You have a captain's log due before 2024-04-21 (in 1 day)! Log it now!
(/captain_logs/5596018/edit)

0x0E. SQL - More queries

SQL **MySQL**

Weight: 1

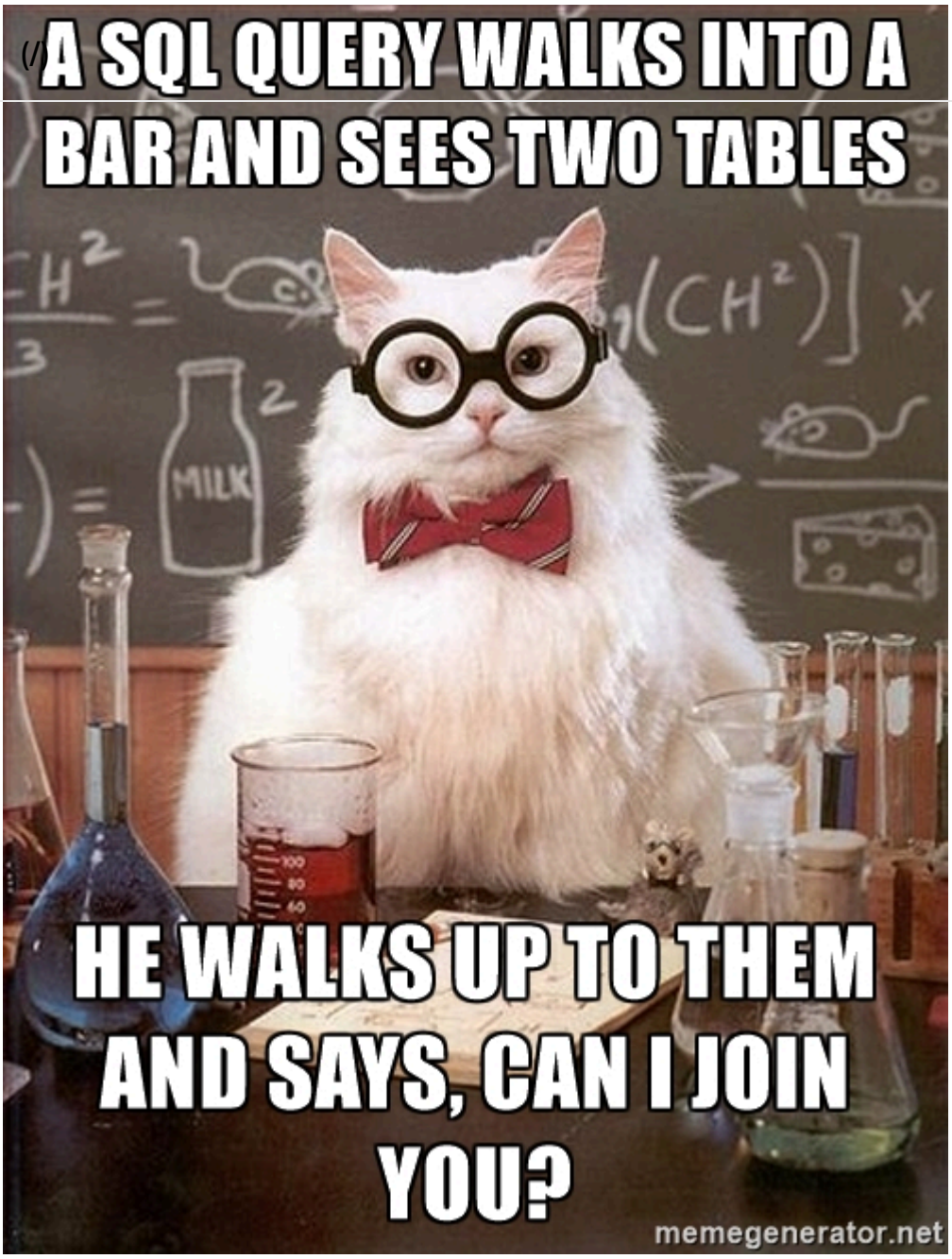
Project over - took place from Dec 13, 2023 6:00 AM to Dec 14, 2023 6:00 AM

