

# SQL

## ❖ **Structured Query Language (SQL) :**

- it is used to store the data,manupulate the data and retrieve the data from the database as we required.

## ❖ **Data:**

- it is the raw fact or material which describes the property of object.

Property -> Attributes

Object -> Entity

## ❖ **DataBase :**

- It is the place or medium which is used to store the data in systematic and organized manner.

C - Create/Insert

R - Read/Retrive

E -Edit / Update

D – Delete /Drop

## ❖ **Database Management System(DBMS) :**

- It is the software which is used to manage and maintain the data.
- It have two important features 1. Security  
2.Authorization

In DBMS data store in file format

In DBMS we communicate with the database with help of quert language.

## ❖ **Relational Database Management System (R-DBMS) ;**

- It is the type of DBMS software which is used to manage and maintain the database
- It have same two features 1. Security

## 2. Authorization

In R-DBMS we store the data in table format

In R-DBMS we communicate with database with help of structured query language .

### ❖ **Table :**

➤ It is the combination of rows and column.

### ❖ **Row's (Touple) :**

➤ Horizontal portion of table is called rows.

### ❖ **Column(field) :**

➤ Vertical portion of table is called column.

### ❖ **Cell :**

➤ Intersection of row's and column is known as cell.

### ❖ **Relational Model :**

It is the theory pruposed by the data scientist E.F Codd.

Any DBMS which follows the relational model that is R-DBMS

In relational model we store the data in table format

### ❖ **Rules of E.F Codd :**

1.Data entered in a cell should accept single value data

2.In any R-DBMS data stored in table format and it includes metadata

3. According to E.F Codd we store the data in multiple tables if we needed we can establish the connection between the two tables with help of key attributes.

4.We can validaiating the data into two types by

✚ Assigning the **Datatypes**

✚ Assigning the **Constraints**

## ❖ DataTypes :

- It is used to store the what type of data to be stored in a particular memory allocation.
- It that we have five datatypes
  1. CHAR
  2. VARCHAR / VARCHAR2
  3. NUMBER
  4. DATE
  5. LARGE OBJECT
    - i. CHARACTER LARGE OBJECT
    - ii. BINARY LARGE OBJECT

### 1. CHAR :

- It will accept 'A-Z','a-z','0-9','\$#@','Ba7'.

**Syntax :**

**CHAR(Size);**

- It allocates fixed length memory allocation
- It will accept upto 2000 characters.
- Mo.no, PRN no, Adhar are the fixed memory allocation

### 2. VARCHAR :

- It will accept 'A-Z','a-z','0-9','\$#@','Ba7'

**Syntax :**

**VARCHAR(Size);**

- It allocates variable length memory allocation
- It will accept upto 2000 characters.

### ❖ VARCHAR2 :

**Syntax :**

**VARCHAR2(Size);**

It is updated version of VARCHAR

It will accept upto 2000 characters.

### 3. NUMBER :

- It accepts only integer value or numeric value
- It passes two arguments as follows

Precision : integer values

Scale : decimal values

**Syntax :**

**NUMBER(PRECISION,[SCALE]);**

- Precision accepts only integer value.
- Scale accepts only decimal value.
- By default value of scale is zero.

- NOTE : [ ] are optional

### 4. DATE :

- In Oracle we follow standard date format that will be 'DD-MON-YY' OR 'DD-MON-YYYY'

**Syntax :**

**'DD-mon-YYYY';**

### 5. LARGE OBJECT :

#### i. CHARACTER LARGE OBJECT :

- It will accept up to 4GB of characters

**Syntax :**

**CLOB(SIZE);**

#### ii. BINARY LARGE OBJECT:

- It will accept up to 4GB of digits.

**Syntax :**

**BLOB(SIZE);**

## ❖ CONSTRAINTS :

- It is used to validating the column.
- Five types of it's

**A. UNIQUE**

**B. NOT NULL**

**C. CHECK**

**D. PRIMARY KEY**

**E. FOREIGN KEY**

### **A. UNIQUE:**

- It will be used to avoid duplicate/repeated values in the table.

