ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ: ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ: ΕΡΕΥΝΗΤΙΚΗΣ ΟΜΑΔΑΣ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ ΑΝΘΡΩΠΟΥ ΥΠΟΛΟΓΙΣΤΗ

Διπλωματική Εργασία

του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής του Πανεπιστημίου Πατρών

Ξηροτύρης Νικόλαος του Κωνσταντίνου

Αριθμός Μητρώου: 6596

Θέμα

«Σχεδιασμός και Ανάπτυξη Προσαρμοζόμενου Συστήματος για την Διδασκαλία του Μαθήματος Βάσεις Δεδομένων»

Επιβλέπων

Νικόλαος Αβούρης, Καθηγητής

Αριθμός Διπλωματικής Εργασίας:

Πάτρα, Ιούλιος 2012

ΠΙΣΤΟΠΟΙΗΣΗ

Πιστοποιείται ότι η Διπλωματική Εργασία με θέμα

«Σχεδιασμός και Ανάπτυξη Προσαρμοζόμενου Συστήματος για την Διδασκαλία του Μαθήματος Βάσεις Δεδομένων»

Του φοιτητή του Τμήματος Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

Ξηροτύρη Νικολάου του Κωνσταντίνου

Αριθμός Μητρώου: 6596

Παρουσιάστηκε δημόσια και εξετάστηκε στο Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών στις

……../……../……

Ο Επιβλέπων	Ο Διευθυντής του Τομέα
Καθηγητής	Καθηγητής
κ. Νικόλαος Αβούρης	κ. Ευθύμιος Χούσος

Αριθμός Διπλωματικής Εργασίας:

Θέμα: «Σχεδιασμός και Ανάπτυξη Προσαρμοζόμενου Συστήματος για την Διδασκαλία του Μαθήματος Βάσεις Δεδομένων»

Φοιτητής : Ξηροτύρης Νικόλαος
Επιβλέπων : Καθηγητής Νικόλαος Αβούρης

Περίληψη

Ο στόχος της παρούσας διπλωματικής εργασίας είναι η παρουσίαση της χρησιμότητας του ηλεκτρονικού υπολογιστή στην εκπαιδευτική διαδικασία, η ευρεία χρήση των βάσεων δεδομένων στον σύγχρονο κόσμο καθώς και η μελέτη των λειτουργικών περιοχών μίας βάσης δεδομένων. Επίσης, παρουσιάζεται ο τρόπος με τον οποίο οργανώνονται τα δεδομένα μιας βάσης δεδομένων στο σκληρό δίσκο και η χρησιμότητα των ευρετηρίων.

Στο **πρώτο κεφάλαιο**, γίνεται μια εισαγωγή στην χρησιμότητα του ηλεκτρονικού υπολογιστή στην καθημερινότητα μας και κυρίως, στην εισβολή του στην εκπαιδευτική διαδικασία. Παρουσιάζονται κάποια κύρια στοιχεία που κάνουν τον Η/Υ απαραίτητο - πλέον - στην εκπαίδευση, οι τρόποι με τους οποίους μπορεί να χρησιμοποιηθεί στην εκπαίδευση και φυσικά η μεγάλη συμβολή της συνεργατικής μάθησης με την βοήθεια των ηλεκτρονικών υπολογιστών.

Στο **δεύτερο κεφάλαιο**, αναφέρεται η γενική ιδέα των Βάσεων Δεδομένων, η εξέλιξή τους και κάποια ιστορικά στοιχεία. Στη συνέχεια παρουσιάζονται τα βασικά στοιχεία της δομής των βάσεων δεδομένων και οι λειτουργικές περιοχές τους (αρχιτεκτονική, μοντέλα, γλώσσες, ασφάλεια).

Στο **τρίτο κεφάλαιο**, ασχολούμαστε με την αποθήκευση των βάσεων δεδομένων, κυρίως σε δευτερεύοντα αποθηκευτικά μέσα, όπως ο σκληρός δίσκος. Αναφέρονται τα επιμέρους στοιχεία ενός σκληρού δίσκου και τον τρόπο με τον οποίο αποθηκεύονται τα δεδομένα σε μορφή εγγραφών, τον

τρόπο οργάνωσης και διάταξης των εγγραφών σε αρχεία καθώς και περιληπτικά η μέθοδος του κατακερματισμού.

Στο **τέταρτο κεφάλαιο**, παρουσιάζεται η δομή των ευρετηρίων για την διάταξη των εγγραφών σε ένα αρχείο. Αναφέρονται οι βασικοί τύποι ευρετηρίων (πρωτέυον, δευτερεύον, συστάδων και πολυεπίπεδο) καθώς και τα δυναμικά πολυεπίπεδα ευρετήρια με την χρήση B-δένδρων και B^+-δένδρων.

Στο **πέμπτο κεφάλαιο**, παρουσιάζεται μία εφαρμογή των ευρετηρίων σε μία δειγματική βάση δεδομένων (Sakila), η οποία χρησιμοποιείται για εκπαιδευτικούς σκοπούς. Συντάσσονται ερωτήματα SQL με την βοήθεια εντολών δημιουργίας ευρετηρίων, ώστε να δειχθεί και πειραματικά ότι τα ευρετήρια είναι μία πολύ καλή μέθοδος μείωσης του χρόνου αναζήτησης, ειδικά σε μία βάση με πάνω από 1000-2000 εγγραφές.

ABSTRACT

The aim of the present thesis is to show the utility of computers in education, the widespread use of databases in the modern world and the study of the functional areas of a database. It is also presented the way the data are organized in a database on the hard disk and the structure of indexes.

The **first chapter** includes an introduction to the usefulness of the personal computer to our everyday life and especially, in its invasion in the educational process. There are presented some key elements that make the PC necessary in education, the ways in which it can be used in education and of course the great contribution of collaborative learning with the help of computers.

The **second chapter** refers to the concept of databases, their development and some historical data. Also, there are presented the basic elements of the structure of databases and their functional areas (architecture, models, languages, security).

The **third chapter** deals with the storage of databases, particularly in a secondary storage media, such as the hard disk. There are indicated the components of a hard disk and how the data is stored in records, the organization and provision of records in files and briefly the method of hashing.

The **fourth chapter** presents the structure of indexes for the order of the records in a file. The basic types of indexes are described (primary, secondary, clustering and multilevel), as well as, the dynamic multi-level indexes using B-trees and B^+-trees.

The **fifth chapter** presents an application of indexes in a sample database (Sakila), which is used for educational purposes. SQL queries are drafted with the help of create indexes command, in order to show – experimentally - that the indexes are a great way to reduce search time, especially, in a database which contains over 1000-2000 records.

ΠΡΟΛΟΓΟΣ

Η εκπόνηση της παρούσας διπλωματικής εργασίας έγινε υπό την επίβλεψη και τη καθοδήγηση του καθηγητή κ. Νικόλαου Αβούρη καθώς και του μεταπτυχιακού κ. Γεώργιου Πασχάλη το διάστημα 10/2010 – 07/2012. Ο στόχος της εργασίας, όπως αναφέρθηκε είναι η χρησιμότητα των βάσεων δεδομένων και η αξιοποίηση των ευρετηρίων.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Νικόλαο Αβούρη για την συνεργασία και το χρόνο που μου πρσέφερε το παραπάνω χρονικό διάστημα, καθώς επίσης και τον κ. Γιώργο Πασχάλη. Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου για την αμέριστη υποστήριξη που μου έδειξε – με κάθε μέσο – καθ' όλη τη διάρκεια των σπουδών μου, και ιδιαίτερα τον πατέρα μου, ο οποίος δίνει την πιο δύσκολη μάχη της ζωής του και τα πηγαίνει περίφημα.

Πάτρα 2012

Ξηροτύρης Νικόλαος

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1 : Σχέση Ανθρώπου - Υπολογιστή στην Εκπαίδευση 13

1.1 Εισαγωγή 13

1.2 Υπολογιστής Στην Εκπαίδευση 16

1.3 Ηλεκτρονικός Υπολογιστής & Συνεργατική Μάθηση 20

ΚΕΦΑΛΑΙΟ 2 : Βάσεις Δεδομένων 25

2.1 Εισαγωγή 25

2.2 Δεδομένα και Πληροφορία 26

2.2.1 Εισαγωγή στις Βάσεις Δεδομένων και τα ΣΔΒΔ και ορισμός 27

2.3 Ιστορικά Στοιχεία 28

2.3.1 Η Γενική Ιδέα της Βάσης Δεδομένων 28

2.3.2 Εξέλιξη των Βάσεων Δεδομένων και των ΣΔΒΔ 29

2.4 Δομή μιας Βάσης Δεδομένων 32

2.4.1 Σύνοψη 32

2.4.2 Σύστημα Βάσης Δεδομένων 33

2.5 Λειτουργικές περιοχές των Βάσεων Δεδομένων 34

2.5.1 Η Αρχιτεκτονική των ΣΔΒΔ 34

2.5.2 Μοντέλα Δεδομένων 36

2.5.3 Σχεσιακά ΣΔΒΔ (RDBMS) και Γλώσσες Βάσεων Δεδομένων 39

2.5.4 Ασφάλεια Βάσεων Δεδομένων 41

ΚΕΦΑΛΑΙΟ 3 : Αποθηκευτικές Μονάδες Δίσκων, Οργανώσεις Αρχείων και Κατακερματισμός 43

3.1 Εισαγωγή 43

3.2 Αποθήκευση Βάσεων Δεδομένων 44

3.3 Μαγνητικοί Δίσκοι 46

3.3.1 Παράμετροι των Δίσκων 48

3.3.2 Μεταφορά Μπλοκ σε Ενδιάμεση Μνήμη 49

3.4 Εγγραφές 50

3.4.1 Αρχεία, Εγραφές Σταθερού και Μεταβλητού Μήκους 51

3.4.2 Ομαδοποίηση Εγγραφών 52

3.4.3 Πράξεις Σε Εγγραφές Αρχείων 53

3.5 Διάταξη Εγγραφών Σε Αρχεία 54

3.5.1 Μη Διατεταγμένα Αρχεία – Αρχεία Σωρού (Heap File) 54

3.5.2 Ταξινομημένα Αρχεία 55

3.6 Κατακερματισμός 56

3.6.1 Εσωτερικός Κατακερματισμός 56

3.6.2 Εξωτερικός Κατακερματισμός 58

ΚΕΦΑΛΑΙΟ 4 : Δομές Ευρετηρίων για Αρχεία 63

4.1 Εισαγωγή 63

4.2 Διατεταγμένα Ευρετήρια Ενός Επιπέδου 65

4.2.1 Πρωτεύον Ευρετήριο 65

4.2.2 Ευρετήρια Συστάδων 67

4.2.3 Δευτερεύον Ευρετήριο 67

4.3 Ευρετήρια Πολλών Επιπέδων 69

4.4 Δυναμικά Πολυεπίπεδα Ευρετήρια Με Τη Χρήση Β-Δένδρων Και B^+-Δένδρων 71

4.4.1 Δένδρα Αναζήτησης 71

4.4.2 Β-Δένδρα 73

4.4.3 B^+-Δένδρα 75

ΚΕΦΑΛΑΙΟ 5 : Εφαρμογή των Ευρετηρίων σε Δειγματική Βάση Δεδομένων 79

5.1 Εισαγωγή 79

5.2 Η Βάση Δεδομένων Sakila 80

5.2.1 Το σχήμα της Sakila 81

5.2.2 Οι πίνακες της Sakila 82

5.3 Ερωτήματα στην MySQL 86

5.4 Συμπεράσματα 93

ΚΕΦΑΛΑΙΟ 1

Σχέση Υπολογιστή – Ανθρώπου Στην Εκπαίδευση

1.1 Εισαγωγή

Η πλειονότητα των ανθρώπων μέχρι τα μέσα του 19^{ου} αιώνα, διέθετε πάρα πολύ χρόνο για την καλλιέργεια και συλλογή των καρπών. Οι γεωργικές δραστηριότητες ήταν ένας τρόπος ζωής. Η βιομηχανική επανάσταση στα μέσα του 19^{ου} αιώνα επέφερε σημαντικές αλλαγές. Για πρώτη φορά στην ιστορία άρχισαν να δουλεύουν στις βιομηχανίες περισσότεροι άνθρωποι από ότι στους αγρούς. Η εποχή της βιομηχανίας είχε αρχίσει.

Σήμερα ζούμε στην εποχή της πληροφορίας. Καθώς τα εργοστάσια γίνονται περισσότερο αυτοματοποιημένα, εργάζονται λιγότεροι άνθρωποι για την παραγωγή αγαθών. Η ίδια χρονική περίοδος χαρακτηρίζεται από την έκρηξη των γνώσεων, της πληροφορίας και της ραγδαίας ανάπτυξης της τεχνολογίας, με ξεχωριστή θέση της τεχνολογίας των ηλεκτρονικών υπολογιστών. Συνώνημο της πληροφορίας είναι ο όρος *δεδομένα* (data). Νέα συστήματα επικοινωνίας μας επιτρέπουν να αξιοποιούμε τα δεδομένα-πληροφορίες με σχετική ευκολία. Για το λόγο αυτό πολλοί άνθρωποι αναφέρονται στην εποχή μας χαρακτηρίζοντάς την ως η εποχή της πληροφορίας.

Από όλα τα τεχνολογικά επιτεύγματα της εποχής, αυτό που ξεχώρισε και άνοιξε νέους ορίζοντες για ένα καλύτερο κόσμο είναι ο *Ηλεκτρονικός Υπολογιστής(Η/Υ)*. Ξεκινώντας σαν ένα εργαλείο που προοριζόταν για επιστημονική έρευνα, εξελίχθηκε σε ένα πολυεργαλείο γενικής χρήσεως. Λόγω αυτής της γενικότητάς του έχει εισβάλει σε όλους τους τομείς της ανθρώπινης δραστηριότητας.

Η αρχική εμφάνιση των ηλεκτρονικών υπολογιστών και στη συνέχεια η εισβολή τους σχεδόν σε κάθε πτυχή της καθημερινής μας ζωής συμβάλλει στην αστραπιαία διακίνηση των γνώσεων και των ιδεών, καθώς και στην αντιμετώπιση πολλών πολύπλοκων καταστάσεων. Ο ηλεκτρονικός υπολογιστής έχει εισχωρήσει σε όλους τους τομείς της επιστήμης και κάθε άλλης παραγωγικής δραστηριότητας συμβάλλοντας έτσι, με έμμεσο και άμεσο τρόπο, στην ίδια τη ραγδαία εξέλιξή τους. Είναι προφανές πως οι κοινωνικές επιπτώσεις από τη νέα αυτή παραγωγική δύναμη είναι σημαντικές, σύνθετες και ως ένα σημείο, απρόβλεπτες. Επηρεάζουν άμεσα την ποιότητα της ζωής μας, ακόμη και τη διαμόρφωση του χαρακτήρα μας και τις κοινωνικές μας σχέσεις. Είναι ένα πεδίο που χάνονται και κερδίζονται οικονομικοί και πολιτικοί πόλεμοι, όμως και ένα μέσο που ανοίγει νέους ορίζοντες επικοινωνίας.

Όσο περνά ο καιρός, τόσο περισσότερο αναγνωρίζεται ότι οι πολίτες του αύριο θα πρέπει να αποκτήσουν στοιχειώδη μόρφωση πάνω στους ηλεκτρονικούς υπολογιστές, για να είναι σε θέση να αντεπεξέλθουν στην τεχνολογική πολυπλοκότητα ή στην χειρότερη περίπτωση πρέπει να έχουν ιδέα για το πόσο ζωτικός είναι ο ρόλος του στο μοντέρνο κόσμο.

Είναι αυτονόητο ότι η εκπαίδευση, που είναι αναπόσπαστο τμήμα της κοινωνίας δεν θα μπορούσε να μείνει ανεπηρέαστη. Υποχρεωτικά έπρεπε να προσαρμοστεί σε αυτή την πραγματικότητα και να προσπαθήσει να ετοιμάζει πολίτες, οι οποίοι να έχουν τις απαραίτητες γνώσεις και δεξιότητες ώστε να χρησιμοποιούν και να αξιοποιούν ουσιαστικά την σύγχρονη τεχνολογία, καλλιεργώντας τους ταυτόχρονα και τις απαραίτητες δεξιότητες για πραγματική συνεργασία.

Από την πληθώρα των μέσων που προσφέρει σήμερα η εκπαιδευτική τεχνολογία, σημαντική θέση κατέχει ο ηλεκτρονικός υπολογιστής (Η/Υ), του οποίου η εισαγωγή και χρήση στην εκπαίδευση, όσον αφορά τη διδακτική υποστήριξη διάφορων αντικειμένων, βρίσκεται ακόμη σε πρώιμο στάδιο. Επομένως, είναι προφανές ότι ο χώρος προσφέρεται για δημιουργική ενασχόληση και ειδικότερα για την διερεύνηση των δυνατοτήτων συμβολής της πληροφορικής στην εποπτικοποίηση της διδακτικής πράξης, γεγονός που αποτελεί συστηματική επιδίωξη της σύγχρονης διδακτικής.

Ο ηλεκτρονικός υπολογιστής μπορεί να χρησιμοποιηθεί στην εκπαίδευση με τρεις κυρίως τρόπους:

- ως ξεχωριστό μάθημα,
- σαν ένα δυνατό βοήθημα διδασκαλίας, και
- για τα διοικητικά του κάθε εκπαιδευτικού οργανισμού.

Ένας άλλος υποσχόμενος νεοαναπτυχθείς τομέας στο χώρο της πληροφορικής είναι αυτός των δικτύων, έχοντας ως καλύτερο αντιπρόσωπο το Internet. Αυτή η εξέλιξη δίνει τη δυνατότητα στους χρήστες να πλοηγηθούν σε ένα τεράστιο χώρο πληροφοριών, να ανταλλάξουν ηλεκτρονικά μηνύματα (e-mail), αλλά και να πάρουν μέρος σε online συζητήσεις.

Οι Η/Υ κερδίζουν μέρα με τη μέρα όλο και σημαντικότερο ρόλο στη ζωή μας. Έχουν ήδη αλλάξει ριζικά τον τρόπο που ζούμε και δουλεύουμε. Η λογική συνέπεια είναι να αλλάξουν και την φύση της εκπαίδευσης καθώς και τις δεξιότητες που χρειάζονται να αναπτύξουν οι μαθητές μας. Η ύπαρξη επεξεργαστή κειμένου έχει αντικαταστήσει τις χειρόγραφες εργασίες, κάνοντας με τον τρόπο αυτό καλό τον γραφικό χαρακτήρα και όλες τις δεξιότητες που σχετίζονται με ένα εμφανίσιμο χειρόγραφο, λιγότερο αναγκαίες. Επιπλέον, τα γνωστά σε όλους μας κομπιουτεράκια μπορούν να κάνουν ακόμα και τις πιο περίπλοκες πράξεις με φοβερή ακρίβεια, θέτοντας σε προτεραιότητα την ανάπτυξη των δεξιοτήτων της λογικής σκέψης και της εκτίμησης για το ποια πράξη πρέπει να πραγματοποιηθεί. Ταυτόχρονα, οι μαθητές θα πρέπει να αναπτύσσουν τις ικανότητές τους στην έρευνα και την κρίση των πληροφοριών που είναι διαθέσιμες μέσω των CD-ROM, των βάσεων δεδομένων και φυσικά του Internet.

Από τα μέσα του αιώνα που διανύουμε, παρατηρούμε την συνεχόμενη προσπάθεια να ενταχθούν οι νέες τεχνολογίες στην εκπαίδευση, κάτι στο οποίο δεν θα μπορούσαν να μην πάρουν μέρος οι ηλεκτρονικοί υπολογιστές. Πλέον, έχουν γίνει αναπόσπαστο κομμάτι όχι μόνο της καθημερινής ζωής -όπως αναφέρθηκε παραπάνω- αλλά πρέπει να γίνει και αναπόσπαστο κομμάτι της εκπαιδευτικής διαδικασίας. Ταυτόχρονα, γίνεται όλο και πιο δημοφιλής η χρήση της συνεργατικής μάθησης σαν διδακτική πρακτική και εδραιώνεται όλο και περισσότερο η άποψη ότι η συνεργατική μάθηση είναι μια διδακτική προσσέγγιση που υπερτερεί έναντι των άλλων, αφού σε αυτήν, τα

πλεονεκτήματα είναι πολύ περισσότερα από τα τυχόν μειονεκτήματα της. Παρακάτω θα υπάρξει εκτενέστερη αναφορά στο συγκεκριμένο θέμα.

1.2 Υπολογιστής Στην Εκπαίδευση

Πολλοί υποστηρίζουν σήμερα ότι ο υπολογιστής είναι το καλύτερο εργαλείο που μπορεί να μας πάει απο την θεωρία στην πράξη. Η άποψη αυτή στηρίζεται σε δύο βασικά χαρακτηριστικά του ηλεκτρονικού υπολογιστή. Το πρώτο είναι ότι μπορεί να επεξεργαστεί μεγάλο όγκο δεδομένων πολύ γρήγορα και να συνδυάσει πολλά μέσα. Μπορεί να χρησιμοποιηθεί σαν πίνακας, σαν μέσο αναπαραγωγής ήχου και εικόνας ή σαν συνδυασμός όλων των νέων τεχνολογιών των πολυμέσων ή υπερμέσων, ανάλογα με τις ανάγκες που θέλει να καλύψει ο εκπαιδευτικός και τους στόχους που έχει βάλει. Το δεύτερο χαρακτηριστικό του υπολογιστή είναι ότι μας δίνει τη δυνατότητα να έχουμε πληροφορία οποιουδήποτε τύπου, τη στιγμή που τη θέλουμε και να εκμεταλλεύεται κατά την επιθυμία του δασκάλου.

Είναι κατανοητό από όλους βέβαια, ότι οι υπολογιστές δεν είναι μαγικά εργαλεία και ότι από μόνοι τους δεν μπορούν να αλλάξουν την μαθησιακή διδασκαλία. Από έρευνες των Elliot (1994), Salomon (1990) και De Corte (1994) τονίζεται ο ουσιαστικός ρόλος που μπορεί να διαδραματίσει η εκπαιδευτική τεχνολογία στη δημιουργία υποστηρικτικού μαθησιακού περιβάλλοντος, υπό το πρίσμα της Γνωστικής Επιστήμης.

Σύμφωνα με τα δεδομένα πρόσφατων ερευνών, οι υπολογιστές παρουσιάζουν πολλές εκπαιδευτικές δυνατότητες. Ενδεικτικά αναφέρονται οι παρακάτω:

- Η μάθηση γίνεται πιο ενδιαφέρουσα και διασκεδαστική.
- Τα γεγονότα και οι πληροφορίες παρουσιάζονται με πολλαπλό τρόπο (κείμενο, ήχος, εικόνα, βίντεο)
- Τονίζεται ο ενεργητικός ρόλος του μαθητή στη διαδικασία της μάθησης.
- Εξατομικεύεται η διδασκαλία.
- Είτε ο δάσκαλος, είτε ο μαθητής μπορεί να έχει τον έλεγχο της διαδικασίας.
- Η μαθησιακή δραστηριότητα συνδέεται με την καθημερινή ζωή.
- Δημιουργούνται συνθήκες συνεργατικής μάθησης.
- Υπογραμμίζεται ο διευκολυντικός, παροτρυντικός, συντονιστικός και διαμεσολαβητικός ρόλος του εκπαιδευτικού στη μαθησιακή διαδικασία.

Εικόνα 1.1 *Η συμμετοχή των υπολογιστών στην εκπαιδευτική διαδικασία*.

Πρέπει να τονίσουμε ότι για να αξιοποιηθούν οι παραπάνω δυνατότητες του υπολογιστή, απαιτείται κατά κύριο λόγο το ανάλογο μαθησιακό περιβάλλον και κατά δεύτερο λόγο, το κατάλληλο λογισμικό.

Η ανάγκη χρήσης-αξιοποίησης του Ηλεκτρονικού Υπολογιστή στην αίθουσα διδασκαλίας ως διαχειριστή των σύγχρονων Μέσων Πολλαπλής ή Πολύμορφης Επικοινωνίας/Διδασκαλίας (Multimedia) για την υποβοήθηση του καθηγητή και του μαθητή έχει προσχωρήσει και στη συνείδηση όλων μας αλλά και στην πράξη, στο βαθμό βέβαια που το διαθέσιμο hardware, software αλλά και humanware (!) το επιτρέπουν.

Παρ' όλο ότι η δυνατότητα αμφίδρομης αλληλεπίδρασης χρήστη-προγράμματος καθιστά τον υπολογιστή ένα ελκυστικό παιχνίδι και συνεπώς ένα χρήσιμο εργαλείο, είναι ευρέως αποδεκτό ότι η εργαστηριακή πρακτική είναι το πιο αποτελεσματικό μέσο εκπαίδευσης και έρευνας και δεν μπορεί να υποκατασταθεί. Ο υπολογιστής και το αντίστοιχο εκπαιδευτικό λογισμικό χρησιμοποιούνται (σχεδόν πάντα συμπληρωματικά) κατά τη διάρκεια της εκπαιδευτικής διαδικασίας ή στο εργαστήριο κατά κύριο λόγο, όπου βοηθούν στην εποπτεία φαινομένων μη προσιτών εργαστηριακά και ακόμα και στην ερμηνεία αυτών των φαινομένων.

Μια άλλη δυνατότητα του Η/Υ είναι αυτή της άμεσης επικοινωνίας του με το περιβάλλον και επίδρασής του σε αυτό, η εκμετάλλευση της οποίας συνιστά πρωτοποριακή πρακτική στην εκπαιδευτική μας πραγματικότητα.

Με την εισβολή του Η/Υ στην εκπαίδευση παρουσιάζεται και μια σημαντική ευκαιρία για τη δημιουργία ενός περιβάλλοντος, το οποίο παρέχει δυνατότητες επανεξέτασης και επαναθεώρησης γενικών και ειδικών προβλημάτων στους χώρους μάθησης και εκπαίδευσης, όπως προβλήματα διδακτικού σχεδιασμού, εξατομικευμένης διδασκαλίας και αλληλεπίδρασης ανάμεσα στους συμμετέχοντες στην εκπαιδευτική διαδικασία. Αυτό το περιβάλλον μπορεί να προσφέρει νέους τρόπους απόκτησης της επιστημονικής πληροφορίας, τρόποι οι οποίοι δεν στηρίζονται αποκλειστικά στη χρήση ενός εκπαιδευτικού λογισμικού υλικού, ενώ η απόκτηση αυτής της πληροφορίας θα ενισχύσει δραστηριότητες που ήταν αδύνατο να γίνουν προηγουμένως.

Η διάθεση πληροφοριών ή ακόμα η διάθεση προσβάσεων στην πληροφορία αντιμετωπίζεται ως η πιο ενδιαφέρουσα πλευρά της εκπαίδευσης. Η επικοινωνία, για παράδειγμα, ανάμεσα σε τάξεις διαφορετικών χωρών μέσω ενός πολύ απλού δικτύου επικοινωνίας, προσφέρει μοναδικές εμπειρίες σε μια ευρεία κλίμακα θεμάτων διάφορων επιστημών. Το δίκτυο επικοινωνίας μεταξύ των σχολείων-πανεπιστημίων μπορεί να είναι επίσης συνδεδεμένο και με το χώρο του επαγγελματικού λογισμικού υλικού, αξιοποιώντας έτσι την ευελιξία και την ποιότητα του, αλλά και τον εμπλουτισμό του σχολείου-πανεπιστημίου από το χώρο εργασίας.

Στις σημερινές κοινωνίες θεωρείται κάποιος αναλφάβητος αν δεν γνωρίζει να χρησιμοποιεί τον Η/Υ κατά κάποιο τρόπο. Άρα, ο Η/Υ θα πρέπει να διδάσκεται στο μαθητή ως γνωστικό αντικείμενο που θα τον προετοιμάζει για την εκπαιδευτική και επαγγελματική ανάπτυξη και επιβίωση του. Επίσης, η χρήση του υπολογιστή συνδέεται με ένα σύνολο δεξιοτήτων οι οποίες θα ειναι αναγκαίες για τις καθημερινές ανάγκες του σύγχρονου ανθρώπου: από την απλή αλληλογραφία και σύνταξη κειμένων μέχρι τις πιο εξειδικευμένες απαιτήσεις της εργασίας σε διάφορους επαγγελματικούς χώρους, καθώς και στην επιστημονική έρευνα.

Μία από τις σημαντικότερες λειτουργίες του Η/Υ είναι ότι λειτουργεί σαν πηγή πληροφόρησης. Ο Η/Υ κατέχει τεράστιες δυνατότητες υπολογισμού, απομνημόνευσης, απεικόνισης και συσχετισμού των πληροφοριών. Η δημιουργία δικτύων πληροφόρησης που διαθέτει με τη βοήθεια των τηλεπικοινωνιακών μέσων, τον καθιστά μια σπουδαία πηγή πληροφόρησης, γεγονός πολύ σημαντικό στην εποχή μας εφόσον αυτή χαρακτηρίζεται εποχή της πληροφορίας και η πρόσβαση στην πληροφορία είναι δείκτης δημοκρατίας για μια σύγχρονη χώρα.

Ένα εξίσου σημαντικό στοιχείο του υπολογιστή είναι ότι μπορεί να ανταποκριθεί στις ανάγκες ατόμων διαφορετικής ιδιοσυγκρασίας. Άτομα, δηλαδή, με διαφορετικές προσωπικότητες και ενδιαφέροντα μπορούν να αξιοποιήσουν στο μέγιστο βαθμό τις ικανότητές τους μέσω του υπολογιστή. Για παράδειγμα, μαθητές (στην περίπτωση μας) οι οποίοι δεν είναι αρκετά εξικοιωμένοι με την τεχνολογία και κλίνουν σε ένα πιο θεωρητικό υπόβαθρο μπορούν μέσω των απεριόριστων δυνατοτήτων των υπολογιστών να συνδέονται με βιβλιοθήκες και να έχουν πρόσβαση σε οποιοδήποτε κείμενο-βιβλίο επιθυμούν για να ικανοποιήσουν τα ενδιαφέροντά τους.

Πρέπει να επισημάνουμε το γεγονός ότι ο Η/Υ δεν είναι σημαντικός μόνο γι' αυτό που κάνει, αλλά και για το πως κάνει κάποιον να αισθάνεται. Ο υπολογιστής ικανοποιεί και την ανάγκη για αναζήτηση ταυτότητας. Για παράδειγμα, οι μαθητές που ξεκινούν να προγραμματίζουν αισθάνονται ότι δημιουργούν, ότι φτιάχνουν κάτι δικό τους, οι hackers ή οι πιο έμπειροι γύρω από τον υπολογιστή, που μπορεί να μην έχουν άλλες ακαδημαϊκές επιδόσεις, χρησιμοποιούν την ικανότητα και την κυριαρχία τους πάνω στη μηχανή για να δείξουν την αξία τους, σε αυτόν τον τομέα τουλάχιστον.

Το παραπάνω παράδειγμα μας δίνει την ευκαιρία να αναφέρουμε μια άλλη διάσταση της πληροφορικής στην εκπαίδευση, η οποία είναι ο ρόλος του υπολογιστή ως «*διανοητικό εργαλείο*» (*Mindtool*).

Διανοητικά εργαλεία ονομάζουμε επιμέρους μοντέλα γνωστικών δομών που μπορεί να γενικευθούν και μέσω των οποίων η ανακαλυπτική μάθηση μπορεί να διευκολυνθεί και να παραχθεί. Ένα διανοητικό εργαλείο, για να είναι αποτελεσματικό, πρέπει να μεσολαβεί μεταξύ του θεσμικού και πολιτισμικού πλαισίου από τη μία μεριά, και της τεχνολογίας από την άλλη, μέσω της οποίας δομήθηκε, προσφέροντας έτσι, ένα φυσικό τρόπο προσέγγισης στο χρήστη.

Η εισβολή της πληροφορικής στην εκπαίδευση είναι μια αναμφισβήτητη πραγματικότητα και μάταια μπορεί καποιος να το αρνηθεί. Και δεν εννοείται απλά η εισαγωγή ενός νέου εργαλείου στην εκπαιδευτική διαδικασία αλλά την ανάπτυξη μιας νέας διάστασης αυτής. Με την αλληλεπίδραση γονέων, εκπαιδευτικών, μαθητών, βιβλίων, κοινωνικών ομάδων και του παιδαγωγικού πλαισίου, το εκπαιδευτικό σύστημα θα αλλάξει τις επόμενες δεκαετίες και ο πηρύνας αυτής της αλλαγής θα είναι ο ηλεκτρονικός υπολογιστής.

1.3 Ηλεκτρονικός Υπολογιστής & Συνεργατική Μάθηση

Η σύγχρονη κοινωνία ξεχωρίζει για την πολυπλοκότητά του τρόπου ζωής και των προβλημάτων με τα οποία έρχεται αντιμέτωπος καθημερινά ο άνθρωπος. Είναι η εποχή της έκρηξης των γνώσεων και της ραγδαίας ανάπτυξης της τεχνολογίας, με ξεχωριστή θέση της τεχνολογίας των υπολογιστών. Η δειλή στην αρχή εμφάνιση των ηλεκτρονικών υπολογιστών και στη συνέχεια η εισβολή τους σχεδόν σε κάθε πτυχή της ζωής μας συμβάλλει στην αστραπιαία διακίνηση των γνώσεων και των ιδεών, καθώς και στην αντιμετώπιση πολλών πολύπλοκων καταστάσεων.

