CoderDojo

INSTALLING XAMPP

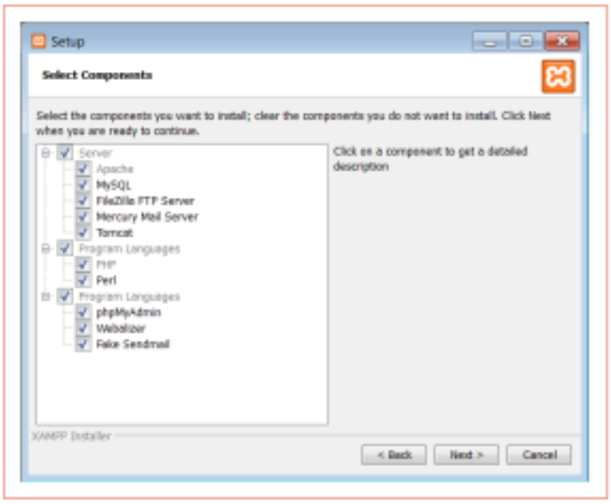
Card 1 of 8

I'm Learning: MySQL

MySQL is the world's most popular open source database. With it's proven performance, reliability and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, Yahoo! and many more.

1 Double click the XAMPP installer (.exe) to start the setup process. A setup wizard will then open and will show you a welcome page. Click **Next** to proceed to the next step.

2 This page shows you what features or components you want to install. Make sure all boxes are ticked and then click **Next**.



Setup

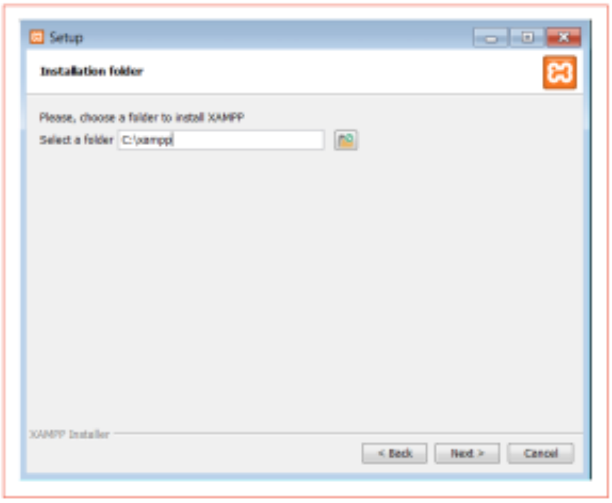
Select Components

Select the components you want to install; clear the components you do not want to install. Click Next when you are ready to continue.

- ✓ Server
  - ✓ Apache
  - ✓ MySQL
  - ✓ FileZilla FTP Server
  - ✓ Mercury Mail Server
  - ✓ Tomcat
- ✓ Program Languages
  - ✓ PHP
  - ✓ Perl
- ✓ Program Languages
  - ✓ phpMyAdmin
  - ✓ Webalizer
  - ✓ Fake Sendmail

Click on a component to get a detailed description

XAMPP Installer



Setup

Installation folder

Please, choose a folder to install XAMPP

Select a folder C:\xampp

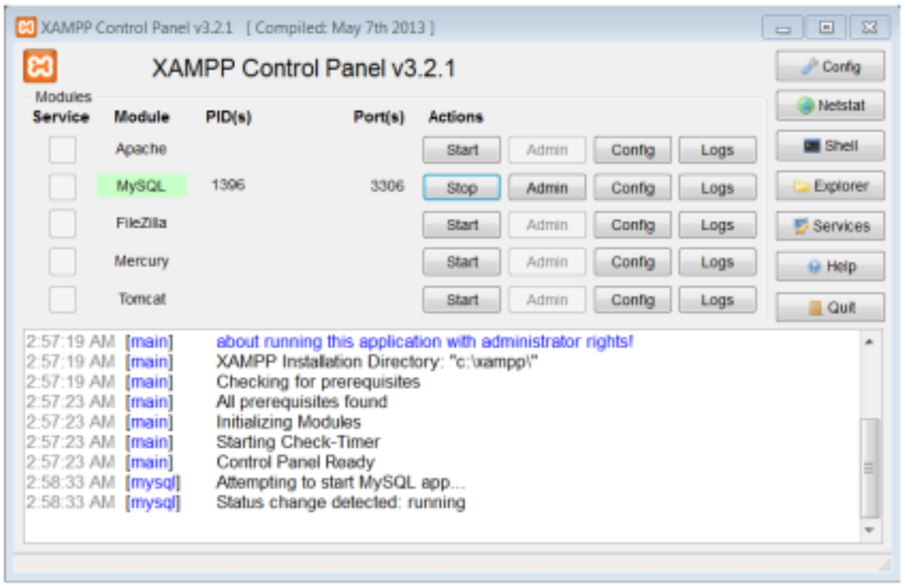
XAMPP Installer

3 You can then choose the directory where you want the XAMPP files to be placed. You can leave thE default path written in the box so it will be installed in your default drive, then click **Next**.

4 **Untick** the box and then click **Next** to proceed with the setup process.

5 Now that we're done with the setup process, click **Next** again to being the begin installing XAMPP in your computer.

6 Wait for the installation process to finish, leave the box ticked in the next page and click **Finish**. This will open the XAMPP Control Panel. You can also open it manually by going to applications folder, then to XAMPP folder, and double clicking **manager-osx.app**.


XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]
XAMPP Control Panel v3.2.1
Config
Netstat
Shell
Explorer
Services
Help
Quit
Modules

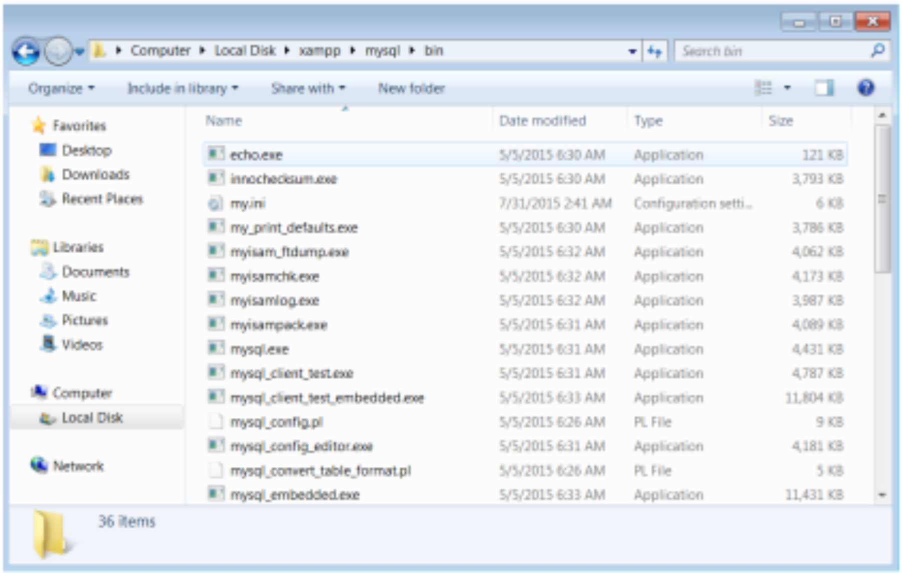
Service	Module	PID(s)	Port(s)	Actions			
	Apache			Start	Admin	Config	Logs
	MySQL	1396	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
2:57:19 AM  [main]   about running this application with administrator rights!
2:57:19 AM  [main]   XAMPP Installation Directory: "c:\xampp\"
2:57:19 AM  [main]   Checking for prerequisites
2:57:23 AM  [main]   All prerequisites found
2:57:23 AM  [main]   Initializing Modules
2:57:23 AM  [main]   Starting Check-Timer
2:57:23 AM  [main]   Control Panel Ready
2:58:33 AM  [mysql]  Attempting to start MySQL app...
2:58:33 AM  [mysql]  Status change detected: running
```

7 In the XAMPP control panel, press the **Start** button beside the MySQL since that's the only feature we will be using for now. After clicking, check that the status of MySQL is green to ensure that it is running properly. **Ask** a mentor for help if a yellow or red colour appears instead.

8 Click the **Explorer** button in the right portion of the control panel. This will take you to the XAMPP folder.

9 In the XAMPP folder, **find** the **mysql** folder and press double click the folder to open. Once inside, find the **bin** folder and open it..



Name	Date modified	Type	Size
echo.exe	5/5/2015 6:30 AM	Application	121 KB
innochecksum.exe	5/5/2015 6:30 AM	Application	3,793 KB
my.ini	7/31/2015 2:41 AM	Configuration setti...	6 KB
my_print_defaults.exe	5/5/2015 6:30 AM	Application	3,786 KB
myisam_ftdump.exe	5/5/2015 6:32 AM	Application	4,062 KB
myisamchk.exe	5/5/2015 6:32 AM	Application	4,173 KB
myisamlog.exe	5/5/2015 6:32 AM	Application	3,987 KB
myisampack.exe	5/5/2015 6:31 AM	Application	4,089 KB
mysql.exe	5/5/2015 6:31 AM	Application	4,431 KB
mysql_client_test.exe	5/5/2015 6:31 AM	Application	4,787 KB
mysql_client_test_embedded.exe	5/5/2015 6:33 AM	Application	11,804 KB
mysql_config.pl	5/5/2015 6:26 AM	PL File	9 KB
mysql_config_editor.exe	5/5/2015 6:31 AM	Application	4,181 KB
mysql_convert_table_format.pl	5/5/2015 6:26 AM	PL File	5 KB
mysql_embedded.exe	5/5/2015 6:33 AM	Application	11,431 KB

10 Insde the bin folder, find the **mysql.exe** file and press double click on your mouse to open it. This will open a console terminal that we will use to create databases.

CoderDojo

USING A DATABASE

Card 2 of 8

I'm Learning: MySQL

SQL stands for Structured Query Language. It is a standard language used for accessing and manipulating databases. Most of the actions you need to perform on a batabase are done with SQL statements.

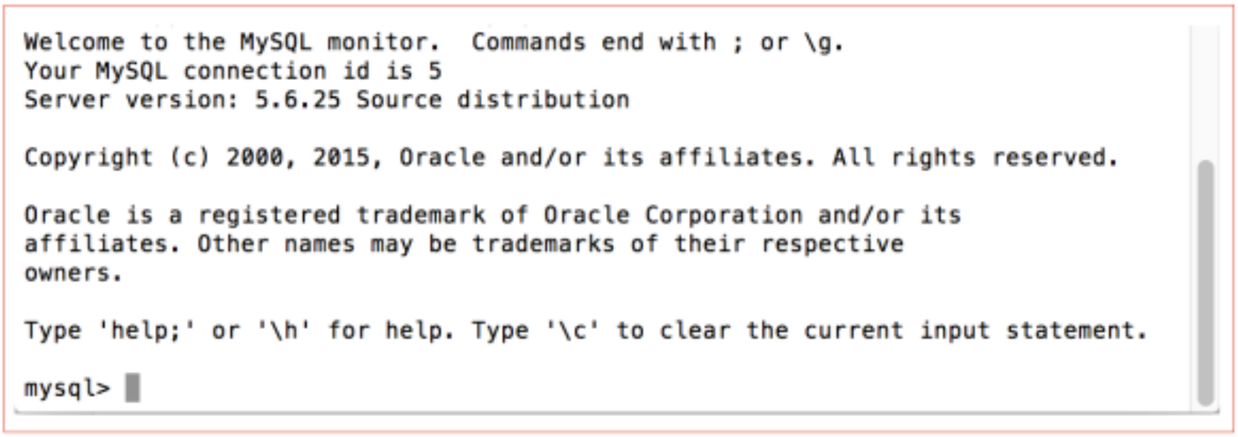
Syntax:

SELECT * FROM Customers **WHERE** name="Mary";

Above is an example of a simple SQL statement to display all customer information in the Customers table with a name of Mary. Notice that it ends with a semicolon (;).

Tips:

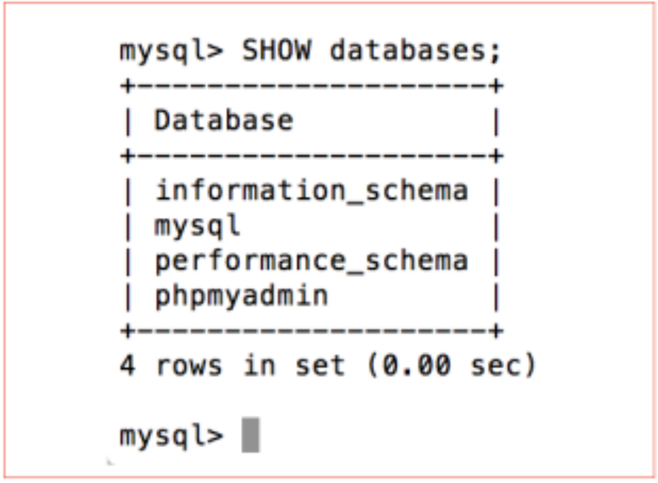
- SQL is **NOT** case sensitive: select is the same as SELECT
- Database systems require semicolon (;) after each SQL statement. So two SQL statement can be written in one line as long as you separate them with with a semicolon.



