Type: MCQ

Q1. If r∩s if Φ, then r ⋈ s is (0.5)
1. Φ
2. r
3. s
4. **None of the above

Q2. Consider the following transaction involving two accounts x and y. read(x);x=x-50; write(x); read(y);y=y+50; write(y). The constraint that the sum of the accounts x and y should remain constant is that of _____(0.5)
1. Atomicity

2. **Consistency

3. Isolation

4. All of the mentioned

Q3. What does the following execution result in?(Assuming tables are created initially and non-empty)
insert into Table1 values('A', 9, 'CSE')
savepoint s1;
insert into Table1 values('B', 9.5, 'EEE')
savepoint s2;
delete * from Table 1;
commit;
insert into Table1 values('C', 7, 'CSE')
rollback to s1; (0.5)
1. **Error that savepoint doesn't exist
2. Rollsback to savepoint s2
3. Transaction halts
4. None of the above

Q4. Set of all permissible values for an attribute is: (0.5)
a) Constraint
b) **Domain
c) Tuple
d) Relation

Q5. Let faculty(Id, name, dept_name) and depart(dept_name, location, budget) be two schemas. If dept_name attribute was specified using unique constraint in depart, which of the following SQL commands must be used to specify foreign key constraint from faculty to depart? (0.5)
A) alter table faculty add constraint FK_faculty foreign key(dept_name) references depart;
B) **alter table faculty add constraint FK_faculty foreign key(dept_name) references depart(dept_name);
C) alter table depart add constraint UQ_depart unique(dept_name);
D) Both A) and B)

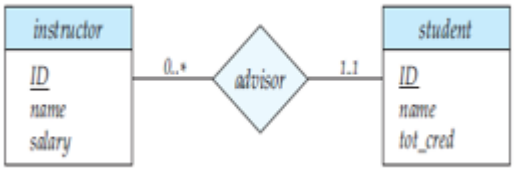
Q6. Use University schema: Select name, ID from student where exists (select * from takes where takes.ID=student.ID); (0.5)

Which of the below queries produce same output as the query above ?
a. **Select name, ID from student where ID in (select ID from takes natural join student);
b. Select name, ID from student where ID = (select ID from takes natural join student);
c. Select name, ID from student where ID = all(select ID from takes natural join student);
d. None of the mentioned

CO2

Q7. Student table :
ID Name
101 Abhishek
102 Ananth
103 NULL
In the Student table , what does the following query return?
select count(Name) from Student; (0.5)
1. **2
2. 1
3. 3
4. Error

Q8. What does the above diagram shows: (0.5)



1. **Relationship from one to many from instructor to student with total participation of student in advisor
2. Relationship from many to one from instructor to student with total participation of student in advisor
3. Relationship from one to many from student to instructor with total participation of instructor in advisor
4. Relationship from one to many from instructor to student with partial participation of student in advisor

Q9. Given R(A,B,C,D,E,F,G)
F= { A->C, A->DE, B-> ACD, D->FG}, which of the following is a candidate key? (0.5)
1. A
2. **B
3. D
4. AD

Q10. Which normal form sometimes is not able to preserve functional dependency after the lossless decomposition.

1. 3NF
2. 4NF
3. **BCNF**
4. 5NF

Type: DES

Q11. A. Consider R(A,B,C,D,E,H) with FDs, F = {A->C; AC->D; E->AD; E->H}
Find key for R. Is R in 3NF? If not normalize R into 3NF using 3NF Decomposition algorith (4)

Soln:

Key: BE as E+ is R (1)

Applying 3NF Decomposition algorithm on R;

1 Find Fc;

Fc: E→ AH; A→ CD ***with justification (1)***

2. Create a relation for each FD:

R1(EAH); R2(ACD) (1)

3. Check whether any of the Rs contain Key. Yes (0.5)

BE is not contained in any relation

4. Check for redundant relations. No such Relations.

Final : **R1(EAH); R2(ACD); R3(BE) (0.5)**

Q12. Consider a relational database given below :

Employee(person_name, street, city)

Works(person_name, company_name, salary)

Company(company_name, city)

Manager(person_name, manager_name)

Write relational algebraic expressions for the following queries:

a. Give all employees of First Bank Corporation a 10 percent salary raise.
b. Find the company with the most employees.
c. Find those companies whose employees earn a higher salary, on average, than the average salary at First Bank Corporation.
d. Delete all tuples in the works relation for employees of Small Bank Corporation. (4)

Soln:

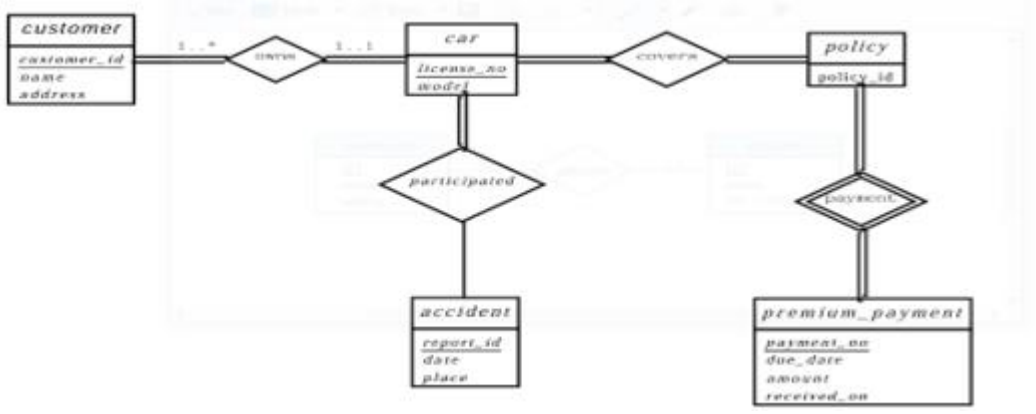
a. works ← Πperson-name,company-name,1.1∗salary(σ(company-name="First Bank Corporation") (works)) ∪ (works − company-name="First Bank Corporation" (works))

b. t1 ← company-nameGcount-distinct person-name (works) t2 ← maxnum-employees(ρcompany-strength(company-name,num-employees)(t1)) Πcompany-name(ρt3(company-name,num-employees)(t1) ρt4(num-employees)(t2))

c. . t1 ← company-nameGavg salary(works) t2 ← σcompany-name = “First Bank Corporation” (t1)

d. works ← works − σcompany−name=“Small Bank Corporation” (works)

Q13. Construct a ER diagram for a car insurance company where each customer can have one or more cars. Each car is associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars, and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received. (3)



Q14. Discuss the role of SQL views in database management. Explain their advantages, limitations, with appropriate example. (3)

Soln:

SQL views are virtual tables derived from one or more base tables or other views. They provide a logical representation of data, allowing users to query and manipulate data without directly accessing the underlying tables.

The **advantages** of SQL views include:

Data Abstraction: Views hide the complexity of underlying table structures and provide a simplified interface for users to access specific data subsets.
Security Enforcement: Views can restrict access to sensitive data by exposing only relevant columns or rows to users based on their permissions, enhancing data security.
Query Simplification: Views encapsulate frequently used joins, filters, and aggregations, simplifying complex queries and promoting code reuse and maintainability.
However, SQL views also have **limitations**, such as performance overhead due to query nesting and inability to index or modify view data directly.
Practical use cases of SQL views include generating customized reports, implementing access control mechanisms, abstracting complex data relationships, and presenting data in a user-friendly format tailored to specific application requirements.

Q15. Consider the following tables:

Book(ISBN, Title, Category, Price)
Author(ID, Name, Address)
Author_Book(ID, ISBN)
Write the SQL expressions for the following: (3)

a. List the books belonging to the category Textbook or Novel
b. List the book categories for which no. of books published is less than 5
c. List the authors who have written a book in each of the categories.(3)

Soln:

a. Select * from book
where Category in ('Textbook', 'Novel');

b. Select Category, count(*) from Book
Group by Category
Having count(*) <5;

c. Select ID, Name
From Author A
Where not exists((select Category from Book)
Minus
(select Category
from Book natural join Author_Book natural join Author B
where B.ID = A.ID));

Q16. List any six non-trivial functional dependencies satisfied by the relation in the table provided below.

A	B	C	D
a1	b1	c1	d1
a1	b1	c2	d2
a2	b1	c1	d1
a2	b1	c3	d3

(3)

Soln:
A-->B, B-->A, **C-->B**, **C-->D**, D-->C, **CD --> B, BC-->D, BD --> C**

Q17. Illustrate the usage of the following SQL constructs:

a) correlated subquery

b) like

c) left outer join (3)

Soln:

a. Find all courses taught in both the Fall 2009 semester and in the Spring 2010 semester

```
select course id from section as S
where semester = 'Fall' and year= 2009 and
exists (select * from section as T where semester = 'Spring'
and year= 2010 and S.course id= T.course id);
```

b. Find the names of all instructors whose name includes the substring "dar".

```
select name
from instructor
where name like '%dar%'
```

c. Find all students who have not taken a course

```
Select ID
from student natural left outer join takes
where course_id is null
```

Q18. Explain the importance of role names in ER model with an example (2)

Soln:

The function that an entity plays in a relationship is called that entity's role. Since entity sets participating in a relationship set are generally distinct, roles

are implicit and are not usually specified. However, they are useful when the meaning of a relationship needs clarification. Such is the case when the entity sets of a relationship set are not distinct; that is, the same entity set participates in a relationship set more than once, in different roles. In this type of relationship set, sometimes called a recursive relationship set, explicit role names are necessary to specify how an entity participates in a relationship instance. For example, consider the entity set course that records information about all the courses offered in the university. To depict the situation where one course (C2) is a prerequisite for another course (C1) we have relationship set prereq that is modeled by ordered pairs of course entities. The first course of a

pair takes the role of course C1, whereas the second takes the role of prerequisite course C2. In this way, all relationships of prereq are characterized by (C1, C2) pairs; (C2, C1) pairs are excluded.