



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

	<u>Μεταπτυχιακή Διατριβή</u>
Τίτλος Διατριβής	E-Learning SQLite: Ένα σύστημα διδασκαλίας για εκμάθηση βασικών εννοιών της SQLite.
Όνοματεπώνυμο Φοιτητή	Μπερκέτης Γρηγόριος
Πατρώνυμο	Πέτρος
Αριθμός Μητρώου	ΜΠΠΛ 17031
Επιβλέπων Καθηγητής	Αλέπης Ευθύμιος, Επίκουρος Καθηγητής

Ημερομηνία Παράδοσης **Δεκέμβριος 2019**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Ευθύμιος Αλέπης
Επίκουρος Καθηγητής

Γεώργιος Τσιχριντζής
Καθηγητής

Μαρία Βίββου
Καθηγητής

Ευχαριστίες

Για την υλοποίηση της διπλωματικής αυτής εργασίας θεωρώ υποχρέωσή μου να ευχαριστήσω τον Καθηγητή του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς κ. Ευθύμιο Αλέπη για την εμπιστοσύνη που μου έδειξε, και για τον συμβουλευτικό και καθοδηγητικό του ρόλο σε όλη την διάρκεια υλοποίησης του Project.

Ευχαριστώ ιδιαίτερα όλους τους καθηγητές του προγράμματος μεταπτυχιακών σπουδών Πληροφορικής του Πανεπιστημίου Πειραιώς που με δίδαξαν σημαντικά πράγματα στο κλάδο της Πληροφορικής.

Τέλος, θα ήθελα να εκφράσω τις βαθύτατες ευχαριστίες μου και την απέραντη ευγνωμοσύνη μου στην Οικογένειά μου, που τόσα χρόνια στηρίζει τις προσπάθειές μου και με καθοδηγεί όταν χρειάζεται.

Περιεχόμενα

Ευχαριστίες.....	2
1. Περίληψη	5
2. Abstract.....	6
3. Εισαγωγή.....	7
4. E-Learning	7
4.1 Ορισμός.....	7
4.2 Βασικά στοιχεία ενός προγράμματος e-Learning.....	8
4.3 Μορφές e-Learning	9
4.4 Τεχνολογίες που χρησιμοποιούνται στην εκπαίδευση e-Learning.....	10
5. Δομημένη Γλώσσα Ερωτημάτων - (Structured Query Language - SQL).....	10
5.1 Ορισμός.....	10
5.2 Ιστορική Αναδρομή.....	11
5.3 Αρχιτεκτονική.....	11
5.4 Δημοφιλείς Επεκτάσεις της SQL	12
5.4.1 Microsoft SQL Server & T-SQL (Transact SQL).....	12
5.4.2 Oracle Server – PL/SQL (Procedural Language/Structured Query Language).....	13
5.4.3 MySQL	13
5.4.4 PostgreSQL.....	14
5.4.5 SQLite	15
6. Η Ιστοσελίδα SQLite e-learning.....	16
6.1 Σκοπός της Ιστοσελίδας.....	16
6.2 Προαπαιτούμενα παρακολούθησης του e-Learning	17
6.3 Χρήση της Ιστοσελίδας e-Learning SQLite	17
6.4 Λειτουργικότητα Ιστοσελίδας e-Learning SQLite	17
6.5 Διδακτικές Ενότητες Ιστοσελίδας e-Learning SQLite	18
6.5.1 Ενότητα 1: Γενική Εικόνα (OVERVIEW)	18
6.5.2 Ενότητα 2: Κατέβασμα & Εγκατάσταση (DOWNLOAD & INSTALLATION).....	19
6.5.3 Ενότητα 3: Εντολές & Σύνταξη (COMMANDS & SYNTAX).....	22
6.5.4 Ενότητα 4: Τύποι Δεδομένων (DATA TYPES).....	26
6.5.5 Ενότητα 5: CREATE & DELETE DATABASE (ΔΗΜΙΟΥΡΓΙΑ & ΔΙΑΓΡΑΦΗ ΒΑΣΗΣ).....	29
6.5.6 Ενότητα 6: CREATE Table (ΔΗΜΙΟΥΡΓΙΑ Πίνακα).....	31
6.5.7 Ενότητα 7: DROP Table (ΔΙΑΓΡΑΦΗ - ΚΑΤΑΣΤΡΟΦΗ του Πίνακα).....	32
6.5.8 Ενότητα 8: INSERT (ΕΙΣΑΓΩΓΗ Δεδομένου (ων) σε Πίνακα)	32
6.5.9 Ενότητα 9: SELECT (ΕΠΙΛΟΓΗ – ΕΥΡΕΣΗ Δεδομένου(ων) σε Πίνακα).....	33
6.5.10 Ενότητα 10: Operators (Τελεστές)	33
6.5.11 Ενότητα 11: WHERE (ΟΠΟΥ).....	36
6.5.12 Ενότητα 12: UPDATE (ΕΝΗΜΕΡΩΣΗ Πίνακα).....	36

6.5.13 Ενότητα 13: DELETE (ΔΙΑΓΡΑΦΗ Δεδομένου (ων) Πίνακα).....	37
6.5.14 Ενότητα 14: AND (ΚΑΙ)	38
6.5.15 Ενότητα 15: OR (Η)	39
6.5.16 Ενότητα 16: LIKE (‘ΟΜΟΙΟ ΜΕ’)	39
6.5.17 Ενότητα 17: LIMIT (ΟΡΙΟ).....	40
6.5.18 Ενότητα 18: MIN (ΕΛΑΧΙΣΤΟ) - MAX (ΜΕΓΙΣΤΟ)	42
6.5.19 Ενότητα 19: SUM (ΑΡΙΘΜΗΤΙΚΟ ΑΘΡΟΙΣΜΑ).....	43
6.5.20 Ενότητα 20: DISTINCT (ΔΙΑΚΡΙΤΟ) & COUNT (ΠΛΗΘΟΣ)	43
6.5.21 Ενότητα 21: AVG (ΜΕΣΟΣ ΟΡΟΣ)	45
6.5.22 Ενότητα 22: ORDER BY (ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΣΕΙΡΑ)	46
6.5.23 Ενότητα 23: GROYP BY (ΟΜΑΔΟΠΟΙΗΣΗ).....	47
6.5.24 Ενότητα 24: HAVING ().....	48
6.5.25 Ενότητα 25: JOIN (ΕΝΩΣΗ ΠΙΝΑΚΩΝ).....	48
6.6 Ανασκόπηση πεδίου	52
7. Παρουσίαση και χρήση εφαρμογής.....	54
7.1 Εγχειρίδιο χρήσης εφαρμογής.....	54
8. Αρχιτεκτονική Συστήματος	64
8.1 Εργαλεία που χρησιμοποιήθηκαν	64
9. Συμπεράσματα και μελλοντικές επεκτάσεις	65
10. Βιβλιογραφία.....	67
11. Ιστοσελίδες	68
12. Παραρτήματα	69

1. Περίληψη

Εδώ και αρκετά χρόνια η ηλεκτρονική μάθηση ή αλλιώς e-learning έχει βοηθήσει εκατομμύρια ανθρώπους να αποκτήσουν με πολύ εύκολο τρόπο πρόσβαση στην γνώση καταργώντας τον χώρο και το χρόνο της συμβατικής διδασκαλίας. Το μόνο που χρειάζεται ο χρήστης ο οποίος επιδιώκει να μορφωθεί είναι η πρόσβαση στο διαδίκτυο μέσω ενός ηλεκτρονικού μέσου, είτε αυτό είναι ηλεκτρονικός υπολογιστής, είτε οποιαδήποτε φορητή συσκευή. Η διδασκαλία εξ αποστάσεως ή αλλιώς ηλεκτρονική διδασκαλία εντελώς ευέλικτη και προσαρμόσιμη στις ιδιαιτερότητες κάθε χρήστη έδωσε την δυνατότητα σε άτομα που δεν είχαν τη δυνατότητα να φοιτήσουν σε κάποιο εκπαιδευτικό ίδρυμα να διευρύνουν τις γνώσεις τους και σε πολλές περιπτώσεις να αλλάξουν τις ζωές τους.

Η συγκεκριμένη πτυχιακή εργασία αποσκοπεί στη δημιουργία ενός προγράμματος ηλεκτρονικής εκμάθησης e-Learning με αντικείμενο την SQLite, ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων που χαρακτηρίζεται απόλυτα από την φορητότητα και την ευκολία ως προς την χρήση του. Θα παρουσιαστούν αναλυτικά οι λειτουργικές και μη απαιτήσεις του συστήματος αλλά και η ίδια η ιστοσελίδα σε λειτουργία. Θα αναλυθεί η αρχιτεκτονική της ιστοσελίδας και οι τεχνολογίες που χρησιμοποιήθηκαν για την κατασκευή της, θα περιγραφεί το εγχειρίδιο χρήστη και τέλος θα αναφερθούμε σε δυνατότητες μελλοντικής επέκτασης και εξέλιξης του προγράμματος εκμάθησης.

2. Abstract

For many years now e-learning has helped millions of people gain access to knowledge with such an easy way by almost eliminating the space and time of conventional teaching. The only thing needed from someone seeking to learn is access to the internet through an electronic medium, whether is a desktop computer or a mobile device. Distance learning also known as e-learning is completely flexible and adaptable to the particularities of each user, enabling individuals who have not been able to study in any educational institution or acquire education by any means to expand their knowledge and in many cases change their lives.

This undergraduate thesis aims to create an e-learning program for SQLite, which is a relational database management system that is fully characterized by its portability and ease of use. There will be presented the functional and non-functional requirements of the system and the website itself in detail. It will be analyzed the website architecture and the technologies used to build it, the user manual will be described, and the possibilities of future extension and development of the tutorial will be discussed.

3. Εισαγωγή

Πρόσβαση στην γνώση μέσα από λίγα μόνο «κλικ» του υπολογιστή ή αλλιώς e-learning. Μια τάση που ξεκίνησε πριν αρκετά χρόνια με αρκετούς πολέμιους και αμφισβητίες ως προς το κατά πόσο μπορεί το ηλεκτρονικό μέσο να αντικαταστήσει τον διδάσκοντα έγινε βασικό εργαλείο στις μέρες μας στα χέρια ανθρώπων που ήθελα να μορφωθούν, εταιρειών που ήθελαν να εξελίξουν τις γνώσεις των υπαλλήλων τους ακόμα και καταρτισμένων ανθρώπων που ήθελα να γίνουν πιο ελκυστικοί στην αγορά εργασίας.

Σκοπός του προγράμματος μας είναι η εκμάθηση της SQLite η οποία αποτελεί μια βιβλιοθήκη λογισμικού που μας παρέχει ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Αντίθετα με άλλες βιβλιοθήκες και παραλλαγές της SQL η SQLite έχει αποκτήσει τεράστια δημοφιλία εξαιτίας της ευκολίας εγκατάστασης της(λειτουργεί χωρίς διακομιστές), του μικρού της μεγέθους, της μηδενικής διαμόρφωσης και της ευκολίας μεταφοράς μεταξύ διαφορετικών λειτουργικών συστημάτων. Ακολουθώντας το συγκεκριμένο πρόγραμμα άτομα με μηδενική εμπειρία σε SQLite ή ακόμα και άτομα με υπόβαθρο σε κάποια από τις παραλλαγές της SQL θα μπορέσουν να αποκτήσουν τις βασικές γνώσεις γύρω από την λειτουργία της SQLite.

Προσπαθήσαμε να δημιουργήσουμε ένα πρόγραμμα που θα είναι φιλικό προς όλους τους χρήστες, απλό και κατανοητό στην χρήση του καθώς και εύκολα προσβάσιμο στον μέσο χρήστη. Κάθε χρήστης μετά την υποχρεωτική εγγραφή του στην ιστοσελίδα μπορεί να περιηγηθεί στις ενότητες του μαθήματος, να μελετήσει την θεωρία και να εξασκηθεί στις ασκήσεις που διαθέτει κάθε διδακτική ενότητα. Κάθε ενότητα διαθέτει ασκήσεις πολλαπλής επιλογής και συμπλήρωσης κενού έτσι ώστε ο χρήστης να δοκιμάζει τις γνώσεις που αποκόμισε. Αναλυτικά για κάθε χρήστη υπάρχει πίνακας βαθμολογιών όπου εμφανίζει το βαθμό επιτυχίας που είχε σε κάθε διδακτική ενότητα αλλά και συνολικά σε όλο το μάθημα ώστε να βλέπει τις αδυναμίες του και να ανατρέχει στις ενότητες για επανάληψη.

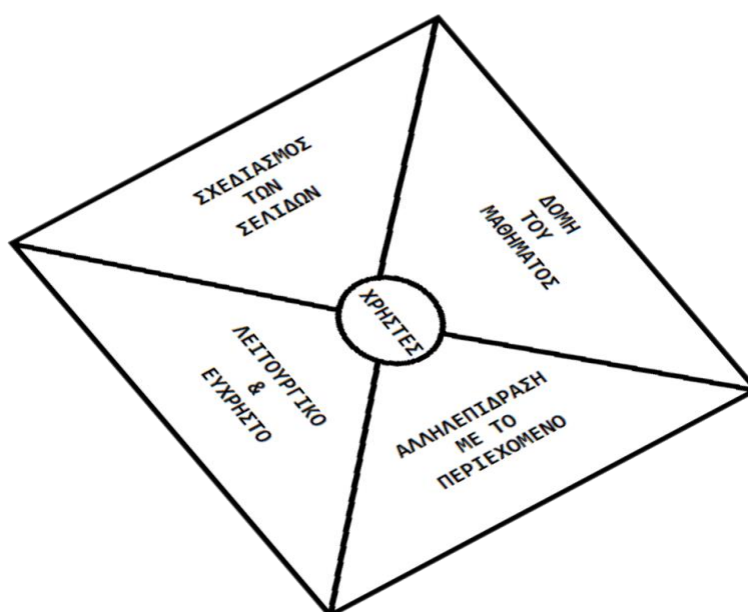
4. E-Learning

4.1 Ορισμός

Ένα εκπαιδευτικό σύστημα βασισμένο στην επίσημη διδασκαλία αλλά με τη βοήθεια ηλεκτρονικών πόρων είναι γνωστό ως ηλεκτρονική μάθηση. Εν αντιθέσει με την συμβατική διδασκαλία που ως βάση της έχει την αίθουσα εκμάθησης, και την ταυτόχρονη συνύπαρξη διδάσκοντα και διδασκόμενων, η ηλεκτρονική μάθηση απαιτεί μόνο την χρήση ηλεκτρονικών υπολογιστών και του Διαδικτύου για να επιτύχει το ίδιο αποτέλεσμα. Η ηλεκτρονική μάθηση μπορεί επίσης να χαρακτηριστεί ως μια (δια)-δικτυακή μεταβίβαση δεξιοτήτων, γνώσεων και παροχή εκπαίδευσης που απευθύνεται σε μεγάλο αριθμό αποδεκτών στην ίδια ή διαφορετική χρονική στιγμή. Αρκετά χρόνια πριν, η προσέγγιση αυτή δεν έγινε δεκτή ολόψυχα, διότι θεωρήθηκε ότι το σύστημα αυτό δεν διέθετε το ανθρώπινο στοιχείο που απαιτείται για την μεταλαμπάδευση της γνώσης. Η ραγδαία εξέλιξη της τεχνολογίας όμως αποτέλεσε τη βάση μιας επανάστασης η οποία με την πάροδο του χρόνου και υποβοηθούμενη από την εποχή των φορητών έξυπνων συσκευών οδήγησε ολοένα και περισσότερους εκπαιδευτικούς να αναθεωρήσουν τις συμβατικές και οπισθοδρομικές αντιλήψεις τους. Τα βιβλία αντικαθίστανται σταδιακά από ηλεκτρονικά εκπαιδευτικά υλικά όπως οπτικοί δίσκοι, usb sticks, ταμπλέτες, βιβλία σε ψηφιακή μορφή ανεβασμένα στο cloud κ.α. Η γνώση μπορεί επίσης να μοιραστεί μέσω του Διαδικτύου, το οποίο είναι προσβάσιμο 24 ώρες το 24ωρο, οπουδήποτε και οποτεδήποτε. Η ηλεκτρονική μάθηση έχει αποδειχθεί ότι είναι το καλύτερο μέσο ακόμα και στον εταιρικό τομέα, ειδικά όταν τα προγράμματα κατάρτισης διεξάγονται από πολυεθνικές εταιρείες για επαγγελματίες σε όλο τον κόσμο και οι εργαζόμενοι μπορούν να αποκτήσουν σημαντικές δεξιότητες, ενώ βρίσκονται σε μια αίθουσα συνεδριάσεων ή στον χώρο εργασίας τους οπουδήποτε ανά την υφήλιο. Τα σχολεία που χρησιμοποιούν τεχνολογίες ηλεκτρονικής μάθησης θα αποκτούν πλεονέκτημα και θα είναι πάντα ένα βήμα μπροστά από εκείνα που εξακολουθούν να έχουν την παραδοσιακή προσέγγιση της συμβατικής διδασκαλίας. Χωρίς

αμφιβολία, είναι εξίσου σημαντικό να προωθηθεί η έννοια της μη ηλεκτρονικής διδασκαλίας με τη βοήθεια βιβλίων και διαλέξεων, αλλά η σπουδαιότητα και η αποτελεσματικότητα της μάθησης με βάση την τεχνολογία δεν μπορεί να θεωρηθεί αμελητέα ή να αγνοηθεί εντελώς. Πιστεύεται ότι ο ανθρώπινος εγκέφαλος μπορεί εύκολα να θυμηθεί και να συσχετιστεί με αυτό που βλέπει και ακούει μέσω κινούμενων εικόνων ή βίντεο. Έχει επίσης βρεθεί ότι η οπτικοποίηση, εκτός από το να κρατά την προσοχή του μαθητή, διατηρείται επίσης από τον εγκέφαλο για μεγαλύτερες περιόδους. Διάφοροι τομείς, όπως η γεωργία, η ιατρική, η εκπαίδευση, οι υπηρεσίες, οι επιχειρήσεις και οι κυβερνητικές ρυθμίσεις, έχουν αρχίσει να προσαρμόζονται στην έννοια της ηλεκτρονικής μάθησης που βοηθά στην πρόοδο ενός έθνους.

4.2 Βασικά στοιχεία ενός προγράμματος e-Learning



3.Σύνδεση Βασικών Στοιχείων Ηλεκτρονικής Μάθησης

Υπάρχουν πέντε βασικά στοιχεία ηλεκτρονικής μάθησης που είναι απαραίτητα για όλα τα επιτυχημένα online μαθήματα. Το παραπάνω διάγραμμα απεικονίζει τον τρόπο σύνδεσης αυτών των στοιχείων. Κάθε στοιχείο ηλεκτρονικής μάθησης παίζει τον δικό του σημαντικό ρόλο στο σχεδιασμό ενός διαδικτυακού μαθήματος. Μεταξύ όλων των στοιχείων όμως κανένα δεν παίζει μεγαλύτερο ρόλο από το κοινό δηλαδή τους ΧΡΗΣΤΕΣ.

ΧΡΗΣΤΕΣ: Από τον σχεδιασμό μέχρι και την υλοποίηση οι χρήστες είναι ο καθοριστικός παράγοντας στην ανάπτυξη μιας εφαρμογής ηλεκτρονικής μάθησης. Σκοπός είναι η δημιουργία ενός προγράμματος που θα καλύπτει τις απαιτήσεις τους, τις μαθησιακές τους ικανότητες, την πρόσβαση που έχουν σε υλικολογισμικό ηλεκτρονικών υπολογιστών καθώς και το περιβάλλον στο οποίο θα μπορεί ο χρήστης να πραγματοποιήσει την εκμάθηση(σπίτι, σχολείο, επαγγελματικό περιβάλλον).

ΔΟΜΗ ΤΟΥ ΜΑΘΗΜΑΤΟΣ: Με τον όρο δομή του μαθήματος αναφερόμαστε στο πως ένα μάθημα έχει σχεδιαστεί για την ηλεκτρονική μάθηση. Παίζει κρίσιμο ρόλο στον τρόπο με τον οποίο οι χρήστες μαθαίνουν το υλικό η δομή ενός μαθήματος. Η ομαδοποίηση του περιεχομένου σε λογικές ενότητες, η διάρθρωση των πληροφοριών σε μικρά κομμάτια διευκολύνει πολύ τον χρήστη να παρακολουθεί πιο εύκολα την ροή του προγράμματος και να αφομοιώνει την πληροφορία που του σερβίρεται.

ΣΧΕΔΙΑΣΜΟΣ ΤΩΝ ΣΕΛΙΔΩΝ: Όπως ο χαρισματικός εκπαιδευτικός που με τη ροή του λόγου μπορεί να μαγνητίσει τους εκπαιδευόμενους του να έχουν μάτια και αυτιά μόνο για αυτόν μέσα στην αίθουσα διδασκαλίας έτσι και ο σχεδιασμός της σελίδας ενός διαδικτυακού μαθήματος είναι κρίσιμης σημασίας για τη διαδικασία μάθησης. Η ισορροπία μεταξύ κειμένου και γραφικών, η ύπαρξη λευκού χώρου, η συνέπεια σε γραμματοσειρές, εικόνες, αναδυόμενα παράθυρα και οτιδήποτε δεν τραβά την προσοχή του χρήστη με στόχο να τον αποπροσανατολίσει τον βοηθά να ολοκληρώσει τον στόχο του ο οποίος είναι η αφομοίωση της γνώσης που του προσφέρεται.

ΑΛΛΗΛΕΠΙΔΡΑΣΗ ΜΕ ΤΟ ΠΕΡΙΕΧΟΜΕΝΟ: Επειδή η ηλεκτρονική μάθηση λειτουργεί σαν ένα μέσο αυτοδιδασκαλίας η αλληλεπίδραση του χρήστη – εκπαιδευόμενου με το περιεχόμενο τον βοηθά να αναπληρώσει ως ένα βαθμό το κενό της απουσίας του εκπαιδευτικού. Παρόμοια με την εκπαίδευση στην τάξη, πρέπει να υπάρχει ισορροπία στην εφαρμογή περιεχομένου αλληλεπίδρασης. Πολύ μεγάλη συμμετοχή κινδυνεύει να επισκιάσει τους μαθησιακούς στόχους ενώ πολύ μικρή κινδυνεύει να χαθεί το ενδιαφέρον του χρήστη για το μάθημα. Σύντομα τεστάκια, διαδραστικά γραφικά, υπερσύνδεσμοι για επιπλέον έννοιες, ορισμούς και επεξηγήσεις βελτιώνουν σημαντικά την εμπειρία και προσφέρουν προστιθέμενη αξία στο περιεχόμενο του θέματος κρατώντας παράλληλα αμείωτο το ενδιαφέρον του χρήστη.

ΛΕΙΤΟΥΡΓΙΚΟ & ΕΥΧΡΗΣΤΟ: Έχει συμβεί πολλές φορές εξαιρετικές ιδέες και δημιουργικά προγράμματα να απορρίπτονται επειδή πολύ απλά δεν λειτουργούν. Ομοίως και ένα καλά οργανωμένο και πρωτοποριακό μάθημα ηλεκτρονικής εκμάθησης να αποδειχθεί λανθασμένο αν δεν λειτουργεί σωστά. Ακόμα και η δυσκολία ως προς την χρήση με υπερσυνδέσμους που δεν λειτουργούν, περιεχόμενο με ορθογραφικά λάθη, δραστηριότητες που δεν λειτουργούν όπως σχεδιάστηκαν αποθαρρύνουν τους χρήστες και τους οδηγούν σε άλλα πιο εύχρηστα προγράμματα εκμάθησης.

4.3 Μορφές e-Learning

Στην ηλεκτρονική διδασκαλία με την μορφή του e-learning υπάρχουν 2 κατηγορίες εκπαίδευσης, η *Σύγχρονη* και η *Ασύγχρονη*.

Σύγχρονη Εκπαίδευση

Όπως πολύ σωστά λέει και η λέξη στην σύγχρονη μορφή ηλεκτρονικής μάθησης όλοι οι συμμετέχοντες πρέπει συγχρόνως – «ταυτόχρονα» να βρίσκονται στις εικονικές αίθουσες διδασκαλίας online και να αλληλεπιδρούν μεταξύ τους ή με τον εκπαιδευτή τους σε πραγματικό χρόνο μέσω του παγκόσμιου Ιστού. Η αλληλεπίδραση επιτυγχάνεται είτε μέσω άμεσων μηνυμάτων, μέσω συνομιλίας και τηλεδιάσκεψων. Και όλα αυτά ενώ η διδασκαλία μπορεί παράλληλα να βιντεοσκοπείται με σκοπό την μελλοντική μελέτη και επανάληψη του μαθήματος.

Μερικά από τα πλεονεκτήματα αυτής της μεθόδου είναι:

- Δυνατότητα εξατομίκευσης εκπαίδευσης για κάθε εκπαιδευόμενο
- Δυνατότητα καταγραφής διδακτικών δραστηριοτήτων
- Συνεχής παρακολούθηση και βελτίωση των ικανοτήτων
- Πιθανότητες συνεργασίας και διασύνδεσης για χρήστες που βρίσκονται σε όλο τον κόσμο

Ασύγχρονη Εκπαίδευση

Η ασύγχρονη ηλεκτρονική μάθηση πέραν του γεγονότος ότι δεν απαιτεί την ταυτόχρονη συμμετοχή του εκπαιδευτή και των εκπαιδευόμενων θυμίζει πιο πολύ ένα είδος μάθησης παύσης και επανάληψης. Στην ασύγχρονη εκπαίδευση μπορούν να χρησιμοποιηθούν

τεχνολογίες όπως ψηφιακά βιβλία , forum συζητήσεων, ψηφιακοί δίσκοι, ιστοσελίδες κ.α.. Οι εκπαιδευόμενοι μπορούν να καθορίζουν οι ίδιοι τον χρόνο τους, να εργάζονται ατομικά με το υλικό εκπαίδευσης που τους διατίθεται, να συνομιλούν μέσω mail με τους εισηγητές-καθηγητές τους για τυχόν απορίες ή ακόμα και με τους υπόλοιπους εκπαιδευόμενους μέσω άμεσων μηνυμάτων. Στις μέρες μας πολλοί χρήστες επιθυμούν την ασύγχρονη αντί για τη σύγχρονη μάθηση διότι τους δίνει τη δυνατότητα να μην επιβαρύνονται οι καθημερινές τους υποχρεώσεις.

4.4 Τεχνολογίες που χρησιμοποιούνται στην εκπαίδευση e-Learning

Κάποιες απ' τις ευρέως χρησιμοποιούμενες τεχνολογίες στην ηλεκτρονική εκπαίδευση είναι οι εξής:

- Φορητές έξυπνες συσκευές (smartphones, tablets, laptops)
- Φορητοί εξωτερικοί δίσκοι USB
- Ψηφιακοί δίσκοι πολυμέσων (CD - DVD)
- Σελίδες που βρίσκονται στον παγκόσμιο ιστό (World Wide Web)
- Ψηφιακά βίντεο πολυμέσων σε δημοφιλείς ιστότοπους(YouTube)
- Ηλεκτρονικό ταχυδρομείο
- Τηλεδιασκέψεις
- Ηλεκτρονικά Παιχνίδια με εκπαιδευτικό περιεχόμενο
- Εκπαιδευτικά κινούμενα σχέδια

5. Δομημένη Γλώσσα Ερωτημάτων - (Structured Query Language - SQL)

5.1 Ορισμός

Η SQL από τα αρχικά των λέξεων Structured Query Language τα οποία σημαίνουν δομημένη γλώσσα ερωτημάτων είναι μια γλώσσα που σχεδιάστηκε για τη διαχείριση δεδομένων που διατηρούνται σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS - Relational Database Management System) ή για την επεξεργασία ροής σε ένα σύστημα διαχείρισης ροής σχεσιακών δεδομένων (RDSMS). Είναι ιδιαίτερα χρήσιμη για τον χειρισμό δομημένων δεδομένων όπου υπάρχουν σχέσεις μεταξύ διαφορετικών οντοτήτων - μεταβλητών των δεδομένων. Η SQL προσφέρει δύο βασικά πλεονεκτήματα έναντι των παλαιότερων γλωσσών που χρησιμοποιούνταν, διότι εισήγαγε την έννοια της πρόσβασης σε πολλά αρχεία με μία μόνο εντολή και επιπλέον, εξάλειψε την ανάγκη καθορισμού τρόπων επίτευξης μιας εγγραφής, π.χ. με ή χωρίς δείκτη. Η SQL αρχικά βασιζόμενη στη σχεσιακή άλγεβρα και τον σχεσιακό λογισμό πλειάδων αποτελείται από πολλούς τύπους δηλώσεων οι οποίοι μπορεί να κατηγοριοποιούνται ανεπίσημα ως υπογλώσσες, όπως: μια γλώσσα ερωτημάτων δεδομένων (Data Query Language - DQL), μια γλώσσα ορισμού δεδομένων (Data Definition Language - DDL), μια γλώσσα ελέγχου δεδομένων (Data Control Language - DCL). Το πεδίο εφαρμογής της SQL περιλαμβάνει ερώτημα δεδομένων(data query), χειρισμό δεδομένων (Εισαγωγή-(INSERT), Ενημέρωση-(UPDATE) και Διαγραφή-(DELETE)), ορισμό δεδομένων (δημιουργία σχημάτων(SCHEMA CREATION) και τροποποίηση) και έλεγχο πρόσβασης δεδομένων. Παρόλο που η SQL περιγράφεται συχνά, και σε μεγάλο βαθμό είναι, ως μια δηλωτική γλώσσα περιλαμβάνει επίσης και διαδικαστικά στοιχεία.

Ο Edgar F. Codd χρησιμοποίησε την SQL ως μια από τις πρώτες εμπορικές γλώσσες για την υλοποίηση του σχεσιακού του μοντέλου. Το μοντέλο παρουσιάστηκε σε μια δημοσίευση του που άσκησε μεγάλη επιρροή με τίτλο, "Ένα σχεσιακό μοντέλο δεδομένων για μεγάλες κοινές τράπεζες δεδομένων" που εκδόθηκε το 1970. Παρά το γεγονός ότι δεν ακολουθεί πλήρως το σχεσιακό μοντέλο όπως περιγράφεται από τον Codd, έγινε η πιο διαδεδομένη γλώσσα της βάσης δεδομένων.

Η SQL έγινε πρότυπο του Αμερικανικού Εθνικού Ινστιτούτου Προτύπων (ANSI) το 1986 και του Διεθνούς Οργανισμού Τυποποίησης (ISO) το 1987. Έκτοτε, το πρότυπο έχει αναθεωρηθεί ώστε να περιλαμβάνει ένα ευρύτερο σύνολο χαρακτηριστικών. Παρά την ύπαρξη τέτοιων

προτύπων, το μεγαλύτερο μέρος του SQL κώδικα δεν είναι εντελώς «φορητό» μεταξύ διαφορετικών συστημάτων βάσεων δεδομένων χωρίς προσαρμογές.

5.2 Ιστορική Αναδρομή

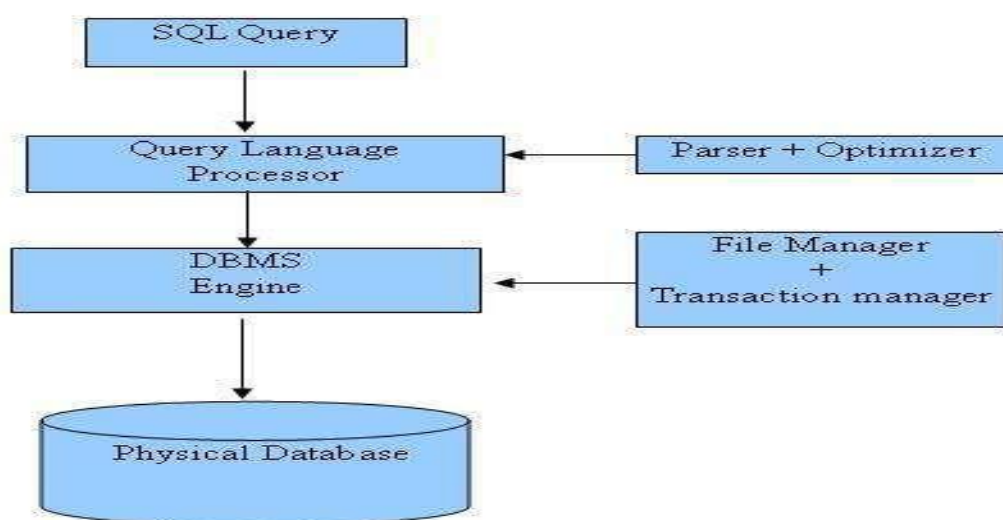
Η SQL αναπτύχθηκε αρχικά στην IBM από τους Donald D. Chamberlin και Raymond F. Boyce οι οποίοι ακολούθησαν τη δημοσίευση σχετικά με το σχεσιακό μοντέλο από τον Ted Codd στις αρχές της δεκαετίας του 1970. Αυτή η έκδοση, που αρχικά ονομάστηκε SEQUEL (Structured English Query Language), σχεδιάστηκε για να χειραγωγήσει και να ανακτήσει τα δεδομένα που ήταν αποθηκευμένα στο αρχικό σύστημα διαχείρισης της βάσης δεδομένων της IBM, με το όνομα System R, το οποίο είχε αναπτύξει μια ομάδα της IBM κατά τη δεκαετία του 1970.

Η πρώτη προσπάθεια των Chamberlin και Boyce για μια γλώσσα σχεσιακής βάσης δεδομένων πήρε το όνομα SQUARE, αλλά ήταν πολύ δύσκολο να χρησιμοποιηθεί. Αφού μετακόμισαν στο Ερευνητικό Εργαστήριο της IBM το 1973, άρχισαν να εργάζονται στο SEQUEL που αργότερα μετατράπηκε σε SQL επειδή το "SEQUEL" ήταν εμπορικό σήμα της εταιρείας Hawker Siddeley Dynamics Engineering Limited με έδρα το Ηνωμένο Βασίλειο.

Μετά την δοκιμή της SQL σε ιστότοπους δοκιμών πελατών για να προσδιοριστεί η χρησιμότητα και η πρακτικότητα του συστήματος, η IBM άρχισε να αναπτύσσει εμπορικά προϊόντα βασισμένα στο πρωτότυπο με το όνομα System R, συμπεριλαμβανομένων των System/38, SQL/DS και DB2, τα οποία ήταν εμπορικά διαθέσιμα το 1979, 1981 και 1983, αντίστοιχα.

