



PROGRAMACIÓN DE BASE DE DATOS

Operaciones lógicas. Subconsultas.
Pivoteo de tablas. Group By. Having.

Introducción

- Quiz



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"



Poll Everywhere

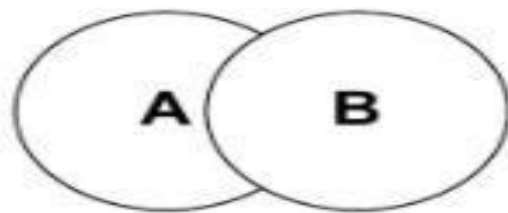


Agenda

Manejo Avanzado sobre Tablas

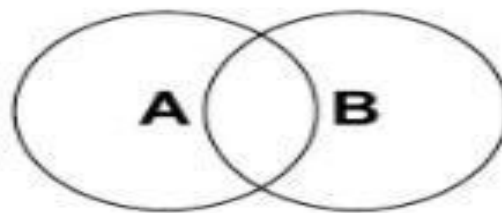
- Operadores Lógicos (unión, unión all, intersect, except)
- Manejo de subconsultas, tablas derivadas.
- Pivoteo de tablas (pivot, unpivot).
- Uso de GROUP BY, GROUPING, SET, HAVING.

Operadores Lógicos –Tipos de Operadores



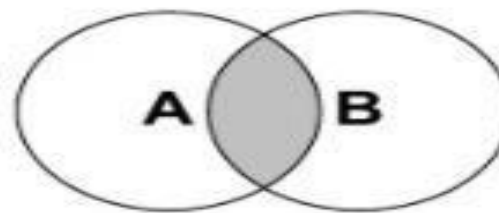
UNION

Une distintas consultas, no muestra los registros duplicados.



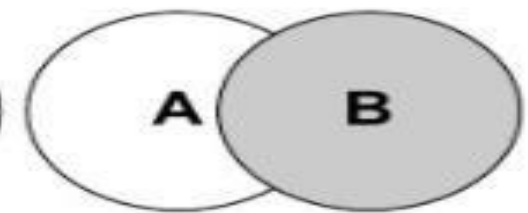
UNION ALL

Une distintas consultas, si muestra los registros duplicados.



INTERSECT

Une distintas consultas, retorna únicamente los registros que son duplicados.



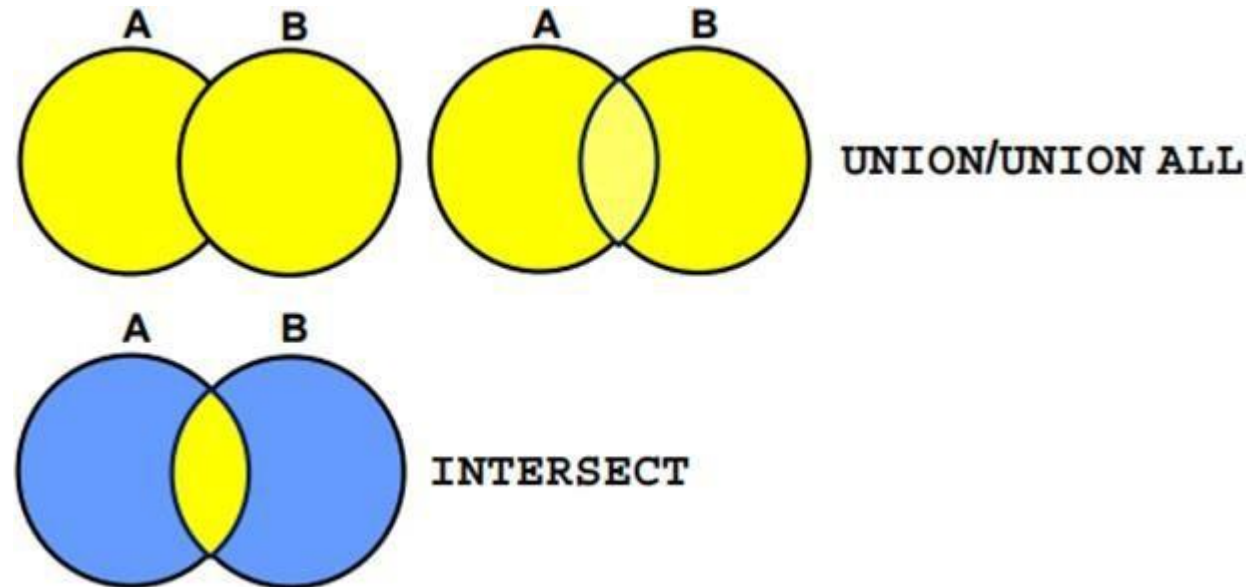
EXCEPT

Une distintas consultas, y solo muestra los registros de A y no los de B.

Manejo de operadores de conjunto

El operador UNION se utiliza para **combinar** el conjunto de resultados de dos o más instrucciones SELECT.

- Cada instrucción SELECT dentro de UNION debe tener el mismo número de columnas.
- Las columnas también deben tener tipos de datos similares.
- Las columnas en cada instrucción SELECT también deben estar en el mismo orden



Manejo de operadores de conjunto

Operador	Retorno
Union	Filas de ambas consultas después de eliminar duplicaciones
Union All	Filas de ambas consultas, incluidas todas las duplicaciones
Intersect	Filas comunes a ambas consultas.

```
select employee_id, job_id from [dbo].[employees]  
union  
select employee_id, job_id from [dbo].[job_history]
```

employee_id	job_id
100	AD_PRES
101	AC_ACCOUNT
101	AC_MGR

UNION

```
select employee_id, job_id from [dbo].[employees]  
union all  
select employee_id, job_id from [dbo].[job_history]
```

employee_id	job_id
100	AD_PRES
101	AD_VP
102	AD_VP
103	IT_PROG

UNION ALL



Manejo de operadores de conjunto

Operador	Retorno
Union	Filas de ambas consultas después de eliminar duplicaciones
Union All	Filas de ambas consultas, incluidas todas las duplicaciones
Intersect	Filas comunes a ambas consultas.

```
-select employee_id, job_id from [dbo].[employees]
intersect
select employee_id, job_id from [dbo].[job_history]
```

%

Results Messages

employee_id	job_id
176	SA_REP
200	AD_ASST

INTERSECT

Operadores Lógicos –Union

Lista los códigos de cada empleado, con su cargo actual y su historio.

```
SELECT employee_id, job_id
FROM   employees
UNION
SELECT employee_id, job_id
FROM   job_history;
```

employee_id	job_id
100	AD_PRES
101	AC_ACCOUNT
101	AC_MGR
101	AD_VP
102	AD_VP
...	

