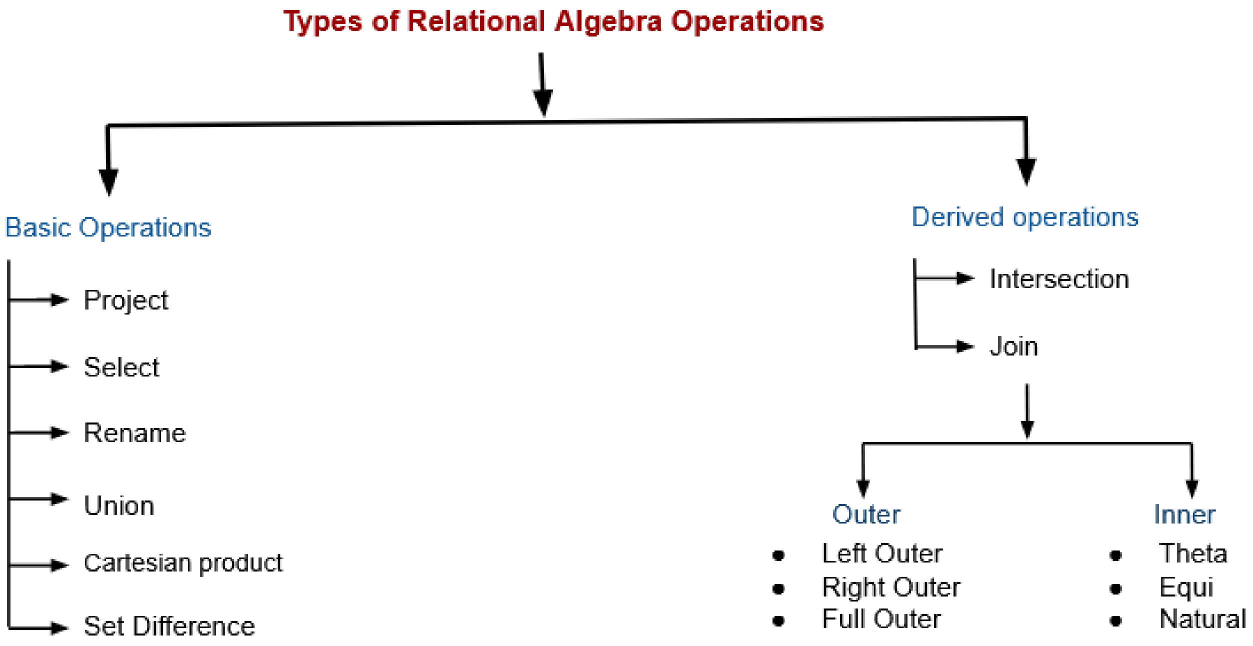
RELATIONAL ALGEBRA

Relational algebra is a procedural query language, which takes instances of relations as input and yields instances of relations as output. It gives a step by step process to obtain the result of the query. It uses operators to perform queries. They accept relations as their input and yield relations as their output.

The operation of relational algebra are as follows −



Types of Relational operation

1.Project Operation

- This operation only shows those attributes that we wish to appear in the result.
- Rest of the attributes are eliminated from the table.
- Duplicate rows are automatically eliminated, as relation is a set.

- It is denoted by ∏.

NOTATION: ∏ A1, A2, An (r)
Where: **A1**, **A2**, **A3** is used as an attribute name of relation **r**.

EX - Employee Relation

Emp_ID	Name	Department	City
101a	Rajesh	DBMS	Delhi
102a	Hari	HR	Dehradun
103a	Anuradha	Marketing	Delhi
104a	Shree	DBMS	Varanasi

Input: ∏ Name, City (Employee)

Output:

Name	City
Rajesh	Delhi
Hari	Dehradun
Anuradha	Delhi
Shree	Varanasi

2. Select Operation

- The select operation selects tuples that satisfy a given predicate.
- It is denoted by sigma (σ).

NOTATION: σ p(r)

Where:

σ, used for selection prediction

r, used for relation

P, used as a propositional logic formula which may use connectors like: AND OR and NOT. These relational can use as relational operators like =, ≠, ≥, <, >, ≤.

EX - Employee Relation

Emp_ID	Name	Department	City
101a	Rajesh	DBMS	Delhi
102a	Hari	HR	Dehradun
103a	Anuradha	Marketing	Delhi
104a	Shree	DBMS	Varanasi

Input: σ City="Delhi" (Employee)

Output:

Emp_ID	Name	Department	City
101a	Rajesh	DBMS	Delhi
103a	Anuradha	Marketing	Delhi

3. Rename Operation

- To rename the output relation.
- It is denoted by **rho** (ρ).

NOTATION: ρ_x (E)

Where: the result of expression **E** is saved with name of **x**.

EX - We can use the rename operator to rename Employee relation to Employee1.

4. Union Operation

- If there are two given relations A and B. The union operation contains all the tuples that are either in A or B or both in A & B.
- It eliminates the duplicate tuples. It is denoted by ∪.
- A and B must have the attribute of the same number.
- Duplicate tuples are eliminated automatically.

NOTATION: A ∪ B

Where: **A** and **B** are either database relations or relation result set (temporary relation).

EX - Employee Relation

Emp_ID	Name	Department	City
101a	Rajesh	DBMS	Delhi
102a	Hari	HR	Dehradun
103a	Anuradha	Marketing	Delhi
104a	Shree	DBMS	Varanasi

Salary Relation

Emp_ID	Name	Department	Salary
101a	Rajesh	DBMS	102500
102a	Hari	HR	92500
103a	Anuradha	Marketing	98700
104a	Shree	DBMS	102500
107b	Yash	HR	92500

Input: $\prod_{\text{Name}}$ (Employee) ∪ ∏ Name (Salary)

Output:

Name
Rajesh
Hari
Anuradha
Shree
Hari
Anuradha
Shree
Yash

5. Cartesian product

- It is used to combine each tuple in one table with each tuple in the other table, also known as a cross product.
- It is denoted by X.

NOTATION: A X B

Where: **A** and **B** are relations

EX - Employee Relation

Emp_ID	Name	Department
101a	Rajesh	DBMS
102a	Hari	HR
103a	Anuradha	Marketing

Department Relation

Dep_ID	Dep_name
11a	DBMS

22b	HR
33c	Marketing

Input: Employee X Department

Output:

Emp_ID	Name	Department	Dep_ID	Dep_name
101a	Rajesh	DBMS	11a	DBMS
101a	Rajesh	DBMS	22b	HR
101a	Rajesh	DBMS	33c	Marketing
102a	Hari	HR	11a	DBMS
102a	Hari	HR	22b	HR
102a	Hari	HR	33c	Marketing
103a	Anuradha	Marketing	11a	DBMS
103a	Anuradha	Marketing	22b	HR
103a	Anuradha	Marketing	33c	Marketing

6. Set Difference

- The result of set difference operation is tuples, which are present in one relation but are not in the second relation.
- It is denoted by intersection minus (-).

NOTATION: A - B

Where: **A** and **B** are relations

EX - Employee Relation

Emp_ID	Name	Department	City
101a	Rajesh	DBMS	Delhi

102a	Hari	HR	Dehradun
103a	Anuradha	Marketing	Delhi
104a	Shree	DBMS	Varanasi

Salary Relation

Emp_ID	Name	Department	Salary
101a	Rajesh	DBMS	102500
102a	Hari	HR	92500
103a	Anuradha	Marketing	98700
104a	Shree	DBMS	102500
107b	Yash	HR	92500

Input: ∏ Name (Employee) - ∏ Name (Salary)

Output:

Name
Yash

7. Set Intersection

- If there are two relations A and B. The set intersection operation contains all tuples that are in both A & B.
- It is denoted by intersection ∩.

NOTATION: A ∩ B

Where: **A** and **B** are relations

EX - Employee Relation

Emp_ID	Name	Department	City
101a	Rajesh	DBMS	Delhi
102a	Hari	HR	Dehradun
103a	Anuradha	Marketing	Delhi
104a	Shree	DBMS	Varanasi

Salary Relation

Emp_ID	Name	Department	Salary
101a	Rajesh	DBMS	102500
102a	Hari	HR	92500
103a	Anuradha	Marketing	98700
104a	Shree	DBMS	102500
107b	Yash	HR	92500

Input: ∏ Name (Employee) ∩ ∏ Name (Salary)

Output:

Name
Rajesh
Hari
Anuradha
Shree

The Join Operation

Inner Join

Join operation combines related rows from different relations,only when a given join condition is satisfied. It is denoted by ⋈.

