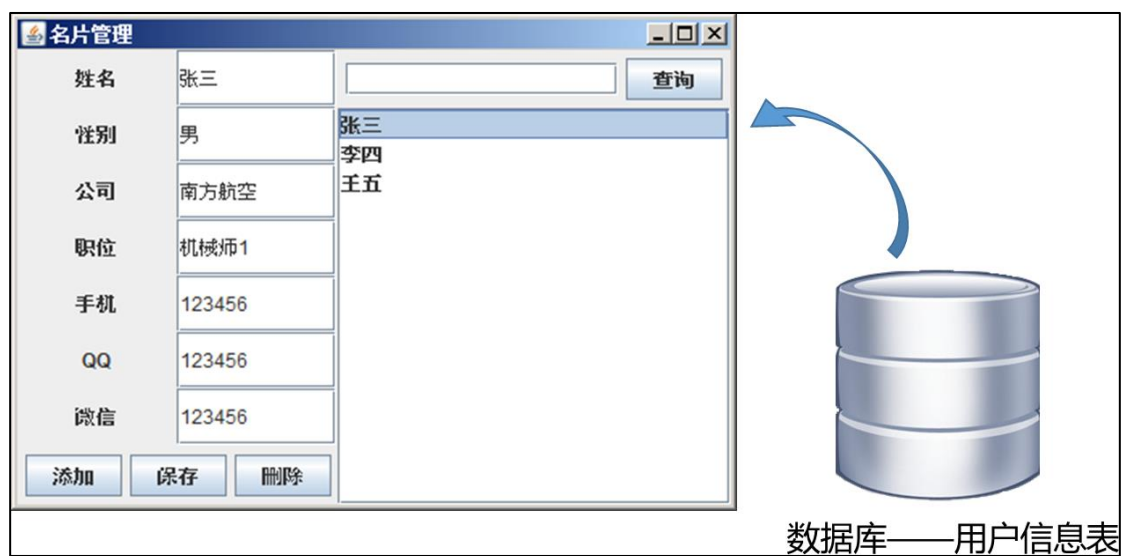


学习任务四 通讯录管理系统开发

子任务 4.2 使用 JDBC 处理数据库数据

1、任务引入：

上个任务完成的通讯录管理界面要访问的数据来自数据库，如下图。

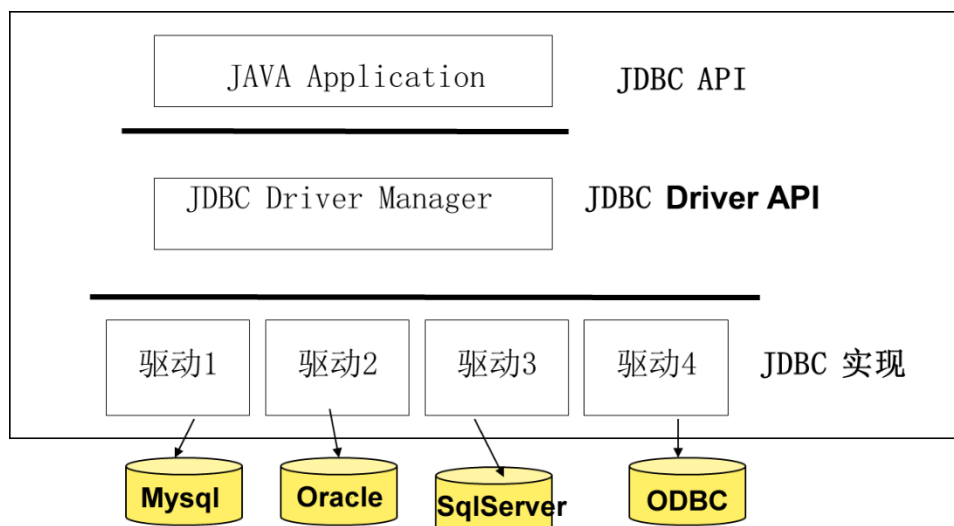


所以，我们要先学习如何使用 Java 技术来访问数据库。

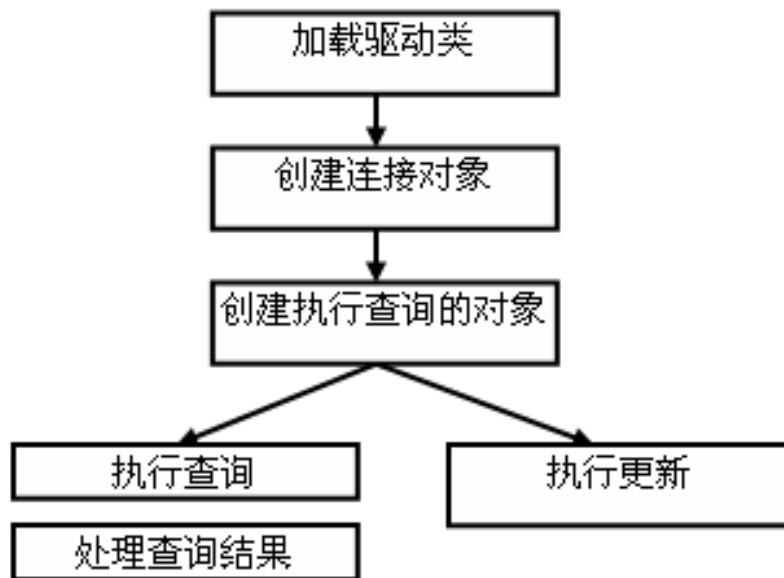
2、任务准备

1) 了解 JDBC 框架

Java 访问数据库，需要使用 JDBC 框架，该框架结构如下：



使用该框架访问数据库的流程是：



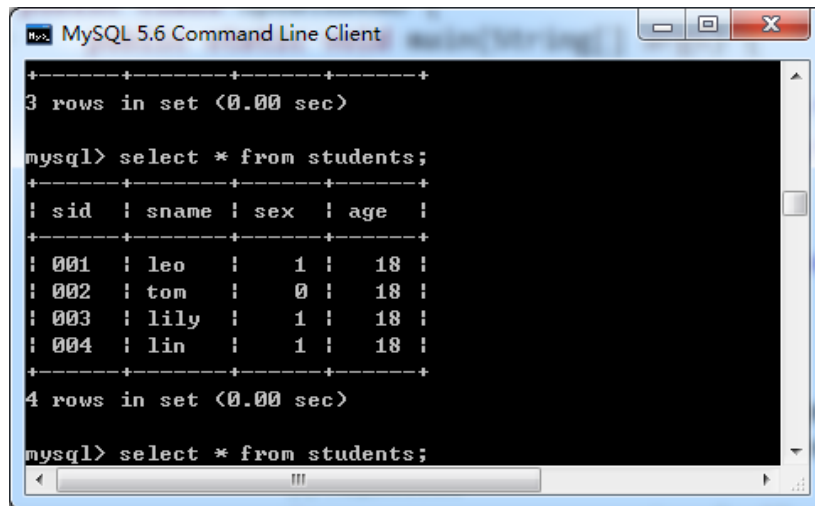
2) 使用 JDBC 增删改查数据库案例学习

第一步：在 scholl.sql 里面建立 card 表，在 Mysql 中执行 school.sql

*****scholl.sql*****

```
drop database school;
go
create database school;
go
use school;
go
create table students(
    sid varchar(8),
    sname varchar(50),
    sex int,
    age int
);
insert into students values('001','mary',1,18);
insert into students values('002','tom',0,18);
insert into students values('003','lily',1,18);
```

