IARE '24 DISCORD TEAM

II-II

DBMS MODULE 2 SOLUTIONS

VISHAL • UJJWAL • IKRAM

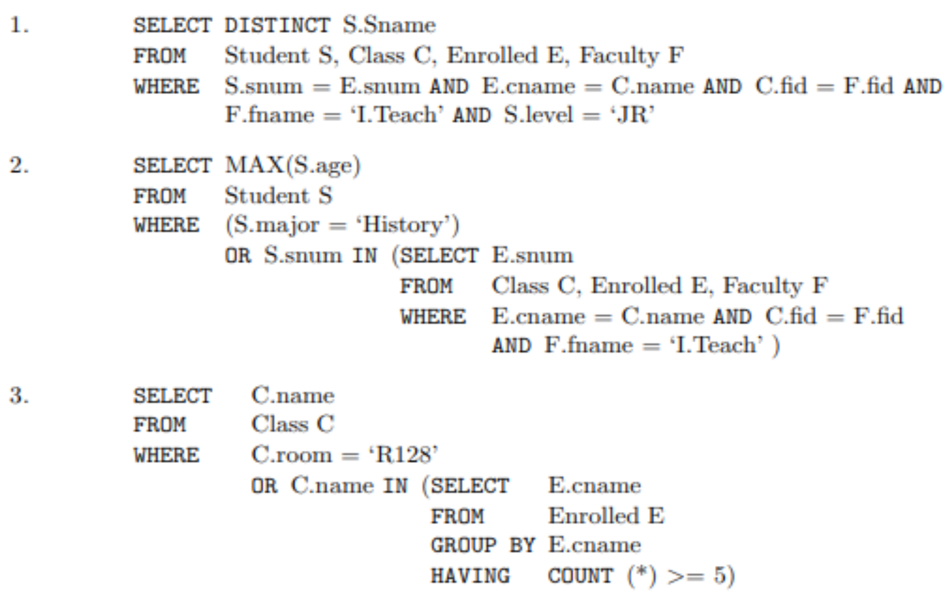
RELATIONAL APPROACH

DBMS MODULE 2

Part - A

1Q) For the following relational database, give the expressions in RA (Relational Algebra). Student (stuno, stuname, major, level, age), Class(Class name, meets at room, fid), faculty(fid, fname, deptID).

1. **Find the names of all juniors (level = JR) who are enrolled in a class taught by I.teach.**
2. **Find the age of the oldest student who is either a history major or is enrolled in a course taught by I.Teach.**
3. **Find the names of all classes that either meet in room R128 or have five or more students enrolled**

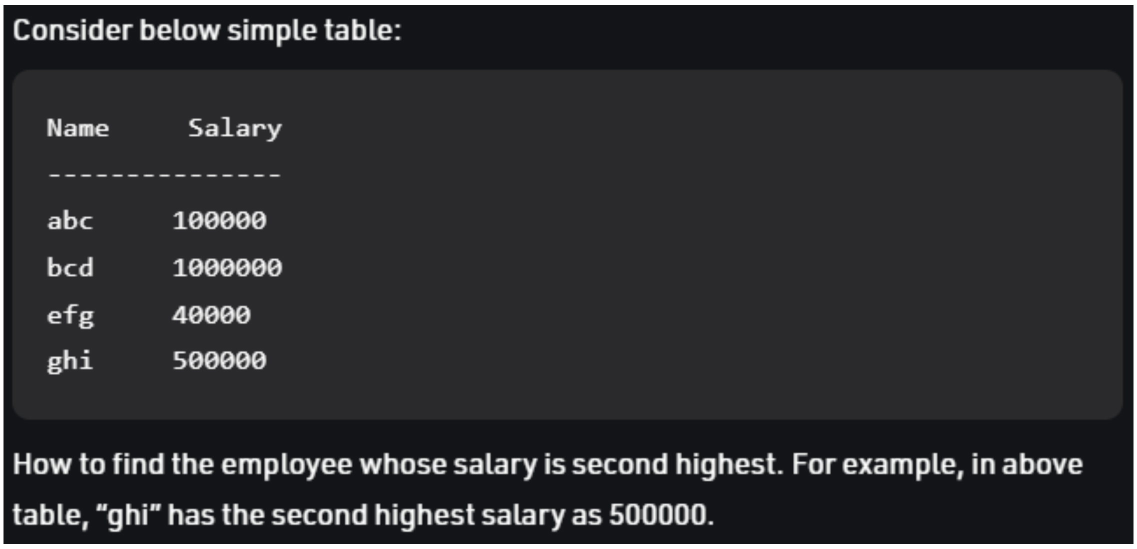


```
1.      SELECT DISTINCT S.Sname
        FROM    Student S, Class C, Enrolled E, Faculty F
        WHERE   S.snum = E.snum AND E.cname = C.name AND C.fid = F.fid AND
                F.fname = 'I.Teach' AND S.level = 'JR'

2.      SELECT MAX(S.age)
        FROM    Student S
        WHERE   (S.major = 'History')
                OR S.snum IN (SELECT E.snum
                              FROM   Class C, Enrolled E, Faculty F
                              WHERE  E.cname = C.name AND C.fid = F.fid
                                     AND F.fname = 'I.Teach' )

3.      SELECT    C.name
        FROM      Class C
        WHERE     C.room = 'R128'
                  OR C.name IN (SELECT    E.cname
                                FROM      Enrolled E
                                GROUP BY E.cname
                                HAVING    COUNT (*) >= 5)
```

2Q) Illustrate the relational employee (name, salary, deptno), department (deptno, deptname, address). Solve which query cannot be expressed using the basic relational algebra operations.

The sum of all employees' salaries
The six basic operators of relational algebra are the selection(σ), the projection(π), the Cartesian product (x) (also called the cross product or cross join), the set union (U), the set difference (-), and the rename (p). These six operators are fundamental in the sense that none of them can be omitted without losing expressive power. Many other operators have been defined in terms of these six. Among the most important are set intersection, division, and the natural join, but aggregation is not possible with these basic relational algebra operations. So, we cannot run sum of all employees' salaries with the six operations

3Q) Discuss Query in relational algebra to find second highest salary of employee from employee relation.



Consider below simple table:

```
Name     Salary
---------------
abc     100000
bcd     1000000
efg     40000
ghi     500000
```

How to find the employee whose salary is second highest. For example, in above table, "ghi" has the second highest salary as 500000.

Simple query:

```
select * from employee where salary=(select Max(salary) from
employee);
```

We can nest the above query to find the second largest salary.

```
select * from employee
group by salary
order by  salary desc limit 1,1;
```

Note that instead of nesting for second, third, etc largest salary, we can find nth salary using general query like in MySQL:

```
SELECT salary
FROM employee
ORDER BY salary desc limit n-1,1
```

4Q) Consider the following schema given. The primary keys are underlined. Sailors (Sailor-ID, sailor-name, sailor-rating, sailor-age) Boats (boat-ID, boat-name, boat color), Reserves (Sailor ID, boat-ID, day) Write queries in relational algebra.