### **B. NOT NULL:**

- It will not accept null value.

### **C. CHECK:**

- It is used for extra validation purpose if all the conditions are satisfied we will get value else not.

### **D. PRIMARY KEY:**

- It is used to identify uniquely records from the table.

## ❖ Characteristics:

- In a table there should be only one primary key.
- It will not accept null value.
- It will not accept duplicate records.
- It is a combination of unique and not null constraint.

### **E. FOREIGN KEY :**

- It is used to establish the connection between two more tables.

## ❖ Characteristics:

- In F.K we have multiple in th table.
- It will accept null value.
- It will accepts duplicate records
- It is not combination of not null and unique constraints

❖ **Overview of SQL statements :**

1. **Data Defination language (DDL)**
2. **Data Manupultion Language (DML)**
3. **Transctional Control Language (TCL)**
4. **Data Control language (DCL)**
5. **Data Control Language(DCL)**

**1. Data Query language :**

- It is used to retrieve the data from database is called Data Query Langauage (DQL).

- a. **Select**
- b. **Selection**
- c. **Projection**
- d. **Joins**

**a. Select :**

- It is used to retrrive the data from the table and diplay it.

**b. Projection :**

- It used tto retrrive the data from the table by selecting only columns.

- **Syntax :**

**SELECT \*/[DISTINCT]**

**COLUMN\_NAME/EXPRESSION[ALIAS]**

- **ORDER OF EXECUTION :**

➔ **FROM**

➔ **SELECT**

\* -> All the details of the table

\* -> End of Query

### c. Selection :

- It is used to retrieve the data from the table by selecting only both row's and column.

- **Syntax :**

**SELECT \*/[DISTINCT]  
COLUMN\_NAME/EXPRESSION[ALIAS]**

- **Order Of Execution :**

- ➔ From
- ➔ Where <Row by Row>
- ➔ Select

- **Where Clause :**

- ➔ It is used to filter the condition.

- **Expression :**

- ➔ The Statement which gives the result is known as **Expression.**

### d. Joins :

- It is used to retrieve the data from multiple tables simultaneously.

#### ❖ **Oprand :**

- This is the values which are given by the users.

#### ❖ **Operators:**

- This is the symbol which performs some specific task.

#### ❖ **Types Of Operator :**

- There are seven types of operators.

**1) Arithmetic Operators : [ + , - , \* , / ]**

**2) Concatenation Operator : [ || ]**

**3) Comparison Operators : [ = , != ]**

**4) Relational Operators : [ < , > , <= , >= ]**

5) Logical Operators : [ AND , OR ,NOT ]

6) Special Operators : [ IN,NOT IN,BETWEEN,NOT BETWEEN,IS,IS NOT,LIKE,NOT LIKE]

7) SubQuery Operators : [ALL ,ANY ,EXIST ,NOT EXIST]

### 5) Logical Operators :

- **AND :**

➔ If all the conditions are satisfied then we use **AND** Operator.

- **OR :**

➔ If any one of the condition is satisfied we get the result that is **OR**

- **NOT :**

➔ It is logical universe operator.

### 6) Special Operators :

#### **IN :**

➤ it is multivalued operator which accepts multiple value at the R.H.S.

**Syntax :**

**COLUMN\_NAME/EXPRESSION IN (V1,V2,V3,.....,Vn);**

#### **NOT IN :**

➤ It is similar to IN operator instead of selecting value, value gets rejected.

**Syntax :**

**COLUMN\_NAME/EXPRESSION NOT IN (V1,V2,V3,.....,Vn);**

#### **BETWEEN :**

- It is used to select the range of value.
- If between is mentioned in the question then increment in initial value and decrement in final value.

