package application;

import java.sql.\*;

import javafx.application.Application;

import javafx.beans.property.SimpleStringProperty;

import javafx.beans.value.ObservableValue;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.geometry.Insets;

import javafx.stage.Stage;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.BorderPane;

import javafx.scene.layout.FlowPane;

import javafx.scene.layout.VBox;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableColumn.CellDataFeatures;

import javafx.scene.control.TableView;

import javafx.scene.text.Font;

import javafx.util.Callback;

public class Main extends Application{

static final String DB\_URL = "jdbc:mysql://localhost:3306/pluginsdb";

static final String USER = "####";

static final String PASS = "###########";

TextField statementTF = newTextField();

Label myLabel = new Label();

Button submitButton = newButton("Submit", 110, 50);

Button quitButton = newButton("Quit", 110, 50);

ObservableList<ObservableList<String>> data;

TableView table = new TableView();

@Override

public void start(Stage primaryStage)

{

try

{

BorderPane root = new BorderPane();

root.setLeft(inputSection());

root.setRight(tableSection());

Scene scene = new Scene(root, 670, 450);

primaryStage.setScene(scene);

primaryStage.show();

} catch (Exception e)

{

e.printStackTrace();

}

}

private VBox inputSection() {

VBox vbox = new VBox();

statementTF.setPromptText("Enter SQL Statement");

submitButton.setOnAction(t ->

{

performSQLStatement(statementTF.getText());

});

quitButton.setOnAction(t ->

{

Stage stage = (Stage) quitButton.getScene().getWindow();

stage.close();

});

vbox.getChildren().addAll(statementTF, submitButton, quitButton);

return vbox;

}

private VBox tableSection() {

VBox vbox = new VBox();

vbox.getChildren().addAll(myLabel, table);

return vbox;

}

private void performSQLStatement(String SQLStatement) {

try {

// test driver

try {

Class.forName("com.mysql.cj.jdbc.Driver");

System.out.println("Driver loaded!");

} catch (ClassNotFoundException e) {

throw new IllegalStateException("Cannot find the driver in the classpath!", e);

}

Connection conn = DriverManager.getConnection(DB\_URL, USER, PASS); // fix DB\_URL

Statement stmt = conn.createStatement(ResultSet.TYPE\_SCROLL\_INSENSITIVE, ResultSet.CONCUR\_UPDATABLE);

data = FXCollections.observableArrayList();

table.setEditable(true);

table.getItems().clear();

table.getColumns().clear();

if(SQLStatement.indexOf("update") != -1 || SQLStatement.indexOf("delete") != -1 || SQLStatement.indexOf("insert") != -1)

stmt.executeUpdate(SQLStatement);

else

stmt.execute(SQLStatement);

ResultSet rs = stmt.getResultSet();

// Retrieved from https://stackoverflow.com/questions/18941093/how-to-fill-up-a-tableview-with-database-data

for(int i = 0 ; i < rs.getMetaData().getColumnCount(); i++){

//We are using non property style for making dynamic table

final int j = i;

TableColumn col = new TableColumn(rs.getMetaData().getColumnName(i+1));

col.setCellValueFactory(new Callback<CellDataFeatures<ObservableList,String>,ObservableValue<String>>(){

public ObservableValue<String> call(CellDataFeatures<ObservableList, String> param) {

return new SimpleStringProperty(param.getValue().get(j).toString());

}

});

table.getColumns().addAll(col);

System.out.println("Column ["+i+"] ");

}

int j = 1;

while(rs.next()){

//Iterate Row

ObservableList<String> row = FXCollections.observableArrayList();

for(int i=1; i<=rs.getMetaData().getColumnCount(); i++){

//Iterate Column

if(rs.getString(i) == null)

row.add("NULL");

else

row.add(rs.getString(i));

}

System.out.println("Row [" + j + "] added "+row );

data.add(row);

j++;

}

//FINALLY ADDED TO TableView

table.setItems(data);

table.setEditable(false);

} catch(SQLException se) {

se.printStackTrace();

} catch(Exception e) {

e.printStackTrace();

}

myLabel.setText("Tables loaded!");

}

// Creates a text field with a preferred height of 50px

private TextField newTextField()

{

TextField function = new TextField();

function.setPrefHeight(50);

function.setPrefWidth(300);

return function;

}

// Creates a button with the given text, width, and height

private Button newButton(String text, int width, int height)

{

Button btn = new Button(text);

btn.setPrefWidth(width);

btn.setPrefHeight(height);

return btn;

}

public static void main(String [] args) {

launch(args);

}

}