

Así como la cláusula "where" permite seleccionar (o rechazar) registros individuales; la cláusula "having" permite seleccionar (o rechazar) un grupo de registros.

Si queremos saber la cantidad de libros agrupados por editorial usamos la siguiente instrucción ya aprendida:

```
select editorial, count(*) from libros
group by editorial;
```

Si queremos saber la cantidad de libros agrupados por editorial pero considerando sólo algunos grupos, por ejemplo, los que devuelvan un valor mayor a 2, usamos la siguiente instrucción:

```
select editorial, count(*) from libros
group by editorial
having count(*)>2;
```

Se utiliza "having", seguido de la condición de búsqueda, para seleccionar ciertas filas retornadas por la cláusula "group by".

Veamos otros ejemplos. Queremos el promedio de los precios de los libros agrupados por editorial:

```
select editorial, avg(precio) from libros
group by editorial;
```

Ahora, sólo queremos aquellos cuyo promedio supere los 25 pesos:

```
select editorial, avg(precio) from libros
group by editorial
having avg(precio)>25;
```

En algunos casos es posible confundir las cláusulas "where" y "having". Queremos contar los registros agrupados por editorial sin tener en cuenta a la editorial "Planeta".

Analicemos las siguientes sentencias:

```
select editorial, count(*) from libros
where editorial<>'Planeta'
group by editorial;
select editorial, count(*) from libros
group by editorial
having editorial<>'Planeta';
```

Ambas devuelven el mismo resultado, pero son diferentes.

La primera, selecciona todos los registros rechazando los de editorial "Planeta" y luego los agrupa para contarlos. La segunda, selecciona todos los registros, los agrupa para contarlos y finalmente rechaza la cuenta correspondiente a la editorial

"Planeta".

No debemos confundir la cláusula "where" con la cláusula "having"; la primera establece condiciones para la selección de registros de un "select"; la segunda establece condiciones para la selección de registros de una salida "group by".

Veamos otros ejemplos combinando "where" y "having".

Queremos la cantidad de libros, sin considerar los que tienen precio nulo, agrupados por editorial, sin considerar la editorial "Planeta":

```
select editorial, count(*) from libros
where precio is not null
group by editorial
having editorial<>'Planeta';
```

Aquí, selecciona los registros rechazando los que no cumplan con la condición dada en "where", luego los agrupa por "editorial" y finalmente rechaza los grupos que no cumplan con la condición dada en el "having".

Generalmente se usa la cláusula "having" con funciones de agrupamiento, esto no puede hacerlo la cláusula "where". Por ejemplo queremos el promedio de los precios agrupados por editorial, de aquellas editoriales que tienen más de 2 libros:

```
select editorial, avg(precio) from libros
group by editorial
having count(*) > 2;
```

Podemos encontrar el mayor valor de los libros agrupados por editorial y luego seleccionar las filas que tengan un valor mayor o igual a 30:

```
select editorial, max(precio) from libros
group by editorial
having max(precio)>=30;
```

Esta misma sentencia puede usarse empleando un "alias", para hacer referencia a la columna de la expresión:

```
select editorial, max(precio) as 'mayor' from libros
group by editorial
having mayor>=30;
```

Una empresa tiene registrados sus clientes en una tabla llamada "clientes".

1- Elimine la tabla "clientes", si existe.

2- Créela con la siguiente estructura:

```
create table clientes (  
  codigo int unsigned auto_increment,  
  nombre varchar(30) not null,  
  domicilio varchar(30),  
  ciudad varchar(20),  
  provincia varchar (20),  
  telefono varchar(11),  
  primary key(codigo)  
);
```

3- Ingrese algunos registros:

```
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Lopez Marcos', 'Colon 111', 'Córdoba','Cordoba','null');  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Perez Ana', 'San Martin 222', 'Cruz del Eje','Cordoba','4578585');  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Garcia Juan', 'Rivadavia 333', 'Villa Maria','Cordoba','4578445');  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Perez Luis', 'Sarmiento 444', 'Rosario','Santa Fe',null);  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Pereyra Lucas', 'San Martin 555', 'Cruz del Eje','Cordoba','4253685');  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Gomez Ines', 'San Martin 666', 'Santa Fe','Santa Fe','0345252525');  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Torres Fabiola', 'Alem 777', 'Villa del Rosario','Cordoba','4554455');  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Lopez Carlos', 'Irigoyen 888', 'Cruz del Eje','Cordoba',null);  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Ramos Betina', 'San Martin 999', 'Cordoba','Cordoba','4223366');  
insert into clientes (nombre,domicilio,ciudad,provincia,telefono)  
  values ('Lopez Lucas', 'San Martin 1010', 'Posadas','Misiones','0457858745');
```

4- Obtenga el total de los registros agrupados por provincia:

5- Obtenga el total de los registros agrupados por ciudad y provincia:

6- Obtenga el total de los registros agrupados por ciudad y provincia sin considerar los que tienen menos de 2 clientes:

7- Obtenga el total de los registros sin teléfono nulo, agrupados por ciudad y provincia sin considerar los que tienen menos de 2 clientes: