

# Tema 6. DML – DQL. Consultas

---

# Agrupaciones. GROUP BY

---

Es habitual realizar consultas en las que se quiera agrupar los datos con el objetivo de realizar cálculos o conocer datos sobre un determinado grupo o conjunto de datos. Para estos casos aparecen dos nuevas cláusulas que se añaden a la SELECT: GROUP BY y HAVING. A estas consultas habitualmente también se las llama consultas resumen

```
SELECT listaDeExpresiones FROM listaDeTablas  
[JOIN tablasRelacionadasYcondicionesDeRelacion]  
[WHERE condiciones]  
[GROUP BY grupos]  
        [HAVING condicionesDeGrupo]  
[ORDER BY columnas];
```

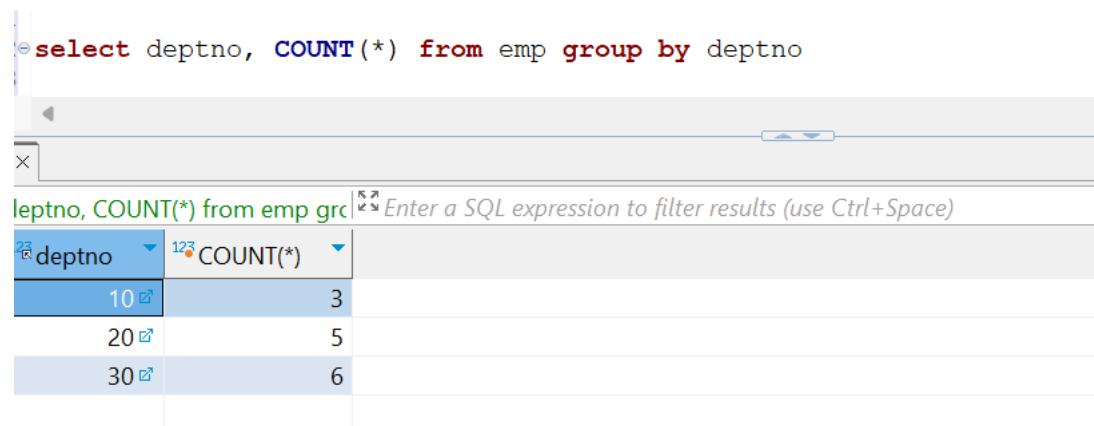
# Agrupaciones. GROUP BY

---

Si en una consulta queremos recuperar los valores de otros campos junto con el resultado de una función agregada, hemos de hacer que el resultado de dicha función se refiera a cada combinación exclusiva de valores de los campos que se van a obtener.

Esto se consigue con la cláusula GROUP BY

Es obligatorio que todos los campos que aparecen en la SELECT también aparezcan en el GROUP BY



The screenshot shows a SQL query editor with the following query:

```
select deptno, COUNT(*) from emp group by deptno
```

Below the query editor, the results are displayed in a table. The table has two columns: 'deptno' and 'COUNT(\*)'. The data is as follows:

deptno	COUNT(*)
10	3
20	5
30	6

# Agrupaciones. GROUP BY

---

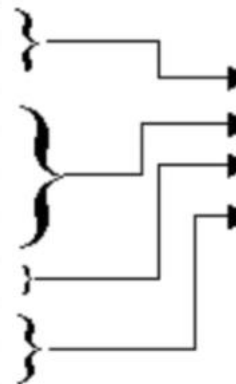
En este ejemplo aparece una consulta normal en la que se visualizan las filas de la tabla oficinas ordenadas por región, en este caso cada fila del resultado se corresponde con una sola fila de la tabla oficina, mientras que la otra consulta es una consulta resumen, en la que cada fila del resultado se corresponde con una o varias filas de la tabla oficina

```
SELECT oficina,region,ventas  
FROM oficinas  
ORDER BY region
```

oficina	region	ventas
24	centro	150.000 Pts
23	centro	
28	este	0 Pts
13	este	368.000 Pts
12	este	735.000 Pts
11	este	693.000 Pts
26	norte	
22	oeste	186.000 Pts
21	oeste	836.000 Pts

```
SELECT region,SUM(ventas)  
FROM oficinas  
GROUP BY region
```

