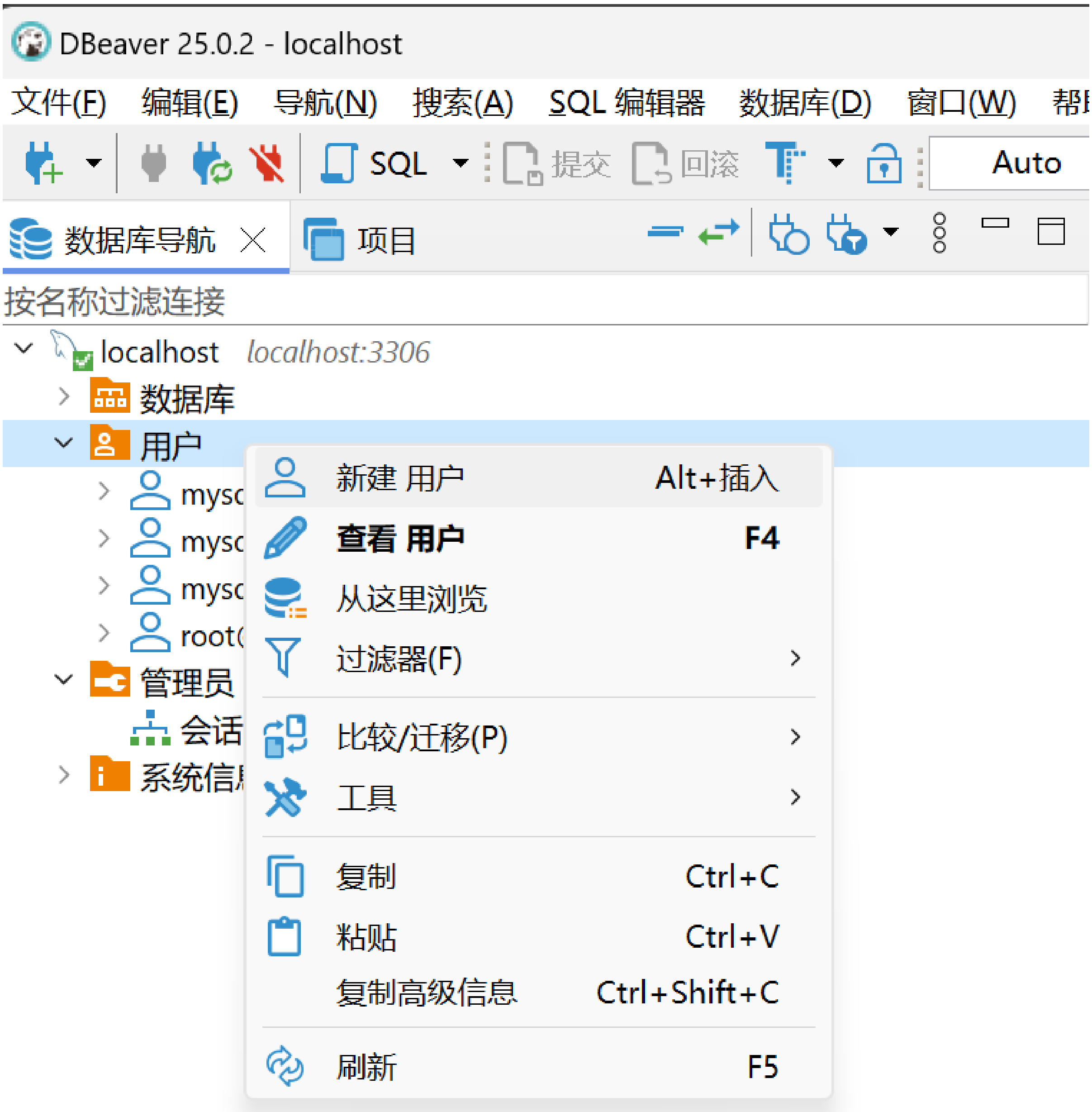
实验环境

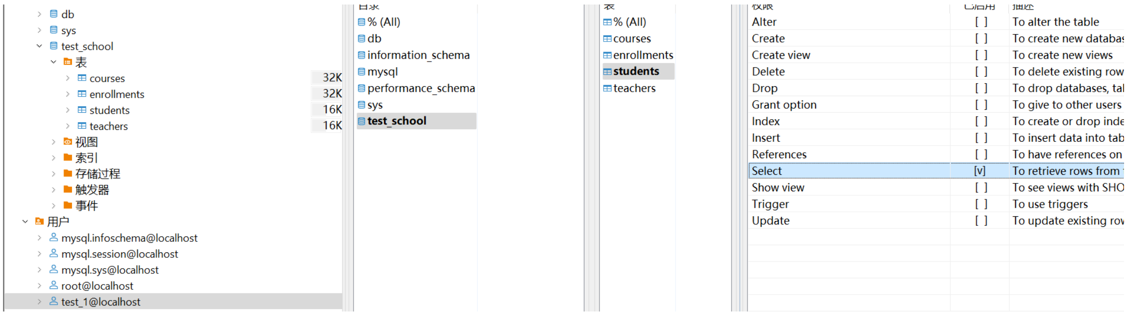
- Windows 11 系统
- MySQL 9.2
- DBeaver 25.0.2
- 交互式 SQL 所用数据库

