Assignment 17

-- 1. Re-create the Customersand Orderstables, enhancing their definition with all primary and foreign keys constraints

Create table Customers (CustomerId char(5) not null,

CompanyName varchar(40) not null, contactName char(30) null,

Address Varchar(60) null, City char(15) null, Phone char(24) null,

Fax Char(24) null);

Create table Orders (OrderId integer not null, customerId char(5) not null,

Orderdate datetime null, Shippeddate datetime null, Freight money null,

Shipname varchar(40) null, Shipaddress varchar(60) null,

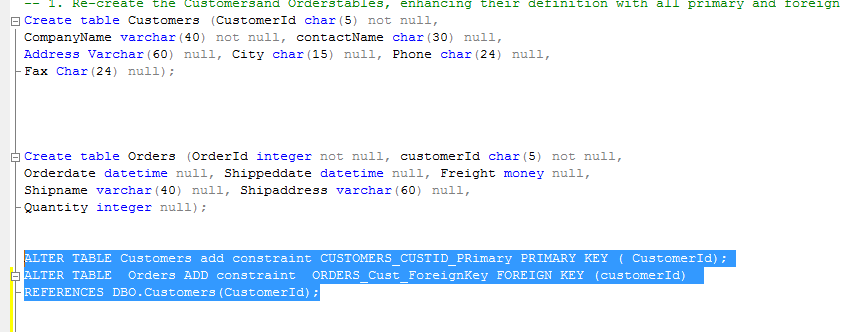
Quantity integer null);

ALTER TABLE Customers add constraint CUSTOMERS\_CUSTID\_PRimary PRIMARY KEY ( CustomerId);

ALTER TABLE Orders ADD constraint ORDERS\_Cust\_ForeignKey FOREIGN KEY (customerId)

REFERENCES DBO.Customers(CustomerId);

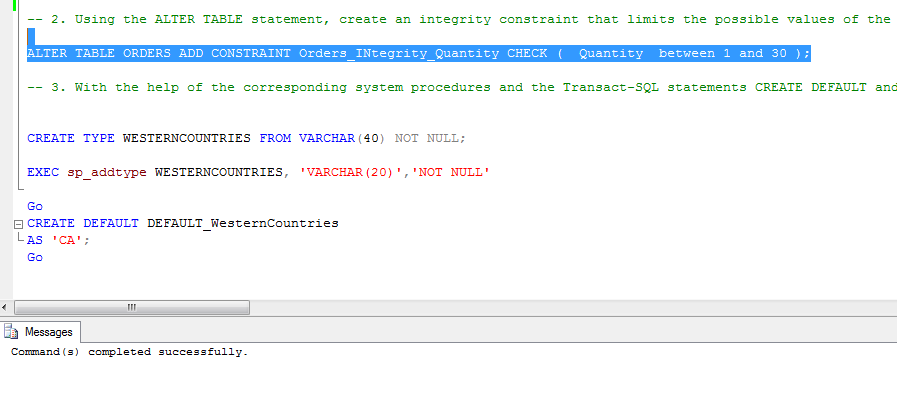
Output:



-- 2. Using the ALTER TABLE statement, create an integrity constraint that limits the possible values of the quantity column in the Orderstable to values between 1 and 30

ALTER TABLE ORDERS ADD CONSTRAINT Orders\_INtegrity\_Quantity CHECK ( Quantity between 1 and 30 );

Output:



-- 3. With the help of the corresponding system procedures and the Transact-SQL statements CREATE DEFAULT and CREATE RULE, create the alias data type “Western Countries”. The possible values for the new data type are CA(for California), WA( for Washington), OR( for Oregon), and NM( for New Mexico). The default value is CA. Finally, create a table called Regionswith the columnsCityand Countryusing the new data type for the later

CREATE TYPE WESTERNCOUNTRIES FROM VARCHAR(40) NOT NULL;

Go

CREATE DEFAULT DEFAULT\_WesternCountries

AS 'CA';

