Relational Databases with MySQL Week 6 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In Eclipse, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your Java project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

This week you will be working together as a **team** to create a full CRUD application.

Your console CRUD application will need to use a database to store all the application data.

As a team, decide what you want your project to do. Get instructor approval early in the week before beginning development.

You need to have at least 3 entities.

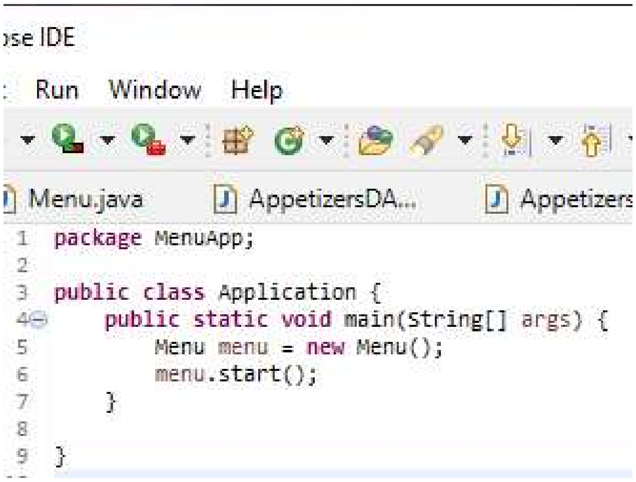
Users should be able to interact via the console (i.e. Scanner(System.in)))

Use git to collaborate.

Everyone will be graded on their individual contributions.

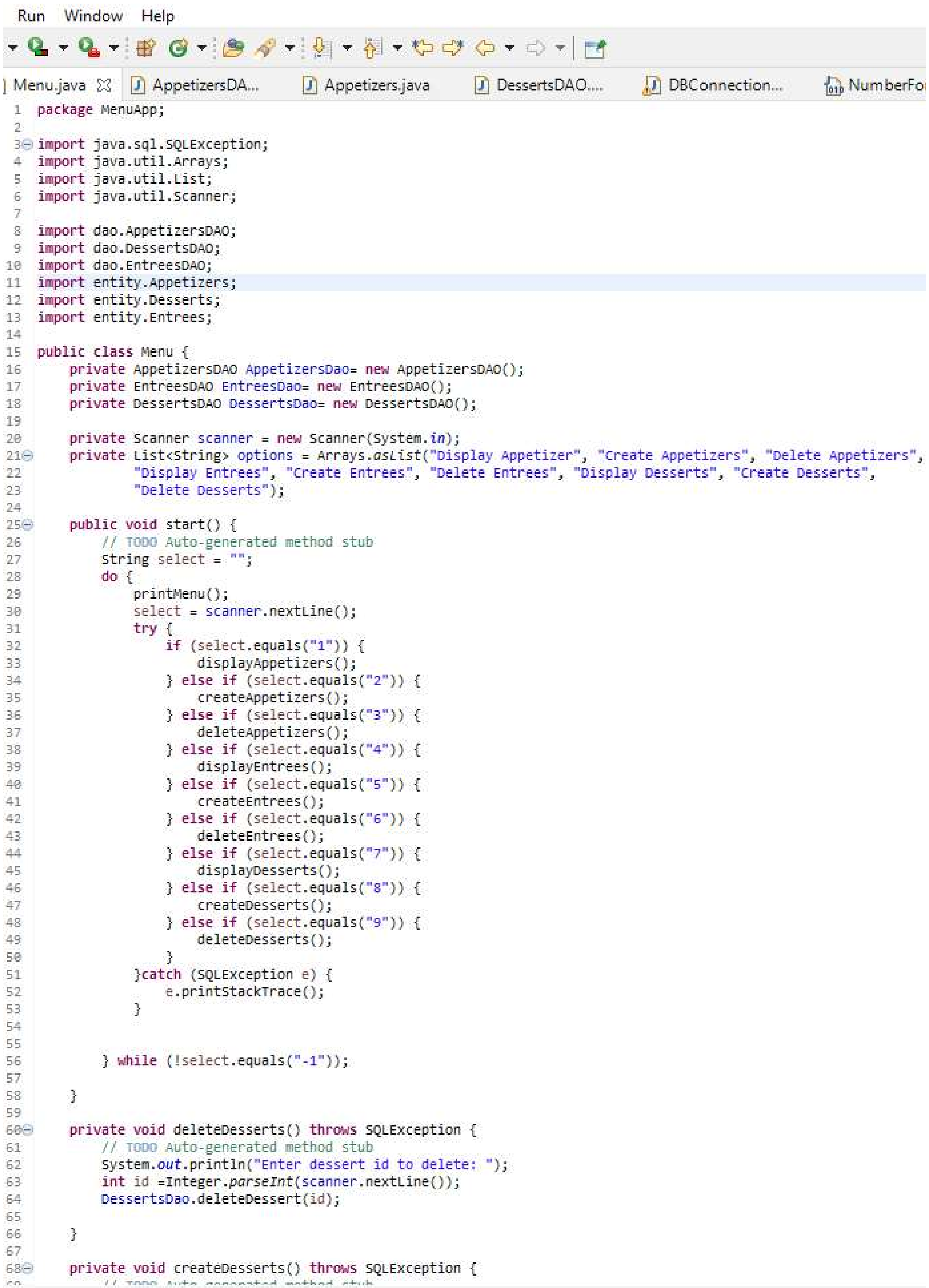
Screenshots of Code:

Application:

