

《数据库系统》实验报告

| | | | | |
|--|---------------------|------|---|-----|
| 年级、专业、班级 | 2021 计科卓越 2 班 | | 姓名 | 文红兵 |
| 实验题目 | 基本 SQL 操作 | | | |
| 实验时间 | 2023.3.27~2023.4.10 | 实验地点 | DS33401 | |
| 实验成绩 | | 实验性质 | <input type="checkbox"/> 验证性 <input type="checkbox"/> 设计性 <input checked="" type="checkbox"/> 综合性 | |
| <p>教师评价：</p> <p><input type="checkbox"/>算法/实验过程正确； <input type="checkbox"/>源程序/实验内容提交 <input type="checkbox"/>程序结构/实验步骤合理；</p> <p><input type="checkbox"/>实验结果正确； <input type="checkbox"/>语法、语义正确； <input type="checkbox"/>报告规范；</p> <p>其他：</p> <p>评价教师签名：</p> | | | | |
| <p>一、实验目的</p> <p>1. 了解 GaussDB 数据库的基本操作</p> <p>2、掌握基本的数据库创建、数据库表创建、基本查询。</p> <p>3、学习并掌握多条件查询和多表查询。</p> | | | | |
| <p>二、实验项目内容</p> <p>1. 基本表操作</p> <p>1) 按下图创建四个表：teacher、student、course 和 SC，为属性选择合适的域、合适的主码和外键约束，并为他们插入所列出数据；</p> <p>teacher (TID, TNAME, DEPT, SALARY)</p> <p>中文语义：教师（教工号，姓名，系，薪水）</p> <p>student (SID, SNAME, DEPT, AGE, GENDER)</p> <p>中文语义：学生（学号，姓名，系，年龄，性别）</p> <p>course (CID, CNAME, DEPT, CREDITS, TEACHER)</p> <p>中文语义：课程（课程号，课程名，系，学分，任课教师）</p> <p>SC (SID, CID, GRADE)</p> <p>中文语义：学习（学号，课程号，成绩）</p> | | | | |

报告创建时间：

表 teacher

| TID | TNAME | DEPT | SALARY |
|-------|----------|---------|--------|
| 14001 | Katz | CS | 75000 |
| 14002 | Crick | Biology | 72000 |
| 14003 | Gold | Physics | 87000 |
| 14004 | Einstein | Physics | 95000 |
| 14005 | Kim | CS | 65000 |
| 14006 | Wu | Finance | 90000 |
| 14007 | Brandt | CS | 65000 |
| 14008 | Singh | Finance | 80000 |

表 student

| SID | SNAME | DEPT | AGE | GENDER |
|-----|----------|---------|-----|--------|
| S1 | Wangfeng | Physics | 20 | M |
| S2 | Liu fang | Physics | 19 | M |
| S3 | Chen yun | CS | 22 | M |
| S4 | Wu kai | Finance | 19 | M |
| S5 | Liu li | CS | 21 | F |
| S6 | Dongqing | Finance | 18 | F |
| S7 | Li | CS | 19 | F |
| S8 | Chen | CS | 21 | F |
| S9 | Zhang | Physics | 19 | M |
| S10 | Yang | CS | 22 | F |
| S11 | Wang | CS | 19 | F |

表 course

| CID | CNAME | DEPT | CREDITS | TEACHER |
|-----|-----------|-------------|---------|----------|
| C1 | DB | CS | 2.0 | Li |
| C2 | maths | Mathematics | 2.0 | Ma |
| C3 | chemistry | Chemistry | 2.5 | Zhou |
| C4 | physics | Physics | 1.5 | Shi |
| C5 | OS | CS | 2.0 | Wen |
| C6 | Database | CS | 2.0 | Katz |
| C7 | Algorithm | CS | 2.5 | Gold |
| C8 | Java | CS | 1.5 | Einstein |
| C9 | Marketing | Finance | 2.0 | Wu |

表 SC

| SID | CID | GRADE |
|-----|-----|-------|
| S1 | C1 | 70 |
| S1 | C3 | 81 |
| S2 | C4 | 92 |
| S2 | C2 | 85 |
| S3 | C1 | 65 |
| S3 | C5 | 57 |
| S4 | C1 | 87 |
| S5 | C4 | 83 |

2) 在 student 表中, 为姓名为' Zhang' 且系信息错填为' Physics' 的同学修改信息, 将其系信息修改为' CS' ;

3) 删除 teacher 表中, 属于 Finance 学院的教师信息;

