



GESTIÓN DE DATOS

Consulta usando funciones integradas.
Consultas con JOINS. Group By. Having.

Introducción

- Quiz



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"



Poll Everywhere



Introducción

- Transact SQL (Structured Query Language)
- Categorías de las sentencias SQL.
- DDL Lenguaje de base de datos. y DML Lenguaje de manipulación de datos.
- Consultas condicionales.
- Operadores lógicos.
- Ejercicios prácticos.

Consultando la información de Northwind

- Para consultar los datos que están almacenados en una Base de datos se utiliza el lenguaje de manipulación de datos **SQL**
- Para obtener la información de las compañías que envían los paquetes se ejecuta el siguiente comando o **query**

```
select ShipperID, CompanyName, Phone  
from Shippers
```

	ShipperID	CompanyName	Phone
1	1	Speedy Express	(503) 555-9831
2	2	United Package	(503) 555-3199
3	3	Federal Shipping	(503) 555-9931

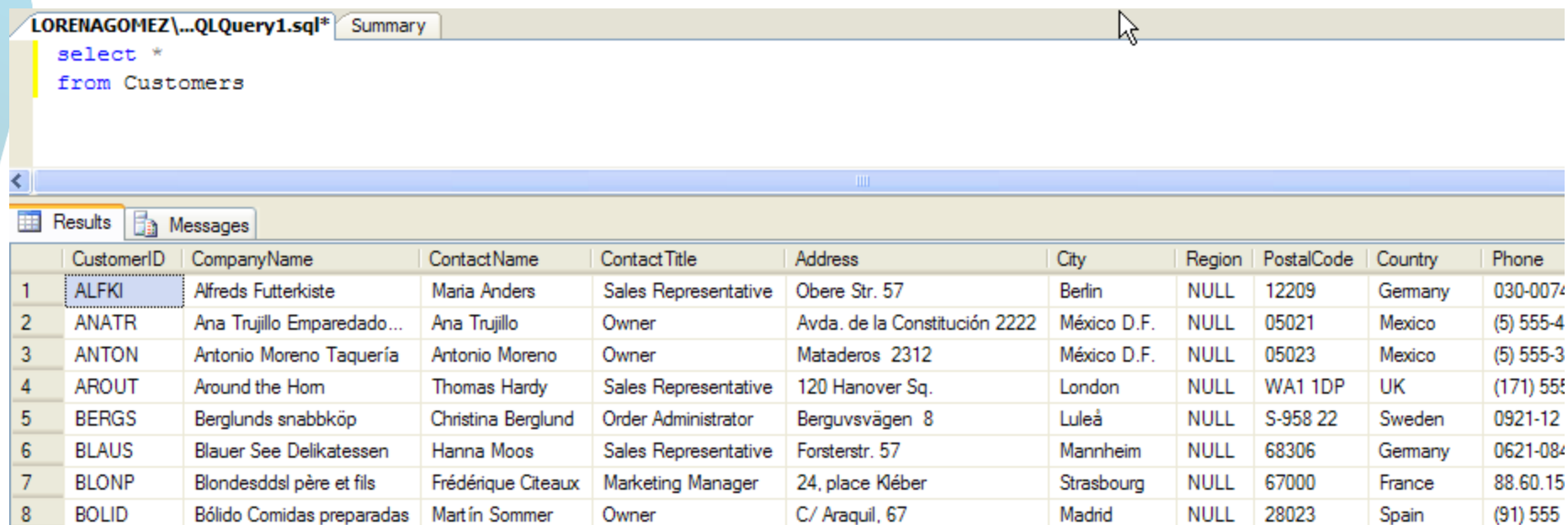
Atributos

Tabla



Ejemplo usando Select *

- La consulta más compacta es obtener **todos** los **atributos** de una **tabla**
- Por ejemplo para obtener toda la información de los **clientes**, utilizamos el siguiente query
- El **(*)** **asterisco** indica que se deben obtener todos los atributos del cliente:
select * from Customers



LORENAGOMEZ\...QLQuery1.sql* Summary

```
select *  
from Customers
```

Results Messages

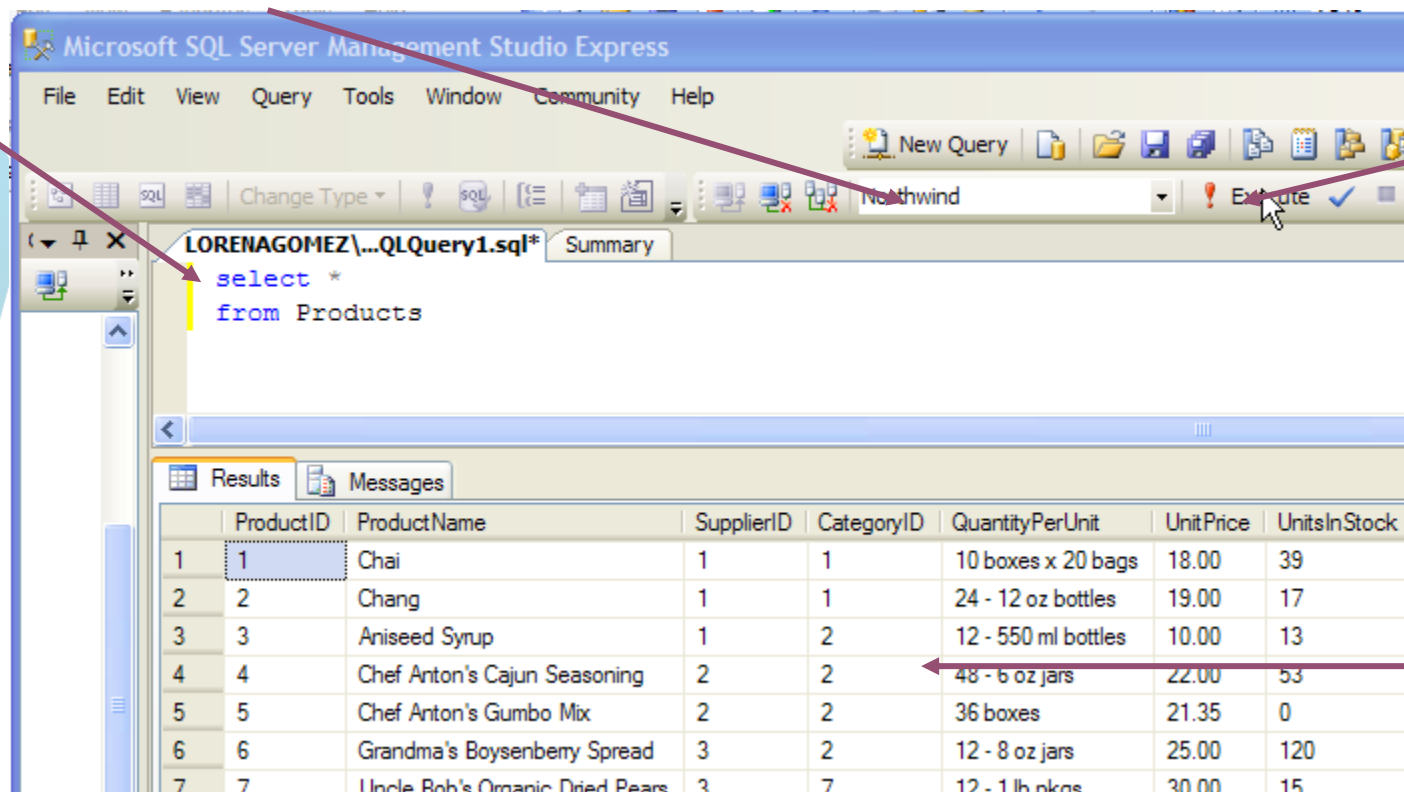
	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone
1	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	NULL	12209	Germany	030-0074
2	ANATR	Ana Trujillo Emparedado...	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	NULL	05021	Mexico	(5) 555-4
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	NULL	05023	Mexico	(5) 555-3
4	AROUT	Around the Hom	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	NULL	WA1 1DP	UK	(171) 555
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	NULL	S-958 22	Sweden	0921-12
6	BLAUS	Blauer See Delikatessen	Hanna Moos	Sales Representative	Forsterstr. 57	Mannheim	NULL	68306	Germany	0621-084
7	BLONP	Blondesddsl père et fils	Frédérique Citeaux	Marketing Manager	24, place Kléber	Strasbourg	NULL	67000	France	88.60.15
8	BOLID	Bólide Comidas preparadas	Martín Sommer	Owner	C/ Araquil, 67	Madrid	NULL	28023	Spain	(91) 555

Ejecutando las consultas

- Utilizando Microsoft SQL Server Management Studio:

Query Base de datos utilizada

Ejecutar query



The screenshot shows the Microsoft SQL Server Management Studio Express interface. The query editor at the top contains the following SQL query:

```
select *  
from Products
```

The query is executed, and the results are displayed in the Results pane below. The results are shown in a table with the following columns: ProductID, ProductName, SupplierID, CategoryID, QuantityPerUnit, UnitPrice, and UnitsInStock. The table contains 7 rows of data.

	ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock
1	1	Chai	1	1	10 boxes x 20 bags	18.00	39
2	2	Chang	1	1	24 - 12 oz bottles	19.00	17
3	3	Aniseed Syrup	1	2	12 - 550 ml bottles	10.00	13
4	4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22.00	53
5	5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.35	0
6	6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25.00	120
7	7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs	30.00	15

Resultados

SQL: STRUCTURED QUERY LANGUAGE



Universidad
Nacional de
Cajamarca

"Norte de la Universidad Peruana"

- Los ejemplos anteriores muestran un ejemplo sencillo del SQL, para hacer consultas más elaboradas se requiere conocer la **sintaxis** del **SQL**:

```
SELECT    [DISTINCT | ALL]  
           { * | [columnExpression [AS newName]] [,...] }  
FROM      TableName [alias] [, ...]  
[WHERE     condition]  
[GROUP BY  columnList]  
           [HAVING   condition]  
[ORDER BY  columnList]
```



LA SENTENCIA SELECT

SELECT

Especifica las columnas a mostrar en el resultado

FROM

Especifica la(s) tabla(s) a utilizar.

WHERE

Filtra renglones de acuerdo a la condición especificada.

GROUP BY

Forma grupos de renglones con el mismo valor en la columna

HAVING

Filtra grupos de acuerdo a la condición especificada.

ORDER BY

Especifica como ordenar el resultado.

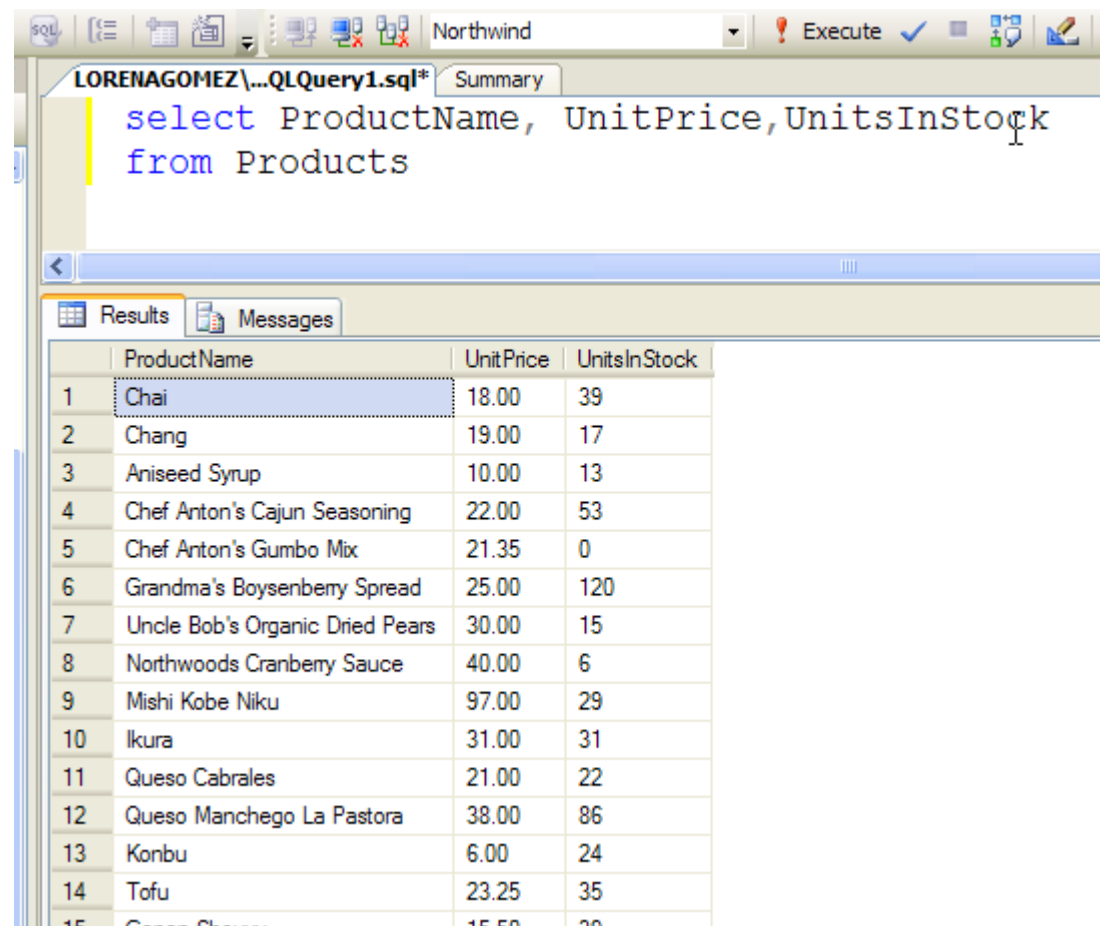
Mostrar solo algunos atributos

- Obtener solo algunos atributos y todos los registros de una tabla

SELECT

A_1, A_2, \dots, A_n

FROM r_1



The screenshot shows a SQL Server Enterprise Manager window with a query executed. The query is:

```
select ProductName, UnitPrice, UnitsInStock  
from Products
```

The results are displayed in a table with the following columns: ProductName, UnitPrice, and UnitsInStock. The table contains 15 rows of data.

