

数据库第四次上机

22373386 高铭

TASK 1

CID	Gender	City	FID	Name	City	Price
1	男	北京	1	重庆小面	重庆	10
2	男	重庆	2	酸辣粉	重庆	7
3	女	北京	3	火锅	重庆	80
4	女	广东	4	夫妻肺片	四川	12
5	女	湖南	5	串串香	四川	35
6	男	广东	6	白切鸡	广东	20
7	男	广东	7	烧鹅	广东	30
8	女	江西	8	肉夹馍	陕西	5
9	女	甘肃	9	过桥米线	云南	10
10	女	吉林	10	沙县小吃	福建	20
11	男	游末邦	11	北京烤鸭	北京	100
12	男	水晶都	12	魔匠药水	游末邦	12
13	女	水晶都	13	巨匠药水	水晶都	85
14	男	游末邦	14	柠檬水	游末邦	18
15	男	水晶都	15	咖啡曲奇	水晶都	20

OID	CID	FID	Quantity
1	1	5	2
2	2	1	1
3	3	1	5
4	4	2	3
5	5	2	4
6	6	3	7
7	7	4	2
8	8	5	1
9	9	5	2
10	10	5	1
11	11	6	1
12	12	7	2
13	13	9	3
14	14	9	1
15	15	9	2
16	16	11	3
17	17	11	2
18	18	14	1
19	19	11	4
20	20	13	3
21	21	13	2
22	22	15	1

Q1: 查询与CID=1的顾客同一个城市的所有顾客ID

```
1  select CID
2  from customer c1
3  where c1.City = (
4      select City
5      from customer c2
6      where c2.CID =1
7  );
```

CID	
1	1
2	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

```

1  select CID, count(City)
2  from food
3  join orders on food.FID = orders.FID
4  group by CID
5  having count(city) = (
6      select count(distinct City)
7      from food
8  )
9  # 就给定的表中数据而言，没有符合要求的元组

```

CID	count(City)
-----	-------------

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

```

1  select o1.CID
2  from orders o1
3  where o1.FID in(
4      select FID
5      from orders o2
6      where o2.CID = 4
7  ) and o1.CID <> 4

```

CID	
1	6
2	9

TASK 2

Q1: 创建一个新表Sales，查询每种食物的总销量，将结果插入表中

```

1  create table Sales(
2      FID int primary key,
3      Name varchar(20),
4      Quantity int
5  );

```

```

6
7  insert into Sales
8      select food.FID, Name, sum(Quantity)
9      from orders, food
10     where orders.FID = food.FID
11     group by food.FID, Name

```

console_17

sales

customer

<

<

15 行

>

>

↺

🕒

■

+

−

↶

🔍

WHERE

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

```

1  insert into orders
2  values (23, 7, 12, 114514);
3
4  update sales
5  set Quantity = Quantity + 114514
6  where FID = 12

```

console_17

orders

sales

23 行

刷新

🕒

+

-

sales

Tx: 自动

WHERE

	OID	CID	FID	Quantity
8	8	5	6	1
9	9	5	1	2
10	10	5	3	1
11	11	6	7	1
12	12	7	2	2
13	13	9	4	3
14	14	9	9	1
15	15	9	7	2
16	16	11	14	3
17	17	11	15	2
18	18	14	11	1
19	19	11	12	4
20	20	13	8	3
21	21	13	10	2
22	22	15	13	1
23	23	7	12	114514

console_17

orders

sales

15 行

刷新

🕒

+

-

↩

🔄

WHERE

	FID	Name	Quantity
1	1	重庆小面	3
2	2	酸辣粉	11
3	3	火锅	1
4	4	夫妻肺片	3
5	5	串串香	9
6	6	白切鸡	1
7	7	烧鹅	5
8	8	肉夹馍	3
9	9	过桥米线	1
10	10	沙县小吃	5
11	11	北京烤鸭	1
12	12	魔匠药水	114518
13	13	巨匠药水	1
14	14	柠檬水	3
15	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

```
1  alter table sales
2  add Score float default 6 check (Score >= 0 and Score <= 10)
```

	FID	Name	Quantity	Score
1	1	重庆小面	3	6
2	2	酸辣粉	11	6
3	3	火锅	1	6
4	4	夫妻肺片	3	6
5	5	串串香	9	6
6	6	白切鸡	1	6
7	7	烧鹅	5	6
8	8	肉夹馍	3	6
9	9	过桥米线	1	6
10	10	沙县小吃	5	6
11	11	北京烤鸭	1	6
12	12	魔匠药水	114518	6
13	13	巨匠药水	1	6
14	14	柠檬水	3	6
15	15	咖啡曲奇	2	6

Task 3

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）

```
1  create view Shu_view as
2  select c.cid, c.City
3  from customer c, food f, orders o
4  where c.CID = o.CID and f.FID = o.FID
5  and (f.City = '重庆' or f.City = '四川')
6  group by o.CID
```

console_17

shu_view

6 行

WHERE

	cid	City
1	1	北京
2	5	湖南
3	2	重庆
4	7	广东
5	9	甘肃
6	3	北京

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）

```

1  # 方法1: 避免子查询, 使用排序->LIMIT查找最大值
2  select shu_view.cid, SUM(Price * Quantity)
3  from shu_view, orders, food
4  where shu_view.cid = orders.CID and food.FID and orders.FID
5  group by orders.CID
6  ORDER BY SUM(Price * Quantity) DESC
7  LIMIT 1;
8
9  # 方法2: 用子查询, 不用LIMIT语句, 与方法一等价
10 select shu_view.cid
11 from shu_view, orders, food
12 where shu_view.cid = orders.CID and food.FID and orders.FID
13 group by orders.CID
14 having sum(Price * Quantity) >= all(
15     select SUM(Price * Quantity)
16     from shu_view, orders, food
17     where shu_view.cid = orders.CID and food.FID and orders.FID
18     group by orders.CID
19 )

```

cid
7

- 使用视图，好处在于可以将复杂的查询逻辑封装起来，使得查询变得简单和可重用，简化用户操作。
- 在这种情况下，我们可以利用已经创建的 `Shu_view` 视图来执行更简洁的查询，避免了多重子查询，提高可读性。

Q3：向视图Shu-view加入表项（16，湖南），能成功吗，为什么？

- 不能成功，会产生报错：The target table shu_view of the INSERT is not insertable-into。
- 因为只有可更新视图才能向视图加入表项，可更新视图是指：在视图中的行和基表中的行之间必须具有一对一的关系
- 如果视图包含下述结构中的任何一种，那么它就是不可更新的：
 1. SELECT中含有聚合函数；
 2. DISTINCT关键字；
 3. JOIN、GROUP BY、ORDER BY、HAVING子句；
 4. UNION运算符；
 5. 位于选择列表中的子查询；
 6. FROM子句中包含多个表；
 7. SELECT语句中引用了不可更新视图；
 8. WHERE子句中的子查询，引用FROM子句中的表

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）

```
1 create view Male_view as
2     select CID, City
3     from customer
4     where Gender = '男'
5     with check option
```

	CID	City
1	1	北京
2	2	重庆
3	6	广东
4	7	广东
5	11	游末邦
6	12	水晶都
7	14	游末邦
8	15	水晶都

Q5：向视图Male-view加入表项（17，湖南），能成功吗，为什么？

- 不能成功，会产生报错 CHECK OPTION failed 'lab4.male_view'。
- 因为建立 Male_view 表时有限制 CHECK OPTION，规定对视图进行增删改操作时会检查数据是否满足 Gender = '男' 这一条件。CID为17的记录并不一定是男性顾客（实际上表中是 null），因此操作不会成功。