GATE

CRASH COURSE

DS & AI

Database Management System

Relational Algebra (Part-1)

By – Vishal Rawtiya Sir

Lecture No. 01

Topics *to be covered*

1. Introduction to relation
2. Query languages
3. Introduction to relational algebra
4. Basic relational algebra operation

Topic : Relational database

* In relational database information is organized in the form of table {Collection of rows and Columns}
* Dr. Codd defined 13 rules {from 0-12} for a table to be called a relation.
* In a relation duplicate records are not allowed.

Topic : Relational database

eg:

Student

Sid	Sname	Branch
S_1	A	CS
S_2	A	CS
S_3	B	IT
S_4	C	CS

Attributes/fields

In a relational table each row is called a record / tuple

- **Degree / Arity :-** No. of attributes (Columns) in a relation is called degree/arity of that relation
- **Record / Tuple :-** Each row of relational table is called record / tuple
- **Cardinality :-** No. of tuples (rows) in a relation is defined as cardinality of that relation
- **Relational Schema:** Relational schema provides the abstract detail of the relation. eg. $name\_of\_relation(Attr_1, Attr_2, Attr_3, \ldots Attr_k)$
  - $Attr_1$ → Name of 1st attribute
  - $Attr_2$ → Name of 2nd Attribute
  - ...
- **Relational Instance:** If the records (tuples) are present in the relation at a given time, then set of all those records is called relational instance of that time.
  - → Relational instance may change because of insert, delete & update opn

Topic : Relational database

Student

- Degree = 3
- Cardinality = 4
- Relational Schema :
  - ↳ Student (Sid, Sname, Branch)

Sid	Sname	Branch
S_1	A	CS
S_2	A	CS
S_3	B	IT
S_4	C	CS

Relational instance

$= \{ (S_1, A, CS), (S_4, C, CS), (S_2, A, CS), (S_3, B, IT) \}$

Topic : Query languages

Query languages

Procedural query language

- If we want to retrieve any data (record) from the database, then we need to define the procedure to retrieve that data from database

Eg: Relational Algebra

Non-procedural query language

We only need to know the "Syntax" provided by query language in order to access required information from database

Eg: SQL, Tuple Relational Calculus, Domain Relational Calculus.

Note:- { Query condition evaluates tuple by tuple, only One tuple at a time.

∴ If we want to compare two or more tuples of the same table or different tables, the we need to Join the tuples of those tables into single tuple

These are various Join opⁿ

Eg Cross Join, Theta join, Natural join etc

Note :- Relational algebra query will always produce distinct tuples.

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

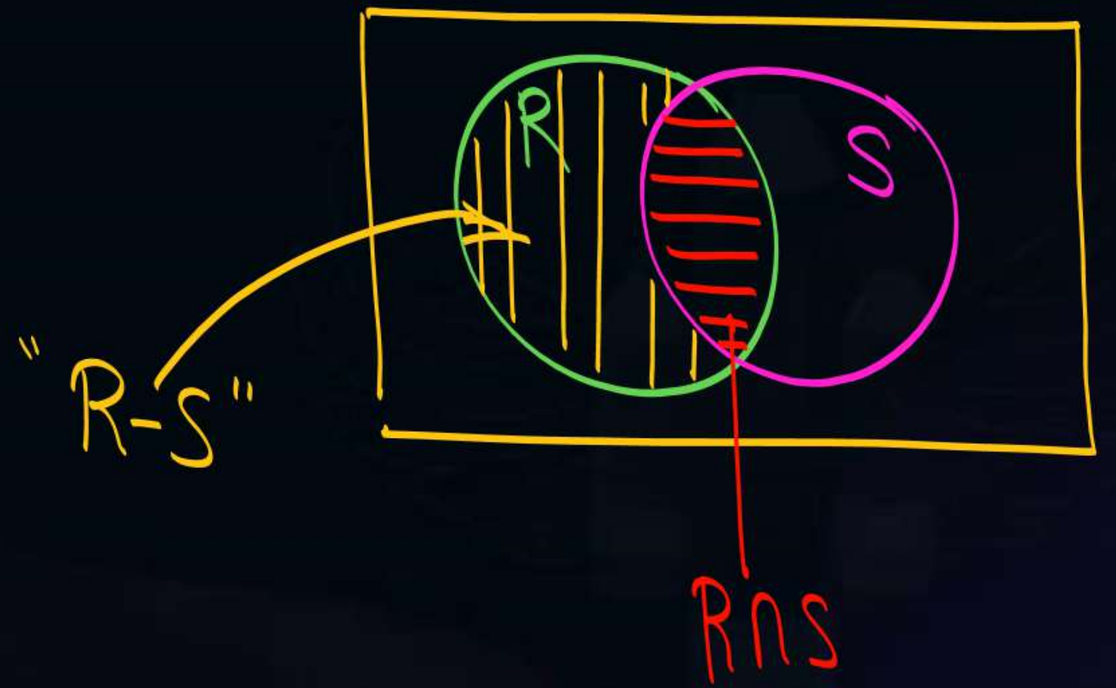
Topic : Basic Relational Algebra operators

1. Projection(π)
2. Selection (σ)
3. Cross Product (×)
4. Union (U)
5. Set Difference (−)
6. Rename (ρ)

Topic : Derived Relational Algebra operators

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

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



Topic : Projection (π)

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

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

Name of relation

Attributes/Columns required in O/P

Projection operator need not obey commutative property i.e.

