, υπάρχει ένα animate όπου εκεί γίνεται η προβολή των video. Παρακάτω, ένας media player, καθορίζει την λειτουργία του media player και κατ' επέκταση του video δηλαδή τις συνηθισμένες λειτουργίες Play, Pause, Stop, Next, Prev, Step και Back.

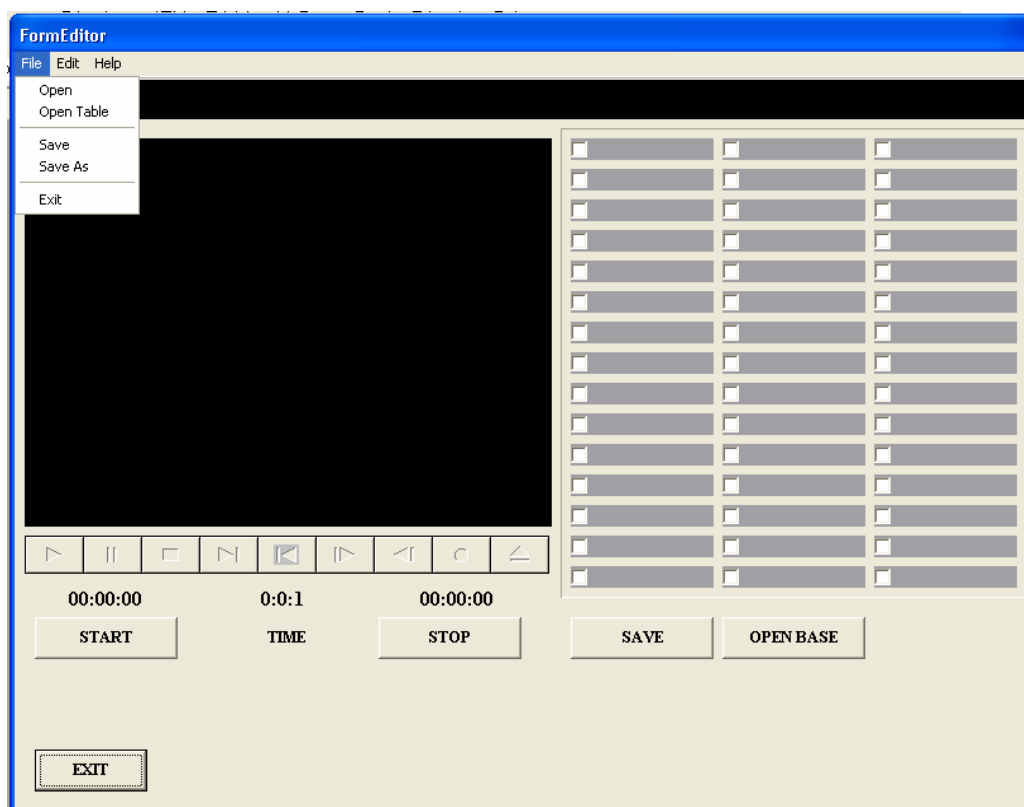
Ο Timer που υπάρχει διαδραματίζει πολύ σημαντικό ρόλο στην όλη εργασία αφού με βάση αυτό το χρόνο καθορίζονται τα “κομμάτια” των video που θα αποθηκευθούν μέσα στη Βάση Δεδομένων. Τα δύο κουμπιά με ονόματα Start και Stop, καταγράφουν την χρονική στιγμή που εμφανίζεται στον Timer2. Ο Timer2 δείχνει ουσιαστικά τον χρόνο της ταινίας που προβάλλεται.

Δεξιά της φόρμας, φαίνεται ένα σύνολο από CheckBoxes όπου θεωρείται το σημαντικότερο στοιχείο της παρούσας εργασίας. “Τσεκάροντας” τα παραπάνω, “τσεκάρονται” ταυτόχρονα και τα ανάλογα πεδία στην βάση δεδομένων.

Τέλος, το κουμπί στο κάτω μέρος του πλαισίου Bevel, το κουμπί save απλά αποθηκεύει και καθαρίζει τα πεδία ώστε να χρησιμοποιηθούν ενδεχομένως ξανά στην συνέχεια και το κουμπί Open Base μας οδηγεί στη φόρμα που υπάρχουν οι δύο πίνακες και καταχωρούνται τα στοιχεία.

Παρακάτω, ακολουθεί αναλυτική επεξήγηση όλων των παραπάνω.

Αρχικά, θα αναφερθεί το menu και μάλιστα το menu-file. Όπως φαίνεται στη παρακάτω εικόνα, Εικόνα 18, το menu-file περιέχει το open, open table, save, save as και exit.



Εικόνα 18

Το menu-file-open ανοίγει ένα open dialog όπου από 'κει μπορεί ο χρήστης να επιλέξει ένα οποιοδήποτε αρχείο video με σκοπό να το προβάλλει μέσω του media player στο animate. Για την εμφάνιση του Open Dialog υπάρχει και άλλος τρόπος μέσω του εικονιδίου Open από το Toolbar που βρίσκεται κάτω από τη γραμμή menu.

Ο κώδικας για την λειτουργία όλων των παραπάνω επιλογών παρουσιάζεται παρακάτω:

// MENU-FILE-OPEN

```
void __fastcall TFormEditor::Open1Click(TObject *Sender)
{
if (OpenDialog1->Execute()) //ME THN EKTELESH (EAN)
{
sec1=0; //ARXIKOPOIHSH
min1=0; //ARXIKOPOIHSH
hour1=0; //ARXIKOPOIHSH
tim=0; //ARXIKOPOIHSH
Katholic=0; //ARXIKOPOIHSH
Timer2->Enabled = false; //O Timer2 APENERGOPOIEITAI
MediaPlayer1->Enabled = true; //O MediaPlayer ENERGOPOIEITAI
MediaPlayer1->FileName=OpenDialog1->FileName; //KAI PAIRNEI TO NAME
MediaPlayer1->Open(); // AYTO POY EXEI KAI TO OpenDialog
MediaPlayer1->AutoEnable = false; // ANOIGEI KAI DEN EXEI
Label3->Caption=("0 : 0 : 0"); // AYTOMATH LEITOYRGIA
Label4->Caption=("0 : 0 : 0"); //MHDENISMOS - "ARXIKOPOIHSH"
Label6->Caption=("0 : 0 : 0"); // TIMER
}
}
```

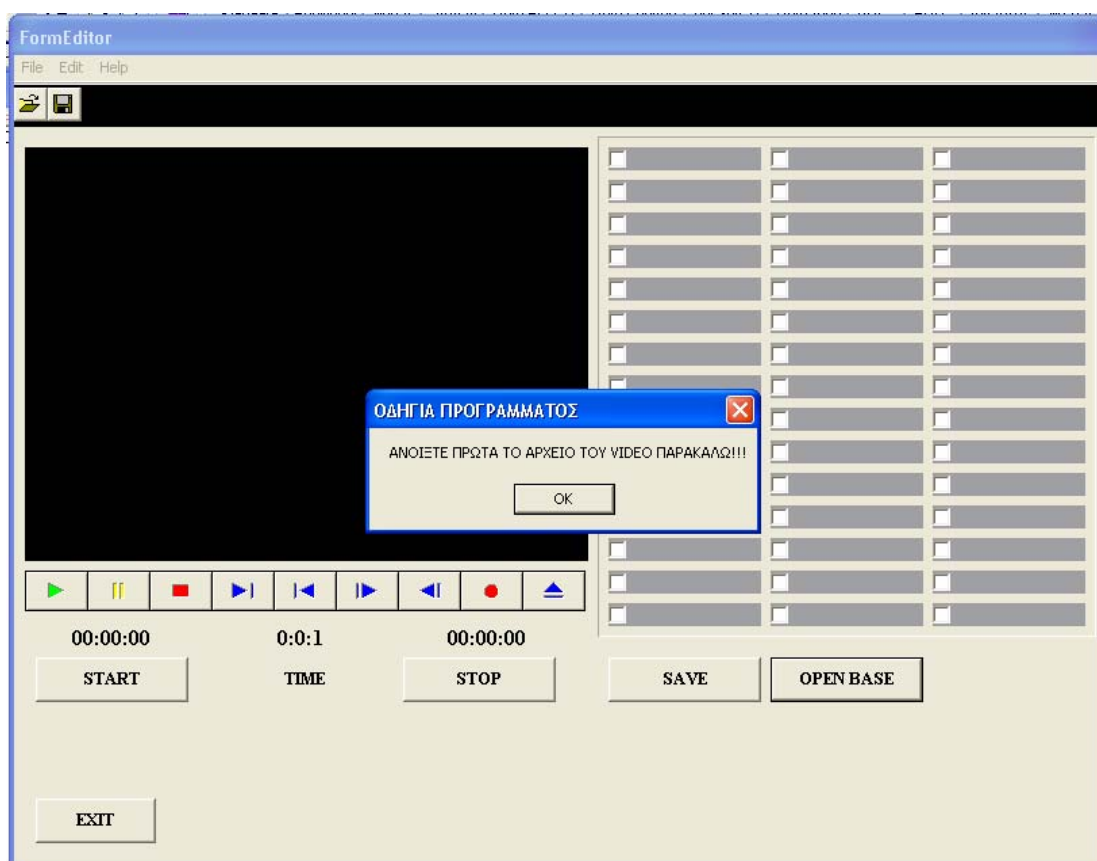
////////////////////////////////////

Έτσι λοιπόν, όπως φαίνεται και από τα παραπάνω, εάν το OpenDialog γίνει ενεργό και εκτελεστεί τότε συμβαίνουν οι ακόλουθες ενέργειες. Αρχικά θέτονται ίσες με το μηδέν οι συναρτήσεις sec1, min1, hour1, tim και Catholic όπου χρησιμεύονται αργότερα κατά τη διάρκεια της αναπαραγωγής του video.

Στη συνέχεια, ο MediaPlayer ενεργοποιείται και στο FileName του δίνει το ίδιο όνομα με αυτό που ανοίγει ο OpenFileDialog.

Τα Label, όπου εμφανίζονται οι χρόνοι κατά τη διάρκεια της προβολής, μηδενίζονται.

Αφού γίνει εκεί η επιλογή του επιθυμητού αρχείου τότε συνεχίζει ο χρήστης στη επόμενη επιλογή του menu, που είναι η file - Open Table. Στο σημείο αυτό θα πρέπει να διευκρινιστεί ότι πρέπει πρώτα να επιλεγεί το επιθυμητό video και ύστερα να ανοιχθεί ο πίνακας για να γίνουν οι αλλαγές στα CheckBoxes. Σε αντίθετη περίπτωση θα εξαφανιστεί ένα μήνυμα με τις οδηγίες, δηλαδή πρώτα να ανοιχτεί το αρχείο του video και ύστερα η παρούσα φόρμα. Εικόνα 19.



Εικόνα 19

Ο κώδικας για τη παραπάνω οδηγία φαίνεται παρακάτω:

```

//Menu-File-Open Table
void __fastcall TFormEditor::OpenTable1Click(TObject *Sender)
{
if(!MediaPlayer1->FileName.IsEmpty()) //EAN TO FileName TOY
  CheckBoxTable->Show(); //MediaPlayer DEN DEN EINAI "KENO"
else // TOTE NA EMFANISTEI O PINAKAS CheckBoxTable
  ALLIWS NA EMFANISTEI ENA MHNHYMA "ODHGIAS PROGRAMMATOS"
  {
  int i = MessageBox(NULL, "ΑΝΟΙΞΤΕ ΠΡΩΤΑ ΤΟ ΑΡΧΕΙΟ ΤΟΥ VIDEO
ΠΑΡΑΚΑΛΩ!!!" , "ΟΔΗΓΙΑ ΠΡΟΓΡΑΜΜΑΤΟΣ" , MB_OK);
  Hide();
  if((i == ID_OK) || (i == ID_CLOSE))
  Show();
  }
}
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

Η επιλογή Open Table ανοίγει έναν πίνακα που περιέχει σαράντα πέντε (45) πεδία που αντιστοιχούν στα 45 CheckBoxes. Αυτός ο πίνακας αναφέρεται στον χαρακτηρισμό των αρχικών video με δυνατότητα χειρονακτικού προσδιορισμού. Δηλαδή, καλώντας το menu - Open Table ανοίγει ο πίνακας της Εικόνας 20 με ονομασία CheckBox Table. Πίσω από την επιλογή του menu αυτού υπάρχει ο παραπάνω κώδικας.

Όπως φαίνεται και από τον κώδικα πατώντας την επιλογή menu-file-open table ανοίγει η ακόλουθη φόρμα. Αλλιώς, εμφανίζεται ένα μήνυμα το οποίο ζητάει στο χρήστη απλά να ανοίξει πρώτα το αρχείο του video και ύστερα να ακολουθήσει αυτήν την επιλογή.

Key	Value
ΠΕΔΙΟ 1	
ΠΕΔΙΟ 2	
ΠΕΔΙΟ 3	
ΠΕΔΙΟ 4	
ΠΕΔΙΟ 5	
ΠΕΔΙΟ 6	
ΠΕΔΙΟ 7	
ΠΕΔΙΟ 8	
ΠΕΔΙΟ 9	
ΠΕΔΙΟ 10	
ΠΕΔΙΟ 11	
ΠΕΔΙΟ 12	
ΠΕΔΙΟ 13	
ΠΕΔΙΟ 14	
ΠΕΔΙΟ 15	
ΠΕΔΙΟ 16	
ΠΕΔΙΟ 17	
ΠΕΔΙΟ 18	
ΠΕΔΙΟ 19	
ΠΕΔΙΟ 20	
ΠΕΔΙΟ 21	
ΠΕΔΙΟ 22	
ΠΕΔΙΟ 23	
ΠΕΔΙΟ 24	
ΠΕΔΙΟ 25	

Εικόνα 20

Όπως φαίνεται αριστερά υπάρχουν τα 45 Πεδία αριθμημένα όπως και στην FormEditor και δεξιά 45 πεδία κενά για να συμπληρωθούν από τον χρήστη. Πατώντας το κουμπί Change στα δεξιά γίνονται αυτόματα οι αλλαγές στα 45 πεδία της FormEditor. Αν κάποια από τα πεδία δεν συμπληρωθούν τότε δεν αλλάζει το όνομα τους και παραμένουν απενεργοποιημένα. Το τελευταίο θα φανεί κατά την παρουσίαση ενός παραδείγματος στη συνέχεια. Το κουμπί exit οδηγεί στην έξοδο από τον παρόντα πίνακα.

Το κουμπί Change “κρύβει” τον κώδικα που καθορίζει τον τρόπο με τον οποίο θα γίνουν οι αλλαγές. Έτσι αφού καθοριστούν οι τιμές που θα παίρνει μία μεταβλητή τύπου String, χρησιμοποιείται μία if και μία else για να

καθοριστούν ποιές τιμές θα πάρουν τα πεδία. Δηλαδή, το Checkbox θα είναι απενεργοποιημένο, `enabled=false`, αν το πεδίο είναι κενό αλλιώς θα είναι ενεργοποιημένο, `enabled=true`, και θα παίρνει την τιμή που έχει πληκτρολογήσει ο χρήστης.

Παρακάτω φαίνεται ένα μέρος του κώδικα που αφορά ένα από τα CheckBoxes και πως λειτουργούν αυτά:

```
//KOYMPI CHANGE
void __fastcall TCheckBoxTable::Button1Click(TObject *Sender)
{
if(!DM->Table3->IsEmpty() ) //EAN O Table3 (TCAPTIONS) DEN EINAI
{
// ADEIOS TOTE
DM->Table3->Last() ; //NA KLHTHEI H Last() KAI H First() AFOY
int j=DM->Table3->RecordCount , i=0; // KSEKINHSEI TO "METRHMA"
DM->Table3->First() ; // TWN EGGRAFWN STO PINAKA
while(i++<j)
{
if(DM->Table3NAME->Value == FormEditor->MediaPlayer1->FileName)
{
DM->Table3->Edit() ; //KATHWS GINONTAI TA PARAPANW
break; //EAN O PINAKAS EXEI THN TIMH POY EXEI
} //KAI O TO FileName TOY MediaPlayer
else //TOTE NA ARXISEI H KATAGRAFH KAI NA
{ //NA PROSTHETEI EGGRAFES STO TELOS THS
DM->Table3->Next() ; //LISTAS. DHLADH EDW KANEI ENAN ELEGXO
//TWN PEDIWN TOY PINAKA

if(i==j)
DM->Table3->Append() ;
}
}
}
else
DM->Table3->Append() ;

DM->Table3NAME->Value = FormEditor->MediaPlayer1->FileName ;
String Temp;
```



```

//DBCheckBox1-----
Temp=CheckBoxTable->EditTable->Strings[0][0]; //DINEI MIA TIMH STH
Temp.Delete(1,8); // SYNARTHSH TEMP POY EINAI ISH ME TO Strings[0][0]
//GIA NA KATXWRHTHEI TO ONOMA POY THA DWSOUME
if (Temp==String("")) //EAN H TIMH TEMP EINAI ISH ME
{
//String(""), DEN EXEI KAPOIA TIMH
FormEditor->CheckBox1->Enabled = false; // TOTE TO CheckBox NA
FormEditor->CheckBox1->Visible = false; // GINETAI KAI ANENERGO
} // ALLA KAI NA MHN EMFANIZETAI
else //ALLIWS
{
FormEditor->CheckBox1->Caption=Temp; //STO Caption TOY
DM->Table3CH01->Value = Temp; // CheckBox NA DWTHEI TO
FormEditor->CheckBox1->Enabled=true; // NAME POY DWSAME
FormEditor->CheckBox1->Visible = true; // STO CH01KAI NA GINEI
} // ENERGO KAI EMFANISIMO
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

Ο παραπάνω ο κώδικας μας επιτρέπει να καταλάβουμε πως γίνεται η σύνδεση αυτού του πίνακα και των πεδίων αυτών με τα πεδία που υπάρχουν στη φόρμα “Form Editor”. Έτσι λοιπόν αν ο πίνακας που αφορά τα στοιχεία αυτής της φόρμας είναι κενός τότε είναι ελεύθερος για να ξεκινήσουν οι εγγραφές. Και έτσι, ο MediaPlayer παίρνει στοιχεία πλέον από τα στοιχεία του πίνακα αυτού αφού οι τιμές τους γίνονται πλέον ίδιες.

Έτσι, το CheckBox θα παίρνει δύο τιμές, ή true ή false, θα είναι δηλαδή ενεργό και θα παίρνει την τιμή που θα πληκτρολογούμε εμείς στον προηγούμενο πίνακα ή θα είναι ανενεργό και απλά δε θα φαίνεται. Ίδια διεργασία ακολουθείται και για τα υπόλοιπα CheckBoxes.

Αφού γίνουν οι απαραίτητες αυτές αλλαγές, τότε μπορεί να αρχίσει και η προβολή του video. Με την έναρξη του video ξεκινάει και η λειτουργία του Timer2 και έτσι αρχίζει να μετράει και ο χρόνος από την αρχή του video. Κατά την διάρκεια της προβολής πιέζοντας το κουμπί Start καταγράφεται ο χρόνος ακριβώς από πάνω σ’ ένα Label αλλά και παράλληλα στον πίνακα της Βάσης Δεδομένων. Το ίδιο ακριβώς γίνεται αν πιεστεί και το κουμπί Stop, με

την διαφορά ότι σταματάει και η προβολή του video τη συγκεκριμένη εκείνη χρονική στιγμή.

Ο κώδικας που έχει χρησιμοποιηθεί στη γλώσσα προγραμματισμού C++ για τα κουμπιά Start και Stop φαίνεται παρακάτω:

```
//KOYMPI "Start"
void __fastcall TFormEditor::Button1Click(TObject *Sender)
{
  AnsiString h = hour1;      //DHLWSH METAVLHTWN
  AnsiString m = min1;      //DHLWSH METAVLHTWN
  AnsiString s = sec1;      //DHLWSH METAVLHTWN
  Label3->Caption=(h + ":" + m + ":" + s);

  DM->Table1->Last() ;

  if(!DM->Table1->IsEmpty())    //EAN O PINAKAS THS BASHS EXEI
    i = DM->Table1->FieldValues["A/A"] + 1; // EGGRAFES TOTE NA
  else                          // PROXWRHSEI STHN EPOMENH KAI NA
    i= 1;                        // PROSTHESEI ENAN AYKSODA ARITHMO
                                // AKOMA, DIAFORETIKA ARXIKOPOIEI TON
  DM->Table1->Append();    // AYKSODA ARITHMO ME ENA, H TIMH
                          // START DINETAI TO ONOMA TOY ARXEIOY POY ANOIXTHKE APO
  DM->Table1->FieldValues["A/A"] = i; // TO OpenFileDialog KAI EINAI ISH ME THN
  DM->Table1->FieldValues["NAME"] = OpenFileDialog1->FileName;
  DM->Table1->FieldValues["START"] = Catholic; // METAVLHTH Catholic
}

////////////////////////////////////
```

Αρχικά λοιπόν τα AnsiString h, AnsiString m και AnsiString s παίρνουν την τιμή που τους δόθηκε πριν, δηλαδή ίση με τα hour1, min1 και sec1 δηλαδή ίση με το μηδέν και αυτά καταγράφονται στο Label που εμφανίζεται ο χρόνος.

Παράλληλα, στο DataModule όπου υπάρχουν οι πίνακες και καταγράφεται τόσο ο αύξοντα αριθμός του video όσο και το όνομα του που δεν είναι άλλο από το path του.

```

//KOYMPI STOP
void __fastcall TFormEditor::Button2Click(TObject *Sender)
{
AnsiString h=hour1;           //DHLWSH METAVLHTWN
AnsiString m=min1;           //DHLWSH METAVLHTWN
AnsiString s=sec1;           //DHLWSH METAVLHTWN

Label4->Caption=(h + ":" + m + ":" + s);

Timer2->Enabled=False;       //STAMATAEI O TIMER KAI O
MediaPlayer1->Stop();        // MediaPlayer KAI H TIMH STOP EINAI
DM->Table1->FieldValues["STOP"] = Catholic; // AYTH POY
KATAGRAFETAI STH METAVLHTH Catholic TOY TIMER
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

```

Όσο αφορά το κώδικα του “Stop” όταν ο Timer απενεργοποιηθεί, σταματάει την λειτουργία του ο MediaPlayer και καταγράφεται ο χρόνος στο πεδίο που αντιστοιχεί στο stop.

Στη συνέχεια, αφού έχει καταγραφεί το χρονικό αυτό διάστημα, μελετάτε από το χρήστη το συγκεκριμένο κομμάτι του video με σκοπό να εντοπίσει το περιεχόμενο του και να καταχωρηθεί ανάλογα στη data base. Από την στιγμή που “τσεκάρει” τα ανάλογα πεδία του πίνακα και έχει ολοκληρώσει τη φάση αυτή, πιέζει το κουμπί save ώστε να ενημερωθεί η βάση δεδομένων, να αποθηκευθούν οι αλλαγές σ’ αυτή και να καθαριστούν τα πεδία, διότι στη συνέχεια μπορεί να ξαναχρησιμοποιηθούν τα ίδια πεδία ανάλογα με το περιεχόμενο του video. Έτσι, στη Βάση Δεδομένων το ίδιο πεδίο μπορεί να χρησιμοποιηθεί παραπάνω από μία φορές.

Στο σημείο αυτό θα πρέπει να αναφερθεί και η λειτουργία του menu – file - save όπου κάνει ακριβώς την ίδια λειτουργία με το κουμπί save που βρίσκεται στην δεξιά κάτω πλευρά της φόρμας, όπως αναφέρθηκε και παραπάνω. Υπάρχει και ένας τρίτος τρόπος για την αποθήκευση και τον καθαρισμό των πεδίων. Η λειτουργία του εικονιδίου Save που βρίσκεται στη ToolBar.

Η λειτουργία του κουμπιού “Save” βασίζεται στο ακόλουθο κώδικα όπου αρχικά στο πεδίο δίνει την τιμή που έχει και το CheckBox όταν είναι τσεκαρισμένο και στη συνέχεια αρχίζει το “διάβασμα” των πεδίων και έτσι όπου υπάρχει τιμή την αποθηκεύει στη βάση δεδομένων αλλιώς της δίνει την τιμή false:

```
//Button Save
void __fastcall TFormEditor::Button3Click(TObject *Sender)
{
//APOTHIKEYSH STON Table1 TSEKARISMENA TACheckBoxes KAI
//PAIRNOYN THN ANTISTOIXH TIMH
DM->Table1->FieldValues["CH01"] = CheckBox1->Checked;
DM->Table1->FieldValues["CH02"] = CheckBox2->Checked;
DM->Table1->FieldValues["CH03"] = CheckBox3->Checked;
.....
DM->Table1->FieldValues["CH44"] = CheckBox44->Checked;
DM->Table1->FieldValues["CH45"] = CheckBox45->Checked;

DM->Table1->Post();

int n;
for ( n=0; n<ControlCount; n++)
    if (Controls[n]->ClassNameIs("TCheckBox"))
        if (((TCheckBox*)Controls[n])->Enabled)
            ((TCheckBox*)Controls[n])->Checked=false;
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Έπειτα, πατώντας το play του media player ξεκινάει τόσο η προβολή του video όσο και ο χρόνος από κει που σταμάτησε. Πατώντας πάλι το start και το stop ξεκινάει πάλι η ίδια διαδικασία μέχρι το τέλος της προβολής του video.

Ο κώδικας που αναφέρεται στην λειτουργία του MediaPlayer είναι ο ακόλουθος:

```
//MediaPlayer
void __fastcall TFormEditor::MediaPlayer1Click(TObject *Sender,
    TMPBtnType Button, bool &DoDefault)
{
    if (Button==btPlay)           //EAN PATITHEI TO KOYMPI Play
    {
        Timer2->Enabled=true;    //TOTE ENERGOPOIEITAI O TIMER
        MediaPlayer1->Play();    //KAI KSEKINAEI NA "PAIZEI" KAI O
    }                             // MediaPlayer

    else                          //ALLIWS
    if (Button==btStop)          //EAN PATITHEI TO KOYMPI Stop
    {
        Timer2->Enabled=false;   //TOTE APENERGOPOIEITAI O TIMER
        MediaPlayer1->Pause();   //KAI STAMATAEI NA "PAIZEI" KAI O
    }                             // MediaPlayer
}
}
```

////////////////////////////////////

Έτσι, αν πατηθεί το κουμπί Play του MediaPlayer τότε ενεργοποιείται ο Timer, και αρχίζει να “παίζει” το video. Στην αντίθετη περίπτωση, που πατηθεί δηλαδή το κουμπί Stop, σταματάει και ο χρόνος, αφού απενεργοποιείται ο Timer, αλλά και η προβολή του video.

Όλες οι παραπάνω διεργασίες αποθηκεύονται, όπως έχει αναφερθεί, σε μία Βάση Δεδομένων. Η βάση αυτή μπορεί να εμφανιστεί μόνο πατώντας το κουμπί στη φόρμα Form Editor με ονομασία Open Base.

The screenshot shows a software window titled "ΠΙΝΑΚΕΣ" (Tables) with a title bar containing standard Windows window controls. The main area is divided into two sections:

ΠΙΝΑΚΑΣ ΣΤΟΙΧΕΙΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ
 This section contains a table with the following columns: A/A, NAME, START, STOP, CH01, CH02, CH03, and a small square icon. Below the table is a scroll bar and a set of navigation buttons (back, forward, search, etc.) and a "CLEAR" button.

ΠΙΝΑΚΑΣ CAPTIONS ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ
 This section contains a table with the following columns: NAME and CH01. Below the table is a scroll bar and a set of navigation buttons (back, forward, search, etc.) and a "CLEAR" button.

At the bottom of the window, there is an "EXIT" button.

Εικόνα 21

Στη παραπάνω φόρμα, Πίνακες, όπως φαίνεται και στη Εικόνα 21, υπάρχουν δύο πίνακες οι οποίοι περιέχουν όλα τα στοιχεία εκείνα που είναι απαραίτητα για την λειτουργία του προγράμματος, δηλαδή τον χαρακτηρισμό των κομματιών και τις χρονικές στιγμές στις οποίες εμφανίζονται.

Ο πρώτος πίνακας, εμφανίζει όλα τα στοιχεία του video. Η πρώτη στήλη τον αύξοντα αριθμό των κομματιών, η δεύτερη στήλη την ονομασία του video που ουσιαστικά είναι το path στο οποίο είναι αποθηκευμένο το video, στις απόμενες δύο στήλες φαίνεται ο χρόνος έναρξης και τερματισμού του συγκεκριμένου κομματιού και τέλος στις επόμενες 45 στήλες φαίνονται τα 45 πεδία. Το σπουδαιότερο κομμάτι αυτού του μέρους είναι η λειτουργία των παραπάνω πεδίων. Τα παραπάνω πεδία παίρνουν μόνο δύο τιμές, την True και False. Όταν δηλαδή τσεκάρουμε ένα πεδίο στην Form Editor τότε η τιμή που παίρνει το πεδίο αυτό στη βάση δεδομένων είναι True, διαφορετικά η τιμή του είναι False.

Ο δεύτερος πίνακας της παρούσας φόρμας, μας βοηθάει στην εμφάνιση των χαρακτηρισμένων, μέσω της χειρονακτικής διαδικασίας,

πεδίων. Έτσι, όταν δίνουμε ένα όνομα για κάποιο πεδίο αλλάζει στον δεύτερο πίνακα αφού οι δύο παραπάνω πίνακες είναι συνδεδεμένοι μεταξύ τους.

Κάτω από τους δύο πίνακες φαίνονται δύο DBNavigators όπου βοηθούν στην μετακίνηση του χρήστη μέσα στη βάση δεδομένων καθώς και δύο κουμπιά με το όνομα Clear όπου μπορούν να «καθαρίσουν» όλες τις εγγραφές των πινάκων αντίστοιχα.

//CLEAR

```
void __fastcall TForm3::Button1Click(TObject *Sender)
{
    DM->Table1->First();      //KSEKINONTAS APO THN PRWTH TIMH
    while(!DM->Table1->IsEmpty()) //KAI OPOY VLEPEI OTI YPARXEI AGGRAFH
    {
        DM->Table1->Delete(); //DIEGRAPSE
        DM->Table1->Next();   //KAI PROXWERA STO EPOMENO
    }
}
```

////////////////////////////////////

Όλα τα παραπάνω, θα φανούν καλύτερα σε παράδειγμα που θα δοθεί στη συνέχεια.

Η επιλογή menu – file - Save As, ανοίγει ένα Save Dialog όπου μπορεί να αποθηκεύσει τα video με την επιθυμητή ονομασία που θέλει ο κάθε χρήστης.

//FILE-SAVE AS