	ProductName	UnitPrice	UnitsInStock
1	Chai	18.00	39
2	Chang	19.00	17
3	Aniseed Syrup	10.00	13
4	Chef Anton's Cajun Seasoning	22.00	53
5	Chef Anton's Gumbo Mix	21.35	0
6	Grandma's Boysenberry Spread	25.00	120
7	Uncle Bob's Organic Dried Pears	30.00	15
8	Northwoods Cranberry Sauce	40.00	6
9	Mishi Kobe Niku	97.00	29
10	Ikura	31.00	31
11	Queso Cabrales	21.00	22
12	Queso Manchego La Pastora	38.00	86
13	Konbu	6.00	24
14	Tofu	23.25	35
15	Genen Shoyu	15.50	30

Evitar tuplas duplicadas

Mostrar en que ciudades
hay clientes

Se repiten las ciudades

Usar **DISTINCT**

No se repiten

LORENAGOMEZ\...QLQuery1.sql*

```
select City  
from Customers
```

Results Messages

	City
1	Aachen
2	Albuquerque
3	Anchorage
4	Århus
5	Barcelona
6	Barquisimeto
7	Bergamo
8	Berlin
9	Bem
10	Boise
11	Bräcke
12	Brandenburg
13	Bruxelles
14	Buenos Aires
15	Buenos Aires
16	Buenos Aires
17	Butte
18	Campinas

LORENAGOMEZ\...QLQuery1.sql* Summary

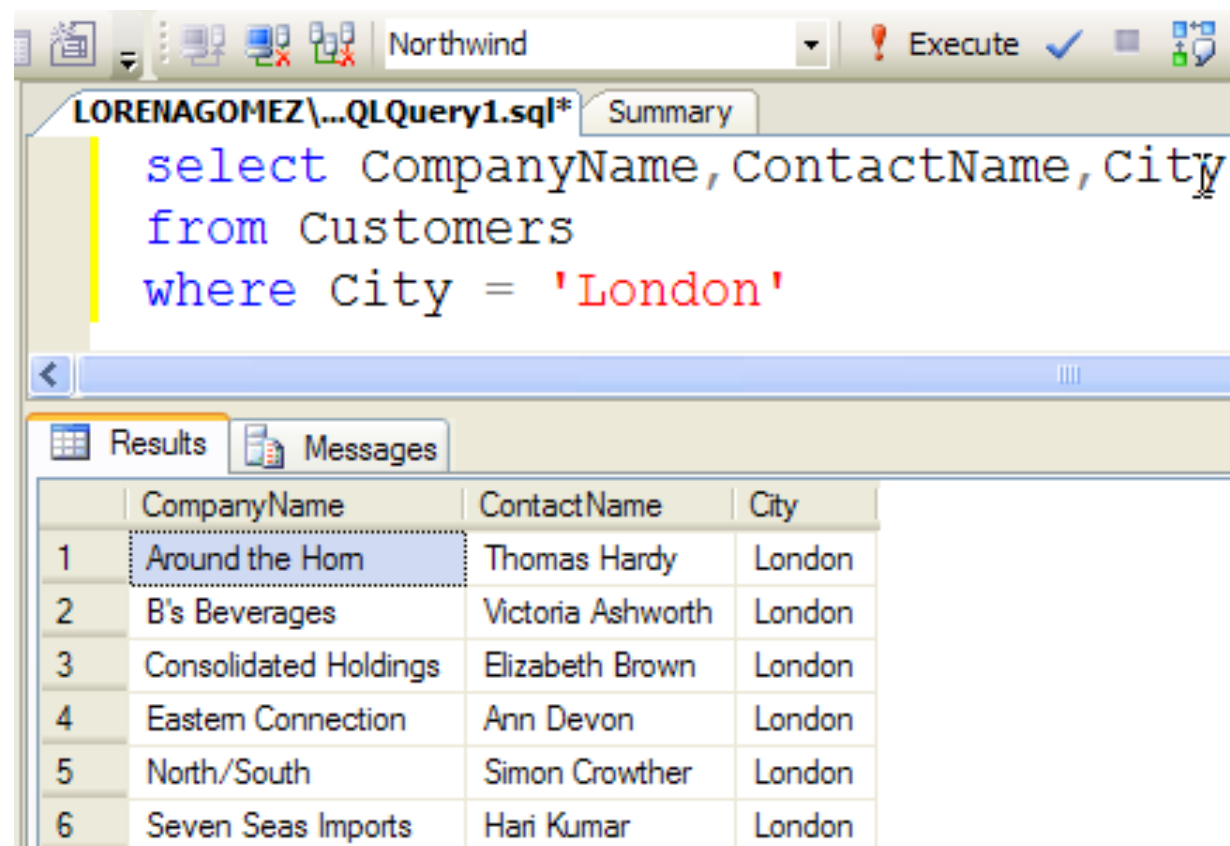
```
select distinct City  
from Customers
```

Results Messages

	City
1	Aachen
2	Albuquerque
3	Anchorage
4	Århus
5	Barcelona
6	Barquisimeto
7	Bergamo
8	Berlin
9	Bem
10	Boise
11	Bräcke
12	Brandenburg
13	Bruxelles
14	Buenos Aires
15	Butte
16	Campinas
17	Caracas
18	Charleroi
19	Cork

Especificando una condición

- Obtener nombre de compañía y del contacto de los clientes que viven en **London**



The screenshot shows a SQL query window titled "LORENAGOMEZ\...QLQuery1.sql*" with a "Summary" tab. The query is:

```
select CompanyName, ContactName, City
from Customers
where City = 'London'
```

Below the query, the "Results" tab is active, displaying a table with 6 rows and 4 columns: CompanyName, ContactName, and City. The first row is highlighted.

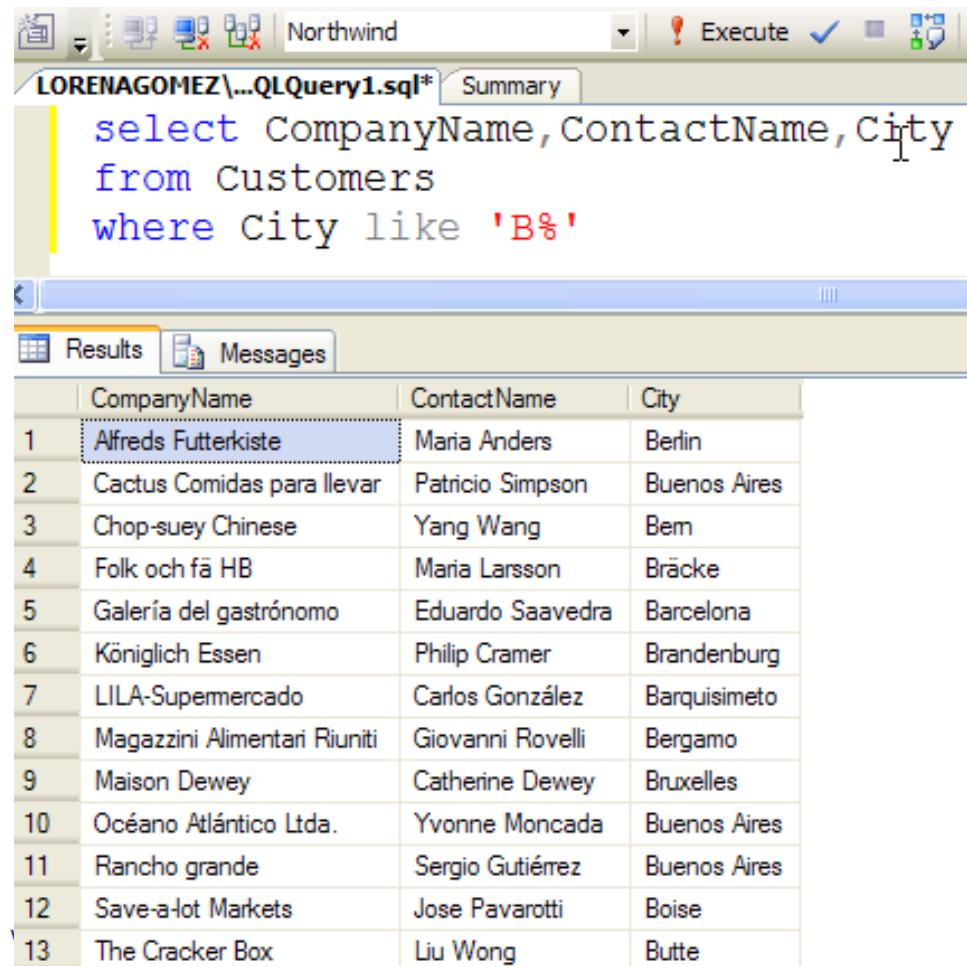
	CompanyName	ContactName	City
1	Around the Horn	Thomas Hardy	London
2	B's Beverages	Victoria Ashworth	London
3	Consolidated Holdings	Elizabeth Brown	London
4	Eastern Connection	Ann Devon	London
5	North/South	Simon Crowther	London
6	Seven Seas Imports	Hari Kumar	London

Especificando una condición con patrones

- Obtener nombre de compañía y del contacto de los clientes cuya ciudad comienza con B

patrones

- %
 - Ignora los valores de cero o más caracteres
- _
 - Ignora el valor de un caracter



The screenshot shows the Microsoft Access interface. The top window displays a SQL query in the 'LORENAGOMEZ\...\QLQuery1.sql*' file. The query is:

```
select CompanyName, ContactName, City
from Customers
where City like 'B%'
```

The bottom window shows the 'Results' tab with a table of 13 rows. The first row is highlighted.

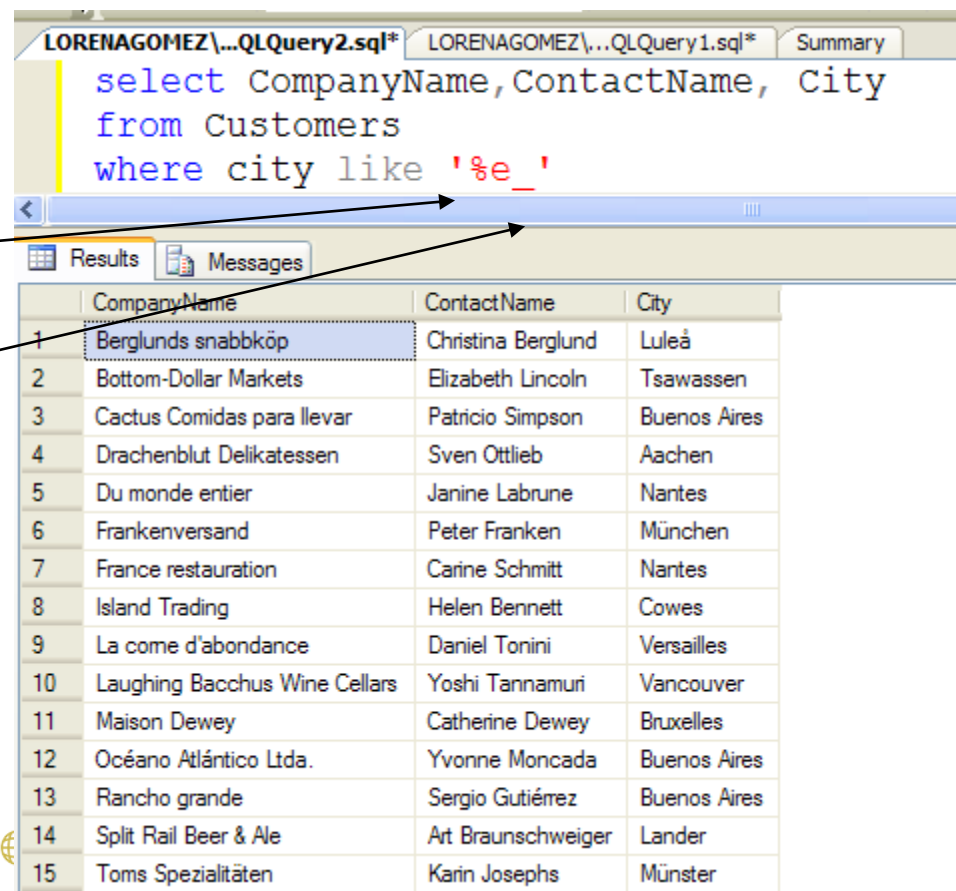
	CompanyName	ContactName	City
1	Alfreds Futterkiste	Maria Anders	Berlin
2	Cactus Comidas para llevar	Patricio Simpson	Buenos Aires
3	Chop-suey Chinese	Yang Wang	Bem
4	Folk och fä HB	Maria Larsson	Bräcke
5	Galería del gastrónomo	Eduardo Saavedra	Barcelona
6	Königlich Essen	Philip Cramer	Brandenburg
7	LILA-Supernmercado	Carlos González	Barquisimeto
8	Magazzini Alimentari Riuniti	Giovanni Rovelli	Bergamo
9	Maison Dewey	Catherine Dewey	Bruxelles
10	Océano Atlántico Ltda.	Yvonne Moncada	Buenos Aires
11	Rancho grande	Sergio Gutiérrez	Buenos Aires
12	Save-a-lot Markets	Jose Pavarotti	Boise
13	The Cracker Box	Liu Wong	Butte

Especificando una condición con patrones

- Obtener nombre de compañía y del contacto de los clientes cuya ciudad tenga como penultima letra una **e**

% cero o más caracteres

_ 1 sólo carácter



```
LORENAGOMEZ\...QLQuery2.sql* LORENAGOMEZ\...QLQuery1.sql* Summary
select CompanyName, ContactName, City
from Customers
where city like '%e_'
```

	CompanyName	ContactName	City
1	Berglunds snabbköp	Christina Berglund	Luleå
2	Bottom-Dollar Markets	Elizabeth Lincoln	Tsawassen
3	Cactus Comidas para llevar	Patricio Simpson	Buenos Aires
4	Drachenblut Delikatessen	Sven Ottlieb	Aachen
5	Du monde entier	Janine Labrune	Nantes
6	Frankenversand	Peter Franken	München
7	France restauration	Carine Schmitt	Nantes
8	Island Trading	Helen Bennett	Cowes
9	La come d'abondance	Daniel Tonini	Versailles
10	Laughing Bacchus Wine Cellars	Yoshi Tannamuri	Vancouver
11	Maison Dewey	Catherine Dewey	Bruxelles
12	Océano Atlántico Ltda.	Yvonne Moncada	Buenos Aires
13	Rancho grande	Sergio Gutiérrez	Buenos Aires
14	Split Rail Beer & Ale	Art Braunschweiger	Lander
15	Toms Spezialitäten	Karin Josephs	Münster



Especificando varias condiciones (OR)

- Obtener nombre de compañía y del contacto de los clientes cuya ciudad sea **London** o **Buenos Aires**

