

## Κεφάλαιο 1

### ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

1.1 Εισαγωγή

1.2 Είδη και τύποι αρχείων

1.3 Διεργασίες αρχείων

1.4 Ασκήσεις

# Κεφάλαιο 1

## ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ

### 1.1 Εισαγωγή

Στις επιστημονικές εφαρμογές τα δεδομένα είναι απλού τύπου, δηλαδή είναι διάφορα βαθμωτά μεγέθη. Όμως σε εμπορικές εφαρμογές είναι πιο σύνθετα και ονομάζονται **οντότητες** (entities). Για παράδειγμα, οντότητες είναι οι υπάλληλοι μίας εταιρείας, οι συναλλαγές σε ένα τραπεζικό σύστημα, οι φοιτητές στο σύστημα της γραμματείας κλπ. Απαιτεί, λοιπόν, ιδιαίτερη προσοχή ο σωστός καθορισμός των οντοτήτων ενός συστήματος και αποτελεί κομμάτι της ύλης του μαθήματος των Βάσεων Δεδομένων.

Κάθε οντότητα αποτελείται από **χαρακτηριστικά** (attributes) που διευκολύνουν την καλύτερη διάκριση μεταξύ των οντοτήτων. Για παράδειγμα, η οντότητα 'Υπάλληλος' έχει ως χαρακτηριστικά: το ονοματεπώνυμο, τον κωδικό, τη διεύθυνση, το τμήμα εργασίας, την ειδικότητα, το μισθό κλπ. Γενικά, τα χαρακτηριστικά μπορεί να είναι διαφορετικού τύπου (ακέραιος, πραγματικός, συμβολοσειρά) και συνεπώς η αποθήκευσή τους δεν μπορεί να γίνει με πίνακες απλών τύπων αλλά με εγγραφές.

Οι **εγγραφές** (records) είναι δομές που μπορούν να στεγάσουν διαφορετικών τύπων δεδομένα, που περιγράφουν ένα στιγμιότυπο κάποιας οντότητας. Κάθε εγγραφή αποτελείται από **πεδία** (fields). Τα πεδία της εγγραφής μπορεί να είναι αριθμητικά, αλφαβητικά, αλφαριθμητικά και κωδικοποιημένα/συμπιεσμένα. Η κωδικοποίηση και η συμπίεση των δεδομένων είναι ιδιαίτερα μεγάλη περιοχή και θίγεται στο τελευταίο κεφάλαιο των ειδικών θεμάτων. Στο σημείο αυτό απλώς σημειώνεται, ότι με σωστή κωδικοποίηση

επιτυγχάνεται οικονομία χώρου και χρόνου, ευκολία στην ταξινόμηση, τον προγραμματισμό κλπ.

Οι δομές αρχείων (file structures) ή οργανώσεις αρχείων (file organizations) ή φυσικές οργανώσεις (physical organizations) ή, τέλος απλούστερα, τα αρχεία ορίζονται ως μία συλλογή εγγραφών του ίδιου τύπου. Η διαφορά ενός αρχείου για ένα σύστημα διαχείρισης βάσεων δεδομένων ως προς ένα αρχείο ενός λειτουργικού συστήματος είναι ότι στην πρώτη περίπτωση το αρχείο θεωρείται ως μία συλλογή εγγραφών ή σελίδων (δες Κεφάλαια 2 και 3), ενώ στη δεύτερη περίπτωση θεωρείται ως μία συμβολοσειρά χαρακτήρων.

Σε λογικό επίπεδο τα αρχεία μπορεί να θεωρηθούν ως δισδιάστατες δομές. Η οριζόντια διάσταση παριστά τα πεδία μίας εγγραφής, ενώ η κατακόρυφη διάσταση παριστά τον αριθμό των εγγραφών. Ο ορισμός αυτός συμπεριλαμβάνει και τα αρχεία κειμένου. Στο Σχήμα 1.1 παρουσιάζεται το αρχείο ενός συντάκτη κειμένου, όπου κάθε γραμμή παριστά μία γραμμή του κειμένου και έχει δύο χαρακτηριστικά. Τα αρχεία πρέπει να θεωρούνται ως ένα ενιαίο αντικείμενο και όχι ως άθροισμα αντικειμένων, αφού άλλωστε περνούν μεταξύ των υποπρογραμμάτων ως απλές παράμετροι.

