北京航空航天大學
BEIHANG UNIVERSITY

《数据库系统原理》实验

GaussDB 实验报告书

学号 杜启嵘

姓名 22373362

2024 年 10 月 12 日

任务一：建表并完善约束

1. 写出**建表时**每个字段的类型和建表时所使用的约束。

表 1-1 学生表

字段名	类型	约束
Sid	char(8)	PRIMARY KEY
Sname	varchar(20)	NOT NULL
Sgender	char(2)	NOT NULL
Sage	int	NOT NULL CHECK (Sage <= 200)

表 1-2 课程表

字段名	类型	约束
Cid	char(8)	PRIMARY KEY
Cname	varchar(30)	NOT NULL
Ctype	varchar(10)	NOT NULL
Ccredit	int	NOT NULL CHECK (Ccredit <= 99)
Tid	char(9)	FOREIGN KEY (`Tid`) REFERENCES Teacher(`Tid`)

表 3 教师表

字段名	类型	约束
Tid	char(9)	PRIMARY KEY
Tname	varchar(20)	NOT NULL
Tgender	char(2)	NOT NULL
Tage	int	NOT NULL CHECK (Tage <= 999)
Temail	varchar(127)	
Tphone	varchar(30)	UNIQUE

表 4 学生-课程关系表

字段名	类型	约束
Sid	char(8)	PRIMARY KEY (`Sid`,`Cid`), ,FOREIGN KEY (`Sid`) REFERENCES Student(`Sid`)

Cid	char(8)	PRIMARY KEY (`Sid`,`Cid`), FOREIGNKEY (`Cid`) REFERENCES Course(`Cid`)
Score	int	CHECK (Score <= 100)

2. 列出表中应该为外键的字段。

   (1) Course 表中的 Tid

   (2) SC 表中的 Sid 和 Cid

3. 请提交建立约束的 sql 语句。

在建立表时同时声明字段的约束：

```sql
CREATE TABLE `Student` (
    `Sid` CHAR (8) PRIMARY KEY,
    `Sname` VARCHAR (20) NOT NULL,
    `Sgender` CHAR (2) NOT NULL,
    `Sage` INT UNSIGNED NOT NULL CHECK (`Sage` <= 200)
);

CREATE TABLE `Teacher` (
  `Tid` CHAR (9) PRIMARY KEY ,
  `Tname` VARCHAR (20) NOT NULL,
  `Tgender` CHAR (2) NOT NULL,
  `Tage` INT UNSIGNED NOT NULL CHECK (`Tage` <= 999),
  `Temail` VARCHAR (127),
  `Tphone` VARCHAR (30) UNIQUE
);

CREATE TABLE `Course` (
  `Cid` CHAR(8) PRIMARY KEY ,
  `Cname` VARCHAR (30) NOT NULL,
  `Ctype` VARCHAR (10) NOT NULL,
  `Ccredit` INT UNSIGNED NOT NULL CHECK (`Ccredit` <= 99),
  `Tid` CHAR (9),
  FOREIGN KEY (`Tid`) REFERENCES `Teacher`(`Tid`)
);

CREATE TABLE `SC` (
  `Sid` CHAR(8),
  `Cid` CHAR(8),
  `Score` INT UNSIGNED CHECK (`Score` <= 100),
  PRIMARY KEY(`Sid`,`Cid`),
  FOREIGN KEY (`Sid`) REFERENCES `Student`(`Sid`),
  FOREIGN KEY (`Cid`) REFERENCES `Course`(`Cid`)
);
```

任务二：插入数据

请阐述云数据库、分布式数据库的特性。

云数据库的特性

1. **弹性扩展**
    - **水平扩展**：可以通过增加更多的数据库实例来处理更多的请求和数据量。
    - **垂直扩展**：可以通过增加单个数据库实例的资源（如 CPU、内存）来提升性能。
    - **自动扩展**：许多云数据库提供自动扩展功能，根据负载自动调整资源。
2. **高可用性**
    - **多副本机制**：数据在多个地理位置的副本之间同步，确保在一个副本故障时，其他副本可以继续提供服务。
    - **自动故障转移**：在检测到故障时，自动将流量切换到健康的副本，确保服务不中断。
    - **服务级别协议（SLA）**：云服务提供商通常提供高可用性的 SLA，保证一定的正常运行时间。
3. **按需付费**
    - **使用量计费**：根据实际使用的存储、计算和网络资源计费，避免了资源浪费。
    - **灵活定价**：提供多种定价模式，如按小时、按月或按年计费，用户可以根据需求选择最合适的方案。
4. **自动化管理**
    - **自动备份**：定期自动备份数据，确保数据安全。
    - **自动监控**：实时监控数据库性能和健康状态，提供告警和报告。
    - **自动更新**：自动应用安全补丁和版本更新，减少运维工作量。
5. **全球分布**
    - **多区域部署**：可以在全球多个数据中心部署数据库，提供低延迟的访问体验。
    - **数据主权**：支持将数据存储在特定的地理位置，以满足数据主权和合规要求。

