SQL[(Stractured query Language)]

SQL is a programing Language is used to communicate with and Manipulate the Data Base.

* It is also known as standard computer language.

**History of SQL :**

* **SQL** was developed in **1970’s** by the **IBM(International Business Machines)** researcher’s named **Raymond Boyce** And Donald Chamberlin.
* The **E.F. Codd** invented relational model for SQL called **System. R** model, Since he known as Co-Founder of **SQL** .
* Earlier SQL known as SEQUEL (Simple English Query Language).
* The **ANSI**(American National Standard Institute) has convert **SEQUEL** to **SQL** for Standardization of Language**.**

**Why/When we use SQL :**

1. By using SQL we can **Create/Insert** Records in Database.
2. By using SQL we can **Read/Retrieve/Fetch** data in Database.
3. By using SQL we can **Update /Modify** Records in Database.
4. By using SQL we can **Delete/Drop** data from Database.
5. By using SQL we can Create new Tables in Database.
6. By using SQL we can Execute queries against Database.
7. By using SQL we can Create View on Database.

**DATA :**

**Data** is a Row fact which describes the properties of an **Entity/Object .**

**DATABASE :**

**Database** is a Place/Media where we can store Data in **Systematic** and **Organized** manner.

Basically we can perform some operation in Database are :

* **Create/Insert**
* **Read/Retrieve**
* **Update /Modify**
* **Delete/Drop**

These Operations are usually known as “**CRUD”** Operation.

**Database Management System (DBMS) :**

* **DBMS** is a Software which is used to ***Maintain And Manage*** the Database.
* In DBMS we can store the Data in the **form of Files**.
* Here in DBMS **Security and Authorization** is the most important key Features.
* In DBMS we can use **Any Query Language** to communicate with the Database

**Relational Database Management System (RDBMS) :**

* **RDBMS** is a type of DBMS Software which is also used to ***Maintain And Manage*** the Database.
* In RDBMS we can store Data in the **form of** **Tables**.
* Here in RDBMS **Security and Authorization** is the most important key Features .
* In RDBMS we use only **SQL** to communicate with the Database.

|  |  |
| --- | --- |
| **Database Management System**  **(DBMS)** | **Relational Database Management System (RDBMS)** |
| 1. Founder of DBMS is Raymond Boyce. | 1. Founder of RDBMS is E. F. Codd. |
| 1. It doesn’t follows Relational model rule. | 1. It follows Relational model rule. |
| 1. It stores Less Amount of Data. | 1. It can store Huge Amount of Data. |
| 1. Here Only Single User Can Acess it. | 1. Here Multiple User can Access it. |
| 1. Normalization is Not Possible. | 1. Normalization Is Possible. |
| 1. It has Small Scale Industry. | 1. It has Medium-Large scale Industry. |
| 1. Here Data modification is Difficult. | 1. Here Data modification is Easier then DBMS. |
| 1. Here we store Data in the form of Files. | 1. Here we store Data in the form of Tables. |
| 1. It has Slow Execution Speed. | 1. It has Faster Execution Speed. |
| 1. Here we can’t Establish Connection between Multiple Files. | 1. Here we can Establish Connection between Multiple Tables. |

**Relational Model :-**

**Relational Model** was discovered by E.F. Codd, where we can store all the data in the form of **Rows and Columns.**

**E.F. Codd Rule**

DBMS

RDBMS

**RELATIONAL MODEL**

(**SYSTEM.R) Model**

* ***Any DBMS software which follows Relational Model or System. R model will Become RDBMS.***

**Tables:-**

***"It is a logical organization of data which consists of Rows & Columns.***

**Rules For Relational Model:-**

* **The** Data enteredto the cell must be **Single Valued** data.
* **According to E .F . Codd** we can store Data in multiple tables, If necessary are can establish connection between tables with the help of **Key Attributes.**
* In RDBMS we can store everything in the form of tables including **Meta Data**.
* The Data entered to the cell **must be Validated**.

Here Validation classified as Two types:

* By assigning **Data Type** (Mandatory).
* By assigning **Constraints** (Optional).

