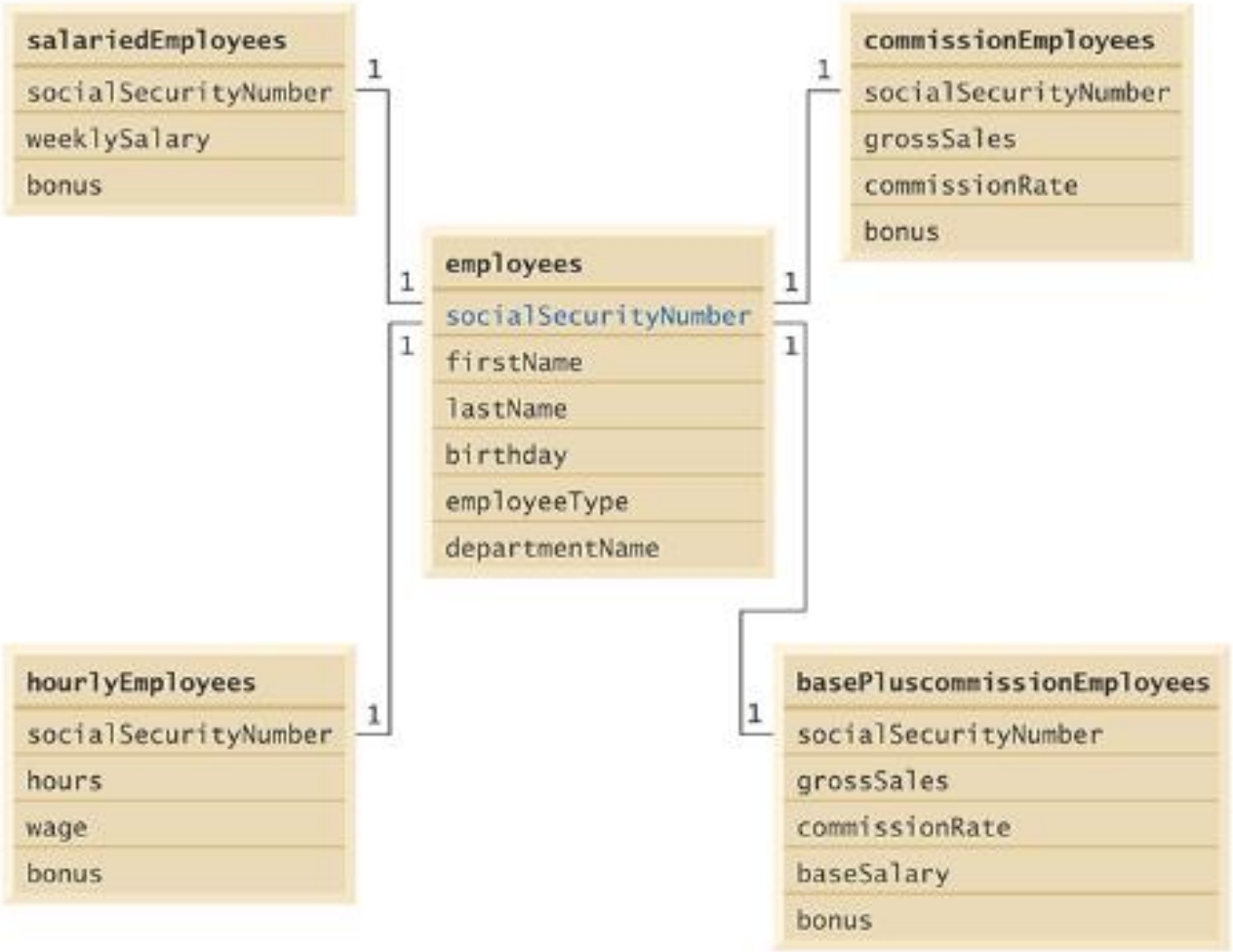
Mingzhi Xu

CSC22100 Software Design Laboratory Fall 2018

Project Employee Database

In this project, we will be creating a local database which will store information of the employees and design a user interface to manipulate and display the data from the database. The Employee database follows the hierarchy shown as below.



As we can see, we will have five tables in this Employee database, and the variable social security number exists in all five table which serves as the primary key to connect the tables to the employees table. We will be using SQLite, a relational database management system, to connect the database to the interface.

```java
//Mingzhi Xu
package com.company;

import ...

public class Main extends Application {
    private Connection DBConnect;
    String url = "jdbc:sqlite:C:¥¥Users¥¥Ming¥¥Desktop¥¥test6¥¥employee.db";
    private Stage stage;
    private Statement lines;
    private Button addEmployee, addSalariedEmployee, addCommissionEmployee, addBasePlusCommissionEmployee, addHourlyEmployee, Submit;
    private ComboBox comboBox;
    private ResultSet setOutput;
    private ObservableList<ObservableList> data = FXCollections.observableArrayList();
    private TableView tableview;
    private VBox vbox;
    private TextField input;

    @Override
    public void start(Stage primaryStage) {
        //connect to database
        try{
            DBConnect = DriverManager.getConnection(url);
            System.out.println("Connect Success");
        } catch(SQLException e){
            System.err.println("Connection Error");
            e.printStackTrace();
            System.exit( status: 1);
        }
```

Now we will create an interface such that allows the user to add employees to the Employee table and payroll information to the appropriate table for each new employee. For example, for a salaried employee add the payroll information to the SalariedEmployees table.

```java
        //24.4 Add employee and employee infos
        addEmployee = new Button( text: "Add New Employee");
        addSalariedEmployee = new Button( text: "Add Salaried Employee");
        addCommissionEmployee = new Button( text: "Add Commission Employee");
        addBasePlusCommissionEmployee = new Button( text: "Add Base Plus Commission Employee");
        addHourlyEmployee = new Button( text: "Add Hourly Employee");
        addEmployee.setOnAction(e -> actionPerformed(addEmployee));
        addSalariedEmployee.setOnAction(e -> actionPerformed(addSalariedEmployee));
        addCommissionEmployee.setOnAction(e -> actionPerformed(addCommissionEmployee));
        addBasePlusCommissionEmployee.setOnAction(e -> actionPerformed(addBasePlusCommissionEmployee));
        addHourlyEmployee.setOnAction(e -> actionPerformed(addHourlyEmployee));
        HBox addButtons = new HBox( spacing: 8);
        addButtons.getChildren().addAll(addEmployee, addSalariedEmployee, addCommissionEmployee,
                addBasePlusCommissionEmployee, addHourlyEmployee);
```

This creates buttons for inserting new employees into the employees table and insert

information of provided new employee social security number into the other tables corresponds to the employee's type.

```java
    //function for addbuttons
    public void actionPerformed( Button event) {
        String socialSecurityNumber = showTextInput( title: "Employee Social Security Number",  message: "Enter Emppployee's SSN",  defaultValue: "xxx-xx-xxxx");
        String insertQuery = "";
        if (event == addEmployee) {
            String fName = showTextInput( title: "Add Employee Info", message: "Enter First Name",  defaultValue: "First");
            String lName = showTextInput( title: "Add Employee Info", message: "Enter Last Name",  defaultValue: "Last");
            String day = showTextInput( title: "Add Employee Info", message: "Enter Employee Birthday",  defaultValue: "xxxx-xx-xx");
            String eType = showTextInput( title: "Add Employee Info", message: "Enter Employee Type",  defaultValue: "salariedEmployee");
            String dName = showTextInput( title: "Add Employee Info", message: "Enter Department Name",  defaultValue: "SALES");
            insertQuery = "INSERT INTO employees VALUES ( '" + socialSecurityNumber + "', '" + fName + "', '" + lName
                    + "', '" + day + "', '" + eType + "', '" + dName + "')";
        }
        else if (event == addSalariedEmployee) {
            double weeklySalary = Double.parseDouble(showTextInput( title: "Add Salaried Employee Info", message: "Enter Weekly Salary",  defaultValue: "0.0"));
            insertQuery = "INSERT INTO salariedEmployees VALUES ( '" + socialSecurityNumber + "', '" + weeklySalary + "', '0' )";
        } else if (event == addHourlyEmployee) {
            int hours = Integer.parseInt(showTextInput( title: "Add Hourly Employee Info", message: "Enter Hours",  defaultValue: "0"));
            double wage = Double.parseDouble(showTextInput( title: "Add Hourly Employee Info", message: "Enter Wage",  defaultValue: "0.0"));
            insertQuery = "INSERT INTO hourlyEmployees VALUES ( '" + socialSecurityNumber + "', '" + hours + "', '" + wage + "', '0' )";
        } else if (event == addCommissionEmployee) {
            int grossSales = Integer.parseInt(showTextInput( title: "Add Commission Employee Info", message: "Enter Gross Sales",  defaultValue: "0"));
            double commissionRate = Double.parseDouble(showTextInput( title: "Add Commission Employee Info", message: "Enter Commission Rate",  defaultValue: "0.00"));
            insertQuery = "INSERT INTO commissionEmployees VALUES ( '" + socialSecurityNumber + "', '" + grossSales + "', '" + commissionRate + "', '0' )";
        } else if (event == addBasePlusCommissionEmployee) {
            int grossSales = Integer.parseInt(showTextInput( title: "Add Base Plus Commission Employee Info", message: "Enter Gross Sales",  defaultValue: "0"));
            double commissionRate = Double.parseDouble(showTextInput( title: "Add Base Plus Commission Employee Info", message: "Enter Commission Rate",  defaultValue: "0.00"));
            double baseSalary = Double.parseDouble(showTextInput( title: "Add Base Plus Commission Employee Info", message: "Enter Base Salary",  defaultValue: "0.00"));
            insertQuery = "INSERT INTO basePlusCommissionEmployees " + "VALUES ( '" + socialSecurityNumber + "','"
                    + grossSales + "', '" + commissionRate + "', '" + baseSalary + "', '0')";
        } else {
            System.out.println("Fail to Insert");
```