分布式数据库的特性

1. **数据分片**
    - **水平分片**：将数据按行分布到不同的节点，每个节点存储部分数据，提升存储和处理能力。
    - **垂直分片**：将数据按列分布到不同的节点，每个节点存储部分列的数据，适用于特定查询优化。
2. **高可用性和容错性**
    - **数据复制**：在多个节点之间复制数据，确保在一个节点故障时，其他节点可以继续提供服务。
    - **冗余设计**：通过冗余设计，确保系统在部分节点故障时仍能正常运行。
3. **一致性**
    - **强一致性**：确保所有节点上的数据在任何时间点都是一致的，适用于对数据一致性要求高的应用。
    - **最终一致性**：允许数据在短时间内不一致，但最终会达到一致状态，适用于对一致性要求较低的应用。
4. **扩展性**
    - **水平扩展**：通过增加更多的节点来提升系统的处理能力和存储容量。
    - **无缝扩展**：支持在线扩展和缩减节点，确保系统在扩展过程中不中断服务。
5. **分布式事务**
    - **两阶段提交**：通过两阶段提交协议，确保跨节点的事务处理的一致性和完整性。
    - **分布式锁**：通过分布式锁机制，确保在多个节点之间的并发操作的一致性。
6. **负载均衡**
    - **请求分发**：通过负载均衡器将请求分发到不同的节点，确保系统的高性能和高可用性。
    - **动态调整**：根据节点的负载情况，动态调整请求分发策略，优化系统性能。
7. **数据分布**
    - **地理分布**：支持将数据分布在不同的地理位置，提供低延迟的访问体验。
    - **数据局部性**：通过数据局部性优化，减少跨节点的数据访问，提高查询性能。

插入数据 SQL 语句如下：

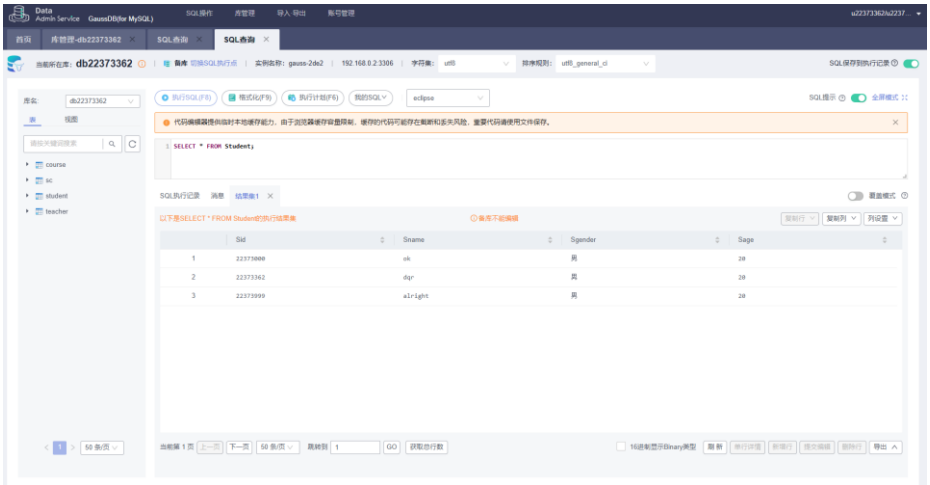
```sql
INSERT INTO student VALUES
("22373362", "dqr", "男", 20),
("22373000", "ok", "男", 20),
("22373999", "alright", "男", 20);

INSERT INTO teacher VALUES
("t00000001", "张莉", "女", 50, "zhangli@buaa.edu.cn", "10000000000"),
("t00000002", "沃天宇", "男", 46, "woty@buaa.edu.cn", "10000000001");

INSERT INTO course VALUES
("c0000001", "编译技术", "必修", 4, "t00000001"),
("c0000002", "Ruby语言设计", "选修", 4, "t00000002");

INSERT INTO SC VALUES
("22373362", "c0000001", 100),
("22373362", "c0000002", 100),
("22373000", "c0000001", 99),
("22373000", "c0000002", 99),
("22373999", "c0000001", 98),
("22373999", "c0000002", 98);
```

