



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

---

## Εισαγωγή στις Βάσεις Δεδομένων II

**Ενότητα:** Εισαγωγή στις Βάσεις Δεδομένων I Δεδομένα

Κωστής Πηγουνάκης

Τμήμα Οικονομικών Επιστημών

# 1. Εισαγωγή

## 1.1. Επικοινωνία, Πληροφορία και Δεδομένα

### 1.1.1. Η έννοια της Πληροφορίας

Στη σημερινή εποχή που κατακλύζεται από τα Μέσα Μαζικής Επικοινωνίας (ΜΜΕ) και τις Τεχνολογίες Πληροφορίας και Επικοινωνίας (ΤΠΕ), γίνεται ακόμη σοβαρή ακαδημαϊκή συζήτηση πάνω στο τι είναι Επικοινωνία. Πάντως σαν ορισμός εργασίας είναι γενικά αποδεκτό ότι **η επικοινωνία είναι η ανταλλαγή πληροφορίας**. Η διαδικασία της επικοινωνίας προϋποθέτει την αποστολή και λήψη ενός μηνύματος μεταξύ ενός πομπού και ενός δέκτη, όπου και οι δύο χρησιμοποιούν ένα **κοινό κώδικα**. Το μήνυμα που μεταβιβάζεται από τον πομπό στο δέκτη συνίσταται από ένα σύνολο πληροφοριών.

Οι πληροφορίες αυτές μπορεί να είναι οποιασδήποτε μορφής, και να αφορούν σκέψεις, περιγραφές, ή ακόμη και συναισθήματα. Μπορούν να παραμένουν οι ίδιες - **στατικές** - ή να αλλάζουν με την πάροδο του χρόνου - **δυναμικές**. 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.

Έτσι, *η πληροφορία ορίζεται ως στοιχείο γνώσης που μετουσιώνει και δίνει αξία στα πράγματα, τα οποία αποκτούν σημασία με αυτόν τον τρόπο*. Πρέπει πάντως να γίνει ο διαχωρισμός μεταξύ πληροφορίας και γνώσης: η πληροφορία είναι το μέγεθος που, καθώς αυξάνει και επεξεργάζεται, προσδίδει σημασία στα πράγματα και τα γεγονότα και παράγει γνώση. Αυτό μπορεί να γίνει καταληπτό στο σημερινό τρόπο ζωής που, ενώ κατακλυζόμαστε από πληροφορίες, η επεξεργασία του και τελικά η παραγωγή γνώσης δεν είναι αυτονόητη.

Στη Φυσική η πληροφορία σχετίζεται με τις έννοιες της εντροπίας και της ενέργειας. Το να μπει ένα άτακτο σύστημα με ανακατεμένα μόρια σε τάξη (οργάνωση στα μόρια), να προκύψει δηλαδή πληροφορία από την επεξεργασία των δεδομένων του καθενός μορίου, χρειάζεται δαπάνη ενέργειας. Για ένα κλειστό σύστημα, το οποίο δεν ανταλλάσει ενέργεια με το περιβάλλον του, η αύξηση της οργάνωσης και άρα της πληροφορίας δεν είναι δυνατή σύμφωνα με το δεύτερο νόμο της Θερμοδυναμικής.

Ο Claude E. Shannon με την εργασία του *Μαθηματική Θεωρία Επικοινωνίας* (1948) πάνω στην επεξεργασία σήματος έδωσε το έναυσμα για τη *Θεωρία της Πληροφορίας*, στην οποία η πληροφορία συνδέεται με μια γενικότερη έννοια εντροπίας (εντροπία της θεωρίας της πληροφορίας) και όχι με τη θερμοδυναμική εκδοχή της. Η θεωρία της πληροφορίας εξετάζει την πληροφορία με όρους στατιστικής. Ακριβέστερα, εξετάζονται στατιστικά οι φορείς πληροφορίας, όπως τα σύμβολα και τα σήματα και όχι η ίδια η πληροφορία.

