https://github.com/danielc0423/main

```
create table customer(
    cus_id int primary key,
    cus_name varchar(50) not null,
    cus_phone int not null,
    cus_email varchar(50) not null,
    payment text default 'credit card',
    balance money default 0.0);

insert into customer(cus_id,cus_name,cus_phone,cus_email,balance)
values(99,'Kitty',24689654,'hkuspacedata@email.com',50);

select * from customer;
```

this will give result

99|Kitty|24689654|hkuspacedata@email.com|credit card|50

```
create table hello(
    cus_id int primary key,
    cus_name varchar(50) not null,
    cus_phone int not null,
    cus_email varchar(50) not null,
    payment text default 'credit card',
    balance money default 0.0);

insert into customer(cus_id,cus_name,cus_phone,cus_email,payment,balance)
values(99,'Kitty',24689654,'hkuspacedata@email.com','gift card',50);

select * from customer;
```

this will give result

99|Kitty|24689654|hkuspacedata@email.com|gift card|50

```
create table product(
    prod_id varchar(30) primary key,
    prod_name varchar(50) not null,
    prod_price int not null,
    quantity int not null,
    brand char not null,
    category char not null,
    shop_id int not null);
```

```
insert into product(prod_id,prod_name,prod_price,quantity,brand,category,shop_id)
values('a1234560','ilone',10499,1,'orange','smartphone',699);

select * from product;
```

this will give

```
a1234560|ilone|10499|1|orange|smartphone|699
```

```
create table orders(
   order_id int primary key,
   order_date date not null,
   cus_id int not null,
   shop_id int not null,
   total_price money not null,
   total_unit int not null,
   weight int not null,
   delivary_fee money not null,
   txn_tax money not null);

insert into
orders(order_id,order_date,cus_id,shop_id,total_price,total_unit,weight,delivary_fee,t
xn_tax)

values(888,'2019-2-28',99,699,10499,1,0.5,23,0);

   select * from orders;
```

this will give

```
888|2019-2-28|99|699|10499|1|0.5|23|0
```

//start of sql

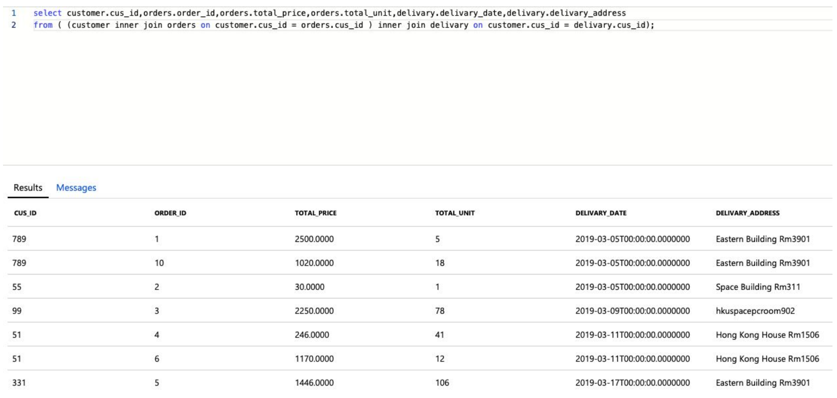
log 28-2-2019 6:54pm

```
create table customer(
   cus_id int primary key,
   cus_name varchar(50) not null,
   cus_phone int not null,
    cus_address varchar(50) not null,
    cus_email varchar(50) not null,
    payment text default 'credit card',
    balance money default (0.0);

insert into
customer(cus_id,cus_name,cus_phone,cus_address,cus_email,payment,balance)
values(99,'Kitty',24689654,'hkuspacepcroom902','hkuspacedata@email.com','gift
card',50);

insert into customer(cus_id,cus_name,cus_phone,cus_address,cus_email,balance)
values(66,'james',32324465,'hkuspacepcroom406','hkuspaceiae@email.com',0);

select * from customer;

create table product(
    prod_id varchar(30) primary key,
    prod_name varchar(50) not null,
    prod_price int not null,
    quantity int not null,
    brand char not null,
    category char not null,
    shop_id int not null);

insert into product(prod_id,prod_name,prod_price,quantity,brand,category,shop_id)
values('a1234560','ilone',10499,1,'orange','smartphone',699);

select * from product;

create table shop(
    seller_id int not null ,
    shop_id int not null,
    area_code int not null,
    seller_phone int not null,
    seller_bankac int );

insert into shop(seller_id,shop_id,area_code,seller_phone,seller_bankac)
values(5,699,'a1234560',555,34656597,1111555599993333);

select * from test;




create table orders(
    order_id int primary key reference on customer(cus_id),
    order_date date not null,
    cus_id int not null,
    shop_id int not null,
    total_price money not null,
    total_unit int not null,
    weight int not null,
    delivary_fee money not null,
    txn_tax money not null);

insert into
orders(order_id,order_date,cus_id,shop_id,total_price,total_unit,weight,delivary_fee,t
xn_tax)
values(888,'2019-2-28',99,699,10499,1,0.5,23,0);

select * from orders;


create table delivary(
    order_id int not null,
    cus_id int not null,
    order_date date not null,
    delivary_date date,
    delivary_address not null,
    estimated_time varchar(50),
    status text default 'On delivary',
    foreign key (order_id) references orders (order_id));

insert into
delivary(order_id,cus_id,order_date,delivary_date,delivary_address,estimated_time)
values(888,99,'2019-2-28','2019-3-15','hkuspacepcroom902','15 days');
select * from delivary;



create view [test] as select
customer.cus_id,orders.order_id,orders.total_price,orders.total_unit,delivary.delivary
_date,delivary.delivary_address from customer,orders,delivary;
select * from test;
```

