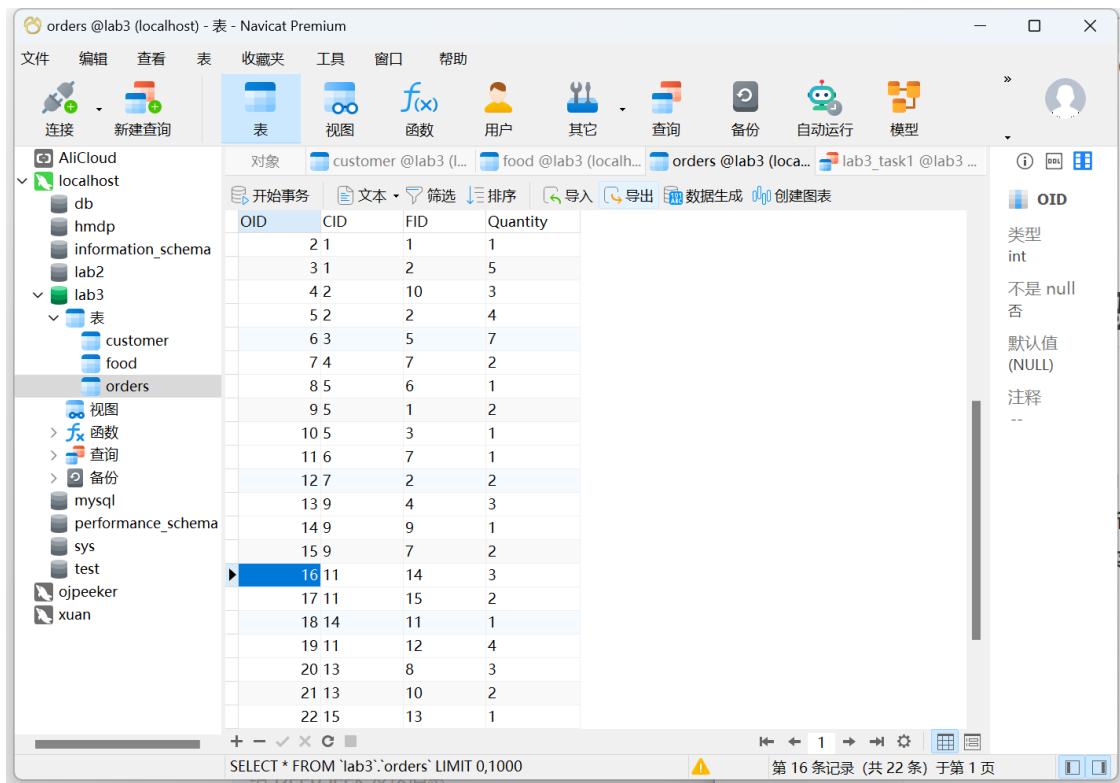


# 第三次实验

## Task1

- 导入表



- Q1: 查询与CID=1的顾客同一个城市的所有顾客ID



```
# Q1
select CID
from customer
where city = (
    select city
    from customer
    where CID = 1
);
```

信息	摘要	结果 1	剖析	状态
		CID		
		1		
		3		

- Q2: 查询购买过所有省份（Food表中出现过的City）的食物的顾客ID

```
select orders.CID
from orders INNER JOIN food ON orders.FID = food.FID
group by orders.CID
HAVING COUNT(DISTINCT food.City) = (
select COUNT(DISTINCT City)
from food);
```



信息	摘要	结果 1	剖析	状态
		CID		
		▶(N/A)		

+ - ✓ ✕  

- Q3: 查询至少购买过ID为4的顾客买过的全部食物的顾客ID

```
SELECT distinct o.CID
FROM Orders o
WHERE NOT EXISTS (
    SELECT FID FROM Orders WHERE CID = 4
    EXCEPT
    SELECT FID FROM Orders WHERE CID = o.CID
);
```

信息	摘要	结果 1	剖析	状态
		CID		
		▶4		
		6		
		9		

+ - ✓ ✕  

## Task2

- Q1: 创建一个新表Sales，字段为：Food ID（主键） 数字型，食物名（非空）字符型 长度20，总销量 数字型。查询每种食物的总销量，将结果插入表中。（两条SQL语句，一条为create语句，一条为insert语句）。

```
# Q1
create table sales (
    FoodID INT PRIMARY KEY,
    FoodName VARCHAR(20) NOT NULL,
    count INT
);

INSERT INTO sales(FoodID, FoodName, count)
SELECT f.FID, f.`Name`, SUM(o.Quantity)
FROM food as f INNER JOIN orders as o
ON f.FID = o.FID
GROUP BY f.FID, f.Name;
```

