

DB67 โจทย์ฝึกทำข้อสอบ Stored Procedure

1. stored procedure ทุกข้อ ต้องมีคำสั่งทดสอบการทำงานของ procedure ด้วย เช่น
exec procedure นั้น ๆ
select * from table ที่เกี่ยวข้องกับการทำงาน
2. ข้อใดที่มีการ create table, create view ต้องเขียนคำสั่ง create table view ด้วย
3. เมื่อทำเสร็จแต่ละข้อ ควร drop procedure ทั้ง และ
ควรรัน script เพื่อ drop table หลัก และ insert ข้อมูลใหม่ เพื่อให้ข้อมูลกลับมาเหมือนเดิม

1. msp_demo1 (@year, @shop_name)

แสดง รหัสออเดอร์ รหัสลูกค้า ชื่อลูกค้า วันที่ซื้อ ยอดการซื้อของออเดอร์นั้น และ ชื่อร้านค้า
โดยแสดงเฉพาะออเดอร์ในปี @year จากร้านชื่อ @shop_name

ตัวอย่างการรันคำสั่ง

```
exec msp_demo1 '2023', 'pDrink'
exec msp_demo1 '2024', 'wDrink'
```

```
create procedure msp_demo1
    @year      varchar(4),
    @shop_name varchar(100)
as
begin
    select orders.oid, customer.cid, customer.name, odate, final_price
    from orders, shop, customer
    where orders.shopid = shop.sid
    and orders.cusid = customer.cid
    and year(odate) = @year
    and shop.name = @shop_name
end
```

```
exec msp_demo1 '2023', 'pDrink'
```

	oid	cid	name	odate	final_price	name
1	61	c01	Jim	2023-01-10	50	pDrink
2	63	c03	Joe	2023-03-15	1700	pDrink
3	65	c05	Jane	2023-05-25	1140	pDrink
4	72	c12	Cat	2023-12-01	220	pDrink
5	73	c13	Rat	2023-12-15	1200	pDrink
6	75	c15	Ant	2023-12-31	1200	pDrink
7	76	c01	Jim	2023-11-01	130	pDrink

```
exec msp_demo1 '2024', 'wDrink'
```

	oid	cid	name	odate	final_price	name
1	82	c02	Jack	2024-02-12	68000	pLuxury
2	87	c07	Pam	2024-07-15	69000	pLuxury
3	93	c13	Rat	2024-12-15	68000	pLuxury
4	104	c01	Jim	2024-09-04	40000	pLuxury
5	107	c01	Jim	2024-09-07	1249000	pLuxury
6	115	c15	Ant	2024-09-15	108000	pLuxury

2.msp_demo2(@year, @shop_name) (คล้ายข้อ 1. แต่มีการเพิ่มเงื่อนไข)

แสดง รหัสออเดอร์ รหัสลูกค้า ชื่อลูกค้า วันที่ซื้อ ยอดการซื้อของออเดอร์นั้น และ ชื่อร้านค้า

โดยแสดงเฉพาะออเดอร์ในปี @year จากร้านชื่อ @shop_name

ถ้าปี @year ที่ส่งมาเป็น null ให้แสดงข้อมูลออเดอร์ของทุกปี

```
create procedure msp_demo2
    @year      varchar(4),
    @shop_name  varchar(100)
as
begin
    if(@year is null)    -- แสดงข้อมูลทุกปี ดังนั้นไม่ต้องใส่เงื่อนไข @year
        select orders.oid, customer.cid, customer.name, odate, final_price, shop.name
        from orders, shop, customer
        where orders.shopid = shop.sid
        and orders.cusid = customer.cid
        and shop.name = @shop_name
    else    -- แสดงข้อมูลทุกปี
        select orders.oid, customer.cid, customer.name, odate, final_price, shop.name
        from orders, shop, customer
        where orders.shopid = shop.sid
        and orders.cusid = customer.cid
        and year(odate) = @year
        and shop.name = @shop_name
end
```

