CSCC43: Introduction to Databases
Assignment 2

Out: June 14, 2005
Due: July 5, at the beginning of the lecture

This assignment must by typed. Handwritten assignments will **not be remarked**. Pages must be numbered. Make sure there are no pages missing.

Exercise 1

Consider the following schema for a database recording information about a research lab. The research lab, composed of several research units, employs researchers, interns, lab and administrative staff who work on various projects. A researcher may supervise students and be involved in several research projects. Every intern is supervised by a researcher and may work on several research projects. All research units and projects are managed by a researcher.

$Researcher(\underline{r\_number}, r\_name, category, r\_salary, r\_area, ru\_code)$

$Intern(\underline{i\_number}, i\_name, i\_start\_date, duration, salary, r\_number)$

$Lab\_staff(\underline{s\_number}, l\_name, l\_start\_date, salary, clearance, area)$

$Administrative\_staff(\underline{s\_number}, a\_name, a\_start\_date, salary, description, experience, ru\_code)$

$Project(\underline{p\_code}, start\_date, budget, duration, r\_number, ru\_code)$

$Research\_unit(\underline{ru\_code}, security\_level)$

$Intern\_project(\underline{i\_number, p\_code})$

$Researcher\_project(\underline{r\_number, p\_code})$

$Staff\_project(\underline{s\_number, p\_code})$

$Staff\_lab(\underline{s\_number, ru\_code})$

$Manages(\underline{r\_number, ru\_code})$

The following (inclusion dependencies) hold:

$\pi_{i\_number}(Intern\_project) \subset \pi_{i\_number}(Intern)$

$\pi_{p\_code}(Intern\_project) \subset \pi_{p\_code}(Project)$

$\pi_{r\_number}(Researcher\_project) \subset \pi_{r\_number}(Researcher)$

$\pi_{p\_code}(Researcher\_project) \subset \pi_{p\_code}(Project)$

$\pi_{s\_number}(Staff\_project) \subset \pi_{s\_number}(Lab\_staff)$

$\pi_{p\_code}(Staff\_project) \subset \pi_{p\_code}(Project)$

$\pi_{r\_number}(Manages) \subset \pi_{r\_number}(Researcher)$

$\pi_{ru\_code}(Manages) \subset \pi_{ru\_code}(Research\_unit)$

$\pi_{p\_code, r_number}(Project) \subset \pi_{p\_code, r\_number}(Researcher\_project)$

Part 1 - SQL (100 marks)

1. (10 marks) List for each researcher her/his *r_number* and the average budget of the projects s/he manages. For researchers who did not manage any project list 0 as the average.

```
(SELECT r_number, AVG(budget) AS average_budget
```

```
 FROM Project P
 GROUP BY  r_number)

UNION

(SELECT r_number, 0 AS average_budget
FROM Researcher R
WHERE r.r_number NOT IN ( SELECT r_number
                          FROM   Project))
```

2. (10 marks) List the research units that have the highest ratio of administrative staff to researchers.

```
SELECT T5.ru_code
FROM   (SELECT max(ratio) as maxim
        FROM   (SELECT ru_code, nr_of_staff/nr_of_researchers AS ratio
                FROM   (SELECT   ru_code, COUNT(*) AS nr_of_staff
                        FROM     Administrative_staff
                        GROUP BY ru_code) T1,
                       (SELECT   ru_code, COUNT(*) AS nr_of_researchers
                        FROM     Researcher
                        GROUP BY ru_code) T2,
                WHERE  t1.ru_code=T2.ru_code AND
                       t2_nr_of_researchers > 0) T3) T4,
       (SELECT ru_code, nr_of_staff/nr_of_researchers AS ratio
        FROM   (SELECT   ru_code, COUNT(*) AS nr_of_staff
                FROM     Administrative_staff
                GROUP BY ru_code) T1,
               (SELECT   ru_code, COUNT(*) AS nr_of_researchers
                FROM     Researcher
                GROUP BY ru_code) T2,
        WHERE  t1.ru_code=T2.ru_code AND t2_nr_of_researchers > 0) T5
WHERE T5.ratio = T4.maxim
```

