

数据库系统原理 课程设计报告

学生选课系统

班级： 19 计科三班

姓名： 周吉瑞

学号： 20190521340

任课教师： 秦海菲

目录

一、 学生选课系统开发的背景和目的.....	1
二、 学生选课系统需求分析.....	2
(一) 数据流图.....	2
三、 概要设计（系统设计）	5
(一) 功能模块图.....	5
(二) 实体属性图.....	6
(三) 学生选课系统 E-R 图.....	8
(四) 逻辑结构设计——ER 图向关系模型转换.....	10
(五) 表的设计.....	11
四、 学生选课系统登录界面.....	17
五、 数据库连接.....	21

一、学生选课系统开发的背景和目的

随着无纸化办公的普遍实现，信息的自动处理以及网络式的信息交互方式已经被人们广泛应用。让计算机来管理学生的信息是现在各个高校都在积极进行的工作之一，也是高校教学管理工作的重要内容之一。网上选课与传统的选课方式相比更加节约资源，增加了学生选课自主权。

学生选课系统作为一种现代化的教学技术，越来越受到人们的重视，是一个学校不可或缺的部分，学生选课系统就是为了管理好选课信息而设计的。学生选课系统的将使选课管理工作规范化、系统化、程序化，避免选课管理的随意性，提高信息处理的速度和准确性，能够准确、及时、有效的查询和修改学生选课情况。

与传统的选课方式相比，网上选课系统利用局域网为学生选课带来了极大的便捷。学生在公共机房，或者宿舍的个人电脑上便可以通过校园网络来选课。在选课期间内，学生能够使用选课系统灵活的修改自己的选课情况，大大提高了学校选课工作的效率。教务处的教师则可以通过选课系统的管理员子系统来管理学生的选课情况，使得学生选课工作达到系统化和自动化，大大提高了学校的工作效率，为广大师生及相关人员节省了极多的时间。

全校性选修课开设的目的在于扩大学生的知识面、加强学生素质教育、培养复合型高级人才，具有不可替代的重要性。随着教育的不断深入和素质教育的加强，学分制的实施，选修课在一个学生的培养计划中占的比重将越来越大。

网上选课系统的出现使同学们能够更加自主、便捷、准确的进行选课。但是，由于一般高校中的学生都比较多，因此带来了诸如信息管理等问题，鉴于需要将学生信息、选课信息等信息数字化以便于管理维护，我便想到了利用数据库能够比较良好地解决此类问题，由此下面我设计出一个高校选课系统以供参考。

二、学生选课系统需求分析

（一）数据流图

数据流图（Data Flow Diagram）：简称 DFD，它从数据传递和加工角度，以图形方式来表达系统的逻辑功能、数据在系统内部的逻辑流向和逻辑变换过程，是结构化系统分析方法的主要表达工具及用于表示软件模型的一种图示方法。

以下是学生选课系统的数据流图：

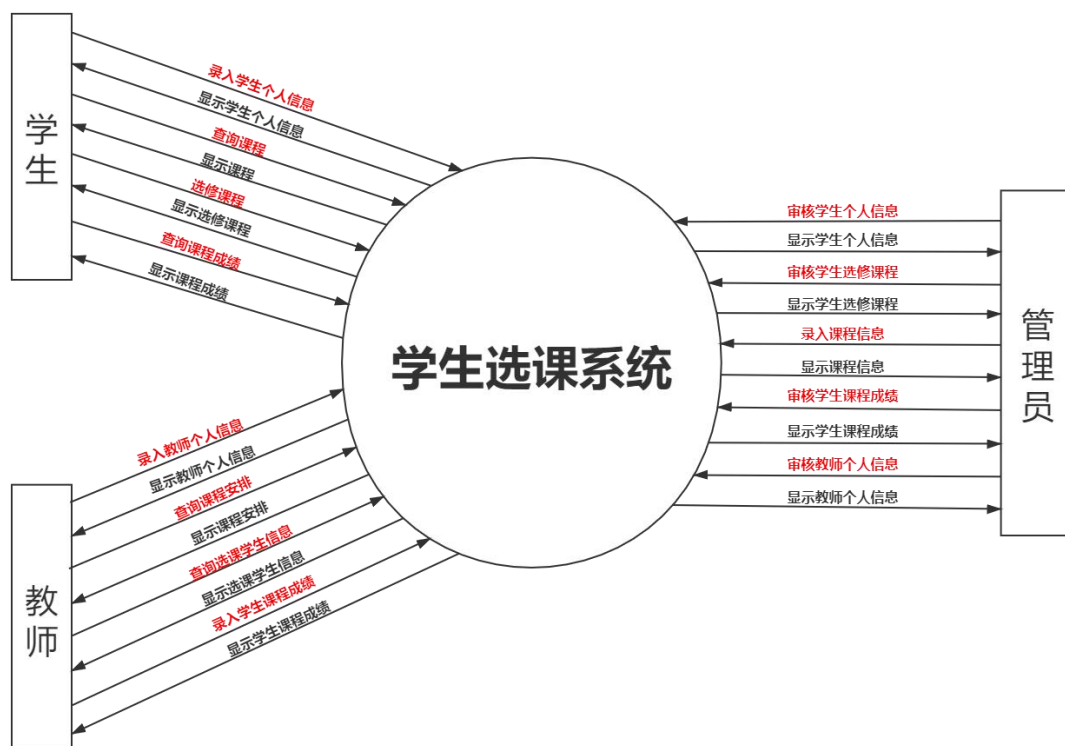
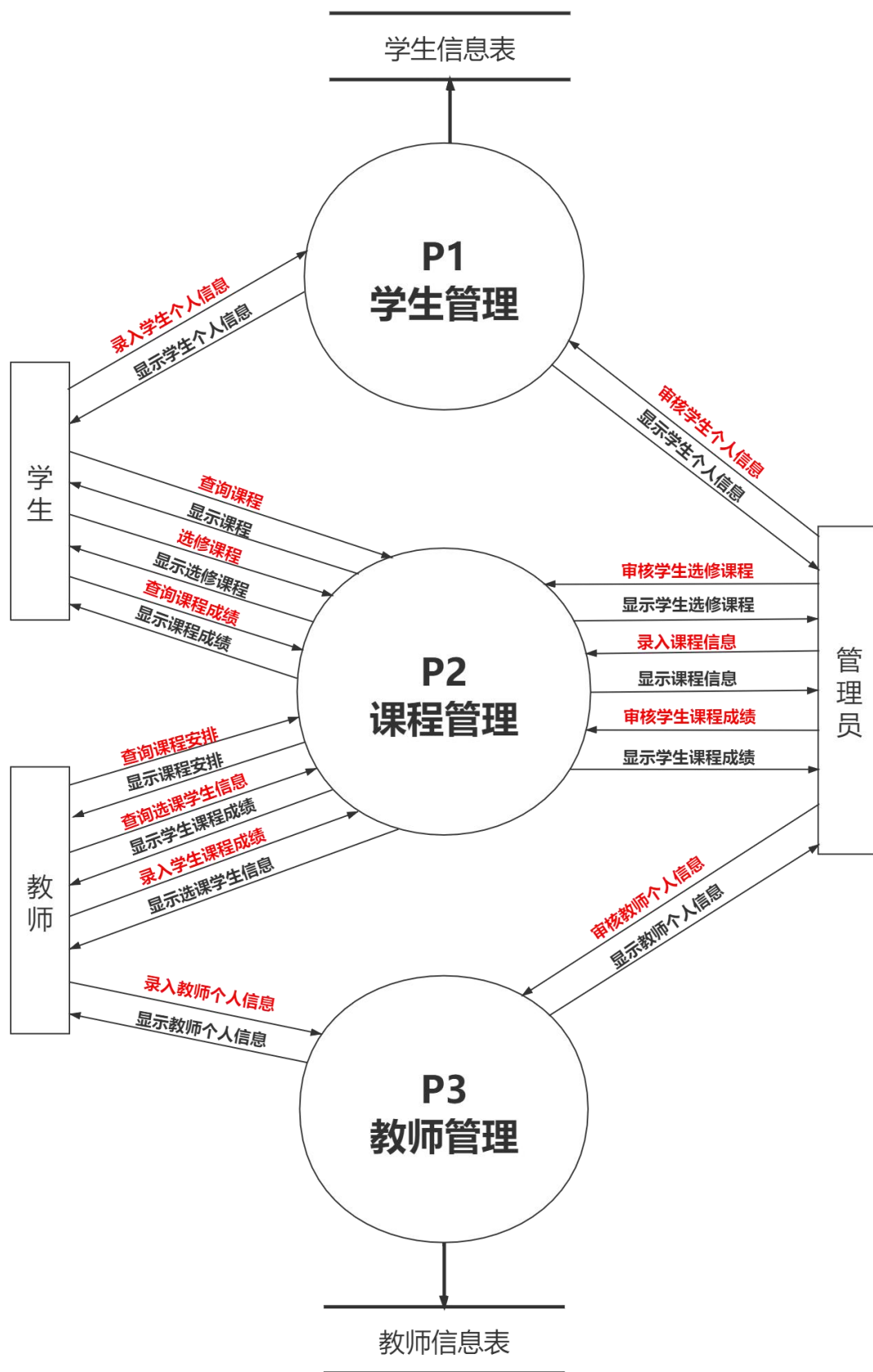


