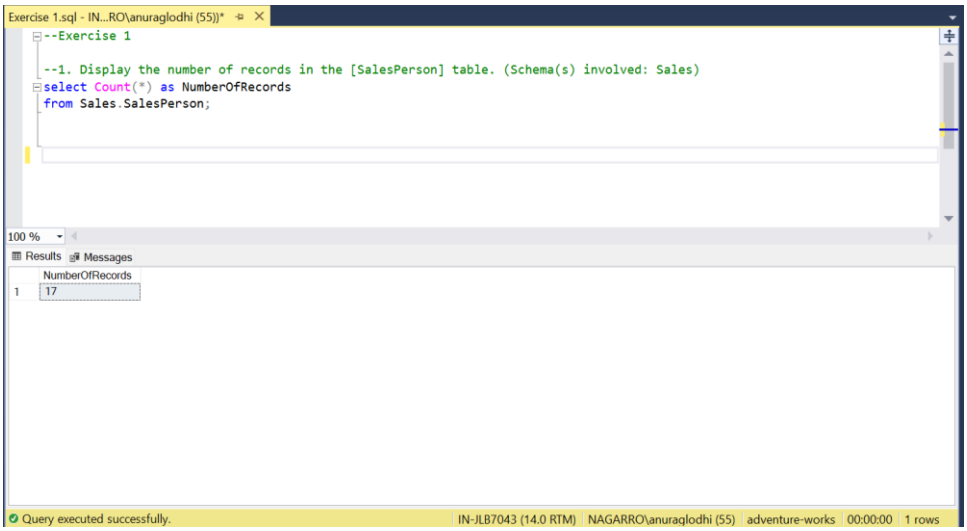
--Exercise 1

--1. Display the number of records in the [SalesPerson] table. (Schema(s) involved: Sales)

select Count(*) as NumberOfRecords

from Sales.SalesPerson;

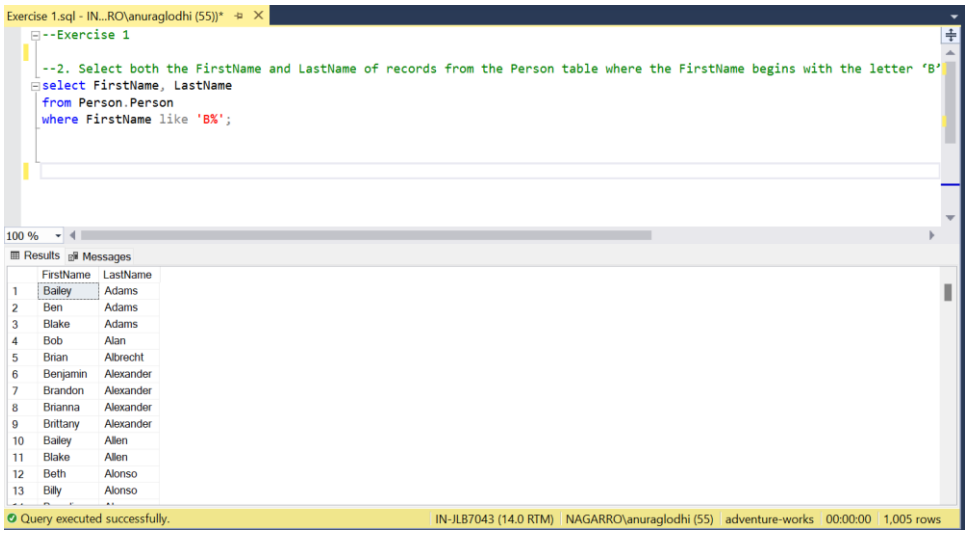


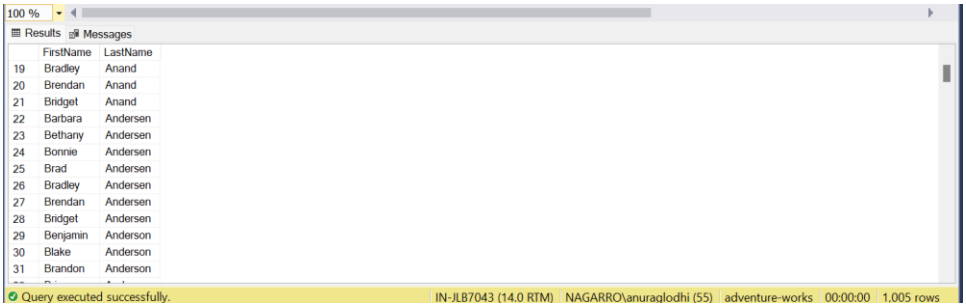
--2. Select both the FirstName and LastName of records from the Person table where the FirstName begins with the letter ‘B’. (Schema(s) involved: Person)

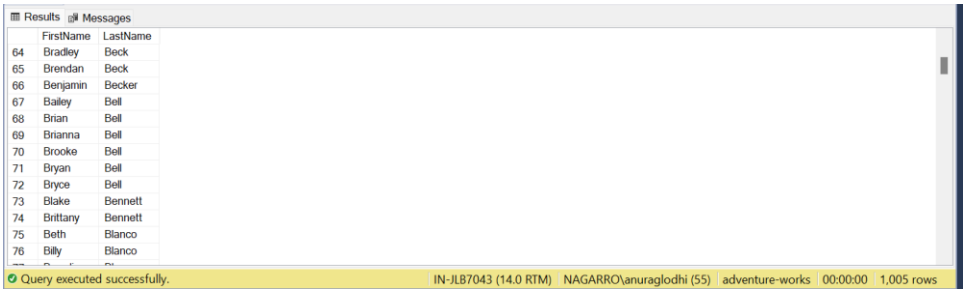
select FirstName, LastName

from Person.Person

where FirstName like 'B%';





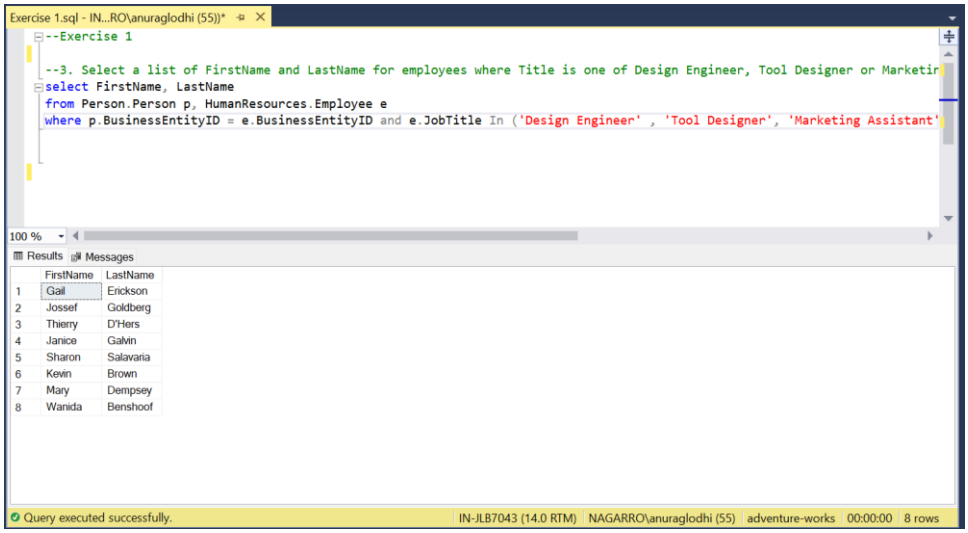


--3. Select a list of FirstName and LastName for employees where Title is one of Design Engineer, Tool Designer or Marketing Assistant. (Schema(s) involved: HumanResources, Person)

```
select FirstName, LastName

from Person.Person p, HumanResources.Employee e

where p.BusinessEntityID = e.BusinessEntityID and e.JobTitle In ('Design Engineer' , 'Tool Designer', 'Marketing Assistant');
```

