Our code has 5 classes and 2 interfaces, there you can check it.

-----------------------------------------------------------------------------

<https://github.com/yerqssyl/Rolls_Royce_Store.git>

-----------------------------------------------------------------------------

package com.company;  
  
public class Main { // giving variables  
 public static void main (String[] args) {  
 IntDB db = new SqlDB();  
 IntOrderArchive repo = new OrderArchive(db);  
 OrderControl controller = new OrderControl(repo);  
 DBW app = new DBW(controller);  
 app.start();  
 }  
}

package com.company;  
  
public interface IntOrderArchive { // interface of functions(methods and getters)  
 String getProductList();  
 String getColor();  
 boolean createUser(String customer\_name, String customer\_number);  
 void purchaseProduct(String customer\_name, String customer\_number, int id, String color);  
 void getOrderDetails(String customer\_name, String customer\_number);  
}

package com.company;  
  
  
public class OrderControl{  
 private final IntOrderArchive repo;  
  
 public OrderControl(IntOrderArchive repo) {  
 this.repo = repo;  
 }  
  
 public String getProductList(){  
 String response = repo.getProductList();  
 return (response == null ? "" : response.toString());  
 }// getter for Product List  
 public String getColor(){  
 String response = repo.getColor(); // to output color list  
 return (response == null ? "" : response.toString());  
 }// getter color  
 public String createUser(String customer\_name, String customer\_phone){  
 boolean created = repo.createUser(customer\_name, customer\_phone);  
 return (created ? "" : ""); // method to add user in sql  
 }// to create User  
 public void purchaseProduct(String customer\_name, String customer\_phone, int id, String color){  
 repo.purchaseProduct(customer\_name, customer\_phone, id, color);  
 }//method to purchase RR  
 public void getOrderDetails(String customer\_name, String customer\_phone){  
 repo.getOrderDetails(customer\_name, customer\_phone);  
 }//getter for Order Details  
}

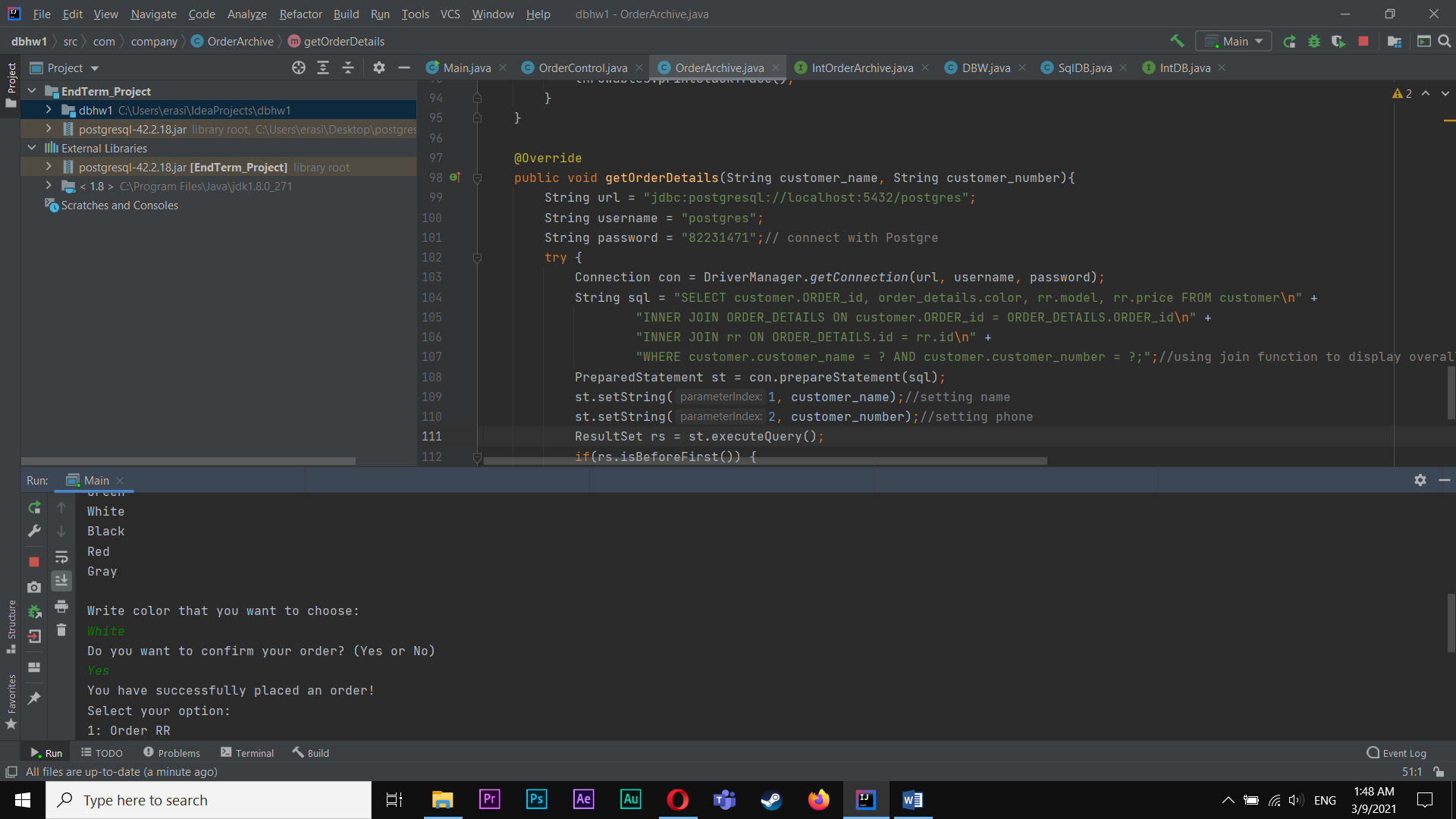
package com.company;  
  
