



## 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:



# PROMINEO TECH

```
App.java
FitnessApp ▸ src ▸ (default package) ▸ App ▸ main(String[]) : void
1= import java.sql.SQLException;
2 import java.text.ParseException;
3
4 public class App {
5
6     public static void main(String[] args) throws SQLException, ParseException {
7         Menu menu = new Menu();
8         menu.start();
9     }
10 }
11
12 }
13 |
```

Console

<terminated> App (5) [Java Application] C:\Program Files\Java\jre1.8.0\_231\bin\javaw.exe (Feb 2, 2020, 8:14:22 PM)  
Hello World



# PROMINEO TECH

Menu.java

FitnessApp > src > (default package) > Menu

```
1 import java.sql.Date;
16
17 public class Menu {
18
19     private ClientsDao clientsDao = new ClientsDao();
20     private ClassesDao classesDao = new ClassesDao();
21     private InstructorsDao instructorDao = new InstructorsDao();
22
23     private Scanner scanner = new Scanner(System.in);
24     private List<String> options = Arrays.asList(
25         "View List of All Classes",
26         "View List of All Instructors",
27         "View List of All Clients",
28         "View Clients Within a Specific Class",
29         "Update Class Day/Time",
30         "Delete Class",
31         "Add New Class",
32         "Update Client's Class",
33         "Delete Client",
34         "Add New Client",
35         "Update Instructor Pay Rate",
36         "Delete Instructor",
37         "Add New Instructor");
38
39     public void start() throws SQLException, ParseException {
40         String selection = "";
41
42         do {
43             printMenu();
44             selection = scanner.nextLine();
45
46             try {
47                 if(selection.equals("1")) {
48                     displayClasses();
49                 } else if (selection.equals("2")) {
```



# PROMINEO TECH

```
50         viewAllInstructors();
51     } else if (selection.equals("3")) {
52         viewAllClients();
53     } else if (selection.equals("4")) {
54         viewClientsInSpecificClass();
55     } else if (selection.equals("5")) {
56         updateClassDateAndTime();
57     } else if (selection.equals("6")) {
58         deleteClass();
59     } else if (selection.equals("7")) {
60         createClass();
61     } else if (selection.equals("8")) {
62         updateClientClass();
63     } else if (selection.equals("9")) {
64         deleteClient();
65     } else if (selection.equals("10")) {
66         addNewClient();
67     } else if (selection.equals("11")) {
68         updatePayRate();
69     } else if (selection.equals("12")) {
70         deleteInstructor();
71     } else if (selection.equals("13")) {
72         addNewInstructor();
73     }
74 } catch (SQLException e) {
75     e.printStackTrace();
76 }
77
78 System.out.println("Press enter to continue...");
79 scanner.nextLine();
80 } while (!selection.equals("-1"));
81 }
```



# PROMINEO TECH

```
82
83 private void printMenu() {
84     System.out.println("Select an Option:\n-----");
85     for (int i = 0; i < options.size(); i++) {
86         System.out.println(i + 1 + " " + options.get(i));
87     }
88 }
89
90
91 private void displayClasses() throws SQLException {
92     List<Classes> classes = classesDao.getClasses();
93     for (Classes classe : classes) {
94         System.out.println(classe.getClassId() + ": " + classe.getClassType() + " Date and
95     }
96 }
97
98
99 private void deleteInstructor() throws SQLException {
100     System.out.println("Enter nstructor ID you would like to remove: ");
101     int instructor_ID = Integer.parseInt(scanner.nextLine());
102
103     instructorDao.removeInstructor(instructor_ID);
104     System.out.println("Instructor Removed");
105 }
106
107
108 private void updatePayRate() throws SQLException {
109     System.out.println("Enter Instructor ID: ");
110     int instructor_ID = Integer.parseInt(scanner.nextLine());
111     System.out.println("Enter Pay Rate (xxxx.xx): ");
112     double pay_rate = Double.parseDouble(scanner.nextLine());
113     instructorDao.updatePay(instructor_ID, pay_rate);
114
115     System.out.println("Pay Rate Updated");
116 }
117
118
119 private void addNewInstructor() throws SQLException {
120
121     System.out.println("Enter First Name: ");
122     String instructor_FN = scanner.nextLine();
123     System.out.println("Enter Last Name: ");
124     String instructor_LN = scanner.nextLine();
125     System.out.println("Enter Classes (Zumba, Yoga, ...)");
126     String classes_taught = scanner.nextLine();
127     System.out.println("Enter Pay Rate (xxxx.xx): ");
128     double pay_rate = Double.parseDouble(scanner.nextLine());
129
130     instructorDao.newInstructor(instructor_FN, instructor_LN, classes_taught, pay_rate);
131
132     System.out.println("New Instructor Added");
133 }
134
135
136
137 private void viewAllInstructors() throws SQLException {
138     List<Instructors> instructors = instructorDao.getInstructors();
139     for (Instructors instructor : instructors) {
140         System.out.println("Instructor ID: " + instructor.getInstructor_ID() + ", Instructo
141     }
142
143 }
144
```



