# LAB8. LẬP TRÌNH CƠ SỞ DỮ LIỆU (BUỔI 1)

## Mục tiêu

* Thực hiện cấu hình SQLServer
* Cấu hình Netbeans kết nối với SQLServer
* Thao tác cơ bản với CSDL: Tạo bảng, select dữ liệu, thêm, sửa, xóa bản ghi.

## Yêu cầu

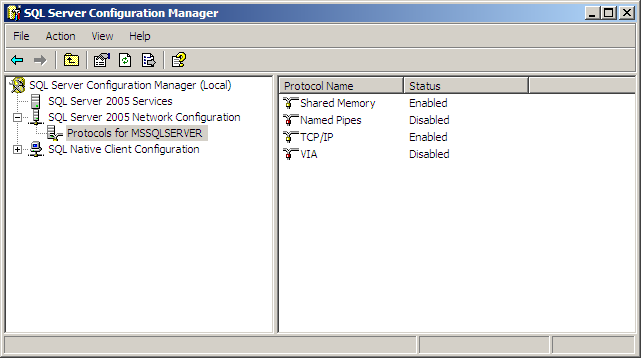
* Sinh viên tự cài SQL Server ở nhà (phiên bản tùy chọn, khuyến nghị bản 2008 hoặc 2012).
* Download bộ thư viện jdbc.jar tại hệ thống elearning của bộ môn.

## Hướng dẫn

* ***Hướng dẫn cấu hình SQL Server***

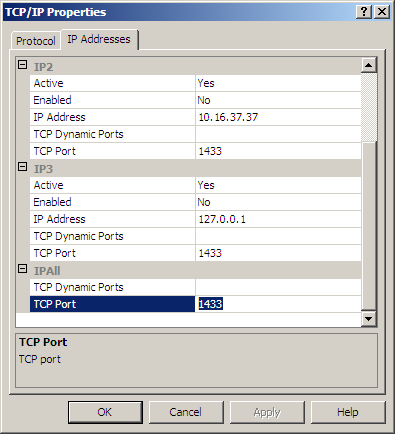
Để kết nối với SQL cần mở cổng cho SQL, cách cài đặt như sau:

Mở ***Configuration Tools > SQL Server Configuration Manager.***



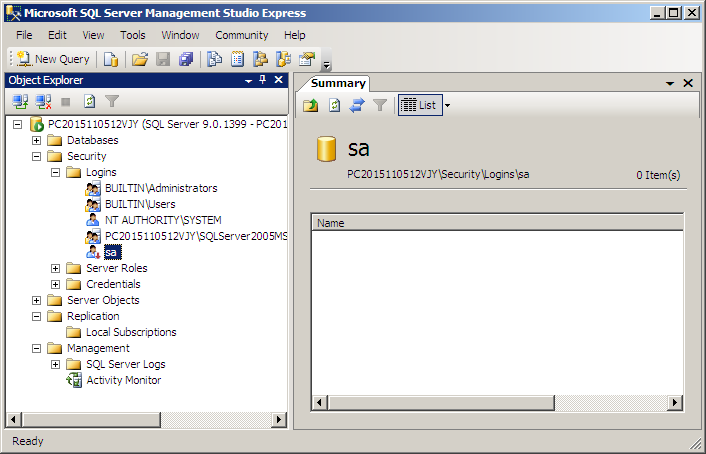
Mở ***SQL Server Network Configuration > Protocols for MSSQLSERVER.*** Chọn ***TCP/IP*** và thực hiện đặt trạng thái Enabled.