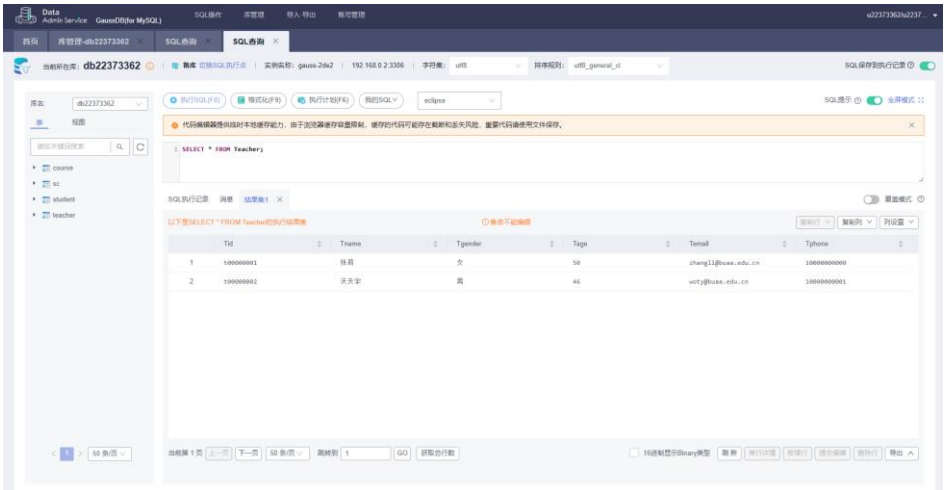
任务三：查询数据

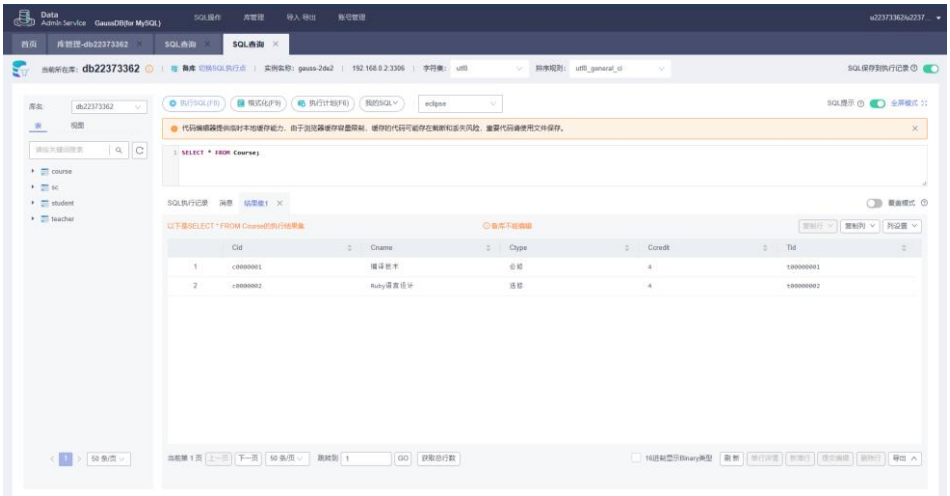
写出操作 1 和操作 2 的查询语句，以及操作 2 的查询条件 A、B、C，并提交查询结果截图。

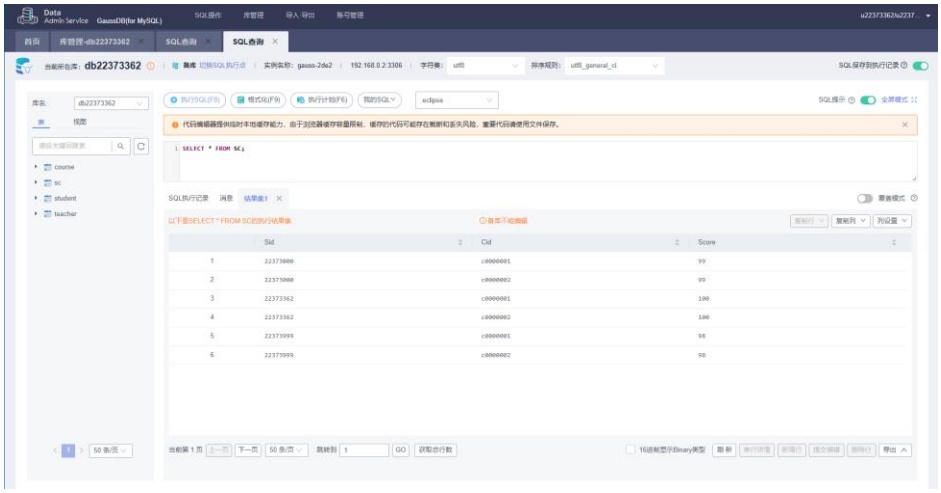
操作一：

```sql
SELECT * FROM Student;
SELECT * FROM Teacher;
SELECT * FROM Course;
SELECT * FROM SC;
```