# PROMINEO TECH

```
145
146 private void viewAllClients() throws SQLException {
147     List<Clients> clients = clientsDao.getClients();
148     for (Clients client : clients) {
149         System.out.println("Client ID:" + client.getId() + ", Client Name:" + client.getfNa
150             + ", DOB:" + client.getBirthdate() + ", Class ID:" + client.getClassId());
151     }
152 }
153
154 private void viewClientsInSpecificClass() throws SQLException {
155     System.out.println("Enter class id: ");
156     int classId = Integer.parseInt(scanner.nextLine());
157     Clients clients = clientsDao.getClientsByClassID(classId);
158     System.out.println(clients.getfName() + " " + clients.getlName());
159 }
160
161 private void updateClassDateAndTime() throws SQLException {
162     System.out.println("Enter Class ID you want to change the date and time for: ");
163     int classId = Integer.parseInt(scanner.nextLine());
164     System.out.println("Enter new date and time: ");
165     String dateAndTime = scanner.nextLine();
166     classesDao.updateClassDateAndTimeById(classId, dateAndTime);
167 }
168
169 private void deleteClass() throws SQLException {
170     System.out.println("Enter the class ID you would like to delete: ");
171     int classId = Integer.parseInt(scanner.nextLine());
172     classesDao.deleteClassById(classId);
173 }
174
175 private void createClass() throws SQLException {
176     System.out.println("Enter class type: ");
177     String classType = scanner.nextLine();
178     System.out.println("Enter the date and time of the class: ");
```



# PROMINEO TECH

```
'4
'5 private void createClass() throws SQLException {
'6     System.out.println("Enter class type: ");
'7     String classType = scanner.nextLine();
'8     System.out.println("Enter the date and time of the class: ");
'9     String dateAndTime = scanner.nextLine();
'10    classesDao.createNewClassById(classType, dateAndTime);
'11 }
'12
'13
'14 private void updateClientClass() throws SQLException {
'15     System.out.println("Enter client's new class id:");
'16     int classId = Integer.parseInt(scanner.nextLine());
'17     System.out.println("Enter client id:");
'18     int clientId = Integer.parseInt(scanner.nextLine());
'19     clientsDao.updateClientClassById(classId, clientId);
'20 }
'21
'22 private void deleteClient() throws SQLException {
'23     System.out.println("Enter client id to delete:");
'24     int id = Integer.parseInt(scanner.nextLine());
'25     clientsDao.deleteClientById(id);
'26 }
'27
'28 private void addNewClient() throws SQLException, ParseException {
'29     System.out.println("Enter first name:");
'30     String fName = scanner.nextLine();
'31     System.out.println("Enter last name:");
'32     String lName = scanner.nextLine();
'33     System.out.println("Enter birthdate (YYYY-MM-DD)");
'34     SimpleDateFormat simpleDateFormat = new SimpleDateFormat("YYYY-MM-DD");
'35     Date birthdate = (Date) simpleDateFormat.parse(scanner.nextLine());
'36     System.out.println("Enter 4-digit class ID:");
'37     int classId = Integer.parseInt(scanner.nextLine());
'38     clientsDao.createNewClient(fName, lName, birthdate, classId);
'39     ps.executeUpdate();
'40 }
'41
'42
'43
'44
'45
'46 public void deleteClassById(int classId) throws SQLException {
'47     PreparedStatement ps = connection.prepareStatement(DELETE_CLASS_QUERY);
'48     ps.setInt(1, classId);
'49     ps.executeUpdate();
'50 }
'51
'52 public void createNewClassById(String classType, String dateAndTime) throws SQLException {
'53     PreparedStatement ps = connection.prepareStatement(CREATE_NEW_CLASS_QUERY);
'54     ps.setString(1, classType);
'55     ps.setString(2, dateAndTime);
'56     ps.executeUpdate();
'57 }
'58
'59
'60
'61 private Classes populateClasses(int classId, String classType, String dateAndTime) throws SQ
'62     return new Classes(classId, classType, dateAndTime, classesDao.getClientsByClassID(class
'63
'64 }
'65
'66 public class ClassesDao {
'67
'68
'69 }
'70
```