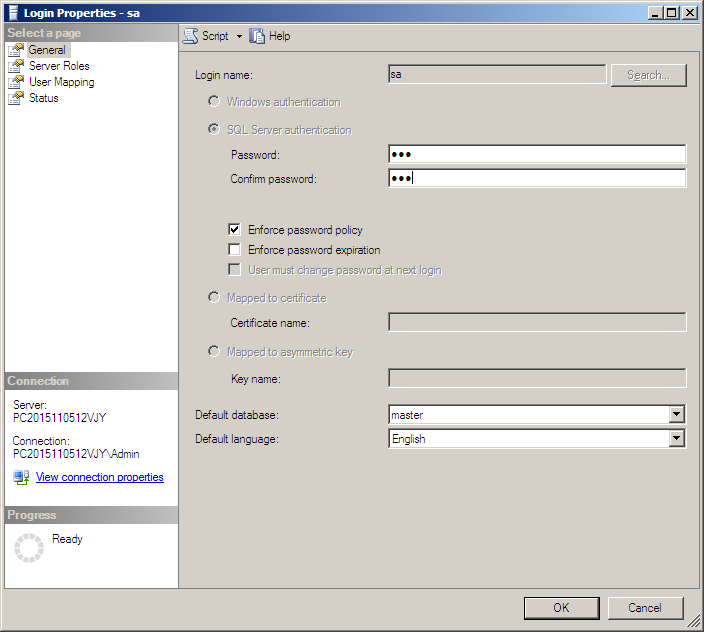
Click double vào ***TCP/IP*** và đặt cổng là 1433, sau đó hãy khởi động lại SQL Server.

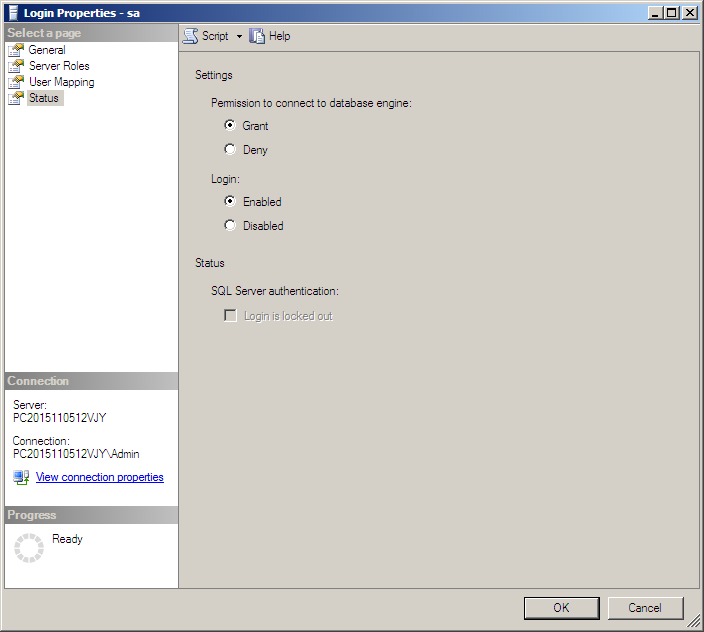


Như vậy chúng ta đã mở cổng cho SQL Server để kết nối vào cơ sở dữ liệu.

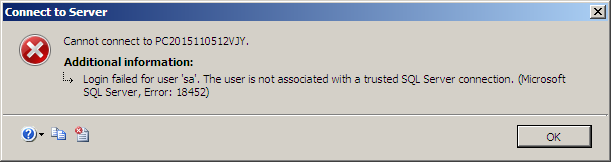
Bước tiếp theo chúng ta cần thiết lập 1 tài khoản để sử dụng cho việc kết nối vào SQL Server. Trong trường hợp này, chúng ta sẽ sử dụng tài khoản ***sa*** (tài khoản root) để kết nối. Thực hiện đổi mật khẩu cho tài khoản ***sa*** và đặt trạng thái Enable cho phép sử dụng tài khoản **sa** để đăng nhập vào cơ sở dữ liệu như sau:





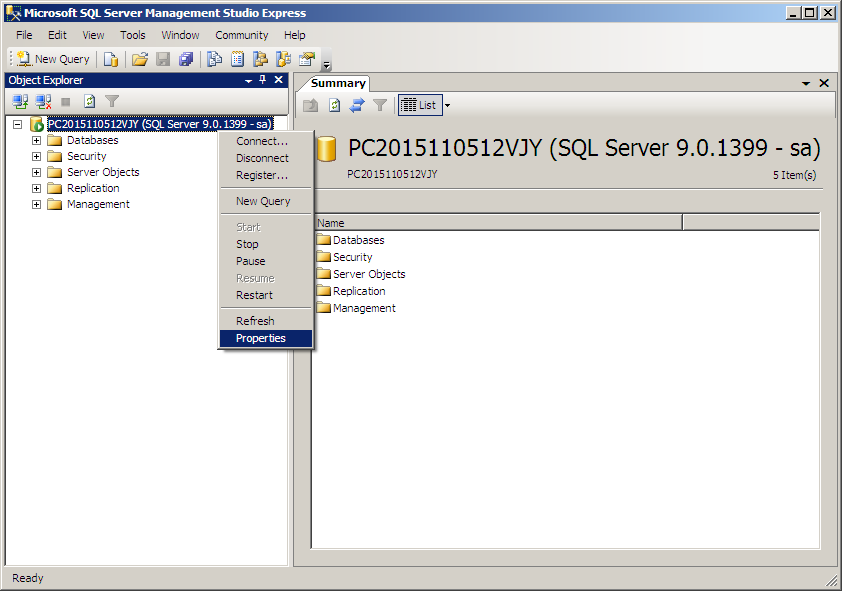


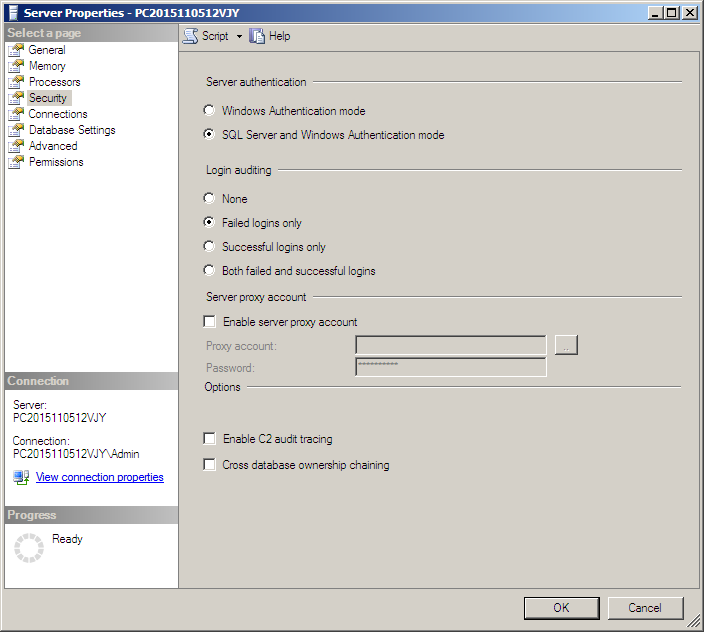
Thực hiện ngắt kết nối với SQL Server và đăng nhập bằng tài khoản ***sa***. Trong trường hợp không đăng nhập được với lỗi như hình dưới:



Ta thực hiện cấu hình lại như sau:

Click phải chuột vào ***Server Name> Properties,*** chuyển sang ***Tab Security,*** chọn chế độ ***SQL Server and Windows Authentication mode.***

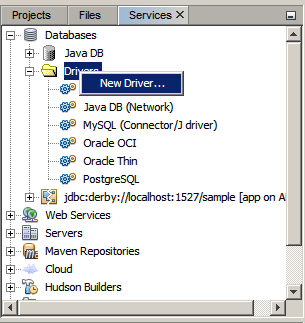
******

******

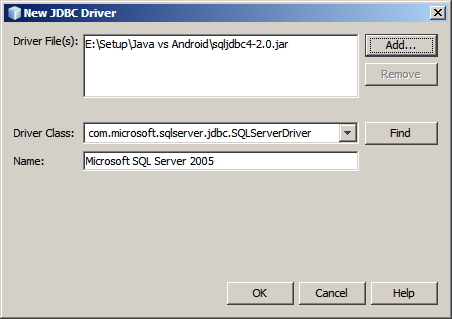
Thực hiện khởi động lại SQL Server và đăng nhập bằng tài khoản **sa.**

* ***Hướng dẫn kết nối tới SQLServer***

Trên ***Netbean***, tại cửa sổ ***Services (Windows > Services),***  chúng ta tiến hành thêm Driver của SQLServer