LORENAGOMEZ\...QLQuery1.sql* Summary

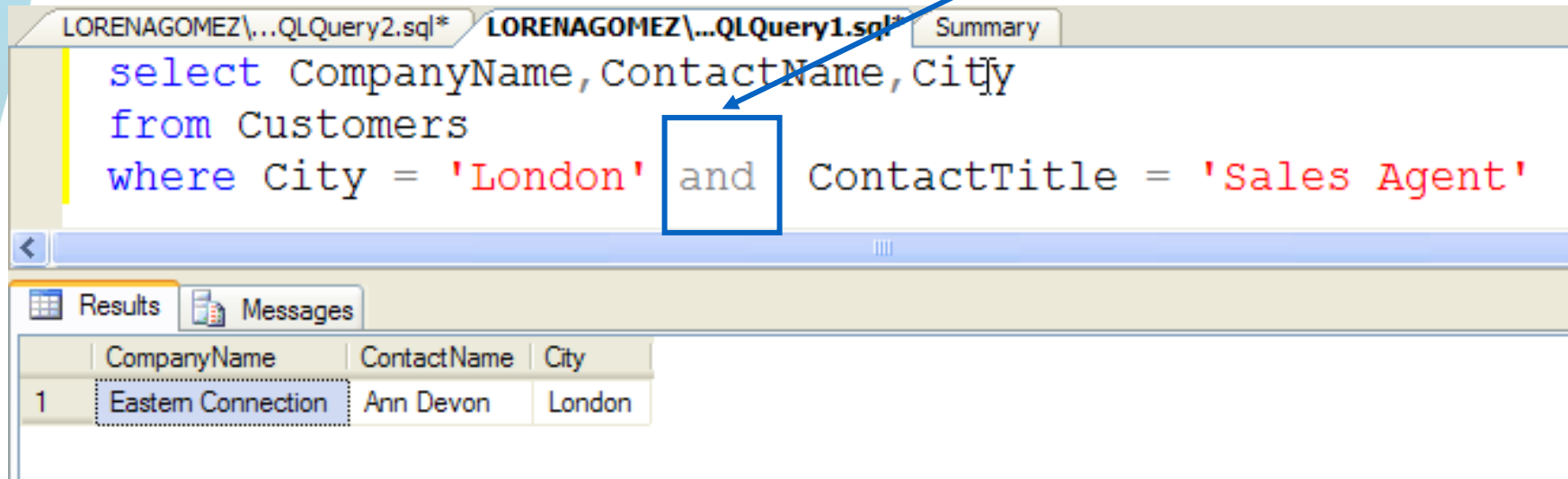
```
select CompanyName, ContactName, City
from Customers
where City = 'London' or City = 'Buenos Aires'
```

Results Messages

	CompanyName	ContactName	City
1	Around the Horn	Thomas Hardy	London
2	B's Beverages	Victoria Ashworth	London
3	Cactus Comidas para llevar	Patricio Simpson	Buenos Aires
4	Consolidated Holdings	Elizabeth Brown	London
5	Eastern Connection	Ann Devon	London
6	North/South	Simon Crowther	London
7	Océano Atlántico Ltda.	Yvonne Moncada	Buenos Aires
8	Rancho grande	Sergio Gutiérrez	Buenos Aires
9	Seven Seas Imports	Hari Kumar	London

Especificando varias condiciones (AND)

- Obtener nombre de compañía y del contacto de los clientes cuya ciudad sea **London** y que la persona contacto sea un **Agente de ventas** (Sales Agent)



```
LORENAGOMEZ\...QLQuery2.sql* LORENAGOMEZ\...QLQuery1.sql* Summary
select CompanyName, ContactName, City
from Customers
where City = 'London' and ContactTitle = 'Sales Agent'
```

Results Messages

	CompanyName	ContactName	City
1	Eastern Connection	Ann Devon	London



Uso de Conjuntos en una Condición (IN)

- Obtener nombre de compañía, nombre y título del contacto de los clientes cuya persona contacto sea un **Agente de ventas (Sales Agent)** o un representante de ventas (**Sales Representative**) o un asociado de ventas (**Sales Associate**)

```
LORENAGOMEZ\...QLQuery2.sql* LORENAGOMEZ\...QLQuery1.sql* Summary
select CompanyName,ContactName,City, ContactTitle
from Customers
where ContactTitle in
( 'Sales Agent', 'Sales Associate', 'Sales Representative' )
```

	CompanyName	ContactName	City	Contact Title
1	Alfreds Futterkiste	Maria Anders	Berlin	Sales Representative
2	Around the Horn	Thomas Hardy	London	Sales Representative
3	Blauer See Delikatessen	Hanna Moos	Mannheim	Sales Representative
4	B's Beverages	Victoria Ashworth	London	Sales Representative
5	Cactus Comidas para llevar	Patricio Simpson	Buenos Aires	Sales Agent
6	Comércio Mineiro	Pedro Afonso	Sao Paulo	Sales Associate
7	Consolidated Holdings	Elizabeth Brown	London	Sales Representative
8	Eastern Connection	Ann Devon	London	Sales Agent
9	Franchi S.p.A.	Paolo Accorti	Torino	Sales Representative
10	Gourmet Lanchonetes	André Fonseca	Campinas	Sales Associate
11	HILARION-Abastos	Carlos Hernández	San Cristóbal	Sales Representative
12	Hungry Coyote Import Store	Yoshi Latimer	Elgin	Sales Representative
13	Hunny Owl All-Night Grocers	Patricia McKenna	Cork	Sales Associate



Conjuntos en una condición (NOT IN)

- Obtener nombre de compañía, nombre y título del contacto de los clientes cuya persona contacto **NO sea** un **Agente de ventas (Sales Agent)** o un representante de ventas (**Sales Representative**) o un asociado de ventas (**Sales Associate**)

```
select CompanyName, ContactName, City, ContactTitle
from Customers
where ContactTitle NOT IN
( 'Sales Agent', 'Sales Associate', 'Sales Representative' )
```

Results

	CompanyName	ContactName	City	ContactTitle
1	Ana Trujillo Emparedados y helados	Ana Trujillo	México D.F.	Owner
2	Antonio Moreno Taquería	Antonio Moreno	México D.F.	Owner
3	Berglunds snabbköp	Christina Berglund	Luleå	Order Administrator
4	Blondesddsl père et fils	Frédérique Citeaux	Strasbourg	Marketing Manager
5	Bóldo Comidas preparadas	Martín Sommer	Madrid	Owner
6	Bon app'	Laurence Lebihan	Marseille	Owner
7	Bottom-Dollar Markets	Elizabeth Lincoln	Tsawassen	Accounting Manager
8	Centro comercial Moctezuma	Francisco Chang	México D.F.	Marketing Manager
9	Chop-suey Chinese	Yang Wang	Bem	Owner
10	Drachenblut Delikatessen	Sven Ottlieb	Aachen	Order Administrator



Verificando valores NULL

- Obtener **Id del cliente**, nombre de la **compañía** y la **Region** para aquellos clientes cuya **Region** sea **NULL**

Null

Valor que significa
que al atributo no
se le asignó un valor
o se le asignó NULL

LORENAGOMEZ\...QLQuery2.sql* LORENAGOMEZ\...QLQuery1.sql* Summary

```
select CustomerID, CompanyName, Region  
from Customers  
where Region is NULL
```

Results Messages

	CustomerID	CompanyName	Region
1	ALFKI	Alfreds Futterkiste	NULL
2	ANATR	Ana Trujillo Emparedados y helados	NULL
3	ANTON	Antonio Moreno Taquería	NULL
4	AROUT	Around the Hom	NULL
5	BERGS	Berglunds snabbköp	NULL
6	BLAUS	Blauer See Delikatessen	NULL
7	BLONP	Blondesddsl père et fils	NULL
8	BOLID	Bólide Comidas preparadas	NULL
9	BONAP	Bon app'	NULL
10	BSBEV	B's Beverages	NULL
11	CACTU	Cactus Comidas para llevar	NULL
12	CENTC	Centro comercial Moctezuma	NULL



