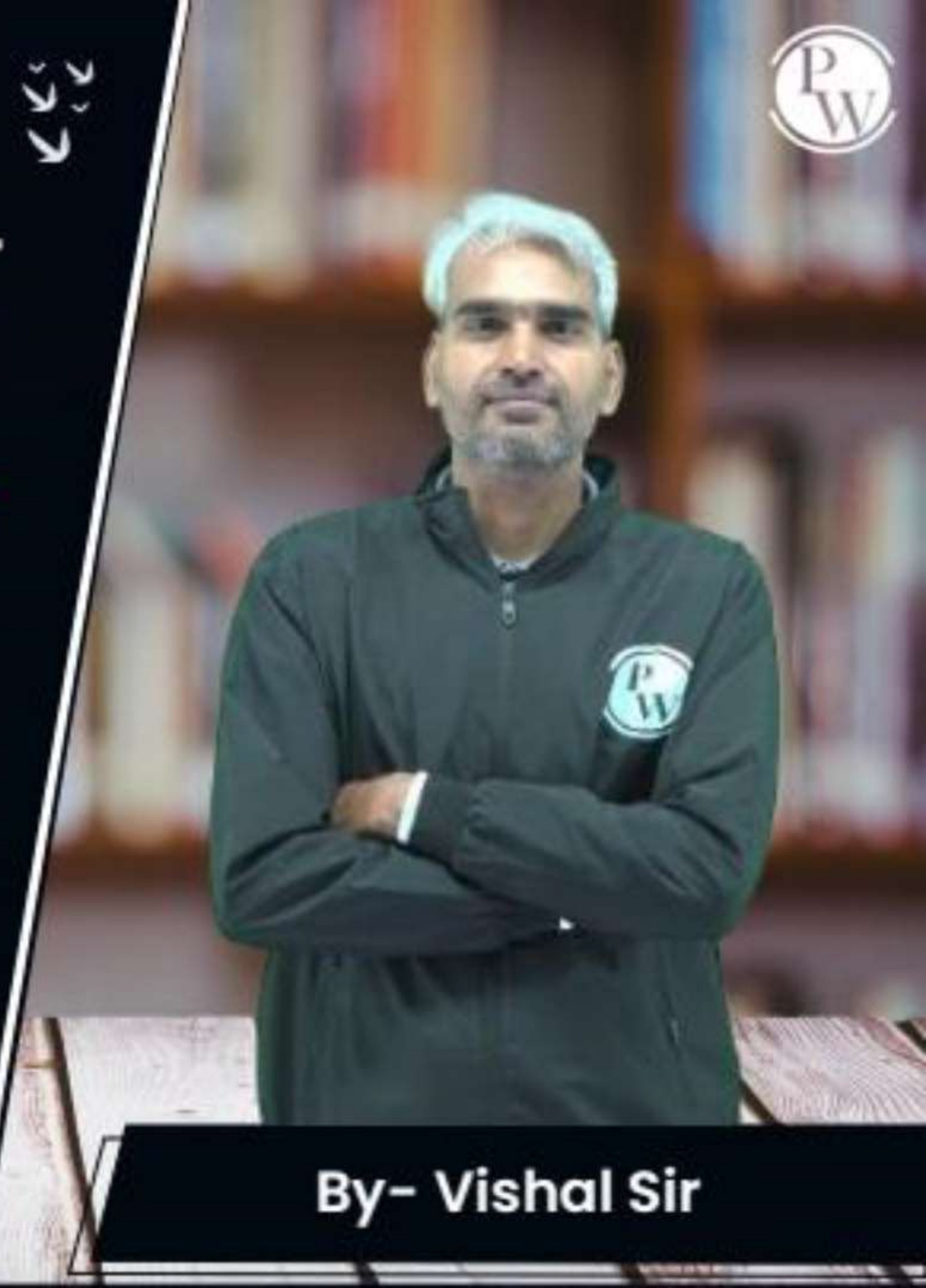
Computer Science & IT

Database Management System

Query Languages

Lecture No. 11



By- Vishal Sir

Recap of Previous Lecture

✓ **Topic** Practice questions on SQL

Slide

Topics to be Covered

✓ **Topic** Relational calculus

✓ **Topic** Tuple relational calculus (TRC)

Slide

Topic : Relational Calculus

∃ : There exists

∀ : for all

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

  ∃, ∀, ∧, ∨, - - -

- 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)

R.A. Query: $\sigma_{marks > 50}(\text{Student})$ ✓

$\pi_{sid}(\sigma_{marks>50}(\text{student}))$ tuples

t:

t.sid

SQL:- Select distinct *
from Student
where (Marks > 50)

Select distinct Sid
from Student
where (Marks > 50)

Student

Sid	Sname	Marks	
S_1	A	45	×
S_2	A	58	✓
S_3	B	51	✓
S_4	C	40	×
S_5	D	55	✓

Topic : Syntax of TRC

tuple variable

Predicate formula over tuple variable

Basic syntax of TRC is $\{t \mid P(t)\}$

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). $t \in \text{Relation} \equiv \text{Relation}(t)$
3. If t is a tuple variable, “t[A]” / “t.A” is value of attribute A in tuple t.
4. P is a predicate formula similar to that of the predicate calculus.

NOTE: In $\{\, t \mid P(t) \,\}$

It is called output tuple variable, it is used to produce output.

