



UT5. CONSULTAS DE RECUPERACIÓN MONOTABLA EN SQL.

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

Invoice

Invoice_id
Customer_id
Order_id
Product_id
Date_time
Status
Total
Remark



CONTENIDOS

- Consultas de resumen (Sentencia SELECT Parte II).
 - Agrupamiento de registros.
 - Filtrado sobre agrupaciones.



AGRUPAMIENTO DE REGISTROS



SINTAXIS DE SELECT HASTA AHORA

```
SELECT [ DISTINCT ] select_expr [, select_expr ...]  
FROM table_references  
[ WHERE filtro ]  
[ ORDER BY {col_name | expr | position} [ASC | DESC] , ... ]  
[ LIMIT [ desplazamiento , ] nfilas ]
```



AGRUPACIONES DE REGISTROS O FILAS

- Se pueden hacer agrupaciones de filas de la tabla del FROM por uno o varios criterios.
- Normalmente, cada criterio es una columna.
- Para hacer estas agrupaciones, se utiliza GROUP BY después del WHERE (y antes del ORDER BY).

GROUP BY *expr* [, *expr* ...]

- Si se realizan agrupaciones, en la parte del SELECT solo pueden aparecer las expresiones (columnas) indicadas en el GROUP BY. También pueden aparecer las consultas de resumen que veremos después.
- Los valores nulos se consideran iguales. Para GROUP BY, NULL=NULL



GROUP BY: EJEMPLOS

SELECT editorial FROM libros GROUP BY editorial;

LIBROS			
Codigo	Titulo	Num_paginas	Editorial
34587	Int. Artificial	50	Paraninfo
1022305	Concep. Y Dis.	48	Rama
493942	Turbo C++	125	Mc Graw-Hill
45307	Virus Informát.	50	NULL
21566	Optimización	30	Paraninfo
12548	Seguridad	50	Paraninfo
568978	Hardware	70	Paraninfo
112313	Sist. Informac.	358	Rama



Editorial
Paraninfo
Rama
Mc Graw-Hill
NULL

Se agrupan todas las filas que tengan el mismo valor en editorial (hay 4 valores distintos, por lo que salen 4 grupos). De cada grupo, se muestra el campo editorial.



OTRA EXPLICACIÓN DE GROUP BY

SELECT editorial FROM libros GROUP BY editorial;

LIBROS			
Codigo	Titulo	Num_paginas	Editorial
34587	Int. Artificial	50	Paraninfo
21566	Optimización	30	Paraninfo
12548	Seguridad	50	Paraninfo
568978	Hardware	70	Paraninfo
112313	Sist. Informac.	358	Rama
1022305	Concep. Y Dis.	48	Rama
493942	Turbo C++	125	Mc Graw-Hill
45307	Virus Informát.	50	NULL



Editorial
Paraninfo
Rama
Mc Graw-Hill
NULL

En vez de tener todas las filas, al agruparse, es como si solo se tuviera una fila de cada grupo (cada una marcada con un color). Como en cada fila solo se puede tener un valor para cada columna, solo la columna **editorial** se puede utilizar (el resto de columnas no tienen valores únicos).



GROUP BY: EJEMPLOS

- Si aparecen varios criterios, se realiza la agrupación por todos ellos a la vez. Es decir, que se agrupan las filas que para esas columnas tengan el mismo valor.

```
SELECT num_paginas, editorial FROM libros GROUP BY num_paginas, editorial;
```

LIBROS			
Codigo	Titulo	Num_paginas	Editorial
34587	Int. Artificial	50	Paraninfo
1022305	Concep. Y Dis.	48	Rama
493942	Turbo C++	125	Mc Graw-Hill
45307	Virus Informát.	50	NULL
21566	Optimización	30	Paraninfo
12548	Seguridad	50	Paraninfo
568978	Hardware	70	Paraninfo
112313	Sist. Informac.	358	Rama



Num_paginas	Editorial
50	Paraninfo
48	Rama
125	Mc Graw-Hill
50	NULL
30	Paraninfo
70	Paraninfo
358	Rama



GROUP BY: EJEMPLOS

- No tiene porqué mostrarse (en las expresiones del SELECT) todas las columnas:

