



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)))`



PROMINEO TECH

Use git to collaborate.

Everyone will be graded on their individual contributions.

Screenshots of Code:

```
package Application;

public class Application {

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



PROMINEO TECH

```
package Application;

import java.sql.SQLException;

public class Menu {

    private CustomerDao CustomerDao = new CustomerDao();
    private Scanner scanner = new Scanner(System.in);
    private List<String> options = Arrays.asList(
        "Display Customers",
        "Display a Customer",
        "Add a new Customer",
        "Update an Existing Customer",
        "Delete a Customer");

    public void start() {
        String selection = "";

        do {
            printMenu();
            selection = scanner.nextLine();

            try {
                if(selection.equals("1")) {
                    displayCustomers();
                } else if (selection.equals("2")) {
                    displayCustomer();
                } else if (selection.equals("3")) {
                    addCustomer();
                } else if (selection.equals("4")) {
                    updateCustomer();
                } else if (selection.equals("5")) {
                    deleteCustomer();
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }

            System.out.println("Press enter to continue...");
            scanner.nextLine();
        } while (!selection.equals("-1"));
    }

    private void printMenu() {
        System.out.println("Please Select an Option... \n-----");
        for(int i = 0; i < options.size(); i++) {
            System.out.println(i + 1 + " " + options.get(i));
        }
    }
}
```



PROMINEO TECH

```
private void displayCustomers() throws SQLException {
    List<Customers> customers = CustomerDao.getCustomers();
    for(Customers customer : customers) {
        System.out.println(customer.getCustomersId() + ": " + "First Name: " + customer.getFirstName() + ", " +
            " Last Name: " + customer.getLastName() + ", " + "Email Address: " + customer.getEmail_address() +
            ", " + " Phone Number: " + customer.getPhone_number() + ", " + "Point's Held: " + customer.getPoints_held()
        );
    }
}

private void displayCustomer() throws SQLException {
    System.out.print("Enter the Customer ID: ");
    int id = Integer.parseInt(scanner.nextLine());
    Customers customers = CustomerDao.getCustomerById(id);
    System.out.println(customers.getCustomersId() + ": " + "First Name: " + customers.getFirstName() + ", " +
        " Last Name: " + customers.getLastName() + ", " + "Email Address: " + customers.getEmail_address() +
        ", " + " Phone Number: " + customers.getPhone_number() + ", " + "Point's Held: " + customers.getPoints_held());
}

private void addCustomer() throws SQLException {
    System.out.print("Please enter Customer's First Name: ");
    String firstName = scanner.nextLine();

    System.out.print("Please enter Customer's Last Name: ");
    String lastName = scanner.nextLine();

    System.out.print("Please enter Customer's Email Address: ");
    String emailAddress = scanner.nextLine();

    System.out.print("Please enter Customers Phone Number: ");
    String phoneNumber = scanner.nextLine();

    CustomerDao.addCustomer(firstName, lastName, emailAddress, phoneNumber);
}

public void updateCustomer() throws SQLException {
    System.out.print("Please enter the customer ID you want to update: ");
    int id = Integer.parseInt(scanner.nextLine());

    System.out.print("Please enter the First Name: ");
    String firstName = scanner.nextLine();

    System.out.print("Please enter the Last Name: ");
    String lastName = scanner.nextLine();

    System.out.print("Please enter the Email Address: ");
    String email = scanner.nextLine();

    System.out.print("Please enter the Phone Number: ");
    String phoneNumber = scanner.nextLine();

    CustomerDao.updateExistingCustomer(id, firstName, lastName, email, phoneNumber);
}

public void deleteCustomer() throws SQLException {
    System.out.print("Enter Customer ID to delete: ");
    int id = Integer.parseInt(scanner.nextLine());
    CustomerDao.DeleteCustomerByID(id);
}
}
```



PROMINEO TECH