1. **Find the names of sailors who have reserved boat number 120**
2. **Find the names of sailors who have reserved a green boat**
3. **Find the names of sailors who have not reserved a green boat.**
4. **Find the names of sailors with the highest rating**

Sailors(sid: integer, sname: string, rating: integer, age: real);
Boats(bid: integer, bname: string, color: string);
Reserves(sid: integer, bid: integer, day: date).

```
SQL> select * from sailors;

       SID SNAME                    RATING        AGE
---------- -------------------- ---------- ----------
        22 Dustin                        7         45
        29 Brutus                        1         33
        31 Lubber                        8       55.5
        32 Andy                          8       25.5
        58 Rusty                        10         35
        64 Horataio                      7         35
        71 Zorba                        10         16
        74 Horataio                      9         35
        85 Art                           3       25.5
        95 Bob                           3       63.5
10 rows selected.
```

```
SQL> select * from reserves;
       SID        BID DAY
---------- ---------- ---------
        22        101 10-OCT-98
        22        102 10-OCT-98
        22        103 08-OCT-98
        22        104 07-OCT-98
        31        102 10-NOV-98
        31        103 06-NOV-98
        31        104 12-NOV-98
        64        101 05-SEP-98
        64        102 08-SEP-98
        74        103 08-SEP-98
10 rows selected.
```

```
SQL> select * from boats;

       BID BNAME                COLOR
---------- -------------------- --------------------
       101 Interlake            blue
       102 Interlake            red
       103 Clipper              green
       104 Marine               red
```

If boat Number is 103.Then find the name of sailors?

```
select s.sname
from sailors s,reserves r
where s.sid=r.sid and r.bid=103;

Output:

SNAME
--------------------
Dustin
Lubber
Horataio
```

What is the color of boat reverse by Lubber?

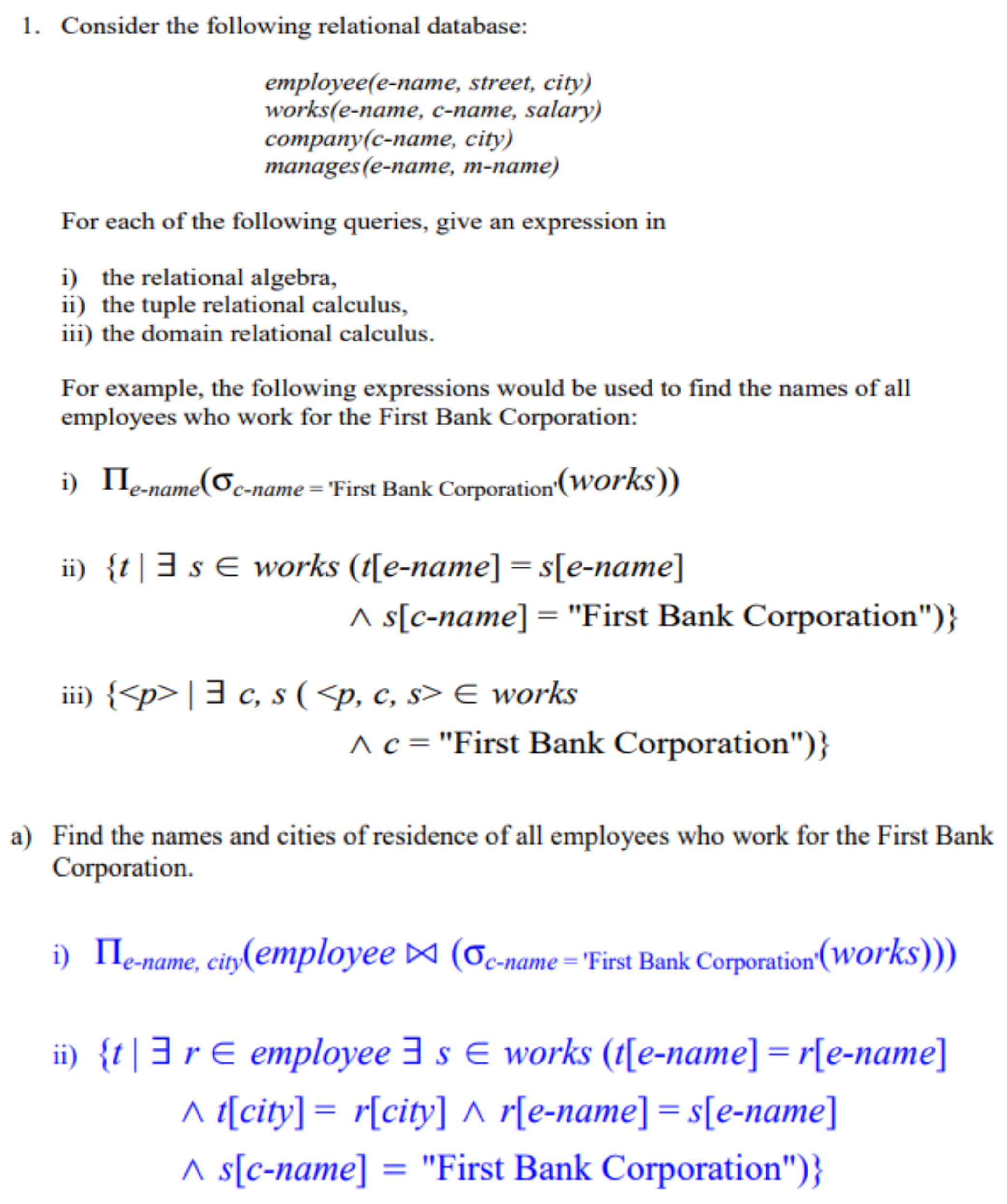
```
select b.color
from boats b,sailors s,reserves r
where s.sid=r.sid and b.bid=r.bid and s.sname='Lubber';

COLOR
--------------------
red
green
red
```

Find the sids of sailors with age over 20 who have not reserved a red boat.

```
select s.sid,s.sname
from sailors s,boats b,reserves r
where s.sid=r.sid and b.bid=r.bid and s.age>20 and b.color!='red';

       SID SNAME
---------- --------------------
        22 Dustin
        22 Dustin
        31 Lubber
        64 Horataio
        74 orataio
```

5Q) Consider the following database. Employee(employee-name, street, city), works (employee-name, company-name, salary), company (company-name, city), Manager (Employee-name, manager-name).

1. **Given an expression in the relational algebra, the tuple relational calculus, for the following query.**
2. **Find the names of all employees who work for the estate bank.**



1. Consider the following relational database:

employee(e-name, street, city)
works(e-name, c-name, salary)
company(c-name, city)
manages(e-name, m-name)

For each of the following queries, give an expression in

i) the relational algebra,
ii) the tuple relational calculus,
iii) the domain relational calculus.

For example, the following expressions would be used to find the names of all employees who work for the First Bank Corporation:

i) $\Pi_{e\text{-}name}(\sigma_{c\text{-}name = \text{'First Bank Corporation'}}(works))$

ii) $\{t \mid \exists\ s \in works\ (t[e\text{-}name] = s[e\text{-}name]$
$\wedge\ s[c\text{-}name] = \text{"First Bank Corporation"})\}$

iii) $\{<p> \mid \exists\ c, s\ (<p, c, s> \in works$
$\wedge\ c = \text{"First Bank Corporation"})\}$

a) Find the names and cities of residence of all employees who work for the First Bank Corporation.

i) $\Pi_{e\text{-}name,\ city}(employee \bowtie (\sigma_{c\text{-}name = \text{'First Bank Corporation'}}(works)))$

ii) $\{t \mid \exists\ r \in employee\ \exists\ s \in works\ (t[e\text{-}name] = r[e\text{-}name]$
$\wedge\ t[city] = r[city]\ \wedge\ r[e\text{-}name] = s[e\text{-}name]$
$\wedge\ s[c\text{-}name] = \text{"First Bank Corporation"})\}$

6Q) Define the RA expressions for the following queries. Sailor Schema (Sailor ID, sailor name, rating, age), Reserves (Sailor ID, boat ID, day), boat schema (boat ID, boat name, color).

