

Session 9**JDBC API****Phần I - Thực hiện trong 120 phút****1. Mục tiêu**

- Hiểu biết cơ bản về JDBC và kiến trúc của nó.
- Cách gọi và xây dựng kết nối CSDL sử dụng JDBC.
- Tạo truy vấn cơ bản với cú pháp SQL.

2. Thực hiện

Bài thực hành 1: Tạo CSDL tên là qlsv để quản lý sinh viên, tạo ứng dụng java kết nối tới csdl này, hiển thị thông báo nếu kết nối thành công hoặc không.

Bước 1: Tạo CSDL cho ứng dụng.

Khởi động "SQL Server Management Studio", Kích vào tạo một New Query và tạo csdl theo lệnh dưới đây

```
-- Tạo database qlsv
CREATE DATABASE qlsv;
go
-- Sử dụng database qlsv
USE qlsv;
go

-- Tạo bảng
CREATE TABLE tblsinhvien (
    id int NOT NULL identity,
    rollnumber varchar(15) NOT NULL unique,
    name nvarchar(50) NOT NULL,
    address nvarchar(200) DEFAULT NULL,
    phone varchar(11) DEFAULT NULL,
    gender bit NOT NULL,
    PRIMARY KEY (id),
);

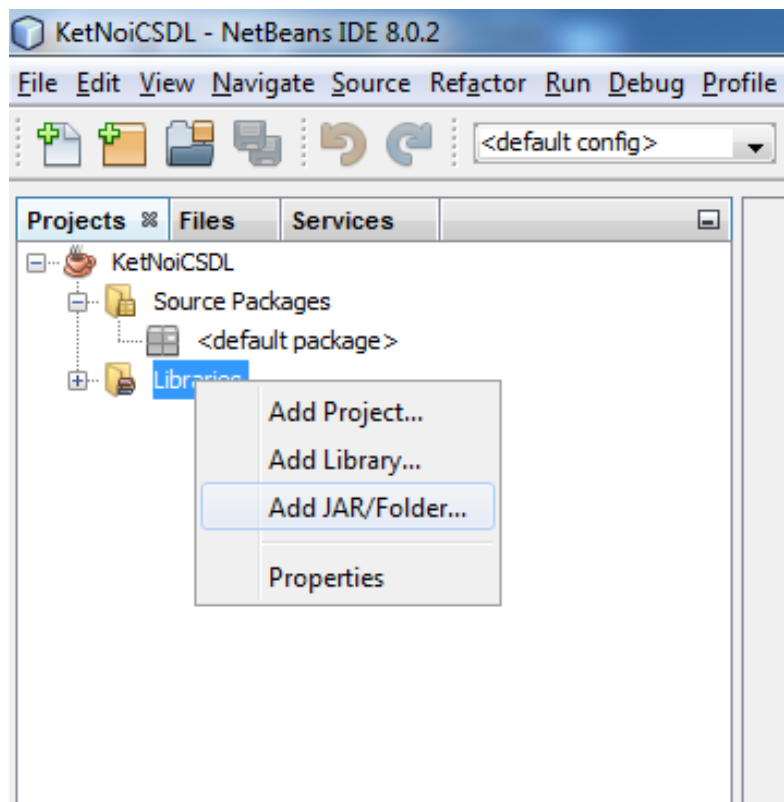
--Thêm dữ liệu vào bảng tblsinhvien

INSERT INTO tblsinhvien (rollnumber, name, address, phone, gender) VALUES
('C1505001', 'Ma Sieu', 'Tay Luong', '19001265', 1),
('C1505002', 'Dieu Thuyen', 'To Chau', '19001001', 0),
('C1505003', 'Ma Sieu', 'Tay Luong', '19001265', 1);

select * from tblsinhvien
```

Bước 2: Tạo một project trong NetBean đặt tên " KetNoiCSDL".