实验内容与完成情况

- 授权与回收
    - 在数据库中由 DBA 创建若干用户 右键用户栏，新建用户



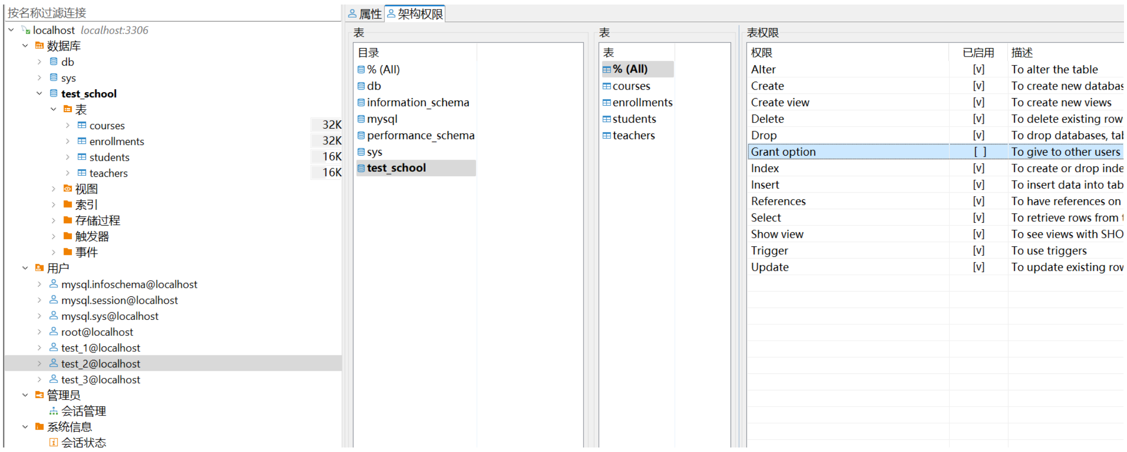
    - 对用户进行授权和回收
        - 将查询 Students 表的权限授予给用户 test_1 选中用户 test_1，授予其对 Students 表的 SELECT 权限



权限	已启用	描述
Alter	[ ]	To alter the table
Create	[ ]	To create new databas
Create view	[ ]	To create new views
Delete	[ ]	To delete existing row
Drop	[ ]	To drop databases, tal
Grant option	[ ]	To give to other users
Index	[ ]	To create or drop inde
Insert	[ ]	To insert data into tab
References	[ ]	To have references on
Select	[v]	To retrieve rows from
Show view	[ ]	To see views with SHO
Trigger	[ ]	To use triggers
Update	[ ]	To update existing rov

```
GRANT SELECT ON test_school.students TO `test_1`@`localhost`
```

