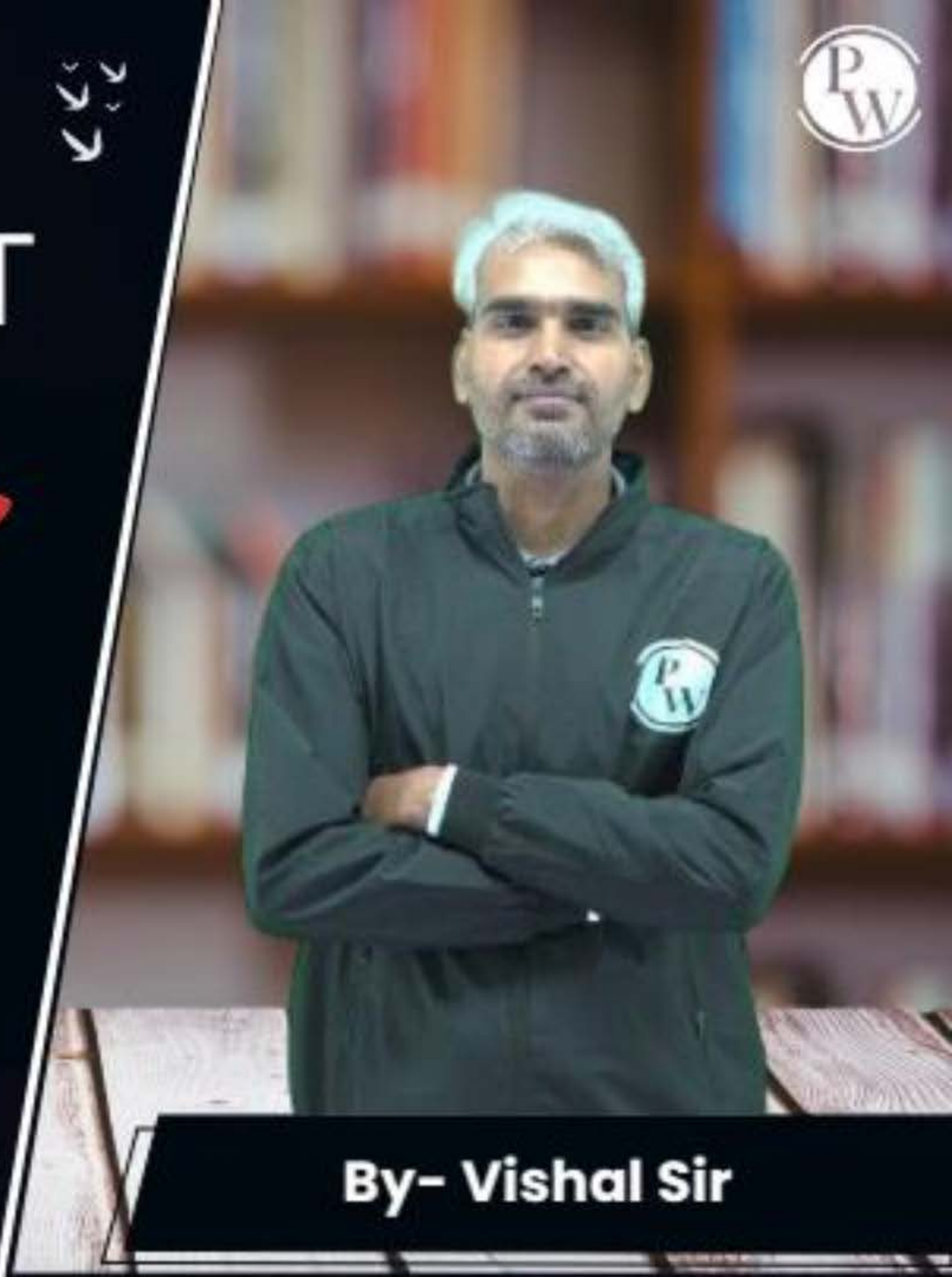
Computer Science & IT

Database Management System

Query Languages

Lecture No. 09



By- Vishal Sir

Recap of Previous Lecture

- Topic: Comparison with NULL
- Topic: Comparison with regular expression
- Topic: Practice questions

Topics to be Covered

- Topic — Practice questions
- Topic — Tuple relational calculus (TRC)

#Q. H.W.

