

Assignment #2

Instructor: Ahmed El-Roby*Name:* Ryan Lo, *ID:* 101117765

Instructions: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- The accepted formats for your submission are: pdf, txt, and java. More details below.
- You can either write your solutions in the tex file (then build to pdf) or by writing your solution by hand or using your preferred editor (then convert to pdf). However, you are encouraged to write your solutions in the tex file. If you decide not to write your answer in tex, it is your responsibility to make sure you write your name and ID on the submission file.
- If you use the tex file, make sure you edit line 28 to add your name and ID. Only write your solution and do not change anything else in the tex file. If you do, you will be penalized.
- All questions in this assignment use the university schema discussed in class (available on Brightspace under Resources → University_Toy_Database), unless otherwise stated.
- For SQL questions, upload a text file with your queries in the format shown in the file “template.txt” uploaded on culearn. An example submission is in the file “sample.txt”. You will be penalized if the format is incorrect or there is no text file submission.
- For programming questions, upload your .java file.
- Late submissions are allowed for 24 hours after the deadline above with a penalty of 10% of the total grade of the assignment. Submissions after more than 24 are not allowed.

Q 1:

(3 points)

Consider the following DDL statements:

```
create table takes
  (ID          varchar(5),
   course_id   varchar(8),
   sec_id      varchar(8),
   semester    varchar(6),
   year        numeric(4,0),
   grade       varchar(2),
   primary key (ID, course_id, sec_id, semester, year),
   foreign key (course_id, sec_id, semester, year) references section
     on delete cascade,
   foreign key (ID) references student
     on delete cascade
);

create table section
  (course_id   varchar(8),
   sec_id      varchar(8),
   semester    varchar(6)
     check (semester in ('Fall', 'Winter', 'Spring', 'Summer')),
   year        numeric(4,0)
     check (year > 1701 and year < 2100),
   building    varchar(15),
   room_number varchar(7),
```

```

time_slot_id      varchar(4),
primary key (course_id, sec_id, semester, year),
foreign key (course_id) references course
    on delete cascade,
foreign key (building, room_number) references classroom
    on delete set null
);

```

Now, consider the following SQL query:

```

select course_id, semester, year, sec_id, avg (tot_cred)
from takes natural join student
where year = 2017
group by course_id, semester, year, sec_id
having count (ID) >= 2;

```

Will appending **natural join** *section* in the **from** clause change the returned result? Explain why?

Appending natural join section in the from clause will not change the returned result because all the tuples that are being select are apparent in both the queried table and the section table. It also doesn't concatenate like the cartesian product. It does a join on the tuples that are in both the tables and since they both have the same tuples that are being selected, it won't change the result.

Q 2:

(2 points)

Write an SQL query using the university schema to find the names of each instructor who has never taught a course at the university. Do this using no subqueries and no set operations.

Q 3:

(2 points)

Rewrite the following query to replace the natural join with an inner join with **using** condition:

```

select *
from section natural join classroom;

```

Q 4:

(2 points)

Consider the following relation definition:

```

create table manager(
emp_id      char(20),
manager_id  char(20),
primary key emp_id,
foreign key (manager_id) references manager(emp_id) on delete cascade)

```

The foreign key constraint means that every manager has to be an employee. Explain what is going to happen when a manager is deleted.

With a foreign key that on delete cascades, everytime a manager is deleted, all other records with that manager_id or emp_id in it will also be deleted. So, this manager table acts moreso like a pair. Since the manager_id references the emp_id, any deletion of manager_id and emp_id would cause other managers with the same manager_id or emp_id to be deleted also. Say for example emp_id 1 has manager_id 4 but emp_id 2 also has manager_id 4, the deletion of either emp_id 1 or emp_id 2 or manager_id 4 will cause all emp_id 1, emp_id 2, manager_id 4 and every other emp_id or manager_id with those same values to also be deleted.

Q 5:

(5 points)

Using the university schema, define a view *tot_credits* (*year*, *num_credits*), giving the total number of credits taken in each year. Then, explain why insertions would not be possible into this view.

Insertions would not be possible into this view because of the sum aggregation as well as the group by aggregation. Any type of aggregation does not with with view and group by because it's not able to update

it since it already calculated the sum. When trying to perform an insertion into this view, "Views containing GROUP BY are not automatically updatable".

Q 6:

(10 points)

Write a Java program that finds all prerequisites for a given course using JDBC. The program should:

- Takes a course id value as input using keyboard.
- Finds the prerequisites of this course through a SQL query.
- For each course returned, repeats the previous step until no new prerequisites can be found.
- Prints the results.

Don't forget to handle the case for cyclic prerequisites. For example, if course A is prerequisite to course B, course B is prerequisite to course C, and course C is prerequisite to course A, do not infinite loop.

Q 7:

(5 points)

Consider the following schema:

employee(emp_name, street, city)

works(emp_name, company_name, salary)

Write a function *avg_sal* that takes a company name as input and finds the average salary of employees in the company. Then, write a SQL query that uses this function to find companies whose employees earn (on average) higher salary than the company "Losers Inc."