CS & IT ENGINEERING

Database Management System

DBMS

Lecture No. 5

By- Ravindrababu Ravula Sir

Recap of Previous Lecture

Topic

NAm

Topics to be Covered

- Topic — Queries
- Topic — Relational Algebra
- Topic — SOME,EVERY,ATLEAST
- Topic — MAXIMUM, MINIMUM

Topic: Prerequisites

- Set Theory of Discrete Mathematics

Topic: Prerequisites

- Set Theory of Discrete Mathematics
- First Order logic (or) Predicate Calculus

Topic: Queries

Queries

- Procedural Query Language
- Non-Procedural Query Language

Topic: Queries

Queries

Procedural Query Language

- Formulation of
  HOW to retrieve
  WHAT data is retrieved
  FROM DB Tables

Non-Procedural Query Language

Topic: Queries

Queries

Procedural Query Language

- Formulation of
  HOW to retrieve
  WHAT data is retrieved
  FROM DB Tables

Non-Procedural Query Language

- Formulation of
  WHAT data is retrieved
  FROM DB Tables

Topic: Queries

Queries

Procedural Query Language

- Formulation of
  HOW to retrieve
  WHAT data is retrieved
  FROM DB Tables

Relational Algebra Queries

Non-Procedural Query Language

- Formulation of
  WHAT data is retrieved
  FROM DB Tables

Topic: Queries

Queries

Procedural Query Language

- Formulation of
  HOW to retrieve
  WHAT data is retrieved
  FROM DB Tables

Relational Algebra Queries

Non-Procedural Query Language

- Formulation of
  WHAT data is retrieved
  FROM DB Tables

Relational Calculus Queries

Topic: Queries

Queries

Procedural Query Language

Relational Algebra Queries

Non-Procedural Query Language

Relational Calculus Queries

Topic: Queries

Queries

Procedural Query Language

Relational Algebra Queries

Non-Procedural Query Language

Relational Calculus Queries

[User, FOL, PC formulas]

Topic: Queries

Queries

Procedural Query Language

Relational Algebra Queries

Non-Procedural Query Language

Relational Calculus Queries

[User, FOL, PC formulas]

Tuple Relational Calculus (TRC)

Topic: Queries

Queries

Procedural Query Language

Relational Algebra Queries

Non-Procedural Query Language

Relational Calculus Queries

[User, FOL, PC formulas]

- Tuple Relational Calculus (TRC)
- Domain Relational Calculus (DRC)

Topic: Queries

Queries

Procedural Query Language

Relational Algebra Queries

Non-Procedural Query Language

Relational Calculus Queries

[User, FOL, PC formulas]

Tuple Relational Calculus (TRC)

SQL Queries are Close to TRC (IBM)

Domain Relational Calculus (DRC)

Topic: Queries

Queries

Procedural Query Language

Relational Algebra Queries

Non-Procedural Query Language

Relational Calculus Queries

[User, FOL, PC formulas]

Tuple Relational Calculus (TRC)

SQL Queries are Close to TRC
(IBM)

~~Domain Relational Calculus (DRC)~~

~~Queries By Example(QBE)~~
(Microsoft)

Topic: Relational Algebra

- Relational Algebra Result is always distinct record.

No duplicate values

Topic: Relational Algebra

- Relational Algebra Result is always distinct record.

Basic Operators:

π : Projection

ρ : Rename

σ : Selection

Topic: Relational Algebra

- Relational Algebra Result is always distinct record.

Basic Operators:

π : Projection

ρ : Rename

σ : Selection

X : Cross Product

$\cup$: Union

– : Set difference

Topic: Relational Algebra

- Relational Algebra Result is always distinct record.

Basic Operators:

- π : Projection
- ρ : Rename
- σ : Selection

→ Unary Operators → one table

- X : Cross Product
- ∪ : Union
- – : Set difference

→ Binary Operators → two tables

Topic: Relational Algebra

- Relational Algebra Result is always distinct record.

Basic Operators:

- π : Projection
- ρ : Rename
- σ : Selection

Unary Operators

- X : Cross Product
- ∪ : Union
- – : Set difference

Binary Operators

Derived Operators:

- ⋈ : Join (Using X, σ, π)

Topic: Relational Algebra

- Relational Algebra Result is always distinct record.

Basic Operators:

- π : Projection
- ρ : Rename
- σ : Selection

Unary Operators

- X : Cross Product
- $\cup$: Union
- – : Set difference

Binary Operators

Derived Operators:

- $\bowtie$: Join (Using X, σ, π)
- $\cap$: Intersection (Using –)

Topic: Relational Algebra

- Relational Algebra Result is always distinct record.

Basic Operators:

- π : Projection
- ρ : Rename
- σ : Selection

Unary Operators

- X : Cross Product
- $\cup$: Union
- – : Set difference

Binary Operators

Derived Operators:

- $\bowtie$: Join (Using X, σ, π)
- $\cap$: Intersection (Using –)
- $\div$ or / : Division (Using X, π, –)

Topic: Relational Algebra

- Queries not only works for RDBMS table, it works for table with duplicate entries as well.

RA

NO duplicates

Topic: Relational Algebra

- Queries not only works for RDBMS table, it works for table with duplicate entries as well.
- These table have bag of records → duplicate allowed

  where as RDBMS have set of records → No duplicates allowed

Topic: Relational Algebra

Projection (π):

- Used to project required attributes from R.

Topic: Relational Algebra

Projection (π):

- Used to project required attributes from R.
- Denoted by π_{attr_list} (R)

Topic: Relational Algebra

Projection (π):

- Used to project required attributes from R.
- Denoted by π_{attr_list} (R)

Ex: π_{BC} (R)

R

A	B	C
4	6	8
8	5	4
7	3	4
7	3	4

Topic: Relational Algebra

Projection (π):

- Used to project required attributes from R.
- Denoted by π_{attr_list} (R)

Ex: π_{BC} (R)

R

A	B	C
4	6	8
8	5	4
7	3	4
7	3	4

- The result is:

B	C
6	8
5	4
3	4

Unique or distinct rows only.

Topic: Relational Algebra

Selection (σ):

- Used to retrieve records of R that satisfies the given condition (P)
- Denoted by σ_P (R), where P is condition

Topic: Relational Algebra

Selection (σ):

- Used to retrieve records of R that satisfies the given condition (P)
- Denoted by σ_P (R), where P is condition

Ex: $\sigma_{A>5}$ (R)

R

A	B	C
4	6	8
8	5	4
7	3	4
7	3	4

Topic: Relational Algebra

Selection (σ):

- Used to retrieve records of R that satisfies the given condition (P)
- Denoted by σ_P (R), where P is condition

Ex: $\sigma_{A>5}$ (R)

R

A	B	C
4	6	8
8	5	4
7	3	4
7	3	4

- The result is:

Q	B	C
8	5	4
7	3	4

} Distinct tuples

Inspiring Stories : Frida Kahlo

Background: **Mexican girl, survived polio and accident.**

Education: **Studied art during long recovery and pain.**

Achievements: **Became a world-famous painter despite pain.**

Impact: **Icon of strength and creativity.**

Topic: Relational Algebra

#Q. Which one of the following are true?

(i) π is Commutative

(ii) σ is Commutative

A Only (i)

B Only (ii)

C (i) and (ii)

D None of the two

Topic: Relational Algebra

#Q. Which one of the following are true?

(i) π is Commutative (ii) σ is Commutative

A Only (i) **B** Only (ii) **C** (i) and (ii) **D** None of the two

Sol.: Understanding the Question,

(i) $\pi_{List2}(\pi_{List1}(R))$ and $\pi_{List1}(\pi_{List2}(R))$

Topic: Relational Algebra

#Q. Which one of the following are true?

(i) π is Commutative (ii) σ is Commutative

A Only (i) **B** Only (ii) **C** (i) and (ii) **D** None of the two

Sol.: Understanding the Question,

(i) $\pi_{List2}(\pi_{List1}(R))$ and $\pi_{List1}(\pi_{List2}(R))$

Ex: $\pi_{AB}(\pi_{ABC}(R))$ and $\pi_{ABC}(\pi_{AB}(R))$

Topic: Relational Algebra

#Q. Which one of the following are true?

(i) π is Commutative (ii) σ is Commutative

A Only (i) **B** Only (ii) **C** (i) and (ii) **D** None of the two

Sol.: Understanding the Question,

(i) $\pi_{List2}(\pi_{List1}(R))$ and $\pi_{List1}(\pi_{List2}(R))$

Ex: $\pi_{AB}(\pi_{ABC}(R))$ and $\pi_{ABC}(\pi_{AB}(R))$ ✗

π_{ABC} cannot select ABC from AB so invalid.

Topic: Relational Algebra

#Q. Which one of the following are true?

(i) π is Commutative (ii) σ is Commutative

A Only (i) **B** Only (ii) **C** (i) and (ii) **D** None of the two

Sol.: Understanding the Question,

(i) $\pi_{List2}(\pi_{List1}(R))$ and $\pi_{List1}(\pi_{List2}(R))$

Ex: $\pi_{AB}(\pi_{ABC}(R))$ and $\pi_{ABC}(\pi_{AB}(R))$

π_{ABC} cannot select ABC from AB so invalid.

Both are not equal.

π is not commutative.

Topic: Relational Algebra

(ii) $\sigma_{C2}(\sigma_{C1}(R))$ and $\sigma_{C1}(\sigma_{C2}(R))$

To Prove lets take examples for both:

R

A	B	C
4	6	8
8	5	4
7	3	4
7	3	4

$\sigma_{B>3}(\sigma_{A>5}(R))$

A > 5

A	B	C
8	5	4
7	3	4

B > 3

A	B	C
8	5	4

and

$\sigma_{A>5}(\sigma_{B>3}(R))$

B > 3

A	B	C
4	6	8
8	5	4

A > 5

A	B	C
8	5	4

Both are equal. σ is commutative. Also, $\sigma_{C1}(\sigma_{C2}(R)) = \sigma_{C2}(\sigma_{C1}(R)) = \sigma_{C1 \wedge C2}(R)$

Topic: Relational Algebra

#Q. Which one of the following are true?

