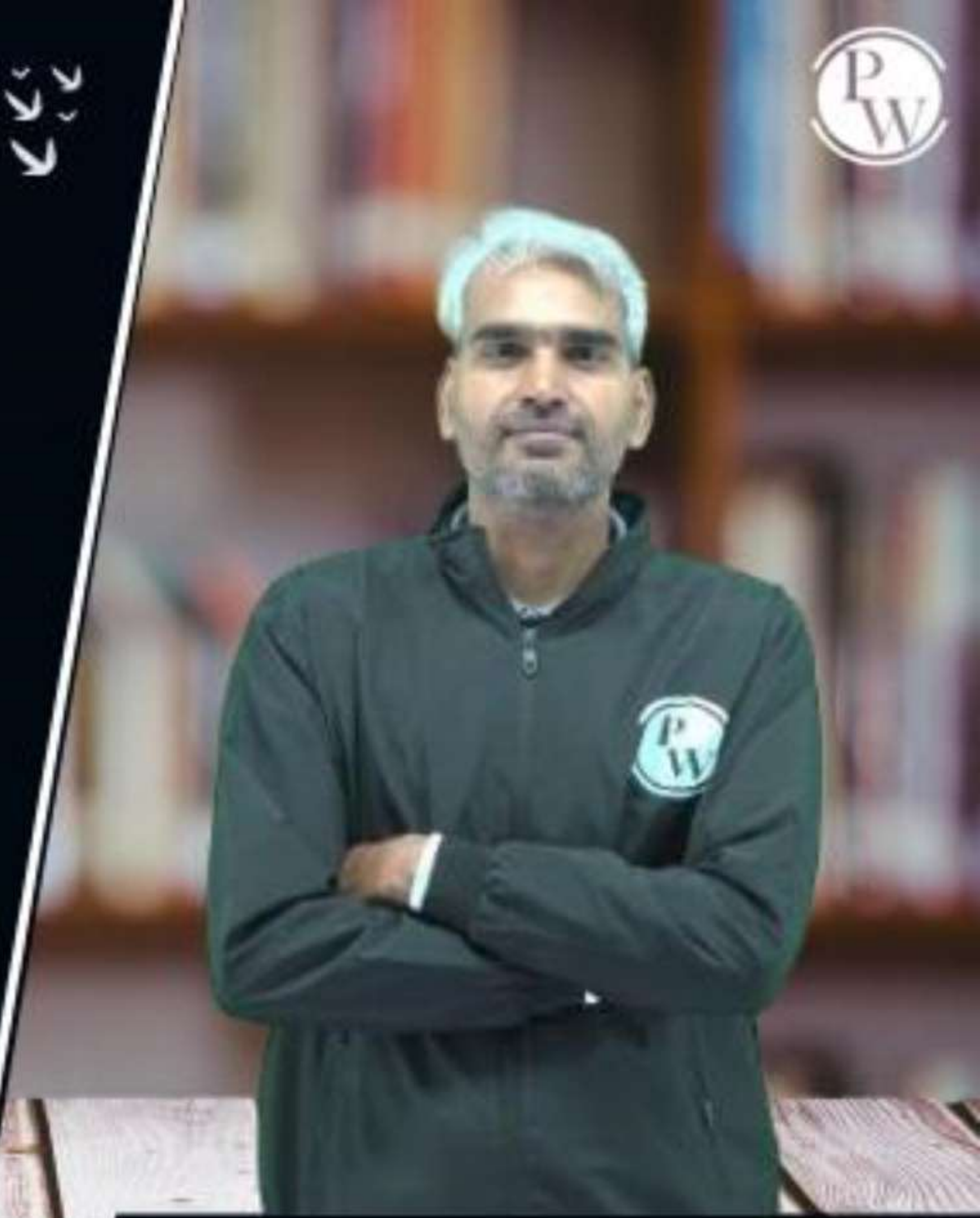
Computer Science & IT

Database Management System

Query Languages

Lecture No. 01



By- Vishal Sir

Recap of Previous Lecture

- Topic — Multivalued dependency
- Topic — Properties of multivalued dependency
- Topic — Fourth normal form (4NF)

