

Final Examination

Due: In 120 minutes. 4 to 6pm
Course: CSci 5702 Fall 93

Instructions:

0. Provide YOUR NAME, IDENTIFICATION NUMBER on the cover.
1. There are 5 questions in this paper. Problem 1 has one part, problem 2 has two parts, problem 3 has two parts, problem 4 has three parts, and problem 5 has 1 part.
2. Please use the space provided (below the questions) to write the answers. You may use the back side of pages if needed.
3. Budget your time to answer various questions to avoid spending too much time on a particular question.
4. It is an open book examination. You may refer to textbook and your class notes. Please do not use other books.

NAME:

U of Mn Identification Number:

SCORE TABLE

| Question | Score |
|----------|-------|
| Q1 | |
| Q2a | |
| Q2b | |
| Q3a | |
| Q3b | |
| Q4a | |
| Q4b | |
| Q4c | |
| Q5 | |

Question 1. (SQL)

Below is a subset of relations from COMPANY schema (in Figure 6.6 of the textbook). The keys have been underlined.

```
EMPLOYEE(FNAME,MINIT,LNAME,SSN,BDATE,ADDRESS,SEX,SALARY,SUPERSSN,DNO)
WORKS_ON(ESSN,PNO,HOURS)
PROJECT(PNAME,PNUMBER,PLOCATION,DNUM)
```

Below is a set of database queries written in English.

- (a) Find the employees who works on some projects located in Stafford.
- (b) Find the employees who works on all projects located in Stafford.
- (c) Find the employees who work only on projects located in Stafford.
- (d) Find the employees who work on some but not all projects located in Stafford.
- (e) Find the employees who work on no project located in Stafford.
- (f) Find the employees who works on all projects located in Stafford, and only those projects located in Stafford.

For each of these queries, find the SQL statement(s) below that produces the corresponding result. (Hint: it is possible to have an English statement not matched to any SQL statement, and vice versa. Moreover, more than one SQL statement may match the same English query statement.)

- | | |
|--|--|
| (1) select E.FNAME,E.LNAME from EMPLOYEE E where not exists (select * from PROJECT P where P.plocation ≠ 'Stafford' and exists (select * from WORKS_ON W where W.ESSN=E.SSN and W.PNO=P.PNUMBER)) | (4) select E.FNAME,E.LNAME from EMPLOYEE E, WORKS_ON W, PROJECT P where E.ESSN=W.ESSN and W.PNO=P.PNUMBER and P.PLOCATION='Stafford' |
| (2) select E.FNAME,E.LNAME from EMPLOYEE E where not exists (select * from PROJECT P where P.PLOCATION='Stafford' and not exists (select * from WORKS_ON W where W.ESSN=E.ESSN and W.PNO=P.PNUMBER)) | (5) select E.FNAME,E.LNAME from EMPLOYEE E,WORKS_ON W1, PROJECT P1 where E.SSN=W1.ESSN and W1.PNO=P1.PNUMBER and exists (select * from PROJECT P2 where not exists (select * from WORKS_ON W2 where W2.PNO=P.PNUMBER and W2.ESSN=E.SSN)) |
| (3) select E.FNAME,E.LNAME from EMPLOYEE E where not exists (select * from PROJECT P, WORKS_ON W where P.LOCATION ≠ 'Stafford' and W.ESSN=E.SSN and W.PNO=P.PNUMBER) | (6) select E.FNAME,E.LNAME from EMPLOYEE E where not exists (select * from PROJECT P, WORKS_ON W where P.PLOCATION='Stafford' and W.ESSN=E.SSN and W.PNO=P.PNUMBER) |

ANSWER TABLE

| English Query Statement | Equivalent SQL Statement(s) |
|-------------------------|-----------------------------|
| a | |
| b | |
| c | |
| d | |
| e | |
| f | |

Question 2. (Postgres)

The schema used in this question is shown below:

EMP

| Name | <u>SSN</u> | Bdate | Address | Sex | Salary | SuperSSN | Dno |
|------|------------|-------|---------|-----|--------|----------|-----|
|------|------------|-------|---------|-----|--------|----------|-----|

a. Express the following integrity constraints as Postgres rules by specifying the event and qualification for each rule. Do not specify the POSTQUEL command. You can use SQL or POSTQUEL to specify the qualifications.

(i) Salary of an employee cannot exceed \$1000,000 unless he/she works for department 007.

(ii) Salary of an employee never decreases if his/her birthdate is before 1945.

(iii) Salary for a supervisor is always more than the supervisee. (Hint: Transitive closure is not relevant)

(iv) Salary for an employee cannot be retrieved if his/her salary is more than the average salary of all employees.

b. Represent the following recursive query in POSTQUEL using retrieve*.

Retrieve all supervisors (direct and indirect) for John B. Smith.

The answer should include 1st level, 2nd level and n th level managers for John, where n is not known beforehand.

