**Easy Problem:-**

import java.sql.\*;

public class Main {

public static void main(String[] args) {

// Database credentials

String url = "jdbc:mysql://localhost:3306/your\_database";

String user = "your\_username";

String password = "your\_password";

// SQL query to retrieve data

String query = "SELECT \* FROM Employee";

// Load and register JDBC driver (optional for newer Java versions)

try {

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

} catch (ClassNotFoundException e) {

System.out.println("MySQL JDBC Driver not found.");

return;

}

// Database connection and data retrieval

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

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query)) {

// Display results

System.out.println("EmpID\tName\tSalary");

System.out.println("---------------------------");

while (rs.next()) {

int empID = rs.getInt("EmpID");

String name = rs.getString("Name");

double salary = rs.getDouble("Salary");

System.out.println(empID + "\t" + name + "\t" + salary);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}

**Medium level:-**

import java.sql.\*;

import java.util.Scanner;

public class Main {

// Database credentials

private static final String URL = "jdbc:mysql://localhost:3306/your\_database";

private static final String USER = "your\_username";

private static final String PASSWORD = "your\_password";

public static void main(String[] args) {

try {

Class.forName("com.mysql.cj.jdbc.Driver"); // Load MySQL driver

Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);

conn.setAutoCommit(false); // Enable transaction handling

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("\nProduct Management System");

System.out.println("1. Add Product");

System.out.println("2. View Products");

System.out.println("3. Update Product");

System.out.println("4. Delete Product");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

addProduct(conn, scanner);

break;

case 2:

viewProducts(conn);

break;

case 3:

updateProduct(conn, scanner);

break;

case 4:

deleteProduct(conn, scanner);

break;

case 5:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Try again.");

}

} while (choice != 5);

conn.close();

scanner.close();

} catch (Exception e) {

e.printStackTrace();

}

}

// Method to add a new product

private static void addProduct(Connection conn, Scanner scanner) {

try {

System.out.print("Enter Product Name: ");

scanner.nextLine(); // Consume newline

String name = scanner.nextLine();

System.out.print("Enter Price: ");

double price = scanner.nextDouble();

System.out.print("Enter Quantity: ");

int quantity = scanner.nextInt();

String query = "INSERT INTO Product (ProductName, Price, Quantity) VALUES (?, ?, ?)";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setString(1, name);

pstmt.setDouble(2, price);

pstmt.setInt(3, quantity);

int rows = pstmt.executeUpdate();

conn.commit();

System.out.println(rows + " product(s) added successfully.");

} catch (SQLException e) {

try { conn.rollback(); } catch (SQLException ex) { ex.printStackTrace(); }

e.printStackTrace();

}

}

// Method to view all products

private static void viewProducts(Connection conn) {

try {

String query = "SELECT \* FROM Product";

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(query);

System.out.println("\nProduct List:");

System.out.println("ID\tName\t\tPrice\tQuantity");

System.out.println("--------------------------------------");

while (rs.next()) {

int id = rs.getInt("ProductID");

String name = rs.getString("ProductName");

double price = rs.getDouble("Price");

int quantity = rs.getInt("Quantity");

System.out.println(id + "\t" + name + "\t\t" + price + "\t" + quantity);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

// Method to update a product

private static void updateProduct(Connection conn, Scanner scanner) {

try {

System.out.print("Enter Product ID to update: ");

int id = scanner.nextInt();

System.out.print("Enter New Price: ");

double price = scanner.nextDouble();

System.out.print("Enter New Quantity: ");

int quantity = scanner.nextInt();

String query = "UPDATE Product SET Price = ?, Quantity = ? WHERE ProductID = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setDouble(1, price);

pstmt.setInt(2, quantity);

pstmt.setInt(3, id);

int rows = pstmt.executeUpdate();

conn.commit();

System.out.println(rows + " product(s) updated successfully.");

} catch (SQLException e) {

try { conn.rollback(); } catch (SQLException ex) { ex.printStackTrace(); }

e.printStackTrace();

}

}

// Method to delete a product

private static void deleteProduct(Connection conn, Scanner scanner) {

try {

System.out.print("Enter Product ID to delete: ");

int id = scanner.nextInt();

String query = "DELETE FROM Product WHERE ProductID = ?";

PreparedStatement pstmt = conn.prepareStatement(query);

pstmt.setInt(1, id);

int rows = pstmt.executeUpdate();

conn.commit();

