COMPUTER SCIENCE

Database Management System

Query Language

Lecture_7

Vijay Agarwal sir


TOPICS TO BE COVERED
01
SQL Operators
02
TRC

SQL

ANY

ALL

IN | NOT IN

EXIST | NOT EXIST

Supplier (Sid, Sname, Rating)
Parts (Pid, Pname, Color)
Catalog (Sid, Pid, Cost)

Supplier

Sid	Sname	Rating
S_1		
S_2		
S_3		
S_4		
S_5		

Catalog

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_3
S_4	P_4

Parts

Pid	Color
P_1	Red
P_2	Green
P_3	Red
P_4	Yellow

Q. Retrieve Sid of the Supplier who supplied some Red Color Parts?

Query I:

```
Select    Sid
From      Catalog C, Parts P
WHERE     P.Pid = C.Pid
          Color = Red
```

output →

S_1
S_2

Query II:

Select Sid
From Catalog

WHERE Pid = (Select Pid
FROM Parts
WHERE Color = Red)

Pid
P_1
P_3

→ One to many
Comparison not
Directly allowed

Query III:

Select Sid
FROM Catalog
WHERE Pid

IN

(Select Pid
FROM Parts
WHEREColor = Red)

Pid
P_1
P_2

Output

Sid
S_1
S_2

NOT IN

Output

Sid
S_1
S_4

EXISTS: (Checks): Return True if Inner Query Result Not Empty

EXISTS: (Checks): Return True if Inner Query Result Not Empty

NOT EXIST: Return True if Inner Result Empty

Correlated Nested Query: Inner Query Using attribute defined in Outer Query .

```
Select   C.Sid
FROM     Catalog C
WHERE EXISTS
(Select *
 FROM Part P
 WHERE P.Pid = C.Pid)
```

In Inner Query Catalog [outer query table Attribute using.

Inner Query .

Corelated Nested Query

```
Select   C.sid
FROM     Catalog
WHERE EXISTS ( Select*
               FROM Parts P
               WHEREP P.Pid = C.Pid
               AND Color = Red )
```

Inner Query using Attributes defined in the Outer Query

Catalog . Pid

TOP → Bottom → Top.

i = 1 ⟶ j = 1 to m

i = 2 ⟶ j = 1 to m

⋮

Nested Queries

(Independent)
Normal Nest Query

Inner → Outer

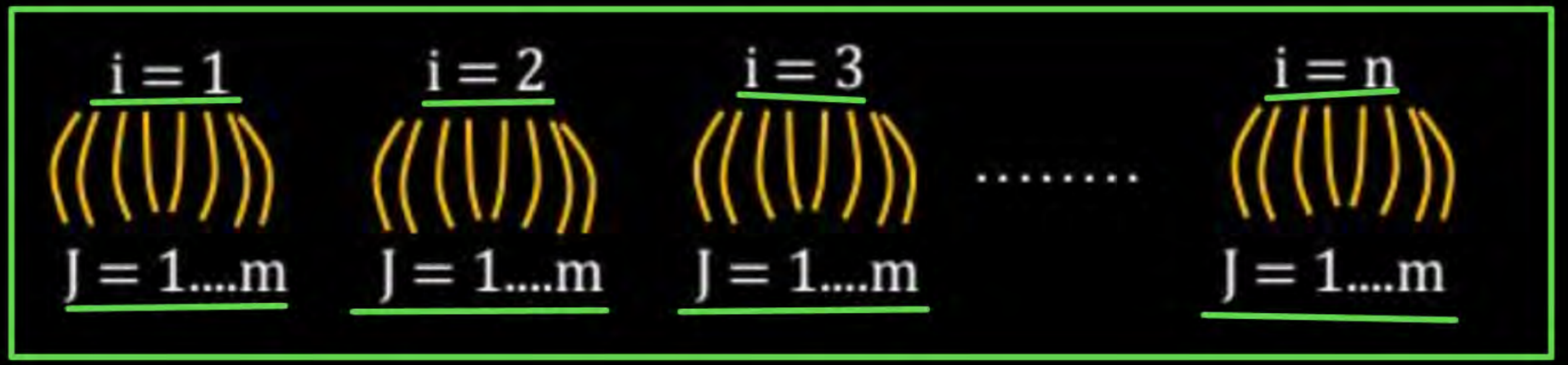
Bottom → Top

Corelated Nested
Query

Outer → Inner → Outer

Top → Bottom → Top

```
for(i = 1; i < = n; i++)
  for(j = 1; j <= m, j++)
```


i = 1
J = 1....m
i = 2
J = 1....m
i = 3
J = 1....m
.....
i = n
J = 1....m

Corelated Nested Query

```
Select   C.sid
FROM     Catalog
WHERE EXISTS ( Select*
               FROM Parts P
               WHEREP P.Pid = C.Pid
               AND Color = Red )
```

Inner Query using Attributes defined in the Outer Query

Catalog . Pid

TOP → Bottom → Top.

i=1 → j= 1 to m

i=2 → j=1 to m

I[st] Iteration:

[Pid Match & color = Red]

1 Tuple Return

S_1

NOT EXIST : False

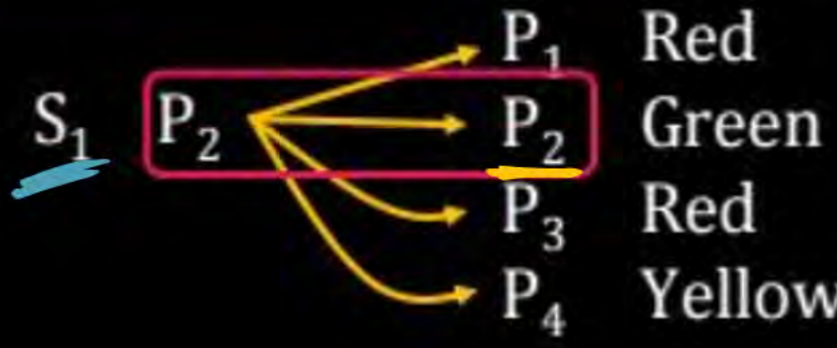
Catalog

Sid	Pid
S_1	P_1
S_1	P_2
S_2	P_3
S_4	P_4

Parts

Pid	Color
P_1	Red
P_2	Green
P_3	Red
P_4	Yellow

IInd Iteration:



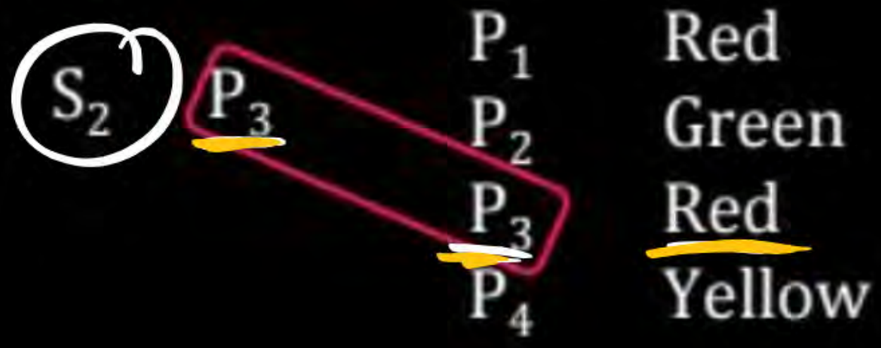
[Pid Match but color not Red]

