Contents

MySQL Questions

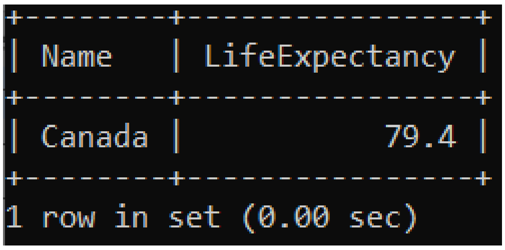
Import the *world* database from *world.sql* to MySQL and write queries to satisfy the following.

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.



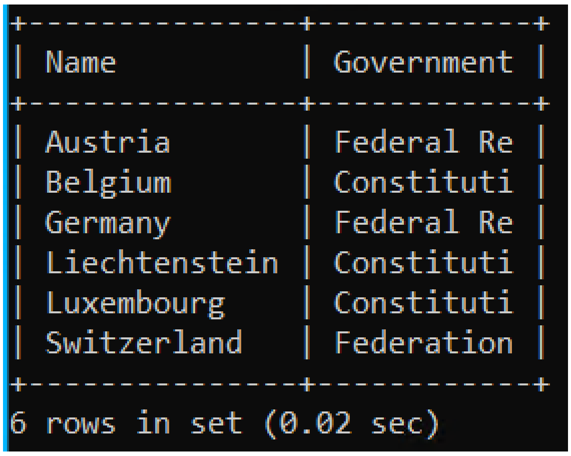
```
+--------+----------------+
| Name   | LifeExpectancy |
+--------+----------------+
| Canada |           79.4 |
+--------+----------------+
1 row in set (0.00 sec)
```

Figure 1 Example of output required for this question

Question C (MySQLQC.txt)

Show the *Name* and first 10 characters of GovernmentForm (as *Government*) of countries where “German” is an official language.

The results should be sorted alphabetically by Name.



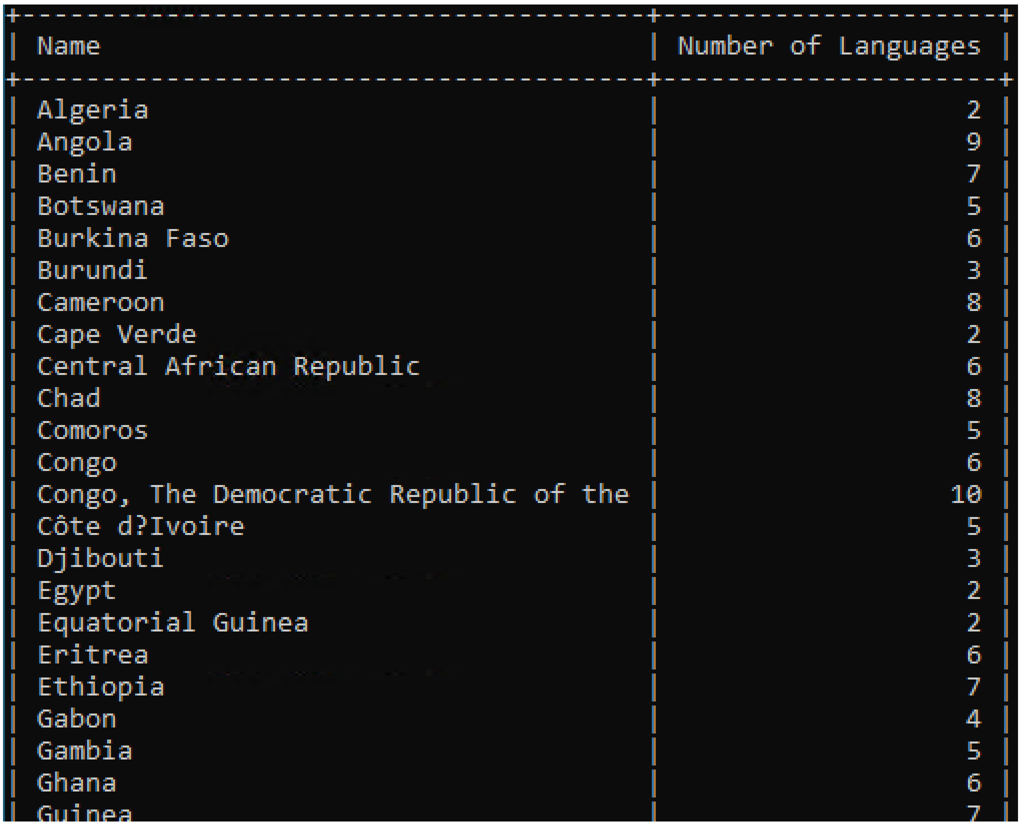
```
+---------------+------------+
| Name          | Government |
+---------------+------------+
| Austria       | Federal Re |
| Belgium       | Constituti |
| Germany       | Federal Re |
| Liechtenstein | Constituti |
| Luxembourg    | Constituti |
| Switzerland   | Federation |
+---------------+------------+
6 rows in set (0.02 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 |
```

