T - Web Development Seminar

T-WEB-x00

Day 11

SQL

1.45

Day 11

repository name: web_seminar_day11_$ACADEMICYEAR
repository rights: ramassage-tek
language: SQL

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).

Thanks to the coming SQL 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,
- SQLite, PostgreSQL, MariaDB or even MySQL for free and open DBMS.

For practical reasons, we will use **MySQL** which a very common free DBMS.

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

EXERCISE 0 - SET UP

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

```
Terminal
~/T-WEB-x00> docker pull mariadb
```

Create a "datadir" directory containing the "coding.sql" file available in the intranet.

Then execute

```
Terminal
~/T-WEB-x00> docker run -name some-mariadb -v /path/to/datadir/:/var/lib/mysql -e
MYSQL_ROOT_PASSWORD=gecko -d mariadb:latest
docker exec -it some-mariadb bash
```

You can now execute MariaDB:

```
Terminal
~/T-WEB-x00> docker start some-mariadb
```

You are now in the MySQL console.
Create a database with the following command:

```
CREATE DATABASE coding;
```

To "use" this database, type:

```
USE coding;
```

Finally, to populate the database, enter:

```
SOURCE /var/lib/mysql/coding.sql;
```

You now have everything you need to write your SQL requests in the console in front of you.

EXERCISE 1 — 1PT

Turn in: ./ex_01/ex_01.sql

Write a query that displays the **list of all the tables** in the database.

EXERCISE 2 — 1PT

Turn in: ./ex_02/ex_02.sql

Write a query that displays the **description** of the movies table.

EXERCISE 3 — 1PT

Turn in: ./ex_03/ex_03.sql

Write a query that displays the **current date** in a column "Date" at the "YYYY-MM-DD" format.

EXERCISE 4 — 1PT

Turn in: ./ex_04/ex_04.sql

Write a query that displays the **title and summary** of all the movies sorted in alphabetical order.

EXERCISE 5 — 1PT

Turn in: ./ex_05/ex_05.sql

Write a query that displays the **name of all the genres** in the table genres in uppercase.
The column must be named "NAME OF ALL THE GENRES".

EXERCISE 6 — 1PT

Turn in: ./ex_06/ex_06.sql

Write a query that displays the **title of the last 42 movies** in the table movies.
The column must be named "Title of the last 42 movies".
The results must be ordered by id descendant.

EXERCISE 7 — 1PT

Turn in: ./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 must be respectively named: "Name of the most expensive subscription" and "Price".

EXERCISE 8 — 1PT

Turn in: ./ex_08/ex_08.sql

Write a query that displays the movies' **title** whose genre is "action" or "romance".

EXERCISE 9 — 1PT

Turn in: ./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 must be named "Duration of the shortest movie".

EXERCISE 10 — 1PT

Turn in: ./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 must be named "Identifier".

EXERCISE 11 — 1PT

Turn in: ./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 must be named "Number of movies ending with 'tion' ".

EXERCISE 12 — 1PT

Turn in: ./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 must be named "Number of 'western' movies".

EXERCISE 13 — 1PT

Turn in: ./ex_13/ex_13.sql

Write a query that displays the **number and name** of the rooms that have more than 0 seats and that are not on the first floor (first floor, not ground floor).
The columns must be named "Room numbers" and "Room names".

EXERCISE 14 — 1PT

Turn in: ./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 — 1PT

Turn in: ./ex_15/ex_15.sql

Write a query that displays the **average duration** of the movies rounded to 2 decimals.
The column must be named "Average duration".

EXERCISE 16 — 1PT

Turn in: ./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 must be named "month of birth".

EXERCISE 17 — 1PT

Turn in: ./ex_17/ex_17.sql

Write a query that displays the **title of the longest movie**.
The column must be named "Title of the longest movie".

EXERCISE 18 — 1PT

Turn in: ./ex_18/ex_18.sql

Write a query that displays the **last name 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 must be in upper case.
The members should be displayed from the youngest to the oldest.
The column must be named "Full name".

EXERCISE 19 — 1PT

Turn in: ./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 must be named "Movie title".

EXERCISE 20 — 1PT

Turn in: ./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 descending year of production.
The columns must be named "Number of movies" and "Year of production".