### 1.1.2. Πληροφορία και Δεδομένα

Η έννοια της πληροφορίας ορίζεται συχνά σε σχέση με την έννοια των **δεδομένων (data)**. Οι δύο έννοιες πολλές φορές συγχέονται ακριβώς γιατί αποκτώντας τα δεδομένα σημασία μεταπίπτουν σε πληροφορία. Τα δεδομένα δεν είναι πληροφορία, όταν δεν έχουν λάβει από τη νόηση συγκεκριμένη σημασία, ενώ η πληροφορία είναι δεδομένα με σημασία, δεδομένα που έχουν «ουσιαστικό» περιεχόμενο.

Η μετάβαση από τα δεδομένα στην πληροφορία μπορεί να αφορά την επίλυση κάποιου συγκεκριμένου ζητήματος, την απάντηση κάποιου συγκεκριμένου ερωτήματος και γενικότερα παραπέμπει σε καινούριο στοιχείο γνώσης για κάτι.

Σε απλοποιημένη ή σχηματική μορφή, **δεδομένα + σημασία = πληροφορία**

Σε αντίθεση με τα δεδομένα η πληροφορία έχει ορισμένο νόημα και είναι οργανωμένη για συγκεκριμένο σκοπό. Με την προσθήκη αξιών στα δεδομένα, με διαμόρφωση, οργάνωση, μαθηματική ή στατιστική ανάλυση, διόρθωση λαθών ή συμπίεση παράγεται η πληροφορία. Τη διαδικασία αυτή την ονομάζουμε **επεξεργασία δεδομένων**.

Στην Πληροφορική συχνά δεν γίνεται διάκριση ανάμεσα στα δεδομένα και την πληροφορία. Έτσι, τα δεδομένα 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.

*Ενώ ένας δεκαψήφιος αριθμός είναι μάλλον ένα στοιχείο δεδομένων χωρίς νόημα, η σύνδεση του με ένα ονοματεπώνυμο σε έναν τηλεφωνικό κατάλογο ενδεχομένως αποτελεί χρήσιμη πληροφορία.*

*Τα εικονοστοιχεία (pixels) που έχουν ορισμένο χρώμα στην επιφάνεια μιας οθόνης θεωρούνται απλά δεδομένα και μπορούν να αποδοθούν, π.χ. με 5 αριθμούς. Οι δύο εκφράζουν τη θέση του εικονοστοιχείου πάνω στην οθόνη (X,Y) και οι επόμενοι τρεις την ανάλυση του χρώματος (R,G,B). Όμως η σύνθεση πολλών εικονοστοιχείων που σχηματίζουν ένα ανθρώπινο πρόσωπο, μας δίνουν επιπλέον πληροφορία. Η πληροφορία αυτή μπορεί να είναι εξαιρετικά σύνθετη, όπως η ταυτοποίηση του προσώπου, τα συναισθήματά του κλπ.*

Η διαδικασία επιλογής του κατάλληλου είδους αναπαράστασης ώστε να μπορεί να αποθηκευτεί η επιθυμητή πληροφορία χωρίς να υπάρξουν απώλειες και χωρίς να γίνει σπατάλη πόρων ονομάζεται **Μοντελοποίηση Δεδομένων (Data Modeling)**.

### 1.1.3. Αδόμητα δεδομένα

Καθώς προσπαθούμε να διακρίνουμε τις διαφορές μεταξύ πληροφορίας και δεδομένων, πρέπει να κάνουμε αναφορά και στην έννοια των **αδόμητων ή ακατέργαστων δεδομένων (raw data)**. Συνήθως πρόκειται για δεδομένα που συλλέγονται από κάποια πηγή χωρίς επεξεργασία και αναφέρονται και ως **πρωτεύοντα δεδομένα (primary data)**. Τα αδόμητα δεδομένα μπορεί να προέρχονται, π.χ., από χειρονακτική εισαγωγή στοιχείων σε ένα πρόγραμμα. Έτσι, τα αδόμητα δεδομένα εν γένει περιέχουν σφάλματα και διαφορετική ή καθόλου *μορφοποίηση (format)*: μια εισαγωγή δεδομένων ημερομηνίας μπορεί να περιέχει τις παρακάτω μορφές: "31<sup>st</sup> January 1999", "31/01/1999", "31/1/99", "31 Jan". Άπαξ και αντληθούν τα αδόμητα δεδομένα, πρέπει να υποστούν επεξεργασία για να αποκτήσουν ενιαία μορφή, απαλλαγμένη από σφάλματα, ώστε να παρέχουν πραγματική πληροφορία.