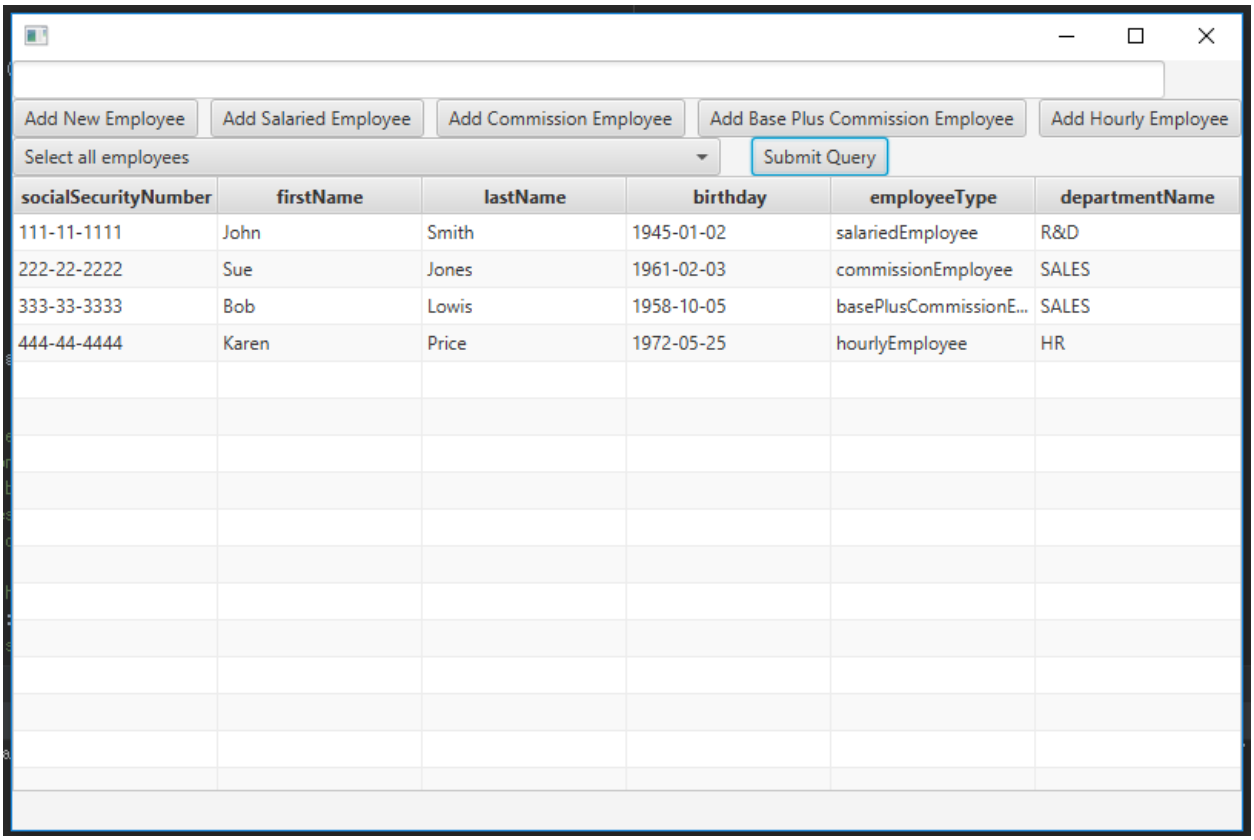
```java
        } else {
            System.out.println("Fail to Insert");
        }
        // execute insert query
        try
        {
            lines = DBConnect.createStatement();
            lines.executeUpdate(insertQuery);
        }
        catch ( SQLException exception)
        {
            System.err.println("Error");
            exception.printStackTrace();
        }
    }
```

The buttons will perform the actionPerformed function to prompt for user input using the showTextInput function for variables shown in the Employee database hierarchy. Then the user input will be inserted into the prepared statements which will be executed to insert information into the database.

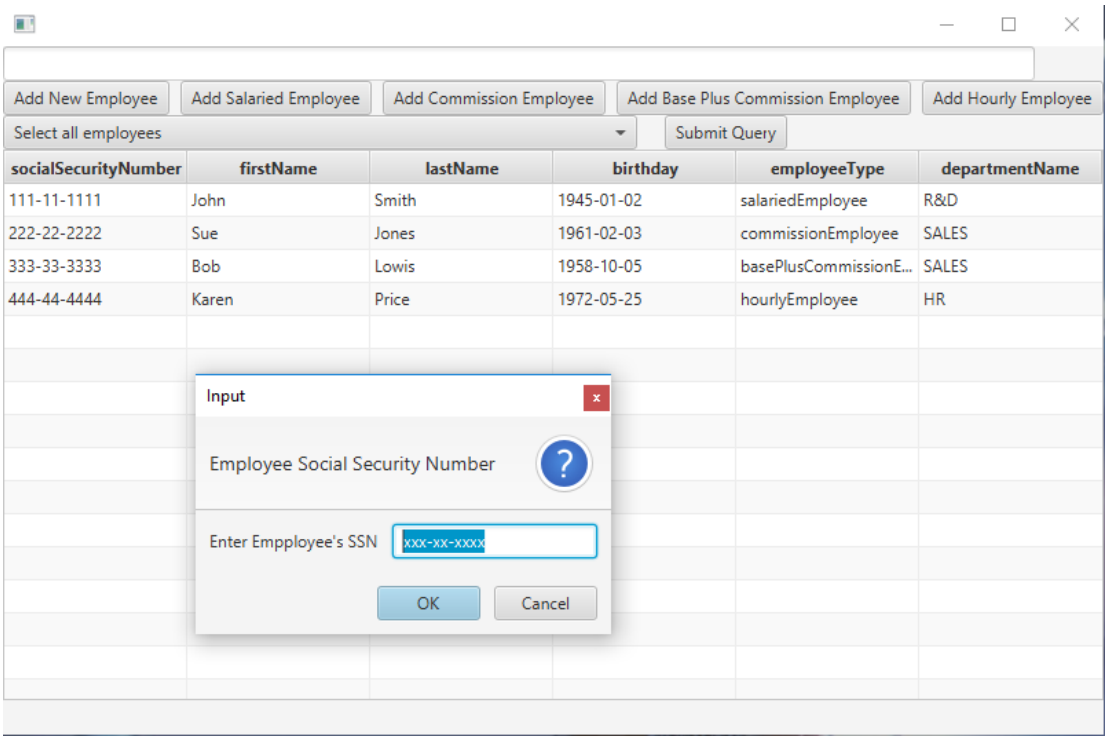
```java
    //User Input Box
    public static String showTextInput(String title, String message, String defaultValue) {
        TextInputDialog dialog = new TextInputDialog(defaultValue);
        dialog.initStyle(StageStyle.UTILITY);
        dialog.setTitle("Input");
        dialog.setHeaderText(title);
        dialog.setContentText(message);
        Optional<String> result = dialog.showAndWait();
        if (result.isPresent()) {
            return result.get();
        } else {
            return null;
        }
    }
```

