

Introducción a Base de Datos

SQL

Funciones de Agregación

```
SELECT    [DISTINCT]
atrib1    [AS otroNombre1],
atrib2    [AS otroNombre2],
.....,
atribN    [AS otroNombreN]

FROM      lista_de_tablas

[WHERE    condición]

[GROUP BY lista_atributos_que_agrupan

[HAVING   condición_para_el_grupo] ]

[ORDER BY lista_atributos]
```

Funciones de Agregación

Permiten buscar un único representante entre un conjunto de valores.
Las mismas pueden ser:

- Max
- Min
- Avg (admite quitar repetidos antes de aplicar)
- Count (admite quitar repetidos antes de aplicar)
- Sum (admite quitar repetidos antes de aplicar)

Se utilizan en la clausulas SELECT o HAVING

IMPORTANTE

**Toda función de agregación, salvo COUNT,
no se aplica sobre atributos NULL
(se descartan antes).**

**Si se aplica sobre un conjunto vacío devuelve una única
tupla con valor NULL**

IMPORTANTE

**Si es COUNT(nombreCol) no se aplica sobre atributos NULL
(se descartan antes).**

**Si se aplica sobre un conjunto vacío devuelve una única
tupla con valor 0**

Si es COUNT(*) no se descartan los valores NULL

EMPLEADO		
Nombre	Sueldo	Edad
'Ana'	500	23
'Juan'	1200	34
'Clara'	2400	30
'Jose'	3000	34
'Maria'	1000	Null
'Ema'	2400	null

```
SELECT COUNT(sueldo)  
FROM empleado;
```



Rta: 6

EMPLEADO		
Nombre	Sueldo	Edad
'Ana'	500	23
'Juan'	1200	34
'Clara'	2400	30
'Jose'	3000	34
'Maria'	1000	Null
'Ema'	2400	null

```
SELECT COUNT(DISTINCT sueldo)
FROM empleado;
```



Rta: 5

EMPLEADO		
Nombre	Sueldo	Edad
'Ana'	500	23
'Juan'	1200	34
'Clara'	2400	30
'Jose'	3000	34
'Maria'	1000	Null
'Ema'	2400	null

```
SELECT COUNT(edad)  
FROM empleado;
```



Rta: 4

EMPLEADO		
Nombre	Sueldo	Edad
'Ana'	500	23
'Juan'	1200	34
'Clara'	2400	30
'Jose'	3000	34
'Maria'	1000	Null
'Ema'	2400	null

```
SELECT COUNT(DISTINCT edad)  
FROM empleado;
```



Rta: 3

EMPLEADO		
Nombre	Sueldo	Edad
'Ana'	500	23
'Juan'	1200	34
'Clara'	2400	30
'Jose'	3000	34
'Maria'	1000	Null
'Ema'	2400	null

```
SELECT AVG(edad)  
FROM empleado;
```



Rta: 30.25

EMPLEADO		
Nombre	Sueldo	Edad
'Ana'	500	23
'Juan'	1200	34
'Clara'	2400	30
'Jose'	3000	34
'Maria'	1000	Null
'Ema'	2400	null

```
SELECT AVG(DISTINCT edad)  
FROM empleado;
```



Rta: 29

EMPLEADO		
Nombre	Sueldo	Edad
'Ana'	500	23
'Juan'	1200	34
'Clara'	2400	30
'Jose'	3000	34
'Maria'	1000	Null
'Ema'	2400	null

```
SELECT MAX(edad) , nombre
FROM empleado;
```

Si aparece una función de agregación en esas cláusulas, todo atributo que aparezca en ella debe **ser agregado** o participar de **GROUP BY**



SOPRESA	
nro	
	Null
	Null
	Null

```
SELECT MAX(nro) , MIN(nro) , AVG(nro) ,  
        SUM(nro) , COUNT(nro) , COUNT(*)  
FROM sorpresa;
```

Rta:

NULL NULL NULL NULL 0 3

Ejemplo

Nombre del ítem que arrojó la máxima medición histórica. Mostrar también dicho valor máximo

```
SELECT name, value  
FROM item JOIN measurement on code= item_code  
WHERE value = (SELECT max (value) FROM measurement)
```