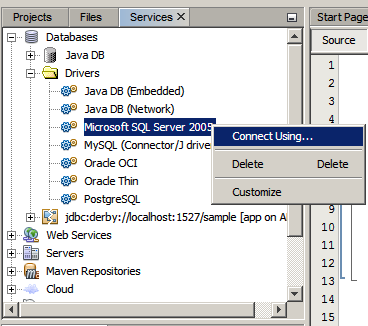
**

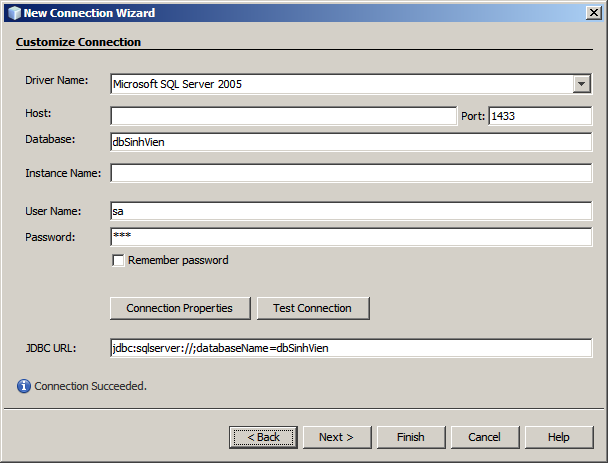
*Thêm Driver của SQL Server*



*Add driver từ đường dẫn*

Sau khi thêm driver xong, tại cửa sổ ***Service*** chúng ta sẽ thấy có Driver ***Microsoft SQL Server 2005***. Click phải chuột chọn ***Connect Using...*** Tiến hành cấu hình kết nối với các thông tin như hình:





Trong đó:

* ***Port:*** Là cổng kết nối chúng ta đã cài đặt trên SQL Server ở phần 4.3.1
* ***Database:*** Tên cơ sở dữ liệu chúng ta cần kết nối tới
* ***Username, password:*** Tên tài khoản/Mật khẩu chúng ta đã thiết lập
* Nhấn ***Test Connection*** để kiểm tra kết nối tới SQL Server thành công hay chưa.

## Bài tập

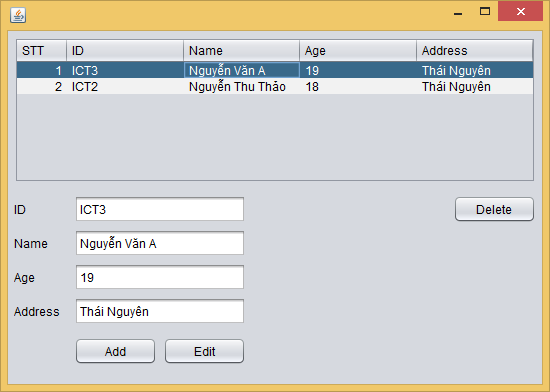
***Bài 1.*** Xây dựng chương trình quản lý sinh viên với các thông tin sau:

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| ID | int | Mã sinh viên |
| Name | String | Tên sinh viên |
| Address | String | Địa chỉ |
| Age | int | Tuổi |

1. Thiết kế giao diện cho chương trình
2. Hãy viết chương trình hiển thị dữ liệu ra màn hình
3. Viết chương trình thêm/sửa/xóa dữ liệu cho bảng tblSinhVien

**Hướng dẫn:**

1. Giao diện chương trình ta sẽ xây dựng như sau:

****

* Khi nhấn chọn 1 dòng trên JTable, thông tin của sinh viên đó sẽ được điền vào các JTextField tương ứng.
* Khi nhấn nút Add/Edit thì sẽ thực hiện thêm/sửa thông tin một sinh viên vào CSDL với các thông tin lấy từ JTextField.

1. Ta xây dựng lớp ***DBHelper*** để thực hiện thao tác với CSDL như sau:

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class DBHelper {

// Tạo biến tên cơ sở dữ liệu

public static String dbName = "";

// Phương thức kết nối tới cơ sở dữ liệu

public Connection getConnectDB(String dbName) {

Connection conn = null;

try {

String URL =

"jdbc:sqlserver://127.0.0.1:1433;"

+ "databaseName=" + dbName;

String userName = "sa";

String pwd = "123";

/\*

Trong đó:

- 127.0.0.1:1433 – Địa chỉ ip localhost

và port 1433 ta đã thiết lập.

- userName và pwd là mật khẩu và password

của tài khoản đăng nhập vào SQLServer

\*/

Class.forName("com.microsoft.sqlserver."