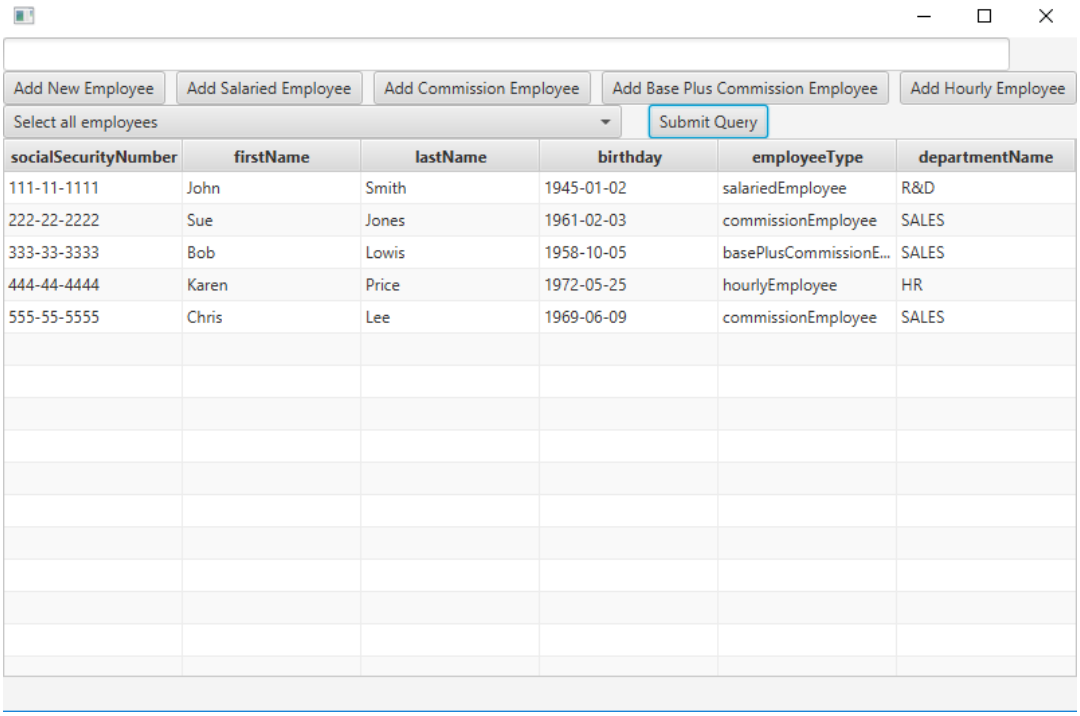
The function showTextInput works similarly to the JOptionPane in swing but since we are using Javafx, we are not able to use JOptionPane.

Result for 24.4

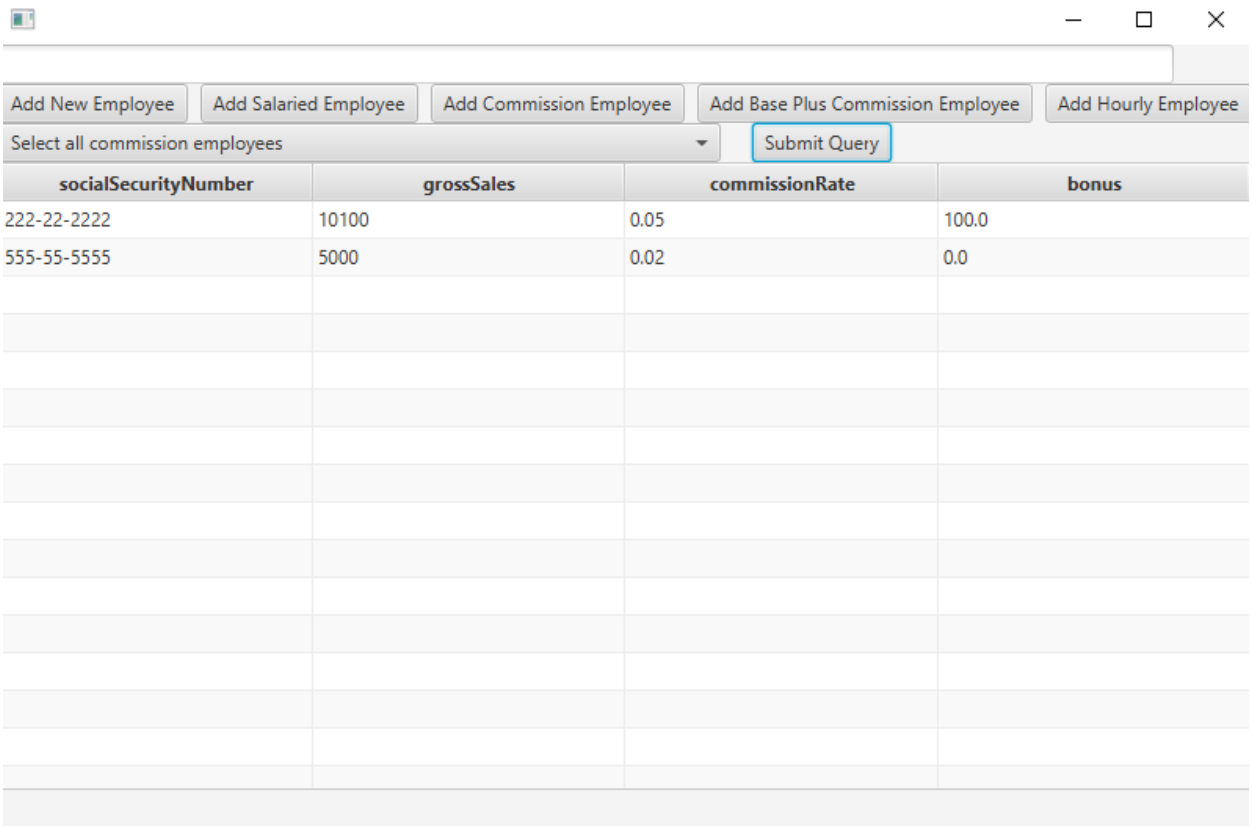


Now we will insert a new employee with the social security number 555-55-5555, first name Chris, last name Lee, birthday 1969-06-09, employee type commissionEmployee, and in the sales department. A window will pop out to prompt for those user inputs.





After that we want to insert payroll information for the new employee, so we look at the employee type which is commissionEmployee, so we click on the button Add Commission Employee and same thing as add employee a pop out window will prompt for user input.



Next, we will modify the interface to contain a combo box and a text area to allow the user to perform a query that is either selected from the combo box or input into the text area. Some predefined queries would include 1) Select all employees working in the department SALES. 2) Select hourly employees working over 30 hours. 3) Select all commission employees in descending order of the commission rate.

```java
        //24.5 Show Table and Query
        input = new TextField();
        input.setPrefWidth(750);
        input.setOnAction(e -> {
                try{
                    String newQuery = input.getText();
                    lines = DBConnect.createStatement();
                    exeQ(newQuery);
                }
                catch(SQLException ex){
                    System.err.println("Error");
                    ex.printStackTrace();
                }
            }
        );
        Submit = new Button( text: "Submit Query");
        Submit.setOnAction(event -> {
            tableview.getItems().clear();
            tableview.getColumns().clear();
            showDBTable();
            tableview.setColumnResizePolicy(TableView.CONSTRAINED_RESIZE_POLICY);
        });
```

Here we will create a text field which is used for the text area for user input of their own queries and a submit button that will display the table based on the preselected queries in the combo box. Below is list for the combo box and layouts of the interface.

```java
        ObservableList<String> options = FXCollections.observableArrayList(
                ...items: "Select all employees",
                "Select all base plus commission employees",
                "Select all commission employees",
                "Select all hourly employees",
                "Select all salaried employees",
                "Select all employees working in Department SALES",
                "Select hourly employees working over 30 hours",
                "Select all commission employees in descending order of the commission rate",
                "Increase base salary by 10% for all base plus commission employees",
                "If the employee's birthday is in the current month, add a $100 bonus",
                "For all commission employees with gross sales over $10000, add a $100 bonus"
                );
        comboBox = new ComboBox(options);
        HBox inQ = new HBox( spacing: 20);
        inQ.getChildren().addAll(input);
        HBox comSub = new HBox( spacing: 20);
        comSub.getChildren().addAll(comboBox, Submit);
        BorderPane pane = new BorderPane();
        pane.setTop(inQ);
        pane.setCenter(addButtons);
        pane.setBottom(comSub);
        tableview = new TableView();
        vbox = new VBox();
        vbox.getChildren().add(tableview);
        BorderPane panel = new BorderPane();
        panel.setTop(pane);
        panel.setCenter(vbox);
        Scene scene = new Scene(panel, width: 800, height: 500);
        stage = new Stage();
        stage.setScene(scene);
        stage.show();
    }
```

```java
//Combo Box Statement
private void showDBTable(){
    try{
        String queryStatement = null;
        String s = comboBox.getSelectionModel().getSelectedItem().toString();
        switch(s){
            case "Select all employees":
                queryStatement = "SELECT * FROM employees"; break;
            case "Select all base plus commission employees":
                queryStatement = "SELECT * FROM basePlusCommissionEmployees"; break;
            case "Select all commission employees":
                queryStatement = "SELECT * FROM commissionEmployees"; break;
            case "Select all hourly employees":
                queryStatement = "SELECT * FROM hourlyEmployees"; break;
            case "Select all salaried employees":
                queryStatement = "SELECT * FROM salariedEmployees"; break;
            case "Select all employees working in Department SALES":
                queryStatement = "SELECT * FROM employees WHERE " + "departmentName = 'SALES'"; break;
            case "Select hourly employees working over 30 hours":
                queryStatement = "SELECT * FROM hourlyEmployees WHERE hours >= 30"; break;
            case "Select all commission employees in descending order of the commission rate":
                queryStatement = "SELECT * FROM commissionEmployees ORDER BY " + "commissionRate DESC"; break;
            case "Increase base salary by 10% for all base plus commission employees":
```

This contains prepared statements for the selected queries that exists in the combo box.

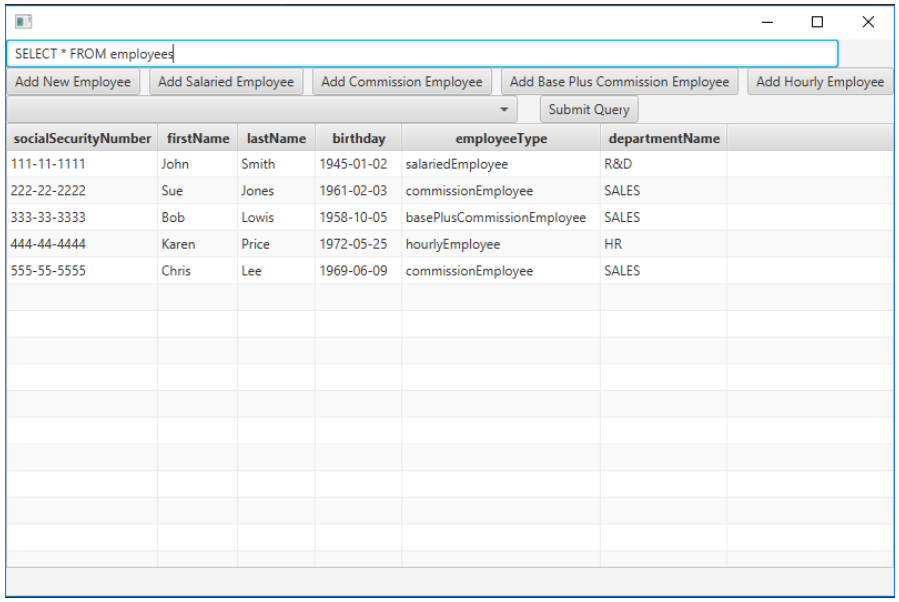
```java
//Execute Query Statement
private void exeQ(String queryStatement) throws SQLException {
    lines = DBConnect.createStatement();
    if(queryStatement.substring(0,6).equals("SELECT")){
        setOutput = lines.executeQuery(queryStatement);
        displaySet(setOutput);
    }
    else lines.executeUpdate(queryStatement);
}
```

The function exeQ was created so there is not needing to keep calling create statement and execute.

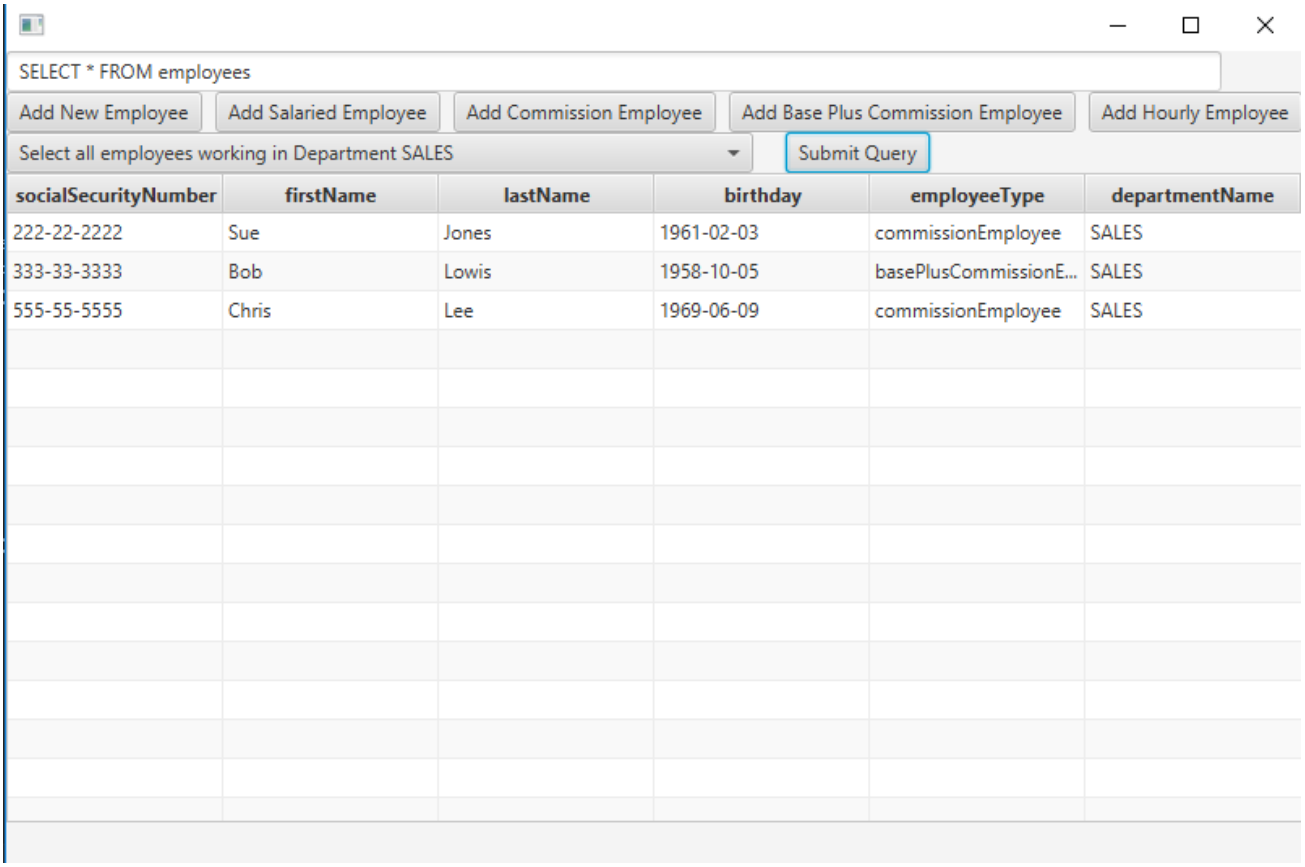
```java
    //Display Table
    private void displaySet(ResultSet r)throws SQLException {
        tableview.getItems().clear();
        tableview.getColumns().clear();
        boolean numRecords = r.next();
        if (!numRecords) {
            System.out.println("No records to display");
            return;
        }
        try {
            int i;
            for (i = 0; i < r.getMetaData().getColumnCount(); i++) {
                final int j = i;
                TableColumn col = new TableColumn(r.getMetaData().getColumnName(i + 1));
                col.setCellValueFactory((Callback<TableColumn.CellDataFeatures<ObservableList, String>, ObservableValue<String>>)
                        param -> new SimpleStringProperty(param.getValue().get(j).toString()));
                tableview.getColumns().addAll(col);
            }
            ObservableList<String> frow = FXCollections.observableArrayList();
            for (i = 0; i < r.getMetaData().getColumnCount(); i++) {
                frow.add(r.getString( columnIndex: i + 1));
            }
            data.add(frow);
            while (r.next()) {
                ObservableList<String> row = FXCollections.observableArrayList();
                for (i = 0; i < r.getMetaData().getColumnCount(); i++) {
                    row.add(r.getString( columnIndex: i + 1));
                    }
                    data.add(row);
            }
            tableview.setItems(data);
        }catch(SQLException ex){
            System.err.println("Error");
            ex.printStackTrace();
        }
```

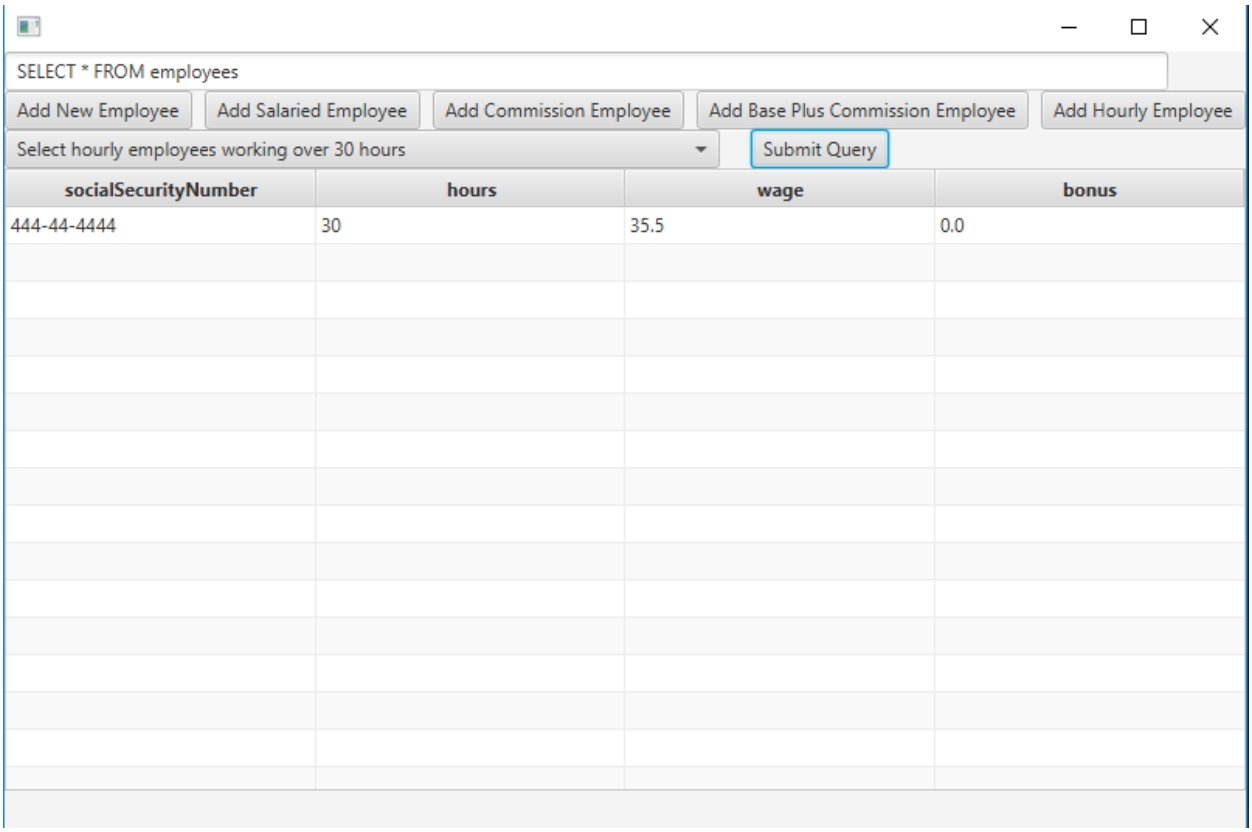
The displaySet will take data from the database and insert into table view which will be displayed on the interface.

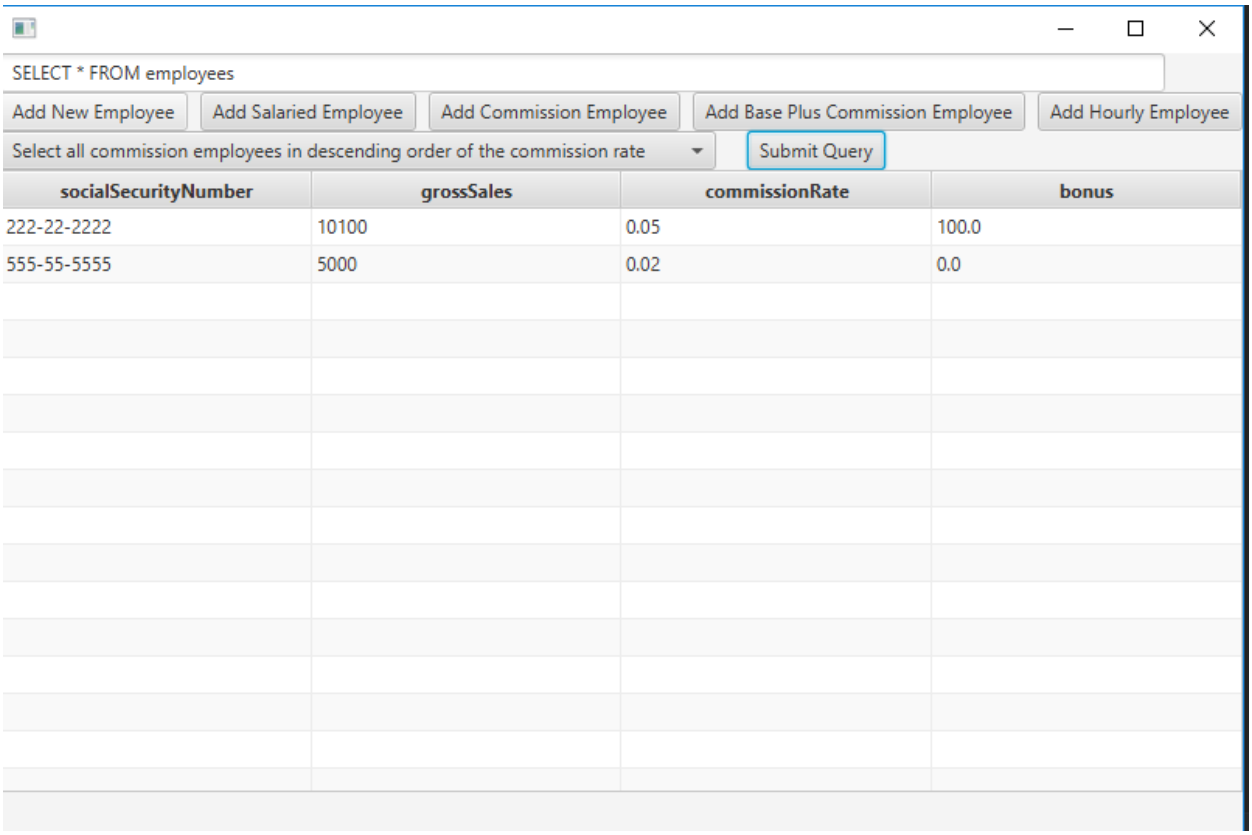
Result for 24.5



socialSecurityNumber	firstName	lastName	birthday	employeeType	departmentName
111-11-1111	John	Smith	1945-01-02	salariedEmployee	R&D
222-22-2222	Sue	Jones	1961-02-03	commissionEmployee	SALES
333-33-3333	Bob	Lowis	1958-10-05	basePlusCommissionEmployee	SALES
444-44-4444	Karen	Price	1972-05-25	hourlyEmployee	HR
555-55-5555	Chris	Lee	1969-06-09	commissionEmployee	SALES


SELECT * FROM employees
Add New Employee
Add Salaried Employee
Add Commission Employee
Add Base Plus Commission Employee
Add Hourly Employee
Select all employees working in Department SALES
Submit Query
socialSecurityNumber
firstName
lastName
birthday
employeeType
departmentName
222-22-2222
Sue
Jones
1961-02-03
commissionEmployee
SALES
333-33-3333
Bob
Lowis
1958-10-05
basePlusCommissionE...
SALES
555-55-5555
Chris
Lee
1969-06-09
commissionEmployee
SALES


SELECT * FROM employees
Add New Employee
Add Salaried Employee
Add Commission Employee
Add Base Plus Commission Employee
Add Hourly Employee
Select hourly employees working over 30 hours
Submit Query
socialSecurityNumber
hours
wage
bonus
444-44-4444
30
35.5
0.0



Next We will modify the interface to perform the following tasks, increase base salary by 10% for all base plus commission employees, if the employee's birthday is in the current month add a $100 bonus, and for all commission employees with gross sales over $10000, add a $100 bonus. These are going to be implemented as predefined queries, so it will be in the combo box as you can see from the query list from above and similarly, we will put these cases into showDBTable function.

