

CHƯƠNG 9 - LẬP TRÌNH TRUY XUẤT CƠ SỞ DỮ LIỆU (JDBC)

I. Tóm tắt lý thuyết

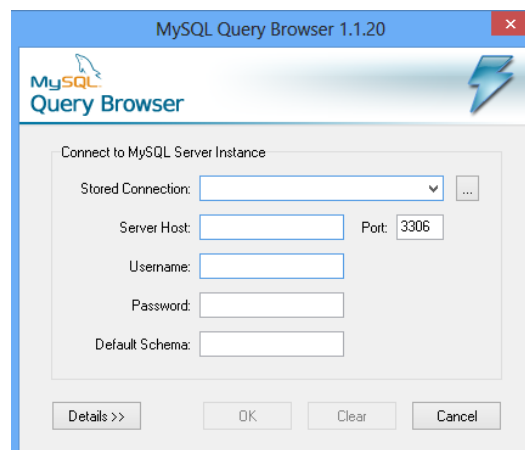
- Java cung cấp thư viện (gói java.sql) gồm các lớp và interface cho phép chương trình kết nối với các hệ cơ sở dữ liệu để truy xuất và xử lý dữ liệu.
- Các hệ cơ sở dữ liệu như MySQL Server, Access, Oracle,... được xem như các chương trình server lưu trữ dữ liệu. Để chương trình Java kết nối và trao đổi dữ liệu được với các hệ cơ sở dữ liệu, chương trình Java cần phải nạp Driver tương ứng.
- Driver được xem như chương trình trung gian có nhiệm vụ:
 - o Chuyển đổi các yêu cầu gửi đến hệ cơ sở dữ liệu từ chương trình Java thành định dạng mà hệ cơ sở dữ liệu có thể hiểu được,
 - o Chuyển đổi dữ liệu gửi lên chương trình Java từ hệ cơ sở dữ liệu thành định dạng dữ liệu có thể biểu diễn ở chương trình Java.
- Các bước kết nối và truy xuất cơ sở dữ liệu của một chương trình Java
 - 1) Nạp Driver tương ứng với hệ cơ sở dữ liệu cần kết nối.
 - 2) Thực hiện kết nối đến hệ cơ sở dữ liệu (tạo đối tượng Connection).
 - 3) Trao đổi dữ liệu với hệ cơ sở dữ liệu (Tạo đối tượng của các lớp Statement hoặc PreparedStatement hoặc CallableStatement để thực hiện việc trao đổi dữ liệu với hệ cơ sở dữ liệu thông qua các phương thức như executeQuery(), executeUpdate(), execute (...)).
- Kết quả truy vấn dữ liệu bằng câu lệnh SELECT được lưu trữ bởi đối tượng của lớp ResultSet.
- ResultSet tương tự như một bảng gồm các hàng và các cột. Để truy xuất dữ liệu của đối tượng ResultSet, cần phải di chuyển con trỏ đến từng hàng bằng hàm next() và lấy giá trị các ô dữ liệu bằng các hàm tương ứng kiểu dữ liệu tại ô đó (getString(), getInt(), getFloat(...)).
- Đối tượng ResultSet vừa được tạo ra có con trỏ ở vị trí hàng trước hàng đầu tiên (beforeFirst) không có dữ liệu, do đó cần phải di chuyển con trỏ đến hàng đầu để bắt đầu truy xuất dữ liệu. Khi con trỏ di chuyển đến vị trí hàng sau hàng cuối cùng (afterLast), không còn chứa dữ liệu thì hàm next() trả lại giá trị false.
- Các đối tượng DatabaseMetaData và ResultSetMetaData chứa thông tin tương ứng về cấu trúc của hệ cơ sở dữ liệu và của ResultSet.
- Dùng đối tượng PreparedStatement hiệu quả hơn đối tượng Statement khi làm việc với cùng một câu lệnh SQL mà được thực hiện nhiều lần bởi các đối số khác nhau.
- Đối tượng CallableStatement được dùng để truy xuất dữ liệu các procedures của hệ cơ sở dữ liệu.
- Dùng phương thức setAutoCommit(false) để ngăn không cho việc gửi câu lệnh SQL đến hệ cơ sở dữ liệu một cách tự động. Câu lệnh SQL được

thực sự gửi đến hệ cơ sở dữ liệu và có hiệu lực đến khi hàm commit() được gọi.