exec msp_demo2 '2023', 'wDrink' -- แสดงออเดอร์ของปี 2023

	oid	cid	name	odate	final_price	name
1	69	c09	Pome	2023-09-12	4000	wDrink

exec msp_demo2 '2024', 'wDrink' -- แสดงออเดอร์ของปี 2024

	oid	cid	name	odate	final_price	name
1	85	c05	Jane	2024-05-25	4000	wDrink
2	105	c01	Jim	2024-09-05	4000	wDrink
3	119	c04	Joy	2024-09-19	24000	wDrink

exec msp_demo2 null, 'wDrink' -- แสดงออเดอร์ของทุกปี

	oid	cid	name	odate	final_price	name
1	3	c03	Joe	2020-03-10	12000	wDrink
2	10	c04	Joy	2020-04-20	4000	wDrink
3	16	c10	Pool	2020-10-22	4000	wDrink
4	26	c06	Pete	2021-06-05	4000	wDrink
5	33	c13	Rat	2021-12-15	4000	wDrink
6	41	c01	Jim	2022-01-10	4000	wDrink
7	53	c13	Rat	2022-12-15	4000	wDrink
8	69	c09	Pome	2023-09-12	4000	wDrink
9	85	c05	Jane	2024-05-25	4000	wDrink
10	105	c01	Jim	2024-09-05	4000	wDrink
11	119	c04	Joy	2024-09-19	24000	wDrink

3.msp_demo3(@year)

เพื่อแสดงรหัสลูกค้า ชื่อลูกค้า ปี ยอดรวมแต้มสะสมที่ใช้ในปี @year และ ยอดรวมการซื้อในปี @year

หมายเหตุ ต้องสร้าง view demo_view3(cusid, year, total_discount, total_buy) ก่อน

การแสดงผลข้อมูลใน procedure ต้องมีการ join: customer, demo_view2

```
create view demo_view3(cusid, year, total_discount, total_buy)
as
    select
        orders.cusid,
        year(orders.odate) as year,
        sum(orders.discount_point) as total_discount,
        sum(orders.final_price) as total_buy
    from
        orders
    group by
        orders.cusid, year(orders.odate);
```

```
create procedure msp_demo3
    @year varchar(4)
as
begin
    select cid, name, year, total_discount, total_buy
    from customer, demo_view3
    where customer.cid = demo_view3.cusid
    and year = @year
end
```

exec msp_demo3 2022

	cid	name	year	total_discount	total_buy
1	c01	Jim	2022	5000	399090
2	c02	Jack	2022	0	90020
3	c03	Joe	2022	0	350
4	c04	Joy	2022	5000	96200
5	c05	Jane	2022	2000	58020
6	c06	Pete	2022	0	700
7	c07	Pam	2022	1000	49000
8	c08	Paul	2022	2000	78000
9	c09	Pome	2022	600	39400
10	c10	Pool	2022	0	900

exec msp_demo3 2024

	cid	name	year	total_discount	total_buy
1	c01	Jim	2024	11000	1683230
2	c02	Jack	2024	22110	571590
3	c03	Joe	2024	5115	97085
4	c04	Joy	2024	0	24700
5	c05	Jane	2024	5	11395
6	c06	Pete	2024	0	220
7	c07	Pam	2024	1000	69000
8	c08	Paul	2024	0	7200
9	c09	Pome	2024	5000	175630
10	c10	Pool	2024	0	7750

4.msp_demo4 (@year, @seller_name)

เพื่อแสดงรหัสร้านค้า ชื่อร้านค้า ปี และ ยอดรวมการขายในปี @year ร้านที่เป็นของผู้ขายชื่อ @seller_name

