

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΟΡΥΚΤΩΝ ΠΟΡΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

**ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΗΝ ΑΝΑΛΥΣΗ
ΠΕΙΡΑΜΑΤΙΚΩΝ ΔΕΔΟΜΕΝΩΝ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΠΕΤΡΩΜΑΤΩΝ**

ΒΑΛΛΑΣ ΘΩΜΑΣ
ΜΗΧΑΝΙΚΟΣ ΟΡΥΚΤΩΝ ΠΟΡΩΝ

ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

ΖΑΧΑΡΙΑΣ ΑΓΙΟΥΤΑΝΤΗΣ, ΚΑΘΗΓΗΤΗΣ (Επιβλέπων)
ΓΕΩΡΓΙΟΣ ΕΞΑΔΑΚΤΥΛΟΣ, ΚΑΘΗΓΗΤΗΣ
ΜΙΧΑΛΗΣ ΓΑΛΕΤΑΚΗΣ, ΛΕΚΤΟΡΑΣ

ΧΑΝΙΑ
ΜΑΡΤΙΟΣ 2006

ΠΡΟΛΟΓΟΣ

Για την εκπόνηση της μεταπτυχιακής διατριβής, είχα την ουσιαστική βοήθεια και συμπαράσταση ορισμένων προσώπων τα οποία θα ήθελα να ευχαριστήσω:

Τον Επιβλέποντα της μεταπτυχιακής διατριβής, Καθηγητή του Πολυτεχνείου Κρήτης κ. Αγιουτάντη Ζαχαρία, για τη συνεργασία, την καθοδήγηση και τις ουσιαστικές συμβουλές του κατά τη διάρκεια εκπόνησης της μεταπτυχιακής μου διατριβής.

Τον Αναπληρωτή Καθηγητή του Πολυτεχνείου Κρήτης κ. Εξαδάκτυλο Γεώργιο και τον Λέκτορα του Πολυτεχνείου Κρήτης κ. Γαλετάκη Μιχάλη, για τις παρατηρήσεις και την συμμετοχή τους στην αξιολόγηση της διατριβής αυτής.

Τον Μηχανικό Μεταλλείων κ. Μαυριγιαννάκη Στυλιανό, για την ουσιαστική συνεργασία που είχαμε κατά την διάρκεια φοίτησης μου στο μεταπτυχιακό πρόγραμμα σπουδών “Γεωτεχνολογία και Περιβάλλον” και για την βοήθεια του στην εκπόνηση αυτής της μεταπτυχιακής διατριβής.

Τον προγραμματιστή κ. Χερουβείμ Ιωάννη, για τις πολύτιμες και ουσιαστικές συμβουλές του σε θέματα βάσεων δεδομένων και προγραμματισμού.

Τον Καθηγητή Δευτεροβάθμιας Εκπαίδευσης κ. Καγιά Μανόλη και την φοιτήτρια του τμήματος Φυσικών Πόρων και Περιβάλλοντος του Α.Τ.Ε.Ι Κρήτης Παπαδημητρίου Γεωργία για την αμέριστη βοήθεια και συμπαράστασή τους κατά τη διάρκεια της εκπόνησης της διατριβής αυτής.

Τέλος θα ήθελα να ευχαριστήσω τον πατέρα μου Δημήτρη, την μητέρα μου Ελένη καθώς και όλους όσους με στήριξαν καθ’ όλη τη διάρκεια της φοίτησης μου στο μεταπτυχιακό πρόγραμμα σπουδών “Γεωτεχνολογία και Περιβάλλον” του τμήματος Μηχανικών Ορυκτών Πόρων του Πολυτεχνείου Κρήτης.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ	1
----------------	---

ΚΕΦΑΛΑΙΟ 2

ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΤΗ VISUAL BASIC .NET	
2.1 Εισαγωγή	3
2.2 Γλώσσες προγραμματισμού.....	3
2.3 Visual Basic .NET	5
2.4 Το .NET Framework και το Common Language Runtime.....	7
2.5 Το περιβάλλον της Visual Basic .NET	8
2.5.1 Εργαλεία	9
2.5.2 Γεγονότα	10
2.5.3 Ιδιότητες	10
2.5.4 Ολοκληρωμένο περιβάλλον ανάπτυξης	11
2.6 Δομή ενός προγράμματος σε Visual Basic .NET	14
2.7 Τύποι δεδομένων, αριθμητικοί τελεστές και τελεστές σύγκρισης.....	15
2.8 Δομές ελέγχου.....	18
2.9 Ανίχνευση και διόρθωση σφαλμάτων στη Visual Basic .NET	19
2.10 Γιατί επιλέχτηκε η Visual Basic .NET.....	20

ΚΕΦΑΛΑΙΟ 3

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΔΟΚΙΜΕΣ ΜΗΧΑΝΙΚΗΣ ΠΕΤΡΩΜΑΤΩΝ	
3.1 Εισαγωγή	22
3.2 Ιδιότητες των πετρωμάτων.....	22
3.3 Φυσικές παράμετροι των πετρωμάτων	22
3.4 Μηχανικές παράμετροι των πετρωμάτων.....	24
3.4.1 Παράμετροι αντοχής	24
3.4.2 Εσωτερική τριβή και συνοχή	25
3.4.3 Παράμετροι ελαστικότητας	26
3.5 Αστοχία ενός υλικού.....	26
3.6 Επίδραση ασυνεχειών στη μηχανική συμπεριφορά του πετρώματος.....	27
3.7 Δειγματοληψία και προπαρασκευή δοκιμών	28
3.8 Προσδιορισμός της αντοχής σε μονοαξονική θλίψη	28
3.9 Προσδιορισμός καμπύλης τάσης-παραμόρφωσης και ελαστικών σταθερών	29
3.10 Έμμεσος προσδιορισμός της αντοχής σε εφελκυσμό.....	32
3.11 Προσδιορισμός της αντοχής σε τριαξονική καταπόνηση	33
3.12 Προσδιορισμός της αντοχής σε κάμψη	35
3.13 Προσδιορισμός ταχυτήτων διάδοσης θλιπτικών και διατμητικών κυμάτων	36

ΚΕΦΑΛΑΙΟ 4

ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΔΙΑΧΕΙΡΗΣΗ ΠΕΙΡΑΜΑΤΩΝ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΠΕΤΡΩΜΑΤΩΝ	
4.1 Εισαγωγή	38
4.2 Βάσεις δεδομένων.....	38
4.3 Σύστημα βάσεων δεδομένων MySQL	39
4.4 Πίνακες και πρωτεύοντα κλειδιά	41
4.5 Βασικοί τύποι δεδομένων του MySQL.....	42
4.6 Δομημένη γλώσσα ερωτημάτων (SQL).....	43
4.6.1 Η βασική εντολή δημιουργίας πίνακα CREATE	43
4.6.2 Το βασικό ερώτημα SELECT	44
4.6.3 Σχέση «ένα προς πολλά».....	46

4.6.4 Η φράση ORDER BY	48
4.6.5 Η φράση INSERT	49
4.6.6 Η φράση UPDATE.....	50
4.6.7 Η φράση DELETE	51
4.6.8 Η φράση LIMIT	51
4.6.9 Η φράση AS	52
4.6.10 Οι συναρτήσεις COUNT και SUM.....	53
4.6.11 Οι συναρτήσεις MAX και MIN	53
4.7 Πρόγραμμα MySQL-Front	54
4.8 Βάση δεδομένων rmlab	58
4.8.1 Πίνακας master	59
4.8.2 Πίνακας uniaxial	60
4.8.3 Πίνακας bending	61
4.8.4 Πίνακας triaxial	62
4.8.5 Πίνακας tension	63
4.8.6 Πίνακας ultrasonic	63
4.8.7 Δομή και σχέσεις	64
ΚΕΦΑΛΑΙΟ 5	
ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΔΙΑΧΕΙΡΙΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΠΕΤΡΩΜΑΤΩΝ	
5.1 Εισαγωγή	66
5.2 Διασύνδεση Visual Basic .NET με το σύστημα MySQL.....	66
5.3 Φόρμα σύνδεσης με τη βάση δεδομένων rmlab	68
5.4 Φόρμα προσθήκης δεδομένων στη βάση rmlab	70
5.4.1 Φόρτωση των επιλογών των comboboxes	74
5.4.2 Προσθήκη εγγραφών στους υπό-πίνακες.....	75
5.4.3 Προσθήκη εγγραφών στους βασικούς πίνακες	76
5.5 Φόρμα αναζήτησης εγγραφών	79
5.5.1 Λειτουργία προβολής εγγραφών (view mode)	81
5.5.2 Λειτουργία αναζήτησης (search mode)	83
5.6 Φόρμα επεξεργασίας εγγραφών	85
5.7 Φόρμα διαγραφής εγγραφών	90
5.8 Φόρμα σχεδιασμού διαγραμμάτων.....	91
ΚΕΦΑΛΑΙΟ 6	
ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΡΟΤΑΣΕΙΣ	
6.1 Σύνοψη	94
6.2 Προτάσεις.....	95
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	97

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Ο σκοπός της μεταπτυχιακής αυτής διατριβής είναι η κατασκευή μιας βάσης δεδομένων για την ομαδοποίηση και ταξινόμηση των δεδομένων που προκύπτουν από τις πειραματικές δοκιμές της Μηχανικής Πετρωμάτων. Επιπλέον στόχος της διατριβής αποτελεί και η κατασκευή ενός λογισμικού σε περιβάλλον Windows με το οποίο θα είναι δυνατόν να εκτελεστούν με ευκολία οι βασικές λειτουργίες της βάσης δεδομένων. Πιο συγκεκριμένα μέσω του λογισμικού αυτού είναι δυνατόν να παρουσιαστούν, να ταξινομηθούν και να αναζητηθούν οι εγγραφές της βάσης δεδομένων. Παράλληλα το λογισμικό θα πρέπει να δίνει τη δυνατότητα προσθήκης, επεξεργασίας και διαγραφής εγγραφών.

Επιπρόσθετα μέσω του λογισμικού αυτού θα υπάρχει η δυνατότητα σχεδιασμού διαγραμμάτων συγκεκριμένων παραμέτρων μιας ομάδας εγγραφών της βάσης δεδομένων, με στόχο την οπτικοποίηση και καλύτερη ανάλυση πληθώρας αριθμητικών πειραματικών δεδομένων.

Το πρόγραμμα αυτό απευθύνεται τόσο σε μηχανικούς που εκτελούν πειραματικές δοκιμές Μηχανικής Πετρωμάτων όσο και σε άλλους μηχανικούς και φοιτητές που επιθυμούν να κατανοήσουν τη μεθοδολογία σχεδίασης και ανάπτυξης μιας οποιασδήποτε βάσης δεδομένων καθώς και υλοποίησης λογισμικού – πελάτη για τη βάση αυτή, για την ομαδοποίηση οποιουδήποτε είδους πειραματικών δεδομένων.

Η επιλογή των πειραματικών δοκιμών και των παραμέτρων τους, που απαρτίζουν τη βάση δεδομένων έγινε σε συνεργασία με τον Καθηγητή του Πολυτεχνείου Κρήτης κ. Αγιουτάντη Ζαχαρία και είχε σαν στόχο την όσο το δυνατόν πληρέστερη και λεπτομερέστερη αποθήκευση πληροφοριών σχετικά με πειράματα της Μηχανικής Πετρωμάτων.

Κατά την συγγραφή της μεταπτυχιακής διατριβής έγινε μεγάλη προσπάθεια να αντιστοιχηθούν οι αγγλικοί όροι που περιγράφουν στοιχεία και λειτουργίες της γλώσσας προγραμματισμού και της βάσης δεδομένων, με ελληνικούς όρους. Η προσπάθεια αυτή δεν έγινε δυνατόν να υλοποιηθεί πλήρως εφ' όσον πάρα πολλοί όροι είτε δεν είναι δυνατόν να αποδοθούν στα ελληνικά είτε και να αποδοθούν χάνουν το νόημα τους. Έτσι πολλές φορές

στο κείμενο παρεμβάλλονται αγγλικοί όροι με στόχο την καλύτερη κατανόηση των διαδικασιών που περιγράφονται.

Η βάση δεδομένων που κατασκευάστηκε ονομάστηκε rmlab από τα αρχικά των λέξεων Rock Mechanics Laboratory (Εργαστήριο Μηχανικής Πετρωμάτων). Από τα ίδια αρχικά προέκυψε και η ονομασία του λογισμικού που είναι RockLab.

ΚΕΦΑΛΑΙΟ 2

ΓΕΝΙΚΑ ΣΤΟΙΧΕΙΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΣΤΗ VISUAL BASIC .NET

2.1 Εισαγωγή

Στο κεφάλαιο αυτό αρχικά παρουσιάζονται κάποια ιστορικά στοιχεία για την εξέλιξη των υπολογιστών και των διαφόρων γλωσσών προγραμματισμού. Επιπλέον γίνεται μια σύντομη περιγραφή της γλώσσας προγραμματισμού Visual Basic .NET που χρησιμοποιήθηκε κατά την ανάπτυξη του λογισμικού που περιγράφεται στην εργασία αυτή.

2.2 Γλώσσες προγραμματισμού

Η χρήση των υπολογιστών εξαπλώνεται σχεδόν σε κάθε πεδίο της επιστήμης. Σε μια εποχή σταθερής ανόδου του κόστους, το υπολογιστικό κόστος έχει μειωθεί σημαντικά, λόγω της ταχείας ανάπτυξης τόσο της τεχνολογίας υλικού όσο και της τεχνολογίας λογισμικού. Οι υπολογιστές που κάλυπταν τεράστιους χώρους και κόστιζαν εκατομμύρια δολάρια μόλις δύο δεκαετίες πριν, σήμερα μπορούν να «εγχαραχθούν» στις επιφάνειες ολοκληρωμένων κυκλωμάτων πυριτίου, μικρότερων από ένα γραμματόσημο, και κοστίζουν λίγα δολάρια ο καθένας. Το πυρίτιο είναι από τα υλικά με άφθονα αποθέματα στη γη καθώς αποτελεί το συστατικό της απλής άμμου. Η τεχνολογία ολοκληρωμένων κυκλωμάτων πυριτίου έχει καταστήσει τη λειτουργία των υπολογιστών τόσο οικονομική, ώστε εκατοντάδες εκατομμύρια υπολογιστών γενικής χρήσης να λειτουργούν σε ολόκληρο τον κόσμο, βοηθώντας τους ανθρώπους στις επιχειρήσεις, στη βιομηχανία, στις κυβερνήσεις και στην προσωπική τους ζωή.

Ένας υπολογιστής είναι μια συσκευή που μπορεί να εκτελέσει υπολογισμούς και να πάρει λογικές αποφάσεις, με ταχύτητες εκατομμύρια ή και δισεκατομμύρια φορές μεγαλύτερες εκείνων των ανθρώπινων όντων. Οι υπολογιστές επεξεργάζονται τα δεδομένα υπό τον έλεγχο συνόλων οδηγιών, τα οποία ονομάζονται προγράμματα υπολογιστών (programs). Αυτά τα προγράμματα οδηγούν τους υπολογιστές μέσω μεθοδικών συνόλων ενεργειών, που καθορίζονται από τους γνωστούς ως προγραμματιστές υπολογιστών (developers) (Deitel κ.α., 2002).

Οι προγραμματιστές γράφουν οδηγίες σε διάφορες γλώσσες προγραμματισμού (programming languages), μερικές από τις οποίες είναι άμεσα κατανοητές από τους υπολογιστές ενώ άλλες απαιτούν ενδιάμεσα βήματα μεταγλώττισης (compile). Αν και εκατοντάδες γλώσσες προγραμματισμού χρησιμοποιούνται σήμερα, τα διάφορα προϊόντα χωρίζονται σε τρεις γενικές κατηγορίες (Deitel κ.α., 2002):

- Γλώσσες μηχανής
- Συμβολομεταφραστικές γλώσσες
- Γλώσσες υψηλού επιπέδου

Κάθε υπολογιστής μπορεί να κατανοήσει μόνο τη δική του γλώσσα μηχανής άμεσα. Ως η «φυσική γλώσσα» ενός συγκεκριμένου υπολογιστή η γλώσσα μηχανής ορίζεται από τη σχεδίαση του υλικού του υπολογιστή. Οι γλώσσες μηχανής γενικώς αποτελούνται από ακολουθίες αριθμών (τελικώς περιορισμένων σε διαδοχικά 0 και 1) που οδηγούν τους υπολογιστές να εκτελέσουν τις πλέον στοιχειώδεις λειτουργίες τους. Οι γλώσσες μηχανής είναι εξαρτώμενες από τη μηχανή, το οποίο σημαίνει ότι μια συγκεκριμένη γλώσσα μηχανής μπορεί να χρησιμοποιηθεί μόνο σε έναν τύπο υπολογιστή (Deitel κ.α., 2002).

Καθώς αυξανόταν η δημοτικότητα των υπολογιστών, ο προγραμματισμός σε γλώσσες μηχανής αποδείχθηκε εξαιρετικά αργός, ανιαρός και επιρρεπής σε σφάλματα. Αντί να χρησιμοποιούνται ακολουθίες αριθμών που οι υπολογιστές κατανοούν άμεσα, οι προγραμματιστές άρχισαν να χρησιμοποιούν συντομεύσεις της αγγλικής γλώσσας για να αναπαραστήσουν τις στοιχειώδεις λειτουργίες του υπολογιστή. Αυτές οι συντομεύσεις διαμόρφωσαν τη βάση των συμβολομεταφραστικών γλωσσών. Μεταφραστικά προγράμματα που λέγονται assemblers μετατρέπουν τα προγράμματα των συμβολομεταφραστικών γλωσσών σε γλώσσα μηχανής, με τις ταχύτητες του υπολογιστή (Deitel κ.α., 2002).

Αν και η χρήση των υπολογιστών αυξήθηκε ραγδαία με την έλευση των συμβολομεταφραστικών γλωσσών, αυτές οι γλώσσες εξακολουθούσαν να απαιτούν πολλές οδηγίες για να τελέσουν απλές διεργασίες. Για να επιταχυνθεί η διαδικασία του προγραμματισμού, αναπτύχθηκαν γλώσσες υψηλού επιπέδου, στις οποίες απλές προτάσεις τελούν σημαντικές διεργασίες. Μεταφραστικά προγράμματα που ονομάζονται μεταγλωττιστές (compilers) μετατρέπουν τα προγράμματα γλωσσών υψηλού επιπέδου σε γλώσσα μηχανής. Οι γλώσσες υψηλού επιπέδου δίνουν τη δυνατότητα στους

προγραμματιστές να γράφουν οδηγίες που μοιάζουν με τα καθομιλούμενα αγγλικά και περιέχουν συνήθη μαθηματικά σύμβολα (Deitel κ.α., 2002).

Η μεταγλώττιση ενός προγράμματος γλώσσας υψηλού επιπέδου σε γλώσσα μηχανής μπορεί να απαιτήσει ένα σημαντικό χρονικό διάστημα. Αυτό το πρόβλημα επιλύθηκε από την ανάπτυξη των προγραμμάτων διερμηνεύσης (interpreters), που μπορούν να εκτελέσουν άμεσα προγράμματα γλωσσών υψηλού επιπέδου, παρακάμπτοντας το βήμα της μεταγλώττισης. Αν και τα ήδη μεταγλωττισμένα προγράμματα εκτελούνται ταχύτερα απ' ό,τι τα διερμηνευμένα, οι διερμηνευτές είναι δημοφιλείς στα περιβάλλοντα ανάπτυξης προγραμμάτων. Σε τέτοια περιβάλλοντα, οι προγραμματιστές αλλάζουν συχνά τα προγράμματα καθώς προσθέτουν νέα χαρακτηριστικά και διορθώνουν σφάλματα. Όταν ένα πρόγραμμα έχει αναπτυχθεί πλήρως, μια μεταγλωττισμένη έκδοση μπορεί να δημιουργηθεί, έτσι ώστε το πρόγραμμα να εκτελείται με τη βέλτιστη αποτελεσματικότητα (Deitel κ.α., 2002).

2.3 Visual Basic .NET

Η Visual Basic .NET αναπτύχθηκε από την BASIC (Beginner's All-Purpose Symbolic Instruction Code), που είχε αναπτυχθεί στα μέσα της δεκαετίας του '60 από τους John Kemeny και Thomas Kurtz, καθηγητές του Dartmouth College, ως μια γλώσσα για την εγγραφή απλών προγραμμάτων. Πρωταρχικός στόχος της BASIC ήταν η εξοικείωση των αρχαρίων με τις τεχνικές προγραμματισμού. Η ευρεία χρήση της BASIC σε διάφορους τύπους υπολογιστών οδήγησε σε πολλές βελτιώσεις στη γλώσσα. Όταν ο Bill Gates ίδρυσε τη Microsoft Corporation, υλοποίησε τη BASIC σε διάφορους από τους πρώτους προσωπικούς υπολογιστές (Deitel κ.α., 2002).

Μέχρι την εμφάνιση της Visual Basic το 1991, η ανάπτυξη εφαρμογών βασισμένων στα Microsoft Windows ήταν μια δύσκολη και επίπονη διαδικασία. Η γλώσσα Visual Basic είναι μια σαφώς διαφορετική από την γλώσσα BASIC που προσφέρει δυνατότητες όπως τα γραφικά περιβάλλοντα χρηστών, το χειρισμό συμβάντων, την πρόσβαση στο Windows 32-bit Application Programming Interface (Win32 API), τον αντικειμενοστραφή προγραμματισμό και το χειρισμό των εξαιρέσεων. Η Visual Basic .NET είναι μία απεικονιστική, οδηγούμενη από συμβάντα γλώσσα προγραμματισμού, στην οποία τα προγράμματα δημιουργούνται χρησιμοποιώντας ένα Ολοκληρωμένο Περιβάλλον Ανάπτυξης (ΟΠΑ, Integrated Development Environment – IDE). Με το ΟΠΑ, ένας προγραμματιστής

μπορεί να γράψει, να εκτελέσει, να ελέγξει και να διορθώσει αποτελεσματικά προγράμματα της Visual Basic μειώνοντας το χρόνο που απαιτείται για την ανάπτυξη ενός προγράμματος. Η διαδικασία της γρήγορης δημιουργίας μιας εφαρμογής αναφέρεται ως Rapid Application Development – RAD. Η Visual Basic είναι η πιο διαδεδομένη γλώσσα RAD που χρησιμοποιείται σήμερα (Deitel κ.α., 2002).

Η ολοκλήρωση τμημάτων λογισμικού από τις διάφορες γλώσσες, αποδείχθηκε δύσκολη και τα προβλήματα εγκατάστασης ήταν συχνά, επειδή οι νέες εκδόσεις των διανεμόμενων συστατικών ήταν ασύμβατες με το παλιό λογισμικό. Οι προγραμματιστές ανακάλυψαν επίσης ότι χρειαζόνταν εφαρμογές βασισμένες στο Web, που θα μπορούσαν να προσπελαστούν και να χρησιμοποιηθούν μέσω του διαδικτύου. Καθώς οι προγραμματιζόμενες συσκευές, όπως τα personal digital assistants (PDAs) και τα κινητά τηλέφωνα, γίνονταν δημοφιλή στα τέλη της δεκαετίας του '90, η ανάγκη για να αλληλεπιδρούν τα τμήματα αυτά με άλλα μέσω του διαδικτύου έγινε τεράστια. Οι προγραμματιστές αναγνώρισαν την ανάγκη για λογισμικό προσπελάσιμο από σχεδόν κάθε τύπο συσκευής (Deitel κ.α., 2002).

Για να ικανοποιήσει τις ανάγκες αυτές η Microsoft ανακοίνωσε το 2000 την εισαγωγή της στρατηγικής Microsoft .NET. Η πλατφόρμα .NET είναι μια πλατφόρμα πάνω από την οποία μπορούν να διανεμηθούν εφαρμογές βασισμένες στο Web σε μια ποικιλία συσκευών όπως σε κινητά τηλέφωνα και σε προσωπικούς υπολογιστές. Η πλατφόρμα .NET προσφέρει ένα νέο μοντέλο προγραμματισμού που επιτρέπει στα προγράμματα να δημιουργούνται σε ετερόκλητες γλώσσες προγραμματισμού και να επικοινωνούν μεταξύ τους (Deitel κ.α., 2002).

Η Microsoft σχεδίασε μια έκδοση της Visual Basic για το .NET. Οι προγενέστερες εκδόσεις της Visual Basic προσέφεραν αντικειμενοστραφείς δυνατότητες, αλλά η Visual Basic .NET προσφέρει βελτιωμένη αντικειμενοστρέφεια και δίνει τη δυνατότητα προωθημένης διαλειτουργικότητας των γλωσσών. Οι προγραμματιστές μπορούν να χρησιμοποιούν ακόμα και παλιό λογισμικό για να συνεργάζεται με τα νέα προγράμματα της Visual Basic .NET. Οι βελτιώσεις που ενσωματώθηκαν στη Visual Basic .NET θα οδηγήσουν σε ένα νέο προγραμματιστικό στυλ, στο οποίο οι εφαρμογές δημιουργούνται από συστατικά, αποκαλούμενες υπηρεσίες Web, διαθέσιμα μέσω διαδικτύου (Deitel κ.α., 2002).

2.4 Το .NET Framework και το Common Language Runtime

Το Microsoft .NET Framework είναι η καρδιά της στρατηγικής .NET. Αυτό το πλαίσιο εργασίας διαχειρίζεται και εκτελεί εφαρμογές και υπηρεσίες Web, περιέχει μια βιβλιοθήκη κλάσεων (την framework class library ή FCL), ενισχύει την ασφάλεια και παρέχει πολλές άλλες προγραμματιστικές δυνατότητες. Οι λεπτομέρειες του .NET Framework δίνονται στο Common Language Specification (CLS), που περιέχει πληροφορίες για την αποθήκευση των τύπων δεδομένων καθώς και τα αντικείμενα. Το CLS έχει υποβληθεί για τυποποίηση στην ECMA (European Computer Manufacturers Association) καθιστώντας ευκολότερη τη δημιουργία του .NET Framework για άλλες πλατφόρμες. Παρόλα αυτά σήμερα το .NET Framework υπάρχει μόνο για τη πλατφόρμα Microsoft Windows (Deitel κ.α., 2002).

Το Common Language Runtime (CLR) είναι ένα άλλο ζωτικό μέρος του .NET Framework και εκτελεί προγράμματα Visual Basic .NET. Τα προγράμματα μεταγλωττίζονται σε οδηγίες συγκεκριμένες για τη μηχανή, σε δύο βήματα. Πρώτα, το πρόγραμμα μεταγλωττίζεται σε Microsoft Intermediate Language (MSIL), που ορίζει οδηγίες για το CLR. Κατόπιν ένας άλλος μεταγλωττιστής στο CLR μεταφράζει την MSIL σε κώδικα μηχανής (για μια συγκεκριμένη πλατφόρμα), δημιουργώντας μια μόνο εφαρμογή. Οι βασικοί λόγοι για την υλοποίηση του ενδιάμεσου αυτού βήματος, που φαινομενικά στερεί σε ταχύτητα, είναι η διαλειτουργικότητα ανάμεσα στις γλώσσες και τα χαρακτηριστικά διαχείρισης της εκτέλεσης, όπως η διαχείριση της μνήμης και η ασφάλεια (Deitel κ.α., 2002).

Εάν το .NET Framework υπάρχει και είναι εγκατεστημένο για μια πλατφόρμα, αυτή η πλατφόρμα μπορεί να εκτελέσει οποιοδήποτε πρόγραμμα .NET. Η δυνατότητα ενός προγράμματος να εκτελεστεί (χωρίς τροποποιήσεις) σε διαφορετικές πλατφόρμες, είναι γνωστή ως ανεξαρτησία από την πλατφόρμα. Κώδικας γραμμένος μια φορά μπορεί να χρησιμοποιηθεί σε μία άλλη μηχανή χωρίς τροποποιήσεις, εξοικονομώντας χρόνο και χρήμα. Το .NET Framework παρέχει επίσης ένα υψηλό επίπεδο διαλειτουργικότητας γλωσσών. Προγράμματα που γράφονται σε διαφορετικές γλώσσες μεταγλωττίζονται όλα σε MSIL και έτσι είναι δυνατόν διαφορετικά μέρη να συνδυαστούν και να συνθέσουν ένα ενιαίο πρόγραμμα. Η MSIL επιτρέπει στο .NET Framework να είναι ανεξάρτητο από γλώσσες. Οποιαδήποτε γλώσσα μπορεί να μεταγλωττιστεί σε MSIL ονομάζεται γλώσσα συμβατή με το .NET. Μερικές από τις γνωστότερες γλώσσες συμβατές με το .NET

Framework εκτός από τη Visual Basic .NET είναι οι: C#, COBOL, Fortran, J#, Pascal, Perl, Jscript, Visual C++ .NET. Με το .NET Framework, προγραμματιστές διαφορετικών γλωσσών μπορούν να εργάζονται παράλληλα στην ίδια εργασία και ο κώδικας που παράγουν αφού μεταγλωττιστεί σε MSIL είναι δυνατό να συνδεθεί και να σχηματίσει ένα ενιαίο πρόγραμμα (Deitel κ.α., 2002).

2.5 Το περιβάλλον της Visual Basic .NET

Στους σύγχρονους υπολογιστές το περιβάλλον διεπαφής με το χρήστη βασίζεται στα παράθυρα (windows), τα εικονίδια (icons), τα μενού (menus) και το δείκτη (pointer). Το περιβάλλον αυτό σχεδιάστηκε για πρώτη φορά στο Palo Alto Research Center της Xerox (PARC) και υλοποιήθηκε με επιτυχία από την Apple για τους υπολογιστές Macintosh και από τη Microsoft με την οικογένεια των Windows. Η επιφάνεια εργασίας χρησιμοποιεί ως βάση τη μεταφορά του γραφείου (desktop metaphor). Ο χρήστης μετακινεί πάνω στην οθόνη παράθυρα με τη χρήση του δείκτη όπως θα κινούσε έγγραφα στο γραφείο με τα χέρια του. Βασικά τεχνολογικά στοιχεία για τη λειτουργία του περιβάλλοντος αυτού είναι η οθόνη χαρτογραφικής απεικόνισης (bitmap display) και το ποντίκι (mouse) ή κάποιος άλλος αντίστοιχος μηχανισμός που επιτρέπει στο χρήστη να δείχνει αντικείμενα στην οθόνη. Με τη χρήση εικονιδίων ορισμένα στοιχεία του περιβάλλοντος μπορούν να παρασταθούν με αμεσότητα, ενώ τα μενού κάνουν τις λειτουργίες του περιβάλλοντος προσιτές χωρίς να χρειάζεται ο χρήστης να απομνημονεύει εντολές και τη σύνταξή τους (Σπινέλλης, 2004).

Το περιβάλλον προγραμματισμού της Visual Basic .NET επιτρέπει τη γραφική σχεδίαση εφαρμογών με τη χρήση εργαλείων (controls) και τον άμεσο ορισμό κώδικα που θα αποκρίνεται σε συγκεκριμένα γεγονότα. Για το σκοπό αυτό η Visual Basic .NET απλοποιεί το περιβάλλον προγραμματισμού που ορίζουν τα Windows δίνοντας στο χρήστη μια σειρά από εξειδικευμένα εργαλεία (βασισμένα σε αυτά των Windows) καθώς και έναν τρόπο με τον οποίο ορίζει ο χρήστης την εμφάνιση του κάθε εργαλείου, τις λειτουργίες του και τον τρόπο που αυτό ανταποκρίνεται σε γεγονότα. Έτσι, κάθε εργαλείο παριστάνεται ως ένα αντικείμενο που διαθέτει (Σπινέλλης, 2004):

- Ιδιότητες (properties) όπως είναι για παράδειγμα η θέση και το χρώμα του
- Μεθόδους (methods) που μπορούν να κληθούν για εκτέλεση όπως είναι για παράδειγμα η μέθοδος της επανασχεδίασης

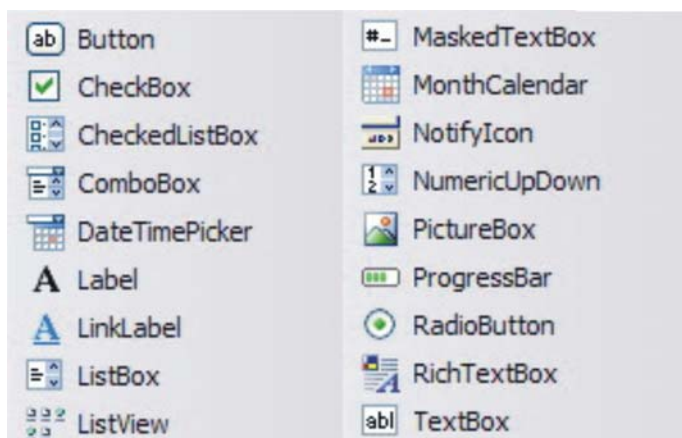
- Γεγονότα (events) στα οποία μπορεί ο χρήστης να συνδέσει κώδικα όπως είναι για παράδειγμα το γεγονός click

Ο έλεγχος που έχει ο χρήστης πάνω στα όργανα διαφέρει ανάλογα με την κατάσταση της εφαρμογής (Σπινέλλης, 2004):

- Κατά τη διάρκεια σχεδιασμού της εφαρμογής, ο χρήστης μπορεί να διαβάσει και να αλλάξει τιμή στις περισσότερες ιδιότητες καθώς και να συνδέσει κώδικα με όλα τα γεγονότα του κάθε εργαλείου
- Κατά τη διάρκεια εκτέλεσης της εφαρμογής, το πρόγραμμα μπορεί να διαβάσει και να αλλάξει τιμή στις περισσότερες ιδιότητες καθώς και να καλέσει τις μεθόδους του κάθε εργαλείου. Όταν παραλαμβάνεται από την εφαρμογή ένα γεγονός, αυτό προωθείται αυτόματα στον κώδικα που έχει συνδεθεί με το αντίστοιχο εργαλείο. Αν δεν υπάρχει τέτοιος κώδικας, τότε η εφαρμογή εκτελεί αυτόματα κώδικα ο οποίος εκτελεί τη βασική λειτουργία για το αντίστοιχο γεγονός

2.5.1 Εργαλεία

Τα εργαλεία τοποθετούνται μέσα στην εφαρμογή με γραφικό τρόπο κατά την υλοποίησή της. Κάθε εργαλείο διακρίνεται από το όνομά του και μπορεί να εφαρμόζεται απεριόριστες φορές. Αν πολλά εργαλεία ίδιου τύπου έχουν το ίδιο όνομα τότε αυτά ορίζουν έναν πίνακα εργαλείων ο οποίος μπορεί να προσπελαστεί με τη χρήση ενός δείκτη. Στον σχήμα 2.1 εμφανίζονται μερικά από τα βασικότερα εργαλεία της Visual Basic .NET.



