Data Bases Lab

SQL (part 1)

Create, Insert, Delete, Update, Alter

SQL

- SQL (Structured Query Language) is a standard language for storing, manipulating and retrieving data in databases.
- The scope of SQL includes data query, data manipulation (insert, update, and delete), data definition (schema creation and modification), and data access control.
- A database comprises of Tables which have names, attributes (columns) and records (lines with data)

How to (with sql statements)

- **Create database:**

```
CREATE DATABASE dbname;
```

- **Choose database to use :**

```
USE dbname;
```

- **Delete database (do not do it):**

```
DROP DATABASE dbname;
```

- **Show all the tables in the database that you currently use:**

```
SHOW TABLES;
```

- **Show the attributes of a table:**

```
DESCRIBE table_name;
```

- **Show the instruction create table:**

```
SHOW CREATE TABLE table_name;
```

- **Show the contents of a table**

```
SELECT * FROM table_name;
```

Examples 1

```
mysql> CREATE DATABASE dblab;
Query OK, 1 row affected (0.00 sec)

mysql> USE dblab;
Database changed

mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql> DROP DATABASE dblab;
Query OK, 1 row affected (0.01 sec)
```

Examples 2

- Available databases:

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| maria              |
| maria_2            |
+--------------------+
3 rows in set (0.03 sec)
```

- Selection of a database to use:

```
mysql> use maria
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

Examples 3

- Show all the tables in the database in use:
  `SHOW TABLES;`

```
mysql> show tables;
+------------------+
| Tables_in_maria  |
+------------------+
| APPLIES_FOR      |
| CANDIDATE        |
| Course           |
| JOB              |
| Professors       |
| Staff            |
| SubjectCoverAreas |
| SupportingStaff  |
| Teach            |
| books            |
| course           |
| lecture          |
| product          |
| professor        |
| registration     |
| student          |
+------------------+
16 rows in set (0.02 sec)
```

Examples 5

- **`DESCRIBE table_name;`**

```
mysql> describe course
    -> ;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| title      | varchar(255) | NO   | UNI | unknown |                |
| material   | text         | YES  |     | NULL    |                |
| course_id  | int(4)       | NO   | PRI | NULL    | auto_increment |
| supervisor | varchar(255) | NO   | MUL | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
```

Examples 6

- **SHOW CREATE TABLE** table_name;

```
mysql> show create table course;
+--------+---------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
| Table  | Create Table


+--------+---------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
| course | CREATE TABLE `course` (
  `title` varchar(255) NOT NULL DEFAULT 'unknown',
  `material` text,
  `course_id` int(4) NOT NULL AUTO_INCREMENT,
  `supervisor` varchar(255) NOT NULL,
  PRIMARY KEY (`course_id`),
  UNIQUE KEY `title` (`title`),
  KEY `SUPERVISED` (`supervisor`),
  CONSTRAINT `SUPERVISED` FOREIGN KEY (`supervisor`) REFERENCES `professor` (`email`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=latin1 |
+--------+---------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
----------------------------------------------------------------------------------------------------------
1 row in set (0.00 sec)
```

Examples 7

- Show all the contents of the table

  `SELECT * FROM table_name;`

```
mysql> select * from course;
+--------------+-------------------------+-----------+----------------------+
| title        | material                | course_id | supervisor           |
+--------------+-------------------------+-----------+----------------------+
| Databases    | Intro to relational DBs |         2 | pap@ceid.upatras.gr  |
| Databases II | Advanced DBs            |         3 | alex@ceid.upatras.gr |
+--------------+-------------------------+-----------+----------------------+
2 rows in set (0.00 sec)
```

Tables

- Each table consists of fields (columns), which represent the attributes.
- Each field must have its data type defined
- The proper data type to store the values that the attribute can take should always be used.
- The size of the field must be defined, according to the needs.

SQL Data Types, arithmetic values

- **INT**
  - A medium integer. Signed range is from -2147483648 to 2147483647. Unsigned range is from 0 to 4294967295. The size parameter specifies the maximum display width (which is 255)
- **TINYINT**: integer up to 4 digits
  - (1 byte) -128 to 127 normal. 0 to 255 UNSIGNED
- **SMALLINT**: integer up to 5 digits
  - (2 bytes) -32768 to 32767 normal. 0 to 65535 UNSIGNED
- **MEDIUMINT**: (3 bytes) integer up to 9 digits
- **BIGINT**: (8 bytes) integer up to 20 digits
- **FLOAT(M,D):** floating point M digits and D decimal digits (up to 23), SIGNED
- **DOUBLE(M,D):** double precision, M digits and D decimal digits (up to 53), SIGNED

SQL Data Types– date and time

- **DATE**:
  YYYY-YY-DD (1973-12-30, 2017-01-26)
- **TIME:**
  HH:MM:SS (18:46:05)
- **DATETIME:** DATE AND TIME
  YYYY-YY-DD HH:MM:SS (1992-12-30 15:30:00)
- **YEAR(M):** year. With two digits the years 1970-2069 are included . With 4 digits years 1901-2155 are included.
- **TIMESTAMP:** 14(YYYYMMDDHHMMSS) 12(YYMMDDHHMMSS) 8(YYYYMMDD) 6(HHMMSS)
  - EG19731230101550