- 将查询和修改所有表的权限授予给用户 test_2 和 test_3 依次选中用户 test_2 和 test_3，授予其对所有表的所有权限

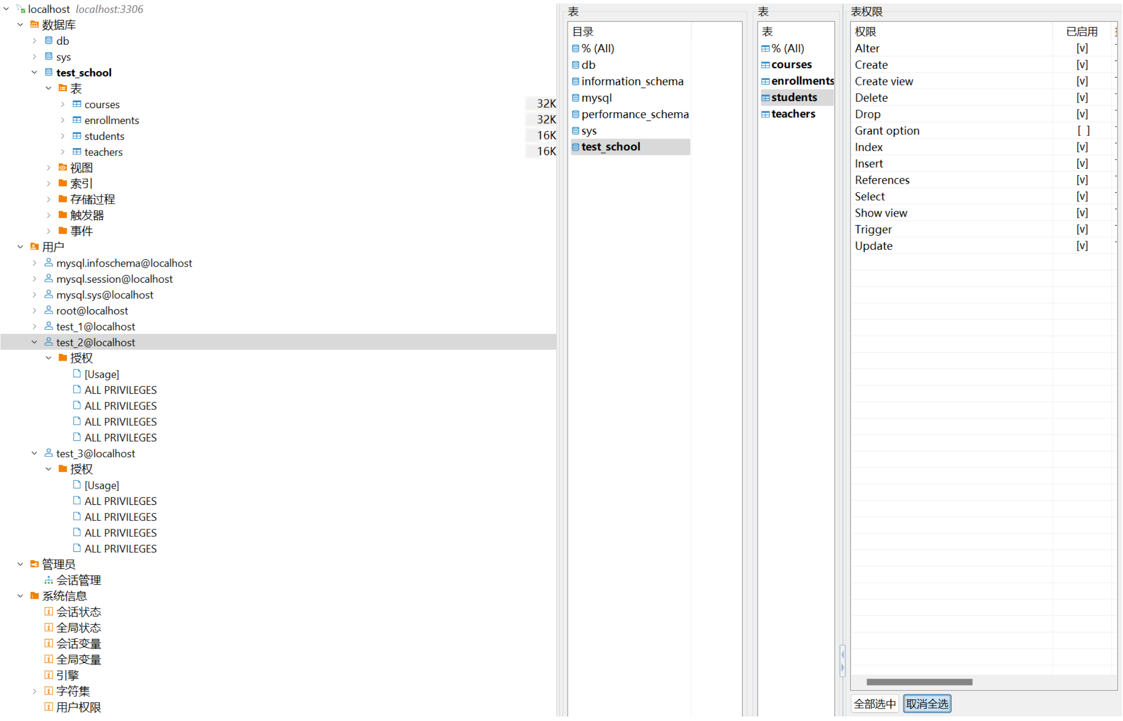


权限	已启用	描述
Alter	[v]	To alter the table
Create	[v]	To create new databas
Create view	[v]	To create new views
Delete	[v]	To delete existing row
Drop	[v]	To drop databases, tal
Grant option	[ ]	To give to other users
Index	[v]	To create or drop inde
Insert	[v]	To insert data into tab
References	[v]	To have references on
Select	[v]	To retrieve rows from t
Show view	[v]	To see views with SHO
Trigger	[v]	To use triggers
Update	[v]	To update existing rov

```
GRANT ALL PRIVILEGES ON test_school.courses TO
'test_2'@'localhost';
GRANT ALL PRIVILEGES ON test_school.enrollments TO
'test_2'@'localhost';
GRANT ALL PRIVILEGES ON test_school.students TO
'test_2'@'localhost';
GRANT ALL PRIVILEGES ON test_school.teachers TO
'test_2'@'localhost';

GRANT ALL PRIVILEGES ON test_school.courses TO
'test_3'@'localhost';
GRANT ALL PRIVILEGES ON test_school.enrollments TO
'test_3'@'localhost';
GRANT ALL PRIVILEGES ON test_school.students TO
'test_3'@'localhost';
GRANT ALL PRIVILEGES ON test_school.teachers TO
'test_3'@'localhost';
```

- 将用户 test_2 修改 Students 表的权限收回 选中用户 test_2，收回其对 Students 表的 Update 权限



localhost localhost:3306
数据库
db
sys
test_school
表
courses 32K
enrollments 32K
students 16K
teachers 16K
视图
索引
存储过程
触发器
事件
用户
mysql.infoschema@localhost
mysql.session@localhost
mysql.sys@localhost
root@localhost
test_1@localhost
test_2@localhost
授权
[Usage]
ALL PRIVILEGES
ALL PRIVILEGES
ALL PRIVILEGES
ALL PRIVILEGES
test_3@localhost
授权
[Usage]
ALL PRIVILEGES
ALL PRIVILEGES
ALL PRIVILEGES
ALL PRIVILEGES
管理员
会话管理
系统信息
会话状态
全局状态
会话变量
全局变量
引擎
字符集
用户权限