## 1.2. Τύποι Δεδομένων

Στον προγραμματισμό τα δεδομένα χαρακτηρίζονται από τον **τύπο** τους (**data type**). Ο τύπος δεδομένων είναι μια ταξινόμηση που καθορίζει :

- τις πιθανές **τιμές** που μπορεί να έχουν τα δεδομένα μας (π.χ. ακέραιος αριθμός ή πραγματικός, συμβολοσειρά κλπ)
- τις **πράξεις** που μπορούν να γίνουν με τις τιμές αυτού του τύπου
- το **νόημα** των δεδομένων, και
- τον **τρόπο αποθήκευσης** των δεδομένων.

Έτσι μπορούμε να πούμε ότι ο τύπος των δεδομένων είναι μια τυπική περιγραφή των τιμών που μπορεί να αποθηκευτούν σε μια σταθερά ή μεταβλητή καθώς και των πράξεων που μπορεί να εκτελεστούν με τις τιμές αυτές.

Οι τύποι δεδομένων χρησιμοποιούνται εντός **συστημάτων τύπων**, τα οποία προσφέρουν διαφορετικούς τρόπους για τον ορισμό, την ανάπτυξη και τη χρήση τους και διαφοροποιούνται ως προς τους βαθμούς ασφάλειας των τύπων. Έτσι, είναι άλλο το σύστημα δεδομένων της γλώσσας C και άλλο της PHP. Ο κάθε τύπος δεδομένων αναπαριστά και έναν περιορισμό για το σύστημα τύπων δεδομένων, περιγράφοντας μια αναπαράσταση που μεταφράζεται σε μια δομή τιμών ή αντικειμένων τα οποία αποθηκεύονται στη μνήμη του υπολογιστή. Το σύστημα τύπων δεδομένων χρησιμοποιεί την πληροφορία των τύπων για να ελέγξει την ορθότητα των προγραμμάτων που προσπελαίνουν ή χειρίζονται τα δεδομένα.

### 1.2.1. Κατηγοριοποίηση τύπων δεδομένων

Σχεδόν όλες οι γλώσσες προγραμματισμού περιέχουν την έννοια του τύπου δεδομένων, αν και η ορολογία μπορεί να διαφέρει από γλώσσα σε γλώσσα. Συνήθεις τύποι δεδομένων μπορεί να είναι:

- integers - ακέραιοι,

- booleans – λογικοί ή δεδομένων αληθείας,
- characters – χαρακτήρες ,
- floating-point numbers – αριθμοί κινητής υποδιαστολής (πραγματικοί αριθμοί),
- alphanumeric strings – αλφαριθμητικές συμβολοσειρές.

Επίσης, οι περισσότερες γλώσσες προγραμματισμού επιτρέπουν τον ορισμό πρόσθετων τύπων δεδομένων που προκύπτουν συνήθως με συνδυασμούς των υφιστάμενων τύπων. Παρακάτω θα δούμε τους κυριότερους τύπους δεδομένων.