5.3 Αρχιτεκτονική

Η SQL καταφέρνει με διάφορους τρόπους να αποκλίνει από την θεωρητική της θεμελίωση για τον υπολογισμό της πλειάδας και το σχεσιακό μοντέλο. Στο μοντέλο αυτό ένας πίνακας είναι ένα σύνολο πλειάδων, ενώ στην SQL οι πίνακες και τα αποτελέσματα των εντολών αναζήτησής (query results) αποτελούν λίστες σειρών (list of rows) έτσι κάθε σειρά μπορεί να εμφανιστεί πολλές φορές και η αλληλουχία των σειρών μπορεί να χρησιμοποιηθεί σε εντολές αναζήτησης π.χ. στην ρήτρα LIMIT. Πατώντας πάνω σε αυτό οι επικριτές της υποστηρίζουν ότι πρέπει να αντικατασταθεί με μια γλώσσα η οποία «πατάει» αυστηρά πάνω στην αρχική θεμελίωση. Παρ' όλα αυτά δεν υπάρχει τίποτα που να απαγορεύει μια τέτοια μοναδικότητα από το να προστεθεί στην SQL ή τουλάχιστον μια παραλλαγή της έτσι ώστε η βιομηχανία να μην χρειαστεί να στραφεί σε μια νέα εντελώς γλώσσα ερωτημάτων ώστε να αποκτήσει αυτήν τη μοναδικότητα. Ένα απλό διάγραμμα το οποίο δείχνει την αρχιτεκτονική της SQL είναι το παρακάτω.



Αρχιτεκτονική της SQL

Όταν εκτελείται μια εντολή SQL για οποιοδήποτε μοντέλο σχεσιακής βάσης δεδομένων, το σύστημα καθορίζει τον καλύτερο τρόπο για να εκτελέσει το αίτημά και ο μηχανισμός SQL υπολογίζει τον τρόπο ερμηνείας της εργασίας.

Υπάρχουν διάφορα στοιχεία που περιλαμβάνονται σε αυτή τη διαδικασία μερικά από τα οποία είναι:

- Αποστολέας ερωτημάτων
- Μηχανές βελτιστοποίησης
- Κλασική μηχανή ερωτήματος
- SQL μηχανή ερωτήματος, κλπ.

Μια κλασική μηχανή ερωτήματος χειρίζεται όλα τα ερωτήματα που δεν αφορούν SQL, αλλά μια μηχανή αναζήτησης ερωτήματος SQL δεν χειρίζεται λογικά αρχεία.

Επιμέρους γλωσσικά στοιχεία στα οποία υποδιαιρείται η SQL περιλαμβάνουν:

- Ρήτρες (Clauses) οι οποίες αποτελούν συστατικά στοιχεία των δηλώσεων και των ερωτημάτων.
- Εκφράσεις (Expressions) οι οποίες μπορούν να παράγουν είτε πίνακες που αποτελούνται από στήλες και γραμμές με δεδομένα είτε με τιμές κλίμακας.
- Ευρήματα (Predicates) τα οποία καθορίζουν συνθήκες οι οποίες μπορούν να αξιολογηθούν στην λογική των τριών τιμών της SQL (αληθής / ψευδής / άγνωστη) και χρησιμοποιούνται για τον περιορισμό των αποτελεσμάτων των δηλώσεων (statements) και των ερωτημάτων (queries) ή ακόμα και για να αλλάξουν τη ροή του προγράμματος.
- Ερωτήματα (Queries) τα οποία ανακτούν τα δεδομένα από τη βάση σύμφωνα με συγκεκριμένα κριτήρια.
- Δηλώσεις (Statements) οι οποίες μπορούν να έχουν μόνιμη επίδραση στα σχήματα των βάσεων και στα δεδομένα ή να ελέγχουν τις συναλλαγές, τη ροή του προγράμματος, συνδέσεις, συνεδρίες ή διαγνωστικά. Οι δηλώσεις περιλαμβάνουν ακόμα το ερωτηματικό (;) το οποίο τερματίζει τη δήλωση και παρόλο που δεν απαιτείται σε κάθε πλατφόρμα ορίζεται σαν ένα βασικό κομμάτι της γραμματικής της SQL.

Από όλα τα παραπάνω είναι φανερό ότι η SQL έχει τις δυνατότητες να εισάγει δεδομένα, να αναζητά, να ενημερώνει / διαγράφει δεδομένα σε μια βάση δεδομένων και να δημιουργεί / τροποποιεί ένα σχήμα βάσης δεδομένων (database schema). Μεταξύ αυτών, τα ερωτήματα (queries) είναι αυτά που χρησιμοποιούνται περισσότερο. Ο χρήστης ορίζει τα ερωτήματα με τρόπο τέτοιο ώστε να περιγράφει τις επιθυμητές ιδιότητες του υποσυνόλου των δεδομένων που χρειάζεται για ανάκτηση από τη βάση δεδομένων. Έπειτα πάνω στο ερώτημα το Σύστημα Διαχείρισης Βάσεων Δεδομένων εκτελεί τις απαραίτητες βελτιστοποιήσεις καθώς και τις απαραίτητες φυσικές λειτουργίες για την παραγωγή των αποτελεσμάτων του ερωτήματος. Η SQL επιτρέπει να συμπεριληφθούν σε στήλες των βάσεων δεδομένων τύποι δεδομένων όπως: χορδές χαρακτήρων (character strings), αριθμοί, χορδές δυαδικών ψηφίων (bit strings), ημερομηνία και ώρα. Μολονότι η SQL ανήκει στο πρότυπο ANSI, υπάρχουν πολλές διαφορετικές εκδόσεις της γλώσσας. Στην προσπάθεια τους όμως να είναι σύμφωνες με το πρότυπο ANSI, όλες αυτές οι εκδόσεις υποστηρίζουν τις ευρέως χρησιμοποιούμενες εντολές όπως SELECT, UPDATE, DELETE, INSERT, WHERE με παρόμοιο τρόπο.

5.4. Δημοφιλείς Επεκτάσεις της SQL

5.4.1 Microsoft SQL Server & T-SQL (Transact SQL)

Όπως αναλύσαμε διεξοδικά νωρίτερα η SQL αποτελεί μια γλώσσα υπολογιστών για βάσεις δεδομένων η οποία χρησιμοποιείται κατά κόρον για να έχουμε πρόσβαση και να χειριζόμαστε τα δεδομένα σε Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων. Έτσι λοιπόν η Microsoft

δημιούργησε τον δικό της SQL διακομιστή (server) ο οποίος είναι στην ουσία ένας διακομιστής βάσης δεδομένων που χρησιμοποιεί την SQL. Πιο συγκεκριμένα χρησιμοποιεί την T-SQL και ANSI SQL ως τις κύριες γλώσσες ερωτημάτων του. Η T-SQL επεκτείνει την SQL με το να της προσθέτει περισσότερα χαρακτηριστικά όπως διαδικαστικό προγραμματισμό, τοπικές μεταβλητές και με το να υποστηρίζει μεθόδους για επεξεργασία των δεδομένων. Λίγο πιο αναλυτικά η T-SQL χρησιμοποιώντας λέξεις κλειδιά ή ρήτρες βελτιώνει την απόδοση από τον χρήστη ξεχωριστών εντολών, μπορεί να επιτρέψει την εισαγωγή συνδέσεων στις δηλώσεις DELETE και UPDATE με τη ρήτρα FROM ή και να παρέχει δυνατότητες ελέγχου ροής στον χρήστη χρησιμοποιώντας λέξεις κλειδιά όπως BEGIN - END, BREAK, IF - ELSE κ.α. Αυτά τα χαρακτηριστικά καθιστούν την T-SQL πλήρης. Μια εφαρμογή η οποία θέλει να επικοινωνήσει με τον διακομιστή MS SQL πρέπει υποχρεωτικά να στείλει μια δήλωση σε T-SQL στον διακομιστή. Ο διακομιστής μπορεί να χρησιμοποιηθεί για την δημιουργία εφαρμογών βάσης δεδομένων οι οποίες μπορούν να προσεγγιστούν από επιτραπέζιους ή επιχειρηματικούς σταθμούς εργασίας, από το Διαδίκτυο ακόμα και από έξυπνες φορητές συσκευές όπως κινητά τηλέφωνα ή ταμπλέτες. Ο MS SQL server που κυκλοφόρησε το 1989 αναπτύχθηκε για το λειτουργικό σύστημα OS2, αποτέλεσε την πρώτη έκδοση για την Microsoft και του δόθηκε η ονομασία SQL Server 1.0. Από τότε κυκλοφόρησαν πολλές εκδόσεις που περιλάμβαναν χαρακτηριστικά προσαρμοσμένα σε διαφορετικούς χρήστες.

5.4.2 Oracle Server – PL/SQL (Procedural Language/Structured Query Language)

Ομοίως με την Microsoft έτσι και η Oracle δημιούργησε τον δικό της SQL διακομιστή και κατ' επέκταση μια γλώσσα που θα μπορούσε να χρησιμοποιήσει στον διακομιστή της και που στην ουσία θα επέκτεινε με την σειρά της την SQL. Αυτή δεν ήταν άλλη από την PL/SQL (Procedural Language/Structured Query Language). Όπως λέει και το όνομα της η PL/SQL έδωσε τη δυνατότητα στους προγραμματιστές να συνδυάσουν τη δύναμη της SQL με διαδικαστικό προγραμματισμό χρησιμοποιώντας δηλώσεις διαδικασιών (procedural statements). Παρέχει υποστήριξη για μεταβλητές, λούπες όπως (WHILE loops, FOR loops και Cursor FOR loops), δηλώσεις υπό όρους, εξαιρέσεις και πίνακες (arrays). Αντίθετα με άλλες γλώσσες που επέκτειναν την SQL στην PL/SQL δηλώσεις όπως CREATE, DROP ή ALTER δεν επιτρέπονται ενώ στις μεθόδους μπορούν να περιέχονται δηλώσεις σε SQL ή σε PL/SQL και να επιστρέφει τιμή στον χρήστη. Οι διαδικασίες (procedures) από την άλλη πλευρά όμως σε PL/SQL δεν μπορούν να περιέχουν εντολές σε SQL. Ένα από τα πλεονεκτήματα της PL/SQL είναι ότι υποστηρίζει συγκεκριμένες έννοιες του αντικειμενοστραφούς προγραμματισμού όπως η ενθυλάκωση, η υπερφόρτωση μεθόδων και η απόκρυψη πληροφορίας αλλά δεν υποστηρίζει την κληρονομικότητα. Η PL/SQL χρησιμοποιεί ακόμα την έννοια των πακέτων τα οποία μπορούν να χρησιμοποιηθούν για την ομαδοποίηση των μεταβλητών, των διαδικασιών, των λειτουργιών κ.λ.π. από τον χρήστη το οποίο σημαίνει την πολύ εύκολη επαναχρησιμοποίηση κώδικα και την εξοικονόμηση πολύτιμου χρόνου από τους χρήστες. Τέλος η χρήση του PL/SQL κώδικα στον διακομιστή της Oracle θα οδηγούσε σε αισθητά βελτιωμένα απόδοσης διότι ο διακομιστής της Oracle πριν εκτελέσει την εντολή από τον χρήστη προετοιμάζει-προσυντάσει για μεταγλώπτιση (precompile) τον κώδικα. Η βάση δεδομένων της Oracle επέκτεινε το απλό σχεσιακό μοντέλο σε αντικειμενοστραφές σχεσιακό μοντέλο κάνοντας δυνατή την αποθήκευση πολύπλοκων επαγγελματικών μοντέλων σε μια σχεσιακή βάση δεδομένων. Το 1979 η τότε Relational Software Inc γνωστή αργότερα και ως Oracle εισήγαγε το Oracle V2 (Έκδοση 2) ως το πρώτο εμπορικά διαθέσιμο SQL μοντέλο σχεσιακών βάσεων δεδομένων, ένα γεγονός ορόσημο στην ιστορία των σχεσιακών βάσεων δεδομένων. Από τότε μέχρι σήμερα η Oracle διαρκώς επέκτεινε με προσθήκες και βελτιώσεις το ήδη υπάρχων μοντέλο της ώστε να το εξελίξει και να το κρατάει ενήμερο με τις ανάγκες της εποχής.

5.4.3 MySQL

Ένα σύστημα ανοιχτού κώδικα για τη διαχείριση σχεσιακών βάσεων δεδομένων δομημένων ερωτημάτων αποτελεί η MySQL. Η MySQL βασίζεται στο μοντέλο διακομιστή - πελάτη με τον

διακομιστή MySQL server να αποτελεί τον πυρήνα της MySQL ο οποίος χειρίζεται όλες τις εντολές-οδηγίες προς τη βάση δεδομένων. Η MySQL αναπτύχθηκε αρχικά για γρήγορη διαχείριση μεγάλων βάσεων δεδομένων και παρόλο που είναι συνήθως εγκατεστημένη σε ένα μόνο μηχάνημα, είναι σε θέση να αποστείλει τη βάση δεδομένων σε πολλαπλές τοποθεσίες, καθώς οι χρήστες έχουν πρόσβαση σε αυτές μέσω διαφορετικών διεπαφών πελάτη MySQL. Αυτές οι διεπαφές στέλνουν καταστάσεις SQL στο διακομιστή και στη συνέχεια εμφανίζουν τα αποτελέσματα. Η MySQL επιτρέπει την αποθήκευση και πρόσβαση σε δεδομένα σε πολλαπλές μηχανές αποθήκευσης, συμπεριλαμβανομένων των InnoDB(μηχανή αποθήκευσης για τις λειτουργίες της MySQL), CSV(αρχεία Excel) και NDB(άλλη μια μηχανή αποθήκευσης για τις λειτουργίες αποθήκευσης γραμμών και στηλών). Η MySQL είναι επίσης ικανή να αναπαράγει πίνακες δεδομένων και διαμερισμάτων για καλύτερη απόδοση και αντοχή ενώ οι χρήστες της δεν απαιτείται να μάθουν νέες εντολές, μπορούν να έχουν πρόσβαση στα δεδομένα τους χρησιμοποιώντας τυπικές εντολές SQL. Είναι γραμμένη σε C και C++ και είναι προσβάσιμη και διαθέσιμη σε περισσότερες από 20 πλατφόρμες, συμπεριλαμβανομένων Mac, Windows, Linux και Unix. Παρόλο που μπορεί να χρησιμοποιηθεί σε ένα ευρύ φάσμα εφαρμογών, η MySQL συσχετίζεται συχνότερα με εφαρμογές ιστού και ηλεκτρονική δημοσίευση. Εδώ αξίζει να αναφερθεί ότι η MySQL είναι ένα σημαντικό συστατικό μιας στήριξης επιχειρήσεων ανοικτού κώδικα που ονομάζεται LAMP. Το LAMP είναι μια πλατφόρμα ανάπτυξης ιστού που χρησιμοποιεί το Linux ως λειτουργικό σύστημα, τον Apache ως διακομιστή ιστού, τη MySQL ως το σύστημα διαχείρισης σχεσιακής βάσης δεδομένων και την PHP ως αντικειμενοστραφή γλώσσα προγραμματισμού. (Μερικές φορές η Perl ή η Python χρησιμοποιείται αντί της PHP). Η πρώτη έκδοση της MySQL που αρχικά δημιουργήθηκε για προσωπική χρήση εμφανίστηκε το 1995. Σχεδιασμένη αρχικά από την σουηδική εταιρεία MySQL AB, η MySQL αποκτήθηκε από την Sun Microsystems το 2008 και στη συνέχεια από την Oracle όταν αγόρασε την Sun το 2010. Οι προγραμματιστές που θέλουν να χρησιμοποιήσουν την MySQL το κάνουν υπό την Γενική Άδεια Δημόσιας Χρήσης – (GNU General Public License (GPL)) η οποία αποτελεί πιθανόν την πιο δημοφιλή άδεια χρήσης ελεύθερου λογισμικού ενώ οι επιχειρήσεις πρέπει να αποκτήσουν άδεια από την Oracle αν θέλουν να ενσωματώσουν το συγκεκριμένο λογισμικό σε εμπορικές εφαρμογές τους.

5.4.4 PostgreSQL

Η PostgreSQL αποτελεί ακόμα ένα ανοικτού κώδικα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων το οποίο όμως διαθέτει και αντικειμενοστραφή χαρακτηριστικά. Παρόλο που σχεδιάστηκε για να λειτουργεί σε πλατφόρμες UNIX σχεδιάστηκε επίσης ώστε να είναι φορητή και να μπορεί να λειτουργεί σε διαφορετικές πλατφόρμες όπως υπολογιστές με λειτουργικό Solaris, Windows, Mac OS X. Ο πηγαίος κώδικας είναι διαθέσιμος υπό την άδεια PostgreSQL μια ελεύθερη άδεια ανοικτού κώδικα η οποία μοιάζει αρκετά με την Γενική Άδεια Δημόσια Χρήσης και αυτό δίνει τη δυνατότητα στους προγραμματιστές να χρησιμοποιούν, τροποποιούν και να διανείμουν την PostgreSQL σε οποιαδήποτε μορφή επιθυμούν είτε πρόκειται για ακαδημαϊκή, εμπορική ή ιδιωτική χρήση. Ένα μεγάλο μέρος των θεμελιωδών αρχών του προτύπου της SQL υποστηρίζεται από την PostgreSQL και προσφέρει πολλά χαρακτηριστικά όπως: ξένα κλειδιά, πολύπλοκα ερωτήματα βάσης, προβολές, ενεργοποιητές και ακεραιότητα συναλλαγών. Ακόμα η αντικειμενοστρέφεια της γλώσσας δίνει πολλά πλεονεκτήματα και διευκολύνσεις στους χρήστες της καθιστώντας την αν και δωρεάν ένα πολύ χρήσιμο και εύκολο συντηρήσιμο από άποψη οικονομικών πόρων προγραμματιστικό εργαλείο. Η PostgreSQL έρχεται με πολλά χαρακτηριστικά που στοχεύουν να βοηθήσουν τους προγραμματιστές να δημιουργήσουν εφαρμογές, τους διαχειριστές να προστατεύσουν την ακεραιότητα των δεδομένων, να δημιουργήσουν περιβάλλοντα ανεκτικά σε σφάλματα και να βοηθήσουν όλους τους χρήστες να διαχειριστούν τα δεδομένα ανεξάρτητα από το μέγεθος τους. Εκτός από ελεύθερη και ανοικτού κώδικα, η PostgreSQL είναι εξαιρετικά επεκτάσιμη. Για παράδειγμα, ο χρήστης μπορεί να ορίσει τους δικούς του τύπους δεδομένων, να δημιουργήσει προσαρμοσμένες λειτουργίες, ακόμα και να γράψει κώδικα από διαφορετικές γλώσσες προγραμματισμού χωρίς να ανασυγκροτήσει τη βάση δεδομένων του! Ο χρήστης έχει την δυνατότητα να επεκτείνει την PostgreSQL προσθέτοντας νέες μεθόδους ερευτηρίου, γλώσσες

διαδικασιών, συναρτήσεις, μεθόδους και τύπους δεδομένων. Η PostgreSQL προέκυψε κατά την εξέλιξη του πρότζεκτ με το όνομα Ingres στο πανεπιστήμιο Berkeley της Καλιφόρνια και από το 1986 μέχρι σήμερα μετρά πάνω από 30 χρόνια συνεχούς και ενεργής εξέλιξης της κύριας πλατφόρμας της.

5.4.5 SQLite

Σε αντιδιαστολή με τα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων τα οποία περιγράψαμε προηγουμένως καθώς και με πολλά άλλα, η SQLite δεν είναι μηχανισμός βάσης δεδομένων πελάτη-διακομιστή. Αντίθετα, είναι ενσωματωμένη στο τελικό πρόγραμμα. Γι' αυτό το λόγο επειδή η SQLite δεν έχει ξεχωριστή διαδικασία διακομιστή διαβάζει και γράφει απευθείας σε συνηθισμένα αρχεία δίσκου. Μια πλήρης βάση δεδομένων SQL με πολλαπλούς πίνακες (tables), δείκτες (indices), ενεργοποιητές (triggers) και προβολές (views), περιέχεται σε ένα αρχείο ενός δίσκου. Η μορφή του αρχείου βάσης δεδομένων μπορεί να χρησιμοποιηθεί από πολλές διαφορετικές πλατφόρμες – μπορούμε για παράδειγμα να αντιγράψουμε ελεύθερα μια βάση δεδομένων μεταξύ συστημάτων 32-bit και 64-bit αρχιτεκτονικών. Αυτές οι δυνατότητες καθιστούν την SQLite μια δημοφιλή επιλογή ως μορφή αρχείου εφαρμογών. Η SQLite είναι μια συμπαγής βιβλιοθήκη. Με όλες τις δυνατότητες ενεργοποιημένες, το μέγεθος της βιβλιοθήκης μπορεί να είναι μικρότερο από 600KB, ανάλογα με την πλατφόρμα προορισμού και τις ρυθμίσεις βελτιστοποίησης του μεταγλωττιστή. (Ο κώδικας 64-bit είναι μεγαλύτερος και μερικές βελτιστοποιήσεις του μεταγλωττιστή, όπως η επιθετική συναρμολόγηση και η ξεδίπλωση βρόχων μπορούν να προκαλέσουν πολύ μεγαλύτερο μέγεθος του αντικειμένου). Η SQLite τρέχει γενικά πιο γρήγορα όσο περισσότερη μνήμη της δίνεται. Παρ' όλα αυτά, η απόδοση είναι συνήθως αρκετά καλή ακόμη και σε περιβάλλον με χαμηλή μνήμη και ανάλογα με τον τρόπο με τον οποίο χρησιμοποιείται, η SQLite μπορεί να είναι ταχύτερη από λειτουργίες του συστήματος αρχείων όπως διάβασμα, εγγραφή, άνοιγμα, κλείσιμο κ.λπ. Συνοψίζοντας τα κύρια χαρακτηριστικά της και προσθέτοντας μερικά ακόμα θα μπορούσαμε να πούμε ότι οι λόγοι που οδήγησαν στην αυξανόμενη δημοφιλία της ήταν οι εξής:

- **Μηδενική διαμόρφωση:** Η SQLite σχεδιάστηκε έτσι ώστε να μην απαιτεί αρχείο ρυθμίσεων (configuration file), να μην απαιτεί εγκατάσταση και αρχικές ρυθμίσεις διότι δεν υπάρχει διακομιστής και είναι άμεσα ενσωματωμένη στην εφαρμογή μας. Επίσης δεν υπάρχει διαχειριστής όπως σε άλλες βάσεις δεδομένων η παρουσία του οποίου είναι απαραίτητη για τη δημιουργία, τη διατήρηση και την ρύθμιση των δικαιωμάτων των χρηστών σε μια βάση δεδομένων. Επιπλέον δεν απαιτούνται ενέργειες από διαχειριστές βάσεων δεδομένων σε περίπτωση σύγκρουσης συστήματος ή διακοπής ρεύματος για την ανάκτηση βάσεων δεδομένων. Όταν η SQLite διαβάζει μια βάση δεδομένων, εκτελεί αυτόματα τις απαραίτητες ενέργειες ανάκτησης στη βάση δεδομένων με τρόπο διαφανή για το χρήστη. Η SQLite τέλος υποστηρίζει το σύνολο των ιδιοτήτων οι οποίες εγγυόνται ότι οι συναλλαγές στην βάση δεδομένων (BD) λειτουργούν αξιόπιστα, δηλαδή την ατομικότητα (Atomicity), τη συνέπεια (Consistency), την απομόνωση (Isolation) και την ανθεκτικότητα (Durability) γνωστές και με το ακρωνύμιο ACID.
- **Παρουσία σε συστήματα με εντελώς διαφορετική Αρχιτεκτονική:** Τα αρχεία που υπάρχουν στη βάση δεδομένων μπορούν να μεταφερθούν από ένα σύστημα Αρχιτεκτονικής A σε ένα σύστημα Αρχιτεκτονικής B χωρίς καμία επιβάρυνση και χωρίς καμία ταλαιπωρία διότι το αρχείο της βάσης δεδομένων είναι σε δυαδική μορφή και συνεπώς όλα τα μηχανήματα χρησιμοποιούν την ίδια μορφή.
- **Ασφαλής σε πολυνηματικό προγραμματισμό:** Η SQLite δίνει τη δυνατότητα στα νήματα μια πολυνηματικής εφαρμογής να έχουν πρόσβαση στην ίδια ή σε διαφορετικές βάσεις δεδομένων ταυτόχρονα. Η SQLite αναλαμβάνει να επιλύσει το επίπεδο ανταγωνισμού νήματος.
- **Δηλωτική πληκτρολόγηση:** Οι περισσότερες μηχανές για βάσεις δεδομένων SQL χρησιμοποιούν στατική πληκτρολόγηση. Κάθε στήλη σε έναν πίνακα σχετίζεται με ένας

τύπο δεδομένων και στη συγκεκριμένη στήλη αποθηκεύονται μόνο οι τιμές αυτού του συγκεκριμένου τύπου δεδομένων. Η SQLite χαλαρώνει αυτόν τον περιορισμό χρησιμοποιώντας την Δηλωτική πληκτρολόγηση. Στην στατική πληκτρολόγηση, ο τύπος δεδομένων είναι μια ιδιότητα της ίδιας της τιμής, όχι της στήλης στην οποία αποθηκεύεται η τιμή. Η SQLite επιτρέπει έτσι στον χρήστη να αποθηκεύει οποιαδήποτε τιμή οποιασδήποτε μορφής δεδομένων σε οποιαδήποτε στήλη, ανεξάρτητα από τον δηλωμένο τύπο αυτής της στήλης. Ωστόσο, οι περισσότερες άλλες μηχανές βάσεων δεδομένων SQL ακολουθούν την στατική πληκτρολόγηση οδηγώντας αρκετούς ανθρώπους στο συμπέρασμα ότι η χρήση δηλωτικής πληκτρολόγησης είναι ένα σφάλμα της SQLite. Από την άλλη πλευρά οι δημιουργοί της SQLite ισχυρίζονται πολύ έντονα ότι αυτό είναι ένα χαρακτηριστικό γνώρισμα και ότι η χρήση δηλωτικής πληκτρολόγησης στην SQLite είναι μια σκόπιμη απόφαση σχεδιασμού που έχει αποδειχθεί στην πράξη ότι κάνει την SQLite πιο αξιόπιστη και πιο εύκολη στη χρήση, ειδικά όταν χρησιμοποιείται σε συνδυασμό με δυναμικά πληκτρολογούμενες γλώσσες προγραμματισμού όπως η Tcl και η Python.

- Συμπαγής: Όπως αναφέρθηκε και νωρίτερα μια βάση δεδομένων σε SQLite είναι ένα απλό αρχείο ενός συνηθισμένου δίσκου και συνεπώς έχει πολύ μικρό αντίκτυπο στον περιορισμένο αποθηκευτικό χώρο πολλών συσκευών και ιδιαίτερα φορητών. Γι' αυτό η δημοφιλία της σε έξυπνα κινητά και σε φορητές συσκευές εκτοξεύτηκε κατακόρυφα.
- Εύκολα παραμετροποιήσιμη: Παρέχει ένα καλό πλαίσιο στο οποίο μπορεί ο χρήστης να ορίσει και να κάνει χρήση μεθόδων SQL, συναρτήσεων, αλληλουχιών σύγκλισης φτιαγμένων από τον ίδιο.
- Αξιοπιστία απέναντι σε σφάλματα: Παρόλο που η SQLite είναι ανοιχτού κώδικα και συνεπώς δωρεάν, κάθε νέα έκδοσή της ελέγχεται προσεκτικά πριν από κάθε δημοσίευση μέσω μιας αυτοματοποιημένης σουίτας δοκιμών η οποία εκτελεί εκατομμύρια δοκιμαστικές περιπτώσεις που περιλαμβάνουν εκατοντάδες εκατομμύρια εξατομικευμένες δηλώσεις SQL και επιτυγχάνει κάλυψη δοκιμών κλάδου 100%. οι περιπτώσεις δοκιμής και τα σενάρια δοκιμών στο SQLite έχουν περίπου 1084 φορές περισσότερο κώδικα από το τον πηγαίο κώδικα της βιβλιοθήκης SQLite και διεκδικούν 100 τοις εκατό κάλυψη δοκιμών κλάδου. Αυτό το επίπεδο δοκιμών επιβεβαιώνει την εμπιστοσύνη στην SQLite από τους προγραμματιστές.

Ο κώδικας της SQLite υποστηρίζεται από μια διεθνή ομάδα προγραμματιστών που εργάζονται διαρκώς για να επεκτείνουν τις δυνατότητες της SQLite και να ενισχύσουν την αξιοπιστία και την απόδοση της, διατηρώντας παράλληλα τη συμβατότητα με τα δημοσιευμένα χαρακτηριστικά διεπαφής, τη σύνταξη SQL και τη μορφή αρχείου βάσης δεδομένων. Ο πηγαίος κώδικας είναι απολύτως δωρεάν για όποιον το θέλει, αλλά υπάρχει και επαγγελματική υποστήριξη.

Το έργο SQLite ξεκίνησε το 2000 όταν ο D. Richard Hipp την σχεδίασε με στόχο να χρησιμοποιήσει μια βάση δεδομένων στο λογισμικό που κατασκεύαζε χωρίς να απαιτείται η εγκατάσταση ενός επιπλέον συστήματος διαχείρισης βάσης δεδομένων. Το μέλλον είναι πάντα δύσκολο να προβλεφθεί, αλλά η πρόθεση των προγραμματιστών είναι να υποστηρίξουν την SQLite μέχρι το έτος 2050. Οι αποφάσεις σχεδιασμού γίνονται με αυτόν τον στόχο.

Η SQLite έχει καταφέρει να είναι η πιο ευρέως χρησιμοποιούμενη βάση δεδομένων στον κόσμο με περισσότερες εφαρμογές από ό, τι μπορούμε να μετρήσουμε, χρησιμοποιείται σήμερα από ευρέως διαδεδομένους φυλλομετρητές διαδικτύου, λειτουργικά συστήματα ηλεκτρονικών υπολογιστών καθώς και φορητών συσκευών (ταμπλέτες, κινητά τηλέφωνα, έξυπνα ρολόγια κ.α.).

6. Η Ιστοσελίδα SQLite e-learning

6.1 Σκοπός της Ιστοσελίδας

Η ιστοσελίδα ηλεκτρονικής εκμάθησης με τίτλο “e-Learning SQLite” σχεδιάστηκε με γνώμονα την παροχή γνώσης σε όσους επιθυμούν να γνωρίσουν και να εξοικειωθούν με την SQLite ή

ακόμα και σε άτομα τα οποία προερχόμενα από κάποια παραλλαγή της SQL επιθυμούν να δουν τις βασικές λειτουργίες της SQLite και αν τους ταιριάζει σε μελλοντικά σχέδια τους.

Κάθε υποψήφιος χρήστης της Ιστοσελίδας αποκτά πρόσβαση στη γνώση τόσο θεωρητικών όσο και πρακτικών εννοιών της SQLite. Με διαδραστικές μεθόδους όπως υπερσυνδέσεις, ηχητική και οπτική παρουσίαση του διδασκόμενου αντικειμένου το ενδιαφέρον των χρηστών διατηρείται αμείωτο ενώ με την παροχή ανατροφοδότησης μέσω ασκήσεων ερωτήσεων-απαντήσεων μπορούν να ξέρουν ανά πάσα στιγμή το επίπεδο στο οποίο βρίσκονται.

Η Ιστοσελίδα και κατ' επέκταση το πρόγραμμα εκμάθησης απευθύνεται κυρίως σε άτομα που έχουν μικρή ή και καθόλου εμπειρία με SQL είτε σε άτομα τα οποία είναι εξοικειωμένα με την γλώσσα SQL ή οποιαδήποτε από τις επεκτάσεις της και θέλουν να γνωρίσουν εξαρχής τη λειτουργία και την ευκολία χρήσης της SQLite.

6.2 Προαπαιτούμενα παρακολούθησης του e-Learning

Τα προαπαιτούμενα για την παρακολούθηση της ηλεκτρονικής εκμάθησης από τους υποψήφιους εκπαιδευόμενους χρήστες είναι:

- Βασικές γνώσεις χειρισμού ηλεκτρονικών υπολογιστών
- Ηλεκτρονικός Υπολογιστής
- Πρόσβαση στο Διαδίκτυο
- Κάτοχος email

6.3 Χρήση της Ιστοσελίδας e-Learning SQLite

Ο υποψήφιος χρήστης – εκπαιδευόμενος πρέπει αρχικά να δημιουργήσει έναν προσωπικό λογαριασμό στην Ιστοσελίδα. Συμπληρώνει τα προσωπικά του στοιχεία στην φόρμα εγγραφής καθώς και ένα συνθηματικό χρήστη (username) και έναν κωδικό πρόσβασης(password) με τα οποία θα έχει μετέπειτα πρόσβαση στην ιστοσελίδα.

Κατόπιν και εφόσον το σύστημα επιβεβαιώσει στον χρήστη τη μοναδικότητα των στοιχείων που καταχώρησε όσο αναφορά τα username και password τότε μπορεί να μεταβεί στην φόρμα Log In όπου πληκτρολογώντας τα και πατώντας το ανάλογο κουμπί μπορεί να αποκτήσει πρόσβαση στις διδακτικές ενότητες του προγράμματος εκμάθησης. Η πλειοψηφία κάθε ενότητας του προγράμματος έχει το θεωρητικό σκέλος της ακολουθούμενο από 1 - 2 πρακτικές ασκήσεις για να ελέγξει ο χρήστης αν κατανόησε αυτά που διάβασε στο θεωρητικό κομμάτι.

Ο χρήστης για κάθε ενότητα που ολοκληρώνει μπορεί να βλέπει το ποσοστό επιτυχίας του αλλά και σε ποια ή ποιες ενότητες υστερεί ώστε να τις επαναλάβει για καλύτερα αποτελέσματα.

6.4 Λειτουργικότητα Ιστοσελίδας e-Learning SQLite

Στην ιστοσελίδα εκμάθησης e-Learning SQLite δημιουργήσαμε όλες τις ενότητες οι οποίες κρίθηκαν αναγκαίες τόσο σε θεωρητικό όσο και σε πρακτικό υπόβαθρο ότι θα καθιστούσαν σαφές σε κάθε υποψήφιο χρήστη, την φιλοσοφία και τις θεμελιώδεις βασικές αρχές της SQLite.

Γι' αυτό το λόγο κάθε ενότητα είναι τοποθετημένη ιεραρχικά ώστε ο χρήστης να καθοδηγείται σταδιακά από πιο εύκολες και εισαγωγικές έννοιες σε ενότητες με πιο πολύπλοκες και συνδυαστικές έννοιες αλλά απ' την άλλη μεριά κάθε ενότητα είναι προσβάσιμη και ανοιχτή προς επίλυση είτε ο χρήστης έχει ολοκληρώσει την προηγούμενη της ενότητα είτε όχι. Αυτό συμβαίνει διότι το πρόγραμμα εκμάθησης που δημιουργήσαμε απευθύνεται όχι μόνο σε αρχάριους χρήστες αλλά και σε ανθρώπους που έχουν πρότερη εμπειρία με SQL και απλά θέλουν να δουν

με μια γρήγορη «ματιά» τι είναι αυτό που καθιστά την SQLite τόσο προσιτή και τόσο δημοφιλή στις μέρες μας.

Τα αποτελέσματα των ασκήσεων κάθε διδακτικής ενότητας εκφράζονται ως ποσοστό επιτυχίας στον συγκεντρωτικό πίνακα βαθμολογιών του χρήστη ώστε να ξέρει αναλυτικά σε ποια ενότητα χρειάζεται να επιστρέψει για επανάληψη, σε ποιες ενότητες άριστευσε καθώς και την συνολική του επίδοση και κατεπέκταση γνώση που απέκτησε απ' την εκμάθηση της SQLite.