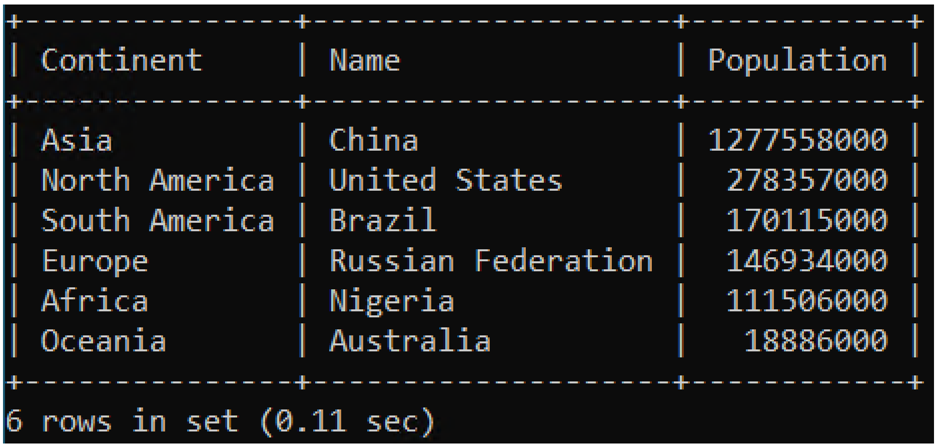
Figure 3 Example of output required for this question.

Question H (MySQLQH.txt)

Show the *Continent*, and the *Name* and *Population* of the country with the biggest population in each continent.

NOTE: Only include countries where the population is greater than 0.

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



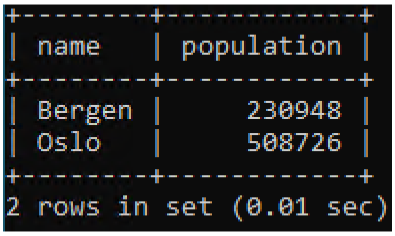
```
+---------------+--------------------+------------+
| Continent     | Name               | Population |
+---------------+--------------------+------------+
| Asia          | China              | 1277558000 |
| North America | United States      |  278357000 |
| South America | Brazil             |  170115000 |
| Europe        | Russian Federation |  146934000 |
| Africa        | Nigeria            |  111506000 |
| Oceania       | Australia          |   18886000 |
+---------------+--------------------+------------+
6 rows in set (0.11 sec)
```

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.



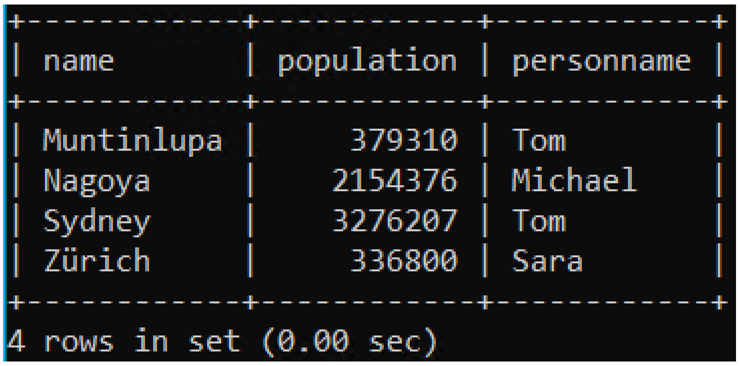
```
+--------+------------+
| name   | population |
+--------+------------+
| Bergen |     230948 |
| Oslo   |     508726 |
+--------+------------+
2 rows in set (0.01 sec)
```

Figure 5 Example of output required for this question.

