CREATING DATABASE & TABLES IN SQL

SQL Series Part 2

-Mayuri Dandekar

CREATING DATABASE

A database is a collection of related tables, queries and views etc

To create a database in MySQL, we use the **CREATE DATABASE** keywords. A keyword is a word that has a predefined meaning in SQL. In other words, if you want to create a database, you have to type CREATE DATABASE, you cannot be creative and type other words like MAKE DATABASE or CREATE COLLECTION etc.

Keywords are generally **not case sensitive in SQL**. Hence, you can also write create database or CREATE DATABASE.

Syntax-

CREATE DATABASE name_of_database;

VIEW DATABASE

We have to let the DBMS know that we want to **use this database**.

This is because the DBMS may be managing more than one databases concurrently.

We have to let it know that all subsequent code that we write applies to the stated database.

Syntax:

USE name_of_database;

DELETE/DROP DATABASE

If after you create your database, you realize that you have typed the name wrongly. There is no easy way to rename a database in MySQL.
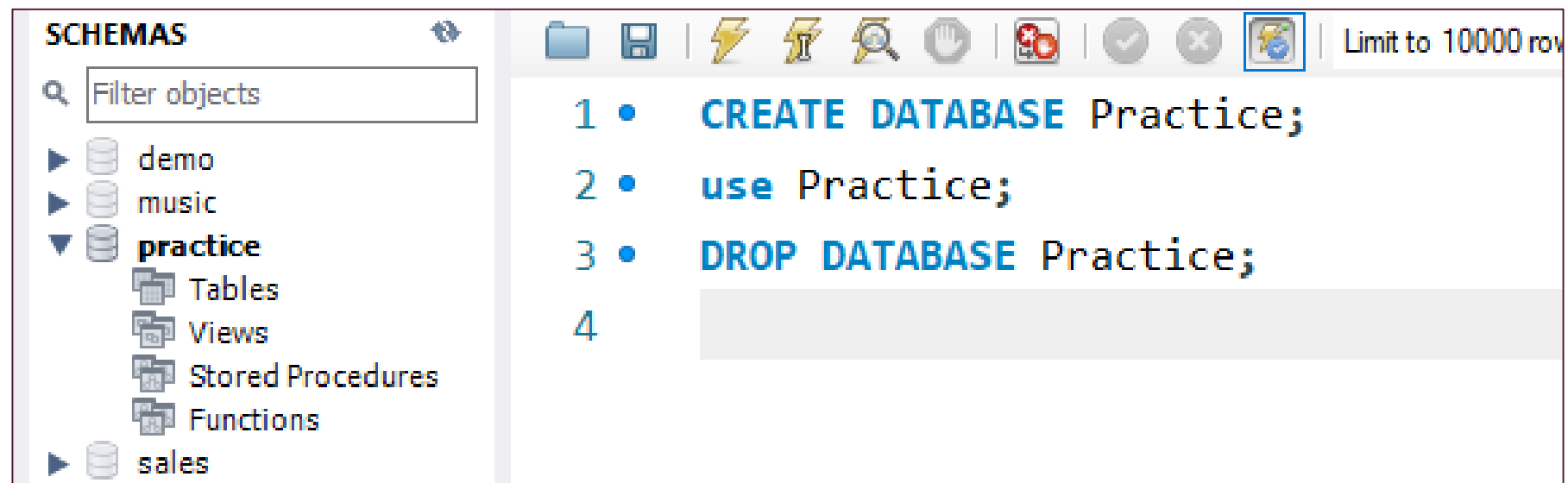
What you can do is create a new database and delete the old database.

Syntax-

DROP DATABASE [IF EXISTS] name_of_database;

