4. Consultas con agrupamientos

4.1. Objetivos

Los objetivos de esta actividad son:

- Realizar consultas agrupando filas del conjunto de resultados y hacer cálculos con las funciones de agrupamiento.
- Realizar consultas estableciendo condiciones para los grupos.

4.2. Actividad

4.2.1. Consultas con agrupamiento de filas

Se llama grupo a la información correspondiente a un conjunto de filas que tienen el mismo valor en una o varias columnas de una tabla denominadas columnas de agrupamiento.

Existe un tipo de consultas con agrupamiento de filas que da como resultado una tabla resumen que tiene como columnas a las columnas de agrupamiento y tiene una fila por cada valor distinto que toman estas columnas. Se pueden realizar cálculos con las funciones de agrupamiento sobre las filas de cada grupo, en lugar de actuar sobre todas las filas seleccionadas, como se ha hecho hasta este momento.

La cláusula GROUP BY se utiliza para hacer consultas con agrupamiento de filas.

4.2.2. Cláusula GROUP BY

Permite especificar las columnas de agrupamiento. Sintaxis:

```
[GROUP BY {nombre_columna | expresión | posición} [ASC| DESC], ...]
```

Las columnas de agrupamiento se pueden representar por una lista de uno o más nombres de columnas, expresiones, o bien el número de orden que ocupa la columna dentro de la lista de selección, igual que en la cláusula ORDER BY. No es recomendable utilizar el número de orden de la columna porque en el caso de cambiar el orden de las columnas en la lista de selección, o añadir o eliminar alguna columna, se pueden producir resultados inesperados si no se cambia la cláusula GROUP BY.

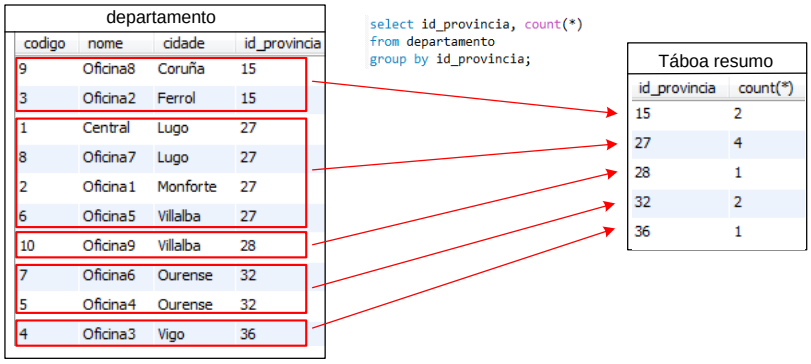
En MySQL, cuando se utiliza esta cláusula, se ordenan las filas por las columnas de agrupamiento como si fuera una cláusula ORDER BY, antes de formar los grupos. Las opciones [ASC | DESC] son una extensión de MySQL al estándar, e indican el orden en la que se ordenan las filas antes de formar los grupos (ASC = ascendente, DESC = descendiente); si no se indica nada, se toma ASC como valor por defecto.

Por ejemplo: calcular el número de departamentos que hay en cada provincia teniendo en cuenta los datos contenidos en la tabla *departamento*.

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

El SGBD selecciona todas las filas de la tabla *departamento* y las ordena por la columna *id_provincia*, que es la columna de agrupamiento, para que estén juntos todos los departamentos que pertenecen a la misma provincia; después va cogiendo fila a fila y cada vez que cambia el valor de la columna *id_provincia* se produce una ruptura y se forma un nuevo grupo.

Una representación gráfica del resultado de la ejecución podría ser:



En la parte izquierda de la imagen se muestran las filas y columnas que forman la tabla *departamento*, remarcando las filas que tienen el mismo valor en la columna *id_provincia*. Cada uno de los marcos representa un grupo formado por todas las filas que tienen el mismo valor en la columna de agrupamiento.

