**Name: TongWu CWID: A20410395**

Table of Contents

[PART 1: 2](#_Toc441827106)-23

[PART 1: 2](#_Toc441827106)4-25

**PART1:**

**Code:**

Car:

**package** Model;

**public** **class** Car {

**private** String id;

**private** String name;

**private** String type;

**private** **double** buyPrice;

**private** **double** rentPrice;

**public** Car() {

}

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** **double** getBuyPrice() {

**return** buyPrice;

}

**public** **void** setBuyPrice(**double** buyPrice) {

**this**.buyPrice = buyPrice;

}

**public** **double** getRentPrice() {

**return** rentPrice;

}

**public** **void** setRentPrice(**double** rentPrice) {

**this**.rentPrice = rentPrice;

}

}

Customer:

**package** Model;

**public** **class** Customer **extends** User {

**private** String email;

**private** String sex;

**private** **double** balance;

**public** Customer() {

**super**();

}

**public** String getEmail() {

**return** email;

}

**public** **void** setEmail(String email) {

**this**.email = email;

}

**public** String getSex() {

**return** sex;

}

**public** **void** setSex(String sex) {

**this**.sex = sex;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

}

CustomerAccount:

**package** Model;

**public** **class** CustomerAccount {

**private** String accountid;

**private** String userid;

**private** **int** rentCarNumber;

**private** **int** buyCarNumber;

**private** **double** balance;

**public** CustomerAccount() {

}

**public** String getAccountid() {

**return** accountid;

}

**public** **void** setAccountid(String accountid) {

**this**.accountid = accountid;

}

**public** String getUserid() {

**return** userid;

}

**public** **void** setUserid(String userid) {

**this**.userid = userid;

}

**public** **int** getRentCarNumber() {

**return** rentCarNumber;

}

**public** **void** setRentCarNumber(**int** rentCarNumber) {

**this**.rentCarNumber = rentCarNumber;

}

**public** **int** getBuyCarNumber() {

**return** buyCarNumber;

}

**public** **void** setBuyCarNumber(**int** buyCarNumber) {

**this**.buyCarNumber = buyCarNumber;

}

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

}

User:

**package** Model;

**public** **class** User {

**private** String id;

**private** String fullName;

**private** String firstName;

**private** String lastName;

**private** String isAdmin;

**private** String password;

**public** User() {

}

**public** String getId() {

**return** id;

}

**public** **void** setId(String id) {

**this**.id = id;

}

**public** String getFullName() {

**return** fullName;

}

**public** **void** setFullName(String fullName) {

**this**.fullName = fullName;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** String isAdmin() {

**return** isAdmin;

}

**public** **void** setAdmin(String isAdmin) {

**this**.isAdmin = isAdmin;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

}

ParseCar:

package Control;

import java.io.IOException;

import java.util.ArrayList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

import Model.Car;

public class ParseCar {

public static ArrayList<Car> getParseCar() {

ArrayList<Car> list= new ArrayList<Car>();

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder;

try {

builder = factory.newDocumentBuilder();

Document doc;

doc = builder.parse("File//Car.xml");

NodeList carNodes = doc.getElementsByTagName("Car");

for(int i = 0; i<carNodes.getLength();i++){

Element carElement = (Element) carNodes.item(i);

Car car = new Car();

NodeList childNodes =carElement.getChildNodes();

for (int j = 0; j < childNodes.getLength(); j++) {

if(childNodes.item(j).getNodeType()==Node.ELEMENT\_NODE){

if("CarID".equals(childNodes.item(j).getNodeName())){

car.setId((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("CarName".equals(childNodes.item(j).getNodeName())){

car.setName((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("CarType".equals(childNodes.item(j).getNodeName())) {

car.setType((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("BuyPrice".equals(childNodes.item(j).getNodeName())) {

car.setBuyPrice(Double.parseDouble((childNodes.item(j).getFirstChild().getNodeValue())));