Γραμμή	Κείμενο
1	Τρία κόκκινα περιστέρια μέσα στο φώς
2	χαράζοντας τη μοίρα μας μέσα στο φώς με
3	χρώματα και χειρονομίες ανθρώπων που
4	αγαπήσαμε.

Σχήμα 1.1: Αρχείο συντάκτη κειμένου.

Τα πεδία που χρησιμοποιούνται για την αναζήτηση ή την ταξινόμηση των εγγραφών ονομάζονται κλειδιά (keys). Το σύνολο των δυνατών τιμών που μπορεί να λάβει ένα κλειδί λέγεται πεδίο ορισμού (domain). Το κλειδί που χαρακτηρίζει κατά μοναδικό και αμφιμονοσήμαντο τρόπο την κάθε εγγραφή λέγεται πρωτεύον (primary) (πχ. ο κωδικός του υπαλλήλου), ενώ τα υπόλοιπα πεδία λέγονται δευτερεύοντα (secondary). Είναι δυνατόν να υπάρχουν περισσότερα του ενός κλειδιά που να μπορούν να χρησιμοποιηθούν ως πρωτεύοντα κλειδιά. Τα κλειδιά αυτά ονομάζονται υποψήφια (candidate), ενώ αυτά που τελικά δεν επιλέγονται για τέτοια χρήση λέγονται εναλλακτικά (alternate). Άλλοτε πάλι ένα πρωτεύον κλειδί είναι ένας συνδυασμός μερικών απλών κλειδιών, οπότε ονομάζεται σύνθετο (composite).

Μερικές φορές το πρωτεύον κλειδί δεν έχει πραγματικό νόημα αλλά είναι τεχνητό κατασκευάσμα. Το κλειδί αυτό ονομάζεται **εξωτερικό** (external). Για παράδειγμα, ο αριθμός ταυτότητας αντικειμένου (object identification number, oid) είναι ένας αριθμός που δίνεται από μία αντικειμενοστραφή βάση δεδομένων σε εγγραφές αντικειμένων.

Τα αρχεία αποθηκεύονται αποκλειστικά στη δευτερεύουσα μνήμη επειδή:

- το πλήθος των εγγραφών είναι πολύ μεγάλο, οπότε η κύρια μνήμη δεν επαρκεί,
- κάθε φορά μόνο ένα κομμάτι του αρχείου χρησιμοποιείται κατά την επεξεργασία του αρχείου, και τέλος έτσι
- πολλά προγράμματα εφαρμογών μπορούν να προσπελάσουν το ίδιο αρχείο ταυτόχρονα.

## 1.2 Είδη και τύποι αρχείων

Τα είδη των αρχείων είναι τα εξής:

**σειριακά** (sequential), όπου οι εγγραφές είναι αποθηκευμένες διαδοχικά στη μνήμη και προσπελάζονται κατά τον ίδιο τρόπο,

**τυχαία** (random) ή **άμεσα** (direct), όπου η διεύθυνση στη μνήμη προκύπτει με μετασχηματισμό του πρωτεύοντος κλειδιού. Υποπερίπτωση είναι τα **σχετικά** (relative) αρχεία, όπου η θέση κάθε εγγραφής προσδιορίζεται σε σχέση με τη θέση της πρώτης εγγραφής του αρχείου,

**σειριακά με δείκτη** (index sequential), που από άποψη χρόνου προσπέλασης είναι ένας συμβιβασμός μεταξύ του σειριακού και του τυχαίου αρχείου,

**δενδρικά** (tree-structured), που χρησιμοποιούνται κατά κόρο στα εμπορικά πακέτα, και τέλος

**ανάκτησης με βάση δευτερεύον κλειδί** (secondary key retrieval), που συνήθως στηρίζονται στη δομή της συνδεδεμένης λίστας ή στις δομές των δένδρων. Στην κατηγορία αυτή ανήκουν και οι λεγόμενες πολυδιάστατες δομές (όπως το  $k$ -d δένδρο, το δικτυωτό αρχείο κλπ), που αποτελούν σύγχρονες και αποτελεσματικές λύσεις.

