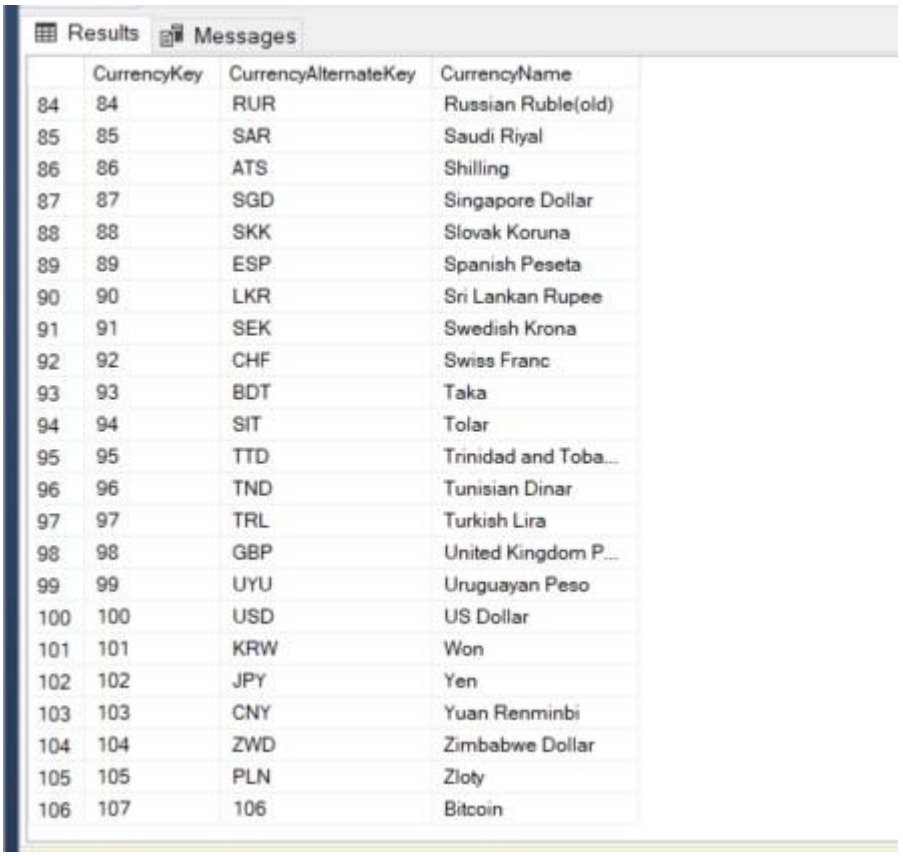
Ejercicios a realizar
De SQL básico:
Tema 1 y 2

1. Realizar una inserción en cualquier tabla.

Insertar Bitcoin en la tabla de monedas



Results | Messages

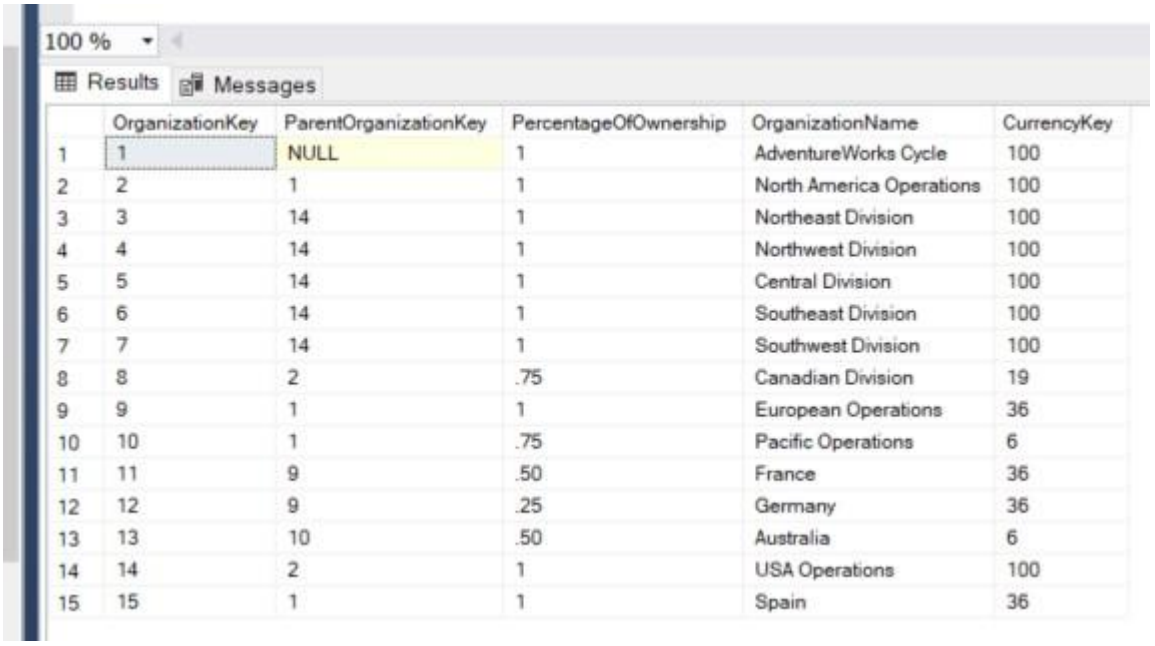
	CurrencyKey	CurrencyAlternateKey	CurrencyName
84	84	RUR	Russian Ruble(old)
85	85	SAR	Saudi Riyal
86	86	ATS	Shilling
87	87	SGD	Singapore Dollar
88	88	SKK	Slovak Koruna
89	89	ESP	Spanish Peseta
90	90	LKR	Sri Lankan Rupee
91	91	SEK	Swedish Krona
92	92	CHF	Swiss Franc
93	93	BDT	Taka
94	94	SIT	Tolar
95	95	TTD	Trinidad and Toba...
96	96	TND	Tunisian Dinar
97	97	TRL	Turkish Lira
98	98	GBP	United Kingdom P...
99	99	UYU	Uruguayan Peso
100	100	USD	US Dollar
101	101	KRW	Won
102	102	JPY	Yen
103	103	CNY	Yuan Renminbi
104	104	ZWD	Zimbabwe Dollar
105	105	PLN	Zloty
106	107	106	Bitcoin

```
INSERT INTO DimCurrency
  VALUES (106, 'Bitcoin');
```

2. Realizar una inserción en cualquier tabla que posea una clave ajena.

Insertar los valores 15,1,1, Spain y 36 en la tabla DimOrganization

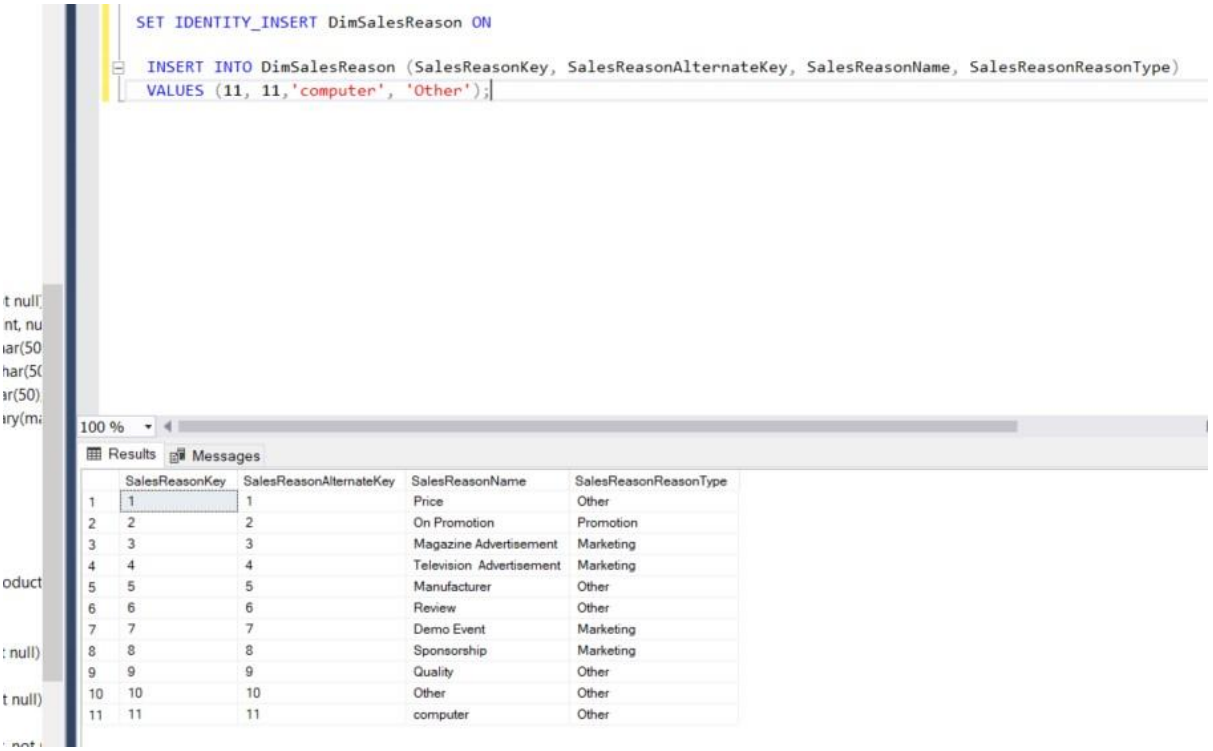
```
SET IDENTITY_INSERT DimOrganization ON
 INSERT INTO DimOrganization
(OrganizationKey,ParentOrganizationKey,PercentageOfOwnership,OrganizationName,Curre
ncyKey)
 VALUES (15,1,1,'Spain',36)
```



100 %

Results | Messages

	OrganizationKey	ParentOrganizationKey	PercentageOfOwnership	OrganizationName	CurrencyKey
1	1	NULL	1	AdventureWorks Cycle	100
2	2	1	1	North America Operations	100
3	3	14	1	Northeast Division	100
4	4	14	1	Northwest Division	100
5	5	14	1	Central Division	100
6	6	14	1	Southeast Division	100
7	7	14	1	Southwest Division	100
8	8	2	.75	Canadian Division	19
9	9	1	1	European Operations	36
10	10	1	.75	Pacific Operations	6
11	11	9	.50	France	36
12	12	9	.25	Germany	36
13	13	10	.50	Australia	6
14	14	2	1	USA Operations	100
15	15	1	1	Spain	36



```
SET IDENTITY_INSERT DimSalesReason ON

INSERT INTO DimSalesReason (SalesReasonKey, SalesReasonAlternateKey, SalesReasonName, SalesReasonReasonType)
VALUES (11, 11,'computer', 'Other');
```

100 %

Results | Messages

	SalesReasonKey	SalesReasonAlternateKey	SalesReasonName	SalesReasonReasonType
1	1	1	Price	Other
2	2	2	On Promotion	Promotion
3	3	3	Magazine Advertisement	Marketing
4	4	4	Television Advertisement	Marketing
5	5	5	Manufacturer	Other
6	6	6	Review	Other
7	7	7	Demo Event	Marketing
8	8	8	Sponsorship	Marketing
9	9	9	Quality	Other
10	10	10	Other	Other
11	11	11	computer	Other

3. Realizar una actualización de una fila en cualquier tabla.

Cambiar el valor de la monedas que valgan 100 a 66 en DimOrganitation

UPDATE DimOrganization SET CurrencyKey = 66 where CurrencyKey = 100

```
UPDATE DimOrganization SET CurrencyKey = 66 where CurrencyKey = 100
```