4) 在 teacher 表中, 为工资低于或等于 70000 的教师增长 10%的工资, 为

工资高于 70000 的教师增长 5%的工资。

2. 基本数据查询

- 1) 基于 teacher 表, 找出“物理系 Physics 和生物系 Biology”所有教师的名字和工资;
- 2) 基于 teacher 表, 列出教师所在的各个系名, 要求每个系名仅出现一次;
- 3) 基于 teacher 表, 希望查看“若将每位教师的工资提高 20%后”各位教师的姓名和工资是多少;
- 4) 基于表 student、SC 和 course, 查看到计算机系 CS 的每位学生所选课程的所有信息, 包括学生姓名、所在系、课程名称、课程学分的情况。

3. 复杂数据查询

- 1) 查询全体学生的姓名、年龄;
- 2) 查询所有选修过课的学生的学号;
- 3) 查询考试成绩低于 60 分的学生的学号;
- 4) 查询年龄在 20 至 23 之间的学生姓名、性别和年龄;
- 5) 查询所有姓 liu 的学生的学号、姓名和年龄;
- 6) 查询学习 C1 课程的学生最高分数;
- 7) 查询各个课程号与相应的选课人数;
- 8) 查询选修 C3 课程的学生的姓名;
- 9) 查询每一门课程的平均成绩。

4、学生自主上机实验内容（选做）（未获满分时，可酌情加分）

在必做题的数据库中完成以下要求：

学生只能选择自己学院开设的课程。发现 CS 学院有的同学选择了其他学院开设的课程。在 SC 表中删除这些错选的记录。

三、实验过程或算法（源程序）

[illegible]

```
51
52 insert into course values ('C1', 'DB', 'CS', 2.0, 'Li'),
53                           ('C2', 'maths', 'Mathematics', 2.0, 'Ma'),
54                           ('C3', 'chemistry', 'Chemistry', 2.5, 'Zhou'),
55                           ('C4', 'physics', 'Physics', 1.5, 'Shi'),
56                           ('C5', 'OS', 'CS', 2.0, 'Wen'),
57                           ('C6', 'Database', 'CS', 2.0, 'Katz'),
58                           ('C7', 'Algorithm', 'CS', 2.5, 'Gold'),
59                           ('C8', 'Java', 'CS', 1.5, 'Einstein'),
60                           ('C9', 'Marketing', 'Finance', 2.0, 'Wu') ;
61
62 insert into sc values ('S1', 'C1', 70),
63                      ('S1', 'C3', 81),
64                      ('S2', 'C4', 92),
65                      ('S2', 'C2', 85),
66                      ('S3', 'C1', 65),
67                      ('S3', 'C5', 57),
68                      ('S4', 'C1', 87),
69                      ('S5', 'C4', 83) ;
70
71
72 update student
73 set DEPT='CS'
74 where SNAME = 'Zhang' and DEPT = 'Physics' ;
75
76 delete from teacher
77 where DEPT = 'Finance' ;
78
79 update teacher
80 set salary = case
81             when salary <= 70000 then salary * 1.1
82             else salary * 1.05
83             end;
84
85 select tname, salary
86 from teacher
87 where dept = 'Physics' or dept = 'Biology' ;
88
89 SELECT DISTINCT dept
90 FROM teacher;
91
92 SELECT tname, salary*1.2
93 FROM teacher;
94
95
96 SELECT s.sname, s.dept, c.cname, c.credits
97 FROM student s
98 JOIN SC ON s.sid = SC.sid
99 JOIN course c ON SC.cid = c.cid
100 WHERE s.dept = 'CS' ;
101
102 SELECT sname, age
103 FROM student ;
104
105 SELECT DISTINCT sname, age
```

```
106 FROM student as s, sc
107 WHERE sc.SID = s.SID ;
108
109 SELECT DISTINCT sid
110 FROM SC
111 WHERE grade < 60;
112
113 SELECT sname, gender, age
114 FROM student
115 WHERE age BETWEEN 20 AND 23;
116
117 SELECT sid, sname, age
118 FROM student
119 WHERE sname LIKE 'Liu%';
120
121 SELECT MAX(grade)
122 FROM sc
123 WHERE cid = 'C1' ;
124
125 SELECT cid, count(*) as num
126 FROM sc
127 GROUP BY cid ;
128
129 SELECT sname
130 FROM sc JOIN student as s ON sc.sid = s.sid
131 WHERE sc.CID = 'C3' ;
132
133 SELECT cid, AVG(grade)
134 FROM sc
135 GROUP BY cid
136
137 DELETE FROM sc
138 WHERE sc.sid in
139 (
140     SELECT sid
141     from student
142     where dept = 'CS'
143 ) and sc.cid not in
144 (
145     select cid
146     from course
147     where dept = 'CS'
148 );
149
```