En la parte derecha de la imagen, se muestra la tabla resumen resultante que tiene una fila por cada grupo. Las columnas que tiene la tabla resumen son, en primer lugar, la columna de agrupación (*id_provincia*) para mostrar el valor que identifica a las filas de cada grupo, y en segundo lugar, los cálculos que se quieren hacer con las filas del grupo, utilizando en este caso, la función de agrupamiento COUNT(*), que cuenta el número de filas del grupo. Se puede observar que no tiene sentido poner cualquier otra columna de la tabla *departamento* en la tabla resumen porque puede que tenga valores distintos en distintas filas del grupo y el valor que se mostraría en la columna sería inconsistente.

Algunas consideraciones a tener en cuenta cuando se utiliza la cláusula GROUP BY:

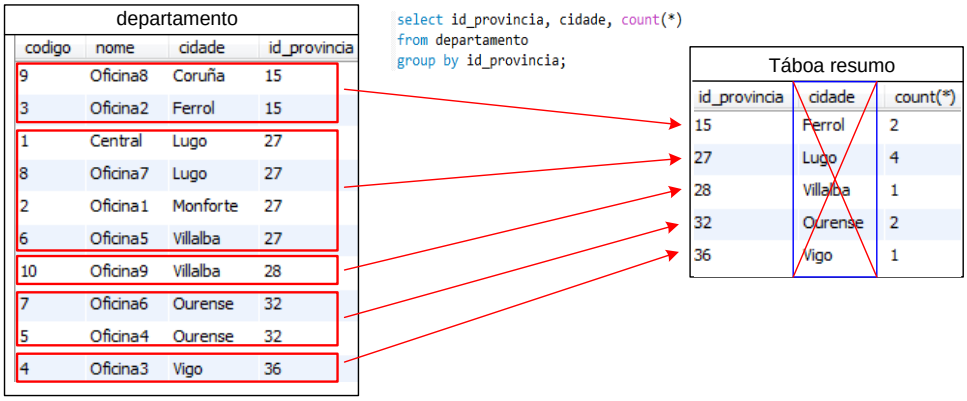
- Para indicar las columnas de agrupación, pueden utilizarse los alias de columna.
- Puede haber diferencias en el tratamiento de los valores NULL a la hora de formar grupos. Algunos SGBD consideran que no se agrupan las filas que contengan el valor NULL en la columna de agrupación y otros consideran que todas las filas que tengan el valor NULL en la columna de agrupación forman un grupo. MySQL utiliza la segunda opción.
- El estándar SQL establece que en la lista de selección de la cláusula SELECT sólo se puede hacer referencia a las columnas por las que se agrupa, funciones de agrupamiento, constantes, o expresiones que combinen a las anteriores. En la mayoría de los SGBD, en el caso de incluir cualquier otra columna daría lugar a un error. MySQL extiende el uso de GROUP BY permitiendo hacer referencia en la lista de selección de columnas distintas de las columnas de agrupamiento, y deja en manos del usuario controlar que las columnas tomen un valor único para cada grupo. Se puede cambiar este comportamiento, y desactivar las extensiones de MySQL para GROUP BY, habilitando el modo SQL 'ONLY_FULL_GROUP_BY'.
- Para cada grupo (cada valor distinto que tomen las columnas de agrupamiento) se crea una fila en la tabla resumen. Todos los elementos de la lista de selección deben tener un valor único para cada grupo.

A continuación se muestra un ejemplo en el que se incluye en la selección columnas distintas de las columnas de agrupamiento.

- Ejemplo incluyendo la columna *ciudad:*

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

Representación gráfica del resultado de la ejecución:



El contenido de la columna *ciudad en la* tabla *resumen* es inconsistente. La columna puede tomar más de un valor para cada grupo. En la tabla resumen se muestra en la columna un valor correspondiente a una fila del grupo, pero no representa una información resumen del grupo para los casos de grupos que tienen más de una fila. Por ejemplo, la primera fila de la tabla resumen parece que nos indica que en la ciudad de Ferrol hay dos departamentos, cuando en la ciudad sólo hay un departamento, el 2 corresponde al número de departamentos que hay en la provincia que tiene el *id_provincia* (columna de agrupamiento) 15.