0 %

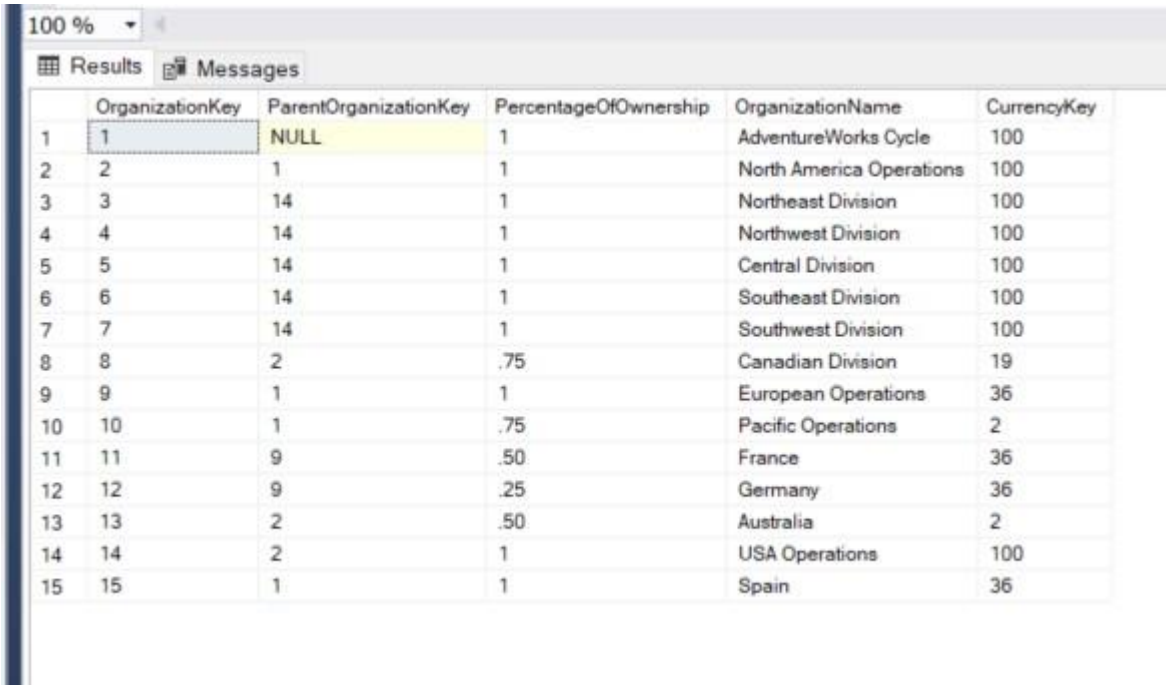
Results | Messages

	OrganizationKey	ParentOrganizationKey	PercentageOfOwnership	OrganizationName	CurrencyKey
	1	NULL	1	AdventureWorks Cycle	66
	2	1	1	North America Operations	66
	3	14	1	Northeast Division	66
	4	14	1	Northwest Division	66
	5	14	1	Central Division	66
	6	14	1	Southeast Division	66
	7	14	1	Southwest Division	66
	8	2	.75	Canadian Division	19
	9	1	1	European Operations	36
0	10	1	.75	Pacific Operations	2
1	11	9	.50	France	36
2	12	9	.25	Germany	36
3	13	2	.50	Australia	2
4	14	2	1	USA Operations	66

4. Realizar una actualización de una fila en cualquier tabla padre que posea referencias de clave ajena en una tabla hija.

Actualizar de la tabla DimOrganization la ParentOrgatization key, de 10 a 2

UPDATE DimOrganization SET ParentOrganizationKey = 2 where ParentOrganizationKey = 10



100 %

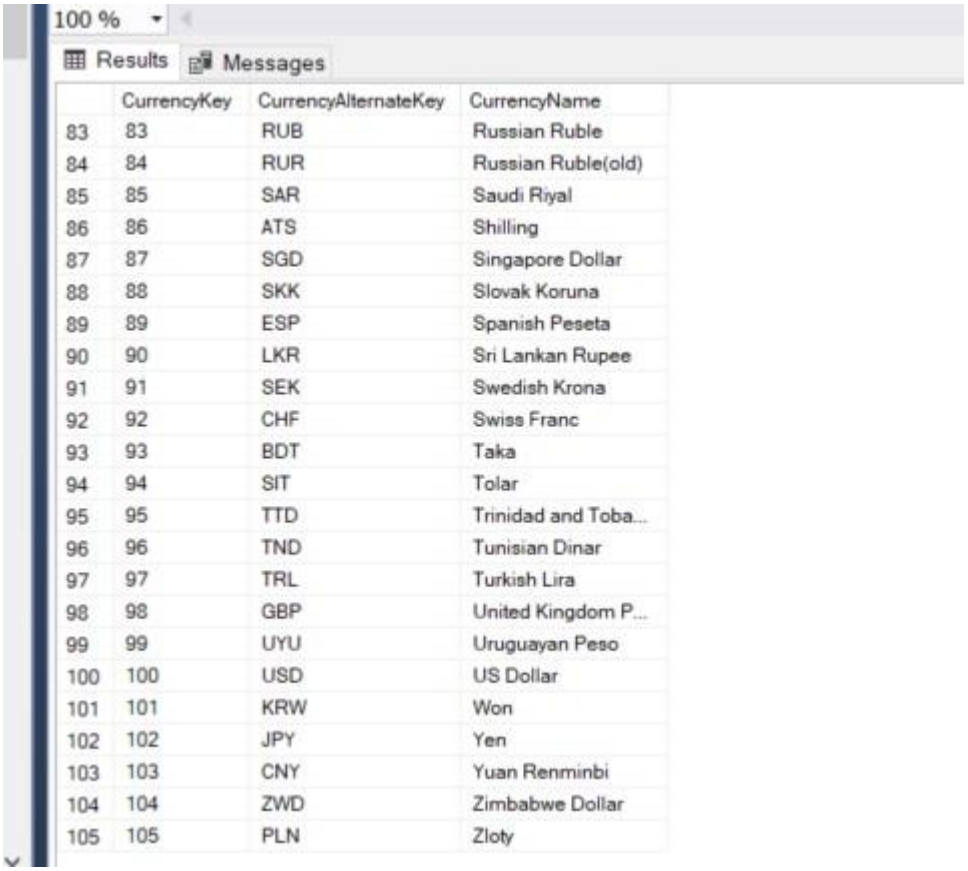
Results | Messages

	OrganizationKey	ParentOrganizationKey	PercentageOfOwnership	OrganizationName	CurrencyKey
1	1	NULL	1	AdventureWorks Cycle	100
2	2	1	1	North America Operations	100
3	3	14	1	Northeast Division	100
4	4	14	1	Northwest Division	100
5	5	14	1	Central Division	100
6	6	14	1	Southeast Division	100
7	7	14	1	Southwest Division	100
8	8	2	.75	Canadian Division	19
9	9	1	1	European Operations	36
10	10	1	.75	Pacific Operations	2
11	11	9	.50	France	36
12	12	9	.25	Germany	36
13	13	2	.50	Australia	2
14	14	2	1	USA Operations	100
15	15	1	1	Spain	36

5. Realizar una eliminación de una fila en cualquier tabla.

Borrar la fila de bitcoin de la tabla DimCurrency

```
DELETE FROM DimCurrency
 WHERE CurrencyName = 'Bitcoin'
```



100 %

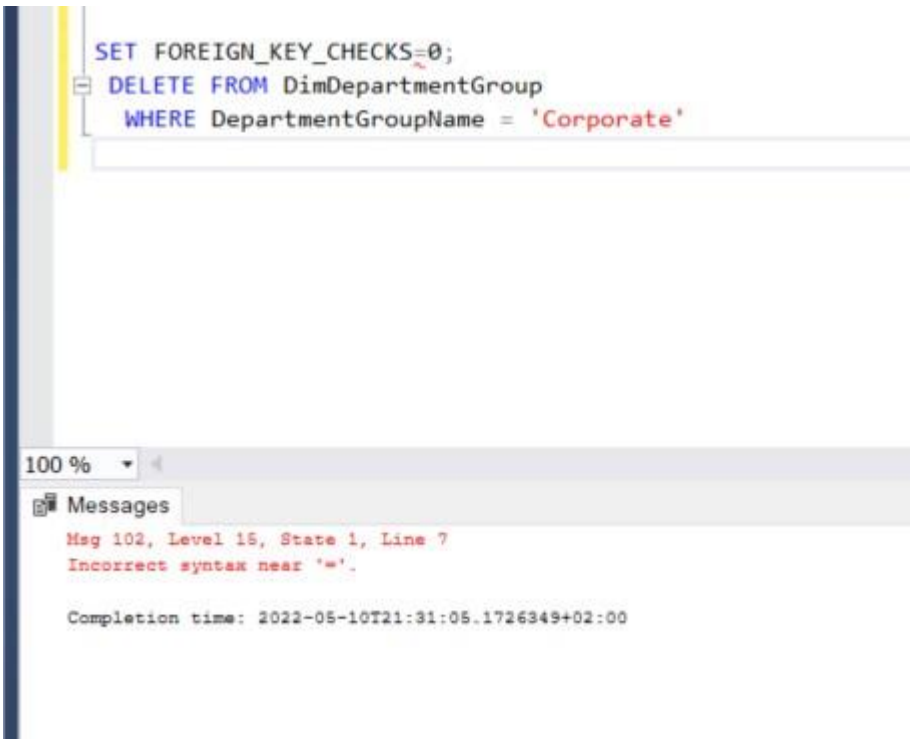
Results | Messages

	CurrencyKey	CurrencyAlternateKey	CurrencyName
83	83	RUB	Russian Ruble
84	84	RUR	Russian Ruble(old)
85	85	SAR	Saudi Riyal
86	86	ATS	Shilling
87	87	SGD	Singapore Dollar
88	88	SKK	Slovak Koruna
89	89	ESP	Spanish Peseta
90	90	LKR	Sri Lankan Rupee
91	91	SEK	Swedish Krona
92	92	CHF	Swiss Franc
93	93	BDT	Taka
94	94	SIT	Tolar
95	95	TTD	Trinidad and Toba...
96	96	TND	Tunisian Dinar
97	97	TRL	Turkish Lira
98	98	GBP	United Kingdom P...
99	99	UYU	Uruguayan Peso
100	100	USD	US Dollar
101	101	KRW	Won
102	102	JPY	Yen
103	103	CNY	Yuan Renminbi
104	104	ZWD	Zimbabwe Dollar
105	105	PLN	Zloty

6. Realizar una eliminación de una fila en cualquier tabla padre que posea referencias de clave ajena en una tabla hija.

Borrar el valor Corporate de la fila DeparmentGroupName

```
SET FOREIGN_KEY_CHECKS=0;
 DELETE FROM DimDepartmentGroup
  WHERE DepartmentGroupName = 'Corporate'
```



Tema 3

7. Realizar una consulta que contenga una condición de búsqueda compuesta (incluir al menos dos de los siguientes elementos: un test de comparación, test de rango, test de pertenencia a conjunto, test de correspondencia con patrón o un test de valor nulo).

