題目描述：

寫一支 Java 程式，把 input.xml 中之資料寫入 XMLDB 之 class table.

(Write a Java program that can read information from input.xml and write this information into the class table in XMLDB database.)

rollno 寫入 id 欄位

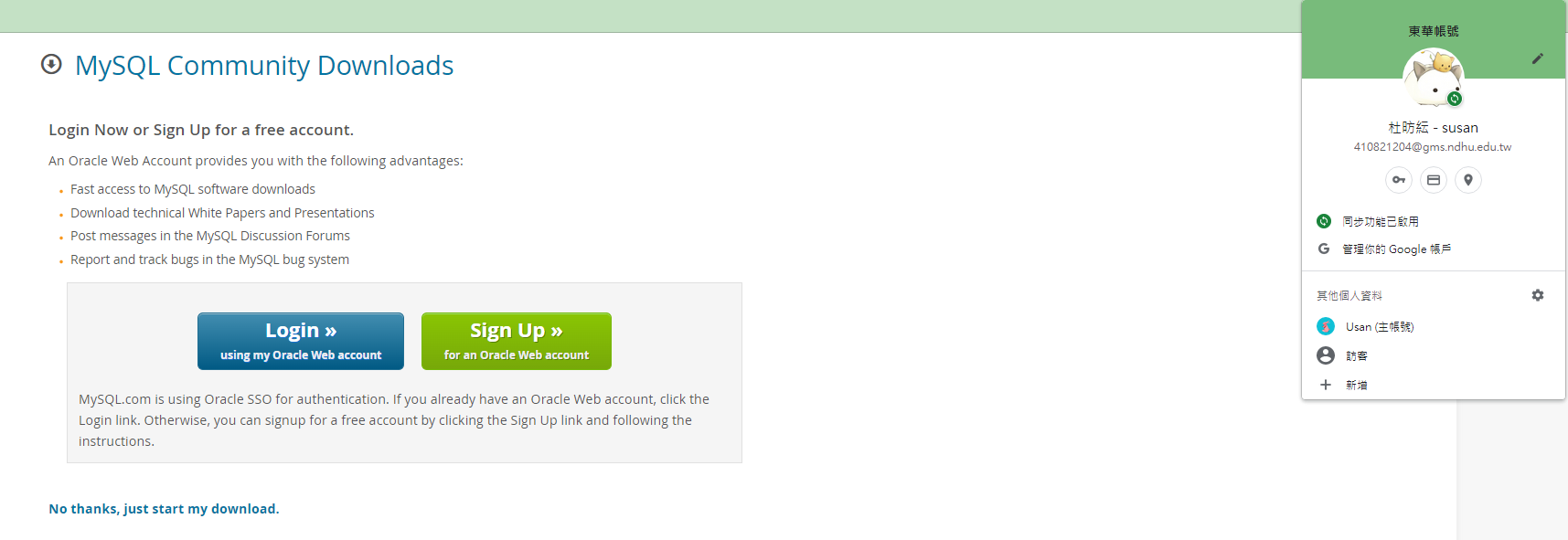
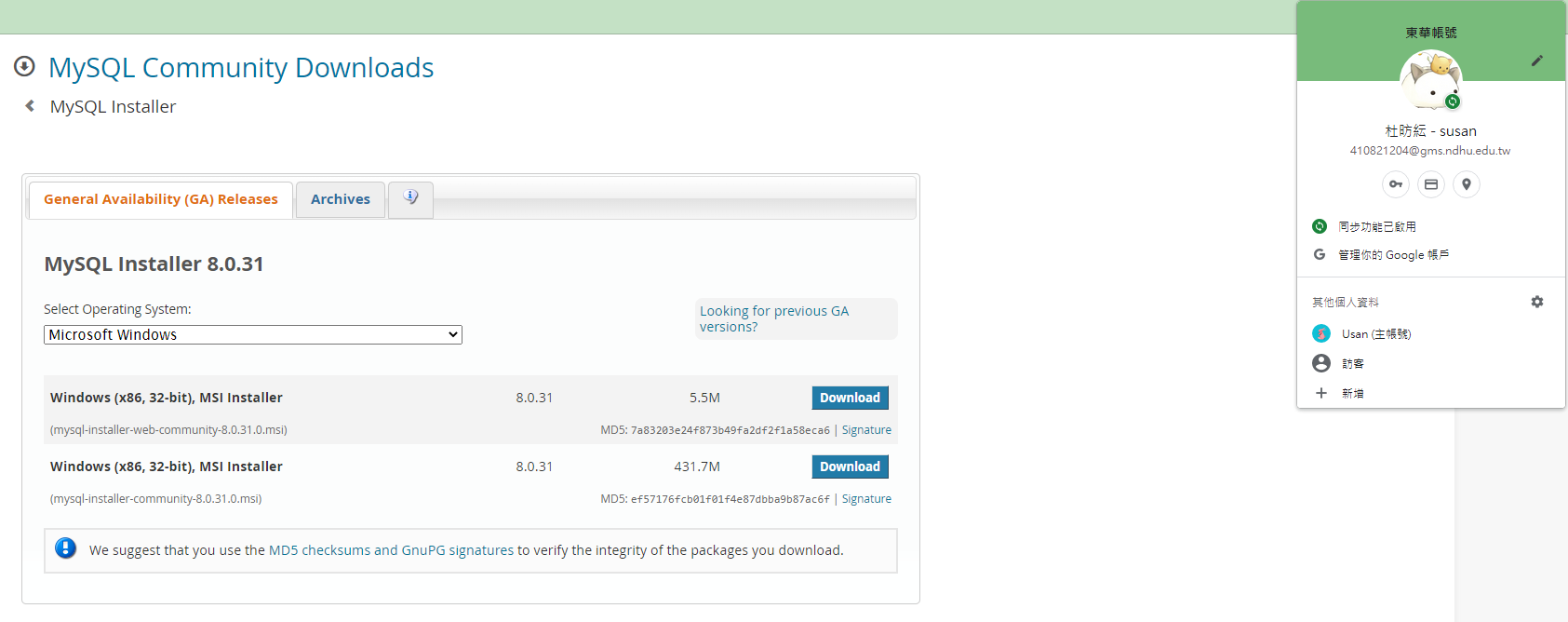
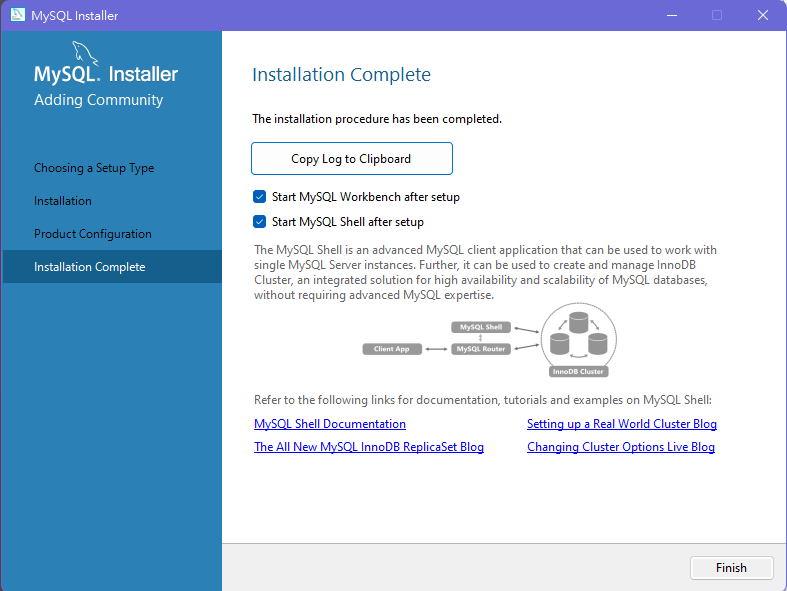
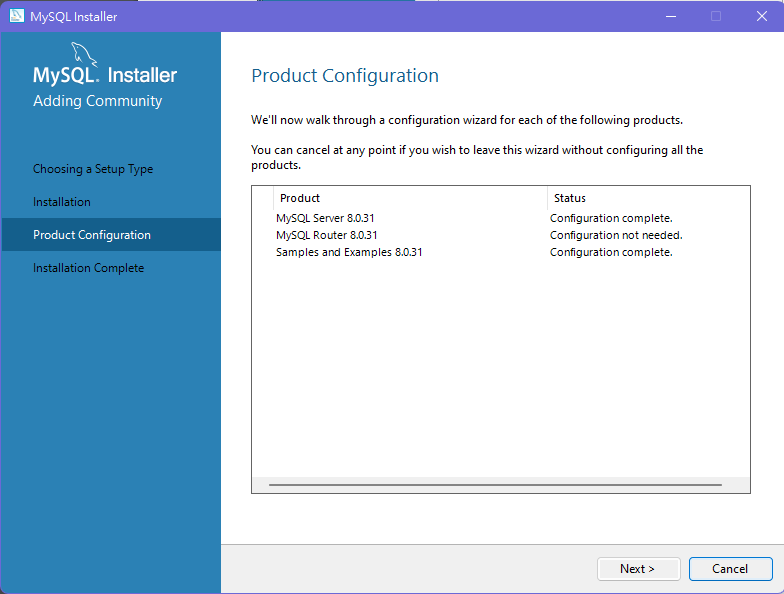