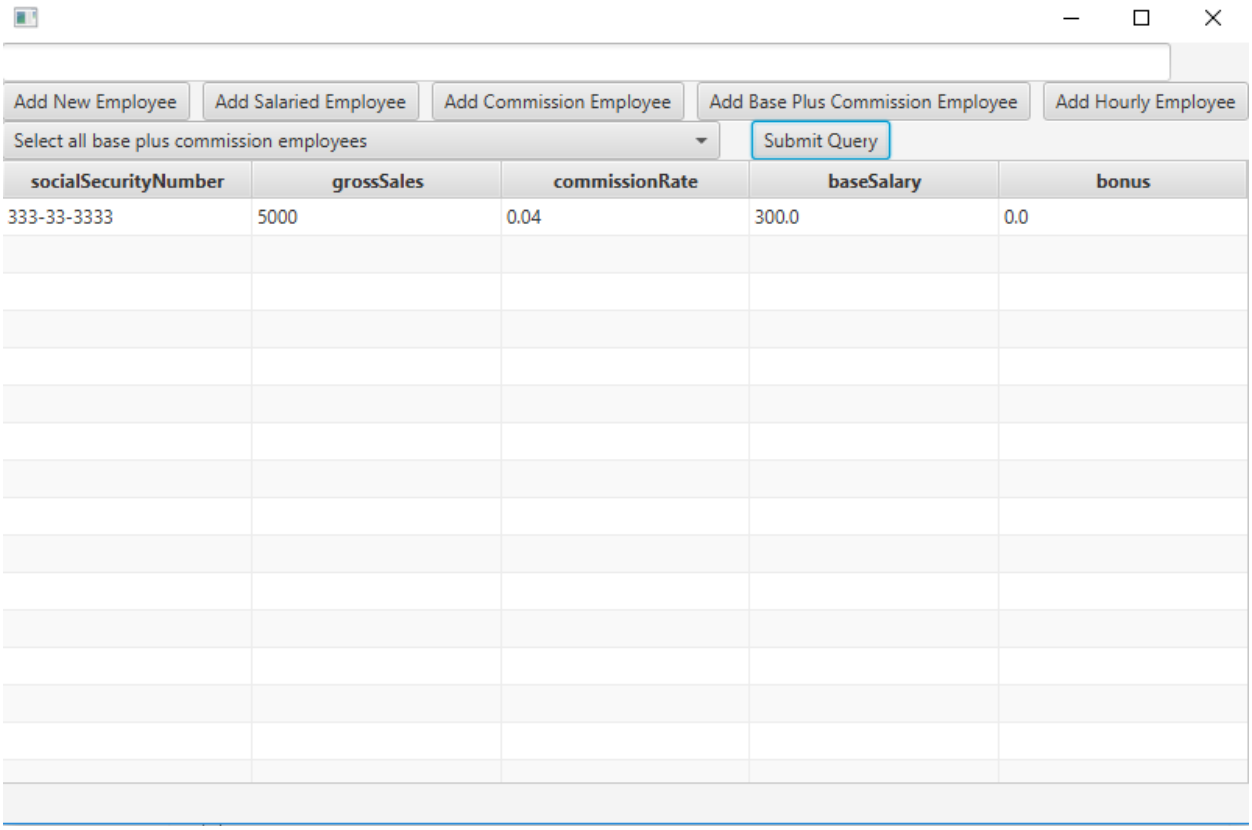
```java
                case "Increase base salary by 10% for all base plus commission employees":
                    queryStatement = "UPDATE basePlusCommissionEmployees SET " + "baseSalary = baseSalary * 1.1";
                    lines = DBConnect.createStatement();
                    exeQ(queryStatement);
                    queryStatement = "SELECT * FROM basePlusCommissionEmployees"; break;
                case "For all commission employees with gross sales over $10000, add a $100 bonus":
                    queryStatement = "UPDATE commissionEmployees SET " + "bonus = bonus + 100.00 WHERE grossSales >= 10000";
                    lines = DBConnect.createStatement();
                    exeQ(queryStatement);
                    queryStatement = "SELECT * FROM commissionEmployees"; break;
                case "If the employee's birthday is in the current month, add a $100 bonus":
                    bdBonus();
                    queryStatement = "SELECT * FROM employees"; break;
            }
            lines = DBConnect.createStatement();
            exeQ(queryStatement);
        }
        catch(SQLException ex){
            System.err.println("Error");
            ex.printStackTrace();
        }
    }
```

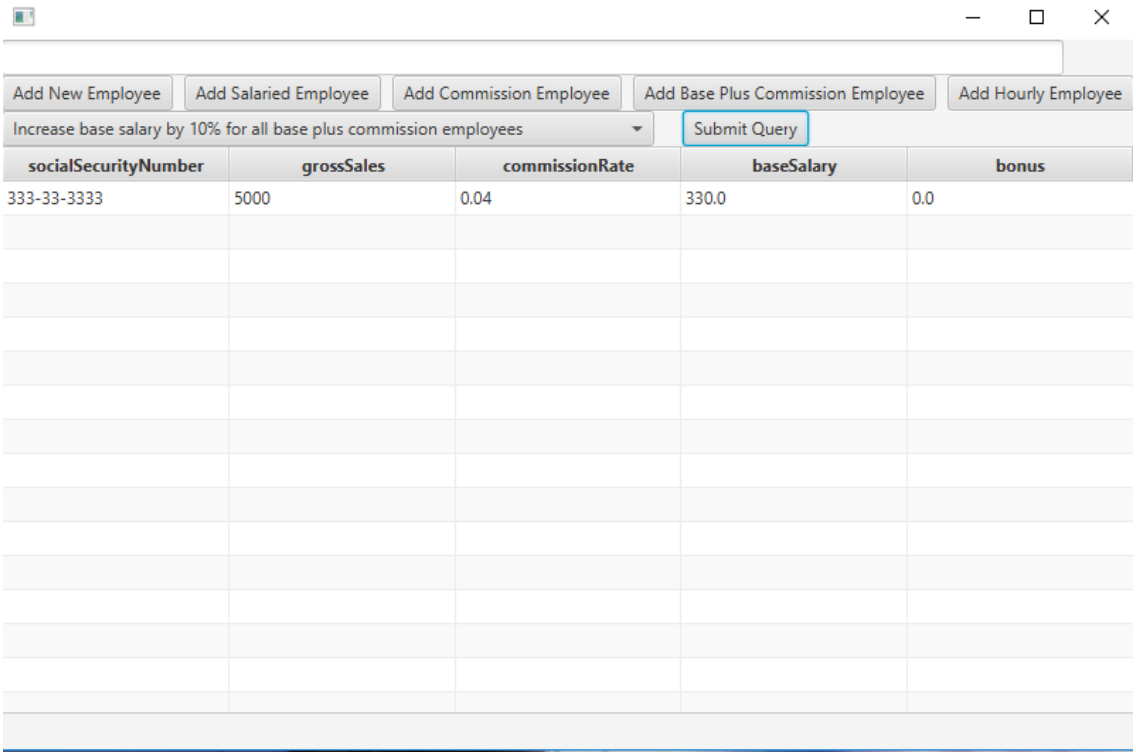
However, the case which adds bonus to employee if birthday is in the current month requires you to define the month of the current month, so the function bdBonus is created to prompt for user input for current month and takes data from database to compare and execute the bonus into the payroll information.

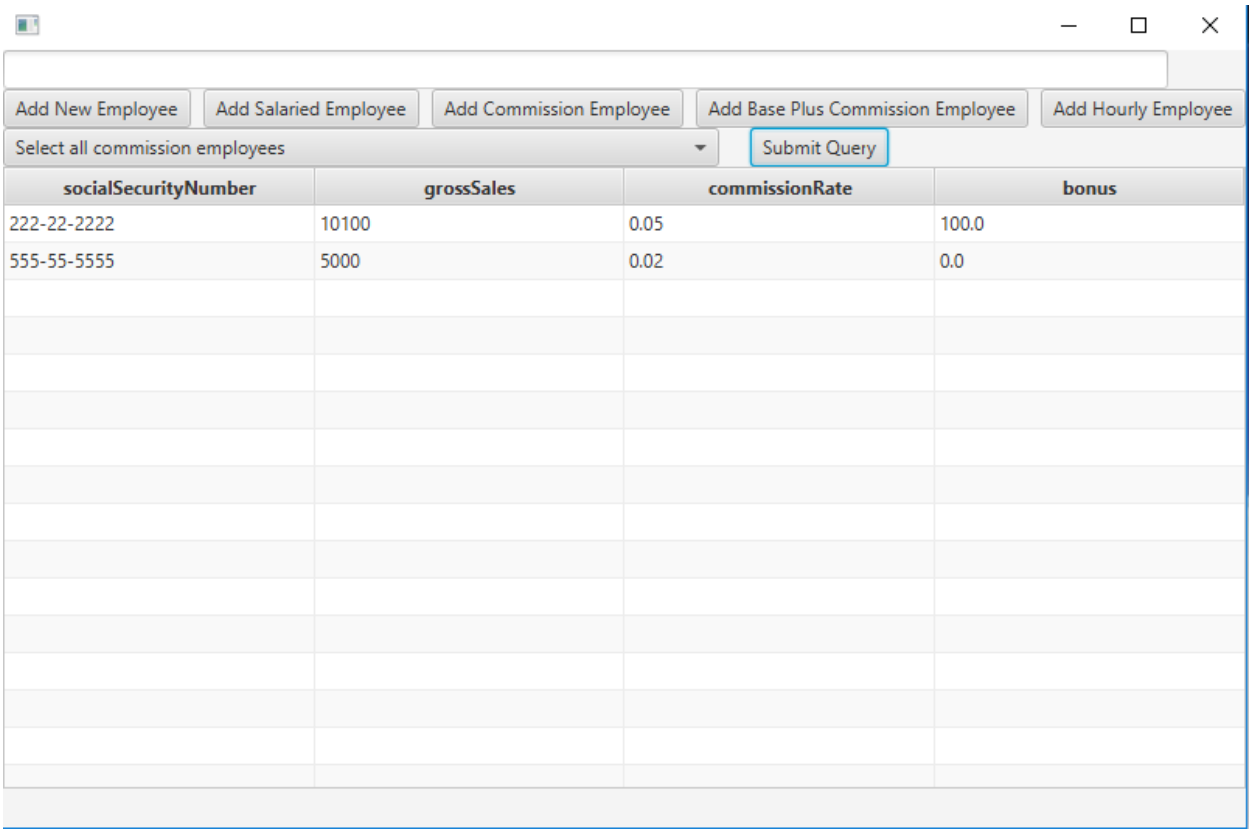
```java
    //Birthday Bonus
    private void bdBonus() throws SQLException {
        String day, ebdMonth;
        String bdMonth = showTextInput( title: "Birthday Bonus to Employees", message: "Enter Current Month",  defaultValue: "xx");
        try{
            lines = DBConnect.createStatement();
            setOutput = lines.executeQuery( sql: "SELECT * FROM employees");
            while(setOutput.next()){
                day = setOutput.getString( columnIndex: 4);
                ebdMonth = day.substring(5, 7);
                checkbd(ebdMonth, bdMonth);
            }
        }
        catch (SQLException e){ e.printStackTrace();}
    }
```

