CSCI 3901 Assignment 5

Due date: 4pm Monday, December 6, 2021 in git.cs.dal.ca at
https://git.cs.dal.ca/courses/2021-fall/csci3901/assignment5/xxxx.git
where xxxx is your CSID (this repository already exists, so clone it and then add your code to it).

Problem 1

5 marks – Referencing the sales database from the database lab, provide SQL queries that answer the following questions:

a) Which customers are in a different city than their sales representative?
b) Which orders included sales that are below the manufacturer's suggested retail price (MSRP)?
c) Which product line has the highest profit margin in 2003. Include the profit margin. The profit of an item is the sales of the item less the cost of the item. The profit margin is the profit divided by the total base cost.
d) List the 5 employees with the highest sales in 2004. Include their total sales values and ensure that the top seller is first in the list.
e) Which employees had the value of their 2004 orders exceed the value of their 2003 orders?

Problem 2

Goal

Access SQL through Java. Gain some exposure to XML.

Question

You work for the Mini-Me Toy Car company. Management periodically wants a summary of the company's operation over a period of time.

Your job is to extract the summary information from the database. You will store the summary information in a file that follows an XML format. Someone else will then use XML tools (notably XSLT) to convert your information into something that management will review.

Input

Your program will obtain the following information from the keyboard in the following order on separate input lines:

- The starting date for the period to summarize
- The ending date for the period to summarize
- The name of the file for the output (full path specified with the file name)

All dates will be in a YYYY-MM-DD format.

Output

Your program will send all of its output to the specified file.

The data that you extract will be in 2 categories:

1. Manager information
    a. Report, for each manager, the manager's name, the city of the manager's office, the number of employees who report to the manager, the number of customers who are served by staff who report to the manager, and the total sales by staff who report to the manager
2. Product line information
    a. Report, for each product line, its name, its description and then for each customer who ordered some item of the product line, report the customer's name and the total value of this product line's order by this customer in the given period

In all of the reporting, do not report any employees, customers, orders, or offices that have not had any interaction over the reporting period.

Your output file will be in an XML format. XML uses a set of tags to surround data to let you know what the data is. Some tags can be nested in other tags.

We will use a simple version of XML. The first line of your XML file should provide information on the version of XML to use. The following line will be sufficient:

```
<?xml version="1.0" encoding="UTF-8" ?>
```

Following this first line, we get a set of nested tags to store the data. The starting tag has the format <…> and the matching ending tag has the format <…/> (differing by the ending slash) where … is the tag name. The outermost tag is year_end_report

Here is a description of the correct nesting (in a data type definition (DTD) format):

```
<!ELEMENT year_end_report (year, manager_list, product_line_list) >
<!ELEMENT year (start_date, end_date) >
<!ELEMENT manager_list (manager*) >
<!ELEMENET manager (manager_name, manager_city, staff, customers, sales_value) >
<!ELEMENT producdt_line_list (product_line*)>
<!ELEMENT product_line (product_line_name, product_line_description, customer*) >
<!ELEMENT customer (customer_name, order_value)>
```

All items without an ELEMENT line are of type #PCDATA (if that matters to you). The address in customer is just the first address line in the database.

In an XML file, the spacing doesn't matter. I encourage you to use spacing and tabs to make the XML file readable by a person.

Information on XML can be found at w3schools.com.

Sample output (just one entry shown for each section, for brevity)…data is not representative of the database's contents:

```
<?xml version="1.0" encoding="UTF-8" ?>
<year_end_report>
        <year>
                <start_date> 2003-01-01 </ start_date>
                <end_date> 2003-12-31 </ end_date>
        </year>
        <manager_list>
                <manager>
                        <manager_name> Diane Murphy </manager_name>
                        <manager_city>  Paris </manager_city>
                        <staff>  3 </staff>
                        <customers> 10 </customers>
                        <sales_value> 145258.23</sales_value>
                </manager>
        </manager_list>
        <product_line_list>
                <product_line>
                        <product_line_name> Classic Cars </product_line_name>
                        <product_line_description> Longer description of classic cars here
</product_line_description>
                        <customer>
                                <customer_name> Atelier graphique</customer_name>
                                <order_value>23045.84</order_value>
                        </customer>
                        <customer>
                                <customer_name> Signal Gift Stores </customer_name>
                                <order_value>169248.55</order_value>
                        </customer>
                </product_line>
        </product_line_list>
</year_end_report >
```

Constraints

- You may use any data structure from the Java Collection Framework.
- Write your solution in Java. The solution code must be your own.
- Use the mysql JDBC connection for Java.

- If in doubt for testing, I will be running your program on timberlea.cs.dal.ca. Correct operation of your program shouldn't rely on any packages that aren't available on that system.

Notes

- Use SQL vs Java as you deem best.
- Be sure to document your approach and any resources that you use.
- Look at where the bulk of the marks are in the marking scheme to help focus your efforts.
- You are allowed to use an XML library if it is running on timberlea. However, this XML is simple enough that you can also generate it with your own code.
- You can run your queries against the csci3901 database on db.cs.dal.ca I will also make the sql file for the database available to you so that you can create your own copy of the database.
- You are expected to submit
    - Your Java code
    - External documentation
    - An argument as to why your solution is ready to be deployed

Marking scheme

- Documentation (internal and external) – 3 marks
- Program design, organization, and style – 2 marks
- Proper XML file format, including human readability – 3 marks
- Correct extraction of manager information – 3 marks
- Correct extraction of product line information – 3 marks
- Quality of the argument on why your solution is ready to deploy – 2 marks

Problem 3

Goal

Practice some database design.

Question

The database from problem 2 isn't complete. Although it stores our orders, it doesn't have enough information to complete our billing to customers. Your task is to design changes to the database to augment what it can do.

Specifically, your design change should let us:

1. Create promotions by office as a percentage discount to the order value. Promotions are identified by a promotion code tied to the order. We will want to track the uptake of promotions among the orders that use it.

2. Add shipping costs to orders. We will have a small (3-10) number of shipping companies that we use. Each shipping company will charge a percentage of the order value as the shipping cost. The shipping cost is set when the order is made. Meanwhile, the shipping company can change the percentage over time as we renegotiate our contract with the shippers. For those negotiations, we will want to report on the amount of activity (number of orders, total order value) sent through each shipper in a given period of time.
3. Add taxes to the invoices. Each office will have zero or more taxes that will be applied to any bill issued to a customer whose representative is in the office. The tax amounts are set when the order is made. The tax percentages can change over time.

When calculating the final value of an order, we first apply any promotional discount. Next, we add the shipping costs and end with the taxes on the order.

For this question, submit

1. An updated entity-relation diagram with your modifications. The diagram should be in 3NF or better.
2. An SQL file that, when run on the mysql database, would make whatever changes are needed to the database to realize your design modifications.
3. An SQL file that, when run on the mysql database that has your changes, will populate the database with 3 sample shipper records and one tax record for each office except the Paris office, which will have two tax records.
4. A Java class called "OrderManager" that includes two methods:
    a. checkout(int orderNumber, int shipperId) that is called once an order is set in the database and when we now need to apply the promotional discounts, shipping costs, and taxes. If the shipperId is 0 then the customer picked up the order and there is no shipping cost; otherwise, the shipperId will match a shipper id that you include in your database update.
    b. orderValue(int orderNumber) that returns the total cost to the customer of the given order number.

Marking scheme

- Database design to handle the new information. Looking for correctness, natural fit with what already exists, minimizing the changes, flexibility, extensibility, and, of course, normalization – 6 marks
- Sample SQL to create the structures and populate tables – 2 marks
- Checkout and orderValue methods – 4 marks