6.5 Διδακτικές Ενότητες Ιστοσελίδας e-Learning SQLite

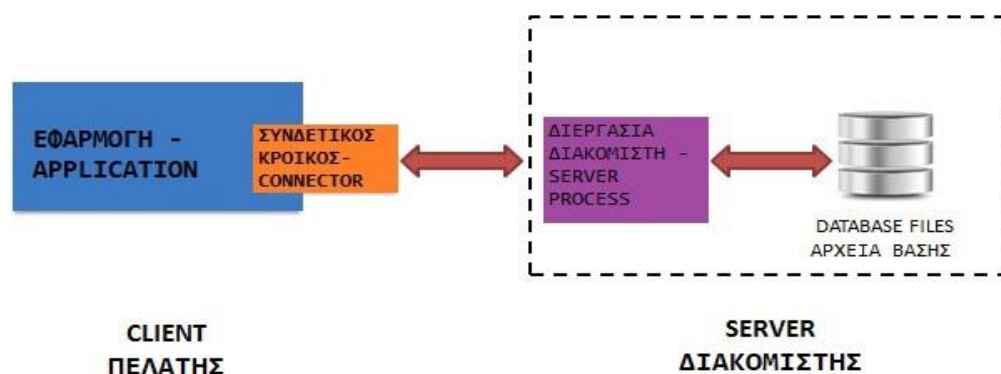
Η δημιουργία των διδακτικών ενότητων της Ιστοσελίδας, απαιτήσε πολύωρη μελέτη του αντικειμένου, καταγραφή και ιεράρχηση της διαθέσιμης ύλης ώστε να απευθύνεται σε όσο τον δυνατόν μεγαλύτερη μερίδα χρηστών. Οι διδακτικές ενότητες που απαρτίζουν το πρόγραμμα εκπαίδευσης είναι οι ακόλουθες:

6.5.1 Ενότητα 1: Γενική Εικόνα (OVERVIEW)

Στην συγκεκριμένη ενότητα δίνουμε μια σύντομη αναφορά στον χρήστη για το τι είναι η SQLite ποια τα πλεονεκτήματα και τα μειονεκτήματα της και τι οδήγησε στην μεγάλη δημοφιλία της.

Η SQLite είναι μια βιβλιοθήκη λογισμικού η οποία παρέχει ένα σύστημα για την διαχείριση σχεσιακών βάσεων δεδομένων. Η χρησιμοποίηση της λέξης ελαφριά (lite) στο όνομα της σημαίνει το ελάχιστο μέγεθος που απαιτεί σε όρους εγκατάστασης, διαχείρισης της βάσης δεδομένων και των απαιτούμενων πόρων του συστήματος. Η SQLite διαθέτει τα ακόλουθα αξιοπρόσεχτα χαρακτηριστικά: είναι αυτοτελής, δεν χρειάζεται διακομιστές(serverless), χρειάζεται μηδενική διαμόρφωση, είναι συναλλακτική(transactional) μα πάνω απ' όλα είναι ανοιχτού κώδικα άρα δωρεάν για όλους τους χρήστες.

Αναφερόμαστε ιδιαίτερα στο γεγονός ότι η SQLite δεν χρειάζεται διακομιστή διότι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων όπως το ξέραμε μέχρι σήμερα όπως π.χ. MySQL, PostgreSQL απαιτεί μια ξεχωριστή λειτουργία διακομιστή για να λειτουργήσει. Η εφαρμογή ή οι εφαρμογές που θέλουν πρόσβαση στον διακομιστή της βάσης δεδομένων χρησιμοποιούν το πρωτόκολλο TCP/IP για να στείλουν και να λάβουν αιτήματα. Η συγκεκριμένη αρχιτεκτονική ονομάζεται πελάτη - διακομιστή και απεικονίζεται χαρακτηριστικά στο παρακάτω διάγραμμα.



Διάγραμμα Αρχιτεκτονικής Πελάτη – Διακομιστή

Η SQLite όμως δεν λειτουργεί έτσι διότι δεν απαιτείται η εκτέλεση της λειτουργίας ενός διακομιστή. Η βάση δεδομένων SQLite είναι ενσωματωμένη στην εφαρμογή που έχει πρόσβαση στη βάση δεδομένων. Οι εφαρμογές αλληλεπιδρούν με την βάση δεδομένων SQLite και διαβάζουν και γράφουν απευθείας από τα αρχεία βάσης δεδομένων που είναι αποθηκευμένα στον δίσκο. Το παρακάτω διάγραμμα απεικονίζει την αρχιτεκτονική της SQLite χωρίς διακομιστή.



Διάγραμμα Αρχιτεκτονικής της SQLite με απουσία Διακομιστή

Μέσα σε όλα τα θετικά στοιχεία και πλεονεκτήματα ως προς τη χρήση της, η SQLite στις περισσότερες των χρήσεων της περιορίζεται σε ένα μέγιστο μέγεθος αρχείου της τάξης των 2 GB γι' αυτό και χρησιμοποιείται κατά κόρον σε αιτήματα HTTP χαμηλής ή μέσης επισκεψιμότητας. Ίσως είναι ένα από τα ελάχιστα μειονεκτήματα που κανείς μπορεί να της προσάψει.

6.5.2 Ενότητα 2: Κατέβασμα & Εγκατάσταση (DOWNLOAD & INSTALLATION)

Η SQLite φημίζεται για την μηδενική διαμόρφωση χαρακτηριστικών της το οποίο σημαίνει ότι δεν απαιτείται καμία πολύπλοκη ή πολύωρη εγκατάσταση για την χρησιμοποίηση της ούτε επίσης απαραίτητα δικαιώματα διαχειριστή. Σ' αυτήν την ενότητα θα δείξουμε την διαδικασία σύμφωνα με την οποία οι χρήστες μπορούν να την εγκαταστήσουν στον προσωπικό υπολογιστή τους. Φυσικά η όλη διαδικασία είναι προαιρετική για όσους χρήστες επιθυμούν την SQLite εγκατεστημένη στο σύστημα τους ώστε να πειραματιστούν μόνοι τους πάνω στις δικές τους εφαρμογές.

Για την εγκατάσταση της SQLite σε λειτουργικό σύστημα Windows ακολουθήστε με σειρά τα παρακάτω βήματα:

1. Πληκτρολογείτε σε οποιονδήποτε φυλλομετρητή την διεύθυνση <http://www.sqlite.org/download.html>
2. Στην ενότητα Precompiled Binaries for Windows κατεβάζουμε το αρχείο που είναι σε κόκκινο περιγράμμα.

Precompiled Binaries for Windows

[sqlite-dll-win32-x86-3290000.zip](#) (474.63 KiB) 32-bit DLL (x86) for SQLite version 3.29.0. (sha1: 00435a36f5e6059287cde2cebb2882669cdba3a5)

[sqlite-dll-win64-x64-3290000.zip](#) (788.61 KiB) 64-bit DLL (x64) for SQLite version 3.29.0. (sha1: c88204328d6ee3ff49ca0d58cbbec05243172c3a)

[sqlite-tools-win32-x86-3290000.zip](#) (1.71 MiB) A bundle of command-line tools for managing SQLite database files, including the `sqlite3` program, and the `sqlite3_analyzer.exe` program. (sha1: f009ff42b8c22886675005e3e57c94d62bca12b3)

3. Αφού κατέβει το αρχείο δημιουργούμε έναν νέο φάκελο π.χ **C:\sqliteFolder** και αδειάζουμε(extract) το περιεχόμενο του αρχείου που κατεβάσαμε μέσα στο φάκελο που δημιουργήσαμε. Θα πρέπει να μπορούμε να δούμε πλέον το αρχείο με όνομα **sqlite3.exe** μέσα στο φάκελο.

Για την επιβεβαίωση της εγκατάστασης μπορούμε να κάνουμε τα εξής:

- Πρώτα ανοίγουμε ένα παράθυρο εντολών των windows και μετακινούμαστε στον φάκελο που δημιουργήσαμε προηγουμένως **C:\sqliteFolder**

```

C:\Users\grigo>cd C:\sqliteFolder
C:\sqliteFolder>

```

- Έπειτα πληκτρολογούμε **sqlite3** και πατάμε **Enter**, θα πρέπει να βλέπουμε ότι και στην παρακάτω εικόνα

```

C:\sqliteFolder>sqlite3
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>

```

- Πληκτρολογώντας την εντολή **.help** μπορούμε να δούμε όλες τις διαθέσιμες εντολές στην **sqlite3**

```

sqlite> .help
.archive ...                Manage SQL archives
.auth ON|OFF                Show authorizer callbacks
.backup ?DB? FILE           Backup DB (default "main") to FILE
.bail on|off                Stop after hitting an error. Default OFF
.binary on|off              Turn binary output on or off. Default OFF
.cd DIRECTORY               Change the working directory to DIRECTORY
.changes on|off             Show number of rows changed by SQL
.check GLOB                  Fail if output since .testcase does not match
.clone NEWDB                 Clone data into NEWDB from the existing database
.databases                  List names and files of attached databases
.dbconfig ?op? ?val?       List or change sqlite3_db_config() options
.dbinfo ?DB?                Show status information about the database
.dump ?TABLE? ...           Render all database content as SQL
.echo on|off                 Turn command echo on or off
.eop on|off|full|...        Enable or disable automatic EXPLAIN QUERY PLAN
.excel                       Display the output of next command in a spreadsheet
.exit ?CODE?                 Exit this program with return-code CODE
.expert                      EXPERIMENTAL. Suggest indexes for specified queries
.filectrl CMD ...           Run various sqlite3_file_control() operations
.fullschema ?--indent?     Show schema and the content of sqlite_stat tables
.headers on|off              Turn display of headers on or off
.help ?-all? ?PATTERN?     Show help text for PATTERN
.import FILE TABLE         Import data from FILE into TABLE
.imposter INDEX TABLE     Create imposter table TABLE on index INDEX
.indexes ?TABLE?            Show names of indexes
.limit ?LIMIT? ?VAL?       Display or change the value of an SQLITE_LIMIT
.lint OPTIONS                Report potential schema issues.
.load FILE ?ENTRY?          Load an extension library
.log FILE|off               Turn logging on or off. FILE can be stderr/stdout
.mode MODE ?TABLE?          Set output mode
.nullvalue STRING           Use STRING in place of NULL values
.once (-e|-x|FILE)          Output for the next SQL command only to FILE
.open ?OPTIONS? ?FILE?      Close existing database and reopen FILE
.output ?FILE? ?FILE?       Send output to FILE or stdout if FILE is omitted
.parameter CMD ...          Manage SQL parameter bindings
.print STRING...            Print literal STRING
.progress N                  Invoke progress handler after every N opcodes
.prompt MAIN CONTINUE       Replace the standard prompts
.quit                        Exit this program
.read FILE                   Read input from FILE
.recover                     Recover as much data as possible from corrupt db.
.restore ?DB? FILE           Restore content of DB (default "main") from FILE
.save FILE                   Write in-memory database into FILE

```

- Για να εγκαταλείψουμε την **sqlite>**, χρησιμοποιούμε την εντολή **.quit**

```

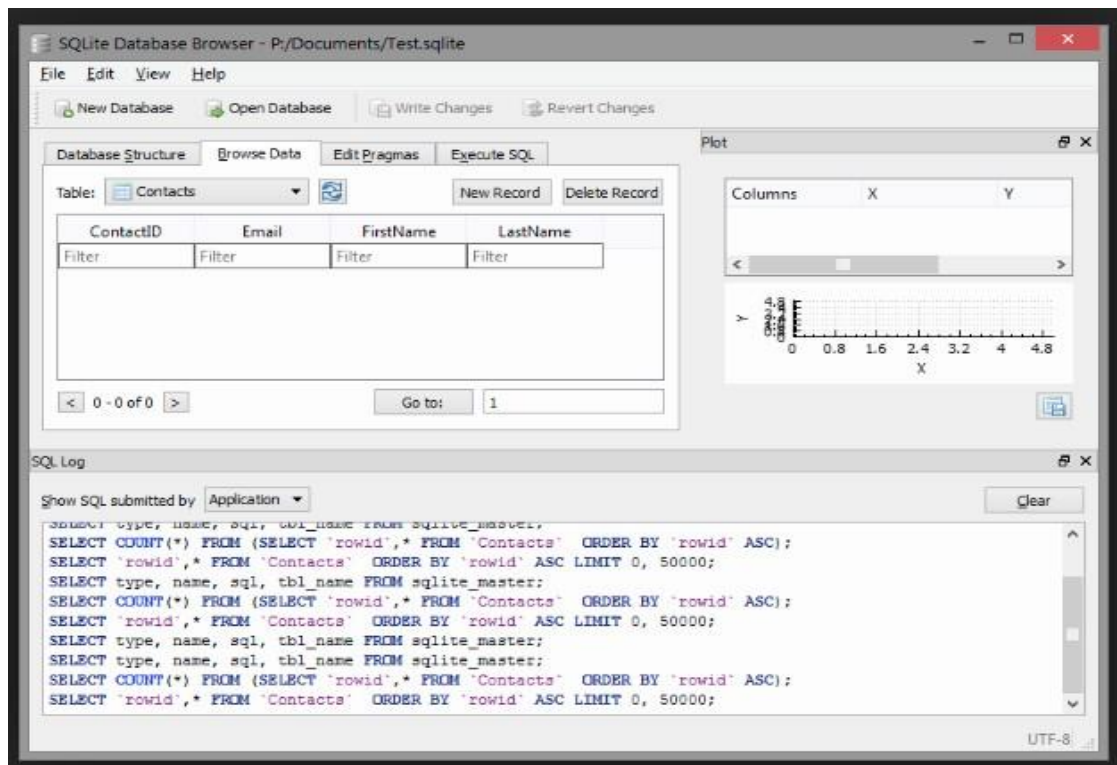
C:\sqliteFolder> sqlite> .quit
C:\sqliteFolder>

```

Εγκατάσταση εργαλείου Γραφικού περιβάλλοντος χρήστη της SQLite

Η γραμμή εντολών των Windows και το κέλυφος sqlite3 λειτουργούν άψογα και είναι κάτι που οι περισσότεροι χρήστες προτιμούν αλλά μερικές φορές ένα γραφικό περιβάλλον όχι μόνο διευκολύνει σπικτικά τον χρήστη αλλά μπορεί να περιέχει και λειτουργίες που κάνουν τη ζωή του πιο εύκολη. Όπως για παράδειγμα το DB Browser για SQLite το οποίο αποτελεί ένα δωρεάν εργαλείο με γραφικό περιβάλλον χρήστη για την διαχείριση βάσεων της SQLite. Είναι πολύ ελαφρύ όπως και η SQLite, μπορεί να λειτουργήσει σε πολλαπλές πλατφόρμες ενώ ακόμα πιο σημαντικό είναι η δυνατότητα που δίνει στον χρήστη να εισάγει και να εξάγει δεδομένα από μια βάση σε διαφορετικά φορματ όπως CSV, XML και JSON.

Αν σε οποιοδήποτε φυλλομετρητή πληκτρολογήσουμε την παρακάτω διεύθυνση <https://sqlitebrowser.org/dl/> οδηγούμαστε στην σελίδα με τις εκδόσεις της εφαρμογής ανάλογα με το λειτουργικό σύστημα που διαθέτουμε. Κατεβάζουμε το κατάλληλο αρχείο και τρέχοντας το διαθέτει οδηγό εγκατάστασης ο οποίος πολύ εύκολα μας καθοδηγεί με απλά βήματα μέχρι την ολοκλήρωση της εγκατάστασης του προγράμματος.



6.5.3 Ενότητα 3: Εντολές & Σύνταξη (COMMANDS & SYNTAX)

Όπως αναφέραμε και στην ενότητα της εγκατάστασης η SQLite διαθέτει ένα απλό εργαλείο γραμμής εντολών με το όνομα `sqlite3` που επιτρέπει στους χρήστες να αλληλεπιδρούν με τις βάσεις δεδομένων SQLite χρησιμοποιώντας δηλώσεις και εντολές SQL.

Οι εντολές της SQLite μοιάζουν με τις εντολές της SQL. Υπάρχουν 3 τύποι εντολών:

- **DDL:** Γλώσσα Δεδομένων Ορισμού - Data Definition Language
Υπάρχουν 3 τύποι σ' αυτή την ομάδα εντολών:
 - **CREATE:** Η εντολή χρησιμοποιείται για να δημιουργήσουμε έναν πίνακα ή οποιοδήποτε άλλο αντικείμενο στη βάση δεδομένων.
 - **ALTER:** Χρησιμοποιείται για να τροποποιήσει ένα αντικείμενο που υπάρχει στη βάση δεδομένων όπως ένας πίνακας.
 - **DROP:** Η εντολή χρησιμοποιείται για να διαγράψουμε έναν πίνακα εντελώς ή οποιοδήποτε άλλο αντικείμενο στη βάση δεδομένων.
- **DML:** Γλώσσα Χειρισμού Δεδομένων - Data Manipulation Language
Υπάρχουν 3 τύποι και σ' αυτή την ομάδα εντολών:
 - **INSERT:** Χρησιμοποιείται για να δημιουργήσει μια εγγραφή στη βάση.
 - **UPDATE:** Χρησιμοποιείται για να τροποποιήσει τις εγγραφές που υπάρχουν στη βάση.
 - **DELETE:** Χρησιμοποιείται για να διαγράψει τις εγγραφές που υπάρχουν στη βάση.
- **DQL:** Γλώσσα Ερωτημάτων Δεδομένων - Data Query Language
 - **SELECT:** Χρησιμοποιείται για να ανακτήσει συγκεκριμένες εγγραφές από έναν ή περισσότερους πίνακες.

Σύνδεση σε μια βάση δεδομένων SQLite

```

C:\Users\grigo>cd C:\sqliteFolder
C:\sqliteFolder>sqlite3
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
  
```

Με την εντολή `sqlite> .help` σε γραμμή εντολών windows μπορούμε να δούμε όλες τις διαθέσιμες εντολές της SQLite.

Στον παρακάτω Πίνακα αναφέρονται αναλυτικά οι εντολές της SQLite:

ΕΝΤΟΛΕΣ & ΠΕΡΙΓΡΑΦΗ
.backup ?DB? FILE
Δημιουργεί αντίγραφο ασφαλείας της βάσης (εξορισμού "main") σε ένα ΑΡΧΕΙΟ
.bail ON OFF
Διακόπτεται όταν βρει λάθος. Εξ' ορισμού: OFF(ανενεργή)
.databases

Δημιουργεί λίστα από ονόματα και αρχεία των συνδεδεμένων βάσεων
.dump ?TABLE?
Εξάγει την βάση δεδομένων σε μορφή κειμένου SQL. Αν καθορίστηκε ΠΙΝΑΚΑΣ, εξάγει μόνο τους πίνακες που ταιριάζουν με τον πίνακα μοτίβο που ορίστηκε
.echo ON OFF
Ανοίγει και κλείνει την εντολή echo
.exit
Έξοδος από τις εντολές
.explain ON OFF
Ενεργοποιεί ή απενεργοποιεί τη λειτουργία εξόδου κατάλληλη για την εντολή EXPLAIN. Χωρίς παραμέτρους, ενεργοποιεί την εντολή EXPLAIN
.header(s) ON OFF
Ενεργοποιεί ή απενεργοποιεί την εμφάνιση κεφαλίδων
.help
Εμφανίζει όλες τις εντολές της sqlite
.import FILE TABLE
Εισάγει δεδομένα από ένα αρχείο σε έναν πίνακα
.indices ?TABLE?
Εμφάνιση ονομάτων όλων των δεικτών. Εάν έχει οριστεί ΠΙΝΑΚΑΣ, εμφανίζονται μόνο δείκτες για τους πίνακες που ταιριάζουν στο μοτίβο του πίνακα που ορίστηκε
.load FILE ?ENTRY?
Φορτώνει μια βιβλιοθήκη επέκτασης
.log FILE off
Ενεργοποιεί ή απενεργοποιεί τη σύνδεση. Το αρχείο μπορεί να είναι τυπικού σφάλματος(stderr) ή τυπικής εξόδου(stdout)
.mode MODE

<p>Ορίζει τη λειτουργία εξόδου όταν η εντολή MODE είναι ένα από τα :</p> <p>csv – Τιμές διαχωρισμένες με κόμμα</p> <p>column – Αριστερά ευθυγραμμισμένες στήλες</p> <p>html – HTML <Πίνακας> κώδικας</p> <p>insert – SQL INSERT δήλωση για Πίνακα</p> <p>line – Μία τιμή για κάθε γραμμή</p> <p>list – Τιμές οριοθετημένες από τον διαχωριστή συμβολοσειράς</p> <p>tabs – Τιμές χωρισμένες με Tab</p> <p>tcl – στοιχεία λίστας TCL (Μια λίστα Tcl αποτελείται από μια συμβολοσειρά των οποίων τα στοιχεία χωρίζονται από λευκό κενό)</p>
.nullvalue STRING
Εκτύπωση ενός αλφαριθμητικού στην θέση null τιμών
.output FILENAME
Αποστολή εξόδου στο όνομα Αρχείου
.output stdout
Αποστολή εξόδου στην οθόνη
.print STRING...
Εκτυπώνει λεκτικό αλφαριθμητικό
.prompt MAIN CONTINUE
Αντικαθιστά τις κλασσικές εντολές
.quit
Έξοδος απτή γραμμή εντολών της SQLite
.read FILENAME
Εκτελεί SQL σε ένα αρχείο
.schema ?TABLE?
Εμφάνιση των δηλώσεων CREATE. Αν καθορίστηκε πίνακας εμφανίζονται μόνο οι πίνακες που ταιριάζουν στο μοτίβο πίνακα που ορίστηκε
.separator STRING
Αλλαγή του διαχωριστή που χρησιμοποιείται από τη λειτουργία εξόδου και την εισαγωγή
.show

Δείχνει τις τρέχουσες τιμές για διάφορες ρυθμίσεις
.stats ON OFF
Ενεργοποιεί απενεργοποιεί τα στατιστικά
.tables ?PATTERN?
Δημιουργεί λίστα με τα ονόματα πινάκων που ταιριάζουν στο μοτίβο που ορίστηκε
.timeout MS
Δοκιμάζει να ανοίξει κλειδωμένους πίνακες για χιλιοστά του δευτερολέπτου
.width NUM NUM
Ορίζει τα πλάτη των στηλών για την λειτουργία "στήλη"
.timer ON OFF
Ενεργοποιεί ή απενεργοποιεί τον μετρητή χρονομέτρου της CPU

Σύνταξη της SQLite

Όλες οι δηλώσεις της SQLite ξεκινούν με οποιαδήποτε από τις δεσμευμένες λέξεις SELECT, INSERT, UPDATE, DELETE, ALTER, DROP κ.λ.π. και τελειώνουν με το ελληνικό ερωτηματικό (;). Είναι σημαντικό να τονίσουμε ότι η SQLite δεν επηρεάζεται όταν η εντολή ή δήλωση που θα κάνει ο χρήστης είναι με πεζά ή κεφαλαία γράμματα. Αυτός είναι ο γενικός κανόνας αλλά υπάρχουν και κάποιες εξαιρέσεις όπως οι εντολές GLOB και glob οι οποίες έχουν διαφορετική έννοια όταν χρησιμοποιούνται με πεζά ή με κεφαλαία.

Σχόλια:

Η SQLite μας δίνει τη δυνατότητα προσθήκης σχολίων στον κώδικα μας ώστε να γίνει πιο ευανάγνωστος. Τα σχόλια ξεκινούν με 2 διαδοχικούς "-" χαρακτήρες και είναι σχόλια μιας γραμμής κώδικα. Επίσης έχουμε τη δυνατότητα σχολίων όπως στη C που ξεκινούν με τους χαρακτήρες "/*" και τελειώνουν με τους χαρακτήρες "*/" που δίνει τη δυνατότητα σχολίου σε πολλαπλές γραμμές κώδικα ταυτόχρονα.

- **sqlite> .help -- Αυτό είναι σχόλιο σε μια γραμμή του κώδικα**
- ```
SELECT employee_id, last_name, first_name
/*
Εδώ έχουμε ένα
σχόλιο που εκτείνεται
σε πολλές γραμμές
*/
FROM employees;
```

### Συμβάσεις κατά την Ονοματοδοσία στην SQLite

Μια σύμβαση ονομασίας είναι ένα σύνολο κανόνων για την επιλογή της ακολουθίας χαρακτήρων που θα χρησιμοποιηθεί για αναγνωριστικά που υποδηλώνουν το όνομα της βάσης, πίνακα, στήλης, δείκτη κ.α. στην SQLite. Το όνομα ενός αναγνωριστικού πρέπει να ξεκινά με ένα γράμμα ή τον χαρακτήρα υπογράμμισης "\_", ο οποίος ακολουθείται από οποιονδήποτε αλφαριθμητικό χαρακτήρα ή χαρακτήρα υπογράμμισης,. Άλλοι χαρακτήρες δεν είναι έγκυροι.

Παραδείγματα έγκυρων αναγνωριστικών:

- tablename
- table\_name
- \_Table9
- table9

Παραδείγματα μη έγκυρων αναγνωριστικών:

- table name
- table-name
- 9table

Οι χρήστες μπορούν όμως να χρησιμοποιούν άλλους χαρακτήρες αν εμπεριέχονται σε διπλά εισαγωγικά “ ” όπως στο παράδειγμα:

- `CREATE TABLE "table name"("column 1", " ");`

Η SQLite δεν έχει κανένα ανώτατο όριο για το μήκος του αναγνωριστικού ονόματος. Οι χρήστες μπορούν να χρησιμοποιήσουν οποιοδήποτε μήκος αναγνωριστικού μπορεί να είναι κατάλληλο για αυτούς.

### Δεσμευμένες λέξεις-κλειδιά

Ως γενικός κανόνας, θα πρέπει να αποφεύγεται η χρήση οποιονδήποτε λέξεων-κλειδιών της SQLite ως αναγνωριστικό από τους χρήστες. Αν όμως επιβάλλεται από τις καταστάσεις αυτό πρέπει να συμβεί με εξαιρετικά μεγάλη προσοχή. Για να συμβεί θα πρέπει ο χρήστης να χρησιμοποιήσει μια ειδική λέξη-κλειδί η οποία θα περικλείεται μέσα σε αγκύλες [ ].

Για παράδειγμα εδώ χρησιμοποιούμε τη δεσμευμένη λέξη “TABLE”:

- `CREATE TABLE [TABLE] ([SELECT], [INTEGER] INTEGER, [FROM], [TABLE]);`

Αναλυτικά στην σύνταξη που ακολουθείται σε κάθε εντολή ή δήλωση της SQLite θα αναφερθούμε σε κάθε ενότητα του E-Learning.

### 6.5.4 Ενότητα 4: Τύποι Δεδομένων (DATA TYPES)

Στην SQLite, κάθε πίνακας σε μια βάση δεδομένων έχει έναν τύπο δεδομένων και ένα όνομα. Ο τύπος δεδομένων ορίζει τι είδους δεδομένα μπορούν να αποθηκευτούν σε μια στήλη ενός πίνακα. Οι περισσότερες βάσεις δεδομένων χρησιμοποιούν ισχυρούς, στατικούς τύπους στήλης για πίνακες. Αυτό σημαίνει ότι η στήλη μπορεί να κρατήσει μόνο τιμή που είναι συμβατή με καθορισμένο τύπο στήλης. Η SQLite χρησιμοποιεί τη δυναμική τεχνική πληκτρολόγησης που είναι επίσης γνωστή ως Manifest Typing. Αυτό σημαίνει ότι επιτρέπει σχεδόν σε οποιοδήποτε στοιχείο οποιασδήποτε σειράς να κατέχει σχεδόν οποιοδήποτε τύπο τιμής.

#### Τάξεις αποθήκευσης στην SQLite

Η SQLite υποστηρίζει 5 συγκεκριμένους τύπους δεδομένων, οι οποίοι είναι επίσης γνωστοί ως κλάσεις αποθήκευσης. Ακολουθούν οι 5 κλάσεις αποθήκευσης που υποστηρίζονται από το SQLite:

| ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ                   | ΠΕΡΙΓΡΑΦΗ                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>NULL</b>                       | Η τιμή NULL θεωρείται ένας ξεχωριστός τύπος. Ένας τύπος NULL δεν έχει τιμή. Αντιπροσωπεύεται από τη λέξη-κλειδί NULL και κατέχει μόνο NULL.                                                                                                                                                                                      |
| <b>INTEGER – ΑΚΕΡΑΙΟΣ ΑΡΙΘΜΟΣ</b> | Η τιμή είναι ένας υπογεγραμμένος ακέραιος αριθμός με μήκος 8 Byte. Οι τιμές των ακεραίων είναι ολόκληροι αριθμοί δηλαδή μπορεί να είναι θετικές ή αρνητικές. Μπορούν να ποικίλουν σε μέγεθος: 1, 2, 3, 4, 6 ή 8 bytes. Οι ακέραιοι έχουν εύρος από -9,223,372,036,854,775,808 έως +9,223,372,036,854,775,807 ή περίπου 19 ψηφία. |

|                                   |                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>REAL - ΠΡΑΓΜΑΤΙΚΟΣ ΑΡΙΘΜΟΣ</b> | Ένας αριθμός κινητής υποδιαστολής, αποθηκευμένος ως τιμή 8-byte που περιέχει δεκαδικό σημείο ή εκθέτη. Οι αριθμοί κυμαινόμενου σημείου αντιπροσωπεύονται από οποιαδήποτε σειρά αριθμητικών ψηφίων που περιλαμβάνουν ένα δεκαδικό σημείο ή έναν εκθέτη. Το SQLite χρησιμοποιεί floats(τύπος κινητής υποδιαστολής) των 8-byte για την αποθήκευση πραγματικών αριθμών. Για παράδειγμα, τιμές όπως 23,5, -54,25, 5,87545, 2, κλπ. |
| <b>TEXT - ΚΕΙΜΕΝΟ</b>             | Οι τιμές κειμένου είναι δεδομένα χαρακτήρων μεταβλητού μήκους. Οι τιμές κειμένου αναπαρίστανται ως χαρακτήρες που περικλείονται μέσα σε μονά εισαγωγικά ' '. Η μέγιστη τιμή συμβολοσειράς στην SQLite είναι απεριόριστη. Για παράδειγμα, περιέχει τιμές όπως 'abc', '65xyz' κ.λπ                                                                                                                                              |
| <b>BLOB</b>                       | Μια τιμή BLOB είναι ακατέργαστα bytes μεταβλητού μήκους. Οι λέξεις BLOB αντιπροσωπεύονται ως δεκαεξαδικές συμβολοσειρές κειμένου που προηγούνται από ένα x. Τα δεδομένα BLOB (Binary Large Object - δυαδικά μεγάλα αντικείμενα) είναι οποιουδήποτε είδους δεδομένα. Η μέγιστη τιμή BLOB στην SQLite είναι απεριόριστη. Για παράδειγμα, διατηρεί τιμή όπως το x η συμβολοσειρά x'9856ABCD' αντιπροσωπεύει ένα BLOB 4-byte.     |

#### Τύποι Συνάφειας(Affinity Types) στην SQLite

Η SQLite υποστηρίζει την έννοια της συνάφειας τύπου σε στήλες. Κάθε στήλη μπορεί να αποθηκεύει οποιοδήποτε τύπο δεδομένων, αλλά η προτιμώμενη κλάση αποθήκευσης για μια στήλη ονομάζεται συνάφεια της. Η συνάφεια μιας στήλης προσδιορίζεται απευθείας από τον δηλωμένο τύπο. Επομένως, όταν δηλώνετε μια στήλη σε έναν πίνακα, ο τύπος που επιλέγετε να δηλώσετε, θα καθορίσει τελικά τη συνάφεια της στήλης. Αυτοί οι τύποι εισήχθησαν στην SQLite για να μεγιστοποιήσουν τη συμβατότητα μεταξύ SQLite και άλλου συστήματος διαχείρισης βάσεων δεδομένων. Σε κάθε στήλη πίνακα σε μια βάση δεδομένων SQLite3 εκχωρείται μία από τις 5 ακόλουθες συνάφειες τύπου:

| <b>ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ</b>              | <b>ΠΕΡΙΓΡΑΦΗ</b>                                                                                                                                                                                                |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TEXT</b>                         | Μια στήλη με αυτήν τη συνάφεια μπορεί να αποθηκεύσει μόνο τιμές τύπου TEXT, NULL ή BLOB. Εάν αποθηκεύσετε INTEGER σε αυτήν τη συνάφεια, τότε μετατρέπεται σε τύπο κειμένου.                                     |
| <b>NUMERIC - ΑΡΙΘΜΗΤΙΚΟ</b>         | Μια στήλη με αριθμητική συνάφεια θα αποθηκεύσει έναν από τους πέντε τύπους. Οι τιμές με τύπους integer και τύπου float, μαζί με τους τύπους NULL και BLOB, αποθηκεύονται χωρίς μετατροπή                        |
| <b>INTEGER - ΑΚΕΡΑΙΟΣ ΑΡΙΘΜΟΣ</b>   | Μια στήλη με συνάφεια με ακέραιους αριθμούς λειτουργεί ουσιαστικά όπως και η αριθμητική συνάφεια. Εάν αποθηκεύσετε την τιμή float, τότε μετατρέπεται σε ακέραιο τύπο.                                           |
| <b>FLOAT - ΚΙΝΗΤΗΣ ΥΠΟΔΙΑΣΤΟΛΗΣ</b> | Μια στήλη με συνάφεια κινητής υποδιαστολής λειτουργεί επίσης ουσιαστικά ίδια με την αριθμητική συνάφεια. Η μόνη διαφορά είναι εάν αποθηκεύετε την ακέραια τιμή τότε μετατρέπεται σε τιμή με κινητή υποδιαστολή. |
| <b>NONE</b>                         | Μια στήλη με καμία συνάφεια δεν έχει προτίμηση σε σχέση με την τάξη αποθήκευσης. Ονομάζεται συνάφεια BLOB.                                                                                                      |

Η SQLite εκχωρεί τη συνάφεια μιας στήλης σύμφωνα με τους ακόλουθους κανόνες:

- Από προεπιλογή, η συνάφεια μιας στήλης είναι NUMERIC. Δηλαδή, αν μια στήλη δεν είναι INTEGER, TEXT, ή NONE, τότε αυτόματα εκχωρείται NUMERIC συνάφεια.
- Αν ο δηλωμένος τύπος μιας στήλης περιέχει τη συμβολοσειρά «INT» ή «int», τότε η στήλη έχει εκχωρηθεί συνάφεια INTEGER.
- Εάν ο δηλωμένος τύπος μιας στήλης περιέχει οποιαδήποτε από τις συμβολοσειρές 'CHAR', 'BLOB' ή 'TEXT', τότε στη στήλη αυτή έχει εκχωρηθεί συνάφεια TEXT. Σημειώστε ότι το 'VARCHAR' περιέχει τη συμβολοσειρά 'CHAR' και έτσι θα δώσει συνάφεια TEXT.
- Εάν ο δηλωμένος τύπος μιας στήλης περιέχει τη συμβολοσειρά BLOB ή αν δεν έχει δηλωμένο τύπο, τότε έχει εκχωρηθεί συγγένεια NONE.

