# British Columbia Institute of Technology

# COMP 2714-Relational Database Systems

**Final Exam**

**Friday, Dec 13, 2024**

**Duration: 110 Minutes**

**Name: Enter in the Answersheet**

**Student ID: Enter in the Answersheet**

**The exam is 35 multi-choice questions.**
**There are two written questions**
**Please refer to the answers booklet for the instructions**

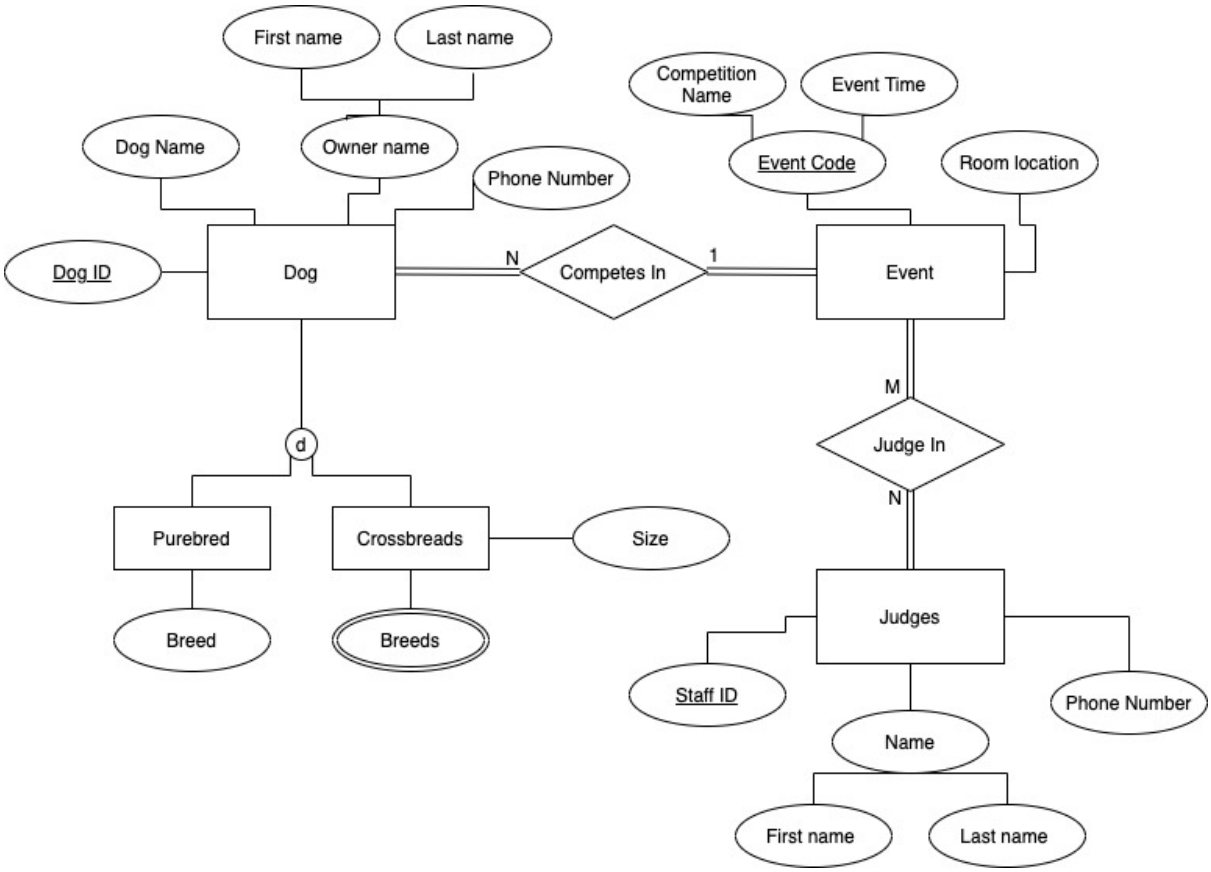
**Note: This question booklet is confidential, and distribution of it to anyone (student, non-student, from BCIT, or outside BCIT) is strictly prohibited and will result in penalties.**

**Note: You will enter your answers in the answers booklet, not in this document. Make sure to download it. You will only submit the answer booklet on Learning Hub at the end.**

# Section 1:

**35 Multiple Choice Questions**
**Circle: One single answer**
**Square: One or more correct answers**

## 1- Which of the following statements is TRUE?




| | |
|---|---|
| ☐ A. | Each dog can compete in multiple events |
| ☐ B. | An event can be identified by its Competition Name or Event Code |
| ☐ C. | An event does not require at least one judge |
| ☐ D. | A dog can be both purebred and crossbred |
| ☐ E. | B and C |
| ☐ F. | None of the above |

**2- What's the minimal cover (with union) and consequently the final answer of decomposing the following relation to 3NF?**

**R[A,B,C,D,E]**

$A \longmapsto B,C$

$DC \longmapsto E$

$A \longmapsto E$

$B \longmapsto E$

$D \longmapsto C$

A.

Minimal Cover:

A ⟼ B
A ⟼ C
D ⟼ E
B ⟼ E
D ⟼ C

Final Answer:
R1[A,B]
R2[A,C]
R3[D,E]
R4[B,E]
R5[D,C]
Where the FD are the ones listed above (in the minimal cover)

B.

Minimal Cover:

A⟼ B,C
DC ⟼ E
A ⟼ E
B ⟼ E
D ⟼ C

Final Answer:
R1[A,B,C]
R2[D,C,E]
R3[A,E]
R4[B,E]
R5[D,C]
Where the FD are the ones listed above (in the minimal cover).

C.

Minimal Cover:

A ⟼ B, C
B ⟼ E
D ⟼ C, E

Final Answer:
R1[A,B,C]
R2[B,E]
R3[D,C,E]
Where the FD are the ones listen above (in the minimal cover).

D.

Minimal Cover:

A⟼ B,C, E
D ⟼ C, E
B ⟼ E

Final Answer:
R1[A,B,C, E]
R2[D,C,E]
R3[B,E]
Where the FD are the ones listed above (in the minimal cover).

**3- The following functional dependencies are given:**

**AB --> CD,  AF --> D,  DE --> F,  C --> G,  F --> E,  G--> A;**

**Which one of the following options is false ?**

| | |
|---|---|
| ◯ A. | $CF^+$ = { ADCEFG } |
| ◯ B. | $BG^+$ = { ABCDG } |
| ◯ C. | $AF^+$ = { ACDEFG } |
| ◯ D. | $AB^+$ = { ABCDG } |

**4- Which of the following is not a superkey for the relation R[A,B,C,D,E,F,G]?**

**AC --> GF**

**D --> C**

**F --> C**

**E --> A**

| | |
|---|---|
| ◯ A. | ABCDEFG |
| ◯ B. | ABCDE |
| ◯ C. | BDE |
| ◯ D. | ABD |

**5- What will the following statement display from the given table *Employee*?**

**Employee [ssn, name, dob, address, sex, salary]**

```
SELECT DISTINCT A.name, A.salary, A.ssn
FROM Employee A, Employee B
WHERE A.salary > B.salary
```

| | |
|---|---|
| ◯ A. | Display the name, ssn and salary of the employee(s) with the highest salary. |
| ◯ B. | Display the name, salary and ssn of employees that earn less than the highest paid employee(s) |
| ◯ C. | Display the name, ssn and salary of employees that earn more than the highest paid employee(s). |
| ◯ D. | Display the name, salary and ssn of employees that earn more than the lowest paid employee(s). |

**6- Consider**

Employee

| ID | Name | Salary | Department |
|---|---|---|---|
| 1751 | Paris | 60,000 | 2 |
| 4671 | Anna | 70,000 | 1 |
| 1023 | Ben | 40,000 | 4 |
| 2670 | Grace | 50,000 | 2 |

Department

| ID | Name | Manager |
|---|---|---|
| 1 | Marketing | 4671 |
| 2 | Development | 1751 |
| 4 | HR | 1023 |

If insert a tuple to Empolyee table

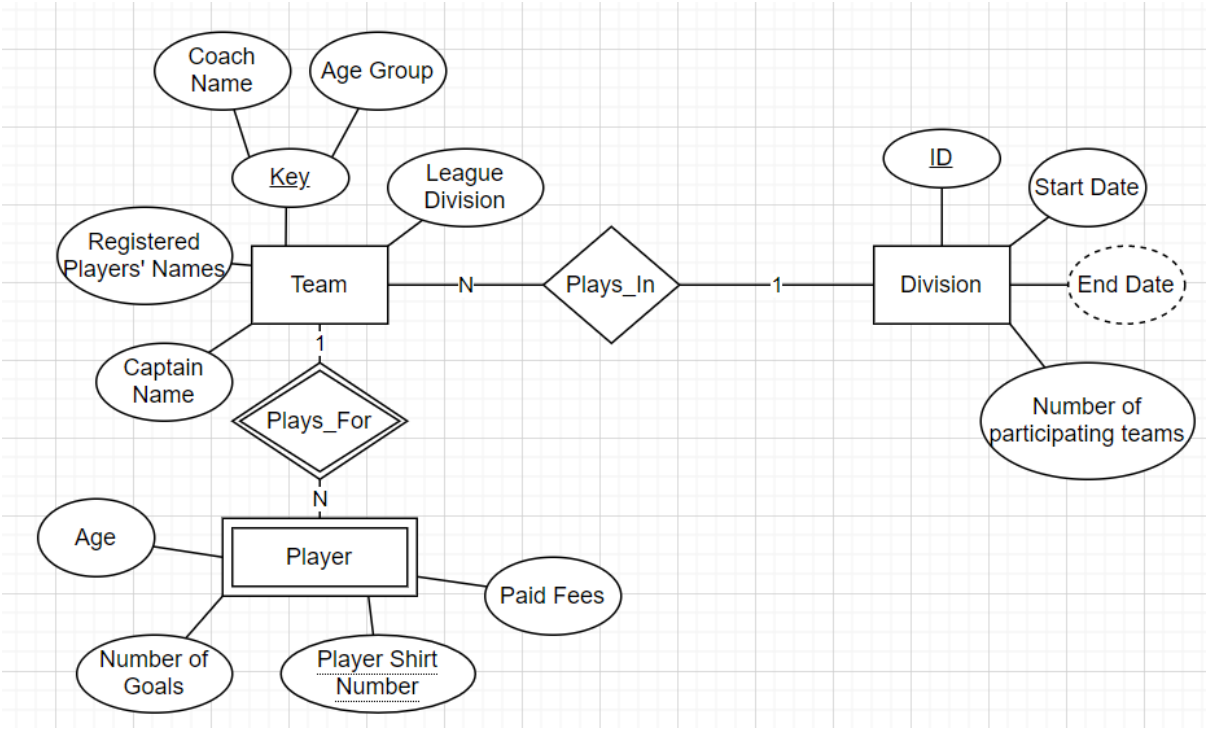
| NULL | John | 70,000 | 4 |
|---|---|---|---|

which constrains was violated?

| | |
|---|---|
| ◯ A. | Domain constraints |
| ◯ B. | Key constraints |
| ◯ C. | Referential Integrity constraints |
| ◯ D. | Entity Integrity constraints |

**7- Consider the UoD for the following ER diagram. For the following ER Diagram, select the answers that correspond with changes that should be made to the ER diagram to improve its accuracy.**

A local soccer team, uniquely identified by their coach's name and age group, plays in a local division. This league division, alongside all registered player's names (it is assumed that, for a team to exist, they must have registered players) and their captain's name(s) are recorded as the team's details. The league division itself is identifiable due to its unique identification number and stores information about the competition start and end dates, alongside the number of teams competing in the division. Players, themselves, play for one team, with information about their age, number of goals and number of associated fees paid being recorded, alongside them being identified by a combination of their team key and shirt number. Below is the attached ER diagram for this UoD:




| | |
|---|---|
| ☐ A. | The captain name attribute should be a multi-valued attribute. |
| ☐ B. | The relationship of player-to-team should be one-to-one. |
| ☐ C. | The end date attribute should be a normal attribute and not one derived from the start date, as in the ER diagram. |
| ☐ D. | There should be a relationship (many-to-one) between the entity "player" and the entity "division". |

## 8- If ID —>level then {ID, payment} —> {level, payment} ,this is?

| | |
|---|---|
| ☐ A. | Reflexivity |
| ☐ B. | Augmentation |
| ☐ C. | Transitivity |
| ☐ D. | None of hte above |

**9- Given the three relations R1, R2, and R3, what tuple will be found in both R1/R2 and R1/R3?**

R1

| Name | Pet | Number of Children |
|---|---|---|
| John | Fish | 1 |
| John | Cat | 1 |
| Andy | Dog | 0 |
| David | Cat | 2 |
| David | Dog | 2 |
| David | Fish | 2 |
| Chloe | Dog | 1 |
| Chloe | Fish | 1 |
| Amy | Dog | 1 |
| Amy | Fish | 1 |
| Amy | Cat | 1 |

R2

| Pet | Number of Children |
|---|---|
| Cat | 1 |

R3

| Pet | Number of Children |
|---|---|
| Fish | 1 |
| Dog | 1 |

- A. (John)
- B. (Andy)
- C. (David)
- D. (Chloe
- E. (Amy)

**10- Given the below instance in a database, which of the following SQL statements will commit the following changes to the 2 tuples?**

Students

| SID | Name | Grp |
|---|---|---|
| 16777216 | John Doe | 1 |
| 12435687 | Mary Lam | 2 |
| 15607740 | Harry Fred | 2 |
| 14269541 | Kim Doris | 1 |
| 12003562 | Nathan Vernons | 4 |
| 16589002 | Serena Vernons | 2 |
| 15950251 | Paul Jake | 4 |
| 16322693 | Reed Manson | 3 |

(12003562, Nathan Vernons, 4) => (12003562, Nathan Vernons, 5)

(16589002, Serena Vernons, 2) => (16589002, Serena Vernons, 5)

A.

```
UPDATE TABLE Students
SET Grp = 5
WHERE Name LIKE '%Vernons'
```

B.

```
UPDATE Students
SET Grp = 5
WHERE Name LIKE '%Vernons'
```

C.

```
UPDATE Students
SET Grp = 5
WHERE SID = 12003562 AND SID = 16589002
```

D.

```
UPDATE Students
SET Grp = 5
WHERE Name = "Nathan Vernons" AND Name = "Serena Vernons"
```

## 11- What is logical data independence?

| | |
|---|---|
| ◯ A. | The ability for the DBMS to define integrity constraints by itself. |
| ◯ B. | Capacity to change internal schema (e.g. indexes, location, etc.) without having to change conceptual (or external) schemas |
| ◯ C. | Capacity to change conceptual schema (e.g. attributes, tables, constraints, data types, etc.) without having to change external schema or the application programs that access the DB via external schemas |
| ◯ D. | Both B and C |