Seleccionar SalesReasonName donde coincida typo Marketing y SalesReasonKey sea igual a 3

```
SELECT SalesReasonName
  FROM DimSalesReason
    WHERE SalesReasonReasonType='Marketing' and  SalesReasonKey=3
```

```
SELECT SalesReasonName
FROM DimSalesReason
WHERE SalesReasonReasonType='Marketing' and  SalesReasonKey=3
```

8. Realizar una consulta que muestre filas únicas descartando filas duplicadas (DISTINCT).

Mostrar solamente (sin duplicar) las razones de venta de los artículos en la tabla SalesReason

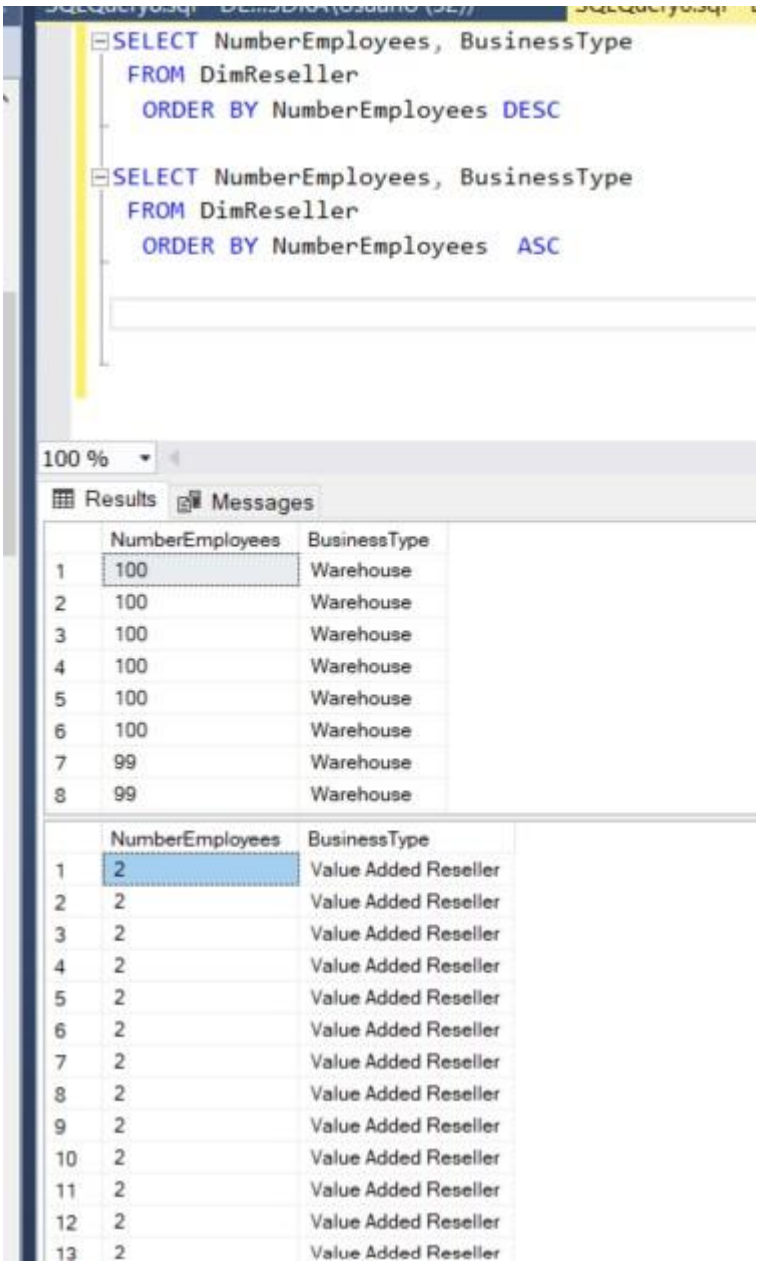
```
SELECT DISTINCT SalesReasonReasonType
 FROM DimSalesReason
```

9. Realizar una consulta que ordene los datos por dos campos (uno de ellos de forma ascendente y el otro de forma descendente).

Ordenar por el número de empleado en la tabla DimReseller el tipo de negocio de cada uno

```
SELECT NumberEmployees, BusinessType
 FROM DimReseller
  ORDER BY NumberEmployees DESC

SELECT NumberEmployees, BusinessType
 FROM DimReseller
  ORDER BY NumberEmployees  ASC
```



10. Realizar una consulta que utilice la cláusula GROUP BY

Seleccionar ProductLine y agruparlos por ProductLine

```
SELECT ProductLine
 FROM DimReseller
  GROUP BY ProductLine
```

```
SELECT ProductLine
FROM DimReseller
GROUP BY ProductLine
```

100 %

Results | Messages

	ProductLine
1	Mountain
2	Road
3	Touring

11. Realizar una consulta que utilice las cláusulas GROUP BY y HAVING.

Mostrar los tipos de negocio según el año de apertura y que este sea que sea posterior al año 2000

```
SELECT YearOpened, BusinessType
FROM DimReseller
GROUP BY YearOpened, BusinessType
  HAVING YearOpened>2000
```

```
SELECT YearOpened, BusinessType
FROM DimReseller
GROUP BY YearOpened, BusinessType
  HAVING YearOpened>2000
```

100 %

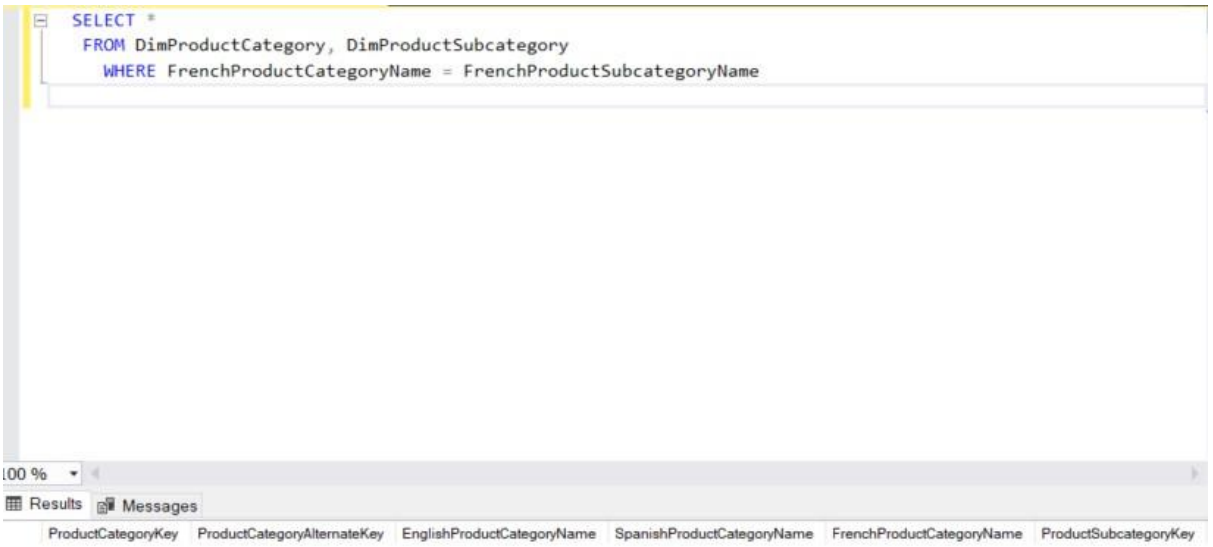
Results | Messages

	YearOpened	BusinessType
1	2001	Specialty Bike Shop
2	2001	Value Added Reseller
3	2002	Specialty Bike Shop
4	2002	Warehouse
5	2003	Value Added Reseller
6	2003	Warehouse
7	2004	Value Added Reseller
8	2005	Specialty Bike Shop

Tema 4

12. Realizar una consulta que combine dos tablas.

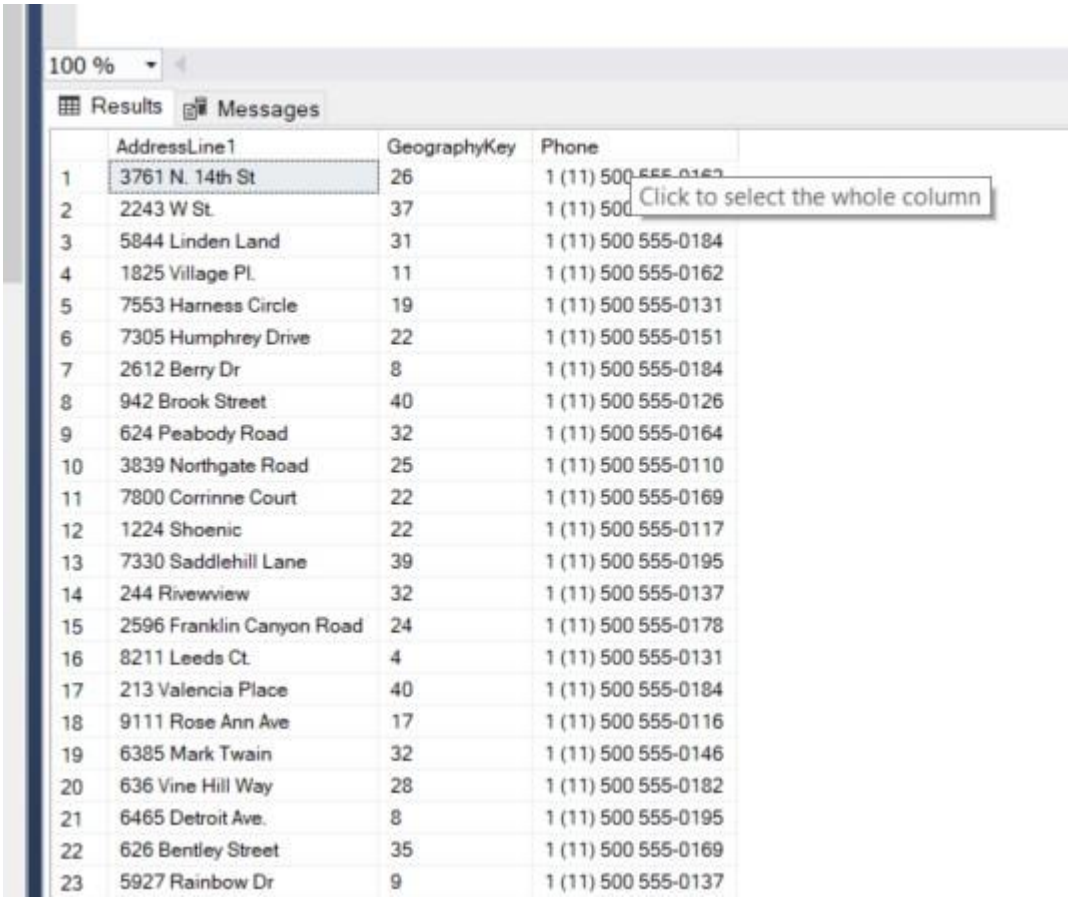
```
SELECT *
 FROM DimProductCategory, DimProductSubcategory
  WHERE FrenchProductCategoryName = FrenchProductSubcategoryName
```



13. Realizar una autocombinación.

Seleccionar de la tabla DimCustomer, la dirección, teléfono y geographykey donde esta última sea menor de 50

```
SELECT AddressLine1, GeographyKey, Phone
  FROM DimCustomer
    WHERE GeographyKey < 50
```