(105 rows)

Operadores Lógicos –Union



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

Lista los códigos de cada empleado, con su cargo actual y su historio.

```
SELECT employee_id, job_id
FROM   employees
UNION
SELECT employee_id, job_id
FROM   job_history;
```

employee_id	job_id
100	AD_PRES
101	AC_ACCOUNT
101	AC_MGR
101	AD_VP
102	AD_VP
...	...

(105 rows)

Operadores Lógicos –Union all

Lista los códigos de cada empleado, con su cargo actual y su historio, sin importar que se repitan.

```
SELECT employee_id, job_id, department_id
FROM employees
UNION ALL
SELECT employee_id, job_id, department_id
FROM job_history
ORDER BY employee_id;
```

employee_id	job_id	department_id
200	AC_ACCOUNT	90
200	AD_ASST	10
200	AD_ASST	90
201	MK_REP	20

(117 rows)

Operadores Lógicos –Intersect



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

Lista los códigos de cada empleado, con su cargo actual y su historio, siempre y cuando se repitan.

```
SELECT employee_id, job_id
FROM   employees
INTERSECT
SELECT employee_id, job_id
FROM   job_history
```

employee_id	job_id
176	SA_REP
200	AD_ASST

(2 rows)

Operadores Lógicos – Except



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

Lista los códigos de cada empleado, con su cargo actual y su historio, siempre y cuando no existan en la segunda consulta.

```
SELECT employee_id, job_id
FROM employees
EXCEPT
SELECT employee_id, job_id
FROM job_history;
```

employee_id	job_id
199	SH_CLERK
201	MK_MAN
202	MK_REP
203	HR_REP

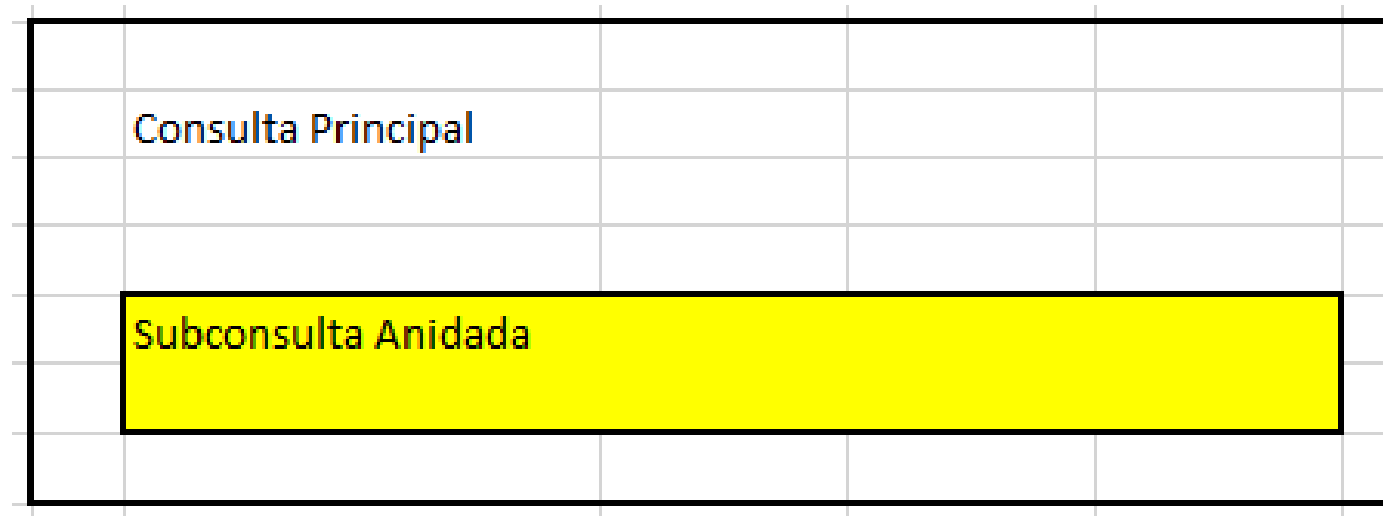
(105 rows)



¿Qué es una subconsulta?

Es una consulta se encuentra embebida en una consulta principal.

- Para extraer un grupo de datos específicos, los cuales dependen de una consulta principal.
- Se puede utilizar en cualquier parte de una consulta





Aplicando una subconsulta

```
SELECT  
E.APELLIDOS+' '+E.NOMBRES AS NOMBRES,  
E.SUELDO  
FROM dbo.CL_EMPLEADOS E  
WHERE E.SUELDO >=  
(SELECT CE.SUELDO  
FROM dbo.CL_EMPLEADOS CE WHERE CE.NOMBRES =  
'LEX')
```

→ Solo Devuelve
una fila el sueldo
de Lex

Results		Messages
NOMBRES	SUELDO	
KING STEVEN	24000.00	
KOCHHAR NEENA	17000.00	
DE HAAN LEX	17000.00	

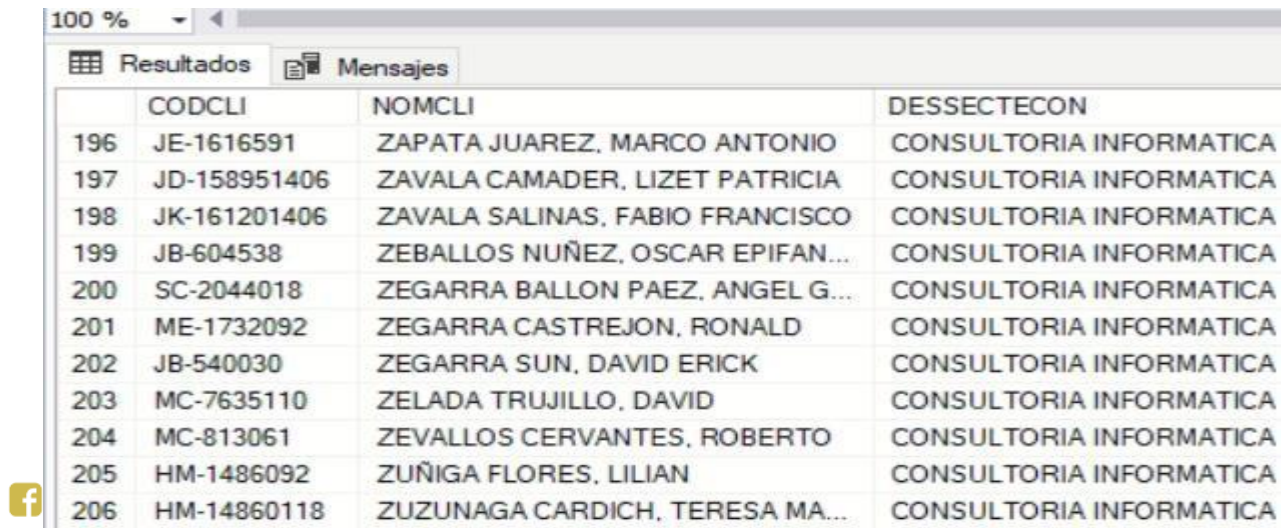
Aplicando una subconsulta

```
SELECT
CP.CODCLI,
CP.NOMCLI,
CP.DESSECTECON
FROM MM_CLIENTE CP
WHERE CP.DESSECTECON =
    (SELECT
        C.DESSECTECON
    FROM MM_CLIENTE C
        WHERE C.CODCLI = 'SK-1999059'); -- RESULTADO DE
LA SUBCONSULTA 'CONSULTORIA INFORMATICA'
```



