

Κεφάλαιο 1

Εισαγωγή

Στο κεφάλαιο αυτό παρουσιάζεται μια συνοπτική εισαγωγή στη θεωρία των βάσεων δεδομένων και των συστημάτων διαχείρισης βάσεων δεδομένων, προκειμένου ο αναγνώστης να κατανοήσει τις βασικές αρχές λειτουργίας τους και τους λόγους για τους οποίους αυτά χρησιμοποιούνται

Η ΕΝΝΟΙΑ ΤΟΥ ΔΕΔΟΜΕΝΟΥ ΚΑΙ ΤΟΥ ΤΥΠΟΥ ΔΕΔΟΜΕΝΩΝ

Όπως είναι σε όλους μας γνωστό, ο άνθρωπος κατά τη διάρκεια της ζωής του βρίσκεται σε μια συνεχή επικοινωνία με το περιβάλλον του. Η επικοινωνία αυτή μπορεί να γίνει με πολλούς τρόπους, όπως είναι η ομιλία, ο γραπτός λόγος και κυρίως τα αισθητήρια όργανά του. Το κύριο χαρακτηριστικό αυτής της επικοινωνίας, είναι η ανταλλαγή πληροφοριών που λαμβάνει χώρα ανάμεσα στον κάθε άνθρωπο και το χώρο μέσα στον οποίο ζει και κινείται. Ο άνθρωπος είναι τόσο πομπός όσο και αποδέκτης της πληροφορίας, δηλαδή, μπορεί τόσο να μεταδώσει, όσο και να λάβει πληροφορίες. Οι πληροφορίες αυτές μπορεί να είναι οποιασδήποτε μορφής, και να αφορούν σκέψεις, περιγραφές, ή ακόμη και συναισθήματα. Μπορούν να παραμένουν οι ίδιες ή να αλλάζουν με την πάροδο του χρόνου. Σε όλες όμως τις περιπτώσεις έχουν ένα κοινό χαρακτηριστικό: μεταφέρουν κάποιο περιεχόμενο. Το περιεχόμενο αυτό, εξαρτάται άμεσα από το φυσικό σύστημα στο οποίο αναφέρεται: μπορεί να είναι η αξία κάποιας μετοχής στο χρηματιστήριο, το αποτέλεσμα μιας ποδοσφαιρικής αναμέτρησης, η θερμοκρασία ενός καταψύκτη, αλλά και το όνομα κάποιου φίλου μας, ο αριθμός της τηλεφωνικής μας σύνδεσης, και η χωρητικότητα του σκληρού δίσκου που βρίσκεται στον υπολογιστή μας.

Από τα όσα αναφέρονται παραπάνω, είναι προφανές πως η πληροφορία αποτελεί απαραίτητο συστατικό κάθε φυσικού συστήματος. Στην επιστήμη της Πληροφορικής, όλες αυτές οι πληροφορίες χαρακτηρίζονται ως **δεδομένα (data)**. Τα δεδομένα λοιπόν μπορούν να οριστούν ως **τρόποι αναπαράστασης εννοιών και γεγονότων που δύνανται να υποστούν διαχείριση και επεξεργασία από τον άνθρωπο, ή από ένα αυτοματοποιημένο υπολογιστικό σύστημα**. Αυτοί οι τρόποι αναπαράστασης, συσχετίζονται άμεσα με τη φύση της πληροφορίας που μεταφέρεται από τα δεδομένα.

Στην επιστήμη της Πληροφορικής, τα δεδομένα αναπαρίστανται και μοντελοποιούνται με εντελώς συγκεκριμένο τρόπο. Είναι σε όλους μας γνωστό πως

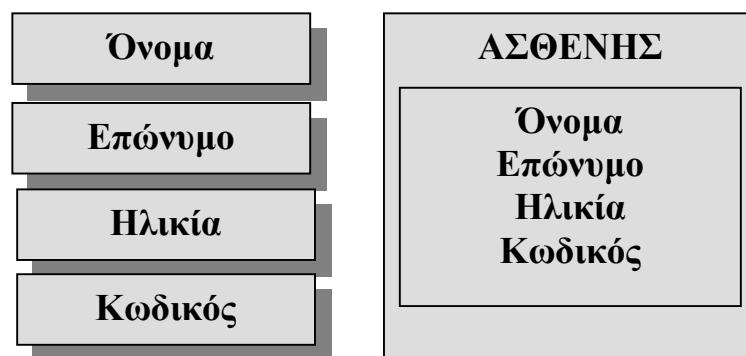
το αριθμητικό σύστημα που χρησιμοποιούν οι ηλεκτρονικοί υπολογιστές, είναι το **δυαδικό αριθμητικό σύστημα**. Στο σύστημα αυτό υπάρχουν μόνο δύο αριθμοί, το **0** και το **1**. Ο κάθε ένας από αυτούς τους αριθμούς, αποτελεί τη στοιχειώδη ποσότητα πληροφορίας που μπορεί να αποθηκευτεί στον υπολογιστή, και ονομάζεται **bit**. Η αμέσως μεγαλύτερη μονάδα πληροφορίας, είναι το **byte**. Ένα byte είναι ίσο με **8 bits**. Πώς όμως όλα αυτά συσχετίζονται με την αναπαράσταση των δεδομένων στον υπολογιστή? Πολύ απλά, με το να ορίσουμε **ειδικούς τύπους δεδομένων**, οι οποίοι θα περιγράφουν τα δεδομένα μας με ένα και μοναδικό τρόπο. Εάν ξαναγυρίσουμε στα παραδείγματα των δεδομένων που αναφέραμε στην προηγούμενη παράγραφο, δεν είναι δύσκολο να καταλάβουμε, ότι εφόσον η αξία της μετοχής στο χρηματιστήριο είναι ένας **αριθμός**, θα πρέπει να χρησιμοποιήσουμε ένα τύπο δεδομένων που να περιγράφει τους αριθμούς. Ένας αριθμός όμως μπορεί να είναι **ακέραιος, δεκαδικός**, ή ακόμη και **μικαδικός**, οπότε θα πρέπει να ορίσουμε αντίστοιχους τύπους δεδομένων. Το όνομα του φίλου μας είναι μία **λέξη**, πρέπει συνεπώς να υπάρχει κάποιος τύπος δεδομένων που να περιγράφει τις λέξεις, κ.ο.κ. Διαπιστώνουμε επομένως, ότι **υπάρχει μια αντιστοιχία ανάμεσα στα δεδομένα του πραγματικού μας κόσμου, και στην αναπαράστασή τους, στον ηλεκτρονικό υπολογιστή**.

