

Agrupamentos

Índice

1.	Consultas resumo con agrupamentos	3
1.1	Consultas con agrupamento de filas	3
1.2	Cláusula GROUP BY	3
1.2.1	Agrupamento por máis dunha columna	6
1.2.2	Modificador WITH ROLLUP	7
1.3	Cláusula HAVING	9

1. Consultas resumo con agrupamentos

1.1 Consultas con agrupamento de filas

Chámase grupo á información correspondente a un conxunto de filas que teñen o mesmo valor nunha ou varias columnas dunha táboa denominadas columnas de agrupamento.

Existe un tipo de consultas con agrupamento de filas que dá como resultado unha táboa resumo que ten como columnas, as columnas de agrupamento e os cálculos, e ten unha fila por cada valor distinto que toman as columnas de agrupamento. Os cálculos realízanse coas funcións de agrupamento ou agregado que actúan sobre as filas de cada grupo en lugar de actuar sobre todas as filas seleccionadas como se fixo ata este momento.

A cláusula GROUP BY utilízase para facer consultas con agrupamento de filas.

1.2 Cláusula GROUP BY

Permite especificar as columnas de agrupamento. Sintaxe:

```
[GROUP BY {nome_columna | expresión | posición} [ASC | DESC], ... [WITH ROLLUP]]
```

- As columnas de agrupamento pódense representar por unha lista dun ou máis nomes de columnas, expresións, ou ben o número de orde que ocupa a columna dentro da lista de selección, igual que na cláusula ORDER BY. Non é recomendable utilizar o número de orde da columna porque no caso de cambiar a orde das columnas na lista de selección, ou engadir ou eliminar algunha columna, pódense producir resultados inesperados se non se cambia a cláusula GROUP BY.
- O modificador WITH ROLLUP, permite engadir filas na táboa resumo. Estas filas corresponden a cálculos feitos creando un grupo novo que contén todas as filas, e creando grupos a varios niveis no caso de agrupar por máis dunha columna.

En MySQL, cando se utiliza esta cláusula, ordénanse as filas polas columnas de agrupamento como se fose unha cláusula ORDER BY, antes de formar os grupos. As opcións [ASC | DESC] son unha extensión de MySQL ao estándar, e indican a orde na que se ordenan as filas antes de formar os grupos (ASC = ascendente, DESC = descendente); se non se indica nada, tómase ASC como valor por defecto.

Exemplo: calcular o número de departamentos que hai en cada provincia tendo en conta os datos contidos na táboa *departamento* da base de datos *practic5*.

```
select id_provincia, count(*)  
from departamento  
group by id_provincia;
```

O SXBDR selecciona todas as filas da táboa *departamento* e as ordena pola columna *id_provincia*, que é a columna de agrupamento, para que estean xuntos todos os departamentos que pertencen á mesma provincia; despois vai collendo fila a fila e cada vez que cambia o valor da columna *id_provincia* prodúcese un ruptura e fórmase un novo grupo. Esta explicación do funcionamento ten unha finalidade didáctica, pero a forma en que se executa internamente a sentenza a decide o optimizador de consultas, e pode variar dun xestor a outro.

Representación gráfica do resultado da execución:

departamento			
codigo	nome	cidade	id_provincia
9	Oficina8	Coruña	15
3	Oficina2	Ferrol	15
1	Central	Lugo	27
8	Oficina7	Lugo	27
2	Oficina1	Monforte	27
6	Oficina5	Villalba	27
10	Oficina9	Villalba	28
7	Oficina6	Ourense	32
5	Oficina4	Ourense	32
4	Oficina3	Vigo	36

```
select id_provincia, count(*)  
from departamento  
group by id_provincia;
```

Táboa resumo	
id_provincia	count(*)
15	2
27	4
28	1
32	2
36	1

Na parte esquerda da imaxe móstranse as filas e columnas que forman a táboa *departamento*, remarcando as filas que teñen o mesmo valor na columna *id_provincia*. Cada un dos marcos representa un grupo formado por todas as filas que teñen o mesmo valor na columna de agrupamento.

Na parte dereita da imaxe, móstrase a táboa resumo resultante que ten unha fila por cada grupo. As columnas que ten a táboa resumo son, en primeiro lugar, a columna de agrupamento (*id_provincia*) para mostrar o valor que identifica ás filas de cada grupo, e en segundo lugar, os cálculos que se queren facer coas filas do grupo, utilizando neste caso, a función de agregado `COUNT(*)` que conta o número de filas do grupo. Pódese observar que non ten sentido poñer calquera outra columna da táboa *departamento* na táboa resumo porque pode que teña valores distintos en distintas filas do grupo e o valor que se mostraría na columna sería inconsistente.

