* Índice:

**No se encontraron elementos de tabla de contenido.**



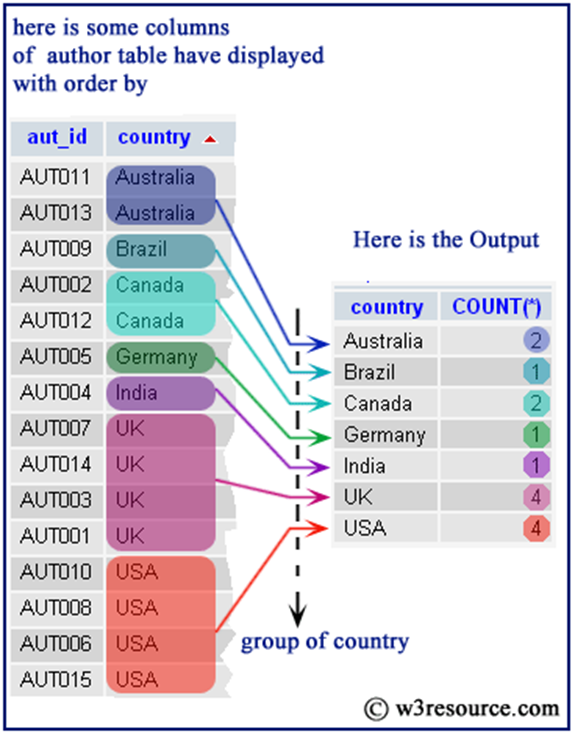
# GROUP BY

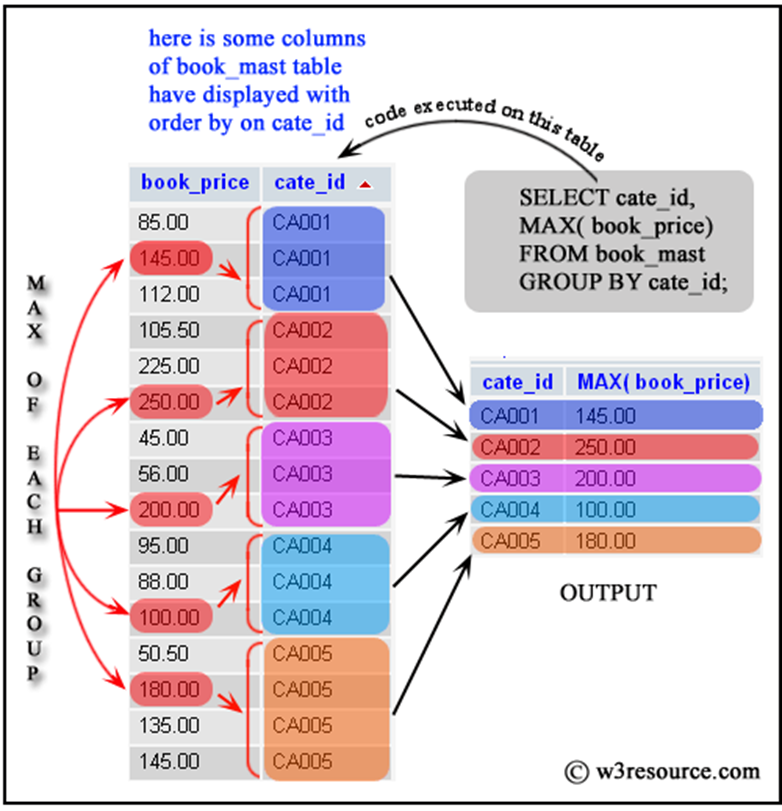
## Teoría

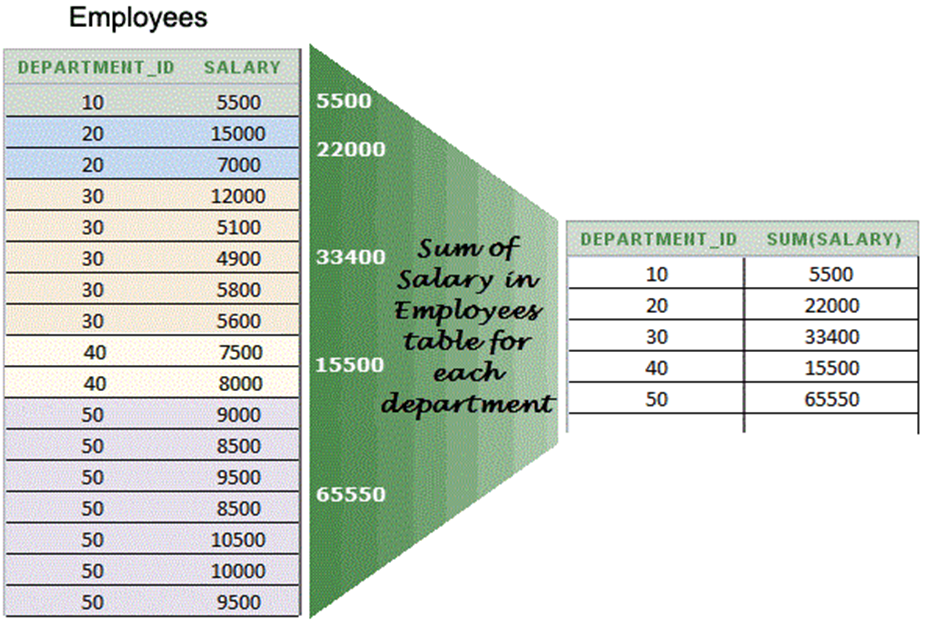
### GROUP BY

Crea una subtabla para cada valor del atributo (o atributos) indicados en GROUP BY.

1. Además crea una tabla para el valor NULL, si es que en algún registro el atributo toma el valor NULL







SELECT department\_id, SUM (salary) FROM employees GROUP BY department\_id

## Ejercicios

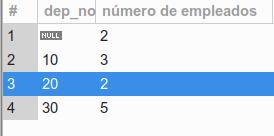
1. Indicar el número de empleados de cada departamento, el título de las columnas será:

* Departamento
* Número de empleados

SELECT dep\_no, COUNT(\*) AS 'número de empl'

FROM empleados

GROUP BY dep\_no;



DEVUELVE 4 FILAS

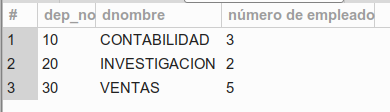
--------------------------------

SELECT dnombre AS 'Departamento', COUNT(\*) AS 'Número de empleados'

FROM empleados NATURAL JOIN departamentos

GROUP BY dep\_no

;



DEVUELVE 3 FILAS, ¿por qué?

--------------------------------

# Esta variante todavía no la conoces, pero se incluye aquí para que entiendas

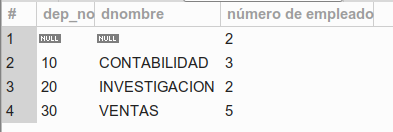
# la divergencia de resultados de las dos soluciones anteriores.

SELECT empleados.dep\_no, dnombre, COUNT(\*) AS 'número de empleados'

FROM empleados LEFT JOIN departamentos ON

empleados.dep\_no = departamentos.dep\_no

GROUP BY dep\_no;



1. Indicar el número de empleados que ganan más de 160.000 en cada departamento. El título de las columnas será:

