DataBase Laboratory, 2nd year - Introduction

I. Database fundamentals

What is a database ?

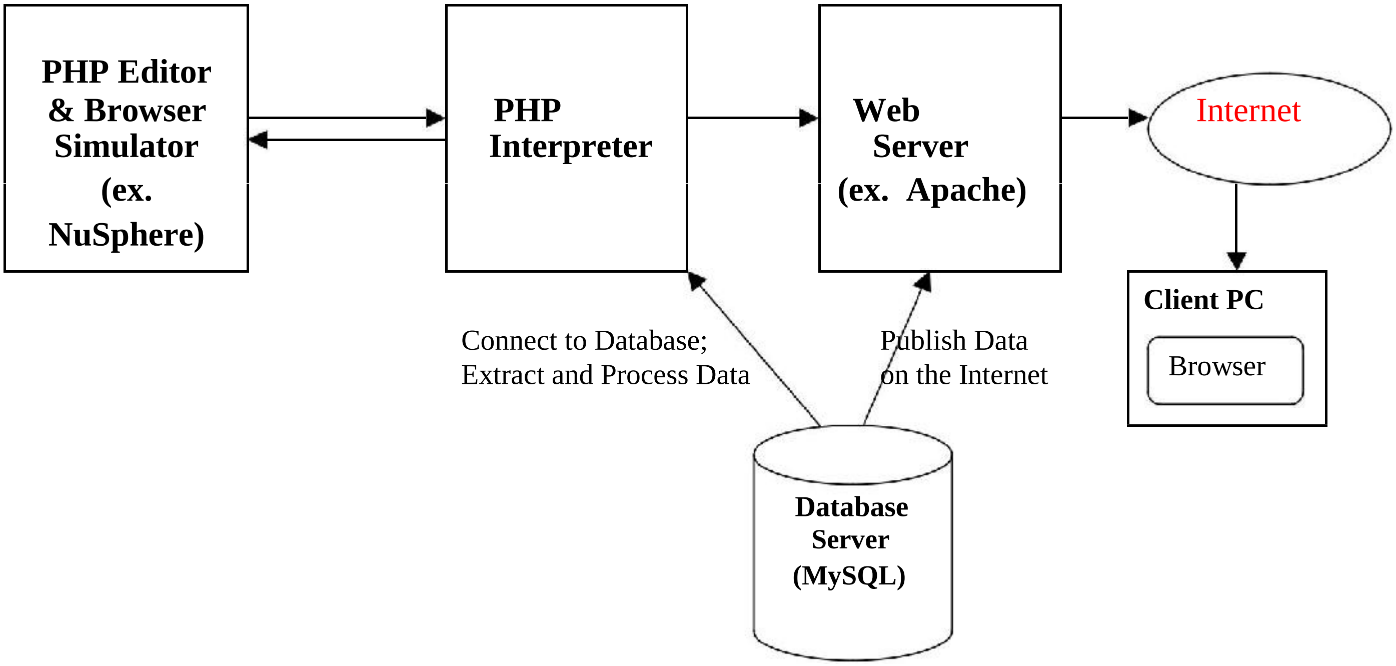
The structure of a database

How to build a database?

- Create tables|
- Define relations between the tables|
- Define constraints and referential integrity rules|
- Create views within the database|

How to extract information from a database? (queries)

II. Creating dynamic web pages that comunicate with a database


PHP Editor & Browser Simulator (ex. NuSphere)
PHP Interpreter
Web Server (ex. Apache)
Internet
Client PC
Browser
Connect to Database; Extract and Process Data
Publish Data on the Internet
Database Server (MySQL)

Environments and languages:

SQL Server 2008 – Database Server and DBMS – Database description an d data manipulation. Transact SQL Language;

MySQL – SQL RDBMS, very efficient in developing web based software applications

HTML – Hypertext Markup Language – the language for web page design and description

PHP – Hypertext Preprocessor; server-side HTML embedded Scripting Language; connect to Database and Process Web Data

Required tools:

SQL Server 2008 Express Edition + SQL Server 2008 Management Studio Express

Download: http://www.microsoft.com/express/database/

MoWes Portable(Apache+MySQL+PHP); MySQL Workbench interface

http://www.softpedia.com/get/Internet/Servers/Server-Tools/MoWeS-Portable.shtml

OR

Wamp (Apache + MySQL + PHP). "TOAD" interface

Download: http://www.wampserver.com/en/download.php

PHP editor (Eclipse, NuSphere, PhpEdit)