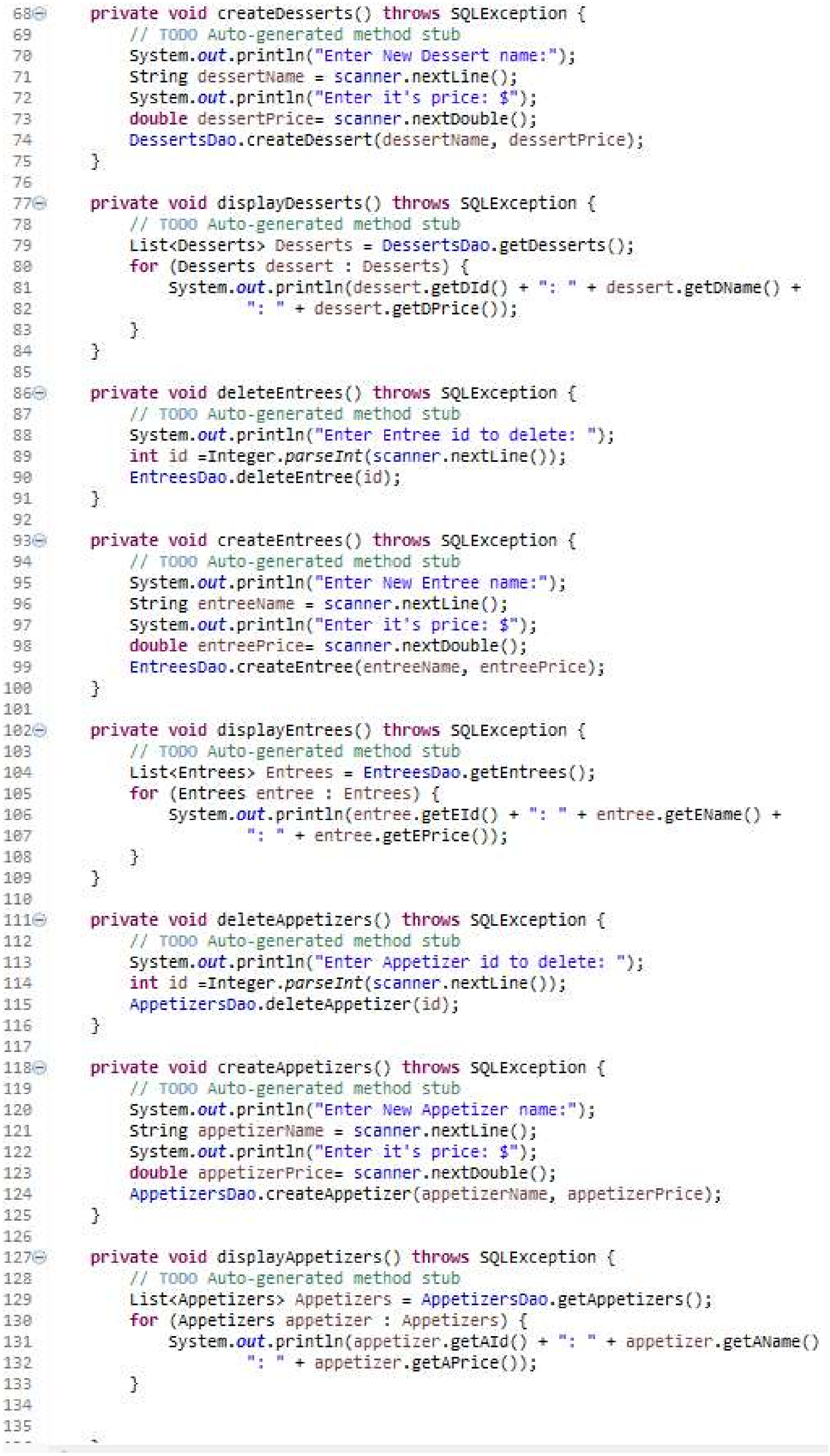


```java
package MenuApp;

public class Application {
    public static void main(String[] args) {
        Menu menu = new Menu();
        menu.start();
    }

}
```

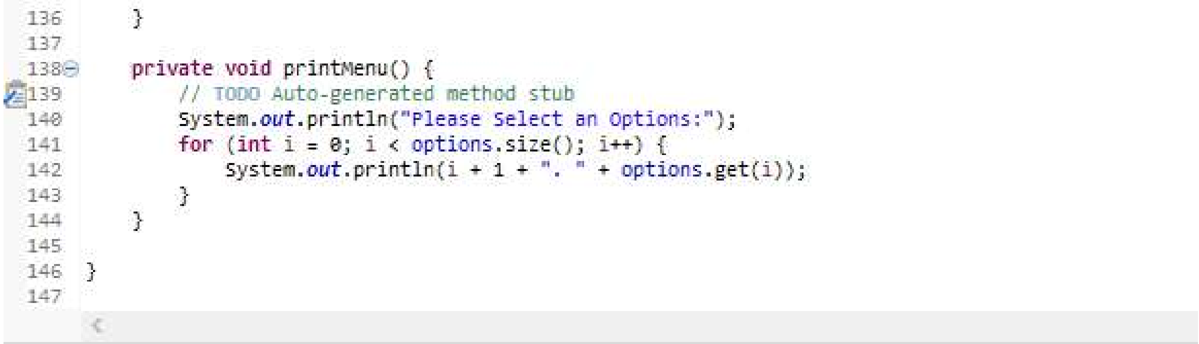
Menu:



```java
package MenuApp;

import java.sql.SQLException;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;

import dao.AppetizersDAO;
import dao.DessertsDAO;
import dao.EntreesDAO;
import entity.Appetizers;
import entity.Desserts;
import entity.Entrees;

public class Menu {
    private AppetizersDAO AppetizersDao= new AppetizersDAO();
    private EntreesDAO EntreesDao= new EntreesDAO();
    private DessertsDAO DessertsDao= new DessertsDAO();

    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList("Display Appetizer", "Create Appetizers", "Delete Appetizers",
            "Display Entrees", "Create Entrees", "Delete Entrees", "Display Desserts", "Create Desserts",
            "Delete Desserts");

    public void start() {
        // TODO Auto-generated method stub
        String select = "";
        do {
            printMenu();
            select = scanner.nextLine();
            try {
                if (select.equals("1")) {
                    displayAppetizers();
                } else if (select.equals("2")) {
                    createAppetizers();
                } else if (select.equals("3")) {
                    deleteAppetizers();
                } else if (select.equals("4")) {
                    displayEntrees();
                } else if (select.equals("5")) {
                    createEntrees();
                } else if (select.equals("6")) {
                    deleteEntrees();
                } else if (select.equals("7")) {
                    displayDesserts();
                } else if (select.equals("8")) {
                    createDesserts();
                } else if (select.equals("9")) {
                    deleteDesserts();
                }
            }catch (SQLException e) {
                e.printStackTrace();
            }


        } while (!select.equals("-1"));

    }

    private void deleteDesserts() throws SQLException {
        // TODO Auto-generated method stub
        System.out.println("Enter dessert id to delete: ");
        int id =Integer.parseInt(scanner.nextLine());
        DessertsDao.deleteDessert(id);

    }

    private void createDesserts() throws SQLException {
        // TODO Auto-generated method stub
```



