Phurba Sherpa Prof - Kevin Gravesande

Final Project.

Write and test seven (7) triggers for seven (7) separate tables to implement the business rules.

- At least two (2) of the triggers must be for delete.

1st Delete Trigger

SQLQuery2.sql - O...CST\23307960 (68)) SQLQuery1.sql - O...CST\23

```
Create Trigger Orders_delete
on orders
after delete
as
begin
Declare @orderid int
select @orderid = deleted.orderid
from deleted
insert into orderlogs
values(@orderid,'Deleted')
END

delete from orders where orderid = '3';
delete from orders where orderid = '88';

select * from orders;
select * From orderlogs;
```

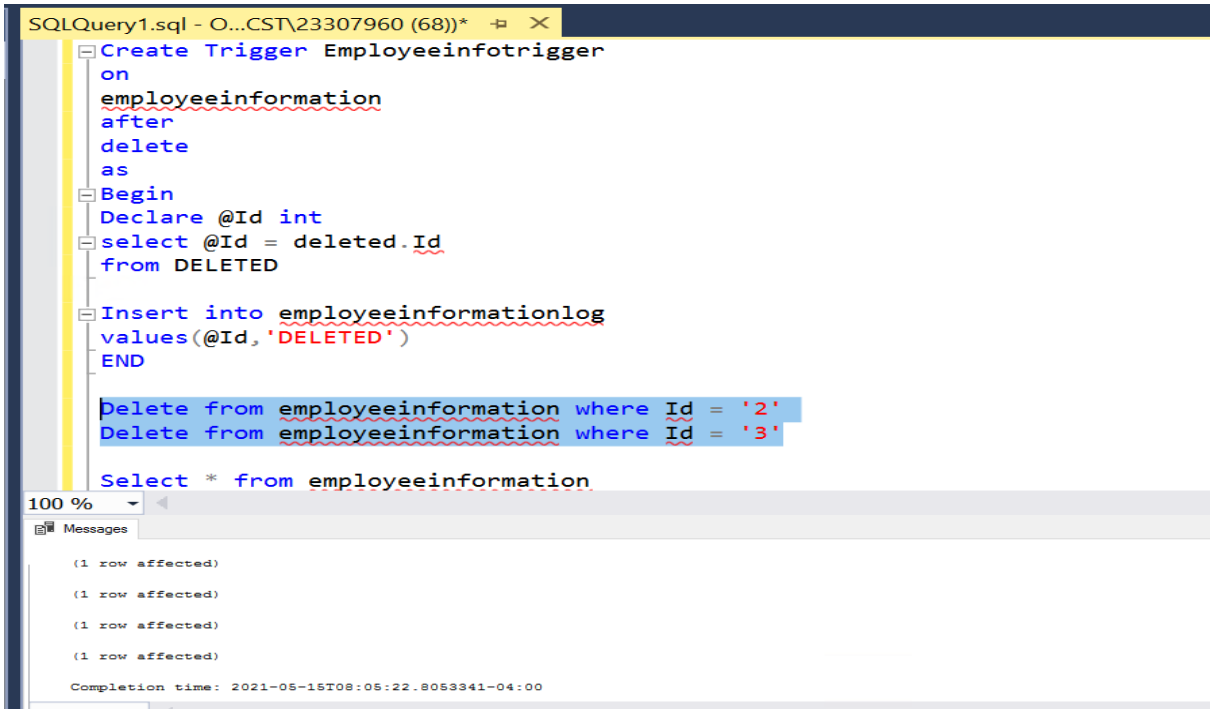
100 %

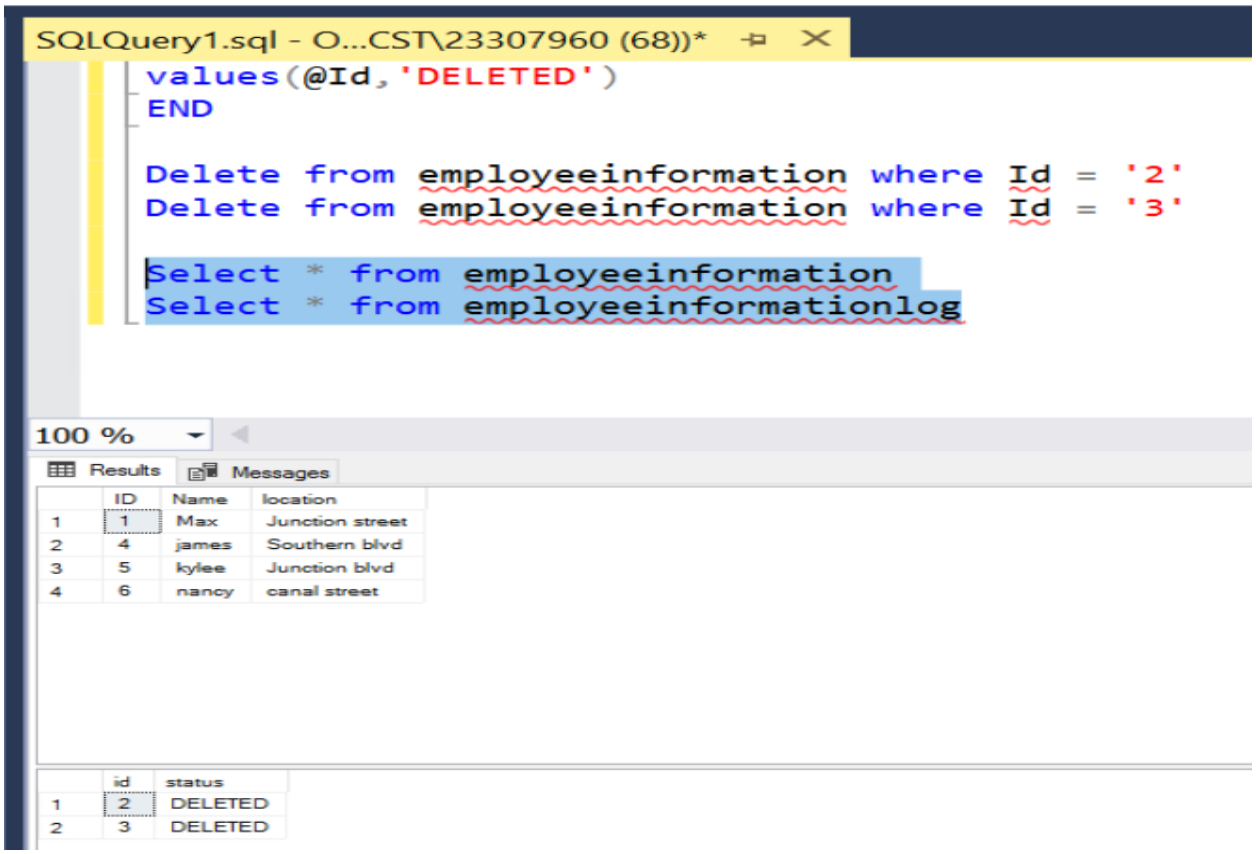
Results Messages

	orderid	customer_customerid	employee_employeeid	orderdate	Product_ProductID
1	50	1	20	2020-09-02	30
2	51	2	21	2020-01-02	31
3	52	3	22	2020-01-06	32
4	53	4	23	2020-07-06	33
5	54	5	24	2020-07-06	34
6	99	777	889	NULL	NULL