หมายเหตุ ต้องสร้าง view demo_view4(shopid, year , total_sell) ก่อน

การแสดงผลข้อมูลใน procedure ต้องมีการ join: shop, seller, demo_view4

ตัวอย่างการทำงาน

```
exec msp_demo4 '2023', 'prymania'
```

	sid	name	year	total_sell
1	s01	pDrink	2023	5640
2	s02	pLuxury	2023	704400
3	s03	pFruit	2023	1695

```
exec msp_demo4 '2024', 'prymania'
```

	sid	name	year	total_sell
1	s01	pDrink	2024	16440
2	s02	pLuxury	2024	1602000
3	s03	pFruit	2024	16580

```
exec msp_demo4 '2023', 'iSell'
```

	sid	name	year	total_sell
1	s07	iSellGood	2023	691980

5.msp_demo5 (@year)

เพื่อเพิ่มแต้มสะสมให้ลูกค้า ที่มียอดการซื้อทั้งหมดในปี @year ตามเงื่อนไขต่อไปนี้

- ถ้ายอดซื้อมากกว่า 50000 เพิ่มแต้มสะสมให้ 500 แต้ม
- ถ้ายอดซื้อมากกว่า 20000 เพิ่มแต้มสะสมให้ 100 แต้ม
- ถ้ายอดซื้อมากกว่า 5000 เพิ่มแต้มสะสมให้ 20 แต้ม

หมายเหตุ ใช้ view demo_view3 ช่วยในการทำงาน

ตัวอย่างการตรวจสอบการทำงานของ procedure

`select * from demo_view3 where year = 2024` -- เพื่อดูยอดซื้อปี 2024 ของลูกค้า

	cusid	year	total_discount	total_buy
1	c01	2024	11000	1683230
2	c02	2024	22110	571590
3	c03	2024	5115	97085
4	c04	2024	0	24700
5	c05	2024	5	11395
6	c06	2024	0	220
7	c07	2024	1000	69000
8	c08	2024	0	7200
9	c09	2024	5000	175630
10	c10	2024	0	7750

ดังนั้น เมื่อ exec msp_demo5 แล้ว ลูกค้าจะได้รับแต้มสะสมเพิ่มขึ้น ตามยอดซื้อ เช่น

- c01 ได้รับแต้มสะสม 500 แต้ม
- c04 ได้รับแต้มสะสม 100 แต้ม
- c06 ไม่ได้รับแต้มสะสม
- c08 ได้รับแต้มสะสม 20 แต้ม

`select * from customer` -- เพื่อดู point ของลูกค้าก่อนรัน procedure

	cid	name	phone	address	birthday	point	suggestedid
1	c01	Jim	0666	bkk 335	2000-01-01	71	NULL
2	c02	Jack	0555	bkk 335	2000-01-01	97	NULL
3	c03	Joe	0633	bkk 335	2001-05-01	0	NULL
4	c04	Joy	0645	bkk 335	1995-09-01	14	NULL
5	c05	Jane	023	bkk 335	2002-03-01	86	NULL
6	c06	Pete	NULL	NULL	2016-01-01	58	c01
7	c07	Pam	NULL	NULL	2015-02-01	7	c01
8	c08	Paul	NULL	NULL	2014-03-01	64	c01
9	c09	Pome	NULL	NULL	2012-04-01	72	c01
10	c10	Pool	NULL	NULL	2012-05-01	41	c01

`exec msp_demo5 '2024'` -- รัน procedure เพื่อเพิ่มแต้มสะสมให้ลูกค้า

`select * from customer` -- เพื่อดู point ของลูกค้าหลังรัน procedure

	cid	name	phone	address	birthday	point	suggestedid
1	c01	Jim	0666	bkk 335	2000-01-01	571	NULL
2	c02	Jack	0555	bkk 335	2000-01-01	597	NULL
3	c03	Joe	0633	bkk 335	2001-05-01	500	NULL
4	c04	Joy	0645	bkk 335	1995-09-01	114	NULL
5	c05	Jane	023	bkk 335	2002-03-01	106	NULL
6	c06	Pete	NULL	NULL	2016-01-01	58	c01
7	c07	Pam	NULL	NULL	2015-02-01	507	c01
8	c08	Paul	NULL	NULL	2014-03-01	84	c01
9	c09	Pome	NULL	NULL	2012-04-01	572	c01