An auto review will be launched at the deadline

In a nutshell…

- **Auto QA review:** 96.0/96 mandatory & 24.0/24 optional
- **Altogether: 200.0%**
  - Mandatory: 100.0%
  - Optional: 100.0%
  - Calculation: 100.0% + (100.0% * 100.0%) == **200.0%**

(/)



Resources

Read or watch:

- How To Create a New User and Grant Permissions in MySQL (/rltoken/RniBKj48bnlN8xpXhGl1yA)
- How To Use MySQL GRANT Statement To Grant Privileges To a User (/rltoken/lhaKcRpljC_zRu1NRTpU6Q)
- MySQL constraints (/rltoken/LrovGa6N-OE2ID_tpWZRaQ)
- SQL technique: subqueries (/rltoken/kR71h5zjkPtx4kBoVf7q0g)
- Basic query operation: the join (/rltoken/rNMJeQ1jbNTCljbvCSjf6w)
- SQL technique: multiple joins and the distinct keyword (/rltoken/HhZ6TJ1q5S0aR4lhfpKdOQ)
- SQL technique: join types (/rltoken/T6FZUQdsMzr8hgNlnBzudA)
- SQL technique: union and minus (/rltoken/Nd-sdM8QUpf0YKIlXzVv4w)
- MySQL Cheat Sheet (/rltoken/iSNyinU6SPWTGDUWMmcRkg)
- The Seven Types of SQL Joins (/rltoken/-plhBsra0N7BOuFoEg--zg)

- MySQL Tutorial (/rltoken/I4Lws_eQrIrNTbkZvvk-oQ)
- SQL Style Guide (/rltoken/051eAEP_rePBU7jeh879GA)
- MySQL 8.0 SQL Statement Syntax (/rltoken/YavbYiraYFr8oTukT_N6eQ)

Extra resources around relational database model design:

- Design (/rltoken/EWLRPeqr5sQ9AqfoG_KXxw)
- Normalization (/rltoken/mqBhYoSYbhH5ZZrhDcY0kA)
- ER Modeling (/rltoken/R0exkJmf-2ddKjGfa8D0dA)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/rltoken/0qci3VdlVdKJXldEZ6zAjA), **without the help of Google**:

General

- How to create a new MySQL user
- How to manage privileges for a user to a database or table
- What's a `PRIMARY KEY`
- What's a `FOREIGN KEY`
- How to use `NOT NULL` and `UNIQUE` constraints
- How to retrieve datas from multiple tables in one request
- What are subqueries
- What are `JOIN` and `UNION`

Copyright - Plagiarism

- You are tasked to come up with solutions for the tasks below yourself to meet with the above learning objectives.
- You will not be able to meet the objectives of this or any following project by copying and pasting someone else's work.
- You are not allowed to publish any content of this project.
- Any form of plagiarism is strictly forbidden and will result in removal from the program.

Requirements

General

- Allowed editors: `vi`, `vim`, `emacs`
- All your files will be executed on Ubuntu 20.04 LTS using `MySQL 8.0` (version 8.0.25)
- All your files should end with a new line
- All your SQL queries should have a comment just before (i.e. syntax above)
- All your files should start by a comment describing the task
- All SQL keywords should be in uppercase (`SELECT`, `WHERE` ...)
- A `README.md` file, at the root of the folder of the project, is mandatory
- The length of your files will be tested using `wc`

More Info

Comments for your SQL file:

```
$ cat my_script.sql
-- 3 first students in the Batch ID=3
-- because Batch 3 is the best!
SELECT id, name FROM students WHERE batch_id = 3 ORDER BY created_at DESC LIMIT 3;
$
```

Install MySQL 8.0 on Ubuntu 20.04 LTS

```
$ sudo apt update
$ sudo apt install mysql-server
...
$ mysql --version
mysql  Ver 8.0.25-0ubuntu0.20.04.1 for Linux on x86_64 ((Ubuntu))
$
```

Connect to your MySQL server:

```
$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.25-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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>
mysql> quit
Bye
$
```