region	SumaDeventas
centro	150000
este	1796000
norte	
oeste	1022000

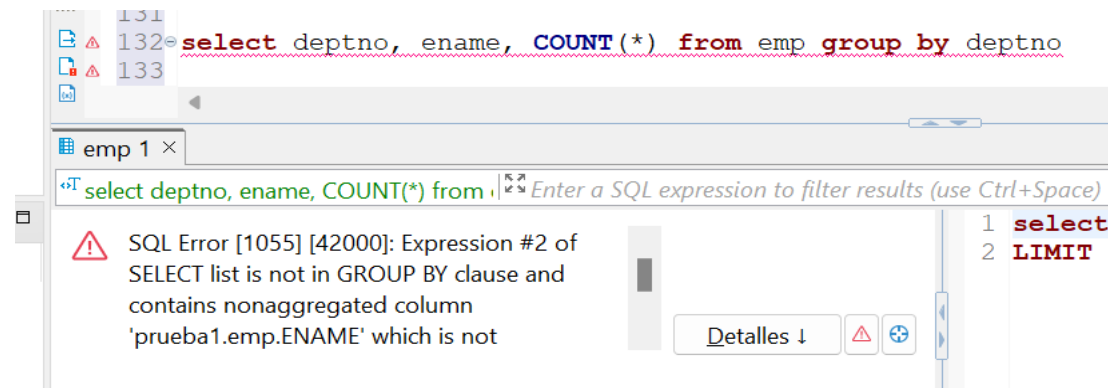


# Agrupaciones. GROUP BY

---

En el GROUP BY se indican las columnas por las que se agrupa. La siguiente consulta produciría error:

```
select deptno, ename, COUNT(*) from emp group by deptno
```



Ename no está disponible directamente ya que tiene valores distintos para los registros del mismo grupo -> Si usamos GROUP BY, en la SELECT sólo deben aparecer los campos del GROUP BY y funciones

# GROUP BY y Funciones de agregación

---

COUNT (\*)  
MAX (expresión)  
SUM (expresión)  
STDDEV (expresión)  
AVG(expresión)  
VARIANCE (expresión)  
MIN (expresión)

```
SELECT deptno, comm, SUM(sal)
FROM emp
GROUP BY deptno, comm;
```

<u>deptno</u>	<u>comm</u>	sum(sal)
10	NULL	8750.00
20	NULL	10875.00
30	NULL	3800.00
30	0.00	1500.00
30	300.00	1600.00
30	500.00	1250.00
30	1400.00	1250.00

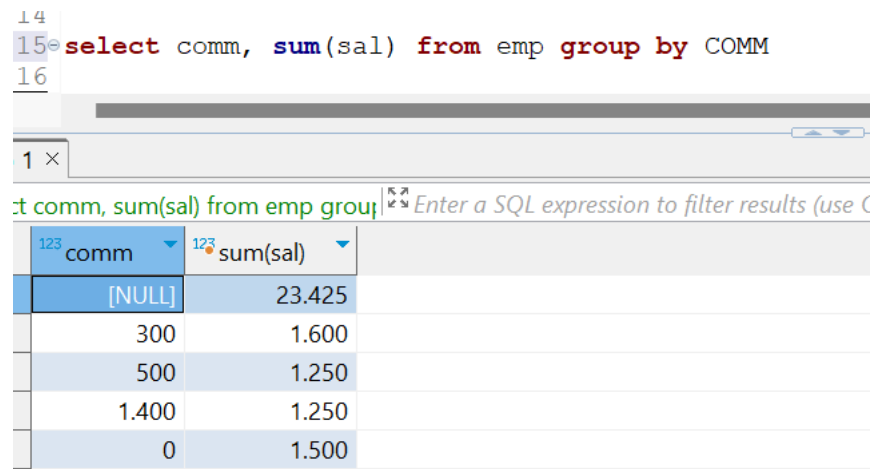
# GROUP BY. Null. Ejemplos

---

Todas las filas que tienen valor nulo en el campo de agrupación, pasan a formar un único grupo.