```
void __fastcall TFormEditor::SaveAs1Click(TObject *Sender)
{
    if (SaveDialog1->Execute())
    {
        fn=SaveDialog1->FileName;
        MediaPlayer1->Save();
    }
}
```

////////////////////////////////////

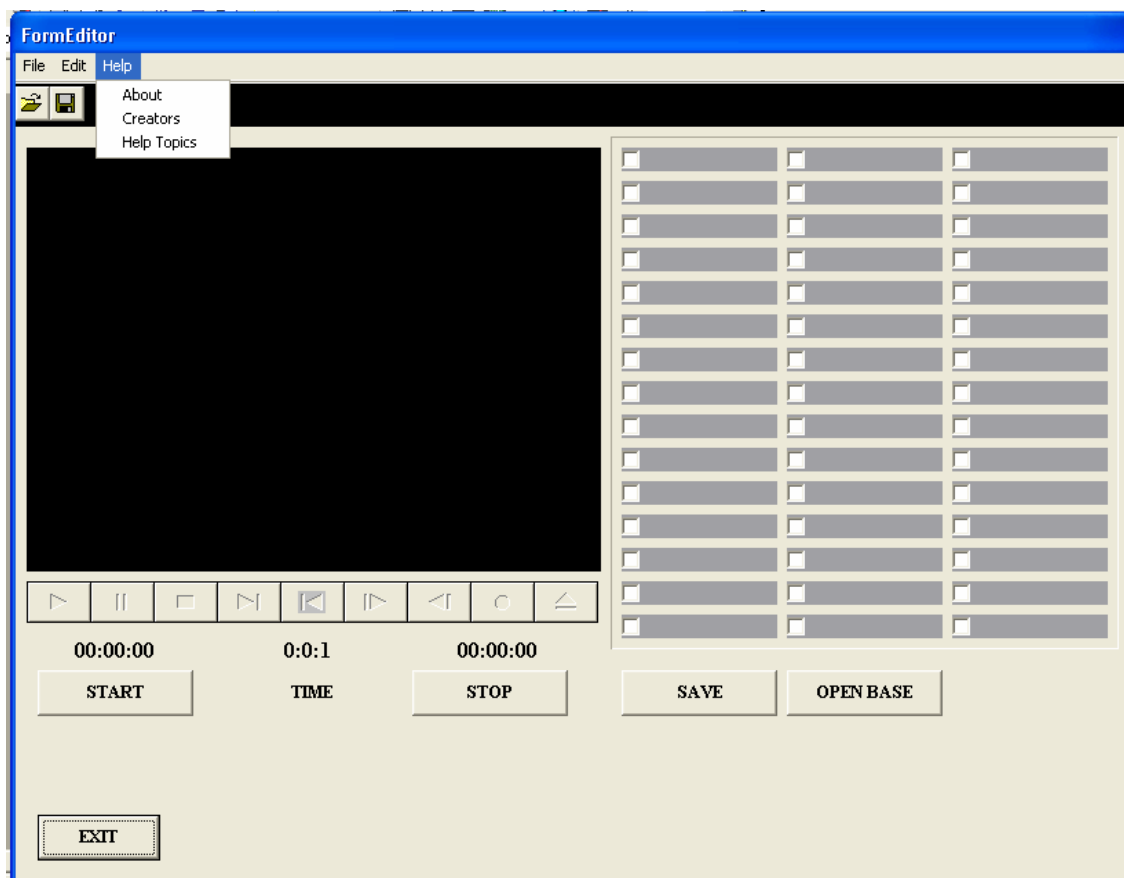
Όταν ολοκληρωθεί λοιπόν η προβολή και η κατάλληλη επιλογή και καταχώρηση των “κομματιών”, τότε κλείνει την Form Editor. Εδώ, μπορούμε να χρησιμοποιήσουμε την επιλογή menu – file - exit όπου κλείνει την παρούσα φόρμα στην οποία εργαζόμασταν και επιστρέφουμε στην αρχική μας φόρμα, τη FormStart ή το κουμπί στη κάτω αριστερή γωνία με το όνομα exit.

```
//MENU-FILE-EXIT
```

```
void __fastcall TFormEditor::Exit1Click(TObject *Sender)
{
  FormEditor->Close();      //KLEISIMO FORMAS
}
```

```
////////////////////////////////////
```

Το menu - Help μας οδηγεί σε τρεις φόρμες την About, Creators και Help Topics, Εικόνα 22.



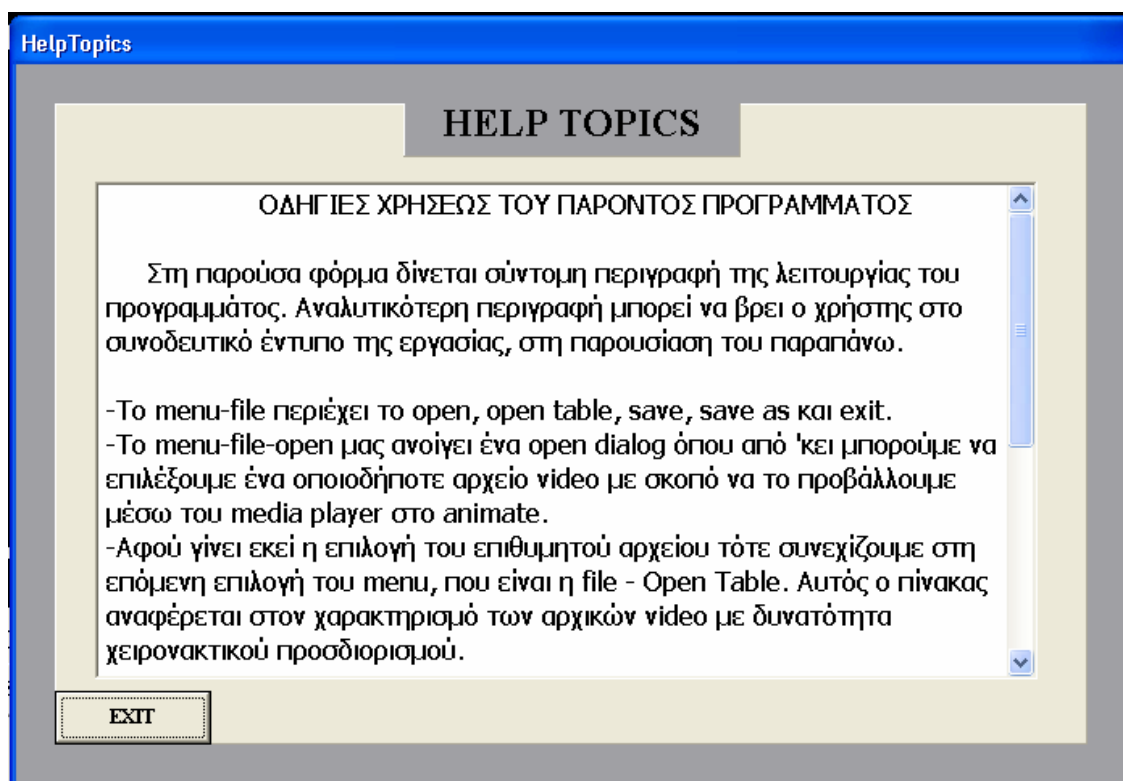
Εικόνα 22

Οι δύο πρώτες φόρμες, About και Creators βρίσκονται και στην FormStart.

Από το menu – Help - About καταλήγει κανείς στην φόρμα About όπου εκεί αναφέρονται περιληπτικά στοιχεία της εργασίας για τη λειτουργία της και τη χρησιμότητα της.

Από το menu – Help - Creators οδηγείται ο χρήστης στην φόρμα Creators όπου εκεί αναφέρονται κάποια στοιχεία για τον κατασκευαστή του προγράμματος αυτού και τον υπεύθυνο καθηγητή που έχει αναλάβει την πτυχιακή εργασία.

Τέλος, από το menu – Help - Help Topics καταλήγει ο χρήστης στην φόρμα Help Topics, Εικόνα 23, όπου εκεί αναφέρονται συνοπτικά στοιχεία που μπορούν να βοηθήσουν τον απλό χρήστη να χρησιμοποιήσει το παρόν πρόγραμμα.



Εικόνα 23

//FORMA ABOUT

```
void __fastcall TFormEditor::About1Click(TObject *Sender)
{
    About->Show();           //ANOIGMA FORMAS
}
```

```
//-----
```

//FORMA CREATORS

```
void __fastcall TFormEditor::N3Click(TObject *Sender)
{
    Creators->Show();       //ANOIGMA FORMAS
}
```

```
//-----
```

//FORMA HELP TOPICS

```
void __fastcall TFormEditor::He1Click(TObject *Sender)
{
    HelpTopics->Show();     //ANOIGMA FORMAS
}
```

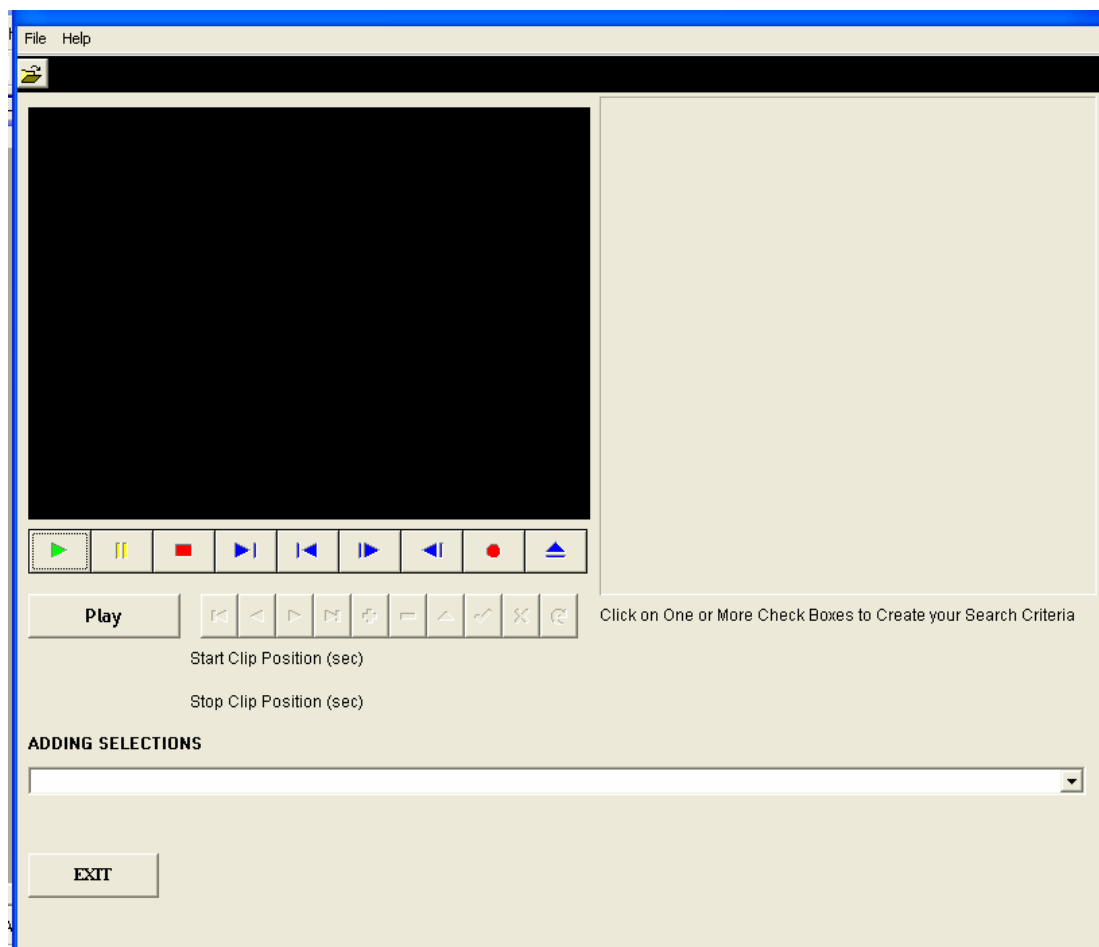
```
////////////////////////////////////
```

ANOIGMA ΦΟΡΜΑΣ ΠΡΟΒΟΛΗΣ: Στη συνέχεια αφού έχει ολοκληρωθεί η επεξήγηση της νέας φόρμας, και ο χρήστης βρίσκεται στην αρχική φόρμα, πιέζοντας το κουμπί Άνοιγμα Φόρμας Προβολής, κατευθύνεται στην φόρμα που φαίνεται στην Εικόνα 24, και είναι η φόρμα προβολής ή FormScreening.

//ANOIGMA FORMAS PROVOLHS

```
void __fastcall TFormStart::Button2Click(TObject *Sender)
{
    FormScreening->Show();   //ANOIGMA FORMAS
}
```

```
////////////////////////////////////
```



Εικόνα 24

Στη φόρμα αυτή, υπάρχει ένα `animate` και ένας `Media Player` όπου προβάλλεται το `video` και μπορεί ο χρήστης να δει τα αποτελέσματα της προηγούμενης διαδικασίας.

Δεξιά υπάρχει ένας χώρος που φιλοξενεί μια σειρά από `CheckBoxes`, με σκοπό κάθε φορά που επιλέγει κάποιο από τα `CheckBoxes` να προβάλλεται και το ανάλογο κομμάτι. Τα ονόματα που έχουν τα παραπάνω δεν είναι άλλα από τα ονόματα που έχουν δοθεί στα `CheckBoxes` της προηγούμενης φάσης.

Ακριβώς από κάτω βρίσκεται ένα `ComboBox`, με όνομα `Adding Sollections`, όπου εμφανίζονται τα `video` που έχουν χρησιμοποιηθεί και μέσω εκείνου μπορεί ο χρήστης να επιλέξει όποια από αυτά τα `video` θέλει και να τα προβάλλει. Αυτόματα, με το που γίνεται η επιλογή του `video`, εμφανίζονται και τα `CheckBoxes` με τα συγκεκριμένα ονόματα, έτοιμο για προβολή.

Όσο αφορά το κουμπί της φόρμας Play δε κάνει κάτι άλλο από το να παίζει τα συγκεκριμένα κομμάτια.

Τέλος, το menu – file περιέχει τις επιλογές Open που ανοίγει τα video που έχουν χρησιμοποιηθεί κατά την προηγούμενη διαδικασία, καθώς και την επιλογή exit όπου οδηγεί στη έξοδο από την παρούσα φόρμα. Το menu – Help περιλαμβάνει μόνο μία επιλογή, την Help topics όπου οδηγεί σε φόρμα με σύντομες οδηγίες χρήσης του προγράμματος όπως ακριβώς και στην φόρμα Form Editor.

Έτσι λοιπόν, αφού εισέλθει ο χρήστης στη παρούσα φόρμα, αρχικά θα πρέπει να οδηγηθεί στο menu – file- open. Κάνοντας αυτή την ενέργεια δε θα ανοίξει κάποιο Open Dialog, ούτε φαινομενικά θα γίνει κάποια ενέργεια. Κι όμως αυτό που γίνεται είναι το άνοιγμα του ενός ή των περισσοτέρων video που χρησιμοποιήθηκαν κατά την Α΄ Φάση. Αυτό μπορεί να το δει ο χρήστης πηγαίνοντας απλά Adding Selection όπου εκεί θα εμφανιστούν όλα τα χρησιμοποιούμενα videos, από όπου μπορεί να επιλέξει όποια από αυτά θέλει.,

Για το άνοιγμα λοιπόν των video που ήδη έχουν χρησιμοποιηθεί στην Α΄ Φάση “ευθύνεται” ο επόμενος κώδικας:

//Open

```
void __fastcall TFormScreening::ToolButton1Click(TObject *Sender)
{
if(ComboBox1->Items->Count == 0 )    //EAN TO ComboBox EXEI TH TIMH
{
        //MH DEN TOTE KALOYNTAI OI SYNARTHSEIS
DM->Table3->Last() ;                // Last() KAI First() KAI GINETAI
int ok = DM->Table3->RecordCount , i=0; //H KATAMETRHS
DM->Table3->First() ;
while(i++ <ok)
{
    ComboBox1->AddItem(DM->Table3NAME->Value , DM->Table3 );
    DM->Table3->Next() ;                //KATHWS GINONTAI AYTA, STO ComboBox THA
}
        //PROSTITHETAI MIA TIMH POY THA EINAI TO
DM->Table3->First() ;                //NAME TOY KATHE VIDEO POY EXOYME
}
        //XRHSIMOPOIHSEI KAI KATALHGEI TELIKA
}
        //KSANA STH PRWTH EGGRAFH
}
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

Αρχίζει λοιπόν και διαβάζει από τον πίνακα που έχουν καταχωρηθεί τα ονόματα των videos. Και έτσι όταν βρίσκει κάποιο path στη στήλη με τα name τότε το “ενεργοποιεί” και το τοποθετεί στη λίστα για να προβληθεί στη συνέχεια.

