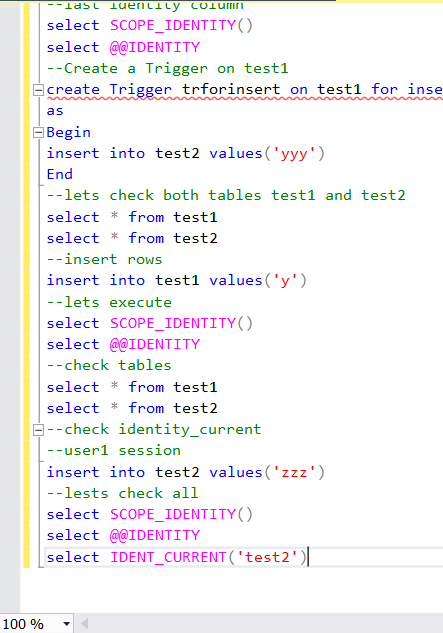
**SQL SERVER Advanced 4**



select \* from Person

-- add check constraints

alter table person

add constraint ck\_person\_age check(age>0 and age<150)

-- Drop contraints

alter table person

drop constraint ck\_person\_age

--Task 1 SQL advanced --------------------

create table tblperson(id int primary key,name nvarchar(50),age int null)

alter table tblperson

add constraint ck\_tblperson\_age check(age>0 and age<70)

insert into tblperson values(1,'ajman',34)

insert into tblperson values(2,'ajmal',35),(3,'waqar',36),(4,'zubair',45)

--check id primary key

insert into tblperson values(2,'ajmal',35)

--check age greater than 70

insert into tblperson values(5,'ajmal',75)

-- task2------------------------------------------sql server part7-------------

create table person2(personid int identity(1,1),name nvarchar(50))

insert into person2 values ('ajman'),('kalim'),('noman'),

('ajman2'),('kalim2'),('noman2'),('ajman3'),('kalim3'),('noman3'),

('ajman4'),('kalim4'),('noman4'),('ajman5'),('kalim5'),('noman5')

--check identity personid--

insert into person2 values (16,'ajman6')

--check table insertion

select \* from person2

--set identity insert on

set identity\_insert person2 on

insert into person2(personid,name) values (16,'ajman6')

--check row

select \* from person2

--check again without personid insert

insert into person2(name) values ('ajman7')

--off insert personid

set identity\_insert person2 off

--again check insert without personid

insert into person2(name) values ('ajman7')

select \* from person2

--create table and idenetity

create table Person3

(

Id int Identity(1,5),

Name varchar(20)

)

--insert values to person 3

insert into person3 values ('ajman1'),('ajman2')

,('kalim2'),

('noman2'),('ajman3'),('kalim3'),('noman3'),

('ajman4'),('kalim4'),('noman4'),

('ajman5'),('kalim5'),('noman5')

--check table person3

select \* from person3

--drop column

alter table person3

drop column name

--add identity again

DBCC CHECKIDENT('Person3', RESEED)

--add column

alter table person3

add id int identity(5,1)

--insert again

insert into person3 values ('ajman1'),('ajman2')

,('kalim2'),

('noman2'),('ajman3'),('kalim3'),('noman3'),

('ajman4'),('kalim4'),('noman4'),

('ajman5'),('kalim5'),('noman5')

--check table person3

select \* from person3

--------------------------SQL Part 8 tasks -----------

create table test1

(

id int identity(1,1),

value nvarchar(20)

)

create table test2

(

id int identity(1,1),

value nvarchar(20)

)

--insert values

insert into test1 values('x')

insert into test1 values('x')

insert into test1 values('x')

insert into test1 values('x')

select \* from test1

--last identity column

select SCOPE\_IDENTITY()

select @@IDENTITY

--Create a Trigger on test1

create Trigger trforinsert on test1 for insert

as

Begin

insert into test2 values('yyy')

End

--lets check both tables test1 and test2

select \* from test1

select \* from test2

--insert rows

insert into test1 values('y')

--lets execute

select SCOPE\_IDENTITY()

select @@IDENTITY

--check tables

select \* from test1

select \* from test2

--check identity\_current

--user1 session

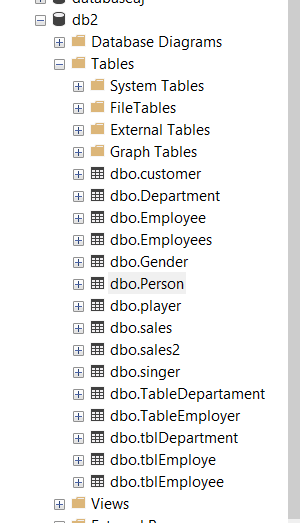
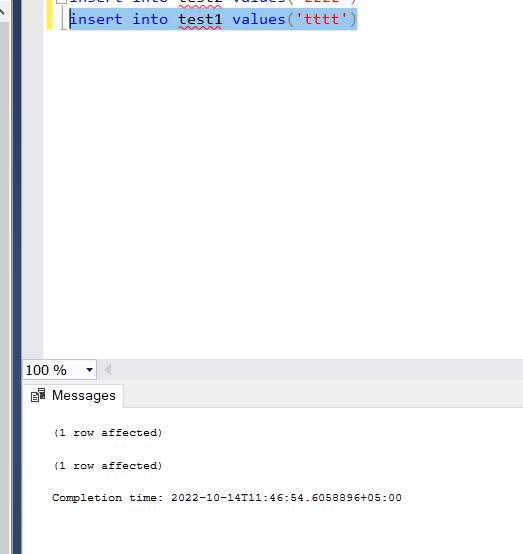
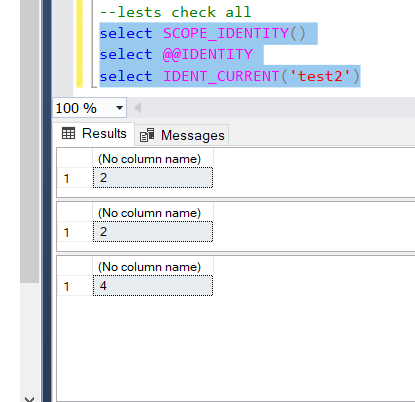
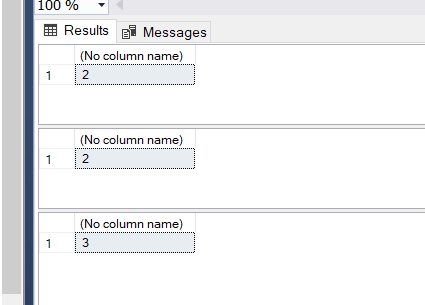
insert into test2 values('zzz')