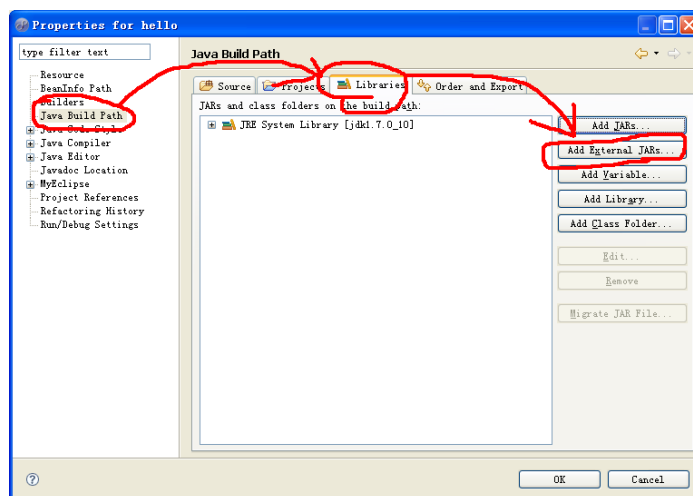
打开 mysql，如下图所示测试你的数据库是否成功创建：



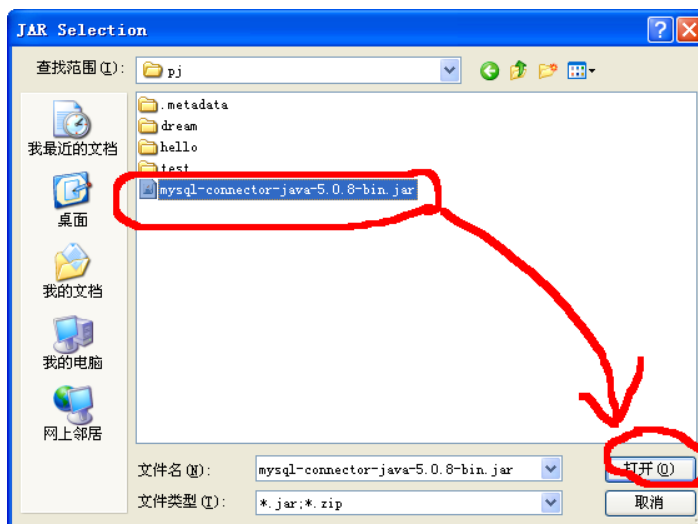
第二步：在 Java 项目中，添加外部 jar 包

下载 mysql-connector-java-5.0.8-bin.jar，并将该包倒入工程的资源，操作提示如下：

1) 右击工程名，点击“属性”也就是 properties



2) 在出现的窗口中，选择右边的 java build path，选择 libraries，选择 add external jars



3)在出现的对话框中选择你刚刚下载下来的 jar 包，刷新工程就可以了

第三步：在 Java 的 java project 工程下添加 UpdateDemo.java,实现使用 JDBC 修改数据库中数据表的功能，如下图：

```
public class UpdateDemo {
    public static void main(String[] args) {
        //*****
        String driver = "com.mysql.jdbc.Driver"; //*****
        String url = "jdbc:mysql://localhost:3306/school?useUnicode=true&characterEncoding=utf8";
        String uid = "root";
        String psd = "123456";
        String sql = "update students set sname='leo' where sid='001'"; //*****
        try {
            Class.forName(driver);
            //*****
            Connection cn = DriverManager.getConnection(url, uid, psd);
            Statement st = cn.createStatement();
            //*****
            int rows = st.executeUpdate(sql);
            if (rows > 0) {
                System.out.println("更新成功");
            } else {
                System.out.println("更新失败");
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

第四步：在 Java 的 java project 工程下添加 SelectDemo.java,实现使用 JDBC 查询数据库中数据表的功能，如下图：

```
package db;
import java.sql.*;
public class SelectDemo {
    public static void main(String args[]){
        //定义数据库连接必备的常量
        String driver="com.mysql.jdbc.Driver"; //驱动用哪一个
        String url="jdbc:mysql://localhost:3306/school";
        String uid="root";
        String psd="123456";
        String sql="select * from students"; //业务逻辑
        try {
            Class.forName(driver);
            //建立连接
            Connection cn=DriverManager.getConnection(url, uid, psd);
            //得到执行查询的对象
            Statement st=cn.createStatement();
            //执行查询获得结果
            ResultSet rs=st.executeQuery(sql);
            while(rs.next()){
                //从rs中逐条取出字段的值
                String sid=rs.getString("sid");
                System.out.println(sid);
            }
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