above is intented to create a delivary record for customer / delivary staff
however it creates data redundancy

corrected it into

function 1(for staff and customer):
select customer.cus_id,orders.order_id,orders.total_price,orders.total_unit,delivary.delivary_date,delivary.delivary_address from ((customer inner join orders on customer.cus_id = orders.cus_id) inner join delivary on customer.cus_id = delivary.cus_id);



```
select customer.cus_id,orders.order_id,orders.total_price,orders.total_unit,delivary.delivary_date,delivary.delivary_address
from ( (customer inner join orders on customer.cus_id = orders.cus_id ) inner join delivary on customer.cus_id = delivary.cus_id);
```

Results Messages

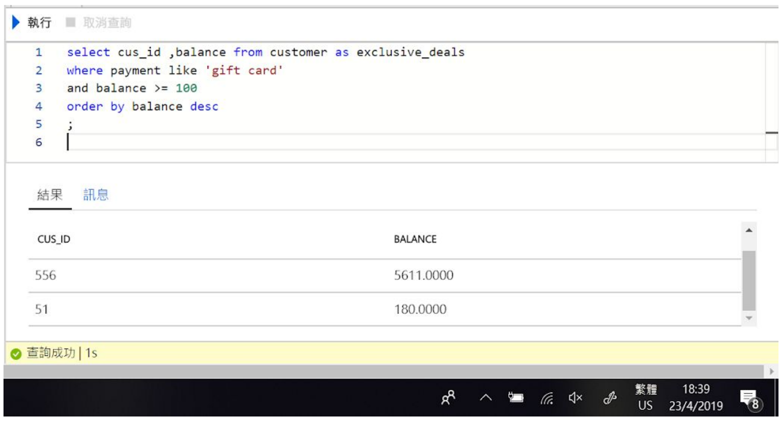
CUS_ID	ORDER_ID	TOTAL_PRICE	TOTAL_UNIT	DELIVARY_DATE	DELIVARY_ADDRESS
789	1	2500.0000	5	2019-03-05T00:00:00.0000000	Eastern Building Rm3901
789	10	1020.0000	18	2019-03-05T00:00:00.0000000	Eastern Building Rm3901
55	2	30.0000	1	2019-03-05T00:00:00.0000000	Space Building Rm311
99	3	2250.0000	78	2019-03-09T00:00:00.0000000	hkuspacepcroom902
51	4	246.0000	41	2019-03-11T00:00:00.0000000	Hong Kong House Rm1506
51	6	1170.0000	12	2019-03-11T00:00:00.0000000	Hong Kong House Rm1506
331	5	1446.0000	106	2019-03-17T00:00:00.0000000	Eastern Building Rm3901