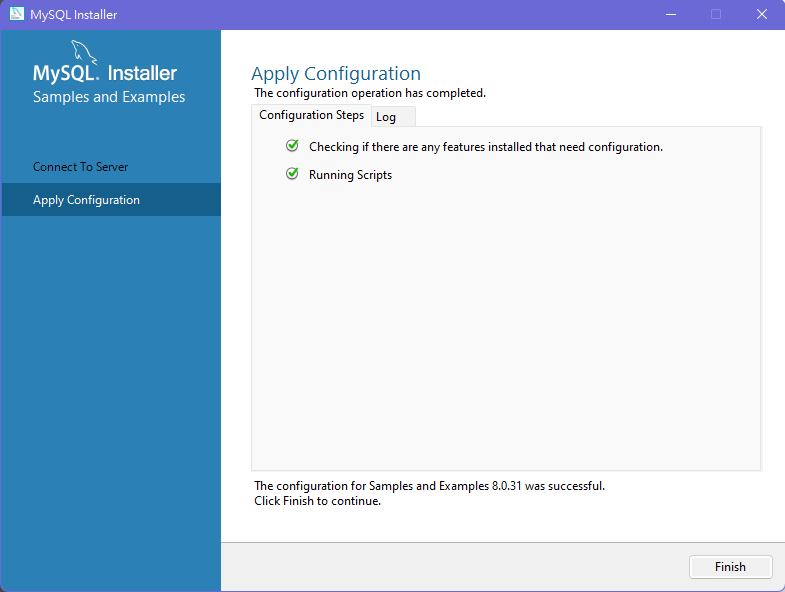
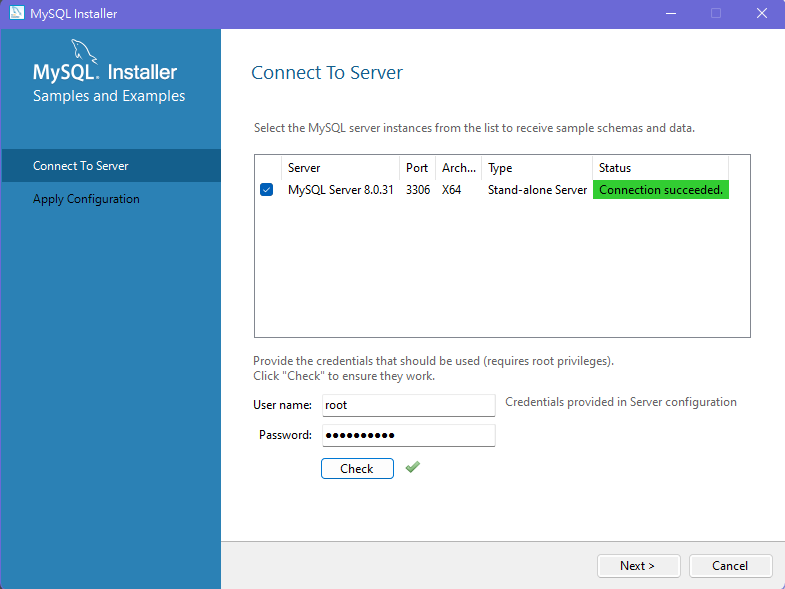
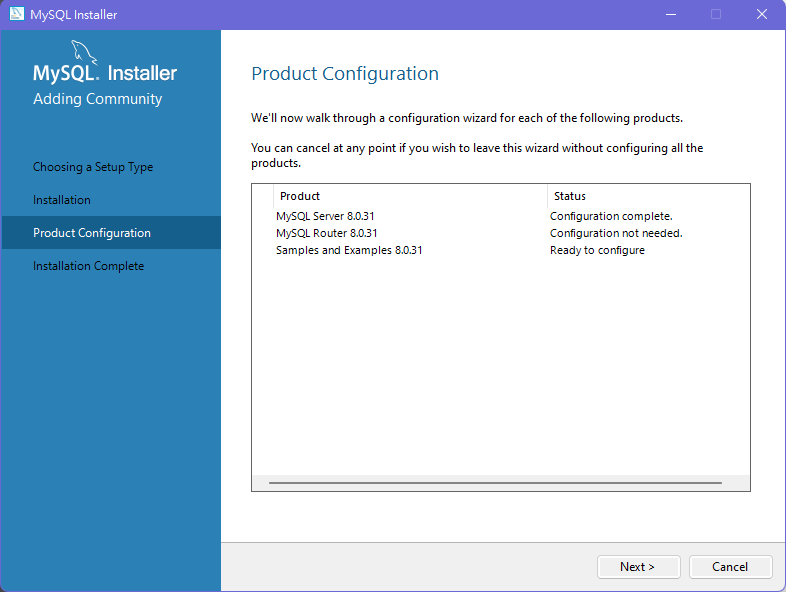
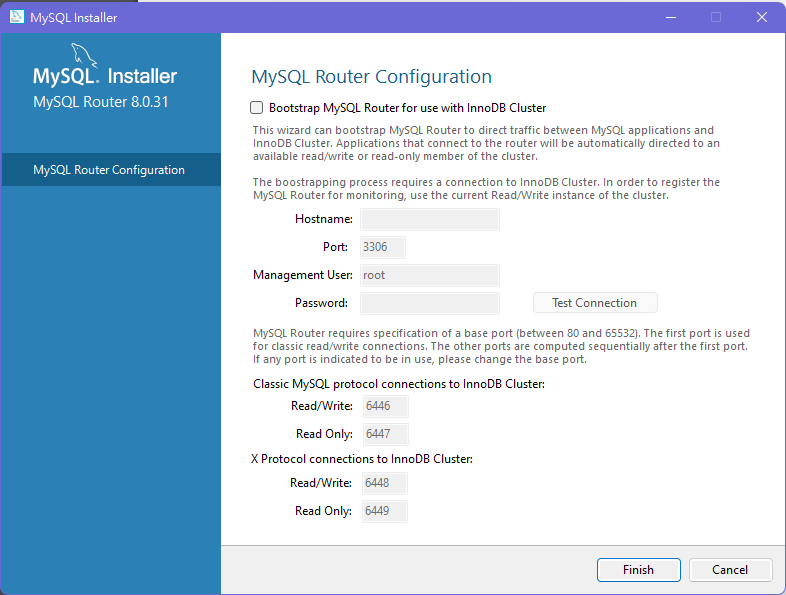
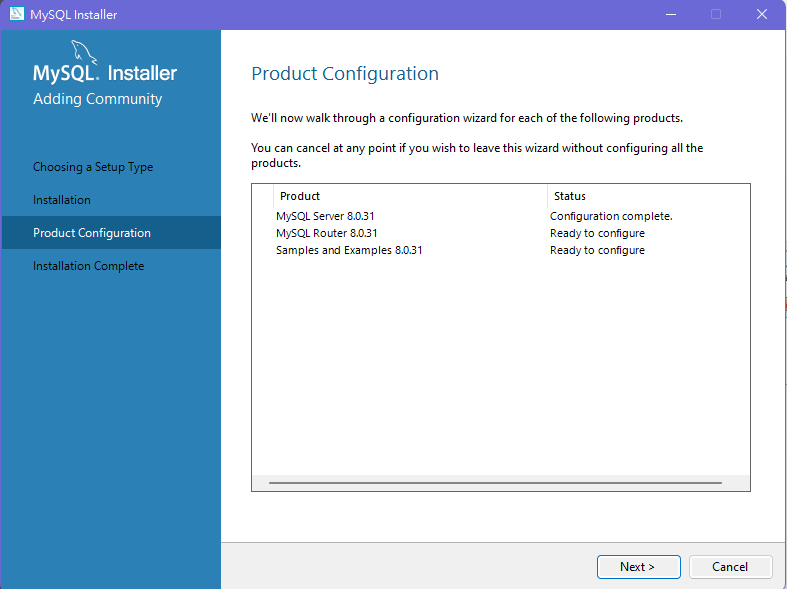