【思考】刚才的训练过程中，哪些代码是重复的？有没有方法可以简化代码？

- 1) 试着实现一个 DB 类，将连接数据库的重复代码放入这个 DB 类中,DB 类的结构如下所示

DB: 数据库操作类

```
conn
stmt
driver —— 数据库驱动
password —— 数据库密码
username —— 数据库用户
url —— 数据库连接地址

DB() —— 无参构造
DB(String, String, String, String) —— 有参构造
closeConnection() : void
executeQuery(String) : ResultSet —— 执行查询
executeUpdate(String) : boolean —— 执行一条更新
executeUpdate(String[]) : boolean —— 执行多条查询
```

- 2) 使用 DB 类实现对 student 数据表的访问，范例代码如下所示：

```
1 package db;
2 //代表一个学生信息的javaBean, 叫做实体类
3 public class Stu {
4     String sid;
5     String sname;
6     int sex;
7     int age;
8
9     @Override
10    public String toString() {
11        String s=(sex==0)?"男":"女";
12        return sid + "\t" + sname + "\t" + s + "\t" + age ;
13    }
14
15    public Stu(String sid, String sname, int sex, int age) {
16        super();
17        this.sid = sid;
18        this.sname = sname;
19        this.sex = sex;
20        this.age = age;
21    }
22
23    public Stu() {
24        super();
25        // TODO Auto-generated constructor stub
26    }
27 }
```

1、完成Stu类

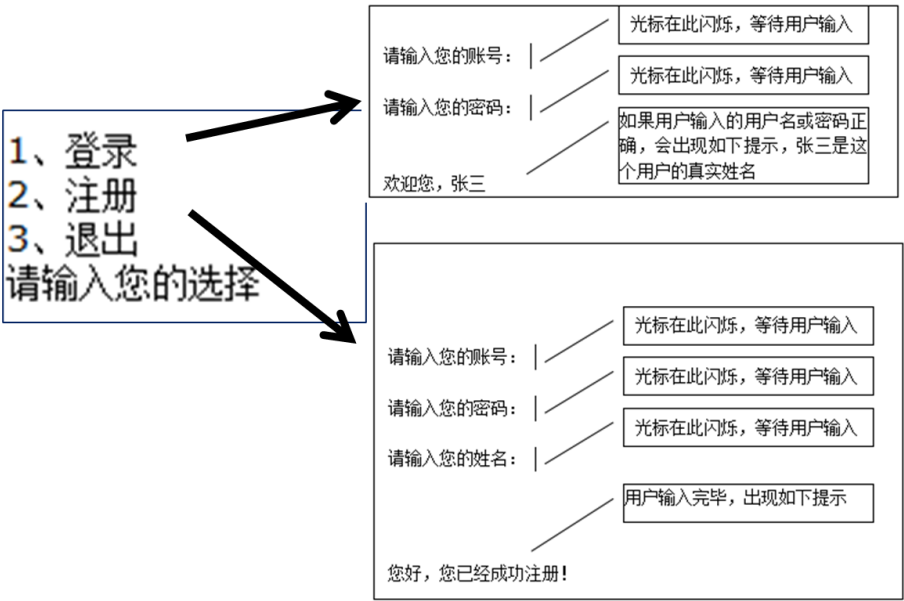
2、完成Stus类

```
1 package db;
2 //实现业务处理的类，访问数据库
3 public class Stus extends DB {
4     //注册一个新学生，参数是Stu对象，包含所有新学生的信息
5     public boolean newStu(Stu newstu){
6         String sql = "insert into students values('"+newstu.getSid()+"','"+
7             +newstu.getSname()+"','"+
8             +newstu.getSex()+"','"+
9             +newstu.getAge()+"')";
10        System.out.println("newStu"+sql); //测试你的sql语句有没有语法错误
11        return super.executeUpdate(sql);
12    }
13
14    public static void main(String[] args) {
15        ArrayList<Stu> temp=new Stus().getAllStus("y");
16        if(temp==null) {
17            System.out.println("没有符合条件的学生信息");
18        }else{
19            for(Stu t:temp){
20                System.out.println(t);
21            }
22        }
23    }
24
25    //修改学生
26    public boolean updateStu(Stu newstu){
27        String sql="update students set sname='"+newstu.getSname()+"',"+
28            "sex='"+newstu.getSex()+"',"+
29            "age='"+newstu.getAge()+"'"+
30            "where sid='"+newstu.getSid()+"'";
31        System.out.println("updateStu"+sql);
32        return super.executeUpdate(sql);
33    }
34
35    //删除学生
36    public boolean delStu(String sid){
37        String sql="delete from students where sid='"+sid+"'";
38        System.out.println("delStu"+sql);
39        return super.executeUpdate(sql);
40    }
41
42    //获取学生的信息
43    public ArrayList<Stu> getAllStus(String query){
44        String sql="select * from students";
45        if(query!=null && query.length()>0){ //用户给参数
46            sql+=" where sname like '"+query+"%'";
47        }
48        System.out.println("getAllStus++++"+sql);
49        ArrayList<Stu> temp=null; //准备符合条件的学生对象
50        ResultSet rs=super.executeQuery(sql);
51        try {
52            while(rs.next()){
53                if(temp==null) temp=new ArrayList<Stu>();
54                Stu stu=new Stu();
55                stu.setSid(rs.getString("sid"));
56                stu.setSname(rs.getString("sname"));
57                stu.setSex(rs.getInt("sex"));
58                stu.setAge(rs.getInt("age"));
59                temp.add(stu);
60            }
61        } catch (SQLException e) {
62            // TODO Auto-generated catch block
63            e.printStackTrace();
64        }
65        return temp;
66    }
67 }
```

