
SQL - Funciones de Agregación y Agrupamiento



Bibliografía

- ▣ RIVERO C. Enrique, *et.al.* ***Introducción al SQL para Usuarios y Programadores*** (2^o edic.). Thompson. [cap 9-10].

I. Funciones de Agregación

➔ *Introducción*

- También conocidas como funciones colectivas, permiten obtener un solo valor como resultado de aplicar una determinada **operación** (suma, media, máximo, etc) a los valores contenidos en una columna.

```
funcAgregación ([DISTINCT] expr)
```

- Operan sobre una o varias columnas de la tabla
- Sus argumentos son los valores de la(s) columna(s)
- **DISTINCT**: Indica que antes de aplicar la función al conjunto de valores, elimine los valores repetidos si los hubiera.
- Devuelven **un único valor (colectivo)**.
- En la cláusula **SELECT** se pueden especificar más de una función de agregación.
- Ejemplos: Si A y B son columnas numéricas:
 - **MAX**(A)
 - **MAX**(3*A)
 - **MAX**((A+B)/2)
 - **SELECT MAX**(A) , **MAX**((A+B)/2) **FROM** ...
- **Ejemplo:** *Hallar el salario máximo para el conjunto de todos los empleados.*

```
SELECT MAX(SALAR) AS MAXIMO FROM TEMPLE
```

I. Funciones de Agregación

➔ *Funciones disponibles*

□ **Funciones disponibles**

- Todas estas funciones, excepto COUNT(*) eliminan cualquier valor Nulo en sus argumentos.

FUNCIÓN	TIPO DE DATOS	DESCRIPCIÓN
AVG	Numérico	Halla el valor medio de la columna
COUNT	Indiferente	Halla el número de valores de la columna
COUNT (*)	Indiferente	Halla el número de filas incluyen aquellas con Nulos
MAX	Numérico	Halla el valor máximo de la columna
MIN	Numérico	Halla el valor mínimo de la columna
STDDEV	Numérico	Halla la desviación típica de la columna
SUM	Numérico	Halla la suma
VARIANCE	Numérico	Halla la varianza

I. Funciones de Agregación

➔ Introducción

- Permiten obtener un **único** valor colectivo (resultado de aplicar la función a una lista de valores de la columna) al aplicar una determinada **operación** (suma, media, máximo, etc) a los valores contenidos en una columna.


```
funcAgregación ([DISTINCT] expr)
```

- **Ejemplo:** Hallar el salario máximo para el conjunto de todos los empleados.

```
SELECT MAX(SALAR) AS MAXIMO FROM TEMPLE
```

TEMPLE	NUMEM	NUMDE	SALAR
	110	121	1850
	120	112	2100

- 1.- Para cada fila de la tabla **se evalúa la expresión** que se especifica como argumento (SALAR).
- 2.- **Elimina** los valores **nulos** obtenidos al evaluar la expresión.
- 3.- Finalmente, **se ejecuta la función de agregación** sobre los valores del conjunto
- 4.- El resultado **es una única tabla con UNA única fila.**



Resultado	MAXIMO
	4300

I. Funciones de Agregación

➔ Introducción

- Permiten obtener un **único** valor colectivo (resultado de aplicar la función a una lista de valores de la columna) al aplicar una determinada **operación** (suma, media, máximo, etc) a los valores contenidos en una columna.

```
funcAgregación ([DISTINCT] expr)
```

- **Ejemplo:** Hallar el salario máximo y mínimo para el conjunto de todos los empleados.

```
SELECT MAX(SALAR) AS MAXIMO, MIN(SALAR) AS MINIMO FROM TEMPLE
```

TEMPLE	NUMEM	NUMDE	SALAR
	110	121	1850
	120	112	2100



Resultado	MAXIMO	MINIMO
	4300	600

- 1.- Para cada fila **se evalúa la expresión** que se especifica como argumento (SALAR).
- 2.- **Elimina** los valores **nulos** obtenidos al evaluar la expresión.
- 3.- Finalmente, **se ejecuta la función de agregación** sobre los valores del conjunto
- 4.- El resultado **es una única tabla con DOS columnas y UNA fila.**

