**Centennial College**

**COMP 228: Java Programming**

**LAB #5 - Developing Database Applications using JDBC.**

**Student:** Aryan Patel

Due Date: Week 12.

Purpose: The purpose of this Lab assignment is to:

1. Practice JDBC in Java Applications
2. Develop a GUI Java application with data access capabilities

References: Read the textbook, ppt slides, and consult references (if any).

This material provides the necessary information you need to complete the exercises.

Be sure to read the following general instructions carefully:

This lab should be completed individually by all the students.

YOU NEED TO SUBMIT THE FOLLOWING 2 DOCUMENTS IN THE DROPBOX TITLED LAB5:

1. THE FIRST ONE IS A WORD DOCUMENT. USE THIS DOCUMENT AND ADD SCREEN SHOTS OF THE RUNNING STATE OF EACH EXERCISE (If there are more than 1 exercise). DO NOT DELETE THE QUESTIONS. THE SCREEN SHOTS SHOULD FOLLOW EACH QUESTION AND COVER ALL THE ASPECTS/FUNCTIONALITIES OF EACH EXERCISE. AFTER THE SCREEN SHOTS PLEASE COPY THE CODE FROM THE CODE WINDOW AND PASTE THE COMPLETE CODE. DO NOT GIVE ME SCREEN SHOTS OF THE CODE. DO NOT ZIP THIS FILE AND KEEP IT SEPARATE FROM YOUR ZIPPED PROGAM FILE.

2. SUBMIT ALSO ONE ZIPPED PROJECT FILE THAT CONTAINS ALL THE EXERISES SEPARATELY INTO THE SAME DROP BOX.

This material provides the necessary information you need to complete the exercises.

You must name your Eclipse project according to the following rule:

**YourFullName\_COMP228Labnumber**

Example: **JohSmith\_COMP228Lab5**

Each exercise should be placed in a separate package named *exercise1*, *exercise2*, etc.

Submit your assignment in a **zip file** that is named according to the following rule:

**YourLastName\_COMP228Labnumber.zip**

Example: **JohSmith\_COMP228Lab5.zip**

Apply the naming conventions for variables, methods, classes, and packages:

- *variable names* start with a *lowercase* character

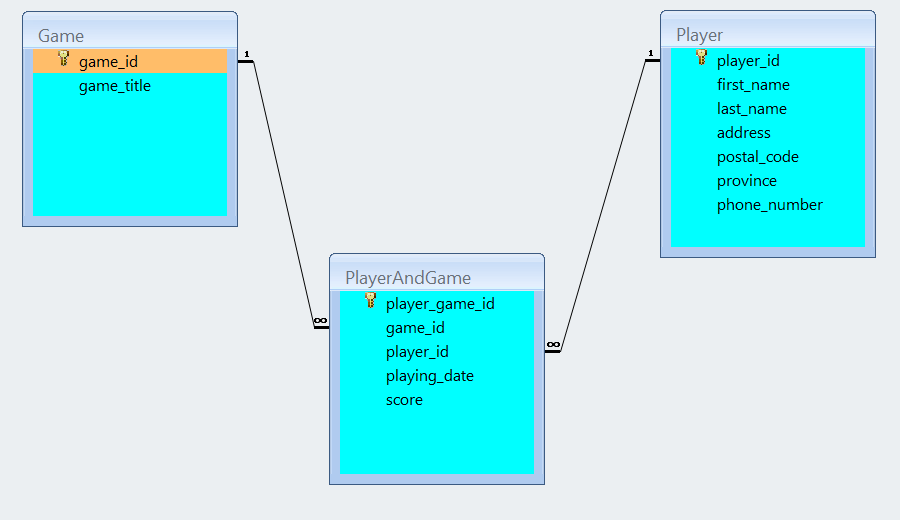
- *classes* start with an *uppercase* character

- **packages** use only *lowercase* characters

- *methods* start with a *lowercase* character

### **Exercise 1:**

Develop a GUI Java application that will allow the players to submit information about themselves and the games that they are playing on-line. The information will be stored in a simple Oracle database. The database tables are shown in the following picture:



You can use SQL Developer to create your database in Oracle server.