(i) π is Commutative

(ii) σ is Commutative

A Only (i)

B Only (ii)

C (i) and (ii)

D None of the two

Topic: Relational Algebra

Cross Product (×):

- R × S : All attributes of R followed by all attributes of S

and

Every record of R pairs with every record of S.

Topic: Relational Algebra

Cross Product (×):

- R × S : All attributes of R followed by all attributes of S

and

Every record of r pairs with every record of S.

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

⇒

R×S

A	B	C	C	D
4	2	5	5	7
4	2	5	7	8
6	5	8	5	7
6	5	8	7	8
3	4	5	5	7
3	4	5	7	8

Topic: Relational Algebra

- X = No. of attributes in R & Y = No. of attributes in S

  then, R × S contain (X + Y) No. of attributes.

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

⇒

R×S

A	B	C	C	D
4	2	5	5	7
4	2	5	7	8
6	5	8	5	7
6	5	8	7	8
3	4	5	5	7
3	4	5	7	8

Topic: Relational Algebra

- X = No. of attributes in R & Y = No. of attributes in S

  then, R × S contain (X + Y) No. of attributes.

R

A	B	C
4	2	5
6	5	8
3	4	5

X

S

C	D
5	7
7	8

Y

$\Rightarrow$

R×S

A	B	C	C	D
4	2	5	5	7
4	2	5	7	8
6	5	8	5	7
6	5	8	7	8
3	4	5	5	7
3	4	5	7	8

(X+Y)

Topic: Relational Algebra

- n = No. of distinct tuples in R m = No. of distinct tuples in S

  then, R × S contains n × m number of tuples.

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

$\Rightarrow$

R×S

A	B	C	C	D
4	2	5	5	7
4	2	5	7	8
6	5	8	5	7
6	5	8	7	8
3	4	5	5	7
3	4	5	7	8

Topic: Relational Algebra

- n = No. of distinct tuples in R m = No. of distinct tuples in S

  then, R × S contains n × m number of tuples.

R

A	B	C
4	2	5
6	5	8
3	4	5

n

S

C	D
5	7
7	8

m

⇒

R×S

A	B	C	C	D
4	2	5	5	7
4	2	5	7	8
6	5	8	5	7
6	5	8	7	8
3	4	5	5	7
3	4	5	7	8

n x m

Topic: Relational Algebra

#Q. A Relation R has n distinct tuples and Relation S has 0 tuples then R × S contains how many tuples?

Topic: Relational Algebra

#Q. A Relation R has n distinct tuples and Relation S has 0 tuples then R × S contains how many tuples?

Sol.

R

A	B
4	2
7	2

S

B	C

} 0

Topic: Relational Algebra

#Q. A Relation R has n distinct tuples and Relation S has 0 tuples then R × S contains how many tuples?

Sol.

R

A	B
4	2
7	2

S

B	C

$\Rightarrow$

R×S

A	B	B	C

0

Zero tuples

Topic: Relational Algebra

#Q. A Relation R has n distinct tuples and Relation S has 0 tuples then R × S contains how many tuples?

Sol.

R

A	B
4	2
7	2

S

B	C

⇒

R×S

A	B	B	C

Zero tuples

∴ R × S has zero tuples.

→ R × S result is empty record set if either R or S relation have 0 tuples.

Topic: Joins

JOINS:

1. Natural Join ($\bowtie$) :

$$R \bowtie S = \pi_{\text{distinct attr}} \left(\sigma_{\text{equality btn common attr of R and S}} (R \times S) \right)$$

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of R ⋈ S?

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of $R \bowtie S$?

Sol. $R \bowtie S = \pi_{ABCD} (\sigma_{R.C = S.C} (R \times S))$

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of $R \bowtie S$?

Sol. $R \bowtie S = \pi_{ABCD} (\sigma_{R.C = S.C} (R \times S))$

R×S

A	B	C	C	D
4	2	5	5	7
4	2	5	7	8
6	5	8	5	7
6	5	8	7	8
3	4	5	5	7
3	4	5	7	8

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of R ⋈ S?

Sol. $R \bowtie S = \pi_{ABCD} (\sigma_{R.C = S.C} (R \times S))$

R×S

A	B	C	C	D	
4	2	5	5	7	✓
4	2	5	7	8	
6	5	8	5	7	
6	5	8	7	8	
3	4	5	5	7	✓
3	4	5	7	8	

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of $R \bowtie S$?

Sol. $R \bowtie S = \pi_{ABCD} (\sigma_{R.C = S.C} (R \times S))$

$R \bowtie S$

A	B	C	D
4	2	5	7
3	4	5	7

$R \times S$

A	B	C	C	D
4	2	5	5	7
4	2	5	7	8
6	5	8	5	7
6	5	8	7	8
3	4	5	5	7
3	4	5	7	8

Topic: Joins

→ If there are no common attributes in the relation then

$R \bowtie S = R \times S$ ✓

Ex: for R(AB), S(CD).

Topic: Joins

Ex: For two relation R(ABC) and S(BCDE).

What is R ⋈ S?

Topic: Joins

Ex: For two relation R(ABC) and S(BCDE).

What is $R \bowtie S$?

Sol. $R \bowtie S = \pi_{ABCDE}(\sigma_{R.B = S.B \wedge R.C = S.C}(R \times S))$

Topic: Joins

Ex: For two relation R(ABC) and S(BCDE).

What is $R \bowtie S$?

Sol. $R \bowtie S = \pi_{ABCD} (\sigma_{R.B = S.B \wedge R.C = S.C} (R \times S))$

R(ABC) and S(BCDE) , RXS = ABCBCDE

2 B's and 2 C's are present. So, each one B and C are dropped.

Topic: Joins

JOINS:

$\bowtie_C$

2. Conditional Join ($\bowtie_C$) :

Joined based on the condition.

$R \bowtie_C S = \sigma_C(R \times S)$

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of $R \bowtie_{R.C>S.C} S$?

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of $R \bowtie_{R.C>S.C} S$?

Sol. From conditional join eq. $R \bowtie_{R.C>S.C} S = \sigma_{R.C>S.C} (R \times S)$

Topic: Joins

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of $R \bowtie_{R.C>S.C} S$?

Sol. From conditional join eq. $R \bowtie_{R.C>S.C} S = \sigma_{R.C>S.C} (R \times S)$

R×S

A	B	C	C	D	
4	2	5	5	7	
4	2	5	7	8	
6	5	8	5	7	✓
6	5	8	7	8	✓
3	4	5	5	7	
3	4	5	7	8	

Topic: Joins

$\times$ → Repeat

$\bowtie$ → no Repeat

$\bowtie_C$ → Repeat

Ex:

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Then what is the result of $R \bowtie_{R.C>S.C} S$?

Sol. From conditional join eq. $R \bowtie_{R.C>S.C} S = \sigma_{R.C>S.C} (R \times S)$

A	B	C	C	D
6	5	8	5	7
6	5	8	7	8

R×S

A	B	C	C	D	
4	2	5	5	7	
4	2	5	7	8	
6	5	8	5	7	✓
6	5	8	7	8	✓
3	4	5	5	7	
3	4	5	7	8	

Topic: Outer Joins

OUTER JOINS:

1. Left Outer Join ($\bowtie$) :

   $R \bowtie S = (R \bowtie S) \cup$ (records of R that are not considered in join condition with NULL values in remaining attributes)

Topic: Outer Joins

OUTER JOINS:

1. Left Outer Join (⟕) :

$R ⟕ S = (R ⋈ S) \cup$ (records of R that are not considered in join condition with NULL values in remaining attributes)

Example:

R

A	B	C	D
4	2	5	7
3	4	5	7
6	5	8	NULL

Rows from $R ⋈ S$

Topic: Outer Joins

OUTER JOINS:

1. Left Outer Join (⟕) :

$R \leftouterjoin S = (R \bowtie S) \cup$ (records of R that are not considered in join condition with NULL values in remaining attributes)

Example:

R

A	B	C	D
4	2	5	7
3	4	5	7
6	5	8	NULL

Rows from $R \bowtie S$

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Topic: Outer Joins

OUTER JOINS:

2. Right Outer Join ($\bowtie$) :

$R \bowtie S = (R \bowtie S) \cup$ (records of S that are not considered in join condition with NULL values in remaining attributes)

Topic: Outer Joins

OUTER JOINS:

2. Right Outer Join (⟖) :

$R \rightouterjoin S = (R \bowtie S) \cup$ (records of S that are not considered in join condition with NULL values in remaining attributes)

Example:

R

A	B	C	D
4	2	5	7
3	4	5	7
NULL	NULL	7.	. 8

Rows from $R \bowtie S$

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Topic: Outer Joins

OUTER JOINS:

3. Full Join ($\bowtie$) :

$$R \bowtie S = (R \bowtie S) \cup (R \bowtie S)$$

Topic: Outer Joins

OUTER JOINS:

3. Full Join (⟗) :

$$R \fullouterjoin S = (R \bowtie S) \cup (R \leftouterjoin S)$$

Example:

R

A	B	C	D
4	2	5	7
3	4	5	7
6	5	8	NULL
NULL	NULL	7	8

Rows from $R \bowtie S$

R

A	B	C
4	2	5
6	5	8
3	4	5

S

C	D
5	7
7	8

Topic: Outer Joins

Ex.: Conditional left outer join

$$R \bowtie_{R.C > S.C} S$$

Topic: Outer Joins

Ex.: Conditional left outer join

$R \bowtie_{R.C > S.C} S$

A	B	C	C	D
6	5	8	5	7
6	5	8	7	8
4	2	5	NULL	NULL
3	4	5	NULL	NULL

R×S

A	B	C	C	D	
4	2	5	5	7	
4	2	5	7	8	
6	5	8	5	7	✓
6	5	8	7	8	✓
3	4	5	5	7	
3	4	5	7	8	

S

C	D
5	7
7	8

R

A	B	C	
4	2	5	
6	5	8	✓
3	4	5	

Topic: Join Queries

Join Queries:

Joins two Relations R(A ...) , S(B ...) on some conditions.

Topic: Join Queries

Join Queries:

Joins two Relations R(A ...) , S(B ...) on some conditions.

Ex. Retrieve "A" values of R that are more than atleast one "B" value of S.

Some
any

R

A
10
20
30
40

S

B
15
25
30

Topic: Join Queries

Join Queries:

Joins two Relations R(A ...) , S(B ...) on some conditions.

Ex. Retrieve "A" values of R that are more than atleast one "B" value of S.

R

A
10
20
30
40

S

B
15
25
30

Sol. There are many ways to write a query, one of them is:

i.e., $\pi_A (\sigma_{R.A > S.B} (R \times S))$

Topic: Join Queries

Join Queries:

Joins two Relations R(A ...) , S(B ...) on some conditions.

Ex. Retrieve "A" values of R that are more than atleast one "B" value of S.

R

A
10
20
30
40

S

B
15
25
30

Sol. There are many ways to write a query, one of them is:

$$\pi_{A}(\sigma_{R.A > S.B}\ (R \times S))$$

R×S

A	B
10	15
10	25
10	30
20	15
20	25
⋮	⋮
⋮	⋮

Topic: Join Queries

Join Queries:

Joins two Relations R(A ...) , S(B ...) on some conditions.

Ex. Retrieve "A" values of R that are more than atleast one "B" value of S.

R

A
10
20
30
40

S

B
15
25
30

Sol. There are many ways to write a query, one of them is:

$\pi_A (\sigma_{R.A > S.B} (R \times S))$

i.e.,

R×S

A	B
10	15
10	25
10	30
20	15
20	25
⋮	⋮
⋮	⋮

R.A > S.B

Project A

A	B
20	15
30	15
30	25
40	15
40	25
40	30

A
20
30
40

Topic: Join Queries

This can also be written as:

$$\pi_A(R \bowtie_{R.A>S.B} S) \cong \pi_A\left(\sigma_{R.A>S.B}(R \times S)\right)$$

Topic: Join Queries

This can also be written as:

$$\pi_A(R \bowtie_{R.A>S.B} S) \cong \pi_A\left(\sigma_{R.A>S.B}(R \times S)\right)$$

some/any/atleast one

$R \bowtie S$ is used if common attr exist

Else $R \bowtie_C S = \sigma_C (R \times S)$

Topic: Join Queries

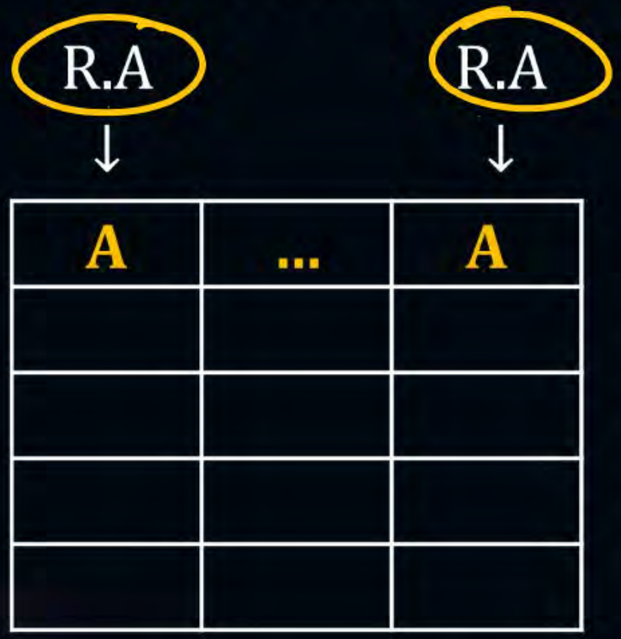
Ex.: Now lets say we want to join a relation R with itself

$\pi_{attr}\ (\sigma_{Condition}\ (R \times R))$

Topic: Join Queries

Ex.: Now lets say we want to join a relation R with itself

$\pi_{attr}\ (\sigma_{Condition}\ (R \times R))$



Topic: Join Queries

Ex.: Now lets say we want to join a relation R with itself

$\pi_{attr}\ (\sigma_{Condition}\ (R \times R))$

R.A	...	R.A
A		A

→ There will be ambiguity in selecting the attributes.

Topic: Join Queries

Ex.: Stud (Sid, age) Enroll (Sid, Cid)

$$\pi_{\text{Stud.Sid}} (\sigma_{\text{Stud.Sid} = \text{Enroll.Sid}} (\text{Stud} \times \text{Enroll}))$$

Topic: Join Queries

Ex.: Stud (Sid, age) Enroll (Sid, Cid)

$$\pi_{\text{Stud.Sid}} (\sigma_{\text{Stud.Sid} = \text{Enroll.Sid}} (\text{Stud} \times \text{Enroll}))$$

Stud.Sid

Enroll.Sid

Sid	age	Sid	Cid

Topic: Join Queries

Ex.: Stud (Sid, age) Enroll (Sid, Cid)

$\pi_{Stud.Sid}\ (\sigma_{Stud.Sid = Enroll.Sid}\ (Stud \times Enroll))$

No Ambiguity

Stud.Sid

Enroll.Sid

Sid	age	Sid	Cid

Topic: Join Queries

Ex.: Now we want to compare one student data with other.

Topic: Join Queries

Ex.: Now we want to compare one student data with other.

Sol. Stud (Sid, age)

Topic: Join Queries

Ex.: Now we want to compare one student data with other.

Sol. Stud (Sid, age)

$\pi_{\cdot}(\sigma_{\cdot}(\text{Stud} \times \text{Stud}))$

Topic: Join Queries

Ex.: Now we want to compare one student data with other.

Sol. Stud (Sid, age)

$\pi\ (\sigma\ (\text{Stud} \times \text{Stud}))$

In this case applying predicate and projection will be ambiguous

$\pi_{\text{Stud.Sid}}\ (\sigma_{\text{Stud.age} > \text{Stud.age}}\ (\text{Stud} \times \text{Stud}))$

Topic: Join Queries

Ex.: Now we want to compare one student data with other.

Sol. Stud (Sid, age)

π (σ (Stud × Stud))

In this case applying predicate and projection will be ambiguous

$\pi_{Stud.Sid}$ ($\sigma_{Stud.age > Stud.age}$ (Stud × Stud))

The expression is invalid due to Ambiguity.

- To avoid this ambiguity, we go for Renaming.

Inspiring Stories : Ram Nath Kovind

Background: **Born in a mud house in UP.**

Education: **Studied law with great effort.**

Achievements: **Became President of India.**

Impact: **Proved humble roots can lead to the top.**

Topic: Join Queries

Rename (ρ):

- Used to rename table name or attribute names.

Topic: Join Queries

Rename (ρ):

- Used to rename table name or attribute names.

(i) Renaming table name Stud(Sid, Sname, age)

ρ (Temp, Stud) ⇒ Temp (Sid, Sname, age)

Topic: Join Queries

Rename (ρ):

- Used to rename table name or attribute names.

(i) Renaming table name Stud(Sid, Sname, age)

ρ (Temp, Stud) $\Rightarrow$ Temp (Sid, Sname, age)

(ii) Renaming attributes Sid to I, Sname to N, age to A

$\rho_{I, N, A}$ (Stud) $\Rightarrow$ Stud (I, N, A)

Topic: Join Queries

Rename (ρ):

- Used to rename table name or attribute names.

(i) Renaming table name Stud(Sid, Sname, age)

ρ (Temp, Stud) $\Rightarrow$ Temp (Sid, Sname, age)

(ii) Renaming attributes Sid to I, Sname to N, age to A

$\rho_{I, N, A}$ (Stud) $\Rightarrow$ Stud (I, N, A)

Or, the other way to do it, on Stud (Sid, Sname, age)

$\rho_{Sid \to I,\ age \to A}$ (Stud) $\Rightarrow$ Stud (I, Sname, A)

Topic: Join Queries

#Q. Now for $\pi_{Attr}(\sigma_{Condition}(R \times R))$

R.A ↓	...	R.A ↓
A	...	A

Topic: Join Queries

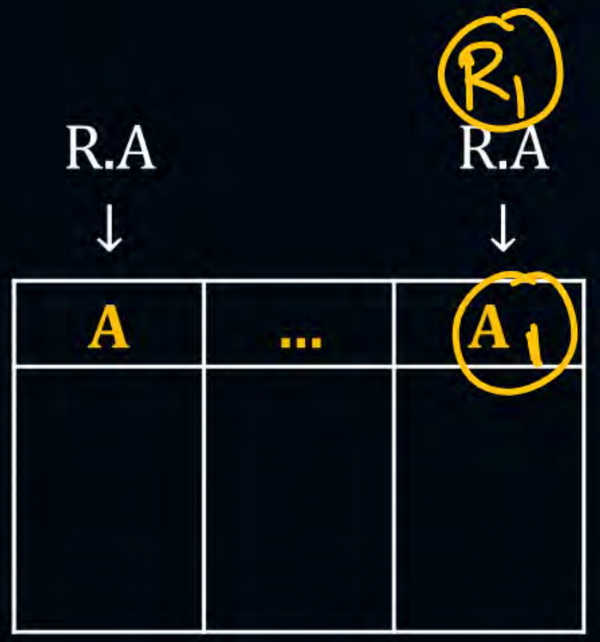
#Q. Now for $\pi_{Attr}\ (\sigma_{Condition}\ (R \times R))$

Sol. Rename the second R to R_1 or some other name

R.A	...	R.A
A		A

Topic: Join Queries

#Q. Now for $\pi_{Attr}\ (\sigma_{Condition}\ (R \times R))$



Sol. Rename the second R to R_1 or some other name

Or,

Rename the attribute in condition that causes Ambiguity into some other name.