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

R ≡ Relation_name

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

① $\pi_{Sid, Cid, Branch}(R) \equiv \pi_{\cdot}(R)$

list of attributes

Output of this relational algebra query will be Complete relation "R"

No projection oph is specified, then o/p of query will Contain all the attributes of relation "R"

② Retrieve Sids of all the Students from relation R

$\pi_{Sid}(R)$ ⇒ o/p =

Sid
S_1
S_2
S_3

③ Retrieve all Cids from the relation R.

$\pi_{Cid}(R)$ ⇒ o/p =

Cid
C_1
C_2
~~C_1~~

duplicate not allowed

=

Cid
C_1
C_2

R

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

(4) $\pi_{Sid,Cid}(R) \Rightarrow$ %p =

Sid	Cid
S_1	C_1
S_2	C_2
S_3	C_1

tuple wise they are different

$$\pi_{Sid}(\pi_{Sid,Cid}(R)) \not\equiv \pi_{Sid,Cid}(\pi_{Sid}(R))$$

Valid

invalid, because Cid is not present in internal schema

Sid
S_1
S_2
S_3

Topic : Selection(σ)

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

$$\sigma_{<selection\_condition>}(R)$$

Name of Relation

Condition to select tuples

NOTE:

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

OR

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

$$\Pi_{\text{list of attributes}}\left(\sigma_{\text{Selection Condition}}(\text{Relation})\right)$$

R

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

① $\pi_{Sid, Cid, Branch}(R)$

↳ we did not specify any selection Cond^n, therefore all tuples will be selected.

② Select the records Corresponding to Cid = "C_1"

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

i.e. all attributes are required in o/p

did not specify Projection opn

∴ In the o/p all attributes are present

o/p

Sid	Cid	Branch
S_1	C_1	CS
S_3	C_1	CS

R

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

③ Select Sids of the students who enrolled for Course with Cid = C_1

$$\pi_{Sid}\left(\sigma_{Cid='C_1'}(R)\right)$$

Sid	Cid	Branch
S_1	C_1	CS
S_3	C_1	CS

⇐

Sid
S_1
S_3

④ Retrieve records in which Cid = C_1 and Branch = 'IT'

R

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

∴ No Projection

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

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

o/p:

Sid	Cid	Branch

o/p will be Empty table

---

⑤ Retrieve records in which Cid = C_1 or Branch = 'IT'

⇒ No Projection

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

↑ OR

R

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

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

Topic : NOTE

A & B represent Condⁿ

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

OR

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

OR

$\sigma_A(R) \cap \sigma_B(R)$

(OR)

$\sigma_B(R) \cap \sigma_A(R)$

$\sigma_{A \vee B}(R) \equiv \sigma_{B \vee A}(R)$

(OR)

$\sigma_A(R) \cup \sigma_B(R)$

(OR)

$\sigma_B(R) \cup \sigma_A(R)$

Topic : Cross Product (×)

two operands (relation)

- 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.

i.e, each tuple of R Combined with each tuple of S.

'x' attributes

R

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

m tuples

'y' attributes

S

Sid	Sname
S_1	Ram
S_2	Mohan
S_3	Ram

n tuples

'x+y' attributes

$R \times S =$

R.Sid	R.Cid	R.Branch	S.Sid	S.Sname
S_1	C_1	CS	S_1	Ram
S_1	C_1	CS	S_2	Mohan
S_1	C_1	CS	S_3	Ram
S_2	C_2	IT	S_1	Ram
S_2	C_2	IT	S_2	Mohan
S_2	C_2	IT	S_3	Ram
S_3	C_1	CS	S_1	Ram
S_3	C_1	CS	S_2	Mohan
S_3	C_1	CS	S_3	Ram

(m*n) tuples

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. (No. of attributes)
2. Same domain of each corresponding attributes of relations

type of values that attribute can acquire.

