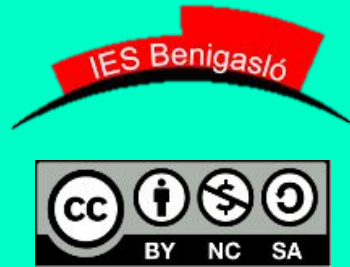


AGRUPAMENT DE DADES

CFGs DAM



1. AGRUPAMENT DE DADES

- Una **consulta agrupada** s'utilitza per considerar els registres on **alguns camps** dels quals **tenen el mateix valor**, i processar-los de la mateixa manera, per comptar-los, sumar-los, fer la mitjana etc.

1.1. GROUP BY

- La clàusula **GROUP BY** unida a un **SELECT** permet agrupar files segons les columnes que s'indiquen com a paràmetres, i se sol utilitzar en conjunt **amb les funcions d'agrupació**, per obtenir dades resumides i agrupades per les columnes que es necessiten.
- En aquests casos, **GROUP BY divideix el conjunt de resultats en grups de files** i les **funcions agregades fan un càlcul a cada grup**, per exemple, màxim, mínim, mitja, etc.
 - És a dir, per a cada grup, podeu aplicar una funció agregada, per exemple, `SUM()` per calcular la suma d'elements o `COUNT()` per obtenir el nombre d'elements dels grups.

1.1. GROUP BY

- La clàusula divideix les files pels valors de les columnes especificades a la clàusula GROUP BY i calcula un valor per a cada grup.
- La sintaxi bàsica de la clàusula GROUP BY:

SELECT columna1, columna2, ..., funcio_agregada(columna_n)

FROM nom_taula

GROUP BY columna_1, columna_2, ...;

Utilitzar la clàusula GROUP BY no garanteix que les dades es tornen ordenades. Sol ser una pràctica recomanable incloure una clàusula ORDER BY per les mateixes columnes que utilitzem en GROUP BY, especificant l'ordre que ens interesse.

1.1. GROUP BY

- En resum:
- Primer, seleccioneu les columnes que vulgueu agrupar, per exemple, columna1 i columna2, i la columna que vulgueu aplicar una funció agregada (columna3).
- En segon lloc, enumereu les columnes que voleu agrupar a la clàusula GROUP BY.

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **1) Exemple de GROUP BY sense cap funció agregada.** Podem utilitzar la clàusula **GROUP BY** sense aplicar cap funció agregada. La consulta següent obté dades de la taula de vendes (que té 28 files) i agrupa el resultat per identificador de client.

```
1 SELECT client
2 FROM vendes
3 GROUP BY client
4 ORDER BY client ASC;
```

→ Seleccionem el codi del client (columna client) de la taula vendes, agrupem per codi de client (les vendes del mateix client s'agrupen) i ordenem per codi de client.

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **1) Exemple de GROUP BY sense cap funció agregada.**
- El resultat ens retorna 17 files; ha agrupat les línies de vendes per clients i per això no es mostren codis de client repetits.
- En aquest cas, **GROUP BY** funciona com la clàusula DISTINCT que elimina les files duplicades del conjunt de resultats.

	client character varying (10)
1	C01
2	C02
3	C03
4	C05
5	C06
6	C07
7	C08
8	C09
9	C10
10	C11
11	C12
12	C13
13	C14
14	C15
15	C16
16	C19
17	C20

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **2) Exemple de GROUP BY amb la funció agregada SUM.** La clàusula **GROUP BY** és molt útil quan s'utilitza conjuntament amb una funció agregada.
- Per exemple, per seleccionar les unitats totals que ha comprat cada client, utilitzem la clàusula **GROUP BY** per dividir les files de la taula de vendes en grups, agrupades per identificador de client. Per a cada grup, calculem les unitats totals de productes comprats mitjançant la funció SUM ().

```
SELECT client, SUM(unitats)  
FROM vendes  
GROUP BY client  
ORDER BY client ASC;
```