Es decir, considera el valor nulo como un valor cualquiera a efectos de agrupación

```
select comm, sum(sal) from emp group by COMM
```



The screenshot shows a SQL query editor with the following code:

```
14  
15 select comm, sum(sal) from emp group by COMM  
16
```

Below the editor, a table of results is displayed. The table has two columns: 'comm' and 'sum(sal)'. The first row shows a NULL value for 'comm' and a sum of 23.425. The subsequent rows show values for 'comm' (300, 500, 1.400, 0) and their corresponding sums (1.600, 1.250, 1.250, 1.500).

comm	sum(sal)
[NULL]	23.425
300	1.600
500	1.250
1.400	1.250
0	1.500

# GROUP BY. Ejemplos

---

```
131  
132 SELECT deptno, comm, COUNT(*) FROM emp GROUP BY deptno, comm;
```

mp 1 ×

ELECT deptno, comm, COUNT(\*) FROM *Enter a SQL expression to filter results (use Ctrl+Space)*

	<sup>123</sup> deptno ▼	<sup>123</sup> comm ▼	<sup>123</sup> COUNT(*) ▼
1	20 ↗	[NULL]	5
2	30 ↗	300	1
3	30 ↗	500	1
4	30 ↗	1.400	1
5	30 ↗	[NULL]	2



# GROUP BY. Ejemplos

```
32= SELECT deptno, job, ROUND(AVG(sal),2) as 'Salario Medio'
33 FROM emp
34 GROUP BY deptno, job;
35
```

CT deptno, job, ROUND(AVG(sal),2) Enter a SQL expression to filter results (use Ctrl+Space)

deptno	job	Salario Medio
20	CLERK	950
30	SALESMAN	1.400
20	MANAGER	2.975
30	MANAGER	2.850
10	MANAGER	2.450

```
132= SELECT job, deptno, ROUND(AVG(sal),2) as 'Salario Medio'
133 FROM emp
134 GROUP BY job, deptno;
135
```

emp 1 ×

SELECT job, deptno, ROUND(AVG(sal),2) Enter a SQL expression to filter results (use Ctrl+Space)

	job	deptno	Salario Medio
1	CLERK	20	950
2	SALESMAN	30	1.400
3	MANAGER	20	2.975
4	MANAGER	30	2.850
5	MANAGER	10	2.450

# GROUP BY. Ejemplos

---

En los campos de agrupamiento podemos poner campos de la base de datos y columnas calculadas por funciones de fila.

Por ejemplo si se quiere agrupar los empleados por los miles de euros que ganan podríamos hacer la siguiente consulta

```
139
140 SELECT TRUNCATE(sal/1000,0) "Miles", COUNT(*) FROM emp
141 GROUP BY TRUNCATE(sal/1000,0);
142
143
```

Resultados 1 x

SELECT TRUNCATE(sal/1000,0) "Miles", COUNT(\*)

	123 Miles	123 COUNT(*)
1	0	2
2	1	6
3	2	3
4	3	2
5	5	1

# GROUP BY. Ejemplos

---

Calcular los empleados que son el primero y el último en el índice alfabético de cada departamento:

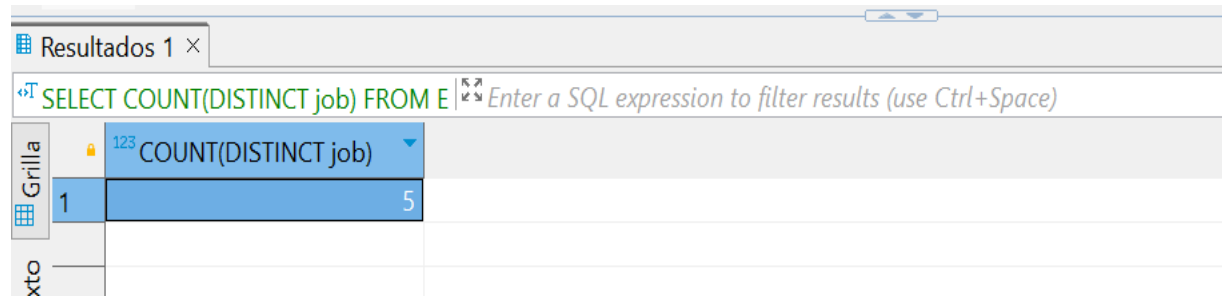
```
select deptno, min(ename) primero, max(ename) ultimo  
from emp  
group by deptno
```