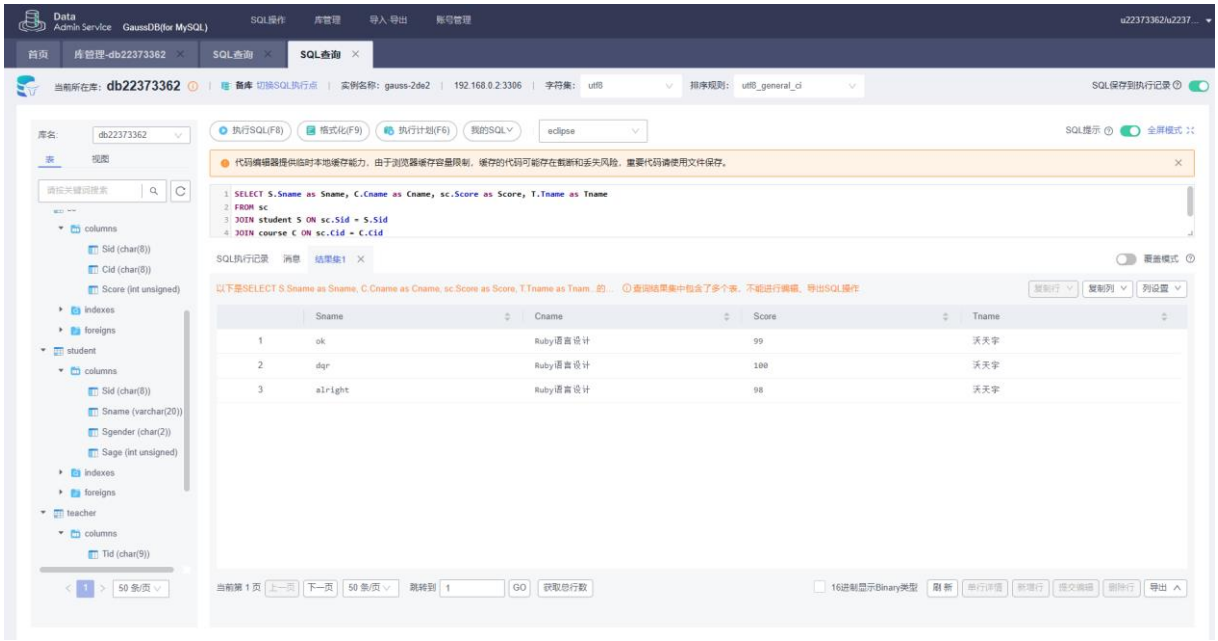




操作二：

设置条件为：A:男, B:1, C:98，SQL 操作语句为

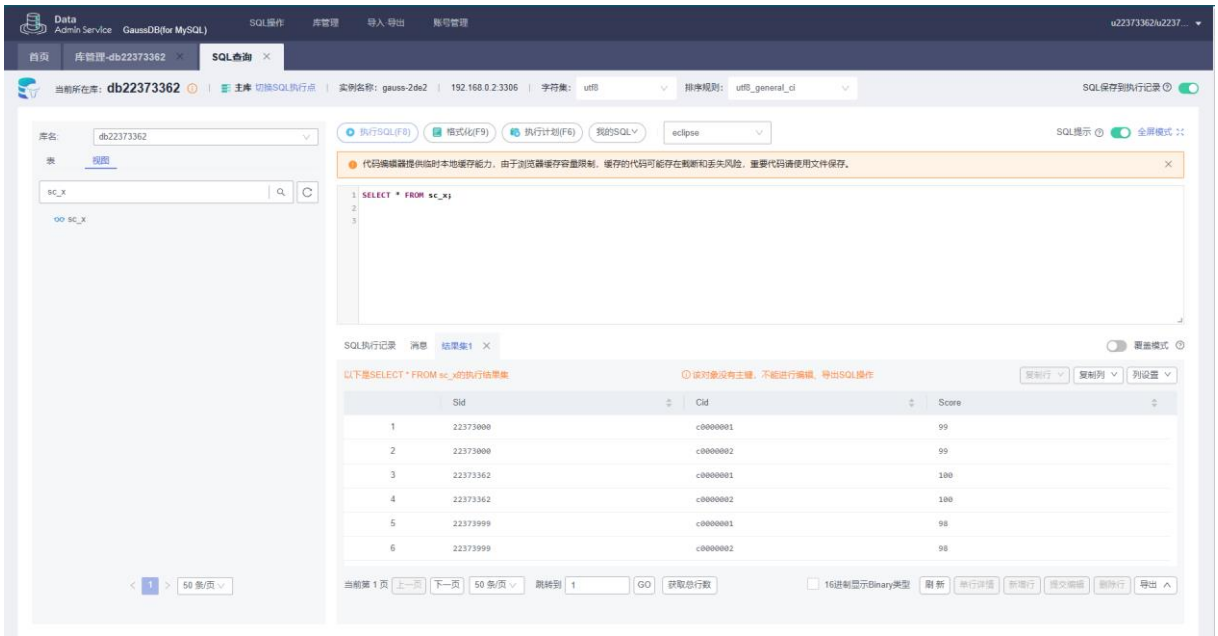
```
SELECT S.Sname as Sname, C.Cname as Cname, sc.Score as Score, T.Tname as Tname
FROM SC
JOIN Student S ON sc.Sid = S.Sid
JOIN Course C ON sc.Cid = C.Cid
JOIN Teacher T ON C.Tid = T.Tid
WHERE T.Tgender = "男" AND C.Ccredit >= 1 AND SC.Score >= 98;
```