--lests check all

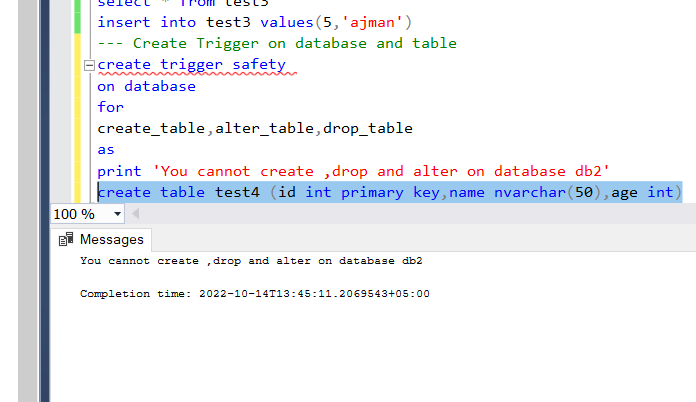
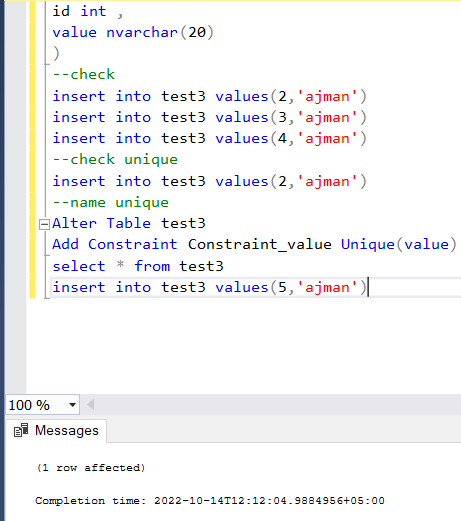
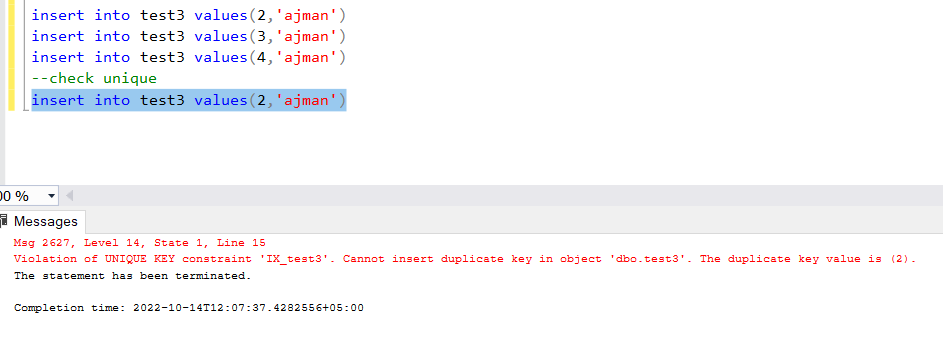
select SCOPE\_IDENTITY()

select @@IDENTITY

select IDENT\_CURRENT('test2')



Alter Table Table\_Name  
Add Constraint Constraint\_Name Unique(Column\_Name)



create trigger safety

on database

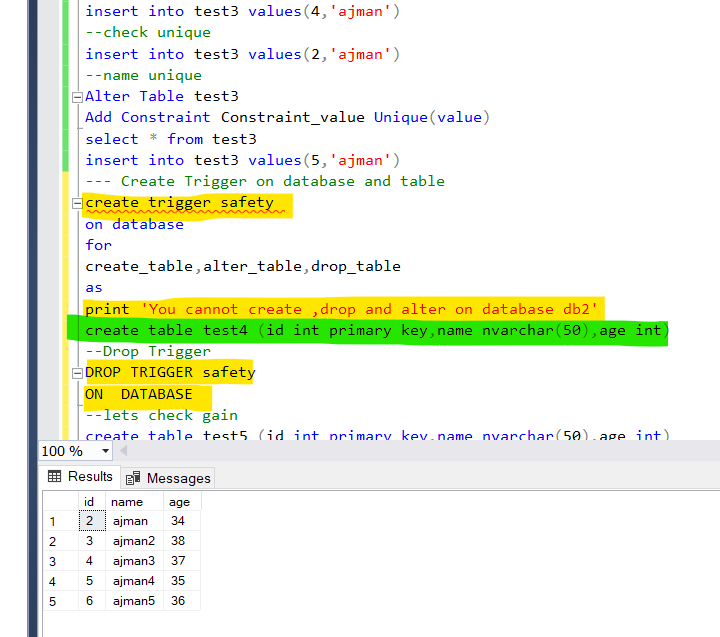
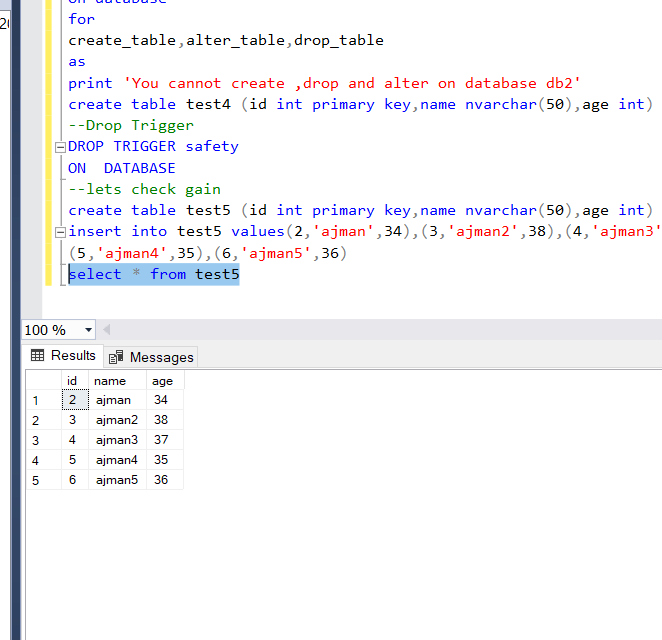
for