	AddressLine1	GeographyKey	Phone
1	3761 N. 14th St	26	1 (11) 500 555-0162
2	2243 W St.	37	1 (11) 500
3	5844 Linden Land	31	1 (11) 500 555-0184
4	1825 Village Pl.	11	1 (11) 500 555-0162
5	7553 Harness Circle	19	1 (11) 500 555-0131
6	7305 Humphrey Drive	22	1 (11) 500 555-0151
7	2612 Berry Dr	8	1 (11) 500 555-0184
8	942 Brook Street	40	1 (11) 500 555-0126
9	624 Peabody Road	32	1 (11) 500 555-0164
10	3839 Northgate Road	25	1 (11) 500 555-0110
11	7800 Corrinne Court	22	1 (11) 500 555-0169
12	1224 Shoenic	22	1 (11) 500 555-0117
13	7330 Saddlehill Lane	39	1 (11) 500 555-0195
14	244 Rivewview	32	1 (11) 500 555-0137
15	2596 Franklin Canyon Road	24	1 (11) 500 555-0178
16	8211 Leeds Ct.	4	1 (11) 500 555-0131
17	213 Valencia Place	40	1 (11) 500 555-0184
18	9111 Rose Ann Ave	17	1 (11) 500 555-0116
19	6385 Mark Twain	32	1 (11) 500 555-0146
20	636 Vine Hill Way	28	1 (11) 500 555-0182
21	6465 Detroit Ave.	8	1 (11) 500 555-0195
22	626 Bentley Street	35	1 (11) 500 555-0169
23	5927 Rainbow Dr	9	1 (11) 500 555-0137

14. Realizar una combinación interna.

Conocer los clientes de Berlín, con su dirección nºtelefono y código postal:

```
SELECT AddressLine1, Phone, PostalCode
  FROM DimCustomer, DimGeography
  WHERE City ='Berlin'
```

```
SELECT AddressLine1, Phone, PostalCode
FROM DimCustomer, DimGeography
WHERE City ='Berlin'
```

100 %

Results | Messages

	AddressLine1	Phone	PostalCode
1	3761 N. 14th St	1 (11) 500 555-0162	14197
2	2243 W St.	1 (11) 500 555-0110	14197
3	5844 Linden Land	1 (11) 500 555-0184	14197
4	1825 Village Pl.	1 (11) 500 555-0162	14197
5	7553 Harness Circle	1 (11) 500 555-0131	14197
6	7305 Humphrey Drive	1 (11) 500 555-0151	14197
7	2612 Berry Dr	1 (11) 500 555-0184	14197
8	942 Brook Street	1 (11) 500 555-0126	14197
9	624 Peabody Road	1 (11) 500 555-0164	14197
10	3839 Northgate Road	1 (11) 500 555-0110	14197
11	7800 Corrinne Court	1 (11) 500 555-0169	14197
12	1224 Shoenic	1 (11) 500 555-0117	14197
13	4785 Scott Street	717-555-0164	14197
14	7902 Hudson Ave.	817-555-0185	14197
15	9011 Tank Drive	431-555-0156	14197
16	244 Willow Pass Road	208-555-0142	14197
17	9666 Northridge Ct.	135-555-0171	14197
18	7330 Saddlehill Lane	1 (11) 500 555-0195	14197
19	244 Rivewview	1 (11) 500 555-0137	14197
20	7832 Landing Dr	262-555-0112	14197
21	7156 Rose Dr	550-555-0163	14197

15. Realizar una combinación externa izquierda.

```
SELECT AddressLine1, Phone, PostalCode
  FROM DimCustomer LEFT JOIN DimGeography ON City='Berlin'
```

```
SELECT AddressLine1, Phone, PostalCode
FROM DimCustomer LEFT JOIN DimGeography ON City='Berlin'
```

100 %

Results | Messages

	AddressLine1	Phone	PostalCode
1	3761 N. 14th St	1 (11) 500 555-0162	14197
2	3761 N. 14th St	1 (11) 500 555-0162	12171
3	3761 N. 14th St	1 (11) 500 555-0162	13441
4	3761 N. 14th St	1 (11) 500 555-0162	14111
5	3761 N. 14th St	1 (11) 500 555-0162	14129
6	3761 N. 14th St	1 (11) 500 555-0162	10210
7	3761 N. 14th St	1 (11) 500 555-0162	10791
8	3761 N. 14th St	1 (11) 500 555-0162	12311
9	3761 N. 14th St	1 (11) 500 555-0162	14111
10	3761 N. 14th St	1 (11) 500 555-0162	10501
11	3761 N. 14th St	1 (11) 500 555-0162	14197
12	3761 N. 14th St	1 (11) 500 555-0162	12171
13	3761 N. 14th St	1 (11) 500 555-0162	12311
14	3761 N. 14th St	1 (11) 500 555-0162	14197
15	2243 W St.	1 (11) 500 555-0110	14197
16	2243 W St.	1 (11) 500 555-0110	12171
17	2243 W St.	1 (11) 500 555-0110	13441
18	2243 W St.	1 (11) 500 555-0110	14111
19	2243 W St.	1 (11) 500 555-0110	14129
20	2243 W St.	1 (11) 500 555-0110	10210
21	2243 W St.	1 (11) 500 555-0110	10791
22	2243 W St.	1 (11) 500 555-0110	12311

16. Realizar una combinación externa derecha.

```
SELECT AddressLine1, Phone, PostalCode
  FROM DimCustomer RIGHT JOIN DimGeography ON City='Berlin'
```

```
SELECT AddressLine1, Phone, PostalCode
FROM DimCustomer RIGHT JOIN DimGeography ON City='Berlin'
```

100 %

Results | Messages

	AddressLine1	Phone	PostalCode
1	NULL	NULL	91932
2	NULL	NULL	90802
3	NULL	NULL	92625
4	NULL	NULL	94566
5	NULL	NULL	91776
6	NULL	NULL	90401
7	NULL	NULL	2450
8	NULL	NULL	2300
9	NULL	NULL	1002
10	NULL	NULL	4700
11	NULL	NULL	3220
12	NULL	NULL	T5
13	NULL	NULL	90505
14	NULL	NULL	94596
15	NULL	NULL	80537
16	NULL	NULL	06901
17	NULL	NULL	33143
18	NULL	NULL	V3A 4R2
19	NULL	NULL	V6B 3P7
20	NULL	NULL	V7L 4J4
21	NULL	NULL	L6W 2T7
22	NULL	NULL	L5B 3V4

17. Realizar una combinación externa completa.

Conocer los clientes con su nombre, su dirección nºtelefono, código postal que vivan en Gran Bretaña (GB)

```
SELECT AddressLine1, Phone, PostalCode,FirstName,City
  FROM DimCustomer JOIN DimGeography ON CountryRegionCode='GB'
```

```
SELECT AddressLine1, Phone, PostalCode,FirstName,City
FROM DimCustomer JOIN DimGeography ON CountryRegionCode='GB'
```

100 %

Results | Messages

	AddressLine1	Phone	PostalCode	FirstName	City
1	3761 N. 14th St	1 (11) 500 555-0162	OX14 4SE	Jon	Abingdon
2	2243 W St.	1 (11) 500 555-0110	OX14 4SE	Eugene	Abingdon
3	5844 Linden Land	1 (11) 500 555-0184	OX14 4SE	Ruben	Abingdon
4	1825 Village Pl.	1 (11) 500 555-0162	OX14 4SE	Christy	Abingdon
5	7553 Harness Circle	1 (11) 500 555-0131	OX14 4SE	Elizabeth	Abingdon
6	7305 Humphrey Drive	1 (11) 500 555-0151	OX14 4SE	Julio	Abingdon
7	2612 Berry Dr	1 (11) 500 555-0184	OX14 4SE	Janet	Abingdon
8	942 Brook Street	1 (11) 500 555-0126	OX14 4SE	Marco	Abingdon
9	624 Peabody Road	1 (11) 500 555-0164	OX14 4SE	Rob	Abingdon
10	3839 Northgate Road	1 (11) 500 555-0110	OX14 4SE	Shannon	Abingdon
11	7800 Corrinne Court	1 (11) 500 555-0169	OX14 4SE	Jacquelyn	Abingdon
12	1224 Shoenic	1 (11) 500 555-0117	OX14 4SE	Curtis	Abingdon
13	4785 Scott Street	717-555-0164	OX14 4SE	Lauren	Abingdon
14	7902 Hudson Ave.	817-555-0185	OX14 4SE	Ian	Abingdon
15	9011 Tank Drive	431-555-0156	OX14 4SE	Sydney	Abingdon
16	244 Willow Pass Road	208-555-0142	OX14 4SE	Chloe	Abingdon
17	9666 Northridge Ct.	135-555-0171	OX14 4SE	Wyatt	Abingdon
18	7330 Saddlehill Lane	1 (11) 500 555-0195	OX14 4SE	Shannon	Abingdon
19	244 Riveewview	1 (11) 500 555-0137	OX14 4SE	Clarence	Abingdon
20	7832 Landing Dr	262-555-0112	OX14 4SE	Luke	Abingdon

Tema 5

18. Realizar una consulta con subconsulta con referencias externas.

Conocer todos los clientes de nombre Jon y que tengan 5 hijos

```
SELECT FirstName
  FROM DimCustomer
  WHERE FirstName = 'Jon'  (SELECT TotalChildren  = '5')
```

```
SELECT FirstName
FROM DimCustomer
WHERE FirstName = 'Jon'  (SELECT TotalChildren  ='5')
```

100 %

Results | Messages

	FirstName
1	Jon
2	Jon
3	Jon
4	Jon
5	Jon
6	Jon
7	Jon
8	Jon
9	Jon
10	Jon
11	Jon
12	Jon
13	Jon
14	Jon
15	Jon

	TotalChildren
1	5

SQLQuery2.sql - DE...3DKA\Usuario (55))*

```sql
SELECT ResellerName, ProductLine
FROM DimReseller
WHERE AnnualSales >
                    (SELECT AVG (AnnualSales)
                        FROM DimReseller)
```

100 %

Results | Messages

	ResellerName	ProductLine
1	Budget Toy Store	Mountain
2	Catalog Store	Mountain
3	Chic Department Stores	Mountain
4	Eastside Department Store	Road
5	Sports Sales and Rental	Mountain
6	Cycle Merchants	Road
7	Global Sports Outlet	Mountain
8	Exotic Bikes	Road
9	Fitness Hotel	Road
10	Healthy Activity Store	Mountain
11	Every Bike Shop	Road
12	Grand Industries	Touring
13	Ideal Components	Touring
14	Larger Cycle Shop	Road
15	Leading Sales & Repair	Road
16	Main Bicycle Services	Road
17	Metro Bike Mart	Mountain
18	Neighborhood Store	Road
19	Online Bike Catalog	Touring
20	Outdoor Equipment Store	Road

19. Realizar una consulta con subconsulta con función EXISTS o NOT EXISTS.

Seleccionar todos los productos de la categoría Bikes a través de que tengan como subproducto a Mountain Bike

```
Select  ProductCategoryKey
 From DimProductSubCategory
 Where exists  (select *  from DimProductCategory
                    where EnglishProductSubcategoryName= 'Mountain Bikes')
```

```
SELECT ProductLine, ResellerName
 FROM DimReseller
 WHERE GeographyKey < ALL (SELECT BusinessType
                           FROM DimReseller
                           WHERE YearOpened=LastOrderYear)
```