SQL Data Types - Alp arithmetic

- **CHAR(M):**A FIXED length string (can contain letters, numbers, and special characters). The size parameter specifies the column length in characters - can be from 0 to 255. Default is 1 **VARCHAR(M):** Αλφαριθμητικό μεταβλητού μεγέθους, από 1-255 χαρακτήρες. *Πρέπει να καθορίζουμε το Μ*.
- **TEXT**: Holds a string with a maximum length of 65,535 bytes
- **TINYTEXT**: Holds a string with a maximum length of 255 characters
- **MEDIUMTEXT**: Holds a string with a maximum length of 16.777.215 characters
- **LONGTEXT**: Holds a string with a maximum length of 4.294.967.295 characters

SQL Data Types - Alp arithmetic

- **ENUM(v1, v2, v3, ...)** A string object that can have ONLY one value, chosen from a list of possible values. If a value is inserted that is not in the list, a blank value will be inserted. The values are sorted in the order you enter them
  - eg ENUM('A', 'B', 'C') the filed can have valu A or B or C
- **SET(v1, v2, v3, ...)** A string object similar to ENUM, that can have 0 or more values, chosen from a list of possible values. You can list up to 64 values in a SET list
  - eg SET('A', 'B', 'C') the field can have value A or B or A,B,C

MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LONGBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

Create table

- CREATE TABLE statement syntax :

CREATE TABLE [IF NOT EXISTS] table_name(
column_name column_type [options],
column_name2 column_type2 [options2], κλπ..
PRIMARY KEY (column_key_name1,…)
UNIQUE (index_col_name,…)
[**CONSTRAINT** constraint_code
FOREIGN KEY (index_col_name,…)
REFERENCES tbl_name [(index_col_name,…)]
[**ON DELETE** reference_option]
[**ON UPDATE** reference_option]
);

Create table

- CREATE TABLE statement syntax:

```
CREATE TABLE [IF NOT EXISTS] table_name(
column_name column_type [options],
 column_name2 column_type2 [options2], κλπ..
PRIMARY KEY (column_key_name1,…)
UNIQUE (index_col_name,…)
[CONSTRAINT constraint_code
FOREIGN KEY (index_col_name,…)
REFERENCES tbl_name [(index_col_name,…)]
[ON DELETE reference_option]
[ON UPDATE reference_option]
);
```

Create table

CREATE TABLE [IF NOT EXISTS] table_name(
column_name column_type [**options**],
column_name2 column_type2 [**options2**], κλπ..
……

- **options** = [NOT NULL | NULL] [DEFAULT default_value] [AUTO_INCREMENT]
  - **NOT NULL**: the value NULL is not allowed
  - **NULL**: the value NULL is allowed
  - **DEFAULT**: a default value that is used when a value is not given
  - **AUTO_INCREMENT**: auto increment by 1. starting from 1 except we set another starting value (e.g. AUTO_INCREMENT =30)

Create table

```
CREATE TABLE …….

REFERENCES tbl_name [(index_col_name,...)]
[ON DELETE reference_option]
[ON UPDATE reference_option]
```

- **reference option** = RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT
  - **Restrict : Nothing is going to be delete if there is a child row**
  - **Cascade : the child row will be delete / update too**
  - **Set Null : the child column will be set to null if you delete the parent**
  - **No action : The child row will not be concerned of the delete / update**

Example : Create table (columns)

- We define an attribute that contains the code of a product. It is a 9-digit integer. We want it to never be null and to automatically increase its value as each new product is added. CREATE TABLE statement would be:

  product_id **INT**(9) **NOT NULL** **AUTO_INCREMENT**

- We also want another feature which is the buyer's ID number. It is alphanumeric, 7 characters long.

  buyer_id **VARCHAR**(7)

- There is another attribute the product name, a 20-character alphanumeric that is never NULL. It has a default value of “unknown”.

  product_name **VARCHAR**(20) **DEFAULT** 'unknown' **NOT NULL**

Example : Create table (keys)

- We want to set the product_id argument to be a primary key in the table. We add the line::

  PRIMARY KEY(product_id)

- We know that the product name is unique and can be used as an alternate key. We state it as follows:

  UNIQUE (product_name)

Example : Create table (Foreign keys)

- Let buyer_id be a foreign key that "points" to the Ar_Tayt argument of the Customer table (pelatis). We put a constraint named PRDCTBR and declare the referential integrity constraint(Referential integrity is the condition of a set of tables in a database in which all references from one table to another are valid.)

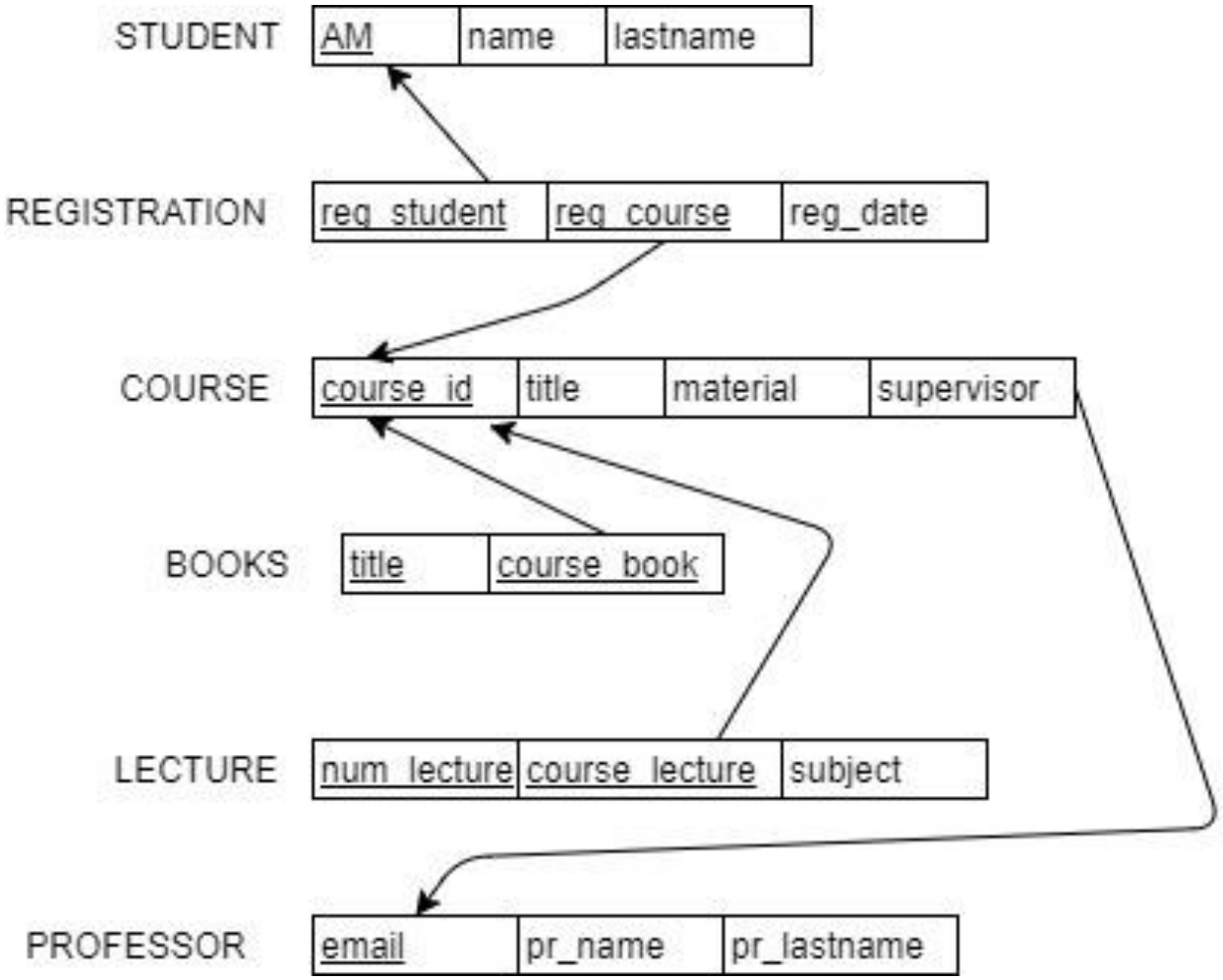
  CONSTRAINT PRDCTBR
  FOREIGN KEY(buyer_id) **REFERENCES** pelatis(Ar_Tayt)
  ON DELETE SET NULL ON UPDATE CASCADE

  - The operation ON DELETE SET NULL means that in case the corresponding customer record is deleted, the field is set to NULL.
  - The ON UPDATE CASCADE action means that if the customer ID number (Ar_Tayt) changes, the buyer_id field is automatically updated.