1. **Find the names of sailors who have reserved boat name 103.**
2. **Find the sailor ID of sailors who have reserved a red boat.**
3. **Find the colors of boats reserved by the sailor rubber.**
4. **Find the names of sailors who have reserved a red boat.**

(Q1) Find the names of sailors who have reserved boat 103.

This query can be written as follows:

$$\pi_{sname}((\sigma_{bid=103}Reserves) \bowtie Sailors)$$

We first compute the set of tuples in Reserves with $bid = 103$ and then take the natural join of this set with Sailors. This expression can be evaluated on instances of Reserves and Sailors. Evaluated on the instances $R2$ and $S3$, it yields a relation that contains just one field, called *sname*, and three tuples $\langle Dustin \rangle$, $\langle Horatio \rangle$, and $\langle Lubber \rangle$. (Observe that there are two sailors called Horatio, and only one of them has reserved a red boat.)

(Q2) Find the names of sailors who have reserved a red boat.

$$\pi_{sname}((\sigma_{color='red'}Boats) \bowtie Reserves \bowtie Sailors)$$

This query involves a series of two joins. First we choose (tuples describing) red boats. Then we join this set with Reserves (natural join, with equality specified on the *bid* column) to identify reservations of red boats. Next we join the resulting intermediate relation with Sailors (natural join, with equality specified on the *sid* column) to retrieve the names of sailors who have made reservations of red boats. Finally, we project the sailors' names. The answer, when evaluated on the instances $B1$, $R2$ and $S3$, contains the names Dustin, Horatio, and Lubber.

An equivalent expression is:

$$\pi_{sname}(\pi_{sid}((\pi_{bid}\sigma_{color='red'}Boats) \bowtie Reserves) \bowtie Sailors)$$

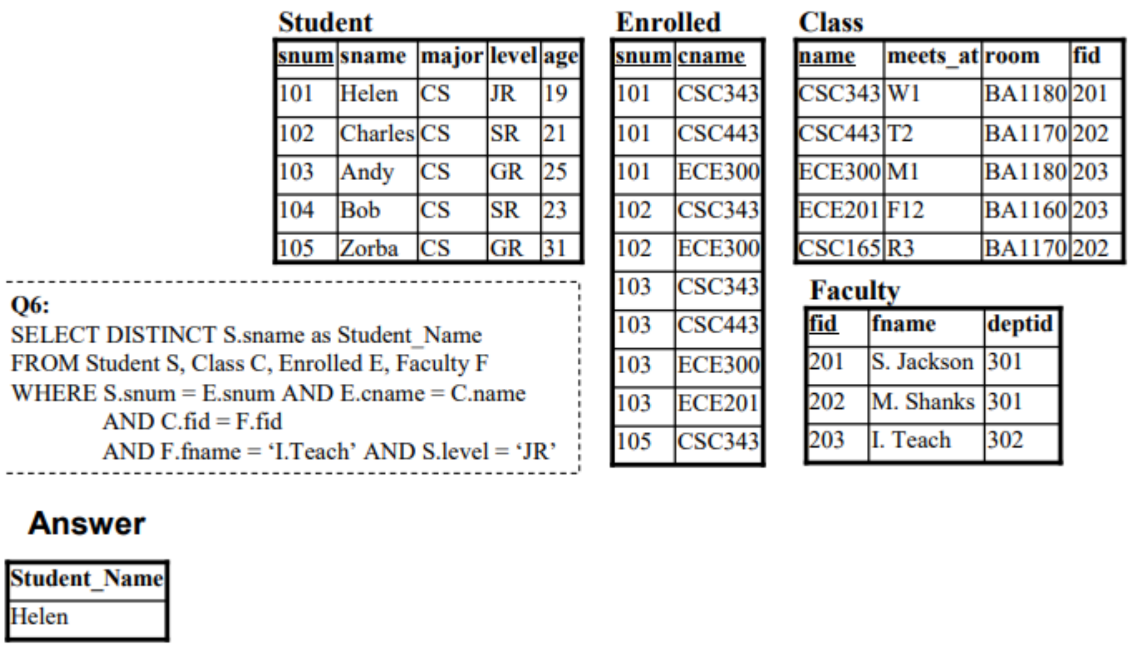
(Q3) Find the colors of boats reserved by Lubber.

$$\pi_{color}((\sigma_{sname='Lubber'}Sailors) \bowtie Reserves \bowtie Boats)$$

This query is very similar to the query we used to compute sailors who reserved red boats. On instances $B1$, $R2$, and $S3$, the query will return the colors gren and red.

7Q) Observe the following relational database, give the expressions in RA. Student (stuno, sstuname, major, level, age) class (classname, meets at room, fid), faculty(fid, fname, deptID)

1. **Find the names of all juniors (level = JR) who are enrolled in a class taught by I.Teach.**
2. **Find the age of the oldest student who is either a history major or is enrolled in a course taught by I>Tech.**



Student

snum	sname	major	level	age
101	Helen	CS	JR	19
102	Charles	CS	SR	21
103	Andy	CS	GR	25
104	Bob	CS	SR	23
105	Zorba	CS	GR	31

Enrolled

snum	cname
101	CSC343
101	CSC443
101	ECE300
102	CSC343
102	ECE300
103	CSC343
103	CSC443
103	ECE300
103	ECE201
105	CSC343

Class

name	meets_at	room	fid
CSC343	W1	BA1180	201
CSC443	T2	BA1170	202
ECE300	M1	BA1180	203
ECE201	F12	BA1160	203
CSC165	R3	BA1170	202

Faculty

fid	fname	deptid
201	S. Jackson	301
202	M. Shanks	301
203	I. Teach	302

Q6:

```
SELECT DISTINCT S.sname as Student_Name
FROM Student S, Class C, Enrolled E, Faculty F
WHERE S.snum = E.snum AND E.cname = C.name
        AND C.fid = F.fid
        AND F.fname = 'I.Teach' AND S.level = 'JR'
```

Answer

Student_Name
Helen

8Q) Sailor schema (Sailor ID, sailor name, rating, Age), reserves(sailor-ID, boat ID, day) boat schema(boat ID, boatname, color)

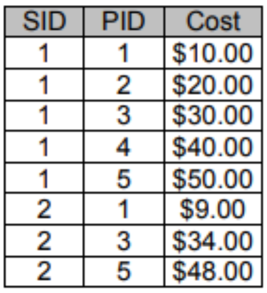
1. **Find the age of the youngest sailor for each rating level.**
2. **Find the age of the youngest sailor who is eligible to vote for each rating level with at least two such sailors?**
3. **Find the no. of reservations for each red boat?**

4. **Find the average age of sailors for each rating level that is at least 2 sailors.**

9Q) How the statement “the SID’s of suppliers who supply some red or green parts” can be represented in the form of relational algebra and tuple relational algebra and tuple relational calculus from the above relations.