Πιέζοντας το κουμπί του ComboBox εμφανίζεται η λίστα με τα video που είχαμε ήδη επιλέξει κατά την πρώτη φάση. Αφού λοιπόν επιλέξει τα επιθυμητά videos, τότε ακριβώς από πάνω στον κενό χώρο εμφανίζονται τα CheckBoxes που δεν είναι άλλα από τα CheckBoxes που χρησιμοποιήθηκαν στην άλλη φάση και έχουν τα ίδια ονόματα. Εκεί βέβαια θα εμφανιστούν μόνο τα CheckBoxes εκείνα που έχουν χρησιμοποιηθεί και στην Φάση της καταχώρησης των videos. Όλα τα υπόλοιπα CheckBoxes δεν εμφανίζονται διότι δεν επιτελούν καμία εργασία. Για την απόκρυψη τους “υπεύθυνο” είναι το παρακάτω μέρος του κώδικα:

//APOKRIPSI CheckBox

```
void __fastcall TFormScreening::FormCreate(TObject *Sender)
{
    for(int i=0; i<ComponentCount; i++) //GIA KATHE METRHMA POY THA
    {
        // GINETAI
        if(Components[i]->ClassNameIs("TCheckBox")) //KAI AN H TCheckBox GINEI
        {
            // TIMH TOY Components[i] TOTE
            ((TCheckBox*)Components[i])->Visible = false; //NA MHN EMFANIZONTAI KAI
            ((TCheckBox*)Components[i])->Checked = false; // NA GINONTAI ANENERGA
            ((TCheckBox*)Components[i])->Caption = ""; // KATHWS KAI TO Caption NA
        }
        //PARAMENEI KENO GIA KATHE PERIPTWSH POY TA
    }
    //TCheckBox THS PRWTHS FASHS DEN EXOUN TIMH
}
}
```

////////////////////////////////////

Ξεκινάει λοιπόν το διάβασμα των τιμών από την αρχή μέχρι το τέλος, και όταν δεν συναντάει στο CheckBox κάποιο όνομα τότε δίνει την ιδιότητα στο CheckBox να μην είναι ορατό και να παραμένει ανενεργό.

Επιλέγοντας λοιπόν ένα video τα CheckBoxes που εμφανίζονται έχουν τα ίδια ονόματα με τα CheckBoxes της FormEditor. Κάνει λοιπόν μία αντιστοιχία των Captions εκτελώντας ένα ερώτημα:

//ANTISTOIXIA

```
void __fastcall TFormScreening::ComboBox1Change(TObject *Sender)
{

AnsiString sql2;           //DHLWSH METAVLHTWN
int itemplay2 ;           //DHLWSH METAVLHTWN
AnsiString sqlFirst2;     //DHLWSH METAVLHTWN
AnsiString sqlLast2;     //DHLWSH METAVLHTWN
sql2="";
itemplay2=0;

itemplay2 = ComboBox1->ItemIndex ;    //DHLWSH METAVLHTWN

for(int i=0; i<ComponentCount; i++)
{
if(Components[i]->ClassNameIs("TCheckBox"))
{
((TCheckBox*)Components[i])->Visible = false;
((TCheckBox*)Components[i])->Checked = false;
((TCheckBox*)Components[i])->Caption = "";
}
}

sqlFirst2="select * from TCAPTIONS where Name = '"+ ComboBox1->Items-
>Strings[itemplay2] + "'";
DM->Query2->SQL->Text=(sqlFirst2);    //EKTELESH ERWTHMATOS
DM->Query2->ExecSQL();                //EPELEKSE APO TON PINAKA TCAPTIONS
DM->Query2->Active=true;              // OPOY TO ONOMA EINAI TO PATH POY
                                     // VRISKETAI TO ARXEIO

if(!DM->Query2->FieldByName("CH01")->IsNull) //EAN KATA THN
{
                                     // EKTELESH VREI THN EGGRAPH ME TO ONOMA CH01
```

```

if(CheckBox1->Visible==false)    //KAI AN AYTO EXEI THN TIMH FALSE
{
    //KANTO TRUE, KAI DWSE STO CAPTION TOY
    CheckBox1->Visible = true;    //TH TIMH POY EPELEXTHIKE ME TO
    CheckBox1->Caption=" ";      // ERWTHMA
    CheckBox1->Caption = DM->Query2->FieldByName("CH01")->Value ;
}
}

```

////////////////////////////////////

Στη συνέχεια, επιλέγοντας απλά ένα από τα CheckBox και πατώντας το κουμπί Play ξεκινάει η προβολή εκείνου του συγκεκριμένου κομματιού που περιέχεται σ' αυτό το CheckBox και παράλληλα στα δύο DBText εμφανίζεται ο χρόνος έναρξης του κομματιού παίζει αυτή τη στιγμή και ο χρόνος λήξης.

//PLAY

```

void __fastcall TFormScreening::Button1Click(TObject *Sender)

```

```

{

```

```

    AnsiString sql;           //DHLWSH METAVLHTWN

```

```

    int itemplay ;           //DHLWSH METAVLHTWN

```

```

    AnsiString sqlFirst;     //DHLWSH METAVLHTWN

```

```

    AnsiString sqlLast;      //DHLWSH METAVLHTWN

```

```

    sql="";                  //DHLWSH METAVLHTWN

```

```

    itemplay=0;

```

```

    itemplay=ComboBox1->ItemIndex;

```

```

    sqlFirst="select * from TVIDEOBASE where Name = '"+ ComboBox1->Items-
    >Strings[itemplay] + "'";

```

```

    if (CheckBox1->Checked)    //EKTELESH ERWTHMATOS

```

```

        sql=sql+ " and CH01=true" ; //EPELEKSE APO TON PINAKA TVIDEOBASE

```

```

    if (CheckBox2->Checked)    //OPOY TO ONOMA EINAI TO PATH TO OPOIO

```

```

        sql=sql+ " and CH02=true" ; //EMFANIZETAI KAI STO ComboBox, OPOY KAI

```

```

    if (CheckBox3->Checked)    //PAIRNEI APO KEI TH TIMH.

```

```

        sql=sql+ " and CH03=true" ; //EAN TO CheckBox EINAI TSEKARISMENO

```

```

    if (CheckBox4->Checked)    //EXEI DHLADH TH TIMH TRUE

```

```

        sql=sql+ " and CH04=true" ; //TOTE H sql THA EINAI ISH ME THN sql +

```

```

    if (CheckBox5->Checked)    // THN TSEKARISMENH TIMH TOY

```

```

    sql=sql+ " and CH05=true" ;      //SYGKEKRIMENOY CheckBox DHLADH
if (CheckBox6->Checked)           //TO KATHE CheckBox THS PAROYSAS
    sql=sql+ " and CH06=true" ;    //FORMAS THA ADISTOIXEI ME TA
if (CheckBox7->Checked)           // CheckBoxes THS FORM EDITOR
    sql=sql+ " and CH07=true" ;
if (CheckBox8->Checked)
    sql=sql+ " and CH08=true" ;

.....

    sql=sql+ " and CH43=true" ;
if (CheckBox44->Checked)
    sql=sql+ " and CH44=true" ;
if (CheckBox45->Checked)
    sql=sql+ " and CH45=true" ;

sqlLast=sql;                      //ETSI H sqlLast THA EINAI ISH ME THN sql
DM->Query1->SQL->Text=(sqlFirst+sqlLast);
DM->Query1->ExecSQL();             //KAI THA EKTELEITAI TO PROTO ERWTHMA
DM->Query1->Active=true;          //OPOY TO TEXT THA EINAI ISO ME THN
                                  // sqlFirst + sqlLast
if(DM->Query1->FindFirst()){       //EAN TO Query1 VREI THN PRVTH EGRAFH
    DBText1->DataSource=DM->DataSource3; //TOTE STA DBText NAME,
    DBText1->DataField="Name";      // START, KAI STOP NA KATAGRAFOUN
    DBText2->DataSource=DM->DataSource3; // OI ANALOGES TIMES POY HDH
    DBText2->DataField="Start";     // EINAI APOTHIKEUMENES STH
    DBText3->DataSource=DM->DataSource3; // VASH DEDOMENWN
    DBText3->DataField="Stop";

FormScreening->MediaPlayer1->Enabled = true; //GINETAI O MediaPlayer
FormScreening->MediaPlayer1->FileName=DBText1->Caption; // ENERGOS
FormScreening->MediaPlayer1->Open(); // "PAIRNEI" TO NAME APO TO
                                  // DBText KAI ANOIGEI

MediaPlayer1->TimeFormat = tfMilliseconds;

MediaPlayer1->StartPos = StrToInt(DBText2->Caption) * 1000;
MediaPlayer1->EndPos = StrToInt(DBText3->Caption) * 1000;
MediaPlayer1->Play();

```



```

DBNavigator1->DataSource=DM->DataSource3;
DM->Query1->Next();           //EKTELEITE TO ERWTHMA GIA THN
MediaPlayer1->StartPos = StrToInt(DBText2->Caption) * 1000; //EPOMENH
MediaPlayer1->EndPos = StrToInt(DBText3->Caption) * 1000; // EGGRAPH
MediaPlayer1->Play();         // KATAGRAFONTAI OI XRONIKES STIGMES KAI
}                               //ARXIZEI NA "PAIZEI"
}

```

////////////////////////////////////

Όπως φαίνεται από τα παραπάνω εκτελείται το ερώτημα και ζητάει αρχικά να επιλέξουμε από συγκεκριμένο πίνακα και που έχει όνομα ουσιαστικά το path του και σε περίπτωση που το βρει το CheckBox να είναι τσεκαρισμένο τότε θα αντιστοιχήσει ουσιαστικά τα CheckBox της πρώτης φάσης με τα CheckBox της Β' φάσης. Όλα τα παραπάνω θα παρουσιαστούν στη συνέχεια και αυτά σε συγκεκριμένο παράδειγμα.

ABOUT: Πιέζοντας το κουμπί About της αρχικής φόρμας, ανοίγει μία ακόμα φόρμα με το ίδιο όνομα όπου δίνονται στοιχεία για τη Πτυχιακή Εργασία, τη λειτουργία της και τον σκοπό της. (Εικόνα 22)

CREATORS: Τέλος, το κουμπί Creators, οδηγεί στην φόρμα εκείνη που δίνει στοιχεία για τον υπεύθυνο καθηγητή και κατασκευαστή της εργασίας αυτής. Εικόνα 25.



Εικόνα 25

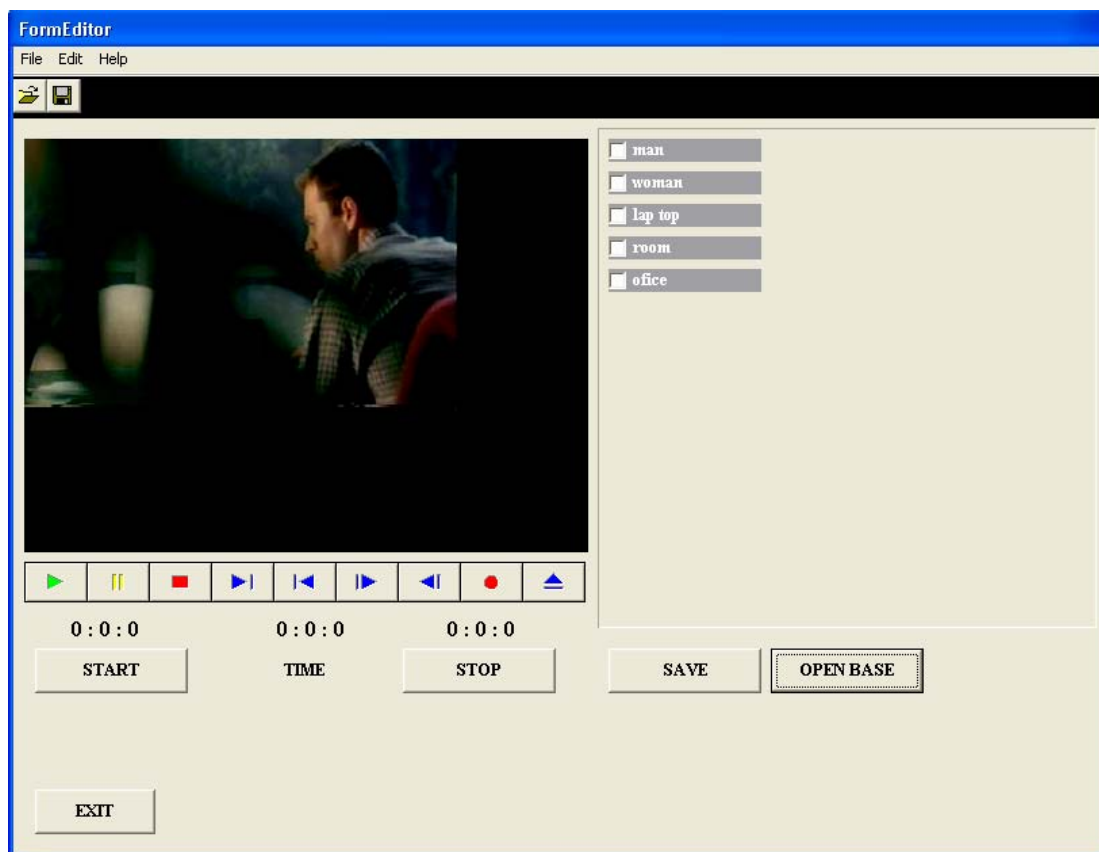
8. ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΕΡΓΑΣΙΑΣ - ΠΑΡΑΔΕΙΓΜΑΤΑ.

Στη συνέχεια θα παρουσιαστούν ορισμένα παραδείγματα για να φανούν τα αποτελέσματα της εφαρμογής.

