CSCI – 3901: Assignment 5

Overview

Implement a Java program to work with MYSQL and get specific data as per the requirement. This prototype uses Java and MYSQL to manipulate the data from MYSQL to a specific XML file which can be used as a backup file as well as a data review document.

Requirements:

The requirements for this problem from the user's perspective:

1. Users would be able to enter the starting date and the ending date:
   - On the first line of the console user will be asked to enter starting date from where we need to generate a file.
   - On the second line user will be asked to enter the last date till the information is needed.
   - Validate the format in "YYYY-MM-DD".

2. Users would be able to select the file name.
   - In this line user would be able to specify the output file name.
   - In this case, we file extension is in fixed XML format so we will reject the file name with any specified extension.

The requirements for this problem from the developer's perspective:

1. Generation of queries to fetch the data of customers, products, and stores.
   - The test of the queries and modification based on the required output format is the primary goal.
   - Making these queries more compact is the second milestone.
2. Make a JDBC connection to the database.
   - This required the database to be accessed in such a way that minimum information can be displayed.
3. Write the code for the XML generation in the desired format.
   - This format follows the format needed.

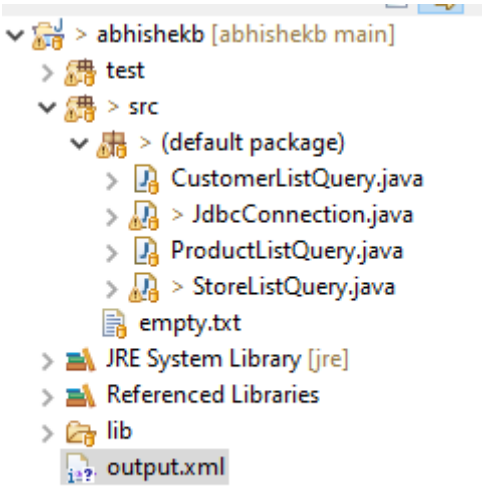
```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
Bind to grammar/schema...
<activity_summary>
    <time_span>
        <start_date>2017-03-12</start_date>
        <end_date>2017-03-13</end_date>
    </time_span>
    <customer_list>
        <customer>
            <customer_name>Jonna Brown</customer_name>
            <address>
                <street_address>1 Spring Drive </street_address>
                <city>Mahopac</city>
                <state>NY</state>
                <zip_code>10541</zip_code>
            </address>
            <order_value>1394.9907</order_value>
            <bicycles_purchased>1</bicycles_purchased>
        </customer>
        <customer>
            <customer_name>Yevette Elliott</customer_name>
            <address>
                <street_address>182 Euclid Street </street_address>
                <city>Jackson Heights</city>
                <state>NY</state>
                <zip_code>11372</zip_code>
            </address>
            <order_value>479.9920</order_value>
            <bicycles_purchased>1</bicycles_purchased>
        </customer>
        <customer>
            <customer_name>Yevette Elliott</customer_name>
            <address>
                <street_address>182 Euclid Street </street_address>
                <city>Jackson Heights</city>
                <state>NY</state>
                <zip_code>11372</zip_code>
            </address>
            <order_value>1042.1310</order_value>
```

Sets of design:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<activity_summary>
        <time_span>
                <start_date>2017-03-12</start_date>
                <end_date>2017-03-13</end_date>
        </time_span>
        <customer_list>
                <customer>
                        <customer_name>Jonna Brown</customer_name>
                <address>
                        <street_address>1 Spring Drive </street_address>
                        <city>Mahopac</city>
                        <state>NY</state>
                        <zip_code>10541</zip_code>
                </address>
                <order_value>1394.9907</order_value>
                <bicycles_purchased>1</bicycles_purchased>
                </customer>
        </customer_list>
        <product_list>
                <product>
                        <product_name>Electra Townie Original 7D EQ - Women's - 2016</product_name>
                        <brand>Electra</brand>
                <category>Cruisers Bicycles</category>
                <store_sales>
                        <store_name>Baldwin Bikes</store_name>
                        <units_sold>85</units_sold>
                </store_sales>
                </product>
        </product_list>
        <store_list>
                <store>
                        <store_name>Baldwin Bikes</store_name>
                        <city>Baldwin</city>
                        <employee_count>3</employee_count>
                        <customers_served>1</customers_served>
                        <avg_sales_per_customer>20414.87400000</avg_sales_per_customer>
                <customer_list/>
                </store>
        </store_list>
</activity_summary>
```

Files and Data

In this code, I used 4 classes that use the SOLID principle of design where every functionality is separated into a different class.



This code mainly focused on the 3 queries which gather the information from the database and display it on the XML file.

Work-flow

I divided the workflow into 3 steps:

1. Get the information in the SQL workbench to confirm the output needed.
2. Get that output on the console to verify it.
3. Generate the XML file for that output.

Assumption

- Data is present in the database.

Explanation: (step by step)

This is a Java program that performs database operations, retrieves data from a MySQL database, and generates an XML file based on the retrieved data. Here is a brief summary of the main steps performed by the program:

- The program starts by defining the database connection parameters and the list of valid file extensions.
- The user is prompted to enter a start date, an end date, and a file name. The input is validated to ensure that the dates are in the correct format and that the file name does not have an extension.
- The program connects to the MySQL database using the JDBC driver and executes a SELECT query to retrieve customer data within the specified time range.

- The program uses the retrieved data to build an XML document using the Java DOM API. It creates a root element called "activity_summary", and adds child elements for "time_span" (containing the start and end dates), and "customer_list" (containing a list of customer elements).
- For each customer retrieved from the database, the program creates a new "customer" element and adds child elements for the customer's name, address, and order value.
- Once all customer elements have been added to the XML document, the program creates a new file with the specified name and writes the XML content to the file.

Limitations

- The program creates a new connection to the database for each query, which is inefficient and can lead to performance issues. It is recommended to use a connection pool to reuse existing connections.

Benefits:

- All the necessary import statements are present at the beginning of the code.
- The code defines a constructor and the main method to execute the program.
- The required input is taken from the user using Scanner.
- The input validation is done to ensure that the user enters the correct input.
- The code connects to the database using the DriverManager.getConnection method.
- The query is executed, and the results are stored in the ResultSet object.
- The XML file is created using the DocumentBuilder and Document classes.
- The XML elements are created and added to the XML document using the createElement and appendChild methods.
- The data from the ResultSet is added to the appropriate XML elements.
- The XML document is written to a file using the Transformer class.

Overall, the code uses standard Java classes and follows the best practices for connecting to a database, querying it, and generating an XML file from the results. Hence, it is implementable.