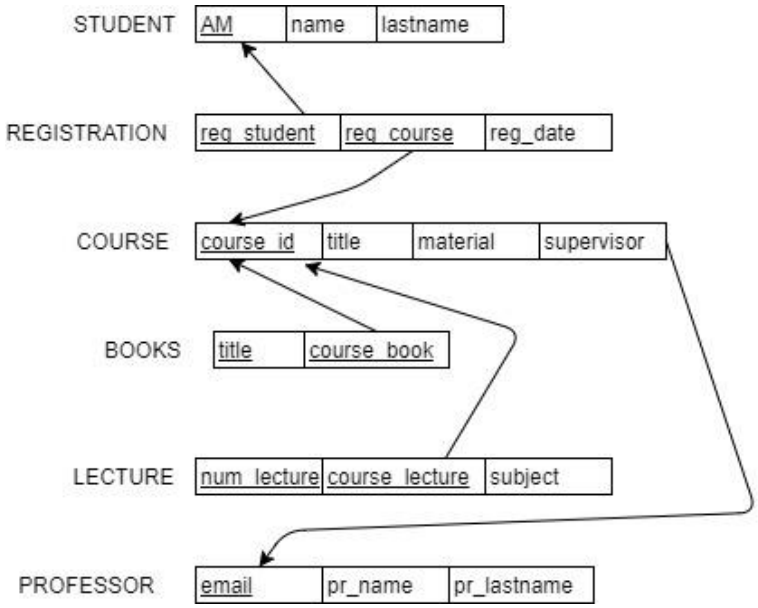
Example

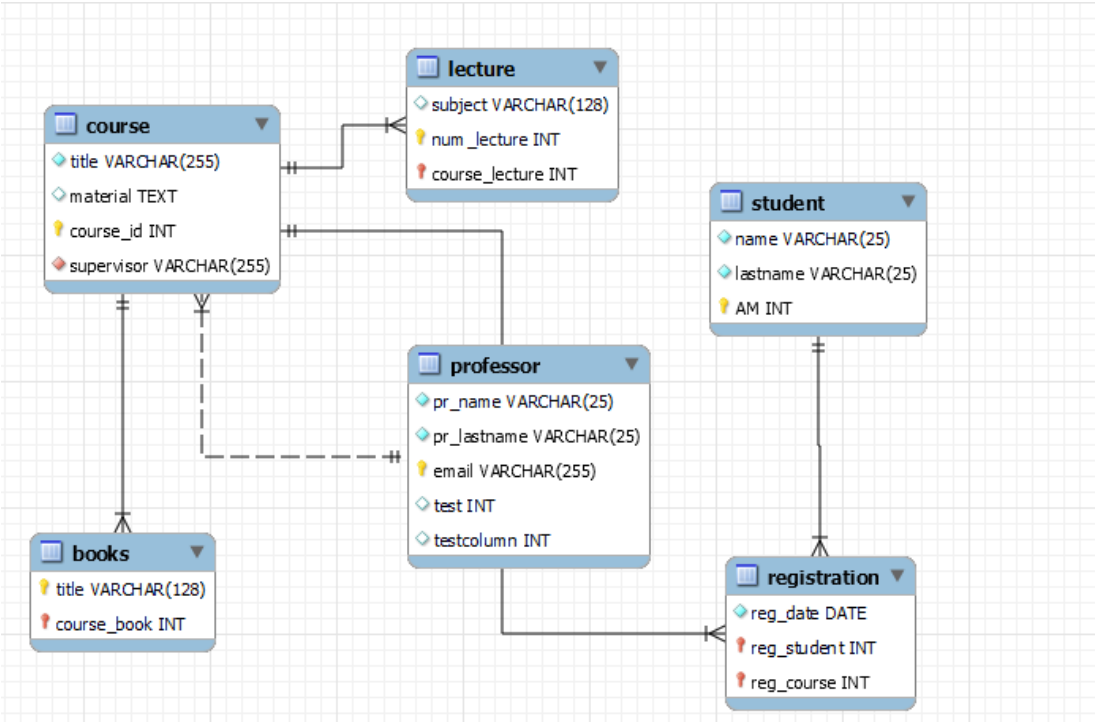
```
mysql> CREATE TABLE product(
    -> product_id INT(9) NOT NULL AUTO_INCREMENT,
    -> buyer_id VARCHAR(7),
    -> product_name VARCHAR(20) DEFAULT 'unknown' NOT NULL,
    -> PRIMARY KEY(product_id),
    -> UNIQUE (product_name),
    -> CONSTRAINT PRDCTBR
    -> FOREIGN KEY(buyer_id) REFERENCES pelatis(Ar_Tayt)
    -> ON DELETE SET NULL ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected (0.08 sec)
```

Example

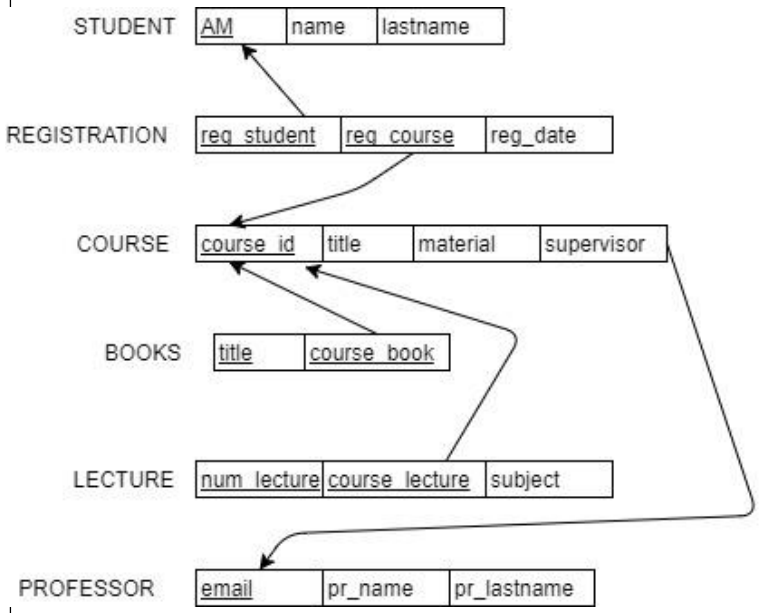

STUDENT
AM
name
lastname
REGISTRATION
reg_student
reg_course
reg_date
COURSE
course_id
title
material
supervisor
BOOKS
title
course_book
LECTURE
num_lecture
course_lecture
subject
PROFESSOR
email
pr_name
pr_lastname