任务四：建立视图和索引

1. 写出操作 1 中建立视图的语句，以及条件 X；在查询窗口中查询该视图并输出其所有列，提交查询结果截图。

```
CREATE VIEW SC_X AS
SELECT SC.Sid, SC.Cid, SC.Score
FROM SC
WHERE SC.Sid LIKE '22%';
```



2. 写出操作 2 中建立普通索引的语句。

```sql
CREATE INDEX `22373362`
ON Teacher(Tid, Tname, Tage);
```

teacher	2024-10-12 17:49:44	0（预估值）	16KB（预估值）	16KB（预估值）	utf8mb4	SQL查询 | 打开表 | 查看表详情 | 修改表 | 重命名 | 更多

列 虚拟列 索引 外键

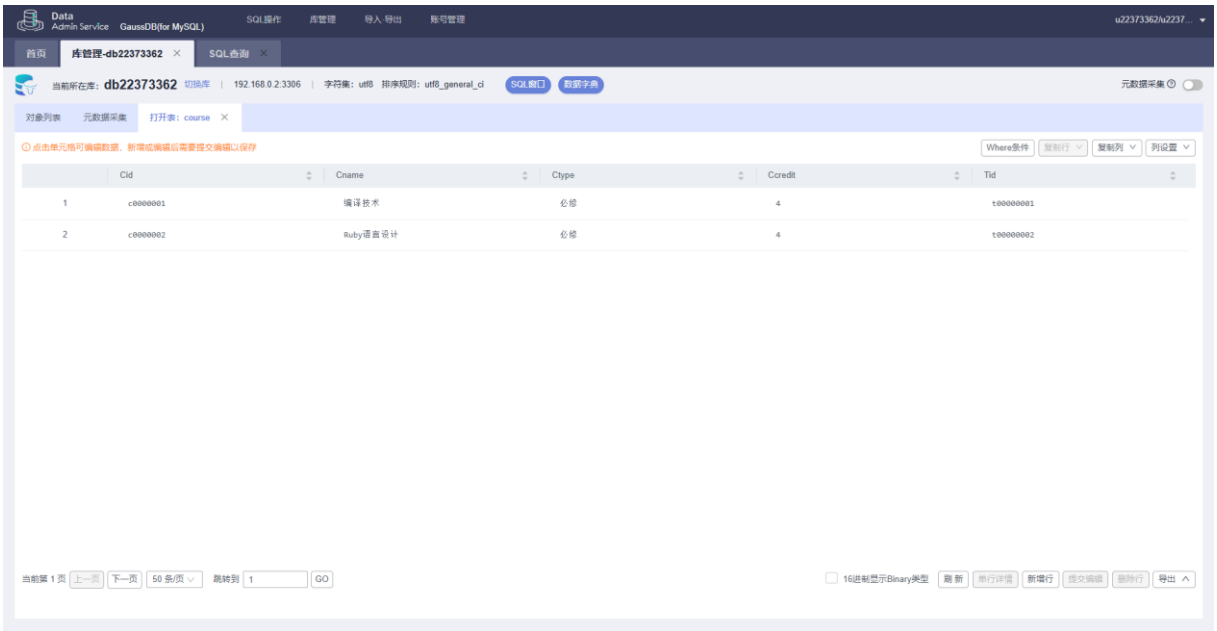
索引名	包含列	索引类型	索引方式
22373362	Tid,Tname,Tage	NORMAL	BTREE
Tphone	Tphone	UNIQUE	BTREE
PRIMARY	Tid	PRIMARY	BTREE

任务五：更新数据

1. 写出操作 1 和操作 2 中更新数据的语句，以及操作 2 的条件 Y、Z，并提交更新后的数据截图。

   操作一：

```sql
UPDATE course
SET Ctype = "必修"
WHERE Cid IN (
  SELECT Cid FROM (
    SELECT Cid FROM Course
    ORDER BY Cid DESC
    LIMIT 2
  ) AS Template
);
```



Data Admin Service GaussDB(for MySQL) SQL操作 库管理 导入 导出 账号管理 u22373362/u2237...