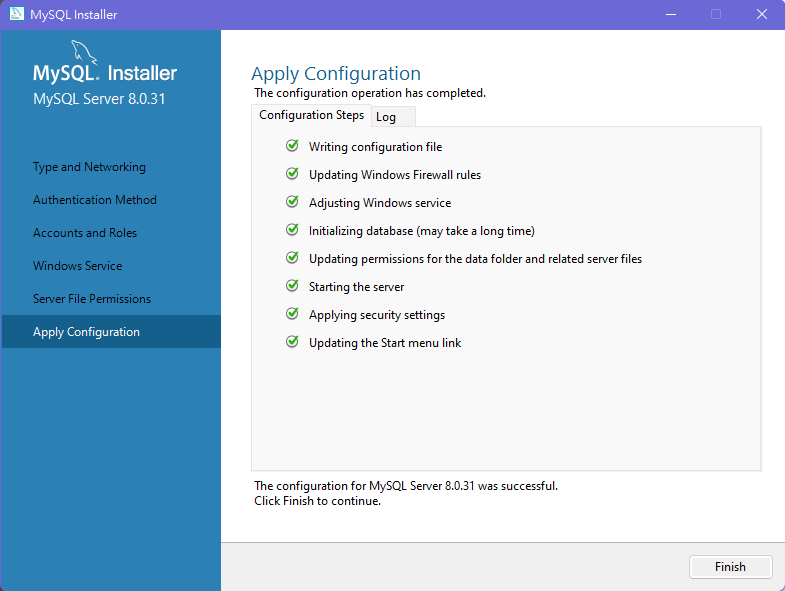
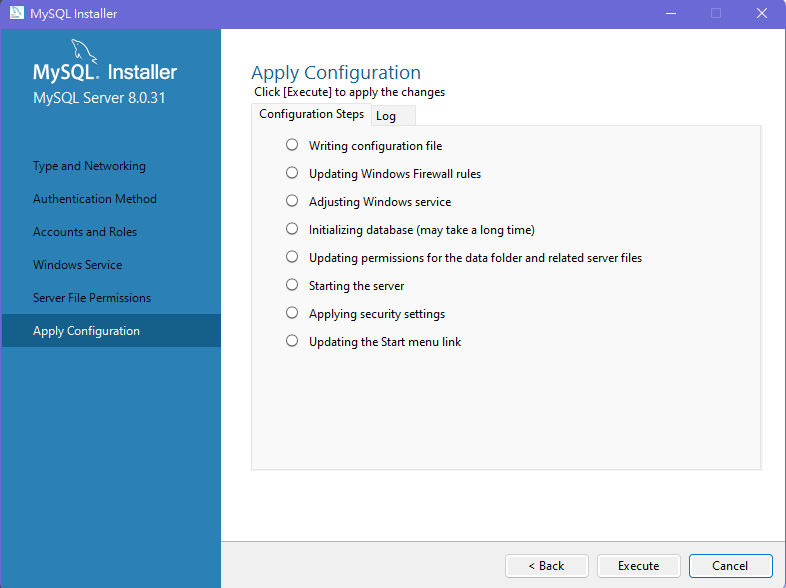
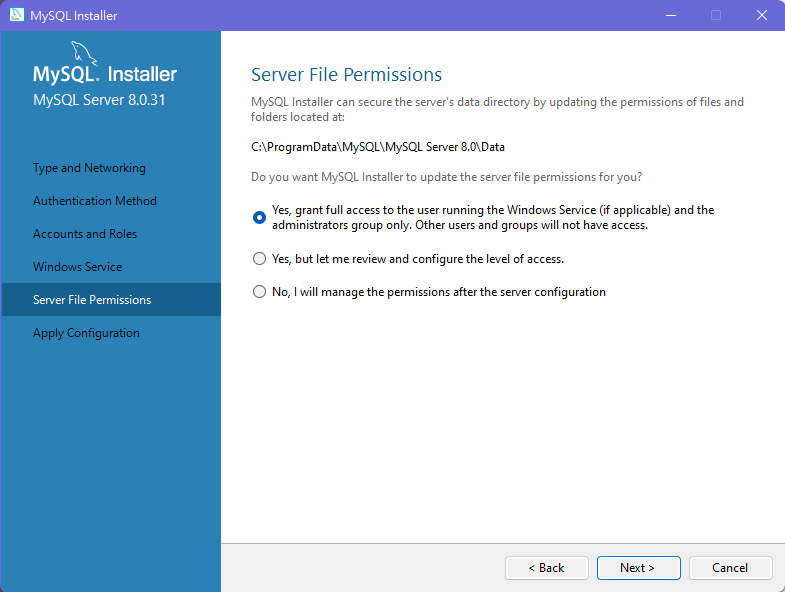
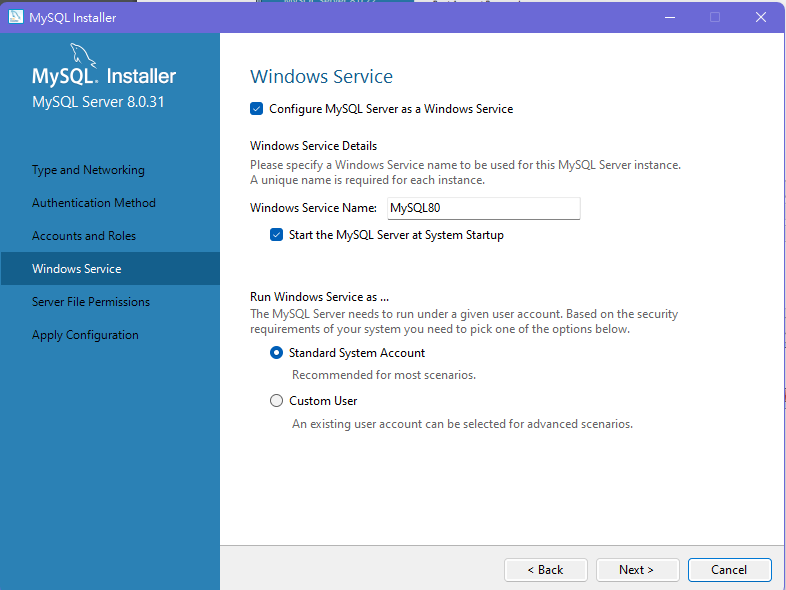
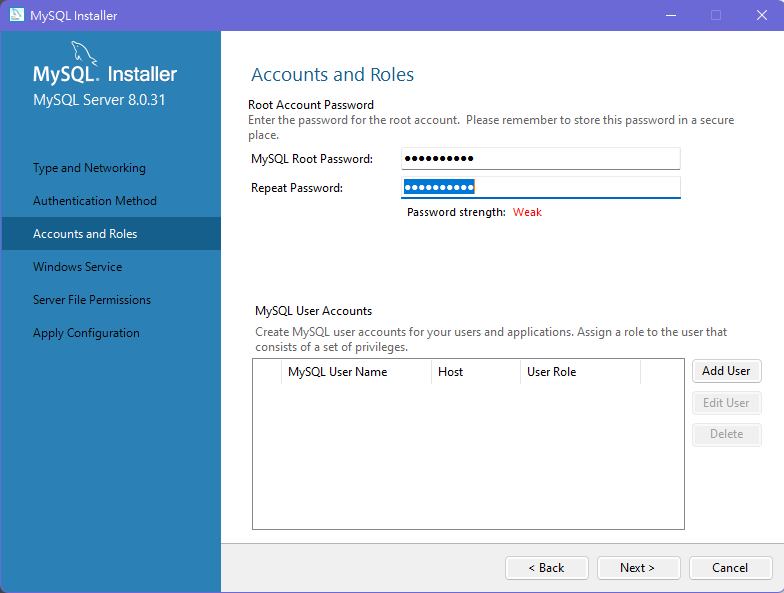
