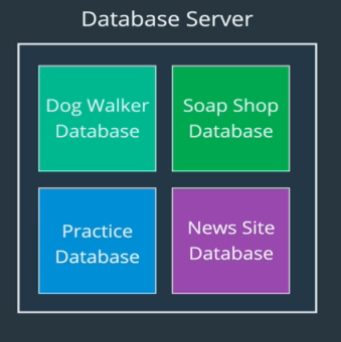
**What is Database?**

A database is **an organized collection of structured information, or data, typically stored electronically in a computer system**. A database is usually controlled by a database management system (DBMS).

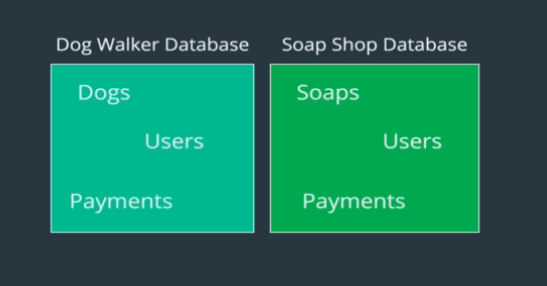
Difference between SQL and MySQL

|  |  |  |
| --- | --- | --- |
| No | SQL | MySQL |
| 1. | SQL is a Language we use to “talk” to our Database. Like CRUD operation. | MySQL is Database Management System that uses SQL to interact with MySQL databases. |
| 2. | Once you learn SQL, it's really not that hard to switch to another database that also uses SQL. | what makes them unique are the features that they offer when compared with other DBMS (SQLite or Oracle). |

So, in this Doc, we will be using a MySQL server. For example, let’s consider we are creating a database in MySQL server.



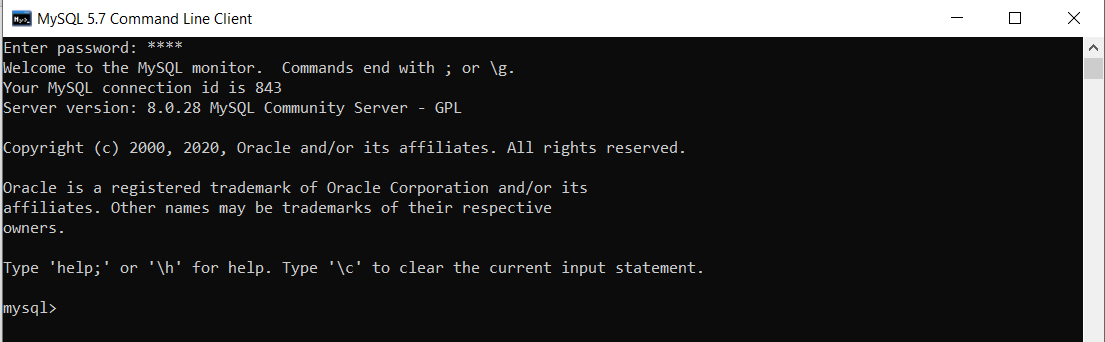
In the above image, we have a database for a dog walking app, and another one for a soap shop. We could have a practice database if you were just practicing some new code, some new lines of SQL.



Let’s have a closer look at the Dog walker Database and Soap Shop Database. So, inside those two databases, we'll store various entities or different data. So, the dog walking database will probably need to store information about dogs. And in our soap shop database, we'll need to store something about soaps. And then here's where it gets really important. Our dog walking app likely has users, but so does our soap shop app.

So, they have to be separated. If we just had one database that everything was using on this server, there would be a lot of crossovers. There would be issues. What if you had somebody who had the same name, who used your dog walking app and they also use the soap shop app and they kind of the streams got crossed, that's problematic.

Once the MySQL server is started, we can run the MySQL command Line Tool to run SQL commands as shown below.



Command used to show list of all database present in MySQL server is

mysql> SHOW DATABASES;



How to create a database?

CREATE DATABASE <name>; Best Practice to use is CREATE DATABASE database\_name;

Command used to create a dog Walker app and soap shop is

CREATE DATABASE DogApp; CREATE DATABASE soap\_shoap;

**To drop a database:**

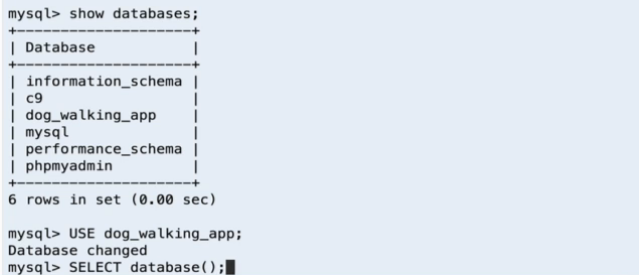
DROP DATABASE database\_name;

**For Example:**

DROP DATABASE hello\_world\_db;

Remember to be careful with this command! Once you drop a database, it's gone!

