

# SQL Programming

M70 – Bonus: Over

# OVER

# OVER

Determina las particiones y el orden de un conjunto de filas antes de que se aplique la función de ventana asociada.

Es decir, la cláusula OVER define una ventana o un conjunto de filas definido por el usuario en un conjunto de resultados de la consulta.

Una función de ventana calcula entonces un valor para cada fila de la ventana. Puede utilizar la cláusula OVER con funciones para calcular valores agregados tales como medias móviles, agregados acumulados, totales acumulados o N elementos superiores por resultados del grupo.



# Notas generales

Se pueden utilizar varias funciones de ventana en una sola consulta con una única cláusula FROM. La cláusula OVER de cada función puede diferir en particiones y también en orden.

Si no se especifica PARTITION BY, la función trata todas las filas del conjunto de resultados de la consulta como un único grupo.



# Limitaciones y restricciones

No se puede utilizar la cláusula OVER con la función de agregado CHECKSUM.

No se puede usar RANGE con <especificación de valor sin signo> PRECEDING o <especificación de valor sin signo> FOLLOWING.

Dependiendo de la función de clasificación, de agregado o analítica usada con la cláusula OVER, puede que no se admite la <cláusula ORDER BY> o la <cláusula ROWS y RANGE>.



# Funciones de categoría

Las funciones de categoría devuelven un valor de categoría para cada fila de una partición. Según la función que se utilice, algunas filas pueden recibir el mismo valor que otras.

Las funciones de categoría son no deterministas.

Transact-SQL proporciona las siguientes funciones de categoría:

<b>RANK</b>	<b>NTILE</b>
<b>DENSE_RANK</b>	<b>ROW_NUMBER</b>

# Funciones de categoría

## Sintaxis

```
SELECT PP.LastName, SO.[SalesOrderID],  
ROW_NUMBER() OVER(ORDER BY LastName) Orden,  
ROW_NUMBER() OVER(PARTITION BY LastName ORDER BY LastName) OrdenPersona  
FROM Sales.SalesOrderHeader SO  
INNER JOIN Sales.SalesPerson SP  
ON SO.SalesPersonID=SP.BusinessEntityID  
INNER JOIN Person.Person PP  
ON SP.BusinessEntityID=PP.BusinessEntityID  
WHERE 2013=YEAR(OrderDate) AND 3=DATEPART(QQ, OrderDate)  
AND SP.TerritoryID=4 AND SO.SalesOrderID BETWEEN 55265 AND 55315
```

```
-----  
LastName      SalesOrderID      Orden  OrdenPersona  
Ito            55265            1      1  
Ito            55282            2      2  
Ito            55290            3      3  
Mitchell      55315            4      1  
Mitchell      55285            5      2  
Mitchell      55271            6      3  
Mitchell      55276            7      4
```

# Funciones de agregado

Una función de agregado realiza un cálculo sobre un conjunto de valores y devuelve un solo valor. Con la excepción de COUNT, las funciones de agregado ignoran los valores NULL. Las funciones de agregado se suelen usar con la cláusula GROUP BY de la instrucción SELECT.

Todas las funciones de agregado son deterministas. En otras palabras, las funciones de agregado devuelven el mismo valor cada vez que se las llama con un conjunto específico de valores de entrada.

Vea [Funciones deterministas y no deterministas](#) para obtener más información sobre el determinismo de las funciones.

La cláusula OVER puede seguir todas las funciones de agregado excepto GROUPING o GROUPING\_ID.

Las funciones de agregado solo se pueden usar como expresiones en las situaciones siguientes:

La lista de selección de una instrucción SELECT (una subconsulta o una consulta externa).



## Funciones de agregado

Transact-SQL proporciona las siguientes funciones de agregado:

<b>AVG</b>	<b>MIN</b>
<b>CHECKSUM_AGG</b>	<b>SUM</b>
<b>COUNT</b>	<b>STDEV</b>
<b>COUNT_BIG</b>	<b>STDEVP</b>
<b>GROUPING</b>	<b>STRING_AGG</b>
<b>GROUPING_ID</b>	<b>VAR</b>
<b>MAX</b>	<b>VARP</b>

# Funciones de agregado

## Sintaxis

```
SELECT TerritoryID Territorio, SalesYTD AS TotalVentas,  
AVG(SalesYTD)  
OVER (PARTITION BY TerritoryID ORDER BY TerritoryID) AS Promedio  
FROM Sales.SalesPerson  
WHERE TerritoryID < 5
```

Territorio	TotalVentas	Promedio
1	1352577,1325	1500717,4224
1	1573012,9383	1500717,4224
1	1576562,1966	1500717,4224
2	3763178,1787	3763178,1787
3	3189418,3662	3189418,3662
4	4251368,5497	3354952,0833
4	2458535,6169	3354952,0833



# Funciones Analíticas

SQL Server admite estas funciones analíticas:

Las funciones analíticas calculan un valor agregado a partir de un grupo de filas. A diferencia de las funciones de agregado, estas funciones pueden devolver varias filas para cada grupo.

Puede usar las funciones analíticas para calcular medias móviles, totales acumulados, porcentajes o resultados de N valores superiores dentro de un grupo.

<b>CUME_DIST</b>	<b>LEAD</b>
<b>FIRST_VALUE</b>	<b>PERCENTILE_CONT</b>
<b>LAG</b>	<b>PERCENTILE_DISC</b>
<b>LAST_VALUE</b>	<b>PERCENT_RANK</b>

# Funciones Analíticas

## Sintaxis

```
SELECT BusinessEntityID, YEAR(QuotaDate) AS Anio,  
SalesQuota AS Cuota, LEAD(SalesQuota, 1, 0)  
OVER (ORDER BY YEAR(QuotaDate)) AS ProximaCuota  
FROM Sales.SalesPersonQuotaHistory  
WHERE BusinessEntityID = 275 AND YEAR(QuotaDate) IN (2011,2012);
```

BusinessEntityID	Anio	Cuota	ProximaCuota
275	2011	367000,00	556000,00
275	2011	556000,00	502000,00
275	2011	502000,00	550000,00
275	2012	550000,00	1429000,00
275	2012	1429000,00	1324000,00
275	2012	1324000,00	729000,00
275	2012	729000,00	0,00



# ¡Muchas gracias!

¡Sigamos trabajando!