EXIST

0 Tuple Return

NOT EXIST : True

IIIrd Iteration:

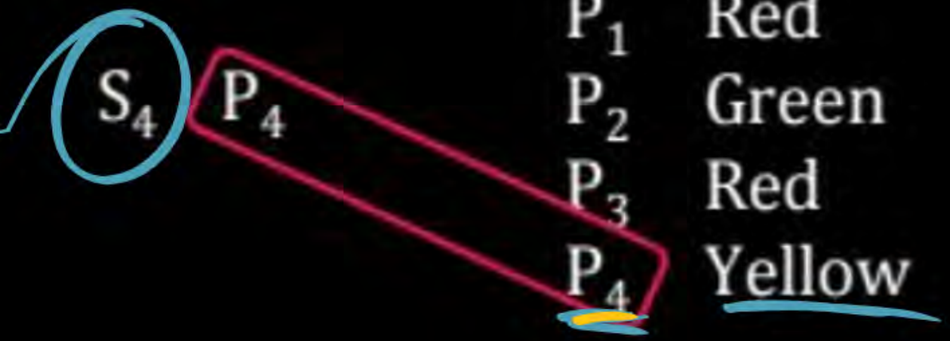


[Pid Match & Color Red]

1 Tuple Return

NOT EXIST : False

IVth Iteration:



P_1	Red
P_2	Green
P_3	Red
P_4	Yellow

[Pid Match but color not Red]

0 Tuple Return

↳ NOT EXIST : True.

↳ S_4

NOT EXIST

EXIST

↓

O/P

EXISTS

o/p

Sid
S_1
S_2

If NOT EXIST then output

Sid
S_1
S_4

Before EXIST & NOT EXISTS No Attribute is required.

Before IN & NOT IN Attribute is Required.

Q. Given Relative Schema
Emp(Eid, Ename, Salary)
Department(Eid, dname, code)
Retrieve Employee ID who have no Department?

Which is true?
A) Q_1 ✓ Q_2 ×
B) Q_2 ✓ Q_1 ×
C) Q_1 ✓ Q_2 ✓
D) Q_1 × Q_2 ×

Query I: Select Eid
FROM Emp E, Dep D
WHERE E.Eid <> D.Eid

E1 E1
E1 E2
E2 <> E1
E2 <> E1
E1 <> E2
E3 <> E1
E1 <> E1
E2 <> E2

4 X 3 = 12 Tuple.

E4 <> E1
E4 <> E1
E4 <> E2

Eid
E_1
E_2
E_3
E_4

Eid	Dname
E_1	A
E_1	B
E_2	A

Query II: Select Eid
FROM Emp E
WHERE NOT EXISTS (Select *
FROM Dep D
WHERE E. Eid = D. Eid)

Query I:

$E_1 <> E_1 \rightarrow F$
$E_1 <> E_1 \rightarrow F$
$E_1 <> E_2 \rightarrow T$
$E_2 <> E_1 \rightarrow T$
$E_2 <> E_1 \rightarrow T$
$E_2 <> E_2 \rightarrow F$
$E_3 <> E_1 \rightarrow T$
$E_3 <> E_1 \rightarrow T$
$E_3 <> E_2 \rightarrow T$
$E_4 <> E_1 \rightarrow T$
$E_4 <> E_1 \rightarrow T$
$E_4 <> E_2 \rightarrow T$

Output of Query I

Eid
E_1
E_2
E_2
E_3
E_3
E_3
E_4
E_4
E_4

Output of Query II

E_3
E_4

E_3 → E1, E1, E2

0 Tuple Return
Inner Query Result Empty.

E_1 → E_1, E_1, E_2
E_2 → E_1, E_1, E_2
E_3 → E_1, E_1, E_2
E_4 → E_1, E_1, E_2

E1 → E1 ✓, E1 ✓, E2

NOT EXIST: False

NULL:

- Non Zero
- Unknown
- Un existed
- No Two Null are equal

Q. Retrieve Eid who does not have any Passport Details?

```
Select  Eid
FROM    Emp
WHERE PPro=NULL
```

WHERE PPro IS NULL.

Employee

eid	ename	PassPortNo
E1	A	20
E2	B	NULL
E3	B	16

NOTE: for comparison with NULL SQL support IS/ IS Not Clause

```
WHERE PPro IS NULL
```

Q. Retrieve Eid who are having some passport details?

```
Select  Eid
FROM    Emp
WHERE PPro IS NOT NULL
```

Regular Expression:

%: Zero or More Character

_ : Exactly One character

_ : Exactly one character

% : Zero or More character.

Q. Retrieve Student whose Name Start with 'S' & end with M & at least 5 character?

'S _ _ _ _ % M'

S _ _ _ _ % M

```
Select *
FROM Student
WHERE Sname = 'S ___ % M'
```

X

Output: Pattern → 'S ___ % M'

WHERE Sname LIKE 'S _ _ _ % M'

LIKE: is used to compare to specify certain search Condition for a Pattern in a column.

```
Select *
FROM Student
WHERE Sname LIKE 'S ___ % M'
```

Q. Retrieve all student whose Name NOT Start with 'C'?

```
Select   *
FROM   Student
WHERE Sname NOT LIKE 'C%'
```

Sname NOT LIKE 'C%'

```
Select   *
FROM   Student
WHERE Sname
```

LIKE

A% →	Starts with A
%J →	end with J
%I% →	Contain I
'____' →	All 4 length Name
'S___' →	Starts with S & exactly 4 length character
'S___%' →	Starts with S & at least length 4.

$R \times S$:	CROSS Join
$R \cup S$:	Union Join
$R \bowtie S$:	Inner Join
$R \overset{\cdot}{\bowtie} S$:	Outer Join

ORDER By: Order By Clause is used to sort the Rows.

Company (Name, invoice no)

Q. Display all the company in alphabetical order of their Name?

```
Query:  Select   *
        FROM   Company
        Order by Name asc;
```

Q. If Reverse alphabetical order then

```
Order by Name Desc
```

Q.

Consider the following relation schemas:

b-Schema = (b-name, b-city, assets)

a-Schema = (a-num, b-name, bal)

d-Schema = (c-name, a-number)

Let branch, account and depositor be respectively instances of the above schemas. Assume that account and depositor relations are much bigger than the branch relation.