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

→ Devuelve filas
de acuerdo a el
dato de la
subconsulta



	CODCLI	NOMCLI	DESSECTECON
196	JE-1616591	ZAPATA JUAREZ, MARCO ANTONIO	CONSULTORIA INFORMATICA
197	JD-158951406	ZAVALA CAMADER, LIZET PATRICIA	CONSULTORIA INFORMATICA
198	JK-161201406	ZAVALA SALINAS, FABIO FRANCISCO	CONSULTORIA INFORMATICA
199	JB-604538	ZEBALLOS NUÑEZ, OSCAR EPIFAN...	CONSULTORIA INFORMATICA
200	SC-2044018	ZEGARRA BALLON PAEZ, ANGEL G...	CONSULTORIA INFORMATICA
201	ME-1732092	ZEGARRA CASTREJON, RONALD	CONSULTORIA INFORMATICA
202	JB-540030	ZEGARRA SUN, DAVID ERICK	CONSULTORIA INFORMATICA
203	MC-7635110	ZELADA TRUJILLO, DAVID	CONSULTORIA INFORMATICA
204	MC-813061	ZEVALLOS CERVANTES, ROBERTO	CONSULTORIA INFORMATICA
205	HM-1486092	ZUÑIGA FLORES, LILIAN	CONSULTORIA INFORMATICA
206	HM-14860118	ZUZUNAGA CARDICH, TERESA MA...	CONSULTORIA INFORMATICA



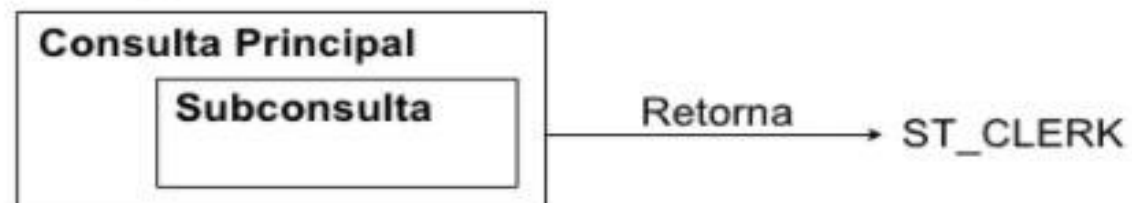
Consideraciones de una subconsulta

Se deben de considerar las siguientes situaciones:

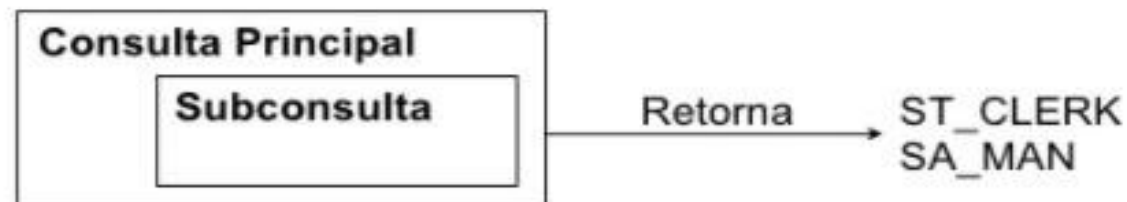
- Encerrar las subconsultas dentro de los paréntesis.
- Escriba la su consulta del lado derecho de la condición.
- No usar las clausulas order by dentro de la su consulta, ya que no solo no es necesario sino que compromete el rendimiento de la misma.
- Utilice los operadores adecuados para las subconsultas que arrojan uno o múltiples registros.

Consideraciones de una subconsulta

- Por un solo registro:



- Por múltiples registros:





Por un solo registro

- Retorna un solo valor.
- Se utilizan los siguientes operadores de comparación:

Operador	Significado
=	Igual
>	Mayor que
<	Menor que
>=	Mayor e igual que
<=	Menor e igual que
<>	Diferente

Errores Comunes ejemplo 1



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

```
SELECT employee_id, last_name
FROM   employees
WHERE  salary = (SELECT MIN(salary)
                 FROM   employees
                 GROUP BY department_id);
```

ERROR: more than one row returned by a
subquery used as an expression

min

7000
2500
8300
4200
17000
4400
6500
10000
.
.
.
(12 rows)

Errores Comunes ejemplo 2

```
SELECT last_name, job_id
FROM employees
WHERE job_id = (SELECT job_id
                 FROM employees
                 WHERE last_name = 'Haas');
```

last_name	job_id
(0 rows)	

job_id
(0 rows)



Valores Nulos en las subconsultas

```
SELECT emp.last_name  
FROM   employees emp  
WHERE  emp.employee_id NOT IN (SELECT mgr.manager_id  
                                FROM   employees mgr);
```

last_name

(0 rows)

manager_id

100
100
100
100
100
145
145
145
145
...
(1 row)



¿Cómo retornar un solo valor?

```
SELECT  
E.APELLIDOS+ ' ' +E.NOMBRES AS NOMBRES,  
E.ID_CARGOS, E.SUELDO  
FROM dbo.CL_EMPLEADOS E  
WHERE E.ID_CARGOS =  
(SELECT CE.ID_CARGOS  
FROM dbo.CL_EMPLEADOS CE WHERE CE.ID_EMPLEADO =  
141)  
AND E.SUELDO >  
(SELECT CE1.SUELDO  
FROM dbo.CL_EMPLEADOS CE1 WHERE CE1.ID_EMPLEADO =  
143);
```

	NOMBRES	ID_CARGOS	SUELDO
1	NAYER JULIA	ST_CLERK	3200.00
2	MIKKILINENI IRENE	ST_CLERK	2700.00
3	BISSOT LAURA	ST_CLERK	3300.00
4	ATKINSON MOZHE	ST_CLERK	2800.00
5	MALLIN JASON	ST_CLERK	3300.00
6	ROGERS MICHAEL	ST_CLERK	2900.00
7	LADWIG RENSKE	ST_CLERK	3600.00
8	STILES STEPHEN	ST_CLERK	3200.00
9	SEO JOHN	ST_CLERK	2700.00
10	RAJS TRENNNA	ST_CLERK	3500.00
11	DAVIES CURTIS	ST_CLERK	3100.00



Usando la clausula Having

```
SELECT
E.ID_DPTO,
MIN(E.SUELDO)
FROM DBO.CL_EMPLEADOS E
GROUP BY E.ID_DPTO
HAVING MIN( E.SUELDO) >
        (SELECT MIN(S.SUELDO) AS MIN_SUELDO
        FROM DBO.CL_EMPLEADOS S WHERE S.ID_DPTO = 50)
ORDER BY 1,2;
```

100 %		
Results Messages		
	ID_DPTO	(No column name)
1	NULL	7000.00
2	10	4400.00
3	20	6000.00
4	30	2500.00
5	40	6500.00
6	60	4200.00
7	70	10000.00
8	80	6100.00
9	90	17000.00
10	100	6900.00
11	110	8300.00



Manejo de subconsultas, tablas derivadas

Sub Consultas: no usa valores de la consulta externa

Tablas Derivadas: usa valores de la consulta externa.

Como parte
de un where

```
select *  
from Employees e  
left join [dbo].[EmployeeTerritories] et on et.[EmployeeID] = e.[EmployeeID]  
where et.TerritoryID in (select TerritoryID  
from [dbo].[Territories] where RegionID = 3)
```

Como una
columna

```
select f.*,  
(select sum(d.UnitPrice*Quantity)  
from [dbo].[Order Details] as d  
where f.[OrderID]=d.[OrderID]) as total  
from [dbo].[Orders] f
```



Manejo de subconsultas, tablas derivadas

Sub Consultas: no usa valores de la consulta externa

Tablas Derivadas: usa valores de la consulta externa.

Obtener los clientes que han
hecho por lo menos un pedido

```
select c.CustomerID, c.CompanyName, c.ContactName
from Customers c
where c.CustomerID in (select CustomerID from Orders )
```

Otra forma de escribirse es
utilizando la clausula **EXISTS**

```
select c.CustomerID, c.CompanyName, c.ContactName
from Customers c
where exists (select CustomerID from Orders o
              where o.CustomerID = c.CustomerID)
```

Manejo de subconsultas, tablas derivadas



Universidad
Nacional de
Cajamarca
"Núcleo de la Universidad Peruana"

Usando el operador Any

/ Quiero saber los nombres de los empleados que ganan menos en sus respectivos cargos */*

```
SELECT
E.ID_EMPLEADO,
E.APELLIDOS+' '+E.NOMBRES AS NOMBRES, E.ID_CARGOS,
E.SUELDO,
E.ID_DPTO
FROM DBO.CL_EMPLEADOS E
WHERE E.SUELDO = ANY
(
    SELECT
    DISTINCT A.MIN_SUELDO
    FROM
    (SELECT S.ID_CARGOS,
min(S.SUELDO) AS MIN_SUELDO
    FROM DBO.CL_EMPLEADOS S
    --WHERE S.ID_CARGOS = 'IT_PROG'
    GROUP BY S.ID_CARGOS) A
);
```

Results		Message
MIN_SUELDO		
1	2100.00	
2	2500.00	
3	4200.00	
4	4400.00	
5	5800.00	
6	6000.00	
7	6100.00	
8	6500.00	
9	6900.00	
10	8300.00	
11	10000.00	
12	10500.00	

	ID_EMPLEADO	NOMBRES	ID_CARGOS	SUELDO	ID_DPTO
1	100	KING STEVEN	AD_PRES	24000.00	90
2	101	KOCHHAR NEENA	AD_VP	17000.00	90
3	102	DE HAAN LEX	AD_VP	17000.00	90
4	104	ERNST BRUCE	IT_PROG	6000.00	60
5	107	LORENTZ DIANA	IT_PROG	4200.00	60
6	108	GREENBERG NANCY	FI_MGR	12000.00	100
7	113	POPP LUIS	FI_ACCOUNT	6900.00	100
8	114	RAPHAELY DEN	PU_MAN	11000.00	30
9	119	COLMENARES KAREN	PU_CLERK	2500.00	30
10	123	VOLLMAN SHANTA	ST_MAN	6500.00	50
11	124	MOURGOS KEVIN	ST_MAN	5800.00	50
12	131	MARLOW JAMES	ST_CLERK	2500.00	50

Query executed successfully.

Manejo de subconsultas, tablas derivadas

Usando el operador All



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

```
SELECT
E.ID_EMPLEADO,
E.APELLIDOS+ ' ' +E.NOMBRES AS NOMBRES,
E.ID_CARGOS,
E.SUELDO,
E.ID_DPTO
FROM DBO.CL_EMPLEADOS E
WHERE E.SUELDO > ALL
(
SELECT
DISTINCT A.MIN_SUELDO
FROM
(SELECT S.ID_CARGOS, min(S.SUELDO) AS
MIN_SUELDO
FROM DBO.CL_EMPLEADOS S
WHERE S.SUELDO < 17000
GROUP BY S.ID_CARGOS) A
);
```

	MIN_SUELDO
1	2100.00
2	2500.00
3	4200.00
4	4400.00
5	5800.00
6	6000.00
7	6100.00
8	6500.00
9	6900.00
10	8300.00
11	10000.00
12	10500.00

	ID_EMPLEADO	NOMBRES	ID_CARGOS	SUELDO	ID_DPTO
1	100	KING STEVEN	AD_PRES	24000.00	90
2	101	KOCHHAR NEENA	AD_VP	17000.00	90
3	102	DE HAAN LEX	AD_VP	17000.00	90
4	145	RUSSELL JOHN	SA_MAN	14000.00	80
5	146	PARTNERS KAREN	SA_MAN	13500.00	80

Mas formas de filtrado

```
select *  
from [dbo].[Employees]  
where EmployeeID = any (select EmployeeID  
                        from Orders  
                        group by EmployeeID  
                        having count(1) > 100)
```

Results				
Messages				
EmployeeID	LastName	FirstName	Title	TitleOfCourtesy
1	Davolio	Nancy	Sales Representative	Ms.
3	Leverling	Janet	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
8	Callahan	Laura	Inside Sales Coordinator	Ms.