```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.6.25 Source distribution

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

1 After opening MySQL in yourr terminal write the SQL statement written below to show the list of databases.

```
SHOW databases;
```



```
mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| phpmyadmin         |
+--------------------+
4 rows in set (0.00 sec)

mysql> 
```

CoderDojo

BEGINNER DATABASES

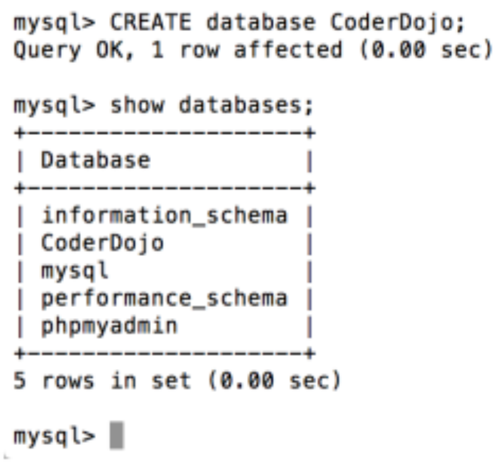
Card 2 of 8

I'm Learning: MySQL

2 Let's create a database named **CoderDojo**. Type the SQL statement written below in the terminal.

```
CREATE database CoderDojo;
```

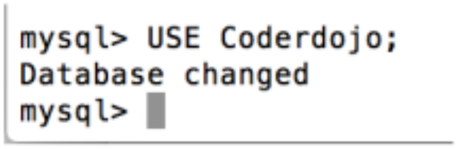
3 Write the SQL statement to show the list of databases again and you will see the that **CoderDojo** is now on the list of databases.



```
mysql> CREATE database CoderDojo;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| CoderDojo          |
| mysql              |
| performance_schema |
| phpmyadmin         |
+--------------------+
5 rows in set (0.00 sec)

mysql> 
```

4 Let's use the CoderDojo database now to start our project. Wirte the command below to **use** SQL statement to use it.

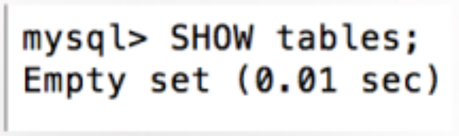
```
USE CoderDojo;
```



```
mysql> USE Coderdojo;
Database changed
mysql> 
```

5 To show the tables that is inside the databases, we need to use the show SQL statement. Write the SQL statement written below in your terminal to show the list of tables currently in the CoderDojo database. Why do you think the result said empty?

```
SHOW tables;
```



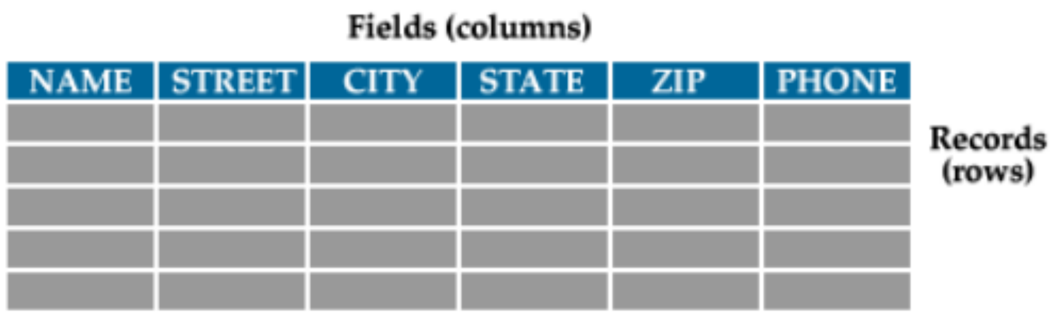
```
mysql> SHOW tables;
Empty set (0.01 sec)
```

CoderDojo

CREATING TABLES

Card 3 of 8

I'm Learning: MySQL



A **Table** is a collection of data in a structured format inside a database. It consists of **columns** (fields) and **rows** (tuples or records).

