

Please read Sections 4 – 4.4 of the textbook and then answer the following, trying not to look at your notes or at the textbook. Quiz #2, on Wednesday 20th September, will consist exclusively of questions taken from the Part 1 of this homework.

Part I — Short Questions

Question 1

What does it mean to say that SQL is at the same time a “data definition language” and a “data manipulation language”?

Question 2

Name three kind of objects (for lack of a better word) a CREATE statement can create.

Question 3

Complete the following table, giving at least two different examples when asked for examples:

Data Type	Examples
Int	4, -32
Char(4)
.....	'Train', 'Michelle'
Bit(4)
.....	TRUE, UNKNOWN

Question 4

Explain what the following SQL statement does

```
CREATE SCHEMA faculty;
```

Question 5

If I want to enter January 21, 2016, as a value for an attribute with the DATE datatype, what value should I enter?

Question 6

Write a statement that inserts the values “Thomas” and “4” into the table **Trains**.

Question 7

In the following statement, can PkgName be a primary key?

```
SELECT * FROM myTable WHERE PkgName = 'MySQL';
```

Question 8

The CHECK clause implement an implicit, an explicit, or a semantic constraint? Justify your answer.

Question 9

If you want every tuple referencing the tuple you’re about to delete to be deleted as well, what mechanism should you use?

Question 10

If a database designer is using the **ON UPDATE SET NULL** for a foreign key, what mechanism is he implementing (i.e., describe how the database will react a certain operation)?

Question 11

If the following is part of the design of a schema:

```
FOREIGN KEY (DptNumber) REFERENCES DEPARTMENT(DptNumber)
```

```
ON DELETE SET DEFAULT
```

```
ON UPDATE CASCADE
```

- (a) What happen to the tuples with foreign key DptNumber set to 3 if the tuple with primary key DptNumber set to 3 is deleted?
- (b) What happen to the tuples with foreign key DptNumber set to 3 if the tuple with primary key DptNumber set to 3 is updated?

Question 12

Given a relation **TOURIST**(Name, EntryDate, Address), write a SQL statement printing the name and address of all the persons who entered the territory the 15 September, 2012.

Question 13

What is the fully qualified name of an attribute? Give an example.

Question 14

If DEPARTMENT is a database, what is DEPARTMENT.*?

Question 15

What is a multi-set? What does it mean to say that SQL treats tables as multisets?

Question 16

What is the difference between those two queries?

```
SELECT ALL * FROM myTable;
```

and

```
SELECT DISTINCT * FROM myTable;
```

How are the results the same? How are they different?



Part II — Problems

This part will help you in assessing your level of understanding of this lecture, and give you an idea of the kind of problem you will be asked to solve during the exams. This time, all the problems require a computer. They might look long, but it is because the instructions are detailed: follow them carefully and the three problems shouldn't take you very long. I'll assume that you will have successfully completed them by the time Homework #3 is released (Wednesday 20th September), so don't wait and let me know if you had difficulties doing them.

Problem 1

This part will make your set-up easier and safer to use, by distinguishing properly the *administrator* of the database (called *root*) and its *user*, *testuser*. We will first grant (as an administrator) the right to *testuser* to create and administrate databases whose name starts with HW. Then, we will create our first database as *testuser*. Note that you won't be able to do any of the problems of the homeworks if you don't complete this exercise successfully

1. Connect to your MySQL DBMS as *root*:

- In Windows, open a command prompt (search for “cmd”) and type

```
cd "C:\Program Files\MySQL\MySQL Server 5.7\bin"
```

- In Linux, open a shell and log-in as *root*.

Then, in both cases, type

```
mysql -u root -p
```

and enter the password for the *root* account you picked during the first homework. If you forgot it, think again, wait a bit, and think again. If you really don't remember it, you may want to try what's at <https://dev.mysql.com/doc/refman/5.7/en/resetting-permissions.html>, but it is a complex procedure.

Once you're logged in, enter

```
GRANT ALL PRIVILEGES ON `HW%`.* TO 'testuser'@'localhost';
```

and be careful: back-ticks are surrounding “HW%”, whereas single quotes are surrounding “testuser” and “localhost” (normally, copy-and-pasting from this document should work). Then, exit, using

```
EXIT;
```

2. Connect to your MySQL DBMS as *testuser*:

- In windows, open a command prompt (search for “cmd”) and type

```
cd "C:\Program Files\MySQL\MySQL Server 5.7\bin"
```

- In Linux, open a shell (as a normal user)

Then, in both cases, type

```
mysql -u testuser -p
```

and enter the password password.