Consider the following query:

$\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{“agra”} \wedge\ \text{bal} < 0}\ (\text{branch} \bowtie \text{account} \bowtie \text{depositor}))$

Which one of the following queries is the most efficient version of the above query? [GATE-2007: 2 Marks]

A $\Pi_{\text{c-name}}\ (\sigma_{\text{bal} < 0}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie \text{account}) \bowtie \text{depositor})$

B $\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie\ (\sigma_{\text{bal} < 0 =}\ \text{account}) \bowtie \text{depositor})$

C $\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie \sigma_{\text{b-city} = \text{"Agra} \wedge \text{bal} < 0}\ \text{account} \bowtie \text{depositor})$

D $\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie (\sigma_{\text{b-city} = \text{"Agra} \wedge \text{bal} < 0}\ \text{account} \bowtie \text{depositor}))$

Q.

Consider two relations $R_1(A, B)$ with the tuples (1, 5), (3, 7)and R_2 (A, C) = (1, 7)(4, 9)

Assume that R(A, B, C) is the full natural outer join of R_1 and R_2. Consider the following tuples of the form (A, B, C); a = (1, 5, null), b = (1, null, 7), c = (3, null, 9), d = (4, 7, null), e = (1, 5, 7), f = (3, 7, null), g = (4, null, 9). Which one of the following statements is correct? [GATE-2015: 1 Mark]

A	B	C
1	5	N
1	N	7
3	N	9
4	7	N
1	5	7
3	7	N
4	N	9

(A) R contains a, b, e, f, g, but not c, d

(B) R contains all of a, b, c, d, e, f, g

(C) R contains e, f, g, but not a, b

(D) R contains e but not f, g

A	B
1	5
3	7

A	C
1	7
4	9

$R_1 \bowtie R_2$:

A	B	C
1	5	7
3	7	NULL
4	NULL	9

Ans(C)

NAT

Consider a database that has the relation schema CR(StudentName, CourseName). An instance of the schema CR is as given below:

The following query is made on the database.

$T1 \leftarrow \pi_{CourseName}(\sigma_{StudentName='SA'}(CR))$

$T2 \leftarrow CR \div T1;$

The number of rows in T2 is _____.

[GATE-2017-CS: 2M]

CR	
Student Name	Course Name
SA	CA
SA	CB
SA	CC
SB	CB
SB	CC
SC	CA
SC	CB
SC	CC
SD	CA
SD	CB

Student Name	Course Name
SD	CC
SD	CD
SE	CD
SE	CA
SE	CB
SF	CA
SF	CB
SF	CC

MCQ

The following relation records the age of 500 employees of a company, where empNo {indicating the employee number} is the key:

empAge(empNo, age)

Consider the following relational algebra expression:

$\Pi_{empNo}(\text{empAge} \bowtie_{(age > age1)} \rho_{empNo1, age1}(\text{empAge}))$

What does the above expression generate? [GATE-2020-CS: 1M]

A. Employee numbers of only those employees whose age is the maximum

B. Employee numbers of only those employees whose age is more than the age of exactly one other employee

C. Employee numbers of all employees whose age is not the minimum

D. Employee numbers of all employees whose age is the minimum

NAT

Consider the following relations P(X, Y, Z), Q(X, Y, T) and R(Y, V)

P		
X	Y	Z
X1	Y1	Z1
X1	Y1	Z2
X2	Y2	Z2
X2	Y4	Z4

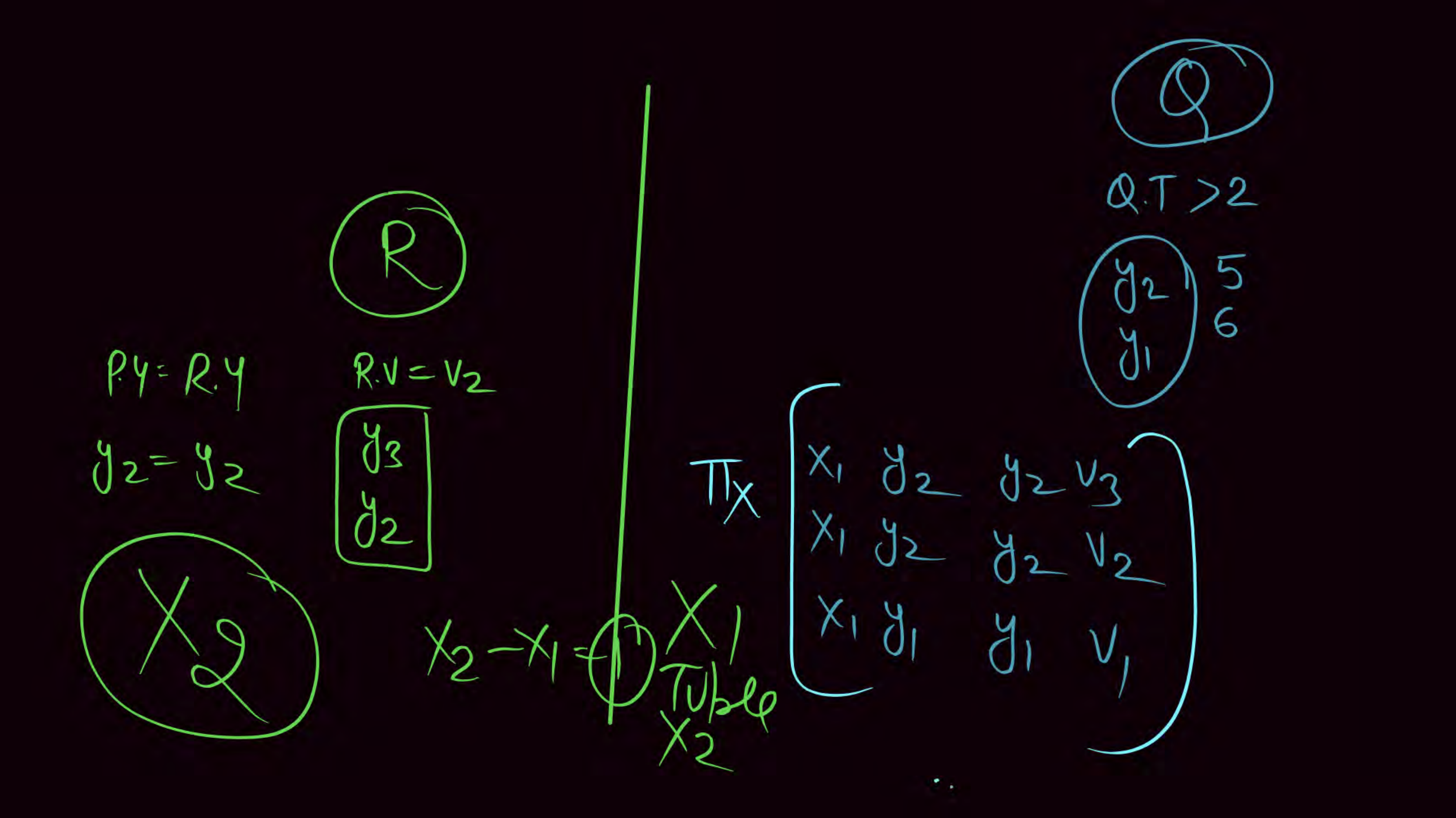
Q		
X	Y	T
X2	Y1	2
X1	Y2	5
X1	Y1	6
X3	Y3	1