	Orderid	status
1	3	Deleted
2	88	ADDED
3	88	ADDED
4	88	Deleted

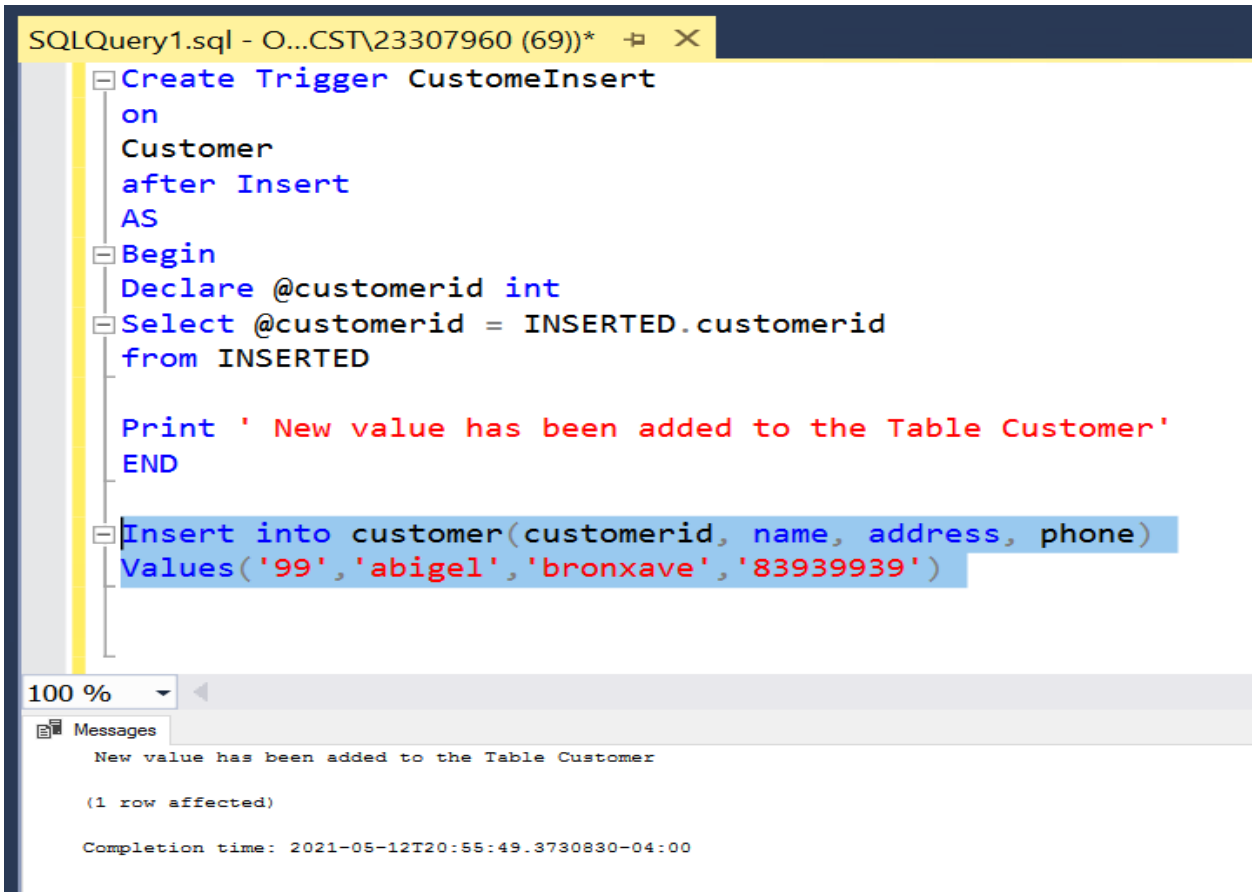
2nd Delete Trigger.

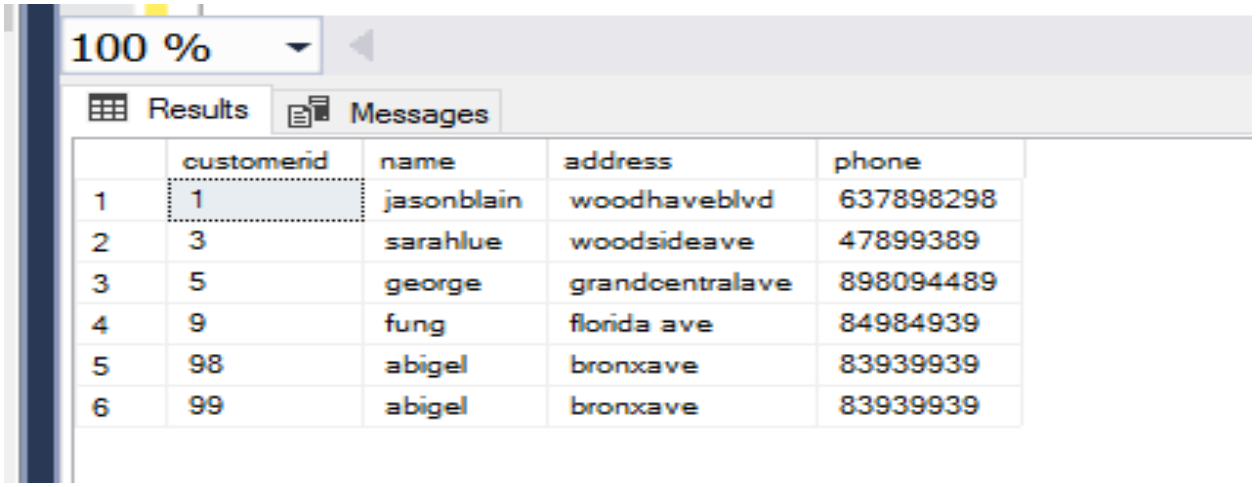




- At least two (2) of the triggers must be for insert.