Tutorials:

SQL interactive tutorial: http://www.sqlzoo.net/

SQL Server 2008 documentation

- SQL Server 2008 Books Online (Microsoft SQL Server 2008 package, on the computer)

http://msdn2.microsoft.com/en-us/sql/

http://blogs.msdn.com/b/sqlserverue/archive/2008/08/06/sql-server-2008-books-online-now-available.aspx

- R. Dollinger, Utilizarea Sistemului SQL Server (SQL 7.0, SQL 2000)

MySQL http://www.tutorialspoint.com/mysql/mysql-select-query.htm

PHP Tutorial – PHP Documentation integrated in NuSphere; or, http://www.php.net/manual/en/

HTML Tutorial- HTML Documentation Integrated in NuSphere, or: http://www.2createawebsite.com/build/html.html - download htmltutorial.exe

Schedule:

1. Learn **SQL language (http://www.sqlzoo.net) - 3 weeks**
2. Learn **SQL Server 2008 - 3 weeks**
3. Learn **HTML, PHP and MySQL - 3 weeks**
4. Work with Wamp for building a small web-site (individual work) 3 labs **- 3 weeks**

5. **Assessment:** *Laboratory activity* and *individual work (the small web-site)*. – **2 weeks**

Laboratory no. 1

1.) Read about databases in http://en.wikipedia.org/wiki/Database

2.) The basic structure of the SQL phrase
http://msdn.microsoft.com/en-us/library/ms189499%28v=sql.100%29.aspx

3.) http://www.sqlzoo.net

Sections: 1.) SELECT basics; 2.) SELECT from World; 3.) SELECT from Nobel

⎠ **Use the following options: SQL Server; MySQL|**

Laboratory no. 2

4.) http://www.sqlzoo.net

Sections:

4. Select within Select
5.SUM and COUNT
6.JOIN

Read all the indications, study the model queries and find the appropriate solutions for the unsolved queries. Solve the additional three queries:

a. Using the *bbc* table, show the region that contains the country which has the minimum population from all the existing countries.
b. Using the *bbc* table, for each region, show the country that has the maximum population
c. Using the following two tables, *ttmd (games, color, team, country)* and *team (id, name)* show the games to which the country called 'CHN' p articipated. List also the names of the team members. Order the result by the names of the team members.

1.) Additional reading: a.
SQL phrase structure:
http://msdn.microsoft.com/en-us/library/ms189499(SQL.90).aspx
b. aggregate functions and Group By Clause:
SUM: http://msdn.microsoft.com/en-us/library/ms187810(SQL.90).aspx
COUNT: http://msdn.microsoft.com/en-us/library/ms175997(SQL.90).aspx
AVG: http://msdn.microsoft.com/en-us/library/ms177677(SQL.90).aspx
MAX: http://msdn.microsoft.com/en-us/library/ms187751(SQL.90).aspx
MIN: http://msdn.microsoft.com/en-us/library/ms179916(SQL.90).aspx
GROUP BY: http://msdn.microsoft.com/en-us/library/ms177673(SQL.90).aspx
ORDER BY:
c.the JOIN operator in SQL :
http://msdn.microsoft.com/en-us/library/ms191517.aspx

Laboratory no. 3

2.) http://www.sqlzoo.net

Sections:

7. More JOIN operations
8. Using NULL
9. Self Join
10. Tutorial quizzes (optional)

Read all the indications, study the model queries and find the appropriate solutions for the unsolved queries.

Additional reading

the JOIN operator in SQL :
http://msdn.microsoft.com/en-us/library/ms191517.aspx
using JOINS (Inner Join, Outer Join, Cross Join, Self Join, joining three or more tables): http://msdn.microsoft.com/en-us/library/ms191472.aspx

NULL values and Joins:
http://msdn.microsoft.com/en-us/library/ms190409.aspx
Create and Drop, Insert and Delete, Date and Time, Functions,
http://www.sqlzoo.net

Laboratory no. 4

1.) In Microsoft SQL Server 2008 create the “Exams” dat abase having the following structure (*tables in the database*):

> **Students** – fields: Stud_Id (int), Name (varchar), Surname (varchar), Gender(char), Date_of_Birth (datetime), Group_Id(int),
> **Groups** – fields: Gr_Id (int), Gr_name (varchar), Year_of_S tudy (int), Comments (varchar)
> **Teachers** – fields: Teacher_Id (int), Name (varchar), Surname (varchar), Date_of_Birth (datetime), degree (varchar)
> **Disciplines** – fields: Discipline_Id (int), Disc_name (varchar) , Discipline_Type (varchar – hardware or software)
> **Marks** – fields: Student_Id (int), Teacher_Id (int), Disc_ Id (int), Mark (int), Date_of_Exam (datetime)

Choose the **primary keys** of the tables (Stud_Id, Gr_Id, Teacher_Id, Discipline_Id). Modify (alter) the structure(design) of the Students table by adding a new field, “Has_scholarship”, that can take the values “yes” or “no”. Build the SQL Server Diagram of this database. Establish **one-to-many and many-to-many** relationships between the tables.