首页 库管理-db22373362 SQL查询

当前所在库：db22373362 切换库 | 192.168.0.2:3306 | 字符集：utf8 排序规则：utf8_general_ci SQL窗口 数据字典 元数据采集

对象列表 元数据采集 打开表：course

点击单元格可编辑数据，新增或编辑后需要提交编辑以保存 Where条件 复制行 复制列 列设置

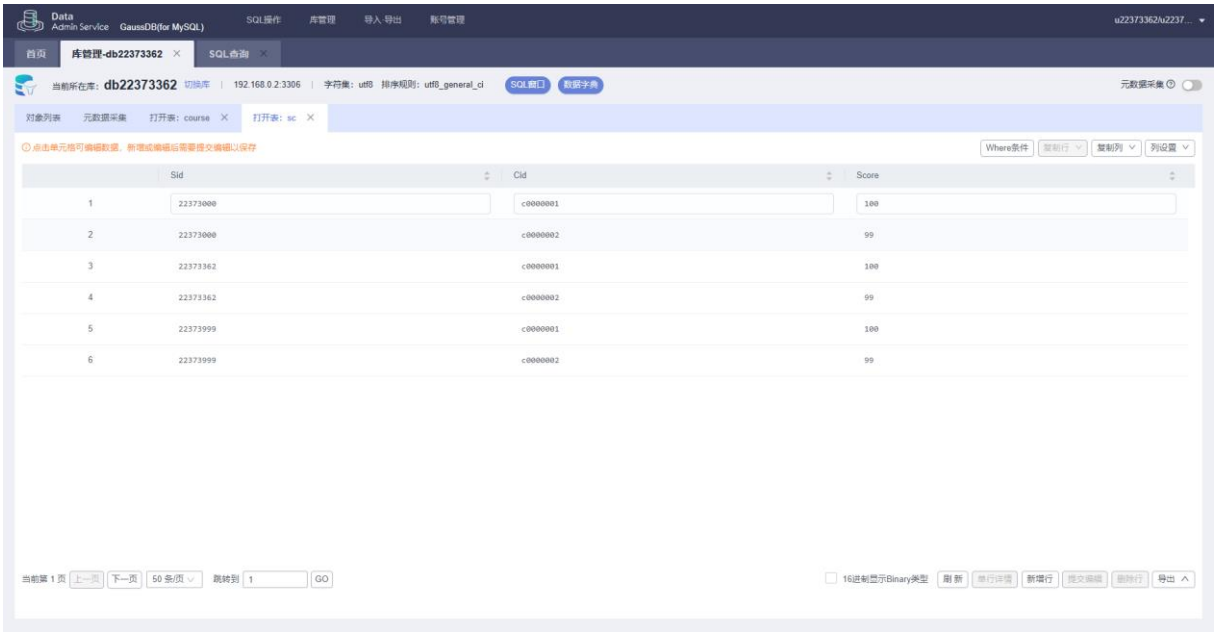
	Cid	Cname	Ctype	Ccredit	Tid
1	c0000001	编译技术	必修	4	t00000001
2	c0000002	Ruby语言设计	必修	4	t00000002

当前第 1 页 上一页 下一页 50 条/页 跳转到 1 GO 16进制显示Binary类型 刷新 单行详情 新增行 提交编辑 删除行 导出

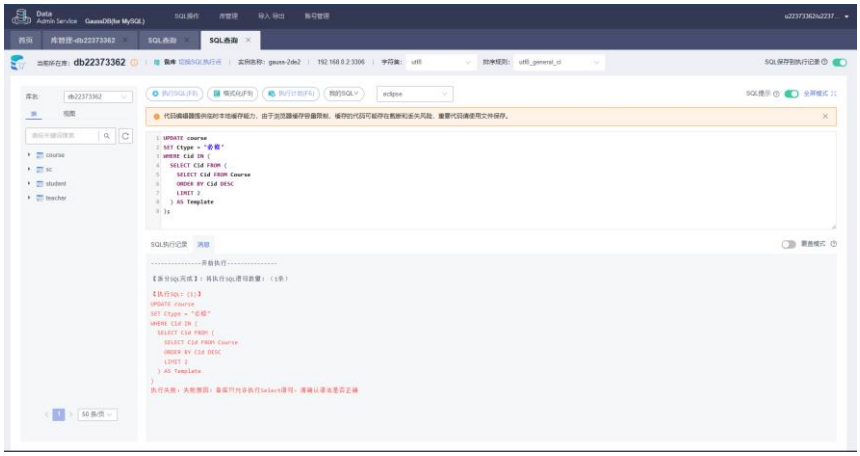
操作二：

条件 : Y:张莉, Z:100

```sql
UPDATE sc
SET Score = 100
WHERE Cid IN (
  SELECT Cid FROM (
    SELECT SC.Cid
    FROM course C
    JOIN sc SC ON C.Cid = SC.Cid
    JOIN teacher T ON C.Tid = T.Tid
    WHERE T.Tname = '张莉'
  ) as subquery
);
```



2. 将第 3 条操作的结果截图附上，并说明分布式数据库读写分离的好处。



在备库中只能进行 SELECT 操作(读操作)，不能进行 UPDATE 操作(写操作)，通过将读写操作分离，可以一定程度上保护数据库中数据的安全，便于高效操作数据库。

