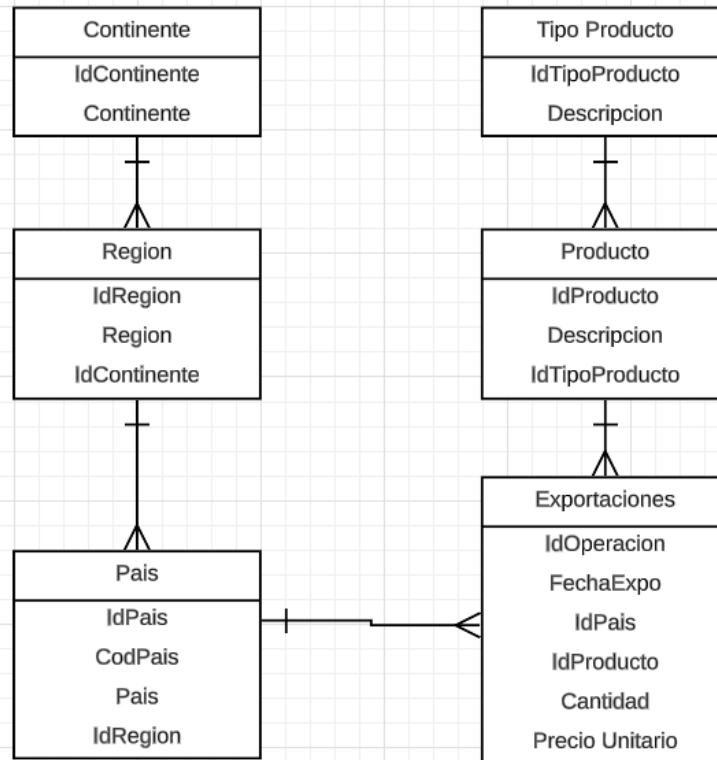


FUNCIONES ANALÍTICAS

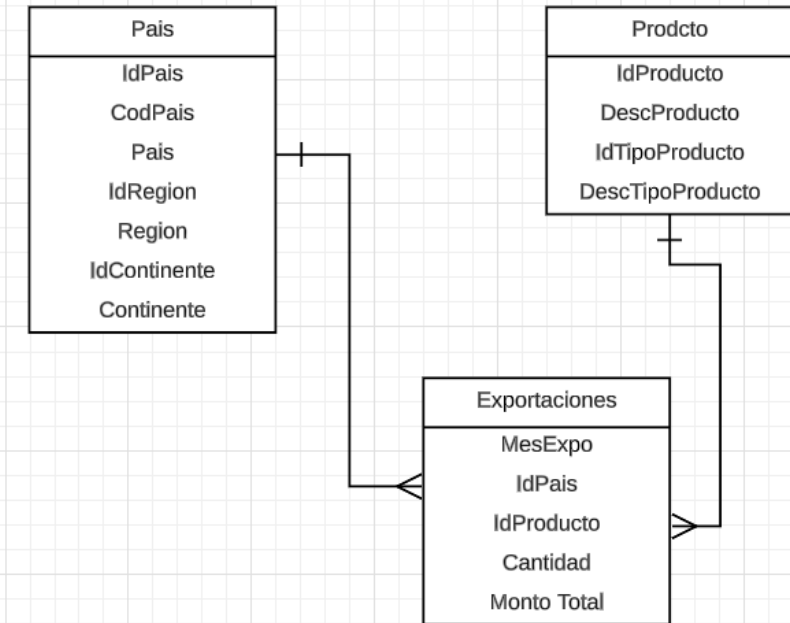
Funcionamiento y ejemplos en SQL SERVER

Del Modelo Relacional al Dimensional

Relacional



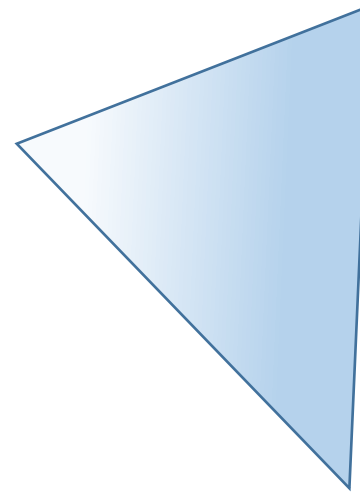
Dimensional



Funciones agregadas

- Las funciones agregadas realizan cálculos sobre un conjunto de valores y retorna un valor. A menudo se usan con las cláusulas *GROUP BY* y *HAVING*.