Οι πιο συνηθισμένοι τύποι δεδομένων που χρησιμοποιούνται στα σύγχρονα πληροφοριακά συστήματα, είναι οι ακόλουθοι:

Ονομασία	Περιγραφή	Μέγεθος σε bytes
Integer	Ακέραιοι αριθμοί	2
Real	Πραγματικοί αριθμοί	4
Char	Χαρακτήρες	1
String	Λέξεις και φράσεις	Μεταβλητό
Date	Ημερομηνία	8

Πίνακας 1: Τύποι δεδομένων που χρησιμοποιούνται στους Ηλεκτρονικούς Υπολογιστές

Σύνθετοι τύποι δεδομένων: Είναι προφανές, πως τα όσα έχουν αναφερθεί μέχρι τώρα, αφορούν **στοιχειώδεις μορφές δεδομένων**. Τα δεδομένα αυτά συσχετίζονται με μια και μοναδική τιμή, και δεν μπορούν να αναλυθούν σε μικρότερες μονάδες, με κανένα τρόπο. Στις πιο συνηθισμένες όμως περιπτώσεις, τα δεδομένα υφίστανται ευκολότερη και αποτελεσματικότερη διαχείριση εάν ομαδοποιηθούν σε μεγαλύτερες οντότητες, τις οποίες διαπραγματευόμαστε όπως ακριβώς και τα απλά δεδομένα. Ας πάρουμε για παράδειγμα την περίπτωση ενός νοσοκομείου. Εάν θελήσουμε να δημιουργήσουμε ένα πρόγραμμα στο οποίο θα καταχωρούμε τα στοιχεία των ασθενών, τότε τα στοιχεία εκείνα που θα κρατάμε για κάθε **ασθενή**, θα είναι **το όνομά του, το επώνυμό του, η ηλικία του, και ο κωδικός του** – ας περιοριστούμε μόνο σε αυτά. Στην περίπτωση αυτή, ο προγραμματιστής έχει δυο επιλογές: είτε να δημιουργήσει τέσσερα διαφορετικά πεδία στα οποία θα καταχωρεί τα στοιχεία του ασθενούς, είτε να δημιουργήσει ένα σύνθετο τύπο δεδομένων – τον οποίο ας ονομάσουμε **Patient** – που θα περιέχει τα τέσσερα πεδία που περιγράψαμε παραπάνω. Οι δύο αυτές προσεγγίσεις παρουσιάζονται στο επόμενο σχήμα:



Σχήμα 1: Περιγραφή οντότητας με απλούς και σύνθετους τύπους δεδομένων

Είναι προφανές, ότι η δεύτερη προσέγγιση είναι πιο αποτελεσματική σε σχέση με την πρώτη, διότι τα τέσσερα δεδομένα που σχηματίζουν τον σύνθετο τύπο δεδομένων **Patient**, υφίστανται διαχείριση, σαν να ήταν ένας απλός τύπος δεδομένων. Αυτό όπως θα δούμε, είναι πολύ σημαντικό, διότι μας επιτρέπει να ανακτούμε πληροφορίες από την εφαρμογή με απλό και εύκολο τρόπο.

Στις συμβατικές γλώσσες προγραμματισμού, αυτοί οι σύνθετοι τύποι δεδομένων ονομάζονται **δομές δεδομένων (data structures)**. Στην τεχνολογία των βάσεων δεδομένων, οι δομές αυτές ονομάζονται **εγγραφές (records)**.

ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ

Στις προηγούμενες παραγράφους ορίσαμε την έννοια του δεδομένου, και είδαμε πως τα δεδομένα της καθημερινής μας ζωής, περιγράφονται με τη βοήθεια των τύπων δεδομένων. Στη συνέχεια ομαδοποιήσαμε τα δεδομένα σε σύνθετους τύπους δεδομένων και εξηγήσαμε τα πλεονεκτήματα αυτής της σύνθετης αναπαράστασης, σε σχέση με την απλή αναπαράσταση που στηρίζεται στη χρήση ενός πεδίου, για κάθε ένα από τα δεδομένα της εφαρμογής.

Στην παράγραφο αυτή, θα προχωρήσουμε ένα βήμα παραπέρα και θα δούμε με ποιο τρόπο μπορούμε να μοντελοποιήσουμε ένα σύστημα, χρησιμοποιώντας σύνθετους τύπους δεδομένων. Θα στηριχθούμε ξανά στο παράδειγμα του νοσοκομείου που περιγράψαμε παραπάνω. Ένα νοσοκομείο είναι γενικά ένα πολύπλοκο σύστημα. Περιλαμβάνει **γιατρούς, νοσοκόμες, ασθενείς**, και ένα σωρό άλλα πράγματα. **Κάθε γιατρός έχει μια ειδικότητα, και εργάζεται σε κάποια από τις κλινικές του νοσοκομείου. Το ίδιο ισχύει και για τις νοσοκόμες. Κάθε ασθενής έχει κάποια πάθηση, του γίνεται κάποια διάγνωση, και ακολουθεί κάποια θεραπεία.** Εάν περιγράψουμε σε πλήρη έκταση τη λειτουργία του νοσοκομείου, θα βρεθούμε αντιμέτωποι με ένα τεράστιο και πολύπλοκο σύστημα, το οποίο θα πρέπει να μηχανογραφήσουμε. Τι θα πρέπει να κάνουμε στην περίπτωση αυτή?