Topic : Equivalence between SQL, RA and TRC

SQL ➤ SELECT distinct * FROM Employee { output will be complete Employee table }

R.A. ➤ (Employee) { output will be Complete Employee table }

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

or

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

$$\{\, t \mid \underbrace{t \in \text{Employee}}_{\text{Predicate Cond}^n} \,\}$$

(or)

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

Topic : Free and Bounded variables

- A variable which is preceded by a $\underbrace{\text{quantifier}}_{\forall \text{ or } \exists}$ is called bounded variable.

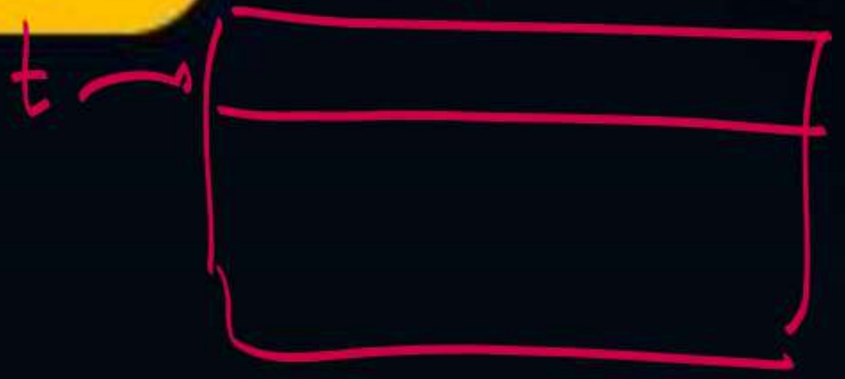
- A variable which is not preceded by any quantifier is called free variable.

- 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}

Employee(t)

or

{ t[ename] | t ∈ Employee}

Employee(t)

or

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

or

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

$\{ \underbrace{t.ename}_{\text{output info}} \mid t \in Employee \}$

Creation of attribut 'ename' in output tuple 't'

Creation of attribute w.r.t. output tuple 't'

$$\{\; t \;\mid\; \exists e \in \text{Employee} \wedge (e.ename = t.ename) \;\}$$

output tuple variable

'e' is a variable Used to represent a tuple of Employee table

Not same as output tuple variable

∴ It must be bounded

Or

$t.ename = e.ename$

Q: Consider the following relation

Supplier (Sid, Sname, rating)

Catalog (Sid, Pid, Cost)

Parts (Pid, Pname, Color)

Q: Retrieve name of the supplier and different cost at which supplier supplied the parts.

$\cancel{t.Sname \mid t \in Supplier}$

$\cancel{t.cost \mid t \in Catalog}$

$\{\cancel{t.Sname, t.cost \mid t \in ?}\}$

$$\{ t. \mid \exists S \in Supplier \left(\exists C \in Catalog \, (C.Sid = S.Sid) \land (t.Sname = S.Sname \land t.cost = C.Cost) \right) \}$$

#Q. Retrieve tuples from the relation students where marks are more than 50
Student(<u>sid</u>, name, marks)

① $\{ t \mid t \in \text{Student}\cancel{\text{s}} \land t.\text{marks} > 50 \}$

② $\{ t \mid t \in \text{Student}\cancel{\text{s}} (t.\text{marks} > 50) \}$

③ $$\left\{ t \mid \exists S \in \text{Student} \left(S.\text{marks} > 50 \land \left(t.\text{sid} = S.\text{sid} \land t.\text{name} = S.\text{name} \land t.\text{marks} = S.\text{marks} \right) \right) \right\}$$

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

Student(sid, name, marks)

① $\{ t.\text{sid} \mid t \in \text{Student}\cancel{\text{s}} \wedge t.\text{marks} > 50 \}$

② $\{ t \mid \exists S \in \text{Student} (S.\text{marks} > 50 \wedge (t.\text{sid} = S.\text{sid})) \}$

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

Student(sid, name, marks)

① $\{ t.Sid, t.name \mid t \in Students (t.marks > 50) \}$

② $\{ t \mid \exists s \in Student (s.marks > 50 \wedge (t.Sid = s.Sid \wedge t.name = s.name)) \}$

#Q. Consider the following relational schema:

Suppliers(<u>sid</u>, sname, city)

Parts(<u>pid</u>, pname, color)

Catalog(<u>sid</u>, <u>pid</u>, cost)

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

① $\{ t.sid \mid t \in Catalog \wedge (t.pid == P_1) \}$

② $\{ t \mid \exists C \in Catalog ((C.pid == P_1) \wedge (t.sid = C.sid)) \}$

#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

① $\{ t.Sid \mid \underline{t \in} Catalog \wedge \underline{\exists P \in Parts} (t.Pid = P.Pid \wedge P.Color = 'Red') \}$ ✓

② $\{ t \mid \exists C \in Catalog (\exists P \in Parts (C.Pid = P.Pid \wedge P.Color = 'Red') \wedge t.Sid = C.Sid) \}$

③ $\{ t \mid \exists C \in Catalog (\underline{t.Sid = C.Sid} \wedge \exists P \in Parts (C.Pid = P.Pid \wedge P.Color = 'Red')) \}$

2 mins Summary

✓ **Topic** Relational calculus

✓ **Topic** Tuple relational calculus (TRC)

Slide

THANK - YOU

Slide