Applied Databases

Higher Diploma in Science in Data Analytics

1 Description 3
2 Marks 3
2.1 Marking Scheme 3
2.1.1 Plagiarism 3
3 Submission 4
4 Functionality 5
4.1 MySQL 5
4.2 Neo4j 5
4.3 Testing Your MySQL and Neo4j Queries 6
4.3.1 How to test your MySQL queries 7
4.3.2 How to test your Neo4j queries 9
4.4 Python 12
4.4.1 1 (View Employees & Departments) 13
4.4.2 2 (View Salary Details) 14
4.4.3 3 (View by Month of Birth) 15
4.4.4 4 (Add New Employee) 17
4.4.5 5 (View Departments managed by Employee) 19
4.4.6 6 (Add Manager to Department) 20
4.4.7 x (Exit Application) 23
4.4.8 Anything Else 23

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.

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 institute's Plagiarism policy.

3 Submission

Deadline for submissions is **Friday May 6th 2022 at 9:00am**.

- Firstly, download the file GXXXXXXXX.7z from the *MySQL & Neo4j* section of 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 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 (using 7zip) and upload to Moodle before the deadline.

4 Functionality

4.1 MySQL

See *Questions.pdf*.

4.2 Neo4j

See *Questions.pdf*.

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.7z** 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 world < MySQLQA.txt > MySQLA-myAns.txt
  ```

  mysql.exe is the location of mysql.exe e.g. "\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).

  world is the MySQL database the query will be run against, in this case world.

  < 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\appDB2022\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\appDB2022\Downloads\MyAnswers\MySQL-Queries\MySQLQA-myAns.txt".

```
C:\Users\appDB2022>"\Program Files\MySQL\MySQL Server 8.0\bin\mysql.exe" -u root -proot wo
rld < C:\Users\appDB2022\Downloads\GXXXXXXXX\MySQL-Queries\MySQLQA.txt > C:\Users\appDB202
2\Downloads\MyAnswers\MySQL-Queries\MySQLQA-myAns.txt
mysql: [Warning] Using a password on the command line interface can be insecure.

C:\Users\appDB2022>
```

Figure 1 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 your query result.

  MySQLQA-Ans.txt is the location of the official answer for this query.

```
C:\Users\appDB2022>fc /C C:\Users\appDB2022\Downloads\MyAnswers\MySQL-Queries\MySQLQA-myAn
s.txt C:\Users\appDB2022\Downloads\OfficialQueryResults\MySQL\MySQLQA-Ans.txt
Comparing files C:\USERS\APPDB2022\DOWNLOADS\MYANSWERS\MYSQL-QUERIES\MySQLQA-myAns.txt and
 C:\USERS\APPDB2022\DOWNLOADS\OFFICIALQUERYRESULTS\MYSQL\MYSQLQA-ANS.TXT
FC: no differences encountered


C:\Users\appDB2022>
```

Figure 2 Checking Your MySQL result with the Official result

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

4.3.2 How to test your Neo4j queries

- Download *personDB.txt* from moodle.

- Update the *neo4j.conf* file to create a new database if you wish (when *personDB.txt* has been used to create the new database all data in the current database will be deleted).

- Create the Neo4j database as follows:
  `type personDB.txt | cypher-shell.bat -u neo4j -p neo4j –format plain`

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

  `personDB.txt` The commands to set up the Neo4j database.

  `|` 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 neo4j` The Neo4j password, in this case neo4j.

  `--format plain` Minimal formatting should be used.

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

C:\Users\appDB2022\Downloads>
```

Figure 3 Setting up the Neo4j database

- 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 Neo4jQA.txt | c:\Users\appDB2022\Documents\neo4j-community-4.4.3-windows\neo4j-community-4.4.3\bin\cypher-shell.bat -u neo4j -p neo4j --format plain > c:\Users\appDB2022\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 neo4j The Neo4j password, in this case neo4j.