}else if("RentPrice".equals(childNodes.item(j).getNodeName())) {

car.setRentPrice(Double.parseDouble((childNodes.item(j).getFirstChild().getNodeValue())));

}

}

}

list.add(car);

}

} catch (SAXException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (ParserConfigurationException e) {

e.printStackTrace();

}

return list;

}

}

ParseCustomer:

package Control;

import java.io.IOException;

import java.util.ArrayList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

import Model.Customer;

public class ParseCustomer {

public static ArrayList<Customer> getParseCustomer() {

ArrayList<Customer> list= new ArrayList<Customer>();

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder;

try {

builder = factory.newDocumentBuilder();

Document doc;

doc = builder.parse("File//Customer.xml");

NodeList cusNodes = doc.getElementsByTagName("Customer");

for(int i = 0; i<cusNodes.getLength();i++){

Element cusElement = (Element) cusNodes.item(i);

Customer customer = new Customer();

NodeList childNodes =cusElement.getChildNodes();

for (int j = 0; j < childNodes.getLength(); j++) {

if(childNodes.item(j).getNodeType()==Node.ELEMENT\_NODE){

if("UserID".equals(childNodes.item(j).getNodeName())){

customer.setId((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("FullName".equals(childNodes.item(j).getNodeName())){

customer.setFullName((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("FirstName".equals(childNodes.item(j).getNodeName())) {

customer.setFirstName(childNodes.item(j).getFirstChild().getNodeValue());

}else if("LastName".equals(childNodes.item(j).getNodeName())) {

customer.setLastName(childNodes.item(j).getFirstChild().getNodeValue());

}else if("Password".equals(childNodes.item(j).getNodeName())) {

customer.setPassword(childNodes.item(j).getFirstChild().getNodeValue());

}else if("IsAdmin".equals(childNodes.item(j).getNodeName())) {

customer.setAdmin((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("Email".equals((childNodes.item(j).getNodeName()))) {

customer.setEmail((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("Sex".equals((childNodes.item(j).getNodeName()))){

customer.setSex((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("Balance".equals((childNodes.item(j).getNodeName()))) {

customer.setBalance(Double.parseDouble((childNodes.item(j).getFirstChild().getNodeValue())));

}

}

}

list.add(customer);

}

} catch (SAXException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (ParserConfigurationException e) {

e.printStackTrace();

}

return list;

}

}

ParseUser

package Control;

import java.io.IOException;

import java.util.ArrayList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

import Model.User;

public class ParseUser {

public static ArrayList<User> getParseUser() {

ArrayList<User> list= new ArrayList<User>();

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder;

try {

builder = factory.newDocumentBuilder();

Document doc;

doc = builder.parse("File//User.xml");

NodeList userNodes = doc.getElementsByTagName("User");

for(int i = 0; i<userNodes.getLength();i++){

Element userElement = (Element) userNodes.item(i);

User user = new User();

NodeList childNodes =userElement.getChildNodes();

for (int j = 0; j < childNodes.getLength(); j++) {

if(childNodes.item(j).getNodeType()==Node.ELEMENT\_NODE){

if("UserID".equals(childNodes.item(j).getNodeName())){

user.setId((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("FullName".equals(childNodes.item(j).getNodeName())){

user.setFullName((childNodes.item(j).getFirstChild().getNodeValue()));

}else if("FirstName".equals(childNodes.item(j).getNodeName())) {

user.setFirstName(childNodes.item(j).getFirstChild().getNodeValue());

}else if("LastName".equals(childNodes.item(j).getNodeName())) {

user.setLastName(childNodes.item(j).getFirstChild().getNodeValue());

}else if("Password".equals(childNodes.item(j).getNodeName())) {

user.setPassword(childNodes.item(j).getFirstChild().getNodeValue());

}else if("IsAdmin".equals(childNodes.item(j).getNodeName())) {

user.setAdmin((childNodes.item(j).getFirstChild().getNodeValue()));

}

}

}

list.add(user);

}

} catch (SAXException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (ParserConfigurationException e) {

e.printStackTrace();

}

return list;