图 1：顶层数据流图

顶层数据流图描述了整个系统的作用范围，对系统的总体功能、输入和输出进行了抽象，反映了系统和环境的关系。



图二：0层数据流图

[illegible]

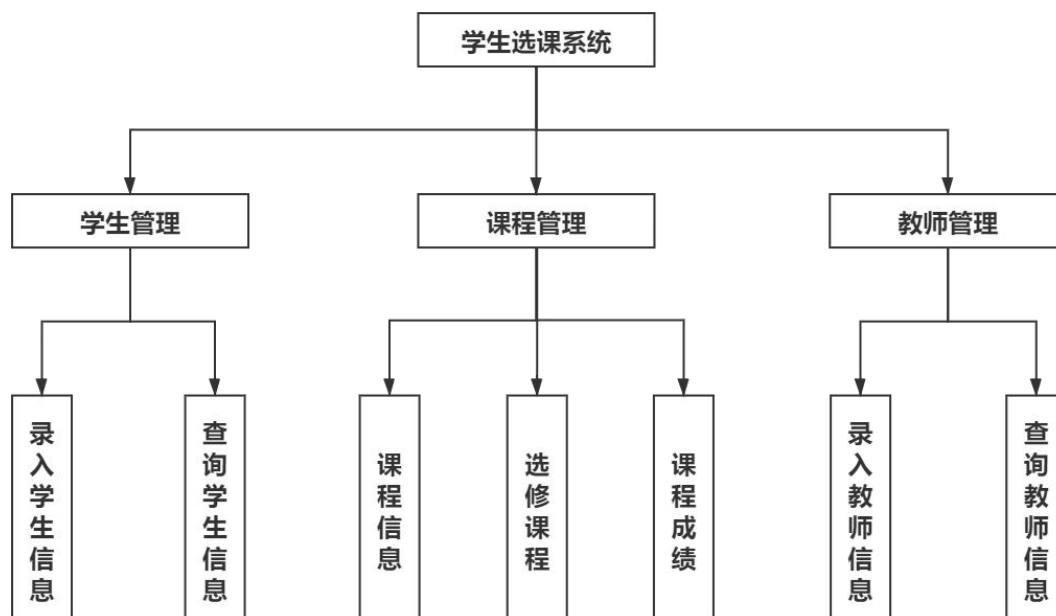
系统课程管理的百分

三、概要设计（系统设计）

（一）功能模块图

一般情况下产品或系统的总功能可分解为若干分功能，各分功能又可进一步分解为若干二级分功能，如此继续，直至各分功能被分解为功能单元为止。这种由分功能或功能单元按照其逻辑关系连成的结构称为功能结构。分功能或功能单元的相互关系可以用图来描述，表达分功能或功能单元相互关系或从属关系的图称为功能模块图。