import java.sql.\*;  
  
public class OrderArchive implements IntOrderArchive {  
 private final IntDB db;  
  
 public OrderArchive(IntDB db) {  
 this.db = db;  
 }  
  
 @Override  
 public String getProductList() { // for output car’s id, model and price  
 String url = "jdbc:postgresql://localhost:5432/postgres";  
 String username = "postgres";  
 String password = "82231471";// connect with Postgre  
 try {  
 Connection con = DriverManager.*getConnection*(url, username, password);  
 String sql = "select \* from rr";//to see all cars  
 PreparedStatement st = con.prepareStatement(sql);  
 ResultSet rs = st.executeQuery();  
 while ((rs.next())) {//   
 System.*out*.println(rs.getInt("id") + " " + rs.getString("model") + " " + rs.getInt("price"));  
 } output car’s id, model and price  
 } catch (SQLException throwables) {//error  
 throwables.printStackTrace();  
 }  
 return null;  
 }  
  
 @Override  
 public String getColor() {  
 String url = "jdbc:postgresql://localhost:5432/postgres";  
 String username = "postgres";  
 String password = "82231471";// connect with Postgre  
 try {  
 Connection con = DriverManager.*getConnection*(url, username, password);  
 String sql = "select \* from color";//to see all colors  
 PreparedStatement st = con.prepareStatement(sql);  
 ResultSet rs = st.executeQuery();  
 while ((rs.next())) {//finding and enter color  
 System.*out*.println(rs.getString("color"));  
 }  
 } catch (SQLException throwables) {//error  
 throwables.printStackTrace();  
 }  
 return null;  
 }  
  
  
 @Override  
 public boolean createUser(String customer\_name, String customer\_number){// create User is user wants to make an order to car  
 String url = "jdbc:postgresql://localhost:5432/postgres";  
 String username = "postgres";  
 String password = "82231471";// connect with Postgre  
 try {  
 Connection con = DriverManager.*getConnection*(url, username, password);  
 String sql = "INSERT INTO customer (customer\_name, customer\_number) VALUES (?,?)";// sql function to insert information  
 PreparedStatement st = con.prepareStatement(sql);  
 st.setString(1, customer\_name);// enter in sql table customer name  
 st.setString(2, customer\_number);// enter in sql table customer phone  
 st.executeUpdate();//update sql table  
 } catch (SQLException throwables) {//error  
 throwables.printStackTrace();  
 }  
 return false;  
 }  
  