}

}

ParseCustomerAccount

package Control;

import java.io.IOException;

import java.util.ArrayList;

import javax.xml.parsers.DocumentBuilder;

import javax.xml.parsers.DocumentBuilderFactory;

import javax.xml.parsers.ParserConfigurationException;

import org.w3c.dom.Document;

import org.w3c.dom.Element;

import org.w3c.dom.Node;

import org.w3c.dom.NodeList;

import org.xml.sax.SAXException;

import Model.CustomerAccount;

public class ParseCustomerAccount {

public static ArrayList<CustomerAccount> getParseCustomerAccount() {

ArrayList<CustomerAccount> list= new ArrayList<CustomerAccount>();

DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

DocumentBuilder builder;

try {

builder = factory.newDocumentBuilder();

Document doc;

doc = builder.parse("File//CustomerAccount.xml");

NodeList cusaNodes = doc.getElementsByTagName("CustomerAccount");

for(int i = 0; i<cusaNodes.getLength();i++){

Element cusaElement = (Element) cusaNodes.item(i);

CustomerAccount cusa = new CustomerAccount();

NodeList childNodes =cusaElement.getChildNodes();

for (int j = 0; j < childNodes.getLength(); j++) {

if(childNodes.item(j).getNodeType()==Node.ELEMENT\_NODE){

if("CustomerAccountID".equals(childNodes.item(j).getNodeName())){

cusa.setAccountid(((childNodes.item(j).getFirstChild().getNodeValue())));

}else if("UserID".equals(childNodes.item(j).getNodeName())){

cusa.setUserid(((childNodes.item(j).getFirstChild().getNodeValue())));

}else if("RentCarNumber".equals(childNodes.item(j).getNodeName())) {

cusa.setRentCarNumber(Integer.parseInt((childNodes.item(j).getFirstChild().getNodeValue())));

}else if("BuyCarNumber".equals(childNodes.item(j).getNodeName())) {

cusa.setBuyCarNumber(Integer.parseInt((childNodes.item(j).getFirstChild().getNodeValue())));

}else if("Balance".equals(childNodes.item(j).getNodeName())) {

cusa.setBalance(Double.parseDouble((childNodes.item(j).getFirstChild().getNodeValue())));

}

}

}

list.add(cusa);

}

} catch (SAXException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} catch (ParserConfigurationException e) {

e.printStackTrace();

}

return list;

}

}

Connector:

package Control;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import com.mysql.jdbc.Connection;

import Model.Car;

import Model.Customer;

import Model.CustomerAccount;

import Model.User;

public class Connector {

static String sqlStr = "jdbc:mysql://localhost/CarService";

static String rootName = "root";

static String rootPwd = "Tongky100%";

public static void writeToMysql(User user) {

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

System.out.println("Unable to find driver class, load driver failed.");

e.printStackTrace();

}

Statement st = null;

Connection con =null;

try {

con = (Connection)DriverManager.getConnection(sqlStr,rootName,rootPwd);

String UserID= user.getId();

String FullName = user.getFullName();

String FirstName = user.getFirstName();

String LastName = user.getLastName();

String Password = user.getPassword();

String IsAdmin = user.isAdmin();

String sql = "insert into user(UserID,FullName,FirstName,LastName,Password,IsAdmin) values(\""+UserID+"\",\""+FullName+"\",\""+FirstName+"\",\""+LastName+"\",\""+Password+"\",\""+IsAdmin+"\")";

System.out.println(sql);

st = con.createStatement();

st.executeUpdate(sql);

} catch (SQLException e) {

e.printStackTrace();

}finally{

try {

st.close();

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void writeToMysql(Customer customer) {

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

System.out.println("Unable to find driver class, load driver failed.");

e.printStackTrace();

}

Statement st = null;

Connection con =null;

try {

con = (Connection)DriverManager.getConnection(sqlStr,rootName,rootPwd);

String UserID= customer.getId();

String FullName = customer.getFullName();

String FirstName = customer.getFirstName();