Παρ'όλα αυτά, είναι ταυτόχρονα αποδεκτό πως πολλά από τα προβήματα με τα οποία έρχεται αντιμέτωπος ο σύγχρονος άνθρωπος θα μπορούσαν να επιλυθούν πιο εύκολα και πιο αποτελεσματικά με τη συνεργασία και την αλληλοβοήθεια, αφού ζούμε σε μια κοινωνία αλληλεξάρτησης.

Έτσι, γίνεται όλο και πιο δημοφιλής η χρήση της συνεργατικής μάθησης σαν διδακτική πρακτική και εδραιώνεται όλο και περισσότερο η άποψη ότι η συνεργατική μάθηση είναι μια διδακτική προσέγγιση που υπερτερεί έναντι των άλλων – αφού σε αυτή τα πλεονεκτήματα είναι πολύ περισσότερα από τα τυχόν μειονεκτήματά της.

Κατά καιρούς έχουν δοθεί πολλοί διαφορετικοί ορισμοί για τη συνεργατική μάθηση. Σύμφωνα με τους Johnson and Holubec (1990) συνεργατική μάθηση είναι η οργάνωση της τάξης σε μικρές ομάδες με σκοπό τη δημιουργική συνεργασία των μαθητών για μεγιστοποίηση της δικής τους μάθησης, αλλά και της μάθησης των άλλων μελών μέσα στην ομάδα. Μέσω των συνεργατικών δραστηριοτήτων που τους ανατίθενται από τον εκπαιδευτικό, τα μέλη της ομάδας επιδιώκουν αποτελέσματα τα οποία είναι επωφελή για τους ίδιους, αλλά και για τα άλλα μέλη της ομάδας.

Είναι κοινώς αποδεκτό ότι ο ηλεκτρονικός υπολογιστής και η συνεργατική μάθηση μπορούν να συνδυαστούν, αφού η συνεργατική μάθηση είναι μια προσέγγιση που όχι μόνο επιτρέπει την χρήση του, αλλά και που σχεδόν την επιβάλλει, αν ληφθεί υπ'όψιν η αναλογία ηλεκτρονικών υπολογιστών και μαθητών-σπουδαστών.

Οι βασικές αρχές της συνεργατικής μάθησης μπορούν να εφαρμοστούν εύκολα στους Η/Υ. Ένας παράγοντας που δίνει μια άλλη διάσταση στην χρήση των Η/Υ στην εκπαίδευση είναι η ραγδαία ανάπτυξη των δικτύων υπολογιστών (computer networks). Η χρήση αυτής της τεχνολογίας αίρει το πρόβλημα της υποχρεωτικής παρουσίας των συμβαλλομένων στον ίδιο χώρο ή ακόμα και χρόνο. Όλες αυτές οι δυνατότητες του Η/Υ τον κάνουν ένα πολύ δυνατό εργαλείο στα χέρια της εκπαίδευσης, αφού η γενικότητα του επιτρέπει τη χρήση του σε πάρα πολλά αντικείμενα.

Στη συνεργατική μάθηση παύει να υπάρχει ανταγωνιστικότητα μεταξύ των μαθητών, εφόσον δρουν σαν ομάδα με ένα γενικό στόχο, που για να επιτευχθεί απαιτείται αρμονική συνεργασία μεταξύ των μελών της. Τα βασικά συστατικά στοιχεία της συνεργατικής μάθησης είναι τα εξής:

- ***Κοινός στόχος*** : Για να υπάρχει συνεργατική προσπάθεια πρέπει να υπάρχει ο κοινός μαθησιακός στόχος, το ομαδικό αποτέλεσμα.
- ***Αλληλεπίδραση πρόσωπο με πρόσωπο*** : Δεν μπορεί να νοηθεί συνεργασία, αν δεν υπάρχει μια συνεχής αλληλεπίδραση των μελών της ομάδας. Η αλληλεπίδραση εκδηλώνεται ως αμοιβαία βοήθεια, αμοιβαίος επηρεασμός, ενίσχυση και ενθάρρυνση, προσφορά γνώσεων και πληροφοριών, ανταλλαγή υλικού, ανατροφοδότηση μελών της ομάδας (*feedback*) κλπ.
- ***Αλληλεξάρτηση*** : Η έννοια της αλληλεξάρτησης είναι το κλειδί της συνεργατικής μάθησης. Αλληλεξάρτηση υπάρχει όταν η ομάδα για να επιτύχει το έργο της χρειάζεται απαραίτητα και εξασφαλίζει τη συμβολή του κάθε μέλους της. Αλλά και αντίστροφα, κάθε μέλος της ομάδας επιτυγχάνει το στόχο του μόνο αν και τα υπόλοιπα μέλη της ομάδας επιτύχουν τους δικούς τους.
- ***Κοινωνικές δεξιότητες*** : Μέλη της ομάδας που δεν κατέχουν κοινωνικές δεξιότητες είναι δύσκολο να επιτύχουν συνεργασία υψηλής ποιότητας. Η χαμηλή ένταση της φωνής, η άσκηση ηγεσίας, η αποδοχή της διαφορετικότητας και η έκφραση διαφωνίας είναι μερικές από τις δεξιότητες που πρέπει να διδαχθούν στους εκπαιδευόμενους προκειμένου να επιτευχθεί αποδοτικότερη συνεργασία.
- ***Προσωπική ευθύνη*** : Ο μεγαλύτερος κίνδυνος για να αποτύχει η συνεργατική μάθηση είναι όταν αφεθεί ένα μέλος να κυριαρχήσει στην ομάδα και να επιβάλλει τη γνώμη του, τις λύσεις και τις απαντήσεις σε ένα τρέχον ζήτημα. Στην περίπτωση αυτά τα υπόλοιπα μέλη όχι μόνο δεν επωφελούνται αλλά αδρανοποιούνται και οπισθοδρομούν. Χρειάζεται ιδιαίτερη μέριμνα ώστε κάθε μέλος να καθίστανται προσωπικά υπεύθυνο για την επιτυχία της ομάδας. Αυτό επιτυγχάνεται αν εξασφαλιστεί η θετική αλληλεξάρτηση, όπως αναφέρθηκε παραπάνω.

Η θετική αλληλεξάρτηση είναι ο πιο σημαντικός παράγοντας για τη δημιουργία αποδοτικών συνθηκών συνεργασίας.

Όσον αφορά την αξιολόγηση, η οποία αποτελεί το τελικό στάδιο της συνεργατικής μάθησης και ταυτόχρονα εξίσου σημαντικό με τα προαναφερθέντα, οποιαδήποτε κι αν είναι η μορφή της συνεργατικής μάθησης, η αξιολόγηση γίνεται κυρίως μέσα σε κάθε ομάδα με την ανταλλαγή απόψεων των μελών της για το πως μπορεί να γίνει πιο αποδοτική η εργασία τους. Επιπλέον, γίνεται αξιολόγηση της εργασίας της ομάδας από τον επιβλέποντα εκπαιδευτικό, ενώ σε συζήτηση οι μαθητές καταθέτουν τις απόψεις τους και ασκούν κριτική.

Εικόνα 1.2 *Συνεργατική μάθηση: Καθηγητής, φοιτητές και ΗΥ.*

Συνοψίζοντας τα πλεονεκτήματα της συνεργατικής μάθησης παρατηρούμε ότι συμβάλλει στην ανάπτυξη των σχέσεων μεταξύ των μελών της ομάδας, με τη συνεπακόλουθη ανάπτυξη των αισθημάτων αλληλοβοήθειας, εμπιστοσύνης, συνεργασίας (συζήτηση και επίλυση ενός προβλήματος) μεταξύ των μελών της ομάδας. Συμβάλλει επίσης και στην ανάπτυξη του αισθήματος της θετικής αλληλεπίδρασης, αφού η επιτυχία της ομάδας εξαρτάται από την επιτυχία του κάθε μέλους της και αντίστροφα.

Παράλληλα, έχουμε καλύτερη αξιοποίηση του διδακτικού χρόνου, καθώς ελαττώνονται οι ανταγωνιστικές δομές, αφού μεταξύ των εκπαιδευομένων μειώνεται ο ανταγωνισμός και ενισχύεται η συνεργασία. Επίσης, έχουμε κάποιο είδος ανεξαρτησίας από τον εκπαιδευτικό, αφού οι ομάδες έχουν υψηλότερο βαθμό αυτονομίας. Ελευθερώνεται με αυτό τον τρόπο ο εκπαιδευτικός, ο οποίος μπορεί να διαθέσει τον χρόνο του πιο εποικοδομητικά.

Δύο άλλα στοιχεία που αναπτύσσουν τα μέλη της ομάδας είναι η προσωπική ευθύνη, αφού ο εκπαιδευόμενος επιδιώκει να μάθει ο ίδιος και ελέγχει αν τα υπόλοιπα μέλη μαθαίνουν, και η αυτοπεποίθηση, αφού τώρα πια ο εκπαιδευόμενος είναι ενεργό μέλος μιας ομάδας με άποψη και προσφορά για την επίτευξη του κοινού στόχου.

Όλα τα μέλη, λόγω ηθελημένης ανομοιογένειας, μπορούν να αφομοιώσουν σκέψεις, λύσεις και προτάσεις από τα υπόλοιπα μέλη της ομάδας, κάτι που βοηθάει στην μάθηση και την κατάκτηση της γνώσης πιο γρήγορα.

Μελετώντας και αναλύοντας την συνεργατική μάθηση αντιλαμβάνεται κανείς πόσο πολύ ταιριάζει σε αυτήν η χρήση του ηλεκτρονικού υπολογιστή. Οι πολλές δυνατότητες του ΗΥ και ειδικά η δυνατότητά του να συγκεντρώνει, να επεξεργάζεται, να ταξινομεί, να παρουσιάζει και να διακινεί πληροφορίες, τον καθιστούν μοναδικό εργαλείο στα χέρια του εκπαιδευτικού. Ύστερα από έρευνες που έχουν πραγματοποιηθεί, έχει παρατηρηθεί ότι η υποβοηθούμενη από Η/Υ συνεργατική μάθηση προάγει υψηλότερη ποσότητα και ποιότητα καθημερινής επίτευξης, μεγαλύτερη ικανότητα τεκμηριωμένης μάθησης καθώς και καλύτερη ικανότητα να χρησιμοποιούν οι εκπαιδευόμενοι τις γνώσεις τους στην επίλυση των προβλημάτων. Επίσης, οι συνεργατικές ομάδες παρουσιάστηκαν πιο γρήγορες και ακριβείς από τις ατομικιστικές και ανταγωνιστικές ομάδες.

Αναφέροντας συγκεκριμένους τομείς της εκπαίδευσης, μπορούμε να πούμε ότι η δημιουργία βάσεων δεδομένων για συγκεκριμένα ερευνητικά θέματα είναι ένας τομέας στον οποίο πραγματικά μπορεί να συμβάλλει αποτελεσματικά ο Η/Υ – ένας τομέας που θα μας απασχολήσει ιδιαίτερα στα πλαίσια αυτής της εργασίας και θα αναφερθούμε εκτενέστερα – αφού μας διευκολύνει στην ετοιμασία της με τα κατάλληλα λογισμικά βάσεων δεδομένων. Τα κατάλληλα προβλήματα ενθαρρύνουν τους μαθητές να συνεργάζονται και να τα αναλύουν στα συστατικά τους, να εντοπίζουν τα δεδομένα που χρειάζονται, να τα εξάγουν από τη βάση δεδομένων, να τα ταξινομούν με κάποιο χρήσιμο τρόπο, να χρησιμοποιούν αυτά τα δεδομένα και να παρουσιάζουν τα ευρήματά τους στους άλλους.

Στο μάζεμα πληροφοριών και δεδομένων για τα διάφορα projects ή εργασίες που έχουν οι εκπαιδευόμενοι (φοιτητές στη συγκεκριμένη περίπτωση), μπορούν να συμβάλλουν πολύ αποτελεσματικά και τα διάφορα δίκτυα, είτε τα τοπικά είτε και το internet. Η επικοινωνία των φοιτητών των διάφορων πανεπιστημίων που είναι συνδεδεμένα μέσω δικτύου συμβάλλει στην καλλιέργεια συνεργατικών δεξιοτήτων. Ανταλλάσσουν πληροφορίες, συζητούν για θέματα αμοιβαίου ενδιαφέροντος, βρίσκουν λύσεις σε κοινά προβλήματα, αλληλοενημερώνοντα για τρέχοντα θέματα κ.ο.κ. Μια άλλη διάσταση που δίνουν τα δίκτυα είναι ότι δεν είναι απαραίτητη η παρουσία των συμβαλλομένων στον ίδιο χρόνο αλλά η συνεργασία μπορεί να γίνει όχι μόνο σε πραγματικό χρόνο (real time), αλλά και ασύγχρονα με απλή χρήση του email ή και με τοποθέτηση δεδομένων σε διάφορες ιστοσελίδες επιτρέποντας στους χρήστες να δουλεύουν με το δικό τους ρυθμό και χρόνο. Η χρήση του internet συμβάλλει στο σπάσιμο των φραγμών που επιβάλλουν τα σύνορα, καθώς και στην αλληλοκατανόηση, την αποδοχή της διαφορετικότητας, αφού φέρνει σε επαφή άτομα με διαφορετικό πολιτισμό και κουλτούρα.

ΚΕΦΑΛΑΙΟ 2

Βάσεις Δεδομένων

2.1 Εισαγωγή

Η αλματώδης ανάπτυξη της επιστήμης της πληροφορικής και των επικοινωνιών τα τελευταία χρόνια έχει καταστήσει την πληροφορία ως ένα από τα βασικότερα και πολυτιμότερα αγαθά. Είναι κοινός τόπος σήμερα η εκτίμηση ότι το αγαθό της πληροφορίας είναι επιθυμητό απ' όλους τους εργαζόμενους αλλά και τους εκπαιδευόμενους, ώστε να είναι πιο αποδοτικοί, ανταγωνιστικοί αλλά και παραγωγικοί στην εργασία τους.

Οι βάσεις δεδομένων και η τεχνολογία βάσεων δεδομένων εξασκούν σημαντική επίδραση στην αυξανόμενη χρήση των υπολογιστών. Είναι εύλογο να ειπωθεί ότι οι βάσεις δεδομένων διαδραματίζουν ένα σημαντικό ρόλο σε όλες τις περιοχές όπου χρησιμοποιούνται υπολογιστές, συμπεριλαμβανομένων των επιχειρήσεων, του ηλεκτρονικού εμπορίου, της μηχανικής, της ιατρικής, της νομικής, της εκπαίδευσης και της βιβλιοθηκονομίας, για να αναφέρουμε μερικές μόνο από αυτές.

Τα συστήματα βάσεων δεδομένων τα χρησιμοποιούμε για να μπορούμε να αποθηκεύσουμε, να επεξεργαστούμε αλλά και να εκμεταλλευτούμε αποδοτικά αυτόν τον τεράστιο όγκο των πληροφοριών που αυξάνονται με αλματώδεις ρυθμούς καθημερινά.

2.2 Δεδομένα και Πληροφορία

Με τον όρο *πληροφορία* αναφερόμαστε συνήθως σε ειδήσεις, γεγονότα και έννοιες που αποκτάμε από την καθημερινή μας επικοινωνία και τα θεωρούμε ως αποκτηθείσα γνώση, ενώ τα δεδομένα μπορούν να είναι μη κατάλληλα επεξεργασμένα και μη ταξινομημένα σύνολα πληροφοριών. Ένας αυστηρός ορισμός για το τι είναι δεδομένα και τι είναι πληροφορία, σύμφωνα με την επιτροπή ANSI των ΗΠΑ, είναι ο εξής :

- ***Δεδομένα*** (data) είναι μια παράσταση, όπως γράμματα, αριθμοί, σύμβολα κ.ά. στα οποία μπορούμε να δώσουμε κάποια σημασία (έννοια).
- ***Πληροφορία*** (information) είναι η σημασία που δίνουμε σ' ένα σύνολο από δεδομένα, τα οποία μπορούμε να επεξεργαστούμε βάσει προκαθορισμένων κανόνων και να βγάλουμε έτσι κάποια χρήσιμα συμπεράσματα. Με τις πληροφορίες περιορίζεται η αβεβαιότητα που έχουμε για διάφορα πράγματα και βοηθιόμαστε, έτσι, στο να λάβουμε σωστές αποφάσεις.

Τα δεδομένα μπορούν να θεωρηθούν ως τρόποι αναπαράστασης εννοιών και γεγονότων που μπορούν να υποστούν διαχείριση και επεξεργασία. Η συλλογή και αποθήκευση ενός τεράστιου όγκου δεδομένων όπως απαιτούν οι κοινωνικές συνθήκες σήμερα, δεν λύνει τελείως το πρόβλημα της σωστής οργάνωσης και ταξινόμησης των δεδομένων. Τα δεδομένα θα πρέπει να οργανωθούν με τέτοιο τρόπο έτσι ώστε να μπορούμε να τα εντοπίζουμε και να τα αξιοποιούμε εύκολα και γρήγορα και τη στιγμή που τα χρειαζόμαστε.

Ένα κλασικό παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν για παράδειγμα ο τηλεφωνικός κατάλογος της πόλης της Αθήνας, όπου οι συνδρομητές δεν θα ήταν καταχωρημένοι αλφαβητικά σύμφωνα με το

επώνυμο και το όνομά τους, αλλά εντελώς τυχαία. Ένας τέτοιος τηλεφωνικός κατάλογος θα περιείχε μια τεράστια ποσότητα δεδομένων αλλά θα ήταν ουσιαστικά άχρηστος.

Ένα άλλο παράδειγμα μη σωστής οργάνωσης δεδομένων θα ήταν μια πολύ μεγάλη βιβλιοθήκη με χιλιάδες τόμους βιβλίων και χωρίς να διαθέτει κάποιο υποτυπώδες σύστημα οργάνωσης και ταξινόμησης των βιβλίων. Ούτε οι υπάλληλοι της βιβλιοθήκης θα μπορούσαν να κάνουν τη δουλειά τους αλλά ούτε και οι επισκέπτες θα μπορούσαν να αξιοποιήσουν την πληθώρα των πληροφοριών που περιέχονται στα βιβλία. Εκτός λοιπόν από τη μόνιμη αποθήκευση των δεδομένων, χρειαζόμαστε και κάποιους τρόπους ευέλικτης και αποδοτικής οργάνωσής τους.

Χαρακτηριστικά παραδείγματα δεδομένων που απαιτούν σωστή και αποδοτική οργάνωση είναι τα εξής :

- Τα στοιχεία υπαλλήλων, πελατών, προμηθευτών και παραγγελιών μιας εμπορικής επιχείρησης.
- Τα στοιχεία υλικών μιας αποθήκης.
- Τα στοιχεία ταινιών, πελατών και δανεισμών μιας βιντεολέσχης.
- Τα στοιχεία υπαλλήλων, γιατρών, ασθενών αλλά και υλικών ενός νοσοκομείου.
- Τα στοιχεία βιβλίων, χρηστών (δανειστών) και δανεισμών μιας βιβλιοθήκης.

2.2.1 Εισαγωγή στις Βάσεις Δεδομένων και τα ΣΔΒΔ και ορισμός

Με βάση το γεγονός ότι η χρήση των ηλεκτρονικών υπολογιστών και συνεπώς η ηλεκτρονική καταχώρηση και επεξεργασία δεδομένων αυξήθηκε κατακόρυφα ήδη από τη δεκαετία του ’70 στις μεγάλες επιχειρήσεις και άρα είχαμε πάρα πολλές εφαρμογές να επεξεργάζονται δεδομένα σε πάρα πολλά αρχεία ταυτόχρονα, προτάθηκε η συνένωση όλων των αρχείων μιας εφαρμογής. Εκτός, όμως, από τη συνένωση των αρχείων, απαιτείται και μια σωστή οργάνωσή τους. Δημιουργήθηκαν έτσι οι Τράπεζες Πληροφοριών ή Βάσεις Δεδομένων (Data Bases).

Μια ***Βάση Δεδομένων*** *(ΒΔ)* είναι ένα σύνολο αρχείων με υψηλό βαθμό οργάνωσης τα οποία είναι συνδεδεμένα μεταξύ τους με λογικές σχέσεις, έτσι ώστε να μπορούν να χρησιμοποιούνται από πολλές εφαρμογές και από πολλούς χρήστες ταυτόχρονα. Υπάρχει ένα ειδικό λογισμικό το οποίο μεσολαβεί ανάμεσα στα αρχεία δεδομένων και τις εφαρμογές που χρησιμοποιούν οι χρήστες και αποκαλείται ***Σύστημα Διαχείρισης Βάσης Δεδομένων*** (*ΣΔΒΔ*) ή *DBMS (Data Base Management System).* Το ΣΔΒΔ είναι στην ουσία ένα σύνολο από προγράμματα και υπορουτίνες που έχουν να κάνουν με τον χειρισμό της βάσης δεδομένων, όσον αφορά τη δημιουργία, τροποποίηση, διαγραφή στοιχείων, με ελέγχους ασφαλείας κ.ά.

Ένας τρόπος για να κατηγοριοποιήσει κάποιος τις βάσεις δεδομένων περιλαμβάνει τον τύπο του περιεχομένου τους, για παράδειγμα: βιβλιογραφική, έγγραφο-κείμενο, στατιστική, ή αντικειμένων πολυμέσων. Ένας άλλος τρόπος είναι από την περιοχή εφαρμογής τους, για παράδειγμα: τη λογιστική, την σύνθεση μουσικής, τις ταινίες, την τραπεζική, την κατασκευή ή την ασφάλιση.

Τα δεδομένα που υπάρχουν στις βάσεις δεδομένων πρέπει να είναι :

- ***Ολοκληρωμένα*** *(Integrated)*, δηλ. τα δεδομένα πρέπει να είναι αποθηκευμένα σε ομοιόμορφα οργανωμένα σύνολα αρχείων όπου δεν πρέπει να υπάρχει επανάληψη ή πλεονασμός (redundancy) των ίδιων στοιχείων.
- ***Καταμεριζόμενα*** *(Shared)*, δηλ. να μπορούν περισσότεροι του ενός χρήστες να βλέπουν και να μοιράζονται τα ίδια δεδομένα την ίδια χρονική στιγμή.

Οι στόχοι μιας βάσης δεδομένων είναι οι εξής :

- Ο περιορισμός της πολλαπλής αποθήκευσης των ίδιων στοιχείων (redundancy).
- Ο καταμερισμός (sharing) των ίδιων στοιχείων σ' όλους τους χρήστες.
- Η ομοιομορφία (uniformity) στον χειρισμό και την αναπαράσταση των δεδομένων.
- Η επιβολή κανόνων ασφαλείας (security).
- Η διατήρηση της ακεραιότητας (integrity) και της αξιοπιστίας (reliability) των δεδομένων.
- Η ανεξαρτησία των δεδομένων (data independence) και των προγραμμάτων από τον φυσικό τρόπο αποθήκευσης των δεδομένων.

2.3 Ιστορικά Στοιχεία

2.3.1 Η Γενική Ιδέα της Βάσης Δεδομένων

Η έννοια Βάση Δεδομένων έχει εξελιχθεί από τη δεκαετία του 1960 για να διευκολύνει τις αυξανόμενες δυσκολίες στο σχεδιασμό, την κατασκευή και την συντήρηση πολύπλοκων συστημάτων πληροφορίας (συνήθως με πολλούς ταυτόχρονους τελικούς χρήστες και ένα ευρύ και μεγάλο όγκο δεδομένων). Έχει εξελιχθεί από κοινού με τα Συστήματα Διαχείρησης Βάσεων Δεδομένων (ΣΔΒΔ ή DBMS) που επιτρέπουν την αποτελεσματική διαχείρηση των βάσεων δεδομένων. Αν και οι όροι Βάση Δεδομένων και ΣΔΒΔ καθορίζουν δύο διαφορετικές οντότητες, είναι άρρηκτα συνδεδεμένοι: οι ιδιότητες μιας Βάσης Δεδομένων καθορίζονται από το ΣΔΒΔ που την

υποστηρίζει και το αντίστροφο. Το αγγλικό λεξικό της Οξφόρδης παραθέτει το 1962 μία τεχνική αναφορά ως την πρώτη που χρησιμοποιεί τον όρο «βάση δεδομένων».

Με την πρόοδο της τεχνολογίας στους τομείς των επεξεργαστών, της μνήμης των υπολογιστών, του χώρου αποθήκευσης στους υπολογιστές και των δικτύων υπολογιστών, τα μεγέθη, οι δυνατότητες και η απόδοση των Βάσεων Δεδομένων και των ΣΔΒΔ έχουν αυξηθεί κατά μεγάλες τάξεις μεγέθους. Για δεκαετίες θεωρούταν απίθανο το γεγονός ότι ένα πολύπλοκο σύστημα πληροφοριών μπορεί να χτιστεί αποτελεσματικά χωρίς μία σωστή βάση δεδομένων που να υποστηρίζεται από ένα ΣΔΒΔ. Η αξιοποίηση των βάσεων δεδομένων έχει εξαπλωθεί πολύ μεγάλο βαθμό. Πολλά εμπορικά προιόντα βασίζονται στις Βάσεις Δεδομένων και τα ΣΔΒΔ για την ανάπτυξη και την εμπορευματοποίησή τους Επίσης, οργανισμοί και εταιρείες, από τις πιο μικρές έως τις πιο μεγάλες, εξαρτώνται σε μεγάλο βαθμό από τις βάσεις δεδομένων για την λειτουργία τους.

Δεν υπάρχει ένας ευρέως αποδεκτός ακριβής ορισμός για το ΣΔΒΔ. Ωστόσο, ένα σύστημα πρέπει να παρέχει σημαντικές λειτουργίες για να χαρακτηριστεί ως ένα ΣΔΒΔ. Ανάλογα με την υποστηριζόμενη συλλογή των δεδομένων του, πρέπει να πληρούνται και οι αντίστοιχες απαιτήσεις ευχρηστίας για να χαρακτηριστεί ως μια Βάση Δεδομένων. Έτσι, μια Βάση Δεδομένων και το ΣΔΒΔ που την υποστηρίζει ορίζονται εδώ με μια σειρά από γενικές απαιτήσεις που απαριθμούνται κατωτέρω. Σχεδόν όλα τα υπάρχοντα «ώριμα» προϊόντα ΣΔΒΔ ανταποκρίνονται στις απαιτήσεις αυτές σε μεγάλο βαθμό, ενώ τα λιγότερο «ώριμα», είτε καλύπτουν μερικώς αυτές τις απαιτήσεις, είτε κλίνουν προς αυτές.

2.3.2 Εξέλιξη των Βάσεων Δεδομένων και των ΣΔΒΔ

- ***Βάσεις Δεδομένων***

Η εισαγωγή του όρου «Βάση Δεδομένων» συνέπεσε με τη διαθεσιμότητα της άμεσης αποθήκευσης σε κατάλληλα μέσα (σκληρούς δίσκους και δισκέτες) από τα μέσα της δεκαετίας του 1960 και μετά. Ο όρος αντιπροσώπευε μία αντίθεση με τα taped-based συστήματα του παρελθόντος, επιτρέποντας την κοινή διαδραστική χρήση παρά την καθημερινή μαζική επεξεργασία.

Στα πρώτα συστήματα βάσεων δεδομένων, η αποτελεσματικότητα ήταν ίσως το κυριότερο μέλημα, αλλά είχε ήδη αναγνωριστεί ότι υπήρχαν κι άλλοι στόχοι. Ένας από τους βασικούς στόχους ήταν να γίνουν τα δεδομένα ανεξάρτητα από τη λογική της εφαρμογής αποκλειστικά σε κάποια

προγράμματα, έτσι ώστε να μπορούν να διατεθούν σε διαφορετικές εφαρμογές.

Η πρώτη γενιά των συστημάτων βάσεων δεδομένων ήταν *βάσεις πλοήγησης* (navigational), όπου οι εφαρμογές συνήθως είχαν πρόσβαση σε δεδομένα ακολουθώντας δείκτες από μία εγγραφή σε μία άλλη. Τα δύο κύρια μοντέλα αυτή την εποχή ήταν το *ιεραρχικό μοντέλο* (*Hierarchical Model*) που αποτελεί την επιτομή του συστήματος IMS της IBM και το *Codesyl μοντέλο* (*μοντέλο δικτύου – Network Model)*, που εφαρμόζονται σε ορισμένα προϊόντα, όπως το IDMS.

Το *Σχεσιακό Μοντέλο – Relational Model*, προτάθηκε για πρώτη φορά το 1970 από τον Edgar F. Codd, ο οποίος επέμενε ότι οι αιτήσεις (ενός χρήστη) θα πρέπει να αναζητούν δεδομένα με βάση το περιεχόμενο και όχι ακολουθώντας συνδέσμους. Αυτό θεωρήθηκε αναγκαίο, για να επιτραπεί στο περιεχόμενο μίας βάσης δεδομενων να εξελιχθεί χωρίς τη συνεχή αναμόρφωση των εφαρμογών. Τα σχεσιακά συστήματα έθεσαν σοβαρές απαιτήσεις όσον αφορά τους πόρους επεξεργασίας, και δεν ήταν μέχρι τα μέσα της δεκαετίας του 1980 όπου το υλικό των υπολογιστών έγινε αρκετά δυνατό έτσι ώστε να μπορούν να χρησιμοποιηθούν ευρέως. Από τις αρχές του 1990, ωστόσο, τα σχεσιακά συστήματα ήταν κυρίαρχα σε όλες τις εφαρμογές μεγάλης κλίμακας επεξεργασίας στοιχείων και παραμένουν κυρίαρχα μέχρι και σήμερα, εκτός από εξειδικευμένες περιοχές. Η κυρίαρχη γλώσσα των Βάσεων Δεδομένων είναι το πρότυπο *SQL* (*Structured Query Language*) ή *Δομημένη Γλώσσα Ερωτημάτων* για το Σχεσιακό μοντέλο, το οποίο έχει επηρεάσει γλώσσες βάσεων δεδομένων και για άλλα μοντέλα δεδομένων.

Επειδή το Σχεσιακό μοντέλο δίνει έμφαση στην αναζήτηση παρά στην πλοήγηση, δεν παριστάνει τις σχέσεις μεταξύ διαφόρων οντοτήτων αποκλειστικά με τη μορφή δεικτών, αλλά τους αντιπροσωπεύει αντί να χρησιμοποιεί *πρωτεύοντα (primary)* και *ξένα (foreign)* κλειδιά. Ενώ αυτό είναι μία καλή βάση για μία γλώσσα ερωταπαντήσεων, είναι λιγότερο κατάλληλο ως γλώσσα μοντελοποίησης. Για το λόγο αυτό, ένα διαφορετικό μοντέλο, το μοντέλο *Οντοτήτων-Συσχετίσεων* (*Entity-Relationship model*) που αναπτύχθηκε λίγο αργότερα (1976), κέρδισε τη δημοτικότητα για το σχεδιασμό βάσεων δεδομένων.

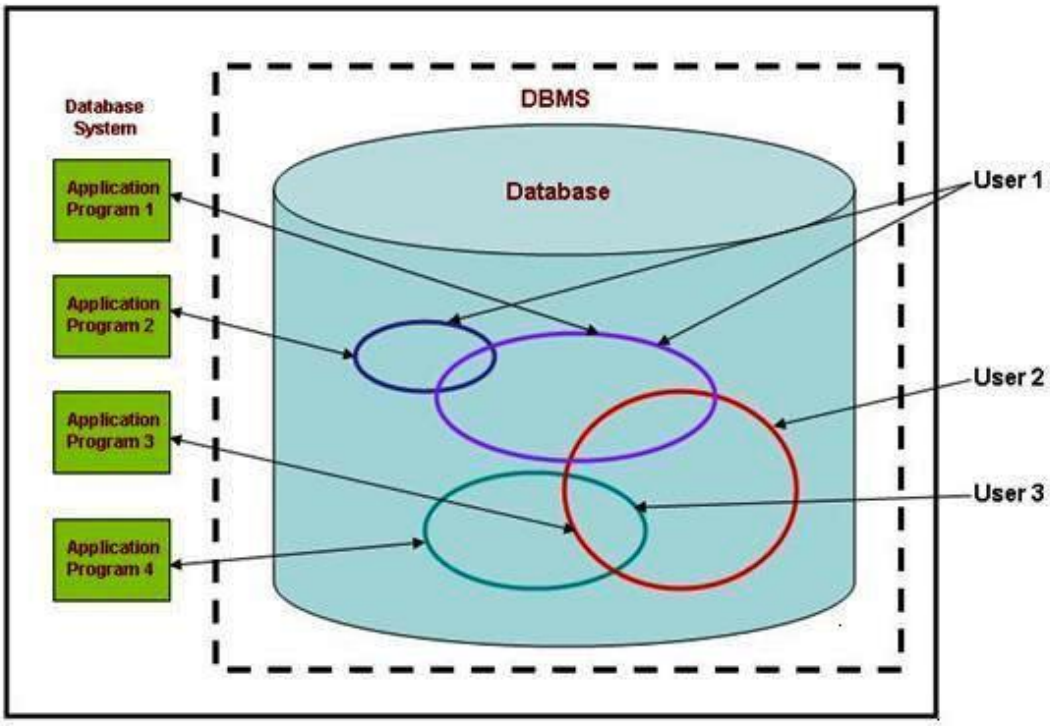
Από το 1970 και μετά, η τεχνολογία βάσεων δεδομένων έχει συμβαδίσει με τους αυξανόμενους πόρους που θα καταστούν διαθέσιμοι μέσω της τεχνολογίας των υπολογιστών και συγκεκριμένα μέσω της ταχείας αύξησης της χωρητικότητας και της ταχύτητας (με ταυτόχρονη μείωση των τιμών) των μέσων αποθήκευσης και της αύξησης της χωρητικότητας της κύριας μνήμης.

