Contents

MySQL Questions

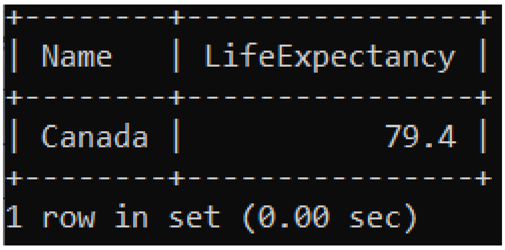
Import the *world* database from *world.sql* to MySQL and write queries to satisfy the following.

Write only the exact MySQL command for each question into the appropriate file.

Question A (MySQLQA.txt)

Show the *Name* and *LifeExpectancy* of all countries in *"North America"* where the country's *LifeExpectancy* is the maximum *LifeExpectancy* for countries in "*North America*".

The results should be sorted alphabetically by name.



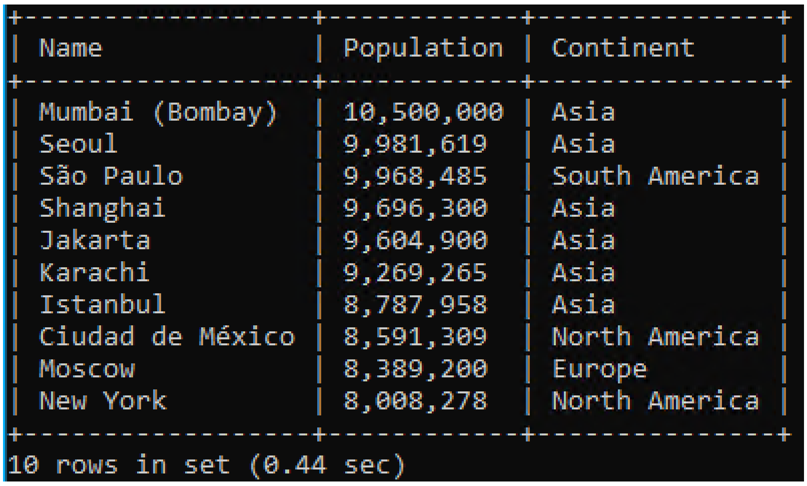
```
+--------+----------------+
| Name   | LifeExpectancy |
+--------+----------------+
| Canada |           79.4 |
+--------+----------------+
1 row in set (0.00 sec)
```

Figure 1 Example of output required for this question

Question B (MySQLQB.txt)

Show the Name, Population (with commas) and Continent of all cities with populations greater than 8,000,000.

The results should be sorted from largest to smallest population, and within that alphabetically by name.



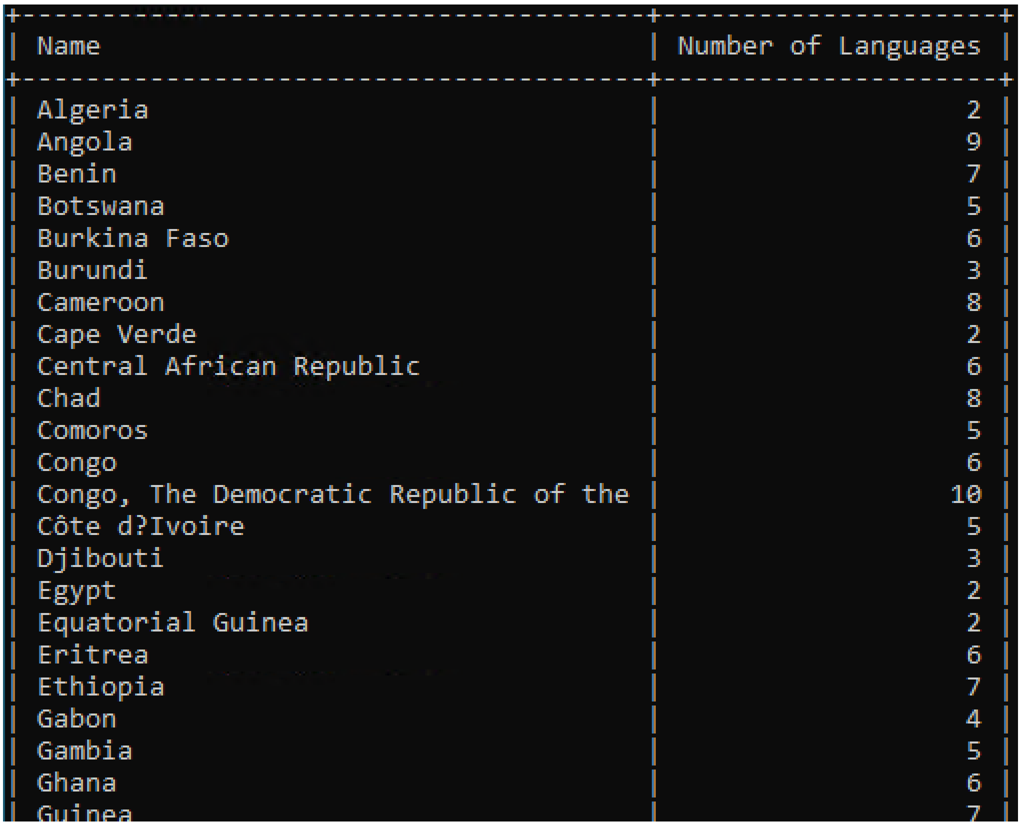
```
+------------------+------------+---------------+
| Name             | Population | Continent     |
+------------------+------------+---------------+
| Mumbai (Bombay)  | 10,500,000 | Asia          |
| Seoul            | 9,981,619  | Asia          |
| São Paulo        | 9,968,485  | South America |
| Shanghai         | 9,696,300  | Asia          |
| Jakarta          | 9,604,900  | Asia          |
| Karachi          | 9,269,265  | Asia          |
| Istanbul         | 8,787,958  | Asia          |
| Ciudad de México | 8,591,309  | North America |
| Moscow           | 8,389,200  | Europe        |
| New York         | 8,008,278  | North America |
+------------------+------------+---------------+
10 rows in set (0.44 sec)
```