 @Override  
 public void purchaseProduct (String customer\_name, String customer\_number, int id, String color){// to purchase product  
 String url = "jdbc:postgresql://localhost:5432/postgres";  
 String username = "postgres";  
 String password = "82231471";// connect with Postgre  
 try {  
 Connection con = DriverManager.*getConnection*(url, username, password);  
 String sql = "SELECT ORDER\_id FROM customer WHERE customer\_name = ? AND customer\_number = ? ORDER BY order\_id DESC LIMIT 1;";// find customer information and order result  
 PreparedStatement st = con.prepareStatement(sql);  
 st.setString(1, customer\_name);  
 st.setString(2, customer\_number);  
 ResultSet rs = st.executeQuery();  
 if(rs.isBeforeFirst()) {// to fill order\_details  
 while (rs.next()) {  
 int order\_id = rs.getInt(1);  
 String sql2 = "INSERT INTO order\_details (order\_id, id, color) VALUES (?, ?, ?)";// sql function to insert  
 PreparedStatement st2 = con.prepareStatement(sql2);  
 st2.setInt(1, order\_id);//to enter order\_id  
 st2.setInt(2, id);//to enter car\_id  
 st2.setString(3, color);//entering color  
 st2.execute();  
 }  
 }  
 } catch (SQLException throwables) {//error  
 throwables.printStackTrace();  
 }  
 }  
  
 @Override  
 public void getOrderDetails(String customer\_name, String customer\_number){  
 String url = "jdbc:postgresql://localhost:5432/postgres";  
 String username = "postgres";  
 String password = "82231471";// connect with Postgre  
 try {  
 Connection con = DriverManager.*getConnection*(url, username, password);  
 String sql = "SELECT customer.ORDER\_id, order\_details.color, rr.model, rr.price FROM customer\n" +  
 "INNER JOIN ORDER\_DETAILS ON customer.ORDER\_id = ORDER\_DETAILS.ORDER\_id\n" +  
 "INNER JOIN rr ON ORDER\_DETAILS.id = rr.id\n" +  
 "WHERE customer.customer\_name = ? AND customer.customer\_number = ?;";//using join function to display overall info  
 PreparedStatement st = con.prepareStatement(sql);  
 st.setString(1, customer\_name);//setting name  
 st.setString(2, customer\_number);//setting phone  
 ResultSet rs = st.executeQuery();  
 if(rs.isBeforeFirst()) {  
 while (rs.next()) {//finding name  
 System.*out*.println(rs.getInt("order\_id") + ": " + rs.getString("model") + ", " + rs.getInt("price")+" dollars - " + "color: " +rs.getString("color")); // for output order id, car’s model, price and color  
 }  
 }  
 } catch (SQLException throwables) {//error  
 throwables.printStackTrace();  
 }  
 }  
}

