**Endterm project, Makhmutov Eldar**

**(Please turn on viewer mode to web-site for better experience)**

**I used these links during this project:**

[**https://www.baeldung.com/java-using-not-in-if-conditions#:~:text=The%20not%20operator%20is%20a,represented%20in%20Java%20by%20the%20!&text=It's%20a%20unary%20operator%20that,the%20value%20of%20its%20operand**](https://www.baeldung.com/java-using-not-in-if-conditions#:~:text=The%20not%20operator%20is%20a,represented%20in%20Java%20by%20the%20!&text=It's%20a%20unary%20operator%20that,the%20value%20of%20its%20operand)**.**

[**https://moodle.astanait.edu.kz/mod/resource/view.php?id=15207**](https://moodle.astanait.edu.kz/mod/resource/view.php?id=15207)

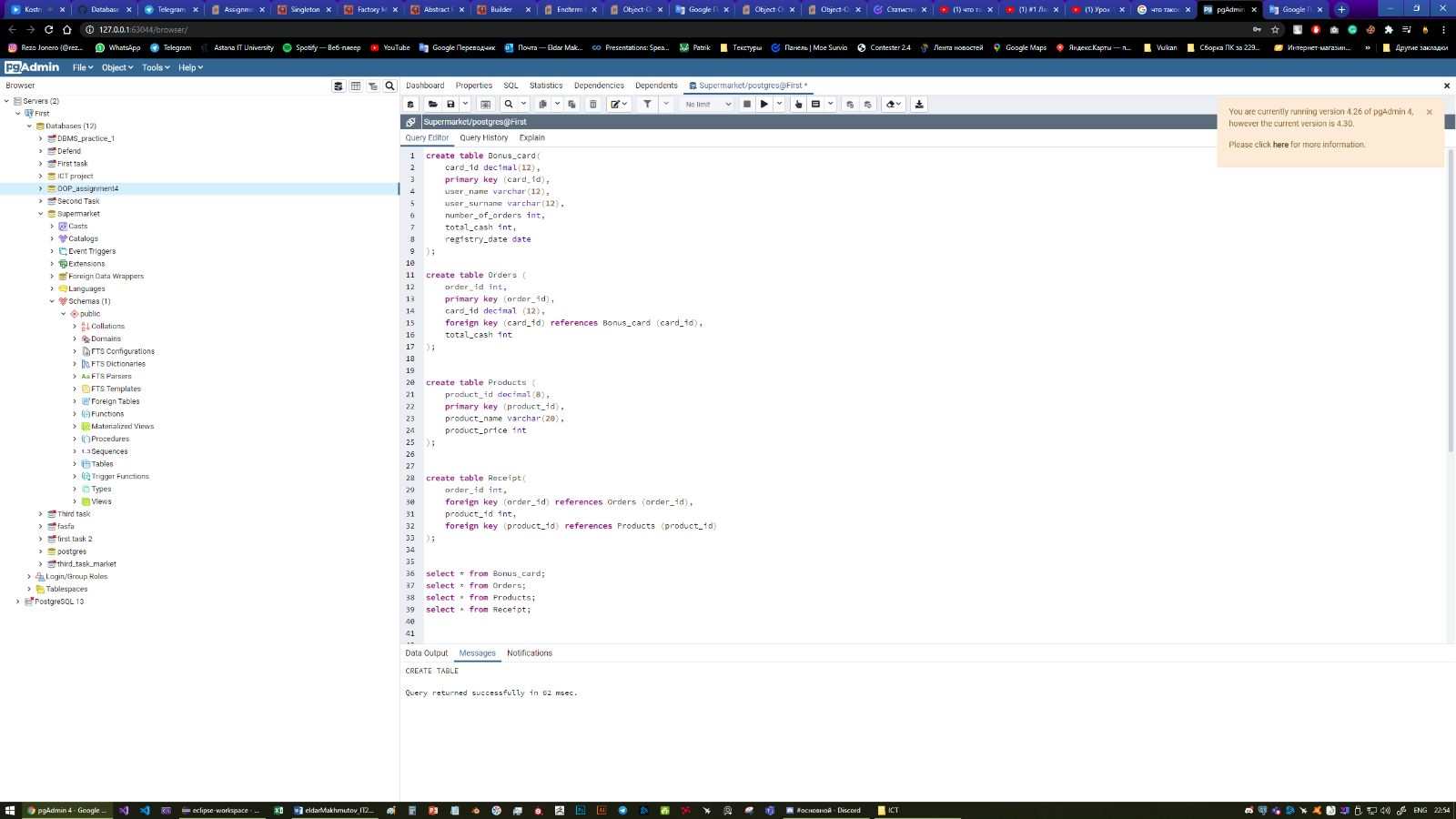
[**https://coderoad.ru/19778887/%D0%A7%D1%82%D0%BE-%D0%B2%D1%8B%D0%B7%D1%8B%D0%B2%D0%B0%D0%B5%D1%82-java-lang-ClassNotFoundException-%D0%B2-%D0%B7%D0%B0%D0%BF%D1%83%D1%89%D0%B5%D0%BD%D0%BD%D0%BE%D0%BC-%D0%BA%D0%BE%D0%B4%D0%B5**](https://coderoad.ru/19778887/%D0%A7%D1%82%D0%BE-%D0%B2%D1%8B%D0%B7%D1%8B%D0%B2%D0%B0%D0%B5%D1%82-java-lang-ClassNotFoundException-%D0%B2-%D0%B7%D0%B0%D0%BF%D1%83%D1%89%D0%B5%D0%BD%D0%BD%D0%BE%D0%BC-%D0%BA%D0%BE%D0%B4%D0%B5)