表

目录
% (All)
db
information_schema
mysql
performance_schema
sys
test_school

表

表
% (All)
courses
enrollments
students
teachers

表权限

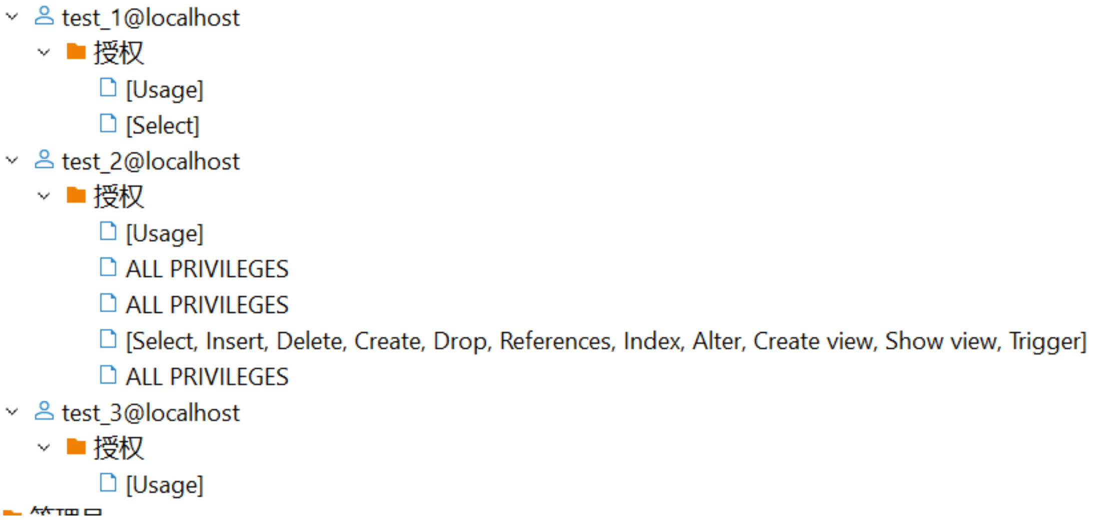
权限	已启用
Alter	[v]
Create	[v]
Create view	[v]
Delete	[v]
Drop	[v]
Grant option	[ ]
Index	[v]
Insert	[v]
References	[v]
Select	[v]
Show view	[v]
Trigger	[v]
Update	[v]

```sql
REVOKE UPDATE ON test_school.students FROM 'test_2'@'localhost';
```

- 将用户 test_3 所有权限收回

```sql
REVOKE ALL PRIVILEGES ON test_school.courses FROM
'test_3'@'localhost';
REVOKE ALL PRIVILEGES ON test_school.enrollments FROM
'test_3'@'localhost';
REVOKE ALL PRIVILEGES ON test_school.students FROM
'test_3'@'localhost';
REVOKE ALL PRIVILEGES ON test_school.teachers FROM
'test_3'@'localhost';
```

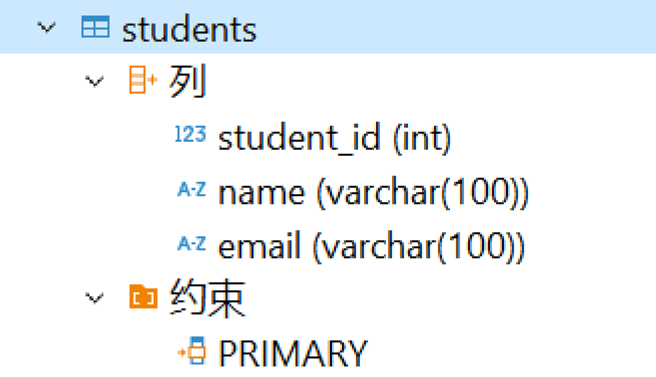
- 最终效果



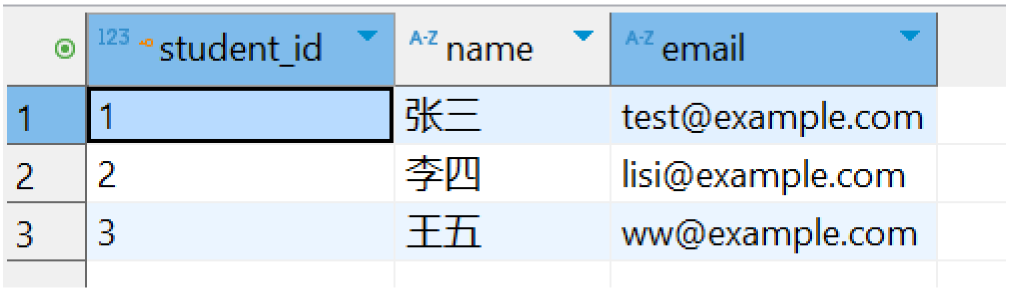
test_1@localhost
授权
[Usage]
[Select]
test_2@localhost
授权
[Usage]
ALL PRIVILEGES
ALL PRIVILEGES
[Select, Insert, Delete, Create, Drop, References, Index, Alter, Create view, Show view, Trigger]
ALL PRIVILEGES
test_3@localhost
授权
[Usage]
管理员

- 完整性控制

- 实体完整性
    1. Students 表的主码为 student_id，该表中存在 3 行数据（表的创建已在交互式 SQL 实验中完成）



```
students
  列
    student_id (int)
    name (varchar(100))
    email (varchar(100))
  约束
    PRIMARY
```