Σχήμα 2.1: Βασικά εργαλεία της Visual Basic .NET

2.5.2 Γεγονότα

Κάθε εργαλείο έχει το δικό του σύνολο από γεγονότα τα οποία μπορεί να δεχτεί. Γεγονότα μπορούν να λαμβάνουν τόσο οι φόρμες, όσο και τα εργαλεία που περιέχουν. Ορισμένα από τα γεγονότα που αφορούν πολλά εργαλεία της Visual Basic είναι (Σπινέλλης, 2004):

- Click: Click του ποντικιού πάνω στο εργαλείο
- DblClick: Διπλό click του ποντικιού πάνω στο εργαλείο
- GotFocus: Το εργαλείο γίνεται η εστία εισόδου
- LostFocus: Το εργαλείο παύει να είναι η εστία εισόδου
- KeyDown: Ένα πλήκτρο πατιέται πάνω από το εργαλείο
- KeyUp: Ένα πλήκτρο αφήνεται πάνω από το εργαλείο
- KeyPress: Ένας χαρακτήρας γράφεται πάνω από το εργαλείο
- MouseDown: Ένα πλήκτρο του ποντικιού πατιέται πάνω από το εργαλείο
- MouseUp: Ένα πλήκτρο του ποντικιού αφήνεται πάνω στο εργαλείο.
- MouseMove: Το ποντίκι κινείται πάνω από το εργαλείο
- Change: Τα δεδομένα που περιέχει το εργαλείο άλλαξαν

2.5.3 Ιδιότητες

Οι ιδιότητες που αφορούν κάθε εργαλείο μπορούν να μεταβληθούν τόσο κατά το σχεδιασμό, όσο και κατά την εκτέλεση της εφαρμογής. Όλα τα εργαλεία έχουν προεπιλεγμένες τιμές σε όλες τις ιδιότητες τους. Αμέσως μόλις τοποθετηθεί κάποιος μηχανισμός στο παράθυρο μιας εφαρμογής η Visual Basic .NET χρησιμοποιεί ένα σύνολο προεπιλεγμένων τιμών για την ιδιότητά του. Επιπρόσθετα η Visual Basic .NET αναθέτει προεπιλεγμένα ονόματα (name) και λεζάντες (caption) στα εργαλεία. Ορισμένες ιδιότητες που εμφανίζονται σε πολλά εργαλεία είναι:

- Name: Το όνομα του συγκεκριμένου εργαλείου
- BackColor: Το χρώμα του φόντου
- ForeColor: Το χρώμα σχεδίασης
- Enabled: Αληθές αν το εργαλείο είναι ενεργό
- Visible: Αληθές αν το εργαλείο είναι ορατό

- Text: Το κείμενο που περιέχει το εργαλείο
- Caption: Το κείμενο που εμφανίζει το εργαλείο
- Value: Η τιμή που έχει το εργαλείο (π.χ. αληθές/ψευδές)
- Height: Το ύψος του εργαλείου
- Left: Η θέση του εργαλείου στη φόρμα από αριστερά
- Top: Η θέση του εργαλείου στη φόρμα από πάνω
- Width: Το πλάτος του εργαλείου

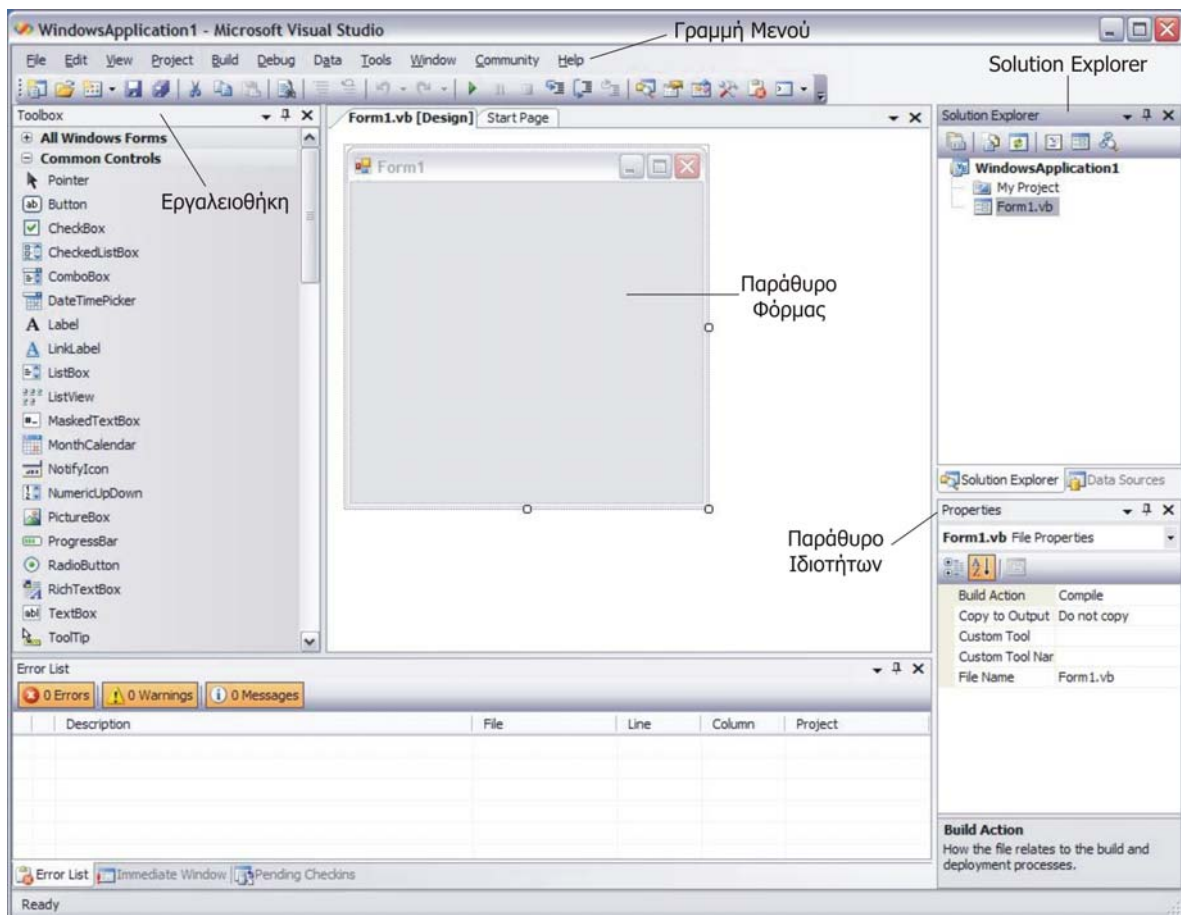
2.5.4 Ολοκληρωμένο περιβάλλον ανάπτυξης

Το ολοκληρωμένο περιβάλλον της Visual Basic .NET οργανώνει τα προγράμματα σε εργασίες (projects) και λύσεις (solutions), που περιέχουν μία ή και περισσότερες εργασίες. Ο πιο συνηθισμένος τύπος εργασίας της Visual Basic .NET είναι μια Εφαρμογή Windows, η οποία είναι ένα πρόγραμμα που εκτελείται στο λειτουργικό σύστημα Windows (πχ. Windows 2000 ή Windows XP). Στο σχήμα 2.2 εμφανίζεται το περιβάλλον ανάπτυξης της Visual Basic .NET. Τα πιο κοινά στοιχεία του είναι (Deitel κ.α., 2002):

- Γραμμή μενού
- Παράθυρο Solution Explorer
- Παράθυρο Φόρμας
- Εργαλειοθήκη
- Παράθυρο Ιδιοτήτων

Οι εντολές για τη διαχείριση του IDE και την ανάπτυξη, τη συντήρηση και την εκτέλεση προγραμμάτων περιέχονται σε μενού, που βρίσκονται στη γραμμή των μενού. Τα μενού περιέχουν ομάδες σχετικών εντολών (ονομάζονται και επιλογές μενού) που, όταν επιλέγονται, προκαλούν το IDE να εκτελέσει συγκεκριμένες ενέργειες (π.χ άνοιγμα ενός παραθύρου, αποθήκευση ενός αρχείου και εκτέλεση ενός προγράμματος) (Deitel κ.α., 2002).

Το παράθυρο Solution Explorer παρέχει πρόσβαση σε όλα τα αρχεία μίας λύσης. Όταν φορτώνεται αρχικά το IDE, το παράθυρο Solution Explorer είναι κενό καθώς δεν έχει φορτωθεί κάποια λύση και επομένως δεν υπάρχουν αρχεία για προβολή. Όταν ανοίγει κάποια λύση το παράθυρο Solution Explorer προβάλλει τα αρχεία που απαρτίζουν τη λύση αυτή. Το αρχείο που αντιστοιχεί στη φόρμα του σχήματος 2.2 είναι το Form1.vb (Deitel κ.α., 2002).



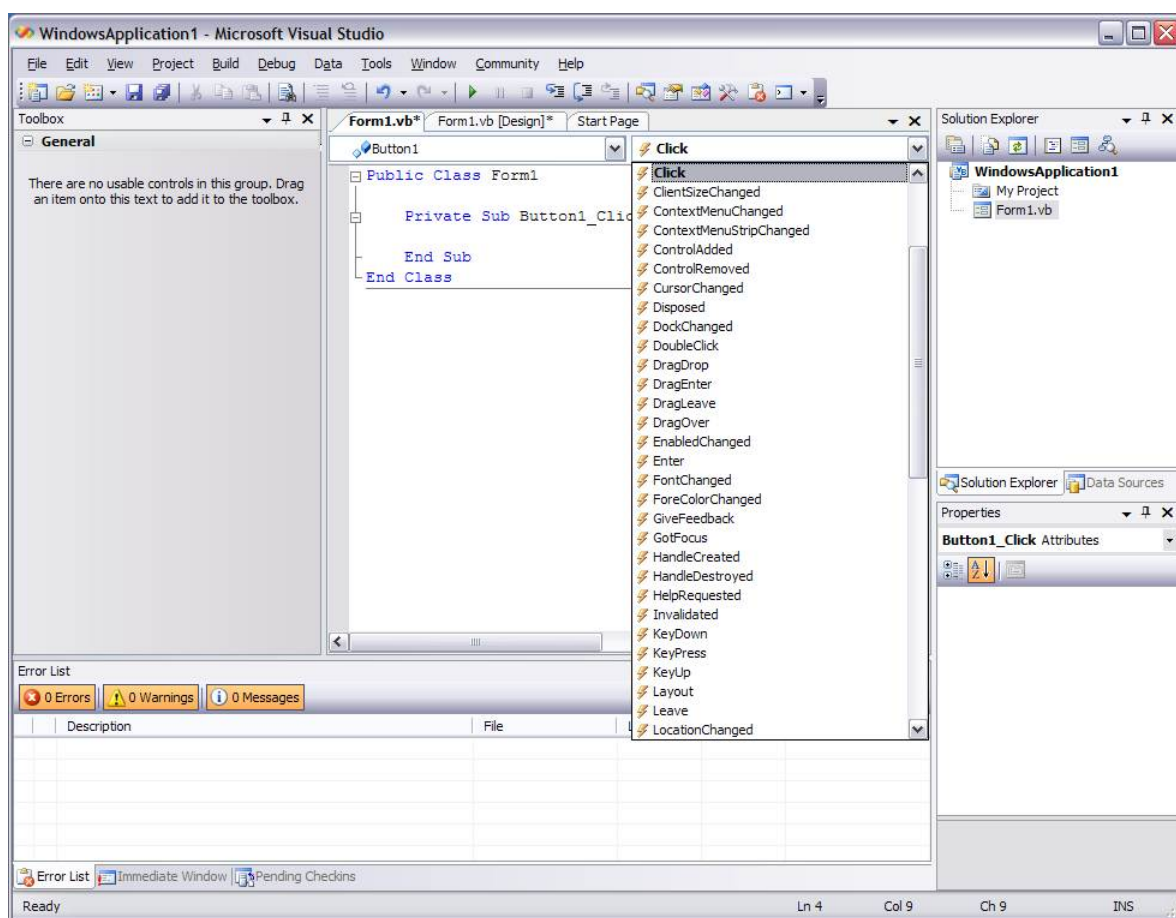
Σχήμα 2.2: Το Ολοκληρωμένο Περιβάλλον Ανάπτυξης της Visual Basic .NET

Το παράθυρο της φόρμας είναι η περιοχή που σχεδιάζονται οι φόρμες του προγράμματος. Χρησιμοποιώντας παραστατικό προγραμματισμό οι προγραμματιστές μπορούν να «σύρουν και αποθέσουν» εργαλεία, από την εργαλειοθήκη (toolbox) στο παράθυρο της φόρμας, αντί να τα δημιουργήσουν γράφοντας κώδικα. Ακριβώς όπως οι άνθρωποι δεν χρειάζεται να γνωρίζουν πώς να δημιουργήσουν έναν κινητήρα για να οδηγήσουν ένα αυτοκίνητο, έτσι και οι προγραμματιστές δεν χρειάζεται να ξέρουν πως θα δομήσουν ένα εργαλείο για να το χρησιμοποιήσουν σε κάποιο πρόγραμμα. Η μεγάλη ποικιλία εργαλείων αποτελεί ένα ισχυρό χαρακτηριστικό στο Visual Studio .NET IDE (Deitel κ.α., 2002).

Το παράθυρο ιδιοτήτων (properties window) προβάλλει τις ιδιότητες για μια φόρμα ή για ένα εργαλείο. Είναι ζωτικής σημασίας για τον παραστατικό προγραμματισμό καθώς επιτρέπει στους προγραμματιστές να τροποποιούν οπτικά τα στοιχεία ελέγχου, χωρίς να γράφουν κώδικα. Αυτή η δυνατότητα παρέχει ένα πλήθος πλεονεκτημάτων όπως κατά

πρώτον το γεγονός ότι οι προγραμματιστές μπορούν να δουν ποιες ιδιότητες είναι διαθέσιμες για τροποποίηση και σε πολλές περιπτώσεις μπορούν να μάθουν το εύρος των αποδεκτών τιμών μια ιδιότητας και κατά δεύτερον το γεγονός ότι το παράθυρο αυτό προβάλλει μια σύντομη περιγραφή της εκάστοτε επιλεγμένης ιδιότητας, βοηθώντας έτσι στη κατανόηση του σκοπού της ιδιότητας αυτής (Deitel κ.α., 2002).

Για να οριστεί ο κώδικας που εκτελείται για κάποιο γεγονός θα πρέπει ο χρήστης να κάνει διπλό κλικ σε κάποιο εργαλείο και αυτόματα δημιουργείται η δήλωση για τη συνάρτηση που αφορά το πιο χρήσιμο γεγονός για το αντίστοιχο εργαλείο (Σχήμα 2.3). Από επιλογές στην κορυφή του παραθύρου είναι δυνατόν να επιλεγθεί το εργαλείο και το γεγονός για τα οποία επιθυμεί ο χρήστης να αντιστοιχίσει κώδικα (Σπινέλλης, 2004).



Σχήμα 2.3: Δημιουργία δήλωσης συνάρτησης για το γεγονός click στο εργαλείο Button1

2.6 Δομή ενός προγράμματος σε Visual Basic .NET

Τα προγράμματα της Visual Basic αποτελούνται από πολλά κομμάτια συμπεριλαμβανομένων των λειτουργικών μονάδων (modules) και των κλάσεων (classes). Ο προγραμματιστής συνδυάζει νέες λειτουργικές μονάδες και κλάσεις με «προκατασκευασμένες» κλάσεις διαθέσιμες στη .NET Framework Class Library (FCL). Αυτές οι λειτουργικές μονάδες και κλάσεις συντίθενται από μικρότερα κομμάτια που ονομάζονται διαδικασίες (Deitel κ.α., 2002).

Η FCL παρέχει μια πλούσια συλλογή κλάσεων για την εκτέλεση συνήθων μαθηματικών υπολογισμών, τη διαχείριση ακολουθιών χαρακτήρων (strings), τις λειτουργίες εισόδου / εξόδου, τον έλεγχο σφαλμάτων και για πολλές άλλες χρήσιμες λειτουργίες. Αν και η FCL παρέχει κλάσεις που εκτελούν πολλές συνήθεις εργασίες, δεν μπορεί να παρέχει κάθε νοητό χαρακτηριστικό που θα ήθελε ένας προγραμματιστής. Έτσι η Visual Basic επιτρέπει στους προγραμματιστές τη δημιουργία των δικών τους διαδικασιών, ώστε να ικανοποιούν τις μοναδικές απαιτήσεις ενός συγκεκριμένου προβλήματος. Υπάρχουν τρεις τύποι διαδικασιών: οι διαδικασίες Sub (procedures), οι διαδικασίες Function (συναρτήσεις), και οι διαδικασίες συμβάντων (methods) (Deitel κ.α., 2002).

Συνάρτηση είναι μια συλλογή από εντολές και παραστάσεις που εκτελούν μια συγκεκριμένη λειτουργία. Όταν ολοκληρώνεται η λειτουργία μιας συνάρτησης, αυτή επιστρέφει μια τιμή στην εντολή που έκανε την κλήση της. Η μόνη πραγματική διαφορά μεταξύ της συνάρτησης και της διαδικασίας (sub) είναι ότι η συνάρτηση επιστρέφει τιμή ενώ η διαδικασία δεν επιστρέφει. Θα μπορούσαν να χρησιμοποιούνται παντού αντί για διαδικασίες, συναρτήσεις που θα επέστρεφαν την τιμή true αν ολοκληρώθηκαν με επιτυχία και επιτέλεσαν τον σκοπό τους ή την τιμή false αν παρουσιάστηκε κάποιο λάθος κατά την εκτέλεσή τους. Οι διαδικασίες συμβάντων ή μέθοδοι είναι οι διαδικασίες που περιέχονται σε κλάσεις.

Υπάρχουν διάφορα κίνητρα για τη διαίρεση του κώδικα σε διαδικασίες. Κατ' αρχήν, η προσέγγιση «διαίρει και βασίλευε» καθιστά την ανάπτυξη προγραμμάτων ευκολότερα διαχειρίσιμη. Ένα άλλο κίνητρο είναι η επαναληπτική χρήση λογισμικού – η δυνατότητα χρήσης υπαρχουσών διαδικασιών ως δομικών στοιχείων νέων προγραμμάτων. Όταν εφαρμόζονται κατάλληλες συμβάσεις ονοματολογίας και ορισμού, τα προγράμματα μπορούν να δημιουργηθούν από τυποποιημένα κομμάτια που εκτελούν συγκεκριμένες

εργασίες, ελαχιστοποιώντας την ανάγκη δημιουργίας προσαρμοσμένου κώδικα. Ένα τρίτο κίνητρο αφορά την αποφυγή επαναλήψεων κώδικα σ' ένα πρόγραμμα. Όταν ο κώδικας διαμορφώνεται ως διαδικασία, μπορεί να εκτελεστεί από πολλά σημεία σ' ένα πρόγραμμα καλώντας ή απλώς εμπλέκοντας τη διαδικασία (Deitel κ.α., 2002).

2.7 Τύποι δεδομένων, αριθμητικοί τελεστές και τελεστές σύγκρισης

Τα προγράμματα που γράφονται με τη Visual Basic απαιτούν τη χρήση και την επεξεργασία αρκετών τύπων δεδομένων (μεταβλητών), όπως για παράδειγμα αριθμοί μικρού μήκους. Η Visual Basic, υποστηρίζει πολλούς τύπους δεδομένων (data types) κάνοντας έτσι εφικτή την αντιμετώπιση οποιασδήποτε ανάγκης προγραμματισμού. Η Visual Basic, είναι «ιδιότροπη» σε θέματα τύπου δεδομένων. Συγκεκριμένα απαιτεί από το χρήστη πριν εργαστεί με μια τιμή δεδομένων, να δηλώσει επακριβώς τον τύπο της. Στον πίνακα 2.1 φαίνονται οι βασικότεροι τύποι δεδομένων της Visual Basic, η περιοχή τιμών τους και το μέγεθος αποθήκευσής τους (Vick, 2004).

Δύο βασικές παράμετροι που χαρακτηρίζουν μια μεταβλητή είναι η εμβέλεια της (scope) και ο χρόνος ζωής της (lifetime). Εμβέλεια ονομάζεται ο χώρος μέσα στον οποίο μία μεταβλητή μπορεί να χρησιμοποιηθεί. Μια μεταβλητή που ορίζεται μέσα σε μία συνάρτηση ή διαδικασία είναι τοπική για τη διαδικασία αυτή. Η εμβέλεια της είναι η διαδικασία, διότι έξω από αυτή δεν είναι δυνατόν να ανακτηθεί η τιμή της. Χρόνος ζωής είναι ο χρόνος κατά τον οποίο μια μεταβλητή ορίζεται και έχει τιμή. Μια μεταβλητή που ορίζεται μέσα σε μία συνάρτηση ή διαδικασία και είναι τοπική, έχει χρόνο ζωής όσο εκτελείται η ρουτίνα αυτή, με τον τερματισμό της οποίας παύει και η μεταβλητή να υπάρχει (Vick, 2004).

Τα περισσότερα προγράμματα εκτελούν αριθμητικούς υπολογισμούς. Οι αριθμητικοί τελεστές καθώς και η αντίστοιχη έκφρασή τους, τόσο αλγεβρικά όσο και προγραμματιστικά (σε Visual Basic .NET) εμφανίζονται στο πίνακα 2.2. Η πλειονότητα των αριθμητικών τελεστών είναι δυαδικοί τελεστές, διότι καθένας λειτουργεί χρησιμοποιώντας δύο τελεστέους. Η Visual Basic .NET παρέχει επίσης μοναδιαίους τελεστές. Για παράδειγμα οι μοναδιαίες εκδόσεις των συν (+) και μείον (-) παρέχονται έτσι ώστε οι προγραμματιστές να μπορούν να γράψουν εκφράσεις όπως +9 ή -19 (Deitel κ.α., 2002).

Πίνακας 2.1: Βασικοί τύποι δεδομένων της Visual Basic .NET (Vick, 2004)

Ονομασία	Περιοχή τιμών	Μέγεθος σε bytes
Boolean	True - False	2 bytes
Byte	0 - 255	1 byte
Currency	-922.337.203.685.477,5808 – 922.337.203.685.477,5808	8 bytes
Date	1 January 100 – 31 December 9999	8 bytes
Double	-1,79769313486232E+308 – -4,94065645841247E-324 (αρνητικές) 1,79769313486232E+308 – 4,94065645841247E-324 (θετικές)	8 bytes
Integer	-32768 – 32767	2 bytes
Long	-2.147.483.648 – 2.147.483.647	4 bytes
Object	Οποιοδήποτε αντικείμενο	4 bytes
Single	-3,402823E+38 - -1,401298E-45 1,401298E-45 - 3,402823E+38	4 bytes
String	0 – 2 εκατομμύρια χαρακτήρες	10+μέγεθος string
Variant (character)	0 – 2 εκατομμύρια χαρακτήρες	22 bytes + μέγεθος string
Variant (number)	όπως το double	16 bytes

Πίνακας 2.2: Συνήθεις αριθμητικοί τελεστές της Visual Basic .NET (Deitel κ.α., 2002)

Λειτουργία Visual Basic	Αριθμητικός Τελεστής	Αλγεβρική Έκφραση	Έκφραση της Visual Basic
Πρόσθεση	+	$f+7$	$f + 7$
Αφαίρεση	-	$p-c$	$p - c$
Πολλαπλασιασμός	*	bm	$b * m$
Διαίρεση	/	x/y	x / y
Διαίρεση ακεραίων	\	καμία	$v \setminus u$
Υπόλοιπο διαίρεσης	Mod	$r \text{ modulo } s$	$r \text{ Mod } s$
Υψωση σε δύναμη	^	q^p	$q \wedge p$
Μοναδιαία άρνηση	-	-e	-e
Μοναδιαία κατάφαση	+	+g	+g

Η Visual Basic εφαρμόζει τους τελεστές στις αριθμητικές εκφράσεις με σειρά ακρίβειας, που καθορίζεται από τους ακόλουθους κανόνες προτεραιότητας τελεστών, που γενικώς είναι ίδιοι με εκείνους που ακολουθούνται από την άλγεβρα (Deitel κ.α., 2002):

- Οι τελεστές σε εκφράσεις που περιέχονται σε παρενθέσεις υπολογίζονται πρώτοι. Έτσι οι παρενθέσεις χρησιμοποιούνται για να αναγκάσουν τη σειρά των υπολογισμών να εκτελεστεί κατά τη βούληση του προγραμματιστή. Με ένθετες (εσωτερικές) παρενθέσεις, οι τελεστές που περιέχονται στο εσωτερικό ζεύγος παρενθέσεων εφαρμόζονται πρώτοι.
- Η ύψωση σε δύναμη ακολουθεί. Εάν μια έκφραση περιέχει πολλές πράξεις ύψωσης σε δυνάμεις, οι τελεστές εφαρμόζονται από δεξιά προς τα αριστερά.
- Οι μοναδιαίοι τελεστές + και – εφαρμόζονται στη συνέχεια. Εάν μια έκφραση περιέχει πολλούς τελεστές πρόσθετων, οι τελεστές εφαρμόζονται από δεξιά προς τα αριστερά.
- Οι πράξεις πολλαπλασιασμού και διαίρεσης πραγματικών αριθμών εφαρμόζονται στη συνέχεια.
- Η ακέραια διαίρεση ακολουθεί στη διαδοχή της εφαρμογής των τελεστών.
- Οι πράξεις υπολοίπου διαίρεσης εφαρμόζονται στη συνέχεια.
- Τελευταίες εφαρμόζονται οι πράξεις πρόσθεσης και αφαίρεσης.

Στον πίνακα 2.3 εμφανίζονται οι τελεστές σύγκρισης της Visual Basic.NET που χρησιμοποιούνται από τους προγραμματιστές για να δημιουργούν συνθήκες ελέγχου, οι οποίες περιέχονται σε ευρύτερες δομές ελέγχου ή επανάληψης και για τις οποίες γίνεται σύντομη αναφορά στην επόμενη παράγραφο.

Πίνακας 2.3: Τελεστές σύγκρισης στη Visual Basic.NET (Deitel κ.α.,2002)

Αλγεβρικός Τελεστής	Visual Basic Τελεστής	Παράδειγμα συνθήκης	Σημασία συνθήκης
=	=	$x=y$	x ισούται με y
≠	<>	$x<>y$	x δεν ισούται με y
>	>	$x>y$	x μεγαλύτερο του y
<	<	$x<y$	x μικρότερο y
≥	>=	$x>=y$	x μεγαλύτερο ή ίσο του y
≤	<=	$x<=y$	x μικρότερο ή ίσο του y

2.8 Δομές ελέγχου

Κανονικά οι προτάσεις σε ένα πρόγραμμα εκτελούνται η μία μετά την άλλη, με τη σειρά που γράφονται, πράγμα που καλείται σειριακή εκτέλεση. Ωστόσο διάφορες προτάσεις στη Visual Basic δίνουν τη δυνατότητα στον προγραμματιστή να καθορίζει τη σειρά εκτέλεσης του προγράμματος. Μια μεταβίβαση ελέγχου λαμβάνει χώρα όταν η εκτελούμενη πρόταση δεν ακολουθεί άμεσα την προηγούμενη εκτελούμενη πρόταση στο πρόγραμμα (Deitel κ.α., 2002).

Κατά τη δεκαετία του 1960 έγινε σαφές ότι η ανεξέλεγκτη χρήση των μεταβιβάσεων ελέγχου προκαλούσε δυσκολίες στις ομάδες ανάπτυξης λογισμικού. Το πρόβλημα ήταν η πρόταση GoTo, που επιτρέπει στον προγραμματιστή να καθορίζει μεταβίβαση ελέγχου σε ένα ευρύ φάσμα πιθανών προορισμών σε ένα πρόγραμμα. Από τότε, η σημειολογία του δομημένου προγραμματισμού έγινε σχεδόν συνώνυμη με τον περιορισμό των GoTo. Από έρευνες προέκυψε ότι όλα τα προγράμματα θα μπορούσαν να γραφτούν μόνο με τρεις δομές ελέγχου: τη σειριακή δομή, τη δομή επιλογής και τη δομή επανάληψης (Deitel κ.α., 2002).

Η σειριακή δομή είναι ενσωματωμένη στη Visual Basic. Ο υπολογιστής εκτελεί σειριακά της προτάσεις της Visual Basic, εκτός και αν πάρει διαφορετική οδηγία. Επιπρόσθετα η Visual Basic διαθέτει τρεις τύπους δομών επιλογής (Deitel κ.α., 2002):

- Η δομή επιλογής If/Then εκτελεί (επιλέγει) μια ενέργεια (ή σειρά ενεργειών) εάν μια συνθήκη είναι αληθής ή παρακάμπτει την ενέργεια (ή τη σειρά ενεργειών) αν η συνθήκη είναι ψευδής
- Η δομή επιλογής If/Then/Else εκτελεί μια ενέργεια εάν μια συνθήκη είναι αληθής και εκτελεί μια διαφορετική ενέργεια αν η συνθήκη είναι ψευδής
- Η δομή Select Case εκτελεί μία από πολλές ενέργειες ανάλογα με την τιμή μιας έκφρασης

Μια δομή επανάληψης επιτρέπει στον προγραμματιστή να καθορίζει ότι θα πρέπει να επαναλαμβάνει μια ενέργεια, ανάλογα με την τιμή κάποιας συνθήκης. Οι βασικότερες δομές επανάληψης της Visual Basic είναι οι (Deitel κ.α., 2002):

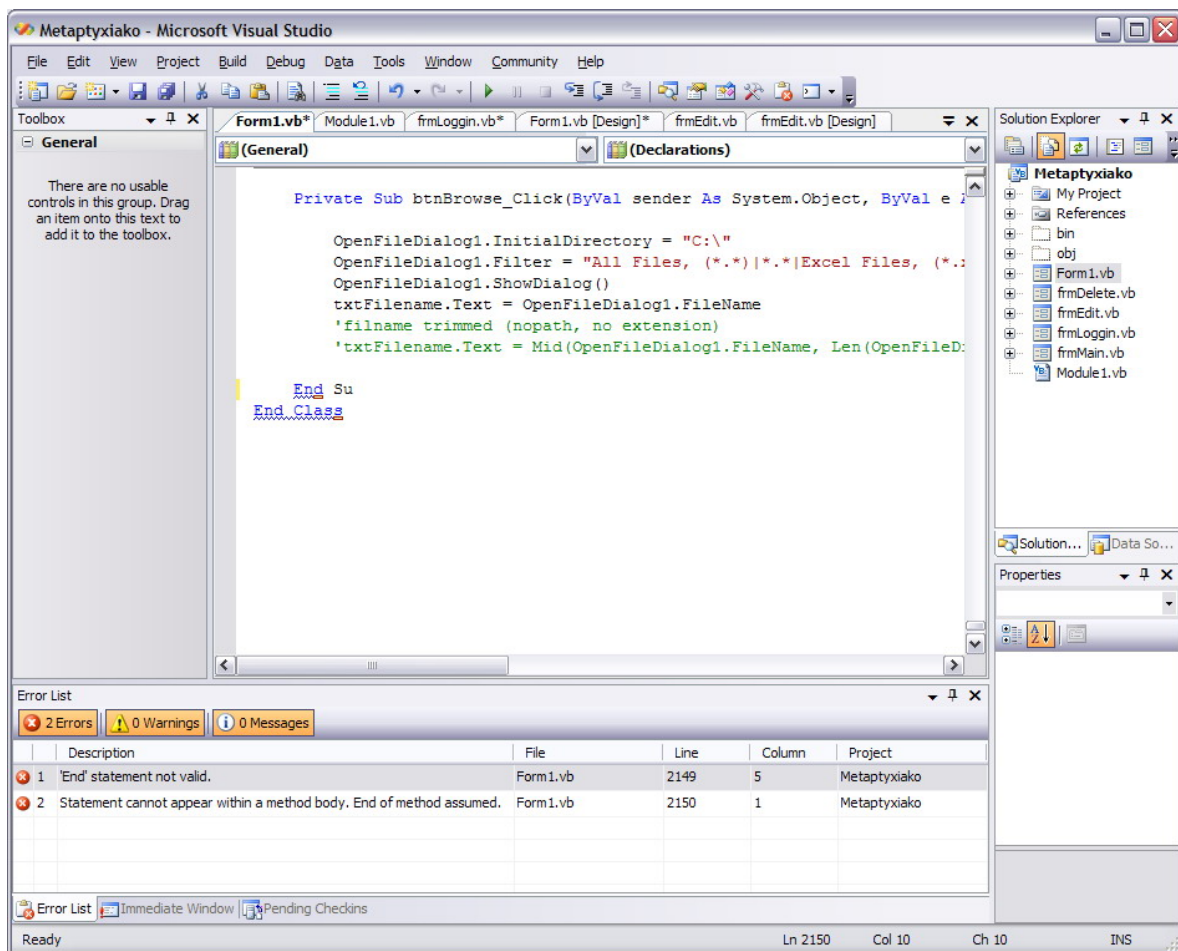
- While, Do While/Loop και Do/Loop While
- Do Until/Loop και Do/Loop Until
- For/Next και For Each/Next

2.9 Ανίχνευση και διόρθωση σφαλμάτων στη Visual Basic .NET

Συντακτικά σφάλματα (ή σφάλματα μεταγλώττισης) λαμβάνουν χώρα όταν οι προτάσεις του κώδικα του προγράμματος παραβιάζουν τους γραμματικούς κανόνες της γλώσσας προγραμματισμού. Τα συντακτικά σφάλματα εντοπίζονται από τον μεταγλωττιστή. Στο Visual Studio .NET, τα συντακτικά σφάλματα εμφανίζονται στο παράθυρο Error List μαζί με μια περιγραφή, τον αριθμό γραμμής και στήλης καθώς και τον το όνομα του αρχείου όπως φαίνεται στο σχήμα 2.4. Προγράμματα που περιέχουν συντακτικά σφάλματα, όπως γίνεται εύκολα αντιληπτό, δεν μπορούν να εκτελεστούν. (Deitel κ.α., 2002).

Ανίχνευση και διόρθωση σφαλμάτων (debugging) είναι η διαδικασία εύρεσης και διόρθωσης λογικών σφαλμάτων σε εφαρμογές. Τα λογικά σφάλματα είναι πιο περίπλοκα από τα συντακτικά σφάλματα επειδή το πρόγραμμα μεταγλωττίζεται επιτυχώς, αλλά δεν εκτελείται όπως αναμενόταν. Τα λογικά σφάλματα είναι συχνά δύσκολο να εντοπιστούν επειδή ο προγραμματιστής δε μπορεί να δει τον κώδικα καθώς εκτελείται. Μερικοί προγραμματιστές προσπαθούν να διορθώσουν προγράμματα χρησιμοποιώντας πλαίσια μηνυμάτων. Αυτή η μέθοδος είναι «δυσκίνητη», επειδή οι προγραμματιστές πρέπει να γράψουν μια γραμμή κώδικα κάθε φορά που υποπτεύονται ότι μπορεί να υπάρχει ένα πρόβλημα. Όταν το πρόγραμμα έχει διορθωθεί, ο προγραμματιστής πρέπει να αφαιρέσει αυτές τις προτάσεις προβολής (Deitel κ.α., 2002).

Οι debuggers είναι σύνολα εργαλείων που επιτρέπουν στον προγραμματιστή να αναλύει ένα πρόγραμμα, καθώς εκτελείται. Αυτά τα εργαλεία επιτρέπουν στον προγραμματιστή να σταματά την εκτέλεση του προγράμματος, να εξετάζει και να θέτει τις μεταβλητές καθώς και να καλεί διαδικασίες χωρίς να χρειάζεται να τροποποιήσει το πρόγραμμα. Ένα από τα ισχυρότερα εργαλεία ανίχνευσης λαθών είναι τα σημεία διακοπής (breakpoints), που είναι σημάδια που μπορούν να τεθούν σε οποιαδήποτε γραμμή εκτέλεσιμου κώδικα. Όταν ένα πρόγραμμα φτάσει σε ένα σημείο διακοπής, παύει η εκτέλεσή του επιτρέποντας στον προγραμματιστή να εξετάσει την κατάσταση του προγράμματος και να εξασφαλίσει τη σωστή λειτουργία του (Deitel κ.α., 2002).



Σχήμα 2.4: Παράδειγμα συντακτικού λάθους και παράθυρο Error List στη Visual Basic.NET

2.10 Γιατί επιλέχτηκε η Visual Basic .NET

Όπως γίνεται αντιληπτό από την παραπάνω σύντομη παρουσίαση της, η Visual Basic .NET είναι μία γλώσσα προγραμματισμού αρκετά απλή με τεράστιες δυνατότητες, που επιτρέπουν στον προγραμματιστή να σχεδιάσει και να προγραμματίσει οποιαδήποτε λειτουργία επιθυμεί χωρίς να απαιτούνται ιδιαίτερα εξειδικευμένες γνώσεις. Ένα από τα σημαντικότερα πλεονεκτήματα της Visual Basic .NET είναι η ευκολία με την οποία μπορεί κάποιος προγραμματιστής να σχεδιάσει ένα περιβάλλον διεπαφής (interface) ιδιαίτερα απλό και εύχρηστο για τον απλό χρήστη ηλεκτρονικού υπολογιστή. Είναι ιδιαίτερα εντυπωσιακό το γεγονός ότι ο προγραμματιστής απλώς ενσωματώνει τα διάφορα εργαλεία στην εφαρμογή του χωρίς να ασχολείται με το πώς δουλεύουν αυτά εσωτερικά ή με τον τρόπο που αυτά συνδέονται με το λειτουργικό σύστημα.

Εκτός από τις ευκολίες που παρέχει στα γραφικά, η Visual Basic περιλαμβάνει μια πολύ μεγάλη συλλογή μαθηματικών συναρτήσεων και τελεστών, στοιχεία απαραίτητα για την πραγματοποίηση αλγεβρικών πράξεων καθώς και ένα πλήθος εξωτερικών εργαλείων για την διενέργεια πολύπλοκων διαδικασιών όπως είναι η επικοινωνία με βάσεις δεδομένων ή ο σχεδιασμός διαγραμμάτων. Τέλος αξίζει να αναφερθεί ότι η Visual Basic .NET διαθέτει μία από τις πληρέστερες βιβλιοθήκες (MSDN Libraries) με θέματα βοήθειας καθώς και παραδείγματα, γεγονός που καθιστά ακόμη πιο εύκολη την εκμάθηση της ακόμη και από αρχάριους χρήστες.

ΚΕΦΑΛΑΙΟ 3

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΔΟΚΙΜΕΣ ΜΗΧΑΝΙΚΗΣ ΠΕΤΡΩΜΑΤΩΝ

3.1 Εισαγωγή

Το κεφάλαιο αυτό αναφέρεται στις φυσικές και μηχανικές ιδιότητες των πετρωμάτων και στη δυνατότητα μέτρησης τους στο εργαστήριο. Επιπρόσθετα περιγράφει το σκοπό των διαφόρων εργαστηριακών δοκιμών, τον εξοπλισμό που απαιτείται καθώς και την απαιτούμενη διαδικασία που ακολουθείται για τη διεξαγωγή των πειραμάτων αυτών.

