SQL

STRUCTURED QUERY LANGUAGE

CONTENTS

DDL – Data Definition Language

1. Create
2. Rename
3. Alter
4. Truncate
5. Drop

DML – Data Manipulation Language

1. Insert
2. Update
3. Delete

TCL – Transaction Control Language

1. Commit
2. Rollback
3. Savepoint

DCL – Data Control Language

1. Grant
2. Revoke

DQL – Data Query Language

1. Projection
   - Select
   - From
   - Distinct
   - Order-By

2. Selection

   - Where
   - group By
   - having

3. Joins

8. Software Installation

9. Operators in SQL and its Types

10. Functions in SQL

   Single Row Function
   Multi Row Function
   - Max ( )
   - Min ( )
   - Sum ( )
   - Avg ( )
   - Count ( )
   - Contact ( )

Joins

1. Cross Join/Cartesian join
2. Inner Join/Equi join
3. Outer join
   a) Left outer join
   b) Right outer join.
   c) c. Full outer join
4. Self join
5. Natural join

11. Introduction to Normalization and ER Diagrams

HISTORY OF SQL

1. The SQL query language was initially developed as **SEQUEL (Sequential English Query Executable Language)** was developed in the year **1970** by IBM Researchers **"Raymond Boyce"** and **"Donald Chamberlin".**

2. **"EF Codd"** he co-developed SQL and he is also called as "**Father of SQL".**

3. IBM developed **SQL** but currently it is owned by **"Oracle"** Company.

What is SQL?

SQL stands for **Structured Query Language.**

It is used to communicate with the database.

What can SQL do?

- SQL can create new tables in a database.
- SQL can create new databases.
- SQL can insert records in a database.
- SQL can update records in a database.
- SQL can delete records in a database.
- SQL can retrieve data from a database.

Who uses SQL?

- Database Developers
- Database Testers
- Database Administrators

Database Developers:

Use SQL to create tables in Data base and to enter values in them.

Database Testers:

Use SQL to perform Back-end Validation.

Back-End Validation: Data entered in the front end is stored in the back end or database of the application. So testers use SQL to check whether the data is properly stored or not.

Database Administrators:

Use SQL to maintain a successful environment in the Data base.

- Here successful environment means preventing any disasters or crashes even if they do how to back up and recover the lost data.

What is a Database?

It is a place or a medium where data will be stored in a systematic and organised manner.

What is a Schema?

A Schema is a collection of tables in a Database.

What is Table?

A Table is a collection of data of one kind.

The Table is a combination of Rows and Columns.

- **Rows** are also called as **records** or **tuples**.
- A **Rows** is used to represent all the properties of a single entity.
- **Columns** are also called as **attributes** or **fields** or **identifiers**.
- The **Columns** is used to represent single properties of all entities.
- The intersection of Rows and Columns is called **Cells**.

DBMS (Database management System)

DBMS is a software which is used to maintain and manage the database.

RDBMS (Relational Database Management System)

RDBMS is also a software which is used to maintain and manage the database.

Advantages of RDBMS:

The basic operations that can be performed on a Database using RDBMS are:

- Create the table.
- Retrieve or read the data from the table.
- Update or modify the data.
- Delete the data from the table.
- Creating or adding the data inside the table.

Difference between DBMS and RDBMS

DBMS	RDBMS
❖ DBMS can be accessed by single user.	❖ RDBMS can be accessed by multiple user.
❖ DBMS does not support normalization.	❖ RDBMS supports normalization.
❖ Better suited for small scale industries.	❖ Better suited for large scale industries.
❖ Data stored in different formats such as pdf, png, .jpeg.	❖ Data stored in table format.
❖ Since it is stored in different formats we cannot establish the relation between these files.	❖ It follows EF Codd principles and here we can establish the relation between different formats.

RULES OF EF CODD

Rule 1: One data must be stored in one cell.

Example:

EMP ID	ENAME	PH.NO.
01	Alisha	9874561230
02	Riyan	9632587412, 9658741236

❌ → Two data cannot be stored in a cell

Rule 2: If there are multiple data present, then they must be stored in different tables/columns and a relationship between them must be established.

Example:

EMP ID	ENAME	PH.NO.	ALT NO.
01	Alisha	9874561230	-
02	Riyan	9632587412	9658741236

Data Integrity:

The process of restriction of storing wrong data or invalid data inside the table in the database.