create\_table,alter\_table,drop\_table

as

print 'You cannot create ,drop and alter on database db2'

create table test4 (id int primary key,name nvarchar(50),age int)



-- Trigger on an INSERT, UPDATE, or DELETE statement to a table or view (DML Trigger)

DROP TRIGGER [ IF EXISTS ] [schema\_name.]trigger\_name [ ,...n ] [ ; ]

-- Trigger on a CREATE, ALTER, DROP, GRANT, DENY, REVOKE or UPDATE statement (DDL Trigger)

DROP TRIGGER [ IF EXISTS ] trigger\_name [ ,...n ]

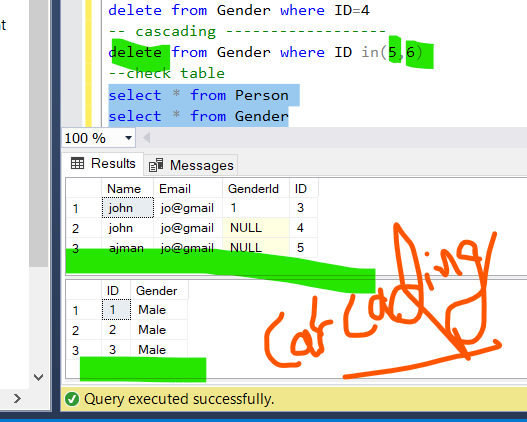
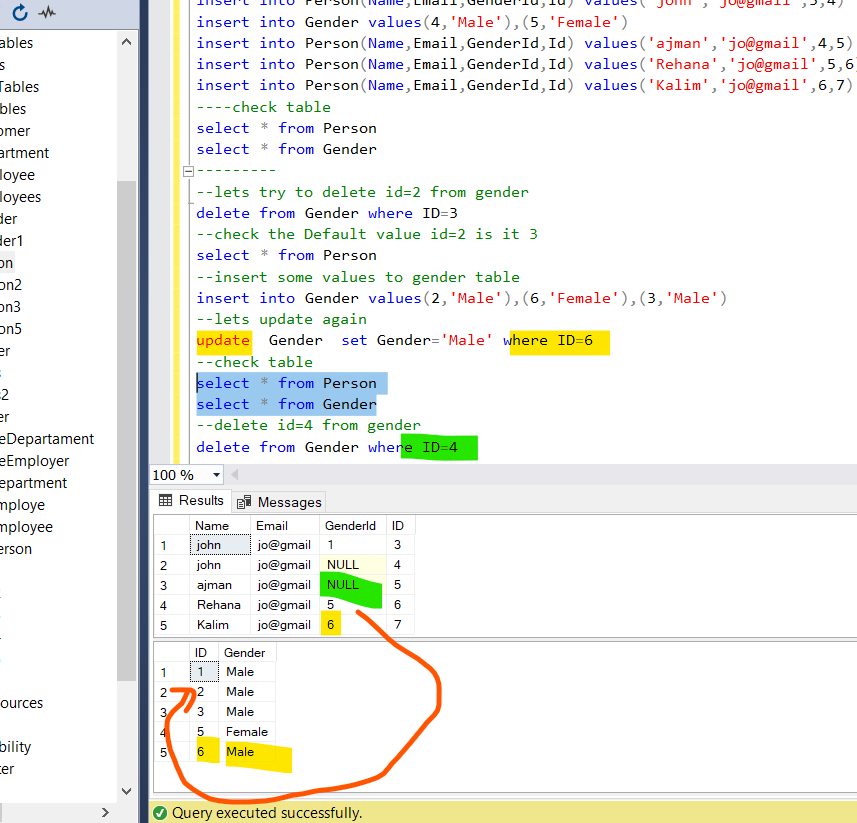
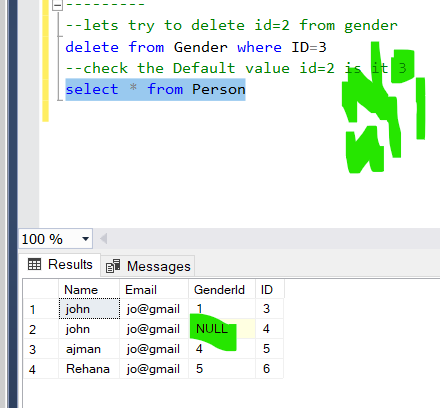
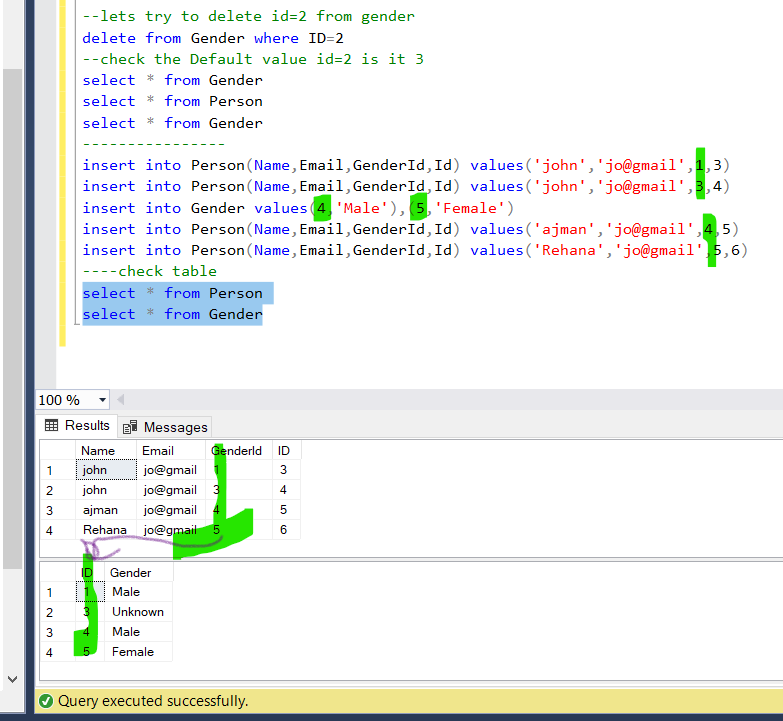
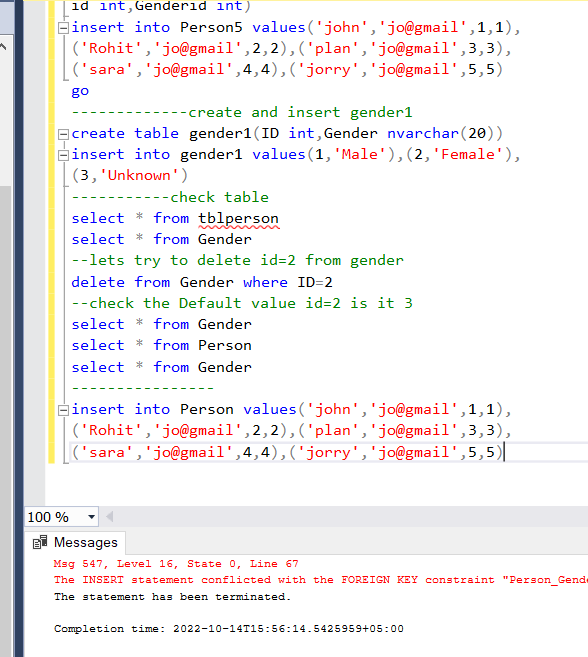
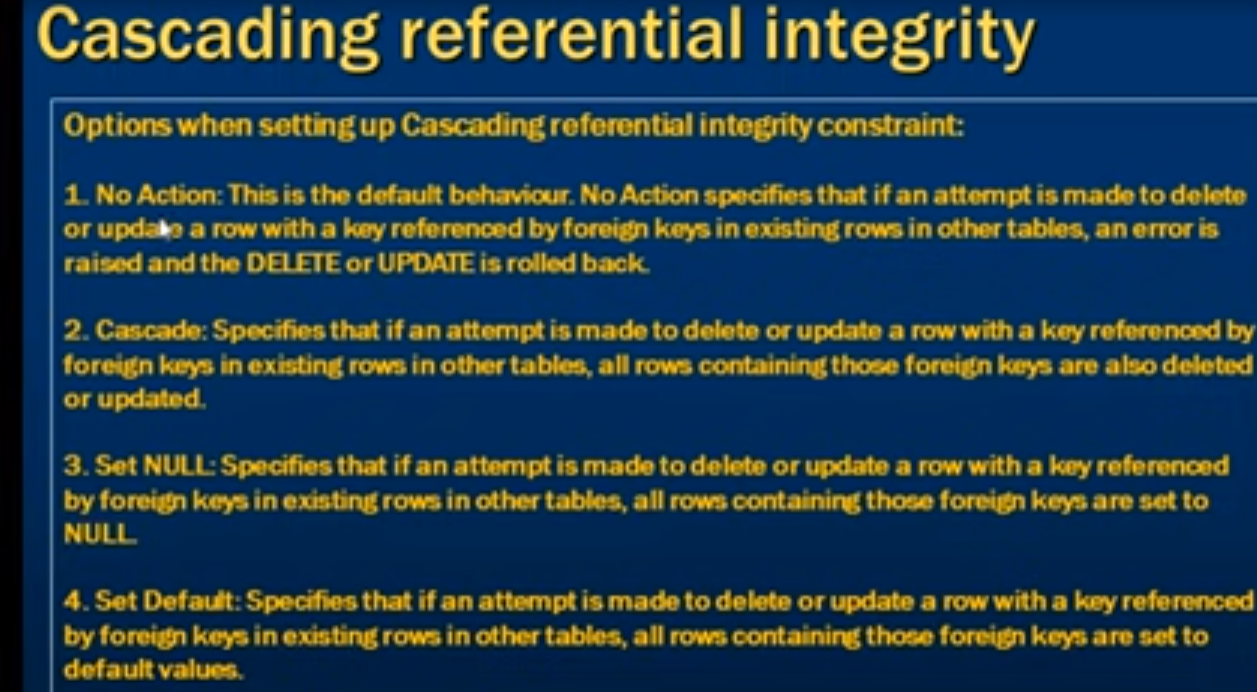
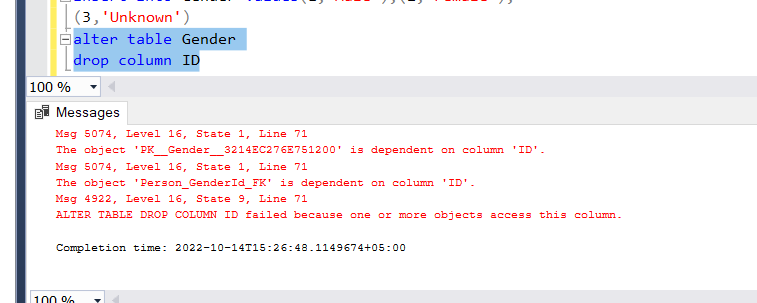
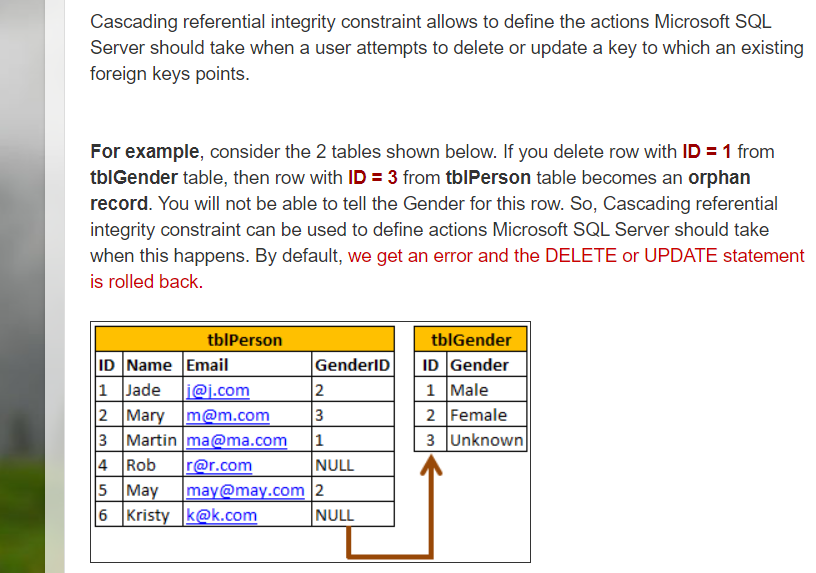
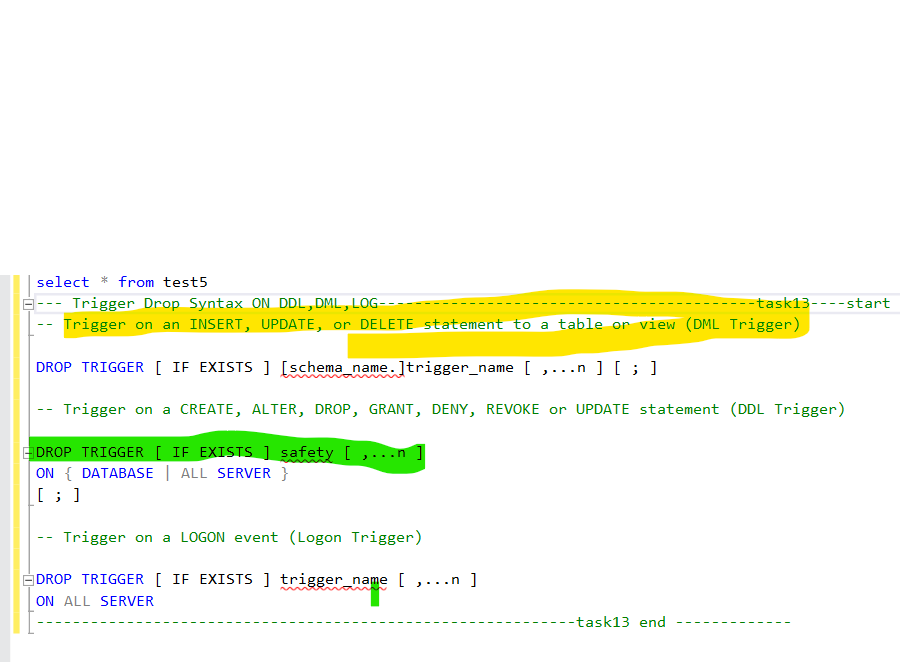
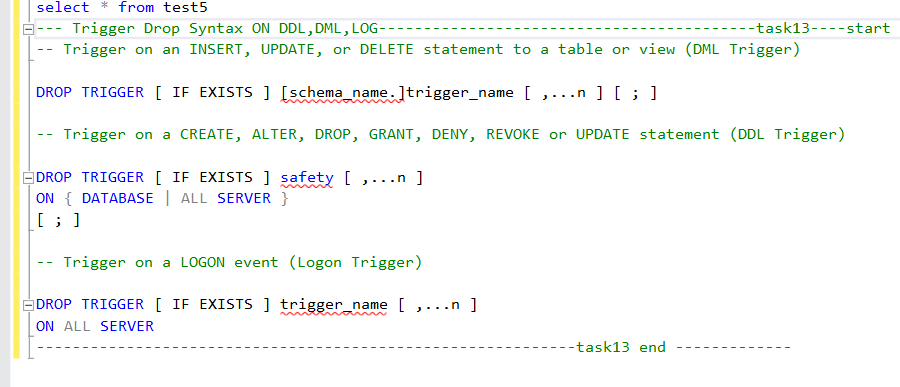
ON { DATABASE | ALL SERVER }