R	
Y	V
Y1	V1
Y3	V2
Y2	V3
Y2	V2

How many tuples will be returned by the following relational algebra query?

$$[\Pi_X(\sigma_{(P.Y=R.Y \wedge R.V=V2)}(P \times R))] - [\Pi_X(\sigma_{(Q.Y=R.Y \wedge Q.T>2)}(Q \times R))];$$

[GATE-2019-CS: 2M]


Q
Q.T > 2
y_2 5
y_1 6
R
R.V = V_2
y_3
y_2
P.Y = R.Y
y_2 = y_2
X_2
$X_2 - X_1 = 1$
X_1
Tuple
X_2
π_X
x_1 y_2 y_2 v_3
x_1 y_2 y_2 v_2
x_1 y_1 y_1 v_1

MCQ

Suppose R_1(<u>A</u>, B) and R_2(<u>C</u>, D) are two relation schemes. Let r_1 and r_2 be the corresponding relation instances. B is a foreign key that refers to C in R_2. If data in r_1 and r_2 satisfy referential integrity constraints, which of the following is ALWAYS TRUE?

[GATE-2013-CS: 2M]

A $\Pi_B(r_1)-\Pi_C(r_2)=\phi$

B $\Pi_C(r_2)-\Pi_B(r_1)=\phi$

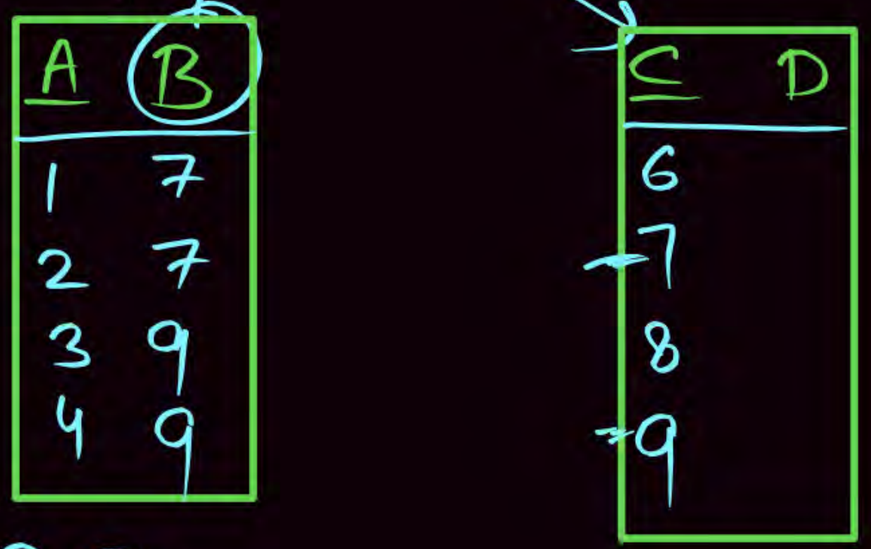
C $\Pi_B(r_1)=\Pi_C(r_2)$

D $\Pi_B(r_1)-\Pi_C(r_2)\neq\phi$

Ans (A)

Referring

Referenced



B : F.K

(a) $\pi_B(\gamma_1) - \pi_C(\gamma_2)$

$(7,9) - (6,7,8,9) = \phi$

(b) $\pi_C(\gamma_2) - \pi_B(\gamma_1)$

$(6,7,8,9) - (7,9) = 6,8 \neq \phi$

(c) $\pi_B(\gamma_1) = \pi_C(\gamma_2)$

$(7,9) \neq (6,7,8,9)$

(d) ϕ

NAT

Consider the following table named Student in a relational database. The primary key of this table is rollNum.

Student

Roll Num	Name	Gender	Marks
1	Naman	M	62
2	Aliya	F	70
3	Aliya	F	80
4	James	M	82
5	Swati	F	65

The SQL query below is executed on this database.

```
SELECT *
FROM Student
WHERE gender = 'F' AND marks > 65;
```

The number of rows returned by the query is

(2) Ans

[GATE-2023-CS: 2M]

MCQ

Consider the following relation A, B and C:

A		
ID	Name	Age
12	Arun	60
15	Shreya	24
99	Rohit	11

B		
ID	Name	Age
15	Shreya	24
25	Hari	40
98	Rohit	20
99	Rohit	11

C		
ID	Phone	Area
10	2200	02
99	2100	01

$(A \cup B) \equiv A$

How many tuples does the result of the following relational algebra expression contain? Assume that the schema of $A \cup B$ is the same as that of A.

$(A \cup B) \bowtie_{A.Id > 40 \vee C.Id < 15} C$

[GATE-2012-CS: 2M]

(A) 7 (B) 4 (C) 5 (D) 9

5 Tuples 2 Tuples

5 X 2 = 10 Tuples

AUB ≡ A

A

ID	Name	Age
12	Arun	60
15	Shreya	24
99	Rohit	11
25	Hari	40
98	Rohit	20

C

ID	Phone	Age
10	2200	02
99	2100	01

$(A \cup B) \bowtie_{A.Id > 40 \ (\vee, OR) \ C.Id < 15}$ C

A.ID A.Name A.Age C.ID C.Phn C.age

1 ---- 12 ---- Arun ---- 60 ---- 10 ---- 2200 ---- 02

2 ---- 12 Arun ---- 99 2100 01

15 S

15 ---- 99 Ro ---- 10

99

4 ---- 99 Rohit ---- 99

5 ---- 25 ---- 10

25 Hari

25 ---- 98 ---- Rohit ---- 10

99

7 ---- 98 Rohit ---- 99

105+ PYQ

1139 Till Now in class.

Q.

Consider the following relation

Cinema (theatre, address, capacity)

Which of the following options will be needed at the end of the SQL query

```
SELECT P1.address
FROM Cinema P1
```

such that it always finds the addresses of theaters with maximum capacity?

[MCQ: 2015]

theatre	Addl	Capacity
T_1	A	100
T_2	B	200
T_3	C	300
T_4	D	400

A. WHERE P1.capacity > = ALL (select P2.capacity from Cinema P2)

4 Tuples (A, B, C, D)

B. WHERE P1.capacity > = Any (select P2.capacity from Cinema P2)

0 Tuples

C. WHERE P1.capacity > ALL (select max (P2.capacity) from Cinema P2)

400

D. WHERE P1.capacity > Any (select max (P2.capacity) from Cinema P2)

400

$x >= ALL\ (100, 200, 300, 400)$

$(x >= 100)\ AND\ (x >= 200)\ AND\ (x >= 300)\ AND\ (x >= 400)$

↳ 400 (D)

~~(B)~~ $x >= ANY\ (100, 200, 300, 400)$

↳ Op A, B, C, D

(C) $x > ALL\ (400)$

Q.