SELECT *columna1, ..., f(X)*
FROM tabla1, ...
[WHERE condicion]
GROUP BY *columna1, ...*
[HAVING condicion]



COUNT()
SUM()
AVG()
MIN()
MAX()

Funciones agregadas

Select Continente, Pais, SUM(montoTotal) Monto, SUM(cantidad) Cantidad,
sum(montoTotal)/sum(cantidad) Promedio

from paises p join exportaciones e
on p.IdPais = e.Idpais
group by p.Continente, p.Pais
order by 1, 2

Continente	Pais	Monto	Cantidad	Promedio
América	Argentina	19500	1160	16,81
América	Brasil	35000	2500	14,00
América	Chile	1500	600	2,50
América	Colombia	27800	1170	23,76
América	Estados Un	16400	2250	7,29
América	Perú	450	90	5,00
Europa	España	2700	780	3,46
Europa	Francia	1000	230	4,35
Europa	Grecia	150	50	3,00

Funciones agregadas

- Pero qué sucede si un gerente nos solicita ...

“Necesitaría un informe que muestre por mes el total exportado de cada país, en cada fila que me vaya mostrando el monto acumulado, el monto total exportado por mes y el porcentaje que representa el monto total del mes respecto del total exportado en el año.”



Funciones agregadas

Año	Mes	País	Monto	Acumulado	Total Mes	Porc.
2019	1	Colombia	4000	4000	11550	10,96%
2019	1	Argentina	2300	6300	11550	10,96%
2019	1	Estados Unidos	2100	8400	11550	10,96%
2019	1	España	1200	9600	11550	10,96%
2019	1	Brasil	1100	10700	11550	10,96%
2019	1	Perú	450	11150	11550	10,96%
2019	1	Chile	400	11550	11550	10,96%
2019	2	Estados Unidos	9000	20550	19700	18,70%
2019	2	Argentina	3200	23750	19700	18,70%
2019	2	Brasil	2300	26050	19700	18,70%
2019	2	Colombia	2200	28250	19700	18,70%
2019	2	España	1500	29750	19700	18,70%
2019	2	Francia	1000	30750	19700	18,70%
2019	2	Chile	500	31250	19700	18,70%
2019	3	Argentina	4000	35250	10000	9,49%
2019	3	Brasil	3200	38450	10000	9,49%
...



- Aquí es donde aparecen las ...

***"FUNCIONES
ANALÍTICAS..."***



Funciones Analíticas

Agreggates

SUM, COUNT, AVG, MAX, MIN

Ranking Functions

ROW_NUMBER(), RANK,
DENSE_RANK, NTITLE(n)

Offsets Functions

FIRST_VALUE, LAST_VALUE,
LAG(), LEAD()

Statistical functions

PERCENT_RANK, CUME_DIST,
PERCENTILE_DISC,
PERCENTILE_CONT

Funciones analíticas - Sintaxis

```
<window function> OVER  
  ([PARTITION BY <expression list>]  
  [ORDER BY <expression [ASC|DESC] list>]  
  [ROWS|RANGE <window frame>])
```

- ✓ La **función analítica**. Ej. AVG, FIRST, LEAD, PERCENTILE_RANK, etc.
- ✓ La cláusula **partición**. Ej PARTITION BY pais, producto
- ✓ La cláusula **order**. Ej ORDER BY job NULLS LAST
- ✓ El **window frame**. Ej. RANGE UNBOUNDED PRECEDING o ROWS UNBOUNDED FOLLOWING

Ejemplo con SUM()