四、实验结果及分析和（或）源程序调试过程

1.创表

teacher

该语句创建了一个名为teacher的表，包含四个列：T_ID、TNAME、DEPT和SALARY。其中，T_ID为主键，TNAME不为空，DEPT可以为空，SALARY必须在2000到1000000之间。

```
1 create table teacher(  
2     T_ID varchar(255) primary key,  
3     TNAME varchar(255) not null,  
4     DEPT varchar(255),  
5     SALARY int check ( salary between 2000 and 1000000 )  
6 );  
7
```

student

该语句创建了一个名为student的表，包含五个列：SID、SNAME、DEPT、AGE和GENDER。其中，SID为主键，SNAME不为空，DEPT可以为空，AGE必须在0到120之间，GENDER为单个字符，取值为'F'或'M'。

```
1 create table student(  
2     SID varchar(255) primary key ,  
3     SNAME varchar(255) not null ,  
4     DEPT varchar(255) ,  
5     AGE int check ( age between 0 and 120 ),  
6     GENDER char(1) check ( gender in('F','M') )  
7 );
```

course

该语句创建了一个名为course的表，包含五个列：CID、CNAME、DEPT、CREDITS和TEACHER。其中，CID为主键，CNAME不为空，DEPT可以为空，CREDITS为小数类型，保留一位小数，不能为null，TEACHER可以为空。

```
1 create table course(  
2     CID varchar(255) primary key,  
3     CNAME varchar(255) not null,  
4     DEPT varchar(255),  
5     CREDITS decimal(2, 1) not null,  
6     TEACHER varchar(255)  
7 );
```


SC

该语句创建了一个名为SC的表，包含三个列：SID、CID和GRADE。其中，SID和CID都是外键，分别参考表student和course的主键，组成了复合主键。GRADE必须在0到100之间。

```
1 create table SC(  
2     SID varchar(255) references student(SID) ,  
3     CID varchar(255) references course(CID) ,  
4     GRADE int check( grade between 0 and 100),  
5     primary key (SID, CID)  
6 ) ;
```

2.修改

1) 插入数据

```
1 insert into teacher values ('14001','Katz','CS',75000),  
2                             ('14002','Crick','Biology',72000),  
3                             ('14003','Gold','Physics',87000),  
4                             ('14004','Einstein','Physics',95000),  
5                             ('14005','Kim','CS',65000),  
6                             ('14006','Wu','Finance',90000),  
7                             ('14007','Brandt','CS',65000),  
8                             ('14008','Singh','Finance',80000) ;  
9  
10 insert into student values ('S1', 'Wangfeng', 'Physics', 20, 'M'),  
11                             ('S2', 'Liu fang', 'Physics', 19, 'M'),  
12                             ('S3', 'Chen yun', 'CS', 22, 'M'),  
13                             ('S4', 'Wu kai', 'Finance', 19, 'M'),  
14                             ('S5', 'Liu li', 'CS', 21, 'F'),  
15                             ('S6', 'Dongqing', 'Finance', 18, 'F'),  
16                             ('S7', 'Li', 'CS', 19, 'F'),  
17                             ('S8', 'Chen', 'CS', 21, 'F'),  
18                             ('S9', 'Zhang', 'Physics', 19, 'M'),  
19                             ('S10', 'Yang', 'CS', 22, 'F'),  
20                             ('S11', 'Wang', 'CS', 19, 'F') ;  
21  
22 insert into course values ('C1', 'DB', 'CS', 2.0, 'Li'),  
23                             ('C2', 'maths', 'Mathematics', 2.0, 'Ma'),  
24                             ('C3', 'chemistry', 'Chemistry', 2.5, 'Zhou'),  
25                             ('C4', 'physics', 'Physics', 1.5, 'Shi'),  
26                             ('C5', 'OS', 'CS', 2.0, 'Wen'),  
27                             ('C6', 'Database', 'CS', 2.0, 'Katz'),  
28                             ('C7', 'Algorithm', 'CS', 2.5, 'Gold'),  
29                             ('C8', 'Java', 'CS', 1.5, 'Einstein'),  
30                             ('C9', 'Marketing', 'Finance', 2.0, 'Wu') ;  
31  
32 insert into sc values ('S1', 'C1', 70),  
33                       ('S1', 'C3', 81),  
34                       ('S2', 'C4', 92),  
35                       ('S2', 'C2', 85),  
36                       ('S3', 'C1', 65),
```

```
37 | ('S3', 'C5', 57),
38 | ('S4', 'C1', 87),
39 | ('S5', 'C4', 83) ;
```

