**Problem 2**

**Overview**

This program extracts the summary of the company’s operation from the database over a period of time. Further, this data converts the information in a file that follows an XML format.

A summary of the extraction requirements is in the csci3901 course assignment#5 information in the course’s Brightspace.

The solution several implementations to extract the information from the database based on the input information and convert the information into XML format.

The input of the program is taken from the keyboard in the following order

* Starting date of the period
* Ending date of the period
* Name of the output file where the XML format to be generated

The output of the program is formatted in three different categories that is extracted from the database.

* Customer Information
* Product information
* Employee information

The program uses JAXB API of the java EE Platform that helps for reading and writing Java objects to and from XML documents.

**Files and external data**

The whole program is divided into different packages to solve the problem

Customer.java – Class that holds the customer details such as customer name, address object, number of orders in the period, and total order value

CustomerAddress.java – Class that holds the customer address details such as address, city, postal code, country

Employee.java – Class that holds the employee details such as name, office, number of customers active in the period, and total order value

Product.java – Class that holds the product details such as name, vendor, units sold, and total value of products

ProductList.java – class that holds all the list of product sets contains the array list of all products

productSet.java – class that holds the product line and each product details contains the list of all product sets.

Summary.java – Class that holds the information of customer, employee, product between the start date and end date

Year – class that holds the start date and end date of the year

CustomerListDao.java – Class that extracts the customer details from the database and stores in array list of customer list

EmployeeListDao.java – Class that extracts the employee details from the database and stores in array list of employee list

productListDao.java – Class that extracts the product details from the database and stores in array list of product list

DBConnection.java – Class that have two functions which helps to connect with the database and close the connection to the database

XMLConvert.java – Class that have function which takes the input data correctly and prints the XML format

XMLSummary.java – Class that have main method which creates an object of XML convert and asks the user to give the required input.

**Data Structures and their relations to each other**

The program extracts the information from the database using sql driver and the customer, employee, product information is stored in array lists in order to convert into XML format.

getAllCustomers -- the program uses statement that is created to execute the sql query and the retrieved data columns are stored in the form of array lists of customer list using result set.

getAllEmployees -- the program uses statement that is created to execute the sql query and the retrieved data columns are stored in the form of array lists of employee list using result set.

getAllproductSets -- the program uses statement that is created to execute the sql query and the retrieved data columns are stored in the form of array lists of product set list using result set.

**Key algorithms**

The program processes the input taken from the user using keyboard and is extracted into specific and formatted XML tags and is printed on the specific file format.

DbConnection.java – This program helps to connect to the specified database provided the correct URL, username, Password and also close the connection of the database.

CustomerListDao.java – this program contains the getAllCustomers method which helps to extract the customer information based on the query executed from the database. Further the result statement is iterated over to add into respective array lists and convert into XML tags using XML Annotations for creating tags

EmployeeListDao.java – this program contains the getAllEmployees method which helps to extract the customer information based on the query executed from the database. Further the result statement is iterated over to add into respective array lists and convert into XML tags using XML Annotations for creating tags

productListDao.java – this program contains the getAllProductSets method which helps to extract the customer information based on the query executed from the database. Further the result statement is iterated over to add into respective array lists and convert into XML tags using XML Annotations for creating tags

XMLConvert.java – this program creates all the Dao class objects and pass the respective input data. And loads the data from the database to the respective array list and passed to the JAXB context object. After retrieving the necessary information using JAXB context object, marshaller object is created in order to marshal the necessary java class to be converted to the XML file tags.

XMLSummary.java – It is having main function which takes start date, end date, output file name. These dates are checked using the date time formatter. It also checks the output file given is existed or not and is passed to the XMLConvert object to get the necessary information for the XML. If it is formatted to xml then it will return true else false.

**Design Elements**

Design elements for extracting the customers, products, or staff information and can be made necessary changes in these files to extract additional information.

Customers information:

Customer.java – contains the customer address details such as address, city, postal code, country that has to represented as xml tags. XML Annotations are appended based for representing the root element names, element names and accessor type that is used by the jaxb at run time.

CustomerAddress.java – contains the customer address details such as address, city, postal code, country that needs to be represented inside the address tag using XML annotations.

CustomerListDao.java – contains the function that helps to execute the query required to extract the information from the database.

Sql query to get the customer Information:

* First get the details of the orders which are not cancelled and are between the specified dates
* Join with the orderdetails table using order number and group by order number to get the total value of the order
* Again, join with the customers table to get the customer details using customer number
* Then group by customer number to get number of orders and total value of sales.

products information:

Product.java – contains the product details such as name, vendor, units sold, and total value of products that has to represented as xml tags. XML Annotations are appended based for representing the root element names, element names and accessor type that is used by the jaxb at run time.

ProductList.java – class that holds all the list of product sets contains the array list of all products

productSet.java – class that holds the product line and each product details contains the list of all product sets.

ProductListDao.java – contains the function that helps to execute the query required to extract the information from the database.

Sql query to get the product Information:

* First get the details of the orders which are not cancelled and are between the specified dates
* Join with the orderdetails table using order number and group by order number to get the units sold and total value of the order and group by product code
* Again, join with the products table to get the product details using product code
* Then order by product line to get all the required information.

staff information:

Employee.java – contains the employee details such as name, office, number of customers active in the period, and total order value that has to represented as xml tags. XML Annotations are appended based for representing the root element names, element names and accessor type that is used by the jaxb at run time.

EmployeeListDao.java – contains the function that helps to execute the query required to extract the information from the database.

Sql query to get the Employee Information:

* First get the details of the orders which are not cancelled and are between the specified dates
* Join with the orderdetails table using order number and group by order number to get the units sold and total value of the order
* Again, join with the customers table using customer number to get the customer details and group by customer number
* Then join the employees and office table to get all the information.

**@XmlRootElement:** This annotation defines root element of XML file.   
**@XmlElement:** This is used to define element in XML file. It sets name of entity.  
**@XmlElementWrapper(name = ‘name to be given to that wrapper’):** It generates a wrapper element around XML

**Constraints**

* All dates will be in a YYYY-MM-DD format for start date and end date.
* Use mysql JDBC connection for Java.

**Assumptions**

* Orders are taken which are not cancelled as these orders had any interaction over the reporting period.

All the necessary requirements to extract the information from the database are implemented and the also checked with various test scenarios of all the components. Database connections can be changed in DbConnection.java and the main function is written in the XMLSummary which evaluates the different types of input from the user. This solution can be deployed on the server.

**Test cases**

* Provide start and end dates with improper format
* Provide start date that is after the end date
* Provide same start date and end date
* Provide only years as dates
* Provide output file which is not present
* Provide output where data is already present
* Provide different formats of output file name
* Provide output file name that is already present

References:

<https://www.javacodegeeks.com/2013/02/jaxb-tutorial-getting-started.html>