

实验报告

姓名：胡瑞康

学号：22336087

数据库表结构

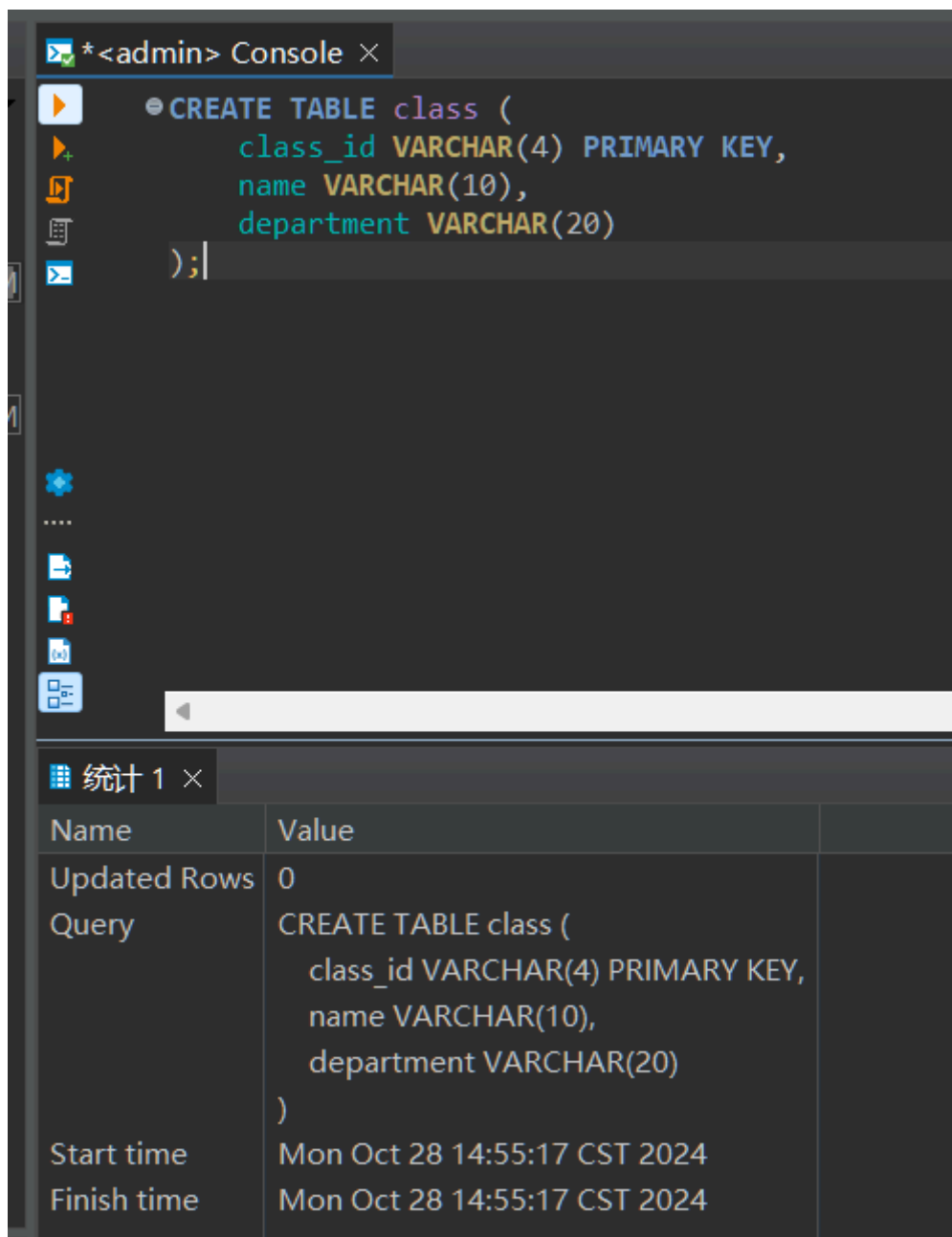
- **students (sid, sname, email, grade)**
学生表，包含学生的编号、姓名、邮箱和年级。
- **teachers (tid, tname, email, salary)**
教师表，包含教师的编号、姓名、邮箱和薪水。
- **courses (cid, cname, hour)**
课程表，包含课程的编号、名称和课时。
- **choices (no, sid, tid, cid, score)**
选课表，包含选课记录的编号、学生编号、教师编号、课程编号和成绩。