- Cantidades exportadas acumuladas por país y por continente.

```
Select p.Continente, p.Pais, e.fechaExpo, e.cantidad,  
       sum(cantidad) over  
         (partition by p.IdPais) totalPais,  
       sum(cantidad) over  
         (partition by p.IdContinente) totalCont  
from paises p join exportaciones e  
  on p.IdPais = e.Idpais
```

Continente	Pais	Fecha	Cantidad	Total Pais	Total Continente
Europa	España	1/1/2019	380	780	1010
Europa	España	1/2/2019	400	780	1010
Europa	Francia	6/2/2019	230	230	1010
América	Estados Un	1/1/2019	200	2250	7770
América	Estados Un	1/3/2019	300	2250	7770
América	Estados Un	1/2/2019	350	2250	7770
América	Estados Un	2/4/2019	200	2250	7770
América	Estados Un	3/5/2019	300	2250	7770
América	Estados Un	4/6/2019	400	2250	7770
América	Estados Un	5/7/2019	500	2250	7770
América	Argentina	2/1/2019	100	1160	7770
América	Argentina	3/2/2019	400	1160	7770
América	Argentina	4/3/2019	200	1160	7770
América	Argentina	5/4/2019	200	1160	7770
América	Argentina	6/5/2019	150	1160	7770
América	Argentina	7/6/2019	110	1160	7770
América	Brasil	8/1/2019	200	2500	7770
América	Brasil	9/2/2019	100	2500	7770
...

Ejemplo con SUM() y *order by*

- Exportaciones acumuladas por país y por fecha, los totales GENERALES.

Select p.Pais, e.fechaExpo, e.cantidad,

sum(cantidad) over

(partition by p.IdPais ORDER BY e.fechaExpo) AcumPais,

sum(cantidad) over

(ORDER BY p.idpais, e.fechaExpo) AcumTotal

from paises p join exportaciones e on p.IdPais = e.Idpais

Pais	Fecha	Cantidad	Acum Pais	Acum Total
España	1/1/2019	380	380	380
España	1/2/2019	400	780	780
Francia	6/2/2019	230	230	1010
Estados Unidos	1/1/2019	200	200	1210
Estados Unidos	1/2/2019	350	550	1560
Estados Unidos	1/3/2019	300	850	1860
Estados Unidos	2/4/2019	200	1050	2060
Estados Unidos	3/5/2019	300	1350	2360
Estados Unidos	4/6/2019	400	1750	2760
Estados Unidos	5/7/2019	500	2250	3260
Argentina	2/1/2019	100	100	3360
Argentina	3/2/2019	400	500	3760
Argentina	4/3/2019	200	700	3960
Argentina	5/4/2019	200	900	4160
Argentina	6/5/2019	150	1050	4310
Argentina	7/6/2019	110	1160	4420
Brasil	8/1/2019	200	200	4620
Brasil	9/2/2019	100	300	4720
...

Ejemplo con AVG con *window frame*

- Exportaciones a países y el promedio de los últimos 3 meses.

```
select pais, fechaExpo, montoTotal,  
       avg(montoTotal) over  
         (partition by pais order by fechaExpo  
          ROWS BETWEEN 2 PRECEDING and CURRENT ROW) promedio  
from exportaciones e join países p on e.idPais = p.IdPais
```

Pais	FechaExpo	Monto	Promedio
Argentina	2/1/2019	2300	2.300
Argentina	3/2/2019	3200	2.750
Argentina	4/3/2019	4000	3.167
Argentina	5/4/2019	3800	3.667
Argentina	6/5/2019	4000	3.933
Argentina	7/6/2019	2200	3.333
Brasil	8/1/2019	1100	1.100
Brasil	9/2/2019	2300	1.700
Brasil	10/3/2019	3200	2.200
Chile	1/1/2019	400	400
Chile	1/2/2019	500	450
...

```
<window function> OVER  
([PARTITION BY <expression list>  
 [ORDER BY <expression [ASC|DESC] list>  
 [ROWS|RANGE <window frame>]])
```

ROWS BETWEEN n PRECEDING/FOLLOWING and n PRECEDING/FOLLOWING

ROWS n PRECEDING

CURRENT ROW

UNBOUNDED PRECEDING/FOLLOWING

Ranking Functions: ROW_NUMBER()

ROW_NUMBER() es una función que asigna un número entero secuencial a cada fila dentro de la partición del conjunto de resultados. El número empieza en 1 en la primera fila de cada partición.

