COMP-306 HOMEWORK-2

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
barissss-mbp:~ barissss$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, 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>
```

Connecting to the mysql client in the Macosx terminal

Q1:

In order to import the “world.sql file” inside the mysql client, I have used the SOURCE command with the absolute path of the “world.sql” file right after the command.

SOURCE file_path

The file path I used here is: **/Users/barissss/Desktop/world.sql**

```
mysql> use world
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

The SQL command for using the database called "world"

```
mysql> SHOW TABLES;
+-----------------+
| Tables_in_world |
+-----------------+
| city            |
| country         |
| countrylanguage |
+-----------------+
3 rows in set (0,01 sec)
```

Screenshot taken from terminal after I run SHOW TABLES command inside the mysql

Q2:

Part-A:

```
mysql> SELECT Name,GNP
    -> FROM country
    -> WHERE Continent='Europe' AND indepYear>1930;
+-------------------------+------------+
| Name                    | GNP        |
+-------------------------+------------+
| Bosnia and Herzegovina |    2841.00 |
| Belarus                 |   13714.00 |
| Czech Republic          |   55017.00 |
| Germany                 | 2133367.00 |
| Estonia                 |    5328.00 |
| Croatia                 |   20208.00 |
| Iceland                 |    8255.00 |
| Lithuania               |   10692.00 |
| Latvia                  |    6398.00 |
| Moldova                 |    1579.00 |
| Macedonia               |    1694.00 |
| Malta                   |    3512.00 |
| Russian Federation      |  276608.00 |
| Slovakia                |   20594.00 |
| Slovenia                |   19756.00 |
| Ukraine                 |   42168.00 |
+-------------------------+------------+
16 rows in set (0,01 sec)
```

The output of the SELECT-FROM-WHERE command that I have written for the part-A of Q2 (Along with the command itself).

Part-B:

```
mysql> SELECT ct.CountryCode,cnt.Name,ct.Name,ct.Population
    -> FROM city ct,country cnt
    -> WHERE ct.CountryCode=cnt.Code AND cnt.LifeExpectancy>75 AND cnt.Population<1000000;
+-------------+--------------------------+-----------------------------------+------------+
| CountryCode | Name                     | Name                              | Population |
+-------------+--------------------------+-----------------------------------+------------+
| ABW         | Aruba                    | Oranjestad                        |      29034 |
| AIA         | Anguilla                 | South Hill                        |        961 |
| AIA         | Anguilla                 | The Valley                        |        595 |
| AND         | Andorra                  | Andorra la Vella                  |      21189 |
| ASM         | American Samoa           | Tafuna                            |       5200 |
| ASM         | American Samoa           | Fagatogo                          |       2323 |
| BMU         | Bermuda                  | Saint George                      |       1800 |
| BMU         | Bermuda                  | Hamilton                          |       1200 |
| CYM         | Cayman Islands           | George Town                       |      19600 |
| CYP         | Cyprus                   | Nicosia                           |     195000 |
| CYP         | Cyprus                   | Limassol                          |     154400 |
| FRO         | Faroe Islands            | Tórshavn                          |      14542 |
| GIB         | Gibraltar                | Gibraltar                         |      27025 |
| GLP         | Guadeloupe               | Les Abymes                        |      62947 |
| GLP         | Guadeloupe               | Basse-Terre                       |      12433 |
| GUF         | French Guiana            | Cayenne                           |      50699 |
| GUM         | Guam                     | Tamuning                          |       9500 |
| GUM         | Guam                     | Agaña                             |       1139 |
| ISL         | Iceland                  | Reykjavík                         |     109184 |
| LIE         | Liechtenstein            | Schaan                            |       5346 |
| LIE         | Liechtenstein            | Vaduz                             |       5043 |
| LUX         | Luxembourg               | Luxembourg [Luxemburg/Lëtzebuerg] |      80700 |
| MAC         | Macao                    | Macao                             |     437500 |
| MCO         | Monaco                   | Monte-Carlo                       |      13154 |
| MCO         | Monaco                   | Monaco-Ville                      |       1234 |
| MLT         | Malta                    | Birkirkara                        |      21445 |
| MLT         | Malta                    | Valletta                          |       7073 |
| MNP         | Northern Mariana Islands | Garapan                           |       9200 |
| MSR         | Montserrat               | Plymouth                          |       2000 |
| MTQ         | Martinique               | Fort-de-France                    |      94050 |
| SHN         | Saint Helena             | Jamestown                         |       1500 |
| SMR         | San Marino               | Serravalle                        |       4802 |
| SMR         | San Marino               | San Marino                        |       2294 |
| SPM         | Saint Pierre and Miquelon | Saint-Pierre                      |       5808 |
| VGB         | Virgin Islands, British  | Road Town                         |       8000 |
| VIR         | Virgin Islands, U.S.     | Charlotte Amalie                  |      13000 |
+-------------+--------------------------+-----------------------------------+------------+
36 rows in set (0,00 sec)
```

The output of the SELECT-FROM-WHERE command that I have written for the part-B of Q2 (Along with the sql command itself).