3.2 Ιδιότητες των πετρωμάτων

Οι ιδιότητες των πετρωμάτων είναι δυνατόν να χωριστούν σε δύο μεγάλες κατηγορίες:

- στις φυσικές ιδιότητες, και
- στις μηχανικές ιδιότητες

Οι φυσικές ιδιότητες περιλαμβάνουν ιδιότητες που περιγράφουν ποσοτικά ή και ποιοτικά τα φυσικά χαρακτηριστικά των πετρωμάτων, ενώ οι μηχανικές ιδιότητες περιγράφουν τις αντοχή των υλικών αυτών σε διάφορες μορφές καταπονήσεων. Στις μηχανικές ιδιότητες περιλαμβάνονται και οι διάφορες σταθερές που επηρεάζουν τη μηχανική αντοχή των υλικών, όπως για παράδειγμα οι σταθερές ελαστικότητας. Στον πίνακα 3.1 δίνεται ενδεικτικός κατάλογος των ιδιοτήτων αυτών. Εκτός των μηχανικών και φυσικών ιδιοτήτων που χαρακτηρίζουν ένα πέτρωμα, διακρίνονται και άλλου τύπου ιδιότητες, όπως ηλεκτρικές (π.χ. ηλεκτρική αγωγιμότητα), θερμικές (π.χ. αγωγιμότητα θερμοχωρητικότητα) και χημικές (Αγιουτάντης, 2002).

3.3 Φυσικές παράμετροι των πετρωμάτων

Δύο από τις σπουδαιότερες φυσικές παραμέτρους που χαρακτηρίζουν ένα πέτρωμα είναι η πυκνότητα (density) και το ειδικό βάρος (specific gravity). Η πυκνότητα, όπως είναι γνωστό, ορίζεται από το πηλίκο της μάζας (m) προς τον όγκο (V) που αυτή καταλαμβάνει.

Πίνακας 3.1 Ενδεικτικός κατάλογος φυσικών και μηχανικών ιδιοτήτων πετρωμάτων (Αγιουτάντης, 2002)

Φυσικές Ιδιότητες	Μηχανικές Ιδιότητες
Ορυκτολογική σύνθεση και δομή	Συνοχή
Ειδική πυκνότητα	Μέτρο ελαστικότητας
Μοναδιαίο βάρος	Λόγος Poisson
Πορώδες	Αρχική εντατική κατάσταση
Λόγος κενών	Αντοχές
Φυσική υγρασία	Ερπυσμός
Υγρασία κορεσμού	Διόγκωση
Υδατοπερατότητα	Γωνία εσωτερικής τριβής
Άλλες ιδιότητες	Άλλες ιδιότητες

$$\rho = \frac{m}{V} \left(\frac{g}{cm^3} \right) \quad [3.1]$$

Ως ειδικό βάρος ορίζεται ο λόγος του βάρους μοναδιαίου όγκου του υλικού προς το βάρος νερού ίδιου όγκου και επομένως είναι αδιάστατο μέγεθος. Σε πολλές περιπτώσεις χρησιμοποιείται το φαινόμενο ειδικό βάρος (bulk or apparent specific gravity), το οποίο είναι ο λόγος του βάρους του υλικού (μαζί με τους πόρους, την υγρασία, κλπ) ως προς όγκο νερού, ίσο με τον συνολικό όγκο του υλικού. Το φαινόμενο ειδικό βάρος είναι επίσης ένα αδιάστατο μέγεθος (Αγιουτάντης, 2002).

Δύο εξίσου σημαντικές φυσικές παράμετροι των πετρωμάτων είναι ο λόγος κενών ή δείκτης πόρων και το πορώδες. Ο λόγος κενών (e) ορίζεται ως ο λόγος του όγκου των κενών (V_v) προς τον όγκο των στερεών συστατικών του πετρώματος (V_s). Το πορώδες ενός πετρώματος ορίζεται ως ο λόγος του όγκου των κενών (V_v), προς τον ολικό όγκο (V) του πετρώματος (Στειακάκης, 2004). Στον πίνακα 3.2 παρουσιάζονται ενδεικτικές τιμές φαινόμενου ειδικού βάρους και πορώδους διαφόρων πετρωμάτων.

$$e = \frac{V_v}{V_s} \quad [3.2]$$

$$e = \frac{V_v}{V} (x100\%) \quad [3.3]$$

Πίνακας 2.2: Ενδεικτικά φαινόμενα ειδικά βάρη και πορώδη πετρωμάτων (Αγιουτάντης, 2002)

Τύπος Πετρώματος	Φαινόμενο ειδικό βάρος	Πορώδες (%)
Βασάλτης	2,21-2,77	0,22-22,06
Διαβάσης	2,82-2,95	0,17-1,00
Γάββρος	2,72-3,00	0,00-3,57
Γρανίτης	2,53-2,62	1,02-2,87
Δολομίτης	2,67-2,72	0,27-4,10
Ασβεστόλιθος	2,67-2,72	0,27-4,10
Ψαμμίτης	1,91-2,58	1,62-26,4
Σχιστοπηλός (shale)	2,00-2,40	20,00-50,00
Γνεύσιος	2,61-3,12	0,32-1,16
Μάρμαρο	2,51-2,86	0,65-0,81
Χαλαζίτης	2,61-2,67	0,40-0,65
Σχιστόλιθος	2,60-2,85	10,00-30,00
Αργιλικός σχιστόλιθος	2,71-2,78	1,84-3,61

3.4 Μηχανικές παράμετροι των πετρωμάτων

3.4.1 Παράμετροι αντοχής

Η σημαντικότερη κατηγορία μηχανικών παραμέτρων που χαρακτηρίζουν τα πετρώματα αλλά και τα φυσικά υλικά αποτελείται από αυτές που αναφέρονται στην αντοχή τους. Η αντοχή (strength) ενός υλικού είναι η ικανότητά του να ανθίσταται σε εξωτερικές δυνάμεις (τάσεις) και εκφράζεται σε μονάδες τάσης. Η ικανότητα αυτή των υλικών εξαρτάται άμεσα από τον τύπο της καταπόνησης. Στη Μηχανική των στερεών σωμάτων διακρίνονται οι ακόλουθοι τρόποι καταπόνησης που μπορεί να δεχθεί ένα υλικό (Αγιουτάντης, 2002):

- Καταπόνηση σε θλίψη
- Καταπόνηση σε εφελκυσμό
- Καταπόνηση σε διάτμηση
- Καταπόνηση σε κάμψη
- Καταπόνηση σε στρέψη

Για τη μελέτη της εντατικής κατάστασης των πετρωμάτων τα οποία εξετάζονται είτε στη φυσική τους θέση είτε στο εργαστήριο, ενδιαφέρουν κυρίως οι τέσσερις πρώτοι τύποι καταπόνησης. Τα πετρώματα χαρακτηρίζονται από τιμές αντοχής ανάλογα με τον τύπο των εξωτερικών δυνάμεων (ή τάσεων) που υφίστανται. Διακρίνεται η αντοχή σε θλίψη (compressive strength) με ή χωρίς την παρουσία πλευρικών τάσεων, η αντοχή σε διάτμηση (shear strength), η αντοχή σε εφελκυσμό (tensile strength) και η αντοχή σε κάμψη (bending strength). Οι τιμές αντοχής του πετρώματος είναι δυνατόν να μετρηθούν στο εργαστήριο (σε κατάλληλα διαμορφωμένα δοκίμια) ή επιτόπου. Και στις δύο περιπτώσεις διακρίνονται μετρήσεις αντοχής του αέριου πετρώματος και μελέτες αντοχής της βραχομάζας, δηλαδή του πετρώματος με τυχόν ασυνέχειες που το χαρακτηρίζουν (Αγιουτάντης, 2002).

3.4.2 Εσωτερική τριβή και συνοχή

Η συνοχή (cohesion) είναι μια πολύ σημαντική ιδιότητα των υλικών, η οποία συμβάλλει στην αύξηση της διατμητικής αντοχής ενός υλικού και ουσιαστικά εκφράζει την ικανότητα ενός υλικού να ανθίσταται σε διατμητικές δυνάμεις. Η συνοχή σε ένα εδαφικό υλικό εξαρτάται από την περιεκτικότητα σε νερό, την πυκνότητα και την πλαστικότητα του εδάφους. Σ' ένα πέτρωμα, η συνοχή εξαρτάται επίσης από την υγρασία του πετρώματος και την πυκνότητά του (Αγιουτάντης, 2002).

Στα εδάφη η γωνία εσωτερικής τριβής (angle of internal friction) εκφράζει ουσιαστικά ένα μέτρο της γωνίας φυσικής απόθεσης ενός εδάφους. Για παράδειγμα στην περίπτωση ξηρής άμμου, η γωνία αυτή είναι περίπου 30° , ενώ μέτριας ή μεγάλης πλαστικότητας εδαφικά υλικά εμφανίζουν γωνία εσωτερικής τριβής της τάξης των 15° (Αγιουτάντης, 2002).

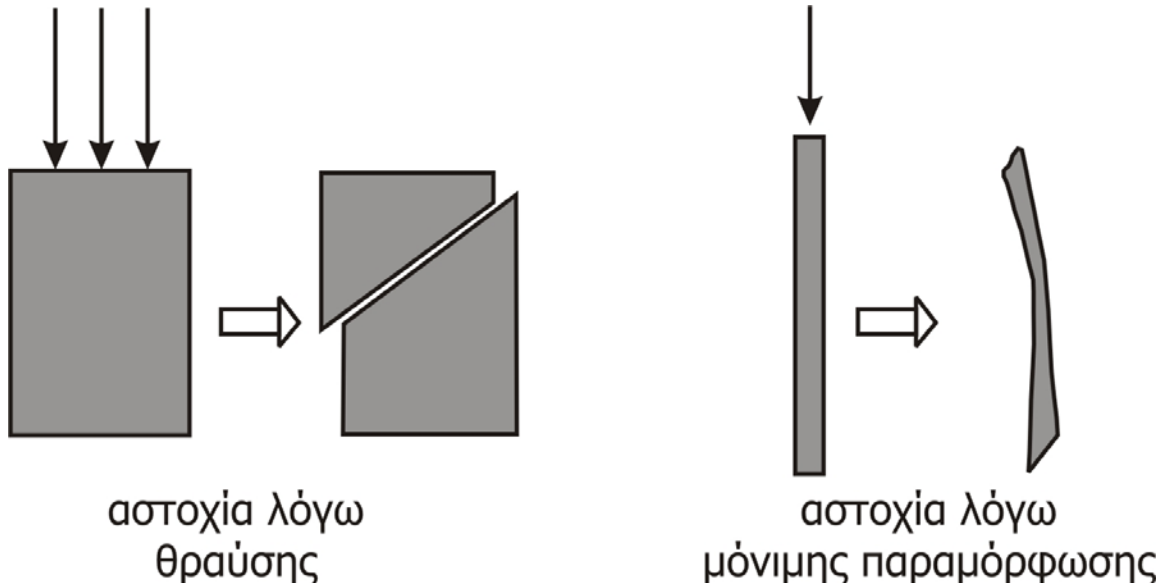
Στα πετρώματα, η έννοια της συνοχής εκφράζεται έμμεσα και όταν η συμπεριφορά ενός υλικού σε διάτμηση προσομοιάζεται με το νόμο Coulomb. Στη πράξη, τις έννοιες της συνοχής και της γωνίας εσωτερικής τριβής, εκφράζει η τριβή μεταξύ των παρειών μιας ασυνέχειας, όταν το υλικό καταπονείται σε διάτμηση, καθώς και μεταξύ των κόκκων του υλικού, όταν το υλικό υφίσταται διατμητική αστοχία (Αγιουτάντης, 2002).

3.4.3 Παράμετροι ελαστικότητας

Οι πλέον γνωστές παράμετροι που χρησιμοποιούνται για να εκφράσουν την ελαστική συμπεριφορά ενός σώματος είναι το μέτρο ελαστικότητας του Young καθώς και ο λόγος του Poisson. Η πρώτη εκφράζει τη σκληρότητα ή την ακαμψία ενός υλικού, παρόμοια με την ακαμψία ενός ελατηρίου, όταν αυτό συμπιέζεται ή εκτείνεται, ενώ η δεύτερη εκφράζει την πλευρική παραμόρφωση ενός υλικού, λόγω καταπόνησης σε μια διεύθυνση (Αγιουτάντης, 2002).

3.5 Αστοχία ενός υλικού

Με τον όρο αστοχία (failure) ενός υλικού μετά από την υπέρβαση της αντοχής του σε κάποιου τύπου καταπόνηση, εννοείται η μεταβολή της μηχανικής συμπεριφοράς του (και ενδεχομένως και της δομής του) κατά τέτοιο τρόπο ώστε να μη μπορεί να αναλάβει τα φορτία για τα οποία σχεδιάστηκε ή επιλέχτηκε. Επομένως, η αστοχία ενός υλικού μπορεί να μη συνεπάγεται και τη θραύση του, αλλά απλά την πλαστική ή μόνιμη παραμόρφωσή του σε μη επιτρεπτά επίπεδα όπως εμφανίζεται στο σχήμα 3.1 (Αγιουτάντης, 2002).



Σχήμα 3.1: Αστοχία ενός υλικού (Αγιουτάντης, 2002)

3.6 Επίδραση ασυνεχειών στη μηχανική συμπεριφορά του πετρώματος

Γενικά, έχει αποδειχθεί πειραματικά ότι επέρχεται μείωση της αντοχής του πετρώματος ή της βραχομάζας, λόγω της παρουσίας ασυνεχειών, που ερμηνεύεται με τις ακόλουθες παρατηρήσεις (Αγιουτάντης, 2002):

- Μια ασυνέχεια δεν μπορεί να μεταφέρει εφελκυστικές τάσεις κάθετα στην επιφάνειά της
- Η διατμητική αντοχή κατά μήκος της επιφάνειας μίας ασυνέχειας είναι συνήθως μειωμένη σε σχέση μ' αυτή του ακέραιου πετρώματος

Επίσης, η παρουσία ασυνεχειών επηρεάζει και τις ελαστικές παραμέτρους του υλικού, καθώς και τις ταχύτητες διάδοσης των ελαστικών κυμάτων μέσα από τη μάζα του υλικού. Είναι επομένως φανερό ότι δεν μπορεί να γίνει μια πλήρης μελέτη της μηχανικής συμπεριφοράς ενός πετρώματος, εάν δε γίνει παράλληλα μία εμπεριστατωμένη μελέτη των ασυνεχειών που διέπουν τη συμπεριφορά του εν λόγω υλικού. Στη βιβλιογραφία αναφέρονται περισσότεροι του ενός τρόποι προσέγγισης του προβλήματος του προσδιορισμού των μηχανικών ιδιοτήτων της βραχομάζας (Αγιουτάντης, 2002):

- Υπολογίζονται οι ιδιότητες του άρρηκτου πετρώματος με εργαστηριακές δοκιμές και στη συνέχεια τα μέτρα των εργαστηριακών αυτών μηχανικών ιδιοτήτων ανάγονται σε αντίστοιχες τιμές για τη βραχομάζα, σύμφωνα με ορισμένους (εμπειρικούς συνήθως) συντελεστές μείωσης.
- Υπολογίζονται οι ιδιότητες του πετρώματος με εργαστηριακές δοκιμές σε αντιπροσωπευτικά δείγματα μεγάλου μεγέθους, στα οποία λαμβάνεται υπόψη και ο προσανατολισμός του δείγματος και τα μέτρα των μηχανικών αυτών ιδιοτήτων λαμβάνονται ως οι τιμές για τη βραχομάζα.
- Υπολογίζονται οι ιδιότητες του πετρώματος με επιτόπου δοκιμές και τα μέτρα των μηχανικών αυτών ιδιοτήτων λαμβάνονται ως οι τιμές για τη βραχομάζα.

Ανάλογα με την εφαρμογή αλλά και την αξιοπιστία του τρόπου υπολογισμού των μηχανικών ιδιοτήτων της βραχομάζας, προσδιορίζονται οι συντελεστές ασφάλειας που θα πρέπει να χρησιμοποιηθούν σε κάθε φάση κατασκευής του έργου (Αγιουτάντης, 2002).

3.7 Δειγματοληψία και προπαρασκευή δοκιμών

Η σωστή δειγματοληψία αποτελεί ίσως το σημαντικότερο στάδιο της μελέτης των μηχανικών ιδιοτήτων των πετρωμάτων, τόσο στο εργαστήριο όσο και στο φυσικό τους χώρο. Ένα δείγμα πρέπει να είναι αντιπροσωπευτικό του πετρώματος από το οποίο προέρχεται και πρέπει απαραίτητα να αξιολογείται ανάλογα με τη θέση και τον προσανατολισμό του στο φυσικό χώρο (Αγιουτάντης, 2002).

Σε περιπτώσεις όπου πρόκειται να εξετασθεί ένας σχηματισμός που βρίσκεται εξολοκλήρου κάτω από την επιφάνεια της γης, ο συνήθης τρόπος συλλογής δειγμάτων γίνεται με την όρυξη γεώτρησης και πυρηνοληψία. Από τους πυρήνες αυτούς επιλέγονται αντιπροσωπευτικά τμήματα, τα οποία στη συνέχεια μεταφέρονται στο εργαστήριο (αφού ληφθούν τα κατάλληλα μέτρα διατήρησης της φυσικής υγρασίας κλπ.), με σκοπό την κατασκευή δοκιμών και την εκτέλεση διαφόρων δοκιμών (Αγιουτάντης, 2002).

Όταν η πρόσβαση προς τον σχηματισμό είναι δυνατή (π.χ. στη περίπτωση εξέτασης της ευστάθειας ενός πρανούς), τότε συνήθως συλλέγονται τμήματα από το πέτρωμα, τα οποία έχουν με κάποιο τρόπο αποκολληθεί από τον υπόλοιπο σχηματισμό, και αφού σημειωθεί ο προσανατολισμός τους, μεταφέρονται στο εργαστήριο για περαιτέρω επεξεργασία. Η κατασκευή ή/και προπαρασκευή ενός δείγματος στο εργαστήριο ακολουθεί συνήθως τα εξής στάδια (Αγιουτάντης, 2002).

- Διάτρηση του ακανόνιστου δείγματος και λήψη πυρήνων υλικού, σύμφωνα με τις αντίστοιχες προδιαγραφές. Γί' αυτό το σκοπό, το δείγμα στερεώνεται σε κατάλληλη βάση διάτρησης (drill press).
- Κοπή των άκρων των κυλίνδρων (πυρήνων) σε κατάλληλο δισκοπρίονο με αδαμαντοτροχό (diamond saw).
- Λείανση των βάσεων των κυλίνδρων σε κατάλληλο λειαντικό μηχάνημα (grinding machine).
- Μέτρηση των διαστάσεων των δοκιμών με τις ανοχές που προβλέπει η σχετική διαδικασία.

3.8 Προσδιορισμός της αντοχής σε μονοαξονική θλίψη

Το αντικείμενο της συγκεκριμένης μεθοδολογίας είναι ο προσδιορισμός της αντοχής σε μονοαξονική θλίψη ενός πετρώματος, χρησιμοποιώντας δοκίμια κανονικής γεωμετρίας.

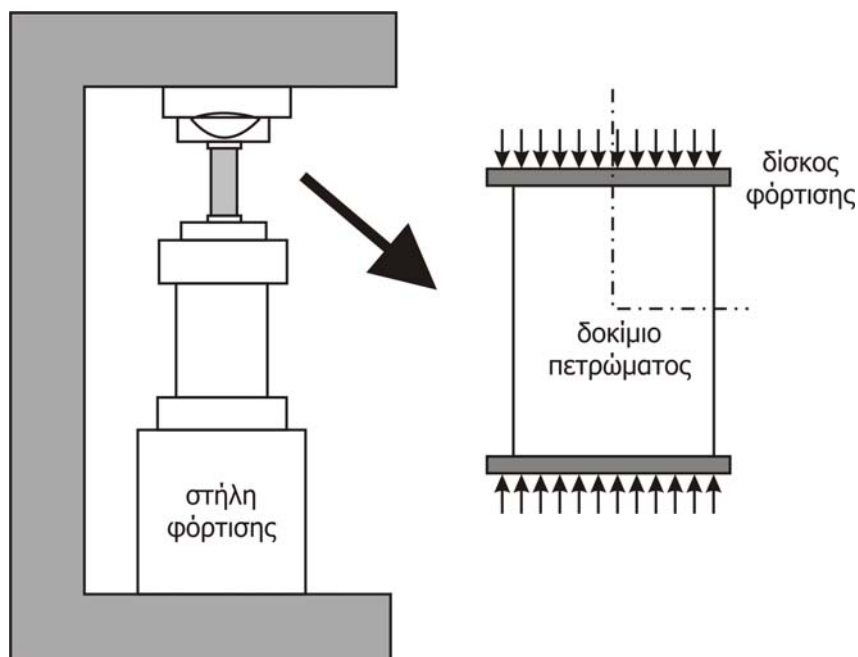
Τα αποτελέσματα των δοκιμών μπορούν να χρησιμοποιηθούν για την ταξινόμηση και το χαρακτηρισμό του ακέραιου πετρώματος από πλευράς αντοχής. Στον εξοπλισμό περιλαμβάνονται τα ακόλουθα όργανα και συσκευές (Αγιουτάντης, 2002):

- Συσκευή δοκιμής: Η συσκευή θα πρέπει να μπορεί να επιβάλλει σταθερό ρυθμό φόρτισης στο δοκίμιο και συγχρόνως να έχει τη δυνατότητα να μετρήσει ή/και να καταγράψει το επιβαλλόμενο φορτίο.
- Πλάκες φόρτισης: Οι πλάκες φόρτισης χρησιμεύουν στο να μεταφέρουν το φορτίο στο δοκίμιο και θα πρέπει να είναι παράλληλες μεταξύ τους.
- Χαλύβδινοι δίσκοι: Οι δίσκοι (platens) τοποθετούνται ανάμεσα στις πλάκες φόρτισης και στο δοκίμιο και αποσκοπούν στο να ελαττώσουν την υπερβολική πλευρική παραμόρφωση του δοκιμίου λόγω δυνάμεων τριβής στα σημεία επαφής.
- Σφαιρική κεφαλή έδρασης: Η σφαιρική κεφαλή έδρασης πρέπει να βρίσκεται στο πάνω μέρος του δοκιμίου και ο άξονας της πρέπει να είναι ευθυγραμμισμένος με τον άξονα του δοκιμίου και το κέντρο της πλάκας φόρτισης

Για τη μέτρηση της αντοχής σε μονοαξονική θλίψη (UCS, uniaxial compressive strength) στο εργαστήριο, κατασκευάζονται δοκίμια σύμφωνα με διεθνείς προδιαγραφές. Τα εργαστηριακά δοκίμια είναι συνήθως κυλινδρικά με μήκος διπλάσιο έως τριπλάσιο της διαμέτρου τους και φορτίζονται είτε με έλεγχο του φορτίου είτε με έλεγχο της μετατόπισης κατά το διαμήκη άξονά τους και συγχρόνως καταγράφεται η αντίστοιχη ανηγμένη παραμόρφωση που υφίστανται. Οι παραμορφώσεις αυτές προκύπτουν μετά από μετρήσεις διαστολής ή/και συστολής των υλικών, οι οποίες επιτυγχάνονται είτε με μηχανικά είτε με ηλεκτρικά μηκυνσιόμετρα κατά τη διαμήκη και εγκάρσια διεύθυνση του δοκιμίου, είτε μετρώντας τη σύγκλιση των πλακών, από τις οποίες εφαρμόζεται η θλιπτική τάση στο δοκίμιο (Αγιουτάντης, 2002).

3.9 Προσδιορισμός καμπύλης τάσης-παραμόρφωσης και ελαστικών σταθερών

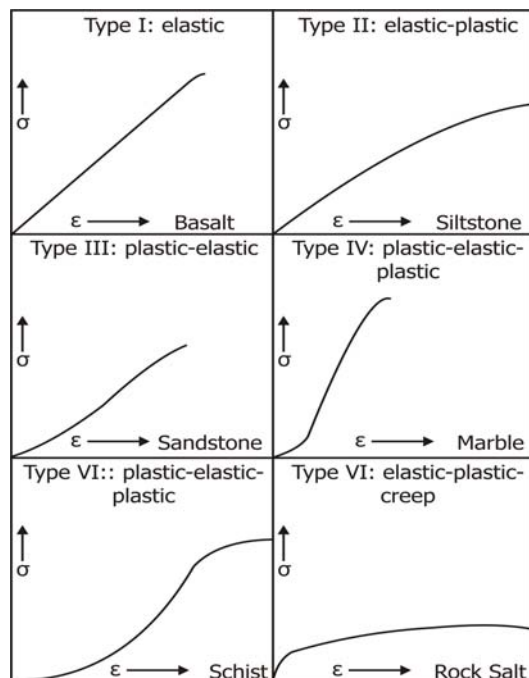
Το αντικείμενο της συγκεκριμένης μεθοδολογίας είναι ο προσδιορισμός του μέτρου ελαστικότητας και του λόγου του Poisson ενός πετρώματος, καθώς και η καταγραφή της καμπύλης τάσης – παραμόρφωσής του. Ο εξοπλισμός περιλαμβάνει τη συσκευή δοκιμής και τις πλάκες φόρτισης που χρησιμοποιούνται και στον προσδιορισμό της αντοχής σε μονοαξονική θλίψη και επιπλέον (Αγιουτάντης, 2002):



Σχήμα 3.2: Σχηματική παράσταση συσκευής θλιπτικών δοκιμών (Αγιουτάντης, 2002)

- Σύστημα μέτρησης παραμορφώσεων: Η μέτρηση τουλάχιστον δύο αξονικών και δύο πλευρικών τιμών των ανηγμένων παραμορφώσεων για κάθε στάδιο φόρτισης γίνεται χρησιμοποιώντας διάφορα όργανα ακριβείας, όπως ηλεκτρικά μηκυνσιόμετρα, γραμμικούς μεταβλητούς διαφορικούς πρεσσοστάτες, οπτικά μέσα, κλπ.
- Καταγραφικό όργανο: Συνίσταται η απευθείας καταγραφή της καμπύλης φορτίου-μετατόπισης με τη χρήση καταγραφικού οργάνου με δύο εισόδους.

Η καμπύλη που προκύπτει από την αποτύπωση των τιμών τάσης – παραμόρφωσης σε ορθογώνιο σύστημα αξόνων αποτελεί την συνάρτηση ή την καμπύλη τάσης – παραμόρφωσης για το εκάστοτε δοκίμιο και κατ' επέκταση για το πέτρωμα από το οποίο αυτό προέρχεται. Η καμπύλη αυτή αποτελείται συνήθως από δύο κλάδους: το κλάδο μέχρι τη διαρροή ή θραύση και το φθίνοντα κλάδο μετά τη διαρροή ή θραύση. Έτσι διακρίνονται καμπύλες που περιγράφουν είτε ελαστική συμπεριφορά (γραμμική ή μη), είτε πλαστική συμπεριφορά (γραμμική ή μη), είτε συνδυασμό αυτών. Το σχήμα 3.3 παρουσιάζει τυπικές μορφές τέτοιων συναρτήσεων που προκύπτουν από πειράματα σε διάφορα πετρώματα (Αγιουτάντης, 2002).



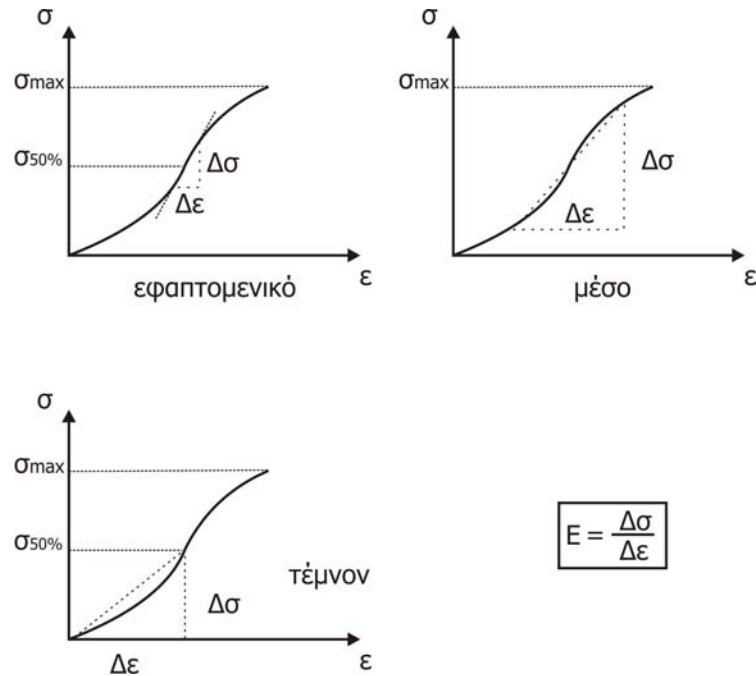
Σχήμα 3.3: Τυπικά διαγράμματα τάσης – παραμόρφωσης για πετρώματα (Αγιουτάντης, 2002)

Το μέτρο ελαστικότητας (E) του υλικού υπολογίζεται από την κλίση της καμπύλης τάσης – παραμόρφωσης, ενώ ο λόγος του Poisson (ν) υπολογίζεται από την αξονική και διαμετρική ανηγμένη παραμόρφωση σύμφωνα με τη σχέση (Αγιουτάντης, 2002):

$$\nu = -\frac{\varepsilon_x}{\varepsilon_y} \quad [3.4]$$

Από της τυπικές καμπύλες τάσης – παραμόρφωσης φαίνεται ότι το μέτρο ελαστικότητας δε παραμένει σταθερό κατά τη διαδικασία φόρτισης του υλικού. Έτσι, έχουν προταθεί διάφοροι τρόποι υπολογισμού ενός αντιπροσωπευτικού μέτρου ελαστικότητας (σχήμα 3.4), οι οποίοι έχουν ως εξής (Αγιουτάντης, 2002):

- Το εφαπτομενικό μέτρο ελαστικότητας (tangent modulus), που στη γενική περίπτωση υπολογίζεται από τη σχέση $d\sigma / d\varepsilon$, αποτελεί την κλίση της καμπύλης σε κάποιο σημείο της, το οποίο δίνεται ως ποσοστό της μέγιστης τάσης που δέχθηκε (π.χ. 50%).
- Το μέσο μέτρο ελαστικότητας (average modulus), που προκύπτει από τη μέση κλίση του ευθύγραμμου τμήματος της καμπύλης και υπολογίζεται από τη σχέση $\Delta\sigma / \Delta\varepsilon$.
- Το τέμνον μέτρο ελαστικότητας (secant modulus), που ορίζεται από την ευθεία που ενώνει τη συμβολή των αξόνων με κάποιο σημείο της καμπύλης, το οποίο δίνεται ως ποσοστό της μέγιστης τάσης που δέχθηκε το υλικό (π.χ. 50%). Το τέμνον μέτρο υπολογίζεται επίσης από τη σχέση $\Delta\sigma / \Delta\varepsilon$.

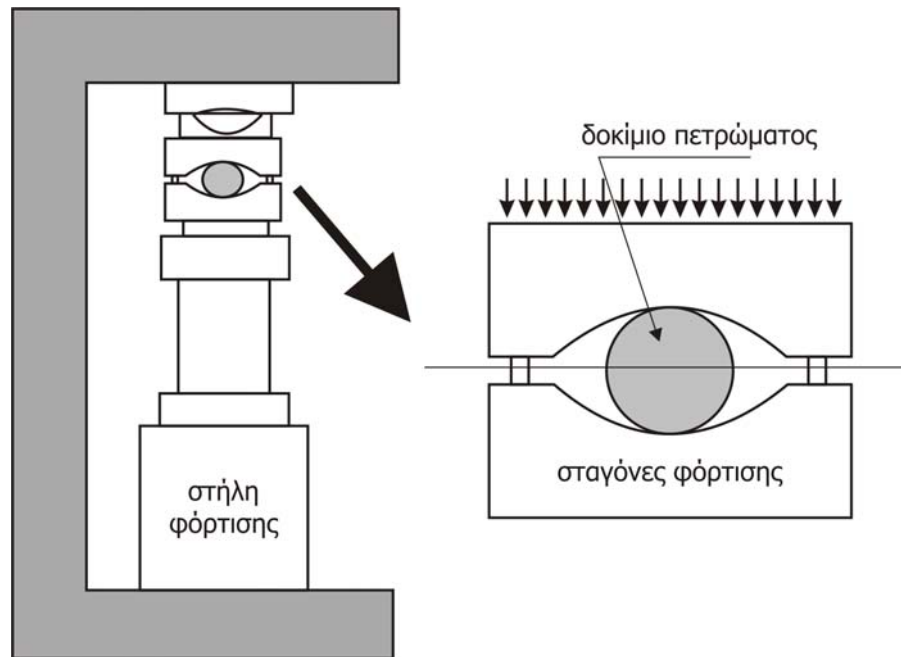


Σχήμα 3.4: Υπολογισμός του μέτρου ελαστικότητας του Young (Αγιουτάντης, 2002)

3.10 Έμμεσος προσδιορισμός της αντοχής σε εφελκυσμό

Το αντικείμενο της συγκεκριμένης μεθοδολογίας είναι ο έμμεσος προσδιορισμός της αντοχής σε εφελκυσμό ενός πετρώματος με τη χρήση δοκιμών κανονικής γεωμετρίας. Χρησιμοποιείται η μέθοδος Brazil (ή δοκιμή αντιδιαμετρικής θλίψης) που βασίζεται στην πειραματική παρατήρηση σύμφωνα με την οποία τα περισσότερα πετρώματα που βρίσκονται σε δισδιάστατα εντατικά πεδία αστοχούν σε εφελκυσμό, όταν η μία κύρια τάση είναι εφελκυστική και η άλλη κύρια τάση είναι θλιπτική με τιμή μικρότερη από το τριπλάσιο της εφελκυστικής τάσης. Στον εξοπλισμό περιλαμβάνονται τα ακόλουθα όργανα και συσκευές (Αγιουτάντης, 2002):

- Συσκευή δοκιμής: Η συσκευή θα πρέπει να μπορεί να επιβάλλει σταθερό ρυθμό φόρτισης στο δοκίμιο και συγχρόνως να έχει τη δυνατότητα να μετρήσει ή/και να καταγράψει το επιβαλλόμενο φορτίο.
- Πλάκες φόρτισης: Χρησιμεύουν στο να μεταφέρουν το φορτίο στο δοκίμιο και θα πρέπει να είναι παράλληλες μεταξύ τους.
- Σιαγόνες φόρτισης: Δύο χαλύβδινοι σιαγόνες φορτίζουν το δοκίμιο (σχήμα 3.5).
- Καταγραφικό όργανο: Συνίσταται η απευθείας καταγραφή της καμπύλης φορτίου-μετατόπισης με τη χρήση καταγραφικού οργάνου με δύο εισόδους.



Σχήμα 3.5: Σύστημα δοκιμής σε έμμεσο εφελκυσμό (Αγιουτάντης, 2002)

Η αντοχή σε εφελκυσμό (T_0) υπολογίζεται σύμφωνα με τη σχέση [3.5], όπου F_{\max} είναι το μέγιστο φορτίο που δέχθηκε το δοκίμιο μέχρι να αστοχήσει, D είναι η διάμετρος και t το πλάτος του δοκιμίου (Αγιουτάντης, 2002):

$$T_0 = \frac{2F_{\max}}{\pi Dt} = \frac{0.637F_{\max}}{Dt} \quad [3.5]$$

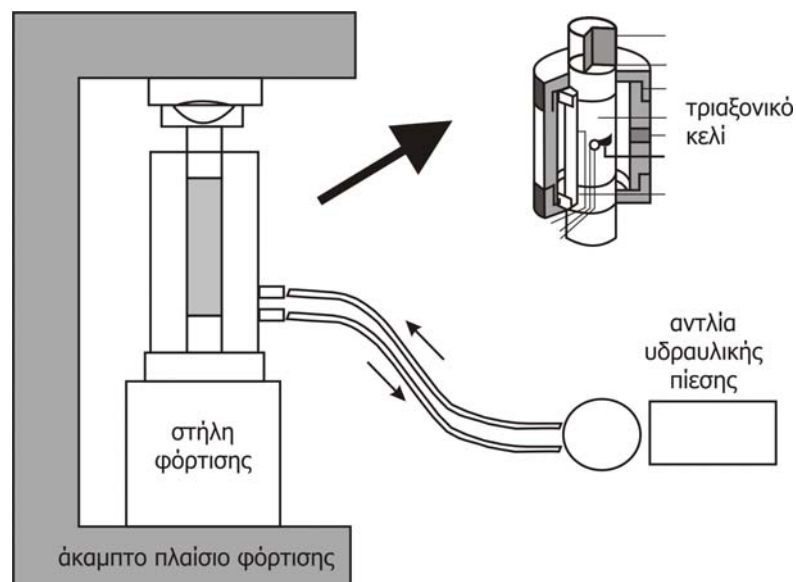
3.11 Προσδιορισμός της αντοχής σε τριαξονική καταπόνηση

Το αντικείμενο της συγκεκριμένης μεθοδολογίας είναι ο προσδιορισμός της αντοχής σε τριαξονική θλίψη για διάφορες τιμές πλευρικών τάσεων ενός πετρώματος. Τα αποτελέσματα των δοκιμών μπορούν να χρησιμοποιηθούν για τον υπολογισμό της γωνίας εσωτερικής τριβής (ϕ), της συνοχής (S_0) καθώς και της περιβάλλουσας αστοχίας του πετρώματος. Στον εξοπλισμό περιλαμβάνονται οι ακόλουθες κύριες συσκευές (Αγιουτάντης, 2002):

- Κυψέλη τριαξονικής δοκιμής: Η κυψέλη αυτή θα πρέπει να διαθέτει βαλβίδα εξαέρωσης, έναν υδραυλικό σύνδεσμο για να συνδεθεί με τη συσκευή επιβολής πλευρικών τάσεων, καθώς και κατάλληλο άνοιγμα για την επιβολή του αξονικού φορτίου. Απαραίτητη είναι και η χρήση ελαστικής μεμβράνης. Όστε να εμποδίζεται η εισροή υδραυλικού υγρού στους πόρους του υλικού.

