

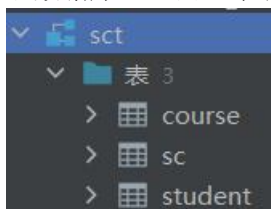
# 实验一、关系数据库应用系统开发实践

姓名：瞿久尧 学号：120L022314

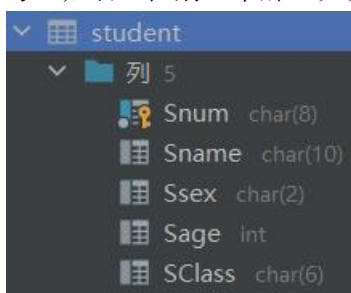
## 1 主要数据结构

### 1.1 库表设计

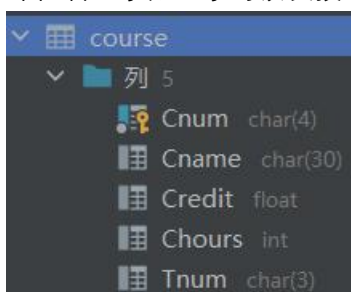
该数据库 sct 由三个表 student、course、sc 组成，分别表示学生、课程、成绩



其中 student 表中包含 5 列：Snum(key)、Sname、Ssex、Sage、SClass，分别对应学生的学号、姓名、性别、年龄、班级



course 表中包含 5 列：Cnum、Cname、Credit、Chours、Tnum，分别对应课程的课程号、课程名、学分、学时以及授课老师编号



sc 表中包含 3 列：Snum、Cnum、Score，分别对应学生的学号、课程编号、学生成绩，并且 Snum 和 Cnum 分别与 student 中的 Snum 和 course 中的 Cnum 相联

### 1.2 记录生成

使用 DriverManager.getConnection 连接数据库后，使用 PreparedStatement 执行 sql 语句，使用 ResultSet 获得查询结果，并将查询到的记录逐条放到二维数组中，并且通过一维数组存储表头

```
rs = pstmt.executeQuery();  
// 将查询获得的记录数据，转换成适合生成JTable的数据形式  
Object[][] info = new Object[count][5];  
String[] title = new String[5];
```

```

case "student":{
    String[] title1 = {"学号", "姓名", "性别", "年龄", "班级"};
    title = title1;
    for(int i=0;i<5;i++){
        title[i]="学号";
    }
    count = 0;
    while (rs.next()) {
        info[count][0] = rs.getString( columnLabel: "Snum");
        info[count][1] = rs.getString( columnLabel: "Sname");
        info[count][2] = rs.getString( columnLabel: "Ssex");
        info[count][3] = Integer.valueOf(rs.getInt( columnLabel: "Sage"));
        info[count][4] = rs.getString( columnLabel: "SClass");
        count++;
    }
    break;
}
}

```

## 2 算法和系统

### 2.1 装载表

获取数据库

```

Class.forName( className: "com.mysql.cj.jdbc.Driver");
String url = "jdbc:mysql://localhost:3306/sct";//注意设置时区
String username = "root";
String passwords = "root";
Connection conn = DriverManager.getConnection(url, username, passwords);

```

根据 sql 语句查询对应表

```

String sql = "select * from " + table;
PreparedStatement pstmt = conn.prepareStatement(sql);
ResultSet rs = pstmt.executeQuery();
int count = 0;
while (rs.next()) {
    count++;
}
rs = pstmt.executeQuery();

```

将查询获得的记录转换成表格形式

```
// 将查询获得的记录数据，转换成适合生成JTable的数据形式
Object[][] info = new Object[count][5];
String[] title = {"学号", "姓名", "性别", "年龄", "班级"};
count = 0;
while (rs.next()) {
    info[count][0] = rs.getString( columnLabel: "Snum");
    info[count][1] = rs.getString( columnLabel: "Sname");
    info[count][2] = rs.getString( columnLabel: "Ssex");
    info[count][3] = rs.getInt( columnLabel: "Sage");
    info[count][4] = rs.getString( columnLabel: "SClass");
    count++;
}
// 创建JTable
this.tabDemo1 = new JTable(info, title);
// 显示表头
this.jth1 = this.tabDemo1.getTableHeader();
// 将JTable加入到带滚动条的面板中
this.scpDemo1.getViewport().add(tabDemo1);
```

## 2.2 挑选数据库中的表装载

设置下拉菜单

```
this.comboBox = new JComboBox<>();
comboBox.addItem("student");
comboBox.addItem("course");
comboBox.addItem("sc");
```

根据下拉菜单中获得到表名，调用与 1.1 同样的步骤装载

```
String table = (String)this.comboBox.getSelectedItem();
String sql = "select * from " + table;
```

## 2.3 联表查询

A. 获取数据库

B. 为每个复选框设置标记

C. 遍历完所有复选框后可以根据标记数组来判断 sql 语句中 select 后接哪几个表，同时是否在表后加上 where 条件

D. 然后根据每个被选择的标记在 sql 语句中加上对应的条件语句

```

if (flag[0] == true) {
    sql += "student.Snum like " + "\"" + jb_Snum.getText() + "\"";
    sum++;
}
if (flag[1] == true) {
    if (sum > 0)
        sql += " && ";
    sql += " Sname like " + "\"" + jb_Sname.getText() + "\"";
    sum++;
}
if (flag[2] == true) {
    if (sum > 0)
        sql += " && ";
    sql += " Ssex = " + "\"" + jb_Ssex.getText() + "\"";
    sum++;
}