Part-C:

```
mysql> SELECT DISTINCT cl.Language as countrylanguage
    -> FROM countrylanguage cl,country cnt
    -> WHERE cl.CountryCode=cnt.Code AND cnt.SurfaceArea>1000000.00 AND cl.Percentage>50.0;
+-----------------+
| countrylanguage |
+-----------------+
| Spanish         |
| English         |
| Portuguese      |
| Chinese         |
| Arabic          |
| Greenlandic     |
| Mongolian       |
| Hassaniya       |
| Hausa           |
| Russian         |
+-----------------+
10 rows in set (0,00 sec)
```

The output of the SELECT-FROM-WHERE command that I have written for the part-C of Q2 (along with the sql command itself)

Part-D:

```
mysql> SELECT ct.CountryCode,ct.Name
    -> FROM city ct,countrylanguage clan,country cnt
    -> WHERE clan.CountryCode=cnt.Code AND ct.CountryCode=cnt.Code AND ct.Population<90000 AND clan.Percentage>1.0 AND clan.Language='German';
+-------------+-----------------------------------+
| CountryCode | Name                              |
+-------------+-----------------------------------+
| CAN         | Kelowna                           |
| CAN         | Barrie                            |
| DEU         | Velbert                           |
| DEU         | Esslingen am Neckar               |
| LIE         | Schaan                            |
| LIE         | Vaduz                             |
| LUX         | Luxembourg [Luxemburg/Lëtzebuerg] |
+-------------+-----------------------------------+
7 rows in set (0,00 sec)
```

The SELECT-FROM-WHERE command that I have written for the part-D of Q2 & the output of this command.

Q3:

```
mysql> create table students(
    -> name varchar(55) not null,
    -> university varchar(60) not null,
    -> s_id varchar(25) unique,
    -> primary key(s_id)
    -> );
Query OK, 0 rows affected (0,03 sec)
```

SQL Command I have used for creating a table named 'students' and also the output of the command

```
mysql> show tables;
+-----------------+
| Tables_in_world |
+-----------------+
| city            |
| country         |
| countrylanguage |
| students        |
+-----------------+
4 rows in set (0,00 sec)
```

SQL Command I have used for displaying tables after I have created the table named 'students' and also the output of the 'show tables' command

```
mysql> INSERT INTO students(name, university,s_id)
    -> VALUES('Barış Kaplan','KU',1);
Query OK, 1 row affected (0,02 sec)
```

SQL command I have used to enter myself as a record to the students table and also the output of the command

```
mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('John Stark','KU',2);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('Yasemin A','SU',3);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('Can Y','BOUN',4);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('Efe G','SU',5);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('Armagan T','SU',6);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('Cem K','KU',7);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('Yang W','KU',8);
Query OK, 1 row affected (0,00 sec)

mysql> INSERT INTO students(name,university,s_id)
    -> VALUES('Uras C','BOUN',9);
Query OK, 1 row affected (0,00 sec)
```

Addition of the other provided records to the table called students & the outputs of the additions of the records

```
mysql> UPDATE students
    -> SET university='KU'
    -> WHERE s_id=5;
Query OK, 1 row affected (0,04 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

SQL Command I have used for updating the university of 'Efe G' to 'KU' , AND the output of this SQL command

```
mysql> DELETE FROM students
    -> WHERE name='John Stark';
Query OK, 1 row affected (0,00 sec)
```

SQL Command I have used for deleting the student which has the name 'John Stark' and the output of this command

Q4:

Part-A:

```
mysql> create table visit(
    -> s_id varchar(25) NOT NULL,
    -> ID int NOT NULL,
    -> primary key(s_id,ID),
    -> foreign key(s_id)
    -> REFERENCES students(s_id)
    -> ON UPDATE CASCADE ON DELETE RESTRICT,
    -> foreign key(ID)
    -> REFERENCES city(ID)
    -> ON UPDATE CASCADE ON DELETE RESTRICT
    -> );
Query OK, 0 rows affected (0,04 sec)
```

SQL command I have used for creating a table named 'visit', and the output of this SQL command

```
mysql> USE world
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

SQL command for changing the Database I use & also the output of this SQL command

Part-B:

```
mysql> SET GLOBAL local_infile=1;
Query OK, 0 rows affected (0,00 sec)

mysql> exit
Bye
barissss-MacBook-Pro:~ barissss$ mysql --local-infile=1 -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 8.0.28 MySQL Community Server - GPL

Copyright (c) 2000, 2022, 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.
```

Preparation to load data

```
mysql> load data local infile "/Users/barissss/Desktop/visit.txt"
    -> into table visit;
Query OK, 20 rows affected (0,01 sec)
Records: 20  Deleted: 0  Skipped: 0  Warnings: 0
```

SQL Command for loading the data inside the "visit.txt" file to the table called "visit" & also the output of this SQL command