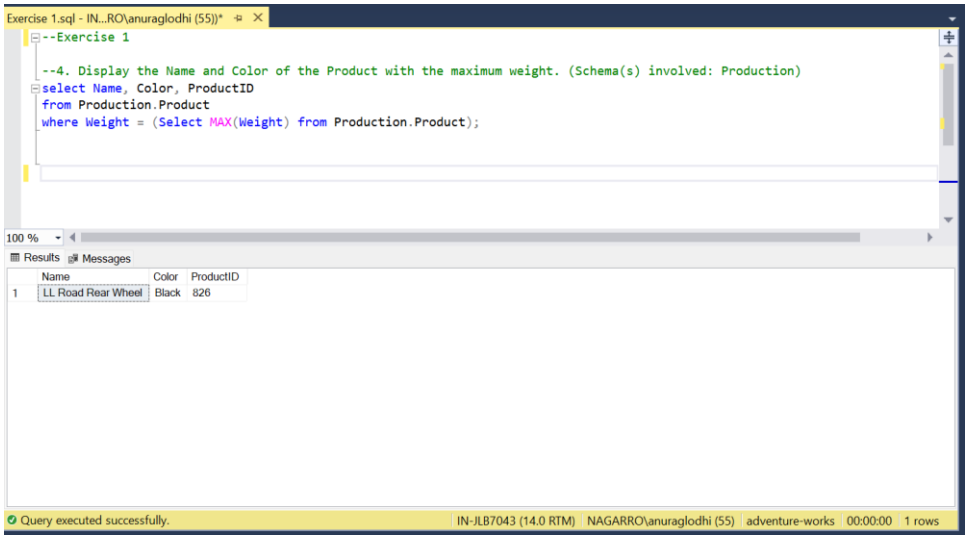


--4. Display the Name and Color of the Product with the maximum weight. (Schema(s) involved: Production)

select Name, Color, ProductID

from Production.Product

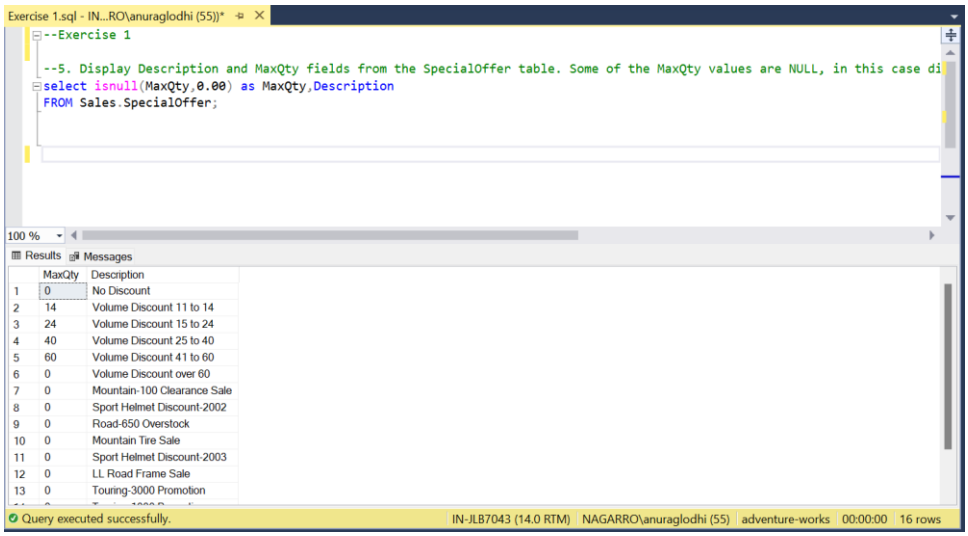
where Weight = (Select MAX(Weight) from Production.Product);



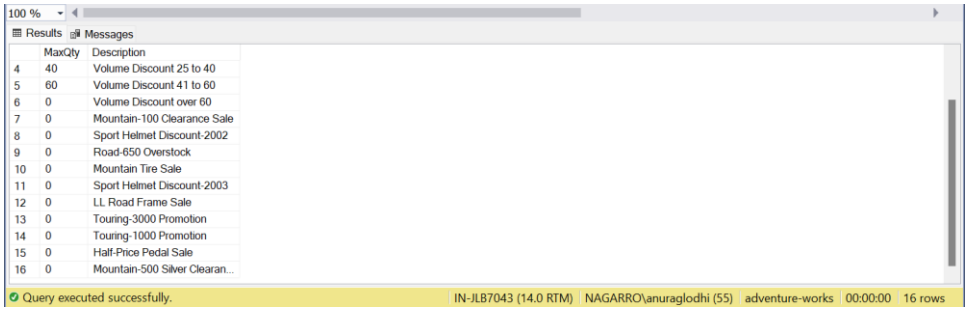
--5. Display Description and MaxQty fields from the SpecialOffer table. Some of the MaxQty values are NULL, in this case display the value 0.00 instead. (Schema(s) involved: Sales)

select isnull(MaxQty,0.00) as MaxQty,Description

FROM Sales.SpecialOffer;



	MaxQty	Description
1	0	No Discount
2	14	Volume Discount 11 to 14
3	24	Volume Discount 15 to 24
4	40	Volume Discount 25 to 40
5	60	Volume Discount 41 to 60
6	0	Volume Discount over 60
7	0	Mountain-100 Clearance Sale
8	0	Sport Helmet Discount-2002
9	0	Road-650 Overstock
10	0	Mountain Tire Sale
11	0	Sport Helmet Discount-2003
12	0	LL Road Frame Sale
13	0	Touring-3000 Promotion



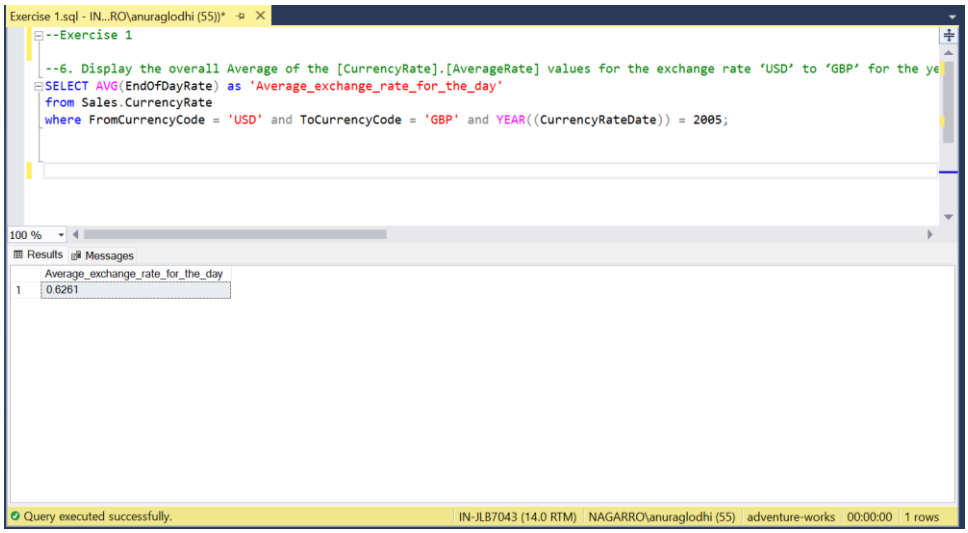
	MaxQty	Description
4	40	Volume Discount 25 to 40
5	60	Volume Discount 41 to 60
6	0	Volume Discount over 60
7	0	Mountain-100 Clearance Sale
8	0	Sport Helmet Discount-2002
9	0	Road-650 Overstock
10	0	Mountain Tire Sale
11	0	Sport Helmet Discount-2003
12	0	LL Road Frame Sale
13	0	Touring-3000 Promotion
14	0	Touring-1000 Promotion
15	0	Half-Price Pedal Sale
16	0	Mountain-500 Silver Clearan...