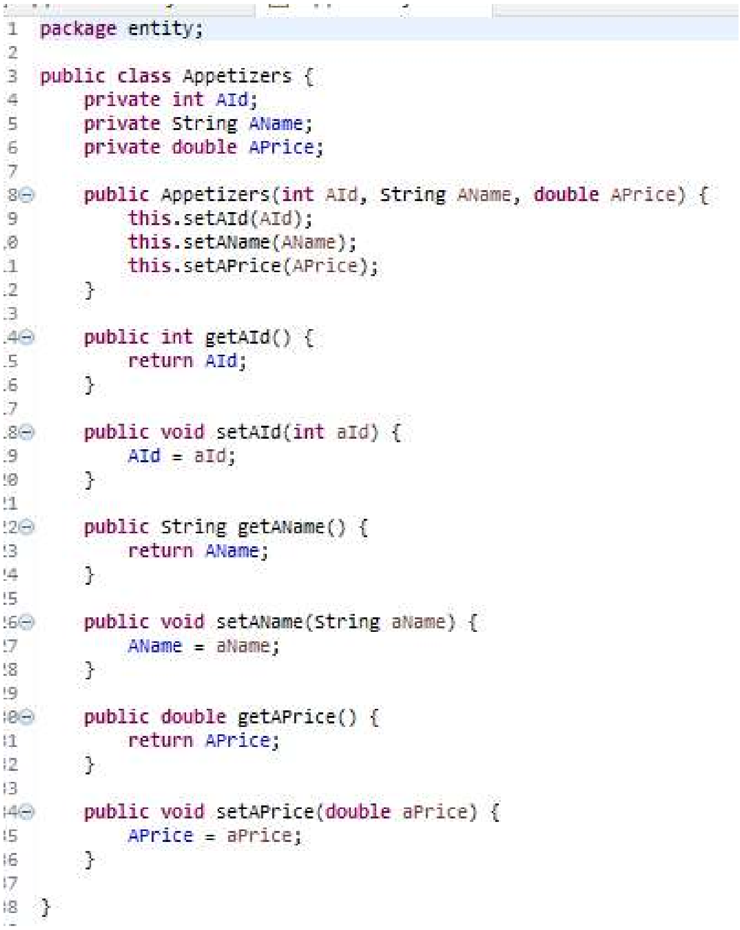
```java
    private void createDesserts() throws SQLException {
        // TODO Auto-generated method stub
        System.out.println("Enter New Dessert name:");
        String dessertName = scanner.nextLine();
        System.out.println("Enter it's price: $");
        double dessertPrice= scanner.nextDouble();
        DessertsDao.createDessert(dessertName, dessertPrice);
    }

    private void displayDesserts() throws SQLException {
        // TODO Auto-generated method stub
        List<Desserts> Desserts = DessertsDao.getDesserts();
        for (Desserts dessert : Desserts) {
            System.out.println(dessert.getDId() + ": " + dessert.getDName() +
                    ": " + dessert.getDPrice());
        }
    }

    private void deleteEntrees() throws SQLException {
        // TODO Auto-generated method stub
        System.out.println("Enter Entree id to delete: ");
        int id =Integer.parseInt(scanner.nextLine());
        EntreesDao.deleteEntree(id);
    }

    private void createEntrees() throws SQLException {
        // TODO Auto-generated method stub
        System.out.println("Enter New Entree name:");
        String entreeName = scanner.nextLine();
        System.out.println("Enter it's price: $");
        double entreePrice= scanner.nextDouble();
        EntreesDao.createEntree(entreeName, entreePrice);
    }

    private void displayEntrees() throws SQLException {
        // TODO Auto-generated method stub
        List<Entrees> Entrees = EntreesDao.getEntrees();
        for (Entrees entree : Entrees) {
            System.out.println(entree.getEId() + ": " + entree.getEName() +
                    ": " + entree.getEPrice());
        }
    }

    private void deleteAppetizers() throws SQLException {
        // TODO Auto-generated method stub
        System.out.println("Enter Appetizer id to delete: ");
        int id =Integer.parseInt(scanner.nextLine());
        AppetizersDao.deleteAppetizer(id);
    }

    private void createAppetizers() throws SQLException {
        // TODO Auto-generated method stub
        System.out.println("Enter New Appetizer name:");
        String appetizerName = scanner.nextLine();
        System.out.println("Enter it's price: $");
        double appetizerPrice= scanner.nextDouble();
        AppetizersDao.createAppetizer(appetizerName, appetizerPrice);
    }

    private void displayAppetizers() throws SQLException {
        // TODO Auto-generated method stub
        List<Appetizers> Appetizers = AppetizersDao.getAppetizers();
        for (Appetizers appetizer : Appetizers) {
            System.out.println(appetizer.getAId() + ": " + appetizer.getAName()
                    ": " + appetizer.getAPrice());
        }


    }
```



```
    }

    private void printMenu() {
        // TODO Auto-generated method stub
        System.out.println("Please Select an Options:");
        for (int i = 0; i < options.size(); i++) {
            System.out.println(i + 1 + ". " + options.get(i));
        }
    }

}
```