+ "jdbc.SQLServerDriver");

conn = DriverManager.getConnection(

URL, userName, pwd);

System.out.println("Kết nối thành công!");

} catch (ClassNotFoundException | SQLException ex){

System.out.println("Kết nối thất bại!");

}

return conn;

}

// Phương thức đóng kết nối

public void CloseConnection(Connection conn) {

if (conn != null) {

try {

conn.close();

} catch (Exception ex) {

}

}

}

public ResultSet ShowData(String sql) {

Connection conn = getConnectDB(dbName);

Statement state = null;

ResultSet rs = null;

try {

state = conn.createStatement();

rs = state.executeQuery(sql);

} catch (Exception ex) {

}

return rs;

}

public boolean UpdateData(String sql) {

Connection conn = getConnectDB(dbName);

int row = 0;

try {

Statement state =

(Statement) conn.createStatement();

row = state.executeUpdate(sql);

if (row > 0) {

return true;

}

} catch (SQLException ex) {

System.err.println("Error: " +

ex.getMessage());

} finally {

CloseConnection(conn);

}

return false;

}

}

Tiếp theo ta xây dựng lớp ***SinhVien*** thuộc file ***SinhVien.java*** cho phép thêm, sửa, xóa dữ liệu:

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.swing.table.DefaultTableModel;

public class SinhVien {

// Khởi tạo đối tượng DBHelper

public DBHelper dbHelper = new DBHelper();

// Hàm khởi tạo SinhVien

public SinhVien() {

dbHelper.dbName = "dbSinhVien";

}

// Hiển thị dữ liệu

public DefaultTableModel ShowData() {

DefaultTableModel model = new DefaultTableModel();

model.addColumn("STT");

model.addColumn("ID");

model.addColumn("Name");

model.addColumn("Age");

model.addColumn("Address");

String sql = "Select \* from tblSinhVien";

try {

ResultSet rs = dbHelper.ShowData(sql);

int stt = 0;

while (rs.next()) {

stt++;

model.addRow(

new Object[]{stt,

rs.getString(0),

rs.getString(1),

rs.getString(2),

rs.getString(3)});

}

} catch (SQLException ex) {

}

return model;

}

// Thêm dữ liệu

public boolean InsertData(

int id, String name, int age, String address) {

String sql = "INSERT INTO tblSinhVien "

+ "values('"

+ id + "', '"

+ name + "', '"

+ age + "', '"

+ address + "')";

if (dbHelper.UpdateData(sql)) {

System.out.println("Thêm thành công!");

return true;

} else {

System.out.println("Thêm thất bại!");

return false;

}

}

// Phương thức sửa thông tin sinh viên

public boolean UpdateData(

int id, String name, int age, String address) {

String sql = "UPDATE tblSinhVien SET Name='" + name

+ "', Address='" + address

+ "', Age='" + age

+ "' WHERE ID='" + id + "'";

if (dbHelper.UpdateData(sql)) {

System.out.println("Sửa thành công!");

return true;

} else {

System.out.println("Sửa thất bại!");

return false;

}

}

// Hàm xóa sinh viên với id truyền vào

public boolean Delete(int id) {

String sql =

"DELETE FROM tblSinhVien WHERE ID='" + id + "'";

if (dbHelper.UpdateData(sql)) {

System.out.println("Xóa thành công!");

return true;

} else {

System.out.println("Xóa thất bại!");

return false;

}

}

}

Trong file giao diện, ta bắt sự kiện **formWindowActivated** cho JFrame để hiển thị dữ liệu mỗi lần Frame được hiển thị lên.

// Khởi tạo đối tượng sinh viên

SinhVien sv = new SinhVien();

**private** **void** formWindowActivated(java.awt.event.WindowEvent evt) {

jTable1.setModel(sv.ShowData());

}

Các phương thức khác thuộc lớp **SinhVien** ta cũng sử dụng tương tự.

## Bài tập nâng cao

***Bài 1.*** Xây dựng chương trình quản lý vật dụng nhà kho. Bảng dữ liệu gồm các trường như sau:

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| MaVatDung | Int | Mã vật dụng |
| TenVatDung | String | Tên vật dụng |
| DonVi | String | Đơn vị (cái, chiếc, ...) |
| SoLuong | Int | Số lượng |

***Bài 2.*** Xây dựng chương trình quản lý danh mục sách. Hướng dẫn: Các trường của bảng dữ liệu gồm: Mã sách, tên sách, số trang, nhà xuất bản, năm xuất bản, thể loại.