3. (10 marks) List the researchers who manage the projects which involve the lowest number of people(researchers, interns and staff).

```
SELECT R.r_number
FROM   (SELECT MIN(count) as minim
        FROM   (SELECT p_code, COUNT(*) as count
                FROM   ((SELECT p_code
                          FROM   Intern_project)
                        UNION
                        (SELECT p_code
                          FROM   Researcher_project)
                        UNION
                        (SELECT p_code
                          FROM   Staff_project)
                        UNION
                        (SELECT p_code
                          FROM   Project)) T1
                GROUP BY p_code) T2) T3,
       (SELECT p_code, COUNT(*) as count
        FROM   ((SELECT p_code
```

```
                 FROM   Intern_project)
                UNION
                (SELECT p_code
                 FROM   Researcher_project)
                UNION
                (SELECT p_code
                 FROM   Staff_project)
                UNION
                (SELECT p_code
                 FROM   Project)) T1
        GROUP BY p_code) T4, Researcher R, Project P
WHERE T4.count = T3.minim AND P.p_code = T4.p_code AND P.r_number = R.r_number
```

4. (10 marks) Find the names of the researchers who coordinate only projects that are not associated with their research unit.

```
SELECT  r_name
FROM    Project P, Researcher R
WHERE   NOT EXISTS (SELECT P1.R_NUMBER
                    FROM   Project P1, Researcher R1
                    WHERE  P1.r_numebr = R1.r_number AND
                           P1.ru_code <> R1.ru_code)
        AND P.r_number = R.r_number
```

5. (10 marks) List the codes of projects whose lab staff members do not work on any other projects.

```
SELECT p_code
FROM   Project P
WHERE  NOT EXISTS (SELECT *
                   FROM   Project P1, staff_project S1, staff_project S2, Project P2
                   WHERE  P1.p_code <> P2.p_code AND
                          P1.p_code = P.p_code AND
                          S1.s_number = S2.s_number AND
                          P1.p_code = S1.p_code AND
                          S2.p_code = P2.p_code AND
                          S1.p_code = P1.p_code AND
                          S1.s_number = S2.s_number)
```

6. (10 marks) Find the interns who have worked on all the projects they have been involved in for the entire duration of the project. For projects not finished yet, an intern listed as working on a project is assumed to be working on that project until the end of the project or the end of the intership, whichever comes first.

```
SELECT i_number
FROM   Intern_project I
WHERE  NOT EXISTS (SELECT *
                   FROM   Intern I1, Intern_project IP, Project P
                   WHERE  I1.i_number = IP.i_number AND
                          IP.p_code = P.p_code AND
                          I.i_number = I1.i_number AND
                          ((days(P.start_date)- days(I.startdate) > 0) OR
                          (days(P.start_date) + P.duration -
                           days(I.start_date) - I.duration > 0))
```

7. (10 marks) Find the researchers who only hired interns they supervise to work for projects they coordinate.

```
SELECT P1.r_number
FROM   Project P1
WHERE  NOT EXISTS (SELECT *
                   FROM   Intern I, Project P, Intern_project IP
                   WHERE  IP.i_number = I.i_number AND
                          IP.p_code = P.p_code AND
                          P.r_number = P1.r_number AND
                          I.r_number <> P1.r_number)
```

8. (10 marks) List the projects coordinated by researchers who have interns working for all the projects they are involved in.

```
SELECT P.p_code
FROM   Project P
WHERE  NOT EXISTS ((SELECT RP.p_code
                    FROM   Researcher_project RP
                    WHERE  P2.r_number = P.r_number)
                   EXCEPT
                   (SELECT P1.p_code
                    FROM   Researcher_project P1, Intern_project IP
                    WHERE  P1.r_number = P.r_number AND
                           P1.p_code = IP.p_code))
```

9. (10 marks) List the projects that involve only people from the research unit the project belongs to.

```
SELECT P.p_code
FROM   Project P
WHERE  NOT EXISTS ((SELECT RP.p_code
                    FROM   Researcher_project RP, Researcher R
                    WHERE  RP.p_code = P.p_code AND
                           R.r_number = RP.r_number AND
                           R.ru_code <> P.ru_code)
                   UNION
                   (SELECT IP.p_code
                    FROM   Intern_project IP, Resercher R, Intern I
                    WHERE  IP.p_code = P.p_code AND
                           I.i_number = IP.i_number AND
                           I.r_number = R.r_number
                           R.ru_code <> P.ru_code)
                  UNION
                   (SELECT SP.p_code
                    FROM   Staff_project SP, Staff_lab S
                    WHERE  SP.p_code = P.p_code AND
                           S.s_number = SP.s_number AND
                           S.ru_code <> P.ru_code))
```