MySQL no muestra ninguna mensaje de error en el momento de ejecutar la consulta pero la sentencia no es correcta y los datos son inconsistentes. La mayoría de los SGBDR, al ejecutar esta sentencia mostrarían un error indicando que en la lista de selección hay una columna que no está en GROUP BY.

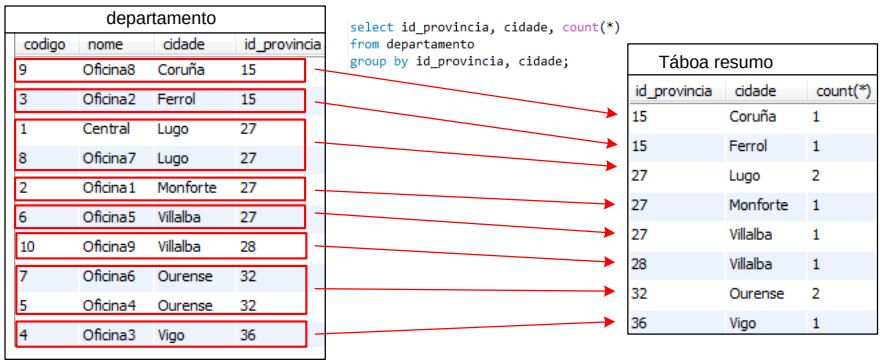
4.2.2.1. Agrupamiento por más de una columna

Es posible agrupar por más de una columna, escribiendo la lista de las columnas por las que se quiere agrupar, separadas por comas en la cláusula GROUP BY.

Ejemplo: contar el número de departamentos que hay en cada ciudad.

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

Representación gráfica del resultado de la ejecución:



Esta solución sería la correcta porque hace primero el agrupamiento por provincias, y dentro de cada provincia, el agrupamiento por ciudad.

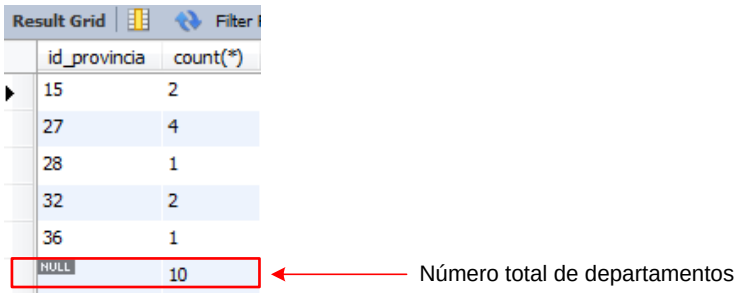
4.2.2.2. Modificador WITH ROLLUP

Permite mostrar nuevas filas en la tabla de resumen. Estas filas representan operaciones de resumen para distintos niveles de análisis.

- En el caso de agrupar sólo por una columna, el modificador añade una nueva fila en la tabla resumen en la que se muestran los cálculos para un nuevo grupo formado por todas filas seleccionadas. En esta fila se muestra el valor NULL en la columna de agrupamiento.

  Ejemplo: mostrar el número de departamentos que hay en cada provincia, y el número total de departamentos de la empresa.