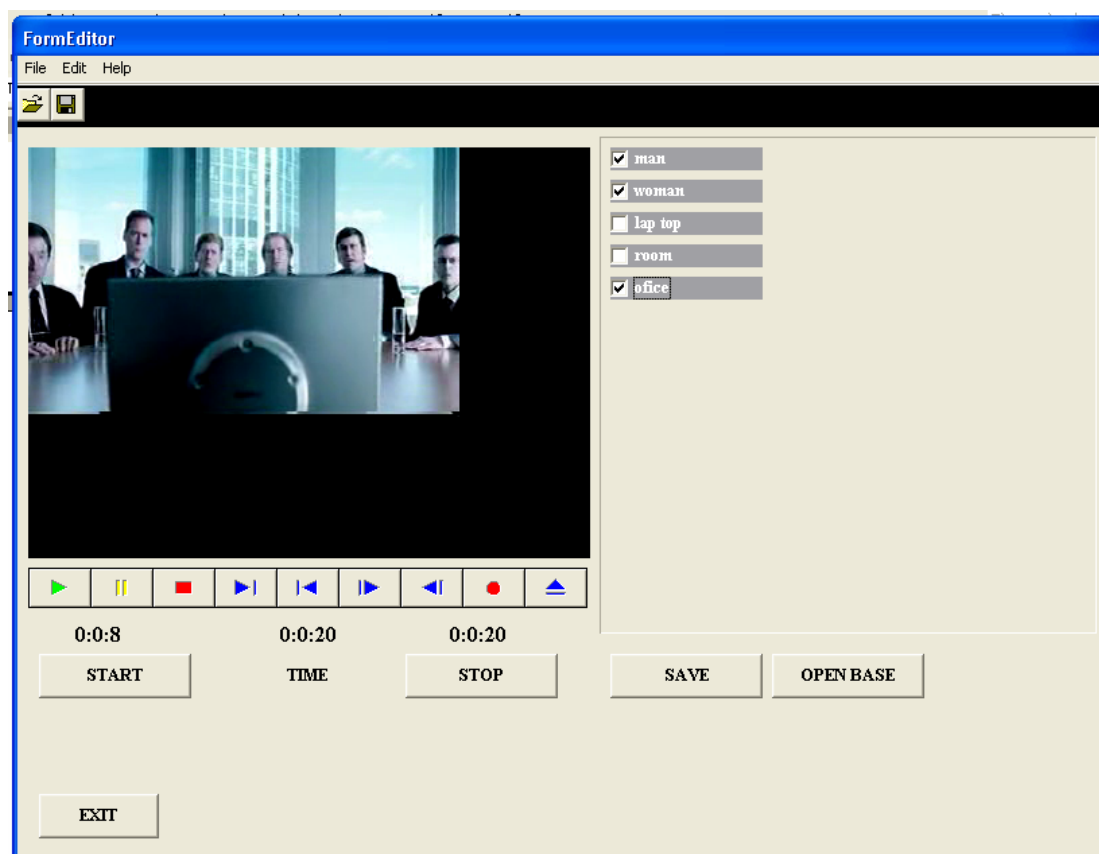
Αρχικά ο χρήστης “τρέχει” την εφαρμογή και εμφανίζεται η αρχική φόρμα απ’ όπου οδηγείται στην Form Editor, στην οποία και θα κάνει την επεξεργασία του video. Αφού ανοίξει πρώτα το video από την επιλογή file – open, στη συνέχεια ανοίγει τον πίνακα για να αλλάξει τα ονόματα των checkboxes.

Η προβολή του video αλλά και ο χρόνος ξεκινάει όταν πατηθεί το κουμπί play από τον media player. Πατώντας το κουμπί Start καταγράφεται ο χρόνος την συγκεκριμένη χρονική στιγμή και πατώντας το κουμπί Stop σταματάει τόσο ο χρόνος, όπου και καταγράφεται, αλλά και η προβολή του κομματιού.

Έτσι, στην Εικόνα 26, μπορεί να δει κανείς την πρώτη αυτή φάση όπου έχει επιλεχθεί το video και έχουν αλλαχθεί τα ονόματα στα checkboxes.

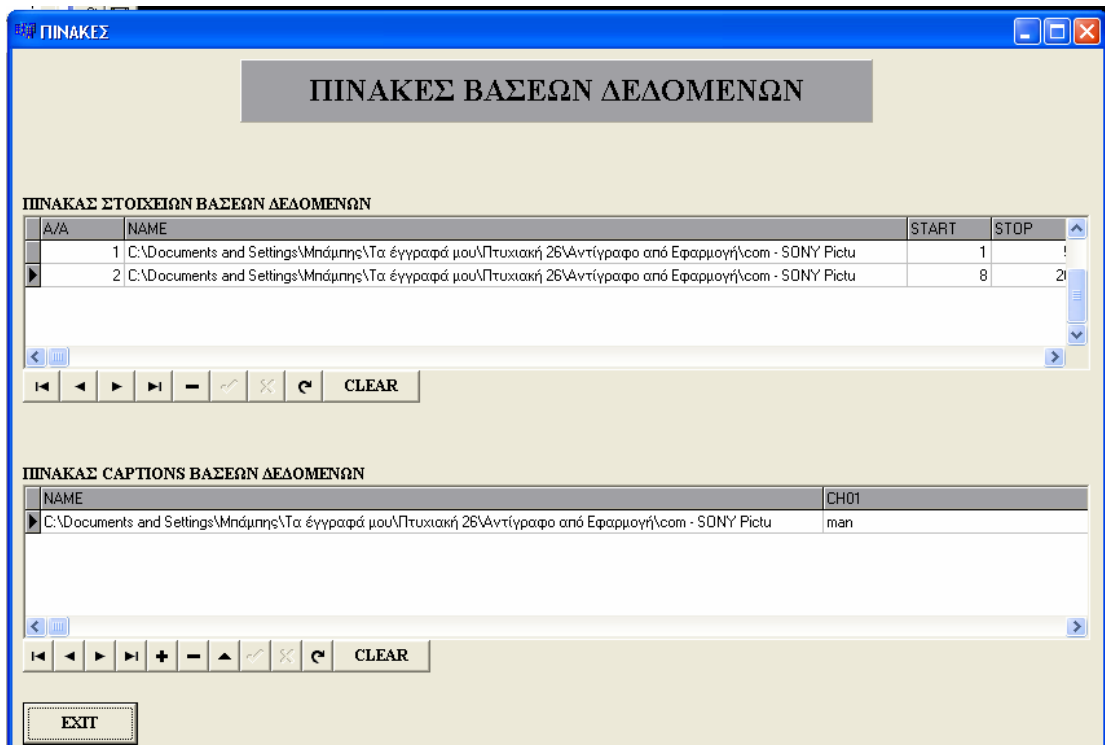


Εικόνα 26

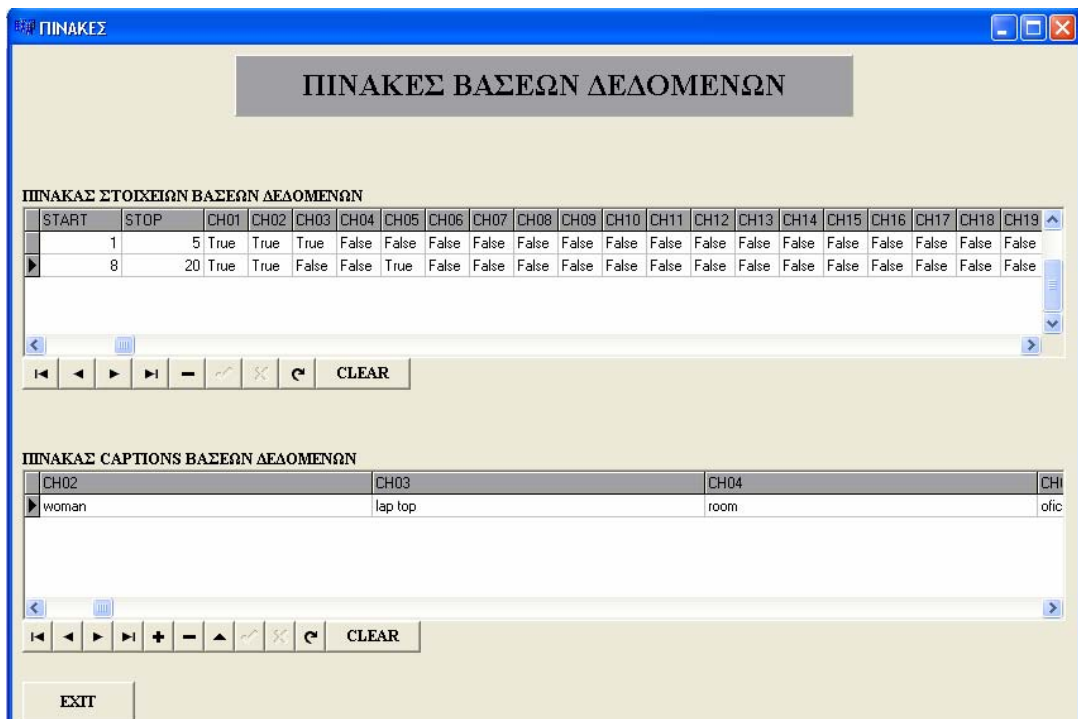


Εικόνα 27

Στην Εικόνα 27 παραπάνω φαίνονται ορισμένα από τα checkboxes που είναι τσεκαρισμένα, καθώς και το video στην αριστερά πλευρά της φόρμας “μέσα” στο animate. Στην Εικόνα 28, μπορεί κανείς να δει την απεικόνιση της ένωση του προγραμματισμού με τις Βάσεις Δεδομένων. Μπορεί να δει δηλαδή την Βάση Δεδομένων, έναν πίνακα ουσιαστικά που είναι καταχωρημένα το όνομα του video που παίχτηκε που δεν είναι άλλο από το path στο οποίο βρίσκεται, τους χρόνους έναρξης και τερματισμού των κάθε κομματιών.



Εικόνα 28



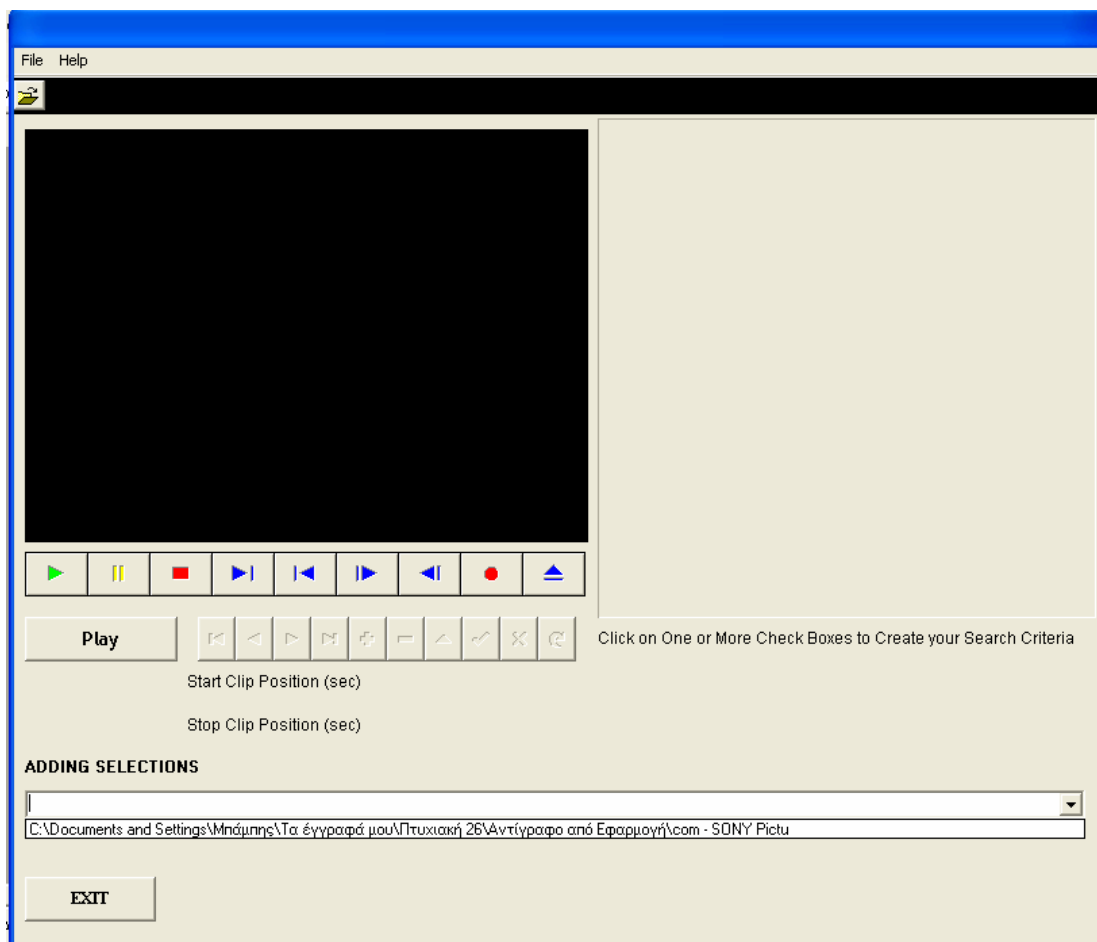
Εικόνα 29

Στην Εικόνα 29, όπου δείχνει τον ίδιο πίνακα με τα παραπάνω αλλά για λόγους χωρητικότητας εμφανίζεται σε δύο εικόνες, φαίνονται τα CheckBoxes όπου αν παρατηρήσουμε καλά υπάρχουν δύο τιμές: η true και η false. True

παίρνουν τα πεδία όπου έχουν τσεκαριστεί ενώ false εκείνα που δεν έχουν τσεκαριστεί καθώς και εκείνα που είναι ανενεργά δηλαδή εκείνα που δεν έχουν πάρει όνομα και δεν εμφανίζονται ούτε στην φόρμα που εργαζόμασταν.

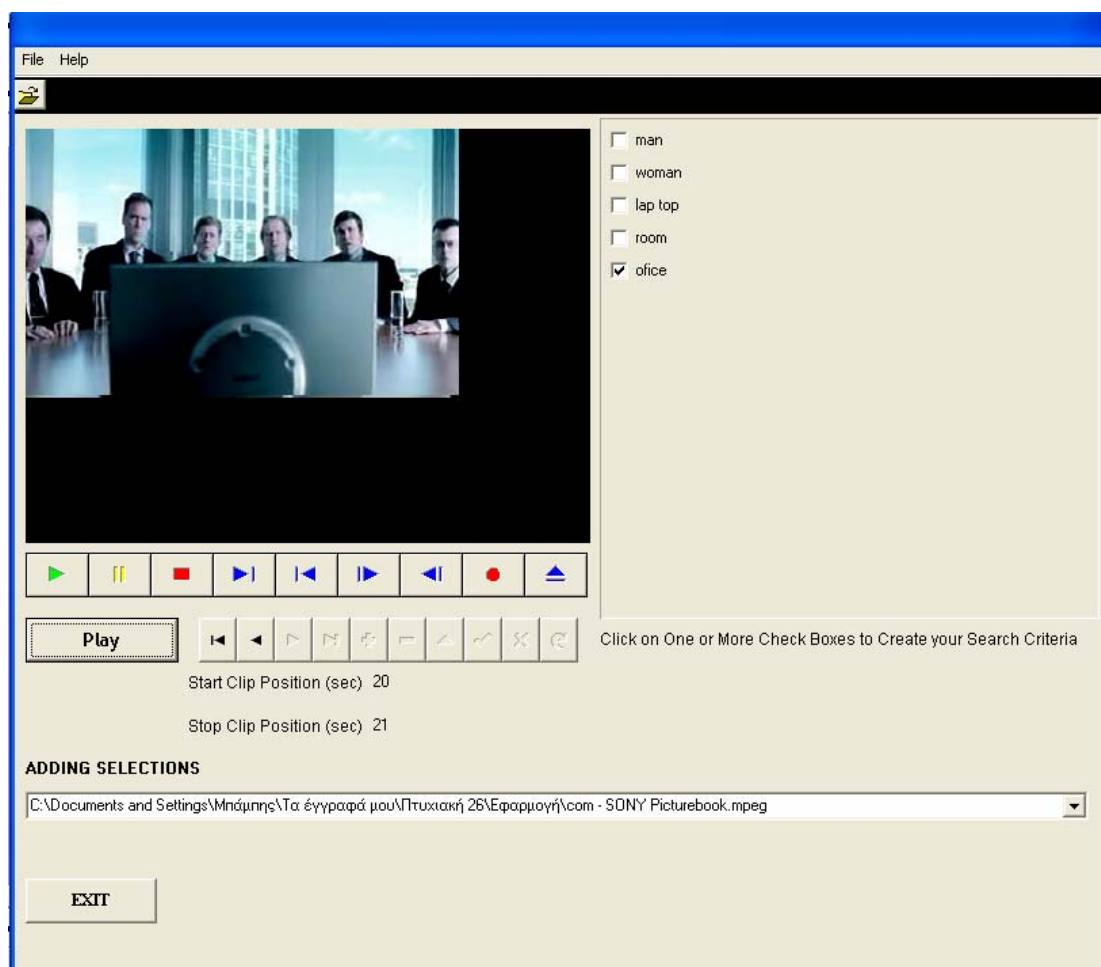
Στον δεύτερο πίνακα της Βάσης Δεδομένων μπορούμε να διακρίνουμε τα ονόματα που έχουν δοθεί στα CheckBoxes και τα οποία βρίσκονται σε αντιστοιχία με τον επάνω πίνακα. Τα πεδία που δεν έχουν λάβει ονόματα κατά την διάρκεια της επεξεργασίας των κομματιών παραμένουν κενά.