ANY

```
select EmployeeID, COUNT(1)  
from Orders  
group by EmployeeID  
order by 2 desc
```

Results		
Messages		
	EmployeeID	(No column name)
1	4	156
2	3	127
3	1	123
4	8	104
5	2	96

Mas formas de filtrado

```
--select *
from [dbo].[Employees]
where EmployeeID = all (select EmployeeID
                        from Orders
                        group by EmployeeID
                        having count(1) > 100)
```

%

Results Messages

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate
------------	----------	-----------	-------	-----------------	-----------	----------

ALL

```
--select EmployeeID, COUNT(1)
from Orders
group by EmployeeID
order by 2 desc
```

121 %

Results Messages

	EmployeeID	(No column name)
1	4	156
2	3	127
3	1	123
4	8	104
5	2	96

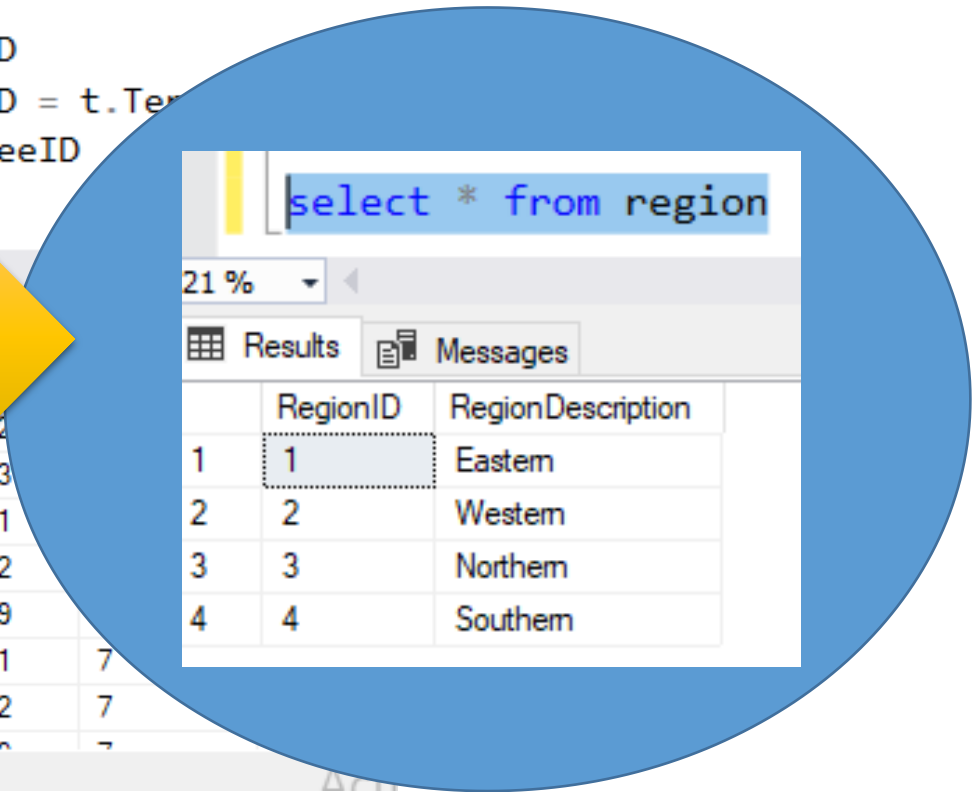
Mas formas de filtrado

```
select *  
from [dbo].[Region] r  
left join [dbo].[Territories] t on t.RegionID = r.RegionID  
left join [dbo].[EmployeeTerritories] et on et.TerritoryID = t.TerritoryID  
left join [dbo].[Employees] e on e.EmployeeID = et.EmployeeID  
where r.RegionID in (1,2)
```

21 %

Results Messages

	RegionID	RegionDescription	TerritoryID	TerritoryDescription	RegionID	EmployeeID	TerritoryID
16	1	Eastern	20852	Rockville	1	4	20852
17	1	Eastern	27403	Greensboro	1	4	27403
18	1	Eastern	27511	Cary	1	4	27511
19	1	Eastern	40222	Louisville	1	2	40222
20	2	Western	60179	Hoffman Estates	2	7	60179
21	2	Western	60601	Chicago	2	7	60601
22	2	Western	80202	Denver	2	7	80202



Mas formas de filtrado

```
select *
from [dbo].[Customers]
where exists
(
    select CustomerID, count(1) NroPedidos
    from [Orders]
    where [Customers].CustomerID = [Orders].CustomerID
    group by CustomerID
    having count(1) > 20
)
```

CustomerID	CompanyName	ContactName	ContactTitle	Address
ERNSH	Ernst Handel	Roland Mendel	Sales Manager	Kirchgasse 6
QUICK	QUICK-Stop	Horst Kloss	Accounting Manager	Taucherstraße 10
SAVEA	Save-a-lot Markets	Jose Pavarotti	Sales Representative	187 Suffolk Ln.

EXISTS

```
select CustomerID, count(1) NroPedidos
from [Orders]
group by CustomerID
```

CustomerID	NroPedidos
ALFKI	6
ANATR	4
ANTON	7
AROUT	13

Operador EXISTS:

- Devuelve true o false.
- **True:** si las subconsultas retornan registros.
- **False:** si las subconsultas no retornan registros.

Manejo de subconsultas, tablas derivadas



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

Usando el operador Exists

/ Mostrar las ventas del 13 de enero de 2019 si existieron perdidas mayores a 100 */*

```
SELECT
PE.FECPEDID,
PE.NOMPROD,
PE.NOMVEND,
PE.MTOVALVTA,
PE.MTOBENEF
FROM DBO.TAB_PEDIDO PE
WHERE PE.FECPEDID = '20190113' -- Mostrar las ventas del 13 de
    enero de 2019 si existieron perdidas mayores a 100
AND
    EXISTS (SELECT
        P.MTOBENEF
        FROM DBO.TAB_PEDIDO P
        WHERE P.MTOBENEF < -100
        AND P.FECPEDID = '20190113'
    )

ORDER BY 5;
```

	FECPEDID	NOMPROD	NOMVEND	MTOVALVTA	MTOBENEF
1	2019-01-13 00:00:00.000	GBC Ibimaster 500 Manual ProClick Binding System	Autoventa	-380.490	-1141.470
2	2019-01-13 00:00:00.000	Dania Floating Shelf Set, Traditional	Christian Peña	585.924	-64.380
3	2019-01-13 00:00:00.000	Sanford Canvas, Water Color	Autoventa	-23.628	-32.220
4	2019-01-13 00:00:00.000	Stanley Canvas, Easy-Erase	Victor Medina	-9.975	-17.934
5	2019-01-13 00:00:00.000	Boston Markers, Blue	Francisco Gutierrez	-0.802	-5.022
6	2019-01-13 00:00:00.000	Advantus Staples, 12 Pack	Autoventa	8.289	-4.161
7	2019-01-13 00:00:00.000	Avery Hole Reinforcements, Economy	Francisco Gutierrez	864.360	0.000
8	2019-01-13 00:00:00.000	Sanford Markers, Easy-Erase	Victor Medina	23.970	0.450
9	2019-01-13 00:00:00.000	Cardinal Binder Covers, Economy	Francisco Gutierrez	48.960	0.960
10	2019-01-13 00:00:00.000	Cardinal Index Tab, Clear	Manuel Gonzalez	9.780	2.520
11	2019-01-13 00:00:00.000	Tenex Folders, Single Width	Christian Peña	391.300	3.800



