COMP9120: Database Management Systems
Quiz
Semester 1 2025

Duration: 90 minutes **Marks: 18**

Student Name:

Student id:

MCQ Questions: 3 questions

Question 1: (1 mark)

Choose an answer which most accurately describes the relational algebra selection operator from the following choices:

a) It is equivalent to the FROM clause in the SQL SELECT statement.
b) It is equivalent to the SELECT clause in the SQL SELECT statement.
c) It is equivalent to the WHERE clause in the SQL SELECT statement.

Question 2: (1 mark)

A schedule of N transactions is *conflict serializable* *if and only if* its precedence graph (Circle the right answer)

a) contains a cycle involving all N transactions.
b) contains a cycle involving any number of transactions.
c) does not contain any cycle.

Question 3: (1 mark)

Circle the statement that is **false**:

a) A relation may have multiple candidate keys and multiple primary keys.
b) If attributes A and B together form the primary key for a relation R, then R may contain two or more rows with the same value for A.
c) A primary key attribute can also be a foreign key.

Essay Questions (6 questions)

Question 4: (2 marks)

Given the following schema:

Flood(floodID, floodYear, floodDetails)

Cities(cityID, cityName, totalPopulation)

FloodAffectedCities(cityID, floodID, totalRainfall, damageDetails)

(a) Write a Relational Algebra expression (RA) that finds the id of all cities which were affected by floods, both before 1995 and after 2020.

(b) Write a Relational Algebra expression (RA) that finds the name of all the cities which were never affected by floods.

Solution:

a.

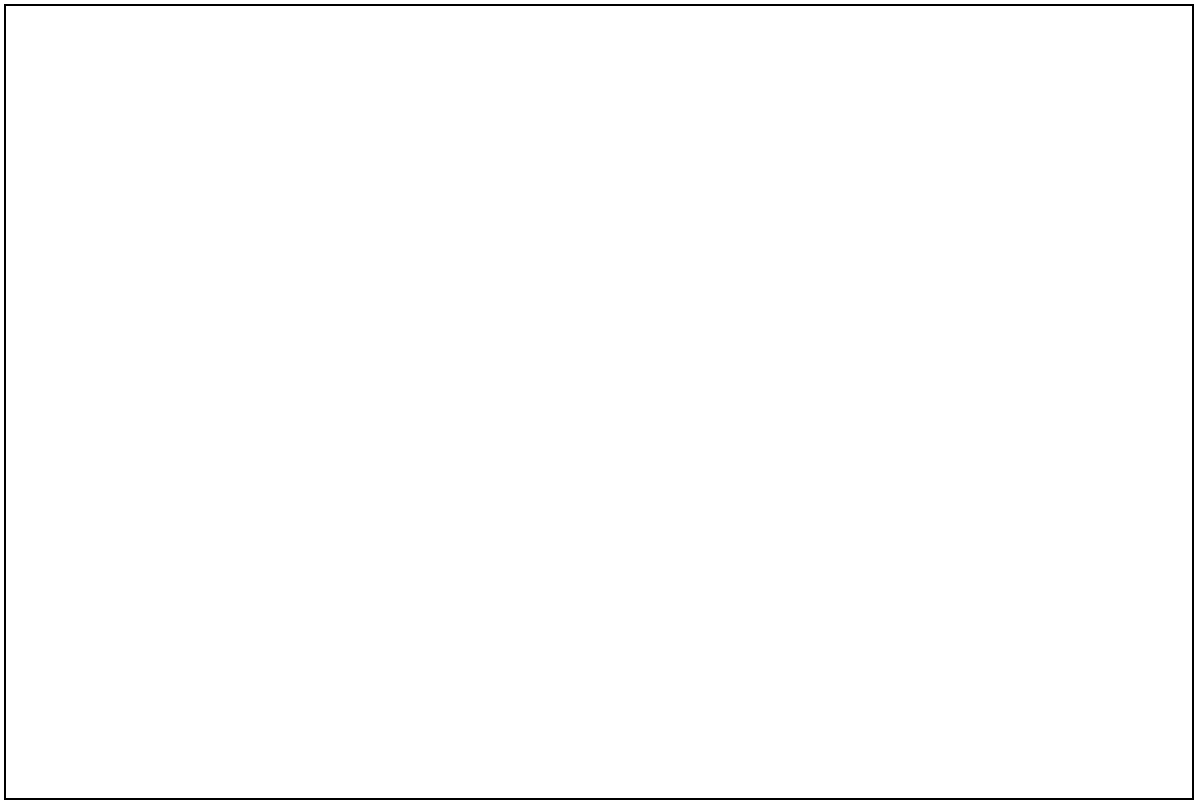
$$\pi_{cityID}(\sigma_{floodYear<\ 1995'}(Flood \bowtie FloodAffectedCities)$$

$\cap$

$$\pi_{cityID}(\sigma_{floodYear>\ 2020'}(Flood \bowtie FloodAffectedCities)$$

b.

$$\pi_{cityName}(Cities \bowtie (\pi_{cityID}(Cities) - \pi_{cityID}(FloodAffectedCities)))$$



Question 5: (2 marks)

Consider the database consisting of the following tables:

Employee(employee_id, employee_name, employee_city, manager_id)

Project(project_id, project_title, budget)

AssignedTo(project_id, employee_id)

Note that a manager must be an employee. Therefore, in employee table, manager_id *references* employee_id.