Use “container-on-demand” to run MySQL

In the container, credentials are `root/root`

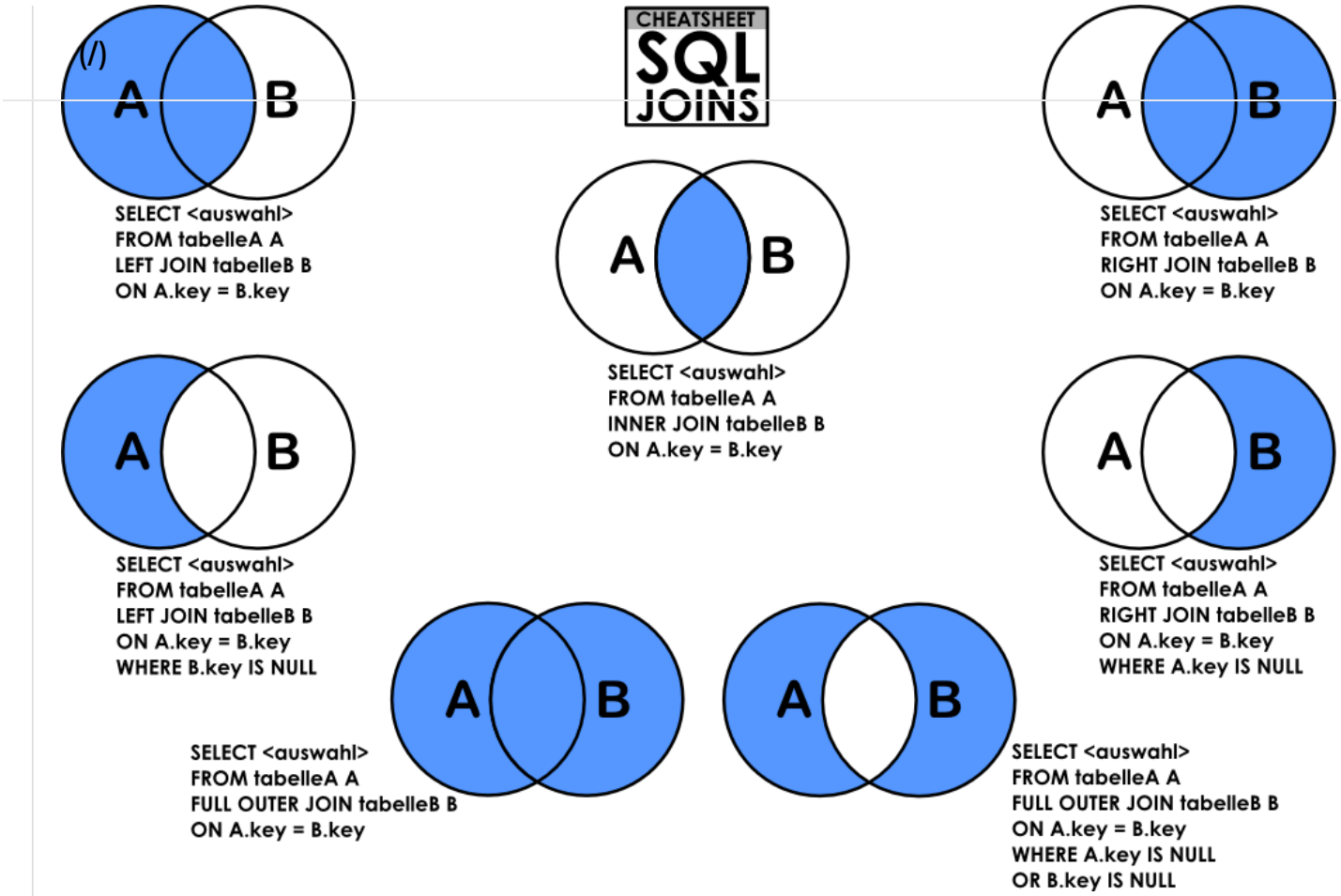
- Ask for container `Ubuntu 20.04`
- Connect via SSH
- OR connect via the Web terminal
- In the container, you should start MySQL before playing with it:

```
$ service mysql start
 * Starting MySQL database server mysqld
$
$ cat 0-list_databases.sql | mysql -uroot -p
Database
information_schema
mysql
performance_schema
sys
$
```

