

## Contents

MySQL Questions .....	2
Question B (MySQLQB.txt) .....	2
Question C (MySQLQC.txt) .....	3
Question F (MySQLQF.txt) .....	4
Question G (MySQLQG.txt) .....	5
MongoDB Questions .....	6
Question B (MongoDBQB.txt) .....	6
Question C (MongoDBQC.txt) .....	7
Question E (MongoDBQE.txt) .....	8
Question G (MongoDBQG.txt) .....	9

## MySQL Questions

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

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

### Question B (MySQLQB.txt)

Show the *certificate* and *filmname* of all films starring "Temuera Morrison".

The results should be sorted by alphabetical *filmname*.

```
+-----+-----+
| certificate | filmname |
+-----+-----+
| 18          | Once Were Warriors |
| PG          | Star Wars: Episode II - Attack of the Clones |
| 12A         | Star Wars: Episode III - Revenge of the Sith |
+-----+-----+
3 rows in set (0.00 sec)
```

Figure 1 Example of output required for Question A

Question C (MySQLQC.txt)

Show the *filmname* of all films that have at least one actor from the United Kingdom.

The results should be sorted by alphabetical *filmname*.

```
+-----+
| filmname |
+-----+
| Around the World in 80 Days |
| Batman Begins |
| Charlie and the Chocolate Factory |
| Harry Potter and the Goblet of Fire |
| Harry Potter and the Order of the Phoenix |
| Harry Potter and the Philosopher's Stone |
| Harry Potter and the Prisoner of Azkaban |
| Jurassic Park |
| Kingdom of Heaven |
| Mission: Impossible III |
| Monsters, Inc. |
| Star Wars: Episode I - The Phantom Menace |
| Star Wars: Episode II - Attack of the Clones |
| Star Wars: Episode III - Revenge of the Sith |
| Superman Returns |
| The Bourne Ultimatum |
+-----+
16 rows in set (0.00 sec)
```

Figure 2 Example of output required for Question C

Question F (MySQLQF.txt)

Show the *directorname* and *lanauage(s)* of films the director directed that were not in English.

The results should be sorted by alphabetical *directorname*.

directorname	language
Akira Kurosawa	Japanese
Akira Kurosawa	Japanese
Akira Kurosawa	Japanese
Ang Lee	Mandarin
Clint Eastwood	Japanese
Jean-Pierre Jeunet	French
Oliver Hirschbiegel	German
Stanley Tong	Cantonese
Wolfgang Peterson	German
Yimou Zhang	Mandarin
Yimou Zhang	Mandarin

11 rows in set (0.00 sec)

Figure 3 Example of output required for Question F

Question G (MySQLQG.txt)

Show the *genrename* and the number of films in that genre.

The results should be sorted by alphabetical *genrename*.

genrename	count(*)
Comedy	2
Drama	9
Muscial	1
Other	172
Romantic	4

5 rows in set (0.00 sec)

Figure 4 Example of partial output required for Question G

## MongoDB Questions

Import *employeesDB.json* into MongoDB as follows:

```
mongoimport.exe --db=employeesDB --collection=employees  
--file=employeesDB.json
```

The database **must** be called **employeesDB**.

The collection **must** be called **employees**.

```
C:\Users\Gerard>"Program Files\MongoDB\Server\4.2\bin\mongoimport.exe" --db=employeesDB --collection=employees --file=C:\Users\Gerard\Downl  
oads\employeesDB.json  
2021-03-16T10:41:05.876+0000    connected to: mongodb://localhost/  
2021-03-16T10:41:05.902+0000    9 document(s) imported successfully. 0 document(s) failed to import.
```

Figure 5 Example mongoimport

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

### Question B (MongoDBQB.txt)

Show *\_id* and "Increase Due" for all employees.

"Increase Due" should be equal to:

- Yes if the employee's *pensionLevel* is less than 9
- No otherwise

Results should be sorted by *\_id*.

```
{ "_id" : "E001", "Increase Due" : "No" }  
{ "_id" : "E002", "Increase Due" : "Yes" }  
{ "_id" : "E003", "Increase Due" : "Yes" }  
{ "_id" : "E004", "Increase Due" : "Yes" }  
{ "_id" : "E005", "Increase Due" : "No" }  
{ "_id" : "E006", "Increase Due" : "No" }  
{ "_id" : "E007", "Increase Due" : "No" }  
{ "_id" : "E008", "Increase Due" : "Yes" }  
{ "_id" : "E009", "Increase Due" : "Yes" }
```

Figure 6 Example of output required for Question B

Question C (MongoDBQC.txt)

Show `_id` and "Salary Bracket" for all employees.

"Salary Bracket" should be equal to:

- Low if the employee's salary is less than 40,000
- Medium if the employee's salary is between 40,000.00 and 47,999.99
- High if greater than or equal to 48,000.00

Results should be sorted by `_id`.

```
< "_id" : "E01", "Salary Bracket" : "Medium" >
< "_id" : "E02", "Salary Bracket" : "High" >
< "_id" : "E03", "Salary Bracket" : "Low" >
< "_id" : "E04", "Salary Bracket" : "Low" >
< "_id" : "E05", "Salary Bracket" : "High" >
< "_id" : "E06", "Salary Bracket" : "High" >
< "_id" : "E07", "Salary Bracket" : "Medium" >
< "_id" : "E08", "Salary Bracket" : "Medium" >
< "_id" : "E09", "Salary Bracket" : "Medium" >
```

Figure 7 Example of output required for Question C

Question E (MongoDBQE.txt)

Show "*Letters in Longest Name*" which is the number of characters in the longest *name* attribute.

```
{ "Letters in Longest Name" : 15 }
```

*Figure 8 Example of output required for Question E*



Question G (MongoDBQG.txt)

Show `_id` and `"roundedSal"` which is the average salary rounded to 2 decimal places.

```
{ "_id" : null, "roundedSal" : 43913.43 }
```

*Figure 9 Example of output required for Question G*