100 %

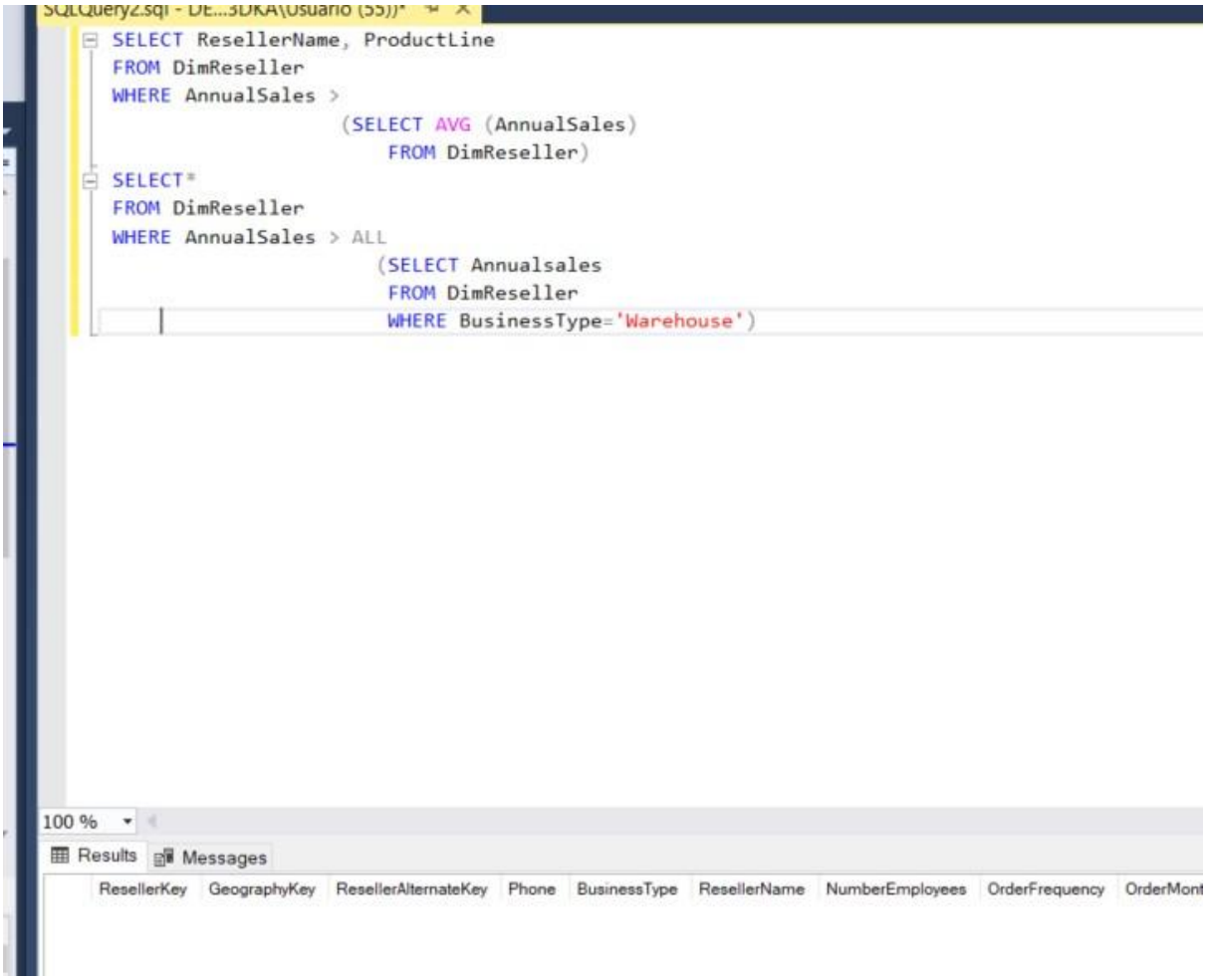
Results | Messages

	ProductLine	ResellerName
1	Road	A Bike Store
2	Mountain	Progressive Sports
3	Road	Advanced Bike Components
4	Road	Modular Cycle Systems
5	Road	Metropolitan Sports Supply
6	Road	Aerobic Exercise Company
7	Mountain	Associated Bikes
8	Touring	Exemplary Cycles
9	Mountain	Tandem Bicycle Store
10	Mountain	Rural Cycle Emporium
11	Mountain	Sharp Bikes
12	Road	Bikes and Motorbikes
13	Road	Country Parts Shop
14	Mountain	Bicycle Warehouse Inc.
15	Mountain	Budget Toy Store
16	Road	Bulk Discount Store
17	Mountain	Trusted Catalog Store
18	Mountain	Catalog Store
19	Road	Center Cycle Shop

20. Realizar una consulta con subconsulta con función ANY o ALL

Mostrar todos los tipos de negocio (BussinesType) con una venta anual (Annualsale) superior a todos los tipos de negocio de la categoría “Warehouse”

```
SELECT*
FROM DimReseller
WHERE AnnualSales > ALL
                    (SELECT Annualsales
                     FROM DimReseller
                     WHERE BusinessType='Warehouse')
```

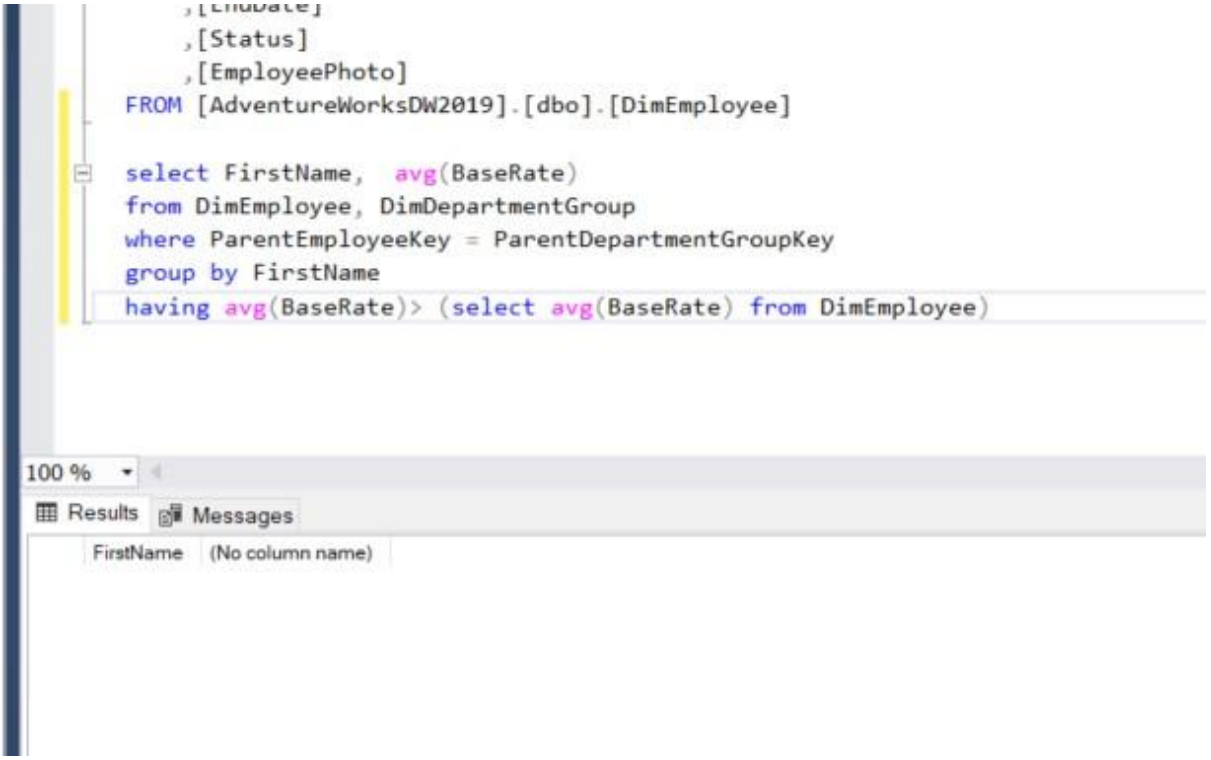


21. Realizar una consulta con subconsulta en la cláusula HAVING.

Listar los el nombre de los vendedores (FirstName) cuya BaseRate media es superior al importe de BaseRate media de todos los empleados que pertenezcan al mismo departamento.

```
SELECT FirstName,  AVG(BaseRate)  FROM DimEmployee, DimDepartmentGroup
WHERE ParentEmployeeKey = ParentDepartmentGroupKey
```

GROUP BY FirstName HAVING AVG(BaseRate) > (SELECT AVG(BaseRate)
FROM DimEmployee)



Tema 6

22. Realizar la inserción con subconsulta de una o varias filas en una tabla.

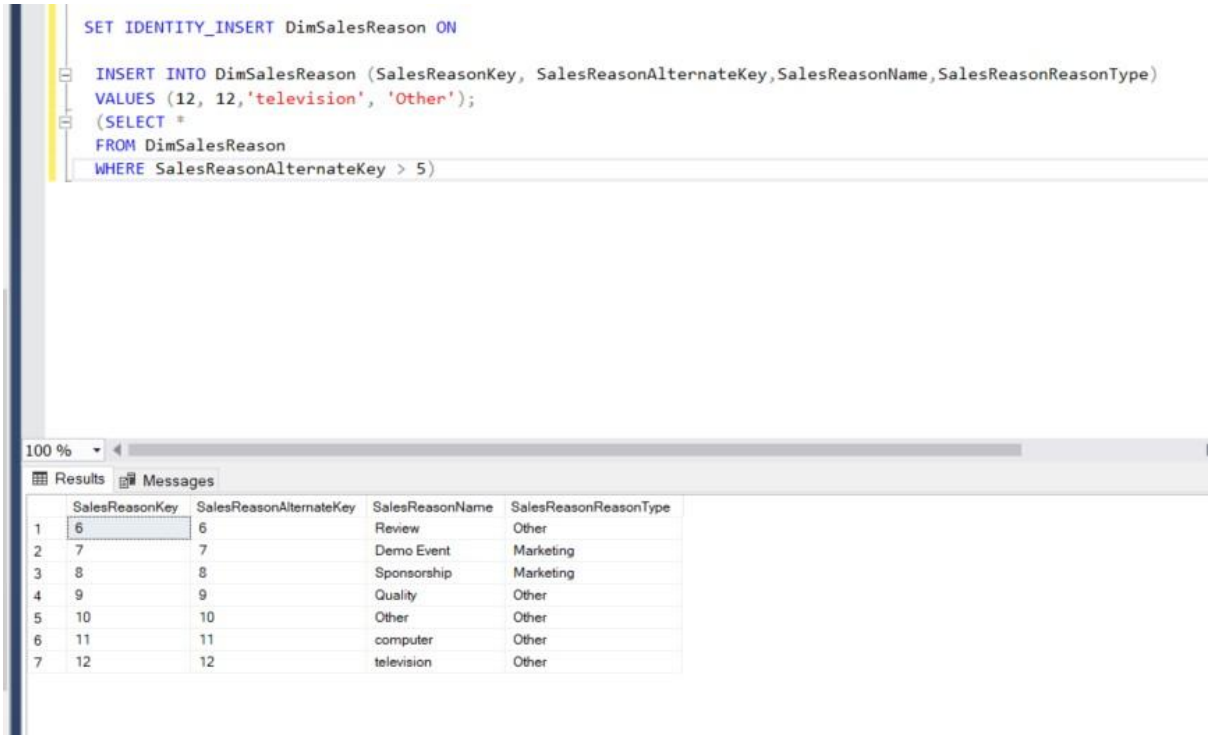
Añadir los valores (12,12, televisión, other) a todas las filas cuya SalesReasonAlternateKey sea mayor que 5

```
SET IDENTITY_INSERT DimSalesReason ON

  INSERT INTO DimSalesReason (SalesReasonKey,
SalesReasonAlternateKey,SalesReasonName,SalesReasonReasonType)
  VALUES (12, 12,'television', 'Other');

                                        (SELECT *
```