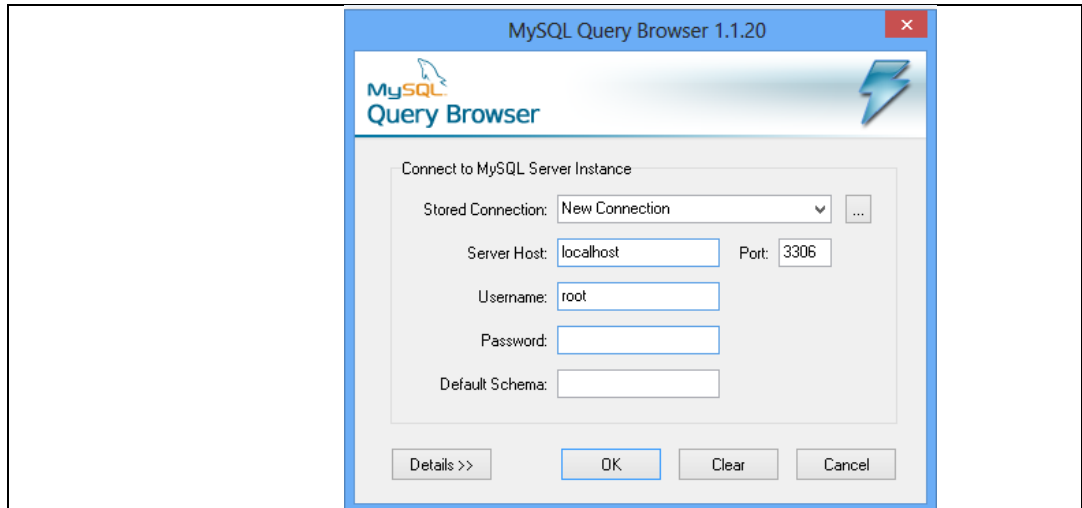
II. Bài tập thực hành mẫu

1. Cài đặt hệ cơ sở dữ liệu My SQL Server và cài đặt môi trường chuẩn bị lập trình truy xuất cơ sở dữ liệu.

- 1) Cài đặt hệ cơ sở dữ liệu MySQL Server và công cụ làm việc đi kèm
 - a. Download và cài đặt hệ cơ sở dữ liệu MySQL Server (MySQL Database Server) tại địa chỉ <http://downloads.mysql.com/archives.php>.
 - b. Download và cài đặt MySQL Query Browser tại địa chỉ: <http://downloads.mysql.com/archives.php?p=mysql-query-browser>
- 2) Khởi động MySQL Server và cấu hình MySQL Query Browser
 - a. Khởi động MySQL Server: đối với hệ điều hành Windows, chạy Command Prompt ở chế độ administrator, tại cửa sổ Command Prompt chạy tệp tin *mysqld.exe* trong thư mục *bin* của MySQL Server (đối với Linux, shell> *mysql.server start*)
 - b. Chạy MySQL Query Browser, sẽ xuất hiện màn hình như sau:



Giả sử MySQL Query Browser chạy trên cùng một máy với MySQL Server, chúng ta sẽ cài đặt như sau:



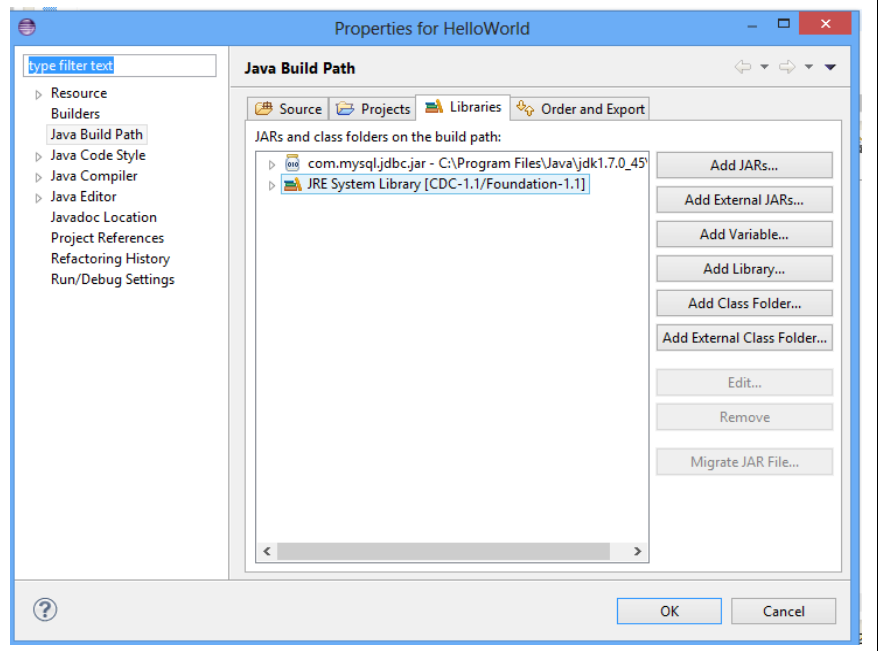
Chọn nút OK, sau đó thao tác với các cơ dữ liệu.

3) Nạp driver vào Eclipse để chương trình Java có thể kết nối đến MySQL Server.

a. Download com.mysql.jdbc.jar

<http://www.java2s.com/Code/Jar/c/Downloadcommysqljdbc515jar.htm>
]

b. Nạp driver vào Eclipse: Trong môi trường Eclipse, chọn Project, chọn Properties, chọn Java Build Path, chọn Add External JARS..., chọn tệp tin com.mysql.jdbc.jar vừa mới download và cuối cùng chọn OK.