Η ακαμψία του Σχεσιακού μοντέλου, στο οποίο όλα τα δεδομένα αποθηκεύονται σε πίνακες με συγκεκριμένη δομή γραμμών και στηλών, όλο και περισσότερο θεωρείται ως περιορισμός κατά τον χειρισμό πληροφοριών, η οποία είναι πιο πλούσια ή ποικίλει στην δομή από το παραδοσιακό «καθολικό βιβλίο» δεδομένων των εταιρικών πληροφοριακών συστημάτων : για παράδειγμα, βάσεις δεδομένων εγγράφου, βάσεις δεδομένων μηχανικής, βάσεις δεδομένων πολυμέσων ή βάσεις δεδομένων που χρησιμοποιούνται στις μοριακές επιστήμες. Διάφορες προσπάθειες έχουν γίνει για να αντιμετωπιστεί αυτό το πρόβλημα, πολλές από αυτές οδηγούν σε λύσεις όπως οι «*μετά-σχεσιακές*» ή η *NoSQL*. Δύο εξελίξεις που αξίζει να σημειωθούν είναι η *Βάση Δεδομένων Αντικειμένων – Object Database* και η *Βάση Δεδομένων XML – eXtensible Markup Language*. Οι προμηθευτές-πωλητές των σχεσιακών βάσεων δεδομένων έχουν πολεμήσει τον ανταγωνισμό από τα νεότερα μοντέλα, επεκτείνοντας τις δυνατότητες των προϊόντων τους ώστε να υποστηρίξουν μια ερύτερη ποικιλία δεδομένων.

➢ *ΣΔΒΔ*

Ένα ΣΔΒΔ έχει εξελχθεί σε ένα πολύπλοκο σύστημα λογισμικού και η ανάπτυξή του απαιτεί συνήθως χιλιάδες ανθρωπο-έτη αναπτυξιακής προσπάθειας. Ορισμένα γενικού-σκοπού ΣΔΒΔ, όπως η Oracle, η Microsoft SQL Server και η IBM DB2 έχουν υποστεί αναβαθμίσεις για τριάντα χρόνια ή και περισσότερο. Τα γενικού-σκοπού ΣΔΒΔ έχουν στόχο να ικανοποιήσουν όσο το δυνατόν περισσότερες εφαρμογές, που συνήθως τις καθιστούν ακόμη πιο πολύπλοκες απ' ότι οι ειδικού σκοπού βάσεις δεδομένων. Ωστόσο, το γεγονός ότι μπορούν να χρησιμοποιηθούν «από το ράφι», καθώς και το αναπόσβεστο κόστος τους σε πολλές εφαρμογές και περιπτώσεις, τα κάνει μια ελκυστική εναλλακτική λύση κάθε φορά που πληρούν τις απαιτήσεις μιας εφαρμογής.

Αν και ελκυστικό σε πολλές περιπτώσεις, ένα γενικού-σκοπού ΣΔΒΔ δεν είναι πάντα η καλύτερη λύση, δηλαδή, όταν ορισμένες εφαρμογές είναι ευρέως διαδεδομένες, με πολλές δυνατότητες λειτουργίας και η καθεμία με πολλούς χρήστες, ένα γενικού-σκοπού ΣΔΒΔ μπορεί να εισάγει περιττή επιβάρυνση (overhead) και πολύ μεγάλο ποσό περιττού και άχρηστου κώδικα λογισμικού. Οι εφαρμογές αυτές δικαιολογούν συνήθως αποκλειστική ανάπτυξη. Τυπικά παραδείγματα είναι τα συστήματα ηλεκτρονικού ταχυδρομείου (e-mail systems), τα οποία πρέπει να διαθέτουν ορισμένες ιδιότητες ενός ΣΔΒΔ : τα συστήματα ηλεκτρονικού ταχυδρομείου είναι χτισμένα με τρόπο που βελτιστοποιούν τη διαχείρηση μηνυμάτων ηλεκτρονικού ταχυδρομείου και δεν χρειάζονται σημαντικά μέρη της λειτουργικότητας ενός γενικού-σκοπού ΣΔΒΔ.



Εικόνα 2.1 *Σχεδιάγραμμα ΣΔΒΔ και Βάσης Δεδομένων.*

2.4 Δομή μιας Βάσης Δεδομένων

2.4.1 Σύνοψη

Ο σχεδιασμός, η κατασκευή και η συντήρηση μιας σύνθετης βάσης δεδομένων απαιτεί ειδικές δεξιότητες: υπάρχει εξιδικευμένο προσωπικό που εκτελεί αυτές τις λειτουργίες και αναφέρονται ως προγραμματιστές εφαρμογών βάσεων δεδομένων και διαχειριστές βάσεων δεδομένων. Τα καθήκοντά τους υποστηρίζονται από εργαλεία που παρέχονται, είτε ως μέρος του ΣΔΒΔ ή ως αυτόνομα προϊόντα λογισμικού. Τα εργαλεία αυτά περιλαμβάνουν εξειδικευμένες γλώσσες βάσεων δεδομένων που περιλαμβάνουν γλώσσες ορισμού δεδομένων (*data definition languages – DDL*), γλώσσες χειρισμού δεδομένων (*data manipulation languages – DML)*, και γλώσσες ερωταπαντήσεων (*query languages*). Αυτές μπορούν να θεωρηθούν ως ειδικού σκοπού γλώσσες προγραμματισμού, ειδικά προσαρμοσμένες για να διχειριστούν τις βάσεις δεδομένων. Μερικές φορές παρέχονται ως επεκτάσεις των υφιστάμενων γλωσσών προγραμματισμού, με προσθήκη εντολών της βάσης δεδομένων. Οι γλώσσες των βάσεων δεδομένων είναι συγκεκριμένες για ένα μοντέλο δεδομένων, και σε πολλές περιπτώσεις για έναν συγκεκριμένο τύπο ΣΔΒΔ. Η πιο ευρέως

υποστηριζόμενη γλώσσα βάσης δεδομένων είναι η SQL, η οποία έχει αναπτυχθεί για το σχεσιακό μοντέλο δεδομένων και συνδυάζει τους ρόλους της ως DDL, DML και ως γλώσσα ερωταπαντήσεων.

Οι χρήστες των εφαρμογών αντλούν τα στοιχεία που τους ενδιαφέρουν από τη βάση δεδομένων χωρίς να είναι σε θέση να γνωρίζουν με ποιο τρόπο είναι οργανωμένα τα δεδομένα σ' αυτήν. Το ΣΔΒΔ παίζει τον ρόλο του μεσάζοντα ανάμεσα στον χρήστη και τη βάση δεδομένων και μόνο μέσω του ΣΔΒΔ μπορεί ο χρήστης να αντλήσει πληροφορίες από τη βάση δεδομένων. Ένα ΣΔΒΔ μπορεί να είναι εγκατεστημένο σ' έναν μόνο υπολογιστή ή και σ' ένα δίκτυο υπολογιστών και μπορεί να χρησιμοποιείται από έναν χρήστη ή και από πολλούς χρήστες.

2.4.2 Σύστημα Βάσης Δεδομένων

Ένα Σύστημα Βάσης Δεδομένων (ΣΒΔ) ή DBS (Data Base System) αποτελείται από το υλικό, το λογισμικό, τη βάση δεδομένων και τους χρήστες. Είναι δηλαδή ένα σύστημα με το οποίο μπορούμε να αποθηκεύσουμε και να αξιοποιήσουμε δεδομένα με τη βοήθεια ηλεκτρονικού υπολογιστή. Αναλυτικά :

- Το ***υλικό*** *(hardware)* αποτελείται όπως είναι γνωστό από τους ηλεκτρονικούς υπολογιστές, τα περιφερειακά, τους σκληρούς δίσκους, τις μαγνητικές ταινίες κ.ά., όπου είναι αποθηκευμένα τα αρχεία της βάσης δεδομένων αλλά και τα προγράμματα που χρησιμοποιούνται για την επεξεργασία τους.
- Το ***λογισμικό*** *(software)* είναι τα προγράμματα που χρησιμοποιούνται για την επεξεργασία των δεδομένων (στοιχείων) της βάσης δεδομένων.
- Η ***βάση δεδομένων*** *(database)* αποτελείται από το σύνολο των αρχείων όπου είναι αποθηκευμένα τα δεδομένα του συστήματος. Τα στοιχεία αυτά μπορεί να βρίσκονται αποθηκευμένα σ' έναν φυσικό υπολογιστή αλλά και σε περισσότερους. Όμως, στον χρήστη δίνεται η εντύπωση ότι βρίσκονται συγκεντρωμένα στον ίδιο υπολογιστή. Τα δεδομένα των αρχείων αυτών είναι ενοποιημένα (data integration), δηλαδή, δεν υπάρχει πλεονασμός (άσκοπη επανάληψη) δεδομένων και διαμερισμένα (data sharing), δηλαδή, υπάρχει δυνατότητα ταυτόχρονης προσπέλασης των δεδομένων από πολλούς χρήστες. Ο κάθε χρήστης έχει διαφορετικά δικαιώματα και βλέπει διαφορετικό κομμάτι της βάσης δεδομένων, ανάλογα με τον σκοπό για τον οποίο συνδέεται.
- Οι **χρήστες (users)** μιας βάσης δεδομένων χωρίζονται στις εξής κατηγορίες :

- ***Προγραμματιστές του ΣΔΒΔ*** *(DBMS developers)*. Αυτοί είναι οι άνθρωποι, οι οποίοι σχεδιάζουν και αναπτύσσουν το ΣΔΒΔ και οι μόνοι που θα «αγγίξουν» τον κώδικά του. Είναι συνήθως οι εργαζόμενοι ενός ΣΔΒΔ προμηθευτή (π.χ. η Oracle, η IBM, η Microsoft, η Sybase), ή στην περίπτωση των open-source ΣΔΒΔ (π.χ. η MySQL) εθελοντές ή άτομα που υποστηρίζονται από τις ενδιαφερόμενες επιχειρήσεις και οργανισμούς. Τα άτομα αυτά έχουν τυπικά προσόντα προγραμματιστών συστημάτων. Η ανάπτυξη ενός ΣΔΒΔ είναι πολύπλοκη και δύσκολη εργασία και μερικά από τα πιο δημοφιλή ΣΔΒΔ είναι υπό συνεχή ανάπτυξη και βελτίωση (ταυτόχρονα με την παρακολούθηση της προόδου της τεχνολογίας) για δεκαετίες.
- ***Διαχειριστές Δεδομένων*** *(Data Administrator)*. Έχουν τη διοικητική αρμοδιότητα και ευθύνη για την οργάνωση της βάσης δεδομένων και την απόδοση δικαιωμάτων πρόσβασης στους χρήστες.
- ***Προγραμματιστές εφαρμογών*** *(Applications developers)* *και* ***Διαχειριστές Βάσεων Δεδομένων*** *(DataBase Administrators)*. Οι πρώτοι αναπτύσσουν τις εφαρμογές του ΣΒΔ σε κάποια από τις γνωστές γλώσσες προγραμματισμού. Οι δεύτεροι σχεδιάζουν την απαιτούμενη βάση δεδομένων, λαμβάνουν οδηγίες από τον διαχειριστή δεδομένων και είναι αυτοί που διαθέτουν τις τεχνικές γνώσεις και αρμοδιότητες για τη σωστή και αποδοτική λειτουργία του ΣΔΒΔ. Και οι δύο είναι εξοικειωμένοι με το προϊόν και χρησιμοποιούν το interface του (καθώς και άλλα εργαλεία συνήθως) για το έργο τους.
- ***Τελικοί χρήστες*** *(end users)*. Χρησιμοποιούν κάποια εφαρμογή για να παίρνουν στοιχεία από μια βάση δεδομένων, έχουν τις λιγότερες δυνατότητες επέμβασης στα στοιχεία της βάσης δεδομένων, χρησιμοποιούν ειδικούς κωδικούς πρόσβασης και το σύστημα τους επιτρέπει ανάλογη πρόσβαση σε συγκεκριμένο κομμάτι της βάσης δεδομένων.

2.5 Λειτουργικές περιοχές των Βάσεων Δεδομένων

2.5.1 Η Αρχιτεκτονική των ΣΔΒΔ

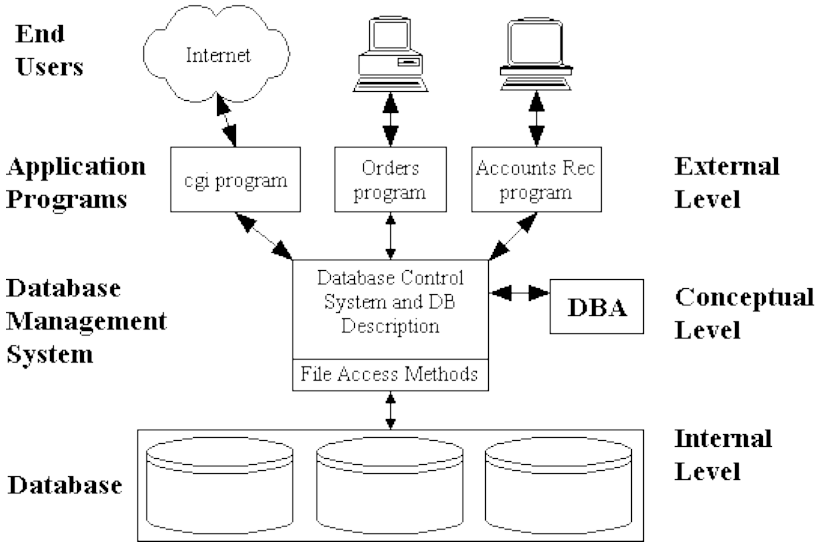
Όπως είδαμε νωρίτερα, ένα ΣΔΒΔ (Σύστημα Διαχείρισης Βάσης Δεδομένων) έχει σαν αποστολή τη διαχείριση των δεδομένων των αρχείων της βάσης, δηλ. την προσθήκη, διαγραφή, τροποποίηση εγγραφών, την

αναζήτηση μέσα στις εγγραφές κ.ά.). Το ΣΔΒΔ δέχεται αιτήσεις από τους χρήστες των εφαρμογών και επικοινωνεί με τα αρχεία της βάσης δεδομένων για να τις διεκπεραιώσει.

Αυτή η κοινή *διεπαφή (interface)* των εφαρμογών με τα αρχεία αποκαλείται λογική διεπαφή. Οι εφαρμογές που δημιουργούμε δεν απασχολούνται με τον τρόπο που είναι αποθηκευμένα τα δεδομένα, πόσο χώρο καταλαμβάνουν κ.ο.κ. και αυτή η ιδιότητα είναι γνωστή ως *ανεξαρτησία δεδομένων*.

Αυτό σημαίνει πρακτικά ότι οποιαδήποτε αλλαγή στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων δεν θα συνεπάγεται και αλλαγή στις εφαρμογές· ένα πρόβλημα που ταλαιπωρούσε πολύ τους προγραμματιστές παλαιοτέρων εποχών. Ακόμη, η προσθήκη, η κατάργηση ή και η τροποποίηση κάποιων εφαρμογών δεν θα έχει καμία επίπτωση στον τρόπο οργάνωσης των αρχείων της βάσης δεδομένων. Στα ΣΔΒΔ έχει επικρατήσει η λεγόμενη αρχιτεκτονική των τριών επιπέδων (βαθμίδων), όπου τα τρία επίπεδα είναι τα εξής :

- ***Εσωτερικό επίπεδο*** *(internal level),* έχει να κάνει με την αποθήκευση των αρχείων στον σκληρό δίσκο, δηλ. την πραγματική ή φυσική κατάστασή τους. Ασχολείται με το κόστος, την απόδοση, την επεκτασιμότητα, με τη διάταξη αποθήκευσης του εννοιολογικού επιπέδου και παρέχει υποστήριξη αποθήκευσης-δομών (όπως π.χ. ευρετήρια όπου θα επικεντρωθούμε στο επόμενο κεφάλαιο).
- ***Εξωτερικό επίπεδο*** *(external level),* έχει να κάνει με τους χρήστες είτε αυτοί είναι απλοί χειριστές, είτε προγραμματιστές ή και διαχειριστές της βάσης δεδομένων και καθορίζει το πως κάθε τελικός χρήστης κατανοεί την οργάνωση των αντίστοιχων σχετικών στοιχείων του στη βάση δεδομένων.
- ***Εννοιολογικό επίπεδο*** *(conceptual level),* είναι ένα ενδιάμεσο επίπεδο που διασυνδέει τα δύο άλλα επίπεδα και έχει να κάνει με τη λογική σχεδίαση των αρχείων της βάσης δεδομένων. Παρέχεται με τον απλούστερο δυνατό τρόπο και αποτελεί τη σπονδυλική στήλη της βάσης δεδομένων. Είναι έξω από το πεδίο εφαρμογής των διάφορων τελικών χρηστών, εξυπηρετεί τους προγραμματιστές εφαρμογών και ορίζεται από τους διαχειριστές της βάσης δεδομένων.



Εικόνα 2.2 *Αρχιτεκτονική επιπέδων ΣΔΒΔ*

2.5.2 Μοντέλα Δεδομένων

Ένα μοντέλο δεδομένων είναι μια αφηρημένη δομή που παρέχει τα μέσα για να περιγράψει κάποιος αποτελεσματικά συγκεκριμένες δομές δεδομένων που απαιτούνται για τη μοντελοποίηση μίας εφαρμογής. Ως τέτοιο, ένα μοντέλο δεδομένων χρειάζεται την απαραίτητη εκφραστική δύναμη για να «συλλάβει» όλες τις απαραίτητες πτυχές των εφαρμογών. Αυτές οι εφαρμογές είναι συνήθως τυπικές στις εμπορικές επιχειρήσεις και τους οργανισμούς (π.χ. κατασκευές, τράπεζες, ανθρώπινοι πόροι κλπ). Για την αποτελεσματική χρήση και το χειρισμό του, είναι επιθυμητό ένα μοντέλο να είναι σχετικά απλό και διαισθητικό. Αυτό μπορεί να έρχεται σε σύγκρουση με την υψηλή εκφραστική δύναμη που απαιτείται από κάποιες πολύπλοκες εφαρμογές. Έτσι, κάθε δημοφιλές γενικού σκοπού μοντέλο δεδομένων συνήθως ισορροπεί μεταξύ της απλότητας, της διαισθητικότητας και της πολυπλοκότητας. Υπάρχουν τέσσερα βασικά μοντέλα που έχουν επικρατήσει στις βάσεις δεδομένων, το *ιεραρχικό*, το *δικτυωτό*, το *σχεσιακό* και το *μοντέλο οντοτήτων-συσχετίσεων*. Τα μοντέλα αυτά αναπτύχθηκαν με βάση αντίστοιχες δομές. Αυτό όμως που έχει επικρατήσει όλων, είναι το *μοντέλο οντοτήτων-συσχετίσεων*. Παρακάτω θα αναφερθούμε στο κάθε ένα ξεχωριστά.

- ***Το Ιεραρχικό Μοντέλο***

Το *ιεραρχικό μοντέλο* (*hierarchical model)* έχει μια ιεραρχική δομή που θυμίζει δένδρο. Οι οντότητες μοιάζουν με απολήξεις από κλαδιά δένδρων και

τοποθετούνται σε επίπεδα ιεραρχίας. Τα κλαδιά παριστάνουν τις συσχετίσεις ανάμεσα στις οντότητες.

Από μια οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο εκκινούν πολλά κλαδιά, καθένα από τα οποία καταλήγει σε μια οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο. Αλλά, σε κάθε οντότητα που βρίσκεται σ' ένα χαμηλότερο επίπεδο αντιστοιχεί μία και μόνο μία οντότητα που βρίσκεται σ' ένα ανώτερο επίπεδο. Το μοντέλο αυτό ήταν το πρώτο που εμφανίσθηκε αλλά σήμερα θεωρείται δύσχρηστο και ξεπερασμένο.

- *Το **Δικτυωτό Μοντέλο***

Και στο *δικτυωτό μοντέλο (network model)*, τα στοιχεία τοποθετούνται σ' ένα επίπεδο ιεραρχίας, αλλά κάθε στοιχείο μπορεί να συσχετισθεί με πολλά στοιχεία είτε σ' ένα κατώτερο ή σ' ένα ανώτερο επίπεδο. Αυτό το μοντέλο είναι πιο γενικό και δυναμικό από το ιεραρχικό και ήταν το πιο δημοφιλές, μέχρι που αντικαταστάθηκε από το σχεσιακό μοντέλο.

- ***Το Σχεσιακό Μοντέλο***

Το *σχεσιακό μοντέλο (relational model)* έχει επικρατήσει σήμερα στην αναπαράσταση των δεδομένων καθώς διαθέτει σημαντικά πλεονεκτήματα ως προς τα άλλα δύο και οι βάσεις δεδομένων που σχεδιάζονται σύμφωνα μ' αυτό αποκαλούνται *σχεσιακές (relational databases).* Με τις σχεσιακές βάσεις δεδομένων διαθέτουμε έναν σαφή, απλό και εύκολα κατανοητό τρόπο για να μπορέσουμε να αναπαραστήσουμε και να διαχειριστούμε τα δεδομένα μας. Υστερούν μόνο σε ταχύτητα υπολογισμών και σε χώρο αποθήκευσης, αλλά μόνο όταν έχουμε να κάνουμε με πολύ μεγάλες βάσεις δεδομένων.

Στο μοντέλο αυτό οι βάσεις δεδομένων περιγράφονται με αυστηρές μαθηματικές έννοιες και ο χρήστης βλέπει τις οντότητες και τις συσχετίσεις με τη μορφή πινάκων (tables) και σχέσεων (relations) αντίστοιχα.

Ένας *πίνακας* (table) αποτελείται από *γραμμές* (rows) και *στήλες* (columns), όπου τοποθετούμε τα στοιχεία σε οριζόντια και κάθετη μορφή. Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται *χαρακτηριστικό* (attribute) ή *πεδίο* (field), ενώ η κάθε γραμμή του πίνακα περιέχει όλες τις πληροφορίες (στήλες) που αφορούν ένα στοιχείο της οντότητας και αποκαλείται *πλειάδα* (tuple) ή *εγγραφή* (record).

Κάθε πεδίο του πίνακα μπορεί να πάρει ορισμένες μόνο τιμές, οι οποίες μπορεί να καθορίζονται από τον τύπο δεδομένων της ιδιότητας, όπως ονόματα ή αριθμοί για παράδειγμα, ή και από αυτό που εκφράζει, όπως, για

παράδειγμα, το ότι δεν μπορούμε να έχουμε αρνητικό βάρος ή αρνητικό ΑΦΜ. Το σύνολο των αποδεκτών τιμών μιας οντότητας αποκαλείται *πεδίο ορισμού* (domain).

Όπως είναι εύκολα κατανοητό, η βασικότερη εργασία που έχουμε να κάνουμε κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι να ορίσουμε τους πίνακες που θα χρησιμοποιήσουμε καθώς και τα πεδία που θα περιέχει ο καθένας απ' αυτούς. Η διαδικασία αυτή αποκαλείται κατασκευή του *σχήματος* (schema) μιας βάσης δεδομένων.

Οι κανόνες που πρέπει να ακολουθούμε πιστά κατά τον σχεδιασμό μιας σχεσιακής βάσης δεδομένων είναι οι εξής :

- Η κάθε οντότητα πρέπει να παριστάνεται ως ένας ξεχωριστός πίνακας.
- Η κάθε στήλη του πίνακα αντιστοιχεί σε μια ιδιότητα της οντότητας.
- Η κάθε γραμμή του πίνακα αντιστοιχεί σε μια εμφάνιση της οντότητας.
- Η κάθε γραμμή πρέπει να είναι μοναδική, δηλ. αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν τα ίδια ακριβώς στοιχεία.
- Η σειρά εμφάνισης των γραμμών δεν έχει καμία σημασία.
- Η κάθε στήλη έχει μια δική της μοναδική ονομασία.
- Οι τιμές που ανήκουν στην ίδια στήλη πρέπει να είναι του ιδίου τύπου, δηλ. ή όλες αριθμοί ή όλες αλφαριθμητικές κοκ.
- Η στήλη που αποτελεί το πρωτεύον κλειδί (primary key) μιας οντότητας, δεν πρέπει να είναι ποτέ κενή (null).
- Αποκλείεται να υπάρχουν δύο ή και περισσότερες γραμμές που να περιέχουν την ίδια τιμή στο πρωτεύον κλειδί.
- Το πρωτεύον κλειδί μιας οντότητας αποκαλείται ξένο κλειδί (foreign key) σε μια άλλη οντότητα, με την οποία υπάρχει συσχετισμός.
- Μπορεί να υπάρχουν πολλές γραμμές που να έχουν την ίδια τιμή στο ξένο κλειδί.

➢ *Το Μοντέλο Οντοτήτων-Συσχετίσεων*

Το μοντέλο που έχει επικρατήσει σήμερα για να παραστήσει τις έννοιες ή τη δομή μιας βάσης δεδομένων είναι το *Μοντέλο Οντοτήτων–Συσχετίσεων (ΟΣ)*. Οι βασικές (θεμελιώδεις) έννοιες του μοντέλου αυτού είναι οι εξής:

- Οντότητες
- Ιδιότητες ή Χαρακτηριστικά
- Συσχετίσεις

Για να αναπαραστήσουμε ένα Μοντέλο Οντοτήτων – Συσχετίσεων χρησιμοποιούμε ειδικά διαγράμματα, όπου τα ορθογώνια συμβολίζουν τις οντότητες, οι ρόμβοι τις συσχετίσεις και οι ελλείψεις τις ιδιότητες. Με ευθείες γραμμές συνδέουμε τις οντότητες που συσχετίζονται με κάποιο τρόπο μεταξύ τους. Όλα τα παραπάνω αποτελούν τη λογική δομή μιας βάσης δεδομένων,

μια εργασία που είναι απαραίτητο να γίνει πριν από την καταχώριση και την επεξεργασία των στοιχείων (πληροφοριών) της βάσης δεδομένων.

Το μοντέλο οντοτήτων–συσχετίσεων αποτελεί μια γενική περιγραφή των γενικών στοιχείων που απαρτίζουν μια βάση δεδομένων και απεικονίζει την αντίληψη που έχουμε για τα δεδομένα (εννοιολογικό), χωρίς να υπεισέρχεται σε λεπτομέρειες υλοποίησης.

Τα προβλήματα που είναι πιθανό να παρουσιασθούν κατά τη διαδικασία της υλοποίησης του σχεδιασμού μιας βάσης δεδομένων είναι η περιττή (άσκοπη) επανάληψη πληροφοριών, που είναι γνωστή με τον όρο redundancy, καθώς και δυσκολίες στην ενημέρωση της βάσης δεδομένων. Τα παραπάνω προβλήματα είναι γνωστά ως *πλεονασμοί δεδομένων* και *ανωμαλίες ενημέρωσης* και για να αντιμετωπιστούν με επιτυχία, θα πρέπει να διασπάσουμε τις μεγάλες σχέσεις σε μικρότερες. Αυτό γίνεται με τη διαδικασία της κανονικοποίησης, έτσι ώστε η βάση δεδομένων να είναι έτοιμη για καταχώριση στοιχείων.

Η μέθοδος της κανονικοποίησης βοηθάει στον λογικό σχεδιασμό μιας βάσης δεδομένων και είναι συμπληρωματική του μοντέλου οντοτήτων-συσχετίσεων. Το κέρδος για μας είναι ότι δεν υπάρχουν προβλήματα συνέπειας, πλεονασμού και εγκυρότητας των πληροφοριών της βάσης δεδομένων. Ακολουθώντας τη διαδικασία της κανονικοποίησης κάνουμε συνεχείς διασπάσεις των πινάκων σε πιο απλές και συμπαγείς μορφές, με στόχο πάντα να αποφύγουμε τον πλεονασμό (επανάληψη) των δεδομένων. Αφαιρούμε πεδία από τις αρχικές μεγάλες σχέσεις και τα τοποθετούμε σε νέες σχέσεις, έτσι ώστε να μπορούμε να έχουμε τις ίδιες πληροφορίες και με τις νέες σχέσεις.

Μπορούμε να πούμε ότι *κανονικοποίηση* (normalization) είναι η διαδικασία μετατροπής των δεδομένων κάποιων σχέσεων (πινάκων) σε πιο απλές και πιο σαφείς σχέσεις, χωρίς πλεονασμούς (επαναλήψεις) των δεδομένων.

2.5.3 Σχεσιακά ΣΔΒΔ (RDBMS) και Γλώσσες Βάσεων Δεδομένων

- ***Σχεσιακά ΣΔΒΔ (RDBMS)***

Τα Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (ΣΣΔΒΔ) ή RBMS (Relational DataBase Management Systems) αναπτύχθηκαν με βάση το σχεσιακό μοντέλο και έχουν επικρατήσει πλήρως στον χώρο. Κατά τον σχεδιασμό και τη δημιουργία μιας σχεσιακής βάσης δεδομένων, οι πίνακες αποτελούν το μοναδικό δομικό και απαραίτητο στοιχείο για να μπορέσουν να αναπαρασταθούν οι πληροφορίες που περιέχονται στη βάση δεδομένων.

Σήμερα, υπάρχουν εξελιγμένα εργαλεία διαχείρισης σε γραφικό και φιλικό προς τον χρήστη περιβάλλον για να κάνουμε τα εξής :

- Δημιουργία πινάκων
- Δημιουργία φορμών
- Δημιουργία ερωτημάτων
- Δημιουργία εκθέσεων (αναφορών)

Τα Σχεσιακά ΣΔΒΔ τα διακρίνουμε στα *μεγάλα*, τα οποία αφορούν κυρίως μεγάλους οργανισμούς και επιχειρήσεις, έχουν τεράστιο όγκο δεδομένων και πολλούς χρήστες ταυτόχρονα, και τέτοια συστήματα είναι τα Oracle, Ingres, Informix, SQL Server κ.ά. και τα *μικρά*, τα οποία αφορούν κυρίως απλούς χρήστες, όπως είναι η Microsoft Access, η Paradox, η FoxPro κ.ά.

➢ *Γλώσσες Βάσεων Δεδομένων – Γλώσσες Ερωταπαντήσεων (Query Languages)*

Οι γλώσσες βάσεων δεδομένων - γλώσσες ερωταπαντήσεων είναι προσαρμοσμένες γλώσσες προγραμματισμού και χρησιμοποιούνται για να :

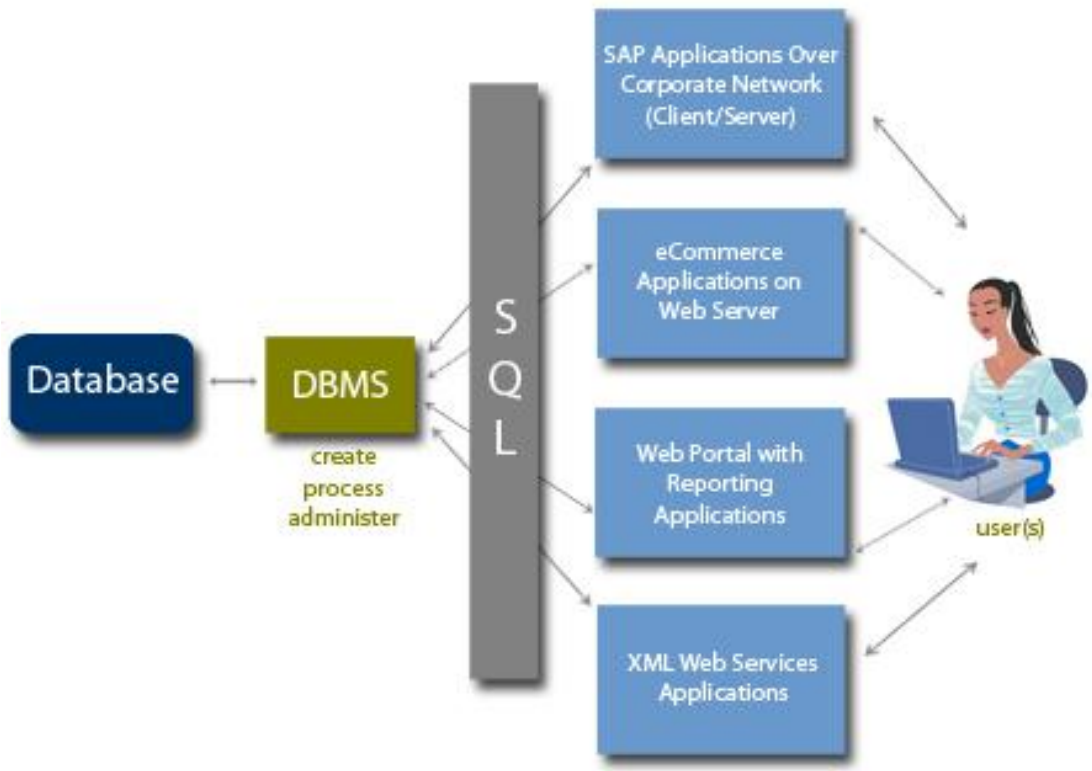
- ορίσουμε μια βάση δεδομένων (δηλαδή, συγκεκριμένους τύπους δεδομένων και τις σχέσεις μεταξύ τους)
- διαχειριστούμε το περιεχόμενό τους (π.χ. να εισάγουμε νέα δεδομένα συμβάντα και να ενημερώσουμε ή να διαγράψουμε υπάρχοντα)
- να κάνουμε ένα *ερώτημα - query* (δηλαδή, να ζητήσουμε πληροφορίες, να υπολογίσουμε και ανακτήσουμε τις πληροφορίες με βάση τα στοιχεία του).

