

《数据库系统》实验报告

实验题目

基本 SQL 操作

一、实验目的

1. 了解 GaussDB 数据库的基本操作
- 2、掌握基本的数据库创建、数据库表创建、基本查询。
- 3、学习并掌握多条件查询和多表查询。

二、实验项目内容

1. 基本表操作

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：

teacher (TID, TNAME, DEPT, SALARY)

中文语义：教师（教工号，姓名，系，薪水）

student (SID, SNAME, DEPT, AGE, GENDER)

中文语义：学生（学号，姓名，系，年龄，性别）

course (CID, CNAME, DEPT, CREDITS, TEACHER)

中文语义：课程（课程号，课程名，系，学分，任课教师）

SC (SID, CID, GRADE)

中文语义：学习（学号，课程号，成绩）

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	Mathemat ics	2.0	Ma
C3	chemistry	Chemistr y	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）

1. 基本表操作

新建数据库

* 数据库来源 GaussDB 请输入实例名称 Q C

数据库实例	数据库引擎版本	实例类型	状态
<input checked="" type="radio"/> gauss-cqu-jsj2020-01	GaussDB 2.7	主备	正常

5 总条数: 1 < 1 >

* 数据库名称

* 登录用户名

* 密码 测试连接

☒ 记住密码 同意DAS使用加密方式记住密码

描述

定时采集 ☒ 若不开启，DAS只能实时的从数据库获取结构定义数据，将会影响数据库实时性能。

确定 取消

1) 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。以 teacher 为例：

```
CREATE TABLE teacher (  
    T_ID INT PRIMARY KEY,  
    TNAME VARCHAR(20),  
    DEPT VARCHAR(20),  
    SALARY INT  
);  
  
INSERT INTO teacher (T_ID, TNAME, DEPT, SALARY) VALUES
```

```
(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);
```

-----开始执行-----

【拆分SQL完成】：将执行SQL语句数量：（2条）

【执行SQL：（1）】

```
CREATE TABLE teacher (
    T_ID INT PRIMARY KEY,
    TNAME VARCHAR(20),
    DEPT VARCHAR(20),
    SALARY INT
);
```

执行成功，耗时：[8ms.]

warning:

CREATE TABLE / PRIMARY KEY will create implicit index "teacher_pkey" for table "teacher"

【执行SQL：（2）】

```
INSERT INTO teacher (T_ID, TNAME, DEPT, SALARY) VALUES
(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);
```

执行成功，耗时：[5ms.]

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' ；

```
UPDATE student
SET DEPT = 'CS'
WHERE SNAME = 'Zhang' AND DEPT = 'Physics';
```

-----开始执行-----

【拆分SQL完成】：将执行SQL语句数量：（1条）

【执行SQL：（1）】

```
UPDATE student
SET DEPT = 'CS'
WHERE SNAME = 'Zhang' AND DEPT = 'Physics';
```

执行成功，耗时：[5ms.]

3) 删除 teacher 表中，属于 Finance 学院的教师信息；

```
DELETE FROM teacher
WHERE DEPT = 'Finance';
```

-----开始执行-----

【拆分SQL完成】：将执行SQL语句数量：（1条）

【执行SQL：（1）】

```
DELETE FROM teacher
WHERE DEPT = 'Finance';
执行成功，耗时：[6ms.]
```

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。

```
UPDATE teacher
SET SALARY = CASE
    WHEN SALARY <= 70000 THEN SALARY * 1.1
    ELSE SALARY * 1.05
END;
```

-----开始执行-----

【拆分SQL完成】：将执行SQL语句数量：（1条）

【执行SQL：（1）】

```
UPDATE teacher
SET SALARY = CASE
    WHEN SALARY <= 70000 THEN SALARY * 1.1
    ELSE SALARY * 1.05
END;
执行成功，耗时：[8ms.]
```

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；

```
SELECT TNAME, SALARY
FROM teacher
WHERE DEPT = 'Physics' OR DEPT = 'Biology';
```