1st Insert





	customerid	name	address	phone
1	1	jasonblain	woodhaveblvd	637898298
2	3	sarahlue	woodsideave	47899389
3	5	george	grandcentralave	898094489
4	9	fung	florida ave	84984939
5	98	abigel	bronxave	83939939
6	99	abigel	bronxave	83939939

2nd Insert

SQLQuery2.sql - O...CST\23307960 (68)) SQLQuery1.sql - O...CST\23307960 (69))*

```
Create Trigger OrderInsert
on
Orders
after insert
as
begin
Declare @orderid int
select @orderid = INSERTED.orderid
from INSERTED
INSERT INTO orderlogs
values(@orderid,' ADDED')
END

insert into orders (orderid,customer_customerid,employee_employeeid) values('88','544','443')
insert into orders (orderid,customer_customerid,employee_employeeid) values('03','494','889')

select * from orderlogs
```

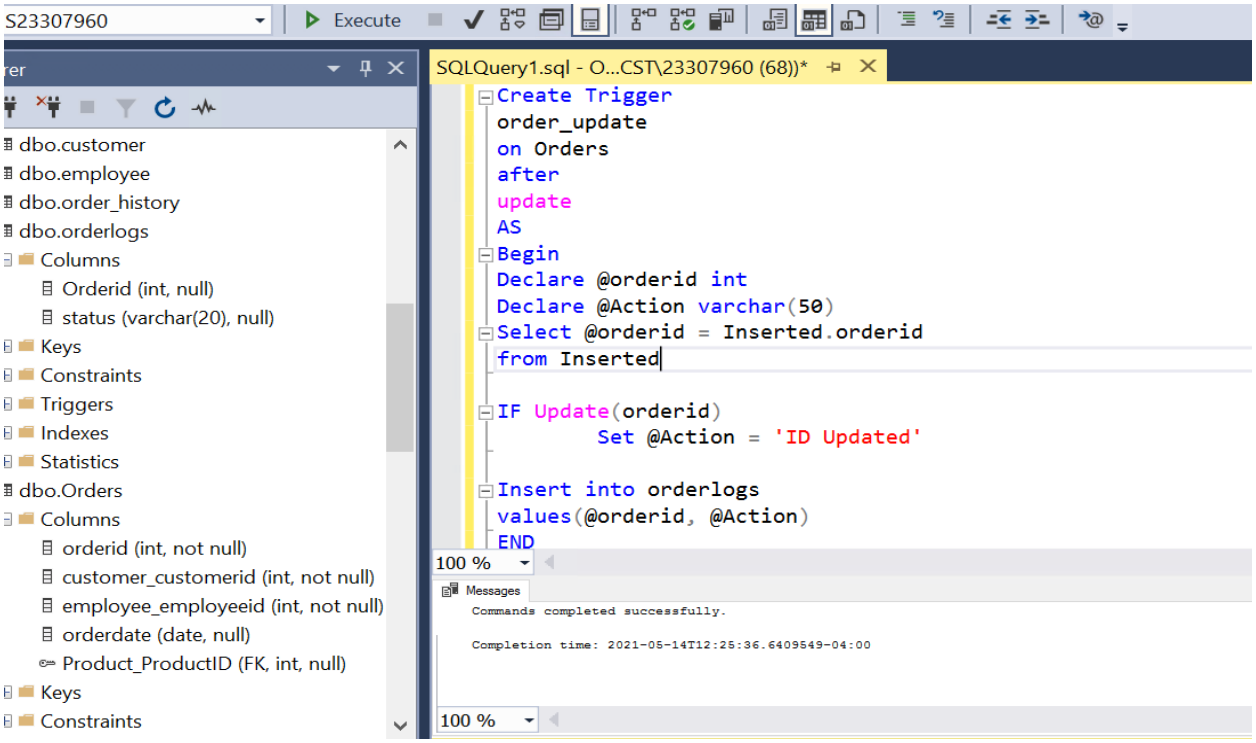
100 %

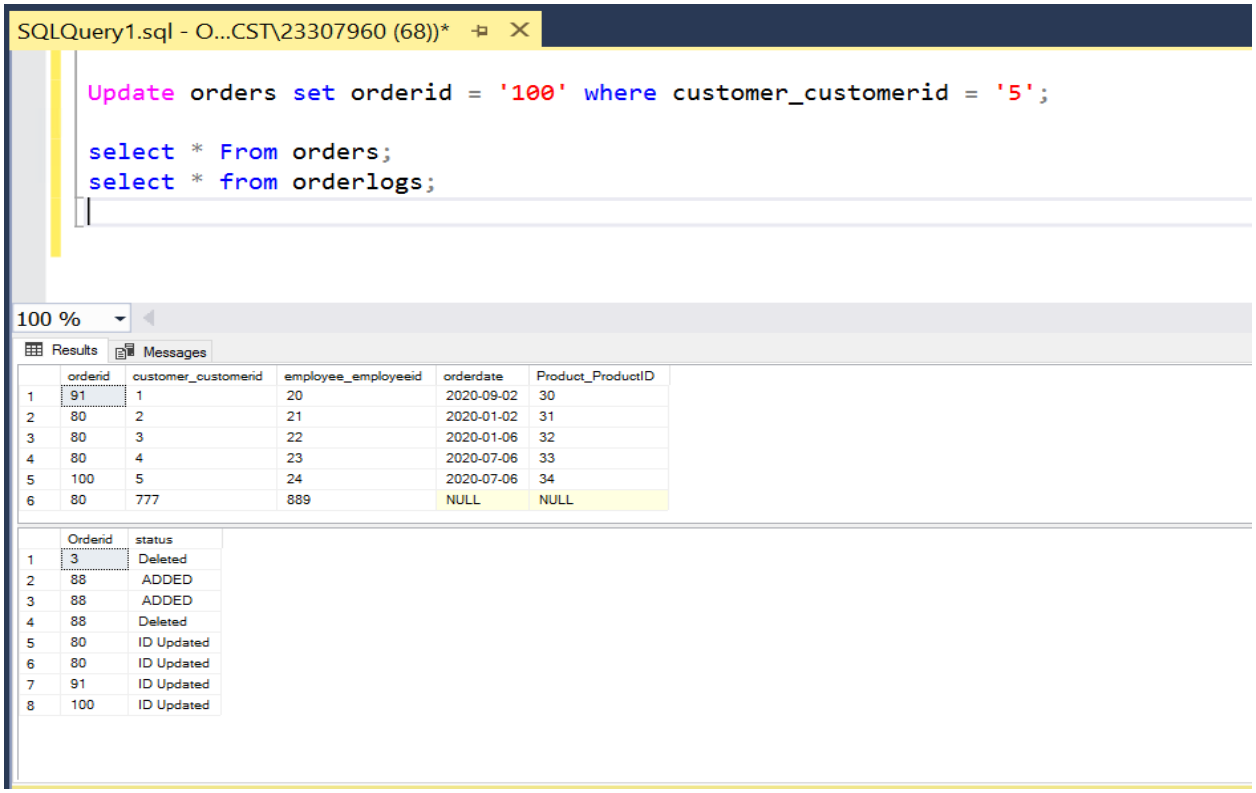
Results Messages

	Orderid	status
1	99	ADDED
2	88	ADDED
3	88	ADDED
4	3	ADDED

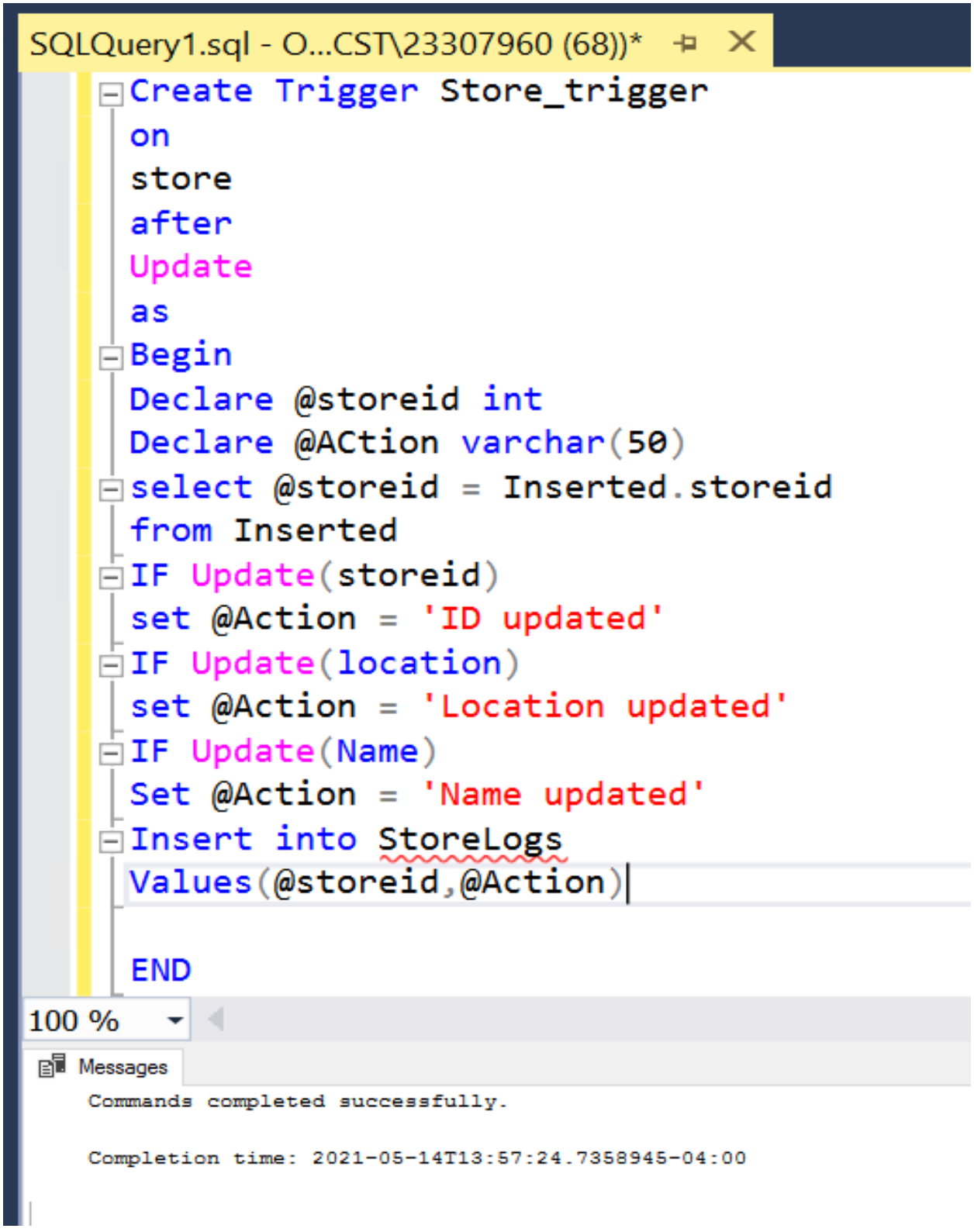
- At least two (2) of the triggers must be for update.

