

# Ιατρική Πληροφορική



Δρ. Π. ΑΣΒΕΣΤΑΣ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΒΙΟΪΑΤΡΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ Τ.Ε.

## Χρήσιμοι Σύνδεσμοι



- Σημειώσεις μαθήματος:  
<http://medisp.bme.teiath.gr/eclass/courses/TIO103/>  
<https://eclass.teiath.gr/courses/TIO100/>
- E-mail:  
pasv@teiath.gr

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η δημιουργία και η διαχείριση μίας βάσης δεδομένων επιτυγχάνεται με χρήση εντολών της γλώσσας **Structured Query Language (SQL)**
- Η SQL είναι μία γλώσσα υψηλού επιπέδου και περιλαμβάνει εντολές που επιτρέπουν:
  - Τη δημιουργία πινάκων
  - Την τροποποίηση της δομής πινάκων
  - Την εισαγωγή δεδομένων σε πίνακες
  - Τη διαγραφή δεδομένων από πίνακες
  - Την ανανέωση των δεδομένων πινάκων
  - Την επιλογή δεδομένων από πίνακες με χρήση κριτηρίων

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οι εντολές της SQL χρησιμοποιούνται σε εφαρμογές που κάνουν χρήση βάσεων δεδομένων
- Οι εφαρμογές αυτές αναπτύσσονται σε κάποια γλώσσα προγραμματισμού (π.χ. JAVA, PHP, Matlab κ.λπ.)
- Στον κώδικα ενσωματώνονται εντολές SQL για τη διαχείριση δεδομένων

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### • Απόσπασμα κώδικα σε Matlab με SQL

```

dbname = 'askhsh11';
username = 'student';
password = 'student';
driver = 'com.mysql.jdbc.Driver';
dburl = ['jdbc:mysql://192.168.1.200:3306/' dbname];

% Η ΕΠΟΜΕΝΗ ΕΝΤΟΛΗ ΕΚΤΕΛΕΙΤΑΙ ΜΟΝΟ ΜΙΑ ΦΟΡΑ
javaclasspath('C:\labwork\mysql-connector-java-5.1.22\mysql-connector-java-5.1.22-bin.jar');

conn = database(dbname, username, password, driver, dburl); % ΣΥΝΔΕΣΗ ΜΕ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ
if(isconnection(conn) == 0) % ΕΛΕΓΧΟΣ ΑΝ Η ΣΥΝΔΕΣΗ ΗΤΑΝ ΕΠΙΤΥΧΗΣ
    fprintf('ΠΡΟΒΛΗΜΑ ΣΥΝΔΕΣΗΣ ΜΕ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ\n\n');
else
    setdbprefs('DataReturnFormat','structure'); % ΟΡΙΖΕΤΑΙ ΟΤΙ ΤΑ ΔΕΔΟΜΕΝΑ ΘΑ ΕΙΝΑΙ ΣΕ ΜΙΑ ΜΕΤΑΒΛΗΤΗ
    sql = 'SELECT first name, last name, weight, height FROM astheneis'; % Η ΠΡΟΣ ΕΚΤΕΛΕΣΗ ΕΝΤΟΛΗ
    X = fetch(conn, sql); % ΕΚΤΕΛΕΣΗ ΕΝΤΟΛΗΣ. ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΙΝΑΙ ΣΤΟΝ ΠΙΝΑΚΑ X
    N = length(X.first_name);
    BMI = zeros(N,1);
    c1 = 0;
    c2 = 0;
    c3 = 0;

```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### • Απόσπασμα κώδικα σε JAVA με SQL

```

import java.sql.*;
import java.text.NumberFormat;

public class DBTestApp
{
    public static void main(String args[])
    {
        String query =
            "SELECT vendor_name, invoice_number, invoice_total " +
            "FROM vendors INNER JOIN invoices " +
            "    ON vendors.vendor_id = invoices.vendor_id " +
            "WHERE invoice_total >= 500 " +
            "ORDER BY vendor_name, invoice_total DESC";

        String dbUrl = "jdbc:mysql://localhost:3306/ap";
        String username = "root";
        String password = "sesame";

        try (Connection connection = DriverManager.getConnection(
            dbUrl, username, password);
            Statement statement = connection.createStatement());

