Relational Algebra :

Relational Algebra is procedural query language, which takes Relation as input and generate relation as output. Relational algebra mainly provides theoretical foundation for relational databases and SQL.

Operators in Relational Algebra

Projection (π)

Projection is used to project required column data from a relation.

Example :

```
    R
  (A  B  C)
  ----------
   1  2  4
   2  2  3
   3  2  3
   4  3  4
```

```
π (BC)
B  C
-----
2  4
2  3
3  4
```

Note: By Default projection removes duplicate data.

Selection (σ)

Selection is used to select required tuples of the relations.

for the above relation

σ (c>3)R will select the tuples which have c more than 3.

Note: selection operator only selects the required tuples but does not display them. For displaying, data projection operator is used.

For the above selected tuples, to display we need to use projection also.

```
π (σ (c>3)R ) will show following tuples.
```

```
A  B  C
-------
1  2  4
4  3  4
```

Union (U) :

Union operation in relational algebra is same as union operation in set theory, only constraint is for union of two relation both relation must have same set of Attributes.

Set Difference (-) :

Set Difference in relational algebra is same set difference operation as in set theory with the constraint that both relation should have same set of attributes.

Rename (ρ) :

Rename is a unary operation used for renaming attributes of a relation.
ρ (a/b)R will rename the attribute 'b' of relation by 'a'.

Cross Product (X) :

Cross product between two relations let say A and B, so cross product between A X B will results all the attributes of A followed by each attribute of B. Each record of A will pairs with every record of B.

below is the example

```
     A                                   B
    (Name    Age   Sex )                (Id    Course)
    ------------------                  -------------
    Ram     14    M                      1      DS
    Sona    15    F                      2      DBMS
    kim     20    M
```

```
     A X B
  Name    Age    Sex    Id    Course
---------------------------------
  Ram     14     M       1     DS
  Ram     14     M       2     DBMS
  Sona    15     F       1     DS
  Sona    15     F       2     DBMS
  Kim     20     M       1     DS
  Kim     20     M       2     DBMS
```

Note: if A has 'n' tuples and B has 'm' tuples then A X B will have 'n*m' tuples.

Natural Join (⋈) :

Natural join is a binary operator. Natural join between two or more relations will result set of all combination of tuples where they have equal common attribute.

Let us see below example

```
          Emp                                  Dep
   (Name    Id    Dept_name )          (Dept_name    Manager)
   ------------------------           ---------------------
     A      120     IT                    Sale       Y
     B      125     HR                    Prod       Z
     C      110     Sale                  IT         A
     D      111     IT
```

```
Emp ⋈ Dep

Name    Id    Dept_name    Manager
-------------------------------
A      120    IT           A
C      110    Sale         Y
D      111    IT           A
```

Conditional Join :

Conditional join works similar to natural join. In natural join, by default condition is equal between common attribute while in conditional join we can specify the any condition such as greater than, less than, not equal.

Let us see below example

```
         R                                  S
  (ID    Sex    Marks)              (ID    Sex    Marks)
  ------------------               --------------------
   1    F    45                      10    M    20
   2    F    55                      11    M    22
   3    F    60                      12    M    59
```

```
Join between R And S with condition R.marks >= S.marks

R.ID    R.Sex    R.Marks    S.ID    S.Sex    S.Marks
-----------------------------------------------
1        F        45         10      M         20
1        F        45         11      M         22
2        F        55         10      M         20
2        F        55         11      M         22
```

```
3       F       60      10      M       20
3       F       60      11      M       22
3       F       60      12      M       59
```

SQL

DDL:

DDL is the short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

- **CREATE – to create a database and its objects like (table, index, views, store procedure, function, and triggers)**
- **ALTER – alters the structure of the existing database**
- **DROP – delete objects from the database**
- **TRUNCATE – remove all records from a table, including all spaces allocated for the records are removed**
- **RENAME – rename an object**

DML:

DML is the short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

- **SELECT – retrieve data from a database**
- **INSERT – insert data into a table**
- **UPDATE – updates existing data within a table**
- **DELETE – Delete all records from a database table**
- **MERGE – UPSERT operation (insert or update)**

DCL:

DCL is the short name of Data Control Language which includes commands such as GRANT and is mostly concerned with rights, permissions, and other controls of the database system.

- **GRANT – allow users access privileges to the database**
- **REVOKE – withdraw users access privileges given by using the GRANT command**

TCL:

TCL is the short name of Transaction Control Language which deals with a transaction within a database.

- **COMMIT – commits a Transaction**
- **ROLLBACK – rollback a transaction in case of any error occurs**
- **SAVEPOINT – to roll back the transaction making points within groups**

Joins:

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

INNER JOIN:

The INNER JOIN keyword selects records that have matching values in both tables.

Syntax –

- **SELECT *column_name(s)* FROM *table1* INNER JOIN *table2* ON**

***table1.column_name* = *table2.column_name*;**

LEFT (OUTER) JOIN:

The LEFT JOIN keyword returns all records from the left table (table1), and the matching records from the right table (table2). The result is 0 records from the right side if there is no match.

Syntax –

- **SELECT *column_name(s)* FROM *table1* LEFT JOIN *table2***

ON *table1.column_name* = *table2.column_name*;

RIGHT (OUTER) JOIN:

The RIGHT JOIN keyword returns all records from the right table (table2), and the matching records from the left table (table1). The result is 0 records from the left side if there is no match.

Syntax –

- **SELECT *column_name(s)* FROM *table1* RIGHT JOIN *table2***

ON *table1.column_name* = *table2.column_name*;

FULL (OUTER) JOIN:

The FULL OUTER JOIN keyword returns all records when there is a match in the left (table1) or right (table2) table records.

Syntax:

- **SELECT *column_name(s)* FROM *table1* FULL OUTER JOIN *table2***

ON *table1.column_name* = *table2.column_name* WHERE *condition*;