Syntax on creating tables:

```
                                              data type
CREATE table Persons( person_id int(11) not null auto_increment,
                      first_name varchar(255),
     column name      last_name varchar(255),
                      age int(3),
                      date_of_birth date,
     constraint  ──>  primary key(person_id));
```

There are three parts to remember when creating a table:
column name is the name of a column
data type is the type of data that can be stored in that column
constraints are rules for the data in a table

1 Let's **create** a table to store information about your favourite cartoon shows. We will store a cartoon show's ID, name, description, and the year it was released. **Type** the create SQL statement below to create the Cartoons table.

```
CREATE table Cartoons(cartoon_id int not null auto_increment,
                      cartoon_name varchar(50),
                      description varchar(255),
                      year_released year,
                      primary key(cartoon_id));
```

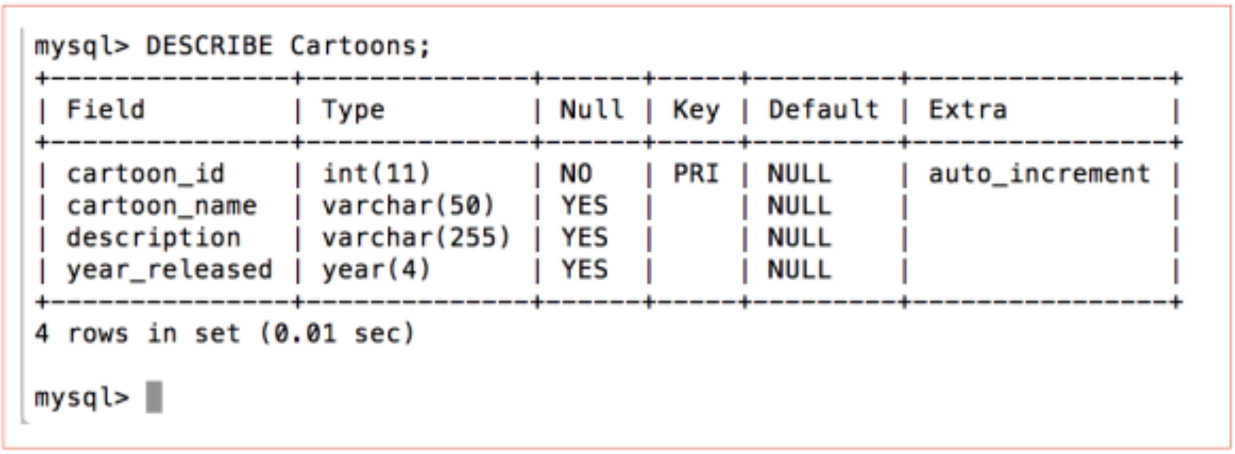
CoderDojo

CREATING TABLES

Card 3 of 8

I'm Learning: MySQL

Tip: Primary key constraint is used to specify which column is used to **uniquely** identify each record of data in the table. That column is now then called Primary key of that table.



```
mysql> DESCRIBE Cartoons;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| cartoon_id    | int(11)      | NO   | PRI | NULL    | auto_increment |
| cartoon_name  | varchar(50)  | YES  |     | NULL    |                |
| description   | varchar(255) | YES  |     | NULL    |                |
| year_released | year(4)      | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)

mysql>
```

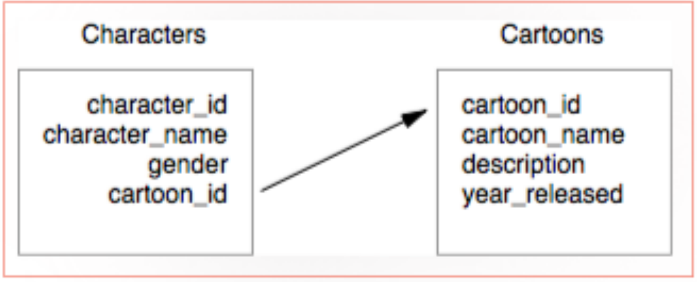
2 After creating the table, you can check the columns and their properties using the **descirbe** SQL statement. Write the SQL statement below to see the properties of each column inside the Cartoons table.

```
DESCRIBE Cartoons;
```

3 Let's make a table another table called **Characters** to store information about cartoon characters and we will **use a foreign key constraint** to **link** characters to a certain show in the Cartoons table.

```
CREATE table Characters(character_id int not null auto_increment,
                        character_name varchar(255),
                        gender enum("male","female","undisclosed"),
                        cartoon_id int,
                        primary key(character_id),
                        foreign key(cartoon_id) REFERENCES Cartoons(cartoon_id));
```

Tip: Foreign key constraint is used to specify which column is used to **link** a record of data to another data in a different table. For example, our **Character** table is now linked to the **Cartoons** table using the **cartoon_id**.