### Τύποι Μηχανής

Όλα τα δεδομένα στους υπολογιστές που βασίζονται στην ψηφιακή ηλεκτρονική αναπαρίστανται στο κατώτερο επίπεδο ως **bits**, δηλ. εναλλακτικές καταστάσεις (0 ή 1). Η μικρότερη μονάδα μνήμης δεδομένων είναι συνήθως μια ομάδα από bits που ονομάζονται **byte** και συνήθως πρόκειται για μια οκτάδα από bits (1 byte = 8 bits). Η μονάδα που μπορεί να επεξεργαστεί ένας υπολογιστής σε επίπεδο κώδικα μηχανής λέγεται **λέξη (word)** και σήμερα (2012) είναι μήκους 32 ή 64bits.

### Λογικός Τύπος (Boolean)

Τα δεδομένα αληθείας αποτυπώνονται δύο καταστάσεις : **ΑΛΗΘΕΣ** – **TRUE** και **ΨΕΥΔΕΣ** – **FLASE**. Τα δεδομένα αυτά αποτελούν το **λογικό τύπο ή τύπο Boole (Boolean type)**.

Παρότι μόνο δύο τιμές είναι πιθανές, σπάνια αναπαρίστανται στους υπολογιστές με ένα μόνο δυαδικό ψηφίο για λόγους αποδοτικότητας.

Πολλές γλώσσες προγραμματισμού δεν είναι εφοδιασμένες με ρητό τύπο αληθείας. Αντί αυτού ερμηνεύουν το 0 (για παράδειγμα) ως ψευδές και τις υπόλοιπες τιμές ως αληθείς .

### Αριθμητικοί Τύποι

Αριθμητικά χαρακτηρίζονται τα δεδομένα που αναπαριστούν αριθμούς. Το ενδιαφέρον σε αυτούς τους τύπους δεδομένων είναι ότι μπορούμε να επιλέξουμε ή να ορίσουμε το μέγεθος του τύπου δεδομένων ανάλογα με τη χρήση που θέλουμε. Έτσι, για παράδειγμα, μπορούμε να επιλέξουμε έναν τύπο που να είναι ακέραιος ή φυσικός (θετικός ακέραιος) και να έχει τιμές, π.χ. 0-255.

Για τη MySQL οι αριθμητικοί τύποι δεδομένων είναι

ΟΝΟΜΑΣΙΑ ΤΥΠΟΥ	ΘΕΣΕΙΣ ΑΠΟΘΗΚΕΥΣΗΣ	ΕΥΡΟΣ ΤΙΜΩΝ
<b>TINYINT</b>	1 byte	-128 ΕΩΣ 127 [0 ΕΩΣ 255 ΧΩΡΙΣ ΠΡΟΣΗΜΟ - UNSIGNED]
<b>SMALLINT</b>	2 bytes	-32,768 ΕΩΣ 32,767 [0 ΕΩΣ 65,535]
<b>MEDIUMINT</b>	3 bytes	-8,388,608 ΕΩΣ 8,388,607 [0 ΕΩΣ 16,777,215]
<b>INT</b>	4 bytes	-/+2.147E+9 [0 ΕΩΣ 4.294E+9]
<b>BIGINT</b>	8 bytes	-/+9.223E+18 [0 ΕΩΣ 18.45E+18]
<b>FLOAT</b>	4 bytes	Min=+/-1.15E-38 Max=+/-3.403E+38
<b>DOUBLE</b>	8 bytes	Min=+/-2.225E-308 Max=+/-1.798E+308
<b>DECIMAL [ (M, [D] ) ]</b> αποθηκεύονται ως συμβολοσειρά	M+2	Max εύρος = το εύρος <b>DOUBLE</b>

## Τύποι Κειμένου και Συμβολοσειρών

Τα δεδομένα κειμένου αποτελούν μια από τις γενικότερες κατηγορίες δεδομένων γιατί ως τέτοια μπορούν να θεωρηθούν σχεδόν όλα τα δεδομένα που χειριζόμαστε. Η διαφοροποίηση σε άλλους τύπους δεδομένων γίνεται γιατί οι τύποι δεδομένων κειμένου δεν έχουν ευελιξία π.χ. στο να ορίσουμε αριθμητικές πράξεις με αυτά.

