



Τ.Ε.Ι. ΗΠΕΙΡΟΥ

ΣΧΟΛΗ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ

ΤΜΗΜΑ ΤΗΛΕΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

**ΑΝΑΠΤΥΞΗ  
ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ  
ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΣΕ  
VISUAL C++ 6**

*Αναστασιάδης Σταύρος*

Εισηγητής: Κονετός Δημήτρης

---

Οκτώβριος 2003

**Θέμα:**

**« ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΔΙΑΧΕΙΡΙΣΗΣ  
ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΣΕ VISUAL C++ 6 »**

**Αναστασιάδης Σταύρος**

**Οκτώβριος 2003**

**Πτυχιακή εργασία μέρος των απαιτήσεων  
του τμήματος Τηλεπληροφορικής και Διοίκησης**

## **Ευχαριστώ**

**Τον καθηγητή μου Δημήτρη Κονετά για τις πολύτιμες συμβουλές του,  
Τους φίλους μου για το ενδιαφέρον τους,  
Τους γονείς μου για την υπομονή τους...**

## ABSTRACT

Η πληροφορία αποτελεί ιδιαίτερα σημαντικό πόρο στη σύγχρονη οικονομία. Οι επιχειρήσεις ανέκαθεν αναζητούν μεθόδους με τις οποίες επιδιώκουν βελτίωση της ανταγωνιστικότητάς τους. Οι αναβαθμίσεις των προϊόντων και η οικονομική βελτιστοποίηση αποτελούσαν και αποτελούν μέχρι σήμερα βασικούς παράγοντες που συντελούν προς αυτήν την κατεύθυνση. Οι σύγχρονες θεωρίες του management και του marketing υπαγορεύουν τη χρήση μεθόδων που βασίζονται στην πληροφορία για την περαιτέρω βελτίωση της λειτουργίας και της ανταγωνιστικότητας των επιχειρήσεων. Η πληροφορία ως πόρος πλέον χρησιμοποιείται σε όλα τα επίπεδα της διοικητικής ιεραρχίας. Από την έρευνα για το σχεδιασμό ενός νέου προϊόντος έως και την διαδικασία διανομής του, και από τις διαδικασίες παραγωγής έως και τις διοικητικές αποφάσεις, στη σύγχρονη επιχείρηση λειτουργεί ένα παράλληλο σύστημα ροής και επεξεργασίας πληροφοριών.

Με την υιοθέτηση της πληροφορικής και των νέων τεχνολογιών, ως εργαλείο για την διαχείριση των πληροφοριών, οι επιχειρήσεις προχωρούν στην εγκαθίδρυση αρμόδιου τμήματος. Το τμήμα αυτό ευθύνεται για τη σωστή λειτουργία του πληροφοριακού συστήματος και για την συντήρησή του.

Η έννοια του πληροφοριακού συστήματος περιλαμβάνει την συνεργασία του ανθρώπινου δυναμικού με υπολογιστικά συστήματα για την επίτευξη λειτουργικών και διοικητικών στόχων. Οι ηλεκτρονικοί υπολογιστές χρησιμοποιούνται σε κάθε περίπτωση με ειδικά σχεδιασμένο και προσαρμοσμένο στις απαιτήσεις λογισμικό. Το λογισμικό έχει ζωτική συμμετοχή στο πληροφοριακό σύστημα εφ' όσον περιλαμβάνει την περιγραφή των διαδικασιών που πραγματοποιούνται από Ηλεκτρονικούς Υπολογιστές γενικού σκοπού σε συνεργασία με το αρμόδιο προσωπικό.

Οι σύγχρονες επιχειρήσεις έχουν την ανάγκη αποθήκευσης και επεξεργασίας μεγάλου όγκου δεδομένων που συγκεντρώνονται από το εσωτερικό και το εξωτερικό τους περιβάλλον, όπως επίσης και ανάγκη εύκολης πρόσβασης στις πληροφορίες που προκύπτουν. Η αποθήκευση του όγκου των δεδομένων αναλαμβάνεται από ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS – Data Base Management System). Ο ορισμός των δεδομένων (Data Definition) και η διαχείρισή τους (Data Manipulation) αναλαμβάνεται από ένα δυναμικό γραφικό περιβάλλον που συνεργάζεται με το σύστημα βάσης δεδομένων. Η προβολή των πληροφοριών πραγματοποιείται με τη χρήση της τεχνολογίας των δικτύων και συγκεκριμένα των δυνατοτήτων του Διαδικτύου (Internet).

Θα παρακολουθήσουμε τον κύκλο ζωής ενός πληροφοριακού συστήματος που σχεδιάστηκε έτσι ώστε να δώσει έναν μεγάλο συντελεστή ευελιξίας στους χρήστες του.

## ΠΕΡΙΕΧΟΜΕΝΑ

### ABSTRACT

<b>1. ΣΥΣΤΗΜΑΤΑ.....</b>	<b>8</b>
<b>2. ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ.....</b>	<b>9</b>
2.1 Εισαγωγή.....	9
2.2 Υπολογιστικά συστήματα.....	9
2.3 Πληροφοριακά συστήματα υποστήριξης αποφάσεων.....	10
2.4 Ο κύκλος ζωής των πληροφοριακών συστημάτων.....	10
<b>3. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ.....</b>	<b>12</b>
<b>4. ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ.....</b>	<b>14</b>
<b>5. ΦΥΣΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ.....</b>	<b>16</b>
<b>6. ΑΝΑΠΤΥΞΗ ΠΡΟΓΡΑΜΜΑΤΩΝ.....</b>	<b>17</b>
<b>7. ΥΛΟΠΟΙΗΣΗ.....</b>	<b>18</b>
<b>8. ΛΕΙΤΟΥΡΓΙΑ.....</b>	<b>19</b>
<b>9. ΑΡΧΙΤΕΚΤΟΝΙΚΗ CLIENT SERVER.....</b>	<b>20</b>
<b>10. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ.....</b>	<b>21</b>
10.1 Γενικά.....	21
10.2 Το σχεσιακό μοντέλο.....	22
<b>11. OLE DB.....</b>	<b>23</b>
11.1 Εισαγωγή.....	23
11.2 Καταναλωτές και παροχείς.....	23
11.3 Συστατικά μέρη συστήματος τεχνολογίας OLE DB.....	24
<b>12. ACTIVE SERVER PAGES.....</b>	<b>25</b>
12.1 Εισαγωγή.....	25
12.2 Δυνατότητες.....	25
12.3 Προγραμματισμός.....	25
<b>13. Η ΕΦΑΡΜΟΓΗ SUITECASE.....</b>	<b>26</b>
13.1 Εισαγωγή.....	26
13.2 Πρωτεύοντα εργαλεία.....	26
13.3 Βοηθητικά εργαλεία.....	27
<b>14. MASTEROLEDB.....</b>	<b>28</b>
14.1 Εισαγωγή.....	28
14.2 Πηγή δεδομένων (Datasource).....	29
14.3 Περιβάλλον αλληλεπίδρασης (Session).....	29

14.4 Εκτέλεση εντολών.....	29
14.5 Εκτέλεση εντολών ανάκτησης δεδομένων.....	30
14.6 Ανάκτηση δεδομένων δομής σχήματος.....	31
14.7 Μηνύματα σφαλμάτων.....	32
14.8 Βοηθητικές λειτουργίες.....	32
<b>15. ΤΟ ΕΡΓΑΛΕΙΟ DATAVIEW.....</b>	<b>34</b>
15.1 Εισαγωγή.....	34
15.2 Επιλογή βάσης δεδομένων.....	35
15.3 Επιλογή αντικειμένου.....	36
15.4 Εμφάνιση δεδομένων.....	37
15.5 Ταξινόμηση και φιλτράρισμα.....	37
15.6 Εισαγωγή, ενημέρωση και αντιγραφή γραμμής.....	39
15.7 Εισαγωγή / διαγραφή στήλης.....	40
15.8 Δημιουργία / διαγραφή δείκτη.....	41
15.9 Δημιουργία αναφορών.....	42
<b>16. ΤΟ ΕΡΓΑΛΕΙΟ SCRIPTEDIT.....</b>	<b>44</b>
16.1 Εισαγωγή.....	44
16.2 Επεξεργασία και εκτέλεση scripts.....	44
16.3 Η μπάρα εργαλείων.....	45
<b>17. ΤΟ ΕΡΓΑΛΕΙΟ QUERYWIZARD.....</b>	<b>47</b>
17.1 Εισαγωγή.....	47
17.2 Τα βήματα του οδηγού.....	47
<b>18. DDL ΚΑΙ DML.....</b>	<b>54</b>
18.1 DDL.....	54
18.1.1 Δημιουργία αντικειμένων.....	54
18.1.2 Διαγραφή αντικειμένων.....	54
18.2 DML.....	55
18.2.1 Εισαγωγή δεδομένων.....	55
18.2.2 Ενημέρωση δεδομένων.....	55
18.2.3 Αντιγραφή δεδομένων.....	57
<b>19. ΔΗΜΙΟΥΡΓΙΑ ΑΝΑΦΟΡΩΝ ΚΑΙ ΦΟΡΜΩΝ ΕΙΣΑΓΩΓΗΣ ΔΕΔΟΜΕΝΩΝ.....</b>	<b>58</b>
19.1 Αναφορές δεδομένων.....	58
19.2 Φόρμες εισαγωγής δεδομένων.....	59
<b>20. ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ SUITECASE.....</b>	<b>60</b>
<b>21. Η ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....</b>	<b>67</b>

---

21.1 ΣΧΕΔΙΑΣΜΟΣ.....	67
21.2 Η ΔΟΜΗ ΤΩΝ ΦΑΚΕΛΩΝ.....	67
21.3 ΑΝΑΠΤΥΞΗ ΒΙΒΛΙΟΘΗΚΩΝ ΔΥΝΑΜΙΚΩΝ ΣΥΝΔΕΣΕΩΝ.....	67
21.4 Η ΕΠΙΛΟΓΗ ΤΟΥ ΣΤΟΙΧΕΙΟΥ ΕΛΕΓΧΟΥ ΚΕΛΙΩΝ.....	68
21.5 Η ΕΠΙΛΟΓΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ OLE DB.....	69
21.6 Η ΜΕΤΑΤΡΟΠΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ.....	69
21.7 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ.....	70
<b>ΠΑΡΑΡΤΗΜΑ Α: ΣΤΟΙΧΕΙΟ ΕΛΕΓΧΟΥ ΚΕΛΙΩΝ.....</b>	<b>72</b>
<b>ΠΑΡΑΡΤΗΜΑ Β: ΜΕΤΑΤΡΟΠΗ ΔΕΔΟΜΕΝΩΝ.....</b>	<b>73</b>
<b>ΠΑΡΑΡΤΗΜΑ Γ: ΤΑ ΑΡΧΕΙΑ ΕΠΙΚΕΦΑΛΙΔΩΝ IMPORT.....</b>	<b>75</b>
<b>ΠΑΡΑΡΤΗΜΑ Δ: ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>78</b>

## 1. ΣΥΣΤΗΜΑΤΑ

Σύστημα ονομάζεται ένα σύνολο λειτουργικών μονάδων που αλληλεπιδρούν για την επίτευξη ενός ενιαίου στόχου, και αλληλεξαρτώνται για την επίτευξη των επιμέρους στόχων που αναλαμβάνει η κάθε μία. Ένα σύστημα χαρακτηρίζεται από την σωστή οργάνωσή του και από την ολοκλήρωση των διαδικασιών που αναλαμβάνει να διεκπεραιώσει ώστε να επιτευχθούν οι στόχοι του.[1]

Ένα σύστημα περιγράφεται από τα επιμέρους συστατικά του, από τις διαδικασίες που αυτά πραγματοποιούν και από την αλληλεπίδραση μεταξύ τους. Κάθε λειτουργική μονάδα, ως συστατικό μέρος ενός συστήματος, έχει καλά προσδιορισμένους επιμέρους στόχους και διαθέτει ένα ή περισσότερα σημεία συνεργασίας με τις υπόλοιπες ή με κάποιες από τις υπόλοιπες μονάδες του συστήματος. Η λειτουργική αλληλεπίδραση μεταξύ των μονάδων επικεντρώνεται στην ανταλλαγή εισερχόμενων και εξερχόμενων. Τα εξερχόμενα ενός συστατικού κάποιου συστήματος αποτελούν εισερχόμενα για κάποιο άλλο συστατικό του συστήματος. Σε ένα ευρύτερο πλαίσιο, ένα σύστημα μπορεί να θεωρηθεί ως μια οντότητα η οποία δέχεται κάποια εισερχόμενα και παράγει κάποια εξερχόμενα, με σκοπό την απόκτηση πόρων.

Τα σημεία συνεργασίας είναι στην ουσία αυτά που προσδιορίζουν την λειτουργικότητα του συστήματος. Ο σωστός σχεδιασμός τους διευκολύνει τις διαδικασίες της διοίκησης, του ελέγχου και της συντήρησης.

Για την ανάλυση ενός συστήματος μπορεί να ακολουθηθούν δύο διαφορετικές προσεγγίσεις:

- Από τους επιμέρους στόχους του συστήματος προς τους γενικότερους στόχους (bottom-up).
- Από τους γενικότερους στόχους του συστήματος προς τους επιμέρους στόχους (top-down).

Ανάλογα με τις ιδιαιτερότητες του συστήματος επιλέγεται μια από τις δύο προσεγγίσεις.

Η bottom-up προσέγγιση περιλαμβάνει αρχικά τον ακριβή προσδιορισμό και σαφή καθορισμό των επιμέρους στόχων του συστήματος, έτσι ώστε να αποφασιστεί η δομή των συστατικών λειτουργικών μονάδων και έπειτα να τους απονεμηθούν οι αρμοδιότητες. Στο επόμενο στάδιο καθορίζεται η αλληλεπίδραση μεταξύ των επιμέρους λειτουργικών μονάδων, προσδιορίζεται η ανταλλαγή εισερχόμενων και εξερχόμενων και τελικά σχηματίζεται ένα σύστημα που εξυπηρετεί έναν γενικότερα ωφέλιμο σκοπό.

Η top-down προσέγγιση αρχικά υπαγορεύει τον προσδιορισμό των γενικών στόχων του συστήματος. Έπειτα, οι γενικοί στόχοι διαχωρίζονται σε μικρότερα τμήματα τα οποία εκτελούν μια καλά καθορισμένη διαδικασία το κάθε ένα. Κάθε επιμέρους τμήμα, αρχικά, εκτελεί διαδικασίες οι οποίες έχουν άμεση σχέση με τα υπόλοιπα τμήματα που επικοινωνούν μαζί του. Μετά από μια περαιτέρω ανάλυση διασπώνται τα τμήματα κι άλλο, ώσπου κάθε επιμέρους υποσύστημα πλέον να είναι ανεξάρτητο από το περιβάλλον του.



## 2. ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ

### 2.1 ΕΙΣΑΓΩΓΗ

Τα πληροφοριακά συστήματα ορίζουν μια ειδική κατηγορία συστημάτων, τα στοιχεία των οποίων είναι άνθρωποι, διαδικασίες και μηχανήματα, που αλληλεπιδρούν και συνεργάζονται για να συλλέξουν, να αποθηκεύσουν και να επεξεργαστούν δεδομένα ώστε να παρέχουν πληροφορία στους χρήστες.

Πληροφοριακά συστήματα υπήρχαν πολύ πριν από την εμφάνιση των Ηλεκτρονικών υπολογιστών. Από την αρχαία εποχή, ακόμα, άτομα, επιχειρήσεις και κράτη χρησιμοποιούσαν πολύ στοιχειώδη και άτυπα πληροφοριακά συστήματα με το να συγκεντρώνουν και να επεξεργάζονται πληροφορίες. Με την πάροδο του χρόνου οι επιχειρήσεις και οι οργανισμοί άρχισαν να αναπτύσσουν χειρογραφικά πληροφοριακά συστήματα για την συλλογή, αποθήκευση, επεξεργασία και χρήση των πληροφοριών [2].

Σήμερα, ένα πληροφοριακό σύστημα ορίζεται ως μια αποθήκη δεδομένων καλά οργανωμένη ώστε να υποστηρίζει την εκτέλεση ενεργειών που απαιτεί από αυτήν ο χρήστης. Πέρα από την αποθήκη δεδομένων ένα πληροφοριακό σύστημα χρειάζεται συνήθως και κάποιον χειριστή ο οποίος εκτελεί τις δραστηριότητες που υποστηρίζονται. Ως βοηθητικό στοιχείο στις ενέργειες που εκτελεί ένας χειριστής κάποιου πληροφοριακού συστήματος χρησιμοποιείται και ένα υπολογιστικό σύστημα.

Ο χειριστής (ή οι χειριστές) του πληροφοριακού συστήματος αναλαμβάνει να απαντήσει στις ερωτήσεις των χρηστών εκτελώντας κάποιες σαφώς προσδιορισμένες διαδικασίες που υποστηρίζονται από το πληροφοριακό σύστημα. Οι χρήστες είναι αυτοί που αναθέτουν στους χειριστές μια εργασία και αυτοί που επωφελούνται τελικά από την λειτουργία του συστήματος και την εξαγωγή χρήσιμων πληροφοριών.

Τα υπολογιστικά συστήματα που ενσωματώνονται στα πληροφοριακά συστήματα αναλαμβάνουν την αυτοματοποίηση των εργασιών που εκτελεί ο χρήστης. Η αυτοματοποίηση ως χαρακτηριστικό των πληροφοριακών συστημάτων δεν αντικαθιστά την ανθρώπινη παρουσία, αλλά ενισχύει την αποδοτικότητα ενός χειριστή προσφέροντας πλεονεκτήματα, όπως η τεράστια ταχύτητα και μνήμη.

### 2.2 ΥΠΟΛΟΓΙΣΤΙΚΑ ΣΥΣΤΗΜΑΤΑ

Οι σύγχρονοι ηλεκτρονικοί υπολογιστές γενικού σκοπού συναντούνται σήμερα σχεδόν σε κάθε υπηρεσία, στην εφαρμογή πληροφοριακών συστημάτων διαφόρων ειδικοτήτων. Συναντούνται σε εφαρμογές διατήρησης αποθεμάτων μιας αποθήκης εμπορευμάτων, έως την διαχείριση ηλεκτρονικών παραγγελιών μέσω Internet (e-Commerce), και από εφαρμογές ανάλυσης στατιστικών στοιχείων έως προγράμματα συνεργασίας ιατρικών κέντρων και εκπαιδευτικών οργανισμών.

Η ποικιλομορφία και η προσαρμογή των πληροφοριακών συστημάτων στις εκάστοτε απαιτήσεις, όσον αφορά το υπολογιστικό σύστημα που χρησιμοποιούν, βασίζεται στο λογισμικό (software) που κατασκευάζεται ειδικά για κάθε περίπτωση. Υπάρχουν περιπτώσεις που χρησιμοποιείται και ειδικά διαμορφωμένο υλικό (hardware) ώστε να καλυφθούν οι απαιτήσεις ενός πληροφοριακού συστήματος (π.χ. ρομποτική σε γραμμές παραγωγής).

### 2.3 ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΥΠΟΣΤΗΡΙΞΗΣ ΑΠΟΦΑΣΕΩΝ

Τα πληροφοριακά συστήματα συχνά χρησιμοποιούνται και ως συστήματα υποστήριξης αποφάσεων. Σε αυτού του είδους τα πληροφοριακά συστήματα έχει ιδιαίτερη σημασία η επίλυση προβλημάτων που συνδέονται άμεσα με τις αποφάσεις του διοικητικού και του λειτουργικού επιπέδου της ιεραρχίας ενός οργανισμού. Τα είδη των προβλημάτων που καλείται να λύσει σε αυτές τις περιπτώσεις ένα πληροφοριακό σύστημα μπορεί να είναι δομημένα ή και αδόμητα (πλαίσιο κατά SIMON).

Τα δομημένα προβλήματα είναι αυτά τα οποία τυποποιούνται και μπορεί να σχεδιαστεί και να εφαρμοστεί ένα συγκεκριμένο μοντέλο επίλυσής τους. Αυτά τα προβλήματα είναι συνήθως επαναλαμβανόμενα σε τακτά χρονικά διαστήματα κι έτσι αποκτάται εμπειρία στην επίλυσή τους. Επίσης, τα δομημένα προβλήματα χαρακτηρίζονται και από την μεγάλη δυνατότητα συλλογής στοιχείων που βοηθούν στην επίλυσή τους.

Τα αδόμητα προβλήματα έχουν πιο πολύπλοκη μορφή. Συνήθως δεν μπορεί να ακολουθηθεί ένα συγκεκριμένο μοντέλο επίλυσης γι' αυτά. Αντί αυτού, για την επίλυσή τους, χρησιμοποιείται η κεκτημένη ανθρώπινη εμπειρία, η διαίσθηση και η ενόραση.

Για τα δομημένα προβλήματα, τα πληροφοριακά συστήματα αναλαμβάνουν να υλοποιήσουν και να εφαρμόσουν τους κανόνες που καθορίζει το μοντέλο επίλυσής τους. Το μοντέλο υλοποιείται ως μια εφαρμογή (application) η οποία λαμβάνει ως είσοδο (input) την πλούσια συλλογή δεδομένων που αφορά το δομημένο πρόβλημα, επεξεργάζεται τα δεδομένα με τον σαφώς προσδιορισμένο τρόπο επίλυσης και τελικά εξάγει έτοιμη την λύση του προβλήματος (output).

Για τα αδόμητα προβλήματα, τα πληροφοριακά συστήματα έχουν υποστηρικτικό ρόλο. Εφ' όσον δεν υπάρχει τυποποιημένος τρόπος επίλυσής τους, τα πληροφοριακά συστήματα αναλαμβάνουν να επεξεργαστούν στατιστικά στοιχεία προηγούμενων παρόμοιων καταστάσεων, να προβάλουν κάποιο είδος αξιολόγησης των αποφάσεων που είχαν παρθεί και να συγκρίνουν τα αποτελέσματα που είχαν προκύψει. Γενικότερα, ο στόχος των πληροφοριακών συστημάτων σε περιπτώσεις αδόμητων προβλημάτων είναι η βελτίωση της οργάνωσης και παρουσίας των πληροφοριών που παρέχονται στους αποφασίζοντες.[2]

### 2.4 Ο ΚΥΚΛΟΣ ΖΩΗΣ ΤΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ο κύκλος ζωής ενός πληροφοριακού συστήματος περιλαμβάνει όλες τις δραστηριότητες που απαιτούνται για την ανάπτυξη, λειτουργία και συντήρηση του πληροφοριακού συστήματος. Ένας τυπικός κύκλος ζωής περιλαμβάνει έναν αριθμό φάσεων κάθε μία από τις οποίες αποτελεί βάση επάνω στην οποία θα στηριχτεί η επόμενη. Με το τέλος κάθε φάσης παράγεται κάποιο τεκμηριωτικό υλικό στο οποίο συνοψίζονται οι δραστηριότητες που εκτελέστηκαν κατά την φάση αυτή και τα αποτελέσματα που προέκυψαν.

Οι σύγχρονοι οργανισμοί χρησιμοποιούν συνήθως τυποποιημένους κύκλους ζωής για την ανάπτυξη των πληροφοριακών τους συστημάτων και τυποποιημένες μεθοδολογίες για τις τεχνικές εκτέλεσης κάθε φάσης. Τα πλεονεκτήματα της τυποποίησης του κύκλου ζωής ενός πληροφοριακού συστήματος είναι:

- Ο σαφής καθορισμός των δραστηριοτήτων που πρέπει να εκτελεστούν σε έργα ανάπτυξης πληροφοριακών συστημάτων.
- Η συνέπεια μεταξύ των διάφορων έργων ανάπτυξης πληροφοριακών συστημάτων στον ίδιο οργανισμό.
- Η παροχή σημείων ελέγχου για τη διοίκηση του έργου [ref1].

### 3. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

Η πρώτη φάση του κύκλου ζωής ενός πληροφοριακού συστήματος περιλαμβάνει την μελέτη των απαιτήσεων του ενδιαφερόμενου οργανισμού. Αρχικά, προσδιορίζονται τα οφέλη που θα προκύψουν με την υιοθέτησή του, ως στόχοι του νέου πληροφοριακού συστήματος. Αναζητούνται διάφορες εναλλακτικές λύσεις οι οποίες εξυπηρετούν τις απαιτήσεις του οργανισμού και μελετάται η εφικτότητα της κάθε μιας. Τέλος, ετοιμάζεται ένα πρόγραμμα δραστηριοτήτων για την υλοποίηση του έργου και καθορίζονται τα ορόσημα, ως σημεία ελέγχου και αξιολόγησης.

Με τις αρχικές διαπραγματεύσεις μεταξύ του οργανισμού και της εταιρίας που αναλαμβάνει την υλοποίηση του πληροφοριακού συστήματος προκύπτουν:

- Το πλαίσιο λειτουργίας του πληροφοριακού συστήματος
- Τα πιθανά σημεία επικοινωνίας του με άλλα σχετικά έργα και πληροφοριακά συστήματα
- Τα αναμενόμενα οφέλη από την υλοποίησή του.

Στο τέλος αυτής της διαδικασίας ετοιμάζεται μια έκθεση από την εταιρία υλοποίησης, που προσδιορίζει την πρόταση προς τον οργανισμό με μια συνοπτική περιγραφή του έργου και των επιδιωκόμενων στόχων του. Η έκθεση αυτή περιλαμβάνει:

- Μελέτη του παρόντος συστήματος και του τρόπου ενσωμάτωσής του στην λειτουργία του οργανισμού.
- Καθορισμός των πληροφοριακών απαιτήσεων του οργανισμού.
- Σκιαγράφηση εναλλακτικών λύσεων που μπορούν να εφαρμοστούν.

Για κάθε μια από τις εναλλακτικές λύσεις που έχουν σκιαγραφηθεί, γίνεται αξιολόγηση με τη μελέτη της εφικτότητάς τους. Η εφικτότητα κάθε λύσης εξετάζεται σε τρεις τομείς:

- Τεχνική εφικτότητα (technical efficiency)
- Λειτουργική εφικτότητα (operation feasibility)
- Οικονομική εφικτότητα (economic feasibility).

Με την ολοκλήρωση της μελέτης εφικτότητας ο οργανισμός αποφασίζει αν θα προχωρήσει στην ανάπτυξη του πληροφοριακού συστήματος ή όχι. Στην περίπτωση που υπάρχει θετική ανταπόκριση ετοιμάζεται το πρόγραμμα έργου που περιλαμβάνει τα εξής σημεία:

- Τους στόχους του έργου.
- Τις δραστηριότητες που αναλαμβάνονται.
- Τα ορόσημα, ως σημεία ελέγχου και αξιολόγησης.

Η εφαρμογή που αναλαμβάνουμε να πραγματοποιήσουμε έχει ως σκοπό την ικανοποίηση κάποιων πληροφοριακών αναγκών οποιασδήποτε εταιρίας ή οργανισμού. Ο ρόλος του ταχυδρομείου για την μεταφορά τυποποιημένων δεδομένων μεταξύ των συνεργατών αντικαθίσταται από την τεχνολογία του Internet και των δυναμικών ιστοσελίδων. Η αναζήτηση των δεδομένων σε παλιά αρχεία και φακέλους, από μια δύσκολη και χρονοβόρα εργασία μετατρέπεται σε μια στιγμιαία δραστηριότητα προτυποποιημένη και απαλλαγμένη από σφάλματα. Επιπλέον, η ενημέρωση των δεδομένων, που διατηρούνται σε ένα κεντρικό σύστημα βάσης

δεδομένων, καθιστά δυνατή την ενημέρωση πραγματικού χρόνου στους απομακρυσμένους συνεργάτες.