Manejo de subconsultas, tablas derivadas



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

Usando el operador Exists

/* Mostrar fecha de pedido, cliente, producto
cuando los un clientes haya realizado mas de un
pedido*/

```
SELECT  
P.FECPEDID,  
C.NOMCLI,  
P.CODPROD,  
PR.NOMPROD,  
P.CTDPEDID  
FROM HD_PEDIDO P  
LEFT JOIN MM_CLIENTE C ON P.CODCLI = C.CODCLI  
LEFT JOIN MM_PRODUCTO PR ON P.CODPROD = PR.CODPROD  
WHERE P.FECPEDID = '20190113' AND  
EXISTS
```

```
(SELECT  
PE.CODCLI,  
COUNT(1) AS NROPEDIDOS  
FROM DBO.HD_PEDIDO PE  
WHERE PE.FECPEDID = '20190113'  
AND P.CODCLI = PE.CODCLI  
GROUP BY PE.CODCLI  
HAVING COUNT(1) > 1)
```

ORDER BY 2;

	FECPEDID	NOMCLI	CODPROD	NOMPROD	CTDPEDID
1	2019-01-13 00:00:00.000	CALDERON VALENCIA, ANGEL	OFF-BI-3718	Cardinal Binder Covers, Economy	2.000
2	2019-01-13 00:00:00.000	CALDERON VALENCIA, ANGEL	OFF-BI-6376	Wilson Jones Binder Covers, Economy	4.000
3	2019-01-13 00:00:00.000	CAMPDONICO PEREYRA, PAOLA	OFF-PA-6505	Xerox 195	6.000
4	2019-01-13 00:00:00.000	CAMPDONICO PEREYRA, PAOLA	FUR-FU-4329	Flat Face Poster Frame	2.000

Manejo de subconsultas, tablas derivadas



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

Usando el operador not Exists

/ Mostrar las ventas del 13 de enero de 2019 si no existieron perdidas mayores a 100 */*

```
SELECT
PE.FECPEDID,
PE.NOMPROD,
PE.NOMVEND,
PE.MTOVALVTA,
PE.MTOBENEF
FROM DBO.TAB_PEDIDO PE
WHERE PE.FECPEDID = '20190113' -- Mostrar las ventas del 13 de
    enero de 2019 si existieron perdidas mayores a 100
AND
NOT EXISTS (SELECT
                P.MTOBENEF
            FROM DBO.TAB_PEDIDO P
            WHERE P.MTOBENEF < -100
            AND P.FECPEDID = '20190113'
        )

ORDER BY 5;
```

FECPEDID	NOMPROD	NOMVEND	MTOVALVTA	MTOBENEF
----------	---------	---------	-----------	----------

Pivoteo de tablas (píivot, unpivot).

Convertir filas en
columnas **PIVOT** y
columnas en filas
UNPIVOT

anio	Name	SubTotal
2011	Southeast	20565,6206
2011	Southeast	1294,2529
2011	Southwest	419,4589
2011	Northwest	24432,6088
2011	Northwest	14352,7713
2011	Southwest	5056,4896
2011	Central	6107,082
2011	Northwest	714,7043

PIVOT

Name	2011	2012	2013	2014
Northwest	1248473,5306	2937972,7031	3584751,866	1596395,5334
Northeast	875823,8318	3375456,8947	3985374,8995	1057247,3786
Central	1311627,2918	4317306,5741	3396776,2674	1040093,4071
Southwest	2117312,6152	6129119,2246	6498550,9672	2049030,1735
Southeast	1521289,1881	2674436,3518	2188082,7813	787204,4289

Name	2011	2012	2013	2014
Northwest	1248473,5306	2937972,7031	3584751,866	1596395,5334
Northeast	875823,8318	3375456,8947	3985374,8995	1057247,3786
Central	1311627,2918	4317306,5741	3396776,2674	1040093,4071
Southwest	2117312,6152	6129119,2246	6498550,9672	2049030,1735
Southeast	1521289,1881	2674436,3518	2188082,7813	787204,4289

UNPIVOT

anio	Name	SubTotal
2011	Southeast	20565,6206
2011	Southeast	1294,2529
2011	Southwest	419,4589
2011	Northwest	24432,6088
2011	Northwest	14352,7713
2011	Southwest	5056,4896
2011	Central	6107,082
2011	Northwest	714,7043



Pivoteo de tablas (pívo, unpivot).

Convertir filas en
columnas **PIVOT** y
columnas en filas
UNPIVOT

```
SELECT <non-pivoted column>,  
      [first pivoted column] AS <column name>,  
      [second pivoted column] AS <column name>,  
      ...  
      [last pivoted column] AS <column name>  
FROM  
      (<SELECT query that produces the data>  
      AS <alias for the source query>  
PIVOT  
(  
      <aggregation function>(<column being aggregated>  
FOR  
[<column that contains the values  
      that will become column headers>]  
      IN ( [first pivoted column], [second pivoted column],  
      ... [last pivoted column])  
) AS <alias for the pivot table>  
<optional ORDER BY clause>;
```



Uso de ROLLUP, CUBE, GROUPING SET

Si deseamos generar resúmenes de los datos de tipo OLAP con subtotales y totales, tenemos funciones que facilitan estas operaciones

GROUP BY crea agrupaciones que están definidas por un conjunto de expresiones. Cada fila tiene una combinación única, para usar funciones agregadas como **COUNT** o **SUM**

Sin embargo, para agrupar datos por múltiples combinaciones para ello utilice GROUPING SETS junto con la cláusula GROUP BY y definir cada grupo de agrupación dentro de una sola consulta.

```
GROUP BY {  
    column-expression  
    | ROLLUP ( <group_by_expression> [ ,...n ] )  
    | CUBE ( <group_by_expression> [ ,...n ] )  
    | GROUPING SETS ( <grouping_set> [ ,...n ] )  
    | () --calculates the grand total  
} [ ,...n ]
```



Clausulas que permiten realizar múltiples agregaciones en una única query; con las columnas especificadas en la clausula **Group BY**

Uso de ROLLUP, CUBE, GROUPING SET

ROLLUP, creara una nueva fila con la suma de cada grupo, además de una fila del gran total

CUBE, creara una nueva fila con el nivel de agregación interna. Adiciona un gran total. Y adiciona un total por el nivel de agregación por cada grupo

Seafood	Inlagd Sill	14542.60	0
Seafood	Jack's New England Clam...	9098.10	0
Seafood	Konbu	5234.40	0
Seafood	Nord-Ost Matjeshering	14775.54	0
Seafood	Röd Kaviar	4200.00	0
Seafood	Rogede sild	4740.50	0
Seafood	Spegesild	6144.00	0
Seafood	NULL	141623.09	1
NULL	NULL	1354458...	3

149	Condiments	Vegie-spread	17696.30	0
150	NULL	Vegie-spread	17696.30	2
151	Grains/Cereals	Wimmers gute S...	23009.00	0
152	NULL	Wimmers gute S...	23009.00	2
153	Confections	Zaanse koeken	4358.60	0
154	NULL	Zaanse koeken	4358.60	2
155	NULL	NULL	135445...	3
156	Beverages	NULL	286526...	1
157	Condiments	NULL	113694...	1
158	Confections	NULL	177099...	1
159	Dairy Products	NULL	251330...	1
160	Grains/Cereals	NULL	100726...	1
161	Meat/Poultry	NULL	178188...	1
162	Produce	NULL	105268...	1
163	Seafood	NULL	141623...	1

Uso de ROLLUP, CUBE, GROUPING SET

GROUPING SETS, creara totales según la agrupación que desee aplicar.