I. Funciones de Agregación

→ Consideraciones

```
funcAgregación ([DISTINCT] expr)
```

- ❑ La función de agregación se aplica al conjunto de valores que obtienen como resultado de evaluar **expresión** y que no sean **Nulos**.
- ❑ En caso de que el conjunto de valores sea vacío:
 - COUNT: 0
 - Otras: **Nulo**
- ❑ En la cláusula **SELECT**, **puede haber más de una función** de agregación.
- ❑ En la cláusula **SELECT**, **no pueden aparecer valores no colectivos**, es decir, conjuntos de valores individuales (columnas), ya que es imposible generar una tabla con una fila con un único valor en la columna `MAX (SALAR)` y un conjunto de valores individuales para la columna (`NUMEM`)

IMPORTANTE!!

```
SELECT NUMEM, MAX (SALAR) AS MAXIMO FROM TEMPLE
```

ERROR!!

II. Funciones de Agrupamiento

➔ *Introducción*

- Las funciones de agregación se suelen utilizar combinadas con la cláusula de agrupamiento `GROUP BY`. El objetivo es formar grupos de filas de acuerdo a un determinado criterio (especificado en la cláusula `GROUP BY`) para aplicarles luego una función de agregación.

```
SELECT [DISTINCT|ALL] {*[exprColumna[AS nuevoNombre]][,...]}  
FROM nombreTabla [alias][,...]  
[WHERE predicado]  
[GROUP BY exprAgrupamiento] [HAVING condición]  
[ORDER BY {[nombreColumna[DESC]][,...]}]
```

Cláusula de agrupamiento

- **Ejemplo:** *Hallar el número y salario medio de cada departamento*

```
SELECT NUMDE, AVG(SALAR) AS SALARIO_MEDIO  
FROM TEMPLE  
GROUP BY NUMDE  
ORDER BY 1
```


II. Funciones de Agrupamiento

→ *Funcionamiento*

- **Ejemplo:** Hallar el número y salario medio de cada departamento

```
SELECT NUMDE, AVG(SALAR) AS SALARIO_MEDIO
FROM TEMPLE
GROUP BY NUMDE
ORDER BY 1
```

Los pasos que sigue el SGBD son:

1.- **FROM:** Considerar la tabla **TEMPLE**

2.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMDE**).

3.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:

- **NUMDE:** Todas tienen el mismo valor.
- **AVG(SALAR):** Se calcula el resultado de la función de

agregación.

Por cada grupo se crea una fila en la tabla de resultados.

4.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMDE	SALAR
	110	121	1850
	120	112	2100
	130	112	1750
	150	121	2650
	160	111	1850

II. Funciones de Agrupamiento

➔ *Funcionamiento*

- ❑ **Ejemplo:** Hallar el número y salario medio de cada departamento

```
SELECT NUMDE, AVG(SALAR) AS SALARIO_MEDIO
FROM TEMPLE
GROUP BY NUMDE
ORDER BY 1
```

Los pasos que sigue el SGBD son:

1.- **FROM:** Considerar la tabla **TEMPLE**

2.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMDE**).

3.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:

- **NUMDE:** Todas tienen el mismo valor.
- **AVG(SALAR):** Se calcula el resultado de la función de agregación.

Por cada grupo se crea una fila en la tabla de resultados.

4.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMDE	SALAR
	110	121	1850
	150	121	2650
	120	112	2100
	130	112	1750
	160	111	1850

II. Funciones de Agrupamiento

→ *Funcionamiento*

- **Ejemplo:** Hallar el número y salario medio de cada departamento

```
SELECT NUMDE, AVG(SALAR) AS SALARIO_MEDIO
FROM TEMPLE
GROUP BY NUMDE
ORDER BY 1
```

Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**
- 2.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMDE**).
- 3.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMDE:** Todas tienen el mismo valor.
 - **AVG(SALAR):** Se calcula el resultado de la función de agregación.
- 4.- **ORDER BY:** se ordenan las filas de la tabla resultante.

Por cada grupo se crea una fila en la tabla de resultados.

TEMPLE	NUMEM	NUMDE	SALAR
	110	121	1850
	150	121	2650
	120	112	2100
	130	112	1750
	160	111	1850



Resultado	NUMDE	SALARIO_MEDIO
	121	2250
	112	1950
	111	1850

II. Funciones de Agrupamiento

➔ *Funcionamiento*

- ❑ **Ejemplo:** Hallar el número y salario medio de cada departamento

```
SELECT NUMDE, AVG(SALAR) AS SALARIO_MEDIO
FROM TEMPLE
GROUP BY NUMDE
ORDER BY 1
```

Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**
- 2.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMDE**).
- 3.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMDE:** Todas tienen el mismo valor.
 - **AVG(SALAR):** Se calcula el resultado de la función de agregación.

