

CSCI 3901 Assignment 5

Due date: 11:59pm Monday, November 18, 2019 in git.cs.dal.ca at

<https://git.cs.dal.ca/courses/2019-fall/csci-3901/assignment-5/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:

- Which customers are in a different city than their sales representative?
- Which orders included sales that are below the manufacturer's suggested retail price (MSRP)?
- List the top 5 products for 2004 with the highest average mark-up percentage per order. Include the mark-up. The mark-up is the ratio of (sale price – cost) and cost.
- List the top 3 employees with the greatest average diversity of products in their orders.
- What is the average time needed to ship orders from each office in 2005, relative to the order date?

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:

- The starting date for the period to summarize
- The ending date for the period to summarize
- The name of the file for the output

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 3 categories:

1. Customer information
 - a. Report the customer name, address, number of orders in the period, and total order value
2. Product information
 - a. Report, for each product line, the product line name and for each product in the product line report the name, vendor, units sold, and total value of products sold
3. Employee information
 - a. Report, for each employee, their name, office, number of customers active in the period, and total order value

In all of the reporting, do not report any customers or products who 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_summary (year, customer_list, product_list) >
<!ELEMENT year (start_date, end_date) >
<!ELEMENT customer_list (customer*) >
<!ELEMENT customer (customer_name, address, num_orders, order_value) >
<!ELEMENT address (street_address, city, postal_code, country) >
<!ELEMENT product_list (product_set*) >
<!ELEMENT product_set (product_line_name, product*) >
<!ELEMENT product (product_name, product_vendor, units_sold, total_sales) >
<!ELEMENT staff_list (employee*) >
<!ELEMENT employee (first_name, last_name, office_city, active_customers, total_sales)>
```

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.

As an example to read this information, the tags `year_end_summary` must contain nested tags for each of `year`, `customer_list`, `product_list`, and `staff_list`. The tag `customer_list` will

contain zero or more tags with name “customer”, as identified by the * after the “customer” tag in the ELEMENT clause.

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):

```
<?xml version="1.0" encoding="UTF-8" ?>
<year_end_summary>
  <year>
    <start_date> 2003-02-12 </ start_date>
    <end_date> 2003-02-19 </ end_date>
  </year>
  <customer_list>
    <customer>
      <customer_name> Rovelli Gifts </customer_name>
      <address>
        <street_address> Via Ludovico il Moro 22 </street_address>
        <city> Bergamo </city>
        <postal_code> 24100 </postal_code>
        <country> Italy </country>
      </address>
      <num_orders> 1 </num_orders>
      <order_value> 52151.81 </order_value>
    </customer>
  </customer_list>
  <product_list>
    <product_set>
      <product_line_name> Planes </product_line_name>
      <product>
        <product_name> 1980s Black Hawk Helicopter </product_name >
        <product_vendor> Red Start Diecast </product_vendor >
        <units_sold> 36 </units_sold>
        <total_sales> 4825.44 </total_sales>
      </product>
    </product_set>
  </product_list>
  <staff_list>
    <employee>
      <first_name> Pamela </first_name>
      <last_name> Castillo </last_name>
```

```
        <office_city>Paris</office_city>
        <active_customers> 1 </active_customers>
        <total_sales> 52151.81 </total_sales>
    </employee>
</staff_list>
</year_end_summary >
```

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 bluenose.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 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 customer information – 4 marks
- Correct extraction of product information – 4 marks
- Correct extraction of staff information – 4 marks
- Design elements of your program that would make it easy to report additional information for customers, products, or staff with minimal code changes (and avoiding cut-and-paste of code segments) – 2 marks