2.) Using the SQL Server 2008 Management Studio Express, impose restrictions on the created database:

a. **Make the “Name” field in the Disciplines table u nique** (manually or using the Alter Table or Create Unique Index statement)

b. **Check Constraints:**

- using the Check clause within the Alter Table statement, make sure that, from now on, the group number from the Student Table will be situated between 1 and 5.|
- also, make sure that the Mark field in the Marks table is always a positive number, >0.|

http://msdn.microsoft.com/en-us/library/ms190273(SQL.90).aspx

Referential Integrity Restrictions:

c. **Every foreign key value from the child table must be contained also in the parent table** (tables Students – Groups, Marks – Students, Marks – Teachers, Marks – Disciplines)

d. **Referential Integrity (RI) Constraints:** set cascade on delete, restrict on insert and update - manually as well as using Transact-SQL code within the Alter Table statement http://msdn.microsoft.com/en-us/library/ms186973(SQL.90).aspx

3.) Read about Transact SQL Data Types:

In SQL Server 2008 Books Online folllow the links: Database Engine -> Database Engine Developer Info Center ->Languages - Transact SQL Reference -> DataTypes (Transact SQL) – left window http://msdn.microsoft.com/en-us/library/ms187752(SQL.90).aspx

4.) Use the appropriate Transact SQL statements in order to create or drop tables, insert, delete and update data from the database. Write the appropriate Transact SQL-statements in order to drop the table Groups, then to create it again, then to alter it in order to add a new column – Group_description, varc har(100). Create a new table Teachers/Disciplines in order to make direct correspondence between the teachers and the disciplines they teach. Insert data in all the tables from the database.

SQL Server 2008 Books Online : Create Table, Drop Table, Alter Table, Insert, Delete, Update http://msdn.microsoft.com/en-us/library/ms174979(SQL.90).aspx
http://msdn.microsoft.com/en-us/library/ms174335(SQL.90).aspx

Homework:

Solve the following SQL queries:

1. Select the names and surnames of the students from the 3-rd group.
2. Select the name, surname, date of birth and group for those students whose name begin with the ‘A’ letter

Laboratory no. 5

Solve the following SQL Queries in SQL Server 2008 Management Studio Express, using the Exams database and the SELECT statement:

(Inner) Join operator

a. Name, surname and group for students who sustained the exam with the teacher named 'Popovici'.

Select In Select

b. Find those students whose marks at Databases are greater than Popescu Ion's Mark at the same discipline.

Order By, Top

c. Find the first three students (name, surname and group) with the highest marks at Computer Programming.

Group By, Aggregate Functions

d. Name, surname and average mark (for all the exams sustained) for each student from group 3021 (use Group By, AVG function).

e. Name, description and number of students for each group (Group By clause, COUNT function).

Outer Join, Aggregate functions:

f. Students' name, surname and number of exams sustained. For students that didn't sustain any exam, the *0* value should appear.

In/Not In Operator; **Select In Select**

g. Find the teachers that teach both Databases and Computer Programming (***clue*** - those teachers that appear both in the following result sets: teachers that teach Databases and teachers that teach Computer Programming)

h. Name, surname and group for students who didn't sustain the exam at Databases.(***clue:*** those students do not appear in the Exams table; the Student_id from the Students table doesn't appear in the Exams table)

Homework:

Solve the same query in two ways:

- Using Exists/ Not exists function
- Using the Count aggregate function and the Group By clause

1.) Find the students who didn't sustain the exam with the teacher named 'Ionescu Vasile'

2.) Find the pairs student/teacher that didn't sustain any exam together.