Figure 2 Example of output required for this question.

Question E (MySQLQE.txt)

Show the Name, and number of languages (as Number of Languages) spoken in that country for countries in “Africa”.

The results should be sorted alphabetically by Name.



```
+---------------------------------------+---------------------+
| Name                                  | Number of Languages |
+---------------------------------------+---------------------+
| Algeria                               |                   2 |
| Angola                                |                   9 |
| Benin                                 |                   7 |
| Botswana                              |                   5 |
| Burkina Faso                          |                   6 |
| Burundi                               |                   3 |
| Cameroon                              |                   8 |
| Cape Verde                            |                   2 |
| Central African Republic              |                   6 |
| Chad                                  |                   8 |
| Comoros                               |                   5 |
| Congo                                 |                   6 |
| Congo, The Democratic Republic of the |                  10 |
| Côte d?Ivoire                         |                   5 |
| Djibouti                              |                   3 |
| Egypt                                 |                   2 |
| Equatorial Guinea                     |                   2 |
| Eritrea                               |                   6 |
| Ethiopia                              |                   7 |
| Gabon                                 |                   4 |
| Gambia                                |                   5 |
| Ghana                                 |                   6 |
| Guinea                                |                   7 |
```

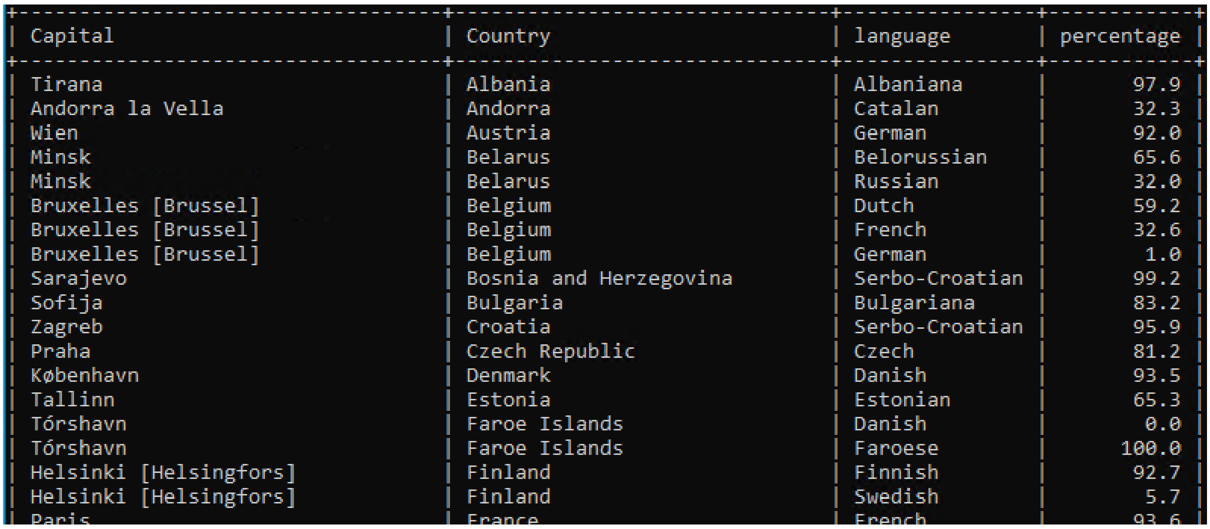
Figure 3 Example of output required for this question.