```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Απόσπασμα κώδικα σε PHP με SQL

```
<?php
$query =
    "SELECT vendor_name, invoice_number, invoice_total
    FROM vendors INNER JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
    WHERE invoice_total >= 500
    ORDER BY vendor_name, invoice_total DESC";

$dns = 'mysql:host=localhost;dbname=ap';
$username = 'root';
$password = 'sesame';

try {
    $db = new PDO($dns, $username, $password);
} catch (PDOException $e) {
    $error_message = $e->getMessage();
    echo $error_message;
    exit();
}

$rows = $db->query($query);
?>
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Σύμφωνα με τη SQL, οι ονομασίες βάσεων δεδομένων, πινάκων και πεδίων (στηλών) ονομάζονται **αναγνωριστικά (identifiers)**
- Κάθε αναγνωριστικό μπορεί να περιλαμβάνει μέχρι 64 χαρακτήρες
- Επιτρεπτοί χαρακτήρες είναι:
  - Γράμματα (a-z, A-Z)
  - Ψηφία (0-9)
  - Κάτω παύλα (\_)

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Ένα αναγνωριστικό μπορεί να ξεκινάει με ψηφίο, αλλά δεν μπορεί να αποτελείται αποκλειστικά από ψηφία
- Ένα αναγνωριστικό δεν μπορεί να περιλαμβάνει διαστήματα (spaces)
- Δεν μπορεί να χρησιμοποιηθεί ως αναγνωριστικό κάποια δεσμευμένη λέξη της SQL
- Παραδείγματα αποδεκτών αναγνωριστικών:
  - Table1
  - Users\_table
  - 2ndTable

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για τα αναγνωριστικά πεδίων (στηλών) πινάκων δεν παίζει ρόλο εάν είναι με πεζούς ή κεφαλαίους χαρακτήρες
- Για τα αναγνωριστικά πινάκων και βάσεων δεδομένων:
  - Σε υπολογιστές με λειτουργικό σύστημα Windows ή MacOS X (με HFS+) δεν παίζει ρόλο εάν τα αναγνωριστικά είναι με πεζούς ή κεφαλαίους χαρακτήρες (**case insensitive**)
  - Σε υπολογιστές με λειτουργικό σύστημα Linux ή MacOS X (χωρίς HFS+) παίζει ρόλο εάν τα αναγνωριστικά είναι με πεζούς ή κεφαλαίους χαρακτήρες (**case sensitive**)

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η SQL υποστηρίζει διάφορους τύπους δεδομένων:
  - Συμβολοσειρές σταθερού μήκους
  - Συμβολοσειρές μεταβλητού μήκους
  - Κείμενο
  - Ακέραιοι αριθμοί με πρόσημο
  - Ακέραιοι αριθμοί χωρίς πρόσημο
  - Δεκαδικοί αριθμοί σταθερής υποδιαστολής
  - Δεκαδικοί αριθμοί κινητής υποδιαστολής
  - Ημερομηνίες

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Συνήθεις τύποι δεδομένων

Κατηγορία	Τύπος	Περιγραφή
Συμβολοσειρά	CHAR(L)	Συμβολοσειρά σταθερού μήκους L χαρακτήρων (0≤L≤255)
	VARCHAR(L)	Συμβολοσειρά μεταβλητού μήκους με μέγιστο πλήθος χαρακτήρων L (0≤L≤65.535)
	TINYTEXT	Συμβολοσειρά μεταβλητού μήκους με μέγιστο πλήθος χαρακτήρων 255
	TEXT	Συμβολοσειρά μεταβλητού μήκους με μέγιστο πλήθος χαρακτήρων 65.535
	MEDIUMTEXT	Συμβολοσειρά μεταβλητού μήκους με μέγιστο πλήθος χαρακτήρων 16.777.215
	LONGTEXT	Συμβολοσειρά μεταβλητού μήκους με μέγιστο πλήθος χαρακτήρων 4.294.967.295
	ENUM(val1, val2,...)	Συμβολοσειρά με τιμή από μια λίστα επιτρεπτών τιμών. Παράδειγμα: ENUM('XRAY', 'CT', 'MRI', 'PET', 'SPECT')

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Παρατηρήσεις για τις συμβολοσειρές:
  - Συνήθως, κάθε χαρακτήρας αποθηκεύεται με ένα byte
  - Μία συμβολοσειρά με M χαρακτήρες αποθηκεύεται με M+1 bytes, όπου το πρόσθετο byte είναι για τον τερματικό χαρακτήρα
- Διαφορές CHAR(L) και VARCHAR(L):
  - Εάν χρησιμοποιείται ο τύπος CHAR(L) για ένα πεδίο, τότε εάν η τιμή αποτελείται από λιγότερους από L χαρακτήρες, οι υπόλοιποι χαρακτήρες συμπληρώνονται με διαστήματα και πάντα θα δεσμεύονται L+1 bytes
  - Εάν χρησιμοποιείται ο τύπος VARCHAR(L) για ένα πεδίο, τότε εάν η τιμή αποτελείται από λιγότερους από L χαρακτήρες, π.χ. N χαρακτήρες (N < L), τότε η συμβολοσειρά θα αποθηκευθεί με N+1 bytes

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Συνήθεις τύποι δεδομένων

Κατηγορία	Τύπος	Περιγραφή
Ακέρατοι με πρόσημο	TINYINT	Ακέρατος μεταξύ -128 και 127
	SMALLINT	Ακέρατος μεταξύ -32.768 και 32.767
	MEDIUMINT	Ακέρατος μεταξύ -8.388.608 και 8.388.607
	INT	Ακέρατος μεταξύ -2.147.483.648 και 2.147.483.647
	BIGINT	Ακέρατος μεταξύ -9.223.372.036.854.775.808 και 9.223.372.036.854.775.807
Ακέρατοι χωρίς πρόσημο	TINYINT UNSIGNED	Ακέρατος μεταξύ 0 και 255
	SMALLINT UNSIGNED	Ακέρατος μεταξύ 0 και 65.535
	MEDIUMINT UNSIGNED	Ακέρατος μεταξύ 0 και 16.777.215
	INT UNSIGNED	Ακέρατος μεταξύ 0 και 4.294.967.295
	BIGINT UNSIGNED	Ακέρατος μεταξύ 0 και 18.446.744.073.709.551.615

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Συνήθεις τύποι δεδομένων

Κατηγορία	Τύπος	Περιγραφή
Δεκαδικοί Αριθμοί	DECIMAL(g, f) ή DEC(g, f)	Δεκαδικός σταθερής υποδιαστολής: g είναι το συνολικό πλήθος ψηφίων, f είναι το πλήθος των ψηφίων μετά την υποδιαστολή
	FLOAT	Δεκαδικός κινητής υποδιαστολής απλής ακρίβειας. Μπορούν να αποθηκευθούν αρνητικοί αριθμοί από $-3,402823466 \times 10^{38}$ έως $-1,175494351 \times 10^{-38}$ , το 0 και θετικοί αριθμοί από $1,175494351 \times 10^{-38}$ έως $3,402823466 \times 10^{38}$
	DOUBLE	Δεκαδικός κινητής υποδιαστολής διπλής ακρίβειας. Μπορούν να αποθηκευθούν αρνητικοί αριθμοί από $-1,7976931348623154 \times 10^{308}$ έως $-2.2250738585072014 \times 10^{-308}$ , το 0 και θετικοί αριθμοί από $2.2250738585072014 \times 10^{-308}$ έως $1,7976931348623154 \times 10^{308}$

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Συνήθεις τύποι δεδομένων

Κατηγορία	Τύπος	Περιγραφή
Χρόνος	DATE	Ημερομηνία από 1 <sup>η</sup> Ιανουαρίου 1000 ('1000-01-01') έως 31 Δεκεμβρίου 9999 ('9999-12-31'). Η τυπική μορφοποίηση είναι 'YYYY-MM-DD'
	TIME	Ώρα από '00:00:00' έως '23:59:59'. Η τυπική μορφοποίηση είναι 'hh:mm:ss'
	DATETIME	Ημερομηνία και ώρα από '1000-01-01 00:00:00' έως '999-12-31 23:59:59'. Η τυπική μορφοποίηση είναι 'YYYY-MM-DD hh:mm:ss'
	TIMESTAMP	Ημερομηνία και ώρα στη μορφή 'YYYY-MM-DD hh:mm:ss' από '1970-01-01 00:00:00' έως '2037-12-31 23:59:59'. Χρησιμοποιείται ως χρονοσφραγίδα και λαμβάνει αυτόματα τιμή, όταν γίνεται προσθήκη ή διαγραφή γραμμών σε έναν πίνακα

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η σωστή επιλογή του τύπου δεδομένων είναι πολύ σημαντική γιατί επηρεάζει:
  - Το μέγεθος του αποθηκευτικού χώρου που θα χρησιμοποιηθεί
  - Την ταχύτητα με την οποία γίνονται οι διαδικασίες σε μία βάση δεδομένων
  - Την ακρίβεια με την οποία θα αποθηκευθούν τα δεδομένα