Question L (MySQLQL.txt)

Show *Name*, *Population* and *PersonName* of all cities visited by people, where the city population is greater than the maximum population of “Polynesia”.

The results should be sorted alphabetically by name.



```
+------------+------------+------------+
| name       | population | personname |
+------------+------------+------------+
| Muntinlupa |     379310 | Tom        |
| Nagoya     |    2154376 | Michael    |
| Sydney     |    3276207 | Tom        |
| Zürich     |     336800 | Sara       |
+------------+------------+------------+
4 rows in set (0.00 sec)
```

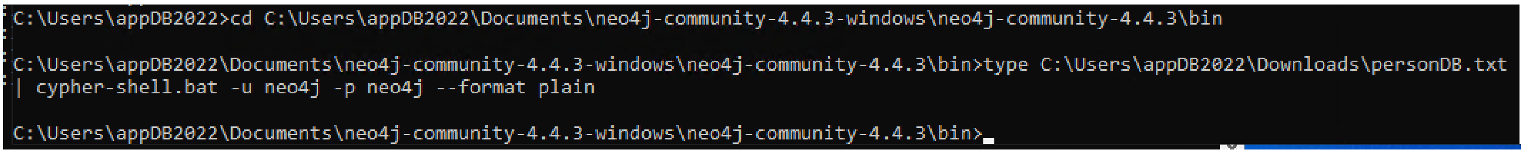
Figure 6 Example of output required for this question.

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>_
```

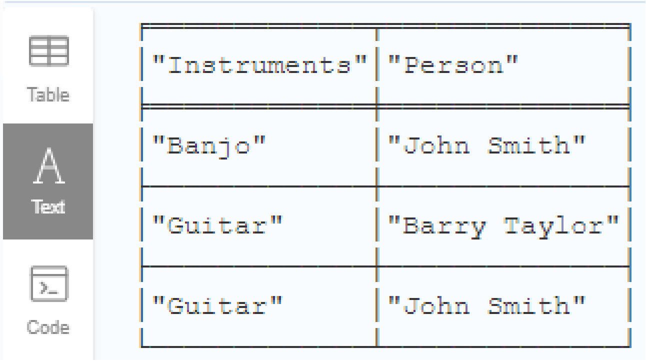
Figure 7 Import Neo4j database

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.



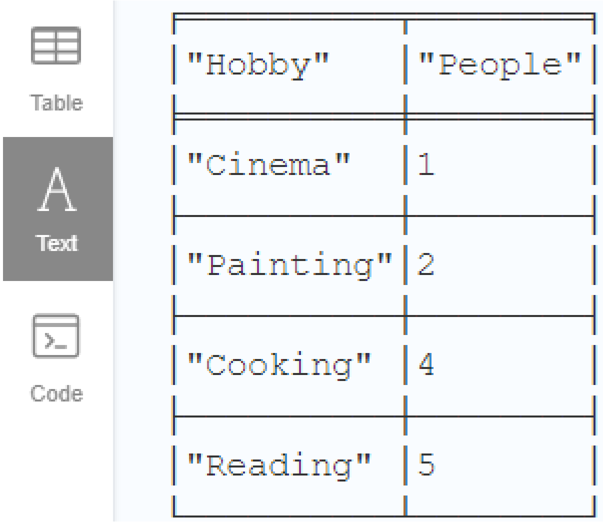
```
╒═════════════╤══════════════╕
│"Instruments"│"Person"      │
╞═════════════╪══════════════╡
│"Banjo"      │"John Smith"  │
├─────────────┼──────────────┤
│"Guitar"     │"Barry Taylor"│
├─────────────┼──────────────┤
│"Guitar"     │"John Smith"  │
└─────────────┴──────────────┘
```

Figure 8 Example of output required for this question.

Question C (Neo4jQC.txt)

Return the names of hobbies (as *Hobby*) and the number of people who have that hobby (as *People*).

The results should be sorted in increasing People order and within that by Hobby.

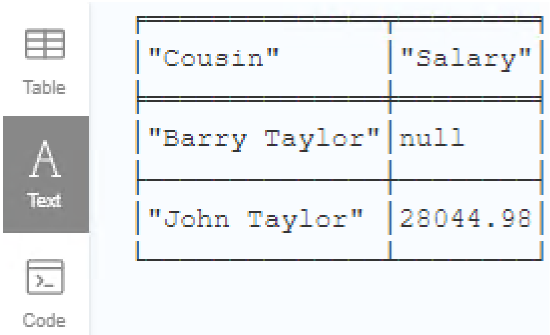


