

DATATYPES AND CONSTRAINTS(SQL)

1.WHAT are SQL Data Types?

Data types define what kind of data a column can store
(example: text, numbers, dates, decimals, boolean, binary files).

2. WHY do we use Data Types?

Data types are used to:

- Ensure correct data is stored
 - Save memory
 - Improve performance
 - Prevent invalid values
 - Allow database to understand how to process data
-

3. WHEN do we use Data Types?

You use data types whenever you create a table, column, or variable.

Example:

```
CREATE TABLE Student (  
    id INT,  
    name VARCHAR(50),  
    age INT,  
    dob DATE  
)
```

Here each column needs a data type.

SQL DATA TYPES :

1. Character / String Data Types

1) CHAR(n)

WHAT:

Fixed-length string. If you store fewer characters, remaining space is filled with spaces.

WHY:

Useful for storing *fixed-size* values.

WHEN USED:

PIN numbers, codes, gender ("M"/"F"), IFSC codes, etc.

EXAMPLE:

code CHAR(4)

2) VARCHAR(n)

WHAT:

Variable-length string. Stores only required space.

WHY:

Saves memory; flexible.

WHEN USED:

Names, city, address, email.

EXAMPLE:

name VARCHAR(50)

3) TEXT

WHAT:

Large text storage.

WHY:

Used for long descriptions.

WHEN USED:

Blog content, comments, product descriptions.

2. Numeric Data Types

1) INT

WHAT:

Whole numbers (without decimals).

WHY:

Used for counting values.

WHEN USED:

IDs, age, quantity.

2) BIGINT

WHAT:

Larger integer values.

WHEN USED:

Population, bank transaction IDs.

3) FLOAT

WHAT:

Approximate decimal value.

WHY:

Stores decimal values but not exact.

WHEN USED:

Scientific measurements.

4) DOUBLE

WHAT:

Higher precision than FLOAT.

WHEN USED:

Complex calculations.

5) DECIMAL(p,s)

WHAT:

Exact decimal value.

p = total digits, s = digits after decimal.

WHY:

Financial data must be exact.

WHEN USED:

Money, salary, price, GST, interest rate.

EXAMPLE:

price DECIMAL(10,2)

3. Date & Time Data Types

1) DATE

Stores date only (YYYY-MM-DD).

2) TIME

Stores time only.

3) DATETIME

Stores date + time.

4) TIMESTAMP

Auto-takes current time when row is inserted/updated.

4. Boolean

BOOLEAN

Stores TRUE or FALSE.

5. Binary Data Types

BLOB

Stores binary data like images, pdf, videos.

VARBINARY

Stores binary data of limited size

6. ENUM

WHAT:

Allows only one value from predefined list.

WHY:

Good for controlled inputs.

WHEN USED:

Status, gender.

EXAMPLE:

status ENUM('Active','Inactive','Pending')

7. SET

WHAT:

Stores multiple values from predefined list.

EXAMPLE:

A user having multiple hobbies.

WHAT are SQL Constraints?

Constraints are rules applied on columns to control the type of data inserted.

They help maintain:

- **Data accuracy**
 - **Data consistency**
 - **Data reliability**
-

1. WHY do we use Constraints?

To prevent incorrect data such as:

- **NULL values where not allowed**
- **Duplicate values**
- **Invalid foreign keys**
- **Wrong formats**

Constraints act like filters.

2. WHEN do we use Constraints?

While creating or altering tables.

Example:

id INT PRIMARY KEY

3. TYPES OF CONSTRAINTS:

1. NOT NULL

WHAT:

Prevents column from having NULL (empty) value.

WHY:

To ensure mandatory data is entered.

WHEN USED:

Name, phone number, price, DOB.

EXAMPLE:

`name VARCHAR(50) NOT NULL`

2. UNIQUE

WHAT:

Prevents duplicate values.

WHY:

Ensures each value remains different.

WHEN USED:

Email, phone number, username, roll number.

EXAMPLE:

`email VARCHAR(100) UNIQUE`

3. PRIMARY KEY

WHAT:

A column that uniquely identifies each row.

FEATURES:

- **UNIQUE + NOT NULL**
- **Only one primary key per table**

WHEN USED:

Student ID, product ID, employee ID.

EXAMPLE:

`id INT PRIMARY KEY`

4. FOREIGN KEY

WHAT:

Creates relationship between two tables.

WHY:

Maintains referential integrity (data linking).

WHEN USED:

Department ↔ Employee

Order ↔ Customer

EXAMPLE:

```
dept_id INT,
```

```
FOREIGN KEY (dept_id) REFERENCES Department(id)
```

5. CHECK

WHAT:

Sets a condition for value validation.

WHY:

To ensure logical correctness.

WHEN USED:

Age limit, salary range, quantity must be positive.

EXAMPLE:

```
age INT CHECK (age >= 18)
```

6. DEFAULT

WHAT:

Sets default value when no value is provided.

WHEN USED:

Status → Active

Country → India

EXAMPLE:

`status VARCHAR(10) DEFAULT 'Active'`

7. AUTO_INCREMENT

WHAT:

Automatically increments value for each new row.

WHEN USED:

Primary keys.

EXAMPLE:

`id INT PRIMARY KEY AUTO_INCREMENT`

8. INDEX

WHAT:

Increases speed of searching and retrieving data.

WHEN USED:

Search-heavy tables.

EXAMPLE:

`CREATE INDEX idx_name ON Student(name);`

9. COMPOSITE KEY

WHAT:

Using two or more columns as a primary key.

WHEN USED:

When single column is not enough to identify a row.

EXAMPLE:

`PRIMARY KEY (student_id, subject_id)`