Algunhas consideracións a ter en conta cando se utiliza a cláusula `GROUP BY`:

- Para indicar as columnas de agrupamento, pódense utilizar os alias de columna.
- Pode haber diferenzas no tratamento dos valores `NULL` á hora de formar grupos. Algúns SGBDR consideran que non se agrupan as filas que conteñan o valor `NULL` na columna de agrupamento e outros consideran que todas as filas que teñan o valor `NULL` na columna de agrupamento forman un grupo. MySQL utiliza a segunda opción.
- O estándar SQL establece que na lista de selección da cláusula `SELECT` só se pode facer referencia ás columnas polas que se agrupa, funcións de agregado, constantes, ou expresións que combinen ás anteriores. Na maioría dos SGBDR, no caso de incluír calquera outra columna daría lugar a un erro. MySQL estende o uso de `GROUP BY` permitindo facer referencia na lista de selección de columnas distintas das columnas de agrupamento, e deixa en mans do usuario controlar que as columnas tomen un valor único para cada grupo. Pódese cambiar este comportamento, e desactivar as extensións de MySQL para `GROUP BY`, habilitando o modo `SQL 'ONLY_FULL_GROUP_BY'`.
- Para cada grupo (cada valor distinto que tomen as columnas de agrupamento) créase unha fila na táboa resumo. Todos os elementos da lista de selección deben ter un valor único para cada grupo.

Os seguintes exemplos mostran dúas situacións diferentes producidas ao incluír na selección columnas distintas das columnas de agrupamento.

- Exemplo incluíndo a columna *cidade*.

```
select id_provincia, cidade , count(*)  
from departamento  
group by id_provincia;
```

Representación gráfica do resultado da execución:

departamento			
codigo	nome	cidade	id_provincia
9	Oficina8	Coruña	15
3	Oficina2	Ferrol	15
1	Central	Lugo	27
8	Oficina7	Lugo	27
2	Oficina1	Monforte	27
6	Oficina5	Villalba	27
10	Oficina9	Villalba	28
7	Oficina6	Ourense	32
5	Oficina4	Ourense	32
4	Oficina3	Vigo	36

```
select id_provincia, cidade, count(*)
from departamento
group by id_provincia;
```

Táboa resumo		
id_provincia	cidade	count(*)
15	Ferrol	2
27	Lugo	4
28	Villalba	1
32	Ourense	2
36	Vigo	1

O contido da columna *cidade* na táboa *resumo* é inconsistente. A columna pode tomar máis dun valor para cada grupo. Na táboa resumo móstrase na columna un valor correspondente a unha fila do grupo, pero non representa unha información resumo do grupo para os casos de grupos que teñen máis dunha fila. Por exemplo, a primeira fila da táboa resumo parece que nos indica que na cidade de Ferrol hai dous departamentos, cando na cidade só hai un departamento, o 2 corresponde ao número de departamentos que hai na provincia que ten o *id_provincia* (columna de agrupamento) 15.

MySQL non mostra ningunha mensaxe de erro no momento de executar a consulta pero a sentenza non é correcta e os datos son inconsistentes. A maioría dos SXBDR, ao executar esta sentenza mostrarían un erro indicando que na lista de selección hai unha columna que non está en GROUP BY.

- Exemplo incluíndo a columna *provincia* que contén o nome da provincia.

```
select id_provincia, provincia, count(*)
from departamento natural join provincia
group by id_provincia;
```

Representación gráfica do resultado da execución:

departamento			
codigo	nome	cidade	id_provincia
9	Oficina8	Coruña	15
3	Oficina2	Ferrol	15
1	Central	Lugo	27
8	Oficina7	Lugo	27
2	Oficina1	Monforte	27
6	Oficina5	Villalba	27
10	Oficina9	Villalba	28
7	Oficina6	Ourense	32
5	Oficina4	Ourense	32
4	Oficina3	Vigo	36

```
select id_provincia, provincia, count(*)
from departamento natural join provincia
group by id_provincia;
```

Táboa resumo		
id_provincia	provincia	count(*)
15	Coruña, A	2
27	Lugo	4
28	Madrid	1
32	Ourense	2
36	Pontevedra	1

A columna *provincia* da táboa resumo toma valores consistentes, porque para cada valor de *id_provincia* (columna de agrupamento) só hai un valor posible para a columna *provincia*. Neste caso para cada grupo a columna *provincia* só toma un valor independentemente do número de filas que teña o grupo, polo que a sentenza é correcta.