Στη συνέχεια αφού έχει ολοκληρωθεί η πρώτη φάση της εργασίας κλείνει η φόρμα και ο χρήστης από την αρχική φόρμα μεταπηδά στη Φόρμα όπου θα γίνει η προβολή των κομματιών. Πηγαίνοντας λοιπόν, στην επιλογή file – open αρχικά ο χρήστης θα διαπιστώσει πως δεν προέκυψε καμία μεταβολή. Κι όμως, αν ο χρήστης κατευθυνθεί στο Adding Selections θα διαπιστώσει ότι εκεί υπάρχουν όλα τα video που έχουν χρησιμοποιηθεί, στη παρούσα φάση μόνο ένα, Εικόνα 30.



Εικόνα 30

Επιλέγοντας λοιπόν το video, τότε στη δεξιά πλευρά της φόρμα θα εμφανιστούν τα CheckBoxes. Τα ονόματα που έχουν τα CheckBoxes δεν είναι άλλα από τα ίδια που είχαν δοθεί στα CheckBoxes της πρώτης φόρμας αφού είναι άμεσα συνδεδεμένα μεταξύ τους μέσα από ερωτήματα. Έτσι λοιπόν, αν πατήσει ο χρήστης ένα από τα CheckBoxes, έστω το πέμπτο και πατήσει το κουμπί Play ξεκινάει η προβολή των κομματιών τα οποία έχουν καταχωρηθεί στο συγκεκριμένο πεδίο της Βάσης Δεδομένων.



Εικόνα 31

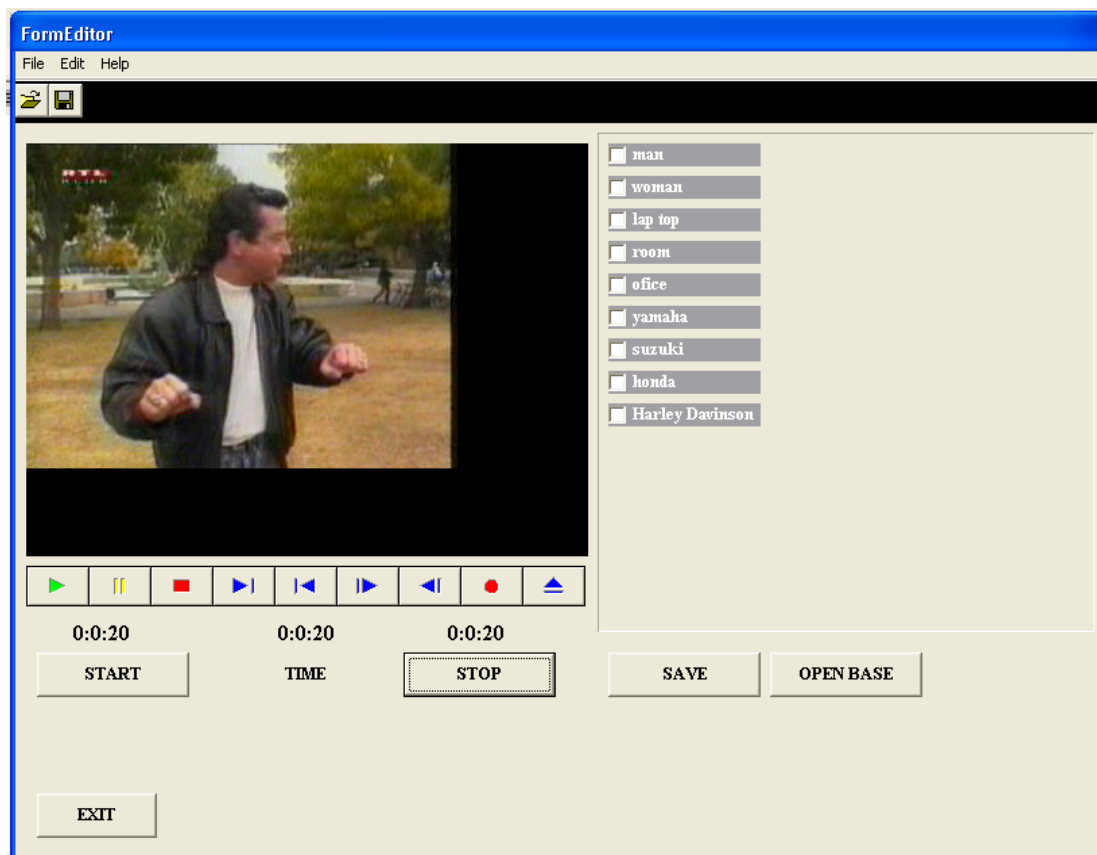
Αφού ολοκληρωθεί η προβολή όλων των κομματιών του συγκεκριμένου video που ήταν καταχωρημένο στο συγκεκριμένο πεδίο της Βάσης Δεδομένων, μπορούμε να συνεχίσουμε κανονικά την προβολή επιλέγοντας κάποιο άλλο CheckBox και ακολουθούμε την ίδια διαδικασία.

Στη συνέχεια θα παρουσιαστούν συγκεκριμένα κομμάτια του παραδείγματος που δείχνουν το άνοιγμα, την επεξεργασία και την προβολή δύο video.

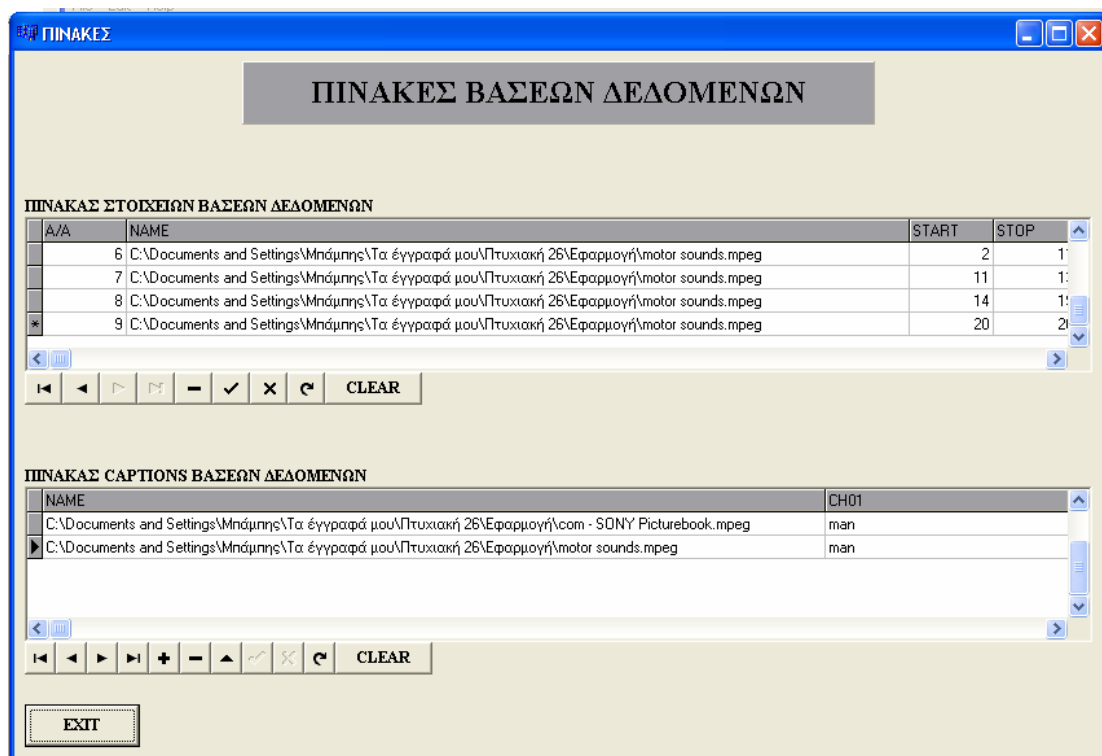
Έτσι λοιπόν, όπως φαίνεται και στην Εικόνα 31, έχει ανοιχθεί το funny video με τους ήχους γνωστών μηχανών, αλλά και ένα video που προβάλλει μία διαφήμιση της SONY, όπως και πριν. Το καθένα έχει τα δικά του πεδία, τα οποία καθώς επιλέγουμε την κάθε ταινία ξεχωριστά εμφανίζονται στο πλαίσιο και τα ανάλογα πεδία.

Γίνεται λοιπόν αρχικά η προβολή του πρώτου video, γίνονται οι καταχωρίσεις που πρέπει και έπειτα ακολουθείτε η ίδια διαδικασία και με το δεύτερο video χωρίς βέβαια να αποκλειστεί το ενδεχόμενο ότι μπορεί το ένα video να χρησιμοποιήσει πεδία της Βάσης Δεδομένων του άλλου εφόσον περιέχει στοιχεία που ταιριάζουν με το περιεχόμενο του.

Στη Εικόνα 32, μπορεί κανείς να δει πως “συνυπάρχουν” τα δύο video σε μία κοινή Βάση Δεδομένων, όπου και εδώ ισχύουν τα ίδια για τα πεδία του καθενός δηλαδή παίρνουν τις τιμές true και false όπως και παραπάνω.

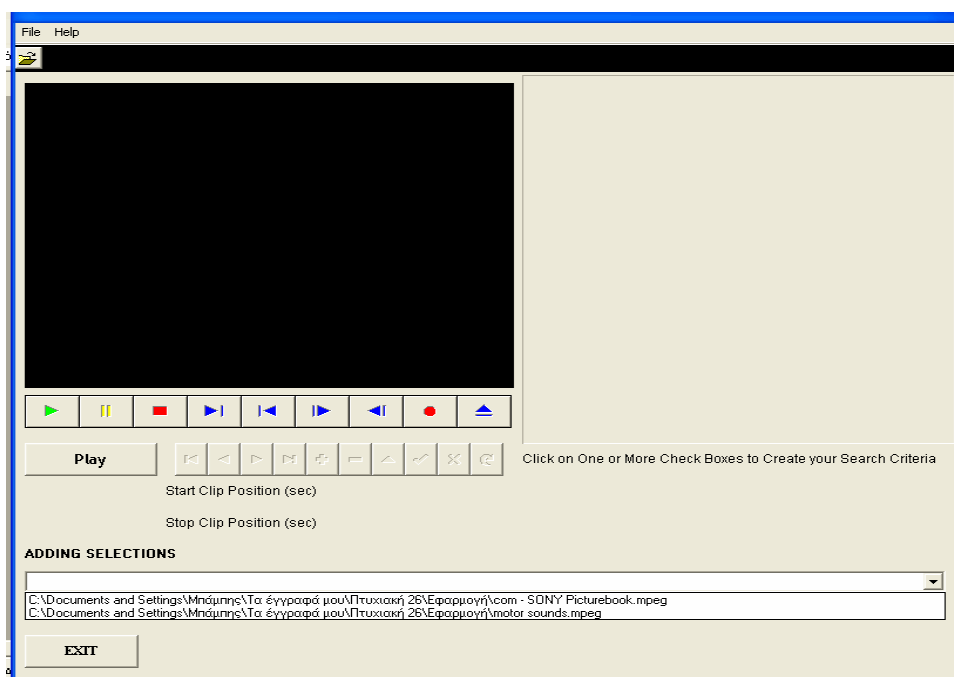


Εικόνα 32



Εικόνα 33

Αν ο χρήστης ακολουθήσει την παραπάνω διαδικασία και οδηγηθεί στο menu – file – open της Form Screening τότε διαπιστώσει πως θα ανοίξουν πράγματι και τα δύο video που έχουν χρησιμοποιηθεί. Για την επιλογή του καθ' ενός κλικάρει στο όνομα του κάθε video ξεχωριστά και εμφανίζονται τα ανάλογα πεδία.

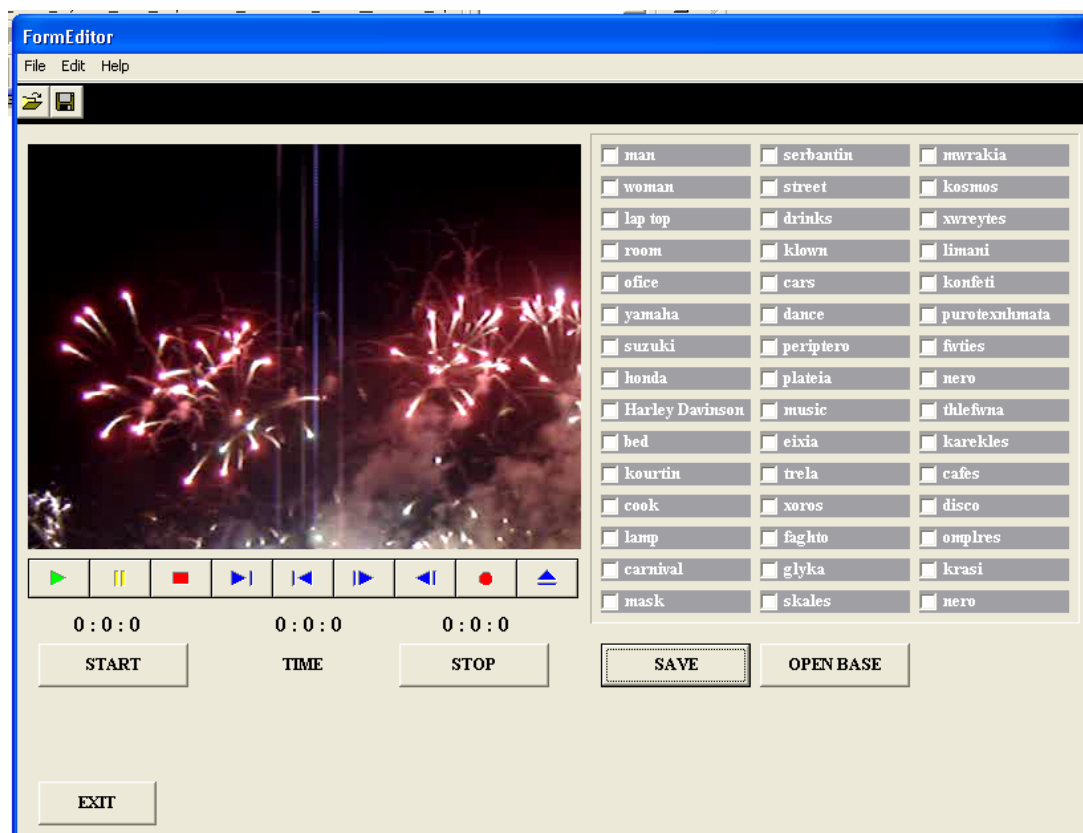


Εικόνα 34

Στην παραπάνω εικόνα, Εικόνα 34, φαίνονται ότι έχουν ανοιχθεί και επιλεγεί και τα δύο video που έχουν χρησιμοποιηθεί κατά την πρώτη φάση.