Ο οργανισμός που πρόκειται να χρησιμοποιήσει αυτό το εργαλείο με μια μικρή αλλαγή στο οργανόγραμμά του και ένα χαμηλό κόστος υλοποίησης μπορεί να απολαμβάνει τα παραπάνω πλεονεκτήματα.

Η εγκατάσταση δικτύου Η/Υ στους χώρους των γραφείων του οργανισμού και ενός κέντρου εξυπηρέτησης ιστοσελίδων και βάσεων δεδομένων προϋποθέτουν την εγκαθίδρυση ενός νέου τμήματος. Το νέο αυτό τμήμα είναι σήμερα γνωστό ως τμήμα Μηχανογράφησης και αποτελείται συνήθως από 2 έως 6 άτομα, ανάλογα με το μέγεθος του οργανισμού, που αναλαμβάνουν την διαχείριση του δικτύου και την συντήρηση των εξοπλισμών. Οι γραμματείες των υπόλοιπων τμημάτων του οργανισμού αναβαθμίζεται με τη χρήση Η/Υ συνδεδεμένων στο δίκτυο του οργανισμού.[2]

## 4. ΛΟΓΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

Σε αυτή τη φάση του κύκλου ζωής του πληροφοριακού συστήματος, οι απαιτήσεις του οργανισμού μετασχηματίζονται σε ένα εννοιολογικό μοντέλο. Το εννοιολογικό μοντέλο σχηματίζεται με την μελέτη του κάθε λογικού τμήματος του οργανισμού και τελικά προκύπτει μια έκθεση προδιαγραφών του δομημένου λογικού σχεδιασμού.

Η λειτουργία των τμημάτων δημιουργείται κατά μια τεχνική που ονομάζεται λειτουργική διάσπαση (functional decomposition) και αποτελεί μια από τις πιο κύριες και περίπλοκες δραστηριότητες που αναλαμβάνονται κατά τη διαδικασία ανάπτυξης ενός πληροφοριακού συστήματος.

Τα αποτελέσματα των δραστηριοτήτων που αναλαμβάνονται κατά τη φάση του λογικού σχεδιασμού καταγράφονται στην αναφορά των προδιαγραφών του δομημένου λογικού σχεδιασμού (structured logical design specification). Η αναφορά αυτή περιέχει κυρίως τα εξής στοιχεία:

- Περιγραφή των αναγκών του οργανισμού.
- Διαγραμματική αναπαράσταση των κύριων συστατικών των υποσυστημάτων και των τρόπων επικοινωνίας μεταξύ τους.
- Διαγραμματική αναπαράσταση των εισερχόμενων και της ροής των δεδομένων μέχρι τους τελικούς χρήστες.
- Προδιαγραφές των εισερχόμενων δεδομένων.
- Προδιαγραφές των επεξεργασιών ή μετασχηματισμών των δεδομένων από τις λειτουργίες του συστήματος.
- Περιγραφή βάσεων δεδομένων, πινάκων και αρχείων που θα χρησιμοποιηθούν και των τρόπων μεταφοράς ήδη υπαρχόντων δεδομένων στο νέο σύστημα.
- Προδιαγραφές των εξερχόμενων πληροφοριών.
- Σημεία επικοινωνίας του νέου πληροφοριακού συστήματος με άλλα ήδη υπάρχοντα συστήματα του οργανισμού και με κατανεμημένα κέντρα επεξεργασίας δεδομένων.
- Απαιτήσεις σε πόρους, προσωπικού, υπολογιστών και οικονομικών πόρων.
- Πρόγραμμα έργου υλοποίησης των υποσυστημάτων με ορόσημα και χρονοδιαγράμματα.

Οι οργανισμοί έχουν συνήθως την ανάγκη αποθήκευσης μεγάλου όγκου ιστορικών και στατιστικών δεδομένων. Η συλλογή, αποθήκευση, επεξεργασία και παρουσίαση πληροφοριών απαιτεί κάθε φορά συνεργασία ανθρώπων από το εσωτερικό ή από το εξωτερικό περιβάλλον και είναι διαδικασία συχνά επαναλαμβανόμενη και παραμετροποιήσιμη. Οι διαδικασίες αυτές ως χειρογραφικές είναι αρκετά χρονοβόρες και επίπονες.

Η υιοθέτηση των τεχνολογιών απομακρυσμένης πρόσβασης και ηλεκτρονικής επεξεργασίας των δεδομένων βελτιστοποιούν την απόδοση του οργανισμού σε αυτές τις δραστηριότητες.

Η συλλογή των δεδομένων γίνεται πλέον με τη χρήση του Internet μέσα από ειδικά σχεδιασμένες, ανάλογα με τις απαιτήσεις, φόρμες εισαγωγής δεδομένων. Η αποθήκευση πραγματοποιείται με τη χρήση των σύγχρονων συστημάτων διαχείρισης βάσεων δεδομένων που αναλαμβάνουν πολλές φορές σημαντικό ρόλο και στη

διαδικασία επεξεργασίας των δεδομένων. Τέλος, η προβολή των πληροφοριών γίνεται επίσης με τη χρήση της τεχνολογίας του Internet.

Μια ομάδα εξουσιοδοτημένων χρηστών έχουν πρόσβαση σε αυτό το σύστημα για να το χρησιμοποιήσουν ανεξάρτητα από την γεωγραφική θέση στην οποία βρίσκονται. Το τμήμα της Μηχανογράφησης του οργανισμού προσαρμόζει, ανάλογα με τις απαιτήσεις της εκάστοτε εργασίας, την βάση δεδομένων και τις ιστοσελίδες που θα χρησιμοποιηθούν μέσα από το περιβάλλον της εφαρμογής.[2]

## 5. ΦΥΣΙΚΟΣ ΣΧΕΔΙΑΣΜΟΣ

Σε αυτή τη φάση του κύκλου ζωής του πληροφοριακού συστήματος χρησιμοποιούνται τα αποτελέσματα του λογικού σχεδιασμού για την ανάπτυξη λεπτομερών προδιαγραφών για τα υποσυστήματα, τη βάση δεδομένων και τα προγράμματα εφαρμογών.

Για την προσαρμογή αυτή των λογικών τμημάτων σε υπολογιστικό περιβάλλον απαιτείται προετοιμασία διαγραμμάτων δομών και ροών που σκοπό έχουν τον προσδιορισμό της ακεραιότητας των τμημάτων και της ανθεκτικότητάς τους. Ένα τμήμα θεωρείται ανθεκτικό όταν είναι ανεξάρτητο από το περιβάλλον του, πραγματοποιεί μια καλά προσδιορισμένη εργασία και έχει σχετικά μικρό μέγεθος.

Το λογισμικό που σχεδιάζεται σε αυτή τη φάση, προκειμένου να διατηρηθεί σε μεγάλο βαθμό την ακεραιότητα και την ανθεκτικότητά του, τμηματοποιείται (modular applications software). Το τμηματοποιημένο λογισμικό προωθεί την αρχή της απόκρυψης πληροφοριών ώστε οι διαδικασίες και τα δεδομένα κάθε τμήματος να είναι προσπελάσιμα από άλλα τμήματα και να μην μπορεί να επηρεαστούν.

Στη φάση αυτή του κύκλου ζωής παράγεται επίσης ένα πρόγραμμα δοκιμών του πληροφοριακού συστήματος το οποίο θα χρησιμοποιηθεί στη διαδικασία εντοπισμού σφαλμάτων.

Τα κύρια χαρακτηριστικά του νέου πληροφοριακού συστήματος είναι η χρήση της τεχνολογίας του Internet και των βάσεων δεδομένων. Στους γρήγορα αναπτυσσόμενους και ανταγωνιστικούς οργανισμούς της σύγχρονης οικονομίας όλες οι δυνατότητες του πληροφοριακού συστήματος θα πρέπει να είναι δυναμικές και γρήγορα προσαρμόσιμες.

Η εφαρμογή SuiteCase σχεδιάστηκε με αυτές ακριβώς τις προδιαγραφές, προσφέροντας εργαλεία στους χρήστες της, ώστε να ανταπεξέλθουν στις απαιτήσεις ταχύτητας και προσαρμοστικότητας. Τα εργαλεία που παρέχει η εφαρμογή διευκολύνουν τη διαχείριση της βάσης δεδομένων και τη δημιουργία αναφορών και φορμών εισαγωγής μέσα από ένα γραφικό περιβάλλον.

Το εργαλείο DataView επιτρέπει την προβολή των αποθηκευμένων δεδομένων χωρίς μορφοποίηση, την εύκολη δημιουργία και προσαρμογή βάσεων δεδομένων και αντικειμένων που την αποτελούν καθώς και εισαγωγή, ενημέρωση και διαγραφή δεδομένων.

Το εργαλείο ScriptEdit επιτρέπει την εκτέλεση δέσμης ενεργειών στην βάση δεδομένων για πολύπλοκες διαδικασίες που μπορεί να είναι και συχνά επαναχρησιμοποιήσιμες. Το ίδιο εργαλείο επιτρέπει την αποθήκευση του κώδικα που γράφει ένας χρήστης, ή και το άνοιγμα και εκτέλεση αρχείων κώδικα που έχουν συνταχθεί από τρίτους.

Το εργαλείο QueryWizard υποβοηθά τους χρήστες να συντάξουν ερωτήματα ανάκτησης και επεξεργασίας δεδομένων τα οποία μπορούν να χρησιμοποιηθούν για τη δημιουργία αναφορών.

Τα εργαλεία δημιουργίας αναφορών και φορμών εισαγωγής δεδομένων παρέχουν στους χρήστες την δυνατότητα κατασκευής αυτόματων ιστοσελίδων στατικών αλλά και δυναμικών.[2]



## 6. ΑΝΑΠΤΥΞΗ ΠΡΟΓΡΑΜΜΑΤΩΝ

Σε αυτή τη φάση του κύκλου ζωής αναπτύσσεται το λογισμικό που σχεδιάστηκε σύμφωνα με τις αναλύσεις που πραγματοποιήθηκαν στις προηγούμενες φάσεις. Συνήθως επιλέγεται μια γλώσσα προγραμματισμού ανωτέρου επιπέδου για την ανάπτυξη των προγραμμάτων και των υποσυστημάτων τους.

Αρχικά, κωδικοποιούνται τα τμήματα ελέγχου ροής των διαδικασιών και έπειτα τα τμήματα που πραγματοποιούν τις επεξεργασίες.

Η συγγραφή ενός προγράμματος, οποιουδήποτε μήκους και πολυπλοκότητας μπορεί να πραγματοποιηθεί με την χρήση των τριών λογικών δομών: της ακολουθίας, της επιλογής και της επανάληψης. Η χρήση αυτών των δομών συνιστά τον δομημένο προγραμματισμό ο οποίος προωθεί την βελτίωση της ποιότητας των προγραμμάτων, την αύξηση της παραγωγικότητας των προγραμματιστών και την ελάττωση των χρονικών απαιτήσεων των δοκιμών.

Στον δομημένο προγραμματισμό οι προγραμματιστές που αναλαμβάνουν την ανάπτυξη του λογισμικού διαρθρώνονται σε μία ιεραρχία ανάλογα με την εμπειρία τους. Οι λιγότερο έμπειροι προγραμματιστές αναλαμβάνουν την υλοποίηση των ροών του προγράμματος και οι περισσότεροι έμπειροι τις επεξεργασίες των δεδομένων.

Ο προϊστάμενος προγραμματιστών εποπτεύει τεχνικά και διοικητικά την ομάδα και φέρει την ευθύνη των υλοποιήσεων των διαδικασιών των ροών και των επεξεργασιών των δεδομένων.

Ο προγραμματιστής υποστήριξης παίζει επίσης τον ρόλο του προϊσταμένου προγραμματισμών σε περιπτώσεις ανάγκης και έχει σφαιρική τεχνική άποψη του πληροφοριακού συστήματος.

Ο βιβλιοθηκάριος είναι υπεύθυνος για τη συντήρηση της βιβλιοθήκης υποστήριξης και του τεκμηριωτικού υλικού που προκύπτει από την ανάπτυξη των προγραμμάτων. Γενικότερα, αναλαμβάνει τις διοικητικές διαδικασίες γραφείου που αφορούν το εσωτερικό και το εξωτερικό τμήμα του πληροφοριακού συστήματος.[2]

Κάθε εργαλείο της εφαρμογής SuiteCase αναπτύχθηκε ως αυτόνομη εφαρμογή. Σχεδιάστηκαν τα παράθυρα και οι λειτουργίες του κάθε εργαλείου ξεχωριστά. Τελικά, εφ' όσον διαπιστώθηκε η σωστή λειτουργία της κάθε αυτόνομης εφαρμογής, ο κώδικας μεταφέρθηκε σε περιβάλλον ανάπτυξης βιβλιοθήκης δυναμικών συνδέσεων.

Εκτός από τα τρία βασικά εργαλεία, που αποτελούν το καθένα από μια βιβλιοθήκη δυναμικών συνδέσεων, υπάρχουν και τα υποσυστήματα της υπογλώσσας ορισμού δεδομένων, της υπογλώσσας διαχείρισης δεδομένων και των εργαλείων δημιουργίας ιστοσελίδων. Αυτές οι δευτερεύουσες βιβλιοθήκες δυναμικών συνδέσεων χρησιμοποιούνται από τα τρία κυρίως εργαλεία.

## 7. ΥΛΟΠΟΙΗΣΗ

Κατά τη φάση της υλοποίησης του πληροφοριακού συστήματος ολοκληρώνονται οι δοκιμές του λογισμικού. Το προσωπικό του οργανισμού εκπαιδεύεται στη χρήση του νέου πληροφοριακού συστήματος και προετοιμάζεται για την μετάβαση σε αυτό. Τα δεδομένα του προηγούμενου πληροφοριακού συστήματος μετατρέπονται σε μορφή κατάλληλη ώστε να χρησιμοποιηθούν και από το νέο.[2]

Για την ποιοτική εξασφάλιση του νέου πληροφοριακού συστήματος χρησιμοποιούνται τα εξής επίπεδα ελέγχου:

- Δοκιμή: Διαδικασία εκτέλεσης των δραστηριοτήτων με σκοπό την ανεύρεση σφαλμάτων.
- Επαλήθευση: Διαδικασία εκτέλεσης ενός προγράμματος με σκοπό την επιβεβαίωση της σωστής λειτουργίας του.
- Έλεγχος σωστής λειτουργίας: Διαδικασία εκτέλεσης ενός προγράμματος σε πραγματικό περιβάλλον με σκοπό την ανεύρεση λαθών.
- Πιστοποίηση: Διαδικασία επικύρωσης της ορθότητας ενός προγράμματος.

Το προσωπικό του τμήματος Μηχανογράφησης εξοικειώνεται με το περιβάλλον της εφαρμογής και μαθαίνει να χρησιμοποιεί τα εργαλεία. Βασική προϋπόθεση χρήσης της εφαρμογής είναι η γνώση κάποιων βασικών μονάχα κανόνων της γλώσσας SQL και της θεωρίας σχεδίασης βάσεων δεδομένων.

Τα αρχικά αρχεία σεναρίων και κάποιες εναλλαγές τους συντάσσονται από το προσωπικό της εταιρίας υλοποίησης του πληροφοριακού συστήματος και παρέχεται εφ' όσον ζητηθεί υπηρεσία υποστήριξης για τη συντήρηση των αρχείων σεναρίων.

## 8. ΛΕΙΤΟΥΡΓΙΑ

Κατά την τελευταία φάση του κύκλου ζωής εξετάζεται το πληροφοριακό σύστημα και οι δυνατότητές του ως προς το όφελος που παρέχει στον οργανισμό.

Μέχρι την σταθεροποίηση και την πλήρη ενσωμάτωση του πληροφοριακού συστήματος στον οργανισμό γίνεται αξιολόγηση της λειτουργίας του. Η αξιολόγηση της λειτουργίας γίνεται σε δύο επίπεδα:

- Στο πρώτο επίπεδο συγκρίνεται η λειτουργία του συστήματος με τις προβλεπόμενες από τις προδιαγραφές δυνατότητές του που προέκυψαν από τον λογικό σχεδιασμό.
- Στο δεύτερο επίπεδο περιλαμβάνεται η διαδικασία μάθησης και εκτίμησης για την μελλοντική ανάπτυξη των πληροφοριακών συστημάτων, καθώς και η αξιολόγηση της αξιοποίησης και της παραγωγικότητας της ομάδας ανάπτυξης.[2]

## 9. ΑΡΧΙΤΕΚΤΟΝΙΚΗ CLIENT SERVER

Το μοντέλο client-server είναι μια επέκταση του μοντέλου ανάπτυξης λογισμικού σε αυτοτελή τμήματα (modules), όπου μεγάλα τμήματα λογισμικού είναι δομημένα από μικρότερα τμήματα συνδεδεμένα μεταξύ τους. Η σύνδεση μεταξύ των διαφόρων τμημάτων πραγματοποιείται με ανταλλαγή μηνυμάτων ή με την ανταλλαγή κλήσης απομακρυσμένων διαδικασιών (RPC – Remote Procedure Calling). Το τμήμα του λογισμικού που εκτελεί μια κλήση ονομάζεται client και το τμήμα που λαμβάνει την κλήση ονομάζεται server.

Το μοντέλο client-server υλοποιείται συνήθως σε ένα δίκτυο Η/Υ όπου client είναι η εφαρμογή που χρησιμοποιούν οι χρήστες και server είναι μια υπηρεσία που παρέχεται από ένα υπολογιστικό σύστημα υψηλότερων, συνήθως, προδιαγραφών. Επίσης, οι διεργασίες του client και του server μπορεί να φιλοξενούνται στο ίδιο υπολογιστικό σύστημα, ανταλλάσσοντας πληροφορίες που αφορούν την αίτηση ενός χρήστη. Οι χρήστες, χρησιμοποιώντας την client εφαρμογή, απαιτούν από τον server την αναζήτηση μιας ή περισσότερων εγγραφών από μια βάση δεδομένων ή την επιστροφή ενός αρχείου.

Η λειτουργία μιας client εφαρμογής περιλαμβάνει την δυνατότητα εισαγωγής δεδομένων από τον χρήστη μέσα σε ένα γραφικό περιβάλλον αλληλεπίδρασης (GUI – Graphical User Interface), τον έλεγχο εγκυρότητας των δεδομένων και την επικοινωνία με την διεργασία του server για την αποστολή τους.

Η λειτουργία της διεργασίας του server περιλαμβάνει την ανάλυση των δεδομένων που λαμβάνει από τους clients, την επεξεργασία τους ανάλογα με το είδος της υπηρεσίας που προσφέρει και την επιστροφή των αποτελεσμάτων. Το τελευταίο στάδιο της αλληλεπίδρασης μεταξύ του client και του server περιλαμβάνει την προβολή των αποτελεσμάτων στην εφαρμογή του client.

Σε μεγάλα πληροφοριακά συστήματα όπου χρησιμοποιούνται περισσότεροι από ένας servers και αντίστοιχα πολλοί clients, συνήθως κρίνεται απαραίτητη η χρήση μιας ενδιάμεσης διεργασίας. Αυτή η ενδιάμεση διεργασία αναλαμβάνει να αποθηκεύσει συχνά ανακτόμενες πληροφορίες (caching), να διαμοιράσει τις αιτήσεις των clients σε περισσότερους από έναν servers σε περιπτώσεις παράλληλης επεξεργασίας και να εκτελέσει ενέργειες μετάφρασης και πιστοποίησης όπου είναι απαραίτητο με σκοπό την βελτιστοποίηση της λειτουργίας του συστήματος.[12]

## 10. ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

### 10.1 ΓΕΝΙΚΑ

Η τεχνολογία των βάσεων δεδομένων προέκυψε από τις έρευνες για βελτίωση του τρόπου αποθήκευσης, οργάνωσης, ανάκτησης και προβολής μεγάλου όγκου δεδομένων.

Η θεωρητική τεχνολογική βάση επάνω στην οποία στηρίζονται τα συστήματα βάσεων δεδομένων είναι το σχεσιακό μοντέλο. Το σχεσιακό μοντέλο προτείνει έναν τρόπο οργάνωσης των αποθηκευμένων δεδομένων, ο οποίος έχει ως στόχο τη σωστή και εύκολη διαχείριση των δεδομένων. Το σχεσιακό μοντέλο σχεδιάστηκε ειδικά για να βελτιώσει τους τρόπους οργάνωσης των αποθηκευμένων δεδομένων και τους τρόπους ανάκτησής τους, χρησιμοποιώντας και εκμεταλεύοντας τις δυνατότητες των υπολογιστικών συστημάτων στο έπακρο.

Τα συστήματα διαχείρισης βάσεων δεδομένων (DBMS – Data Base Management Systems) παρέχουν την δυνατότητα μορφοποίησης των δομών και των δεδομένων που διατηρούν μέσω μιας ειδικά σχεδιασμένης γλώσσας προγραμματισμού, της SQL (Structured Query Language). Η γλώσσα αυτή είναι τυποποιημένη και αρκετά σύντομη σε έκταση ώστε να γίνεται γρήγορα κατανοητή και αναγνώσιμη από τους προγραμματιστές. Παρ' όλα αυτά, οι διαφορετικές υλοποιήσεις του σχεσιακού μοντέλου από τους κατασκευαστές προκαλούν μικρές ή και μεγαλύτερες διαφοροποιήσεις οι οποίες ονομάζονται διάλεκτοι (dialects). Η διαφοροποιημένη γλώσσα SQL που προτείνει ο SQL Server της Microsoft ονομάζεται Transact-SQL, και είναι επικεντρωμένη στις δυνατότητες του συγκεκριμένου server.

Η γλώσσα SQL περιέχει εντολές που χωρίζονται σε δύο κατηγορίες, καθεμία από τις οποίες ονομάζεται επίσης γλώσσα:

- Γλώσσα ορισμού δεδομένων (DDL – Data Definition Language): είναι μια υπογλώσσα της SQL που περιλαμβάνει τις εντολές δημιουργίας και κατάργησης αντικειμένων. Τα αντικείμενα είναι τα συστατικά στοιχεία της δομής μιας βάσης δεδομένων και η οργάνωσή τους είναι σημαντικός παράγοντας αποδοτικότητας για την βάση δεδομένων.
- Γλώσσα διαχείρισης δεδομένων (DML – Data Manipulation Language): είναι η υπογλώσσα της SQL που περιλαμβάνει τις εντολές εισαγωγής, ενημέρωσης, διαγραφής, επεξεργασίας και ανάκτησης των δεδομένων. Τα δεδομένα είναι στην ουσία τα ωφέλιμα στοιχεία που αποθηκεύονται σε μία βάση δεδομένων και προσδιορίζουν την χρησιμότητά της όσον αφορά την πληρότητα των πληροφοριών που προκύπτουν από την επεξεργασία.

Τα συστήματα βάσεων δεδομένων υλοποιούνται χρησιμοποιώντας την αρχιτεκτονική client – server. Ένα σύγχρονο σύστημα διαχείρισης βάσεων δεδομένων διαθέτει ένα τμήμα server το οποίο αναλαμβάνει να εξυπηρετήσει τις συνδέσεις των χρηστών/clients.

Η εφαρμογή του server είναι συνήθως εγκατεστημένη σε ένα υπολογιστικό σύστημα υψηλών προδιαγραφών, μαζί με τα δομημένα, οργανωμένα αρχεία των δεδομένων που διατηρεί. Οι χρήστες/clients συνδέονται με αυτό το υπολογιστικό σύστημα μέσω δικτύου.

Οι Database servers, έτσι όπως διανέμονται από τις κατασκευάστριες εταιρίες, διαθέτουν εργαλεία τα οποία επιτρέπουν την διαχείριση των συνδέσεων μέσω δικτύου μεταξύ των clients και των servers. Συνήθως, ένα υπολογιστικό σύστημα το οποίο πρόκειται να χρησιμοποιηθεί ως client ενός database server, θα πρέπει να έχει εγκατεστημένο ένα ειδικά σχεδιασμένο εργαλείο το οποίο αναλαμβάνει το κομμάτι της επικοινωνίας του client με τον server.

Στο υπολογιστικό σύστημα που χρησιμοποιείται ως database server, υπάρχει επίσης εγκατεστημένο ένα ειδικά σχεδιασμένο λογισμικό το οποίο εγκαθιστά τα interfaces με τα οποία μπορούν να συνδεθούν οι clients. Ο database administrator μπορεί να ενεργοποιήσει και να απενεργοποιήσει τα κατάλληλα πρωτόκολλα και να ορίσει τα ports που θα χρησιμοποιούνται.

## 10.2 ΤΟ ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Ένα σύστημα μπορεί να θεωρηθεί ότι ακολουθεί το σχεσιακό μοντέλο αν υποστηρίζει:

- Τις δομικές απαιτήσεις του σχεσιακού μοντέλου
- Τους κανόνες προσθήκης, ενημέρωσης και διαγραφής
- Μια δυνατή υπογλώσσα, όπως την σχεσιακή άλγεβρα.

Μια σχέση αποτελείται από ένα σύνολο συστάδων που περιλαμβάνουν ισάριθμο πλήθος ιδιοτήτων, και μπορεί να παρασταθεί ως ένας διδιάστατος πίνακας. Ο πίνακας αυτός έχει τις εξής ιδιότητες:

- Οι γραμμές/συστάδες δεν επαναλαμβάνονται.
- Η σειρά εμφάνισης των γραμμών/συστάδων δεν έχει σημασία.
- Η σειρά εμφάνισης των στηλών/ιδιοτήτων δεν έχει σημασία.
- Κάθε πεδίο του πίνακα περιέχει αδιάσπαστες πληροφορίες.

Το σχεσιακό μοντέλο αποτελείται από:

- Ένα σύνολο σχέσεων με μορφή διδιάστατου πίνακα
- Τους κανόνες προσθήκης, ενημέρωσης και διαγραφής
- Την σχεσιακή άλγεβρα.

Οι σχέσεις ως σύνολα υποστηρίζουν τις πράξεις της ένωσης, της τομής και της διαφοροποίησης. Μια επιπλέον πράξη που υποστηρίζεται είναι το καρτεσιανό γινόμενο, το οποίο, εν αντιθέσει με τις άλλες πράξεις, δεν απαιτεί τα σύνολα να είναι συμβατά με την πράξη της ένωσης.[13]

## 11. OLE DB

### 11.1 ΕΙΣΑΓΩΓΗ

Οι σύγχρονες απαιτήσεις των οργανισμών για πληροφορίες χρήσιμες στη λύση δομημένων και αδόμητων προβλημάτων που προκύπτουν, υπαγορεύουν τη διαχείριση δεδομένων που προέρχονται από διαφόρων ειδών ηλεκτρονικές πηγές πέρα από τα συστήματα βάσεων δεδομένων σχεσιακού μοντέλου. Τέτοιες ηλεκτρονικές πηγές μπορεί να είναι το σύστημα αρχείων (file system), λογιστικά φύλλα (spreadsheets), εργαλεία διαχείρισης έργων (project management tools) και εφαρμογές ηλεκτρονικής αλληλογραφίας (e-mail).