Ο πιο απλός τύπος δεδομένων κειμένου είναι ο **αλφαριθμητικός χαρακτήρας (alphanumeric character)** ή απλά χαρακτήρας. Μπορεί να είναι ένα γράμμα της αλφαβήτου (λατινικής ή άλλης), ένα αριθμητικό ψηφίο (0-9), το κενό, τα σημεία στίξης, ειδικά σύμβολα όπως @ # \$ & κλπ. Οι χαρακτήρες απαιτούν ένα «πλάτος» για την αποθήκευσή τους. Έτσι, για το σύνολο των χαρακτήρων ASCII το πλάτος για κάθε χαρακτήρα είναι 7-bit, αλλά το σύνολο αυτό δεν μπορεί να αποδώσει χαρακτήρες που δεν βασίζονται στο ελληνο-λατινικό αλφάβητο. Για να είναι δυνατή η απόδοση γλωσσών όπως τα αραβικά, τα εβραϊκά ή τα κινέζικα χρησιμοποιούνται σύνολα με πλάτος 8-bit (utf-8) ή 16-bit (utf-16).

Μια σειρά από τέτοιους χαρακτήρες καλείται **αλφαριθμητική συμβολοσειρά (alphanumeric string)** ή απλά συμβολοσειρά. Η συμβολοσειρά χαρακτηρίζεται από το «μήκος» των χαρακτήρων που περιλαμβάνει.

Στους τύπους κειμένου ανήκουν και τα δεδομένα **απαρίθμησης (enumeration)**, τα οποία είναι μια ιδιαίτερη κατηγορία συμβολοσειρών. Το χαρακτηριστικό τους είναι ότι μπορούν να πάρουν τιμές από ένα προκαθορισμένο σύνολο, π.χ. τα χρώματα των χαρτιών της τράπουλας, τα ονόματα των ημερών ή των μηνών του έτους.

Στα δεδομένα κειμένου περιλαμβάνονται και τα **δυναμικά μεγάλα αντικείμενα (Binary Large Objects – BLOBs)**, τα οποία αποτελούν τη ψηφιακή αποτύπωση *πολυμεσικών αρχείων* (αρχείων κειμένου, εικόνων, ήχου ή βίντεο).

Για τη MySQL οι τύποι δεδομένων κειμένου είναι

<b>CHAR (size)</b>	Αποθηκεύει ένα σταθερό μήκος συμβολοσειράς Το μέγεθος (μήκος) ορίζεται σε παρένθεση. Μπορεί να αποθηκεύσει μέχρι 255 χαρακτήρες.
<b>VARCHAR (size)</b>	Αποθηκεύει ένα μεταβλητό μήκος συμβολοσειράς Το μέγιστο μέγεθος (μήκος) ορίζεται σε παρένθεση. Μπορεί να αποθηκεύσει μέχρι 255 χαρακτήρες. <b>Σημείωση:</b> Αν το μήκος της συμβολοσειράς είναι μεγαλύτερο από 255 χαρακτήρες, μετατρέπεται σε τύπο TEXT
<b>TINYTEXT</b>	Αποθηκεύει το πολύ 255 χαρακτήρες
<b>TEXT</b>	Αποθηκεύει το πολύ 65,535 χαρακτήρες
<b>BLOB</b>	Binary Large Objects. Αποθηκεύει το πολύ 65,535 bytes δεδομένων
<b>MEDIUMTEXT</b>	Αποθηκεύει το πολύ 16,777,215 χαρακτήρες
<b>MEDIUMBLOB</b>	Binary Large Objects. Αποθηκεύει το πολύ 16,777,215 bytes δεδομένων
<b>LONGTEXT</b>	Αποθηκεύει το πολύ 4,294,967,295 χαρακτήρες
<b>LOBLOB</b>	Binary Large Objects. Αποθηκεύει το πολύ 4,294,967,295 bytes δεδομένων
<b>ENUM (x, y, z, etc.)</b>	Επιτρέπει την εισαγωγή λίστας πιθανών τιμών (μέχρι 65535 τιμές). Αν η τιμή που εισάγεται δεν είναι μεταξύ των πιθανών τιμών, αποθηκεύεται η κενή τιμή.
<b>SET</b>	Παρόμοιος τύπος με το ENUM εκτός του ότι το SET μπορεί να περιέχει μέχρι 64 τιμές και μπορεί να αποθηκεύσει περισσότερες από μία