# PROMINEO TECH

```
1 package dao;
2
3 import java.sql.Connection;
4
5 public class ClientsDao {
6
7     private Connection connection;
8     private final String GET_CLIENTS_QUERY = "SELECT * FROM clients";
9     private final String GET_CLIENTS_BY_CLASSID_QUERY = "SELECT * FROM clients WHERE class_id = ?";
10    private final String UPDATE_CLIENT_CLASS_QUERY = "UPDATE clients SET class_id = ? WHERE id = ?";
11    private final String DELETE_CLIENT_BY_ID_QUERY = "DELETE FROM clients WHERE id = ?";
12    private final String CREATE_NEW_CLIENT_QUERY = "INSERT INTO clients(client_fn, client_ln, birthdate, class_id) VALUES(????)";
13
14    public ClientsDao() {
15        connection = DBConnector.getConnection();
16    }
17
18    public List<Clients> getClients() throws SQLException {
19        ResultSet rs = connection.prepareStatement(GET_CLIENTS_QUERY).executeQuery();
20        List<Clients> clients = new ArrayList<Clients>();
21
22        while (rs.next()) {
23            clients.add(populateClients(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getDate(4), rs.getInt(5)));
24        }
25
26        return clients;
27    }
28
29    public Clients getClientsByClassID(int classID) throws SQLException {
30        PreparedStatement ps = connection.prepareStatement(GET_CLIENTS_BY_CLASSID_QUERY);
31        ps.setInt(1, classID);
32        ResultSet rs = ps.executeQuery();
33        rs.next();
34        return populateClients(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getDate(4), rs.getInt(5));
35    }
36
37    public void updateClientClassById(int classId, int clientId) throws SQLException {
38        PreparedStatement ps = connection.prepareStatement(UPDATE_CLIENT_CLASS_QUERY);
39        ps.setInt(1, classId);
40        ps.setInt(2, clientId);
41        ps.executeUpdate();
42    }
43
44    public void deleteClientById(int id) throws SQLException {
45        PreparedStatement ps = connection.prepareStatement(DELETE_CLIENT_BY_ID_QUERY);
46        ps.setInt(1, id);
47        ps.executeUpdate();
48    }
49
50    public void createNewClient(String fName, String lName, Date birthdate, int classId) throws SQLException {
51        PreparedStatement ps = connection.prepareStatement(CREATE_NEW_CLIENT_QUERY);
52        ps.setString(1, fName);
53        ps.setString(2, lName);
54        ps.setDate(3, birthdate);
55        ps.setInt(4, classId);
56        ps.executeUpdate();
57    }
58
59    private Clients populateClients(int id, String fName, String lName, Date birthdate, int classId) {
60        return new Clients(id, fName, lName, birthdate, classId);
61    }
62 }
63
64 }
```





# PROMINEO TECH

› FitnessApp › src › entity › Clients ›

```
1 package entity;
2
3 import java.sql.Date;
4
5 public class Clients {
6
7     private int id;
8     private String fName;
9     private String lName;
10    private Date birthdate;
11    private int classId;
12
13
14    public Clients (int id, String fName, String lName, Date birthdate, int classId) {
15        this.setId(id);
16        this.setfName(fName);
17        this.setlName(lName);
18        this.setBirthdate(birthdate);
19        this.setClassId(classId);
20    }
21
22
23    public int getId() {
24        return id;
25    }
26    public void setId(int id) {
27        this.id = id;
28    }
29    public String getfName() {
30        return fName;
31    }
32    public void setfName(String fName) {
33        this.fName = fName;
34    }
35 }
```



# PROMINEO TECH

```
33     this.fName = fName;
34 }
35 public String getName() {
36     return lName;
37 }
38 public void setName(String lName) {
39     this.lName = lName;
40 }
41
42
43 public Date getBirthdate() {
44     return birthdate;
45 }
46
47
48 public void setBirthdate(Date birthdate) {
49     this.birthdate = birthdate;
50 }
51
52
53 public int getClassId() {
54     return classId;
55 }
56
57
58 public void setClassId(int classId) {
59     this.classId = classId;
60 }
61
62
63 }
64
```

**Screenshots of Running Application:**

**URL to GitHub Repository:** <https://github.com/Jammersg/FitnessApp>