**Assessment 2**

**SQL Database Testing**

Observation: XYZ is healthcare information system which contains sensitive patient records and must adhere to strict data integrity and security standards.

Test Scenarios:

Example 1: Data integrity and Security

Authorized users can access data through the credentials provided by database admin. Admin have all the permissions to grant are revoke permissions to the users.

Below mentioned sql queries are the examples for sample patient records.

Example 2:

Below mentioned java script are the examples for data migration scripts.

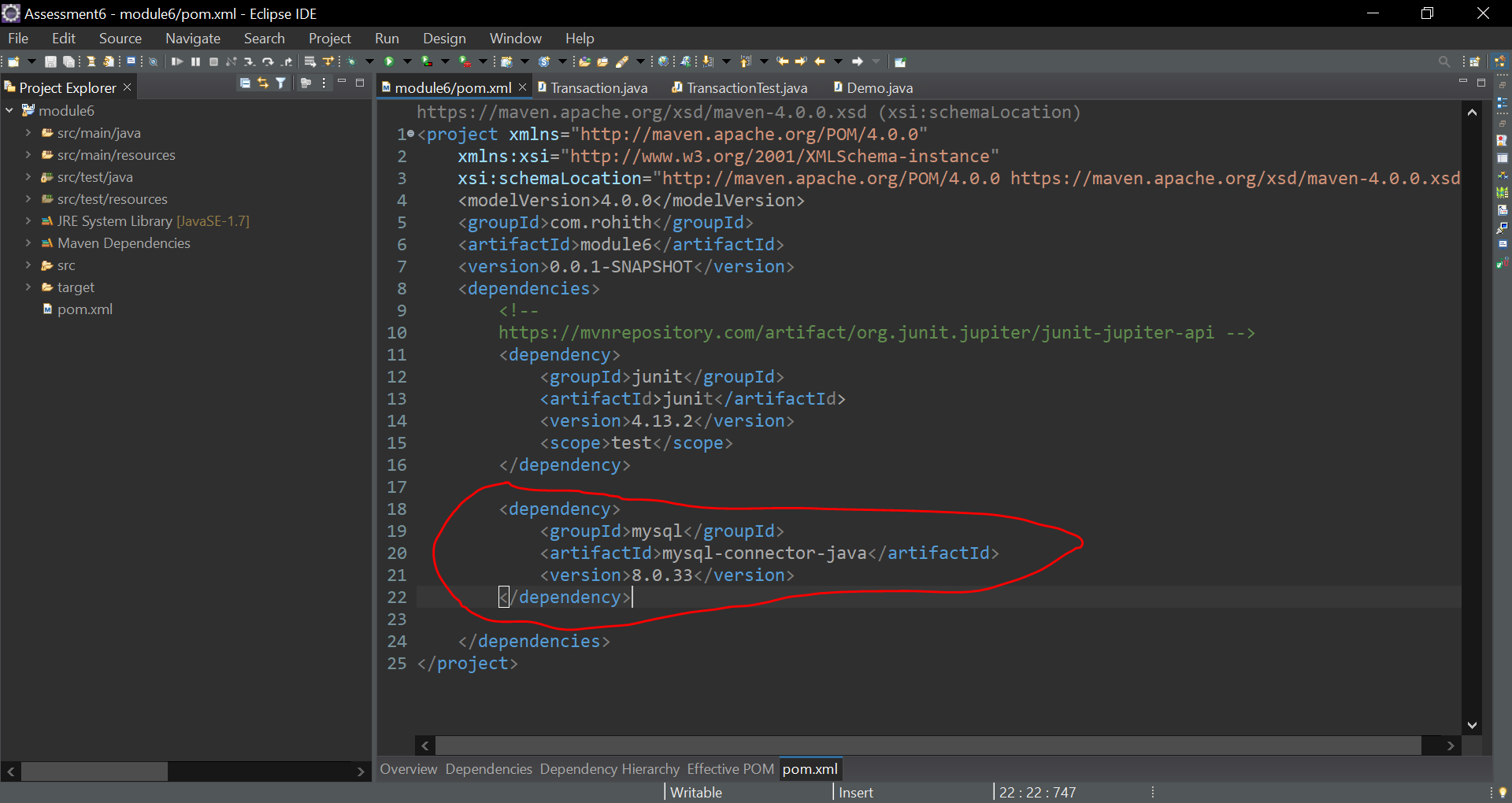
We can check database response times by performing performance tuning.

Requirements: Java, Eclipse, MySQL, Workbench

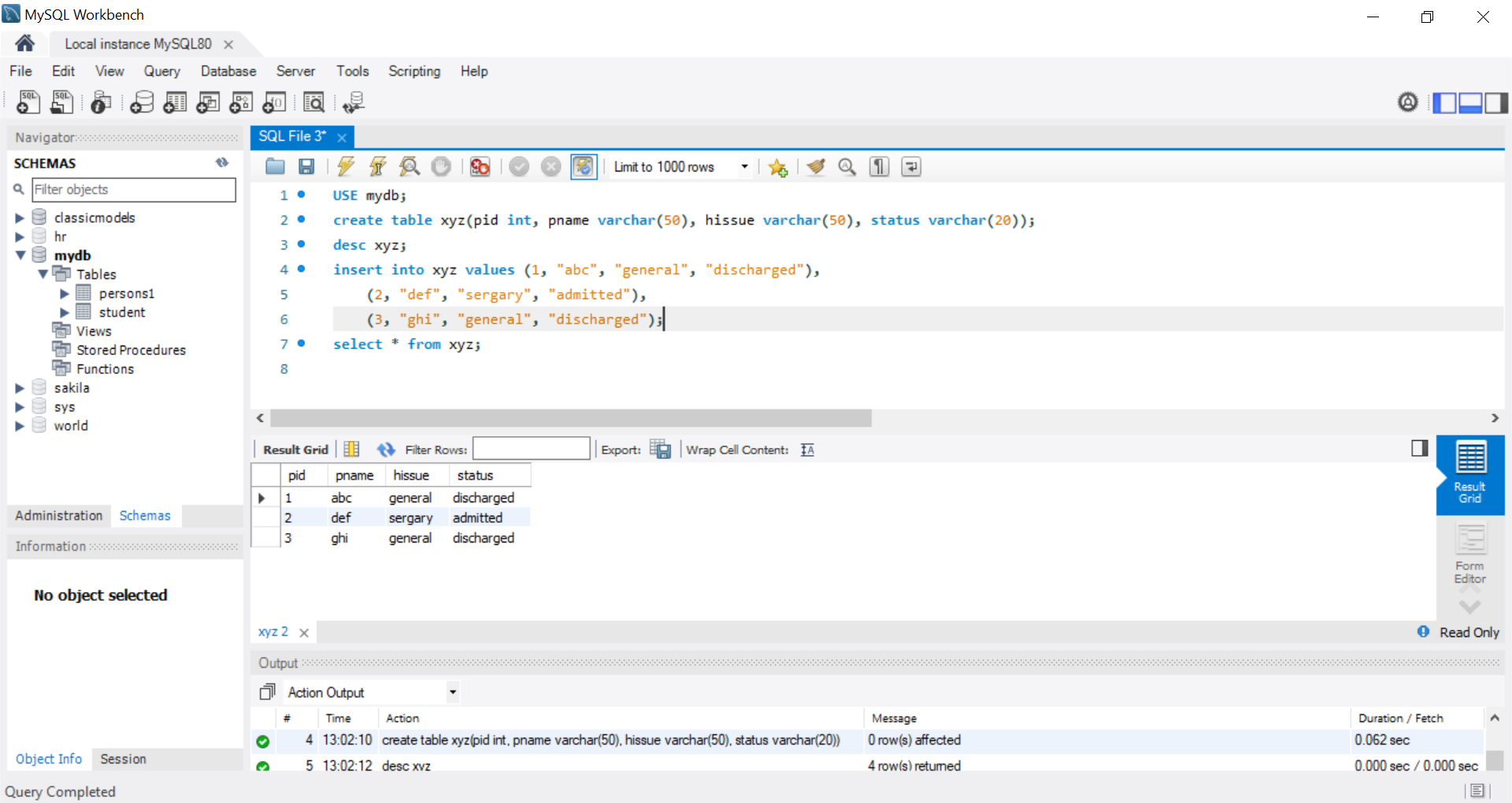
Project Setup:

1. Launch Eclipse IDE and create basic Maven project

2. Add mysql dependencies to interact with mysql databased through java.



Loading sample data into mysql database.



Java script to interact with MySql DB

package com.xyz;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class Data {

public static void main(String[] args) throws SQLException {

// **TODO** Auto-generated method stub

Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/mydb","root","root");

Statement stmt = con.createStatement();

String insertQuerry = "insert into xyz values(4, \"yyy\", \"general\", \"admitted\")";

String selecQuerry = "select \* from xyz";

stmt.execute(insertQuerry);

ResultSet rs = stmt.executeQuery(selecQuerry);

while(rs.next()) {

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

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

String issue = rs.getString("hissue");

String status = rs.getString("status");

System.***out***.println(id + " " + name + " " + issue + " " + status);

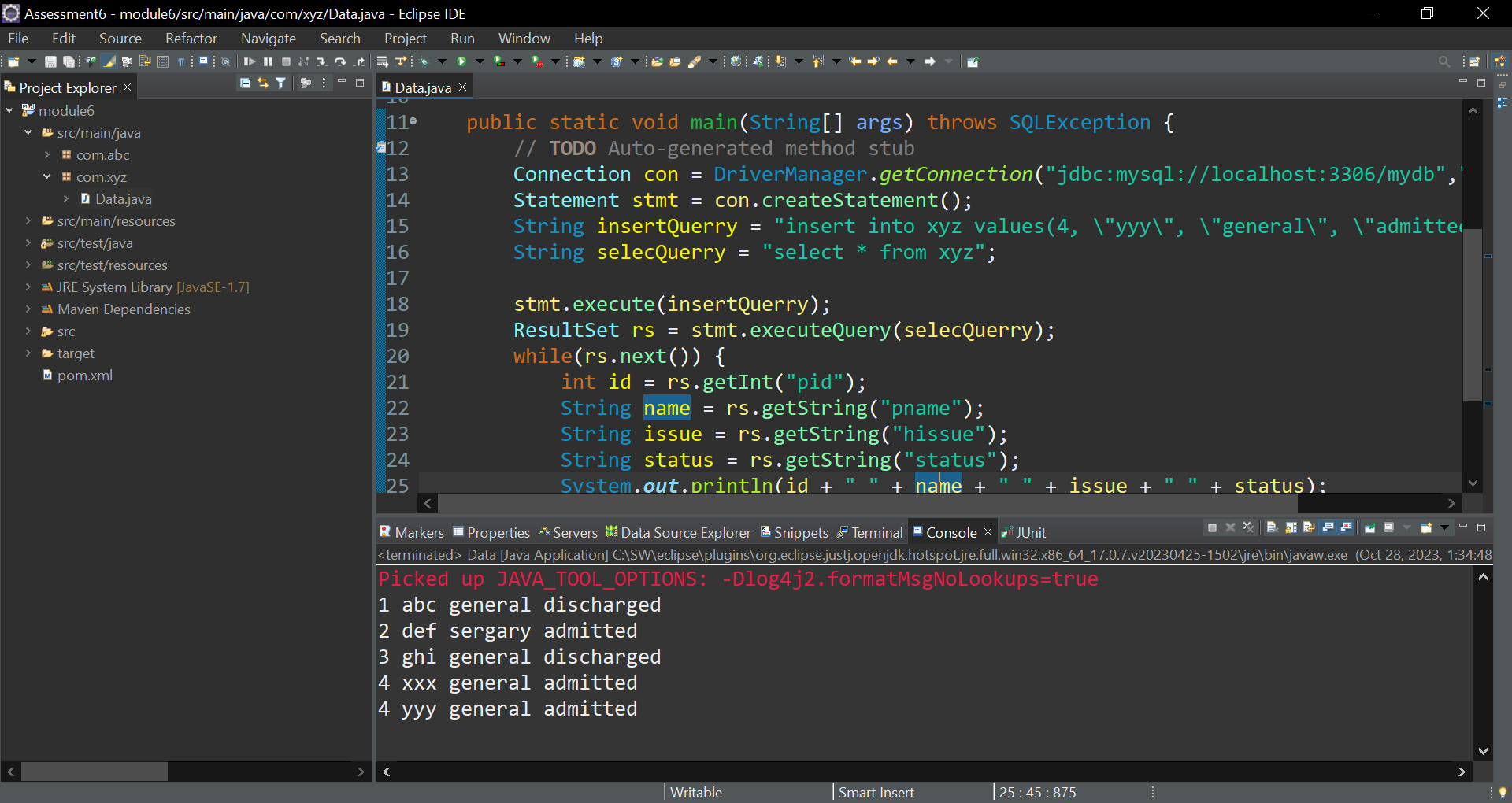
}

con.close();

}

}

Output of the above-mentioned code



**Submitted by:**

**Rohith Reddy**

**vrddam@amazon.com**