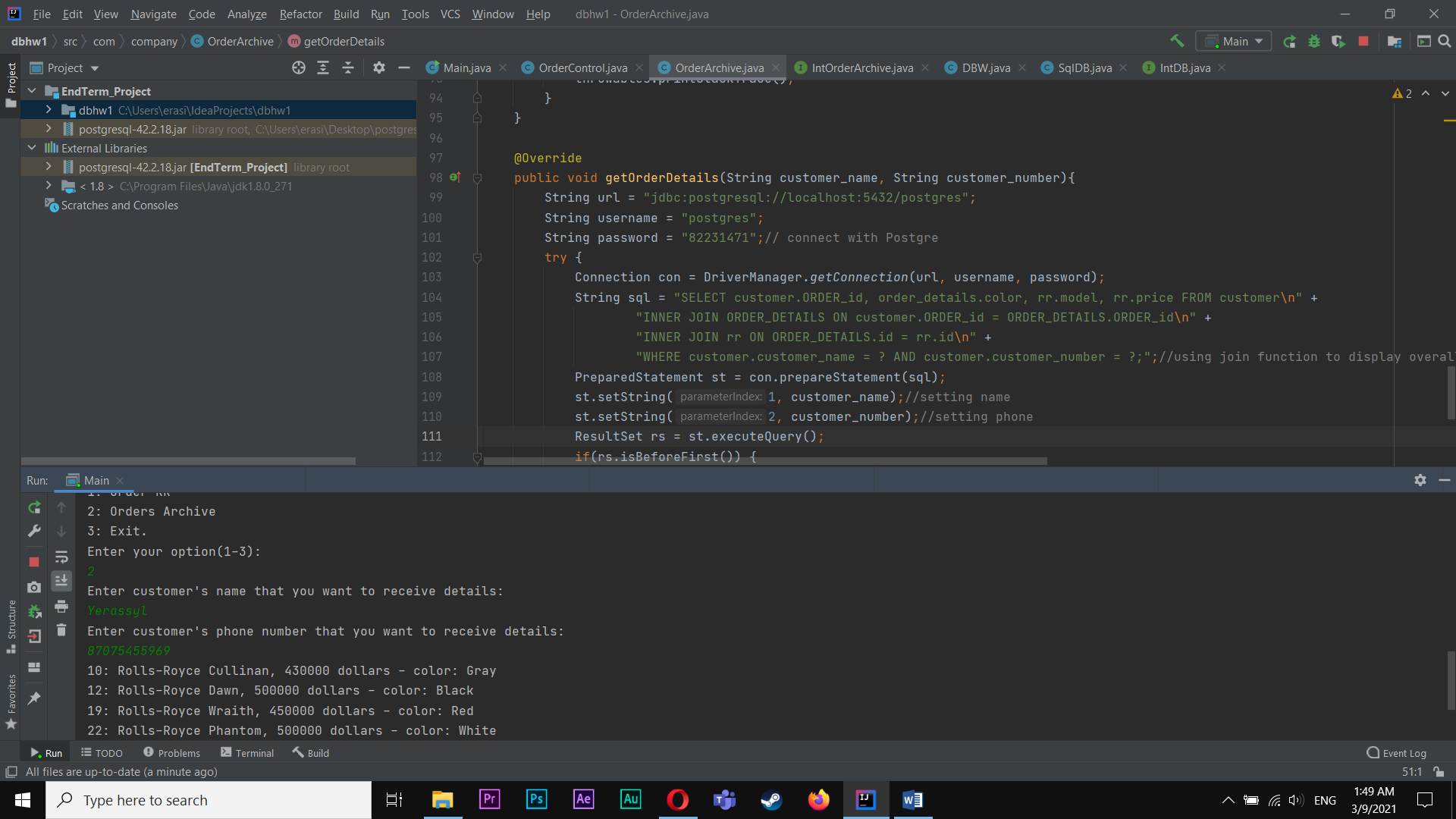
package com.company;  
  
import java.util.Scanner;  
import java.util.InputMismatchException;  
  
public class DBW { //application (code which output and input all variables)  
 private OrderControl controller;  
 private Scanner input;  
  
 public DBW(OrderControl controller) {  
 this.controller = controller;  
 input = new Scanner(System.*in*); //Scanner (input)  
 }  
  
 public void start() {  
 while (true) {  
 System.*out*.println("Select your option:\n" +  
 "1: Order RR\n" + "2: Orders Archive\n" + "3: Exit.");// List  
 try {  
 System.*out*.println("Enter your option(1-3): ");//ask to choose what to do  
 int choice = input.nextInt();  
 if (choice == 1) {//Purchase RR  
 option1();  
 } else if (choice == 2) {//See order archive (what you ordered)  
 option2();  
 } else {//Exit  
 break;  
 }  
 } catch (InputMismatchException e) {//error  
 System.*out*.println("Please choose integer from 1 to 3,check list");  
 input.nextLine();//input if there if error  
 } catch (Exception e) {  
 System.*out*.println(e);  
 }  
  