Consider The Following Relational Scheme

Sch-name may be same for two schools with different School-id

1. Student (school-id, sch-roll-no, sname, saddress)
2. School (school-id, sch-name, sch-address, sch-phone)
3. Enrolment (school-id, sch-roll-no, erollno, examname)
4. ExamResult (Erollno, examname, marks)

(A)for each school with more than 200 students appearing in exams, the name of the school and the number of 100s scored by its students

(B)for each school with more than 200 students in it, the name of the school and the number of 100s scored by its students

(C)for each school with more than 200 students in it, the name of the school and the number of its students scoring 100 in at least one exam

(D)nothing; the query has a syntax error

Question Continues in Next Slide

[MCQ: 2008: 2M]

#Q. What does the following SQL query output?

```
SELECT sch-name, COUNT (*)
FROM School C, Enrolment E, ExamResult R
WHERE   E.school-id = C.school-id
                AND
        E.examname = R.examname
                AND
        E.erollno = R.erollno
                AND
        R.marks = 100
                AND
        E.school-id IN
                (SELECT school-id
                FROM student GROUP BY school-id HAVING COUNT (*) > 200)
GROUP By School-id, Sch-name
```

How:

For option 'B' to be true

What changes needs to be performed in the query when

(i) Sch-name may be same for schools with different School-id

(ii) Sch-name are also unique

Max

School_id	Sch-name	E-roll no.	Exam	Marks
01 ← #Students > 200	Saraswati	05	GATE	100
05 ← #Students > 200	Saraswati	07	JEE	100
05	Saraswati	09	GATE	100
06 ← #Student > 20	St. Joseph	15	JEE	100

Sch-name	Count (*)
Saraswati	1
Saraswati	2

[MCQ: 2008: 2M]

#Q. What does the following SQL query output?

```
SELECT sch-name, COUNT (*)
FROM School C, Enrolment E, ExamResult R
WHERE   E.school-id = C.school-id
                AND
        E.examname = R.examname
                AND
        E.erollno = R.erollno
                AND
        R.marks = 100
                AND
        E.school-id IN
                (SELECT school-id
                FROM student GROUP BY school-id HAVING COUNT (*) > 200)
GROUP By Sch-name
```

H.W:

For option 'B' to be true

What changes needs to be performed in the query when

(i) Sch-name may be same for schools with different school-id

(ii) Sch-name are also unique

	School_id	Sch-name	E-rollno.	Exam	Marks
#Students>200 ←	01	Saraswati	05	GATE	100
#Students>200 ←	05	St. Johns	07	JEE	100
	05	St. Johns	09	GATE	100

Sch-name	Count (*)
Saraswati	1
St. Johns	2

#Q. H.W.

[MCQ: 2009: 2M]

Consider the following relational schema:
Suppliers(sid:integer, sname:string, city:string, street:string)
Parts(pid:integer, pname:string, color:string)
Catalog(sid:integer, pid:integer, cost:real)

Consider the following relational query on the above database:

```
SELECT S.sname
        FROM Suppliers S
        WHERE S.sid NOT IN
                (SELECT C.sid
                        FROM Catalog C
                        WHERE C.pid NOT IN
                                (SELECT P.pid
                                        FROM Parts P
                                        WHERE P.color<> \'blue\'))
```

(Options on the Next Slide)

Hint: Just provide your answer (None of the option given may be true)

Question Continues in Next Slide

#Q. [MCQ: 2009: 2M]

H.W.