SELECT editorial FROM libros GROUP BY num_paginas, editorial;

LIBROS			
Codigo	Titulo	Num_paginas	Editorial
34587	Int. Artificial	50	Paraninfo
1022305	Concep. Y Dis.	48	Rama
493942	Turbo C++	125	Mc Graw-Hill
45307	Virus Informát.	50	NULL
21566	Optimización	30	Paraninfo
12548	Seguridad	50	Paraninfo
568978	Hardware	70	Paraninfo
112313	Sist. Informac.	358	Rama



Editorial
Paraninfo
Rama
Mc Graw-Hill
NULL
Paraninfo
Paraninfo
Rama



GROUP BY: EJEMPLOS

- Se pueden utilizar expresiones en el SELECT, pero solo sobre las columnas que aparecen en el GROUP BY
- ```
SELECT num_paginas * 100 AS num_paginas, editorial FROM libros GROUP BY num_paginas, editorial;
```

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |



| Num_paginas | Editorial    |
|-------------|--------------|
| 5000        | Paraninfo    |
| 4800        | Rama         |
| 12500       | Mc Graw-Hill |
| 5000        | NULL         |
| 3000        | Paraninfo    |
| 7000        | Paraninfo    |
| 35800       | Rama         |



# CONSULTAS DE RESUMEN

- Se pueden obtener como expresión consultas que resumen cierta información calculada a partir de un conjunto de registros. Para ello, se utilizan las funciones de columna, de agregación o de agrupación (reciben todos estos nombres):
  - **SUM (expr)** → Suma los valores resultado de la expresión. Sirve para números.
  - **AVG (expr)** → Calcula la media de los valores. Sirve para números.
  - **MIN (expr)** → Calcula el mínimo de los valores. Sirve para números, y para cadenas (orden alfabético).
  - **MAX (expr)** → Calcula el máximo de los valores. Sirve para números, y para cadenas (orden alfabético).
  - **COUNT (expr)** → Cuentan el número de valores. Si la expresión es una columna, no contabiliza los nulos. Si la expresión es '\*', incluye los nulos.



# CONSULTAS DE RESUMEN

- Suelen utilizarse en sentencias con GROUP BY, para poder calcular un valor del resto de columnas por los que no se agrupa.
- Si se aplica sin GROUP BY, devuelve solo un valor (solo una fila), por lo que no debería utilizarse con otras columnas en el SELECT.
- A cada columna de tipo consulta resumen que aparezca en un SELECT de este tipo, debería darse un alias.
- La expresión, normalmente es una columna.



# CONSULTAS DE RESUMEN: EJEMPLO

```
SELECT COUNT(*) AS num_libros FROM libros;
```

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 21567   | Optimización2   | 30          | NULL         |



| num_libros |
|------------|
| 9          |

Cuenta cuantos libros hay. Se ha dado el alias “num\_libros”. Como no está en un group by, cuenta el total de libros (total de filas) de la tabla indicada en el FROM (libros).



# CONSULTAS DE RESUMEN: EJEMPLO

SELECT COUNT(editorial) AS num\_libros FROM libros;

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 21567   | Optimización2   | 30          | NULL         |



| num_libros |
|------------|
| 7          |

Cuenta cuantos libros hay que tienen editorial. Es decir, cuenta cuantos valores distintos de NULL hay en editorial.





# CONSULTAS DE RESUMEN: EJEMPLO

```
SELECT COUNT(DISTINCT editorial) AS num_libros FROM libros;
```

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 21567   | Optimización2   | 30          | NULL         |



| num_libros |
|------------|
| 3          |

Dentro del count va una expresión (normalmente una columna), que puede ir precedida de distinct para calcular los valores diferentes. En este caso, cuenta cuantos valores distintos de NULL hay en el campo editorial.



# CONSULTAS DE RESUMEN: EJEMPLO



```
SELECT COUNT(num_paginas), editorial AS num_libros FROM libros;
```

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 21567   | Optimización2   | 30          | NULL         |

Quien en el examen intente recuperar en el select una columna que no esté en el GROUP BY, salvo dentro de una función resumen, ¡¡está suspenso, aunque haga bien el resto de las consultas!!