 }  
 }  
  
 public void option1() {//Purchase RR  
 System.*out*.println("Enter your name: ");//ask to enter customer\_name  
 String customer\_name = input.next();  
 System.*out*.println("Enter your phone number: ");// ask to enter customer\_number  
 String customer\_number = input.next();  
 controller.createUser(customer\_name, customer\_number);// create user in sql  
 System.*out*.println(controller.getProductList());  
 System.*out*.println("Enter id number of the product that you want to purchase: ");//ask to enter car's id  
 int id = input.nextInt();  
 System.*out*.println(controller.getColor());  
 System.*out*.println("Write color that you want to choose:");  
 String color = input.next(); // enter color  
 controller.purchaseProduct(customer\_name, customer\_number, id, color);//enter in sql information  
 System.*out*.println("Do you want to confirm your order? (Yes or No)");// conform customer request  
 String confirm = input.next();  
 if (confirm.equals("Yes")) {  
 System.*out*.println("You have successfully placed an order!");  
 }  
 }  
  
 public void option2() {// See order  
 System.*out*.println("Enter customer's name that you want to receive details: ");// ask to enter customer\_name  
 String customer\_name = input.next();  
 System.*out*.println("Enter customer's phone number that you want to receive details: ");//ask to enter customer number  
 String customer\_number= input.next();  
 controller.getOrderDetails(customer\_name, customer\_number);//getter to see order  
 }  
}

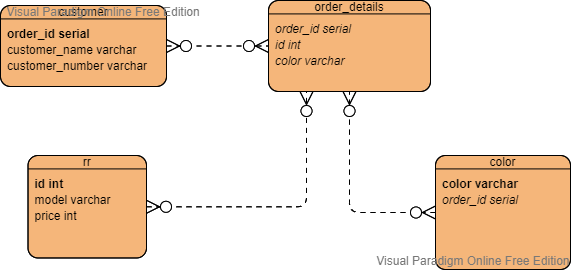
The following code lines are dedicated to make connection through SQL Database:

package com.company;  
  
import java.sql.\*;  
  
public interface IntDB {  
 void getConnection() throws SQLException, ClassNotFoundException;// to connect DB  
}