2. Tạo cơ sở dữ liệu có tên StudentManagement, tạo bảng Students gồm các trường ID, Name, Math, Phys, Chem và Aver trong MySQL Server. Viết chương trình Java kết nối đến cơ sở dữ liệu StudentManagement, truy vấn và hiển thị tên cột của bảng Students.

```

import java.sql.*;
class ResultSetMetaDataDemo
{
    public static void main( String args[] )
    {
        try
        {
            /*Nạp driver */
            Class.forName("com.mysql.jdbc.Driver");

            /*Kết nối đến cơ sở dữ liệu */
            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/studentmana
            gement","root", "");

            /* Tạo đối tượng Statement để gửi các câu lệnh SQL*/
            Statement sm = conn.createStatement();

            /*Truy vấn thông tin từ bảng students */
            ResultSet rs=sm.executeQuery("Select * from students");

            /*Tạo ResultSetMetaData để Lấy thông tin của ResultSet*/
            ResultSetMetaData rsm= rs.getMetaData();

            /*Lấy thông tin số cột của bảng students từ đối tượng rsm*/
            int col_num = rsm.getColumnCount();

            /*Lấy tên cột của bảng students từ đối tượng rsm*/
            for (int i=1;i<=col_num;i++)
                System.out.print(rsm.getColumnLabel(i)+ " ");

            } catch( Exception e )
            {
                System.out.println( e );
            }
        }
    }
}

```

Kết quả chương trình:

```

<terminated> ResultSetMetaDataDemo [Java Application] C:\Program Files\Java\jdk1.7.0_45\jre\bin\javaw.exe (Nov 13, 2
ID   Name   Math   Phys   Chem   Aver

```

- Viết chương trình Java kết nối đến cơ sở dữ liệu StudentManagement, thêm thông tin vào bảng Students sau đó truy vấn thông tin từ bảng Students để hiển thị ra màn hình.

```
import java.sql.* ;
```

```

class JDBCdemo
{
public static void main( String args[] )
{
try
{

/*Nạp driver */
Class.forName("com.mysql.jdbc.Driver");

/*Kết nối đến cơ sở dữ liệu và tạo đối tượng Statement*/
Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/student
management","root", "");

Statement sm = conn.createStatement();

/*Thêm thông tin vào bảng students */
sm.executeUpdate("Insert into students
values(9,\"NVQuan\",8,7,9,(8+7+9)/3)");
sm.executeUpdate("Insert into students
values(10,\"PVThuan\",8,5,9,(8+5+9)/3)");

/*Truy vấn thông tin từ bảng students */
ResultSet rs=sm.executeQuery("Select * from students");

/*Tạo ResultSetMetaData để lấy thông tin số cột của ResultSet*/
ResultSetMetaData rsm=rs.getMetaData();
int col_num = rsm.getColumnCount();

/*Lấy tên cột của bảng students từ đối tượng rsm */
for (int i=1;i<=col_num;i++)
System.out.print(rsm.getColumnLabel(i)+ "\t");
System.out.println("");

/*Hiển thị thông tin truy vấn được từ bảng students */
while (rs.next())
{

/*Mỗi hàng của ResultSet duyệt các ô để lấy dữ liệu*/
for (int i=1;i<=col_num;i++)
System.out.print(rs.getString(i)+ "\t");

/*Xuống dòng mới*/
System.out.println("");

}
} catch( Exception e )
{
System.out.println( e );
}
}
}

```

Kết quả chương trình:

ID	Name	Math	Phys	Chem	Aver
1	NVQuan	8	7	9	8
2	PVThuan	8	5	9	7.33333

4. Viết chương trình truy vấn và hiển thị tên tất cả các bảng (tables) hiện có của một cơ sở dữ liệu.

```

import java.sql.*;

class TableNames
{
    public static void main( String args[] )
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver");

            /*Kết nối đến csdl có tên mysql*/
            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/mysql","root",
            "");

            /*Tạo đối tượng DatabaseMetaData để Lấy thông tin của csdl*/

            DatabaseMetaData dbm= conn.getMetaData();

            /*Lấy tên các bảng của csdl*/

            ResultSet rs = dbm.getTables(null, null, "%", null);

            /*Cột thứ 3 chứa tên bảng */

            while (rs.next()) {
                System.out.println(rs.getString(3));
            }

            } catch( Exception e )
            {
                System.out.println( e ) ;
            }
        }
    }
}

```

Kết quả chương trình:

```
Problems @ Javadoc Declaration Console Console
<terminated> TableNames [Java Application] C:\Program Files\Java\jdk1.7.0_45\jre\bin\javaw.exe (Nov 13, 2013, 9:29:55 PM)
columns_priv
db
event
func
general_log
help_category
help_keyword
help_relation
help_topic
innodb_index_stats
innodb_table_stats
ndb_binlog_index
plugin
proc
```

5. Sử dụng PreparedStatement, viết chương trình thêm thông tin vào bảng students của cơ sở dữ liệu StudentManagement.

```
import java.sql.*;

public class PreparedStatementDemo {

    public static void main(String[] args) {

        /*Khởi tạo dữ liệu để cập nhật*/
        String[] names= {"Tran C","Le D", "Ha E"};
        float[] Math={7,8,7};
        float[] Phys={8,9,8};
        float[] Chem={6,7,9};

        try
        {

            /*Nạp driver */
            Class.forName("com.mysql.jdbc.Driver");

            /*Kết nối đến cơ sở dữ liệu */
            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/studentmanagem
            ent","root", "");

            /*Tạo đối tượng PreparedStatement để gửi câu lệnh sql*/
            PreparedStatement ps = conn.prepareStatement("Insert into
            students (Name, Math, Phys, Chem, Aver) values(?,?,?,?,?)");

            /* Thiết lập giá trị cho các đối số (dấu ?) của PreparedStatement*/

            for (int i=0;i<names.length;i++)
            {
                ps.setString(1,names[i]);
                ps.setFloat(2,Math[i]);
                ps.setFloat(3,Phys[i]);
                ps.setFloat(4,Chem[i]);
                ps.setFloat(5,(Math[i]+Phys[i]+Chem[i])/3);

                /* Gọi hàm ra lệnh cập nhật dữ liệu*/
                ps.executeUpdate();
            }
        }
    }
}
```

```

/*Truy vấn thông tin từ bảng students */
    ResultSet rs=ps.executeQuery("Select * from students");

    /*Tạo ResultSetMetaData để lấy thông tin số cột của ResultSet*/
    ResultSetMetaData rsm=rs.getMetaData();
    int col_num = rsm.getColumnCount();

    /*Lấy tên cột của bảng students từ đối tượng rsm*/
    for (int i=1;i<=col_num;i++)
        System.out.print(rsm.getColumnLabel(i)+ "\t");
        System.out.println("");

    /*Hiển thị thông tin truy vấn được từ bảng students*/
    while (rs.next())
    {
        for (int i=1;i<=col_num;i++)
            System.out.print(rs.getString(i)+ "\t");
            System.out.println("");

        }

        ps.close();
        conn.close();
    } catch( Exception e )
    {
        System.out.println( e ) ;
    }

}

}

