Минобрнауки России | РГРТУ | Кафедра ВПМ

Курс «ПРОГРАММИРОВАНИЕ НА JAVA»

**Отчет о практической работе №**8

Выполнил:

Вербицкая Ирина Сергеевна

студент группы 143

электронная почта oora.frt@gmail.com

Проверил:

Пруцков Александр Викторович

д-р техн. наук, профессор кафедры ВПМ

Рязань 2023

# Задание

Ведение заказов

Сущности

Товары: код товара; цена; единица измерения; описание.

Заказчики: код заказчика; наименование; адрес; телефон; фамилия контактного лица.

Заказы: код заказа; код заказчика; код товара; количество; дата.

Запросы

1. Получить список заказов за определенный период времени. Вместо кодов использовать значения из других таблиц.

2. Получить список заказчиков заданного товара.

3. Найти количество заказов по каждому заказчику.

# Основные классы, реализующие задание

## Класс Runner

package ru.rsreu.verbickaya0805;

import com.prutzkow.resourcer.ProjectResourcer;

import com.prutzkow.resourcer.Resourcer;

import ru.rsreu.verbickaya0805.datalayer.BookingManagerDAO;

import ru.rsreu.verbickaya0805.datalayer.DAOFactory;

import ru.rsreu.verbickaya0805.datalayer.DBType;

import ru.rsreu.verbickaya0805.datalayer.data.BookingList;

import ru.rsreu.verbickaya0805.datalayer.data.CustomerList;

import ru.rsreu.verbickaya0805.datalayer.data.CustomerWithCountOfBookingsList;

public class Runner {

private static Resourcer resourcer = ProjectResourcer.getInstance();

private Runner() {

};

public static void main(String[] args) {

StringBuilder resultString = new StringBuilder("\n");

try {

DAOFactory factory = DAOFactory.getInstance(DBType.ORACLE);

BookingManagerDAO bookingManager = factory.getBookingManagerDAO();

// query 1

try {

BookingList query1 = bookingManager.getBookingsForPeriodOfTime(resourcer.getString("query.date1"),

resourcer.getString("query.date2"));

resultString.append(

resourcer.getString("message.bookingForPeriodOfTime") + "\n" + query1.toStringTable() + "\n");

} catch (MyException e) {

resultString.append(e.toString() + "\n");

}

// query 2

int productId = Integer.parseInt(resourcer.getString("query.productId"));

CustomerList query2 = bookingManager.getCustomersOfSomeProduct(productId);

resultString.append(resourcer.getString("message.customersOfSomeProduct") + " "

+ resourcer.getString("query.productId") + "\n" + query2.toStringTable() + "\n");

// query 3

CustomerWithCountOfBookingsList query3 = bookingManager.getBookingCountForEachCustomer();

resultString.append(

resourcer.getString("message.bookingCountForEachCustomer") + "\n" + query3.toStringTable() + "\n");

} catch (Exception e) {

e.printStackTrace();

} finally {

System.out.println(resultString.toString());

}

}

}

## Класс BookingManagerDAO

package ru.rsreu.verbickaya0805.datalayer;

import ru.rsreu.verbickaya0805.MyException;

import ru.rsreu.verbickaya0805.datalayer.data.BookingList;

import ru.rsreu.verbickaya0805.datalayer.data.CustomerList;

import ru.rsreu.verbickaya0805.datalayer.data.CustomerWithCountOfBookingsList;

public interface BookingManagerDAO {

BookingList getBookingsForPeriodOfTime(String date1, String date2) throws MyException;

CustomerList getCustomersOfSomeProduct(int productId);

CustomerWithCountOfBookingsList getBookingCountForEachCustomer();

}

## Класс OracleBookingManagerDAO

package ru.rsreu.verbickaya0805.datalayer.oracle;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.text.DateFormat;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import com.prutzkow.resourcer.ProjectResourcer;

import com.prutzkow.resourcer.Resourcer;

import ru.rsreu.verbickaya0805.MyException;

import ru.rsreu.verbickaya0805.datalayer.BookingManagerDAO;

import ru.rsreu.verbickaya0805.datalayer.data.BookingList;

import ru.rsreu.verbickaya0805.datalayer.data.CustomerList;

import ru.rsreu.verbickaya0805.datalayer.data.CustomerWithCountOfBookingsList;

public class OracleBookingManagerDAO implements BookingManagerDAO {

private static Resourcer resourcer = ProjectResourcer.getInstance();

private Connection connection;

public OracleBookingManagerDAO(Connection connection) {

this.connection = connection;