Ejemplo

Notar que la siguiente consulta parece equivalente, pero no siempre da el mismo resultado:

```
SELECT name, value  
FROM item JOIN measurement on code= item_code  
ORDER BY value DESC  
LIMIT 1
```

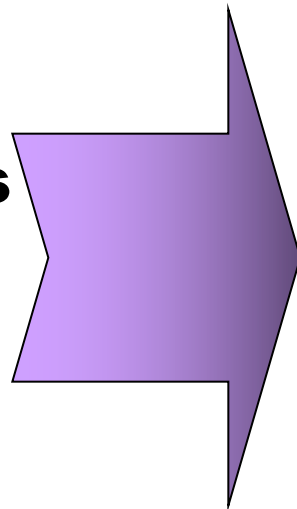
¿Por qué?

Clausula Group BY

SELECT **MAX**(precio)
FROM provee

Las funciones de
agregación se
aplican a las
tuplas
recuperadas en la
consulta

**Para aplicar
agregaciones a
“subgrupos de tuplas
recuperadas en la
consulta”
y NO al total de las
tuplas devueltas**



**Construir grupos a
través de la
cláusula group by
donde se indican
el/los atributo/s que
definen el subgrupo**

SELECT proveedor.nombre, direccion, **MAX**(precio)
FROM provee, proveedor
WHERE provee.nombre = proveedor.nombre
GROUP BY proveedor.nombre, direccion

Primero:
 Producto cartesiano

codigo	provee.nombre	precio	proveedor.nombre	direccion
10	IBM	350	Compucompras	Cordoba 1200
20	IBM	370	Compucompras	Cordoba 1200
30	IBM	270	Compucompras	Cordoba 1200
40	IBM	1100	Compucompras	Cordoba 1200
60	Compucompras	1100	Compucompras	Cordoba 1200
....				
100	IBM	2500	IBM	Catalinas 1100
100	Compucompras	2650	IBM	Catalinas 1100
100	Baidat	2550	IBM	Catalinas 1100
100	Florida I	2700	IBM	Catalinas 1100
100	Compushow	3200	IBM	Catalinas 1100
70	Florida I	5200	IBM	Catalinas 1100


```

SELECT proveedor.nombre, direccion, MAX(precio)
FROM provee, proveedor
WHERE provee.nombre = proveedor.nombre
GROUP BY proveedor.nombre, direccion

```

Segundo:

Selección de los
que coinciden en
nombre

codigo	provee.nombre	precio	proveedor.nombre	direccion
60	Compucompras	1100	Compucompras	Cordoba 1200
100	Compucompras	2650	Compucompras	Cordoba 1200
60	Baidat	1099	Baidat	Juramento 2500
100	Baidat	2550	Baidat	Juramento 2500
60	Florida I	1200	Florida I	Florida 50
100	Florida I	2700	Florida I	Florida 50
70	Florida I	5200	Florida I	Florida 50
100	Compushow	3200	Compushow	Florida 537
10	IBM	350	IBM	Catalinas 1100
20	IBM	370	IBM	Catalinas 1100
30	IBM	270	IBM	Catalinas 1100
40	IBM	1100	IBM	Catalinas 1100
100	IBM	2500	IBM	Catalinas 1100

SELECT proveedor.nombre, direccion, MAX(precio)
FROM provee, proveedor
WHERE provee.nombre = proveedor.nombre
GROUP BY proveedor.nombre, direccion

Tercero:

**Agrupamiento de tuplas
 en subgrupos
 formado por atributos
 nombre y direccion**

proveedor.nombre	direccion	precio	codigo	provee.nombre
Compucompras	Cordoba 1200	1100	60	Compucompras
		2650	100	Compucompras
Baidat	Juramento 2500	1099	60	Baidat
		2550	100	Baidat
Florida I	Florida 50	1200	60	Florida I
		2700	100	Florida I
		5200	70	Florida I
Compushow	Florida 537	3200	100	Compushow
IBM	Catalinas 1100	350	10	IBM
		370	20	IBM
		270	30	IBM
		1100	40	IBM
		2500	100	IBM

SELECT proveedor.nombre, direccion, MAX(precio)
FROM provee, proveedor
WHERE provee.nombre = proveedor.nombre
GROUP BY proveedor.nombre, direccion