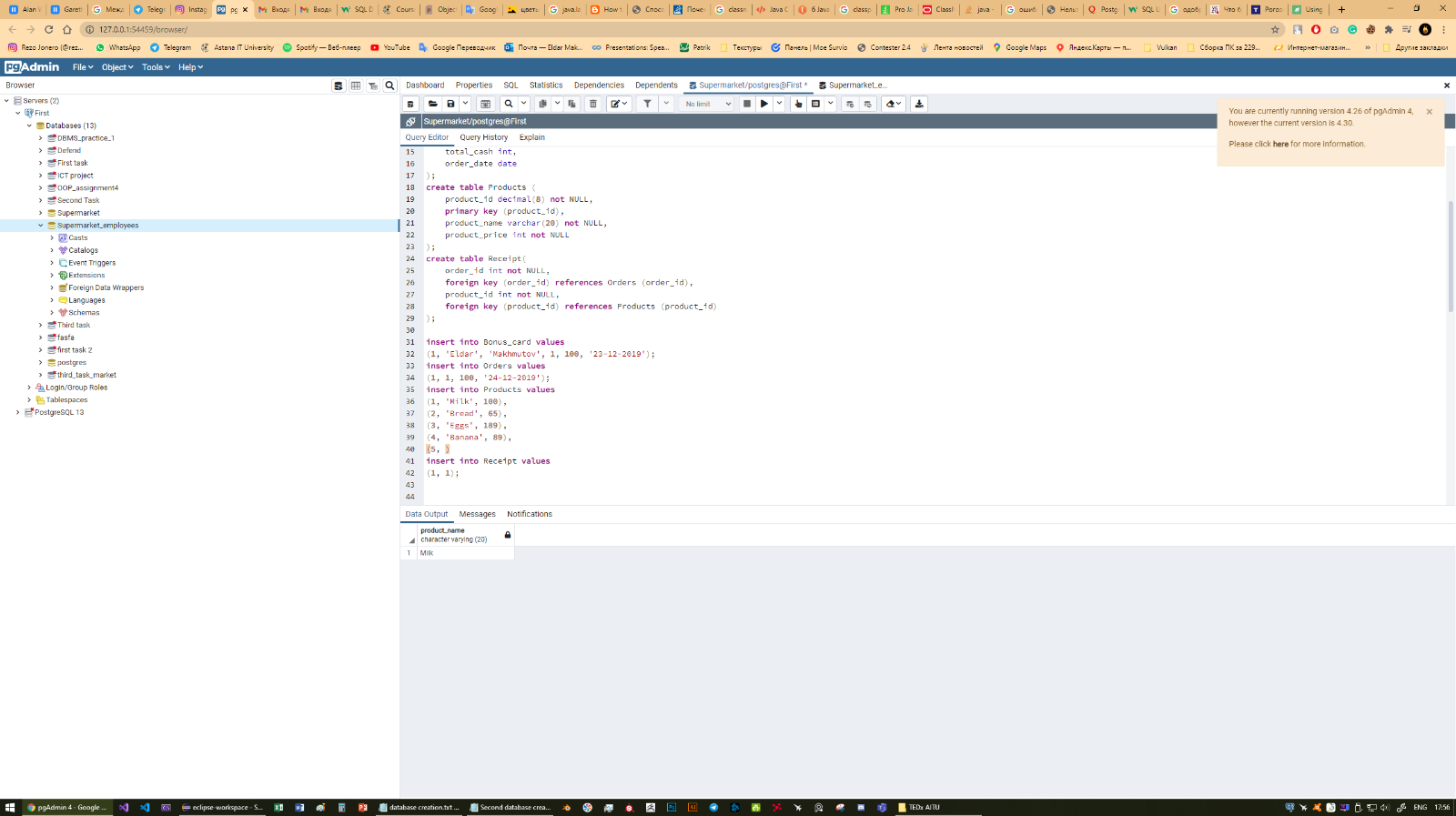
[**http://pr0java.blogspot.com/2015/04/classpath-java.html**](http://pr0java.blogspot.com/2015/04/classpath-java.html)