2) 修改student表中姓名为'Zhang'且系信息错填为'Physics'的同学的系为'CS'的语句:

```
1 | update student
2 | set DEPT='CS'
3 | where SNAME = 'Zhang' and DEPT = 'Physics' ;
```

该语句中，通过where关键字指定了符合条件的行（即姓名为'Zhang'且系信息错填为'Physics'的同学），并使用set关键字指定了要修改的列和新的值。

3) 删除teacher表中属于Finance学院的教师信息的语句:

```
1 | delete from teacher
2 | where DEPT = 'Finance' ;
```

该语句中，使用delete关键字删除符合条件（即属于Finance学院的教师信息）的行。

4) 将teacher表中工资低于或等于70000的教师增长10%的工资，工资高于70000的教师增长5%的工资的语句:

```
1 | update teacher
2 | set salary = case
3 |         when salary <= 70000 then salary * 1.1
4 |         else salary * 1.05
5 | end;
```

该语句中，使用了case语句来根据不同的条件进行不同的操作，实现了对工资的增长。

3.基本数据查询

1) 基于teacher表，找出“物理系Physics和生物系Biology”所有教师的名字和工资的语句:

```
1 | select tname, salary
2 | from teacher
3 | where dept = 'Physics' or dept = 'Biology' ;
```

该语句中，使用了where关键字根据条件过滤出所需数据，并通过select关键字选择要查询的列。

| | tname | salary |
|---|----------|--------|
| 1 | Crick | 75600 |
| 2 | Gold | 91350 |
| 3 | Einstein | 99750 |

2) 基于teacher表，列出教师所在的各个系名，要求每个系名仅出现一次的语句：

```
1 SELECT DISTINCT dept
2 FROM teacher;
```

该语句中，使用了distinct关键字来去重，排除重复的系名。

| | dept |
|---|---------|
| 1 | Physics |
| 2 | Biology |
| 3 | CS |

3) 基于teacher表，希望查看“若将每位教师的工资提高20%后”各位教师的姓名和工资是多少的语句：

```
1 SELECT tname, salary*1.2
2 FROM teacher;
```

该语句中，通过select关键字将提高工资后的结果输出到结果集，使用计算公式将原工资乘以1.2而得到提高后的工资。

| | tname | ?column? |
|---|----------|----------|
| 1 | Katz | 94500.0 |
| 2 | Crick | 90720.0 |
| 3 | Gold | 109620.0 |
| 4 | Einstein | 119700.0 |
| 5 | Kim | 85800.0 |
| 6 | Brandt | 85800.0 |

4) 基于表student、SC和course，查看到计算机系CS的每位学生所选课程的所有信息，包括学生姓名、所在系、课程名称、课程学分的情况的语句：

```
1 SELECT s.sname, s.dept, c.cname, c.credits
2 FROM student s
3 JOIN SC ON s.sid = SC.sid
4 JOIN course c ON SC.cid = c.cid
5 WHERE s.dept = 'CS' ;
```

该语句中，使用了join关键字将三张表连接在一起，使用了where关键字根据条件过滤出符合要求的数据。

| | sname | dept | cname | credits |
|---|----------|------|---------|---------|
| 1 | Chen yun | CS | DB | 2.0 |
| 2 | Chen yun | CS | OS | 2.0 |
| 3 | Liu li | CS | physics | 1.5 |

4.复杂数据查询

1) 查询全体学生的姓名、年龄的语句：

```
1 SELECT sname, age
2 FROM student ;
```

该语句中，通过select关键字选择要查询的列，从student表中查询出所有学生的姓名和年龄信息。

| | sname | age |
|----|----------|-----|
| 1 | Wangfeng | 20 |
| 2 | Liu fang | 19 |
| 3 | Chen yun | 22 |
| 4 | Wu kai | 19 |
| 5 | Liu li | 21 |
| 6 | Dongqing | 18 |
| 7 | Li | 19 |
| 8 | Chen | 21 |
| 9 | Yang | 22 |
| 10 | Wang | 19 |
| 11 | Zhang | 19 |