Τα σαφή πλεονεκτήματα του σχεσιακού μοντέλου, έναντι των υπολοίπων τεχνολογιών αποθήκευσης δεδομένων, υπαγορεύουν την μεταφορά όλων των δεδομένων ενός οργανισμού σε ένα σύστημα διαχείρισης βάσεων δεδομένων (DBMS – Data Base Management System). Η διαδικασία της μεταφοράς αυτής αποτελεί μια πολύπλοκη δραστηριότητα και ιδιαίτερα απαιτητική σε πόρους.

Η τεχνολογία OLE DB παρέχει ένα σύστημα στοιχείων ελέγχου COM που παρέχει στις εφαρμογές την δυνατότητα ενοποιημένης πρόσβασης σε διαφόρων ειδών αποθήκες δεδομένων. Η μορφή της ως βιβλιοθήκη επιτρέπει την χρήση των ιδιαίτερων χαρακτηριστικών κάθε είδους πηγής δεδομένων μέσα από ειδικά σχεδιασμένες εφαρμογές.

Η βιβλιοθήκη OLE DB επιτρέπει στις εφαρμογές να χρησιμοποιήσουν αντικείμενα αποθήκευσης γραμμών δεδομένων, επεξεργαστές ερωτημάτων (queries) και διαχειριστές περιβάλλοντος αλληλεπίδρασης ανεξάρτητα από το είδος της ηλεκτρονικής πηγής δεδομένων με την οποία συνεργάζονται.

### 11.2 ΚΑΤΑΝΑΛΩΤΕΣ ΚΑΙ ΠΑΡΟΧΕΙΣ

Τα συστατικά μέρη ενός συστήματος όπου εφαρμόζεται η τεχνολογία OLE DB είναι οι καταναλωτές (consumers) και οι παροχείς (providers). Ως καταναλωτής ορίζεται μια εφαρμογή η οποία χρησιμοποιεί ένα αντικείμενο πρόσβασης σε βάση δεδομένων OLE DB. Ως παροχέας ορίζεται μια εφαρμογή η οποία παρέχει την δυνατότητα σύνδεσης με αυτή σε άλλες εφαρμογές μέσω αντικειμένων πρόσβασης OLE DB.

Υπάρχουν δύο ειδών παροχείς OLE DB:

- Οι παροχείς υπηρεσιών είναι συστατικά ενός συστήματος το οποίο δεν διαθέτει αποθηκευμένα δεδομένα, αλλά παρέχει πρόσβαση σε μια πηγή δεδομένων για μια εφαρμογή – καταναλωτή. Οι παροχείς υπηρεσιών σε αυτές τις περιπτώσεις λειτουργούν ως παροχείς αλλά και ως καταναλωτές.
- Οι παροχείς δεδομένων είναι συστήματα αποθήκευσης δεδομένων τα οποία παρέχουν τη δυνατότητα στις εφαρμογές – καταναλωτές να συνδεθούν μαζί τους μέσω OLE DB αντικειμένων και χωρίς την παρεμβολή άλλου μεσάζοντος.

### 11.3 ΣΥΣΤΑΤΙΚΑ ΜΕΡΗ ΣΥΣΤΗΜΑΤΟΣ ΤΕΧΝΟΛΟΓΙΑΣ OLE DB

Τα συστατικά μέρη του συστήματος πρόσβασης σε πηγές δεδομένων OLE DB είναι τα εξής:

- Enumerators: Αναλαμβάνουν την αναζήτηση των διαθέσιμων παροχέων OLE DB για την επιλογή χρήσης από την εφαρμογή – καταναλωτή.
- Data Sources: Περιέχουν την λειτουργικότητα σύνδεσης μιας εφαρμογής - καταναλωτή με μια πηγή δεδομένων.
- Sessions: Παρέχουν το περιβάλλον αλληλεπίδρασης μιας εφαρμογής – καταναλωτή με μια πηγή δεδομένων.
- Transactions: Παρέχουν τη δυνατότητα αποθήκευσης αλλαγών μέσα σε ένα περιβάλλον αλληλεπίδρασης.
- Commands: Αντικείμενα που αναλαμβάνουν να εκτελέσουν μια εντολή SQL σε μορφή κειμένου και να δημιουργήσουν πιθανόν αντικείμενο Rowset.
- Rowsets: Αντικείμενα που χρησιμοποιούνται για την ανάκτηση δεδομένων από την πηγή δεδομένων σε μορφή πίνακα δύο διαστάσεων.
- Errors: Αντικείμενα περιγραφής σφαλμάτων που προκύπτουν από τις αλληλεπιδράσεις της εφαρμογής – καταναλωτή με την πηγή δεδομένων.[10]



## 12. ACTIVE SERVER PAGES (ASP)

### ΕΙΣΑΓΩΓΗ

Η τεχνολογία ASP δημιουργήθηκε με σκοπό τον εμπλουτισμό της γλώσσας περιγραφής ιστοσελίδων HTML με δυνατότητες υποστήριξης όλων των αλγοριθμικών δομών και της χρήσης βιβλιοθηκών. Η γλώσσα περιγραφής ιστοσελίδων HTML είναι αρκετά λειτουργική αλλά περιορίζεται στην δημιουργία στατικών προτύπων. Οι απαιτήσεις της σύγχρονης εποχής για δυναμική διαχείριση περιεχομένου υπαγορεύουν την χρήση νέων τεχνολογιών, όπως η ASP, που υπερκαλύπτουν τις δυνατότητες της HTML.

### ΔΥΝΑΤΟΤΗΤΕΣ

Η ενσωμάτωση μιας γλώσσας προγραμματισμού στην HTML επιτρέπει να χρησιμοποιηθεί η δύναμη ενός Web Server στην επεξεργασία των απαιτήσεων των χρηστών, ώστε να δημιουργηθούν δυναμικές και ατομικοποιημένες ιστοσελίδες. Τα δεδομένα που μπορεί να περιλαμβάνουν τέτοιες ιστοσελίδες μπορεί να προέρχονται από βάσεις δεδομένων αλλά και από αρχεία.

Η συνδιάλεξη μεταξύ του Web Server και του Client (browser) γίνεται κάθε φορά αντιμετωπίζοντας τον κάθε client ως διαφορετική οντότητα με δυνατότητες:

- Αποθήκευσης προσωπικών ρυθμίσεων.
- Ανάκτησης δεδομένων από το σύστημα αρχείων (file system).
- Ανάκτησης δεδομένων από βάσεις δεδομένων.
- Χρήσης οποιουδήποτε στοιχείου ελέγχου COM.

### ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Η αποτελεσματική χρήση της τεχνολογίας ASP απαιτεί από τους προγραμματιστές γνώση πολλών τεχνικών, γλωσσών προγραμματισμού και τεχνολογιών, καθώς και χρήση εργαλείων. Τα πλεονεκτήματα της τεχνολογίας ASP για τους προγραμματιστές είναι σε γενικές γραμμές:

- Η δυνατότητα χρήσης πολλών γλωσσών προγραμματισμού ακόμα και ταυτόχρονα,
- Η εύκολη χρήση των μεταγλωττισμένων γλωσσών προγραμματισμού που συνήθως χρησιμοποιούνται,
- Η ευκολία συντήρησης και υποστήριξης του κώδικα.

Τα scripts της τεχνολογίας ASP αναλαμβάνουν να δημιουργήσουν και να εμφανίσουν στους clients δυναμικές ιστοσελίδες που ανταποκρίνονται στις εκάστοτε απαιτήσεις. Αυτό είναι το χαρακτηριστικό που εδραιώνει τη χρήση της στο σύγχρονο Internet.[9]

## 13. Η ΕΦΑΡΜΟΓΗ SUITECASE

### ΕΙΣΑΓΩΓΗ

Η εφαρμογή SuiteCase είναι ένα εργαλείο διαχείρισης βάσεων δεδομένων. Ο σκοπός της εφαρμογής είναι να συνεργάζεται με έναν SQL Server, συγκεκριμένα με τον Microsoft SQL Server 2000 ώστε να παρέχει ένα φιλικό γραφικό περιβάλλον στον χρήστη – προγραμματιστή. Μέσα από αυτό το γραφικό περιβάλλον ο χρήστης – προγραμματιστής μπορεί να δημιουργήσει και να τροποποιήσει βάσεις δεδομένων, πίνακες και όψεις. Επίσης, μπορεί να εισάγει, να διαγράψει και ενημερώσει τα δεδομένα που βρίσκονται αποθηκευμένα στη βάση δεδομένων.

Με άλλα λόγια, η εφαρμογή έχει σκοπό να διευκολύνει τους προγραμματιστές βάσεων δεδομένων προσφέροντας ένα γραφικό περιβάλλον – εργαλείο για τον σχηματισμό των εντολών SQL. Ως απώτερος στόχος της εφαρμογής θεωρείται η δημιουργία ιστοσελίδων με την μορφή αναφοράς δεδομένων (Reports) και τη μορφή φορμών εισαγωγής δεδομένων.

Οι ιστοσελίδες που προκύπτουν μπορούν να χρησιμοποιηθούν από έναν οργανισμό στο Intranet που διαθέτει, σε ένα εικονικό Ιδιωτικό δίκτυο (VPN – Private Virtual Network) ή ακόμα και στο Internet. Αυτή η δυνατότητα δίνει την ευκαιρία στα στελέχη και στους συνεργάτες του οργανισμού να έχουν απομακρυσμένη πρόσβαση στα δεδομένα.

Η εφαρμογή SuiteCase συνδέεται με οποιαδήποτε βάση δεδομένων ενός Microsoft SQL Server 2000, ζητώντας από τον χρήστη να εισάγει το όνομα και τον κωδικό πρόσβασης που χρησιμοποιεί σε αυτό το σύστημα. Υπάρχει και η δυνατότητα πιστοποίησης ενός χρήστη από τα στοιχεία εισαγωγής του στο λειτουργικό σύστημα (Windows NT, Windows 2000) έτσι όπως υποστηρίζεται από το σύστημα του SQL Server 2000 (Trusted Connection).

Κατά την εισαγωγή του ένας χρήστης έχει την δυνατότητα να επιλέξει την βάση δεδομένων στην οποία επιθυμεί να συνδεθεί, αλλά και το όνομα του υπολογιστικού συστήματος που φιλοξενεί τον SQL Server, σε περίπτωση που ο οργανισμός διαθέτει περισσότερους από έναν SQL Servers για την κάλυψη των αναγκών του.

### ΠΡΩΤΕΥΟΝΤΑ ΕΡΓΑΛΕΙΑ

Τα τμήματα από τα οποία αποτελείται η εφαρμογή είναι αρχικά εννοιολογικά αλλά και λειτουργικά διαχωρισμένα. Τα εργαλεία που συνήθως χρειάζονται και χρησιμοποιούν οι προγραμματιστές βάσεων δεδομένων είναι τα εξής:

- Γραφικό περιβάλλον καθορισμού της μορφολογίας των βάσεων δεδομένων, διαχείρισης δεδομένων και προβολής (DataView).
- Επεξεργαστής κειμένου για την συγγραφή script και την εκτέλεσή τους (ScriptEdit).
- Οδηγός δημιουργίας εντολών ανάκτησης δεδομένων, χρήσιμος κυρίως στους λιγότερο έμπειρους προγραμματιστές (QueryWizard).

## **ΒΟΗΘΗΤΙΚΑ ΕΡΓΑΛΕΙΑ**

Βοηθητικά, δευτερεύοντα συστατικά μέρη της εφαρμογής που χρησιμοποιούνται από τα πρωτεύοντα είναι τα εξής:

- Γραφικό περιβάλλον δημιουργίας και κατάργησης αντικειμένων βάσης δεδομένων.
- Γραφικό περιβάλλον εισαγωγής, ενημέρωσης και διαγραφής δεδομένων.
- Εργαλείο δημιουργίας αναφορών δεδομένων με χρήση της τεχνολογίας του Internet και της τεχνολογίας ASP.
- Εργαλείο δημιουργίας φορμών εισαγωγής δεδομένων σε ASP μορφή.

## 14. MASTEROLEDB

### ΕΙΣΑΓΩΓΗ

Η βάση επάνω στην οποία χτίστηκε η εφαρμογή SuiteCase είναι η class CMasterOledb. Αυτή η class περιλαμβάνει όλες τις συναρτήσεις που χρειάζονται για την σύνδεση της εφαρμογής με τον Microsoft SQL Server 2000, χρησιμοποιώντας την βιβλιοθήκη OLE DB (Object Linking and Embedding, Data Base). Για την ανάπτυξη αυτής της class χρησιμοποιήθηκε ο παροχέας (Provider) “SQLOLEDB” της Microsoft από το MSDAC SDK v2.6 (Microsoft Data Access Components Software Development Kit) που είναι ειδικά σχεδιασμένος για την ανάπτυξη εφαρμογών με τον SQL Server της Microsoft.

Η λειτουργία αυτής της class ακολουθεί τα εξής βήματα:

- Δημιουργία σύνδεσης με την πηγή δεδομένων (DataSource).
- Δημιουργία περιβάλλοντος αλληλεπίδρασης με την πηγή δεδομένων (Session).
- Δημιουργία αντικείμενου αλληλεπίδρασης με την πηγή δεδομένων.

Το αντικείμενο αλληλεπίδρασης με την πηγή δεδομένων μπορεί να είναι:

- Ένα σύνολο επιστρεφόμενων δεδομένων / πληροφοριών (Rowset) μετά από την εκτέλεση μιας εντολής ανάκτησης δεδομένων (π.χ. `select * from MyTable`).
- Μια εντολή που δεν επιστρέφει κάποιο σύνολο δεδομένων / πληροφοριών. (π.χ. `use Northwind, Insert into MyTable(ID, Name, Surname) values(1, 'Chris', 'Maunder')`, `delete from MyTable where ID=5`, `create database MyEnterprise` κτλ).

Μια επιπλέον δυνατότητα της class CMasterOledb είναι η ανίχνευση των σφαλμάτων. Ο χρήστης της εφαρμογής είναι δυνατό να εκτελέσει οποιαδήποτε εντολή SQL, είτε δημιουργώντας την μέσα από τα παράθυρα διαλόγου της εφαρμογής είτε συντάσσοντάς την στο παράθυρο του εργαλείου ScriptEdit. Μεταξύ αυτών των εντολών που εκτελεί ο χρήστης είναι πιθανό να προκύψουν κάποιες εσφαλμένες, για τους εξής λόγους:

- Ως χρήστης της βάσης δεδομένων μπορεί να μην έχει πρόσβαση σε ένα αντικείμενο που προσπαθεί να προσπελάσει.
- Ο SQL Server δεν καταφέρνει να εντοπίσει το αντικείμενο το οποίο προσπαθεί να προσπελάσει ο χρήστης.
- Η εντολή, που επιχειρήσε να εκτελέσει ο χρήστης, περιέχει συντακτικά λάθη.

Η δομή εργασίας με την class CmasterOledb χρησιμοποιεί το μοντέλο δημιουργίας – εκτέλεσης – καταστροφής (ή αρχικοποίησης – εκτέλεσης – τερματισμού). [Σχήμα 1].

## ΠΗΓΗ ΔΕΔΟΜΕΝΩΝ (DATASOURCE)

Με την έναρξη της εφαρμογής SuiteCase δημιουργείται το αντικείμενο σύνδεσης με την πηγή δεδομένων:

```
CMasterOledb* master;
master = new CMasterOledb;
master->InitDataSource();
```

Η συνάρτηση `InitDataSource` αρχικοποιεί έναν δείκτη σε ένα αντικείμενο `IUnknown`, το αντικείμενο που αποτελεί βάση κληρονομικότητας για όλα τα `Interfaces` της τεχνολογίας COM (Component Object Model). Το συγκεκριμένο αντικείμενο αρχικοποιείται ως παράθυρο εισαγωγής ονόματος χρήστη (User Name), κωδικού εισόδου (Password) και του ονόματος του SQL Server (Server Name).

Το αντικείμενο σύνδεσης με την πηγή δεδομένων αποθηκεύεται ως ιδιωτικό μέλος του αντικειμένου της class `CMasterOledb` (private data member). Το ίδιο αντικείμενο καταστρέφεται κατά την έξοδο από την εφαρμογή:

```
master->TermDataSource();
```

## ΠΕΡΙΒΑΛΛΟΝ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ (SESSION)

Αμέσως μετά από την αρχικοποίηση του δείκτη στην πηγή δεδομένων (DataSource) αρχικοποιείται ένας δείκτης σε ένα νέο περιβάλλον αλληλεπίδρασης (Session) το οποίο θα χρησιμοποιείται από την εφαρμογή μέχρι το κλείσιμό της.

Η αρχιτεκτονική του διαχωρισμού των δεικτών στην πηγή δεδομένων και στο περιβάλλον αλληλεπίδρασης επιτρέπει την ταυτόχρονη σύνδεση (αλληλεπίδραση) πολλών χρηστών με την πηγή δεδομένων. Στην εφαρμογή SuiteCase δημιουργείται μόνο ένα αντικείμενο αλληλεπίδρασης κατά τη διάρκεια εργασίας ενός χρήστη.

Οι συναρτήσεις που χρησιμοποιούνται για την αρχικοποίηση και τον τερματισμό του αντικειμένου περιβάλλοντος αλληλεπίδρασης έχουν ως εξής:

- Συνάρτηση αρχικοποίησης:  
master->InitSession();
- Συνάρτηση τερματισμού:  
master->TermSession();

Έχοντας αρχικοποιήσει τον δείκτη στην πηγή δεδομένων και τον δείκτη στο αντικείμενο του περιβάλλοντος αλληλεπίδρασης, η εφαρμογή μπορεί να δημιουργήσει αντικείμενα εντολών με ή χωρίς ανάκτηση δεδομένων.

## ΕΚΤΕΛΕΣΗ ΕΝΤΟΛΩΝ

Στην περίπτωση των εντολών που δεν ανακτούν δεδομένα από τη βάση δεδομένων, ο κώδικας χρησιμοποιεί πάλι το μοτίβο αρχικοποίησης – εκτέλεσης – τερματισμού. Δημιουργεί ένα αντικείμενο τύπου `Command` που προαπαιτεί την ύπαρξη ενός αντικειμένου `Session`.

Ένα αντικείμενο Session έχει τη δυνατότητα υποστήριξης πολλών αντικειμένων εντολών. Ένα αντικείμενο εντολής με τη σειρά του υποστηρίζει τη δημιουργία ενός αντικειμένου κειμένου εντολής (CommandText) η οποία πρόκειται να εκτελεστεί.

```
Ο κώδικας έχει ως εξής:
master->InitCommand();
WCHAR wszCommand[512];
wszCommand = L"use Northwind";
master->ExecuteMyQuery(wszCommand);
master->TermCommand();
```

**Σημείωση:** Το κείμενο της εντολής εισάγεται σε μορφή Wide Char, δηλαδή σε strings όπου ο κάθε χαρακτήρας καταλαμβάνει 2 bytes μνήμης για την αποθήκευσή του.

## ΕΚΤΕΛΕΣΗ ΕΝΤΟΛΩΝ ΑΝΑΚΤΗΣΗΣ ΔΕΔΟΜΕΝΩΝ

Στην περίπτωση των εντολών που ανακτούν δεδομένα από την πηγή δεδομένων τα πράγματα είναι λίγο πιο πολύπλοκα. Δημιουργείται αντικείμενο ειδικά σχεδιασμένο για την ανάκτηση δεδομένων σε μορφή πίνακα δύο διαστάσεων. Η class CMasterOleDb και ο τρόπος με τον οποίο χρησιμοποιείται σε αυτές τις περιπτώσεις αναλαμβάνει τις εξής λειτουργίες / βήματα:

- Δημιουργία αντικειμένου – δείκτη στα ανεκτημένα δεδομένα (Rowset).
- Αναγνώριση των τύπων δεδομένων.
- Φόρτωση των ανεκτημένων δεδομένων σε προσωρινή τοπική μνήμη (buffer).
- Μετατροπή των δεδομένων σε κείμενο, με σκοπό την προβολή.
- Εμφάνιση δεδομένων
- Καταστροφή του αντικειμένου – δείκτη στα ανεκτημένα δεδομένα.

Το αντικείμενο - δείκτης στα ανεκτημένα δεδομένα, είναι και αυτό ένα αντικείμενο IUnknown. Η αρχικοποίησή του προκύπτει από την εκτέλεση μιας εντολής μέσω ενός αντικειμένου εντολής (Command).

Η αναγνώριση των τύπων δεδομένων περιλαμβάνει την εκμετάλλευση του αντικειμένου Rowset για ανάκτηση διευκρινιστικών πληροφοριών σχετικά με το πλήθος των στηλών και των γραμμών που αποτελούν τον πίνακα των ανεκτημένων δεδομένων και σχετικά με τους τύπους δεδομένων. Η διαδικασία περιλαμβάνει δύο στάδια:

- Εγκατάσταση συνδέσεων με τα ανεκτημένα δεδομένα (Bindings). Περιλαμβάνει την ανάκτηση των αριθμών που αναπαριστούν το πλήθος των γραμμών και των στηλών και πληροφοριών σχετικά με τους τύπους δεδομένων και τη μετατροπή τους σε κείμενο.
- Ομαδοποίηση των συνδέσεων που ανταποκρίνονται στον πίνακα των ανεκτημένων δεδομένων. Δημιουργία αντικειμένου Accessor.

Τα δεδομένα αντιγράφονται από την πηγή δεδομένων στην τοπική προσωρινή μνήμη (buffer) με περιορισμούς γραμμής έναρξης (offset) και πλήθους γραμμών.

Η μετατροπή των τύπων δεδομένων στα ανεκτημένα δεδομένα γίνεται με τον έλεγχο των πληροφοριών που παρέχουν οι σύνδεσμοι, κατά τη διάρκεια αίτησης προβολής ενός συγκεκριμένου πεδίου του πίνακα.. Τα δεδομένα μετατρέπονται τελικά σε κείμενο Wide Char.

Ο προγραμματιστής – χρήστης της class CMasterOledb βλέπει τα δεδομένα που ανακτούν οι εντολές του σε μορφή κειμένου, συνήθως διαμορφωμένα μέσα σε ένα στοιχείο ελέγχου κελιών που σχηματίζουν πίνακες δύο διαστάσεων.

Εφ’ όσον ολοκληρωθεί η διαδικασία, τερματίζεται ο δείκτης – αντικείμενο στα αποτελέσματα της εντολής ανάκτησης δεδομένων που εκτελέστηκε. Τέλος, είναι πολύ σημαντικό να τονιστεί ότι πρέπει να εκκενωθεί και το τμήμα της τοπικής μνήμης που χρησιμοποιήθηκε ως προσωρινός αποθηκευτικός χώρος για τα ανεκτημένα δεδομένα.

Ο κώδικας έχει ως εξής:

```
WCHAR wszSelectCommand[512];
wszSelectCommand = L“select * from MyTable”;
master->InitRowset(wszSelectCommand);
master->FillTheRowsetBuffer(0,1000);
WCHAR wszData[256];
for (i=0; i<master->GetRowNum(); i++)
    for (j=0; j<master->GetColNum(); j++)
        master->GetMyString(i,j,wszData);
master->EmptyTheRowsetBuffer();
master->TermRowset();
```

## ΑΝΑΚΤΗΣΗ ΔΕΔΟΜΕΝΩΝ ΔΟΜΗΣ ΣΧΗΜΑΤΟΣ

Κατά την ανάπτυξη της εφαρμογής υπήρξε μια επιπλέον ανάγκη για ανάκτηση δεδομένων σε μορφή πίνακα. Η ειδική αυτή περίπτωση συναντάται όταν η εφαρμογή θέλει να ανακτήσει μια λίστα με τα διαθέσιμα αντικείμενα της βάσης δεδομένων, οποιουδήποτε τύπου, και με τις ιδιότητες / χαρακτηριστικά τους.

Αυτά τα δεδομένα ονομάζονται δεδομένα δομής σχήματος (metadata) και σε όλα τα προϊόντα βάσεων δεδομένων των κατασκευαστικών εταιριών αποθηκεύονται στην ίδια την βάση δεδομένων. Στον τόπο αποθήκευσης υπάρχει ομοιογένεια μεταξύ των διαφορετικών συστημάτων βάσεων δεδομένων. Ο τρόπος, όμως, αποθήκευσης και ανάκτησης των δεδομένων αυτών διαφέρει πολλές φορές ριζικά λόγω της διαφοροποίησης των υλοποιήσεων του σχεσιακού μοντέλου (relational model).

Η βιβλιοθήκη OLE DB, παρέχει μια ενιαία μέθοδο ανάκτησης πληροφοριών δομής σχήματος για διάφορα είδη βάσεων δεδομένων. Αυτός ο τρόπος περιλαμβάνει την χρήση ενός ιδιαίτερου δείκτη σε ανεκτημένα δεδομένα (Rowset). Αυτός ο δείκτης – αντικείμενο ονομάζεται δείκτης σε πίνακα δεδομένων σχήματος (Schema Rowset) και λειτουργεί με παρόμοιο τρόπο, όπως ο δείκτης – αντικείμενο σε ανεκτημένα δεδομένα πίνακα.

Ιδιαιτερότητα του αντικειμένου – δείκτη σε πίνακα ανεκτημένων δεδομένων δομής σχήματος αποτελεί η χρήση κωδικών (αντί για εντολές SQL) που προσδιορίζουν το είδος των δεδομένων δομής που πρέπει να ανακτηθούν κάθε φορά.

Το αντικείμενο – δείκτης σε πίνακα ανεκτημένων δεδομένων δομής σχήματος είναι διαφορετικό από το αντικείμενο – δείκτη σε πίνακα ανεκτημένων δεδομένων,

αλλά χρησιμοποιεί το ίδιο τμήμα της τοπικής προσωρινής μνήμης (buffer) στην υλοποίησή του από την class CMasterOledb.

Οι συναρτήσεις πρόσβασης στα δεδομένα δομής σχήματος διαφέρουν ως προς την ονοματολογία τους αλλά και ελάχιστα ως προς την λειτουργικότητά τους. Έτσι ο κώδικας ανάκτησης δεδομένων δομής σχήματος έχει ως εξής:

```

master->InitSchemaRowset(DBSCHEMA_TABLES);
master->FillTheSchemaRowsetBuffer(0,1000);
WCHAR wszData[256];
for (i=0; i<master->GetRowNum(); i++)
    for (j=0; j<master->GetColNum(); j++)
        master->GetMyString(i,j,wszData);
master->EmptyTheSchemaRowsetBuffer();
master->TermSchemaRowset();

```

### ΜΗΝΥΜΑΤΑ ΣΦΑΛΜΑΤΩΝ

Για την ανάκτηση των μηνυμάτων σφαλμάτων, η class CMasterOledb χρησιμοποιεί ένα σύστημα συναρτήσεων. Σύμφωνα με την λειτουργία της βιβλιοθήκης OLE DB, υπάρχει ένα συγκεκριμένο σημείο της μνήμης όπου αποθηκεύονται τα μηνύματα λάθους. Η class CMasterOledb διαθέτει μια συνάρτηση ανάκτησης των μηνυμάτων σφαλμάτων, και βοηθητικές συναρτήσεις για την εμφάνισή τους. Ενδεικτικά μια διαδικασία ανάκτησης μηνυμάτων σφαλμάτων έχει ως εξής:

```

master->ErrorExamination();
BSTR bstrSource;
BSTR bstrDescription;
for (i=0; i<master->GetErrorCount(); i++)
{
    bstrSource = master->GetErrorSource(i);
    bstrDescription = master->GetErrorDescription(i);
}

```

### ΒΟΗΘΗΤΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

Η class CMasterOledb διαθέτει και κάποιες συναρτήσεις βοηθητικής λειτουργίας στην ανάκτηση δεδομένων. Αυτές οι συναρτήσεις υλοποιούν την αναζήτηση και την επιστροφή ονομάτων στηλών.