How to use the Database?

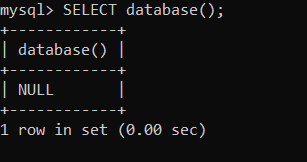




USE <database name>;

USE dog\_walking\_app;

The command to know which database is selected 🡪 SELECT database();



Returns NULL if no Database is selected.

**Tables**

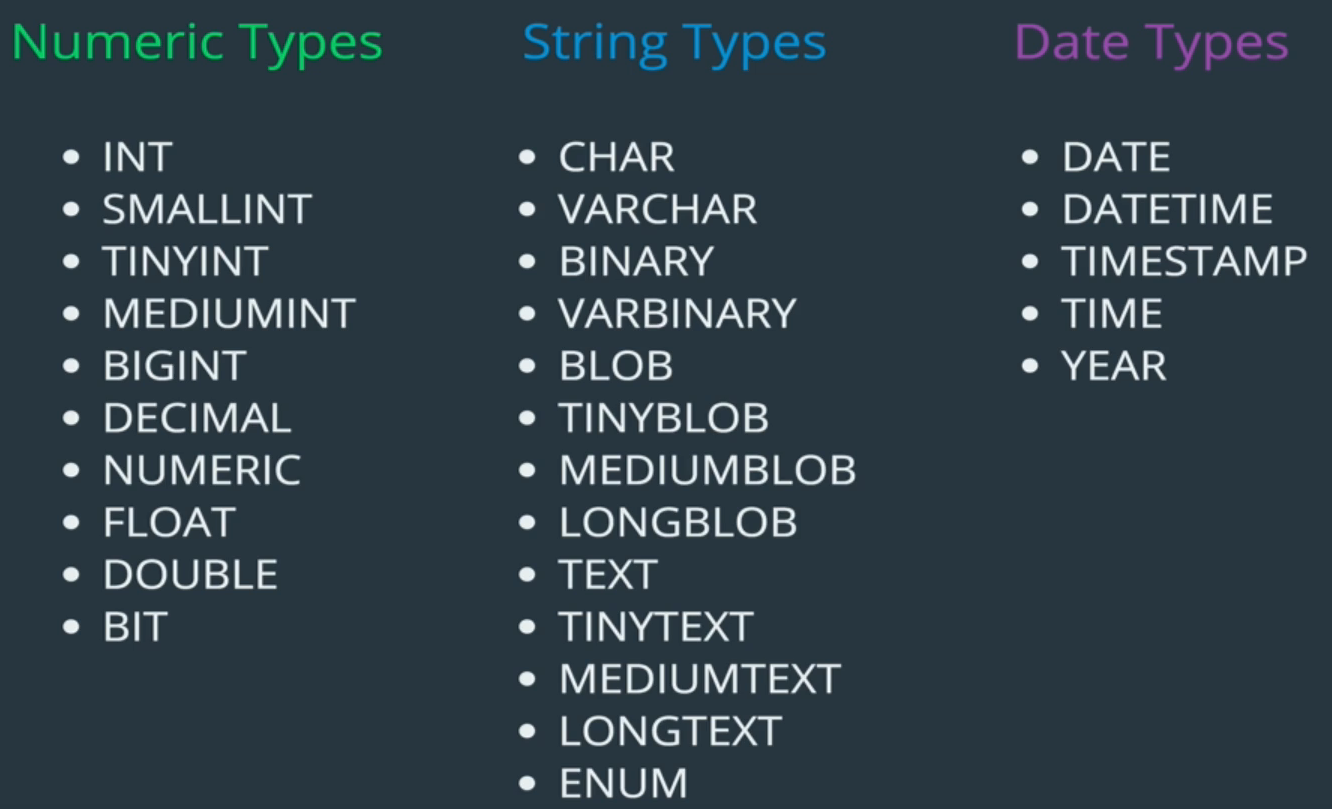
A database is just a bunch of Tables, In a relational database at least.

Tables are collections or related data held in a structural format within the database. (Tables hold the data).

For example, let’s consider the cat’s data and the data of the cat’s table looks as shown below:



Data Types:



As mentioned in the above image is the list of Data Types available in MySQL. But we will focus on only two which are commonly used which are INT for Numeric Type and VARCHAR for String Type.

**INT** is a whole number, having a MAX value of 4294967295

**VARCHAR** A variable length String, it’s between 1 and 255 characters.

**Creating Your Own Tables**

CREATE TABLE tablename

(

    column\_name data\_type,

    column\_name data\_type

);

Example for cat’s data.