Κάθε συνάφεια επηρεάζει τον τρόπο αποθήκευσης των τιμών στη σχετική στήλη. Οι κανόνες κύριας αποθήκευσης έχουν ως εξής.

- Μια στήλη NUMERIC μπορεί να περιέχει και τις πέντε κατηγορίες αποθήκευσης. Αν προσπαθήσετε να εισαγάγετε τιμή TEXT στη στήλη NUMERIC, θα επιχειρήσει πρώτα να τη μετατρέψει σε κατηγορία αποθήκευσης INTEGER. Εάν αποτύχει να τη μετατρέψει, τότε την αποθηκεύει χρησιμοποιώντας την τάξη αποθήκευσης TEXT.
- Μια στήλη INTEGER προσπαθεί να συμπεριφέρεται σαν μια στήλη NUMERIC. Εάν προσπαθήσετε να εισαγάγετε τιμή REAL στη στήλη INTEGER, τότε θα την αποθηκεύσει ως REAL τύπου μόνο. Ωστόσο, αν το REAL δεν έχει δεκαδικό μέρος τότε θα αποθηκευτεί ως INTEGER. Στήλη INTEGER θα προσπαθήσει να αποθηκεύσει την τιμή TEXT ως REAL αν είναι δυνατόν. Αν όχι, θα προσπαθήσει να την αποθηκεύσει ως INTEGER. Εάν αποτύχει επίσης, τότε θα το αποθηκεύσει ως κείμενο.
- Μια στήλη TEXT θα μετατρέψει όλες τις τιμές INTEGER ή REAL σε TEXT.
- Μια στήλη NONE δεν επιχειρεί να μετατρέψει οποιεσδήποτε τιμές.
- Καμία στήλη δεν θα προσπαθήσει ποτέ να μετατρέψει τις τιμές NULL ή BLOB χωρίς να υπολογίζει τη συνάφεια. Οι τιμές NULL και BLOB αποθηκεύονται πάντα όπως είναι σε κάθε στήλη.

Ο παρακάτω πίνακας απαριθμεί διάφορα ονόματα τύπων δεδομένων τα οποία μπορούν να χρησιμοποιηθούν κατά τη δημιουργία πινάκων SQLite3 με την αντίστοιχη εφαρμοσμένη συνάφεια.

| ΤΥΠΟΣ ΔΕΔΟΜΕΝΩΝ                                                                                                                                                                                                      | ΣΥΝΑΦΕΙΑ                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| <ul style="list-style-type: none"> <li>• INT</li> <li>• INTEGER</li> <li>• TINYINT</li> <li>• SMALLINT</li> <li>• MEDIUMINT</li> <li>• BIGINT</li> <li>• UNSIGNED BIG INT</li> <li>• INT2</li> <li>• INT8</li> </ul> | <p style="text-align: center;"><b>INTEGER</b></p> |
| <ul style="list-style-type: none"> <li>• CHARACTER(20)</li> <li>• VARCHAR(255)</li> <li>• VARYING CHARACTER(255)</li> <li>• NCHAR(55)</li> <li>• NATIVE CHARACTER(70)</li> <li>• NVARCHAR(100)</li> </ul>            | <p style="text-align: center;"><b>TEXT</b></p>    |

|                                                                                                                                              |                |
|----------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <ul style="list-style-type: none"> <li>• TEXT</li> <li>• CLOB</li> </ul>                                                                     |                |
| <ul style="list-style-type: none"> <li>• BLOB</li> <li>• no datatype specified</li> </ul>                                                    | <b>NONE</b>    |
| <ul style="list-style-type: none"> <li>• REAL</li> <li>• DOUBLE</li> <li>• DOUBLE PRECISION</li> <li>• FLOAT</li> </ul>                      | <b>REAL</b>    |
| <ul style="list-style-type: none"> <li>• NUMERIC</li> <li>• DECIMAL (10,5)</li> <li>• BOOLEAN</li> <li>• DATE</li> <li>• DATETIME</li> </ul> | <b>NUMERIC</b> |

**Τύποι Δεδομένων Boolean (Αληθές ή Ψευδές):**

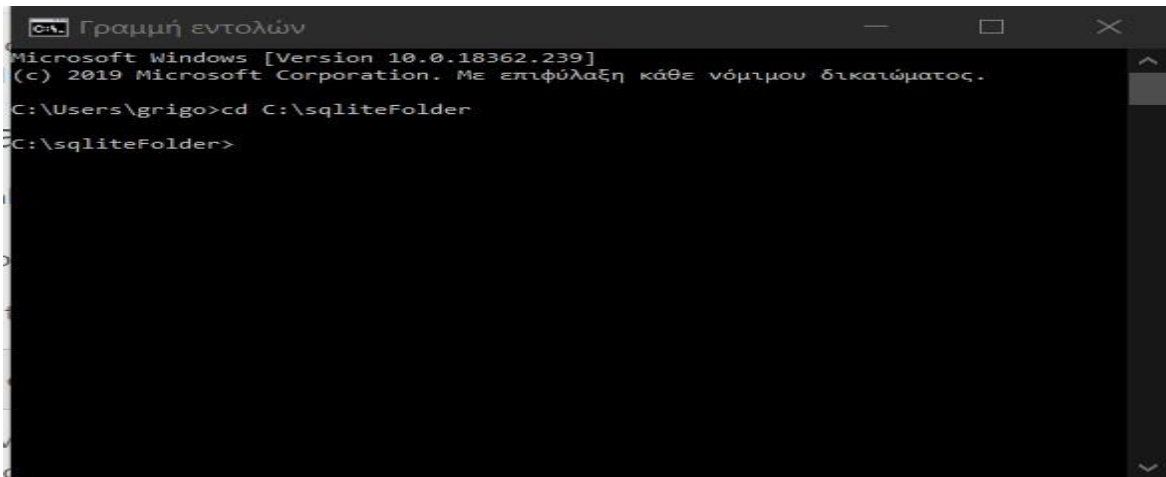
Η SQLite δεν διαθέτει ξεχωριστή κλάση αποθήκευσης για τύπους δεδομένων Boolean. Αντίθετα, οι τιμές Boolean αποθηκεύονται ως ακέραιοι αριθμοί 0 (ψευδείς) και 1 (αληθείς).

**Τύποι Δεδομένων Ημερομηνίας & Ώρας:**

Η SQLite δεν διαθέτει ξεχωριστή κλάση αποθήκευσης για την αποθήκευση ημερομηνιών ή και χρονικών στιγμών πχ ώρα, αλλά δύναται να αποθηκεύει ημερομηνίες και ώρες ως τιμές TEXT, REAL ή INTEGER.

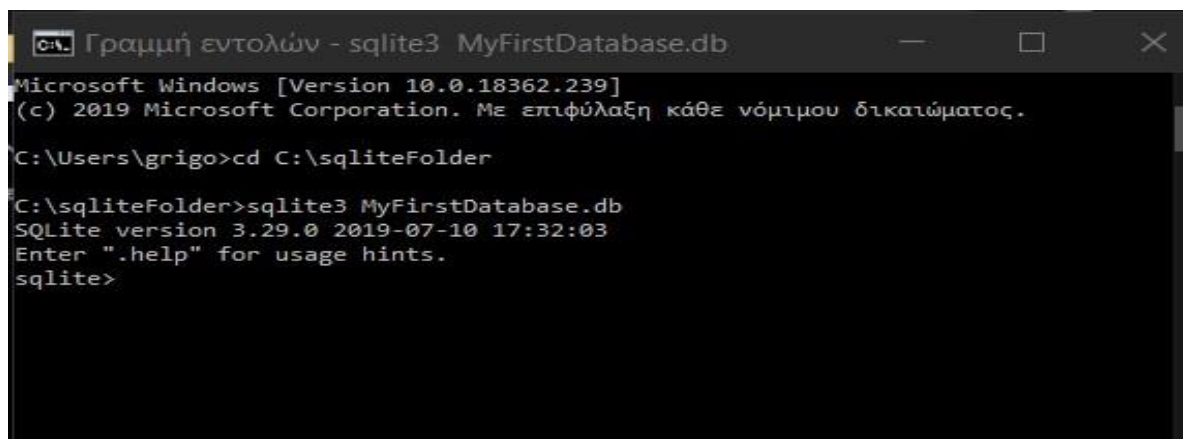
**6.5.5 Ενότητα 5: CREATE & DELETE DATABASE (ΔΗΜΙΟΥΡΓΙΑ & ΔΙΑΓΡΑΦΗ ΒΑΣΗΣ)****Δημιουργία της βάση στην SQLite**

Αντίθετα με τα άλλα συστήματα σχεσιακών βάσεων δεδομένων η SQLite δεν χρησιμοποιεί την εντολή CREATE DATABASE για να δημιουργήσει μια βάση δεδομένων. Η διαδικασία δημιουργίας της βάσης είναι ακόμα πιο απλή και ολόκληρη η βάση αποθηκεύεται σε ένα απλό αρχείο δίσκου που είναι συμβατό με όλες τις πλατφόρμες. Για να δημιουργήσουμε την βάση ανοίγουμε την γραμμή εντολών των Windows και μετακινούμαστε στον φάκελο που δημιουργήσαμε στην ενότητα Κατέβασμα & Εγκατάσταση και μέσα στον οποίο περιέχεται το αρχείο sqlite3.exe όπως στην επόμενη εικόνα.



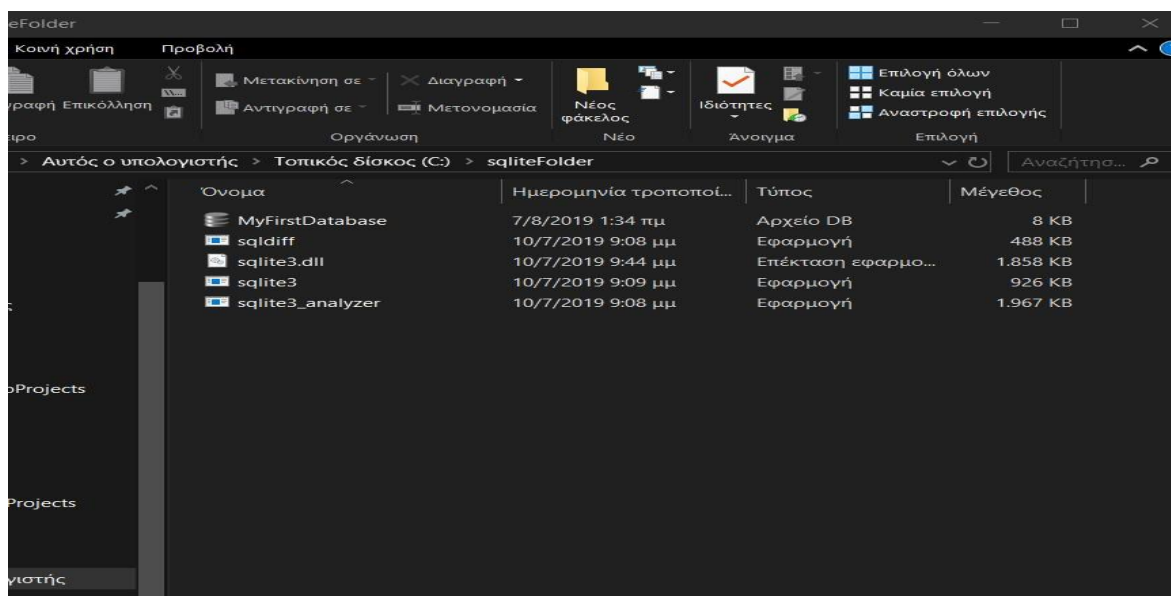
```
C:\Users\grigo>cd C:\sqliteFolder
C:\sqliteFolder>
```

Μόλις βρεθούμε στον φάκελο όπου είναι εγκατεστημένη η SQLite πληκτρολογούμε την παρακάτω εντολή δημιουργίας της βάσης: **sqlite3 MyFirstDatabase.db**

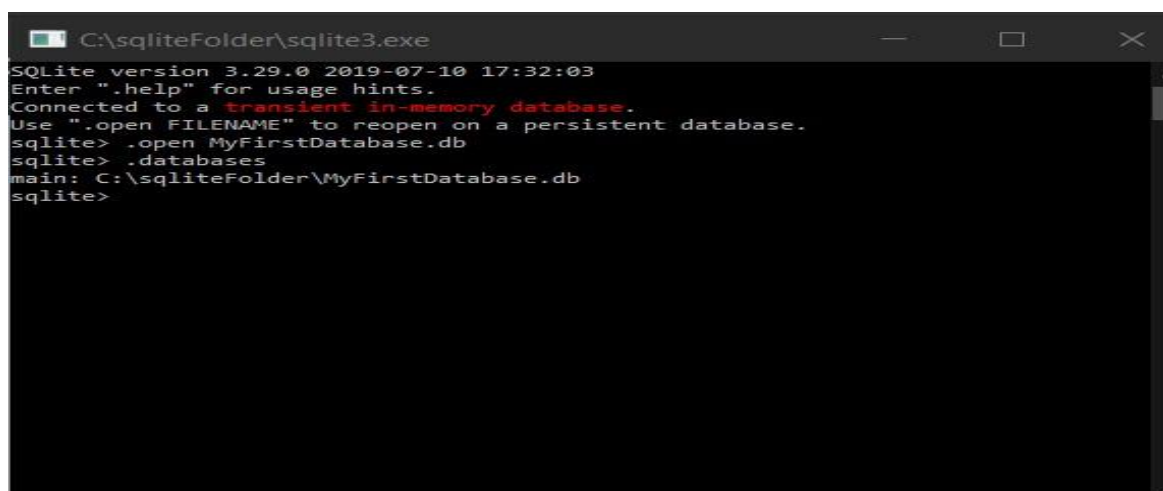


```
C:\Users\grigo>cd C:\sqliteFolder
C:\sqliteFolder>sqlite3 MyFirstDatabase.db
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
sqlite>
```

Με την πιο πάνω εντολή η SQLite δημιουργεί μια νέα βάση με το όνομα **MyFirstDatabase.db**. Αν υπάρχει ήδη βάση με αυτό το όνομα τότε απλά θα μας την ανοίξει. Μπορούμε να δούμε τη βάση που δημιουργήσαμε πηγαίνοντας στον φάκελο C:/sqliteFolder Όπως φαίνεται και στην επόμενη εικόνα:



Μπορούμε ακόμα να επιβεβαιώσουμε την δημιουργία της βάσης απ' την γραμμή εντολών των windows πληκτρολογώντας την εντολή **.databases** η οποία θα μας εμφανίσει όσες βάσεις έχουμε δημιουργήσει όπως στην εικόνα



```

C:\sqliteFolder\sqlite3.exe
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open MyFirstDatabase.db
sqlite> .databases
main: C:\sqliteFolder\MyFirstDatabase.db
sqlite>

```

Αν θέλουμε να δημιουργήσουμε νέα βάση σε φάκελο της επιλογής μας μπορούμε να πληκτρολογήσουμε την εντολή **.open C:/Users/User-PC/Desktop/newDb.db**. Η εντολή πριν δημιουργήσει τη νέα βάση ελέγχει στην διαδρομή που της δώσαμε αν προϋπάρχει βάση με αυτό το όνομα και αν τη βρει την ανοίγει αλλιώς την δημιουργεί.

#### Διαγραφή της Βάσης που δημιουργήσαμε

Οι περισσότερες σχεσιακές βάσεις δεδομένων χρησιμοποιούν την εντολή DROP DATABASE για την διαγραφή της βάσης δεδομένων η SQLite όμως ΌΧΙ. Για να διαγράψουμε τη βάση μας πρέπει απλά να πλοηγηθούμε στον φάκελο που την δημιουργήσαμε και χειροκίνητα να πατήσουμε διαγραφή του αρχείου.

#### 6.5.6 Ενότητα 6: CREATE Table (ΔΗΜΙΟΥΡΓΙΑ Πίνακα)

Η δήλωση CREATE TABLE στην SQLite χρησιμοποιείται για την δημιουργία ενός νέου πίνακα σε οποιαδήποτε βάση δεδομένων διαθέτουμε. Η δημιουργία ενός βασικού πίνακα απαιτεί από μας να δώσουμε ένα όνομα στον πίνακα που δημιουργούμε, να ορίσουμε τις στήλες του και τον τύπο δεδομένων κάθε στήλης του. Η βασική σύνταξη για την CREATE TABLE (δημιουργία πίνακα) είναι η εξής:

- **CREATE TABLE** [IF NOT EXISTS] [database\_name].table\_name (
  - column\_1 data\_type PRIMARY KEY,
  - column\_2 data\_type NOT NULL,
  - column\_3 data\_type DEFAULT 0,
 );

Στην παραπάνω σύνταξη ο όρος [IF NOT EXISTS] είναι προαιρετικός και χρησιμοποιείται για να πει στην SQLite να δημιουργήσει έναν νέο πίνακα μόνο αν ο πίνακας με το συγκεκριμένο όνομα δεν υπάρχει. Ακολουθεί το όνομα της βάσης στην οποία θα ανήκει ο πίνακας που φτιάχνουμε και το όνομα του νέου μας πίνακα. Κάθε στήλη που δημιουργούμε έχει το δικό της όνομα, τον δικό της τύπο δεδομένων και ένα προαιρετικό περιορισμό με το όνομα constraint. Η SQLite υποστηρίζει τους εξής περιορισμούς: PRIMARY KEY (Πρωτεύον κλειδί του πίνακα μας), UNIQUE (κάθε τιμή που εισάγεται πρέπει να είναι μοναδική), NOT NULL (τα δεδομένα της στήλης δεν μπορεί να είναι null) και CHECK (κάθε τιμή που εισάγεται ελέγχεται αν ικανοποιεί μια συνθήκη). Το πρωτεύον κλειδί του πίνακα είναι μια στήλη ή μια ομάδα στηλών που αναγνωρίζει με μοναδικό τρόπο μια σειρά στον πίνακα. Σε περίπτωση που το πρωτεύον κλειδί αποτελείται από πολλαπλές στήλες, πρέπει να χρησιμοποιήσετε τον περιορισμό πίνακα



αντί του περιορισμού πρωτεύοντος κλειδιού στήλης. Ακολουθεί ένα παράδειγμα που δείχνει την εφαρμογή των κανόνων σύνταξης:

- **CREATE TABLE** Students (  
     Id INTEGER PRIMARY KEY,  
     First\_name TEXT NOT NULL,  
     Last\_name TEXT NOT NULL,  
     Email text NOT NULL UNIQUE,  
     Phone text NOT NULL UNIQUE  
   );

### 6.5.7 Ενότητα 7: DROP Table (ΔΙΑΓΡΑΦΗ - ΚΑΤΑΣΤΡΟΦΗ του Πίνακα)

Η δήλωση DROP TABLE στην SQLite μας δίνει τη δυνατότητα να καταργήσουμε ή να διαγράψουμε έναν πίνακα από τη βάση δεδομένων μας. Μόλις διαγράψουμε τον πίνακα όλα τα δεδομένα που περιέχει αφαιρούνται οριστικά από τη βάση δεδομένων. Οτιδήποτε είναι συσχετισμένο με αυτόν το πίνακα όπως: ευρετήρια, ενεργοποιητές (triggers) και προδιαγραφές πρόσβασης διαγράφονται και αυτά επίσης. Αν υπάρχει ενεργοποιημένος περιορισμός ξένου κλειδιού στον πίνακα, τότε αυτό θα καταργηθεί ισοδύναμα για κάθε γραμμή του πίνακα. Αν όμως κατά την διαγραφή του πίνακα, προκύψει παραβίαση περιορισμού (constraint violation), τότε η δήλωση DROP TABLE αποτυγχάνει και δεσμεύει την τρέχουσα συναλλαγή. Η βασική σύνταξη για την διαγραφή πίνακα είναι η εξής:

- **DROP TABLE** [IF EXISTS] [database\_name].table\_name;

Μετά την δήλωση DROP TABLE ακολουθεί το όνομα του πίνακα που θέλουμε να διαγράψουμε. Στην παραπάνω σύνταξη ο όρος [IF EXISTS] είναι προαιρετικός και χρησιμοποιείται στην SQLite για να καταργήσει έναν πίνακα μόνο εάν υπάρχει αλλιώς αγνοεί τη δήλωση και δεν κάνει τίποτα. Αν προσπαθούσαμε να καταργήσουμε έναν πίνακα που δεν υπάρχει χωρίς να χρησιμοποιήσουμε τον όρο [IF EXISTS] η SQLite θα μας έβγαζε σφάλμα.

Στην εξ' ορισμού λειτουργία, η δήλωση DROP TABLE θα δημιουργήσει ελεύθερο χώρο στη βάση δεδομένων αφαιρώντας τον πίνακα και τα δεδομένα, αλλά δεν θα μειώσει το μέγεθος του αρχείου βάσης δεδομένων. Έτσι, για να αφαιρέσουμε τον ελεύθερο χώρο της βάσης δεδομένων, το SQLite χρησιμοποιεί την εντολή VACUUM. Εάν η λειτουργία AUTOVACUUM είναι ενεργοποιημένη για μια βάση δεδομένων, τότε ο χώρος θα είναι ελεύθερος αυτόματα από το DROP TABLE. Παραδείγματα διαγραφής πινάκων:

- **DROP TABLE** students; ή
- **DROP TABLE** [IF EXISTS] students; αν δεν είμαστε σίγουροι για την ύπαρξη ή μη του πίνακα

### 6.5.8 Ενότητα 8: INSERT (ΕΙΣΑΓΩΓΗ Δεδομένου (ων) σε Πίνακα)

Για να γεμίσουμε έναν πίνακα με δεδομένα στην SQLite χρησιμοποιούμε την δήλωση INSERT INTO. Υπάρχουν 2 βασικοί τρόποι σύνταξης στην SQLite για την INSERT INTO:

1. **INSERT INTO** TABLE\_NAME [(column1, column2, column3,...columnN)]  
**VALUES** (value1, value2, value3,...valueN);

Εδώ τα column1, column2, column3,...columnN είναι τα ονόματα των στηλών του πίνακα στις οποίες θέλουμε να εισάγουμε τις τιμές value1, value2, value3,...valueN για κάθε στήλη του πίνακα. Τα δεδομένα μέσα στις 2 λίστες χωρίζονται υποχρεωτικά με κόμμα. Για παράδειγμα:

- **INSERT INTO** students (Id, First\_name, Last\_name, Email, Phone)  
**VALUES** (1,'Sakis','Mpougas','mpougas@gmail.com',6974512879);

2. **INSERT INTO** TABLE\_NAME **VALUES** (value1,value2,value3,...valueN);

Στην δεύτερη μορφή σύνταξης της INSERT INTO δεν είναι αναγκαίο να καθορίσουμε τα ονόματα των στηλών του πίνακα AN εισάγουμε δεδομένα σε όλες τις στήλες του πίνακα. Θα πρέπει όμως να προσέξουμε η σειρά των τιμών που εισάγουμε να είναι στην ίδια σειρά με τις στήλες του πίνακα. Για παράδειγμα:

- **INSERT INTO** students  
**VALUES** (2,'Makis','Sakas',sakasm@gmail.com,6977943520);

Η σύνταξη και των 2 παραπάνω τρόπων γεμίζει με δεδομένα μια γραμμή του πίνακα κάθε φορά. Αν εμείς θέλουμε να εισάγουμε δεδομένα που καταλαμβάνουν πολλαπλές γραμμές του πίνακα σε μια δήλωση μπορούμε να το κάνουμε με την εξής σύνταξη:

- **INSERT INTO** TABLE\_NAME (column1,column2,column3,...columnN)  
**VALUES**  
(value1,value2, value3,...valueN),  
(value1,value2,value3,...valueN),  
.....  
(value1,value2,value3,...valueN);

### 6.5.9 Ενότητα 9: SELECT (ΕΠΙΛΟΓΗ – ΕΥΡΕΣΗ Δεδομένου(ων) σε Πίνακα)

Για να μπορέσουμε να «φέρουμε»(εξάγουμε) τα δεδομένα τα οποία υπάρχουν στους πίνακες της βάσης μας πρέπει να χρησιμοποιήσουμε την δήλωση SELECT. Ίσως η πιο κοινή και περισσότερο χρησιμοποιούμενη δήλωση όχι μόνο στην SQLite αλλά και στις υπόλοιπες γνωστές παραλλαγές της SQL. Είναι μια εντολή μόνο για ανάγνωση και επιστρέφει το σύνολο των αποτελεσμάτων με την μορφή πίνακα. Ακόμα μπορεί να χρησιμοποιηθεί για να εκτελέσουμε απλούς υπολογισμούς ή για την επιστροφή της τιμής απλών εκφράσεων. Η σύνταξη τη είναι πολύ απλή και μπορεί να γίνει ως εξής:

- **SELECT** column1, column2, columnN **FROM** table\_name

Εδώ ζητάμε από την SQLite να μας φέρει τα δεδομένα που υπάρχουν στις στήλες column1, column2, columnN(με αυτή την σειρά) του πίνακα table\_name.

Αν όμως επιθυμούμε τα δεδομένα από όλες τις στήλες του πίνακα μπορούμε να ακολουθήσουμε την παρακάτω σύνταξη:

- **SELECT** \* **FROM** table\_name;

Το σύμβολο \* εδώ υποδηλώνει πως θέλουμε όλα τα δεδομένα δηλαδή σαν να γράφαμε SELECT ALL FROM TABLE.

Όπως αναφέραμε νωρίτερα η SELECT μπορεί να χρησιμοποιηθεί και σε απλούς υπολογισμούς όπως για παράδειγμα:

- **SELECT** 1+1 → το οποίο μας επιστρέφει 2 σαν αποτέλεσμα

### 6.5.10 Ενότητα 10: Operators (Τελεστές)

Οι τελεστές στην SQLite είναι δεσμευμένες λέξεις ή χαρακτήρες που χρησιμοποιούνται στις δηλώσεις της SQLite κυρίως όταν χρησιμοποιούμε την παράμετρο WHERE για να εκτελέσουμε λειτουργίες όπως συγκρίσεις και αριθμητικές λειτουργίες. Οι τελεστές μπορούν να χρησιμοποιηθούν για να καθορίσουν τις συνθήκες και ο συνδυασμός τους για πολλαπλές συνθήκες στις δηλώσεις SQLite.

Υπάρχουν κυρίως 4 τύποι τελεστών στην SQLite:

- Αριθμητικοί τελεστές
- Τελεστές Σύγκρισης
- Λογικοί τελεστές
- Δυαδικοί (Bitwise) τελεστές

**Αριθμητικοί Τελεστές**Έχουμε 2 μεταβλητές  $\chi=20$  και  $\psi=10$ 

| ΤΕΛΕΣΤΗΣ                       | ΠΕΡΙΓΡΑΦΗ                                                                                                                | ΠΑΡΑΔΕΙΓΜΑ        |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------|-------------------|
| <b>+</b><br>(Πρόσθεση)         | Προσθέτει τις τιμές και από τις 2 πλευρές του τελεστή                                                                    | $\chi+\psi = 30$  |
| <b>-</b><br>(Αφαίρεση)         | Αφαιρεί τον όρο που βρίσκεται στα δεξιά του τελεστή από αυτόν που βρίσκεται στα αριστερά                                 | $\chi-\psi = 10$  |
| <b>*</b><br>(Πολλαπλασιασμός)  | Πολλαπλασιάζει τους όρους που βρίσκονται δεξιά και αριστερά του τελεστή                                                  | $\chi*\psi = 200$ |
| <b>/</b><br>(Διαίρεση)         | Διαιρεί τον όρο που βρίσκεται στα αριστερά του τελεστή με τον όρο που βρίσκεται στα δεξιά του                            | $\chi/\psi = 2$   |
| <b>%</b><br>(Ακέραιο Υπόλοιπο) | Διαιρεί τον όρο που βρίσκεται στα αριστερά του τελεστή με τον όρο που βρίσκεται στα δεξιά του και επιστρέφει το υπόλοιπο | $\chi\%\psi = 0$  |

**Τελεστές Σύγκρισης**Έχουμε 2 μεταβλητές  $\chi=20$  και  $\psi=10$ 

| ΤΕΛΕΣΤΗΣ        | ΠΕΡΙΓΡΑΦΗ                                                                                                      | ΠΑΡΑΔΕΙΓΜΑ                |
|-----------------|----------------------------------------------------------------------------------------------------------------|---------------------------|
| <b>==</b>       | Ελέγχει αν οι τιμές των δύο όρων είναι ίσες ή όχι, αν ναι τότε η συνθήκη γίνεται αληθής                        | $\chi==\psi$ είναι ψευδής |
| <b>=</b>        | Ελέγχει αν οι τιμές των δύο όρων είναι ίσες ή όχι, αν ναι τότε η συνθήκη γίνεται αληθής                        | $\chi=\psi$ είναι ψευδής  |
| <b>!=</b>       | Ελέγχει αν οι τιμές των δύο όρων είναι ίσες ή όχι, αν όχι τότε η συνθήκη γίνεται αληθής                        | $\chi!=\psi$ είναι αληθής |
| <b>&lt;&gt;</b> | Ελέγχει αν οι τιμές των δύο όρων είναι ίσες ή όχι, αν όχι τότε η συνθήκη γίνεται αληθής                        | $\chi<>\psi$ είναι αληθής |
| <b>&gt;</b>     | Ελέγχει αν η τιμή του όρου στα αριστερά του τελεστή είναι μεγαλύτερη από αυτή του όρου που βρίσκεται στα δεξιά | $\chi>\psi$ είναι αληθής  |
| <b>&lt;</b>     | Ελέγχει αν η τιμή του όρου στα αριστερά του τελεστή είναι μικρότερη από αυτή του όρου που βρίσκεται στα δεξιά  | $\chi<\psi$ είναι ψευδής  |

|              |                                                                                                                      |                            |
|--------------|----------------------------------------------------------------------------------------------------------------------|----------------------------|
| <b>&gt;=</b> | Ελέγχει αν η τιμή του όρου στα αριστερά του τελεστή είναι μεγαλύτερη - ίση από αυτή του όρου που βρίσκεται στα δεξιά | $x \geq \psi$ είναι αληθής |
| <b>&lt;=</b> | Ελέγχει αν η τιμή του όρου στα αριστερά του τελεστή είναι μικρότερη - ίση από αυτή του όρου που βρίσκεται στα δεξιά  | $x \leq \psi$ είναι ψευδής |

### Λογικοί Τελεστές

| ΤΕΛΕΣΤΗΣ       | ΠΕΡΙΓΡΑΦΗ                                                                                                                                                             |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>AND</b>     | Ο τελεστής AND επιτρέπει να ορίσουμε πολλαπλές συνθήκες στις εντολές SQLite και θα επιστρέψει εγγραφές μόνο όταν ισχύουν όλες οι καθορισμένες συνθήκες.               |
| <b>OR</b>      | Ο τελεστής OR επιτρέπει να ορίσουμε πολλαπλές συνθήκες στις εντολές SQLite και θα επιστρέψει εγγραφές εάν οποιαδήποτε από τις καθορισμένες προϋποθέσεις είναι αλήθεια |
| <b>IN</b>      | Χρησιμοποιώντας τον τελεστή IN μπορούμε να ελέγξουμε ότι η δεδομένη τιμή υπάρχει σε λίστα τιμών ή σε σύνολο αποτελεσμάτων υποερωτήματος.                              |
| <b>NOT IN</b>  | Το αντίθετο του τελεστή IN                                                                                                                                            |
| <b>BETWEEN</b> | Μπορούμε να επιστρέψουμε τιμές που υπάρχουν μεταξύ ορισμένου εύρους τιμών.                                                                                            |
| <b>EXISTS</b>  | Με τη χρήση του τελεστή EXISTS μπορούμε να ελέγξουμε αν υπάρχει η απαιτούμενη τιμή στο σύνολο αποτελεσμάτων ή όχι                                                     |
| <b>LIKE</b>    | Χρησιμοποιώντας τον τελεστή LIKE μπορούμε να έχουμε εγγραφές των οποίων οι τιμές ταιριάζουν με τη δεδομένη τιμή                                                       |
| <b>GLOB</b>    | Ίδια λειτουργία με τον τελεστή LIKE με τη διαφορά ότι ο τελεστής GLOB κάνει διάκριση μεταξύ πεζών και κεφαλαίων                                                       |
| <b>IS</b>      | Ο τελεστής IS λειτουργεί σαν τον τελεστή =                                                                                                                            |
| <b>IS NOT</b>  | Ο τελεστής IS NOT λειτουργεί σαν τον τελεστή !=                                                                                                                       |
| <b>UNIQUE</b>  | Ο τελεστής UNIQUE αναζητά κάθε γραμμή ενός καθορισμένου πίνακα για μοναδικότητα (χωρίς διπλότυπα)                                                                     |

### Διαδικικοί Τελεστές

| ΤΕΛΕΣΤΗΣ        | ΠΕΡΙΓΡΑΦΗ                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------|
| <b>&amp;</b>    | Ο δυαδικός τελεστής AND αντιγράφει 1 bit, αν υπάρχει και στους δύο όρους.                                         |
| <b> </b>        | Ο δυαδικός τελεστής OR αντιγράφει 1 bit, αν υπάρχει σε οποιονδήποτε από τους δύο όρους.                           |
| <b>~</b>        | Έχει ως αποτέλεσμα την "ανατροπή" των δυαδικών ψηφίων                                                             |
| <b>&lt;&lt;</b> | Θα μετακινήσει τον όρο που βρίσκεται στα αριστερά του τελεστή προς τα αριστερά σύμφωνα με τον αριθμό των δυαδικών |

|    |                                                                                                                                                               |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | ψηφίων που ορίζονται στο δεξιό όρο του τελεστή                                                                                                                |
| >> | Θα μετακινήσει τον όρο που βρίσκεται στα αριστερά του τελεστή προς τα δεξιά σύμφωνα με τον αριθμό των δυαδικών ψηφίων που ορίζονται στο δεξιό όρο του τελεστή |

### 6.5.11 Ενότητα 11: WHERE (ΟΠΟΥ)

Ο όρος WHERE χρησιμοποιείται για να καθορίσει μια συνθήκη κατά την φόρτωση των δεδομένων από έναν πίνακα ή από πολλούς πίνακες. Εάν η δεδομένη συνθήκη ικανοποιείται, τότε επιστρέφει τη συγκεκριμένη τιμή από τον πίνακα. Ο όρος WHERE χρησιμοποιείται για να φιλτράρει τα δεδομένα και να επιστρέψει στο χρήστη μόνο αυτά που πληρούν τη συνθήκη που θέσαμε με το WHERE. Μπορεί να χρησιμοποιηθεί όχι μόνο στη δήλωση SELECT, αλλά και στις UPDATE, DELETE κ.λπ., οι οποίες θα καλυφθούν στα επόμενα κεφάλαια. Η βασική σύνταξη της δήλωσης SELECT με τον όρο WHERE είναι η εξής:

- **SELECT** column1, column2, columnN  
**FROM** table\_name  
**WHERE** [condition(συνθήκη)];

Στο επόμενο παράδειγμα γίνεται λίγο πιο κατανοητή η χρήση του όρου WHERE:

Μας δίνεται ο πίνακας “Professors” με την μισθολογική κατάσταση των καθηγητών των τμημάτων ενός Πανεπιστημιακού Ιδρύματος.