FROM DimSalesReason
WHERE SalesReasonAlternateKey > 5)



```
SET IDENTITY_INSERT DimSalesReason ON

INSERT INTO DimSalesReason (SalesReasonKey, SalesReasonAlternateKey,SalesReasonName,SalesReasonReasonType)
VALUES (12, 12,'television', 'Other');
(SELECT *
FROM DimSalesReason
WHERE SalesReasonAlternateKey > 5)
```

100 %

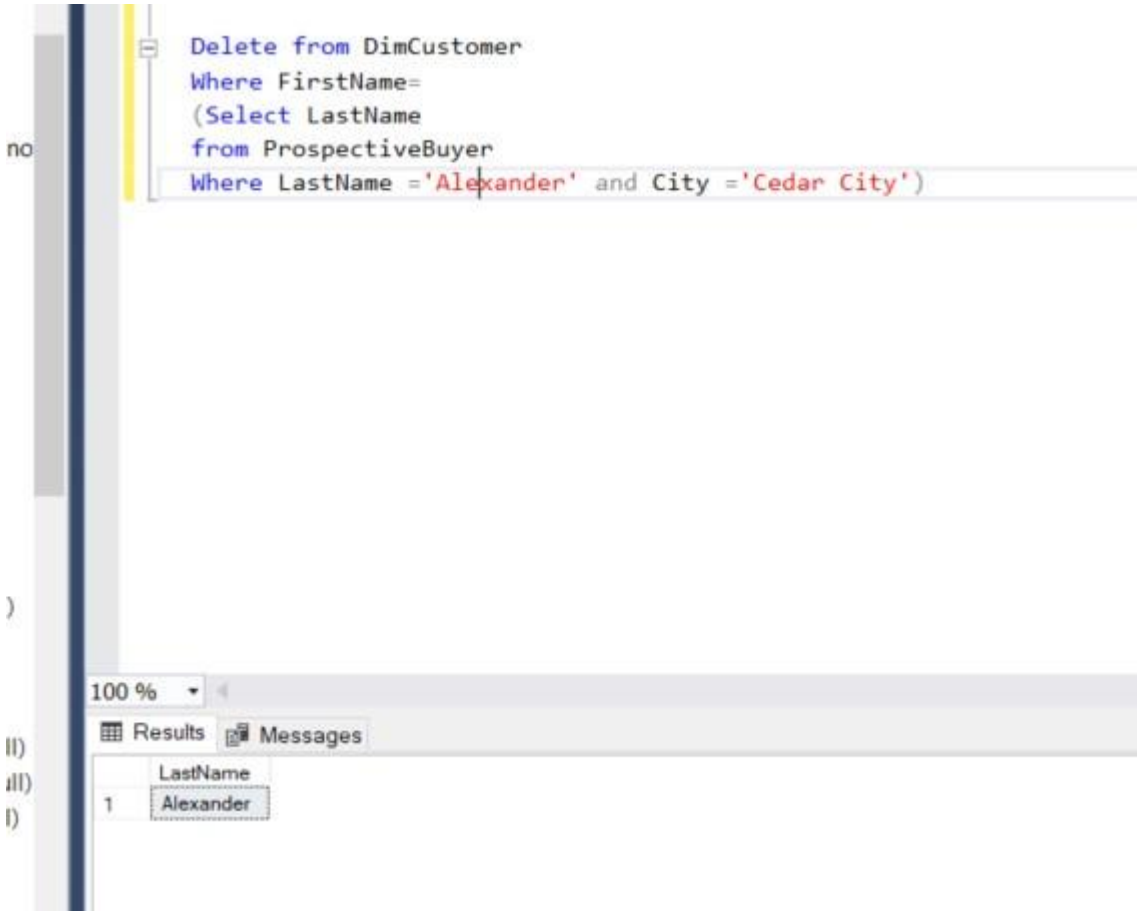
Results | Messages

	SalesReasonKey	SalesReasonAlternateKey	SalesReasonName	SalesReasonReasonType
1	6	6	Review	Other
2	7	7	Demo Event	Marketing
3	8	8	Sponsorship	Marketing
4	9	9	Quality	Other
5	10	10	Other	Other
6	11	11	computer	Other
7	12	12	television	Other

23. Realizar el borrado con subconsulta de una o varias filas en una tabla.

Borrar de la tabla DimCustomer el nombre que coincida de la tabla ProspectiveBuyer que se llame Alexander y viva en Cedar city

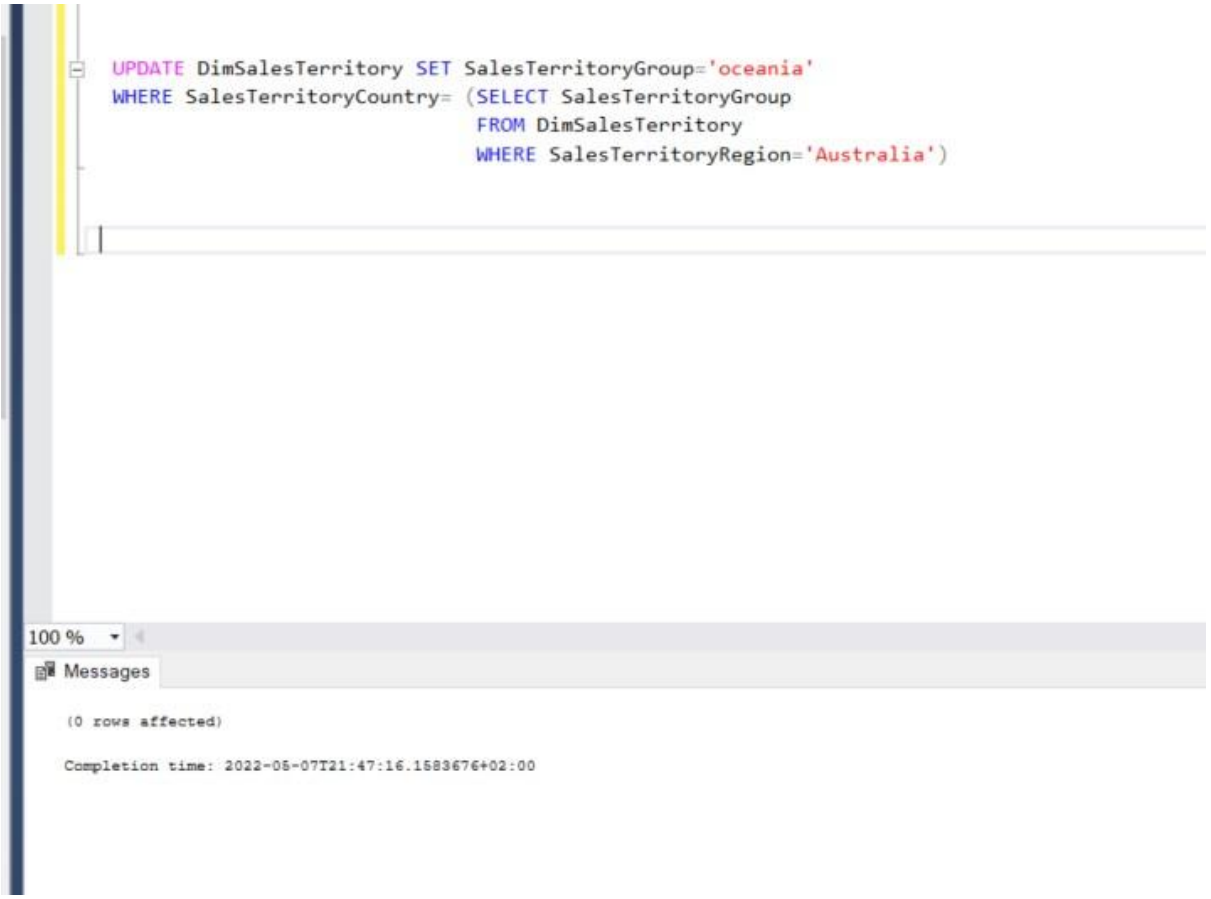
```
Delete from DimCustomer
 Where FirstName=(Select LastName
            from ProspectiveBuyer
            Where LastName ='Alexander' and City ='Cedar City')
```



24. Realizar la modificación con subconsulta de una o varias filas en una tabla.

Actualizar la SalesTerritoryKey de Australia que es 9, a la nueva que es 12

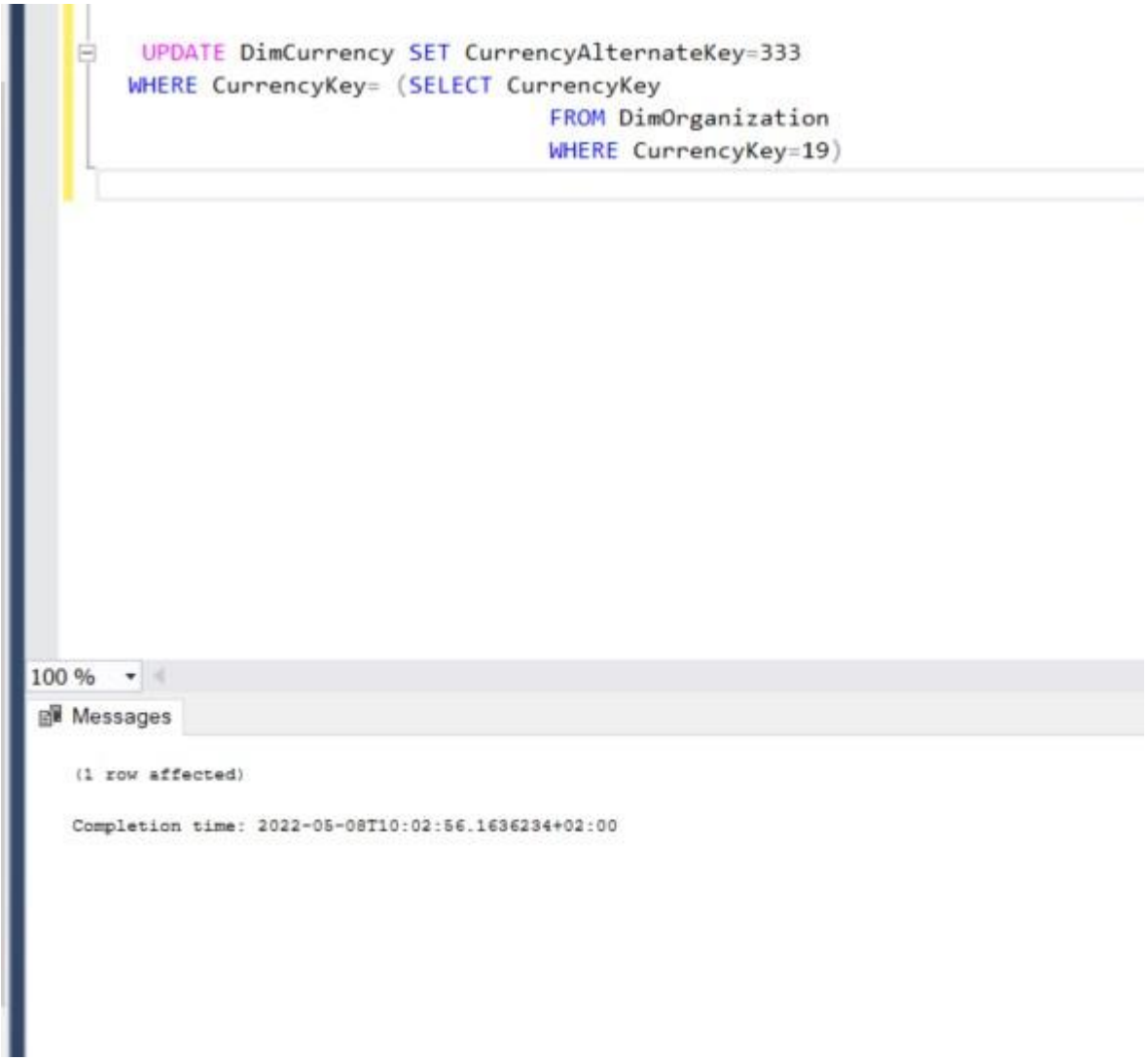
```
Update DimSalesTerretory Set SalesTerretoryGroup =’oceania’
WHERE SalesTerretoryCountry=(SELECT SalesTerrotoryGroup
                                    FROM DimSalesTerrotory
                                    WHERE SalestTerritoryRegion =’Australia’)
```