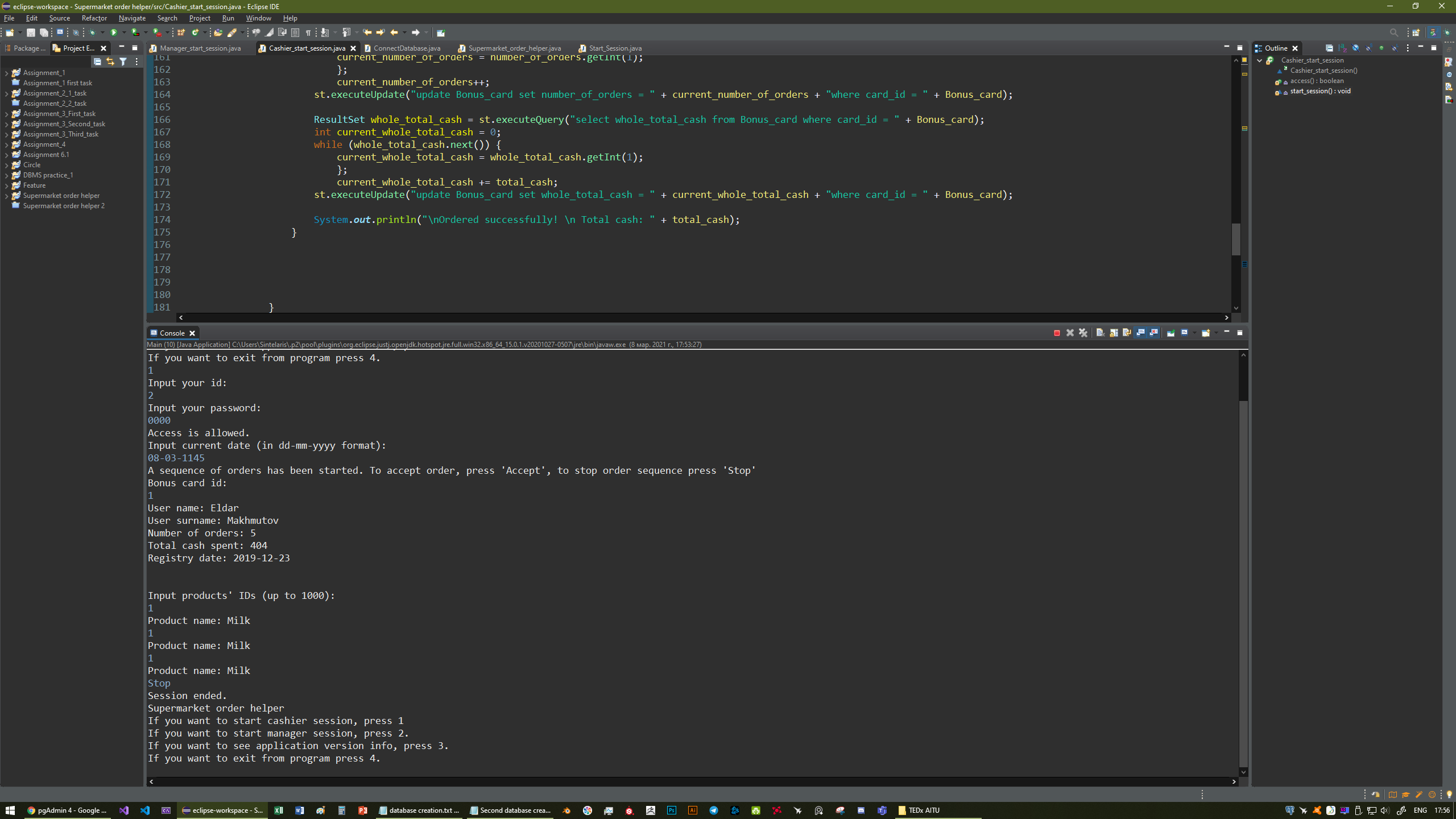
[**https://javarevisited.blogspot.com/2011/08/classnotfoundexception-in-java-example.html#axzz6oVRY6S2a**](https://javarevisited.blogspot.com/2011/08/classnotfoundexception-in-java-example.html#axzz6oVRY6S2a)

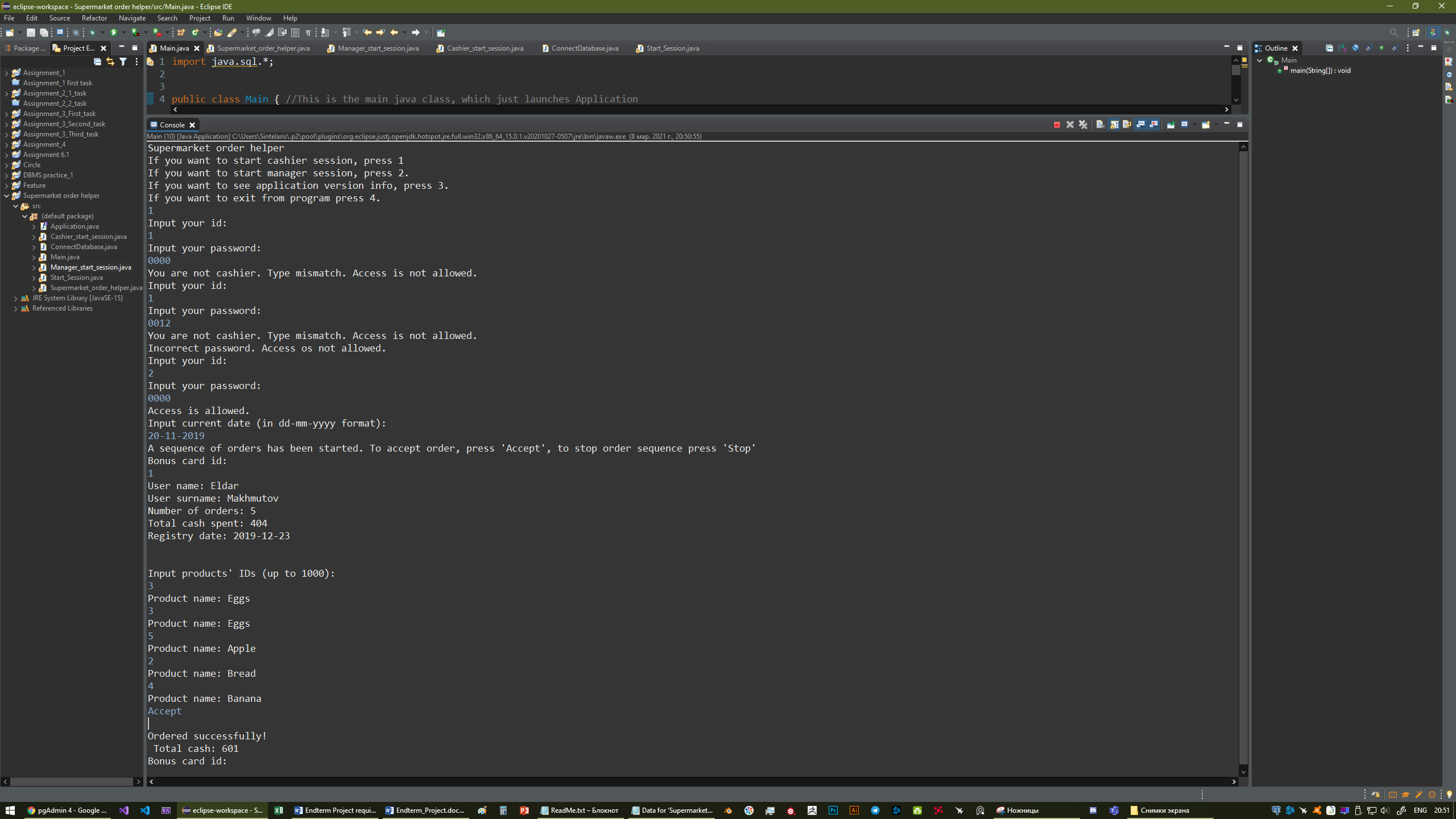
[**https://javarush.ru/groups/posts/817-6-java-iskljucheniy-kotorihe-presledujut-novichkov**](https://javarush.ru/groups/posts/817-6-java-iskljucheniy-kotorihe-presledujut-novichkov)