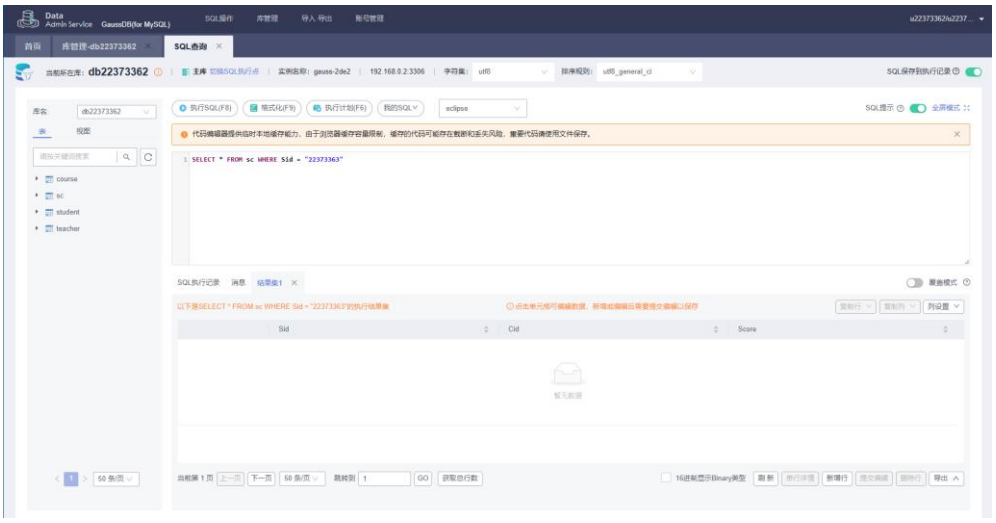
任务六：删除数据

1. 写出操作 1 中删除数据的语句，以及条件 A，并提交查询该同学修的课程及其成绩的语句及结果截图。

   条件 A：学号为 22373363

```
DELETE FROM student WHERE Sid = "22373363";
```

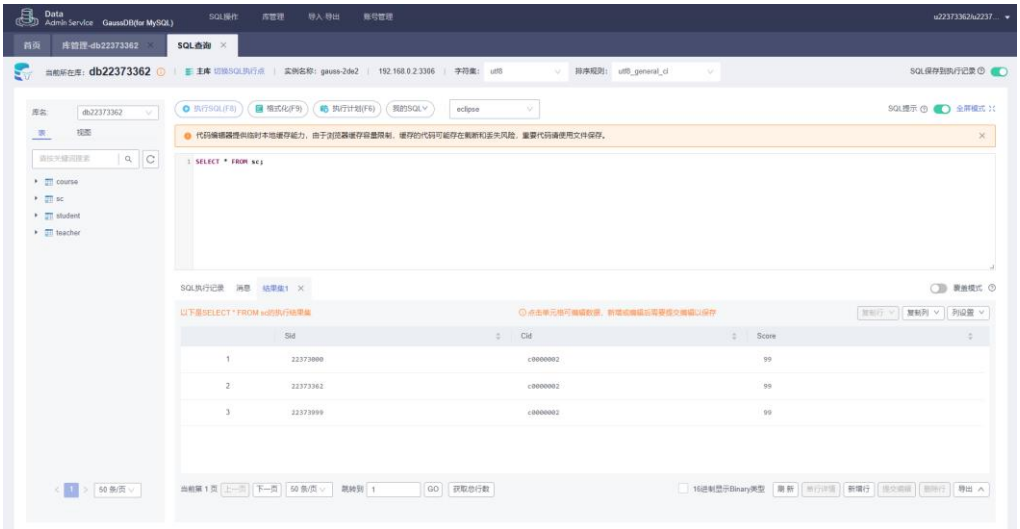
```
SELECT * FROM sc WHERE Sid = "22373363"
```