In the container, credentials are `root/root`

How to import a SQL dump

```
$ echo "CREATE DATABASE hbtn_0d_tvshows;" | mysql -uroot -p
Enter password:
$ curl "https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programa
mming+/274/hbtn_0d_tvshows.sql" -s | mysql -uroot -p hbtn_0d_tvshows
Enter password:
$ echo "SELECT * FROM tv_genres" | mysql -uroot -p hbtn_0d_tvshows
Enter password:
id  name
1   Drama
2   Mystery
3   Adventure
4   Fantasy
5   Comedy
6   Crime
7   Suspense
8   Thriller
$
```



Quiz questions

Great! You've completed the quiz successfully! Keep going! (Show quiz)

Tasks

0. My privileges!

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that lists all privileges of the MySQL users `user_0d_1` and `user_0d_2` on your server (in `localhost`).

(7)

```
guillaume@ubuntu:~/$ cat 0-privileges.sql | mysql -hlocalhost -uroot -p
Enter password:
ERROR 1141 (42000) at line 3: There is no such grant defined for user 'user_0d_1' on host 'l
ocalhost'
guillaume@ubuntu:~/$
guillaume@ubuntu:~/$ echo "CREATE USER 'user_0d_1'@'localhost';" |  mysql -hlocalhost -uroot
-p
Enter password:
guillaume@ubuntu:~/$ echo "GRANT ALL PRIVILEGES ON *.* TO 'user_0d_1'@'localhost';" |  mysql
-hlocalhost -uroot -p
Enter password:
guillaume@ubuntu:~/$ cat 0-privileges.sql | mysql -hlocalhost -uroot -p
Enter password:
Grants for user_0d_1@localhost
GRANT SELECT, INSERT, UPDA..., DROP ROLE ON *.* TO `user_0d_1`@`localhost`
GRANT APPLICATION_PASSWORD_ADMIN,AUDIT...,XA_RECOVER_ADMIN ON *.* TO `user_0d_1`@`localhost`
ERROR 1141 (42000) at line 4: There is no such grant defined for user 'user_0d_2' on host 'l
ocalhost'
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `0-privileges.sql`

Done! | Check your code | >_ Get a sandbox | QA Review

1. Root user

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that creates the MySQL server user `user_0d_1` .

- `user_0d_1` should have all privileges on your MySQL server
- The `user_0d_1` password should be set to `user_0d_1_pwd`
- If the user `user_0d_1` already exists, your script should not fail

(7)

```
guillaume@ubuntu:~/$ cat 1-create_user.sql | mysql -hlocalhost -uroot -p
Enter password:
guillaume@ubuntu:~/$ cat 0-privileges.sql | mysql -hlocalhost -uroot -p
Enter password:
Grants for user_0d_1@localhost
GRANT SELECT, INSERT..., DROP ROLE ON *.* TO `user_0d_1`@`localhost`
GRANT APPLICATION_PASSWORD_ADMIN,...,XA_RECOVER_ADMIN ON *.* TO `user_0d_1`@`localhost`
ERROR 1141 (42000) at line 4: There is no such grant defined for user 'user_0d_2' on host 'l
ocalhost'
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `1-create_user.sql`

Done! | Check your code | Get a sandbox | QA Review

2. Read user

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that creates the database `hbtn_0d_2` and the user `user_0d_2`.

- `user_0d_2` should have only SELECT privilege in the database `hbtn_0d_2`
- The `user_0d_2` password should be set to `user_0d_2_pwd`
- If the database `hbtn_0d_2` already exists, your script should not fail
- If the user `user_0d_2` already exists, your script should not fail

```
guillaume@ubuntu:~/$ cat 2-create_read_user.sql | mysql -hlocalhost -uroot -p
Enter password:
guillaume@ubuntu:~/$ cat 0-privileges.sql | mysql -hlocalhost -uroot -p
Enter password:
Grants for user_0d_1@localhost
GRANT SELECT, ..., DROP ROLE ON *.* TO `user_0d_1`@`localhost`
GRANT APPLICATION_PASSWORD_ADMIN,...,XA_RECOVER_ADMIN ON *.* TO `user_0d_1`@`localhost`
Grants for user_0d_2@localhost
GRANT USAGE ON *.* TO `user_0d_2`@`localhost`
GRANT SELECT ON `hbtn_0d_2`.* TO `user_0d_2`@`localhost`
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`

- Directory: `0x0E-SQL_more_queries`
- File: `2-create_read_user.sql`

Done! | Check your code | Get a sandbox | QA Review

3. Always a name

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that creates the table `force_name` on your MySQL server.

- `force_name` description:
  - `id` INT
  - `name` VARCHAR(256) can't be null
- The database name will be passed as an argument of the `mysql` command
- If the table `force_name` already exists, your script should not fail