Por cada grupo se crea una fila en la tabla de resultados.

- 4.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMDE	SALAR
	160	111	1850
	120	112	2100
	130	112	1750
	150	121	2650
	110	121	1850



Resultado	NUMDE	SALARIO_MEDIO
	111	1850
	112	1950
	121	2250

II. Funciones de Agrupamiento

→ Ejemplo

- **Ejemplo:** Hallar el nº de empleados de los departamentos 112 y 121 ordenados por departamento.



Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**
 - 2.- **WHERE:** Elimina aquellas filas que no cumplen el predicado.
 - 3.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMDE**).
 - 4.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMDE:** Todas tienen el mismo valor.
 - **COUNT(*):** Se calcula el resultado de la función de agregación.
- Por cada grupo se crea una fila en la tabla de resultados.
- 5.- **ORDER BY:** se ordenan las filas de la tabla resultante.

II. Funciones de Agrupamiento

➔ *Consideraciones*

- Según la definición dada anteriormente:

Las funciones de agregación se suelen utilizar combinadas con la cláusula de agrupamiento `GROUP BY`. El objetivo es formar grupos de filas de acuerdo a un determinado criterio (especificado en la cláusula `GROUP BY`) para aplicarles luego una función de agregación.

- Entonces, cuando en una sentencia se utilizan funciones de agregación (max, min, etc) y no se indica la cláusula `GROUP BY`, ¿se forman grupos de filas?
 - **Ejemplo:** Hallar el máximo salario de todos los empleados

```
SELECT MAX(SALAR) AS MAXIMO  
FROM TEMPLE
```

III. Cláusula HAVING

➔ *Introducción*

- Se utiliza para **descartar grupos de filas**.

```
SELECT [DISTINCT|ALL] { * | [exprColumna[AS nuevoNombre]] [, ...] }  
FROM nombreTabla [alias] [, ...]  
[WHERE predicado]  
[GROUP BY exprAgrupamiento] [HAVING condición]  
[ORDER BY { [nombreColumna[DESC]] [, ...] }]
```

Predicado que han de cumplir los grupos

- Únicamente tiene sentido (y se puede aplicar) cuando existe en la sentencia la cláusula GROUP BY.
- Después de haber creado los grupos de filas, según la cláusula GROUP BY, se eliminan aquéllos que no cumplan el predicado especificado en la cláusula HAVING.
- Es equivalente a la cláusula WHERE, pero aplicada a grupos de filas en lugar de a filas.

III. Cláusula HAVING

➔ *Introducción*

- Se utiliza para descartar grupos de filas.

```
SELECT [DISTINCT|ALL] {*[exprColumna[AS nuevoNombre]][,...]}  
FROM nombreTabla [alias][,...]  
[WHERE predicado]  
[GROUP BY exprAgrupamiento] [HAVING condición]  
[ORDER BY {[nombreColumna[DESC]][,...]}]
```

Predicado que han de cumplir los grupos

- **Ejemplo:** *Hallar el salario máximo y mínimo para cada grupo de empleados con igual número de hijos, y sólo si el salario máximo del grupo excede a 2000 euros.*



III. Cláusula HAVING

→ Ejemplo

- **Ejemplo:** Hallar el salario máximo y mínimo para cada grupo de empleados con igual número de hijos y que ganen más de 800 euros al mes, y sólo si hay más de 2 empleados en el grupo y el salario máximo del grupo excede a 2000 euros.



Los pasos que sigue el SGBD son:

1.- **FROM:** Considerar la tabla **TEMPLE**.

2.- **WHERE:** Elimina las filas que no cumplen el predicado.

3.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMHI**).

4.- **HAVING:** Elimina los grupos de filas que no cumplen el predicado.

5.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:

- NUMHI.

- **MAX**(SALAR): Se calcula el resultado de la función de agregación.

- **MIN**(SALAR): Se calcula el resultado de la función de agregación.

Por cada grupo se crea una fila en la tabla de resultados.