| ID | First_name (Όνομα) | Last_name (Επώνυμο) | Department (Τμήμα) | Salary (Μισθός) |
|----|--------------------|---------------------|--------------------|-----------------|
| 1  | George             | Delaportas          | Biology            | 35.000          |
| 2  | John               | Agantas             | Informatics        | 45.000          |
| 3  | Timothy            | Dalton              | Economics          | 40.000          |
| 4  | Maria              | Tsekeri             | Mathematics        | 36.000          |

Και μας ζητείται να βρεθεί ο καθηγητής ή οι καθηγητές(Όνομα και Επώνυμο) με μισθό μεγαλύτερο των 40.000. Η σύνταξη η οποία θα μας φέρει τα δεδομένα που ζητάμε είναι η εξής:

- **SELECT** First\_name, Last\_name, Salary  
**FROM** Professors  
**WHERE** Salary > 40.000

| ID | First_name (Όνομα) | Last_name (Επώνυμο) | Department (Τμήμα) | Salary (Μισθός) |
|----|--------------------|---------------------|--------------------|-----------------|
| 2  | John               | Agantas             | Informatics        | 45.000          |

### 6.5.12 Ενότητα 12: UPDATE (ΕΝΗΜΕΡΩΣΗ Πίνακα)

Στην SQLite η δήλωση UPDATE χρησιμοποιείται για να τροποποιήσει τις υπάρχουσες εγγραφές σε ένα πίνακα. Η UPDATE μπορεί να ενημερώσει μία ή περισσότερες γραμμές ενός πίνακα αλλά όλες οι γραμμές πρέπει να είναι μέρος του ίδιου πίνακα. Στην περίπτωση που θέλουμε να ενημερώσουμε μια γραμμή στον πίνακα θα πρέπει να καθορίσουμε με τον όρο WHERE τη συνθήκη που θα πρέπει να ισχύει στην δήλωση UPDATE ώστε να γίνει η ενημέρωση της συγκεκριμένης εγγραφής για την οποία ικανοποιείται η συνθήκη που θέσαμε, αλλιώς η δήλωση UPDATE θα ενημερώσει όλες τις γραμμές του πίνακα. Η βασική σύνταξη της δήλωσης UPDATE για ενημέρωση μιας ή περισσότερων γραμμών είναι η ακόλουθη:

- **UPDATE** table\_name  
**SET** column1 = value1, column2 = value2....., columnN = valueN  
**WHERE** [condition];

Για παράδειγμα μας δίνεται ο πίνακας “Students” με τα στοιχεία κάποιων μαθητών και μας ζητείται να αλλάξουμε το τηλέφωνο του μαθητή με email «skolovos@gmail.com» διότι είχε καταχωρηθεί λάθος εξαρχής στο σωστό το οποίο είναι «6987102433»

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Phone<br>(Τηλέφωνο) | Email<br>(Ηλεκτρονικό<br>ταχυδρομείο) |
|----|-----------------------|------------------------|---------------------|---------------------------------------|
| 1  | Spiros                | Kolovos                | 6987102453          | skolovos@gmail.com                    |
| 2  | John                  | Macintosh              | 6974286535          | John_mac@hotmail.com                  |
| 3  | Mike                  | Corby                  | 6938526457          | CorbyM45@yahoo.com                    |
| 4  | Sarah                 | Parker                 | 6942512365          | parkerSarah@mail.com                  |

Η σύνταξη της δήλωσης UPDATE που θα τροποποιήσει την εγγραφή στον πίνακα είναι η εξής:

- **UPDATE** Students  
**SET** Phone = 6987102433  
**WHERE** Email = [skolovos@gmail.com](mailto:skolovos@gmail.com);

Πληκτρολογώντας:

- **SELECT** \*  
**FROM** Students  
**WHERE** Email = [skolovos@gmail.com](mailto:skolovos@gmail.com);

Βλέπουμε στον πίνακα ότι το τηλέφωνο έχει διορθωθεί

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Phone<br>(Τηλέφωνο) | Email<br>(Ηλεκτρονικό<br>ταχυδρομείο) |
|----|-----------------------|------------------------|---------------------|---------------------------------------|
| 1  | Spiros                | Kolovos                | 6987102433          | skolovos@gmail.com                    |

Στην προηγούμενη σύνταξη της UPDATE αν δεν χρησιμοποιούσαμε τον όρο WHERE τότε η δήλωση UPDATE θα τροποποιούσε όλη τη στήλη Phone του πίνακα Students με το νέο τηλέφωνο που θέσαμε:

- **UPDATE** Students  
**SET** Phone = 6987102433

Με **SELECT \*FROM** Students; έχουμε τον πίνακα:

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Phone<br>(Τηλέφωνο) | Email<br>(Ηλεκτρονικό<br>ταχυδρομείο) |
|----|-----------------------|------------------------|---------------------|---------------------------------------|
| 1  | Spiros                | Kolovos                | 6987102433          | skolovos@gmail.com                    |
| 2  | John                  | Macintosh              | 6987102433          | John_mac@hotmail.com                  |
| 3  | Mike                  | Corby                  | 6987102433          | CorbyM45@yahoo.com                    |
| 4  | Sarah                 | Parker                 | 6987102433          | parkerSarah@mail.com                  |

### 6.5.13 Ενότητα 13: DELETE (ΔΙΑΓΡΑΦΗ Δεδομένου (ων) Πίνακα)

Για να διαγράψουμε τις εγγραφές που υπάρχουν σε ένα πίνακα στην SQLite χρησιμοποιούμε την δήλωση DELETE. Αν χρησιμοποιήσουμε μαζί με την DELETE τον όρο WHERE τότε μπορούμε να καθορίσουμε τη γραμμή ή τις γραμμές που θα διαγραφούν αλλιώς η δήλωση DELETE θα διαγράψει όλες τις γραμμές του πίνακα. Η βασική σύνταξη της DELETE είναι η εξής:

- **DELETE**  
**FROM** table\_name

**WHERE** [condition];

Για παράδειγμα μας δίνεται ο πίνακας “Professors” με τη μισθολογική κατάσταση των καθηγητών ενός πανεπιστημιακού ιδρύματος και μας ζητείται να διαγράψουμε τον καθηγητή με Επώνυμο «Tsekeri» διότι μεταφέρθηκε σε νέο πανεπιστημιακό ίδρυμα.

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | George                | Delaportas             | Biology               | 35.000             |
| 2  | John                  | Agantas                | Informatics           | 45.000             |
| 3  | Timothy               | Dalton                 | Economics             | 40.000             |
| 4  | Maria                 | Tsekeri                | Mathematics           | 36.000             |

Η σύνταξη που θα διαγράψει την εγγραφή από τον πίνακα είναι η εξής:

- **DELETE**  
**FROM** Professors  
**WHERE** Last\_name = 'Tsekeri';

Με **SELECT \*FROM** Professors; έχουμε τον πίνακα:

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | George                | Delaportas             | Biology               | 35.000             |
| 2  | John                  | Agantas                | Informatics           | 45.000             |
| 3  | Timothy               | Dalton                 | Economics             | 40.000             |

Όπως και προηγούμενα στην δήλωση UPDATE έτσι και στην DELETE αν δεν χρησιμοποιήσουμε τον όρο WHERE ώστε να καθορίσουμε την συνθήκη η οποία πρέπει να ισχύει για να διαγραφεί μια ή πολλές εγγραφές του πίνακα μας, η δήλωση DELETE θα διαγράψει όλες τις εγγραφές του πίνακα.

- **DELETE**  
**FROM** Professors;

Με **SELECT \*FROM** Professors; έχουμε τον κενό από εγγραφές πίνακα:

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
|----|-----------------------|------------------------|-----------------------|--------------------|

#### 6.5.14 Ενότητα 14: AND (ΚΑΙ)

Ο Τελεστής AND(ΚΑΙ) επιτρέπει την ύπαρξη πολλαπλών συνθηκών σε μια δήλωση της SQLite όπου περιέχει τον όρο WHERE. Κατά την χρήση του τελεστή, η πλήρης συνθήκη που θέτουμε θα θεωρηθεί αληθής όταν ισχύουν όλες οι προϋποθέσεις. Για παράδειγμα, η συνθήκη [condition1] AND [condition2] θα είναι αληθής μόνο όταν και οι δύο προϋποθέσεις (condition1 και condition2) είναι αληθείς. Ο τελεστής AND χρησιμοποιείται γενικά με τις δηλώσεις SELECT, UPDATE, DELETE για να συνδυάσει πολλαπλές συνθήκες. Η βασική σύνταξη του τελεστή AND είναι η εξής:

- **SELECT** column1, column2, columnN  
**FROM** table\_name  
**WHERE** [condition1] AND [condition2]...AND [conditionN];

Αν για παράδειγμα στον επόμενο πίνακα «Employees» της μισθολογικής κατάστασης των εργαζομένων μιας εταιρείας μας ζητηθεί να διαγράψουμε έναν εργαζόμενο με Επώνυμο «Dalton» και Τμήμα στο οποίο εργάζεται το «H.R», η σύνταξη που θα ακολουθήσουμε είναι η εξής:

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |

- **DELETE**  
**FROM** Employees  
**WHERE** Last\_name = 'Dalton' **AND** Department = 'H.R' ;

### 6.5.15 Ενότητα 15: OR (Η)

Ομοίως με τον τελεστή AND έτσι και η χρήση του τελεστή OR(ή) επιτρέπει την ύπαρξη πολλαπλών συνθηκών σε μια δήλωση της SQLite όπου περιέχει τον όρο WHERE. Κατά την χρήση του τελεστή, η πλήρης συνθήκη που θέτουμε θα θεωρηθεί αληθής όταν ισχύουν μια από τις 2 προϋποθέσεις. Για παράδειγμα, η συνθήκη [condition1] OR [condition2] θα είναι αληθής όταν μια από τις 2 προϋποθέσεις (condition1 ή condition2) είναι αληθής. Ο τελεστής OR χρησιμοποιείται γενικά με τις δηλώσεις SELECT, UPDATE, DELETE για να συνδυάσει πολλαπλές συνθήκες. Η βασική σύνταξη του τελεστή AND είναι η εξής:

- **SELECT** column1, column2, columnN  
**FROM** table\_name  
**WHERE** [condition1] OR [condition2]...OR [conditionN];

Μας δίνεται ο επόμενος πίνακας «Employees» της μισθολογικής κατάστασης των εργαζομένων μιας εταιρείας και μας ζητείται να βρούμε τον υπάλληλο (Όνομα & Επίθετο) ή τους υπαλλήλους που έχουν μισθό μεγαλύτερο από 60.000 ή ανήκει(ουν) στο τμήμα «H.R»

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |

Η σύνταξη που θα μας επιστρέψει το ζητούμενο είναι η εξής:

- **SELECT** First\_name, Last\_name  
**FROM** Employees  
**WHERE** Salary > 60.000 **OR** Department = 'H.R';

| First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) |
|-----------------------|------------------------|
| Mat                   | Dalton                 |
| Peter                 | Dalton                 |

### 6.5.16 Ενότητα 16: LIKE ('ΟΜΟΙΟ ΜΕ')

Κάποιες φορές επιθυμούμε να εμφανίσουμε εγγραφές από τη βάση δεδομένων μας αλλά δεν θυμόμαστε με ακρίβεια την λέξη κλειδί η οποία θα μας επιτρέψει να βρούμε τα δεδομένα που ψάχνουμε. Για παράδειγμα σε έναν πίνακα που περιέχει ταινίες θέλουμε να βρούμε μια ταινία για την οποία θυμόμαστε μόνο ένα μέρος από το πλήρες όνομα της. Τη λύση σε τέτοια προβλήματα μας δίνει η χρήση του τελεστή LIKE. Ο τελεστής LIKE χρησιμοποιείται για να ελέγξει αν τιμή του αλφαριθμητικού που δόθηκε από τον χρήστη ταιριάζει με συγκεκριμένο



μοτίβο ή όχι. Αν η έκφραση αναζήτησης δηλαδή η τιμή του αλφαριθμητικού ταιριάζει με την τιμή του μοτίβου τότε ο τελεστής LIKE επιστρέφει αλήθεια. Κάτι που πρέπει να τονίσουμε είναι ότι ο τελεστής LIKE δεν διακρίνει πεζά από κεφαλαία γράμματα (case insensitive), και έτσι η έκφραση 'b' LIKE 'B' θα είναι πάντα αληθής. Η βασική σύνταξη του τελεστή LIKE με την δήλωση SELECT είναι η εξής:

- **SELECT** column\_list(λίστα στηλών)  
**FROM** table\_name(όνομα πίνακα)  
**WHERE** expression(έκφραση) **LIKE** pattern(μοτίβο)

expression(έκφραση): μπορεί να είναι μια στήλη ή ένα πεδίο του πίνακα

pattern(μοτίβο): έκφραση που περιέχει μοτίβο ταιριάσματος.

Σε συνδυασμό με τον τελεστή LIKE χρησιμοποιούνται 2 σύμβολα με την ονομασία WILDCARDS(ΜΠΑΛΑΝΤΕΡ) τα οποία είναι:

- % το σύμβολο του ποσοστού
- \_ και ο χαρακτήρας της υπογράμμισης

Το σύμβολο επί τοις εκατό αντιπροσωπεύει μηδέν, έναν ή πολλούς αριθμούς ή χαρακτήρες ενώ η υπογράμμιση αντιπροσωπεύει έναν μόνο αριθμό ή χαρακτήρα. Αυτά τα σύμβολα μπορούν να χρησιμοποιηθούν και σε συνδυασμούς. Ακολουθούν παραδείγματα βασικής σύνταξης του τελεστή LIKE με συνδυασμό των wildcards % και \_ :

1. **SELECT**  
**FROM** table\_name  
**WHERE** column LIKE 'XXXX%'  
Θα επιστρέψει όλες τις τιμές που ξεκινάνε με 'XXXX'.
2. **SELECT**  
**FROM** table\_name  
**WHERE** column LIKE '%XXXX%'  
Θα επιστρέψει όλες τις τιμές που περιέχουν τους χαρακτήρες 'XXXX'.
3. **SELECT**  
**FROM** table\_name  
**WHERE** column LIKE 'XXXX\_'  
Θα επιστρέψει όλες τις τιμές που ξεκινούν με 'XXXX' και περιέχουν 5 χαρακτήρες.
4. **SELECT**  
**FROM** table\_name  
**WHERE** column LIKE '\_X%'  
Θα επιστρέψει όλες τις τιμές που ο δεύτερος χαρακτήρα είναι το 'X' και θα πρέπει να περιέχουν τουλάχιστον 2 χαρακτήρες.
5. **SELECT**  
**FROM** table\_name  
**WHERE** column LIKE 'X\_X'  
Θα επιστρέψει όλες τις τιμές που ξεκινάνε με 'X' τελειώνουν με 'X' και περιέχουν συνολικά 4 χαρακτήρες.

Τα 'XXXX' θα μπορούσαν να έχουν τιμές οποιουδήποτε αριθμού ή αλφαριθμητικού.

### 6.5.17 Ενότητα 17: LIMIT (ΟΡΙΟ)

Όταν θέλουμε να βάλουμε ένα περιορισμό στον αριθμό των εγγραφών(δεδομένων) που θα μας επιστρέψει μια δήλωση στην SQLite χρησιμοποιούμε τον όρο LIMIT. Πολύ συχνά ο όρος LIMIT χρησιμοποιείται με τη δήλωση SELECT για την ανάκτηση γραμμών από έναν ή περισσότερους

πίνακες και τον περιορισμό του αριθμού των δεδομένων που επιστρέφονται βάσει της τιμής που θέσαμε στον όρο LIMIT. Στον όρο LIMIT καθορίζεται ο μέγιστος αριθμός γραμμών που επιστρέφεται από την δήλωση SELECT. Η βασική σύνταξη μέσα σε μια δήλωση SELECT είναι η εξής:

- **SELECT** column1, column2, columnN  
**FROM** table\_name  
[**WHERE** conditions]  
**LIMIT** no of rows(αριθμός γραμμών) [**OFFSET** row num(αριθμός γραμμής)];

Στην παραπάνω σύνταξη η γραμμή [**WHERE** conditions] είναι προαιρετική ανάλογα με τους επιπλέον περιορισμούς που θέλει να θέσει ο χρήστης στα δεδομένα που ψάχνει. Το ίδιο προαιρετική είναι και η χρήση του όρου [**OFFSET** row num(αριθμός γραμμής)] το οποίο μπορεί να χρησιμοποιηθεί για να καθορίσει πόσες γραμμές εγγραφών να παραλειφθούν στην αρχή του συνόλου αποτελεσμάτων με βάση την τιμή που θέσαμε στον όρο OFFSET. Αν υποθέσουμε για παράδειγμα ότι θέλουμε να πάρουμε 4 εγγραφές ξεκινώντας από την 4η σειρά των αποτελεσμάτων, τότε θα πρέπει να ορίσουμε την τιμή της OFFSET ως 4 και η LIMIT αρχίζει να επιλέγει σειρές από την 4η θέση.

Στο επόμενο παράδειγμα μας δίνεται ο πίνακας «Employees» και μας ζητείται να βρούμε τους 2 πρώτους υπαλλήλους με μισθό πάνω από 45.000.

| ID | First_name (Όνομα) | Last_name (Επώνυμο) | Department (Τμήμα) | Salary (Μισθός) |
|----|--------------------|---------------------|--------------------|-----------------|
| 1  | Gregory            | Georgopoulos        | Accounting         | 50.000          |
| 2  | Sam                | Tsitsipas           | IT                 | 60.000          |
| 3  | Mat                | Dalton              | H.R                | 48.000          |
| 4  | Peter              | Dalton              | Sales              | 80.000          |
| 5  | Sofia              | Kourtidou           | IT                 | 65.000          |
| 6  | Alan               | Harper              | Sales              | 75.000          |

Η σύνταξη που θα μας δώσει το ζητούμενο είναι η εξής:

- **SELECT** \*  
**FROM** Employees  
**WHERE** Salary > 45.000 **LIMIT** 2;

| ID | First_name (Όνομα) | Last_name (Επώνυμο) | Department (Τμήμα) | Salary (Μισθός) |
|----|--------------------|---------------------|--------------------|-----------------|
| 1  | Gregory            | Georgopoulos        | Accounting         | 50.000          |
| 2  | Sam                | Tsitsipas           | IT                 | 60.000          |

Αν τώρα μας είχε ζητηθεί να βρούμε τις 2 εγγραφές που βρίσκονται στην 3<sup>η</sup> και 4<sup>η</sup> θέση του πίνακα θα μπορούσαμε να χρησιμοποιήσουμε την σύνταξη με OFFSET ως εξής:

- **SELECT** \*  
**FROM** Employees **LIMIT** 2 **OFFSET** 2;

| ID | First_name (Όνομα) | Last_name (Επώνυμο) | Department (Τμήμα) | Salary (Μισθός) |
|----|--------------------|---------------------|--------------------|-----------------|
| 3  | Mat                | Dalton              | H.R                | 48.000          |
| 4  | Peter              | Dalton              | Sales              | 80.000          |

### 6.5.18 Ενότητα 18: MIN (ΕΛΑΧΙΣΤΟ) - MAX (ΜΕΓΙΣΤΟ)

Η συναρτήσεις MIN() & MAX() αποτελούν συναθροιστικές συναρτήσεις που χρησιμοποιούνται για την εύρεση της ελάχιστης και της μέγιστη τιμής αντίστοιχα, μιας έκφρασης ή μιας στήλης. Γενικά οι συναρτήσεις λειτουργούν με τιμές που δεν είναι NULL ώστε να μπορούν να επιστρέψουν την ελάχιστη ή μέγιστη τιμή μιας στήλης του πίνακα. Αν όμως αυτή η στήλη του πίνακα έχει μόνο NULL τιμές τότε και οι MIN(), MAX() θα επιστρέψουν τιμή NULL. Η βασική σύνταξη είναι η εξής:

- `SELECT MIN(aggregate_expression (συναθροιστική έκφραση))  
FROM table_name  
[WHERE conditions];`
- `SELECT MAX(aggregate_expression (συναθροιστική έκφραση))  
FROM table_name  
[WHERE conditions];`

Στην παραπάνω σύνταξη η γραμμή `[WHERE conditions];` είναι προαιρετική ανάλογα με τους επιπλέον περιορισμούς που θέλει να θέσει ο χρήστης στα δεδομένα που ψάχνει. Οι MIN() & MAX() μπορούν χρησιμοποιηθούν και με τους όρους ταξινόμησης ORDER BY και GROUP BY.

Εδώ θα πρέπει να αναφερθούμε σε μια δυνατότητα που μας δίνει η SQLite να μετονομάσουμε προσωρινά ένα πίνακα ή μια στήλη δίνοντας ένα ψευδώνυμο γνωστό και ως **ALIAS**. Η μετονομασία είτε του πίνακα είτε κάποιας στήλης είναι μια προσωρινή διαδικασία και το πραγματικό όνομα του πίνακα ή της στήλης δεν αλλάζει στη βάση μας. Αυτό το κάνουμε είτε για διευκόλυνση δική μας, είτε για πιο όμορφο τελικό αποτέλεσμα στη δημιουργία των πινάκων με τις εγγραφές που αναζητάμε, είτε γιατί μας έχει ζητηθεί το όνομα κάποιου πίνακα η στήλης να είναι συγκεκριμένο στα αποτελέσματα που θα φαίνονται. Η βασική σύνταξη είναι η εξής:

- `SELECT column1, column2....  
FROM table_name AS alias_name  
WHERE [condition];`
- `SELECT column_name AS alias_name  
FROM table_name  
WHERE [condition];`

Για το επόμενο παράδειγμα μας δίνεται ο πίνακας «Employees» και μας ζητείται να βρούμε τον μεγαλύτερο και τον μικρότερο μισθό του πίνακα καθώς και σε ποια τμήματα της εταιρείας ανήκουν τα 2 ποσά.

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |

Για τον μικρότερο μισθό η σύνταξη είναι ως εξής:

- `SELECT Department, MIN(Salary)as MINIMUM_SALARY  
FROM Employees`

| Department<br>(Τμήμα) | MINIMUM_SALARY<br>(ALIAS) |
|-----------------------|---------------------------|
| H.R                   | 48.000                    |

Για τον μεγαλύτερο μισθό η σύνταξη είναι ως εξής:

- `SELECT Department, MAX(Salary)as MAXIMUM_SALARY  
FROM Employees`

| Department<br>(Τμήμα) | MAXIMUM_SALARY<br>(ALIAS) |
|-----------------------|---------------------------|
| Sales                 | 80.000                    |

### 6.5.19 Ενότητα 19: SUM (ΑΡΙΘΜΗΤΙΚΟ ΑΘΡΟΙΣΜΑ)

Η συνάρτηση SUM () είναι μια συναθροιστική συνάρτηση που χρησιμοποιείται για τον υπολογισμό του αθροίσματος των τιμών σε μια συγκεκριμένη έκφραση ή στήλη. Η συνάρτηση SUM() λειτουργεί με μη NULL αριθμητικές τιμές ώστε να μπορέσει να επιστρέψει το άθροισμα των τιμών μιας στήλης του πίνακα. Αν όμως όλες οι τιμές της συγκεκριμένης στήλης είναι NULL τότε και η SUM() θα επιστρέψει τιμή NULL σαν αποτέλεσμα. Όπως και οι συναρτήσεις MIN() & MAX() έτσι και η SUM() μπορεί να συνδυαστεί με τους όρους ORDER BY & GROUP BY για ακόμα μεγαλύτερο έλεγχο των αποτελεσμάτων. Η βασική σύνταξη της SUM()είναι η εξής:

- `SELECT SUM(aggregate_expression (συναθροιστική έκφραση))  
FROM table_name  
[WHERE conditions];`

Στην παραπάνω σύνταξη η γραμμή `[WHERE conditions];` είναι προαιρετική ανάλογα με τους επιπλέον περιορισμούς που θέλει να θέσει ο χρήστης στα δεδομένα που ψάχνει. Στο επόμενο παράδειγμα μας ζητείται από τον πίνακα «Employees» η συνολική μισθοδοσία των υπαλλήλων της εταιρίας.

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |

Η σύνταξη για να το πετύχουμε αυτό είναι η εξής:

- `SELECT Department, SUM(Salary) as TOTAL_SALARY  
FROM Employees`

| TOTAL_SALARY<br>(ALIAS) |
|-------------------------|
| 378.000                 |

### 6.5.20 Ενότητα 20: DISTINCT (ΔΙΑΚΡΙΤΟ) & COUNT (ΠΛΗΘΟΣ)

Στην SQLite η λέξη κλειδί DISTINCT χρησιμοποιείται για να φιλτράρει τις στήλες και τις τιμές που καθορίζονται στη δήλωση της SELECT και να καθορίσει αν μια γραμμή είναι διπλότυπο κάποιας άλλης. Αν για παράδειγμα ορίσουμε την DISTINCT για μια στήλη του πίνακα μας σε μια δήλωση της SELECT θα μας επιστρέψει τις μοναδικές (unique) τιμές αυτής της στήλης.

Στην SQLite οι NULL τιμές θεωρούνται διπλότυπα έτσι εάν χρησιμοποιήσουμε την ρήτρα DISTINCT σε μια στήλη που περιέχει τιμές NULL, τότε θα διατηρήσει μόνο μια γραμμή τη τιμή NULL. Η βασική σύνταξη είναι η εξής:

- **SELECT DISTINCT** column1, column2,.....columnN  
**FROM** table\_name  
**WHERE** [condition];

Η συναθροιστική συνάρτηση COUNT() χρησιμοποιείται στην SQLite για να μας επιστρέψει τον συνολικό αριθμό των γραμμών ενός πίνακα με βάση μια καθορισμένη έκφραση ή συνθήκη. Η συνάρτηση COUNT () λειτουργεί με μη NULL τιμές ώστε να μπορέσει να επιστρέψει τον σύνολο των φορών που εμφανίζονται μη NULL τιμές μιας στήλης του πίνακα. Όπως και οι συναρτήσεις MIN(), MAX(), SUM() έτσι και η COUNT() μπορεί να συνδυαστεί με τους όρους ORDER BY & GROUP BY για ακόμα μεγαλύτερο έλεγχο των αποτελεσμάτων. Η βασική σύνταξη της είναι η εξής:

- **SELECT COUNT( [ALL | DISTINCT] aggregate\_expression)**  
**FROM** table\_name  
**[WHERE** conditions];
- **SELECT COUNT(\*)**  
**FROM** table\_name
- **SELECT** column1, column2, columnN  
**COUNT(aggregate\_expression)**  
**FROM** table\_name  
**[WHERE** conditions]  
**GROUP BY** column1, column2, columnN;

Η γραμμή **[WHERE conditions];** είναι προαιρετική και μπορεί να χρησιμοποιηθεί αν κριθεί αναγκαίο από τον χρήστη για επιπλέον φιλτράρισμα των αποτελεσμάτων. Εξ' ορισμού η SQLite στην συνάρτηση COUNT() χρησιμοποιεί τη λέξη κλειδί **ALL** για να μετρήσει το πλήθος των γραμμών ενός πίνακα βασισμένη σε μια δήλωση και γι' αυτό η λέξη ALL είναι προαιρετική και δεν χρειάζεται να την γράφουμε ξεχωριστά. Αντίθετα αν χρειαζόμαστε το πλήθος μοναδικών τιμών τότε είναι επιβεβλημένη η χρήση της DISTINCT. Στο επόμενο παράδειγμα μας δίνεται ο πίνακας «Employees» και μας ζητείται αρχικά να βρούμε το πλήθος των εργαζομένων του πίνακα και κατόπιν να βάλουμε σε αλφαβητική σειρά τα επίθετα των εργαζομένων αγνοώντας κάποιο ή κάποια επίθετα που εμφανίζονται πάνω από μία φορές.

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |

Για να βρούμε το πλήθος των εργαζομένων του πίνακα γράφουμε το εξής:

- **SELECT COUNT(\*) AS Total\_Number\_Of\_Employees**  
**FROM** Employees;

| Total_Number_Of_Employees<br>(ALIAS) |
|--------------------------------------|
| 6                                    |

Ενώ για να βάλουμε σε αλφαβητική σειρά τα Επίθετα χωρίς διπλότυπα γράφουμε το εξής:

- `SELECT DISTINCT>Last_name)`  
`FROM Employees;`  
`ORDER BY Last_name;`

| Last_name<br>(Επώνυμο) |
|------------------------|
| Dalton                 |
| Georgopoulos           |
| Harper                 |
| Kourtidou              |
| Tsitsipas              |

### 6.5.21 Ενότητα 21: AVG (ΜΕΣΟΣ ΟΡΟΣ)

Στην SQLite η συναθροιστική συνάρτηση AVG() χρησιμοποιείται για να υπολογίσει την μέση τιμή μιας έκφρασης. Η συνάρτηση λειτουργεί με μη NULL αριθμητικές τιμές για να μπορέσει να επιστρέψει τη μέση τιμή βάσει μιας καθορισμένης δήλωσης. Σε περίπτωση που χρησιμοποιήσουμε την AVG() με μη αριθμητικές τιμές όπως TEXT ή BLOB θα τις θεωρήσει ως μηδέν ενώ αν όλοι οι αριθμοί είναι NULL τότε και η συνάρτηση θα επιστρέψει τιμή NULL. Η τιμή της συνάρτησης AVG είναι πάντοτε τιμή κινητής υποδιαστολής ή τιμή NULL Η βασική σύνταξη είναι η εξής:

- `SELECT AVG(DISTINCT | ALL Expression)`  
`FROM table_name`  
`[WHERE conditions]`  
`[GROUP BY Expression];`

Οι γραμμές `[WHERE conditions]` και `[GROUP BY Expression]` είναι προαιρετικές και χρησιμοποιούνται αν ο χρήστης θέλει να φιλτράρει ή να ταξινομήσει τα αποτελέσματα που θα του επιστραφούν. Εξ' ορισμού η συνάρτηση AVG() χρησιμοποιεί τη λέξη-κλειδί `ALL` για να υπολογίσει το μέσο όρο οπότε μπορούμε να την παραλείψουμε, αντιθέτως αν πρέπει να υπολογίσουμε τον μέσο όρο τιμών που είναι μοναδικές τότε η χρήση της λέξης `DISTINCT` είναι απαραίτητη. Για να υπολογίσουμε στην πράξη τον μέσο όρο κάποιων τιμών θα χρησιμοποιήσουμε την AVG() στον επόμενο πίνακα «Employees». Μας ζητείται ο μέσος μισθός ανά τμήμα των εργαζομένων της εταιρείας σε φθίνουσα ταξινόμηση.

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |

Για να υπολογίσουμε τον μέσο μισθό των τμημάτων γράφουμε το εξής:

- `SELECT Department, AVG(Salary) AS Average_Salary`  
`FROM Employees`  
`GROUP BY Department`  
`ORDER BY Average_Salary DESC;`

| Department<br>(Τμήμα) | Average_Salary<br>(ALIAS) |
|-----------------------|---------------------------|
| Sales                 | 77.500,0                  |

|            |          |
|------------|----------|
| IT         | 62.500,0 |
| Accounting | 50.000,0 |
| H.R        | 48.000,0 |

### 6.5.22 Ενότητα 22: ORDER BY (ΤΑΞΙΝΟΜΗΣΗ ΜΕ ΣΕΙΡΑ)

Στην SQLite τα δεδομένα αποθηκεύονται στους πίνακες με απροσδιόριστη σειρά και με την ίδια απροσδιόριστη σειρά γίνεται και η ανάκτηση των εγγραφών χρησιμοποιώντας την δήλωση SELECT. Αν θέλουμε όμως να ταξινομήσουμε το σύνολο των αποτελεσμάτων που θα μας επιστρέψει μια δήλωση SELECT μπορούμε να χρησιμοποιήσουμε τον όρο ORDER BY. Η ταξινόμηση μπορεί να γίνει είτε με αύξουσα είτε με φθίνουσα σειρά βάσει των απαιτήσεων μας. Η βασική σύνταξη της ORDER BY είναι η εξής:

- **SELECT** column-list (λίστα στηλών)  
**FROM** table\_name  
**[WHERE** condition]  
**[ORDER BY** column1, column2, .. columnN] **[ASC | DESC];**

Στην παραπάνω σύνταξη η γραμμή **[WHERE** condition] είναι προαιρετική ανάλογα με τους επιπλέον περιορισμούς που θέλει να θέσει ο χρήστης στα δεδομένα που ψάχνει. Στην σύνταξη της ORDER BY μπορούν να χρησιμοποιηθούν παραπάνω από μία στήλες αλλά πρέπει να είμαστε προσεκτικοί οι στήλες ή η στήλη που χρησιμοποιήσαμε για την ταξινόμηση να υπάρχουν και στην λίστα των στηλών (column-list). Εξ ' ορισμού δηλαδή αν δεν δηλώσουμε ASC για αύξουσα ταξινόμηση ή DESC για φθίνουσα ταξινόμηση η SQLite χρησιμοποιεί την αύξουσα ταξινόμηση. Για παράδειγμα μας δίνεται ο πίνακας «Employees» με την μισθολογική κατάσταση των εργαζομένων μιας εταιρείας και μας ζητείται να ταξινομηθούν βάσει αύξουσας μισθολογικής κατάστασης.

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |

Η σύνταξη που θα μας επιτρέψει να το κάνουμε αυτό είναι η εξής:

- **SELECT** \*  
**FROM** Employees  
**ORDER BY** Salary **ASC**  
Το **ASC** μπορεί να παραληφθεί και να έχουμε πάλι το ίδιο αποτέλεσμα

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |

### 6.5.23 Ενότητα 23: GROYP BY (ΟΜΑΔΟΠΟΙΗΣΗ)

Στην SQLite μπορούμε να ομαδοποιήσουμε ταυτόσημα δεδομένα χρησιμοποιώντας τον όρο GROUP BY. Ο όρος χρησιμοποιείται μαζί με την δήλωση της SELECT για την ομαδοποίηση των δεδομένων σε μια μόνο σειρά όταν οι τιμές μιας ή περισσότερων στηλών επαναλαμβάνονται. Αυτή η δυνατότητα που μας δίνει η SQLite μπορεί να χρησιμοποιηθεί για να ελαττώσει τον αριθμό των εγγραφών έτσι ώστε να βρεθούν οι τιμές σε μια στήλη οι οποίες είναι μοναδικές. Ο όρος GROUP BY μπορεί να χρησιμοποιηθεί σε μια δήλωση SELECT αμέσως μετά τον όρο WHERE, προηγείται όμως του όρου ORDER BY. Η βασική σύνταξη της GROYP BY είναι η εξής:

- `SELECT column-list  
FROM table_name  
WHERE [ conditions ]  
GROUP BY column1, column2....columnN`

Και εδώ η γραμμή `WHERE [conditions]` είναι προαιρετική ανάλογα με τους επιπλέον περιορισμούς που θέλει να θέσει ο χρήστης στα δεδομένα που ψάχνει ενώ μπορούμε να χρησιμοποιήσουμε παραπάνω από μια στήλες με την GROUP BY απλά πρέπει να προσέξουμε να είναι διαθέσιμες στην δήλωση SELECT. Η GROUP BY μπορεί να συνδυαστεί και με την ORDER BY αλλά και με συναθροιστικές συναρτήσεις όπως MIN, MAX, SUM, COUNT, AVG ώστε να παρέχει περισσότερες πληροφορίες για κάθε ομάδα που ομαδοποιεί.

