# 1. ALTER COMMAND

**ALTER COMMAND IS USED TO:-**

- ADD the column to an existing table.
- DELETE a column from an existing column.
- Modify the definition of existing column.
- RENAME a column from an existing column.
- Change the definition of existing column.

# Syntax

**ADD Single column:-**

**Alter tabe <TableName>**

**Add Column <ColunmName> <datatype><Size>[Constraint];**

**ADD multiple columns:-**

**Alter tabe <TableName>**

**Add ( <ColunmName> <datatype><Size>, <ColunmName> <datatype><Size>);**

# ALTER Command (DROP Column)

- This is use to delete  a particular  column from an existing table.

## Syntax :

**Alter table <TableNmae>**
**Drop Column<ColumnName>**

# ALTER Command (Modify keyword)

- This is used to change the Datatype of the column from an existing table.

## Syntax :

```
ALTER TABLE Table_NAME
MODIFY column_name New Data Type;
```

- **ALTER Command (Change Table Name)**

  - This is used to change the Name of the column from an existing table.

## Syntax :

**ALTER TABLE old_table_name RENAME new_table_name;**

- **ALTER Command (Change Column Name)**

  - This is used to change the Name of the column from an existing table.

## Syntax :

**Alter table <TableNmae> rename column <Old Column_Name> to <Name Column_Name>;**

# Adding a constraint to an existing table.

## Syntax :

```
ALTER TABLE TABLE _NAME  ADD ( PRIMARY KEY
(COLUMN_NAME));
```

## 1. Customer Table

## 2. Order Table

| Column Name | Data Type | Description |
|---|---|---|
| CustomerID | INT | Unique identifier for customers. |
| CustomerName | VARCHAR(100) | Name of the customer. |
| Email | VARCHAR(100) | Email of the customer. |
| Country | VARCHAR(50) | Country of the customer. |

| Column Name | Data Type | Description |
|---|---|---|
| OrderID | INT | Unique identifier for orders. |
| OrderDate | DATE | Date of the order. |
| CustomerID | INT | Foreign key to Customer table. |
| OrderAmount | DECIMAL(10,2) | Amount of the order. |

# Applying Constraints

**1. NOT NULL**

Ensures columns cannot store NULL values.

```
ALTER TABLE Customer
MODIFY CustomerName VARCHAR(100) NOT NULL;

ALTER TABLE Order
MODIFY OrderDate DATE NOT NULL;

ALTER TABLE Employee
MODIFY Salary DECIMAL(10,2) NOT NULL;

ALTER TABLE Department
MODIFY DeptName VARCHAR(100) NOT NULL;
```

**2. Drop NOT NULL:**

```
ALTER TABLE table_name MODIFY
column_name data_type ;

ALTER TABLE Customer ALTER COLUMN
Country DROP DEFAULT;
```

## 2. UNIQUE

Ensures values in a column are unique

```
ALTER TABLE Customer
ADD CONSTRAINT UQ_Email UNIQUE (Email);

ALTER TABLE Department
ADD CONSTRAINT UQ_DeptName UNIQUE
(DeptName);
```

**Drop UNIQUE**:

```
ALTER TABLE table_name DROP INDEX constraint_name;
ALTER TABLE Customer DROP INDEX UQ_Email;
```

**3. CHECK**
Enforces specific conditions for values in a column.

```
ALTER TABLE Customer
ADD CONSTRAINT CHK_Country CHECK (Country IN ('USA', 'India', 'UK'));

ALTER TABLE Order
ADD CONSTRAINT CHK_OrderAmount CHECK (OrderAmount > 0);

ALTER TABLE Employee
ADD CONSTRAINT CHK_Salary CHECK (Salary >= 3000);
```

**Drop CHECK**:

```
ALTER TABLE table_name DROP CONSTRAINT constraint_name;

ALTER TABLE Customer DROP CONSTRAINT CHK_Country;
```

**4. DEFAULT**
Sets default values for a column.

```
ALTER TABLE Customer
MODIFY Country VARCHAR(50) DEFAULT 'USA';

ALTER TABLE Order
MODIFY OrderDate DATE DEFAULT CURRENT_DATE;

ALTER TABLE Employee
MODIFY Salary DECIMAL(10,2) DEFAULT 5000.00;
```

- **Drop DEFAULT**:

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

ALTER TABLE Customer ALTER COLUMN Country DROP DEFAULT;
```

## 5. PRIMARY KEY

Enforces a unique identifier for each record in the table.

```
ALTER TABLE Customer
ADD CONSTRAINT PK_Customer PRIMARY KEY
(CustomerID);

ALTER TABLE Order
ADD CONSTRAINT PK_Order PRIMARY KEY (OrderID);

ALTER TABLE Employee
ADD CONSTRAINT PK_Employee PRIMARY KEY
(EmployeeID);

ALTER TABLE Department
ADD CONSTRAINT PK_Department PRIMARY KEY
(DepartmentID);
```

## Drop PRIMARY KEY:

```
ALTER TABLE table_name DROP PRIMARY KEY;
ALTER TABLE Customer DROP PRIMARY KEY;
```

# Find the real constraint name

```
SELECT
    CONSTRAINT_NAME,
    TABLE_NAME,
    COLUMN_NAME,
    REFERENCED_TABLE_NAME
FROM
    information_schema.KEY_COLUMN_USAGE
WHERE
    TABLE_SCHEMA = 'your_database_name'
    AND TABLE_NAME = table_name'
    AND REFERENCED_TABLE_NAME IS NOT NULL;
```

## 6. FOREIGN KEY

Links one table to another.

```
ALTER TABLE Order
ADD CONSTRAINT FK_Order_Customer FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID);

ALTER TABLE Employee
ADD CONSTRAINT FK_Employee_Department FOREIGN KEY (DepartmentID) REFERENCES
Department(DepartmentID);
```

## Drop FOREIGN KEY:

```
ALTER TABLE table_name DROP FOREIGN KEY constraint_name;

ALTER TABLE Order   DROP FOREIGN KEY FK_Order_Customer;

ALTER TABLE Employee
DROP FOREIGN KEY FK_Employee_Department;
```

**Finding the Foreign Key Constraint Name**

If you don't know the exact name of the foreign key constraint, you can query the database to retrieve it:

```
SELECT CONSTRAINT_NAME
FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE
WHERE TABLE_NAME = 'table_name' AND CONSTRAINT_SCHEMA = 'database_name';
```

# ---The End---