

MySQL | MODIFY | ALTER TABLE Statements

The **ALTER TABLE** statement is used to add, delete, or modify columns in an existing table.

The **ALTER TABLE** statement is also used to add and drop various constraints on an existing table.

1) ALTER TABLE - ADD Column

ALTER TABLE *table_name* **ADD** *column_name datatype*;

Ex:

ALTER TABLE Customer ADD Email VARCHAR(25);

Ex:

ALTER TABLE Customer ADD Description VARCHAR(50) FIRST;

Ex:

ALTER TABLE Customer ADD Location VARCHAR(50) AFTER Description;

2) ALTER TABLE - DROP Column

ALTER TABLE *table_name* **DROP COLUMN** *column_name*;

Ex:

ALTER TABLE Customer DROP Email;

3) ALTER TABLE - RENAME Column

ALTER TABLE *table_name* **RENAME COLUMN** *old_name* **TO** *new_name*;

Ex:

ALTER TABLE Customer RENAME COLUMN Email TO Gmail;

4) ALTER TABLE – ALTER/MODIFY DATA TYPE

ALTER TABLE *table_name* **MODIFY COLUMN** *column_name datatype*;

Ex:

ALTER TABLE Customer MODIFY COLUMN Email char(25);

5) ALTER TABLE – RENAME TABLE NAME

ALTER TABLE *table_name* **RENAME** *new_table_name*;

Ex:

ALTER TABLE Customer RENAME CustomerDetails;

SQL Constraints

SQL constraints are used to specify rules for the data in a table.

Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table.

- NOT NULL - Ensures that a column cannot have a NULL value.

Ex: `nic VARCHAR(15) NOT NULL,`

- UNIQUE - Ensures that all values in a column are different.

Ex: `nic VARCHAR(15) UNIQUE,`

- UNIQUE NOT NULL - Ensures that all values in a column are different and a column cannot have a NULL value.

Ex: `nic VARCHAR(15) UNIQUE NOT NULL,`

- PRIMARY KEY - A combination of a `NOT NULL` and `UNIQUE`. And uniquely identifies each row in a table.

•Primary keys must contain UNIQUE values, and cannot contain NULL values.

•A table can have **only ONE primary key**; and in the table, this primary key can consist of single or multiple columns (fields).

Ex: `nic VARCHAR(15) PRIMARY KEY,`

- CONSTRAINT PRIMARY KEY - Another method of define the PRIMARY key field.

Ex:

```
CREATE TABLE Customer(  
  CustomerID VARCHAR(6),  
  name VARCHAR(30),  
  address VARCHAR(30),  
  contact VARCHAR(10),  
  salary FLOAT(10,2),  
  CONSTRAINT PRIMARY KEY (CustomerID)  
);
```

- CONSTRAINT COMPOSITE PRIMARY KEY - Another method of define the PRIMARY key field.

Ex:

```
CREATE TABLE Customer(  
  CustomerID VARCHAR(6),  
  name VARCHAR(30),  
  address VARCHAR(30),  
  contact VARCHAR(10),  
  salary FLOAT(10,2),  
  CONSTRAINT PRIMARY KEY (CustomerID, name)  
);
```

•FOREIGN KEY – The FOREIGN KEY constraint is a key used to link two tables together.

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

Ex:

```
CREATE TABLE Orders(  
    OrderID VARCHAR(6),  
    date DATE,  
    CustomerID VARCHAR(6) NOT NULL,  
    CONSTRAINT PRIMARY KEY (OrderID),  
    CONSTRAINT FOREIGN KEY(CustomerID) REFERENCES Customer(CustomerID)  
);
```

•CHECK - Ensures that the values in a column satisfies a specific condition.

Ex: age INT CHECK (age<=18),

Ex: address VARCHAR(30) CHECK (address="Galle"),

•DEFAULT - Sets a default value for a column if no value is specified.

Ex: address VARCHAR(30) DEFAULT 'Galle',

=====

- ALTER TABLE Customer DROP PRIMARY KEY; //Remove Primary key
- ALTER TABLE Customer ADD PRIMARY KEY(CustomerID, Name); //Add Primary key
- ALTER TABLE Customer ADD CONSTRAINT UNIQUE (contact); //Add UNIQUE
- ALTER TABLE Customer DROP CONSTRAINT contact; //Remove UNIQUE
- ALTER TABLE Customer MODIFY address VARCHAR(30) NOT NULL;
//Add NOT NULL
- ALTER TABLE Customer MODIFY address VARCHAR(30) ; //Remove NOT NULL

=====

ENUM (Data Type)- Sets predefined values for a column.

Ex: gender ENUM ('MALE','FEMALE','OTHER'),

=====