```

Kết quả chương trình:

```

<terminated> PreparedStatementDemo [Java Application] C:\Program Files\Java\jdk1.7.0_45\jre\bin\javaw.exe (Nov 15, 2013, 9:19:24 PM)
ID   Name  Math  Phys  Chem  Aver
13   Tran C  7     8     6     7
14   Le D   8     9     7     8
15   Ha E   7     8     9     8

```

- Sử dụng chế độ gửi (commit) yêu cầu thực thi các câu lệnh SQL đến cơ sở dữ liệu một cách không tự động để chương trình ở bài tập 5 chạy hiệu quả hơn.

```

import java.sql.*;
public class PreparedStatementDemo {
    public static void main(String[] args) {

        String[] names= {"Tran C", "Le D", "Ha E"};
        float[] Math={7,8,7};
        float[] Phys={8,9,8};
    }
}

```



```

float[] Chem={6,7,9};

try
{

    //Nạp driver
    Class.forName("com.mysql.jdbc.Driver");

    /*Kết nối đến cơ sở dữ liệu */
    Connection conn =
    DriverManager.getConnection("jdbc:mysql://localhost:3306/studentmanagem
ent","root", "");

    /*Tạo đối tượng PreparedStatement */
    PreparedStatement ps = conn.prepareStatement("Insert into students
(Name, Math, Phys, Chem, Aver) values(?,?,?,?,?)");

    /* Thiết Lập chế độ gửi đến csdl không tự động*/
    conn.setAutoCommit(false);

    /* Thiết Lập giá trị cho các đối số (dấu ?) của PreparedStatement*/

    for (int i=0;i<names.length;i++)
    {
        ps.setString(1,names[i]);
        ps.setFloat(2,Math[i]);
        ps.setFloat(3,Phys[i]);
        ps.setFloat(4,Chem[i]);
        ps.setFloat(5,(Math[i]+Phys[i]+Chem[i])/3);

    }

    /* Ra Lệnh thực thi cập nhật dữ liệu*/
    ps.executeUpdate();

    }

    /*Thực hiện gửi đến csdl */
    conn.commit();

    /*Truy vấn thông tin từ bảng students*/
    ResultSet rs=ps.executeQuery("Select * from students");

    /*Tạo ResultSetMetaData để Lấy thông tin số cột của ResultSet*/
    ResultSetMetaData rsm=rs.getMetaData();
    int col_num = rsm.getColumnCount();

    /*Lấy tên cột của bảng students từ đối tượng rsm*/
    for (int i=1;i<=col_num;i++)
        System.out.print(rsm.getColumnLabel(i)+ "\t");
        System.out.println("");

    /*Hiển thị thông tin truy vấn được từ bảng students*/
    while (rs.next())
    {
        for (int i=1;i<=col_num;i++)
            System.out.print(rs.getString(i)+ "\t");
            System.out.println("");
    }

```

```

    }
        ps.close();
        conn.close();
    } catch( Exception e )
    {
        System.out.println( e ) ;
    }

}
}

```

7. Hãy truy vấn và hiển thị thông tin bảng students của cơ sở dữ liệu StudentManagement lên giao diện của chương trình (Gợi ý: Sử dụng lớp javax.swing.JTable để chứa dữ liệu)

```

import java.sql.*;
import java.util.*;
import javax.swing.*;

public class ViewStudents {
    public static void main(String[] args) {
        Vector vData=null, vTitle=null;
        try{

            //Nạp driver
            Class.forName("com.mysql.jdbc.Driver");

            /*Kết nối đến cơ sở dữ liệu và tạo đối tượng Statement*/
            Connection conn =
            DriverManager.getConnection("jdbc:mysql://localhost:3306/studentmanagement
            ","root", "");
            Statement stm = conn.createStatement();

            ResultSet rst = stm.executeQuery("select * from students");
            ResultSetMetaData rstmeta = rst.getMetaData();
            int num_column = rstmeta.getColumnCount();

            /*Chuẩn bị dữ liệu để tạo bảng (JTable) hiển thị thông tin
            vTitle chứa đựng thông tin tên cột của bảng */

            vTitle = new Vector(num_column);
            for (int i=1; i<=num_column;i++)
            {
                vTitle.add(rstmeta.getColumnLabel(i));
            }

            /*vData chứa nội dung của bảng */
            vData = new Vector(10,10);
            while (rst.next())
            {
                Vector row = new Vector(num_column);
                for (int i=1; i<=num_column;i++)
                    row.add(rst.getString(i));
            }
        }
    }
}

```

```

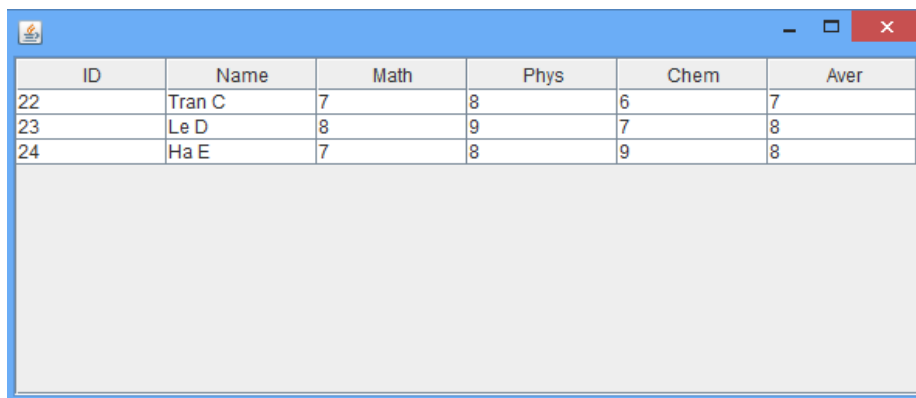
        vData.add(row);
    }
    rst.close();
    stm.close();
    conn.close();
} catch (Exception e)
{
    System.out.println(e.getMessage());
}

    /*tạo bảng để chứa thông tin truy vấn từ csdl*/
    JScrollPane tableResult = new JScrollPane(new
JTable(vData,vTitle));

    /*tạo cửa sổ chứa bảng dữ liệu*/
    JFrame f = new JFrame();
    f.setSize(600,480);
    f.setContentPane(tableResult);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.show();
}
}

```

Kết quả chương trình:



ID	Name	Math	Phys	Chem	Aver
22	Tran C	7	8	6	7
23	Le D	8	9	7	8
24	Ha E	7	8	9	8

8. Viết chương trình quản lý sinh viên gồm các chức năng cơ bản

- Xem danh sách sinh viên,
- Thêm mới, hiệu chỉnh và xóa một sinh viên khỏi danh sách

```

import java.sql.*;
import java.util.*;
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
import javax.swing.table.DefaultTableModel;

public class StudentManagement extends JFrame implements ActionListener,
MouseListener
{
    Connection conn;

```

```

Statement stm;
ResultSet rst;

/*tạo bảng để chứa thông tin truy vấn từ csdl*/
Vector vData=new Vector();
Vector vTitle=new Vector();
JScrollPane tableResult;
DefaultTableModel model;
JTable tb= new JTable();

/*các nút thao tác với dữ liệu*/
JButton edit, delete, insert;

/*vị trí hàng đã chọn ở bảng dữ liệu*/
int selectedrow=0;

public StudentManagement(String s)
{
    super(s);

    try{

        //Nạp driver
        Class.forName("com.mysql.jdbc.Driver");

        /*Kết nối đến cơ sở dữ liệu và tạo đối tượng Statement */
        conn =
        DriverManager.getConnection("jdbc:mysql://localhost:3306/studentm
        anagement","root", "");
        stm = conn.createStatement();

        /*Tạo p để nhóm các nút và đặt vào giao diện chính*/
        JPanel p= new JPanel();

        /*Tạo các nút và găng ống nghe*/
        edit = new JButton("Edit");
        edit.addActionListener(this);

        delete = new JButton("Delete");
        delete.addActionListener(this);

        insert = new JButton("Insert");
        insert.addActionListener(this);

        /*Đặt các nút vào JPanel p*/
        p.add(edit);
        p.add(delete);
        p.add(insert);

        /*Đặt p vào vùng dưới cùng của cửa sổ chính*/
        this.add(p, "South");

        /*Nạp dữ liệu vào 2 Vector: vTitle (tên cột) và vData (các
        hàng dữ liệu) chuẩn bị tạo JTable */
        reload();

        /*tạo bảng hiển thị thông tin lên cửa sổ*/
        model = new DefaultTableModel(vData,vTitle);

```

```

        tb=new JTable(model);

        /*Gắng ống nghe khi ấn chuột chọn hàng*/
        tb.addMouseListener(this);

        /*tạo đối tượng JScrollPane để chứa bảng*/
        tableResult = new JScrollPane(tb);

        /*Đặt bảng ở vùng trên của cửa sổ*/
        this.getContentPane().add(tableResult,"North");

        /*Thiết lập kích thước, vị trí và hiển thị cửa sổ*/
        this.setSize(400,300);
        this.setLocation(200, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setVisible(true);
    }catch(Exception e)
    {
        System.out.println(e.getMessage());
    }
}

/*Đọc dữ liệu từ bảng students và điền dữ liệu vào 2 Vector: vTitle và vData
để chuẩn bị xây dựng đối tượng JTable */
public void reload()
{
    try{

        /*Xóa hết dữ liệu hiện có trong 2 vector*/
        vTitle.clear();
        vData.clear();

        /*truy vấn dữ liệu từ bảng students*/
        ResultSet rst = stm.executeQuery("select * from students");

        /*Tạo đối tượng rstmeta để lấy thông tin của ResultSet*/
        ResultSetMetaData rstmeta = rst.getMetaData();
        int num_column = rstmeta.getColumnCount();

        /*Chuẩn bị dữ liệu để tạo bảng (JTable)
        Tạo các tên cột cho bảng*/
        for (int i=1; i<=num_column;i++)
        {
            vTitle.add(rstmeta.getColumnLabel(i));
        }

        /*Tạo dữ liệu các hàng cho bảng:
        * mỗi phần tử của Vector vData là một Vector */
        while (rst.next())
        {
            Vector row = new Vector(num_column);
            for (int i=1; i<=num_column;i++)
                row.add(rst.getString(i));
            vData.add(row);
        }
        rst.close();
    }
}