**Data Types :**

**Datatypes** are used to determine or specify the type or kind of Data we store in particular memory allocation.

Here Datatypes are classified in to 5 types as follows:

1. **Char *(Used to store Characters such as ‘A-Z’ , ‘a-z’ , ‘0-9’ and special characters like ‘$’, ‘#’, ‘%’, ‘&’ )***
2. **Varchar/varchar2 *(Used to store Characters such as ‘A-Z’ , ‘a-z’ , ‘0-9’ and special characters like ‘$’, ‘#’, ‘%’, ‘&’ )***
3. **Number *(Used to store Numeric numbers Only like Precision-Integer value (Mandatory) and Scale-Decimal(Optional))***
4. **Date *(Used to store Specific Date Format i.e. ‘01-JAN-1998’ or ‘01-JAN-98’)***
5. **Large object**

* **CLOB –** Character Large Object
* It is used to store characters up to 4 GB of size.
* **BLOB –** Binary Large Object
* It is used to store binary values of images, mp3, mp4 Documents et, …. Up to 4GB of size.

|  |  |
| --- | --- |
| **Char (Data Type)** | **Varchar/Varchar2(Data Type)** |
| 1. In Char Execution Speed is Faster. | 1. In Varchar Execution Speed is Slower. |
| 1. It can Store Max. 2000 Characters. | 1. It can Store Max. 4000 Characters in Varchar2. |
| 1. It Follows FLMA rule. | 1. It Follows VLMA rules. |
| 1. Default Value of Char is 1. | 1. Default value of Varchar is 0. |
| 1. Memory Wastes Occurs. | 1. Memory Wastes doesn’t Occurs. |
| 1. Syntax: char(size) | 1. Syntax: varchar(size) |

**Constraints ::**

***“It is a rule given to a column for validation.”***

Types of Constraints in SQL:

1. **Unique** *(It is used to avoid Duplicate/Repeated values.)*
2. **Not null** *(It is used to avoid Null Values.)*
3. **Check** *(It is an extra validation with a Column.)*

* *If the condition is satisfied then the value is accepted else Rejected* ".

1. **Primary Key (PK)**

* "*It is a constraint which is used to identify a record Uniquely from the table* ".

1. **Foreign Key (FK)**

* "*It is used to establish a connection between the The tables*".

|  |  |
| --- | --- |
| **Primary Key (PK)** | **Foreign Key (PK)** |
| 1. It is used to identify a records Uniquely from the table. | 1. It is used to establish a connection Between the tables |
| 1. It can be used Once in a Table. | 1. It can be used Multiple Times in a Table. |
| 1. It is the Combination of Unique & Not Null. | 1. It doesn’t combination of Unique & Not null. |
| 1. It doesn’t accept repeated values. | 1. It accepts repeated Values. |
| 1. It doesn’t accept Null. | 1. It accepts Null. |
| 1. It is not mandatory but highly recommended. | 1. It is mandatory for connecting multiple tables. |

**NOTE : NULL::**

Null Is a *keyword* which is used to represent Nothing / Empty Cell.

**Characteristics of Null:**

➢ Null doesn’t represent 0 or Space.

➢ Any operations performed on a Null will result in Null itself

➢ Null doesn’t Occupy any Memory and We cannot Equate Null.

**Statements ::**

1. Data Definition Language ( Ddl )

2. Data Manipulation Language ( Dml )

3. Transaction Control Language ( Tcl )

4. Data Control Language ( Dcl )

5. Data Query Language ( Dql )

**DATA QUERY LANGUAGE ( DQL \_) :**

***DQL is used to read/retrieve the data from the database*** "

**It had 4 statements:**

**1. SELECT :** *"It is used to retrieve the data from the table and display it.*

**2. PROJECTION:** *"It is a process of retrieving the data by selecting only*

*the columns is known as Projection "*

**3. SELECTION:** *"It is a process of retrieving the data by selecting both*

*the columns and rows is known as Selection ".*

**4. JOIN :***"It is a process of retrieving the data from Multiple tables simultaneously is known as Join ".*

**NOTE:**

➢ FROM Clause starts the execution.

➢ For FROM Clause we can pass Table Name as an argument.

➢The job of FROM Clause is to go to the Database and search for the table and put the table under execution.

➢ SELECT Clause will execute after the execution of FROM Clause

➢ For SELECT Clause we pass 3 arguments

 \*

 Column Name

 Expression

➢The job of SELECT Clause is to go the table under execution and select the columns mentioned.

➢ SELECT Clause is responsible for preparing the result table.

➢ Asterisk (\*): it means to select all the columns from the table.