Write down the SQL expressions for the following queries:

(a) Find the name of the employees who live in the same city as their manager.

(b) Find the id of those projects that have the highest number of employees assigned to them.

Solution:

a. SELECT m.employee_name

FROM employee e, employee m

WHERE e.employee_id = m.manager_id and

e.employee_city = m.employee_city;

b.
```
SELECT project_id
    FROM AssignedTo
    GROUP BY project_id
    HAVING COUNT(employee_id) >=ALL (SELECT count(employee_id)
                                     FROM AssignedTo
                                     GROUP BY project_id);
```

Question 6: (2 marks)

Consider two relations Projects(ProjID, Location, Budget) and Employees(EmpID, Name, ProjID), where ProjID in *Employees* is a foreign key to *Projects*. Additionally, an employee can be assigned to *at most one* single project.

1. If we delete a project tuple in the table Projects:
   (i) What type of integrity constraint needs to be checked?
   (ii) What are the possible ways to enforce this type of constraint?

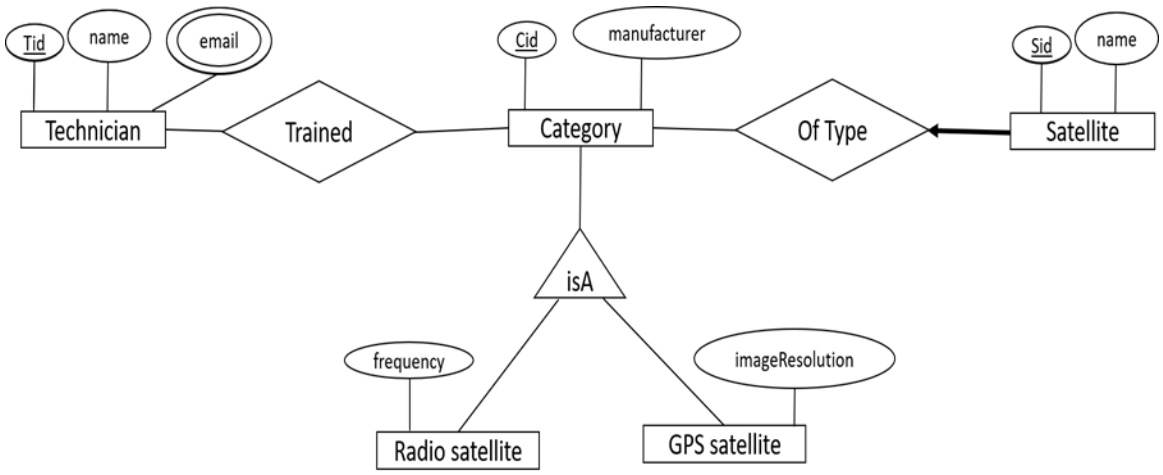
2. If we use a trigger to implement the integrity constraint ON DELETE CASCADE in the Employees table, state the type of TRIGGER clauses you will use, namely choose (i) BEFORE vs. AFTER, and (ii) ROW vs. STATEMENT. State the reasons for your choice. 0 marks for no explanation.

Solution:

1. (i) Yes, there is a referential integrity constraint from Employee that needs to be checked. (ii) to enforce this integrity, we have the following choices: it can be *rejected*, the *delete is cascaded*, or a *default value is set*.
2. We will use a STATEMENT as there may be more than one tuple in Employees that have a match to the deleted row in Projects. We will use the AFTER clause because the deletion of rows in Employee happens after the matching project row is deleted: it is about integrity maintenance.

Question 7: (4 marks)

This question is based on the following E-R diagram which describes the information kept on a satellite station.



a. Translate this E-R diagram into a relational model using the following **textual notation**.

Note: Each relation should be written in the form:

Name(attribute 1, attribute 2, …) PK=(attribute list), FK=(attribute list->parent relation),...

An 'attribute list' is one or more, comma-separated attribute names. Each relation must have a primary key (PK) defined. A relation can have zero or more FKs specified.

For example: Enrolment(studentId, courseId, mark) PK=(studentId, courseId),

FK=(studentId → Student, coursed → Course)

Do not write this relational model in SQL DDL syntax, use the above convention to describe your model.

b. Write SQL CREATE TABLE statements for the *Technician* and *Category* relation from your relational model, as well as any associated relationship relations between these two relations. You should show all relevant attributes, types and key constraints for these relations. You should include foreign key integrity constraints where suitable.