```
guillaume@ubuntu:~/$ cat 3-force_name.sql | mysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
guillaume@ubuntu:~/$ echo 'INSERT INTO force_name (id, name) VALUES (89, "Best School");' |
mysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
guillaume@ubuntu:~/$ echo 'SELECT * FROM force_name;' | mysql -hlocalhost -uroot -p hbtn_0d_
2
Enter password:
id  name
89  Best School
guillaume@ubuntu:~/$ echo 'INSERT INTO force_name (id) VALUES (333);' | mysql -hlocalhost -u
root -p hbtn_0d_2
Enter password:
ERROR 1364 (HY000) at line 1: Field 'name' doesn't have a default value
guillaume@ubuntu:~/$ echo 'SELECT * FROM force_name;' | mysql -hlocalhost -uroot -p hbtn_0d_
2
Enter password:
id  name
89  Best School
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `3-force_name.sql`

Done! | Check your code | Get a sandbox | QA Review

4. ID can't be null

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that creates the table `id_not_null` on your MySQL server.

- `id_not_null` description:
  - `id` INT with the default value `1`
  - `name` VARCHAR(256)
- The database name will be passed as an argument of the `mysql` command
- If the table `id_not_null` already exists, your script should not fail

```
guillaume@ubuntu:~/$ cat 4-never_empty.sql | mysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
guillaume@ubuntu:~/$ echo 'INSERT INTO id_not_null (id, name) VALUES (89, "Best School");' |
mysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
guillaume@ubuntu:~/$ echo 'SELECT * FROM id_not_null;' | mysql -hlocalhost -uroot -p hbtn_0d
_2
Enter password:
id  name
89  Best School
guillaume@ubuntu:~/$ echo 'INSERT INTO id_not_null (name) VALUES ("Best");' | mysql -hlocalh
ost -uroot -p hbtn_0d_2
Enter password:
guillaume@ubuntu:~/$ echo 'SELECT * FROM id_not_null;' | mysql -hlocalhost -uroot -p hbtn_0d
_2
Enter password:
id  name
89  Best School
1   Best
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `4-never_empty.sql`

Done! | Check your code | Get a sandbox | QA Review

5. Unique ID

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that creates the table `unique_id` on your MySQL server.

- `unique_id` description:
  - `id` INT with the default value `1` and must be unique
  - `name` VARCHAR(256)
- The database name will be passed as an argument of the `mysql` command
- If the table `unique_id` already exists, your script should not fail

```
guillaume@ubuntu:~/$ cat 5-unique_id.sql | mysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
guillaume@ubuntu:~/$ echo 'INSERT INTO unique_id (id, name) VALUES (89, "Best School");' | m
ysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
guillaume@ubuntu:~/$ echo 'SELECT * FROM unique_id;' | mysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
id  name
89  Best School
guillaume@ubuntu:~/$ echo 'INSERT INTO unique_id (id, name) VALUES (89, "Best");' | mysql -h
localhost -uroot -p hbtn_0d_2
Enter password:
ERROR 1062 (23000) at line 1: Duplicate entry '89' for key 'unique_id.id'
guillaume@ubuntu:~/$ echo 'SELECT * FROM unique_id;' | mysql -hlocalhost -uroot -p hbtn_0d_2
Enter password:
id  name
89  Best School
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `5-unique_id.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

6. States table

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that creates the database `hbtn_0d_usa` and the table `states` (in the database `hbtn_0d_usa`) on your MySQL server.