Η απάντηση στο ερώτημα αυτό είναι απλή. **Αρχικά θα πρέπει να ταυτοποιήσουμε τις οντότητες που συμμετέχουν στο σύστημα.** Όπως θα δούμε σε επόμενη ενότητα, οι οντότητες (entities) είναι αυτόνομες μονάδες που κινούνται μέσα στο σύστημα, και ο ρόλος τους είναι εντελώς συγκεκριμένος. Στην προκειμένη περίπτωση, ως οντότητες του συστήματος μπορούμε να χαρακτηρίσουμε για παράδειγμα **τους γιατρούς, τις νοσοκόμες, και τους ασθενείς.** Εφ' όσον

ταυτοποιήσουμε τις οντότητες του συστήματος, στο επόμενο βήμα **θα πρέπει να δούμε, ποιες πληροφορίες θα πρέπει να κρατάμε για κάθε οντότητα**. Ταυτοποιώντας και αυτές τις πληροφορίες, μπορούμε στη συνέχεια να δημιουργήσουμε ένα σύνθετο τύπο δεδομένων για κάθε μια από αυτές τις οντότητες. Ένα παράδειγμα αυτής της προσέγγισης, απεικονίζεται παρακάτω:

ΑΣΘΕΝΗΣ	ΙΑΤΡΟΣ	ΝΟΣΟΚΟΜΑ
Όνομα Επώνυμο Ηλικία Κωδικός	Όνομα Επώνυμο Ειδικότητα Κωδικός	Όνομα Επώνυμο Κλινική Κωδικός

Σχήμα 2: Σύνθετοι τύποι δεδομένων για τις οντότητες ενός νοσοκομείου

Μετά τη δημιουργία αυτών των τύπων δεδομένων – και όσων άλλων χρειαστεί – μπορεί στη συνέχεια να αρχίσει η διαδικασία καταχώρησης των στοιχείων που συσχετίζονται με τη λειτουργία του νοσοκομείου. Στην πραγματικότητα υπάρχουν και κάποια άλλα πράγματα που θα πρέπει να γίνουν, αλλά για την ώρα ας περιοριστούμε μόνο σε αυτή την απλή περιγραφή. Το σύνολο όλων αυτών των πληροφοριών που καταχωρούνται στο σύστημα, αποτελούν μια **βάση δεδομένων (data base)**.

Η βάση δεδομένων ορίζεται λοιπόν ως ένα σύνολο δεδομένων που χαρακτηρίζονται από κάποια λογική οργάνωση και ομαδοποίηση έτσι ώστε να είναι εύκολη και αποτελεσματική η διαχείρισή τους.

Οι βάσεις δεδομένων κατέχουν κεντρική θέση στην επιστήμη της Πληροφορικής, διότι όπως αναφέραμε στον πρόλογο αυτών των σημειώσεων, ένας από τους κυριότερους ρόλους των υπολογιστών στη σημερινή εποχή, είναι η διαχείριση της πληροφορίας. Τα τελευταία χρόνια παρατηρείται μια ραγδαία ανάπτυξη της τεχνολογίας των βάσεων δεδομένων τόσο σε θεωρητικό επίπεδο όσο και σε επίπεδο εφαρμογών, γεγονός που επιβεβαιώνει τη σπουδαιότητα και τη μεγάλη σημασία τους σε όλους σχεδόν τους τομείς στους οποίους χρησιμοποιούνται.

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Μια βάση δεδομένων για να θεωρηθεί σωστά σχεδιασμένη και να μπορεί να λειτουργήσει με αποτελεσματικό τρόπο, θα πρέπει να διαθέτει ένα πλήθος χαρακτηριστικών τα πιο σημαντικά εκ των οποίων, είναι τα ακόλουθα:

1) Δεν θα πρέπει να περιλαμβάνει περιττά δεδομένα (redundant data). Αυτό σημαίνει ότι **τα ίδια δεδομένα δεν θα πρέπει να καταχωρούνται στη βάση δύο φορές**. Εάν συμβαίνει κάτι τέτοιο, τότε ανακύπτουν δυο σοβαρά προβλήματα. Το

πρώτο και προφανές πρόβλημα, είναι ότι **σπαταλούμε άσκοπα αποθηκευτικό χώρο στο σκληρό δίσκο**, αφού την ίδια πληροφορία την αποθηκεύουμε δύο φορές. Το δεύτερο και σοβαρότερο πρόβλημα, είναι ότι **υπάρχει ο κίνδυνος δημιουργίας ασυνεπών δεδομένων (inconsistent data)**. Πράγματι, εάν κρατάμε δύο φορές την ίδια πληροφορία και η πληροφορία αυτή σε κάποια χρονική στιγμή υποστεί κάποιο είδος επεξεργασίας – για παράδειγμα την τροποποιήσουμε ή τη διαγράψουμε – τότε η επεξεργασία αυτή θα πρέπει να εφαρμοσθεί και στις δύο καταχωρήσεις που αφορούν το ίδιο δεδομένο, διότι διαφορετικά, η βάση θα περιέχει δεδομένα που δεν είναι συνεπή. Για το λόγο αυτό, ένας από τους πρώτους ελέγχους που πραγματοποιούμε στη βάση αμέσως μετά το σχεδιασμό της, είναι **ο έλεγχος παρουσίας επαναλαμβανόμενων πεδίων, και η απομάκρυνσή τους, εφ' όσον υπάρχουν**.

2) Η βάση θα πρέπει να είναι **σχεδιασμένη έτσι ώστε τα δεδομένα που περιλαμβάνει να ανακτώνται εύκολα και γρήγορα**. Μια κακή σχεδίαση της δομής της βάσης, οδηγεί στη δημιουργία μιας βάσης η οποία είναι εξαιρετικά δυσκίνητη και αναποτελεσματική. Η σωστή σχεδίαση μιας βάσης δεδομένων, επιτυγχάνεται με εφαρμογή επί της δομής της βάσης, μιας τεχνικής, η οποία ονομάζεται **κανονικοποίηση (normalization)**. Η περιγραφή αυτής της τεχνικής παρουσιάζεται διεξοδικά σε επόμενη ενότητα.