Question F (MySQLQF.txt)

For each country in “Europe” show:

- The name of the capital city (as *Capital*)
- The name of the country (as *Country*)
- The official language(s)
- The percentage of people who speak the official language(s)

The results should be sorted alphabetical by country name, within that alphabetically by capital name, and within that alphabetically by language.



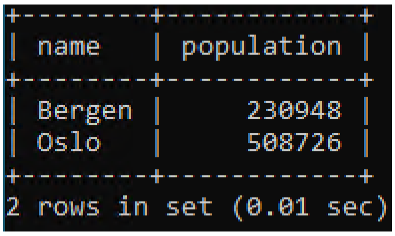
```
+-----------------------------------+-------------------------------+----------------+------------+
| Capital                           | Country                       | language       | percentage |
+-----------------------------------+-------------------------------+----------------+------------+
| Tirana                            | Albania                       | Albaniana      |       97.9 |
| Andorra la Vella                  | Andorra                       | Catalan        |       32.3 |
| Wien                              | Austria                       | German         |       92.0 |
| Minsk                             | Belarus                       | Belorussian    |       65.6 |
| Minsk                             | Belarus                       | Russian        |       32.0 |
| Bruxelles [Brussel]               | Belgium                       | Dutch          |       59.2 |
| Bruxelles [Brussel]               | Belgium                       | French         |       32.6 |
| Bruxelles [Brussel]               | Belgium                       | German         |        1.0 |
| Sarajevo                          | Bosnia and Herzegovina        | Serbo-Croatian |       99.2 |
| Sofija                            | Bulgaria                      | Bulgariana     |       83.2 |
| Zagreb                            | Croatia                       | Serbo-Croatian |       95.9 |
| Praha                             | Czech Republic                | Czech          |       81.2 |
| København                         | Denmark                       | Danish         |       93.5 |
| Tallinn                           | Estonia                       | Estonian       |       65.3 |
| Tórshavn                          | Faroe Islands                 | Danish         |        0.0 |
| Tórshavn                          | Faroe Islands                 | Faroese        |      100.0 |
| Helsinki [Helsingfors]            | Finland                       | Finnish        |       92.7 |
| Helsinki [Helsingfors]            | Finland                       | Swedish        |        5.7 |
| Paris                             | France                        | French         |       93.6 |
```

Figure 4 Example of output required for this question.

Question I (MySQLQI.txt)

Show the Name and Population of cities whose population is greater than the average population of cities where the HeadOfState is “Harald V”.

The results should be sorted alphabetically by Name, and within that from smallest to largest population.

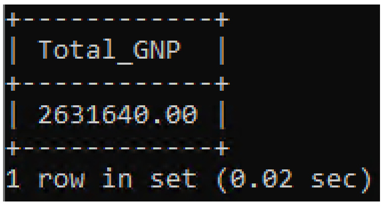


```
+--------+------------+
| name   | population |
+--------+------------+
| Bergen |     230948 |
| Oslo   |     508726 |
+--------+------------+
2 rows in set (0.01 sec)
```

Figure 5 Example of output required for this question.

Question J (MySQLQJ.txt)

Show the total GNP (as *Total_GNP*) of countries visited by the person with the personID = 1.



```
+------------+
| Total_GNP  |
+------------+
| 2631640.00 |
+------------+
1 row in set (0.02 sec)
```

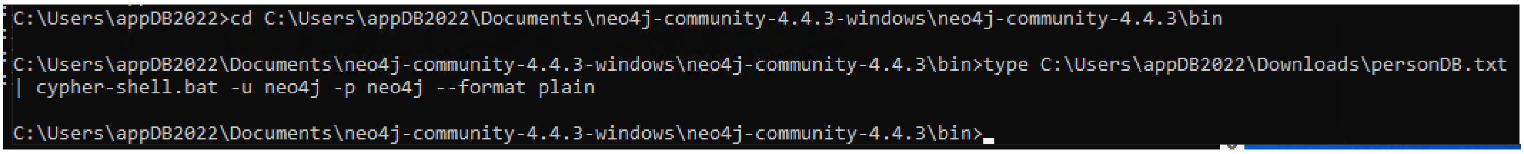
Figure 6 Example of output required for this question.