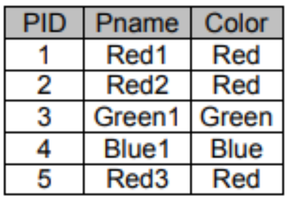
Suppliers scheme: **Suppliers (SID: INTEGER,**
Sname: STRING,
Address: STRING)
Parts (PID: INTEGER,
Pname : STRING
Color: STRING)
Catalog (SID: INTEGER,
PID: INTEGER,
Cost: REAL)

Catalog

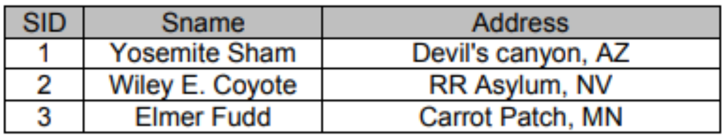


SID	PID	Cost
1	1	$10.00
1	2	$20.00
1	3	$30.00
1	4	$40.00
1	5	$50.00
2	1	$9.00
2	3	$34.00
2	5	$48.00

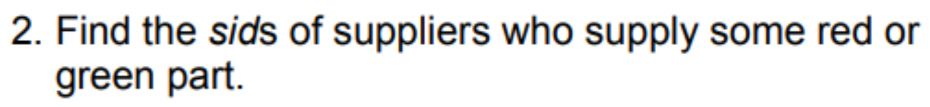
Parts



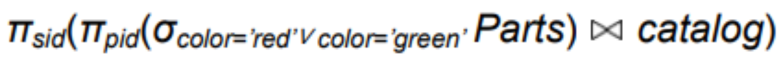
PID	Pname	Color
1	Red1	Red
2	Red2	Red
3	Green1	Green
4	Blue1	Blue
5	Red3	Red



SID	Sname	Address
1	Yosemite Sham	Devil's canyon, AZ
2	Wiley E. Coyote	RR Asylum, NV
3	Elmer Fudd	Carrot Patch, MN

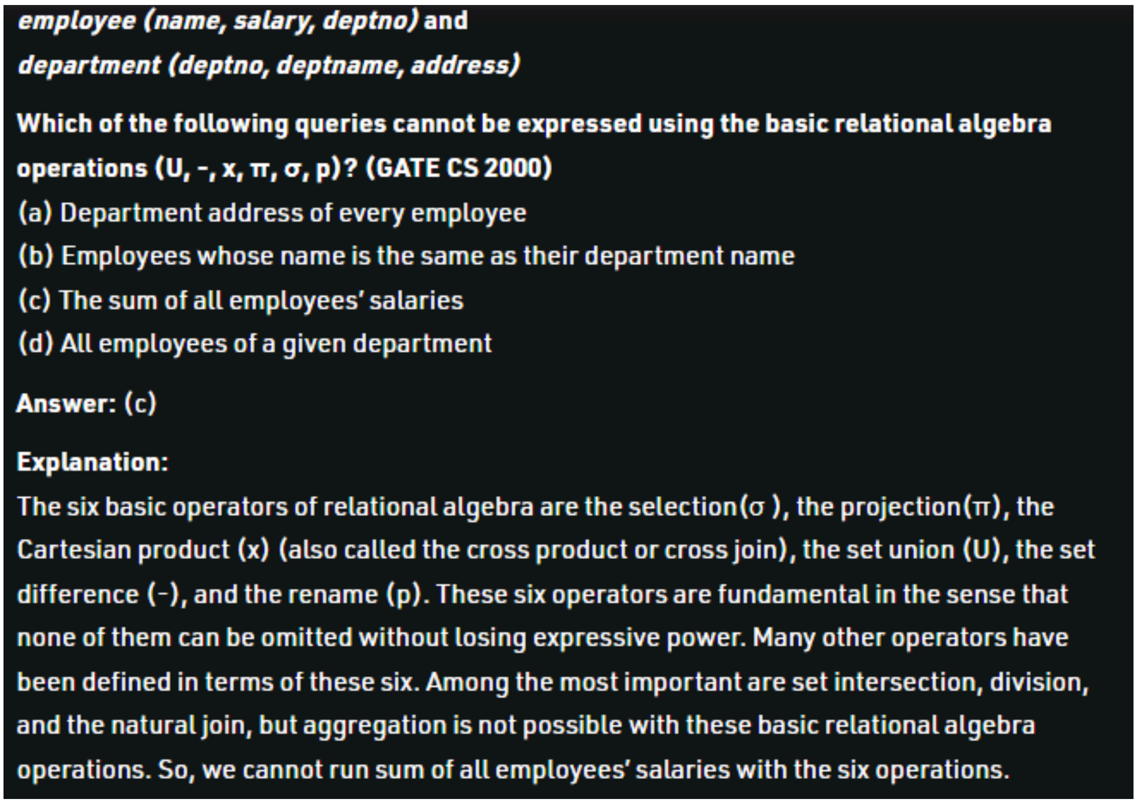


2. Find the *sid*s of suppliers who supply some red or green part.



$$\pi_{sid}(\pi_{pid}(\sigma_{color='red' \vee color='green'} Parts) \bowtie catalog)$$

10Q) Given the relations Employee(name, salary, dept no), department(deptno, deptname, address) Solve which query cannot be expressed using the basic SQL operations.



employee (name, salary, deptno) **and**

department (deptno, deptname, address)

Which of the following queries cannot be expressed using the basic relational algebra operations (U, -, x, π, σ, p)? (GATE CS 2000)

(a) Department address of every employee

(b) Employees whose name is the same as their department name

(c) The sum of all employees' salaries

(d) All employees of a given department

Answer: (c)

Explanation:

The six basic operators of relational algebra are the selection(σ), the projection(π), the Cartesian product (x) (also called the cross product or cross join), the set union (U), the set difference (-), and the rename (p). These six operators are fundamental in the sense that none of them can be omitted without losing expressive power. Many other operators have been defined in terms of these six. Among the most important are set intersection, division, and the natural join, but aggregation is not possible with these basic relational algebra operations. So, we cannot run sum of all employees' salaries with the six operations.

Part - B

1Q) Illustrate different set operations in Relational algebra with an example.

The set operation are mainly categorized into the following:

1. Union operation
2. Intersection operation
3. Set difference or Minus operation

UNION Operation:

Notation:
A ∪ S
where, A and S are the relations,

symbol '∪' is used to denote the Union operator.

The result of Union operation, which is denoted by A ∪ S, is a relation that basically includes all the tuples that are present in A or in S, or in both, eliminating the duplicate tuples.

INTERSECTION Operation:

Notations:
A ∩ S
where, A and S are the relations,

symbol '∩' is used to denote the Intersection operator.

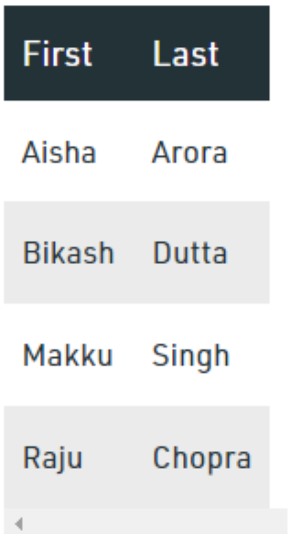
The result of Intersection operation, which is denoted by A ∩ S, is a relation that basically includes all the tuples that are present in both A and S.

MINUS (or SET DIFFERENCE) Operation:

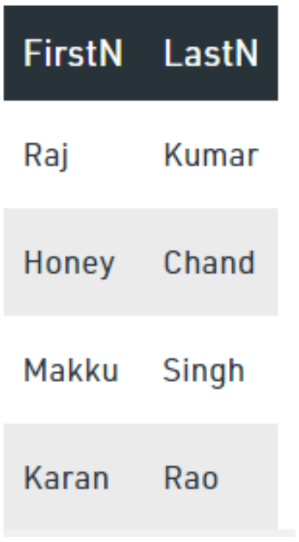
Notations:
A - S

where, A and S are the relations,
symbol ‘ – ’ is used to denote the Minus operator.
The result of Intersection operation, which is denoted by A – S, is a relation that basically includes all the tuples that are present in A but not in S.