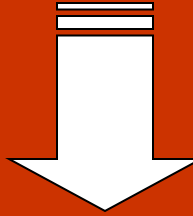
Final:

Proyección solicitada

NUEVA RELACION		
nombre	direccion	(max)
IBM	Catalinas 1100	2500
Compucompras	Cordoba 1200	2650
Baidat	Juramento 2500	2550
Florida I	Florida 50	5200
Compushow	Florida 537	3200

Atención

Cada grupo armado puede tener varias direcciones



SQL exige que los valores que no agrupan
estén afectados de una función de agregación
(para transformarlos en único valor).

```
SELECT proveedor.nombre, direccion , MAX( precio)  
FROM provee, proveedor  
WHERE provee.nombre = proveedor.nombre  
GROUP BY proveedor.nombre
```

MAL !!!

Clausula HAVING

Sirve para pedir condiciones sobre el grupo de tuplas generado. En esta cláusula pueden usarse FUNCIONES de agregación

Ejemplo: Mostrar nombre de proveedor, dirección y precio más caro que éste ofrezca, para aquellos proveedores que ofrezcan por lo menos tres productos

```
SELECT proveedor.nombre, direccion, MAX( precio)
FROM provee, proveedor
WHERE provee.nombre = proveedor.nombre
GROUP BY proveedor.nombre, direccion
HAVING COUNT( * ) > 2
```

Se producen las mismas etapas:

- a) Producto cartesiano
- b) Selección
- c) Agrupamiento
- d) .. Pero antes de proyectar hay que seleccionar subgrupos

Ejemplo: *Mostrar nombre de proveedor, dirección y precio más caro que éste ofrezca, para aquellos proveedores que ofrezcan por lo menos tres productos*

SELECT proveedor.nombre, direccion, MAX(precio)
FROM provee, proveedor
WHERE provee.nombre = proveedor.nombre
GROUP BY proveedor.nombre, direccion
HAVING COUNT(*) > 2

Primero,
Segundo,
Tercero: Idem

Ahora se van elegir
Subgrupos con > 3 tuplas

proveedor.nombre	direccion	precio	codigo	provee.nombre
Compucompras	Cordoba	1100	60	Compucompras
		2650	100	Compucompras
Baidat	Juramento	1099	60	Baidat
		2500	100	Baidat
Florida I	Florida	1200	60	Florida I
		2700	100	Florida I
		5200	70	Florida I
Compushow	Florida	537	100	Compushow
IBM	Catalinas	350	10	IBM
		370	20	IBM
		270	30	IBM
		1100	40	IBM
		2500	100	IBM

Ejemplo: *Mostrar nombre de proveedor, dirección y precio más caro que éste ofrezca, para aquellos proveedores que ofrezcan por lo menos tres productos*

SELECT proveedor.nombre, direccion, MAX(precio)
FROM provee, proveedor
WHERE provee.nombre = proveedor.nombre
GROUP BY proveedor.nombre, direccion
HAVING COUNT(*) > 2

Cuarto:

Quedan solo
Estos subgrupos

proveedor.nombre	direccion	precio	codigo	provee.nombre
Florida I	Florida 50	1200	60	Florida I
		2700	100	Florida I
		5200	70	Florida I
IBM	Catalinas 1100	350	10	IBM
		370	20	IBM
		270	30	IBM
		1100	40	IBM
		2500	100	IBM

Ejemplo: *Listar el nombre del proveedor, dirección y precio más caro que éste ofrezca, para aquellos proveedores que ofrezcan por lo menos tres productos*

```
SELECT proveedor.nombre, direccion, MAX( precio)  
FROM provee, proveedor  
WHERE provee.nombre = proveedor.nombre  
GROUP BY proveedor.nombre, direccion  
HAVING COUNT( * ) > 2
```

Final:
Proyección solicitada

NUEVA RELACION		
nombre	direccion	precio
IBM	Catalinas 1100	2500
Florida I	Florida 50	5200

Orden Real de Evaluación de las Cláusulas

1. Cláusula **FROM**
2. Cláusula **WHERE**
3. Cláusula **GROUP BY**
4. Cláusula **HAVING**
5. Cláusula **ORDER BY**
6. Cláusula **SELECT**