Neo4j Questions

Import *personDB.txt* into Neo4j as follows:

```
cd C:\Users\appDB2022\Documents\neo4j-community-4.4.3-windows\neo4j-community-4.4.3\bin
```

```
type path_to_personDB.txt | cypher-shell.bat -u neo4j -p neo4j --format plain
```



```
C:\Users\appDB2022>cd C:\Users\appDB2022\Documents\neo4j-community-4.4.3-windows\neo4j-community-4.4.3\bin

C:\Users\appDB2022\Documents\neo4j-community-4.4.3-windows\neo4j-community-4.4.3\bin>type C:\Users\appDB2022\Downloads\personDB.txt
| cypher-shell.bat -u neo4j -p neo4j --format plain

C:\Users\appDB2022\Documents\neo4j-community-4.4.3-windows\neo4j-community-4.4.3\bin>_
```

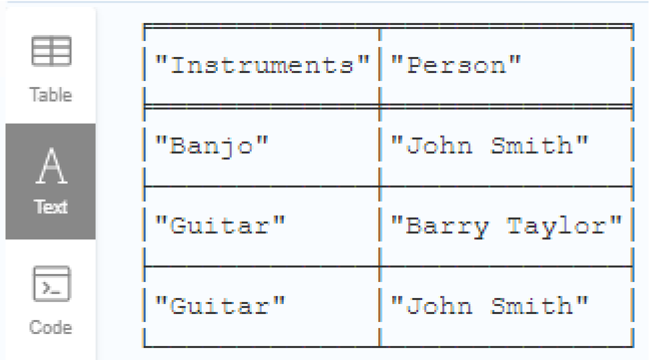
Figure 7 Import Neo4j database

Write only the exact MongoDB command for each question into the appropriate file.

Question A (Neo4jQA.txt)

Return the names of instruments (as Instruments) people play, and the names of people (as Person) who play those instruments, only for people who play Midfield position in either Football or Soccer.

Results should be in alphabetical instrument name, and within that alphabetically by person name.



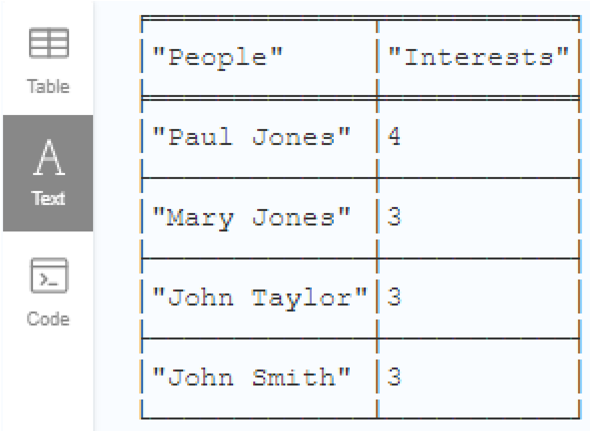
```
╒═════════════╤═══════════════╕
│"Instruments"│"Person"       │
╞═════════════╪═══════════════╡
│"Banjo"      │"John Smith"   │
├─────────────┼───────────────┤
│"Guitar"     │"Barry Taylor" │
├─────────────┼───────────────┤
│"Guitar"     │"John Smith"   │
└─────────────┴───────────────┘
```

Figure 8 Example of output required for this question.

Question B (Neo4jQB.txt)

Return the names of people (as People), and the number of hobbies (HAS_HOBBY), sports (PLAYS), and instruments (PLAYS) they have (as Interests), but only if they have more hobbies (HAS_HOBBY), sports (PLAYS), and instruments (PLAYS), than Barry Taylor.

Results should be in descending order of number of hobbies, and within that alphabetically by person name.

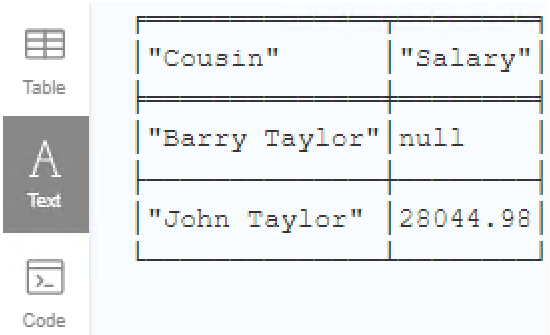


```
╒══════════════╤════════════╕
│"People"      │"Interests" │
╞══════════════╪════════════╡
│"Paul Jones"  │4           │
├──────────────┼────────────┤
│"Mary Jones"  │3           │
├──────────────┼────────────┤
│"John Taylor" │3           │
├──────────────┼────────────┤
│"John Smith"  │3           │
└──────────────┴────────────┘
```

Figure 9 Example of output required for this question.

Question E (Neo4jQE.txt)

Return the names (as Cousin) and salaries (as Salary) of all Tom Smith's first cousins in alphabetical name order, followed by ascending salary order.