**12- If R[A, B, C, D, E]**

**A -> B**

**BCD -> AE**

**What is the highest normal form of this relation?**

| | |
|---|---|
| ○ A. | 1NF |
| ○ B. | 2NF |
| ○ C. | 3NF |
| ○ D. | BCNF |

**13-Assume the following FDs hold for relation R(A, B, C, D, E, F):**

**F={ A→B, (A,C)→D, (E,F)→A, (B,E)→F, (C,E)→F }**

**Which of the following is a key for the above relation?**

| | |
|---|---|
| ○ A. | ED |
| ○ B. | ACE |
| ○ C. | AC |
| ○ D. | CE |

## 14- Which of the following statements are the incorrect Armstrong's Axioms

| | |
|---|---|
| ◯ A. | Reflexivity: If $Y \subseteq X$, then $X \rightarrow Y$ |
| ◯ B. | Transitivity: if $X \rightarrow Y$ and $WY \rightarrow Z$ then $WX \rightarrow Z$ |
| ◯ C. | Augmentation: If $X \rightarrow Y$, then $XZ \rightarrow YZ$ for any Z |
| ◯ D. | Union: if $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$ |

**15- Consider**

| Id | Title | Director | Year | Length_minutes |
|---|---|---|---|---|
| 1 | Toy Story | John Lasseter | 1995 | 81 |
| 2 | A Bug's Life | John Lasseter | 1998 | 95 |
| 3 | Toy Story 2 | John Lasseter | 1999 | 93 |
| 4 | Monsters, Inc. | Pete Docter | 2001 | 92 |
| 5 | Finding Nemo | Andrew Stanton | 2003 | 107 |
| 6 | The Incredibles | Brad Bird | 2004 | 116 |
| 7 | Cars | John Lasseter | 2006 | 117 |
| 8 | Ratatouille | Brad Bird | 2007 | 115 |
| 9 | WALL-E | Andrew Stanton | 2008 | 104 |
| 10 | Up | Pete Docter | 2009 | 101 |

**According to this table, how to Find movies not released in the year between 2000 and 2010?**

A.
```
SELECT title, year
FROM movies
WHERE year Between 2000 and 2010;
```

B.
```
SELECT title, year
FROM movies
WHERE year < 2000 and year > 2010;
```

C.
```
SELECT title, year
FROM movies
WHERE year < 2000 OR year > 2010;
```

D.
```
SELECT title, COUNT(*)
FROM movies
WHERE year < 2000 OR year > 2010;
```

**16- Which of the following is a correct BCNF decompostion for R(ABCDEFG):**

**with FDs D->EF and E->DF**

| | |
|---|---|
| ○ A. | R1(DEF), R2(ABCDG) |
| ○ B. | R1(DEF), R3(EDF), R4(ABCG) |
| ○ C. | R1(EF), R2(ABCDG) |
| ○ D. | None of the above |

**17- A user approaches you and asks to be granted a specific privilege (assume a view and database has already been created and organised and only the query needed to be made). Which access control mechanism(s) would be used?**

| | |
|---|---|
| ○ A. | Discretionary Access Control. |
| ○ B. | Mandatory Access Control. |
| ○ C. | Discretionary Access Control and Data Manipulation Lanugage. |
| ○ D. | Role Based Access Control and Data Manipulation Language. |

**18- Which SQL Query will select all employees' first names and hire_date, who started working in 1985:**

**employees:**

| emp_no | birth_date | first_name | last_name | gender | hire_date |
|---|---|---|---|---|---|
| 10001 | 1953-09-02 | Georgi | Facello | M | 1986-06-26 |
| 10002 | 1964-06-02 | Bezalel | Simmel | F | 1985-11-21 |
| 10003 | 1959-12-03 | Parto | Bamford | M | 1986-08-28 |
| 10004 | 1954-05-01 | Chirstian | Koblick | M | 1986-12-01 |
| 10005 | 1955-01-21 | Kyoichi | Maliniak | M | 1989-09-12 |
| 10006 | 1953-04-20 | Anneke | Preusig | F | 1989-06-02 |
| 10007 | 1957-05-23 | Tzvetan | Zielinski | F | 1989-02-10 |
| 10008 | 1958-02-19 | Saniya | Kalloufi | M | 1994-09-15 |
| 10009 | 1952-04-19 | Sumant | Peac | F | 1985-02-18 |
| 10010 | 1963-06-01 | Duangkaew | Piveteau | F | 1989-08-24 |

**Result:**

| first_name | hire_date |
|---|---|
| Bezalel | 1985-11-21 |
| Sumant | 1985-02-18 |

| | |
|---|---|
| ◯ A. | SELECT first_name, hire_date FROM employees WHERE hire_date = 1985 |
| ◯ B. | SELECT * FROM employees WHERE YEAR(hire_date) == "1985" |
| ◯ C. | SELECT first_name, hire_date FROM employees WHERE YEAR(hire_date) = 1985 |
| ◯ D. | SELECT first_name, hire_date FROM employees WHERE YEAR(hire_date) = YEAR(1985) |