2) 查询所有选修过课的学生的学号的语句：

```
1 SELECT DISTINCT sid
2 FROM SC
3 WHERE grade < 60;
```

该语句中，使用了distinct关键字去重，排除重复的学号。

| | sname | age |
|---|----------|-----|
| 1 | Wangfeng | 20 |
| 2 | Liu li | 21 |
| 3 | Liu fang | 19 |
| 4 | Chen yun | 22 |
| 5 | Wu kai | 19 |

3) 查询考试成绩低于60分的学生的学号的语句：

```
1 SELECT DISTINCT sid
2 FROM SC
3 WHERE grade < 60;
```

该语句中，使用了distinct关键字去重，排除重复的学号，并在where语句中添加条件限制。

| | sid |
|---|-----|
| 1 | S3 |

4) 查询年龄在20至23之间的学生姓名、性别和年龄的语句:

```
1 SELECT sname, gender, age
2 FROM student
3 WHERE age BETWEEN 20 AND 23;
```

该语句中, 使用了between...and...的语法结构, 指定age的取值范围。

| | sname | gender | age |
|---|----------|--------|-----|
| 1 | Wangfeng | M | 20 |
| 2 | Chen yun | M | 22 |
| 3 | Liu li | F | 21 |
| 4 | Chen | F | 21 |
| 5 | Yang | F | 22 |

5) 查询所有姓liu的学生的学号、姓名和年龄的语句:

```
1 SELECT sid, sname, age
2 FROM student
3 WHERE sname LIKE 'Liu%';
```

该语句中, 使用了like关键字指定模糊查询的匹配规则。

| | sid | sname | age |
|---|-----|----------|-----|
| 1 | S2 | Liu fang | 19 |
| 2 | S5 | Liu li | 21 |

6) 查询学习C1课程的学生最高分数的语句:

```
1 SELECT MAX(grade)
2 FROM sc
3 WHERE cid = 'C1' ;
```

该语句中, 使用了max函数来查询最高分数, 并在where语句中添加条件限制。

| | max |
|---|-----|
| 1 | 87 |

7) 查询各个课程号与相应的选课人数的语句:

```
1 SELECT cid, count(*) as num
2 FROM sc
3 GROUP BY cid ;
```

该语句中, 使用了count函数对相应的课程号进行计数, 并通过group by关键字将数据按照cid进行分组。

| | cid | num |
|---|-----|-----|
| 1 | C4 | 2 |
| 2 | C5 | 1 |
| 3 | C3 | 1 |
| 4 | C2 | 1 |
| 5 | C1 | 3 |

8) 查询选修C3课程的学生的姓名的语句：

```

1 SELECT sname
2 FROM sc JOIN student as s ON sc.sid = s.sid
3 WHERE sc.CID = 'C3' ;

```

该语句中，使用了join连接两张表，并通过where关键字限制CID为'C3'的行，从而得到选修C3课程的学生姓名。

| | sname |
|---|----------|
| 1 | Wangfeng |

9) 查询每一门课程的平均成绩的语句：

```

1 SELECT cid, AVG(grade)
2 FROM sc
3 GROUP BY cid;

```

该语句中，使用了avg函数计算平均成绩，并通过group by关键字将数据按照cid进行分组。

| | cid | avg |
|---|-----|---------------------|
| 1 | C4 | 87.5000000000000000 |
| 2 | C5 | 57.0000000000000000 |
| 3 | C3 | 81.0000000000000000 |
| 4 | C2 | 85.0000000000000000 |
| 5 | C1 | 74.0000000000000000 |

5.选做题目

学生只能选择自己学院开设的课程。发现CS学院有的同学选择了其他学院开设的课程。在SC表中删除这些错选的记录。

```

1  DELETE FROM sc
2  WHERE sc.sid in
3  (
4      SELECT sid
5      from student
6      where dept = 'CS'
7  ) and sc.cid not in
8  (
9      select cid
10     from course
11     where dept = 'CS'
12 );
13

```

删除的学生

| | sc.SID | CID | GRADE | s.SID | SNAME | 1 DEPT | AGE | GENDER |
|---|--------|-----|-------|-------|--------|--------|-----|--------|
| 1 | S5 | C4 | 83 | S5 | Liu li | CS | 21 | F |

该 SQL 语句通过两个子查询来筛选出需要删除的记录。第一个子查询选择了 CS 学院内的学生 ID，第二个子查询选择了不属于 CS 学院的课程 ID。最终主查询会根据这两个条件组成的逻辑进行删除操作。