➢ Semicolon: it means end of the query.

**DISTINCT Clause :**

***" It is used to remove the duplicate or repeated values from the Result table ".***

➢ Distinct clause has to be used As the first argument to select clause.

➢ We can use multiple columns As an argument to distinct clause, it will remove the combination of columns in which the records are duplicated.

**EXPRESSION**

***"A statement which gives result is known as Expression ".***

* Expression is a combination Operand and Operator.

**Operand:** These are the values that we pass.

**Operator:** These are the Symbols which perform some Operation on The Operand.

**WHERE Clause:**

***"Where clause is used to filter the records ".***

**Operators in SQL ::**

**In** SQL we have 7 types of Operator which is

1. Arithmetic Operator:: ( + , - , \* , /).
2. Concatenation Operator:: ( || )
3. Comparison Operator :: ( = , != , or <>)
4. Logical Operator :: ( AND , OR , NOT )
5. Relational Operator :: ( < , > , >= , <= )
6. Special Operator :: IN NOT IN

LIKE NOT LIKE

Between Not Between

IS IS NOT

1. Sub-Query Operator :: ( ALL , ANY , EXIST , NOT EXIST)

**FUNCTIONS ::**

*“****Are a block of code or list of instructions which are used to perform a specific task.”***

There are 3 main components of a function*.*

1. Function Name

2. Number of arguments ( no of inputs )

3. Return type

**Types of Functions in SQL :**

1. Single row functions /Server Functions

2. Multi row functions / aggregate / group Functions.

**Multi Row Functions::**

It takes all the inputs at one shot and then executes and provides A single output .

➢ If we pass 'n' number of inputs to a MRF( ) it returns '1' Output .

**List of MRF ( )**

1. MAX( ) : it is used to obtain the maximum value present in the Column
2. MIN ( ) : it is used to obtain the minimum value present in the column
3. SUM ( ) : it is used to obtain the summation of values present in the column
4. AVG ( ) : it is used to obtain the average of values present in the column
5. COUNT( ) : it is used to obtain the number of values present in the column

NOTE :

Multi row functions can accept only one argument, i.e. a Column Name or an Expression

➢MRF ( Column Name / Exp ) Along with a MRF( ) we are not supposed to use any other Column Name in the select clause .

➢ MRF( ) ignore the Null .

➢ We cannot use a MRF( ) in where clause .

**COUNT( ) :** It is the only MRF which can accept \* as an Argument .

**GROUPING :**

**Group by clause:**

Group by clause is used to *group the records* .

***Syntax:***

Select group\_by\_expression / group\_function

From table\_name

[where <filter\_condition>]

Group by column\_name/expression ;

***Order of execution:***

1-from

2-where (if used) [row-by-row]

3-group by [row-by-row]

4-select [group-by-group]

**NOTE ::**

➢ Group By clause is used to group the records .

➢ Group By clause executes row by row .

➢ After the execution of Group By clause we get Groups .

➢ Therefore any clause that executes after group by must execute Group By Group .

➢ The Column\_Name or expression used for grouping can be used In select clause .

➢ Group By clause can be used without using Where clause .

**HAVING Clause ::**

" Having Clause is used to Filter the Group "

***Syntax:***

Select group\_by\_expression / group\_function

From table\_name

[where <filter\_condition>]

Group by column\_name/expression

Having <group\_filter\_condition>

***Order of execution:***

1-from

2-where(if used) [row-by-row]

3-group by(if used) [row-by-row]

4-having (if used ) [group-by-group]

5-select [group-by-group]

Differentiate between Where and Having.

|  |  |
| --- | --- |
| **WHERE** | **HAVING** |
| ➢Where clause is used to Filter the records. | ➢Having clause is used to Filter the groups. |
| ➢Where clause executes row By row. | ➢Having clause executes Group by group. |
| ➢In Where Clause we can not Use MRF( ). | ➢ In Having clause we Can use MRF( ). |
| ➢Where clause executes before Group by clause. | ➢Having clause executes After group by clause . |

**SUB QUERY:**

***" A query written inside another query.”***

Working Principle::

* Let us Consider two Queries Outer Query and Inner Query.
* Inner Query executes first and produces an Output.
* The Output of Inner Query is given/fed as an Input to Outer Query.
* The Outer Query generates the result.
* Therefore we can state that, the Outer Query is completely depends upon Inner Query.