  --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\appDB2022\Downloads\GXXXXXXXX\Neo4j-Queries>type Neo4jQA.txt | c:\Users\appDB2022\Documents\neo4j-community-4.4.3-windows\neo4j-community-4.4.3\bin\cypher-shell.bat -u neo4j -p neo4j --format plain > c:\Users\appDB2022\Downloads\MyAnswers\Neo4j-Queries\Neo4jQA-myAns.txt

C:\Users\appDB2022\Downloads\GXXXXXXXX\Neo4j-Queries>
```

Figure 4 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.

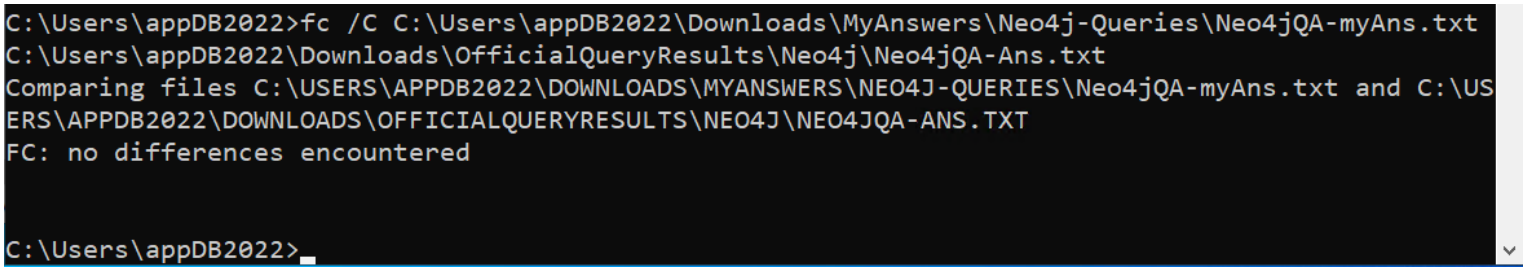

Figure 5 Checking Your Neo4j result with the Official result

- <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
  Download *employees.sql* from Moodle and import into MySQL. (The database will be called **employees**.)

- Neo4j
  Download *mgr.txt* from Moodle and import into Neo4j.

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

```
Employees
---------

MENU
====
1 - View Employees & Departments
2 - View Salary Details
3 - View by Month of Birth
4 - Add New Employee
5 - View Departments managed by Employee
6 - Add Manager to Department
7 - View Departments
x - Exit application
Choice: _
```

Figure 6 Main Menu

The choices are as follows:

4.4.1 1 (View Employees & Departments)

The user is shown the list of Employee Names and the Names of the Department each employee works in, in groups of 2. Results are shown in alphabetical order by Department Name and within that in alphabetical order by Employee Name:

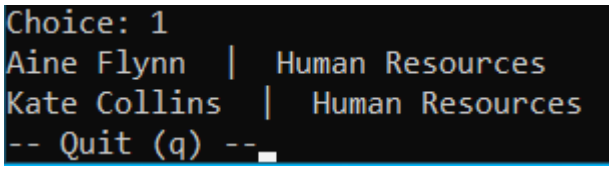

Figure 7 First group of Employees and their Departments

If the user presses any key except *q* the next 2 Employees and their Department are shown:

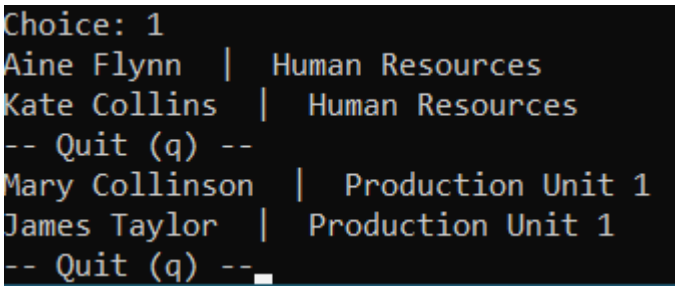

Figure 8 Next group of Employees and their Departments

And so on until the user presses *q*:

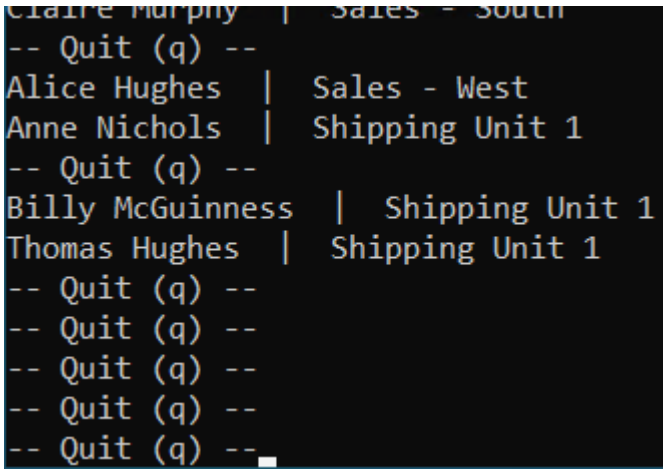

Figure 9 All Employees have been retrieved from the database

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 an Employee ID:

```
Choice: 2

Enter EID : _
```

Figure 10 Enter Employee ID

When an Employee ID is entered, the following details are shown:

- The employee's minimum salary, formatted with commas
- The employee's average salary, formatted with commas
- The employee's maximum salary formatted, with commas to 2 decimal places

```
Choice: 2

Enter EID : E01

Salary Details For Employee:  E01
---------------------------
Minimun    |  Average    |  Maximum
15,000.00  |  29,617.11  |  41,234.21
```

Figure 11 Employee salary details

If a non-existent Employee ID is entered, no salary data is shown.

```
Choice: 2

Enter EID : asdf

Salary Details For Employee:  asdf
---------------------------
Minimun    |  Average    |  Maximum
```

Figure 12 Non-existent Employee ID entered

4.4.3 3 (View by Month of Birth)

The user is asked to enter a Month:

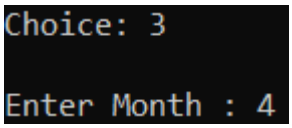

Figure 13 Enter month

The Employee ID, Name, and date of birth for all employees born in that month are then shown:

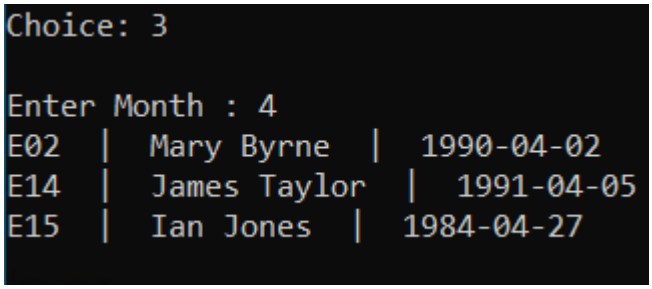

Figure 14 Details of Employees born in month 4

The month can alternatively be entered using the first 3 characters of the month's name:

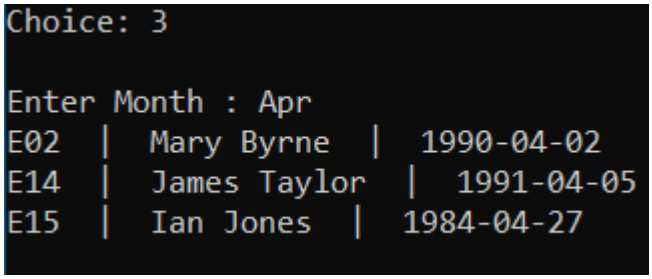

Figure 15 Details of Employees born in April

The case (capitalisation) of the month name should be ignored:

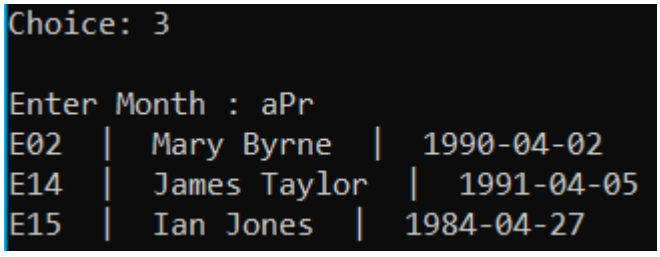

Figure 16 Details of Employees born in April

If an invalid month is entered, the user should continue to be asked to enter a month until a valid month as been entered:

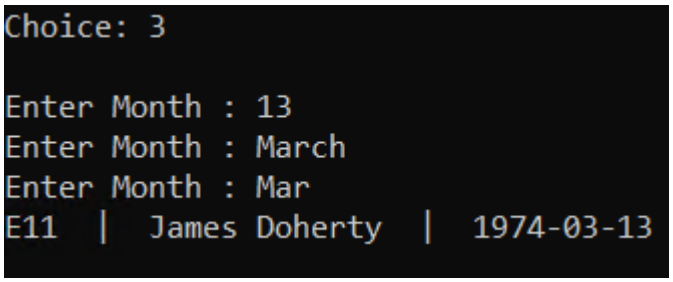

Figure 17 Invalid months entered

If a valid month is entered, but no employees were born in that month, nothing should be shown.

4.4.4 4 (Add New Employee)

The user is asked to enter an Employee ID, Name, and date of birth, as well as the ID of the department the employee will work in.

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

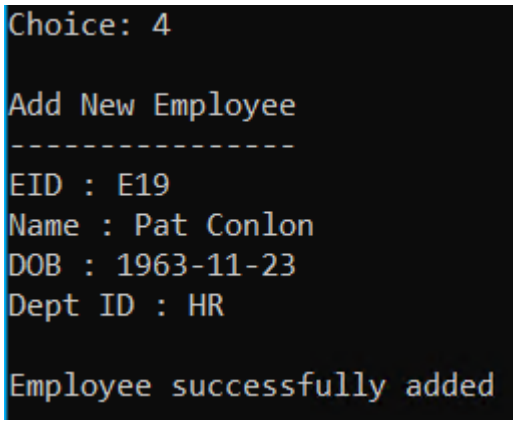

Figure 18 Add New Employee

If any of the following errors occur, the user should be informed and the employee not added to the database:

- Employee ID already exists

```
Choice: 4

Add New Employee
----------------
EID : E01
Name : Thomas Clinton
DOB : 1981-07-25
Dept ID : SHIP1

*** ERROR ***: E01 already exists
```

Figure 19 Employee ID already exists

- Invalid date of birth

```
Choice: 4

Add New Employee
----------------
EID : E20
Name : Thomas Clinton
DOB : 1981-07-99
Dept ID : SHIP1

*** ERROR ***: Invalid DOB:  1981-07-99
```

Figure 20 Invalid date of birth

- Department doesn't exist

```
Choice: 4

Add New Employee
----------------
EID : E20
Name : Thomas Clinton
DOB : 1981-07-25
Dept ID : XXX

*** ERROR ***: Department XXX  does not exist
```

Figure 21 Department doesn't exist

4.4.5 5 (View Departments managed by Employee)

The user is asked to enter an Employee ID. The Name and Budget of all departments managed by the employee with that ID are returned.

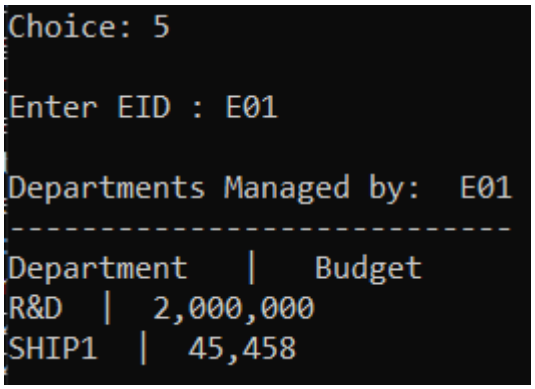

Figure 22 Departments managed by Employee

If no Departments are managed by the employee, or an non-existent employee ID is entered, nothing is returned.

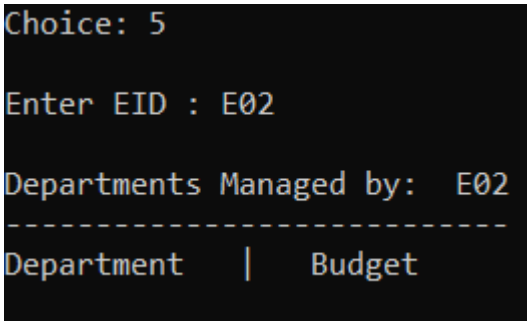

Figure 23 This employee manages no departments

4.4.6 6 (Add Manager to Department)

The user is asked to enter an Employee ID and a Department ID.
When both are entered, the Neo4j database is updated to show that that employee now manages that department.

```
Choice: 6

Enter EID : E03
Enter DID : SNR
Employee  E03  now manages Department  SNR
```

Figure 24 Add Manager to Department

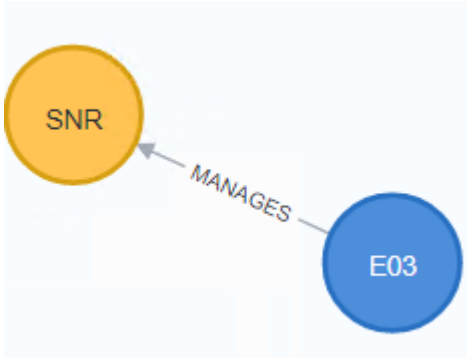

Figure 25 Database updated with relationship.

If an Employee ID and/or Department ID is entered that does not exist in the MySQL database (employees), then an error message(s) is printed, and the user is prompted to enter a new Employee ID and Department ID.

```
Choice: 6

Enter EID : E99
Enter DID : MANUF
Employee E99 does not exist
Department MANUF does not exist

Enter EID :
```

Figure 26 Non-existent Employee ID and Department ID.

If a Department ID is entered that is already managed by another Employee, an error message should be shown:

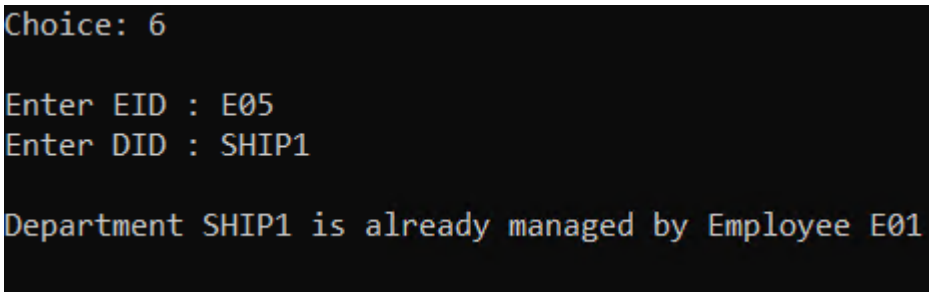

Figure 27 Department already managed by other Employee.

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

Figure 28 An Employee can manage many Departments.

4.4.7 7 (View Departments)

This option shows details of all Departments.

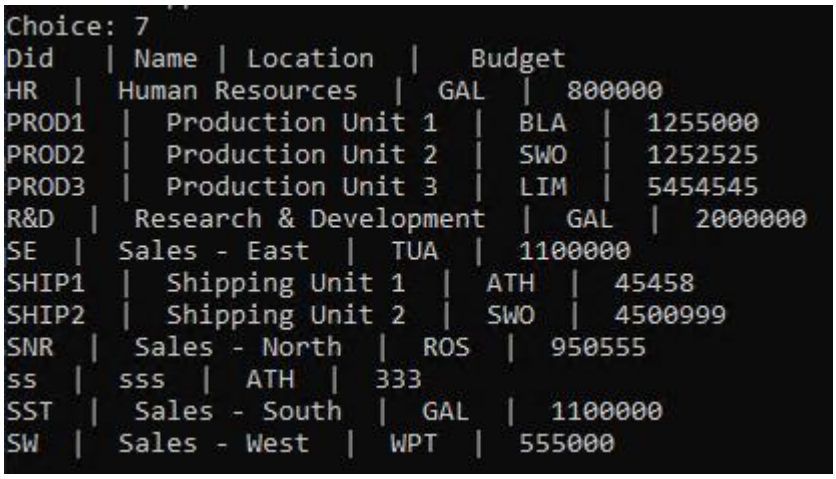

Figure 29 View Departments

The information should be read from the database **only once.**
E.g. If the user chooses 7 (View Departments) the departments are read from the database and stored in the program.

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 used.

4.4.8 x (Exit Application)

The program terminates.

4.4.9 Anything Else

The menu is shown again.