--6. Display the overall Average of the [CurrencyRate].[AverageRate] values for the exchange rate ‘USD’ to ‘GBP’ for the year 2005 i.e. FromCurrencyCode = ‘USD’ and ToCurrencyCode = ‘GBP’. Note: The field [CurrencyRate].[AverageRate] is defined as 'Average exchange rate for the day.' (Schema(s) involved: Sales)

SELECT AVG(EndOfDayRate) as 'Average_exchange_rate_for_the_day'

from Sales.CurrencyRate

where FromCurrencyCode = 'USD' and ToCurrencyCode = 'GBP' and YEAR((CurrencyRateDate)) = 2005;

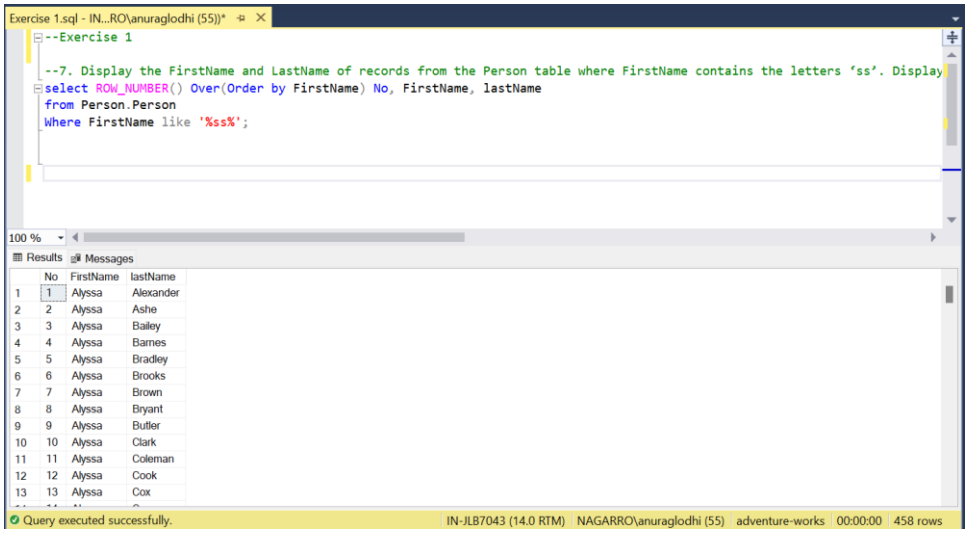


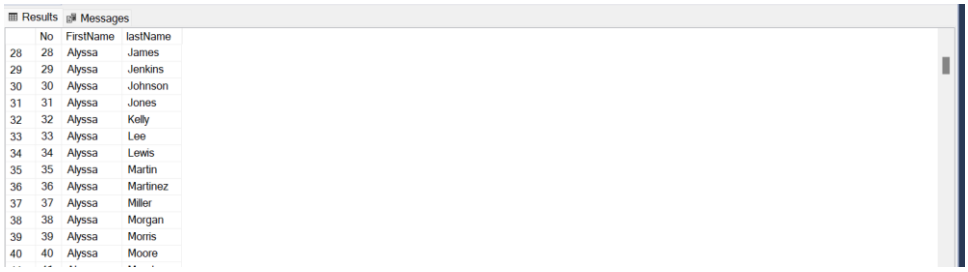
--7. Display the FirstName and LastName of records from the Person table where FirstName contains the letters ‘ss’. Display an additional column with sequential numbers for each row returned beginning at integer 1. (Schema(s) involved: Person)

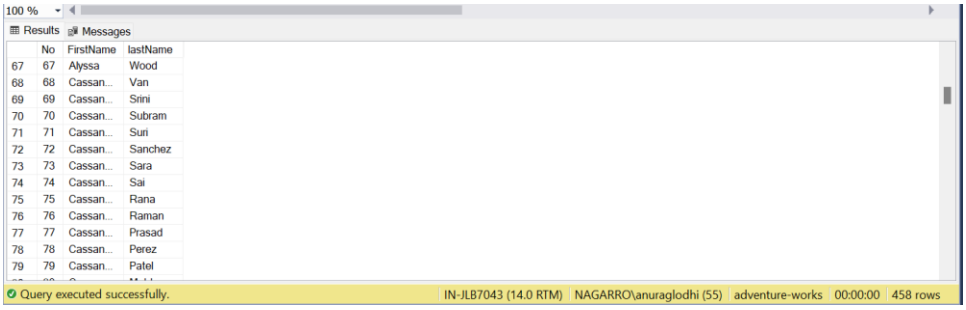
select ROW_NUMBER() Over(Order by FirstName) No, FirstName, lastName

from Person.Person

Where FirstName like '%ss%';







--8. Sales people receive various commission rates that belong to 1 of 4 bands. (Schema(s) involved: Sales)Display the [SalesPersonID] with an additional column entitled ‘Commission Band’ indicating the appropriate band as above

select BusinessEntityID as SalesPersonId,

case when CommissionPct = 0 then 'Band 0'

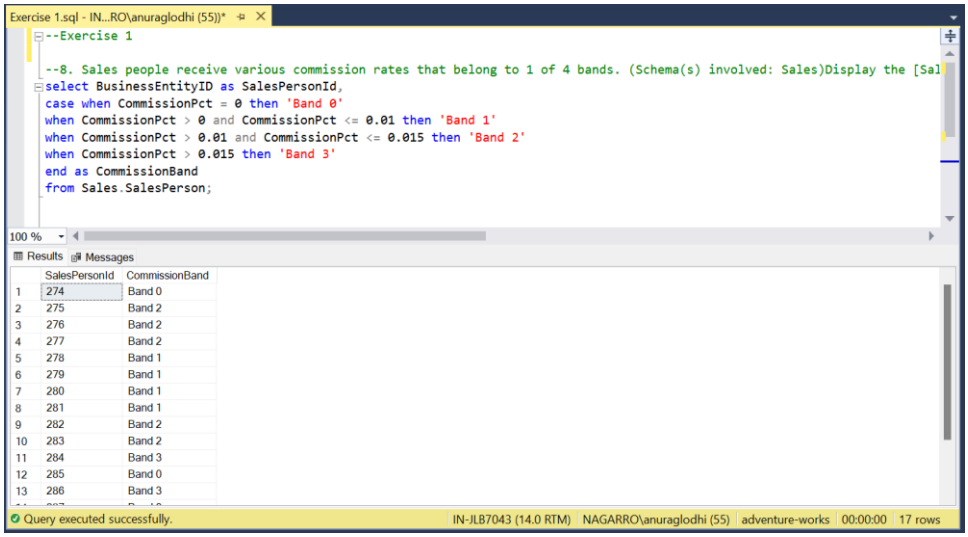
when CommissionPct > 0 and CommissionPct <= 0.01 then 'Band 1'

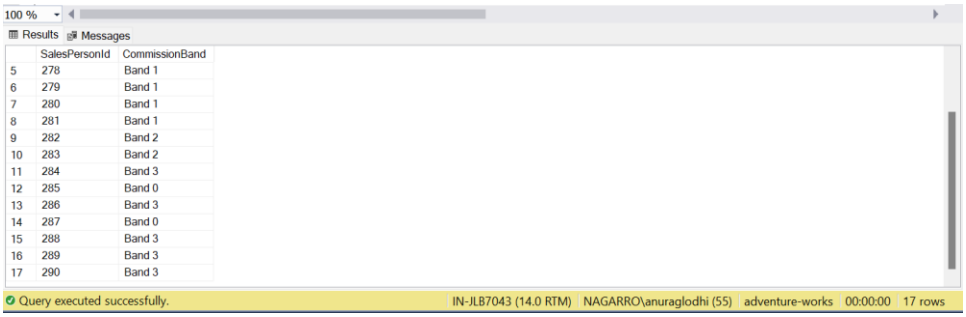
when CommissionPct > 0.01 and CommissionPct <= 0.015 then 'Band 2'

when CommissionPct > 0.015 then 'Band 3'

end as CommissionBand

from Sales.SalesPerson;