Database table by name Loan_Records is given below:
What is the output of the following SQL query?

```
SELECT count(*)
FROM (
        SELECT Borrower, Bank_Manager
        FROM Loan_Records) AS S NATURAL JOIN
        (SELECT Bank-Manager, Loan_Amount
         FROM Loan_Records) AS T
        );
```

[MCQ: 2011:2M]

A 3

B 9

C 5

D 6

Borrower	Bank_Manager	Loan_Amaount
Ramesh	Sunderajan	10000.00
Suresh	Ramgopal	5000.00
Mahesh	Sunderajan	7000.00

Q. SELECT operation in SQL is equivalent to [MCQ: 2015-1M]

A The selection operation in relational algebra.

B The selection operation in relational algebra, except that SELECT in SQL retains duplicates.

C The projection operation in relational algebra.

D The projection operation in relational algebra, except that SELECT in SQL retains duplicates.

Q.

Consider the following database table named top_scorer.
Consider the following SQL Query:

```
SELECT ta.player FROM top_scorer AS ta
WHERE ta.goals > ALL (SELECT tb.goals
      FROM top_scorer AS tb
      WHERE tb.country = 'Spain')
AND ta.goals > ANY (SELECT tc.goals
      FROM top_scorer AS tc
      WHERE tc.country = 'Germany')
```

The number of tuples returned by the above SQL query is _____.

[NAT:2017–2M]

top_scorer		
player	country	goals
Klose	Germany	16
Ronaldo	Brazil	15
G Muller	Germany	14
Fountaine	France	13
Pele	Brazil	12
Klinsmann	Germany	11
Kocsis	Hungary	11
Batistuta	Argentina	10
Cubillas	Peru	10
Lato	Poland	10
Lineker	England	10
T Muller	Germany	10
Rahn	Germany	10

Q.

Consider the following relations:

Student	
Roll_No	Student_Name
1	Raj
2	Rohit
3	Raj

Performance		
Roll_No	Course	Marks
1	Math	80
1	English	70
2	Math	75
3	English	80
2	Physics	65
3	Math	80

Consider the following SQL query.

```
SELECT S . Student_Name, Sum(P. Marks)
FROM Student S, Performance P
WHERE S. Roll_No = P.Roll_No
Group BY S. Student_Name
```

The number of rows that will be returned by the SQL query is _____.

[NAT: 2015–2M]

Q.

Consider the relational database with the following four schemes and their respective instances.

Student(sNo, sName, dNo) Dept(dNo, dName)
Course(cNo, cName, dNo) Register(sNo, cNo)

Student		
sNo	sName	dNo
S01	James	D01
S02	Rocky	D01
S03	Jackson	D02
S04	Jane	D01
S05	Milli	D02

Dept	
dNo	dName
D01	CSE
D02	EEE

Course		
cNo	cName	dNo
C11	DS	D01
C12	OS	D01
C21	DE	D02
C22	PT	D02
C23	CV	D03

Register	
sNo	cNo
S01	C11
S01	C12
S02	C11
S03	C21
S03	C22
S03	C23
S04	C11
S04	C12
S05	C11
S05	C21

Question Continues in Next Slide

SQL Query:

```
SELECT * FROM Student AS S WHERE NOT EXIST
    (SELECT cNo FROM Course WHERE dNo = "D01".
        EXCEPT
SELECT cNo FROM Register WHERE sNo = S.sNo)
```

The number of rows returned by the above SQL query is _____.

[NAT: 2022: 2M]

Q.

Consider a database that has the relation schema EMP(EmpId, EmpName, and DeptName). An instance of the schema EMP and a SQL query on it are given below:

```
SELECT AVG(EC.Num)
FROM EC
WHERE(DeptName, Num) IN
        (SELECT DeptName, COUNT(EmpId)AS
                                EC(DeptName, Num)
        FROM EMP
        GROUP BY DeptName)
```

The output of executing the SQL query is _____.

[NAT:2017-1M]

EMP		
EmpId	EmpName	DeptName
1	XYA	AA
2	XYB	AA
3	XYC	AA
4	XYD	AA
5	XYE	AB
6	XYF	AB
7	XYG	AB
8	XYH	AC
9	XYI	AC
10	XYJ	AC
11	XYK	AD
12	XYL	AD
13	XYM	AE

Q.

A relational database contains two tables Student and Performance as shown below:

Student

Roll_no	Student_name
1	Amit
2	Priya
3	Vinit
4	Rohan
5	Smita

Performance

Roll_no	Student_code	Marks
1	A	86
1	B	95
1	C	90
2	A	89
2	C	92
3	C	80

The primary key of the Student table is Roll_no. For the Performance table, the columns Roll_no. and Subject_code together from the primary key. Consider the SQL query given below:

SELECT S.Student_name, sum(P.Marks) FROM Student S, Performance P
WHERE P. Marks > 84 GROUP BY S. Student_name;

The number of rows returned by the above SQL query is _____.

[NAT: 2019–2M]

Q.

Consider the following database table named water_schemes:
The number of tuples returned by the following SQL query is:

```
with total(name, capacity) as
    select district_name, sum(capacity)
    from water_schemes
    group by district_name
with total_avg(capacity) as
    select avg(capacity)
    from total
    select name
      from total, total_avg
      where total.capacity ≥ total_avg.capacity
```

water_schemes		
scheme_no	district_name	capacity
1	Ajmer	20
1	Bikaner	10
2	Bikaner	10
3	Bikaner	20
1	Churu	10
2	Churu	20
1	Dungargarh	10

[NAT:2016-2M]

2 Avg

Total

Name	Capacity
Ajmer	20
Bikaner	40
Churu	30
Dungargarh	10

Total Avg

Capacity
25

Bikaner
Churu

(2) Avg

$$\frac{20+40+30+10}{4} = \frac{100}{4} = 25$$

Q.

Suppose $R_1(\underline{A}, B)$ and $R_2(\underline{C}, D)$ are two relation schemas. Let r_1 and r_2 be the corresponding relation instances. B is a foreign key that refers to C in R_2. If data in r_1 and r_2 satisfy referential integrity constraints, which of the following is ALWAYS TRUE?

[MCQ: 2012–2M]

A $\Pi_B(r_1) - \Pi_C(r_2) = \phi$

B $\Pi_C(r_2) - \Pi_B(r_1) = \phi$

C $\Pi_B(r_1) = \Pi_C(r_2)$

D $\Pi_B(r_1) - \Pi_C(r_2) \neq \phi$

Q.

Consider a relational database containing the following schemas.