3. Create a new database called HW2Q1

```
CREATE DATABASE HW2Q1;
```

4. List the databases

```
SHOW DATABASES;
```

Make sure HW2Q1 is a part of it.

In the future, we will refer to the commands 2. and 3. as “log-in as *testuser* and create a database HW2Q1” (you won't need to re-do 1., we did it once and for all). If, while you're connected to your MySQL database, you want to remove a database named *wrongName*, type

```
DROP DATABASE wrongName;
```

Note that by pressing the up arrow on your keyboard, you can access the commands you previously typed.

Problem 2

This exercise, and the following ones, assume you successfully completed Problem 1. Log in as testuser and create a database HW2Q2.

Now, let us tell MySQL that we want to use that database

```
USE HW2Q2;
```

and ask what it contains

```
SHOW TABLES;
```

Let's create a first table named name

```
CREATE TABLE name(
    FName VARCHAR(15),
    LName VARCHAR(15),
    Id INT,
    PRIMARY KEY(Id)
);
```

and a second table, named address

```
CREATE TABLE address(
    StreetName VARCHAR(15),
    Number INT,
    Habitants INT,
    PRIMARY KEY(StreetName, Number)
);
```

To make sure we created those two tables, we can use the command

```
SHOW TABLES;
```

But how to make sure that you entered the attributes correctly? One way to make sure is to enter the command

```
DESC address;
```

and to examine carefully the message printed. You should read

```
+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| StreetName | varchar(15)   | NO   | PRI | NULL    |       |
| Number     | int(11)       | NO   | PRI | NULL    |       |
| Habitants  | int(11)       | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

If you believe there is a mistake, you can erase ("drop") the table using

```
DROP TABLE address;
```

and then re-create it. Of course, you can do the same for the name table.

Now, let us add a foreign key to the address table:

```
ALTER TABLE address
    ADD CONSTRAINT fk_id FOREIGN KEY (Habitants)
    REFERENCES HW2Q2.name(Id);
```

And observe the modification:

```
DESC address;
```

Now, let's populate the name table with some data:

```
INSERT INTO name VALUES ('Harrison', 'Yates', 003);
INSERT INTO name VALUES ('Tuong Lu', 'Kim', 004);
INSERT INTO name VALUES ('Lemmiwinks', NULL, 080);
```

and let's have a look to what we have:

```
SELECT * FROM name;
```

Do you notice anything regarding the values we entered for the Id attribute? Let's add some data to the address table (using a different syntax):

```
INSERT INTO address (StreetName, Number, Habitants)
VALUES
    ('Armstrong Drive', 10019, 003),
    ('North Broad St.', 23, 004),
    ('Robert Lane', 120, NULL);
```

And let's use our first update command:

```
UPDATE address SET Habitants = 3 WHERE Number = 120;
```

Now, answer the following:

1. Draw the relations corresponding to that database, including the domains and primary, as well as foreign, keys.
2. Write a statement that violate the entity integrity constraint. What is the error message returned?
3. Execute an UPDATE statement that violate the referential integrity constraint. What is the error message returned?
4. Write a statement that violate another kind of constraint. Explain what constraint you are violating, and explain the error message.

Problem 3

Log in as `testuser` and create a database `HW2Q3`. Tell MySQL that you want to use that database, and create a table (I will assume you named it `example` in the following, but you are free to name it the way you want) with at least two attributes that have different data types. Don't declare a primary key yet. Answer the following:

1. Add a tuple to your table using

```
INSERT INTO example VALUES(X, Y);
```

where "X" and "Y" are values with the right type. Try to add this tuple again. What do you observe? (You can use

```
SELECT * FROM example;
```

to observe what is stored in this table.)

2. Alter your table to add a primary key, using

```
ALTER TABLE example ADD PRIMARY KEY (attribute);
```

where `attribute` is the name of the attribute you want to be a primary key. What do you observe?

3. Empty your table using

```
DELETE FROM example;
```

and alter your table to add a primary key, using the command we gave at the previous step. What do you observe?

4. Try to add the same tuple twice. What do you observe?