Applied Databases

Higher Diploma in Science in Data Analytics

1 Description

This document describes the final project specification for the Applied Databases module.

2 Marks

This project is worth 60% of the marks for the module.

2.1 Marking Scheme

90% of the marks will be awarded for implementing the functionality described in this document.

- MySQL Queries – 20%
- Neo4j Queries – 20%
- Python App – 50%

- 10% of the marks will be awarded for innovation and extra functionality in the Python App. Please describe your innovation (if any) in a document entitled *innovation.doc* or *innovation.pdf* which should be stored in the *Innovation* folder of your project.

  If you are using any extra Python packages in your project as part of your innovation, please note these may not be installed on the machine your programme is being tested on. So, you should preferably have these packages installed automatically by your programme, or at a minimum give the exact command needed to manually install them in the Innovation document.

NOTE: You may be invited to an MS Teams meeting for a viva explanation of any or all parts of your submission.

2.1.1 Plagiarism

Plagiarism will be dealt with in accordance with the university's Student Code.

3 Submission

Deadline for submissions is **Monday May 6th 2024 at 9:00am**.

- Firstly, download the file FinalProject.zip from Moodle and unzip it.
- When unzipped it will contain the following:
  - A file called Questions.pdf (See section 4.2)
  - A folder called OfficialQueryResults (See section 4.3)
  - A folder called GXXXXXXXX
- Rename the unzipped folder from GXXXXXXXX to your Student Number e.g. G12345678
- The folder contains 4 sub-folders:
  - **Innovation**
    Write a Word/PDF document explaining any innovation/extra functionality you provided and place in this folder.
    If none – just leave folder empty.
  - **Neo4j-Queries**
    This folder contains 6 files, corresponding to each Neo4j question.
    Write only the exact Neo4j/Cypher command for each question into the appropriate file.
  - **MySQL-Queries**
    This folder contains 6 files, corresponding to each MySQL question.
    Write only the exact MySQL command for each question into the appropriate file.
  - **PythonApp**
    Write your Python App in this folder.
- When you are finished, compress the folder – which is now called your Student number (e.g. G12345678) – and upload to Moodle before the deadline.
  NOTE: You don't need to upload the OfficialQueryResults, nor your Answers – just your queries.
- Everything will be tested on the Virtual Machine (VM), so if you are using on your own laptop, you should still ensure that everything works on the VM.

4 Functionality

The project specification should not change. If it, or associated files do change (due to errors, omissions etc.), any updates will be posted the Announcements section of Moodle.

NOTE: It is the student's responsibility to ensure you are always working with the latest version of the specification and associated files on Moodle.

4.1 MySQL

1. Import the MySQL database as follows:

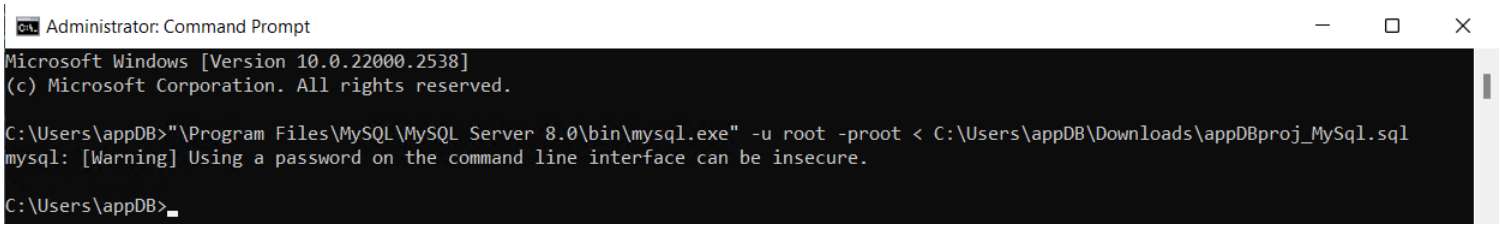

Figure 1 Importing the MySQL database

NOTE: The file containing the MySQL database is called *appDBproj_MySql.sql*, but the database will be called ***appdbproj*** (not *appdbproj_mysql*).

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| appdbproj          |
```

Figure 2 Imported database

See *Questions.pdf* for the MySQL questions.

4.2 Neo4j

1. In neo4j.conf, change the default database to appDBproj:

   ```
   initial.dbms.default_database=appDBproj
   ```

2. Ensure Neo4j is running:

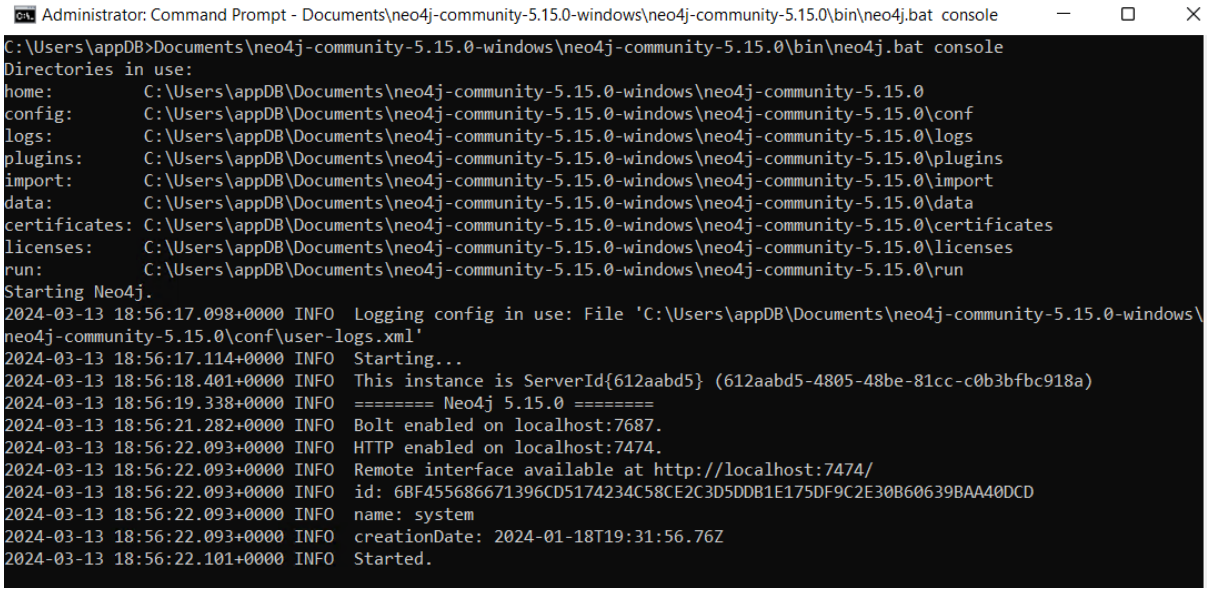

Figure 3 Run Neo4j

3. Import the contents of appDBproj_Neo4j.txt to the **`appDBproj`** database

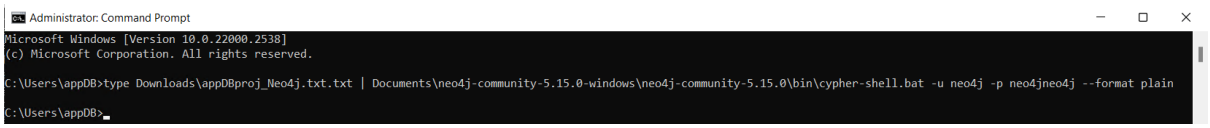

Figure 4 Import database

4. Open your browser to localhost:7474 and select the **`appDBproj`** database from the dropdown list:

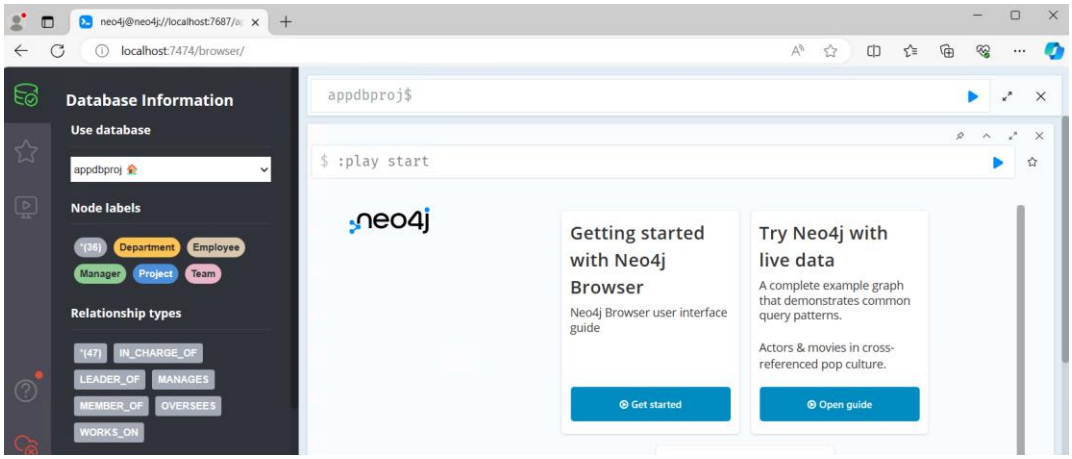

Figure 5 Use the database

See *Questions.pdf* for the Neo4j questions.

4.3 Testing Your MySQL and Neo4j Queries

The MySQL and Neo4j sections are marked a pass/fail basis. Either you get all the marks for a question or none (All questions carry equal marks).

You can test your answers as follows:

- The file **OfficialQueryResults.zip** (which can be downloaded from Moodle) contains two folders:
    - **MySQL**
      This has 6 files which each of which has the correct output for the corresponding MySQL question.

    - **Neo4j**
      This has 6 files which each of which has the correct output for the corresponding Neo4j question.

4.3.1 How to test your MySQL queries

- Write your MySQL query in the MySQL console.

- When you think its correct copy the query to appropriate file in the MySQL-Queries folder of your answer folder.

- Run the following command from the Windows command line:

  `mysql.exe -u root -proot appdbproj < MySQLQA.txt > MySQLA-myAns.txt`

  `mysql.exe` is the location of mysql.exe e.g. "C:\Program Files\MySQL\MySQL Server 8.0\bin mysql.exe".

  `-u root` is the username, in this case root.

  `-proot` is the password, in this case root (no space between p and the password).

  `appdbproj` is the MySQL database the query will be run against, in this case *appdbproj*.

  < The less than symbol means that the contents of the next file mentioned will be used as input to the mysql.exe command.

  `MySQLQA.txt` is the location of the file with your MySQL query for this question e.g. "`C:\Users\appDB\Downloads\G12345678\MySQL\MySQLQA.txt`".

  > The greater than symbol means that the output from the mysql.exe command should be written to the file mentioned next.

  `MySQLA-myAns.txt` is the location of the file your query result will be written to e.g. "`C:\Users\appDB\Downloads\MyAnswers\MySQL-Queries\MySQLQA-myAns.txt`".

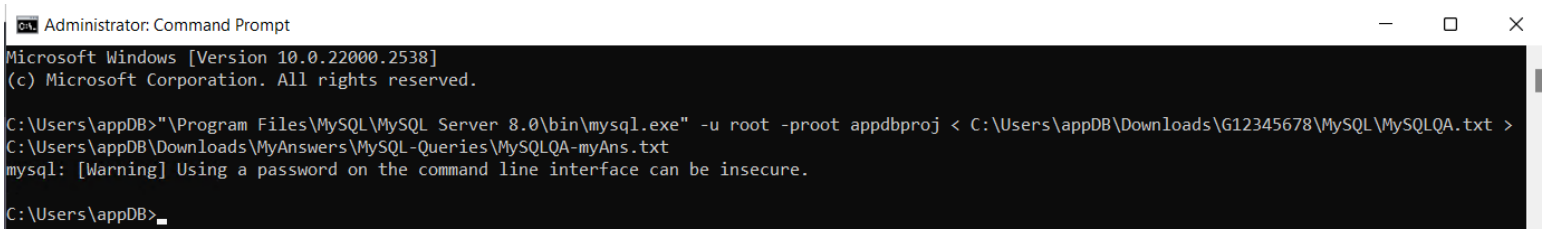

Figure 6 Creating Your MySQL result file

- Compare your answer with the correct answer:

  `fc /C MySQLQA-myAns.txt MySQLQA-Ans.txt`

  `fc` the file tool compare program in windows.

  `/C` Ingore differences in case (capitalization) when comparing files.

  MySQLQA-myAns.txt is the location of the file containing <u>your</u> query result.

  MySQLQA-Ans.txt is the location of the <u>official</u> answer for this query.

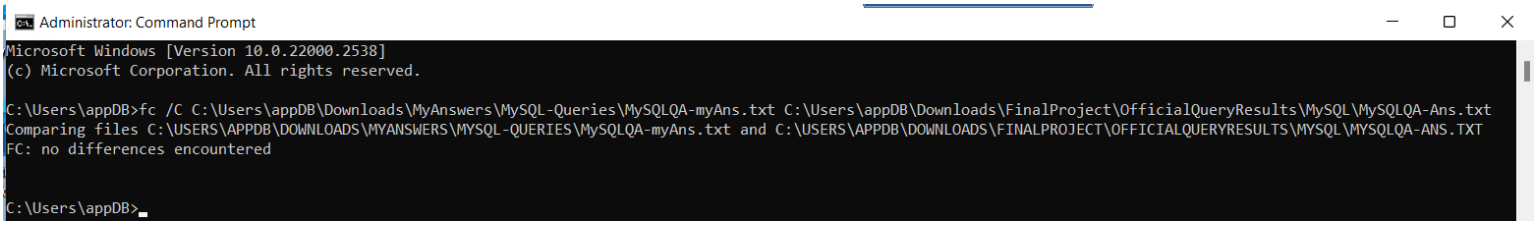

Figure 7 Checking Your MySQL result with the Official result

- <u>If the result of the fc command is not **FC: no differences encountered** no marks will be awarded for the question.</u>

4.3.2 How to test your Neo4j queries

- Write your Cypher query in the Neo4j browser.

- When you think its correct, copy the query to appropriate file in the Neo4j-Queries folder of your answer folder.

- Run the following command from the Windows command line as follows:

```
type C:\Users\appDB\Downloads\G12345678\Neo4j-Queries\Neo4jQA.txt | C:\Users\appDB\Documents\neo4j-community-5.15.0-windows\neo4j-community-5.15.0\bin\cypher-shell.bat -u neo4j -p neo4jneo4j --format plain > C:\Users\appDB\Downloads\MyAnswers\Neo4j-Queries\Neo4jQA-myAns.txt
```

type A Windows program which is used to access the contents of a file.

Neo4jQA.txt is the location of the file with your Neo4j query for this question.

| The pipe symbol, meaning the output of the command to the left of the pipe will be used as input to the command to the right of the pipe.

cypher-shell.bat A Neo4j tool used to execute scripts.

-u neo4j The Neo4j username, in this case neo4j.

-p neo4jneo4j The Neo4j password, in this case neo4jneo4j.

--format plain Minimal formatting should be used.

> The greater than symbol means that the output from cypher-shell.bat should be written to the file mentioned next.

Neo4jQA-myAns.txt is the file your query result will be written to.

Figure 8 Creating Your Neo4j Result file

- Compare your answer with the correct answer:

  `fc /C Neo4jQA-myAns.txt Neo4jQA-Ans.txt`

  `fc` the file tool compare program in windows.

  `/C` Ingore differences in case (capitalization) when comparing files.

  Neo4jQA-myAns.txt is the location of the file containing <u>your</u> query result.

  Neo4jQA-Ans.txt is the location of the <u>official</u> answer for this query.

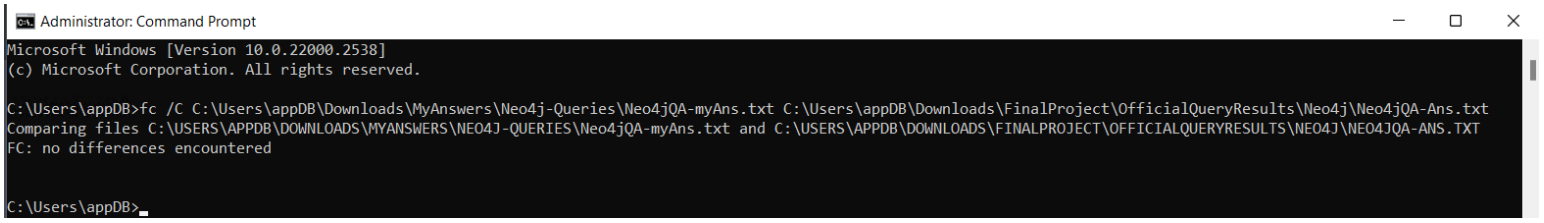

Figure 9 Checking Your Neo4j result with the Official result

- <u>If the result of the fc command is not **FC: no differences encountered** no marks will be awarded for the question.</u>

4.4 Python

The following python application should be based on the following databases:

- MySQL
  appdbproj. The same database used for the MySQL Queries.

- Neo4j
  Download *appDBCity_Neo4j.txt* from Moodle and import into a Neo4j database called *appDBCity*.

Write a python application that displays a main menu as follows:

```
========================================
                MENU
========================================

1 - View Cities by Country
2 - Update City Population
3 - Add New Person
4 - Delete Person
5 - View Countries by population
6 - Show Twinned Cities
7 - Twin with Dublin
x - Exit application
Choice:
```

Figure 10 Main Menu

The choices are as follows:

4.4.1 1 (View Cities by Country)

The user is asked to enter a country name (or part thereof):

```
========================================
                MENU
========================================

1 - View Cities by Country
2 - View Countries by Independence Year
3 - Add New Person
4 - View Countries by name
5 - View Countries by population
6 - Find Students by Address
7 - Add New Course
x - Exit application
Choice: 1

Enter Country : stat
```

Figure 11 Enter Country

The user is then shown the following details of cities in that country/those countries:

- Country Name
- City Name
- City District
- City Population

In groups of 2.

```
Enter Country : stat
Micronesia, Federated States of  |  Weno  |  Chuuk  |  22000
Micronesia, Federated States of  |  Palikir  |  Pohnpei  |  8600
-- Quit (q) --
United States  |  New York  |  New York  |  8008278
United States  |  Los Angeles  |  California  |  3694820
-- Quit (q) --
United States  |  Chicago  |  Illinois  |  2896016
United States  |  Houston  |  Texas  |  1953631
-- Quit (q) --
```

Figure 12 Country & cities in groups of 2

If the user presses any key except *q* the details of the next 2 cities in that country/those countries are shown.

Whenever the user presses *q* he/she is returned to the Main Menu.

4.4.2 2 (Update City Population)

The user is asked to enter a City ID:

```
Choice: 2

Enter City ID : 134

134  |  Adelaide  |  AUS  |  978100  |  None  |  None

[I]ncrease/[D]ecrease Population:
```

Figure 13 Valid City ID entered

When a valid City ID is entered the following details of the city are shown:

- ID
- Name
- CountryCode
- Population
- latitude
- longitude

The user is then asked whether he/she wishes to Increase or Decrease the City's Population, and by how much.

```
[I]ncrease/[D]ecrease Population: d
Enter Population Decrease: 100

========================================
                MENU
========================================

1 - View Cities by Country
2 - Update City Population
```

Figure 14 Increase/Decrease Population

The City's population is then increased/decreased by the specified amount and the user brought back to the Main Menu.

```
mysql> SELECT * FROM city where id = 134;
+-----+----------+-------------+-----------------+------------+----------+-----------+
| ID  | Name     | CountryCode | District        | Population | latitude | longitude |
+-----+----------+-------------+-----------------+------------+----------+-----------+
| 134 | Adelaide | AUS         | South Australia |     978000 |     NULL |      NULL |
+-----+----------+-------------+-----------------+------------+----------+-----------+
1 row in set (0.00 sec)
```

Figure 15 Population Updated

4.4.2.1 Error Conditions

- The user is asked for a City ID until a valid one has been entered.

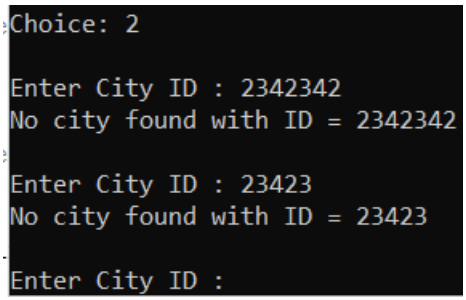

Figure 16 Invalid City ID entered

- Once a valid City ID has been entered, the user is asked whether the city's population should be Increased (I or i) or Decreased (D or d), until a valid response has been entered.

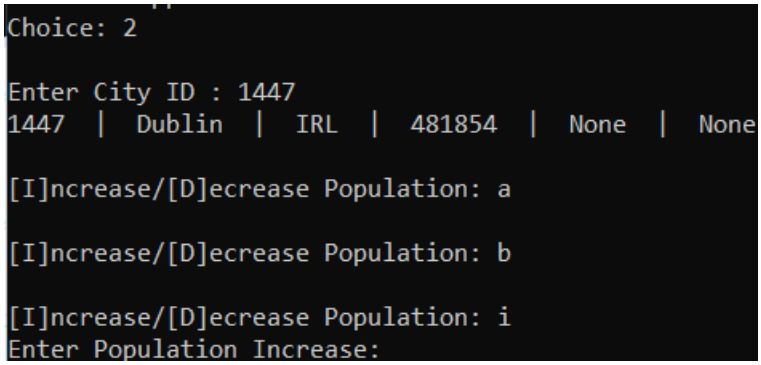

Figure 17 Invalid Increase/Decrease choice entered

4.4.3 3 (Add New Person)

The user is asked to enter the following details for a Person:

- ID
- Name
- Age
- Salary
- ID of the city the person lives in.

When the person has been successfully added to the database, the user is returned to the main menu.

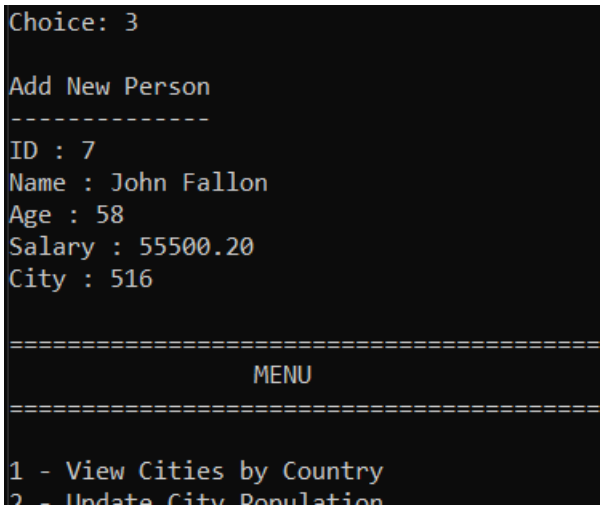

Figure 18 Add new Person

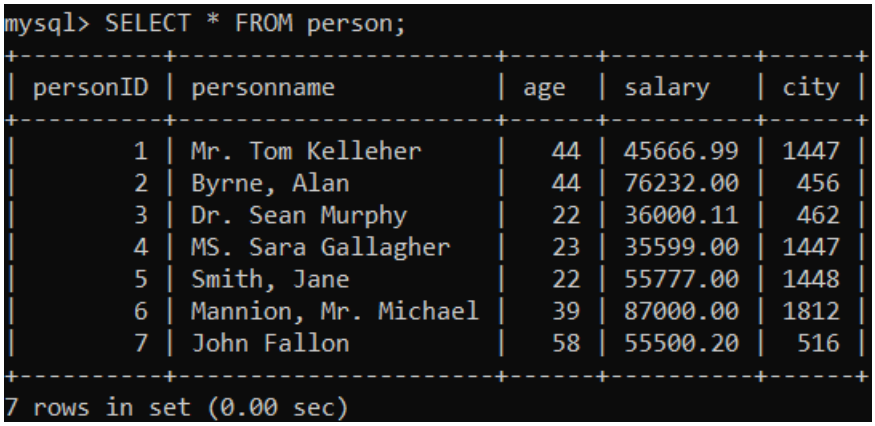

Figure 19 New Person added

4.4.3.1 Error Conditions

If the ID entered for a Person already exists, an error message should say this, and the user returned to the main menu.

```
Choice: 3

Add New Person
--------------
ID : 1
Name : Amanda Kelly
Age : 28
Salary : 35101.77
City : 178

Error: Person ID: 1 already exists

========================================
                MENU
========================================

1 - View Cities by Country
```

Figure 20 Employee Number already exists

If the City entered for a Person does not exit, an error message should say this, and the user returned to the main manu.

```
Choice: 3

Add New Person
--------------
ID : 99
Name : Amanda Kelly
Age : 28
Salary : 35101.77
City : 9999

Error: City ID: 9999 does not exist

========================================
                MENU
========================================

1 - View Cities by Country
```

Figure 21 City does not exist

For any other error e.g. Invalid ID, Name, Age, or City, the error reported by the database can be shown, and the user returned to the main menu.

```
Choice: 3

Add New Person
--------------
ID : 8
Name : Damien Lunny
Age : 59
Salary : abcd
City : 146
(1265, "Data truncated for column 'salary' at row 1")

========================================
                MENU
========================================

1 - View Cities by Country
```

Figure 22 Other error

4.4.4 4 (Delete Person)

The user is asked to enter the ID of the person to be deleted.

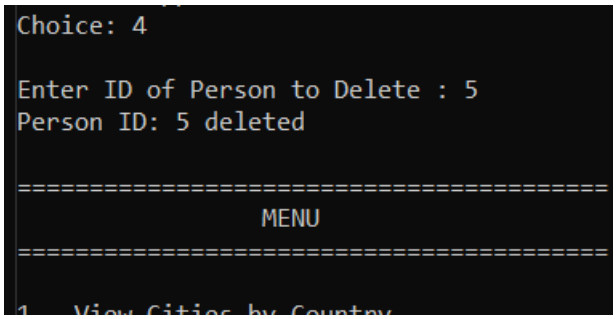

Figure 23 Enter ID of person to be deleted

This person is then deleted from the database, and the user returned to the main menu.

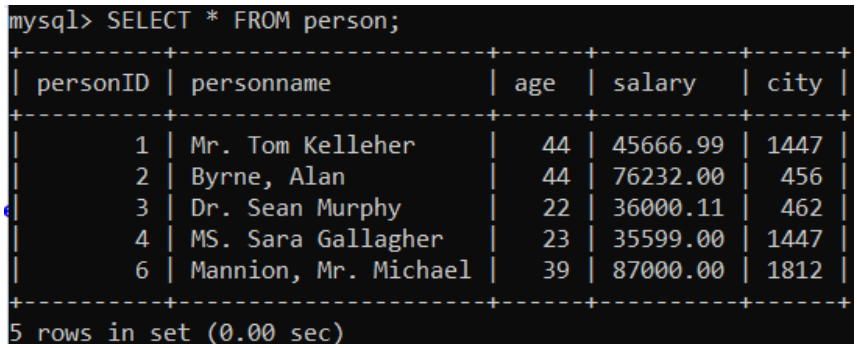

Figure 24 PersonID 5 deleted from database

4.4.4.1 Error Conditions

If the person with the specified ID has visited cities, he/she should not be deleted from the database, and the user should be returned to the main menu.

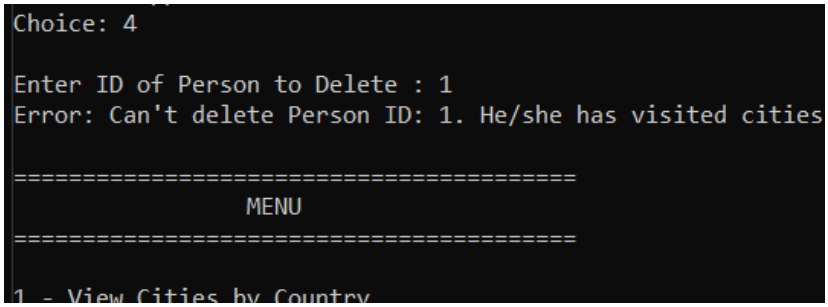

Figure 25 Spouse of Employee Shown

For any other error, the error reported by the database can be shown, and the user returned to the main menu.

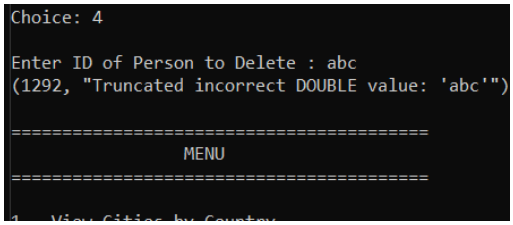

Figure 26 Other Error

4.4.5 5 (View Countries by population)

The user is asked to enter <, > or =, followed by a population. For any country whose population is <, > or = (as appropriate) the population entered by the user, the following information is shown:

- Code
- Name
- Continent
- Population

Figure 27 Countries with Population > 200000000

4.4.5.1 Error Conditions

The user is continually prompted for one of the valid comparison operators, <, > or =, until a valid one is entered.

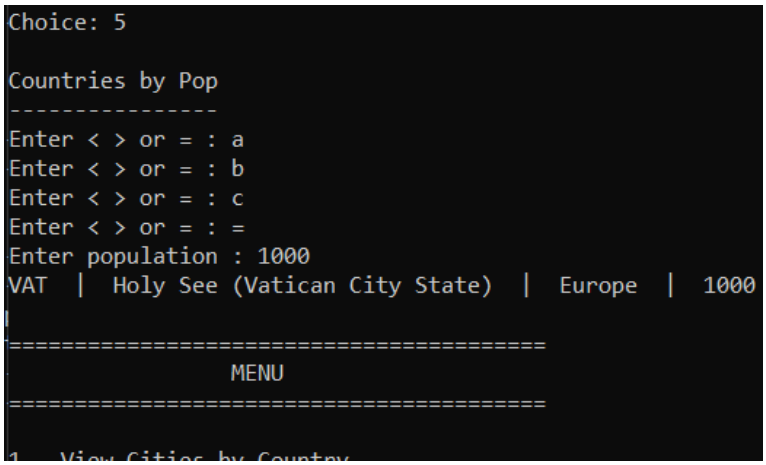

Figure 28 Invalid comparison operators.

4.4.6 6 (Show Twinned Cities)

When this option is chosen, the list of twinned cities (from the Neo4j database) is shown in alphabetical order.

Figure 29 Employee Titles

NOTE: The direction of the twinning is unimportant. In the initial database setup, the relationship between Cork & Toulouse is:

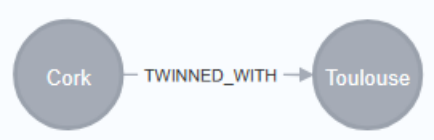

Figure 30 TWINNED_WITH relationship

But in the Python application the following Twinned Cities are shown:

```
Cork <-> Toulouse
```

```
Toulouse <-> Cork
```

4.4.7 7 (Twin with Dublin)

The user is asked to enter the ID of a city to be twinned with Dublin in the Neo4j database.

Scenario 1

The city with the specified ID doesn't already exist in the Neo4j database, so it is created (along with the TWINNED_WITH relationship).

```
Choice: 7

Enter ID of City to twin with Dublin: 72
Dublin is now twinned with Rosario

========================================
                MENU
========================================

1 - View Cities by Country
```

Figure 31 Twin city with Dublin

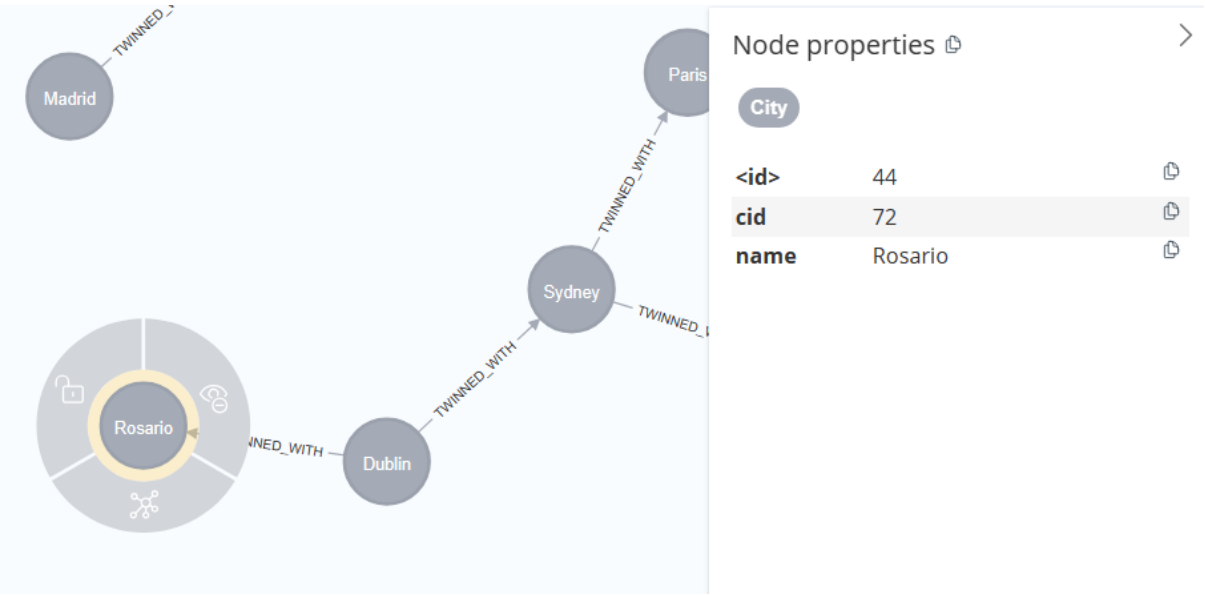

Figure 32 New node with correct cid and name added to Neo4j database and TWINNED_WITH Dublin

Scenario 2

The city with the specified ID already exists in the Neo4j database, so only the TWINNED_WITH relationship is created.

```
Choice: 7

Enter ID of City to twin with Dublin: 653
Dublin is now twinned with Madrid

========================================
                MENU
========================================

1 - View Cities by Country
```

Figure 33 Twin city with Dublin

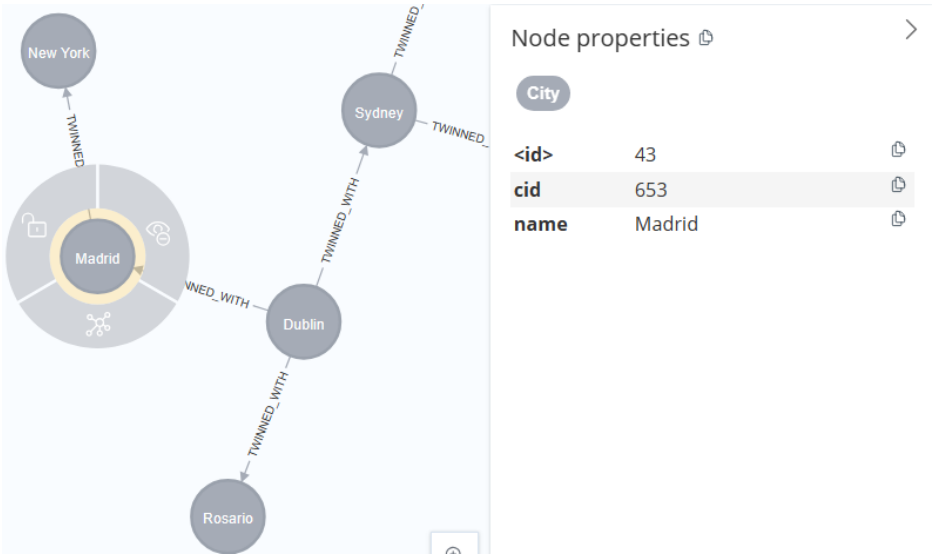

Figure 34 Existing node with TWINNED_WITH Dublin

Scenario 3

The city with the specified ID is already TWINNED_WITH Dublin, so nothing needs to be done.

```
Choice: 7

Enter ID of City to twin with Dublin: 130
Dublin is now twinned with Sydney

========================================
                MENU
========================================

1 - View Cities by Country
```

Figure 35 City 130 already TWINNED_WITH Dublin

4.4.7.1 Error Conditions

If a City ID is entered that does not exist in the MySQL database, an error message should be printed, and the user given the opportunity to enter a valid City ID.

```
Choice: 7

Enter ID of City to twin with Dublin: 12341234
Error: City ID: 12341234 doesn't exist in MySQL database

Enter ID of City to twin with Dublin:
```

Figure 36 City doesn't exist in MySQL database

If Dublin has been deleted from the Neo4j database, then an attempt to twin a valid city with it should result in the following error.

```
Choice: 7

Enter ID of City to twin with Dublin: 515
Error: Dublin does not exist in Neo4j database

========================================
                MENU
========================================

1 - View Cities by Country
```

Figure 37 Dublin has been deleted from Neo4j database

4.4.8 x (Exit Application)

The program terminates.

4.4.9 Anything Else

The menu is shown again.