

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 7th February, will consist of questions taken or inspired from Part I & II 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

If PkgName is a primary key, what can you tell about the number of rows returned by the following statement?

```
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
```

What happen to the row whose foreign key DptNumber is set to 3 if

- (a) the row in the DEPARTEMENT table with primary key DptNumber set to 3 is deleted?
- (b) the row in the DEPARTEMENT table with primary key DptNumber set to 3 as its value for DptNumber updated to 5?

Question 12

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

Question 13

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

Question 14

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

Question 15

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 help you to become more familiar with SQL. It constitutes the practical part of this class and shouldn't be overlooked. 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 four 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 7th February), so don't wait and let me know if you had difficulties doing them.

Problem 1

This exercise assume you successfully completed Problem 2 of Homework 1.

1. 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`. If you are prompted with a message

```
ERROR 1045 (28000): Access denied for user 'testuser'@'localhost' (using
↵ password: YES)
```

then you probably typed the wrong password. Otherwise, you should see a welcoming message from MySQL / MariaDB and a prompt.

2. Create a new database called `HW_2Q1`

```
CREATE DATABASE HW_2Q1;
```

3. List the databases

```
SHOW DATABASES;
```

Make sure `HW_2Q1` is a part of it.

4. Remove (“drop”) the database, using

```
DROP DATABASE HW_2Q1;
```

In the future, we will refer to the commands 1. and 2. as “log-in as `testuser` and create a database `HW_2Q1`”.

Problem 2

The goal of this problem is to learn where to find the documentation of your DBMS, and to have a first look at the syntax of SQL commands.

You can consult your textbook, Table 5.2, p. 140 (6th edition) or Table 7.2, p. 235 (7th edition), for a very quick summary of the most common commands. Make sure you are familiar with the Backus–Naur form (BNF) notation commonly used:

- non-terminal symbols (i.e., variables, parameters) are enclosed in angled brackets, `<...>`
- optional parts are shown in square brackets, `[...]`
- repetitions are shown in braces `{...}`
- alternatives are shown in parenthesis and separated by vertical bars, `(... | ... | ...)`

The most complete lists of commands are probably at

- <https://mariadb.com/kb/en/library/sql-statements/> and
- <https://dev.mysql.com/doc/refman/5.7/en/sql-syntax.html>

Those are the commands implemented in the DBMS you are actually using. Since there are small variations from one implementation to the other, it's better to take one of this link as a reference in the future.

Problem 3

This exercise, and the following ones, assume you successfully completed Problem 1.

Log in as `testuser` and create a database `HW_2Q3`.

- (a) We will first set-up our meta-data (i.e., the schema, the structure where our tables will be, and the tables themselves).

Let us first tell MySQL that we want to use that database

```
USE HW_2Q3;
```

and ask what it contains

```
SHOW TABLES;
```

Now, create a first table named `NAME`

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

Note that the SQL syntax and your DBMS are completely fine with your statement spreading over multiple lines. Let us create a second table, named `ADDRESS`

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

To make sure those two tables were actually created, we can use

```
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* (not the whole database, as we did in Problem 1) 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 FOREIGN KEY (Habitants)
    REFERENCES NAME(Id);
```

And observe the modification:

```
DESC ADDRESS;
```

- (b) This second part is about data, i.e., filling our tables with actual information. We begin by adding some data in the NAME table:

```
INSERT INTO NAME VALUES ('Barbara', 'Liskov', 003);
INSERT INTO NAME VALUES ('Tuong Lu', 'Kim', 004);
INSERT INTO NAME VALUES ('Samantha', NULL, 080);
```

To display the data we just inserted, we can use:

```
SELECT * FROM NAME;
```

Do you notice anything regarding the values we entered for the Id attribute?

We can add some data into 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;
```

- (c) Now, do the following:

1. Have a look at the formal definition of the **ALTER** command, at <https://dev.mysql.com/doc/refman/5.7/en/alter-table.html> or <https://mariadb.com/kb/en/library/alter-table/>.
2. Draw the relations corresponding to that database, including the domains and primary, as well as foreign, keys.
3. Write a **SELECT** statement that returns the Id number of the person whose first name is "Samantha".
4. Write a statement that violate the entity integrity constraint. What is the error message returned?
5. Execute an **UPDATE** statement that violate the referential integrity constraint. What is the error message returned?
6. Write a statement that violate another kind of constraint. Explain what constraint you are violating, and explain the error message.

Problem 4

Log in as `testuser` and create a database `HW_2Q4`. 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?