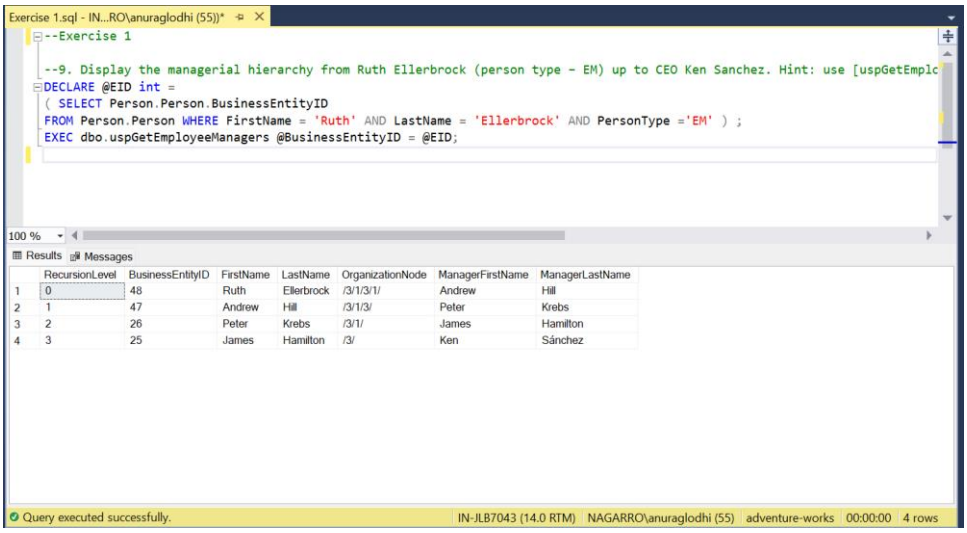
--9. Display the managerial hierarchy from Ruth Ellerbrock (person type – EM) up to CEO Ken Sanchez. Hint: use [uspGetEmployeeManagers] (Schema(s) involved: [Person], [HumanResources])

DECLARE @EID int =

(SELECT Person.Person.BusinessEntityID

FROM Person.Person WHERE FirstName = 'Ruth' AND LastName = 'Ellerbrock' AND PersonType ='EM') ;

EXEC dbo.uspGetEmployeeManagers @BusinessEntityID = @EID;



--10. Display the ProductId of the product with the largest stock level. Hint: Use the Scalar-valued function [dbo]. [UfnGetStock]. (Schema(s) involved: Production)*/

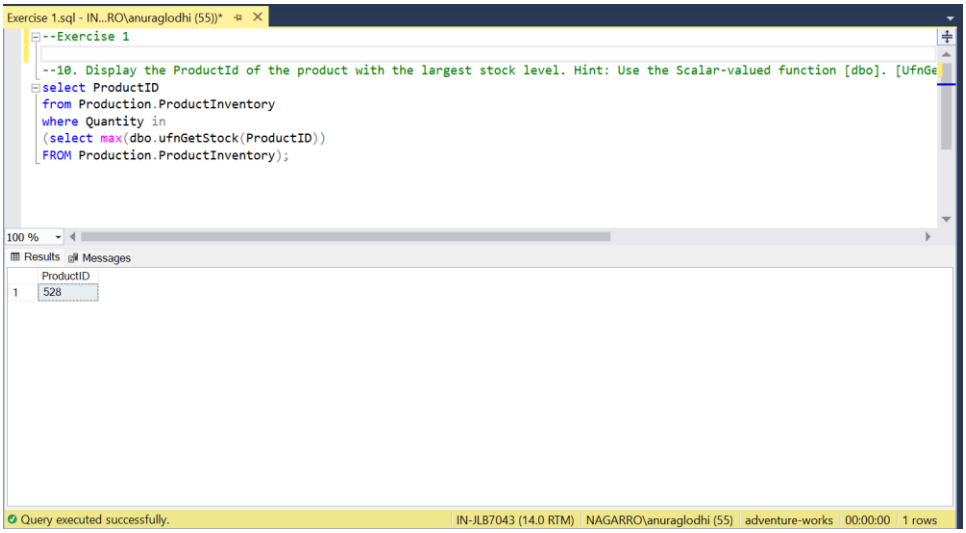
select ProductID

from Production.ProductInventory

where Quantity in

(select max(dbo.ufnGetStock(ProductID))

FROM Production.ProductInventory);



/*

Exercise 2: Write separate queries using a join, a subquery, a CTE, and then an EXISTS to list all AdventureWorks customers who have not placed

an order.

*/

--Using Join

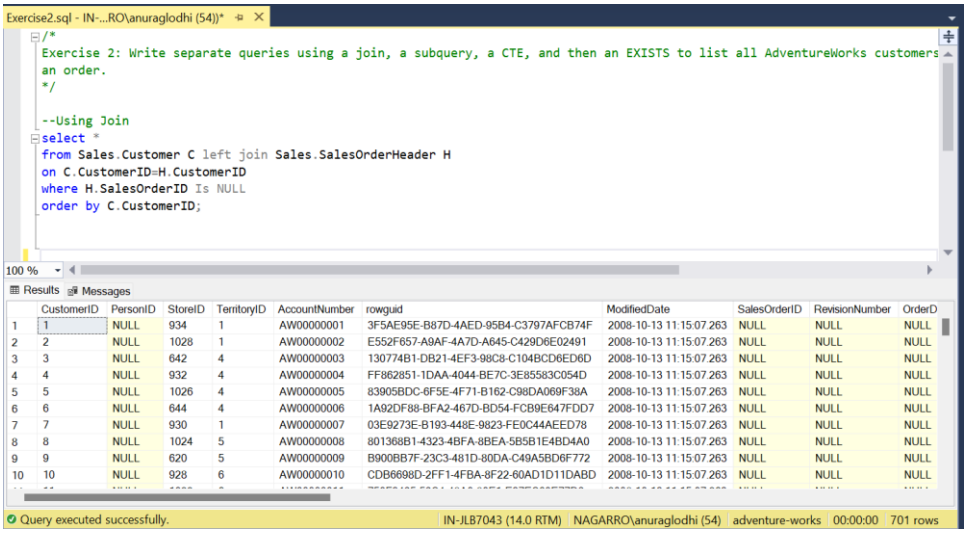
select *

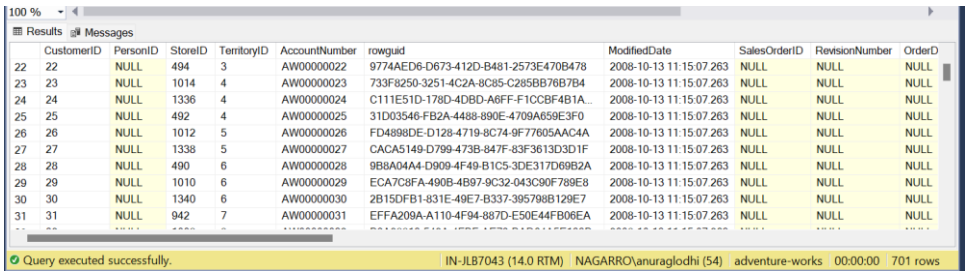
from Sales.Customer C left join Sales.SalesOrderHeader H

on C.CustomerID=H.CustomerID

where H.SalesOrderID Is NULL

order by C.CustomerID;





100 %

Results Messages

	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate	SalesOrderID	RevisionNumber	OrderD
22	22	NULL	494	3	AW00000022	9774AED6-D673-412D-B481-2573E470B478	2008-10-13 11:15:07.263	NULL	NULL	NULL
23	23	NULL	1014	4	AW00000023	733F8250-3251-4C2A-8C85-C285BB76B7B4	2008-10-13 11:15:07.263	NULL	NULL	NULL
24	24	NULL	1336	4	AW00000024	C111E51D-178D-4DBD-A6FF-F1CCBF4B1A...	2008-10-13 11:15:07.263	NULL	NULL	NULL
25	25	NULL	492	4	AW00000025	31D03546-FB2A-4488-890E-4709A659E3F0	2008-10-13 11:15:07.263	NULL	NULL	NULL
26	26	NULL	1012	5	AW00000026	FD4898DE-D128-4719-8C74-9F77605AAC4A	2008-10-13 11:15:07.263	NULL	NULL	NULL
27	27	NULL	1338	5	AW00000027	CACA5149-D799-473B-847F-83F3613D3D1F	2008-10-13 11:15:07.263	NULL	NULL	NULL
28	28	NULL	490	6	AW00000028	9B8A04A4-D909-4F49-B1C5-3DE317D69B2A	2008-10-13 11:15:07.263	NULL	NULL	NULL
29	29	NULL	1010	6	AW00000029	ECA7C8FA-490B-4B97-9C32-043C90F789E8	2008-10-13 11:15:07.263	NULL	NULL	NULL
30	30	NULL	1340	6	AW00000030	2B15DFB1-831E-49E7-B337-395798B129E7	2008-10-13 11:15:07.263	NULL	NULL	NULL
31	31	NULL	942	7	AW00000031	EFFA209A-A110-4F94-887D-E50E44FB06EA	2008-10-13 11:15:07.263	NULL	NULL	NULL