Κατά άλλα η διαδικασία παραμένει η ίδια όπως και με το ένα video όπως παραπάνω.

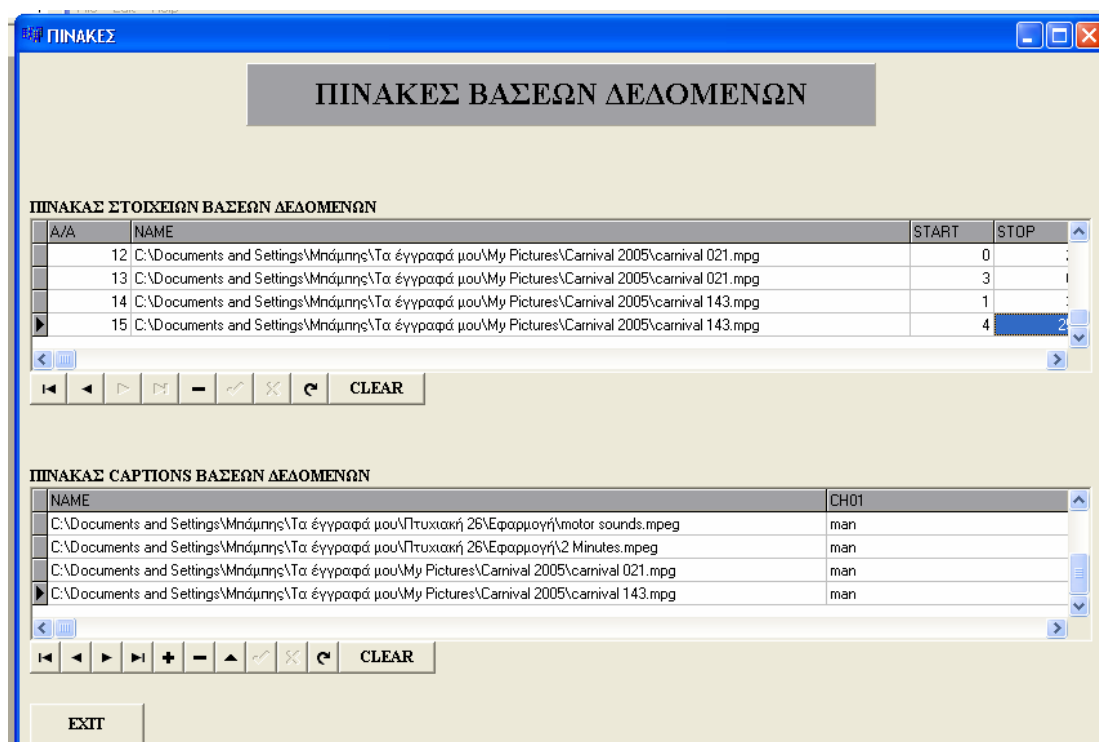
Τέλος, θα παρουσιαστεί και η περίπτωση που έχουν ανοιχθεί για παράδειγμα δέκα πέντε videos και έχουν γίνει τριάντα οκτώ καταχωρήσεις όπως παρακάτω στην Εικόνα 35.



Εικόνα 35

Μέχρι αυτό το σημείο όπως φαίνεται και από την εικόνα 34, έχουν γίνει όλες οι καταχωρήσεις και μάλιστα έχουν χρησιμοποιηθεί και τα σαράντα πέντε διαθέσιμα πεδία χωρίς κανένα απολύτως πρόβλημα.

Στην Εικόνα 36 φαίνεται ο πίνακας με τις καταχωρήσεις αλλά και τα χρησιμοποιούμενα videos.



Εικόνα 36

ΣΥΜΠΕΡΑΣΜΑΤΑ ΠΑΡΑΔΕΙΓΜΑΤΟΣ.

Απ' όλα τα παραπάνω μπορεί να συμπεράνει κανείς ότι η όλη εφαρμογή λειτουργεί σε ικανοποιητικό βαθμό

Τα πλεονεκτήματα που πηγάζουν από τα παραπάνω είναι ότι σε περίπτωση αστοχίας του συστήματος η Βάση Δεδομένων δε χάνει κανένα από τα αποθηκευμένα της στοιχεία και μπορούν να ανακτηθούν στη συνέχεια.

ΔΟΚΙΜΕΣ

Στη φάση αυτή έγινε ο έλεγχος επίδοσης της εφαρμογής που δημιουργήθηκε. Σκοπός του ελέγχου είναι να αποδείξει ότι το πρωτότυπο πληρεί τις προδιαγραφές που τέθηκαν κατά τη σχεδίασή του. Τα σημεία που ελέγχθηκαν είναι:

1. Ικανοποίηση των λειτουργικών απαιτήσεων από το σύστημα.

Το σύστημα σε γενικές γραμμές φάνηκε να ικανοποιεί τους χρήστες που το δοκίμασαν και πληρεί τις προδιαγραφές που τέθηκαν. Ένας απλός χρήστης μπορεί εύκολα και απλά να δημιουργήσει τα δικά του «έργα» να

ενσωματώσει πολλά βίντεο, να τα ομαδοποιήσει πατώντας απλά κλικ, και στο τέλος να συρράψει ομαδοποιημένες σκηνές και να τις προβάλλει είτε μαζικά είτε μεμονωμένα. Του δίνονται δυνατότητες να μελετήσει ομαδοποιημένες ακολουθίες βίντεο με κοινά χαρακτηριστικά και να βγάλει δικά του συμπεράσματα. Οι δυνατότητες αυτές δεν του δινόταν με τις λειτουργίες που παρέχουν είτε οι συσκευές βίντεο, είτε τα PVR είτε τα Media Center PC, τα οποία αν και δίνουν πολλές δυνατότητες στον χρήστη, στο συγκεκριμένο τομέα υστερούν.

2. Ασφάλεια του συστήματος.

Η ασφάλεια του συστήματος αφορά τον τρόπο πρόσβασης στην εφαρμογή. Η βάση δεδομένων paradox που χρησιμοποιήθηκε παρέχει μεγάλη ασφάλεια όσον αφορά την πρόσβαση χρηστών στην εφαρμογή. Αυτό επιτυγχάνεται με την δήλωση των χρηστών που πρόκειται να έχουν πρόσβαση και η αποθήκευση των στοιχείων user id, password. Στην εφαρμογή μπορούν να έχουν πρόσβαση χρήστες είτε μέσω ελέγχου από το λειτουργικό σύστημα, είτε μέσω του DBMS. Στην εφαρμογή αυτή επιλέχθηκε για συντομία η πρόσβαση να γίνεται απλά, και όχι μέσω της βάσης δεδομένων αλλά μέσω του λειτουργικού συστήματος.

3. Αξιοπιστία του συστήματος.

Το σύστημα σε γενικές γραμμές είναι ένα αξιόπιστο σύστημα το οποίο μπορεί να χρησιμοποιηθεί για πολλούς σκοπούς. Είτε επαγγελματικά είτε για διασκέδαση. Η οργάνωση των δεδομένων, η ταξινόμηση των στοιχείων που καταγράφονται στις βάσεις, η ασφάλεια και η ακεραιότητα των δεδομένων που είναι αποθηκευμένα στις σχεσιακές βάσεις και δεν μπορεί εύκολα κάποιος να επέμβει και να τα αλλοιώσει η να τα καταστρέψει, δίνουν μια σιγουριά στον χρήστη ότι δεν θα χάσει τον κόπο της δουλειάς του με την εφαρμογή αυτή, δίνοντας έτσι στο σύστημά μας μια μεγάλη αξιοπιστία.

Η αποκατάσταση της εφαρμογής μετά από αιφνίδιο σταμάτημα ήταν ομαλή και δεν παρατηρήθηκαν απώλειες δεδομένων. Μετά από κάθε αιφνίδια εκκίνηση του Η/Υ η εφαρμογή επανερχόταν στην φυσιολογική της κατάσταση

και τα δεδομένα ήταν καταγεγραμμένα στις βάσεις δεδομένων.

Αυτό οφείλεται στο γεγονός ότι τα συστήματα διαχείρισης βάσεων δεδομένων όπως η *paradox* που χρησιμοποιείται στην εφαρμογή, χρησιμοποιούν στις συναλλαγές τους το πρωτόκολλο επικύρωσης δύο φάσεων. Η επικύρωση δύο φάσεων είναι σημαντική όταν μια δεδομένη συναλλαγή μπορεί να αλληλεπιδρά με πολλούς ανεξάρτητους "διαχειριστές πόρων", που ο καθένας διαχειρίζεται το δικό του σύνολο ανακάμψιμων πόρων (*recoverable resources*) και συντηρεί το δικό του αρχείο πεπραγμένων για την ανάκαμψη

Στην πρώτη φάση γίνεται καταναγκαστική καταγραφή όλων των καταχωρίσεων πεπραγμένων της συναλλαγής σε μόνιμο μέσο αποθήκευσης.

Ότι και αν συμβεί μετά, ο διαχειριστής πόρων θα έχει μια μόνιμη εγγραφή της εργασίας που έκανε για λογαριασμό της συναλλαγής, και έτσι θα μπορεί να επικυρώσει τις ενημερώσεις της ή να τις ανασκευάσει, ανάλογα με την περίπτωση

Στην δεύτερη φάση γίνεται η επικύρωση ή ανασκευή τη συναλλαγής τοπικά, ανάλογα με την τελική υπόδειξη (*commit, abort*)

4. Ομαλή συμπεριφορά του συστήματος στα όρια των προδιαγραφών του (έλεγχος οριακών καταστάσεων).

Παρατηρήθηκε ομαλή συμπεριφορά του συστήματος σε στιγμιαίο φόρτο (έλεγχος πίεσης – *stress test*). Ο έλεγχος αυτός γίνεται την στιγμή που η εφαρμογή μεταβαίνει από την προβολή μιας χρονικής ακολουθίας βίντεο στην επόμενη ακολουθία. Αυτό δοκιμάστηκε με δύο τρόπους. α) Πρώτον όταν η διαδοχική χρονική ακολουθία βίντεο ανήκει στο ίδιο αρχείο βίντεο αλλά σε άλλη χρονική στιγμή και β) Όταν η επόμενη χρονική ακολουθία βίντεο ανήκει σε διαφορετικό αρχείο βίντεο. Και στις δύο περιπτώσεις η απόκριση της εφαρμογής ήταν άμεση και η μετάβαση στις επιθυμητές σκηνές στιγμιαία.

Παρατηρήθηκε ότι σε αυτή την δοκιμή μεγάλο ρόλο παίζει η μνήμη *ram* του Η/Υ. Όσο περισσότερη μνήμη διαθέτει ο Η/Υ τόσο ταχύτερη είναι η απόκριση του συστήματος. Σε δοκιμές με μνήμη Η/Υ 256MB και 512MB

παρατηρήθηκαν διαφορές απόκρισης μέχρι και 10 φορές γρηγορότερα στην περίπτωση με μνήμη 512MB.

5. Ομαλή συμπεριφορά του συστήματος σε μεγάλο όγκο δεδομένων (έλεγχος όγκου δεδομένων – volume test).

Η εφαρμογή έχει δοκιμαστεί να δουλεύει με μεγάλο όγκο βιντεοδεδομένων και έδειξε ότι η συμπεριφορά της είναι σταθερή. Αυτό οφείλεται στο γεγονός ότι η εφαρμογή προβάλλει πάντα ένα βίντεο την φορά και όταν πρόκειται να προβάλλει το επόμενο βίντεο, πρώτα κλείνει το προηγούμενο βίντεο και μετά ανοίγει το επόμενο. Το γεγονός αυτό δίνει μια σταθερότητα στο σύστημα και μία ανεξαρτησία από το πλήθος των βίντεο που περιλαμβάνει το κάθε έργο. Επίσης ο όγκος δεδομένων που μπορεί να αποθηκευτεί στις βάσεις είναι θεωρητικά άπειρος χωρίς να επηρεάζει την απόδοση της εφαρμογής. Αυτός ήταν και ένας από τους λόγους που μας οδήγησαν να χρησιμοποιήσουμε βάσεις δεδομένων και όχι απλά επίπεδα αρχεία για την αποθήκευση των ομαδοποιήσεων των «έργων»

Κάθε μια από τις περιπτώσεις αυτές θα ελέγχθηκε διεξοδικά προκειμένου να αναγνωριστούν αποκλίσεις από την προδιαγεγραμμένη και σχεδιασμένη λειτουργία. Ειδικότερα θα πραγματοποιήθηκε έλεγχος σχετικά με τη λειτουργία όλων των οθονών που συνιστούν διεπαφές χρήστη. Προτείνεται η χρήση της εφαρμογής σε οθόνες 15” και άνω με ανάλυση 1024 X 768 pixel και ανώτερη όσο μεγαλύτερη είναι η οθόνη

Ιδιαίτερη έμφαση δόθηκε στον έλεγχο και τη διαχείριση των συνδέσεων με τις βάσεις δεδομένων καθώς οι ταχύτητες πρόσβασης και επικοινωνίας με τη βάση δεδομένων συνιστούν τον καθοριστικότερο παράγοντα για την τελική ταχύτητα του συστήματος διαχείρισης ψηφιακού βίντεο. Παρατηρήθηκε ότι το πρωτόκολλο επικοινωνίας με την βάση paradox ήταν τάχιστο μιας και δεν δουλεύει ξεχωριστός server δηλ. δεν εκτελείται ως ξέχωρη διεργασία. Η εκτέλεση των ερωτημάτων SQL που εκτελούνται στο παρασκήνιο σε κάθε αναζήτηση είναι ταχύτερη και ανεξάρτητη από την μνήμη του Η/Υ.

Ακόμα μελετήθηκε η ταχύτητα διεκπεραίωσης συγκεκριμένων διαδικασιών προκειμένου να εντοπιστούν σημεία του κώδικα που επιδέχονται

βελτίωση είτε σε αλγοριθμικό επίπεδο, είτε σε επίπεδο τεχνικών και τεχνολογιών κωδικοποίησης. Συγκεκριμένα θα μπορούσαν να βελτιωθούν τα σημεία του κώδικα που αφορούν τις συναρτήσεις προβολής του video προσθέτοντας επιπλέον δυνατότητες όπως zoom, εμπρός και πίσω αργή κίνηση κλπ.

Επίσης θα μπορούσαν να αναπτυχθούν αλγόριθμοι αναγνώρισης πρότυπων για ημι-αυτόματο σχολιασμό και αποθήκευση στην πληροφοριών στην βάση δεδομένων.