Slide

Topics to be Covered

- **Topic** Query Languages
- **Topic** Relational algebra
- **Topic** Basic relational algebra operations

Slide

Topic : Query languages

Query Languages

Procedural query Language

We define, what to retrieve from the database, and we also define the procedure to retrieve that data from the database

eg. Relational Algebra is a Procedural query language

Non-procedural query languages

We define what to retrieve from the database, and we use "Syntax" provided by Non-procedural query languages to retrieve that data from database

eg. (i) Structured query language (SQL)
(ii) Tuple relational Calculus (TRC)
(iii) Domain relational Calculus (DRC)

Note :- Query Condition evaluates tuple by tuple, taken only one tuple at a time,

∴ If we want to Compare two or more tuples of the same table or different tables then we need to $\underbrace{\text{to Combine those tuples}}$ into a single tuple

to Combine the tuples We use join operation

Cross join (Cross Product) (or) Natural join (or) Some other join opn

Note :- Relational algebra query will always produce distinct tuples { No duplicate tuples in O/P }

Topic : Relational Algebra

Relational Algebra is a procedural query language used to query the relational database tables to access data.

* Relational Algebra operation can be classified into two types:
  1) Basic Relational Algebra Operations
  2) Derived Relational Algebra operations

They can be derived using Basic relational Algebra opn.

Topic : Basic Relational Algebra operators

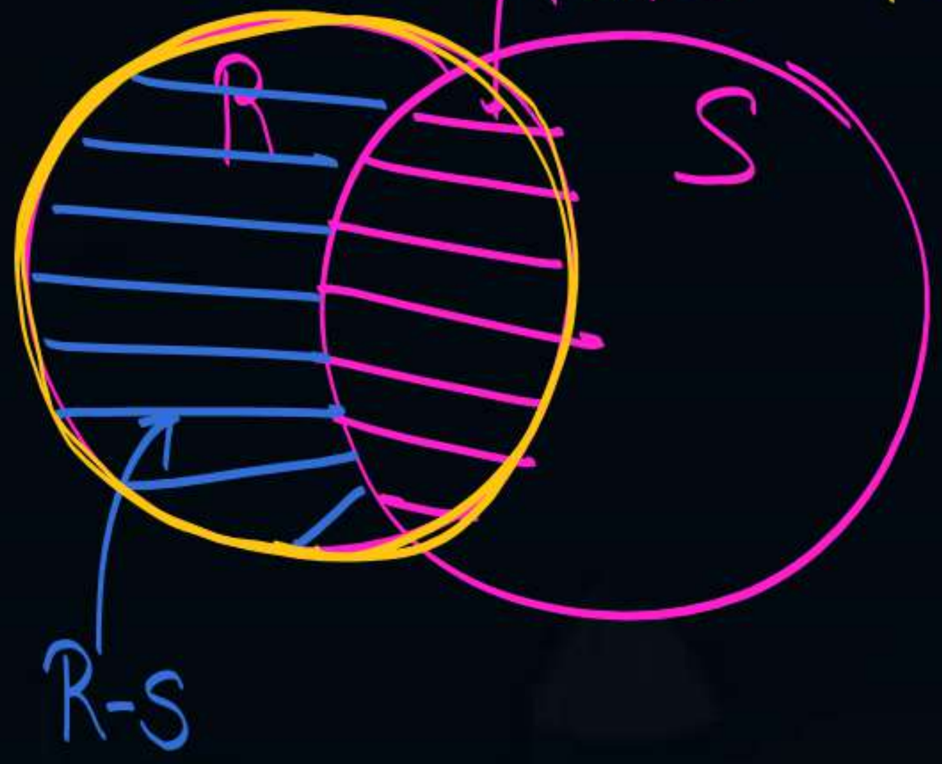
1. Projection(π)
2. Selection (σ)
3. Cross Product ($\times$)
4. Union ($\cup$)
5. Set Difference ($-$)
6. Rename (ρ)

Set operation

Topic : Derived Relational Algebra operators

$R \cap S = R - (R - S)$

1. Intersection (∩)] Set operation
2. Join Operations ("⋈")
3. Division Operation (÷)



Topic : Projection (π)

It is used to project the column data from a relation based on the attributes specified with projection operation.

