Chrystal Mingo

CSC 22100

Professor Auda

Final Project

**The Main.java:**

In main is where I created my table, buttons, and text fields using JavaFX.

**CODE:**

```
package sample;

import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import javafx.application.Application;
import javafx.event.EventHandler;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Border;
import javafx.scene.layout.VBox;
import javafx.scene.layout.HBox;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import static javafx.application.Application.launch;

import javafx.scene.layout.BorderPane;

public class Main extends Application {
  public static Connect connection;

  private TableView table = new TableView();

  @Override
  public void start(Stage stage) throws Exception{
      BorderPane root = new BorderPane();
      Scene scene = new Scene(new Group());
      stage.setTitle("Chrystal Mingo Final Project");
      stage.setWidth(700);
      stage.setHeight(600);
```

```
        final Label label = new Label("Employee Database");
        label.setFont(new Font("Times New Roman", 20));
        //When clicked the button will show all of the data
        final Button showAllButton = new Button();
        showAllButton.setMaxWidth(250);
        showAllButton.setText("Show All Data");

        HBox firstRow = new HBox();
        firstRow.setSpacing(428);
        firstRow.getChildren().addAll(label,showAllButton);

        table.setEditable(true);
        table.setMaxHeight(300);

        //to set Employee info into table
        TableColumn SSNCol = new TableColumn("SSN");
        SSNCol.setCellValueFactory(new PropertyValueFactory<>("SSN"));

        TableColumn firstNameCol = new TableColumn("First Name");
        firstNameCol.setCellValueFactory(new PropertyValueFactory<>("firstName"));

        TableColumn lastNameCol = new TableColumn("Last Name");
        lastNameCol.setCellValueFactory(new PropertyValueFactory<>("lastName"));

        TableColumn birthdayCol = new TableColumn("Birthday");
        birthdayCol.setCellValueFactory(new PropertyValueFactory<>("birthday"));

        TableColumn employeeTypeCol = new TableColumn("Employee Type");
        employeeTypeCol.setCellValueFactory(new PropertyValueFactory<>("employeeType"));

        TableColumn deptCol = new TableColumn("Department Name");
        deptCol.setCellValueFactory(new PropertyValueFactory<>("departmentName"));


        deptCol.setMinWidth(30);
        table.getColumns().addAll( firstNameCol, lastNameCol, birthdayCol, employeeTypeCol, deptCol,
    SSNCol);
        List<Employee> employees;
```

```java
//search textfield
final TextField textbar = new TextField("First name");
textbar.setMaxWidth(340);

//remove by SSN textField
final TextField removebySSNField = new TextField("Last Name or SSN");
removebySSNField.setMaxWidth(340);
removebySSNField.setAlignment(Pos.BOTTOM_RIGHT);

//remove by SSN button
final Button removebySSNButton = new Button();
removebySSNButton.setMaxWidth(500);
removebySSNButton.setText("Remove");

final Button findByFirstButton = new Button();
findByFirstButton.setMaxWidth(150);
findByFirstButton.setText("Find By First name");
final Button findByLastButton = new Button();
findByLastButton.setMaxWidth(150);
findByLastButton.setText("Find By Last Name");

final Button increaseBy10PercentButton = new Button();
increaseBy10PercentButton.setMaxWidth(150);
increaseBy10PercentButton.setText("10 percent");

//Add employee textFields
//to add first name
final TextField fnameField = new TextField("first name");
fnameField.setMaxWidth(100);
//to add last name
final TextField lnameField = new TextField("last name");
lnameField.setMaxWidth(100);
//to add birthday
final TextField bdayField = new TextField("birthday");
bdayField.setMaxWidth(100);
//to add employeeType
final TextField employeeTypeField = new TextField("employeeType");
employeeTypeField.setMaxWidth(100);
```

```
    // to add department name
    final TextField deptNameField = new TextField("department name");
    deptNameField.setMaxWidth(100);
    //to add SSN
    final TextField SSNField = new TextField("SSN");
    SSNField.setMaxWidth(100);
    //Button to add employees
    final Button addEmployeeButton = new Button();
    addEmployeeButton.setMaxWidth(710);
    addEmployeeButton.setText("Add Employee");

    //Query TextField
    final TextField queryTextField = new TextField("Query Script");
    queryTextField.setMaxWidth(710);

    final Button submitQueryButton = new Button();
    submitQueryButton.setMaxWidth(710);
    submitQueryButton.setText("Submit Query");