----
select cus_id ,balance from customer as exclusive_deals
where payment = 'gift card'
and balance >= 100
order by balance desc
limit 10;

this statement may use for checking how much money customer left in account, if it is large than some values(e.g $100 usd) , push notification of exclusive sale.

updated as
function 2(for business) :

```
select cus_id ,balance from customer
where payment like 'gift card'
and balance >= 100
order by balance desc
```



執行 取消查詢

```sql
select cus_id ,balance from customer as exclusive_deals
where payment like 'gift card'
and balance >= 100
order by balance desc
;
```

結果 訊息

```
CUS_ID    BALANCE
556       5611.0000
51        180.0000
```

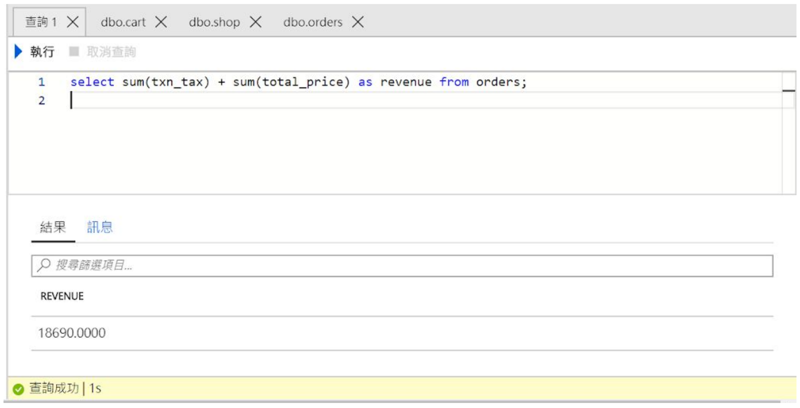
查詢成功 | 1s

繁體 US 18:39 23/4/2019

;

function 3(make use of agg function):

select sum(txn_tax) + sum(total_price) as revenue from orders;



查詢 1 dbo.cart dbo.shop dbo.orders

執行 取消查詢

```sql
select sum(txn_tax) + sum(total_price) as revenue from orders;
```

結果 訊息

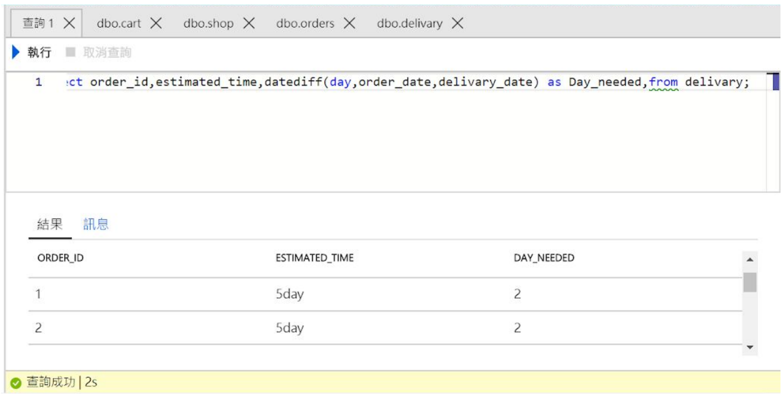
搜尋篩選項目...

```
REVENUE
18690.0000
```

查詢成功 | 1s

function 4 (delivary time needed):

select order_id,estimated_time,datediff(day,order_date,delivary_date) as Day_needed from delivary;