10. (10 marks) Find the interns who are/where involved in most projects.

```
SELECT T3.i_number
FROM  (SELECT MAX(nr_of_projects) as maxim
       FROM   (SELECT   i_number, COUNT(*) AS nr_of_projects
               FROM     Intern_project
```

```
                GROUP BY i_number) T1) T2,
        (SELECT   i_number, COUNT(*) AS nr_of_projects
         FROM     Intern_project
         GROUP BY i_number) T3
WHERE T3.nr_of_projects = T2.maximum
```

Part 2 - SQL (50 marks)

1. (10 marks) Write an assertion that checks if an intern works on more than 5 projects.

```
CREATE ASSERTION a1 CHECK
(EXISTS (SELECT   *
         FROM     Intern_project
         GROUP BY i_number
         HAVING COUNT(*) > 5))
```

2. (10 marks) Write an assertion that checks whether there are projects whose budget is less than the salaries of the people(researchers, interns and staff) who work only on that project.

```
CREATE ASSERTION a2 CHECK
(EXISTS (SELECT p_code
         FROM    (SELECT p_code, salary
                   FROM   Intern_project IP, Intern I
                   WHERE  IP.i_number = I.i_number)
                  UNION
                  (SELECT p_code, salary
                   FROM   Researcher_project RP, Researcher R
                   WHERE  RP.r_number = R.r_number)
                  UNION
                  (SELECT p_code, salary
                   FROM   Staff_project SP, Staff_lab S
                   WHERE  SP.s_number = S.s_number) T,
                  Project P
         WHERE    P.p_code = T.p_code
         GROUP BY P.p_code, P.budget
         HAVING   budget < SUM(salary))
```

3. (10 marks) Write a "BEFORE UPDATE" trigger that raises with 5% the salary of the lab staff who becomes involved in at least 3 projects.

```
CREATE TRIGGER t1
BEFORE UPDATE OF s_number ON Staff_project
REFERENCING NEW ROW AS new
FOR EACH ROW
WHEN (new.s_number IN (SELECT   s_number
                       FROM     Lab_staff L, Staff_project S
                       WHERE    L.s_number = S.s_number AND
                                L.s_number = new.s_number
                       GROUP BY s_number
                       HAVING COUNT(*) >= 2)
UPDATE Lab_staff
SET    salary = 1.05 * (SELECT salary
                        FROM   Lab_staff L
                        WHERE  L.s_number = new.s_number)
WHERE  s_number = new.s_number
```

4. (10 marks) Write an "AFTER DELETE" trigger that decreases by 5% the salaries of the interns supervised by researchers who no longer manage at least 2 projects.

```
CREATE TRIGGER t2
AFTER DELETE ON Project
REFERENCING OLD ROW AS old
FOR EACH statement
WHEN (2 > (SELECT   COUNT(*)
           FROM     Project P
           WHERE    P.r_number = old.r_number)
UPDATE Intern
SET    salary = 0.95 * salry
WHERE  r_number = old.r_number
```

5. (10 marks) Write an "AFTER INSERT" trigger that checks if the salary of a new intern is lower or greater by 5% than the average intern salary in her/his research unit(an intern's research unit is considered to be that of her/his supervisor) and sets it to the average if the condition is not met.

```
CREATE TRIGGER t3
AFTER INSERT ON INTERN
REFERENCING NEW ROW AS new
FOR EACH statement
WHEN ((new.salary > 1.05 * (SELECT AVG(I.slary)
                           FROM  Intern I, Researcher R,
                                 (SELECT r_area
                                  FROM   Intern I1, Researcher R1
                                  WHERE  I1.i_number = new.i_number AND
                                         I1.r_number = R.r_number) T
                           WHERE I.i_number <> new.i_number AND
                                 I.r_number = R.r_number AND
                                 R.r_area = T.r_area))
        OR
       (new.salary < 0.95 * (SELECT AVG(I.slary)
                             FROM  Intern I, Researcher R,
                                  (SELECT r_area
                                   FROM   Intern I1, Researcher R1
                                   WHERE  I1.i_number = new.i_number AND
                                          I1.r_number = R.r_number) T
                             WHERE I.i_number <> new.i_number AND
                                   I.r_number = R.r_number AND
                                   R.r_area = T.r_area)))
UPDATE Intern
SET salary = (SELECT AVG(I.slary)
              FROM   Intern I, Researcher R,
                     (SELECT r_area
                      FROM   Intern I1, Researcher R1
                      WHERE  I1.i_number = new.i_number AND
                             I1.r_number = R.r_number) T
              WHERE I.i_number <> new.i_number AND
                    I.r_number = R.r_number AND
                    R.r_area = T.r_area)
WHERE i_number = new.i_number
```