Catalogue		
sno	pno	cost
S1	P1	150
S1	P2	50
S1	P3	100
S2	P4	200
S2	P5	250
S3	P1	250
S3	P2	150
S3	P5	300
S3	P4	250

Suppliers		
sno	sname	location
S1	M/s Royal furniture	Delhi
S2	M/s Balaji furniture	Bangalore
S3	M/s Premium furniture	Chennai

Parts		
pno	**pname**	**part_spec**
P1	Table	Wood
P2	Chair	Wood
P3	Table	Steel
P4	Almirah	Steel
P5	Almirah	Wood

Question Continues in Next Slide

The primary key of each table is indicated by underlining the constituent fields

```
SELECT s.no, s.name
FROM Suppliers s, Catalogue c
WHERE s.sno = c.sno AND
        cost > (SELECT AVG (cost)
                FROM Catalogue
                WHERE pno = 'p4'
               GROUP BY pno);
```

The number of rows returned by the above SQL query is [MCQ: 2020-2M]

(A) 4

(B) 5

(C) 0

(D) 2

MCQ

Consider a database table T containing two columns X and Y each of type integer. After the creation of the table, one record (X=1, Y=1) is inserted in the table.

Let MX and MY denote the respective maximum values of X and Y among all records in the table at any point in time. Using MX and MY, new records are inserted in the table 128 times with X and Y values being MX+1, 2*MY+1 respectively. It may be noted that each time after the insertion, values of MX and MY change.

What will be the output of the following SQL query after the steps mentioned above are carried out?

SELECT Y FROM T WHERE X=7;

[GATE-2011-CS: 2M]

A 127

B 255

C 129

D 257

Q.

Consider the following three relations in a relational database:

Employee (eId, Name), Brand (bId, bName), Own(eId, bId)

Which of the following relational algebra expressions return the set of eIds who own all the brands? [GATE: 2022]

A $\pi_{eId}\,(\pi_{eId,\,bId}\,(Own / \pi_{bId}\,(Brand))$

B $\pi_{eId}\,(Own) - \pi_{eId}\,((\pi_{eId}\,(Own) \times \pi_{bId}\,(Brand)) - \pi_{eId,\,bId}\,(Own))$

C $\pi_{eId}\,(\pi_{eId,\,bId}\,(Own) / \pi_{bId}\,(Own))$

D $\pi_{eId}\,((\pi_{eId}\,(Own) \times \pi_{bId}\,(Own) / \pi_{bId}\,(Brand))$

Consider the two relation Suppliers and Parts are given below.

Suppliers

S_{no}	P_{no}
S_1	P_1
S_1	P_2
S_1	P_3
S_1	P_4
S_2	P_1
S_2	P_2
S_3	P_2
S_4	P_2
S_4	P_4

Parts

P_{no}
P_2
P_4

$\pi_{S_{no}\,P_{no}}$ (Suppliers) / $\pi_{P_{no}}$ (Parts)

The number of tuples are there in the result when the above relational algebra query executes is _____.

Q.

Consider the following relation schemas:

b-Schema = (b-name, b-city, assets)

a-Schema = (a-num, b-name, bal)

d-Schema = (c-name, a-number)

Let branch, account and depositor be respectively instances of the above schemas. Assume that account and depositor relations are much bigger than the branch relation.

Consider the following query:

$\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{“agra”} \wedge\ \text{bal} < 0}\ (\text{branch} \bowtie \text{account} \bowtie \text{depositor}))$

Q. Which one of the following queries is the most efficient version of the above query? [GATE-2007: 2 Marks]

A $\Pi_{\text{c-name}}\ (\sigma_{\text{bal} < 0}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie \text{account}) \bowtie \text{depositor})$

B $\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie\ (\sigma_{\text{bal} < 0 =}\ \text{account}) \bowtie \text{depositor})$

C $\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie \sigma_{\text{b-city} = \text{"Agra} \wedge \text{bal} < 0}\ \text{account} \bowtie \text{depositor})$

D $\Pi_{\text{c-name}}\ (\sigma_{\text{b-city} = \text{"Agra"}}\ \text{branch} \bowtie (\sigma_{\text{b-city} = \text{"Agra} \wedge \text{bal} < 0}\ \text{account} \bowtie \text{depositor}))$

Q. Consider two relations $R_1(A, B)$ with the tuples (1, 5), (3, 7)and R_2 (A, C) = (1, 7)(4, 9)
Assume that R(A, B, C) is the full natural outer join of R_1 and R_2. Consider the following tuples of the form (A, B, C); a = (1, 5, null), b = (1, null, 7), c = (3, null, 9), d = (4, 7, null), e = (1, 5, 7), f = (3, 7, null), g = (4, null, 9). Which one of the following statements is correct? [GATE-2015: 1 Mark]

A. R contains a, b, e, f, g, but not c, d

B. R contains all of a, b, c, d, e, f, g

C. R contains e, f, g, but not a, b

D. R contains e but not f, g

Procedural Q.L.

- RA
- SQL

Non Procedural Q.L.

- TRC

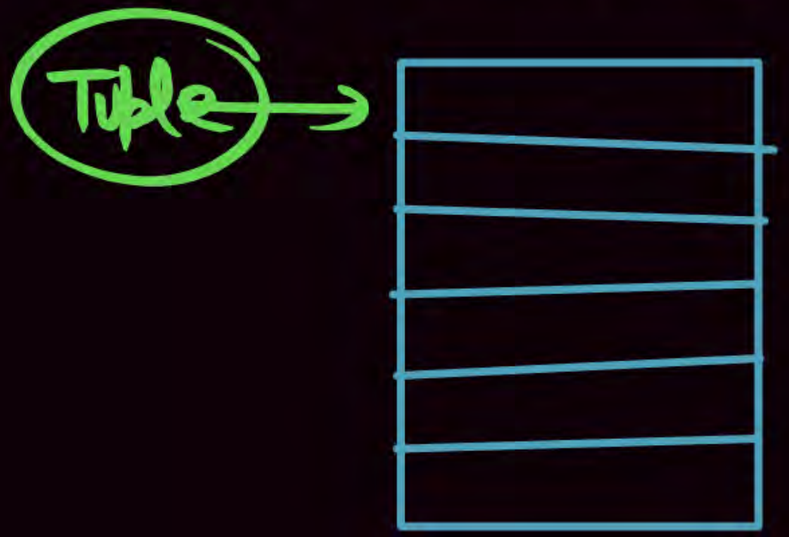
→ DM ⇒ Predicate logic.