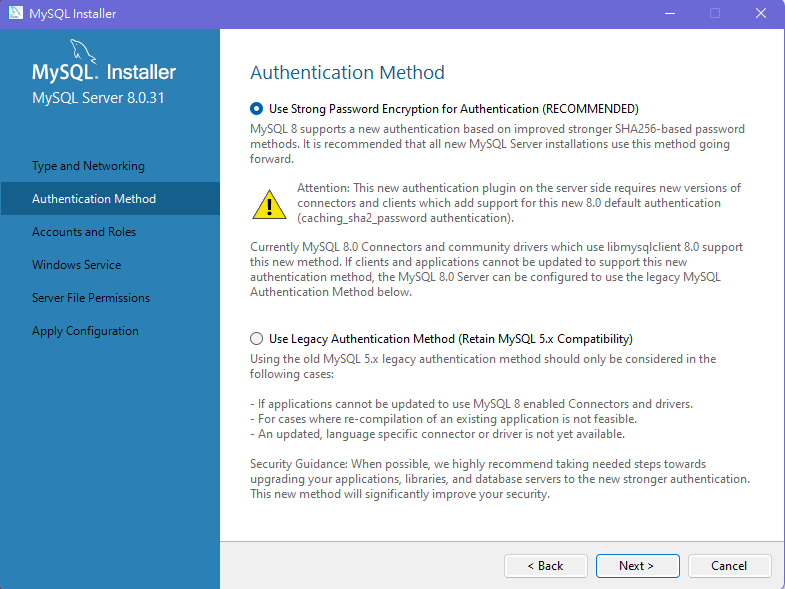
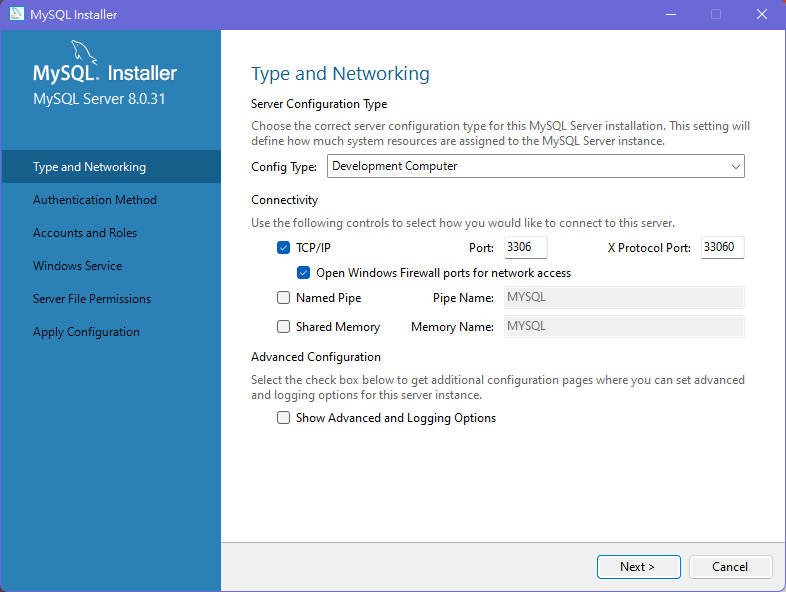
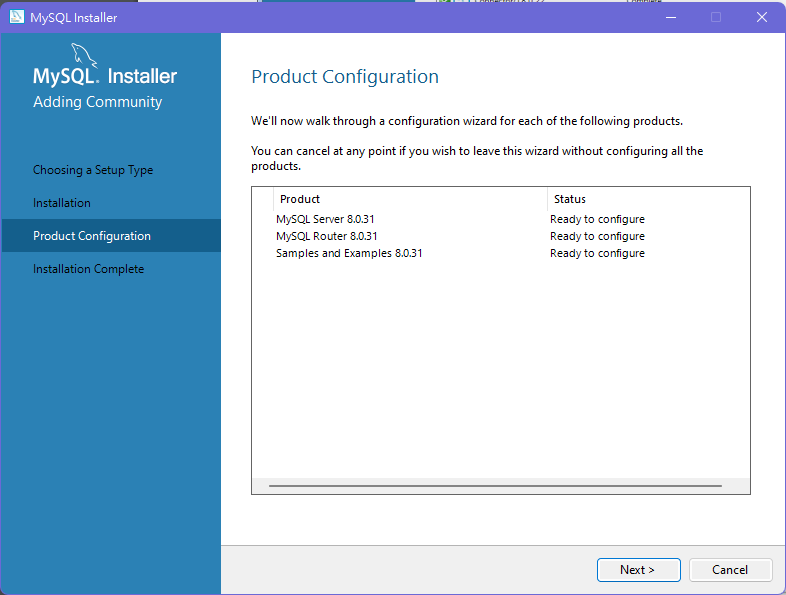
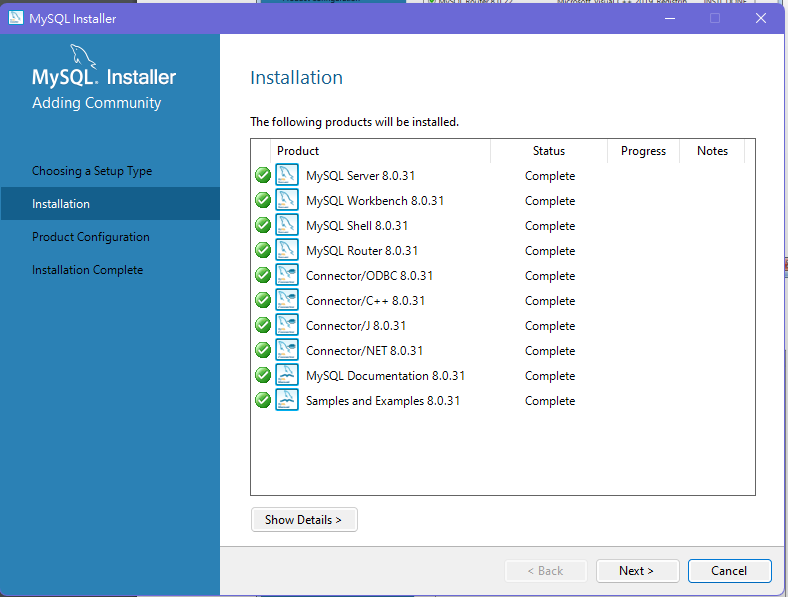
firstname 寫入 first 欄位