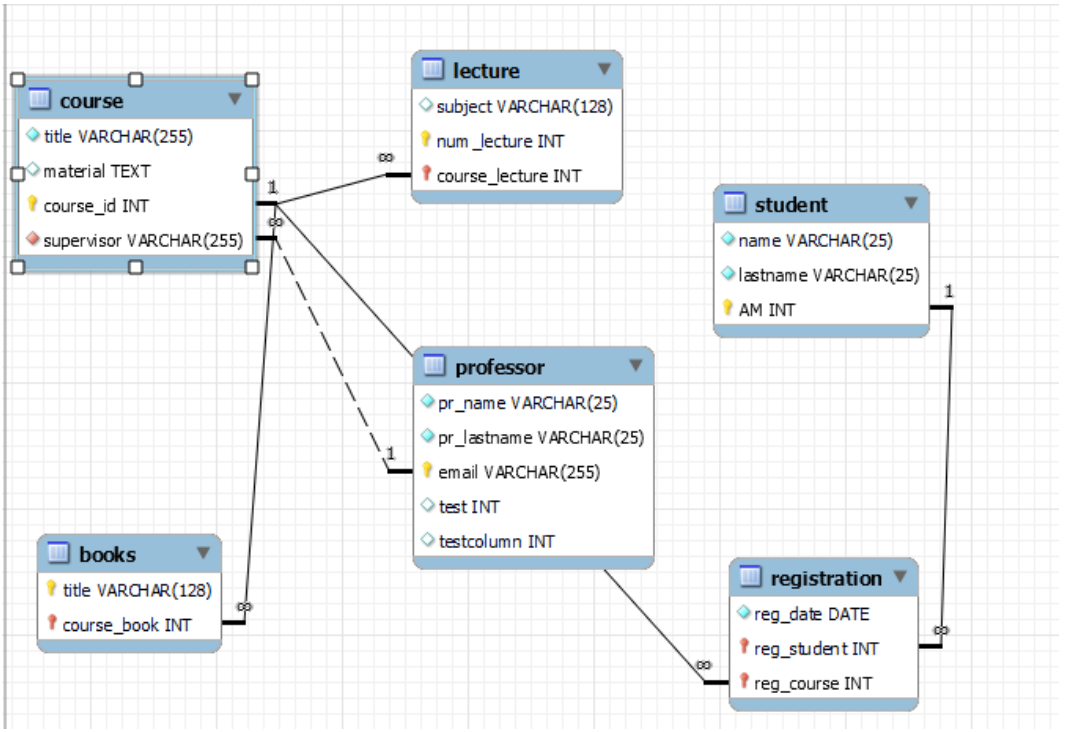
Alternative representations





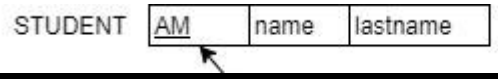
Alternative representations





Example – Create tables

```
CREATE TABLE student(
   name VARCHAR(25) DEFAULT 'unknown' NOT NULL,
   lastname VARCHAR(25) DEFAULT 'unknown' NOT NULL,
   AM INT(5) NOT NULL AUTO_INCREMENT,
   PRIMARY KEY(AM)
   );

CREATE TABLE professor(
   pr_name VARCHAR(25) DEFAULT 'unknown' NOT NULL,
   pr_lastname VARCHAR(25) DEFAULT 'unknown' NOT NULL,
   email VARCHAR(255) NOT NULL,
   PRIMARY KEY(email)
   );
```



command line

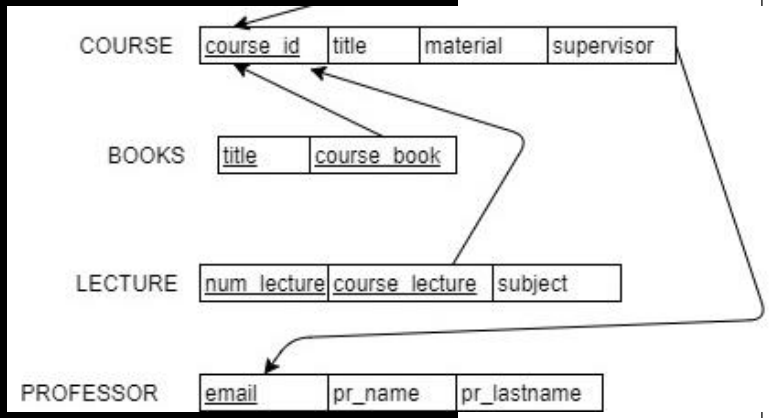
Create databases maria

Create the first two tables

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| empty              |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| travel_agency      |
+--------------------+
6 rows in set (0.00 sec)

mysql> create database maria;
Query OK, 1 row affected (0.01 sec)
```

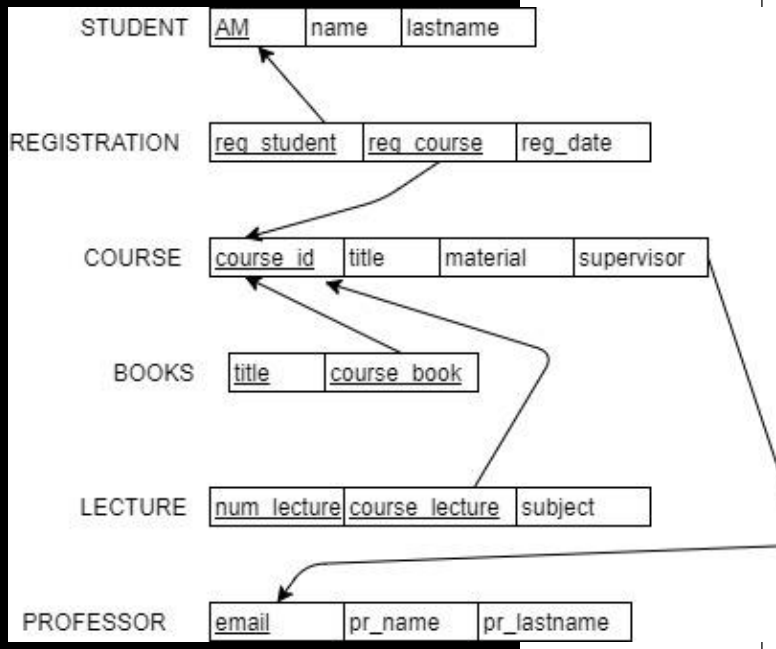
```
mysql> use maria;
Database changed
mysql> CREATE TABLE student(
    -> name VARCHAR(25) DEFAULT 'unknown' NOT NULL,
    -> lastname VARCHAR(25) DEFAULT 'unknown' NOT NULL,
    -> AM INT(5) NOT NULL AUTO_INCREMENT,
    -> PRIMARY KEY(AM)
    -> );
Query OK, 0 rows affected, 1 warning (0.02 sec)