* Departamento
* Número de empleados

SELECT dnombre AS 'Departamento', COUNT(\*) AS 'Número de empleados'

FROM empleados NATURAL JOIN departamentos

WHERE salario > 160000

GROUP BY dep\_no

;

#esta solución cosidera la suma del salario y la comisión:

SELECT dnombre AS 'Departamento', COUNT(\*) AS 'Número de empleados'

FROM empleados

INNER JOIN departamentos ON empleados.dep\_no = departamentos.dep\_no

WHERE IFNULL(salario,0) + IFNULL(comision,0) > 160000

GROUP BY dnombre ;

1. Indicar el número de empleados que ganan más de 160.000 en cada departamento; mostrar solamente aquellos departamentos que tienen, al menos, 2 empleados que ganen más de 160.000. El título de las columnas será:

* Departamento
* Número de empleados

SELECT dnombre AS 'Departamento', COUNT(\*) AS 'Número de empleados'

FROM empleados NATURAL JOIN departamentos

WHERE salario > 160000

GROUP BY dep\_no

HAVING COUNT(\*) >= 2

;

1. Indicar el número de empleados que ganan más de 160.000 en cada departamento; mostrar solamente aquellos departamentos que tienen, al menos 2 empleados (independientemente de cuánto ganen). El título de las columnas será:

* Departamento
* Número de empleados

Todavía no sabemos hacerlo;

Cuando avance el curso se explicará una nueva herramienta (las subconsultas) que se necesita para resolverla. De todas maneras, al final de este boletín se incluye la solución a este ejercicio por si tienes curiosidad.