2. 写出操作 2 条中的删除语句，以及条件 B，并提交查询删除后的结果截图。

条件 B : 100

```
DELETE FROM sc WHERE Score >= 100;
```

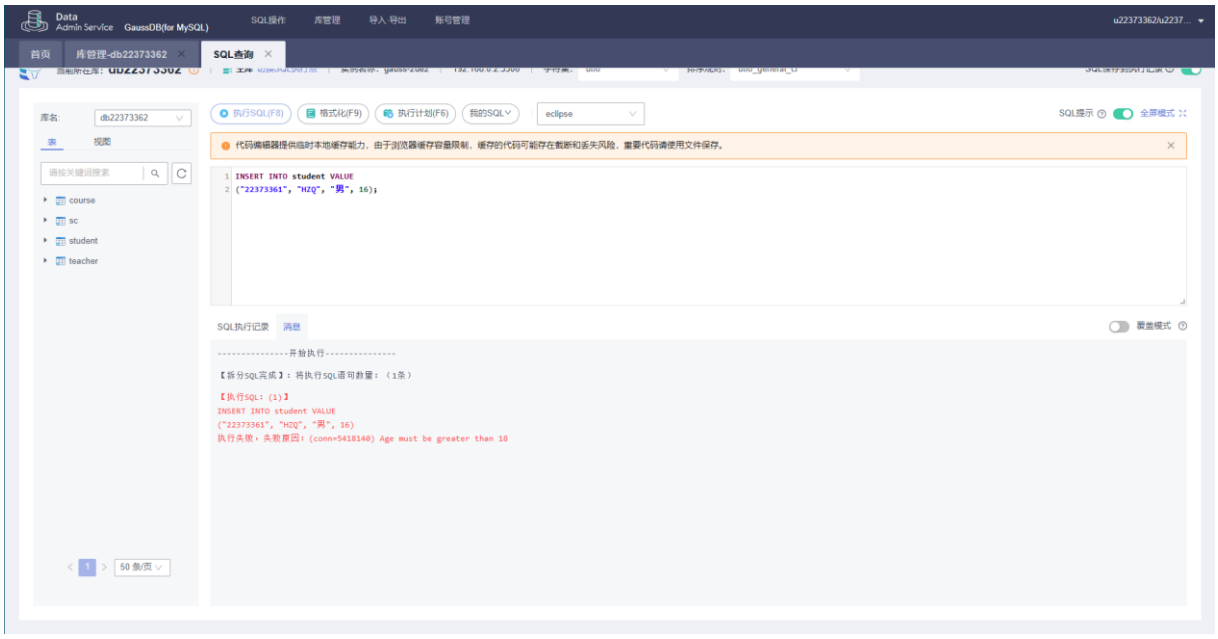


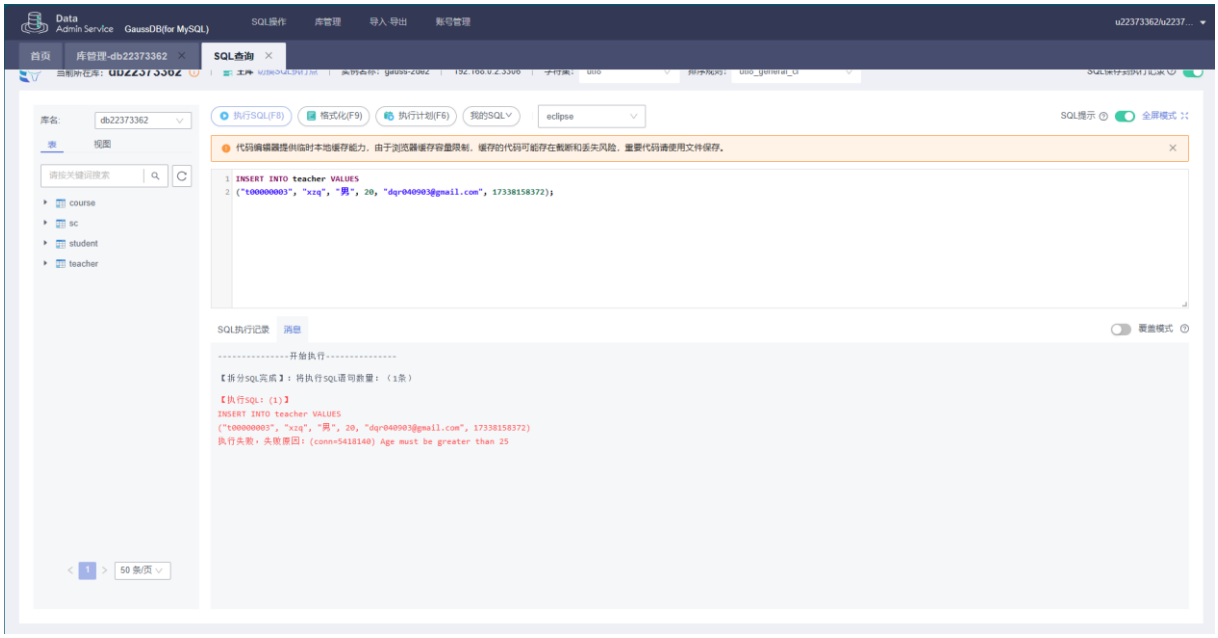
任务七：设置触发器

写出 1, 2 设置触发器的语句，以及条件 C、D，并提交插入数据失败的截图。

条件 C：年龄小于等于 18 的学生信息无法插入

```
DELIMITER //

CREATE TRIGGER before_student_insert
BEFORE INSERT ON student
FOR EACH ROW
BEGIN
    IF NEW.Sage <= 18 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Age must be greater than 18';
    END IF;
END;

//

DELIMITER ;
```



条件 D：年龄小于等于 25 的教师信息无法插入

```
DELIMITER //
CREATE TRIGGER before_teacher_insert
BEFORE INSERT ON Teacher
FOR EACH ROW
BEGIN
    IF NEW.Tage <= 25 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Age must be greater than 25';
    END IF;
END;
//
DELIMITER ;
```