R

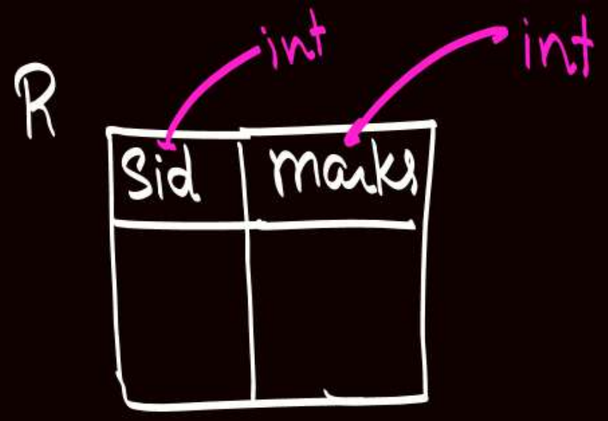
Sid	Cid	Branch

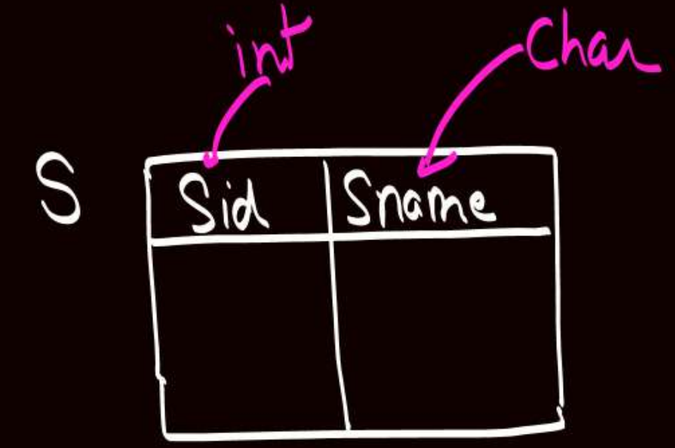
S

Sid	Sname

No. of attributes in R $\neq$ No. of attributes in S

$\therefore$ R & S are not Union Compatible





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

1st Condn is satisfied

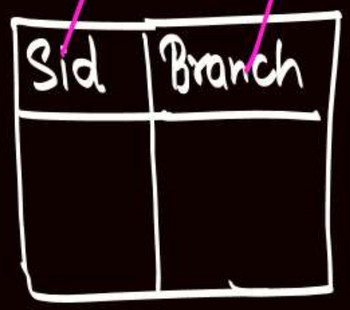
② Domain of 1st attribute of R = Domain of 1st attribute of S.

But, Domain of 2nd attribute of R ≠ Domain of 2nd attribute of S

↳ 2nd Condn dissatisfied

↳ Hence Not Union Compatible

R



S

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

② Domain of R.Sid = Domain of S.Sid

&

Domain of R.Branch = Domain of S.Sname

∴ R & S are Union Compatible.

Note: ① If relations are union Compatible, then only set operations can be performed on those relation.

② After the set operation, the resulting relation will take the names of its attributes from left hand side relation.

ie in "R∪S": names of attributes will be same as names of attributes in relation R.

and in "S∪R": names of attributes will be same as names of attributes in relation "S".

Topic : Union, Set difference, Intersection

- Union (A ∪ B)- It contains unique tuples from both the relations.

- Difference (A – B)- It contains all the tuples that are contained in the relation A but are not present in the relation B

- Intersection (A ∩ B)- It contains all the tuples that are contained in both the relations A as well as in relation B.

Employee (E)

Eid	Ename
1	A
3	B
5	A
8	C

Student (S)

Sid	Sname
2	D
3	B
5	A
7	C

E ∪ S =

Eid	Ename
1	A
3	B
5	A
8	C
2	D
7	C

S ∩ E =

S − (S − E) =

Sid	Sname
3	B
5	A

S − E =

Sid	Sname
2	D
7	C

E − S =

Eid	Ename
1	A
8	C

Topic : Rename (ρ)

Consider the following relational Schema | Student (Stud_id, Stu_name)

Rename operation can be used to rename attribute/s of the relation or name of the relation or both.

- Renaming a relation:

Rename opⁿ

$\rho_{FinalYrStudents}(Students)$

Current name of rel^n

↑ new name required for relation

{ Attribute names will remain same }

- Renaming attributes:

$\rho_{(SID, Sname)}(Students)$

Current name of rel^n

{ Relation name will remain same }

new name for 1st attribute

new name for 2nd attribute

- Renaming both:

$\rho_{FinalYrStudents\,(SID, Sname)}(Students)$

New name for rel^n

Name for 2nd attribute

Current name of rel^n

Name for 1st attribute

Topic : Rename (ρ)

Consider the following relational Schema | Students(Stud_id, Stu_name)

Rename operation can be used to rename attribute of the relation, name of the relation or both.

- Renaming a relation:

$\rho_{FinalYrStudents}(Students)$

o/p FinalYrStudents(Stud_id, Stu_name)

- Renaming attributes:

$\rho_{(SID,\ Sname)}(Students)$

o/p ⇒ Students(SID, Sname)

- Renaming both:

$\rho_{FinalYrStudents\ (SID,\ Sname)}(Students)$

o/p FinalYrStudents(SID, Sname)


Thank
THANK
Keep Hustling!