

Assignment 5

Team Members

Meet Nitinbhai Patel
(B00899516)

Subject:

Software Development
Concepts

Professor:

Mike McAllister

Problem 2 - Access SQL through Java. Gain some exposure to XML

Overview

- This is scenario-based problem where I am working in Mini-Me Toy Car company. The Top Management in the organization periodically wants a summary of the organization's operation over a given period. My job is to extract the summery information from the database. I must store that summery in the xml format.
- The data that I need to extract from the database will be divided into 2 categories:
 - Manager Information
 - Product Information
- It must extract the above information over a given period. The generated xml file will then be used by the management for review.

Files and external data

The program contains following files

1. Main.java

- This contains the main method
- It takes three inputs from the user
 - startDate
 - endDate
 - outputFile Name

2. CreateSummaryReport.java

- It will valid the inputs entered by the user such startDate, endDate, outputFileName and creates the xml file

Methods

- boolean datesValidation(String startDate, String endDate)
 - It is used to Validate the dates
 - It will return false if startDate or endDate is null
 - It will return false if endDate is after the startDate
 - It will return true if the date is valid
- boolean outputFileNameValidaton(String outputFile)
 - It will validate the output file name and also creates the output file
 - It will return false if file name is empty
 - it will return false if file name is empty or don't have extension .xml
 - if there is any error while creating file it will retrun false

- Document createSummaryDoc(String startDate, String endDate)
 - It will create the summary doc over a given period of time.
 - It will basically create the following tags
 - year_end_report
 - start_date
 - end_date
 - year
 - manager_list
 - product_line_list
- boolean generateReport (String startDate, String endDate, String outputFile)
 - it will generate the report
 - it will return false if dates are not valid
 - it will return false if output file name is not valid or it is not able to create file object
 - It will finally create xml file

3. ManagerInformation.java

- This class is used to get manager information from the database

Methods

- Open ()
 - it is used to open the connection with the database
- Close ()
 - it is used to close the connection with the database
- Element getManagerInformationElement(String startDate, String endDate)
 - this will return the information about manager
 - This will create the following tags adds the information from the database
 - manager_list
 - manager_name
 - manager_city
 - staff
 - customers
 - Sales_Values
 - Manager

4. ProductLineInformation.java

- This class is used to get product information from the database

Methods

- Open ()
 - it is used to open the connection with the database
- Close ()
 - it is used to close the connection with the database

- Element getProductLineInformationElement(String startDate, String endDate)
 - It will return the information about product line
 - This will create the following tags and adds information from the database
 - product_line_list
 - product_line_name
 - product_line_description
 - customer_name
 - order_value

Assumptions

- Users will have to enter the startDate and endDate in YYYY-MM-DD format
- Users will have to enter the file name with .xml extension
- Users will always have to open the XML file in the browser for better readability

Limitations

- The program doesn't check for invalid months (month not in range of 1-12) or invalid date (date not in range of 1-31)

Quality of the argument on why your solution is ready to deploy

- I have tested the output of both the queries in MySQL workbench before using them in the program.
- I have used sub query for creating final query so that it would be easy to understand.
- To validate the information extracted from the database using each query I did following things
 - For validating manager information, I took some records from the output and checked manually whether that record contains correct name of the manager, city of that manager's offices, number of employees who reports to that manager, number of customers served by the staff who report to that manager and the total sales by the staff who report to that customer over a given period.
 - For validating Product Line Information, I took some records from the output and checked manually whether that record contains correct product line name, its description, name of customer who ordered that product line and the total value of this product line order by the customer over a given period.
- I have validated that while reporting, the output of the query doesn't report the any employees, customers, orders, or offices that does not have any interaction over the given period.
- I have written the code in such a manner that it adheres to the right object-oriented design principles.
- The code is divided into various classes where each class defines its own purpose for better understandability.

- I made use of the 'javax.xml' and 'org.w3c.dom' classes for creating xml document rather of writing the XML document on my own as it provides flexibility to add additional elements in the future.
- I've included exceptions where necessary. I have also tested various test cases such as input validation tests, boundary cases tests, control flow tests and data flow tests to ensure that the code doesn't break in conditions like user entering incorrect dates, database connection failure, incorrect format of date, incorrect handling of file, issues while extracting data from the database using query and many more.