

Please read Sections 5.1 – 5.1.3, 5.1.7 and 5.4 of the textbook and then answer the following, trying not to look at your notes or at the textbook. Quiz #3, on Wednesday 4th October, will consist exclusively of questions taken from the Part 1 of this homework.

Both Part I and Part II suppose that you have access to the snippet of code hosted at cs.appstate.edu/~aubertc/db/code/2017_09_20_Queries.sql

Part I — Short Questions

Question 1

Describe what the star do in the statement `SELECT ALL * FROM myTable;`.

Question 2

What is wrong with the statement `SELECT * WHERE name = 'CS' FROM department;`?

Question 3

What is a select-project-join query?

Question 4

What is a tuple variable, and when is it useful?

Question 5

Write a query that changes the name of the professor whose login is 'caubert' to 'Hugo Pernot' in the table `prof`.

Question 6

Can an UPDATE statement have a WHERE condition using an attribute that isn't the primary key? If no, justify, if yes, tell what could happen.

Question 7

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

Question 8

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?

Question 9

What are the possible meanings or interpretations for a NULL value?

Question 10

What are the values of the following expressions (i.e., do they evaluate to TRUE, FALSE, or UNKNOWN)?

TRUE AND FALSE TRUE AND UNKNOWN NOT UNKNOWN FALSE OR UNKNOWN

Question 11

What comparison expression should you use to test if a value is different from NULL?

Question 12

Explain this query:

```
SELECT login
FROM prof
WHERE department IN ( SELECT major
                      FROM student
                      WHERE login = 'jrakesh');
```

Question 13

What is wrong with this query?

```
SELECT name
FROM student
WHERE login IN ( SELECT code
                FROM department
                WHERE head = 'aturing');
```

Question 14

Write a query that returns the number of row (i.e., of entries, of tuples) in a table named `test`.

Question 15

Write a query that returns the sum of all the values stored for the `pages` attribute in a `book` table.

Question 16

Write a query that adds a `pages` attribute of type `INT` into a (already existing) `book` table.

Question 17

Write a query that removes the default value for a `pages` attribute in a `book` table.



Part II — Problem

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, the problems requires a computer and to have access to the code used during the lecture. I'll assume that you will have successfully completed it by the time Homework #4 is released (Wednesday 4th October), so don't wait and let me know if you had difficulties doing it.

Problem 1

Log-in as `testuser` and create a database `HW3Q1`. Create the `prof`, `student`, `department` and `grade` tables as during the lecture. Populate them with some data.

- (a) Create a `lecture` table that contains a name, a code, a year and the login of a professor. Pick the datatypes wisely (use `YEAR(4)` for the year), and make the year and code attributes be the primary key (yes, have *two* attributes be the primary key).
- (b) Alter the `lecture` table: make the login of the professor be a foreign key referencing the `login` attribute in `prof`. What seems like the best reference option? Justify your answer.
- (c) Populate the `lecture` table with some made-up data.
- (d) Add two columns to the `grade` table, using

```
ALTER TABLE grade ADD COLUMN lecturecode CHAR(5),  
                  ADD COLUMN lectureyear YEAR(4);
```

if you decided to use respectively `CHAR(5)` and `YEAR(4)` for the datatypes of the code and year attributes of the course table.

- (e) Use `DESCRIBE` and `SELECT` to observe the schema of the `grade` table and its rows (i.e., tuples). Is that what you would have expected?
- (f) Add a foreign key in `grade`, using

```
ALTER TABLE grade ADD FOREIGN KEY (lectureyear, lecturecode)  
REFERENCES lecture(year, code);
```

Note that if the type of `lectureyear` (resp. `lecturecode`) isn't the same as the type of `year` (resp. `code`), you will have an error.

- (g) Update the tuples in `grade` with some made-up data: now, every row should contain, on top of a login and a grade, a lecture year and a lecture code.
- (h) Write `SELECT` statements answering the following questions:
 - i. "I taught class XXX in 20YY, could you give me the logins and grades of the students who took it?"
 - ii. "Could you list the instructors who taught in 20YY?" (and, please, avoid duplicates)
 - iii. "Could your list the name, grade, class and year of the students who obtained a grade of 3 or better?"
 - iv. "Could you tell me which years was the class XXX taught?"
 - v. "Could you print the name of the students who registered after Ava Alyx?"
 - vi. "Is there any department's head teaching this year?"