lastname 寫入 last 欄位

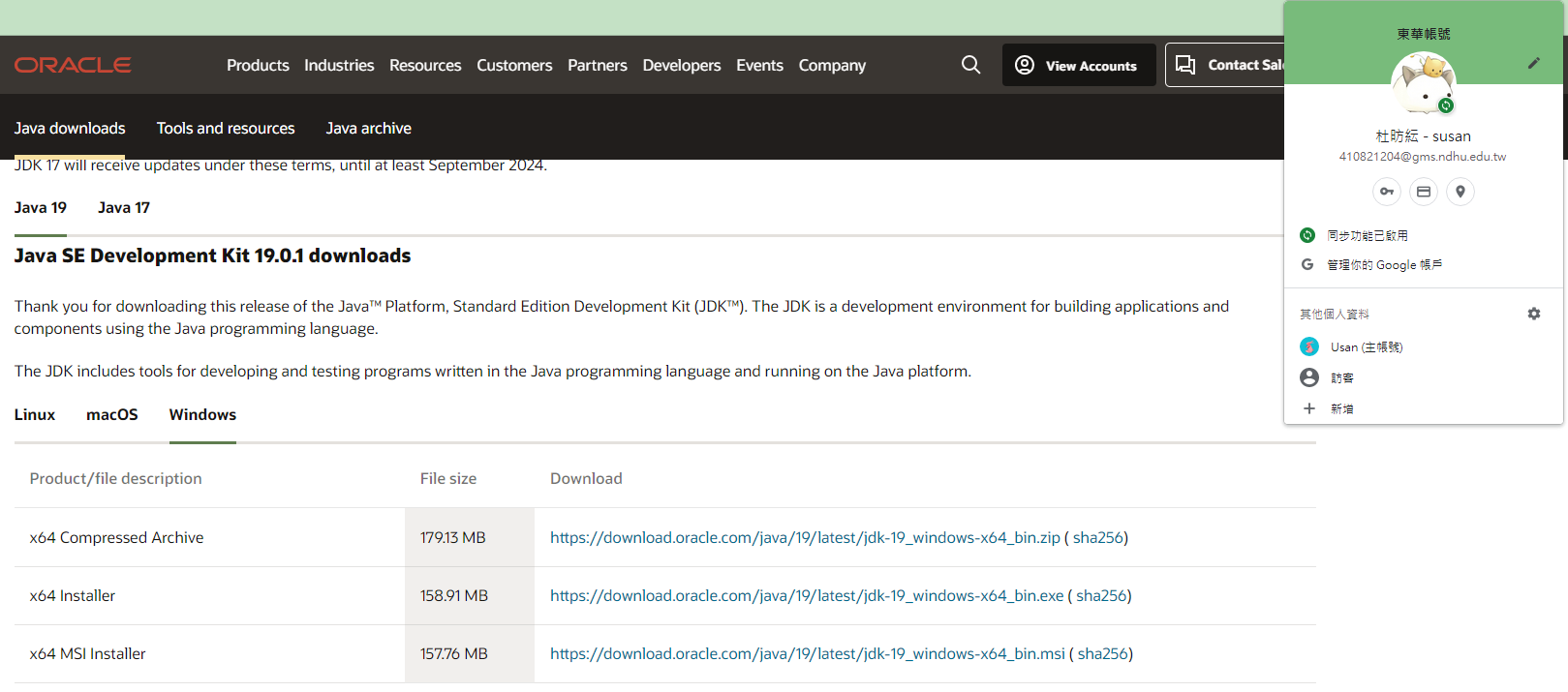
nickname 寫入 nick 欄位

marks 寫入 mark 欄位

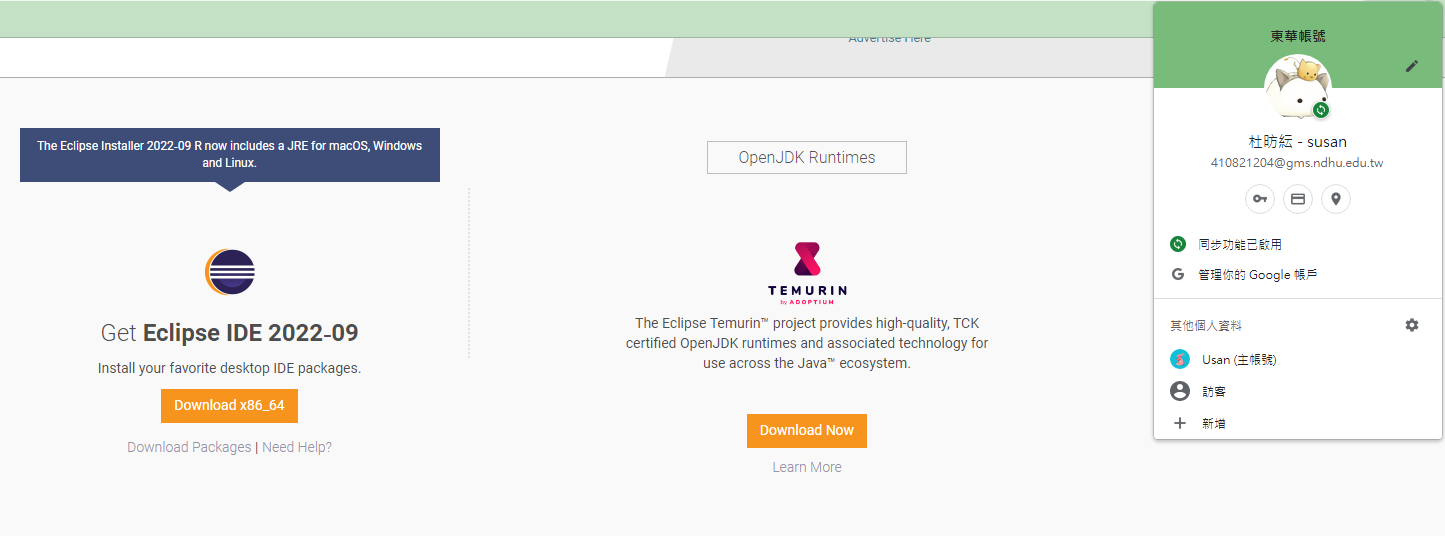
1. **Install MySQL**



1. **Install JDK**