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Διαδικασία επιλογής σωστού τύπου δεδομένων για μία στήλη (πεδίο):
  - Καθορίζεται η κατηγορία στην οποία θα ανήκει ο τύπος δεδομένων της στήλης. Δηλαδή εάν πρόκειται για συμβολοσειρά, αριθμό ή ημερομηνία
  - Για την επιλεγμένη κατηγορία, καθορίζεται ο τύπος που ταιριάζει περισσότερο στα δεδομένα που θα καταχωρηθούν. Για παράδειγμα, εάν είναι γνωστό εκ των προτέρων ότι σε μία στήλη θα καταχωρηθούν τιμές από 1 έως 99.999, τότε ο κατάλληλος τύπος είναι ο **MEDIUMINT UNSIGNED**

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Διαδικασία επιλογής σωστού τύπου δεδομένων για μία στήλη (πεδίο):
  - Εάν πρόκειται να αποθηκευθούν δεκαδικοί αριθμοί και απαιτείται μεγάλη ακρίβεια τότε ο κατάλληλος τύπος είναι **DECIMAL**
  - Εάν η ακρίβεια δεν είναι τόσο σημαντική ή η ταχύτητα έχει την υψηλότερη προτεραιότητα, τότε ο τύπος **DOUBLE** επαρκεί.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### Παράδειγμα

Μία εφαρμογή που παρέχει προσωποποιημένο περιεχόμενο στους χρήστες της χρησιμοποιεί έναν πίνακα που κρατάει τις ακόλουθες πληροφορίες για τους χρήστες:

- Κωδικός (αύξων αριθμός με αρχή το 1)
- Όνομα
- Επώνυμο
- Email
- Όνομα χρήστη (μέγιστο πλήθος χαρακτήρων 15)
- Συνθηματικό (αποθηκεύεται κρυπτογραφημένο με σταθερό μήκος 40 χαρακτήρων)
- Ημερομηνία και ώρα εγγραφής

Αν αναμένεται το πλήθος των χρηστών να μην ξεπεράσει το 1.000.000, να δοθεί ο κατάλληλος τύπος για κάθε πεδίο του πίνακα.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### Λύση

- Ο κωδικός πρέπει να είναι ακέραιος χωρίς πρόσημο με μέγιστη τιμή 1.000.000. Συνεπώς ο κατάλληλος τύπος είναι MEDIUMINT UNSIGNED, που επιτρέπει περίπου 17 εκατομμύρια εγγραφές στον πίνακα.
- Το όνομα θα είναι συμβολοσειρά μεταβλητού πλήθους χαρακτήρων. Θεωρώντας ένα μέγιστο πλήθος 25 χαρακτήρων, ο τύπος είναι VARCHAR(25).
- Το επώνυμο θα είναι συμβολοσειρά μεταβλητού πλήθους χαρακτήρων. Θεωρώντας ένα μέγιστο πλήθος 40 χαρακτήρων, ο τύπος είναι VARCHAR(40).

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### Λύση

- Το email θα είναι συμβολοσειρά μεταβλητού πλήθους χαρακτήρων. Θεωρώντας ένα μέγιστο πλήθος 60 χαρακτήρων, ο τύπος είναι VARCHAR(60).
- Το όνομα χρήστη θα έχει το πολύ 15 χαρακτήρες. Επομένως, ο κατάλληλος τύπος δεδομένων είναι VARCHAR(15).
- Το συνθηματικό θα είναι συμβολοσειρά σταθερού πλήθους 40 χαρακτήρων. Συνεπώς, ο κατάλληλος τύπος δεδομένων είναι CHAR(40).
- Η ημερομηνία εγγραφής θα είναι τύπου DATETIME.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για λόγους ασφαλείας, ένα συνθηματικό αποθηκεύεται πάντα κρυπτογραφημένο σε μία βάση δεδομένων
- Οι πιο γνωστοί αλγόριθμοι κρυπτογράφησης είναι
  - Secure Hash Algorithm 1 (SHA-1)
  - Secure Hash Algorithm 2 (SHA-2)
- Κάθε ένας από τους αλγόριθμους μετατρέπει ένα συνθηματικό (ή οποιαδήποτε κείμενο) σε μια σειρά από bits, που ονομάζεται **hash**

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για παράδειγμα ο SHA-1 μετατρέπει οποιοδήποτε συνθηματικό σε 160 bits
- Κάθε τετράδα bits αναπαριστάται με ένα δεκαεξαδικό ψηφίο (0-9, A-F), με αποτέλεσμα να αποθηκεύεται σε μία βάση δεδομένων ως συμβολοσειρά 40 χαρακτήρων
- Ακόμα και αν δύο συνθηματικά διαφέρουν μόνο σε ένα χαρακτήρα, τα αντίστοιχα hash θα είναι εντελώς διαφορετικά

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### Παράδειγμα

Συνθηματικό	SHA-1 Hash
test1234	9bc34549d565d9505b287de0cd20ac77be1d3f2c
test123	7288edd0fc3ffcbe93a0cf06e3568e28521687bc
test12	4ff1a33e188b7b86123d6e3be2722a23514a83b4
test1	b444ac06613fc8d63795be9ad0beaf55011936ac
test	a94a8fe5ccb19ba61c4c0873d391e987982fbbd3
Test	640ab2bae07bedc4c163f679a746f7ab7fb5d1fa

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οι εντολές της SQL μπορούν να διακριθούν σε τρεις κύριες κατηγορίες:
  - εντολές ορισμού δεδομένων (Data Definition Language - DDL) (π.χ. CREATE TABLE)
  - εντολές χειρισμού δεδομένων (Data Manipulation Language - DML) (π.χ. UPDATE, INSERT, SELECT)
  - εντολές ελέγχου δεδομένων (Data Control Language - DCL) (π.χ. GRANT, REVOKE)
- Συνήθως, οι δεσμευμένες λέξεις της SQL γράφονται με κεφαλαία γράμματα
- Οι εντολές πρέπει να τερματίζονται με ;

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οι εντολές ορισμού δεδομένων αφορούν στη:
  - Δημιουργία βάσης δεδομένων
  - Διαγραφή βάσης δεδομένων και όλων των πινάκων σε αυτή
  - Δημιουργία πίνακα
  - Διαγραφή πίνακα
  - Τροποποίηση της δομής ενός πίνακα

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Εντολή δημιουργίας βάσης δεδομένων

**CREATE DATABASE IF NOT EXISTS mydb;**

- Η εντολή δημιουργεί μία βάση δεδομένων με το αναγνωριστικό mydb, εφόσον δεν υπάρχει ήδη μία βάση δεδομένων με το αναγνωριστικό αυτό.
- Εάν υπάρχει ήδη μία βάση δεδομένων με το αναγνωριστικό αυτό, η εντολή δεν επιφέρει κάποια τροποποίηση

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οι λέξεις **IF NOT EXISTS** μπορούν να παραλειφθούν.
- Στην περίπτωση αυτή, εάν υπάρχει ήδη μία βάση δεδομένων με το συγκεκριμένο αναγνωριστικό, θα παραχθεί το μήνυμα λάθους:

**Error Code: 1007. Can't create database 'mydb'; database exists**

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Εντολή διαγραφής βάσης δεδομένων

**DROP DATABASE IF EXISTS mydb;**

- Η εντολή διαγράφει τη βάση δεδομένων με το αναγνωριστικό mydb, εφόσον υπάρχει ήδη.
- Εάν δεν υπάρχει η βάση δεδομένων με το αναγνωριστικό αυτό, η εντολή δεν επιφέρει κάποια τροποποίηση

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οι λέξεις **IF EXISTS** μπορούν να παραλειφθούν.
- Στην περίπτωση αυτή, εάν δεν υπάρχει η βάση δεδομένων με το συγκεκριμένο αναγνωριστικό, θα παραχθεί το μήνυμα λάθους:

**Error Code: 1008. Can't drop database 'mydb'; database doesn't exist**

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Για να επιλεγεί η βάση δεδομένων με το αναγνωριστικό `mydb` προς χρήση, πρέπει να εκτελεστεί η εντολή:

**USE mydb;**

- Εάν δεν υπάρχει η βάση δεδομένων με το αναγνωριστικό `mydb`, θα παραχθεί το μήνυμα λάθους:

**Error Code: 1049. Unknown database 'mydb'**

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Γενική μορφή εντολή δημιουργίας πίνακα