Projection

Name of relation

e.g., $\pi_{<\text{attribute list}>}(R)$

list of attributes

Projection operator does not obey commutative property i.e.

$$\pi_{<list2>}(\pi_{<list1>}(R)) \neq \pi_{<list1>}(\pi_{<list2>}(R))$$

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_1	CS
S_3	C_2	IT

$$\pi_{\underbrace{Sid, Cid, Branch}_{\text{all attributes of relation R}}}(R) \equiv (R)$$

all attributes of relation R

Output will be Complete relation R

If we do not specify projection operation, then all attributes of relⁿ will be present in O/P

* Retrieve all Sids from relation R.

$$\pi_{Sid}(R) \equiv$$

O/P

Sid
S_1
S_2
S_3

* Retrieve all Cids from relation R.

$$\pi_{Cid}(R) \equiv$$

O/P

Cid
C_1
~~C_1~~
C_2

O/P

Cid
C_1
C_2

duplicate tuples are not present in O/P

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_1	CS
S_3	C_2	IT

$\pi_{sid,\ cid}(R) \equiv$

O/P

Sid	Cid
S_1	C_1
S_2	C_1
S_3	C_2

tuple wise they are distinct

$$\underbrace{\pi_{\underbrace{sid}_{\text{list 2}}}\left(\pi_{\underbrace{sid,cid}_{\text{list 1}}}(R)\right)}_{\text{it will execute}} \neq \pi_{\overbrace{sid,cid}^{\text{List 1}}}\left(\pi_{\overbrace{sid}^{\text{List 2}}}(R)\right)$$

in the O/P there is no attribute Cid

$\therefore$ invalid query

Topic : Selection(σ)

{rows}

It is used to select the tuples from underlying relation based on the predicate condition specified with selection operation.

Selection

Relation

$$\sigma_{\underbrace{<selection\_condition>}_{Cond^n}}(R)$$

NOTE:

$$\sigma_{A \wedge B}(R) = \sigma_{B \wedge A}(R)$$

OR

$$\sigma_B(\sigma_A(R)) = \sigma_A(\sigma_B(R))$$

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_1	CS
S_3	C_2	IT

* Retrieve the tuples Corresponding to Cid = C_1

$$\sigma_{Cid = C_1} (R)$$

O/P :

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_1	CS

Q: Retrieve Sids of the students Who enrolled for Cid = C_1.

$$\pi_{sid} \left(\underbrace{\sigma_{Cid = C_1} (R)}_{\text{Select tuples in which } Cid = C_1} \right)$$

from the selected tuples it will project "Sids"

O/P :

Sid
S_1
S_2

Q:

$$\sigma_{Cid = C_1 \wedge Branch = IT} (R)$$

$\wedge$: and

O/P ⇒

Sid	Cid	Branch

No tuple will be present in o/p

Topic : Selection(σ)

$$\sigma_{Branch=IT}\left(\sigma_{Cid=C_1}(R)\right) = \sigma_{Cid=C_1}\left(\sigma_{Branch=IT}(R)\right)$$

NOTE:

$$\left(\sigma_B(R) \wedge \sigma_A(R)\right) = \left(\sigma_A(R) \wedge \sigma_B(R)\right)$$

$$\sigma_{A \wedge B}(R) = \sigma_{B \wedge A}(R)$$

OR

$$\sigma_B(\sigma_A(R)) = \sigma_A(\sigma_B(R))$$

Note:- $\sigma_{A \vee B}(R) = \sigma_{B \vee A}(R)$

$$\sigma_A(R) \vee \sigma_B(R) = \sigma_B(R) \vee \sigma_A(R)$$

Topic : Cross Product (×)

/ Cross Join

Cross-product is a binary operation. Let R and S are any two relation, then cross product R × S will result in all attributes of R followed by all attribute of S with all possible combinations of tuples from R and S.