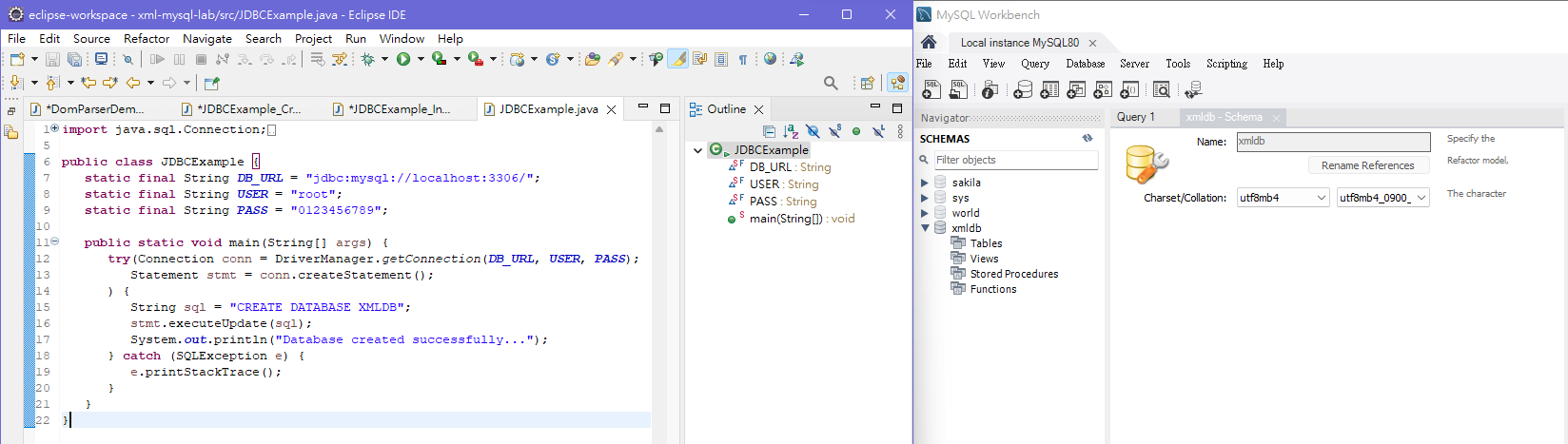


1. **Install Eclipse IDE**



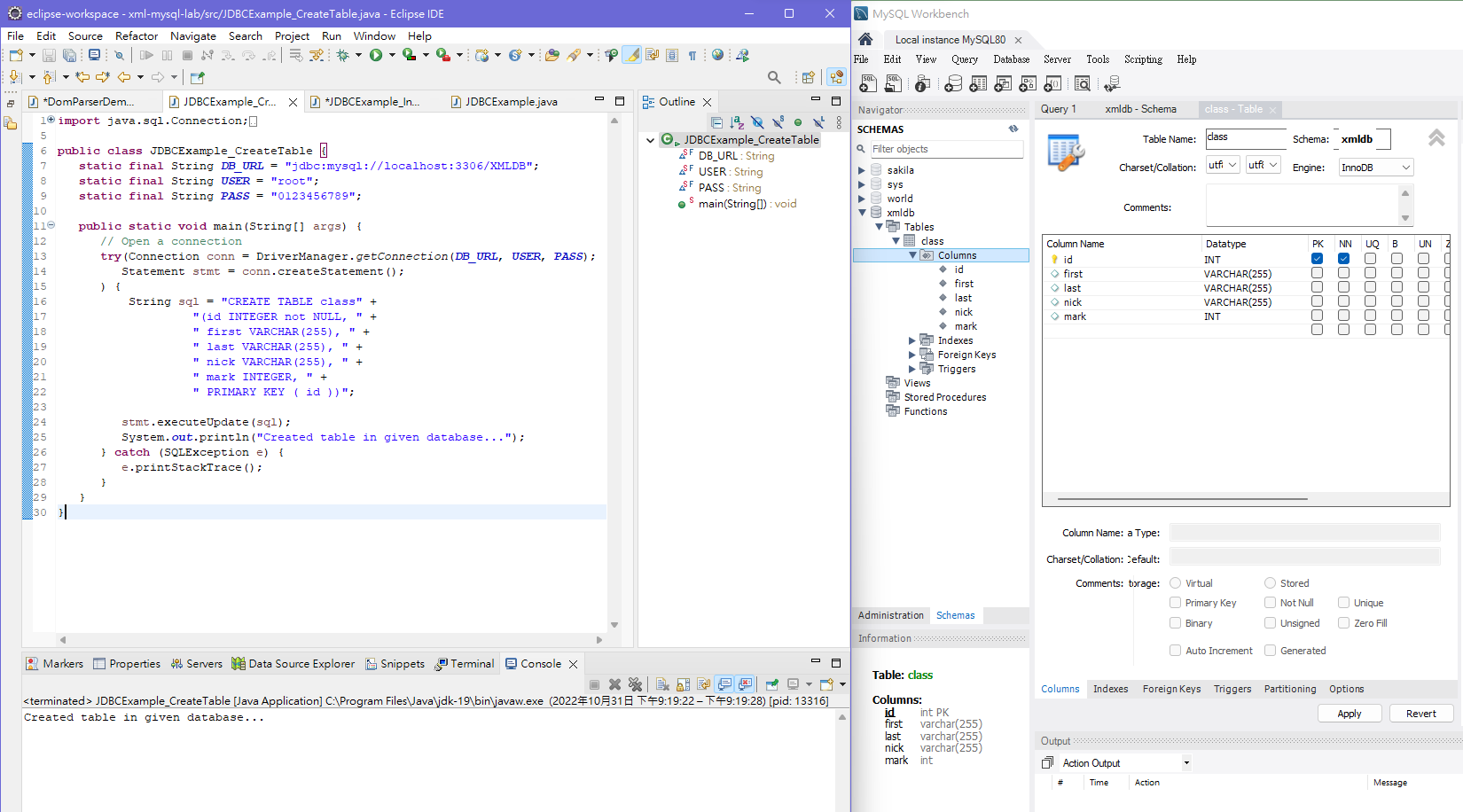
1. **Connect to MySQL database**

Create a database using a Java program. The name of the database is XMLDB.