OR

The process of checking the correctness and accuracy of data inside the table in the database.

➢ The process of data integrity will avoid wrong invalid data.
➢ It can be achieved by assigning "**Data Type**" and "**Constraints**" to the column present in the table.

Data types:

- It is used to determine what type or kind of data will be stored in a particularly memory location.
- In a table each and every column should be mandatorily assigned with data type and only with one data type.

Types of Data type:

1. Character data type
2. Variable character data type
3. Number data type
4. Date data type
5. Large Object

   a) Character Large object
   b) Binary Large object

1. Character data type:

- ➢ The short form of character data type is **"Char"**.
- ➢ Character data type is used to store "A-Z", "a-z", "0-9" and special characters (@,!,#,%....).
- ➢ The default size of character data type is **"One"**.
- ➢ The maximum size of character data type is **2000** characters.
- ➢ Character data type follows **"Fixed length memory allocation"**.

Syntax : Char(Size)

Example : Char(8)

D	I	Y	A				
1	2	3	4	5	6	7	8

Used Memory   Unused Memory

(Memory Wastage)
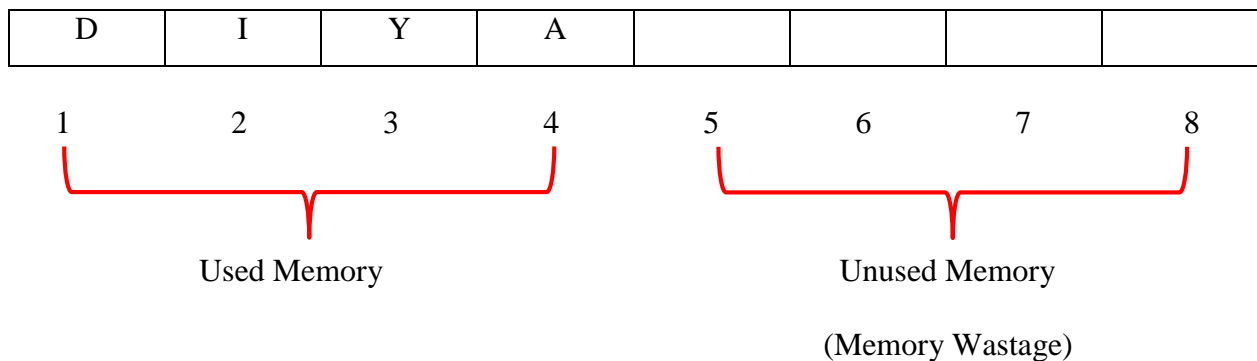
2. Variable character data type:

- ➢ The short form of variable character data type is **"Varchar"**.
- ➢ Variable character data type is used to store "A-Z", "a-z", "0-9" and special characters(@,!,#,%....).
- ➢ The maximum size of the variable character data type is **2000** characters.
- ➢ There is an updated version of varchar that is **varchar2** and its maximum size is **4000** characters.
- ➢ If users are more than 2000 characters then it will automatically update to **varchar2**.
- ➢ Unused memory spaces of Varchar data type are considered null and they can be utilized.

Syntax: Varchar(Size)

Example: Varchar(8)

D	I	Y	A				
1	2	3	4	5	6	7	8

Used Memory   Unused Memory

(Free Space)
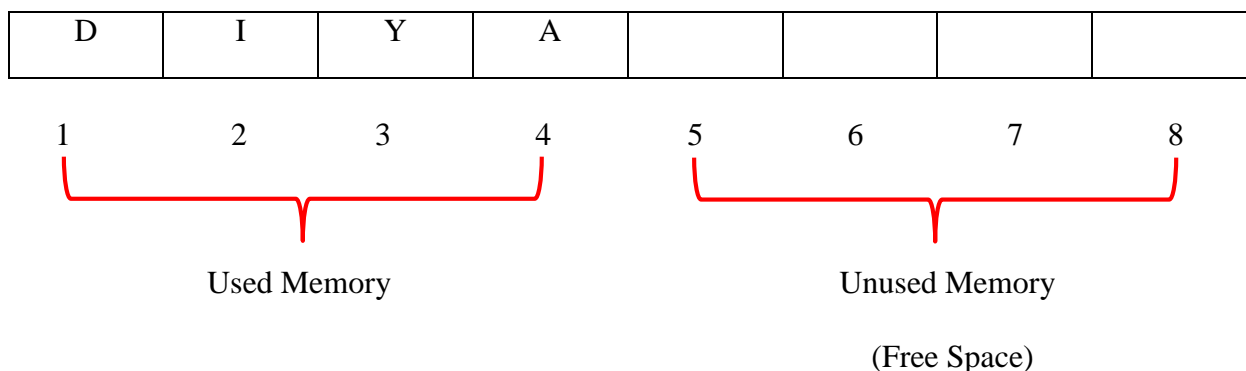
3. Date data type:

➢ The short form of date data type is **"Date"**.
➢ This Data type is used to store the Date in a particular format.
➢ If any column is declared as Date data type then that column will accept any date formats.