Στο σημείο αυτό δεν γίνεται ευρύτερη αναφορά, επειδή κάθε είδος αρχείου που αναφέρθηκε θα εξετασθεί εκτενέστερα στα επόμενα κεφάλαια.

Τα δεδομένα μίας συγκεκριμένης εφαρμογής αποθηκεύονται σε πολλά αρχεία, όπου το κάθε αρχείο χαρακτηρίζεται από διαφορετική λειτουργικότητα. Με βάση, λοιπόν, τη λειτουργία τους τα αρχεία κατηγοριοποιούνται ως εξής.

**κύριο ή βασικό αρχείο (master file).** Περιέχει τα μόνιμα δεδομένα και πρέπει να είναι πάντα ενημερωμένο, όσον αφορά στις εισαγωγές, διαγραφές και ενημερώσεις των εγγραφών. Για παράδειγμα, το κύριο αρχείο σε μία εφαρμογή μισθοδοσίας μεταξύ των άλλων θα περιέχει τα εξής πεδία: κωδικός αριθμός υπαλλήλου, οικογενειακή κατάσταση, επιδόματα, τωρινός μισθός, έσοδα από την αρχή της χρονιάς κλπ.

**αρχείο συναλλαγών ή κινήσεως (transaction file).** Περιέχει τις εγγραφές του κυρίου αρχείου που έχουν εισαχθεί, διαγραφεί ή ενημερωθεί. Για παράδειγμα, ένα αρχείο συναλλαγών σε μία εφαρμογή μισθοδοσίας περιέχει τις ώρες εργασίας κάθε υπαλλήλου για μία περίοδο. Αυτές οι συναλλαγές στο τέλος της περιόδου εφαρμόζονται στο κύριο αρχείο με σκοπό την έκδοση των επιταγών πληρωμών και την ανανέωση των συνολικών εσόδων κάθε υπαλλήλου από την αρχή του έτους.

**βοηθητικό αρχείο ή αρχείο πίνακας (auxiliary file) ή (table file).** Περιέχει μία σειρά στατικών δεδομένων, όπου συχνά αναφέρονται τα άλλα αρχεία. Για παράδειγμα, ένα τέτοιο αρχείο μπορεί να περιέχει εγγραφές με τους κωδικούς των προϊόντων, την περιγραφή και τις τιμές τους. Αυτό το αρχείο θα συνοδεύει ένα κύριο αρχείο αποθήκης, ένα κύριο αρχείο παραγγελιών και ένα αρχείο παραγωγής. Έτσι γίνεται οικονομία στο χώρο αποθήκευσης, επειδή αποφεύγεται η επανάληψη πλεοναζόντων (redundant) δεδομένων.

**αρχείο αναφορών (report file).** Περιέχει πληροφορίες, που έχουν συλλεχθεί και προετοιμασθεί για το χρήστη. Ένα τέτοιο αρχείο είτε εμφανίζεται στην οθόνη, είτε εκτυπώνεται.

**αρχείο ελέγχων (control file).** Συνήθως είναι ένα μικρό αρχείο και περιέχει πληροφορίες σχετικές, για παράδειγμα, με τον αριθμό των εισαγωγών, διαγραφών και ενημερώσεων στο κύριο αρχείο, στο αρχείο συναλλαγών κλπ. Ο αριθμός των εγγραφών στο νέο κύριο αρχείο

προκύπτει από το άθροισμα του πλήθους των εγγραφών στο παλιό κύριο αρχείο και του πλήθους των εισαγωγών και διαγραφών στο αρχείο ελέγχων.