Consider the following relational schema:

Suppliers(sid:integer, sname:string, city:string, street:string)

Parts(pid:integer, pname:string, color:string)

Catalog(sid:integer, pid:integer, cost:real)

Consider the following relational query on the above database:

```
SELECT S.sname
        FROM Suppliers S
        WHERE S.sid NOT IN
        (SELECT C.sid
        FROM Catalog C
        WHERE C.pid NOT IN
            (SELECT P.pid
            FROM Parts P
            WHERE P.color<> \'blue\'))
```

Supplier

S_1
S_2
S_3
S_4

Parts

P_1	Red
P_2	Blue
P_3	Green

Catalog

S_1	P_1
S_1	P_2
S_2	P_2
S_3	P_3

Sname's of suppliers who did not supply any blue color part.

o/p = { S_3 , S_4 }

Sname Sname

Supplier who did not supply any part and also present in supplier table

Sids of the suppliers who have supplied at least one blue color part

o/p = {S_1, S_2}

o/p: Pids of non-blue color parts

{ P_1, P_3 }

Pid not in Pids of non-blue color part

i.e. Pid is actually of blue color part

Question Continues in Next Slide

#Q. Assume that relations corresponding to the above schema are not empty. Which one of the following is the correct interpretation of the above query?

A. Find the names of all suppliers who have supplied a non-blue part.

B. Find the names of all suppliers who have not supplied a non-blue part.

i.e. Supplied only blue parts (or) no parts

C. Find the names of all suppliers who have supplied only blue parts.

D. Find the names of all suppliers who have not supplied only blue parts.

↳ At least one blue + Some other Color Parts as well

Relational Calculus

Topic : Relational Calculus

- Relational Calculus is a non-procedural query language.
- It uses predicate calculus(First Order Logic) instead of algebra.

$\forall, \exists$ etc

- Relational calculus provides description about the query to get the result, whereas Relational Algebra gives the method to get the result.

Topic : Types of Relational Calculus

There are two types of Relational Calculus.

1. Tuple Relational Calculus: Works on tuple (or rows)
2. Domain Relational Calculus: Works on domain of attribute (column)

Topic : Syntax of TRC

A set of tuples t such that Predicate Cond" P is true for those tuples 't'

Basic syntax of TRC is { t | P(t) }

↳ it is called output tuple variable, it is used to produce O/P.

- where t is a tuple variable and P(t) is a predicate formula that describes the conditions that must be satisfied by the resulting tuples.

1. It results the set of tuples t, s.t. P(t) is true for t.
2. t ∈ *R*elation denotes that the 't' is a tuple from Relation, also denoted using Relation(t).
3. If 't' is a tuple variable, then "t[A]" / "t.A" specify attribute A in tuple t.
4. P is a predicate formula similar to that of the predicate calculus.

Topic : Equivalence between SQL, RA and TRC

SQL ➤ SELECT distinct * FROM Employee

R.A ➤ (Employee)

TRC ➤ {t | t ∈ Employee}
or
{t | Employee(t)}

it will be true for all tuples of Employee table,
∴ All tuples of Employee will be present in output

Predicate Condⁿ

$$\{ t \mid t \in \text{Employee} \}$$

$$\equiv$$

$$\{ t \mid \text{Employee}(t) \}$$

Topic : Free and Bounded variables

- A variable which is preceded by a quantifier is called bounded variable.
- A variable which is not preceded by any quantifier is called free variable.
- There is an important restriction:

  The variable ‘t’ that appears before ‘|’ symbol must be the only free variable in formula P(t).

  In other words all other variables must be bounded using quantifier

Topic : Equivalence between SQL, RA and TRC

SQL ➤ SELECT distinct ename FROM Employee

R.A. ➤ π_{ename} (Employee)

TRC ➤ { t.ename | t ∈ Employee}

or

{ t[ename] | t ∈ Employee}

or

{ t | ∃S ∈ Employee (t[ename]= S[ename])}

or

{ t | ∃S ∈ Employee (t.ename= S.ename)}

{ t[ename] | t ∈ Employee }

|||

{ t.ename | t ∈ Employee }

|||

{ t.ename | Employee (t) }

{ t | ∃S ∈ Employee ∧ t = S.ename }

attribute Creation w.r.t tuple t

{ t | ∃S ∈ Employee (t.ename = S.ename) }

#Q. Retrieve tuples from the relation student where marks are more than 50
Student(sid, name, marks)

$$\{ t \mid t \in Student \wedge t.marks > 50 \}$$

Predicate Condⁿ

$$\{ t \mid \exists S \in Student \, (S.Marks > 50 \wedge t.Sid = S.Sid \wedge t.name = S.name \wedge t.marks = S.marks) \}$$

Attribute Creation

#Q. Retrieve sids of all the students from the relation student whose marks are more than 50 .