Go

GO

CREATE RULE rule\_westernCountries AS @WESTERNCOUNTRIES IN ('CA','WA','OR','NM')

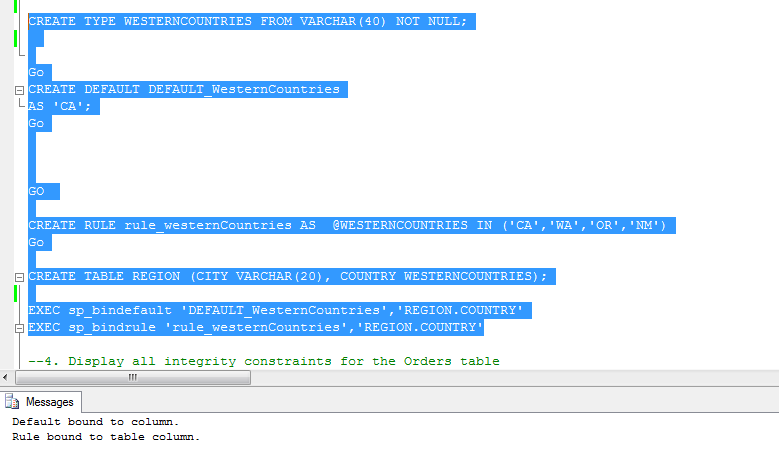
Go

CREATE TABLE REGION (CITY VARCHAR(20), COUNTRY WESTERNCOUNTRIES);

EXEC sp\_bindefault 'DEFAULT\_WesternCountries','REGION.COUNTRY'

EXEC sp\_bindrule 'rule\_westernCountries','REGION.COUNTRY'

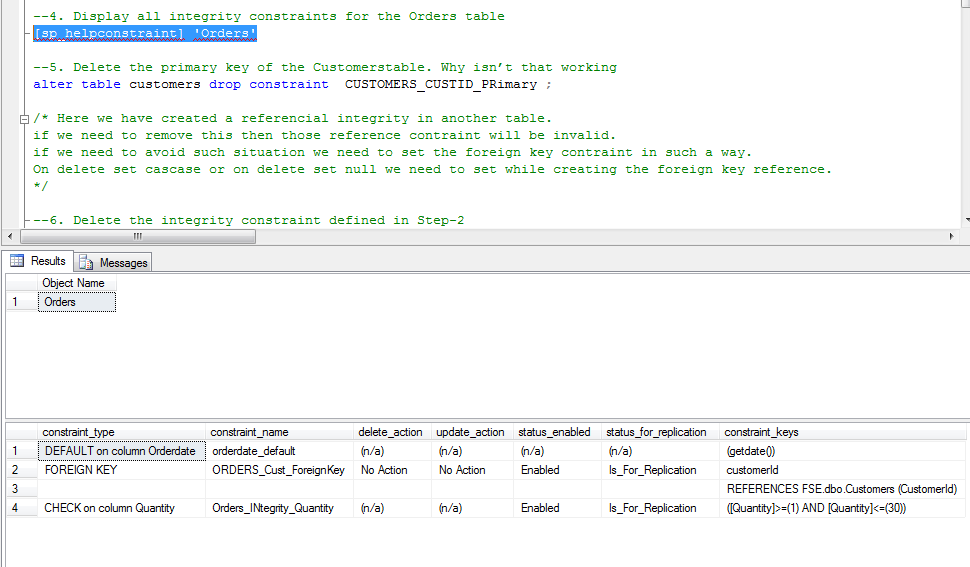
Output:



--4. Display all integrity constraints for the Orders table

[sp\_helpconstraint] 'Orders'

Output:



--5. Delete the primary key of the Customerstable. Why isn’t that working

alter table customers drop constraint CUSTOMERS\_CUSTID\_PRimary ;

/\* Here we have created a referencial integrity in another table.