**Syntax :**

**COLUMN\_NAME/EXPRESSION BETWEEN LOWER RANGE  
AND HIGHER RANGE**

#### **NOT BETWEEN :**

- It is similar to between operator instead of selecting value, value gets rejected.

**Syntax :**

**COLUMN\_NAME/EXPRESSION NOT BETWEEN LOWER  
RANGE AND HIGHER RANGE;**

#### **IS :**

- It is used to compare the null value.

**Syntax :**

**COLUMN\_NAME/EXPRESSION IS NOT NULL;**

#### **IS NOT :**

- It is used to compare not null value.

**Syntax :**

**COLUMN\_NAME/EXPRESSION IS NOT NULL;**

### **LIKE :**

- It is used to pattern matching purpose.

**Syntax :**

**COLUMN\_NAME/EXPRESSION LIKE ' PATTERN TO MATCH '**

### **NOT LIKE :**

- It is similar to like operator instead of selecting value,value get rejected.

**Syntax :**

**COLUMN\_NAME/EXPRESSION NOT LIKE ' PATTERN TO MATCH ';**

### **FUNCTION :**

- It is the block of code or set of instruction which is used to perform the some specific task.
- In function contains there are three main components.
  - ➔ Function name
  - ➔ Number of arguments
  - ➔ Return type

#### ➤ **Types of function :**

##### **1.Single Row Functon :**

- if we pass only one input we get only one optput.
- If passes multiplr input we get multiple outputs.

##### **2.Multi Row Function :**

- if we pass multiple inputs we get only one output.
- In multirow function we can not pass another column name.
- it have five types of multi row function

+ MAX( )

+ MIN( )

+ SUM( )

+ AVG( )

+ COUNT( )

+ Max ( ) :

- It is used to find maximum value in the column.

**Syntax :**

**SELECT MAX(COLUMN\_NAME/EXPRESSION)**

+ Min( ) :

- It is used to find minimum value in the column.

**Syntax :**

**SELECT MIN(COLUMN\_NAME/EXPRESSION)**

+ AVG( ) :

- It is used to find avg value in the column

**Syntax :**

**SELECT AVG(COLUMN\_NAME/EXPRESSION)**

+ SUM( )

- It is used summation of value present in the column.

**Syntax :**

**SELECT SUM(COLUMN\_NAME/EXPRESSION)**

+ COUNT ( ) :

- It is used to count number of value present in the column.

**Syntax :**

**SELECT COUNT(COLUMN\_NAME/EXPRESSION)**

#### ❖ GROUP BY CLAUSE :

- It is used to group the records.

Syntax:

```
SELECT GROUP BY EXPRESSION/GROUP FUNCTION  
FROM TABLE_NAME  
[WHERE < FILTER THE CONDITION>]  
GROUP BY GROUP COLUMN_NAME/EXPRESSION;
```

Order Of Execution :

```
FROM  
[ WHERE < ROW BY ROW >]  
GROUP BY < ROW BY ROW>  
SELECT <GROUP BY GROUP>
```

#### ❖ HAVING BY CLAUSE :

- It is used to filter the groups.

Syntax :

```
SELECT GROUP BY EXPRESSION/GROUP FUNCTION  
FROM TABLE_NAME  
[WHERE < FILTER THE CONDITION>]  
GROUP BY GROUP COLUMN_NAME/EXPRESSION;  
HAVING BY CLAUSE < GROUP BY GROUP>;
```

Order Of Execution :

```
FROM  
[ WHERE < ROW BY ROW >]  
GROUP BY < ROW BY ROW>  
HAVING BY < GROUP BY GROUP >  
SELECT <GROUP BY GROUP>
```

❖ **ORDER BY CLAUSE :**

- It is used to identify the given number in ascending or descending order.

Syntax ;

**SELECT COLUMN\_NAME**

**FROM TABLE\_NAME**

**ORDER BY CLOUMN\_NAME;**

For descending order

**ORDER BY COLUMN\_NAME\_DESC;**

Bangyan A.A