Query executed successfully. | IN-JLB7043 (14.0 RTM) | NAGARRO\anuraglodhi (54) | adventure-works | 00:00:00 | 701 rows

Results Messages

	CustomerID	PersonID	StoreID	TerritoryID	AccountNumber	rowguid	ModifiedDate	SalesOrderID	RevisionNumber	OrderD
692	692	NULL	802	1	AW00000692	9EE96279-78D5-49B7-AC4A-1C43CFB80249	2008-10-13 11:15:07.263	NULL	NULL	NULL
693	693	NULL	838	1	AW00000693	3B38D4CB-9573-4686-A202-9179DBEFC7B8	2008-10-13 11:15:07.263	NULL	NULL	NULL
694	694	NULL	1034	3	AW00000694	D3E874DC-A10C-4055-B943-9DBD5CFE8C89	2008-10-13 11:15:07.263	NULL	NULL	NULL
695	695	NULL	638	3	AW00000695	77A62D9C-4ED6-4266-AE35-40CC330ACC5E	2008-10-13 11:15:07.263	NULL	NULL	NULL
696	696	NULL	840	1	AW00000696	A0140DD4-4D10-4091-9D89-95873DAE9BB7	2008-10-13 11:15:07.263	NULL	NULL	NULL
697	697	NULL	1032	1	AW00000697	B30CE5B4-BBFA-4A57-B5C2-EED1EF13E0BE	2008-10-13 11:15:07.263	NULL	NULL	NULL
698	698	NULL	640	1	AW00000698	6BC4FF5A-0696-4C2A-88F9-08D1DBA91B74	2008-10-13 11:15:07.263	NULL	NULL	NULL
699	699	NULL	842	5	AW00000699	38A528AA-0402-4ED9-ABBB-983BCEB37D40	2008-10-13 11:15:07.263	NULL	NULL	NULL
700	700	NULL	1030	5	AW00000700	540FB57F-F81A-4794-8E9A-C071C4F0BC18	2008-10-13 11:15:07.263	NULL	NULL	NULL
701	701	NULL	844	6	AW00000701	61AE9625-8BD8-48B7-B171-8A90CBBA494C	2008-10-13 11:15:07.263	NULL	NULL	NULL

Query executed successfully. | IN-JLB7043 (14.0 RTM) | NAGARRO\anuraglodhi (54) | adventure-works | 00:00:00 | 701 rows

--Using SubQuery

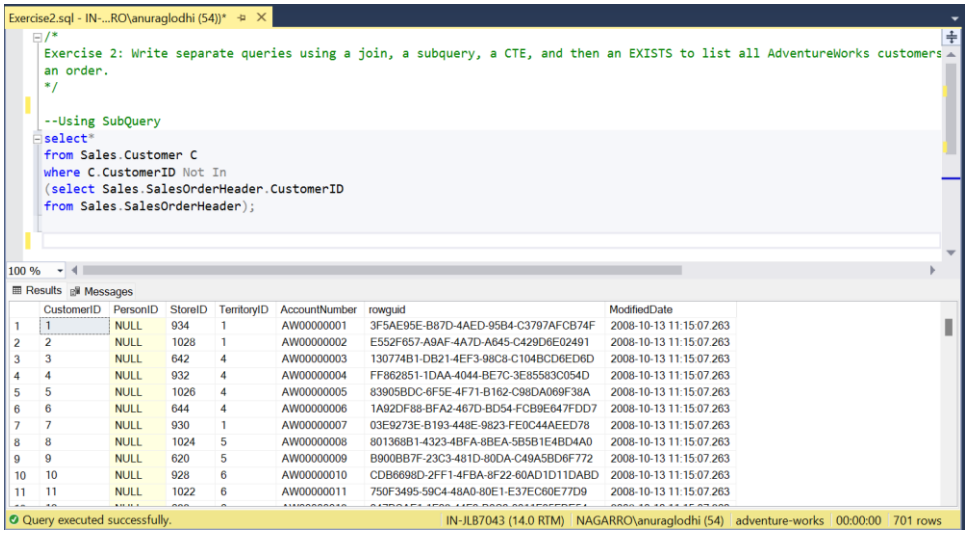
select*

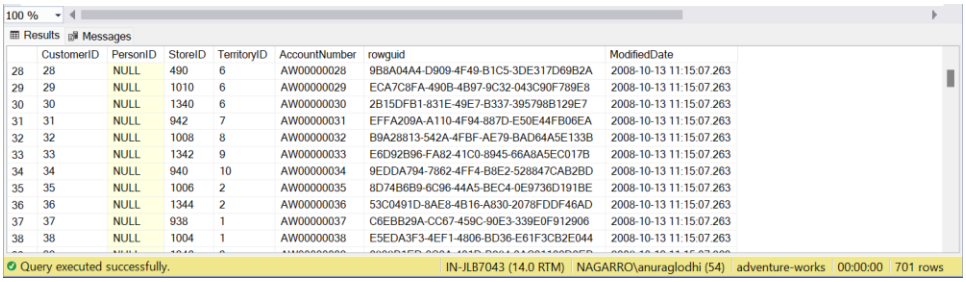
from Sales.Customer C

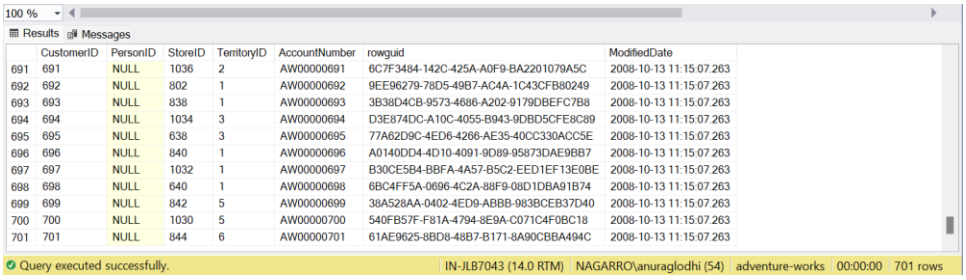
where C.CustomerID Not In

(select Sales.SalesOrderHeader.CustomerID

from Sales.SalesOrderHeader);