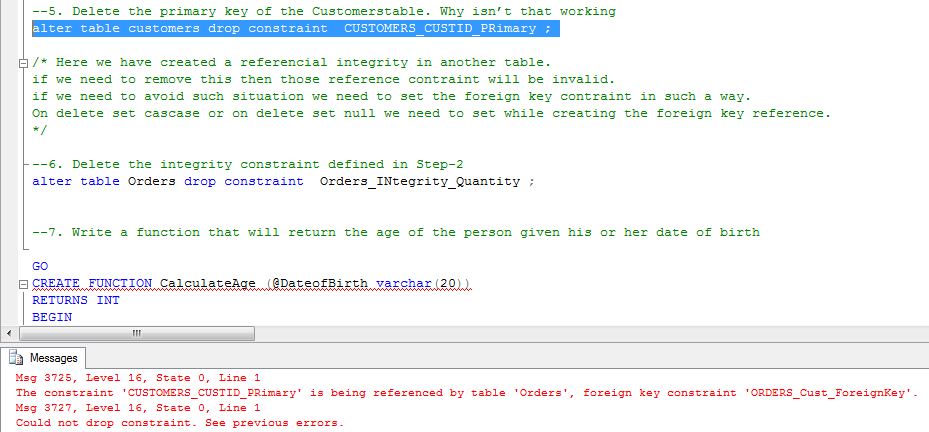
if we need to remove this then those reference contraint will be invalid.

if we need to avoid such situation we need to set the foreign key contraint in such a way.

On delete set cascase or on delete set null we need to set while creating the foreign key reference.

\*/

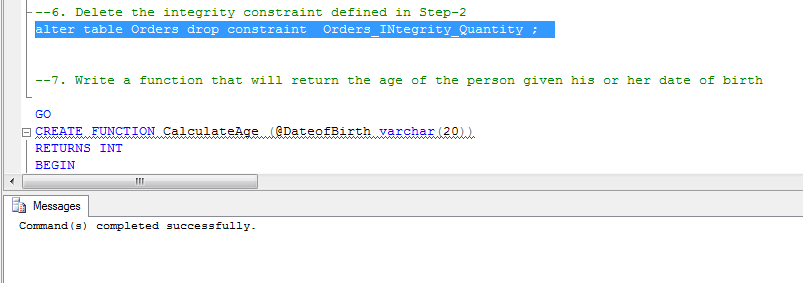
Output:



--6. Delete the integrity constraint defined in Step-2

alter table Orders drop constraint Orders\_INtegrity\_Quantity ;

Output:



--7. Write a function that will return the age of the person given his or her date of birth

GO

CREATE FUNCTION CalculateAge (@DateofBirth varchar(20))

RETURNS INT

BEGIN

DECLARE @RESULT INT;

IF(ISDATE(@DateofBirth) =1)

BEGIN

Select @RESULT= DATEDIFF(year,@DateofBirth,CONVERT(date, getdate()))

END

ELSE

BEGIN

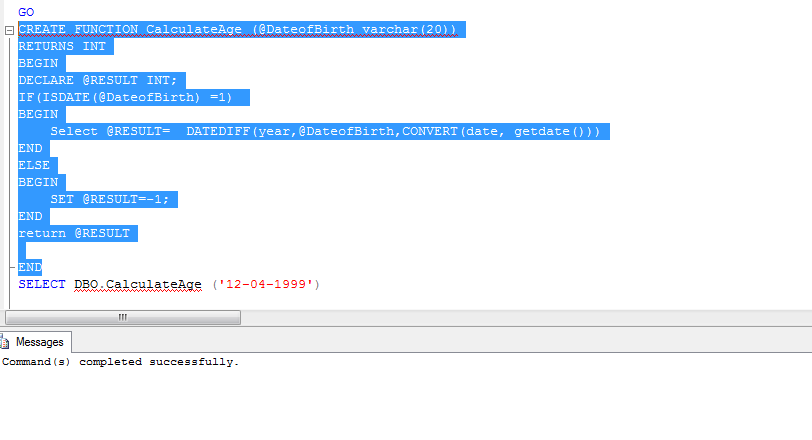
SET @RESULT=-1;