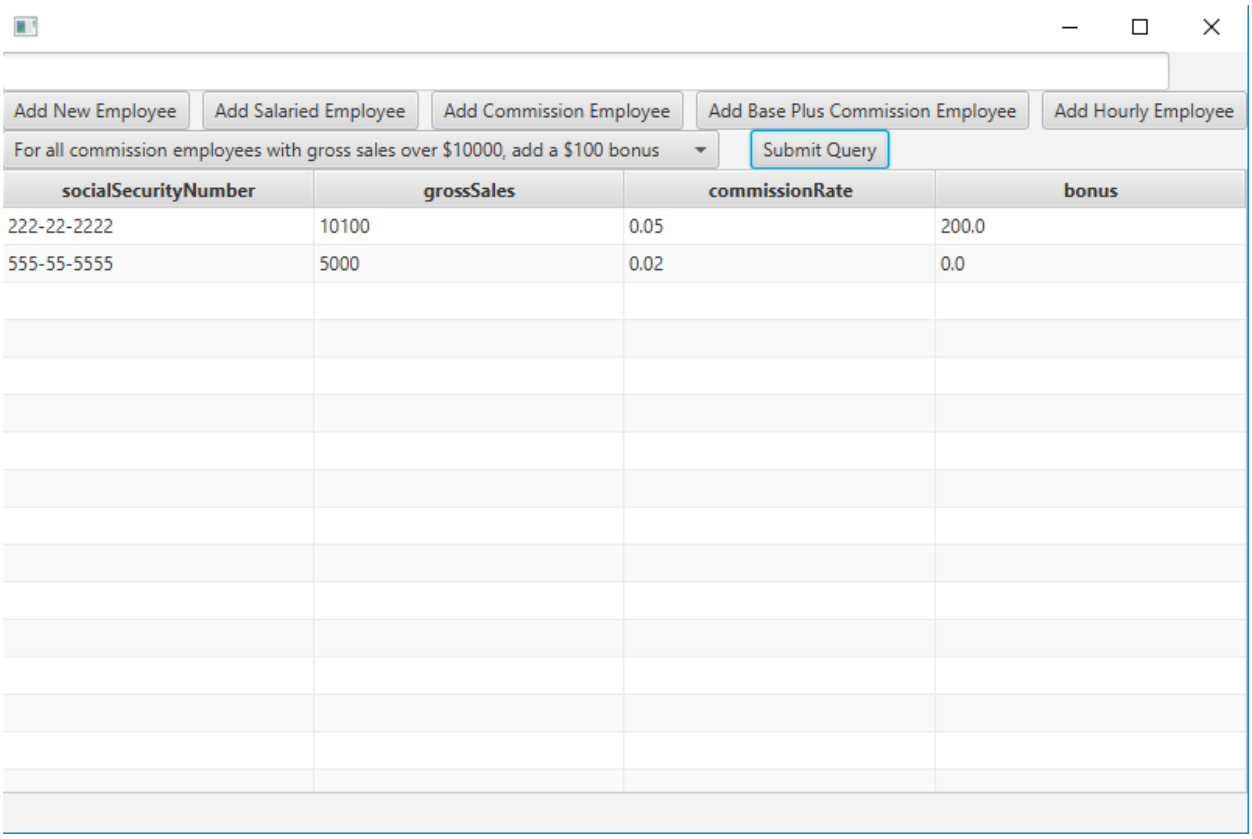
```java
    private void checkbd(String ebdMonth, String bdMonth) throws SQLException {
        String ssn;
        String employType;
        if(ebdMonth.equals(bdMonth)){
            ssn = setOutput.getString( columnIndex: 1);
            employType = setOutput.getString( columnIndex: 5);
            String q = "UPDATE " + employType + "s SET " + "bonus = bonus + 100.00 WHERE socialSecurityNumber = "
                    + "'" + ssn + "'";
            exeQ(q);
        }
    }
```

