Assignment 5 – SQL With Java

Overview:

The problem statement states that, a summary for customers, products and suppliers needs to be fetched from the database for the given duration and showcase in xml format so that it is human readable.

Classes:

- **MainUi**: Main class that takes inputs from the user and invokes the method in inputs class where required inputs are taken from the user.
- **Inputs**: Class that validates the user inputs and invokes the method in ConnectToDB class where connectivity with established with the database.
- **ConnectToDB**: Class that takes credentials from Identity class and invokes the database connectivity method wherein a connection is established with the database.
- **MyIdentity**: Class that holds the database name, username and password.
- **Period**: Class that creates a Dom document and appends the start date and end date given by the user along with the year_end_summary parent tag (starts a tree structure).
- **Customer**: Class that fetches the customer summary from the database as per the query and appends the data into the existing document.
- **Product**: Class that fetches the product summary from the database as per the query and appends the data into the existing document.
- **Prod**: Class that holds the properties of the product query(supplier name, units_sold and value of the product).
- **Supplier**: Class that fetches the supplier summary from the database as per the query and appends the data into the existing document.
- **XmlConfig**: Class that holds the DOM configuration for appending and formatting the data in xml format.

Functions:

- **Inputs class:**
    1. **inputs :** Method that validates the user inputs. The start date and end date should be in "yyyy-MM-dd" format only. The start date should be greater than end date. And all the parameters cannot be null or empty.
- **ConnectToDB:**
    1. **connectDB :** Method that establishes a connection with the database provided by the user along with his credentials and invokes the required methods where summary is fetched.
- **MyIdentity:**
    1. **setIdentity :** Method that holds the database name, the username and password to access the database.
- **Period:**
    1. **setRoot:** Method that sets the root of the xml tree and appends the start date and end date given by the user as a child to the document model.

- **Customer:**
    1. **customerquery:** Method that accesses the database and fetches the summary related to the customer, based on the start and end date provided by the user. It simultaneously appends the data into the existing doc object.
- **Product:**
    1. **productquery:** Method that accesses the database and fetches the summary related to the products, based on the start and end date provided by the user. It simultaneously appends the data into the existing doc object.
    2. **organize_data:** Method that accesses the result set obtained from the database and organizes data in the form that needs to be placed in xml format. Internally uses two maps, one with key values as category name, values as products and other with key values as product name and values as product details from the Prod class, fetched from the database.
- **Supplier:**
    1. **supplierquery:** Method that accesses the database and fetches the summary related to the supplier, based on the start and end date provided by the user. It simultaneously appends the data into the existing doc object.
- **XmlConfig:**
    1. **xmlDoc:** Method that creates accesses the api's from Document Builder class and creates a document which stores the data .
    2. **xmlTrans:** Method that takes the created document and places the data in xml format with proper indentation in human readable format.

Approach to the problem:

- The problem is initially analyzed and divided into different parts.
- The basic requirement of the problem is to store the summary data from the database in xml format using any of the parser.
- The requirement on query output is analyzed and worked on the queries to fetch the summary data from the database using workbench.
- Once the queries are fetching the required output, a study is done on the xml parsers.
- A suitable parser has been chosen to format the obtained result set for each query and place in xml format as required.
- Then, respective classes are designed, each of the one holding its own functionality.
- Based on the DTD, a structure is designed for each of the operation and data structures are used to store the data and fetch in required format, just like the one it has to be showcased in xml.
- Code is completely modularized so that it will be easy to understand the workflow.
- Internal comments are clear and concise so that the code is understandable.
- Each of the module is tested as per the functionality and compared with the required output.
- Required exceptional handling is done so that and exception is thrown if there is an error related to that module.

- Made sure that the obtained output is having the exact indentation and it is in human readable format.

Observations:

The address value from the database is placed in different lines, so the query is designed in such a way that all the next line, tab spaces and enter values will be replaced by empty string to maintain the indentation and tree design of the xml format being extracted.

The region and postal code values is not updated in the database for few instances, in that case, an empty string is shown in the xml format so that it is completely understandable and human readable.

Limitations:

Date is following a single format “yyyy-MM-dd”.

Assumptions:

The file name will be provided with extension.

None of the inputs can be null or empty.

All the inputs are trimmed, and extra spaces are removed.

References:

The database connectivity part of the code is mostly from the lab 6 example program as it is holding all the servers and required data.

For understanding xml : https://www.w3schools.com/xml/

For understanding how the DOM parser works: https://mkyong.com/java/how-to-modify-xml-file-in-java-dom-parser/

For formatting the output : https://stackoverflow.com/questions/18607343/how-to-format-the-xml-in-java-dom-parser