R.A and $R.A_1$

Resolved the ambiguity

Topic: Join Queries

#Q. $\pi_{\text{Stud.Sid}}\ (\sigma_{\text{Stud.age} > \text{Stud.age}}\ (\text{Stud} \times \text{Stud}))$

Stud ↓ Stud ↓

Sid	age	Sid	age

Topic: Join Queries

#Q. $\pi_{Stud.Sid}\ (\sigma_{Stud.age > Stud.age}\ (Stud \times Stud))$

Sol. Rename one of the table i.e., second stud to temp.

Now the selection condition doesn't have ambiguity.

$\pi_{Stud.Sid}\ (\sigma_{Stud.age > temp.age}\ (Stud \times temp))$

Stud ↓ | Temp (Stud) ↓

Sid	age	Sid	age

Topic: Join Queries

Stud ↓		Stud ↓	
Sid	**age**	**Sid**	**age**

#Q. $\pi_{Stud.Sid}\ (\sigma_{Stud.age > Stud.age}\ (Stud \times Stud))$

Sol. Rename one of the table i.e., second stud to temp.

Now the selection condition doesn't have ambiguity.

$\pi_{Stud.Sid}\ (\sigma_{Stud.age > temp.age}\ (Stud \times temp))$

Or, Rename the attributes of second relation

i.e., Sid → i, age → A We can write the equation as:

$\pi_{Stud.Sid}\ (\sigma_{Stud.age > Stud.A}\ (Stud \times \rho_{Sid \to i,\ age \to a}\ (Stud)))$

∴ The ambiguity is resolved.

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol. We can use conditional join

Emp (eid, salary, gender) $\Uparrow$ Female

Emp (eid, salary, gender) $\Uparrow$ Male

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol. We can use conditional join

Emp (eid, salary, gender) ⇑ Female

Emp (eid, salary, gender) ⇑ Male

$$\sigma_{gender=female}(Emp) \times \sigma_{gender=male}(Emp)$$

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol. We can use conditional join

Emp (eid, salary, gender) Emp (eid, salary, gender)

$$\sigma_{gender=female}(Emp) \times \sigma_{gender=male}(Emp)$$

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol. We can use conditional join

Emp (eid, salary, gender) ⇑ Female

Emp (eid, salary, gender) ⇑ Male

$$\sigma_{\text{Emp.Salary} > \text{Emp.Salary}} (\sigma_{\text{gender=female}} (\text{Emp}) \times \sigma_{\text{gender=male}} (\text{Emp}))$$

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol. We can use conditional join

Emp (eid, salary, gender) ⇑ Female Emp (eid, salary, gender) ⇑ Male

$$\pi_{eid}\,(\sigma_{Emp.Salary > Emp.Salary}\,(\sigma_{gender=female}\,(Emp) \times \sigma_{gender=male}\,(Emp)))$$

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol. We can use conditional join

Emp (eid, salary, gender) ⇑ Female Emp (eid, salary, gender) ⇑ Male

$$\pi_{eid}\ (\sigma_{Emp.Salary > Emp.Salary}\ (\sigma_{gender=female}\ (Emp) \times \sigma_{gender=male}\ (Emp)))$$

- This is Ambigous, so lets rename the second relation.

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol Ext.

$$\pi_{eid}\,(\sigma_{Emp.Salary > Emp.Salary}\,(\sigma_{gender=female}\,(Emp) \times \sigma_{gender=male}\,(Emp)))$$

Topic: Join Queries

I S G

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol Ext.

$$\pi_{eid}\ (\sigma_{Emp.Salary > Emp.Salary}\ (\sigma_{gender=female}\ (Emp) \times \sigma_{gender=male}\ (Emp)))$$

$$\sigma_{gender=female}\ (Emp) \times \rho_{I,S,G}\ (\sigma_{gender=male}\ (Emp))$$

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol Ext.

$$\pi_{eid}\ (\sigma_{Emp.Salary > Emp.Salary}\ (\sigma_{gender=female}\ (Emp) \times \sigma_{gender=male}\ (Emp)))$$

$$\sigma_{Sal>s}\ (\sigma_{gender=female}\ (Emp) \times \rho_{I,S,G}\ (\sigma_{gender=male}\ (Emp)))$$

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol Ext.

$$\pi_{eid}\ (\sigma_{Emp.Salary > Emp.Salary}\ (\sigma_{gender=female}\ (Emp) \times \sigma_{gender=male}\ (Emp)))$$

$$\pi_{eid}\ (\sigma_{Sal>s}\ (\sigma_{gender=female}\ (Emp) \times \rho_{I,S,G}\ (\sigma_{gender=male}\ (Emp))))$$

Topic: Join Queries

#Q. Write a RA Query for the following problem.Emp (Eid, Sal, gen),

Retrieve female emp id's whose salary is more than salary of some male emp

(some = any or atleast one)

Sol Ext.

$$\pi_{eid}\ (\sigma_{Emp.Salary > Emp.Salary}\ (\sigma_{gender=female}\ (Emp) \times \sigma_{gender=male}\ (Emp)))$$

$$\pi_{eid}\ (\sigma_{Sal>s}\ (\sigma_{gender=female}\ (Emp) \times \rho_{I,S,G}\ (\sigma_{gender=male}\ (Emp))))$$

$$\Rightarrow \quad \pi_{eid}\ (\sigma_{gender=female}\ (Emp) \bowtie_{Sal>s} \rho_{I,S,G}\ (\sigma_{gender=male}\ (Emp)))$$

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp

eid	Salary	dept No.
e1	10	4
e2	20	5
e3	30	4
e4	40	5
e5	50	4

I S D

Emp

eid	Salary	dept No.
e1	10	4
e2	20	5
e3	30	4
e4	40	5
e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp

eid	Salary	dept No.
e1	10	4
e2	20	5
e3	30	4
e4	40	5
e5	50	4

	I	S	D
Emp	eid	Salary	dept No.
	e1	10	4
✓	e2	20	5
	e3	30	4
✓	e4	40	5
	e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp

eid	Salary	dept No.
e1	10	4
e2	20	5
e3	30	4
e4	40	5
e5	50	4

I S D

Emp

eid	Salary	dept No.
e1	10	4
e2	20	5
e3	30	4
e4	40	5
e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp ×

eid	Salary	dept No.
e1	10	4
e2	20	5
e3	30	4
e4	40	5
e5	50	4

I S D

Emp

eid	Salary	dept No.
e1	10	4
e2	20	5
e3	30	4
e4	40	5
e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp	eid	Salary	dept No.
×	e1	10	4
×	e2	20	5
	e3	30	4
	e4	40	5
	e5	50	4

	I	S	D
Emp	eid	Salary	dept No.
	e1	10	4
	e2	20	5
	e3	30	4
	e4	40	5
	e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp	eid	Salary	dept No.
×	e1	10	4
×	e2	20	5
✓	e3	30	4
	e4	40	5
	e5	50	4

	I	S	D
Emp	eid	Salary	dept No.
	e1	10	4
	e2	20	5
	e3	30	4
	e4	40	5
	e5	50	4

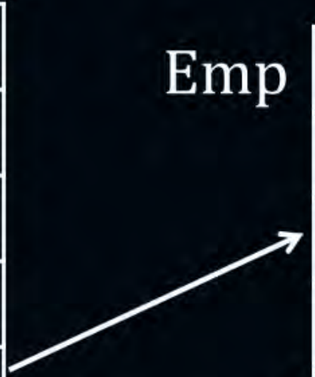
Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp	eid	Salary	dept No.
×	e1	10	4
×	e2	20	5
✓	e3	30	4
✓	e4	40	5
	e5	50	4



	I	S	D
Emp	eid	Salary	dept No.
	e1	10	4
	e2	20	5
	e3	30	4
	e4	40	5
	e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp	eid	Salary	dept No.
✗	e1	10	4
✗	e2	20	5
✓	e3	30	4
✓	e4	40	5
✓	e5	50	4

	I	S	D
Emp	eid	Salary	dept No.
	e1	10	4
	e2	20	5
	e3	30	4
	e4	40	5
	e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

Emp	eid	Salary	dept No.
×	e1	10	4
×	e2	20	5
✓	e3	30	4
✓	e4	40	5
✓	e5	50	4

$\sigma_{deptNo=5}(Emp)$

	I	S	D
Emp	eid	Salary	dept No.
	e1	10	4
✓	e2	20	5
	e3	30	4
✓	e4	40	5
	e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join.

$$Emp \bowtie_{Sal>S} \rho_{I,S,D}\left(\sigma_{deptNo=5}(Emp)\right)$$

I S D

Emp	eid	Salary	dept No.
×	e1	10	4
×	e2	20	5
✓	e3	30	4
✓	e4	40	5
✓	e5	50	4

Emp	eid	Salary	dept No.
	e1	10	4
✓	e2	20	5
	e3	30	4
✓	e4	40	5
	e5	50	4

Topic: Join Queries

#Q. Write a RA Query on the below problem. Emp(eid, Salary, deptNo)

Retrieve empId's whose salary is more than salary of some dept 5 employee.