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



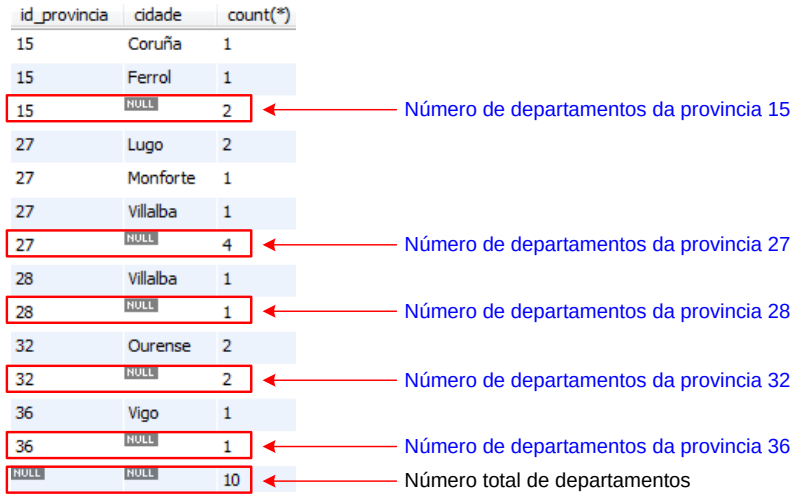
En el resultado de la ejecución se puede ver que además de mostrar una fila por cada grupo, se añade una fila más con el cálculo hecho considerando un nuevo grupo formado por todas las filas de la tabla *departamento*.

- En el caso de agrupar por más de una columna, el modificador añade una nueva fila en la tabla resumen cada vez que hay un cambio de valor en alguna de las columnas de agrupamiento, además de la fila correspondiente al cálculo para todas las filas.

  Ejemplo: contar el número de departamentos que hay en cada ciudad, los que hay por cada provincia y el número total de departamentos de la empresa.

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

Resultado de la ejecución:



En el resultado de la ejecución se puede ver que además de mostrar una fila por cada grupo, se añade una fila más por cada provincia, que es la primera columna de la lista de columnas de agrupación, y al final añade una fila más con el cálculo hecho considerando un nuevo grupo formado por todas las filas de la tabla *departamento*.

Algunas consideraciones sobre el uso de ROLLUP en MySQL:

- Cuando se usa este modificador no se puede utilizar la cláusula ORDER BY.
- Utilizar la cláusula LIMIT con el modificador puede producir resultados difíciles de interpretar, porque la cláusula se aplica después del modificador así que en el límite se cuentan las filas extra añadidas por el modificador.

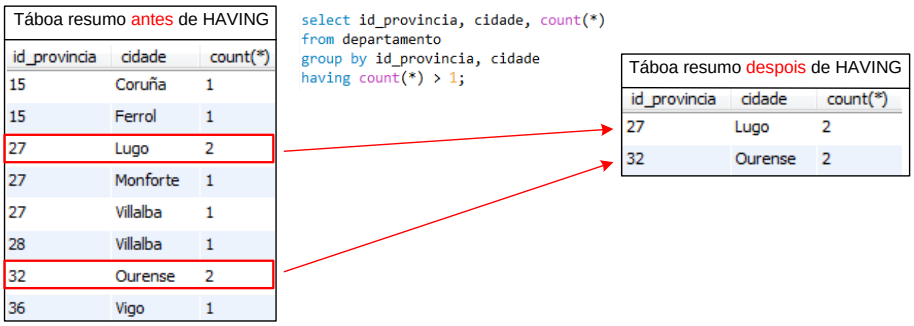
4.2.3. Cláusula HAVING

Esta cláusula está asociada con la cláusula GROUP BY. Permite establecer condiciones para descartar aquellos grupos que no cumplan esas condiciones. Es similar a la cláusula WHERE, pero la diferencia entre ellas es que mientras WHERE analiza la condición en las filas de las tablas de origen de la consulta (las especificadas en la cláusula FROM), la cláusula HAVING lo hace sobre la tabla resumen, es decir, sobre los grupos que se formaron después de agrupar.

Ejemplo: contar los departamentos que hay en cada ciudad, y mostrar solo las ciudades que tienen más de un departamento.

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

Representación gráfica del resultado de la ejecución:



Al aplicar la condición establecida por la cláusula HAVING, se han eliminado de la tabla resumen los grupos que no cumplen la condición.