1

在school数据库中建立一张新表class，包括class_id(varchar(4)), name(varchar(10)), department(varchar(20))三个列，并约束class_id为主键。

按要求写下如下sql：

```
CREATE TABLE class (  
    class_id VARCHAR(4) PRIMARY KEY,  
    name VARCHAR(10),  
    department VARCHAR(20)  
);
```



2

创建事务T3，在事务中插入一个元组（0001，01CSC，CS），并在T3中嵌套创建事务T4，T4也插入和T3一样的元组，编写代码测试，查看结果。

事务可以通过BEGIN TRANSACTION、COMMIT TRANSACTION和ROLLBACK TRANSACTION来管理。嵌套事务可以通过SAVE TRANSACTION来实现。

```
-- 开始事务T3
BEGIN TRANSACTION T3;

-- 插入元组到class表
INSERT INTO class (class_id, name, department)
VALUES ('0001', '01CSC', 'CS');

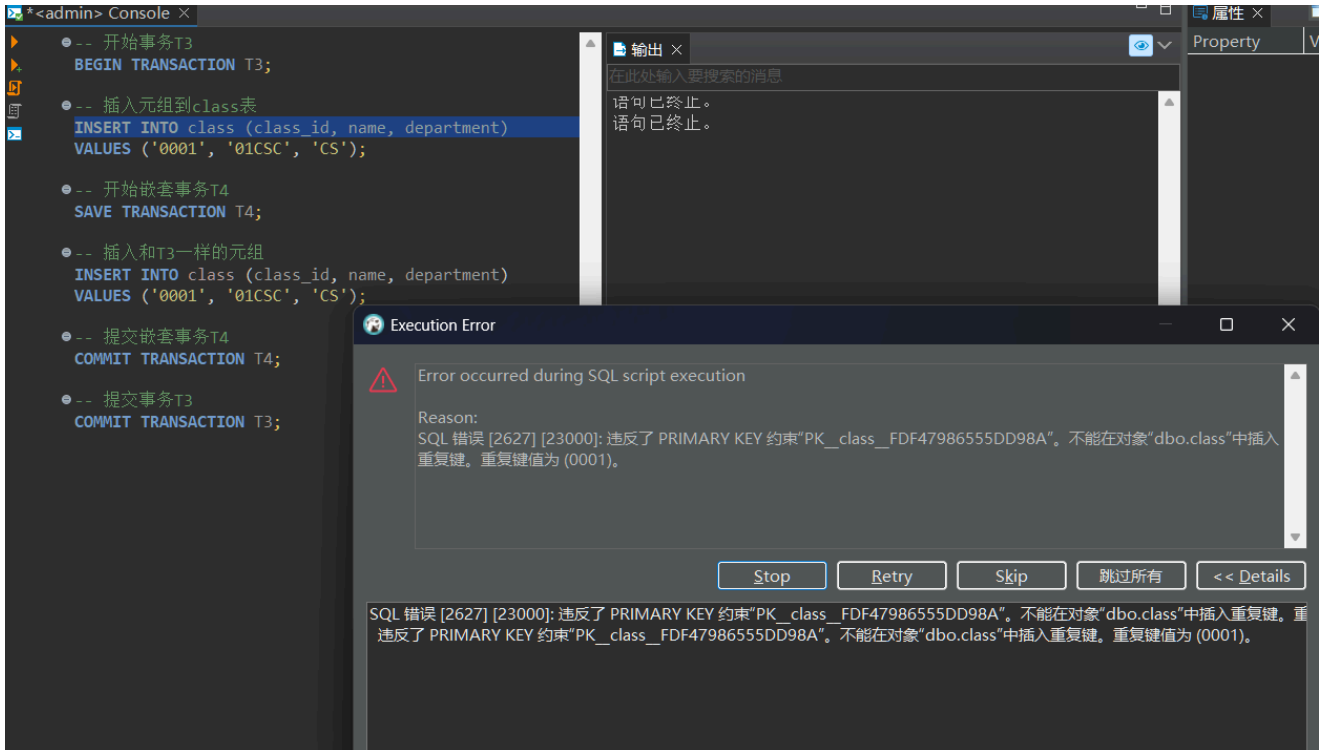
-- 开始嵌套事务T4
SAVE TRANSACTION T4;
```

```
-- 插入和T3一样的元组
INSERT INTO class (class_id, name, department)
VALUES ('0001', '01CSC', 'CS');

-- 提交嵌套事务T4
COMMIT TRANSACTION T4;

-- 提交事务T3
COMMIT TRANSACTION T3;
```