Click phải vào Libraries chọn "Add Jar folder"



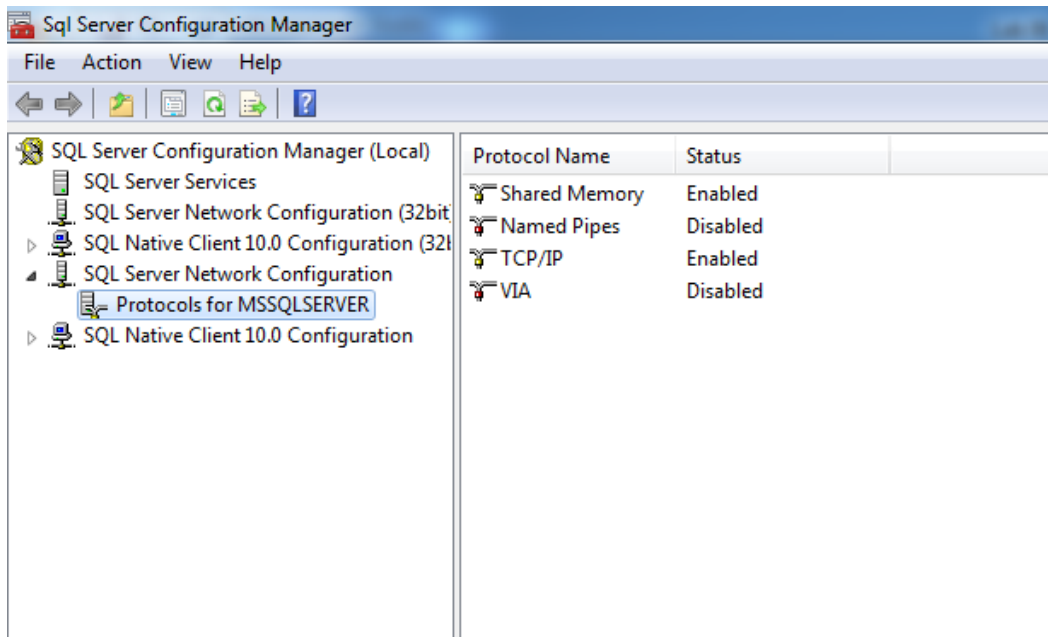
Tải hoặc copy thư viện "sqljdbc4.jar" về máy và tìm đến vị trí lưu thư viện này. Sau khi tích chọn thì chọn nút "Open" ở cửa sổ Add Jar folder.

Bước 3: Xem port kết nối của SQL Server

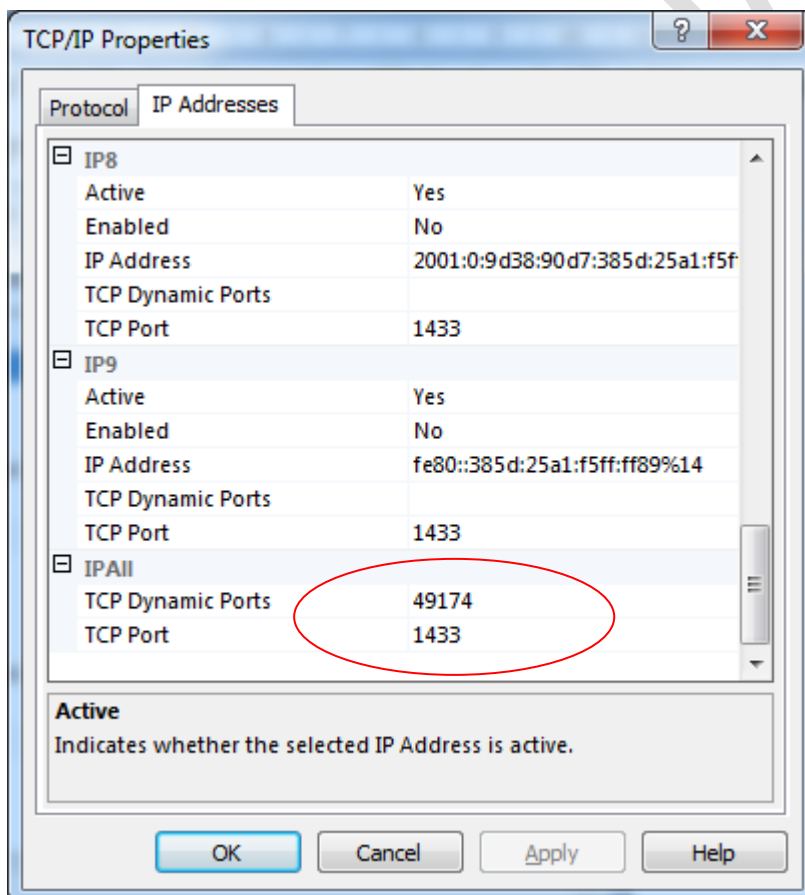
Trong cửa sổ tạo database của Sql Server có thể sử dụng thủ tục sau để xem port

```
sp_readerrorlog 0,1,'server is listening on'
```

Hoặc có thể xem port của sql server qua "SQL Server Configuration Manager" như sau: Khởi động "SQL Server Configuration Manager" chọn "Protocols for XXX" với XXX là tên của SQL Server được cài đặt trên máy.