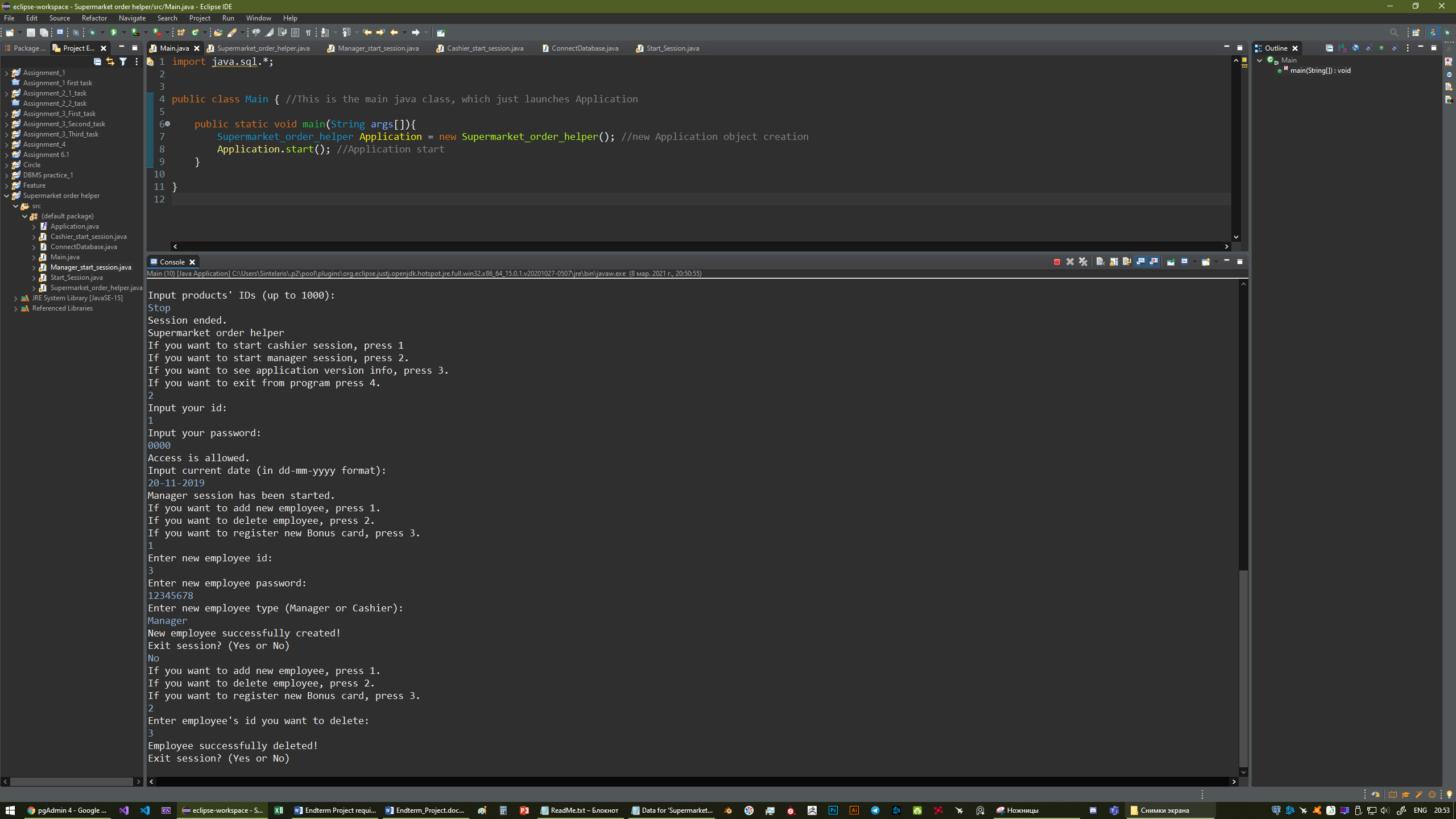
[**https://docs.oracle.com/javase/7/docs/api/java/lang/ClassNotFoundException.html**](https://docs.oracle.com/javase/7/docs/api/java/lang/ClassNotFoundException.html)