1st Update





2nd Update


SQLQuery1.sql - O...CST\23307960 (68))*
Create Trigger Store_trigger
on
store
after
Update
as
Begin
Declare @storeid int
Declare @ACtion varchar(50)
select @storeid = Inserted.storeid
from Inserted
IF Update(storeid)
set @Action = 'ID updated'
IF Update(location)
set @Action = 'Location updated'
IF Update(Name)
Set @Action = 'Name updated'
Insert into StoreLogs
Values(@storeid,@Action)
END
100 %
Messages
Commands completed successfully.
Completion time: 2021-05-14T13:57:24.7358945-04:00

```
Update store set  location = 'richmond' where Name = 'iut';
Update store set Name = 'jerry' where location  = 'rowaway';
Select * from store;
select * from StoreLogs;
```

100 %

Results | Messages

	storeid	location	Name
1	10	nasvill	xyz
2	11	highroad	abc
3	12	highland	xzy
4	13	rowaway	jerry
5	14	richmond	iut
6	15	hilltop	cvb

	storeid	status
1	14	Location updated
2	13	Name updated
3	14	Location updated
4	13	Name updated

- At least one (1) of the triggers must be for delete/insert/update.

SQLQuery1.sql - O...CST\23307960 (68))*

```
Create Trigger newTrigger
ON
Orders
After insert, update, delete
as
Begin
 Select * from inserted
 Select  *from Deleted
 End

 Update orders
 set orderid = '88'
 where customer_customerid = '67';
```