19-

**CREATE TABLE School (**

**studentid INT REFERENCES Student(id) ON DELETE CASCADE,**

**teacherid INT REFERENCES Teacher(id) ON DELETE SET NULL,**

**programid INT REFERENCES Program(id) ON DELETE NO ACTION,**

**PRIMARY KEY(studentid, teacherid, programid)**

**)**

**Which of the following choices are true?**

| | |
|---|---|
| ☐ A. | If a tuple from School is deleted, any tuples in Student that are referenced by this tuple are also deleted. |
| ☐ B. | If a tuple from Teacher is deleted, the *teacherid* of some tuples of School that reference this deleted tuple may have their values changed to NULL. |
| ☐ C. | If a tuple with an id that does not exist in School is inserted into Program, this operation is rejected. |
| ☐ D. | If a tuple with an id that does not exist in Program is inserted into School, this operation is rejected. |

**20- Consider**

| ManagerID | Name |
|---|---|
| 12 | Jack |
| 12 | Faker |
| 14 | Mark |
| 15 | Mary |

**SELECT Distinct ManagerID,Name**

**FROM employees;**

**Which one will return?**

**Note:  ManagerID is not unique just for this case**

A.

| ManagerID | Name |
|---|---|
| 12 | Faker |
| 14 | Mark |
| 15 | Mary |

B.

| ManagerID | Name |
|---|---|
| 12 | Jack |
| 12 | Faker |
| 14 | Mark |
| 15 | Mary |

C.

| ManagerID | Name |
|---|---|
| 12 | Jack |
| 14 | Mark |
| 15 | Mary |

D.

| ManagerID | Name |
|---|---|
| 14 | Mark |
| 15 | Mary |

## 21- Which two descriptions are correct about DELETE and DROP?

| | |
|---|---|
| ☐ A. | DELETE is a DDL command, DROP is a DML command. |
| ☐ B. | DELETE is a DML command, and DROP is a DDL command. |
| ☐ C. | DELETE can remove tuples from a relation, DROP can remove entire table. |
| ☐ D. | DELETE can remove entire table, DROP can remove tuples from a relation. |

**22- Student[ID, Name, Gender, Age]**

**Course[cNumber, cName]**

**Score[ID, cNumber, Grade]**

**Foreign Keys:**

**Score.ID references Student.ID**

**Score.cNumber references Course.cNumber**

**Based on the schema, how to get the average score of courses that have at least 30 students and whose course number starts with 2**

A.

```
SELECT cNumber, AVG(Grade)
FROM Score
WHERE cNumber LIKE '2%'
GROUP BY ID
HAVING COUNT(ID) >=30
```

B.

```
SELECT cNumber, AVG(Grade)
FROM Score
WHERE cNumber LIKE '2%'
GROUP BY cNumber
HAVING COUNT(ID) >=30
```

C.

```
SELECT cNumber, AVG(Grade)
FROM Score
WHERE cNumber LIKE '2%' AND COUNT(ID) >=30
```

D.

```
SELECT cNumber, AVG(Grade)
FROM Course
WHERE cNumber LIKE '2%'
GROUP BY cNumber
HAVING COUNT(ID) >=30
```

**23- Consider**

| Movies | | |
|---|---|---|
| ID | title | year |
| 1 | Star Wars: Episode 1 | 1977 |
| 2 | Star Wars: Episode 2 | 1980 |
| 3 | Star Wars: Episode 3 | 1983 |
| 4 | Solo: A Star Wars Story | 2018 |
| 5 | Rogue One: A Star Wars Story | 2016 |

**Given the relation Movies, which query would select the title of ONLY the first 3 Star Wars movies EXCLUDING any stand alone Star Wars stories.**

A.
```
SELECT title FROM movies WHERE title LIKE "%Star Wars%";
```

B.
```
SELECT title FROM movies WHERE title LIKE "Star Wars%";
```

C.
```
SELECT title FROM movies WHERE title = "Star Wars";
```

D.
```
SELECT title FROM movies WHERE title LIKE "Star Wars";
```

## 24- Which statement below is FALSE?

| | |
|---|---|
| ◯ A. | Inner joins cannot include the tuples that do not satisfy the join condition. |
| ◯ B. | For inter joins, in which a tuple is included in the result relation only if matching tuples exist in both relations. |
| ◯ C. | Aggregation functions can be used in the SELECT clause, GROUP BY clause, and the HAVING clause. |
| ◯ D. | Aggregates can be applied to multiple groups of tuples specified by the GROUP BY clause. |

25-

**Using the following data and relational schema, how would you return the number of pets each student has (even if they have 0). The query should include: the student number, the name of the student, and the number of pets they have titled "number_of_pets".**