R — X attributes

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_1	CS
S_3	C_2	IT

m-tuples

S — Y attributes

Sid	Sname
S_1	A
S_2	B
S_3	A

n-tuples

"X+Y" attributes in RXS

RXS =

R.Sid	R.Cid	R.Branch	S.Sid	S.Sname
S_1	C_1	CS	S_1	A
S_1	C_1	CS	S_2	B
S_1	C_1	CS	S_3	A
S_2	C_1	CS	S_1	A
S_2	C_1	CS	S_2	B
S_2	C_1	CS	S_3	A
S_3	C_2	IT	S_1	A
S_3	C_2	IT	S_2	B
S_3	C_2	IT	S_3	A

m.n tuples

R

Sid	Cid	Branch
S_1	C_1	CS
S_2	C_1	CS
S_3	C_2	IT

S

Sid	Sname
S_1	A
S_2	B
S_3	A

→ Retrieve names of student who enrolled for Course Cid = C_1

$R \times S =$

R.Sid	R.Cid	R.Branch	S.Sid	S.Sname
S_1	C_1	CS	S_1	A
S_1	C_1	CS	S_2	B
S_1	C_1	CS	S_3	A
S_2	C_1	CS	S_1	A
S_2	C_1	CS	S_2	B
S_2	C_1	CS	S_3	A
S_3	C_2	IT	S_1	A
S_3	C_2	IT	S_2	B
S_3	C_2	IT	S_3	A

$$= \pi_{S.Sname}\left(\sigma_{(R.Cid = C_1 \wedge R.Sid = S.Sid)}(R \times S)\right) = \text{O/P}$$

S.Sname
A
B

Topic : Union, Set difference, Intersection

- Union, Set Difference and Intersection are the Set operations.

- To use set theory operators on any two relations, those relations must be union compatible.

- The union compatibility of relations implies that the participating relations must fulfil the following conditions.

1. Same degree, i.e. The two relations must have the same number of attributes.

and

2. Same domain of each corresponding attributes of relations

eg. Let R =

Sid	Sname	Marks

S:

Sid	f_name

① No. of attributes in R ≠ No. of attributes in S

↳ 1st Condn is not satisfied

∴ R & S are not union Compatible

int int

e.g. Let R =

Sid	marks

int Char

S :

Sid	f_name

① No. of attributes in R = No. of attributes in S

└ 1^{st} Cond^{n} is satisfied

② Domain of 2^{nd} attribute of R is 'int'
Whereas Domain of 2^{nd} attribute of S is 'Char'
∴ 2^{nd} Cond^{n} is not satisfied
└→ Hence R&S are not union Compatible

eg. Let

int   char

R :

Sid	Sname

int   char

S :

Sid	f_name

① No. of attributes in R = No. of attributes in S

└ 1^{st} Cond^{n} is satisfied

② Domain of 1^{st} attributes of R = Domain of 1^{st} attribute of S

Domain of 2^{nd} attributes of R = Domain of 2^{nd} attribute of S

$\Big\}$ → 2^{nd} Condition is also satisfied

* Both Cond^{n} of union Compatibility are satisfied for R & S

Hence R & S are union Compatible

Note: ① If relations are union Compatible, then set operations Can be peformed on those relations.

Note: ② After the set operation resulting relation will take the names of its attribute from left hand side relation

ie ① In RUS names of attributes will be same as names of attributes in relation R.

② In SUR names of attributes will be same as names of attributes in relation S.

2 mins Summary

Topic Query Languages

Topic Relational algebra

Topic Basic relational algebra operations

Slide

THANK - YOU

Slide