- Σύστημα αξονικής φόρτισης: Αποτελείται από τη συσκευή δοκιμής που επιβάλλει και καταγράφει το φορτίο καθώς και από σφαιρικές κεφαλές έδρασης.
- Σύστημα πλευρικής φόρτισης: Περιλαμβάνει υδραυλική αντλία η οποία επιβάλλει πλευρική τάση στο δοκίμιο και ενδεικτικό πίεσης για την καταγραφή και παρακολούθηση της επιβαλλόμενης πλευρικής τάσης.

Η καταπόνηση των διαφόρων υλικών στο φυσικό τους χώρο (in situ) σε πολύ λίγες περιπτώσεις μπορεί να θεωρηθεί μονοαξονική. Πειραματικά δεδομένα αποδεικνύουν ότι, όταν εφαρμοσθούν πλευρικές τάσεις σε ένα δοκίμιο υπό αξονική καταπόνηση, τότε το δοκίμιο αστοχεί κάτω από υψηλότερες αξονικές τάσεις και συγχρόνως εμφανίζει μια ευρύτερη πλαστική περιοχή. Η συμπεριφορά των διαφόρων υλικών κάτω από την επίδραση πλευρικών τάσεων ή ισοδύναμα τα χαρακτηριστικά της αντίστοιχης καμπύλης τάσης παραμόρφωσης εξαρτώνται από τον τύπο του υλικού και το μέγεθος της τάσης. Δοκιμές αυτού του τύπου στο εργαστήριο χαρακτηρίζονται τριαξονικές δοκιμές (triaxial tests). Στις τριαξονικές δοκιμές χρησιμοποιούνται συνήθως κυλινδρικά δοκίμια όπου η επιβολή των πλευρικών τάσεων γίνεται με υδραυλικό τρόπο και θεωρείται ότι οι πλευρικές τάσεις παραμένουν σταθερές στο επίπεδο που είναι κάθετο προς τον άξονα φόρτισης ($\sigma_x = \sigma_y$). Σε παραλληλεπίπεδα ή κυβικά δοκίμια υπάρχει η δυνατότητα πραγματικής τριαξονικής δοκιμής, όμως στην πράξη, η πραγματοποίηση μιας τέτοιας δοκιμής απαιτεί αρκετά περίπλοκο εξοπλισμό και παρουσιάζει αρκετές δυσκολίες. Στο σχήμα 3.6 εμφανίζεται η διάταξη του εξοπλισμού τριαξονικών δοκιμών (Αγιουτάντης, 2002).

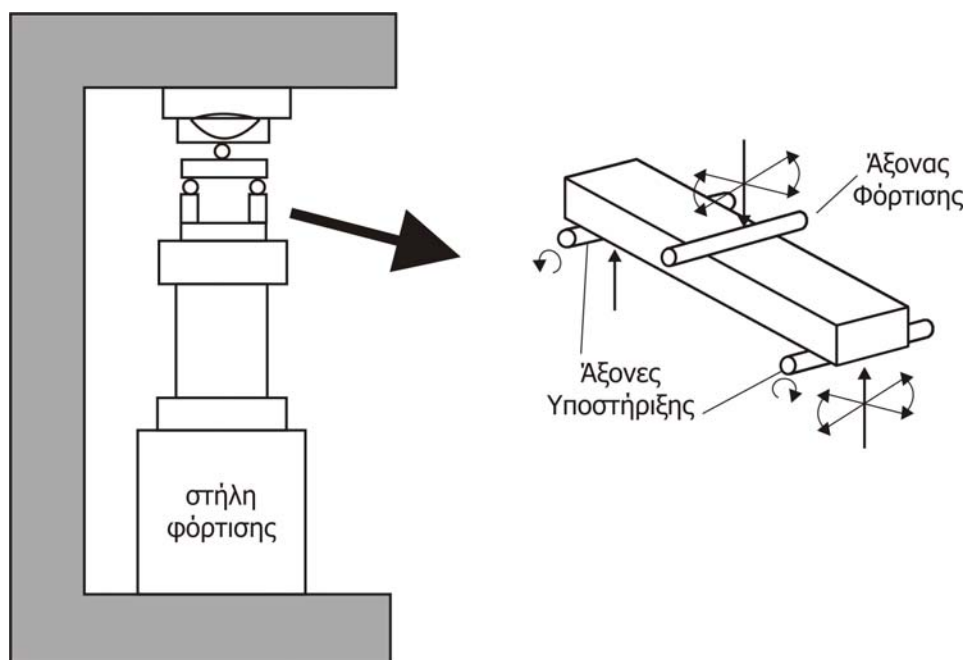


Σχήμα 3.6: Εξοπλισμός τριαξονικών δοκιμών

3.12 Προσδιορισμός της αντοχής σε κάμψη

Το αντικείμενο της συγκεκριμένης μεθοδολογίας είναι ο προσδιορισμός της αντοχής σε κάμψη (bending strength) ενός πετρώματος καθώς και η καταγραφή της καμπύλης τάσης – παραμόρφωσης του. Η συγκεκριμένη μεθοδολογία ονομάζεται και κάμψη τριών σημείων (3 point bending method). Ο εξοπλισμός είναι ίδιος με τον εξοπλισμό για τον προσδιορισμό της καμπύλης τάσης – παραμόρφωσης της μονοαξονικής δοκιμής και επιπλέον περιλαμβάνει τη διάταξη που φαίνεται στο σχήμα 3.7.

Η διάταξη αυτή αποτελείται από δύο παράλληλους κυλινδρικούς άξονες υποστήριξης στα άκρα του δοκιμίου και έναν άξονα φόρτισης στη μέση του δοκιμίου όπου ασκείται η δύναμη. Οι άξονες υποστήριξης πρέπει να τοποθετούνται κατά τέτοιο τρόπο ώστε μπορούν να περιστραφούν ελεύθερα προκειμένου να ελαχιστοποιηθεί η επιρροή της τριβής στη μέτρηση. Ένας από τους άξονες υποστήριξης πρέπει επιπλέον να είναι σε θέση να περιστρέφεται σε έναν άξονα κάθετο προς τον εαυτό του και παράλληλο με τους άξονες του δοκιμίου έτσι ώστε το δοκίμιο να μπορεί να ευθυγραμμιστεί όταν βρίσκεται υπό πίεση. Ο άξονας φόρτισης πρέπει επίσης να έχει τους ίδιους περιστροφικούς άξονες προκειμένου να εξασφαλιστεί η ομοιόμορφη εφαρμογή της δύναμης στο δοκίμιο.

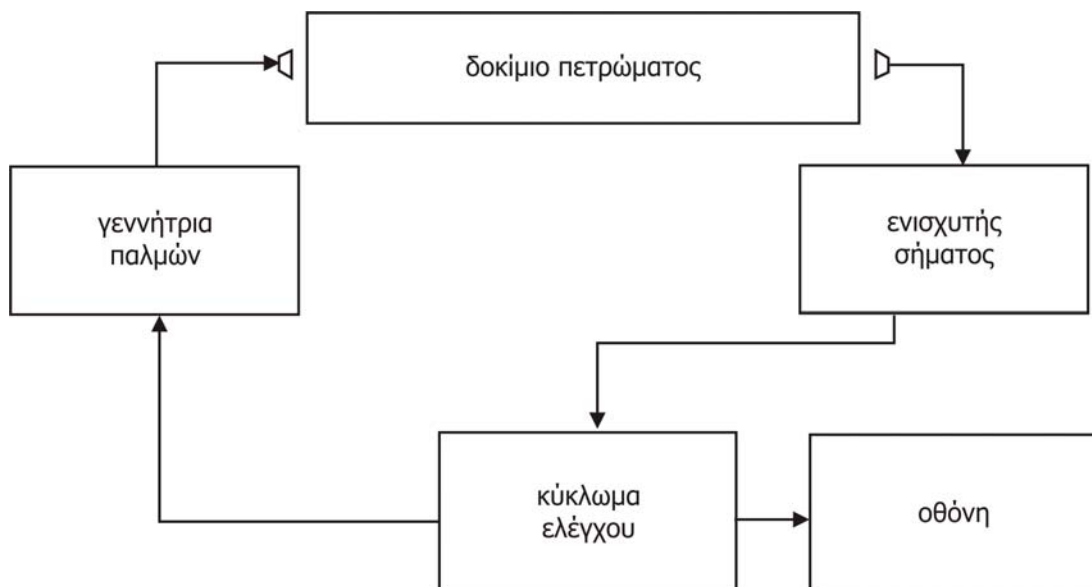


Σχήμα 3.7: Σχηματική αναπαράσταση συστήματος μέτρησης αντοχής σε κάμψη (κάμψη τριών σημείων)

3.13 Προσδιορισμός ταχυτήτων διάδοσης θλιπτικών και διατμητικών κυμάτων

Το αντικείμενο της συγκεκριμένης μεθοδολογίας είναι ο προσδιορισμός της ταχύτητας διάδοσης των ελαστικών κυμάτων σε ακέραιο πέτρωμα χρησιμοποιώντας εργαστηριακά δοκίμια. Τα αποτελέσματα των δοκιμών μπορούν να χρησιμοποιηθούν για τον υπολογισμό των ελαστικών σταθερών του πετρώματος όπως ο λόγος Poisson και το μέτρο ελαστικότητας. Η μέτρηση των σεισμικών ταχυτήτων (v_p και v_s) γίνεται είτε με υπέρηχους υψηλών συχνοτήτων (100 kHz – 10 MHz, high frequency ultrasonic pulse), είτε με υπέρηχους χαμηλών συχνοτήτων (2 – 30 kHz, low frequency ultrasonic pulse). Ένα τυπικό κύκλωμα μέτρησης του χρόνου διάδοσης των κυμάτων με τη βοήθεια γεννήτριας παλμών και συστήματος πομπού δέκτη κυμάτων φαίνεται στο σχήμα 3.8. Στον εξοπλισμό περιλαμβάνονται οι εξής κύριες συσκευές (Αγιουτάντης, 2002):

- Γεννήτρια παλμών.
- Πρεσσοστάτες: Ο πρεσσοστάτης εκπομπής μετατρέπει ηλεκτρικούς παλμούς σε μηχανικούς παλμούς, ενώ ο πρεσσοστάτης που δέχεται τα σήματα εκτελεί την αντίστροφη μετατροπή. Συνήθως οι πρεσσοστάτες αποτελούνται από κεραμικά υλικά με πιεζοηλεκτρικές ιδιότητες.
- Φίλτρα, ενισχυτές, μετρητές: Τα συστήματα αυτά επιλέγονται ανάλογα με την περιοχή συχνοτήτων του πειράματος.



Σχήμα 3.8: Κύκλωμα μέτρησης του χρόνου διάδοσης των κυμάτων σε δοκίμια πετρώματος

Όταν είναι γνωστή η βασική συχνότητα συντονισμού, όπως προκύπτει από την πειραματική διάταξη μέτρησης, τότε το μέτρο ελαστικότητας υπολογίζεται από τη σχέση [3.6], όπου f_b η βασική συχνότητα συντονισμού, l το μήκος του δοκιμίου και ρ η πυκνότητα του δοκιμίου (Αγιουτάντης, 2002):

$$E = 4f_b^2 l^2 \frac{\rho}{g} \quad [3.6]$$

ΚΕΦΑΛΑΙΟ 4

ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΔΙΑΧΕΙΡΙΣΗ ΠΕΙΡΑΜΑΤΩΝ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΠΕΤΡΩΜΑΤΩΝ

4.1 Εισαγωγή

Στο κεφάλαιο αυτό αρχικά γίνεται μια σύντομη περιγραφή των βασικών εννοιών της θεωρίας περί βάσεων δεδομένων. Στη συνέχεια περιγράφονται τα κύρια χαρακτηριστικά του συστήματος MySQL που επιλέχτηκε για την παρούσα εργασία. Τέλος, περιγράφεται αναλυτικά ο τρόπος κατασκευής και η δομή της βάσης δεδομένων στην οποία θα αποθηκεύονται οι διάφορες πληροφορίες σχετικά με τα πειράματα της Μηχανικής Πετρωμάτων.

4.2 Βάσεις δεδομένων

Ο όρος βάση δεδομένων είναι εξαιρετικά γενικός. Μια βάση δεδομένων μπορεί να είναι ένας απλός κατάλογος ονομάτων - διευθύνσεων σε κάποιο λογιστικό φύλλο του ευρύτατα γνωστού προγράμματος Microsoft Excel ή μπορεί να είναι τα αρχεία μιας εταιρείας τηλεπικοινωνιών στα οποία καταχωρούνται καθημερινά εκατομμύρια κλήσεις και επιπλέον υπολογίζονται με ακρίβεια οι αντίστοιχες χρεώσεις. Μια βάση δεδομένων μπορεί να αποτελεί μια αυτόνομη λειτουργία (σε κάποιον προσωπικό υπολογιστή ενός χρήστη), αλλά μπορεί επίσης να χρησιμοποιείται από χιλιάδες χρήστες ταυτόχρονα με τα δεδομένα να μοιράζονται σε πολλούς ηλεκτρονικούς υπολογιστές. Το μέγεθος μιας βάσης δεδομένων σε χώρο υπολογιστικής μνήμης, μπορεί να κυμανθεί από το μέγεθος μιας απλής φωτογραφίας μέχρι το μέγεθος πολλών σκληρών δίσκων τελευταίας τεχνολογίας (από μερικά kilobyte μέχρι και αρκετά terabyte) (Kofler, 2004).

Από τα παραπάνω προκύπτει ότι σαν βάση δεδομένων μπορεί να χαρακτηριστεί μια ολοκληρωμένη συλλογή δεδομένων. Υπάρχουν πολλές διαφορετικές στρατηγικές για την οργάνωση των δεδομένων σε βάσεις δεδομένων καθώς και για τη διευκόλυνση της εύκολης πρόσβασης και τον χειρισμό των δεδομένων. Ένα σύστημα διαχείρισης βάσεων δεδομένων (database management system – DBMS) παρέχει μηχανισμούς για την αποθήκευση και την οργάνωση των δεδομένων με τρόπο που είναι σύμφωνος με τη

μορφή της βάσης δεδομένων. Τα συστήματα διαχείρισης βάσεων δεδομένων δίνουν τη δυνατότητα στους προγραμματιστές να προσπελαίνουν και να αποθηκεύουν δεδομένα, χωρίς να ενδιαφέρονται για την εσωτερική αναπαράσταση των βάσεων δεδομένων στον ηλεκτρονικό υπολογιστή (Deitel κ.α., 2002).

Τα σημερινά δημοφιλέστερα συστήματα βάσεων δεδομένων είναι σχεσιακές βάσεις δεδομένων (relational databases). Σχεδόν καθολικά, οι σχεσιακές βάσεις δεδομένων χρησιμοποιούν μία γλώσσα που ονομάζεται Δομημένη Γλώσσα Ερωτημάτων (Structured Query Language – SQL) για να εκτελούν ερωτήματα (queries), να αναζητούν δηλαδή δεδομένα και πληροφορίες που ικανοποιούν προκαθορισμένα κριτήρια. Ορισμένα από τα δημοφιλέστερα, σε επίπεδο επιχειρήσεων, σχεσιακά συστήματα βάσεων δεδομένων είναι τα Microsoft SQL Server, Oracle, DB2, Sybase, Informix και MySQL (Deitel κ.α., 2002).

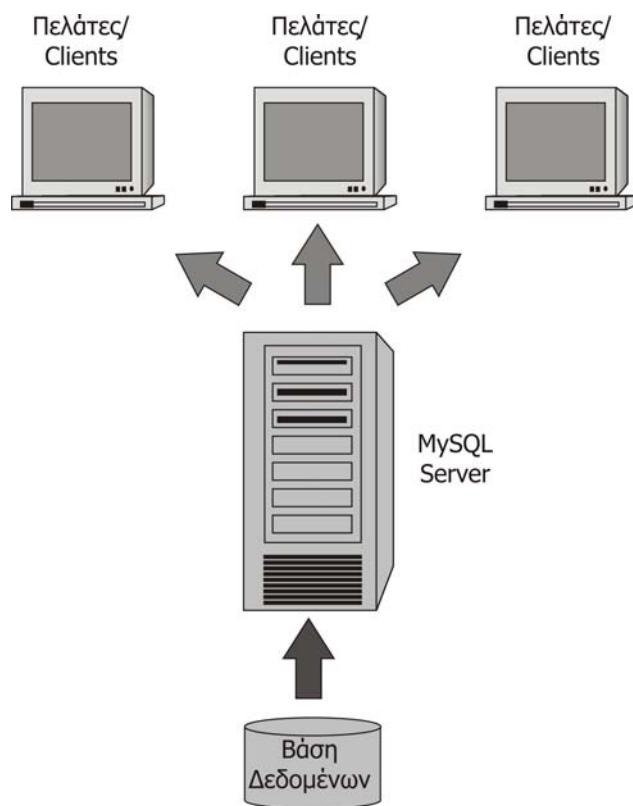
4.3 Σύστημα βάσεων δεδομένων MySQL

Το MySQL είναι ένα από τα δημοφιλέστερα σχεσιακά συστήματα βάσεων δεδομένων. Κατασκευάστηκε από την εταιρεία MySQL AB η οποία αρχικά δημιουργήθηκε στη Σουηδία από τους David Axmark, Allan Larsson και Michael Widenius. Η πρώτη του έκδοση (MySQL 3.23.n) δημοσιεύτηκε τον Απρίλιο του 1999. Η έκδοση που χρησιμοποιήθηκε για την ανάπτυξη της βάσης δεδομένων για τις ανάγκες τις παρούσας εργασίας είναι η MySQL 4.1 (δημοσιεύτηκε το 2003), ενώ η πιο σύγχρονη έκδοση του συστήματος δημοσιεύτηκε στα μέσα του 2005 και πρόκειται για την MySQL 5.1. Το σύστημα MySQL είναι μια εφαρμογή «ανοιχτού κώδικα» (open source), γεγονός το οποίο σημαίνει ότι οποιοσδήποτε χρήστης μπορεί να το χρησιμοποιήσει δωρεάν αλλά και να το τροποποιήσει είτε για να το βελτιώσει είτε για να το προσαρμόσει στις διάφορες προκύπτουσες ανάγκες.

Το σύστημα MySQL, όπως υποδηλώνεται από το όνομα του, υποστηρίζει τη Δομημένη Γλώσσα Ερωτημάτων (SQL) από την οποία διακρίνονται αρκετές διαφορετικές διάλεκτοι. Το σύστημα MySQL ακολουθεί το πρότυπο ANSI-SQL/92. Επιπρόσθετα, το MySQL έχει αρχιτεκτονική τύπου πελάτη/διακομιστή (client/server). Ο διακομιστής (server) είναι το ίδιο το σύστημα MySQL και είναι υπεύθυνος για τον ορισμό, την ασφάλεια, το χειρισμό και την ακεραιότητα των δεδομένων. Οι πελάτες είναι οι διάφορες εφαρμογές που εκτελούνται πάνω από το σύστημα MySQL και είναι εφαρμογές είτε γραμμένες από το χρήστη είτε ενσωματωμένες από τον κατασκευαστή του συστήματος. Υπάρχουν πολλές

εμπορικές εφαρμογές client για την απλούστερη διαχείριση του MySQL Server, όπως για παράδειγμα το πρόγραμμα MySQL-Front που χρησιμοποιήθηκε κατά την συγγραφή της παρούσας εργασίας. Τέτοια προγράμματα μπορούν να βρίσκονται είτε στον ίδιο υπολογιστή με τον MySQL Server είτε σε διαφορετικό υπολογιστή και να επικοινωνούν με αυτόν μέσω διαδικτύου. Στο σχήμα 4.1 εμφανίζεται σχηματικά η αρχιτεκτονική πελάτη/διακομιστή που ακολουθεί το σύστημα MySQL.

Μερικά από τα βασικά χαρακτηριστικά που καθιστούν το σύστημα MySQL ως ένα από τα δημοφιλέστερα του είδους, είναι η ταχύτητα του, η αξιοπιστία του και η ευκολία που παρέχει στη χρήση του. Εξίσου σημαντικό χαρακτηριστικό του αποτελεί η ανεξαρτησία του από το είδος του λειτουργικού συστήματος του ηλεκτρονικού υπολογιστή. Το σύστημα MySQL μπορεί να συνεργαστεί με ένα μεγάλο αριθμό λειτουργικών συστημάτων μερικά από τα οποία είναι τα Apple Macintosh OS X, Linux και Microsoft Windows. Το σύστημα MySQL συνεργάζεται επίσης άριστα με ένα πλήθος γλωσσών προγραμματισμού όπως οι C++, PHP, Java και Visual Basic .NET μέσω διαφόρων προγραμμάτων διασύνδεσης που έχουν αναπτυχθεί (Kofler, 2004).



Σχήμα 4.1: Αρχιτεκτονική πελάτη/διακομιστή

4.4 Πίνακες και πρωτεύοντα κλειδιά

Μια σχεσιακή βάση δεδομένων συντίθεται από πίνακες. Στο σχήμα 4.2 εμφανίζεται ένα παράδειγμα πίνακα που μπορεί να χρησιμοποιηθεί σε κάποιο σύστημα. Ο πίνακας ονομάζεται Employee (Εργαζόμενοι) και βασικός του σκοπός είναι να δείχνει τα συγκεκριμένα χαρακτηριστικά διαφόρων εργαζομένων κάποιας εταιρείας. Μια συγκεκριμένη σειρά του πίνακα ονομάζεται εγγραφή (record) ενώ μια συγκεκριμένη στήλη του πίνακα ονομάζεται πεδίο (field). Το πρωτεύον κλειδί (primary key) είναι το πεδίο εκείνο του πίνακα που περιέχει μοναδικά δεδομένα ή δεδομένα που δεν εμφανίζονται σε άλλες εγγραφές αυτού του πίνακα. Η ύπαρξη του πρωτεύοντος κλειδιού σ' έναν πίνακα εγγυάται ότι κάθε εγγραφή μπορεί να προσδιοριστεί από μία τουλάχιστον μοναδική τιμή.

Ο σκοπός του πρωτεύοντος κλειδιού είναι να εντοπίζει όσο το δυνατόν ταχύτερα μία συγκεκριμένη εγγραφή σε έναν πίνακα (παραδείγματος χάριν, για να εντοπίσει την εγγραφή με id=314159 από έναν πίνακα εκατομμύριο εγγραφών). Με τα περισσότερα συστήματα βάσεων δεδομένων, συμπεριλαμβανομένου του MySQL, επιτρέπεται το πρωτεύον κλειδί ενός πίνακα να δημιουργείται από περισσότερα από ένα πεδία του πίνακα. Το πρωτεύον κλειδί ενός πίνακα θα πρέπει να πληροί της παρακάτω προϋποθέσεις (Kofler, 2004):

- Το πρωτεύον κλειδί θα πρέπει να είναι μοναδικό και κατά συνέπεια δεν επιτρέπεται δύο εγγραφές να έχουν το ίδιο περιεχόμενο στο πεδίο που έχει χαρακτηριστεί ως πρωτεύον κλειδί.
- Το πρωτεύον κλειδί πρέπει να είναι συμπαγές διότι είναι απαραίτητο να διατηρηθεί ένας δείκτης γι' αυτό (ο αρχικός δείκτης) για να μεγιστοποιήσει την ταχύτητα της αναζήτησης (π.χ., για id=314159). Όσο συμπαγέστερο είναι το αρχικό κλειδί, τόσο αποδοτικότερη είναι η διαχείριση αυτού του δείκτη. Επομένως, ένας ακέραιος αριθμός είναι καταλληλότερος από μια ακολουθία χαρακτήρων μεταβλητού μήκους, για χρήση ως πρωτεύον κλειδί.

Στα περισσότερα συστήματα βάσεων δεδομένων, συμπεριλαμβανομένου και του MySQL συνηθίζεται να χρησιμοποιείται ένας ακέραιος αριθμός ως πρωτεύον κλειδί, που παράγεται αυτόματα σε σειρά (1 ..2 ..3 ...) από το σύστημα βάσεων δεδομένων. Κατά συνέπεια ούτε ο προγραμματιστής ούτε ο χρήστης δεν χρειάζεται να ενδιαφερθεί για τον τρόπο που κάθε νέα εγγραφή θα πρέπει να αντιστοιχίζεται σε μία μοναδική τιμή (Kofler, 2004).

	id	name	surname	email	salary	job
	1	Aggeliki	Maragaki	maragaki@yahoo.gr	1000	Developer
	2	Tom	Vallas	tom@hotmail.com	1200	Manager
	3	George	Kontos	kontos@mred.tuc.gr	1900	Designer
Εγγραφή {	4	Iwn	Kontiadis	iwn@yahoo.gr	1500	Developer
	5	Gewrgia	Papadimitriou	g_papdim@yahoo.gr	900	Secretary
	6	Nikos	Mitritsakis	nik@hotmail.com	1400	Designer
	7	Katerina	Mpounou	mpounou@mred.tuc	700	Manager

Πρωτεύον κλειδί

Πεδίο

Σχήμα 4.2: Παράδειγμα πίνακα Employee

4.5 Βασικοί τύποι δεδομένων του MySQL

Κατά τη σχεδίαση ενός πίνακα είναι απαραίτητο να οριστεί ο τύπος του κάθε πεδίου. Όπως φαίνεται στον πίνακα Employee του σχήματος 4.2, κάποιο πεδίο μπορεί να δέχεται τιμές ακολουθίας χαρακτήρων (string) όπως για παράδειγμα το πεδίο surname ενώ κάποιο άλλο πεδίο μπορεί να δέχεται ακέραιες τιμές, όπως για παράδειγμα το πεδίο salary. Η MySQL διαθέτει ένα μεγάλο αριθμό τύπων δεδομένων, οι βασικότεροι από του οποίους εμφανίζονται και επεξηγούνται στον πίνακα 4.1.

Πίνακας 4.1: Βασικοί τύποι δεδομένων του MySQL

Ονομασία	Περιοχή τιμών	Μέγεθος (bytes)
TINYINT	Ακέραιος από -128 ως +127	1 byte
SMALLINT	Ακέραιος από -32.769 ως +32.767	2 bytes
MEDIUMINT	Ακέραιος από - 8.388.608 ως +8.388.607	3 bytes
INT	Ακέραιος από - 2.147.483.648 ως +2.147.483.647	4 bytes
BIGINT	Ακέραιος $\pm 9.22 \times 10^{18}$	8 bytes
FLOAT	Δεκαδικός, με ακρίβεια 8 θέσεων υποδιαστολής	4 bytes
DOUBLE	Δεκαδικός, με ακρίβεια 16 θέσεων υποδιαστολής	8 bytes
DATE	Ημερομηνία με μορφή '2003-12-31', από 1000-01-01 to 9999-12-31	3 bytes
TIME	Ώρα με μορφή '23:59:59', $\pm 838 : 59 : 59$	3 bytes
CHAR	Ακολουθία χαρακτήρων (string) με σταθερό μήκος	Μέγιστο 255 bytes
VARCHAR	Ακολουθία χαρακτήρων (string) με μεταβλητό μήκος	Μέγιστο 255 bytes
TEXT	Ακολουθία χαρακτήρων (string) με μεταβλητό μήκος	Μέγιστο 65.535 bytes
ENUM	Διαλέγει μια από προκαθορισμένες τιμές string (ως 65.535)	1 ή 2 bytes

4.6 Δομημένη γλώσσα ερωτημάτων (SQL)

Η SQL αναπτύχθηκε αρχικά στο τμήμα έρευνας της IBM, στις αρχές της δεκαετίας του 1970 και υλοποιήθηκε για πρώτη φορά σε μεγάλη κλίμακα στο σχεσιακό πρότυπο System R της IBM. Από τις διάφορες διαλέκτους της SQL έχουν προκύψει ένα αμερικάνικο εθνικό πρότυπο (ANSI), ένα διεθνές πρότυπο (ISO), ένα πρότυπο X (X/Open), και ένα αμερικάνικο ομοσπονδιακό πρότυπο επεξεργασίας πληροφοριών (Federal Information Processing Standard – FIPS). Όπως προαναφέρθηκε το σύστημα MySQL ακολουθεί το πρότυπο ANSI-SQL/92. Η SQL χρησιμοποιείται για τη διατύπωση σχεσιακών πράξεων, δηλαδή πράξεων που ορίζουν και χειρίζονται δεδομένα σε σχεσιακή μορφή (Date, 1995).

4.6.1 Η βασική εντολή δημιουργίας πίνακα CREATE

Η εντολή της SQL για τη δημιουργία ενός νέου πίνακα κατά το σχεδιασμό μιας βάσης δεδομένων είναι η εντολή CREATE. Η συγκεκριμένη εντολή συντάσσεται ως εξής:

```
CREATE TABLE όνομαΠίνακα
(
  όνομαΠεδίου1  τύποςΠεδίου1 (μέγεθοςΠεδίου1),
  όνομαΠεδίου2  τύποςΠεδίου2 (μέγεθοςΠεδίου2),
  ...
PRIMARY KEY (όνομαΠεδίουΠρωτεύονΚλειδί)
)
```

Ακολουθεί ο SQL κώδικας που δημιουργεί τον πίνακα Employee του σχήματος 4.2, ο οποίος αποτελείται από τα πεδία id (τύπου ακέραιος με μέγεθος ως 3 ψηφία), surname, email και job (τύπου ακολουθία χαρακτήρων με μέγεθος ως 32 χαρακτήρες) και salary (τύπου ακέραιος με μέγεθος ως 10 ψηφία).

```
CREATE TABLE Employee
(
  id int(3) UNSIGNED NOT NULL AUTO_INCREMENT,
  name varchar(32) NOT NULL,
  surname varchar(32) NOT NULL,
  email varchar(32) NULL,
  salary int(10) UNSIGNED NOT NULL,
  job varchar(32) NOT NULL,
PRIMARY KEY (id)
)
```

Όπως γίνεται αντιληπτό από την ανάγνωση του πιο πάνω κώδικα εκτός από τα ονόματα των πεδίων του πίνακα, καθώς και τον τύπο των δεδομένων που θα δέχεται το κάθε πεδίο, είναι δυνατόν να καθοριστούν και άλλοι παράμετροι που αφορούν το εκάστοτε πεδίο κατά τη δημιουργία του πίνακα. Μια βασική τέτοια παράμετρος ενός πεδίου είναι το αν το πεδίο θα πρέπει να συμπληρώνεται υποχρεωτικά ή αν θα μπορεί να δέχεται και την τιμή «τίποτα» (null). Στην SQL η παράμετρος αυτή καθορίζεται με τις φράσεις NOT NULL και NULL αντίστοιχα, ακριβώς μετά τη δήλωση του τύπου του πεδίου. Έτσι στον πίνακα Employee όλα τα πεδία είναι υποχρεωτικά (εφόσον υποχρεωτικά οι υπάλληλοι μιας εταιρείας έχουν όνομα, επίθετο, μισθό κλπ.) εκτός από το πεδίο email το οποίο δεν είναι υποχρεωτικό (εφόσον δεν είναι υποχρεωτικό ότι όλοι οι υπάλληλοι μια εταιρείας έχουν και κάποιο email).

Η φράση UNSIGNED που ακολουθεί μετά τη δήλωση των πεδίων id και salary καθορίζει ότι τα δεδομένα των πεδίων αυτών που είναι τύπου INT (ακέραιοι αριθμοί), δεν μπορούν να πάρουν αρνητικές τιμές. Η φράση auto_increment καθορίζει ότι τα δεδομένα του πεδίου id παράγονται αυτόματα σε σειρά (1 ..2 ..3 ...) από το σύστημα βάσεων δεδομένων MySQL, και συνηθίζεται να ακολουθεί τη δήλωση των πεδίων που αποτελούν τα πρωτεύοντα κλειδιά των πινάκων.

4.6.2 Το βασικό ερώτημα SELECT

Συχνά, διαφορετικοί χρήστες μιας βάσης δεδομένων ενδιαφέρονται για διαφορετικά δεδομένα και απαιτούν μόνο υποσύνολα των πινάκων. Ένα τυπικό ερώτημα SQL επιλέγει (selects) πληροφορίες από έναν ή και περισσότερους πίνακες σε μια βάση δεδομένων. Τέτοιου τύπου επιλογές εκτελούνται από τα ερωτήματα SELECT. Τα αποτελέσματα ενός ερωτήματος ονομάζονται σύνολα αποτελεσμάτων. Το συγκεκριμένο ερώτημα συντάσσεται ως εξής (Deitel κ.α., 2002):

```
SELECT όνομαΠεδίου1, όνομαΠεδίου2, ...  
FROM όνομαΠίνακα
```

Στο σχήμα 4.3 εμφανίζονται ως παραδείγματα, δύο ερωτήματα SELECT που εφαρμόζονται στον πίνακα Employee καθώς και τα σύνολα αποτελεσμάτων που παράγονται κατά την εκτέλεση τους. Το σύμβολο του αστερίσκου (*) στο πρώτο ερώτημα δηλώνει ότι θα πρέπει να επιλεγούν όλα τα πεδία του πίνακα.

