

## Falta TEMA HAVING I OPERADORES ANTERIORES

Where – Having

Consola SQL

Por último vamos a explicar. El

- Unión externa
  - UNION
  - UNION ALL
  - EXCEPT\*
  - INTERSECT\*
  - MINUS\*
- Unión interna
  - INNER JOIN
  - LEFT JOIN
  - RIGHT JOIN



\*Forman parte del standar pero no leen  
ACCESS o MySQL



## UNION / UNION ALL (Unión externa)

Unión de dos (o+) tablas = número de campos y datos **compatibles**

ID	DESCRIPCION	CANTIDAD	PRECIO	FECHA	PAIS
1	DEPORTES	MANCUERNAS	60,00	13/09/2000	VERDADERO USA
2	CONFECCIÓN	SERRUCHO	30,20	23/03/2001	VERDADERO FRANCIA
3	JUJETERIA	CORREPASILLOS	103,34	11/04/2000	VERDADERO JAPON
4	CONFECCIÓN	PANTALÓN SEÑORA	174,23	10/01/2000	VERDADERO MARRUECOS
5	ALTA COSTURA	PANTALÓN SEÑORA	1.174,23	10/01/2000	VERDADERO MARRUECOS
6	ALTA COSTURA	CAMISA CABALLERO	1.067,13	11/08/2002	FALSO ESPAÑA
7	DEPORTES DE RIESGO	PISTOLA OLIMPICA	1.046,73	02/02/2001	VERDADERO SUECIA
8	ALTA COSTURA	BLUSA SRA.	1.101,06	18/03/2000	VERDADERO CHINA



ID	DESCRIPCION	CANTIDAD	PRECIO	FECHA	PAIS
1	DEPORTES	MANCUERNAS	60,00	13/09/2000	VERDADERO USA
2	CONFECCIÓN	SERRUCHO	30,20	23/03/2001	VERDADERO FRANCIA
3	JUJETERIA	CORREPASILLOS	103,34	11/04/2000	VERDADERO JAPON
4	CONFECCIÓN	PANTALÓN SEÑORA	174,23	10/01/2000	VERDADERO MARRUECOS
5	ALTA COSTURA	PANTALÓN SEÑORA	1.174,23	10/01/2000	VERDADERO MARRUECOS
6	ALTA COSTURA	CAMISA CABALLERO	1.067,13	11/08/2002	FALSO ESPAÑA
7	DEPORTES DE RIESGO	PISTOLA OLIMPICA	1.046,73	02/02/2001	VERDADERO SUECIA
8	ALTA COSTURA	BLUSA SRA.	1.101,06	18/03/2000	VERDADERO CHINA



Tabla  
resultante:

AR06	DEPORTES	MANCUERNAS	60,00	13/09/2000	VERDADERO USA
AR07	CONFECCIÓN	SERRUCHO	30,20	23/03/2001	VERDADERO FRANCIA
AR08	JUJETERIA	CORREPASILLOS	103,34	11/04/2000	VERDADERO JAPON
AR09	CONFECCIÓN	PANTALÓN SEÑORA	174,23	10/01/2000	VERDADERO MARRUECOS
AR54	ALTA COSTURA	PANTALÓN SEÑORA	1.174,23	10/01/2000	VERDADERO MARRUECOS
AR55	ALTA COSTURA	CAMISA CABALLERO	1.067,13	11/08/2002	FALSO ESPAÑA
AR56	DEPORTES DE RIESGO	PISTOLA OLIMPICA	1.046,73	02/02/2001	VERDADERO SUECIA
AR57	ALTA COSTURA	BLUSA SRA.	1.101,06	18/03/2000	VERDADERO CHINA

Une los registros de ambas tablas según la consulta.



## UNION / UNION ALL (Unión externa)

Consulta para  
unir por  
completo dos  
tablas:

**SELECT \* FROM PRODUCTOS**  
**UNION**  
**SELECT \* FROM PRODUCTOSNUEVOS**



Pueden tener campos con nombres diferentes, la consulta toma el nombre de los campos de la primera tabla. Se pueden unir más de dos tablas.

## CONSULTAS UNION

### Ejercicios de práctica

1. Realizar una consulta que visualice los registros de la tabla **PRODUCTOS** con un **precio** entre 200 y 500 y además los registros de la tabla **PRODUCTOSNUEVOS** donde el **precio** sea superior a 1100.
2. Realizar una consulta visualizando los campos **FECHA, SECCIÓN, NOMBRE ARTÍCULO** y **PRECIO** de la tabla **PRODUCTOS** sólo a los artículos de la sección **CERÁMICA**. También que se vean los productos de ALTA COSTURA de la tabla **PRODUCTOSNUEVOS**.
3. Une las dos tablas mostrando solo los productos del año 2001. Da igual si hay campos repetidos, muéstralos.

Estamos hablando de una unión entre diferentes tablas que han de compartir una característica en común. En este caso, estaremos haciendo referencia al mismo rango a una similitud en los campos.

Además en el caso del UNION: Será para hacer la unión pero no que se repitan aquellos valores que ya existan, en cambio cuando hablamos de **UNION ALL**: **estamos haciendo referencia al hecho de que no pasa nada si estos se repiten.**

Fer exercicis 1,2,3 de UNION. Repassar les anteriors. Acabar els apunts del Having not Having i repassar els apunts.