SQL执行记录 消息 结果集1 X 覆盖模式 ②

以下是SELECT TNAME, SALARY FROM teacher WHERE DEPT = 'Physics' ... ① 该表不可编辑。

	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；

```
SELECT DISTINCT dept
FROM teacher;
```

SQL执行记录 消息 结果集1 X 覆盖模式

以下是SELECT DISTINCT dept FROM teacher的执行结果集 该表不可编辑。

	dept
1	Physics
2	CS
3	Biology

复制行 复制列 列设置

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;

```
SELECT TNAME, SALARY*1.2 as NewSALARY
FROM teacher;
```

SQL执行记录 消息 结果集1 X 覆盖模式

以下是SELECT TNAME, SALARY*1.2 as NewSALARY FROM teacher的执行... 该表不可编辑。

	tname	newsalary
1	Katz	94500.0
2	Crick	90720.0
3	Gold	109620.0
4	Einstein	119700.0
5	Kim	85800.0
6	Brandt	85800.0

复制行 复制列 列设置

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。

```
SELECT student.SNAME, student.DEPT, course.CNAME, course.CREDITS
FROM student JOIN SC ON student.S_ID = SC.S_ID JOIN course ON SC.CID =
course.CID
WHERE student.DEPT = 'CS';
```

SQL执行记录 消息 结果集1 X 覆盖模式

以下是SELECT student.SNAME, student.DEPT, course.CNAME, course.CRE... 该表不可编辑。

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

复制行 复制列 列设置

3. 复杂数据查询

1) 查询全体学生的姓名、年龄;

```
SELECT SNAME, AGE
FROM student;
```

SQL执行记录 消息 结果集1 X 覆盖模式

以下是SELECT SNAME, AGE FROM student的执行结果集 该表不可编辑。

	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

复制行 复制列 列设置

2) 查询所有选修过课的学生的学号;

```
SELECT DISTINCT S_ID
FROM SC;
```

SQL执行记录 消息 结果集1 X

覆盖模式 ①

以下是SELECT DISTINCT S_ID FROM SC的执行结果集

① 读表不可编辑。

复制行 复制列 列设置

	s_id
1	S2
2	S1
3	S3
4	S4
5	S5

3) 查询考试成绩低于 60 分的学生的学号;

```
SELECT DISTINCT S_ID
FROM SC
WHERE GRADE < 60;
```

SQL执行记录 消息 结果集1 X

覆盖模式 ①

以下是SELECT DISTINCT S_ID FROM SC WHERE GRADE < 60的执行结果集

① 读表不可编辑。

复制行 复制列 列设置

	s_id
1	S3

4) 查询年龄在 20 至 23 之间的学生姓名、性别和年龄;

```
SELECT SNAME, GENDER, AGE
FROM student
WHERE AGE BETWEEN 20 AND 23;
```

SQL执行记录 消息 结果集1 X

覆盖模式 ①

以下是SELECT SNAME, GENDER, AGE FROM student WHERE AGE BETW...

① 读表不可编辑。

复制行 复制列 列设置

	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;

```
SELECT S_ID, SNAME, AGE
FROM student
WHERE SNAME LIKE '%Liu%';
```

SQL执行记录 消息 结果集1 X

覆盖模式 ①

以下是SELECT S_ID, SNAME, AGE FROM student WHERE SNAME LIKE '%...

① 读表不可编辑。

复制行 复制列 列设置

	s_id	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;

```
SELECT MAX(GRADE)
FROM SC
WHERE CID = 'C1';
```