Η συνάρτηση myFindRowsetColumnIndex και η συνάρτηση myFindSchemaRowsetColumnIndex παίρνουν ως είσοδο το όνομα μιας στήλης που γνωρίζουμε ότι υπάρχει στον πίνακα των ανεκτημένων δεδομένων και επιστρέφει τον αριθμό που αναπαριστά τη θέση της στήλης στον πίνακα. Ο κώδικας έχει ως εξής:

```

WCHAR wszColumnName = L"Surname";
LONG lIndex;
master->myFindRowsetColumnIndex(wszColumnName, &lIndex);

```



Η συνάρτηση `GetRowsetColumnName` και η συνάρτηση `GetSchemaRowsetColumnName` παίρνουν ως είσοδο τον αριθμό που αναπαριστά την θέση μιας στήλης στον πίνακα και επιστρέφουν το όνομα αυτής της στήλης. Ο κώδικας έχει ως εξής:

```
WCHAR wszColumnName = master->GetRowsetColumnName(3);
```

## 15. ΕΡΓΑΛΕΙΟ DATAVIEW

### ΕΙΣΑΓΩΓΗ

Το πρώτο εργαλείο που παρέχει η εφαρμογή SuiteCase είναι το DataView. Το DataView εργαλείο αποτελείται από ένα παράθυρο που χωρίζεται σε δύο στήλες.

Η αριστερή στήλη του παράθυρου περιλαμβάνει ένα στοιχείο ελέγχου αναδυόμενης λίστας (Combo Box) για την επιλογή της βάσης δεδομένων στην οποία επιθυμεί να εργαστεί ο χρήστης, και ένα στοιχείο ελέγχου καρτελών (Tab Control) για την εμφάνιση των ονομάτων των αντικειμένων μιας βάσης δεδομένων σε ένα στοιχείο ελέγχου λίστας (List Control).

Η δεξιά στήλη του παραθύρου περιλαμβάνει μια μπάρα εργαλείων (Tool Bar) που πραγματοποιεί την ταξινόμηση και το φιλτράρισμα των δεδομένων, καθώς επίσης και την δημιουργία στηλών και δεικτών και την εισαγωγή, ενημέρωση και διαγραφή δεδομένων. Επίσης, περιλαμβάνει δύο πλήκτρα για την δημιουργία αναφορών και φορμών εισαγωγής δεδομένων. Το μεγαλύτερο μέρος, όμως, της δεξιάς στήλης του παράθυρου καλύπτεται από ένα στοιχείο ελέγχου καρτελών που αναλαμβάνει να εμφανίσει σε ένα στοιχείο ελέγχου κελιών (Grid Control) τις ιδιότητες ενός πίνακα ή μιας όψης.

The screenshot shows the SuiteCase DataView application window. On the left, there is a navigation pane with a dropdown menu set to 'Northwind'. Below it are tabs for 'System Tables' and 'System Views', with 'Tables' selected. A list of tables is visible, including Categories, Customer, Customers, Employees, Orders, Products, Region, Shippers, Suppliers, Territories, and departments. The main area displays a table with the following columns: ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, UnitsInStock, UnitsOnOrder, ReorderLevel, and Discontinued. The table contains 46 rows of product data.

ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel	Discontinued
1	Chai	1	1	10 boxes x ...	18	39	0	10	False
2	Chang	1	1	24 - 12 oz b...	19	17	40	25	False
3	Aniseed Syrup	1	2	12 - 550 ml ...	10	13	70	25	False
4	Chef Anton's...	2	2	48 - 6 oz jars	22	53	0	0	False
5	Chef Anton's...	2	2	36 boxes	21.35	0	0	0	True
6	Grandma's B...	3	2	12 - 8 oz jars	25	120	0	25	False
7	Uncle Bob's ...	3	7	12 - 1 lb pkgs.	30	15	0	10	False
8	Northwoods...	3	2	12 - 12 oz jars	40	6	0	0	False
9	Mishi Kobe ...	4	6	18 - 500 g p...	97	29	0	0	True
10	Ibura	4	8	12 - 200 ml j...	31	31	0	0	False
11	Queso Cabr...	5	4	1 kg pkg.	21	22	30	30	False
12	Queso Manc...	5	4	10 - 500 g p...	38	86	0	0	False
13	Konbu	6	8	2 kg box	6	24	0	5	False
14	Tofu	6	7	40 - 100 g p...	23.25	35	0	0	False
15	Genen Shouyu	6	2	24 - 250 ml ...	15.5	39	0	5	False
16	Pavlova	7	3	32 - 500 g b...	17.45	29	0	10	False
17	Alice Mutton	7	6	20 - 1 kg tins	39	0	0	0	True
18	Carnarvon ...	7	8	16 kg pkg.	62.5	42	0	0	False
19	Teatime Ch...	8	3	10 boxes x ...	9.2	25	0	5	False
20	Sir Rodney's...	8	3	30 gift boxes	81	40	0	0	False
21	Sir Rodney's...	8	3	24 pkgs. x 4...	10	3	40	5	False
22	Gustaf's Kn...	9	5	24 - 500 g p...	21	104	0	25	False
23	Tunnbrod	9	5	12 - 250 g p...	9	61	0	25	False
24	Guarana Fa...	10	1	12 - 355 ml ...	4.5	20	0	0	True
25	MulluCa Nu...	11	3	20 - 450 g g...	14	76	0	30	False
26	Gumbur Gu...	11	3	100 - 250 g ...	31.23	15	0	0	False
27	Schoggi Sch...	11	3	100 - 100 g ...	43.9	49	0	30	False
28	Rossle Saue...	12	7	25 - 825 x ...	45.6	26	0	0	True
29	Thuringer R...	12	6	50 bags x 3...	123.79	0	0	0	True
30	Nord-Ost M...	13	8	10 - 200 g p...	25.89	10	0	15	False
31	Gorgonzola ...	14	4	12 - 100 g p...	12.5	0	70	20	False
32	Mascarpone...	14	4	24 - 200 g p...	32	9	40	25	False
33	Geitost	15	4	500 g	2.5	112	0	20	False
34	Sasquatch Ale	16	1	24 - 12 oz b...	14	111	0	15	False
35	Steeleye Stout	16	1	24 - 12 oz b...	18	20	0	15	False
36	Inlagd Sil	17	8	24 - 250 g j...	19	112	0	20	False
37	Gravend lax	17	8	12 - 500 g p...	26	11	50	25	False
38	Cote de Blaye	18	1	12 - 75 cl bo...	263.5	17	0	15	False
39	Chartreuse ...	18	1	750 cc per b...	18	69	0	5	False
40	Boston Crab...	19	8	24 - 4 oz tins	18.4	123	0	30	False
41	Jack's New ...	19	8	12 - 12 oz c...	9.65	85	0	10	False
42	Singaporea...	20	5	32 - 1 kg pkgs.	14	26	0	0	True
43	Ispoh Coffee	20	1	16 - 500 g tins	46	17	10	25	False
44	Gula Malacca	20	2	20 - 2 kg bags	19.45	27	0	15	False
45	Rogede sild	21	8	1k pkg.	9.5	5	70	15	False
46	Corvado	21	8	4 - 450 ml	19	66	0	0	False

## ΕΠΙΛΟΓΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Η επιλογή βάσης δεδομένων στην εφαρμογή SuiteCase είναι ένα χαρακτηριστικό που χρησιμοποιείται κατά κόρον και προσφέρει ένα μεγάλο μέρος της ευελιξίας της εφαρμογής. Στο εργαλείο DataView, η επιλογή βάσης δεδομένων επιτρέπει στον χρήστη της εφαρμογής SuiteCase να εργαστεί με όλες τις βάσεις δεδομένων στις οποίες έχει πρόσβαση.

Για την επιλογή της βάσης δεδομένων χρησιμοποιείται ένα αντικείμενο – δείκτης σε πίνακα δεδομένων δομής (Schema Rowset) για την ανάκτηση των ονομάτων των βάσεων δεδομένων που φιλοξενεί ένας Server. Για τον εντοπισμό της στήλης του πίνακα που περιέχει το όνομα της βάσης χρησιμοποιείται η συνάρτηση **myFindSchemaRowsetColumnIndex**. Τα ονόματα των βάσεων δεδομένων γεμίζουν την λίστα του στοιχείου ελέγχου αναδυόμενης λίστας. Ο κώδικας έχει ως εξής:

```
CComboBox cbCatalogs;
// ...
// ...
master->InitSchemaRowset(DBSCHEMA_CATALOGS);
master->FillTheSchemaRowsetBuffer(0,100);
LONG lIndex;
master->myFindSchemaRowsetColumnIndex(L"CATALOG_NAME", &lIndex);
WCHAR wszData[256];
char chData[256];
for (I=0; I<master->GetRowNum(); I++)
{
    master->GetMyString(I,lIndex,wszData);
    GiveMeString(wszData, chData);
    CbCatalogs.AddString(chData);
}
master->EmptyTheSchemaRowsetBuffer();
master->TermSchemaRowset();
```

Η επιλογή μιας βάσης δεδομένων από τον χρήστη επάνω στο στοιχείο ελέγχου αναδυόμενης λίστας πυροδοτεί το γεγονός CBN\_SELCHANGE της βιβλιοθήκης MFC. Με την μακροεντολή ON\_CBN\_SELCHANGE ορίζουμε την συνάρτηση που θα εκτελείται κάθε φορά που πυροδοτείται το συγκεκριμένο γεγονός, για το συγκεκριμένο στοιχείο ελέγχου. Η βιβλιοθήκη MFC προσδιορίζει χάρτη μηνυμάτων συμβάντων, στον οποίο μπορούμε να προσθέσουμε την μακροεντολή:

```
BEGIN_MESSAGE_MAP(CLeftFrame, CFrameWnd)
    ON_CBN_SELCHANGE(IDC_COMBO_SCHEMATA, RefreshObjects)
END_MESSAGE_MAP()
```

Από εδώ και στο εξής η συνάρτηση που θα καλείται σε κάθε πυροδότηση του συμβάντος CBN\_SELCHANGE για το συγκεκριμένο στοιχείο ελέγχου αναδυόμενης λίστας, θα είναι η συνάρτηση **RefreshObjects**. Αυτή η συνάρτηση αναλαμβάνει να

εκτελέσει την εντολή SQL: “use <DataBaseName>” και να δώσει εντολή ανανέωσης της λίστας των εμφανιζόμενων αντικειμένων.

### ΕΠΙΛΟΓΗ ΑΝΤΙΚΕΙΜΕΝΟΥ

Το στοιχείο ελέγχου καρτέλας της αριστερής στήλης του παράθυρου του εργαλείου DataView αναλαμβάνει να φιλτράρει τα ονόματα των αντικειμένων που θα εμφανίζονται στο στοιχείο ελέγχου λίστας που περιλαμβάνει, ανάλογα με τον τύπο τους.

Η ανάκτηση των ονομάτων των αντικειμένων της βάσης δεδομένων πραγματοποιείται με την συνάρτηση:

```
master->InitSchemaRowset(DBSCHEMA_TABLES);
```

Αυτή η συνάρτηση ανακτά έναν πίνακα δεδομένων δομής που περιλαμβάνει τα ονόματα και τα χαρακτηριστικά όλων των πινάκων και των όψεων της βάσης δεδομένων. Με τη βοήθεια της συνάρτησης:

```
master->myFindSchemaRowsetColumnIndex(L"TABLE_TYPE", &lIndex);
```

Καταφέρει να εντοπίσει τον τύπο του αντικειμένου στον πίνακα, που μπορεί να είναι: “TABLE”, “VIEW”, “SYSTEM TABLE”, “SYSTEM VIEW”.

ΜΕ τη βοήθεια της συνάρτησης:

```
master->myFindSchemaRowsetColumnIndex(L"TABLE_NAME", &lIndex);
```

Καταφέρει να εντοπίσει το όνομα του αντικειμένου που θα εμφανίσει τελικά στην λίστα του στοιχείου ελέγχου λίστας.

Κάθε φορά που αλλάζουμε καρτέλα στο στοιχείο ελέγχου καρτελών πυροδοτείται το γεγονός TCN\_SELCHANGE. Στον χάρτη μηνυμάτων συμβάντων προσθέτουμε την μακροεντολή:

```
ON_NOTIFY_REFLECT(TCN_SELCHANGE, OnSelchange)
```

Το μήνυμα TCN\_SELCHANGE παγιδεύεται από το ίδιο το αντικείμενο του στοιχείου ελέγχου καρτελών, και εκτελείται η συνάρτηση μέλος της class: OnSelchange. Ανάλογα με την καρτέλα η οποία επιλέχτηκε, αλλάζει ο τύπος των αντικειμένων που αναζητούνται στον πίνακα ανακτημένων δεδομένων δομής για να εμφανιστούν τα ονόματά τους στο στοιχείο ελέγχου λίστας.

Η επιλογή ενός αντικειμένου στην αριστερή στήλη του παραθύρου του εργαλείου DataView προκαλεί την εμφάνιση των δεδομένων στην δεξιά στήλη του παραθύρου. Το στοιχείο ελέγχου λίστας που παραθέτει τα ονόματα των αντικειμένων δέχεται ως αντικείμενο την πυροδότηση του γεγονότος NM\_DBLCLK, που συμβαίνει με το διπλό click του χρήστη, επάνω στο όνομα ενός αντικειμένου. Ο χάρτης μηνυμάτων συμβάντων περιλαμβάνει την μακροεντολή:

```
ON_NOTIFY_REFLECT(NM_DBLCLK, OnDbclick)
```

Με την πυροδότηση του event NM\_DBLCLK καλείται η συνάρτηση OnDbclk που αναλαμβάνει να προκαλέσει την εμφάνιση των δεδομένων του συγκεκριμένου αντικειμένου στην δεξιά στήλη του εργαλείου DataView.

## ΕΜΦΑΝΙΣΗ ΔΕΔΟΜΕΝΩΝ

Το κυρίως τμήμα της δεξιάς στήλης του παραθύρου του εργαλείου DataView καταλαμβάνεται από ένα στοιχείο ελέγχου κελιών, το οποίο φιλοξενείται και ελέγχεται από ένα στοιχείο ελέγχου καρτελών.

Γενικά, η δεξιά στήλη του παραθύρου του εργαλείου DataView αναλαμβάνει να εμφανίσει τις ιδιότητες του αντικειμένου που επιλέγεται στην αριστερή στήλη του παραθύρου. Τα είδη των πληροφοριών που εμφανίζονται είναι: τα δεδομένα που περιέχει ο πίνακας ή η όψη, τα χαρακτηριστικά των στηλών και τα χαρακτηριστικά των αντικειμένων δεικτών.

Η δεξιά στήλη του παραθύρου του εργαλείου DataView παίρνει ως είσοδο το όνομα ενός αντικειμένου και ανάλογα με την καρτέλα που είναι ενεργοποιημένη εμφανίζει τα δεδομένα, τις ιδιότητες των στηλών ή τους δείκτες που ανήκουν στο συγκεκριμένο αντικείμενο.

Το event που πυροδοτείται και σε αυτό το στοιχείο ελέγχου καρτελών είναι το TCN\_SELCHANGE. Ο χάρτης μηνυμάτων συμβάντων περιλαμβάνει την μακροεντολή:

```
ON_NOTIFY_REFLECT(TCN_SELCHANGE, OnSelchange)
```

Η συνάρτηση OnSelchange ελέγχει ποια καρτέλα του στοιχείου ελέγχου είναι επιλεγμένη και ανάλογα καλεί μια από τις συναρτήσεις: ShowData(), ShowColumns() και ShowIndexes().

Η συνάρτηση ShowData() εκτελεί στην βάση δεδομένων μια εντολή ανάκτησης όλων των δεδομένων, χρησιμοποιώντας το όνομα του αντικειμένου που επιλέχτηκε.

Η συνάρτηση ShowColumns() αρχικοποιεί έναν δείκτη σε πίνακα δεδομένων δομής με τον κωδικό DBSCHEMA\_COLUMNS και χρησιμοποιώντας τη συνάρτηση της class CMasterOledb *myFindSchemaRowsetColumnIndex(L"TABLE\_NAME", &Index)* επιλέγει τα ονόματα των στηλών και τις ιδιότητές τους που ανήκουν στον συγκεκριμένο πίνακα ή όψη.

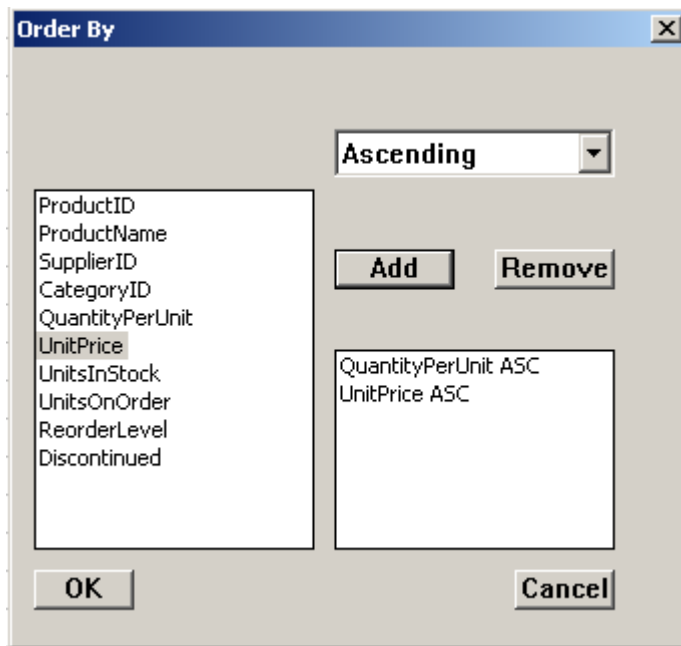
Η συνάρτηση ShowIndexes() λειτουργεί κατά τον ίδιο τρόπο με την συνάρτηση ShowColumns(), αλλά εδώ ο δείκτης σε πίνακα δεδομένων δομής αρχικοποιείται με τον κωδικό DBSCHEMA\_INDEXES.

## ΤΑΞΙΝΟΜΗΣΗ ΚΑΙ ΦΙΛΤΡΑΡΙΣΜΑ

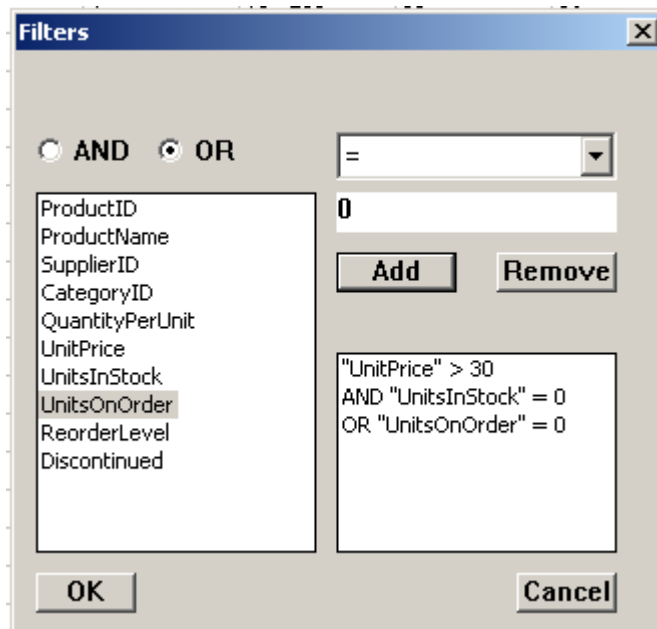
Η δυνατότητα ταξινόμησης και φιλτραρίσματος των δεδομένων στη δεξιά στήλη του παραθύρου του εργαλείου DataView παρέχεται από τα δύο πρώτα πλήκτρα της μπάρας εργαλείων (Tool Bar). Κάθε ένα από αυτά τα πλήκτρα εμφανίζει ένα παράθυρο διαλόγου το οποίο επιτρέπει στον χρήστη να επιλέξει μια στήλη του ενεργού πίνακα ή της ενεργής όψης και να ορίσει μια λειτουργία ή έναν έλεγχο που πρέπει να εκτελεστεί με αυτή.

Το παράθυρο διαλόγου ταξινόμησης επιτρέπει στον χρήστη να επιλέξει στήλες του πίνακα και έναν τρόπο ταξινόμησης (ascending/descending) για την κάθε μία.

Από τις επιλογές του χρήστη, το παράθυρο διαλόγου αναλαμβάνει να σχηματίσει ένα string το οποίο θα χρησιμοποιηθεί στη συνέχεια ως παράμετρος στην συνάρτηση “order by” της εντολής ανάκτησης (select), στην συνάρτηση εμφάνισης δεδομένων ShowData().

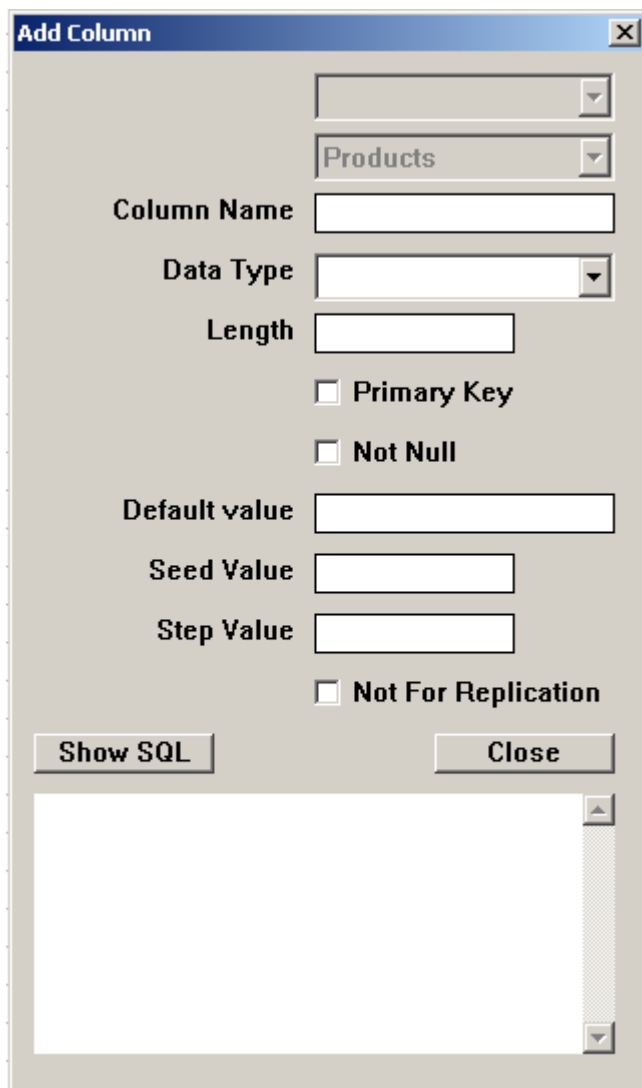


Το παράθυρο διαλόγου φιλτραρίσματος επιτρέπει στον χρήστη να επιλέξει μια στήλη του πίνακα, έναν συντελεστή και μια τιμή, στοιχεία τα οποία χρησιμεύουν στην συνέχεια για να περαστούν στην εντολή ανάκτησης δεδομένων ως string παραμέτρων για τα κριτήρια μετά από την λέξη – κλειδί “where”.



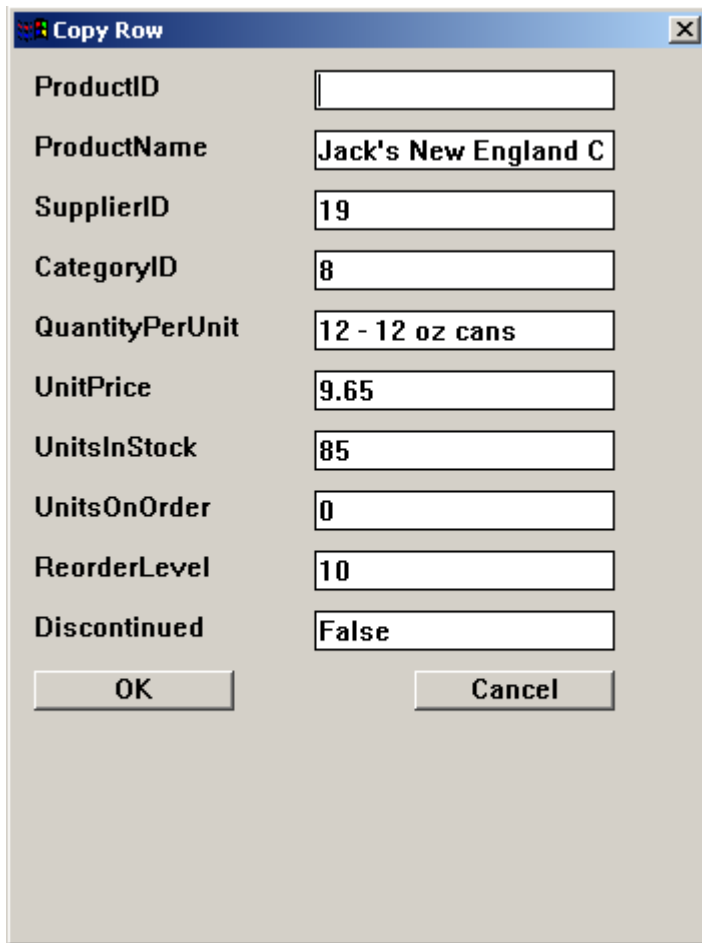
## ΕΙΣΑΓΩΓΗ, ΕΝΗΜΕΡΩΣΗ ΚΑΙ ΑΝΤΙΓΡΑΦΗ ΓΡΑΜΜΗΣ

Για την εισαγωγή, ενημέρωση και αντιγραφή γραμμής δεδομένων χρησιμοποιείται ένα παράθυρο διαλόγου το οποίο δημιουργείται κάθε φορά με τα ονόματα των στηλών και ισάριθμα πλαίσια εισαγωγής κειμένου. Στην περίπτωση της εισαγωγής και της αντιγραφής γραμμής δεδομένων, τα στοιχεία που εισάγονται στο παράθυρο από τον χρήστη, χρησιμοποιούνται για την δημιουργία του αντίστοιχου string εντολής “Insert” στον ενεργό πίνακα. Στην περίπτωση της ενημέρωσης γραμμής δεδομένων τα στοιχεία που εισάγονται στο παράθυρο από τον χρήστη, χρησιμοποιούνται για την δημιουργία του αντίστοιχου string εντολής “Update” στον ενεργό πίνακα.