mysql> show tables;
+-----------------+
| Tables_in_maria |
+-----------------+
| student         |
+-----------------+
1 row in set (0.00 sec)

mysql> CREATE TABLE professor(
    -> pr_name VARCHAR(25) DEFAULT 'unknown' NOT NULL,
    -> pr_lastname VARCHAR(25) DEFAULT 'unknown' NOT NULL,
    -> email VARCHAR(255) NOT NULL,
    -> PRIMARY KEY(email)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----------------+
| Tables_in_maria |
+-----------------+
| professor       |
| student         |
+-----------------+
2 rows in set (0.00 sec)

mysql>
```

```
CREATE  TABLE course(
   title VARCHAR(255) DEFAULT 'unknown' NOT NULL,
   material TEXT,
   course_id INT(4) NOT NULL AUTO_INCREMENT,
   supervisor VARCHAR(255) NOT NULL,
   PRIMARY KEY(course_id),
   UNIQUE(title),
   CONSTRAINT SUPERVISED
   FOREIGN KEY (supervisor) REFERENCES professor(email)
   ON DELETE CASCADE ON UPDATE CASCADE
   );

CREATE  TABLE books(
   title VARCHAR(128)  DEFAULT  'Title' NOT NULL,
   course_book  INT(4) NOT NULL,
   PRIMARY KEY (title, course_book),
   CONSTRAINT CRSBOOK
   FOREIGN KEY (course_book) REFERENCES course(course_id)
   ON DELETE CASCADE ON UPDATE CASCADE
);
```



ON DELETE CASCADE constraint is **used in MySQL to delete the rows from the child table automatically, when the rows from the parent table are deleted**.

```
CREATE  TABLE IF NOT EXIST lecture(
   subject VARCHAR(128),
   num_lecture INT(2) NOT NULL,
   course_lecture INT(4) NOT NULL,
   PRIMARY KEY(num_lecture, course_lecture),
   CONSTRAINT CRSLECTURE FOREIGN KEY (course_lecture)
   REFERENCES course(course_id)
   ON DELETE CASCADE ON UPDATE CASCADE
   );


CREATE TABLE registration(
   reg_date DATE NOT NULL,
   reg_student INT(5) NOT NULL,
   reg_course INT(4) NOT NULL,
   PRIMARY KEY (reg_student, reg_course),
   CONSTRAINT CRSREGISTRATION FOREIGN KEY (reg_course) REFERENCES
   course(course_id) ON DELETE CASCADE ON UPDATE CASCADE,
   CONSTRAINT STDNTREGISTRATION FOREIGN KEY (reg_student)
   REFERENCES student(AM) ON DELETE CASCADE ON UPDATE CASCADE
);
```



Insert data

INSERT INTO table_name [(column1, column2…)]
VALUES (value1, value2,…)

- When we include values for all columns we don't need to mention the column names (not recommended) .
- In any column we do not want to specify a value, we can instead put the value NULL ή DEFAULT.
- **If NULL is given to a field with the AUTO_INCREMENT property, then the value will be calculated automatically.**
- We can insert multiple records with one insert as follows:

INSERT INTO table **VALUES**
(value11,value12,…, value1n),
(value21,value22,…, value2n),
…
(valuen1,valuen2,…, valuenn);

Insert

- **INSERT INTO** professor(pr_name,pr_lastname,email) **VALUES** (DEFAULT, 'Papadopoulos','pap@ceid.gr');

or

- **INSERT INTO professor VALUES** (DEFAULT, 'Georgiou','geo@ceid.gr');

Important !

- **INSERT INTO professor values(**DEFAULT,'Georgiou');

ERROR 1136 (21S01): Column count doesn't match value count at row 1

```
mysql> select * from professor;
+-----------+-------------+----------------------+
| pr_name   | pr_lastname | email                |
+-----------+-------------+----------------------+
| Alexandra | unknown     | alex@ceid.upatras.gr |
| unknown   | Dimitriou   | dim@ceid.upatras.gr  |
| Maria     | Papadopoulou| pap@ceid.upatras.gr  |
+-----------+-------------+----------------------+
3 rows in set (0.02 sec)

mysql> select * from course;
+--------------+-------------------------+-----------+----------------------+
| title        | material                | course_id | supervisor           |
+--------------+-------------------------+-----------+----------------------+
| Databases    | Intro to relational DBs |         2 | pap@ceid.upatras.gr  |
| Databases II | Advanced DBs            |         3 | alex@ceid.upatras.gr |
+--------------+-------------------------+-----------+----------------------+
2 rows in set (0.00 sec)
```

Delete

- To delete a record from a table :

  DELETE FROM table_name
  [**WHERE** some_column=some_value];

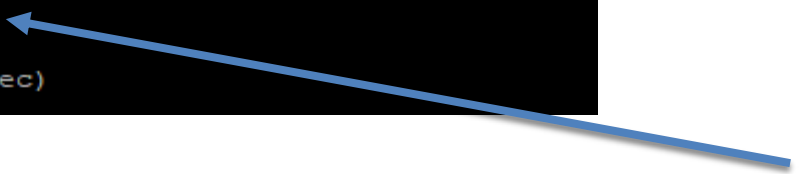
- **where** defines **which records will be deleted**
- If WHERE is not defined , **all** the records will be deleted
- examples:

  DELETE FROM professor; //διαγράφει τα πάντα

  DELETE FROM professor **WHERE** pr_lastname like '%eorgiou%';

Examples Delete