Verificando valores NOT NULL

- Obtener **Id del cliente**, nombre de la **compañía** y la **Region** para aquellos clientes cuya **Region** tenga un valor asignado

LORENAGOMEZ\...QLQuery2.sql* LORENAGOMEZ\...QLQuery1.sql* Summary

```
select CustomerID,CompanyName, Region  
from Customers  
where Region is NOT NULL
```

Results Messages

	CustomerID	CompanyName	Region
1	BOTTM	Bottom-Dollar Markets	BC
2	COMMI	Comércio Mineiro	SP
3	FAMIA	Familia Arquibaldo	SP
4	GOURL	Gourmet Lanchonetes	SP
5	GREAL	Great Lakes Food Market	OR
6	GROSR	GROSELLA-Restaurante	DF
7	HANAR	Hanari Cames	RJ
8	HILAA	HILARION-Abastos	Táchira
9	HUNGC	Hungry Coyote Import Store	OR
10	HUNGO	Hungry Owl All-Night Grocers	Co. Cork
11	ISLAT	Island Trading	Isle of Wight
12	LAUGB	Laughing Bacchus Wine Cellars	BC
13	LAZYK	Lazy K Kountry Store	WA



Ordenando los resultados (ascendente)

- Obtener el **nombre** del producto y su **precio** unitario **ordenando** del producto más barato al más caro
- El default es ordenar de menor a mayor

```
LORENAGOMEZ\...QLQuery2.sql* LORENAGOMEZ\...QLQuery1.sql*
select ProductName, UnitPrice
from products
order by UnitPrice
```

	ProductName	UnitPrice
1	Geitost	2.50
2	Guaraná Fantástica	4.50
3	Konbu	6.00
4	Filo Mix	7.00
5	Tourtière	7.45
6	Rhönbräu Klosterbier	7.75
7	Tunnbröd	9.00
8	Teatime Chocolate Biscuits	9.20
9	Zaanse koeken	9.50
10	Rogede sild	9.50
11	Jack's New England Clam Chowder	9.65
12	Sir Rodney's Scones	10.00
13	Aniseed Syrup	10.00
14	Longlife Tofu	10.00
15	Spegesild	12.00



Ordenando los resultados (descendente)

- Obtener el **nombre** del producto y su **precio** unitario **ordenando** del producto más caro al más barato
- Usar la clausula **DESC**

```
LORENAGOMEZ\...QLQuery2.sql* LORENAGOMEZ\...QLQuery1.sql*
select ProductName, UnitPrice
from products
order by UnitPrice DESC
```

	ProductName	UnitPrice
1	Côte de Blaye	263.50
2	Thüringer Rostbratwurst	123.79
3	Mishi Kobe Niku	97.00
4	Sir Rodney's Marmalade	81.00
5	Camavon Tigers	62.50
6	Raclette Courdavault	55.00
7	Manjimup Dried Apples	53.00
8	Tarte au sucre	49.30
9	Ipoh Coffee	46.00
10	Rössle Sauerkraut	45.60



Funciones en SQL

COUNT

- Regresa cuantos valores (diferentes de NULL) existen en la columna especificada.

COUNT(*)

- Regresa el número de registros en la tabla especificada

MIN

- Regresa el mínimo de los contenidos de la columna especificada. Primero elimina NULLs

MAX

- Regresa el máximo de los contenidos de la columna especificada. Primero elimina NULLs



Funciones en SQL

SUM

- Regresa la **suma** de los contenidos de la columna especificada. Primero elimina NULLs

AVG

- Regresa el **promedio** de los contenidos de la columna especificada. Primero elimina NULLs



Ejemplos del uso del count

- Los primeros 2 ejemplos cuentan el número de registros que tienen un valor diferente de NULL en el atributo region.
- El Ejemplo 3, cuenta los registros que tienen valores NULL en region
- `count(atributo)` elimina atributos NULL, `count(*)` no elimina region con NULL

```
select count(Region) from Customers;  
select count(Region) from Customers where region IS NOT NULL;  
select count(*) from Customers where region IS NULL;
```

Results		Messages
(No column name)		
1	31	
(No column name)		
1	31	
(No column name)		
1	60	



COUNT eliminando valores duplicados

- Usar DISTINCT para eliminar regiones que se repiten. Comparar los resultados obtenidos con cada query

```
select count( Region) from Customers;  
select count(distinct Region) from Customers
```

(No column name)	
1	31

(No column name)	
1	18



Ejemplo de MIN y MAX

- Obtener el costo del producto mas barato
- Obtener el costo del producto más caro

```
select MIN(UnitPrice) AS MasBarato  
from products;
```

```
select MAX(UnitPrice) AS MasCaro  
from products;
```

Results		Messages	
MasBarato			
1	2.50		
MasCaro			
1	263.50		

Ejemplo de AVG y SUM

- Obtener el precio promedio de todos los productos
- Obtener el número total de unidades en inventario de todos los productos

```
select AVG(UnitPrice) AS PrecioPromedio  
from products;
```

```
select SUM(UnitsInStock) AS TotalInventario  
from products;
```

Results		Messages
PrecioPromedio		
1	28.8663	
TotalInventario		
1	3119	



Misceláneo: MIN, MAX, AVG

- Varias funciones en la sentencia Select

```
select MIN(UnitPrice) AS precioMinimo,  
       MAX(UnitPrice) AS precioMaximo,  
       AVG(UnitPrice) AS PrecioPromedio  
from products;
```

Results

Messages

	precioMinimo	precioMaximo	PrecioPromedio
1	2.50	263.50	28.8663



GROUP BY

En el reporte puede verse que tengo varios productos de la misma categoría

Si quisiera saber cuantos productos tengo de cada

Podría **agrupar** y contar todos los registros que tie

Por ejemplo, tengo 12 productos de categoría 1

```
select *  
from products  
order by categoryID;
```

Results				
Messages				
	ProductID	ProductName	SupplierID	CategoryID
1	1	Chai	1	1
2	2	Chang	1	1
3	24	Guaraná Fantástica	10	1
4	34	Sasquatch Ale	16	1
5	35	Steeleye Stout	16	1
6	38	Côte de Blaye	18	1
7	39	Chartreuse verte	18	1
8	43	Ipoh Coffee	20	1
9	70	Outback Lager	7	1
10	67	Laughing Lumberjack Lager	16	1
11	75	Rhönbräu Klosterbier	12	1
12	76	Lakkalikööri	23	1
13	77	Original Frankfurter grüne Soße	12	2
14	61	Simn d'érahle	29	2