[ ; ]

-- Trigger on a LOGON event (Logon Trigger)

DROP TRIGGER [ IF EXISTS ] trigger\_name [ ,...n ]

ON ALL SERVER



-----------------------------sql part3 and part4 ---------Tricky

select \* from Person

select \* from Gender

alter table Person

---drop age

drop column age

alter table Person

drop column id

-----add colunm

alter table Person

add ID int

---insert values------------------create table person 5

create table person5(name nvarchar(20),email nvarchar(20),

id int,Genderid int)

insert into Person5 values('john','jo@gmail',1,1),

('Rohit','jo@gmail',2,2),('plan','jo@gmail',3,3),

('sara','jo@gmail',4,4),('jorry','jo@gmail',5,5)

go

-------------create and insert gender1

create table gender1(ID int,Gender nvarchar(20))

insert into gender1 values(1,'Male'),(2,'Female'),

(3,'Unknown')

-----------check table

select \* from tblperson

select \* from Gender

--lets try to delete id=2 from gender

delete from Gender where ID=2

--check the Default value id=2 is it 3

select \* from Gender

select \* from Person

select \* from Gender

----------------

insert into Person(Name,Email,GenderId,Id) values('john','jo@gmail',1,3)

insert into Person(Name,Email,GenderId,Id) values('john','jo@gmail',3,4)

insert into Gender values(4,'Male'),(5,'Female')

insert into Person(Name,Email,GenderId,Id) values('ajman','jo@gmail',4,5)

insert into Person(Name,Email,GenderId,Id) values('Rehana','jo@gmail',5,6)

insert into Person(Name,Email,GenderId,Id) values('Kalim','jo@gmail',6,7)

----check table

select \* from Person

select \* from Gender

---------

--lets try to delete id=2 from gender

delete from Gender where ID=3

--check the Default value id=2 is it 3

select \* from Person

--insert some values to gender table

insert into Gender values(2,'Male'),(6,'Female'),(3,'Male')

--lets update again

update Gender set Gender='Male' where ID=6

--check table

select \* from Person

select \* from Gender

--delete id=4 from gender

delete from Gender where ID=4

-- cascading ------------------

delete from Gender where ID in(5,6)

--check table

select \* from Person

select \* from Gender

-----------------------------------------end of sql part3 tricky -----lenghty sessions

[\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\](\\\\)

insert into test2 values('zzzz')

insert into test1 values('tttt')

--SQL part 9 -------------

create table test3

(

id int ,

value nvarchar(20)

)

--check

insert into test3 values(2,'ajman')

insert into test3 values(3,'ajman')

insert into test3 values(4,'ajman')

--check unique

insert into test3 values(2,'ajman')

--name unique

Alter Table test3

Add Constraint Constraint\_value Unique(value)

select \* from test3

insert into test3 values(5,'ajman')

--- Create Trigger on database and table

create trigger safety

on database

for

create\_table,alter\_table,drop\_table

as

print 'You cannot create ,drop and alter on database db2'

create table test4 (id int primary key,name nvarchar(50),age int)

--Drop Trigger

DROP TRIGGER safety

ON DATABASE

--lets check gain

create table test5 (id int primary key,name nvarchar(50),age int)

insert into test5 values(2,'ajman',34),(3,'ajman2',38),(4,'ajman3',37),

(5,'ajman4',35),(6,'ajman5',36)

select \* from test5

////////////////////////////////////////////////////////////////

--- Trigger Drop Syntax ON DDL,DML,LOG------------------------------------------task13----start

-- Trigger on an INSERT, UPDATE, or DELETE statement to a table or view (DML Trigger)

DROP TRIGGER [ IF EXISTS ] [schema\_name.]trigger\_name [ ,...n ] [ ; ]

-- Trigger on a CREATE, ALTER, DROP, GRANT, DENY, REVOKE or UPDATE statement (DDL Trigger)

DROP TRIGGER [ IF EXISTS ] safety [ ,...n ]

ON { DATABASE | ALL SERVER }

[ ; ]

-- Trigger on a LOGON event (Logon Trigger)

DROP TRIGGER [ IF EXISTS ] trigger\_name [ ,...n ]

ON ALL SERVER

-------------------- ----------------------------------------task13 end -------------

-----------------------------sql part3 and part4 ---------Tricky

select \* from Person

select \* from Gender

alter table Person

---drop age

drop column age

alter table Person

drop column id

-----add colunm

alter table Person

add ID int

---insert values------------------create table person 5

create table person5(name nvarchar(20),email nvarchar(20),

id int,Genderid int)

