

Java Swing

Casino Database

Andy Valencia



Introduction

This group project was composed of five students who were tasked to design a Database with a Java Swing program that will access the database from MySQL Workbench. The project was to include a database with at least five tables, include images of MySQL queries displaying information or making calculations, and we were given one week to complete the project. Our group decided to design a Casino database with five tables and a Java Swing program that will find information based on the membership ID of the customer. This report will focus more on the parts I designed and developed.

Tools and Graphical User Interface

- To design our Casino database tables, we used MySQL Workbench. We designed a Schema named Casino, which includes all of our five tables, as shown in **Figure 1**.

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** The "Casino" schema is selected.
- Tables:** The "employee" table is selected and highlighted with a blue border.
- Query Editor:** A query is run: `SELECT * FROM Casino.employee;`
- Result Grid:** The results of the query are displayed in a grid format. The columns are: EmployeeID, EmployeeFirstName, EmployeeLastName, EmployeePhoneNumber, and EmployeeAddress. The data includes 20 rows of employee information.

EmployeeID	EmployeeFirstName	EmployeeLastName	EmployeePhoneNumber	EmployeeAddress
1000	Jack	Hammer	5621234567	50 South Lane
1001	Matt	Tress	3238910111	7255 Fairfield Street
1002	Weston	East	3102131415	113 Cactus Lane
1003	Will	Power	2137181920	138 Victoria Road
1004	Chris P	Bacon	5622122234	255A Miller St.
1005	Doug	Hole	3103132418	986 Sunbeam St.
1006	Joy	Rider	5623271498	8840 High Point Dr.
1007	Jasmine	Rice	4408111032	8122 E. Mammoth Lane
1008	Marshaw	Mellow	6434177348	20 Fieldstone Lane
1009	Harry	Beard	3798281327	225 Fifth Dr.
1010	Mila	Davis	4694003307	486 Sinah Lane
1011	Angel	Perry	5128345359	2545 Dove Bills Road
1012	Emmett	Figueroa	6078984740	150C Woodbridge Cliff...
1013	Luis	Russell	4133883695	583 East Springs St.
1014	Jude	Foster	4135948278	978 Little Hampden Dr.
1015	Ella	Mcgee	2392209412	732 Tamworth Garden...
1016	Ryleigh	Ross	8062775932	901G East Oaks Lane
1017	Cherry	Haruno	6128673135	1260 Leaf Dr.
1018	Margot	Houghton	4752826638	5217 Chambers Drift...
1019	Abigail	Stevens	2133458538	612 Charing Cross Road
1020	Caractacus	Burke	6065316991	13B Knockturn Alley

Figure 1: This is the Employee table from our Casino Schema.

- The GUI (Graphical User Interface) was designed using IntelliJ, as shown in Figure 2.1. The compiled version of our GUI is shown in Figure 2.2.

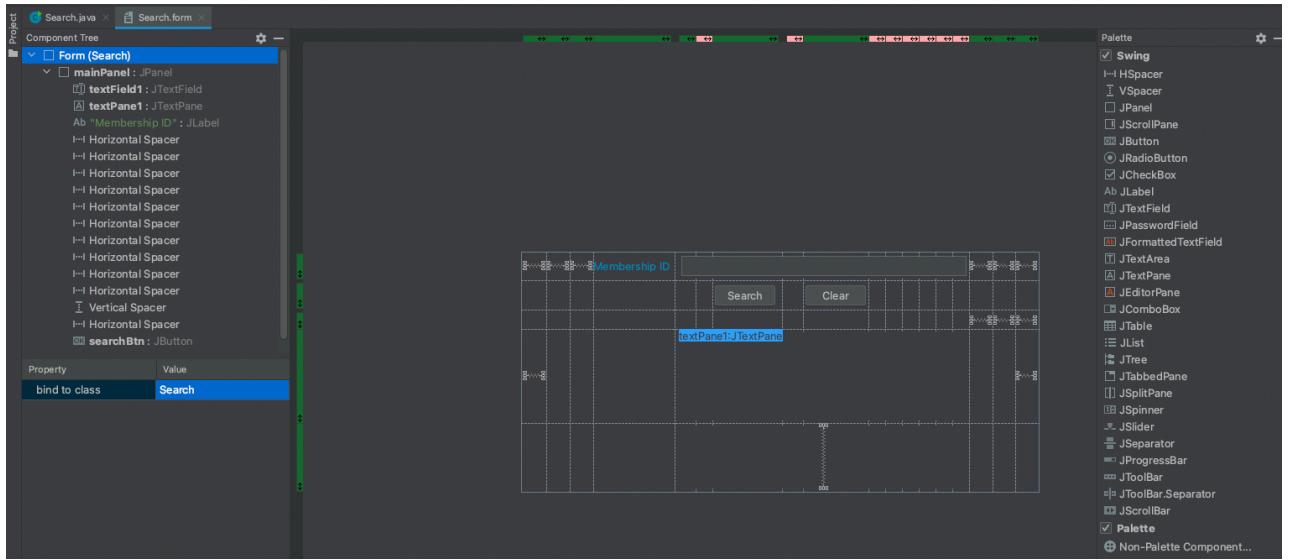


Figure 2.1: IntelliJ IDE GUI design with component tree and palette.

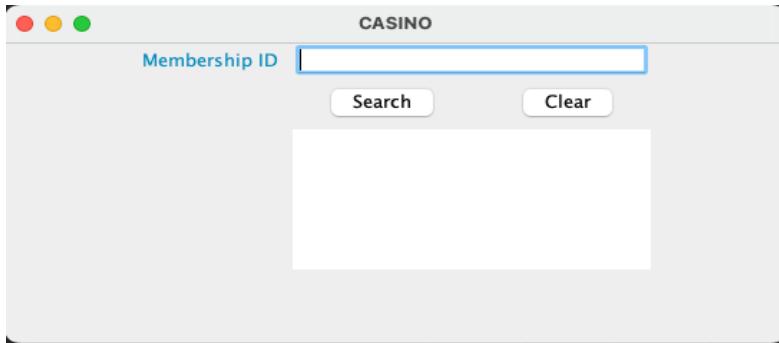


Figure 2.2: Graphical User Interface after building the project.

Java Swing

Swing is a toolkit that is part of Oracle's Java Foundation Classes. In Example 1, we demonstrate the instance variables of mainPanel, buttons, text field, and text pane. We extend JFrame in our class to display a frame with our components. My goal was to connect to our database by pressing the search button on our GUI and access our customer's information based on a member ID search. If the member is found, the customer's information will appear in our text pane, as shown in figure 3.1.

Example 1

```
public class Search extends JFrame {  
  
    private JTextField textField1;  
    private JPanel mainPanel;  
    private JTextPane textPanel;  
    private JButton clearBtn;  
    private JButton searchBtn;  
  
    public Search() {  
        setContentPane(mainPanel);  
        setTitle("CASINO");  
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);  
        setVisible(true);  
    }  
}
```

In the above example, we create an instance of JTextField, JPanel, JTextPane, and JButton. We also set our mainPanel, our title “CASINO” and make the frame visible by setting it to true.

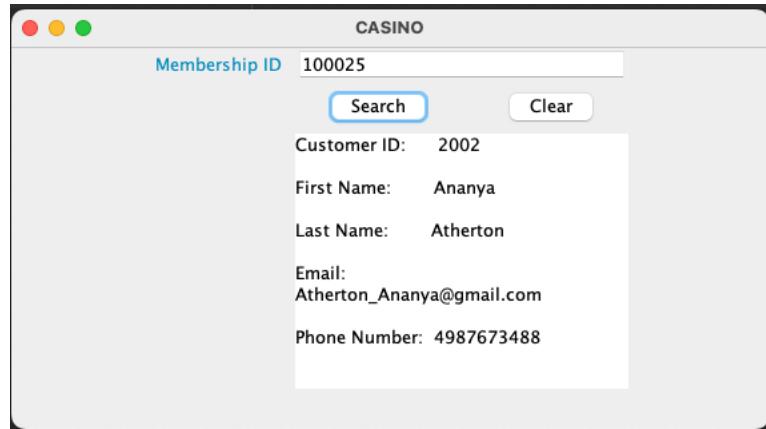


Figure 3.1: GUI displaying customer information after entering a valid Member ID and pressing the search button.

Example 2 shows how the search button was created and how we established a connection with our database. If we needed to search for a different customer, we would have to press the clear button to clear our textfiled1 and textpane1. In Example 3, we created a clearBtn action to perform a clear action.

Example 2

```
searchBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        Connection conn = null;
        try {
            String userName = "root";
            String password = "password";
            String url = URL//localhost/Casino";
            conn = DriverManager.getConnection(url, userName, password);
            System.out.println("\nDatabase Connection Established..");
        } catch (SQLException ex) {
            System.out.println("Error connecting to Database");
        }
    }
});
```

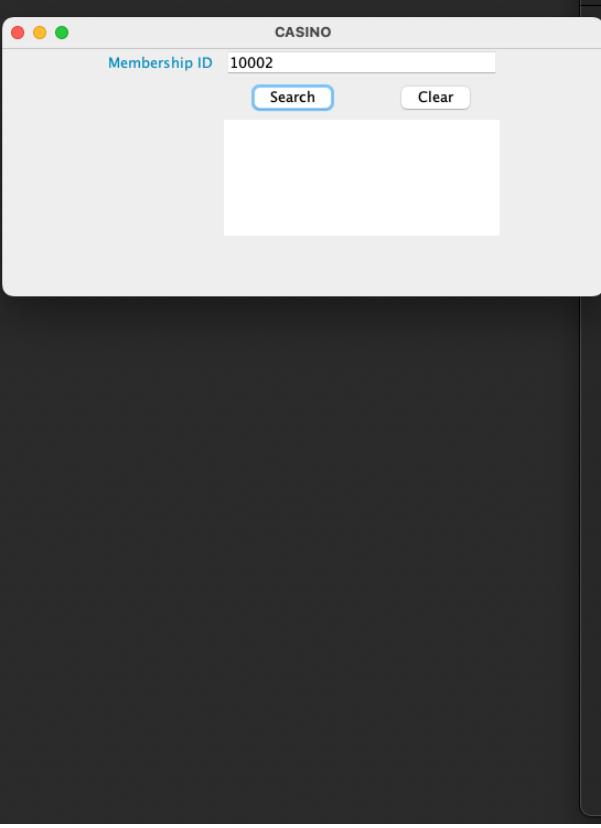
Here we can see the action performed by our search button(searchBtn), which connects to our database using our username, password, and URL (Uniform Resource Locator), all from our MySQL Workbench.

Example 3

```
clearBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        textField1.setText("");
        textPanel1.setText("");
    }
});
```

In this snippet, we create the clear button(clearBtn) that will set our textField1(Member ID number) and textPanel1(customer information) to blank. After the action is performed, we can type a different Membership ID.

If a wrong Member Id would be entered, nothing will be displayed. In Figure 3.2, we show how a wrong Member ID does not display any customer information. If a valid Member ID is entered, the customer's ID, first name, last name, email, and phone number will be displayed in our textPanel1, as shown in figure 3.3.



CustomerID	CustomerFirstName	CustomerLastName	CustomerEmail	CustomerPhoneNumber	MembershipID
2000	Terence	Buck	Buck_Terence@gmail.com	4699422491	10002
2001	Arthur	Sanders	Sanders_Arthur@gmail.com	6258267848	NULL
2002	Ananya	Atherton	Atherton_Ananya@gmail.com	4987673488	100025
2003	Kush	Leach	Leach_Kush@gmail.com	3869870262	100001
2004	Elina	Reyna	Reyna_Elina@gmail.com	7293038778	NULL
2005	Ellis	Shaffer	Shaffer_Ellis@gmail.com	3034521945	100002
2006	Kalem	Redfern	Redfern_Kalem@gmail.com	4587900824	100003
2007	Trixie	Schmidt	Schmidt_Trixie@gmail.com	5024954070	NULL
2008	Sanna	Leigh	Leigh_Sanna@gmail.com	4854609256	100004
2009	Aliesha	Gaines	Gaines_Aliesha@gmail.com	3717613424	NULL
2010	Lacey-May	Burke	Burke_Lacey-May@gmail.com	6755957397	100005
2011	Jareth	Hurley	Hurley_Jareth@gmail.com	8047975014	NULL
2012	Maximus	Corrigan	Corrigan_Maximus@gmail.com	4657807612	100026
2013	Braden	Vb	Vb_Braden@gmail.com	9295939686	NULL
2014	TJ	Cruz	Cruz_TJ@gmail.com	6995821038	100006
2015	Safya	Combs	Combs_Safya@gmail.com	2378137893	NULL
2016	Ameerah	Betts	Betts_Ameerah@gmail.com	6839009709	NULL
2017	Jeremiah	Paterson	Paterson_Jeremiah@gmail.com	9645788496	100007
2018	Yasmine	Richardson	Richardson_Yasmine@gmail.com	4134517906	NULL
2019	Ewan	Byrd	Byrd_Ewan@gmail.com	2999816264	100027
2020	Nojus	Childs	Childs_Nojus@gmail.com	4697537776	NULL
2021	Eben	Ware	Ware_Eben@gmail.com	5574081486	100008
2022	Ronaldo	England	England_Ronaldo@gmail.com	8802490502	NULL
2023	Shola	Briggs	Briggs_Shola@gmail.com	8648052903	100009
2024	Irving	Jacobson	Jacobson_Irving@gmail.com	4435913630	100010
2025	Arian	Gill	Gill_Arian@gmail.com	2735100870	100011
2026	Keith	Ramsey	Ramsey_Keith@gmail.com	2565198560	100012
2027	Daisy	Lucas	Lucas_Daisy@gmail.com	8136675282	NULL
2028	Shelley	Potts	Potts_Shelley@gmail.com	5266264900	NULL
2029	Sultan	Pham	Pham_Sultan@gmail.com	5076382007	NULL
2030	Stefania	Couch	Couch_Stefania@gmail.com	5063178352	NULL
2031	Gia	Watts	Watts_Gia@gmail.com	3897881107	100013
2032	Laibah	Fields	Fields_Laibah@gmail.com	9158952637	100014
2033	Walter	Schaefer	Schaefer_Walter@gmail.com	9534758800	100015
2034	Beatrix	Hodgson	Hodgson_Beatrix@gmail.com	9668780503	NULL
2035	Leonidas	Richards	Richards_Leonidas@gmail.com	6035498635	100016
2036	Amelie	Pugh	Pugh_Amelie@gmail.com	6598597652	NULL

Figure 3.2: Here, we can see our GUI with a search button pressed but not displaying any customer information. This is because we entered an invalid Member ID.

CustomerID	CustomerFirstName	CustomerLastName	CustomerEmail	CustomerPhoneNumber	MembershipID
2000	Terence	Buck	Buck_Terence@gmail.com	4699422491	10002
2001	Arthur	Sanders	Sanders_Arthur@gmail.com	6258267848	NULL
2002	Ananya	Atherton	Atherton_Ananya@gmail.com	4987673488	100025
2003	Kush	Leach	Leach_Kush@gmail.com	3869870262	100001
2004	Elina	Reyna	Reyna_Elina@gmail.com	7293038778	NULL
2005	Ellis	Shaffer	Shaffer_Ellis@gmail.com	3034521945	100002
2006	Kalem	Redfern	Redfern_Kalem@gmail.com	4587900824	100003
2007	Trixie	Schmidt	Schmidt_Trixie@gmail.com	5024954070	NULL
2008	Sanna	Leigh	Leigh_Sanna@gmail.com	4854609256	100004
2009	Aliesha	Gaines	Gaines_Aliesha@gmail.com	3717613424	NULL
2010	Lacey-May	Burke	Burke_Lacey-May@gmail.com	6755957397	100005
2011	Jareth	Hurley	Hurley_Jareth@gmail.com	8047975014	NULL
2012	Maximus	Corrigan	Corrigan_Maximus@gmail.com	4657807612	100026
2013	Braden	Vo	Vo_Braden@gmail.com	9295939686	NULL
2014	TJ	Cruz	Cruz_TJ@gmail.com	6995821038	100006
2015	Safya	Combs	Combs_Safya@gmail.com	2378137893	NULL
2016	Ameerah	Betts	Betts_Ameerah@gmail.com	6839009709	NULL
2017	Jeremiah	Paterson	Paterson_Jeremiah@gmail.com	9645788496	100007
2018	Yasmine	Richardson	Richardson_Yasmine@gmail.com	4134517906	NULL
2019	Ewan	Byrd	Byrd_Ewan@gmail.com	2999816264	100027
2020	Nojus	Childs	Childs_Nojus@gmail.com	4697537776	NULL
2021	Eben	Ware	Ware_Eben@gmail.com	5574081486	100008
2022	Ronaldo	England	England_Ronaldo@gmail.com	8802490502	NULL
2023	Shola	Briggs	Briggs_Shola@gmail.com	8648052903	100009
2024	Irving	Jacobson	Jacobson_Irving@gmail.com	4435913630	100010
2025	Arian	Gill	Gill_Arian@gmail.com	2735100870	100011
2026	Keith	Ramsey	Ramsey_Keith@gmail.com	2565198560	100012
2027	Daisy	Lucas	Lucas_Daisy@gmail.com	8136675282	NULL
2028	Shelley	Potts	Potts_Shelley@gmail.com	5266264900	NULL
2029	Sultan	Pham	Pham_Sultan@gmail.com	5076382007	NULL
2030	Stefania	Couch	Couch_Stefania@gmail.com	5063178352	NULL
2031	Gia	Watts	Watts_Gia@gmail.com	3897881107	100013
2032	Laibah	Fields	Fields_Laibah@gmail.com	9158952637	100014
2033	Walter	Schaefer	Schaefer_Walter@gmail.com	9534758800	100015
2034	Beatrix	Hodgson	Hodgson_Beatrix@gmail.com	9668780503	NULL
2035	Leonidas	Richards	Richards_Leonidas@gmail.com	6035498635	100016
2036	Amelie	Pugh	Pugh_Amelie@gmail.com	6598597652	NULL

Figure 3.3: Displaying customer information after entering the correct Membership ID.

Example 4

```

Statement stmt = conn.createStatement();
String query = "select * from customer where MembershipID = ? ";
PreparedStatement pst = conn.prepareStatement(query);
pst.setString(1, textField1.getText());
ResultSet rs = pst.executeQuery();

while (rs.next())
    textPanel1.setText(
        "Customer ID:      " + rs.getInt(1)
        + "\n" + "First Name:      " + rs.getString(2)
        + "\n" + "Last Name:       " + rs.getString(3)
        + "\n" + "Email:           " + rs.getString(4)
        + "\n" + "Phone Number:   " + rs.getString(5));

conn.close();
    
```

In Example 4, we run the following MySQL query “select *(all) from customer (customer table) where (what column) MembershipID (column name) =? (Whatever matching Membership ID we type). Then we display the customer's information as shown in figure 3.3.

MySQL Queries

After our Java Swing program and our Casino database tables were completed we had to run MySQL queries.

Slot Machine Maximum Payout

The screenshot shows the MySQL Workbench interface. On the left, the schema browser displays the 'Casino' database with its tables: cardtables, customer, employee, membership, and slotmachines. The 'slotmachines' table is selected. On the right, the SQL editor contains the following query:

```
1  SELECT MAX(payout) as max_payout
2  FROM slotmachines;
```

The results grid shows a single row with the column 'max_payout' containing the value '4936.36'. The status bar at the bottom indicates a zoom level of 100% and a duration of 19:2.

- Displays the highest Maximum payout in the slot machines.

Slot Machine payout less than \$1000

The screenshot shows a database interface with a sidebar on the left and a main query and results area on the right.

Left Sidebar:

- Schema: Casino
- Tables:
 - cardtables
 - customer
 - employee
 - membership
 - slotmachines
- Columns:
 - SMNumber
 - CustomerID
 - AmountCust...
 - Payout
 - BetAmount

Object Info and **Session** buttons are visible at the bottom of the sidebar.

Schema: Casino is highlighted in green.

Main Area:

```
1   SELECT Payout
2   FROM slotmachines WHERE Payout < "1000.00";
```

The results grid shows the following data:

Payout
146.96
304.31
863.47
44.44
413.26
515.54
605.25
772.05
864.42
875.29

- Displays payouts that were less than \$1000.

Highest bet placed in slot machine

The screenshot shows a database interface with a sidebar and a main query editor.

Sidebar:

- Schema: Casino
- Tables:
 - cardtables
 - customer
 - employee
 - membership
 - slotmachines
 - Columns:
 - SMNumber
 - CustomerID
 - AmountCust...
 - Payout

Query Editor:

```
1 *   SELECT MAX(betAmount)
2      as Highest_Place_Bet
3   FROM slotmachines;
```

Result Grid:

Highest_Place_B...	
▶	50

100% 1:2

Result Grid Filter Rows: Search Export:

Object Info Session Schema: Casino

- Highest bet placed was \$50

Number of times slot machine was played

The screenshot shows a database interface with a sidebar on the left containing a tree view of schema objects under the 'Casino' schema. The 'slotmachines' table is selected. The main area displays a SQL query and its execution results.

```
1 •   SELECT CustomerID, AmountCustomersPlayed, SMNumber
2     FROM slotmachines
3   ORDER by AmountCustomersPlayed DESC;
```

The Result Grid shows the following data:

	CustomerID	AmountCustomersPlay...	SMNumber
▶	2033	337	34
	2027	331	28
	2041	329	42
	2021	326	22
	2025	323	26
	2012	323	13
	2007	322	8
	2014	322	15
	2016	305	17
	2042	305	43
	2031	278	32
	2045	274	46
	2015	269	16
	2032	269	33
	2006	263	7
	2004	262	5
	2040	260	41
	2038	255	39
	2011	247	12
	2024	227	25
	2020	211	21
	2017	203	18
	2013	196	14
	2002	194	3
	2043	185	44
	2026	183	27

- Displays the customer and the amount of times they played a Slot machine.

Customers with Membership

The screenshot shows a database interface with a sidebar on the left containing a tree view of tables under the schema 'Casino'. The 'membership' table is selected. Below the sidebar is a code editor with a SQL query:

```
1 •  SELECT membership.MembershipID, CustomerFirstName, CustomerLastName  
2   FROM customer Join membership  
3     USING(MembershipID);
```

Below the code editor is a result grid titled 'Result Grid' showing the output of the query:

	MembershipID	CustomerFirstNa...	CustomerLastNa...
▶	100002	Terence	Buck
	100025	Ananya	Atherton
	100001	Kush	Leach
	100002	Ellis	Shaffer
	100003	Kalem	Redfern
	100004	Sanna	Leigh
	100005	Lacey-May	Burke
	100026	Maximus	Corrigan
	100006	Tj	Cruz
	100007	Jeremiah	Paterson
	100027	Ewan	Byrd
	100008	Eben	Ware
	100009	Shola	Briggs
	100010	Irving	Jacobson
	100011	Arian	Gill
	100012	Keith	Ramsey
	100013	Gia	Watts
	100014	Laibah	Fields
	100015	Walter	Schaefer
	100016	Leonidas	Richards
	100017	Juniper	Pruitt

- List of customers who are members.

Average payout of games

The screenshot shows a database interface with a sidebar on the left containing a tree view of tables under the schema 'Casino'. The 'slotmachines' table is selected. Below the sidebar is a code editor with a SQL query:

```
1 •  SELECT (SUM(S.Payout) + SUM(C.CTPayout))/ (COUNT(S.Payout) + COUNT(C.CTPay...  
2   FROM SlotMachines S, CardTables C;
```

Below the code editor is a result grid titled 'Result Grid' showing the output of the query:

	(SUM(S.Payout) + SUM(C.CTPayout))/ (COU...
▶	2130.0954000000065

- Average payout of games is \$2130.10.

Average payout of card tables

The screenshot shows a database interface with a sidebar containing the schema structure. The 'Tables' section includes 'cardtables', 'customer', 'employee', 'membership', and 'slotmachines'. Under 'slotmachines', there is a 'Columns' section with 'SMNumber', 'CustomerID', 'AmountCust...', and 'Payout'. The main area displays a SQL query and its results.

```
1 • SELECT AVG(CTPayout)
2   AS AVGCTPayoutCardTables
3   FROM CardTables;
```

Result Grid | Filter Rows: Search Export:

AVGCTPayoutCardTables		
	1714.88	
▶	1714.88	

- Card Tables have an average payout of \$1714.88.