--Using CTE

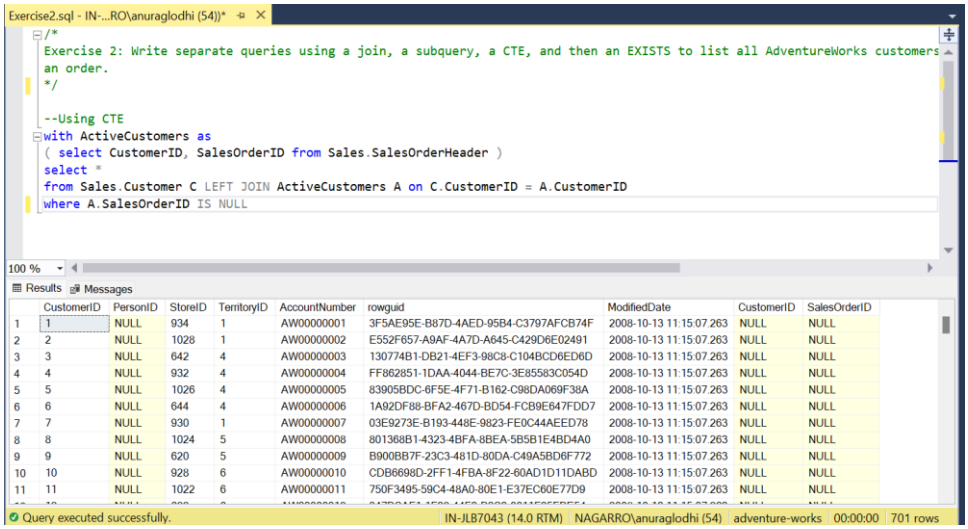
with ActiveCustomers as

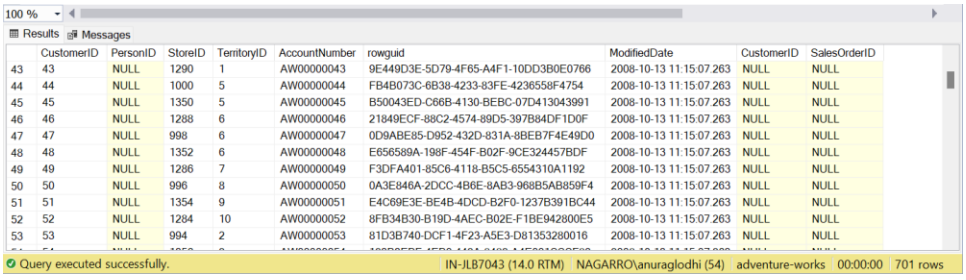
(select CustomerID, SalesOrderID from Sales.SalesOrderHeader)

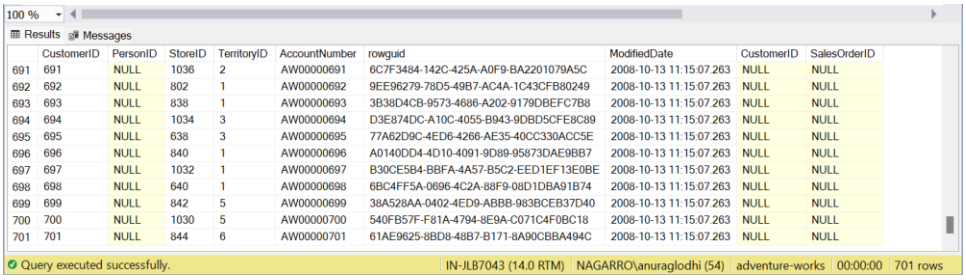
select *

from Sales.Customer C LEFT JOIN ActiveCustomers A on C.CustomerID = A.CustomerID

where A.SalesOrderID IS NULL







--Using Exist

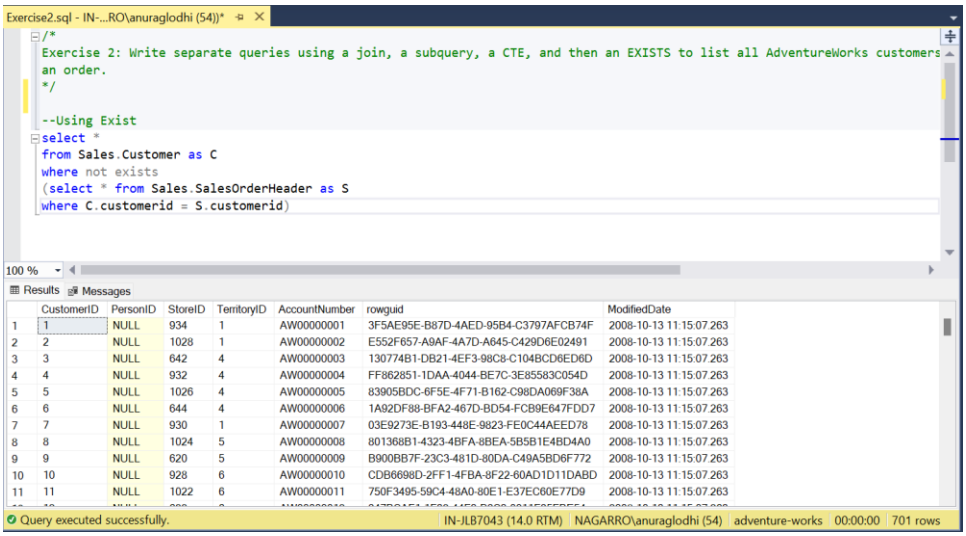
select *

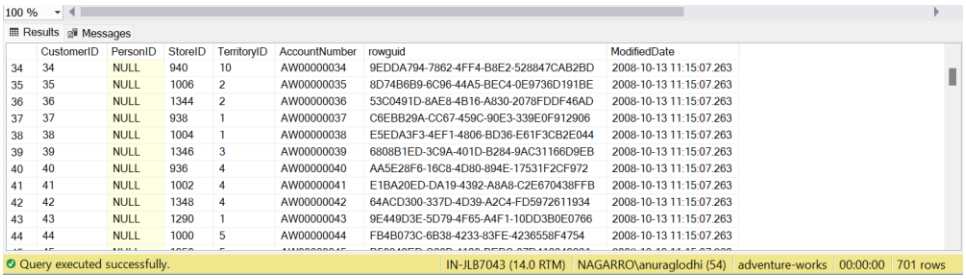
from Sales.Customer as C

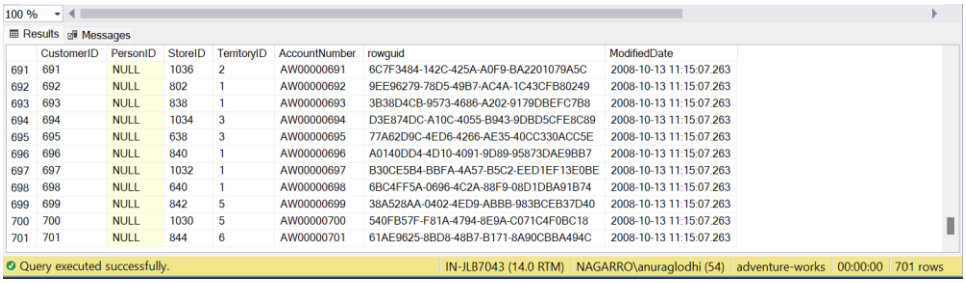
where not exists

(select * from Sales.SalesOrderHeader as S

where C.customerid = S.customerid)







/*

Exercise 3: Show the most recent five orders that were purchased from account numbers that have spent more than $70,000 with

AdventureWorks.

*/

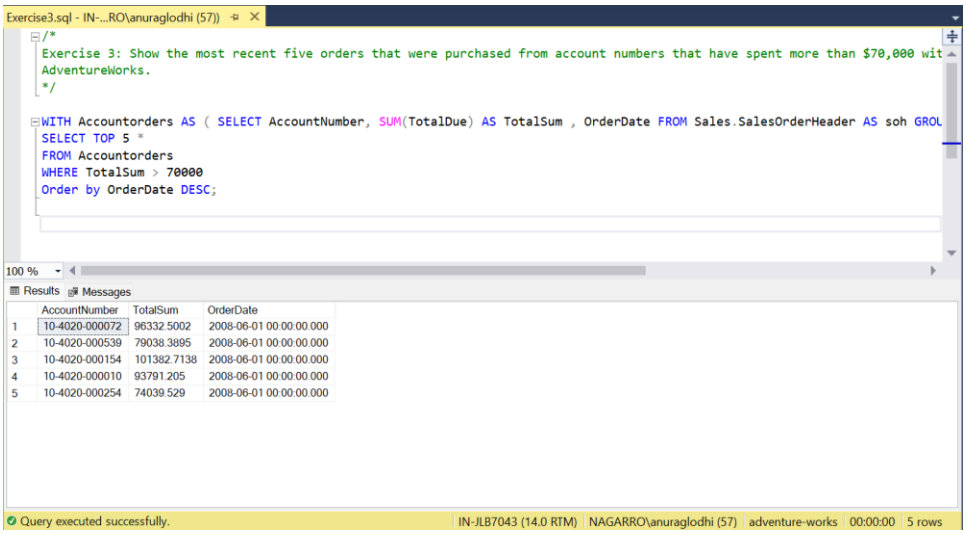
WITH Accountorders AS (SELECT AccountNumber, SUM(TotalDue) AS TotalSum , OrderDate FROM Sales.SalesOrderHeader AS soh GROUP BY AccountNumber, OrderDate)