1. Indicar el número de empleados para cada oficio existente. Cada columna debe de mostrar una etiqueta significativa.

SELECT oficio, COUNT(\*) AS 'Número de empleados con ese oficio'

FROM empleados

GROUP BY oficio;

1. Para cada departamento indicar Cuánto dinero gasta en salarios, cuánto gasta en comisiones y cuánto gasta en total (salario + comisión). Cada columna debe de mostrar una etiqueta significativa.

SELECT dnombre, SUM(salario) AS 'Gasto en salarios', SUM(comision) AS 'Gasto en comisiones', SUM(IFNULL(salario,0) + IFNULL(comision,0)) AS 'Gasto en salarios + comisiones'

FROM empleados INNER JOIN departamentos ON empleados.dep\_no = departamentos.dep\_no

GROUP BY departamentos.dep\_no

;

SELECT dnombre, SUM(salario) AS 'Gasto en salarios', SUM(comision) AS 'Gasto en comisiones', IFNULL(SUM(salario),0)+ IFNULL(SUM(comision),0)AS 'Gasto en salarios + comisiones'

FROM empleados INNER JOIN departamentos ON empleados.dep\_no = departamentos.dep\_no

GROUP BY departamentos.dep\_no

;

1. Indicar cuántos clientes tenemos en cada localidad. Cada columna debe de mostrar una etiqueta significativa.

SELECT localidad, COUNT(\*) AS 'Cantidad de empleados que trabajan en la localidad'

FROM clientes

GROUP BY localidad

;

1. apellidos y nombre de cada empleado que atiende a algún cliente y cuántos clientes tiene a su cargo ese empleado. Cada columna debe de mostrar una etiqueta significativa.
2. La función CONCAT ( ) concatena texto
3. Ejemplo: SELECT CONCAT( apellido1, nombre)
4. returns NULL if any argument is NULL.

SELECT CONCAT(apellido1, ' ', apellido2, ', ', empleados.nombre) AS 'Empleado', COUNT(\*) AS 'Número de clientes que atiende'

FROM empleados INNER JOIN clientes ON empleados.emp\_no = clientes.vendedor\_no

GROUP BY emp\_no;

1. Para los productos que se han vendido alguna vez: Indicar el nombre de cada producto y cuántas unidades se han vendido de ese producto. Cada columna debe de mostrar una etiqueta significativa.

SELECT descripcion, SUM(unidades) AS 'Unidades vendidas'

FROM pedidos INNER JOIN productos ON pedidos.producto\_no = productos.producto\_no

GROUP BY productos.producto\_no

;

1. Indicar cuántos empleados tiene a su cargo cada empleado que sea jefe de otros. Del jefe indicar nombre y apellidos. Cada columna debe de mostrar una etiqueta significativa.

SELECT jefe.emp\_no AS 'Código del jefe', CONCAT(jefe.apellido1, ' ', jefe.apellido2, ', ', jefe.nombre) AS 'Nombre del jefe', COUNT(\*) AS 'Número de empleados a su cargo'

FROM empleados subordinado INNER JOIN empleados jefe ON subordinado.jefe = jefe.emp\_no

GROUP BY jefe.emp\_no, jefe.apellido1, jefe.apellido2, jefe.nombre

;

SELECT jefe.emp\_no AS 'Código del jefe', CONCAT(jefe.apellido1, ' ', jefe.apellido2, ', ', jefe.nombre) AS 'Nombre del jefe', COUNT(\*) AS 'Número de empleados a su cargo'

FROM empleados subordinado INNER JOIN empleados jefe ON subordinado.jefe = jefe.emp\_no

GROUP BY jefe.emp\_no

;

1. Indicar el nombre y la fecha en que se ha vendido por última vez cada producto.

SELECT productos.producto\_no, descripcion, MAX(fecha\_pedido) AS 'Fecha del último pedido'

FROM pedidos INNER JOIN productos ON pedidos.producto\_no = productos.producto\_no

GROUP BY productos.producto\_no

;

# HAVING

1. Indicar el número de empleados que tiene cada oficio, pero sólo mostrar aquellos oficios que tengan 2 o más empleados, ordenar por número de empleados en orden decreciente.