1 SELECT * FROM Employee

2

ic	name	surname	email	salary
1	Aggeliki	Maragaki	maragaki@yahoo.gr	1000
2	Tom	Vallas	tom@hotmail.com	1200
3	George	Kontos	kontos@mred.tuc.gr	1900
4	Iwn	Kontiadis	iwn@yahoo.gr	1500
5	Gewrgia	Papadimitriou	g_papdim@yahoo.gr	900
6	Nikos	Mitritsakis	nik@hotmail.com	1400
7	Katerina	Mpounou	mpounou@mred.tuc	700

1 SELECT name, surname, salary

2 FROM Employee

3

name	surname	salary
Aggeliki	Maragaki	1000
Tom	Vallas	1200
George	Kontos	1900
Iwn	Kontiadis	1500
Gewrgia	Papadimitriou	900
Nikos	Mitritsakis	1400
Katerina	Mpounou	700

Σχήμα 4.3: Παραδείγματα ερωτημάτων SELECT και τα αντίστοιχα σύνολα αποτελεσμάτων

Στις περισσότερες περιπτώσεις, οι χρήστες ψάχνουν μια βάση δεδομένων για εγγραφές που ικανοποιούν ορισμένα κριτήρια αναζήτησης. Η SQL χρησιμοποιεί την προαιρετική φράση WHERE σε ένα ερώτημα SELECT για να καθορίσει τα κριτήρια αναζήτησης για το ερώτημα. Η συνθήκη της φράσης WHERE μπορεί να περιέχει τους τελεστές <, >, <=, >=, = και <>. Η απλούστερη μορφή ενός ερωτήματος SELECT που περιλαμβάνει κριτήρια αναζήτησης είναι (Deitel κ.α., 2002):

```
SELECT όνομαΠεδίου1, όνομαΠεδίου2, ...
FROM όνομαΠίνακα
WHERE κριτήρια
```

Στο σχήμα 4.4 εμφανίζεται ως παράδειγμα ένα ερώτημα SELECT με κριτήριο, το οποίο εφαρμόζεται στον πίνακα Employee καθώς και το σύνολο του αποτελέσματος που παράγεται από την εκτέλεση του συγκεκριμένου ερωτήματος. Το κριτήριο για την επιλογή εγγραφών στο συγκεκριμένο ερώτημα είναι ο μισθός του υπαλλήλου να υπερβαίνει τον αριθμό 1200 (ευρώ).

```
1 SELECT id, name, surname, email, salary
2 FROM Employee
3 WHERE salary>1200
```

id	name	surname	email	salary
3	George	Kontos	kontos@mred.tuc.gr	1900
4	Iwn	Kontiadis	iwn@yahoo.gr	1500
6	Nikos	Mitritsakis	nik@hotmail.com	1400

Σχήμα 4.4: Παράδειγμα ερωτήματος SELECT με κριτήριο και το αντίστοιχο σύνολο αποτελέσματος

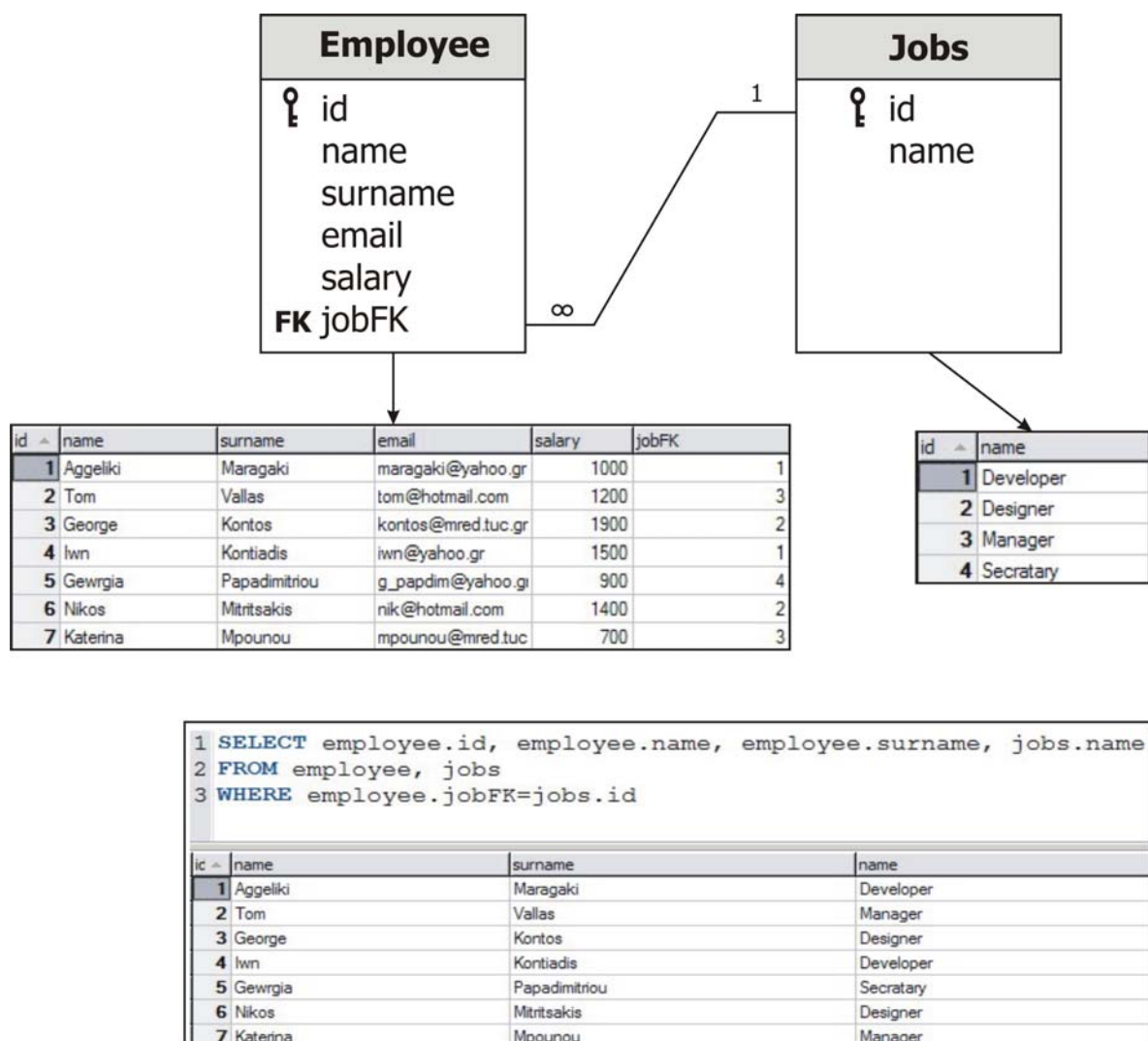
4.6.3 Σχέση «ένα προς πολλά»

Οι σχεδιαστές βάσεων δεδομένων συχνά χωρίζουν σχετικά δεδομένα σε πολλούς πίνακες, ώστε να εξασφαλισθεί το ότι η βάση δεδομένων δεν αποθηκεύει πλεονάζοντα δεδομένα χωρίς σκοπό. Για παράδειγμα στον πίνακα Employee του σχήματος 4.1 γίνεται εύκολα αντιληπτό ότι επαναλαμβάνονται όροι απασχόλησης στο πεδίο jobs. Σε τέτοιες περιπτώσεις συνίσταται η δημιουργία ενός δεύτερου πίνακα (σχήμα 4.5) με το όνομα Jobs ο οποίος θα αποτελείται από δύο πεδία τα id (πρωτεύον κλειδί του συγκεκριμένου πίνακα) και name (το όνομα του κάθε τύπου απασχόλησης). Στον πίνακα Employee δεν θα αποθηκεύεται πλέον το όνομα του τύπου της απασχόλησης του εκάστοτε εργαζόμενου, αλλά ο ακέραιος αριθμός που αντιστοιχεί στον συγκεκριμένο τύπο απασχόλησης από τον πίνακα Jobs. Το πεδίο Jobs του πίνακα Employee αποτελεί ένα ξένο κλειδί (foreign key - FK) εφόσον αντιστοιχεί με το πρωτεύον κλειδί ενός άλλου πίνακα. Έτσι στον πίνακα Employee το πεδίο Jobs μετονομάζεται σε JobsFK και ο τύπος δεδομένων του είναι πλέον ακέραιος αριθμός (Integer).

Η σχέση που είναι απαραίτητο να δημιουργηθεί μεταξύ των δύο πινάκων, ώστε να είναι δυνατή η συνένωση τους ονομάζεται «σχέση ένα προς πολλά» (one to many relationship, 1:∞) διότι κάθε εργαζόμενος της εταιρείας (πίνακας Employee) αντιστοιχεί σε ένα και μόνο ένα τύπο απασχόλησης, ενώ κάθε τύπος απασχόλησης (πίνακας Jobs) μπορεί να αντιστοιχεί σε πολλούς εργαζόμενους. Στο σχήμα 4.5 εμφανίζονται οι δύο νέοι πίνακες (με τα πεδία τους) που προέκυψαν από τον αρχικό πίνακα Employee, η συμβολική αναπαράσταση της ένωσης ένα προς πολλά καθώς και το ερώτημα SELECT με το οποίο επιτυγχάνεται η συνένωση των δύο νέων πινάκων για την σαφέστερη παρουσίαση του αντίστοιχου συνόλου αποτελέσματος.

Στο ερώτημα SELECT του σχήματος 4.5 γίνεται επιλογή πεδίων από δύο πίνακες (Employee και Jobs). Συχνά, κατά το σχεδιασμό μιας βάσης, τα ονόματα των πεδίων διαφορετικών πινάκων συμβαίνει να ταυτίζονται. Για παράδειγμα και οι δύο πίνακες του σχήματος 4.5 έχουν πεδία με τα ονόματα id και name. Για να αποφεύγονται τυχόν τέτοιου είδους συγχύσεις, η SQL χρησιμοποιεί τη σύνταξη όνομαΠίνακα.όνομαΠεδίου (αντί του όνομαΠεδίου) για την αναφορά σε ένα πεδίο, κατά την εκτέλεση ερωτημάτων που επιλέγουν πεδία από περισσότερους από έναν πίνακες. Η σύνταξη αυτή χρησιμοποιείται και στο ερώτημα του σχήματος 4.5 όπου υλοποιείται η συνένωση μεταξύ των δύο πινάκων.

Εκτός από την σχέση «ένα προς πολλά» στη θεωρία βάσεων δεδομένων διακρίνονται και άλλοι τύποι σχέσεων που μπορούν να συνδέουν δύο πίνακες. Η πιο συνηθισμένη από αυτές είναι η «σχέση πολλά προς πολλά» (many to many relationship). Αν υποθεθεί ότι στη βάση δεδομένων του παραδείγματος κρίνεται αναγκαία η δημιουργία ενός πίνακα (Publications) για την αποθήκευση των δημοσιεύσεων κάθε εργαζομένου, τότε η σχέση που θα συνέδεε τον πίνακα αυτόν με τον πίνακα Employee θα ήταν «πολλά προς πολλά», διότι κάθε εργαζόμενος είναι δυνατόν να έχει πολλές δημοσιεύσεις, αλλά και κάθε δημοσίευση μπορεί να αντιστοιχεί σε πολλούς εργαζόμενους.



Σχήμα 4.5: Παράδειγμα ένωσης πινάκων με σχέση «ένα προς πολλά»

4.6.4 Η φράση ORDER BY

Τα αποτελέσματα ενός ερωτήματος μπορούν να ταξινομηθούν σε αύξουσα ή φθίνουσα σειρά χρησιμοποιώντας την προαιρετική φράση ORDER BY. Η απλούστερη μορφή μιας φράσης ORDER BY είναι (Deitel κ.α., 2002):

```
SELECT όνομαΠεδίου1, όνομαΠεδίου2, ...  
FROM όνομαΠίνακα  
ORDER BY πεδίο ASC ή DESC
```

Η λέξη ASC καθορίζει αύξουσα (ascending) σειρά (από τα χαμηλότερα στα υψηλότερα) ενώ η λέξη DESC καθορίζει φθίνουσα (descending) σειρά (από τα υψηλότερα στα χαμηλότερα). Η προεπιλεγμένη (default) σειρά ταξινόμησης είναι η αύξουσα και επομένως όταν δε χρησιμοποιείται καμία από τις δύο αυτές λέξεις η ταξινόμηση γίνεται με αύξουσα σειρά. Το πεδίο καθορίζει το πεδίο του πίνακα σύμφωνα με το οποίο θα γίνει η ταξινόμηση. Στο σχήμα 4.6 εμφανίζονται ως παραδείγματα δύο ερωτήματα ταξινόμησης με αύξουσα και φθίνουσα σειρά αντίστοιχα, καθώς και τα αντίστοιχα σύνολα αποτελεσμάτων που παράγονται κατά την εκτέλεσή τους. Η ταξινόμηση στο πρώτο ερώτημα γίνεται με βάση το πεδίο surname του πίνακα Employee, ενώ στο δεύτερο η ταξινόμηση υλοποιείται με βάση το πεδίο salary του ίδιου πίνακα.

1 SELECT id, name, surname, salary FROM employee 2 ORDER BY surname ASC			
id	name	surname	salary
4	Iwn	Kontiadis	1500
3	George	Kontos	1900
1	Aggeliki	Maragaki	1000
6	Nikos	Mitritsakis	1400
7	Katerina	Mpounou	700
5	Gewrgia	Papadimitriou	900
2	Tom	Vallas	1200

1 SELECT id, name, surname, salary FROM employee 2 ORDER BY salary DESC			
id	name	surname	salary
3	George	Kontos	1900
4	Iwn	Kontiadis	1500
6	Nikos	Mitritsakis	1400
2	Tom	Vallas	1200
1	Aggeliki	Maragaki	1000
5	Gewrgia	Papadimitriou	900
7	Katerina	Mpounou	700

Σχήμα 4.6: Παραδείγματα ερωτημάτων ταξινόμησης με αύξουσα και φθίνουσα σειρά αντίστοιχα

4.6.5 Η φράση INSERT

Η φράση INSERT εισάγει μια νέα εγγραφή σε έναν πίνακα. Η απλούστερη μορφή για τη συγκεκριμένη φράση της SQL είναι:

```
INSERT INTO όνομαΠίνακα (όνομαΠεδίου1, ..., όνομαΠεδίουN)  
VALUES (τιμή1, ..., τιμήN)
```

Το όνομαΠίνακα αντιστοιχεί στον πίνακα στον οποίο εισάγεται η εγγραφή και ακολουθείται από μία λίστα των ονομάτων πεδίων, διαχωρισμένων με κόμματα, σε παρένθεση. Η λίστα των ονομάτων πεδίων ακολουθείται από τη λέξη VALUES της SQL και μια λίστα τιμών, διαχωρισμένη με κόμματα, σε παρένθεση. Οι καθορισμένες τιμές σε αυτή τη λίστα πρέπει να ταιριάζουν με τα ονόματα πεδίων που παρατίθενται μετά το όνομα του πίνακα και ως προς τη σειρά, αλλά και ως προς τον τύπο (Deitel κ.α., 2002). Στο σχήμα 4.7 εμφανίζεται ένα ερώτημα INSERT, το οποίο εισάγει στον πίνακα Employee μια νέα εγγραφή, καθώς και ο πίνακας αυτός μετά την εκτέλεση του ερωτήματος. Όπως γίνεται εύκολα αντιληπτό, δεν καθορίζεται στο παράδειγμα αυτό μια τιμή για το πεδίο id του πίνακα Employee, διότι το πεδίο αυτό είναι ένα πεδίο αυτόματης αρίθμησης (auto increment) και επομένως η MySQL εκχωρεί μια μοναδική τιμή στο πεδίο αυτό αυτόματα. Στο συγκεκριμένο παράδειγμα στην εγγραφή που εκχωρείται δίνεται από τη MySQL για το πεδίο id του πίνακα, η τιμή 8.

1	INSERT INTO employee (name, surname, email, salary, jobFK)				
2	VALUES ("John", "Smith", "john@gmail.com", 1500, 2)				
id	name	surname	email	salary	jobFK
1	Aggeliki	Maragaki	maragaki@yahoo.gr	1000	1
2	Tom	Vallas	tom@hotmail.com	1200	3
3	George	Kontos	kontos@mred.tuc.gr	1900	2
4	Iwn	Kontiadis	iwn@yahoo.gr	1500	1
5	Gewrgia	Papadimitriou	g_papdim@yahoo.gr	900	4
6	Nikos	Mitritsakis	nik@hotmail.com	1400	2
7	Katerina	Mpounou	mpounou@mred.tuc	700	3
8	John	Smith	john@gmail.com	1500	2

Σχήμα 4.7 Παράδειγμα ερωτήματος INSERT και ο πίνακας Employee μετά την εκτέλεση του

4.6.6 Η φράση UPDATE

Η φράση UPDATE τροποποιεί τα δεδομένα σε έναν πίνακα. Η απλούστερη μορφή για μια φράση UPDATE είναι:

```
UPDATE όνομαΠίνακα  
SET όνομαΠεδίου1 = τιμή1, ..., όνομαΠεδίουN = τιμήN  
WHERE κριτήρια
```

Το όνομαΠίνακα αντιστοιχεί στον πίνακα στον οποίο ενημερώνεται μία ή περισσότερες εγγραφές. Ακολουθεί η λέξη της SQL, SET και μία λίστα , διαχωρισμένη με κόμματα, ζευγών ονομάτων πεδίων / τιμών, γραμμένων στη μορφή όνομαΠεδίου = τιμή. Η φράση WHERE καθορίζει τα κριτήρια που χρησιμοποιούνται για να υποδείξουν ποιες εγγραφές του πίνακα πρόκειται να ενημερωθούν. Στο σχήμα 4.8 εμφανίζεται ένα ερώτημα UPDATE με το οποίο ενημερώνεται η εγγραφή με id=13 του πίνακα Employee, καθώς και ο πίνακας αυτός μετά την εκτέλεση της ενημέρωσης. Γίνεται εύκολα αντιληπτό ότι το συγκεκριμένο ερώτημα του σχήματος 4.8, θα είχε ακριβώς το ίδιο αποτέλεσμα αν το κριτήριο id=8 αντικαθιστούνταν από το κριτήριο surname=Smith.

<pre>1 UPDATE employee 2 SET email="smith@yahoo.com", salary=1600 3 WHERE id=8</pre>					
id	name	surname	email	salary	jobFK
1	Aggeliki	Maragaki	maragaki@yahoo.gr	1000	1
2	Tom	Vallas	tom@hotmail.com	1200	3
3	George	Kontos	kontos@mred.tuc.gr	1900	2
4	Iwn	Kontiadis	iwn@yahoo.gr	1500	1
5	Gewrgia	Papadimitriou	g_papdim@yahoo.gr	900	4
6	Nikos	Mitritsakis	nik@hotmail.com	1400	2
7	Katerina	Mpounou	mpounou@mred.tuc.gr	700	3
8	John	Smith	smith@yahoo.com	1600	2

Σχήμα 4.8 Παράδειγμα ερωτήματος UPDATE και ο πίνακας Employee μετά την εκτέλεση του

4.6.7 Η φράση DELETE

Μια SQL φράση DELETE αφαιρεί δεδομένα από έναν πίνακα. Η απλούστερη μορφή για την φράση DELETE είναι:

```
DELETE FROM όνομαΠίνακα WHERE κριτήρια
```

Το όνομαΠίνακα αντιστοιχεί στον πίνακα από τον οποίο πρόκειται να διαγραφεί μία ή περισσότερες εγγραφές. Η φράση WHERE καθορίζει τα κριτήρια που χρησιμοποιούνται για να καθορίσουν ποιες εγγραφές πρόκειται να διαγραφούν. Το παράδειγμα SQL ερωτήματος DELETE που ακολουθεί, διαγράφει την τελευταία εγγραφή, με id=8, που προστέθηκε στον πίνακα Employee (σχήμα 4.7) και παράγει τον αρχικό πίνακα Employee:

```
DELETE FROM Employee WHERE id = 8
```

4.6.8 Η φράση LIMIT

Μια SQL φράση LIMIT καθορίζει τον αριθμό των δεδομένων του συνόλου των αποτελεσμάτων ενός ερωτήματος. Η απλούστερη μορφή της προαιρετικής αυτής φράσης είναι:

```
SELECT όνομαΠεδίου1, όνομαΠεδίου2,...  
FROM όνομαΠίνακα  
LIMIT αρχή, αριθμόςΑποτελεσμάτων
```

Η αρχή αντιστοιχεί στην πρώτη εγγραφή του συνόλου του αποτελέσματος του ερωτήματος, ενώ ο αριθμόςΑποτελεσμάτων αντιστοιχεί στο πλήθος των εγγραφών του αποτελέσματος του ερωτήματος. Ουσιαστικά η φράση LIMIT αποτελεί ένα εργαλείο για τη διαίρεση των αποτελεσμάτων ερωτημάτων, που εκτελούνται σε πίνακες με μεγάλο πλήθος εγγραφών, σε μικρότερα μέρη. Το παράδειγμα ερωτήματος του σχήματος 4.9 εμφανίζει τα ονόματα των τριών πιο υψηλόμισθων υπαλλήλων του πίνακα Employee καθώς και τις ειδικότητες τους. Αρχικά επιλέγονται τα κατάλληλα πεδία που θα εμφανίζονται και υλοποιείται η σχέση «ένα προς πολλά» μεταξύ των δύο πινάκων. Η φράση ORDER BY χρησιμοποιείται για την ταξινόμηση των αποτελεσμάτων με φθίνουσα σειρά (DESC). Η φράση LIMIT χρησιμοποιείται για να καθορίσει ότι το σύνολο του αποτελέσματος θα αποτελείται από τις 3 πρώτες εγγραφές του ερωτήματος.

1	SELECT	employee.name, employee.surname, employee.salary, jobs.name
2	FROM	employee, jobs
3	WHERE	employee.jobFK=jobs.id
4	ORDER BY	employee.salary DESC
5	LIMIT	0,3

name	surname	salary	name
George	Kontos	1900	Designer
John	Smith	1600	Designer
Iwn	Kontiadis	1500	Developer

Σχήμα 4.9: Παράδειγμα ερωτήματος με την SQL φράση LIMIT

4.6.9 Η φράση AS

Η φράση AS χρησιμοποιείται σε ένα ερώτημα ακριβώς μετά το όνομα ενός πεδίου ή ενός πίνακα για να μετονομάσει το πεδίο ή τον πίνακα αντίστοιχα. Η μετονομασία αυτή δεν επιδρά στα ονόματα των πεδίων ή των πινάκων της βάσης, αλλά γίνεται εσωτερικά σε ερωτήματα είτε για την καλύτερη παρουσίαση των αποτελεσμάτων στον απλό χρήστη (στην περίπτωση των πεδίων), είτε για χάρη συντομίας (στην περίπτωση των πινάκων). Στα αποτελέσματα του ερωτήματος του σχήματος 4.9 για παράδειγμα, το πεδίο της απασχόλησης εμφανίζεται με το όνομα name (όπως στην πραγματικότητα ονομάζεται το όνομα του πεδίου του πίνακα Jobs), ενώ θα ήταν ορθότερο και πιο κατανοητό ο τίτλος του πεδίου των αποτελεσμάτων να ήταν JobType ή Job. Στο σχήμα 4.10 εμφανίζεται το ίδιο ερώτημα με την προσθήκη φράσεων AS για την συντόμευση του και την καλύτερη παρουσίαση του αποτελέσματος του.

1	SELECT	e.name, e.surname, e.salary, j.name AS JobType
2	FROM	employee AS e, jobs AS j
3	WHERE	e.jobFK=j.id
4	ORDER BY	e.salary DESC
5	LIMIT	0,3

name	surname	salary	JobType
George	Kontos	1900	Designer
John	Smith	1600	Designer
Iwn	Kontiadis	1500	Developer

Σχήμα 4.10: Παράδειγμα ερωτήματος με φράση AS

4.6.10 Οι συναρτήσεις COUNT και SUM

Οι SQL συναρτήσεις COUNT και SUM χρησιμοποιούνται για την απαρίθμηση και την άθροιση αντίστοιχα, των εγγραφών ενός αποτελέσματος ερωτήματος. Η απλούστερη μορφή των συναρτήσεων αυτών είναι:

```
SELECT COUNT(όνομαΠεδίου) FROM όνομαΠίνακα  
SELECT SUM(όνομαΠεδίου) FROM όνομαΠίνακα
```

Το αποτέλεσμα των συναρτήσεων COUNT και SUM είναι ένας πίνακας που αποτελείται από ένα κελί και κατά συνέπεια μία τιμή (αριθμητική), η οποία παριστάνει είτε το πλήθος είτε το άθροισμα των εγγραφών του πεδίου αντίστοιχα. Στο σχήμα 4.11 εμφανίζονται παραδείγματα των συναρτήσεων αυτών καθώς και τα αντίστοιχα αποτελέσματα που παράγουν.

4.6.11 Οι συναρτήσεις MAX και MIN

Οι SQL συναρτήσεις MAX και MIN χρησιμοποιούνται για τον προσδιορισμό της μέγιστης και της ελάχιστης αντίστοιχα εγγραφής ενός αποτελέσματος ερωτήματος. Η απλούστερη μορφή των συναρτήσεων αυτών είναι:

```
SELECT MAX(όνομαΠεδίου) FROM όνομαΠίνακα  
SELECT MIN(όνομαΠεδίου) FROM όνομαΠίνακα
```

Το αποτέλεσμα των συναρτήσεων MAX και MIN είναι ένας πίνακας που αποτελείται από ένα κελί και κατά συνέπεια μία τιμή (όχι απαραίτητα αριθμητική), η οποία παριστάνει τη μέγιστη ή την ελάχιστη εγγραφή του πεδίου. Στο σχήμα 4.12 εμφανίζονται παραδείγματα των συναρτήσεων αυτών καθώς και τα αντίστοιχα αποτελέσματα που παράγουν.

1 SELECT COUNT(name)	1 SELECT SUM(salary)
2 FROM employee	2 FROM employee
3	3
count(name)	SUM(salary)
8	10200

Σχήμα 4.11: Παραδείγματα συναρτήσεων COUNT και SUM

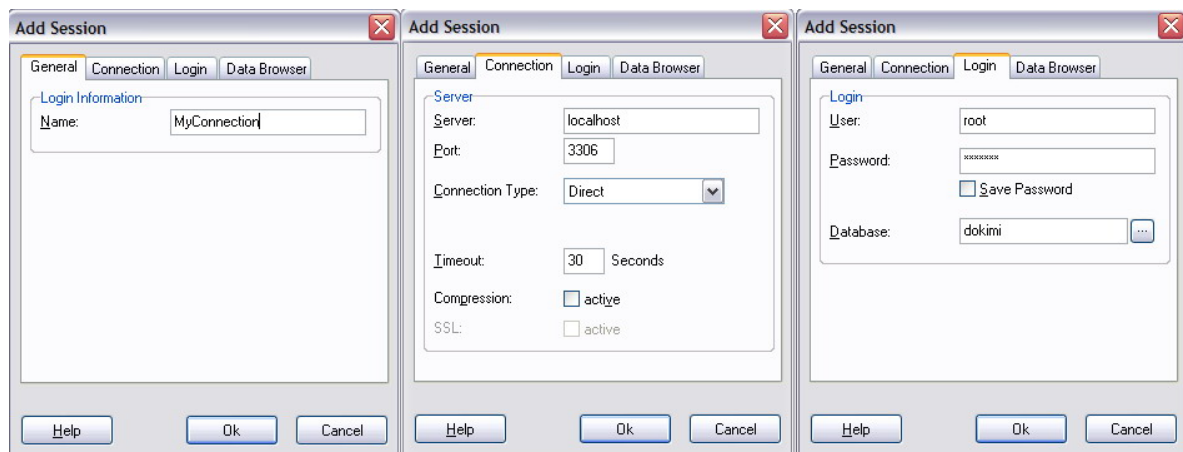
1	SELECT MAX(surname)	1	SELECT MIN(salary)
2	FROM employee	2	FROM employee
3		3	
MAX(surname)		MIN(salary)	
Vallas		700	

Σχήμα 4.12: Παραδείγματα συναρτήσεων MIN και MAX

4.7 Πρόγραμμα MySQL-Front

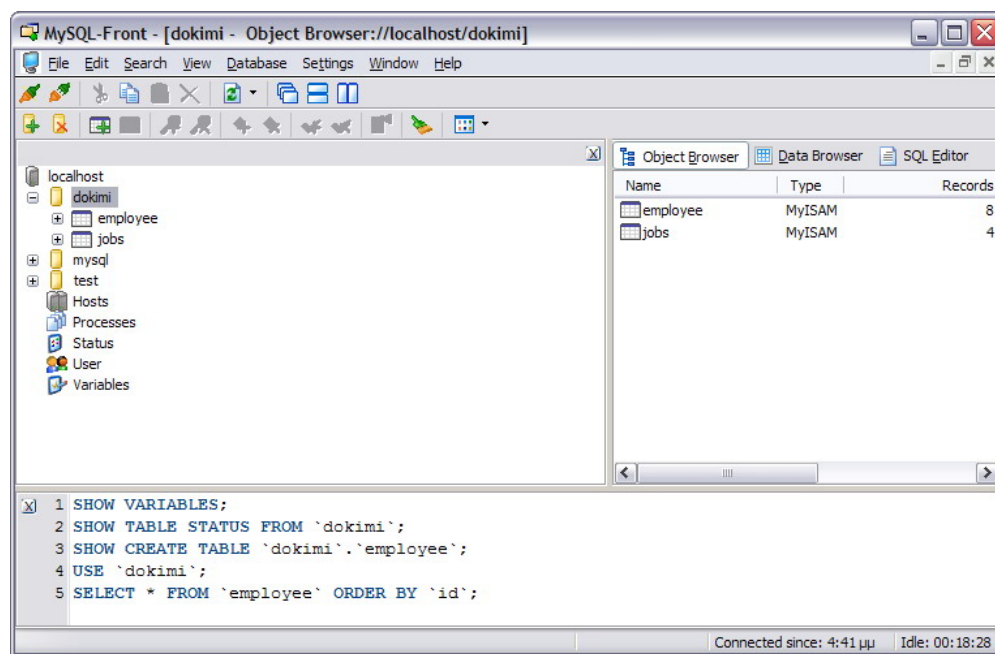
Το πρόγραμμα MySQL-Front είναι μία εφαρμογή τύπου πελάτη (client) του MySQL διακομιστή (server) που διευκολύνει μέσω απλού γραφικού περιβάλλοντος την ανάπτυξη και διαχείριση βάσεων δεδομένων. Τέτοιου τύπου εφαρμογές δίνουν τη δυνατότητα της ταχύτατης δημιουργίας και επεξεργασίας μιας βάσης δεδομένων, παράγοντας αυτοματοποιημένα τον SQL κώδικα που απαιτείται για τις διεργασίες αυτές. Η έκδοση του MySQL-Front που χρησιμοποιήθηκε κατά τη συγγραφή της παρούσας εργασίας είναι η 3.1, ενώ αξίζει να αναφερθεί ότι προτιμήθηκε από παρόμοιου τύπου εφαρμογές, όπως το MySQL-Query Browse (το επίσημο τέτοιου τύπου εργαλείο της MySQL AB) ή το Navicat, λόγω της απλότητας χρήσης του γραφικού του περιβάλλοντος.

Για να επικοινωνήσει μία εφαρμογή τύπου πελάτη με τον αντίστοιχο διακομιστή, θα πρέπει να επιτευχθεί αρχικά μια διασύνδεση μεταξύ τους. Κατά την εκκίνηση της εφαρμογής MySQL-Front η πρώτη απαραίτητη ενέργεια που ζητείται από τον χρήστη, είναι η υλοποίηση της σύνδεσης με τον διακομιστή (MySQL Server). Στο σχήμα 4.13 εμφανίζεται το παράθυρο διαλόγου του προγράμματος MySQL-Front, που χρησιμεύει στη δημιουργία νέας σύνδεσης με το διακομιστή. Εκτός από το όνομα της σύνδεσης, οι απαραίτητες πληροφορίες που ζητούνται στο διάλογο αυτό ώστε να δημιουργηθεί η σύνδεση με το διακομιστή, είναι το όνομα χρήστη και ο αντίστοιχος κωδικός ασφαλείας (username και password), η τοποθεσία του διακομιστή (location) δηλαδή το αν βρίσκεται στον ίδιο υπολογιστή με το MySQL-Front ή σε οποιονδήποτε άλλον υπολογιστή (δηλώνεται μέσω του μοναδικού αριθμού IP του υπολογιστή) και η βάση ή οι βάσεις δεδομένων του διακομιστή στις οποίες πρόκειται να γίνει η διασύνδεση. Επιπλέον ζητούνται και κάποιες προαιρετικές πληροφορίες, όπως ο τύπος της διασύνδεσης ή η θύρα (port) του υπολογιστή μέσω της οποίας υλοποιείται η διασύνδεση.



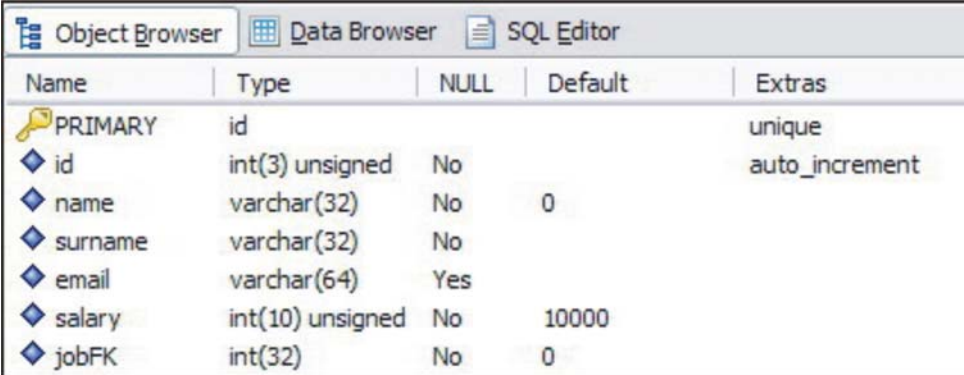
Σχήμα 4.13: Το παράθυρο διαλόγου του MySQL-Front για τη δημιουργία νέας σύνδεσης με το διακομιστή (MySQL Server)

Εφόσον επιτευχθεί η σύνδεση με το διακομιστή, το MySQL-Front εμφανίζει γραφικά τη βάση ή τις βάσεις δεδομένων που αναφέρθηκαν κατά τη σύνδεση. Στο σχήμα 4.14 εμφανίζεται το πρόγραμμα MySQL-Front, με το οποίο έχει γίνει διασύνδεση με τον αντίστοιχο διακομιστή και συγκεκριμένα με τη βάση δεδομένων με όνομα dokimi. Η βάση δεδομένων αυτή περιλαμβάνει τους πίνακες Employee και Jobs του παραδείγματος των προηγούμενων παραγράφων. Στο αριστερό τμήμα της εφαρμογής εμφανίζονται οι βάσεις δεδομένων καθώς και οι αντίστοιχοι πίνακες τους σε δένδρική δομή. Το δεξιό τμήμα του προγράμματος αποτελείται από τρεις επιλογές (tabs):



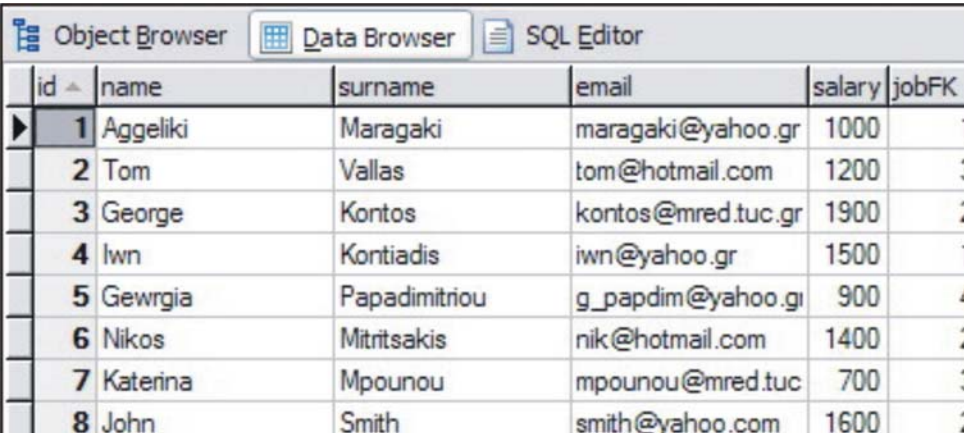
Σχήμα 4.14: Το πρόγραμμα MySQL-Front σε σύνδεση με τη βάση δεδομένων dokimi του διακομιστή

- Object Browser: Μέσω της επιλογής αυτής αναπαριστάται γραφικά και επεξεργάζεται η δομή των πινάκων μιας βάσης δεδομένων (4.15). Ο χρήστης είναι μέσω αυτής της επιλογής είναι δυνατόν να αντλήσει πληροφορίες σχετικά με τα πεδία που απαρτίζουν έναν πίνακα αλλά και να δημιουργήσει καινούρια πεδία.
- Data Browser: Μέσω της επιλογής αυτής εμφανίζονται γραφικά τα δεδομένα (οι εγγραφές) ενός πίνακα (4.16). Ο χρήστης μέσω αυτής της επιλογής είναι δυνατόν να προσθέσει, να επεξεργαστεί ή να διαγράψει εγγραφές σε έναν πίνακα.
- SQL Editor: Μέσω της επιλογής αυτής ο χρήστης μπορεί να υποβάλει ερωτήματα SQL στη βάση δεδομένων και να παίρνει τα αποτελέσματα των ερωτημάτων αυτών σε γραφική μορφή (4.17).