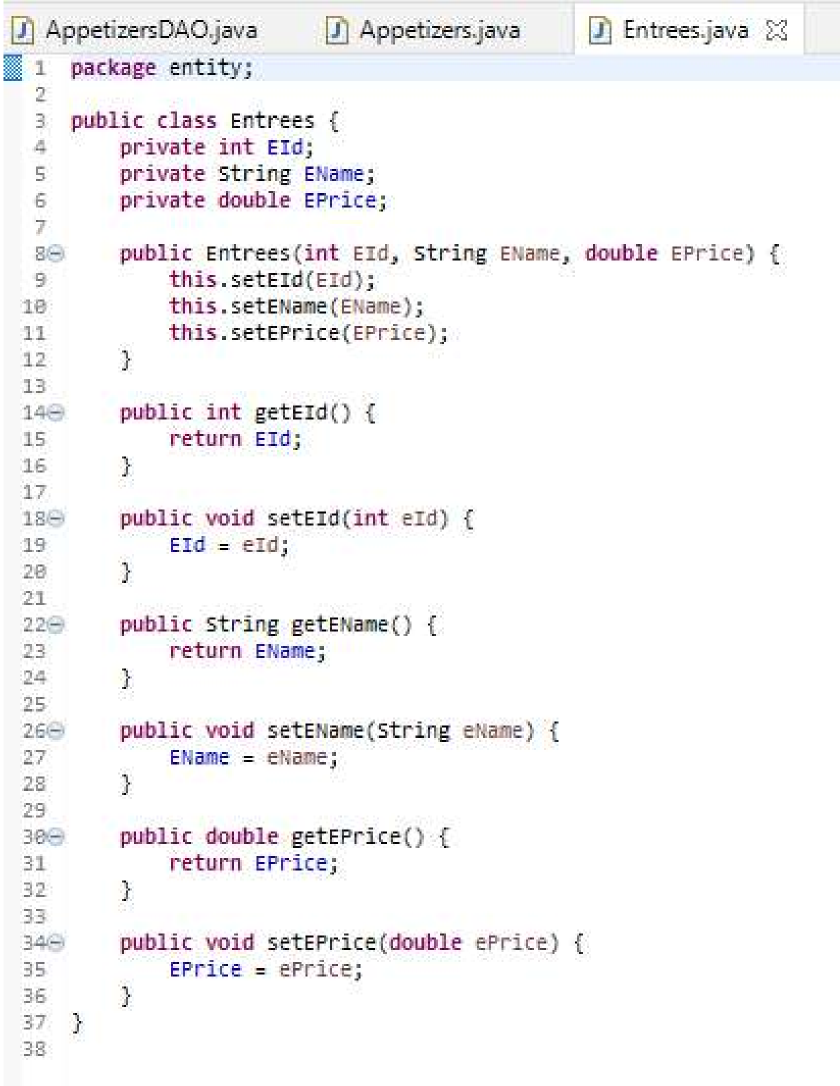
Entities:

Appetizers



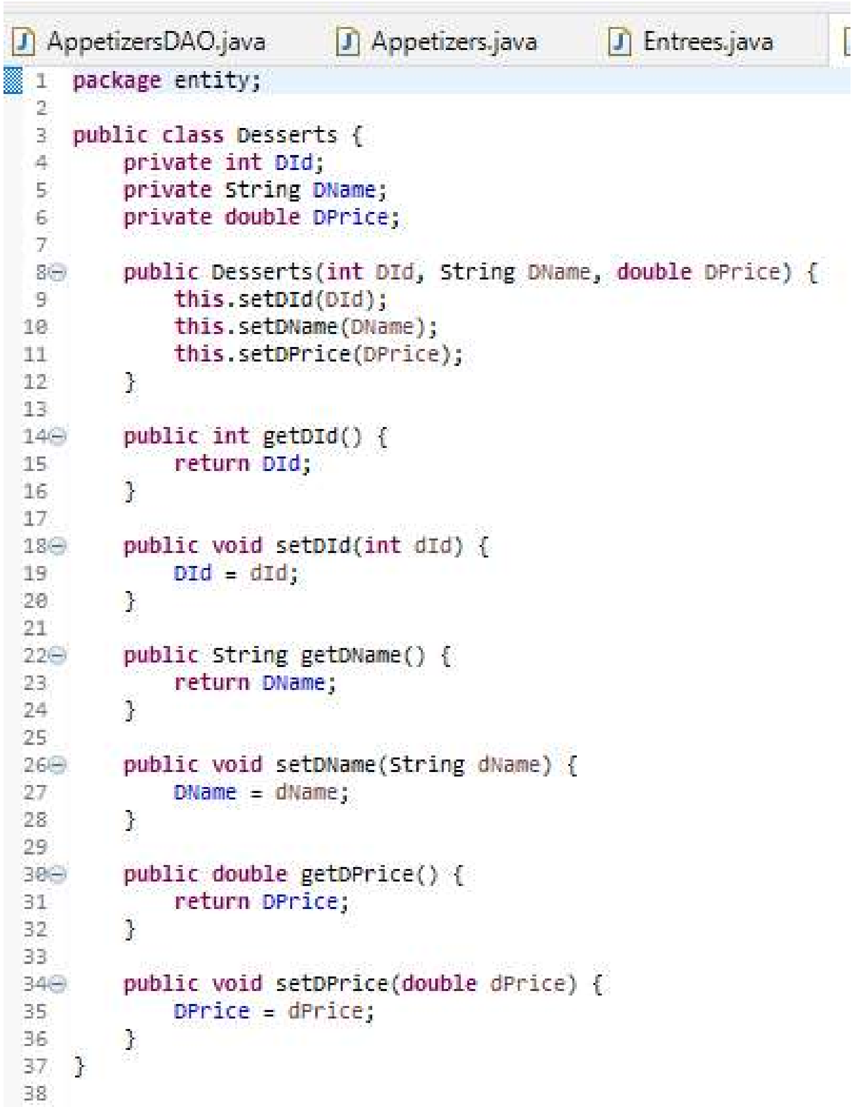
```
package entity;

public class Appetizers {
    private int AId;
    private String AName;
    private double APrice;

    public Appetizers(int AId, String AName, double APrice) {
        this.setAId(AId);
        this.setAName(AName);
        this.setAPrice(APrice);
    }

    public int getAId() {
        return AId;
    }

    public void setAId(int aId) {
        AId = aId;
    }

    public String getAName() {
        return AName;
    }

    public void setAName(String aName) {
        AName = aName;
    }

    public double getAPrice() {
        return APrice;
    }

    public void setAPrice(double aPrice) {
        APrice = aPrice;
    }

}
```

Entrees



```
package entity;

public class Entrees {
    private int EId;
    private String EName;
    private double EPrice;

    public Entrees(int EId, String EName, double EPrice) {
        this.setEId(EId);
        this.setEName(EName);
        this.setEPrice(EPrice);
    }

    public int getEId() {
        return EId;
    }

    public void setEId(int eId) {
        EId = eId;
    }

    public String getEName() {
        return EName;
    }

    public void setEName(String eName) {
        EName = eName;
    }

    public double getEPrice() {
        return EPrice;
    }

    public void setEPrice(double ePrice) {
        EPrice = ePrice;
    }
}

