Homework 1

School of Information Science and Technology

Zhang Jun 12903136

March 15, 2019

1 Single-Relation Queries (36 pts)

1. Consider the following relation:
   Graph(n1 , n2)
   A tuple (n1, n2) in Graph stores a directed edge from a node n1 to a node n2 in the corresponding graph. Your goal is to, for *every* node in the graph, count the number of outgoing edges of that node. Note that for nodes without any outgoing edges, their edge count would be zero; you need to output this as well. You can assume that:

   a) there are no duplicates or null values in the table; and

   b) every node in the graph is involved in at least one edge.

   Answer:

   ```
   select n1, count(n1)
   from Graph
   group by n1
   union
   select distinct n2, 0
   from Graph
   where n2 not in (
   select n1
   from Graph
   )
   ```

2. Consider the following relation:
   Trained(<u>student</u>, master, year)

A tuple (S, M, Y) in Trained specifies that a SQL Master M trained student S who graduated in year Y. Your goal is to find *the count of* SQL Masters who trained a student who graduated in the same year that ’Alice’ or ’Bob’ graduated.
Answer:

```
SELECT COUNT (*)
FROM (
SELECT DISTINCT master
FROM Trained
WHERE year IN (
SELECT DISTINCT year
FROM Trained
WHERE student = ’Alice’ OR student = ’Bob’) )
```

3. Consider the following relation:
DBMS(operator, system, performance)
A tuple (O, S, P) in DBMS specifies an operator O in system S and has the performance value P. Your goal is to find those systems whose operators achieves a higher performance value on average than the average performance value in a system named ’PostgreSQL’.

Answer:

```
SELECT system
FROM (SELECT system, AVG(performance) AS avg_per
FROM DBMS
GROUP BY system) AS new_DBMS
WHERE avg_per>(select AVG(performance)
FROM DBMS
WHERE system = ’PostgreSQL’)
```

2 MULTI-RELATION QUERIES (40 PTS)

Consider the following relations representing student information at UIUC:
Mentorship(mentee_sid, mentor_sid)
Study(sid, credits)
Enrollment(did, sid)
Student(sid, street , city)

- A tuple (M1, M2) in Mentorship specifies that M2 is a mentor of another student M1.
- A tuple (S, C) in Study specifies that the student S has taken C credits.
- A tuple in Enrollment (D, S) specifies that student S is enrolled in department D.
- A (ST, S, C) in Student specifies that student ST lives on street S in city C.

1. Find all students who live in the same city and on the same street as their mentor.

   Answer:

   ```
   SELCET DISTINCT s1.sid
   FROM Student s1, Student s2, Mentorship m
   WHERE s1.city = s2.city
   AND s1.street = s2.street
   AND s1.sid = m.mentee_sid
   AND s2.sid = m.mentor_sid
   ```

2. Find all students(i.e., distinct sid) who have taken more credits than the average credits of all of the students of their department.

   Answer:

   ```
   SELECT DISTINCT sid
   FROM Study s, Enrollment e
   WHERE E.sid = S.sid AND S.credits > (SELECT AVG(credits)
   FROM Study s1, Enrollment e1
   WHERE s1.sid = e1.sid AND
   s.did = s1.did)
   ```

3 DATABASE MANIPULATION AND VIEWS (24 PTS)

1. In the Study relation, insert a new student, whose id is 66666 and has 0 credits.
   Answer:

   ```
   INSERT INTO Student VALUES (66666,NULL,NULL)
   INSERT INTO Study VALUES (66666,0)
   ```

2. In the Study relation, delete students who have graduated (i.e., the ones who have more than 200 credits).
   Answer:

   ```
   DELETE
   FROM Study
   WHERE credits > 200
   ```

3. In the Study relation, add 2 credits for students who are mentors.

   Answer:

   ```
   UPDATE Study SET credits = credits + 2
   WHERE sid in (
   SELECT DISTINCT mentor_sid
   FROM Mentorship )
   ```

4. Incoming students are those who have been accepted (i.e., exist in the **Student** relation) but have not registered in any department (i.e., do not exist in the **Enrollment** relation). Create a View that contains **sid** of all incoming students.

   Answer:

   ```
   CREATE VIEW incoming AS
   SELECT Student.sid
   FROM Student
   WHERE Stuendt.sid not in (
   SELECT Enrollment.sid
   FROM Enrollment)
   ```