SQL执行记录 消息 结果集1 X

覆盖模式 ①

以下是SELECT MAX(GRADE) FROM SC WHERE CID = 'C1'的执行结果集

① 读表不可编辑。

复制行 复制列 列设置

	max
1	87

7) 查询各个课程号与相应的选课人数;

```
SELECT CID, COUNT(*) AS S_NUM
FROM SC
GROUP BY CID;
```

SQL执行记录 消息 结果集1 X 覆盖模式 ②

以下是SELECT CID, COUNT(*) AS S_NUM FROM SC GROUP BY CID,的执... ① 该表不可编辑。

复制行 复制列 列设置

	cid	s_num
1	C4	2
2	C5	1
3	C2	1
4	C3	1
5	C1	3

8) 查询选修 C3 课程的学生的姓名;

```
SELECT student.SNAME FROM student, SC
WHERE student.S_ID = SC.S_ID AND SC.CID = 'C3';
```

SQL执行记录 消息 结果集1 X 覆盖模式 ②

以下是SELECT student.SNAME FROM student, SC WHERE student.S_ID = ... ① 该表不可编辑。

复制行 复制列 列设置

	sname
1	Wangfeng

9) 查询每一门课程的平均成绩。

```
SELECT CID, AVG(GRADE) AS AVG_GRADE
FROM SC
GROUP BY CID;
```

SQL执行记录 消息 结果集1 X 覆盖模式 ②

以下是SELECT CID, AVG(GRADE) AS AVG_GRADE FROM SC GROUP BY CID,的执... ① 该表不可编辑。

复制行 复制列 列设置

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

4、学生自主上机实验内容（选做）：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

```
DELETE FROM SC
WHERE S_ID IN (
    SELECT student.S_ID FROM student
    WHERE student.DEPT = 'CS'
) AND CID NOT IN (
    SELECT course.CID FROM course
    WHERE course.DEPT = 'CS'
);
```

-----开始执行-----

【拆分SQL完成】：将执行SQL语句数量：（1条）

【执行SQL：（1）】

```
DELETE FROM SC
WHERE S_ID IN (
    SELECT student.S_ID FROM student
    WHERE student.DEPT = 'CS'
) AND CID NOT IN (
    SELECT course.CID FROM course
    WHERE course.DEPT = 'CS'
);
执行成功，耗时：[8ms.]
```

四、实验结果及分析和（或）源程序调试过程

结果分析

实验结果截图见上一节，分析如下：本次实验是基本 SQL 操作，包括基本表操作、数据查询、复杂数据查询与修改等。实验在 GaussDB 数据库上创建了 teacher、student、course 和 SC 四个表，并向其中插入了实验要求的数据。通过实验，我成功实践了 SQL 语言的基本操作，学会了如何在实践中创建表、插入和查询数据。

在创建表时，主要使用`CREATE TABLE`语句定义表的数据成员，使用`INSERT VALUES`语句插入数据。在基本数据查询中，我学会了使用 SELECT 语句查询数据，并可以根据需要使用 WHERE 子句进行筛选，以及使用 DISTINCT 关键字去除重复。除此之外，还可以使用 ORDER BY 语句对待查询数据进行分组。在复杂数据查询中，使用 JOIN 语句进行拼接实现复杂查询。通过这些基本语句的复合与技巧的引入，可以更加灵活地查询数据，满足不同的需求。

通过本次实验，我更加熟练地掌握了 SQL 语言的基本操作以及数据库的使用方法。

调试过程

1. 错误：创建数据库时无法成功创建；
解决方法：复制进来的密码前含有空格，无法通过验证；
2. 错误：创建 teacher 表时属性 TID 与内置关键字冲突；

SQL执行中发生错误，是否继续？

```
CREATE TABLE teacher ( TID INT PRIMARY KEY, TNAME  
VARCHAR(20), DEPT VARCHAR(20), SALARY INT );(ERROR:  
column name "tid" conflicts with a system column name)
```

中止执行

继续执行

解决方法：更换属性名即可；

3. 错误：由于此前创建出现问题，后来重新创建时出现问题；

解决方法：表已存在，需要先删除：

```
1 DROP TABLE teacher  
2 DROP TABLE student  
3 DROP TABLE course  
4 DROP TABLE SC
```

4. 错误：定义表的数据结构时数据类型与压入数据冲突；

解决方法：修改表的数据类型，使其与应有数据一致；