Laboratory no. 6

2. **a. Read in SQL Server 2008 Books online about views. http://msdn.microsoft.com/en-us/library/ms190706(SQL.90).aspx**

   b. Create the following views in your database (*Create View* Transact-SQL statement): http://msdn.microsoft.com/en-us/library/ms187956(SQL.90).aspx

   - Student’s name, surname and group, discipline name, date of exam, mark obtained at that exam.
   - Teacher’s name and surname, discipline name that is taught by the teacher. For the teachers that don’t teach any discipline, the ‘no discipline’ string will be displayed.
   - Names and surnames for the teachers that didn’t sustain any exam with the students from the group 3021.
   - Names and surnames for the students that sustained the exam at Databases, but didn’t sustain the exam at Computer Programming.
   - Names and surnames for teachers that teach Mathematics (the discipline name must contain the word ‘Mathematics’).

   c. Modify (alter) the first view in order to display only the marks obtained by the students at the exam at databases. http://msdn.microsoft.com/en-us/library/ms178076(SQL.90).aspx

   d. Update the last created view, as a mistake has been made in the corresponding data: the teacher called Popescu Maria, who teaches mathematics, is actually named Pop Maria. http://msdn.microsoft.com/en-us/library/ms180800(SQL.90).aspx

3. **a. Read in SQL Server 2008 Books online about Transact SQL statements (BEGIN…END, IF…ELSE, CASE, WHILE, WAITFOR, GOTO, GO) and about batch files. http://msdn.microsoft.com/en-us/library/ms189826(SQL.90).aspx http://msdn.microsoft.com/en-us/library/ms175502(SQL.90).aspx**

   b. Create a batch file in order to display the names and surnames of the students that sustain the exam at ‘Computer Programming’. If nobody sustained this exam, a corresponding message should be displayed. Use the IF…ELSE Transact SQL Statement.

Laboratory no. 7

HTML, PHP Basics

1. Read in HTML tutorial about the HTML page structure and about the basic HTML tags.
2. **HTML images:** Build a HTML page with the title “Technical Univers ity of Cluj-Napoca” and insert the Technical University’s symbol, using the “sigla.gif” file in the current folder. Use the **<img src>** HTML tag. Display the image in the upper side of the page, bellow the title, in the center.
3. **HTML Hyperlinks:** Insert, in the same page, bellow the image, a Hyperlink to the Computer Science Department of the Technical University of Cluj site; URL: **http://www.cs.utcluj.ro**.Use the **<a href>** HTML tag. Change the hyperlink colors: active link should be red, visited link should be green and the usual color of the link should be orange.
4. **HTML forms and parameter transmission; Insert, in the HTML page, forms in order to input data about students:**
    a. Insert text-boxes for student’s names, student’s surnames, student’s place of birth and student’s address.
    b. Use select lists – one for student’s gender (Male o r Female) and another group for student’s date of birth (day, month, year).
    c. Use radio-buttons, in order to specify whether the student is in years 1-3, or in years 4-5.
    d. Use a check box, in order to specify whether the student takes scholarship or not.
    e. Use a “Submit” button in order to submit data in th e form and a “Clear” button in order to clear the form at user’s request.

Parameters transmission: use the clauses ACTION= “myfile.php” and METHOD=post inside of the <form> tag, just like in the example from the tutorial. See PHP Manual – Dealing with Forms.

5. Arrange the data from the HTML page, using an HTML table. Put each interface element (e.g. text-box) and its label in a distinct row. Put the label in the left column and the interface element in the right column. Make the borders of your table invisible. Make the first column of your table of width 30%. Vertically align the text in your table in center and horizontally – on the left side.

6. Create a .php file named “myfile.php”. Dysplay t he world “Hello!” using the php statement “ echo. Format the world using the HT ML tags: make it ***bold***. Put the html and php files in a folder called “work ”. Move this folder in
    “C:\Program Files\Apache Group\Apache2\htd ocs\”

Run the .php file in the Internet Explorer Browser, using the following address:

http://your_computer_name/work/myfile.php

7. **Data processing and display with PHP:**
   Getting the parameters: In “myfile.php”, use the $_POST[name_from_input_tag] in order to get the data from the previous HTML page. Assign the values to PHP variables. Display the data on the screen in the following format:

   Name: Popescu
   Surname: Ion
   Gender: Male
   Date of Birth: Day, Month, Year
   Years of study: 1-3, or 4-5
   Takes scholarship (or Doesn’t take scholarship). **Age:** calculate age in years, based on the Date of Birth

Help:
HTML Tutorial: http://www.2createawebsite.com/build/html.html
PHP Tutorial: http://www.php.net/manual/en/

Laboratory no. 8

HTML, PHP Basics; Connections to Databases

1. Read in Php Manual about Types and Operators. Use the string concatenation operation in order to obtain a new string from student's name and surname, separated by a space character. Write on the screen the following phrase: The student name is …. (specify student's name and surname using the string built previously).