Part-C:

```
mysql> SELECT cty.ID,cty.Name
    -> FROM city cty,visit vst,students std
    -> WHERE vst.ID=cty.ID AND vst.s_id=std.s_id
    -> GROUP BY cty.ID
    -> HAVING COUNT(*) = (SELECT COUNT(DISTINCT std.s_id) FROM students std);
+------+----------+
| ID   | Name     |
+------+----------+
| 3357 | Istanbul |
+------+----------+
1 row in set (0,00 sec)
```

The SQL command I have used for the Part-C of Q4 & The output of this SQL command

Part-F:

```
mysql> SELECT std.s_id,std.name
    -> FROM students std,visit vs,country cnt,city ct
    -> WHERE cnt.Continent='Africa' AND vs.s_id=std.s_id AND vs.ID=ct.ID and ct.CountryCode=cnt.Code;
+------+--------+
| s_id | name   |
+------+--------+
| 5    | Efe G  |
| 9    | Uras C |
+------+--------+
2 rows in set (0,00 sec)

mysql>
```

The SQL command I have used for Part-F of Q4 & The output of this SQL command

Part-G:

```
mysql> SELECT DISTINCT cnt.Continent
    -> FROM country cnt
    -> WHERE cnt.Continent NOT IN (SELECT DISTINCT cnt.Continent FROM students std,country cnt,visit vs,city ct WHERE vs.s_id=std.s_id AND vs.ID=ct.ID AND ct.CountryCode=cnt.Code);
+---------------+
| Continent     |
+---------------+
| South America |
| Oceania       |
| Antarctica    |
+---------------+
3 rows in set (0,00 sec)
```

The SQL command I have used for Part-G of Q4 & The output of this SQL command

Part-D:

```
mysql> SELECT std.s_id,std.name
    -> FROM students std
    -> WHERE std.s_id NOT IN (SELECT std.s_id FROM students std,visit vs,city ct WHERE vs.ID=ct.ID AND vs.s_id=std.s_id AND vs.s_id=std.s_id AND ct.Name='Paris') AND std.s_id IN (SELECT std.s_id FROM
students std,visit vs,city ct WHERE vs.ID=ct.ID AND vs.s_id=std.s_id AND ct.Name='Amsterdam');
+------+-----------+
| s_id | name      |
+------+-----------+
| 6    | Armagan T |
+------+-----------+
1 row in set (0,02 sec)

mysql>
```

The SQL Command I used in Q4 Part-D, and the output of this SQL command

The sql command I used in part-D of Question-4:

```
SELECT std.s_id,std.name
FROM students std
WHERE std.s_id NOT IN (SELECT std.s_id FROM students std,visit
vs,city ct WHERE vs.ID=ct.ID AND vs.s_id=std.s_id AND
ct.Name='Paris') AND std.s_id IN (SELECT std.s_id FROM students
std,visit vs,city ct WHERE vs.ID=ct.ID AND vs.s_id=std.s_id AND
ct.Name='Amsterdam');
```

Part-E:

```
mysql> SELECT std.s_id,std.name
    -> FROM students std
    -> WHERE std.s_id NOT IN (SELECT std.s_id FROM students std,visit vs,country cnt,city ct WHERE vs.s_id=std.s_id AND vs.ID=ct.ID AND cnt.name='Russian Federation' AND ct.Co
untryCode=cnt.Code);
+------+----------------+
| s_id | name           |
+------+----------------+
| 1    | Barış Kaplan   |
| 3    | Yasemin A      |
| 4    | Can Y          |
| 5    | Efe G          |
| 8    | Yang W         |
+------+----------------+
5 rows in set (0,00 sec)
```

The SQL Command I used in Part-E of Question4, and also the output of this SQL command

The sql command I used in part-E of Question-4:

```
SELECT std.s_id,std.name
FROM students std
WHERE std.s_id NOT IN (SELECT std.s_id FROM students std,visit
vs,country cnt,city ct WHERE vs.s_id=std.s_id AND vs.ID=ct.ID AND
cnt.name='Russian Federation' AND ct.CountryCode=cnt.Code);
```

Q5:

Initially, we should exit from the MySQL client. You can see the command to exit MySQL client in the below figure.

```
mysql> exit
Bye
```

SQL command to exit the MySQL Client & the output of this command

For exporting the world database to the "world-export.sql" file, I have used the following command:

mysqldump -u root -p world > /Users/barissss/Desktop/world-export.sql

The screenshot after the execution of the above command:

```
barissss-MacBook-Pro:~ barissss$ mysqldump -u root -p world > /Users/barissss/Desktop/world-export.sql
Enter password:
barissss-MacBook-Pro:~ barissss$
```

MY NAME-SURNAME: BARIŞ KAPLAN
MY KU ID NUMBER: 0069054
MY KU EMAIL ADDRESS: bkaplan18@ku.edu.tr
HOMEWORK #: HW-2