Η τελική εφαρμογή πιστεύουμε ότι είναι ένα ολοκληρωμένο λειτουργικό πρωτότυπο όπου αναφέραμε τα συμπεράσματα που προέκυψαν από την αποτίμηση των αποτελεσμάτων, καθώς και τις προτεινόμενες βελτιώσεις της, μετά τη λήξη της πτυχιακής.

7. ΣΥΜΠΕΡΑΣΜΑΤΑ-ΧΡΗΣΕΙΣ.

Τελειώνοντας την παρούσα εργασία και εφαρμόζοντας όλα τα παραπάνω, μπορεί κανείς να καταλάβει την σπουδαιότητα και τη χρησιμότητα γενικά των Βάσεων Δεδομένων αλλά και πιο συγκεκριμένα των Ψηφιακών Video Βάσεων Δεδομένων.

Οι Βάσεις Δεδομένων λοιπόν, ύστερα απ' όλα τα παραπάνω, μπορούμε να συμπεράνουμε ότι είναι πολύ χρήσιμες και σημαντικές λόγω της ακεραιότητάς τους, των δυνατοτήτων αποθήκευσης του τεράστιου όγκου των πληροφοριών, αλλά και της μονιμότητας των πληροφοριών αυτών αφού είναι αποθηκευμένες μέσα σε πίνακες οι οποίες παρέχουν και την ανάλογη ασφάλεια.

Επίσης, μπορούμε να διαπιστώσουμε ότι το περιβάλλον χρήσης των video Βάσεων Δεδομένων είναι αρκετά εύκολο και γρήγορο, γεγονός που καθιστά τις πρώτες ιδιαίτερα δημοφιλείς.

Τέλος, η χειρονακτική διαδικασία σχολιασμού προσφέρει πέρα από την καταγραφή των στοιχείων και την ανάλογη ομαδοποίηση αυτών και έτσι η εφαρμογή μας γίνεται πιο εύχρηστη, λειτουργική και φιλική προς τον χρήστη.

Είναι μία φορητή βίντεο βάση δεδομένων.

Λειτουργεί αποτελεσματικά ανεξάρτητα του όγκου δεδομένων των βίντεο που αναλύει.

Ένα από τα δυνατά μέρη της εφαρμογής μας είναι ότι είναι παραμετρική όσον αφορά τις ετικέτες που δίνει και τις ομαδοποιήσεις που μπορεί να κάνει ένας χρήστης. Μπορούν να διαφέρουν οι ετικέτες και οι ομαδοποιήσεις από βίντεο σε βίντεο χωρίς να χάνεται η πληροφορία αυτή.

Υπάρχει άριστη διεπαφή με την βάση δεδομένων και άμεση εκτέλεση των δυναμικών ερωτημάτων που δημιουργούνται από τους χρήστες.

Ανταποκρίνεται άμεσα είτε στιγμιαίο φόρτο όταν προβάλλονται διαδοχικές ακολουθίες του ίδιου βίντεο, είτε όταν φορτώνεται διαφορετικό αρχείο βίντεο.

Το παρόν πρόγραμμα μπορεί να βρει εφαρμογή σε πάρα πολλούς τομείς αφού οι υπηρεσίες που μπορεί να παρέχει είναι απεριόριστες. Μερικές από τις εφαρμογές της είναι:

- **ΤΗΛΕΟΠΤΙΚΟΙ ΣΤΑΘΜΟΙ** όπου μπορούν να καταχωρούν σκηνές από τα σίριαλ που προβάλλονται με τους πρωταγωνιστές τους ή σκηνές πολλών σίριαλ μαζί που μπορούν να έχουν κοινά στοιχεία. Ακόμα, στα δελτία ειδήσεων είναι πολύ χρήσιμη αφού μπορεί να καταχωρεί για παράδειγμα ρεπορτάζ για το ίδιο θέμα σε μία κοινή Βάση Δεδομένων.
- **ΑΘΛΗΤΙΚΕΣ ΟΜΑΔΕΣ** οι οποίες καταχωρώντας στη παρούσα Βάση Δεδομένων δικούς τους αγώνες ή στιγμιότυπα άλλων αγώνων να βελτιώσουν τις μεθόδους προπόνησης τους προβάλλοντας με ευκολία και ταχύτητα μόνα εκείνα τα κομμάτια που ενδιαφέρουν τους παραπάνω.
- **ΟΙΚΙΑΚΗ ΧΡΗΣΗ** αφού είναι τόσο απλή στη χρήση της. Πάρα πολλές οικογένειες είναι αυτές που έχουν camera στο σπίτι και αναγκάζονται να γράφουν πάνω στο ίδιο memory stick ενδεχομένως ή να “περνούν” στον υπολογιστή τα videos και στη συνέχεια να ψάχνουν την επιθυμητή σκηνή. Με τη παρούσα εφαρμογή μπορούν καταχωρηθούν τα videos όπως επιθυμεί ο καθένας και απλά να τα ανακτεί μέσα από τη Βάση Δεδομένων.
- **ΔΙΑΦΗΜΗΣΤΙΚΕΣ ΕΤΑΙΡΙΕΣ** οι οποίες έχουν τη δυνατότητα να καταχωρούν στους πίνακες της Βάσης τα διαφημιστικά clips των προϊόντων τους ώστε να γίνεται γρήγορα η εύρεση τους ανά πάσα στιγμή.
- **ΚΙΝΗΤΗ ΤΗΛΕΦΩΝΙΑ** με τα καινούργια κινητά τηλέφωνα τρίτης γενιάς η αναπαραγωγή videos είναι ένα από τα κύρια χαρακτηριστικά τους. Αφού υπάρχει η δυνατότητα σύνδεσης με τον υπολογιστή τότε τα videos των κινητών τηλεφώνων μπορούν να καταχωρηθούν με ευκολία σε μία τέτοια Βάση Δεδομένων.
- **ΕΤΑΙΡΙΕΣ ΑΣΦΑΛΕΙΑΣ** και κατ’ επέκταση στις επιχειρήσεις που προστατεύουν, αφού με το παρόν πρόγραμμα έχουν τη δυνατότητα να αποθηκεύουν και να έχουν πρόσβαση άμεσα σε όλα τα videos για λόγους ασφαλείας.

Όλα τα παραπάνω είναι μόνο ελάχιστες περιπτώσεις στις οποίες μπορεί να βρει εφαρμογή η παρούσα εργασία. Η σταθερότητα και η ταχύτητα είναι τα κύρια χαρακτηριστικά που μπορούν να βοηθήσουν το κάθε χρήστη ώστε να τη χρησιμοποιήσει.

10. ΒΙΒΛΙΟΓΡΑΦΙΑ – ΠΗΓΕΣ

1. Εισαγωγή στα Συστήματα Βάσεων Δεδομένων (1^{ος} κ 2^{ος} τόμος), C. J. Date, έκτη αμερικάνικη έκδοση 1995, Κλειδάριθμος.
2. Θεμελιώδεις αρχές συστημάτων Βάσεων Δεδομένων (1^{ος} τόμος), Ramez Elmarī και Sham B. Navathe, 3^η Έκδοση αναθεωρημένη, Δίαυλος.
3. Θεμελιώδεις αρχές συστημάτων Βάσεων Δεδομένων (2^{ος} τόμος), Ramez Elmarī και Sham B. Navathe, 3^η Έκδοση αναθεωρημένη, Δίαυλος.
4. Εισαγωγή στις Βάσεις Δεδομένων, αναπτυξιακή προσέγγιση, Ξένος, Χριστοδουλάκης, 2002, Παπασωτηρίου.
5. Σχεδιασμός Βάσεων Δεδομένων & dBase IV/SQL, Βασιλακόπουλος, 1993, Σταμούλης
6. Προγραμματισμός στον MS SQL SERVER 2000 Βήμα προς Βήμα, Riordan, 2002, Κλειδάριθμος
7. MS SQL SERVER 2000 analysis Services Βήμα Βήμα, Jacobson, 2002, Κλειδάριθμος
8. DataBase Programming & Design, Database Review, InfoDB, DataBase Newsletter.
9. 'Εισαγωγή στον Πληροφορική', Peter Rechenberg, Κλειδάριθμος.
10. 'Πλήρης Οδηγός Χρήσης της C', Ομάδα Waite, Γκιούρδας.
11. 'Δομημένος Προγραμματισμός', Π. Μαστοροκώστας, ΤΕΙ ΣΕΡΡΩΝ.
12. 'Αντικειμενοστραφής Προγραμματισμός με την Turbo C++', Ομάδα Waite, Κλειδάριθμος.
13. Borland C++ Builder 6, Πλήρες Εγχειρίδιο, Hollingwirth, Γκιούρδας.

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

ΕΙΚΟΝΑ 01: περιληπτική παρουσίαση της πληροφορίας για την παράδοση στους κινητούς χρήστες.....	ΣΕΛ 12
ΕΙΚΟΝΑ 02: φάσεις σχεδιασμού Μοντέλων.....	ΣΕΛ 18
ΕΙΚΟΝΑ 03: διάγραμμα του μοντέλου οντοτήτων συσχετίσεων ER.....	ΣΕΛ 21
ΕΙΚΟΝΑ 04: πίνακας συσχετίσεων.....	ΣΕΛ 24
ΕΙΚΟΝΑ 05: μετατροπή ενός διαγράμματος ER σε σχεσιακό μοντέλο.....	ΣΕΛ 26
ΕΙΚΟΝΑ 06: ροή των πληροφοριών μεταξύ του χρήστη και της Βάσης Δεδομένων.....	ΣΕΛ 28
ΕΙΚΟΝΑ 07: βήματα για τον ορισμό Δεδομένων και η κατασκευή της Βάσης Δεδομένων	ΣΕΛ 29
ΕΙΚΟΝΑ 08: βήματα για τον ορισμό Δεδομένων και η κατασκευή της Βάσης Δεδομένων	ΣΕΛ 30
ΕΙΚΟΝΑ 09: BDE Administrator.....	ΣΕΛ 36
ΕΙΚΟΝΑ 10: εξάρτημα Ttable	ΣΕΛ 37
ΕΙΚΟΝΑ 11: εξάρτημα TDataSource	ΣΕΛ 38
ΕΙΚΟΝΑ 12: ιεραρχία κλάσεων για τα συστατικά ADO	ΣΕΛ 43
ΕΙΚΟΝΑ 13: σχέσεις μεταξύ ομάδων συστατικών	ΣΕΛ 44
ΕΙΚΟΝΑ 14: Α΄ ΦΑΣΗ ΚΑΤΑΓΡΑΦΗΣ	ΣΕΛ 46
ΕΙΚΟΝΑ 15: Β΄ ΦΑΣΗ ΚΑΤΑΓΡΑΦΗΣ	ΣΕΛ 47
ΕΙΚΟΝΑ 16: FormStart.....	ΣΕΛ 48
ΕΙΚΟΝΑ 17: FormEditor	ΣΕΛ 50
ΕΙΚΟΝΑ 18: FormEditor (menu)	ΣΕΛ 51
ΕΙΚΟΝΑ 19: FormEditor (Μήνυμα Οδηγίας)	ΣΕΛ 53
ΕΙΚΟΝΑ 20: CheckboxTable	ΣΕΛ 55
ΕΙΚΟΝΑ 21: φόρμα Πίνακες	ΣΕΛ 62
ΕΙΚΟΝΑ 22: menu – Help	ΣΕΛ 64
ΕΙΚΟΝΑ 23: Help Topics	ΣΕΛ 65
ΕΙΚΟΝΑ 24: FormScreening	ΣΕΛ 67
ΕΙΚΟΝΑ 25: Creators	ΣΕΛ 72
ΕΙΚΟΝΑ 26: FormEditor (Παράδειγμα)	ΣΕΛ 73
ΕΙΚΟΝΑ 27: FormEditor (Παράδειγμα)	ΣΕΛ 74

ΕΙΚΟΝΑ 28: φόρμα Πίνακες (Παράδειγμα)	ΣΕΛ 75
ΕΙΚΟΝΑ 29: φόρμα Πίνακες (Παράδειγμα)	ΣΕΛ 75
ΕΙΚΟΝΑ 30: FormScreening (Παράδειγμα)	ΣΕΛ 76
ΕΙΚΟΝΑ 31: FormScreening (Παράδειγμα)	ΣΕΛ 77
ΕΙΚΟΝΑ 32: FormScreening (Παράδειγμα)	ΣΕΛ 78
ΕΙΚΟΝΑ 33: φόρμα Πίνακες (Παράδειγμα)	ΣΕΛ 79
ΕΙΚΟΝΑ 34: FormScreening (Παράδειγμα)	ΣΕΛ 80
ΕΙΚΟΝΑ 35: FormEditor (Παράδειγμα)	ΣΕΛ 81
ΕΙΚΟΝΑ 36: φόρμα Πίνακες (Παράδειγμα)	ΣΕΛ 82

ΕΥΧΑΡΙΣΤΙΕΣ

Τελειώνοντας την εργασία αυτή, τελειώνει και τυπικά και ιδιότητα μου, έστω και προσωρινά, ως ακαδημαϊκός πολίτης. Η μόρφωση και η αναζήτηση πληροφορίας βέβαια δε σταματάει ποτέ.

Με την λήξη λοιπόν αυτού του κύκλου, μου δίνεται η ευκαιρία να ευχαριστήσω κάποιους ανθρώπους που είναι “υπεύθυνοι” για το σημείο στο οποίο κατάφερα να φτάσω και που ποτέ δεν έτυχε να ευχαριστήσω.

Θα ήταν ατόπημα να μην ξεκινήσω τις ευχαριστίες μου από τους καθηγητές μου τόσο του Λυκείου Θέρμης όσο και του φροντιστηρίου μου οι οποίοι μου πρόσφεραν τις γνώσεις και τα απαραίτητα εφόδια για να καταλήξω τον Οκτώβριο του 1999 στις Σέρρες και στο Τμήμα Πληροφορικής & Επικοινωνιών του ΤΕΙ.

Οι καθηγητές του ΤΕΙ μας πρόσφεραν ένα κλίμα που νομίζω όμοιο του δεν υπάρχει, με αποτέλεσμα να δημιουργήσουν τις κατάλληλες συνθήκες τόσο για την μετάδοση των γνώσεων όσο και για την δημιουργία ενός φιλικότατου κλίματος.

Εδώ μου προσέφεραν δουλειά, μου έδωσαν εμπιστοσύνη και το εκτιμώ πολύ.

Θα ‘θελα να ευχαριστήσω ιδιαίτερα την κα Θεοδωρίδου η οποία με βοήθησε πάρα πολύ και αναγνωρίζω τις πράξεις που έκανε για μένα.

Τον κο Τσιμπίρη, υπεύθυνο καθηγητή της πτυχιακής εργασίας, που με η πολύτιμη βοήθεια του και η “πίεση” του προς εμένα, ειδικά προς το τέλος, ήταν παράγοντες που συντέλεσαν στο να έχω φτάσει στην επιθυμητό αποτέλεσμα.

Τέλος, οι ευχαριστίες προς την Οικογένεια μου είναι πάρα πολλές, διότι με την ηθική και οικονομική τους βοήθεια, συντέλεσαν στην επιτυχή και ομαλή παραμονή μου στη πόλη των Σερρών και την απόκτηση του πολυπόθητου πτυχίου. Ευχαριστώ που ανέχτηκαν τις ιδιοτροπίες μου και τα λάθη μου, αφιερώνω το πτυχίο μου στον πατέρα μου, τη μητέρα μου και τον αδελφό μου, και τους δίνω τη υπόσχεση και τη διαβεβαίωση ότι... τα καλύτερα έπονται...