Inner Join

1. Theta Join

- To merge Two different tables, having different attributes but one is common in both, based on the condition represented by theta.
- Theta can use any comparison operator.
- It is denoted by symbol **θ.**

EX - Math Table

St_id	M_Marks
101	91
102	88
103	90

Science Table

St_id	S_Marks
101	87
102	90
103	90

Input: Math $\bowtie_{\text{Math.M\_Marks > Science.S\_Marks}}$ (Science)

Output:

St_id	M_Marks	S_Marks
101	91	81

2. Equi Join

- It is the most common join.
- It is based on matched data present in two different tables as per the equality condition or we can say when a Theta join uses only the equivalence condition.
- The equi join uses the comparison operator(=).

EX - Math Table

St_id	M_Marks
101	91
102	88
103	90

Science Table

St_id	S_Marks
101	87
102	90
103	90

Input: Math $\bowtie_{\text{Math.M\_Marks = Science.S\_Marks}}$ (Science)

Output:

St_id	M_Marks	S_Marks
103	90	90

3. Natural Join

- A natural join is the set of tuples of all combinations in two different tables that are equal on their common attribute names.
- It is denoted by ⋈.

EX - Employee Relation

Emp_ID	Name
101a	Rajesh
102a	Hari
103a	Anuradha
104a	Shree

Salary Relation

Emp_ID	Salary
101a	102500
102a	92500
103a	98700
104a	102500

Input: ∏Name, Salary (Employee ⋈ Salary)

Output:

Name	Salary
Rajesh	102500
Hari	92500
Anuradha	98700
Shree	102500

Outer Join

1. Left Outer Join

- The outer join operation is an extension of the join operation. It is used to deal with missing information.
- Left outer join contains the set of tuples of all combinations in two different tables that are equal on their common attribute names.
- In the left outer join, tuples in one table have no matching tuples in another table.
- It is denoted by ⟕.

EX - Employee Relation

Emp_ID	Name
101a	Rajesh
102a	Hari
103a	Anuradha
104a	Shree

Salary Relation

Emp_ID	Salary
101a	102500
102a	92500
103a	98700

Input: (Employee ⟕ Salary)

Output:

Emp_ID	Name	Salary
101a	Rajesh	102500
102a	Hari	92500

103a	Anuradha	98700
104a	Shree	NULL

2. Right Outer Join

- Right outer join contains the set of tuples of all combinations in two tables that are equal on their common attribute names.
- In right outer join, tuples in one table have no matching tuples in second table.
- It is denoted by ⟖.

EX - Employee Relation

Emp_ID	Name
101a	Rajesh
102a	Hari
103a	Anuradha

Salary Relation

Emp_ID	Salary
101a	102500
102a	92500
103a	98700
104a	102500

Input: (Employee ⟖ Salary)

Output:

Emp_ID	Name	Salary
101a	Rajesh	102500
102a	Hari	92500

103a	Anuradha	98700
NULL	NULL	102500

3. Full Outer Join

- Full outer join is like a left or right join except that it contains all rows from both tables.
- In full outer join, tuples in one table that have no matching tuples in second table and tuples in first table that have no matching tuples in second table in their common attribute name.
- It is denoted by ⟗.

EX - Employee Relation

Emp_ID	Name
101a	Rajesh
103a	Anuradha
104a	Shree

Salary Relation

Emp_ID	Salary
101a	102500
102a	92500
103a	98700

Input: (Employee ⟗ Salary)

Output:

Emp_ID	Name	Salary
101a	Rajesh	102500

NULL	NULL	92500
103a	Anuradha	98700
104a	Shree	NULL

REFERENCES -

Database System Concepts (Abraham Silberschatz, Henry F. Korth, S. Sudarshan)

https://www.javatpoint.com/dbms-relational-algebra

https://www.javatpoint.com/dbms-join-operation