

```
mysql -u root -p --local-infile=1
```

```
create database lab04;

use lab04

create table Customer (
    CID int primary key,
    Gender char(4),
    City varchar(20)
);

create table Food (
    FID int primary key,
    Name varchar(30),
    City varchar(20),
    Price dec(10, 2)
);

create table `Order` (
    OID int primary key,
    CID int,
    FID int,
    Quantity int,
    constraint fk_order_customer foreign key (CID)
        references Customer (CID)
        on update cascade,
    constraint fk_order_food foreign key (FID)
        references Food (FID)
        on update cascade
);

set global local_infile = 1;

load data
local
infile '/Users/hzy/Desktop/2021数据库/LAB04/04/data/Customer.csv'
into table Customer
fields terminated by ','
lines terminated by '\r\n'
ignore 1 lines;

load data
local
infile '/Users/hzy/Desktop/2021数据库/LAB04/04/data/Food.csv'
```

```

into table Food
fields terminated by ','
lines terminated by '\r\n'
ignore 1 lines;

load data
local
infile '/Users/hzy/Desktop/2021数据库/LAB04/04/data/Order.csv'
into table `Order`
fields terminated by ','
lines terminated by '\r\n'
ignore 1 lines;

```

## Task 1

查询与CID=1的顾客同一个城市的所有顾客ID

```

select CID
from Customer
where City = (
    select City
    from Customer
    where CID = 1
);

```

```

+-----+
| CID |
+-----+
| 1 |
| 3 |
+-----+
2 rows in set (0.00 sec)

```

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 CID
from Customer
where not exists (
    select *
    from (
        select distinct City
        from Food
    ) as FoodAllCity
    where not exists (
        select *
        from `Order`, Food
        where Customer.CID = `Order`.CID
    )
);

```

```

        and `Order`.FID = Food.FID
        and FoodAllCity.City = Food.City
    )
);

```

```

+-----+
| CID |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)

```

查询至少购买过ID为13的顾客买过的全部食物的顾客ID

```

select CID
from Customer
where not exists (
    select *
    from `Order` o1
    where CID = 13 and not exists (
        select *
        from `Order` o2
        where o2.CID = Customer.CID
            and o2.FID = o1.FID
    )
);

```

```

+-----+
| CID |
+-----+
| 1 |
| 6 |
| 13 |
+-----+
3 rows in set (0.01 sec)

```

## Task 2

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. 使用 `distinct`

```
create view `Shu-view`
as (
    select distinct Customer.CID, Customer.City
    from Customer, Food, `Order`
    where Customer.CID = `Order`.CID
        and Food.FID = `Order`.FID
        and (Food.City = '重庆' or Food.City = '四川')
);
```

## 2. 使用 `exists`

```
create view `Shu-view`
as (
    select CID, City
    from Customer
    where exists (
        select *
        from Food, `Order`
        where Customer.CID = `Order`.CID
            and Food.FID = `Order`.FID
            and (Food.City = '重庆' or Food.City = '四川')
    )
);
```

挑选出视图Shu-view中订单总消费最高的顾客CID：

```
select `Shu-view`.CID
from `Shu-view`, Food, `Order`
where `Shu-view`.CID = `Order`.CID
    and Food.FID = `Order`.FID
group by `Shu-view`.CID
having sum(Food.Price * `Order`.Quantity) >= all(
    select sum(Food.Price * `Order`.Quantity)
    from `Shu-view`, Food, `Order`
    where `Shu-view`.CID = `Order`.CID
        and Food.FID = `Order`.FID
    group by `Shu-view`.CID
);
```

```
+-----+
| CID   |
+-----+
|      1 |
+-----+
1 row in set (0.01 sec)
```

向视图Shu-view加入表项（16，湖南），能成功吗，为什么？

### 1. 使用 `distinct`

不能。因为使用 `distinct` 后的视图中的行不能与基本表中的行一一对应，所以这种视图不能进行插入、更新、删除操作。

### 2. 使用 `exists`

能。因为这种视图中的行能够与 `Customer` 中的行一一对应，当进行插入、更新、删除操作时，相当于对 `Customer` 表进行操作。

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，并考虑视图的可扩充性）

```
create view `Male-view`  
as (  
    select CID, City  
    from Customer  
    where Gender = '男'  
)  
with check option;
```

向视图Male-view加入表项（17，湖南），能成功吗，为什么？

不能。因为定义视图时指定了 `with check option`，而且插入的数据中没有 Gender 属性，无法确定是否满足约束，所以拒绝插入。