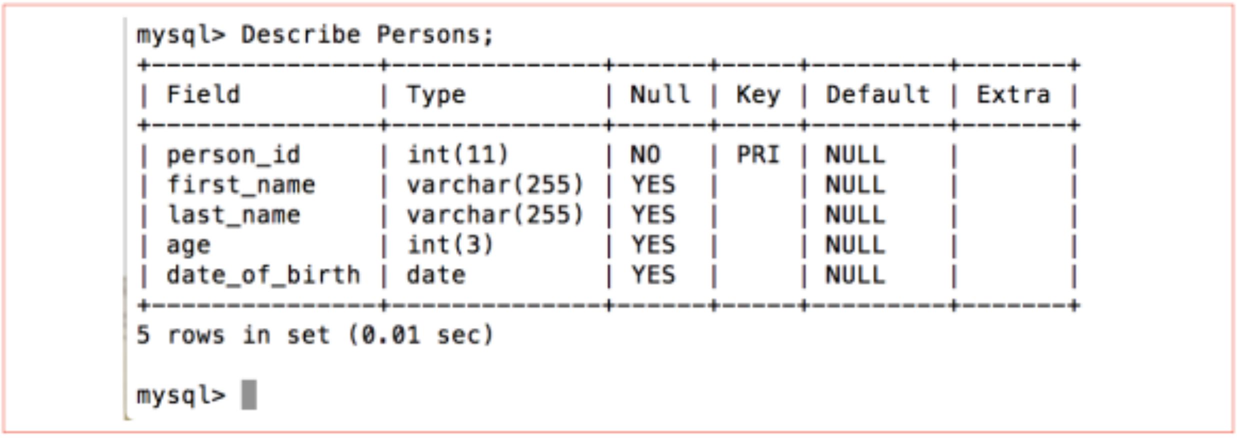


CoderDojo

INSERTING DATA TO TABLES

Card 4 of 8

I'm Learning: MySQL



```
mysql> Describe Persons;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| person_id     | int(11)      | NO   | PRI | NULL    |       |
| first_name    | varchar(255) | YES  |     | NULL    |       |
| last_name     | varchar(255) | YES  |     | NULL    |       |
| age           | int(3)       | YES  |     | NULL    |       |
| date_of_birth | date         | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
5 rows in set (0.01 sec)

mysql> 
```

To insert a new record in a table, we need to use the **INSERT INTO** SQL statement. There are two ways to insert a new record.

Syntax on inserting a new record:

If you want to add a value in **ALL** the columns in a table.

INSERT INTO Persons VALUES(1, "Harry James", "Potter", 35, "1980-07-31");

If you only want to store their only names but not their age and birth date, you need to specify the fields.

INSERT INTO Persons(first_name, last_name) VALUES("Harry James", "Potter");

Tips:

- It's best to check the field names and its datatype using the **DESCRIBE** SQL statement before inserting a new record.
- You can use **auto_increment** constraint on the primary key so you don't have to put a value every time.

1 Let's add the data of **Frozen** cartoon in the Cartoons table!

```
INSERT INTO Cartoons VALUES(1,
                            "Frozen",
                            "Frozen is about Anna who teams up with Kristoff and
                             his reindeer Sven to find Anna's Sister, Elsa.",
                             2013);
```

LIBERTY GLOBAL

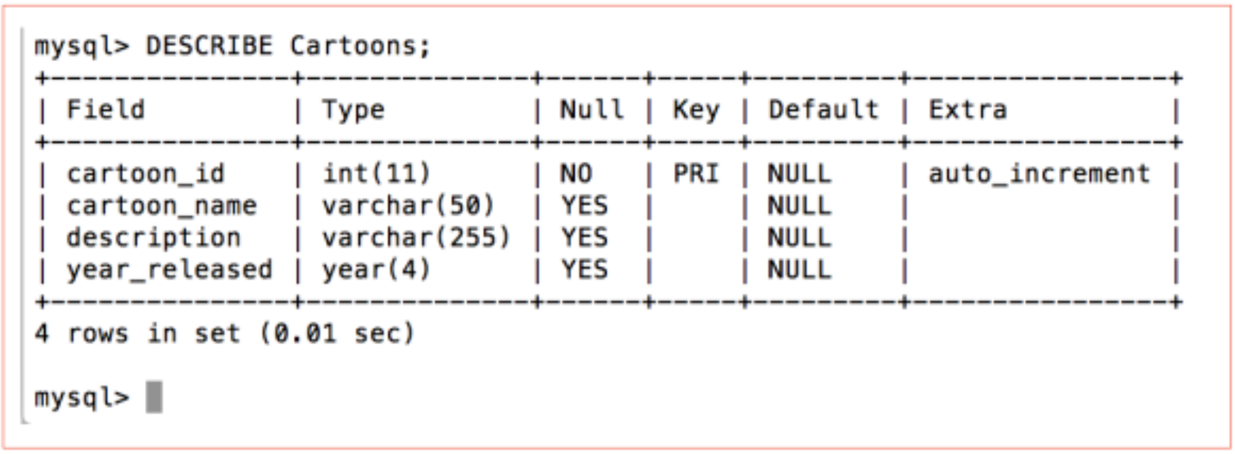
CoderDojo

INSERTING DATA TO TABLES

Card 4 of 8

I'm Learning: MySQL

Tip: auto_increment constraint is used to automatically generate a Primary key id so you won't have to put it manually when inserting a data.



```
mysql> DESCRIBE Cartoons;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| cartoon_id    | int(11)      | NO   | PRI | NULL    | auto_increment |
| cartoon_name  | varchar(50)  | YES  |     | NULL    |                |
| description   | varchar(255) | YES  |     | NULL    |                |
| year_released | year(4)      | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)

mysql> 
```

2 Let's add a new record for another cartoon called **Despicable Me**. This time, we don't need to manually insert the id since we used an auto_increment constraint and also not include a description of the cartoon.

```
INSERT INTO Cartoons(cartoon_name,year_released) VALUES("Despicable Me", 2010);
```

To display the records that you have added, use the SELECT SQL statement below. Try to remember the id of the two cartoons since you'll be using that in the next steps.

```
SELECT * from Cartoons;
```

3 Now let's add a record for **Princess Anna**, a character in Frozen, to the Characters table. We know that Frozen cartoon's ID in the Cartoons table is **1** so we'll put that in the **cartoon_id** as a **foreign key**.

```
INSERT INTO Characters(character_name, gender, cartoon_id)
                VALUES("Princess Anna", "female", 1);
```

Fun Exercise!