Συνήθως περιέχουν εντολές στις οποίες αναθέτουν την εκτέλεση των επιθυμητών ενεργειών στη βάση δεδομένων. Κάθε τέτοια εντολή ισοδυναμεί με μία σύνθετη έκφραση σε κανονική γλώσσα προγραμματισμού. Με αυτό τον τρόπο οι γλώσσες βάσεων δεδομένων απλοποιούν τον χειρισμό των βάσεων δεδομένων.

Η πιο γνωστή και ευρέως διαδεδομένη γλώσσα εντολών για τις σύγχρονες βάσεις δεδομένων είναι η *Δομημένη Γλώσσα Ερωτήσεων* ***SQL*** *(Structured Query Language)*, η οποία αποτελείται από τα εξής μέρη :

- ***DDL*** *(Data Definition Language, Γλώσσα Ορισμού Δεδομένων)*, με την οποία καθορίζουμε τις δομές και τα τμήματα μιας βάσης δεδομένων.
- ***DML*** *(Data Manipulation Language, Γλώσσα Χειρισμού Δεδομένων)*, με την οποία επεξεργαζόμαστε τα δεδομένα μιας βάσης δεδομένων.
- ***DCL*** *(Data Control Language, Γλώσσα Ελέγχου Δεδομένων)*, με την οποία εξασφαλίζουμε την ασφάλεια και την ακεραιότητα των δεδομένων μιας βάσης δεδομένων.

Μπορεί να λειτουργήσει αυτόνομα αλλά και σε συνεργασία μ’ άλλες γλώσσες προγραμματισμού.



Εικόνα 2.3 *Παράδειγμα ενός συστήματος βάσης δεδομένων με χρήση της SQL.*

Μια άλλη, φιλική προς τον χρήστη γλώσσα προγραμματισμού για να μπορούμε να υποβάλουμε ερωτήματα σε σχεσιακές βάσεις δεδομένων και να λαμβάνουμε απαντήσεις είναι η *QBE (Query By Example)*, η οποία χρησιμοποιεί φόρμες για τη γραφική απεικόνιση των ερωτημάτων μας.

2.5.4 Ασφάλεια Βάσεων Δεδομένων

Η ασφάλεια των βάσεων δεδομένων ασχολείται με όλες τις διαφορετικές πτυχές της προστασίας του περιεχομένου της, των ιδιοκτητών της και των χρηστών της. Συμβάλλει στην προστασία από εσκεμμένη μη-εξουσιοδοτημένη χρήση της βάσης δεδομένων καθώς και την ακούσια πρόσβαση από μη-εξουσιοδοτημένες οντότητες (π.χ. ένα άτομο ή ένα πρόγραμμα). Παρακάτω αναφέρονται δύο σημαντικοί τομείς της ασφάλειας σε μια βάση δεδομένων.

- ***Έλεγχος πρόσβασης (Access Control)***

Ο έλεγχος πρόσβασης σε μία βάση δεδομένων ασχολείται με το να ελέγχει ποιος (ένα άτομο ή κάποιο πρόγραμμα του υπολογιστή) επιτρέπεται να έχει πρόσβαση σε αντίστοιχες πληροφορίες στην βάση. Οι πληροφορίες μπορεί να περιλαμβάνουν συγκεκριμένα αντικείμενα της βάσης (π.χ τύπους εγγραφών, συγκεκριμένες εγγραφές, δομές δεδομένων), υπολογισμούς σε

ορισμένα αντικείμενα (π.χ. τύπους ερωτημάτων ή και συγκεκριμένα ερωτήματα) ή να χρησιμοποιούν συγκεκριμένους δρόμους πρόσβασης σε αυτά τα αντικείμενα (π.χ. χρησιμοποιώντας συγκεκριμένα ευρετήρια ή άλλες δομές δεδομένων για να αποκτήσουν πρόσβαση σε πληροφορίες).

Οι έλεγχοι πρόσβασης σε μία βάση δεδομένων ορίζονται από ειδικό εξουσιοδοτημένο προσωπικό (από τον ιδιοκτήτη της βάσης), το οποίο χρησιμοποιεί αποκλειστικά προστατευμένα ΣΔΒΔ interfaces ασφαλείας.

- ***Ασφάλεια Δεδομένων (Data Security)***

Ο ορισμός της ασφάλειας δεδομένων ποικίλει και μπορεί να επικαλύπτεται από διάφορες πλευρές της ασφάλειας της βάσης δεδομένων. Σε γενικές γραμμές, ασχολείται με την προστασία συγκεκριμένων κομματιών δεδομένων, τόσο υλικά (Υλική Ασφάλεια – Physical Security), δηλαδή, από φθορά, καταστροφή ή αφαίρεση, όσο και από την ερμηνεία των δεδομένων ή μέρους αυτών (Κρυπτογράφηση Δεδομένων – Data Encryption),π.χ. εξετάζοντας τις σειρές των bits που αποτελούν ειδικούς έγκυρους αριθμούς πιστωτικών καρτών.

ΚΕΦΑΛΑΙΟ 3

Αποθηκευτικές Μονάδες Δίσκων, Οργανώσεις Αρχείων και Κατακερματισμός

3.1 Εισαγωγή

Οι βάσεις δεδομένων αποθηκεύονται φυσικά σαν αρχεία εγγραφών, που στην τυπική περίπτωση αποθηκεύονται σε μαγνητικούς δίσκους. Στο κεφάλαιο αυτό θα ασχοληθούμε με την οργάνωση της αποθήκευσης των βάσεων δεδομένων και τις τεχνικές για την αποτελεσματική προσπέλασή τους με χρήση διάφορων αλγορίθμων μερικοί από τους οποίους απαιτούν βοηθητικές δομές δεδομένων που ονομάζονται ευρετήρια.

Η συλλογή δεδομένων που αποτελεί μια βάση δεδομένων πρέπει να αποθηκευτεί φυσικά σε κάποιο **αποθηκευτικό μέσο** (storage medium) του υπολογιστή. Το λογισμικό του ΣΔΒΔ μπορεί στη συνέχεια να ανακτήσει, να ενημερώσει και να επεξεργαστεί αυτά τα δεδομένα, ανάλογα με τις ανάγκες. Τα αποθηκευτικά μέσα των υπολογιστών σχηματίζουν μια *ιεραρχία αποθήκευσης* που περιλαμβάνει δύο κύριες κατηγορίες:

- **Πρωτεύουσα αποθήκευση**: Η κατηγορία αυτή περιλαμβάνει τα αποθηκευτικά μέσα, τα οποία μπορεί άμεσα να χειριστεί η *Κεντρική Μονάδα Επεξεργασίας (ΚΜΕ)*, όπως η **κύρια μνήμη** του υπολογιστή που είναι η δυναμική RAM (DRAM, Dynamic Random Access Memory) και η **κρυφή μνήμη** (cache memory) που είναι η στατική RAM. Η πρωτεύουσα αποθήκευση συνήθως παρέχει γρήγορη προσπέλαση στα δεδομένα αλλά έχει περιορισμένη χωρητικότητα αποθήκευσης.
- **Δευτερεύουσα αποθήκευση**: Οι μονάδες δευτερεύουσας αποθήκευσης περιλαμβάνουν μαγνητικούς δίσκους, οπτικούς δίσκους, ταινίες και τύμπανα και συνήθως έχουν μεγαλύτερη χωρητικότα, κοστίζουν λιγότερο και παρέχουν πιο αργή προσπέλαση στα δεδομένα από τις μονάδες πρωτεύουσας αποθήκευσης. Η ΚΜΕ δεν μπορεί να επεξεργαστεί άμεσα δεδομένα που βρίσκονται σε μονάδες δευτερεύουσας αποθήκευσης, αυτά θα πρέπει να αντιγραφούν πρώτα στον πρωτεύοντα αποθηκευτικό χώρο.

3.2 Αποθήκευση Βάσεων Δεδομένων

Οι βάσεις δεδομένων συνήθως αποθηκεύουν μεγάλες ποσότητες δεδομένων που πρέπει να διατηρούνται αναλλοίωτα για μεγάλα χρονικά διαστήματα. Κατά την περίοδο αυτή γίνεται επανειλημμένη προσπέλαση και επεξεργασία των δεδομένων. Αυτο έρχεται σε αντιδιαστολή προς την έννοια των δομών δεδομένων, οι οποίες διατηρούνται μόνο για περιορισμένο χρόνο, δηλαδή κατά την εκτέλεση ενός προγράμματος. Οι περισσότερες βάσεις δεδομένων αποθηκεύονται μόνιμα σε δευτερεύουσες αποθηκευτικές μονάδες δίσκων (*μαγνητικοί δίσκοι*), για τους ακόλουθους κυρίως λόγους:

- Γενικά, οι βάσεις δεδομένων είναι πολύ μεγάλες για να χωρούν ολόκληρες στην κύρια μνήμη.
- Οι συνθήκες που οδηγούν σε μόνιμη απώλεια αποθηκευμένων δεδομένων εμφανίζονται λιγότερο συχνά για δευτερεύουσες αποθηκευτικές μονάδες δίσκων παρά για την πρωτεύουσα αποθήκευση.
- Το κόστος αποθήκευσης ανά μονάδα δεδομένων είναι κατά μία τάξη μεγέθους μικρότερο για τους δίσκους από ότι για την πρωτεύσουσα αποθήκευση.

Με την εισαγωγή των νεότερων τεχνολογιών (οπτικοί δίσκοι, DVD, BluRay) μπορεί να παραχθούν βιώσιμες εναλλακτικές λύσεις αντί της χρήσης

μαγνητικών δίσκων στο μέλλον. Ωστόσο, προβλέπεται ότι τα επόμενα χρόνια οι μαγνητικοί δίσκοι θα συνεχίσουν να αποτελούν το μέσο βασικής επιλογής για μεγάλες βάσεις δεδομένων. Ένα ακόμα θετικό στοιχείο των μαγνητικών δίσκων είναι ότι είναι μονάδες **άμεσης προσπέλασης** (on-line access), στις οποίες έχουμε πρόσβαση ανά πάσα στιγμή.

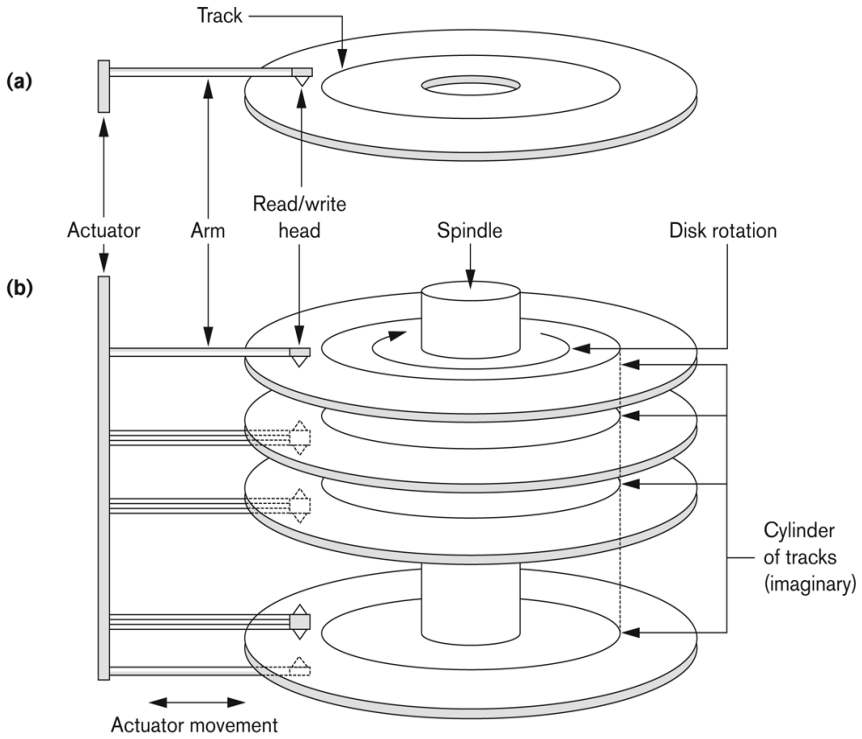
Οι τεχνικές που χρησιμοποιούνται για την αποθήκευση μεγάλων ποσοτήτων δομημένων δεδομένων στο δίσκο είναι σημαντικές για τους σχεδιαστές βάσεων δεδομένων, για το Διαχειριστή Βάσης Δεδομένων (ΔΒΔ) και για όσους υλοποιούν ΣΔΒΔ. Οι σχεδιαστές βάσεων δεδομένων και ο ΔΒΔ πρέπει να ξέρουν τα πλεονεκτήματα και τα μειονεκτήματα της κάθε τεχνικής όταν σχεδιάζουν, υλοποιούν και χειρίζονται μια βάση δεδομένων κάτω από ένα συγκεκριμένο ΣΔΒΔ. Συνήθως το ΣΔΒΔ παρέχει αρκετές διαθέσιμες επιλογές για την οργάνωση των δεδομένων, και η διαδικασία του *φυσικού σχεδιασμού της βάσης δεδομένων* συνεπάγεται την επιλογή της συγκεκριμένης τεχνικής οργάνωσης δεδομένων, μεταξύ των διαθέσιμων επιλογών, που ταιριάζει καλύτερα στις συγκεκριμένες απαιτήσεις της εφαρμογής.

Οι τυπικές εφαρμογές βάσεων δεδομένων χρειάζονται για επεξεργασία μόνο ένα μικρό κάθε στιγμή τμήμα της βάσης δεδομένων. Όταν χρειαστεί ένα τμήμα των δεδομένων θα πρέπει αυτό να εντοπιστεί στο δίσκο, να αντιγραφεί στην κύρια μνήμη για επεξεργασία και στη συνέχεια να γραφτεί ξανά στο δίσκο αν τα δεδομένα αλλάξουν. Τα δεδομένα που αποθηκεύονται στο δίσκο οργανώνονται σε **αρχεία εγγραφών (files of records)**. Κάθε εγγραφή είναι μια συλλογή τιμών δεδομένων που μπορούν να ερμηνευθούν ως γεγονότα σχετικά με οντότητες, τα γνωρίσματά τους και τις συσχετίσεις τους. Οι εγγραφές πρέπει να αποθηκεύονται στο δίσκο κατά τρόπο ώστε να είναι δυνατός ο αποδοτικός εντοπισμός τους όποτε χρειάζεται.

Υπάρχουν διάφορες **πρωτεύουσες οργανώσεις αρχείων (primary file organizations)** που καθορίζουν το πώς οι εγγραφές ενός αρχείου τοποθετούνται φυσικά στο δίσκο και επομένως πώς μπορεί να γίνει η προσπέλαση, όπως οι *μη ταξινομημένες εγγραφές* ή *αρχείο σωρού*, οι *ταξινομημένες εγγραφές* ή *σειριακό αρχείο*, ο *κατακερματισμός* και τα *Β-δένδρα*. Οι **δευτερεύουσες οργανώσεις αρχείων** αποτελούνται κυρίως από *ευρετήρια (index files)*, τα οποία χωρίζονται σε κατηγορίες όπως *πρωτεύοντα, δευτερεύοντα, ευρετήρια πολλών επιπέδων* κλπ., με τα οποία θα ασχοληθούμε αργότερα.

3.3 Μαγνητικοί Δίσκοι

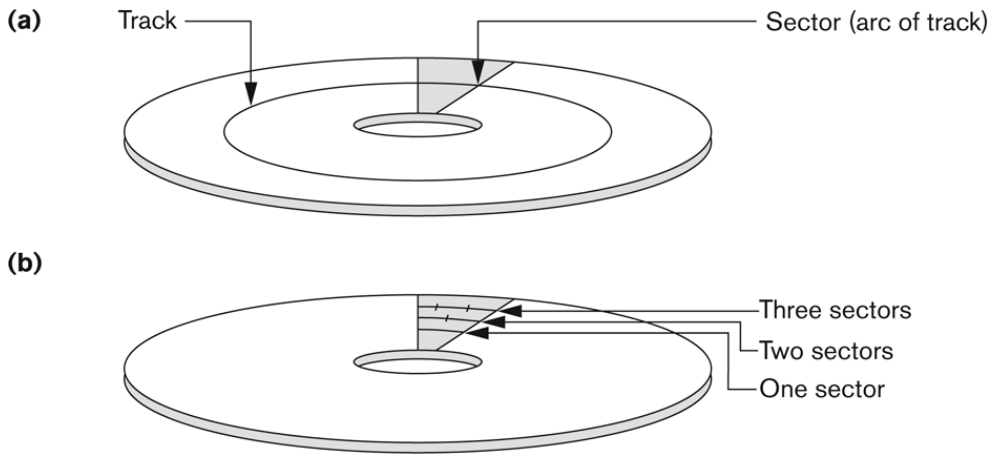
Οι δίσκοι προτιμώνται σαν αποθηκευτικές μονάδες λόγω μεγάλης χωρητικότητας και χαμηλού κόστους. Τα δεδομένα αποθηκεύονται σε επιφάνειες μαγνητικών δίσκων σαν μαγνητισμένες περιοχές. Ένας δίσκος είναι **απλής** όψης (single-sided) αν αποθηκεύει πληροφορίες μόνο στη μία επιφάνειά του και **διπλής** όψης (double-sided) αν χρησιμοποιούνται και οι δύο επιφάνειές του. Ένα **πακέτο δίσκων** περιέχει πολλούς μαγνητικούς δίσκους συνδεδεμένους σε ένα περιστρεφόμενο άξονα. Οι δίσκοι χωρίζονται σε ομόκεντρες κυκλικές επιφάνειες που λέγονται **άτρακτοι** (track) σε κάθε επιφάνεια του δίσκου. Η χωρητικότητα των ατράκτων ποικίλει από 4 σε 150 Kbytes ή περισσότερο και το πλήθος τους από μερικές εκατοντάδες μέχρι λίγες χιλιάδες. Στα πακέτα δίσκων, οι άτρακτοι με την ίδια διάμετρο στις διαφορετικές επιφάνειες λέγονται όλες μαζί **κύλινδρος** (cylinder) λόγω του σχήματος που θα σχημάτιζαν αν συνδέονταν στο χώρο. Η έννοια του κυλίνδρου είναι πολύ σημαντική, διότι δεδομένα που αποθηκεύονται στον ίδιο κύλινδρο μπορούν να ανακτηθούν πολύ γρηγορότερα από δεδομένα κατανεμημένα σε διαφορετικούς κυλίνδρους.



Εικόνα 3.1 *(α) Δίσκος μιας όψης με υλικό ανάγνωσης/εγγραφής. (β) Πακέτο δίσκων με υλικό ανάγνωσης/εγγραφής.*

Μια **άτρακτος** χωρίζεται σε μικρότερα **μπλοκ** (blocks) ή **τομείς** (sectors) επειδή συνήθως περιέχει μεγάλη ποσότητα πληροφοριών. Ο διαμερισμός μιας ατράκτου σε τομείς στην επιφάνεια του δίσκου είναι μόνιμα κωδικοποιημένος και δεν μπορεί να αλλάξει. Ένας τύπος οργάνωσης τομέων είναι αυτός που αποτελούν ένα τμήμα ατράκτου που εκτείνεται σε μια σταθερή γωνία στο κέντρο του τομέα. Ο χωρισμός της ατράκτου σε ισομεγέθη μπλοκ γίνεται κατά τη διάρκεια της **μορφοποίησης** (formatting) του δίσκου.Το μέγεθος του μπλοκ **B** είναι σταθερό για κάθε σύστημα. Τυπικά μεγέθη μπλοκ ποικίλουν από B=512 bytes μέχρι B=4096 bytes. Μεταξύ δίσκου και κύριας μνήμης μεταφέρονται για επεξεργασία ολόκληρα μπλοκ.

Υπάρχει συνεχής βελτίωση στην χωρητικότητα μεταφοράς δεδομένων των δίσκων. Επίσης σταδιακά γίνονται φθηνότεροι με την τιμή να έχει πέσει κάτω από €1/GB!



Εικόνα 3.2 *(α) Τομείς με σταθερή επίκεντρη γωνία. (β) Τομείς που διατηρούν την ίδια πυκνότητα εγγραφής.*

Ένας δίσκος θεωρείται μία συσκευή *τυχαίας προσπέλασης*. Στο μηχανισμό εισόδου/εξόδου του δίσκου παρέχονται η **διεύθυνση υλικού** (hardware address) και η διεύθυνση μιας **ενδιάμεσης μνήμης** (buffer), μιας συνεχόμενης δεσμευμένης περιοχής της κύριας μνήμης που χωρά ένα μπλοκ. Για μια **εντολή ανάγνωσης** (read), το μπλοκ από το δίσκο αντιγράφεται στην ενδιάμεση μνήμη ενώ για μια **εντολή εγγραφής** (write), τα περιεχόμενα της ενδιάμεσης μνήμης αντιγράφονται στα μπλοκ του δίσκου. Μερικές φορές αρκετα συνεχόμενα μπλοκ, που λέγονται **συστάδα** (cluster) μπορούν να μεταφερθούν ως μια ενότητα. Στην περίπτωση αυτή, το μέγεθος της ενδιάμεσης μνήμης προσαρμόζεται για να ταιριάζει στο πλήθος byte που περιέχει η συστάδα.

Ο πραγματικός μηχανισμός που διαβάζει ή γράφει μια σελίδα είναι η **κεφαλή ανάγνωσης/γραφής** (read/write head) του δίσκου, η οποία αποτελεί τμήμα ενός συστήματος που λέγεται **οδηγός δίσκων** (disk drive). Ένας δίσκος ή ένα πακέτο δίσκων τοποθετείται στη μονάδα δίσκων, η οποία περιλαμβάνει και έναν κινητήρα που περιστρέφει τους δίσκους. Μια κεφαλή ανάγνωσης/γραφής περιλαμβάνει ένα ηλεκτρονικό στοιχείο προσαρτημένο σε ένα **μηχανικό βραχίονα** (mechanical arm). Πακέτα δίσκων με πολλές επιφάνειες ελέγχονται από πολλαπλές κεφαλές ανάγνωσης/γραφής – μία για κάθε επιφάνεια (βλ. Εικόνα 3.1.β). Όλοι οι βραχίονες συνδέονται με ένα **χτένι** (comb ή actuator) προσαρτημένο σε έναν άλλο ηλεκτρικό κινητήρα ο οποίος κινεί μαζί τις κεφαλές ανάγνωσης/γραφής και τις τοποθετεί ακριβώς στον κύλινδρο ατράκτων που προσδιορίζει η διεύθυνση της σελίδας.

Η **κεφαλή ανάγνωσης/γραφής** μετακινείται προς την άτρακτο που περιέχει το μπλοκ για μεταφορά. Η περιστροφική κίνηση του δίσκου μεταφέρει το μπλοκ κάτω από την κεφαλή ανάγνωσης/γραφής για διάβασμα ή γράψιμο. Μια φυσική διεύθυνση μπλοκ (hardware) αποτελείται από:

- τον αριθμό κυλίνδρου (υποθετική συλλογή ατράκτων της ίδιας ακτίνας από όλες τις επιφάνειες),
- τον αριθμό ατράκτου ή τον αριθμό της επιφάνειας (στον κύλινδρο),
- τον αριθμό του μπλοκ (στην άτρακτο).

Μερικές μονάδες δίσκων έχουν σταθερές κεφαλές ανάγνωσης/γραφής και μάλιστα τόσες κεφαλές όσες και οι άτρακτοι. Αυτοί οι δίσκοι λέγονται **δίσκοι σταθερής κεφαλής** (fixed head), ενώ οι μονάδες δίσκων με κτένι ονομάζονται **δίσκοι κινητής κεφαλής** (movable head). Στους δίσκους σταθερής κεφαλής μια άτρακτος ή ένας κύλινδρος επιλέγεται ενεργοποιώντας ηλεκτρονικά την κατάλληλη κεφαλή ανάγνωσης/γραφής και όχι με πραγματική μηχανική κίνηση, τεχνική που καθιστά την επιλογή πολύ ταχύτερη. Ωστόσο, το κόστος των επιπλέον κεφαλών ανάγνωσης/γραφής είναι αρκετά υψηλό, και έτσι οι δίσκοι σταθερής κεφαλής δεν χρησιμοποιούνται συχνά.

3.3.1 Παράμετροι των Δίσκων

Στην λειτουργία του, ο δίσκος περιστρέφεται με τη μέγιστη ταχύτητα περιστροφής (συνήθως με 7.200 rpm – 120 περιστροφές το δευτερόλεπτο). Ο χρόνος για να μεταφερθεί ένα μπλοκ από το δίσκο αποτελείται από κάποια βήματα:

- Τον **χρόνο εντοπισμού** (seek time - **s**): είναι ο χρόνος που απαιτείται για να μετακινηθεί η κεφαλή από τον έναν κύλινδρο στον άλλον και κάτω από τη σωστή άτρακτο. Αυτό είναι τυπικά το πιο αργό βήμα της διαδικασίας και μυμαίνεται από 7 έως 10 msec για

τους προσωπικούς υπολογιστές και από 3 έως 8 msec για τους εξυπηρετητές (servers).

- Την **καθυστέρηση περιστροφής** (rotational latency - **rd**): είναι ο χρόνος που απαιτείται για την περιστροφή του επιθυμητού μπλοκ μέχρι να φτάσει κάτω από την κεφαλή ανάγνωσης/γραφής. Αυτό κυμαίνεται από μηδέν έως μία ολόκληρη περιστροφή με τη μέση τιμή να είναι περίπου μισή περιστροφή.(π.χ. για ένα δίσκο με ταχύτητα περιστροφής 7.200 rpm, η μέση καθυστέρηση περιστροφής είναι 1/2*1/120 περιστροφές το δευτερόλεπτο, δηλαδή 4 msec, ένας μεγάλος χρόνος για τα δεδομένα ενός υπολογιστή.
- Τον **χρόνο μεταφοράς μπλοκ** (block transfer time - **btt**): είναι ο επιπλέον χρόνος που απαιτείται για την μεταφορά των δεδομένων. Αυτός ο χρόνος μεταφοράς του μπλοκ εξαρτάται από το μέγεθος του μπλοκ, το μέγεθος της ατράκτου και την ταχύτητα περιστροφής. Αν ο ρυθμός μεταφοράς για το δίσκο είναι **tr byte/msec** και το μέγεθος του μπλοκ είναι **B** byte , τότε **btt=B/tr msec**.

Επομένως, ο ολικός χρόνος που απαιτείται για τον εντοπισμό και τη μεταφορά ενός μπλοκ, όταν δίνεται η διεύθυνσή του, είναι το άθροισμα του χρόνου εντοπισμού, της καθυστέρησης περιστροφής και του χρόνου μεταφοράς του μπλοκ (**s+rd+btt**). Αυτό ισχύει είτε για ανάγνωση είτε για εγγραφή μπλοκ.

3.3.2 Μεταφορά Μπλοκ σε Ενδιάμεση Μνήμη

Η κύρια μέθοδος ελάττωσης αυτού του χρόνου είναι να μεταφέρονται μαζικά πολλά μπλοκ που είναι αποθηκευμένα σε μια ή περισσότερες ατράκτους του ίδιου κυλίνδρου. Τότε ο χρόνος αναζήτησης απαιτείται μόνο για το πρώτο μπλοκ. Για να **μεταφερθούν διαδοχικά k συνεχόμενα μπλοκ** που βρίσκονται στον ίδιο κύλινδρο απαιτούνται κατά προσέγγιση **s+(k*(rd+btt)) msec**. Με τη μέθοδο αυτή μπορούν να δεσμευτούν αρκετές περιοχές ενδιάμεσης μνήμης στην κύρια μνήμη για να επιταχυνθεί η μεταφορά. Ενώ μια ενδιάμεση μνήμη διαβάζεται ή γράφεται, η ΚΜΕ μπορεί να επεξεργάζεται δεδομένα από μια άλλη ενδιάμεση μνήμη. Αυτό είναι δυνατόν επειδή συνήθως υπάρχει ένας ανεξάρτητος επεξεργαστής εισόδου/εξόδου του δίσκου, ο οποίος μόλις ξεκινήσει μπορεί να μεταφέρει ένα μπλοκ δεδομένων μεταξύ μνήμης και δίσκου ανεξάρτητα από, και παράλληλα με τη λειτουργία της ΚΜΕ.

Για επιτάχυνση της μεταφοράς μπορεί να χρησιμοποιηθεί τεχνική **διπλής ενδιάμεσης μνήμης** (double buffering). Η ΚΜΕ μπορεί να αρχίσει

την επεξεργασία ενός μπλοκ μόλις η μεταφορά του στην κύρια μνήμη ολοκληρωθεί. Την ίδια στιγμή ο επεξεργαστής Εισόδου-Εξόδου (Ε/Ε) του δίσκου μπορεί να διαβάζει και να μεταφέρει το επόμενο μπλοκ σε μια διαφορετική ενδιάμεση μνήμη. Η τεχνική διπλής ενδιάμεσης μνήμης επιτρέπει την μεταφορά συνεχόμενων μπλοκ στην ίδια άτρακτο ή στον ίδιο κύλινδρο. Αυτό εξαλείφει τον χρόνο εντοπισμού και την καθυστέρηση περιστροφής για όλα τα μπλοκ εκτός από το πρώτο και επομένως η εκτίμηση για την μεταφορά k συνεχόμενων μπλοκ είναι **s+rd+(k*btt) msec**.

Μια πιο ακριβής εκτίμηση για τη μεταφορά διαδοχικών μπλοκ λαμβάνει υπ' όψην το διάκενο μεταξύ των μπλοκ, το οποίο περιλαμβάνει τις πληροφορίες που επιτρέπουν στην κεφαλή ανάγνωσης/γραφής να προσδιορίσει ποιο μπλοκ να διαβάσει. Συνήθως ο κατασκευαστής του δίσκου δίνει τον **συνολικό ρυθμό μεταφοράς** (bulk transfer rate – **btr**), ο οποίος λαμβάνει υπ΄όψην το μέγεθος του διάκενου όταν διαβάζονται διαδοχικά αποθηκευμένα μπλοκ. Αν το διάκενο έχει μέγεθος **G** byte τότε **btr=(B/B+G)*tr byte/msec**. Επομένως, η εκτίμηση του χρόνου ανάγνωσης k μπλοκ αποθηκευμένων διαδοχικά στον ίδιο κύλινδρο γίνεται **s+rd+(k*(B/btr)) msec**.

Ενδεικτικά, αναφέρουμε ότι ο χρόνος εντοπισμού και μεταφοράς ενός μπλοκ είναι της τάξης των msec (βλ. παραπάνω), συνήθως μεταξύ 12 και 60 χιλιοστών. Για συνεχόμενα μπλοκ, ο εντοπισμός του πρώτου μπλοκ απαιτεί 12-60 msec αλλά η μεταφορά των διαδοχικών μπλοκ μπορεί να χρειαστεί μόνο 1 ή 2 msec για το καθένα. Σε κάθε περίπτωση, ένας χρόνος της τάξης των χιλιοστών δευτερολέπτου θεωρείται αρκετά υψηλός σε σύγκριση με το χρόνο που απαιτείται για την επεξεργασία των δεδομένων στην κύρια μνήμη από τις σύγχρονες ΚΜΕ. Επομένως, ο εντοπισμός δεδομένων στο δίσκο αποτελεί ένα μείζον σημείο συμφόρησης για τις εφαρμογές βάσεων δεδομένων.

3.4 Εγγραφές

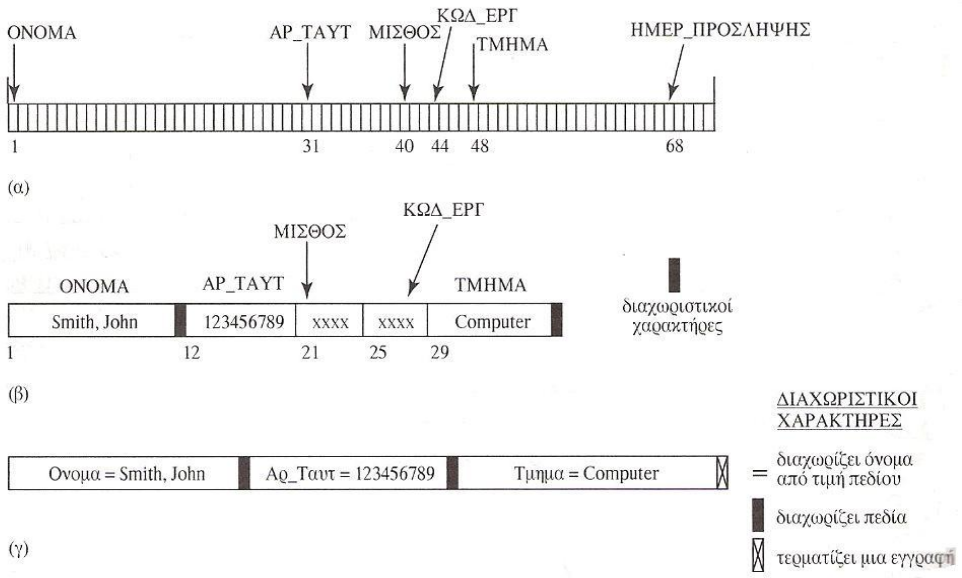
Η **εγγραφή** (record) είναι η μορφή αποθήκευσης των δεδομένων στον δίσκο. Αποτελείται από **τιμές** (values), όπου κάθε τιμή σχηματίζεται από ένα ή περισσότερα byte και αντιστοιχεί σε συγκεκριμένα **πεδία** (fields) της εγγραφής. Οι εγγραφές συνήθως περιγράφουν οντότητες και τα γνωρίσματά τους. Μια συλλογή από ονόματα πεδίων και τους αντίστοιχους τύπους δεδομένων αποτελεί τον ορισμό ενός **τύπου** ή μιας **μορφής εγγραφών** (record type ή record format).

