CSCI: 3901 Assignment 5
Winter 2020

External Documentation:

Why my solution is ready to be deployed:

- The program does not report any customers, products, or suppliers who do not have any interaction over the reporting period provided by the user.
- Based on the requirement, the program can fetch the reports in XML format, which can later be used by someone in the management.
- If there is a requirement to fetch different columns from the DB, then queries can be updated. Accordingly, the resultSet column parameters can be modified by replacing the code segments.

Overview:

This program SQLMain extracts the summary information from the database and stores the summary in XML format. It has been written in Java language without using the collections framework.
A summary of the Northwind food distribution company information requirements is in the CSCI 3901 course assignment #5 problem 5 information in the course's git repository "https://git.cs.dal.ca/courses/2020-winter/csci-3901/assignment-5/souvik.git".

The solution consists of class "SQLMain" and it includes the following methods and sub:

- String isValidDate(String data):- Here, the date sent by the user is checked by using DateTimeFormatter class, and in case of invalid data type, this function is recursively called and prompts the user to enter a valid date. Upon receiving a valid date, the date is returned to the calling method.

- Void main():- The main method asks the user for startDate, endDate, and fileName and checks for its validity. Upon validation, the particular SQL queries are fired, and the resultsets are iterated to form the XML in filename using DocumentBuilderFactory and TransformerFactory class.

Predefined classes used as part of the program:

- No predefined classes were used as part of the program.

Predefined Interfaces used as part of the program:

- No predefined interfaces were used as part of the program.

Predefined Libraries and jar files used as part of the program:

We have used mysql-connector-java-8.0.19.jar file for establishing jdbc connectivity.

- import org.w3c.dom.Document;
- import org.w3c.dom.Element;
- import javax.xml.parsers.*;
- import javax.xml.transform.*;
- import java.io.File;
- import java.sql.*;
- import java.time.format.DateTimeFormatter;
- import java.util.*;

References:

- Referred https://git.cs.dal.ca/courses/2020-winter/csci-3901/lab-6/souvik.git for SQLMain.java and MyIdentity.java

Files and external data:

The program consists of two file:

- SQLMain.java - main for the program that calls the main method and fires the query.
- MyIdentity.java - This Class helps in establishing JDBC connectivity by fetching user credentials.

Data structures and their relations to each other:

- No data structures used as part of this program.

Assumptions:

- We have assumed that Northwind food distribution database column names are the same.
- Date Range in every entry follows the YYYY-MM-DD format.

Choices:

- In the case of invalid input, the user is asked to enter the correct data.
- Empty tags are appended in the XML file when an entry is null for the particular field.

Fundamental algorithms and design elements:

For String isValidDate(String data) method:

- The method checks the input validations of the date range passed, and returns recursively ask the user to enter correct data type format.

- Here the date sent by the user is checked by using DateTimeFormatter. Upon receiving a valid date, the date is returned to the calling method.

For void main() method:

- The program asks the user for startDate, endDate, and filename using scanner class.
- The dates are validated by calling the isValidDate method.
- Connection, Statement, and ResultSet classes are used to s object to build up SQL queries, and the ResultSet data structure is used to receive results by firing the SQL queries.
- Properties structure is used to provide encapsulation for the user's identity.
- MyIdentity class is called, and the parameters like username "user," password, and dbName are fetched.
- "Class.forName("com.mysql.cj.jdbc.Driver");" loads the MySQL driver.
- Connection class object connect stores the instance information for setup to the DB.
- DocumentBuilderFactory class is used to obtain a parser that produces DOM object trees from XML documents.
- The element class is used to create tags for the XML file.

```
Element numProducts = document.createElement("num_products");
```

- Based on each query, resultSet's are iterated and the entries are converted into xml tag format.
- In-Case of resultSet entry returning null for a particular column, the particular entry is ignored, and empty tags are created for that particular entry.

```
String region_Check = resultSet3.getString("region");
if (region_Check != null) {
    region.appendChild(document.createTextNode(region_Check));
}
```

- TransformerFactory class is used to write the data in XML format
- The resultSets statements and connections are closed to free up resources.

Limitations:

- In case tables in the database adds new fields, then this code will need modification in order to retrieve new columns.