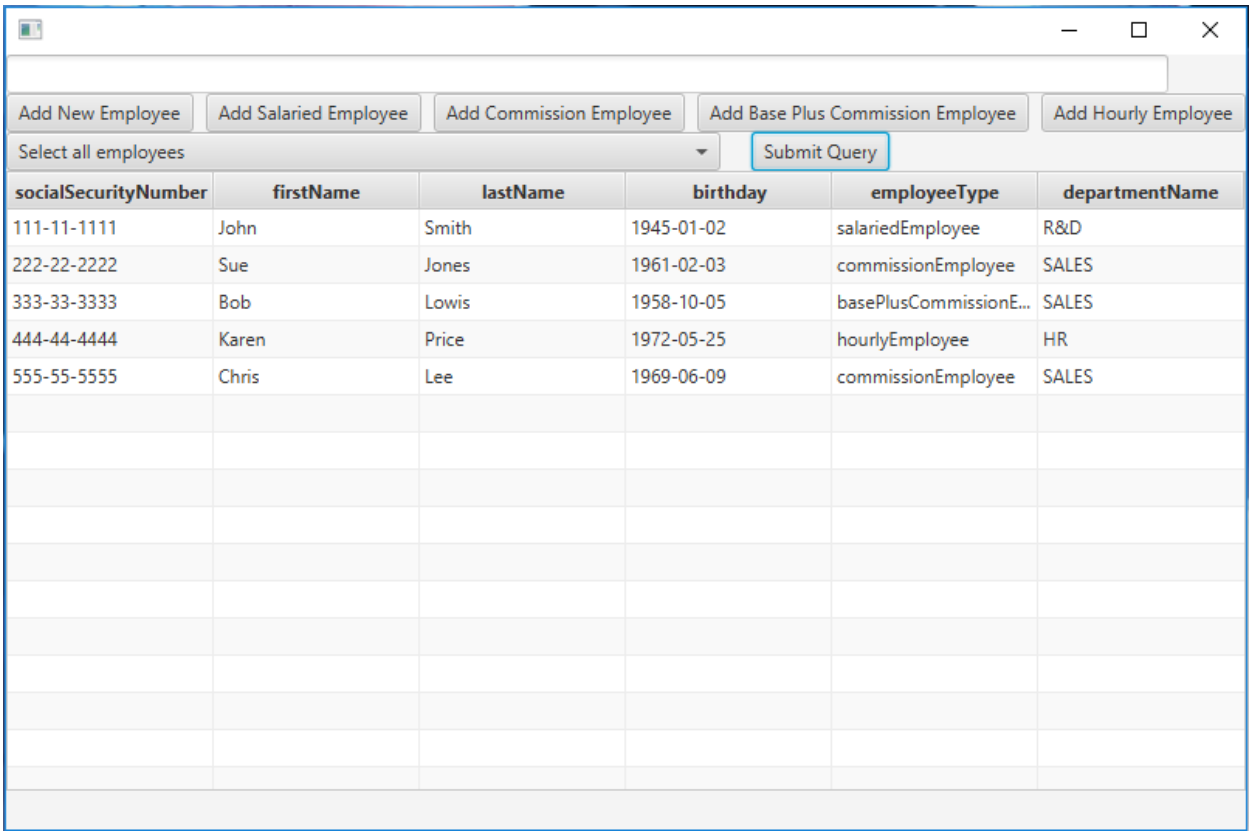
Result for 24.6

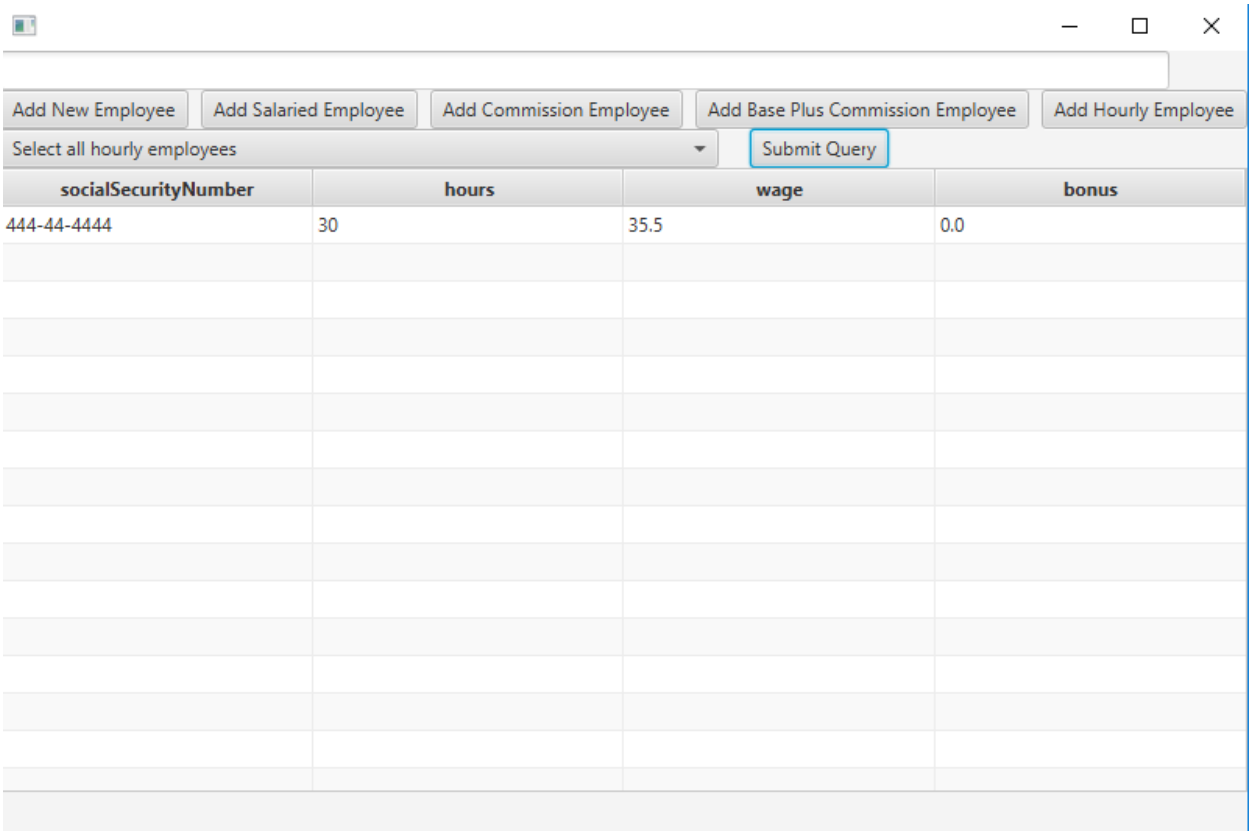


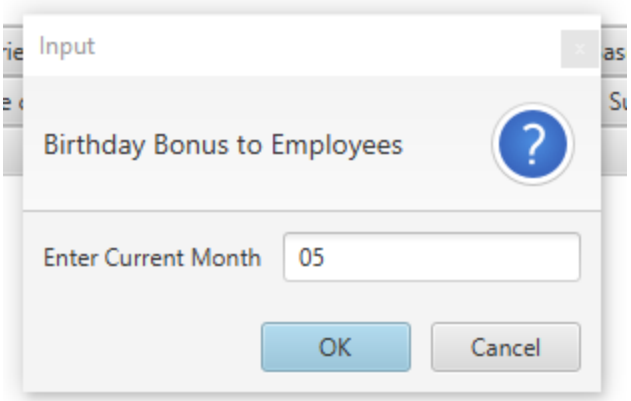


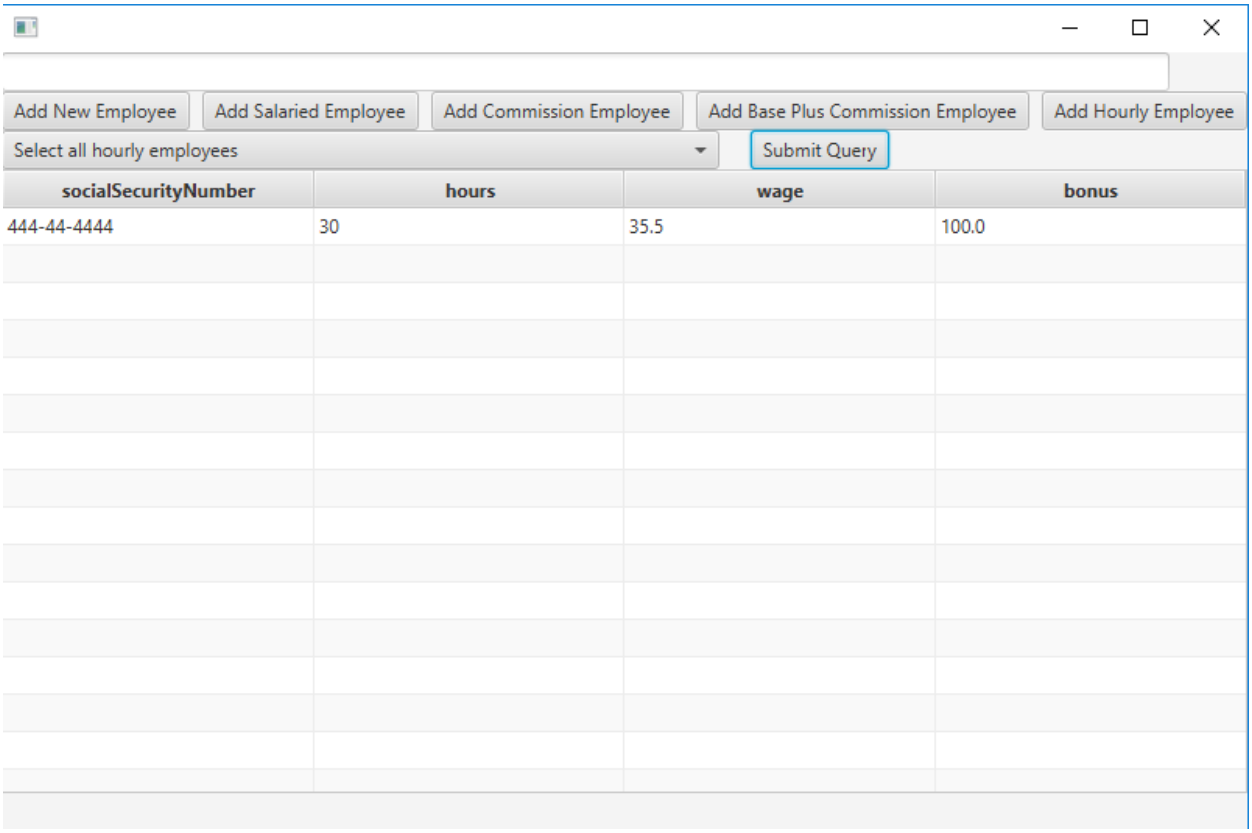

Add New Employee
Add Salaried Employee
Add Commission Employee
Add Base Plus Commission Employee
Add Hourly Employee
Select all commission employees
Submit Query
socialSecurityNumber
grossSales
commissionRate
bonus
222-22-2222
10100
0.05
100.0
555-55-5555
5000
0.02
0.0


Add New Employee
Add Salaried Employee
Add Commission Employee
Add Base Plus Commission Employee
Add Hourly Employee
For all commission employees with gross sales over $10000, add a $100 bonus
Submit Query
socialSecurityNumber
grossSales
commissionRate
bonus
222-22-2222
10100
0.05
200.0
555-55-5555
5000
0.02
0.0


Add New Employee
Add Salaried Employee
Add Commission Employee
Add Base Plus Commission Employee
Add Hourly Employee
Select all employees
Submit Query
socialSecurityNumber
firstName
lastName
birthday
employeeType
departmentName
111-11-1111
John
Smith
1945-01-02
salariedEmployee
R&D
222-22-2222
Sue
Jones
1961-02-03
commissionEmployee
SALES
333-33-3333
Bob
Lowis
1958-10-05
basePlusCommissionE...
SALES
444-44-4444
Karen
Price
1972-05-25
hourlyEmployee
HR
555-55-5555
Chris
Lee
1969-06-09
commissionEmployee
SALES


Add New Employee
Add Salaried Employee
Add Commission Employee
Add Base Plus Commission Employee
Add Hourly Employee
Select all hourly employees
Submit Query
socialSecurityNumber
hours
wage
bonus
444-44-4444
30
35.5
0.0


Input
Birthday Bonus to Employees
Enter Current Month
05
OK
Cancel


Add New Employee
Add Salaried Employee
Add Commission Employee
Add Base Plus Commission Employee
Add Hourly Employee
Select all hourly employees
Submit Query
socialSecurityNumber
hours
wage
bonus
444-44-4444
30
35.5
100.0