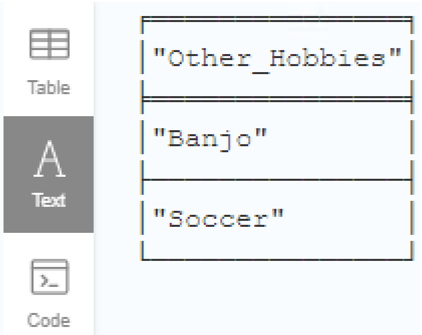


```
╒══════════════╤══════════╕
│"Cousin"      │"Salary"  │
╞══════════════╪══════════╡
│"Barry Taylor"│null      │
├──────────────┼──────────┤
│"John Taylor" │28044.98  │
└──────────────┴──────────┘
```

Figure 10 Example of output required for this question.

Question F (Neo4jQF.txt)

Return the unique names (in alphabetical order as Other_Hobbies) of other Instruments or Sports that people who play the “Guitar” also PLAY.

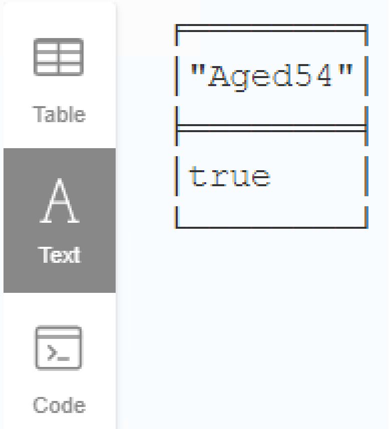


```
╒═══════════════╕
│"Other_Hobbies"│
╞═══════════════╡
│"Banjo"        │
├───────────────┤
│"Soccer"       │
└───────────────┘
```

Figure 11 Example of output required for this question.

Question I (Neo4jQI.txt)

Return true (as Aged54) if any person’s age is 54, otherwise return false.



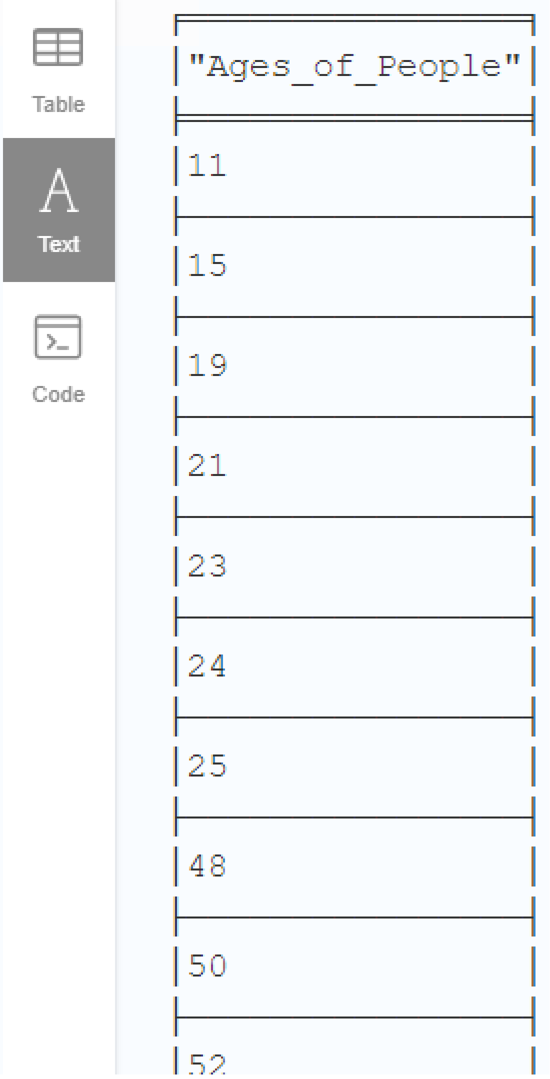
```
╒════════╕
│"Aged54"│
╞════════╡
│true    │
└────────┘
```

Figure 12 Example of output required for this question.

Question J (Neo4jQJ.txt)

Return the ages of people (in ascending order as Ages_of_People).

If more than one person shares the same age, the age should only be returned once.



```
╒════════════════╕
│"Ages_of_People"│
╞════════════════╡
│11              │
├────────────────┤
│15              │
├────────────────┤
│19              │
├────────────────┤
│21              │
├────────────────┤
│23              │
├────────────────┤
│24              │
├────────────────┤
│25              │
├────────────────┤
│48              │
├────────────────┤
│50              │
├────────────────┤
│52              │
```

Figure 13 Example of output required for this question.