Add a record of other characters from Frozen and Despicable Me! If you find it easy to insert records, add a record of **your favourite cartoon shows** in the Cartoons table and its characters in the characters table making sure that they are connected using the correct cartoon_id.

The more data you have on each table, the easier it will be for you to understand the SQL statements in the remaining cards!

CoderDojo

DISPLAYING DATA

Card 5 of 8

I'm Learning: MySQL

SELECT SQL Statement is used to display records in a table.

Syntax:
To display all data in a table.

```
SELECT column_name, column_name FROM table_name;
```

If you want to display only certain data you need to add a WHERE clause in the SELECT statement.

```
SELECT column_name, column_name FROM table_name WHERE [condition];
```

1 To display all the data with all of the columns in a table, you can simply use an asterisk (*) on the SELECT SQL statement. Let's try to display all records in the Characters table.

```
SELECT * FROM Characters;
```

2 You can also display only specific columns of a data by putting the column names instead of an asterisk (*) separated by a comma. Let's try displaying on the data of the character's name and gender.

```
SELECT character_name, gender FROM Characters;
```

3 Let's try adding a WHERE clause on the SELECT statement to display only certain data. Say we want to display characters of Frozen, we know that Frozen's cartoon_id is 1 so we'll be using that as a condition in the WHERE clause.

```
SELECT * FROM Characters WHERE cartoon_id=1;
```

Try playing around with the conditions before moving to the next step!

4 You can also use **multiple conditions** in the WHERE clause using the **AND/OR** operators. Write the SELECT statements below and compare they differ from each other.

```
SELECT * FROM Characters WHERE cartoon_id=1 AND gender="female";
SELECT * FROM Characters WHERE cartoon_id=1 OR gender="female";
```

5 You can use the **LIKE** operator to search for a specified pattern in a column. Try writing the three SELECT statements below with different LIKE parameters and see how they differ from each other. You can change the **letter o** into any letter you prefer.

```
SELECT character_id, character_name FROM characters WHERE character_name LIKE "o%";
SELECT character_id, character_name FROM characters WHERE character_name LIKE "%o";
SELECT character_id, character_name FROM characters WHERE character_name LIKE "%o%";
```

6 To display data in alphabetical order, you will need to add an **ORDER BY** clause on the SELECT statement. Let's try to display the data in the Characters table in ascending nd descending order. Type the SELECT statements below and see how **ORDER BY** clause works.

```
SELECT character_name FROM Characters ORDER BY character_name ASC;
SELECT character_name FROM Characters ORDER BY character_name DESC;
```

Tip: It will alphabetically be ordered in ascending order if you don't put ASC or DESC as parameters in the ORDER BY clause as ascending is the default value.

7 You can order data using a column with number values in it too. Let's try to display the data in Characters table using the character_id column.

```
SELECT * FROM Characters ORDER BY character_id;
SELECT * FROM Characters ORDER BY character_id DESC;
```

8 Let's try using a SELECT statement with a **WHERE** and **ORDER BY** clause. Write the SELECT statement below to display only data of characters from Frozen (or a cartoon show you've added) aphabetically, in descending order.

```
SELECT * FROM Characters WHERE cartoon_id=1 ORDER BY character_name DESC;
```

CoderDojo

UPDATE AND DELETE A DATA

Card 6 of 8

I'm Learning: MySQL

UPDATE SQL Statement is used to update records in a table.

Syntax:

```
UPDATE Persons SET first_name="Claire" WHERE person_id=6;
```

The UPDATE statement above will update the first name of the record with an id of 6 to "Claire".

Tip:

- It is recommended that you have a WHERE clause in your UPDATE statement. Not putting a WHERE clause will result to **all** data in the table being updated.

1 Let's update the name of the record in the Characters table with a character_id of **1** to **Tommy** using the **UPDATE** statement below.

```
UPDATE Characters SET character_name="Tommy" WHERE character_id=1;
```

2 Display the data inside the Characters table using a SELECT statement to see the updated name.

```
SELECT * FROM Characters;
```

Fun Exercise!

Change the names or age in some of the characters in your Character table! Use the character_id as a condition in your WHERE clause to ensure you update the correct one.

DELETE SQL Statement is used to delete records (rows) in a table.

Syntax :

```
DELETE FROM Persons WHERE person_id=6;
```

The DELETE statement above will delete the record with an id value of 6.

Tip:

- The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, **all** records will be deleted!

3 Let's delete the record in the Characters table with a character_id of 7 using the **DELETE** statement below. You can change the condition if you prefer!

```
DELETE FROM Characters WHERE character_id=7;
```

4 Display the data inside the Characters table using a SELECT statement. D you still see a record with a character_id of 7 (or whatever condition you've chosen)?

```
SELECT * FROM Characters;
```

Fun Exercise!

