Drop database if exists DDL\_Commands;

create DATABASE DDL\_Commands;

use DDL\_Commands;

CREATE TABLE my\_table(

id INT,

name VARCHAR(100) Not NULL,

age int NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

INSERT INTO my\_table (id, name, age) VALUES

(1, 'Rohitit', 25),

(2, 'Sachin', 35),

(3, 'Vikas', 28);

-- Create my\_table2

-- Create my\_table2

CREATE TABLE my\_table2 (

id INT,

description VARCHAR(255) NOT NULL

);

-- Insert data into my\_table2

INSERT INTO my\_table2 (id, description) VALUES

(1, 'Item 1'),

(2, 'Item 2'),

(3, 'Item 3');

desc my\_table;

-- Add a new column

ALTER TABLE my\_table ADD COLUMN email VARCHAR(100);

-- add column after any column

ALTER TABLE customers ADD COLUMN middle\_name VARCHAR(50) AFTER first\_name;

-- Adding a Column at first index

ALTER TABLE my\_table ADD COLUMN EMP\_Rank INT FIRST;

-- Modify an existing column

ALTER TABLE my\_table MODIFY COLUMN name VARCHAR(150) NOT NULL;

-- Change a column name and type

ALTER TABLE my\_table CHANGE COLUMN email email\_address VARCHAR(200);

-- Drop a column

ALTER TABLE my\_table DROP COLUMN age;

-- Add a primary key (if not already defined)

ALTER TABLE my\_table ADD PRIMARY KEY (id);

-- Add a foreign key

-- Assuming there is another table my\_table2 with a column id:

-- Add foreign key constraint to my\_table2

ALTER TABLE my\_table2 ADD CONSTRAINT fk\_id FOREIGN KEY (id) REFERENCES my\_table(id);

-- MUL indicates that the column is part of an index that allows for multiple occurrences of the same value

-- see all index (fk\_key and pr\_key and all)

-- we will discuss all indexs in indexing topic

SHOW INDEX FROM my\_table2;

SHOW INDEX FROM my\_table;

-- drop forign key

ALTER TABLE my\_table2 DROP FOREIGN KEY fk\_id; -- (it might show fk\_id but when you run it once again it will show does not exist)

-- Drop primary key

-- Important Notes:

-- You cannot drop the primary key if it is referenced by a foreign key constraint. In that case, you must first drop the foreign key.

-- Ensure that the column(s) you want to use as the new primary key have unique values, as primary keys must be unique.

ALTER TABLE my\_table DROP PRIMARY KEY;

-- RENAME TABLE

RENAME TABLE my\_table TO my\_new\_table;

-- Rename database

-- MySQL does not have a direct RENAME DATABASE command. However, you can effectively rename a database by following these steps:

-- CREATE DATABASE new\_database\_name;

-- RENAME TABLE old\_database\_name.table\_name TO new\_database\_name.table\_name; -- (need to do for all tables)

-- DROP DATABASE old\_database\_name;

-- Drop database

DROP DATABASE ddl\_commands;

-- ADD on

CREATE TABLE customers (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phone VARCHAR(15),

address VARCHAR(255),

city VARCHAR(50),

state VARCHAR(50),

zip\_code VARCHAR(10),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT NOT NULL,

order\_date DATE NOT NULL,

total\_amount DECIMAL(10, 2) NOT NULL CHECK (total\_amount > 0),

status VARCHAR(50) NOT NULL,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

INSERT INTO customers (first\_name, last\_name, email, phone, address, city, state, zip\_code)

VALUES

('Arjun', 'Sharma', 'arjun.sharma@example.com', '9876543210', '123 MG Road', 'Mumbai', 'Maharashtra', '400001'),

('Rohit', 'Verma', 'rohit.verma@example.com', '9876543211', '456 Park Street', 'Delhi', 'Delhi', '110001'),

('Priya', 'Singh', 'priya.singh@example.com', '9876543212', '789 Main Road', 'Bangalore', 'Karnataka', '560001'),

('Anjali', 'Rao', 'anjali.rao@example.com', '9876543213', '101 Lake View', 'Hyderabad', 'Telangana', '500001'),

('Vikram', 'Patel', 'vikram.patel@example.com', '9876543214', '202 High Street', 'Ahmedabad', 'Gujarat', '380001');

INSERT INTO orders (customer\_id, order\_date, total\_amount, status)

VALUES

(1, '2024-07-01', 1500.00, 'Pending'),

(2, '2024-07-02', 2500.00, 'Shipped'),

(3, '2024-07-03', 3500.00, 'Delivered'),

(4, '2024-07-04', 4500.00, 'Cancelled'),

(5, '2024-07-05', 5500.00, 'Pending');

ALTER TABLE orders DROP FOREIGN KEY orders\_ibfk\_1;

-- replace `orders\_ibfk\_1` with the actual constraint name

-- The orders\_ibfk\_1 in the statement ALTER TABLE orders DROP FOREIGN KEY orders\_ibfk\_1; represents the name of the foreign key constraint that you want to drop.

-- This name is automatically generated by MySQL when you create a foreign key constraint,

-- unless you explicitly specify a name for it.

-- Here's a breakdown of the components:

-- orders: The name of the table.

-- ibfk: Indicates that this is an InnoDB foreign key.

-- \_1: A sequential number assigned to the foreign key constraint, starting from 1.

-- These names are automatically generated unless you specify a name for the foreign key constraint when you create it.

-- If you need to drop a foreign key constraint and you haven't explicitly named it, you can use the SHOW CREATE TABLE statement to find the automatically generated name.

-- Find the foreign key constraint name:

SHOW CREATE TABLE orders;

-- The output might look like this:

-- CREATE TABLE `orders` (

-- `order\_id` int(11) NOT NULL AUTO\_INCREMENT,

-- `customer\_id` int(11) NOT NULL,

-- `order\_date` date NOT NULL,

-- `total\_amount` decimal(10,2) NOT NULL CHECK (total\_amount > 0),

-- `status` varchar(50) NOT NULL,

-- PRIMARY KEY (`order\_id`),

-- KEY `customer\_id` (`customer\_id`),

-- CONSTRAINT `orders\_ibfk\_1` FOREIGN KEY (`customer\_id`) REFERENCES `customers` (`customer\_id`)

-- ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- if you have not explictily mentiond the name of forigen key

-- You can Drop the foreign key constraint as below:

ALTER TABLE orders DROP FOREIGN KEY orders\_ibfk\_1;

-- Specifying a Foreign Key Constraint Name

-- When creating a foreign key, you can specify a custom name for the constraint to avoid using the default naming convention:

ALTER TABLE orders ADD CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id);

-- or you can do it while creating a table

CREATE TABLE customers (

customer\_id INT AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

phone VARCHAR(15),

address VARCHAR(255),

city VARCHAR(50),

state VARCHAR(50),

zip\_code VARCHAR(10),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE orders (

order\_id INT AUTO\_INCREMENT PRIMARY KEY,

customer\_id INT NOT NULL,

order\_date DATE NOT NULL,

total\_amount DECIMAL(10, 2) NOT NULL CHECK (total\_amount > 0),

status VARCHAR(50) NOT NULL,

CONSTRAINT fk\_customer FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

-- now you will need to drop as below

-- Drop the foreign key constraint

ALTER TABLE orders DROP FOREIGN KEY fk\_customer;