

# CSC2001F - Assignment 1

## 1. Instructions

- a) Create and load your own MySQL database using the SQL file `classicModels.sql`, available from the Vula assignment page and from the resources section of the site.
- b) 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.**

A diagram of the database schema may be found at the end of this document.

Please use the Vula forum for all questions on this assignment, so all information is in one place. Emails will not be answered. **Check Vula and the forum frequently** for messages on this assignment.

## 2. Marking

**There are 20 questions.** Each correct question earns 5 marks, for a total of 100.

## 3. Submission

Submit one zip file containing separate files 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.

For successful automatic marking, you must use lowercase and uppercase letters exactly as in the assignment question - the output of each of your files will be compared with the expected output.

NOTE:

- The DBMS used by the Automarker is **case sensitive**, so **all table and column names** must use uppercase and lowercase letters exactly as in the diagram on page 3 (i.e. exactly as in the `classicModels.sql` file).
- 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 can't find some.
- The automatic marker 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.
- Please **do not include your database name in the SQL statements when you submit**, as the Automarker uses its own database which will have a different name.

## 4. Questions

1. Show all information in the **offices** table.
2. Show any 1 row in the **orderDetails** table (just one).
3. Find the **customerNumber** and **creditLimit** of all customers who **creditLimit** is more than 120000

4. Show the **phone** number of all USA and UK offices.
5. What countries do the customers come from? Just show the **country** values.
6. Which cities are our offices in? Call the answer column **city**.
7. Show **productCode** values for all products that start with (uppercase) "S" and have an underscore (" \_") as their 5<sup>th</sup> character.
8. Show each **orderNumber** along with its **shippedDate**; but if there is no **shippedDate** value, then show its **requiredDate**; and if that too is missing, show its **orderDate**. Call the 2<sup>nd</sup> value **day**.
9. Give the **customerNumber** and **city** of all customers that do not yet have a **salesRepEmployeeNumber** associated with them (i.e. that value is missing), in alphabetical order of **city**.
10. A **quantityInStock** value is how many items of that product exist in the warehouse. Find the total number of items in the warehouse. Call that value **total**.
11. The **quantityInStock** is different for different products. Considering all the different products, find the average number of items that is kept in stock for a product. Call that value **mean**.
12. Show the markup percentage (rounded to the nearest integer) on each product from the **productVendor** called Red Start Diecast. Markup is the difference between **buyPrice** and **MSRP** (Manufacturer's Suggested Retail Price). Markup percentage is that difference expressed as a percentage of **buyPrice**. E.g. if **buyPrice** is 80 and **MSRP** is 120, then the markup percentage is 50 because  $(120-80)/80$  is 50%. Call that value **markup** and do not include the percent sign.
13. Show each **officeCode** along with the number of employees in that office. Call the 2<sup>nd</sup> value **size**.
14. Show each **orderNumber** with the total number (**quantityOrdered**) of *cheap* items ordered. A *cheap* item is defined as one where the **priceEach** is below R30. Call the 2<sup>nd</sup> value **total**.
15. Show each manager (value shown in **reportsTo**) along with number of employees they manage (i.e. how many report to them), but only for managers who have more than 3 employees reporting to them. Call the 1<sup>st</sup> value **manager** and 2<sup>nd</sup> column **managing**.
16. Which products have never been ordered? Give their **productName** and **productVendor**.
17. Give every **employeeNumber** along with the **city** of the office they work in.
18. Give every **customerNumber** along with the **city** where their **salesRepEmployeeNumber** works.
19. Which customers have a lower **creditLimit** than **customerNumber** 103? Give their **customerNumber** and **creditLimit**.
20. Which customers have made the lowest number of payments (paid the least often) ? Give their **customerNumber(s)**

## 5. Appendices