```
package dao;

import java.sql.Connection;

public class DBconnection {

    private final static String URL = "jdbc:mysql://localhost:3306/reservations";
    private final static String USERNAME = "root";
    private final static String PASSWORD = "root1234";
    private static Connection connection;
    private static DBconnection instance;

    private DBconnection(Connection connection) {
        DBconnection.connection = connection;
    }

    public static Connection getConnection() {
        if(instance == null) {
            try {
                connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
                instance = new DBconnection(connection);
                System.out.println("Connection successful!");
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
        return DBconnection.connection;
    }
}

package dao;

import java.sql.Connection;

public class CustomerDao {

    private Connection connection;
    private final String GET_CUSTOMERS_QUERY = "SELECT * FROM Customers";
    private final String GET_CUSTOMER_BY_ID_QUERY = "SELECT * FROM Customers WHERE id = ?";
    private final String ADD_NEW_CUSTOMER_QUERY = "INSERT INTO Customers(first_name, last_name, email, phone_number) VALUES(?, ?, ?, ?)";
    private final String UPDATE_EXISTING_CUSTOMER_QUERY = "UPDATE Customers SET first_name = ?, last_name = ?, email = ?, phone_number = ?";
    private final String DELETE_CUSTOMER_BY_ID_QUERY = "DELETE FROM Customers WHERE id = ?";

    public CustomerDao() {
        connection = DBconnection.getConnection();
    }

    public List<Customers> getCustomers() throws SQLException {
        ResultSet rs = connection.prepareStatement(GET_CUSTOMERS_QUERY).executeQuery();
        List<Customers> customers = new ArrayList<Customers>();

        while (rs.next()) {
            customers.add(populateCustomers(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getInt(6)))
        }
        return customers;
    }

    public Customers getCustomerById(int id) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(GET_CUSTOMER_BY_ID_QUERY);
        ps.setInt(1, id);
        ResultSet rs = ps.executeQuery();
        rs.next();
        return populateCustomers(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5), rs.getInt(6));
    }

    public void addCustomer(String firstName, String lastName, String emailAddress, String phoneNumber) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(ADD_NEW_CUSTOMER_QUERY);
        ps.setString(1, firstName);
        ps.setString(2, lastName);
        ps.setString(3, emailAddress);
        ps.setString(4, phoneNumber);
        ps.executeUpdate();
    }

    public void updateExistingCustomer(int id, String firstName, String lastName, String emailAddress, String phoneNumber) throws SQLException {
        PreparedStatement ps = connection.prepareStatement(UPDATE_EXISTING_CUSTOMER_QUERY);
        ps.setInt(1, id);
        ps.setString(2, firstName);
        ps.setString(3, lastName);
        ps.setString(4, emailAddress);
        ps.setString(5, phoneNumber);
        ps.executeUpdate();
    }
}
```



PROMINEO TECH

```
        ps.executeUpdate();
    }

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

    private Customers populateCustomers(int id, String firstName, String lastName, String emailAddress, String phoneNumber, int points) {
        return new Customers(id, firstName, lastName, emailAddress, phoneNumber, points);
    }
}

package entity;

public class Customers {

    private int customersId;
    private String firstName;
    private String lastName;
    private String email_address;
    private String phone_number;
    private int points_held;

    public Customers(int customerId, String firstName, String lastName, String email_address, String phone_number, int points_held) {
        this.setCustomersId(customerId);
        this.setFirstName(firstName);
        this.setLastName(lastName);
        this.setEmail_address(email_address);
        this.setPhone_number(phone_number);
        this.setPoints_held(points_held);
    }

    public int getCustomersId() {
        return customersId;
    }

    public void setCustomersId(int customersId) {
        this.customersId = customersId;
    }

    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 getEmail_address() {
        return email_address;
    }

    public void setEmail_address(String email_address) {
        this.email_address = email_address;
    }
}
```



PROMINEO TECH

```
public void setEmail_address(String email_address) {  
    this.email_address = email_address;  
}  
  
public String getPhone_number() {  
    return phone_number;  
}  
  
public void setPhone_number(String phone_number) {  
    this.phone_number = phone_number;  
}  
  
public int getPoints_held() {  
    return points_held;  
}  
  
public void setPoints_held(int points_held) {  
    this.points_held = points_held;  
}  
}
```

Screenshots of Running Application:

```
Connection successful!  
Please Select an Option...  
-----  
1) Display Customers  
2) Display a Customer  
3) Add a new Customer  
4) Update an Existing Customer  
5) Delete a Customer
```

URL to GitHub Repository: <https://github.com/dlutrick/Hotels>