- La cláusula PARTITION BY es opcional.
- La cláusula PARTITION BY divide el resultado en multiples conjuntos agrupados por el valor de la partición. Cada conjunto de resultados tendrá su propia numeración.
- Si no se especifica PARTITION BY, ROW_NUMBER trata TODAS las filas del query como un mismo grupo.

Ejemplo con ROW_NUMBER()

- Numeración de países exportadores por continente.

Nro	Continente	País	Monto
1	Europa	España	2700
2	Europa	Francia	1000
1	América	Brasil	35000
2	América	Colombia	27800
3	América	Argentina	19500
4	América	Estados Unid	16400
5	América	Chile	1500
6	América	Perú	450

```
SELECT ROW_NUMBER() OVER
      (PARTITION BY Idcontinente order by SUM(E.montoTotal) DESC) Nro,
      continente, pais, SUM(E.montoTotal)
FROM paises p join exportaciones E on p.idPais = E.idPais
GROUP BY Idcontinente, continente, Pais
```

Ranking Functions: RANK, DENSE_RANK

The RANK() and DENSE_RANK() Son funciones que asignan un ranking a cada una de las filas dentro de una partición.

Ej. Mostrar el ranking según cantidades mensuales exportadas por Continente.

```
SELECT RANK() OVER (PARTITION BY continente
                    ORDER BY cantidad DESC) Ranking,
       DENSE_RANK() OVER (PARTITION BY continente
                          ORDER BY cantidad DESC) Denso,
       continente, Pais,
       year(fechaExpo)*100 + day(fechaexpo) Periodo,
       cantidad
FROM exportaciones e join países p
                    on e.idPais = p.Idpais
```

Ranking	Denso	continente	Pais	Periodo	cantidad
1	1	América	Estados Un	201905	500
2	2	América	Argentina	201903	400
2	2	América	Brasil	201910	400
2	2	América	Estados Un	201904	400
2	2	América	Brasil	201917	400
6	3	América	Estados Un	201901	350
7	4	América	Estados Un	201901	300
7	4	América	Estados Un	201903	300
...
1	1	Europa	España	201901	400
2	2	Europa	España	201901	380
3	3	Europa	Francia	201906	230
4	4	Europa	Francia	201901	100
5	5	Europa	Grecia	201901	50

Ranking Functions: NTILE(n)

NTILE es una función que divide las filas en un número *n* de grupos y le asigna un número a cada grupo.

Ej. Clasificar o agrupar los países en 4 grupos en base a las cantidades totales exportadas.

```
SELECT NTILE(4) OVER
  (ORDER BY sum(cantidad) DESC) Bucket,
  Pais, sum(cantidad) Cantidades
FROM exportaciones e join países p
  on e.IdPais = p.IdPais
GROUP BY Pais
```

Bucket	Pais	Cantidades
1	Brasil	2500
1	Estados Unidos	2250
1	Colombia	1170
2	Argentina	1160
2	China	900
2	España	780
3	Chile	600
3	Francia	330
4	Perú	90
4	Grecia	50

Offset functions: FIRST_VALUE, LAST_VALUE

Las funciones FIRST_VALUE() y LAST_VALUE() retornan el primer y último valor de una partición.