String LastName = customer.getLastName();

String Password = customer.getPassword();

String IsAdmin = customer.isAdmin();

String Email = customer.getEmail();

String Sex = customer.getSex();

double Balance = customer.getBalance();

String sql = "insert into Customer(UserID,FullName,FirstName,LastName,Password,IsAdmin,Email,Sex,Balance) values(\""+UserID+"\",\""+FullName+"\",\""+FirstName+"\",\""+LastName+"\",\""+Password+"\",\""+IsAdmin+"\",\""+Email+"\",\""+Sex+"\",\""+Balance+"\")";

System.out.println(sql);

st = con.createStatement();

st.executeUpdate(sql);

} catch (SQLException e) {

e.printStackTrace();

}finally{

try {

st.close();

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void writeToMysql(Car car) {

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

System.out.println("Unable to find driver class, load driver failed.");

e.printStackTrace();

}

Statement st = null;

Connection con =null;

try {

con = (Connection)DriverManager.getConnection(sqlStr,rootName,rootPwd);

String CarID= car.getId();

String CarName = car.getName();

String CarType = car.getType();

double BuyPrice = car.getBuyPrice();

double RentPrice = car.getRentPrice();

String sql = "insert into Car(CarID,CarName,CarType,BuyPrice,RentPrice) values(\""+CarID+"\",\""+CarName+"\",\""+CarType+"\",\""+BuyPrice+"\",\""+RentPrice+"\")";

System.out.println(sql);

st = con.createStatement();

st.executeUpdate(sql);

} catch (SQLException e) {

e.printStackTrace();

}finally{

try {

st.close();

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public static void writeToMysql(CustomerAccount cusa) {

try {

Class.forName("com.mysql.jdbc.Driver");

} catch (ClassNotFoundException e) {

System.out.println("Unable to find driver class, load driver failed.");

e.printStackTrace();

}

Statement st = null;

Connection con =null;

try {

con = (Connection)DriverManager.getConnection(sqlStr,rootName,rootPwd);

String CustomerAccountID= cusa.getAccountid();

String UserID = cusa.getUserid();

int RentCarNumber = cusa.getBuyCarNumber();

int BuyCarNumber = cusa.getRentCarNumber();

double Balance = cusa.getBalance();

String sql = "insert into CustomerAccount(CustomerAccountID,UserID,RentCarNumber,BuyCarNumber,Balance) values(\""+CustomerAccountID+"\",\""+UserID+"\",\""+RentCarNumber+"\",\""+BuyCarNumber+"\",\""+Balance+"\")";

System.out.println(sql);

st = con.createStatement();

st.executeUpdate(sql);

} catch (SQLException e) {

e.printStackTrace();

}finally{

try {

st.close();

con.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

Test:

package Control;

import java.util.ArrayList;

import Model.Car;

import Model.Customer;

import Model.CustomerAccount;

import Model.User;

public class Test {

public static void main(String[] args) {

// TODO Auto-generated method stub

ArrayList<User> user= new ArrayList<User>();

ArrayList<Customer> customer=new ArrayList<Customer>();

ArrayList<Car> car=new ArrayList<Car>();

ArrayList<CustomerAccount> cusa=new ArrayList<CustomerAccount>();

user = ParseUser.getParseUser();

customer= ParseCustomer.getParseCustomer();

car=ParseCar.getParseCar();

cusa=ParseCustomerAccount.getParseCustomerAccount();

for(int i=0;i<user.size();i++)

{

if (user.get(i)!=null) {

Connector.writeToMysql(user.get(i));

}

}

for(int i=0;i<customer.size();i++) {

if(customer.get(i)!=null) {

Connector.writeToMysql(customer.get(i));

}

}

for(int i=0;i<car.size();i++) {

if(car.get(i)!=null) {

Connector.writeToMysql(car.get(i));

}

}

for(int i=0;i<car.size();i++) {

if(cusa.get(i)!=null) {

Connector.writeToMysql(cusa.get(i));