提示主键冲突，无法插入。



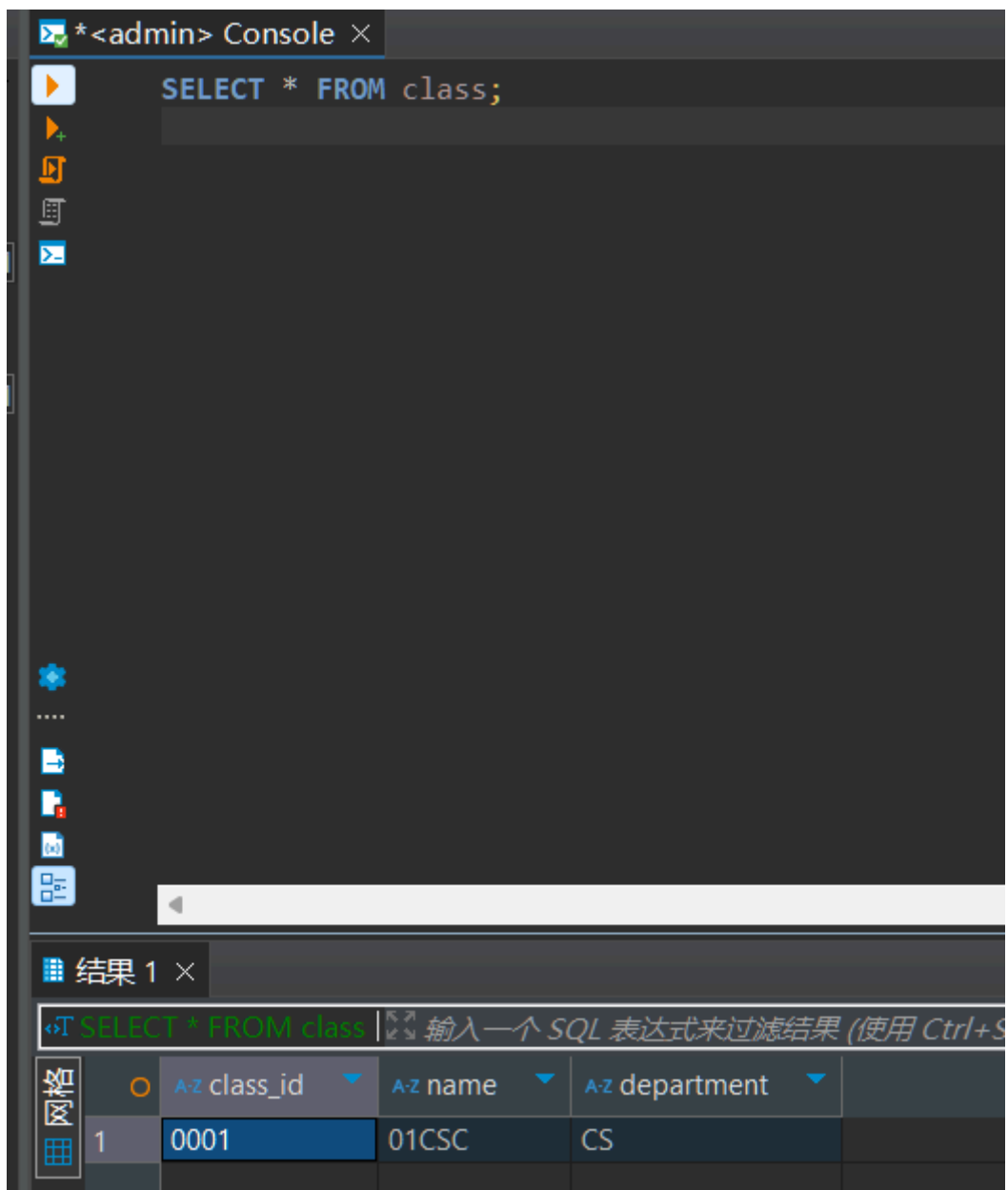
分析：

- 事务T3：由于嵌套事务T4的插入失败，T4会被回滚，但T3的插入操作仍然有效。
- 事务T4：由于主键冲突，T4的插入操作失败，T4会被回滚，不会影响T3的插入操作。

