CS411: Database Systems **Spring 2017**

Assignment 3

Due Monday, March 27 at 11:59pm

General Instructions

- Feel free to talk to other members of the class in doing the homework. You should, however, write down your solutions yourself. *List the names of everyone you worked with at the top of your submission.*
- Keep your solutions brief and clear.
- Please use Piazza if you have questions about the homework but do not post answers. Feel free to use private posts or come to the office hours.

Homework Submission

- We DO NOT accept late homework submissions.
- We will be using Compass for collecting the homework assignments. Please submit your answers via Compass. Hard copies are not accepted.
- Contact the TAs if you are having technical difficulties in submitting the assignment; attempt to submit well in advance of the due date/time.
- The homework must be submitted in **pdf** format. Scanned handwritten and/or hand-drawn pictures in your documents won't be accepted.
- Please do not zip the answer document (PDF) so that the graders can read it directly on Compass. You need to submit one answer document, named as **hw3_netid.pdf**.
- Please see the assignments page for more details. In particular, we will be announcing errata, if any, on this page.

1 SQL queries involving a single relation

Consider a relation with just one attribute, $R(A)$. Express the following queries using SQL. Your queries will be graded based on their simplicity, and correctness.

1. [10] Write a query to compute the largest value of A in R. You are not allowed to use the `Max` operator. For this question, you may assume that there are no duplicates in R.
2. [10] Write a query to compute the median value of A in R. For this question, you may assume that there are no duplicates in R, and that R has odd number of tuples.
3. [10] Write a query to compute the mode of A in R. If multiple distinct values of A are candidates, then all of them should be returned. For this question, R may contain duplicates tuples; please don't assume otherwise.

2 SQL queries involving multiple relations

Consider the following relations:

```
Courses(cid, cname, room, pid)
Professors(pid, pname)
Students(sid, sname)
Enrolled(sid,cid)
```

Express the following queries using SQL. Your queries will be graded based on their simplicity, and correctness.

1. [5] Find the names of students who are not enrolled in any class.
2. [10] Find the names of students who are enrolled in the highest number of classes, i.e., all other students (not outputted) have taken fewer classes.
3. [10] Find the names of professors who teach in every room in which some course is taught.

3 Database Manipulation and Views

Consider the relations `Courses, Professors, Students, Enrolled` as described in question 2.

1. [5] Insert the following new tuple in `Students`: John with `sid` as 22222.
2. [5] Remove all professors from `Professors`, who are not teaching any course.
3. [5] Update the name of the student with `sid` as 22222, to Doe.

4. [10] Create a materialized view BusyProfessors, containing the names of all professors for whom the number of students they teach across their courses is greater than hundred.

   Note: if a student is enrolled in two different courses of the same professor, we count the student only once.

4 Constraints and Triggers

Consider the relations Courses, Professors, Students, Enrolled as described in question 2.

1. [10] Create the tables for Courses, Professors, Students, Enrolled using the SQL CREATE statement, with the following constraints:

   - For Enrolled, all the student and course ids should be present in the corresponding tables
   - For Courses, the professor id must be present in the corresponding table

2. [5] A deletion or update to a sid should be handled appropriately in Enrolled, due to the constraint as explained in the previous question.

   (a) Please rewrite the SQL CREATE statement such that when an sid gets updated/deleted in Students, its value is propagated accordingly to Enrolled.

   (b) Also, explain why we can not set Enrolled.sid to NULL when we delete that sid from Students.

3. [5] Create a trigger so that an insertion into Courses creates an tuple in Professors if the pid corresponding to the insertion doesn't exist in Professors.

5 Closure of Functional Dependencies

Consider a relation $R(A_1, A_2, \dots, A_n)$, with $n \geq 3$. R satisfies the following functional dependency: $A_1 \rightarrow A_2$.

1. [0] How many logically implied functional dependencies are of the form, $P \rightarrow A_2 A_n$, where $P \subseteq \{A_1, A_2, \dots, A_n\}$.[1]

Note: attempt this problem at your leisure, do not submit solutions.

[1] based on a question asked on Piazza.