**STUDENT**

| STUDENT_ID | NAME | DEGREE | GPA |
|---|---|---|---|
| 46420052 | Bill Smith | Arts | 5.75 |
| 46420572 | Steve Infs | Engineering | 7 |
| 46498761 | Frederick Southern | Computer Science | 6.75 |
| 46463339 | Blake McFee | Biomedical Science | 4.5 |

**PETS**

| STUDENT_ID | PET |
|---|---|
| 46420052 | Jerry |
| 46420572 | Tom |
| 46498761 | Fin |
| 46420052 | Biscuit |

**Relations:**

Student (studentID, name, degree, GPA)
Pets (studentID, pet)

**Foreign Keys:**

Pets.studentID references Student.studentID

A.

```
SELECT Student.studentID, Student.name, COUNT(pet)
FROM Student, Pets
```

B.

```
SELECT studentID, name, COUNT(pet) AS “number_of_pets”
FROM Student
INNER JOIN Pets ON Pets.studentID = Student.studentID
```

C.

```
SELECT studentID, name, COUNT(pet) AS “number_of_pets”
FROM Student
INNER JOIN Pets ON Pets.studentID = Student.studentID
GROUP BY studentID
```

D.

```
SELECT studentID, name, COUNT(pet) AS “number_of_pets”
FROM Student
LEFT JOIN Pets ON Pets.studentID = Student.studentID
GROUP BY studentID
```

**26- The following are two tables, Customers and Orders. Writing SQL to find the customers who Never order.**

Customers

| Id | Name |
|---|---|
| 1 | Joe |
| 2 | Henry |
| 3 | Sam |
| 4 | Max |

Orders

| Id | CustomerId |
|---|---|
| 1 | 3 |
| 2 | 1 |

A.

SELECT Name AS Customers

FROM Customers

WHERE Id IN (SELECT Customers.Id FROM Customers, Orders WHERE Customers.Id = CustomerId);

B.

SELECT Name AS Customers

FROM Customers

WHERE Id NOT IN (SELECT Customers.Id FROM Customers, Orders WHERE Customers.Id = CustomerId);

C.

SELECT Name AS Customers

FROM Orders

WHERE Id NOT IN (SELECT Customers.Id FROM Customers, Orders WHERE Customers.Id = CustomerId);

D.

All of the above are wrong

**27- Suppose we have the following relational schema:**

Player [first_name, last_name, points_per_game, team]

"Return the team(s) which have at least 2 players score more than 20 points per game."

Which one of the following queries acheives this?

A.

```
SELECT COUNT(*)
FROM Player
WHERE points_per_game > =20
GROUP BY team
HAVING COUNT(*) > 2
```

B.

```
SELECT UNIQUE team
FROM Player
WHERE points_per_game > 20
AND team IN
(SELECT team
FROM Player
GROUP BY team
HAVING COUNT(*) > 2)
```

C.

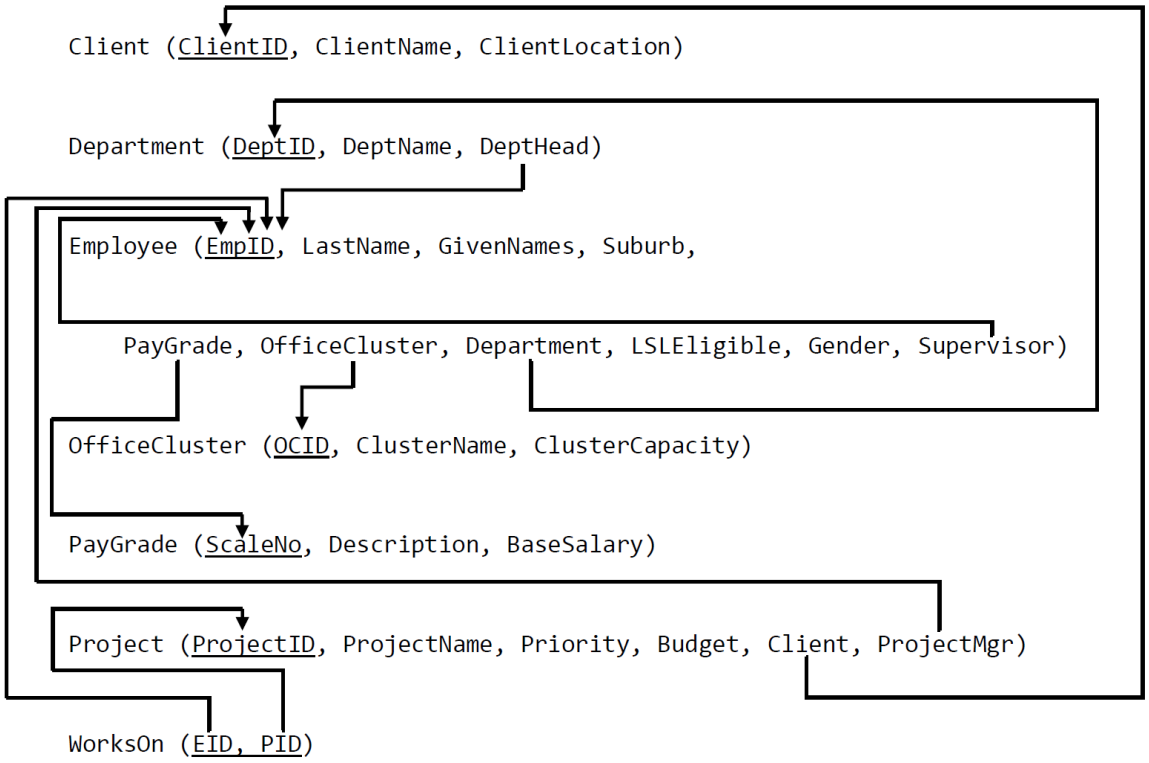
```
SELECT team
FROM Player
WHERE points_per_game > =20
AND COUNT(*) > 2
GROUP BY team
```