Aunque algunos gestores (como MySQL) lo permiten, y devuelven un resultado, no es correcto utilizar otra columna (u expresión) si se realiza una consulta resumen.



# CONSULTAS DE RESUMEN: EJEMPLO

```
SELECT AVG(num_paginas) AS media_pag FROM libros;
```

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 21567   | Optimización2   | 30          | NULL         |



| Media_pag |
|-----------|
| 90,11     |

Calcula la media del número de páginas. Si una fila no tuviera número de páginas (valor NULL), no se tiene en cuenta para la media.



# CONSULTAS DE RESUMEN: EJEMPLO

```
SELECT AVG(DISTINCT num_paginas) AS media_pag FROM libros;
```

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 21567   | Optimización2   | 30          | NULL         |



| Media_pag |
|-----------|
| 113,5     |

Primero calcula los valores distintos de num\_paginas, y con estos valores, calcula la media. No tiene en cuenta los nulos.



# CONSULTAS DE RESUMEN: EJEMPLO

```
SELECT editorial, AVG(num_paginas) AS pag FROM libros GROUP BY editorial;
```

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21567   | Optimización2   | 30          | NULL         |



| Editorial    | pag |
|--------------|-----|
| Paraninfo    | 50  |
| Rama         | 203 |
| Mc Graw-Hill | 125 |
| NULL         | 40  |

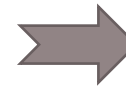
Para cada uno de los grupos, se calcula la media del número de páginas. El criterio con los nulos es el mismo que se ha mencionado, por eso, si algún valor de num\_paginas fuera nulo, no se utilizaría para calcular la media.



# CONSULTAS DE RESUMEN: EJEMPLO

SELECT editorial, AVG(num\_paginas) AS pag, MAX(código) as codM FROM libros GROUP BY editorial;

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21567   | Optimización2   | 30          | NULL         |



| Editorial    | pag | codM    |
|--------------|-----|---------|
| Paraninfo    | 50  | 568978  |
| Rama         | 203 | 1022305 |
| Mc Graw-Hill | 125 | 493942  |
| NULL         | 40  | 45307   |

Pueden aparecer varias funciones de columna en la misma sentencia (tanto con GROUP BY como sin él). Concretamente en la parte SELECT cuando hay un GROUP BY solo pueden utilizarse la columnas del GROUP BY, y el resto de columnas dentro de una función de columna.

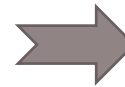




# CONSULTAS DE RESUMEN: EJEMPLO

SELECT editorial, COUNT(\*) AS libros, COUNT(editorial) as edis FROM libros GROUP BY editorial;

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21567   | Optimización2   | 30          | NULL         |



| Editorial    | libros | edis |
|--------------|--------|------|
| Paraninfo    | 4      | 4    |
| Rama         | 2      | 2    |
| Mc Graw-Hill | 1      | 1    |
| NULL         | 2      | 0    |

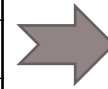
Pueden aparecer varias funciones de columna en la misma sentencia (tanto con GROUP BY como sin él). Concretamente en la parte SELECT cuando hay un GROUP BY solo pueden utilizarse la columnas del GROUP BY, y el resto de columnas dentro de una función de columna.



# CONSULTAS DE RESUMEN: EJEMPLO

```
SELECT editorial, count(*) AS libros, AVG(num_paginas) AS pag FROM libros GROUP BY num_paginas, editorial;
```

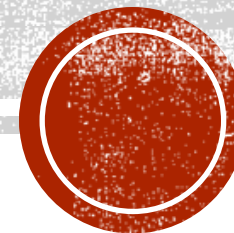
| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 21567   | Optimización2   | 30          | NULL         |



| Editorial    | libros | Pag |
|--------------|--------|-----|
| Paraninfo    | 2      | 50  |
| Rama         | 1      | 48  |
| Mc Graw-Hill | 1      | 125 |
| NULL         | 1      | 50  |
| Paraninfo    | 1      | 30  |
| Paraninfo    | 1      | 70  |
| Rama         | 1      | 358 |
| NULL         | 1      | 30  |



