CSC2001F - DATABASE ASSIGNMENT PART 1 - INDIVIDUAL WORK - 2017

**Instructions:**

In this assignment you will create and load your own MySQL database using the given **.sql** file, 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.**

If MySQL is unable to handle any of the queries, state this clearly in you answer, with a brief reason why. If you can’t find the precise information required in any of these queries, give information as close as possible to what is being requested, explaining clearly what you are providing. If you are unsure about precisely what is being asked (requests are usually ambiguous), choose any one possible meaning and explain the interpretation you have selected.

After the queries, you are required to do one insert, delete and update on your database in (23) – (25) 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 1 place accessible to all.

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

**Marking:**

There are 2 marks each (1) – (12) and (23) – (25) [30]

There are 3 marks each for (13) – (21) [27]

There are 5 marks for (21) [5]

There are 3 marks for overall style (e.g. queries not overly complex) [3]

**Submission:**

Hand in one file on Vula Assignments. This should contain the SQL statements for each of the queries below, with a comment giving the question number and any additional information (e.g. about how you interpreted that question) above the SQL, and a table of the results it produced below the SQL.

**Hand in by 23:00 (11 pm) on Thursday 8 June 2017.**

**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.
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**s1188 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 14% of **buyPrice**).
8. What is the average **payment amount** in the database?
9. In how many cities are **offices** located (how many cities have **offices** in them) ?
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).
17. Show how many **customers** each employee is associated with (as **salesRepEmployeeNumber**), but only for employees whoare the **salesRepEmployeeNumber** for at least 1 customer. Give **employeeNumber** and value each time.
18. What was the total value of **orderNumber** 10100 i.e. the total of (**quantityOrdered \* priceEach)** over all its orderlines?
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** 1313 (i.e. who have the same boss as 1313).
21. Show the **employeeNumber** of all **employees** who are superiors of **employeeNumber** 1313 (i.e. the person 1313 **reportsTo**, and the employee who that person **reportsTo**, ... all the way up)
22. Devise a useful query of your own involving the most interesting usage of SQL you can think of. Explain clearly in a comment what it is meant to find from the database. Also explain how you know the SQL for this query is correct (showing intermediate results if necessary). Marks here will be proportional to the complexity and usefulness of the query you implement.
23. 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’.
24. Employee 1313 is superstitious. Change their employee number in the database, giving them the employee number 1 greater than the largest employee number in the database.
25. **OrderNumber** 10101 was never signed by the customer. Remove it from the database.