3) Η βάση θα πρέπει να **χαρακτηρίζεται από σύστημα ασφάλειας (security system) που να απαγορεύει την πρόσβαση στα δεδομένα μη εξουσιοδοτημένων ατόμων**. Αυτό ισχύει κυρίως όπως θα δούμε, σε μεγάλες βάσεις δεδομένων με πολλούς χρήστες, και η τεχνική που συνήθως εφαρμόζεται, συνίσταται στον καθορισμό **ομάδων χρηστών (user groups)** με διαφορετικά δικαιώματα πρόσβασης στον καθένα από αυτούς. Ο κάθε χρήστης λαμβάνει ένα **κωδικό πρόσβασης (password)** και τα καθήκοντα που μπορεί να επιτελέσει είναι εντελώς συγκεκριμένα και καθορισμένα εκ των προτέρων.

4) Η βάση θα πρέπει να **μπορεί να διαχειρίζεται ταυτόχρονες προσπελάσεις πάνω στα ίδια δεδομένα (concurrency control)**. Ένα σύστημα κράτησης αεροπορικών εισιτηρίων που δέχεται ταυτόχρονες κλήσεις για κρατήσεις θέσεων, θα πρέπει να έχει τη δυνατότητα να αποτρέψει την καταχώρηση της ίδιας θέσης σε δύο άτομα που την έχουν ζητήσει ταυτόχρονα.

5) Η βάση θα πρέπει να **διαθέτει σύστημα δημιουργίας αντιγράφων ασφαλείας των δεδομένων που είναι καταχωρημένα σε αυτή (backups)**. Η ταυτόχρονη αποθήκευση των δεδομένων σε περισσότερους από ένα δίσκους, είναι μια εργασία επιβεβλημένη, προκειμένου να είναι δυνατή η ανάκτησή τους σε περιπτώσεις κατάρρευσης της βάσης για οποιοδήποτε λόγο.

ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Η διαχείριση μιας βάσης δεδομένων πραγματοποιείται με τη βοήθεια ειδικών προγραμμάτων, που ονομάζονται **συστήματα διαχείρισης βάσεων δεδομένων (Database Management Systems, DBMS)**. Ο ρόλος αυτών των συστημάτων είναι η αποτελεσματική επεξεργασία των δεδομένων της βάσης. Πιο συγκεκριμένα, μέσω των συστημάτων διαχείρισης βάσεων δεδομένων, μπορούμε να υλοποιήσουμε και να διαχειριστούμε οποιαδήποτε βάση. Διακρίνουμε τρία στάδια:

Τη σχεδίαση και υλοποίηση της βάσης. Το στάδιο αυτό πραγματοποιείται από ειδικό προγραμματιστή ο οποίος αφού αναλύσει τις απαιτήσεις που έχουμε από τη βάση δεδομένων, προχωρεί στο σχεδιασμό και την υλοποίησή της, χρησιμοποιώντας τα κατάλληλα εργαλεία ανάπτυξης. Το στάδιο αυτό πραγματοποιείται μόνο μια φορά και απαιτεί ιδιαίτερη προσοχή, καθώς μια εσφαλμένη σχεδίαση, είναι δυνατόν να οδηγήσει σε αναποτελεσματική διαχείριση των δεδομένων, κάτι που μακροπρόθεσμα θα κοστίζει, τόσο χρόνο, όσο και χρήμα.

Την καταχώρηση των δεδομένων στη βάση του συστήματος. Μετά τη σωστή σχεδίαση και υλοποίηση της βάσης, μπορούμε πλέον να καταχωρήσουμε σε αυτή τα δεδομένα του συστήματος για το οποίο υλοποιήθηκε. Η διαδικασία της καταχώρησης γίνεται με τη βοήθεια κατάλληλα σχεδιασμένων φορμών, και μπορεί να πραγματοποιηθεί από οποιονδήποτε.

Τη διαχείριση των δεδομένων της βάσης. Η διαχείριση αυτή περιλαμβάνει μια σειρά από δραστηριότητες όπως την ανάκτηση δεδομένων που πληρούν συγκεκριμένα κριτήρια αναζήτησης, τη δημιουργία αντιγράφων ασφαλείας, και τη μέριμνα για τη σωστή συντήρηση της βάσης εφ' όσον η δυνατότητα αυτή παρέχεται από το σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιούμε.

Κατά καιρούς έχουν εμφανιστεί στην αγορά πολλά συστήματα διαχείρισης βάσεων δεδομένων, από πολλές εταιρείες, τα οποία ποικίλλουν, τόσο στη σχεδίαση και διαχείριση της βάσης, όσο και στις δυνατότητες που προσφέρουν. Τα πιο γνωστά και ευρέως χρησιμοποιούμενα συστήματα διαχείρισης βάσεων δεδομένων είναι η **Microsoft Access**, η **Oracle**, η **Informix**, η **Sybase**, και παλαιότερα η **dBase III plus** και η **dBase IV**. Από τα μοντέρνα συστήματα διαχείρισης, η **Microsoft Access** χρησιμοποιείται κυρίως για την ανάπτυξη και διαχείριση βάσεων δεδομένων σε προσωπικούς υπολογιστές και σε επιχειρήσεις μεσαίου μεγέθους, ενώ η **Oracle** και η **Sybase**, χρησιμοποιούνται σε μεγάλες επιχειρήσεις, και υποστηρίζουν **πολλούς χρήστες, πολλές ταυτόχρονες προσπελάσεις πάνω στα ίδια δεδομένα**, καθώς και **απομακρυσμένη πρόσβαση μέσω δικτύου**.