Click phải vào TCP/IP chọn Properties, Click vào tab "IP Addresses" và kéo xuống dưới cùng ta sẽ thấy được port của SQL Server đang dùng



Bước 4: khai báo MainClass trong package test có hàm main() và viết đoạn mã sau:

```
package test;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author HAITHANH
 */
public class MainClass {
    public static void main(String[] args) {
        Connection con = null;
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            con =
DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=qlsv","s
a","1234567");
            if(con!=null)
                System.out.println("Kết nối tới CSDL thành công!");
            else
                System.out.println("Không kết nối vào CSDL được");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

Chú ý:

Để kết nối tới CSDL thì chúng ta cần có user và password để truy cập vào CSDL đó. Khi cài đặt chúng ta nên kích hoạt cho user 'sa' mặc định chạy lên hoặc có thể tạo một tài khoản bất kỳ để kết nối với CSDL qua tài khoản đó.

Bước 5: Chạy chương trình và xem kết quả, nếu không thành công sẽ có thông báo ngoại lệ.



Bài thực hành 2: Trong ứng dụng quản lý sinh viên trên, viết chức năng nhập thông tin từ bàn phím sau đó thêm dữ liệu nhập vào CSDL bằng cách sử dụng lệnh sql với đối tượng Prepared Statement.

Bước 1: Code class SinhVien.java

```
package test;

/**
 *
 * @author HAITHANH
 */
public class SinhVien {
    private int id;
    private String rollNumber;
    private String name;
    private String address;
    private String phoneNumber;
    private boolean gender;

    public SinhVien() {
    }

    public SinhVien(int id, String rollNumber, String name, String address, String
phoneNumber, boolean gender) {
        this.id = id;
        this.rollNumber = rollNumber;
        this.name = name;
    }
}
```

```
this.address = address;
this.phoneNumber = phoneNumber;
this.gender = gender;
}

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getRollNumber() {
    return rollNumber;
}

public void setRollNumber(String rollNumber) {
    this.rollNumber = rollNumber;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public String getPhoneNumber() {
    return phoneNumber;
}

public void setPhoneNumber(String phoneNumber) {
    this.phoneNumber = phoneNumber;
}

public boolean isGender() {
```

```
        return gender;
    }

    public void setGender(boolean gender) {
        this.gender = gender;
    }

    @Override
    public String toString() {
        return "SinhVien{" + "id=" + id + ", rollNumber=" + rollNumber + ", name=" +
name + ", address=" + address + ", phoneNumber=" + phoneNumber + ", gender="
+ gender + '}';
    }
}
```

Trong hàm main của lớp MainClass viết các lệnh để load all, insert, update, delete từ bảng tblsinhvien như sau:

```
package test;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author HAITHANH
 */
public class MainClass {

    public static void main(String[] args) {
        Connection con = null;
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            con =
DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=qlsv",
"sa", "1234567");
            if (con != null) {
                System.out.println("Kết nối tới CSDL thành công!");
            } else {
                System.out.println("Không kết nối vào CSDL được");
            }
        }
    }
}
```

```
} catch (ClassNotFoundException ex) {  
    Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);  
} catch (SQLException ex) {  
    Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);  
}
```

//BƯỚC 2: VIẾT MÃ ĐỂ LẤY THÔNG TIN TỪ BẢNG tblsinhvien

PreparedStatement pstmt = null;

ResultSet rs = null;

```
try {  
    pstmt = con.prepareStatement("select * from tblsinhvien");  
    rs = pstmt.executeQuery();  
    while (rs.next()) {  
        System.out.println("Sinh vien:");  
        System.out.println("\tId:" + rs.getInt(1));  
        System.out.println("\tRollnumber:" + rs.getString(2));  
        System.out.println("\tName:" + rs.getString(3));  
        System.out.println("\tAddress:" + rs.getString(4));  
        System.out.println("\tPhone no:" + rs.getString(5));  
        System.out.println("\tGender:" + (rs.getBoolean(6) ? "Male" : "Female"));  
    }  
} catch (SQLException ex) {  
    Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);  
}
```

//BƯỚC 4: VIẾT MÃ ĐỂ CẬP NHẬT DỮ LIỆU TRONG BẢNG