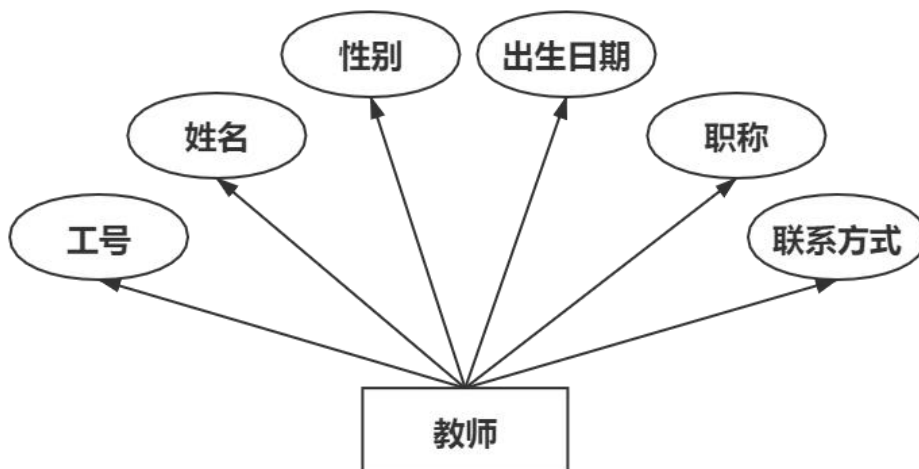
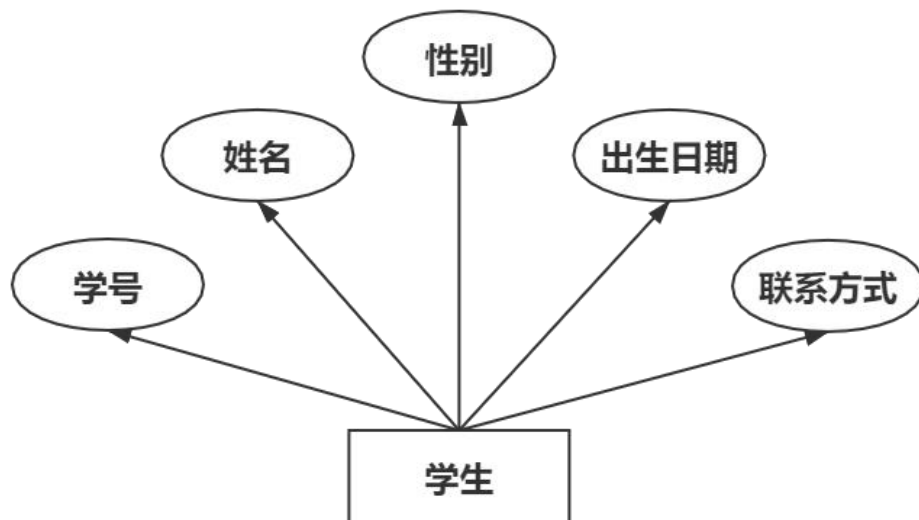
为了对学生选课系统有一个大概的了解，下面给出了学生选课系统的功能模块图，它包括：学生管理、课程管理、教师管理 4 个部分以及它每一部分的功能。

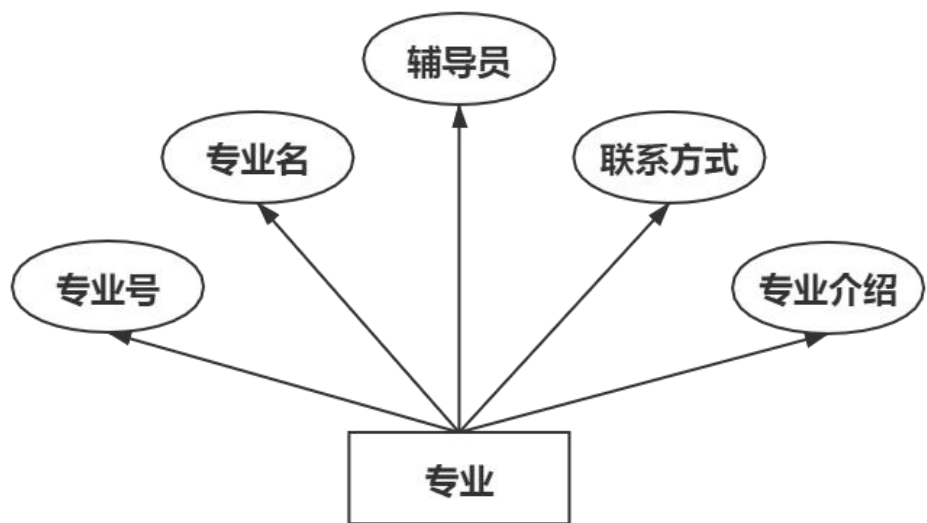
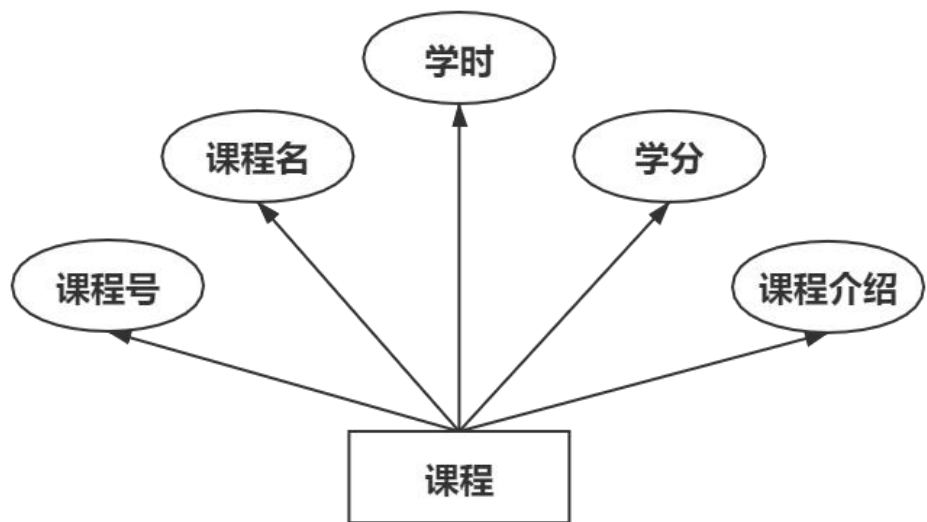
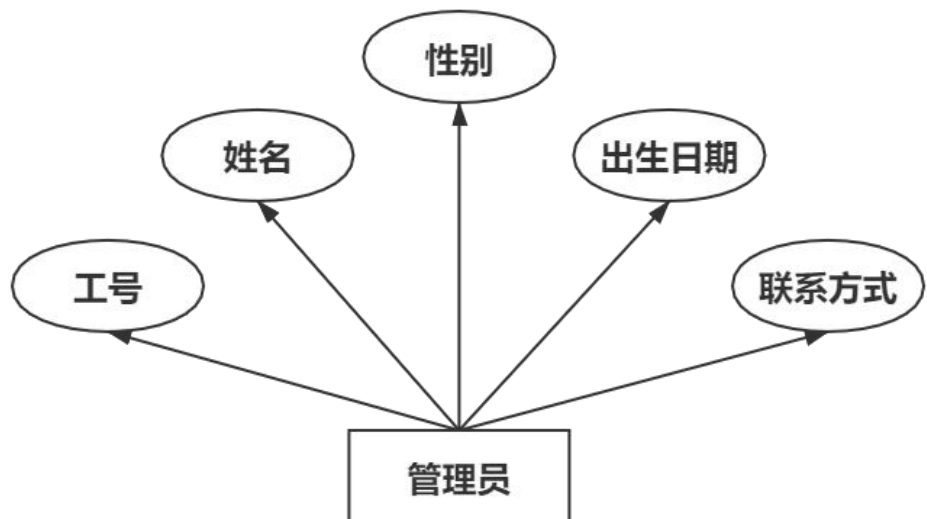


