1. **import java.sql.\*;**
2. **Kết nối với database sử dụng code của java.**

|  |
| --- |
| String server = "LAPTOP-DungDepTrai";  String user = "sa";  String password = "123";  String db = "bai1\_tuan1";  int port = 1433;  String cadena = "jdbc:sqlserver://" + server + ":" + port+";" + "databaseName=" + db + ";trustServerCertificate = true;";  try(Connection conn = DriverManager.getConnection(cadena, user, password))  {  }  Catch(Exception e)  {  System.out.println( e);  } |

* Server là tên của server.
* User: là tài khoản cần kết nối.
* Password: mật khẩu của tài khoản cần kết nối.
* Db: là tên của database cần kết nối.
* Port: coognr port kết nối được thiết lập từ trước.
* Tiếp tục là một khối lệnh try catch để thiết lập kết nối với database.

1. **Một số cách sử lí với code sql trong java.**

* **Lấy dữ liệu từ sql lên.**

|  |
| --- |
| Statement statement = conn.createStatement();  String query = "select \* from CTHD";  ResultSet resultSet = statement.executeQuery(query);  // Hiển thị dữ liệu trong console  while (resultSet.next()) {  int sohd = resultSet.getInt("SOHD");  String name = resultSet.getString("MASP");  // Thêm các cột khác tùy theo cấu trúc của bảng  int sl = resultSet.getInt("SL");  System.out.println("SOHD: " + sohd + ", MASP: " + name + ", SOLUONG: " + sl);  } |

* + khi lấy dữ liệu từ sql lên thì kết quả của câu truy vấn là một bảng.
  + Hiểu một cách đơn giản thì biến resultSet chỏ đến dữ liệu của một mảng hai chiều (mảng hai chiều này tương đương dữ liệu của một bảng). Ban đầu nó chỏ đến vị trí đầu tiên của mảng hai chiều này.
  + Giả sử chúng ta đang lấy dữ liệu từ bảng CTHD(SOHD, MASP, SL).
  + Loại dữ liệu của các thuộc tính của bảng này lần lượt là int, string, string.

|  |
| --- |
| int sohd = resultSet.getInt("SOHD");  String name = resultSet.getString("MASP");  // Thêm các cột khác tùy theo cấu trúc của bảng  int sl = resultSet.getInt("SL"); |

* + Phần code trên có tác dụng lấy ra giá trị của từng hàng trong đó.
* **Insert dữ liệu vào bảng.**

|  |
| --- |
| String query = "insert into CTHD (CTHD.SOHD, CTHD.MASP, CTHD.SL) values (?, ?, ?)";  try (PreparedStatement preparedStatement = conn.prepareStatement(query)) {  // Thiết lập giá trị cho các tham số  preparedStatement.setInt(1, 1023);  preparedStatement.setString(2, "ST05");  preparedStatement.setInt(3, 100);  // Thực hiện truy vấn INSERT  int rowsAffected = preparedStatement.executeUpdate();  if (rowsAffected > 0) {  System.out.println("Dữ liệu đã được thêm thành công!");  } else {  System.out.println("Kh ông có dữ liệu nào được thêm.");  }  }  Catch(Exception e )  **{**  **System.out.println(e);**  **}** |

* + Chúng ta có thể thấy giá trị values trong này được đặt là ?, ?, ?.
  + Các dấu ? này sẽ được thay thế giá trị bằng các giá trị ở trong phần try-catch.
  + preparedStatement.setInt, hay setString,… gồm có hai đối số.
    - đối số đầu tiên tượng trưng cho vị trí của dấu chấm hỏi (vị trí này được đếm từ 1).
    - Đối số thứ hai tượng trưng cho giá trị cần truyền vào dấu ?.
* **Update dữ liệu.**

|  |
| --- |
| String query = "update CTHD\n" +  "set \n" +  " CTHD.MASP = ?,\n" +  " CTHD.SL = ?\n" +  "where \n" +  " CTHD.SOHD = 1023 and \n" +  " CTHD.MASP = 'ST05' and \n" +  " CTHD.SL = 10\n" +  ";";  try (PreparedStatement preparedStatement = conn.prepareStatement(query)) {  // Thiết lập giá trị cho các tham số  preparedStatement.setString(1, "ST10");  preparedStatement.setInt(2, 1001);  // Thực hiện truy vấn INSERT  int rowsAffected = preparedStatement.executeUpdate();  if (rowsAffected > 0) {  System.out.println("Dữ liệu đã được thêm thành công!");  } else {  System.out.println("Kh ông có dữ liệu nào được thêm.");  }  }  }  catch (Exception e)  {  e.printStackTrace();  } |

* + Tương tự như phần insert thì chúng ta có quyền thay thế các ? bằng cách sử dụng các code trong phần try catch.
* **Xử lí code để hiển thị toàn bộ là tiếng việt.**

|  |
| --- |
| import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.ResultSet;  import java.io.PrintWriter;  import java.nio.charset.StandardCharsets;  public class DatabaseConnectionExample {  public static void main(String[] args) {  connectToDatabase();  }  public static void connectToDatabase() {  String server = "LAPTOP-DungDepTrai";  String user = "sa";  String password = "123";  String db = "School\_Management";  int port = 1433;  String cadena = "jdbc:sqlserver://" + server + ":" + port + ";databaseName=" + db + ";trustServerCertificate=true;characterEncoding=UTF-8;useUnicode=true;";  try (Connection conn = DriverManager.getConnection(cadena, user, password)) {  java.sql.Statement statement = conn.createStatement();  String loadQuery = loadStudentData();  ResultSet resultSet = statement.executeQuery(loadQuery);  while (resultSet.next()) {  String id = resultSet.getString("st\_id");  String name = resultSet.getString("st\_name");  String className = resultSet.getString("cl\_name");  // Sử dụng PrintWriter với Charset là UTF-8  PrintWriter out = new PrintWriter(System.out, true, StandardCharsets.UTF\_8);  out.println(id + " " + name + " " + className);  }  } catch (Exception e) {  e.printStackTrace();  }  }  public static String loadStudentData() {  return "select \* from student inner join belonging on student.st\_id = belonging.st\_id inner join class on class.cl\_id = belonging.cl\_id;";  }  } |