```

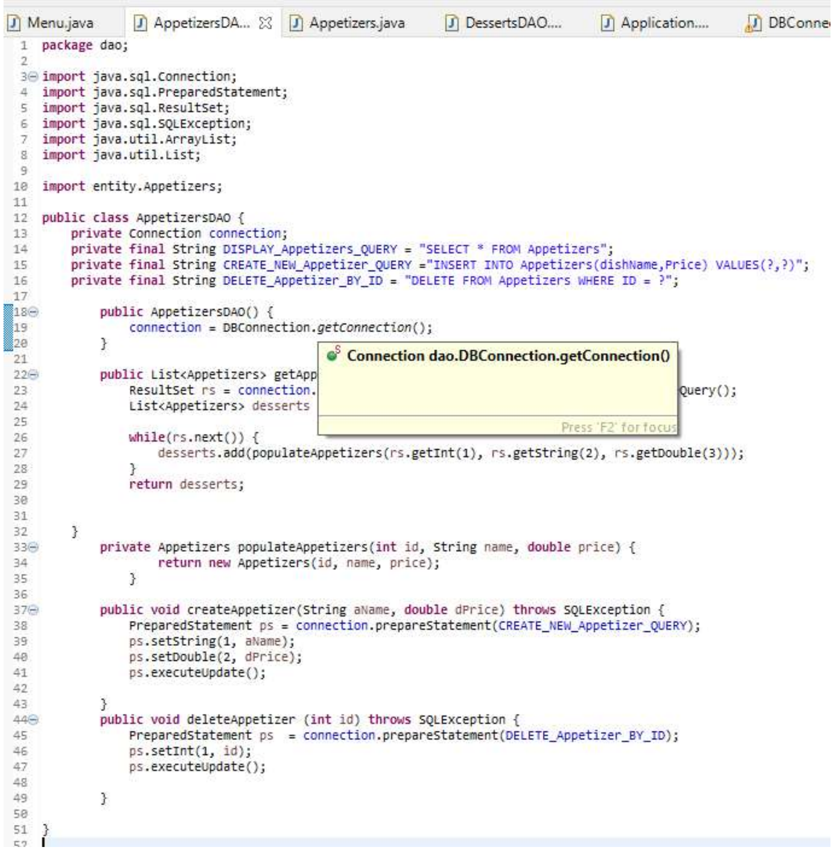
Desserts



```java
package entity;

public class Desserts {
    private int DId;
    private String DName;
    private double DPrice;

    public Desserts(int DId, String DName, double DPrice) {
        this.setDId(DId);
        this.setDName(DName);
        this.setDPrice(DPrice);
    }

    public int getDId() {
        return DId;
    }

    public void setDId(int dId) {
        DId = dId;
    }

    public String getDName() {
        return DName;
    }

    public void setDName(String dName) {
        DName = dName;
    }

    public double getDPrice() {
        return DPrice;
    }

    public void setDPrice(double dPrice) {
        DPrice = dPrice;
    }
}

```

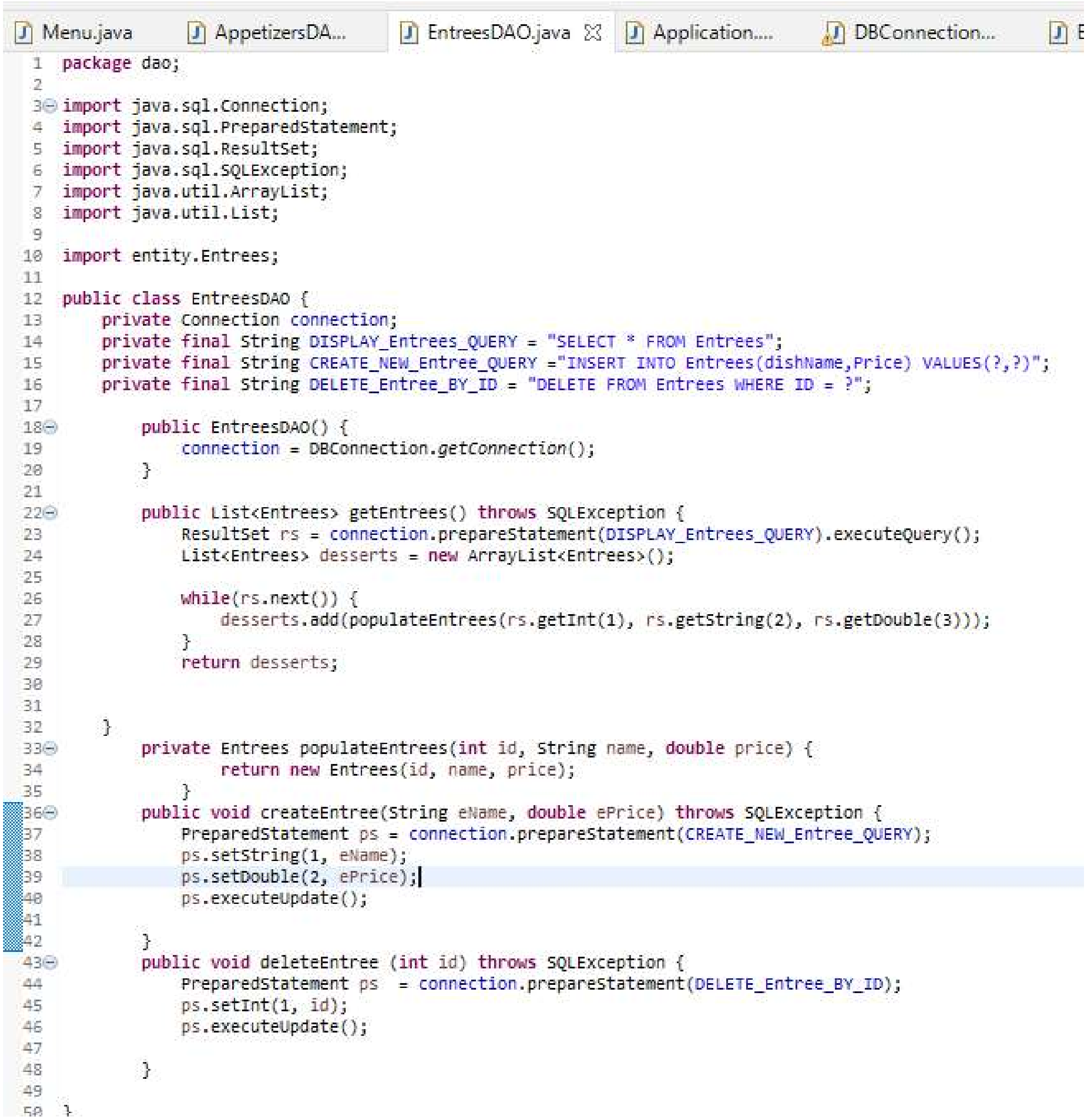
DAO:

Appetizers



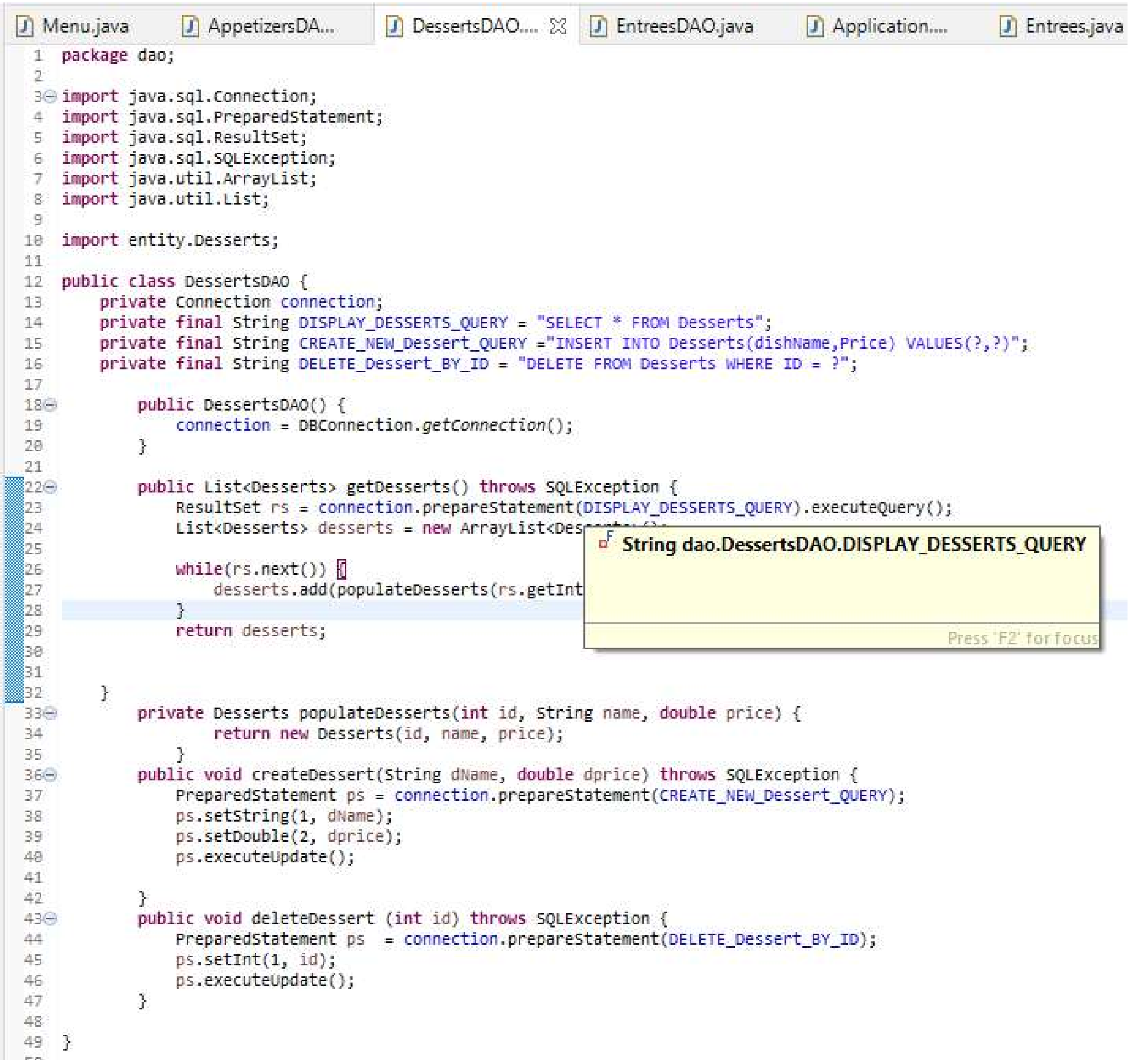
```java
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import entity.Appetizers;

public class AppetizersDAO {
    private Connection connection;
    private final String DISPLAY_Appetizers_QUERY = "SELECT * FROM Appetizers";
    private final String CREATE_NEW_Appetizer_QUERY ="INSERT INTO Appetizers(dishName,Price) VALUES(?,?)";
    private final String DELETE_Appetizer_BY_ID = "DELETE FROM Appetizers WHERE ID = ?";

        public AppetizersDAO() {
            connection = DBConnection.getConnection();
        }

        public List<Appetizers> getApp
            ResultSet rs = connection.                                              Query();
            List<Appetizers> desserts

            while(rs.next()) {
                desserts.add(populateAppetizers(rs.getInt(1), rs.getString(2), rs.getDouble(3)));
            }
            return desserts;


    }
        private Appetizers populateAppetizers(int id, String name, double price) {
                return new Appetizers(id, name, price);
            }

        public void createAppetizer(String aName, double dPrice) throws SQLException {
            PreparedStatement ps = connection.prepareStatement(CREATE_NEW_Appetizer_QUERY);
            ps.setString(1, aName);
            ps.setDouble(2, dPrice);
            ps.executeUpdate();

        }
        public void deleteAppetizer (int id) throws SQLException {
            PreparedStatement ps  = connection.prepareStatement(DELETE_Appetizer_BY_ID);
            ps.setInt(1, id);
            ps.executeUpdate();

        }

}
```

