

哈爾濱工業大學

# 数据库实验报告

题        目 交互式 sql 实验

专        业 信息安全

学        号 1180300829

学        生 余 涛

指导教师 史建焘

# 1 实验目的

熟练掌握数据库语言的基本操作


# 2 实验环境

阿里云 DMS 云数据库，dmslab.aliyun.com

# 3 实验内容及要求

## 3.1 创建学生选课数据库

直接创建数据库 sct\_yutao 即可：create databse yutao;


▼  sct\_yutao




## 3.2 建立学生选课系统中的三个基本表

连接数据库后执行以下 sql 语句操作即可：

```
// 创建Student表
String sql = "CREATE TABLE IF NOT EXISTS Student(`S#` CHAR(8) PRIMARY KEY COMMENT '学号' NOT NULL, SNAME CHAR(10) COMMENT '姓名' NOT NULL, " +
    "SEX CHAR(2) COMMENT '性别', AGE INTEGER COMMENT '年龄', DNAME CHAR(10) COMMENT '系别', CLASS CHAR(6) COMMENT '班号');";
statement.execute(sql);
// 创建Course表
sql = "CREATE TABLE IF NOT EXISTS Course(`C#` CHAR(3) PRIMARY KEY COMMENT '课程号' NOT NULL, CNAME CHAR(12) COMMENT '课程名', " +
    "HOURS INTEGER COMMENT '学时', CREDIT FLOAT(1) COMMENT '学分', SEMSTER INTEGER COMMENT '学期');";
statement.execute(sql);
// 创建SC表
sql = "CREATE TABLE IF NOT EXISTS SC(`S#` CHAR(8) COMMENT '学号' NOT NULL, `C#` CHAR(3) COMMENT '课程号' NOT NULL, " +
    "SCORE FLOAT(1) COMMENT '成绩', PRIMARY KEY(`S#`, `C#`), FOREIGN KEY(`S#`) REFERENCES Student(`S#`), " +
    "FOREIGN KEY(`C#`) REFERENCES Course(`C#`));";
```

创建的表如下：

▼  sct\_yutao

- >  course
- >  sc
- >  student

Student 表：

	 S#	 SNAME	 SEX	 AGE	 DNAME	 CLASS
1	98020101	李三	女	18	能源	980201
2	98020102	李四	男	19	能源	980201
3	98030101	张三	男	20	计算机	980301
4	98030102	张四	女	20	计算机	980301
5	98030103	张五	男	19	计算机	980301
6	98040201	王三	男	20	自动控制	980402
7	98040202	王四	男	21	自动控制	980402
8	98040203	王五	女	19	自动控制	980402

Course 表：

	C#	CNAME	HOURS	CREDIT	SEMSTER
1	001	数据库	40	6	1
2	002	数据结构	40	6	3
3	003	编译原理	40	6	7
4	004	C 语言	30	4.5	6
5	005	高等数学	80	12	2
6	006	计算机网络	20	3	4

SC 表:

	S#	C#	SCORE
1	98020101	001	80
2	98020101	005	95
3	98020102	005	85
4	98030101	001	90
5	98030101	002	86
6	98030101	006	62
7	98030102	001	82
8	98030102	002	78
9	98030102	004	66
10	98030102	005	92
11	98030102	006	50
12	98030103	002	68
13	98030103	006	62

### 3.3 插入数据

执行以下 sql 语句即可:

```
String sql = "insert into Student values ('98030101', '张三', '男', 20, '计算机', '980301');";
allsql.add(sql);
sql = "insert into Student values ('98030102', '张四', '女', 20, '计算机', '980301');";
allsql.add(sql);
sql = "insert into Student values ('98030103', '张五', '男', 19, '计算机', '980301');";
allsql.add(sql);
sql = "insert into Student values ('98040201', '王三', '男', 20, '自动控制', '980402');";
allsql.add(sql);
sql = "insert into Student values ('98040202', '王四', '男', 21, '自动控制', '980402');";
allsql.add(sql);
sql = "insert into Student values ('98040203', '王五', '女', 19, '自动控制', '980402');";
allsql.add(sql);
sql = "insert into Student values ('98020101', '李三', '女', 18, '能源', '980201');";
allsql.add(sql);
sql = "insert into Student values ('98020102', '李四', '男', 19, '能源', '980201');";
```

```

sql = "insert into Course values ('001', '数据库', 40, 6, 1);";
allsql.add(sql);
sql = "insert into Course values ('002', '数据结构', 40, 6, 3);";
allsql.add(sql);
sql = "insert into Course values ('003', '编译原理', 40, 6, 7);";
allsql.add(sql);
sql = "insert into Course values ('004', 'C 语言', 30, 4.5, 6);";
allsql.add(sql);
sql = "insert into Course values ('005', '高等数学', 80, 12, 2);";
allsql.add(sql);
sql = "insert into Course values ('006', '计算机网络', 20, 3, 4);";

sql = "insert into SC values ('98030101', '001', 90);";
allsql.add(sql);
sql = "insert into SC values ('98030101', '002', 86);";
allsql.add(sql);
sql = "insert into SC values ('98030101', '006', 62);";
allsql.add(sql);
sql = "insert into SC values ('98030102', '002', 78);";
allsql.add(sql);
sql = "insert into SC values ('98030102', '004', 66);";
allsql.add(sql);
sql = "insert into SC values ('98030102', '001', 82);";
allsql.add(sql);
sql = "insert into SC values ('98030102', '005', 92);";
allsql.add(sql);
sql = "insert into SC values ('98030102', '006', 50);";
allsql.add(sql);
sql = "insert into SC values ('98030103', '002', 68);";
allsql.add(sql);
sql = "insert into SC values ('98030103', '006', 62);";
allsql.add(sql);
sql = "insert into SC values ('98020101', '001', 80);";
allsql.add(sql);
sql = "insert into SC values ('98020101', '005', 95);";
allsql.add(sql);
sql = "insert into SC values ('98020102', '005', 85);";