```
mysql> CREATE TABLE buildings (
    ->     building_no INT PRIMARY KEY AUTO_INCREMENT,
    ->     building_name VARCHAR(255) NOT NULL,
    ->     address VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.12 sec)

mysql> CREATE TABLE rooms (
    ->     room_no INT PRIMARY KEY AUTO_INCREMENT,
    ->     room_name VARCHAR(255) NOT NULL,
    ->     building_no INT NOT NULL,
    ->     FOREIGN KEY (building_no)
    ->         REFERENCES buildings (building_no)
    ->         ON DELETE CASCADE
    -> );
Query OK, 0 rows affected (0.11 sec)
```



```
mysql> INSERT INTO buildings(building_name,address)
    -> VALUES('ACME Headquaters','3950 North 1st Street CA 95134'),
    ->       ('ACME Sales','5000 North 1st Street CA 95134');
Query OK, 2 rows affected (0.04 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM buildings;
+-------------+-----------------+--------------------------------+
| building_no | building_name   | address                        |
+-------------+-----------------+--------------------------------+
|           1 | ACME Headquaters | 3950 North 1st Street CA 95134 |
|           2 | ACME Sales      | 5000 North 1st Street CA 95134 |
+-------------+-----------------+--------------------------------+
2 rows in set (0.00 sec)
```

```
mysql> INSERT INTO rooms(room_name,building_no)
    -> VALUES('Amazon',1),
    ->        ('War Room',1),
    ->        ('Office of CEO',1),
    ->        ('Marketing',2),
    ->        ('Showroom',2);
Query OK, 5 rows affected (0.02 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM rooms;
+---------+---------------+-------------+
| room_no | room_name     | building_no |
+---------+---------------+-------------+
|       1 | Amazon        |           1 |
|       2 | War Room      |           1 |
|       3 | Office of CEO |           1 |
|       4 | Marketing     |           2 |
|       5 | Showroom      |           2 |
+---------+---------------+-------------+
5 rows in set (0.00 sec)
```

```
mysql> DELETE FROM buildings
    -> WHERE building_no = 2;
Query OK, 1 row affected (0.04 sec)
```

```
mysql> SELECT * FROM rooms;
+---------+---------------+-------------+
| room_no | room_name     | building_no |
+---------+---------------+-------------+
|       1 | Amazon        |           1 |
|       2 | War Room      |           1 |
|       3 | Office of CEO |           1 |
+---------+---------------+-------------+
3 rows in set (0.00 sec)
```

Examples DELETE

```
| name           | lastname                | AM   |
+----------------+-------------------------+------+
| βασιλης | παπαϊωαννου |    1 |
| Spiros         | Petridis                | 2190 |
| Vivi           | Tzekou                  | 2191 |
| unknown        | Dourou                  | 2192 |
| Athanasia      | Koumpouri               | 2193 |
| Vaso           | unknown                 | 2194 |
+----------------+-------------------------+------+
6 rows in set (0.00 sec)
```

DELETE FROM student WHERE name like ‘%βασ%’;

```
mysql> delete from student where name like '%βασ%';
Query OK, 1 row affected (0.08 sec)

mysql> select * from student;
+-----------+-----------+------+
| name      | lastname  | AM   |
+-----------+-----------+------+
| Spiros    | Petridis  | 2190 |
| Vivi      | Tzekou    | 2191 |
| unknown   | Dourou    | 2192 |
| Athanasia | Koumpouri | 2193 |
| Vaso      | unknown   | 2194 |
+-----------+-----------+------+
5 rows in set (0.00 sec)
```

Update

- To update a record in a table :

```
UPDATE table_name
SET column1=value1, column2=value2,...
WHERE some_column=some_value;
```

- **where** defines **which records will be updated**
- **Without where** all the records will be updated

```
UPDATE professor
SET pr_name='Nikos', pr_lastname='Papadakis'
WHERE  pr_lastname='unknown';


UPDATE professor
SET salary=salary*1,05
WHERE  salary<2000;
```

```
mysql> select * from student;
+-----------+-----------+------+
| name      | lastname  | AM   |
+-----------+-----------+------+
| Spiros    | Petridis  | 2190 |
| Vivi      | Tzekou    | 2191 |
| unknown   | Dourou    | 2192 |
| Athanasia | Koumpouri | 2193 |
| Vaso      | unknown   | 2194 |
+-----------+-----------+------+
5 rows in set (0.00 sec)
```

UPDATE student

SET NAME = ‘Eleni’, lastname=‘Voyiatzaki’,

WHERE lastname=‘Koumpouri’;

```
mysql> update student
    -> set name = 'Eleni', lastname='Voyiatzaki' where lastname='Koumpouri';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from student;
+---------+------------+------+
| name    | lastname   | AM   |
+---------+------------+------+
| Spiros  | Petridis   | 2190 |
| Vivi    | Tzekou     | 2191 |
| unknown | Dourou     | 2192 |
| Eleni   | Voyiatzaki | 2193 |
| Vaso    | unknown    | 2194 |
+---------+------------+------+
5 rows in set (1.42 sec)

mysql>
```

ALTER

- We can change the structure of a table
  - Add, delete or modify a table

    ALTER TABLE table_name
    ADD||DROP||MODIFY column_name datatype;

- examples:

  ALTER TABLE professor **DROP** pr_name, **ADD** address **VARCHAR**(70) **NOT NULL**;

  ALTER TABLE student **MODIFY** name varchar(30) not null;

ALTER …ADD

