Assignment 5

Team Members:
1. Dhrumil Amish Shah

Date:

2021-03-28

—

Subject:

Software Development Concepts

—

Professor:

Matthew Amy

In project (Six files)

- **TestCommandLineConnection.java file** (Starting Point 1– Contains the main method):
    - Takes input through command line arguments. First argument is start date (i.e., args[0]), second argument is end date(i.e., args[1]) and last argument is output file name(i.e., args[2]).
- **TestUserConnection.java file** (Starting Point 2– Contains the main method):
    - Takes start date, end date and output file name from a user.
- **SummaryDocument.java file** (Connects to the server and prints the data in the output file)
- **CustomerInformation.java**, **ProductInformation.java** and **SupplierInformation.java files**. (Contains code to query data, extract the data and create the document)

Code specifics

- To run the code, add CSID as user id and BannerID as password in the SummaryDocument.java file (Line 222). These can also be taken from a user as additional arguments.
- I used the API provided by java from packages 'org.w3c.dom' and 'javax.xml' to write XML in a file. I referred to the first answer from the below Stack Overflow post.
    - https://stackoverflow.com/questions/7373567/how-to-read-and-write-xml-files
- I used "null"(null string) to mark a tag empty in an XML document where there is a null value in a table cell. It is not possible to add a null value in the tag as it throws NullPointerException. Instead, I am adding a null string. I verified this by trying it out in the code and from the below Stack Overflow post.
    - https://stackoverflow.com/questions/22522558/create-empty-text-node-in-dom-document

Why my solution is ready to be deployed.

- I tested the robustness of all three queries in MySQL Workbench before implementing them in the program. Also, to verify the output, I did the following:
    - For customers information:- Checked customers data for some customer ids and verified whether the answer contains the correct customer name, street address, city, region, postal code, country, number of orders and total dollar value of orders in that period.
    - For products information:- Checked products data for some categories and verified that a category contains the correct category name and correct products list with each product having the correct product name, supplier name, units sold, and total dollar value of the product sold in that period.
    - For suppliers information:- Checked suppliers data for some suppliers ids with products sold in that period and verified that the answer contains the correct supplier name, street address, city, region, postal code, country, number of orders and total dollar value of orders in that period.
- Wrote the code such that it follows the proper object-oriented design principles. Also, divided the code into custom classes where each class have its standalone purpose for better readability, extensibility, and understandability.
- All the methods and classes in the code are tested thoroughly to avoid any unwanted exceptions or errors.
- Well written code to ensure input validation tests, boundary condition tests, control flow tests and data flow tests pass. These tests provide that the code is robust and does not break in false conditions like problems related to database connectivity, incorrect dates, incorrect file handling and writing, query extraction issues and so on.

- Instead of creating the XML document on my own and adding it to a file, I used the classes provided by 'javax.xml' and 'org.w3c.dom' for flexibility and extensibility. In future, if there are additional fields reported for customers, products, or suppliers, one can directly append the new attributes to the existing and append them to the file with just a few lines of code.
- Documented the code in a way that readers can easily understand the underlying functionality. Written it according to the style guide suggested by Oracle (https://www.oracle.com/in/technical-resources/articles/java/javadoc-tool.html). Using the Javadoc command, API documentation can be created easily.