2. Read in "Php Manual" about Control Structures.
   In the ".php" page, use an array in order to specif y the main subjects that are being studied by the students. Display these subjects on the screen using a bulleted list and a cycle (for, while, or do…while). Then, use an ordered (numbered) list in order to display three rules that you consider very important for students.

3. Rebuild the Exams database in the MySQL. Use the MySql Workbench environment of Php MySQL Portable.

4. Modify the "test1_mysql.php" file from the labor atory folder in order to connect to your own database "Exams", to select dat a from the Students table and to display on the screen the students' names, surnames and group name, within an html table. Use the mysql_connect, mysql_select_db and mysql_query statements. Read in the PHP manual about connections to databases: ordinary and persistent connections. Put the file in the C:\wamp\www\exams\ folder in order to work properly.

5. In SQL Server, Exams database, change the Students table in order to contain the following fields: Name (varchar), Surname (varchar), Gender (Char), Date_of_Birth (Datetime), Year_of_study (varchar), Scholarship (char –'y' or 'n'). Use the form designed in the .h tml page that you've created during the former laboratory (Laboratory no. 7), and the mechanism of parameter transmission, in order to Insert data from the .html file into the Students table of the Exams database, using the SQL statement:

   Insert into table_name(field1, field2,…, fiel dn) values (val1, val2,…, valn)

6. Use a first page, "select.php", containing two sele ct lists, in order to choose the desired student by name and surname. Transmit the data about the selected student in a second page, "update_stud ent.php". There, build a form in order to modify the data about the student, then update the corresponding data into the database, using the MySQL Update statement.

Help: PHP, HTML documentation:
http://www.2createawebsite.com/build/html.html
http://www.php.net/manual/en/
SQL Server 2008 Books Online: http://msdn.microsoft.com/en-us/library/ms189499(SQL.90).aspx
http://msdn.microsoft.com/en-us/library/ms174335(SQL.90).aspx
http://msdn.microsoft.com/en-us/library/ms177523(SQL.90).aspx

Laboratory no. 9

HTML, PHP Web Site

1. Read in PHP Documentation (manual) about Database Connection handling and about Persistent Database Connections. Also read about the require( ) and include( ) statements. Build an extra-file, called “init.php” and write code for database connection here. In the main “.php” file u se the “require” statement in order to include and execute “init.php”.

2. Read the Web Site definition from: http://www.webopedia.com/TERM/W/web_site.htm

3. Build a Web Site corresponding to your own domain, containing multiple .html and .php pages.You will have “index.html”as the first page, consisting in an authentication form. If the login is correct, the user must access the second page, “main.php”. The page “main.php” consis ts in a frameset (describe the frameset, using the <frameset> html tag). The frameset will contain a left frame with a menu, a middle frame in order to display the information required in the left frame and a right frame displaying the author’s (your name).

The menu in the left frame will have to enable the following operations with the database: Insertion, Update, Deletion and Selection (visualization) of data for each entity (e.g.: Students, Disciplines, Teachers, Exams).

All these operations will be done through .php pages displayed in the middle frame. The menu in the left frame will contain the links to these pages; in order to display these pages in the middle frame, you will use the “target” clause (e.g. target=”middleFrame”).

In order to build the frameset properly, use the following documentation: **http://www.htmlcodetutorial.com/frames/FRAMESET.html**

The interface should look like in the “mo del.jpg” file.

Laboratory no. 10 – 14
Individual assignment

1. Build your own website
 a. Build the database in MySQL
 b. Build the authentication page
 c. Build the other web-pages using html, php and eventually other script languages for web page construction.

2. Build your own stylesheet (".css" file) and integra te it in the ".html" page using the link tag, inside of the header. Ex.:

```
<head>
<link rel="stylesheet" type="text/css"
  href="mystyle.css" />
</head>
```

Documentation:
http://www.w3schools.com/css/default.asp
Use different styles on different portions of text – e.g. – for the title use the font Verdana, color: red, size: 3 and for the other text use the font Arial, size.
Use different styles on different groups of links. Define colors for active link, for visited link, for normal link, and use a distinct color when the mouse cursor passes over the link ("hover").
Models: "_style.css"

MySQL Tutorial
http://dev.mysql.com/doc/refman/5.1/en/
Toad
http://www.toadsoft.com/toadmysql/mysql_freeware.html