SELECT TOP 5 *

FROM Accountorders

WHERE TotalSum > 70000

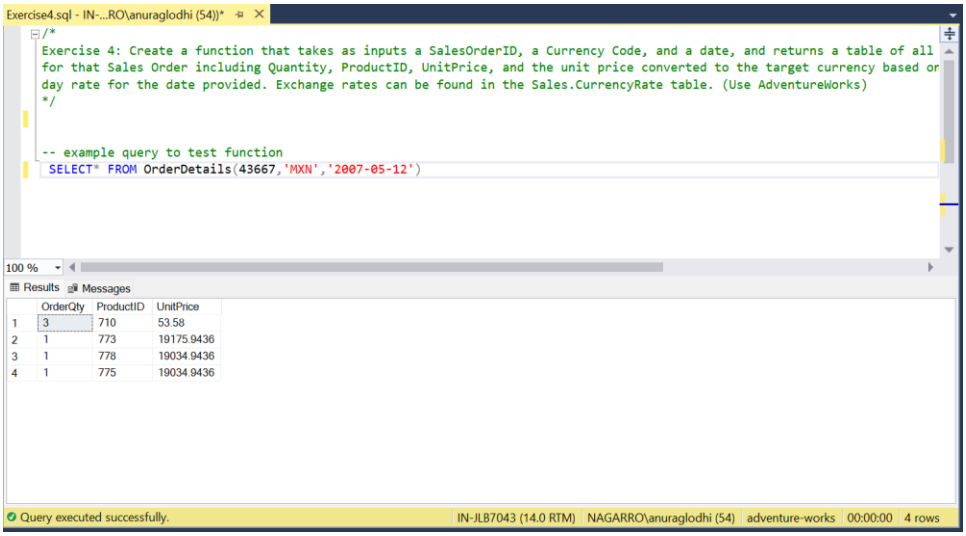
Order by OrderDate DESC;



```
/*
Exercise 4: Create a function that takes as inputs a SalesOrderID, a Currency Code, and a date, and returns a table of all the SalesOrderDetail rows
for that Sales Order including Quantity, ProductID, UnitPrice, and the unit price converted to the target currency based on the end of
day rate for the date provided. Exchange rates can be found in the Sales.CurrencyRate table. (Use AdventureWorks)
*/
CREATE FUNCTION OrderDetails( @SalesOrderId int, @CurrencyCode nvarchar(10), @Date date )
RETURNS TABLE
AS
RETURN
(SELECT OrderQty, ProductID, UnitPrice *
(SELECT EndOfDayRate
FROM    Sales.CurrencyRate
WHERE (ToCurrencyCode = @CurrencyCode) AND (ModifiedDate = @Date)) AS UnitPrice
FROM   Sales.SalesOrderDetail
WHERE (SalesOrderID = @SalesOrderId));
Go
```

-- example query to test function

SELECT* FROM OrderDetails(43667,'MXN','2007-05-12')



/*

Exercise 5: Write a Procedure supplying name information from the Person.Person table and accepting a filter for the first name. Alter the above

Store Procedure to supply Default Values if user does not enter any value. (Use AdventureWorks)

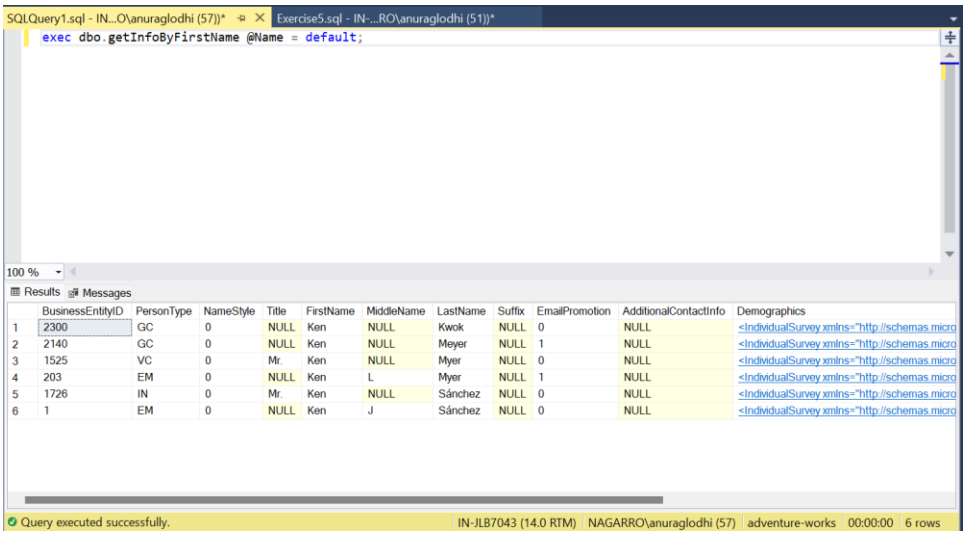
*/

-- if not prvoide any input it return the information according to default value ken

```
create procedure getInfoByFirstName
@name varchar(50) = 'ken'
as
select *
from Person.Person
where FirstName=@Name;
```

-- example query to test with default value and ruth as first name

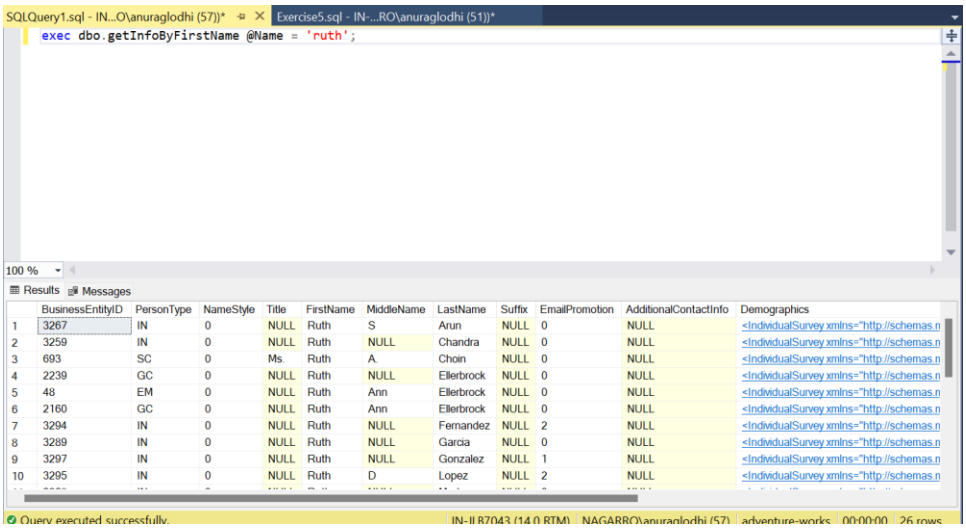
exec dbo.getInfoByFirstName @Name = default;



```
exec dbo.getInfoByFirstName @Name = default;
```

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Demographics
1	2300	GC	0	NULL	Ken	NULL	Kwok	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.micro
2	2140	GC	0	NULL	Ken	NULL	Meyer	NULL	1	NULL	<IndividualSurvey xmlns="http://schemas.micro
3	1525	VC	0	Mr.	Ken	NULL	Myer	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.micro
4	203	EM	0	NULL	Ken	L	Myer	NULL	1	NULL	<IndividualSurvey xmlns="http://schemas.micro
5	1726	IN	0	Mr.	Ken	NULL	Sánchez	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.micro
6	1	EM	0	NULL	Ken	J	Sánchez	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.micro

exec dbo.getInfoByFirstName @Name = 'ruth';



```
exec dbo.getInfoByFirstName @Name = 'ruth';
```