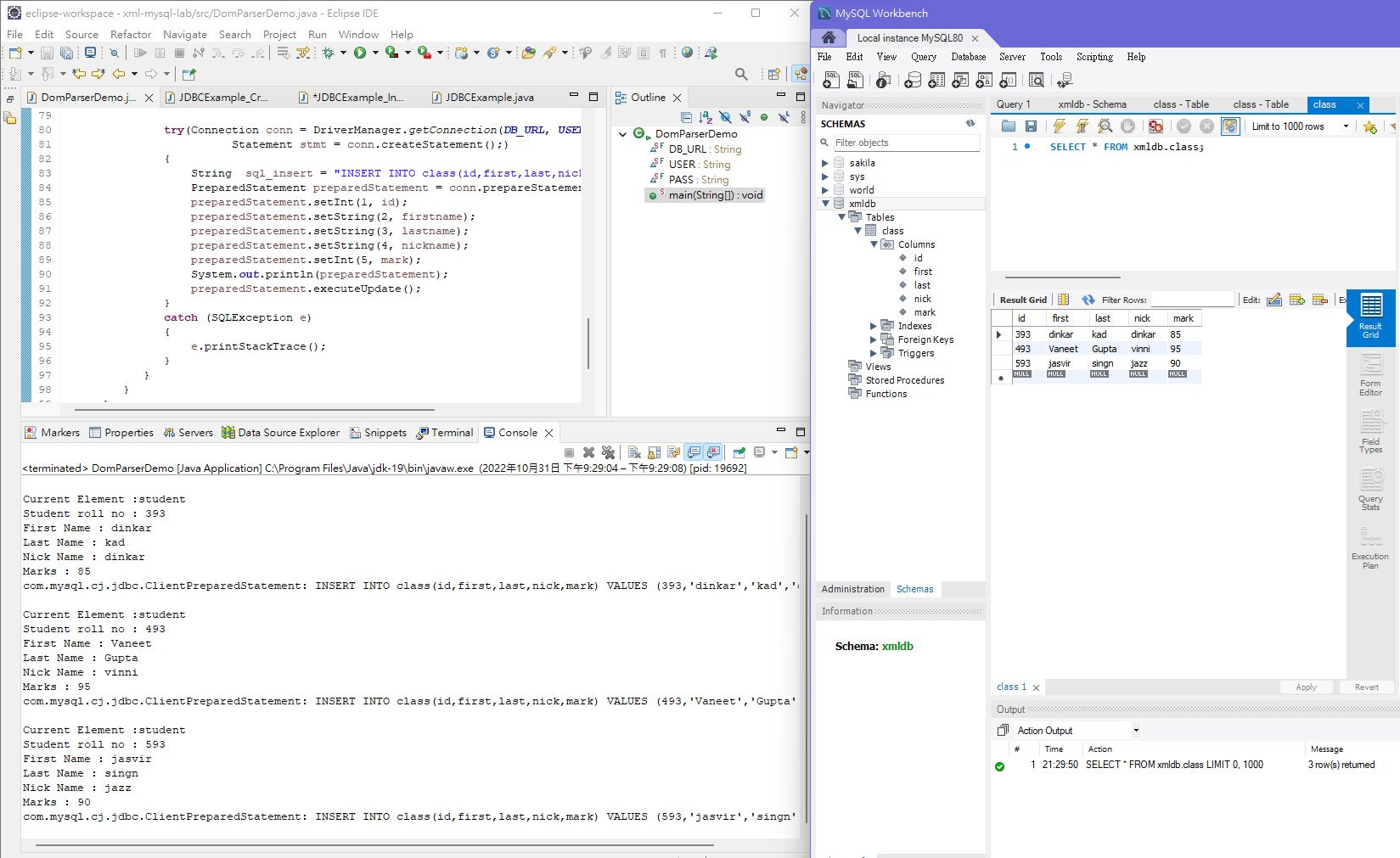


1. **Connect to MySQL database**

Create a table named class in the XMLDB database using a Java program. The class table contains five fields: id, first, last, nick, and mark.



1. **題目實驗程式碼**



|  |
| --- |
| **import** java.io.File;  **import** java.sql.Connection;  **import** java.sql.DriverManager;  **import** java.sql.PreparedStatement;  **import** java.sql.SQLException;  **import** java.sql.Statement;  **import** javax.xml.parsers.DocumentBuilderFactory;  **import** javax.xml.parsers.DocumentBuilder;  **import** org.w3c.dom.Document;  **import** org.w3c.dom.NodeList;  **import** org.w3c.dom.Node;  **import** org.w3c.dom.Element;  **public** **class** DomParserDemo  {  **static** **final** String ***DB\_URL*** = "jdbc:mysql://localhost/XMLDB";  **static** **final** String ***USER*** = "root";  **static** **final** String ***PASS*** = "0123456789";  **public** **static** **void** main(String[] args)  {  **try**  {  Integer id,mark;  String firstname,lastname,nickname;    File inputFile = **new** File("input.xml");  DocumentBuilderFactory dbFactory = DocumentBuilderFactory.*newInstance*();  DocumentBuilder dBuilder = dbFactory.newDocumentBuilder();  Document doc = dBuilder.parse(inputFile);  doc.getDocumentElement().normalize();  System.***out***.println("Root element :" + doc.getDocumentElement().getNodeName());  NodeList nList = doc.getElementsByTagName("student");  System.***out***.println("----------------------------");      **for** (**int** temp = 0; temp < nList.getLength(); temp++)  {  Node nNode = nList.item(temp);  System.***out***.println("\nCurrent Element :" + nNode.getNodeName());    **if** (nNode.getNodeType() == Node.***ELEMENT\_NODE***)  {  Element eElement = (Element) nNode;    System.***out***.println("Student roll no : "  + eElement.getAttribute("rollno"));  id = Integer.*parseInt*(eElement.getAttribute("rollno"));    System.***out***.println("First Name : "  + eElement  .getElementsByTagName("firstname")  .item(0)  .getTextContent());  firstname = eElement.getElementsByTagName("firstname").item(0).getTextContent();    System.***out***.println("Last Name : "  + eElement  .getElementsByTagName("lastname")  .item(0)  .getTextContent());  lastname = eElement.getElementsByTagName("lastname").item(0).getTextContent();    System.***out***.println("Nick Name : "  + eElement  .getElementsByTagName("nickname")  .item(0)  .getTextContent());  nickname = eElement.getElementsByTagName("nickname").item(0).getTextContent();    System.***out***.println("Marks : "  + eElement  .getElementsByTagName("marks")  .item(0)  .getTextContent());  mark = Integer.*parseInt*(eElement.getElementsByTagName("marks").item(0).getTextContent());    **try**(Connection conn = DriverManager.*getConnection*(***DB\_URL***, ***USER***, ***PASS***);  Statement stmt = conn.createStatement();)  {  String sql\_insert = "INSERT INTO class(id,first,last,nick,mark) VALUES (?,?,?,?,?)";  PreparedStatement preparedStatement = conn.prepareStatement(sql\_insert);  preparedStatement.setInt(1, id);  preparedStatement.setString(2, firstname);  preparedStatement.setString(3, lastname);  preparedStatement.setString(4, nickname);  preparedStatement.setInt(5, mark);  System.***out***.println(preparedStatement);  preparedStatement.executeUpdate();  }  **catch** (SQLException e)  {  e.printStackTrace();  }  }  }  }  **catch** (Exception e)  {  e.printStackTrace();  }  }  } |