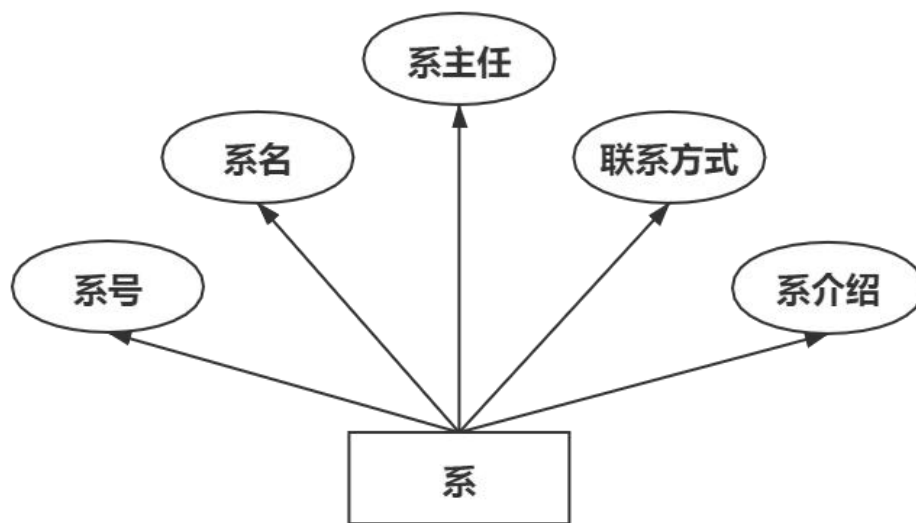
图四：功能模块图

(二) 实体属性图

实体属性图就是对实体属性进行简单、概括的描述和分析，包含了实体的主属性和有必要的属性，能让我们更好的了解这个实体。



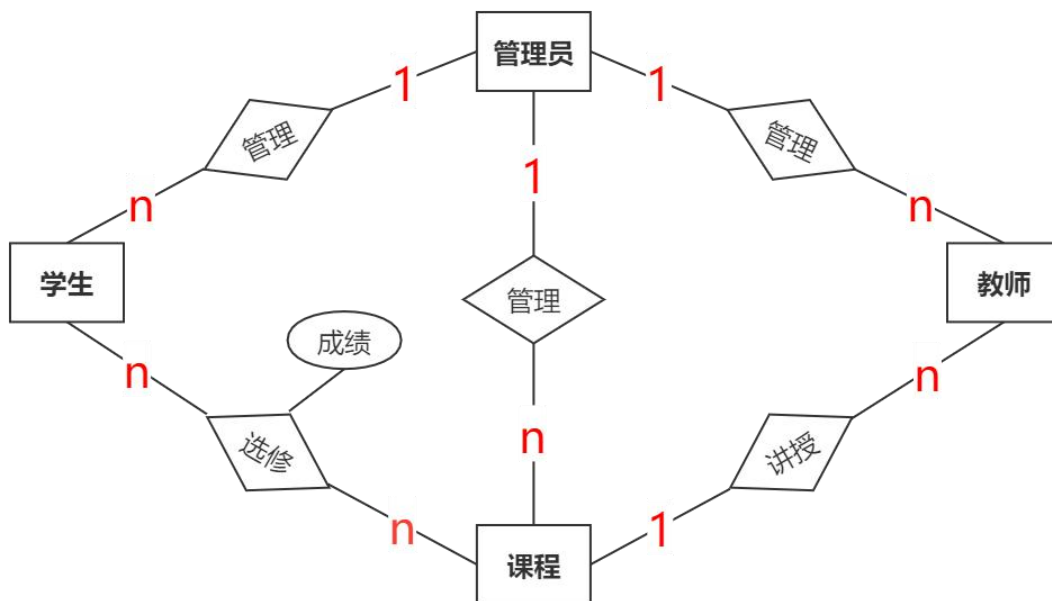




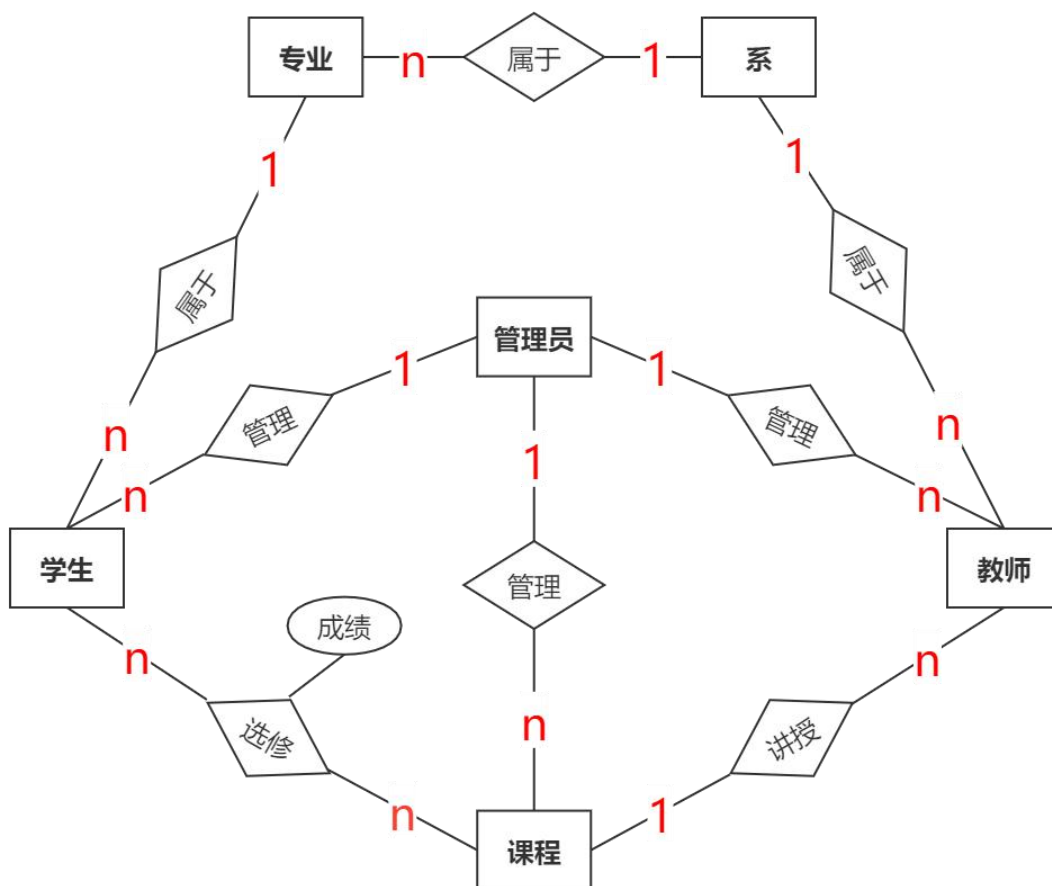
图五：实体属性图

（三）学生选课系统 E-R 图

E-R 图是 E-R 模型中用来描述现实世界的概念模型。E-R 图提供了表示实体、属性和联系的方法。（E-R 图准则：作为属性不在具有需要描述的性质；属性不能与其他实体有联系）。在 E-R 图中，实体用矩形表示、属性用椭圆表示、联系用菱形表示，E-R 图将实体与实体之间的关系（即不同实体型的实体集之间的联系）清楚地呈现出来。在此过程中，如果一个联系具有属性，则这些属性也要用无向边与该联系连接起来。



图六：1 层数据流图



图七：总数据流图

（四）逻辑结构设计——ER 图向关系模型转换

至多：6 个实体 + 8 个关系 = 14 个表

至少：6 个实体 + 1 个多对多关系 = 7 个表

根据图七的 E-R 图可以将系统中的概念模型转换为具体的表(即关系)结构，共分为 7 个关系，详细信息如下所示：

ER 图向关系模型转换归纳：

① 1 对 1：当转换为关系模型时，在两个实体任选一个添加另一个实体的主键即可。

② 1 对多：当转换为关系模型时，在 N 端添加另一端的主键。

③ 多对多：当转换为关系模型时，需要将联系转换为实体，然后在该实体上加上另外两个实体的主键，作为联系实体的主键，然后再加上该联系自身带的属性即可。

（下划线标注部分为主键）

学生（学号、专业号、姓名、性别、出生日期、联系方式）

教师（教师工号、系号、姓名、性别、出生日期、职称、联系方式）

管理员（管理员工号、姓名、性别、出生日期、联系方式）

专业（专业号、系号、专业名、辅导员、联系方式、专业介绍）

院系（系号、系名称、系主任、联系方式、系介绍）

课程（课程号、学时、学分、课程介绍）

选课信息（学号、课程号码、教师工号、成绩）

（五）表的设计

由关系模式构建的表如下所示：

表 1：学生信息表 *Student*

字段名称	数据类型	说明	主键	备注
Sno	CHAR(11)	不许为空	是	学号
Mno	CHAR(11)	不许为空		专业号
Sname	CHAR(20)	不许为空		姓名
Ssex	CHAR(2)	不许为空		性别
Sdate	DATE	不许为空		出生日期
Sphone	CHAR(11)			联系方式

表 2：教师信息表 *Teacher*

字段名称	数据类型	说明	主键	备注
Tno	CHAR(11)	不许为空	是	教师工号
SDno	CHAR(11)	不许为空		系号
Tname	CHAR(20)	不许为空		姓名
Tsex	CHAR(2)	不许为空		性别
Tdate	DATE	不许为空		出生日期
Rank	CHAR(20)	不许为空		职称
Tphone	CHAR(11)			联系方式

表 3: 管理员信息表 *Administrator*