**ιστορικό αρχείο** (history file). Αποτελείται από όλα τα προηγούμενα κύρια αρχεία, αρχεία συναλλαγών και αρχεία ελέγχων. Τα αρχεία αυτά συνήθως δεν αποθηκεύονται σε μαγνητικό δίσκο αλλά σε μαγνητική ταινία.

**εφεδρικό αρχείο** (backup file). Είναι ένα αντίγραφο του κύριου αρχείου, το οποίο χρησιμοποιείται μόνο σε περίπτωση καταστροφής του κύριου αρχείου.

### 1.3 Διεργασίες αρχείων

Ο χρήστης ενός συστήματος αρχείων σε ένα μεγάλο υπολογιστικό σύστημα ανήκει σε μία από τρεις διακριτές κατηγορίες, δηλαδή μπορεί να είναι **τελικός χρήστης** (end user), **προγραμματιστής εφαρμογών** (application programmer) και **προγραμματιστής συστήματος** (systems programmer). Ο τελικός χρήστης θεωρείται ότι βρίσκεται στο υψηλότερο επίπεδο και του παρέχονται οι δυνατότητες μίας γλώσσας ερωτήσεων (query language), δηλαδή μία μικρή ομάδα εντολών πολύ υψηλού επιπέδου. Ο προγραμματιστής εφαρμογών έχει στη διάθεσή του μία γλώσσα υψηλού επιπέδου που λέγεται **φιλοξενούσα γλώσσα** (host language), επειδή είναι ενισχυμένη με μία ομάδα εντολών που αναφέρονται ως **υπογλώσσα δεδομένων** (data sublanguage). Η υπογλώσσα αυτή υποδιαιρείται στην υπογλώσσα **ορισμού δεδομένων** (data definition) και στην υπογλώσσα **χειρισμού δεδομένων** (data manipulation), όπου η κάθε συνιστώσα έχει προφανές περιεχόμενο. Τέλος, ο προγραμματιστής του συστήματος έχει στη διάθεσή του μία γλώσσα χαμηλού επιπέδου και αρμοδιότητά του είναι η διαχείριση του συστήματος αρχείων.

Ο τελικός χρήστης δεν έχει καμία άποψη για το μέσο αποθήκευσης, το είδος και τον τύπο του αρχείου, το μέγεθος του αρχείου και τη γραμμογράφηση της εγγραφής. Η μόνη απαίτηση του χρήστη αυτού είναι αξιόπιστη πληροφορία και γρήγορη απόκριση. Βέβαια, όλα αυτά τα θέματα είναι σε γνώση του προγραμματιστή εφαρμογών, που όμως με τη σειρά του δεν παρεμβαίνει στη διαχείριση του συστήματος αρχείων. Για το σκοπό αυτό

το πρόγραμμα του τελικού χρήστη μεταφράζεται από ένα πρόγραμμα κωδικοποιημένο στη φιλοξενούσα γλώσσα του προγραμματιστή εφαρμογών, το οποίο με τη σειρά του καλεί το σύστημα αρχείων μέσω των σχετικών μεθόδων προσπέλασης που είναι προγραμματισμένες σε γλώσσα χαμηλού επιπέδου. Το σύστημα επιστρέφει στο πρόγραμμα του προγραμματιστή εφαρμογών τις σχετικές εγγραφές, που στη συνέχεια μεταφέρονται στον τελικό χρήστη. Αυτή η διαδοχή μεταφράσεων προγραμμάτων έχει ως συνέπεια ότι αν τυχόν γίνουν κάποιες αλλαγές στο σύστημα των αρχείων, τότε δεν είναι απαραίτητο ο χρήστης να αλλάξει τα προγράμματά του. Δηλαδή, ως προς τους τελικούς χρήστες υπάρχει **ανεξαρτησία των δεδομένων από τα προγράμματα** (program/data independence), ενώ κάτι τέτοιο βέβαια δεν ισχύει για τις άλλες δύο κατηγορίες χρηστών. Στο επόμενο κεφάλαιο θα δοθούν περισσότερες λεπτομέρειες σχετικά με τα θέματα της αρμοδιότητας του προγραμματιστή του συστήματος.