D.

```
SELECT team
FROM Player
WHERE points_per_game > 20
GROUP BY team
HAVING COUNT(*) > 2
```

**28- Consider**

Database Schema




For the above schema, select the query that will list the full names of all employees who are supervisors.

A.

```
SELECT FullName
FROM Supervisors;
```

B.

```
SELECT A.GivenNames, A.LastName
FROM Employee A, Employee B
WHERE A.Supervisor= B.EmpID;
```

C.

```
SELECT GivenNames, LastName
FROM Employee
WHERE Supervisor= EmpID;
```

D.

```
SELECT A.GivenNames, A.LastName
FROM Employee A, Employee B
WHERE B.Supervisor= B.EmpID;
```

**29- Which statement is TRUE for minimal superkey?**

| | |
|---|---|
| ○ A. | All super keys can be primary keys. |
| ○ B. | various super keys together makes the criteria to select the candidate keys. |
| ○ C. | In a relation, number of super keys are more than number of primary keys. |
| ○ D. | Super key's attributes can not contain NULL values. |

**30- Consider**

Employee

| ID | Name | Salary | Department |
|---|---|---|---|
| 314159265 | Maradona Kelly | $55,000 | 1 |
| 123456789 | Johnny Sins | $100,000 | 1 |
| 987654321 | Ben Ben | $60,000 | 3 |
| 960242069 | Joe | $75,000 | 2 |

Department

| Number | dName | ManagerID |
|---|---|---|
| 1 | Product Development | 123456789 |
| 2 | Scouting | 960242069 |
| 3 | Admin | NULL |

Which of the tables below would be the result of the following query?

```
SELECT D.number, D.dname, E.name
FROM Department AS D
LEFT JOIN Employee AS E ON D.managerID = E.ID
```

☐ A.

| Number | dName | Name |
|---|---|---|
| 1 | Product Development | Johnny Sins |
| 2 | Scouting | Joe |
| 3 | Admin | NULL |

☐ B.

| Number | dName | Name |
|---|---|---|
| 1 | Product Development | Johnny Sins |
| 2 | Scouting | Joe |

☐ C.

| ID | Name | dName |
|---|---|---|
| 314159265 | Maradona Kelly | Product Development |
| 123456789 | Johnny Sins | Product Development |
| 987654321 | Ben Ben | Admin |
| 960242069 | Joe | Scouting |

☐ D.

| ID | Name | dName |
|---|---|---|
| 123456789 | Johnny Sins | Product Development |
| 960242069 | Joe | Scouting |

**31- Consider the relation with schema PHONE(BRAND, STORAGE, MODEL, COLOUR, WEIGHT).**

If we issue a query of the form:

SELECT ...

FROM PHONE

WHERE ...

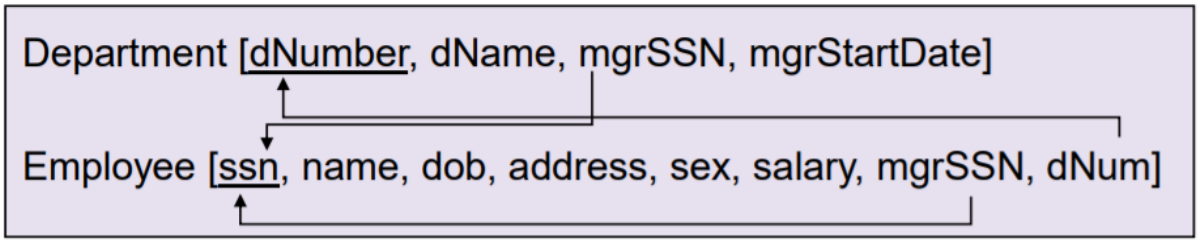
GROUP BY STORAGE, WEIGHT

HAVING ???

