SQL CLAUSES

SQL WHERE Clause:

SELECT column1, column2....columnN FROM table_EID WHERE CONDITION;

SQL LIKEClause:

SELECT column1, column2....columnN FROM table_EID WHERE column LIKE 'XXXX%‘

SELECT FROM table_EID WHERE column LIKE 'XXXX_'

There are two wildcards used in conjunction with the LIKE operator:

1. The percent sign (%)
2. The underscore (_)

SQL TOP Clause:

SELECT TOP number|percent column_EID(s) FROM table_EID WHERE [condition]

SQL UPDATE Statement:

```
UPDATE table_EID
SET column1 = value1, column2 = value2
    ....columnN=valueN
[ WHERE  CONDITION ];
```

SQL DELETE Statement:

- DELETE FROM table_EID WHERE {CONDITION};
- DELETE FROM table_EID
- DELETE table_EID

SQL ALTER TABLE Statement:

- ALTER TABLE table_EID
  ADD
  column_EID {data_type};

- ALTER TABLE table_EID
  DROP Column
  column_EID ;

- ALTER TABLE table_EID
  ALTER Column
  column_EID {data_type};

SQL DROP TABLE Statement:

- DROP TABLE table_EID;
- DROP DATABASE database_EID;

SQL TRUNCATE TABLE Statement :

TRUNCATE TABLE table_EID;

SQL COMMIT Statement:

```
COMMIT;
```

SQL ROLLBACK Statement :

```
ROLLBACK;
```

SQL OPERATORS

SQL Operators

- An operator is a reserved word or a character used primarily in an SQL statement's WHERE clause to perform operation(s), such as comparisons and arithmetic operations.
- Operators are used to specify conditions in an SQL statement

Type of Operators

- Arithmetic Operators (+, -, /, *, %)
- Comparison Operators (=, <>, !=, >, <, >= ,<=, !>, !<)
- Logical Operators (AND, OR, NOT)
- Other Operators (BETWEEN , IN, LIKE, IS NULL, DISTINCT, EXISTS

- Table creation
- Inserting data
- Verifying the data

ASSIGNMENT – 2

In the DEMO database create table EMP _SAL with the following fields:

- **EID DEPT DESI DOJ SALARY**
- **Insert 7 appropriate records in the EMP _SAL table**
- **Use SELECT command to view the contents of EMP _SAL table**

From the EMP table list all the employees with last name as Sharma.

Increase the salary of all Managers by 10%

NORMALIZATION

Normalization

Database normalization is the process of efficiently organizing data in a database. It is a set or rules/ guidelines / statements that we follow while storing the data.

There are two reasons of the normalization process:

- Eliminating redundant data, for example, storing the same data in more than one tables.
- Ensuring data dependencies make sense.

First Normal Form (1NF)

- Define the data items. This means looking at the data to be stored, organizing the data into columns, defining what type of data each column contains, and finally putting related columns into their own table.

- Ensure that there are no repeating groups of data

- Ensure that there is a primary key.

Second Normal Form (2NF)

- It should meet all the rules for 1NF
- There must be no partial dependences of any of the columns on the primary key

Third Normal Form (3NF)

- It should meet all the rules for 2NF
- Tables should have relationship.