Οι κυριότερες διεργασίες ή πράξεις επί των αρχείων είναι οι εξής: η ερώτηση, η διατήρηση, η ταξινόμηση και η συγχώνευση. Από τις διεργασίες αυτές μόνο οι δύο πρώτες είναι στη δικαιοδοσία του τελικού χρήστη και του προγραμματιστή εφαρμογών, ενώ οι άλλες δύο διεργασίες είναι της αρμοδιότητας του προγραμματιστή του συστήματος. Ας σημειωθεί ότι οι πράξεις που αναφέρονται στο λογικό επίπεδο των βάσεων δεδομένων, όπως η επιλογή (selection), η προβολή (projection), ο συνδυασμός (join), το καρτεσιανό γινόμενο κλπ. στο φυσικό επίπεδο στηρίζονται στις στοιχειώδεις πράξεις των αρχείων, οι οποίες αναφέρθηκαν.

**Ερώτηση** (query) στο αρχείο είναι η αναζήτηση εγγραφών, που έχουν συγκεκριμένες τιμές σε ένα ή περισσότερα συγκεκριμένα πεδία. Στη συνέχεια δίνονται οι (περισσότεροι) τύποι ερωτήσεων και αντίστοιχα παραδείγματα θεωρώντας ένα αρχείο μισθοδοσίας με τα πεδία: Κωδικός, Όνομα, Φύλλο, Μισθός, Έτος γέννησης, Τμήμα, Θέση.

**απλή ερώτηση με βάση το πρωτεύον κλειδί** (simple query based on primary key), ή **σημειακή ερώτηση** (point query) όπως: 'Ποιός υπάλληλος έχει κωδικό 999;',

**απλή ερώτηση με βάση το δευτερεύον κλειδί** (simple query based on secondary key), ή **πολυσημειακή ερώτηση** (multipoint query), όπως: 'Ποιοί είναι οι υπάλληλοι του Λογιστηρίου;',

**ερώτηση διαστήματος** (range query), όπως: 'Ποιοί υπάλληλοι γεννήθηκαν πριν το 1960;',

ερώτηση διάταξης (ordering query), όπως: `Να δοθούν τα ονόματα των υπαλλήλων διατεταγμένα κατά φθίνοντα μισθό`,

ερώτηση ταύτισης προθέματος (prefix match query), όπως: `Ποιοί είναι υπάλληλοι, των οποίων το όνομα αρχίζει από Μανω;`,

ακραία ερώτηση (extremal query), όπως: `Ποιοί υπάλληλοι παίρνουν το μεγαλύτερο μισθό;`,

συναρτησιακή ερώτηση (functional query), όπως: `Ποιοί υπάλληλοι παίρνουν μισθό μικρότερο από το μέσο μισθό όλων των υπαλλήλων;`,

ερώτηση ομαδοποίησης (grouping query), όπως: `Ποιός είναι ο μέσος μισθός των υπαλλήλων κάθε τμήματος;`,

ερώτηση επακριβούς ταύτισης (exact match query), όπως: `Ποιοί είναι οι κωδικοί των υπαλλήλων που λέγονται Ιωάννου, είναι άνδρες, παίρνουν μισθό περισσότερο από 300.000 δρχ, γεννήθηκαν πριν το 1950, ανήκουν στο τμήμα πωλήσεων και έχουν θέση προϊσταμένου;`,

ερώτηση μερική ταύτισης (partial match query), όπως: `Ποιοί υπάλληλοι είναι γυναίκες και έχουν θέση διευθυντή;`, είτε τέλος

λογική ερώτηση (boolean query), όπως: `Ποιοί υπάλληλοι είναι γυναίκες είτε/και έχουν θέση διευθυντή;`.