```
mysql> select * from professor;
+-----------+-------------+----------------------+
| pr_name   | pr_lastname | email                |
+-----------+-------------+----------------------+
| Alexandra | unknown     | alex@ceid.upatras.gr |
| unknown   | Dimitriou   | dim@ceid.upatras.gr  |
| Maria     | Papadopoulou| pap@ceid.upatras.gr  |
+-----------+-------------+----------------------+
3 rows in set (0.02 sec)
```

ALTER TABLE professor **DROP** pr_name, **ADD** address **VARCHAR**(70) **NOT NULL**;

```
mysql> ALTER TABLE professor DROP pr_name, ADD address VARCHAR(70) NOT NULL;
Query OK, 3 rows affected (0.32 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from professor;
+-------------+----------------------+---------+
| pr_lastname | email                | address |
+-------------+----------------------+---------+
| unknown     | alex@ceid.upatras.gr |         |
| Dimitriou   | dim@ceid.upatras.gr  |         |
| Papadopoulou| pap@ceid.upatras.gr  |         |
+-------------+----------------------+---------+
3 rows in set (0.00 sec)
```

Can we alter primary key in MySQL?
To change the primary key of a table, delete the existing key using a DROP clause in an ALTER TABLE statement and add the new primary key.

ALTER...MODIFY

```
mysql> describe student;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| name     | varchar(25) | NO   |     | unknown |                |
| lastname | varchar(25) | NO   |     | unknown |                |
| AM       | int(5)      | NO   | PRI | NULL    | auto_increment |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)
```

- ALTER TABLE student MODIFY name varchar(30) NOT NULL;

```
mysql> alter table student modify name varchar(30) not null;
Query OK, 5 rows affected (0.21 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> describe student;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| name     | varchar(30) | NO   |     | NULL    |                |
| lastname | varchar(25) | NO   |     | unknown |                |
| AM       | int(5)      | NO   | PRI | NULL    | auto_increment |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)
```

ALTER...MODIFY

```
mysql> describe student;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| name     | varchar(30) | NO   |     | NULL    |                |
| lastname | varchar(25) | NO   |     | unknown |                |
| AM       | int(5)      | NO   | PRI | NULL    | auto_increment |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)
```

- ALTER table student MODIFY name NULL;

```
mysql> alter table student modify name varchar(30) null;
Query OK, 5 rows affected (0.25 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> describe student;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| name     | varchar(30) | YES  |     | NULL    |                |
| lastname | varchar(25) | NO   |     | unknown |                |
| AM       | int(5)      | NO   | PRI | NULL    | auto_increment |
+----------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)
```

```
mysql> CREATE  TABLE course(
    -> title VARCHAR(255) DEFAULT ‘unknown’ NOT NULL,
    -> material TEXT,
    -> course_id INT(4) NOT NULL AUTO_INCREMENT,
    -> supervisor VARCHAR(255) NOT NULL,
    -> PRIMARY KEY(course_id),
    -> UNIQUE(title),
    -> CONSTRAINT SUPERVISED
    -> FOREIGN KEY (supervisor) REFERENCES professor(email)
    -> ON DELETE CASCADE ON UPDATE CASCADE
    -> );
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near '‘unknown’ N
NULL,material TEXT,course_id INT(4) NOT NULL AUTO_INCREMENT,' at line 2
```

Be careful :‘ ’

```
mysql> CREATE  TABLE course(
    -> title VARCHAR(255) NOT NULL,
    -> material TEXT,
    -> course_id INT(4) NOT NULL AUTO_INCREMENT,
    -> supervisor VARCHAR(255) NOT NULL,
    -> PRIMARY KEY(course_id),
    -> UNIQUE(title),
    -> CONSTRAINT SUPERVISED
    -> FOREIGN KEY (supervisor) REFERENCES professor(email)
    -> ON DELETE CASCADE ON UPDATE CASCADE
    -> );
Query OK, 0 rows affected, 1 warning (0.07 sec)

mysql> alter table course modify title VARCHAR(255) Default 'uknown' not null;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

Wildcard Characters in SQL Server

Retrieved from https://www.w3schools.com/sql/sql_wildcards.asp

Symbol	Description	Example
%	Represents zero or more characters	bl% finds bl, black, blue, and blob
_	Represents a single character	h_t finds hot, hat, and hit
[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
^	Represents any character not in the brackets	h[^oa]t finds hit, but not hot and hat
-	Represents any single character within the specified range	c[a-b]t finds cat and cbt

Here are some examples showing different `LIKE` operators with '%' and '_' wildcards:

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a__%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

- **MySQL** is a **relational database management system based on** SQL – Structured Query Language. The application is used for a wide range of purposes, including data warehousing, e-commerce, and logging applications.
- **MariaDB** is a community-developed, commercially supported fork of the MySQL relational database management system (RDBMS), intended to remain free and open-source software under the GNU General Public License. Development is led by some of the original developers of MySQL, who forked it due to concerns over its acquisition by Oracle Corporation in 2009.A **database engine** (or **storage engine**) is the underlying software component that a database management system (DBMS) uses to create, read, update and delete (CRUD) data from a database.
- **InnoDB** is a storage engine for the database management system MySQL and MariaDB.
- **MySQL Workbench** is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration, backup, and much more.
- **XAMPP** is **a software distribution which provides the Apache web server, MySQL database** (actually MariaDB), Php and Perl (as command-line executables and Apache modules) all in one package. It is available for Windows, MAC and Linux systems. No configuration is necessary to integrate Php with MySQL. MySQL Workbench is available on Windows, Linux and Mac OS X.