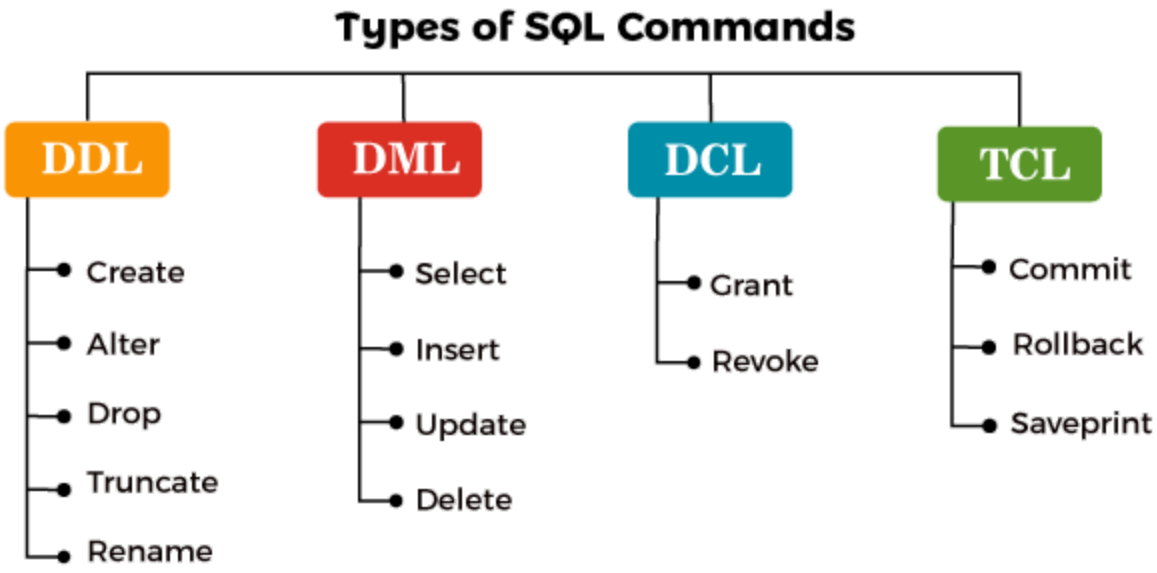
MySQL - 1 : DDL COMMANDS

Notes by Bhoopendra Vishwakarma

What is SQL ?

- ➔SQL (Structured Query Language) is a programming language used for managing and manipulating data in relational databases.
- ➔It allows you to insert, update,retrieve, and delete data in a database.
- ➔It is widely used for data management in many applications, websites, and businesses. In simple terms, SQL is used to communicate with and control databases.

Types of SQL commands



DDL commands for Databases :

1. CREATE
2. DROP

For example 👍

CREATE: Used to create a new database or table.

```sql
-- Create a database
CREATE DATABASE database_name;
```

DROP: Used to delete a database, table, or other objects.

```sql
-- Drop a database
DROP DATABASE database_name;
```

DDL commands for Tables :

1. CREATE
2. TRUNCATE
3. DROP

DDL commands specifically for tables in SQL allow you to create, modify, delete, or manipulate the structure of tables. Here are the common DDL commands for tables:

CREATE TABLE: To create a new table.

Copy code

```
CREATE TABLE table_name (
    column1 datatype constraint,
    column2 datatype constraint,
    column3 datatype constraint,
    ...
);
```

Example:

Copy code

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

1. **ALTER TABLE**: To modify an existing table (e.g., adding or modifying columns).

Add a new column:

Copy code

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

Example:

Copy code

```
ALTER TABLE Employees
ADD Email VARCHAR(100);
```

-

Modify an existing column:

Copy code

```
ALTER TABLE table_name
MODIFY column_name new_datatype constraint;
```

Example:

Copy code
```
ALTER TABLE Employees
MODIFY Email VARCHAR(150);
```

- **Drop a column**:

Copy code
```
ALTER TABLE table_name
DROP COLUMN column_name;
```
Example:

Copy code
```
ALTER TABLE Employees
DROP COLUMN Email;
```

- **DROP TABLE**: To delete an entire table along with its data.

Copy code
```
DROP TABLE table_name;
```
Example:

Copy code
```
DROP TABLE Employees;
```

2. **TRUNCATE TABLE**: To remove all rows from a table, but keep its structure.

Copy code

```
TRUNCATE TABLE table_name;
```

Example:

Copy code

```
TRUNCATE TABLE Employees;
```

RENAME TABLE: To rename an existing table.
sql
Copy code

```
RENAME TABLE old_table_name TO new_table_name;
```

3. Example:

   Copy code

   ```
   RENAME TABLE Employees TO Staff;
   ```

DATA INTEGRITY

- ➔ Data integrity in databases refers to the accuracy, completeness, and consistencyof the data stored in a database.
- ➔ It is a measure of the reliability and trustworthiness of the data and ensures that the data in a database is protected from errors, corruption, or unauthorized changes.

➔ There are various methods used to ensure data integrity, including:

Constraints:

➔ Constraints in databases are rules or conditions that must be met for data to be
➔ inserted, updated, or deleted in a database table. They are used to enforce the
➔ integrity of the data stored in a database and to prevent data from becoming

inconsistent or corrupted.

Transactions: a sequence of database operations that are treated as a single unit

of work.

Normalization: a design technique that minimizes data redundancy and ensures

data consistency by organizing data into separate tables.

CONSTRAINTS IN MYSQL

- Constraints in databases are rules or conditions that must be met for data to be inserted, updated, or deleted in a database table.