}

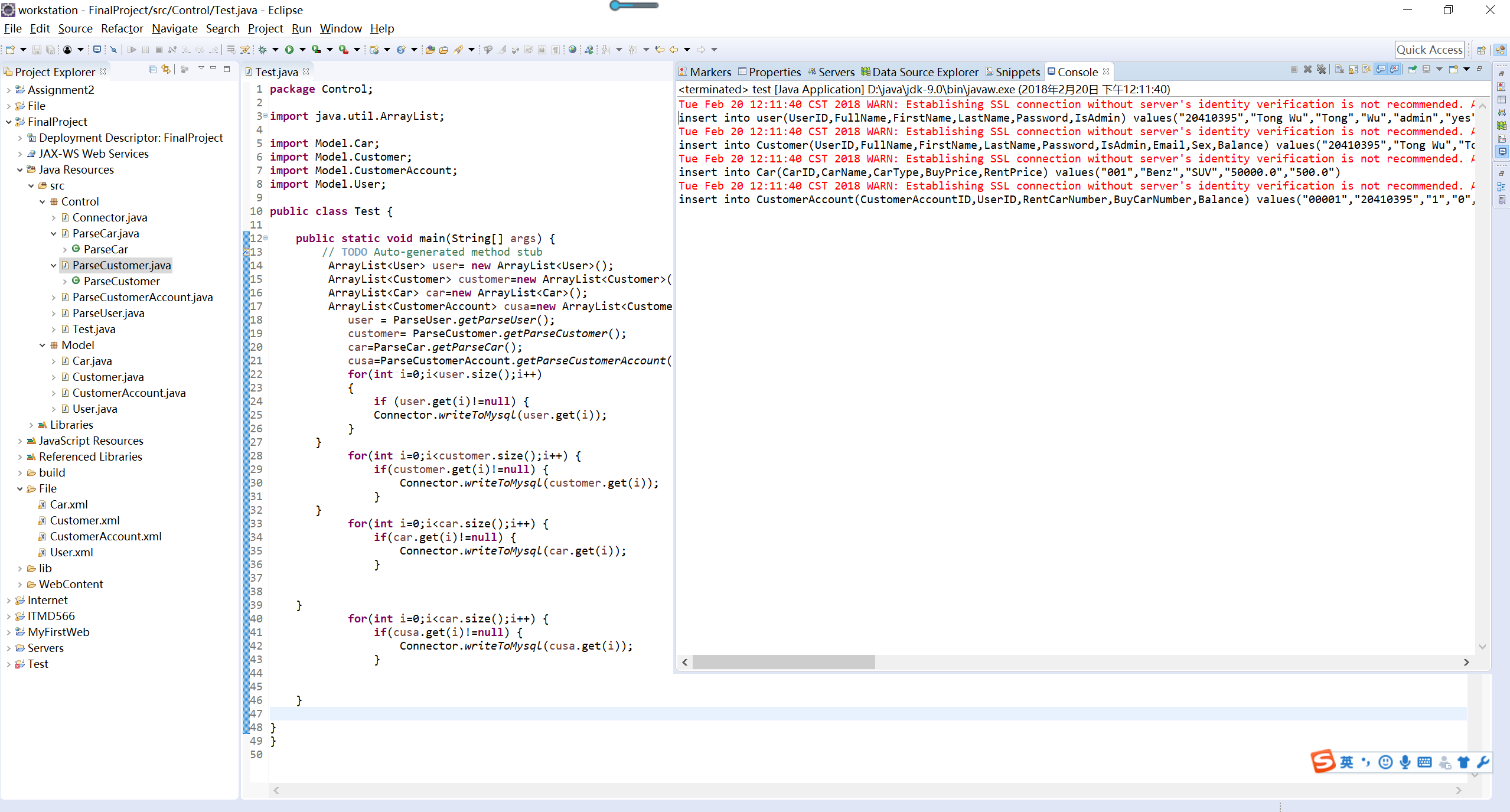
}