### Operadores SQL

**LIMIT** (limita los resultados)

**LIMIT 1** (solo saldrá un resultado)

**LIMIT 100** (saldrán los 100 primeros resultados)

**LIMIT 0,3** (saldrán resultados desde el primer valor, 3 valores totales)

### Consultas de cálculo:

## CONSULTAS DE CÁLCULO

Now()	Nos devuelven la hora y fecha de hoy
Datediff()	Diferencia entre dos fechas
Date_format()	O Format(), Round, Truncate...
Concat()	Concatenar cadenas de texto

### EL ORDEN DE ESCRITURA

## ORDEN DE ESCRITURA

1 COMANDO\*

2 CLÁUSULAS\*

FROM + WHERE + GROUP BY + HAVING + ORDER BY

### WHERE

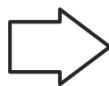
SELECT SECCIÓN  
FROM PRODUCTOS  
WHERE SECCIÓN='CONFECCIÓN' OR SECCIÓN='DEPORTES'  
GROUP BY SECCIÓN;

## WHERE

1

WHERE  
SECCIÓN='CONFECCIÓN'  
OR SECCIÓN='DEPORTES'  
muestra 19 registros en  
esa condición

SECCIÓN
CONFECCIÓN
DEPORTES
DEPORTES
CONFECCIÓN
CONFECCIÓN
DEPORTES
CONFECCIÓN
CONFECCIÓN
DEPORTES
DEPORTES
CONFECCIÓN
DEPORTES
CONFECCIÓN
DEPORTES
CONFECCIÓN
DEPORTES
DEPORTES
DEPORTES



2

GROUP BY SECCIÓN  
muestra de esos 19  
registros solo en 2

SECCIÓN
CONFECCIÓN
DEPORTES

SELECT SECCIÓN  
FROM PRODUCTOS  
WHERE SECCIÓN='OFICINA'  
GROUP BY SECCIÓN;

## WHERE

1

WHERE SECCIÓN='OFICINA'  
muestra tan solo 1 registro en  
esa condición

SECCIÓN
OFICINA



2

GROUP BY SECCIÓN  
muestra de 1 registro

SECCIÓN
OFICINA

Si solo tenemos una sección, y agrupamos por secciones solo nos mostrará un registro.

### Having

SELECT SECCIÓN  
FROM PRODUCTOS GROUP BY SECCIÓN  
HAVING SECCIÓN='CONFECCIÓN' OR SECCIÓN='DEPORTES'

## HAVING

1

GROUP BY SECCIÓN  
muestra 6 registros

SECCIÓN
CERÁMICA
CONFECCIÓN
DEPORTES
FERRETERÍA
JUGUETERÍA
OFICINA



2

SECCIÓN
CONFECCIÓN
DEPORTES

HAVING SECCIÓN='CONFECCIÓN' OR  
SECCIÓN='DEPORTES'  
muestra 2 registros en esa condición  
de los 6 registros anteriores

En este caso se utiliza having en el SQL para explicitar que se mostrarán aquellos elements que en la sección tenga Confección o bien Deportes. En el caso del Group By pasará solo a mostrar dos registros en esa condición de los 6 registros anteriores.

SELECT SECCIÓN  
FROM PRODUCTOS GROUP BY SECCIÓN  
HAVING SECCIÓN='OFICINA'

## HAVING

1

GROUP BY SECCIÓN  
muestra 6 registros

SECCIÓN
CERÁMICA
CONFECCIÓN
DEPORTES
FERRETERÍA
JUGUETERÍA
OFICINA



2

SECCIÓN
OFICINA

HAVING SECCIÓN='OFICINA'  
muestra 1 registro

En el primer caso de select se agrupa por sección. En el segundo caso se pone aquellos que tengan en la sección Oficina. Así que como solo pongo Having Sección='Oficina' aparecerá solo oficina.

<https://carmoreno.com.co/sql/2017/02/09/Diferencia-entre-having-y-where/> Explicación del uso de having.

Las dos consultas anteriores retornan los mismos registros, pero se comportan totalmente distinto. En la primera, seleccionamos genero, director y la suma del recaudo siempre y cuando el genero sea 'Drama' (WHERE) y posteriormente los agrupamos por genero y director (GROUP BY).

En la segunda seleccionamos el genero, director y hacemos la suma del recaudo, sin importar si el genero es o no 'Drama', luego los agrupamos por genero y director (GROUP BY). Por último seleccionamos solo los registros cuyo genero sea 'Drama' (HAVING). Además, si prestaste atención, el resultado de la consulta hecha con HAVING se demora el doble de tiempo que la consulta hecha con WHERE (0.008 seg y 0.004 seg respectivamente).

Quizá te estés preguntando ¿cuándo usar HAVING o WHERE?, desde mi punto de vista, **deberíamos usar HAVING solo cuando se vea implicado el uso de funciones de grupo (AVG, SUM, COUNT, MAX, MIN)**, debido a que con WHERE no podemos realizar condiciones que impliquen estas funciones. Por ejemplo, si intentas esto, tendrás un error: