

Computing Science 3461 -- Group Project

Date assigned: Feb. 27

Time due: see deadlines below

INTRODUCTION

MUC is a national company specializing in used cars. Customers buy cars from or sell cars to MUC, or they take their cars to MUC for repairs. The managers of MUC have hired you to design and implement a database according to the following requirements.

REQUIREMENTS

1. *Customers and Transactions*

For each customer, the following information is required: a unique customer id, first name, last name, address, two phone numbers (cell and other), and a list of possible transactions. A transaction is recorded when a customer takes their car to MUC for repair. For each transaction, the following information needs to be stored: a unique transaction number, amount the customer was charged, the date of the transaction, and the car that was repaired.

2. *Cars*

For each car, the following data need to be stored: a unique id number, plate number, province, color, make, model, year of manufacture, and type (e.g., sedan, van, etc.). Each car has a unique combination of values for plate number and province. Some of the cars are owned by MUC and are listed for sale to customers. For each of those cars, the following information is required: listed price, kilometers, the customer from whom the car was last bought, the price it was bought for, and the date on which it was bought. For each car sale, the following information is required: the car sold; the customer to whom the car was sold, the amount of money for which it was sold, and on what date; the customer from whom the car was last bought, the amount of money for which it was bought, and the date on which it was bought.

3. *Parts, Suppliers, Orders*

MUC wants to keep information about parts, suppliers, and orders. Each part has a unique part id, a price, and a description. For each supplier, the id, name, email address, and list of telephone numbers are required. The id's of suppliers are unique. Suppliers have no telephone numbers in common. For each order, the following information is needed: a unique order id, date, supplier id, and a list of parts together with their quantities.

4. *Monthly Expenses*

MUC wants to keep track of the following monthly expenses: cost of electricity, heat, water, and the rent. The rent is fixed within the same year.

YOUR TASKS (see further below for deadlines)

1. **Form groups:** This is a group project. You are expected to form groups of five (in fact a couple of groups will have four members). Each group will submit in Brightspace, by the deadline specified below, the names and A-numbers of the group members, one per line. If you cannot find a group then again you are expected to submit before the deadline stating that you cannot find a group.
2. **Design a good relational database schema:** Use the ideas of the relational data model to design a good database schema according to the above requirements. Make sure to indicate primary and foreign keys. Please note that the next paragraph already gives further information about the part of the database pertaining to parts, suppliers and orders. *Some hints:* It is a good idea to first design an EER diagram according to the above requirements. Then map your EER diagram to a relational database schema. You might need to modify a couple of schemas to make sure that all schemas are in 3NF.
3. **Create some tables and fill with data from json files:** The process of implementing the entire database schema could be tedious once the design has been completed. *You are required to create (in MySQL) only tables pertaining to parts, suppliers and orders as follows.*

The file `parts_100.json` contains 100 json parts. The file `parts_table.sql` contains the definition of the MySQL table parts. The bash script `j2sql_parts.sh` contains commands that read the 100 json parts and insert corresponding tuples into the MySQL table parts. Your first task here is to simply change the username/password/database names in `j2sql_parts.sh` and execute that script in your account so that it creates and fills with data the parts MySQL table in your account. The file `suppliers_100.json` contains a json *array* of 100 suppliers. The file `orders_4000.json` contains 4000 json orders. Your second task here is to (a) create a file `make_tables.sql` containing definitions of the rest of the MySQL tables pertaining to suppliers and orders; (b) create and execute a bash script file `j2sql_supp+order.sh` containing commands that read the 100 json suppliers and the 4000 json orders, and insert corresponding tuples into the MySQL tables defined in your `make_tables.sql`. *Make sure that your `j2sql_supp+order.sh` begins by assigning your own username/password/database values to variables and uses these variables in subsequent commands, as opposed to your actual username/password/database values.* Note that your script might rely on some other script(s) and that you are required to submit all files/scripts/programs pertaining to this task.

All the above will be tested on the Linux server dbcourse.cs.smu.ca.

4. **Write and present a web application** that allows the user to perform the following operations:
 - *Show table:* the user enters the name of a table, and the application prints the contents of the table.
 - *Add new supplier:* the user enters the attribute values pertaining to a supplier and the application inserts the corresponding rows in the database. The program should consider the case where the new supplier cannot be inserted.
 - *Annual Expenses for Parts:* the user enters a start year and an end year, and the application shows the total amount of money paid for parts in each of the selected years.

- *Budget Projection*: the user enters a number N of years and an inflation rate value (e.g, 2%), and the application shows the total amount of money that would be spent in each of the next N years by applying inflation (starting after the most recent full year, 2022).

The web application will be presented and tested in class. In addition, you are required to prepare a file **report.pdf**, containing a short description of how your web application is structured (what the main files are and how they work together). In the class presentation, the group will demonstrate that each of the four main operations described above work properly. Data about parts, suppliers and orders should already be present in the tables at the beginning of your presentation.

In case you can achieve the above functionality using a language other than PHP you are welcome to do so.

5. **Confidential Peer Evaluation**: Each group member will write a short evaluation report on the work of each member of their group. The evaluation could be qualitative and make comments about the general performance and/or specific skills of each group member.

WHAT TO SUBMIT and DEADLINES (strict)

- 02% In Brightspace, Group Members. **One** submission per group.---> **Mar 02 at 4pm**
- 28% In Brightspace, the file **design.pdf** containing the relational database schema. **One** submission per group. ---> **Mar 14 at 4pm**
- 26% In the Linux server, a zip folder, called **scripts.zip**, containing: make_tables.sql, j2sql_supp+order.sh with any supporting scripts. **One** submission per group. In Brightspace, the full path name to your Linux zip folder must be specified (again **one** submission per group). I will use in the server the command **unzip scripts.zip** to unzip your zip folder, so you are expected to use the server command zip to make scripts.zip. ---> **Apr 05 at 9am**
- 44% Report on the PHP application and in-class presentation of the PHP application. In Brightspace,, submit the file **report.pdf**. **One** submission per group. ---> **Apr 05 at 9am**
- Peer evaluation: In Brightspace, submit the peer evaluation report (every group member will submit their own file). ---> **Apr 06 at 9am**
- Marks will be deducted if there is evidence that your contribution to the project was not satisfactory and/or you fail to send me a confidential email as specified above.