Student(sid, name, marks)

$$\{t.sid \mid t \in Student \wedge t.marks > 50\}$$

$$\{t \mid \exists S \in Student \wedge S.marks > 50 \wedge t.sid = S.sid\}$$

#Q. Retrieve sids and names of all the students from the relation student whose marks are more than 50 .
Student(sid, name, marks)

$$\{ t.sid, t.name \mid t \in Student \wedge t.marks > 50 \}$$

$$\{ t \mid \exists s \in Student \, (s.marks > 50 \wedge t.sid = s.sid \wedge t.nam = s.name) \}$$

#Q. Consider the following relational schema:

Suppliers(sid, sname, city)

Parts(pid, pname, color)

Catalog(sid, pid, cost)

Find sid of all the suppliers who have supplied part with pid=P1

$$\{ t.Sid \mid t \in Catalog \wedge t.Pid = 'P_1' \}$$

$$\{ t \mid \exists C \in Catalog \, (C.Pid = 'P_1' \wedge t.Sid = C.Sid) \}$$

#Q. Consider the following relational schema:

Suppliers(sid, sname, city)

Parts(pid, pname, color)

Catalog(sid, pid, cost)

Find sids of all the suppliers who have supplied some 'red' color parts

o/p from Catalog

$$\{ t.sid \mid t \in Catalog \wedge \exists P \in Parts (t.Pid = P.Pid \wedge P.Color = 'Red') \}$$

$$\{ t \mid \exists C \in Catalog (\exists P \in Parts (C.Pid = P.Pid \wedge P.Color = 'Red') \wedge t.sid = C.sid) \}$$

$$\{ t \mid \exists C \in Catalog (t.Sid = C.Sid \wedge \exists P \in Parts (C.Pid = P.Pid \wedge P.Color = 'Red')) \}$$

#Q. Consider the following relational schema:

Suppliers(sid, sname, city)

Parts(pid, pname, color)

Catalog(sid, pid, cost)

Find Sid and Sname of all the Suppliers who have Supplied Some 'Red' Color parts.

$$\{ t.Sid, t.Sname \mid t \in Supplier \wedge \exists C \in Catalog (\exists P \in Parts (C.Pid = P.Pid \wedge P.Color = 'Red') \wedge t.Sid = C.Sid) \}$$

$$\{ t.Sid, t.Sname \mid t \in Supplier \wedge \exists C \in Catalog (t.Sid = C.Sid \wedge \exists P \in Parts (C.Pid = P.Pid \wedge P.Color = 'Red')) \}$$

#Q. Consider the following relational schema:

Suppliers(sid, sname, city)

Parts(pid, pname, color)

Catalog(sid, pid, cost)

Retrieve Sids of suppliers who supplied some 'Red' Color parts along with the 'pname' of that Red Color Part.

$\{ t.Sid, t.pname \mid$ but possible

$\{ t \mid \exists C \in Catalog \, (\exists P \in Parts \, (C.Pid = P.Pid \wedge P.Color = 'Red' \wedge t.Sid = C.Sid \wedge t.pname = P.pname)) \}$

Topic : Unsafe TRC query

$\{ t \mid t \notin \text{Relation} \}$

or

$\{ t \mid \neg(t \in \text{Relation}) \}$

Here, we are trying to select the tuples that does not belong to the relation

tuples belonging to relation

Not, the tuples belonging to relation

If domain of any attribute happens to be infinite, then Unsafe TRC query may produce infinite number of tuples.

Note :-

① Unsafe TRC query may produce infinite number of tuples.