Στις περιπτώσεις της αντιγραφής και της ενημέρωσης σε γραμμή δεδομένων, το παράθυρο διαλόγου εμφανίζεται με αρχικοποιημένα τα πλαίσια εισαγωγής κειμένου με τα δεδομένα της γραμμής του στοιχείου ελέγχου κελιών που είναι επιλεγμένη εκείνη τη στιγμή.

Στην περίπτωση της αντιγραφής γραμμής δεδομένων το μόνο πλαίσιο που δεν αρχικοποιείται είναι αυτό που αντιστοιχεί στην στήλη του πρωτεύοντος κλειδιού, καθώς δεν μπορεί να αντιγραφεί στον ίδιο πίνακα.



Field Name	Value
ProductID	
ProductName	Jack's New England C
SupplierID	19
CategoryID	8
QuantityPerUnit	12 - 12 oz cans
UnitPrice	9.65
UnitsInStock	85
UnitsOnOrder	0
ReorderLevel	10
Discontinued	False

Στην περίπτωση της ενημέρωσης γραμμής δεδομένων το πλαίσιο κειμένου που αντιστοιχεί στη στήλη του πρωτεύοντος κλειδιού απενεργοποιείται ώστε να μην επηρεαστεί από τον χρήστη, εφ' όσον αποτελεί αναγνωριστικό στοιχείο της γραμμής που ενημερώνεται.

Η διαγραφή μιας γραμμής δεδομένων υλοποιείται με τη συνάρτηση DeleteRow η οποία παίρνει ως είσοδο εκτός από το όνομα του πίνακα και τα ονόματα των στηλών, και τα δεδομένα της γραμμής που είναι εκείνη τη στιγμή επιλεγμένη στο στοιχείο ελέγχου κελιών. Τα δεδομένα αυτά χρησιμοποιούνται για να δημιουργηθεί το string με την εντολή "Delete" που στη συνέχεια εκτελείται.

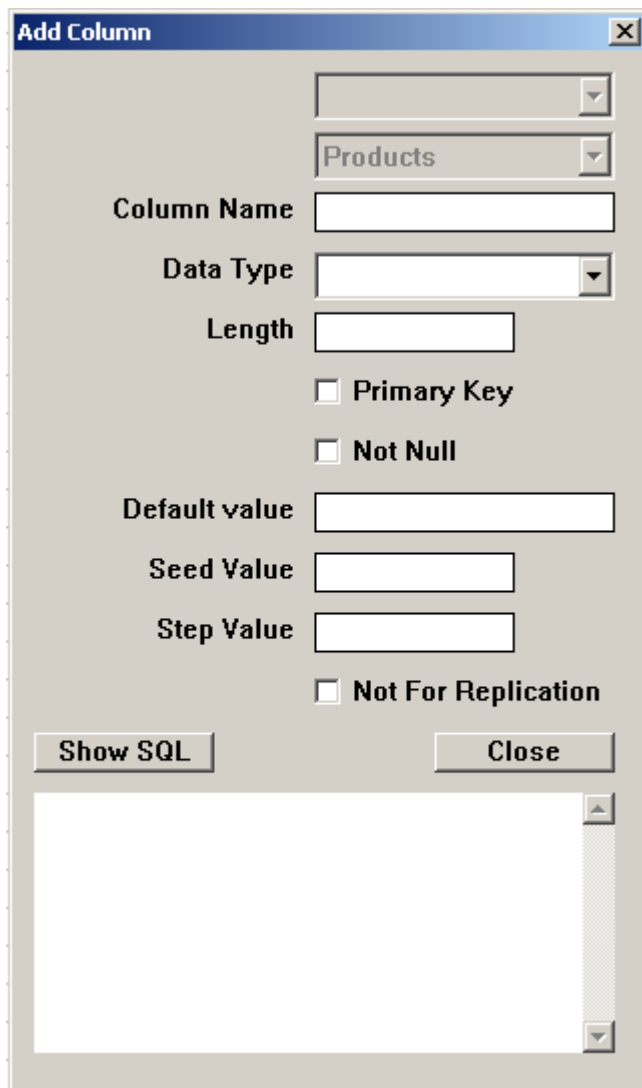
## ΕΙΣΑΓΩΓΗ / ΔΙΑΓΡΑΦΗ ΣΤΗΛΗΣ

Για την εισαγωγή μιας νέας στήλης στον πίνακα εμφανίζεται ένα παράθυρο διαλόγου το οποίο προτρέπει τον χρήστη να προσδιορίσει τα χαρακτηριστικά της νέας στήλης, όπως το όνομά της, τον τύπο δεδομένων, αν είναι πρωτεύον κλειδί κτλ.

Για την διαγραφή μιας στήλης του πίνακα εμφανίζεται ένα παράθυρο διαλόγου που προτρέπει τον χρήστη να επιλέξει την στήλη που επιθυμεί να διαγράψει από τον συγκεκριμένο πίνακα, με το όνομά της από ένα στοιχείο ελέγχου αναδυόμενης

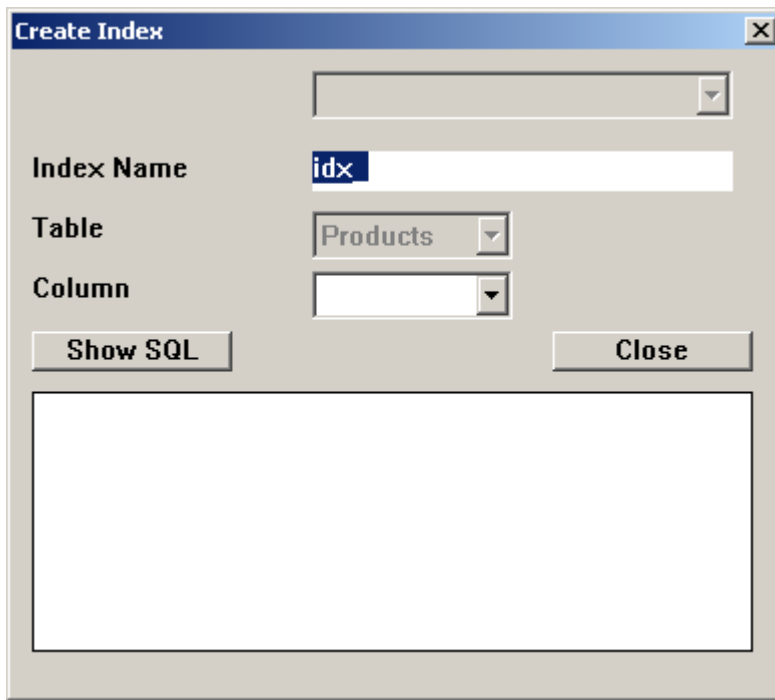


λίστας. Αν έχουμε ήδη επιλεγμένη μια στήλη στην καρτέλα των στηλών στην δεξιά στήλη του παραθύρου του εργαλείου DataView, τότε το παράθυρο εμφανίζεται με προεπιλεγμένη την συγκεκριμένη στήλη.

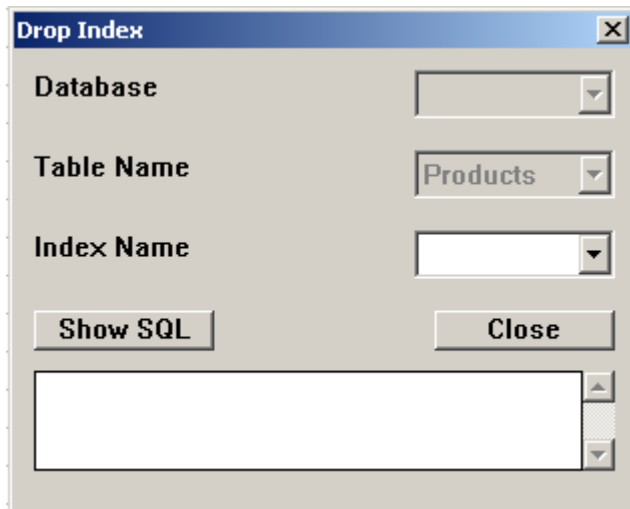


### ΔΗΜΙΟΥΡΓΙΑ / ΔΙΑΓΡΑΦΗ ΔΕΙΚΤΗ

Για την εισαγωγή ενός νέου δείκτη εμφανίζεται ένα παράθυρο διαλόγου το οποίο προτρέπει τον χρήστη να επιλέξει μια στήλη επάνω στην οποία θα δημιουργηθεί ο δείκτης, και ένα όνομα γι' αυτόν.

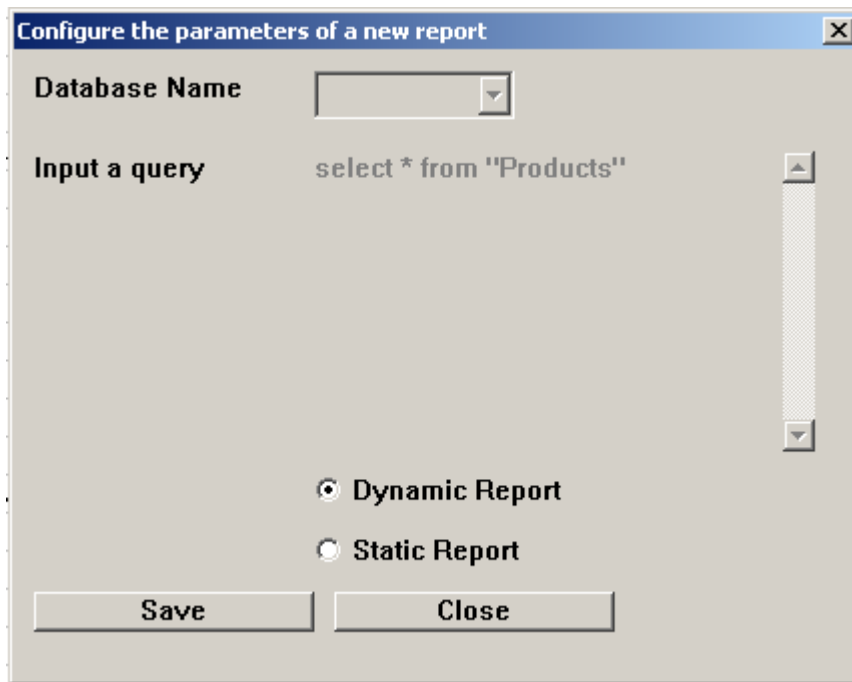


Για την διαγραφή ενός δείκτη εμφανίζεται ένα παράθυρο διαλόγου το οποίο προτρέπει τον χρήστη να επιλέξει έναν δείκτη από ένα στοιχείο ελέγχου αναδυόμενης λίστας.



### ΔΗΜΙΟΥΡΓΙΑ ΑΝΑΦΟΡΩΝ

Το πλήκτρο δημιουργίας αναφοράς εμφανίζει ένα παράθυρο διαλόγου το οποίο προτρέπει τον χρήστη να επιλέξει το είδος της αναφοράς (static / dynamic) και την εντολή ανάκτησης δεδομένων αρχικοποιημένη με την εντολή που χρησιμοποιείται για την ανάκτηση των δεδομένων στο στοιχείο ελέγχου κελιών μετά από ταξινομήσεις και φιλτραρίσματα. Τέλος, ο χρήστης πιέζοντας το πλήκτρο “Save”, έχει την δυνατότητα να επιλέξει τον κατάλογο όπου θα αποθηκευτεί το αρχείο και το όνομά του.



Το πλήκτρο δημιουργίας φορμών εισαγωγής δεδομένων εμφανίζει ένα παράθυρο διαλόγου αρχικοποιημένο με τα ονόματα των στηλών του ενεργού πίνακα και με στοιχεία ελέγχου αναδυόμενης λίστας για την επιλογή του πίνακα και της στήλης στην οποία είναι πιθανό να δείχνει μια στήλη του πίνακα ως ξένο κλειδί (foreign key). Υπάρχει ένα επιπλέον στοιχείο ελέγχου αναδυόμενης λίστας στο οποίο ο χρήστης μπορεί να επιλέξει το όνομα της στήλης που θα εμφανίζεται αντί για την στήλη που ανταποκρίνεται στο ξένο κλειδί.

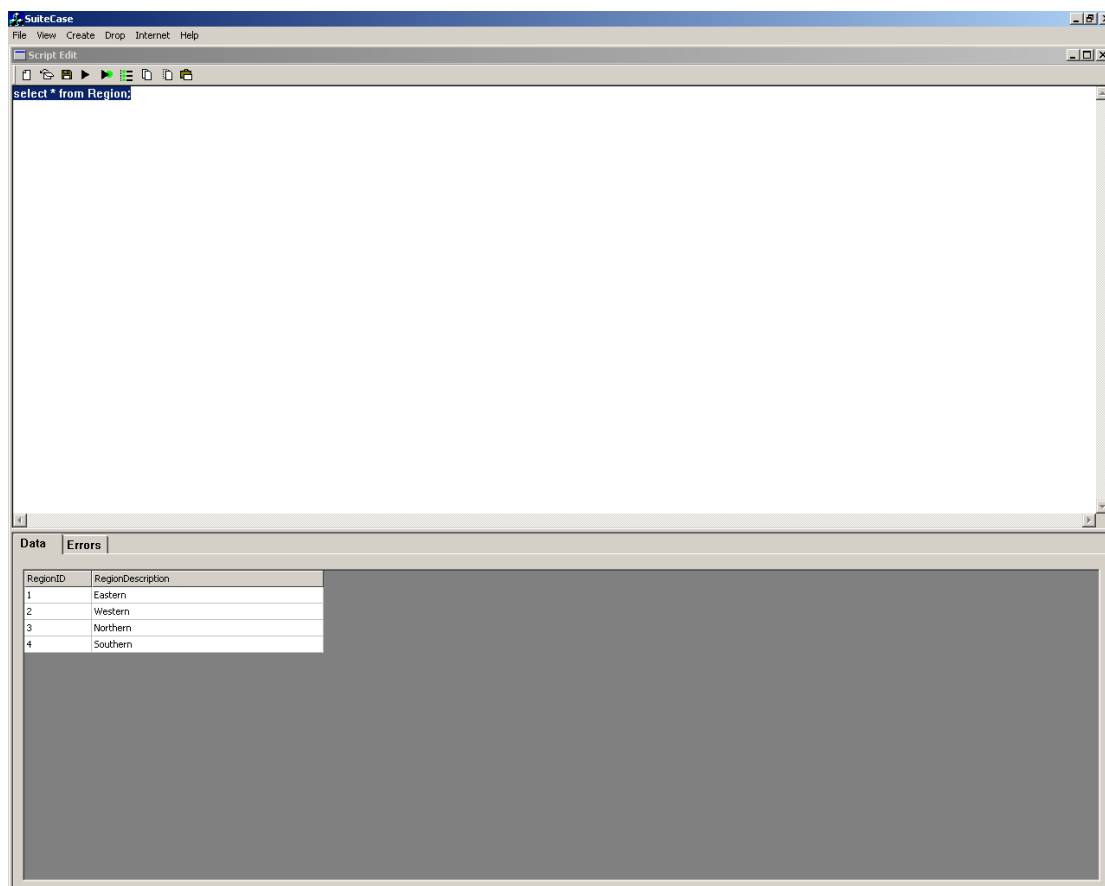
## 16. ΕΡΓΑΛΕΙΟ SCRIPTEDIT

### ΕΙΣΑΓΩΓΗ

Το δεύτερο εργαλείο που παρέχει η εφαρμογή SuiteCase είναι ο κειμενογράφος ScriptEdit. Το εργαλείο αυτό αποτελείται από ένα παράθυρο που χωρίζεται οριζόντια σε δύο γραμμές.

Η επάνω γραμμή είναι ένα στοιχείο ελέγχου επεξεργασίας κειμένου μαζί με μια μπάρα εργαλείων (Tool Bar). Σε αυτό το τμήμα του παραθύρου ο χρήστης μπορεί να γράψει κάποιο σενάριο εντολών (script) SQL ή οποιοδήποτε κείμενο και να το αποθηκεύσει σε αρχεία ή να το εκτελέσει με τρεις τρόπους χρησιμοποιώντας την μπάρα εργαλείων.

Η κάτω γραμμή του παραθύρου περιλαμβάνει ένα στοιχείο ελέγχου καρτελών που αναλαμβάνει να εμφανίσει σε δυο στοιχεία ελέγχου κελιών τα αποτελέσματα και τα μηνύματα λάθους από την εκτέλεση ενός σεναρίου εντολών.



### ΕΠΕΞΕΡΓΑΣΙΑ ΚΑΙ ΕΚΤΕΛΕΣΗ SCRIPTS

Η επάνω γραμμή του παραθύρου του εργαλείου ScriptEdit υλοποιείται με τη χρήση μιας class της βιβλιοθήκης MFC που φιλοξενεί ένα στοιχείο ελέγχου επεξεργασίας κειμένου. Η βιβλιοθήκη MFC παρέχει στο στοιχείο ελέγχου επεξεργασίας κειμένου την λειτουργία αποδοχής κειμένου με πληκτρολόγηση και τη δυνατότητα επιλογής (highlight) τμήματος κειμένου. Ο κώδικας που προστέθηκε σε

αυτή την class περιλαμβάνει τον χειρισμό του ποντικιού (mouse) για την επιλογή κειμένου παγιδεύοντας τα γεγονότα που λαμβάνει το στοιχείο ελέγχου επεξεργασίας κειμένου και έχουν σχέση με το ποντίκι. Ο χάρτης μηνυμάτων συμβάντων περιλαμβάνει τις εξής μακροεντολές:

```
ON_WM_MOUSEMOVE()
ON_WM_LBUTTONDOWN()
ON_WM_LBUTTONUP()
```

Τα μηνύματα των αντίστοιχων συμβάντων είναι τα: WM\_MOUSEMOVE, WM\_LBUTTONDOWN και WM\_LBUTTONUP και οι συναρτήσεις που καλούνται είναι οι: OnMouseMove(), OnLButtonDown() και OnLButtonUp() αντίστοιχα.

Ο κώδικας που προστέθηκε σε αυτές τις συναρτήσεις αποθηκεύει τα σημεία επάνω στο στοιχείο ελέγχου επεξεργασίας κειμένου όπου εκτελέστηκε το κάθε συμβάν και τελικά χρησιμοποιεί αυτά τα δεδομένα για να επιλέξει (highlight) το κείμενο.

#### **Παρατήρηση:**

Η ανάκτηση της γραμμής και της στήλης του κειμένου στα οποία εκτελέστηκε κάθε συμβάν πραγματοποιείται με την συνάρτηση:

```
int CEdit::CharFromPos(CPoint point)
```

όπου point είναι το σημείο του στοιχείου ελέγχου κειμένου, στο οποίο εκτελέστηκε το συμβάν. Ο ακέραιος αριθμός που επιστρέφεται από την συνάρτηση περιγράφει τον αριθμό γραμμής στο πρώτο byte και τον αριθμό στήλης στο δεύτερο byte.

## **Η ΜΠΑΡΑ ΕΡΓΑΛΕΙΩΝ**

Η μπάρα εργαλείων του πάνω παραθύρου του εργαλείου ScriptEdit παρέχει τη δυνατότητα δημιουργίας νέου αρχείου κειμένου, ανοίγματος ενός ήδη υπάρχοντος αρχείου κειμένου και αποθήκευσης του κειμένου που περιλαμβάνεται στο στοιχείο ελέγχου επεξεργασίας κειμένου σε ένα νέο αρχείο. Επίσης, η μπάρα εργαλείων περιλαμβάνει τρία πλήκτρα που υλοποιούν τις εξής λειτουργίες:

- εκτέλεση ολόκληρου του σεναρίου εντολών
- εκτέλεση της εντολής που περιέχεται στο επιλεγμένο κείμενο
- και εκτέλεση ολόκληρου του σεναρίου εντολών με παύση σε κάθε βήμα.

Η λειτουργίες εκτέλεσης των scripts αναγνωρίζουν την αρχή και το τέλος μιας εντολής από τα ερωτηματικά στο τέλος κάθε εντολής.

Η εκτέλεση ολόκληρου του σεναρίου εντολών εκτελεί τις εντολές σαν εντολές που δεν επιστρέφουν δεδομένα ανεξάρτητα από το αν είναι εντολές “Select” ή όχι. Τα ανεκτημένα δεδομένα αποθηκεύονται σε μεταβλητές για να χρησιμοποιηθούν πολλές φορές σε άλλες εντολές εισαγωγής, ενημέρωσης και διαγραφής δεδομένων.

Η εκτέλεση εντολής που περιλαμβάνεται στο επιλεγμένο κείμενο και η εκτέλεση ολόκληρου του σεναρίου εντολών με παύση σε κάθε βήμα, ελέγχουν αν επιστρέφονται δεδομένα από την κάθε εντολή και σε αυτή την περίπτωση γεμίζουν το στοιχείο ελέγχου κελιών της κάτω γραμμής του παράθυρου του εργαλείου ScriptEdit.

Στην εκτέλεση των εντολών ενός σεναρίου με παύσεις σε κάθε βήμα, η εντολή που εκτελέστηκε τελευταία, και για την οποία βλέπουμε τα αποτελέσματα στο στοιχείο ελέγχου κελιών, επιλέγεται ώστε να μπορούμε να εντοπίσουμε την πορεία εκτέλεσης των εντολών του σεναρίου.

Ανεξάρτητα από τη μέθοδο εκτέλεσης, το στοιχείο ελέγχου καρτελών στην κάτω γραμμή του παραθύρου του εργαλείου ScriptEdit διατηρεί ένα επιπλέον στοιχείο ελέγχου κελιών, εκτός αυτού με τα αποτελέσματα, στο οποίο καταχωρούνται οι πηγές και τα μηνύματα των σφαλμάτων που ενδεχομένως προκύπτουν.

Το στοιχείο ελέγχου καρτελών παγιδεύει το event TCN\_SELCHANGE και αναλόγως εμφανίζει το κατάλληλο στοιχείο ελέγχου κελιών.

## 17. ΕΡΓΑΛΕΙΟ QUERYWIZARD

### 17.1 ΕΙΣΑΓΩΓΗ

Το τρίτο εργαλείο που παρέχει η εφαρμογή SuiteCase είναι ο οδηγός δημιουργίας ερωτημάτων QueryWizard. Το εργαλείο αυτό αποτελείται επίσης από ένα παράθυρο χωρισμένο οριζόντια σε δύο γραμμές.

Η επάνω γραμμή του παραθύρου είναι ένα στοιχείο ελέγχου φόρμας το οποίο αναλαμβάνει να εμφανίσει στον χρήστη της επιλογές που είναι διαθέσιμες στα 5 βήματα του οδηγού δημιουργίας ερωτήματος. Η διαδικασία ξεκινάει από την επιλογή των πινάκων που θα πάρουν μέρος στο ερώτημα, συνεχίζει με τον προσδιορισμό των στηλών που θα πρέπει να εμφανίζονται, με τις συνδέσεις στηλών μεταξύ των πινάκων, με τα κριτήρια επιλογής δεδομένων, με τις εντολές ταξινόμησης και ομαδοποίησης και τελικά εμφανίζει το ερώτημα που δημιουργήθηκε με τη μορφή κειμένου εντολής. Ο χρήστης έχει τη δυνατότητα να εκτελέσει την εντολή και να ελέγξει τα επιστρεφόμενα δεδομένα από την εμφάνισή τους στην κάτω γραμμή του παραθύρου του εργαλείου QueryWizard.

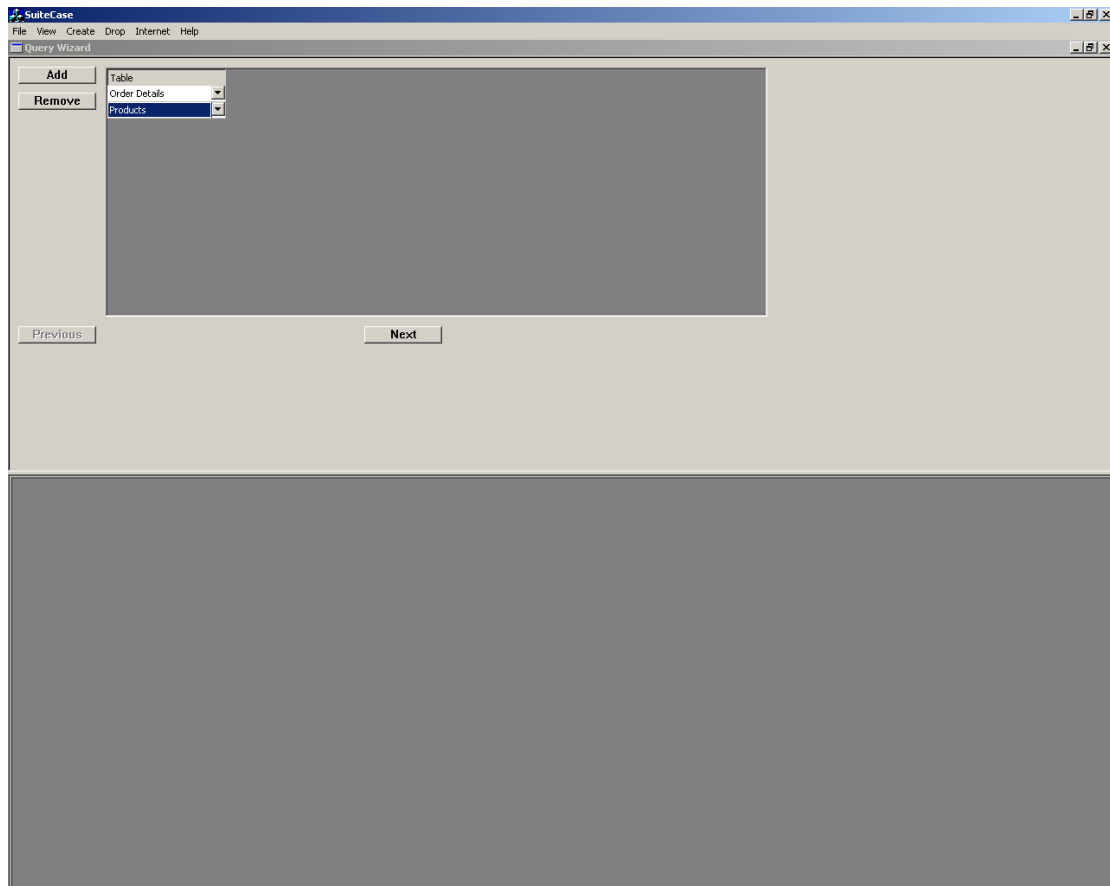
Η κάτω γραμμή του παραθύρου είναι επίσης ένα παράθυρο το οποίο φιλοξενεί ένα στοιχείο ελέγχου κελιών. Αυτό το τμήμα του εργαλείου αναλαμβάνει να εκτελέσει το ερώτημα που δημιουργήθηκε με τη χρήση του οδηγού και να εμφανίσει τα αποτελέσματα στο στοιχείο ελέγχου κελιών.