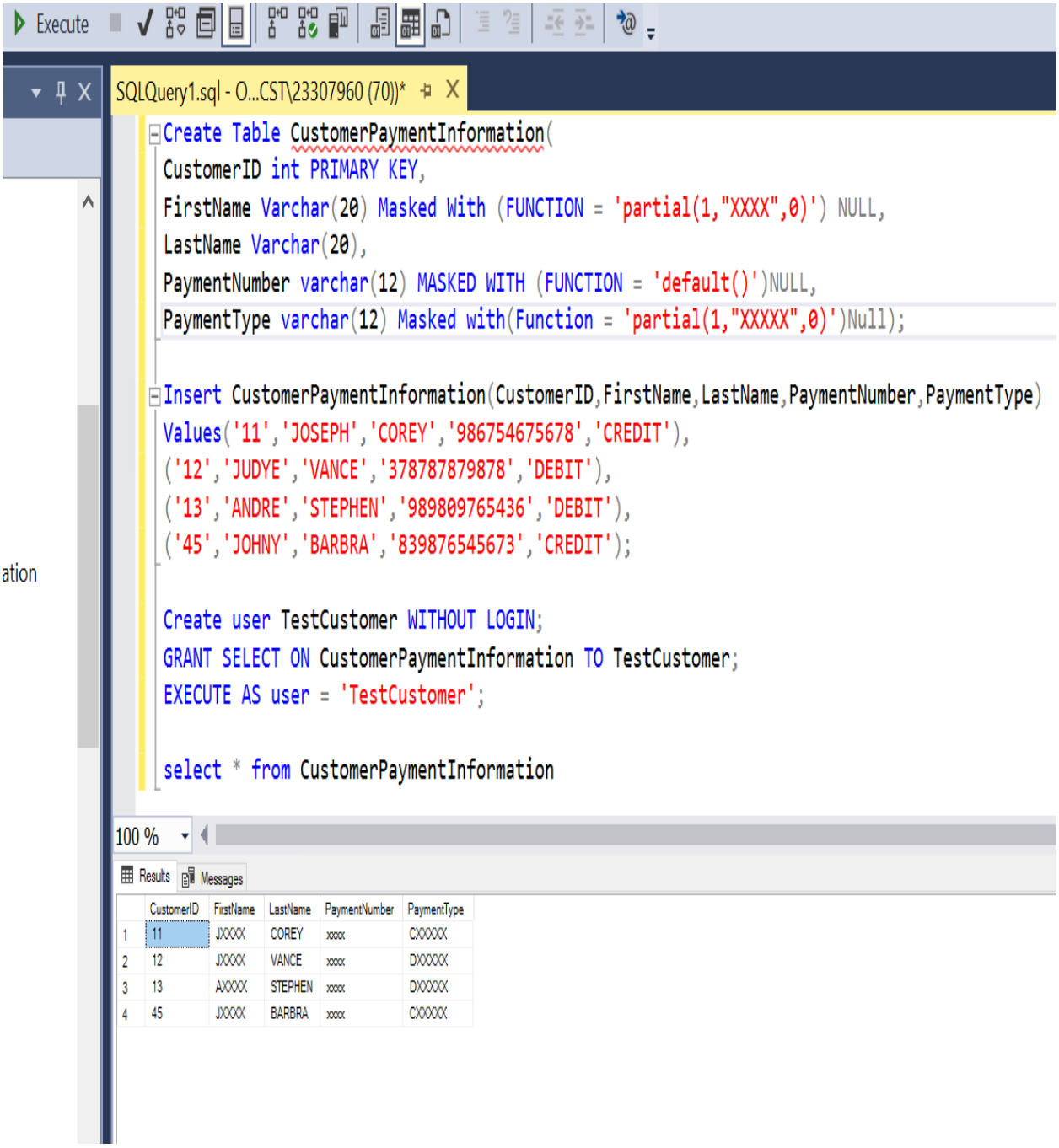
100 %

Results | Messages

	orderid	customer_customerid	employee_employeeid	orderdate	Product_ProductID
1	88	67	66	2012-02-01	NULL