查詢 1 dbo.cart dbo.shop dbo.orders dbo.delivary

執行 取消查詢

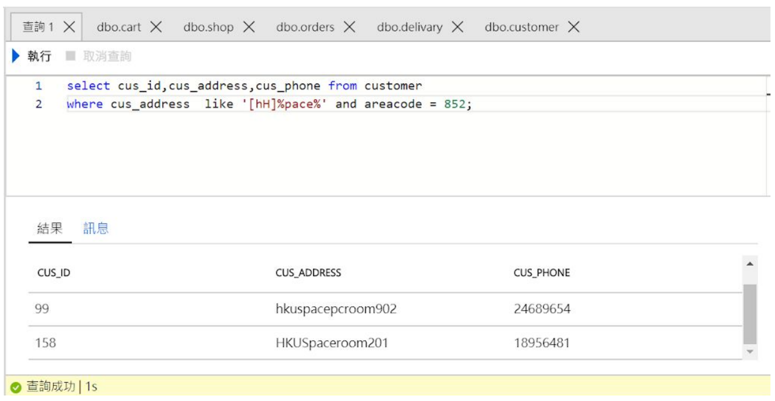
```sql
:ct order_id,estimated_time,datediff(day,order_date,delivary_date) as Day_needed,from delivary;
```

結果 訊息

```
ORDER_ID    ESTIMATED_TIME    DAY_NEEDED
1           5day              2
2           5day              2
```

查詢成功 | 2s

function 5 (if we know our college buy things here, we can offer a fast delivary)(make use of wildcard):



```
select cus_id,cus_address,cus_phone from customer
where cus_address  like '[hH]%pace%' and areacode = 852;
```

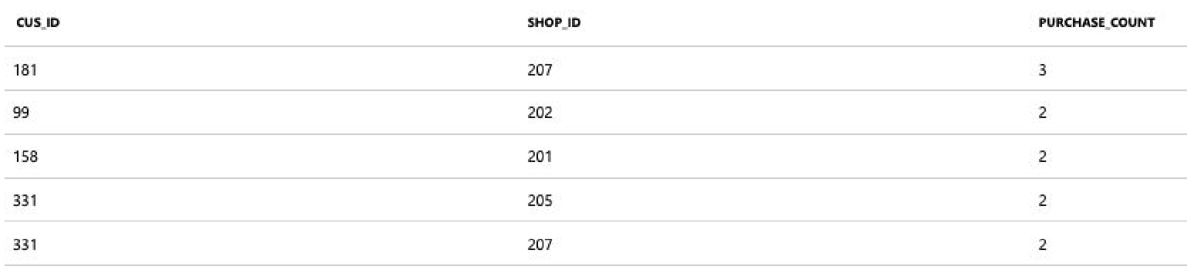
CUS_ID	CUS_ADDRESS	CUS_PHONE
99	hkuspacepcroom902	24689654
158	HKUSpaceroom201	18956481

查詢成功 | 1s

```
select cus_id,cus_address,cus_phone from customer
where cus_address  like '[hH]%pace%' and areacode = 852;
```

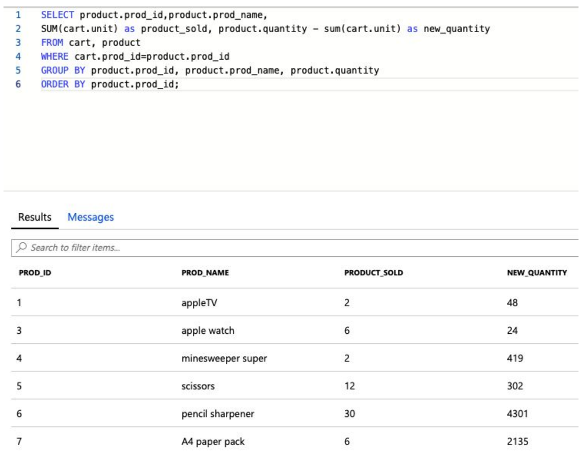
Function 6: provide discount for frequent buyers

```
SELECT TOP 5 cus_id, shop_id, COUNT(shop_id) AS purchase_count
FROM (orders INNER JOIN cart ON orders.order_id = cart.order_id) INNER JOIN product
ON product.prod_id = cart.prod_id
GROUP BY shop_id, cus_id
ORDER BY purchase_count DESC
```



CUS_ID	SHOP_ID	PURCHASE_COUNT
181	207	3
99	202	2
158	201	2
331	205	2
331	207	2

func7 updated



```
SELECT product.prod_id,product.prod_name,
SUM(cart.unit) as product_sold, product.quantity - sum(cart.unit) as new_quantity
FROM cart, product
WHERE cart.prod_id=product.prod_id
GROUP BY product.prod_id, product.prod_name, product.quantity
ORDER BY product.prod_id;
```

Results Messages

Search to filter items...

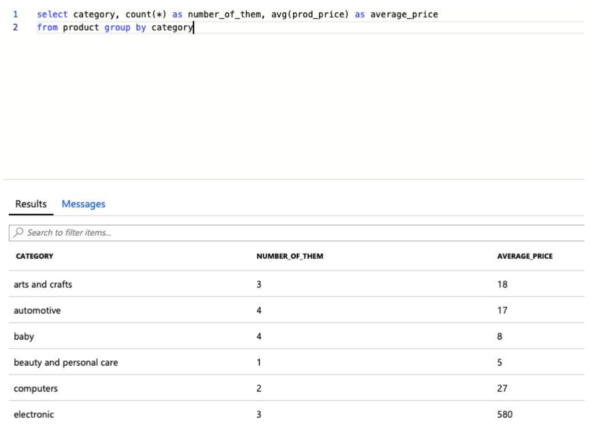
PROD_ID	PROD_NAME	PRODUCT_SOLD	NEW_QUANTITY
1	appleTV	2	48
3	apple watch	6	24
4	minesweeper super	2	419
5	scissors	12	302
6	pencil sharpener	30	4301
7	A4 paper pack	6	2135

```
SELECT product.prod_id,product.prod_name,
SUM(cart.unit) as product_sold, product.quantity - sum(cart.unit) as new_quantity
FROM cart, product
WHERE cart.prod_id=product.prod_id
GROUP BY product.prod_id, product.prod_name, product.quantity
ORDER BY product.prod_id;
```

calculating the number of product sold and the quantity left separately

func8

```
select category, count(*) as number_of_them, avg(prod_price) as average_price
from product group by category
```



```
select category, count(*) as number_of_them, avg(prod_price) as average_price
from product group by category
```

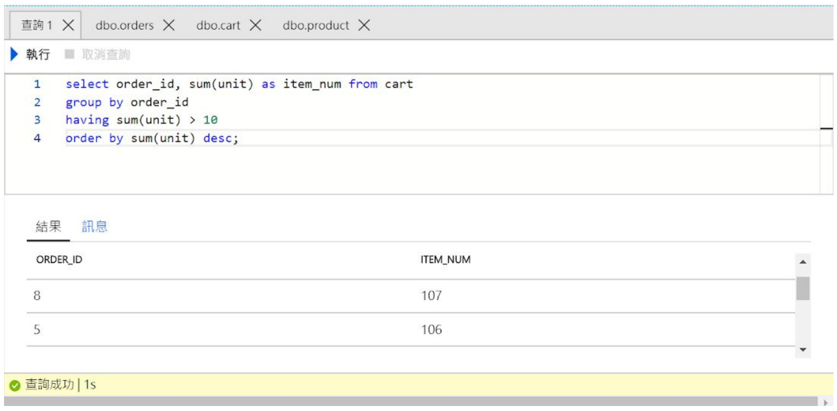
Results Messages

Search to filter items...

CATEGORY	NUMBER_OF_THEM	AVERAGE_PRICE
arts and crafts	3	18
automotive	4	17
baby	4	8
beauty and personal care	1	5
computers	2	27
electronic	3	580

counting total number of product in each category and their average price

function 9 (trace the huge number of item record):



```
select order_id, sum(unit) as item_num from cart
group by order_id
having sum(unit) > 10
order by sum(unit) desc;
```

ORDER_ID	ITEM_NUM
8	107
5	106

```
select order_id, sum(unit) as item_num from cart
group by order_id
having sum(unit) > 10
order by sum(unit) desc;
```