Sentencias SQL avanzadas

IT BOARDING

BOOTCAMP

Índice

IT BOARDING
BOOTCAMP

GROUP BY

// Agrupación de resultados

IT BOARDING

BOOTCAMP

GROUP BY

- Agrupa los resultados según las columnas indicadas.
- Genera un solo registro por cada grupo de filas que compartan las columnas indicadas.
- Reduce la cantidad de filas de la consulta.
- Se suele utilizar en conjunto con **funciones de agregación**, para obtener **datos resumidos y agrupados** por las columnas que se necesiten.

GROUP BY

➔ Por ejemplo, si se tiene una tabla con los pagos realizados por cada persona y se quiere saber cuánto gastó cada una en total. ¿Cómo se podría realizar el reporte?

id	dni	fecha	pago
1	33.241.677	01/01/2017	50
2	35.186.928	02/01/2017	60
3	33.241.677	03/01/2017	70
4	35.186.928	04/01/2017	40

*Tabla **clientes_pagos** que almacena pagos de clientes*

GROUP BY

➔ ¿Cómo se podría descomponer esta consulta?

SELECT dni

SUM (pago) = total

dni	total
35.186.928	100
33.241.677	120

GROUP BY dni

Resultado con los dni y montos pagados por cada cliente

GROUP BY

¿Cómo funciona para este caso?

Agrupando por DNI, se crean grupos diferentes por cada DNI que exista en la tabla.

En este ejemplo, existen dos grupos:

id	dni	fecha	pago
1	33.241.677	01/01/2017	50
2	35.186.928	02/01/2017	60
3	33.241.677	03/01/2017	70
4	35.186.928	04/01/2017	40

id	dni	fecha	pago
1	33.241.677	01/01/2017	50
3	33.241.677	03/01/2017	70

AGRUPACIÓN

id	dni	fecha	pago
2	35.186.928	02/01/2017	60
4	35.186.928	04/01/2017	40

GROUP BY

Sobre cada grupo, se aplica la función de agregación que se indicó en el SELECT.
En este caso, se aplica la función de agregación SUM sobre la columna pago.
El resultado de la consulta, es una tabla que contiene el resultado de cada grupo.

SUM

id	dni	fecha	pago
1	33.241.677	01/01/2017	50
3	33.241.677	03/01/2017	70

SUM

id	dni	fecha	pago
2	35.186.928	02/01/2017	60
4	35.186.928	04/01/2017	40

dni	total
35.186.928	100
33.241.677	120

RESULTADO

Ejemplo: GROUP BY

➔ Aquí podemos observar otro ejemplo de la sentencia GROUP BY con la BD movies.

```
SELECT COUNT(*), mo.title, mo.rating, mo.awards
FROM movies mo INNER JOIN actors ac
ON mo.id = ac.favorite_movie_id
GROUP BY title;
```

HAVING

// Condiciones sobre grupos de registros

IT BOARDING

BOOTCAMP

HAVING

Es muy *similar* a la cláusula WHERE, pero en lugar de afectar a las filas de la tabla, afecta a los grupos obtenidos por el GROUP BY.

La cláusula **HAVING** se utiliza para incluir condiciones con algunas funciones SQL.

WHERE opera sobre registros individuales, mientras que ***HAVING*** lo hace sobre un grupo de registros.

HAVING

Continuando con el ejemplo visto anteriormente, si se desea obtener solo los clientes que realizaron pagos totales superiores a 100.

dni	total
35.186.928	100
33.241.677	120

dni	total
33.241.677	120

HAVING total > 100

HAVING

➔ Si realizamos la consulta a la tabla **clientes_pagos** podría ser de la siguiente forma:

```
SELECT dni, SUM(pago) AS total
FROM clientes_pagos
GROUP BY dni
HAVING total>100;
```

Ejemplo: HAVING

➔ Aquí podemos observar otro ejemplo de la sentencia HAVING con la BD movies.

```
SELECT COUNT(*) AS tot_act, mo.title, mo.rating, mo.awards
FROM movies mo INNER JOIN actors ac
ON mo.id = ac.favorite_movie_id
GROUP BY title HAVING tot_act > 2;
```

WHERE - GROUP BY - HAVING

Para ver el orden de ejecución y diferencia entre estas cláusulas, sigamos este ejemplo.

Si se desea obtener solo las personas que realizaron pagos por un total superior a 100, pero considerando que cada compra individual haya sido superior a 50.

id	dni	fecha	pago
1	33.241.677	01/01/2017	50
2	35.186.928	02/01/2017	60
3	33.241.677	03/01/2017	70
4	35.186.928	04/01/2017	55

WHERE - GROUP BY - HAVING

➔ Si realizamos la consulta a la tabla **clientes_pagos** podría ser de la siguiente forma:

```
SELECT dni, SUM(pago) AS total
FROM clientes_pagos
WHERE pago>50
GROUP BY dni
HAVING total>100;
```

WHERE - GROUP BY - HAVING

Resumiendo la ejecución, en primera instancia con WHERE se filtran los pagos mayores a 50, luego se realiza la agrupación (GROUP BY) por dni. Seguidamente, se aplica la función de agregación, sumando los pagos de cada agrupación por dni.

Por último, la cláusula HAVING filtra aquellos totales mayores a 100.

dni	total
35.186.928	115

Resultado

WHERE - GROUP BY - HAVING

ORDEN DE EJECUCIÓN

Gracias!

IT BOARDING

BOOTCAMP