```
CREATE TABLE IF NOT EXISTS myTable(  
column_name_1 data_type [column_attributes]  
[, column_name_2 data_type [column_attributes]]...  
[, table_level_constraints]  
);
```

- Η εντολή δημιουργεί έναν πίνακα με το αναγνωριστικό myTable, εφόσον δεν υπάρχει ήδη ένας πίνακας με το αναγνωριστικό αυτό.
- Εάν υπάρχει ήδη ένας πίνακας με το αναγνωριστικό αυτό, η εντολή δεν επιφέρει κάποια τροποποίηση

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οι λέξεις **IF NOT EXISTS** μπορούν να παραλειφθούν.
- Στην περίπτωση αυτή, εάν υπάρχει ήδη ο πίνακας με το συγκεκριμένο αναγνωριστικό, θα παραχθεί το μήνυμα λάθους:

**Error Code: 1050. Table 'mytable' already exists**

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Η εντολή περιλαμβάνει τη δημιουργία μίας ή περισσότερες στήλες (πεδία)
- Για κάθε στήλη πρέπει να δοθεί ο τύπος δεδομένων
- Προαιρετικά, σε κάθε στήλη μπορούν να αποδοθούν μία ή περισσότερες ιδιότητες

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Συνήθεις ιδιότητες μιας στήλης

Ιδιότητα	Περιγραφή
<b>NOT NULL</b>	Η στήλη πρέπει υποχρεωτικά να έχει τιμή
<b>NULL</b>	Η στήλη δεν είναι υποχρεωτικό να έχει τιμή
<b>UNIQUE</b>	Η τιμή που αποθηκεύεται σε μία στήλη πρέπει να είναι μοναδική
<b>PRIMARY KEY</b>	Η στήλη είναι πρωτεύον κλειδί
<b>DEFAULT default_value</b>	Η στήλη έχει προκαθορισμένη τιμή
<b>AUTO_INCREMENT</b>	Η τιμές της στήλης αυξάνεται αυτόματα όταν προστίθεται νέα γραμμή. Η ιδιότητα αυτή μπορεί να εφαρμοστεί όταν ο τύπος της στήλης είναι ακέραιος αριθμός κινητής υποδιαστολής

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Παρόλο που δεν υποχρεωτικό, καλό είναι κάθε πίνακας να έχει ένα πρωτεύον κλειδί
- Ο πιο απλός τρόπος είναι να δημιουργηθεί μια στήλη, η οποία να παίρνει ακέραιες τιμές και αυξάνει αυτόματα κατά ένα με κάθε νέα εισαγωγή γραμμής