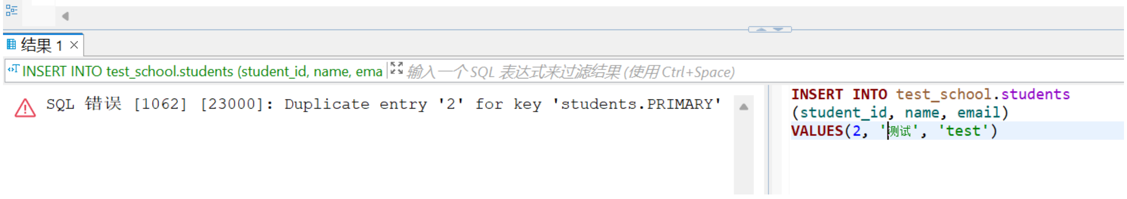


	student_id	name	email
1	1	张三	test@example.com
2	2	李四	lisi@example.com
3	3	王五	ww@example.com

    2. 插入(2, '测试', 'test')

```
INSERT INTO test_school.students
(student_id, name, email)
VALUES(2, '测试', 'test')
```

    3. MySQL 报错

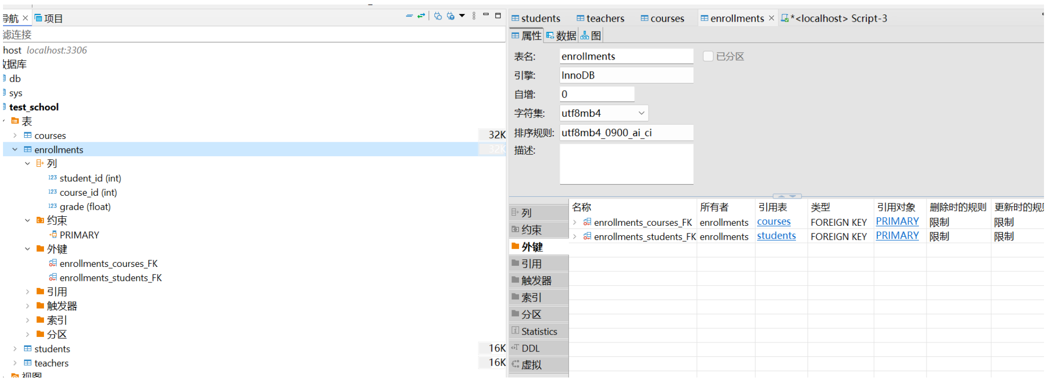


结果 1
INSERT INTO test_school.students (student_id, name, ema
输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

```
SQL 错误 [1062] [23000]: Duplicate entry '2' for key 'students.PRIMARY'
```

```
INSERT INTO test_school.students
(student_id, name, email)
VALUES(2, '测试', 'test')
```

- 参照完整性
    - 在 Enrollments（参照）表插入元组
        1. Enrollments 表参照了 Students 和 Courses 表，该表中存在 2 行数据（表的创建已在交互式 SQL 实验中完成）



host localhost:3306

db

sys

test_school

表

courses 32K

enrollments

列

student_id (int)

course_id (int)

grade (float)

约束

PRIMARY

外键

enrollments_courses_FK

enrollments_students_FK

引用

触发器

索引

分区

students 16K

teachers 16K

表名: enrollments

引擎: InnoDB

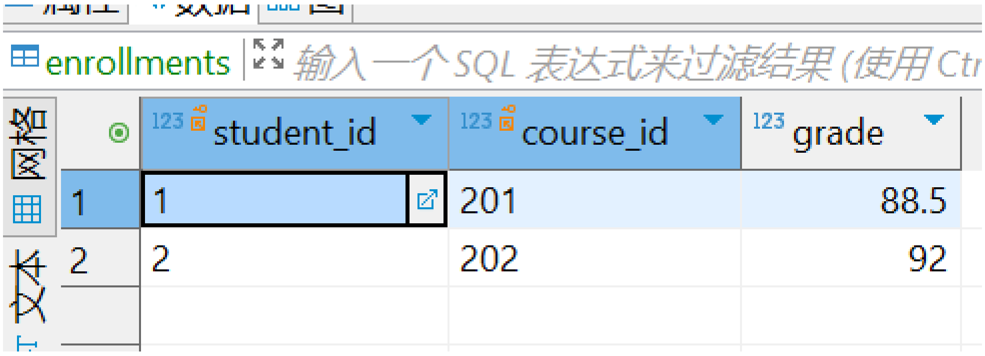
自增: 0

字符集: utf8mb4

排序规则: utf8mb4_0900_ai_ci

描述:

名称	所有者	引用表	类型	引用对象	删除时的规则	更新时的规
enrollments_courses_FK	enrollments	courses	FOREIGN KEY	PRIMARY	限制	限制
enrollments_students_FK	enrollments	students	FOREIGN KEY	PRIMARY	限制	限制



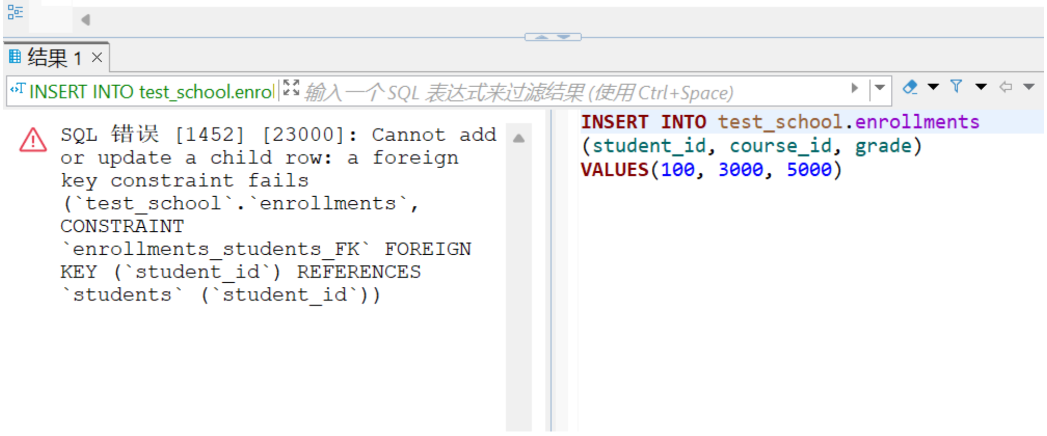
enrollments

	student_id	course_id	grade
1	1	201	88.5
2	2	202	92

2. 在 Enrollments 表中插入(100, 3000, 5000)

```
INSERT INTO test_school.enrollments
(student_id, course_id, grade)
VALUES(100, 3000, 5000);
```

3. MySQL 报错



结果 1
INSERT INTO test_school.enrol
输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

```
SQL 错误 [1452] [23000]: Cannot add
or update a child row: a foreign
key constraint fails
(`test_school`.`enrollments`,
CONSTRAINT
`enrollments_students_FK` FOREIGN
KEY (`student_id`) REFERENCES
`students` (`student_id`))
```

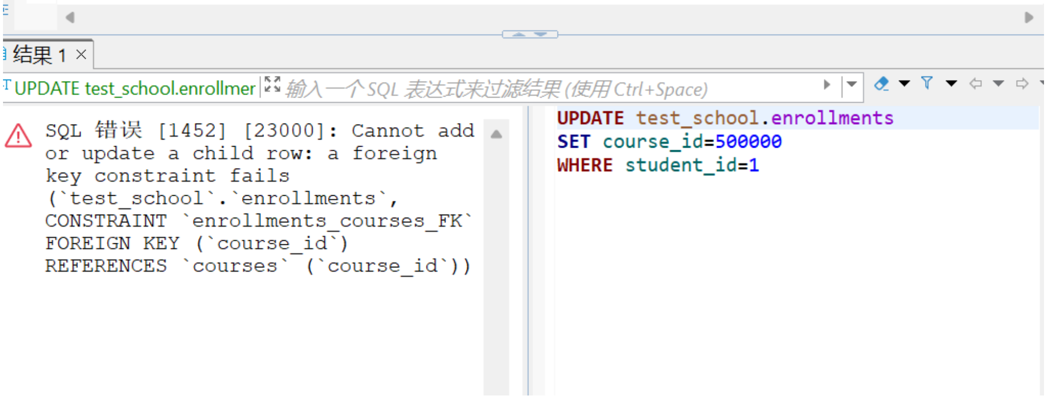
```
INSERT INTO test_school.enrollments
(student_id, course_id, grade)
VALUES(100, 3000, 5000)
```

- 在 Enrollments（参照）表修改外码值

1. 在 Enrollments 表中修改第 1 行数据

```
UPDATE test_school.enrollments
SET course_id=500000
WHERE student_id=1;
```

2. MySQL 报错


结果 1
UPDATE test_school.enrollmer
输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