ดูแต้มสะสมของลูกค้าทุกคน
มีแต้มเพิ่มตามยอดซื้อ

6.msp_demo6 (@year, @amount, @percent)

เพื่อลดราคาสินค้า ที่ในปี @year ขายได้น้อยกว่า @amount ขึ้น โดยลดราคาลง @percent %

ตัวอย่างการตรวจสอบการทำงานของ **procedure**

exec msp_demo6 2024, 20 , 50

select -- คำสั่งเพื่อดูจำนวนสินค้าที่ขายได้มากกว่า 20 ชิ้นในปี 2024 แสดงว่า ไม่ต้องลดราคาสินค้าเหล่านี้

	pid	total_amount
1	p01	26
2	p02	108
3	p04	31
4	p07	23
5	p12	21
6	p22	108
7	p23	32

select * from product -- เพื่อดูราคาสินค้า ก่อนรัน **procedure**

	pid	shopid	name	price	max_point	stock
1	p01	s01	Pepsi	10	0	200
2	p02	s01	Coke	10	0	500
3	p03	s01	Juice	100	0	100
4	p04	s01	Beer	200	0	20
5	p05	s01	Wine	1000	0	10
6	p06	s02	iPad	50000	1000	500
7	p07	s02	iPhone	60000	2000	200
8	p08	s02	iMouse	10000	50	100
9	p09	s02	idoll	20000	500	5
10	p10	s02	iRemote	9662.56363636363	100	6

exec msp_demo6 2024, 20 , 50

-- ลดราคาสินค้าที่ปี 2024 ขายได้น้อยกว่า 20 ชิ้น โดยลดราคาลง 50%

select * from product -- เพื่อดูราคาสินค้า หลังรัน **procedure**

	pid	shopid	name	price	max_point	stock
1	p01	s01	Pepsi	10	0	200
2	p02	s01	Coke	10	0	500
3	p03	s01	Juice	50	0	100
4	p04	s01	Beer	200	0	20
5	p05	s01	Wine	500	0	10
6	p06	s02	iPad	25000	1000	500
7	p07	s02	iPhone	60000	2000	200
8	p08	s02	iMouse	5000	50	100
9	p09	s02	idoll	10000	500	5
10	p10	s02	iRemote	4831.28181818182	100	6

สินค้าที่ขายได้น้อยกว่า 20 ชิ้น เช่น

p03, p05, p06

ลดราคาลง 50 %

7.msp_demo7 (@year, @shop_name)

เพื่อแสดงรหัสการขาย รหัสร้านค้า จำนวนวันที่ใช้ในการจัดการการขาย ปีที่ทำการขาย และ ชื่อร้าน
 ของปี @year จากร้านชื่อ @shop_name โดยเรียงลำดับตามจำนวนวันที่ใช้จัดการการขาย จากมากไปน้อย
 ถ้า @shop_name เป็น null ให้แสดงข้อมูลจากทุกร้าน โดยเรียงลำดับตามชื่อร้าน และ จำนวนวันจากมากไปน้อย

ตัวอย่างการตรวจสอบการทำงานของ procedure

```
exec msp_demo7 '2024', 'wDrink'
```

-- แสดงข้อมูลของร้าน wDrink

	oid	shopid	year	จำนวนวัน	name
1	119	s05	2024	8	wDrink
2	85	s05	2024	5	wDrink
3	105	s05	2024	5	wDrink

```
exec msp_demo7 '2024', 'iSellBad'
```

-- แสดงข้อมูลของร้าน iSellBad

	oid	shopid	year	จำนวนวัน	name
1	98	s08	2024	18	iSellBad
2	117	s08	2024	10	iSellBad
3	91	s08	2024	9	iSellBad
4	90	s08	2024	7	iSellBad
5	113	s08	2024	5	iSellBad

```
exec msp_demo7 '2024', null
```