- They are used to enforce the integrity of the data stored in a database and to prevent data from becoming inconsistent or corrupted.

1. NOT NULL
2. UNIQUE(combo)

-> Another way of creating constraint

3. PRIMARY KEY
4. AUTO INCREMENT
5. CHECK
6. DEFAULT
7. FOREIGN KEY

Referential Actions

1. RESTRICT
2. CASCADE
3. SET NULL
4. SET DEFAULT

For example :

In MySQL, **constraints** are used to enforce rules on the data in a table, ensuring accuracy, integrity, and reliability. Here's a summary of the most commonly used constraints:

1. PRIMARY KEY

- Uniquely identifies each record in a table.
- Only one primary key is allowed per table, and it can consist of one or multiple columns (composite key).
- A primary key column cannot have `NULL` values.

Copy code

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

2. FOREIGN KEY

- Establishes a relationship between two tables by linking a column in one table to the primary key of another table.
- Ensures referential integrity by allowing only values that exist in the referenced table.

Copy code

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

3. UNIQUE

- Ensures that all values in a column (or a set of columns) are unique.
- Unlike the primary key, a table can have multiple `UNIQUE` constraints.
- `NULL` values are allowed in a `UNIQUE` column, but each `NULL` is considered distinct.

Copy code

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Email VARCHAR(100) UNIQUE
);
```

4. NOT NULL

- Ensures that a column cannot have NULL values, meaning the field must be filled.

Code here

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

5. DEFAULT

- Sets a default value for a column if no value is provided during the insert operation.

Code here

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    FirstName VARCHAR(50) NOT NULL,
    HireDate DATE DEFAULT CURRENT_DATE
);
```

6. CHECK (MySQL 8.0+)

- Ensures that all values in a column meet a specific condition.
- This constraint is supported in MySQL 8.0 or higher.

Code here

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Age INT CHECK (Age >= 18)
```

```
);
```

7. AUTO_INCREMENT

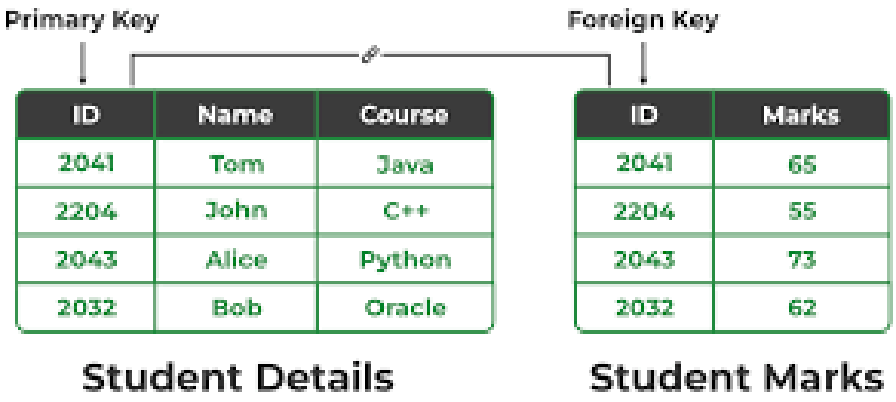
- Automatically generates a unique number when a new record is inserted.
- Typically used for the primary key column.

Code here

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

VISUALIZE THE FOREIGN KEY :

Establishes a relationship between two tables by linking a column in one table to the primary key of another table.



ALTER TABLE COMMAND:

The "ALTER TABLE" statement in SQL is used to modify the structure of an

existing table. Some of the things that can be done using the ALTER TABLE

statement include

1. Add columns
2. Delete columns
3. Modify columns

Here's how you can add, delete, and modify columns in MySQL using the ALTER TABLE statement:

1. Add Columns

To add a new column to an existing table, use the ADD clause.

Copy code

```
ALTER TABLE table_name

ADD column_name datatype [constraint];
```

Example: Add a DateOfBirth column to the Employees table.

Copy code

```
ALTER TABLE Employees

ADD DateOfBirth DATE;
```

You can also add multiple columns at once:

Copy code

```
ALTER TABLE Employees

ADD (Email VARCHAR(100), PhoneNumber VARCHAR(20));
```

2. Delete (Drop) Columns

To delete an existing column from a table, use the `DROP COLUMN` clause.

Copy code

```
ALTER TABLE table_name
DROP COLUMN column_name;
```

Example: Drop the `DateOfBirth` column from the `Employees` table.

Copy code

```
ALTER TABLE Employees
DROP COLUMN DateOfBirth;
```

You can also drop multiple columns by repeating the `DROP COLUMN` clause:

Copy code

```
ALTER TABLE Employees

DROP COLUMN Email, DROP COLUMN PhoneNumber;
```

3. Modify Columns

To modify the data type, constraints, or other attributes of an existing column, use the **MODIFY** or **CHANGE** clause.

MODIFY: To change the data type or constraints while keeping the same column name.

Copy code

```
ALTER TABLE table_name

MODIFY column_name new_datatype [constraint];
```

Example: Modify the **FirstName** column to increase the length to 100 characters.

Copy code

```
ALTER TABLE Employees

MODIFY FirstName VARCHAR(100);
```

CHANGE: To change both the column name and attributes (data type, constraints).

Copy code

```
ALTER TABLE table_name

CHANGE old_column_name new_column_name new_datatype
[constraint];
```

Example: Change the LastName column to Surname with a different data type.

Copy code

```
ALTER TABLE Employees
```

- CHANGE LastName Surname VARCHAR(75);