24.a) Otro

Actualizar la currentAlternateKey a 333 de la tabla DimCurrency, en todos los casos en que en la tabla DimOrganization la CurrencyKey sea 19

```
UPDATE DimCurrency SET CurrencyAlternateKey=333
 WHERE CurrencyKey= (SELECT CurrencyKey
                     FROM DimOrganization
                     WHERE CurrencyKey=19)
```



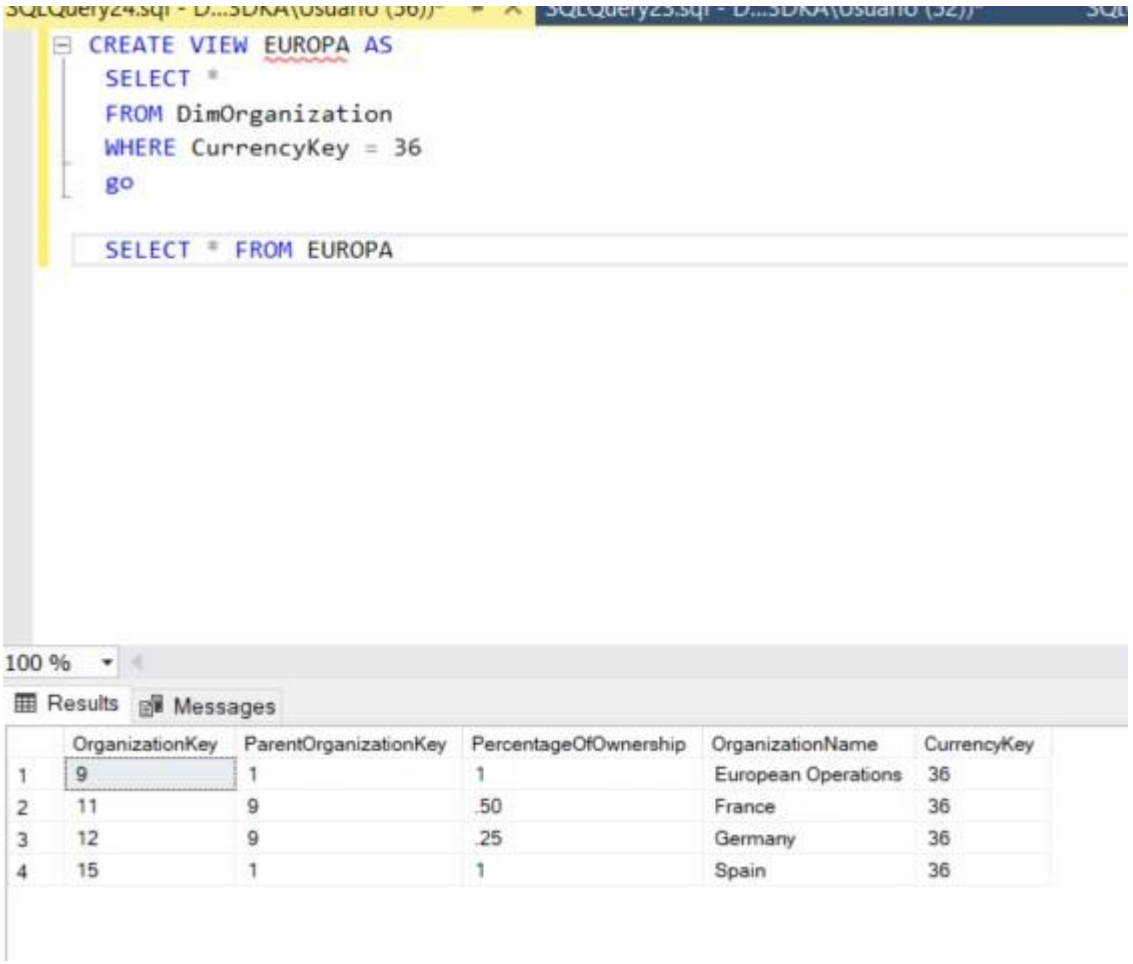
Tema 7

25. crear una vista con datos resumen de una tabla.

Crear una vista de la tabla DimOrganization que contenga los solamente países Europeos para facilitar su consulta.

```
CREATE VIEW EUROPA AS
 SELECT *
 FROM DimOrganization
 WHERE CurrencyKey = 36
 go

 SELECT * FROM EUROPA
```



	OrganizationKey	ParentOrganizationKey	PercentageOfOwnership	OrganizationName	CurrencyKey
1	9	1	1	European Operations	36
2	11	9	.50	France	36
3	12	9	.25	Germany	36
4	15	1	1	Spain	36

26. crear una vista que obtenga datos de al menos dos tablas.

Crear una vista de la tabla DimOrganization y DimCurrency que contenga solamente loa países Europeos que utilizan el Euro como moneda.

```
CREATE VIEW EUROPAIS (CurrencyName,OrganizationName)
 SELECT CurrencyName,OrganizationName
 FROM DimOrganization, DimCurrency
 WHERE CurrencyAlternateKey = 36 and CurrencyName='EURO'
```

Tema 8

27. Crear un índice en alguna de las tablas.

Crear un índice Ordinario en la tabla DimOrganization a partir del OrganizationName

CREATE INDEX MYINDICE ON DimOrganization (OrganizationName)

De Transact-SQL:

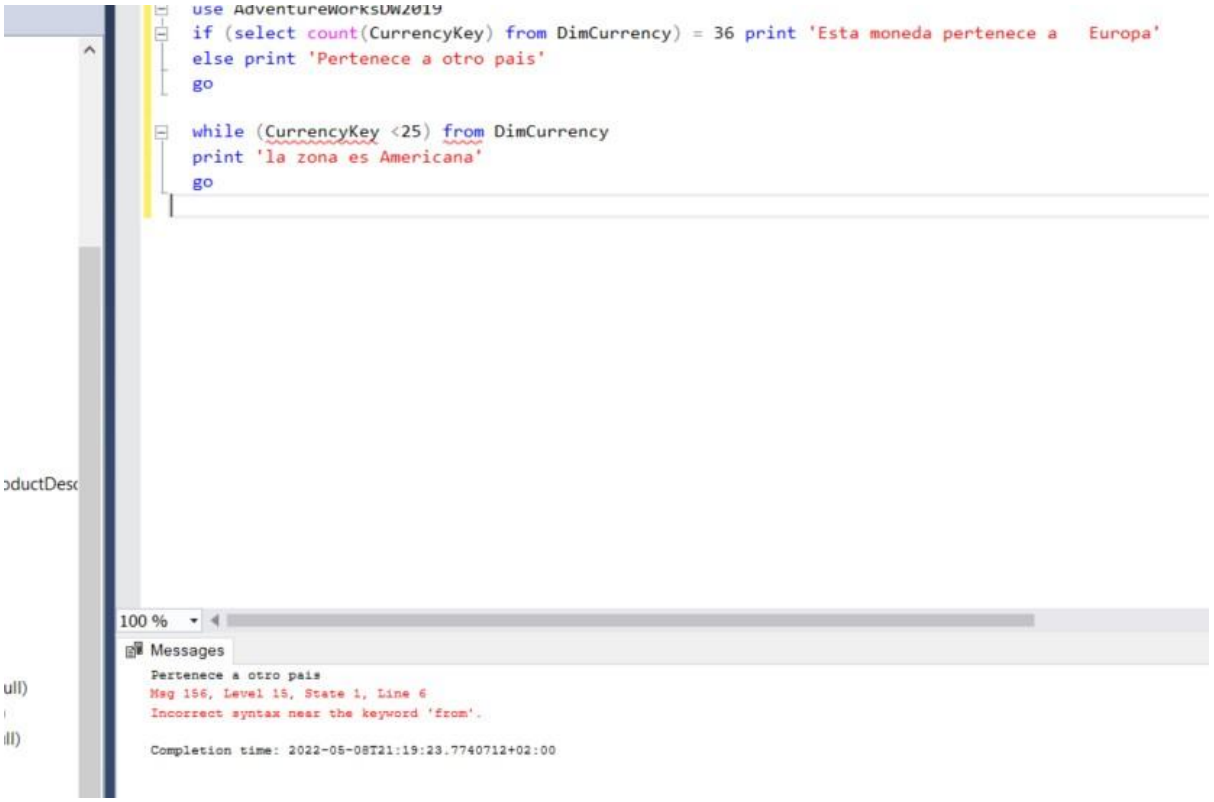
Tema 1: Programación de guiones

28. Realizar un script o guión en el que se utilice al menos una sentencia WHILE y una IF.

use AdventureWorksDW2019

```
 if (select count(CurrencyKey) from DimCurrency) = 36 print 'Esta moneda pertenece a
Europa'
 else print 'Pertenece a otro pais'
 go

 while (CurrencyKey <25) from DimCurrency
 print 'la zona es Americana'
 go
```