Solution:

a.

Technician(Tid, name)

PK=(Tid)

Email(Tid, email)

PK=(Tid, email)

FK=(Tid → Technician)

Category(cid, manufacturer)

PK=(cid)

RadioSatellite(cid, frequency)

PK=(cid)

FK=(cid → Category)

GPSSatellite(cid, imageResolution)

PK=(cid)

FK=(cid → Category)

Trained(cid, Tid)

PK=(cid, Tid)

FK=(cid → Category, Tid → Technician)

Satellite(sid, name, cid)

PK=(sid)

FK=(cid → Category)

b.

```
CREATE TABLE Technician( Tid INTEGER PRIMARY KEY,
                         name Varchar(50);

CREATE TABLE Email( Tid INTEGER,
                    email Varchar(100
                    PRIMARY KEY(Tid, email),
                    FOREIGN KEY(Tid) REFERENCES Technician);

CREATE TABLE Category( Cid INTEGER PRIMARY KEY,
                       manufacturer Varchar(50));

CREATE TABLE Trained( Tid INTEGER,
                      Cid INTEGER,
                      PRIMARY KEY(Tid, Cid),
                      FOREIGN KEY(Tid) REFERENCES Technician,
                      FOREIGN KEY(Cid) REFERENCES Category);
```

Question 8: (3 marks)

Consider a relation **R(EmployeeID, Name, Department, Manager, Salary, Position)** with the following functional dependencies:

EmployeeID → Name, Department, Position
Department → Manager
Position → Salary

1. Compute the closure of **{EmployeeID, Department}**, i.e., **{EmployeeID, Department}$^+$**.
2. Determine whether **{EmployeeID, Department}** is a candidate key. Provide an explanation.
3. Is the relation R in BCNF? If it is not, give a lossless-join and dependency preserving decomposition of the relation *R* into a set of BCNF-compliant relations.

Answer:

1. Compute the closure of {EmployeeID, Department}$^+$

We start with {EmployeeID, Department} and apply the functional dependencies to determine all attributes we can derive.

- EmployeeID → Name, Department, Position- Since we have EmployeeID, we can derive Name, Department, and Position. (We already have Department.)

- Department → Manager- Since we have Department, we can derive Manager.

- Position → Salary- Since we have Position, we can derive Salary.

{EmployeeID, Department}$^+$ = {EmployeeID, Name, Department, Position, Manager, Salary}

2. Is {EmployeeID, Department} a candidate key?

A candidate key is a minimal set of attributes that uniquely determines all attributes in the relation.

From 1. It is a **superkey** because it determines the relation R.

Now let us check whether EmployeeID or Department are keys themselves. Compute {EmployeeID}$^+$ and {Department}$^+$.

{EmployeeID}$^+$ = { EmployeeID, Name, Department, Position, Manager, Salary} which is a key and it is minimal.

{Department}$^+$ = {Department, Manager}

Therefore, {EmployeeID, Department} is **not** a candidate key because it is *not* minimal.

3. The candidate Key is Employee Id from 2. Above. The relation R is not in BCNF because of the violation caused by the FD Department → Manager

Decompose the relation R into relations : R1 = (Department, Manager) and R2 = (EmployeeID, Name, Department, Salary, Position), R1 is BCNF compliant. However, R2 is not because of the violation of FD Position → Salary**.** Decompose R2 into relations: R3 = (Position, Salary) and R4 = (EmployeeID, Name, Department, Position). Now R3 and R4 are BCNF compliant.

我的只满足 joint

不满足 Per servation

Question 9: (2 marks)

Given the following schedule S:

R1(a), R1(b), W1(c), R2(d), W2(a), W2(d), R3(c), R3(e), W4(f), W4(a)

Note that:

- **R** = Read
- **W** = Write
- The number is the **transaction number**
- The letter in parentheses is the **data item**

Explain whether this schedule is conflict serializable. If the schedule is conflict equivalent to a serial schedule, provide an equivalent serial order.

Solution

There are 4 conflicts in this case.

First conflict is between R1(a) and W2(a). Because of this conflict we need to put T1 before T2 i.e T1 --->T2.

Second conflict is between W1(c) vs R3(c). Because of this conflict we need to put T1 before T3 i.e T1 --->T3.

Third conflict is between W2(a) and W4(a). Because of this conflict we need to put T2 before T4 i.e T2 --->T4.

Fourth conflict is between R1(a) and W4(a). Because of this conflict we need to put T1 before T4 i.e T1 --->T4

Since we have no cycle in the precedence graph, this schedule is conflict serializable.

Possible equivalent serial schedules are:

T1, T2, T4, T3

T1, T3, T2, T4

T1, T2, T3, T4

SCRATCH AREA (2 pages) – DO NOT TAKE WITH YOU!

THIS WILL NOT BE MARKED AND WILL BE DISPOSED OF BEFORE MARKING STARTS!