Question 3. (Normal Form)

a. Consider the relation $R(A,B,C,D,E,F)$. (A,B) forms the primary key. The following functional dependencies exist among the attributes of the relation:

$$\begin{aligned}(A, B) &\rightarrow C, \\(A, B) &\rightarrow D, \\(A, B) &\rightarrow E, \\A &\rightarrow F, \\E &\rightarrow D,\end{aligned}$$

Transform this relation into a set of 1NF, 2NF, 3NF relations.

vspace4in

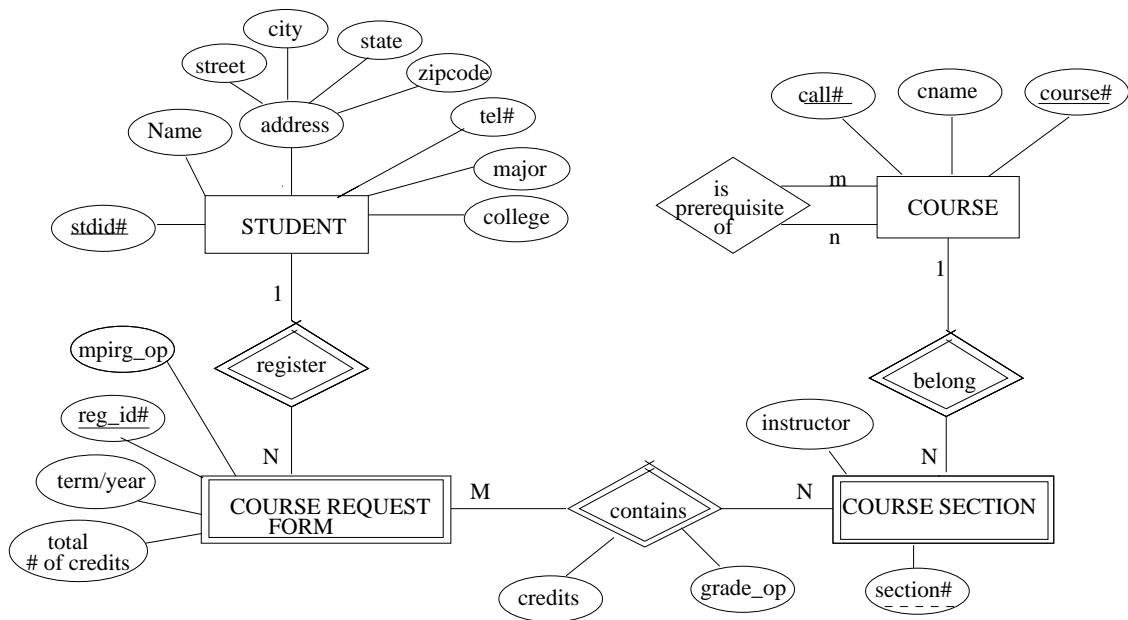
b. Determine the normal form for the relation below (student and grade information).

(Student_SSN, Student_Name, Sex, Age, Course_Number, Course_Name, Grade)

the candidate keys: (Student_SSN, Course_Number) and (Student_SSN, Course_Name)

Question 4. (Data Model)

Below is the ER diagram for Course Registration Database.



- Draw a network database schema in a diagrammatic notation such as Figure 12.4 page 335 of textbook.
- Draw a relational database schema in a diagrammatic notation such as Figure 12.2 page 329 of textbook.
- Specify any constraints that are represented in ER schema, but are not implicit in your relational schema.

Question 5. (Network Query)

In the course registration system, the many-to-many relationship COURSE-PREREQUISITE can be represented as a relation in the relational model or a linking record type in the network model as shown below:

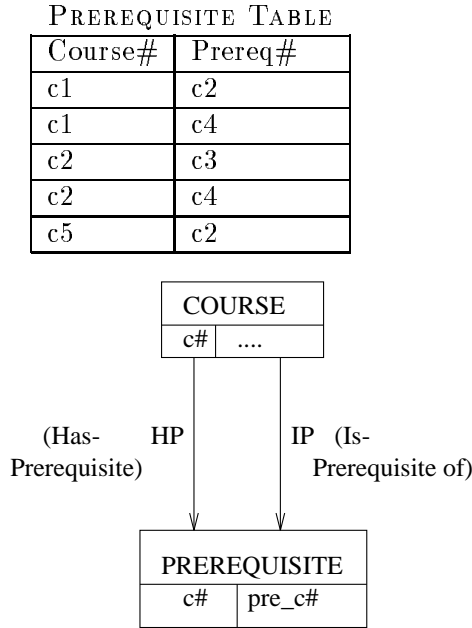


Figure 1:

(NB: In the PREREQUISITE relation, the first tuple means that c2 must be taken before c1. In other words c2 is_prerequisite_of c1, and c1 has_prerequisite c2.)

The record occurrences of Course and Prerequisite record-types for a database is shown in Figure 2. Each record has a uid, which is unique in the entire database. Fill in the COSET TABLE below such that:

- each row of this table represents an instance of a co-set.
- column coset-type can take values of HP(Has-Prerequisite) and IP(Is-Prerequisite-of).

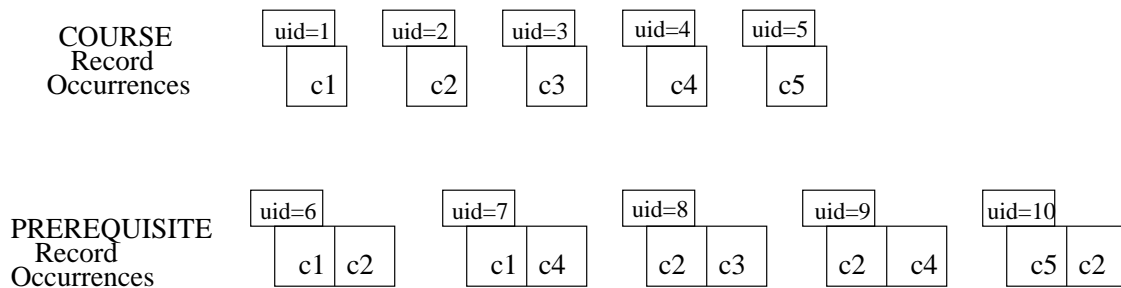


Figure 2:

COSET TABLE

| Coset-Type | Owner uid | Member uids |
|------------|-----------|-------------|
| | | |
| | | |
| | | |
| | | |
| | | |