Ο τύπος των δεδομένων ενός πεδίου που χρησιμοποιείται είναι συνήθως ένας από τους κοινούς τύπους δεδομένων που χρησιμοποιείται

στον προγραμματισμό, δηλαδή, αριθμητικοί τύποι, συμβολοσειρές χαρακτήρων, λογικοί τύποι και μερικές φορές ειδικά κωδικοποιημένοι τύποι για την ημερομηνία και την ώρα. Επίσης, σε σύγχρονες εφαρμογές βάσεων δεδομένων μπορεί να χρειαστεί να αποθηκευτούν μεγάλα αντικείμενα, όπως εικόνες, βίντεο, αποσπάσματα ήχου ή και ελεύθερο κείμενο. Αυτά αναφέρονται ως **BLOBs** (Binary Large Objects). Κανονικά, ένα στοιχείο δεδομένων τύπου BLOB αποθηκεύεται ξεχωριστά από την αντίστοιχη εγγραφή σε μια ομάδα μπλοκ δίσκου και στην εγγραφή περιλαμβάνεται ένας δείκτης προς το BLOB. Το μέγεθος του κάθε τύπου δεδομένων εξαρτάται πολλές φορές από το λειτουργικό σύστημα.

3.4.1 Αρχεία, Εγραφές Σταθερού και Μεταβλητού Μήκους

Ένα **αρχείο** (file) είναι μια ακολουθία εγγραφών, που κάθε εγγραφή είναι μια συλλογή από τιμές δεδομένων. Αν κάθε εγγραφή έχει ακριβώς το ίδιο μέγεθος σε byte, τότε λέμε ότι είναι **εγγραφή σταθερού μήκους** (fixed-length). Αν διαφορετικές εγγραφές του αρχείου έχουν διαφορετικό μέγεθος λέμε ότι το αρχείο αποτελείται από **εγγραφές μεταβλητού μήκους** (variable-length).



Εικόνα 3.3 *(α) Εγγραφή σταθερού μήκους. (β) Εγγραφή μεταβλητού μήκους. (γ) Εγγραφή με τρεις τύπους διαχωριστικών χαρακτήρων.*

Στις εγγραφές σταθερού μήκους, κάθε εγγραφή έχει τα ίδια πεδία και τα μήκη των πεδίων είναι σταθερά, και έτσι μπορούμε να προσδιορίσουμε τη θέση του αρχικού byte κάθε πεδίου σε σχέση με την αφετηρία της εγγραφής. Αυτό διευκολύνει τον εντοπισμό των τιμών ενός πεδίου από προγράμματα

που προσπελαύνουν τέτοια αρχεία. Σημαντικό, επίσης, είναι το γεγονός ότι είναι δυνατόν να παρασταθεί ένα αρχείο που λογικά θα έπρεπε να έχει εγγραφές μεταβλητού μήκους ως αρχείο εγγραφών σταθερού μήκους.

Για τα αρχεία μεταβλητού μήκους, κάθε εγγραφή έχει μια τιμή για κάθε πεδίο, αλλά δεν ξέρουμε το ακριβές μήκος των τιμών μερικών πεδίων. Για να προσδιορίσουμε τα bytes που παριστάνουν κάθε πεδίο μέσα σε μια συγκεκριμένη εγγραφή, μπορούμε να χρησιμοποιήσουμε ειδικούς *διαχωριστικούς* χαρακτήρες (separators) - οι οποίοι δεν εμφανίζονται στην τιμή οποιουδήποτε πεδίου (π.χ. ?, %, $) – για το τέλος των πεδίων μεταβλητού μεγέθους, ή μπορούμε να αποθηκεύουμε το μήκος ενός τέτοιου πεδίου στην εγγραφή.

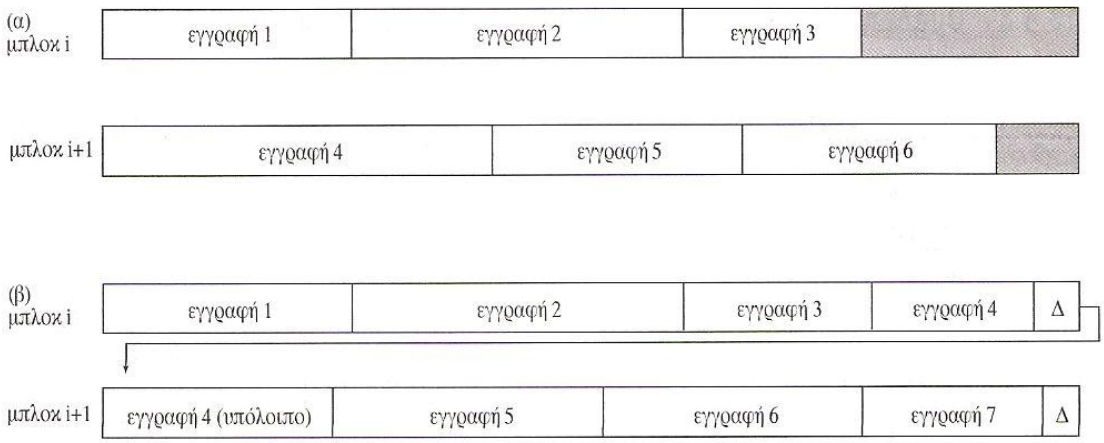
Εύκολα μπορεί να καταλάβει κανείς, ότι τα προγράμματα που επεξεργάζονται εγγραφές μεταβλητού μήκους χρειάζεται να είναι πολυπλοκότερα από αυτά για εγγραφές σταθερού μήκους, όπου η θέση αφετηρίας και το μέγεθος κάθε πεδίου είναι γνωστά και σταθερά.

3.4.2 Ομαδοποίηση Εγγραφών

Ο **περιγραφέας αρχείου** ή **επικεφαλίδα αρχείου** (file header ή file descriptor) περιλαμβάνει πληροφορίες που περιγράφουν το αρχείο, όπως τα ονόματα των πεδίων, τους τύπους δεδομένων και τις διευθύνσεις των μπλοκ του αρχείου στο δίσκο. Οι εγγραφές ενός αρχείου τοποθετούνται σε μπλοκ του δίσκου. Όταν το μέγεθος ενός μπλοκ **B** είναι μεγαλύτερο από το μέγεθος μιας εγγραφής **R**, κάθε μπλοκ θα περιέχει πολλές εγγραφές. Υποθέτωντας ότι το μέγεθος ενός μπλοκ είναι B byte και το μέγεθος των εγγραφών σταθερού μήκους ενός αρχείου είναι R byte με B≥R, μπορούμε να τοποθετήσουμε **bfr=⌊(B/R⌋** εγγραφές ανά μπλοκ, όπου ο αριθμός στρογγυλεύεται προς τα κάτω στον προηγούμενο ακέραιο. Ο αριθμός αυτός ονομάζεται **παράγοντας ομαδοποίησης** (blocking factor – bfr). Συνεπώς, μπορεί να υπολογιστεί και ο αριθμός των μπλοκ που απαιτούνται για ένα αρχείο που περιέχει r εγγραφές, δηλαδή, **b=⌈(r/bfr)⌉**, όπου ο αριθμός στρογγυλεύεται προς τα πάνω στον επόμενο ακέραιο.

Επειδή, το R μπορεί να μην διαιρεί ακριβώς το B, έχουμε γενικά αχρησιμοποίητο χώρο ίσο με **B-(bfr*R) byte**. Για να αξιοποιήσουμε αυτόν τον αχρησιμοποίητο χώρο, μπορούμε να αποθηκεύσουμε μέρος μιας εγγραφής σε ένα μπλοκ και το υπόλοιπο σε ένα άλλο. Ένας **δείκτης** (pointer) στο τέλος του πρώτου μπλοκ δείχνει το μπλοκ που περιέχει το υπόλοιπο μέρος της εγγραφής. Αυτή η οργάνωση των εγγραφών ονομάζεται **εκτεινόμενη** (spanned), επειδή οι εγγραφές εκτείνονται σε περισσότερα από ένα μπλοκ. Αντίθετα, όταν δεν επιτρέπεται να χρησιμοποιείται εκτεινόμενη οργάνωση,

χρησιμοποιούνται εγγραφές σταθερού μήκους με B>R. Αυτή η οργάνωση ονομάζεται **μη-εκτεινόμενη** (unspanned). Στην περίπτωση που έχουμε εγγραφή μεταβλητού μήκους, μπορούμε να χρησιμοποιήσουμε είτε εκτεινόμενη είτε μη εκτεινόμενη οργάνωση, και συγκεκριμένα αν το μέγεθος είναι μεγάλο είναι καλύτερη η εκτεινόμενη οργάνωση για να ελαττώσουμε τον αχρησιμοποίητο χώρο σε κάθε μπλοκ.



Εικόνα 3.4 *(α) Μη εκτεινόμενη οργάνωση εγγραφών. (β) Εκτεινόμενη οργάνωση εγγραφών.*

3.4.3 Πράξεις Σε Εγγραφές Αρχείων

Οι πράξεις σε αρχεία συνήθως χωρίζονται σε **πράξεις ανάκτησης** (retrieval) και **πράξεις ενημέρωσης** (update). Οι πρώτες δεν αλλάζουν οποιαδήποτε δεδομένα στο αρχείο, απλά εντοπίζουν τα επιθυμητά δεδομένα για να υποστούν κάποια επεξεργασία. Οι δεύτερες αλλάζουν το αρχείο με την εισαγωγή ή την διαγραφή εγγραφών ή και με την τροποποίηση των τιμών κάποιων εγγραφών. Όμως, και στις δύο περιπτώσεις, ίσως χρειαστεί να επιλέξουμε μια ή περισσότερες εγγραφές για ανάκτηση, διαγραφή ή τροποποίηση βάση μιας **συνθήκης επιλογής** (selection condition), η οποία προσδιορίζει τα κριτήρια που πρέπει να ικανοποιούν οι επιλεγμένες εγγραφές.

Όταν πολλές εγγραφές ενός αρχείου ικανοποιούν μια συνθήκη αναζήτησης, μόνο η πρώτη εγγραφή - σε σχέση με τη φυσική διάταξη των εγγραφών στο αρχείο – εντοπίζεται αρχικά και χαρακτηρίζεται **τρέχουσα εγγραφή** (current record). Οι επόμενες πράξεις αναζήτησης αρχίζουν από αυτήν την εγγραφή και εντοπίζουν την επόμενη εγγραφή στο αρχείο που ικανοποιεί την συνθήκη.

Οι πραγματικές πράξεις για τον εντοπισμό και την προσπέλαση των εγγραφών διαφέρουν από σύστημα σε σύστημα. Εδώ, θα παραθέσουμε κάποιες αντιπροσωπευτικές πράξεις:

- **Βρες εγγραφή** (Find ή Locate): βρίσκει την εγγραφή σύμφωνα με την συνθήκη αναζήτησης και την τοποθετεί στην ενδιάμεση μνήμη.
- **Διάβασε** (Read): διαβάζει και αντιγράφει την εγγραφή από την ενδιάμεση μνήμη στην περιοχή του χρήστη.
- **Βρες επόμενη εγγραφή** (Find Next): βρίσκει την επόμενη εγγραφή και την τοποθετεί στην ενδιάμεση μνήμη (αν δεν βρίσκεται ήδη εκεί).
- **Διέγραψε** (Delete): διαγράφει την τρέχουσα εγγραφή και ενημερώνει τον δίσκο.
- **Εισήγαγε νέα εγγραφή** (Insert): μεταφέρεται το αντίστοιχο μπλοκ στην ενδιάμεση μνήμη, εισάγεται η εγγραφή και ενημερώνεται ο δίσκος.

Εδώ, αξίζει να επισημάνουμε την διαφορά μεταξύ των όρων οργάνωση αρχείου και μέθοδος προσπέλασης. Μια **οργάνωση αρχείου** (file organization) αναφέρεται στην οργάνωση των δεδομένων ενός αρχείου σε εγγραφές, μπλοκ και δομές προσπέλασης, συμπεριλαμβανομένου και του τρόπου με τον οποίο οι εγγραφές και τα μπλοκ τοποθετούνται στο μέσο αποθήκευσης και συνδέονται μεταξύ τους. Μια **μέθοδος προσπέλασης** (access method) αποτελείται από μια ομάδα προγραμμάτων που επιτρέπουν να εφαρμοστούν κάποιες πράξεις (όπως αυτές που αναφέρθηκαν προηγουμένως) σε ένα αρχείο. Γενικά, είναι πιθανό να χρησιμοποιούνται αρκετές διαφορετικές μέθοδοι προσπέλασης για την ίδια οργάνωση αρχείου.

Δυστυχώς, σε πολλές περιπτώσεις μπορεί να μην υπάρχει κάποια οργάνωση που να μην επιτρέπει την αποτελεσματική υλοποίηση όλων των πράξεων που πρέπει να εκτελούνται επί του αρχείου. Μερικές οργανώσεις αρχείων καθιστούν την ανάκτηση για κάποιες συνθήκες αναζήτησης πολύ αποτελεσματική, με επόμενο όμως ότι οι ενημερώσεις γίνονται πολύ ακριβές. Στη συνέχεια, θα μας απασχολήσει ιδιαίτερα η ευρετηριοποίηση ως μέθοδος οργάνωσης των εγγραφών ενός αρχείου στο δίσκο.

3.5 Διάταξη Εγγραφών Σε Αρχεία

3.5.1 Μη Διατεταγμένα Αρχεία – Αρχεία Σωρού (Heap File)

Αυτός ο τρόπος διάταξης αποτελεί τον πιο απλό και βασικό τύπο οργάνωσης, όπου οι εγγραφές τοποθετούνται στο αρχείο με τη σειρά που

εισάγονται και οι καινούργιες εγγραφές τοποθετούνται στο τέλος του αρχείου. Τα αρχεία σωρού συνήθως χρησιμοποιούνται σε συνδυασμό με τα δευτερεύοντα ευρετήρια.

Στην διαδικασία της εισαγωγής, το τελευταίο μπλοκ του αρχείου στο δίσκο αντιγράφεται σε μια ενδιάμεση μνήμη, η καινούργια εγγραφή προστίθεται και η σελίδα γράφεται ξανά στο δίσκο. Η διεύθυνση του τελευταίου μπλοκ αποθηκεύεται στην επικεφαλίδα του αρχείου. Ωστόσο, η αναζήτηση μιας εγγραφής είναι μια «ακριβή» διαδικασία από την άποψη του χρόνου, διότι συνεπάγεται γραμμική αναζήτηση στο αρχείο για ένα-ένα μπλοκ. Με την προϋπόθεση ότι η εγγραφή υπάρχει στον δίσκο, απαιτούνται κατά την **σειριακή αναζήτηση** ανάγνωση, κατά μέσο όρο, **b/2 μπλοκ**. Είναι κατανοητό ότι στην περίπτωση όπου καμία ή πολλές εγγραφές ικανοποιούν την συνθήκη αναζήτησης, θα πρέπει να προσπελαστούν όλα τα μπλοκ.

Η διαγραφή οδηγεί σε εσωτερικό κατακερματισμό, ο οποίος θα αναφερθεί αργότερα. Η ταξινόμηση είναι μια ακριβή πράξη, όπου χρησιμοποιούνται τεχνικές εξωτερικής ταξινόμησης, οι οποίες δεν θα μας απασχολήσουν στα πλαίσια αυτής της εργασίας.

Για ένα αρχείο μη εκτεινόμενων εγγραφών σταθερού μεγέθους μπορεί να γίνει άμεση προσπέλαση της εγγραφής **i** με βάση τη θέση της στο αρχείο. Δηλαδή, η εγγραφή βρίσκεται στο μπλοκ $\lfloor$ **(i / bfr)** $\rfloor$ και είναι η **(i mod bfr)** εγγραφή στο μπλοκ. Η προσπέλαση της εγγραφής με βάση τη θέση της είναι χρήσιμη για την ευρετηριοποίηση.

3.5.2 Ταξινομημένα Αρχεία

Σε αυτά τα αρχεία, η διάταξη των εγγραφών γίνεται με βάση τις τιμές του **πεδίου διάταξης** (ordering field), το οποίο είναι συνήθως το **πρωτεύον κλειδί** (primary key) που ονομάζεται **κλειδί διάταξης** (ordering key). Η εισαγωγή και διαγραφή εγγραφών του αρχείου είναι ακριβείς πράξεις, όπου χρησιμοποιούνται αρχεία υπερχείλισης (overflow file) ή αρχεία δοσοληψιών (transaction file) για την αντιμετώπιση του προβλήματος.

Η αναζήτηση στα ταξινομημένα αρχεία γίνεται πάνω σε μπλοκ παρά πάνω σε εγγραφές. Ονομάζεται **δυαδική αναζήτηση** (binary search) και απαιτούνται $\mathbf{log_2 b}$ προσπελάσεις ανεξάρτητα από το αν υπάρχει η εγγραφή ή όχι. Σαφώς, αυτή η διαδικασία αποτελεί βελτίωση σε σχέση με την σειριακή αναζήτηση όπου απαιτούνται κατά μέσο όρο b/2 προσπελάσεις μπλοκ.

Τα διατεταγμένα αρχεία έχουν κάποια πλεονεκτήματα σε σχέση με τα μη διατεταγμένα:

1. Η ανάγνωση των εγγραφών με διάταξη των τιμών του πεδίου διάταξης γίνεται εξαιρετικά αποτελεσματική, καθώς δεν απαιτείται ταξινόμηση.
2. Η εύρεση της επόμενης από την τρέχουσα εγγραφής κατά σειρά του πεδίου διάταξης συνήθως δεν απαιτεί ανάγνωση νέου μπλοκ (εκτός αν η επόμενη εγγραφή είναι η τελευταία).
3. Η χρήση μιας συνθήκης αναζήτησης που βασίζεται στην τιμή του κλειδιού διάταξης οδηγεί σε γρηγορότερη προσπέλαση όταν χρησιμοποιείται η τεχνική της δυαδικής αναζήτησης.

Τα διατεταγμένα αρχεία σπάνια χρησιμοποιούνται σε εφαρμογές βάσεων δεδομένων, εκτός αν στο αρχείο συμπεριλαμβάνεται ένας πρόσθετος δρόμος προσπέλασης που λέγεται **πρωτεύον ευρετήριο** (primary index). Αυτό βελτιώνει περαιτέρω το χρόνο τυχαίας προσπέλασης με βάση το πεδίο κλειδιού διάταξης και θα το εξετάσουμε στο επόμενο κεφάλαιο.

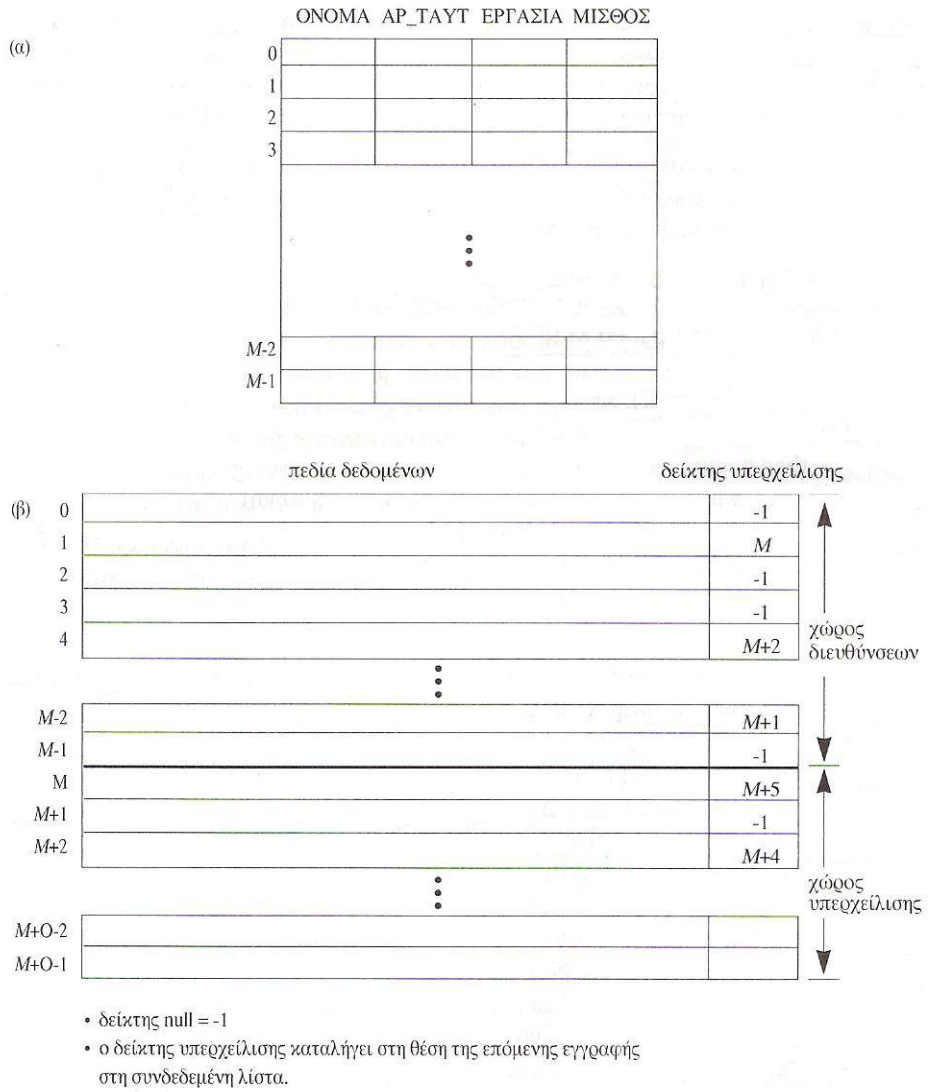
3.6 Κατακερματισμός

Ο κατακερματισμός είναι μια άλλη τεχνική πρωτεύουσας οργάνωσης ενός αρχείου, η οποία παρέχει - υπό ορισμένες συνθήκες - ταχύτατη πρόσβαση στις εγγραφές. Η συνθήκη αναζήτησης πρέπει να είναι μια συνθήκη ισότητας σε ένα πεδίο που ονομάζεται **πεδίο κατακερματισμού** (hash field) του αρχείου. Η **συνάρτηση κατακερματισμού** (hash function) είναι η συνάρτηση η οποία για μια τιμή του πεδίου κατακερματισμού επιστρέφει τη διεύθυνση του μπλοκ του δίσκου στο οποίο βρίσκεται η αντίστοιχη εγγραφή. Για τις περισσότερες εγγραφές απαιτείται προσπέλαση ενός μόνο μπλοκ.

3.6.1 Εσωτερικός Κατακερματισμός

Ο **εσωτερικός κατακερματισμός** υλοπποιείται με τη χρήση ενός πίνακα εγγραφών. Ο δείκτης του πίνακα παίρνει τιμές από 0 έως M-1, έχοντας δηλαδή M θέσεις. Η συνάρτηση κατακερματισμού επιλέγεται έτσι ώστε να μετασχηματίζει την τιμή του πεδίου κατακερματισμού σε ένα ακέραιο μεταξυ 0 και M-1. Μια κοινή συνάρτηση κατακερματισμού που χρησιμοποιείται είναι η: **h(key) = Key mod M**, η οποία επιστρέφει το υπόλοιπο της διαίρεσης με το M ενός ακέραιου πεδίου κατακερματισμού με τιμή Key. Η τιμή αυτή είναι η διεύθυνση της εγγραφής. Για μη ακέραια πεδία μπορεί να χρησιμοποιηθεί στο μετασχηματισμό ο κωδικός ASCII κάθε

χαρακτήρα, όπως π.χ. σε μία συμβολοσειρά. Άλλες τεχνικές κατακερματισμού είναι πέρα από τους σκοπούς αυτής της εργασίας.



Εικόνα 3.5 *(α)Πίνακας Μ θέσεων για χρήση στον εσωτερικό κατακερματισμό. (β) επίλυση συγκρούσεων με αλυσιδωτή σύνδεση των εγγραφών*

Το πρόβλημα με τις περισσότερες τεχνικές κατακερματισμού είναι ότι δεν εξασφαλίζουν ότι διαφορετικές τιμές θα κατακερματιστούν σε διαφορετικές διευθύνσεις, διότι ο **χώρος του πεδίου κατακερματισμού** (hash field space) είναι συνήθως κατά πολύ μεγαλύτερος από το χώρο των διευθύνσεων. Το αποτέλεσμα είναι να συμβαίνουν συγκρούσεις.

Μια **σύγκρουση** (collision) συμβαίνει όταν η τιμή του πεδίου κατακερματισμού μιας νέας εγγραφής που εισάγεται, απεικονίζεται σε μια διεύθυνση όπου περιέχει μια διαφορετική εγγραφή. Επομένως, πρέπει να βρεθεί μια νέα θέση για αυτή την εγγραφή. Η διαδικασία αυτή ονομάζεται

επίλυση συγκρούσεων (collision resolution) και υπάρχουν διάφοροι μέθοδοι, όπως:

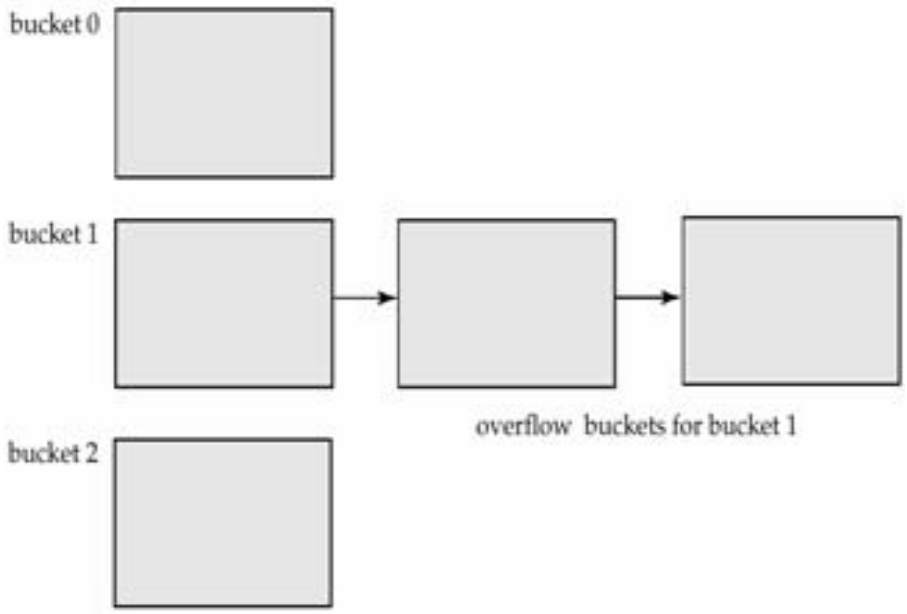
- *Ανοικτή διευθυνσιοδότηση (Open addressing)*: Το πρόγραμμα ελέγχει διαδοχικά τις επόμενες - από την κατειλημμένη – θέσεις μέχρι να βρεθεί μια αχρησιμοποίητη (κενή) θέση.
- *Αλυσιδωτή αντίδραση (Chaining)*: Στη μέθοδο αυτή διατηρούνται περιοχές θέσεων υπερχείλησης, συνήθως επεκτείνοντας τον πίνακα. Επίσης, σε κάθε θέση εγγραφής προστίθεται ένας δείκτης. Η σύγκρουση επιλύεται τοποθετώντας την νέα εγγραφή σε μία κενή θέση υπερχείλησης, με το δείκτη της κατειλημμένης θέσης να δείχνει στην διεύθυνση αυτής της θέσης υπερχείλησης.
- *Πολλαπλός κατακερματισμός (Multiple hashing)*: Το πρόγραμμα εφαρμόζει μια δεύτερη συνάρτηση κατακερματισμού αν επέλθει σύγκρουση. Αν ξανασυμβεί σύγκρουση χρησιμοποιείται ανοικτή διευθυνσιοδότηση ή μια τρίτη συνάρτηση κατακερματισμού κ.ο.κ.

Μια καλή συνάρτηση κατακερματισμού στοχεύει στην ομοιόμορφη κατανομή των εγγραφών στο χώρο διευθύνσεων, ώστε να ελαχιστοποιούνται οι συγκρούσεις χωρίς να μένουν πολλές κενές θέσεις. Συνήθως, είναι καλύτερο να διατηρείται ένας πίνακας κατακερματισμού γεμάτος σε ποσοστό περίπου 70% έως 90%, έτσι ώστε το πλήθος των συγκρούσεων να παραμένει μικρό και να μην σπαταλάμε πολύ χώρο. Αν έχουμε δηλαδή να αποθηκεύσουμε r εγραφές στον πίνακα κατακερματισμού, πρέπει να επιλέξουμε M θέσεις διευθύνσεων, έτσι ώστε ο λόγος r/M να βρίσκεται μεταξύ 0.7 και 0.9. Συγκεκριμένα, αν χρησιμοποιεέιται η mod ως συνάρτηση κατακερματισμού, είναι χρήσιμο να επιλέξουμε ως M έναν πρώτο αριθμό, καθώς έχει δειχθεί ότι κατανέμει καλύτερα τις διευθύνσεις κατακερματισμού στο χώρο διευθύνσεων.

3.6.2 Εξωτερικός Κατακερματισμός

Ο **εξωτερικός κατακερματισμός** (external hashing) πραγματοποιείται για αρχεία δίσκου. Για να ταιριάζει με τα χαρακτηριστικά αποθήκευσης στο δίσκο, ο χώρος των διευθύνσεων αποτελείται από **κάδους** (buckets), οι οποίοι χωρούν πολλές εγγραφές ο καθένας. Ένας κάδος αντιστοιχεί σε ένα μπλοκ στο δίσκο ή σε μια συστάδα συνεχόμενων μπλοκ. Εδώ, η συνάρτηση κατακερματισμού απεικονίζει ένα κλειδί στο σχετικό αριθμό ενός κάδου, αντί να εκχωρεί την απόλυτη διεύθυνση ενός μπλοκ στον κάδο. Ο αριθμός του κάδου μετατρέπεται στην αντίστοιχη διεύθυνση μπλοκ από έναν πίνακα που διατηρείται στην επικεφαλίδα του αρχείου.

Οι συγκρούσεις στον εξωτερικό κατακερματισμό είναι λιγότερο συχνές, διότι όσες εγγραφές χωρούν σε έναν κάδο μπορούν να απεικονισθούν σε ένα κάδο χωρίς προβλήματα. Ωστόσο, στην περίπτωση που ένας κάδος γεμίσει μέχρι την χωρητικότητά του και κατακερματίζεται μια νέα εγγραφή, χρησιμοποιείται μια παραλλαγή της αλυσιδωτής σύνδεσης, στην οποία διατηρείται σε κάθε κάδο ένας δείκτης προς μια συνδεδεμένη λίστα από εγγραφές υπερχείλισης για τον κάδο. Οι δείκτες στην λίστα πρέπει να είναι *δείκτες εγγραφών* που περιλαμβάνουν αφενός μια διεύθυνση μπλοκ και αφετέρου τη σχετική θέση μιας εγγραφής μέσα στο μπλοκ.

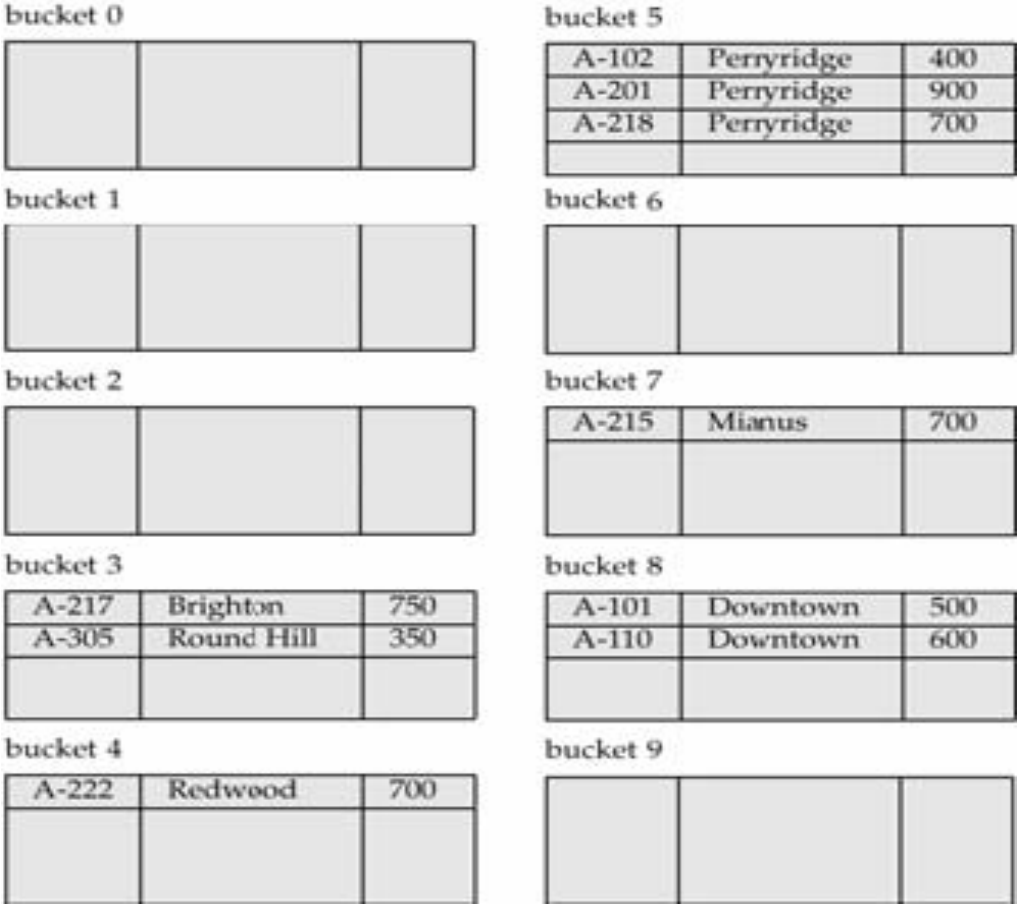


Εικόνα 3.6 *Επίλυση συγκρούσεων με αλυσιδωτή αντίδραση*

Ο κατακερματισμός παρέχει τη γρηγορότερη δυνατή προσπέλαση για ανάκτηση μιας τυχαίας εγγραφής με δεδομένη την τιμή του πεδίου κατακερματισμού της. Οι περισσότερες καλές συναρτήσεις κατακερματισμού δεν διατηρούν τη διάταξη των τιμών στο πεδίο κατακερματισμού, παρά μόνο μερικές συναρτήσεις που ονομάζονται **συναρτήσεις διατήρησης της διάταξης** (order preserving).

