Tutoriat 3

Group by; Having;

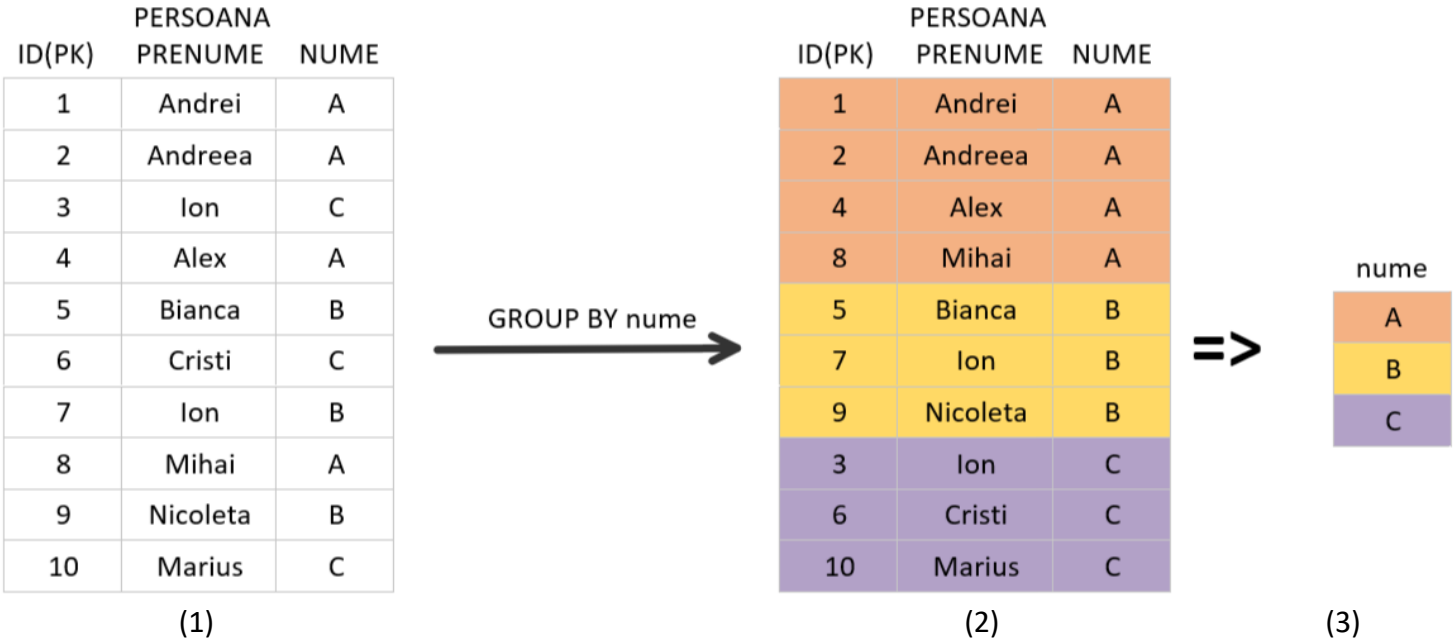
Group by

GROUP BY este clauza de grupare a mai multor linii dintr-un tabel in functie de valorile de pe una sau mai multe coloane specificate. Acest lucru ne ajuta sa impartim informatia dintr-un tabel in mai multe categorii reprezentate de valorile distincte de pe o coloana specificata.

Fie urmatorul tabel in care se afla membrii mai multor familii(2 persoane sunt din aceasi familie daca au nume identic).

ID(PK)	PERSOANA PRENUME	NUME
1	Andrei	A
2	Andreea	A
3	Ion	C
4	Alex	A
5	Bianca	B
6	Cristi	C
7	Ion	B
8	Mihai	A
9	Nicoleta	B
10	Marius	C

Sa vedem ce se intampla cand grupam datele din tabel dupa coloana NUME:



ID(PK)	PERSOANA PRENUME	NUME
1	Andrei	A
2	Andreea	A
3	Ion	C
4	Alex	A
5	Bianca	B
6	Cristi	C
7	Ion	B
8	Mihai	A
9	Nicoleta	B
10	Marius	C

(1)

GROUP BY nume

ID(PK)	PERSOANA PRENUME	NUME
1	Andrei	A
2	Andreea	A
4	Alex	A
8	Mihai	A
5	Bianca	B
7	Ion	B
9	Nicoleta	B
3	Ion	C
6	Cristi	C
10	Marius	C

(2)

=>

nume
A
B
C

(3)

Tabelul final va fi tabelul (3), insa, o vizualizare mai buna a GROUP BY-ului este tabelul (2). Toate liniile de aceasi culoare in tabelul (2) reprezinta numai o linie in tabelul (3). GROUP BY-ul a luat toate liniile cu acelasi nume si le considera ca fiind o singura linie, astfel valorile coloanelor ID si PRENUME sunt incerte deoarece pot avea mai multe valori(de exemplu pentru numele A prenumele poate fi: Andrei, Andreea, Alex sau Mihai), ne putem gandi la ele ca la un fel de „valori multiple”.