Entrees



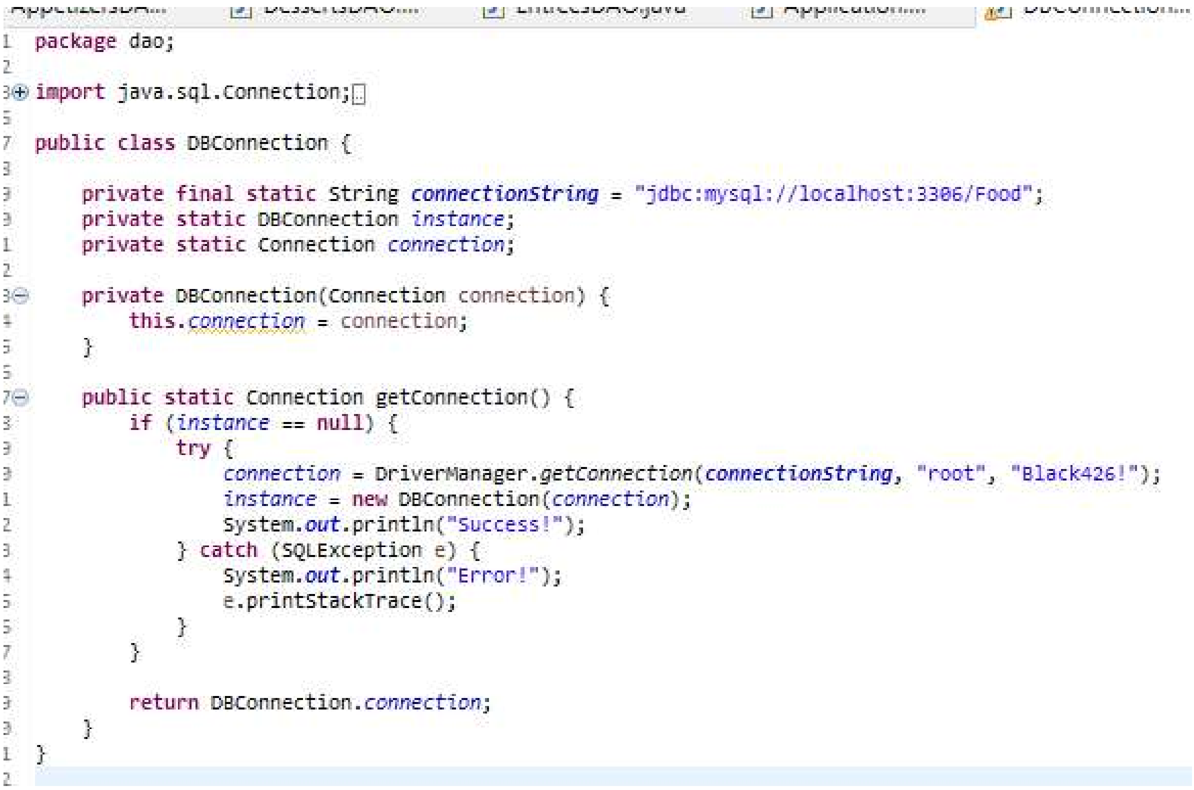
```
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import entity.Entrees;

public class EntreesDAO {
    private Connection connection;
    private final String DISPLAY_Entrees_QUERY = "SELECT * FROM Entrees";
    private final String CREATE_NEW_Entree_QUERY ="INSERT INTO Entrees(dishName,Price) VALUES(?,?)";
    private final String DELETE_Entree_BY_ID = "DELETE FROM Entrees WHERE ID = ?";

        public EntreesDAO() {
            connection = DBConnection.getConnection();
        }

        public List<Entrees> getEntrees() throws SQLException {
            ResultSet rs = connection.prepareStatement(DISPLAY_Entrees_QUERY).executeQuery();
            List<Entrees> desserts = new ArrayList<Entrees>();

            while(rs.next()) {
                desserts.add(populateEntrees(rs.getInt(1), rs.getString(2), rs.getDouble(3)));
            }
            return desserts;


    }
        private Entrees populateEntrees(int id, String name, double price) {
                return new Entrees(id, name, price);
            }
        public void createEntree(String eName, double ePrice) throws SQLException {
            PreparedStatement ps = connection.prepareStatement(CREATE_NEW_Entree_QUERY);
            ps.setString(1, eName);
            ps.setDouble(2, ePrice);
            ps.executeUpdate();

        }
        public void deleteEntree (int id) throws SQLException {
            PreparedStatement ps  = connection.prepareStatement(DELETE_Entree_BY_ID);
            ps.setInt(1, id);
            ps.executeUpdate();

        }

}
```

Desserts



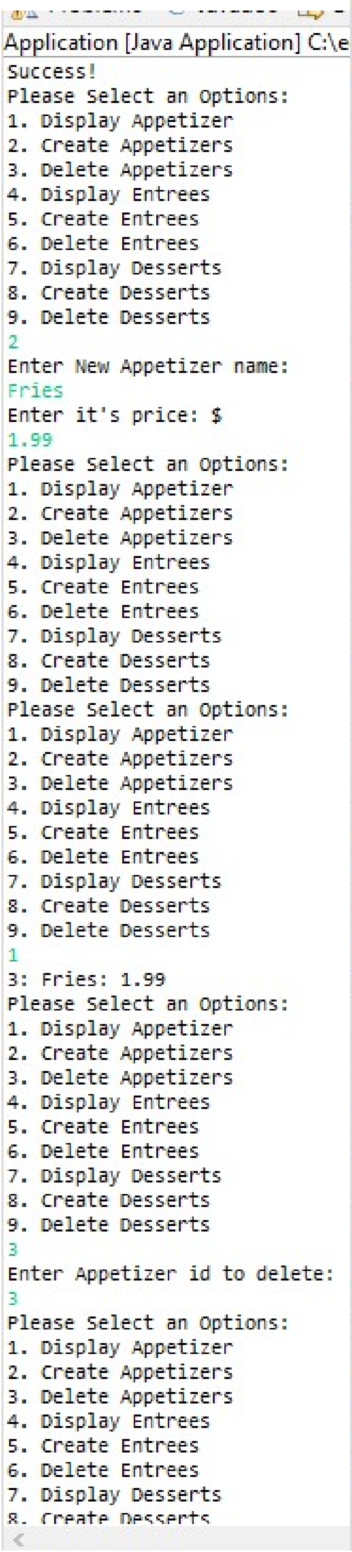
```java
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import entity.Desserts;

public class DessertsDAO {
    private Connection connection;
    private final String DISPLAY_DESSERTS_QUERY = "SELECT * FROM Desserts";
    private final String CREATE_NEW_Dessert_QUERY ="INSERT INTO Desserts(dishName,Price) VALUES(?,?)";
    private final String DELETE_Dessert_BY_ID = "DELETE FROM Desserts WHERE ID = ?";

        public DessertsDAO() {
            connection = DBConnection.getConnection();
        }

        public List<Desserts> getDesserts() throws SQLException {
            ResultSet rs = connection.prepareStatement(DISPLAY_DESSERTS_QUERY).executeQuery();
            List<Desserts> desserts = new ArrayList<Des

            while(rs.next()) {
                desserts.add(populateDesserts(rs.getInt
            }
            return desserts;


    }
        private Desserts populateDesserts(int id, String name, double price) {
                return new Desserts(id, name, price);
            }
        public void createDessert(String dName, double dprice) throws SQLException {
            PreparedStatement ps = connection.prepareStatement(CREATE_NEW_Dessert_QUERY);
            ps.setString(1, dName);
            ps.setDouble(2, dprice);
            ps.executeUpdate();

        }
        public void deleteDessert (int id) throws SQLException {
            PreparedStatement ps  = connection.prepareStatement(DELETE_Dessert_BY_ID);
            ps.setInt(1, id);
            ps.executeUpdate();
        }

}
```