字段名称	数据类型	说明	主键	备注
Ano	CHAR(11)	不许为空	是	管理员工号
Aname	CHAR(20)	不许为空		姓名
Asex	CHAR(2)	不许为空		性别
Adate	DATE	不许为空		出生日期
Aphone	CHAR(11)			联系方式

表 4: 专业信息表 *Major*

字段名称	数据类型	说明	主键	备注
Mno	CHAR(11)	不许为空	是	专业号
SDno	CHAR(11)	不许为空		系号
Mname	CHAR(20)	不许为空		专业名
Minstructor	CHAR(20)	不许为空		辅导员
Mphone	CHAR(11)			联系方式
Mintroduce	TEXT			专业介绍

表 5: 院系信息表 *Sdept*

字段名称	数据类型	说明	主键	备注
SDno	CHAR(11)	不许为空	是	系号
SDname	CHAR(20)	不许为空		系名称
SDdirector	CHAR(20)	不许为空		系主任

SDphone	CHAR(11)	联系方式
SDintroduce	TEXT	系介绍

表 6: 课程信息表 *Course*

字段名称	数据类型	说明	主键	备注
Cno	CHAR(11)	不许为空	是	课程号
Cperiod	INT	不许为空		学时
Ccredit	INT	不许为空		学分
Cintroduce	TEXT			课程介绍

表 7: 选课信息表 *SC*

字段名称	数据类型	说明	主键	备注
Sno	CHAR(11)	不许为空	是	学号
Cno	CHAR(11)	不许为空	是	课程号
Tno	CHAR(11)	不许为空		教师工号
Grade	INT			成绩

构建表的 SQL 语句如下所示:

表 1: 学生信息表 *Student*

```
CREATE TABLE Student (
    Sno CHAR ( 11 ) PRIMARY KEY NOT NULL,
    Mno CHAR ( 11 ) NOT NULL,
```

```

    Sname CHAR ( 20 ) NOT NULL,

    Ssex CHAR ( 2 ) NOT NULL,

    Sdate DATE NOT NULL,

    Sphone CHAR ( 11 ),

    FOREIGN KEY ( Mno ) REFERENCES Major ( Mno )

);

```

表 2: 教师信息表 *Teacher*

```

CREATE TABLE Teacher (

    Tno CHAR ( 11 ) PRIMARY KEY NOT NULL,

    SDno CHAR ( 11 ) NOT NULL,

    Tname CHAR ( 20 ) NOT NULL,

    Tsex CHAR ( 2 ) NOT NULL,

    Tdate DATE NOT NULL,

    Rank CHAR ( 20 ) NOT NULL,

    Tphone CHAR ( 11 ),

    FOREIGN KEY ( SDno ) REFERENCES Sdept ( SDno )

);