1.2.1 Agrupamento por máis dunha columna

É posible agrupar por máis dunha columna, escribindo a lista das columnas polas que se quere agrupar, separadas por comas na cláusula GROUP BY.

Exemplo: contar o número de departamentos que hai en cada cidade.

- Solución 1 agrupando só pola columna *cidade*:

```
select id_provincia, cidade, count(*)  
from departamento  
group by cidade;
```

Representación gráfica do resultado da execución:

codigo	nome	cidade	id_provincia
9	Oficina8	Coruña	15
3	Oficina2	Ferrol	15
1	Central	Lugo	27
8	Oficina7	Lugo	27
2	Oficina1	Monforte	27
6	Oficina5	Villalba	27
10	Oficina9	Villalba	28
7	Oficina6	Ourense	32
5	Oficina4	Ourense	32
4	Oficina3	Vigo	36

```
select id_provincia, cidade, count(*)  
from departamento  
group by cidade;
```

id_provincia	cidade	count(*)
15	Coruña	1
15	Ferrol	1
27	Lugo	2
27	Monforte	1
32	Ourense	2
36	Vigo	1
27	Villalba	2

Créase unha única fila para a cidade 'Villalba' na táboa resumo, sen ter en conta que o mesmo nome de cidade se repite en dúas provincias distintas (27 e 28). Esta solución non sería correcta porque mostra que na cidade 'Villalba' da provincia 27 hai dous departamentos cando só hai un; non ten en conta que se trata de cidades diferentes.

- Solución 2 agrupando polas columnas *id_provincia* e *cidade*.

```
select id_provincia, cidade, count(*)  
from departamento  
group by id_provincia, cidade;
```

Representación gráfica do resultado da execución:

codigo	nome	cidade	id_provincia
9	Oficina8	Coruña	15
3	Oficina2	Ferrol	15
1	Central	Lugo	27
8	Oficina7	Lugo	27
2	Oficina1	Monforte	27
6	Oficina5	Villalba	27
10	Oficina9	Villalba	28
7	Oficina6	Ourense	32
5	Oficina4	Ourense	32
4	Oficina3	Vigo	36

```
select id_provincia, cidade, count(*)  
from departamento  
group by id_provincia, cidade;
```

id_provincia	cidade	count(*)
15	Coruña	1
15	Ferrol	1
27	Lugo	2
27	Monforte	1
27	Villalba	1
28	Villalba	1
32	Ourense	2
36	Vigo	1

Esta solución sería a correcta porque fai primeiro o agrupamento por provincias, e dentro de cada provincia, o agrupamento por cidade. Neste caso, a cidade 'Villalba' móstrase dúas veces, unha asociada ao id_provincia 27 e outra ao id_provincia 28.

1.2.2 Modificador WITH ROLLUP

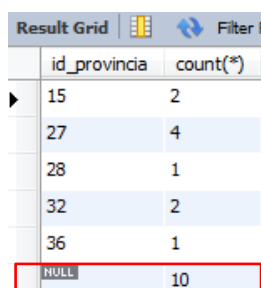
Permite mostrar novas filas na táboa de resumo. Estas filas representan operacións de resumo para distintos niveis de análises. Resumo do comportamento do modificador nun servidor MySQL:

- No caso de agrupar só por unha columna, o modificador engade unha nova fila na táboa resumo na que se mostran os cálculos para un novo grupo formado por todas as filas seleccionadas. Nesta fila móstrase o valor NULL na columna de agrupamento.

Exemplo: contar o número de departamentos que hai en cada provincia, e o número total de departamentos da empresa, tendo en conta os datos contidos na táboa *departamento* da base de datos *practicass5*.

```
/* agrupar por unha columna co modificador WITH ROLLUP*/
```

```
select id_provincia, count(*)  
from departamento  
group by id_provincia with rollup;
```



id_provincia	count(*)
15	2
27	4
28	1
32	2
36	1
NULL	10

Número total de departamentos

No resultado da execución pódese ver que ademais de mostrar unha fila por cada grupo, se engade unha fila máis co cálculo feito considerando un novo grupo formado por todas as filas da táboa *departamento*.