insert into Person5 values('john','jo@gmail',1,1),

('Rohit','jo@gmail',2,2),('plan','jo@gmail',3,3),

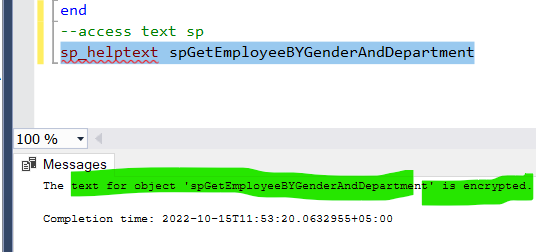
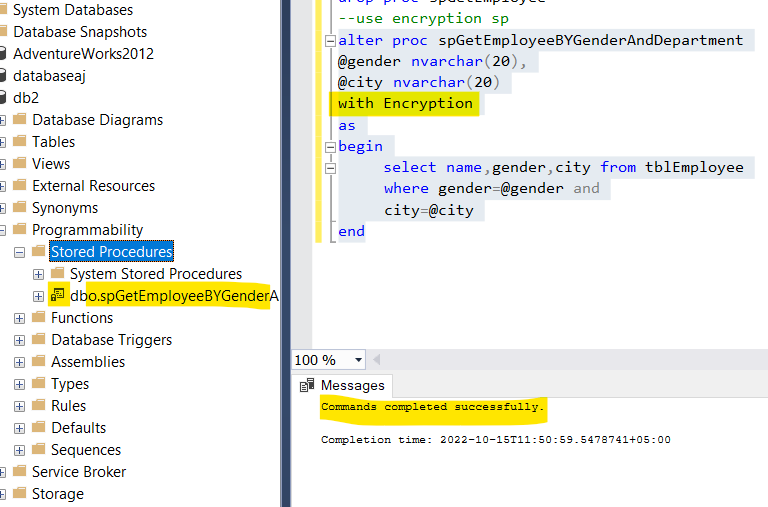
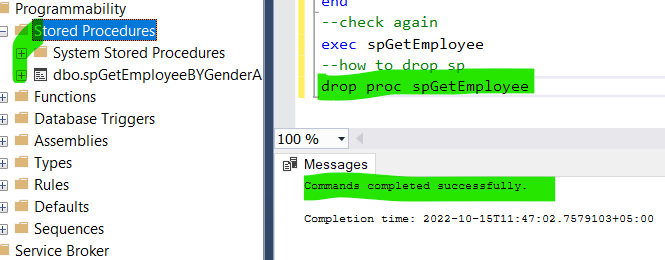
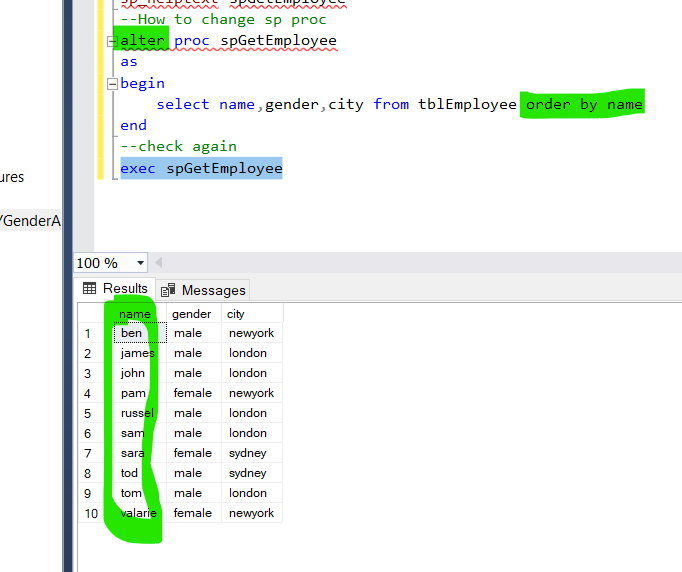
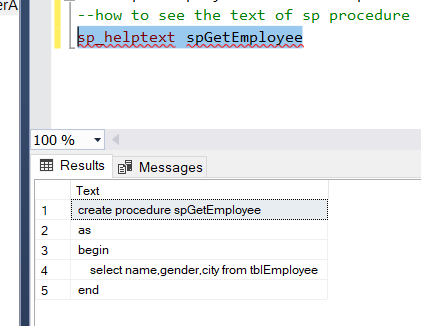