GROUPING_ID, crea una columna lógica que identifica el grupo generado por la cláusula *ROLLUP*, *CUBE* o *GROUPING SETS*

```
select grouping_id ( categoria, subcategoria,  
                    producto) as GrpId,  
       categoria, subcategoria, producto, sum(qty) qty  
from ordenes  
group by grouping sets (  
    (categoria, subcategoria, producto),  
    (categoria, subcategoria),  
    (categoria),  
    ()  
)  
order by GrpId, categoria, subcategoria, producto
```

GrpId	categoria	subcategoria	producto	qty
0	Cate1	SubCat1	Producto1	100
0	Cate2	SubCat2	Producto2	200
0	Cate3	SubCat3	Producto3	300
0	Cate4	SubCat4	Producto4	400
1	Cate1	SubCat1	NULL	100
1	Cate2	SubCat2	NULL	200
1	Cate3	SubCat3	NULL	300
1	Cate4	SubCat4	NULL	400
3	Cate1	NULL	NULL	100
3	Cate2	NULL	NULL	200
3	Cate3	NULL	NULL	300
3	Cate4	NULL	NULL	400
7	NULL	NULL	NULL	1000

Pivot Examples

```

SELECT * FROM
(
  SELECT
    FORMAT(P.FECPEDID, 'yyyyMM') AS PERIODO,
    PR.NOMSUBCAT AS SUBCATEGORIA,
    SUM(P.MTOVALVTA) AS VALVENTA
  FROM HD_PEDIDO P
  — LEFT JOIN MM_PRODUCTO PR ON P.CODPROD =
  — PR.CODPROD
  — WHERE FORMAT(P.FECPEDID, 'yyyy') = 2022
  — GROUP BY FORMAT(P.FECPEDID, 'yyyyMM'),
  — PR.NOMSUBCAT
  — ) T
PIVOT
(
  SUM(VALVENTA)
  FOR PERIODO IN
    ([202201], [202202], [202203], [202204], [202205],
    [202206], [202207])
) AS TPIVOT
  
```



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

	SUBCATEGORIA	202201	202202	202203	202204	202205	202206	202207
1	Accesorios	163048.254	220518.466	203520.084	170926.490	194759.270	279514.966	329991.189
2	Almacenamiento	244001.299	185737.231	263000.077	239153.415	212850.919	256670.419	319555.245
3	Archivadores	84633.150	85860.818	174629.654	105814.098	93446.726	138213.982	127021.099
4	Arte	78555.059	84604.055	90367.311	89031.573	118200.595	119860.743	123605.257
5	Artefactos	358529.891	228574.973	273634.491	274096.523	399777.615	341710.323	341567.234
6	Copiadoras	297566.760	240485.108	398522.882	505165.988	519560.694	459887.996	346598.350
7	Etiquetas	15578.326	16263.774	23128.056	17529.074	28749.800	18629.494	18887.949
8	Grapas	19008.393	21949.173	24821.503	21160.951	30573.693	38816.507	23813.342
9	Librerías	434358.008	341281.860	419175.604	360843.336	466444.642	451086.318	370869.695
10	Máquinas	173600.921	115884.979	148806.857	133243.421	198113.775	195040.441	241064.478
11	Material	77488.915	70466.665	70911.441	78409.451	86195.169	111426.161	76296.598
12	Mesas	155289.092	150541.960	299041.542	219664.960	101155.888	201753.654	97016.302
13	Mobiliario	114667.397	101765.025	99653.959	111342.087	107062.431	158514.867	145005.986
14	Papel	74027.616	57474.846	86338.300	58548.696	107930.990	79538.184	96387.063
15	Sillas	307269.856	336573.102	290111.076	314584.524	389817.734	400188.540	432395.229
16	Sobres	38799.279	34857.939	31840.057	53957.573	50681.085	66211.145	66629.859
17	Teléfonos	426135.411	482318.415	459291.045	529101.877	620831.243	562984.357	652478.505

Pivot Examples

```

SELECT * FROM
(
  SELECT
    FORMAT(P.FECPEDID, 'yyyyMM') AS PERIODO,
    PR.NOMSUBCAT AS SUBCATEGORIA,
    SUM(P.MTOVALVTA) AS VALVENTA
  FROM HD_PEDIDO P
  — LEFT JOIN MM_PRODUCTO PR ON P.CODPROD =
  — PR.CODPROD
  — WHERE FORMAT(P.FECPEDID, 'yyyy') = 2022
  — GROUP BY FORMAT(P.FECPEDID, 'yyyyMM'),
  — PR.NOMSUBCAT
  — ) T
PIVOT
(
  SUM(VALVENTA)
  FOR PERIODO IN
    ([202201],[202202],[202203],[202204],[202205],
    [202206],[202207])
) AS TPIVOT
  
```



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

	SUBCATEGORIA	202201	202202	202203	202204	202205	202206	202207
1	Accesorios	163048.254	220518.466	203520.084	170926.490	194759.270	279514.966	329991.189
2	Almacenamiento	244001.299	185737.231	263000.077	239153.415	212850.919	256670.419	319555.245
3	Archivadores	84633.150	85860.818	174629.654	105814.098	93446.726	138213.982	127021.099
4	Arte	78555.059	84604.055	90367.311	89031.573	118200.595	119860.743	123605.257
5	Artefactos	358529.891	228574.973	273634.491	274096.523	399777.615	341710.323	341567.234
6	Copiadoras	297566.760	240485.108	398522.882	505165.988	519560.694	459887.996	346598.350
7	Etiquetas	15578.326	16263.774	23128.056	17529.074	28749.800	18629.494	18887.949
8	Grapas	19008.393	21949.173	24821.503	21160.951	30573.693	38816.507	23813.342
9	Librerías	434358.008	341281.860	419175.604	360843.336	466444.642	451086.318	370869.695
10	Máquinas	173600.921	115884.979	148806.857	133243.421	198113.775	195040.441	241064.478
11	Material	77488.915	70466.665	70911.441	78409.451	86195.169	111426.161	76296.598
12	Mesas	155289.092	150541.960	299041.542	219664.960	101155.888	201753.654	97016.302
13	Mobiliario	114667.397	101765.025	99653.959	111342.087	107062.431	158514.867	145005.986
14	Papel	74027.616	57474.846	86338.300	58548.696	107930.990	79538.184	96387.063
15	Sillas	307269.856	336573.102	290111.076	314584.524	389817.734	400188.540	432395.229
16	Sobres	38799.279	34857.939	31840.057	53957.573	50681.085	66211.145	66629.859
17	Teléfonos	426135.411	482318.415	459291.045	529101.877	620831.243	562984.357	652478.505