****

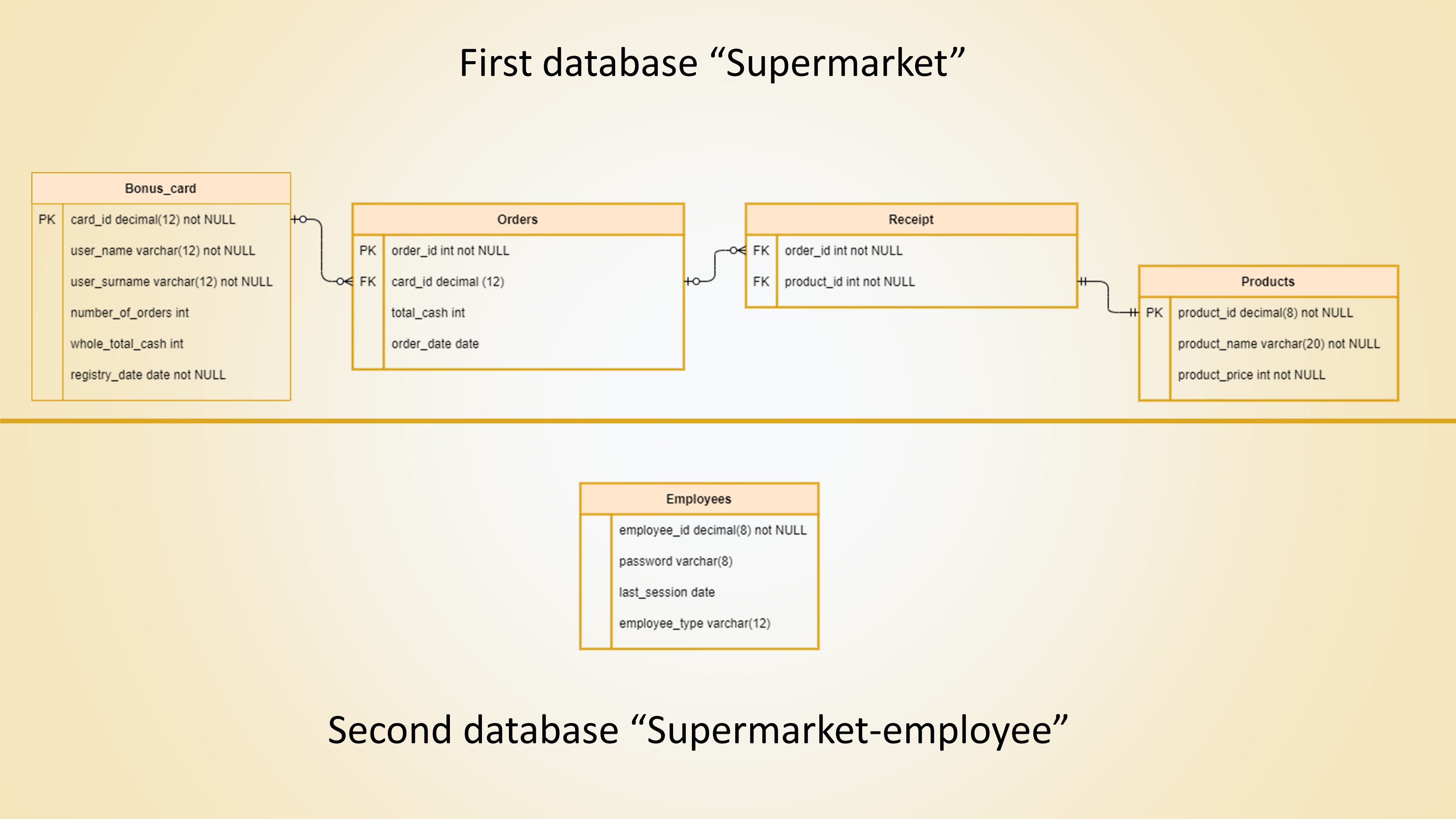








**ER diagram**



**Code:**

**Main**

**import java.sql.\*;**

**public class Main { //This is the main java class, which just launches Application**

**public static void main(String args[]){**

**Supermarket\_order\_helper Application = new Supermarket\_order\_helper(); //new Application object creation**

**Application.start(); //Application start**

**}**

**}**

**Application**

**public interface Application { //default interface for all Applications.**

**public void start();**

**public void end();**

**}**

**Supermarket\_order\_helper**

**import java.sql.\*;**

**import java.util.Scanner;**

**public class Supermarket\_order\_helper implements Application { //this is my Application, which was made with the help of main Application interface.**

**private static int State = 0; //main variable to make decisions**

**private static String Window\_name = "Supermarket order helper";**

**private static String Application\_version = "1.0"; //saves Application version**

**private static void begin(){ //this method runs first console of Application and then takes value to switch to next action**

