- HAVING de SQL especifica que una SELECT solo debe **devolver filas donde los valores agregados cumplan las condiciones** especificadas.
- **HAVING** filtra los datos en la fila del grupo pero no en la fila individual.

SINTAXIS

```
SELECT    COLUMN_1,
          COLUMN_2, ..,
          AGGREGATE_FUNCTION (COLUMN_NAME)

FROM TABLE_NAME [WHERE CONDITION]

GROUP BY COLUMN_1, COLUMN_2, ..,
➡️HAVING   [ conditions ]⬅️

ORDER BY ASC|DESC
```

Las **Columna1,. . .,columna n, ..,** no se resumen dentro de una función agregada.

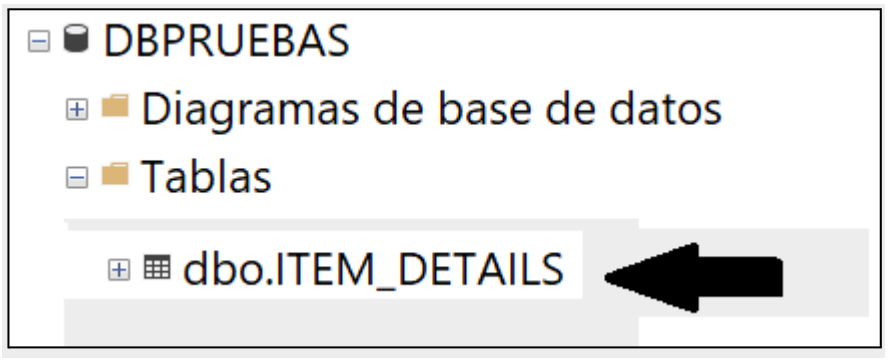
Deben incluirse en la [cláusula GROUP BY](#) .

- **AGGREGATE_FUNCTION** puede ser cualquier función SQL válida como **COUNT ,SUM, MIN,MAX o AVG.**
- **[CONDICIÓN WHERE] es una cláusula WHERE que se utiliza para especificar cualquier condición. Es opcional**.
- **HAVING[condiciones] es una condición que se aplica sólo a los resultados agregados para restringir los grupos de filas devueltas.**
- **ORDER BY** se utiliza para ordenar el resultado en orden ascendente o descendente.
-
- **Veamos un ejemplo de la cláusula Having de SQL.**

Creamos una tabla, la creo en BDPRUEBAS

```
CREATE TABLE dbo.ITEM_DETAILS (

ID INT NOT NULL,

ITEM VARCHAR(50),

ITEM_QTY INT,

ITEM_PURCHASED_DATE DATETIME)
```

Gorka Simplemente me conecto a la BD, hago una nueva consulta y pego el codigo anterior.La tabla se crea correctamente



A continuación añadimos registros a la tabla

```
INSERT INTO dbo.ITEM_DETAILS (ID, ITEM, ITEM_QTY, ITEM_PURCHASED_DATE)

VALUES INSERT INTO dbo.ITEM_DETAILS (ID, ITEM, ITEM_QTY,
ITEM_PURCHASED_DATE)

VALUES

(1, 'Scanner', 300, '20191201'),

(2, 'Scanner', 250, '20191210'),

(3, 'Scanner', 110, '20191112'),

(4, 'Scanner', 110, '20190919'),

(5, 'Scanner', 280, '20190925'),

(6, 'Scanner', 180, '20190815'),

(7, 'Scanner', 180, '20190708'),

(8, 'Pen Drive', 360, '20191215'),

(9, 'Pen Drive', 270, '20190911'),

(10, 'Pen Drive', 210, '20190822'),

(11, 'Pen Drive', 190, '20190819'),

(12, 'Pen Drive', 480, '20190715'),

(13, 'Pen Drive', 680, '20190625'),

(14, 'Pen Drive', 120, '20190618'),

(15, 'Hard Disk', 270, '20190919'),

(16, 'Hard Disk', 210, '20190812'),

(17, 'Hard Disk', 190, '20190829'),

(18, 'Hard Disk', 480, '20190725'),

(19, 'Hard Disk', 680, '20190615'),

(20, 'Hard Disk', 120, '20190628');
```

Gorka Tal y como estaban los datos en la página de donde sigo la explicación me daba

Mens. 242, Nivel 16, Estado 3, Línea 1

La conversión del tipo de datos varchar en datetime produjo un valor fuera de intervalo.

preguntando a Chapgpt me da la siguiente explicación

🔴 Causa del error

Tu SQL Server tiene la configuración regional distinta (por ejemplo `dd/MM/yyyy`) y al insertar fechas como:

```
'2019-12-01 00:00:00.000'
```

SQL Server las interpreta como texto (`varchar`) y al convertirlas a `datetime` no coincide el formato, por eso dice:

> ***"La conversión del tipo de datos varchar en datetime produjo un valor fuera de intervalo"***

Le pido que me haga la corrección necesaria y OK se insertan los registros.