Group by

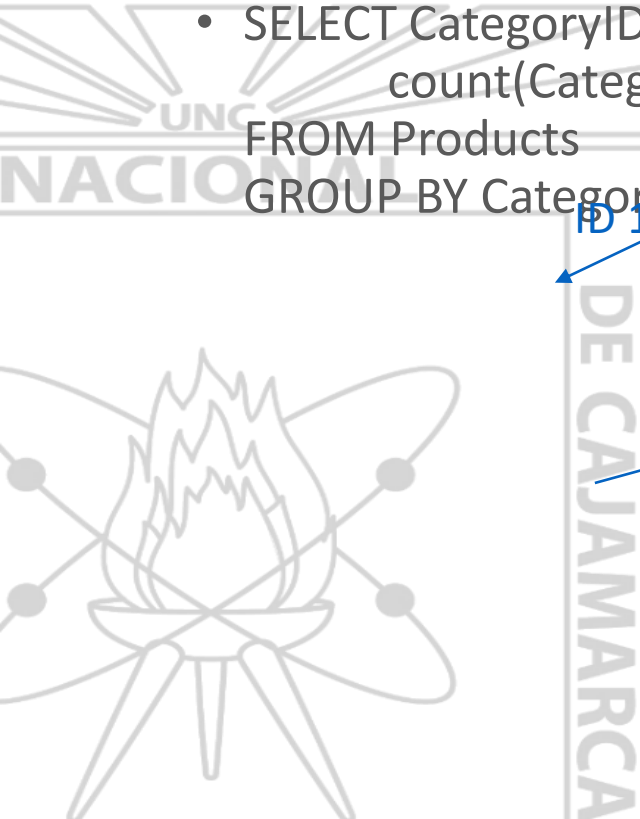
GROUP BY CategoryID

- Agrupa todos los registros que tienen el mismo CategoryID
- Se utiliza con otra función, por ejemplo COUNT

```
SELECT CategoryID,
       count(CategoryID)
FROM Products
GROUP BY CategoryID
```

ProductID	ProductName	SupplierID	CategoryID
1	Chai	1	1
2	Chang	1	1
24	Guaraná Fantástica	10	1
34	Sasquatch Ale	16	1
35	Steeleye Stout	16	1
38	Côte de Blaye	18	1
39	Chartreuse verte	18	1
43	Ipoh Coffee	20	1
70	Outback Lager	7	1
67	Laughing Lumberjack Lager	16	1
75	Rhönbräu Klosterbier	12	1
76	Lakkalikööri	23	1
77	Original Frankfurter grüne Soße	12	2
61	Sirop d'érable	29	2
63	Vegie-spread	7	2
65	Louisiana Fiery Hot Pepper Sauce	2	2
66	Louisiana Hot Spiced Okra	2	2
44	Gula Malacca	20	2
3	Aniseed Syrup	1	2
4	Chef Anton's Cajun Seasoning	2	2
5	Chef Anton's Gumbo Mix	2	2
6	Grandma's Boysenberry Spread	3	2

12 prod.





Count y Group by

- Se obtiene el número de productos que existen de cada categoría

```
SELECT CategoryID,  
count(CategoryID) as NumeroProductosPorCategoria  
FROM Products  
GROUP BY CategoryID
```

	CategoryID	NumeroProductosPorCategoria
1	1	12
2	2	12
3	3	13
4	4	10
5	5	7
6	6	6
7	7	5
8	8	12



ID o nombre?

- El tener un número de categoría no indica nada, es mejor poner el nombre de la categoría como en el resultado siguiente:

	CategoryID	CategoryName	NumeroProductosPorCategoria
1	1	Beverages	12
2	2	Condiments	12
3	3	Confections	13
4	4	Dairy Products	10
5	5	Grains/Cereals	7
6	6	Meat/Poultry	6
7	7	Produce	5
8	8	Seafood	12

- Pero, como se escribe el query? Con un JOIN entre la llave foránea FK IDCategory y la llave Primaria Categories(IDCategory)

Join

- El nombre de la categoría está en la tabla **Categories** y nuestra consulta utiliza la tabla de **Productos**
- Poner en el query las 2 tablas y especificar una condición donde la **FK=PK**

Table - dbo.Categories			Table - dbo.Products	Diagram - L...
CategoryID	CategoryName	Description		
1	Beverages	Soft drinks, coff...		
2	Condiments	Sweet and savo...		
3	Confections	Desserts, candie...		
4	Dairy Products	Cheeses		
5	Grains/Cereals	Breads, crackers...		
6	Meat/Poultry	Prepared meats		
7	Produce	Dried fruit and b...		
8	Seafood	Seaweed and fish		

Table - dbo.Products										Diagram - LOR...ind.Diagram_0	Summary
ProductID	ProductName	SupplierID	CategoryID	QuantityPerUnit	UnitPrice	UnitsInStock	UnitsOnOrder	ReorderLevel			
1	Chai	1	1	10 boxes x 20 b...	18.0000	39	0	10			
2	Chang	1	1	24 - 12 oz bottles	19.0000	17	40	25			
3	Aniseed Syrup	1	2	12 - 550 ml bottles	10.0000	13	70	25			
4	Chef Anton's Cajun Seasoning	2	2	48 - 6 oz jars	22.0000	53	0	0			
5	Chef Anton's Gumbo Mix	2	2	36 boxes	21.3500	0	0	0			
6	Grandma's Boysenberry Spread	3	2	12 - 8 oz jars	25.0000	120	0	25			
7	Uncle Bob's Organic Dried Pears	3	7	12 - 1 lb pkgs.	30.0000	15	0	10			
8	Northwoods Cranberry Sauce	3	2	12 - 12 oz jars	40.0000	6	0	0			
9	Mishi Kobe Niku	4	6	18 - 500 g pkgs.	97.0000	29	0	0			
10	Ikura	4	8	12 - 200 ml jars	31.0000	31	0	0			

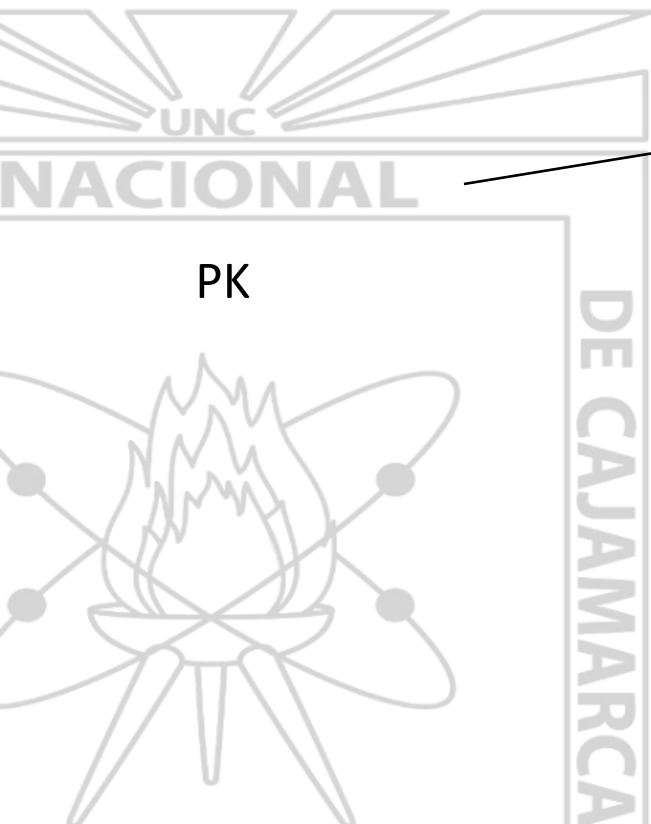


Join entre 2 tablas

Nombre del producto, Id de categoría y Nombre de la categoría

```
SELECT ProductName, Products.CategoryID, CategoryName  
FROM Products, Categories  
WHERE Products.CategoryID= Categories.CategoryID
```