29. Realizar un script o guión en el que se utilice al menos una expresión CASE.

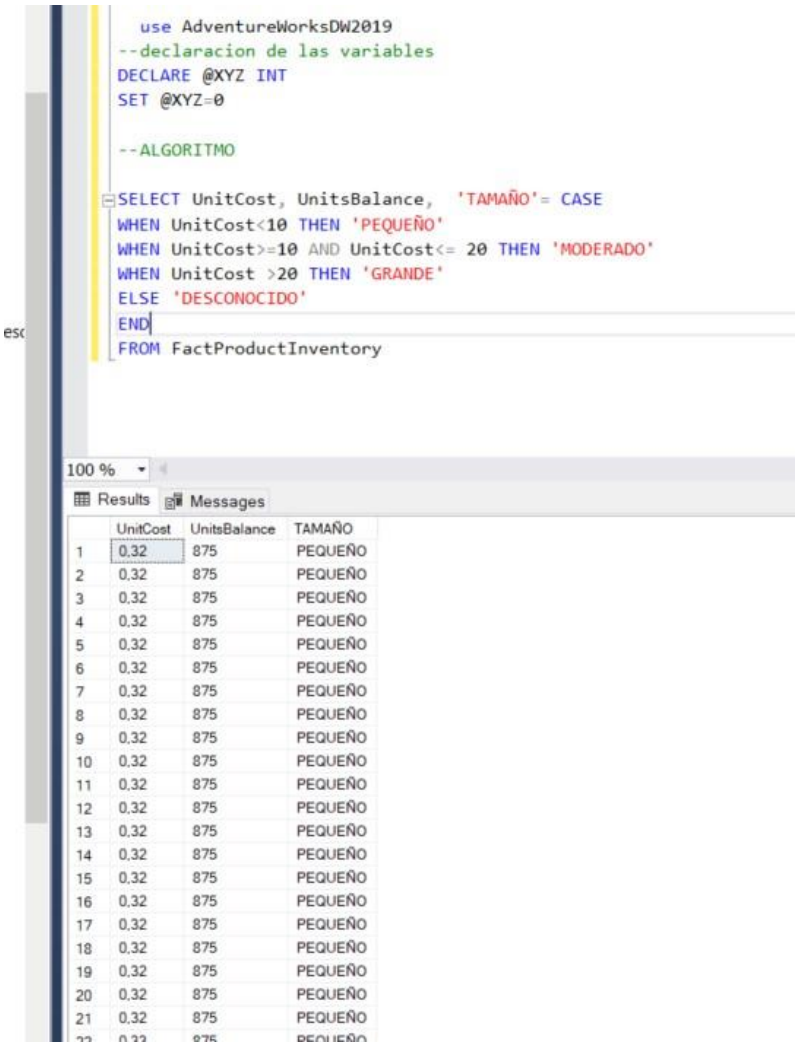
Mostrar el precio UnitCost, y las UnitsBalance clasificados por tamaño:
Si el UnitCost es menor que 10 mostrar ‘pequeño’
Si el UnitCost es mayor que 10 y menor que 20 mostrar ‘moderado’
Si el UnitCost es mayor que 20 mostrar ‘grande’
y si no ‘desconocido’

```
 use AdventureWorksDW2019

--declaración e inicialización de las variables por si las necesitamos
DECLARE @XYZ INT
SET @XYZ=0
```

--ALGORITMO

```
SELECT UnitCost, UnitsBalance,  'TAMAÑO'= CASE
WHEN UnitCost<10 THEN 'PEQUEÑO'
WHEN UnitCost>=10 AND UnitCost<= 20 THEN 'MODERADO'
WHEN UnitCost >20 THEN 'GRANDE'
ELSE 'DESCONOCIDO'
END
FROM FactProductInventory
```



	UnitCost	UnitsBalance	TAMAÑO
1	0,32	875	PEQUEÑO
2	0,32	875	PEQUEÑO
3	0,32	875	PEQUEÑO
4	0,32	875	PEQUEÑO
5	0,32	875	PEQUEÑO
6	0,32	875	PEQUEÑO
7	0,32	875	PEQUEÑO
8	0,32	875	PEQUEÑO
9	0,32	875	PEQUEÑO
10	0,32	875	PEQUEÑO
11	0,32	875	PEQUEÑO
12	0,32	875	PEQUEÑO
13	0,32	875	PEQUEÑO
14	0,32	875	PEQUEÑO
15	0,32	875	PEQUEÑO
16	0,32	875	PEQUEÑO
17	0,32	875	PEQUEÑO
18	0,32	875	PEQUEÑO
19	0,32	875	PEQUEÑO
20	0,32	875	PEQUEÑO
21	0,32	875	PEQUEÑO

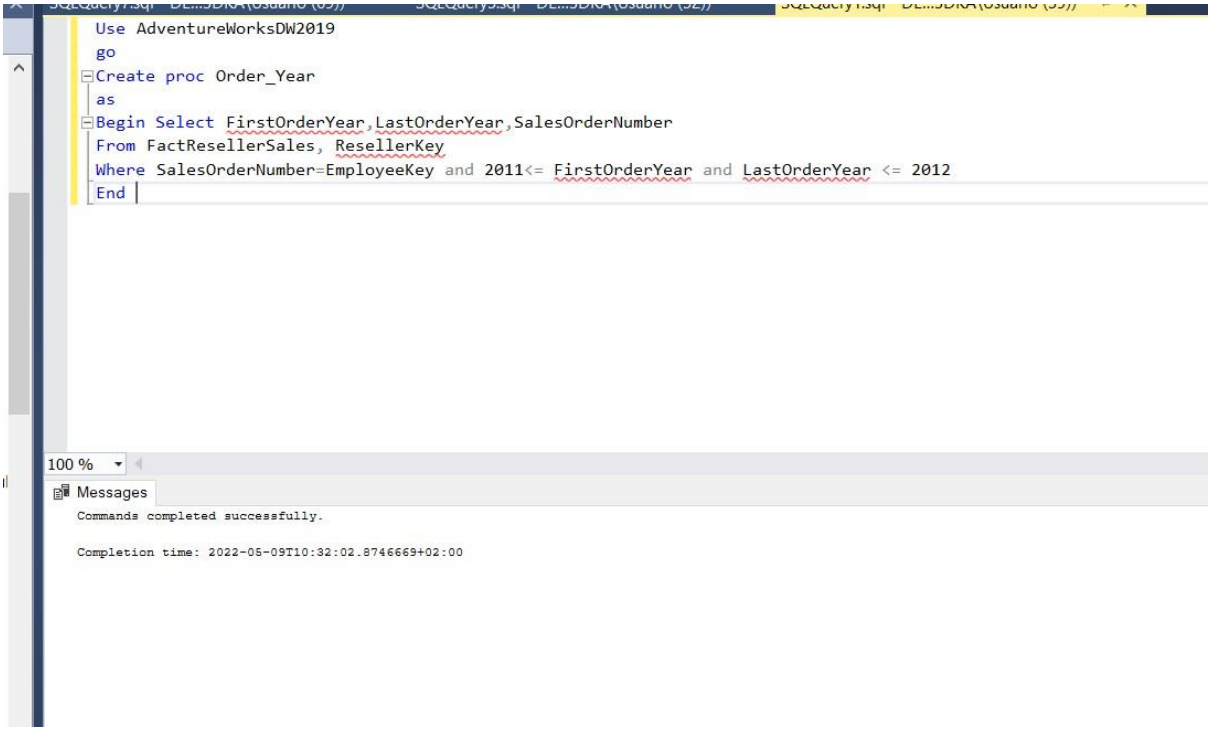
Tema 2: Módulos: procedimientos y funciones

30. Crear un procedimiento almacenado sin parámetros. Probar su funcionamiento.

Usando la sentencia para encontrar los números de SalesOrderNumber entre los años de la primera y última order, crear un procedimiento sin parámetros

```
Use AdventureWorksDW2019
go
Create proc Order_Year
as
Begin
   Select FirstOrderYear,LastOrderYear,SalesOrderNumber
   From FactResellerSales, ResellerKey
   Where SalesOrderNumber=EmployeeKey and 2011<= FirstOrderYear and LastOrderYear
<= 2012
End

execute Order_Year
```



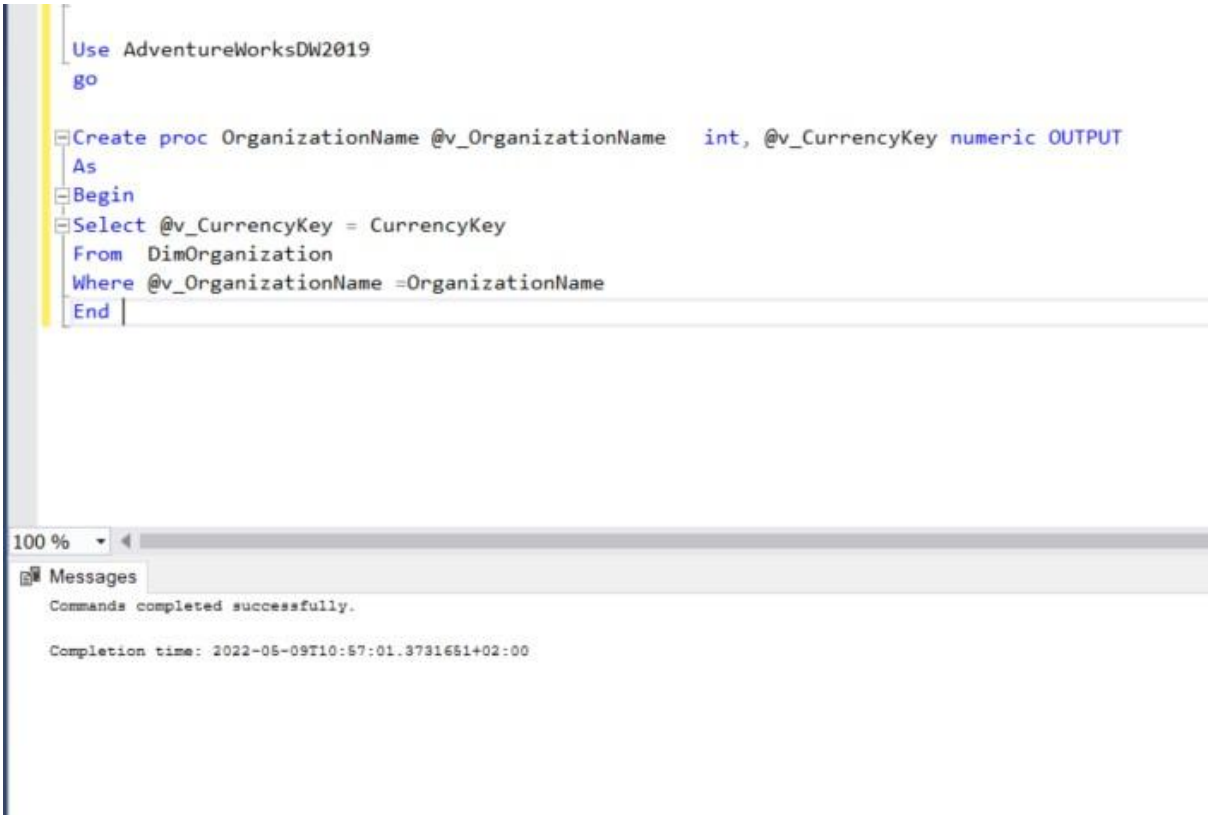
31. Crear un procedimiento almacenado con varios parámetros de entrada, en el que al menos uno sea opcional. Probar su funcionamiento.

Procedimiento almacenado al que se le pasamos como parámetro un ‘OrganizationName’ y nos devuelve la ‘CurrencyKey’ de ese Pais.

```
Use AdventureWorksDW2019
go

Create proc OrganizationName @v_OrganizationName   int, @v_CurrencyKey numeric
OUTPUT
As
Begin
Select @v_CurrencyKey = CurrencyKey
From  DimOrganization
Where @v_OrganizationName =OrganizationName
End

Declare @CurrencyKey numeric
Declare @@v_OrganizationName
Exec OrganizationName 'Spain', @CurrencyKey output
Select  @CurrencyKey
```



32. Crear un procedimiento almacenado con varios parámetros de entrada y al menos dos de salida. Probar su funcionamiento.

33. Crear una función sin parámetros. Probar su funcionamiento.

34. Hacer un script en el que se utilice la función anterior para obtener algún informe o dentro de alguna sentencia.

35. Crear una función que reciba al menos dos parámetros de entrada y devuelva un valor escalar. Probar su funcionamiento.

36. Hacer un script en el que se utilice la función anterior para obtener algún informe o dentro de alguna sentencia.

37. Crear una función que reciba al menos un parámetro de entrada y devuelva una tabla. Probar su funcionamiento.

38. Hacer un script en el que se utilice la función anterior para obtener algún informe o dentro de alguna sentencia.