## Τύποι Χρόνου

Τα δεδομένα χρόνου έχουν συγκεκριμένο τρόπο αποτύπωσης της ώρας, της ημερομηνίας ή του χρόνου με σκοπό την εύκολη διαχείριση τους. Η μετατροπή δεδομένων από ένα τύπο χρόνου σε άλλον είναι εφικτή, αλλά μπορεί να προκαλέσει απώλεια πληροφορίας.

Για τη MySQL οι τύποι δεδομένων χρόνου είναι

<b>DATE ( )</b>	Ημερομηνία με μορφή οργάνωσης: ΕΕΕΕ-ΜΜ-ΗΗ. Υποστηρίζονται ημερομηνίες από '1000-01-01' έως '9999-12-31'
<b>DATETIME ( )</b>	Ημερομηνία και ώρα με μορφή οργάνωσης: ΕΕΕΕ-ΜΜ-ΗΗ ΩΩ:ΛΛ:ΔΔ. Υποστηρίζονται ημερομηνίες από '1000-01-01 00:00:00' έως '9999-12-31 23:59:59'
<b>TIMESTAMP ( )</b>	Τρόπος αποτύπωσης της τρέχουσας ώρας που προέρχεται από το λειτουργικό σύστημα UNIX με μορφή οργάνωσης: ΕΕΕΕ-ΜΜ-ΗΗ ΩΩ:ΛΛ:ΔΔ. Υποστηρίζονται χρόνοι από '1970-01-01 00:00:01' έως '2038-01-09 03:14:07'
<b>TIME ( )</b>	Χρόνος με μορφή οργάνωσης: ΩΩ:ΛΛ:ΔΔ. Υποστηρίζονται χρόνοι από '-838:59:59' έως '838:59:59'
<b>YEAR ( )</b>	Έτος με 2 ή 4 ψηφία. Για 4 ψηφία υποστηρίζονται τα έτη : 1901 έως 2155 ενώ για τα 2 ψηφία τα έτη από 70 έως 69, αναπαριστώντας τα έτη από 1970 έως 2069.

## Απαριθμήσεις

Ο τύπος της **απαρίθμησης (enumeration)** αφορά σε μεταβλητές που παίρνουν δεδομένες τιμές που διαφέρουν μεταξύ τους, οι οποίες μπορούν να συγκριθούν, αλλά δεν έχουν απαραίτητα κάποια συγκεκριμένη αναπαράσταση στην μνήμη του υπολογιστή.

Οι μεταγλωττιστές και οι διερμηνευτές τις αναπαριστούν αυθαίρετα. Στη MySQL αποτελούν μέρος του τύπου κειμένου.

Για παράδειγμα, τα τέσσερα χαρτιά σε μία τράπουλα θα μπορούσαν να είναι τέσσερις απαριθμητές με ονόματα ΣΠΑΘΙ, ΚΑΡΩ, ΜΠΑΣΤΟΥΝΙ, ΚΟΥΠΑ, που θα ανήκουν σε μία απαρίθμηση με όνομα τράπουλα. Αν η μεταβλητή V δηλωθεί να έχει τράπουλα για τύπο, μπορούμε να αναθέσουμε οποιαδήποτε από αυτές τις τιμές σε αυτή την μεταβλητή.

## Σύνθετοι Τύποι

Οι τύποι δεδομένων που αναφέρθηκαν παραπάνω αφορούν στοιχειώδεις μορφές δεδομένων. Τα δεδομένα αυτά συσχετίζονται με μια και μοναδική ατομική τιμή και δεν υπάρχει λόγος να αναλυθούν σε μικρότερες μονάδες.