Για την λειτουργία του οδηγού χρησιμοποιήθηκαν 5 στοιχεία ελέγχου κελιών, ένα για κάθε βήμα, με τις δυνατότητες εισαγωγής (add) και διαγραφής (remove) γραμμών σε κάθε ένα από αυτά.

Τα πλήκτρα που Υπάρχουν σε όλες τις οθόνες του οδηγού είναι τα Add και Remove για την εισαγωγή και διαγραφή, αντίστοιχα, γραμμών. Η λειτουργικότητα του οδηγού υλοποιείται με τα πλήκτρα Previous και Next τα οποία χειρίζονται την εναλλαγή μεταξύ των βημάτων του οδηγού. Στην πραγματικότητα η φόρμα της επάνω στήλης του παραθύρου του εργαλείου QueryWizard περιλαμβάνει κάθε στιγμή δημιουργημένα όλα τα στοιχεία ελέγχου που χρησιμοποιούνται σε όλα τα βήματα του οδηγού. Μια συνάρτηση, η **ShowStep()**, που δέχεται ως παράμετρο έναν ακέραιο αριθμό που αναπαριστά τον αριθμό του βήματος, αναλαμβάνει να εμφανίσει τα στοιχεία ελέγχου που χρησιμοποιεί το εκάστοτε βήμα, και να αποκρύψει τα υπόλοιπα.

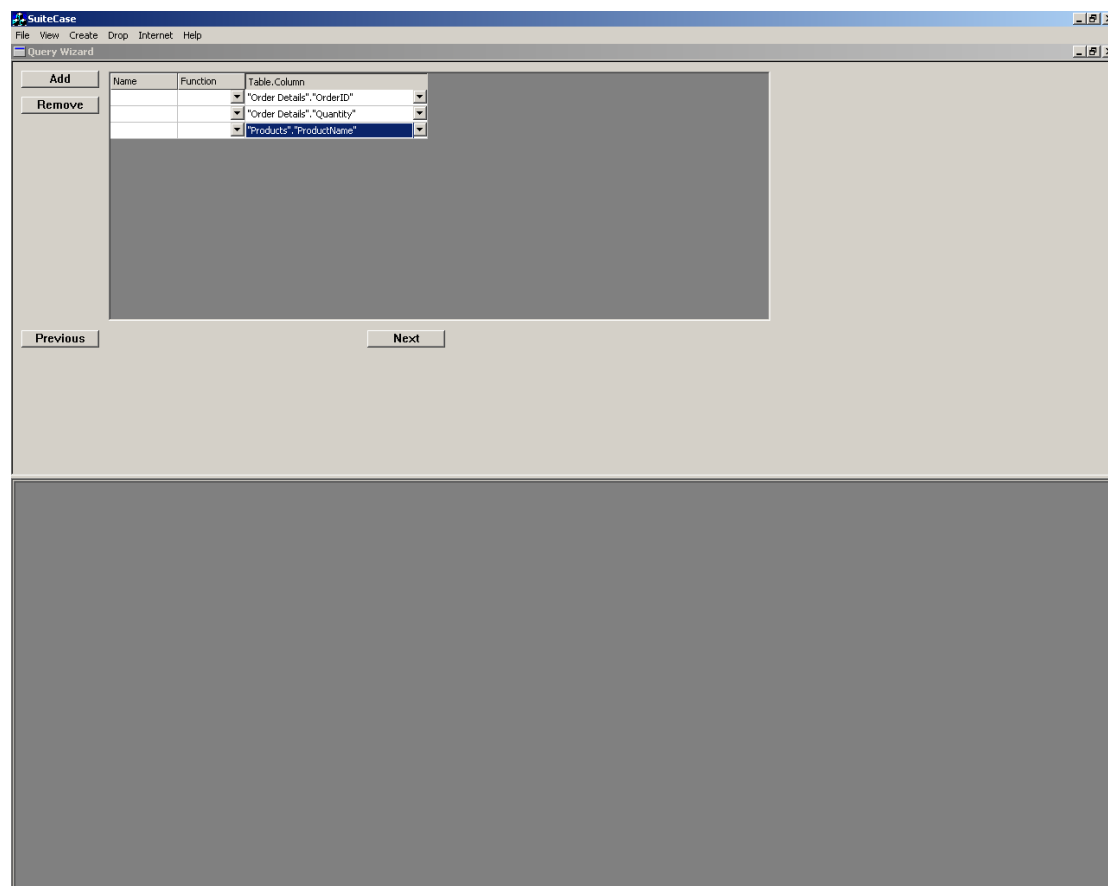
### 17.2 ΤΑ ΒΗΜΑΤΑ ΤΟΥ ΟΔΗΓΟΥ

Στο πρώτο βήμα του οδηγού, το στοιχείο ελέγχου κελιών αποτελείται από μια στήλη, τα κελιά της οποίας αποτελούνται από αναδυόμενες λίστες. Με την προσθήκη μιας γραμμής σε αυτό το στοιχείο ελέγχου κελιών αρχικοποιείται η αναδυόμενη λίστα με τα ονόματα των πινάκων που περιλαμβάνει η ενεργή βάση δεδομένων. Ο χρήστης μπορεί να προσθέσει όσες γραμμές επιθυμεί, επιλέγοντας σε κάθε μία από αυτές το όνομα ενός πίνακα. Στο τέλος του βήματος, οι πίνακες που έχουν επιλεγεί θα χρησιμοποιηθούν για την αρχικοποίηση των στοιχείων ελέγχου αναδυόμενης λίστας στα επόμενα βήματα του οδηγού. Επίσης, αυτοί οι πίνακες είναι που θα περιληφθούν στο τμήμα “from” της εντολής “Select” που τελικά θα προκύψει.

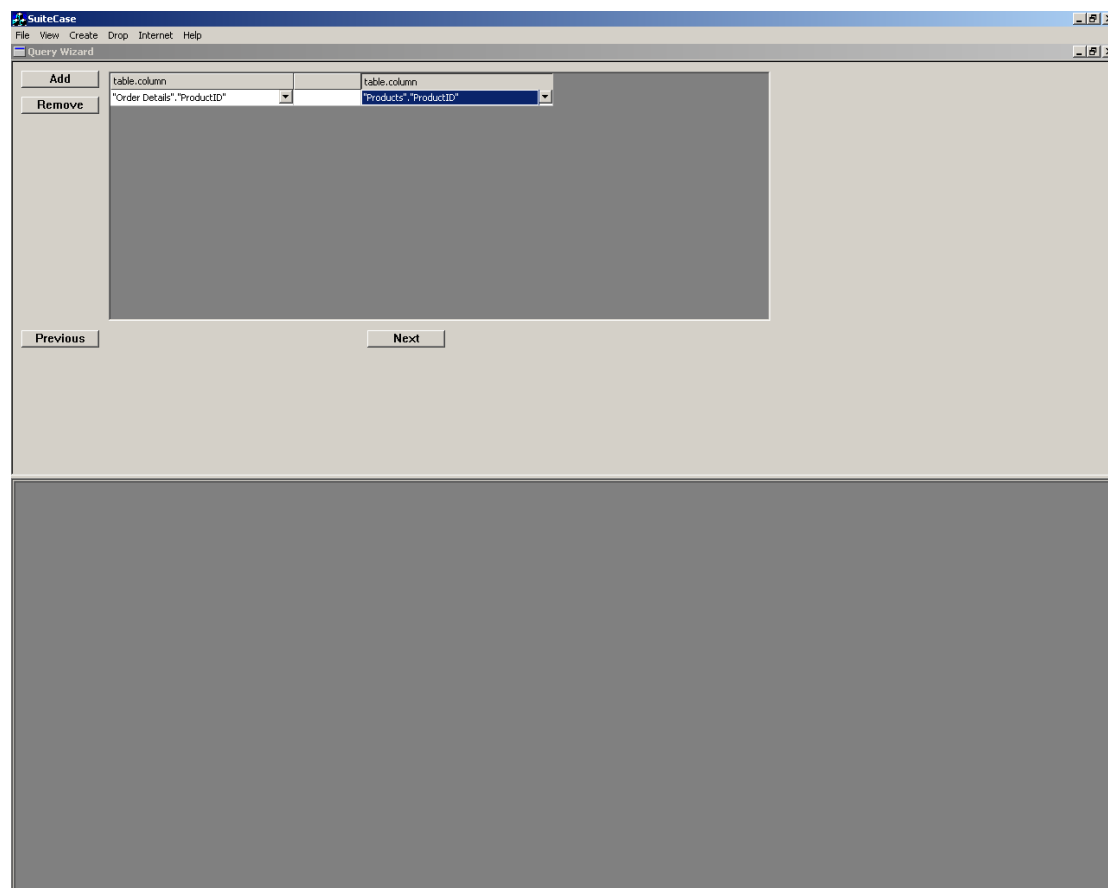


Στο δεύτερο βήμα του οδηγού, το στοιχείο ελέγχου κελιών αποτελείται από 3 στήλες: μια στήλη εισαγωγής κειμένου, και δύο στήλες αναδυόμενης λίστας. Στην πρώτη στήλη ο χρήστης μπορεί να εισάγει ένα όνομα εμφάνισης για την στήλη των επιστρεφόμενων δεδομένων του ερωτήματος, προαιρετικά. Στην δεύτερη και στην τρίτη στήλη ο χρήστης μπορεί να επιλέξει μια συνάρτηση και την στήλη δεδομένων επάνω στην οποία θα εφαρμοστεί, το αποτέλεσμα της οποίας θα επιστραφεί με την εκτέλεση του ερωτήματος. Σε αυτό το βήμα ο χρήστης επιλέγει τις στήλες και τη σειρά τους που επιθυμεί να εμφανίσει το ερώτημα, με πιθανή εφαρμογή συνάρτησης και αλλαγή ονόματος.

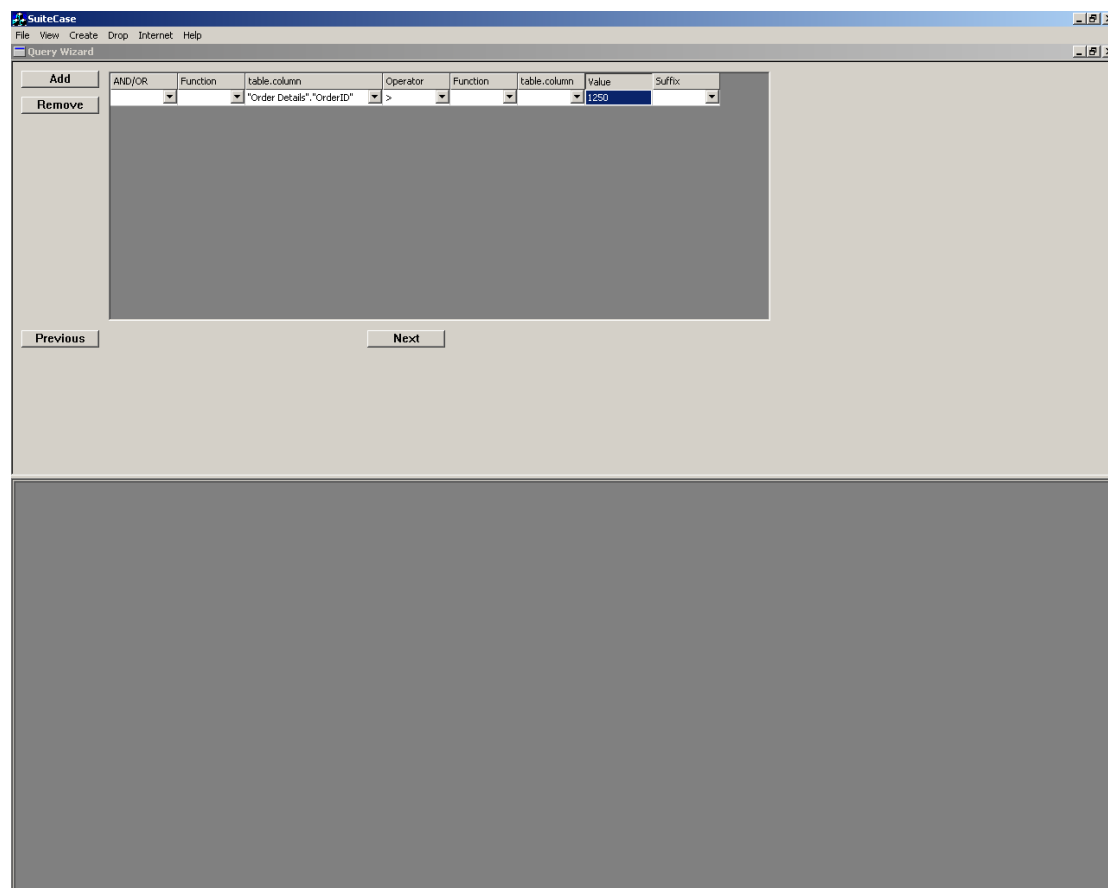




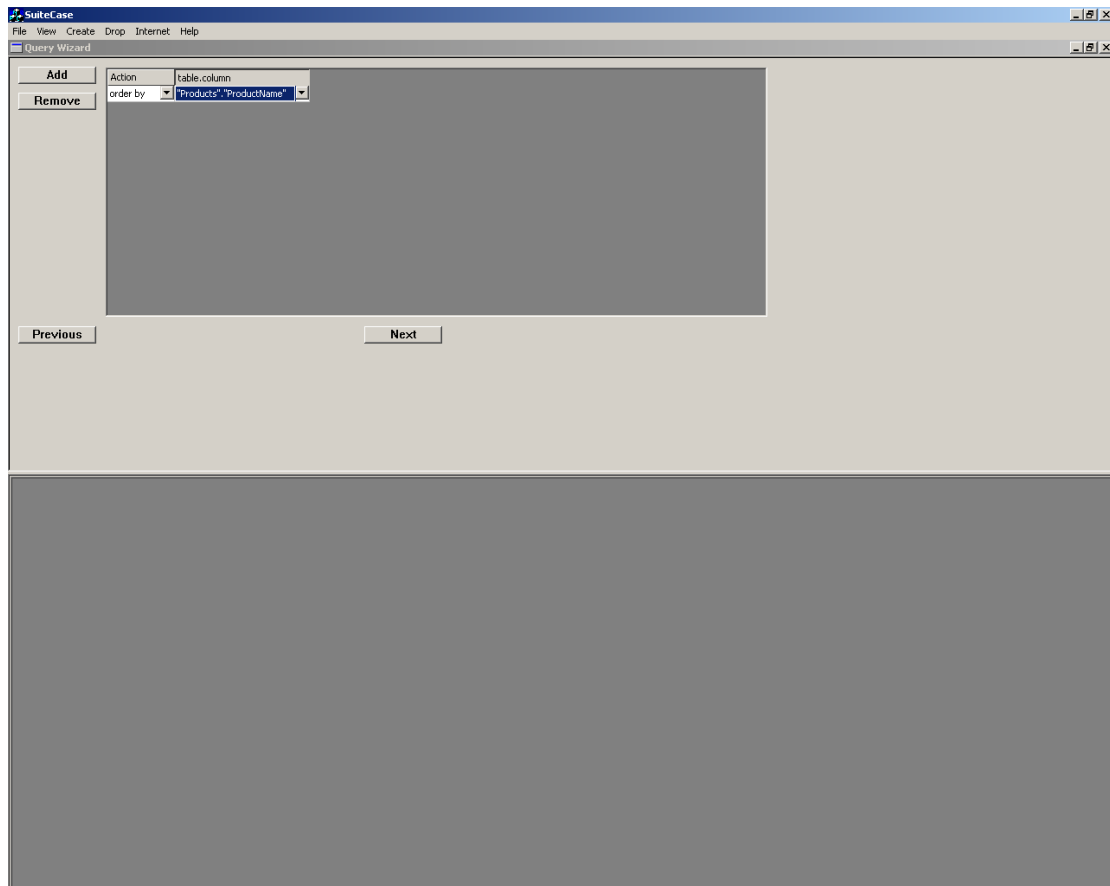
Στο τρίτο βήμα του οδηγού, το στοιχείο ελέγχου κελιών αποτελείται από τρεις στήλες. Η μεσαία στήλη έχει διαχωριστικό ρόλο μεταξύ των άλλων δύο στηλών, τα κελιά των οποίων αποτελούνται από αναδυόμενες λίστες. Στις δύο αναδυόμενες λίστες κάθε γραμμής ο χρήστης μπορεί να ορίσει μια αντιστοιχία στηλών που προσδιορίζει σύνδεση ξένου κλειδιού μεταξύ δύο πινάκων.



Στο τέταρτο βήμα του οδηγού το στοιχείο ελέγχου κελιών αποτελείται από 8 στήλες. Τα στοιχεία που επιλέγονται σε κάθε γραμμή προσδιορίζουν ένα κριτήριο το οποίο πρέπει να τηρηθεί κατά την ανάκτηση των δεδομένων. Ο χρήστης μπορεί να επιλέξει το όνομα μιας στήλης και μια άλλη στήλη ή μια τιμή και τον συντελεστή που πρέπει να ισχύει μεταξύ τους. Επίσης έχει την δυνατότητα να επιλέξει μια συνάρτηση που θα εφαρμόζεται στα δύο μέλη της σχέσης και τις σχέσεις ισχύος μεταξύ των κριτηρίων (AND / OR).



Στο πέμπτο βήμα του οδηγού, το στοιχείο ελέγχου κελιών αποτελείται από 2 στήλες. Η πρώτη στήλη παρέχει τη δυνατότητα επιλογής από αναδυόμενη λίστα, εντολής ταξινόμησης ή ομαδοποίησης. Η δεύτερη στήλη παρέχει τη δυνατότητα επιλογής από αναδυόμενη λίστα της στήλης επάνω στην οποία θα εφαρμοστεί η ενέργεια της ταξινόμησης ή της ομαδοποίησης. Οι επιλογές σε αυτό το στοιχείο ελέγχου κελιών σχηματίζουν τις παραμέτρους του ερωτήματος που αντιστοιχούν στις εντολές “order by” και “group by”.



Στο τελευταίο βήμα του οδηγού εμφανίζεται ένα στοιχείο ελέγχου εισαγωγής κειμένου αρχικοποιημένο με την εντολή SQL ανάκτησης δεδομένων που προκύπτει από τις επιλογές του χρήστη στα προηγούμενα βήματα. Η συνάρτηση που χρησιμοποιείται για να σχηματίσει το κείμενο της εντολής είναι η **ParseGridsAndShowCommand**, η οποία αναλαμβάνει να αναλύσει τα δεδομένα που εισήχθησαν στα στοιχεία ελέγχου κελιών. Τέλος, το τελευταίο βήμα του οδηγού περιλαμβάνει ένα πλήκτρο εντολής για την εκτέλεση του ερωτήματος στη βάση δεδομένων και ένα πλήκτρο εντολής για δημιουργία αναφοράς με χρήση αυτού του ερωτήματος.

The screenshot shows the SuiteCase Query Wizard interface. The main window contains a text area with the following SQL query:

```
select "Order Details"."OrderID","Order Details"."Quantity","Products"."ProductName" from "Order Details"."Products" where "Order Details"."ProductID"="Products"."ProductID" AND "Order Details"."OrderID">1250 order by "Products"."ProductName"
```

Below the query text area are four buttons: "Previous", "Next", "Execute", and "Create Report".

At the bottom of the window, a table displays the results of the query. The table has three columns: OrderID, Quantity, and ProductName. The data is as follows:

OrderID	Quantity	ProductName
10265	30	Alice Mutton
10279	15	Alice Mutton
10294	15	Alice Mutton
10302	40	Alice Mutton
10319	8	Alice Mutton
10338	20	Alice Mutton
10339	70	Alice Mutton
10346	36	Alice Mutton
10415	2	Alice Mutton
10430	45	Alice Mutton
10431	50	Alice Mutton
10444	10	Alice Mutton
10523	25	Alice Mutton
10530	40	Alice Mutton
10550	8	Alice Mutton
10564	16	Alice Mutton
10573	18	Alice Mutton
10607	100	Alice Mutton
10686	30	Alice Mutton
10696	20	Alice Mutton
10698	8	Alice Mutton
10714	27	Alice Mutton
10727	20	Alice Mutton
10773	33	Alice Mutton

## 18. DDL και DML

### DDL

Το εργαλείο SuiteCase παρέχει τη δυνατότητα στους χρήστες να δημιουργήσουν αντικείμενα βάσης δεδομένων αλλά και βάσεις δεδομένων. Αυτές οι δυνατότητες δίνονται μέσα από ειδικά σχεδιασμένα παράθυρα διαλόγου που αναλαμβάνουν να λάβουν στοιχεία από τους χρήστες και να σχηματίσουν μια εντολή SQL την οποία τελικά και θα εκτελέσουν.

#### ΔΗΜΙΟΥΡΓΙΑ ΑΝΤΙΚΕΙΜΕΝΩΝ

Το παράθυρο διαλόγου δημιουργίας πίνακα, ζητάει από τον χρήστη να εισάγει το όνομα του νέου πίνακα που πρόκειται να δημιουργηθεί, και σε ένα στοιχείο ελέγχου κελιών να εισάγει τις στήλες και τις ιδιότητές τους που θα αποτελούν τον πίνακα.

Το παράθυρο διαλόγου δημιουργίας όψης, ζητάει από τον χρήστη να εισάγει το όνομα της νέας όψης που πρόκειται να δημιουργηθεί, και σε ένα στοιχείο ελέγχου εισαγωγής κειμένου να εισάγει την εντολή ανάκτησης δεδομένων με βάση την οποία θα δημιουργηθεί η όψη.

Το παράθυρο διαλόγου δημιουργίας δείκτη, ζητάει από τον χρήστη να επιλέξει το όνομα του πίνακα και της στήλης επάνω στην οποία θα δημιουργηθεί ο δείκτης και τελικά το όνομα του νέου δείκτη.

Το παράθυρο διαλόγου προσθήκης στήλης σε πίνακα, ζητάει από τον χρήστη να επιλέξει τον πίνακα στον οποίο θα προστεθεί η στήλη, το όνομα της στήλης, τον τύπο δεδομένων, αν είναι πρωτεύον κλειδί, αν επιτρέπεται η τιμή NULL, αν θα υπάρχει τιμή εκσφαλμάτωσης (default value) και την τιμή έναρξης και βήματος αν πρόκειται για μια στήλη ακολουθίας.

Το παράθυρο διαλόγου δημιουργίας βάσης δεδομένων προτρέπει τον χρήστη να εισάγει το όνομα της νέας βάσης δεδομένων.

#### 18.1.2 ΔΙΑΓΡΑΦΗ ΑΝΤΙΚΕΙΜΕΝΩΝ

Τα παράθυρα διαλόγου διαγραφής αντικειμένων βάσης δεδομένων προτρέπουν τον χρήστη να επιλέξει από μια σειρά στοιχείων ελέγχου αναδυόμενης λίστας το όνομα του αντικειμένου που επιθυμεί να διαγράψει και στη συνέχεια να εκτελέσει την εντολή που σχηματίζεται.

Όλα τα στοιχεία ελέγχου αναδυόμενης λίστας αρχικοποιούνται με την ανάκτηση δεδομένων δομής χρησιμοποιώντας κάθε φορά τον αντίστοιχο κωδικό αντικειμένων (DBSCHEMA\_CATALOGS, DBSCHEMA\_TABLES, DBSCHEMA\_COLUMNS, DBSCHEMA\_INDEXES).

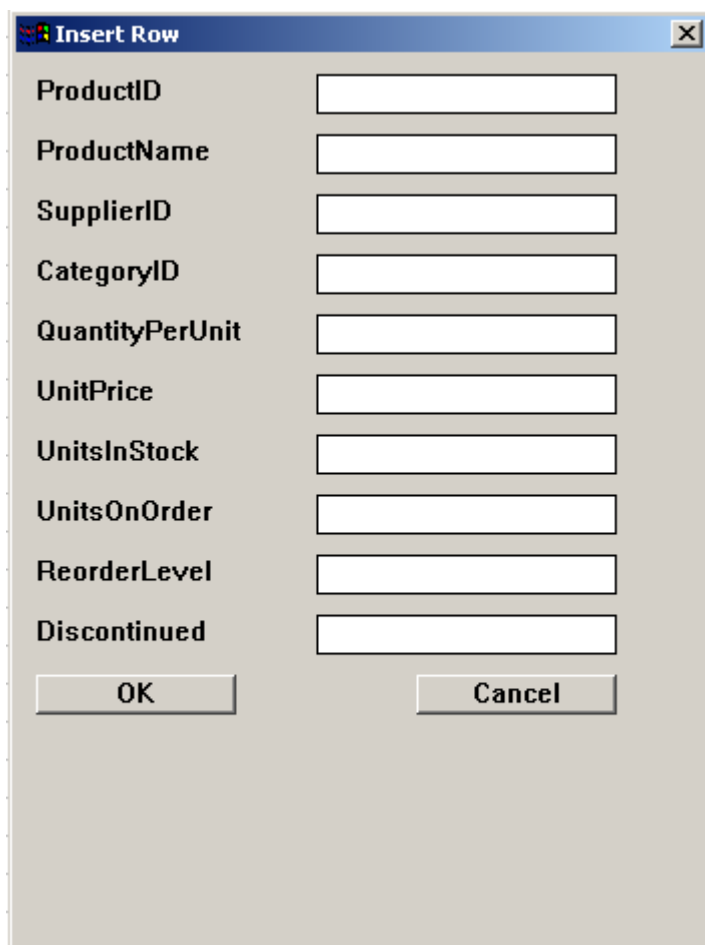
Σε όλα τα παράθυρα διαλόγου χρησιμοποιείται ένα στοιχείο ελέγχου εισαγωγής κειμένου όπου ο χρήστης μπορεί να ελέγξει την εντολή που σχηματίζεται σύμφωνα με τις επιλογές του. Η λειτουργία του κώδικα επικεντρώνεται στο να δημιουργήσει την εντολή αυτή που εισάγεται στο στοιχείο ελέγχου εισαγωγής κειμένου και τελικά να εκτελέσει αυτή την εντολή στην βάση δεδομένων.

## DML

Η διαχείριση δεδομένων στην εφαρμογή SuiteCase πραγματοποιείται μέσω ενός παραθύρου διαλόγου. Οι διαδικασίες της εισαγωγής, ενημέρωσης και αντιγραφής γραμμών δεδομένων πραγματοποιούνται μέσα από το ίδιο παράθυρο διαλόγου.

### 18.2.1 ΕΙΣΑΓΩΓΗ ΔΕΔΟΜΕΝΩΝ

Το παράθυρο διαλόγου αυτό, εμφανίζεται αρχικοποιημένο με τα ονόματα των στηλών ενός πίνακα, μέσα από το εργαλείο DataView. Όταν πρόκειται για εισαγωγή μιας νέας γραμμής, τότε τα στοιχεία ελέγχου εισαγωγής κειμένου εμφανίζονται κενά για την εισαγωγή δεδομένων από τον χρήστη.



The image shows a dialog box titled "Insert Row" with a close button in the top right corner. The dialog box contains ten input fields, each with a label to its left: ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, UnitsInStock, UnitsOnOrder, ReorderLevel, and Discontinued. At the bottom of the dialog box, there are two buttons: "OK" and "Cancel".