```

E.将构造出来的 sql 语句显示在 TestField 中，并且直接使用该 sql 语句查询表，同样将查询结果放到表格中

## 2.4 报错

根据程序中的 ClassNotFoundException 和 SQLException 捕捉到异常，并通知用户不同的错误

```

} catch (ClassNotFoundException cnfe) {
    cnfe.printStackTrace();
    JOptionPane.showMessageDialog( parentComponent: null, message: cnfe.getCause() + ":数据源错误", title: "错误", JOptionPane.ERROR_MESSAGE);
} catch (SQLException sqle) {
    sqle.printStackTrace();

    JOptionPane.showMessageDialog( parentComponent: null, message: sqle.getErrorCode() + ":数据操作错误", title: "错误", JOptionPane.ERROR_MESSAGE);
}

```

## 3 部分程序运行结果

### 3.1 装载

表1: student					装载	
学号	姓名	性别	年龄	班级		
11900001	王尚香	女	33	16		
11900002	朱柳	男	34	91		
11900003	杨香	男	29	17		
11900004	陈忆夏	女	16	87		
11900005	卫忆香	男	28	27		
11900006	郑尔初	女	16	89		
11900007	蒋桃琪	男	27	22		
11900008	孙兰	女	19	13		
11900009	郑忆	女	26	1		
11900010	蒋香	女	24	16		
11900011	龚雄	男	16	21		

表2: course					装载	
课号	课程名	学分	课时	老师		
A100	航天1	1.0	16	A10		
A101	航天2	1.0	16	A10		
A102	航天3	2.0	32	A10		
A103	航天4	1.5	24	A10		
A104	航天5	2.0	32	A10		
A105	航天6	0.5	8	A10		
A106	航天7	1.5	24	A10		
A107	航天8	0.5	8	A10		
A108	航天9	2.0	32	A10		
A109	航天10	1.0	16	A10		
A110	航天11	1.5	24	A11		

表3: sc			装载	
学号	课号	成绩		
12200001	G782	60.2477		
12200001	F682	76.7309		
12200001	F682	71.788		
12200001	I904	71.3061		
12200001	C349	86.2085		
12200001	H803	91.4206		
12200001	H871	66.0193		
12200001	F652	93.1837		
12200001	F640	72.1729		
12200001	C356	97.8723		
12200001	A132	81.1182		

### 3.2 装载数据库中任意表

表: student

student

course

学号	姓名	性别	学号	学号
11900001	王岗	男	33	16
11900002	朱彬	男	34	91
11900003	杨香	男	29	17
11900004	陈忆夏	女	16	87
11900005	卫忆香	男	28	27
11900006	郑尔初	女	16	89
11900007	蒋桃琪	男	27	22
11900008	孙兰	女	19	13
11900009	郑忆	女	26	1
11900010	蒋香	女	24	16
11900011	蓝桃	男	16	21

### 3.3 联表查询

检索条件

退出

☐ 学号

☐ 姓名

☐ 性别

☐ 班级

☐ 年龄  到

☐ 课号

☐ 课程名

☐ 老师

☐ 成绩  到

检索条件

退出

☒ 学号  ☐ 姓名

☐ 性别  ☐ 班级

☒ 年龄  到

☒ 课号  ☐ 课程名

☐ 老师

☒ 成绩  到

构造SQL语句

输入SQL语句

&& course.Cnum like "A%" && 90 <= Score && Score <= 95 && sc.Snum=student.Snum && sc.Cnum=course.Cnum limit 100

执行SQL语句

结果

Snum	Sname	Ssex	Sage	SClass	Cnum	Cname	Credit	Chours	Tnum	Snum	Cnum	Score
1200...	赵香蓉	女	20	51	A118	航天19	2.0	32	A11	1200...	A118	94.42...
1200...	赵馨尔	女	22	10	A101	航天2	1.0	16	A10	1200...	A101	93.44...
1200...	孙梦	男	20	69	A120	航天21	1.5	24	A12	1200...	A120	92.75...
1200...	朱柳	女	20	59	A121	航天22	1.5	24	A12	1200...	A121	93.29...
1200...	周琪	男	21	44	A100	航天1	1.0	16	A10	1200...	A100	92.337
1200...	沈琪尔	男	22	91	A160	航天61	1.0	16	A16	1200...	A160	93.03...
1200...	钱之青	男	20	84	A180	航天81	1.0	16	A18	1200...	A180	90.69...
1200...	周梦	女	22	41	A122	航天23	1.0	16	A12	1200...	A122	92.66...
1200...	赵元尔	男	22	12	A179	航天80	1.5	24	A17	1200...	A179	92.20...
1200...	赵元尔	男	22	12	A117	航天18	1.5	24	A11	1200...	A117	90.29...
1200...	韩元	男	20	13	A182	航天83	1.0	16	A18	1200...	A182	93.945