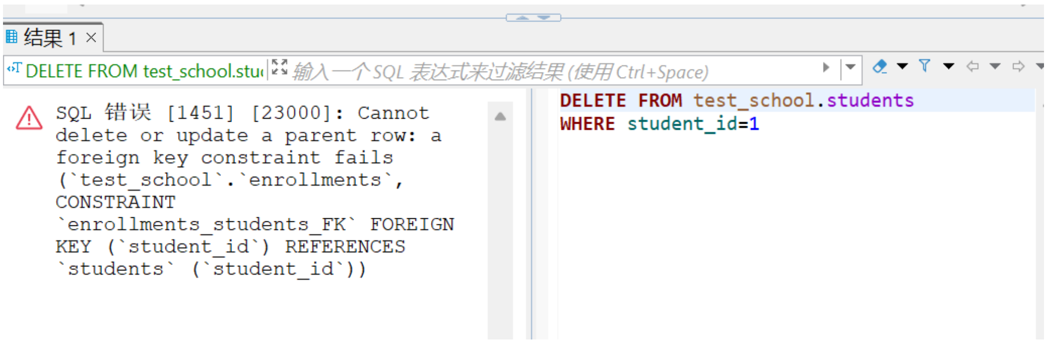
```
SQL 错误 [1452] [23000]: Cannot add
or update a child row: a foreign
key constraint fails
(`test_school`.`enrollments`,
CONSTRAINT `enrollments_courses_FK`
FOREIGN KEY (`course_id`)
REFERENCES `courses` (`course_id`))
```

```
UPDATE test_school.enrollments
SET course_id=500000
WHERE student_id=1
```

- 在 Students（被参照）表删除元组
  1. 在 Students 表中删除被参照的第 1 行数据

```
DELETE FROM test_school.students
WHERE student_id=1;
```

2. MySQL 报错


结果 1
DELETE FROM test_school.stu
输入一个 SQL 表达式来过滤结果 (使用 Ctrl+Space)

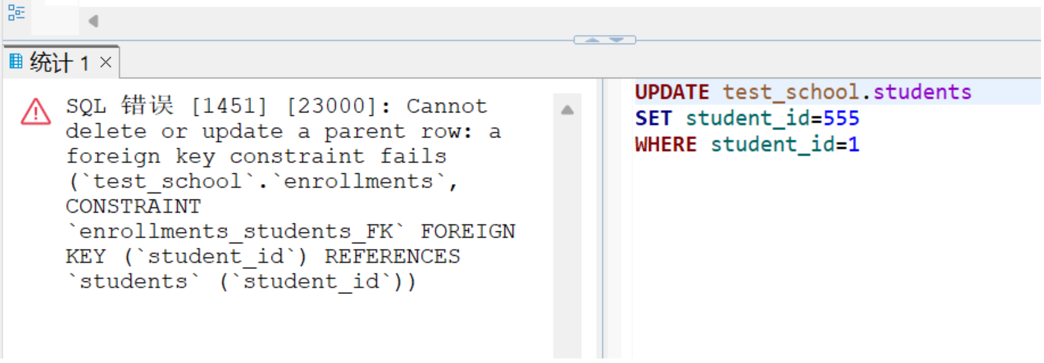
```
SQL 错误 [1451] [23000]: Cannot
delete or update a parent row: a
foreign key constraint fails
(`test_school`.`enrollments`,
CONSTRAINT
`enrollments_students_FK` FOREIGN
KEY (`student_id`) REFERENCES
`students` (`student_id`))
```

```
DELETE FROM test_school.students
WHERE student_id=1
```

- 在 Students（被参照）表修改主码值
  1. 在 Students 表中修改被参照的第 1 行数据

```
UPDATE test_school.students
SET student_id=555
WHERE student_id=1;
```

2. MySQL 报错



```
SQL 错误 [1451] [23000]: Cannot
delete or update a parent row: a
foreign key constraint fails
(`test_school`.`enrollments`,
CONSTRAINT
`enrollments_students_FK` FOREIGN
KEY (`student_id`) REFERENCES
`students` (`student_id`))
```

```
UPDATE test_school.students
SET student_id=555
WHERE student_id=1
```

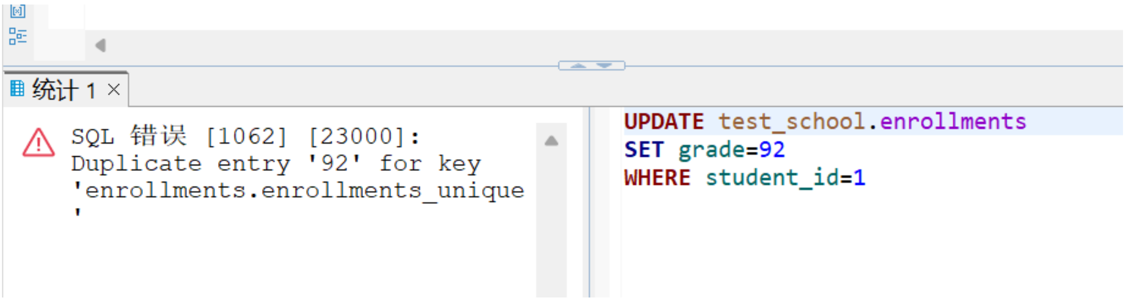
- 用户定义完整性
  1. 设置 Enrollments 表的 grade 列的约束为 UNIQUE

```
ALTER TABLE test_school.enrollments ADD CONSTRAINT enrollments_unique
UNIQUE KEY (grade);
```