1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **2) Exemple de GROUP BY amb la funció agregada SUM.**
- La clàusula **GROUP BY** ordena el resultat establert per l'identificador de client i suma les unitats que pertanyen al mateix client. Sempre que canvia el client, afegeix la fila al conjunt de resultats retornats.

	client character varying (10)	sum bigint
1	C01	24
2	C02	17
3	C03	5
4	C05	7
5	C06	10
6	C07	19
7	C08	13
8	C09	24
9	C10	15
10	C11	55
11	C12	17
12	C13	10
13	C14	46
14	C15	4
15	C16	23
16	C19	45
17	C20	3

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- Com hem comentat, també podem utilitzar les funcions agregades en la clàusula **ORDER BY**. Per exemple:

```
SELECT client, SUM(unitats)
FROM vendes
GROUP BY client
ORDER BY SUM(unitats) DESC;
```

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **3) Exemple de GROUP BY amb la funció agregada COUNT.** Per trobar quantes vendes de productes ha efectuat cada client, agruparem les files de la taula de vendes pels valors de la columna client i utilitzarem la funció COUNT() per obtenir el n° de vendes registrades:

```
1 SELECT client, COUNT(datahora) AS "Num vendes"
2 FROM vendes
3 GROUP BY client;
```

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **3) Exemple de GROUP BY amb la funció agregada COUNT**
- Al veure els resultats podem apreciar que si sumem el total de vendes per client obtindrem 28. És a dir, el total de files de la taula vendes.
- En altres paraules: tenim registrades 28 vendes, però tenim 3 clients que no han comprat encara i dels 17 client que sí que ho han fet, alguns han comprat una vegada, d'altres un parell, o fins i tot tres vegades.

	client character varying (10)	Num vendes bigint
1	C13	2
2	C01	1
3	C09	2
4	C12	1
5	C03	1
6	C07	1
7	C16	2
8	C14	2
9	C19	3
10	C15	1
11	C06	1
12	C05	1
13	C08	2
14	C10	1
15	C02	3
16	C20	1
17	C11	3

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **4) Exemple de GROUP BY amb columnes de tipus data (“fecha”).** En el següent exemple volem agrupar per una columna que és de tipus DATE, és a dir, marca de temps.
- Per agrupar les vendes per dates, utilitzem la funció DATE () per convertir primer les marques de temps en dates i després agrupar per la data del resultat.

Per exemple:




```
SELECT DATE(datahora) data_venda, SUM(unitats)
FROM vendes
GROUP BY DATE(datahora);
```

← és el seu
àlies

1.1. GROUP BY. EXEMPLES (BASE DE DADES BOTIGA)

- **5) Exemple de GROUP BY amb múltiples columnes.** En aquest exemple, la clàusula GROUP BY divideix les files de la taula de clients pels valors de les columnes ciutat i adreça per comprovar quantes persones viuen al mateix domicili. Per a cada grup de (ciutat, adreça), el COUNT () calcula el total de clients.

```
SELECT ciutat, adreca, COUNT(codi)
FROM clients
GROUP BY ciutat, adreca
ORDER BY ciutat;
```

	 ciutat character varying (25)	 adreca character varying (100)	 count bigint
1	Rubí	C/ Major s/n	12
2	Sabadell	C/ Major s/n	2
3	Sant Cugat	C/ Major s/n	3
4	Terrassa	C/ Major s/n	3

1.2. FILTRAT DE DADES AGRUPADES

- També podem filtrar dades agrupades. És possible limitar la consulta amb **WHERE** i amb **HAVING**.
- Si inclou una clàusula **WHERE**, només s'agrupen els registres que compleixen les condicions. Per exemple:

```
select editorial, count(*)  
from libros  
where precio<30  
group by editorial;
```

Comptarem i agruparem per editorial considerant només els llibres el preu dels quals siga menor a 30€.