You should populate the table *Game* with titles of games that you have "played" during this semester.

Your GUI should provide the necessary SWING or JavaFX components that will allow the user to enter and display the data. You will use JDBC to provide the following operations:

1. *Insert* game and player information into the database.
2. *Update* the existing player information.
3. *Display* reports with player and played games information. You may use a *JTable or other components to display the reports. Allow the user to select player\_id*.

Use prepared statements to implement all database operations.

(10 marks)

**Exercise 1 Output Screenshots:**

**Insert:**

Graphical user interface

Description automatically generated

**Display:**

**Graphical user interface, application, Word

Description automatically generated**

**Update:**

**Graphical user interface, website

Description automatically generated**

**Graphical user interface, application, Word

Description automatically generated**

**Database:**

**A picture containing text, receipt

Description automatically generated**

**Exercise 1 Code:**

**Game.java:**

package application;

public class Game {

public String gameTitle;

public int gameScore;

public String gamePlayed;

public Game(String gameTitle, int gameScore, String gamePlayed) {

super();

this.gameTitle = gameTitle;

this.gameScore = gameScore;

this.gamePlayed = gamePlayed;

}

public String getGameTitle() {

return gameTitle;

}

public void setGameTitle(String gameTitle) {

this.gameTitle = gameTitle;

}

public int getGameScore() {

return gameScore;

}

public void setGameScore(int gameScore) {

this.gameScore = gameScore;

}

public String getGamePlayed() {

return gamePlayed;

}

public void setGamePlayed(String gamePlayed) {

this.gamePlayed = gamePlayed;

}

}

**Player.java:**

package application;

class Player {

public int playerId;

public String playerFirstName;

public String playerLastName;

public String playerAddress;

public String playerProvince;

public String playerPostalCode;

public String playerPhoneNumber;

public Player(int playerId, String playerFirstName, String playerLastName, String playerAddress,

String playerProvince, String playerPostalCode, String playerPhoneNumber) {

super();

this.playerId = playerId;

this.playerFirstName = playerFirstName;

this.playerLastName = playerLastName;

this.playerAddress = playerAddress;

this.playerProvince = playerProvince;

this.playerPostalCode = playerPostalCode;

this.playerPhoneNumber = playerPhoneNumber;

}

public int getPlayerId() {

return playerId;

}

public void setPlayerId(int playerId) {

this.playerId = playerId;

}

public String getPlayerFirstName() {

return playerFirstName;

}

public void setPlayerFirstName(String playerFirstName) {

this.playerFirstName = playerFirstName;

}

public String getPlayerLastName() {

return playerLastName;

}

public void setPlayerLastName(String playerLastName) {

this.playerLastName = playerLastName;

}

public String getPlayerAddress() {

return playerAddress;

}

public void setPlayerAddress(String playerAddress) {

this.playerAddress = playerAddress;

}

public String getPlayerProvince() {

return playerProvince;

}

public void setPlayerProvince(String playerProvince) {

this.playerProvince = playerProvince;

}

public String getPlayerPostalCode() {

return playerPostalCode;

}

public void setPlayerPostalCode(String playerPostalCode) {

this.playerPostalCode = playerPostalCode;

}

public String getPlayerPhoneNumber() {

return playerPhoneNumber;

}

public void setPlayerPhoneNumber(String playerPhoneNumber) {

this.playerPhoneNumber = playerPhoneNumber;

}

}

**Main.java:**

package application;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.stage.Stage;

import javafx.scene.Scene;

import javafx.scene.layout.AnchorPane;

public class Main extends Application {

@Override

public void start(Stage primaryStage) {

try {

//BorderPane root = new BorderPane();

AnchorPane root = (AnchorPane)FXMLLoader.load(getClass().getResource("Main.fxml"));

Scene scene = new Scene(root);

scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());

primaryStage.setTitle("Player Registration");

primaryStage.setScene(scene);

primaryStage.show();

} catch(Exception e) {

e.printStackTrace();

}

}

public static void main(String[] args) {

launch(args);

}

}

**MainController.java:**

package application;

import javafx.fxml.FXML;

import javafx.scene.control.Button;

import javafx.scene.control.TextField;

import java.awt.BorderLayout;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Date;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

import javax.swing.JScrollPane;

import javax.swing.JTable;

import javax.swing.table.DefaultTableModel;

import javafx.event.ActionEvent;

import javafx.scene.control.DatePicker;

public class MainController {

@FXML

private TextField playerFirstName;

@FXML

private TextField playerLastName;

@FXML

private TextField playerAddress;

@FXML

private TextField playerProvince;

@FXML

private TextField playerPostalCode;

@FXML

private TextField playerPhoneNumber;

@FXML

private TextField playerId;

@FXML

private Button updateInfo;

@FXML

private DatePicker gamePlayedDate;

@FXML

private TextField gameScore;

@FXML

private TextField gameTitle;

@FXML

private Button createPlayer;

@FXML

private Button displayPlayer;

JFrame frame, frame1;

static JTable table;

// Event Listener on Button[#updateInfo].onAction

@FXML

public void handleButtonAction(ActionEvent event) throws SQLException {

if(event.getSource() == createPlayer)

{

System.out.println("Adding record in Database !");

insertRecord();

System.out.println("Added record in Database Successfully !");

}

if(event.getSource() == displayPlayer)

{

displayRecord();

}

if(event.getSource() == updateInfo)

{

System.out.println("Updating record in Database !");

updateRecord();

System.out.println("Updated record in Database Successfully !");

}

}

public static Connection getConnection()

{

Connection con;

try {

con = DriverManager.getConnection("jdbc:oracle:thin:@199.212.26.208:1521/SQLD"COMP214\_W22\_zor\_67","password");

System.out.println("Database Connection Established");

return con;

}catch(Exception ex) {

System.out.println("Error:" + ex.getMessage());

return null;

}

}

// Insert Record in Database

private void insertRecord() throws SQLException{

Connection conn = getConnection();

// INSERT DATA IN PLAYER TABLE

String playerGeneratedColumns[] = { "PLAYER\_ID" };

String query = "INSERT INTO player (FIRST\_NAME,LAST\_NAME,ADDRESS,POSTAL\_CODE,PROVINCE,PHONE\_NUMBER) VALUES(?,?,?,?,?,?)";

PreparedStatement pstmt = conn.prepareStatement(query,playerGeneratedColumns);

pstmt.setString(1,playerFirstName.getText());

pstmt.setString(2,playerLastName.getText());

pstmt.setString(3,playerAddress.getText());

pstmt.setString(4,playerPostalCode.getText());

pstmt.setString(5,playerProvince.getText());

pstmt.setString(6,playerPhoneNumber.getText());

pstmt.execute();

ResultSet prs = pstmt.getGeneratedKeys();

// INSERT DATA IN GAME TABLE

String gameGeneratedColumns[] = { "GAME\_ID" };

String gameQuery = "INSERT INTO game (GAME\_TITLE) VALUES(?)";

PreparedStatement gpstmt = conn.prepareStatement(gameQuery, gameGeneratedColumns);

gpstmt.setString(1, gameTitle.getText());

gpstmt.execute();

ResultSet grs = gpstmt.getGeneratedKeys();

// INSERT DATA IN PLAYERANDGAME TABLE

if(prs != null && prs.next()){

if(grs != null && grs.next())

{

String pgQuery = "INSERT INTO playerandgame (GAME\_ID,PLAYER\_ID,PLAYING\_DATE,SCORE) VALUES(?,?,?,?)";

PreparedStatement pgpstmt = conn.prepareStatement(pgQuery);

pgpstmt.setInt(1, grs.getInt(1));

pgpstmt.setInt(2, prs.getInt(1));

pgpstmt.setDate(3, Date.valueOf(gamePlayedDate.getValue()));

pgpstmt.setString(4, gameScore.getText());

pgpstmt.execute();

}

}

}

// Update Record in Database

private void updateRecord() throws SQLException{

Connection conn = getConnection();

