Tasks Assigned Today

25-02-2025
Suneetha.V

MySQL Constraints:

1.NOT NULL

2.UNIQUE

3.PRIMARY KEY

4.FOREIGN KEY

5.CHECK

6.DEFAULT

7.CREATE INDEX

1. NOT NULL

- **Definition**: Ensures that a column cannot have a NULL value. The column must always have a valid (non-null) value.
- **Example**: `Name VARCHAR(50) NOT NULL;`

```
Syntax:

column_name data_type NOT NULL;

Example:

CREATE TABLE Employees ( EmployeeID INT NOT NULL, FirstName
VARCHAR(50) NOT NULL, LastName VARCHAR(50) NOT NULL );
```

Tasks Assigned Today

2. UNIQUE

- **Definition**: Ensures that all values in a column (or combination of columns) are unique. No two rows can have the same value in the specified column(s).
- **Example**: `Email VARCHAR(100) UNIQUE;`

```
Syntax:

column_name data_type UNIQUE;

Example:

CREATE TABLE Users ( UserID INT PRIMARY KEY, Email VARCHAR(100)
UNIQUE );
```

3. PRIMARY KEY

- **Definition**: A column or combination of columns that uniquely identifies each row in a table. It cannot have NULL values, and each value must be unique.
- **Example**: `EmployeeID INT PRIMARY KEY;`

```
Syntax:

column_name data_type PRIMARY KEY;

Example:

CREATE TABLE Employees ( EmployeeID INT PRIMARY KEY, FirstName
VARCHAR(50), LastName VARCHAR(50) );

CREATE TABLE Orders ( OrderID INT, ProductID INT, PRIMARY KEY
(OrderID, ProductID) );
```

Tasks Assigned Today

4. FOREIGN KEY

- **Definition**: A column in a table that refers to the **primary key** in another table. It creates a relationship between two tables and ensures referential integrity.
- **Example**: `FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID);`

```
Syntax:

FOREIGN KEY (column_name) REFERENCES parent_table(parent_column);

Example:

CREATE TABLE Orders ( OrderID INT PRIMARY KEY, OrderDate DATE,
EmployeeID INT, FOREIGN KEY (EmployeeID) REFERENCES
Employees(EmployeeID) );
```

5. CHECK

- **Definition**: Ensures that values in a column satisfy a specified condition. It is used to enforce data validity.
- **Example**: `Age INT CHECK (Age >= 18);`

```
Syntax:

column_name data_type CHECK (condition);

Example:

CREATE TABLE Employees ( EmployeeID INT PRIMARY KEY, FirstName
VARCHAR(50), LastName VARCHAR(50), Age INT CHECK (Age >= 18) );
```

Tasks Assigned Today

6. DEFAULT

- **Definition**: Provides a default value for a column when no value is specified during an insert.
- **Example**: `Status VARCHAR(20) DEFAULT 'Active';`

```
Syntax:

column_name data_type DEFAULT default_value;

Example:

CREATE TABLE Employees ( EmployeeID INT PRIMARY KEY, FirstName
VARCHAR(50), LastName VARCHAR(50), Status VARCHAR(20) DEFAULT
'Active' );
```

7. CREATE INDEX

- **Definition**: Creates an index on one or more columns to speed up search queries and improve performance.
- **Example**: `CREATE INDEX idx_lastname ON Employees(LastName);`

```
Syntax:

CREATE INDEX index_name ON table_name (column_name);

Example:

CREATE INDEX idx_lastname ON Employees(LastName);
```