1.2. FILTRAT DE DADES AGRUPADES

- La clàusula **HAVING** s'utilitza sovint amb la clàusula **GROUP BY** per filtrar grups o agregats en funció d'una condició especificada. La clàusula **HAVING** especifica una **condició** de cerca **per a un grup o una funció agregada**.
- **HAVING i WHERE fan el mateix???** No exactament. La clàusula **WHERE** s'aplica a les files mentre que la clàusula **HAVING** s'aplica als grups de files.

1.2. FILTRAT DE DADES AGRUPADES

- Anem a entendre com funciona **HAVING**. La sintaxi bàsica de la clàusula **HAVING**:

SELECT columna1, funcio_agregada (columna2)

FROM taula

GROUP BY columna1

HAVING condicio ;

- En aquesta sintaxi, **GROUP BY** retorna files agrupades per la columna1. La clàusula **HAVING** especifica una condició per filtrar els grups.
- **No podem utilitzar àlies de columna a la clàusula HAVING.** 

1.2. FILTRAT DE DADES AGRUPADES. EXEMPLES(BD BOTIGA)

- **1) Exemple de HAVING amb funció COUNT().** La següent consulta utilitza la clàusula GROUP BY per trobar el nombre de clients per ciutat:

```
SELECT ciutat, COUNT(codi)
FROM clients
GROUP BY ciutat;
```

	ciutat character varying (25)	count bigint
1	Terrassa	3
2	Rubí	12
3	Sant Cugat	3
4	Sabadell	2

1.2. FILTRAT DE DADES AGRUPADES. EXEMPLES(BD BOTIGA)

- **1) Exemple de HAVING amb funció COUNT().** Ara afegim la clàusula HAVING per seleccionar la ciutat que té més de 10 clients

```
SELECT ciutat, COUNT(codi)
FROM clients
GROUP BY ciutat
HAVING COUNT (codi) > 10;
```

	ciutat character varying (25)	count bigint
1	Rubí	12

1.2. FILTRAT DE DADES AGRUPADES. EXEMPLES(BD BOTIGA)

- **2) Exemple de HAVING amb funció SUM().** La següent consulta utilitza la clàusula GROUP BY amb la funció SUM () per trobar la quantitat total d'unitats venudes de cada producte:

```
SELECT producte, SUM(unitats)
FROM vendes
GROUP BY producte;
```

	producte character varying (10)	sum bigint
1	P20	29
2	P03	22
3	P09	52
4	P14	3
5	P12	23
6	P02	4
7	P15	22
8	P04	30
9	P16	10
10	P17	22
11	P05	6
12	P11	36
13	P18	10
14	P10	15
15	P19	19
16	P07	18
17	P06	16

1.2. FILTRAT DE DADES AGRUPADES. EXEMPLES(BD BOTIGA)

- **2) Exemple de HAVING amb funció SUM().** Ara afegim la clàusula HAVING per seleccionar els productes que s'han venut més de 20 unitats:

```
SELECT producte, SUM(unitats)
FROM vendes
GROUP BY producte
HAVING SUM(unitats)>20;
```

	producte character varying (10)	sum bigint
1	P20	29
2	P03	22
3	P09	52
4	P12	23
5	P15	22
6	P04	30
7	P17	22
8	P11	36

1.3. REGLES PER TREBALLAR AMB AGRUPAMENTS



- Per treballar amb l'agrupament és convenient entendre bé el problema i el que es demana i aplicar 3 regles mnemotècniques:
- **Regla d'Or:** Tot el que apareix en el **SELECT** i en el **HAVING** o bé són funcions d'agrupament (o agregades) o bé estan en el **GROUP BY**.
 - Exemple: **SELECT** codiclient, **AVG**(descompte), iva **FROM** factures
 - **GROUP BY** codiclient;
- **Regla de Plata:** Les funcions de columna i de grup (funcions que resumeixen tota una columna en un únic valor, com: avg, count, sum, max,...) no poden aparèixer en el **WHERE**.
- **Regla de Bronze:** Normalment no cal agrupar quan s'ha de tornar una única dada.
- **Regla 4^a:** Normalment no cal agrupar si després (en les clàusules **HAVING** o **SELECT**) no s'utilitzen funcions d'agrupament.