```
id INT AUTO_INCREMENT PRIMARY KEY
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Συνεπώς, ο τρόπος δημιουργίας ενός πίνακα προτείνεται να γίνεται ως εξής:

```
# ΠΙΝΑΚΑΣ ΜΕ ΑΣΘΕΝΕΙΣ  
CREATE TABLE myTable(  
id INT AUTO_INCREMENT PRIMARY KEY, # ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ  
/*  
ΑΚΟΛΟΥΘΟΥΝ ΤΑ ΥΠΟΛΟΙΠΑ ΠΕΔΙΑ ΕΔΩ  
*/  
) ENGINE=INNODB DEFAULT CHARSET = UTF8;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στο προηγούμενο παράδειγμα:
  - Η επιλογή **ENGINE=INNODB** αφορά στην εσωτερική οργάνωση και αποθήκευση των δεδομένων. Εξασφαλίζει τη δημιουργία σχεσιακών περιορισμών μεταξύ πινάκων
  - Η επιλογή **DEFAULT CHARSET=UTF8** χρησιμοποιείται για να υπάρχει η δυνατότητα εισαγωγής δεδομένων με ελληνικούς χαρακτήρες

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### Παράδειγμα

Έστω ότι πρόκειται να δημιουργηθεί μια ιστοσελίδα για συζητήσεις και ανταλλαγή μηνυμάτων. Για το σκοπό αυτό πρόκειται να δημιουργηθεί ένας πίνακας χρηστών με τα ακόλουθα πεδία:

- Κωδικός (αύξων αριθμός με αρχή το 1)
- Όνομα
- Επώνυμο
- Email
- Όνομα χρήστη (μέγιστο πλήθος χαρακτήρων 15)
- Συνθηματικό (αποθηκεύεται κρυπτογραφημένο με σταθερό μήκος 40 χαρακτήρων)
- Ημερομηνία και ώρα εγγραφής

Να γραφτεί η εντολή SQL που δημιουργεί τον πίνακα.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ

Επειδή πρόκειται για πίνακα που κρατάει χρηστών, υπάρχουν οι ακόλουθοι περιορισμοί:

- Όλα τα πεδία πρέπει να έχουν τιμή
- Τα πεδία όνομα χρήστη και email πρέπει να έχουν μοναδικές τιμές, ώστε να αποφευχθούν διπλές εγγραφές
- Η αναζήτηση με βάση το όνομα χρήστη και το συνθηματικό πρέπει να γίνεται αποδοτικά

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### Λύση