    ComboBox searchCombo = new ComboBox();
    searchCombo.getItems().addAll("First Name", "Last Name", "SSN");

    HBox belowTable = new HBox();
    belowTable.setAlignment(Pos.BOTTOM_LEFT);
    belowTable.setSpacing(50);
    belowTable.getChildren().addAll(textbar, removebySSNField, removebySSNButton);

    HBox searchHBox = new HBox();
    searchHBox.setAlignment(Pos.BOTTOM_LEFT);
    searchHBox.setSpacing(70);
    searchHBox.getChildren().addAll(findByFirstButton,findByLastButton, increaseBy10PercentButton);

    HBox addEmployeeHBox = new HBox();
    addEmployeeHBox.setAlignment(Pos.BOTTOM_CENTER);
    addEmployeeHBox.setSpacing(25);
    addEmployeeHBox.getChildren().addAll(fnameField, lnameField, bdayField, employeeTypeField,
deptNameField, SSNField);

    try{
```

```
    connection = new Connect();
    employees = connection.getAllEmployees();

    employees = connection.getAllEmployees();
    for(int i = 0; i<employees.size(); i++){
        System.out.println(employees.get(i).toString());
    }

}catch (SQLException e){
    e.printStackTrace();
    employees = new ArrayList<Employee>();
}
for(int i =0; i<employees.size(); i++){
    table.getItems().add(employees.get(i));
}
final VBox vbox = new VBox();
vbox.setSpacing(15);
vbox.prefWidthProperty().bind(stage.widthProperty().multiply(.95));
vbox.setPadding(new Insets(15, 0, 0, 20));
vbox.getChildren().addAll(firstRow, table, belowTable,
        searchHBox, addEmployeeHBox, addEmployeeButton, queryTextField, submitQueryButton);


//GUI makes it interactive and here is where when a button is clicked it does a certain functionality
removebySSNButton.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {
        connection.deleteEmployee(removebySSNField.getText());

        table.getItems().clear();
        List<Employee> employees1 = connection.getAllEmployees();
        for(int i =0; i<employees1.size(); i++){
            table.getItems().add(employees1.get(i));
        }
    }
});

submitQueryButton.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {
        List<Employee> employees1 =  connection.runQuery(queryTextField.getText());

        table.getItems().clear();
        for(int i =0; i<employees1.size(); i++){
            table.getItems().add(employees1.get(i));
        }
    }
});

addEmployeeButton.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {
        connection.insertEmployee(fnameField.getText(),lnameField.getText(),bdayField.getText(),
                employeeTypeField.getText(), deptNameField.getText(), SSNField.getText());

        table.getItems().clear();
        List<Employee> employees1 = connection.getAllEmployees();
        for(int i =0; i<employees1.size(); i++){
            table.getItems().add(employees1.get(i));
        }
    }
});

findByFirstButton.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {
        List<Employee> employees1 = connection.getEmployeeByFName(textbar.getText());

        table.getItems().clear();
        for(int i =0; i<employees1.size(); i++){
            table.getItems().add(employees1.get(i));
        }
    }
});

findByLastButton.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent mouseEvent) {
            List<Employee> employees1 = connection.getEmployeeByLName(textbar.getText());

            table.getItems().clear();
            for(int i =0; i<employees1.size(); i++){
                table.getItems().add(employees1.get(i));
            }
        }
    });

    showAllButton.setOnMouseClicked(new EventHandler<MouseEvent>() {
        @Override
        public void handle(MouseEvent mouseEvent) {
            List<Employee> employees1 = connection.getAllEmployees();

            table.getItems().clear();
            for(int i =0; i<employees1.size(); i++){
                table.getItems().add(employees1.get(i));
            }
        }
    });

    increaseBy10PercentButton.setOnMouseClicked(new EventHandler<MouseEvent>() {
        @Override
        public void handle(MouseEvent mouseEvent) {
            connection.increaseCommissionBy10Percent();

            table.getItems().clear();
            List<Employee> employees1 = connection.getAllEmployees();
            for(int i =0; i<employees1.size(); i++){
                table.getItems().add(employees1.get(i));
            }
        }
    });
    ((Group) scene.getRoot()).getChildren().addAll(vbox);

    stage.setScene(scene);
    stage.show();
}
```

```
  public static void main(String[] args) {

      launch(args);
  }
}
```

**The Connect.java:**

In this program I connected to my employee.db using SQLite.

**CODE**:

```
package sample;

import java.sql.*;
import java.util.*;

public class Connect {
  private static final String URL =
"jdbc:sqlite:/Users/chrystalmingo3/Desktop/FinalProject_JavaFx/sqlite/employees.db";
  private static final String USERNAME = "deitel";
  private static final String PASSWORD = "deitel";

  // manages connection
  private Connection connection;
  private PreparedStatement selectAllEmpl;
  private PreparedStatement insertEmpl;
  private PreparedStatement selectByFirstname;
  private PreparedStatement selectByLastname;
  private PreparedStatement selectBy30Hours;
  private PreparedStatement increaseBy10Percent;
  private PreparedStatement deleteEmpl;
  private PreparedStatement deleteEmpl1;
  private PreparedStatement deleteEmpl2;
  private PreparedStatement deleteEmpl3;
  private PreparedStatement deleteEmpl4;

  public Connect() throws SQLException {
      try {

          connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
```

```
            //When we want information from a database we create a query
            //Requesting all records and all fields from the employee.db
            //Select* -> choose Field(s) to display
            //From employee -> chose Table(s) containing fields
            selectAllEmpl = connection.prepareStatement("SELECT * FROM employees");
            //How to add Employee info
            //Use INSERT INTO
            insertEmpl = connection.prepareStatement("INSERT INTO employees (firstname, " + "
lastname,birthday,employeetype,departmentname, socialsecuritynumber) values (?, ?, ?,?,?,?)");
            //Checking for employee by first name
            selectByFirstname = connection.prepareStatement("SELECT * FROM employees where firstname = ?");
            //Checking for employee by first name
            selectByLastname = connection.prepareStatement("SELECT * FROM employees where lastname = ?");
            //Checking for employee by first name, last name, SSN, hours,
            selectBy30Hours = connection.prepareStatement("select employees.firstname, employees.lastname," +
                " hourlyemployees.socialsecuritynumber, hourlyemployees.hours from" +
                " hourlyemployees inner join employees on" +
                " hourlyemployees.socialsecuritynumber=employees.socialsecuritynumber;");
            //How to find increase of salary by 10%
            increaseBy10Percent = connection.prepareStatement("update basepluscommissionemployees set
basesalary = 1.10*basesalary");
            //How to delete/remove certain employeetypes based off of their SSN
            deleteEmpl = connection.prepareStatement("DELETE FROM BASEPLUSCOMMISSIONEMPLOYEES
WHERE socialSecurityNumber = ?");
            deleteEmpl1 = connection.prepareStatement("DELETE FROM COMMISSIONEMPLOYEES WHERE
socialSecurityNumber = ?");
            deleteEmpl2= connection.prepareStatement("DELETE FROM HOURLYEMPLOYEES WHERE
socialSecurityNumber = ?");
            deleteEmpl3 = connection.prepareStatement("DELETE FROM SALARIEDEMPLOYEES WHERE
socialSecurityNumber = ?");
            deleteEmpl4 = connection.prepareStatement("DELETE FROM EMPLOYEES WHERE
socialSecurityNumber = ?");
        } catch (SQLException exception){
            throw new SQLException(exception);
        }
    }
    //ArrayList stores all the information about the employees

    public List<Employee> getAllEmployees() {
```

```
    List<Employee> results = new ArrayList<>();

    ResultSet resultSet = null;
    try {
        resultSet = selectAllEmpl.executeQuery();
        while(resultSet.next()){
            results.add(new Employee(
                    resultSet.getString("FIRSTNAME"),
                    resultSet.getString("LASTNAME"),
                    resultSet.getString("BIRTHDAY"),
                    resultSet.getString("EMPLOYEETYPE"),
                    resultSet.getString("DEPARTMENTNAME"),
                    resultSet.getString("SOCIALSECURITYNUMBER")

            ));
        }
    }catch (SQLException e){
        System.out.println(e);
    }finally {
        try {
            resultSet.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
    }
    return results;
}
//function to increase CommisionBy10Percent

public int increaseCommissionBy10Percent(){
    int result = 0;
    try{
        result = increaseBy10Percent.executeUpdate();
    }catch (SQLException e){
        e.printStackTrace();
    }
    return result;
}
//Function to find employee based of their Last Name
```

```
public List<Employee> getEmployeeByLastName(String lname){
    List<Employee> results = new ArrayList<>();

    ResultSet resultSet = null;
    try {
        selectByLastname.setString(1,lname);
        resultSet = selectByLastname.executeQuery();
        while(resultSet.next()){
            results.add(new Employee(
                    resultSet.getString("FIRSTNAME"),
                    resultSet.getString("LASTNAME"),
                    resultSet.getString("BIRTHDAY"),
                    resultSet.getString("EMPLOYEETYPE"),
                    resultSet.getString("DEPARTMENTNAME"),
                    resultSet.getString("SOCIALSECURITYNUMBER")

            ));
        }
    }catch (SQLException e){
        System.out.println(e);
    }finally {
        try {
            resultSet.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
    }
    return results;
}
//Function to find employee based of First Name
public List<Employee> getEmployeeByFirstName(String fname){
    List<Employee> results = new ArrayList<>();

    ResultSet resultSet = null;
    try {
        selectByFirstname.setString(1,fname);
        resultSet = selectByFirstname.executeQuery();
        while(resultSet.next()){
            results.add(new Employee(
                    resultSet.getString("FIRSTNAME"),
                    resultSet.getString("LASTNAME"),
                    resultSet.getString("BIRTHDAY"),
                    resultSet.getString("EMPLOYEETYPE"),
                    resultSet.getString("DEPARTMENTNAME"),
                    resultSet.getString("SOCIALSECURITYNUMBER")

            ));
        }
    }catch (SQLException e){
        System.out.println(e);
    }finally {
        try {
            resultSet.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
    }
    return results;
}
//Function to select those that work 30Hours
public List<Employee> selectBy30Hours(){
    List<Employee> results = new ArrayList<>();

    ResultSet resultSet = null;
    try {
        resultSet = selectBy30Hours.executeQuery();
        while(resultSet.next()){
            results.add(new Employee(
                    resultSet.getString("FIRSTNAME"),
                    resultSet.getString("LASTNAME"),
                    resultSet.getString("BIRTHDAY"),
                    resultSet.getString("EMPLOYEETYPE"),
                    resultSet.getString("DEPARTMENTNAME"),
                    resultSet.getString("SOCIALSECURITYNUMBER")

            ));
        }
    }catch (SQLException e){
        System.out.println(e);
    }finally {
        try {
            resultSet.close();
        }catch(SQLException e){
            e.printStackTrace();
        }
    }
    return results;
}
//function to insert an employee into the Database
public int insertEmployee(String fname, String lname, String birthday,
                          String employeetype, String departmentName, String ssn){
    int result = 0;
    try{
        insertEmpl.setString(1,fname);
        insertEmpl.setString(2,lname);
        insertEmpl.setString(3,birthday);
        insertEmpl.setString(4,employeetype);
        insertEmpl.setString(5,departmentName);
        insertEmpl.setString(6,ssn);
        result = insertEmpl.executeUpdate();
    }catch (SQLException e){
        e.printStackTrace();
    }
    return result;
}
//function to remove employee based off of SSN
public int deleteEmployee(String ssn){
    int result = 0;
    try{
        deleteEmpl.setString(1,ssn);
        deleteEmpl.executeUpdate();
        deleteEmpl1.setString(1,ssn);
        deleteEmpl1.executeUpdate();
        deleteEmpl2.setString(1,ssn);
        deleteEmpl2.executeUpdate();
        deleteEmpl3.setString(1,ssn);
```

```
            deleteEmpl3.executeUpdate();
            deleteEmpl4.setString(1,ssn);
            deleteEmpl4.executeUpdate();

        }catch (SQLException e){
            e.printStackTrace();
        }
        return result;
    }
    //Function to run SQLite based off of SQL Query scripts
    public List<Employee> runQuery(String input){

        List<Employee> results = new ArrayList<>();
        ResultSet resultSet;

        try{
            PreparedStatement statement = connection.prepareStatement(input);
            resultSet = statement.executeQuery();
            System.out.println(resultSet.toString());
            while(resultSet.next()) {
                System.out.println(resultSet.getString("FIRSTNAME"));
                results.add(new Employee(
                        resultSet.getString("FIRSTNAME"),
                        resultSet.getString("LASTNAME"),
                        resultSet.getString("BIRTHDAY"),
                        resultSet.getString("EMPLOYEETYPE"),
                        resultSet.getString("DEPARTMENTNAME"),
                        resultSet.getString("SOCIALSECURITYNUMBER")
                ));

            }
        }catch(SQLException e){
            e.printStackTrace();
        }

        return results;
    }
}
```

**The Employee.java:**

In this program gathers the information of the employee

**CODE**:

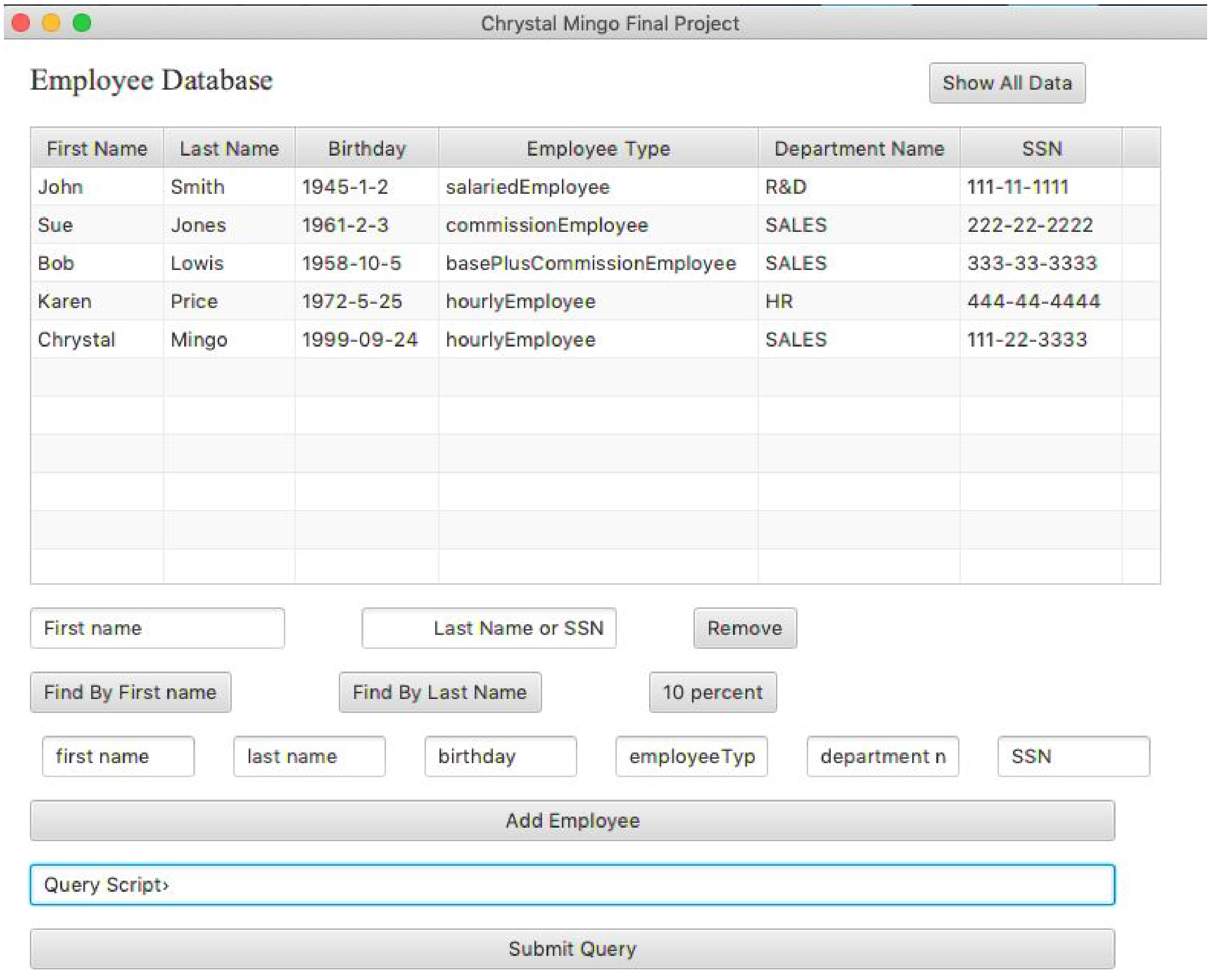
```
package sample;

public class Employee {

  private String firstName;
  private String lastName;
  private String birthday;
  private String employeeType;
  private String departmentName;
  private String SSN;
//Setters
  public Employee(String firstName, String lastName,
                String birthday, String employeeType,
                String departmentName, String SSN ){
    this.firstName = firstName;
    this.lastName = lastName;
    this.birthday = birthday;
    this.employeeType = employeeType;
    this.departmentName = departmentName;
    this.SSN = SSN;
  }
//Getters
  public Employee(){

  }
  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 getBirthday(){
        return birthday;
    }
    public void setBirthday(String birthday){
        this.birthday = birthday;
    }
    public String getEmployeeType(){
        return employeeType;
    }
    public void setEmployeeType(String employeeType){
        this.employeeType = employeeType;
    }
    public String getDepartmentName(){
        return departmentName;
    }
    public void setDepartmentName(String departmentName){
        this.departmentName = departmentName;
    }
    public String getSSN(){
        return SSN;
    }
    public void setSSN(String SSN){
        this.SSN = SSN;
    }
    //Returns Info of each Employee
    public String toString(){
        return "[First name: " + getFirstName() + "] [Last Name: " + getLastName() + "] " +
                "[Birthday: " + getBirthday() + "] [Employee Type: " + getEmployeeType() + "] [Department Name:" +
                getDepartmentName() + "] [SSN: " + getSSN() + "]";
    }

}
```

**Output**:

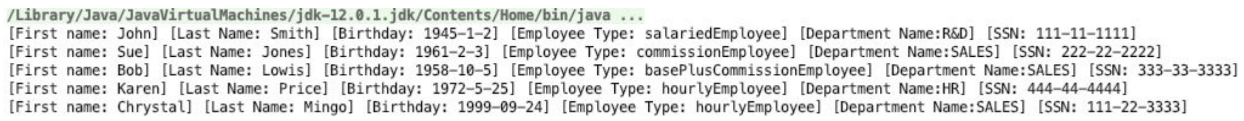
You can enter a name or last name and find it in the employee database.

You can remove an employee by entering the SSN and clicking remove

You can add employee and fill in with their information such as first and last name, birthday, employee type, department name and SSN.



| First Name | Last Name | Birthday | Employee Type | Department Name | SSN |
|---|---|---|---|---|---|
| John | Smith | 1945-1-2 | salariedEmployee | R&D | 111-11-1111 |
| Sue | Jones | 1961-2-3 | commissionEmployee | SALES | 222-22-2222 |
| Bob | Lowis | 1958-10-5 | basePlusCommissionEmployee | SALES | 333-33-3333 |
| Karen | Price | 1972-5-25 | hourlyEmployee | HR | 444-44-4444 |
| Chrystal | Mingo | 1999-09-24 | hourlyEmployee | SALES | 111-22-3333 |

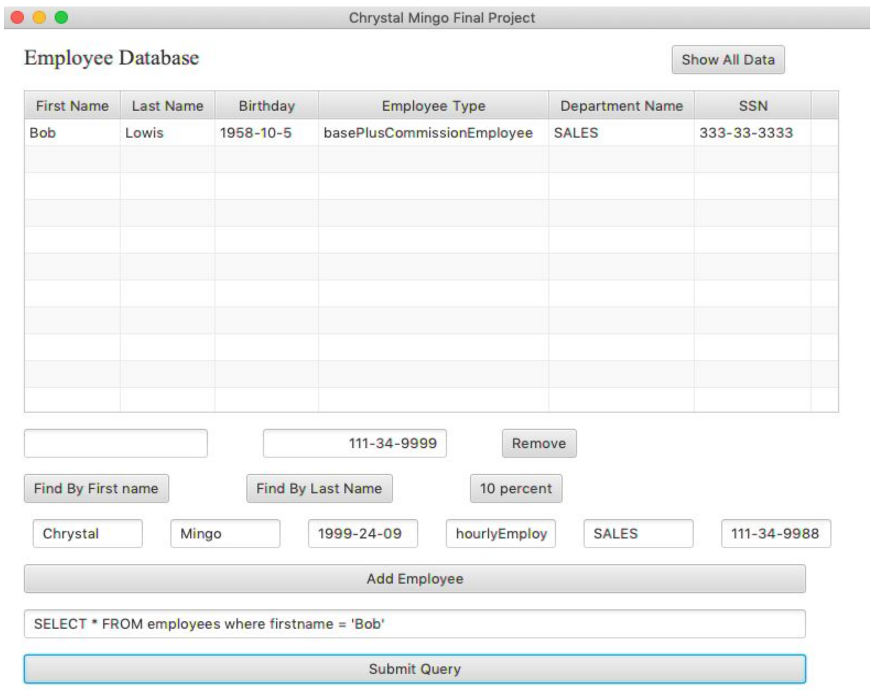


```
/Library/Java/JavaVirtualMachines/jdk-12.0.1.jdk/Contents/Home/bin/java ...
[First name: John] [Last Name: Smith] [Birthday: 1945-1-2] [Employee Type: salariedEmployee] [Department Name:R&D] [SSN: 111-11-1111]
[First name: Sue] [Last Name: Jones] [Birthday: 1961-2-3] [Employee Type: commissionEmployee] [Department Name:SALES] [SSN: 222-22-2222]
[First name: Bob] [Last Name: Lowis] [Birthday: 1958-10-5] [Employee Type: basePlusCommissionEmployee] [Department Name:SALES] [SSN: 333-33-3333]
[First name: Karen] [Last Name: Price] [Birthday: 1972-5-25] [Employee Type: hourlyEmployee] [Department Name:HR] [SSN: 444-44-4444]
[First name: Chrystal] [Last Name: Mingo] [Birthday: 1999-09-24] [Employee Type: hourlyEmployee] [Department Name:SALES] [SSN: 111-22-3333]
```