6.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMHI	SALAR
	110	3	1850
	120	1	2100
	130	2	1750
	150	2	2650
	160	2	1850

	550	0	600

III. Cláusula HAVING

→ Ejemplo

- **Ejemplo:** Hallar el salario máximo y mínimo para cada grupo de empleados con igual número de hijos y que ganen más de 800 euros al mes, y sólo si hay 2 ó más empleados en el grupo y el salario máximo del grupo excede a 2000 euros.

```
SELECT NUMHI, MAX(SALAR), MIN(SALAR) FROM TEMPLE
WHERE SALAR > 800
GROUP BY NUMHI
HAVING COUNT(*) > 2 AND MAX(SALAR) > 2000
ORDER BY NUMHI
```

Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**.
- 2.- **WHERE:** Elimina las filas que no cumplen el predicado.
- 3.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMHI**).
- 4.- **HAVING:** Elimina los grupos de filas que no cumplen el predicado.
- 5.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMHI.**
 - **MAX(SALAR):** Se calcula el resultado de la función de agregación.
 - **MIN(SALAR):** Se calcula el resultado de la función de agregación.

Por cada grupo se crea una fila en la tabla de resultados.

- 6.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMHI	SALAR
	110	3	1850
	120	1	2100
	130	2	1750
	150	2	2650
	160	2	1850

	550	0	600

III. Cláusula HAVING

→ Ejemplo

- **Ejemplo:** Hallar el salario máximo y mínimo para cada grupo de empleados con igual número de hijos y que ganen más de 800 euros al mes, y sólo si hay más de 2 empleados en el grupo y el salario máximo del grupo excede a 2000 euros.

```
SELECT NUMHI, MAX(SALAR), MIN(SALAR) FROM TEMPLE
WHERE SALAR>800
GROUP BY NUMHI
HAVING COUNT(*)>2 AND MAX(SALAR)>2000
ORDER BY NUMHI
```

Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**.
 - 2.- **WHERE:** Elimina las filas que no cumplen el predicado.
 - 3.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMHI**).
 - 4.- **HAVING:** Elimina los grupos de filas que no cumplen el predicado.
 - 5.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMHI.**
 - **MAX(SALAR):** Se calcula el resultado de la función de agregación.
 - **MIN(SALAR):** Se calcula el resultado de la función de agregación.
- Por cada grupo se crea una fila en la tabla de resultados.
- 6.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMHI	SALAR
	110	3	1850
	120	1	2100
	130	2	1750
	150	2	2650
	160	2	1850

III. Cláusula HAVING

→ Ejemplo

- **Ejemplo:** Hallar el salario máximo y mínimo para cada grupo de empleados con igual número de hijos y que ganen más de 800 euros al mes, y sólo si hay más de 2 empleados en el grupo y el salario máximo del grupo excede a 2000 euros.

```
SELECT NUMHI, MAX(SALAR), MIN(SALAR) FROM TEMPLE
WHERE SALAR>800
GROUP BY NUMHI
HAVING COUNT(*)>2 AND MAX(SALAR)>2000
ORDER BY NUMHI
```

Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**.
 - 2.- **WHERE:** Elimina las filas que no cumplen el predicado.
 - 3.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMHI**).
 - 4.- **HAVING:** Elimina los grupos de filas que no cumplen el predicado.
 - 5.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMHI.**
 - **MAX(SALAR):** Se calcula el resultado de la función de agregación.
 - **MIN(SALAR):** Se calcula el resultado de la función de agregación.
- Por cada grupo se crea una fila en la tabla de resultados.
- 6.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMHI	SALAR
	110	3	1850
	120	1	2100
	130	2	1750
	150	2	2650
	160	2	1850

III. Cláusula HAVING

→ Ejemplo

- **Ejemplo:** Hallar el salario máximo y mínimo para cada grupo de empleados con igual número de hijos y que ganen más de 800 euros al mes, y sólo si hay más de 2 empleados en el grupo y el salario máximo del grupo excede a 2000 euros.

```
SELECT NUMHI, MAX(SALAR), MIN(SALAR) FROM TEMPLE
WHERE SALAR>800
GROUP BY NUMHI
HAVING COUNT(*)>2 AND MAX(SALAR)>2000
ORDER BY NUMHI
```

Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**.
- 2.- **WHERE:** Elimina las filas que no cumplen el predicado.
- 3.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMHI**).
- 4.- **HAVING:** Elimina los grupos de filas que no cumplen el predicado.
- 5.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMHI.**
 - **MAX(SALAR):** Se calcula el resultado de la función de agregación.
 - **MIN(SALAR):** Se calcula el resultado de la función de agregación.
- 6.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMHI	SALAR
	130	2	1750
	150	2	2650
	160	2	1850



Result	NUMHI	MAX(SALAR)	MIN(SALAR)
	2	2650	1750

III. Cláusula HAVING

→ Ejemplo

- **Ejemplo:** Hallar el salario máximo y mínimo para cada grupo de empleados con igual número de hijos y que ganen más de 800 euros al mes, y sólo si hay más de 2 empleados en el grupo y el salario máximo del grupo excede a 2000 euros.

```
SELECT NUMHI, MAX(SALAR), MIN(SALAR) FROM TEMPLE
WHERE SALAR>800
GROUP BY NUMHI
HAVING COUNT(*)>2 AND MAX(SALAR)>2000
ORDER BY NUMHI
```

Los pasos que sigue el SGBD son:

- 1.- **FROM:** Considerar la tabla **TEMPLE**.
- 2.- **WHERE:** Elimina las filas que no cumplen el predicado.
- 3.- **GROUP BY:** Construir grupos de filas que tengan igual valor en la expresión de agrupamiento (**NUMHI**).
- 4.- **HAVING:** Elimina los grupos de filas que no cumplen el predicado.
- 5.- **SELECT:** evaluar las expresiones del **SELECT** para las filas de cada grupo:
 - **NUMHI.**
 - **MAX(SALAR):** Se calcula el resultado de la función de agregación.
 - **MIN(SALAR):** Se calcula el resultado de la función de agregación.

Por cada grupo se crea una fila en la tabla de resultados.

6.- **ORDER BY:** se ordenan las filas de la tabla resultante.

TEMPLE	NUMEM	NUMHI	SALAR
	130	2	1750
	150	2	2650
	160	2	1850



Result	NUMHI	MAX(SALAR)	MIN(SALAR)
	2	2650	1750

IV. Errores comunes

➔ *Funciones de Agregación*

- ❑ En la cláusula `SELECT`, **no pueden aparecer valores no colectivos**, es decir, conjuntos de valores individuales (columnas), ya que es imposible generar una tabla con una fila con un único valor en la columna `MAX (SALAR)` y un conjunto de valores individuales para la columna `(NUMEM)`

```
SELECT NUMEM, MAX (SALAR) AS MAXIMO FROM TEMPLE
```

ERROR!!

- ❑ Solución: Utilizar sentencias select .

```
SELECT NUMEM, (SELECT MAX (SALAR) FROM TEMPLE) FROM TEMPLE
```

IV. Errores comunes

➔ *Funciones de Agregación*

- ❑ En el predicado, no pueden aparecer funciones de agregación con valores colectivos, ya que no es posible crear la tabla temporal para la evaluación del predicado

```
SELECT NUMEM  
FROM TEMPLE  
WHERE SALAR*1.5 > MAX(SALAR)
```

ERROR!!

- ❑ Solución: Utilizar sentencias select subordinadas

```
SELECT NUMEM  
FROM TEMPLE  
WHERE SALAR*1.5 > (SELECT MAX(SALAR) FROM TEMPLE)
```


IV. Errores comunes

➔ *Funciones de Agrupamiento*

- Las funciones de agregación se suelen utilizar combinadas con la cláusula de agrupamiento `GROUP BY`. El objetivo es formar grupos de filas de acuerdo a un determinado criterio, los atributos de agrupamiento (que en la cláusula `select`, son valores colectivos, como los que proporciona la función de agregación.. Estos atributos se especifican en la cláusula `GROUP BY`.
 - En cada grupo de filas siempre se cumple que todos los valores de los atributos de agrupamiento son **iguales**.
 - Las expresiones indicadas en la cláusula `SELECT` **se ejecutarán para cada grupo**.
 - Ejemplos:

```
SELECT NUMDE, AVG(SALAR) AS SALARIO_MEDIO  
FROM TEMPLE  
GROUP BY NUMDE
```

```
SELECT NUMDE, AVG(SALAR) AS SALARIO_MEDIO, SALAR  
FROM TEMPLE  
GROUP BY NUMDE
```

ERROR!!

V. Ejercicios