重新查看class表：

```
SELECT * FROM class;
```

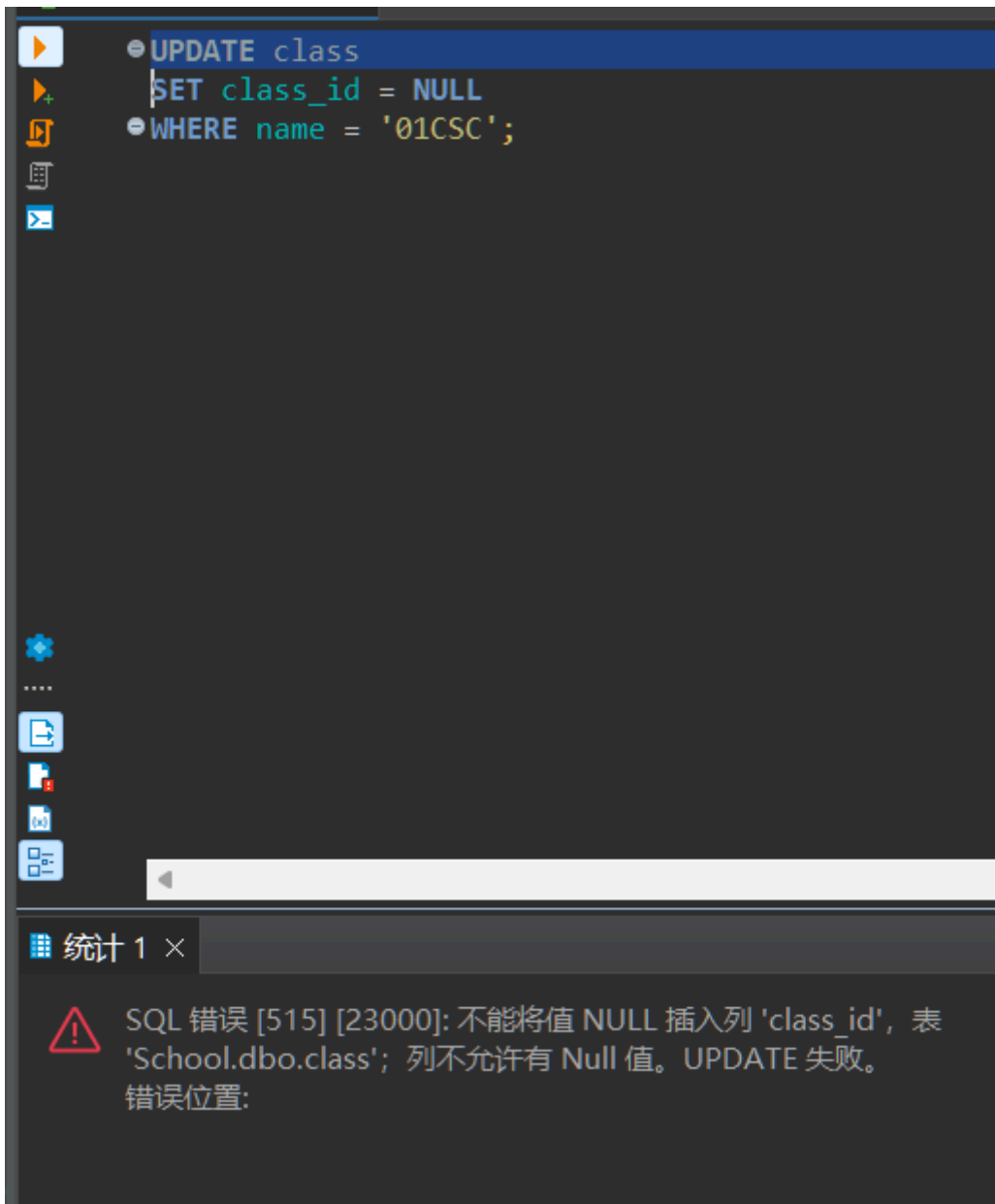
只有一条记录：



3

在表class中，尝试设置name=01CSC的记录的class_id 为NULL，查看结果

```
-- 尝试将class_id设置为NULL
UPDATE class
SET class_id = NULL
WHERE name = '01CSC';
```



结果:

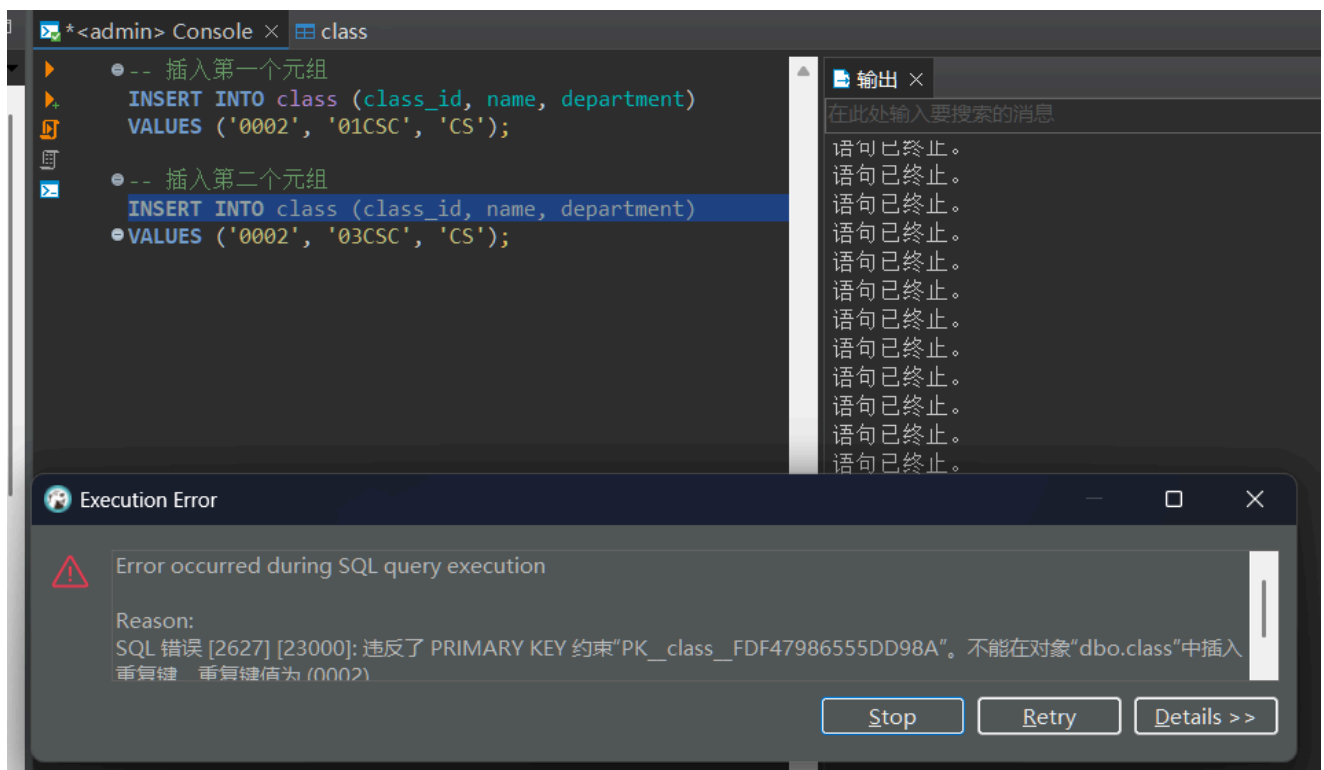
- 由于class_id是主键，主键不能为NULL，因此该操作会失败。

4

在表class中，不创建事务，插入两个元组（0002，01CSC，CS），（0002，03CSC，CS），然后查看表中有多少条记录，为什么？

```
-- 插入第一个元组
INSERT INTO class (class_id, name, department)
VALUES ('0002', '01CSC', 'CS');

-- 插入第二个元组
INSERT INTO class (class_id, name, department)
VALUES ('0002', '03CSC', 'CS');
```



结果:

- 由于class_id是主键，主键必须唯一，因此第二个插入操作会失败，表中仍然只有一条0002的记录。

	class_id	name	department
1	0001	01CSC	CS
2	0002	01CSC	CS

在表class中，创建事务，并设置开启回滚，然后插入两个元组（0003，03CSC。CS），（0001，03CSC，CS），查看结果，表中几条记录？

-- 开始事务

BEGIN TRANSACTION;

-- 插入第一个元组

INSERT INTO class (class_id, name, department)
VALUES ('0003', '03CSC', 'CS');

-- 插入第二个元组

INSERT INTO class (class_id, name, department)
VALUES ('0001', '03CSC', 'CS');

-- 回滚事务

ROLLBACK TRANSACTION;

*<admin> Console × class

```
-- 开始事务
BEGIN TRANSACTION;

-- 插入第一个元组
INSERT INTO class (class_id, name, department)
VALUES ('0003', '03CSC', 'CS');

-- 插入第二个元组
INSERT INTO class (class_id, name, department)
VALUES ('0001', '03CSC', 'CS');

-- 回滚事务
ROLLBACK TRANSACTION;
```

统计 1 ×

Name	Value
Queries	3
Updated Rows	1
Execute time	0.003s
Fetch time	0.000s
Total time	0.003s
Start time	2024-10-28 16:22:56.856
Finish time	2024-10-28 16:22:59.991