```
# Εντολή δημιουργίας πίνακα χρηστών
CREATE TABLE users (
    userId MEDIUMINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
    firstName VARCHAR(25) NOT NULL,
    lastName VARCHAR(40) NOT NULL,
    email VARCHAR(60) NOT NULL UNIQUE,
    username VARCHAR(15) NOT NULL UNIQUE,
    pass CHAR(40) NOT NULL, # Η λέξη password είναι
    δεσμευμένη λέξη της SQL
    registrationDate DATE NOT NULL,
    INDEX(username, pass)
)ENGINE=InnoDB DEFAULT CHARSET = UTF8;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στο προηγούμενο παράδειγμα, προστέθηκε ο περιορισμός **INDEX(username, pass)** για να δημιουργηθεί ένας **δείκτης** με τις στήλες username και pass
- Η δημιουργία του δείκτη διευκολύνει και καθιστά πιο γρήγορα την αναζήτηση εγγραφών με κριτήρια το όνομα χρήστη και το συνθηματικό
- Αντίστοιχα, θα μπορούσε να δημιουργηθεί ο δείκτης **INDEX(lastName)** ώστε να γίνονται πιο αποδοτικές αναζητήσεις με κριτήριο το επώνυμο του χρήστη, εάν αυτό είναι απαραίτητο