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Demographics
1	3267	IN	0	NULL	Ruth	S	Arun	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
2	3259	IN	0	NULL	Ruth	NULL	Chandra	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
3	693	SC	0	Ms.	Ruth	A.	Choin	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
4	2239	GC	0	NULL	Ruth	NULL	Ellerbrock	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
5	48	EM	0	NULL	Ruth	Ann	Ellerbrock	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
6	2160	GC	0	NULL	Ruth	Ann	Ellerbrock	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
7	3294	IN	0	NULL	Ruth	NULL	Fernandez	NULL	2	NULL	<IndividualSurvey xmlns="http://schemas.n
8	3289	IN	0	NULL	Ruth	NULL	Garcia	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
9	3297	IN	0	NULL	Ruth	NULL	Gonzalez	NULL	1	NULL	<IndividualSurvey xmlns="http://schemas.n
10	3295	IN	0	NULL	Ruth	D	Lopez	NULL	2	NULL	<IndividualSurvey xmlns="http://schemas.n

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo	Demographics
17	3299	IN	0	NULL	Ruth	R	Rodriguez	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
18	3266	IN	0	NULL	Ruth	D	Sai	NULL	1	NULL	<IndividualSurvey xmlns="http://schemas.n
19	3303	IN	0	NULL	Ruth	E	Sanchez	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
20	3277	IN	0	NULL	Ruth	L	Schmidt	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
21	3270	IN	0	NULL	Ruth	C	Srini	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
22	1833	SC	0	Ms.	Ruth	J.	Suffin	NULL	1	NULL	<IndividualSurvey xmlns="http://schemas.n
23	3283	IN	0	NULL	Ruth	NULL	Sullivan	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
24	3255	IN	0	NULL	Ruth	A	Suri	NULL	2	NULL	<IndividualSurvey xmlns="http://schemas.n
25	3262	IN	0	NULL	Ruth	D	Vance	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n
26	2015	SC	0	Ms.	Ruth	B.	Ward	NULL	0	NULL	<IndividualSurvey xmlns="http://schemas.n

/*

Exercise 6 :

Write a trigger for the Product table to ensure the list price can never be raised more than 15 Percent in a single change.

Modify the above trigger to execute its check code only if the ListPrice column is updated (Use AdventureWorks Database)

*/

```
Create TRIGGER Production.ListPrice

ON

Production.Product

For UPDATE

AS

IF Update(ListPrice)

BEGIN

IF EXISTS

(

SELECT 'True'

FROM inserted i JOIN deleted d ON i.ProductID = d.ProductID

WHERE i.ListPrice > (d.ListPrice * 1.15)

)

BEGIN

RAISERROR('Price increase may not be greater than 15 percent.Transaction Failed.',16,1)

ROLLBACK TRAN

END

END
```

-- example query to update the trigger this trigger

Before trigger we can change the value of list price to 100.

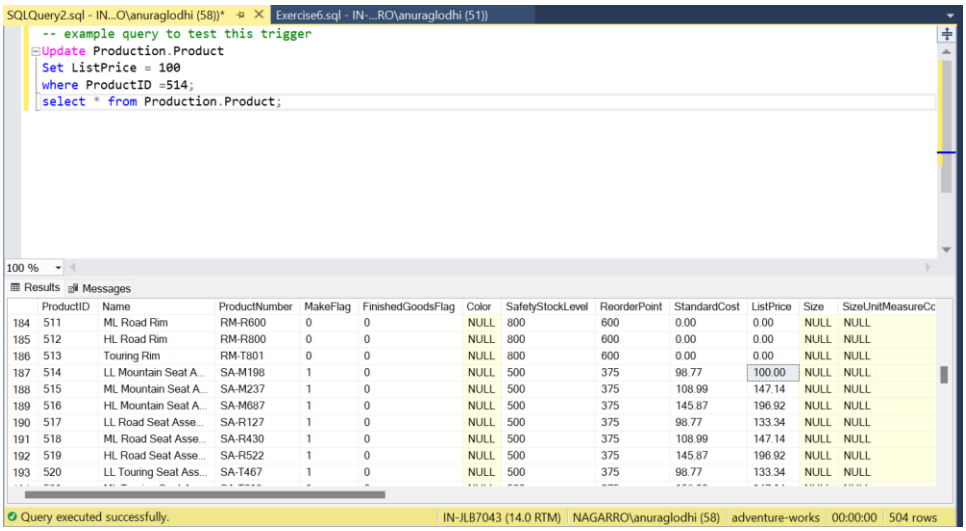
Update Production.Product

Set ListPrice = 100

where ProductID =514;

select * from Production.Product;

List Price is 100 for product id 514



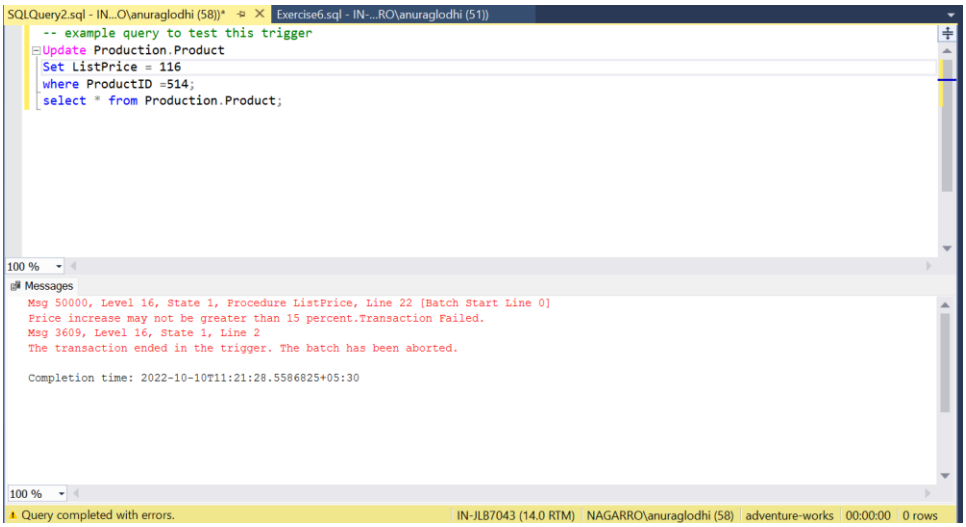
	ProductID	Name	ProductNumber	MakeFlag	FinishedGoodsFlag	Color	SafetyStockLevel	ReorderPoint	StandardCost	ListPrice	Size	SizeUnitMeasureCc
184	511	ML Road Rim	RM-R600	0	0	NULL	800	600	0.00	0.00	NULL	NULL
185	512	HL Road Rim	RM-R800	0	0	NULL	800	600	0.00	0.00	NULL	NULL
186	513	Touring Rim	RM-T801	0	0	NULL	800	600	0.00	0.00	NULL	NULL
187	514	LL Mountain Seat A...	SA-M198	1	0	NULL	500	375	98.77	100.00	NULL	NULL
188	515	ML Mountain Seat A...	SA-M237	1	0	NULL	500	375	108.99	147.14	NULL	NULL
189	516	HL Mountain Seat A...	SA-M687	1	0	NULL	500	375	145.87	196.92	NULL	NULL
190	517	LL Road Seat Asse...	SA-R127	1	0	NULL	500	375	98.77	133.34	NULL	NULL
191	518	ML Road Seat Asse...	SA-R430	1	0	NULL	500	375	108.99	147.14	NULL	NULL
192	519	HL Road Seat Asse...	SA-R522	1	0	NULL	500	375	145.87	196.92	NULL	NULL
193	520	LL Touring Seat Ass...	SA-T467	1	0	NULL	500	375	98.77	133.34	NULL	NULL

After executing trigger we can’t update the list pricefrom more than 115.

Update Production.Product

Set ListPrice = 116

where ProductID =514;

select * from Production.Product;

Now increasing it by more than 15 percent i.e. 116 we can see the trigger error