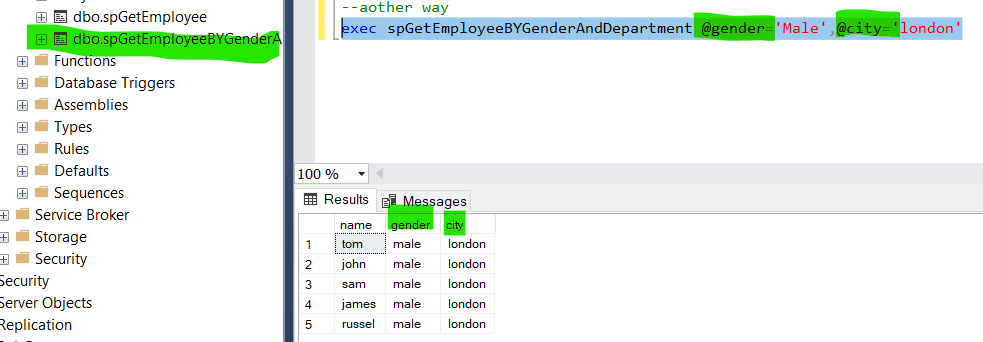
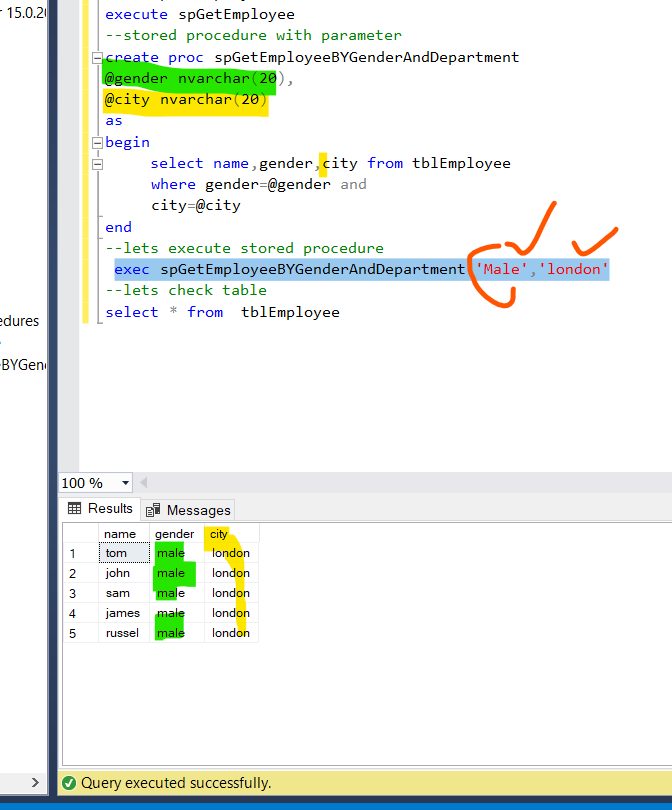
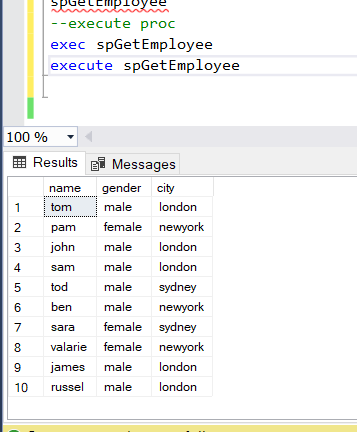
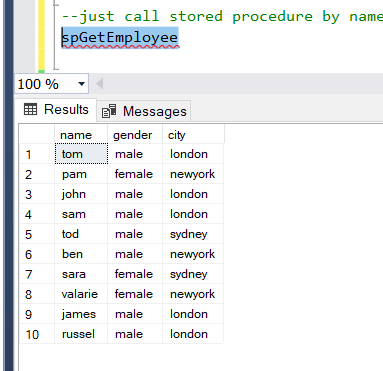
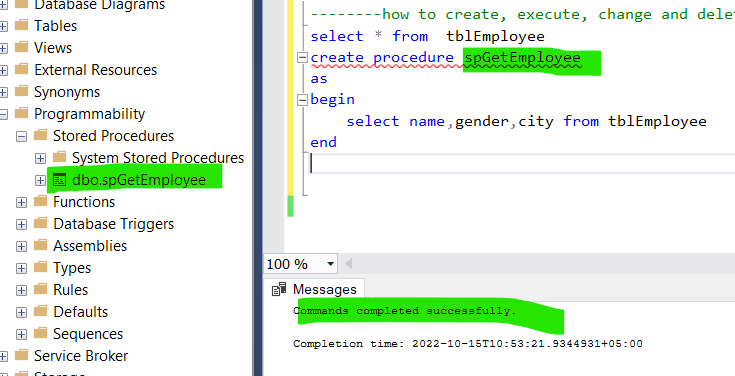
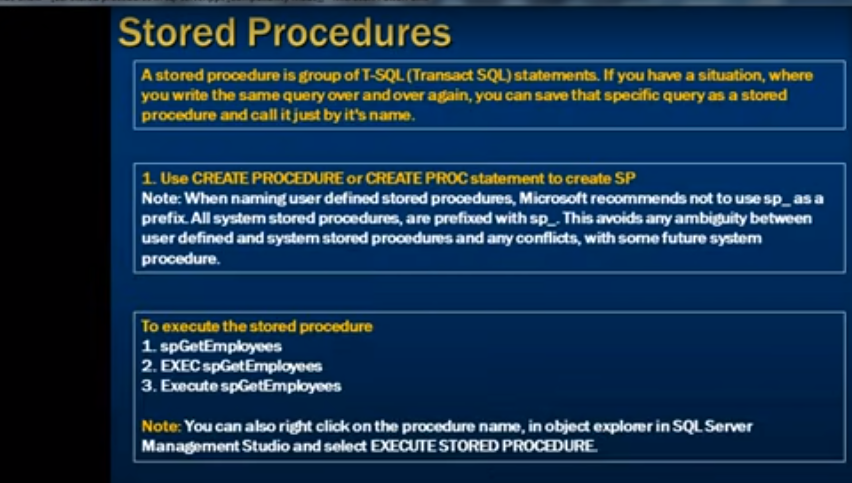
('sara','jo@gmail',4,4),('jorry','jo@gmail',5,5)