	deptno	primero	ultimo	
1	10	CLARK	MILLER	
2	20	ADAMS	SMITH	
3	30	ALLEN	WARD	

# GROUP BY. Ejemplos

---

```
SELECT COUNT(DISTINCT job) FROM EMP;
```



Resultados 1 x

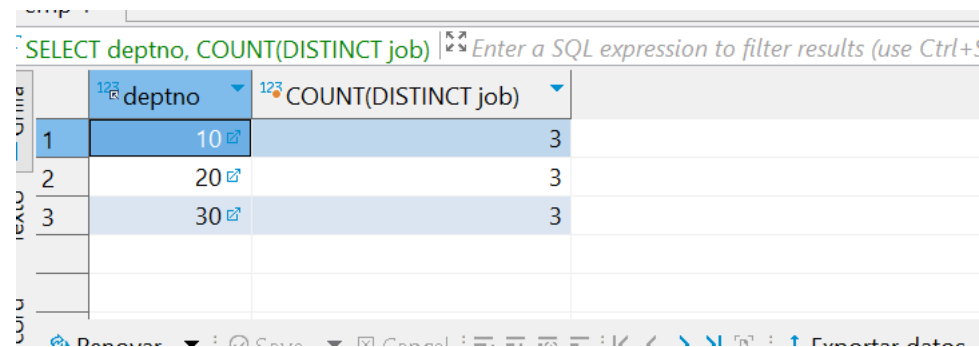
SELECT COUNT(DISTINCT job) FROM EMP

Grilla

	123 COUNT(DISTINCT job)
1	5

xtor

```
SELECT deptno, COUNT(DISTINCT job) FROM EMP GROUP BY deptno;
```



EMP

SELECT deptno, COUNT(DISTINCT job)

Grilla

	123 deptno	123 COUNT(DISTINCT job)
1	10	3
2	20	3
3	30	3

xtor

Revisar Guardar Cancelar Exportar datos

# GROUP BY. Filtrado de filas previo con WHERE

Igual que para el resto de consultas vistas previamente, en una consulta de agrupación o resumen también se pueden realizar filtrado de filas con WHERE

```
SELECT job, COUNT(*) CANTIDAD, MIN(sal) as 'Minimo Salario', MAX(sal) as 'Maximo Salario' FROM emp WHERE hiredate BETWEEN '1981-01-01' AND '1981-12-31' GROUP BY job;
```

emp 1 ×					
SELECT job, COUNT(*) CANTIDAD, MIN   Enter a SQL expression to filter results (use Ctrl+Space)					
Grilla	job	CANTIDAD	Minimo Salario	Maximo Salario	
1	SALESMAN	4	1.250	1.600	
2	MANAGER	3	2.450	2.975	
3	PRESIDENT	1	5.000	5.000	
4	CLERK	1	950	950	
5	ANALYST	1	3.000	3.000	

# Filtrado de agrupación. HAVING

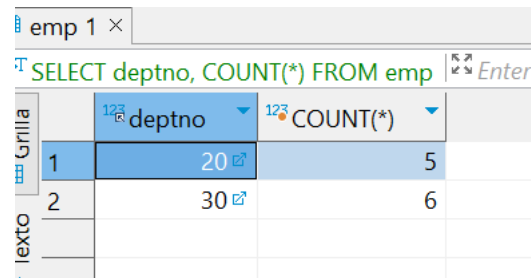
---

La cláusula HAVING se emplea para controlar cuál de los conjuntos de filas se visualiza con lo cual es necesario realizar primero el agrupamiento mediante GROUP BY.

La cláusula HAVING se usa con funciones de grupo, no con campos de agrupación.

Ejemplo: ¿Qué departamento tiene más de tres empleados?

```
SELECT deptno, COUNT(*) FROM emp GROUP BY deptno HAVING COUNT(*) > 3;
```



	deptno	COUNT(*)
1	20	5
2	30	6

# Filtrado de agrupación. HAVING

---

Es importante entender que la clausula HAVING se usa con funciones de grupo, no con campos de agrupación.

Si se desea aplicar una condición a un campo de agrupamiento se usa para ello la clausula WHERE.