Ας περάσουμε τώρα στην περιγραφή των παραπάνω τριών σταδίων, που συσχετίζονται με τη διαδικασία ανάπτυξης μιας βάσης δεδομένων. Στη μελέτη που ακολουθεί, θα προσπαθήσουμε να δώσουμε έμφαση, όχι τόσο στις λεπτομέρειες υλοποίησης μιας βάσης, όσο στη λογική που κρύβεται πίσω από αυτή.

Στάδιο 1: Σχεδίαση και υλοποίηση της βάσης

Η σχεδίαση μιας βάσης δεδομένων περιλαμβάνει **τον καθορισμό της δομής της βάσης, των τύπων δεδομένων που θα καταχωρούνται στη βάση, και του πλήθους και του είδους των διαδικασιών που θα εφαρμόζονται στα δεδομένα της βάσης**. Στο σημείο αυτό θα πρέπει να αναφέρουμε ότι πριν το σχεδιασμό της βάσης, προηγείται **πλήρης ανάλυση της συμπεριφοράς και των απαιτήσεων του συστήματος που πρόκειται να υλοποιήσουμε**. Οι τεχνικές λεπτομέρειες που αφορούν αυτή την ανάλυση είναι αδύνατο να αναφερθούν εδώ, και αποτελούν αντικείμενο μιας ολόκληρης θεματικής ενότητας, με τίτλο **Ανάλυση και Σχεδιασμός Συστημάτων**. Στην ενότητα αυτή θα περιοριστούμε μόνο σε μια συνοπτική περιγραφή της μεθοδολογίας που ακολουθείται.

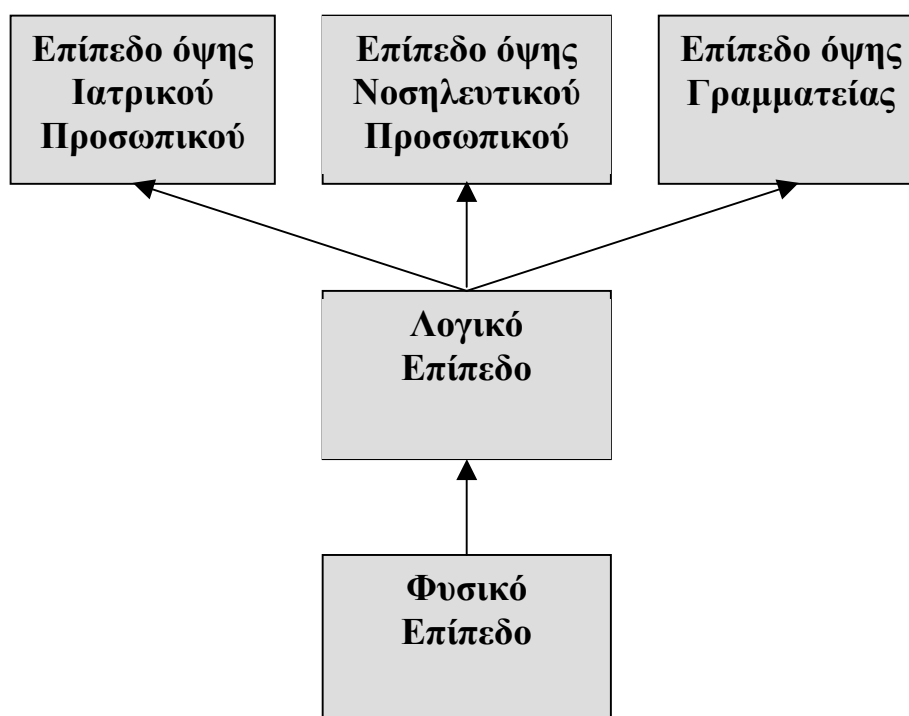
Μιλώντας γενικά, η σχεδίαση μιας βάσης δεδομένων, μπορεί να ορισθεί ως μια διεργασία τριών επιπέδων. Τα επίπεδα αυτά είναι, **το φυσικό επίπεδο (physical level), το λογικό επίπεδο (logical level), και το επίπεδο όψης (view level).**

Στο **φυσικό επίπεδο**, η κύρια μέριμνά μας είναι ο **καθορισμός του τρόπου με τον οποίο τα δεδομένα της βάσης θα αποθηκεύονται στο σύστημα**. Στο επίπεδο αυτό, καθορίζουμε **τους τύπους των δεδομένων της βάσης, τους αλγόριθμους διαχείρισης των δεδομένων**, και γενικά, ασχολούμαστε με διαδικασίες οι οποίες χαρακτηρίζονται από μεγάλο βαθμό πολυπλοκότητας.

Στο **λογικό επίπεδο**, τα πράγματα είναι πιο απλά. Εδώ ξεφεύγουμε από τις λεπτομέρειες και την πολυπλοκότητα του φυσικού επιπέδου, και το μόνο που κάνουμε, είναι **να καθορίζουμε ποια δεδομένα θα αποθηκευτούν στη βάση του συστήματος, και ποιες είναι οι σχέσεις που υφίστανται ανάμεσά τους**. Είναι λοιπόν εύκολα αντιληπτό, πως η διαφορά ανάμεσα στο φυσικό και στο λογικό επίπεδο, έχει να κάνει με τον τύπο διαχείρισης των δεδομένων: **στο φυσικό επίπεδο καθορίζουμε πως τα δεδομένα θα αποθηκευτούν στη βάση του συστήματος, ενώ στο λογικό επίπεδο, καθορίζουμε ποια είναι αυτά τα δεδομένα.**