```

表 3: 管理员信息表 *Administrator*

```

CREATE TABLE Administrator (

    Ano CHAR ( 11 ) PRIMARY KEY NOT NULL,

```



```

Aname CHAR ( 20 ) NOT NULL,

Asex CHAR ( 2 ) NOT NULL,

Adate DATE NOT NULL,

Aphone CHAR ( 11 )

);

```

表 4: 专业信息表 *Major*

```

CREATE TABLE Major (

    Mno CHAR ( 11 ) PRIMARY KEY NOT NULL,

    SDno CHAR ( 11 ) NOT NULL,

    Mname CHAR ( 20 ) NOT NULL,

    Minstructor CHAR ( 20 ) NOT NULL,

    Mphone CHAR ( 11 ),

    Mintroduce TEXT,

    FOREIGN KEY ( SDno ) REFERENCES Sdept ( SDno )

);

```

表 5: 院系信息表 *Sdept*

```

CREATE TABLE Sdept (

    SDno CHAR ( 11 ) PRIMARY KEY NOT NULL,

    SDname CHAR ( 20 ) NOT NULL,

    SDdirector CHAR ( 20 ) NOT NULL,

```

```

SDphone CHAR ( 11 ),

SDintroduce TEXT

);

```

表 6: 课程信息表 *Course*

```

CREATE TABLE Course (

    Cno CHAR ( 11 ) PRIMARY KEY NOT NULL,

    Cperiod INT NOT NULL,

    Ccredit INT NOT NULL,

    Cintroduce TEXT

);

```

表 7: 选课信息表 *SC*

```

CREATE TABLE SC (

    Sno CHAR ( 11 ) NOT NULL,

    Cno CHAR ( 11 ) NOT NULL,

    Tno CHAR ( 11 ) NOT NULL,

    Grade INT,

    PRIMARY KEY ( Son, Cno ),

    FOREIGN KEY ( Son ) REFERENCES Student ( Sno ),

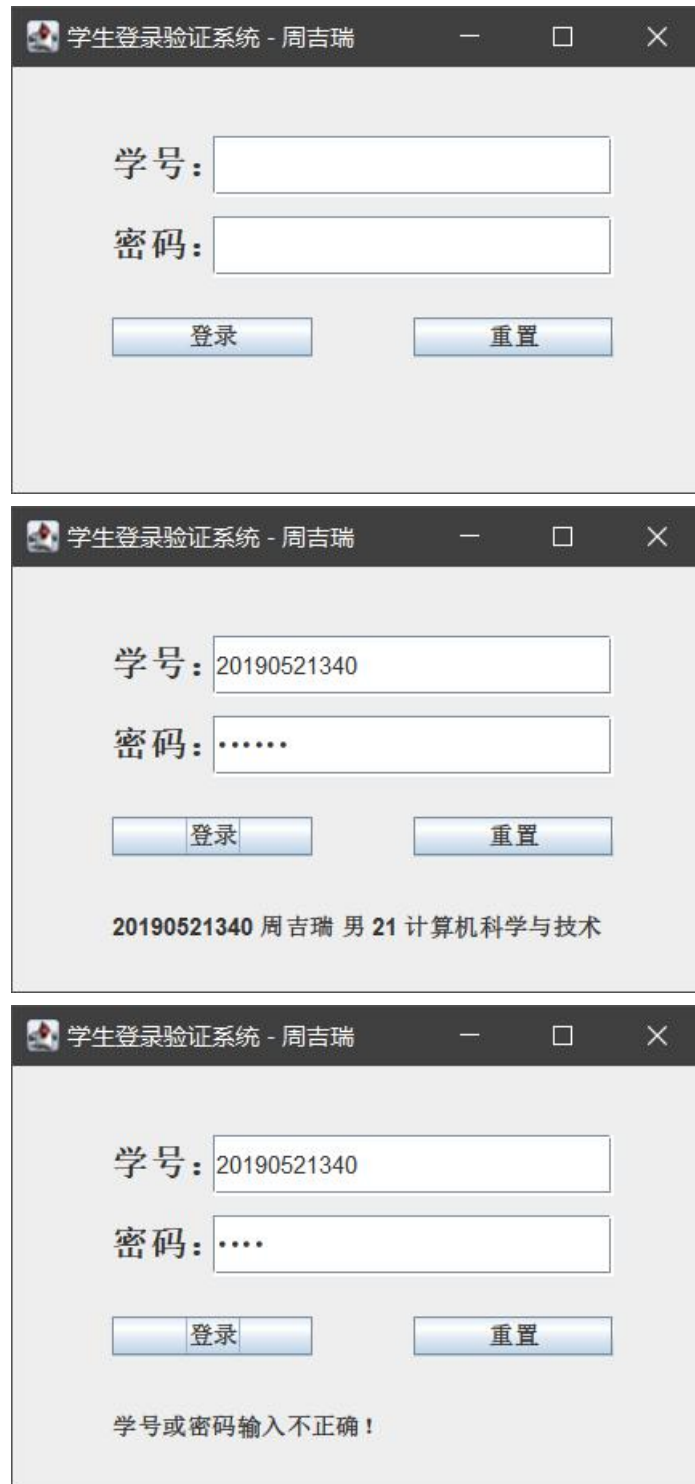
    FOREIGN KEY ( Con ) REFERENCES Course ( Cno )

);

```

四、学生选课系统登录界面

输入学号、密码，Eclipse 通过 JDBC 访问数据库。若密码与对应的学号匹配成功则显示学生基本信息，否则提示登录失败。



The image displays three sequential screenshots of a Java Swing window titled "学生登录验证系统 - 周吉瑞" (Student Login Verification System - Zhou Jirui). The window contains two text input fields labeled "学号:" (Student ID) and "密码:" (Password), and two buttons labeled "登录" (Login) and "重置" (Reset).

Top Screenshot: The login form is empty, with the "学号:" and "密码:" fields ready for input.

Middle Screenshot: The "学号:" field contains the text "20190521340" and the "密码:" field contains ".....". Below the buttons, the system displays the student's information: "20190521340 周吉瑞 男 21 计算机科学与技术" (20190521340 Zhou Jirui Male 21 Computer Science and Technology).

Bottom Screenshot: The "学号:" field contains "20190521340" and the "密码:" field contains "....". Below the buttons, a message indicates a failed login: "学号或密码输入不正确!" (Student ID or password input is incorrect!).