Στις πιο συνηθισμένες όμως περιπτώσεις, τα δεδομένα υφίστανται ευκολότερη και αποτελεσματικότερη διαχείριση εάν ομαδοποιηθούν σε μεγαλύτερες οντότητες, τις οποίες χειριζόμαστε ενιαία. Έτσι, από τους απλούς βαθμωτούς τύπους δημιουργούνται οι σύνθετοι τύποι

Ας πάρουμε για παράδειγμα την περίπτωση μιας εταιρείας. Εάν θελήσουμε να δημιουργήσουμε ένα πρόγραμμα στο οποίο θα καταχωρούμε τα στοιχεία των εργαζομένων, τότε ας δεχθούμε ότι τα στοιχεία εκείνα που θα κρατάμε για κάθε εργαζόμενο είναι, π.χ. 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 (αριστερά) στα οποία θα καταχωρεί τα στοιχεία του εργαζόμενου, είτε να δημιουργήσει ένα σύνθετο τύπο δεδομένων – τον οποίο ας ονομάσουμε *Employee* (δεξιά) – που θα περιέχει τα τέσσερα πεδία που περιγράψαμε παραπάνω:



Είναι προφανές, ότι η δεύτερη προσέγγιση είναι πιο αποτελεσματική σε σχέση με την πρώτη, διότι τα τέσσερα δεδομένα που σχηματίζουν τον σύνθετο τύπο δεδομένων *Employee*, υφίστανται διαχείριση, σαν να ήταν ένας απλός τύπος δεδομένων. Αυτό 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)**, για τις οποίες θα γίνει εκτενής αναφορά στα επόμενα κεφάλαια.

### 1.2.2. Δομές Δεδομένων

Η έννοια της **δομής δεδομένων (data structure)** αναφέρεται στους διαφορετικούς δυνατούς τρόπους οργάνωσης και αποθήκευσης δεδομένων, ώστε αυτά να μπορούν να χρησιμοποιηθούν αποδοτικά στον προγραμματισμό. Σε συγκεκριμένες εφαρμογές χρησιμοποιούνται συγκεκριμένες δομές δεδομένων, δηλαδή αυτές που είναι οι πιο αποδοτικές αλγοριθμικά για το κάθε είδος εφαρμογής.

Δομές δεδομένων χρησιμοποιούνται σχεδόν σε κάθε πρόγραμμα ή σύστημα λογισμικού. Παρέχουν έναν τρόπο αποδοτικής διαχείρισης τεράστιου όγκου δεδομένων, όπως μεγάλες βάσεις δεδομένων και υπηρεσίες ευρετηρίου στο διαδίκτυο. Οι αποδοτικές δομές δεδομένων θεωρούνται συχνά ιδιαίτερα σημαντικές στη δημιουργία ενός αποδοτικού αλγορίθμου, σε τέτοιο βαθμό, ώστε κάποιες μέθοδοι σχεδίασης και γλώσσες προγραμματισμού δίνουν έμφαση σε δομές δεδομένων, παρά σε αλγορίθμους, ως το βασικό κριτήριο σχεδίασης λογισμικού.

Οι δομές δεδομένων διακρίνονται επίσης σε **στατικές** και σε **δυναμικές**:

- Στις στατικές δομές το πλήθος των στοιχείων είναι σταθερό και καθορίζεται στον ορισμό του τύπου στο τμήμα δηλώσεων πχ ο πίνακας.



- Στις δυναμικές δομές το πλήθος των στοιχείων της δομής καθορίζεται κατά τη διάρκεια εκτέλεσης του προγράμματος.