- No caso de agrupar por máis dunha columna, o modificador engade unha nova fila na táboa resumo cada vez que hai un cambio de valor nalgunha das columnas de agrupamento, ademais da fila correspondente ao cálculo para todas as filas.

Exemplo: contar o número de departamentos que hai en cada cidade, os que hai por cada provincia e o número total de departamentos da empresa.

```
/* agrupar por máis dunha columna co modificador WITH ROLLUP*/
```

```
select id_provincia, cidade, count(*)  
from departamento  
group by id_provincia, cidade with rollup;
```

Resultado a execución:

id_provincia	cidade	count(*)	
15	Coruña	1	
15	Ferrol	1	
15	NULL	2	Número de departamentos da provincia 15
27	Lugo	2	
27	Monforte	1	
27	Villalba	1	
27	NULL	4	Número de departamentos da provincia 27
28	Villalba	1	
28	NULL	1	Número de departamentos da provincia 28
32	Ourense	2	
32	NULL	2	Número de departamentos da provincia 32
36	Vigo	1	
36	NULL	1	Número de departamentos da provincia 36
NULL	NULL	10	Número total de departamentos

No resultado da execución pódese ver que ademais de mostrar unha fila por cada grupo, se engade unha fila máis por cada provincia, que é a primeira columna da lista de columnas de agrupamento, e ao final engade unha fila máis co cálculo feito considerando un novo grupo formado por todas as filas da táboa *departamento*.

Algunhas consideracións sobre o uso de ROLLUP en MySQL:

- Cando se usa este modificador non se pode utilizar a cláusula ORDER BY.
- Utilizar a cláusula LIMIT co modificador pode producir resultados difíciles de interpretar, porque a cláusula aplícase despois do modificador así que no límite cóntanse as filas extra engadidas polo modificador.

O resultado obtido cando se agrupa por unha soa columna e se utiliza ROLLUP é o mesmo que utilizando unha UNION de consultas.

```
/* Agrupar por unha columna co modificador WITH ROLLUP*/
select id_provincia, count(*)
from departamento
group by id_provincia with rollup;
/* Alternativa utilizando union*/
select id_provincia, count(*)
from departamento
group by id_provincia
union
select 'Total ...', count(*)
from departamento;
```

Ambas opcións obteñen o mesmo resultado:

Result Grid			Filter
	id_provincia	count(*)	
▶	15	2	
	27	4	
	28	1	
	32	2	
	36	1	
	Total ...	10	

1.3 Cláusula HAVING

Esta cláusula está asociada coa cláusula GROUP BY. Permite establecer condicións para descartar aqueles grupos que non cumpran esas condicións. É similar á cláusula WHERE, pero a diferenza entre elas é que mentres WHERE analiza a condición nas filas das táboas de orixe da consulta (as especificadas na cláusula FROM), a cláusula HAVING o fai sobre a táboa resumo, é dicir, sobre os grupos que se formaron despois de agrupar.

Exemplo: contar os departamentos que hai en cada cidade, e mostrar só as cidades que teñen máis dun departamento.

```
select id_provincia, cidade, count(*)
from departamento
group by id_provincia, cidade
having count(*) > 1;
```

Representación gráfica do resultado da execución:

id_provincia	cidade	count(*)
15	Coruña	1
15	Ferrol	1
27	Lugo	2
27	Monforte	1
27	Villalba	1
28	Villalba	1
32	Ourense	2
36	Vigo	1

```
select id_provincia, cidade, count(*)
from departamento
group by id_provincia, cidade
having count(*) > 1;
```

id_provincia	cidade	count(*)
27	Lugo	2
32	Ourense	2

Ao aplicar a condición establecida na cláusula HAVING, elimináronse da táboa resumo os grupos que non cumpren a condición.

Algunhas consideracións a ter en conta cando se utiliza a cláusula HAVING:

- O estándar ANSI SQL establece que as condicións desta cláusula só poden referirse ás columnas de agrupamento (as que figuran en GROUP BY), ou ás funcións de agregado. A partir da versión 5.0.2 MySQL estende este comportamento, e admite utilizar nas condicións do HAVING outras columnas diferentes das relacionadas en GROUP BY, e subconsultas. Pódese cambiar este comportamento, e desactivar estas extensións de MySQL para GROUP BY, habilitando o modo SQL 'ONLY_FULL_GROUP_BY'.
- Non se debe utilizar HAVING para condicións que deban estar na cláusula WHERE.
- Pódense utilizar os alias para escribir as condicións.