Ej. Obtener las exportaciones mensuales por continente, junto a cada exportación indicar el país que menos exportó del mismo continente.

```
SELECT Continente, Pais,
       fechaExpo, cantidad,
       FIRST_VALUE(Pais) OVER
         (PARTITION by IdContinente
          ORDER BY cantidad ) Menor
FROM exportaciones e join países p
  on p.IdPais = e.IdPais
```

Continente	Pais	Fecha	Cantidad	Menor
Europa	Grecia	1/1/2019	50	Grecia
Europa	Francia	6/2/2019	230	Grecia
Europa	España	1/1/2019	380	Grecia
Europa	España	1/2/2019	400	Grecia
América	Perú	1/1/2019	90	Perú
América	Chile	1/2/2019	100	Perú
América	Colombia	25/6/2019	100	Perú
América	Colombia	26/7/2019	100	Perú
América	Colombia	29/10/2019	100	Perú
América	Argentina	2/1/2019	100	Perú
América	Brasil	9/2/2019	100	Perú
América	Brasil	18/11/2019	100	Perú

OFFSET: Mismo ejemplo con LAST_VALUE

```
SELECT Continente, Pais,
       fechaExpo, cantidad,
       LAST_VALUE(Pais) OVER
       (PARTITION by IdContinente
        ORDER BY cantidad) Mayor
FROM exportaciones e join países p
on p.IdPais = e.IdPais
```

Continente	Pais	fechaExpo	cantidad	Mayor
Europa	Grecia	1/1/2019	50	Grecia
Europa	Francia	1/1/2019	100	Francia
Europa	Francia	6/2/2019	230	Francia
Europa	España	1/1/2019	380	España
Europa	España	1/2/2019	400	España
América	Perú	1/1/2019	90	Perú
América	Brasil	18/11/2019	100	Colombia
América	Chile	1/2/2019	100	Colombia
América	Colombia	29/10/2019	100	Colombia
América	Argentina	2/1/2019	100	Colombia
América	Brasil	9/2/2019	100	Colombia
América	Colombia	21/2/2019	100	Colombia
América	Colombia	24/5/2019	110	Argentina
América	Brasil	14/7/2019	110	Argentina
América	Argentina	7/6/2019	110	Argentina
América	Colombia	27/8/2019	140	Brasil
América	Brasil	19/12/2019	140	Brasil
América



OFFSET: Mismo ejemplo con LAST_VALUE

LAST_VALUE() debe ser usado con un *window frame* *RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING*

```
SELECT Continente, Pais, fechaExpo, cantidad,  
       LAST_VALUE(Pais) OVER  
         (PARTITION by IdContinente ORDER BY cantidad  
          RANGE BETWEEN UNBOUNDED PRECEDING  
                   AND UNBOUNDED FOLLOWING) Mayor  
FROM exportaciones e join paises p  
  on p.IdPais = e.IdPais
```

Continente	Pais	fechaExpo	cantidad	Mayor
Europa	Grecia	1/1/2019	50	España
Europa	Francia	1/1/2019	100	España
Europa	Francia	6/2/2019	230	España
Europa	España	1/1/2019	380	España
Europa	España	1/2/2019	400	España
América	Perú	1/1/2019	90	Estados Unidos
América	Colombia	26/7/2019	100	Estados Unidos
	Chile	1/2/2019	100	Estados Unidos
	Argentina	2/1/2019	100	Estados Unidos
	Brasil	9/2/2019	100	Estados Unidos
	Argentina	5/4/2019	200	Estados Unidos
	Estados Unidos	2/4/2019	200	Estados Unidos
	Chile	1/1/2019	200	Estados Unidos
	Chile	1/3/2019	300	Estados Unidos
	Argentina	3/2/2019	400	Estados Unidos
	Brasil	10/3/2019	400	Estados Unidos
	Estados Unidos	5/7/2019	500	Estados Unidos

Offset functions: LAG(), LEAD()

Estas funciones proveen el acceso a un valor en una fila en una posición determinada respecto a la fila actual.

- ***LAG/LEAD(valor, [offset [,default]])***

Valor: Atributo a devolver .

Offset: Posición relativa a la fila actual. Valor por defecto 1.

Default: Valor a devolver si se va de los límites. Valor por defecto null.