FK



PK

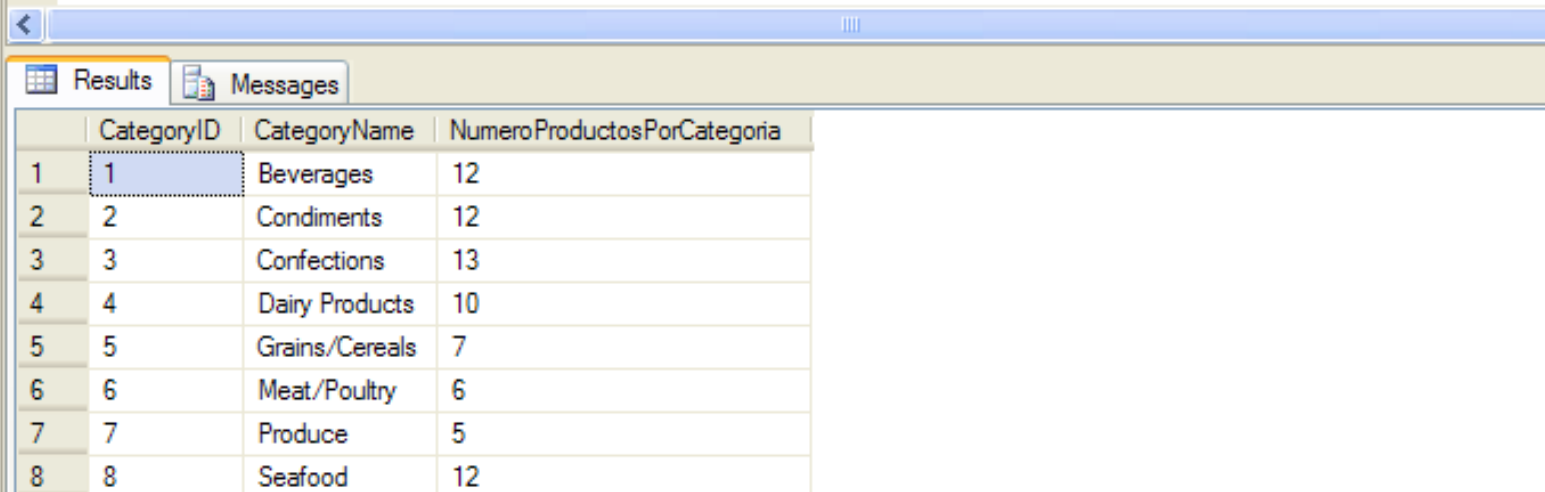
Results		Messages	
	ProductName	CategoryID	CategoryName
1	Chai	1	Beverages
2	Chang	1	Beverages
3	Aniseed Syrup	2	Condiments
4	Chef Anton's Cajun Seasoning	2	Condiments
5	Chef Anton's Gumbo Mix	2	Condiments
6	Grandma's Boysenberry Spread	2	Condiments
7	Uncle Bob's Organic Dried Pears	7	Produce
8	Northwoods Cranberry Sauce	2	Condiments
9	Mishi Kobe Niku	6	Meat/Poultry
10	Ikura	8	Seafood
11	Queso Cabrales	4	Dairy Products
12	Queso Manchego La Pastora	4	Dairy Products
13	Konbu	8	Seafood
14	Tofu	7	Produce
15	Genen Shouyu	2	Condiments
16	Pavlova	3	Confections
17	Alice Mutton	6	Meat/Poultry



Count y Group by

- número de productos existentes de cada categoría

```
SELECT Products.CategoryID, CategoryName,  
count(Products.CategoryID) as NumeroProductosPorCategoria  
FROM Products, Categories  
WHERE Products.CategoryID= Categories.CategoryID  
GROUP BY Products.CategoryID, CategoryName
```



	CategoryID	CategoryName	NumeroProductosPorCategoria
1	1	Beverages	12
2	2	Condiments	12
3	3	Confections	13
4	4	Dairy Products	10
5	5	Grains/Cereals	7
6	6	Meat/Poultry	6
7	7	Produce	5
8	8	Seafood	12



Count, Group by, Having

- número de productos existentes de cada categoría para los que el número de productos sea menor a 10

```
SELECT Products.CategoryID, CategoryName,  
count(Products.CategoryID) as NumeroProductosPorCategoria  
FROM Products, Categories  
WHERE Products.CategoryID= Categories.CategoryID  
GROUP BY Products.CategoryID, CategoryName  
having count(Products.CategoryID) <10
```

	CategoryID	CategoryName	NumeroProductosPorCategoria
1	5	Grains/Cereals	7
2	6	Meat/Poultry	6
3	7	Produce	5



Count, Group by, Having, alias

- número de productos existentes de cada categoría para los que el número de productos sea menor a 10
- Se utilizan **alias** para las tablas

```
SELECT P.CategoryID, CategoryName,  
count(P.CategoryID) as NumeroProductosPorCategoria  
FROM Products P, Categories C  
WHERE P.CategoryID= C.CategoryID  
GROUP BY P.CategoryID, CategoryName  
having count(P.CategoryID) <10;
```

	CategoryID	CategoryName	NumeroProductosPorCategoria
1	5	Grains/Cereals	7
2	6	Meat/Poultry	6
3	7	Produce	5



Union

- Obtener todos los países donde hay un cliente o un proveedor

```
select distinct country
from Suppliers
union
select distinct country
from Customers
```

Results Messages

	country
1	Argentina
2	Australia
3	Austria
4	Belgium
5	Brazil
6	Canada
7	Denmark
8	Finland
9	France
10	Germany



Interseccion

- Obtener todos los países donde hay proveedores y clientes

```
select distinct country  
from Suppliers  
intersect  
select distinct country  
from Customers
```

Results Messages

	country
1	Brazil
2	Canada
3	Denmark
4	Finland
5	France
6	Germany
7	Italy
8	Norway
9	Spain
10	Sweden
11	UK
12	USA

Diferencia (Except)

- Obtener los países donde hay proveedores, pero no hay clientes
- Obtener los países donde sólo hay clientes y no hay proveedores

```
select distinct country  
from Suppliers  
except  
select distinct country  
from Customers
```

	country
1	Australia
2	Japan
3	Netherlands
4	Singapore

```
select distinct country  
from Customers  
except  
select distinct country  
from Suppliers
```

	country
1	Argentina
2	Austria
3	Belgium
4	Ireland
5	Mexico
6	Poland
7	Portugal
8	Switzerland
9	Venezuela

Queries Anidados

- Encuentre los alumnos que tienen una especialidad (major) en computer science y en electrical engineering.

```
select id, name  
from cse_majors  
where id in
```

```
( select id  
  from eee_majors)
```

Evaluacion:

(1) query anidado

(2) query externo utilizando los resultados del query interno



Subqueries

- Encuentre los empleados que no han tomado algun curso

- SQL: `select name`

`from emp E`

`where not exists`

`(select *
from takes T
where T.id = E.id)`

`emp(id, name)
takes(id, crsid, date)
course(crsid, name)`

Para cada tupla de empleado, el query anidado selecciona todas las tuplas de **TAKES** cuyo **ID** es igual al **ID** del **empleado**; Si el resultado del subquery está vacío, entonces significa que el empleado no ha tomado algún curso y entonces **esa tupla** se incluye en el resultado.

EXISTS subquery: **VERDADERO** si el subquery SI regresa tuplas
FALSO si el subquery NO regresa tuplas

NOT EXISTS subquery = NOT(EXISTS subquery)



Herramientas de Trabajo

Online



SQL Server Management Studio

v. 19.1



Fin de la sesión