② Given an unsafe TRC query, we can never write an Equivalent Relational algebra Expression

③ For Every safe TRC query we can always write an Equivalent Relational Algebra Expression

* Power of TRC (Safe + Unsafe) $\neq$ Power of Relational Algebra

Not Equal because of Unsafe TRC query

Power of Safe TRC = Power of Relational Algebra

#Q. Which of the following tuple relational calculus expression(s) is/are equivalent to $\forall t \in r(P(t))$?

H.W

I $\neg\exists t \in r(P(t))$

II $\exists t \notin r(P(t))$

III $\neg\exists t \in r(\neg P(t))$

IV $\neg\exists t \notin r(\neg P(t))$

A. I only

B. II only

C. III only

D. III and IV only

#Q. Which of the following relational calculus expressions is not safe?

H.W

A. $\{t|\exists u \in R1\ (t[A] = u[A]) \wedge \neg\exists s \in R2\ (t[A] = s[A]\}$

B. $\{t|\forall u \in R1\ (u[A] = \text{“x”} \Rightarrow \exists s \in R2\ (t[A]=s[A] \wedge s[A] = u[A]))\}$

C. $\{t|\neg(t \in R1)\}$

D. $\{t|\exists u \in R1\ (t[A] = u[A]) \wedge \exists s \in R2\ (t[A] = s[A])\}$

#Q. H.W

Consider the relation employee (name, sex, supervisorName) with name as the key. Supervisor Name gives the name of the supervisor of the employee under consideration. What does the following Tuple Relational Calculus query produce?

$$\{e.name \mid employee(e) \wedge (\forall x)\ [\neg employee(x) \vee x.supervisorName \neq e.name \vee x.sex = \text{“male”}]\}$$

A. Names of employees with a male supervisor.

B. Names of employees with no immediate male subordinates

C. Names of employees with no immediate female subordinates

D. Names of employees with a female supervisor.

#Q. Consider The Following Relational Scheme

H.W.

Student (school-id, sch-roll-no, sname, saddress)

School (school-id, sch-name, sch-address, sch-phone)

Enrolment (school-id, sch-roll-no, erollno, examname)

ExamResult (Erollno, examname, marks)

Consider the following tuple relational calculus query

$$\{t \mid \exists E \in \text{Enrolment}\ t = E.\text{school-id} \wedge$$

$$\mid \{x \mid x \in \text{Enrolment} \wedge x.\text{school-id} = t \wedge (\exists B \in \text{ExamResult}\ B.\text{erollno} = x.\text{erollno} \wedge$$

$$B.\text{examname} = x.\text{examname} \wedge B.\text{marks} > 35\} \mid \div$$

$$\mid \{x \mid x \in \text{Enrolment} \wedge x.\text{school-id} = t\} \mid * 100 > 35\}$$

If a student needs to score more than 35 marks to pass an exam what does the query return?

Options on next Slide

#Q.

A. The empty set

B. schools with more than 35% of its students enrolled in some exam or the other

C. schools with a pass percentage above 35% over all exams taken together

D. schools with a pass percentage above 35% over each exam

#Q. H.W.

Consider a database that has the relation schemas
EMP(EmpId, EmpName, DeptId) and
DEPT(DeptName, DeptId).
Note that the DeptId can be permitted to a NULL in the relation EMP. Consider the following queries on the database expressed in tuple relational calculus.

I. $\{t \mid \exists u \in EMP\ (t[EMPName] = u[EmpName] \wedge \forall v \in DEPT\ (t[DeptId] \neq DeptId]))\}$

II. $\{t \mid \exists u \in EMP\ (t[EMPName] = u[EmpName] \wedge \exists v \in DEPT\ (t[DeptId] \neq DeptId]))\}$

III. $\{t \mid \exists u \in EMP\ (t[EMPName] = u[EmpName] \wedge \exists v \in DEPT\ (t[DeptId] = DeptId]))\}$

Which of the above queries are safe?

A. I and II only
B. I and III only
C. II and III only
D. I, II and III

2 mins Summary

Topic Practice questions

Topic Tuple relational calculus (TRC)

Slide

THANK - YOU

Slide