Σημειώνεται ότι η λογική ερώτηση αποτελεί ευρύτερο τύπο από την ερώτηση μερικής ταύτισης και την ερώτηση επακριβούς ταύτισης, επειδή δέχεται και τους τρεις λογικούς τελεστές (δηλαδή, and, or, not). Είναι προφανές ότι οι ερωτήσεις αυτές τίθενται σε σύστημα ενός μόνο αρχείου. Ωστόσο, είναι δυνατόν να τεθούν ερωτήσεις που απαιτούν συνδυασμό δεδομένων από δύο ή περισσότερα αρχεία. Έτσι προκύπτει ένας ακόμη τύπος:

ερώτηση σύνδεσης (join query), όπως: `Ποιοί υπάλληλοι παίρνουν μεγαλύτερο μισθό από τους διευθυντές τους;`.

Επίσης, σημειώνεται ότι η προηγούμενη ερώτηση χρησιμοποιεί ένα μόνο αρχείο αλλά συνδυάζει δύο αντίγραφα του και είναι ιδιαίτερα χρονοβόρα. Στο βιβλίο αυτό δεν θα εξετασθούν συστήματα πολλών αρχείων, όμως σε επόμενο κεφάλαιο θα γίνει αναφορά και σε άλλου τύπου ερωτήσεις. Όλες οι προηγούμενες ερωτήσεις υποστηρίζονται άνετα (δηλαδή, με φιλικό τρόπο) από γλώσσες ερωταπαντήσεων, που παρέχονται σε όλα τα συστήματα



διαχείρισης βάσεων δεδομένων (όπως, η γλώσσα SQL). Ο προγραμματιστής συστήματος κατά τη φάση του σχεδιασμού πρέπει να προνοήσει και να οργανώσει κατά τον καλύτερο τρόπο τα αρχεία, έτσι ώστε να απαντώνται αποτελεσματικά οι πιο συχνά υποβαλλόμενες ερωτήσεις.

Με τον όρο **διατήρηση** (maintenance) του αρχείου εννοείται η ανανέωση του λόγω **εισαγωγών** (inserts), **διαγραφών** (deletes) ή **ενημερώσεων** (updates) εγγραφών. Το πρόγραμμα διατήρησης του αρχείου εκτελεί σειριακά ή τυχαία τη διατήρηση ανάλογα με την οργάνωση του κύριου αρχείου. Επίσης, ο προγραμματιστής συστήματος δεν πρέπει να ξεχνά ότι πιθανώς σε μία σύνθετη εφαρμογή μία αλλαγή μπορεί να αναφέρεται σε περισσότερα από ένα αρχεία, επειδή υπάρχει **επανάληψη δεδομένων** (data duplication). Σε ένα σύστημα διαχείρισης βάσεων δεδομένων τα πράγματα είναι απλούστερα επειδή το σύστημα, που λειτουργεί ως μεσάζων μεταξύ αρχείων και χρήστη, γνωρίζει τη δομή των αρχείων. Έτσι ο χρήστης δεν ασχολείται με λεπτομέρειες, όπως σε ποιό αρχείο ανήκει ένα συγκεκριμένο πεδίο, τί τύπου είναι το κάθε πεδίο, πώς είναι οργανωμένο το αρχείο και σε ποιό μέρος του δίσκου είναι αποθηκευμένο κλπ.

Η **ταξινόμηση** (sorting) αρχείων με βάση κάποιο πεδίο είναι ένα σημαντικό πρόβλημα επειδή συναντάται συχνά στην πράξη και είναι ιδιαίτερα χρονοβόρο, είτε το αρχείο είναι αποθηκευμένο σε ταινία, είτε σε δίσκο. Επειδή το σύνολο του αρχείου δεν χωρά στην κύρια μνήμη, όσο μεγάλη και αν είναι αυτή, οι μέθοδοι ταξινόμησης αρχείων διαφοροποιούνται από τις γνωστές μεθόδους για την ταξινόμηση των δομών κύριας μνήμης (πχ. ταξινόμηση φυσαλίδας, ταξινόμηση εισαγωγής, ταξινόμηση σωρού κλπ.). Επίσης, σε αντίθεση με τις μεθόδους ταξινόμησης δομών κύριας μνήμης, το κριτήριο κόστους δεν είναι ο αριθμός των συγκρίσεων/ανταλλαγών των κλειδιών, αλλά ο αριθμός των προσπελάσεων στη δευτερεύουσα μνήμη. Αυτό ισχύει γιατί η τάξη μεγέθους του χρονικού κόστους των πράξεων αυτών είναι 1:1000.