Τέλος, στο **επίπεδο όψης**, καθορίζουμε **εκείνο το τμήμα της βάσης που θα είναι ορατό σε κάθε χρήστη**. Αυτό κυρίως βρίσκει εφαρμογή σε μεγάλες βάσεις δεδομένων με πολύπλοκη σχεδίαση και μεγάλο αριθμό χρηστών. Ας πάρουμε ξανά για παράδειγμα τη βάση δεδομένων του νοσοκομείου. Τα δεδομένα που καταχωρούνται στη βάση αυτή, είναι κάθε είδους, και γενικά μη συσχετιζόμενα μεταξύ τους. Για παράδειγμα, για κάθε **ασθενή**, καταχωρούμε **τα προσωπικά και οικονομικά του στοιχεία, το ιατρικό ιστορικό του, και την εξέλιξη της υγείας του**. Τα στοιχεία αυτά, τα χειρίζονται εν γένει, διαφορετικοί χρήστες: τα **οικονομικά στοιχεία** τα επεξεργάζεται η **γραμματεία του νοσοκομείου**, ενώ τα **ιατρικά στοιχεία, οι θεράποντες ιατροί του ασθενούς**. Αυτό σημαίνει ότι το πρόγραμμα που θα χρησιμοποιείται στη γραμματεία του νοσοκομείου, θα εμφανίζει μόνο τα οικονομικά στοιχεία, ενώ τα προγράμματα που χρησιμοποιούν οι ιατροί, θα εμφανίζουν μόνο τα ιατρικά στοιχεία. Τα δεδομένα λοιπόν που θα διαχειρίζονται οι δυο αυτές κατηγορίες χρηστών, θα είναι εντελώς ξένα μεταξύ τους, και μάλιστα σε τέτοιο σημείο, ώστε να δημιουργείται η ψευδαίσθηση ότι πρόκειται για δυο διαφορετικές βάσεις, ενώ στην πραγματικότητα, η βάση είναι μια και μοναδική. Ένας άλλος λόγος για τον οποίο συμβαίνει αυτός ο διαχωρισμός, είναι **ζητήματα ασφαλείας**. Για παράδειγμα, **ένας υπάλληλος της γραμματείας, δεν θα πρέπει να μπορεί να διαβάσει τον ιατρικό φάκελλο του ασθενούς, διότι παραβιάζει το ιατρικό απόρρητο**. Όπως δε, θα δούμε στη συνέχεια, είναι κοινή πρακτική, κυρίως στις περιπτώσεις των μεγάλων βάσεων δεδομένων, **να ορίζονται ομάδες χρηστών στους οποίους να καταχωρούνται διαφορετικά δικαιώματα πρόσβασης, και διαφορετικά δεδομένα προς διαχείριση**.

Στο επόμενο σχήμα απεικονίζονται τα επίπεδα σχεδιασμού της βάσης για το παράδειγμα του νοσοκομείου που περιγράψαμε παραπάνω.



Σχήμα 3: Επίπεδα Σχεδιασμού Βάσεων Δεδομένων

Η μοντελοποίηση του συστήματος που υλοποιεί η βάση δεδομένων, πραγματοποιείται με τη βοήθεια ειδικών εργαλείων, και με την εφαρμογή γνωστών και καλά δοκιμασμένων τεχνικών. Αυτό γίνεται κυρίως στο **λογικό επίπεδο σχεδιασμού της βάσης**, όπου ταυτοποιούνται τα δεδομένα που θα καταχωρούνται σε αυτή, καθώς και οι σχέσεις που υφίστανται ανάμεσά τους. Η λογική σχεδίαση της βάσης, στηρίζεται και αυτή στη χρήση κάποιων μοντέλων που περιγράφουν τη δομή και τη λειτουργία της. Τα τελευταία χρόνια, που η τεχνολογία των βάσεων δεδομένων έχει γνωρίσει αλματώδη ανάπτυξη, αναπτύχθηκαν πολλά τέτοια μοντέλα, που στηρίζονται σε διαφορετικές εν γένει προσεγγίσεις. Τα τρία πιο γνωστά από αυτά τα μοντέλα, είναι το **ιεραρχικό μοντέλο (hierarchical model)**, το **δικτυωτό μοντέλο (network model)** και το **σχεσιακό μοντέλο (relational model)**. Στα κεφάλαια που ακολουθούν, θα περιγραφεί μόνο το **σχεσιακό μοντέλο**, που χρησιμοποιείται κατά κύριο λόγο στις μοντέρνες βάσεις δεδομένων. Ο βασικός λόγος για τη διάδοση αυτού του μοντέλου και τη χρησιμοποίησή του έναντι των δύο άλλων, είναι, όπως θα δούμε σε επόμενα κεφάλαια, η **χρήση πινάκων, για την περιγραφή του συστήματος που θέλουμε να μοντελοποιήσουμε**.

Ο καθορισμός και η διαχείριση της δομής της βάσης πραγματοποιούνται με τη βοήθεια ειδικών γλωσσών προγραμματισμού. Οι γλώσσες αυτές, χαρακτηρίζονται ως **γλώσσες τέταρτης γενιάς (4th Generation Languages, 4GL)**, διότι σε αντίθεση με τις συμβατικές γλώσσες προγραμματισμού – όπως είναι η **Pascal** και η **Basic** – δεν υλοποιούν πολύπλοκες λειτουργίες χαμηλού επιπέδου, αλλά το μόνο που κάνουν είναι **να ανακτούν δεδομένα από τη βάση του συστήματος, χωρίς ο χρήστης να γνωρίζει πως λαμβάνει χώρα αυτή η διαδικασία**. Εκτός από την ανάκτηση των δεδομένων, οι γλώσσες αυτές προσφέρουν και πιο προχωρημένες λειτουργίες, όπως είναι η **δημιουργία νέων δομών δεδομένων μέσα στη βάση, η διαγραφή και η**

τροποποίηση αυτών, και ο ορισμός συσχετίσεων μεταξύ των στοιχείων της βάσης.