String pQuery = "UPDATE player SET first\_name = ?, last\_name = ?, address = ?, postal\_code = ?, province = ?, phone\_number = ? WHERE player\_id =?";

PreparedStatement statement = conn.prepareStatement(pQuery);

statement.setString(1,playerFirstName.getText());

statement.setString(2,playerLastName.getText());

statement.setString(3,playerAddress.getText());

statement.setString(4,playerPostalCode.getText());

statement.setString(5,playerProvince.getText());

statement.setString(6,playerPhoneNumber.getText());

statement.setString(7, playerId.getText());

int rowsUpdated = statement.executeUpdate();

if (rowsUpdated > 0) {

System.out.println("An existing user was updated successfully!");

}

}

// Display records from database

private void displayRecord() throws SQLException{

Connection conn = getConnection();

String columns[] = { "ID", "NAME", "ADDRESS", "POSTAL CODE", "PROVINCE", "PHONE NUMBER", "GAME TITLE", "SCORE", "DATE PLAYED"};

String pattern = "MM/dd/yyyy";

DateFormat df = new SimpleDateFormat(pattern);

frame1 = new JFrame("Player Information ");

frame1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame1.setLayout(new BorderLayout());

DefaultTableModel model = new DefaultTableModel();

model.setColumnIdentifiers(columns);

table = new JTable();

table.setModel(model);

table.setAutoResizeMode(JTable.AUTO\_RESIZE\_ALL\_COLUMNS);

table.setFillsViewportHeight(true);

JScrollPane scroll = new JScrollPane(table);

scroll.setHorizontalScrollBarPolicy(

JScrollPane.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED);

scroll.setVerticalScrollBarPolicy(

JScrollPane.VERTICAL\_SCROLLBAR\_AS\_NEEDED);

String pQuery = "SELECT p.player\_id, p.first\_name, p.address, p.postal\_code, p.province, p.phone\_number, g.game\_title, pg.score, pg.playing\_date FROM player p JOIN playerandgame pg ON p.player\_id = pg.player\_id JOIN game g ON g.game\_id = pg.game\_id";

PreparedStatement ps = conn.prepareStatement(pQuery);

ResultSet pres = ps.executeQuery();

int i = 0;

while (pres.next()) {

String player\_id = pres.getString("PLAYER\_ID");

String first\_name = pres.getString("FIRST\_NAME");

String address = pres.getString("ADDRESS");

String postal\_code = pres.getString("POSTAL\_CODE");

String province = pres.getString("PROVINCE");

String phone\_number = pres.getString("PHONE\_NUMBER");

String game\_title = pres.getString("GAME\_TITLE");

int score = pres.getInt("SCORE");

String playing\_date = df.format(pres.getDate("PLAYING\_DATE"));

model.addRow(new Object[] {player\_id, first\_name, address, postal\_code, province, phone\_number, game\_title, score, playing\_date});

i++;

}

if (i < 1) {

JOptionPane.showMessageDialog(null, "No Record Found", "Error", JOptionPane.ERROR\_MESSAGE);

}

if (i == 1) {

System.out.println(i + " Record Found");

} else {

System.out.println(i + " Records Found");

}

frame1.add(scroll);

frame1.setVisible(true);

frame1.setSize(900, 400);

}

}

**Main.fxml:**

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.DatePicker?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.text.Font?>

<AnchorPane maxHeight=*"-Infinity"* maxWidth=*"-Infinity"* minHeight=*"-Infinity"* minWidth=*"-Infinity"* prefHeight=*"400.0"* prefWidth=*"648.0"* xmlns=*"http://javafx.com/javafx/18"* xmlns:fx=*"http://javafx.com/fxml/1"* fx:controller=*"application.MainController"*>