Η **συγχώνευση** (merging) είναι μία διαδικασία που γίνεται συχνά σε εφαρμογές αρχείων, όπως για παράδειγμα όταν το αρχείο συναλλαγών συνδυάζεται με το κύριο αρχείο. Αν υποθεθεί ότι και τα δύο αρχεία είναι διατεταγμένα κατά αύξουσα τάξη του πρωτεύοντος κλειδιού, τότε η συγχώνευση επιτυγχάνεται με δύο σαρώσεις/αναγνώσεις των αρχείων και την αντίστοιχη αποθήκευση στο μαγνητικό μέσο.

Οι δομές αρχείων μαζί με τις αντίστοιχες διεργασίες τους αποτελούν αυτό που στην ορολογία των βάσεων δεδομένων αποκαλείται **μέθοδοι προ-**

σπέλασης (access methods). Οι μέθοδοι προσπέλασης και οι τεχνικές επεξεργασίας των ερωτήσεων αποτελούν αναπόσπαστο κομμάτι του φυσικού επιπέδου κάθε εμπορικού συστήματος διαχείρισης βάσεων δεδομένων. Από την άποψη αυτή, το υλικό του βιβλίου αυτού είναι μία πρώτη εισαγωγή στα θέματα φυσικής οργάνωσης των βάσεων δεδομένων, όπως δηλώνει και ο υπότιτλος του βιβλίου.

## 1.4 Ασκήσεις

<1> Μία εγγραφή αποτελείται από τα εξής πεδία με τα αντίστοιχα μήκη: Όνομα (30 χαρακτήρες), Κωδικός (6), Διεύθυνση οδού (26), Πόλη (16), Ταχυδρ. Κωδικός (5). Ποιό είναι το μέγεθος του αρχείου για ένα αρχείο 15.000 εγγραφών; Ποιά χαρακτηριστικά ή συνδυασμός χαρακτηριστικών θα μπορούσε να αποτελέσει μία καλή εκλογή για πρωτεύον κλειδί;

<2> Συχνά οι εταιρείες διατηρούν ταχυδρομικές λίστες πελατών και συλλέγουν τα στοιχεία τους από διάφορες πηγές. Αυτή η πρακτική οδηγεί σε διπλές και τριπλές αποθηκεύσεις του ίδιου προσώπου με αποτέλεσμα οικονομικό κόστος, αλλά και ενόσκληση του πελάτη. Να βρεθεί ένας αλγόριθμος που να εντοπίζει τις εγγραφές αυτές υιοθετώντας τις εξής παραδοχές:

- οι διπλοεγγραφές έχουν το ίδιο όνομα αλλά με παραλλαγές, όπως Γ.Μανωλόπουλος, Ιωάννης Μανωλόπουλος, Ι.Π.Μανωλόπουλος κλπ.,
- οι ταχυδρομικοί κωδικοί διαφέρουν,
- οι διευθύνσεις οδών μπορεί να είναι διαφορετικού τύπου, όπως Γ.Θεοτοκά 3, Θεοτοκά 3, Γεωργίου Θεοτοκά 3 κλπ., και τέλος,
- η πόλη και ο ταχυδ. κωδικός είναι ίδιοι για τις διπλοεγγραφές.

<3> Ποιά οργάνωση και γιατί είναι πιο ιδανική για κάθε μία από τις εξής εφαρμογές: Γενεαλογικό δένδρο, Τηλεφωνικός κατάλογος, Οργανόγραμμα εταιρείας, Ιατρικό δεδομένα, Σύστημα αεροπορικών κρατήσεων, Αρχεία ασφαλισμένων ΙΚΑ.