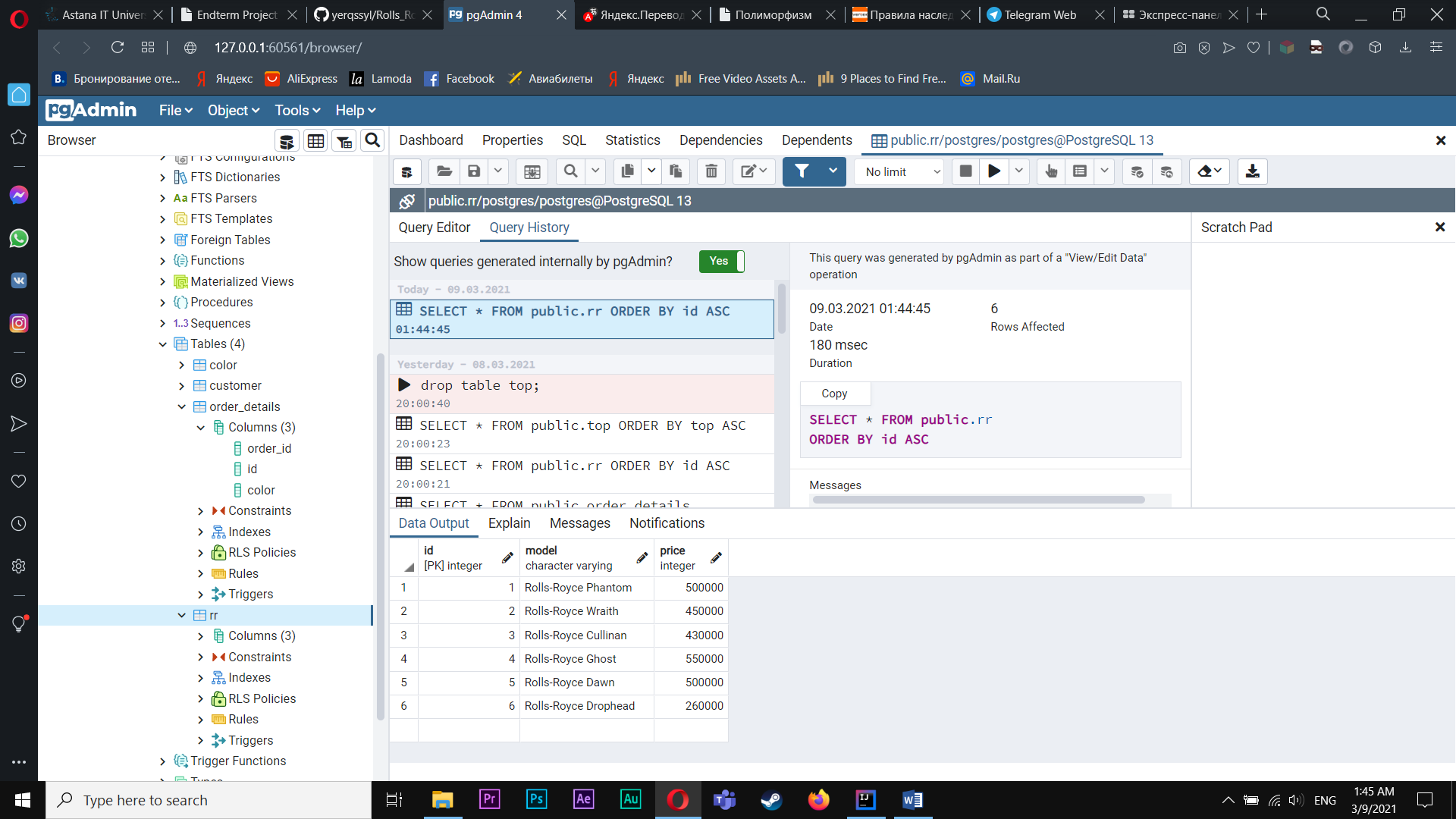
package com.company;  
  
import java.sql.\*;  
  
public class SqlDB implements IntDB{  
 @Override  
 public void getConnection() throws SQLException, ClassNotFoundException {  
 String url = "jdbc:postgresql://localhost:5432/postgres"; //url to DB  
 String username = "postgres";//username  
 String password = "8223147121";//password of DB// connect with Postgre  
 try {  
 Class.*forName*("or" + "g.postgresql.Driver");  
 Connection con = DriverManager.*getConnection*(url, username, password);  
 Statement statement = con.createStatement();  
 System.*out*.println("Connection done!");  
 } catch (SQLException exception) {  
 exception.printStackTrace();  
 }  
 }  
}