Uso de la cláusula Having con la función SUM

Mostrar artículos cuya suma total de la cantidad comprada sea mayor a 1500.

La siguiente declaración devuelve sólo aquellos artículos para los cuales la suma total de la cantidad comprada es mayor a 1500.

```
SELECT ITEM , SUM(ITEM_QTY) AS TOTAL_QTY
FROM dbo.ITEM_DETAILS
GROUP BY Item
```

HAVING SUM(ITEM_QTY) > 1500

Resultados | Mensajes

	ITEM	TOTAL_QTY
1	Hard Disk	1950
2	Pen Drive	2310

Devuelve dos artículos cuya cantidad total comprada es mayor a 1500.

Entendamos la afirmación anterior dividiéndola en dos partes.

Primera parte: la cláusula GROUP BY agrega la cantidad total comprada para cada artículo individual.

```
SELECT ITEM , SUM(ITEM_QTY) AS TOTAL_QTY
FROM dbo.ITEM_DETAILS
GROUP BY Item
```

ITEM	TOTAL_QTY
Hard Disk	1950
Pen Drive	2310
Scanner	1410

Ahora Having filtra el resultado agregado de GROUP BY para verificar la condición especificada y devuelve sólo aquellos artículos para los cuales la cantidad total comprada es mayor a 1500.

```
SELECT ITEM , SUM(ITEM_QTY) AS TOTAL_QTY
FROM dbo.ITEM_DETAILS
GROUP BY Item
HAVING SUM(ITEM_QTY) >1500
```

	ITEM	TOTAL_QTY
1	Hard Disk	1950
2	Pen Drive	2310

Uso de la cláusula Having con la función Count

La siguiente sentencia devuelve la cantidad total de veces que se ha comprado un artículo.

```
SELECT   ITEM             AS [Nombre artículo],
         SUM(ITEM_QTY)    AS [Unidades vendidas],
         COUNT(ITEM_QTY)  AS [Nº ventas por artículo ]

         FROM             [dbo].[ITEM_DETAILS]
         GROUP BY         ITEM
```

158 %

Resultados | Mensajes

	Nombre artículo	Unidades vendidas	Nº ventas por artículo
1	Hard Disk	1950	6
2	Pen Drive	2310	7
3	Scanner	1410	7

Ahora queremos aquellas donde el nº de ventas por artículo haya sido < 6.

```
SELECT   ITEM             AS [Nombre artículo],
         SUM(ITEM_QTY)    AS [Unidades vendidas],
         COUNT(ITEM_QTY)  AS [Nº ventas por artículo ]

         FROM             [dbo].[ITEM_DETAILS]
         GROUP BY         ITEM
       ➡ HAVING ⬅         COUNT(ITEM_QTY)>6
```

158 %

Resultados | Mensajes

	Nombre artículo	Unidades vendidaso	Nº ventas por artículo
1	Pen Drive	2310	7
2	Scanner	1410	7

Devuelve todos los artículos para los cuales el recuento total comprado que se muestra en la columna Nº ventas por artículo es mayor que 6.

Uso de la cláusula Order by con la cláusula Having

Modifiquemos la primera consulta que muestra los artículos cuya suma total de la cantidad comprada es mayor a 1500.

En esta declaración, ordenaremos el resultado por **nombre del elemento** en **orden descendente** .

```
SELECT      ITEM            AS [Nombre artículo],
            SUM(ITEM_QTY)   AS [Unidades vendidas]
FROM        dbo.ITEM_DETAILS
GROUP BY    ITEM
HAVING      SUM(ITEM_QTY) > 1500
ORDER BY    ITEM DESC
```

Resultados | Mensajes

	Nombre artículo	Unidades vendidas
1	Pen Drive	2310
2	Hard Disk	1950

```
 8  SELECT      ITEM            AS [Nombre artículo],
 9              SUM(ITEM_QTY)   AS [Unidades vendidas]
10  FROM        dbo.ITEM_DETAILS
11  GROUP BY    ITEM
12  HAVING      SUM(ITEM_QTY) > 1500
13  ORDER BY    ITEM ASC
```

158 %

Resultados | Mensajes

	Nombre artículo	Unidades vendidas
1	Hard Disk	1950
2	Pen Drive	2310

He puesto también el orden **ASC**ENDENTE