

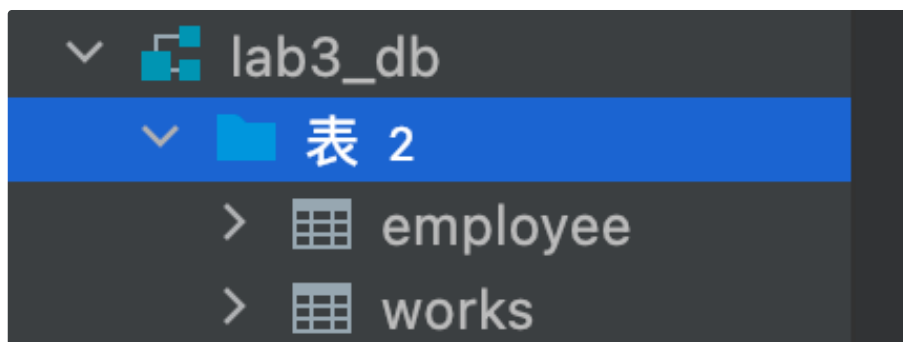
数据库系统 实验3

3200105872 庄毅非

实验过程

1. 定义若干表, 包含primary key, foreign key, check的定义

```
1 use lab3_db;
2
3 create table employee (
4     uid char(10),
5     age int not null,
6     sex char(1),
7     name varchar(20),
8     street varchar(20),
9     city varchar(20),
10    primary key(uid),
11    check(sex in ('M', 'F')),
12    check(age > 0)
13 );
14
15
16 create table works(
17     id int,
18     employee_id char(10),
19     work_id char(20),
20     salary numeric(12,2),
21     primary key(id),
22     foreign key(employee_id) references employee(uid),
23     check(salary >= 0)
24 );
```



2. 想表中插入数据，考察primary key 如何控制实体完整性

```
1 #首先插入四条数据
2 insert into employee values('1234567890',1,'M','test-
  1','yuhangtang','hangzhou');
3 insert into employee values('1234567450',1,'F','test-
  2','yuhangtang','hangzhou');
4 insert into employee values('1234564390',1,'M','test-
  3','yuhangtang','hangzhou');
5 insert into employee values('1234561290',1,'F','test-
  4','yuhangtang','hangzhou');
```

此后，插入和第一个元组主键一致的新元组

```
1 insert into employee values('1234567890',1,'F','test-
  5','yuhangtang','hangzhou');
```

mysql报错: [23000][1062] Duplicate entry '1234567890' for key
'employee.PRIMARY'

3. 删除被引用表中的行，考察foreign key中on delete 子句如何控制参照完整性

首先先在works表中插入和employee表中第一条记录对应的新元组。

```
1 insert into works(id, employee_id, work_id, salary)
  values(1,'1234567890','123',12.2);
```

之后尝试删除employee表中的第一个元组

```
1 delete from employee where uid = '1234567890';
```

在关闭safe mode的条件下，抛出错误：

```
1 Cannot delete or update a parent row: a foreign key constraint
  fails (`lab3_db`.`works`, CONSTRAINT `works_ibfk_1` FOREIGN KEY
    (`employee_id`) REFERENCES `employee` (`uid`))
```

```
[2022-03-24 14:25:50] [23000][1451] Cannot delete or update a parent row: a foreign key constraint fails (`lab3_db`.`works`, CONSTRAINT `works_ibfk_1` FOREIGN KEY (`employee_id`) REFERENCES `emp
[2022-03-24 14:25:50] [23000][1451] Cannot delete or update a parent row: a foreign key constraint fails (`lab3_db`.`works`, CONSTRAINT `works_ibfk_1` FOREIGN KEY (`employee_id`) REFERENCES `emp
```

4. 修改被引用表中的行的primary key，考察foreign key 中on update 子句如何控制参照完整性。

```
1 update employee
2 set uid = '1231231231'
3 where name = 'test-1';
```

结果发生了一样的错误：

```
1 Cannot delete or update a parent row: a foreign key constraint
  fails (`lab3_db`.`works`, CONSTRAINT `works_ibfk_1` FOREIGN KEY
    (`employee_id`) REFERENCES `employee` (`uid`))
```

```
[2022-03-24 14:28:18] [23000][1451] Cannot delete or update a parent row: a fo
[2022-03-24 14:28:18] [23000][1451] Cannot delete or update a parent row: a fo
```

5. 修改或插入表中数据，考察check子句如何控制校验完整性

执行 `insert into employee values('1234asd90',-1,'L','test-5','yuhangtang','hangzhou');`

mysql报错: `Check constraint 'employee_chk_1' is violated.`

```
Check constraint 'employee_chk_1' is violated.
Check constraint 'employee_chk_1' is violated.
```

6. 定义一个assertion，并通过修改表中数据考察断言如何控制数据完整性。

```
1 create assertion assertion_age check
2 (not exists (select * from employee where city = 'New York'))
```

不过mysql 8.0似乎不支持断言，想要实现类似目的的话，可以使用触发器。

7. 定义一个trigger，并通过修改表中数据考察触发器如何起作用。

定义的trigger，每次向employee表中插入一个员工，works表中所有员工的薪水改为原来的1.1倍

```
1 delimiter //
2 create trigger employee_trigger
3     after insert on employee
4     for each row
5     begin
6         update works
7             set salary = salary * 1.1;
8     end //
9
10 delimiter ;
```

插入数据之前的works表：

	id	employee_id	work_id	salary
1	1	1234567890	123	12.20

现在向employee表中插入新的数据

```
1 insert into employee values('12345asd0',2,'F','test-
7','yuhangtang','hangzhou');
```

可以看到两个表都发生变化：

works表：

	id	employee_id	work_id	salary
1	1	1234567890	123	13.42

employee表:

	uid	age	sex	name	street	city
1	1234561290	1	F	test-4	yuhangtang	hangzhou
2	1234564390	1	M	test-3	yuhangtang	hangzhou
3	1234567450	1	F	test-2	yuhangtang	hangzhou
4	1234567890	1	M	test-1	yuhangtang	hangzhou
5	12345asd0	2	F	test-7	yuhangtang	hangzhou