Το σχήμα κατακερματισμού που περιγράψαμε ονομάζεται **στατικός κατακερματισμός** επειδή έχουμε έναν σταθερό αριθμό κάδων M. Στην περίπτωση που έχουμε m εγγραφές που χωράνε στον κάθε κάδο, τότε στον συνολικό χώρο που διατίθεται θα χωρούν M*m εγγραφές. Αν το πλήθος των εγγραφών είναι σημαντικά μικρότερο, τότε θα έχουμε πολύ αχρησιμοποίητο χώρο. Στην αντίθετη περίπτωση, θα προκληθούν πολλές συγκρούσεις και η ανάκτηση μιας εγγραφής θα επιβραδυνθεί σημαντικά λόγω της εμφάνισης

μεγάλων συνδεδεμένων λιστών υπερχείλισης. Αυτό είναι ένα σοβαρό μειονέκτημα για δυναμικά αρχεία.



Εικόνα 3.7 *Παράδειγμα στατικού κατακερματισμού.*

Για την αντιμετώπιση του στατικού κατακερματισμού υπάρχουν τεχνικές κατακερματισμού που επιτρέπουν δυναμική επέκταση ή συρρίκνωση του αρχείου. Ο **επεκτατός κατακερματισμός** (extensible hashing) περιλαμβάνει εκτός από το το αρχείο και μια δομή προσπέλασης (κατάλογος) και είναι περίπου παρόμοιος με την ευρετηριοποίηση. Η κύρια διαφορά είναι ότι η δομή προσπέλασης βασίζεται στις τιμές που προκύπτουν μετά την εφαρμογή της συνάρτησης κατακερματισμού στο πεδίο αναζήτησης. Ο **γραμμικός κατακερματισμός** (linear hashing) δεν απαιτεί τη χρήση οποιωνδήποτε πρόσθετων δομών προσπέλασης. Η βασική ιδέα είναι να επιτρέπεται σε ένα αρχείο κατακερματισμού να επεκτείνει ή να συρρικνώνει τους κάδους του δυναμικά χωρίς να χρειάζεται κάποιον κατάλογο.

ΚΕΦΑΛΑΙΟ 4

Δομές Ευρετηρίων Για Αρχεία

4.1 Εισαγωγή

Τα **ευρετήρια** (index files) είναι επιπλέον βοηθητικές δομές προσπέλασης ενός αρχείου εγγραφών. Αποτελούν δευτερεύοντες δρόμους αναζήτησης σε αρχεία εγγραφών, που παρέχουν εναλλακτικούς τρόπους προσπέλασης χωρίς να επηρεάζουν την φυσική τοποθέτηση των εγγραφών στο δίσκο. Η αποτελεσματική προσπέλαση επιτρέπεται μέσω του **πεδίου ευρετηριοποίησης** (indexing field) που χρησιμοποιείται στην κατασκευή του ευρετηρίου. Το ευρετήριο περιέχει το πεδίο ευρετηριοποίησης και μια λίστα δεικτών προς τα μπλοκ που περιέχουν εγγραφές με την τιμή αυτή στο

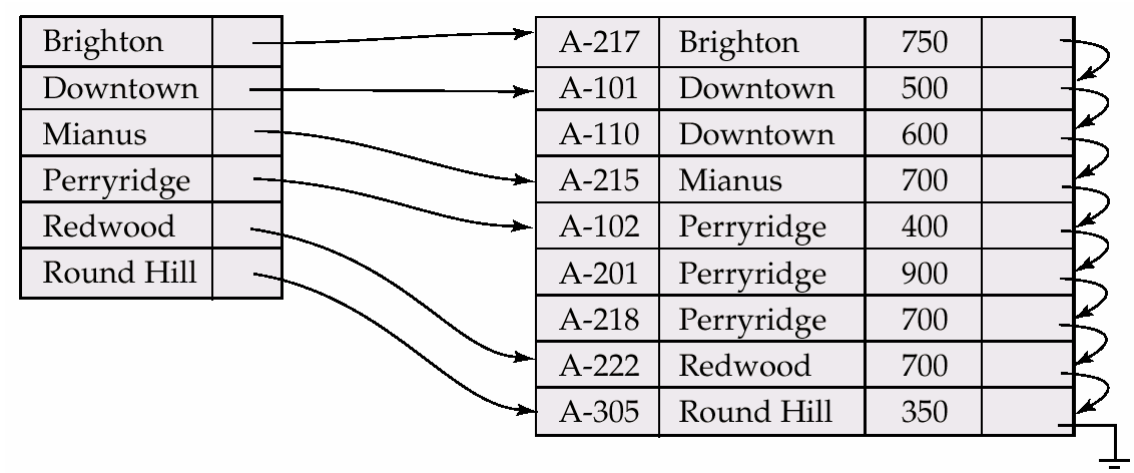
αντίστοιχο πεδίο. Οι τιμές στο ευρετήριο είναι διατεταγμένες για να μπορέσουμε να κάνουμε δυαδική αναζήτηση (εκτός από τα πολυεπίπεδα ευρετήρια). Τα ευρετήρια συνήθως είναι αρκετά μικρότερα από το αρχείο που δεικτοδοτούν.

Τιμή Γνωρίσματος	***Δείκτης στο Μπλοκ της εγγραφής***

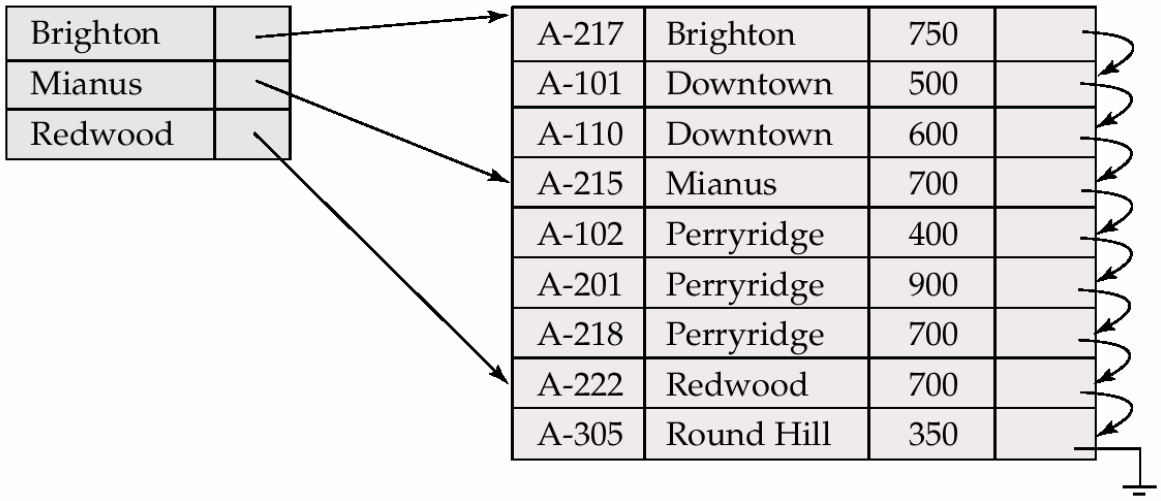
Εικόνα 4.1 *Μορφή καταχώρησης ευρετηρίου.*

Υπάρχουν δύο βασικά είδη ευρετηρίων. Τα **διατεταγμένα ευρετήρια**, τα οποία διατηρούν τη διάταξη των τιμών του πεδίου διάταξης και τα **ευρετηρία κατακερματισμού**, τα οποία έχουν ένα μηχανισμό αντίστοιχο με αυτόν στα αρχεία κατακερματισμού.

Στα διατεταγμένα ευρετήρια διακρίνουμε διάφορους τύπους ευρετηρίων. Το **πρωτεύον ευρετήριο** (primary index) είναι ορισμένο επί του πεδίου κλειδιού διάταξης ενός διατεταγμένου αρχείου εγγραφών. Αν το πεδίο διάταξης δεν είναι πεδίο-κλειδί του αρχείου μπορεί να χρησιμοποιηθεί το **ευρετήριο συστάδων** (clustering index). Ένας τρίτος τύπος ευρετηρίου είναι το **δευτερεύον ευρετήριο** (secondary index) που μπορεί να οριστεί σε κάθε πεδίο του αρχείου που δεν είναι πεδίο διάταξης. Τα παραπάνω αποτελούν τα ευρετήρια ενός επιπέδου. Υπάρχουν επίσης τα **πολυεπίπεδα ευρετήρια** (multilevel index) με τα **Β-Δένδρα** και **B^+-Δένδρα** να ξεχωρίζουν. Τέλος, μια ευρύτερη κατηγορία διαφοροποίησης των ευρετηρίων είναι το αν είναι **πυκνά** (dense index), δηλαδή να περιέχουν μια εγγραφή ανά εγγραφή του αρχείου, ή αν είναι **μη-πυκνά** (sparse index), δηλαδή να περιέχουν μια εγγραφή ανά μπλοκ.



Εικόνα 4.2 *Παράδειγμα **πυκνού** ευρετηρίου.*



Εικόνα 4.3 *Παράδειγμα **μη-πυκνού** ευρετηρίου.*

4.2 Διατεταγμένα Ευρετήρια Ενός Επιπέδου

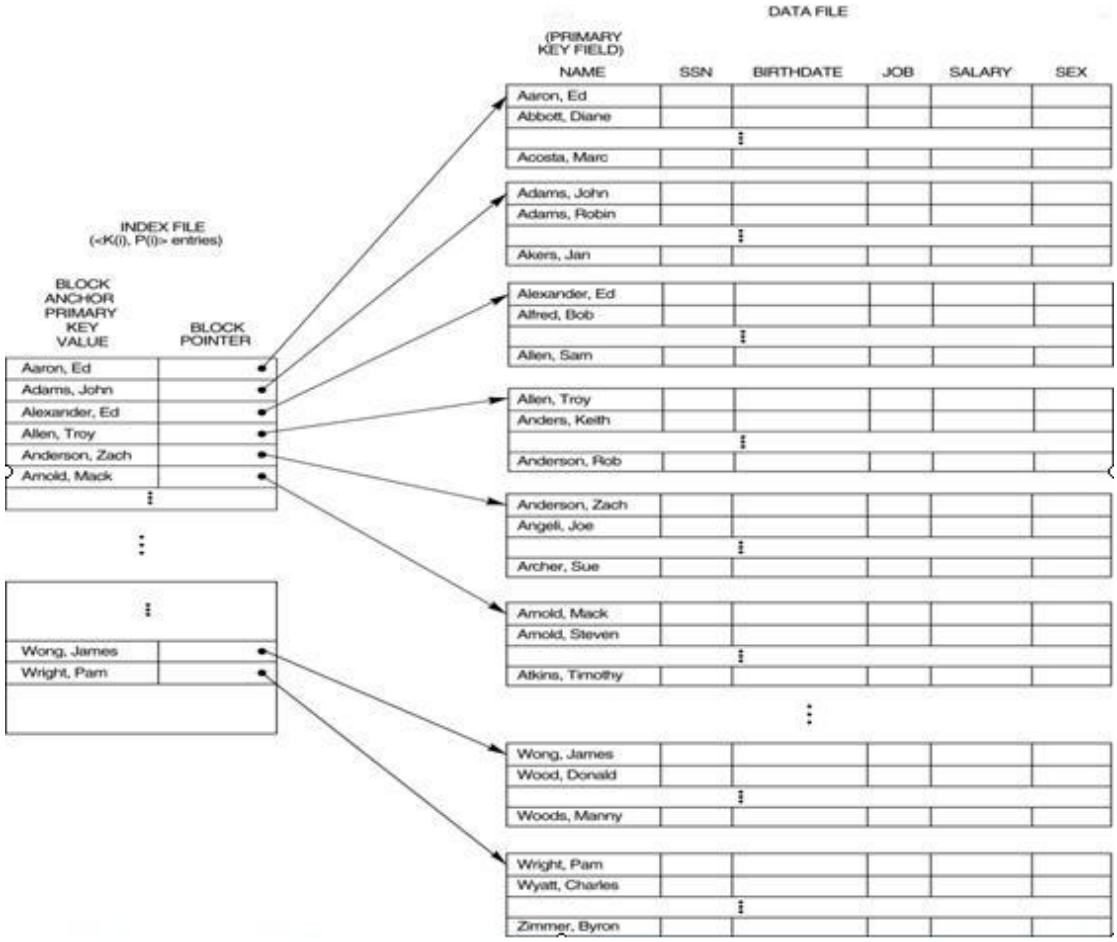
4.2.1 Πρωτεύον Ευρετήριο

Το **πρωτεύον ευρετήριο** ορίζεται σε ένα αρχείο που είναι διατεταγμένο στο (κύριο) **πρωτεύον κλειδί** (primary key) του αρχείου δεδομένων και περιλαμβάνει μια καταχώρηση ή **εγγραφή ευρετηρίου** (index record) για κάθε μπλοκ. Η καταχώρηση έχει την τιμή του κλειδιού της πρώτης εγγραφής στο μπλοκ (συχνά ονομάζεται μη-πυκνό ευρετήριο - sparse index ή non-dense index). Οι εγγραφές του ευρετηρίου έχουν σταθερό μήκος και δύο πεδία. Το πρώτο πεδίο είναι του ίδιου τύπου με το πεδίο του κλειδιού διάταξης και το δεύτερο είναι ένας δείκτης προς ένα μπλοκ δίσκου – μια διεύθυνση μπλοκ.

Ο συνολικός αριθμός καταχωρήσεων στο ευρετήριο είναι ίδιος με τον αριθμό των μπλοκ δίσκου του διατεταγμένου αρχείου. Η πρώτη εγγραφή σε κάθε μπλοκ ονομάζεται **άγκυρα του μπλοκ** (block anchor). Έτσι, το πρωτεύον ευρετήριο απαιτεί λιγότερα μπλοκ από το αρχείο δεδομένων διότι περιέχει λιγότερες εγγραφές από το αρχείο δεδομένων και διότι κάθε καταχώρηση του ευρετηρίου είναι συνήθως μικρότερη σε μέγεθος από μια εγγραφή του αρχείου δεδομένων επειδή έχει μόνο δύο πεδία.

Επομένως, μια δυαδική αναζήτηση στο αρχείο ευρετηρίου απαιτεί λιγότερες προσπελάσεις μπλοκ από μια δυαδική αναζήτηση στο αρχείο δεδομένων. Αν το αρχείο του πρωτεύοντος ευρετηρίου περιέχει $\mathbf{b_i}$ μπλοκ, για

τον εντοπισμό μιας εγγραφής με μια τιμή κλειδιού αναζήτησης απαιτείται μια δυαδική αναζήτηση αυτού του ευρετηρίου και μια επιπλέον προσπέλαση στο μπλοκ που περιέχει την εγγραφή. Άρα, συνολικά έχουμε $\log_2 b_i + 1$ προσπελάσεις.



Εικόνα 4.4 *Πρωτεύον ευρετήριο.*

Ένα σημαντικό πρόβλημα που παρουσιάζεται στο πρωτεύον ευρετήριο (όπως και σε κάθε διατεταγμένο ευρετήριο) είναι η εισαγωγή και η διαγραφή εγγραφών. Αν προσπαθήσουμε να εισάγουμε μια νέα εγγραφή στη σωστή της θέση, πρέπει να μετακινήσουμε εγγραφές για να δημιουργηθεί χώρος για τη νέα, καθώς και να αλλάξουμε και κάποιες καταχωρήσεις του ευρετηρίου διότι θα αλλάξουν και οι άγκυρες των μπλοκ. Μπορούμε να χρησιμοποιήσουμε ένα αρχείο υπερχείλισης ή μια συνδεδεμένη λίστα εγγραφών υπερχείλισης για κάθε μπλοκ. Οι εγγραφές μέσα σε κάθε μπλοκ και στη συνδεδεμένη λίστα υπερχείλισης μπορούν να ταξινομηθούν για να βελτιωθεί ο χρόνος ανάκτησης. Η διαγραφή εγγραφών αντιμετωπίζεται με τη χρήση σημαδιών διαγραφής.

4.2.2 Ευρετήρια Συστάδων

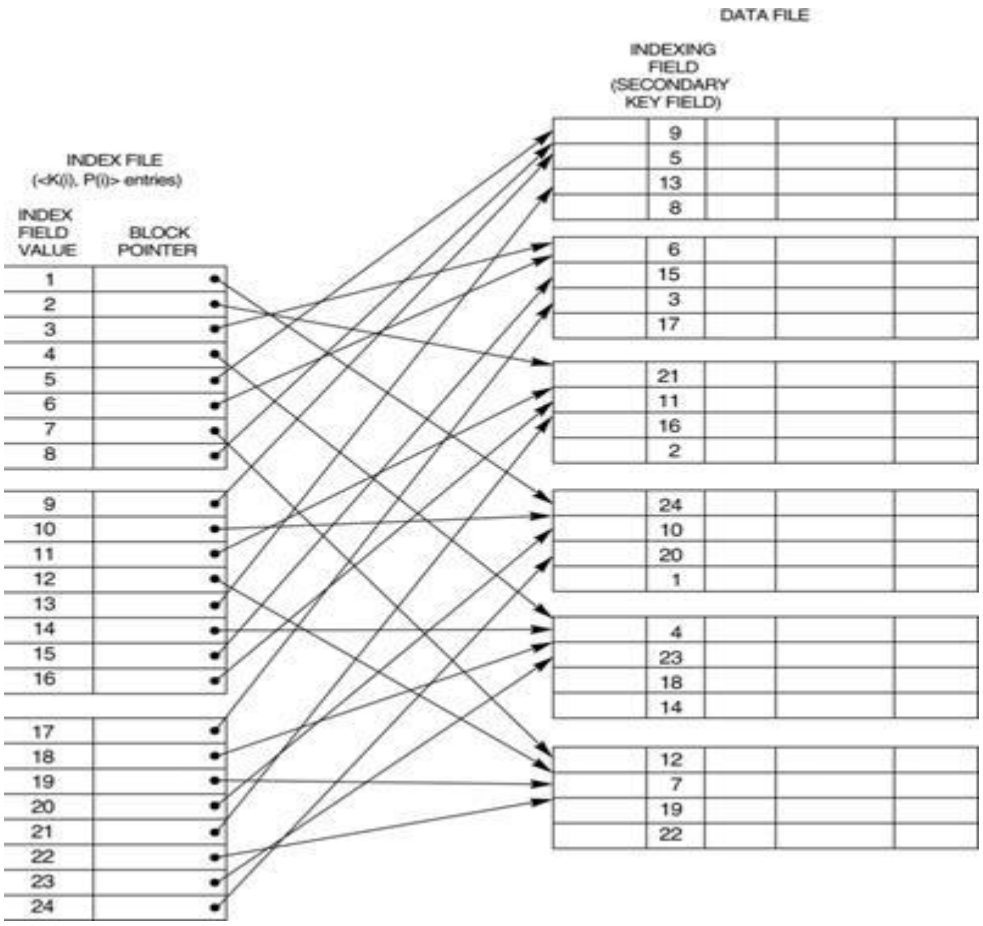
Το **ευρετήριο συστάδων** (clustering index) ορίζεται σε ένα αρχείο που είναι φυσικά διατεταγμένο σε ένα γνώρισμα που δεν είναι κλειδί, δηλαδή δεν έχει διαφορετική τιμή για κάθε εγγραφή, το **πεδίο συστάδας** (clustering field). Ένα ευρετήριο συστάδων είναι και αυτό ένα διατεταγμένο αρχείο δεδομένων με δύο πεδία. Το πρώτο είναι ίδιου τύπου με αυτόν του πεδίου συστάδας και το δεύτερο είναι ένας δείκτης μπλοκ. Υπάρχει μια καταχώρηση για κάθε ξεχωριστή τιμή του γνωρίσματος, η οποία «δείχνει» το πρώτο μπλοκ που περιέχει εγγραφές με αυτή την τιμή γνωρίσματος.

Το ευρετήριο συστάδων είναι ακόμα ένα παράδειγμα μη-πυκνού ευρετηρίου, διότι περιέχει μια καταχώρηση για κάθε διακεκριμένη τιμή του πεδίου ευρετηριοποίησης, αντί για κάθε εγγραφή του αρχείου. Υπάρχει, όμως, κι εδώ το πρόβλημα της εισαγωγής και διαγραφής εγγραφών, επειδή οι εγγραφές είναι φυσικά διατεταγμένες. Για να αντιμετωπιστεί το πρόβλημα της εισαγωγής, αποθηκεύεται ένα ολόκληρο μπλοκ ή μια ακόμα και μια συστάδα μπλοκ για κάθε τιμή του πεδίου συστάδας και τοποθετούμε όλες τις εγγραφές με αυτή την τιμή στο συγκεκριμένο μπλοκ ή την συστάδα μπλοκ.

4.2.3 Δευτερεύον Ευρετήριο

Το **δευτερεύον ευρετήριο** (secondary index) ορίζεται σε άλλα πεδία εκτός του πρωτεύοντος κλειδιού. Μπορεί να είναι σε ένα πεδίο του αρχείου που είναι υποψήφιο κλειδί και έχει μοναδική τιμή σε κάθε εγγραφή ή σε ένα πεδίο που δεν είναι κλειδί και έχει διπλότυπες τιμές. Αν το πεδίο έχει διαφορετικές τιμές σε κάθε εγγραφή λέγεται **δευτερεύον κλειδί**. Το δευτερεύον ευρετήριο είναι και αυτό φυσικά διατεταγμένο και περιλαμβάνει δύο πεδία. Το πρώτο πεδίο είναι του ίδιου τύπου με αυτόν με κάποιο πεδίο από το αρχείο δεδομένων *που δεν είναι πεδίο διάταξης* και το δεύτερο είναι είτε ένας δείκτης μπλοκ είτε ένας δείκτης εγγραφής. Για το ίδιο αρχείο μπορούν να υπάρχουν πολλά δευτερεύοντα ευρετήρια.

Στο δευτερεύον ευρετήριο περιλαμβάνεται μια καταχώρηση για κάθε εγγραφή του αρχείου δεδομένων (συχνά ονομάζεται πυκνό ευρετήριο - dense index). Εφόσον οι καταχωρήσεις του ευρετηρίου είναι διατεταγμένες, μπορούμε να πραγματοποιήσουμε δυαδική αναζήτηση στο ευρετήριο. Φυσικό επόμενο είναι ο χρόνος αναζήτησης να είναι μεγαλύτερος για ένα δευτερεύον ευρετήριο, όπως επίσης και ο χώρος αποθήκευσης εφόσον περιέχει μια καταχώρηση για κάθε εγγραφή. Συγκεκριμένα, αν έχουμε ένα αρχείο με **r** εγγραφές, **b** μπλοκ και παράγοντα ομαδοποίησης **bfr** (για το δευτερεύον ευρετήριο), ο αριθμός των μπλοκ που απαιτούνται είναι $\mathbf{log_2 b_i + 1}$ με $\mathbf{b_{i=}r/bfr}$.



Εικόνα 4.5 Πυκνό *δευτερεύον ευρετήριο.*

Στην περίπτωση όπου το πεδίο ευρετηριοποίησης έχει επαναλαμβανόμενες τιμές, τότε χρησιμοποιούμε ευρετήριο δύο επιπέδων. Έτσι, έχουμε ένα αραιό ευρετήριο με μια εγγραφή ανά διακριτή τιμή του κλειδιού και ένα ενδιάμεσο επίπεδο με τις διευθύνσεις όλων των εγγραφών. Η ανάκτηση μέσω του ευρετηρίου απαιτεί μια πρόσθετη προσπέλαση λόγω του επιπλέον ενδιάμεσου επιπέδου, αλλά οι αλγόριθμοι αναζήτησης του ευρετηρίου και εισαγωγής νέων εγγραφών του αρχείου μπορούν αν γραφούν πολύ απλά. Επίσης, σημειώνουμε ότι το δευτερεύον ευρετήριο παρέχει μια λογική διάταξη των εγγραφών με βάση το πεδίο ευρετηριοποίησης. Δηλαδή, αν προσπελάσουμε τις εγγραφές κατά τη διάταξη των καταχωρήσεων του δευτερεύοντος ευρετηρίου είναι σαν να τις διαβάζουμε κατά τη διάταξη του πεδίου ευρετηριοποίησης.

4.3 Ευρετήρια Πολλών Επιπέδων

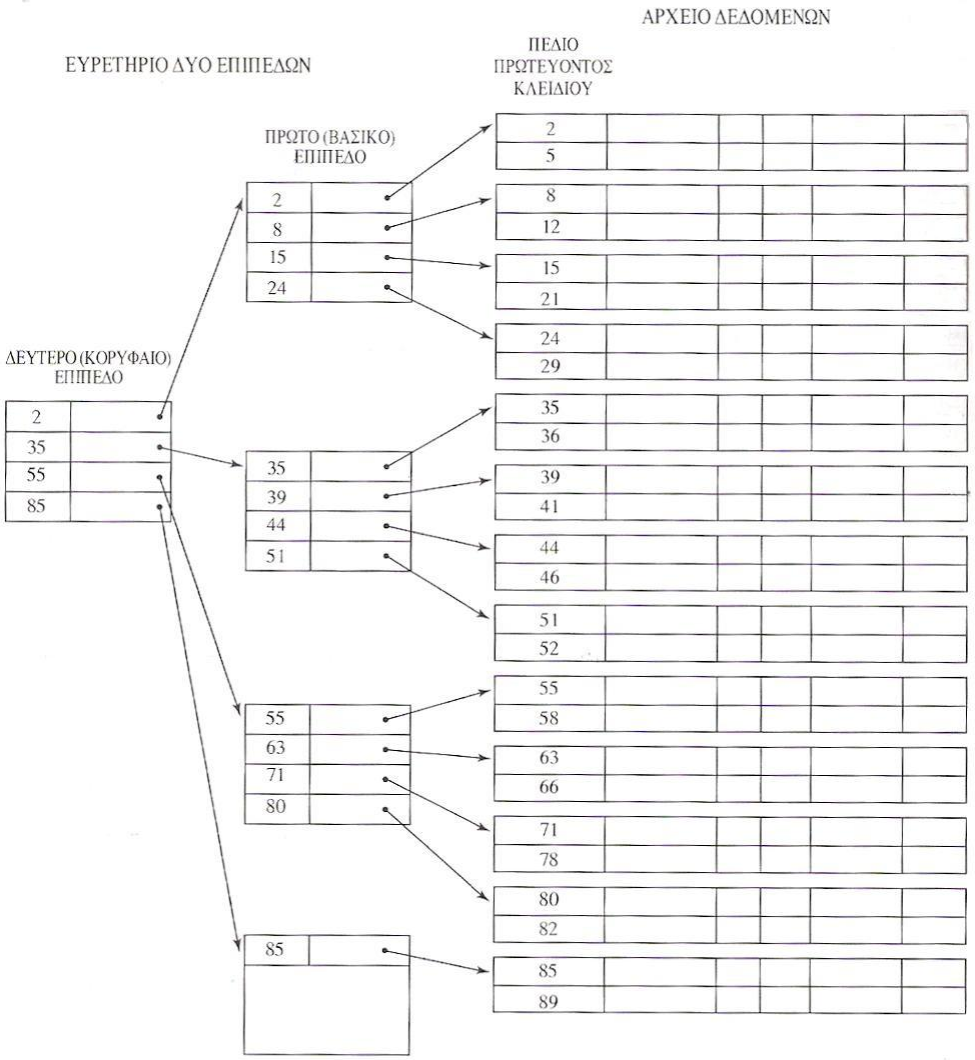
Η βασική ιδέα των **πολυεπίπεδων ευρετηρίων** (multilevel index) είναι να ελλατωθεί το μέγεθος του ευρετηρίου όπου κάνουμε αναζήτηση κατά έναν παράγοντα bfr_i, δηλαδή κατά τον παράγοντα ομαδοποίησης του ευρετηρίου που είναι μεγαλύτερος του 2. Άρα, ο χώρος αναζήτησης μειώνεται γρηγορότερα και συνεπώς και ο χρόνος (σε αριθμό προσπελάσεων μπλοκ). Η τιμή του bfr_i ονομάζεται **παράγοντας διακλάδωσης** (fan-out – fo) και η αναζήτηση απαιτεί κατά προσέγγιση $\mathbf{log_{fo}b_i}$ προσπελάσεις μπλοκ, ένας αριθμός πολύ μικρότερος από αυτόν της δυαδικής αναζήτησης, με την προϋπόθεση ότι ο παράγοντας διακλάδωσης είναι μεγαλύτερος του 2.

Το αρχείο ευρετηρίου θεωρείται ότι είναι ένα διατεταγμένο αρχείο με μια διακεκριμένη τιμή για το πεδίο διάταξης και περιέχει καταχωρήσεις σταθερού μήκους. Πλέον, θεωρείται το **πρώτο** ή **βασικό επίπεδο** (first ή base level) του πολυεπίπεδου ευρετηρίου και επομένως, μπορούμε να δημιουργήσουμε ένα πρωτεύον ευρετήριο (ή δευτερεύον ή ευρετήριο συστάδων) επί του πρώτου επιπέδου. Απαραίτητη προϋπόθεση είναι το αρχείο να είναι διατεταγμένο με διακεκριμένη τιμή για κάθε εγγραφή του πεδίου διάταξης. Εφόσον, το δεύτερο επίπεδο είναι ένα πρωτεύον ευρετήριο μπορούμε να χρησιμοποιήσουμε άγκυρες μπλοκ ώστε να περιέχει μια εγγραφή για κάθε μπλοκ του πρώτου επιπέδου. Ο παράγοντας ομαδοποίησης παραμένει σταθερός (όλες οι καταχωρήσεις έχουν το ίδιο μέγεθος) και επομένως αν το πρώτο επίπεδο έχει $\mathbf{r_1}$ εγγραφές, το δεύτερο θα έχει $\mathbf{r_2 = \lceil r_1/fo \rceil}$.

Επαναλαμβάνοντας την παραπάνω διαδικασία μπορούμε να κατασκευάσουμε και ένα τρίτο επίπεδο το οποίο θα έχει ως καταχωρήσεις τις άγκυρες μπλοκ του δεύτερου επιπέδου και συγκεκριμένα οι εγγραφές του θα είναι $\mathbf{r_3 = \lceil r_2/fo \rceil}$. Η διαδικασία αυτή τελειώνει όταν οι εγγραφές ενός επιπέδου **t** χωρέσουν ολόκληρες σε ένα μπλοκ, δηλαδή το τελευταίο επίπεδο θα περιέχει (ως εγγραφές) τις άγκυρες μπλοκ του προηγουμένου επιπέδου σε ένα μόνο μπλοκ. Αυτό το μπλοκ στο τελευταίο επίπεδο **t** λέγεται **κορυφαίο επιπεδο** (top level) του ευρετηρίου. Σε κάθε επίπεδο ελαττώνονται οι καταχωρήσεις κατά ένα παράγοντα fo, επομένως για να υπολογίσουμε το t μπορούμε να χρησιμοποιήσουμε τον τύπο $\mathbf{1 \leq (r_1/(fo)^t)}$. Έτσι, ένα πολυεπίπεδο ευρετήριο με $\mathbf{r_1}$ καταχωρήσεις στο πρώτο επίπεδο θα έχει κατά προσέγγιση t επίπεδα, όπου $\mathbf{t = \lceil log_{fo} r_1 \rceil}$ και ο αριθμός των προσπελάσεων είναι t+1 (η επιπλέον προσπέλαση είναι για το αρχείο δεδομένων).

Μια συνηθισμένη οργάνωση αρχείων που χρησιμοποιείται στην επεξεργασία δεδομένων οργανισμών είναι ένα ταξινομημένο αρχείο με πολυεπίπεδο πρωτεύον ευρετήριο στο πεδίο κλειδί διάταξης. Μια τέτοια οργάνωση ονομάζεται **σειριακό αρχείο με ευρετήριο** (indexed sequential

file) και χρησιμοποιήθηκε σε πολλά συστήματα της IBM. Οι εισαγωγές αντιμετωπίζονται με κάποια μορφή αρχείου υπερχείλισης που περιοδικά συνενοούνται με το αρχείο των δεδομένων. Κατά την αναδιοργάνωση του αρχείου δημιουργείται το ευρετήριο.