-- แสดงข้อมูลของทุกร้าน

	oid	shopid	year	จำนวนวัน	name
1	98	s08	2024	18	iSellBad
2	117	s08	2024	10	iSellBad
3	91	s08	2024	9	iSellBad
4	90	s08	2024	7	iSellBad
5	113	s08	2024	5	iSellBad
6	97	s07	2024	15	iSellGood
7	99	s07	2024	15	iSellGood
8	116	s07	2024	11	iSellGood
9	118	s07	2024	9	iSellGood
10	89	s07	2024	7	iSellGood
11	94	s07	2024	5	iSellGood
12	111	s07	2024	5	iSellGood
13	102	s07	2024	5	iSellGood
14	101	s07	2024	4	iSellGood
15	92	s01	2024	18	pDrink

8. msp_demo8 (@year, @shopid)

เพื่อแสดงยอดขายรายเดือน ในแต่ละเดือน ของร้านค้ารหัส @shopid ในปี @year

ตัวอย่างการตรวจสอบการทำงานของ **procedure**

`exec msp_demo8 '2023', 's01'`

	month	year	ยอดขายรวม
1	1	2023	50
2	3	2023	1700
3	5	2023	1140
4	11	2023	130
5	12	2023	2620

`exec msp_demo8 '2024', 's01'`

	month	year	ยอดขายรวม
1	1	2024	210
2	6	2024	220
3	7	2024	1200
4	8	2024	1200
5	9	2024	9450
6	12	2024	4160

`exec msp_demo8 '2023', 's07'`

	month	year	ยอดขายรวม
1	7	2023	10000
2	8	2023	95990
3	10	2023	255000
4	11	2023	330990

`exec msp_demo8 '2024', 's07'`

	month	year	ยอดขายรวม
1	8	2024	172000
2	9	2024	861980
3	10	2024	127000
4	12	2024	105000

9. msp_search_customer(@cid, @name, @phone)

เพื่อค้นหาข้อมูลลูกค้า ด้วย รหัสลูกค้า หรือ ชื่อ หรือ เบอร์โทร

โดยสามารถค้นหาด้วยค่าใดค่าหนึ่ง

ถ้าค้นหาด้วย cid หรือ phone ให้ค้นหาแบบ ตรงตัว เช่น c01 หาลูกค้ารหัส c01

ถ้าค้นหาด้วย name ให้ค้นหาด้วยชื่อที่มีตัวอักษรนั้นอยู่ เช่น ฟา หาลูกค้าที่มีคำว่า ฟา อยู่ในชื่อ

ลำดับความสำคัญของการค้นหา คือ รหัสลูกค้า ชื่อ เบอร์โทร

ตัวอย่างการตรวจสอบการทำงานของ procedure

```
exec search_customer null, 'pa', null -- ค้นหาลูกค้าที่มี pa อยู่ในชื่อ
```

	cid	name	phone	address	birthday	point	suggestedid
1	c07	Pam	NULL	NULL	2015-02-01	0	c01
2	c08	Paul	NULL	NULL	2014-03-01	0	c01

```
exec search_customer null, 'j', null -- ค้นหาลูกค้าที่มี j อยู่ในชื่อ
```

	cid	name	phone	address	birthday	point	suggestedid
1	c01	Jim	0666	bkk 335	2000-01-01	10000	NULL
2	c02	Jack	0555	bkk 335	2000-01-01	5000	NULL
3	c03	Joe	0633	bkk 335	2001-05-01	2000	NULL
4	c04	Joy	0645	bkk 335	1995-09-01	3000	NULL
5	c05	Jane	023	bkk 335	2002-03-01	5000	NULL

```
exec search_customer 'c07', 'j', null -- ค้นหาลูกค้ารหัส c07
```

	cid	name	phone	address	birthday	point	suggestedid
1	c07	Pam	NULL	NULL	2015-02-01	0	c01

```
exec search_customer null, null, '0645' -- ค้นหาลูกค้าที่มีเบอร์โทร 0645
```

	cid	name	phone	address	birthday	point	suggestedid
1	c04	Joy	0645	bkk 335	1995-09-01	3000	NULL