SQL Script:

EXAMPLE WE DID IN DEMO:

SELECT * FROM employees where firstname = 'Bob'



| First Name | Last Name | Birthday | Employee Type | Department Name | SSN |
|---|---|---|---|---|---|
| Bob | Lowis | 1958-10-5 | basePlusCommissionEmployee | SALES | 333-33-3333 |

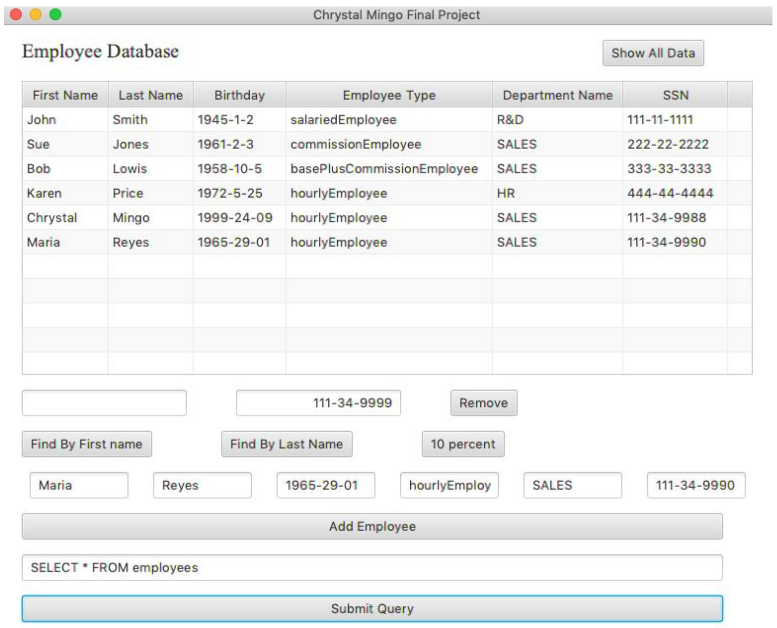
111-34-9999

Chrystal | Mingo | 1999-24-09 | hourlyEmploy | SALES | 111-34-9988

SELECT * FROM employees where firstname = 'Bob'

SQL Script:

SELECT * FROM employees



| First Name | Last Name | Birthday | Employee Type | Department Name | SSN |
|---|---|---|---|---|---|
| John | Smith | 1945-1-2 | salariedEmployee | R&D | 111-11-1111 |
| Sue | Jones | 1961-2-3 | commissionEmployee | SALES | 222-22-2222 |
| Bob | Lowis | 1958-10-5 | basePlusCommissionEmployee | SALES | 333-33-3333 |
| Karen | Price | 1972-5-25 | hourlyEmployee | HR | 444-44-4444 |
| Chrystal | Mingo | 1999-24-09 | hourlyEmployee | SALES | 111-34-9988 |
| Maria | Reyes | 1965-29-01 | hourlyEmployee | SALES | 111-34-9990 |

