

```
import javax.swing.*.*;

import javax.swing.table.DefaultTableCellRenderer;

import javax.swing.table.DefaultTableModel;

import java.awt.*.*;

import java.sql.*.*;

import java.util.Vector;


public class Main {

    private Connection conn;

    private JFrame frame;

    private JComboBox<String> comboBox;

    private JTable table;

    private DefaultTableModel tableModel;

    private JProgressBar progressBar;


    public Main() {

        // Establish connection to the PostgreSQL database

        connectDB();


        // Create the GUI

        createGUI();

    }


    private void connectDB() {

        try {

            String url = "jdbc:postgresql://localhost:5432/formula1";

            String user = "postgres";

            String password = "miguel";

            conn = DriverManager.getConnection(url, user, password);

            System.out.println("Connected to PostgreSQL.");

        } catch (SQLException e) {
```

```
        e.printStackTrace();
    }
}
```

```
private void createGUI() {
    frame = new JFrame("Driver Points by Year");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(600, 400);

    // Combo box to select the race year
    comboBox = new JComboBox<>();
    populateComboBox();
    comboBox.addActionListener(e -> {
        // When a year is selected, update the driver points table
        updateTableInBackground();
    });

    // Progress bar to show while the table is loading
    progressBar = new JProgressBar();
    progressBar.setStringPainted(true);

    // Table to display the driver points data
    tableModel = new DefaultTableModel();
    table = new JTable(tableModel);
    JScrollPane scrollPane = new JScrollPane(table);

    // Center the cell content
    DefaultTableCellRenderer centerRenderer = new DefaultTableCellRenderer();
    centerRenderer.setHorizontalAlignment(JLabel.CENTER);
    table.setDefaultRenderer(Object.class, centerRenderer);
}
```

```
frame.getContentPane().setLayout(new BorderLayout());  
frame.getContentPane().add(comboBox, BorderLayout.NORTH);  
frame.getContentPane().add(progressBar, BorderLayout.SOUTH);  
frame.getContentPane().add(scrollPane, BorderLayout.CENTER);
```

```
frame.setVisible(true);  
}
```

```
private void populateComboBox() {  
    try {  
        Statement stmt = conn.createStatement();  
        ResultSet rs = stmt.executeQuery("SELECT DISTINCT year FROM races ORDER BY year  
DESC");  
        while (rs.next()) {  
            comboBox.addItem(rs.getString("year"));  
        }  
        rs.close();  
        stmt.close();  
    } catch (SQLException e) {  
        e.printStackTrace();  
    }  
}
```

```
private void updateTableInBackground() {  
    String selectedYear = (String) comboBox.getSelectedItem();  
    if (selectedYear != null) {  
        // Create a SwingWorker to execute the query in the background  
        SwingWorker<Void, Void> worker = new SwingWorker<Void, Void>() {  
            @Override  
            protected Void doInBackground() throws Exception {  
                try {
```

```
// Query to get the drivers and their total points for the selected year
String query = "SELECT d.forename || ' ' || d.surname AS driver_name,
SUM(ds.points) AS total_points " +
    "FROM drivers d " +
    "JOIN driver_standings ds ON d.driver_id = ds.driver_id " +
    "JOIN races r ON ds.race_id = r.race_id " +
    "WHERE r.year = ? " +
    "GROUP BY driver_name " +
    "ORDER BY total_points DESC";
```

```
PreparedStatement pstmt = conn.prepareStatement(query);
pstmt.setInt(1, Integer.parseInt(selectedYear));
ResultSet rs = pstmt.executeQuery();
```

```
// Get columns
Vector<String> columnNames = new Vector<>();
columnNames.add("Driver Name");
columnNames.add("Total Points");
```

```
// Get rows
Vector<Vector<Object>> data = new Vector<>();
while (rs.next()) {
    Vector<Object> row = new Vector<>();
    row.add(rs.getString("driver_name"));
    row.add(rs.getDouble("total_points"));
    data.add(row);
}
```

```
// Update the table model on the Swing event dispatch thread
SwingUtilities.invokeLater(() -> {
    tableModel.setDataVector(data, columnNames);
});
```

```

        progressBar.setValue(100); // Complete the progress bar
    });

    rs.close();
    pstmt.close();
} catch (SQLException e) {
    e.printStackTrace();
}
return null;
}

@Override
protected void done() {
    // Additional actions after data loading is complete
    progressBar.setIndeterminate(false); // Stop the indeterminate state
}
};

// Start the SwingWorker and show the progress bar
progressBar.setValue(0); // Reset the progress bar
progressBar.setIndeterminate(true); // Show an indeterminate progress bar
worker.execute();
}
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(Main::new);
}
}

```

Driver Points by Year	
2013	
Driver Name	Total Points
Sebastian Vettel	3646.0
Fernando Alonso	2541.0
Kimi Räikkönen	2272.0
Lewis Hamilton	2099.0
Mark Webber	1892.0
Nico Rosberg	1613.0
Felipe Massa	1246.0
Romain Grosjean	1047.0
Jenson Button	714.0
Paul di Resta	596.0
Adrian Sutil	375.0
Sergio Pérez	358.0
Nico Hülkenberg	338.0
Daniel Ricciardo	227.0
Jean-Éric Vergne	178.0
Esteban Gutiérrez	30.0
Pastor Maldonado	10.0
Valtteri Bottas	8.0
Charles Pic	0.0
100 %	