DB67 โจทย์ฝึกทำข้อสอบ Trigger

ทุกข้อเมื่อทดสอบคำสั่งแล้ว ต้อง drop trigger ที่เพิ่ม

1. tg_customerPhoneLog

เมื่อลูกค้ามีการเปลี่ยนแปลงเบอร์โทร ต้องการเก็บข้อมูลการเปลี่ยนแปลง ไว้ใน table

CustomerPhoneLog(cusid, old_phone, new_phone, change_date)

ข้อนี้ต้องสร้าง 2 อย่างคือ

1. table: CustomerPhoneLog
2. trigger : tg_customerPhoneLog

2. tg_ProductNoDiscount

ต้องการล็อคให้ สินค้า **ไม่สามารถลดราคา** ได้ ดังนั้น

เมื่อมีการแก้ไขราคาสินค้าต้องเช็คค่า ถ้าราคาต่ำกว่าราคาเดิมต้องไม่สามารถแก้ไขได้ และยกเลิกการแก้ไข

ถ้าแก้ไขได้ ให้แสดงข้อความ แก้ไขสำเร็จ

ถ้าแก้ไขไม่ได้ ให้แสดงข้อความยกเลิก และต้องบอกราคาเดิม และราคาใหม่ที่ต้องการแก้ไขด้วย

**เพื่อลดความซับซ้อนของ code ให้คิดว่าเป็นการแก้ไขข้อมูลทีละ record เท่านั้น

ตัวอย่างการทำงาน

```
update product set price = price + 500 where pid = 'p01' -- ok
```

แก้ไขราคาสินค้าสำเร็จ

```
update product set price = price - 20 where pid = 'p01' -- rollback
```

```
Msg 50000, Level 16, State 1, Procedure tg_ProductNoDiscount, Line 21 [Batch Start Line 39]
ราคาสินค้าต่ำกว่าราคาเดิม ยกเลิกคำสั่ง: ราคาเดิม 1010, ราคาใหม่ 990
Msg 3609, Level 16, State 1, Line 41
The transaction ended in the trigger. The batch has been aborted.
```

3. tg_checkProductPrice20Percent

เมื่อมีการเพิ่มราคาสินค้า ต้องมีการเช็คล็อกไม่ให้เพิ่มราคาสินค้าเกิน 20 %

ถ้ามีการเพิ่มราคาเกิน 20% ต้องยกเลิกคำสั่ง update

4. tg_checkProductPointForDiscount

** trigger นี้เฝ้า table order_detail

เมื่อมีการซื้อสินค้า และใช้ point เพื่อเป็นส่วนลดการซื้อสินค้าชิ้นนั้น

ต้องมีการเช็ค point ที่จะใช้เป็นส่วนลด เกินค่า point ของสินค้าที่กำหนดไว้หรือไม่

ถ้าใช้ point เกิน ไม่สามารถขายได้ และยกเลิกการขายสินค้าชิ้นนั้น โดยต้องบอกว่าสินค้านั้นใช้ point ได้เท่าไร

อธิบาย table

product (pid, shopid, name, price, **max_point**, stock)

orders (oid, cusid, shopid, odate, fdate, list_price, discount_point, final_price, status, review)

order_detail(did, oid, pid, amount, **use_point**, item_price)

max_point คือแต้มสูงสุดที่ใช้ลดราคาสินค้าชิ้นนี้ได้

use_point คือแต้มที่ต้องการใช้เป็นส่วนลด

ตัวอย่างการทำงาน -- ข้อมูล product p06 ใช้แต้ม ลดได้ 1000 แต้ม

```
insert into order_detail(oid, pid, amount, use_point) values(1, 'p06', 1, 1000) -ok
```

```
use_point = 1000, max_point = 1000
```