Μια απαραίτητη διευκρίνιση που πρέπει να γίνει στο σημείο αυτό, αφορά το είδος των λειτουργιών που εφαρμόζονται στη βάση δεδομένων. Οι λειτουργίες αυτές μπορούν να διαχωριστούν σε δύο κατηγορίες: **σ' αυτές που τροποποιούν τη δομή της βάσης, και σ' εκείνες που τροποποιούν το περιεχόμενο της βάσης, δηλαδή τα δεδομένα που αποθηκεύονται σε αυτή.** Η διαγραφή ενός ασθενούς από το μητρώο των ασθενών της κλινικής, δεν μεταβάλλει τη δομή της βάσης, απλώς ελαττώνει τον αριθμό των εγγεγραμμένων ασθενών κατά ένα. Αντίθετα η προσθήκη κάποιας καινούριας πληροφορίας στο φάκελο του ασθενή, όπως για παράδειγμα η ομάδα αίματός του, μεταβάλλει τη δομή της βάσης, αφού τα δεδομένα που καταχωρούνται σε αυτή, έχουν πλέον αλλάξει. Είναι προφανές, ότι οι απλοί χρήστες που χρησιμοποιούνται για τη διαχείριση των δεδομένων του συστήματος, δεν πρέπει να μπορούν σε καμιά περίπτωση να αλλάξουν τη δομή της βάσης, κάτι που μπορεί να γίνει μόνο από τον **διαχειριστή της βάσης δεδομένων (Database Administrator, DBA)**. Αυτό σημαίνει ότι για τα δύο αυτά επίπεδα διαχείρισης, χρησιμοποιούνται διαφορετικές γλώσσες προγραμματισμού: **μια γλώσσα που είναι υπεύθυνη για τον ορισμό και την τροποποίηση της δομής της βάσης και η οποία ονομάζεται γλώσσα ορισμού δεδομένων (Data Definition Language, DDL), και μια γλώσσα που είναι υπεύθυνη για τη διαχείριση των δεδομένων της βάσης, και η οποία ονομάζεται γλώσσα διαχείρισης δεδομένων (Data Manipulation Language, DML).** Περισσότερες λεπτομέρειες πάνω στον τρόπο χρησιμοποίησης αυτών των γλωσσών, θα παρουσιαστούν σε επόμενο κεφάλαιο.

Στάδιο 2: Καταχώρηση δεδομένων στη βάση

Στο στάδιο αυτό, η σχεδίαση και υλοποίηση της βάσης έχει πλέον ολοκληρωθεί, και η βάση είναι έτοιμη να χρησιμοποιηθεί για την καταχώρηση των δεδομένων. Η καταχώρηση αυτή δεν απαιτεί ειδικές γνώσεις πληροφορικής, και μπορεί να γίνει από οποιοδήποτε χρήστη, με μικρή σχετικά εκπαίδευση πάνω στη λειτουργία του προγράμματος. Για την καταχώρηση των δεδομένων χρησιμοποιούνται ειδικές φόρμες που περιλαμβάνουν πλαίσια εισαγωγής δεδομένων, συνοδευόμενα από κάποιο τίτλο, που να περιγράφει τη λειτουργία τους. Μια τυπική φόρμα καταχώρησης, φαίνεται στο επόμενο σχήμα:

The screenshot shows a software window titled 'Orders'. It contains several input fields and a table. The 'Bill To' and 'Ship To' fields are identical, showing 'Alfreds Futterkiste' at 'Obere Str. 57', 'Berlin', '12209', 'Germany'. The 'Salesperson' is 'Suyama, Michael'. The 'Ship Via' section has three checkboxes: 'Speedy' (checked), 'United', and 'Federal'. Below these are fields for 'Order ID' (10643), 'Order Date' (25-Aug-1997), 'Required Date' (22-Sep-1997), and 'Shipped Date' (02-Sep-1997). A table lists products with columns: Product, Unit Price, Quantity, Discount, and Extended Price. The products are 'Spegesild', 'Chartreuse verte', and 'Rissle Sauerkraut'. At the bottom right, a summary table shows 'Subtotal: \$814,50', 'Freight: \$29,46', and 'Total: \$843,96'. At the bottom left, there is a 'Display products of the month' button and a 'Print Invoice' button. The bottom status bar shows 'Εγγραφή: 1 από 830'.

Product	Unit Price	Quantity	Discount	Extended Price
Spegesild	\$12,00	2	25%	18,00 Δpx
Chartreuse verte	\$18,00	21	25%	283,50 Δpx
Rissle Sauerkraut	\$45,60	15	25%	513,00 Δpx
*			0%	

Subtotal:	\$814,50
Freight	\$29,46
Total:	\$843,96

Σχήμα 4: Φόρμα Εισαγωγής Δεδομένων

Όπως φαίνεται από το σχήμα αυτό, τα δεδομένα που καταχωρούνται στη βάση, περιγράφονται από ετικέτες που συνήθως βρίσκονται πάνω, αριστερά ή δεξιά από τα πλαίσια εισαγωγής των δεδομένων. Εκτός όμως από τα πλαίσια εισαγωγής, χρησιμοποιούμε και άλλα στοιχεία καταχώρησης δεδομένων, όπως **list boxes**, **combo boxes**, **check boxes** και **radio buttons**.

Η διαδικασία καταχώρησης δεδομένων στη βάση του συστήματος, διευκολύνεται σε πολύ μεγάλο βαθμό, εάν οι φόρμες μέσα από τις οποίες ο χρήστης καταχωρεί αυτά τα δεδομένα, είναι λειτουργικές, έξυπνα σχεδιασμένες, και κυρίως, χωρίς περιττά στοιχεία. Μια εφαρμογή η οποία δεν χαρακτηρίζεται από απλότητα όσον αφορά τη λειτουργία της, είναι σίγουρο ότι θα τρομάζει τον απλό χρήστη και θα μειώσει σημαντικά την αποδοτικότητά του. Αντίθετα μια σωστά σχεδιασμένη εφαρμογή που ανα πάσα στιγμή θα δίνει στο χρήστη να καταλάβει τι ακριβώς πρέπει να κάνει, θα του επιτρέψει να ολοκληρώσει την εργασία του εύκολα και γρήγορα. **Ο σχεδιασμός της αλληλεπίδρασης του προγράμματος με τους χρήστες (User Interface, UI), αποτελεί αναπόσπαστο στοιχείο κάθε εφαρμογής, και απαιτεί πολύ μεγάλη προσοχή.**