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Ιδιαίτερη προσοχή χρειάζεται όταν πρόκειται να δημιουργηθούν πίνακες που συνδέονται μεταξύ τους
- Πρέπει να δημιουργηθούν πρώτα οι πίνακες που δεν έχουν καμία εξάρτηση, δηλ. δεν έχουν εξωτερικό κλειδί
- Στη συνέχεια, δημιουργούνται οι πίνακες “παιδιά”

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΠΑΡΑΔΕΙΓΜΑ

Συνεχίζοντας το προηγούμενο παράδειγμα, η βάση δεδομένων θα περιέχει επιπλέον:

- έναν πίνακα με τις θεματικές περιοχές συζητήσεων (forums) (κωδικός, τίτλος θεματικής περιοχής)
- έναν πίνακα για τα μηνύματα (messages) (κωδικός, κωδικός χρήστη, κωδικός θεματικής περιοχής, κωδικός αρχικού μηνύματος, θέμα μηνύματος, σώμα μηνύματος, ημερομηνία ανάρτησης μηνύματος)

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ

Σε ένα forum θα υπάρχουν πολλά μηνύματα, οπότε υπάρχει μία σχέση ένα προς πολλά ανάμεσα στους δύο πίνακες. Το πεδίο κωδικός θεματικής περιοχής θα είναι το εξωτερικό κλειδί για τη σχέση αυτή.

Αντίστοιχα, ένας χρήστης θα αναρτά πολλά μηνύματα, οπότε υπάρχει μία σχέση ένα προς πολλά ανάμεσα στους δύο πίνακες. Το πεδίο κωδικός χρήστη θα είναι το εξωτερικό κλειδί για τη σχέση αυτή.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ

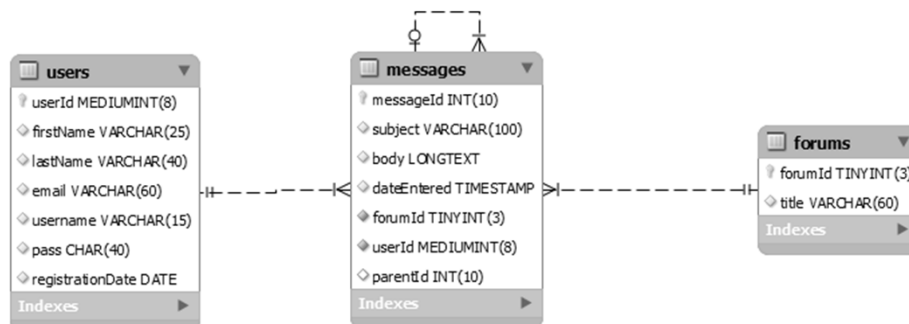
Ένα μήνυμα μπορεί να είναι απάντηση σε κάποιο προηγούμενο μήνυμα. Συνεπώς, ένα μήνυμα μπορεί να έχει πολλές απαντήσεις, δηλαδή ανάμεσα στον πίνακα μηνύματα και στον εαυτό του υπάρχει μία σχέση ένα προς πολλά.