Employee working at the card table

The screenshot shows a database interface with a sidebar containing the schema structure. The 'Tables' section includes 'cardtables', 'customer', 'employee', 'membership', and 'slotmachines'. Under 'slotmachines', there is a 'Columns' section with 'SMNumber', 'CustomerID', 'AmountCust...', and 'Payout'. The main area displays a SQL query and its results.

```
1 • SELECT EmployeeID, EmployeeFirstName, EmployeeLastName
2   FROM employee
3   WHERE EmployeeID IN
4   (SELECT DISTINCT EmployeeID
5   FROM cardtables);
```

Result Grid | Filter Rows: Search Export:

EmployeeID	EmployeeFirstName	EmployeeLastName
1000	Jack	Hammer
1001	Matt	Tress
1002	Weston	East
1003	Will	Power
1004	Chris P	Bacon
1005	Doug	Hole
1006	Joy	Rider
1007	Jasmine	Rice
1008	Marshaw	Mellow
1009	Harry	Beard

Customers Playing Slot Machine over 200 times

The screenshot shows a database interface with a sidebar on the left containing a tree view of database objects under the 'Casino' schema. The 'slotmachines' table is selected, and its columns (CustomerID, SMNumber, CustomerID, AmountCust..., Payout) are listed. A main panel displays a SQL query and its results.

```
1 •  SELECT slotmachines.CustomerID, customer.CustomerFirstName,
2      customer.CustomerLastName, slotmachines.AmountCustomersPlayed
3  FROM customer
4  JOIN slotmachines
5    ON AmountCustomersPlayed > 200 AND customer.CustomerID = slotmachines.CustomerID;
```

The results grid shows the following data:

CustomerID	CustomerFirstName	CustomerLastName	AmountCustomersPlay...
2004	Elina	Reyna	262
2006	Kalem	Redfern	263
2007	Trixie	Schmidt	322
2011	Jareth	Hurley	247
2012	Maximus	Corrigan	323
2014	Tj	Cruz	322
2015	Safiya	Combs	269
2016	Ameerah	Betts	305
2017	Jeremiah	Paterson	203
2020	Nojus	Childs	211
2021	Eben	Ware	326
2024	Irving	Jacobson	227
2025	Arian	Gill	323
2027	Daisy	Lucas	331
2031	Gia	Watts	278
2032	Laibah	Fields	269
2033	Walter	Schaefer	337
2038	Juniper	Pruitt	255
2040	Azeem	Portillo	260
2041	Sullivan	Schmitt	329
2042	Ross	Solis	305
2045	Kassar	Meadows	371

Played on Card Tables and Slot Machines

The screenshot shows the Oracle SQL Developer interface. On the left, the object browser displays a schema named 'Casino' containing tables like 'cardtables', 'customer', 'employee', 'membership', and 'slotmachines', along with their columns. The main pane shows a multi-level nested query:

```
1 *  SELECT CustomerID, CustomerFirstName, CustomerLastName
2   FROM customer
3   WHERE CustomerID IN
4     (SELECT CustomerID
5      FROM cardtables
6      WHERE CustomerID IN
7        (SELECT CustomerID
8          FROM slotmachines WHERE MembershipID IN
9            (SELECT MembershipID
10         FROM membership
11        WHERE MembershipID IS NOT NULL)));

```

The result grid below displays 15 rows of data from the query:

	CustomerID	CustomerFirstName	CustomerLastName
▶	2000	Terence	Buck
	2003	Kush	Leach
	2005	Ellis	Shaffer
	2008	Sanna	Leigh
	2012	Maximus	Corrigan
	2021	Eben	Ware
	2025	Arian	Gill
	2032	Laibah	Fields
	2033	Walter	Schaefer
	2040	Azeem	Portillo
	2041	Sullivan	Schmitt
	2044	Sheldon	Benson
	2045	Kacper	Meadows
	2048	Danica	Silva

Credits

Thanks to my group members Ezra Benitez, Berenis Castruita, Ramon Mata, Naomi Nguyen.