

➤ Cláusula HAVING

La cláusula HAVING nos permite seleccionar filas de la tabla resultante de una consulta de resumen. Las cláusulas WHERE, GROUP BY, y HAVING

Las cláusulas WHERE y HAVING en un comando SELECT controlan qué filas de las tablas fuentes serán usadas para construir el conjunto de resultados. Las cláusulas WHERE y HAVING son filtros. Especifican una serie de condiciones de búsqueda, y solo se utilizan para construir el conjunto de resultados aquellas filas que satisfacen las condiciones de filtro. Se dice que estas filas califican para participar del conjunto de resultados.

La cláusula HAVING se usa típicamente en conjunción con la cláusula GROUP BY, aún cuando se puede especificar sin la cláusula GROUP BY. La cláusula **HAVING** especifica filtros adicionales y posteriores a los aplicados por la cláusula WHERE.

➤ La cláusula GROUP BY

La cláusula GROUP BY se utiliza para producir valores agregados para cada fila en el conjunto de resultados. Cuando no se usa la cláusula GROUP BY, las funciones de agregación retornan un solo valor para un comando SELECT.

La palabra clave **GROUP BY** se sigue de una lista de columnas, conocidas como columnas de agrupamiento. La cláusula GROUP BY restringe las filas del conjunto de resultados. Habrá una sola fila para cada valor distinto de la o las columnas de agrupamiento. Cada fila del conjunto de resultados contendrá datos resumidos relacionados con el valor específico de la columna de agrupamiento.

➤ Cómo se procesan las cláusulas WHERE, GROUP BY y HAVING

Entender la secuencia correcta en la que las cláusulas WHERE, GROUP BY, y HAVING se aplican, ayuda a formular consultas eficientes:

- * La cláusula WHERE se usa para filtrar las filas que resultan de aplicar las operaciones indicadas en la cláusula FROM.

- * La cláusula GROUP BY se aplica para agrupar el resultado de la cláusula WHERE.

- * Por último, la cláusula HAVING se utiliza para filtrar las filas del resultado agrupado.

Para cualquier condición de búsqueda que se podría aplicar indistintamente antes o después de la operación de agrupamiento, es más eficiente indicarla en la cláusula WHERE. Esto reduce el número de filas a ser agrupadas. Las únicas condiciones de filtro que deben ser indicadas en la cláusula HAVING son aquellas que deben ser aplicadas después que se realicen las operaciones de agrupamiento.

➤ La cláusula ORDER BY

La cláusula ORDER BY ordena el resultado de una consulta por una o más columnas (hasta 8060 bytes). Un ordenamiento puede ser ascendente (ASC) o descendente (DESC). Si no se especifica un ordenamiento, se asume ASC. Si se indican más de una columna en la cláusula ORDER BY el ordenamiento es anidado.

Para la condición de selección se pueden utilizar los mismos tests de comparación descritos en la cláusula WHERE, también se pueden escribir condiciones compuestas (unidas por los operadores OR, AND, NOT), pero existe una restricción.

Ejemplo: Queremos saber las oficinas con un promedio de ventas de sus empleados mayor que 50,000 córdobas.

```
SELECT oficina FROM empleados GROUP BY oficina HAVING AVG(ventas) > 50000
```

NOTA: Para obtener lo que se pide hay que calcular el promedio de ventas de los empleados de cada oficina, por lo que hay que utilizar la tabla empleados. Tenemos que agrupar los empleados por oficina y calcular el promedio para cada oficina, por último nos queda seleccionar del resultado las filas que tengan un promedio superior a 50,000 córdobas.

➤ **¿Cómo se ejecuta internamente una consulta de resumen?**

- Primero se forma la tabla origen de datos según la cláusula FROM.
- Se seleccionan del origen de datos las filas según la cláusula WHERE.
- Se forman los grupos de filas según la cláusula GROUP BY.
- Por cada grupo se obtiene una fila en la tabla resultante con los valores que aparecen en las cláusulas GROUP BY, HAVING y en la lista de selección.
- Se seleccionan de la tabla resultante las filas según la cláusula HAVING.
- Se eliminan de la tabla resultante las columnas que no aparecen en la lista de selección.
- Se ordenan las filas de la tabla resultante según la cláusula ORDER BY.

Una consulta se convierte en consulta de resumen en cuanto aparece GROUP BY, HAVING o una función de columna.

CLASE PRÁCTICA #2: SQL

Dadas las siguientes tablas:

	cod_hospital	dni	apellidos	funcion	salario
▶ 1		12345678	Garcia Hernández, Eladio	CONSERJE	1500
4		22233311	Martínez Molina, Gloria	MEDICO	1600
2		22233322	Tristán García, Ana	MEDICO	1900
2		22233333	Martínez Molina, Andrés	MEDICO	1600
4		33222111	Mesa del Castillo, Juan	MEDICO	2200
3		55544411	Ruiz Hernández, Caridad	MEDICO	1900
4		55544412	Jiménez Jiménez, Dolores	CONSERJE	1200
2		55544433	González Marín, Alicia	CONSERJE	1200
1		66655544	Castillo Montes, Pedro	MEDICO	1700
1		87654321	Fuentes Bermejo, Carlos	DIRECTOR	2000
3		99988333	Serrano Díaz, Alejandro	DIRECTOR	2400
*					

	cod_hospital	dni	apellidos	especialidad
	1	22233311	Roque Vanegas Hector	Internista
	3	22233332	López Hernandez Julio	Psiquiatra
	4	22233333	Esquivel José	Cirujano
	2	33222111	Delgado Carmen	Internista
	2	66655544	Castillo Montes Luis	Cirujano

	cod_hospital	nombre	direccion	num_plazas
	1	Vivian Pellas	Managua	500
	2	Oscar Danilo Rosales	Leon	230
	3	Bertha Calderon	Managua	210
✎	4	La Mascota	Managua	156
	5	La Asuncion	Chontales	120

Responda las siguientes consultas con SQL:

1. Mostrar el número de médicos que pertenecen a cada hospital, ordenado por número descendente de hospital.

Solución: `select nombre,hospital.cod_hospital,count(*) from hospital, medicos where hospital.cod_hospital=medicos.cod_hospital group by hospital.cod_hospital order by cod_hospital desc`

2. Realizar una consulta en la que se muestre por cada hospital el nombre de las especialidades que tiene.

Solución: `select nombre,especialidad,medicos.cod_hospital from hospital,medicos where hospital.cod_hospital=medicos.cod_hospital`

3. Realizar una consulta en la que aparezca por cada hospital y en cada especialidad el número de médicos.

Solución: `select nombre,especialidad,count(*) from hospital,medicos where hospital.cod_hospital=medicos.cod_hospital group by nombre,especialidad`

4. Obtener por cada hospital el número de empleados.

Solución: `select hospital.cod_hospital,nombre,count(*) from personas,hospital where personas.cod_hospital=hospital.cod_hospital group by personas.cod_hospital`

5. Obtener por cada especialidad el número de trabajadores.

Solución: `select especialidad,count(*) from medicos group by especialidad`

6. Visualizar la especialidad que tenga más médicos.

Solución: `select especialidad from medicos group by especialidad having count(*)=(select max(count(*)) from medicos group by especialidad)`

7. ¿Cuál es el nombre del hospital que tiene mayor número de plazas?

Solución: `select cod_hospital,nombre from hospital where num_plazas=(select max(num_plazas) from hospital)`