- `states` description:
  - `id` INT unique, auto generated, can't be null and is a primary key
  - `name` VARCHAR(256) can't be null
- If the database `hbtn_0d_usa` already exists, your script should not fail
- If the table `states` already exists, your script should not fail

(7)

```
guillaume@ubuntu:~/$ cat 6-states.sql | mysql -hlocalhost -uroot -p
Enter password:
guillaume@ubuntu:~/$ echo 'INSERT INTO states (name) VALUES ("California"), ("Arizona"), ("T
exas");' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
guillaume@ubuntu:~/$ echo 'SELECT * FROM states;' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
id  name
1   California
2   Arizona
3   Texas
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `6-states.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

7. Cities table

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that creates the database `hbtn_0d_usa` and the table `cities` (in the database `hbtn_0d_usa`) on your MySQL server.

- `cities` description:
  - `id` INT unique, auto generated, can't be null and is a primary key
  - `state_id` INT, can't be null and must be a `FOREIGN KEY` that references to `id` of the `states` table
  - `name` VARCHAR(256) can't be null
- If the database `hbtn_0d_usa` already exists, your script should not fail
- If the table `cities` already exists, your script should not fail

(/)

```
guillaume@ubuntu:~/$ cat 7-cities.sql | mysql -hlocalhost -uroot -p
Enter password:
guillaume@ubuntu:~/$ echo 'INSERT INTO cities (state_id, name) VALUES (1, "San Francisco");'
| mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
guillaume@ubuntu:~/$ echo 'SELECT * FROM cities;' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
id  state_id    name
1   1   San Francisco
guillaume@ubuntu:~/$ echo 'INSERT INTO cities (state_id, name) VALUES (10, "Paris");' | mysq
l -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
ERROR 1452 (23000) at line 1: Cannot add or update a child row: a foreign key constraint fai
ls (`hbtn_0d_usa`.`cities`, CONSTRAINT `cities_ibfk_1` FOREIGN KEY (`state_id`) REFERENCES `
states` (`id`))
guillaume@ubuntu:~/$ echo 'SELECT * FROM cities;' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
id  state_id    name
1   1   San Francisco
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `7-cities.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

8. Cities of California

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that lists all the cities of California that can be found in the database `hbtn_0d_usa` .

- The `states` table contains only one record where `name = California` (but the `id` can be different, as per the example)
- Results must be sorted in ascending order by `cities.id`
- You are not allowed to use the `JOIN` keyword
- The database name will be passed as an argument of the `mysql` command

(/)

```
guillaume@ubuntu:~/$ echo 'SELECT * FROM states;' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
id  name
1   California
2   Arizona
3   Texas
4   Utah
guillaume@ubuntu:~/$ echo 'SELECT * FROM cities;' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
id  state_id    name
1   1   San Francisco
2   1   San Jose
4   2   Page
6   3   Paris
7   3   Houston
8   3   Dallas
guillaume@ubuntu:~/$ cat 8-cities_of_california_subquery.sql | mysql -hlocalhost -uroot -p h
btn_0d_usa
Enter password:
id  name
1   San Francisco
2   San Jose
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `8-cities_of_california_subquery.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

9. Cities by States

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Write a script that lists all cities contained in the database `hbtn_0d_usa` .

- Each record should display: `cities.id - cities.name - states.name`
- Results must be sorted in ascending order by `cities.id`
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

(7)

```
guillaume@ubuntu:~/$ echo 'SELECT * FROM states;' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
id  name
1   California
2   Arizona
3   Texas
4   Utah
guillaume@ubuntu:~/$ echo 'SELECT * FROM cities;' | mysql -hlocalhost -uroot -p hbtn_0d_usa
Enter password:
id  state_id    name
1   1   San Francisco
2   1   San Jose
4   2   Page
6   3   Paris
7   3   Houston
8   3   Dallas
guillaume@ubuntu:~/$ cat 9-cities_by_state_join.sql | mysql -hlocalhost -uroot -p hbtn_0d_us
a
Enter password:
id  name    name
1   San Francisco   California
2   San Jose    California
4   Page    Arizona
6   Paris   Texas
7   Houston Texas
8   Dallas  Texas
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `9-cities_by_state_join.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

10. Genre ID by show

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql)

Write a script that lists all shows contained in `hbtn_0d_tvshows` that have at least one genre linked.