- `SELECT column-list  
FROM table_name  
WHERE [ conditions ]  
GROUP BY column1, column2....columnN  
ORDER BY column1, column2....columnN`
- `SELECT column1, MAX(column2)  
FROM table_name  
WHERE [ conditions ]  
GROUP BY column1  
ORDER BY MAX(column2)`

Στο επόμενο παράδειγμα μας δίνεται ο πίνακας «Employees» και μας ζητείται να βρούμε το τμήμα της εταιρείας που συνολικά κοστίζει το περισσότερο σε μισθούς.

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |

Η σύνταξη για να το πετύχουμε αυτό είναι:

- `SELECT Department, SUM(Salary) AS TOTAL_SALARY  
FROM Employees  
GROUP BY Department  
ORDER BY TOTAL_SALARY DESC LIMIT 1;`

| Department<br>(Τμήμα) | TOTAL_SALARY |
|-----------------------|--------------|
| Sales                 | 155.000      |



### 6.5.24 Ενότητα 24: HAVING ()

Σαν λειτουργία ο όρος HAVING() είναι πανομοιότυπος με τον όρο WHERE. Ο όρος HAVING() είναι άλλη μια συνθήκη που εφαρμόζεται μετά από την εφαρμογή μιας συνάθροισης μαζί με τον όρο GROUP BY σε μια δήλωση SELECT. Γενικά ο όρος WHERE χρησιμοποιείται για να εφαρμόσει μια συνθήκη σε μεμονωμένα στοιχεία ενός πίνακα ενώ ο όρος HAVING() χρησιμοποιείται για να προσθέσει επιπλέον συνθήκες φιλτραρίσματος στις ομάδες που δημιουργήθηκαν από την GROUP BY. Είναι αναγκαίο να χρησιμοποιούμε τον όρο HAVING με την ομαδοποίηση GROUP BY ειδάλως θα λειτουργούσε όπως ο όρος WHERE. Ομαδοποιώντας τις σειρές ενός πίνακα με την GROUP BY σε σύνολο ομάδων η εφαρμογή του όρου HAVING θα εφαρμόσει επιπλέον φίλτρα στις ομάδες που δημιουργήθηκαν. Η σύνταξη του όρου HAVING σε μια δήλωση SELECT είναι η εξής:

- `SELECT column_1, column_2`  
`FROM table_name1, table_name2`  
`WHERE [ conditions ]`  
`GROUP BY column_1, column_2`  
`HAVING [ conditions ]`  
`ORDER BY column_1, column_2;`

Αν στον επόμενο πίνακα των «Employees» μας ζητηθεί να βρούμε τα τμήματα της εταιρείας που εργάζονται πάνω από 1 άτομα η σύνταξη που ακολουθούμε είναι η εξής:

| ID | First_name<br>(Όνομα) | Last_name<br>(Επώνυμο) | Department<br>(Τμήμα) | Salary<br>(Μισθός) |
|----|-----------------------|------------------------|-----------------------|--------------------|
| 1  | Gregory               | Georgopoulos           | Accounting            | 50.000             |
| 2  | Sam                   | Tsitsipas              | IT                    | 60.000             |
| 3  | Mat                   | Dalton                 | H.R                   | 48.000             |
| 4  | Peter                 | Dalton                 | Sales                 | 80.000             |
| 5  | Sofia                 | Kourtidou              | IT                    | 65.000             |
| 6  | Alan                  | Harper                 | Sales                 | 75.000             |

- `SELECT Department, COUNT(*)`  
`FROM Employees`  
`GROUP BY Department`  
`HAVING COUNT(*) > 1;`

| Department | COUNT(*) |
|------------|----------|
| IT         | 2        |
| SALES      | 2        |

### 6.5.25 Ενότητα 25: JOIN (ΕΝΩΣΗ ΠΙΝΑΚΩΝ)

Στην SQLite ο όρος JOIN χρησιμοποιείται για να συνδυάσει τις εγγραφές από δύο ή περισσότερους πίνακες της βάσης δεδομένων. Ενώνει τα πεδία από δύο πίνακες χρησιμοποιώντας τις τιμές που έχουν κοινές και στους δύο πίνακες. Γενικά με τον όρο JOIN στην SQLite θα ενώνουμε δύο πίνακες σε έναν άλλο πίνακα. Σε περίπτωση όμως που έχουμε τρεις ή περισσότερους πίνακες τότε μπορούμε να τους ενώσουμε χρησιμοποιώντας τον όρο JOIN πολλές φορές σε πολλές δηλώσεις SQLite. Υπάρχουν κυρίως τρεις τύποι ενώσεων με την χρήση του όρου JOIN στην SQLite:

- SQLite INNER JOIN
- SQLite OUTER JOIN
- SQLite CROSS JOIN

**INNER JOIN (ΕΣΩΤΕΡΙΚΗ ΕΝΩΣΗ)**

Η INNER JOIN δημιουργεί έναν νέο πίνακα αποτελεσμάτων ο οποίος περιέχει τα αποτελέσματα που ικανοποιούν τη συνθήκη JOIN για την ένωση του πίνακα A με τον πίνακα B. Συγκρίνει κάθε γραμμή του πίνακα A με κάθε γραμμή του πίνακα B για να βρει όλα τα ζευγάρια γραμμών που ικανοποιούν τη συνθήκη JOIN σε μια δήλωση SQLite. Όταν πληρούνται οι συνθήκες ένωσης τα ταιριασμένα ζεύγη γραμμών των πινάκων A και B συνδυάζονται στον νέο πίνακα σε μια νέα γραμμή αποτελεσμάτων. Η INNER JOIN είναι ο προεπιλεγμένος τύπος ένωσης στην SQLite και παράλληλα ο πιο συχνά χρησιμοποιούμενος. Η χρησιμοποίηση της λέξης INNER είναι προαιρετική. Η βασική σύνταξη της INNER JOIN είναι η εξής:

- `SELECT column1,column2,.....columnN`  
`FROM table_name1 [INNER] JOIN table_name2 ON conditional_expression(συνθήκες για την ένωση)`  
`[WHERE [ conditions ]];` (προαιρετική η χρήση του WHERE)

Πολλές φορές για να κάνουμε την δήλωση πιο σύντομη και για να αποφύγουμε τον πλεονασμό οι συνθήκες για την ένωση των δυο πινάκων της INNER JOIN μπορούν να δηλωθούν χρησιμοποιώντας την έκφραση USING. Εδώ μέσα στην παρένθεση του όρου USING χρησιμοποιούμε τις στήλες που είναι κοινές και στους δυο πίνακες που κάνουμε την ένωση. Η βασική της σύνταξη είναι η εξής:

- `SELECT column1,column2,.....columnN`  
`FROM table_name1 JOIN table_name2 USING ( column1 ,...columnN )`  
`[WHERE [ conditions ]];` (προαιρετική η χρήση του WHERE);

Μια παραλλαγή της JOIN....USING είναι η NATURAL JOIN με τη διαφορά όμως ότι με την έκφραση αυτή η SQLite αυτομάτως ελέγχει αν υπάρχει ισότητα στις τιμές κάθε στήλης ανάμεσα στους δυο πίνακες, δεν χρειάζεται να ορίσουμε εμείς τις στήλες για την ένωση των δυο πινάκων. Η βασική της σύνταξη είναι η εξής:

- `SELECT column1,column2,.....columnN`  
`FROM table_name1 NATURAL JOIN table_name2`  
`[WHERE [ conditions ]];` (προαιρετική η χρήση του WHERE)

Στο επόμενο παράδειγμα για την χρήση της INNER JOIN θα χρησιμοποιήσουμε δυο πίνακες «Students» & «Departments» ενός πανεπιστημίου.

Ο πίνακας «Students»

| ID | First_Name<br>(Όνομα) | Last_Name<br>(Επώνυμο) | Age<br>(Ηλικία) | Department_Id |
|----|-----------------------|------------------------|-----------------|---------------|
| 1  | Foivos                | Dellas                 | 21              | 1             |
| 2  | Stamatis              | Sarantoglou            | 19              | 2             |
| 3  | Panagiotis            | Mpampalas              | 22              | 2             |
| 4  | Xristos               | Barbatis               | 18              | 3             |
| 5  | Andreas               | Veinoglou              | 25              | 4             |
| 6  | Emmanouhl             | Kalapodis              | 24              | 3             |

Και ο πίνακας «Departments»

| ID | Department_Name<br>(Όνομα Τμήματος) |
|----|-------------------------------------|
| 1  | Economics                           |
| 2  | Informatics                         |
| 3  | Marketing                           |
| 4  | Business Administration             |
| 5  | Public Administration               |
| 6  | Maritime                            |

Παρατηρούμε ότι στον πίνακα «Students» η στήλη Department\_Id απεικονίζει σε ποιο τμήμα του πίνακα «Departments» ανήκει κάθε φοιτητής. Για να μπορέσουμε να κάνουμε την ένωση των δυο πινάκων θα ακολουθήσουμε την εξής σύνταξη:

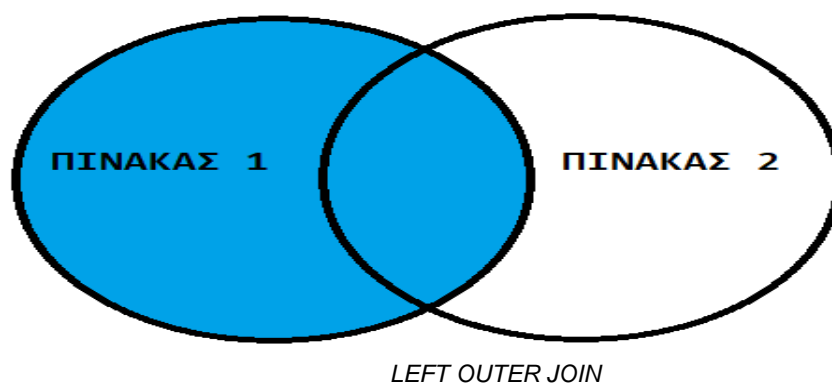
- **SELECT** First\_Name, Last\_Name, Department\_Name, Age  
**FROM** Students **JOIN** Departments **ON** Students.Department\_Id = Departments.ID;

Και έχουμε ως αποτέλεσμα τον επόμενο πίνακα:

| First_Name<br>(Όνομα) | Last_Name<br>(Επώνυμο) | Department_Name<br>(Όνομα Τμήματος) | Age<br>(Ηλικία) |
|-----------------------|------------------------|-------------------------------------|-----------------|
| Foivos                | Dellas                 | Economics                           | 21              |
| Stamatis              | Sarantoglou            | Informatics                         | 19              |
| Panagiotis            | Mpampalas              | Informatics                         | 22              |
| Xristos               | Barbatis               | Marketing                           | 18              |
| Andreas               | Veinoglou              | Business Administration             | 25              |
| Emmanouhl             | Kalapodis              | Marketing                           | 24              |

### OUTER JOIN(ΕΞΩΤΕΡΙΚΗ ΕΝΩΣΗ)

Στα πρότυπα της SQL ορίζονται τρεις τύποι εξωτερικής ένωσης πινάκων με OUTER JOIN: LEFT, RIGHT & FULL, η SQLite όμως υποστηρίζει μόνο την LEFT OUTER JOIN ή για συντομία LEFT JOIN. Οι ενώσεις πινάκων με OUTER JOINS χρησιμοποιούν μια συνθήκη όμοια με αυτή των ενώσεων INNER JOINS η οποία εκφράζεται με τις λέξεις-κλειδιά ON, USING, NATURAL. Η LEFT OUTER JOIN χρησιμοποιείται για να ενώσει όλες τις γραμμές του πίνακα που βρίσκονται από την αριστερή πλευρά της λέξης-κλειδί ON με εκείνες τις γραμμές του πίνακα από την δεξιά πλευρά της λέξης-κλειδί ON για τις οποίες ικανοποιείται η συνθήκη JOIN. Στο επόμενο σχήμα αναπαρίσται η λογική της LEFT OUTER JOIN.



Κατά την διαδικασία της ένωσης με LEFT OUTER JOIN αφού βρεθούν τα κοινά στοιχεία των πινάκων όπως σε μια ένωση INNER JOIN τότε η OUTER JOIN θα πάρει τις γραμμές από τους δυο πίνακες που είναι μη συνδεδεμένες, θα τους δώσει την τιμή NULL και θα τις προσθέσει στον τελικό πίνακα αποτελεσμάτων. Η βασική σύνταξη της LEFT OUTER JOIN είναι η εξής:

- **SELECT** column1,column2,.....columnN  
**FROM** table\_name1 **LEFT OUTER JOIN** table\_name2 **ON** conditional\_expression  
[**WHERE** [ conditions ]]; (προαιρετική η χρήση του WHERE);
- ή
- **SELECT** column1,column2,.....columnN  
**FROM** table\_name1 **LEFT OUTER JOIN** table\_name2 **USING** (column1,..... columnN)  
[**WHERE** [ conditions ]]; (προαιρετική η χρήση του WHERE);

ή

- `SELECT column1,column2,.....columnN  
FROM table_name1 NATURAL LEFT OUTER JOIN table_name2  
[WHERE [ conditions ]];` (προαιρετική η χρήση του WHERE);

Χρησιμοποιώντας τους πίνακες «Students» και «Departments» θα εφαρμόσουμε την σύνταξη της LEFT OUTER JOIN:

- `SELECT Students.ID, First_Name, Last_Name, Department_Name, Age  
FROM Departments LEFT OUTER JOIN Students ON Students.Department_Id =  
Departments.ID`

Και παίρνουμε ως αποτέλεσμα τον πίνακα:

| ID   | First_Name<br>(Όνομα) | Last_Name<br>(Επώνυμο) | Department_Name         | Age<br>(Ηλικία) |
|------|-----------------------|------------------------|-------------------------|-----------------|
| 1    | Foivos                | Dellas                 | Economics               | 21              |
| 3    | Panagiotis            | Μpampalas              | Informatics             | 22              |
| 2    | Stamatis              | Sarantoglou            | Informatics             | 19              |
| 6    | Emmanouhl             | Kalapodis              | Marketing               | 24              |
| 4    | Xristos               | Barbatis               | Marketing               | 18              |
| 5    | Andreas               | Veinoglou              | Business Administration | 25              |
| NULL | NULL                  | NULL                   | Public Administration   | NULL            |
| NULL | NULL                  | NULL                   | Maritime                | NULL            |

### CROSS JOIN()

Με την χρήση της CROSS JOIN η SQLite αντιστοιχεί κάθε γραμμή του πρώτου πίνακα με κάθε γραμμή του δεύτερου πίνακα. Αν οι δυο πίνακες έχουν για παράδειγμα x και y γραμμές αντίστοιχα τότε ο τελικός πίνακας θα έχει σύνολο γραμμών όσο το γινόμενο των  $x * y$ . Επειδή η χρήση της CROSS JOIN έχει την δυνατότητα να δημιουργεί εξαιρετικά μεγάλους πίνακες πρέπει να χρησιμοποιείται με προσοχή και μόνο όταν και όπου χρειάζεται. Η σύνταξη της CROSS JOIN είναι η εξής:

- `SELECT column1,column2,.....columnN  
FROM table_name1 CROSS JOIN table_name2`

Εφαρμόζοντας την παραπάνω σύνταξη στους πίνακες «Students» και «Departments» ως εξής:

- `SELECT First_Name, Last_Name, Departments.Name, Age  
FROM Students CROSS JOIN Departments;`

Παίρνουμε ως αποτέλεσμα τον παρακάτω πίνακα:

| First_Name<br>(Όνομα) | Last_Name<br>(Επώνυμο) | Department_Name<br>(Όνομα Τμήματος) | Age<br>(Ηλικία) |
|-----------------------|------------------------|-------------------------------------|-----------------|
| Foivos                | Dellas                 | Economics                           | 21              |
| Foivos                | Dellas                 | Informatics                         | 21              |
| Foivos                | Dellas                 | Marketing                           | 21              |
| Foivos                | Dellas                 | Business Administration             | 21              |
| Foivos                | Dellas                 | Public Administration               | 21              |
| Foivos                | Dellas                 | Maritime                            | 21              |
| Stamatis              | Sarantoglou            | Economics                           | 19              |
| Stamatis              | Sarantoglou            | Informatics                         | 19              |
| Stamatis              | Sarantoglou            | Marketing                           | 19              |
| Stamatis              | Sarantoglou            | Business Administration             | 19              |
| Stamatis              | Sarantoglou            | Public Administration               | 19              |
| Stamatis              | Sarantoglou            | Maritime                            | 19              |

|            |           |                         |    |
|------------|-----------|-------------------------|----|
| Panagiotis | Mpampalas | Economics               | 22 |
| Panagiotis | Mpampalas | Informatics             | 22 |
| Panagiotis | Mpampalas | Marketing               | 22 |
| Panagiotis | Mpampalas | Business Administration | 22 |
| Panagiotis | Mpampalas | Public Administration   | 22 |
| Panagiotis | Mpampalas | Maritime                | 22 |
| Xristos    | Barbatis  | Economics               | 18 |
| Xristos    | Barbatis  | Informatics             | 18 |
| Xristos    | Barbatis  | Marketing               | 18 |
| Xristos    | Barbatis  | Business Administration | 18 |
| Xristos    | Barbatis  | Public Administration   | 18 |
| Xristos    | Barbatis  | Maritime                | 18 |
| Andreas    | Veinoglou | Economics               | 25 |
| Andreas    | Veinoglou | Informatics             | 25 |
| Andreas    | Veinoglou | Marketing               | 25 |
| Andreas    | Veinoglou | Business Administration | 25 |
| Andreas    | Veinoglou | Public Administration   | 25 |
| Andreas    | Veinoglou | Maritime                | 25 |
| Emmanouhl  | Kalapodis | Economics               | 24 |
| Emmanouhl  | Kalapodis | Informatics             | 24 |
| Emmanouhl  | Kalapodis | Marketing               | 24 |
| Emmanouhl  | Kalapodis | Business Administration | 24 |
| Emmanouhl  | Kalapodis | Public Administration   | 24 |
| Emmanouhl  | Kalapodis | Maritime                | 24 |

Παρατηρούμε ότι ο παραπάνω πίνακας αποτελείται από 36 γραμμές, όσες δηλαδή και το γινόμενο των γραμμών των δυο πινάκων «Students» και «Departments»  $6 * 6 = 36$  από την ένωση των οποίων προέκυψε.

## 6.6 Ανασκόπηση πεδίου

- Udemy  
**SQLite for beginners**

Το συγκεκριμένο πρόγραμμα εκμάθησης στο οποίο αποκτά πρόσβαση κάποιος κατόπιν πληρωμής δίνει στον ενδιαφερόμενο τη δυνατότητα να αποκτήσει μια σφαιρική εικόνα της SQLite και μερικών από των πιο χρήσιμων και δημοφιλών χαρακτηριστικών της. Απαρτίζεται από εκπαιδευτικά βίντεο αρκετών ωρών, αρχείων διαθέσιμων προς κατέβασμα και χορήγηση πιστοποιητικού επιτυχημένης εκπλήρωσης κατά την ολοκλήρωση του προγράμματος εκμάθησης.

Πηγή: <https://www.udemy.com/course/sqlite-for-beginners/>

- DataCamp  
**Beginners Guide to SQLite**

Το πρόγραμμα κατόπιν εγγραφής αποτελεί μια δωρεάν λύση για όποιον επιθυμεί να πάρει μια γρήγορη γεύση για το τι είναι η SQLite, και τις δυνατότητες της. Είναι αρκετά σύντομο και περιεκτικό και κύριο στόχο έχει να δώσει το ερέθισμα στον χρήστη για να ψάξει βαθύτερα και να χρησιμοποιήσει την SQLite.

Πηγή: <https://www.datacamp.com/community/tutorials/beginners-guide-to-sqlite>

- SQLite tutorial  
**SQLite tutorial**

Ένα δωρεάν πρόγραμμα εκμάθησης για το τι είναι η SQLite και πως μπορεί να μάθει κανείς εύκολα και γρήγορα να την χρησιμοποιεί. Αξιοσημείωτο του προγράμματος είναι η αναλυτική και εις βάθος επεξήγηση των εννοιών της SQLite καθώς και τα πολλά παραδείγματα για την αποτελεσματικότερη κατανόηση των λειτουργιών της. Επίσης αξίζει να αναφέρουμε ότι δίνεται η δυνατότητα στον χρήστη να μελετήσει και να πειραματιστεί στην αλληλεπίδραση της SQLite με άλλες δημοφιλείς γλώσσες προγραμματισμού όπως Java, Python κ.α. Το γεγονός ότι είναι δωρεάν και δεν απαιτεί ούτε εγγραφή από τον χρήστη το καθιστά αρκετά ελκυστικό αλλά για τον ίδιο ακριβώς λόγο θα πρέπει κι ο εκάστοτε χρήστης να αντιμετωπίζει την ορθότητα των εννοιών του προγράμματος με κριτική σκέψη και διάθεση για περαιτέρω έρευνα και μελέτη.

Πηγή: <https://www.sqlitetutorial.net/>

- Quackit  
**SQLite tutorial**

Ένα σύντομο και δωρεάν πρόγραμμα εκμάθησης που απευθύνεται κυρίως σε αρχάριους χρήστες. Δεν απαιτείται εγγραφή και δίνει στον ενδιαφερόμενο μια σφαιρική εικόνα των απολύτως απαραίτητων λειτουργιών της SQLite.

Πηγή: <https://www.quackit.com/sqlite/tutorial/>

- Javatpoint  
**SQLite tutorial**

Άλλο ένα δωρεάν πρόγραμμα εκμάθησης που παρέχει πρόσβαση σε βασικές και προηγμένες έννοιες της SQLite σχεδιασμένο όπως χαρακτηριστικά αναφέρουν οι δημιουργοί του τόσο για αρχάριους όσο και για επαγγελματίες.

Το SQLite Tutorial από java-t-point περιλαμβάνει όλα τα θέματα της SQLite όπως SQLite με ιστορικό, χαρακτηριστικά, πλεονεκτήματα, εγκατάσταση, εντολές, σύνταξη, τύπους δεδομένων, χειριστές, εκφράσεις, βάσεις δεδομένων, πίνακες, λειτουργίες CRUD, ρήτρες κ.α. Επίσης αξίζει να αναφέρουμε ότι δίνεται στον χρήστη η δυνατότητα και σε αυτό το πρόγραμμα να μελετήσει και να πειραματιστεί στην αλληλεπίδραση της SQLite με άλλες δημοφιλείς γλώσσες προγραμματισμού όπως Java, Python κ.α.

Πηγή: <https://www.javatpoint.com/sqlite-tutorial>

- Tutlane  
**SQLite tutorial**

Ένα πρόγραμμα εκμάθησης που αναλύει εις βάθος όχι μόνο τις βασικές έννοιες αλλά και προχωρημένα θέματα τα οποία μπορεί να χειριστεί κανείς με την SQLite. Αξίζει και εδώ να αναφέρουμε ότι και σε αυτό το πρόγραμμα υπάρχουν παραδείγματα και μεθοδολογία για την αλληλεπίδραση της SQLite με δημοφιλείς γλώσσες προγραμματισμού. Είναι δωρεάν, δεν απαιτείται εγγραφή για να αρχίσει κάποιος να το χρησιμοποιεί και απευθύνεται όπως είναι φανερό τόσο σε αρχάριους όσο και προχωρημένους χρήστες που θέλουν να εμβαθύνουν ακόμα περισσότερο στην SQLite.

Πηγή: <https://www.tutlane.com/tutorial/sqlite>

- Eduonix  
**SQLite For Beginners Learn SQL from Scratch**

Ένα δωρεάν πρόγραμμα ηλεκτρονικής εκμάθησης που καθοδηγεί τον χρήστη μέσα από βίντεο γύρω από τις βασικές έννοιες και την εγκατάσταση, χρήση και διαχείριση βάσεων δεδομένων της SQLite. Με την ολοκλήρωση του προγράμματος αποστέλλεται στον χρήστη πιστοποιητικό ολοκλήρωσης.

Πηγή: <https://www.eduonix.com/courses/Web-Development/sqlite-for-beginners-learn-sql-from-scratch#courseContentNav>

- Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών  
**Διαχειριστής Βάσεων Δεδομένων και Κέντρων Δεδομένων**

Το συγκεκριμένο εκπαιδευτικό πρόγραμμα το οποίο υλοποιείται κατόπιν πληρωμής έχει δύο βασικούς στόχους. Αρχικά να διδάξει τις βασικές αρχές από τις οποίες διέπονται οι βάσεις δεδομένων καθώς και τις βασικές γνώσεις και εργαλεία για τον σχεδιασμό, τη χρήση και την υλοποίηση κέντρων δεδομένων και βάσεων δεδομένων. Ο δευτερεύων στόχος είναι να φέρει τους σπουδαστές στο επιθυμητό επίπεδο γνώσεων ώστε να μπορούν να χειρίζονται αποτελεσματικά γλώσσες προγραμματισμού όπως η PHP σε συνδυασμό με βάσεις δεδομένων καθώς και θέματα και εφαρμογές που άπτονται των σύγχρονων κέντρων δεδομένων.

Πηγή: <https://elearningekpa.gr/courses/diaxeirisths-basewn-dedomenwn-kai-kentrwn-dedomenwn>

## 7. Παρουσίαση και χρήση εφαρμογής

Στην ενότητα αυτή παρουσιάζεται το εγχειρίδιο χρήστη για το πρόγραμμα ηλεκτρονικής εκμάθησης που υλοποιήσαμε.

### 7.1 Εγχειρίδιο χρήσης εφαρμογής

Η ιστοσελίδα που δημιουργήσαμε για το πρόγραμμα ηλεκτρονικής εκμάθησης ξεκινά με την **ΑΡΧΙΚΗ** σελίδα η οποία κάνει μια πρώτη εισαγωγή στον χρήστη για την SQLite και για το πρόγραμμα ηλεκτρονικής εκμάθησης που δημιουργήσαμε.

The screenshot shows a web interface for 'Ηλεκτρονική Εκμάθηση SQLite'. The navigation bar includes 'ΑΡΧΙΚΗ', 'ΕΓΓΡΑΦΗ', 'ΣΤΑΤΙΣΤΙΚΑ', 'ΑΡΧΕΙΑ & ΒΙΝΤΕΟ', 'ΕΠΙΚΟΙΝΩΝΙΑ', and 'Σύνδεση'. The main content area has a sidebar with database management options like 'CREATE & DELETE DB', 'CREATE TABLE', 'DROP TABLE', 'INSERT', 'SELECT', 'WHERE', and 'UPDATE'. The main text explains SQLite and includes a diagram of the architecture.

**Γιατί SQLite;**

Η SQLite είναι μια βιβλιοθήκη λογισμικού η οποία παρέχει ένα σύστημα για την διαχείριση σχεσιακών βάσεων δεδομένων. Η χρησιμοποίηση της λέξης ελαφριά (lite) στο όνομα της σημαίνει το ελάχιστο μέγεθος που απαιτεί σε όρους εγκατάστασης, διαχείρισης της βάσης δεδομένων και των απαιτούμενων πόρων του συστήματος. Η SQLite διαθέτει τα ακόλουθα αξιοπρόσεχτα χαρακτηριστικά: είναι αυτοτελής, δεν χρειάζεται διακομιστές(serverless), χρειάζεται μηδενική διαμόρφωση, είναι συναλλακτική (transactional) με πάνω απ' όλα είναι ανοιχτού κώδικα άρα δωρεάν για όλους τους χρήστες. Αναφερόμαστε ιδιαίτερα στο γεγονός ότι η SQLite δεν χρειάζεται διακομιστή διότι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων όπως το ξέραμε μέχρι σήμερα όπως π.χ. MySQL, PostgreSQL απαιτεί μια ξεχωριστή λειτουργία διακομιστή για να λειτουργήσει. Η εφαρμογή ή οι εφαρμογές που θέλουν πρόσβαση στον διακομιστή της βάσης δεδομένων χρησιμοποιούν το πρωτόκολλο TCP/IP για να στείλουν και να λάβουν αιτήματα. Η συγκεκριμένη αρχιτεκτονική ονομάζεται πελάτη - διακομιστή και απεικονίζεται χαρακτηριστικά στο παρακάτω διάγραμμα:

The diagram illustrates the architecture between a CLIENT and a SERVER. On the CLIENT side, there is an 'ΕΦΑΡΜΟΓΗ - APPLICATION' (Application) which connects to a 'ΣΥΜΒΕΤΙΚΟΣ ΚΡΟΝΟΣ - CONNECTOR' (Connector). On the SERVER side, there is a 'ΔΙΕΡΓΑΣΙΑ ΕΞΑΓΩΓΗΣ - SERVER PROCESS' (Server Process) which interacts with 'DATABASE FILES ΑΡΧΕΙΑ ΒΑΣΗΣ' (Database Files). Arrows indicate the flow of data and control between these components.

#### 1. Αρχική Σελίδα

Μέσα στην αρχική σελίδα υπάρχουν χρήσιμοι υπερσύνδεσμοι οι οποίοι ανακατευθύνουν τον χρήστη σε σελίδες που περιέχουν οδηγίες για να κατεβάσει και να εγκαταστήσει την SQLite στον υπολογιστή του, να μάθει τις βασικές εντολές και σύνταξη μέσω της γραμμής εντολών των Windows καθώς και για τους τύπους δεδομένων και τελεστών που χρησιμοποιεί η SQLite και θα του φανούν πολύ χρήσιμοι στην κατανόηση της θεωρίας των ενοτήτων καθώς και στην επίλυση των ασκήσεων που περιλαμβάνονται σε κάθε ενότητα.

The screenshot displays a web application interface with a dark sidebar on the left containing navigation options: CREATE & DELETE DB, CREATE TABLE, DROP TABLE, INSERT, SELECT, WHERE, and UPDATE. The main content area is titled 'CLIENT ΠΕΛΑΤΗΣ' and 'SERVER ΔΙΑΚΟΜΙΣΤΗΣ'. It explains that SQLite is a client-server database system where the server is a file-based database. A diagram shows an 'ΕΦΑΡΜΟΓΗ - APPLICATION' connected to 'SYNΔΕΣΗ ΚΑΤΕΒΑΣΜΑ & ΕΓΚΑΤΑΣΤΑΣΗ' (Connector) and 'DATABASE FILES ΑΡΧΕΙΑ ΒΑΣΗΣ' (Database Files). Below the diagram, the text reads: 'Ξεκινώντας με την SQLite; Εγκατάσταση της SQLite'. It provides instructions on downloading and installing SQLite, and lists 'Εντολές & Σύνταξη' (Commands & Syntax) and 'Τύποι Δεδομένων & Τελεστές' (Data Types & Operators).

## 2. Χρήσιμοι Υπερσύνδεσμοι στην Αρχική σελίδα

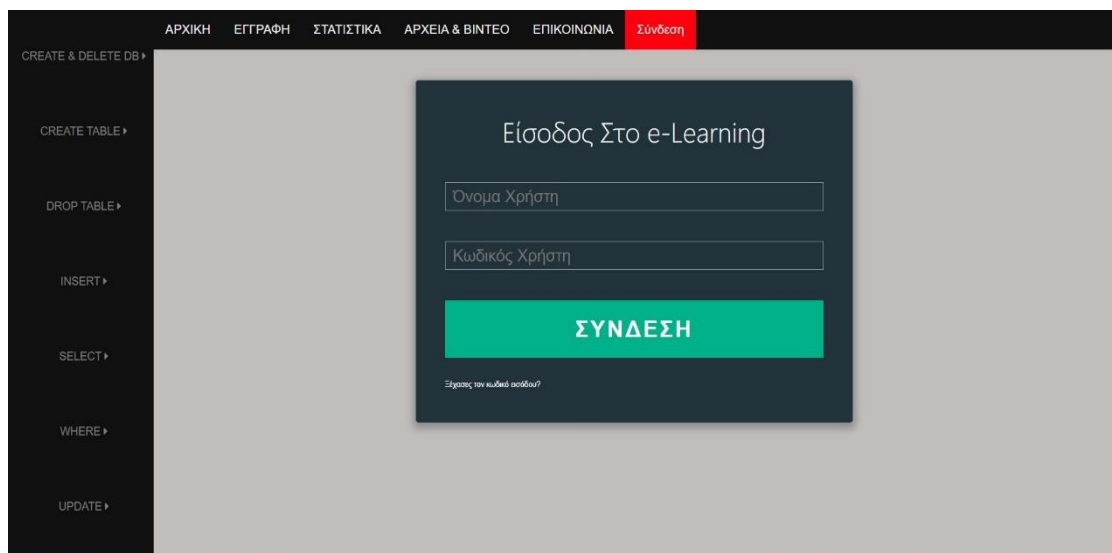
Ο χρήστης μπορεί να πλοηγηθεί στην ιστοσελίδα είτε από την πλάγια μπάρα πλοήγησης η οποία περιέχει όλα τα κεφάλαια της SQLite (Θεωρία & Ασκήσεις) είτε από την πάνω μπάρα πλοήγησης η οποία περιέχει βασικές λειτουργίες της ιστοσελίδας όπως **ΣΥΝΔΕΣΗ – ΑΠΟΣΥΝΔΕΣΗ** χρήστη, **ΕΓΓΡΑΦΗ** νέου χρήστη κ.α. Απαραίτητη προϋπόθεση για να μπορεί ο χρήστης να πλοηγηθεί ελεύθερα στις υπόλοιπες σελίδες της εφαρμογής μας πέρα από την **ΑΡΧΙΚΗ** σελίδα είναι να έχει πραγματοποιήσει **ΕΓΓΡΑΦΗ** νέου χρήστη. Πατώντας το κουμπί ΕΓΓΡΑΦΗ στην επάνω μπάρα πλοήγησης, ο χρήστης μεταφέρεται στην σελίδα που φαίνεται και στην επομένη εικόνα για την εγγραφή του.

The screenshot shows a web browser window displaying a user registration form titled 'Εγγραφή Νέου Χρήστη'. The form includes input fields for 'Όνομα', 'Επώνυμο', 'Διεύθυνση', 'Τηλέφωνο', 'Email', 'Όνομα Χρήστη', and 'Κωδικός Χρήστη'. A prominent green button labeled 'ΚΑΤΑΧΩΡΗΣΗ' is at the bottom of the form. The browser's address bar shows 'localhost:44369/WebForm1.aspx'.

## 3. Εγγραφή νέου χρήστη

Με την συμπλήρωση όλων των πεδίων και την επιτυχή ολοκλήρωση της εγγραφής του ανακατευθύνεται στην σελίδα για την **ΣΥΝΔΕΣΗ** του στην εφαρμογή.



