C1- Code & Go

C-WEB-150

SQL

Day 01

CODING ACADEMY
BY EPITECH

Administrative Details

- The project must be done alone
- Sources must be turned in with BLIH
- From this day onward there will not be videos, you can find documentation for SQL here: `https://www.w3schools.com/sql/default.asp`

Introduction

For the next two days, you will become familiar with databases and the universal language to communicate with them: SQL.

There are many database management systems (DBMS): Microsoft SQL Server and Oracle for commercial DBMS, and SQLite, PostgreSQL, MariaDB or even MySQL for free and open DBMS.

For practical reasons, we will use MySQL which is free and is also one the most widely used DBMS.

You should take note that we will not be testing with the same DB as you. Which means that you shall only use general request and no request like "id = 42" for example.

Restriction

To install MySQL, open a terminal and execute the following command:

```
Terminal
~/C-WEB-150> sudo apt-get install mysql-client mysql-server
```

During installation process, you will be prompt to enter a "root" password to connect to MySQL. Once the installation is done, you can connect to MySQL with the command:

```
Terminal
~/C-WEB-150> mysql -u root -p
```

Enter the password you set during the installation. You are now in the MySQL command prompt.

Procedure

Once you're in the MySQL console, you should be able to create a database with the following command:

Terminal

```
~/C-WEB-150> mysql -u root -p
> # To 'use' this database, type:
> USE coding;
> # You now need to import the 'coding.sql' database available on the intranet.
> # To do this, type the following command:
> source <path to your coding.sql file>
> # You now have everything you need to write your SQL requests in the console in front of
you.
```

Exercises

Exercise 01: 1 Pts

File to turn in: SQL_Day_01/ex_01/ex_01.sql
Write a query that displays the list of all the tables in the database

Exercise 02: 1 Pts

File to turn in: SQL_Day_01/ex_02/ex_02.sql
Write a query that displays the description of the **movies** table.

Exercise 03: 1 Pts

File to turn in: SQL_Day_01/ex_03/ex_03.sql
Write a query that displays the current date in a column "Date" with the format "YYYY-MM- DD".

Exercise 04: 1 Pts

File to turn in: SQL_Day_01/ex_04/ex_04.sql
Write a query that displays the **title** and **summary** of all the movies sorted in alphabetical order.

Exercise 05: 1 Pts

File to turn in: SQL_Day_01/ex_05/ex_05.sql
Write a query that displays the **name** of all the genres in the table **genres** in uppercase. The column will be named "NAME OF ALL THE GENRES"

Exercise 06: 1 Pts

File to turn in: SQL_Day_01/ex_06/ex_06.sql
Write a query that displays the **title** of the last 42 movies in the table **movies**. The column will be named "Title of the last 42 movies". The results must be ordered by decreasing id.

Exercise 07: 1 Pts

File to turn in: SQL_Day_01/ex_07/ex_07.sql
Write a query that displays the **name** of the most expensive subscription in the **subscriptions** table, as well as its **price**. The columns will be respectively named: "Name of the most expensive subscription" and "Price".

Exercise 08: 1 Pts

File to turn in: SQL_Day_01/ex_08/ex_08.sql
Write a query that displays the movie's title whose genre is "action" or "romance"

Exercise 09: 1 Pts

File to turn in: SQL_Day_01/ex_09/ex_09.sql
Write a query that displays how long the shortest movie is in minutes. The movies with NULL or 0 duration should not be taken into account. The column will be named "Duration of the shortest movie".

Exercise 10: 1 Pts

File to turn in: SQL_Day_01/ex_10/ex_10.sql
Write a query that displays the id of the movies whose title contains the chain of characters "tard" (case-insensitive). The column will be named "Identifier".

Exercise 11: 1 Pts

File to turn in: SQL_Day_01/ex_11/ex_11.sql
Write a query that displays the number of movies whose title ends up with the string "tion" (case- insensitive). The column will be named "Number of movies ending with 'tion'".

Exercise 12: 1 Pts

File to turn in: SQL_Day_01/ex_12/ex_12.sql
Write a query that displays the total number of movies whose genre is "western" and whose producers is "tartan movies" or "lionsgate uk". The column will be named "Number of 'western' movies".

Exercise 13: 1 Pts

File to turn in: SQL_Day_01/ex_13/ex_13.sql
Write a query that displays the number of a room and its name, for the rooms that have more than 0 seats and that are not on the first floor (First floor and not ground floor). The columns will be named ''Room numbers'' and ''Room names''.

Exercise 14: 1 Pts

File to turn in: SQL_Day_01/ex_14/ex_14.sql
Write a query that displays the number of movies whose title starts with “eX” (case-sensitive). The column must be named 'Number of movies that starts with ''eX''".

Exercise 15: 1 Pts

File to turn in: SQL_Day_01/ex_15/ex_15.sql
Write a query that displays the average duration of a movie rounded to 2 decimals. The column will be named ''Average duration''.

Exercise 16: 1 Pts

File to turn in: SQL_Day_01/ex_16/ex_16.sql
Write a query that displays the month of birth in English from the 42nd to the 84th member (the 42nd and the 84th must be included). The column will be named ''month of birth''.

Exercise 17: 1 Pts

File to turn in: SQL_Day_01/ex_17/ex_17.sql
Write a query that displays the **title** of the longest movie. The column will be named: ''Title of the longest movie''.

Exercise 18: 1 Pts

File to turn in: SQL_Day_01/ex_18/ex_18.sql
Write a query that displays the **lastname** followed by a dash, followed by the **firstname** of each member from the table **profiles**. The first letter of the last name and the first name's first letter will be in upper case. The members should be displayed from the youngest to the oldest. The column will be named: ''Full name''.

Exercise 19: 1 Pts

File to turn in: SQL_Day_01/ex_19/ex_19.sql
Write a query that displays the **title** of the movies whose **id** is 21, 87, 263, 413 or 633. The column will be named "Movie title".

Exercise 20: 1 Pts

File to turn in: SQL_Day_01/ex_20/ex_20.sql
Write a query that displays the number of produced movies per year. The year must not be 0. The result has to be ordered by decreasing year of production. The columns will be named "Number of movies" and "Year of production".