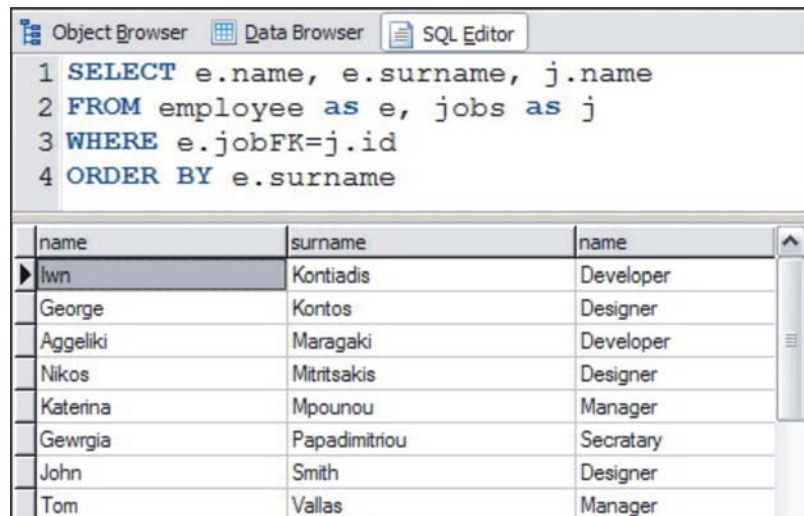
Name	Type	NULL	Default	Extras
PRIMARY	id			unique
id	int(3) unsigned	No		auto_increment
name	varchar(32)	No	0	
surname	varchar(32)	No		
email	varchar(64)	Yes		
salary	int(10) unsigned	No	10000	
jobFK	int(32)	No	0	

Σχήμα 4.15: Απεικόνιση της δομής του πίνακα Employee μέσω της επιλογής Object Browser του προγράμματος MySQL-Front



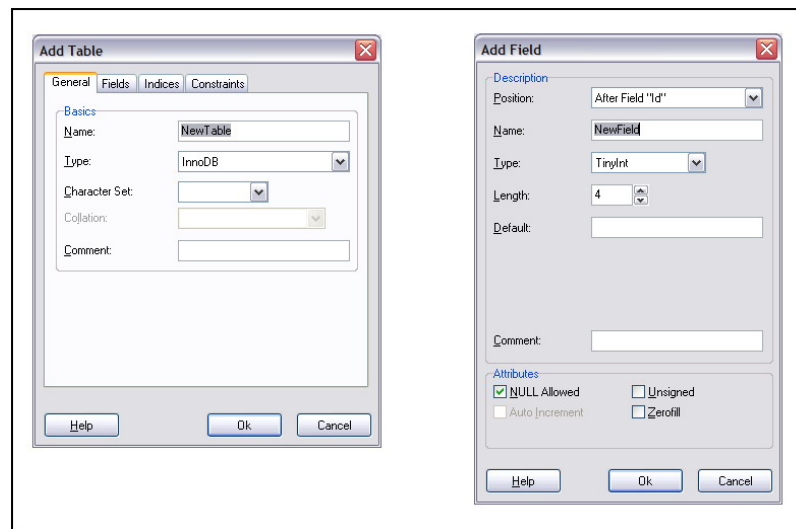
	id	name	surname	email	salary	jobFK
1	1	Aggeliki	Maragaki	maragaki@yahoo.gr	1000	1
2	2	Tom	Vallas	tom@hotmail.com	1200	3
3	3	George	Kontos	kontos@mred.tuc.gr	1900	2
4	4	Iwn	Kontiadis	iwn@yahoo.gr	1500	1
5	5	Gewrgia	Papadimitriou	g_papdim@yahoo.gr	900	4
6	6	Nikos	Mitritsakis	nik@hotmail.com	1400	2
7	7	Katerina	Mpounou	mpounou@mred.tuc	700	3
8	8	John	Smith	smith@yahoo.com	1600	2

Σχήμα 4.16: Απεικόνιση των δεδομένων του πίνακα Employee μέσω της επιλογής Data Browser του προγράμματος MySQL-Front



Σχήμα 4.17: Υποβολή ερωτήματος και απεικόνιση των αποτελεσμάτων του μέσω της επιλογής SQL Editor του προγράμματος MySQL-Front

Στο κάτω τμήμα της εφαρμογής εμφανίζεται ο SQL κώδικας που παράγεται με κάθε κίνηση που πραγματοποιεί ο χρήστης, χαρακτηριστικό το οποίο αποτελεί σημαντική αρωγή για την εξοικείωση με την SQL. Στο σχήμα 4.18 εμφανίζονται οι διάλογοι του προγράμματος MySQL-Front για την δημιουργία νέων πινάκων και νέων πεδίων. Κατά τη δημιουργία ενός πίνακα, εκτός από το όνομα του και τα πεδία του, ο χρήστης θα πρέπει να καθορίσει τον τύπο του πίνακα, δηλαδή τον τρόπο με τον οποίο ο MySQL διακομιστής θα αποθηκεύει τα δεδομένα. Οι πίνακες που χρησιμοποιήθηκαν κατά τη συγγραφή της παρούσας εργασίας είναι τύπου MyISAM, ενώ ένας άλλος ευρεία χρησιμοποιούμενος τύπος πίνακα του MySQL διακομιστή είναι ο τύπος InnoDB.



Σχήμα 4.18: Οι διάλογοι για δημιουργία νέων πινάκων και πεδίων του προγράμματος MySQL-Front

4.8 Βάση δεδομένων rmlab

Στην ενότητα αυτή περιγράφεται η δομή της βάσης δεδομένων rmlab που κατασκευάστηκε στο σύστημα βάσεων δεδομένων MySQL για τις ανάγκες τις παρούσας εργασίας. Το όνομα της βάσης προκύπτει από τα αρχικά των λέξεων rock mechanics lab (εργαστήριο μηχανικής πετρωμάτων) και ο σκοπός της είναι να αποθηκεύει και να ομαδοποιεί μια πληθώρα δεδομένων και παραμέτρων που σχετίζονται με τα πέντε συνηθέστερα εργαστηριακά πειράματα της μηχανικής πετρωμάτων που περιγράφονται στο κεφάλαιο 3 και εν συντομία είναι: μονοαξονική θλίψη (uniaxial), κάμψη (bending), τριαξονική θλίψη (triaxial), εφελκυσμός (tension) και υπέρηχοι (ultrasonic).

Οι παράμετροι των πειραμάτων που πρόκειται να αποθηκεύονται στη βάση δεδομένων rmlab είναι δυνατόν να διακριθούν σε δύο κατηγορίες: τις κοινές παραμέτρους ανεξαρτήτως τύπου πειράματος και τις μη κοινές παραμέτρους οι οποίες ομαδοποιούνται περαιτέρω σε πέντε κατηγορίες ανάλογα με τον τύπο του πειράματος. Η δομή της βάσης δεδομένων rmlab επομένως περιλαμβάνει έναν κεντρικό πίνακα που ονομάζεται master και έχει για πεδία του τις κοινές παραμέτρους όλων των πειραμάτων καθώς και άλλους πέντε πίνακες που έχουν για πεδία τους τις παραμέτρους του εκάστοτε πειράματος. Οι πέντε αυτοί πίνακες ονομάζονται uniaxial, bending, triaxial, tension και ultrasonic και καθένας τους συνδέεται με σχέση ένα προς πολλά με τον κεντρικό πίνακα master.

Για τις περισσότερες από τις κοινές παραμέτρους των πειραμάτων κρίθηκε σκόπιμη η αποθήκευση τους σε ξεχωριστούς πίνακες ώστε να αποφευχθεί η επανάληψη στα δεδομένα του πίνακα master. Για παράδειγμα μια παράμετρος που πρόκειται να αποθηκεύει η βάση δεδομένων rmlab ανεξάρτητα με τον τύπο του πειράματος είναι το όνομα του ατόμου που εκτέλεσε το πείραμα. Για να αποφευχθεί η επανάληψη μεγάλων σε μήκος ονομάτων στον πίνακα master (εφόσον κάθε άτομο μπορεί να εκτελέσει παραπάνω από ένα πειράματα) κρίνεται σκόπιμη η δημιουργία ενός πίνακα με όνομα contactperson ο οποίος θα αποτελείται από τα πεδία id (αύξων αριθμός) και name (όνομα ατόμου). Ο πίνακας αυτός θα πρέπει να συνδεθεί με σχέση ένα προς πολλά με τον πίνακα master στον οποίο κατασκευάζεται το αντίστοιχο πεδίο (ξένο κλειδί) contactpersonsFK.

Οι μικροί πίνακες αυτού του είδους για λόγους απλότητας αποκαλούνται υπό-πίνακες (sub tables). Εκτός από τον υπό-πίνακα contactpersons κατασκευάστηκαν άλλοι 13 υπό-πίνακες καθένας από τους οποίους συνδέεται με σχέση ένα προς πολλά με το

























αντίστοιχο πεδίο του πίνακα master και αντιπροσωπεύει μία από τις κοινές παραμέτρους των πειραμάτων. Οι υπό-πίνακες της βάσης δεδομένων rmlab είναι οι:

- projects - αποθηκεύει τα ονόματα των project στα πλαίσια των πειραμάτων
- groups – αποθηκεύει τα ονόματα των ομάδων που ανήκουν τα πειράματα
- contactperson – αποθηκεύει τα ονόματα των ατόμων που διεξήγαγαν τα πειράματα
- places – αποθηκεύει τα τοπωνύμια από τα οποία προέρχονται τα δοκίμια
- materials – αποθηκεύει τα ονόματα των υλικών των δοκιμών
- forms – αποθηκεύει τα ονόματα των σχημάτων που έχουν τα δοκίμια
- discontinuities – αποθηκεύει το πλήθος των ασυνεχειών των δοκιμών
- inclusions – αποθηκεύει το πλήθος των εγκλεισμάτων των δοκιμών
- porosity – αποθηκεύει τους τύπους του πορώδους των δοκιμών
- preparation – αποθηκεύει τις διαβαθμίσεις της προετοιμασίας των δοκιμών
- confidence – αποθηκεύει τις διαβαθμίσεις της εμπιστοσύνης του πειράματος
- failure – αποθηκεύει τα είδη της αστοχίας που υφίστανται τα δοκίμια
- loading – αποθηκεύει τα είδη της καταπόνησης που υφίστανται τα δοκίμια
- tests – αποθηκεύει τα είδη των πειραμάτων

4.8.1 Πίνακας master

Στο σχήμα 4.19 εμφανίζεται η δομή του κεντρικού πίνακα master της βάσης δεδομένων rmlab, δηλαδή τα πεδία που διαθέτει, καθώς και ο τύπος δεδομένων που δέχεται το κάθε πεδίο. Εκτός από τα πεδία – ξένα κλειδιά που συνδέονται με σχέση ένα προς πολλά με τους υπό-πίνακες που προαναφέρθηκαν (τα ονόματά τους έχουν το όνομα του αντίστοιχου υπό-πίνακα με την κατάληξη FK) ο πίνακας master διαθέτει και τα εξής πεδία:

- date: αποθηκεύει την ημερομηνία διεξαγωγής του εκάστοτε πειράματος
- axialDim: αποθηκεύει την κατακόρυφη διάσταση του εκάστοτε δοκιμίου (mm)
- transverseDim1 και transverseDim2: αποθηκεύουν τις οριζόντιες διαστάσεις του εκάστοτε δοκιμίου (mm)
- density: αποθηκεύει την πυκνότητα του εκάστοτε δοκιμίου (gr/cm^3)
- filename: αποθηκεύει το όνομα του αρχείου που σχετίζεται με το εκάστοτε πείραμα (π.χ ένα αρχείο excel με γραφικές παραστάσεις ή ένα αρχείο εικόνας)
- comments: αποθηκεύει πιθανόν σχόλια και παρατηρήσεις για το εκάστοτε πείραμα

Name	Type	NULL
 PRIMARY	Id	
 Id	int(6) unsigned	No
 projectFK	tinyint(4) unsigned	No
 groupFK	tinyint(4) unsigned	No
 contactpersonFK	tinyint(4) unsigned	No
 date	date	No
 placeFK	tinyint(4) unsigned	No
 materialFK	tinyint(4) unsigned	No
 formFK	tinyint(4) unsigned	No
 axialdim	float unsigned	No
 traversedim1	float unsigned	No
 traversedim2	float unsigned	Yes
 density	float unsigned	No
 discontinuitiesFK	tinyint(4) unsigned	No
 inclusionsFK	tinyint(4) unsigned	No
 porosityFK	tinyint(4) unsigned	No
 preparationFK	tinyint(4) unsigned	No
 confidenceFK	tinyint(4) unsigned	No
 failureFK	tinyint(4) unsigned	No
 filename	varchar(100)	Yes
 testFK	tinyint(4) unsigned	No
 childFK	int(6) unsigned	No
 loadingFK	tinyint(4) unsigned	No
 comments	text	Yes












Σχήμα 4.19: Η δομή του πίνακα master της βάσης δεδομένων rmlab

Τα πεδία transversedim2, filename και comments του πίνακα master δεν είναι υποχρεωτικά, δηλαδή επιτρέπεται να μείνουν κενά, ενώ όλα τα υπόλοιπα πεδία του εν λόγω πίνακα είναι υποχρεωτικά. Το πρωτεύον κλειδί του πίνακα master είναι το πεδίο id (ακέραιος, αυτόματα αυξανόμενος αριθμός). Το πεδίο childFK σχεδιάστηκε για να συνδέει τον πίνακα master με τους πίνακες των επιμέρους πειραμάτων.

4.8.2 Πίνακας uniaxial

Ο πίνακας uniaxial είναι υπεύθυνος για την αποθήκευση παραμέτρων δοκιμών προσδιορισμού της αντοχής σε μονοαξονική θλίψη. Στο σχήμα 4.20 εμφανίζεται η δομή του πίνακα uniaxial. Το πρωτεύον κλειδί του πίνακα αποτελεί το πεδίο id (ακέραιος, αυτόματα αυξανόμενος αριθμός). Τα υπόλοιπα πεδία που συνθέτουν τον πίνακα uniaxial είναι:

- UCS: αποθηκεύει την αντοχή σε μονοαξονική θλίψη (Mpa)
- adjustedUCS: αποθηκεύει τη διορθωμένη αντοχή σε μονοαξονική θλίψη (Mpa)

Name	Type	NULL
 PRIMARY	Id	
 Id	int(6) unsigned	No
 UCS	double unsigned	No
 adjustedUCS	double unsigned	Yes
 averagemodulus	int(7) unsigned	No
 tangentmodulus	int(7) unsigned	Yes
 secantmodulus	int(7) unsigned	Yes
 poissonratio	double unsigned	Yes
 maxdeformation	double unsigned	Yes
 energytofailure	double unsigned	Yes
 duration	int(5) unsigned	Yes






Σχήμα 4.20: Η δομή του πίνακα uniaxial της βάσης δεδομένων rmlab

- averagemodulus: αποθηκεύει το μέσο μέτρο ελαστικότητας (Gpa)
- tangentmodulus: αποθηκεύει το εφαπτομενικό μέτρο ελαστικότητας (Gpa)
- secantmodulus: αποθηκεύει το τέμνον μέτρο ελαστικότητας (Gpa)
- poissonratio: αποθηκεύει το λόγο του Poisson
- maxdeformation: αποθηκεύει τη μέγιστη παραμόρφωση (mm)
- energytofailure: αποθηκεύει το ποσό της ενέργειας που απαιτείται για αστοχία του δοκιμίου (Joule)
- duration: αποθηκεύει τη χρονική διάρκεια του πειράματος (sec)

4.8.3 Πίνακας bending

Ο πίνακας bending είναι υπεύθυνος για την αποθήκευση παραμέτρων των δοκιμών προσδιορισμού της αντοχής σε κάμψη. Στο σχήμα 4.21 εμφανίζεται η δομή του πίνακα bending. Το πρωτεύον κλειδί του πίνακα αποτελεί το πεδίο id (ακέραιος, αυτόματα αυξανόμενος αριθμός). Τα υπόλοιπα πεδία που συνθέτουν τον πίνακα bending είναι:

- bendingstrength: αποθηκεύει την αντοχή σε κάμψη (Mpa)
- maxdeformation: αποθηκεύει τη μέγιστη παραμόρφωση (mm)
- duration: αποθηκεύει τη χρονική διάρκεια του πειράματος (sec)













Name	Type	NULL
 PRIMARY	Id	
 Id	int(6) unsigned	No
 bendingstrength	float unsigned	No
 maxdeformation	double unsigned	Yes
 duration	int(5) unsigned	Yes

Σχήμα 4.21: Η δομή του πίνακα bending της βάσης δεδομένων rmlab

4.8.4 Πίνακας triaxial

Ο πίνακας triaxial είναι υπεύθυνος για την αποθήκευση παραμέτρων των δοκιμών προσδιορισμού της αντοχής σε τριαξονική καταπόνηση. Στο σχήμα 4.22 εμφανίζεται η δομή του πίνακα triaxial. Το πρωτεύον κλειδί του πίνακα αποτελεί το πεδίο id (ακέραιος, αυτόματα αυξανόμενος αριθμός). Τα υπόλοιπα πεδία του πίνακα triaxial είναι:

- maximumstress: αποθηκεύει τη μέγιστη τάση (Mpa)
- lateralstress: αποθηκεύει την κατακόρυφη στον άξονα φόρτισης τάση (Mpa)
- averagemodulus: αποθηκεύει το μέσο μέτρο ελαστικότητας (Gpa)
- tangentmodulus: αποθηκεύει το εφαπτομενικό μέτρο ελαστικότητας (Gpa)
- secantmodulus: αποθηκεύει το τέμνον μέτρο ελαστικότητας (Gpa)
- maxdeformation: αποθηκεύει τη μέγιστη παραμόρφωση (mm)
- energytofailure: αποθηκεύει το ποσό της ενέργειας που απαιτείται για αστοχία (Joule)
- cohesion: αποθηκεύει τη συνοχή (kPa)
- intervalfriction: αποθηκεύει τη γωνία εσωτερικής τριβής (deg)
- duration: αποθηκεύει τη χρονική διάρκεια του πειράματος (sec)

Name	Type	NULL
 PRIMARY	Id	
 Id	int(6) unsigned	No
 maximumstress	double unsigned	No
 lateralstress	double unsigned	Yes
 averagemodulus	int(7) unsigned	No
 tangentmodulus	int(7) unsigned	Yes
 secantmodulus	int(7) unsigned	Yes
 maxdeformation	double unsigned	Yes
 energytofailure	double unsigned	Yes
 cohesion	double unsigned	Yes
 intervalfriction	int(4) unsigned	Yes
 duration	int(5) unsigned	Yes

Σχήμα 4.22: Η δομή του πίνακα triaxial της βάσης δεδομένων rmlab

4.8.5 Πίνακας tension






Ο πίνακας tension είναι υπεύθυνος για την αποθήκευση παραμέτρων των δοκιμών έμμεσου προσδιορισμού της αντοχής σε εφελκυσμό. Στο σχήμα 4.23 εμφανίζεται η δομή του πίνακα tension. Το πρωτεύον κλειδί του πίνακα αποτελεί το πεδίο id (ακέραιος, αυτόματα αυξανόμενος αριθμός). Τα υπόλοιπα πεδία του πίνακα tension είναι:

- tensilestrength: αποθηκεύει την αντοχή σε εφελκυσμό (Mpa)
- energytofailure: αποθηκεύει το ποσό της ενέργειας που απαιτείται για αστοχία (Joule)
- duration: αποθηκεύει τη χρονική διάρκεια του πειράματος (sec)







4.8.6 Πίνακας ultrasonic

Ο πίνακας ultrasonic είναι υπεύθυνος για την αποθήκευση παραμέτρων των δοκιμών προσδιορισμού των ταχυτήτων διάδοσης θλιπτικών και διατμητικών κυμάτων. Στο σχήμα 4.24 εμφανίζεται η δομή του πίνακα ultrasonic. Το πρωτεύον κλειδί του πίνακα αποτελεί το πεδίο id (ακέραιος, αυτόματα αυξανόμενος αριθμός). Τα υπόλοιπα πεδία του πίνακα ultrasonic είναι:

- vp: αποθηκεύει τη ταχύτητα διάδοσης των θλιπτικών κυμάτων (m/sec)
- vs: αποθηκεύει τη ταχύτητα διάδοσης των διατμητικών κυμάτων (m/sec)
- edynamic: αποθηκεύει το μέτρο ελαστικότητας (Gpa)
- poissonratio: αποθηκεύει το λόγο του Poisson

Name	Type	NULL
 PRIMARY	Id	
 Id	int(6) unsigned	No
 tensilestrength	double unsigned	No
 energytofailure	double unsigned	Yes
 duration	int(5) unsigned	Yes

Σχήμα 4.23: Η δομή του πίνακα tension της βάσης δεδομένων rmlab

Name	Type	NULL
 PRIMARY	Id	
 Id	int(6) unsigned	No
 vp	double unsigned	No
 vs	double unsigned	No
 edynamic	int(7) unsigned	Yes
 poissonratio	double unsigned	Yes

Σχήμα 4.24: Η δομή του πίνακα ultrasonic της βάσης δεδομένων rmlab

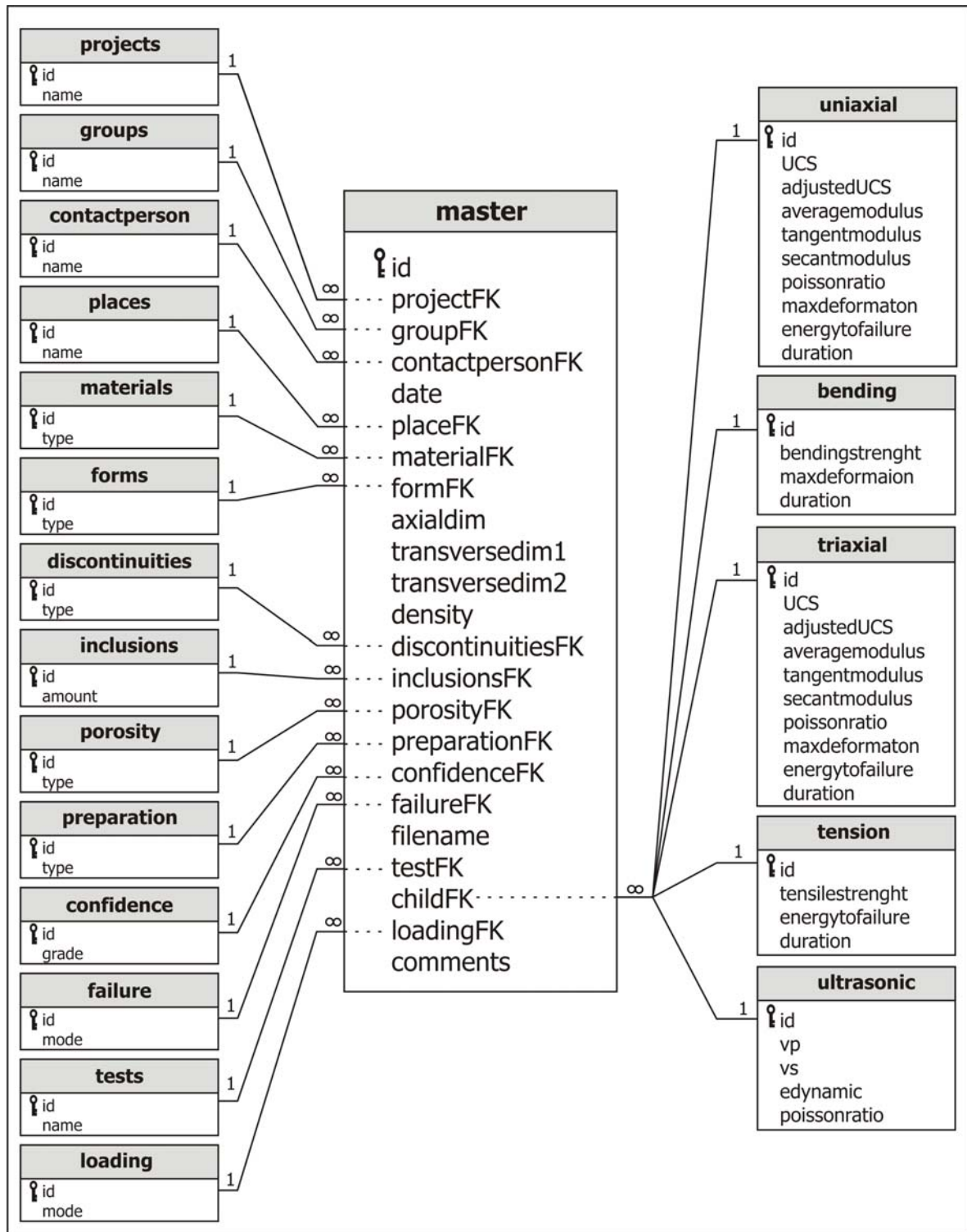
4.8.7 Δομή και σχέσεις

Στο σχήμα 4.25 εμφανίζεται η συνολική δομή της βάσης δεδομένων rmlab. Η βάση δεδομένων αυτή αποτελείται από έξι κύριους πίνακες (master, uniaxial, bending, triaxial, tension και ultrasonic) και 14 υπό-πίνακες οι οποίοι συνδέονται με σχέση ένα προς πολλά με τα αντίστοιχα πεδία του πίνακα master.

Κάθε εγγραφή της βάσης δεδομένων αποθηκεύεται σε δύο πίνακες: στον πίνακα master και σε έναν από τους πίνακες των επιμέρους πειραμάτων, ανάλογα με τον τύπο της δοκιμής. Επομένως κάθε εγγραφή χαρακτηρίζεται από δύο μοναδικούς ακέραιους αριθμούς (id) οι οποίοι είναι το id της εγγραφής στον πίνακα master και το id της εγγραφής στον επιμέρους πίνακα της δοκιμής. Ο πίνακας master επομένως θα πρέπει να διαθέτει δύο πληροφορίες για κάθε εγγραφή ώστε να επιτευχθεί η ένωση του με κάποιον από τους επιμέρους πίνακες των δοκιμών και να γίνει δυνατή η ταυτόχρονη ανάγνωση όλων των παραμέτρων που συνθέτουν μια εγγραφή. Η πρώτη πληροφορία είναι ο τύπος της δοκιμής στην οποία ανήκει η εγγραφή, δηλαδή ένας από τους πέντε πίνακες με τον οποίο πρόκειται να επιτευχθεί η ένωση. Η πληροφορία αυτή αποθηκεύεται στο πεδίο testFK του πίνακα master. Η δεύτερη πληροφορία είναι ο κωδικός (id) που έχει η συγκεκριμένη εγγραφή στον επιμέρους πίνακα του πειράματος στο οποίο ανήκει. Η πληροφορία αυτή αποθηκεύεται στο πεδίο childFK του πίνακα master.

Αν υποθεθεί ότι η πρώτη εγγραφή της βάσης δεδομένων rmlab είναι τα αποτελέσματα και οι παράμετροι μιας δοκιμής μονοαξονικής θλίψης (uniaxial), τότε η εγγραφή αυτή θα έχει ως id στον πίνακα master τον αριθμό 1 (εφόσον είναι η πρώτη εγγραφή της βάσης δεδομένων) και ως id στον πίνακα uniaxial τον αριθμό 1 (εφόσον είναι η πρώτη δοκιμή μονοαξονικής θλίψης που αποθηκεύεται). Επομένως στον πίνακα master τα πεδία id, testFK και childFK θα έχουν τις τιμές 1,1 (ο κωδικός που αντιστοιχεί στον uniaxial πίνακα) και 1 αντίστοιχα.

Αν υποθεθεί ότι η δεύτερη εγγραφή της βάσης δεδομένων rmlab είναι τα αποτελέσματα και οι παράμετροι μιας δοκιμής κάμψης (bending), τότε η εγγραφή αυτή θα έχει ως id στον πίνακα master τον αριθμό 2 (εφόσον είναι η δεύτερη εγγραφή της βάσης δεδομένων) και ως id στον πίνακα bending τον αριθμό 1 (εφόσον είναι η πρώτη δοκιμή κάμψης που αποθηκεύεται). Επομένως στον πίνακα master τα πεδία id, testFK και childFK θα έχουν τις τιμές 2,2 (ο κωδικός που αντιστοιχεί στον bending πίνακα) και 1 αντίστοιχα.



Σχήμα 4.25: Η δομή και η σχέσεις της βάσης δεδομένων rmlab

ΚΕΦΑΛΑΙΟ 5

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΔΙΑΧΕΙΡΙΣΗ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ ΤΗΣ ΜΗΧΑΝΙΚΗΣ ΠΕΤΡΩΜΑΤΩΝ

5.1 Εισαγωγή

Στο κεφάλαιο αυτό αρχικά περιγράφεται ο τρόπος με τον οποίο υλοποιείται η διασύνδεση της γλώσσας προγραμματισμού Visual Basic .NET με το σύστημα βάσεων δεδομένων MySQL. Εν συνεχεία περιγράφεται αναλυτικά το λογισμικό RockLab που αναπτύχθηκε για τη διαχείριση της βάσης δεδομένων rmlab. Ειδικότερα περιγράφεται ο τρόπος κατασκευής του εν λόγω λογισμικού αλλά και ο τρόπος χρήσης του.

5.2 Διασύνδεση Visual Basic .NET με το σύστημα MySQL

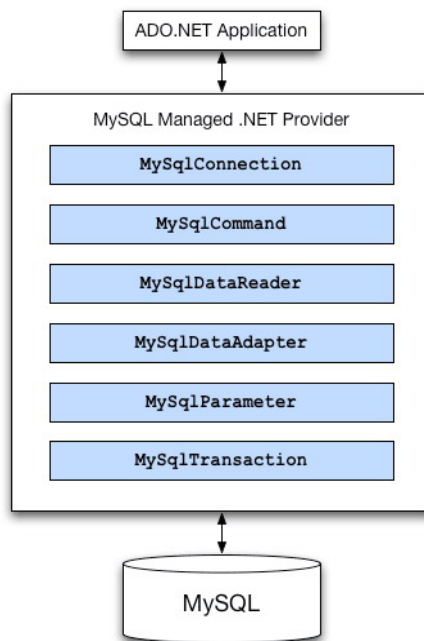
Για τη διασύνδεση οποιασδήποτε εφαρμογής, που έχει αναπτυχθεί σε γλώσσα προγραμματισμού Visual Basic .NET, με τον διακομιστή MySQL, είναι απαραίτητη η ύπαρξη κάποιου συνδέσμου (connector) ή κάποιου οδηγού (driver). Τα αντικείμενα αυτά είναι υπεύθυνα για την ομαλή επικοινωνία μεταξύ των δύο ετερογενών συστημάτων. Τα αντικείμενα αυτά μπορεί να είναι είτε μικρά προγράμματα είτε συλλογές από κλάσεις εξωτερικών αντικειμένων της Visual Basic .NET. Σήμερα υπάρχουν τρεις διαφορετικές λύσεις ώστε να μπορούν οι προγραμματιστές να έχουν πρόσβαση στον διακομιστή MySQL από το περιβάλλον της Visual Basic .NET:

- Το πρόγραμμα (οδηγός) ODBC.NET: Το ODBC (Open Database Connectivity) παρέχει έναν τρόπο για την πρόσβαση σε ευρύ φάσμα βάσεων δεδομένων. Αναπτύχθηκε σύμφωνα με τις προδιαγραφές της SQL και παρέχει ένα σύνολο συναρτήσεων, κωδικών λαθών και τύπων δεδομένων που είναι δυνατόν να χρησιμοποιηθούν για την ανάπτυξη μιας εφαρμογής. Το ODBC συνήθως χρησιμοποιείται όταν απαιτείται η ταυτόχρονη πρόσβαση σε διαφορετικές πηγές στοιχείων.
- Ο σύνδεσμος MySQL Connector/.NET: Παρέχει τη δυνατότητα στους προγραμματιστές να δημιουργήσουν εύκολα εφαρμογές στη Visual Basic .NET που απαιτούν ασφαλή, και υψηλής απόδοσης διασύνδεση δεδομένων με το σύστημα MySQL. Είναι γραμμένο σε γλώσσα προγραμματισμού C# και χαρακτηρίζεται από την μεγάλη του ταχύτητα.

- Ο σύνδεσμος OLEDB.NET: Λειτουργεί παρόμοια με το πρόγραμμα ODBC.NET, αλλά δεν υποστηρίζεται επίσημα από την εταιρεία MySQL AB.

Από τις τρεις αυτές διαφορετικές λύσεις επιλέχτηκε για τη διασύνδεση της υπό ανάπτυξη εφαρμογής με τον διακομιστή MySQL η δεύτερη, δηλαδή ο σύνδεσμος MySQL Connector/NET, κυρίως λόγω της απλότητας, της ταχύτητας και της αξιοπιστίας που τον διακρίνει. Στο σχήμα 5.1 εμφανίζεται η βασική αρχιτεκτονική του συνδέσμου. Μερικές από της κυριότερες κλάσεις αντικειμένων που παρέχονται από τον σύνδεσμο Connector/NET και χρησιμοποιήθηκαν κατά την ανάπτυξη του λογισμικού για τις ανάγκες της παρούσας εργασίας, είναι:

- MySqlConnection: τα αντικείμενα αυτής της κλάσης είναι υπεύθυνα για τη διασύνδεση με το διακομιστή MySQL
- MySqlCommand: δίνει τη δυνατότητα εκτέλεσης οποιασδήποτε εντολής (π.χ. ερώτημα) στη βάση δεδομένων
- MySqlDataReader: δίνει τη δυνατότητα γρήγορης ανάγνωσης δεδομένων από τη βάση
- MySqlDataAdapter: λειτουργεί ως διασυνδετική διάταξη μεταξύ της κλάσης δεδομένων της MySQL και της κλάσης DataTable της Visual Basic .NET
- MySqlParameter: χρησιμοποιείται για να αποθηκεύει δυναμικές παραμέτρους για μια εντολή



Σχήμα 5.1: Η βασική αρχιτεκτονική του συνδέσμου Connector/NET

5.3 Φόρμα σύνδεσης με τη βάση δεδομένων rmlab

Η αρχική φόρμα του προγράμματος RockLab, δηλαδή η φόρμα εκείνη που εμφανίζεται με την εκτέλεση της εφαρμογής είναι η φόρμα σύνδεσης (login) με τον MySQL διακομιστή. Για να επιτευχθεί η σύνδεση αυτή είναι απαραίτητη η δημιουργία ενός αντικειμένου από την κλάση MySqlConnection. Το αντικείμενο αυτό λειτουργεί ως ενδιάμεσος κρίκος μεταξύ των υπόλοιπων αντικειμένων που περιλαμβάνονται στον Connector/NET και του διακομιστή MySQL. Ο ρόλος του είναι να χειρίζεται τη διαδικασία σύνδεσης και μέσω αυτού είναι δυνατόν να ελεγχθούν οι πληροφορίες σύνδεσης ενός χρήστη.

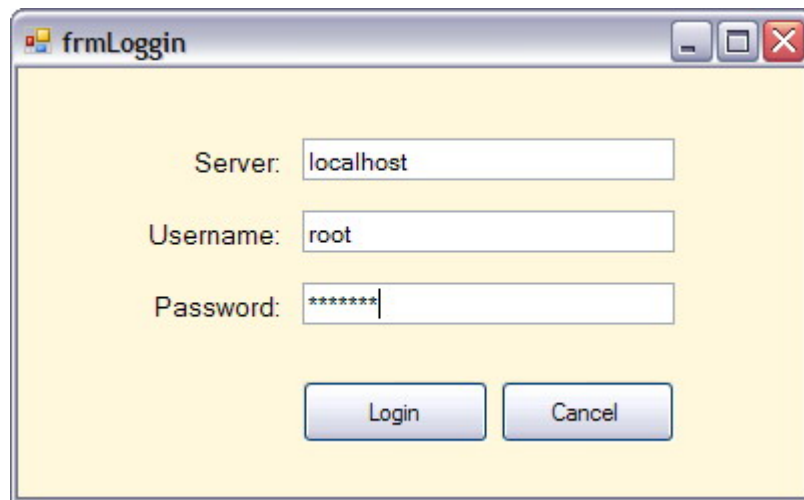
Το αντικείμενο τύπου MySqlConnection χρησιμοποιεί μια ακολουθία χαρακτήρων (string) για να οριστεί η πληροφορία σχετικά με την τοποθεσία του διακομιστή στον οποίο πρόκειται να συνδεθεί, το όνομα της βάσης δεδομένων στην οποία θα αποκτήσει πρόσβαση και το όνομα καθώς και τον κωδικό χρήστη με τα οποία θα επιτευχθεί η σύνδεση. Η τυπική μορφή μιας τέτοιας ακολουθίας χαρακτήρων σύνδεσης (connection string) είναι:

```
server=localhost; user id=tv; password=12345; database=in_out
```

Στο σχήμα 5.2 εμφανίζεται η φόρμα του προγράμματος όπου υλοποιείται η σύνδεση με το διακομιστή MySQL. Η φόρμα αποτελείται από τρία πεδία εισαγωγής δεδομένων (txtServer, txtUsername και mtxtPassword), όπου ο χρήστης καλείται να εισάγει την τοποθεσία του διακομιστή το όνομα χρήστη και τον κωδικό χρήστη αντίστοιχα. Στο πεδίο εισαγωγής κωδικού χρήστη έχει εφαρμοστεί κατάλληλη μάσκα ώστε να μην εμφανίζεται ο κωδικός, αλλά χαρακτήρες αστερίσκου όπως συνηθίζεται σε πεδία τέτοιου τύπου. Στη φόρμα κατασκευάστηκαν επίσης δύο κουμπιά (command buttons) των οποίων τα ονόματα είναι btnCancel και btnLogin. Ο σκοπός του πρώτου είναι να κλείνει την εφαρμογή ενώ του δεύτερου να υλοποιεί τη σύνδεση με βάση τα στοιχεία που εισάγει ο χρήστης και να εμφανίζει την κύρια φόρμα του προγράμματος εφόσον τα στοιχεία αυτά είναι αποδεκτά.

Στο σχήμα 5.3 εμφανίζεται ο κώδικας που εκτελείται όταν ο χρήστης πατήσει το κουμπί btnLogin. Αρχικά δημιουργείται ένα αντικείμενο τύπου MySqlConnection το οποίο ονομάζεται conn και στη συνέχεια ορίζεται η ακολουθία χαρακτήρων σύνδεσης από τα δεδομένα που εισάγει ο χρήστης, ώστε να είναι δυνατή η υλοποίηση της σύνδεσης με τη

βάση δεδομένων rmlab. Εάν τα στοιχεία είναι αποδεκτά το πρόγραμμα ανοίγει τη σύνδεση με το διακομιστή, εμφανίζει την κύρια φόρμα του προγράμματος (frmMain) και κλείνει τη σύνδεση. Εάν τα στοιχεία δεν είναι αποδεκτά το πρόγραμμα εμφανίζει ένα μήνυμα με το οποίο πληροφορεί τον χρήστη για την ύπαρξη λάθους κατά την προσπάθεια σύνδεσης με τον διακομιστή.



