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.
  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 8th 2023 at 9:00am**.

- Firstly, download the file GXXXXXXXX.zip from Moodle.
- Unzip it.
- 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 and upload to Moodle before the deadline.

- 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.

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:

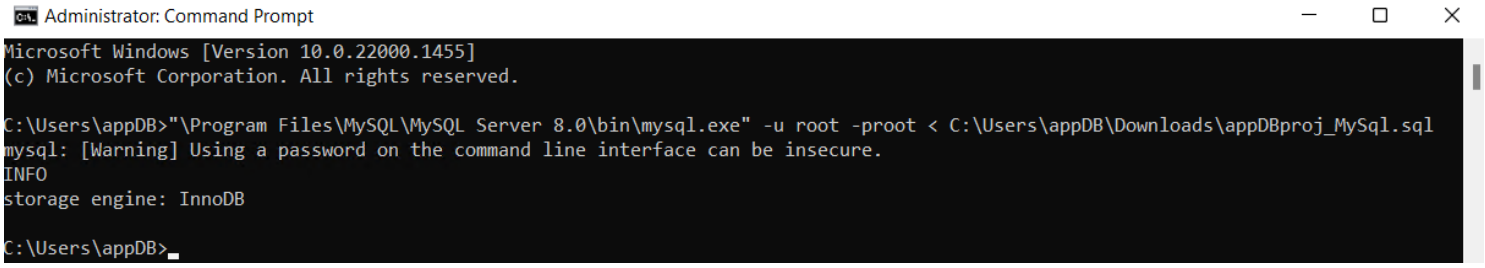

Figure 1 Importing the MySQL database

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

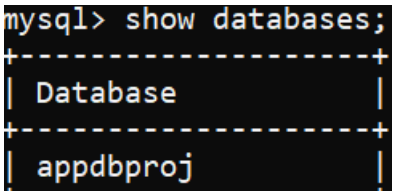

Figure 2 Imported database

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:

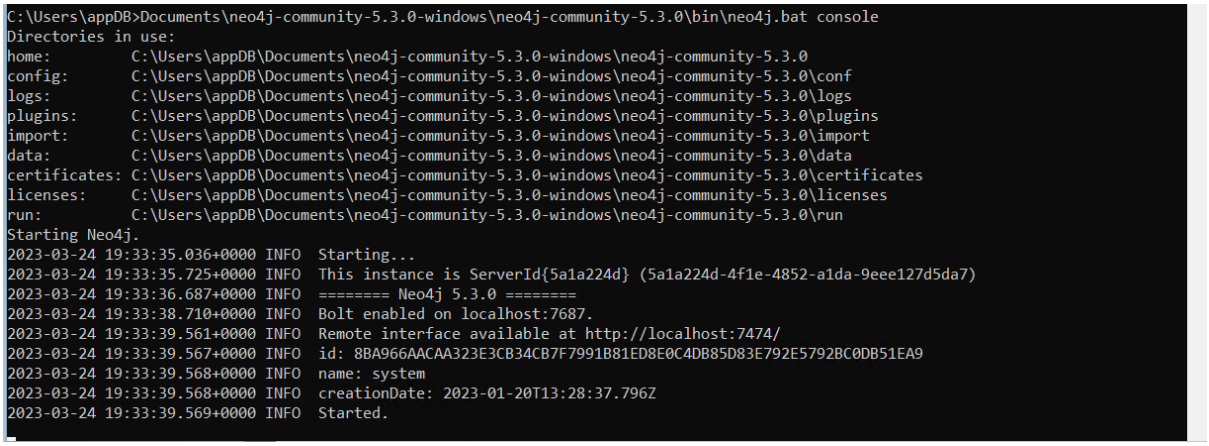

Figure 3 Run Neo4j

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:

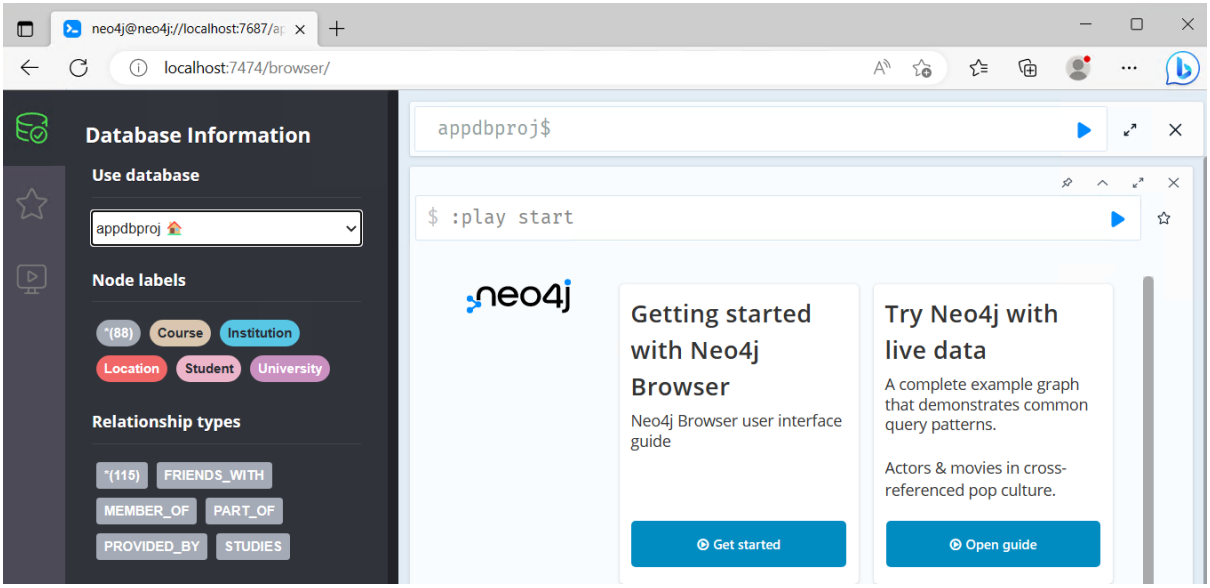

Figure 5 Use the database

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\GXXXXXXXX\MySQL-Queries\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`".

```
C:\Program Files\MySQL\MySQL Server 8.0\bin>mysql.exe -u root -proot appdbproj < C:\Users\appDB\Downloads\G12345678\MySQL-Queries\MySQLQA.txt >
C:\Users\appDB\Downloads\MyAnswers\MySQL-Queries\MySQLQA-myAns.txt
mysql: [Warning] Using a password on the command line interface can be insecure.

C:\Program Files\MySQL\MySQL Server 8.0\bin>
```

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.

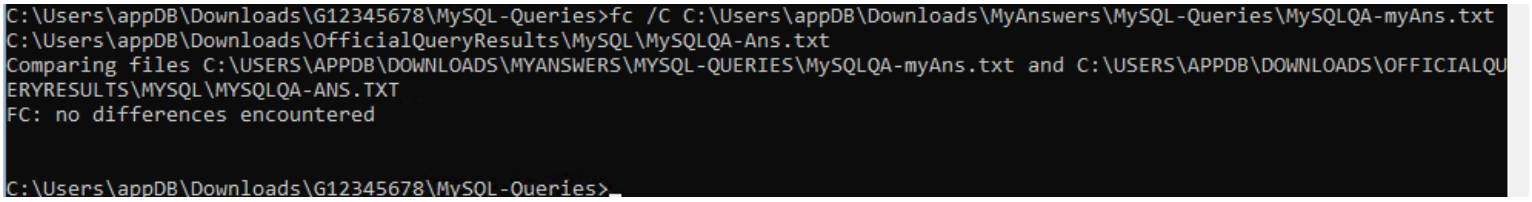

Figure 7 Checking Your MySQL result with the Official result

- <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-5.3.0-windows\neo4j-community-5.3.0\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.

```
C:\Users\appDB\Downloads>type G12345678\Neo4j-Queries\Neo4jQA.txt | C:\Users\appDB\Documents\neo4j-community-5.3.0-windo
ws\neo4j-community-5.3.0\bin\cypher-shell.bat -u neo4j -p neo4jneo4j --format plain > MyAnswers\Neo4j-Queries\Neo4jQA-my
Ans.txt

C:\Users\appDB\Downloads>
```

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.