CREATE TABLE cats

(

    name VARCHAR(100),

    age INT

);

How do we know the table got created?

SHOW TABLES;

SHOW COLUMNS FROM tablename;

  OR

DESC tablename;

Dropping the table or deleting the table:

DROP TABLE <tablename>;

A specific example: DROP TABLE cats;

Inserting data into tables

Syntax:

INSERT INTO table\_name(column\_name) VALUES (data);

Example:

INSERT INTO cats(name, age) VALUES ('Jetson', 7);

How to know that the data got inserted?

SELECT \* FROM cats;

How to do Multiple Inserts

INSERT INTO table\_name

(column\_name, column\_name)

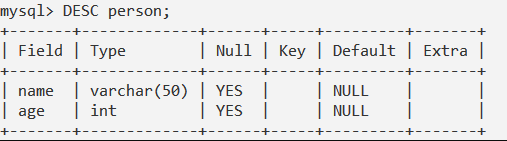
VALUES (value, value),

(value, value),

(value, value);

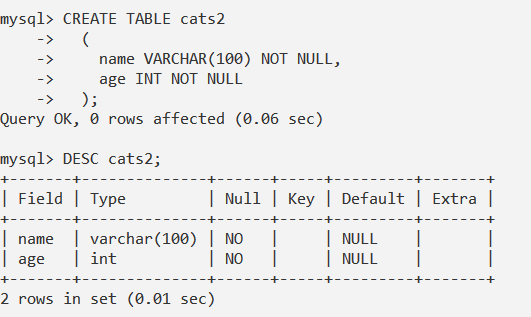
SHOW WARNING; // To know the warning messages.

**NULL and NOT NULL**

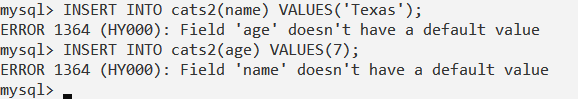


In the above table if we Null column its saying YES YES which means name and age will accept NULL values while inserting into the table.

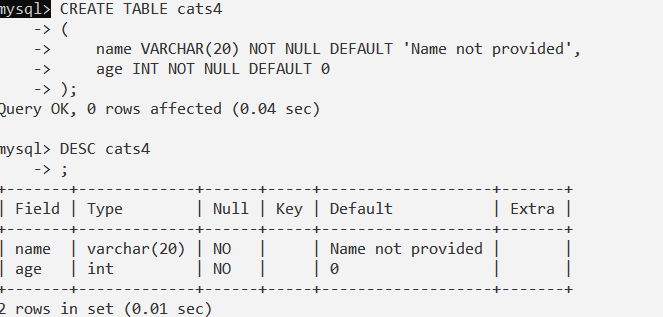
This can be avoided by adding NOT NULL to the column name while creating the table. Like this,

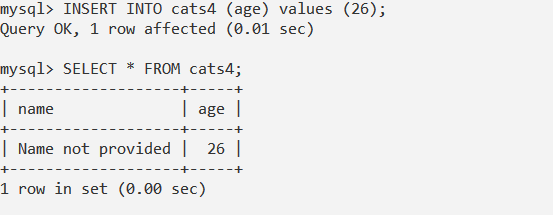


Now inserting NULL values gives an error

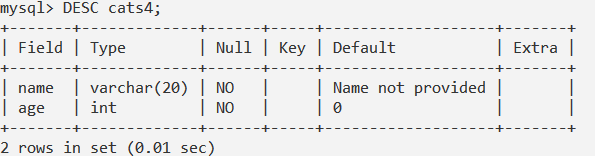


With NOT NULL and DEFAULT



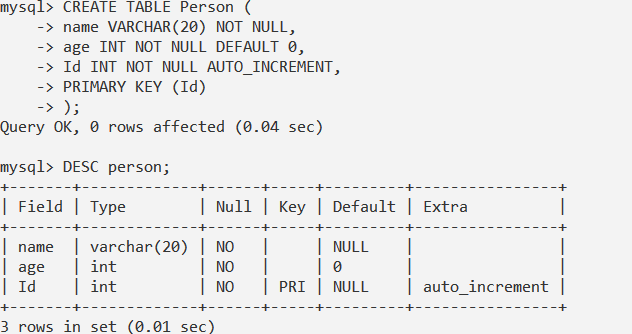


One Other thing



**Primary Key**

The PRIMARY KEY constraint uniquely identifies each record in a database table.



If we would have not added any PRIMARY KEY while creating a Table then, in the above we would have not seen PRI value for Id row under key column.