⇓

'BABA SIR'

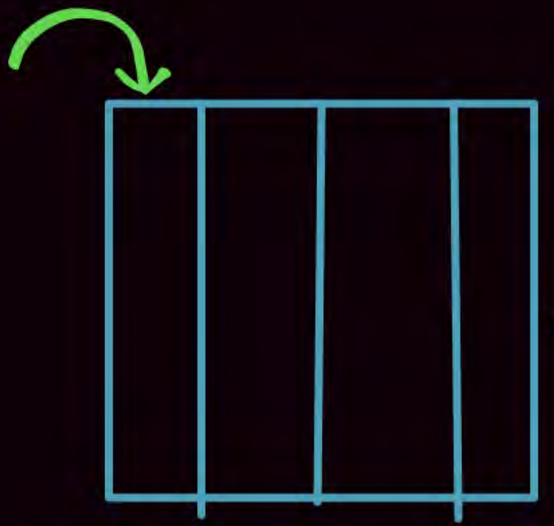
TRC

Tuple Relational Calculus



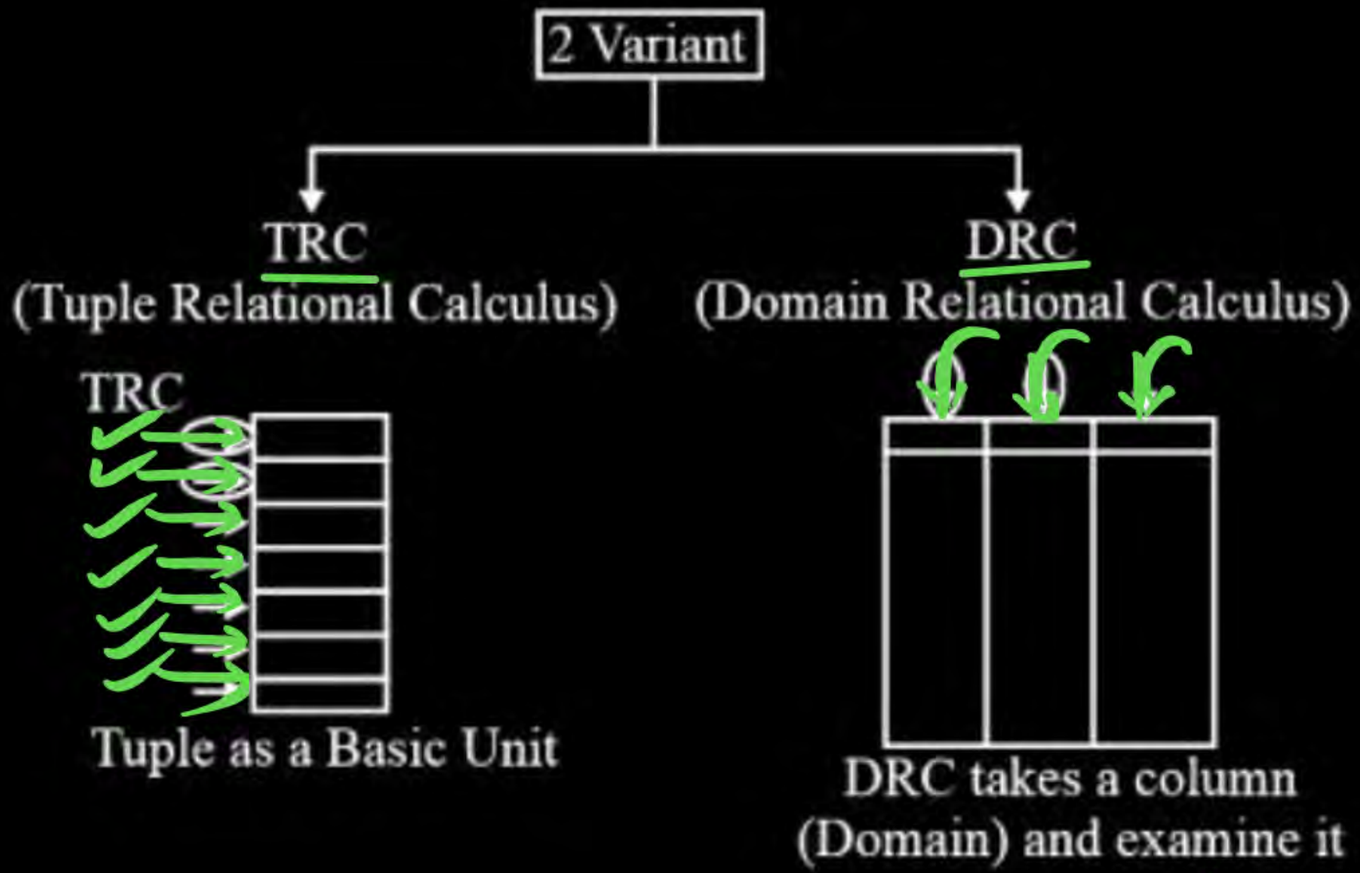
DRC

Domain Relational Calculus



Relational Calculus (Based on predicate calculus)

Non procedural query language and has



TRC

- Query describe result in the form of set of Tuples.

Tuple Variable

- It is variable that takes on Tuples of a relation schema as value.

DRC

- Query describe result in the form of set of Column. (Domain).

Domain Variable

- It is variable that Range over the domain of some attribute (column).

Form of Query

$\{T \mid P(T)\}$

Tuple Variable

Formula which Describe Tuple variable

Form of Query

$\{<x_1 x_2 x_3 \ldots\ldots x_n> \mid P<x_1 x_2 x_3 \ldots\ldots x_n>\}$

Domain variable

Formula which describe Domain variable

TRC

$\{T \mid P(T)\}$ ≡

T : Tuple variable

P(T): Formula over Tuple (T)

Such that P(T) is satisfied

Form of Query

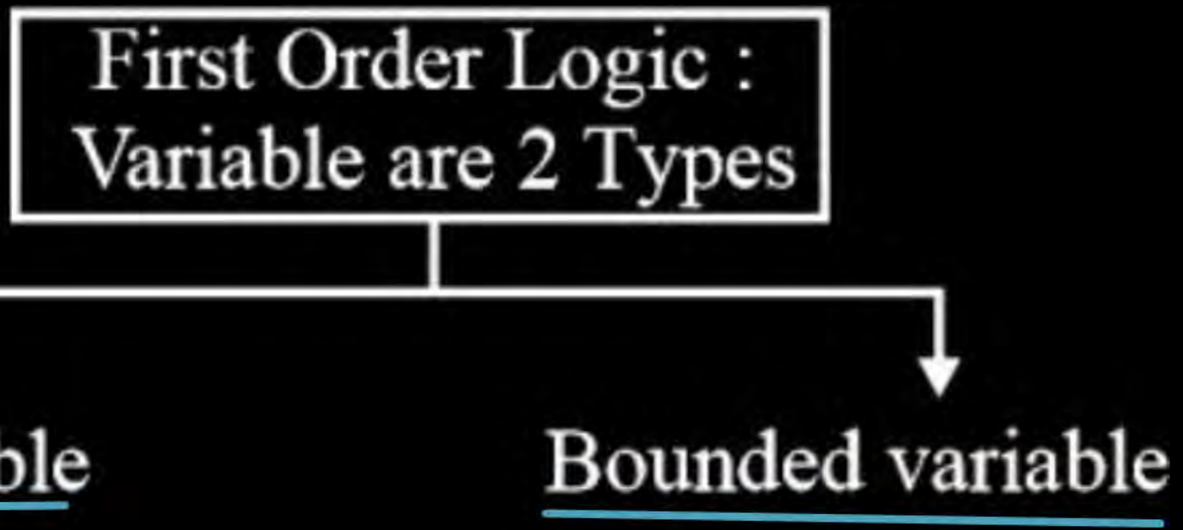
T {Selected Sid

P(T) { FROM Student

WHERE Condition

CGPA > 8

NOTE: Formula of TRC is expressed using first order logic **(Predicate logic)**.