```

```

        }catch(Exception e)
        {
            System.out.println(e.getMessage());
        }

    }

    /* Xóa dữ liệu khi người dùng chọn hàng và ấn nút Delete */
    public void delete()
    {
        try{

            /*Lấy nội dung hàng đã chọn */
            Vector st = (Vector)vData.elementAt(selectedrow);

            /*tạo câu lệnh SQL và xóa dữ liệu khỏi bảng students trong csdl*/
            String sql = "Delete from students where id =
            \""+st.elementAt(0)+"\"";
            stm.executeUpdate(sql);

            /*Xóa nội dung hàng tương ứng trong vData*/
            vData.remove(selectedrow);

            /*Cập nhật lại nội dung bảng hiển thị trên màn hình*/
            model.fireTableDataChanged();

        }catch(Exception e)
        {
            e.printStackTrace();
        }
    }

    /*Xử lý khi người dùng ấn các nút trên cửa sổ */
    public void actionPerformed(ActionEvent e)
    {

        //xử lý khi ấn nút Delete
        if(e.getActionCommand().equals("Delete"))
        {
            delete();
        }

        //xử lý khi ấn nút Insert
        if(e.getActionCommand().equals("Insert"))
        {

            //Tạo cửa sổ để nhập mới
            new UpdateForm("Insert form",this,"","","0","0","0");
        }

        //xử lý khi ấn nút Edit
        if(e.getActionCommand().equals("Edit"))
        {

            //Lấy nội dung hàng đã chọn
            Vector st = (Vector)vData.elementAt(selectedrow);

            //Tạo cửa sổ để hiệu chỉnh hàng đã chọn

```

```

        new UpdateForm("Edit
form", this, (String)st.elementAt(0), (String)st.elementAt(1), (String)st.ele
mentAt(2), (String)st.elementAt(3), (String)st.elementAt(4));
    }
}

/*Lấy vị trí hàng ở bảng JTable khi người dùng ấn chuột
*/
public void mouseClicked(MouseEvent e)
{
    selectedrow = tb.getSelectedRow();
}
public void mouseEntered(MouseEvent e)
{}
public void mouseExited(MouseEvent e)
{}
public void mousePressed(MouseEvent e)
{}
public void mouseReleased(MouseEvent e)
{}

    public static void main(String[] args) {
        new StudentManagement("Student Management");
    }
}

/*Xây dựng cửa sổ nhập mới và hiệu chỉnh dữ liệu */
class UpdateForm extends JFrame implements ActionListener
{
    //các thành phần trên giao diện
    JLabel nameLabel;
    JTextField name;
    JLabel mathlb;
    JTextField math;
    JLabel physlb;
    JTextField phys;
    JLabel chemlb;
    JTextField chem;

    /*dùng để hiển thị lỗi khi người dùng nhập dữ liệu sai */
    JLabel errorlb;
    JLabel errordetails;

    /*Hai nút để xác nhận và hủy việc cập nhật thông tin */
    JButton ok;
    JButton cancel;

    /*nhận đối tượng của sổ chính truyền đến*/
    StudentManagement mst;

    /*nhận id của một record của bảng students khi hiệu chỉnh*/
    String id;

    public UpdateForm(String s, StudentManagement st, String i, String na,
String m, String ph, String ch)
    {
        super(s);
    }
}

```

```

/*nhận đối tượng cửa sổ chính truyền đến để gọi hàm
   cập nhật dữ liệu giao diện cửa sổ chính và cập nhật csdl
   */
mst=st;

/*tạo giao diện để cập nhật thông tin */
Container cont = this.getContentPane();
cont.setLayout(new GridLayout(6,2));

name1b = new JLabel("Name");
name = new JTextField(na);
cont.add(name1b);
cont.add(name);

math1b= new JLabel("Math");
math= new JTextField(m);
cont.add(math1b);
cont.add(math);

phys1b= new JLabel("Physics");
phys= new JTextField(ph);
cont.add(phys1b);
cont.add(phys);

chem1b= new JLabel("Chemistry");
chem= new JTextField(ch);
cont.add(chem1b);
cont.add(chem);

error1b= new JLabel("");
errordetails= new JLabel("");
error1b.setVisible(false);
errordetails.setVisible(false);
cont.add(error1b);
cont.add(errordetails);

JButton ok = new JButton("Ok");
JButton cancel = new JButton("Cancel");
cont.add(ok);
cont.add(cancel);
ok.addActionListener(this);
cancel.addActionListener(this);

this.setSize(230,200);
this.setLocation(250, 100);
setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
this.setVisible(true);

/*nhận id được truyền đến từ cửa sổ chính
   * để phục vụ việc cập nhật thông tin vào csdl */
id=i;
}

/* Gọi khi ấn nút Ok hoặc Cancel */
public void actionPerformed(ActionEvent e)
{
    if(e.getActionCommand().equals("Ok"))
    {