Σχήμα 5.2: Φόρμα σύνδεσης με τη βάση δεδομένων rmlab

```
Private Sub btnLogin_Click()  
  
    Server = txtServer.Text  
    UserName = txtUsername.Text  
    PassWord = mtxtPassword.Text  
  
    conn = New MySqlConnection()  
    conn.ConnectionString = "server=" & Server & ";" _  
        & "user id=" & UserName & ";" _  
        & "password=" & PassWord & ";" _  
        & "database=rmlab"  
  
    Try  
        conn.Open()  
        frmMain.Show()  
        Me.Hide()  
        conn.Close()  
    Catch myerror As MySqlException  
        MessageBox.Show("Error Connecting to Database: " & myerror.Message)  
    Finally  
        conn.Dispose()  
    End Try
```

Σχήμα 5.3: Κώδικας που εκτελείται με το πάτημα του κουμπιού btnLogin

Όπως γίνεται αντιληπτό από την ανάγνωση του κώδικα που εκτελείται όταν ο χρήστης πατήσει το κουμπί `btnLogin`, η Visual Basic .NET διαθέτει μια ειδική για τον χειρισμό σφαλμάτων (εξαιρέσεων, *exceptions*) σύνταξη, τύπου `Try...Catch...Finally`. Μετά την λέξη `Try` και πριν τη λέξη `Catch` τοποθετείται ο κώδικας με το πιθανό σφάλμα. Η λέξη `Catch` χρησιμοποιείται για την υπόδειξη κώδικα που πρόκειται να εκτελεστεί εάν συμβεί το πιθανό σφάλμα, ενώ ο κώδικας που τοποθετείται μετά την λέξη `Finally` εκτελείται ανεξάρτητα με το αν συνέβη το σφάλμα ή όχι.

5.4 Φόρμα προσθήκης δεδομένων στη βάση `rmlab`

Στο σχήμα 5.4 εμφανίζεται η φόρμα του προγράμματος RockLab που δίνει τη δυνατότητα στο χρήστη να εισάγει (`add`) δεδομένα στη βάση `rmlab` και ονομάζεται `frmAdd`. Τα διάφορα εργαλεία της φόρμας `frmAdd` ομαδοποιούνται σε δύο ομάδες που αντιπροσωπεύουν τις κοινές περιγραφικές παραμέτρους (*common descriptive properties*) και τις μετρήσιμες παραμέτρους (*measured properties*) του εκάστοτε τύπου πειράματος.

Ειδικότερα, το αριστερό τμήμα της φόρμας (σύνολο κοινών παραμέτρων) αποτελείται από 13 `comboboxes` (`cmbProject`, `cmbGroup`, `cmbContactPerson`, `cmbPlace`, `cmbForm`, `cmbMaterial`, `cmbDiscontinuities`, `cmbInclusions`, `cmbPorosity`, `cmbPreparation`, `cmbConfidence`, `cmbFailure` και `cmbLoading`) για την επιλογή της εκάστοτε παραμέτρου από μια καθορισμένη (από τη βάση) λίστα, 6 `textboxes` (`txtAxialDim`, `txtTransverseDim1`, `txtTransverseDim2`, `txtDensity`, `txtFilename` και `txtComments`) για τη συμπλήρωση των αντίστοιχων παραμέτρων του πίνακα `master` της βάσης `rmlab` και 1 εργαλείο `DatePicker` για τη διευκόλυνση εισαγωγής της παραμέτρου της ημερομηνίας.

Για τη διευκόλυνση της εισαγωγής της παραμέτρου ονόματος αρχείου (`filename`) κατασκευάστηκε ένα κουμπί το οποίο ονομάζεται `btnBrowse`. Με το πάτημα του εν λόγω κουμπιού το πρόγραμμα εμφανίζει ένα διάλογο (`Open File Dialog`) με τον οποίο δίνεται η δυνατότητα στο χρήστη να ανατρέξει στο σκληρό δίσκο του υπολογιστή του και να δείξει το αρχείο του οποίου το όνομα πρόκειται να αποθηκευτεί στην προς προσθήκη εγγραφή. Στο σχήμα 5.5 εμφανίζεται ο διάλογος αυτός ο οποίος αποτελεί ένα εργαλείο της Visual Basic .NET που στο πρόγραμμα ονομάζεται `OpenFileDialog1`. Εφόσον ο χρήστης επιλέξει το επιθυμητό αρχείο και πατήσει το κουμπί `Open`, ο διάλογος κλείνει και το πεδίο `txtFilename` εμφανίζει το όνομα του αρχείου που επιλέχτηκε.

Add Record

Common Descriptive Properties

* Project Name:

* Group Name:

* Contact Person:

* Date:

* Place of Origin:

* Specimen Form:

* Axial Dim:

* Transverse Dim1:

Transverse Dim 2:

* Density:

* Material Type:

* Discontinuities:

* Inclusions:

* Porosity:

* Preparation:

* Confidence:

* Failure Mode:

* Loading Mode:

Filename:

Comments:

Measured Properties

* UCS:

Adjusted UCS:

* Average Modulus:

Tangent Modulus:

Secant Modulus:

Poisson Ratio:

Max Deformation:

Energy to Failure:

Test Duration:

Uniaxial

Bending

Triaxial

Tension

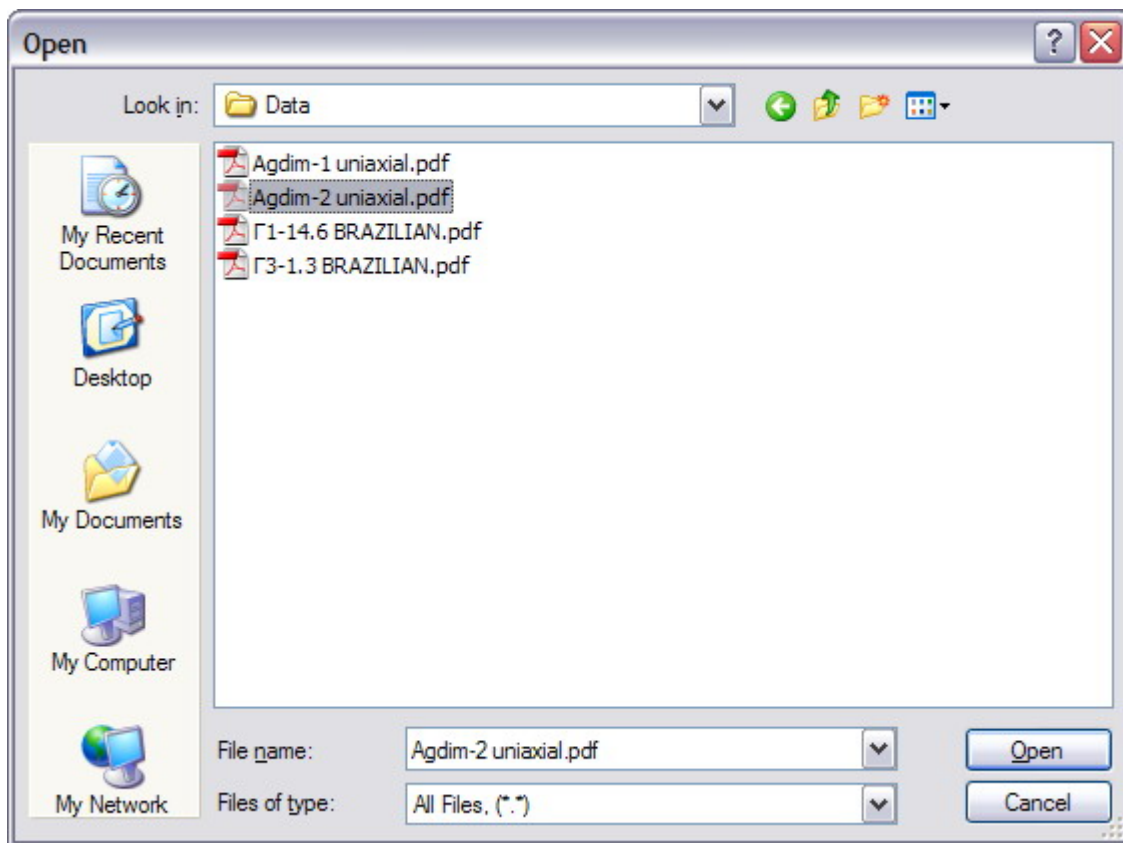
Ultrasonic

Cancel

Add Record

Update Subtable

Σχήμα 5.4: Η φόρμα προσθήκης δεδομένων (frmAdd) του προγράμματος RockLab



Σχήμα 5.5: Ο διάλογος για την υπόδειξη του ονόματος αρχείου

Το δεξί τμήμα της φόρμας frmAdd αποτελείται από ένα εργαλείο της Visual Basic .NET που ονομάζεται TabControl και δίνει τη δυνατότητα ομαδοποίησης διαφόρων εργαλείων μιας εφαρμογής σε καρτέλες (tabs). Το εργαλείο που στη συγκεκριμένη εφαρμογή ονομάστηκε TabControl1 αποτελείται από πέντε καρτέλες (uniaxial, bending, triaxial, tension, ultrasonic) κάθε μία από τις οποίες αντιπροσωπεύει έναν από τους αντίστοιχους πέντε τύπους πειραμάτων και αποτελείται από textboxes για τη συμπλήρωση των παραμέτρων του εκάστοτε τύπου πειράματος. Στο σχήμα 5.4 εμφανίζεται η καρτέλα uniaxial με τα αντίστοιχα πεδία της ενώ οι υπόλοιπες καρτέλες και τα αντίστοιχα πεδία τους παρουσιάζονται στο σχήμα 5.6. Ο χρήστης έχει τη δυνατότητα να ανατρέξει στην κατάλληλη καρτέλα πατώντας τον τίτλο της που βρίσκεται στο κάτω μέρος του εργαλείου TabControl1.

Στο κάτω μέρος της φόρμας κατασκευάστηκαν τρία κουμπιά btnCancel, btnUpdate και btnAddRecord. Ο σκοπός του πρώτου είναι να κλείνει τη φόρμα, του δεύτερου να εισάγει εγγραφές στους υποπίνακες της βάσης και του τρίτου να εισάγει εγγραφές στους κύριους πίνακες της βάσης.

<p>* Bending Strength: <input type="text"/> MPa</p> <p>Max Deformation: <input type="text"/> mm</p> <p>Test Duration: <input type="text"/> sec</p>	<p>* Maximum Stress: <input type="text"/> MPa</p> <p>Lateral Stress: <input type="text"/> MPa</p> <p>* Average Modulus: <input type="text"/> GPa</p> <p>Tangent Modulus: <input type="text"/> GPa</p> <p>Secant Modulus: <input type="text"/> GPa</p> <p>Max Deformation: <input type="text"/> mm</p> <p>Energy to Failure: <input type="text"/> Joule</p> <p>Cohesion: <input type="text"/> kPa</p> <p>Internal Friction: <input type="text"/> deg</p> <p>Test Duration: <input type="text"/> sec</p>
<p>Uniaxial Bending Triaxial Tension Ultrasonic</p>	<p>Uniaxial Bending Triaxial Tension Ultrasonic</p>
<p>* Tensile Strength: <input type="text"/> MPa</p> <p>Energy to Failure: <input type="text"/> Joule</p> <p>Test Duration: <input type="text"/> sec</p>	<p>* Vp: <input type="text"/> m/s</p> <p>* Vs: <input type="text"/> m/s</p> <p>E(dynamic): <input type="text"/> GPa</p> <p>Poisson Ratio: <input type="text"/></p>
<p>Uniaxial Bending Triaxial Tension Ultrasonic</p>	<p>Uniaxial Bending Triaxial Tension Ultrasonic</p>

Σχήμα 5.6: Οι καρτέλες bending, triaxial, tension και ultrasonic του εργαλείου TabControl1

5.4.1 Φόρτωση των επιλογών των comboboxes

Τα comboboxes είναι εργαλεία της Visual Basic .NET που δίνουν τη δυνατότητα στο χρήστη να συμπληρώσουν ένα πεδίο είτε επιλέγοντας από κάποιες προκαθορισμένες τιμές με τη μορφή λίστας, είτε εισάγοντας μια διαφορετική από τις προκαθορισμένες τιμή με το πληκτρολόγιο. Κατά τη φόρτωση της φόρμας frmAdd, τα 13 comboboxes παίρνουν τις προκαθορισμένες τιμές τους από τις τιμές των αντίστοιχων υπό-πινάκων της βάσης δεδομένων rmlab. Στο σχήμα 5.7 εμφανίζεται ο κώδικας με τον οποίο γίνεται η αντιστοίχιση τις προκαθορισμένης λίστας τιμών του combobox cmbProject με τον αντίστοιχο υπό-πίνακα Projects της βάσης δεδομένων rmlab:

```
Dim conn As New MySqlConnection()
Dim myCommand As New MySqlCommand()
Dim myAdapter As New MySqlDataAdapter()

Dim myDataProjects As New DataTable()
Dim PROJECTS As String

conn.ConnectionString = "..."
PROJECTS = "select p.id, p.name from projects as p"

Try
    conn.Open()
    Try
        myCommand.Connection = conn
        myCommand.CommandText = PROJECTS

        myAdapter.SelectCommand = myCommand
        myAdapter.Fill(myDataProjects)

        cmbProject.DisplayMember = "name"
        cmbProject.ValueMember = "Id"
        cmbProject.SelectedValue = 0
        cmbProject.Text = "-Please Select-"
    Catch myerror As MySqlException
        MsgBox("There was an error filling Project combo: " & myerror.Message)
    End Try
Catch myerror As MySqlException
    MessageBox.Show("Error Connecting to Database: " & myerror.Message)
Finally
    conn.Close()
End Try
```

Σχήμα 5.7: Κώδικας για την φόρτωση των τιμών των comboboxes από τη βάση δεδομένων rmlab

Αρχικά δημιουργείται ένα αντικείμενο τύπου MySqlConnection (conn), ένα αντικείμενο τύπου MySqlCommand (myCommand), ένα αντικείμενο τύπου MySqlDataAdapter (myAdapter) και ένα αντικείμενο τύπου DataTable (myDataProjects). Αφού οριστεί η ακολουθία χαρακτήρων σύνδεσης (connectionstring) το πρόγραμμα ανοίγει τη σύνδεση με τη βάση. Στη συνέχεια μέσω του αντικειμένου myCommand εκτελείται ένα ερώτημα SELECT το οποίο ορίζεται από την ακολουθία χαρακτήρων (string) PROJECTS. Το ερώτημα αυτό επιλέγει όλες τις τιμές που περιέχονται στα πεδία id και name του πίνακα projects της βάσης δεδομένων rmlab και έτσι παράγεται ένας πίνακας αποτελεσμάτων που αποτελείται από δύο στήλες (id και name). Μέσω του αντικειμένου MyAdapter οι τιμές αυτού του πίνακα αποτελεσμάτων αποθηκεύονται στο αντικείμενο MyDataProjects. Οι τιμές του πεδίου name φορτώνονται στη λίστα του cmbProjects, ενώ οι τιμές του πεδίου id αντιστοιχούνται εσωτερικά με τον αριθμό της λίστας της κάθε προκαθορισμένης τιμής του cmbProjects μέσω της ιδιότητας των comboboxes που ονομάζεται ValueMember.

Για παράδειγμα εάν ο πίνακας projects της βάσης δεδομένων rmlab διαθέτει δύο τιμές id=1 name=pened και id=2 name=arximidis τότε οι προκαθορισμένες τιμές του combobox cmbProject θα είναι οι pened και arximidis που εσωτερικά (χωρίς να φαίνεται στο χρήστη) θα αντιστοιχούν στον αριθμό 1 η πρώτη και στον αριθμό 2 η δεύτερη. Στο τέλος του κώδικα καθορίζεται το αρχικό κείμενο που θα εμφανίζει το κείμενο και το οποίο προτρέπει το χρήστη να επιλέξει μία από τις προκαθορισμένες τιμές (please select) και κλείνει η σύνδεση με τη βάση δεδομένων. Με ακριβώς αντίστοιχο τρόπο φορτώνονται οι κατάλληλες προκαθορισμένες τιμές σε όλα τα comboboxes.

5.4.2 Προσθήκη εγγραφών στους υπό-πίνακες

Τα comboboxes δίνουν τη δυνατότητα εισαγωγής δεδομένων εκτός του προκαθορισμένων τιμών της λίστας. Έτσι ο χρήστης έχει τη δυνατότητα να προσθέτει εγγραφές στους διάφορους υπό-πίνακες της βάσης δεδομένων rmlab μέσω του αντίστοιχου combobox της φόρμας frmAdd. Αρχικά θα πρέπει να εισάγει την τιμή που επιθυμεί στο αντίστοιχο combobox και στη συνέχεια να πατήσει το κουμπί btnUpdate. Το πρόγραμμα διενεργεί έναν έλεγχο για το ποιο combobox πρόκειται να ανανεωθεί και στη συνέχεια εκτελεί ένα SQL ερώτημα προσθήκης στον αντίστοιχο πίνακα. Στο σχήμα 5.8 εμφανίζεται ο κώδικας που εκτελείται όταν ο χρήστης πατήσει το κουμπί btnUpdate για να εισάγει μία νέα εγγραφή στον πίνακα projects της βάσης δεδομένων rmlab:


```

Dim conn As New MySqlConnection()
Dim myCommand As New MySqlCommand()

conn.ConnectionString = "server=" + frmLoggin.Server + "; user id=" + frmLoggin.User

Try
    myCommand.Connection = conn
    myCommand.CommandText = "INSERT INTO projects(name) VALUES(?name)"
    myCommand.Parameters.Add("?name", cmbProject.Text)
    myCommand.ExecuteNonQuery()
Catch myerror As MySqlException
    MsgBox("There was an error adding a Project to database: " & myerror.Message)
End Try

```

Σχήμα 5.8: Κώδικας που εκτελείται για την προσθήκη νέας τιμής στο cmbProject και στον αντίστοιχο υπό-πίνακα projects της βάσης δεδομένων rmlab

Αφού πραγματοποιηθεί η σύνδεση με τη βάση δεδομένων το πρόγραμμα εκτελεί ένα ερώτημα Insert Into στη βάση δεδομένων, με το οποίο πρόκειται να προστεθεί η νέα εγγραφή στον πίνακα projects. Το ερώτημα αυτό είναι παραμετρικό καθώς η τιμή που πρόκειται να εγγραφεί στη βάση δεδομένων είναι μια μεταβλητή που ονομάζεται ?name (το ερωτηματικό στην αρχή δηλώνει ότι πρόκειται για μεταβλητή). Στη συνέχεια με την εντολή Parameters.Add του αντικειμένου myCommand αντιστοιχίζεται η μεταβλητή αυτή που πρόκειται να εγγραφεί στη βάση δεδομένων με το κείμενο (text) που έχει εισάγει ο χρήστης στο πεδίο cmbProject. Εφόσον η μεταβλητή έχει πάρει συγκεκριμένη τιμή (το κείμενο του χρήστη) το ερώτημα είναι δυνατόν να εκτελεστεί με την εντολή ExecuteNonQuery του αντικειμένου myCommand. Για να προστεθεί η νέα τιμή του υπό-πίνακα σαν επιλογή στο combobox εκτελείται ξανά ο κώδικας της προηγούμενης παραγράφου και η νέα τιμή της βάσης φορτώνεται (μαζί με το id της) ως καινούρια επιλογή στο αντίστοιχο combobox. Στο σχήμα 5.9 παρουσιάζεται η προσθήκη της νέας τιμής «pithagoras» στον υπό-πίνακα projects και στο cmbProject.

5.4.3 Προσθήκη εγγραφών στους βασικούς πίνακες

Εφόσον ο χρήστης επιλέξει κατάλληλες τιμές παραμέτρων για όλα τα υποχρεωτικά πεδία των πινάκων (δηλώνονται με το σύμβολο του μπλε αστερίσκου δίπλα από την επιγραφή του κάθε πεδίου) και πατώντας το κουμπί btnAddRecord, η εγγραφή αποθηκεύεται στη βάση. Τα σχήματα 5.10, 5.11 και 5.12 παρουσιάζουν τον κώδικα που εκτελείται για την εγγραφή των παραμέτρων ενός πειράματος εφελκυσμού (bending). Τα τρία βήματα που ακολουθούνται στον αλγόριθμο είναι τα εξής:

Σχήμα 5.9: Προσθήκη της τιμής «pithagoras» στον υπό-πίνακα projects της βάσης δεδομένων rmlab και στο cmbProject της φόρμας frmAdd

- Ανάλογα με την καρτέλα πειράματος που έχει επιλέξει ο χρήστης εισάγονται οι μη κοινές παράμετροι στον αντίστοιχο πίνακα, μέσω παραμετρικών ερωτημάτων προσθήκης εγγραφής (σχήμα 5.10)
- Εκτελείται ένα ερώτημα στον πίνακα αυτό για να βρεθεί το id της τελευταίας εγγραφής του πίνακα (δηλαδή το id της εν λόγω εγγραφής) και αποθηκεύεται το id αυτό σε μια μεταβλητή (MaxId) (σχήμα 5.11)
- Εκτελείται ερώτημα προσθήκης εγγραφής με το οποίο εισάγονται οι κοινές παράμετροι της εγγραφής στον πίνακα master, και το πεδίο childFK του πίνακα αυτού παίρνει την τιμή της μεταβλητής MaxId (σχήμα 5.12)

```

Try
myCommand.Connection = conn
myCommand.CommandText = "INSERT INTO bending(bendingstrength, " _
& "maxdeformation, duration) VALUES(?bendingstrength,?maxdeformation,?duration)"
myCommand.Parameters.Add("?bendingstrength", Val(txtBendingStrengthBe.Text))
If txtMaxDeformationBe.Text = "" Then
myCommand.Parameters.Add("?maxdeformation", DBNull.Value)
Else
myCommand.Parameters.Add("?maxdeformation", Val(txtMaxDeformationBe.Text))
End If
If txtDurationBe.Text = "" Then
myCommand.Parameters.Add("?duration", DBNull.Value)
Else
myCommand.Parameters.Add("?duration", Int(Val(txtDurationBe.Text)))
End If
myCommand.ExecuteNonQuery()
Catch myerror As MySqlException
MsgBox("There was an error adding a Record in Bending Table: " & myerror.Message)
End Try

```

Σχήμα 5.10: Κώδικας για την εισαγωγή δεδομένων στον πίνακα bending

```

Try
    myCommand.Connection = conn
    myCommand.CommandText = "SELECT MAX(bending.id) FROM bending"
    MaxID = myCommand.ExecuteScalar
Catch myerror As MySqlException
    MsgBox("There was an error adding a Record in Bending Table: " & myerror.Message)
End Try

```

Σχήμα 5.11: Κώδικας για την εύρεση του id της τελευταίας εγγραφής που προστέθηκε στον πίνακα bending

```

Try
    myCommand.Connection = conn
    myCommand.CommandText = "INSERT INTO master(projectFK, groupFK, " _
    & "contactpersonFK, date, placeFK, materialFK, " _
    & "formFK, axialdim, traversedim1, traversedim2, density, " _
    & "discontinuitiesFK, inclusionsFK, porosityFK, preparationFK, " _
    & "confidenceFK, failureFK, filename, testFK, childFK, " _
    & "loadingFK, comments) VALUES (?projectFK, ?groupFK, " _
    & "?contactpersonFK, ?date, ?placeFK, ?materialFK, ?formFK, " _
    & "?axialdim, ?traversedim1, ?traversedim2, ?density, " _
    & "?discontinuitiesFK, ?inclusionsFK, ?porosityFK, ?preparationFK, " _
    & "?confidenceFK, ?failureFK, ?filename, 2, ?childFK, " _
    & "?loadingFK, ?comments)"
    myCommand.Parameters.Add("?projectFK", cmbProject.SelectedValue)
    myCommand.Parameters.Add("?groupFK", cmbGroup.SelectedValue)
    myCommand.Parameters.Add("?contactpersonFK", cmbContactPerson.SelectedValue)
    myCommand.Parameters.Add("?date", datPicker.Value.Date)
    myCommand.Parameters.Add("?placeFK", cmbPlace.SelectedValue)
    myCommand.Parameters.Add("?materialFK", cmbMaterial.SelectedValue)
    myCommand.Parameters.Add("?formFK", cmbForm.SelectedValue)
    myCommand.Parameters.Add("?axialdim", Val(txtAxialDim.Text))
    myCommand.Parameters.Add("?traversedim1", Val(txtTraverseDim1.Text))
    myCommand.Parameters.Add("?traversedim2", Val(txtTraverseDim2.Text))
    myCommand.Parameters.Add("?density", Val(txtDensity.Text))
    myCommand.Parameters.Add("?discontinuitiesFK", cmbDiscontinuities.SelectedValue)
    myCommand.Parameters.Add("?inclusionsFK", cmbInclusions.SelectedValue)
    myCommand.Parameters.Add("?porosityFK", cmbPorosity.SelectedValue)
    myCommand.Parameters.Add("?preparationFK", cmbPreparation.SelectedValue)
    myCommand.Parameters.Add("?confidenceFK", cmbConfidence.SelectedValue)
    myCommand.Parameters.Add("?failureFK", cmbFailure.SelectedValue)
    myCommand.Parameters.Add("?filename", txtFilename.Text)
    myCommand.Parameters.Add("?childFK", MaxID)
    myCommand.Parameters.Add("?loadingFK", cmbLoading.SelectedValue)
    If txtComments.Text = "" Then
        myCommand.Parameters.Add("?comments", DBNull.Value)
    Else
        myCommand.Parameters.Add("?comments", txtComments.Text)
    End If
    myCommand.ExecuteNonQuery()
    MsgBox("Record Successfully Added to Database!", MsgBoxStyle.Exclamation)
Catch myerror As MySqlException
    MsgBox("There was an error adding a Record in Master Table: " & myerror.Message)
End Try

```

Σχήμα 5.12: Κώδικας για την αποθήκευση των κοινών παραμέτρων της εγγραφής στον πίνακα master της βάσης δεδομένων rmlab

5.5 Φόρμα αναζήτησης εγγραφών

Η φόρμα του προγράμματος RockLab, όπου δίνεται στο χρήστη η δυνατότητα να ανατρέξει σε όλες τις εγγραφές της βάσης δεδομένων αλλά και να αναζητήσει εγγραφές με συγκεκριμένα χαρακτηριστικά, είναι η φόρμα αναζήτησης εγγραφών ή κύρια φόρμα του προγράμματος και ονομάζεται frmMain (σχήμα 5.13). Το βασικό εργαλείο της φόρμας αυτής είναι ένα εργαλείο τύπου DataGridView που ονομάζεται dgvDatabase και σκοπός του είναι να δείχνει πλήθη εγγραφών από τη βάση δεδομένων. Με το εργαλείο αυτό εμφανίζονται επιλεγμένα πεδία (τα σημαντικότερα) του πίνακα master για κάθε εγγραφή.

Η φόρμα frmMain διαθέτει δύο τύπους λειτουργίας (modes). Ο πρώτος τύπος λειτουργίας επιτρέπει στο χρήστη να ανατρέχει σε όλες τις εγγραφές της βάσης δεδομένων rmlab (view mode), ενώ ο δεύτερος δίνει τη δυνατότητα για αναζήτηση συγκεκριμένων εγγραφών που πληρούν κάποια κριτήρια τα οποία επιλέγονται από τον χρήστη (search mode). Για την εναλλαγή από τη μία λειτουργία στην άλλη χρησιμοποιήθηκε ένα αντικείμενο της Visual Basic .NET που ονομάζεται CheckBox. Το CheckBox ονομάστηκε chbEnableSearch και βρίσκεται στην πάνω δεξιά γωνία της φόρμας frmMain. Όταν αυτό το CheckBox δεν είναι επιλεγμένο τότε η φόρμα βρίσκεται στον πρώτο τύπο λειτουργίας (view), ενώ όταν είναι επιλεγμένο η φόρμα βρίσκεται στο δεύτερο τύπο λειτουργίας (search). Κατά τη φόρτωση της φόρμας το CheckBox δεν είναι επιλεγμένο και επομένως η φόρμα βρίσκεται σε λειτουργία προβολής (σχήμα 5.13).

Στο κάτω μέρος της φόρμας κατασκευάστηκαν 6 κουμπιά (command buttons) των οποίων τα ονόματα και ο σκοπός τους είναι:

- btnCancel: Το πάτημα του κλείνει την εφαρμογή.
- btnAddRecord: Με το πάτημα του εμφανίζεται η φόρμα frmAdd για την προσθήκη εγγραφών στη βάση δεδομένων rmlab.
- btnEditRecord: Με το πάτημα του εμφανίζεται η φόρμα frmEdit για την επεξεργασία των δεδομένων κάποια συγκεκριμένης εγγραφής της βάσης δεδομένων rmlab.
- btnDeleteRecord: Με το πάτημα του εμφανίζεται η φόρμα frmDelete για τη διαγραφή συγκεκριμένης εγγραφής από τη βάση δεδομένων rmlab.
- btnOpenFile: Με το πάτημα του ανοίγει το αρχείο του οποίου το όνομα έχει αποθηκευτεί στο πεδίο filename της βάσης δεδομένων rmlab.
- btnCartPlot: Με το πάτημα του εμφανίζεται η φόρμα frmChartPlot για την κατασκευή διαγραμμάτων από συγκεκριμένα αριθμητικά δεδομένα της βάσης rmlab.

id	Test_Type	date	Contact_Person	Project	Group Name	Place	Material	Form
31	tension	14/12/2005	Kritikakis	none	skynrodema	ag. dimitrios	limestone	cylinder
28	uniaxial	14/12/2005	Kritikakis	pened	skynrodema	ag. dimitrios	limestone	cylinder
27	uniaxial	17/12/2004	Kritikakis	pened	skynrodema	ag. dimitrios	limestone	prism
25	uniaxial	15/12/2003	Kritikakis	pened	skynrodema	ag. dimitrios	limestone	cylinder
23	tension	26/5/2004	Kritikakis	aximidis	skynrodema	chania	limestone	cube
22	triaxial	26/5/2004	Kritikakis	aximidis	skynrodema	chania	limestone	cube
20	uniaxial	26/5/2004	Kritikakis	aximidis	skynrodema	chania	limestone	cube
19	uniaxial	3/12/2003	Kapetanakis	pened	tuc	chania	cement	prism
18	uniaxial	13/12/2003	Kritikakis	pened	skynrodema	ag. dimitrios	limestone	cylinder
17	uniaxial	3/12/2003	Kritikakis	pened	skynrodema	ag. dimitrios	limestone	prism

Total Records: 25 Current Record: 22 Sort By: Id Asc Desc Enable Search ☐

Project Name: All Group Name: All Contact Person: All Date: before 12/3/2006 Place of Origin: All Specimen Form: All Material Type: All Test Type: All

Number Of Displayed Records: 10 << page 1 of 3 >>

Open File Delete Record Edit Record Add Record Cancel ChartPlot

Σχήμα 5.13: Η κύρια φόρμα frmMain του προγράμματος RockLab σε λειτουργία προβολής

5.5.1 Λειτουργία προβολής εγγραφών (view mode)

Στη λειτουργία προβολής τα διάφορα εργαλεία (εκτός των κουμπιών) που βρίσκονται γύρω από το κύριο εργαλείο `dgvDatabase` είναι ενεργά. Τα εργαλεία αυτά καθώς και ο σκοπός τους είναι:

- Ένα εργαλείο label (πινακίδα) που ονομάζεται `lblNumTotRec` και πληροφορεί το χρήστη σχετικά με το πλήθος όλων των εγγραφών της βάσης δεδομένων
- Ένα εργαλείο label που ονομάζεται `lblId` και πληροφορεί το χρήστη σχετικά με το `id` της εγγραφής που είναι επιλεγμένη
- Ένα εργαλείο `combobox` που ονομάζεται `cmbShort` και δίνει τη δυνατότητα στο χρήστη να επιλέξει το πεδίο του πίνακα master με βάση το οποίο το πρόγραμμα θα ταξινομεί τις εγγραφές. Οι προκαθορισμένες επιλογές του εργαλείου αυτού, δηλαδή τα πεδία με βάση τα οποία είναι δυνατόν να γίνεται η ταξινόμηση των εγγραφών είναι τα : `id`, `project`, `group`, `contactperson`, `date`, `place`, `material`, `axialdim`, `transversedim1`, `density` και `testtype`
- Δύο εργαλεία `RadioButton` που ονομάζονται `rdbAsc` και `rdbDesc` και δίνουν τη δυνατότητα στο χρήστη να επιλέξει μεταξύ αύξουσας (`ascending`) και φθίνουσας (`descending`) ταξινόμησης αντίστοιχα.
- Ένα εργαλείο `combobox` το οποίο ονομάζεται `cmbNumDisRec` και δίνει τη δυνατότητα στο χρήστη να επιλέξει το πλήθος των εγγραφών που θα εμφανίζονται σε κάθε «σελίδα» του `dgvDatabase`
- Δύο κουμπιά που ονομάζονται `btnBackward` και `btnForward` που δίνουν τη δυνατότητα στο χρήστη να ανατρέξει μεταξύ των διαφορετικών «σελίδων» του `dgvDatabase`
- Δύο εργαλεία label που ονομάζονται `lblArithmitis` και `lblParonomastis` και πληροφορούν το χρήστη για τον αριθμό της τρέχουσας «σελίδας» καθώς και για τον αριθμό του συνόλου των «σελίδων» του `dgvDatabase` αντίστοιχα

Στο σχήμα 5.14 εμφανίζεται το ερώτημα που εκτελείται στη βάση δεδομένων `rm1lab` προκειμένου να εμφανιστούν οι σημαντικότερες κοινές παράμετροι όλων των εγγραφών της βάσης. Η μεταβλητή `strShort` παίρνει την τιμή της από το `cmbShort` και καθορίζει το πεδίο με βάση το οποίο γίνεται η ταξινόμηση των αποτελεσμάτων. Η μεταβλητή `Fora` συνδέεται με τα `rdbAsc` και `rdbDesc` και καθορίζει τη φορά της ταξινόμησης. Οι μεταβλητές `Arithmitis` και `Paronomastis` συνδέονται με το `cmbNumDisRec` και με το πλήθος των πατημάτων των κουμπιών `btnForward` και `btnBackWard` από το χρήστη.


```

SQL = "select master.id, tests.name as Test_Type, master.date, contactpersons.name" _
& "as Contact_Person, " _
& "projects.name as Project, groups.name as GroupName, places.name as Place, " _
& "materials.type as Material, forms.type as Form, master.axialdim as Axial_Dim, " _
& "master.traversedim1 as Trav_Dim1, master.density, master.filename " _
& "from master, tests, contactpersons, projects, groups, places, materials, forms " _
& "where master.contactpersonFK=contactpersons.id and master.testFK=tests.id" _
& "and master.projectFK=projects.id " _
& "and master.groupFK=groups.id and master.placeFK=places.id and " _
& "master.materialFK=materials.id and master.formFK=forms.id " _
& "order by " + strShort + " " + Fora + " limit " + CStr(Arxi) + "," + CStr(Vima)

```

Σχήμα 5.14: Το ερώτημα που εκτελείται στη βάση rmlab κατά τη λειτουργία προβολής της κύριας φόρμας του προγράμματος RockLab

Στο σχήμα 5.15 εμφανίζεται ο κώδικας που εκτελείται για να επιτευχθεί η μεταφορά των δεδομένων από τη βάση rmlab στο εργαλείο dgvDatabase της κύριας φόρμας του προγράμματος RockLab, όταν αυτή βρίσκεται σε κατάσταση προβολής. Το αντικείμενο conn ανοίγει τη σύνδεση με το MySQL διακομιστή, το αντικείμενο MySqlCommand ορίζει την ακολουθία χαρακτήρων σύνδεσης (connection string) και εκτελεί το ερώτημα SQL του σχήματος 5.14, το αντικείμενο MyAdapter μεταφέρει τα αποτελέσματα του ερωτήματος στο αντικείμενο MyData, από το οποίο διαβάζει τις τιμές που πρόκειται να εμφανίσει το εργαλείο dgvDatabase. Για να επιλέξει ο χρήστης μια εγγραφή του αντικειμένου dgvDatabase (και κατ' επέκταση της βάσης) αρκεί να κάνει κλικ με το ποντίκι στο πρώτο κελί (κελί-τίτλος) της εγγραφής (γραμμής).

```

Try
    conn.Open()
Try
    myCommand.Connection = conn
    myCommand.CommandText = SQL
    myAdapter.SelectCommand = myCommand
    myAdapter.Fill(myData)
    dgvDatabase.ReadOnly = True
    dgvDatabase.DataSource = myData
Catch myerror As MySqlException
    MsgBox("error reading from the database: " & myerror.Message)
End Try
Catch myerror As MySqlException
    MessageBox.Show("Error Connecting to Database: " & myerror.Message)
Finally
    conn.Close()
End Try

```

Σχήμα 5.15: Ο κώδικας που εκτελείται για την μεταφορά των δεδομένων από τη βάση rmlab στο εργαλείο dgvDatabase της εφαρμογής RockLab

5.5.2 Λειτουργία αναζήτησης (search mode)