- Each record should display: `tv_shows.title - tv_show_genres.genre_id`
- Results must be sorted in ascending order by `tv_shows.title` and `tv_show_genres.genre_id`

- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 10-genre_id_by_show.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvsho
ws
Enter password:
title   genre_id
Breaking Bad    1
Breaking Bad    6
Breaking Bad    7
Breaking Bad    8
Dexter  1
Dexter  2
Dexter  6
Dexter  7
Dexter  8
Game of Thrones 1
Game of Thrones 3
Game of Thrones 4
House   1
House   2
New Girl    5
Silicon Valley  5
The Big Bang Theory 5
The Last Man on Earth   1
The Last Man on Earth   5
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `10-genre_id_by_show.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

11. Genre ID for all shows

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump of `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `10-genre_id_by_show.sql`)

Write a script that lists all shows contained in the database `hbtn_0d_tvshows` .

- Each record should display: `tv_shows.title - tv_show_genres.genre_id`
- Results must be sorted in ascending order by `tv_shows.title` and `tv_show_genres.genre_id`
- If a show doesn't have a genre, display `NULL`

- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 11-genre_id_all_shows.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvs
hows
Enter password:
title   genre_id
Better Call Saul    NULL
Breaking Bad    1
Breaking Bad    6
Breaking Bad    7
Breaking Bad    8
Dexter  1
Dexter  2
Dexter  6
Dexter  7
Dexter  8
Game of Thrones 1
Game of Thrones 3
Game of Thrones 4
Homeland    NULL
House   1
House   2
New Girl    5
Silicon Valley  5
The Big Bang Theory 5
The Last Man on Earth   1
The Last Man on Earth   5
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `11-genre_id_all_shows.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

12. No genre

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `11-genre_id_all_shows.sql`)

Write a script that lists all shows contained in `hbtn_0d_tvshows` without a genre linked.

- Each record should display: `tv_shows.title - tv_show_genres.genre_id`

- Results must be sorted in ascending order by `tv_shows.title` and `tv_show_genres.genre_id`
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 12-no_genre.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
Enter password:
title   genre_id
Better Call Saul    NULL
Homeland    NULL
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `12-no_genre.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

13. Number of shows by genre

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `12-no_genre.sql`)

Write a script that lists all genres from `hbtn_0d_tvshows` and displays the number of shows linked to each.

- Each record should display: `<TV Show genre> - <Number of shows linked to this genre>`
- First column must be called `genre`
- Second column must be called `number_of_shows`
- Don't display a genre that doesn't have any shows linked
- Results must be sorted in descending order by the number of shows linked
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 13-count_shows_by_genre.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
Enter password:
genre   number_of_shows
Drama   5
Comedy  4
Mystery 2
Crime   2
Suspense    2
Thriller    2
Adventure   1
Fantasy 1
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `13-count_shows_by_genre.sql`

Done! | Check your code | Get a sandbox | QA Review

14. My genres

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `13-count_shows_by_genre.sql`)

Write a script that uses the `hbtn_0d_tvshows` database to lists all genres of the show `Dexter` .

- The `tv_shows` table contains only one record where `title = Dexter` (but the `id` can be different)
- Each record should display: `tv_genres.name`
- Results must be sorted in ascending order by the genre name
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 14-my_genres.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
Enter password:
name
Crime
Drama
Mystery
Suspense
Thriller
guillaume@ubuntu:~/$
```

(/)

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `14-my_genres.sql`

Done! | Check your code | Get a sandbox | QA Review

15. Only Comedy

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `14-my_genres.sql`)

Write a script that lists all Comedy shows in the database `hbtn_0d_tvshows` .

- The `tv_genres` table contains only one record where `name = Comedy` (but the `id` can be different)
- Each record should display: `tv_shows.title`
- Results must be sorted in ascending order by the show title
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 15-comedy_only.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
Enter password:
title
New Girl
Silicon Valley
The Big Bang Theory
The Last Man on Earth
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `15-comedy_only.sql`

Done! | Check your code | Get a sandbox | QA Review

16. List shows and genres

mandatory

Score: 100.0% (*Checks completed: 100.0%*)

(/)

import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `15-comedy_only.sql`)

Write a script that lists all shows, and all genres linked to that show, from the database `hbtn_0d_tvshows` .

- If a show doesn't have a genre, display `NULL` in the genre column
- Each record should display: `tv_shows.title - tv_genres.name`
- Results must be sorted in ascending order by the show title and genre name
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 16-shows_by_genre.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
Enter password:
title   name
Better Call Saul    NULL
Breaking Bad    Crime
Breaking Bad    Drama
Breaking Bad    Suspense
Breaking Bad    Thriller
Dexter  Crime
Dexter  Drama
Dexter  Mystery
Dexter  Suspense
Dexter  Thriller
Game of Thrones Adventure
Game of Thrones Drama
Game of Thrones Fantasy
Homeland    NULL
House   Drama
House   Mystery
New Girl    Comedy
Silicon Valley  Comedy
The Big Bang Theory Comedy
The Last Man on Earth   Comedy
The Last Man on Earth   Drama
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `16-shows_by_genre.sql`

17. Not my genre

#advanced

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `16-shows_by_genre.sql`)

Write a script that uses the `hbtn_0d_tvshows` database to list all genres not linked to the show `Dexter`

- The `tv_shows` table contains only one record where `title = Dexter` (but the `id` can be different)
- Each record should display: `tv_genres.name`
- Results must be sorted in ascending order by the genre name
- You can use a maximum of two `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