### 18.2.2 ΕΝΗΜΕΡΩΣΗ ΔΕΔΟΜΕΝΩΝ

Στην περίπτωση της ενημέρωσης μιας γραμμής, τα στοιχεία ελέγχου εισαγωγής κειμένου εμφανίζονται αρχικοποιημένα με τα δεδομένα της επιλεγμένης γραμμής από το εργαλείο DataView. Το πρωτεύον κλειδί εντοπίζεται και το στοιχείο ελέγχου που του αντιστοιχεί απενεργοποιείται, ώστε να μην αλλάξει το περιεχόμενό του, και στη συνέχεια να χρησιμοποιηθεί στην εντολή ενημέρωσης.

The image shows a dialog box titled "Update Row" with a close button (X) in the top right corner. The dialog box contains a list of fields and their corresponding values, each in a text input field:

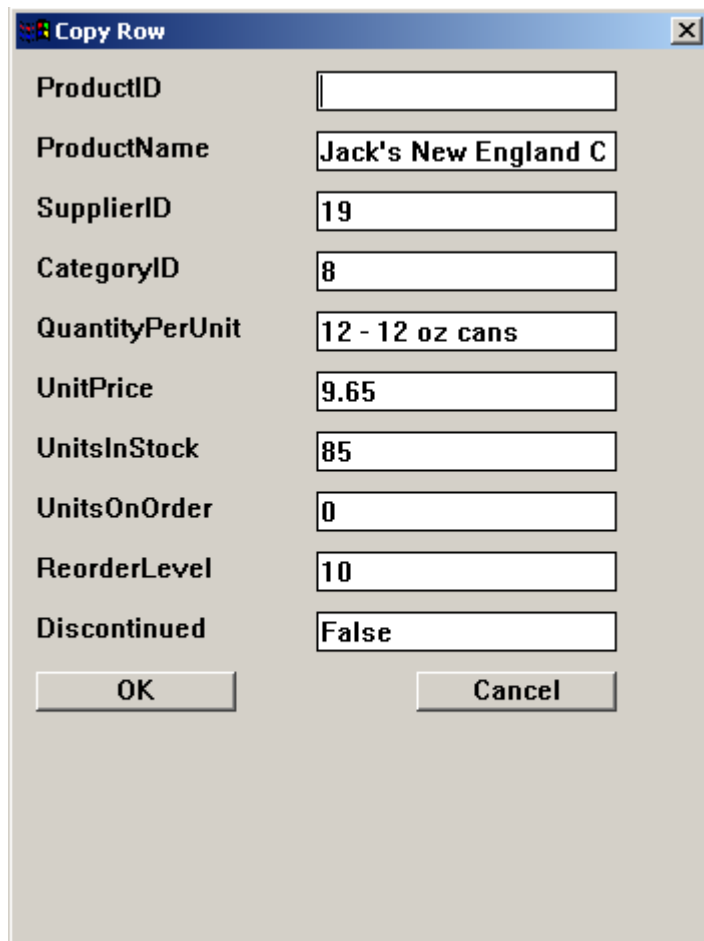
Field Name	Value
ProductID	5
ProductName	Chef Anton's Gumbo
SupplierID	2
CategoryID	2
QuantityPerUnit	36 boxes
UnitPrice	21.35
UnitsInStock	0
UnitsOnOrder	0
ReorderLevel	0
Discontinued	True

At the bottom of the dialog box, there are two buttons: "OK" on the left and "Cancel" on the right.



### 18.2.3 ΑΝΤΙΓΡΑΦΗ ΔΕΔΟΜΕΝΩΝ

Στην περίπτωση της αντιγραφής μιας γραμμής, τα στοιχεία ελέγχου εισαγωγής κειμένου εμφανίζονται αρχικοποιημένα με τα δεδομένα της επιλεγμένης γραμμής από το εργαλείο DataView, επίσης, εκτός από το στοιχείο ελέγχου εισαγωγής κειμένου το οποίο αντιστοιχεί στη στήλη του πρωτεύοντος κλειδιού. Ο χρήστης μπορεί να εισάγει ένα νέο πρωτεύον κλειδί, ή να αφήσει να εισαχθεί ένα αυτόματα καθώς το πρωτεύον κλειδί δεν μπορεί να αντιγραφεί στον ίδιο πίνακα.



Field	Value
ProductID	
ProductName	Jack's New England C
SupplierID	19
CategoryID	8
QuantityPerUnit	12 - 12 oz cans
UnitPrice	9.65
UnitsInStock	85
UnitsOnOrder	0
ReorderLevel	10
Discontinued	False

## 19. ΔΗΜΙΟΥΡΓΙΑ ΑΝΑΦΟΡΩΝ ΚΑΙ ΦΟΡΜΩΝ ΕΙΣΑΓΩΓΗΣ ΔΕΔΟΜΕΝΩΝ

### 19.1 ΑΝΑΦΟΡΕΣ ΔΕΔΟΜΕΝΩΝ

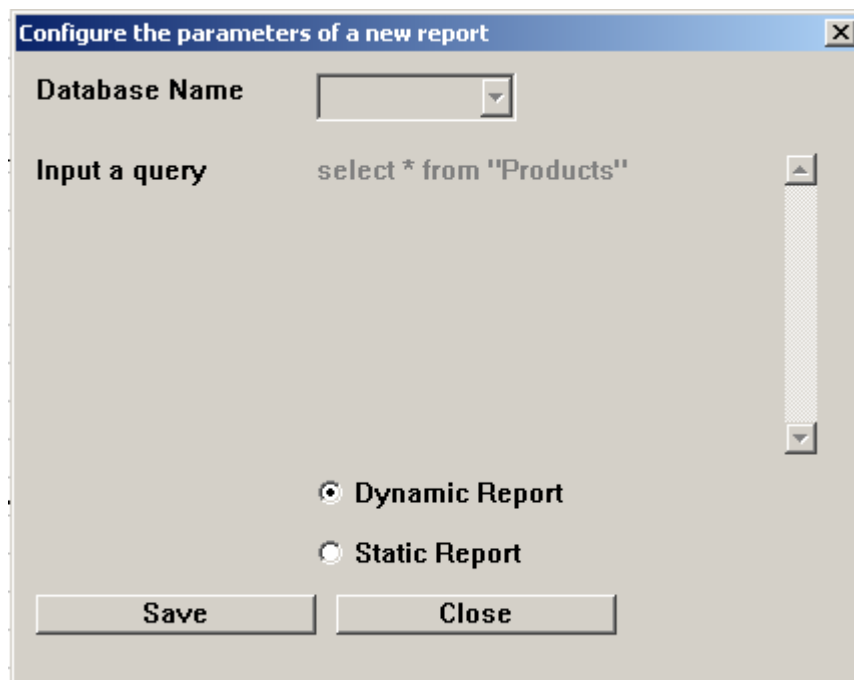
Η δυνατότητα δημιουργίας αναφορών (Reports) από την εφαρμογή SuiteCase παρέχεται μέσω ενός παραθύρου διαλόγου το οποίο προτρέπει τον χρήστη να επιλέξει μια βάση δεδομένων και να εισάγει το ερώτημα ανάκτησης δεδομένων που θα χρησιμοποιείται.

Οι αναφορές δεδομένων μπορεί να είναι δύο ειδών: στατικές αναφορές τρέχουσας κατάστασης και δυναμικές αναφορές δεδομένων που συμφωνούν με κάποια κριτήρια.

Για την δημιουργία των ιστοσελίδων html ή asp χρησιμοποιείται ως base class η class CHTMLFile η οποία παρέχει την δυνατότητα εισαγωγής, σε ένα αρχείο κειμένου, των πιο συχνά χρησιμοποιούμενων ετικετών της γλώσσας προσδιορισμού ιστοσελίδων HTML. Επίσης, παρέχει την δυνατότητα αποθήκευσης του αρχείου.

Οι class που χρησιμοποιούνται για την δημιουργία των στατικών και των δυναμικών αναφορών κληρονομούν από την class CHTMLFile όλες τις ετικέτες της HTML και διαθέτουν από μία συνάρτηση η οποία αναλαμβάνει να συνθέσει το αρχείο html ή asp αντίστοιχα. Η συνάρτηση αυτή λέγεται MakeTheReport και παίρνει ως παραμέτρους τον τίτλο και την επικεφαλίδα του αρχείου που αναλαμβάνει να συνθέσει.

Στα δύο είδη των αναφορών, χρησιμοποιείται η δυνατότητα σύνθεσης πίνακα της HTML για την προβολή των δεδομένων. Στις στατικές αναφορές, τα δεδομένα εισάγονται ως κείμενο στο αρχείο html που δημιουργείται, ενώ στις δυναμικές αναφορές το αρχείο asp δημιουργείται με ένα script σε VBScript το οποίο αναλαμβάνει να εκτελεί την εντολή ανάκτησης δεδομένων χρησιμοποιώντας την βιβλιοθήκη ADO.



## 19.2 ΦΟΡΜΕΣ ΕΙΣΑΓΩΓΗΣ ΔΕΔΟΜΕΝΩΝ

Η δυνατότητα δημιουργίας φορμών εισαγωγής δεδομένων παρέχεται από την εφαρμογή SuiteCase μέσω ενός παραθύρου διαλόγου στο οποίο ο χρήστης μπορεί να επιλέξει τις συνδέσεις των ξένων κλειδιών που υπάρχουν στον πίνακα.

Οι φόρμες εισαγωγής δεδομένων δημιουργούνται πάντα σε μορφή asp καθώς είναι επιφορτισμένες με την εκτέλεση μιας εντολής εισαγωγής δεδομένων σε έναν πίνακα. Η class που αναλαμβάνει την δημιουργία των φορμών εισαγωγής κληρονομεί επίσης όλες τις ετικέτες της γλώσσας HTML από την class CHtmlFile, και διαθέτει την συνάρτηση MakeTheForm η οποία αναλαμβάνει την σύνθεση του κειμένου που θα αποτελεί το τελικό αρχείο asp.

Το τελικό αρχείο asp χωρίζεται σε δύο τμήματα: το τμήμα δημιουργίας της φόρμας εισαγωγής δεδομένων και το τμήμα εκτέλεσης της εντολής καταχώρησης δεδομένων στον πίνακα, με τη μέθοδο κλήσης querystring.

Η φόρμα εισαγωγής δεδομένων μπορεί να περιλαμβάνει στοιχεία ελέγχου εισαγωγής κειμένου ή αναδυόμενης λίστας, ανάλογα με το αν ο χρήστης επέλεξε πίνακα, πεδίο εμφάνισης και πεδίο καταχώρισης στο αντίστοιχο παράθυρο διαλόγου της εφαρμογής SuiteCase.

Το τμήμα εκτέλεσης της εντολής καταχώρισης δεδομένων ανακτά τα δεδομένα που εισήχθησαν στη φόρμα και σχηματίζει την εντολή insert η οποία και εκτελείται με χρήση της βιβλιοθήκης ADO.

	Link to table	Column	Show Column
ProductID			
ProductName			
SupplierID			
CategoryID			
QuantityPerUnit			
UnitPrice			
UnitsInStock			
UnitsOnOrder			
ReorderLevel			
Discontinued			

OK Cancel

## 20. ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ SUITECASE

Ως παράδειγμα για την χρήση της εφαρμογής SuiteCase χρησιμοποιείται μια εικονική δημοσκόπηση. Για το παράδειγμα κατασκευάζεται μια βάση δεδομένων με πίνακες στοιχείων σχετικά με διάφορες πληροφορίες που αφορούν τα προσωπικά στοιχεία των ερωτούμενων και τις προτιμήσεις τους σε κάποια θέματα. Τα δεδομένα που χρησιμοποιούνται παρακάτω είναι εικονικά και ανώνυμα.

Για την δημιουργία της βάσης δεδομένων εκτελείται το αρχείο σεναρίου db\_creation.scr, που παρουσιάζεται παρακάτω, με το εργαλείο ScriptEdit.

```
*****
```

```
** db_creation.scr
```

```
create database gallop;
```

```
use gallop;
```

```
create table AgeGroup (id integer, name varchar(20));
```

```
insert into AgeGroup values(1, '12-16');
```

```
insert into AgeGroup values(2, '17-24');
```

```
insert into AgeGroup values(3, '25-34');
```

```
insert into AgeGroup values(4, '35-44');
```

```
insert into AgeGroup values(5, '45-54');
```

```
insert into AgeGroup values(6, '55-64');
```

```
insert into AgeGroup values(7, '65+');
```

```
create table Sex (id integer, name varchar(10));
```

```
insert into Sex values(1, 'Male');
```

```
insert into Sex values(2, 'Female');
```

```
create table MaritalStatus (id integer, name varchar(10));
```

```
insert into MaritalStatus values(1, 'Single');
```

```
insert into MaritalStatus values(2, 'Married');
```

```
insert into MaritalStatus values(3, 'Divorced');
```

```
insert into MaritalStatus values(4, 'Widow -er');
```

```
create table JobKind (id integer, name varchar(20));
```

```
insert into JobKind values(1, 'Employee');
```

```
insert into JobKind values(2, 'Worker');
```

```
insert into JobKind values(3, 'Employer');
```

```
insert into JobKind values(4, 'Manager');
```

```
insert into JobKind values(5, 'Free Lancer');
```

```
insert into JobKind values(6, 'Unemployed');
```

```
create table Education (id integer, name varchar(20));
```

```

insert into Education values(1, 'Illiterate');
insert into Education values(2, 'Primary school');
insert into Education values(3, 'Junior high School');
insert into Education values(4, 'High School');
insert into Education values(5, 'College');
insert into Education values(6, 'University');

create table MusicStyle (id integer, name varchar(20));

insert into MusicStyle values(1, 'Acid');
insert into MusicStyle values(2, 'Ballads');
insert into MusicStyle values(3, 'Blues');
insert into MusicStyle values(4, 'Classic');
insert into MusicStyle values(5, 'Jazz');
insert into MusicStyle values(6, 'Heavy metal');
insert into MusicStyle values(7, 'Latin');
insert into MusicStyle values(8, 'Rock');
insert into MusicStyle values(9, 'Soul');

create table Color (id integer, name varchar(20));

insert into Color values(1, 'White');
insert into Color values(2, 'Yellow');
insert into Color values(3, 'Orange');
insert into Color values(4, 'Red');
insert into Color values(5, 'Green');
insert into Color values(6, 'Magenta');
insert into Color values(7, 'Blue');
insert into Color values(8, 'Black');

create table Inform (id integer, name varchar(20));

insert into Inform values(1, 'News papers');
insert into Inform values(2, 'Magazines');
insert into Inform values(3, 'Radio');
insert into Inform values(4, 'Television');
insert into Inform values(5, 'Internet');

create table main (AgeGroupID integer, SexID integer, MaritalStatusID
integer, JobKindID integer, EducationID integer, MusicStyleID
integer, ColorID integer, InformID integer);

```

\*\*\*\*\*

Σε αυτό το αρχείο περιλαμβάνονται οι εντολές δημιουργίας της βάσης δεδομένων που χρησιμοποιούνται από την δημοσκόπηση. Δημιουργούνται 8 πίνακες που αποθηκεύουν κάποιες διακριτές τιμές από τις οποίες θα κληθεί να επιλέξει κάθε ερωτούμενος. Οι επιλογές κάθε ερωτούμενου αποθηκεύονται τελικά στον πίνακα main. Ο πίνακας main περιλαμβάνει 8 ξένα κλειδιά σε κάθε έναν από αυτούς τους πίνακες. Σκοπός είναι να δοθεί η δυνατότητα στους χρήστες του συστήματος να επιλέξουν μια από τις τιμές κάθε ενός πίνακα και τελικά να αποθηκευτούν στον πίνακα main τα πρωτεύοντα κλειδιά του κάθε πίνακα σύμφωνα με αυτές τις επιλογές.

Εφ' όσον εκτελεστεί το σενάριο εντολών, η βάση δεδομένων είναι έτοιμη για την εισαγωγή δεδομένων. Η φόρμα εισαγωγής με την οποία θα αλληλεπιδρούν οι χρήστες θα δημιουργηθεί με το αντίστοιχο εργαλείο της εφαρμογής SuiteCase. Από το μενού της εφαρμογής ο χρήστης μπορεί να επιλέξει:

Internet -> Data Insert Form

Και εμφανίζεται το παράθυρο επιλογής βάσης δεδομένων και πίνακα. Ο χρήστης μπορεί να επιλέξει τη βάση δεδομένων “gallop“ και τον πίνακα “main” που μόλις δημιουργήθηκε με την εκτέλεση του σεναρίου εντολών.

Το παράθυρο που θα εμφανιστεί θα έχει την εξής μορφή, με επιλεγμένες τις κατάλληλες παραμέτρους που προσδιορίζουν τους πίνακες σύνδεσης:

	Link to table	Column	Show Column
AgeGroupID	AgeGroup	id	name
SexID	Sex	id	name
MaritalStatusID	MaritalStatu	id	name
JobKindID	JobKind	id	name
EducationID	Education	id	name
MusicStyleID	MusicStyle	id	name
ColorID	Color	id	name
InformID	Inform	id	name

OK Cancel

Όταν επιλεγεί το πλήκτρο OK τότε θα ζητηθεί από τον χρήστη να επιλέξει τον φάκελο και το όνομα του αρχείου που στην συνέχεια αποθηκεύεται.

Το αρχείο ASP είναι έτοιμο να χρησιμοποιηθεί από κάποιον Web Server. Όταν οι χρήστες θα επισκέπτονται την ιστοσελίδα που φιλοξενεί αυτή την φόρμα θα συμπληρώνουν τη φόρμα με την εξής μορφή, μέσα από τον web browser που χρησιμοποιούν:

The screenshot shows a Microsoft Internet Explorer window with the title 'Suitecase project generated form - Microsoft Internet Explorer'. The address bar shows 'http://localhost/stavros/main\_gallop.asp'. The main content area displays a form titled 'TestForm' with the following fields:

- AgeGroupID: 12-16
- SexID: Male
- MaritalStatusID: Single
- JobKindID: Employee
- EducationID: Illiterate
- MusicStyleID: Acid
- ColorID: White
- InformID: News papers

A 'Submit' button is positioned below the InformID field.

Το σενάριο εντολών εισαγωγής εικονικών δεδομένων που χρησιμοποιήθηκε για την εφαρμογή είναι το data.scr:

\*\*\*\*\*

\*\* data.scr

use gallop;

```
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(1, 1, 1, 1, 4,
6, 2, 1);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(2, 2, 1, 1, 1,
8, 1, 2);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 1, 1, 1, 3,
1, 3, 3);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(4, 1, 2, 2, 1,
2, 1, 1);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(1, 2, 1, 1, 4,
5, 7, 3);
```

```
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(2, 2, 1, 1, 2,
2, 7, 3);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(2, 2, 1, 2, 5,
1, 8, 3);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(2, 2, 1, 1, 1,
2, 4, 4);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(2, 1, 1, 3, 2,
8, 2, 2);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(2, 1, 1, 1, 1,
2, 1, 4);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(2, 1, 2, 1, 5,
1, 3, 4);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 1, 1, 1, 6,
1, 1, 4);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 2, 3, 1, 6,
8, 8, 3);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(1, 2, 1, 1, 1,
2, 1, 5);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(5, 1, 1, 4, 6,
1, 5, 1);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(1, 2, 4, 1, 5,
1, 1, 3);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(4, 1, 1, 1, 6,
3, 6, 1);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(1, 1, 1, 4, 1,
2, 7, 5);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 1, 1, 1, 2,
3, 1, 2);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(4, 1, 1, 1, 3,
9, 1, 1);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(1, 1, 1, 1, 4,
1, 1, 5);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 2, 1, 4, 1,
3, 3, 2);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(4, 2, 2, 1, 3,
2, 7, 1);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 1, 2, 1, 1,
1, 4, 5);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 2, 1, 1, 6,
8, 6, 3);
```



```

insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(1, 1, 1, 1, 3,
4, 4, 5);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(4, 1, 1, 3, 3,
2, 1, 2);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 1, 4, 1, 4,
6, 5, 1);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(3, 2, 1, 1, 1,
1, 3, 2);
insert into main(AgeGroupID, SexID, MaritalStatusID, JobKindID,
EducationID, MusicStyleID, ColorID, InformID) values(4, 2, 1, 1, 1,
9, 8, 1);

```

\*\*\*\*\*

Για να δημιουργηθούν αναφορές δεδομένων που υπάρχουν στην βάση χρησιμοποιούμε το εργαλείο δημιουργίας αναφορών της εφαρμογής SuiteCase. Στο μενού της εφαρμογής ο χρήστης μπορεί να επιλέξει:

Internet -> Data Display Report

Στη συνέχεια συμπληρώνει τη φόρμα που εμφανίζεται σύμφωνα με τις απαιτήσεις:

Configure the parameters of a new report

Database Name: gallop

Input a query:
 

```

select Sex.name, AgeGroup.name,
Color.name, MusicStyle.name from
main, Sex, AgeGroup, Color,
MusicStyle where main.SexID =
Sex.id and main.AgeGroupID =
AgeGroup.id and main.ColorID =
Color.id and main.MusicStyleID =
MusicStyle.id and Sex.name =
'Male'

```

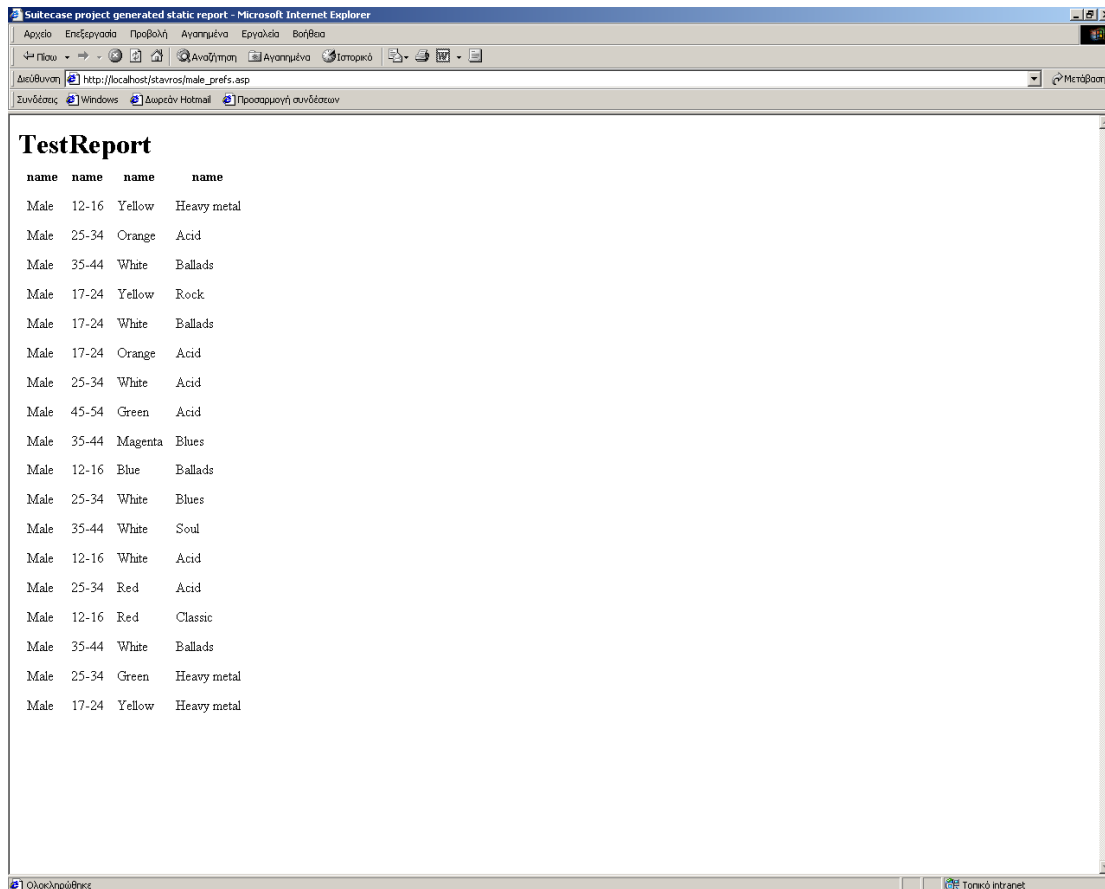
Dynamic Report  
 Static Report

Save Close

Στην προκειμένη υποθετική περίπτωση επιλέγεται να εμφανιστεί μια δυναμική αναφορά που θα παρουσιάζει τις ηλικίες, τα προτιμώμενα χρώματα και τα προτιμώμενα είδη μουσικής των ανδρών που έλαβαν μέρος στην δημοσκόπηση. Όταν επιλεγεί το πλήκτρο Save τότε ο χρήστης έχει τη δυνατότητα να επιλέξει το φάκελο

και το όνομα του αρχείου όπου θα αποθηκευτεί η ιστοσελίδα. Η ιστοσελίδα είναι πλέον έτοιμη να εγκατασταθεί σε έναν Web Server.

Η εμφάνιση της ιστοσελίδας στους επισκέπτες θα είναι κάπως έτσι:



Ο web server που χρησιμοποιήθηκε για τις δοκιμές είναι ο Internet Information Services (IIS) της Microsoft.

## 21. Η ΑΝΑΠΤΥΞΗ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

### 21.1 ΣΧΕΔΙΑΣΜΟΣ

Ο σχεδιασμός της εφαρμογής έγινε με τρόπο που διευκολύνει την μελλοντική της επέκταση. Η ανάπτυξη των βιβλιοθηκών δυναμικών συνδέσεων προωθούν την συντήρηση της εφαρμογής, την προσθήκη νέων δυνατοτήτων και εργαλείων και την επαναχρησιμοποίησή της με άλλους Database servers.

Η λειτουργικότητα των εργαλείων μπορεί να αλλάξει, εφ' όσον ζητηθεί, χωρίς να χρειάζεται η αλλαγή του κυρίως σώματος της εφαρμογής. Ο εντοπισμός κάποιου λογικού σφάλματος ή του τμήματος μιας λειτουργίας που πρέπει να βελτιωθεί γίνεται ευκολότερα ακόμα και σε περιπτώσεις εισαγωγής μιας νέας λειτουργίας ή ενός νέου εργαλείου ως μια νέα βιβλιοθήκη δυναμικών συνδέσεων.

Επίσης, η τμηματοποίηση της εφαρμογής σε βιβλιοθήκες δυναμικών συνδέσεων διευκολύνει:

- την παραγωγή νέων εκδόσεων της εφαρμογής με προσαρμοσμένες αλλαγές στη λειτουργικότητα των εργαλείων
- και την αναδιανομή τους στους χρήστες.

### 21.2 Η ΔΟΜΗ ΤΩΝ ΦΑΚΕΛΩΝ

Κατά τη διάρκεια της ανάπτυξης της εφαρμογής χρησιμοποιήθηκε μια συγκεκριμένη δομή φακέλων έτσι ώστε να αποφευχθεί η ύπαρξη πολλαπλών αντιγράφων κάποιων αρχείων.