Sol. we go for conditional join. $\pi_{eid}\left(Emp \bowtie_{Sal>S} \rho_{I,S,D}\left(\sigma_{deptNo=5}(Emp)\right)\right)$

Emp	eid	Salary	dept No.
✗	e1	10	4
✗	e2	20	5
✓	e3	30	4
✓	e4	40	5
✓	e5	50	4

I ✓ S ✓ D ✓

Emp	eid	Salary	dept No.
	e1	10	4
✓	e2	20	5
	e3	30	4
✓	e4	40	5
	e5	50	4

Topic: Join Queries

Conclusion:

$\pi_A (R \bowtie_{R.A > S.B} S)$

→ A conditional join retrieves the data of the table which matches SOME or ANY or ATLEAST ONE of other relation.

Topic: Join Queries

#Q. Write a RA Query for the following problem.

Stud (Sid, age) Dept (deptid, age)

Retrieve Sid's whose age is more than some dept.

Topic: Join Queries

#Q. Write a RA Query for the following problem.

Stud (Sid, age) Dept (deptid, age)

Retrieve Sid's whose age is more than some dept.

some → any → atleast 1

Sol. Simple using conditional join

π_{Sid} (Stud $\bowtie_{Stud.age > Dept.age}$ Dept)

Topic: Queries

Comparison:

For two relation, R(A . . .) and S(B . . .)

Case 1:

Retrieve A values of R those are more than some B value of S

Case 2:

Retrieve A values of B those are more than every B values of S.

Topic: Queries

Comparison:

For two relation, R(A . . .) and S(B . . .)

Case 1:

Retrieve A values of R those are more than some B value of S

Some = any or atleast one

Case 2:

Retrieve A values of B those are more than every B values of S.

Topic: Queries

Comparison:

For two relation, R(A . . .) and S(B . . .)

Case 1:

Retrieve A values of R those are more than some B value of S

Some = any or atleast one

$\bowtie_c$ is used

$\Pi_A (R \bowtie_{R.A>S.B} S)$

Case 2:

Retrieve A values of B those are more than every B values of S.

Topic: Queries

Comparison:

For two relation, R(A . . .) and S(B . . .)

Case 1:

Retrieve A values of R those are more than some B value of S

Some = any or atleast one

$\bowtie_c$ is used

$\Pi_A (R \bowtie_{R.A>S.B} S)$

Case 2:

Retrieve A values of B those are more than every B values of S.

Cannot be done using $\bowtie_c$

Topic: Queries

Retrieve A values of B those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Topic: Queries

Retrieve A values of B those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value – $\neg$(A > every B)

Topic: Queries

Retrieve A values of R those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value $\Rightarrow$ (All A values) $-\neg$(A > every B) $-$ (A value $\leq$ some B value)

$\sim(A \geq \text{every } B)$

$\leq$ some

Topic: Queries

Retrieve A values of R those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value – $\neg$(A > every B)

$\Rightarrow$(All A values) – (A value $\leq$ some B value)

R

A	
10	
20	
30	
40	

✓

$\bowtie_{R.A<=S.B}$

S

B	
15	
25	
30	

Topic: Queries

Retrieve A values of B those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value – $\neg$(A > every B)

$\Rightarrow$(All A values) – (A value $\leq$ some B value)

R	A	
✓	10	
✓	20	
	30	
	40	

$\bowtie_{R.A<=S.B}$

S	B	
	15	
	25	
	30	

Topic: Queries

Retrieve A values of B those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value – $\neg$(A > every B)

$\Rightarrow$(All A values) – (A value $\leq$ some B value)

R	A	
✓	10	
✓	20	
✓	30	
	40	

$\bowtie_{R.A<=S.B}$

S	B	
	15	
	25	
	30	

Topic: Queries

Retrieve A values of B those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value $-\neg$(A > every B)

$\Rightarrow$(All A values) $-$ (A value $\leq$ some B value)

R

A	
10	
20	
30	
40	

$-$

R

A	
10	
20	
30	
40	

S

B	
15	
25	
30	

$\bowtie_{R.A<=S.B}$

Topic: Queries

Retrieve A values of B those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value $-\neg$(A > every B)

$\Rightarrow$(All A values) – (A value ≤ some B value)

R

	A	
	10	
	20	
	30	
✓	40	

R

	A	
✓	10	
—✓	20	
✓	30	
	40	

$\bowtie_{R.A<=S.B}$

S

B	
15	
25	
30	

Topic: Queries

Retrieve A values of B those are more than every B values of S.

For Case 2, the problem can be solved in so many ways:

Let's solve this way.

A value not > every B value $\Rightarrow$ All A value $-\neg$(A > every B)

$\Rightarrow$(All A values) $-$ (A value $\leq$ some B value)

$$\Pi_A(R) - \Pi_A(R \bowtie_{R.A \leq S.B} S)$$

R

	A	
	10	
	20	
	30	
✓	40	

R

	A	
✓	10	
—✓	20	
✓	30	
	40	

S

B	
15	
25	
30	

$\bowtie_{R.A<=S.B}$

Topic: Queries

Frequent model of question in GATE:

#Q. Emp(eid, dno, sal)

Retrieve eids whose salary is more than every sal of dept 5.

Topic: Queries

Frequent model of question in GATE:

#Q. Emp(eid, dno, sal)

Retrieve eids whose salary is more than every sal of dept 5.

Sol.: eid's sal > every sal of dept 5.

Topic: Queries

Frequent model of question in GATE:

#Q. Emp(eid, dno, sal)

Retrieve eids whose salary is more than every sal of dept 5.

Sol.: eid's sal > every sal of dept 5.

= All employees - ¬(Sal > every sal of dept 5)

Topic: Queries

Frequent model of question in GATE:

#Q. Emp(eid, dno, sal)

Retrieve eids whose salary is more than every sal of dept 5.

Sol.: eid's sal > every sal of dept 5.

= All employees - ¬(Sal ≥ every sal of dept 5)

= All employees – eid's sal ≤ some sal of dept 5.

Topic: Queries

Frequent model of question in GATE:

#Q. Emp(eid, dno, sal)

Retrieve eids whose salary is more than every sal of dept 5.

Sol.: eid's sal > every sal of dept 5.

= All employees - ¬(Sal > every sal of dept 5)

= All employees – eid's sal ≤ some ($\bowtie_c$) sal of dept 5.

Topic: Queries

Frequent model of question in GATE:

#Q. Emp(eid, dno, sal)

Retrieve eids whose salary is more than every sal of dept 5.

Sol.: eid's sal > every sal of dept 5.

= All employees - ¬(Sal > every sal of dept 5)

= All employees – eid's sal ≤ some ($\bowtie_c$) sal of dept 5.

= $\Pi_{eid}(emp) - \Pi_{eid}(emp \bowtie_{sal \leq s} \rho_{I,D,S}(\sigma_{dno=5}(Emp)))$

Topic: Queries

#Q. Emp(eid, salary, gender) Retrieve eid's of female employees whose salary is more than every salary of a male.

Topic: Queries

#Q. Emp(eid, salary, gender) Retrieve eid's of female employees whose salary is more than every salary of a male.

Sol.:

female salary > every male salary

Topic: Queries

#Q. Emp(eid, salary, gender) Retrieve eid's of female employees whose salary is more than every salary of a male.

Sol.:

female salary > every male salary

= (All female employees) – ¬(female salary ≥ every male salary)

Topic: Queries

#Q. Emp(eid, salary, gender) Retrieve eid's of female employees whose salary is more than every salary of a male.

Sol.:

female salary > every male salary

= (All female employees) – ¬(female salary > every male salary)

= All female emp – (female emp ≤ some $(\bowtie_c)$ salary of male emp)

Topic: Queries

#Q. Emp(eid, salary, gender) Retrieve eid's of female employees whose salary is more than every salary of a male.

Sol.:

female salary > every male salary

= (All female employees) – ¬(female salary > every male salary)

= All female emp – (female emp ≤ some ($\bowtie_c$) salary of male emp)

= $\Pi_{eid}(\sigma_{gender=female}(emp)) - \Pi_{eid}(\sigma_{gender=female}(emp) \bowtie_{sal \leq s} \rho_{I,S,G}(\sigma_{gender=male}(emp))))$

All female employees who salary ≤ Some male

Topic: Queries

= In SQL as well, the same logic is used.

= RA → SQL

→ TRC

Topic: Queries

= In SQL as well, the same logic is used.

= RA → SQL

→ TRC

= The basic prerequisites to write optimized queries are

1. Set theory
2. Predicate Calculus & First Order Logic

=> female salary > every male salary

= All females – ¬(female salary > every male salary)

= All females – (female salary ≤ some male salary)

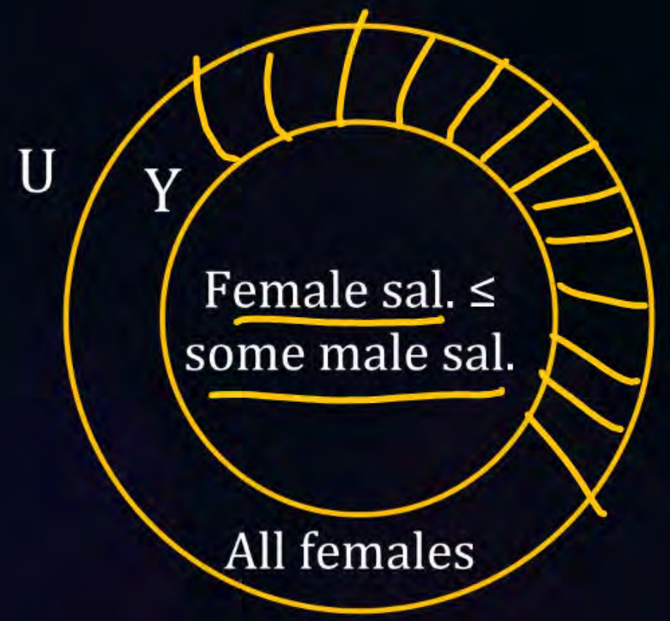
Y

=> female salary > every male salary

= All females – ¬(female salary > every male salary)

= All females – (female salary ≤ some male salary)

Y



U – Y = comp of Y.