When deleting a database, the **IF EXISTS keywords are optional**. We use them to prevent an error from occurring when we accidentally try to delete a database that does not exist.
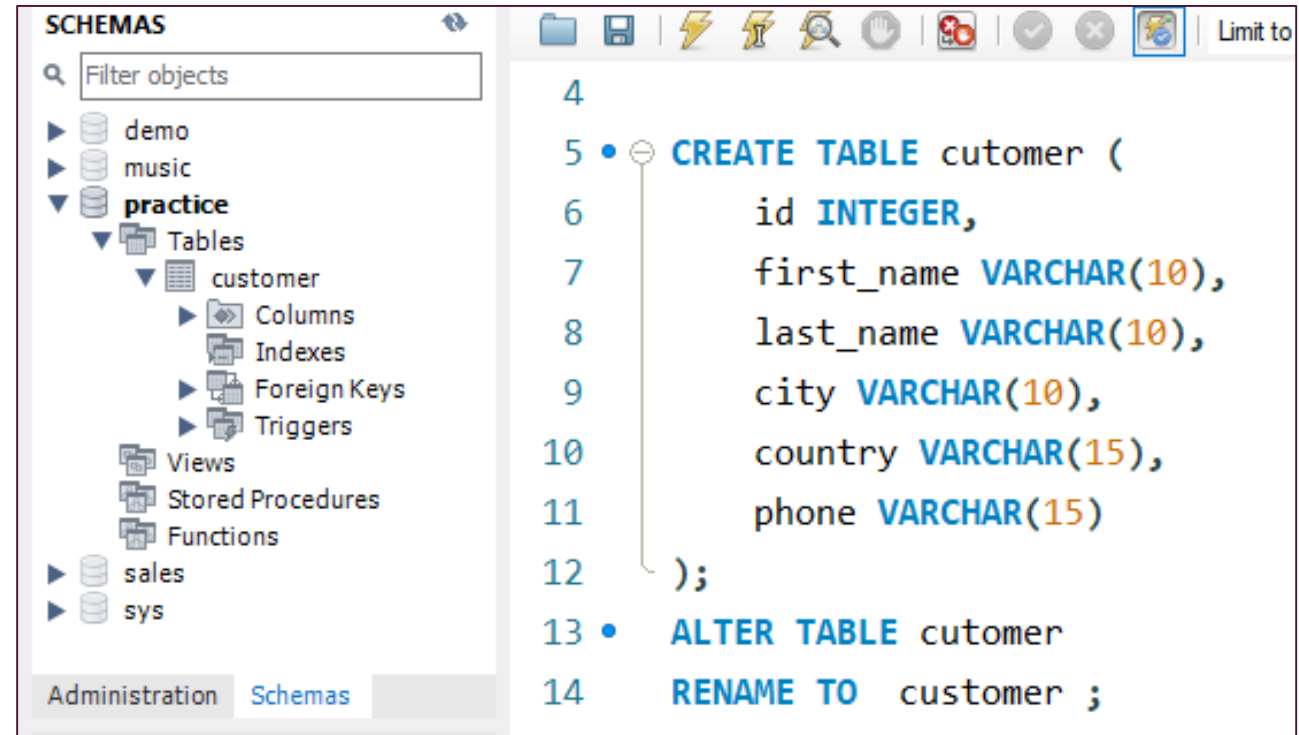
CREATE TABLES

The CREATE TABLE statement in SQL is used to **create a new table** in a database.

Syntax

CREATE TABLE table_name (

column1 data_type,

column2 data_type,

column3 data_type,

....

);

INSERT RECORDS IN TABLES

The INSERT INTO statement in SQL is used to **insert new records** in a table. Below are two ways of inserting records.

```
16      -- INSERT INTO --
17 •    INSERT INTO customer(id, first_name, last_name, city, country, phone)
18      VALUES (1, "sam", "xyz", "Mumbai", "India", 123456789);
19
20 •    INSERT INTO customer
21      VALUES (2, "pqr", "xyz", "Goa", "India", 123456789),
22      (3, "abc", "mno", "Kerala", "India", 0123456789);
23
24 •    SELECT * FROM customer;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	first_name	last_name	city	country	phone
1	sam	xyz	Mumbai	India	123456789
2	pqr	xyz	Goa	India	123456789
3	abc	mno	Kerala	India	123456789

UPDATE RECORDS IN TABLES

The UPDATE statement in SQL is used to **modify the existing records** in a table

```
26 •    SET SQL_SAFE_UPDATES=0;
27      -- update --
28 •    UPDATE customer
29      SET city = "pune"
30      WHERE id = 2;
31 •    SELECT * FROM customer;
32
33
```

| Result Grid | Filter Rows: | | Export: | Wrap Cell Content: |

	id ▲	first_name	last_name	city	country	phone	age
▶	1	sam	xyz	Mumbai	India	123456789	NULL
	2	pqr	xyz	pune	India	123456789	NULL
	3	abc	mno	Kerala	India	123456789	NULL

DELETE RECORDS IN TABLES

The DELETE statement is used to **delete existing records** in a table

```
33      -- delete --
34 •    DELETE FROM customer
35      WHERE id = 3;
36
37
```

	id	first_name	last_name	city	country	phone	age
▶	1	sam	xyz	Mumbai	India	123456789	NULL
	2	pqr	xyz	pune	India	123456789	NULL

Result Grid | Filter Rows: | Export: | Wrap Cell Conten

ALTER TABLES

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

```
30      -- alter --
31  •   ALTER TABLE customer
32      ADD COLUMN age int ; -- add column --
33  ⊖ /* ALTER TABLE customer
34    └ DROP COLUMN age; */  -- delete column --
35
```

id	first_name	last_name	city	country	phone	age
1	sam	xyz	Mumbai	India	123456789	NULL
2	pqr	xyz	Goa	India	123456789	NULL
3	abc	mno	Kerala	India	123456789	NULL

TRUNCATE TABLES

The TRUNCATE TABLE command **deletes the data inside a table, but not the table** itself

DROP TABLES

The DROP TABLE command **deletes a table** in the database

THANK YOU!!!

SQL Series Part 2

-Mayuri Dandekar