```
try {  
    pstmt = con.prepareStatement("update tblsinhvien set name=?, address=?,  
phone=? where id=?");  
    pstmt.setString(1, "Lưu Bị");  
    pstmt.setString(2, "Thục quốc");  
    pstmt.setString(3, "0123989898");  
    pstmt.setInt(4, 3);  
    int i = pstmt.executeUpdate();  
    if (i > 0) {  
        System.out.println("Da cap nhat du lieu cho ban ghi co id =3");  
    } else {  
        System.out.println("Khong cap nhat duoc du lieu");  
    }  
} catch (SQLException ex) {  
    Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);  
}
```

//BƯỚC 3: VIẾT MÃ ĐỂ THÊM THÔNG TIN SINH VIÊN VÀO BẢNG

tblsinhvien


```

try {
    pstmt = con.prepareStatement("insert into tblsinhvien values (?,?,?,?,?)");
    pstmt.setString(1, "C1505004");
    pstmt.setString(2, "Tào Tháo");
    pstmt.setString(3, "Nguyễn quốc");
    pstmt.setString(4, "19001080");
    pstmt.setBoolean(5, true);
    int i = pstmt.executeUpdate();
    if (i > 0) {
        System.out.println("Da them du lieu vao bang");
    } else {
        System.out.println("Khong them duoc du lieu vao bang");
    }
} catch (SQLException ex) {
    Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);
}

//BƯỚC 5: VIẾT MÃ ĐỂ XÓA DỮ LIỆU TRONG BẢNG tblsinhvien
try {
    pstmt = con.prepareStatement("delete from tblsinhvien where id=3");
    int i = pstmt.executeUpdate();
    if (i > 0) {
        System.out.println("Da xoa du lieu cua ban ghi co id=3");
    } else {
        System.out.println("Khong xoa duoc du lieu");
    }
} catch (SQLException ex) {
    Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);
} finally {
    try {
        con.close();
        pstmt.close();
        rs.close();
    } catch (SQLException ex) {
        Logger.getLogger(MainClass.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}
}

```

Bài thực hành 3: Sử dụng bảng tblsinhvien ở trên, tạo một thủ tục lưu trữ (store procedure) trong sql server để lấy thông tin các sinh viên và hiển thị kết quả lên màn hình console.

Bước 1: Tạo thủ tục lưu trữ sau trong sql server

```
create procedure GetAllSinhVien
as

select * from tblsinhvien
```

Bước 2: Tạo class GoiThuTuc có hàm main và viết lệnh để gọi thủ tục và xử lý kết quả

```
package test;

import java.sql.CallableStatement;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author HAITHANH
 */
public class GoiThuTuc {

    public static void main(String[] args) {
        Connection con = null;
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            con =
DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=qlsv",
"sa", "1234567");
            if (con != null) {
                System.out.println("Kết nối tới CSDL thành công!");
            } else {
                System.out.println("Không kết nối vào CSDL được");
            }
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(GoiThuTuc.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(GoiThuTuc.class.getName()).log(Level.SEVERE, null, ex);
        }

        //VIẾT LỆNH ĐỂ GỌI THỦ TỤC VÀ XỬ LÝ KẾT QUẢ
        CallableStatement cstmt;
        ResultSet rs;

        try {
            cstmt = con.prepareCall("{call GetAllSinhVien}");
```

```
rs = pstmt.executeQuery();
while (rs.next()) {
    System.out.println("Sinh vien:");
    System.out.println("\tId:" + rs.getInt(1));
    System.out.println("\tRollnumber:" + rs.getString(2));
    System.out.println("\tName:" + rs.getString(3));
    System.out.println("\tAddress:" + rs.getString(4));
    System.out.println("\tPhone no:" + rs.getString(5));
    System.out.println("\tGender:" + (rs.getBoolean(6) ? "Male" : "Female"));
}
} catch (SQLException ex) {
    Logger.getLogger(GoiThuTuc.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
```

Phần II - Bài tập tự làm

1. Tạo ứng dụng quản lý nhân viên có menu như sau:

===== MENU =====

1. Nhập dữ liệu nhân viên.
2. Hiển thị danh sách nhân viên.
3. Tìm kiếm nhân viên theo email
4. Tìm kiếm nhân viên theo tên
5. Sửa thông tin nhân viên theo mã nhân viên.
6. Xóa thông tin nhân viên theo mã nhân viên.
7. Thoát chương trình.

=====

Yêu cầu:

- Tạo bảng tblNhanVien gồm các thông tin: (MaNV, HoTen, NamSinh, PhongBan, Email)
- Khóa chính là MaNV (kiểu int, tự tăng)