登录数据库表：

对象

student @test_zjr (mysql) - 表

student @test_zjr (mysql) - 表

保存

添加字段

插入字段

删除字段

主键

上移

下移

字段

索引

外键

触发器

选项

注释

SQL 预览

名	类型	长度	小数点	不是 null	虚拟	键	注释
▶ Sno	char	11		<input checked="" type="checkbox"/>	<input type="checkbox"/>	1	学号
Sname	char	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		姓名
Ssex	char	2		<input checked="" type="checkbox"/>	<input type="checkbox"/>		性别
Sage	char	3		<input checked="" type="checkbox"/>	<input type="checkbox"/>		年龄
Smajor	char	20		<input checked="" type="checkbox"/>	<input type="checkbox"/>		专业
Spassword	char	6		<input checked="" type="checkbox"/>	<input type="checkbox"/>		密码

对象

student @test_zjr (mysql) - 表

student @test_zjr (mysql) - 表

开始事务

文本

筛选

排序

导入

导出

Sno	Sname	Ssex	Sage	Smajor	Spassword
20190521340	周吉瑞	男	21	计算机科学与技术	245424
20190521354	小周周	男	18	数字媒体技术	123456
▶ 20190521369	杰瑞瑞	女	19	人工智能	542454

登录界面代码：

```
package Gui;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Scanner;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPasswordField;
import javax.swing.JTextField;

public class LoginFrame extends JFrame implements ActionListener {
    public static void main(String[] args) {
        new LoginFrame();
    }
}
```

```

}

private JLabel userNameLabel = new JLabel("学号: ");
private JTextField userNameField = new JTextField();
private JLabel passwordLabel = new JLabel("密码: ");
private JPasswordField passwordField = new JPasswordField();
private JLabel tipsLabel = new JLabel();
private JButton okBtn = new JButton("登录");
private JButton resetBtn = new JButton("重置");
private String sno;
private String spass;

public LoginFrame() {
    setTitle("学生登录验证系统 - 周吉瑞");
    Font textFont = new Font("宋体", Font.BOLD, 18);
    userNameLabel.setFont(textFont);
    passwordLabel.setFont(textFont);
    setBounds(100, 100, 360, 250);
    setLayout(null);
    userNameLabel.setBounds(50, 30, 200, 40);
    add(userNameLabel);
    userNameField.setBounds(100, 35, 200, 30);
    add(userNameField);
    passwordLabel.setBounds(50, 70, 200, 40);
    add(passwordLabel);
    passwordField.setBounds(100, 75, 200, 30);
    add(passwordField);
    okBtn.setBounds(50, 125, 100, 20);
    add(okBtn);
    okBtn.addActionListener(this);
    resetBtn.setBounds(200, 125, 100, 20);
    add(resetBtn);
    resetBtn.addActionListener(this);
    tipsLabel.setBounds(50, 170, 500, 20);
    add(tipsLabel);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setVisible(true);
}

@Override
public void actionPerformed(ActionEvent e) {
    // TODO Auto-generated method stub
    if (e.getSource() == okBtn) {

```

```

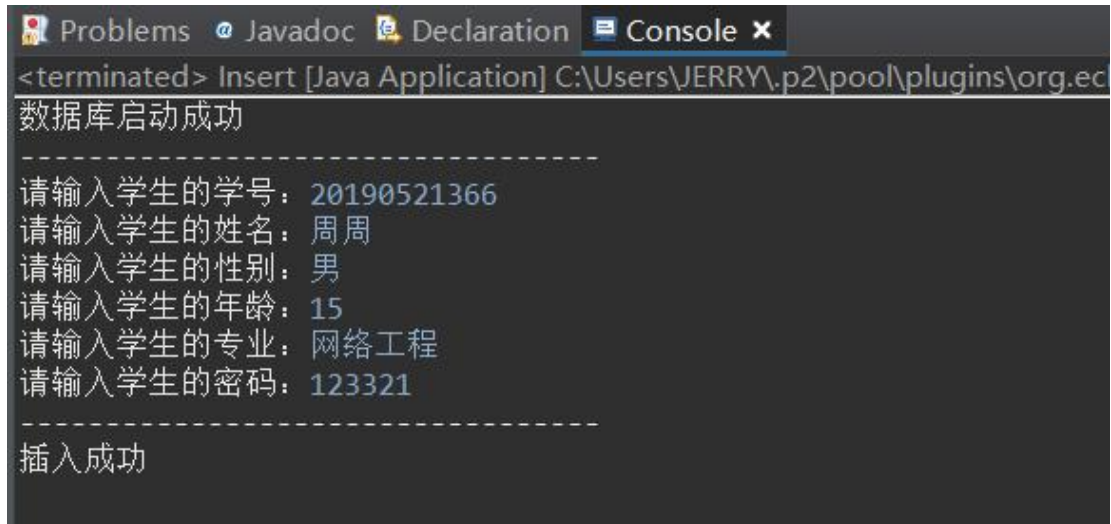
        sno = userNameField.getText();
        spass = new String(passwordField.getPassword());
        try {
            db();
        } catch (Exception e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    } else if (e.getSource() == resetBtn) {
        userNameField.setText("");
        passwordField.setText("");
        tipsLabel.setText("");
    }
}

public void db() throws Exception {
    Class.forName("com.mysql.jdbc.Driver");
    Connection conn = DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/test_zj
r", "root", "245424");
    String sql = "SELECT * FROM student WHERE Sno=? and Spassword=?";
    // 直接用 PreparedStatement 语句, sql 语句可以写在里面
    PreparedStatement sta = conn.prepareStatement(sql);
    sta.setString(1, sno); // 第一个?的内容
    sta.setString(2, spass); // 第二个?的内容
    ResultSet rs = sta.executeQuery();
    // 输出
    if (rs.next()) {
        tipsLabel.setText(rs.getString(1) + " " + rs.getString(2) + " " + rs.getString
(3) + " " + rs.getString(4) + " "
            + rs.getString(5));
    } else {
        tipsLabel.setText("学号或密码输入不正确!");
    }
    conn.close();
}
}