```
╒══════════╤════════╕
│"Hobby"   │"People"│
╞══════════╪════════╡
│"Cinema"  │1       │
├──────────┼────────┤
│"Painting"│2       │
├──────────┼────────┤
│"Cooking" │4       │
├──────────┼────────┤
│"Reading" │5       │
└──────────┴────────┘
```

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│
└──────────────┴────────┘
```

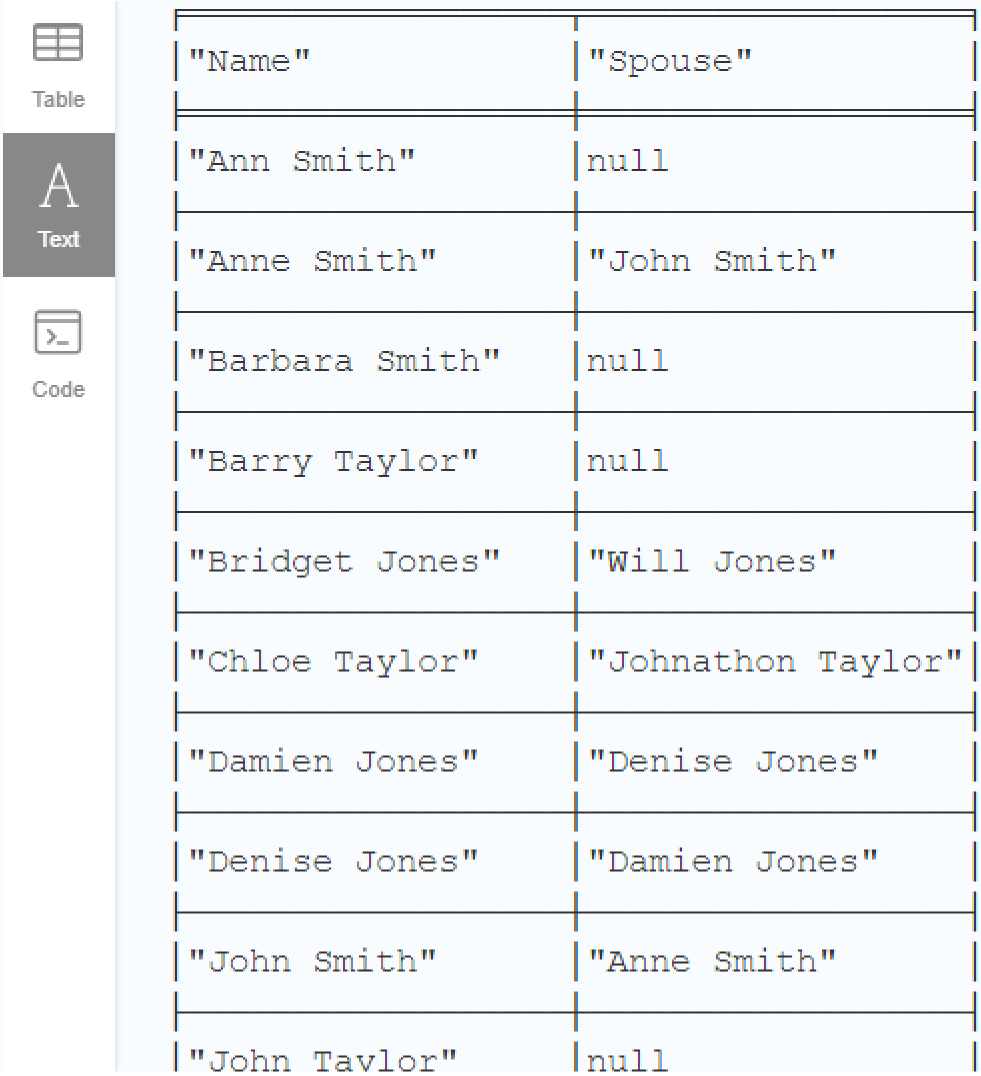
Figure 10 Example of output required for this question.