SELECT oficio, COUNT(\*) AS 'Número de empleados con ese oficio'

FROM empleados

GROUP BY oficio

HAVING COUNT(\*) > 2

ORDER BY 2 DESC

;

1. Nombre y apellidos de cada empleado que atiende a algún cliente y Cuántos clientes tiene a su cargo ese empleado. Mostrar sólo aquellos empleados que atiendan a más de 2 clientes.

SELECT CONCAT(empleados.nombre, ' ', apellido1,' ', apellido2) AS 'Nombre del empleado', COUNT(\*) AS 'Número de clientes'

FROM empleados INNER JOIN clientes ON empleados.emp\_no = clientes.vendedor\_no

GROUP BY emp\_no

HAVING COUNT(\*)>2;

1. Para los productos que se han vendido alguna vez: Indicar el nombre del producto y cuántas unidades se han vendido en total de ese producto, pero sólo mostrar aquellos productos que se han vendido en más de 2 pedidos.

SELECT productos.producto\_no, descripcion, SUM(unidades)

FROM pedidos INNER JOIN productos ON pedidos.producto\_no = productos.producto\_no

GROUP BY productos.producto\_no, descripcion

HAVING COUNT(\*)>2;

1. Para los productos que se han vendido alguna vez: Indicar el nombre de cada producto y cuántas unidades se han vendido en total de ese producto, pero sólo mostrar aquellos de los que se han vendido más de 15 unidades.

SELECT productos.producto\_no, descripcion, SUM(unidades)

FROM pedidos INNER JOIN productos ON pedidos.producto\_no = productos.producto\_no

GROUP BY productos.producto\_no, descripcion

HAVING SUM(unidades)>15;

1. Indicar cuántos clientes tenemos en cada localidad que tenga más de 1 cliente, ordenar por número de clientes en orden decreciente.

SELECT localidad, COUNT(\*)

FROM clientes

GROUP BY localidad

HAVING COUNT(\*)>1

ORDER BY 2 DESC;

1. Indicar cuántos empleados tiene a su cargo cada empleado que sea jefe de otros. Sólo mostrar los que son jefes de 2 personas exactamente. Se debe de mostrar el nombre y apellidos de esos jefes.

SELECT jefe.emp\_no, jefe.apellido1, jefe.apellido2, jefe.nombre, COUNT(\*)

FROM empleados subordinado INNER JOIN empleados jefe ON subordinado.jefe = jefe.emp\_no

GROUP BY jefe.emp\_no

HAVING COUNT(\*)=2;

1. Indicar el nombre del producto, la fecha en que se ha vendido por última vez ese producto y el importe total de ventas de ese producto, mostrar sólo aquellos productos cuyo importe total de ventas sea superior a 500.000

SELECT productos.producto\_no, descripcion, MAX(fecha\_pedido), SUM(precio\_actual \* unidades)

FROM pedidos INNER JOIN productos ON pedidos.producto\_no = productos.producto\_no

GROUP BY productos.producto\_no, descripcion

HAVING SUM(precio\_actual \* unidades)>500000

;

1. Solución al ejercicio que se propuso anteriormente y del que se dijo que todavía no sabíamos resolverlo:
2. Indicar el número de empleados que ganan más de 160.000 en cada departamento; mostrar solamente aquellos departamentos que tienen, al menos 2 empleados (independientemente de cuánto ganen). El título de las columnas será:

* Departamento
* Número de empleados

**SELECT dnombre AS 'Departamento', COUNT(\*) AS 'Número de empleados'**

**FROM empleados NATURAL JOIN departamentos**

**WHERE salario > 160000 AND DEP\_NO IN (**

**SELECT dep\_no FROM empleados**

**GROUP BY dep\_no HAVING COUNT(\*)>=2)**

**GROUP BY dep\_no**

**;**