Pivot Examples GROUPING Y GROUP BY CUBE

```
SELECT
FORMAT(P.FECPEDID, 'yyyyMM') AS PERIODO,
PR.NOMSUBCAT AS SUBCATEGORIA,
SUM(P.MTOVALVTA) AS VALVENTA,
GROUPING(PR.NOMSUBCAT) AS SUBCAT_GROUP,
GROUPING(FORMAT(P.FECPEDID, 'yyyyMM')) AS
PER_GROUP
FROM HD_PEDIDO P
LEFT JOIN MM_PRODUCTO PR ON P.CODPROD =
PR.CODPROD
WHERE FORMAT(P.FECPEDID, 'yyyy') = 2022
GROUP BY CUBE (FORMAT(P.FECPEDID, 'yyyyMM'),
PR.NOMSUBCAT)
```

Grouping

	PERIODO	SUBCATEGORIA	VALVENTA	PER_GROUP	SUBCAT_GROUP
126	202206	Sobres	66211.145	0	0
127	202207	Sobres	66629.859	0	0
128	NULL	Sobres	342976.937	1	0
129	202201	Teléfonos	426135.411	0	0
130	202202	Teléfonos	482318.415	0	0
131	202203	Teléfonos	459291.045	0	0
132	202204	Teléfonos	529101.877	0	0
133	202205	Teléfonos	620831.243	0	0
134	202206	Teléfonos	562984.357	0	0
135	202207	Teléfonos	652478.505	0	0
136	NULL	Teléfonos	3733140.853	1	0
137	NULL	NULL	23892467....	1	1
138	202201	NULL	3062557.627	0	1
139	202202	NULL	2775158.389	0	1
140	202203	NULL	3356793.939	0	1
141	202204	NULL	3282574.037	0	1
142	202205	NULL	3726152.269	0	1
143	202206	NULL	3880048.097	0	1
144	202207	NULL	3809183.380	0	1

Group
By cube

Pivot Examples GROUPING Y GROUP BY CUBE

SELECT * FROM

```
(
SELECT
CASE T.PER_GROUP WHEN 0 THEN T.PERIODO ELSE 'SUBTOTAL' END AS
PERIODO,
CASE T.SUBCAT_GROUP WHEN 0 THEN T.SUBCATEGORIA ELSE 'TOTAL' END
AS SUBCAT_GROUP,
T.VALVENTA
FROM
```

```
(
SELECT
    FORMAT(P.FECPEDID, 'yyyyMM') AS PERIODO,
    PR.NOMSUBCAT AS SUBCATEGORIA,
    SUM(P.MTOVALVTA) AS VALVENTA,
    GROUPING(FORMAT(P.FECPEDID, 'yyyyMM')) AS PER_GROUP,
    GROUPING(PR.NOMSUBCAT) AS SUBCAT_GROUP
FROM HD_PEDIDO P
LEFT JOIN MM_PRODUCTO PR ON P.CODPROD = PR.CODPROD
WHERE FORMAT(P.FECPEDID, 'yyyy') = 2022
GROUP BY CUBE (FORMAT(P.FECPEDID, 'yyyyMM'),
PR.NOMSUBCAT)
) T
```

) T2

PIVOT

```
(
SUM(VALENTA)
FOR PERIODO IN
([202201],[202202],[202203],[202204],[202205],[202206],[202207],
[SUBTOTAL])
) AS TPivot
```



Universidad
Nacional de
Cajamarca
"Norte de la Universidad Peruana"

	SUBCAT_GROUP	202201	202202	202203	202204	202205	202206	202207	SUBTOTAL
1	Accesorios	163048.254	220518.466	203520.084	170926.490	194759.270	279514.966	329991.189	1562278.719
2	Almacenamiento	244001.299	185737.231	263000.077	239153.415	212850.919	256670.419	319555.245	1720968.605
3	Archivadores	84633.150	85860.818	174629.654	105814.098	93446.726	138213.982	127021.099	809619.527
4	Arte	78555.059	84604.055	90367.311	89031.573	118200.595	119860.743	123605.257	704224.593
5	Artefactos	358529.891	228574.973	273634.491	274096.523	399777.615	341710.323	341567.234	2217891.050
6	Copiadoras	297566.760	240485.108	398522.882	505165.988	519560.694	459887.996	346598.350	2767787.778
7	Etiquetas	15578.326	16263.774	23128.056	17529.074	28749.800	18629.494	18887.949	138766.473
8	Grapas	19008.393	21949.173	24821.503	21160.951	30573.693	38816.507	23813.342	180143.562
9	Librerías	434358.008	341281.860	419175.604	360843.336	466444.642	451086.318	370869.695	2844059.463
10	Máquinas	173600.921	115884.979	148806.857	133243.421	198113.775	195040.441	241064.478	1205754.872
11	Material	77488.915	70466.665	70911.441	78409.451	86195.169	111426.161	76296.598	571194.400
12	Mesas	155289.092	150541.960	299041.542	219664.960	101155.888	201753.654	97016.302	1224463.398
13	Mobiliario	114667.397	101765.025	99653.959	111342.087	107062.431	158514.867	145005.986	838011.752
14	Papel	74027.616	57474.846	86338.300	58548.696	107930.990	79538.184	96387.063	560245.695
15	Sillas	307269.856	336573.102	290111.076	314584.524	389817.734	400188.540	432395.229	2470940.061
16	Sobres	38799.279	34857.939	31840.057	53957.573	50681.085	66211.145	66629.859	342976.937
17	Teléfonos	426135.411	482318.415	459291.045	529101.877	620831.243	562984.357	652478.505	3733140.853
18	TOTAL	3062557.627	2775158.389	3356793.939	3282574.037	3726152.269	3880048.097	3809183.380	23892467.738



Referencias

- Pivot
 - <https://docs.microsoft.com/en-us/sql/t-sql/queries/from-using-pivot-and-unpivot?view=sql-server-ver15>
- Sub Consultas
 - <https://docs.microsoft.com/es-es/sql/relational-databases/performance/subqueries?view=sql-server-ver15>
- Grouping
 - <https://docs.microsoft.com/en-us/sql/t-sql/queries/select-group-by-transact-sql?view=sql-server-ver15>



**Universidad
Nacional de
Cajamarca**
"Norte de la Universidad Peruana"

Herramientas de Trabajo

Online



**SQL Server
Management Studio**

v. 19.1



Fin de la sesión