Ejemplo con LAG() y LEAD()

Mostrar las exportaciones mensuales de cada país junto a sus meses anterior y posterior.

```
SELECT Pais, year(fechaExpo)*100+month(fechaExpo) Mes ,  
       LAG(cantidad) OVER (PARTITION BY pais  
                           ORDER BY fechaExpo) Anterior,  
                           cantidad,  
       LEAD(cantidad) OVER (PARTITION BY pais  
                           ORDER BY fechaExpo) Posterior  
FROM exportaciones e join paises p  
  on p.IdPais = e.IdPais
```

Pais	Mes	Anterior	cantidad	Posterior
Argentina	201901	NULL	100	400
Argentina	201902	100	400	200
Argentina	201903	400	200	200
Argentina	201904	200	200	150
Argentina	201905	200	150	110
Argentina	201906	150	110	NULL
Brasil	201901	NULL	200	100
Brasil	201902	200	100	400
Brasil	201903	100	400	NULL
Chile	201901	NULL	200	NULL
España	201901	NULL	380	400
España	201902	380	400	NULL

Funciones estadísticas. PERCENT_RANK y CUME_DIST

Estas funciones evalúan la **posición relativa** de un valor dentro de una partición o en un conjunto de resultados.

PERCENT_RANK calcula el ranking relativo, mientras que CUME_DIST calcula la distribución acumulada.

Continente	Pais	total	PerRank	CumeDist
América	Perú	450.00	0	0,166667
América	Chile	1500.00	0,2	0,333333
América	Estados Unidos	16400.00	0,4	0,5
América	Argentina	19500.00	0,6	0,666667
América	Colombia	27800.00	0,8	0,833333
América	Brasil	35000.00	1	1
Asia	China	100000.00	0	1
Europa	Grecia	150.00	0	0,333333
Europa	Francia	1500.00	0,5	0,666667
Europa	España	2700.00	1	1

```
SELECT Continente, Pais, sum(montoTotal) total,  
       PERCENT_RANK() OVER (PARTITION BY continente  
                             ORDER BY sum(montoTotal)) PerRank,  
       CUME_DIST() OVER (PARTITION BY continente  
                          ORDER BY sum(montoTotal)) CumeDist  
FROM exportaciones e join paises p  
  on p.IdPais = e.IdPais  
GROUP BY Continente, Pais
```

Cant.elem. <
/ n -1

Cant.elem. <=
/ n

PERCENTILE_CONT() y PERCENTILE_DISC()

Estas funciones calculan *percentiles* en base a una distribución continua o discreta del valor de una columna, respectivamente.

Recordatorio: El **percentil** es una medida de posición usada en estadística que indica sobre un conjunto de datos ordenados, el valor de la variable por debajo del cual se encuentra un [porcentaje](#) dado de observaciones en un grupo. Por ejemplo, el percentil 30 es el valor en el cual se encuentran el 30 % de las observaciones y luego el 70 % restante. El percentil 50 es la **mediana**.

Sintaxis

```
PERCENTILE_DISC ( numeric_literal )  
    WITHIN GROUP ( ORDER BY order_by_expression [ ASC | DESC ] )  
    OVER ( [ <partition_by_clause> ] )
```

PERCENTILE_CONT() y PERCENTILE_DISC()

Ej. Calcular el percentil 30 de los montos exportados por países.

```
SELECT pais, SUM(montoTotal) Montos,  
       PERCENTILE_CONT(0.3)  
         WITHIN GROUP(ORDER BY sum(montoTotal) desc  
                        OVER () AS perCont,  
       PERCENTILE_DISC(0.3)  
         WITHIN GROUP(ORDER BY sum(montoTotal) desc  
                        OVER () AS perDisc  
FROM paises p join exportaciones e  
   on e.IdPais = p.IdPais  
GROUP BY pais
```

pais	Montos	perCont	perDisc
China	100000.00	21990,00	27800.00
Brasil	35000.00	21990,00	27800.00
Colombia	27800.00	21990,00	27800.00
Argentina	19500.00	21990,00	27800.00
Estados Unidos	16400.00	21990,00	27800.00
España	2700.00	21990,00	27800.00
Chile	1500.00	21990,00	27800.00
Francia	1500.00	21990,00	27800.00
Perú	450.00	21990,00	27800.00
Grecia	150.00	21990,00	27800.00

PREGUNTAS