Which term(s) can not appear in the HAVING condition?

| | |
|---|---|
| ◯ A. | BRAND |
| ◯ B. | STORAGE |
| ◯ C. | COUNT (STORAGE) |
| ◯ D. | All can appear |
| ◯ E. | None can appear |

**32- Consider**




According to this relational schema, how to list all the departments and their managers who have salaries larger than 30K?

| | |
|---|---|
| ◯ A. | SELECT D.dName, M.Name, M.salary<br>FROM Deparment D, Employee M<br>WHERE D.mgrSSN = M.ssn AND M.salary > 30000; |
| ◯ B. | SELECT D.dName, M.Name<br>FROM Department D, Manager M<br>WHERE D.mgrSSN = M.ssn and M.salary > 30000; |
| ◯ C. | SELECT D.dName, M.Name<br>FROM Deparment AS D<br>LEFT JOIN Employee AS M ON D.mgrSSN = M.ssn AND M.salary < 30000; |
| ◯ D. | None of the above |

**33- Now, we have a table named Scores and 'id' is the primary key. There are 10 rows in this table and one of which contains id=11.**

**First**

```
CREATE VIEW View_score AS

SELECT * FROM Scores;
```

**Then**

```
DELETE FROM View_score

WHERE id=11;
```

**LAST**

```
SELECT * FROM Scores;

SELECT * FROM View_score;
```

QUESTION: How many rows will return in the **LAST** part?

| | |
|---|---|
| ◯ A. | 10, 10 |
| ◯ B. | 10, 9 |
| ◯ C. | 9, 10 |
| ◯ D. | 9, 9 |

**34- Assume the following schema**

Player [ID, name]

Game [playerID, gameName]

Game.playerID references Player.ID

write a query that returns the name of the player who has played at least all the games that player Y has played. (the ID of player Y is 1)

A.

```
SELECT name
FROM Player
WHERE gameName IN (
    SELECT gameName
    FROM Game
    WHERE Game.playerID =1
)
```

B.

```
SELECT name
FROM Player
WHERE NOT EXISTS (
    SELECT *
    FROM Game
    WHERE playerID = 1 AND name NOT IN (
        SELECT name
        FROM Player
        WHERE Player.ID = Game.playerID
    )
)
```

C.

```
SELECT name
FROM Player
WHERE NOT EXISTS (
    SELECT *
    FROM Game
    WHERE playerID = 1 AND gameName NOT IN (
        SELECT gameName
        FROM Game
        WHERE Player.ID = Game.playerID
    )
)
```

**35-Consider the Following table, "Devices", which is a bell-LaPadula Model:**

If Leah executed the following SQL Query in the table Devices, What would be returned?

(Assume polyinstantiation is in effect.)

SELECT firstName, status, device

FROM Devices

WHERE status LIKE "OFF"

| User | Security Clearance |
|---|---|
| Lisa | TS |
| John | S |
| Cassie | C |
| Leah | U |
| Zack | U |
| David | U |

| ID | firstName | status | device | TC |
|---|---|---|---|---|
| 1001 U | Jake U | ON C | phone C | C |
| 1002 U | Ryan U | OFF C | phone C | C |
| 1003 U | Natasha C | OFF U | tablet U | C |
| 1004 U | Leah C | ON C | dekstop C | C |
| 1005 U | James C | OFF U | laptop U | C |
| 1006 TS | Christie TS | OFF TS | tablet TS | TS |

| Functional Dependencies |
|---|
| {ID} > {firstName, device} |

- A. Natasha, NULL, NULL
  NULL, NULL, NULL
- B. NULL, OFF, tablet
  NULL, OFF, laptop
  NULL, NULL, NULL
- C. NULL, OFF, tablet
  NULL, OFF, laptop
- D. Error

# Section 2:

**2 Written questions (Question 36 and Question 37)**

**Insert your solution to the answer sheet.**
**Go to the next page!**

## Question 36:

**Events Inc. is a small start-up company which provides its users with an event tracking and recommendation platform for various local community activities. A simplified version of their database schema has been provided below including foreign key constraints.**

## Relational Schema

User [id, fName, mInitial, lName, age, phone, email, nationality, significantOther]

Event [title, date, description, location, sponsor]

Attends [id, title, date, travelMethod]

Friends [requestor, requestee, startDate]

## Foreign Keys

User.significantOther references User.id

Attends.{title, date} references Event.{title, date}

Attends.id references User.id

Friends.requestor references User.id

Friends.requestee references User.id

## Query1:

Return a list with the number of events each user has attended in descending order of the number of events. This query should return a table with two columns, one for user id and one for the number of events attended by that user. Users who have not attended any events can be ignored.

## Query2:

Find the total number of users where the nationality of their significant other has at least three people. That is to say, the system has recorded at least three users of that nationality including the significant other. This query should return a table containing a single column which has a single numerical tuple.

**Question 37: Based on the following relational schema and functional dependencies, find minimal cover for relation R and then decompose R to 3NF such that all functional dependencies and candidate keys are preserved.**

**R [A, B, C, D, E, F, G, H, I, J, K]**

**{K}->{J,I,H,G,F}**

**{F}->{A,B,C,D}**

**{B}->{A}**

**{I}->{J,K,G}**

**{F, B} → {K, D, E}**

**{B,C} →{A}**