Εικόνα 4.6 *Πρωτεύον ευρετήριο δύο επιπέδων (θυμίζει ISAM).*

Η σειριακή μέθοδος προσπέλασης με ευρετήριο (Indexed Sequential Access Method – **ISAM**) είναι μια πολύ-επίπεδη δομή ευρετηρίου (δέντρο) για άμεση πρόσβαση σε εγγραφές αρχείου διατεταγμένου στο πεδίο κλειδί. Κάθε κόμβος του Δέντρου είναι ένα μπλοκ στο δίσκο. Οι κόμβοι του δέντρου κρατούν < τιμή κλειδιού, δείκτης > ζεύγη, ταξινομημένα στην τιμή κλειδιού. Οι εσωτερικοί κόμβοι «δείχνουν» σε χαμηλότερου επιπέδου κόμβους ενώ τα φύλλα-κόμβοι «δείχνουν» σε μπλοκ του αρχείου. Ένας δείκτης «δείχνει» ένα υπο-δέντρο με τιμές κλειδιού *μεγαλύτερες* ή *ίσες* της αντίστοιχης τιμής κλειδιού και *μικρότερες* της τιμής κλειδιού του επόμενου δείκτη. Αν

υποθέσουμε ότι έχουμε D μπλοκ για τα δεδομένα και k δείκτες για κάθε κόμβο (θεωρούμε $D=k^L$), τότε η διαδικασία αναζήτησης απαιτεί $\mathbf{log_k (D) + 1 = L + 1}$ προσπελάσεις.

Τα πλεονεκτήματα του ISAM είναι:

- Παρέχει έναν **ταξινομημένο κατάλογο** για το αρχείο.
- Αποτελεί εξαιρετική δομή για **ακριβείς ερωτήσεις (exact queries)** π.χ., *Salary = 400000.*
- Το ISAM διευκολύνει την εκτέλεση των **ερωτήσεων διακύμανσης** (**range queries).** π.χ., *Salary μεταξύ 350000 και 600000.*

Τα μειονεκτήματα:

- Είναι μια **στατική** δομή που εύκολα χάνει την «ισορροπία» της (**unbalanced) -** *ισορροπία ύψους και πληρότητας των κόμβων.*
- Αν στο αρχείο γίνονται πολλές ενημερώσεις (volatile δεδομένα), τότε το αρχείο μπορεί να χάσει την ταξινόμησή του.
- Το ευρετήριο απαιτεί μεγάλο χώρο στο δίσκο.

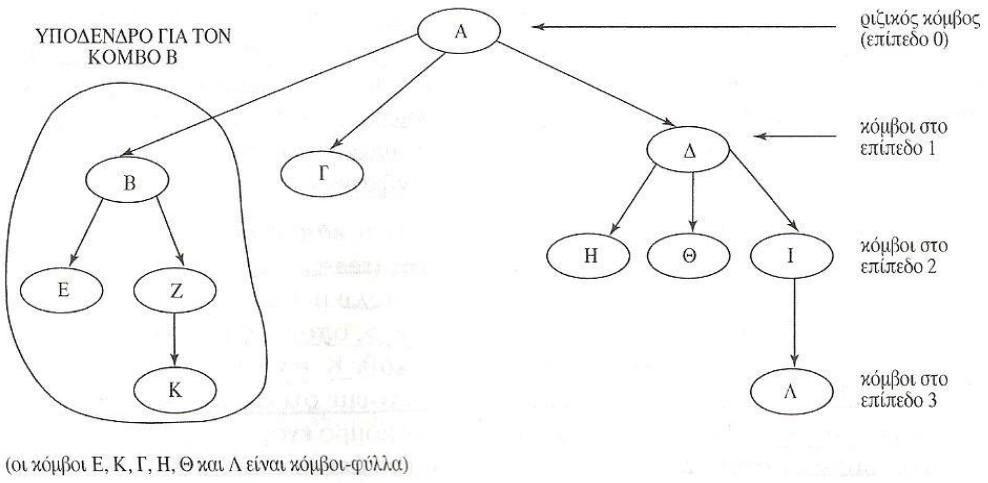
4.4 Δυναμικά Πολυεπίπεδα Ευρετήρια Με Τη Χρήση B-Δένδρων Και B^+-Δένδρων

4.4.1 Δένδρα Αναζήτησης

Ο σκοπός των **δυναμικών πολυεπίπεδων ευρετηρίων** (dynamic multilevel index) είναι να επιλύσει τα προβλήματα που δημιουργούνται και εδώ από την εισαγωγή και τη διαγραφή εγγραφών στο ευρετήριο, καθώς όλα τα επίπεδα του ευρετηρίου είναι φυσικά διατεταγμένα αρχεία. Αυτή η οργάνωση αφήνει κάποιο χώρο σε κάθε μπλοκ για την εισαγωγή νέων εγγραφών και συχνά υλοποιείται με τη χρήση B-δένδρων και B^+-δένδρων.

Ένα **δένδρο** (tree) σχηματίζεται από **κόμβους** (nodes), όπου κάθε κόμβος έχει έναν **κόμβο-γονέα** (parent node) – εκτός από την **ρίζα** (root) εφόσον είναι ο αρχικός κόμβος του δένδρου – και μηδέν ή περισσότερους **κόμβους-παιδιά** (child nodes). Ένας κόμβος που δεν έχει παιδιά ονομάζεται **κόμβος-φύλλο** (leaf ή terminal node), ενώ ένας μη τερματικός κόμβος λέγεται **εσωτερικός κόμβος** (internal node). Το επίπεδο ενός κόμβου είναι πάντα κατά ένα μεγαλύτερο από το επίπεδο του γονέα του και το επίπεδο της ρίζας είναι πάντα μηδέν. Ένα **υποδένδρο** (subtree) αποτελείται από έναν

κόμβο και όλους τους απογόνους του, δηλαδή από όλους τους κόμβους-παιδιά του.

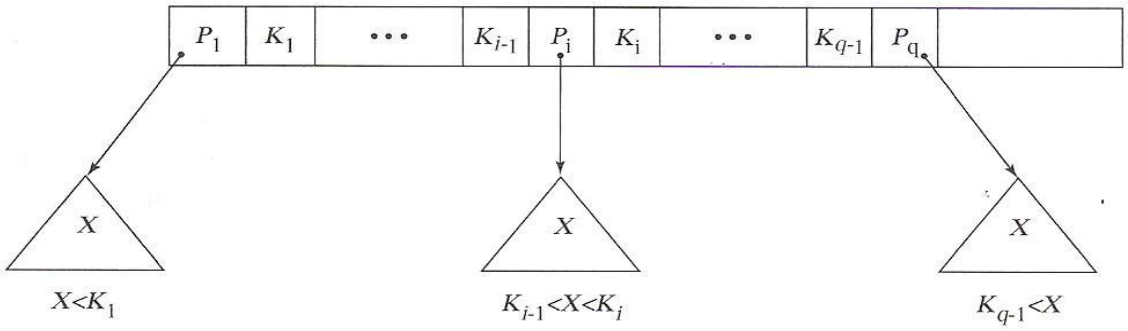


Εικόνα 4.7 *Παράδειγμα δενδρικής δομής.*

Σε κάθε κόμβο έχουμε τόσους δείκτες όσοι και οι κόμβοι-παιδιά του και μερικές φορές αποθηκεύεται και ένας δείκτης προς το γονέα του. Εκτός από τους δείκτες, ένας κόμβος περιέχει συνήθως και πληροφορίες. Όταν ένα πολυεπίπεδο ευρετήριο υλοποιείται ως δενδρική δομή, αυτές οι πληροφορίες περιλαμβάνουν τις τιμές του πεδίου ευρετηριοποίησης του αρχείου δεδομένων που χρησιμοποιούνται για να καθοδηγήσουν την αναζήτηση μιας συγκεκριμένης εγγραφής.

Ένα **δένδρο αναζήτησης** (search tree) τάξης p είναι ένα δένδρο όπου κάθε κόμβος του περιέχει το πολύ p-1 τιμές αναζήτησης και p δείκτες σε διάταξη $<P_1,K_1, P_2,K_2,\ldots, P_{q-1},K_{q-1}, P_q>$ με $q \leq p$. Κάθε P_i είναι δείκτης προς ένα κόμβο-παιδί και κάθε K_i είναι μια τιμή αναζήτησης από κάποιο διατεταγμένο σύνολο τιμών. Υποθέτουμε τους δύο παρακάτω περιορισμούς:

- Σε κάθε κόμβο ισχύει $K_1 < K_2 <\ldots< K_{q-1}$
- Για όλες τις τιμές X στο υποδένδρο που δείχνεται από το P_i έχουμε $K_{i-1}<X<K_i$ για $1<i<q$, $X<K_i$ για $i=1$ και $K_{i-1}<X$ για $i=q$.



Εικόνα 4.8 *Κόμβος δένδρου αναζήτησης.*

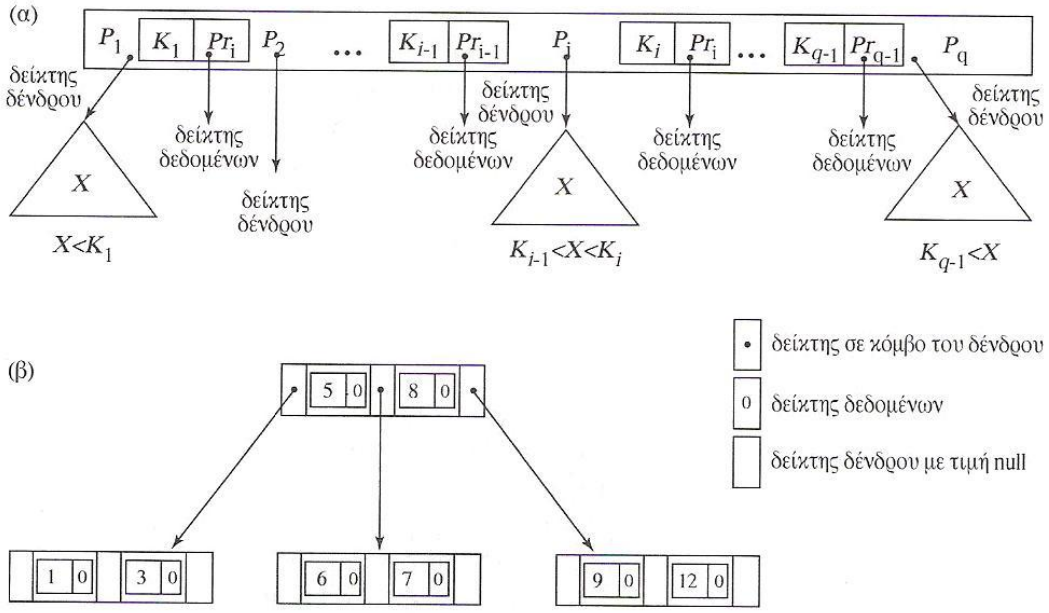
Ένα δένδρο αναζήτησης μπορεί να χρησιμοποιηθεί ως μηχανισμός αναζήτησης εγγραφών σε ένα αρχείο στο δίσκο. Οι τιμές στο δένδρο θα είναι οι τιμές από ένα πεδίο του αρχείου, που λέγεται πεδίο αναζήτησης και κάθε τιμή στο δένδρο συνδέεται με το αρχείο με ένα δείκτη προς την εγγραφή που περιέχει αυτή την τιμή στο αρχείο δεδομένων ή προς το μπλοκ του δίσκου που περιέχει την εγγραφή αυτή. Το δένδρο αποθηκεύεται στο δίσκο τοποθετώντας κάθε κόμβο του σε ένα μπλοκ του δίσκου. Στην εισαγωγή μιας νέας εγγραφής ενημερώνουμε το δένδρο με την τιμή της νέας εγγραφής καθώς και με τον δείκτη που δείχνει σε αυτήν.

Στα δένδρα αναζήτησης επανέρχεται το πρόβλημα της εισαγωγής και διαγραφής εγγραφών, καθώς δεν εξασφαλίζεται ότι το δένδρο θα είναι **ισοζυγισμένο** (balanced), δηλαδή, να βρίσκονται όλοι οι κόμβοι-φύλλα στο ίδιο επίπεδο. Ένα ισοζυγισμένο δένδρο εξασφαλίζει ότι δεν θα υπάρχουν κόμβοι σε πολύ υψηλά επίπεδα οι οποίοι θα απαιτούν πολλές προσπελάσεις μπλοκ κατά την αναζήτηση. Επίσης, στην διαδικασία της διαγραφής μπορεί να μείνουν κάποιοι κόμβοι σχεδόν κενοί, και συνεπώς να σπαταλάται πολύς χώρος αποθήκευσης και να αυξάνεται το πλήθος των επιπέδων.

4.4.2 Β-Δένδρα

Ένα **Β-Δένδρο τάξεως p** (B-tree of order p) χρησιμοποιείται ως δομή προσπέλασης σε ένα πεδίο-κλειδί για αναζήτηση εγγραφών σε ένα αρχείο δεδομένων, παρουσιάζοντας όμως περισσότερους περιορισμούς που εξασφαλίζουν την ισορροπία του δένδρου. Συγκεκριμένα, οι αλγόριθμοι εισαγωγής και διαγραφής γίνονται περισσότερο πολύπλοκοι αν και είναι σχετικά απλές διαδικασίες. Μεγαλύτερη σημασία έχουν πολύπλοκες διαδικασίες όπως η εισαγωγή σε ένα γεμάτο κόμβο ή η διαγραφή από έναν κόμβο που θα μείνει κατά λιγότερο από το ήμισυ πλήρης. Οι περιορισμοί για ένα Β-Δένδρο είναι οι εξής:

- Κάθε εσωτερικός κόμβος του δένδρου είναι της μορφής $<P_1,<K_1,P_{r1}>,P_2,<K_2,P_{r2}>,\ldots, <K_{q-1}, P_{q-1}>, P_q>$, όπου $q \leq p$. κάθε P_i είναι ένας **δείκτης δένδρου** (tree pointer) κάθε P_{ri} είναι ένας **δείκτης δεδομένων** (data pointer).
- Σε κάθε κόμβο ισχύει $K_1 < K_2 <\ldots< K_{q-1}$.
- Για όλες τις τιμές X στο υποδένδρο που δείχνεται από το P_i έχουμε $K_{i-1}<X<K_i$ για $1<i<q$, $X<K_i$ για $i=1$ και $K_{i-1}<X$ για $i=q$.
- Κάθε κόμβος έχει το πολύ p δείκτες δένδρου.
- Κάθε κόμβος, εκτός της ρίζας και των κόμβων-φύλλων, έχει τουλάχιστον $\lceil (p/2) \rceil$ δείκτες δένδρου. Ο κόμβος της ρίζας έχει τουλάχιστον δύο δείκτες δένδρου εκτός αν είναι ο μόνος κόμβος του δένδρου.
- Ένας κόμβος με q δείκτες δένδρου, όπου $q \leq p$, περιέχει q-1 τιμές πεδίου κλειδιού αναζήτησης (και επομένως περιέχει q-1 δείκτες δεδομένων).
- Όλοι οι κόμβοι-φύλλα βρίσκονται στο ίδιο επίπεδο. Οι κόμβοι-φύλλα έχουν την ίδια δομή με τους εσωτερικούς κόμβους εκτός από το ότι όλοι οι δείκτες δένδρου P_i είναι null.



Εικόνα 4.9 *Δομή Β-δένδρου. (α) Κόμβος ενός Β-δένδρου με q-1 τιμές αναζήτησης. (β) Ένα δένδρο τάξης p=3.(σειρά εισαγωγής τιμών: 8,5,1,7,3,12,9,6)*

Στην εισαγωγή στοιχείων στο δένδρο ξεκινάμε πάντα από τον κόμβο ρίζα (που είναι και κόμβος-φύλλο) στο επίπεδο 0. Όταν γεμίσει ο κόμβος της

ρίζας με p-1 τιμές του κλειδιού αναζήτησης, τότε διασπάται σε δύο κόμβους στο επίπεδο 1, όπου εισάγεται η επόμενη τιμή. Η ίδια διαδικασία επαναλαμβάνεται όταν ένας άλλος κόμβος από τη ρίζα είναι γεμάτος και γίνεται μια καταχώρηση σε αυτόν. Τότε, διασπάται και αυτός σε δύο κόμβους στο ίδιο επίπεδο και η μεσαία καταχώρηση μεταφέρεται στο κόμβο γονέα μαζί με δύο δείκτες που δείχνουν τους κόμβους από τους οποίους προήλθαν. Αν χρειαστεί να διασπαστεί και ο κόμβος-γονέας, τότε η διάσπαση μπορεί να συνεχιστεί μέχρι και την ρίζα, δημιουργώντας ένα νέο επίπεδο κάθε φορά που διασπάται η ρίζα.

Κατά τη διαγραφή μιας τιμής, ένας κόμβος μπορεί να μείνει κατά λιγότερο από το ήμισυ γεμάτος. Στην περίπτωση αυτή ενώνεται με τους γειτονικούς κόμβους και αυτή η ένωση μπορεί να συνεχιστεί μέχρι την ρίζα. Επομένως, με την διαγραφή μπορεί να ελαττωθούν τα επίπεδα του δένδρου. Σύμφωνα με προσομοιώσεις και αναλύσεις που έχουν γίνει, έχει δειχτεί ότι μετά από τυχαίες εισαγωγές και διαγραφές που έχουν γίνει σε ένα δένδρο, οι κόμβοι παραμένουν περίπου κατά 69% γεμάτοι, όταν ο αριθμός των τιμών σταθεροποιηθεί. Έτσι, σπάνια θα εμφανίζονται διασπάσεις και ενώσεις κόμβουν και η εισαγωγή και διαγραφή θα γίνουν πολύ πιο αποδοτικές.

4.4.3 B^+-Δένδρα

Τα B^+-Δένδρα και οι παραλλαγές τους είναι η κυρίως δομή που χρησιμοποιούν τα εμπορικά συστήματα διαχείρισης βάσεων δεδομένων. Σε αντίθεση με τα B-Δένδρα, όπου κάθε τιμή του πεδίου αναζήτησης αποθηκεύεται μια φορά σε κάποιο επίπεδο του δένδρου, στα B^+-Δένδρα οι δείκτες δεδομένων αποθηκεύονται μόνο στα φύλλα. Επομένως, οι κόμβοι-φύλλα διαφέρουν από τους εσωτερικούς κόμβους.

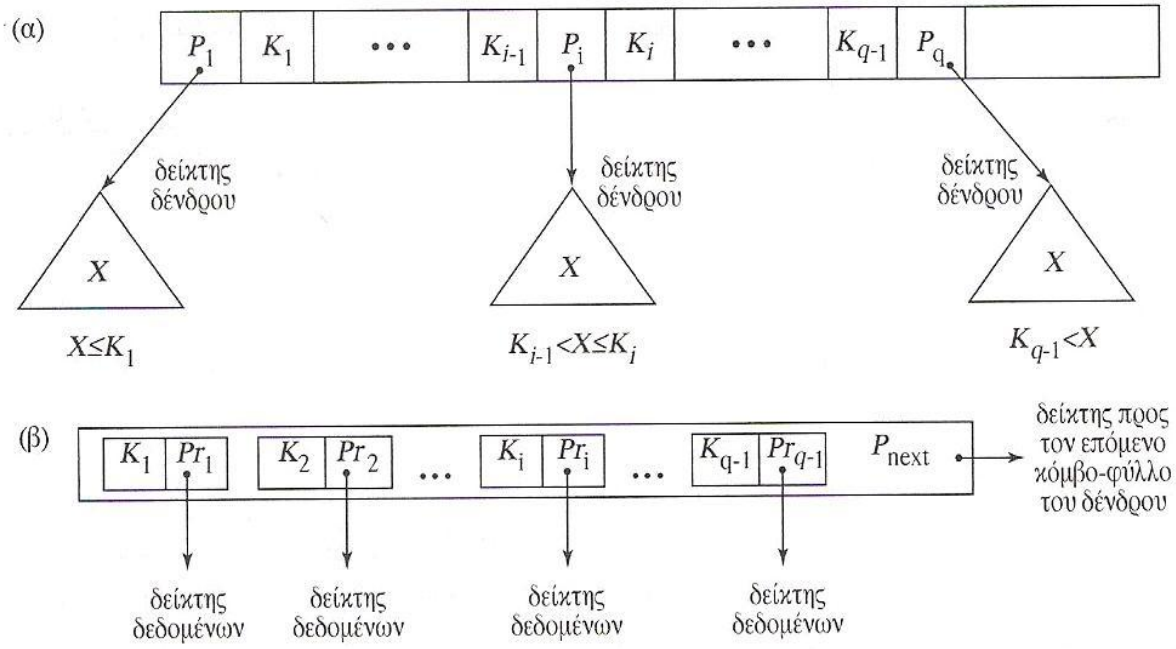
Συνήθως, οι κόμβοι-φύλλα του B^+-Δένδρου συνδέονται μεταξύ τους, ώστε να παράσχουν διατεταγμένη ως προς το πεδίο αναζήτησης προσπέλαση των εγγραφών και είναι παρόμοιοι με το πρώτο επίπεδο ενός ευρετηρίου. Οι εσωτερικοί κόμβοι αντιστοιχούν στα άλλα επίπεδα ενός πολυεπίπεδου ευρετηρίου. Μερικές τιμές του πεδίου αναζήτησης επαναλαμβάνονται στους εσωτερικούς κόμβους για να καθοδηγούν την αναζήτηση. Η δομή για τους εσωτερικούς κόμβους έχει ως εξής:

- Κάθε εσωτερικός κόμβος είναι της μορφής $<P_1,K_1,P_2,K_2,\ldots,K_{q-1},P_q>$ $q \leq p$ και κάθε P_i είναι δείκτης δένδρου.
- Σε κάθε εσωτερικό κόμβο ισχύει $K_1 < K_2 <\ldots< K_{q-1}$.

- Για όλες τις τιμές X στο υποδένδρο που δείχνεται από το P_i έχουμε $K_{i-1}<X<K_i$ για $1<i<q$, $X<K_i$ για $i=1$ και $K_{i-1}<X$ για $i=q$.
- Κάθε εσωτερικός κόμβος έχει το πολύ p δείκτες δένδρου.
- Κάθε εσωτερικός κόμβος εκτός από τη ρίζα έχει το λιγότερο $\lceil (p/2) \rceil$ δείκτες δένδρου. Ο κόμβος της ρίζας έχει τουλάχιστον δυο δείκτες δένδρου.
- Ένας εσωτερικός κόμβος με q δείκτες, $q \leq p$, έχει q-1 τιμές πεδίου αναζήτησης

Η δομή των κόμβων-φύλλων έχει ως εξής:

- Κάθε κόμβος-φύλλο είναι της μορφής: $<<K_1,P_{r1}>,<K_2,P_{r2}>,\ldots,<K_{q-1},Pr_{q-1}>,P_{next}>$, όπου $q \leq p$, κάθε Pr_i είναι δείκτης δεδομένων και το P_{next} δείχνει τον επόμενο κόμβο-φύλλο του B+δένδρου.
- Σε κάθε κόμβο ισχύει $K_1 < K_2 <\ldots< K_{q-1}$, $q \leq p$.
- Κάθε Pr_i είναι ένας **δείκτης δεδομένων** που δείχνει την εγγραφή της οποίας η τιμή του πεδίου του ευρετηρίου είναι K_i.
- Κάθε κόμβος-φύλλο έχει το λιγότερο $\lceil p/2 \rceil$ τιμές.
- Όλοι οι κόμβοι-φύλλα είναι στο ίδιο επίπεδο.



Εικόνα 4.10 *Δομή B^+-δένδρου. (α) Εσωτερικός κόμβος ενός B^+-δένδρου με q-1 τιμές αναζήτησης. (β) Κόμβος-φύλλο ενός B^+-δένδρου.*

Οι δείκτες των εσωτερικών κόμβων είναι *δείκτες δένδρου* προς μπλοκ που είναι κόμβοι δένδρου, ενώ οι δείκτες των κόμβων-φύλλων είναι *δείκτες δεδομένων* προς εγγραφές ή μπλοκ δεδομένων του αρχείου (εκτός από τον δείκτη P_{next} που δείχνει στον επόμενο κόμβο-φύλλο). Έτσι, μπορούμε να σαρώσουμε όλους τους κόμβους-φύλλα σαν μια συνδεδεμένη λίστα μέσω των δεικτών P_{next}.

Επειδή, οι καταχωρήσεις στους εσωτερικούς κόμβους περιλαμβάνουν τιμές αναζήτησης και δείκτες δένδρου και όχι δείκτες δεδομένων, μπορούν να χωρέσουν περισσότερες τιμές απ' ότι γίνεται σε ένα B-Δένδρο. Επομένως, για το ίδιο μέγεθος μπλοκ, η τάξη p θα είναι μεγαλύτερη για το δένδρο B^+-Δένδρο παρά για ένα B-Δένδρο. Αυτό μπορεί να οδηγήσει σε λιγότερα επίπεδα, βελτιώνοντας έτσι τον χρόνο αναζήτησης. Χρησιμοποιούμε το p για να δηλώσουμε την τάξη του δένδρου και το p_{leaf} για να δηλώσουμε την τάξη των κόμβων-φύλλων, που ορίζεται ως ο μέγιστος αριθμός δεικτών δεδομένων σε ένα κόμβο-φύλλο.

Όταν ένας κόμβος-φύλλο είναι γεμάτος και γίνει εισαγωγή μιας καταχώρησης σε αυτόν, τότε υπερχειλίζει και πρέπει να διασπασθεί. Οι πρώτες $j=\lfloor((p_{leaf} + 1)/2)\rceil$ καταχωρήσεις του αρχικού κόμβου παραμένουν εκεί και οι υπόλοιπες καταχωρήσεις μεταφέρονται στον νέο κόμβο-φύλλο. Η j-οστή τιμή αναζήτησης επαναλαμβάνεται στον εσωτερικό κόμβο-γονέα και δημιουργείται στο γονέα ένας επιπλέον δείκτης προς το νέο κόμβο. Αν ο κόμβος-γονέα είναι και αυτός γεμάτος, τότε θα υπερχειλίσει και αυτός και θα πρέπει να διασπαστεί. Αυτή η διαδικασία μπορεί να συνεχιστεί μέχρι και το ανώτερο, όπου μπορεί να δημιουργηθεί ένας καινούργιος κόμβος-ρίζα και να αυξηθεί έτσι ο αριθμός των επιπέδων.

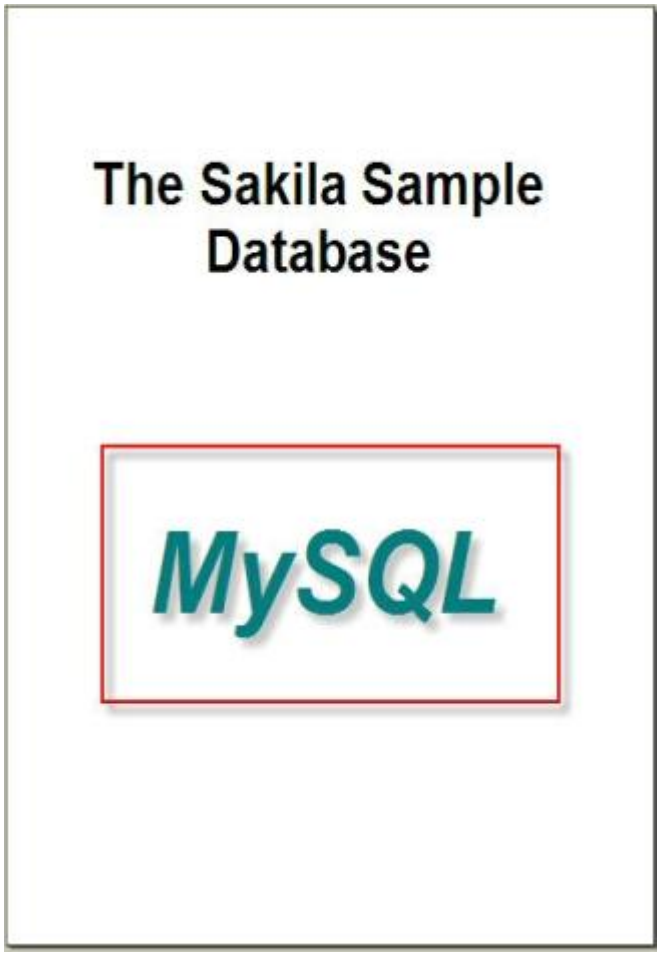
Στην διαγραφή ξεκινάμε από την ρίζα και βρίσκουμε τον κόμβο-φύλλο όπου ανήκει η τιμή που θέλουμε να διαγράψουμε. Αν ο κόμβος μετά την διαγραφή μείνει τουλάχιστον μισο-γεμάτος τότε η διαδικασία ολοκληρώνεται. Αν ο κόμβος-φύλλο δεν περιέχει αρκετές καταχωρήσεις (δηλαδή **υποχειλίζεται**), τότε προσπαθούμε να βρούμε έναν **αδελφό κόμβο-φύλλο** (ακριβώς αριστερά ή δεξιά) και να ανακατανείμουμε τις καταχωρήσεις μεταξύ του κόμβου και του αδελφού του. Αν η ανακατανομή αποτύχει τότε συγχωνεύουμε τους δύο κόμβους και έτσι το πλήθος των κομβων-φύλλων μειώνεται. Στην περίπτωση αυτή, η υποχείλιση μπορεί να διαδοθεί στους εσωτερικούς κόμβους, διότι χρειάζονται ένας δείκτης και μια τιμή αναζήτησης λιγότεροι. Αυτή η διάδοση μπορεί να οδηγήσει μέχρι την ρίζα του δένδρου, προκαλώντας ελάττωση των επιπέδων του δένδρου.

Τα πλεονεκτήματα από τη χρήση των B^+-Δένδρων είναι ότι έχουν τη δυνατότητα αυτόματης αναδιοργάνωσης με μικρές, τοπικές αλλαγές κατά τη διάρκεια εισαγωγών και διαγραφών. Για τη διατήρηση της απόδοσης δεν απαιτείται αναδιοργάνωση όλου του αρχείου. Το (βασικό) μειονέκτημα είναι

ότι υπάρχει επιπλέον κόστος I/O για την εισαγωγή και διαγραφή καθώς και επιπλέον απαίτηση χώρου. Παρ΄όλα αυτά, τα πλεονεκτήματα των B+-δένδρων υπερκαλύπτουν τα μειονεκτήματα και αυτό δικαιολογεί την ευρεία χρήση τους.

ΚΕΦΑΛΑΙΟ 5

Εφαρμογή των Ευρετηρίων σε Δειγματική Βάση Δεδομένων



5.1 Εισαγωγή

Προκειμένου να αποδειχθεί ότι, πράγματι, τα ευρετήρια είναι πολύ χρήσιμα και ιδιαίτερα στις διαδικασίες αναζήτησης, τις οποίες κάνουν αρκετά ταχύτερες, θα χρησιμοποιήσουμε μία δειγματική βάση δεδομένων. Σκοπός είναι να χρησιμοποιηθεί μία βάση με έναν σχετικά μεγάλο αριθμό εγγραφών

στους πίνακες, ώστε να είναι ξεκάθαρη η αξιοποίηση των ευρετηρίων όσον αφορά το χρόνο αναζήτησης.

Στα πλαίσια του εργαστηρίου του μαθήματος Βάσεις Δεδομένων, χρησιμοποιούνται βάσεις οι οποίες δεν προσφέρονται για εξάσκηση στο συγκεκριμένο θέμα, εφόσον ο μέγιστος αριθμός εγγραφών που αποτελούν ένα πίνακα δεν ξεπερνάει τις 250. Κύριο μέλημα είναι να εργαστούμε σε μία βάση η οποία περιέχει τουλάχιστον 1000 εγγραφές στους πίνακές τις, ώστε να είναι εμφανής η διαφορά στην χρησιμοποίηση και στη μη χρησιμοποίηση ευρετηρίων.

Στην συγκεκριμένη εργασία, θα χρησιμοποιηθεί η δειγματική βάση δεδομένων **Sakila**, η οποία παρουσιάζει την οργάνωση ενός βίντεο κλαμπ. Η βάση βρέθηκε στο διαδίκτυο και προσφέρεται ελεύθερα για σκοπούς παρόμοιους με τον σκοπό και αυτής της εργασίας.

5.2 Η Βάση Δεδομένων Sakila

Η δειγματική βάση δεδομένων Sakila αναπτύχθηκε από τον Mike Hillyer, πρώην μέλος της ομάδας τεκμηρίωσης MySQL AB, και προορίζεται να παρέχει ένα τυποποιημένο σχήμα που μπορεί να χρησιμοποιηθεί για παράδειγμα μέσα σε βιβλία, μαθήματα, άρθρα, δείγματα, κοκ.

Η δειγματική βάση δεδομένων Sakila σχεδιάστηκε ως υποκατάστατο της βάσης δεδομένων **world**, η οποία επίσης παρέχεται από την MySQL AB. Η βάση δεδομένων **world** παρέχει μια σειρά από πίνακες που περιέχουν πληροφορίες για τις χώρες και τις πόλεις του κόσμου και είναι χρήσιμη για τα βασικά ερωτήματα, αλλά δεν έχει τις δομές για τον έλεγχο των εξειδικευμένων λειτουργιών της MySQL και τα νέα χαρακτηριστικά που βρέθηκαν στην MySQL 5.