System.out.println(rows + " product(s) deleted successfully.");

} catch (SQLException e) {

try { conn.rollback(); } catch (SQLException ex) { ex.printStackTrace(); }

e.printStackTrace();

}

}

}

**Hard Problem:-**

import java.sql.\*;

import java.util.\*;

class Student {

private int studentID;

private String name;

private String department;

private double marks;

public Student(int studentID, String name, String department, double marks) {

this.studentID = studentID;

this.name = name;

this.department = department;

this.marks = marks;

}

public int getStudentID() { return studentID; }

public String getName() { return name; }

public String getDepartment() { return department; }

public double getMarks() { return marks; }

@Override

public String toString() {

return "ID: " + studentID + ", Name: " + name + ", Department: " + department + ", Marks: " + marks;

}

}

class StudentController {

private static final String URL = "jdbc:mysql://localhost:3306/your\_database";

private static final String USER = "your\_username";

private static final String PASSWORD = "your\_password";

public Connection connect() throws SQLException {

return DriverManager.getConnection(URL, USER, PASSWORD);

}

public void addStudent(Student student) {

String query = "INSERT INTO Student (StudentID, Name, Department, Marks) VALUES (?, ?, ?, ?)";

try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {

pstmt.setInt(1, student.getStudentID());

pstmt.setString(2, student.getName());

pstmt.setString(3, student.getDepartment());

pstmt.setDouble(4, student.getMarks());

pstmt.executeUpdate();

System.out.println("Student added successfully.");

} catch (SQLException e) {

e.printStackTrace();

}

}

public void viewStudents() {

String query = "SELECT \* FROM Student";

try (Connection conn = connect(); Statement stmt = conn.createStatement(); ResultSet rs = stmt.executeQuery(query)) {

System.out.println("\nStudent List:");

while (rs.next()) {

int id = rs.getInt("StudentID");

String name = rs.getString("Name");

String department = rs.getString("Department");

double marks = rs.getDouble("Marks");

System.out.println("ID: " + id + ", Name: " + name + ", Department: " + department + ", Marks: " + marks);

}

} catch (SQLException e) {

e.printStackTrace();

}

}

public void updateStudent(int id, double marks) {

String query = "UPDATE Student SET Marks = ? WHERE StudentID = ?";

try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {

pstmt.setDouble(1, marks);

pstmt.setInt(2, id);

int rows = pstmt.executeUpdate();

System.out.println(rows + " student(s) updated.");

} catch (SQLException e) {

e.printStackTrace();

}

}

public void deleteStudent(int id) {

String query = "DELETE FROM Student WHERE StudentID = ?";

try (Connection conn = connect(); PreparedStatement pstmt = conn.prepareStatement(query)) {

pstmt.setInt(1, id);

int rows = pstmt.executeUpdate();

System.out.println(rows + " student(s) deleted.");

} catch (SQLException e) {

e.printStackTrace();

}

}

}

public class Main {

public static void main(String[] args) {

try {

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

} catch (ClassNotFoundException e) {

System.out.println("MySQL JDBC Driver not found.");

return;

}

Scanner scanner = new Scanner(System.in);

StudentController controller = new StudentController();

int choice;

do {

System.out.println("\nStudent Management System");

System.out.println("1. Add Student");

System.out.println("2. View Students");

System.out.println("3. Update Student Marks");

System.out.println("4. Delete Student");

System.out.println("5. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

System.out.print("Enter Student ID: ");

int id = scanner.nextInt();

scanner.nextLine();

System.out.print("Enter Name: ");

String name = scanner.nextLine();

System.out.print("Enter Department: ");

String department = scanner.nextLine();

System.out.print("Enter Marks: ");

double marks = scanner.nextDouble();

controller.addStudent(new Student(id, name, department, marks));

break;

case 2:

controller.viewStudents();

break;

case 3:

System.out.print("Enter Student ID to update: ");

int updateID = scanner.nextInt();

System.out.print("Enter new Marks: ");

double newMarks = scanner.nextDouble();

controller.updateStudent(updateID, newMarks);

break;

case 4:

System.out.print("Enter Student ID to delete: ");

int deleteID = scanner.nextInt();

controller.deleteStudent(deleteID);

break;

case 5:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Try again.");

}

} while (choice != 5);

scanner.close();

}

}