```

五、数据库连接

【增】



```
<terminated> Insert [Java Application] C:\Users\JERRY\p2\pool\plugins\org.ec
数据库启动成功

-----
请输入学生的学号: 20190521366
请输入学生的姓名: 周周
请输入学生的性别: 男
请输入学生的年龄: 15
请输入学生的专业: 网络工程
请输入学生的密码: 123321
-----
插入成功
```

```
package Test;

import java.util.Scanner;
import java.sql.*;

public class Insert {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        try {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("数据库启动成功");
        } catch (ClassNotFoundException ce) {
            System.out.println("SQLException:" + ce.getMessage());
        }
        try {
            java.sql.Connection con = DriverManager.getConnection("jdbc:mysql://localhost:
3306/test_zjr", "root",
                "245424");
            Statement stmt = con.createStatement();
            System.out.println("-----");
            System.out.printf("请输入学生的学号: ");
            String sno = sc.next();
            System.out.printf("请输入学生的姓名: ");
            String sname = sc.next();
            System.out.printf("请输入学生的性别: ");
```

```

        String ssex = sc.next();
        System.out.printf("请输入学生的年龄: ");
        String sage = sc.next();
        System.out.printf("请输入学生的专业: ");
        String smajor = sc.next();
        System.out.printf("请输入学生的密码: ");
        String spass = sc.next();
        System.out.println("-----");
        String sqlstr = "insert into Student values(" + "'" + sno + "'" + "," + "'" + sname + "'" + "," + "'" + ssex + "'" + "," + "'" + sage + "'" + "," + "'" + smajor + "'" + "," + "'" + spass + "'" + ")";
        stmt.executeUpdate(sqlstr);
        System.out.println("插入成功");
        stmt.close();
        con.close();
    } catch (SQLException e) {
        System.out.println("SQLException:" + e.getMessage());
    }
}
}

```

【删】



```

package Test;

import java.util.Scanner;
import java.sql.*;

public class Delete {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
    }
}

```



```

        try {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("数据库启动成功");
        } catch (ClassNotFoundException ce) {
            System.out.println("SQLException:" + ce.getMessage());
        }
        try {
            java.sql.Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/test_zjr", "root",
                "245424");

            Statement stmt = con.createStatement();
            System.out.println("-----");
            System.out.printf("请输入要删除学生的学号: ");
            String sno = sc.next();
            System.out.println("-----");
            String sql = "DELETE FROM student where Sno=" + "'" + sno + "'";
            stmt.executeUpdate(sql);
            System.out.println(sno + " 删除成功");
            stmt.close();
            con.close();
        } catch (SQLException e) {
            System.out.println("SQLException:" + e.getMessage());
        }
    }
}

```

【改】



```

<terminated> Update [Java Application] C:\Users\JERRY\.p2\pool\pl
数据库启动成功
-----
请输入要修改学生的学号: 20190521369
-----
请输入新的姓名: 吉瑞瑞
请输入新的年龄: 18
请输入新的专业: 大数据
请输入新的密码: 323232
-----
20190521369 修改成功

```

```

package Test;

import java.util.Scanner;
import java.sql.*;

public class Update {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        try {
            Class.forName("com.mysql.jdbc.Driver");
            System.out.println("数据库启动成功");
        } catch (ClassNotFoundException ce) {
            System.out.println("SQLException:" + ce.getMessage());
        }
        try {
            java.sql.Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/test_zjr", "root",
                "245424");
            Statement stmt = con.createStatement();
            System.out.println("-----");
            System.out.printf("请输入要修改学生的学号: ");
            String sno = sc.next();
            System.out.println("-----");
            System.out.printf("请输入新的姓名: ");
            String sname = sc.next();
            System.out.printf("请输入新的年龄: ");
            String sage = sc.next();
            System.out.printf("请输入新的专业: ");
            String smajor = sc.next();
            System.out.printf("请输入新的密码: ");
            String spass = sc.next();
            System.out.println("-----");
            String sql = "update student set Sname=" + "'" + sname + "'" + ",Sage=" + "'"
+ sage + "'" + ",Smajor="
                + "'" + smajor + "'" + ",Spassword=" + "'" + spass + "'" + " WHERE Sno
=" + "'" + sno + "'";
            stmt.executeUpdate(sql);
            System.out.println(sno + " 修改成功");
            stmt.close();
            con.close();
        } catch (SQLException e) {
            System.out.println("SQLException:" + e.getMessage());
        }
    }
}

```

```
}  
}
```

【查】

```
Problems @ Javadoc Declaration Console x  
<terminated> Select [Java Application] C:\Users\JERRY\.p2\pool\plugins\org.eclipse.j  
数据库启动成功  
-----  
学号      姓名    性别    年龄    专业  
-----  
20190521340    周吉瑞    男      21      计算机科学与技术  
20190521354    小周周    男      18      数字媒体技术  
20190521369    吉瑞瑞    女      18      大数据
```

```
package Test;  
  
import java.sql.*;  
  
public class Select {  
  
    // MySQL 8.0 以下版本 - JDBC 驱动名及数据库 URL  
    static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";  
    // MySQL 数据库端口及数据库名  
    static final String DB_URL = "jdbc:mysql://localhost:3306/test_zjr";  
  
    // 数据库的用户名与密码  
    static final String USER = "root";  
    static final String PASS = "245424";  
  
    public static void main(String[] args) {  
        Connection conn = null;  
        Statement stmt = null;  
        try {  
            // 注册 JDBC 驱动  
            Class.forName(JDBC_DRIVER);  
            System.out.println("数据库启动成功");  
            // 打开链接  
            // 连接数据库...  
            conn = DriverManager.getConnection(DB_URL, USER, PASS);
```

```

// 执行查询
// 实例化 Statement 对象...
stmt = conn.createStatement();
String sql;
sql = "SELECT Sno, Sname, Ssex, Sage, Smajor FROM student";
ResultSet rs = stmt.executeQuery(sql);

// 展开结果集数据库
System.out.println("-----");
-----");

System.out.println("学号" + "\t\t" + "姓名" + "\t" + "性别" + "\t" + "年龄
" + "\t" + "专业");
System.out.println("-----");
-----");

while (rs.next()) {
    // 通过字段检索
    String sno = rs.getString("Sno");
    String sname = rs.getString("Sname");
    String ssex = rs.getString("Ssex");
    String sage = rs.getString("Sage");
    String smajor = rs.getString("Smajor");

    // 输出数据
    System.out.println(sno + "\t" + sname + "\t" + ssex + "\t" + sage + "\t" +
smajor);

    System.out.print("\n");
}

// 完成后关闭
rs.close();
stmt.close();
conn.close();
} catch (SQLException se) {
    // 处理 JDBC 错误
    se.printStackTrace();
} catch (Exception e) {
    // 处理 Class.forName 错误
    e.printStackTrace();
} finally {
    // 关闭资源
    try {
        if (stmt != null)

```

```
        stmt.close();
    } catch (SQLException se2) {
    }
    try {
        if (conn != null)
            conn.close();
    } catch (SQLException se) {
        se.printStackTrace();
    }
}
}
```