Στάδιο 3 : Διαχείριση των δεδομένων της βάσης

Η διαχείριση των δεδομένων της βάσης, πραγματοποιείται ανά πάσα στιγμή μετά την υλοποίηση αυτής, και εφαρμόζεται στα δεδομένα εκείνα που έχουν ήδη καταχωρηθεί. Η κυριότερη μορφή διαχείρισης των δεδομένων της εφαρμογής, έχει να κάνει με την επιλογή όσων από αυτά πληρούν κάποια κριτήρια αναζήτησης ή έχουν κάποια συγκεκριμένη τιμή. Ας πάρουμε για παράδειγμα την περίπτωση μιας βιβλιοθήκης, για την οποία έχουμε αναπτύξει μια βάση δεδομένων. Έστω ότι θέλουμε να μάθουμε ποια βιβλία είναι γραμμένα στην Ελληνική γλώσσα, αναφέρονται στην Ιστορία του Βυζαντινού Πολιτισμού, και έχουν εκδοθεί στη δεκαετία του 1990. Είναι προφανές ότι χωρίς τη βάση δεδομένων, η ανάκτηση αυτής της

πληροφορίας, θα απαιτούσε πολλές ώρες δουλειάς και πολύ μεγάλη προσπάθεια. Με τη χρησιμοποίηση όμως της βάσης, αυτή η διαδικασία αναζήτησης, είναι δουλειά μόλις λίγων λεπτών. Η ικανότητα αυτή των βάσεων δεδομένων να ανακτούν τις πληροφορίες που θέλουμε σε ελάχιστο χρονικό διάστημα, είναι ο βασικός λόγος που οδήγησε στην αλματώδη ανάπτυξή τους τα τελευταία χρόνια.

Μετά την ανάκτηση των δεδομένων από τη βάση του συστήματος, μπορούμε να εφαρμόσουμε πάνω σε αυτά πολλούς τρόπους διαχείρισης, όπως για παράδειγμα, **να τα εκτυπώσουμε**. Οι μηχανογραφικές καταστάσεις με τις βαθμολογίες των μαθητών που αναρτώνται στα Γυμνάσια και στα Λύκεια στο τέλος κάθε εξεταστικής περιόδου, δεν αποτελούν παρά τέτοιου είδους εκτυπώσεις, οι οποίες δημιουργούνται από τα προγράμματα που διαχειρίζονται τα δεδομένα αυτού του είδους. Ένα άλλο είδος διαχείρισης **είναι η ταξινόμησή τους με βάση κάποια κριτήρια**, όπως για παράδειγμα, με αλφαβητική σειρά. Τέλος μπορούμε να παρέμβουμε στα ίδια τα δεδομένα και να αλλάζουμε τις τιμές τους, ή ακόμη και να τα διαγράψουμε από τη βάση.

Όπως είναι λοιπόν εύκολο να διαπιστωθεί, οι διαδικασίες που εφαρμόζονται πάνω στα δεδομένα της βάσης, μπορούν να διαχωρισθούν σε δύο κατηγορίες. Στην πρώτη κατηγορία, ανήκουν διεργασίες **οι οποίες περιορίζονται μόνο στην εμφάνιση των δεδομένων στην οθόνη του υπολογιστή και γενικά σε ενέργειες παρόμοιας μορφής**, οι οποίες όμως **δεν μεταβάλλουν τα δεδομένα**, ενώ στη δεύτερη κατηγορία, ανήκουν διεργασίες, **οι οποίες τροποποιούν τις τιμές των δεδομένων, ή ακόμη και τα διαγράφουν τελείως**. Για το λόγο αυτό απαιτείται ιδιαίτερη προσοχή κυρίως για τις διεργασίες της δεύτερης κατηγορίας, διότι στις πιο πολλές περιπτώσεις τα αποτελέσματα αυτών των διεργασιών δεν είναι αναστρέψιμα, και δύνανται να οδηγήσουν ακόμη και σε οριστική απώλεια των δεδομένων από τη βάση της εφαρμογής.

Σε όλα τα συστήματα διαχείρισης βάσεων δεδομένων, η διαχείριση των δεδομένων γίνεται με τη βοήθεια ειδικών εργαλείων. Τα εργαλεία αυτά είναι όπως έχουμε ήδη περιγράψει **η γλώσσα ορισμού δεδομένων και η γλώσσα χειρισμού δεδομένων**. Υπάρχουν πολλές εκδόσεις και πολλές μορφές για αυτές τις γλώσσες, τα τελευταία όμως χρόνια έχει επικρατήσει ένα κοινό πρότυπο που φέρει το όνομα **δομημένη γλώσσα ερωτοαποκρίσεων (Structured Query Language, SQL)**. Η γλώσσα αυτή απαντάται σε όλα ανεξαιρέτως τα συστήματα διαχείρισης βάσεων δεδομένων, και παρά το γεγονός ότι εμφανίζεται σε πολλές παραλλαγές, η βασική ιδέα που περιγράφει τη λειτουργία της, είναι ουσιαστικά η ίδια.

Η σύνταξη και χρήση της γλώσσας **SQL** θα περιγραφεί με μεγάλη λεπτομέρεια σε επόμενο κεφάλαιο. Στο σημείο όμως αυτό μπορούμε να αναφέρουμε, πως οι βασικές λειτουργίες που αυτή η γλώσσα επιτελεί, διαχωρίζονται σε τέσσερις μεγάλες κατηγορίες, που είναι **η εισαγωγή δεδομένων (data insertion), η διαγραφή δεδομένων (data deletion), η ανάκτηση δεδομένων (data retrieval) και η τροποποίηση δεδομένων (data update)**.