Go

///////////////////////////////////////////////////////////////

### Stored procedures - Part 18

A stored procedure is group of T-SQL (Transact SQL) statements. If you have a situation, where you write the same query over and over again, you can save that specific query as a stored procedure and call it just by it's name.



--------how to create, execute, change and delete stored procedures.-------------------------

select \* from tblEmployee

create procedure spGetEmployee

as

begin

select name,gender,city from tblEmployee

end

--just call stored procedure by name

spGetEmployee

--execute proc

exec spGetEmployee

execute spGetEmployee

--stored procedure with parameter

create proc spGetEmployeeBYGenderAndDepartment

@gender nvarchar(20),

@city nvarchar(20)

as

begin

select name,gender,city from tblEmployee

where gender=@gender and

city=@city

end

--lets execute stored procedure

exec spGetEmployeeBYGenderAndDepartment 'Male','london'

--lets check table

select \* from tblEmployee

--aother way

exec spGetEmployeeBYGenderAndDepartment @gender='Male',@city='london'

--how to see the text of sp procedure

sp\_helptext spGetEmployee

--How to change sp proc

alter proc spGetEmployee

as

begin

select name,gender,city from tblEmployee order by name

end

--check again

exec spGetEmployee

--how to drop sp

drop proc spGetEmployee

--use encryption sp

alter proc spGetEmployeeBYGenderAndDepartment

@gender nvarchar(20),

@city nvarchar(20)

with Encryption

as

begin

select name,gender,city from tblEmployee

where gender=@gender and

city=@city

end

--access text sp

sp\_helptext spGetEmployeeBYGenderAndDepartment