```
create table tblNhanVien (MaNV int identity primary key,
HoTen nvarchar(50), NamSinh int, PhongBan nvarchar(200), Email nvarchar(100))
```

Thực hiện các chức năng có trên menu.

2. Tạo ứng dụng quản lý chi tiêu có menu như sau:

===== MENU =====

1. Danh sách danh mục

2. Nhập giao dịch
3. Sửa giao dịch
4. Xóa giao dịch
5. Thống kê chi tiêu theo danh mục.
6. Thoát

=====

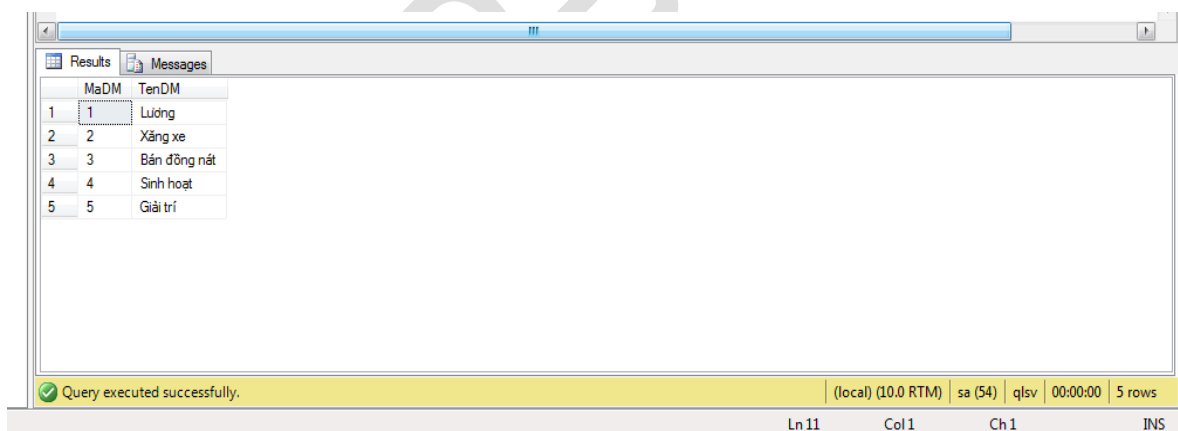
Yêu cầu:

- Tạo bảng tblDanhMuc gồm các thông tin: (MaDM, TenDM), dữ liệu được nhập cố định vào database

```
create table tblDanhMuc (MaDM int identity primary key, TenDM nvarchar(200))

insert into tblDanhMuc values (N'Lương')
insert into tblDanhMuc values (N'Xăng xe')
insert into tblDanhMuc values (N'Bán đồ ăn vặt')
insert into tblDanhMuc values (N'Sinh hoạt')
insert into tblDanhMuc values (N'Giải trí')
```

Dữ liệu của bảng tblDanhMuc



	MaDM	TenDM
1	1	Lương
2	2	Xăng xe
3	3	Bán đồ ăn vặt
4	4	Sinh hoạt
5	5	Giải trí

Query executed successfully. (local) (10.0 RTM) sa (54) qlsv 00:00:00 5 rows

Ln 11 Col 1 Ch 1 INS

- Tạo bảng tblGiaoDich gồm các thông tin

```
create table tblGiaoDich (MaGD int identity primary key, TenGD nvarchar(200),
NgàyGD date, MaDM int foreign key references tblDanhMuc(MaDM), SoTien
float)
```

MaDM là khóa ngoại liên kết với cột MaDM trong bảng tblDanhMuc.

- Thực hiện các chức năng có trên menu.
- Sửa, xóa giao dịch thông qua id.

- Giao dịch được gán thuộc danh mục cố định (tạo mảng, hiển thị lựa chọn cho người dùng). Ví dụ:
 1. Lương
 2. Bán đồng nát
 3. Sinh hoạt
 4. Giải trí
 5. Xăng xe
- Thống kê chi tiêu theo danh mục hiển thị % số tiền. Ví dụ tổng chi là 100 đồng, trong đó chi cho sinh hoạt là 30 đồng => Danh mục sinh hoạt 30%