**System.out.println(Window\_name);**

**System.out.println("If you want to start cashier session, press 1");**

**System.out.println("If you want to start manager session, press 2.");**

**System.out.println("If you want to see application version info, press 3.");**

**System.out.println("If you want to exit from program press 4.");**

**Scanner input\_value = new Scanner(System.in);**

**State = input\_value.nextInt();**

**}**

**static void Option\_switcher() { //this method switches possible actions**

**switch (State) {**

**case (0):**

**System.out.println("Logic error: please ask user for the option.");**

**break;**

**case (1):**

**Start\_Session cashier\_session = new Cashier\_start\_session();**

**boolean access\_check = false; //this variable defines if the user (Cashier) got access or not**

**do {**

**cashier\_session.input\_User\_id(); //importing User\_id and password to check access...**

**cashier\_session.input\_User\_password();**

**access\_check = cashier\_session.access();**

**} while (access\_check == false); //it will run until access will not be get**

**cashier\_session.input\_current\_date(); //it will be needed later.**

**cashier\_session.start\_session(); //starting session...**

**break;**

**case (2):**

**Start\_Session manager\_session = new Manager\_start\_session();**

**boolean access\_check\_manager = false; //this variable defines if the user (Manager) got access or not**

**do {**

**manager\_session.input\_User\_id(); //importing User\_id and password to check access...**

**manager\_session.input\_User\_password();**

**access\_check\_manager = manager\_session.access();**

**} while (access\_check\_manager == false); //it will run until access will not be get**

**manager\_session.input\_current\_date();//it will be needed later.**

**manager\_session.start\_session(); //starting session...**

**break;**

**case (3):**

**System.out.println(Window\_name);**

**System.out.println("Application version: " + Application\_version); //showing application version...**

**begin();**

**break;**

**case (4):**

**System.out.println("Exitting...");//exits program**

**break;**

**default:**

**System.out.println("Logic error: impossible situation.");**

**break;**

**}**

**}**

**@Override**

**public void start() {**

**do {//the program will run until the user exit**

**begin();**

**Option\_switcher();**

**}**

**while (State != 4);**

**}**

**@Override**

**public void end() {**

**}**

**}**

**Manager\_start\_session**

**import java.sql.\*;**

**import java.util.Scanner;**

**public class Manager\_start\_session extends Start\_Session {**

**@Override**

**public boolean access() { //this method connects to Database "Supermarket\_employees" and checks access**

**Connection connection = null;**

**Statement st = null;**

**ConnectDatabase Database = new ConnectDatabase();**

**connection = Database.get\_connection("localhost", "5432", "Supermarket\_employees", "postgres", "12455689");**

**ResultSet Result = null;**

**boolean check = false;**

**try {**

**st = connection.createStatement();**

**ResultSet Password\_and\_Employee\_type = st.executeQuery("select password, employee\_type from Employees where employee\_id = " + employee\_id);**

**String password\_from\_DB = "";**

**String type\_from\_DB = "";**

**while (Password\_and\_Employee\_type.next()) {**

**password\_from\_DB = Password\_and\_Employee\_type.getString(1);**

**type\_from\_DB = Password\_and\_Employee\_type.getString(2);**

**}**

**if (type\_from\_DB.equals("Manager") && password\_from\_DB.equals(employee\_password)) {**

**check = true;**

**System.out.println("Access is allowed.");**

**}**

**else {**

**if (!type\_from\_DB.equals("Manager")) {**

**System.out.println("You are not cashier. Type mismatch. Access is not allowed.");**

**}**

**if (!password\_from\_DB.equals(employee\_password)) {**

**System.out.println("Incorrect password. Access os not allowed.");**

**}**

**check = false;**

**}**

**}**

**catch(Exception e) {**

**System.out.println("This employee does not exist.");**

**check = false;**

**}**

**finally {**

**try {**

**st.close();**

**connection.close();**

**}**

**catch(Exception u) {**

**System.out.println(u);**

**}**

**}**

**return check;**

**}**

**@Override**

**void start\_session() { //In manager session it is possible to add new employee, delete employee and register new Bonus card**

**/\*Connection initialization. Manager can operate with two databases in the session\*/**

**Connection connection = null;**

**Connection connection2 = null;**

**Statement st = null;**

**Statement st2 = null;**

**ConnectDatabase Database\_1 = new ConnectDatabase();**

**connection = Database\_1.get\_connection("localhost", "5432", "Supermarket", "postgres", "12455689");**

**ConnectDatabase Database\_2 = new ConnectDatabase();**

**connection2 = Database\_2.get\_connection("localhost", "5432", "Supermarket\_employees", "postgres", "12455689");**

**try {**

**st = connection.createStatement();**

**st2 = connection2.createStatement();**

**} catch (Exception e) {**

**System.out.println(e);**

**}**

**boolean abort\_session = false;**

**System.out.println("Manager session has been started.");**

**do {//asking for actions...**

**System.out.println("If you want to add new employee, press 1.");**

**System.out.println("If you want to delete employee, press 2.");**

**System.out.println("If you want to register new Bonus card, press 3.");**

**Scanner input\_value = new Scanner(System.in);**

**int Switch = input\_value.nextInt(); //as in Supermarket\_order\_helper class this variable switches actions**

**if (Switch == 1) { //this part of "if" creates new employee in "Supermarket\_employees" database**

**System.out.println("Enter new employee id:");**

**input\_value = new Scanner(System.in);**

**String employee\_id = input\_value.nextLine();**

**System.out.println("Enter new employee password:");**

**input\_value = new Scanner(System.in);**

**String password = input\_value.nextLine();**

**System.out.println("Enter new employee type (Manager or Cashier):");**

**input\_value = new Scanner(System.in);**

**String Employee\_Type = input\_value.nextLine();**

**try {**

**st2.executeUpdate("insert into Employees values(" + employee\_id + ", '" + password + "', '" + Date + "', '" + Employee\_Type + "')");**

**System.out.println("New employee successfully created!");**

**} catch (Exception e) {**

**System.out.println(e);**

**}**

**}**

**else if (Switch == 2) { //this part of "if" deletes employee in "Supermarket\_employees" database**

**System.out.println("Enter employee's id you want to delete:");**

**input\_value = new Scanner(System.in);**

**String employee\_id = input\_value.nextLine();**

**try {**

**st2.executeUpdate("delete from Employees where employee\_id = " + employee\_id + ";");**

**System.out.println("Employee successfully deleted!");**

**} catch (Exception e) {**

**System.out.println(e);**

**}**

**} else if (Switch == 3) {//this "if" creates new Bonus card. Order\_id applies automatically. The same with whole\_total\_cash, number\_of\_orders and Date**