Το πεδίο κωδικός αρχικού μηνύματος θα είναι το εξωτερικό κλειδί για τη σχέση αυτή

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ



## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ

```
# ΠΙΝΑΚΑΣ ΜΕ ΘΕΜΑΤΙΚΕΣ ΠΕΡΙΟΧΕΣ
CREATE TABLE forums (
forumId TINYINT UNSIGNED AUTO_INCREMENT PRIMARY KEY, # ΠΡΩΤΕΥΟΝ
ΚΛΕΙΔΙ
title VARCHAR(60) NOT NULL
)ENGINE=INNODB DEFAULT CHARSET=UTF8;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ

```
# ΠΙΝΑΚΑΣ ΜΕ ΜΗΝΥΜΑΤΑ
CREATE TABLE messages (
messageId INT UNSIGNED AUTO_INCREMENT PRIMARY KEY, # ΠΡΩΤΕΥΟΝ ΚΛΕΙΔΙ
subject VARCHAR(100) NOT NULL,
body LONGTEXT NOT NULL,
dateEntered TIMESTAMP,
forumId TINYINT UNSIGNED NOT NULL,
userId MEDIUMINT UNSIGNED NOT NULL,
parentId INT UNSIGNED,
FOREIGN KEY(forumId) REFERENCES forums(forumId),
FOREIGN KEY(userId) REFERENCES users(userId),
FOREIGN KEY(parentId) REFERENCES messages(messageId)
)ENGINE=INNODB DEFAULT CHARSET=UTF8;
```

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Στο προηγούμενο παράδειγμα οι γραμμές  
`FOREIGN KEY(forumId) REFERENCES forums(forumId),`  
`FOREIGN KEY(userId) REFERENCES users(userId),`  
`FOREIGN KEY(parentId) REFERENCES messages(messageId)`  
Δημιουργούν τις σχέσεις του πίνακα `messages` με τους αντίστοιχους πίνακες.

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΠΑΡΑΔΕΙΓΜΑ

Η βάση δεδομένων του ΤΕΙ Αθήνας περιέχει μεταξύ άλλων τους ακόλουθους πίνακες:

- Τμήματα (ονομασία)
- Καθηγητές (επώνυμο, όνομα, βαθμίδα)
- Φοιτητές (αριθμός μητρώου, επώνυμο, όνομα, ημερομηνία εγγραφής)
- Μαθήματα (ονομασία, κωδικός καθηγητή, κωδικός τμήματος)
- Εγγραφές (κωδικός φοιτητή, κωδικός μαθήματος, ημερομηνία εγγραφής, βαθμός)

Να γίνει το διάγραμμα που δείχνει τους πίνακες και τις σχέσεις μεταξύ τους

Να γραφτούν οι εντολές SQL για τη δημιουργία των πινάκων

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ

- *Παρόλο που δεν αναφέρεται, κάθε πίνακας θα έχει επιπρόσθετα μία στήλη με ακέραιες τιμές που θα αυξάνουν αυτόματα, η οποία θα είναι το πρωτεύον κλειδί του πίνακα*
- Υπάρχουν οι ακόλουθες σχέσεις μεταξύ των πινάκων:
  - Ένα τμήμα έχει πολλούς καθηγητές
  - Ένα τμήμα έχει πολλούς φοιτητές
  - Ένα καθηγητής διδάσκει πολλά μαθήματα
  - Ένας φοιτητής κάνει πολλές εγγραφές σε μαθήματα
  - Σε ένα μάθημα γίνονται πολλές εγγραφές

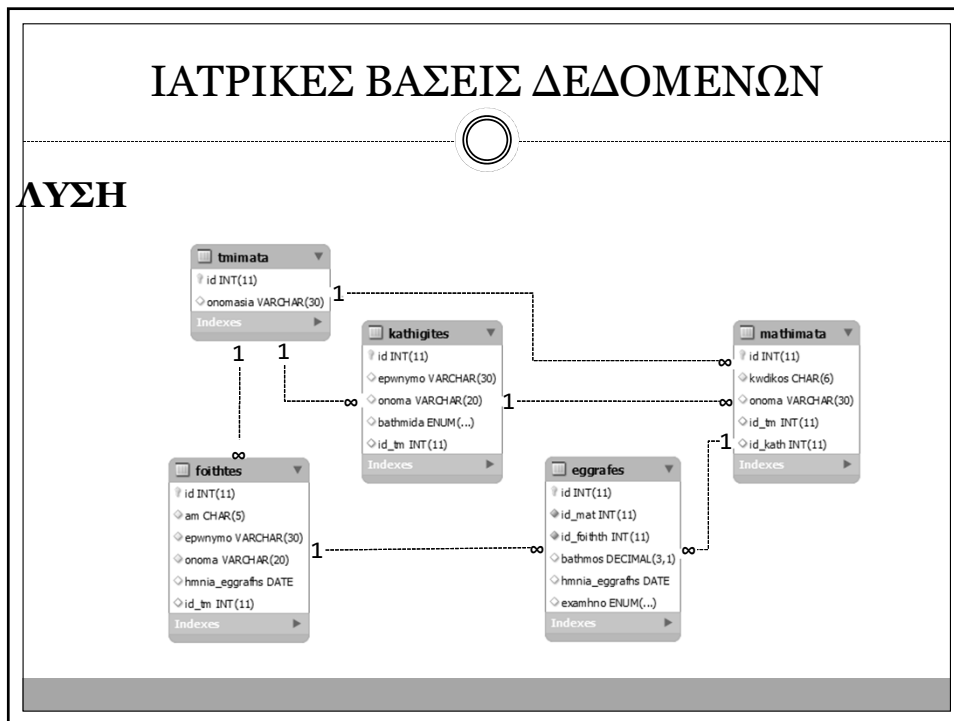
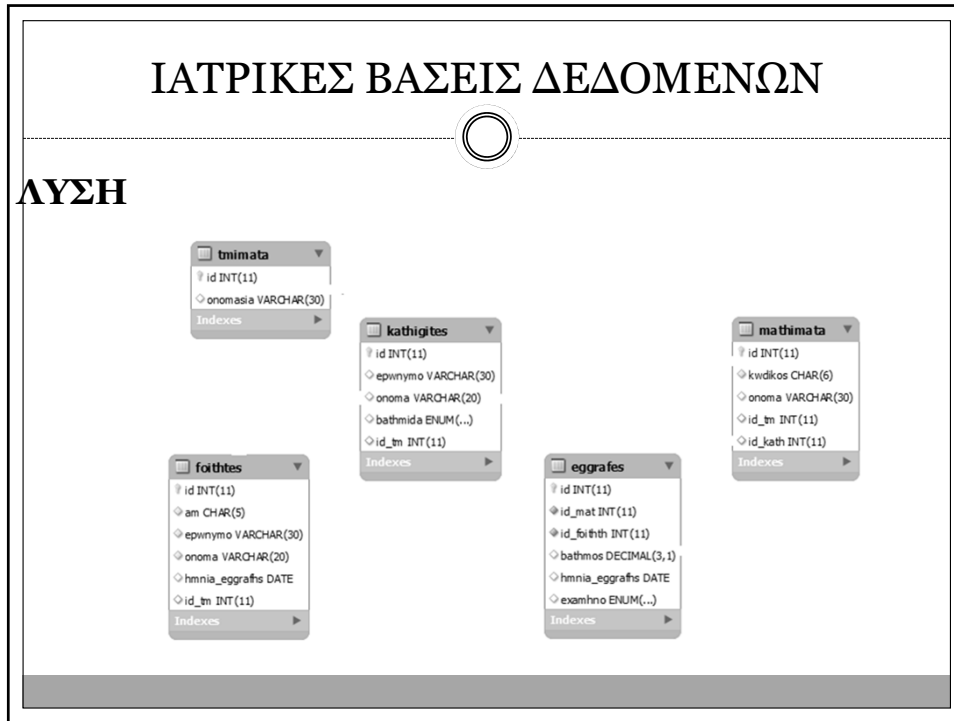
## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΛΥΣΗ

Για να δημιουργηθούν οι σχέσεις μεταξύ των πινάκων, χρειάζεται **εξωτερικό κλειδί** στον:

- πίνακα με τους καθηγητές για τη σχέση με τον πίνακα με τα τμήματα
- πίνακα με τους φοιτητές για τη σχέση με τον πίνακα με τα τμήματα
- πίνακα με τα μαθήματα για τη σχέση με τον πίνακα με τους καθηγητές
- πίνακα με τις εγγραφές για τη σχέση με τον πίνακα με τους φοιτητές
- πίνακα με τις εγγραφές για τη σχέση με τον πίνακα με τα μαθήματα



## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



### ΑΣΚΗΣΗ 8

Για το προηγούμενο παράδειγμα να γραφτούν οι εντολές SQL που δημιουργούν τους πίνακες

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Εντολή διαγραφής πίνακα

**DROP TABLE IF EXISTS myTable;**

- Η εντολή διαγράφει τον πίνακα με το αναγνωριστικό myTable, εφόσον υπάρχει ήδη.
- Εάν δεν υπάρχει ο πίνακας με το αναγνωριστικό αυτό, η εντολή δεν επιφέρει κάποια τροποποίηση

## ΙΑΤΡΙΚΕΣ ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ



- Οι λέξεις **IF EXISTS** μπορούν να παραλειφθούν.
- Στην περίπτωση αυτή, εάν δεν υπάρχει ο πίνακας με το συγκεκριμένο αναγνωριστικό, θα παραχθεί το μήνυμα λάθους:

**Error Code: 1051. Unknown table 'mydb.mytable'**