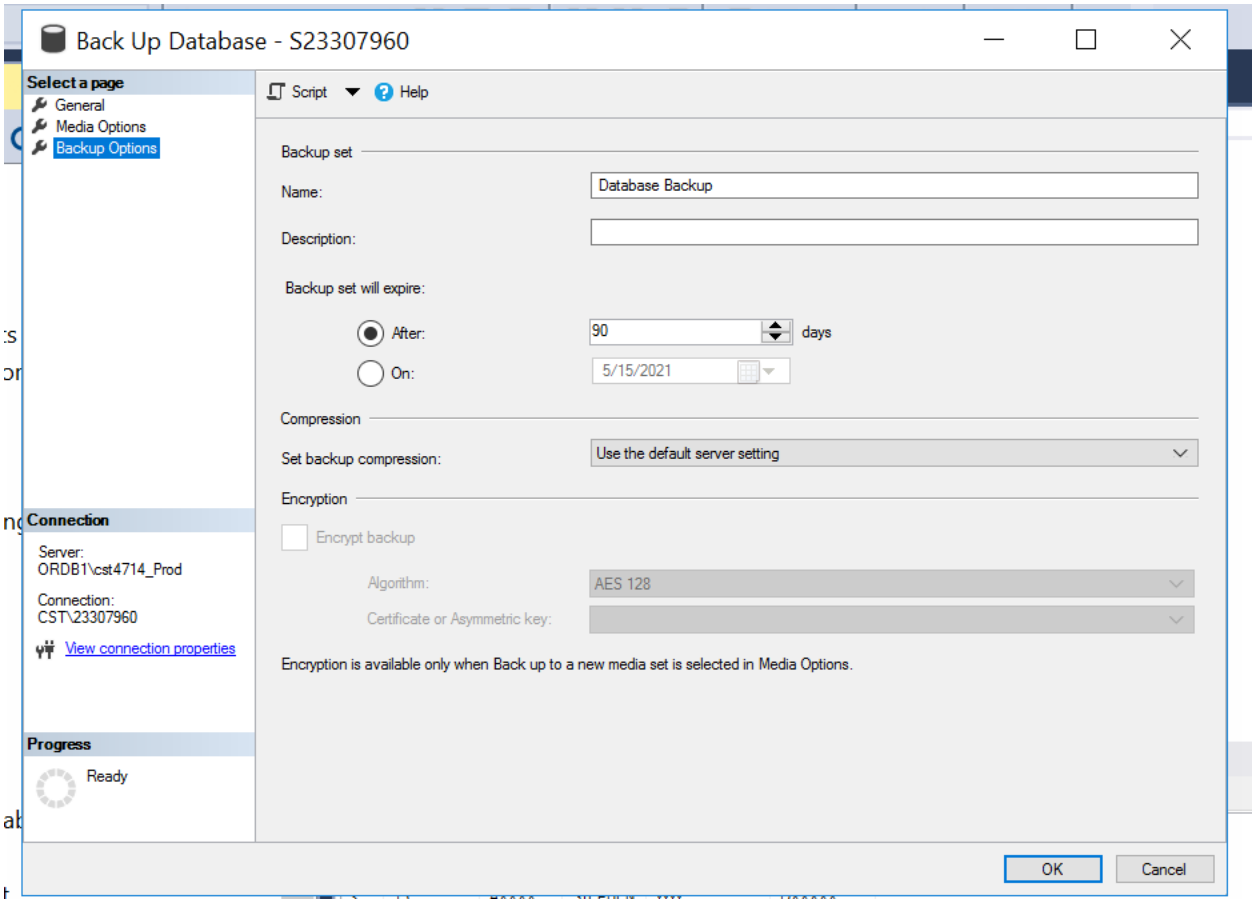
	orderid	customer_customerid	employee_employeeid	orderdate	Product_ProductID
1	91	67	66	2012-02-01	NULL