### 1.2.3. Βάση Δεδομένων και Σύστημα Διαχείρισης Βάσεων Δεδομένων

Μέχρι τώρα είδαμε την έννοια των δεδομένων και τον τρόπο οργάνωσής τους σε σύνθετες δομές. Προχωρώντας μπορούμε να πούμε ότι **Βάση Δεδομένων - ΒΔ (Database - DB) είναι μια συλλογή δεδομένων με οργάνωση τέτοια ώστε τα δεδομένα να προσπελούνται, να διαχειρίζονται και να ενημερώνονται εύκολα και αποδοτικά.**

Η βάση δεδομένων, ως όρος, αποδίδει σωστά τα δεδομένα και τις δομές δεδομένων που τα υποστηρίζουν, αλλά συχνά συγχέεται με το όρο **Σύστημα Διαχείρισης Βάσεων Δεδομένων – ΣΔΒΔ (DataBase Management System - DBMS)**, ο οποίος χρησιμοποιείται για τις ειδικές εφαρμογές λογισμικού που χρησιμοποιούνται για την υλοποίηση, τη συντήρηση, την επεξεργασία και γενικά τη διαχείριση των βάσεων δεδομένων. Η βάση δεδομένων και το ΣΔΒΔ αποτελούν ένα **Σύστημα Βάσης Δεδομένων - ΣΒΔ**.

Κατά καιρούς έχουν εμφανιστεί στην αγορά πολλά ΣΔΒΔ, από πολλές εταιρείες, τα οποία ποικίλλουν, τόσο στη σχεδίαση και διαχείριση της βάσης, όσο και στις δυνατότητες που προσφέρουν. Τα περισσότερα ΣΔΒΔ υποστηρίζουν πολλούς χρήστες, πολλές ταυτόχρονες προσπελάσεις πάνω στα ίδια δεδομένα, καθώς και απομακρυσμένη πρόσβαση μέσω δικτύου.

Από τα πιο γνωστά και ευρέως χρησιμοποιούμενα εμπορικά ΣΔΒΔ είναι τα **Microsoft Access, Microsoft SQL Server, Oracle** και **Sybase/SAP** ενώ στο χώρο του ανοικτού λογισμικού δύο από τα ευρύτερα χρησιμοποιούμενα ΣΔΒΔ είναι τα **MySQL** και **Postgress**. Από τα μοντέρνα συστήματα διαχείρισης, το **Microsoft Access** χρησιμοποιείται κυρίως για την ανάπτυξη και διαχείριση βάσεων δεδομένων σε προσωπικούς υπολογιστές και σε εφαρμογές μικρού μεγέθους ενώ τα **MS SQL Server, Oracle** και **Sybase/SAP**, χρησιμοποιούνται σε εφαρμογές μεσαίου και μεγάλου μεγέθους, κυρίως τα δύο τελευταία. Τα **MySQL** και **Postgress** εστιάζονται σε εφαρμογές διαδικτύου σε συνεργασία με γλώσσες προγραμματισμού όπως η PHP.

## 1.3. Ερωτήσεις Κατανόησης

1. Τι είναι τα δεδομένα ;
2. Τι εννοούμε λέγοντας ότι η πληροφορία είναι στατική και τι δυναμική;
3. Τι είναι η μοντελοποίηση δεδομένων;
4. Τι είναι τα αδόμητα δεδομένα;
5. Τι είναι οι τύποι δεδομένων και τι τα συστήματα τύπων δεδομένων;
6. Τι είναι οι δομές δεδομένων και ποια η σχέση τους με τις βάσεις δεδομένων ;
7. Τι είναι οι στατικές και τι οι δυναμικές δομές δεδομένων;
8. Τι είναι η βάση δεδομένων;
9. Τι είναι ένα Σύστημα Διαχείρισης ΒΔ; Σε τι διαφέρει από τη ΒΔ ;

## Σημειώματα

### Σημείωμα αναφοράς

Copyright Πανεπιστήμιο Κρήτης, Κωστής Πηγουνάκης, 2015. Κωστής Πηγουνάκης. «Εισαγωγή στις Βάσεις Δεδομένων II». Έκδοση: 1.0. Ρέθυμνο 2014.

### Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση, Όχι Παράγωγο Έργο 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

### Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή δασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.

## Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

