Introduction of Datatype in SQL :

* we must always specify the type of data that will be inserted in each column of the table

1. String Types :

* CHAR(n): fixed-length string, upto 255 characters
* VARCHAR(n): Variable-length string, upto 65,535 character
* TEXT: Variable-length text field.
  - ⇒ TINYTEXT: Up to ~~22~~ 255 characters
  - ⇒ TEXT: upto 65,535 charters,
  - ⇒ MEDIUMTEXT: upto 16,777,215 character
  - ⇒ LONGTEXT: upto 4,294,967,295 characters
* BLOB: Binary large object for storing binary data.
  - ⇒ TINYBLOB, BLOB, MEDIUMBLOB, LONGBLOB for different size
* ENUM : Enumerated list of values (eg. ENUM('small', 'medium', 'large'))
* SET : Similar to ENUM but allows multiple values from a list (e.g. SET('A', 'B', 'C'))

2: Numeric Types of Datatypes :

⇒ Integer

* INT: INteger, commonly used for whole Numbers.
  Range: -2,147,483,648 to 2,147,483,647 (4bytes)
* TINYINT: Very Small integer: Range: -128 to 127 (1 byte).

* SMALLINT: small integer Range: -32,768 to 32,767 (2 bytes)
* ~~BIGINT~~ * MEDIUMINT: medium Integer.
Range: -8,388,608 to 8,388,607 (3 bytes)
* BIGINT: large integer Range: -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 (8 bytes)
~~3~~ * Fixed point data represent exact values.
* ~~FLOAT:~~
* DECIMAL (M,D): Exact fixed-point decimal values, where M is the total number of digit and D is the number after the decimal.
* Float: Single-precision floating-point
* Double: Double-precision floating-point.
* BIT: Stores Binary values, useful for bitwise ~~ope~~ operator

4. Date and Time Types:
* Date: Stores data in YYYY-MM-DD format
* TIME: Stores time in HH:MM:SS format
* DATETIME: combines date and time in YYYY-MM-DD HH:MM:SS format
* TIMESTAMP: Similar to DateTime, but automatically updates with the current timestamp
* YEAR: Stores a year as a 4-digit format (e.g. 2024)

SQL Syntax

* Data Definition Language (DDL) :- DDL in SQL is a subset of SQL commands used to define and manage database structure, such as tables, index, schemas and views. DDL commands are responsible for setting up the structure of the database objects without manipulating the data itself. The main DDL commands are:

1. CREATE, 2- ALTER, 3- DROP, 4- TRUNCATE
5- RENAME

* Create command :- The CREATE statement in SQL is used to define and create a new database objects such as tables, database, views, indexes and more. Here are some common uses of the create Statement in SQL.

1. Creating a Database

Syntax :- ~~Cre~~ CREATE DATABASE database-name;

Example :- CREATE DATABASE CompanyDB;

(2) Creating a Table

Syntax :- CREATE TABLE table_name (
    column1 datatype constraints, } optional
    Column2 datatype constraints,
    .....
);

Example:- CREATE TABLE Employees(

```
ID INT Primary KEY,
Name VARCHAR(50) NOT NULL,
Age INT,
Salary DECIMAL(10,2)
);
```

3. Creating a view

Syntax:- CREATE VIEW view_name AS
SELECT columns From table
WHERE condition;

Example:- CREATE VIEW HighSalaryEmployees AS
SELECT Name, Salary FROM Employees
WHERE Salary > 50000;

4- Creating an Index

CREATE INDEX index_name ON table_name (column);

Example:- CREATE INDEX idx_name ON Employees (Name);

5. Creating a User-Defined Function (UDF)

Syntax:- CREATE FUNCTION function_name(Parameters)
RETURNS return_type
AS
BEGIN
... function logic
END;

Example:

```
CREATE FUNCTION GetFullName(@
FirstName VARCHAR(50), @LastName VARCHAR(50))
RETURNS VARCHAR(100)
AS
BEGIN
  RETURN @FirstName + ' ' + @LastName;
END;
```

(2) ALTER Command

The ALTER statement in SQL is used to modify the structure of an ~~existan~~ existing database Object, typically a table, with ALTER you can add, modify or drop columns and constraints, or even rename a table

Here are some common uses of the ALTER Statement

1 Adding a column

Syntax:

```
ALTER TABLE table_name
  ADD column_name datatype constraints;
```

Example:

```
ALTER TABLE Employees
ADD Department VARCHAR(50);
```

2. Modifying a column (Update datatypes in columns.)

```
ALTER TABLE table_name
MODIFY COLUMN column_name new_datatype
Contraints;
```

* Modifying Example:- (Syntax may vary depending on the SQL dialect):

```
ALTER TABLE Employees
MODIFY COLUMN Age INT NOT NULL;
```

3. DROPPING a column

Syntax :- ALTER TABLE table_name
DROP COLUMN column_name;

-Example:- ALTER TABLE @Employees
DROP COLUMN Age;

4 RENAMING a column

Syntax :- ALTER TABLE Table_name
RENAME COLUMN old_column_name TO new_column_name

Example:- ALTER TABLE Employees
RENAME COLUMN Salary TO Betan

5. Adding constraints :-

Syntax :- ALTER TABLE table_name
ADD CONSTRAINT constraint_name constraint_type (column_name);

Example :- ALTER TABLE Employees
ADD CONSTRAINT chk_salary CHECK (Salary>0);

6- DROPPING CONSTRAINTS

Syntax :- ALTER TABLE table-name
DROP CONSTRAINT constraint_name;

Exap :- ALTER TABLE Employees
DROP CONSTRAINT chk-salary;

7- RENAMING the TABLE

Syntax :- ALTER TABLE old-table-name
RENAME TO new_table_name;

Example :- ALTER TABLE Employeese
RENAME TO Staff;

The ALTER command is versatile and essential for evolving database structures without requiring complete table recreation.