Consider a relation Student(FIRST, LAST) and Faculty(FIRSTN, LASTN) given below :



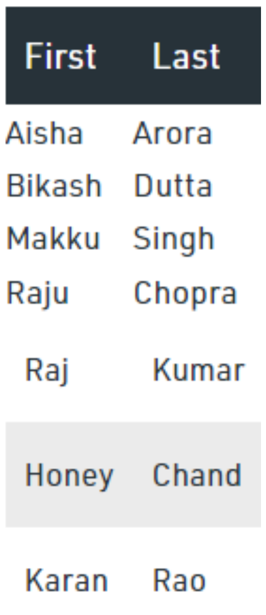
First	Last
Aisha	Arora
Bikash	Dutta
Makku	Singh
Raju	Chopra



FirstN	LastN
Raj	Kumar
Honey	Chand
Makku	Singh
Karan	Rao

1. Student UNION Faculty :

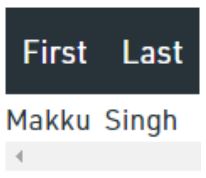
Student ∪ Faculty



First	Last
Aisha	Arora
Bikash	Dutta
Makku	Singh
Raju	Chopra
Raj	Kumar
Honey	Chand
Karan	Rao

2. Student INTERSECTION Faculty :

Student ∩ Faculty



First	Last
Makku	Singh

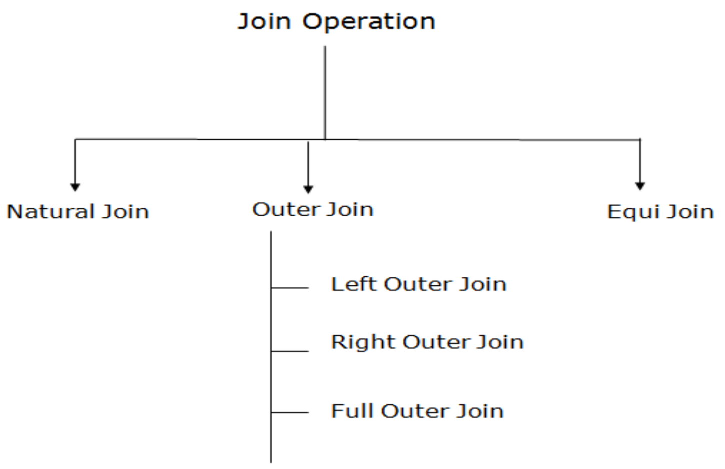
3. Student MINUS Faculty :

Student - Faculty

First	Last
Aisha	Arora
Bikash	Dutta
Raju	Chopra
Raj	Kumar
Honey	Chand
Karan	Rao

2Q) Define Join. Explain different types of joins in relational algebra.

In DBMS, a join statement is mainly used to combine two tables based on a specified common field between them. In terms of Relational algebra, it is the cartesian product of two tables followed by the selection operation. Thus, we can execute the product and selection process on two tables using a single join statement.



Refer all operations from DBMS Join Operation - javatpoint

3Q) Discuss Tuple Relational calculus in detail.

Tuple Relational Calculus is a non-procedural query language unlike relational algebra. Tuple Calculus provides only the description of the query but it does not provide the methods to solve it. Thus, it explains what to do but not how to do it. In Tuple Calculus, a query is expressed as

`{t| P(t)}`

where `t = resulting tuples,`

`P(t) = known as Predicate` and these are the conditions that are used to fetch t .

Thus, it generates a set of all tuples t, such that Predicate P(t) is true for t.

P(t) may have various conditions logically combined with OR (∨), AND (∧), NOT(¬).

It also uses quantifiers:

∃ t ∈ r (Q(t)) = ”there exists” a tuple in t in relation r such that predicate Q(t) is true.

∀ t ∈ r (Q(t)) = Q(t) is true “for all” tuples in relation r.

For example:

```
{ T.name | Author(T) AND T.article = 'database' }
```

OUTPUT: This query selects the tuples from the AUTHOR relation. It returns a tuple with 'name' from the Author who has written an article on 'database'.

TRC (tuple relational calculus) can be quantified. In TRC, we can use Existential (∃) and Universal Quantifiers (∀).

4Q) Define the difference between Relational Algebra and Relational Calculus

Relational Algebra	**Relational Calculus**
It is a Procedural Language.	While Relational Calculus is Declarative Language
Relational Algebra means how to	While Relational Calculus means what

obtain the results.	results we have to obtain
The order is specified in which the operations have to be performed.	The order is not specified
Relational Algebra is independent of the domain	Can be domain dependent.
This is a language in which queries can be expressed but the queries should also be expressed in relational calculus to be relationally complete	Database language to be relationally complete., the query written in ikt must be expressed in relational calculus

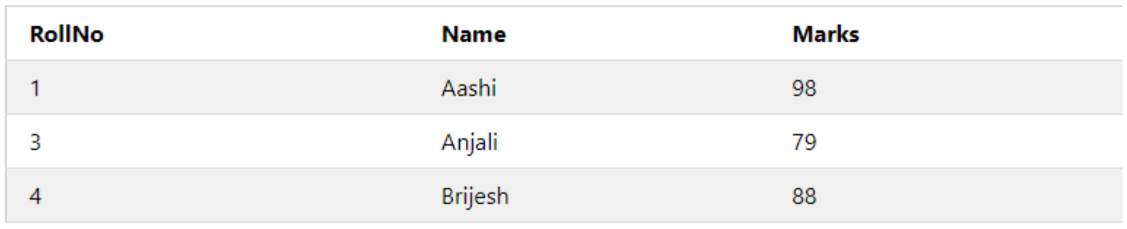
5Q) Describe Extended relational operations with examples.

Extended operators are those operators which can be derived from basic operators. There are mainly three types of extended operators in Relational Algebra:

- Join
- Intersection
- Divide

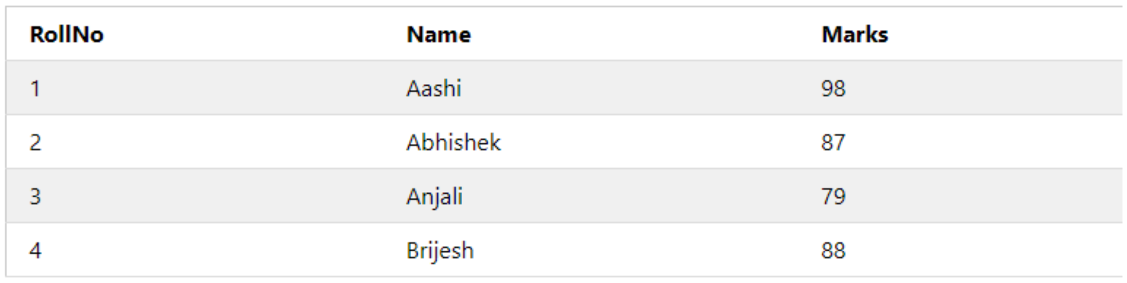
Let us consider two tables named as A and B:

A –



RollNo	Name	Marks
1	Aashi	98
3	Anjali	79
4	Brijesh	88

B –

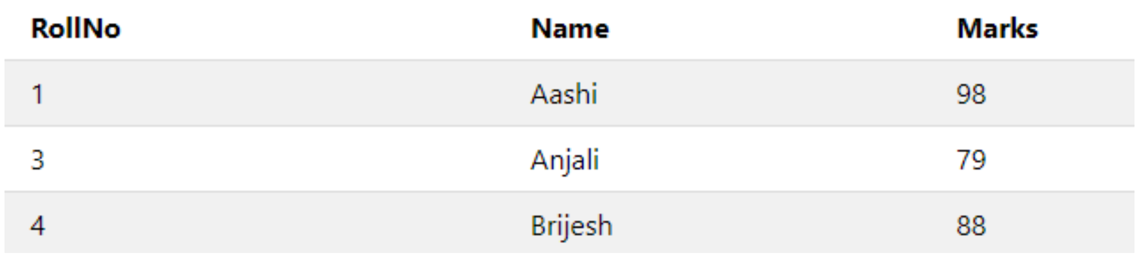


RollNo	Name	Marks
1	Aashi	98
2	Abhishek	87
3	Anjali	79
4	Brijesh	88

- Intersection
  Intersection works on the relation as 'this and that'. In relational algebra, A ∩ B returns a relation instance that contains every tuple that occurs in relation to instance A and relation instance B (both together). Here, A and B need to be union-compatible, and the schema of both result and A must be identical.

Syntax:

```
SELECT * FROM A INTERSECT SELECT * FROM B;
```



RollNo	Name	Marks
1	Aashi	98
3	Anjali	79
4	Brijesh	88

- Divide

  The divide operator is used for the queries that contain the keyword ALL.

  For e.g. – Find all the students who have chosen additional subjects Machine Learning and Data Mining.

  Student –

Student Name	Subject
Ashish	Machine Learning
Ashish	Data Mining
Shivam	Network Security
Shivam	Data Mining
Tarun	Network Security
Tarun	Machine Learning
Yash	Machine Learning
Yash	Data Mining

Subject –

Student Name
Machine Learning
Data Mining

Output: Student ÷ Subject

Student
Ashish
Yash

Read join from here

6Q) Discuss procedural language in SQL

PL/SQL (Procedural Language/Structured Query Language) is a block-structured language extension of SQL that consists of statements and language elements that can be used to implement procedural logic in SQL statements which enable developers to combine the power of SQL with Procedural statements. All the statements of a block are passed to the Oracle engine all at once which increases processing speed and decreases the traffic. SQL PL provides statements for declaring variables and condition handlers, assigning values to variables, and implementing procedural language.

Features of PL/SQL:

- PL/SQL is basically a procedural language, which provides the functionality of decision making, iteration and many more features of procedural programming languages.
- PL/SQL can execute a number of queries in one block using single command.
- One can create a PL/SQL unit such as procedures, functions, packages, triggers, and types, which are stored in the database for reuse by applications.
- PL/SQL provides a feature to handle the exception which occurs in PL/SQL block known as exception handling block.
- Applications written in PL/SQL are portable to computer hardware or operating system where Oracle is operational.
- PL/SQL Offers extensive error checking.

7Q) Discuss the structure of the query in TRC with an example.

Tuple Relational Calculus is a non-procedural query language unlike relational algebra. Tuple Calculus provides only the description of the query but it does not provide the methods to solve it. Thus, it explains what to do but not how to do.

In Tuple Calculus, a query is expressed as : `{t| P(t)}`

where t = resulting tuples,

P(t) = known as Predicate and these are the conditions that are used to fetch t

Thus, it generates a set of all tuples t, such that Predicate P(t) is true for t.

P(t) may have various conditions logically combined with OR (∨), AND (∧), NOT(¬).

It also uses quantifiers:

`∃ t ∈ r (Q(t)) = ”there exists” a tuple in t in relation r such that predicate Q(t) is true.`

`∀ t ∈ r (Q(t)) = Q(t) is true “for all” tuples in relation r.`

Example:

Find the loan number, branch, amount of loans of greater than or equal to 10000 amount.

Ans: `{t| t ∈ loan ∧ t[amount]>=10000}`

8Q) Illustrate a query in TRC to find the names of sailors who have reserved both a red and green boat? Write a query in TRC to find the names of sailors who have reserved all boats?

9Q) Write a query in TRC to find the names of sailors who have reserved both a red and green boat? Write a query in TRC to find the names of sailors who have not reserved a red boat?

The following is a TRC query for finding the name of sailors who have reserved both red and green boats.

10Q) r Write a TRC query to find the names of sailors who have reserved boat 103?

This query can be written as follows:

$$\pi_{sname}((\sigma_{bid=103}Reserves) \bowtie Sailors)$$

TRC query:

$\{P \mid \exists S \in Sailors\ \exists R \in Reserves(R.sid = S.sid \wedge R.bid = 103 \wedge P.sname = S.sname)\}$

This query can be read as follows: "Retrieve all sailor tuples for which there exists a tuple in Reserves, having the same value in the *sid* field, and with *bid* = 103." That is, for each sailor tuple, we look for a tuple in Reserves that shows that this sailor has reserved boat 103. The answer tuple P contains just one field, *sname*.

11Q) Let R = (ABC) and S = (DEF) let r(R) and s(S) both relations on schema R and S. Give an expression in the tuple relational calculus that is equivalent to each of the following.ρ B=19(r) A,F,(ρC=D(r×s)) r $\bigcap$ s

12Q) Sketch the following schema instruction (ID, name, dept name), teaches (ID, course ID, sec ID, semester, year), section (course ID, sec ID, semester, year), student (ID, name, dept name), takes (ID, course ID, sec ID, semester, year, grade). Write the following query in RA, TRC and DRC. Find the names of the instructors not teaching any course.

13Q)r Find the names of the sailors who have reserved a green boat.

```
SELECT S.sname
FROM Sailors S, Boats B, Reserves R
WHERE B.color='green' AND B.bid=R.bid AND S.sid = R.sid
```

14Q) r Describe the sides of sailors who have reserved red and green boats. Find sides of all sailors who have reserved a red boat but not a green boat.

```
SELECT S1.sname FROM Sailors S1, Reserves R1, Boats B1 WHERE
S1.sid=R1.sid AND R1.bid=B1.bid AND B1.color=`red'
INTERSECT
SELECT S2.sname FROM Sailors S2, Reserves R2, Boats B2 WHERE
S2.sid=R2.sid AND R2.bid=B2.bid AND B2.color=`green'

SELECT S1.sid FROM Sailors S1, Reserves R1, Boats B1 WHERE
S1.sid=R1.sid AND R1.bid=B1.bid AND B1.color=`red'
EXCEPT
SELECT S2.sid FROM Sailors S2, Reserves R2, Boats B2 WHERE
S2.sid=R2.sid AND R2.bid=B2.bid AND B2.color=`green'
```

15Q)Describe sides of all sailors who have a rating of 10 or reserved boat 104 find a sailor whose rating is better than every sailor called Horatio.

```
(SELECT sid FROM s WHERE rating>=10) UNION (SELECT sid FROM r
WHERE bid=104)

SELECT sname FROM s WHERE rating > all ( SELECT rating FROM s s2
WHERE s2.sname=`Horatio')
```

16Q) Find the sailors with the highest rating. Find the names of all branches in the loan relation.

17Q) Define set operations with syntax and examples.

ans)

The SQL Set operation is used to combine the two or more SQL SELECT statements.