**System.out.println("Enter new Bonus\_card owner name:");**

**input\_value = new Scanner(System.in);**

**String Bonus\_card\_user\_name = input\_value.nextLine();**

**System.out.println("Enter new Bonus\_card owner surname:");**

**input\_value = new Scanner(System.in);**

**String Bonus\_card\_user\_surname = input\_value.nextLine();**

**try {**

**ResultSet Last\_id = st.executeQuery("select card\_id from Bonus\_card where card\_id = (select max(card\_id) from Bonus\_card);");**

**int current\_id = 0;**

**while (Last\_id.next()) {**

**current\_id = Last\_id.getInt(1);**

**};**

**current\_id++;**

**st.executeUpdate("insert into Bonus\_card values(" + current\_id + ", '" + Bonus\_card\_user\_name + "', '" + Bonus\_card\_user\_surname + "', 0, 0, '" + Date + "')");**

**} catch (Exception e) {**

**System.out.println(e);**

**}**

**}**

**System.out.println("Exit session? (Yes or No)"); //it asks for exit after every action.**

**String check = input\_value.nextLine();**

**if(check.equals("Yes")) {**

**abort\_session = true;**

**}**

**}**

**while(abort\_session == false);**

**try { //closing connection for both databases...**

**st.close();**

**connection.close();**

**st2.close();**

**connection2.close();**

**}**

**catch(Exception y) {**

**System.out.println(y);**

**}**

**};**

**};**

**Cashier\_start\_session**

**import java.sql.\*;**

**import java.util.Scanner;**

**public class Cashier\_start\_session extends Start\_Session{**

**@Override**

**public boolean access() { //this method connects to Database "Supermarket\_employees" and checks access**

**Connection connection = null;**

**Statement st = null;**

**ConnectDatabase Database = new ConnectDatabase();**

**connection = Database.get\_connection("localhost", "5432", "Supermarket\_employees", "postgres", "12455689");**

**ResultSet Result = null;**

**boolean check = false;**

**try {**

**st = connection.createStatement();**

**ResultSet Password\_and\_Employee\_type = st.executeQuery("select password, employee\_type from Employees where employee\_id = " + employee\_id);**

**String password\_from\_DB = "";**

**String type\_from\_DB = "";**

**while (Password\_and\_Employee\_type.next()) {**

**password\_from\_DB = Password\_and\_Employee\_type.getString(1);**

**type\_from\_DB = Password\_and\_Employee\_type.getString(2);**

**}**

**if (type\_from\_DB.equals("Cashier") && password\_from\_DB.equals(employee\_password)) {**

**check = true;**

**System.out.println("Access is allowed.");**

**}**

**else {**

**if (!type\_from\_DB.equals("Cashier")) {**

**System.out.println("You are not cashier. Type mismatch. Access is not allowed.");**

**}**

**if (!password\_from\_DB.equals(employee\_password)) {**

**System.out.println("Incorrect password. Access os not allowed.");**

**}**

**check = false;**

**}**

**}**

**catch(Exception e) {**

**System.out.println("This employee does not exist.");**

**check = false;**

**}**

**finally {**

**try {**

**st.close();**

**connection.close();**

**}**

**catch(Exception u) {**

**System.out.println(u);**

**}**

**}**

**return check;**

**}**

**@Override**

**void start\_session() { //this method starts Cashier session**

**/\*Connection initialization\*/**

**Connection connection = null;**

**Statement st = null;**

**ConnectDatabase Database\_1 = new ConnectDatabase();**

**connection = Database\_1.get\_connection("localhost", "5432", "Supermarket", "postgres", "12455689");**

**ResultSet Result = null;**

**try {**

**st = connection.createStatement();**

**} catch (Exception e) {**

**System.out.println(e);**

**}**

**boolean Abort\_sequence = false; //Aborts all order sequence when true**

**System.out.println("A sequence of orders has been started. To accept order, press 'Accept', to stop order sequence press 'Stop'"); //the main role of cashier session is to make order sequence at the checkout**

**boolean Stop\_sequence = false; //Aborts current user product sequence when true**

**String check = " "; //exactly this variable checks all input before, and stops it when it equals "Stop" or "Accept"**

**do {**

**System.out.println("Bonus card id:");**

**Scanner input\_value = new Scanner(System.in); //getting Bonus card id**

**if (input\_value.equals("Stop")) {**

**Abort\_sequence = true;**

**}**

**String Bonus\_card;**

**String Product\_id[] = new String[1000]; //all product IDs stored in this array**

**boolean Bonus\_card\_fail = false; //sometimes cards can be unregistered and value of this variable will be false until it will get correct card id**

**do {**

**Bonus\_card = input\_value.nextLine();**

**try {**

**Result = st.executeQuery("select user\_name, user\_surname, number\_of\_orders, whole\_total\_cash, registry\_date from Bonus\_card where card\_id = " + Bonus\_card);**

**while (Result.next()) {**

**System.out.println("User name: " + Result.getString(1)); //getting Bonus card's User's data...**

**System.out.println("User surname: " + Result.getString(2));**

**System.out.println("Number of orders: " + Result.getInt(3));**

**System.out.println("Total cash spent: " + Result.getInt(4));**

**System.out.println("Registry date: " + Result.getDate(5) + "\n");**

**}**

**Bonus\_card\_fail = false;**

**}**

**catch(Exception e) {**

**System.out.println("This bonus card unknown or unregistered. Please go to manager to register it.");**

**Bonus\_card\_fail = true;**

**}**

**} while (Bonus\_card\_fail == true);**

**try {**

**System.out.println("\nInput products' IDs (up to 1000):"); //getting all IDs**

**for (int i = 0; i < 1000 && Stop\_sequence == false && Abort\_sequence == false; i++) {**

**check = input\_value.nextLine(); //checking for "Accept" and "Stop"...**

**if (check.equals("Accept")) {**

**Stop\_sequence = true;**

**if (i != 0) {**

**int total\_cash = 0;**

**int product\_current\_price = 0;**

**for (int c = 0; c < i; c++) { //finding total cash of all ordered products in current order**

**ResultSet Product\_price = st.executeQuery("select product\_price from Products where product\_id = " + Product\_id[c]);**