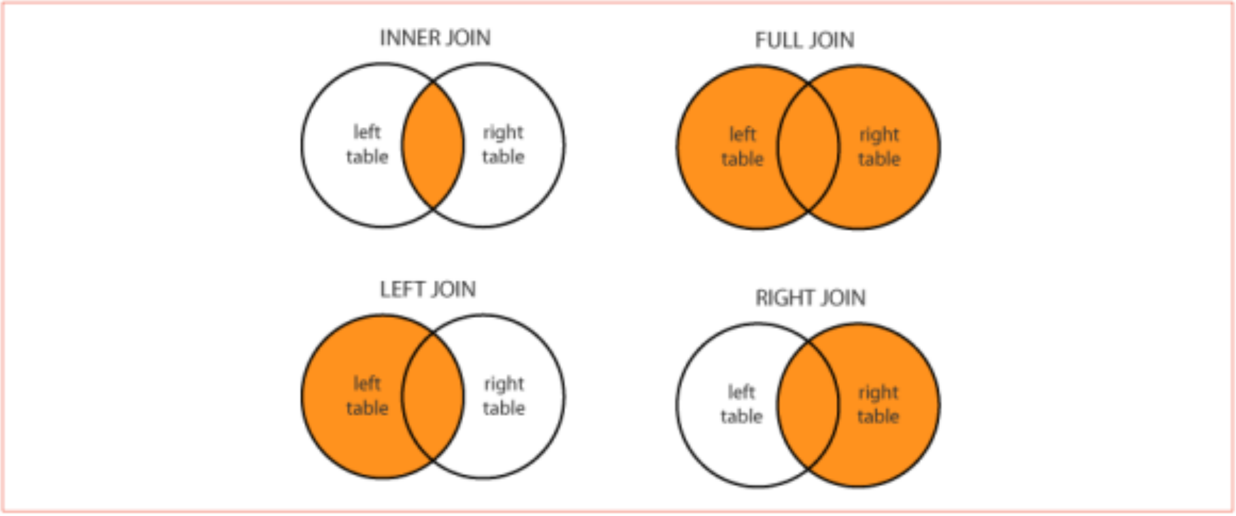
Delete some of the records in the Characters table using the **DELETE** statement and with different conditions.

CoderDojo

JOINING TABLES

Card 7 of 8

I'm Learning: MySQL



An **SQL JOIN** clause is used to combine rows from two or more tables, based on a common field between them (in our case, the cartoon_id is the common field).

Syntax:

```
SELECT p.person_id, p.person_name, o.order_name
FROM Persons p JOIN Orders o ON p.person_id=o.order_id;
```

Tip:

- **p** and **o** in the syntax above are **aliases** of a table to reduce the amount of typing required to enter a query. It can be any word/letter you want them to be.

1 To better explain thte different kinds of **SQL JOINS**, we need to add more records on the tables. Add the data for cartoon shows **Inside Out** and **Spongebob Squarepants** in the Cartoons table. None of its characters will be added in the Characters table.

```
INSERT INTO Cartoons(cartoon_name, description, year_released)
VALUES("Spongebob Squarepants",
        "The adventure of spongebob and his friends in the city of Bikini Bottom.",
         1999);

INSERT INTO Cartoons(cartoon_name, description, year_released)
VALUES("Inside Out",
        "Riley Anderson tries to lead through life as she moves to a new city.",
         2015);
```

CoderDojo

JOINING TABLES

2 **Add** records for the characters of the cartoon show **Kung Fu Panda** in the Characters table **without** adding a value on the cartoon_id column as we have not added Kung Fu Panda in the Cartoons table..

```
INSERT INTO Characters(character_name, gender) VALUES("Po", "male");
INSERT INTO Characters(character_name, gender) VALUES("Tigress", "female");
INSERT INTO Characters(character_name, gender) VALUES("Shifu", "male");
INSERT INTO Characters(character_name, gender) VALUES("Viper", "female");
```

3 Let's use an **INNER JOIN** to display the list of characters and the cartoon show they are part of. You will see that only characters linked to a cartoon show are visible. Note that the two tables are joined by the common field **cartoon_id** in the **ON** clause.

```
SELECT ch.character_name, c.cartoon_name FROM Characters ch
JOIN Cartoons c ON ch.cartoon_id=c.cartoon_id;
```

4 We can use the **LEFT JOIN** to display a list of **all** the characters in the characters table regardless of whether or not they are linked to a record in the Cartoons table. You will see that characters not linked to any cartoons will have a **Null** value.

```
SELECT ch.character_name, c.cartoon_name FROM Characters ch
LEFT JOIN Cartoons c ON ch.cartoon_id=c.cartoon_id;
```

5 We can use the **RIGHT JOIN** to display all the data in the right table, in this case the Cartoons table, to display all its records first then check if it is linked to any record in the Characters table. If it's not linked to any Characters, it will display it as **Null**.

```
SELECT ch.character_name, c.cartoon_name FROM Characters ch
RIGHT JOIN Cartoons c ON ch.cartoon_id=c.cartoon_id;
```

Fun Exercise!