Ejemplo: Obtener el salario máximo de los departamentos número 10 y 30.

```
SELECT deptno, MAX(sal) MAXSAL FROM emp WHERE deptno IN (10,30) GROUP BY deptno;
```

```
OJO, si hacemos: SELECT deptno, MAX(sal) MAXSAL FROM emp GROUP BY deptno IN (10,30); DA ERROR
```

# Filtrado de agrupación. HAVING

---

Una consulta con agrupación admite dos tipos de restricciones:

- Las restricciones de fila van en la cláusula del WHERE y se aplican a cada fila (consulta no agregada)
- Las restricciones de grupo van en la cláusula de HAVING y se aplican al grupo: HAVING va después del GROUP BY

Ejemplo: Calcular el descuento medio aplicado en los tickets cuyo código es mayor que 15 para cada cliente pero sólo si dicho descuento supera el 4%

```
SELECT cliente, AVG(dto) FROM ticket WHERE codigo>25 GROUP BY cliente  
HAVING AVG(dto)>4;
```



# Filtrado de agrupación. HAVING

---

WHERE -> RESTRICCIÓN DE FILA

HAVING -> RESTRICCIÓN DE COLUMNA

- Ejemplo. Filtrar tickets de una fecha: Restricción de fila **WHERE**
- Ejemplo. Filtrar clientes más de 15 tickets: Restricción de grupo **HAVING**
- Ejemplo. Filtrar clientes con el IVA máximo del 21%. Restricción de grupo **HAVING**
- Ejemplo. Filtrar tickets con IVA igual a 18%: Restricción de fila **WHERE**

# Filtrado de agrupación. HAVING

---

- En la condición de selección de HAVING sólo pueden (deben) aparecer:
  - Valores constantes
  - Funciones de columna
  - Columnas de agrupación: Columnas que aparecen en la cláusula del GROUP BY
  - Cualquier expresión basada en las anteriores

Lo que hace la cláusula HAVING es la selección de una serie de filas de la tabla resultante de una consulta resumen.

# Ejemplo. HAVING

Mostrar un listado en el número de empleados y el total de los salarios agrupados por departamentos y oficios. Sólo se deben de tener en cuenta los empleados que sean 'CLERK' o 'ANALYST' y las agrupaciones que tengan menos de dos empleados. El listado debe de estar ordenado por número de departamento en orden descendente

```
SELECT deptno, job, COUNT(*) as NUMEMP, SUM(sal) as SUMSAL
FROM emp
WHERE job IN ('CLERK', 'ANALYST')
GROUP BY deptno, job
HAVING COUNT(*) < 2
ORDER BY deptno DESC;
```

T SELECT deptno, job, COUNT(*) as NUM   Enter a SQL expression to filter results					
	deptno	job	NUMEMP	SUMSAL	
1	30	CLERK	1	950	
2	10	CLERK	1	1.300	

# GROUP BY. Extensiones: ROLLUP

ROLLUP genera subresúmenes, información adicional a una consulta resumen. Por ejemplo, en el siguiente caso, genera una fila más por cada agrupación, y una fila más del total, mostrando así el total de salario por departamento (deptno - fila NULL) y el total de salarios

```
SELECT deptno, job, SUM(SAL) as SUMSAL  
FROM emp  
GROUP BY deptno, job WITH ROLLUP
```

SELECT deptno, job, SUM(SAL) as SUM | Enter a SQL expres:

	deptno	job	SUMSAL
1	10	CLERK	1.300
2	10	MANAGER	2.450
3	10	PRESIDENT	5.000
4	10	[NULL]	8.750
5	20	ANALYST	6.000
6	20	CLERK	1.900
7	20	MANAGER	2.975
8	20	[NULL]	10.875
9	30	CLERK	950
10	30	MANAGER	2.850
11	30	SALESMAN	5.600
12	30	[NULL]	9.400
13	[NULL]	[NULL]	29.025

# GROUP BY. Extensiones: ROLLUP.

## Ejemplo

---

Se desea realizar una estadística de las fechas de contratación de los empleados. Para ello, se desea que se desglose que número de empleados ha empezado a trabajar en qué mes y en qué año.

```
SELECT EXTRACT(YEAR FROM hiredate) as 'Year',  
EXTRACT(MONTH FROM hiredate) as 'Month', COUNT(*) as 'Empleados'  
FROM EMP  
GROUP BY EXTRACT(YEAR FROM hiredate), EXTRACT(MONTH FROM hiredate) WITH  
rollup  
order by EXTRACT(YEAR FROM hiredate);
```