DBConnection

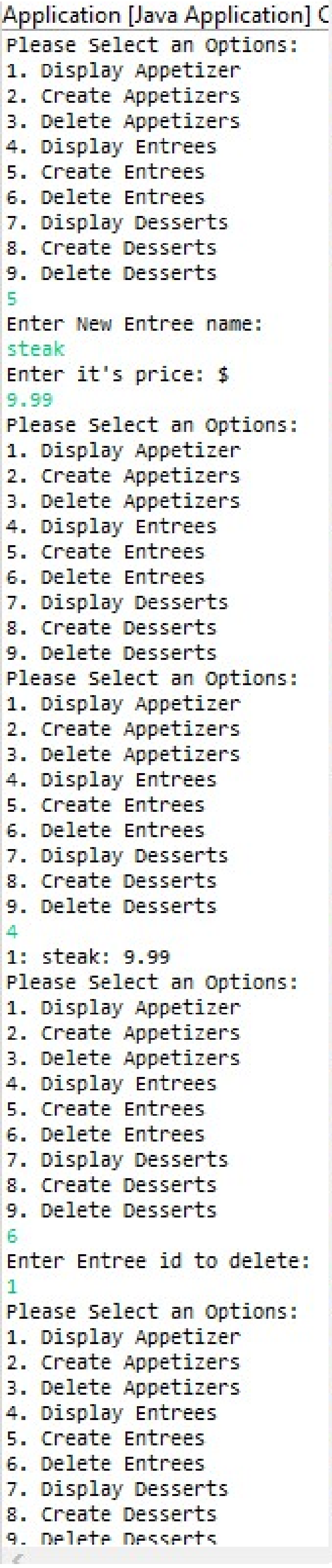


```java
package dao;

import java.sql.Connection;

public class DBConnection {

    private final static String connectionString = "jdbc:mysql://localhost:3306/Food";
    private static DBConnection instance;
    private static Connection connection;

    private DBConnection(Connection connection) {
        this.connection = connection;
    }

    public static Connection getConnection() {
        if (instance == null) {
            try {
                connection = DriverManager.getConnection(connectionString, "root", "Black426!");
                instance = new DBConnection(connection);
                System.out.println("Success!");
            } catch (SQLException e) {
                System.out.println("Error!");
                e.printStackTrace();
            }
        }

        return DBConnection.connection;
    }
}
```

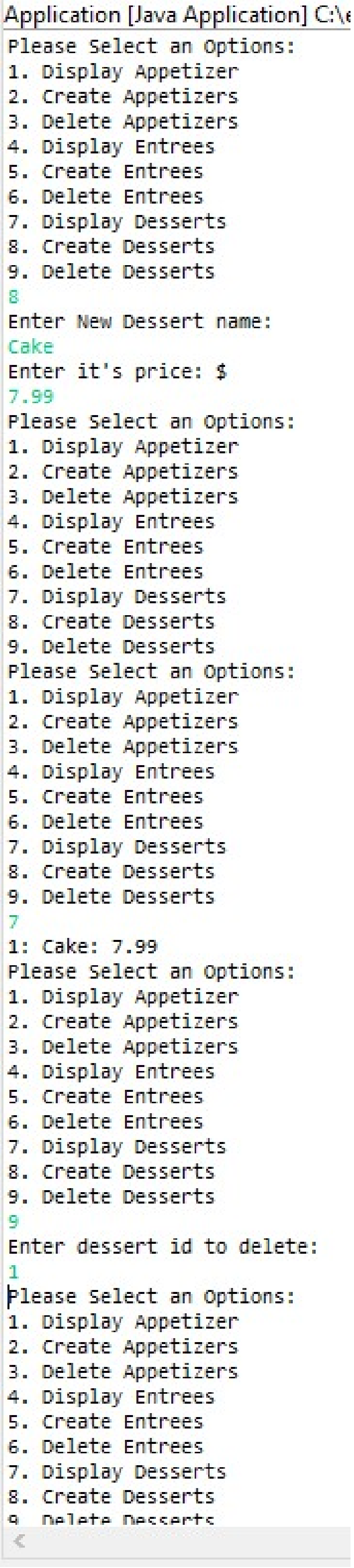
Screenshots of Running Application:



```
Success!
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
2
Enter New Appetizer name:
Fries
Enter it's price: $
1.99
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
1
3: Fries: 1.99
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
3
Enter Appetizer id to delete:
3
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
```



```
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
5
Enter New Entree name:
steak
Enter it's price: $
9.99
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
4
1: steak: 9.99
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
6
Enter Entree id to delete:
1
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
```



```
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
8
Enter New Dessert name:
Cake
Enter it's price: $
7.99
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
7
1: Cake: 7.99
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
9
Enter dessert id to delete:
1
Please Select an Options:
1. Display Appetizer
2. Create Appetizers
3. Delete Appetizers
4. Display Entrees
5. Create Entrees
6. Delete Entrees
7. Display Desserts
8. Create Desserts
9. Delete Desserts
```

URL to GitHub Repository:

https://github.com/marlon214/SQLWeek6/