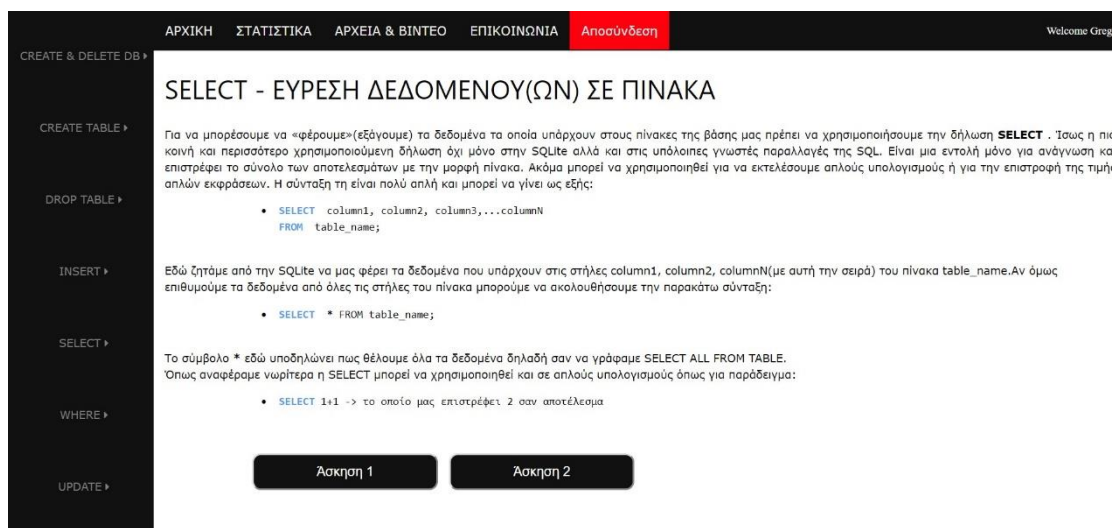


#### 4. Σύνδεση Χρήστη

Πληκτρολογώντας το όνομα χρήστη και τον κωδικό χρήστη γίνεται η ταυτοποίηση του και εισέρχεται στην εφαρμογή είτε ως **ΜΑΘΗΤΗΣ** με πρόσβαση μόνο στο εκπαιδευτικό κομμάτι της εφαρμογής (Θεωρία & Ασκήσεις SQLite ) είτε ως **ΚΑΘΗΓΗΤΗΣ** που αποκτά πρόσβαση και σε άλλες λειτουργίες της εφαρμογής όπως στατιστικά χρηστών, ανέβασμα αρχείων, αλλαγή των υφιστάμενων ασκήσεων των ενότητων κ.α. Τα στοιχεία του χρήστη κατά την επιτυχή του σύνδεση αποθηκεύονται σε ένα cookie το οποίο καταστρέφεται κατά την αποσύνδεση του χρήστη από την ιστοσελίδα για λόγους ασφάλειας.

#### Είσοδος ως Μαθητής

Μετά την επιτυχή σύνδεση του ο χρήστης - Μαθητής μπορεί να ξεκινήσει να πλοηγείται στις ενότητες της εφαρμογής και να διαβάζει την θεωρία κάθε ενότητας. Οι ενότητες στην πλάγια μπάρα πλοήγησης δεν έχουν τοποθετηθεί με τυχαία σειρά αλλά οδηγούν τον χρήστη να ξεκινήσει από τις πιο βασικές και εισαγωγικές έννοιες της SQLite στις πιο απαιτητικές. Παράλληλα η γνώση και η αφομοίωση των εννοιών στις αρχικές ενότητες είναι αναγκαία για να μπορέσει ο χρήστης να ανταποκριθεί στις επόμενες.



#### 5. Ενότητα SELECT

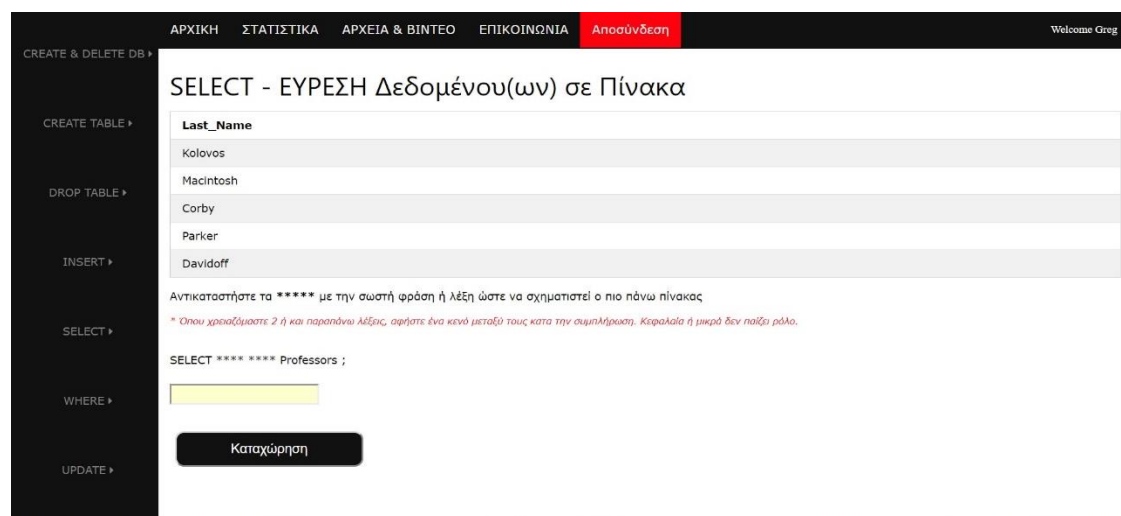
Επιλέγοντας κάποια από τις ενότητες μεταφέρεται στην αντίστοιχη σελίδα η οποία περιέχει την θεωρία για την συγκεκριμένη ενότητα της SQLite, κάποιο ενδεικτικό παράδειγμα κατανόησης της θεωρίας καθώς και τα κουμπιά που οδηγούν στις ασκήσεις εξάσκησης της συγκεκριμένης

ενότητας. Η Άσκηση 1 κάθε ενότητας είναι μια ερώτηση πολλαπλής επιλογής ενώ η Άσκηση 2 αντιστοιχεί σε ερώτηση συμπλήρωσης κενού.



6. Άσκηση 1 – Πολλαπλής Επιλογής

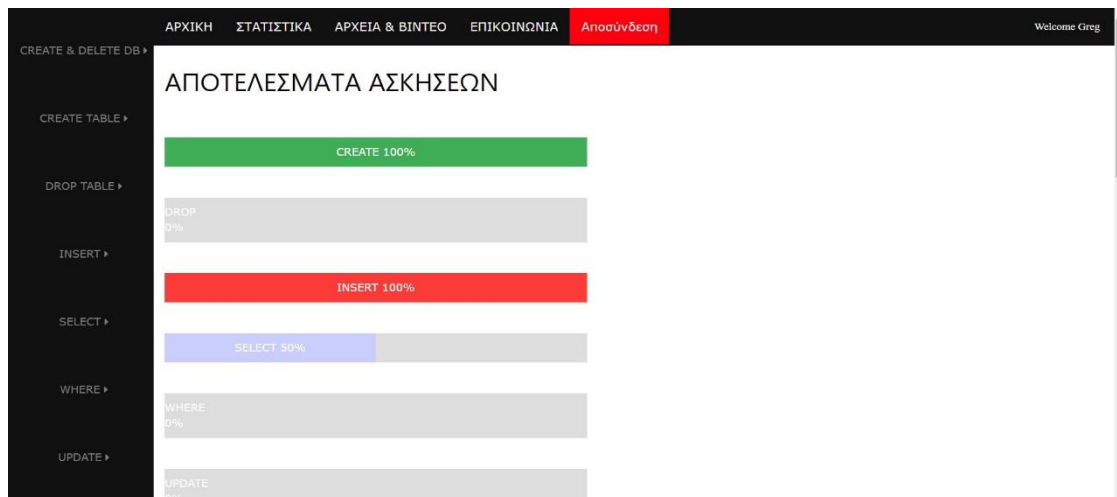
Ο χρήστης μπορεί να επιλέξει μια εκ των τεσσάρων πιθανών επιλογών και πατώντας το κουμπί καταχώρηση η βάση δεδομένων ενημερώνεται με την απάντησή του.



7. Άσκηση 2 – Συμπλήρωση Κενού

Εδώ ο χρήστης μπορεί να συμπληρώσει στο κατάλληλο πεδίο την απάντηση που θεωρεί πως είναι σωστή και πατώντας το κουμπί καταχώρηση να αποθηκεύσει την απάντησή του στην βάση δεδομένων της ιστοσελίδας. Επίσης ενημερώνεται πριν τη συμπλήρωση του κατάλληλου λεκτικού για τον τρόπο με τον οποίο θα πρέπει να το κάνει ώστε η απάντησή του να είναι έγκυρη.

Για κάθε ενότητα που ολοκληρώνει ο χρήστης οι σωστές και λάθος απαντήσεις του αποθηκεύονται στην βάση δεδομένων της εφαρμογής και ο χρήστης μπορεί να παρακολουθήσει την πρόοδο του πατώντας το κουμπί **ΣΤΑΤΙΣΤΙΚΑ** που βρίσκεται στην πάνω μπάρα πλοήγησης της ιστοσελίδας.



### 8. Στατιστικά χρήση

Στην πάνω μπάρα πλοήγησης πατώντας το κουμπί **ΕΠΙΚΟΙΝΩΝΙΑ** ο χρήστης μπορεί να έχει πρόσβαση στην σελίδα επικοινωνίας με τον διαχειριστή (ΚΑΘΗΓΗΤΗ) της ιστοσελίδας για να ζητήσει βοήθεια σε τυχόν απορίες, να αναφέρει κάποιο πρόβλημα ή σφάλμα της εφαρμογής και να ζητήσει από τον καθηγητή να ανεβάσει σημειώσεις ή χρήσιμα έγγραφα για διάβασμα στην κατάλληλη ενότητα της ιστοσελίδας.

### 9. Επικοινωνία

Η ενότητα της ιστοσελίδας στην οποία είναι διαθέσιμα τα αρχεία και οι σημειώσεις του καθηγητή για κατέβασμα και αποθήκευση από τους χρήστες γίνεται διαθέσιμη πατώντας το κουμπί **ΑΡΧΕΙΑ & ΒΙΝΤΕΟ** που βρίσκεται στην πάνω μπάρα πλοήγησης της ιστοσελίδας. Στην συγκεκριμένη σελίδα πέρα από τα χρήσιμα έγγραφα προς κατέβασμα μπορεί να παρακολουθήσει και εκπαιδευτικά βίντεο με χρήσιμες συμβουλές για την χρήση της SQLite.

| Όνομα Αρχείου    |           |
|------------------|-----------|
| ggggg.txt        | Κατέβασμα |
| SpotifySetup.exe | Κατέβασμα |
| usbTYPEC hub.jpg | Κατέβασμα |

Χρήσιμα βίντεο με βασικές λειτουργίες της SQLite.

10. Αρχεία &amp; Βίντεο

Οποιαδήποτε στιγμή επιθυμεί ο χρήστης πατώντας το κουμπί ΑΠΟΣΥΝΔΕΣΗ στην επάνω μπάρα πλοήγησης μπορεί να εξέλθει από το πρόγραμμα εκμάθησης.

### Είσοδος ως Καθηγητής

Με την επιτυχή σύνδεση του χρήστη ως Καθηγητή εισέρχεται αυτόματα στην σελίδα διαχείρισης της ιστοσελίδας εν αντιθέσει με τους υπόλοιπους χρήστες που εισέρχονται πάντα στην **ΑΡΧΙΚΗ** σελίδα και δεν έχουν δικαιώματα πρόσβασης στην σελίδα διαχείρισης.

11. Σελίδα Διαχειριστή

Από εδώ ο χρήστης μπορεί να επιλέξει είτε να κάνει αλλαγές στις ασκήσεις της κάθε ενότητας, είτε να ανεβάσει αρχεία και χρήσιμα έγγραφα για τους μαθητές είτε τέλος να δει τα στατιστικά όλων των χρηστών – μαθητών της ιστοσελίδας. Πατώντας το κουμπί **ΑΛΛΑΓΗ ΑΣΚΗΣΕΩΝ** μεταφέρεται στην σελίδα για την αλλαγή και την διόρθωση των ασκήσεων εξάσκησης των εννοιών του προγράμματος εκμάθησης. Εδώ συμπληρώνοντας το όνομα της ενότητας της SQLite, τον αριθμό της Άσκησης και πατώντας **ΕΥΡΕΣΗ** φορτώνεται από τη βάση δεδομένων η άσκηση για επεξεργασία από τον χρήστη.

The screenshot shows a web application interface with a dark sidebar on the left containing navigation options: CREATE & DELETE DB, CREATE TABLE, DROP TABLE, INSERT, SELECT, WHERE, and UPDATE. The main content area has a top navigation bar with links: ΑΡΧΙΚΗ, ΣΤΑΤΙΣΤΙΚΑ, ΔΙΑΧΕΙΡΙΣΤΗΣ, ΑΡΧΕΙΑ & ΒΙΝΤΕΟ, ΕΠΙΚΟΙΝΩΝΙΑ, and Αποσύνδεση (highlighted in red). A 'Welcome ALB' message is in the top right. The central panel is titled 'ΕΥΡΕΣΗ ΕΡΩΤΗΣΗΣ' and contains two input fields: 'ΕΝΟΤΗΤΑ π.χ 'CREATE'' and 'ΑΡΙΘΜΟΣ ΑΣΚΗΣΗΣ π.χ '1,2''. Below these is a large green button labeled 'ΕΥΡΕΣΗ'.

12. Αλλαγή Ασκήσεων

The screenshot shows the 'Επεξεργασία Ασκήσης' (Edit Exercise) form. The sidebar and top navigation are the same as in the previous screenshot. The main panel is titled 'ΕΥΡΕΣΗ ΕΡΩΤΗΣΗΣ' and contains a text area with the instruction: 'ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΑΡΑΠΑΝΩ ΠΙΝΑΚΑ.' Below this is a 'CREATE' button, a text input field containing the number '1', and a code editor area. The code editor shows the SQL command: `CREATE TABLE Professors ( ID INTEGER PRIMARY KEY, First_Name TEXT NOT NULL, Last_Name TEXT NOT NULL, Department TEXT NOT NULL, Salary REAL NOT NULL);`. A green box highlights the same command with a slightly different formatting: `CREATE TABLE Professors ( ID, First_Name, Last_Name, Department, Salary );`.

13. Επεξεργασία Ασκήσης

Στην νέα φόρμα που φορτώνεται τα πεδία έχουν συμπληρωθεί με τα δεδομένα από τη βάση και είναι διαθέσιμα στον Καθηγητή για επεξεργασία. Κάθε πεδίο μπορεί ο χρήστης να το μεγαλώσει ή να το μικρύνει σε μέγεθος, προς διευκόλυνση ανάγνωσης και επεξεργασίας του κειμένου που εμπεριέχεται σε αυτό. Τέλος μπορεί να κάνει όποιες αλλαγές θέλει και πατώντας το κουμπί **ΕΝΗΜΕΡΩΣΗ** να αποθηκευτούν οι αλλαγές στην βάση δεδομένων. Επιστρέφοντας πάλι πίσω στην κεντρική σελίδα διαχείρισης και επιλέγοντας το κουμπί **ΑΝΕΒΑΣΜΑ ΑΡΧΕΙΩΝ** ο Καθηγητής μεταφέρεται στην ενότητα για την διαχείριση και το ανέβασμα των αρχείων – σημειώσεων του προγράμματος εκμάθησης.

ΑΡΧΙΚΗ ΣΤΑΤΙΣΤΙΚΑ ΔΙΑΧΕΙΡΙΣΤΗΣ ΑΡΧΕΙΑ & ΒΙΝΤΕΟ ΕΠΙΚΟΙΝΩΝΙΑ Αποσύνδεση Welcome ALB

CREATE & DELETE DB >

CREATE TABLE >

DROP TABLE >

INSERT >

SELECT >

WHERE >

UPDATE >

## ΑΝΕΒΑΣΜΑ ΑΡΧΕΙΩΝ

Επιλογή αρχείου | Δεν επιλέχθηκε κανένα αρχείο.

| Όνομα Αρχείου    | Κατέβασμα | Διαγραφή |
|------------------|-----------|----------|
| 99999.txt        | Κατέβασμα | Διαγραφή |
| SpotifySetup.exe | Κατέβασμα | Διαγραφή |
| usbTYPEC hub.jpg | Κατέβασμα | Διαγραφή |

Upload

#### 14. Ανέβασμα αρχείων

Εδώ φορτώνεται δυναμικά ένας πίνακας ο οποίος περιέχει όλα τα αρχεία που έχουν ήδη ανέβει μέχρι τώρα στην ιστοσελίδα. Ο χρήστης - Καθηγητής έχει τη δυνατότητα να διαγράψει όποια δεν επιθυμεί πλέον πατώντας το κουμπί **Διαγραφή** καθώς και να ανεβάσει νέα πατώντας το κουμπί **Επιλογή αρχείου**. Θα του ανοίξει ένα νέο παράθυρο διαλόγου για να επιλέξει το αρχείο που επιθυμεί προς ανέβασμα και αφού το επιλέξει μπορεί να πατήσει το κουμπί **Upload**. Αν το αρχείο είναι μεγαλύτερο από 100 MB η διαδικασία θα αποτύχει και θα ενημερωθεί ο χρήστης με το ανάλογο μήνυμα σφάλματος διαφορετικά το αρχείο θα ανέβει στον server και θα ενημερωθεί άμεσα ο πίνακας με τα περιεχόμενα αρχεία. Επιστρέφοντας πάλι πίσω στην κεντρική σελίδα διαχείρισης και επιλέγοντας την τρίτη επιλογή **ΣΤΑΤΙΣΤΙΚΑ ΧΡΗΣΤΩΝ** ο χρήστης μεταφέρεται στην σελίδα που περιέχει όλα τα στατιστικά των χρηστών της ιστοσελίδας.

ΑΡΧΙΚΗ ΣΤΑΤΙΣΤΙΚΑ ΔΙΑΧΕΙΡΙΣΤΗΣ ΑΡΧΕΙΑ & ΒΙΝΤΕΟ ΕΠΙΚΟΙΝΩΝΙΑ Αποσύνδεση Welcome alb

CREATE & DELETE DB >

CREATE TABLE >

DROP TABLE >

INSERT >

SELECT >

WHERE >

UPDATE >

## ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΤΗ

ΑΛΛΑΓΗ ΑΣΚΗΣΕΩΝ

ΑΝΕΒΑΣΜΑ ΑΡΧΕΙΩΝ

ΣΤΑΤΙΣΤΙΚΑ ΧΡΗΣΤΩΝ

#### 15. Στατιστικά χρηστών

Στην σελίδα αυτή φορτώνεται δυναμικά ένας πίνακας με όλους τους χρήστες της ιστοσελίδας και σε κάθε έναν από αυτούς πατώντας το κουμπί **Επιλέξτε** δημιουργείται ένας νέος πίνακας με αναλυτικά στατιστικά των ασκήσεων κάθε ενότητας που έχει ολοκληρώσει ο συγκεκριμένος χρήστης.

**ΣΤΑΤΙΣΤΙΚΑ ΧΡΗΣΤΩΝ**

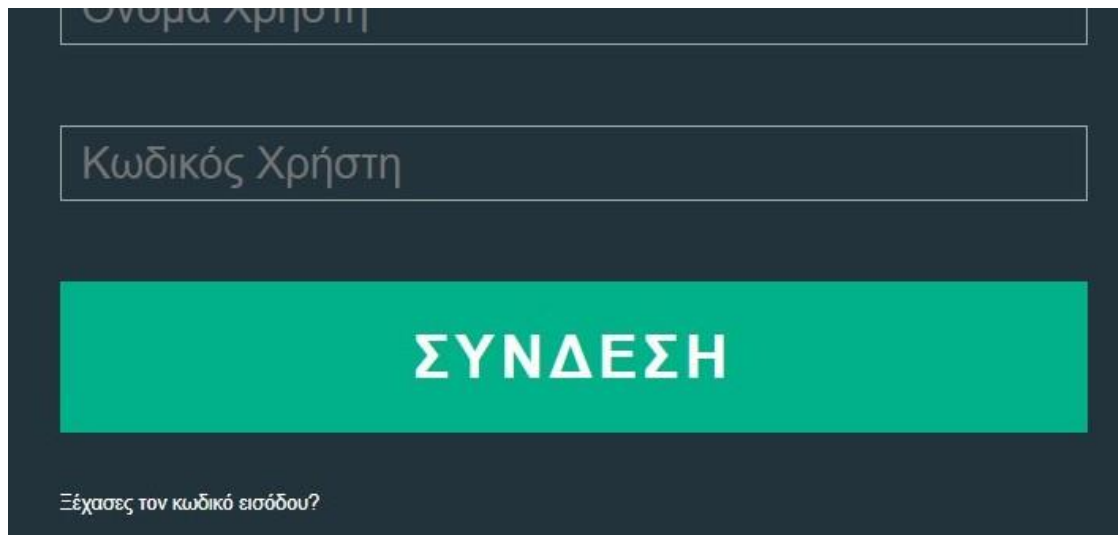
| UserId | A.A | ΟΝΟΜΑ     | ΕΠΩΝΥΜΟ  | ΣΩΣΤΕΣ ΑΠΑΝΤΗΣΕΙΣ | ΛΑΘΟΣ ΑΠΑΝΤΗΣΕΙΣ | ΣΥΝΟΛΟ ΑΠΑΝΤΗΜΕΝΩΝ | ΣΥΝΟΛΟ ΕΡΩΤΗΣΕΩΝ |          |
|--------|-----|-----------|----------|-------------------|------------------|--------------------|------------------|----------|
| 1      | 1   | GG        | BB       | 11                | 4                | 15                 | 31               | Επιλέξτε |
| 2      | 2   | Albert    | Einstein | 29                | 2                | 31                 | 31               | Επιλέξτε |
| 5      | 3   | GRIGORIOS | BERKETIS | 4                 | 0                | 4                  | 31               | Επιλέξτε |

Αναλυτική βαθμολογία: Albert Einstein

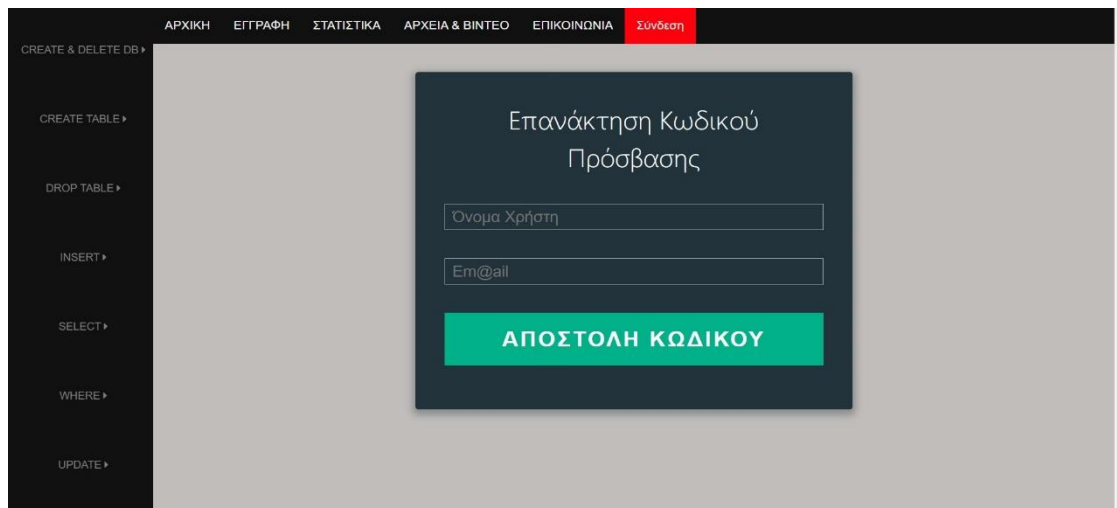
| A.A | ΚΕΦΑΛΑΙΟ      | ΣΩΣΤΕΣ ΑΠΑΝΤΗΣΕΙΣ | ΛΑΘΟΣ ΑΠΑΝΤΗΣΕΙΣ | ΣΥΝΟΛΟ ΑΠΑΝΤΗΜΕΝΩΝ |
|-----|---------------|-------------------|------------------|--------------------|
| 1   | ANDOR         | 2                 | 0                | 2                  |
| 2   | AVERAGE       | 2                 | 0                | 2                  |
| 3   | CREATE        | 2                 | 0                | 2                  |
| 4   | DELETE        | 1                 | 0                | 1                  |
| 5   | DISTINCTCOUNT | 2                 | 0                | 2                  |
| 6   | DROP          | 1                 | 0                | 1                  |
| 7   | GROUPBY       | 2                 | 0                | 2                  |
| 8   | HAVING        | 2                 | 0                | 2                  |
| 9   | INSERT        | 2                 | 0                | 2                  |
| 10  | JOIN          | 2                 | 0                | 2                  |

16. Στατιστικά χρηστών

Αν ο χρήστης που θέλει να συνδεθεί στο πρόγραμμα εκμάθησης έχει ξεχάσει τον κωδικό πρόσβασης μπορεί να πατήσει τον σύνδεσμο **Ξέχασες τον κωδικό εισόδου?** Που βρίσκεται στην σελίδα **ΣΥΝΔΕΣΗ** και να μεταφερθεί στην σελίδα ανάκτησης του κωδικού πρόσβασης.



17. Ξέχασες τον κωδικό εισόδου?



18. Επανάκτηση κωδικού πρόσβασης

Συμπληρώνοντας το όνομα χρήστη και το email που είχε καταχωρήσει ο χρήστης κατά την αρχική του εγγραφή και πατώντας το κουμπί **ΑΠΟΣΤΟΛΗ ΚΩΔΙΚΟΥ** του αποστέλλεται ένα email με τον κωδικό πρόσβασης που είναι αποθηκευμένος στη βάση δεδομένων της ιστοσελίδας. Πλέον μπορεί με το όνομα χρήστη και τον κωδικό πρόσβασης που ανέκτησε να εισέλθει στην εφαρμογή χωρίς να χρειάζεται να δημιουργήσει νέο λογαριασμό. Τέλος αν ο χρήστης θέλει να σταματήσει την εκμάθηση και ην χρήση της ιστοσελίδας μπορεί να πατήσει το κουμπί **ΑΠΟΣΥΝΔΕΣΗ** που βρίσκεται στην πάνω μπάρα πλοήγησης. Στην προσπάθεια μας να κάνουμε πιο φιλικό το πρόγραμμα στον χρήστη ακόμα κι αν κάτι έχει πάει στραβά δημιουργήσαμε τις ακόλουθες σελίδες σφαλμάτων:



19. Σφάλμα 404





20. Σφάλμα 500

## 8. Αρχιτεκτονική Συστήματος

Στην ενότητα αυτή παρουσιάζονται οι γλώσσες προγραμματισμού, οι εφαρμογές και γενικότερα τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του προγράμματος.

### 8.1 Εργαλεία που χρησιμοποιήθηκαν

Για την υλοποίηση του προγράμματος εκμάθησης και της εργασίας γενικότερα, χρησιμοποιήθηκαν:

- *Γλώσσες Προγραμματισμού & Frameworks* : HTML, CSS, Javascript, C-Sharp (C#), .NET Framework, Microsoft SQL Server 2017, Entity Framework.
- *Εφαρμογή Κατασκευής Ιστοσελίδας* : Microsoft Visual Studio 2019,
- *Εφαρμογή Διαχείρισης Βάση Δεδομένων*: Microsoft SQL Server Management Studio 18
- *Εφαρμογή Δημιουργίας & Επεξεργασίας Εγγράφων* : Microsoft Word for Office 365

Αναλυτικότερα, για την δημιουργία της εφαρμογής μας χρησιμοποιήθηκε το Visual Studio 2019 το οποίο αποτελεί το βασικό εργαλείο δημιουργίας διαδικτυακών εφαρμογών σε asp.Net framework. Το asp.Net αποτελεί ένα framework για διαδικτυακές εφαρμογές το οποίο υλοποιήθηκε και υποστηρίζεται πλήρως από την Microsoft και επιτρέπει σε μηχανικούς λογισμικού να δημιουργούν δυναμικές ιστοσελίδες. Μέσω της ευελιξίας που μας χαρίζει το συγκεκριμένο framework χρησιμοποιήθηκαν για το front end κομμάτι της ιστοσελίδας η γλώσσα μορφοποίησης HTML (HyperText Markup Language) που χρησιμοποιείται για την ανάπτυξη ιστοσελίδων καθώς και τα απαραίτητα CSS (Cascading Style Sheets ) τα οποία καθορίζουν την εμφάνιση των στοιχείων HTML όλης της ιστοσελίδας. Επιπλέον μέσω της γλώσσας Javascript κάναμε το περιεχόμενο της ιστοσελίδας μας πιο διαδραστικό και έτοιμο να ανταποκριθεί στις ενέργειες και στις ιδιαιτερότητες του εκάστοτε χρήστη. Για το back end της εφαρμογής χρησιμοποιήθηκε η αντικειμενοστραφής γλώσσα προγραμματισμού C# η οποία υποστηρίζεται πλήρως από την Microsoft και από το asp.net που χρησιμοποιήσαμε για την υλοποίηση της ιστοσελίδας μας. Ως βάση δεδομένων χρησιμοποιήσαμε τον Microsoft SQL Server 2017 και για την διαχείριση της βάσης μας χρησιμοποιήσαμε τις δυνατότητες που

μας προσφέρει το Entity framework. Το συγκεκριμένο framework το οποίο δημιουργήθηκε για .Net εφαρμογές μας προσφέρει υψηλότερο επίπεδο αφαίρεσης και μας επέτρεψε να δημιουργήσουμε και να διαχειριστούμε όλα τα δεδομένα της βάσης μας με μεγαλύτερη ευελιξία και με λιγότερες γραμμές κώδικα.

## 9. Συμπεράσματα και μελλοντικές επεκτάσεις

Το πρόγραμμα ηλεκτρονικής εκμάθησης που υλοποιήθηκε φροντίσαμε να τηρεί κάποιους απλούς αλλά βασικούς κανόνες. Θέλαμε να προσφέρει εύκολη πρόσβαση στην γνώση όλων των χρηστών, να καθιστά έννοιες δύσκολες και δυσνόητες εύκολα αφομοιώσιμες από τους χρήστες μα πάνω από όλα να διεγείρει το ενδιαφέρον του χρήστη για την περαιτέρω ενασχόληση και εις βάθος κατανόηση της SQLite.

Στην ιστοσελίδα που δημιουργήσαμε ο υποψήφιος χρήστης μπορεί εύκολα να δημιουργήσει τον προσωπικό του λογαριασμό κάνοντας εγγραφή και στη συνέχεια να ξεκινήσει την εκμάθηση του προγράμματος. Κάθε ενότητα του προγράμματος είναι έτσι τοποθετημένη ώστε η αφομοίωση των εννοιών της να λειτουργεί σαν σκαλοπάτι για την μετάβαση στην επόμενη. Κάθε σωστή ή λανθασμένη άσκηση που απαντάει ο χρήστης καταγράφεται στη βάση δεδομένων του προγράμματος έτσι ώστε ανά πάσα στιγμή να ξέρει τα λάθη του και να μπορεί εύκολα να ανατρέξει για επανάληψη στην κατάλληλη ενότητα.

Παράλληλα τα συγκεντρωτικά δεδομένα της βάσης του προγράμματος δίνουν μια συνολική εικόνα στον διαχειριστή του προγράμματος εκμάθησης για τους «καλούς» και «κακούς» μαθητές, το επίπεδο δυσκολίας των ασκήσεων και το κατά πόσο αφομοιώσιμες είναι οι έννοιες που αναλύονται.

### Μελλοντικές επεκτάσεις του προγράμματος

Μελλοντικές επεκτάσεις οι οποίες με την προσθήκη τους στο Πρόγραμμά μας μπορούν να το κάνουν ιδιαίτερο και ακόμα πιο ελκυστικό είναι:

- Αλλαγή των ενότητων του προγράμματος εκμάθησης έτσι ώστε να μην περιέχουν μόνο κείμενο αλλά ολοένα και περισσότερα χρήσιμα εκπαιδευτικά βίντεο που θα καθοδηγούν και θα κάνουν πιο φιλικό στον χρήστη το περιεχόμενο της ιστοσελίδας.
- Διαχωρισμός των ρόλων του διαχειριστή της ιστοσελίδας και του καθηγητή ώστε ο μεν διαχειριστής να είναι επιφορτισμένος μόνο με τα τεχνικά θέματα της ιστοσελίδας ενώ ο δε καθηγητής να επικεντρώνεται στον εκπαιδευτικό του ρόλο και στην επιμορφωτική οργάνωση της ιστοσελίδας ώστε να γίνεται πιο κατανοητή και πιο ελκυστική στον μέσο χρήστη.
- Δημιουργία ασκήσεων και τέστ τα οποία θα περιέχουν και χρονικό περιορισμό για την ολοκλήρωσή τους έτσι ώστε ο χρήστης να ελέγχει τις γνώσεις του και το επίπεδο του κάτω από την πίεση του χρόνου και του άγχους που ενδεχομένως αυτό να δημιουργεί στον χρήστη.
- Δημιουργία δυνατότητας άμεσης ζωντανής επικοινωνίας (live-chat) των χρηστών μεταξύ τους αλλά και με τον καθηγητή για τυχόν απορίες, προβλήματα ή ό,τι άλλο χρειαστούν σε πραγματικό χρόνο. Η μόνη δυνατότητα επικοινωνίας που δίνεται μέχρι στιγμής στους χρήστες είναι μέσω email με τον διαχειριστή της ιστοσελίδας.
- Δημιουργία περισσότερων εκπαιδευτικών επιπέδων μέσα στο πρόγραμμα τα οποία θα ξεκλειδώνονται ανάλογα με την πρόοδο και το επίπεδο του μαθητή και θα τον οδηγούν από τις πολύ απλές και εισαγωγικές έννοιες σε ολοένα και πολυπλοκότερες παρέχοντας του πληρέστερη και εις βάθος γνώση του αντικειμένου. Τέλος με την

επιτυχή ολοκλήρωση του προγράμματος θα δίνεται και πιστοποιητικό ολοκλήρωσης στον χρήστη.

- Ύπαρξη εξειδικευμένης ενότητας μέσα στην ιστοσελίδα αφιερωμένη αποκλειστικά και μόνο στην βοήθεια και στην επεξήγηση της λειτουργίας του εκπαιδευτικού προγράμματος στον χρήστη. Θα δίνονται σαφείς οδηγίες για το πως πρέπει κάποιος να ξεκινήσει το διάβασμα , επεξήγηση για την συμπλήρωση των ασκήσεων καθώς και βίντεο από τον καθηγητή που θα καθοδηγούν τον χρήστη μέσα στις ενότητες της ιστοσελίδας.
- Μια ακόμα μελλοντική και πιο απαιτητική προέκταση του προγράμματος θα ήταν η δημιουργία «εργαστηριακών μαθημάτων» από τον καθηγητή τα οποία θα υλοποιούνται με τους εξής τρόπους:
  - A. Είτε με βίντεο εκμάθησης διάρκειας λίγων λεπτών που θα αναρτώνται από τον καθηγητή σε συγκεκριμένη ενότητα της ιστοσελίδας.
  - B. Είτε σε ζωντανή μετάδοση από τον καθηγητή σαν μια «εικονική τάξη» όπου ο καθηγητής θα κάνει την παράδοση ζωντανά στους μαθητές που θα είναι συνδεδεμένοι εκείνη την ώρα. Θα υπάρχει μηνιαίο πρόγραμμα παραδόσεων ώστε οι μαθητές να είναι προετοιμασμένοι και δεν θα υπάρχει όριο ατόμων που θα μπορούν να παρακολουθήσουν τις διαλέξεις. Μετά το τέλος της διάλεξης οι μαθητές θα μπορούν να υποβάλουν μέσω ζωντανών άμεσων μηνυμάτων τις απορίες τους στον καθηγητή ώστε να πάρουν απαντήσεις.