Can you display a list of all female characters in the Characters table and the cartoon show they are part of? **Hint:** You will need to use a WHERE clause

BEGINNER DATABASES

DELETE A TABLE/DATABASE

Card 8 of 8

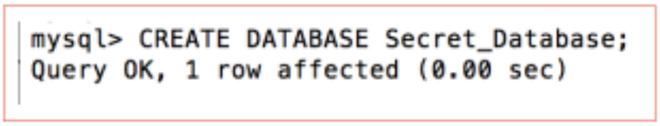
I'm Learning: MySQL

DROP TABLE SQL statement is used to delete a table.
Syntax:
    DROP TABLE table_name;

DROP DATABASE SQL statement is used to delete a database.
Syntax:
    DROP DATABASE database_name;

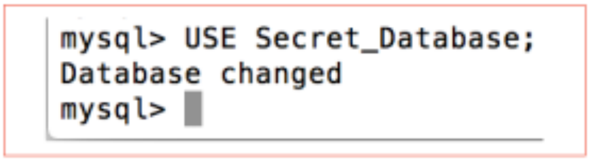
1 Let's **create** a new database and name it as **Secret_Database**.

```
CREATE DATABASE Secret_Database;
```

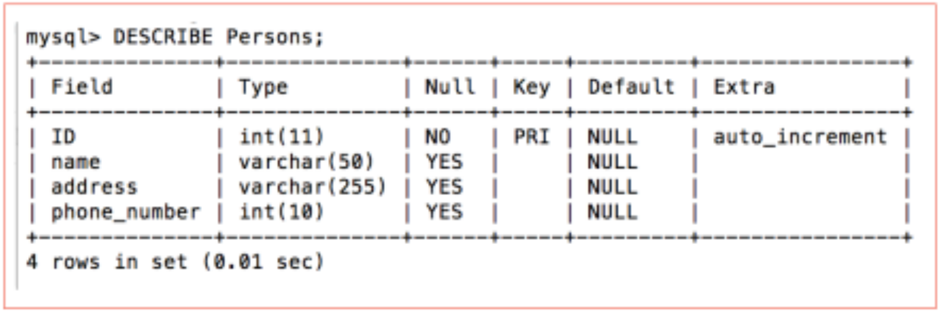


```
mysql> CREATE DATABASE Secret_Database;
Query OK, 1 row affected (0.00 sec)
```

2 **Use Secret_Database** and **create** a new table inside it called **Persons** with an ID, name, address, and a phone number column.



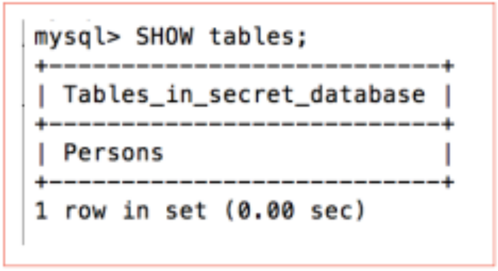
```
mysql> USE Secret_Database;
Database changed
mysql> 
```



```
mysql> DESCRIBE Persons;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| ID           | int(11)      | NO   | PRI | NULL    | auto_increment |
| name         | varchar(50)  | YES  |     | NULL    |                |
| address      | varchar(255) | YES  |     | NULL    |                |
| phone_number | int(10)      | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

3 Show the list of tables inside the **Secret_Database** and you will see that Persons table is now listed there.

```
Show tables;
```



```
mysql> SHOW tables;
+---------------------------+
| Tables_in_secret_database |
+---------------------------+
| Persons                   |
+---------------------------+
1 row in set (0.00 sec)
```

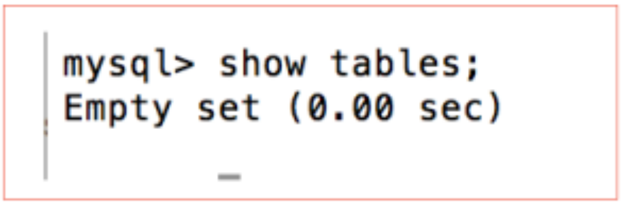
CoderDojo

DELETE A TABLE/DATABASE

4 Let's use the **DROP TABLE** statement to delete Persons table.

```
DROP TABLE Persons;
```

5 **Show** the list of tables again using the **SHOW** statement and you will see that Person table has now been deleted and Secret_Database is now empty.



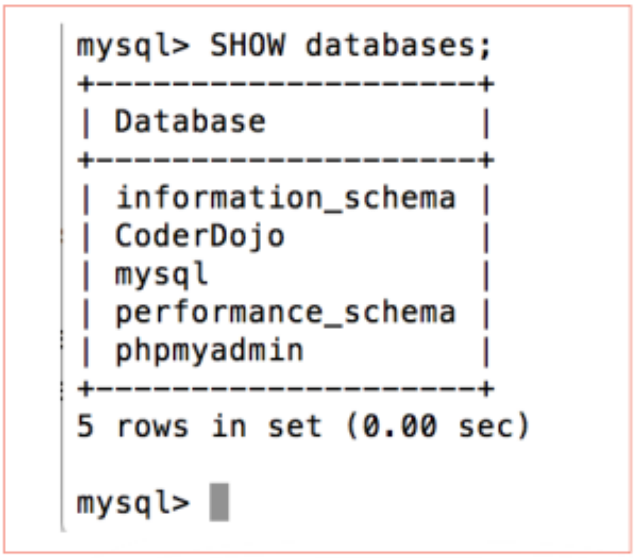
```
mysql> show tables;
Empty set (0.00 sec)
```

6 Let's use the **DROP DATABASE** statement to delete Secret_Database.

```
DROP DATABASE Secret_Database;
```

7 **Display** the list of databases and you will find that **Secret_Database** is not in the list anymore as it has now been deleted.

8 To quit MySQL in your console, simply type **exit** and press enter.



```
mysql> SHOW databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| CoderDojo          |
| mysql              |
| performance_schema |
| phpmyadmin         |
+--------------------+
5 rows in set (0.00 sec)

mysql> 
```

Great job! You've now learned the basics of Databases!.