∀: for all
∃: there Exist

Quantifiers

1. ∀ : For all (ALL)
2. ∃ : There exists (Any)

Bounded Variable: If tuple variable is preceded by the quantifier then it is Bounded variable.

Belong

1. $\forall \in$ Supplier
2. $\exists \in$ Supplier

Free variable: If tuple variable is not bounded by the quantifier then it is free variable.

NOTE: The result of TRC should be free variable.

Ques.

LOAN (Loanno, Branch name, Amount)

Q. Find Loan no of amount above 50000.

SQL: Select LoanNo From Loan Where Amount > 50000

RA: $\Pi_{LoanNo}(\sigma_{amount > 50000}(LOAN))$

TRC: {T | T ∈ Loan (Total amount > 50,000)}

(T belongs)

DRC: {<Loanno., branch name, amount> | < Loan no, Branch, amount ∈ Loan (amount > 50000)}

Q. Retrieve Sid of the supplier who supplied some Red color parts.

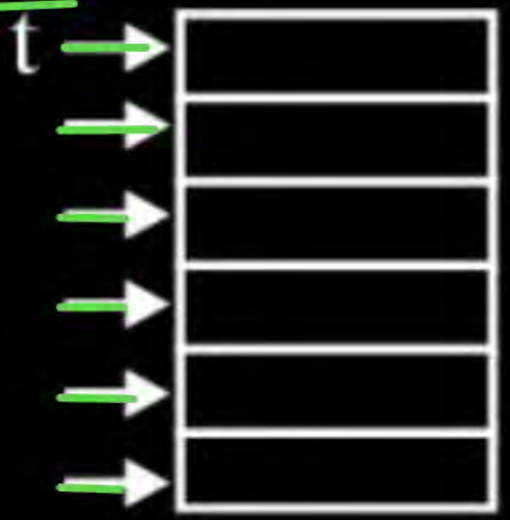
R.A:

$$\pi_{Sid}\left(\sigma_{C.Pid=P.Pid}\left(Catalog \times \sigma_{colour=Red}\left(Parts\right)\right)\right)$$

TRC:

$$\left(T \mid \exists C \in Catalog\left(\exists P \in parts\left(P.color = Red \wedge P.Pid = C.Pid\right) \wedge T = C.Sid\right)\right)$$

TRC:

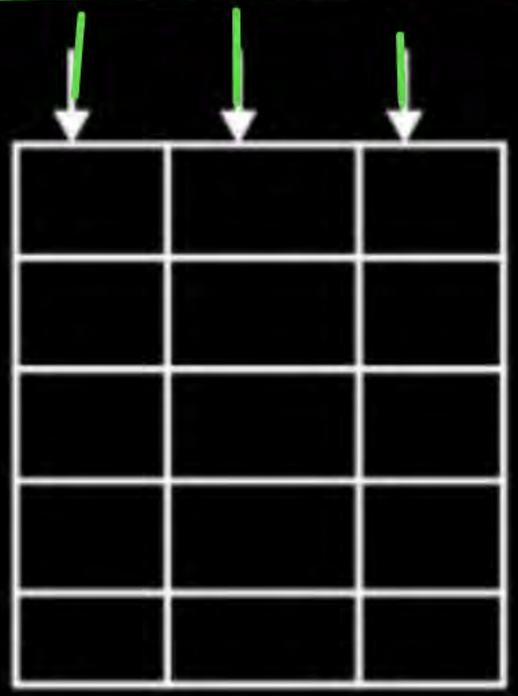


Takes a tuple and examine it

Tuple as a Basic unit

DRC: Takes a column

Range over the column or Domain and examine it



TRC: Unsafe operation occur

Unsafe operation: Student (t)

~(t)

~(t): Not belongs to student table (infinite/Universe all tuple)

So, its unsafe operation.

Note

Whenever Two Table are Joined (Natural Join) With Respect to Primary & foreign key then

Maximum Number of Tuples in the Resulting Relation is equal to Number of tuples in the Referencing Relation.

Q.1

Consider the relations:

Supply (SupplierID, Itemcode) with 1000 tuples

Inventory (Itemcode, color) with 2500 tuples

Let p and q be the number of maximum and minimum records in Supply JOIN Inventory, the value of p + q is _____. (Itemcode is FK in Supply table).

Q. 2

Consider the relations:

Employee(Eid, Ename) with 2000 tuples

Department(Eid, Did) with 1000 tuples

Let p and q be the number of maximum and minimum records in Employee JOIN Department, the value of p + q is _____. (Eid is FK in Department table).

Q.3

Consider the relations:

R1(A, B,C) with n tuples

R2(B, D,E) with m tuples (B is FK in R1).

maximum tuples in $R1 \bowtie R2$

minimum tuples in $R1 \bowtie R2$

Q.4

Consider the relations:

R1(A, B,C) with n tuples

R2(B, D,E) with m tuples (B is Not FK in R1).

maximum tuples in $R1 \bowtie R2$

minimum tuples in $R1 \bowtie R2$

Q.5

Consider the join R ⋈ S between R (A B C) and S (A D E) with attribute A being the primary key in both relations and attribute A in S is a Foreign Key referencing to attribute A in R. R has m tuples and S has n tuples, maximum and minimum number of tuples in R ⋈ S respectively are?

(a) m, m + n

(b) m*n, 0

(c) min (m, n), min (m, n)

(d) max (m, n), min (m, n)

Q. 6

Consider the relations:

R1(A, B, C) with n tuples

R2(D, B, E) with m tuples

$\{A \rightarrow B,\ B \rightarrow C,\ D \rightarrow B,\ B \rightarrow E\}$

maximum tuples in $R1 \bowtie R2$

minimum tuples in $R1 \bowtie R2$

MCQ

The following functional dependencies hold for relations R(A, B, C) and S(B, D, E) FD for both the tables.

$B \rightarrow A$

$A \rightarrow C$

The relation R contains 200 tuples and the relation S contains 100 tuples. What is the maximum number of tuples possible in the natural join $R \bowtie S$?

[GATE-2010-CS: 2M]

A 100

B 200

C 300

D 2000

THANK YOU!