```

结果如上 3.2 所示

### 3.4 用 sql 语句完成如下查询操作：

(1) 查询所有学生信息

执行的 sql 语句如下：

```
String sql = "select * from Student;";
```

结果如下：

查询所有学生信息

查询成绩在80~90 ( $\geq 80, \leq 90$ ) 分之之间的学生的选课信息(学号、课程号和成绩)

查询并列出所有系名(不重复)

查询有多少个同学姓'王'

查询数据库课程的最高成绩

将学生的成绩按课号升序, 成绩降序排列

统计每个学生选修的课程数

Sid	SNAME	SEX	AGE	DNAME	CLASS
98020101	李三	女	18	能源	980201
98020102	李四	男	19	能源	980201
98030101	张三	男	20	计算机	980301
98030102	张四	女	20	计算机	980301
98030103	张五	男	19	计算机	980301
98040201	王三	男	20	自动控制	980402
98040202	王四	男	21	自动控制	980402
98040203	王五	女	19	自动控制	980402

(2) 查询成绩在 80~90 ( $\geq 80, \leq 90$ ) 分之之间的学生的选课信息(学号、课程号和成绩)

执行的 sql 语句如下:

```
String sql = "select * from SC where SCORE >= 80 and SCORE <= 90;";
```

结果如下:

查询所有学生信息

查询成绩在80~90 ( $\geq 80, \leq 90$ ) 分之之间的学生的选课信息(学号、课程号和成绩)

查询并列出所有系名(不重复)

查询有多少个同学姓'王'

查询数据库课程的最高成绩

将学生的成绩按课号升序, 成绩降序排列

统计每个学生选修的课程数

Sid	C#	SCORE
98020101	001	80
98020102	005	85
98030101	001	90
98030101	002	86
98030102	001	82

(3) 查询并列出所有系名(不重复)

执行的 sql 语句如下：

```
String sql = "select distinct DNAME from Student;";
```

结果如下：

查询所有学生信息

查询成绩在80~90 ( $\geq 80, \leq 90$ ) 分之之间的学生的选课信息(学号、课程号和成绩)

查询并列出所有系名 (不重复)

查询有多少个同学姓'王'

查询数据库课程的最高成绩

将学生的成绩按课号升序，成绩降序排列

统计每个学生选修的课程数

查询结果

DNAME
能源
计算机
自动控制

(4) 查询有多少个同学姓 ‘王’

执行的 sql 语句如下：

```
String sql = "select count(SNAME) as SUMWANG from Student where SNAME like '王%';";
```

结果如下：

查询所有学生信息

查询成绩在80~90 ( $\geq 80, \leq 90$ ) 分之之间的学生的选课信息(学号、课程号和成绩)

查询并列出所有系名 (不重复)

查询有多少个同学姓'王'

查询数据库课程的最高成绩

将学生的成绩按课号升序，成绩降序排列

统计每个学生选修的课程数

查询结果

SUMWANG
3



(5) 查询数据库课程的最高成绩

执行的 sql 语句如下：

```
String sql = "select max(SCORE) as MAXSCORE from SC, Course C where C.CNAME = '数据库' and C.`C#` = SC.`C#`";
```

结果如下：

The screenshot shows a window with a list of queries on the left and a results pane on the right. The queries are:

- 查询所有学生信息
- 查询成绩在80~90 ( $\geq 80, \leq 90$ ) 分之之间的学生的选课信息(学号、课程号和成绩)
- 查询并列出所有系名(不重复)
- 查询有多少个同学姓'王'
- 查询数据库课程的最高成绩
- 将学生的成绩按课号升序，成绩降序排列
- 统计每个学生选修的课程数

The '查询数据库课程的最高成绩' query is selected. The results pane shows a table with one column, 'MAXSCORE', and one row with the value '90'.

MAXSCORE
90

(6) 将学生的成绩按课号升序，成绩降序排列

执行的 sql 语句如下：

```
String sql = "select * from SC order by `C#`, SCORE desc;";
```

结果如下：

The screenshot shows a window with a list of queries on the left and a results pane on the right. The queries are:

- 查询所有学生信息
- 查询成绩在80~90 ( $\geq 80, \leq 90$ ) 分之之间的学生的选课信息(学号、课程号和成绩)
- 查询并列出所有系名(不重复)
- 查询有多少个同学姓'王'
- 查询数据库课程的最高成绩
- 将学生的成绩按课号升序，成绩降序排列
- 统计每个学生选修的课程数

The '将学生的成绩按课号升序，成绩降序排列' query is selected. The results pane shows a table with three columns: 'Sid', 'C#', and 'SCORE'. The data is sorted by 'C#' ascending and 'SCORE' descending.

Sid	C#	SCORE
98030101	001	90
98030102	001	82
98020101	001	80
98030101	002	86
98030102	002	78
98030103	002	68
98030102	004	66
98020101	005	95
98030102	005	92
98020102	005	85
98030101	006	62

(7) 统计每个学生选修的课程数

执行的 sql 语句如下：

```
String sql = "select `S#`, count(`C#`) as SUMC from SC group by `S#`";
```

结果如下：

查询所有学生信息

查询成绩在80~90 ( $\geq 80, \leq 90$ ) 分之间的学生的选课信息(学号、课程号和成绩)

查询并列出现所有系名 (不重复)

查询有多少个同学姓'王'

查询数据库课程的最高成绩

将学生的成绩按课号升序，成绩降序排列

统计每个学生选修的课程数

S#	SUMC
98020101	2
98020102	1
98030101	3
98030102	5
98030103	2

查询结果