# FILTRADO SOBRE AGRUPAMIENTOS



# FILTRADO SOBRE AGRUPACIONES

- Los filtros sobre agrupaciones de filas, deben realizarse mediante el uso de la cláusula HAVING, ya que WHERE actúa antes de agrupar los registros.

```
SELECT [DISTINCT] select_expr [, select_expr ...]
FROM table_references
[WHERE filtro]
[GROUP BY expr [, expr ...]
 [HAVING filtro_grupos]]
```

- Con HAVING se aplican filtros sobre grupos, es decir, la restricción la debe cumplir el grupo (no cada fila). Esta es la única diferencia que existen con los filtros del WHERE (además del momento en el que se utilizan). Por ello, el filtro de HAVING suele tener funciones de agregación.



# HAVING: EJEMPLO

SELECT editorial, AVG(num\_paginas) AS pag FROM libros GROUP BY editorial HAVING COUNT(\*) >= 2;

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21567   | Optimización2   | 30          | NULL         |



| Editorial | pag |
|-----------|-----|
| Paraninfo | 50  |
| Rama      | 203 |
| NULL      | 40  |

Para cada uno de los grupos, se muestran (filtran) los que contengan 2 o más filas originales. Por eso se elimina la fila agrupa de editorial = 'Mc Graw-Hill'.



# HAVING: EJEMPLO

SELECT editorial, AVG(num\_paginas) AS pag FROM libros GROUP BY editorial HAVING AVG(num\_paginas) < 100;

| LIBROS  |                 |             |              |
|---------|-----------------|-------------|--------------|
| Codigo  | Titulo          | Num_paginas | Editorial    |
| 34587   | Int. Artificial | 50          | Paraninfo    |
| 21566   | Optimización    | 30          | Paraninfo    |
| 12548   | Seguridad       | 50          | Paraninfo    |
| 568978  | Hardware        | 70          | Paraninfo    |
| 112313  | Sist. Informac. | 358         | Rama         |
| 1022305 | Concep. Y Dis.  | 48          | Rama         |
| 493942  | Turbo C++       | 125         | Mc Graw-Hill |
| 45307   | Virus Informát. | 50          | NULL         |
| 21567   | Optimización2   | 30          | NULL         |



| Editorial | pag |
|-----------|-----|
| Paraninfo | 50  |
| NULL      | 40  |

Para cada uno de los grupos, se muestran (filtran) los que contengan una media de número de páginas menor que 100.





# HAVING: EJEMPLO

SELECT editorial, AVG(num\_paginas) AS pag FROM libros WHERE num\_paginas <> 30 GROUP BY editorial HAVING AVG(num\_paginas) < 100;

| LIBROS           |                          |               |                      |
|------------------|--------------------------|---------------|----------------------|
| Codigo           | Titulo                   | Num_paginas   | Editorial            |
| 34587            | Int. Artificial          | 50            | Paraninfo            |
| <del>21566</del> | <del>Optimización</del>  | <del>30</del> | <del>Paraninfo</del> |
| 12548            | Seguridad                | 50            | Paraninfo            |
| 568978           | Hardware                 | 70            | Paraninfo            |
| 112313           | Sist. Informac.          | 358           | Rama                 |
| 1022305          | Concep. Y Dis.           | 48            | Rama                 |
| 493942           | Turbo C++                | 125           | Mc Graw-Hill         |
| 45307            | Virus Informát.          | 50            | NULL                 |
| <del>21567</del> | <del>Optimización2</del> | <del>30</del> | <del>NULL</del>      |



| Editorial | pag   |
|-----------|-------|
| Paraninfo | 56.66 |
| NULL      | 50    |

Se puede combinar WHERE y HAVING. Primero se realiza el WHERE (se elimina los libros con 30 páginas), después el GROUP BY, y por último, el HAVING.





### Invoice

|             |
|-------------|
| Invoice_id  |
| Customer_id |
| Order_id    |
| Product_id  |
| Date_time   |
| Status      |
| Total       |
| Remark      |

# UT5. CONSULTAS DE RECUPERACIÓN MONOTABLA EN SQL.

Módulo: BASES DE DATOS

Curso 2022/2023. 1º DAM

Ruth Lospitao Ruiz