Εντάσσοντας αρκετές από τις παραπάνω προεκτάσεις στο πρόγραμμα που υλοποιήσαμε παρόλο που το κάνουν πλήρες και καινοτόμο αλλάζουν την μορφή του ως e-learning με την συμβατική της έννοια. Η δημιουργία ενός προγράμματος e-learning απαιτεί μελέτη των αναγκών των χρηστών στους οποίους απευθύνεται και συνεχής αλληλεπίδραση με τους χρήστες ώστε να κρατάει το ενδιαφέρον τους αμείωτο και να ικανοποιεί στο μέγιστο τις ανάγκες που τους οδήγησαν σε αυτό.

## 10. Βιβλιογραφία

1. Sunny Kumar Aditya and Vikash Kumar Karn. Android SQLite Essentials, Packt Publishing 2014.
2. Mike Owens and Grant Allen. The Definitive Guide to SQLite, 2nd edition, 2010.
3. Jay A. Kreibich. Using SQLite, O'Reilly Media, 2010.
4. Rob Miles. C# Programming Yellow Book, "Cheese" Edition 8.2, November 2016.
5. Hugh E. Williams, Saied M.M. Tahaghoghi. Learning MySQL, O'Reilly Media, Inc. 2006.
6. Bruce Johnson. Professional Visual Studio 2017, Wrox, November 2017.
7. Καμπουράκης Γιώργος , Λουκής Ευριπίδης Ν. Ηλεκτρονική μάθηση, Εκδόσεις Κλειδάριθμος, 2006.
8. William Horton. E-Learning by Design, Second Edition September 2011.
9. Παναγιώτης Συμεωνίδης, Αναστάσιος Γούναρης. Βάσεις, Αποθήκες και Εξόρυξη Δεδομένων με τον SQL Server, ΣΕΑΒ 2015.
10. Steven Feuerstein and Bill Pribyl. Oracle PL/SQL Programming Fifth Edition, O'Reilly 2009.

## 11. Ιστοσελίδες

- <https://www.sqlite.org/index.html>
- <https://www.sqlitetutorial.net>
- <https://www.tutorialspoint.com/sqlite/index.htm>
- <https://www.quackit.com/sqlite/tutorial/>
- <https://www.javatpoint.com/sqlite-tutorial>
- <https://www.datacamp.com/community/tutorials/beginners-guide-to-sqlite>
- <https://onlinelearningconsortium.org/updated-e-learning-definitions-2/>
- <https://economictimes.indiatimes.com/definition/e-Learning>
- <https://docs.oracle.com/en/database/oracle/oracle-database/19/books.html>
- <https://www.postgresql.org/about/>
- <https://www.mysql.com/why-mysql/>
- <https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15>
- <https://www.w3resource.com/sqlite/>
- <https://en.wikipedia.org/wiki/SQL>
- [https://en.wikipedia.org/wiki/E-learning\\_\(theory\)](https://en.wikipedia.org/wiki/E-learning_(theory))
- <https://msdn.microsoft.com/>
- <https://docs.microsoft.com/en-us/aspnet/web-forms/overview/getting-started/getting-started-with-aspnet-45-web-forms/introduction-and-overview>
- <http://www.w3schools.com/>
- <https://www.entityframeworktutorial.net>
- <https://www.tutorialspoint.com/asp.net/>
- <http://reviews.in.gr/greece/education/article/?aid=1231053712>
- [http://elearning.kar.edu.gr/file.php/1/KAR\\_AboutElearningv1.pdf](http://elearning.kar.edu.gr/file.php/1/KAR_AboutElearningv1.pdf)

## 12. Παραρτήματα

### Ερωτήσεις – Απαντήσεις Διδακτικών Ενοτήτων

| ID | QUESTION                                                                                                                                                     | ANSWER 1                                                                                                                                              | ANSWER 2                                                                   | ANSWER 3                                                                                                                                                       | ANSWER 4                                                                                                     | CORRECT_ANSWER                                                                                                                                        | CHAPTER | QUESTION_NO |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------|
| 1  | ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΤΗ ΔΗΜΙΟΥΡΓΙΑ ΤΟΥ ΠΑΡΑΠΑΝΩ ΠΙΝΑΚΑ.                                                                                            | CREATE TABLE Professors ( ID INTEGER PRIMARY KEY, First_Name TEXT NOT NULL, Last_Name TEXT NOT NULL, Department TEXT NOT NULL, Salary REAL NOT NULL); | CREATE TABLE Professors ( ID, First_Name, Last_Name, Department, Salary ); | CREATE DATABASE TABLE Professors ( ID INTEGER PRIMARY KEY, First_Name TEXT NOT NULL, Last_Name TEXT NOT NULL, Department TEXT NOT NULL, Salary REAL NOT NULL); | CREATE TABLE IF NOT EXISTS Professors ;                                                                      | CREATE TABLE Professors ( ID INTEGER PRIMARY KEY, First_Name TEXT NOT NULL, Last_Name TEXT NOT NULL, Department TEXT NOT NULL, Salary REAL NOT NULL); | CREATE  | 1           |
| 2  | ΜΕ ΠΟΙΑ ΕΝΤΟΛΗ ΔΗΜΙΟΥΡΓΟΥΜΕ ΝΕΟ ΠΙΝΑΚΑ ΚΑΙ ΠΑΡΑΛΛΗΛΑ ΕΛΕΓΧΟΥΜΕ ΑΝ ΥΠΑΡΧΕΙ ΗΔΗ ΣΤΗ ΒΑΣΗ ΜΑΣ.                                                                  | *****                                                                                                                                                 | *****                                                                      | *****                                                                                                                                                          | *****                                                                                                        | TABLE IF NOT EXISTS                                                                                                                                   | CREATE  | 2           |
| 3  | ΕΠΙΛΕΞΤΕ ΤΟ ΣΩΣΤΟ QUERY ΓΙΑ ΤΗΝ ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΙΝΑΚΑ Professors του Καθηγητή Jim Fallon, ηλικίας 43 χρονών, στο τμήμα Ναυτιλιακών(Shipping) με μισθό 49.000€ | INSERT INTO TABLE (First_Name,Last_Name,Age,Department,Salary) Values (Jim,Fallon,43,Shipping,49.000)                                                 | INSERT INTO Professors (Jim,Fallon,Shipping,43,49.000);                    | INSERT INTO Professors VALUES (Jim,Fallon,Shipping,43,49.000)                                                                                                  | INSERT INTO Professors (First_Name,Last_Name,Age,Department,Salary) VALUES (Jim,Fallon,43,Shipping,49.000) ; | INSERT INTO Professors (First_Name,Last_Name,Age,Department,Salary) VALUES (Jim,Fallon,43,Shipping,49.000) ;                                          | INSERT  | 1           |

|   |                                                                                                                                                                                       |                                                         |                                                             |                                                               |                                                            |                                                             |        |   |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|-------------------------------------------------------------|---------------------------------------------------------------|------------------------------------------------------------|-------------------------------------------------------------|--------|---|
| 4 | INSERT INTO Professors ('Jim', 'Fallon', 43, 'Shipping', 49.000);                                                                                                                     | EMPTY SPACE                                             | VALUES                                                      | TABLE                                                         | IF EXISTS                                                  | VALUES                                                      | INSERT | 2 |
| 5 | ΣΤΗΝ ΕΚΦΡΑΣΗ Η DROP TABLE Professors [IF EXISTS]; AN ΥΠΑΡΧΕΙ ΕΝΕΡΓΟ ΠΟΙΗΜΕΝΟΣ ΠΕΡΙΟΡΙΣΜΟΣ ΞΕΝΟΥ ΚΛΕΙΔΙΟΥ ΣΤΟΝ ΠΙΝΑΚΑ ΕΠΙΛΕΞΤΕ ΤΙ ΘΑ ΣΥΜΒΕΙ:                                           | Η SQLite ΘΑ ΜΑΣ ΕΜΦΑΝΙΣΕΙ ΜΗΝΥΜΑ ΣΦΑΛΜΑΤΟΣ              | ΑΥΤΟ ΘΑ ΚΑΤΑΡΓΗΘΕΙ ΙΣΟΔΥΝΑΜΑ ΓΙΑ ΚΑΘΕ ΓΡΑΜΜΗ ΤΟΥ ΠΙΝΑΚΑ     | Ο ΠΙΝΑΚΑΣ ΘΑ ΚΑΤΑΡΓΗΘΕΙ ΠΑΡΟΛΟ ΤΟΥ ΠΕΡΙΟΡΙΣΜΟΥ ΞΕΝΟΥ ΚΛΕΙΔΙΟΥ | Η ΔΗΛΩΣΗ DROP TABLE ΑΠΟΤΥΓΧΑΝΕΙ                            | ΑΥΤΟ ΘΑ ΚΑΤΑΡΓΗΘΕΙ ΙΣΟΔΥΝΑΜΑ ΓΙΑ ΚΑΘΕ ΓΡΑΜΜΗ ΤΟΥ ΠΙΝΑΚΑ     | DROP   | 1 |
| 6 | DROP                                                                                                                                                                                  | DROP                                                    | DROP                                                        | DROP                                                          | DROP                                                       | DROP                                                        | DROP   | 2 |
| 7 | Η ΔΙΕΥΘΥΝΣΗ ΤΟΥ ΜΑΘΗΤΗ John Stankoglu ΕΧΕΙ ΚΑΤΑΧΩΡΗΘΕΙ ΕΚΠΑΡΑΔΡΟΜΗΣ ΛΑΘΟΣ ΣΤΟΝ ΠΙΝΑΚΑ Students. ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΤΗΝ ΔΙΟΡΘΩΣΗ ΤΗΣ ΔΙΕΥΘΥΝΣΗΣ ΣΤΗ ΣΩΣΤΗ "Ifigeneias 41". | UPDATE Students Address = 'Ifigeneias 41' WHERE ID = 3; | UPDATE Students WHERE ID = 3 SET Address = 'Ifigeneias 41'; | UPDATE Students SET Address = 'Ifigeneias 41' WHERE ID = 3;   | UPDATE Students SET Address = 'Ifigeneias 41' WHERE ID = 3 | UPDATE Students SET Address = 'Ifigeneias 41' WHERE ID = 3; | UPDATE | 1 |
| 8 | UPDATE Students SET University = 'Piraeus' Department = 'Informatics';                                                                                                                | UPDATE                                                  | UPDATE                                                      | UPDATE                                                        | UPDATE                                                     | ,                                                           | UPDATE | 2 |

|    |                                                                                                                                                                          |                                                           |                                                                             |                                                                               |                                                                                                   |                                                                                                   |        |   |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|--------|---|
| 9  | ΛΟΓΩ ΠΕΡΙΚΟΠΩΝ ΤΟ ΤΜΗΜΑ ΤΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΜΕ ΤΙΤΛΟ 'Informatics' ΚΛΕΙΝΕΙ. ΕΠΙΛΕΞΤΕ ΤΗΝ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΩΣΤΕ ΝΑ ΔΙΑΓΡΑΦΟΥΝ ΟΛΟΙ ΟΙ ΚΑΘΗΓΗΤΕΣ ΠΟΥ ΑΝΗΚΟΥΝ Σ'ΑΥΤΟ ΤΟ ΤΜΗΜΑ . | DELETE *FROM Professors WHERE Department = 'Informatics'  | DELETE FROM Professors WHERE Department = 'Informatics' ;                   | DELETE Professors WHERE Department = 'Informatics' ;                          | DELETE FROM Professors WHERE Department = 'Informatics' ;                                         | DELETE FROM Professors WHERE Department = 'Informatics' ;                                         | DELETE | 1 |
| 10 | DELETE                                                                                                                                                                   | DELETE                                                    | DELETE                                                                      | DELETE                                                                        | DELETE                                                                                            | DELETE                                                                                            | DELETE | 2 |
| 11 | ΕΠΙΛΕΞΤΕ ΤΟ QUERY ΠΟΥ ΕΠΙΣΤΡΕΦΕΙ ΤΙΣ ΣΤΗΛΕΣ ΤΟΥ ΠΑΡΑΠΑΝΩ ΠΙΝΑΚΑ.                                                                                                         | SELECT First_Name,Last_Name,Department FROM Students      | SELECT * FROM Students ;                                                    | SELECT ALL FROM Students ;                                                    | SELECT First_Name,Last_Name,Department FROM Students ;                                            | SELECT First_Name,Last_Name,Department FROM Students ;                                            | SELECT | 1 |
| 12 | SELECT **** * FROM Professors ;                                                                                                                                          | SELECT                                                    | SELECT                                                                      | SELECT                                                                        | SELECT                                                                                            | LAST_NAME FROM                                                                                    | SELECT | 2 |
| 13 | ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΝΑ ΒΡΕΘΟΥΝ ΟΙ ΚΑΘΗΓΗΤΕΣ (ΟΝΟΜΑ & ΕΠΩΝΥΜΟ) ΠΟΥ ΑΝΗΚΟΥΝ ΣΤΟ ΤΜΗΜΑ "Economics" ΚΑΙ ΕΧΟΥΝ ΜΙΣΘΟ ΜΕΓΑΛΥΤΕΡΟ ΑΠΟ 40.000.                        | SELECT * FROM Professors WHERE Department = 'Economics' ; | SELECT FROM Professors WHERE Department = 'Economics' AND Salary > 40.000 ; | SELECT * FROM Professors WHERE Department = 'Economics' AND Salary > 40.000 ; | SELECT First_Name, Last_Name FROM Professors WHERE Department = 'Economics' AND Salary > 40.000 ; | SELECT First_Name, Last_Name FROM Professors WHERE Department = 'Economics' AND Salary > 40.000 ; | WHERE  | 1 |
| 14 | SELECT * FROM Professors WHERE ***** = '*****' ;                                                                                                                         | ****                                                      | ****                                                                        | ****                                                                          | ****                                                                                              | FIRST_NAME GEORGE                                                                                 | WHERE  | 2 |



|    |                                                                                                                                                                                                                                                                                   |                                                                                             |                                                                                             |                                                                                            |                                                                                                     |                                                                                             |       |   |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|-------|---|
| 15 | ΤΟ ΤΜΗΜΑ 'Economics' ΤΟΥ ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΥΣΤΕΡΑ ΑΠΟ ΑΛΛΕΠΑΛΛΕΣ ΠΙΕΣΕΙΣ ΚΑΤΑΦΕΡΕ ΝΑ ΠΑΡΕΙ ΑΥΞΗΣΗ ΣΤΟΥΣ ΜΙΣΘΟΥΣ ΤΩΝ ΚΑΘΗΓΗΤΩΝ ΠΟΥ ΗΤΑΝ ΜΙΚΡΟΤΕΡΟΙ ή ΙΣΟΙ ΑΠΟ 40.000. ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΩΣΤΕ ΟΙ ΚΑΘΗΓΗΤΕΣ ΠΟΥ ΔΙΚΑΙΟΥΝΤΑΙ ΑΥΞΗΣΗ ΣΤΟ ΤΜΗΜΑ 'Economics' ΝΑ ΕΧΟΥΝ ΜΙΣΘΟ 50.000. | UPDATE Professors SET SALARY = 50.000 WHERE SALARY < = 40.000 AND Department = 'Economics'; | UPDATE Professors SET SALARY = 50.000 WHERE SALARY < = 40.000 AND Department = 'Economics'; | UPDATE Professors SET SALARY = 50.000 WHERE SALARY < = 40.000 OR Department = 'Economics'; | UPDATE Professors SET VALUE (SALARY = 50.000) WHERE SALARY < = 40.000 AND Department = 'Economics'; | UPDATE Professors SET SALARY = 50.000 WHERE SALARY < = 40.000 AND Department = 'Economics'; | ANDOR | 1 |
| 16 | SELECT * FROM Professors WHERE Salary >= 40.000 AND Department = 'informatics';                                                                                                                                                                                                   | *****                                                                                       | *****                                                                                       | *****                                                                                      | *****                                                                                               | OR INFORMATICS                                                                              | ANDOR | 2 |
| 17 | ΕΠΙΛΕΞΤΕ ΤΗΝ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΝΑ ΒΡΕΘΕΙ Ο ΚΑΘΗΓΗΤΗΣ ΤΟ ΕΠΩΝΥΜΟ ΤΟΥ ΟΠΟΙΟΥ ΞΕΚΙΝΑΕΙ ΑΠΟ ΤΟ ΓΡΑΜΜΑ 'J' ΚΑΙ ΑΝΗΚΕΙ ΣΤΟ ΤΜΗΜΑ, ΣΤΟΝ ΤΙΤΛΟ                                                                                                                                           | SELECT * FROM Professors WHERE Last_Name LIKE '%J%' AND Department LIKE '%_chem_%';         | SELECT * FROM Professors WHERE Last_Name LIKE 'J%' AND Department LIKE '%chem%';            | SELECT * FROM Professors WHERE Last_Name LIKE '%J%' AND Department LIKE '%chem%';          | SELECT * FROM Professors WHERE Last_Name LIKE 'J_%' AND Department LIKE '%_chem%';                  | SELECT * FROM Professors WHERE Last_Name LIKE 'J%' AND Department LIKE '%chem%';            | LIKE  | 1 |

|    |                                                                                                                                                                                                                                       |       |       |       |       |      |       |   |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|-------|-------|------|-------|---|
|    | <p>ΤΟΥ ΟΠΟΙΟΥ ΠΕΡΙΛΑΜΒΑΝΕΤΑΙ Η ΛΕΞΗ 'chem' .</p>                                                                                                                                                                                      |       |       |       |       |      |       |   |
| 18 | <p>ΣΥΜΠΛΗΡΩΣΤΕ ΤΗΝ ΕΚΦΡΑΣΗ Η ΜΕΤΑ ΤΟΝ ΤΕΛΕΣΤΗ Η ΛIKE ΩΣΤΕ ΝΑ ΒΡΟΥΜΕ ΤΟΝ ΚΑΘΗΓΗΤΗ ΤΟΥ ΟΠΟΙΟΥ ΤΟ ΟΝΟΜΑ ΠΕΡΙΕΧΕΙ ΤΟΝ ΧΑΡΑΚΤΗΡΑ 'g' ΚΑΙ ΤΕΛΕΙΩΝΕΙ ΣΤΟΝ ΧΑΡΑΚΤΗΡΑ 's'.</p> <p>SELECT * FROM Professor s WHERE Last_Name LIKE '*****' .</p> | ***** | ***** | ***** | ***** | %g%s | LIKE  | 2 |
| 19 | <p>ΘΕΛΟΥΜΕ ΝΑ ΒΡΟΥΜΕ ΤΟΥΣ ΥΠΑΛΛΗΛΟΥΣ ΠΟΥ ΒΡΙΣΚΟΝΤΑΙ ΣΤΗΝ 5η ΚΑΙ 6η ΘΕΣΗ ΤΟΥ ΠΙΝΑΚΑ Employees .</p> <p>SELECT * FROM Employees LIMIT ***** OFFSET ***** .</p>                                                                          | ***** | ***** | ***** | ***** | 2 4  | LIMIT | 1 |

| 20 | LIMIT                                                                                                                                | LIMIT                                                                               | LIMIT                                                                                       | LIMIT                                                                                       | LIMIT                                                                                       | LIMIT                                                                                       | LIMIT      | 2 |
|----|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|------------|---|
| 21 | ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΤΟ ΣΥΝΟΛΙΚΟ ΑΘΡΟΙΣΜΑ ΤΩΝ ΜΙΣΘΩΝ ΤΟΥ ΤΜΗΜΑΤΟΣ 'Sales' ΤΟΥ ΠΙΝΑΚΑ Employees.                            | SELECT Department SUM(Salary) FROM Employees WHERE Department = 'Sales';            | SELECT Department, SUM(Salary) FROM Employees WHERE Department = 'sales';                   | SELECT SUM(Salary) FROM Employees;                                                          | SELECT Department, SUM(Salary) FROM Employees WHERE Department = 'Sales';                   | SELECT Department, SUM(Salary) FROM Employees WHERE Department = 'Sales';                   | MINMAX SUM | 1 |
| 22 | ΠΟΙΟΣ ΕΙΝΑΙ Ο ΥΠΑΛΛΗΛΟΣ ΜΕ ΤΟΝ ΜΙΚΡΟΤΕΡΟ ΜΙΣΘΟ ΣΤΟ ΤΜΗΜΑ 'IT' ΣΤΟΝ ΠΙΝΑΚΑ Employees.                                                 | *****                                                                               | *****                                                                                       | *****                                                                                       | *****                                                                                       | AS                                                                                          | MINMAX SUM | 2 |
|    | SELECT MIN(Salary) FROM Employees WHERE Department = 'IT';                                                                           |                                                                                     |                                                                                             |                                                                                             |                                                                                             |                                                                                             |            |   |
| 23 | ΕΠΙΛΕΞΤΕ ΤΟ QUERY ΠΟΥ ΕΠΙΣΤΡΕΦΕΙ ΤΟ ΟΝΟΜΑ, ΤΟ ΕΠΩΝΥΜΟ ΚΑΙ ΤΟΝ ΜΙΣΘΟ ΤΟΥ ΥΠΑΛΛΗΛΟΥ ΜΕ ΤΟΝ ΤΡΙΤΟ ΥΨΗΛΟΤΕΡΟ ΜΙΣΘΟ ΣΤΟΝ ΠΑΡΑΠΑΝΩ ΠΙΝΑΚΑ. | SELECT First_Name, Last_Name FROM Professors ORDER BY Salary DESC LIMIT 1 OFFSET 2; | SELECT First_Name, Last_Name, Salary FROM Professors ORDER BY Salary DESC LIMIT 1 OFFSET 2; | SELECT First_Name, Last_Name, Salary FROM Professors ORDER BY Salary DESC LIMIT 1 OFFSET 2; | SELECT First_Name, Last_Name, Salary FROM Professors ORDER BY Salary DESC LIMIT 1 OFFSET 1; | SELECT First_Name, Last_Name, Salary FROM Professors ORDER BY Salary DESC LIMIT 1 OFFSET 2; | ORDERBY    | 1 |

|    |                                                                                                                                                                                   |                                                                                                |                                                                                                |                                                                                                             |                                                                                    |                                                                                                |         |   |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|---------|---|
| 24 | <p>ΜΑΣ ΖΗΤΕΙΤΑΙ ΝΑ ΒΡΟΥΜΕ ΤΟΥΣ 4 ΠΙΟ ΑΚΡΙΒΟΠΛΗΡΩΜΕΝΟΥΣ ΚΑΘΗΓΗΤΕΣ ΤΟΥ ΠΙΝΑΚΑ.</p> <pre>SELECT First_Name, Last_Name, Salary FROM Professors ORDER BY Salary **** **** **** .</pre> | ****                                                                                           | ****                                                                                           | ****                                                                                                        | ****                                                                               | DESC<br>LIMIT 4                                                                                | ORDERBY | 2 |
| 25 | <p>ΕΠΙΞΕΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΝΑ ΒΡΟΥΜΕ ΠΟΣΟ ΚΟΣΤΙΖΕΙ ΤΟ ΣΥΝΟΛΟ ΤΩΝ ΜΙΣΘΩΝ ΤΩΝ ΥΠΑΛΛΗΛΩΝ ΣΕ ΚΑΘΕ ΤΜΗΜΑ ΤΗΣ ΕΤΑΙΡΙΑΣ .</p>                                                      | <pre>SELECT Department, SUM(Salary) AS TOTAL_SALARY FROM Professors GROUP BY Department;</pre> | <pre>SELECT Department, SUM(Salary) AS TOTAL_SALARY FROM Professors ORDER BY Department;</pre> | <pre>SELECT Department, SUM(Salary) AS TOTAL_SALARY FROM Professors GROUP BY Department; SUM(Salary);</pre> | <pre>SELECT SUM(Salary) AS TOTAL_SALARY FROM Professors GROUP BY Department;</pre> | <pre>SELECT Department, SUM(Salary) AS TOTAL_SALARY FROM Professors GROUP BY Department;</pre> | GROUPBY | 1 |
| 26 | <p>ΜΑΣ ΖΗΤΕΙΤΑΙ ΤΟ ΤΜΗΜΑ ΤΗΣ ΕΤΑΙΡΙΑΣ ΠΟΥ ΚΟΣΤΙΖΕΙ ΤΟ ΠΕΡΙΣΣΟ ΤΕΡΟ ΚΑΘΩΣ ΚΑΙ ΤΟ ΠΟΣΟ ΠΟΥ ΚΟΣΤΙΖΕΙ .</p> <pre>SELECT Department, SUM(SALARY) AS TOTAL_SALARY FROM</pre>            | ****                                                                                           | ****                                                                                           | ****                                                                                                        | ****                                                                               | GROUP<br>ORDER<br>DESC                                                                         | GROUPBY | 2 |

|    |                                                                                                                                                                                                        |                                            |                                     |                                            |                                  |                                            |               |   |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|-------------------------------------|--------------------------------------------|----------------------------------|--------------------------------------------|---------------|---|
|    | PROFESSORS<br>***** BY Department<br>***** BY SUM(SALARY) *****<br>LIMIT 1;                                                                                                                            |                                            |                                     |                                            |                                  |                                            |               |   |
| 27 | ΒΡΕΙΤΕ ΤΟΝ ΑΡΙΘΜΟ ΤΟΝ ΣΥΓΓΡΑΦΕΩΝ ('Author') ΠΟΥ ΥΠΑΡΧΟΥΝ ΣΤΟΝ ΠΙΝΑΚΑ 'Books' ΧΩΡΙΣ ΔΙΠΛΟΤΥΠΑ!!                                                                                                         | SELECT DISTINCT(COUNT(Author)) FROM Books; | SELECT DISTINCT(Author) FROM Books; | SELECT COUNT(DISTINCT(Author)) FROM Books; | SELECT COUNT(Author) FROM Books; | SELECT COUNT(DISTINCT(Author)) FROM Books; | DISTINCTCOUNT | 1 |
| 28 | ΒΡΕΙΤΕ ΤΟ ΤΜΗΜΑ (Department) ΜΕ ΤΟΥΣ ΠΕΡΙΣΣΟΤΕΡΟΥΣ ΥΠΑΛΛΗΛΟΥΣ ΚΑΙ ΤΟΝ ΑΚΡΙΒΗ ΑΡΙΘΜΟ ΤΟΥΣ.<br>SELECT Department , *****()<br>FROM Professors<br>GROUP BY *****<br>ORDER BY COUNT()<br>*****<br>LIMIT 1; | *****                                      | *****                               | *****                                      | *****                            | COUNT Department DESC                      | DISTINCTCOUNT | 2 |

|    |                                                                                                                                            |                                                                                                                                     |                                                                                                                              |                                                                                                                                    |                                                                                                                                     |                                                                                                                                     |         |   |
|----|--------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|---------|---|
| 29 | ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΩΣΤΕ ΝΑ ΒΡΕΘΕΙ ΤΟ ΤΜΗΜΑ ΜΕ ΤΟΝ ΥΨΗΛΟΤΕΡΟ Μ.Ο. ΣΕ ΜΙΣΘΟΥΣ ΚΑΘΩΣ ΚΑΙ ΤΟΝ ΜΕΣΟ ΜΙΣΘΟ ΤΟΥ ΤΜΗΜΑΤΟΣ.                 | SELECT Department , AVG(Salary) FROM Professors GROUP BY Department ORDER BY AVG(Salary) DESC LIMIT 1;                              | SELECT Department , AVG(Salary) FROM Professors GROUP BY Salary DESC LIMIT 1;                                                | SELECT Department , AVG(Salary) FROM Professors GROUP BY Department DESC ;                                                         | SELECT Department , AVG(Salary) FROM Professors GROUP BY AVG(Salary) ORDER BY Department DESC LIMIT 1;                              | SELECT Department , AVG(Salary) FROM Professors GROUP BY Department ORDER BY AVG(Salary) DESC LIMIT 1;                              | AVERAGE | 1 |
| 30 | ΝΑ ΒΡΕΘΕΙ Ο ΜΕΣΟΣ ΟΡΟΣ ΣΕΛΙΔΩΝ ΚΑΘΕ ΣΥΓΓΡΑΦΕΑ ('Author') ΤΟΥ ΠΙΝΑΚΑ 'Books'.                                                               | ****                                                                                                                                | ****                                                                                                                         | ****                                                                                                                               | ****                                                                                                                                | Author GROUP ORDER                                                                                                                  | AVERAGE | 2 |
| 31 | ΕΠΙΛΕΞΤΕ ΤΗ ΣΩΣΤΗ ΑΠΑΝΤΗΣΗ ΓΙΑ ΝΑ ΒΡΕΘΟΥΝ ΟΙ ΣΥΓΓΡΑΦΕΙΣ ΠΟΥ ΕΧΟΥΝ ΜΕΤΑΞΥ ΕΝΟΣ ΚΑΙ ΔΥΟ ΒΙΒΛΙΩΝ ΣΤΟΝ ΠΙΝΑΚΑ ΤΑΞΙΝΟΜΗΜΕΝΟΙ ΜΕ ΦΘΙΝΟΥΣΑ ΣΕΙΡΑ. | SELECT Author , COUNT(Book_Name) FROM Books GROUP BY Author HAVING COUNT(Book_Name) BETWEEN 1 AND 2 ORDER BY COUNT(Book_Name) DESC; | SELECT Author , Book_Name FROM Books GROUP BY Author HAVING COUNT(Book_Name) BETWEEN 1 AND 2 ORDER BY COUNT(Book_Name) DESC; | SELECT Author , COUNT(Book_Name) FROM Books GROUP BY Author HAVING COUNT(Book_Name) BETWEEN 1 OR 2 ORDER BY COUNT(Book_Name) DESC; | SELECT Author , COUNT(Book_Name) FROM Books GROUP BY Author HAVING COUNT(Book_Name) BETWEEN 1 AND 2 ORDER BY COUNT(Book_Name) DESC; | SELECT Author , COUNT(Book_Name) FROM Books GROUP BY Author HAVING COUNT(Book_Name) BETWEEN 1 AND 2 ORDER BY COUNT(Book_Name) DESC; | HAVING  | 1 |

|           |                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                        |                                                                                                                                                                                              |                                                                                                                                                                                   |                                                                                                                                                                                             |                                                                                                                                                                                        |               |          |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|----------|
| <p>32</p> | <p>ΝΑ<br/>ΒΡΕΘΟΥΝ<br/>ΤΑ<br/>ΒΙΒΛΙΑ<br/>ΠΟΥ<br/>ΕΧΟΥΝ<br/>ΠΑΝΩ<br/>ΑΠΟ 390<br/>ΣΕΛΙΔΕΣ<br/>ΤΑΞΙΝΟΜ<br/>ΗΜΕΝΑ<br/>ΣΕ<br/>ΦΘΙΝΟΥ<br/>ΣΑ<br/>ΣΕΙΡΑ.</p> <p>SELECT<br/>Book_Name,<br/>Pages</p> <p>FROM<br/>Books</p> <p>GROUP<br/>BY *****</p> <p>*****<br/>Pages &gt;<br/>390</p> <p>ORDER<br/>BY Pages<br/>DESC;</p> | <p>*****</p>                                                                                                                                                                           | <p>*****</p>                                                                                                                                                                                 | <p>*****</p>                                                                                                                                                                      | <p>*****</p>                                                                                                                                                                                | <p>Book_Name<br/>HAVING</p>                                                                                                                                                            | <p>HAVING</p> | <p>2</p> |
| <p>33</p> | <p>ΕΠΙΛΕΞΤΕ<br/>ΤΗ<br/>ΣΩΣΤΗ<br/>ΑΠΑΝΤΗ<br/>ΣΗ ΩΣΤΕ<br/>ΝΑ<br/>ΒΡΕΙΤΕ<br/>ΤΟΥΣ<br/>ΑΝΘΡΩΠ<br/>ΟΥΣ<br/>(First_Name,<br/>Last_Name)<br/>ΠΟΥ<br/>ΕΧΟΥΝ<br/>ΕΠΙΛΕΞΕΙ<br/>ΤΟ<br/>(Check_Out)<br/>ΤΟ<br/>ΒΙΒΛΙΟ<br/>ΜΕ<br/>ΤΙΤΛΟ<br/>'Lord of<br/>the Flies'.</p>                                                         | <p>SELECT<br/>First_Name,<br/>Last_Name<br/>FROM<br/>Check_Out JOIN<br/>Books ON<br/>Check_Out.Book_Id =<br/>Books.Id<br/>WHERE<br/>Books.Book_Name =<br/>'Lord of<br/>the Flies';</p> | <p>SELECT<br/>First_Name,<br/>Last_Name<br/>FROM<br/>Check_Out JOIN<br/>Books ON<br/>Check_Out.Book_Id ==<br/>Books.Id<br/>WHERE<br/>Books.Book_Name ==<br/>'Lord<br/>of the<br/>Flies';</p> | <p>SELECT<br/>First_Name,<br/>Last_Name<br/>FROM<br/>Check_Out JOIN<br/>Books ON<br/>Check_Out.Id =<br/>Books.Id<br/>WHERE<br/>Books.Book_Name =<br/>'Lord of<br/>the Flies';</p> | <p>SELECT<br/>First_Name,<br/>Last_Name<br/>FROM<br/>Check_Out JOIN<br/>Books ON<br/>Check_Out.Book_Id =<br/>Books.Id<br/>WHERE<br/>Books.Book_Name ==<br/>'Lord<br/>of the<br/>Flies';</p> | <p>SELECT<br/>First_Name,<br/>Last_Name<br/>FROM<br/>Check_Out JOIN<br/>Books ON<br/>Check_Out.Book_Id =<br/>Books.Id<br/>WHERE<br/>Books.Book_Name =<br/>'Lord of<br/>the Flies';</p> | <p>JOIN</p>   | <p>1</p> |

|    |                                                                                                                                                                                                                                                          |       |       |       |       |                  |      |   |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|-------|-------|------------------|------|---|
| 34 | <p>ΒΡΕΙΤΕ ΠΟΣΕΣ ΦΟΡΕΣ ΕΧΕΙ ΕΠΙΛΕΓΕΙ ΚΑΘΕ ΒΙΒΛΙΟ ΑΠΟ ΤΟΥΣ ΧΡΗΣΤΕ Σ:</p> <pre> SELECT Book_Name, Author, ***** (Book_Name) AS TIMES_C HOSEN  FROM Check_Out JOIN Books ON Check_Out.***** = Books.Id  GROUP BY Author  ORDER BY TIMES_C HOSEN DESC; </pre> | ***** | ***** | ***** | ***** | COUNT<br>Book_Id | JOIN | 2 |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|-------|-------|-------|------------------|------|---|