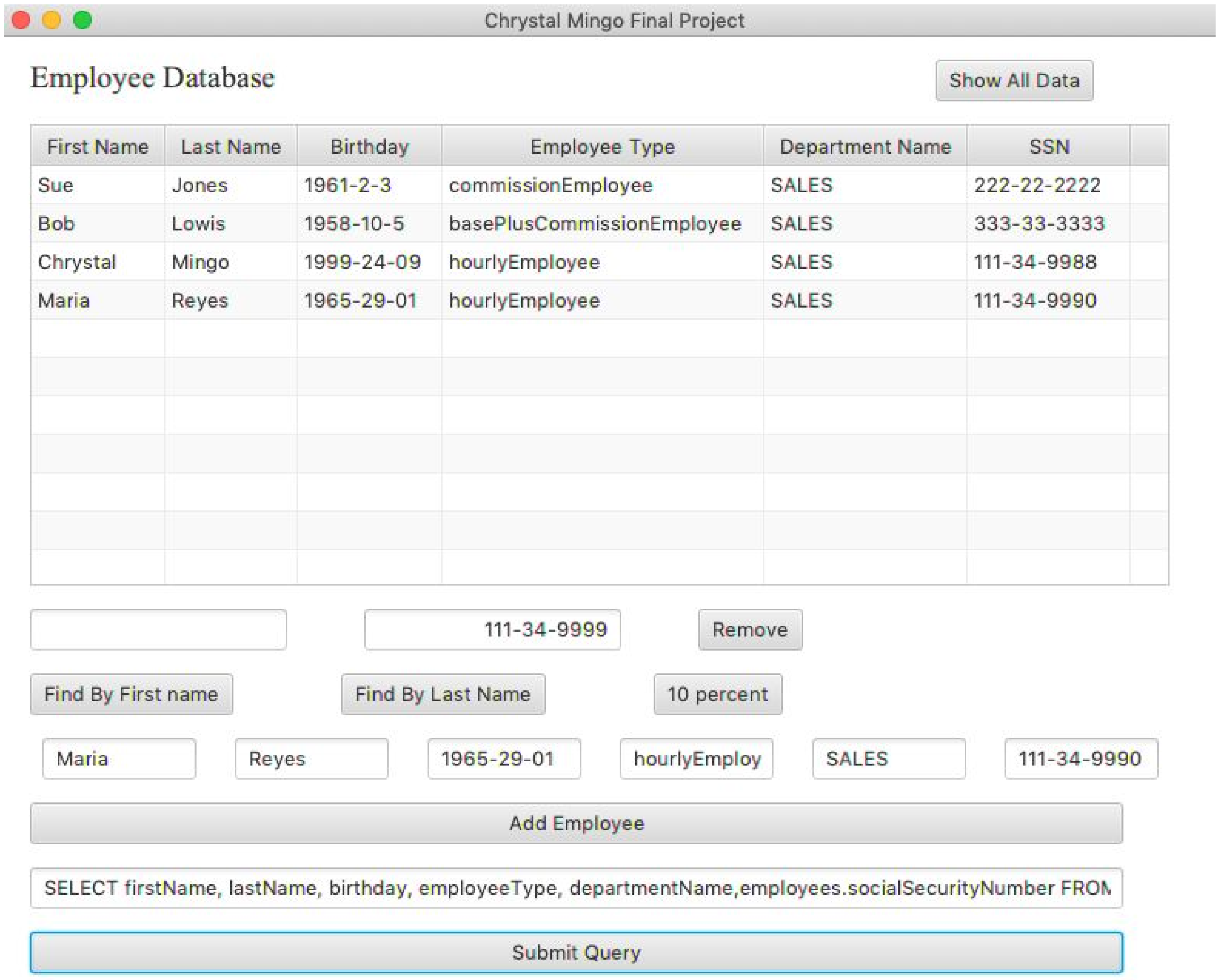
111-34-9999

Maria | Reyes | 1965-29-01 | hourlyEmploy | SALES | 111-34-9990

SELECT * FROM employees

SQL Script:

SELECT firstName, lastName, birthday, employeeType, departmentName,employees.socialSecurityNumber FROM Employees WHERE departmentName = 'SALES'



| First Name | Last Name | Birthday | Employee Type | Department Name | SSN |
|---|---|---|---|---|---|
| Sue | Jones | 1961-2-3 | commissionEmployee | SALES | 222-22-2222 |
| Bob | Lowis | 1958-10-5 | basePlusCommissionEmployee | SALES | 333-33-3333 |
| Chrystal | Mingo | 1999-24-09 | hourlyEmployee | SALES | 111-34-9988 |
| Maria | Reyes | 1965-29-01 | hourlyEmployee | SALES | 111-34-9990 |

111-34-9999

Maria | Reyes | 1965-29-01 | hourlyEmploy | SALES | 111-34-9990

SELECT firstName, lastName, birthday, employeeType, departmentName,employees.socialSecurityNumber FRON