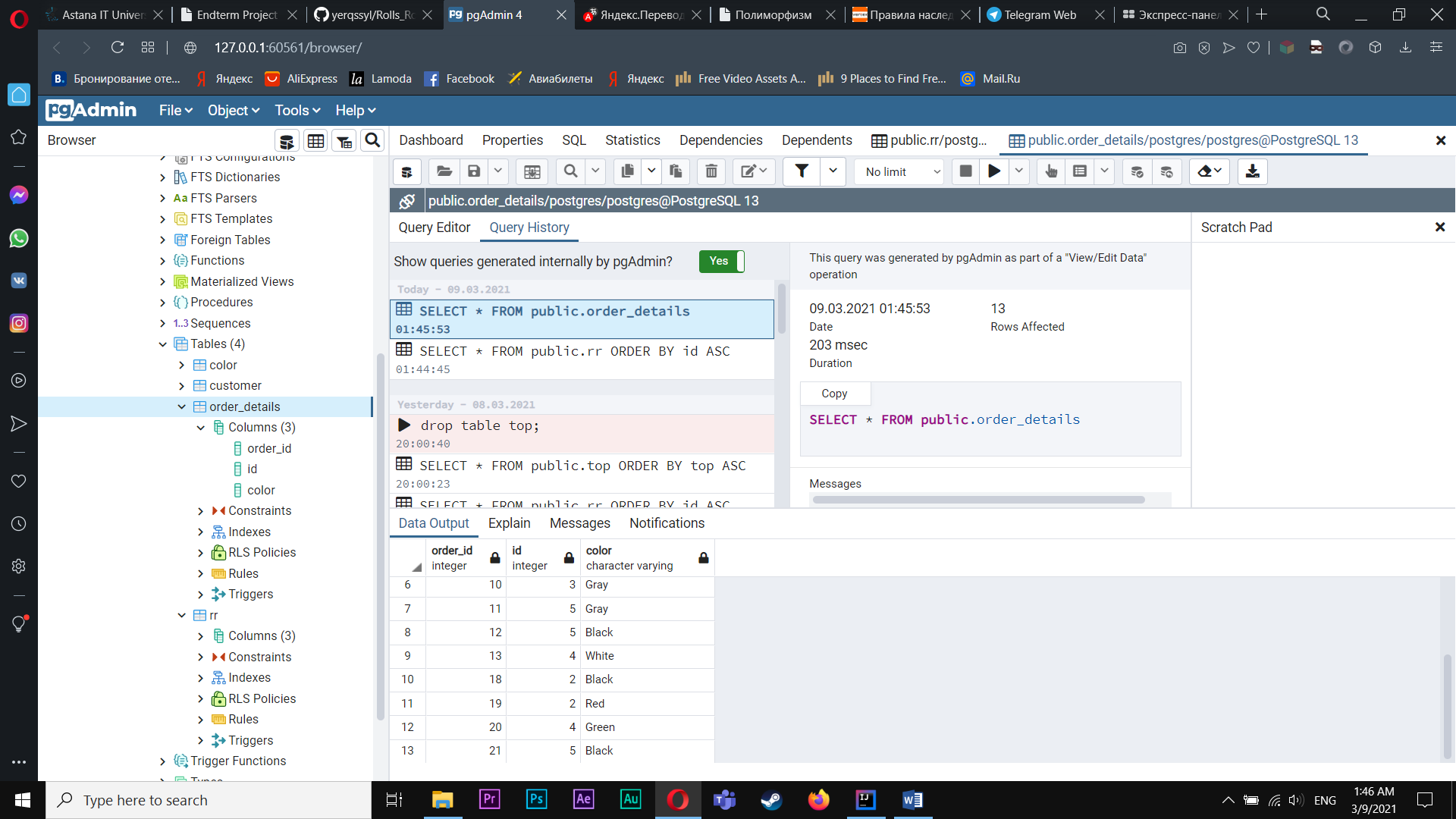


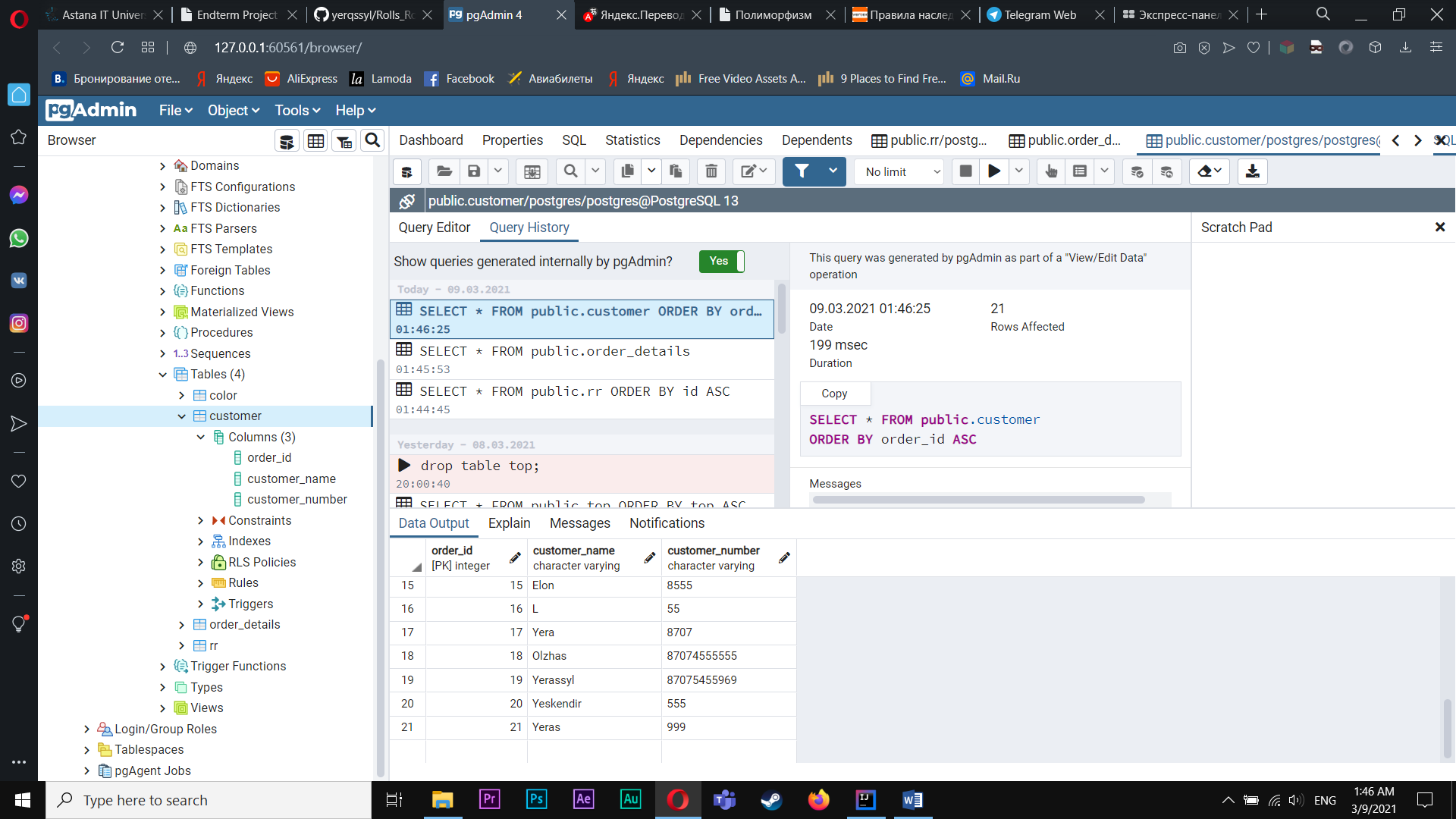




Here you are screenshots from PostgreSQL, to begin we have started with ER Diagram. After finishing with ERD we made DB in PostgreSQL:







Yerassyl Nurgali, from IT-2006, suggested me to team-up, because we have done several projects together and we knew each other well and what we are capable of, on the other hand the required said that we can create teams up to 2 students, so that was great opportunity. Yerassyl suggested to create website where you will be able to order cars, in our case it was Rolls Royce. He was truly inspired by the car ordering websites. After the topic was defined I’ve sent my teammate the ER Diagram (ERD). Yerassyl was responsible to create Database, because he had more experience than I did, we took right to code together at home of Yerassyl’s. Needless to say that this way it will be so much easier to interact and work rather that via online services. We have equally divided responsibilities, each of us did that what he can do the best. Last night before the deadline we have our code written and ready and we were about to submit it. But then suddenly our groupmate informed whole group that we have some essential tasks, that we unfortunately didn’t include, we missed some vital points while doing the project. At this point I am just wondering how did we forgot such an important part? “*To inform teacher about our teammates*”. To be honest we wouldn’t notice that missed parts if our groupmate wouldn’t tell about it. I wrote you email as soon as I noticed the message about requirement. We truly hope that you will understand our situation and accept the project, and we will try to never repeat those mistakes.

P.s. This week was super busy, we had our endterms and finals upcoming and many other deadlines expiring, so we started working on our Java Projects only 3 days earlier.