ขายสินค้าสำเร็จ

```
insert into order_detail(oid, pid, amount, use_point) values(1, 'p06', 1, 5000) --rollback
```

```
use_point = 5000, max_point = 1000
```

Msg 50000, Level 16, State 1, Procedure tg_checkProductPointForDiscount, Line 21 [Batch Start Line 154]
ไม่สามารถขายได้: สินค้าสามารถใช้ point ลดราคาได้เพียง 1000 เท่านั้น

Msg 3609, Level 16, State 1, Line 156

The transaction ended in the trigger. The batch has been aborted.

5. tg_increaseCustomerPointAfterOrders

** trigger นี้เฝ้า table orders

ทุกครั้งที่ลูกค้าซื้อสินค้า ให้เพิ่มแต้มสะสมให้ลูกค้า โดยยอดขายทุก 100 บาท ได้แต้มสะสมเพิ่ม 1 แต้ม

เช่น ถ้ายอดขาย 200 บาท จะได้แต้มเพิ่ม 2 แต้ม
 ถ้ายอดขาย 3058 บาท จะได้แต้มเพิ่ม 30 แต้ม

trigger ต้องแสดงข้อมูล

ยอดขายครั้งนี้	_____ บาท
ลูกค้าได้รับแต้มเพิ่มจากการซื้อ	_____ แต้ม
แต้มสะสมของลูกค้าหลังการซื้อครั้งนี้	_____ แต้ม

ตัวอย่างการทำงาน -- ข้อมูล customer c02 มีแต้มสะสม 5000 แต้ม

```
insert into orders(oid, cusid, shopid, final_price) values (700, 'c02', 's01', 200)
```

(1 row affected)

ยอดขายครั้งนี้	200
ลูกค้าได้รับแต้มเพิ่ม =	2
แต้มสะสมของลูกค้า =	5002

```
insert into orders(oid, cusid, shopid, final_price) values (701, 'c02', 's01', 250)
```

(1 row affected)

ยอดขายครั้งนี้	250
ลูกค้าได้รับแต้มเพิ่ม =	2
แต้มสะสมของลูกค้า =	5004

```
insert into orders(oid, cusid, shopid, final_price) values (702, 'c02', 's01', 2328)
```

(1 row affected)

ยอดขายครั้งนี้	2328
ลูกค้าได้รับแต้มเพิ่ม =	23
แต้มสะสมของลูกค้า =	5027

6. tg_checkShopStatus

trigger นี้เฝ้า table orders

ร้านค้า มีสถานะ (status) สามารถเป็นได้ 3 สถานะคือ 1- เปิดขายปกติ, 2 – ปิดชั่วคราว และ 0-ยกเลิกร้านค้า
 เมื่อมีการสั่งซื้อสินค้า ต้องเช็คค่า ชื่อจากสินค้าที่มีสถานะ เป็นการเปิดขายปกติหรือไม่
 ถ้าสถานะร้านค้าไม่ใช่เปิดขายปกติ ต้องยกเลิกการสั่งซื้อนั้น

7. tg_bonusSuggestedCustomer

trigger นี้เฝ้า table customer

เมื่อมีการเพิ่มลูกค้าใหม่ ให้เพิ่มแต้มสะสม (point) ให้กับลูกค้าที่เป็นผู้แนะนำจำนวน 10 แต้ม

เช่น

```
insert into CUSTOMER values('c20', 'Killua', null, null, '7-7-2000', 0, 'c05')
```

แสดงว่า ลูกค้า C05 เป็นผู้แนะนำ c20 ดังนั้นให้เพิ่ม point ให้ c05 จำนวน 10 แต้ม

แสดงข้อความบอกด้วย ว่าลูกค้ารหัส cid. ได้รับ point เพิ่มจำนวน ทำให้ตอนนี้มี point =