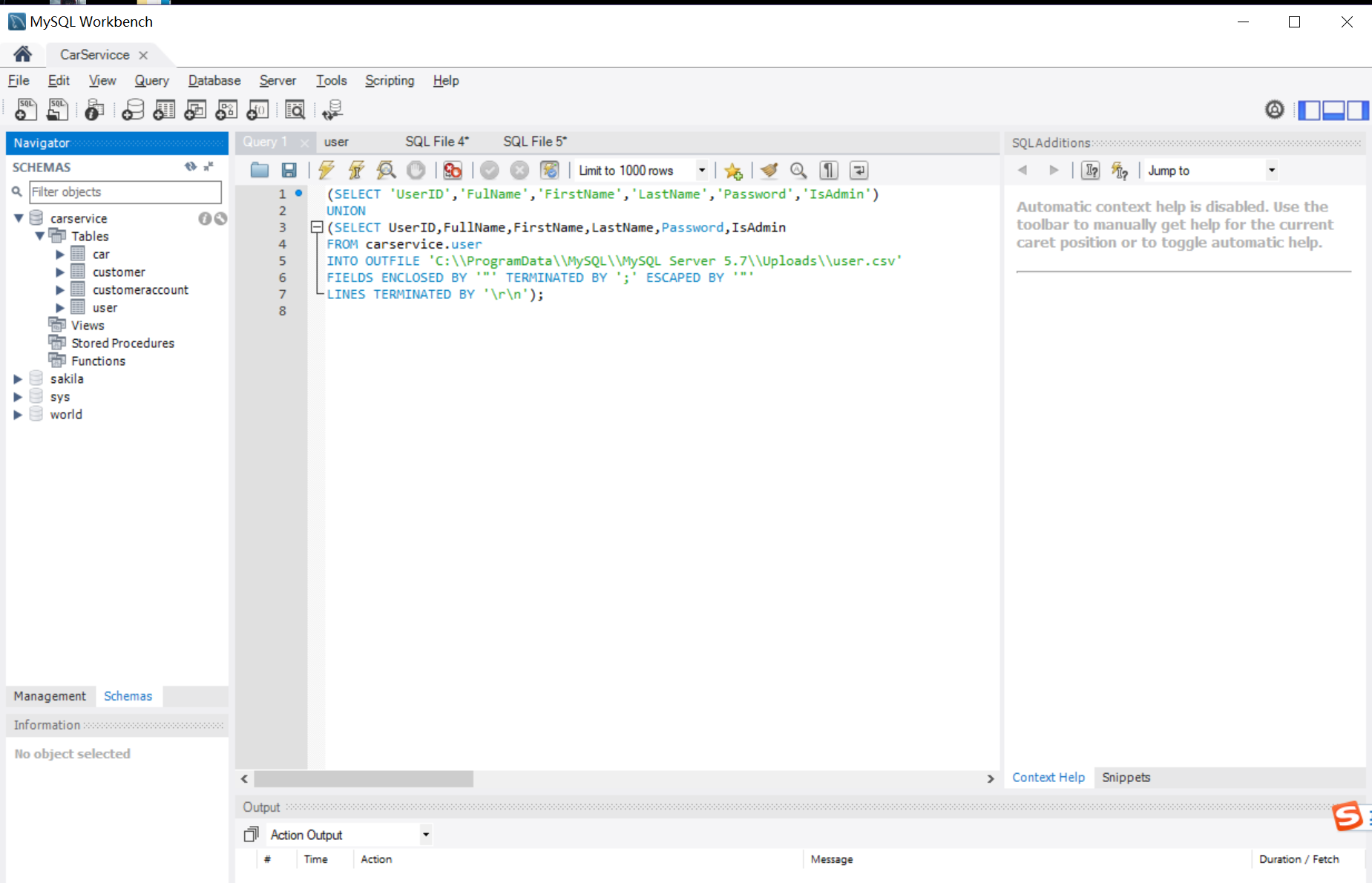
}

}

**PART2:**

**Screenshot:**

****

****