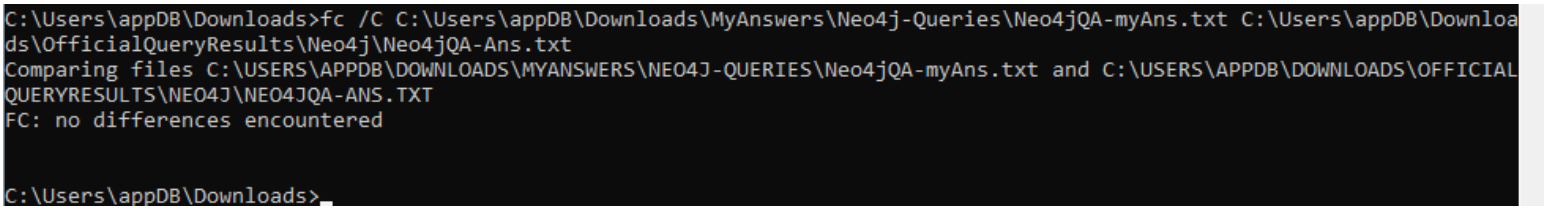

Figure 9 Checking Your Neo4j result with the Official result

- <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 *employeesMarried.txt* from Moodle and import into Neo4j.

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

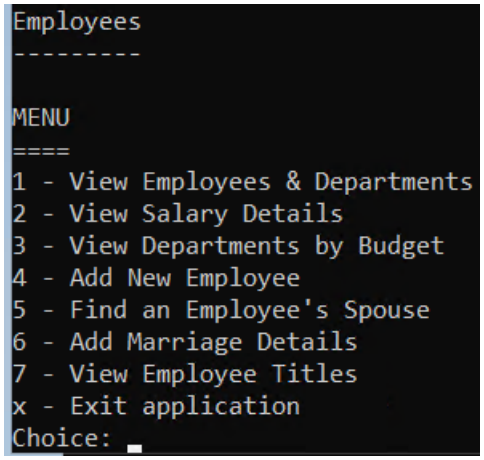

Figure 10 Main Menu

The choices are as follows:

4.4.1 1 (View Employees & Departments)

For each employee, the user is shown the employee's:

- Number
- Last name
- The number of any department he/she works/worked in
- The name of the department he/she works/worked in.

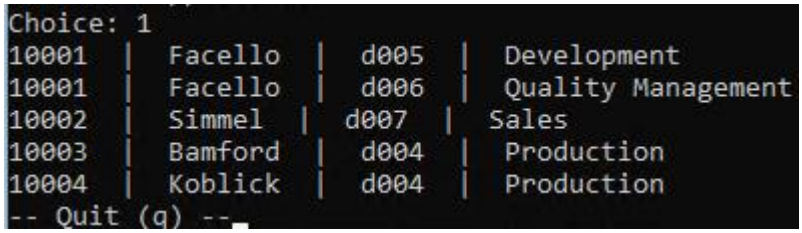

Figure 11 First group of Employees and their Departments

If the user presses any key except *q* the details of the next 5 Employees and their Department are shown:

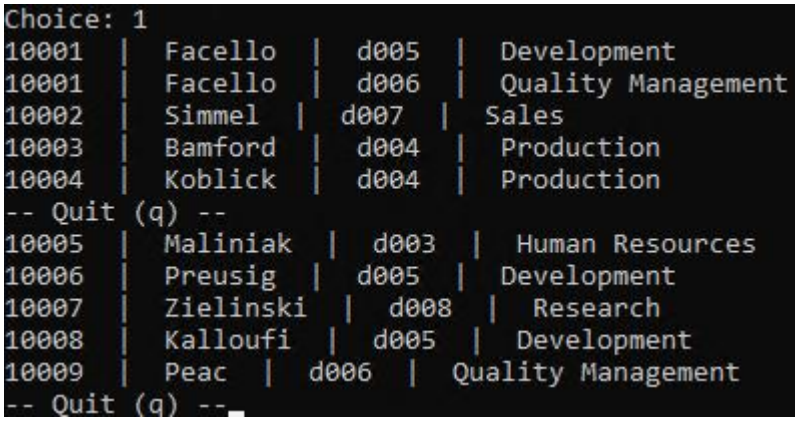

Figure 12 Next group of Employees and their Departments

And so on until the user presses *q*:

Figure 13 All Employees have been retrieved from the database

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

4.4.2 2 (View Salary Details)

The user is asked to enter a month:

```
Choice: 2
Enter Month : feb
```

Figure 14 Enter Month

When a valid month is entered, the following details are shown:

- The minimum salary, that any employee born in that month was ever on.
- The maximum salary, that any employee born in that month was ever on.
- The average salary of all employees born in that month.

```
Choice: 2
Enter Month : feb

Min, Max and Average Salaries
-----------------------------
41,538.00  |  95,035.00  |  63,772.46
```

Figure 15 Salary details of employees born in February

If an invalid month is entered, the user is prompted to enter a valid month.
Valid month values are:

- A number from 1 to 12 inclusive
- A String containing the first 3 characters (uppercase, lowercase, or both) of the month e.g. Jan, Feb.

```
Choice: 2
Enter Month : 13
Enter Month : 0
Enter Month : March
Enter Month : 3

Min, Max and Average Salaries
-----------------------------
43,590.00  |  67,492.00  |  53,316.07
```

Figure 16 3 Invalid months, followed by a valid month

```
Enter Month : ApR

Min, Max and Average Salaries
-----------------------------
39,551.00  |  94,443.00  |  55,684.62
```

Figure 17 Valid month

4.4.3 3 (View Departments by Budget)

The user is asked to enter a budget:

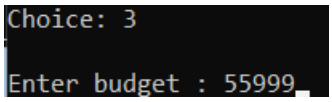

Figure 18 Enter budget

The department number, and the total number of employees currently working in that department are shown:

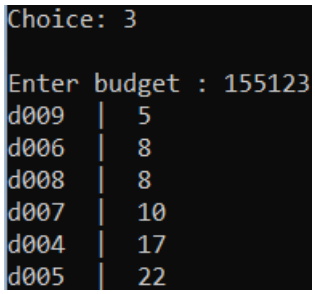

Figure 19 Details departments with budget greater than amount entered

If a non-numeric budget is entered, the user should be prompted to re-enter a correct budget:

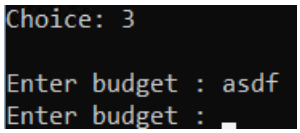

Figure 20 Incorrect budget entered

4.4.4 4 (Add New Employee)

The user is asked to enter an Employee Number, First Name, Last Name, Gender, Birth Date and Hire Date.

When the employee has been successfully added to the database the message “Employee successfully added” should be shown:

```
Choice: 4

Add New Employee
----------------
Employee Number : 10076
First Name : John
Last Name : Murphy
Gender : M
Birth Date : 1966-04-12
Hire Date : 2004-08-11

Employee successfully added
```

Figure 21 Add New Employee

If the Employee Number already exists, the user should be informed via the following error message:

```
Choice: 4

Add New Employee
----------------
Employee Number : 10070
First Name : Jim
Last Name : Connolly
Gender : M
Birth Date : 1971-01-01
Hire Date : 2009-06-15
*** ERROR *** 10070 already exists
```

Figure 22 Employee Number already exists

For any other input error e.g. Invalid Employee Number, Gender, Birth Date, Hire Date etc., the error reported by the database can be shown:

```
Choice: 4

Add New Employee
----------------
Employee Number : 10076
First Name : Jim
Last Name : Connolly
Gender : M
Birth Date : 1971-01-99
Hire Date : 2009-06-15
*** ERROR ***  (1292, "Incorrect date value: '1971-01-99' for column 'birth_date' at row 1")
```

Figure 23 Invalid date of birth

4.4.5 5 (Find an Employee's Spouse - *MySQL & Neo4j*)

The user is asked to enter an Employee Number.

The Employee Number, First Name and Last Name of the spouse of the Employee with the entered Employee Number is shown.

```
Choice: 5

Enter Employee Number : 10015
Spouse of:  10015
----------------------------
10018  |  Kazuhide  |  Peha
```

Figure 24 Spouse of Employee Shown

If no Employee with the entered Employee Number exists, or the Employee has no spouse, nothing is shown.

```
Choice: 5

Enter Employee Number : 77777
Spouse of:  77777
----------------------------
```

Figure 25 Employee does not exist, or has no spouse

4.4.6 6 (Add Marriage Details – *MySQL & Neo4j*)

The user is asked to two Employee Numbers. If all conditions are satisfied (see below) the Neo4j database should be updated with a *MARRIED_TO* relationship between these nodes..

```
Choice: 6

Enter 1st Employee Number : 10075
Enter 2nd Employee Number : 10076
Employee  10075  is now married to  10076
```

Figure 26 Two Employees Married

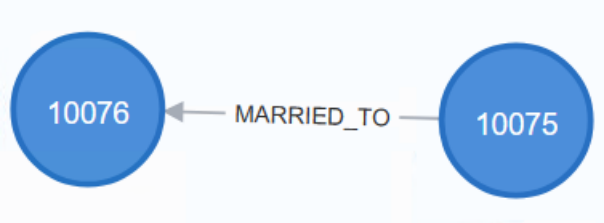

Figure 27 Neo4j database updated with relationship.

If one or both Employee Numbers do not exist in the MySQL database, both should be re-entered.

```
Choice: 6

Enter 1st Employee Number : 10074
Enter 2nd Employee Number : 99999

Employee 99999 does not exist

Enter 1st Employee Number : 88888
Enter 2nd Employee Number : 99999
Employee 88888 does not exist
Employee 99999 does not exist

Enter 1st Employee Number :
```

Figure 28 Non-existent Employee Number(s).

If either Employee already has a *MARRIED_TO* relationship, an error message indicating this should be shown.:

```
Choice: 6

Enter 1st Employee Number : 10074
Enter 2nd Employee Number : 10009

ERROR: Employee 10074 is already married
```

Figure 29 Employee 10074 is already married.

Note: It is possible for an Employee to manage many Departments.

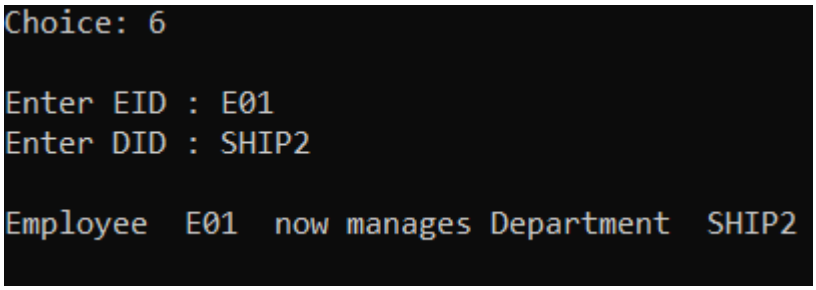

Figure 30 An Employee can manage many Departments.

4.4.7 7 (View Employee Titles)

When this option is chosen, each employees Employee Number and Title(s) is/are shown:

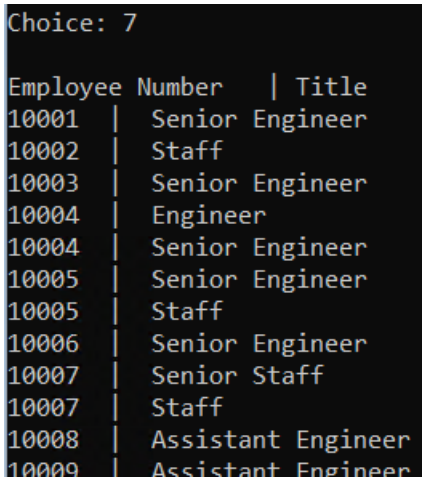

Figure 31 Employee Titles

The information should be read from the database only once.

E.g. If the user chooses 7 (View Employee Titles) the required information is read from the database and stored in the programme. If the user chooses option 7 again, the information is **not** read from the database again. Instead, the information read the first-time option 7 was chosen is output.

4.4.8 x (Exit Application)

The program terminates.

4.4.9 Anything Else

The menu is shown again.