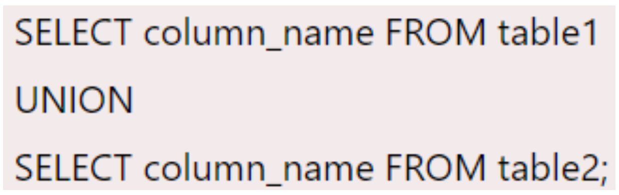
Types of Set Operation

1. Union
2. UnionAll
3. Intersect
4. Minus

1. Union

- The SQL Union operation is used to combine the result of two or more SQL SELECT queries.
- In the union operation, all the number of data type and columns must be same in both the tables on which the UNION operation is being applied.
- The union operation eliminates the duplicate rows from its resultset.

Syntax:

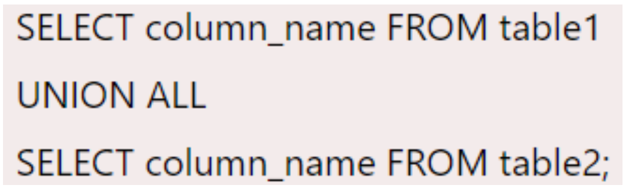


```
SELECT column_name FROM table1
UNION
SELECT column_name FROM table2;
```

2. Union All

Union All operation is equal to the Union operation. It returns the set without removing duplication and sorting the data.

Syntax:

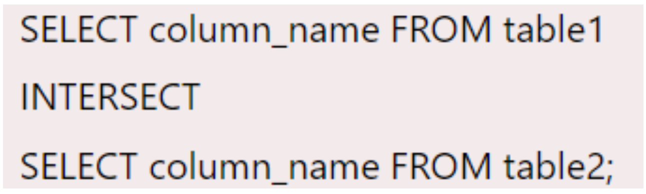


```
SELECT column_name FROM table1
UNION ALL
SELECT column_name FROM table2;
```

3. Intersect

- It is used to combine two SELECT statements. The Intersect operation returns the common rows from both the SELECT statements.

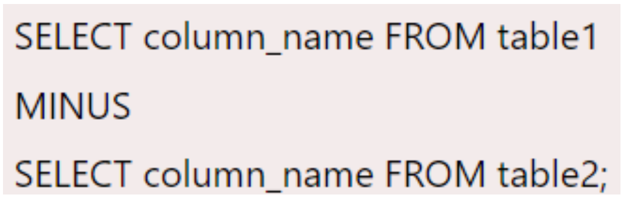
- In the Intersect operation, the number of data type and columns must be the same.
- It has no duplicates and it arranges the data in ascending order by default.

Syntax: 

```
SELECT column_name FROM table1
INTERSECT
SELECT column_name FROM table2;
```

4. Minus

- It combines the result of two SELECT statements. Minus operator is used to display the rows which are present in the first query but absent in the second query.
- It has no duplicates and data arranged in ascending order by default.

Syntax: 

```
SELECT column_name FROM table1
MINUS
SELECT column_name FROM table2;
```

For examples please refer to :
[DBMS SQL Set Operation - javatpoint]

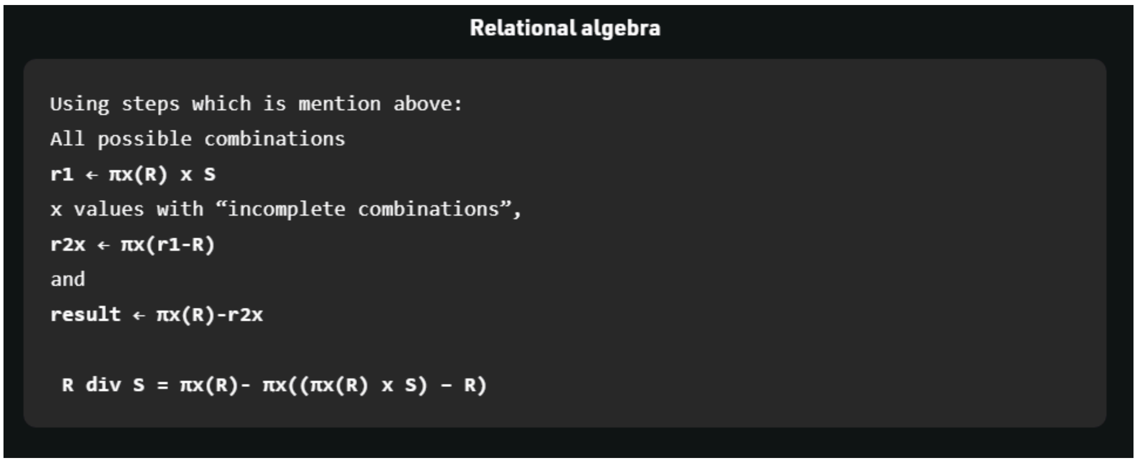
18Q) Write about Division operation in relational algebra with an example.

Division is typically required when you want to find out entities that are interacting with all entities of a set of different type entities.

The division operator is used when we have to evaluate queries which contain the keyword ‘all’.
Some instances where division operator is used are:

- Which person has an account in all the banks of a particular city?
- Which students have taken all the courses required to graduate?

In all these queries, the description after the keyword ‘all’ defines a set which contains some elements and the final result contains those units who satisfy these requirements.



Relational algebra

```
Using steps which is mention above:
All possible combinations
r1 ← πx(R) x S
x values with “incomplete combinations”,
r2x ← πx(r1-R)
and
result ← πx(R)-r2x

 R div S = πx(R)- πx((πx(R) x S) – R)
```

Another way you can identify the usage of division operator is by using the logical implication of if...then. In context of the above two examples, we can see that the queries mean that,

1. If there is a bank in that particular city, that person must have an account in that bank.
2. If there is a course in the list of courses required to be graduated, that person must have taken that course.

19Q) Write about join operations with syntax and examples

SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

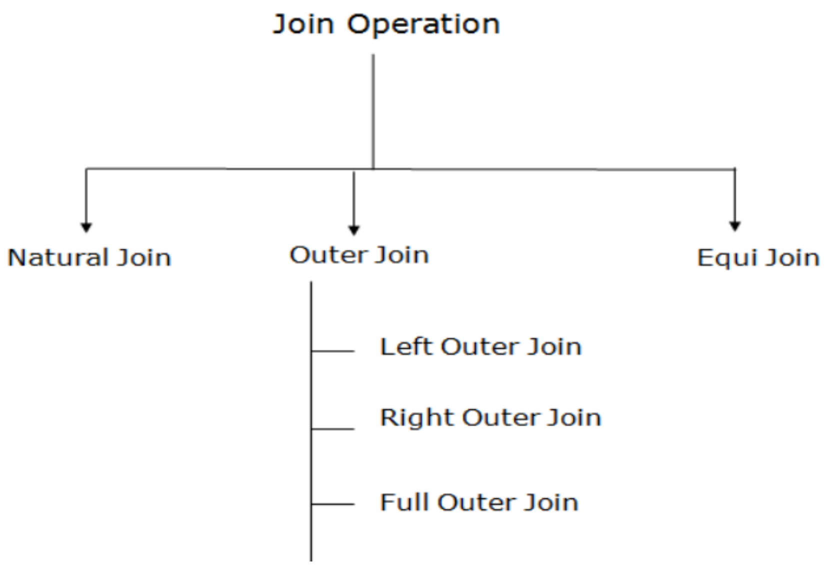
INNER JOIN

LEFT JOIN

RIGHT JOIN

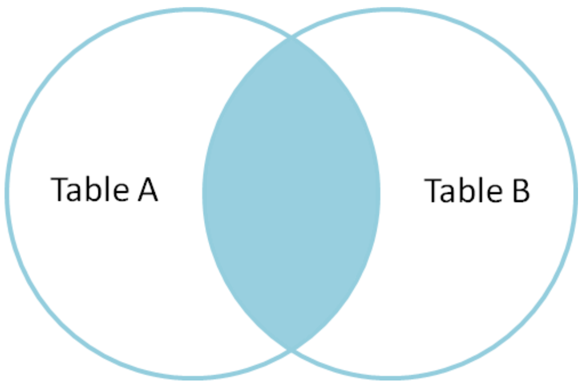
FULL JOIN

A Join operation combines related tuples from different relations, if and only if a given join condition is satisfied. It is denoted by ⋈.