END

return @RESULT

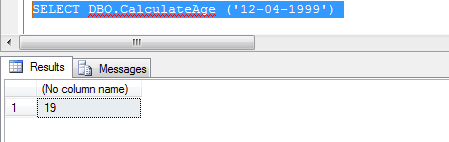
END

Output:



SELECT DBO.CalculateAge ('12-04-1999')

Output:



--8. Write a procedure that accepts name and data of birth of the student and inserts the data in the student table with the date computed. The SID should be largest sid in the table +1

--

CREATE TABLE STUDENT ([SID] INT, [NAME] VARCHAR(50) NOT NULL, [DOB] DATETIME NOT NULL);

Go

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE PROCEDURE INSERTSTUDENT

@Name VARCHAR(50) ,

@DOB VARCHAR(20)

AS

BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from

-- interfering with SELECT statements.

DECLARE @SID INTEGER;

DECLARE @DateofBirth DATETIME ;

SELECT @SID=MAX([SID]) from STUDENT;

IF(@SID iS NULL)

BEGIN

SET @SID=1;

END

ELSE

BEGIN

SET @SID=@SID +1;

END

IF(ISDATE(@DOB) =1 )

BEGIN

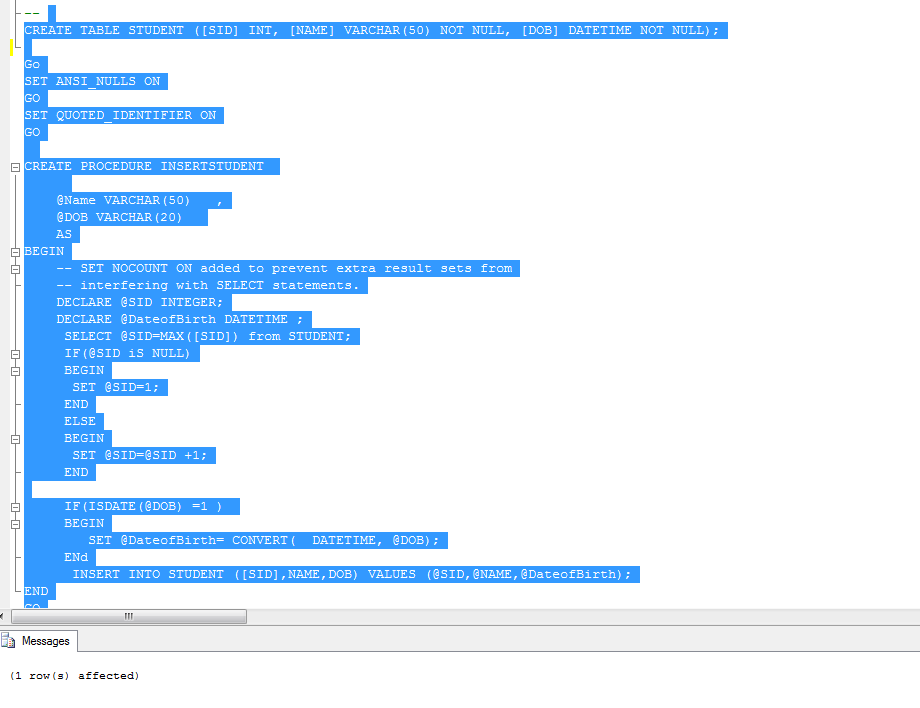
SET @DateofBirth= CONVERT( DATETIME, @DOB);

ENd

INSERT INTO STUDENT ([SID],NAME,DOB) VALUES (@SID,@NAME,@DateofBirth);

END

GO



-- inserting sample procedure call

EXEC INSERTSTUDENT 'Manoj','12-04-2008'

SELECT \* FROM STUDENT -- to validate the inserted record

Output:

