CSC2001F Assignment 4

Databases I

Introduction

In this assignment you will create and load your own MySQL database using a given SQL file, and you will write SQL queries to answer the questions below and run them against your database. Your queries must be correct for any instance of the database schema, and not just for the given sample data.

The SQL file in question is called '`classicModels.sql`' and can be found on the Vula page for the assignment.

After the queries, you are required to do one insert, delete and update on your database in (22) – (24) below. Make sure you do these only after the select queries, so that these do not affect the results of your queries in (1) – (22).

Please use the Vula forum for all questions about the assignment: emails will not be answered, on the forum all information is in one place accessible to all.

This assignment must be your own work, it is NOT a group assignment.

Marking

There are 2 marks each (1) – (12) and (22) – (24)	[30]
There are 3 marks each for (13) – (20)	[24]
There are 5 marks for (21)	[5]
There are 3 marks for overall style (e.g. queries not overly complex)	[3]
Total	**[62]**

Submission

Submit a zip file containing a separate file for each query to the automatic marker.

Call the file with your answer to question 1 'query1.sql', the one with your answer to question 2 'query2.sql', and so forth.

Make sure that you use lowercase and uppercase letters exactly as in the assignment question, as the output of each of your files will be compared with the expected output.

NOTE:

• Your ZIP file must only contain your answer files. It **should not contain a folder** containing your answer files.

• You can include single line comments in your answer files. A comment must begin with '--' and may not contain any semicolons, ";", hyphens. "-" or single quotes. "'". **Avoid pasting from PDFs or Word documents.**

• You do not need to complete all questions before trying out your answers on the automatic marker – it will just report that it cannot t find some.

• The automarker is used by all CS students and at times can be under heavy load. If there is not an immediate response when submitting your work, please be patient.

Continued

Queries

1. Show all information in the **offices** relation.
2. Show any one tuple in the **payments** relation (just one, no more).
3. Show how many tuples there are in the **orders** relation. Call the result column '**numOrders**'.
4. Show all **employees** tuples where **reportsTo** is the same as **employeeNumber**.
5. Show all information in the **payments** relation for payments exceeding 100 000, in decreasing order (i.e. from highest payment downwards).
6. Show all information in the **employees** relation for **employeeNumber**s 1188 and 1504.
7. Show the **productCode** of all **products** having their **quantityInStock** below 100, along with their total price. The total price is the **buyPrice** plus VAT (VAT is 15% of **buyPrice**).
8. What is the average **payment amount** in the database? Call the result column '**averagePayment**'.
9. In how many cities are **offices** located (how many cities have **offices** in them) ? Call the result column '**numCities**'.
10. Show all information in the **offices** relation where the **state** is missing/unknown.
11. Show the **customerNumber** and **amount** for all **payments** with a 'Q' as the 2nd character of the **checkNumber** (a check is a cheque!)
12. What **jobTitle**s exist in the database?
13. Show **productName** and **buyPrice** of the product(s) with the highest **buyPrice**.
14. Show **orderNumber, status, quantityOrdered** and **productName** for all **products** from **productVendor** 'Exoto Designs' that have **status** 'Cancelled'.
15. Show the **productCode** of all **products** that have never been ordered.
16. Show how many **employees** there are in each office (give **officeCode** and value each time). Call the value column '**numEmployees**'.
17. Show how many **customers** each employee is associated with (as **salesRepEmployeeNumber**), but only for employees who are the **salesRepEmployeeNumber** for at least 1 customer. Give **employeeNumber** and value each time. Call the value column '**numCustomers**'.
18. What was the total value of **orderNumber** 10100 i.e. the total of (**quantityOrdered * priceEach)** over all its orderlines? Call the result column '**total**'.
19. Show the **productName** of the product/s with the largest **quantityInStock**.
20. Show the **employeeNumber** of **employees** who **reportsTo** the same person as does **employeeNumber** 1166 (i.e. who have the same boss as 1166).
21. Show the **employeeNumber, firstName, lastName and reportsTo** of all **employees** who are superiors of **employeeNumber** 1625 (i.e. the person 1625 **reportsTo**, and the employee who that person **reportsTo**, ... recursively all the way up)

Insert, Update, Delete

22. Add a new **office** to the database, giving it **officeCode** 999 (meaning planned for later). This office will be in Cape Town, but no other details are known yet. Make **state** 'Western Province'.
23. Employee 1625 is superstitious. Change their employee number in the database, giving them the employee number 1 greater than the largest employee number in the database.
24. **OrderNumber** 10101 was never signed by the customer. Remove it from the database.

End