Syntax: Date

Example: MySQL supports only one date format **(YYYY-MM-DD)**

Date Of Joining, Hiredate, Joining Date

Date – **'1999-05-02'**

4. Number Data type:

➢ The short form of Number data type is **"Number"**.
➢ This data type is used to store numerical values.

OR

➢ If any column is assigned with number data type then that column will accept any number.

Syntax: Number(Precison, Scale)

Case1: Number(Precision)

Case 2: Number(Precision, Scale)

Only Precision	Number(5)	+/- 99999
Precision with Scale (P>S)	Number (5,2)	+/- 999.99
Precision with Scale (P=S)	Number (5,5)	+/-.99999
Precision with Scale (P>S)	Number (3,5)	+/-.00999

5. Large Objects Data type:

➢ Large objects are used to store huge amount of data.
➢ There are two types of large objects.

1. Character large objects.
2. Binary large objects

1. **Character large objects:** Character large objects are used to store characters up to 4GB in size.

Example: Terms and Condition, Documents, Wikipedia

2. **Binary large object:** Binary large objects are used to store binary values of images such as MP3, MP4 and documents up to 4GB in size.

Example: MP3 (Audio), MP4 (Video), Images (JPG, PNG, PDF)

Constraints:

Constraints are set of rules and regulations that is given to a column to restrict the wrong data being stored.

Types of Constraint:

1. Not Null
2. Unique
3. Check
4. Primary Key
5. Foreign Key

1. Not Null:

- **Null:** Null is nothing but an empty space.
- Null does not occupy memory.

➢ If a column is assigned with not null constraint then it will not accept null values i.e., it is mandatory to enter some data.
➢ If any column is assigned with not null it does not accept null values but it will still accept duplicate values.

2. Unique:

➢ If a column is assigned with a unique constraint then it will not accept duplicate values, but can accept null value.

3. Check:

➢ Check constraint is used to perform additional validation on column values.

Syntax: Column_Name Check(Condition)

Example: 1. Percentage check(Percentage>0)

2. Age Check(Age>18)

4. Primary Key:

➢ Primary key is a constraint given to a column which is used to identify or represent a record uniquely from the table.

Primary Key Eligibility:

➢ A column or attribute which is mandatory and it should consist of distinct values. That is combination of **not null** and **unique** constraints is eligible to become primary key.

Characteristics of Primary key:

✓ The primary key is a combination of **NOT NULL** and **UNIQUE** constraints.
✓ In a Table, more than one column can be eligible for the primary key, but only one column can be considered as the primary key.
✓ In a Table, primary key is not mandatory but it is highly recommended.
✓ If a Table does not have the primary key then that table is called a "**Weak entity".**

Candidate Key:

The columns which are all eligible to be Primary key is called **Candidate Key.**

Alternate Key:

The columns which are all eligible but not chosen to be Primary Key are called as **Alternate Key.**

5. Foreign Key:

➢ The foreign key is referential integrity constraint (if there is connection established between two tables then the parent table cannot be deleted).
➢ It is a column that belongs to another table.
➢ It is used to establish the relationship between tables.

Characteristics of Foreign Key:

✓ The table in which a foreign key is present is called the **"Child table".**
✓ The table to which actually foreign key belongs is called the **"Parent table".**
✓ In a table, we can have more than one column as Foreign Key.

✓ If a column is assigned as a Foreign key then it will accept Null and duplicate values, but it will not accept the values which are not present in the Parent table.

Primary Key and Foreign Key:

Primary Key   **Foreign Key**

Department Table   **Employee Table**

DeptNo	DeptName	Location
105	Testers	Bangalore
106	Developers	Mumbai
107	HR	Chennai

EmpNo	EmpName	DeptNo
001	Abhi	105
002	Sarah	106
003	Disha	107

Parent Table   **Child Table**