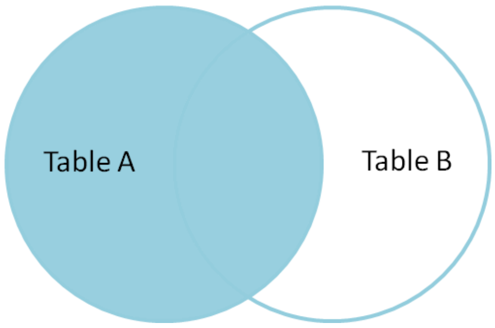
A. INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.



B. LEFT JOIN

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain null. LEFT JOIN is also known as LEFT OUTER JOIN.



SQL | Join (Inner, Left, Right and Full Joins) - GeeksforGeeks

Refer to this: DBMS SQL Joins - javatpoint

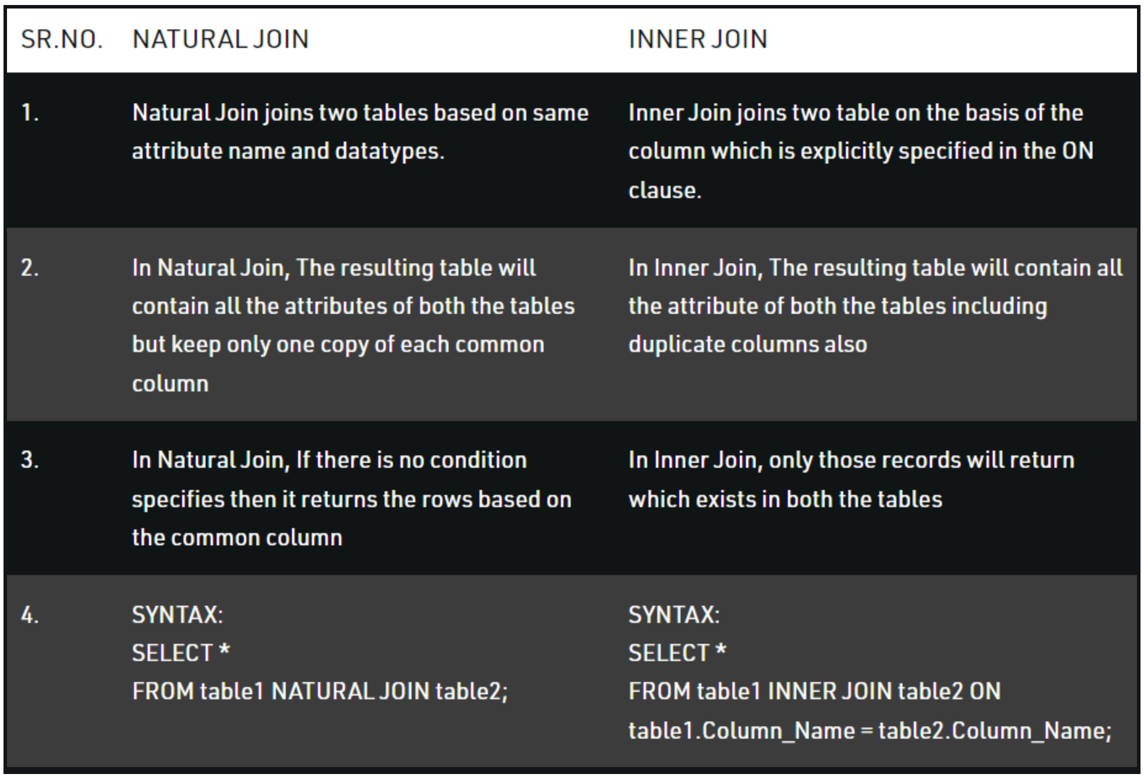
20Q) Differentiate natural join and inner join operations with examples.

1. Natural Join :

Natural Join joins two tables based on the same attribute name and datatypes. The resulting table will contain all the attributes of both the tables but keep only one copy of each common column.

2. Inner Join :

Inner Join joins two tables on the basis of the column which is explicitly specified in the ON clause. The resulting table will contain all the attributes from both the tables including the common column also.



SR.NO.	NATURAL JOIN	INNER JOIN
1.	Natural Join joins two tables based on same attribute name and datatypes.	Inner Join joins two table on the basis of the column which is explicitly specified in the ON clause.
2.	In Natural Join, The resulting table will contain all the attributes of both the tables but keep only one copy of each common column	In Inner Join, The resulting table will contain all the attribute of both the tables including duplicate columns also
3.	In Natural Join, If there is no condition specifies then it returns the rows based on the common column	In Inner Join, only those records will return which exists in both the tables

4.	SYNTAX:
	SELECT *
	FROM table1 NATURAL JOIN table2;
		SYNTAX:
		SELECT *
		FROM table1 INNER JOIN table2 ON
		table1.Column_Name = table2.Column_Name;

Part - C

1Q) Describe a create statement with an example.

2Q) Define DML statements in SQL. Give an example.

It is used for accessing and manipulating data in a database. It handles user requests.

Select: It is used to retrieve data from a database.

Insert: It is used to insert data into a table.

Update: It is used to update existing data within a table.

Delete: It is used to delete all records from a table.

Merge: It performs UPSERT operation, i.e., insert or update operations.

Call: It is used to call a structured query language or a Java subprogram.

3Q) Discuss various Aggregate functions used in SQL.

An aggregate function in SQL returns one value after calculating multiple values of a column. We often use aggregate functions with the GROUP BY and HAVING clauses of the SELECT statement.

Various types of SQL aggregate functions are:
Count()
Sum()
Avg()
Min()
Max()

4Q) Define the primary key.

5Q) State the syntax of foreign key constraint.

6Q) What are the data types in SQL?

7Q) Write a SQL statement to find employees whose commission is greater than their salaries.

8Q) Write a SQL statement to find the employees who are not clerks, analysts or salesmen.

9Q) Write a SQL statement to display the names of all the employees and positions where the string AR occurs in the name.

10Q) List out all classes in the select statement.

11Q) Define redundancy and its problems.

12Q) Define functional dependency. Why are some functional dependencies trivial?

13Q) Discuss normalization.

14Q) Differentiate between trivial and non-trivial dependencies.

15Q) If relation R consists of only simple candidate keys then R should be in which normal form?

16Q) Define the First Normal form

17Q) Define the Second Normal form

18Q) Define the Third Normal form

19Q) Define the Fourth Normal form

20Q) Identify the normal forms of the relation R. R(ABCD) FD: {A → B, B → C}