Inspiring Stories : N. Krishnan

Background: **Award-winning chef from Tamil Nadu.**

Education: **Culinary arts graduate.**

Achievements: **Quit job to feed homeless people.**

Impact: **Served lakhs of free meals, inspires humanity.**

Topic: Queries

#Q. Emp(eid, sal) Retrieve eids who gets maximum salary.

Topic: Queries

#Q. Emp(eid, sal) Retrieve eids who gets maximum salary.

Sol.: In SQL, we have a max function that gets the max directly but in relational algebra, we have to do with the σ, Π, ρ, etc only

Topic: Queries

#Q. Emp(eid, sal) Retrieve eids who gets maximum salary.

Sol.: In SQL, we have a max function that gets the max directly but in relational algebra, we have to do with the σ, Π, ρ, etc only

= eid's whose salary is maximum

Topic: Queries

#Q. Emp(eid, sal) Retrieve eids who gets maximum salary.

Sol.: In SQL, we have a max function that gets the max directly but in relational algebra, we have to do with the σ, Π, ρ, etc only

= eid's whose salary is maximum

= eid's salary $\geq$ every eid's salary

Topic: Queries

10
20
30
40

40

#Q. Emp(eid, sal) Retrieve eids who gets maximum salary.

Sol.: In SQL, we have a max function that gets the max directly but in relational algebra, we have to do with the σ, Π, ρ, etc only

= eid's whose salary is maximum

= eid's salary ≥ every eid's salary (Cannot say greater than because no such record exists)

Topic: Queries

#Q. Emp(eid, sal) Retrieve eids who gets maximum salary.

Sol.: In SQL, we have a max function that gets the max directly but in relational algebra, we have to do with the σ, Π, ρ, etc only

= eid's whose salary is maximum

= eid's salary ≥ every eid's salary (Cannot say greater than because no such record exists)

= All eid's – ¬(eid's salary ≥ every eid's salary)

Topic: Queries

#Q. Emp(eid, sal) Retrieve eids who gets maximum salary.

Sol.: In SQL, we have a max function that gets the max directly but in relational algebra, we have to do with the σ, Π, ρ, etc only

= eid's whose salary is maximum

= eid's salary ≥ every eid's salary (Cannot say greater than because no such record exists)

= All eid's – ¬(eid's salary ≥ every eid's salary)

= All eid's – (eid's salary < some ($\bowtie_c$) eid's salary)

Π_{eid} (Emp) —

Π_{eid} (Emp $\bowtie_{sal<s}$ $\rho_{I,S}$ (emp))

Eids of employees who salary is < Some employee salary

Π_{eid} — Π_{eid} (Emp $\bowtie_{sal<s}$ $\rho_{I,S}$ (emp))

Emp

eid	sal
e1	40
e2	80
e3	60
e4	40
e5	80

Emp

I	S
e1	40
e2	80
e3	60
e4	40
e5	80

Π_{eid} — Π_{eid} $(Emp \bowtie_{sal<s} \rho_{I,S}(emp))$

Emp

	eid	sal
✓	e1	40
	e2	80
✓	e3	60
✓	e4	40
	e5	80

Emp

I	S
e1	40
e2	80
e3	60
e4	40
e5	80

Π_{eid} — Π_{eid} (Emp $\bowtie_{sal<s}$ $\rho_{I,S}$ (emp))

eid
e1
e2
e3
e4
e5

Emp

eid	sal
e1	40
e2	80
e3	60
e4	40
e5	80

Emp

I	S
e1	40
e2	80
e3	60
e4	40
e5	80

Π_{eid} — Π_{eid} (Emp $\bowtie_{sal<s}$ $\rho_{I,S}$ (emp))

eid
e1
e2
e3
e4
e5

Emp

eid	sal
e1	40
e2	80
e3	60
e4	40
e5	80

Emp

I	S
e1	40
e2	80
e3	60
e4	40
e5	80

eid
e2
e5

$$\Pi_{eid} \quad - \quad \Pi_{eid} \quad (Emp \bowtie_{sal<s} \rho_{I,S} (emp))$$

eid
e1
e2
e3
e4
e5

Emp

	eid	sal
✓	e1	40
	e2	80
✓	e3	60
✓	e4	40
	e5	80

Emp

I	S
e1	40
e2	80
e3	60
e4	40
e5	80

eid
e2
e5

Topic: Queries

#Q. Generic Question

R(A...) max "A" value of R.

Topic: Queries

#Q. Generic Question

R(A...) max "A" value of R.

Sol.: max A value

Topic: Queries

#Q. Generic Question

R(A...) max "A" value of R.

Sol.: max A value

= A value ≥ every A values

Topic: Queries

#Q. Generic Question

R(A...) max "A" value of R.

Sol.: max A value

= A value ≥ every A values

= All A values – ¬(A value ≥ every A value)

Topic: Queries

#Q. Generic Question

R(A...) max "A" value of R.

Sol.: max A value

= A value ≥ every A values

= All A values – ¬(A value ≥ every A value)

= All A values – (A value < some ($\bowtie_c$) A value)

Topic: Queries

#Q. Generic Question

R(A...) max "A" value of R.

Sol.: max A value

= A value ≥ every A values

= All A values – ¬(A value ≥ every A value)

= All A values – (A value < some ($\bowtie_c$) A value)

= $\prod_A - \prod_A (R \bowtie_{A<A1} \rho_{A \to A1} (R))$

Topic: Queries

#Q. Emp(eid, sal)

eid's who gets minimum salary.

Topic: Queries

#Q. Emp(eid, sal)

eid's who gets minimum salary.

Sol.: max = { ≥ every } min = { ≤ every }

Topic: Queries

#Q. Emp(eid, sal)

eid's who gets minimum salary.

Sol.: max = { ≥ every } min = { ≤ every }

Ex. 10

20

30

40

Topic: Queries

#Q. Emp(eid, sal)

eid's who gets minimum salary.

Sol.: max = { ≥ every } min = { ≤ every }

Ex. 10
20
30
40

Value ≤ every value is 10 → min
Value ≥ every value is 40 → max

Topic: Queries

#Q. Emp(eid, sal)

eid's who gets minimum salary.

Sol.: max = { ≥ every } min = { ≤ every }

Ex. 10

20

30

40

Value ≤ every value is 10 → min

Value ≥ every value is 40 → max

= eid's whose salary is min

Topic: Queries

#Q. Emp(eid, sal)

eid's who gets minimum salary.

Sol.: max = { ≥ every } min = { ≤ every }

Ex. 10
20
30
40

Value ≤ every value is 10 → min
Value ≥ every value is 40 → max

= eid's whose salary is min

= eid's ≤ every eid = All eids – ¬(eid ≤ every eid)

Topic: Queries

#Q. Emp(eid, sal)

eid's who gets minimum salary.

Sol.: max = { ≥ every } min = { ≤ every }

Ex. 10
20
30
40

Value ≤ every value is 10 → min
Value ≥ every value is 40 → max

= eid's whose salary is min

= eid's ≤ every eid = All eids – ¬(eid ≤ every eid)

= All eids – (eid > some $(\bowtie_c)$ eid) = $\prod_{eid} - \prod_{eid}$ (Emp $\bowtie_{sal>s}$ $\rho_{I,S}$(Emp))

Not min ✓

Topic: Queries

#Q. Emp(eid, salary, dept No) Eids who get maximum salary for each department.

Sol.: (eids whose salary) ≥ every (every emp of same dept)

eid	sal	dept. No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

Topic: Queries

#Q. Emp(eid, salary, dept No) Eids who get maximum salary for each department.

Sol.: (eids whose salary) ≥ every (every emp of same dept)

= All emp – {eids whose salary < some ($\bowtie_c$) emp of same dept}

Not max

eid	sal	dept. No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

Topic: Queries

I S D

#Q. Emp(eid, salary, dept No) Eids who get maximum salary for each department.

Sol.: (eids whose salary) ≥ every (every emp of same dept)

= All emp – {eids whose salary < some ($\bowtie_c$) emp of same dept}

= $\prod_{eid} - \{Emp \bowtie_{(sal<S) \wedge (dept=D)} \rho_{I,S,D} (Emp)\}$

Not "max"

eid	sal	dept. No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

$$\Pi_{eid} \; - \; \{Emp \; \bowtie_{(sal<S)\wedge(dept=D)} \; \rho_{I,S,D} \; (Emp)\}$$

Emp

eid	sal	dept No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

I S D

eid	Sal	dept. No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

$$\Pi_{eid} \; - \; \{Emp \; \bowtie_{(sal<S)\wedge(dept=D)} \; \rho_{I,S,D} \; (Emp))$$