- Τα αρχεία επικεφαλίδων εισαγωγής των προτύπων των συναρτήσεων των βιβλιοθηκών δυναμικών συνδέσεων, που χρησιμοποιούνται συχνά από πολλά διαφορετικά συστατικά στοιχεία της εφαρμογής, εισάγονται σε κάθε βιβλιοθήκη δυναμικών συνδέσεων που τα χρησιμοποιούν από το σημείο που βρίσκονται στον φάκελο `__MyIncludes`.
- Το ίδιο συμβαίνει και με τα αρχεία υλοποίησης του στοιχείου ελέγχου κελιών.
- Τα αρχεία `lib` (μαζί με τα αρχεία `dll`) βρίσκονται επίσης τοποθετημένα σε κάποιο συγκεκριμένο φάκελο, από τον οποίο και γίνεται η εισαγωγή τους στο κάθε περιβάλλον εργασίας κάθε στοιχείου που τα χρησιμοποιεί. Ο φάκελος αυτός είναι ο `__MyDLLs`.

Κάθε περιβάλλον εργασίας εισάγει αυτά τα αρχεία επικεφαλίδων και τα αρχεία `lib` από τους συγκεκριμένους καταλόγους, όπου τοποθετούνται πάντα οι πιο πρόσφατες εκδόσεις τους.

### 21.3 ΑΝΑΠΤΥΞΗ ΒΙΒΛΙΟΘΗΚΩΝ ΔΥΝΑΜΙΚΩΝ ΣΥΝΔΕΣΕΩΝ

Η ανάπτυξη καθενός πρωτεύοντος ή δευτερεύοντος εργαλείου πραγματοποιήθηκε μέσα από εφαρμογές δοκιμαστικής λειτουργίας (test applications) έτσι ώστε να ελεγχθεί η λειτουργία τους. Κατόπιν, ο κώδικας κάθε εργαλείου

απομονώθηκε και εισήχθη σε περιβάλλον ανάπτυξης βιβλιοθήκης δυναμικών συνδέσεων (DLL – Dynamic Link Library). Για κάθε βιβλιοθήκη δυναμικών συνδέσεων δημιουργήθηκαν από τον compiler της Visual C++ 6 αρχεία βιβλιοθήκης (lib) και αρχεία δυαδικού κώδικα (dll). Επίσης, για την εξαγωγή των συναρτήσεων κάθε βιβλιοθήκης δημιουργήθηκε ένα αρχείο επικεφαλίδων που προσδιορίζει τα πρότυπα των συναρτήσεων αυτών (π.χ. MyLibraryAPI.h).

Για τη χρήση μιας βιβλιοθήκης δυναμικών συνδέσεων από μια εφαρμογή απαιτείται η ενσωμάτωση στην εφαρμογή του αντίστοιχου αρχείου lib, και ενός αρχείου επικεφαλίδων το οποίο προσδιορίζει τα πρότυπα των συναρτήσεων που εξάγονται από την βιβλιοθήκη (π.χ. impMyLibrary.h). Επίσης, το αρχείο δυαδικού κώδικα dll θα πρέπει να βρίσκεται σε κάποιο κατάλογο του συστήματος όπου μπορεί να εντοπιστεί από την εφαρμογή που το χρησιμοποιεί.

Για την εξαγωγή των προτύπων των συναρτήσεων και έπειτα για την εισαγωγή τους χρησιμοποιήθηκε η δήλωση `__declspec(dllexport)` και `__declspec(dllimport)` αντίστοιχα.

Παράδειγμα:

```
void __declspec(dllexport) MyFunc(int a, int b);
```

Η δήλωση αυτή περιλαμβάνεται στον κώδικα της δυναμικής βιβλιοθήκης και πραγματοποιεί την εξαγωγή της συνάρτησης MyFunc η οποία λαμβάνει ως ορίσματα δύο ακέραιους αριθμούς, πραγματοποιεί κάποια επεξεργασία και δεν επιστρέφει κάποια τιμή.

```
void __declspec(dllimport) MyFunc(int a, int b);
```

Η δήλωση αυτή περιλαμβάνεται στον κώδικα της εφαρμογής που χρησιμοποιεί την συγκεκριμένη συνάρτηση και ενημερώνει το περιβάλλον εργασίας ότι η υλοποίηση της συνάρτησης περιλαμβάνεται σε μία βιβλιοθήκη δυναμικών συνδέσεων.

## 21.4 Η ΕΠΙΛΟΓΗ ΤΟΥ ΣΤΟΙΧΕΙΟΥ ΕΛΕΓΧΟΥ ΚΕΛΙΩΝ

Αρχικά, το στοιχείο ελέγχου κελιών που επιλέχτηκε να χρησιμοποιηθεί ήταν το ActiveX Hierarchical Flex Grid Control της Microsoft σε μορφή OCX. Το συγκεκριμένο στοιχείο ελέγχου, λόγω της μορφής του ως OCX, κυρίως, κρίθηκε ακατάλληλο για δύο λόγους:

- Μικρή ταχύτητα απόκρισης και λειτουργίας.
- Οι χρήστες της εφαρμογής θα πρέπει να εγκαταστήσουν το συγκεκριμένο στοιχείο ελέγχου στο σύστημά τους (registry).

Το grid control του Chris Maunder από το [www.codeproject.com](http://www.codeproject.com) κρίθηκε καταλληλότερο καθώς έχει τα πλεονεκτήματα της ταχύτητας και της ενσωμάτωσης στην εφαρμογή. Η μορφή στην οποία διανέμεται είναι ο πηγαίος της κώδικας σε μορφή κλάσεων (classes) και υποστηρίζει όλες τις δυνατότητες που απαιτεί η εφαρμογή.

## 21.5 Η ΕΠΙΛΟΓΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ OLE DB

Η βιβλιοθήκη προσπέλασης βάσεων δεδομένων που επιλέχτηκε είναι η OLE DB της Microsoft. Χρησιμοποιώντας την τεχνολογία OLE, η βιβλιοθήκη αυτή παρέχει interfaces στους προγραμματιστές εφαρμογών βάσεων δεδομένων. Η υλοποίηση της βιβλιοθήκης είναι χαμηλού επιπέδου και έχει σχεδιαστεί ειδικά ώστε να δίνει τη δυνατότητα προσπέλασης σε δεδομένα πολλών διαφορετικών πηγών (Data Sources).

Για να χρησιμοποιηθεί η βιβλιοθήκη OLE DB χρειάζεται να είναι εγκατεστημένο στο υπολογιστικό σύστημα η βιβλιοθήκη MSDAC (Microsoft Data Access Components) η οποία περιέχει δυναμικές βιβλιοθήκες οδηγών σύνδεσης και αλληλεπίδρασης με τις πηγές δεδομένων. Εκτός από την βιβλιοθήκη OLE DB, η MSDAC περιέχει και τις υλοποιήσεις των οδηγών ODBC (Open Database Connectivity) και ADO (ActiveX Data Objects). Η τεχνολογία ADO είναι βασισμένη στην τεχνολογία OLE DB, με έμφαση στην ευκολία χρήσης.

Η βιβλιοθήκη MFC (Microsoft Foundation Classes) περιέχει κάποιες κλάσεις περιτύλιξης (wrapper classes), των interfaces της βιβλιοθήκης OLE DB, για την παροχή υπηρεσιών σύνδεσης και αλληλεπίδρασης με βάσεις δεδομένων. Οι σημαντικότερες από αυτές τις classes είναι: η CDataSource (IDataSource), CSession (ISession), CRowset (IRowset), CCommand (ICommand) και CAccessor (IAccessor).

Αρχικά, στην ανάπτυξη της βιβλιοθήκης δυναμικών συνδέσεων MasterOledb.dll της εφαρμογής SuiteCase είχε επιλεγεί να χρησιμοποιηθεί η δομή των αντικειμένων της βιβλιοθήκης MFC. Η δημιουργία, όμως, πινάκων ανεκτιμημένων δεδομένων με δυναμικές διαστάσεις, απαιτούσε την λεπτομερή και χρονοβόρα μετατροπή δεδομένων χρησιμοποιώντας το interface IDataConvert.

Τελικά, η εφαρμογή SuiteCase, χρησιμοποιεί απ' ευθείας τα interfaces της βιβλιοθήκης OLE DB για την παροχή ενός αντικειμένου που ενσωματώνει, ολοκληρωμένες και απλοποιημένες για τον χρήστη, όλες τις λειτουργίες.

Στην εφαρμογή SuiteCase χρησιμοποιείται η βιβλιοθήκη OLE DB για προσπέλαση δεδομένων ενός MS SQL Server 2000, με τον provider SQLOLEDB και με εκτέλεση εντολών της γλώσσας Transact-SQL. Μια μετατροπή της εφαρμογής, έτσι ώστε να χρησιμοποιεί οποιονδήποτε OLE DB provider (επιλογή από χρήστη κατά τη διάρκεια της εκτέλεσης), και να εκτελεί εντολές της ANSI SQL, θα μπορούσε να καταστήσει δυνατή την μεταφορά της εφαρμογής για χρήση και με άλλους database servers.

## 21.6 Η ΜΕΤΑΤΡΟΠΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ

Η μετατροπή των δεδομένων σε διαφορετικούς τύπους είναι ένα από τα μεγάλα προβλήματα που καλείται να λύσει ένας προγραμματιστής, ειδικά σε έργα διαχείρισης βάσεων δεδομένων. Οι database servers αποθηκεύουν τα δεδομένα σε δυαδική μορφή και με αυτόν τον τρόπο μπορούν να υποστηρίξουν διάφορους προηγμένους τύπους δεδομένων. Οι βιβλιοθήκες πρόσβασης βάσεων δεδομένων αναλαμβάνουν να μετατρέψουν τα δυαδικά δεδομένα σε αναγνώσιμη από τον άνθρωπο μορφή και γι' αυτό το σκοπό παρέχουν ειδικά σχεδιασμένες συναρτήσεις και classes. Η βιβλιοθήκη OLE DB παρέχει εργαλεία αναγνώρισης των τύπων δεδομένων αλλά και μετατροπής τους.

Για την εφαρμογή SuiteCase αρχικά χρησιμοποιήθηκε το Interface IDataConvert το οποίο αναλαμβάνει να μετατρέψει δεδομένα από κάποιον τύπο της

βιβλιοθήκης OLE DB σε οποιονδήποτε άλλο. Η συνάρτηση που παρέχει το συγκεκριμένο interface για μετατροπή των δεδομένων είναι η:

```
HRESULT DataConvert (
    DBTYPE                wSrcType,
    DBTYPE                wDstType,
    ULONG                 cbSrcLength,
    ULONG *               pcbDstLength,
    void *                pSrc,
    void *                pDst,
    ULONG                 cbDstMaxLength,
    DBSTATUS             dbsSrcStatus,
    DBSTATUS *          pdbsStatus,
    BYTE                  bPrecision,
    BYTE                  bScale,
    DBDATAACONVERT      dwFlags);
```

Αυτή η συνάρτηση μετατροπής χρησιμοποιήθηκε με τα αντικείμενα CDataSource, CSession, CCommand και CDynamicAccessor σε κάθε πεδίο του πίνακα των ανεκτημένων δεδομένων. Η εκτέλεσή της είναι επαναλαμβανόμενη και αρκετά χρονοβόρα.

Η βιβλιοθήκη OLE DB παρέχει επίσης το interface IAccessor το οποίο συνεργάζεται με ένα αντικείμενο IRowset παρέχοντας μια πιο πλήρη λύση στο πρόβλημα της μετατροπής των τύπων δεδομένων. Οι δύο τρόποι μοιάζουν μεταξύ τους και έχουν μικρή διαφορά δυσκολίας στην εφαρμογή τους. Η χρήση του IAccessor είναι όμως λίγο πιο γρήγορη στην μετατροπή και αυτός είναι ο λόγος που χρησιμοποιήθηκε τελικά. Οι συναρτήσεις που χρησιμοποιούνται για την υλοποίηση της δημιουργίας των bindings και του αντικειμένου IAccessor παρέχονται από την Microsoft στην τεκμηρίωση της βιβλιοθήκης OLE DB.

## 21.7 ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Η εφαρμογή SuiteCase υποστηρίζει πολλές από τις βασικές λειτουργίες ενός database server. Παρ' όλα αυτά, πάντα υπάρχει περιθώριο βελτίωσης και αναβάθμισης. Μερικές από τις λειτουργίες που υποστηρίζει ο MS SQL Server 2000 και δεν εκμεταλλεύεται η εφαρμογή, προς το παρόν, είναι:

- Η δημιουργία χρηστών και ρόλων
- Η απονομή ρόλων στους χρήστες
- Η δημιουργία triggers, functions και stored procedures.

Το γραφικό περιβάλλον εμφάνισης των δεδομένων χρησιμοποιεί ένα στοιχείο ελέγχου κελιών. Για την εισαγωγή νέων δεδομένων, για την ενημέρωση και την αντιγραφή των δεδομένων, όμως, χρησιμοποιούνται βοηθητικά παράθυρα διαλόγου. Ο χρήστης της εφαρμογής SuiteCase θα έχει τη δυνατότητα στο μέλλον να εισάγει δεδομένα και να ενημερώνει τα ήδη υπάρχοντα, με επεξεργασία επάνω στο στοιχείο ελέγχου κελιών.

Η διαδικασία αυτή είναι περισσότερο θέμα ανταλλαγής μηνυμάτων μεταξύ των στοιχείων ελέγχου της εφαρμογής και σήμανσης του στοιχείου ελέγχου κελιών για τα δεδομένα που έχουν αλλάξει και πρέπει να ενημερωθούν στη βάση δεδομένων.

Η δημιουργία των ιστοσελίδων εισαγωγής δεδομένων και αναφοράς χρησιμοποιεί την γλώσσα HTML και την τεχνολογία ASP με VBScript. Στο μέλλον θα μπορούσαν να υλοποιηθούν συναρτήσεις οι οποίες κατασκευάζουν ιστοσελίδες με τις γλώσσες JavaScript και PHP, ή ακόμα με JSP και Java applets.

Επίσης, ο χρήστης θα μπορεί να επεξεργαστεί με περισσότερες επιλογές την εμφάνιση της ιστοσελίδας. Τέτοιες επιλογές θα μπορούσαν να είναι:

- Η επιλογή τίτλου
- Η εισαγωγή κειμένου, εικόνων και υπερσυνδέσεων
- Η χρωματική εναλλαγή στην παρουσίαση τιμών που υπερβαίνουν κάποια όρια (π.χ. κόκκινο για τις αρνητικές τιμές αριθμητικής στήλης).

## ΠΑΡΑΡΤΗΜΑ Α: ΤΟ ΣΤΟΙΧΕΙΟ ΕΛΕΓΧΟΥ ΚΕΛΙΩΝ

Το στοιχείο ελέγχου κελιών που χρησιμοποιήθηκε στην εφαρμογή SuiteCase είναι μια δουλειά του Chris Maunder, ιδρυτή και διαχειριστή του δικτυακού χώρου [www.codeproject.com](http://www.codeproject.com). Η δουλειά αυτή βασίστηκε στην δουλειά άλλων ατόμων και συνεισέφεραν πολλοί στην ανάπτυξή του με διορθώσεις και συστάσεις.

Η άδεια χρήσης αυτού του προϊόντος έχει ως εξής:

Ο κώδικας μπορεί να χρησιμοποιηθεί σε μεταφρασμένη μορφή με οποιονδήποτε τρόπο (ακόμα και εμπορικό). Ο κώδικας μπορεί να διανεμηθεί ξανά χωρίς τροποποιήσεις με οποιοδήποτε μέσο, με την προϋπόθεση ότι δεν έχει εμπορικούς σκοπούς χωρίς την γραπτή άδεια του συγγραφέα, και ότι όλα τα σημειώματα διατήρησης των πνευματικών δικαιωμάτων παραμένουν ανεπηρέαστα. Ο κώδικας και το κείμενο που τον συνοδεύει δεν πρέπει να δημοσιευτεί σε άλλο δικτυακό χώρο χωρίς την γραπτή άδεια του συγγραφέα.

Το προϊόν παρέχεται «έτσι όπως είναι» χωρίς εγγύηση. Οποιοδήποτε ρίσκο εμπίπτει στον χρήστη του.

Οι δυνατότητες που παρέχει το στοιχείο ελέγχου κελιών είναι:

- Επιλογή κελιών με το ποντίκι και τα πλήκτρα Ctrl και Shift.
- Προσαρμογή μεγέθους στηλών και γραμμών.
- Αυτόματη προσαρμογή μεγέθους στηλών ανάλογα με τα περιεχόμενα με διπλό click.
- Οποιοδήποτε πλήθος στηλών και γραμμών.
- Κάθε κελί μπορεί να έχει δική του ξεχωριστή γραμματοσειρά, χρώμα φόντου, προστασία διαγραφής.
- Δυνατότητα Drag and Drop.
- Δυνατότητα επιλογής όλων, αντιγραφής, αποκοπής και επικόλλησης.
- Επεξεργασία δεδομένων επάνω στο στοιχείο ελέγχου κελιών.
- Επιλογή απόκρυψης διαχωριστικών γραμμών.
- Εικόνες σε κάθε κελί.
- «Εικονική» λειτουργία για μεγάλα σύνολα δεδομένων.
- Υποστήριξη εκτύπωσης.
- Επιλογή λειτουργίας ως λίστα.
- Εύκολη επέκταση λειτουργιών.
- Υποστήριξη Unicode.
- Υποστήριξη WinCE.
- Βοηθητικά παράθυρα εμφάνισης δεδομένων για κελιά τα οποία είναι μικρά σχετικά με τα περιεχόμενά τους.
- Κρυφές στήλες και γραμμές.
- Υποστήριξη παλαιότερων εκδόσεων του Visual C++ compiler.

Για περισσότερες πληροφορίες επισκεφθείτε την ιστοσελίδα:  
[www.codeproject.com/miscctrl/gridctrl.asp](http://www.codeproject.com/miscctrl/gridctrl.asp)



## ΠΑΡΑΡΤΗΜΑ Β: ΜΕΤΑΤΡΟΠΗ ΔΕΔΟΜΕΝΩΝ

Τα μεγάλα συστήματα διαχείρισης βάσεων δεδομένων λειτουργούν και χρησιμοποιούνται σε πολλές χώρες του κόσμου, με χρήση βέβαια των αντίστοιχων γλωσσών. Τα υπολογιστικά συστήματα, ως γνωστόν, χρησιμοποιούν 8 bits για την αποθήκευση ενός χαρακτήρα. Αυτά τα συστήματα υποστηρίζουν πίνακες 256 χαρακτήρων. Οι ανατολικές κυρίως γλώσσες περιέχουν περισσότερα από 256 χαρακτήρες και γι' αυτό τα συστήματα διαχείρισης βάσεων δεδομένων χρησιμοποιούν πίνακες περισσότερων χαρακτήρων χρησιμοποιώντας 16 bits ή 2 bytes για την αποθήκευση των δεδομένων.

Το σύστημα χρήσης 2 bytes για την αποθήκευση των χαρακτήρων ονομάζεται WideChar και το σύστημα χρήσης 1 byte για την αποθήκευση χαρακτήρων ονομάζεται συγκριτικά MultiByte. Για την μετατροπή των δεδομένων από το ένα σύστημα στο άλλο υπάρχουν στις βιβλιοθήκες της Visual C++ οι συναρτήσεις WideCharToMultiByte και MultiByteWideChar. Οι αντίστοιχοι τύποι δεδομένων είναι ο wchar\_t για τους WideChar και ο char για τους MultiByte χαρακτήρες.

Τα πρότυπα των συναρτήσεων WideCharToMultiByte και MultiByteToWideChar έχουν ως εξής:

```
int WideCharToMultiByte(
    UINT CodePage,           // code page
    DWORD dwFlags,         // performance and mapping flags
    LPCWSTR lpWideCharStr,  // wide-character string
    Int cchWideChar,       // number of chars in string
    LPSTR lpMultiByteWideChar, // buffer for new string
    Int cbMultiByte,       // size of buffer
    LPCSTR lpDefaultChar,  // default for unmappable chars
    LPBOOL lpUsedDefaultChar // set when default char used
);
```

```
int MultiByteToWideChar(
    UINT CodePage,           // code page
    DWORD dwFlags,         // character-type options
    LPCSTR lpMultiByteStr,  // string to map
    Int cbMultiByte,       // number of bytes in string
    LPWSTR lpWideCharStr   // wide-character buffer
    int cchWideChar       // size of buffer
);
```

Οι μετατροπές των δεδομένων μεταξύ αυτών των δύο τύπων χρησιμοποιούνται αρκετά συχνά στην εφαρμογή SuiteCase και γι' αυτό σχεδιάστηκαν δύο συναρτήσεις που διευκολύνουν την διαδικασία χρησιμοποιώντας τις παραπάνω συναρτήσεις:

```
Void GiveMeString(WCHAR* lpszSource, char* strDest)
{
    int much;
    much = WideCharToMultiByte(
        CP_ACP,
        WC_COMPOSITECHECK|WC_SEPCHARS|WC_DISCARDNS,
        lpszSource,
        wcslen(lpszSource),
        strDest,
        wcslen(lpszSource),
        NULL,
        NULL);
    strDest[much] = '\\0';
}
```

```
Void GiveMeWideChar(char* strSource, WCHAR* lpszDest)
{
    int much;
    much = MultiByteToWideChar(
        CP_ACP,
        MB_PRECOMPOSED,
        strSource,
        strlen(strSource),
        lpszDest,
        strlen(strSource));
    lpszDest[much] = L'\\0';
}
```

## ΠΑΡΑΡΤΗΜΑ Γ: ΑΡΧΕΙΑ ΕΠΙΚΕΦΑΛΙΔΩΝ IMPORT

```
// Suitecase project
// Import header file for DataView library
BOOL __declspec(dllimport) ShowDataView(
    HWND hWnd,
    CMasterOledb* master);

// SuiteCase Project
// Import header file for ScriptEdit library
BOOL __declspec(dllimport) ShowScriptEdit(
    HWND hWnd,
    CMasterOledb* master);

// SuiteCase Project
// Import header file for the QueryWizard library
BOOL __declspec(dllimport) ShowQueryWizard(
    HWND hWnd,
    CMasterOledb* master);

// Suitecase project
// Import header file for the DDL library
#ifdef IMPORT_DDL
#define DDL
void __declspec(dllimport) CreateTableDialog(
    CMasterOledb* master);

void __declspec(dllimport) CreateViewDialog(
    CMasterOledb* master,
    CString* selectstatement = NULL);

void __declspec(dllimport) CreateIndexDialog(
    CMasterOledb* master,
    CString* tablename = NULL,
    CString* columnname = NULL);

void __declspec(dllimport) CreateDatabaseDialog(
    CMasterOledb* master);

void __declspec(dllimport) DropTableDialog(
    CMasterOledb* master,
    CString* tablename = NULL);

void __declspec(dllimport) DropViewDialog(
    CMasterOledb* master,
```

```
CString* viewname = NULL);

void __declspec(dllimport) DropIndexDialog(
    CMasterOledb* master,
    CString* tablename = NULL,
    CString* indexname = NULL);

void __declspec(dllimport) DropDatabaseDialog(
    CMasterOledb* master);

void __declspec(dllimport) AddColumnDialog(
    CMasterOledb* master,
    CString* tablename = NULL);

void __declspec(dllimport) DelColumnDialog(
    CMasterOledb* master,
    CString* tablename = NULL,
    CString* columnname = NULL);

#endif

// SuiteCase project
// Import header file for DML library
void __declspec(dllimport) GetPKColumnNames(
    CMasterOledb* master,
    CString* strTblName,
    int* iPKColumns,
    CString** strPKColumns);

BOOL __declspec(dllimport) GetColumnIsNumeric(
    CMasterOledb* master,
    CString* strTableName,
    CString* strColumnName);

// row data manipulation
void __declspec(dllimport) InsertRowDialog(
    CMasterOledb* master,
    CString* strTableName);

void __declspec(dllimport) UpdateRowDialog(
    CMasterOledb* master,
    CString* strTableName,
    CString** aPredefinedData);

void __declspec(dllimport) DeleteRow(
    CMasterOledb* master,
    CString* strTableName,
    int count,
    CString** ppColumnNames,
    CString** ppPredefinedData);

void __declspec(dllimport) CopyRowDialog(
    CMasterOledb* master,
    CString* strTableName,
    CString** aPredefinedData);
```

---

```
// SuiteCase project
// Import header file for Report library
BOOL __declspec(dllexport) ShowDefineFormDialog(
    CMasterOledb* object,
    CString* strTableName = NULL);

BOOL __declspec(dllexport) ShowReportDialog(
    CMasterOledb* object,
    CString* strSelectStatement = NULL);
```

## ΠΑΡΑΡΤΗΜΑ Δ: ΒΙΒΛΙΟΓΡΑΦΙΑ

### ΒΙΒΛΙΑ

#### 1. ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΓΙΑ ΤΗ ΔΙΟΙΚΗΣΗ ΕΠΙΧΕΙΡΗΣΕΩΝ (ΤΟΜΟΣ Α')

Γεώργιος Σ. Οικονόμου, Νικόλαος Β. Γεωργόπουλος  
Εκδόσεις Ευγ. Μπένου, Αθήνα 1995

#### 2. ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΔΙΟΙΚΗΣΗΣ

Γ. Βασιλακόπουλος, Β. Χρυσικόπουλος  
Εκδόσεις Α. Σταμούλης, Πειραιάς 1990

#### 3. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΤΟ MICROSOFT SQL SERVER 2000

Rebecca M. Riordan  
Εκδόσεις Κλειδάριθμος, Ιούλιος 2001

#### 4. ΣΧΕΣΙΑΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ – ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΥΛΟΠΟΙΗΣΗ

Χρήστος Σκουρλάς  
Εκδόσεις Νέων Τεχνολογιών, 1997

#### 5. ΕΙΣΑΓΩΓΗ ΣΤΑ ΣΥΣΤΗΜΑΤΑ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ (ΤΟΜΟΣ Α')

C. J. Date  
Εκδόσεις Κλειδάριθμος, 1996

#### 6. ΕΓΧΕΙΡΙΔΙΟ C++

Jesse Liberty  
Εκδόσεις Μ.Γκιούρδας, 1999

#### 7. VISUAL C++ 6 – ΒΗΜΑ ΠΡΟΣ ΒΗΜΑ

Steven Holzner  
Εκδόσεις Μ. Γκιούρδας, 1999

#### 8. DATA STRUCTURES AND OTHER OBJECTS USING C++

Michael Main, Walter Savitch  
Εκδόσεις Addison Wesley, 2001

#### 9. MASTERING ACTIVE SERVER PAGES 3

A. Russell Jones  
Εκδόσεις SYBEX, 2000

### ELECTRONIC DOCUMENTATION

#### 10. MSDN Library October 2000

**11. SQL Server Books Online****12. Client/Server Term Definition in Encyclopedia of Computer Science**

D. Hemmendinger, A. Ralston, E.D. Reilly

International Thomson Computer Publishing, 1998

Silvano Maffeis, Olsen & Associates, Zurich, December 5, 1997

**13. Extending the Database Relational Model to Capture More Meaning**

E. F. Codd, IBM Research Laboratory

Presented at the 1979 International Conference on Management of Data

**INTERNET**

14. [www.codeguru.com](http://www.codeguru.com)

& [www.codeguru.com/forum](http://www.codeguru.com/forum)

15. [www.codeproject.com](http://www.codeproject.com)