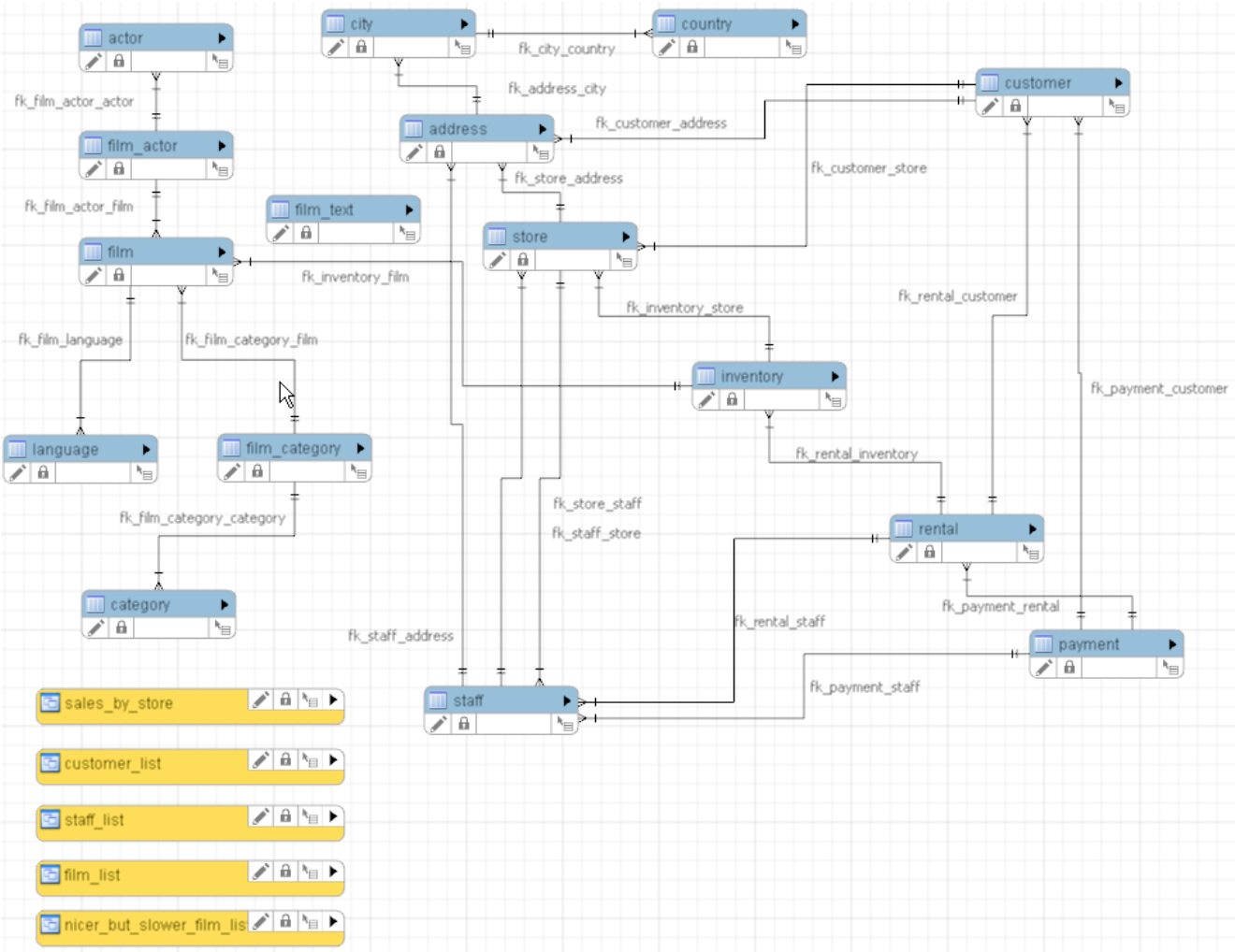
Η ανάπτυξη της βάσης δεδομένων Sakila ξεκίνησε στις αρχές του 2005. Τα πρόωρα σχέδια βασίστηκαν στη βάση δεδομένων που χρησιμοποιείται στο έργο της Dell «*Three Approaches to MySQL Applications on Dell PowerEdge Servers*». Ενώ το δείγμα βάσης δεδομένων της Dell έχει σχεδιαστεί για να αντιπροσωπεύει ένα ηλεκτρονικό κατάστημα DVD, το δείγμα βάσης δεδομένων Sakila έχει σχεδιαστεί για να αντιπροσωπεύει ένα DVD κατάστημα ενοικίασης. Το δείγμα βάσης δεδομένων Sakila δανείζεται ακόμα ονόματα ταινιών και ηθοποιών από την βάση δεδομένων της Dell.

Η ανάπτυξη ολοκληρώθηκε με το MySQL Query Browser, για το σχεδιασμό του schema, με τους πίνακες που το αποτελούν από έναν συνδυασμό του MySQL Query Browser και custom script.

Μετά την ολοκλήρωση του βασικού σχήματος, διάφορες προβολές και αποθηκευμένες ρουτίνες προστέθηκαν στο σχήμα. Τότε προστέθηκαν τα δεδομένα του δείγματος. Μετά από μια σειρά εκδόσεων επανεξέτασης, η πρώτη επίσημη έκδοση της βάσης δεδομένων του δείγματος Sakila κυκλοφόρησε τον Μάρτιο του 2006.

5.2.1 Το σχήμα της Sakila

Το διάγραμμα που ακολουθεί παρέχει μία επισκόπηση της δομής της βάσης δεδομένων Sakila.



Εικόνα 5.1 Sakila schema

5.2.2 Οι πίνακες της Sakila

Ο πίνακας **actor** περιέχει πληροφορίες για τους ηθοποιούς και συνενώνεται με τον πίνακα **film** μέσω του **film_actor**. Οι στήλες του πίνακα είναι οι εξής:

- **actor_id** (πρωτεύον κλειδί)
- **first_name**
- **last_name**

Ο πίνακας **address** περιέχει πληροφορίες για τους πελάτες, το προσωπικό και τα καταστήματα. Οι στήλες είναι οι εξής:

- **address_id** (πρωτεύον κλειδί – εμφανίζεται και ως ξένο κλειδί στους πίνακες των πελατών, του προσωπικού και των καταστημάτων)
- **address**
- **address2**
- **district**
- **city_id** (ξένο κλειδί που δείχνει στον πίνακα **city**)
- **postal_code**
- **phone**

Ο πίνακας **category** περιέχει τις κατηγορίες στις οποίες ανήκουν οι ταινίες. Οι στήλες είναι οι εξής:

- **category_id** (πρωτεύον κλειδί – ενώνεται με τον πίνακα **film** μέσω του **film_category**.)
- **name**

Ο πίνακας **city** περιέχει μία λίστα από πόλεις και αναφέρεται στον πίνακα **country** χρησιμοποιώντας ένα ξένο κλειδί. Οι στήλες είναι οι εξής:

- **city_id** (πρωτεύον κλειδί)
- **city**
- **country_id**

Ο πίνακας **country** περιέχει χώρες. Οι στήλες είναι οι εξής:

- **country_id** (πρωτεύον κλειδί)
- **country**

Ο πίνακας **customer** περιέχει τους πελάτες και αναφέρεται στους πίνακες **rental** και **payment** και ενώνεται με τους πίνακες **address** και **store** χρησιμοποιώντας ένα ξένο κλειδί. Οι στήλες είναι οι εξής:

- **customer_id** (πρωτεύον κλειδί)
- **store_id**
- **first_name**
- **last_name**
- **email**
- **address_id**
- **active**
- **create_date**

Ο πίνακας **film** περιέχει όλες τις ταινίες που βρίσκονται (υποθετικά) στα καταστήματα. Τα πραγματικά αντίγραφα βρίσκονται στο πίνακα **inventory**. Οι στήλες είναι οι εξής:

- **film_id** (πρωτεύον κλειδί)
- **title**
- **description**
- **release_year**
- **language_id**
- **original_language_id**
- **rental_duration**
- **rental_rate**
- **length**
- **replacement_cost**
- **rating** (G, PG, PG-13, R, ή NC-17)
- **special_features** (Trailers, Commentaries, Deleted Scenes, Behind the Scenes)

Ο πίνακας **film_actor** υποστηρίζει τις σχέσεις μεταξύ ηθοποιών και ταινιών. Οι στήλες είναι οι εξής:

- **actor_id**
- **film_id**

Ο πίνακας **film_category** υποστηρίζει τις σχέσεις μεταξύ ταινιών και κατηγοριών ταινιών. Ο πίνακας αναφέρεται στις ταινίες και τις κατηγορίες ταινιών χρησιμοποιώντας ξένα κλειδιά. Οι στήλες είναι οι εξής:

- **film_id**
- **category_id**

Ο πίνακας **film_text** επιτρέπει την αναζήτηση ταινιών με βάση τον τίτλο και περιέχει στήλες του πίνακα **film**. Οι στήλες είναι οι εξής:

- **film_id**
- **title**
- **description**

Τα περιεχόμενα του πίνακα δεν πρέπει να ενημερώνονται απ'ευθείας. Πρώτα ενημερώνεται ο πίνακας **film**.

Ο πίνακας **inventory** περιέχει κάθε αντίγραφο μιας ταινίας σε κάθε κατάστημα. Χρησιμοποιεί ξένα κλειδιά και σχετίζεται με τον πίνακα **rental** και τον πίνακα **store**. Οι στήλες είναι οι εξής:

- **inventory_id** (πρωτεύον κλειδί)
- **film_id**
- **store_id**

Ο πίνακας **language** περιέχει τις πιθανές γλώσσες που οι ταινίες έχουν, καθώς και την αρχική γλώσσα της ταινίας. Οι στήλες είναι οι εξής:

- **language_id** (πρωτεύον κλειδί)
- **name**

Ο πίνακας **payment** περιέχει κάθε πληρωμή που έχει γίνει από ένα πελάτη και δείχνει τους πίνακες **customer, staff** και **rental**. Οι στήλες είναι οι εξής:

- **payment_id** (πρωτεύον κλειδί)
- **customer_id**
- **staff_id**
- **rental_id**
- **amount**
- **payment_date**

Ο πίνακας **rental** περιέχει κάθε ενοικίαση που έχει γίνει για κάθε στοιχείο του πίνακα **inventory** και δείχνει τους πίνακες **inventory, customer** και **staff**. Οι στήλες είναι οι εξής:

- **rental_id** (πρωτεύον κλειδί)
- **rental_date**
- **inventory_id**
- **customer_id**
- **return_date**
- **staff_id**

Ο πίνακας **staff** περιέχει το προσωπικό των καταστημάτων και σχετίζεται με τους πίνακες **address** και **store.** Οι στήλες είναι οι εξής:

- **staff_id (**πρωτεύον κλειδί)
- **first_name**
- **last_name**
- **address_id**
- **picture**
- **email**
- **store_id**
- **active**
- **username**
- **password**

Ο πίνακας **store** περιέχει τα καταστήματα και σχετίζεται με τους πίνακες **store** και **address**. Οι στήλες είναι οι εξής:

- **store_id**
- **manager_staff_id**
- **address_id**

Στους παραπάνω πίνακες, σε κάθε πελάτη και εργαζόμενο έχει «ανατεθεί» ένα home store. Για τους πελάτες, αυτό είναι το κατάστημα στο οποίο εγγράφηκαν. Παρ’όλα αυτά μπορούν να ενοικιάσουν ταινίες από οποιοδήποτε κατάστημα.

5.3 Ερωτήματα στην MySQL

Εφόσον παρουσιάσαμε την οργάνωση της βάσης και εξετάσαμε για τον κάθε πίνακα ξεχωριστά τα γνωρίσματα που τον απαρτίζουν, μπορούμε να αναπτύξουμε κάποια ερωτήματα (queries) για να δούμε στην πράξη πως βελτιώνεται η αναζήτηση με την χρήση ευρετηρίων.

Για την περάτωση των ερωτημάτων χρησιμοποιήθηκε το ακόλουθο λογισμικό:

- Ο XAMMP 1.7.7 για την δημιουργία τοπικού server στον προσωπικό υπολογιστή.
- Η πλατφόρμα **phpmyadmin** (**http://localhost/phpmyadmin**), όπου φορτώνουμε την βάση και μπορούμε να εργαστούμε με πλήρη ελευθερία και άνεση. Το περιβάλλον είναι παρόμοιο με το περιβάλλον που χρησιμοποιείται και στο εργαστήριο του μαθήματος.

Οι γνώσεις που απαιτούνται είναι η σύνταξη ερωτημάτων **SELECT** σε συνδυασμό με τους τελεστές **GROUP BY, HAVING** και **ORDER BY**. Επίσης, τους ποσοτικούς τελεστές **COUNT, SUM, AVG, MAX** και **MIN**, καθώς και το **JOIN** που χρησιμοποείται στην συνένωση πινάκων.

Τέλος, θα πρέπει να αναφέρουμε και την σύνταξη της εντολής που προσθέτει ένα ευρετήριο στο γνώρισμα ενός πίνακα, καθώς το εργαστήριο του μαθήματος δεν καλύπτει αυτό το θεωρητικό κομμάτι. Συγκεκριμένα, υπάρχουν διαφορετικοί τρόποι για την δημιουργία ευρετηρίου, αλλά για τους σκοπούς αυτής της εργασίας θα χρησιμοποιηθεί μόνο ένας εφόσον η βάση είναι ήδη έτοιμη. Η εντολή δημιουργίας ευρετηρίου στην περίπτωση που θέλουμε να προσθέσουμε index σε ήδη υπάρχουσα στήλη είναι η εξής:

ALTER TABLE *table_name* **ADD INDEX** (*attribute_name*)

Παρακάτω παρουσιάζονται οχτώ ερωτήματα τα οποία δείχνουν ότι με την βοήθεια των ευρετηρίων ο χρόνος αναζήτησης μειώνεται αισθητά από κλίμακα χιλιοστών του δευτερολέπτου μέχρι και δεκάτων του δευτερολέπτου.

Ερώτημα 1:

Να βρεθούν ταινίες με μήκος μικρότερο των 60’ και αυτές που ξεκινάνε με το γράμμα “P” καθώς και ο αριθμός των ηθοποιών που έπαιξαν σε αυτές τις ταινίες. Να εμφανιστούν πρώτα οι ταινίες με το μικρότερο μήκος και οι τίτλοι αλφαβητικά.

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT title AS "Title", length AS "Length", COUNT(actor_id) AS "No of Actors"
FROM film
INNER JOIN film_actor ON film.film_id=film_actor.film_id
WHERE length<60 AND title>'P' AND title<'PZZZZZZZZZZZZZZZ'
GROUP BY title
ORDER BY length, title
```

Παρατηρούμε ότι για την εκτέλεση του ερωτήματος απαιτούνται 0.0082 sec. Με την δημιουργία ευρετηρίου στο γνώρισμα **title** του πίνακα **film** καθώς και στο **film_id** του **film** και του **film_actor**, ο χρόνος ελλατώνεται σε 0.0014 sec. Αν και ο χρόνος και στις δύο περιπτώσεις είναι μικρός, παρατηρείται ότι με την χρήση ευρετηρίου γίνεται περίπου έξι φορές μικρότερος. Οι εντολές που χρησιμοποιήθηκαν είναι οι:

```
ALTER TABLE film ADD INDEX (film_id)
ALTER TABLE film ADD INDEX (title)
ALTER TABLE film_actor ADD INDEX (film_id)
```

Ερώτημα 2:

Να βρεθούν οι ταινίες που έχουν replacement cost μικρότερο του 10 και υπάρχουν σε περισσότερα από 7 αντίγραφα στα καταστήματα.

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT title AS "Title", count(inventory.film_id) AS "No of Copies"
FROM film
INNER JOIN inventory ON film.film_id=inventory.film_id
WHERE replacement_cost<10
GROUP BY title
HAVING COUNT(inventory.film_id)>7
```

Και εδώ παρατηρείται μείωση του χρόνου αναζήτησης με την χρήση ευρετηρίων. Αρχικά, απαιτήθηκαν 0.0120 sec, ενώ με την χρήση ευρετηρίων 0.0035 sec. Τα ευρετήρια που προσθέσαμε είναι τα εξής:

```
ALTER TABLE film ADD INDEX (film_id)
ALTER TABLE film ADD INDEX (replacement_cost)
ALTER TABLE inventory ADD INDEX (film_id)
```

Ερώτημα 3:

Να βρεθούν οι πελάτες που πλήρωσαν πάνω από $10 και ποιες ταινίες νοικιάσανε. Οι πελάτες να εμφανιστούν αλφαβητικά.

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT payment.customer_id AS “Customer ID”, last_name AS “Last Name”, first_name AS “First Name”, amount AS “Amount”, title AS “Title”
FROM payment
INNER JOIN rental ON payment.rental_id=rental.rental_id
INNER JOIN inventory ON rental.inventory_id=inventory.inventory_id
INNER JOIN film ON inventory.film_id=film.film_id
INNER JOIN customer ON payment.customer_id=customer.customer_id
```

```
WHERE amount>10

ORDER BY title, last_name
```

Σε αυτό το ερώτημα, η διαφορά μεταξύ χρήσης ευρετηρίου και μη είναι αρκετά εμφανής και αυτό οφείλεται στο γεγονός ότι χρησιμοποιείται ένας από τους δύο μεγαλύτερους πίνακες της βάσης. Ο πίνακας **payment** έχει 16.000 περίπου θέσεις και ενδείκνυται για ερωτήματα με τη βοήθεια ευρετηρίων. Συγκεκριμένα, χωρίς την χρήση ευρετηρίου, ο χρόνος είναι 0.1086 sec, ενώ με την χρήση ευρετηρίου πέφτει στα 0.0049 sec. Χρησιμοποιήσαμε ένα μόνο ευρετήριο στο πεδίο **amount** του πίνακα **payment**:

```
ALTER TABLE payment ADD INDEX (amount)
```

Ερώτημα 4:

Να βρεθούν οι ταινίες που επιστράφηκαν διαφορετικό μήνα από το μήνα που ενοικιάστηκαν και χρεώθηκαν πάνω από $11, καθώς και οι ενοικιαστές τους. Να εμφανιστούν ημερολογιακά (οι παλαιότερες στην αρχή).

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT rental.rental_id AS “Rental ID”, last_name AS “Last Name”,
first_name AS “ First Name”, title As “Title”

FROM rental

INNER JOIN payment ON rental.rental_id=payment.rental_id

INNER JOIN customer ON rental.customer_id=customer.customer_id

INNER JOIN inventory ON rental.inventory_id=inventory.inventory_id

INNER JOIN film ON inventory.film_id=film.film_id

WHERE MONTH(rental_date) <> MONTH(return_date) AND
amount>11
```

Και εδώ, υπάρχει μεγάλη διαφορά με τη βοήθεια του ευρετηρίου, εφόσον αρχικά χρειάστηκαν 0.0729 sec και με την πρόσθεση του ευρετηρίου χρειάστηκαν 0.0015 sec. Δηλαδή, ο χρόνος έγινε περίπου 48 φορές μικρότερος, μία τιμή που δείχνει την πραγματική συμβολή των ευρετηρίων

στην μείωση του χρόνου αναζήτησης. Σε αυτό το ερώτημα χρησιμοποιήθηκε το ίδιο ευρετήριο με το προηγούμενο ερώτημα.

Ερώτημα 5:

Να βρεθούν οι πελάτες που νοίκιασαν ταινία το 2006 και πλήρωσαν πάνω από $11.

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT CONCAT(customer.last_name, ', ', customer.first_name) AS
`Name of Customer`, payment.amount,

DATE(rental.rental_date) AS `Date`

FROM customer

INNER JOIN rental ON customer.customer_id = rental.customer_id

INNER JOIN payment ON rental.customer_id = payment.customer_id

WHERE payment.amount > 11 AND YEAR(rental.rental_date) = 2006
```

Παρομοίως, με την πρόσθεση ευρετηρίου στο γνώρισμα **amount** του πίνακα **payment** παρατηρείται μείωση του χρόνου αναζήτησης από 0.425 sec σε 0.0020 sec. Συγκεκριμένα, ο χρόνος έγινε 213 περίπου φορές μικρότερος.

Ερώτημα 6:

Να βρείτε σε πόσα έργα έπαιξε κάθε ηθοποιός για τους ηθοποιούς που έπαιξαν σε πάνω από 40 έργα.

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT CONCAT(actor.last_name, ', ', actor.first_name) AS `Name of
Actor`, COUNT(actor.actor_id) AS `Num of Movies Starred`,

actor.actor_id AS “Actor ID”
```

```
FROM actor

INNER JOIN film_actor ON actor.actor_id = film_actor.actor_id

INNER JOIN film ON film_actor.film_id = film.film_id

GROUP BY `actor_id`

HAVING COUNT(`Num of Movies Starred`)>40
```

Σε αυτό το ερώτημα, παρατηρείται μείωση του χρόνου όταν προσθέτουμε το ευρετήριο, αν και μικρότερη από τα προηγούμενα ερωτήματα. Συγκεκριμένα, στην αρχή απαιτήθηκαν 0.0945 sec, ενώ όταν προσθέσαμε ένα ευρετήριο στα γνωρίσματα **film_id** και **actor_id** του πίνακα **film_actor** και στο **film_id** του πίνακα **film**, ο χρόνος αναζήτησης έγινε 0.0109 sec. Οι εντολές που χρησιμοποιήθηκαν είναι οι εξής:

```
ALTER TABLE film_actor ADD INDEX (film_id)

ALTER TABLE film_actor ADD INDEX (actor_id)

ALTER TABLE film ADD INDEX (film_id)
```

Ερώτημα 7:

Να βρεθούν ποια έργα ανήκουν στην κατηγορία Action και πληρώθηκαν τον μήνα Ιούνιο. Να εμφανιστεί και ο αριθμός πληρωμής των ταινιών αυτών τον συγκεκριμένο μήνα.

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT DISTINCT film.title AS "Title", category.name AS "Category",
COUNT(title) AS "No of Payments"

FROM category

INNER JOIN film_category ON category.category_id =
film_category.category_id

INNER JOIN film ON film_category.film_id = film.film_id

INNER JOIN inventory ON film.film_id=inventory.film_id

INNER JOIN rental ON inventory.inventory_id=rental.inventory_id
```

```
INNER JOIN payment ON rental.rental_id=payment.rental_id

WHERE category.name ="Action" AND MONTH(payment_date)=06

GROUP BY title
```

Σε αυτό το ερώτημα, η μείωση του χρόνου που παρατηρείται είναι ελάχιστη και μπορεί να θεωρηθεί αμελητέα. Συγκεκριμένα, αρχικά απαιτήθηκαν 0.0192 sec ενώ με την πρόσθεση ευρετηρίου ο χρόνος έφτασε στα 0.0175 sec. Οι εντολές που χρησιμοποιήθηκαν είναι οι εξής:

```
ALTER TABLE film ADD INDEX (film_id)

ALTER TABLE inventory ADD INDEX (film_id)

ALTER TABLE inventory ADD INDEX (inventory_id)

ALTER TABLE rental ADD INDEX (inventory_id)

ALTER TABLE rental ADD INDEX (rental_id)

ALTER TABLE payment ADD INDEX (rental_id)
```

Ερώτημα 8:

Να βρεθούν ποιοι πελάτες έχουν νοικιάσει έργα για πάνω από 6 ημέρες και το επώνυμό τους ξεκινάει από «Μ».

Η απάντηση δίνεται από το ακόλουθο query:

```
SELECT DISTINCT CONCAT(customer.last_name, ', ',
customer.first_name) AS `Name of Customer`, film.rental_duration

FROM customer

INNER JOIN rental ON customer.customer_id = rental.customer_id

INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id

INNER JOIN film ON inventory.film_id = film.film_id

WHERE film.rental_duration > 6 AND ((customer.last_name > "M") AND
(customer.last_name < "MZZZZZZZZZZZZZZZ"))
```

Στο τελευταίο ερώτημα παρατηρείται μία ακόμα (μικρή) μείωση του χρόνου αναζήτησης. Συγκεκριμένα, αρχικά χρειάστηκαν 0.0067 sec για την εκτέλεση του ερωτήματος, ενώ με την πρόσθεση των ευρετηρίων χρειάστηκαν 0.0026 sec. Όπως βλέπουμε, η μείωση στον χρόνο αναζήτησης είναι πολύ μικρή, χωρίς όμως να θεωρηθεί αμελητέα. Οι εντολές που χρησιμοποιήθηκαν είναι οι εξής:

```
ALTER TABLE customer ADD INDEX (customer_id)

ALTER TABLE rental ADD INDEX (customer_id)

ALTER TABLE rental ADD INDEX (inventory_id)

ALTER TABLE inventory ADD INDEX (inventory_id)

ALTER TABLE inventory ADD INDEX (film_id)

ALTER TABLE film ADDINDEX (film_id)

ALTER TABLE film ADD INDEX (rental_duration)

ALTER TABLE customer ADD INDEX (last_name)
```

5.4 Συμπεράσματα

Από τα παραπάνω αποτελέσματα αλλά και κατά την διάρκεια εκπόνησης αυτής της εργασίας βγήκαν κάποια πολύ χρήσιμα συμπεράσματα, όσον αφορά την χρησιμοποίηση ευρετηρίων σε μία βάση δεδομένων.

Αρχικά, η πρόσθεση ευρετηρίου σε ένα γνώρισμα ενός πίνακα γίνεται με την προϋπόθεση ότι δεν έχει δημιουργηθεί άλλο παρόμοιο ευρετήριο. Στις βάσεις δεδομένων που κυκλοφορούν, όμως, καθώς και στην βάση που χρησιμοποιήσαμε στα πλαίσια αυτής της εργασίας, για κάθε πίνακα έχουν δημιουργηθεί ευρετήρια, τουλάχιστον για τα πρωτεύοντα κλειδιά του πίνακα.

Σε πίνακες οι οποίοι έχουν ευρετήριο σε κάποιο πρωτεύον κλειδί δεν μπορούμε να αξιοποιήσουμε περισσότερο τα πλεονεκτήματα των ευρετηρίων, καθώς φτιάχνοντας κι άλλα ευρετήρια κινδυνεύουμε να αυξήσουμε τον χρόνο αναζήτησης(!). Αυτό αιτιολογείται από την προσπάθεια πρόσθεσης ευρετηρίων στον πίνακα στα γνωρίσματα που μας ενδιαφέρουν με αποτέλεσμα η τελική αναζήτηση να ξεπεράσει σε χρόνο μια απλή (σειριακή) αναζήτηση.

Επίσης, πίνακες οι οποίοι χρησιμοποιούν ξένα κλειδιά (και συνεπώς ευρετήρια σε αυτά τα γνωρίσματα) δεν μπορούν να αξιοποιήσουν παραπάνω ευρετήρια σε αυτά τα γνωρίσματα, παρά μόνο σε γνωρίσματα που δεν έχουν κάποιο ευρετήριο. Όλα τα παραπάνω ισχύουν με την προϋπόθεση ότι όταν χρησιμοποιούμε μια βάση για εκπαιδευτικό σκοπό, όπως είναι και ο σκοπός αυτής της εργασίας, δεν μπορούμε να σβήσουμε τα ήδη υπάρχοντα ευρετήρια στους πίνακες της βάσης, ώστε να πειραματιστούμε με queries χρησιμοποιώντας και μη ευρετήρια για την εκτέλεσή τους.

Συγκεκριμένα, είναι προτιμότερο να δημιουργούμε ευρετήρια για τα γνωρίσματα που αναφέρονται στη δήλωση **WHERE** ενός query, εφόσον εκεί τίθεται ένας πιθανός περιορισμός. Επίσης, και στα γνωρίσματα που περιελαμβάνονται με την εντολή **JOIN** τα ευρετήρια είναι αποτελεσματικά. Το τελευταίο ισχύει ειδικά στην περίπτωση όπου έχουμε ένα πίνακα, ο οποίος περιέχει μόνο ξένα κλειδιά και κανένα πρωτεύον κλειδί, επειδή ουσιαστικά είναι πίνακας με καταχωρήσεις μη ευρετηριοποιημένες σε κάποιο γνώρισμα. Στην βάση που χρησιμοποιούμε στην εργασία αυτή, χαρακτηριστικό παράδειγμα αποτελεί ο πίνακας **film_actor**.

Ένα άλλο σημείο στο οποίο πρέπει να δώσουμε σημασία είναι ο τύπος των γνωρισμάτων στα οποία προσθέτουμε ευρετήρια. Δηλαδή, σε γνωρίσματα τύπου *decimal* ή *varchar* μία πιθανή πρόσθεση ευρετηρίου θα ήταν πιο αποτελεσματική από την πρόσθεση ευρετηρίου σε ένα γνώρισμα τύπου *smallint*. Αυτό ισχύει, διότι οι πρώτοι τύποι γνωρισμάτων καταλαμβάνουν σχεδόν διπλάσια byte στην αποθήκευσή τους δίσκο από ένα γνώρισμα τύπου *smallint*.

Χαρακτηριστικό παράδειγμα είναι το γνώρισμα **length** στον πίνακα **film** (που αναφέρεται στο ερώτημα 1), το οποίο με μία πρόσθεση ευρετηρίου δεν δίνει ταχύτερη αναζήτηση, διότι με το μικρό μέγεθος του γνωρίσματος παρέχεται ήδη μία γρήγορη αναζήτηση. Πρόσθέτοντας ένα ευρετήριο στο γνώρισμα **title** του πίνακα **film** (το οποίο είναι τύπου varchar) παρατηρούμε ότι η αναζήτηση γίνεται ταχύτερη περίπου έξι φορές.

Τέλος, θα ήταν χρήσιμο να συνοψίσουμε τα πλεονεκτήματα και τα μειονεκτήματα που παρουσιάζονται στην χρήση των ευρετηρίων στην MySQL.

<u>Πλεονεκτήματα για τα MySQL Indexes</u>

Γενικά με το MySQL indexing σε μία βάση έχουμε τρία πλεονεκτήματα:

- *Βελτιστοποίηση των queries*: Τα index βοηθούν ώστε τα ερωτήματα αναζήτησης να εκτελούνται πιο γρήγορα.

- *Μοναδικότητα*: Τα index όπως τα primary key index βοηθούν ώστε να αποφύγεις την επανάληψη δεδομένων.
- *Αναζήτηση κειμένου*: Στα Full-text indexes οι χρήστες έχουν τη δυνατότητα να βελτιστοποιήσουν την αναζήτηση ακόμα και για μεγάλα κείμενα που έχουν τοποθετηθεί σε indexed field.

Μειονεκτήματα για τα MySQL indexes

Όταν δημιουργηθεί το index μιας στήλης η MySQL δημιουργεί ένα ξεχωριστό ταξινομημένο αρχείο που περιέχει μόνο τα πεδία που ενδιαφέρουν:

- Τα indexes καταλαμβάνουν χώρο στο δίσκο. Συνήθως δεν είναι σημαντικός χώρος αλλά λόγω του ότι δημιουργείται index σε κάθε στήλη και με κάθε δυνατό συνδιασμό το αρχείο του index θα μπορούσε γρήγορα να γίνει πιο μεγάλο από το αρχείο δεδομένων. Εάν είχαμε έναν ήδη μεγάλο πίνακα, το index θα μπορούσε να καταλάβει όλο τον διαθέσιμο χώρο στο δίσκο.
- Επίσης, τα index καθυστερούν τη διαδικασία εκτέλεσης ερωτημάτων όπως τα INSERT, UPDATE and DELETE. Αυτό συμβαίνει επειδή η MySQL πρέπει να ανανεώσει τους "δείκτες" στις γραμμές που τροποποιήθηκαν. Έτσι θυσιάζονται πόροι από το σύστημα. Ωστόσο, εάν τα ερωτήματα συνταχθούν κατάλληλα δεν θα προκαλέσουν αξιοσημείωτη μείωση της απόδοσης της βάσης δεδομένων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] R. Elmasri – S.B. Navanthe, ΘΕΜΕΛΙΩΔΕΙΣ ΑΡΧΕΣ ΣΥΣΤΗΜΑΤΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ, Εκδόσεις Δίαυλος, 2005

[2] Larry Ulman, ΕΙΣΑΓΩΓΗ ΣΤΙΣ PHP 6 & MySQL 6 ΜΕ ΕΙΚΟΝΕΣ, Εκδόσεις Κλειδάριθμος, 2009

[3] William Horton – Katherine Horton, E-LEARNING TOOLS AND TECHNOLOGIES, Εκδόσεις Wiley, Indianapolis 2003

[4] Αντώνης Λιοναράκης, FORMS IF DEMOCRACY IN EDUCATION: OPEN ACCESS AND DISTANCE EDUCATION, Εκδόσεις Προπομπός, Αθήνα 2007

[5] Νικόλαος Αβούρης – Οδυσσέας Κουφοπαύλου – Δημήτριος Σερπάνος, ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ ΥΠΟΛΟΓΙΣΤΕΣ, Εκδόσεις Typorama, Πάτρα 2004

[6] Νικόλαος Αβούρης, ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΕΠΙΚΟΙΝΩΝΙΑ ΑΝΘΡΩΠΟΥ – ΥΠΟΛΟΓΙΣΤΗ, Εκδόσεις Δίαυλος, Αθήνα 2000

ΗΛΕΚΤΡΟΝΙΚΕΣ ΠΗΓΕΣ

[7] http://www.tsakalidis.gr/interviews/talks9.html (Συνέντευξη Ιωάννη Σ. Χριστοδούλου)

[8] http://dide.flo.sch.gr/Plinet/plinet.html (Κέντρο ΠΛΗΝΕΤ Ν. Φλώρινας)

[9] http://www.wikipedia.com

[10] http://www.cs.uoi.gr/ (Τμήμα Πληροφορικής Πανεπιστημίου Ιωαννίνων)

[11] http://dev.mysql.com/doc/index-other.html (Sample database Sakila)