*<admin> Console

class ×

属性 数据 ER 图

class 输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

	A-Z class_id	A-Z name	A-Z department
1	0001	01CSC	CS
2	0002	01CSC	CS

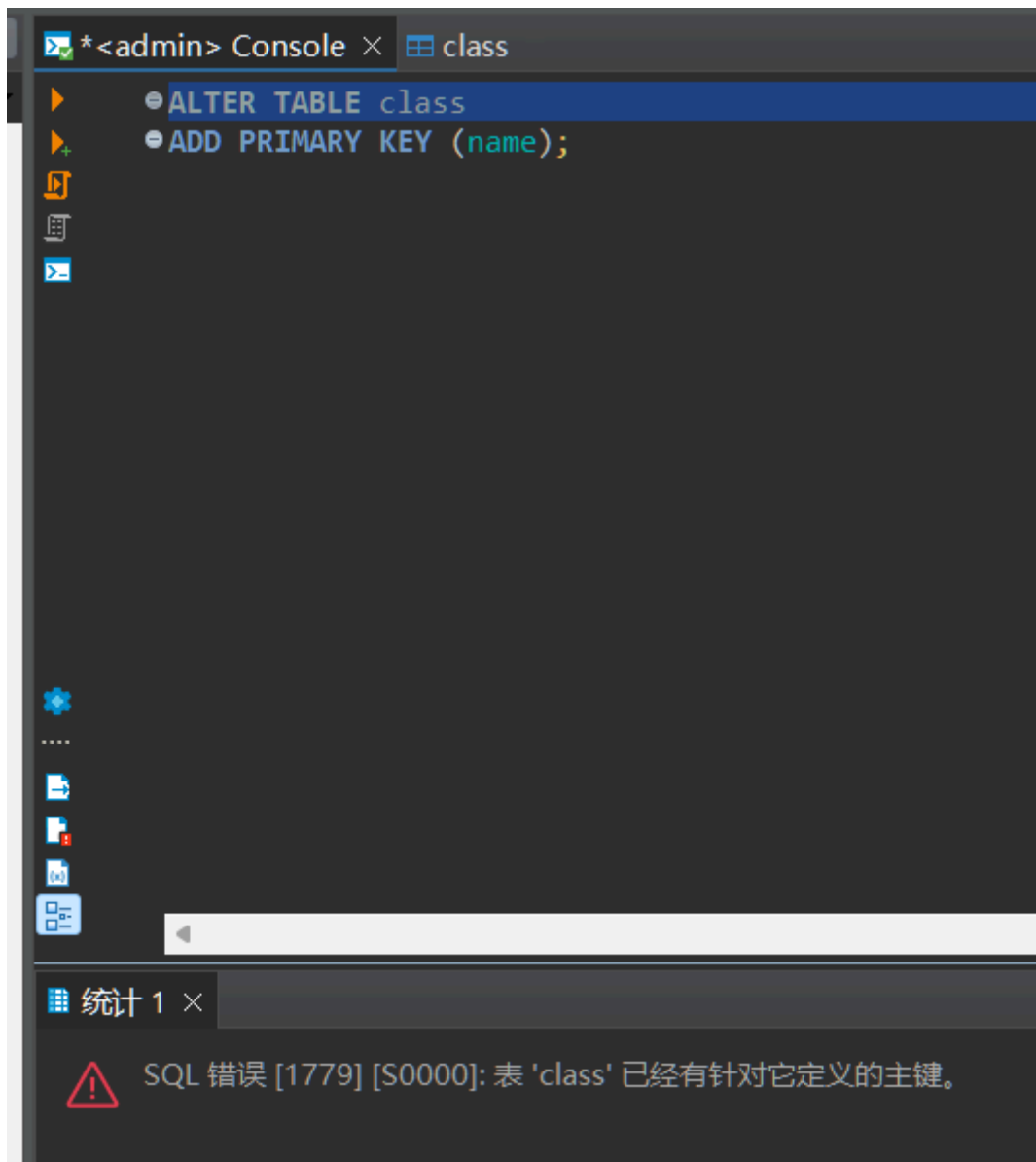
结果:

- 由于事务被回滚，所有插入操作都不会生效，表中记录数不变。

6

在完成上面几步的前提下，尝试设置name为主键，看能否成功，并思考原因。

```
-- 尝试将name设置为主键
ALTER TABLE class
ADD PRIMARY KEY (name);
```



结果：

- 提示SQL 错误 [1779] [S0000]: 表 'class' 已经有针对它定义的主键。

原因：每个表只能有一个主键。