Emp	eid	sal	dept No.
✓	e1	40	2
	e2	60	2
✓	e3	30	2
✓	e4	30	3
	e5	40	3

eid	Sal	dept. No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

$$\Pi_{eid} \; - \; \{Emp \; \bowtie_{(sal<S)\wedge(dept=D)} \; \rho_{I,S,D} \; (Emp)\}$$

eid
e1
e2
e3
e4
e5

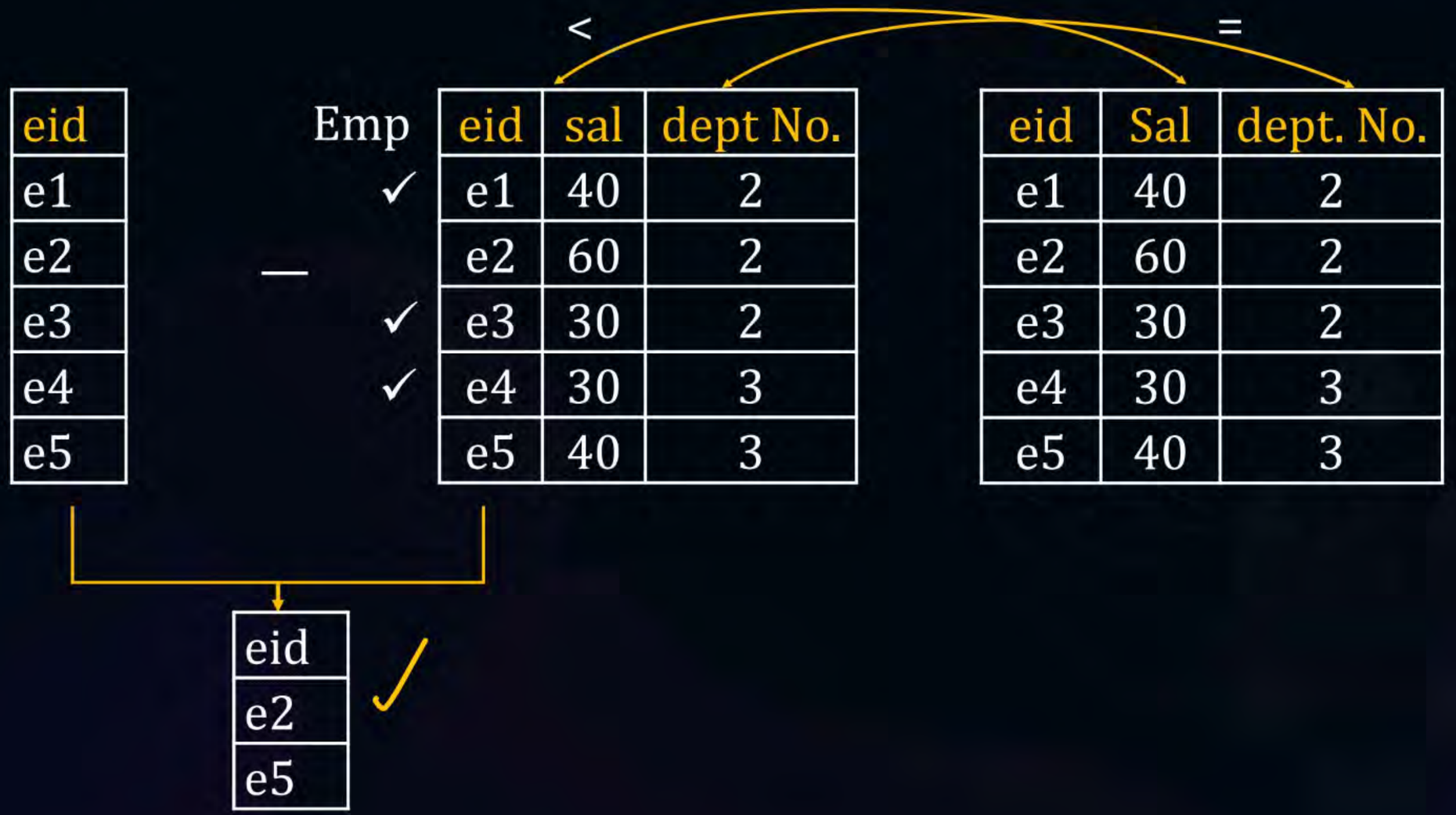
—

Emp
✓
✓
✓

eid	sal	dept No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

eid	Sal	dept. No.
e1	40	2
e2	60	2
e3	30	2
e4	30	3
e5	40	3

$$\Pi_{eid} \; - \; \{Emp \; \bowtie_{(sal<S)\wedge(dept=D)} \; \rho_{I,S,D} \; (Emp)\}$$



Topic: Queries

#Q. family(Parent, child, child DOB)

Retrieve the youngest child of each parent.

parent	child	child DOB
A	B	1995
A	C	1998
A	D	1996
E	F	1996
E	G	1996

Topic: Queries

#Q. family(Parent, child, child DOB)

Retrieve the youngest child of each parent.

parent	child	child DOB
A	B	1995
A	C	1998
A	D	1996
E	F	1996
E	G	1996

Sol.: Child whose DOB ≥ every child DOB of same parent.

Topic: Queries

#Q. family(Parent, child, child DOB)

Retrieve the youngest child of each parent.

parent	child	child DOB
A	B	1995
A	C	1998
A	D	1996
E	F	1996
E	G	1996

Sol.: Child whose DOB ≥ every child DOB of same parent.

= All children – ¬ (Child whose DOB ≥ every child DOB of same parent)

Not younger

Topic: Queries

#Q. family(Parent, child, child DOB)

Retrieve the youngest child of each parent.

parent	child	child DOB
A	B	1995
A	C	1998
A	D	1996
E	F	1996
E	G	1996

Sol.: Child whose DOB ≥ every child DOB of same parent.

= All children – ¬ (Child whose DOB ≥ every child DOB of same parent)

= All children – (Child whose DOB < some ($\bowtie_c$) child DOB of same parent)

Topic: Queries

#Q. family(Parent, child, child DOB)

Retrieve the youngest child of each parent.

P C D

parent	child	child DOB
A	B	1995
A	C	1998
A	D	1996
E	F	1996
E	G	1996

Sol.: Child whose DOB ≥ every child DOB of same parent.

= All children – ¬ (Child whose DOB ≥ every child DOB of same parent)

= All children – (Child whose DOB < some ($\bowtie_c$) child DOB of same parent)

= $\Pi_{child} - \Pi_{child}$ (family $\bowtie_{(child\ DOB < D \wedge parent = P)}$ $\rho_{P,C,D}$(family))

Not younger

Topic: Queries

DIVISION (/ or ÷):

Used to answer queries that involve the keyword "all".

i.e. R1 ÷ R2 gives tuples from R1 that are associated with all tuples of R2.

Ex:

Enrol

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

Course

Cid	---
C1	
C2	
C3	

Retrieve Sid's enrolled in every course of course relation

Ex:

Enrol

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

Course

Cid	---
C1	
C2	
C3	

Retrieve Sid's enrolled in every course of course relation

Sid	Cid	Cid
S1	C1	C1
S1	C2	C2
S1	C3	C3
S2	C1	
S2	C2	
S3	C1	

S1

Ex:

Enrol

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

Course

Cid	---
C1	
C2	
C3	

Retrieve Sid's enrolled in every course of course relation

Sid	Cid	Cid
S1	C1	C1
S1	C2	C2
S1	C3	C3
S2	C1	
S2	C2	
S3	C1	

Ex:

Enrol

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C1

Course

Cid	---
C1	
C2	
C3	

Retrieve Sid's enrolled in every course of course relation

Sid	Cid	Cid
S1	C1	C1
S1	C2	C2
S1	C3	C3
S2	C1	
S2	C2	
S3	C1	

Result

Sid
S1

This is done by $\Pi_{Sid\ Cid}(Enroll) / \Pi_{Cid(course)}$

Topic: Queries

Expansion

#Q. (a) ~~Extension~~ **of division:**

$\prod_{Sid, Cid}(E) / \prod_{Cid}(C) \rightarrow$ Sid enrolled for every Course

(a) Write a relational algebra query for Sids not enrolled for every course.

Topic: Queries

Sol.: Sid's enrolled for a proper subset of all Courses

Sid	Cid
S1	C1
S2	C2
S3	C3

Topic: Queries

Sol.: Sid's enrolled for a proper subset of all Courses

$$(\prod_{Sid}(E) \times \prod_{Cid}(C))$$

Sid
S1
S2
S3

Cid
C1
C2
C3

$\Rightarrow$

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S2	C3
S3	C1
S3	C2
S3	C3

Topic: Queries

Sol.: Sid's enrolled for a proper subset of all Courses

$$(\prod_{Sid}(E) \times \prod_{Cid}(C)) - \prod_{Sid, Cid}(E)$$

Sid	Cid
S1	C1
S2	C2
S3	C3

$\Rightarrow$

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S2	C3
S3	C1
S3	C2
S3	C3

—

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C3

Topic: Queries

Not registered for all the Courses

Sol.: Sid's enrolled for a proper subset of all Courses

$$\Pi_{Sid}(\ (\Pi_{Sid}(E) \times \Pi_{Cid}(C)) - \ \Pi_{Sid, Cid}(E))$$

Sid
S1
S2
S3

Cid
C1
C2
C3

$\Rightarrow$

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S2	C3
S3	C1
S3	C2
S3	C3

$-$

Sid	Cid
S1	C1
S1	C2
S1	C3
S2	C1
S2	C2
S3	C3

$=$

Sid	Cid
S2	C3
S3	C1
S3	C2

$\Rightarrow$

Sid
S2
S3

Topic: Queries

#Q. (b) Sid’s enrolled for every course

Topic: Queries

#Q. (b) Sid’s enrolled for every course

Sol.: {Sid’s enrolled for every course }

Topic: Queries

#Q. (b) Sid's enrolled for every course

Sol.: {Sid's enrolled for every course }

= {Sid's enrolled for some course } – { Sid's not enrolled for every course }

Topic: Queries

#Q. (b) Sid's enrolled for every course

Sol.: {Sid's enrolled for every course }

= {Sid's enrolled for some course } – { Sid's not enrolled for every course }

= $\Pi_{Sid,\ Cid}\ (E) / \Pi_{Cid}\ (C)$

= $\Pi_{Sid}\ (E) - \Pi_{Sid}\ ((\Pi_{Sid}\ (E) \times \Pi_{Cid}\ (C)) - \Pi_{Sid,Cid}\ (E))$

Topic: Queries

#Q. (b) Sid's enrolled for every course

Sol.: {Sid's enrolled for every course }

= {Sid's enrolled for some course } – { Sid's not enrolled for every course }

= $\prod_{Sid, Cid}(E) / \prod_{Cid}(C)$

= $\prod_{Sid}(E) - \prod_{Sid}((\prod_{Sid}(E) \times \prod_{Cid}(C)) - \prod_{Sid,Cid}(E))$

= To expand division, $\prod$, X, – are used.

