
Relational Algebra

It is a procedural query language. It is a step by step result of arriving at a definite result. It collects records of relations as input and gives occurrences of relations as output. Relational Algebra uses different operators to perform queries. The operator may be either unary or binary.

Let us assume we have three tables –

Customer

Cust_Id	Cust_Name	Address
1	Indu	Huda, Panipat
2	Piyush	Jaipur
3	Raj	Noida

Product1

Prod_ID	Prod_Name	Price
B3	Tshirt	505
A1	Shoes	1000
B1	Kurta	500
N3	Watch	2000

Product2

Prod_ID	Prod_Name	Price
B3	Tshirt	505
Z1	Denim	1500

Purchase

Purchase_ID	Cust_ID	Prod_ID
190	1	A1
102	3	N3

Selection (σ)

It is used to select some attributes or some tuples from the table.

Example

$\sigma_{\text{Cust_Name}="Indu"}(\text{Customer})$

It will select a customer whose name is Indu but it'll not display.

Note: This Select is DIFFERENT from MySQL SELECT.

Projection (π)

It is used to print / project attributes or some tuple from the table. Projection removes duplicate data.

Example

$\pi_{\text{Cust_Name}="Indu"}(\text{Customer})$

Output –

Cust_Id	Cust_Name	Address
1	Indu	Huda, Panipat

Union (U)

It performs union between two given relations. Suppose there are two tuples X and Y, the union operation would contain all the tuples that are either in X or Y or both in X and Y. It removes duplicate tuples.

Example

$\pi_{\text{Prod_name}}(\text{Product1}) \cup \pi_{\text{Prod_name}}(\text{Product2})$

Output –

Prod_Name
Tshirt
Shoes
Kurta
Watch
Denim

Intersection (\cap)

It performs an intersection between two given relations. Suppose you have two tuples X and Y, then X intersection Y will contain only those tuples which are common in both X and Y.

Example

$$\Pi_{\text{Prod_name}}(\text{Product1}) \cap \Pi_{\text{Prod_name}}(\text{Product2})$$

Output –

Prod_Name
Tshirt

Set Difference ($-$)

If you have two tuples X and Y then $X - Y$ contains all the tuples which are present in X but not in Y.

Example

$$\Pi_{\text{Prod_name}}(\text{Product1}) - \Pi_{\text{Prod_name}}(\text{Product2})$$

Output –

Prod_Name
Shoes
Kurta
Watch

Cartesian Product (\times)

The cartesian product is an operation used to combine each row in a given table with each row of another table. It is also known as cross product

Example

Customer X Purchase

Output –

Cust_Id	Cust_Name	Address	Purchase_ID	Cust_ID	Prod_ID
1	Indu	Huda, Panipat	190	1	A1
2	Piyush	Jaipur	190	1	A1
3	Raj	Noida	190	1	A1
1	Indu	Huda, Panipat	102	3	N3
2	Piyush	Jaipur	102	3	N3
3	Raj	Noida	102	3	N3

Rename (ρ)

It is used to rename the output relation. Let say you have a tuple named Add and you want to rename it to Address, then you can use Rename operator.

Example

ρ (Product2 , Out_Of_Stock)

This will rename the Product2 to Out_Of_Stock