**while (Product\_price.next()) {**

**product\_current\_price = Product\_price.getInt(1);**

**};**

**total\_cash += product\_current\_price;**

**}**

**ResultSet Last\_id = st.executeQuery("select order\_id from Orders where order\_id = (select max(order\_id) from Orders);");//checking for highest order ID to add next ID**

**int current\_id = 0;**

**while (Last\_id.next()) {**

**current\_id = Last\_id.getInt(1);**

**};**

**current\_id++;**

**st.executeUpdate("insert into Orders values (" + current\_id + ", " + Bonus\_card + ", " + total\_cash + ", '" + Date + "')");//creating new order with current day and unique ID...**

**for (int c = 0; c < i; c++) {**

**st.executeUpdate("insert into Receipt values (" + current\_id + ", " + Product\_id[c] + ")"); //creating sequence of receipts with current day and unique ID...**

**}**

**ResultSet number\_of\_orders = st.executeQuery("select number\_of\_orders from Bonus\_card where card\_id = " + Bonus\_card); //Updating data in Bonus card (number of orders increased)**

**int current\_number\_of\_orders = 0;**

**while (number\_of\_orders.next()) {**

**current\_number\_of\_orders = number\_of\_orders.getInt(1);**

**};**

**current\_number\_of\_orders++;**

**st.executeUpdate("update Bonus\_card set number\_of\_orders = " + current\_number\_of\_orders + "where card\_id = " + Bonus\_card); //applying results...**

**ResultSet whole\_total\_cash = st.executeQuery("select whole\_total\_cash from Bonus\_card where card\_id = " + Bonus\_card); //Updating data in Bonus card (whole total spent cash increased)**

**int current\_whole\_total\_cash = 0;**

**while (whole\_total\_cash.next()) {**

**current\_whole\_total\_cash = whole\_total\_cash.getInt(1);**

**};**

**current\_whole\_total\_cash += total\_cash;**

**st.executeUpdate("update Bonus\_card set whole\_total\_cash = " + current\_whole\_total\_cash + "where card\_id = " + Bonus\_card); //applying results...**

**System.out.println("\nOrdered successfully! \n Total cash: " + total\_cash); //Showing total cash of order...**

**}**

**}**

**else if (check.equals("Stop")) { //abort cashier order sequence if is is "Stop".**

**Abort\_sequence = true;**

**} else {**

**Product\_id[i] = check; //writing down if check is neither "Stop" nor "Accept"**

**ResultSet current\_product = st.executeQuery("select product\_name from Products where product\_id = " + check); //showing current product...**

**while (current\_product.next()) {**

**System.out.println("Product name: " + current\_product.getString(1));**

**}**

**}**

**}**

**}**

**catch (Exception d) {**

**System.out.println(d);**

**}**

**Stop\_sequence = false;**

**}**

**while (Abort\_sequence == false); //do while session is not aborted**

**System.out.println("Session ended.");**

**try { //closing connection...**

**st.close();**

**connection.close();**

**}**

**catch(Exception y) {**

**System.out.println(y);**

**}**

**};**

**}**

**Start\_session**

**import java.util.Scanner;**

**import java.sql.\*;**

**public class Start\_Session {**

**Start\_Session() {**

**}**

**public boolean access() { //kinda "default" access**

**return false;**

**};**

**String host = "localhost";**

**String port = "5432";**

**String username = "postgres";**

**String employee\_id = " ";**

**String employee\_password = " ";**

**String Date = " ";**

**public void input\_User\_id() { //import user name method**

**System.out.println("Input your id:");**

**Scanner input\_value = new Scanner(System.in);**

**employee\_id = input\_value.nextLine();**

**};**

**public void input\_User\_password() { //import user password method**

**System.out.println("Input your password:");**

**Scanner input\_value = new Scanner(System.in);**

**employee\_password = input\_value.nextLine();**

**};**

**public void input\_current\_date() { //import current date method**

**System.out.println("Input current date (in dd-mm-yyyy format):");**

**Scanner input\_value = new Scanner(System.in);**

**Date = input\_value.nextLine();**

**};**

**void start\_session() { //kinda "default" start session access**

**Connection connection = null;**

**Statement st = null;**

**ConnectDatabase Database = new ConnectDatabase();**

**connection = Database.get\_connection("localhost", "5432", "Supermarket", "postgres", "12455689");**

**ResultSet Result = null;**

**try {**

**st = connection.createStatement();**

**st.close();**

**connection.close();**

**}**

**catch(Exception e) {**

**System.out.println(e);**

**}**

**}**

**}**

**ConnectDatabase**

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**public class ConnectDatabase { //this class connects optional databases**

**public Connection get\_connection(String host, String port, String db\_name, String username, String password) {**

**Connection connection = null;**

**try {**

**Class.forName("org.postgresql.Driver");**

**// Establish the connection**

**//connection = DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres", "postgres", "0000");**

**connection = DriverManager.getConnection("jdbc:postgresql://"+host+":"+port+"/"+db\_name, username, password);**

**if(connection!=null) {**

**} else {**

**System.out.println("Database failed to open \n");**

**}**

**}catch(Exception e) {**

**System.out.println("problem");**

**System.out.println(e);**

**System.out.println("problem");**

**}**

**return connection;**

**}**

**}**