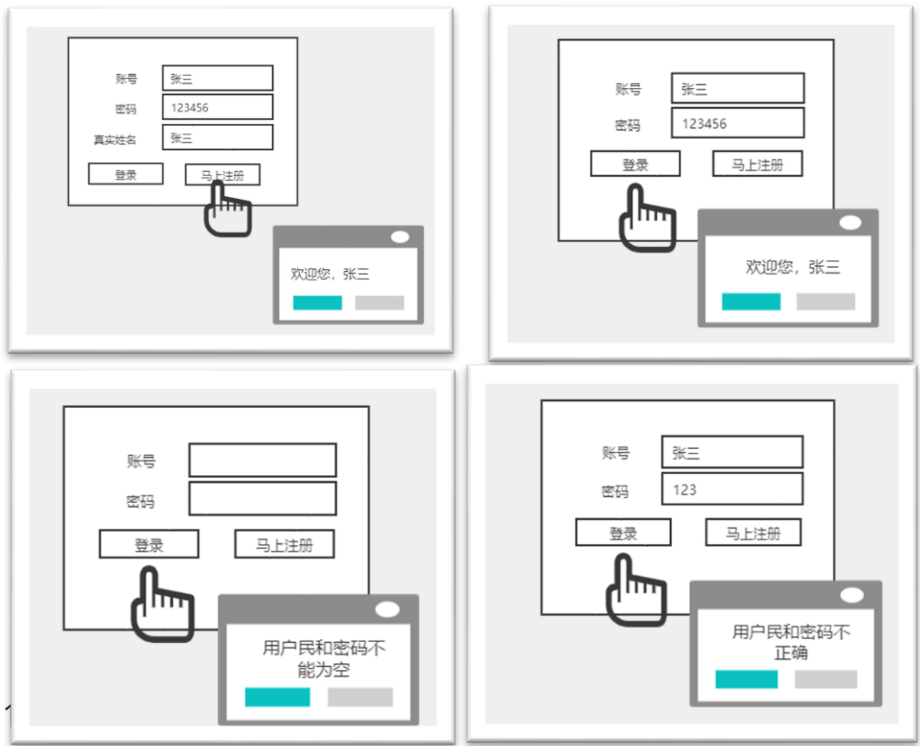
3、任务实现

【任务需求】使用 JDBC 完成如下功能：

界面需求一：



界面需求二：



在mysql中设计并创建他

表名	Usertable		
字段名	类型	约束	描述
id	自动编号	主键	流水号
Account	字符串		账号
realname	字符串		真实姓名
Password	字符串		密码

2) 设计一个菜单或者图形界面，实现注册和登录