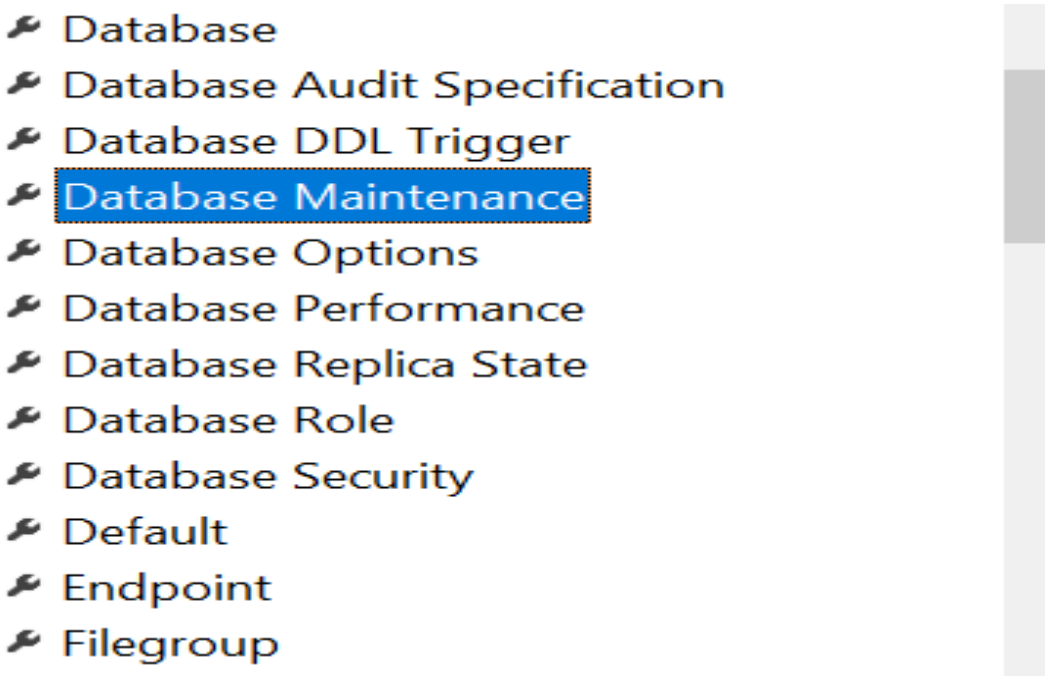
Data security: (at least three fields) client's sensitive data/information such as email/phone number/ credit cards etc. are expected to be secure and mask when displaying.



Data retention policy: data is kept indefinitely. *There are numerous ways to resolved this issue, find one and implement*.

--> In Order to resolve the Problem of Data Retention Policy we need to maintain the backups of our data because the data keeps on piling up. We need to auto clean up and delete backups older than a certain period like a Month. With the Maintenance plan we can deletes backups data, and we can specify the amount of time the data should be deleted in.




Database
Database Audit Specification
Database DDL Trigger
Database Maintenance
Database Options
Database Performance
Database Replica State
Database Role
Database Security
Default
Endpoint
Filegroup