Topic: Queries

#Q. Generalizing the Question R(A, B. . .) S(B. . .) What is $\prod_{AB}(R)/\prod_{B}(S)$?

Topic: Queries

#Q. Generalizing the Question R(A, B. . .) S(B. . .) What is $\prod_{AB}(R)/\prod_{B}(S)$?

Sol.: $\prod_{AB}(R)/\prod_{B}(S)$

$\prod_{AB}(R)$		$\prod_{B}(S)$
a1	b1	b1
a1	b2	b2
a1	b3	b3
a2	b2	
a2	b3	

Topic: Queries

#Q. Generalizing the Question R(A, B. . .) S(B. . .) What is $\Pi_{AB}(R)/\Pi_B(S)$?

Sol.: $\Pi_{AB}(R)/\Pi_B(S) = \Pi_A(R) - (\Pi_A(\Pi_{A,}(R) \times \Pi_B(S) - \Pi_{A,B}(R))$

A value which is paired by a proper subset of B values of S.

$\Pi_{AB}(R)$		$\Pi_B(S)$
a1	b1	b1
a1	b2	b2
a1	b3	b3
a2	b2	
a2	b3	

Topic: Queries

#Q. For multiple attributes, $\prod_{ABCD}(R) / \prod_{CD}(S)$?

Topic: Queries

#Q. For multiple attributes, Π_{ABCD} (R)/Π_{CD} (S)?

Sol.: The output will be a table with AB attributes.

Π_{ABCD} (R)/Π_{CD} (S)

A	B	C	D
a1	b1	c1	d1
a1	b1	c1	d2
a2	b2	c1	d2

C	D
c1	d1
c1	d2

Topic: Queries

#Q. For multiple attributes, $\Pi_{ABCD}(R)/\Pi_{CD}(S)$?

Sol.: The output will be a table with AB attributes.

$$\Pi_{ABCD}(R)/\Pi_{CD}(S) = \Pi_{AB}(R) - \Pi_{AB}(\Pi_{AB}(R) \times \Pi_{CD}(S) - \Pi_{ABCD}(R)) =$$

A	B

a1	b1	c1	d1	c1	d1
a1	b1	c1	d2	c1	d2
a2	b2	c1	d2		

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(i) Retrieve Sids who enrolled for some course taught by KORTH.

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(i) Retrieve Sids who enrolled for some ($\bowtie_c$) course taught by KORTH.

Sol.: Enroll

Sid	Cid
S1	C1
S1	C2
S1	C3
S1	C5
S2	C2
S2	C5
S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(i) Retrieve Sids who enrolled for some ($\bowtie_c$) course taught by KORTH.

Sol.: Enroll

	Sid	Cid
✓	S1	C1
✓	S1	C2
✓	S1	C3
	S1	C5
✓	S2	C2
	S2	C5
	S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(i) Retrieve Sids who enrolled for some ($\bowtie_c$) course taught by KORTH.

Sol.: Enroll

	Sid	Cid
✓	S1	C1
✓	S1	C2
✓	S1	C3
	S1	C5
✓	S2	C2
	S2	C5
	S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

✓

$$\Pi_{Sid}(\sigma_{Enroll.Cid = Course.Cid}(Enroll \times \sigma_{Instr='KORTH'}(Course)))$$

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(i) Retrieve Sids who enrolled for some ($\bowtie_c$) course taught by KORTH.

Sol.: Enroll

	Sid	Cid
✓	S1	C1
✓	S1	C2
✓	S1	C3
	S1	C5
✓	S2	C2
	S2	C5
	S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

$$\prod_{Sid} (\sigma_{Enroll.Cid = Course.Cid} (Enroll \times \sigma_{Instr = 'KORTH'}(Course)))$$

$$\prod_{Sid} (Enroll \bowtie \sigma_{Instr = 'KORTH'}(cours)))$$

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(ii) Retrieve sid's who enrolled for every course taught by KORTH

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(ii) Retrieve sid's who enrolled for every course taught by KORTH

Sol.:

Enroll

Sid	Cid
S1	C1
S1	C2
S1	C3
S1	C5
S2	C2
S2	C5
S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(ii) Retrieve sid's who enrolled for every course taught by KORTH

Sol.:

Enroll

	Sid	Cid
	S1	C1
✓	S1	C2
	S1	C3
	S1	C5
	S2	C2
	S2	C5
	S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(ii) Retrieve sid's who enrolled for every course taught by KORTH

Sol.:

Enroll

	Sid	Cid
	S1	C1
✓	S1	C2
	S1	C3
	S1	C5
	S2	C2
	S2	C5
	S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

$$\Pi_{Sid}(Enroll) - \Pi_{Sid}(\Pi_{Sid}(Enroll) \times \Pi_{Cid}(\sigma_{Instr='KORTH'}(Course)) - \Pi_{Sid, Cid}(Enroll))$$

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(ii) Retrieve sid's who enrolled for every course taught by KORTH

Sol.:

Enroll

	Sid	Cid
✓	S1	C1
✓	S1	C2
✓	S1	C3
✓	S1	C5
	S2	C2
	S2	C5
	S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

$$\Pi_{Sid}(Enroll) - \Pi_{Sid}(\Pi_{Sid}(Enroll) \times \Pi_{Cid}(\sigma_{Instr='KORTH'}(Course)) - \Pi_{Sid, Cid}(Enroll))$$

Sid's pairs with every KORTH course

Topic: Queries

#Q. Enroll(Sid, Cid), Course(Cid, Instructor)

(ii) Retrieve sid's who enrolled for every course taught by KORTH

Sol.:

Enroll

	Sid	Cid
✓	S1	C1
✓	S1	C2
✓	S1	C3
✓	S1	C5
	S2	C2
	S2	C5
	S3	C4

Course

Cid	Instr
C1	KORTH
C2	KORTH
C3	KORTH
C4	Navathe
C5	Ullman

Sid
S1

$\Pi_{sidcid}(E) / \Pi_{cid}(\sigma(course)_{Instr = KORTH}$

Π_{Sid} (Enroll) − Π_{Sid} (Π_{Sid} (Enroll) × Π_{Cid} ($\sigma_{Instr= 'KORTH'}$ (Course)) − $\Pi_{Sid, Cid}$ (Enroll))

Sid's pairs with every KORTH course

Topic: Queries

#Q. Works For(Eid, Pid), Project(Pid, Pname)

(i) Retrieve Eids who work for some ($\bowtie_c$) DB Pname project.

(ii) Retrieve Eids who work for every DB project.

Topic: Queries

#Q. Works For(Eid, Pid), Project(Pid, Pname)

(i) Retrieve Eids who work for some ($\bowtie_c$) DB Pname project.

Sol.:

Works for

Eid	Pid
e1	p1
e1	p2
e1	P3
e2	p2
e2	p3
e3	p3

Project

Pid	Pname
p1	DB
P2	DB
p3	OS
p4	NW

Topic: Queries

#Q. Works For(Eid, Pid), Project(Pid, Pname)

(i) Retrieve Eids who work for some ($\bowtie_c$) DB Pname project.

Sol.:

Works for

	Eid	Pid
✓	e1	p1
✓	e1	p2
	e1	P3
✓	e2	p2
	e2	p3
	e3	p3

Project

Pid	Pname
p1	DB
P2	DB
p3	OS
p4	NW

Topic: Queries

#Q. Works For(Eid, Pid), Project(Pid, Pname)

(i) Retrieve Eids who work for some ($\bowtie_c$) DB Pname project.

Sol.:

Works for

	Eid	Pid
✓	e1	p1
✓	e1	p2
	e1	P3
✓	e2	p2
	e2	p3
	e3	p3

Project

Pid	Pname
p1	DB
P2	DB
p3	OS
p4	NW

Eid
e1
e2

$\Pi_{eid}(\text{works for} \bowtie \sigma_{Pname=DB}(\text{Project})$

Topic: Queries

Sol.: (ii) Retrieve Eids who works for every (÷) DB project.

Topic: Queries

Sol.: (ii) Retrieve Eids who works for every (÷) DB project.

eid	pid
e1	p1
e1	p2
e1	p3
e2	p2
e2	p3
e3	p3

pid
p1
p2

Topic: Queries

Sol.: (ii) Retrieve Eids who works for every (÷) DB project.

	eid	pid	pid
✓	e1	p1	p1
✓	e1	p2	p2
✓	e1	p3	
	e2	b2	
	e2	b3	
	e3	p3	

Topic: Queries

Sol.: (ii) Retrieve Eids who works for every (÷) DB project.

$$\Pi_{eid,\ pid}(WorksFor) / \Pi_{pid}(\sigma_{pname='DB'}(Project))$$

	eid	pid	pid
	e1	p1	p1
✓	e1	p2	p2
	e1	p3	
	e2	b2	
	e2	b3	
	e3	p3	

$= \Pi_{eid}(W) - \Pi_{eid}(\Pi_{eid}(W) \times \Pi_{pid}(\sigma_{pname=DB}(P)) - \Pi_{eid,pid}(W))$:

Every eid of works for relation pairs with all DB projects

eid's who not works for every DB projects

eids who works for some project.

= employee working for every DB project.

= Important point to remember

$\bowtie$: some

$\div$: every

Inspiring Stories : Biju Nair

Background: **Born in Kerala; faced poverty.**

Education: **Studied engineering on scholarships.**

Achievements: **Built low-cost housing projects for the poor.**

Impact: **Gave safe homes to thousands.**

Telegram channel

@RBRSIRPW

THANK - YOU