```



```

        //gọi hàm cập nhật dữ liệu
        insertDB();
    } else this.dispose();
    //Tắt cửa sổ khi ấn Cancel
}

/* Cập nhật csdl dữ liệu và giao diện cửa sổ chính */
public void insertDB()
{
    /*Nếu có một trường không có dữ liệu thì báo lỗi */
    if
    (name.getText().equals("")||math.getText().equals("")||phys.getText().equals("")||chem.getText().equals(""))
    {
        /*Tạo nội dung Lỗi */
        errorlb.setText("Error");
        errordetails.setText("empty value");
        errorlb.setForeground(Color.RED);
        errordetails.setForeground(Color.RED);

        /*Hiển thị Lỗi*/
        errorlb.setVisible(true);
        errordetails.setVisible(true);
    }
    else
    {
        try{
            //Lấy nội dung đã nhập ở giao diện
            String na = name.getText();
            float m =Float.parseFloat(math.getText());
            float ph =Float.parseFloat(phys.getText());
            float ch =Float.parseFloat(chem.getText());

            String sql="";

            //Nếu Là nhập mới
            if (this.getTitle().equals("Insert form"))
                sql ="insert into students(Name,Math,Phys,Chem,Aver)
values (\"+na+\" \",\"+m+\", \"+ph+\", \"+ch+\", \"+(m+ph+ch)/3+\"";

            else //Nếu Là hiệu chỉnh
                sql ="update students set
Name=\"+na+\" \",Math=\"+m+\",Phys=\"+ph+\",Chem=\"+ch+\",Aver=\"+(m+ph+ch)/3+\"
where ID=\"+id+\"";

            //cập nhật vào csld
            mst.stm.executeUpdate(sql);

            //cập nhật giao diện cửa sổ chính
            mst.reload();
            mst.model.fireTableDataChanged();

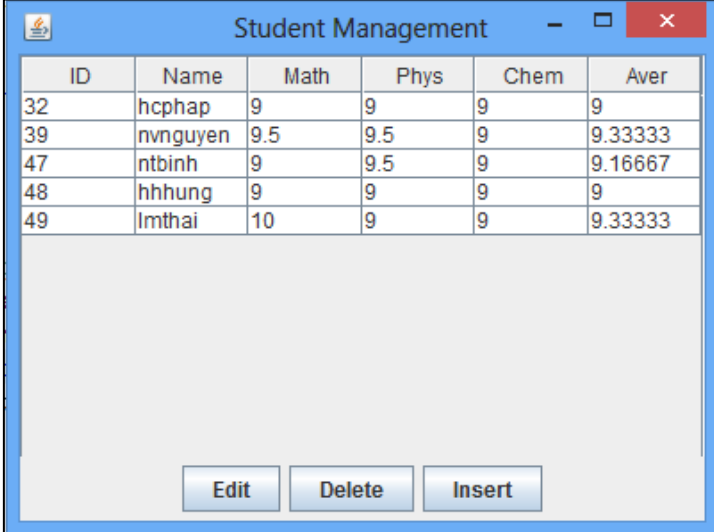
            //tắt cửa sổ
            this.dispose();
        }
    }
}

```

```
    }catch(Exception ex)
    {
        ex.printStackTrace();
    }
}
}
```

Kết quả chương trình:

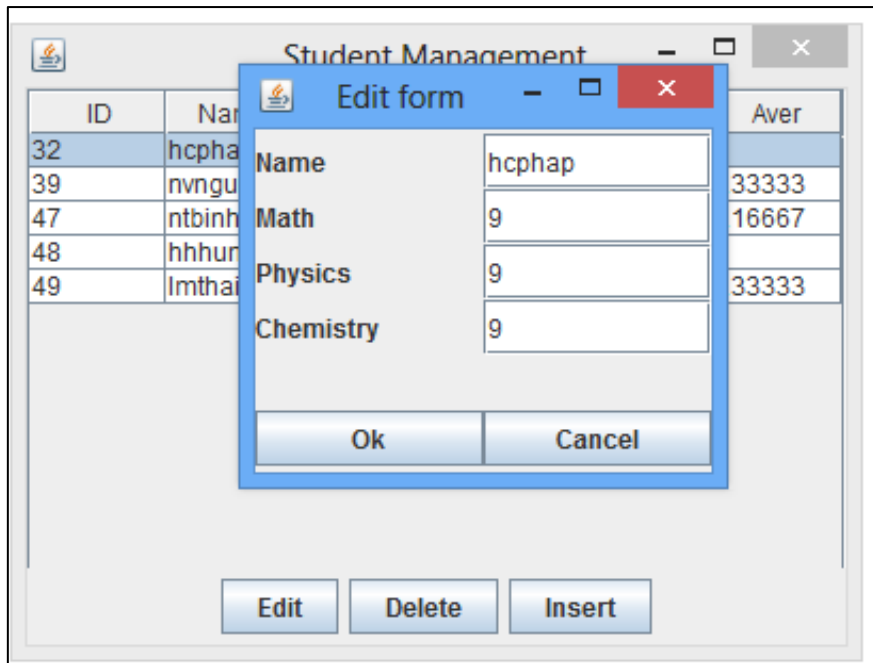
Cửa sổ chính chương trình



ID	Name	Math	Phys	Chem	Aver
32	hcpchap	9	9	9	9
39	nvnguyen	9.5	9.5	9	9.33333
47	ntbinh	9	9.5	9	9.16667
48	hhhung	9	9	9	9
49	lmthai	10	9	9	9.33333