FoodID	FoodName	count
1	重庆小面	3
2	酸辣粉	11
3	火锅	1
4	夫妻肺片	3
5	串串香	9
6	白切鸡	1
7	烧鹅	5
8	肉夹馍	3
9	过桥米线	1
10	沙县小吃	5
11	北京烤鸭	1
12	魔匠药水	4
13	巨匠药水	1
14	柠檬水	3
15	咖啡曲奇	2

- Q2: 向Order表添加一条交易记录，内容自定义，并更新对应食物的总销量字段（两条SQL语句）

```
# Q2
INSERT INTO orders(OID, CID, FID, Quantity)
VALUES (1, 1, 1, 1000);

UPDATE sales
SET count = count + 1000
WHERE FoodID = 1;
```

FoodID	FoodName	count
1	重庆小面	1003
2	酸辣粉	11
3	火锅	1
4	夫妻肺片	3
5	串串香	9
6	白切鸡	1
7	烧鹅	5
8	肉夹馍	3
9	过桥米线	1
10	沙县小吃	5
11	北京烤鸭	1
12	魔匠药水	4
13	巨匠药水	1
14	柠檬水	3
15	咖啡曲奇	2

- Q3: 为新表添加一个评分字段（数字型），要求分数范围限定为0-10，并设置默认值6

```
# Q3
ALTER TABLE sales
ADD pf INT default 6
check(pf BETWEEN 0 AND 10);
```

FoodID	FoodName	count	pf
1	重庆小面	1003	6
2	酸辣粉	11	6
3	火锅	1	6
4	夫妻肺片	3	6
5	串串香	9	6
6	白切鸡	1	6
7	烧鹅	5	6
8	肉夹馍	3	6
9	过桥米线	1	6
10	沙县小吃	5	6
11	北京烤鸭	1	6
12	魔匠药水	4	6
13	巨匠药水	1	6
14	柠檬水	3	6
15	咖啡曲奇	2	6

## Task3

- Q1: 建立购买过重庆或四川食物的顾客视图Shu-view（包含Customer中CID, City）

```
select distinct `c`.`CID` AS `CID`,`c`.`City` AS `City` from
((`customer` `c` join `orders` `o` on((`c`.`CID` =
`o`.`CID`))) join `food` `f` on((`o`.`FID` = `f`.`FID`)))
where (`f`.`City` in ('重庆','四川'))
```

	CID	City
	1	北京
	2	重庆
	3	北京
	5	湖南
▶	7	广东
	9	甘肃

- Q2: 查询购买过重庆或四川食物的顾客中订单总消费最高的顾客CID (使用视图Shu-view, 思考使用视图的好处)

```
SELECT s.CID
FROM Shu_view s
JOIN orders o ON s.CID = o.CID
JOIN food f ON o.FID = f.FID
GROUP BY s.CID
ORDER BY SUM(o.Quantity * f.Price) DESC
LIMIT 1;
```

	CID
▶	1

**视图好处:** 可以简化复杂逻辑, 提高可读性和可维护性

- Q3: 向视图Shu-view加入表项 (16, 湖南), 能成功吗, 为什么?

答: 不能成功, 因为原视图存在 JOIN 等复杂连接修饰词限制了DML进行视图修改

- Q4: 建立男性顾客的视图Male-view (包含Customer中CID, City), 并要求对该视图进行的更新操作 只涉及男性顾客。 (WITH CHECK OPTION)

```
select `customer`.`CID` AS `CID`,`customer`.`City` AS `City`
from `customer` where (`customer`.`Gender` = '男')
```

	CID	City
	1	北京
	2	重庆
	6	广东
	7	广东
	11	游末邦
	12	水晶都
	14	游末邦
▶	15	水晶都

- Q5: 向视图Male-view加入表项 (17, 湖南) , 能成功吗, 为什么?

答: 不能成功, 因为存在 check option, 而未指定 Gender 无法插入不满足完整性的数据