Η λειτουργία αναζήτησης της φόρμας frmMain ενεργοποιείται με το εργαλείο chbEnableSearch. Όταν η φόρμα βρίσκεται σε λειτουργία αναζήτησης εγγραφών τα διάφορα εργαλεία που βρίσκονται γύρω από το κύριο εργαλείο dgvDatabase απενεργοποιούνται (το εργαλείο πλέον δεν αποτελείται από πολλαπλές «σελίδες» αλλά από μία μοναδική) ενώ τα 9 comboboxes που βρίσκονται στο δεξί μέρος της φόρμας ενεργοποιούνται. Τα comboboxes αυτά φορτώνουν τις προκαθορισμένες τιμές τους από τη βάση δεδομένων rmlab, με τον ίδιο τρόπο που εφαρμόστηκε για τα comboboxes της φόρμας frmAdd και ο οποίος αναλύθηκε στην προηγούμενη παράγραφο. Τα κριτήρια με τα οποία είναι δυνατόν ο χρήστης να αναζητήσει συγκεκριμένες ή συγκεκριμένη εγγραφή και τα αντίστοιχα εργαλεία είναι:

- Όνομα Project – cmbProject
- Όνομα Group – cmbGroup
- Contact Person – cmbContactPerson
- Ημερομηνία – DatePicker
- Τόπος προέλευσης δοκιμίου – cmbPlace
- Σχήμα δοκιμίου – cmbForm
- Υλικό δοκιμίου – cmbMaterial
- Τύπος δοκιμής – cmbTest

Ειδικά για το κριτήριο της ημερομηνίας κατασκευάστηκε ένα combobox το οποίο ονομάστηκε cmbTelestisDate και σκοπός του είναι να ορίζει τον τελεστή σύγκρισης της ημερομηνίας (πριν από τις [ημερομηνία], μετά από τις [ημερομηνία], στις [ημερομηνία]). Στο σχήμα 5.16 εμφανίζεται η κύρια φόρμα (frmMain) του προγράμματος RockLab σε λειτουργία αναζήτησης. Καθώς ο χρήστης επιλέγει κριτήρια το πρόγραμμα «φιλτράρει» τις εγγραφές εμφανίζοντας στο dgvDatabase αυτές που ικανοποιούν τα επιλεγμένα κριτήρια. Για παράδειγμα στο σχήμα 5.16 εμφανίζονται οι εγγραφές που ανήκουν στο Project pened, στο Group tuc και είναι πειράματα μονοαξονικής θλίψης (uniaxial). Στο πάνω αριστερό μέρος της φόρμας το πρόγραμμα πληροφορεί το χρήστη σχετικά με το πλήθος των αποτελεσμάτων αναζήτησης. Στο σχήμα 5.17 εμφανίζεται το ερώτημα SEARCHSQL που εκτελείται καθώς ο χρήστης επιλέγει κριτήρια αναζήτησης καθώς και η μορφή που έχει μία (strForm) από τις μεταβλητές που χρησιμοποιούνται για τον ορισμό των διαφόρων κριτηρίων.

frmMain

Search Records: 3 Current Record: 0

☐ Id
 ☐ Asc
 ☒ Desc

Id	Test_Type	date	Contact_Person	Project	Group Name	Place	Material	Form
12	uniaxial	16/2/2004	Kritikakis	pened	tuc	ag. dimitrios	limestone	cube
13	uniaxial	16/2/2004	Kapetanakis	pened	tuc	ag. dimitrios	limestone	cube
19	uniaxial	3/12/2003	Kapetanakis	pened	tuc	charia	cement	prism

Project Name: pened
 Group Name: tuc
 Contact Person: -----All-----
 Date: before 13/ 3 /2006
 Place of Origin: -----All-----
 Specimen Form: -----All-----
 Material Type: -----All-----
 Test Type: uniaxial

Enable Search ☒

Open File
 Delete Record
 Edit Record
 Add Record
 Cancel
 ChartPlot

Σχήμα 5.16: Η κύρια φόρμα frmMain του προγράμματος RockLab σε λειτουργία αναζήτησης


```

SEARCHSQL = "select master.id, tests.name as Test_Type," _
& "master.date, contactpersons.name as Contact_Person, " _
& "projects.name as Project, groups.name as GroupName, places.name as Place, " _
& "materials.type as Material, forms.type as Form, master.axialdim as Axial_Dim, " _
& "master.traversedim1 as Trav_Dim1, master.density, master.filename " _
& "from master, tests, contactpersons, projects, groups, places, materials, forms " _
& "where master.contactpersonFK=contactpersons.id" _
& "and master.testFK=tests.id and master.projectFK=projects.id " _
& "and master.groupFK=groups.id and master.placeFK=places.id and " _
& "master.materialFK=materials.id and master.formFK=forms.id " _
& +strProject + strGroup + strContactPerson + strTelestisDate + _
+strDate + strPlace + strForm + strMaterial + strTest + "order by master.id"

If cmbForm.SelectedValue = 0 Then
    strForm = ""
Else
    strForm = "and master.formFK=?formFK "
End If

```

Σχήμα 5.17: Το ερώτημα SEARCHSQL για την αναζήτηση συγκεκριμένων εγγραφών και η μορφή της μεταβλητής strForm που χρησιμοποιείται στο ερώτημα αυτό

5.6 Φόρμα επεξεργασίας εγγραφών

Η φόρμα επεξεργασίας (edit) εγγραφών ονομάζεται frmEdit και είναι δυνατόν να εμφανιστεί με δύο τρόπους: Είτε επιλέγοντας κάποια εγγραφή στο dgvDatabase της κύριας φόρμας και κάνοντας κλικ στο κουμπί btnEdit είτε κάνοντας διπλό κλικ στην πρώτη στήλη της εγγραφής στο εργαλείο dgvDatabase. Στο σχήμα 5.18 εμφανίζεται η φόρμα επεξεργασίας εγγραφών, η οποία έχει παρόμοια μορφή με εκείνη της προσθήκης εγγραφών. Στα comboboxes και στα textboxes φορτώνονται οι παράμετροι της εκάστοτε εγγραφής. Εάν ο χρήστης αλλάξει κάποια από αυτές τις παραμέτρους τότε το πρόγραμμα αλλάζει το χρώμα του αντίστοιχου εργαλείου σε κόκκινο, όπως φαίνεται στο σχήμα 5.18 για την παράμετρο Project.

Η φόρτωση των προκαθορισμένων επιλογών των comboboxes γίνεται με τον ίδιο τρόπο που περιγράφηκε στην παράγραφο 5.4. Επιπλέον, προκειμένου να επιλεχτούν οι τιμές των εκάστοτε παραμέτρων, δηλαδή η ενεργή τιμή που θα εμφανίζει κάθε combobox εκτελείται ο κώδικας που εμφανίζεται στο σχήμα 5.19. Το συγκεκριμένο απόσπασμα κώδικα αφορά την εμφάνιση της τιμής της παραμέτρου Place. Αφού πραγματοποιηθεί η σύνδεση με τη βάση δεδομένων rmlab το πρόγραμμα εκτελεί ένα ερώτημα Select για την εύρεση της τιμής της παραμέτρου Place της εγγραφής που έχει επιλεχτεί. Η τιμή αυτή αντιγράφεται σε μία μεταβλητή που ονομάζεται InitialPlace και εξισώνεται με την ιδιότητα SelectedValue του αντικειμένου cmbPlace.

Edit Record Id 22

Common Descriptive Properties

* Project Name:

* Group Name:

* Contact Person:

* Date:

* Place of Origin:

* Specimen Form:

* Axial Dim: mm

* Transverse Dim1: mm

Transverse Dim 2: mm

* Density: g/cm³

* Material Type:

* Discontinuities:

* Inclusions:

* Porosity:

* Preparation:

* Confidence:

* Failure Mode:

* Loading Mode:

Filename:

Comments:

Measured Properties

* Maximum Stress: MPa

Lateral Stress: MPa

* Average Modulus: GPa

Tangent Modulus: GPa

Secant Modulus: GPa

Max Deformation: mm

Energy to Failure: Joule

Cohesion: kPa

Internal Friction: deg

Test Duration: sec

Uniaxial Bending Triaxial Tension Ultrasonic

Cancel Update Record

Σχήμα 5.18: Η φόρμα επεξεργασίας εγγραφών (frmEdit) του προγράμματος RockLab

```

Try
    conn.Open()
Try
    myCommand.Connection = conn
    myCommand.CommandText = "SELECT master.placeFK from master where master.id=?masterid"
    myCommand.Parameters.Add("?masterid", frmMain.CurrentId)
    InitialPlace = myCommand.ExecuteScalar
    cmbPlace.SelectedValue = InitialPlace
Catch myerror As MySqlException
    MsgBox("There was an error Selecting Record's Place of Origin: " & myerror.Message)
End Try
Catch myerror As MySqlException
    MessageBox.Show("Error Connecting to Database: " & myerror.Message)
Finally
    conn.Close()
End Try

```

Σχήμα 5.19: Κώδικας για την επιλογή της κατάλληλης τιμής της παραμέτρου Place

Αφού επιλέξει όλες τις τιμές για τις κοινές παραμέτρους της εγγραφής το πρόγραμμα εκτελεί δύο ερωτήματα στη βάση δεδομένων, προκειμένου να βρει σε πιο πείραμα ανήκει η συγκεκριμένη εγγραφή (testFK) καθώς και το id της εγγραφής αυτής στον πίνακα του πειράματος στο οποίο ανήκει (childFK). Οι τιμές αυτές αποθηκεύονται σε δύο μεταβλητές που ονομάζονται InitialTest και InitialChild αντίστοιχα. Ανάλογα με τον τύπο του πειράματος της εγγραφής, εμφανίζεται η κατάλληλη καρτέλα του εργαλείου Tabcontrol1. Με πολλαπλά ερωτήματα στον πίνακα του πειράματος παίρνουν τις τιμές τους τα textboxes που αντιπροσωπεύουν τις παραμέτρους του εκάστοτε πειράματος. Ο κώδικας για τη φόρτωση της παραμέτρου maximumstress ενός πειράματος τριαξονικής καταπόνησης (triaxial) στο αντίστοιχο πεδίο του προγράμματος (txtMaximumStressTr) εμφανίζεται στο σχήμα 5.20.

```

Try
    conn.Open()
Try
    myCommand.Connection = conn
    myCommand.CommandText = "SELECT triaxial.maximumstress" _
    & "from triaxial where triaxial.id=?triaxialid"
    myCommand.Parameters.Add("?triaxialid", InitialChild)

    InitialMaximumStressTr = myCommand.ExecuteScalar
    txtMaximumStressTr.Text = CStr(InitialMaximumStressTr)
Catch myerror As MySqlException
    MsgBox("error Selecting Record's Maximum Stress: " & myerror.Message)
End Try
Catch myerror As MySqlException
    MessageBox.Show("Error Connecting to Database: " & myerror.Message)
Finally
    conn.Close()
End Try

```

Σχήμα 5.20: Κώδικας για τη φόρτωση της παραμέτρου maximumstress στο αντίστοιχο textbox

Αφού ο χρήστης τροποποιήσει τις τιμές των παραμέτρων της εκάστοτε εγγραφής που επιθυμεί και πατώντας του κουμπι `btnUpdateRecord` το πρόγραμμα εκτελεί στη βάση δεδομένων ένα ερώτημα `Update` και ανανεώνει τις τιμές των παραμέτρων του πίνακα του εκάστοτε πειράματος καθώς και τις τιμές των παραμέτρων του πίνακα `master` με αντίστοιχο τρόπο. Στο σχήμα 5.21 εμφανίζεται ο κώδικας που εκτελείται για την ανανέωση των τιμών των παραμέτρων μίας εγγραφής κάμψης (`bending`). Στα μη υποχρεωτικά πεδία διενεργείται έλεγχος για το αν είναι κενά, και εφόσον είναι η τιμή που αποθηκεύεται στη βάση δεδομένων είναι η τιμή `DBNull.Value` που αντιπροσωπεύει την τιμή του κενού (`null`).

Αντίστοιχα στο σχήμα 5.22 εμφανίζεται ο κώδικας που εκτελείται για την ανανέωση των τιμών των παραμέτρων του πίνακα `master` κατά το πάτημα του κουμπιού `btnUpdateRecord`. Το ερώτημα είναι ένα παραμετρικό `Update` ερώτημα και στις μη υποχρεωτικές παραμέτρους του διενεργείται έλεγχος για το αν έχουν την τιμή του κενού. Στην περίπτωση που κάποια παράμετρος είναι κενή τότε στο αντίστοιχο πεδίο του πίνακα της βάσης δεδομένων αντιστοιχίζεται η τιμή `DBNull.Value` που αντιπροσωπεύει την τιμή του κενού. Εάν όλα τα στοιχεία είναι αποδεκτά και το ερώτημα `Update` εκτελεστεί κανονικά, το πρόγραμμα εμφανίζει ένα μήνυμα στο χρήστη, ώστε να τον πληροφορήσει για την επιτυχία ανανέωσης της εγγραφής.

```
Try
    myCommand.Connection = conn
    myCommand.CommandText = "UPDATE bending SET bendingstrength=?bendingstrength," _
    & "maxdeformation=?maxdeformation, duration=?durationbe " _
    & "WHERE bending.Id=?bendingid"
    myCommand.Parameters.Add("?bendingstrength", Val(txtBendingStrengthBe.Text))
    If txtMaxDeformationBe.Text = "" Then
        myCommand.Parameters.Add("?maxdeformation", DBNull.Value)
    Else
        myCommand.Parameters.Add("?maxdeformation", Val(txtMaxDeformationBe.Text))
    End If
    If txtDurationBe.Text = "" Then
        myCommand.Parameters.Add("?durationbe", DBNull.Value)
    Else
        myCommand.Parameters.Add("?durationbe", Int(Val(txtDurationBe.Text)))
    End If
    myCommand.Parameters.Add("?bendingid", InitialChild)
    myCommand.ExecuteNonQuery()
Catch myerror As MySqlException
    MsgBox("There was an error Updating Current Bending Record: " & myerror.Message)
End Try
```

Σχήμα 5.21: Κώδικας για την ανανέωση των τιμών του πίνακα `bending` μιας επιλεγμένης εγγραφής που αντιστοιχεί σε πείραμα κάμψης


```

Try
myCommand.Connection = conn
myCommand.CommandText = "UPDATE master SET projectFK=?projectFK" _
& ", groupFK=?groupFK, contactpersonFK=?contactpersonFK, date=?date," _
& "placeFK=?placeFK, materialFK=?materialFK, formFK=?formFK," _
& "axialdim=?axialdim, traversedim1=?traversedim1," _
& "traversedim2=?traversedim2, density=?density, discontinuitiesFK=" _
& "?discontinuitiesFK, inclusionsFK=?inclusionsFK, porosityFK=?porosityFK," _
& "preparationFK=?preparationFK, confidenceFK=?confidenceFK," _
& "failureFK=?failureFK, filename=?filename, loadingFK=?loadingFK," _
& "comments=?comments WHERE master.id=?masterid"
myCommand.Parameters.Add("?projectFK", cmbProject.SelectedValue)
myCommand.Parameters.Add("?groupFK", cmbGroup.SelectedValue)
myCommand.Parameters.Add("?contactpersonFK", cmbContactPerson.SelectedValue)
myCommand.Parameters.Add("?date", datePicker.Value.Date)
myCommand.Parameters.Add("?placeFK", cmbPlace.SelectedValue)
myCommand.Parameters.Add("?materialFK", cmbMaterial.SelectedValue)
myCommand.Parameters.Add("?formFK", cmbForm.SelectedValue)
myCommand.Parameters.Add("?axialdim", Val(txtAxialDim.Text))
myCommand.Parameters.Add("?traversedim1", Val(txtTraverseDim1.Text))
If txtTraverseDim2.Text = "" Then
    myCommand.Parameters.Add("?traversedim2", DBNull.Value)
Else
    myCommand.Parameters.Add("?traversedim2", Val(txtTraverseDim2.Text))
End If
myCommand.Parameters.Add("?density", Val(txtDensity.Text))
myCommand.Parameters.Add("?discontinuitiesFK", cmbDiscontinuities.SelectedValue)
myCommand.Parameters.Add("?inclusionsFK", cmbInclusions.SelectedValue)
myCommand.Parameters.Add("?porosityFK", cmbPorosity.SelectedValue)
myCommand.Parameters.Add("?preparationFK", cmbPreparation.SelectedValue)
myCommand.Parameters.Add("?confidenceFK", cmbConfidence.SelectedValue)
myCommand.Parameters.Add("?failureFK", cmbFailure.SelectedValue)
If txtFilename.Text = "" Then
    myCommand.Parameters.Add("?filename", DBNull.Value)
Else
    myCommand.Parameters.Add("?filename", txtFilename.Text)
End If
myCommand.Parameters.Add("?loadingFK", cmbLoading.SelectedValue)
If txtComments.Text = "" Then
    myCommand.Parameters.Add("?comments", DBNull.Value)
Else
    myCommand.Parameters.Add("?comments", txtComments.Text)
End If
myCommand.Parameters.Add("?masterid", frmMain.CurrentId)
myCommand.ExecuteNonQuery()
MsgBox("Record Successfully Updated!", MsgBoxStyle.Exclamation)
Catch myerror As MySqlException
    MsgBox("There was an error Updating Master Table: " & myerror.Message)
End Try

```

Σχήμα 5.22: Κώδικας για την ανανέωση των τιμών του πίνακα master της επιλεγμένης εγγραφής

5.7 Φόρμα διαγραφής εγγραφών

Η φόρμα διαγραφής (delete) εγγραφών ονομάζεται frmDelete και εμφανίζεται όταν ο χρήστης επιλέξει την προς διαγραφή εγγραφή από το εργαλείο dgvDatabase της κύριας φόρμας frmMain και πατήσει το κουμπί btnDeleteRecord της ίδιας φόρμας. Η φόρμα αποτελείται από ένα μήνυμα που προτρέπει το χρήστη για την επιβεβαίωση ή την ακύρωση της διαγραφής και δύο κουμπιά btnCancel και btnDelete. Με κλικ στο κουμπί btnCancel η διαγραφή ακυρώνεται και η φόρμα αποκρύπτεται. Αντίθετα με κλικ στο κουμπί btnDelete το πρόγραμμα διαγράφει την επιλεγμένη εγγραφή διενεργώντας δύο ερωτήματα Delete στον πίνακα του πειράματος στο οποίο ανήκει η εγγραφή και στον πίνακα master. Ο κώδικας που εκτελείται κατά τη διαγραφή μιας εγγραφής που αντιστοιχεί σε πείραμα προσδιορισμού ταχυτήτων διάδοσης (ultrasonic) εμφανίζεται στο σχήμα 5.23. Στο σχήμα 5.24 εμφανίζεται η φόρμα frmDelete του προγράμματος RockLab

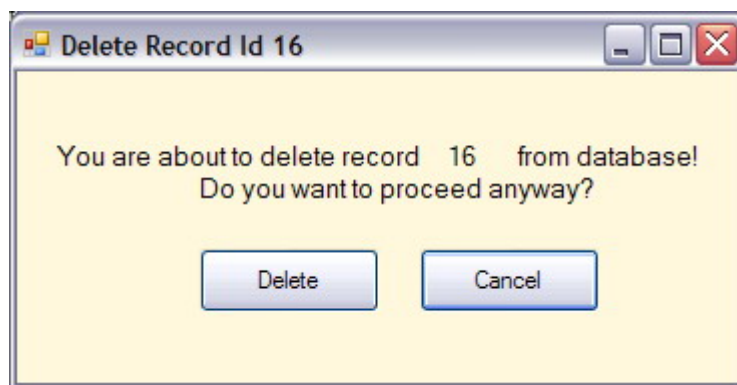
```
Try
    conn.Open()
    Try
        myCommand.Connection = conn
        myCommand.CommandText = "DELETE FROM ultrasonic WHERE Id=?Id"
        myCommand.Parameters.Add("?Id", InitialChild)

        myCommand.ExecuteNonQuery()
    Catch myerror As MySqlException
        MsgBox("error deleting Record from Ultrasonic Table" & myerror.Message)
    End Try
Catch myerror As MySqlException
    MessageBox.Show("Error Connecting to Database: " & myerror.Message)
Finally
    conn.Close()
End Try

Try
    conn.Open()
    Try
        myCommand.Connection = conn
        myCommand.CommandText = "DELETE FROM master WHERE Id=?Idm"
        myCommand.Parameters.Add("?Idm", frmMain.CurrentId)

        myCommand.ExecuteNonQuery()
        Me.Close()
    Catch myerror As MySqlException
        MsgBox("error deleting Record from Master Table" & myerror.Message)
    End Try
Catch myerror As MySqlException
    MessageBox.Show("Error Connecting to Database: " & myerror.Message)
Finally
    conn.Close()
End Try
```

Σχήμα 5.23: Κώδικας που εκτελείται κατά τη διαγραφή μιας ultrasonic εγγραφής

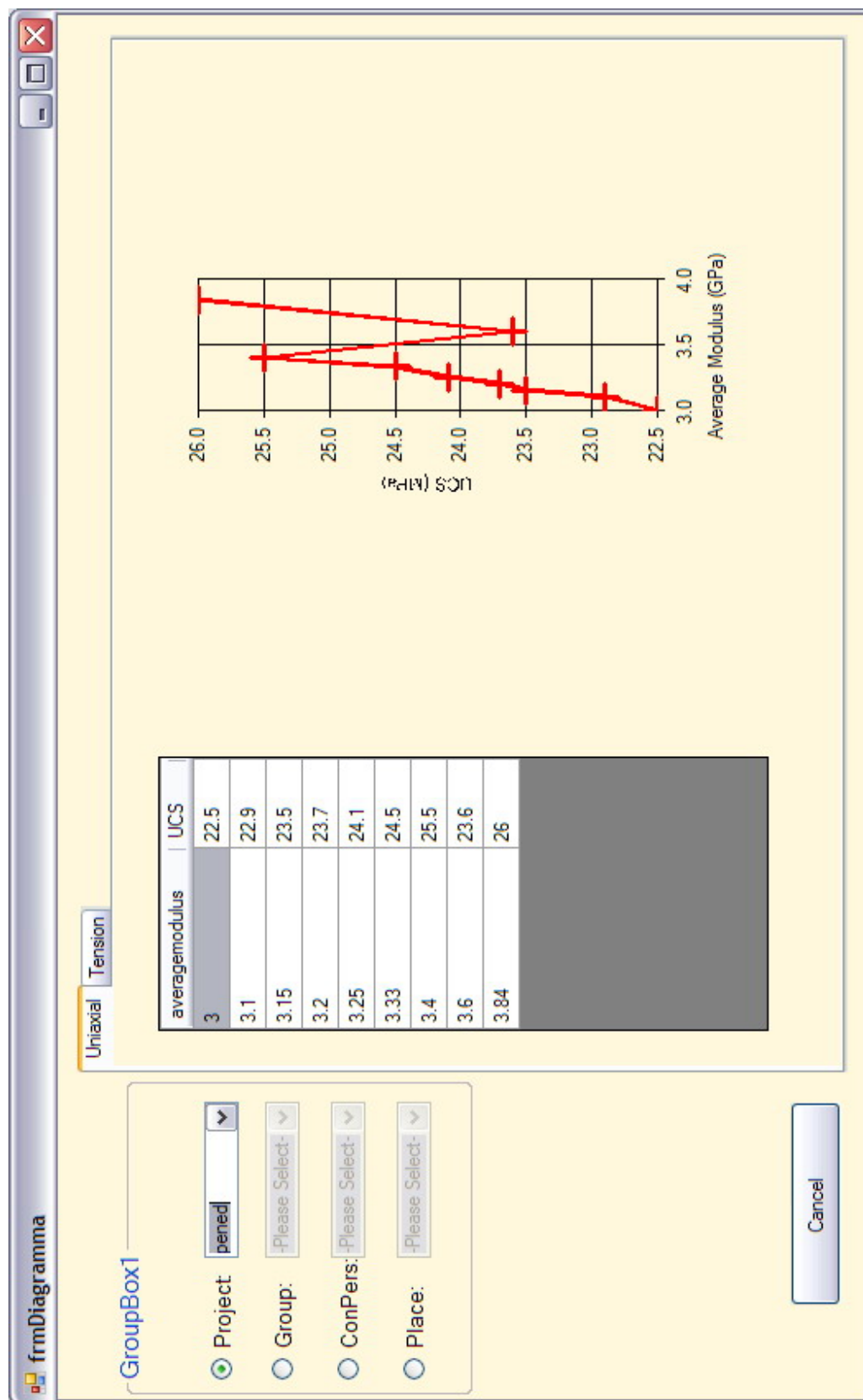


Σχήμα 5.24: Η φόρμα διαγραφής εγγραφής frmDelete του προγράμματος RockLab

5.8 Φόρμα σχεδιασμού διαγραμμάτων

Η φόρμα σχεδιασμού διαγραμμάτων (σχήμα 5.25) ονομάζεται frmChartPlot και εμφανίζεται όταν ο χρήστης κάνει κλικ στο κουμπί btnChartPlot της κύριας φόρμας frmMain. Για το σχεδιασμό των διαγραμμάτων χρησιμοποιήθηκε ένα εργαλείο της Visual Basic .NET που ονομάζεται MSChart. Το πρόγραμμα είναι δυνατόν να σχεδιάσει δύο τύπους διαγραμμάτων. Ο πρώτος τύπος είναι διάγραμμα αντοχής σε μονοαξονική θλίψη (UCS) συναρτήσεως του μέτρου ελαστικότητας (averagemodulus) και αφορά τις εγγραφές που αντιστοιχούν σε πειράματα μονοαξονικής θλίψης (uniaxial). Ο δεύτερος τύπος είναι διάγραμμα αντοχής σε κάμψη (tensilestrength) συναρτήσεως της ενέργειας που απαιτείται για αστοχία (energytofailure) και αφορά τις εγγραφές που αντιστοιχούν σε πειράματα κάμψης (tension).

Από τα comboboxes που βρίσκονται στο αριστερό τμήμα της φόρμας, είναι δυνατόν να επιλέξει ο χρήστης κάποιο συγκεκριμένο κριτήριο που θα πρέπει να πληρούν οι εγγραφές για τις οποίες πρόκειται να σχεδιαστεί το διάγραμμα (πχ. σχήμα 5.25 ProjectName=pened). Για να σχεδιαστεί ένα διάγραμμα με το αντικείμενο MSChart αρκεί να οριστεί ένας πίνακας δύο στηλών όπου η πρώτη στήλη αντιπροσωπεύει τις τιμές x και η δεύτερη τις τιμές y της γραφικής παράστασης. Το πρόγραμμα εκτελεί κατάλληλα ερωτήματα Select στους πίνακες uniaxial και tension της βάσης δεδομένων rmlab και εμφανίζει τα αποτελέσματα σε ένα αντικείμενο DataGridView. Στη συνέχεια οι τιμές του αντικειμένου αυτού δημιουργούν έναν πίνακα δύο στηλών από τον οποίο αντλεί τα δεδομένα του το αντικείμενο MSChart και σχεδιάζει τη γραφική παράσταση. Στο σχήμα 5.26 εμφανίζεται αναλυτικά ο κώδικας που εκτελείται για το σχεδιασμό γραφικής παράστασης που αφορά εγγραφές μονοαξονικής θλίψης.



Σχήμα 5.25: Η φόρμα σχεδιασμού διαγραμμάτων (frmChartPlot) του προγράμματος RockLab


```

Try
    conn.Open()
Try
    myCommand.Connection = conn
    myCommand.CommandText = "select u.averagemodulus, u.UCS from uniaxial as u," _
    &"master where u.id = master.childFK and master.contactpersonFK = ?contactpersonFK" _
    &"order by u.averagemodulus"
    myCommand.Parameters.Add("?contactpersonFK", cmbContactPerson.SelectedValue)
    myAdapter.SelectCommand = myCommand
    myAdapter.Fill(myDataUniaxial)
    dgvUniaxial.ReadOnly = True
    dgvUniaxial.DataSource = myDataUniaxial
    dgvUniaxial.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.DisplayedCells

    Dim UniaxialArray(dgvUniaxial.RowCount - 1, dgvUniaxial.ColumnCount - 1) As Single
    Dim i As Integer
    Dim j As Integer

    For i = 0 To dgvUniaxial.RowCount - 1
        For j = 0 To dgvUniaxial.ColumnCount - 1
            UniaxialArray(i, j) = dgvUniaxial.Item(j, i).Value
        Next j
    Next i

    AxMSChart1.Plot.Axis(MSChart20Lib.VtChAxisId.VtChAxisIdX).AxisTitle.Text = "Average Modulus (GPa)"
    AxMSChart1.Plot.Axis(MSChart20Lib.VtChAxisId.VtChAxisIdY).AxisTitle.Text = "UCS (MPa)"
    AxMSChart1.Plot.SeriesCollection(1).SeriesMarker.Show = True
    AxMSChart1.ChartData = UniaxialArray

Catch myerror As MySqlException
    MsgBox("There was an error reading from the database: " & myerror.Message)
End Try
Catch myerror As MySqlException
    MessageBox.Show("Error Connecting to Database: " & myerror.Message)
Finally
    conn.Close()
End Try

```

Σχήμα 5.26: Κώδικας που εκτελείται για το σχεδιασμό διαγράμματος αντοχής σε μονοαξονική θλίψη συναρτήσει του μέσου μέτρου ελαστικότητας

ΚΕΦΑΛΑΙΟ 6

ΣΥΜΠΕΡΑΣΜΑΤΑ – ΠΡΟΤΑΣΕΙΣ

6.1 Σύνοψη

Στη μεταπτυχιακή αυτή διατριβή περιγράφεται ο σχεδιασμός και η ανάπτυξη μιας βάσης δεδομένων για την ομαδοποίηση και ανάλυση πειραματικών δεδομένων της Μηχανικής Πετρωμάτων καθώς και η υλοποίηση λογισμικού το οποίο είναι δομημένο με βάση τη φιλοσοφία πελάτη/διακομιστή και με το οποίο απλοποιείται ο χειρισμός της βάσης και υλοποιούνται πολύπλοκες διαδικασίες όπως ο σχεδιασμός διαγραμμάτων με βάση ομάδες πειραματικών δεδομένων.

Τα πλεονεκτήματα της αρχιτεκτονικής πελάτη/διακομιστή του συστήματος του λογισμικού και της βάσης δεδομένων που κατασκευάστηκαν είναι:

- Ο διακομιστής (MySQL server) και κατ' επέκταση η βάση δεδομένων μπορούν να μεταφερθούν σε οποιοδήποτε εμπορικό λειτουργικό σύστημα (π.χ Unix, Linux κλπ.).
- Η εφαρμογή RockLab υλοποιήθηκε σε γλώσσα προγραμματισμού Visual Basic .NET 2005 και μπορεί να λειτουργήσει σε οποιοδήποτε λειτουργικό σύστημα αρκεί να υπάρχει εγκατεστημένο το .NET Framework.

Η βάση δεδομένων rmlab που αναπτύχθηκε για τις ανάγκες της διατριβής αυτής αποτελείται από 20 πίνακες (έξι κύριους και 14 υπό-πίνακες) και υλοποιήθηκε στο σύστημα βάσεων δεδομένων MySQL που αποτελεί ένα από τα πιο αξιόπιστα και σύγχρονα συστήματα αυτού του είδους.

Στον πίνακα master αποθηκεύονται όλες οι κοινές παράμετροι των πειραμάτων ενώ στους υπόλοιπους πέντε κύριους πίνακες που ονομάζονται uniaxial, bending, triaxial, tension και ultrasonic αποθηκεύονται οι μη κοινές παράμετροι του εκάστοτε πειράματος. Οι πίνακες αυτοί συνδέονται με σχέση «ένα προς πολλά» με τον πίνακα master μέσω του πεδίου childFK αυτού. Οι 14 υποπίνακες αντιπροσωπεύουν μερικές από τις κοινές παραμέτρους των πειραμάτων και συνδέονται με σχέση «ένα προς πολλά» με το αντίστοιχο πεδίο του πίνακα master.

Το πρόγραμμα RockLab αποτελείται από έξι φόρμες ενώ για τον προγραμματισμό του απαιτήθηκαν συνολικά 7800 γραμμές κώδικα.

Τα πλεονεκτήματα από τη χρήση ενός λογισμικού αυτού του είδους, σε καθημερινή βάση, σε εργαστήρια όπου διενεργούνται πληθώρα δοκιμών, είναι:

- Άμεση αρχειοθέτηση και κωδικοποίηση των πειραματικών δεδομένων ώστε να είναι δυνατή η εύκολη διαχείριση τους.
- Άμεση αρχειοθέτηση των πρωτογενών αρχείων συλλογής δεδομένων σε μορφή ASCII ή σε μορφή λογιστικού φύλλου ώστε να είναι δυνατή η άμεση πρόσβαση στα πρωτογενή στοιχεία όποτε ζητηθεί.
- Άμεση αρχειοθέτηση τυχόν φωτογραφικού υλικού που σχετίζεται με την εκάστοτε δοκιμή.
- Συγκριτική αξιολόγηση των μετρήσεων για διάφορες κατηγορίες δεδομένων.
- Άμεση εκτύπωση των πρωτογενών φύλλων δοκιμής

6.2 Προτάσεις

Το λογισμικό που αναπτύχθηκε στα πλαίσια της διατριβής αυτής δίδεται σαν εκτελέσιμο πρόγραμμα (stand-alone) δηλαδή χρειάζεται εγκατάσταση σε κάποιο λειτουργικό σύστημα για να είναι δυνατή η εκτέλεση του. Οι σύγχρονες τάσεις επιβάλλουν τη κατασκευή προγραμμάτων με βάση πρωτόκολλα διαδικτύου (web-based) δηλαδή προγραμμάτων που εκτελούνται μέσα από μία ιστοσελίδα του διαδικτύου και δεν χρειάζονται εγκατάσταση. Προτείνεται λοιπόν η κατάλληλη μετατροπή του λογισμικού σε web-based τύπο μέσω των σύγχρονων τεχνολογιών που παρέχει το πακέτο Visual Studio .NET και πιο συγκεκριμένα μέσω τεχνολογιών ASP, XML, VBScript κλπ., ώστε να είναι δυνατή η ευρύτερη χρήση του μέσω διαδικτύου.

Κατά το σχεδιασμό της βάσης δεδομένων rmlab περιλήφθηκαν οι πιο σημαντικές δοκιμές της Μηχανικής Πετρωμάτων και οι κυριότερες παράμετροι που διέπουν αυτές. Κατά συνέπεια προτείνεται η κατάλληλη επέκταση της βάσης δεδομένων και του λογισμικού που αναπτύχθηκαν ώστε να συμπεριληφθούν όλες οι δοκιμές της Μηχανικής Πετρωμάτων καθώς και όσο το δυνατόν περισσότερες παράμετροι αυτών.

Η φόρμα σχεδιασμού διαγραμμάτων του προγράμματος RockLab είναι προγραμματισμένη ώστε να σχεδιάζει διαγράμματα με βάση συγκεκριμένες παραμέτρους των δύο (από τι πέντε) ομάδων δοκιμών. Προτείνεται λοιπόν η κατάλληλη μετατροπή του προγράμματος ώστε να είναι δυνατή η σχεδίαση διαγραμμάτων με βάση όλες τις

παραμέτρους όλων των ομάδων των πειραμάτων και σύμφωνα με τις απαιτήσεις του εκάστοτε χρήστη.

Η βάση δεδομένων αποθηκεύει τα ονόματα των πρωτογενών αρχείων που αντιστοιχούν σε κάθε δοκιμή. Τα ονόματα αυτά των αρχείων είναι συνήθως μεγάλα και ανομοιομορφα με αποτέλεσμα να χάνεται πολύτιμος αποθηκευτικός χώρος στη βάση δεδομένων. Κατά συνέπεια προτείνεται η εύρεση ενός κατάλληλου συστήματος κωδικοποίησης των ονομάτων των αρχείων με στόχο την καλύτερη λειτουργία του συστήματος βάση δεδομένων – λογισμικό αλλά και τη σωστότερη ταξινόμηση και αποθήκευση των αρχείων αυτών στο σκληρό δίσκο του διακομιστή.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Deitel, H., J. Deitel & T.Nieto (2002). *Visual Basic .NET Προγραμματισμός*, Εκδόσεις Μ. Γκιούρδας, Αθήνα.
2. Σπινέλλης, Δ. (2004). *Υλοποίηση Διεπαφών σε Visual Basic*, <http://www.dmst.aueb.gr/dds/c3/vb/indexw.htm>.
3. Vick, P. (2004). *The Visual Basic .NET Programming Language*. Εκδόσεις Addison Wesley, Boston.
4. Αγιουτάντης, Ζ. (2002). *Στοιχεία Γεωμηχανικής, Μηχανικής Πετρωμάτων*, Εκδόσεις ΙΩΝ, Αθήνα.
5. Στειακάκης, Ε. (2004). *Σημειώσεις Τεχνικής Γεωλογίας - Εδαφομηχανικής*, Πολυτεχνείο Κρήτης, Χανιά.
6. Kofler, M. (2004). *The Definitive Guide to MySQL*, Εκδόσεις Apress, Berkeley.
7. Date, C. (1995). *Εισαγωγή στα συστήματα βάσεων δεδομένων*, Εκδόσεις Κλειδάριθμος, Αθήνα.