<children>

<Label layoutX=*"19.0"* layoutY=*"18.0"* text=*"Player Information:"*>

<font>

<Font name=*"System Bold"* size=*"12.0"* />

</font>

</Label>

<Label layoutX=*"310.0"* layoutY=*"69.0"* text=*"Update Player by ID"*>

<font>

<Font name=*"System Bold"* size=*"12.0"* />

</font></Label>

<Label layoutX=*"19.0"* layoutY=*"57.0"* text=*"First Name:"* />

<Label layoutX=*"19.0"* layoutY=*"84.0"* text=*"Last Name:"* />

<Label layoutX=*"20.0"* layoutY=*"111.0"* text=*"Address:"* />

<Label layoutX=*"21.0"* layoutY=*"141.0"* text=*"Province:"* />

<Label layoutX=*"21.0"* layoutY=*"170.0"* text=*"Postal Code:"* />

<Label layoutX=*"21.0"* layoutY=*"200.0"* text=*"Phone Number:"* />

<TextField id=*"playerFirstName"* fx:id=*"playerFirstName"* layoutX=*"114.0"* layoutY=*"53.0"* />

<TextField id=*"playerLastName"* fx:id=*"playerLastName"* layoutX=*"114.0"* layoutY=*"81.0"* />

<TextField id=*"playerAddress"* fx:id=*"playerAddress"* layoutX=*"114.0"* layoutY=*"110.0"* />

<TextField id=*"playerProvince"* fx:id=*"playerProvince"* layoutX=*"114.0"* layoutY=*"139.0"* />

<TextField id=*"playerPostalCode"* fx:id=*"playerPostalCode"* layoutX=*"114.0"* layoutY=*"166.0"* />

<TextField id=*"playerPhoneNumber"* fx:id=*"playerPhoneNumber"* layoutX=*"114.0"* layoutY=*"195.0"* />

<TextField id=*"playerId"* fx:id=*"playerId"* layoutX=*"426.0"* layoutY=*"66.0"* />

<Button id=*"updatePlayer"* fx:id=*"updateInfo"* layoutX=*"486.0"* layoutY=*"98.0"* mnemonicParsing=*"false"* onAction=*"#handleButtonAction"* text=*"Update"* />

<Label layoutX=*"316.0"* layoutY=*"124.0"* text=*"Game Information:"*>

<font>

<Font name=*"System Bold"* size=*"12.0"* />

</font>

</Label>

<Label layoutX=*"324.0"* layoutY=*"157.0"* text=*"Game Title:"* />

<Label layoutX=*"325.0"* layoutY=*"188.0"* text=*"Game Score:"* />

<Label layoutX=*"324.0"* layoutY=*"219.0"* text=*"Date Played:"* />

<TextField id=*"gameTitle"* fx:id=*"gameTitle"* layoutX=*"398.0"* layoutY=*"154.0"* prefHeight=*"25.0"* prefWidth=*"174.0"* />

<TextField id=*"gameScore"* fx:id=*"gameScore"* layoutX=*"398.0"* layoutY=*"185.0"* prefHeight=*"25.0"* prefWidth=*"174.0"* />

<DatePicker fx:id=*"gamePlayedDate"* layoutX=*"399.0"* layoutY=*"216.0"* prefHeight=*"25.0"* prefWidth=*"174.0"* />

<Button id=*"createPlayer"* fx:id=*"createPlayer"* layoutX=*"356.0"* layoutY=*"335.0"* mnemonicParsing=*"false"* onAction=*"#handleButtonAction"* text=*"Create Player"* />

<Button id=*"displayPlayers"* fx:id=*"displayPlayer"* layoutX=*"468.0"* layoutY=*"335.0"* mnemonicParsing=*"false"* onAction=*"#handleButtonAction"* text=*"Display All Players"* />

</children>

</AnchorPane>

**Evaluation:**

|  |  |
| --- | --- |
| **Functionality** |  |
| Correct implementation of UI and event handling. | 35% |
| Correct implementation of JDBC | 50% |
| Comments, correct naming of variables, methods, classes, etc. | 5% |
| **Friendly input/output** | 10% |
| **Total** | 100% |

**The UI may look like the following:**

Graphical user interface

Description automatically generated

**Display All Players**

Table

Description automatically generated

**Update Operation:**

Graphical user interface, website

Description automatically generated

**After Updating Display All Players:**

Table

Description automatically generated

**Tables You May Wish to Create (May not be comprehensive, just a suggestion.)**

A picture containing diagram

Description automatically generated