Question H (Neo4jQH.txt)

Return the name of each person (as *Name*) and the person they have a MARRIED_TO relationship with (as *Spouse*).

If someone does not have a MARRIED_TO relationship their *Spouse* should be *null*.

Results should be returned in alphabetical *Name* order, followed by alphabetical *Spouse* order.

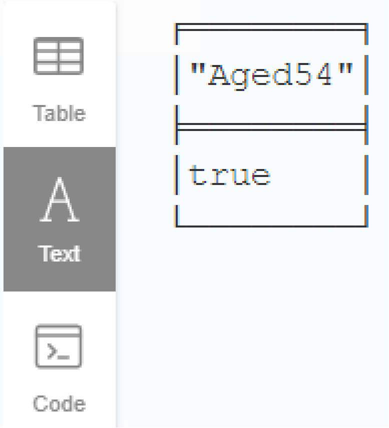


```
╒════════════════╤═══════════════════╕
│"Name"          │"Spouse"           │
╞════════════════╪═══════════════════╡
│"Ann Smith"     │null               │
├────────────────┼───────────────────┤
│"Anne Smith"    │"John Smith"       │
├────────────────┼───────────────────┤
│"Barbara Smith" │null               │
├────────────────┼───────────────────┤
│"Barry Taylor"  │null               │
├────────────────┼───────────────────┤
│"Bridget Jones" │"Will Jones"       │
├────────────────┼───────────────────┤
│"Chloe Taylor"  │"Johnathon Taylor" │
├────────────────┼───────────────────┤
│"Damien Jones"  │"Denise Jones"     │
├────────────────┼───────────────────┤
│"Denise Jones"  │"Damien Jones"     │
├────────────────┼───────────────────┤
│"John Smith"    │"Anne Smith"       │
├────────────────┼───────────────────┤
│"John Taylor"   │null               │
```

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.



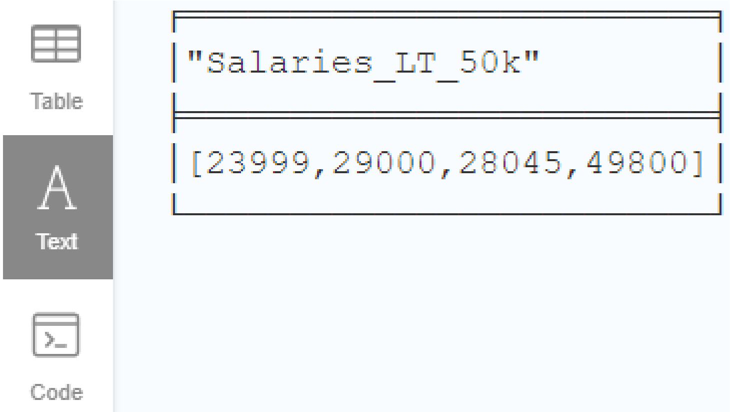
```
╒════════╕
│"Aged54"│
╞════════╡
│true    │
└────────┘
```

Figure 12 Example of output required for this question.

Question L (Neo4jQL.txt)

Return the list of salaries of people who are less than 50,000 (as *Salaries_LT_50k*). The salaries should be rounded up or down to the nearest whole number.

E.g. 100.5 becomes 101, 100.4 becomes 100.



```
╒════════════════════════════╕
│"Salaries_LT_50k"           │
╞════════════════════════════╡
│[23999,29000,28045,49800]│
└────────────────────────────┘
```

Figure 13 Example of output required for this question.