Edit Delete Insert

Cửa sổ nhập mới và hiệu chỉnh dữ liệu



III. Bài tập tự giải

- Viết chương trình kết nối đến cơ sở dữ liệu, hiển thị tất cả các tên bảng của cơ sở dữ liệu đang kết nối trong một danh sách (JComboBox). Khi người dùng chọn một bảng trong danh sách, chương trình hiển thị nội dung bảng đã chọn (Sử dụng JTable).
- Viết chương trình tạo một bảng cấu hình có một nút tên "Config" và 4 Textfield cho phép người dùng nhập thông tin kết nối cơ sở dữ liệu gồm Username, Password, Database URL và Driver. Khi người dùng nhập đầy đủ thông tin vào bảng cấu hình và ấn nút "Config" chương trình sẽ lưu trữ thông tin cấu hình xuống tệp tin đồng thời hiển thị tất cả các tên bảng của cơ sở dữ liệu tương ứng vào danh sách (JComboBox) ở giao diện để người dùng truy vấn thông tin các bảng của cơ sở dữ liệu như bài tập 1.
- Tạo thêm một bảng Accounts trong cơ sở dữ liệu StudentManagement gồm các trường sau: username, password, access_rights (biểu diễn quyền truy cập hệ thống gồm 1 trong 3 giá trị Admin, Teacher, Student).

Viết chương trình tạo một cửa sổ cho phép đăng ký (lưu dữ liệu xuống bảng Accounts) và một cửa sổ đăng nhập thông tin bằng cách kiểm tra dữ liệu ở bảng Accounts.

4. Viết chương trình giao diện cho phép thực hiện các thao tác xem, xóa, sửa dữ liệu ở bảng Accounts của cơ sở dữ liệu StudentManagement. (tham khảo bài 8 ở phần trên).
5. Viết chương trình tạo một cửa sổ tìm kiếm thông tin sinh viên theo tên. Chương trình kết nối với cơ sở dữ liệu StudentManagement và truy xuất dữ liệu ở bảng Students để phục vụ tìm kiếm thông tin.
6. Viết chương trình tạo một cửa sổ hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm trung bình. Chương trình kết nối với cơ sở dữ liệu StudentManagement và truy xuất dữ liệu ở bảng Students để phục vụ sắp xếp và hiển thị thông tin.
7. Cải tiến chức năng thêm mới và hiệu chỉnh của bài tập 8 ở phần Bài tập thực hành mẫu của chương này bằng cách sử dụng PreparedStatement thay cho Statement.
8. Tiếp tục cải tiến bài tập 8 ở phần Bài tập thực hành mẫu của chương này bằng cách thêm vào cửa sổ chính:
 - Một thanh Menu chứa mục System gồm 2 mục con: Accounts, About. Mục Accounts có 3 mục con Sign up, Sign in, User accounts.
 - Hai nút (Jbutton) có tên "Search" và "Sort and View" ngay sau nút "Insert".

Chương trình cho phép thực hiện thêm các thao tác sau:

- Khi người dùng chọn mục menu "Sign up", chương trình sẽ gọi và mở cửa sổ đăng ký tài khoản người dùng ở bài tập 3.
- Khi người dùng chọn mục menu "Sign in", chương trình sẽ gọi và mở cửa sổ đăng nhập ở bài tập 3.
- Khi người dùng chọn mục menu "User accounts", chương trình sẽ gọi và mở cửa sổ cho phép xem, xóa, sửa dữ liệu ở bảng Accounts ở bài tập 4.
- Khi người dùng chọn mục menu "About", chương trình sẽ mở cửa sổ hiển thị thông tin về tác giả, chương trình...
- Khi người dùng chọn nút "Search", chương trình sẽ gọi và mở cửa sổ tìm kiếm sinh viên theo tên ở bài tập 5.
- Khi người dùng chọn nút "Sort and View", chương trình sẽ gọi và mở cửa sổ hiển thị danh sách sinh viên theo thứ tự tăng dần của điểm trung bình ở bài tập 6.

Lưu ý: Tùy vào quyền truy cập của mỗi người dùng đăng nhập vào hệ thống mà chương trình cho phép thực hiện các thao tác khác nhau, cụ thể nếu giá trị `access_rights` của người đăng nhập là:

- Admin: Thực hiện tất cả các thao tác
- Teacher: Thực hiện tất cả các thao tác với dữ liệu của bảng `students` và chỉ một số thao tác với dữ liệu bảng `accounts` gồm đăng ký, đăng nhập, xem.
- Student: Thực hiện chỉ một số thao tác với dữ liệu của bảng `students` gồm xem, tìm kiếm và một số thao tác với dữ liệu bảng `Accounts` gồm đăng ký, đăng nhập, xem.