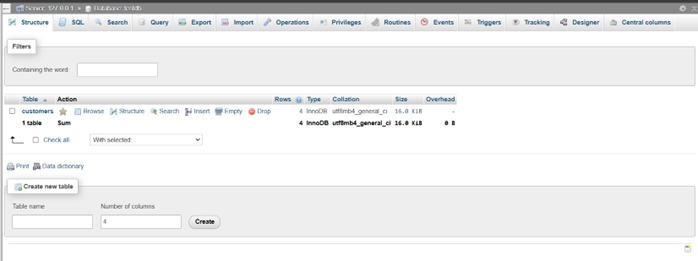
Section-A

A screenshot of a computer

AI-generated content may be incorrect.

This the database I created



THIS IS MY CODE:

import javafx.application.Application;

import javafx.collections.FXCollections;

import javafx.collections.ObservableList;

import javafx.geometry.Insets;

import javafx.scene.Scene;

import javafx.scene.control.\*;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.layout.GridPane;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;

import java.sql.\*;

public class Main extends Application {

private TableView<Customer> table = new TableView<>();

private ObservableList<Customer> data = FXCollections.observableArrayList();

private TextField idField = new TextField();

private TextField nameField = new TextField();

private TextField emailField = new TextField();

private final String DB\_URL = "jdbc:mysql://localhost:3306/testdb";

private final String DB\_USER = "root";

private final String DB\_PASSWORD = "";

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage primaryStage) {

primaryStage.setTitle("Customer Management - Ved Patel - ID: 23094884 - 2025-07-02");

TableColumn<Customer, Integer> idCol = new TableColumn<>("ID");

idCol.setCellValueFactory(new PropertyValueFactory<>("id"));

TableColumn<Customer, String> nameCol = new TableColumn<>("Name");

nameCol.setCellValueFactory(new PropertyValueFactory<>("name"));

TableColumn<Customer, String> emailCol = new TableColumn<>("Email");

emailCol.setCellValueFactory(new PropertyValueFactory<>("email"));

table.getColumns().addAll(idCol, nameCol, emailCol);

GridPane form = new GridPane();

form.setPadding(new Insets(10));

form.setVgap(10);

form.setHgap(10);

form.add(new Label("ID:"), 0, 0);

form.add(idField, 1, 0);

form.add(new Label("Name:"), 0, 1);

form.add(nameField, 1, 1);

form.add(new Label("Email:"), 0, 2);

form.add(emailField, 1, 2);

Button viewBtn = new Button("View Data");

viewBtn.setOnAction(e -> loadData());

Button insertBtn = new Button("Insert");

insertBtn.setOnAction(e -> insertData());

Button updateBtn = new Button("Update");

updateBtn.setOnAction(e -> updateData());

Button deleteBtn = new Button("Delete");

deleteBtn.setOnAction(e -> deleteData());

form.add(viewBtn, 0, 3);

form.add(insertBtn, 1, 3);

form.add(updateBtn, 0, 4);

form.add(deleteBtn, 1, 4);

VBox root = new VBox(10, table, form);

root.setPadding(new Insets(10));

Scene scene = new Scene(root, 600, 500);

primaryStage.setScene(scene);

primaryStage.show();

}

private Connection connect() throws SQLException {

return DriverManager.getConnection(DB\_URL, DB\_USER, DB\_PASSWORD);

}

private void loadData() {

data.clear();

try (Connection conn = connect()) {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery("SELECT \* FROM customers");

while (rs.next()) {

data.add(new Customer(rs.getInt("id"), rs.getString("name"), rs.getString("email")));

}

table.setItems(data);

} catch (SQLException e) {

showAlert("Error", e.getMessage());

}

}

private void insertData() {

String name = nameField.getText();

String email = emailField.getText();

try (Connection conn = connect()) {

PreparedStatement stmt = conn.prepareStatement("INSERT INTO customers (name, email) VALUES (?, ?)");

stmt.setString(1, name);

stmt.setString(2, email);

stmt.executeUpdate();

loadData();

} catch (SQLException e) {

showAlert("Error", e.getMessage());

}

}

private void updateData() {

int id = Integer.parseInt(idField.getText());

String name = nameField.getText();

String email = emailField.getText();

try (Connection conn = connect()) {

PreparedStatement stmt = conn.prepareStatement("UPDATE customers SET name = ?, email = ? WHERE id = ?");

stmt.setString(1, name);

stmt.setString(2, email);

stmt.setInt(3, id);

stmt.executeUpdate();

loadData();

} catch (SQLException e) {

showAlert("Error", e.getMessage());

}

}

private void deleteData() {

int id = Integer.parseInt(idField.getText());

try (Connection conn = connect()) {

PreparedStatement stmt = conn.prepareStatement("DELETE FROM customers WHERE id = ?");

stmt.setInt(1, id);

stmt.executeUpdate();

loadData();

} catch (SQLException e) {

showAlert("Error", e.getMessage());

}

}

private void showAlert(String title, String message) {

Alert alert = new Alert(Alert.AlertType.ERROR);

alert.setTitle(title);

alert.setContentText(message);

alert.showAndWait();

}

public static class Customer {

private int id;

private String name;

private String email;

public Customer(int id, String name, String email) {

this.id = id;

this.name = name;

this.email = email;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

}

}