2. 更改 Enrollments 表的第 1 行数据的 grade 列为 92 使其与第 2 行数据的 grade 列相同

```
UPDATE test_school.enrollments
SET grade=92
WHERE student_id=1;
```

3. MySQL 报错



```
SQL 错误 [1062] [23000]:
Duplicate entry '92' for key
'enrollments.enrollments_unique
'
```

```
UPDATE test_school.enrollments
SET grade=92
WHERE student_id=1
```

- CHECK 短语

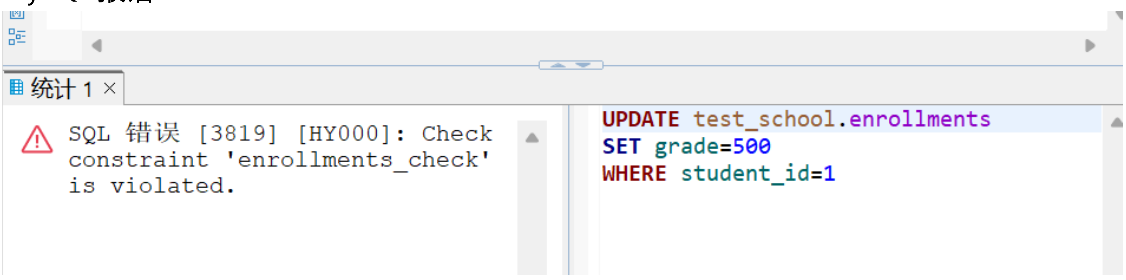
1. 设置 Enrollments 表的 grade 列的约束为 0<=grade<=100

```
ALTER TABLE test_school.enrollments ADD CONSTRAINT enrollments_check
CHECK (grade <= 100 and grade >= 0);
```

2. 更改 Enrollments 表的第 1 行数据的 grade 列为 5000

```
UPDATE test_school.enrollments
SET grade=500
WHERE student_id=1;
```

3. MySQL 报错



```
SQL 错误 [3819] [HY000]: Check
constraint 'enrollments_check'
is violated.
```

```
UPDATE test_school.enrollments
SET grade=500
WHERE student_id=1
```

- CONSTRAINT 子句

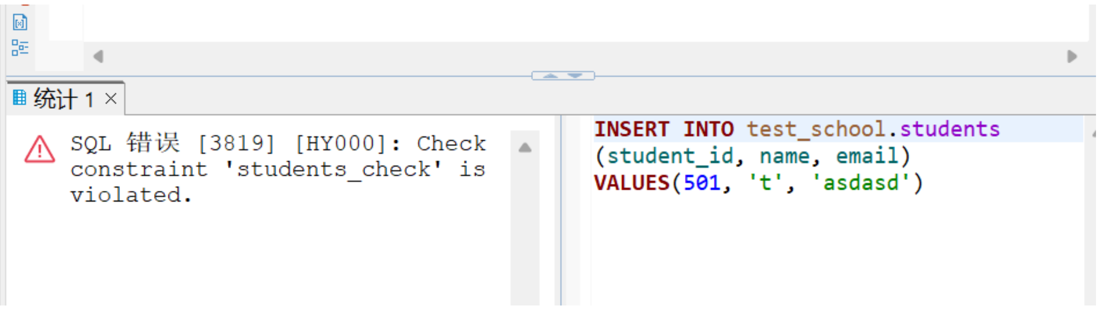
1. 设置 Students 表的约束为学号小于等于 500 并且名字长度大于等于 2

```
ALTER TABLE test_school.students ADD CONSTRAINT students_check CHECK
(student_id <= 500 AND LENGTH(name) >= 2);
```

2. 插入(501, 't', 'asdasd')

```
INSERT INTO test_school.students
(student_id, name, email)
VALUES(501, 't', 'asdasd');
```

3. MySQL 报错



```
SQL 错误 [3819] [HY000]: Check constraint 'students_check' is violated.
```

```
INSERT INTO test_school.students
(student_id, name, email)
VALUES(501, 't', 'asdasd')
```

出现的问题及解决方案

- 无