```
guillaume@ubuntu:~/$ cat 100-not_my_genres.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
Enter password:
name
Adventure
Comedy
Fantasy
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `100-not_my_genres.sql`

Done! | Check your code | Get a sandbox | QA Review

18. No Comedy tonight!

#advanced

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows.sql) (same as `100-not_my_genres.sql`)

Write a script that lists all shows without the genre `Comedy` in the database `hbtn_0d_tvshows` .

- The `tv_genres` table contains only one record where `name = Comedy` (but the `id` can be different)
- Each record should display: `tv_shows.title`
- Results must be sorted in ascending order by the show title
- You can use a maximum of two `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

(/)

```
guillaume@ubuntu:~/$ cat 101-not_a_comedy.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
Enter password:
title
Better Call Saul
Breaking Bad
Dexter
Game of Thrones
Homeland
House
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `101-not_a_comedy.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

19. Rotten tomatoes

#advanced

Score: 100.0% (*Checks completed: 100.0%*)

Import the database `hbtn_0d_tvshows_rate` dump to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows_rate.sql)

Write a script that lists all shows from `hbtn_0d_tvshows_rate` by their rating.

- Each record should display: `tv_shows.title - rating sum`
- Results must be sorted in descending order by the rating
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

(7)

```
guillaume@ubuntu:~/$ cat 102-rating_shows.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows_
rate
Enter password:
title   rating
Better Call Saul    163
Homeland    145
Silicon Valley  82
Game of Thrones 79
Dexter  24
House   21
Breaking Bad    16
The Last Man on Earth   10
The Big Bang Theory 0
New Girl    0
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `102-rating_shows.sql`

☑ Done! | Check your code | >_ Get a sandbox | QA Review

20. Best genre

#advanced

Score: 100.0% (*Checks completed: 100.0%*)

Import the database dump from `hbtn_0d_tvshows_rate` to your MySQL server: download (https://s3.amazonaws.com/intranet-projects-files/holbertonschool-higher-level_programming+/274/hbtn_0d_tvshows_rate.sql) (same as `102-rating_shows.sql`)

Write a script that lists all genres in the database `hbtn_0d_tvshows_rate` by their rating.

- Each record should display: `tv_genres.name - rating sum`
- Results must be sorted in descending order by their rating
- You can use only one `SELECT` statement
- The database name will be passed as an argument of the `mysql` command

(7)

```
guillaume@ubuntu:~/$ cat 103-rating_genres.sql | mysql -hlocalhost -uroot -p hbtn_0d_tvshows
_rate
Enter password:
name    rating
Drama   150
Comedy  92
Adventure   79
Fantasy 79
Mystery 45
Crime   40
Suspense    40
Thriller    40
guillaume@ubuntu:~/$
```

Repo:

- GitHub repository: `alx-higher_level_programming`
- Directory: `0x0E-SQL_more_queries`
- File: `103-rating_genres.sql`

Done! | Check your code | Get a sandbox | QA Review