}

@Override

public BookingList getBookingsForPeriodOfTime(String date1, String date2) throws MyException {

BookingList bookingList = null;

try {

PreparedStatement st = this.connection

.prepareStatement(resourcer.getString("request.bookingForPeriodOfTime"));

DateFormat format = new SimpleDateFormat("dd.MM.yyyy");

st.setDate(1, new java.sql.Date(format.parse(date1).getTime()));

st.setDate(2, new java.sql.Date(format.parse(date2).getTime()));

ResultSet rs = st.executeQuery();

bookingList = new BookingList(rs);

} catch (SQLException e) {

e.printStackTrace();

} catch (ParseException e) {

throw new MyException(resourcer.getString("myException.wrongDate"));

}

return bookingList;

}

@Override

public CustomerList getCustomersOfSomeProduct(int productId) {

CustomerList customerList = null;

try {

PreparedStatement st = this.connection

.prepareStatement(resourcer.getString("request.customersOfSomeProduct"));

st.setInt(1, productId);

ResultSet rs = st.executeQuery();

customerList = new CustomerList(rs);

} catch (SQLException e) {

e.printStackTrace();

}

return customerList;

}

@Override

public CustomerWithCountOfBookingsList getBookingCountForEachCustomer() {

CustomerWithCountOfBookingsList resultList = null;

try {

PreparedStatement st = this.connection

.prepareStatement(resourcer.getString("request.bookingCountForEachCustomer"));

ResultSet rs = st.executeQuery();

resultList = new CustomerWithCountOfBookingsList(rs);

} catch (SQLException e) {

e.printStackTrace();

}

return resultList;

}

}

## Класс CustomerWithCountOfBookingsList

package ru.rsreu.verbickaya0805.datalayer.data;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

import com.prutzkow.resourcer.ProjectResourcer;

import com.prutzkow.resourcer.Resourcer;

import ru.rsreu.verbickaya0805.utilities.StringOfTableFormer;

public class CustomerWithCountOfBookingsList {

private static Resourcer resourcer = ProjectResourcer.getInstance();

private List<CustomerWithBookingCount> list;

public CustomerWithCountOfBookingsList(ResultSet args) throws SQLException {

this.list = new ArrayList<CustomerWithBookingCount>();

this.add(args);

}

public void add(ResultSet args) throws SQLException {

final String id = "CustomerId";

final String count = "COUNT(\*)";

while (args.next()) {

CustomerWithBookingCount item = new CustomerWithBookingCount(args.getInt(id), args.getInt(count));

this.list.add(item);

}

}

public List<CustomerWithBookingCount> getList() {

return this.list;

}

public String toStringTable() {

final int columnWidth = 15;

String header = StringOfTableFormer.formStringOfTable(columnWidth,

resourcer.getString("customerWithCountOfBookingsList.id"),

resourcer.getString("customerWithCountOfBookingsList.count")) + "\n";

StringBuilder resultString = new StringBuilder(header);

Iterator<CustomerWithBookingCount> iterator = this.list.iterator();

while (iterator.hasNext()) {

CustomerWithBookingCount item = iterator.next();

resultString.append(StringOfTableFormer.formStringOfTable(columnWidth,

Integer.toString(item.getCustomerId()), Integer.toString(item.getBookingCount())) + "\n");

}

return resultString.toString();

}

}

## Класс StringOfTableFormer

package ru.rsreu.verbickaya0805.utilities;

public class StringOfTableFormer {

private StringOfTableFormer() {

}

private static final String SEPARATOR = "|";

public static String formStringOfTable(int columnWidth, String... args) {

StringBuilder resultString = new StringBuilder(StringOfTableFormer.SEPARATOR);

for (String item : args) {

resultString.append(StringOfTableFormer.completeBySpaces(columnWidth, item));

resultString.append(StringOfTableFormer.SEPARATOR);

}

return resultString.toString();

}

public static String formCellOfTableWithoutRightBourder(int columnWidth, String arg) {

StringBuilder resultString = new StringBuilder(StringOfTableFormer.SEPARATOR);

resultString.append(StringOfTableFormer.completeBySpaces(columnWidth, arg));

return resultString.toString();

}

private static String completeBySpaces(int spacesCount, String string) {

StringBuilder resultString = new StringBuilder(string);

for (int i = 0; i < spacesCount - string.length(); i++) {

resultString.append(" ");

}

return resultString.toString();

}

}