Tocmai din cauza acestor „valori multiple” exista anumite restrictii pe care GROUP BY-ul ni le impune:

- O data ce am facut GROUP BY dupa o coloana eu voi putea selecta numai valorile din acea coloana in SELECT, a selecta alte valori este imposibil deoarece compilatorul nu stie pe care sa o afiseze. Toate celelalte coloane au „valori multiple” deci nu se va sti ce valoare sa fie afisata.
- O coloana ce nu se afla in GROUP BY poate fi folosita numai intr-o functie de agregare. Acest lucru este posibil deoarece functiile de agregare returneaza mereu valori unice ce pot fi atasate fiecarei linii din tabelul final, astfel eliminanduse problema „valorilor multiple”. Functiile de agregare sunt :
    - AVG(x) – media valorilor de pe coloana x.
    - SUM(x) – suma valorilor de pe coloana x.
    - MAX(x) – valoarea maxima de pe coloana x.

- MIN(x) – valoarea minima de pe coloana x.
- COUNT(*) – numarul de „valori multiple”, inclusiv NULL.
- COUNT([DISTINCT] expr) – numarul de „valori multiple” [DISTINCTE] care sunt egale cu expr.
- VARIANCE(x) – dispersia valorilor coloanei x.
- STDDEV(x) – abaterea standard a valorilor coloanei x.

In continuare voi atasa tabelei PERSOANA o coloana noua VARSTA:

PERSOANA

ID(PK)	PRENUME	NUME	VARSTA
1	Andrei	A	15
2	Andreea	A	20
3	Ion	C	45
4	Alex	A	16
5	Bianca	B	20
6	Cristi	C	18
7	Ion	B	32
8	Mihai	A	41
9	Nicoleta	B	57
10	Marius	C	21

Si acum sa folosim functiile agregate pentru a afla varsta minima, maxima, media varstelor si numarul de membrii din fiecare familie:

```
1. SELECT nume, MIN(varsta), MAX(varsta), AVG(varsta), COUNT(prenume)
2. FROM persoana
3. GROUP BY nume;
```

nume	MIN(varsta)	MAX(varsta)	AVG(varsta)	COUNT(PRENUME)
A	15	41	23	4
B	20	57	71	3
C	18	45	28	3

In cazul in care in GROUP BY se alfa mai multe coloane atunci se va face prima data grupare dupa prima coloana, dupa aceea, pentru fiecare linie rezultata se va face GROUP BY pentru a doua coloana, …etc.

Having

Clauza HAVING este folosita pentru a filtra liniile folosind functii agregate dupa o clauza GROUP BY. Ea are rolul lui WHERE dar pentru functii agregate.

Clauza HAVING este folosita deoarece, in timp ce in WHERE pot sa filtrez informatia la nivel de linie, in HAVING pot filtra informatia la nivel de linie agregata, astfel permitandu-ne sa folosim functiile de agragare intr-un mod practic.

Sa luam ca exemplu tabela EMPLOYEES si sa afisam toate id-urile departamentelor si numarul de angajati ce lucreaza in ele pentru departamentele ce au mai mult de 20 de angajati:

```
1.  SELECT department_id, COUNT(employee_id)
2.  FROM employees
3.  GROUP BY department_id
4.    HAVING COUNT(employee_id) > 20;
```

Department_id	COUNT(employee_id)
50	45
80	34

Am folosit GROUP BY pentru a grupa dupa DEPARTMENT_ID si apoi am numarat in HAVING cati angajati se afla in fiecare departament pastrand numai departamentele cu mai mult de 20 de angajati.

Alte detalii despre GROUP BY si HAVING:

- HAVING se poate folosi de si fara GROUP BY insa atunci el va verifica conditia pe tot tabelul.
- Functiile de agregare se pot folosi numai in SELECT, ORDER BY si HAVING.
- Functiile MAX si MIN pot fi folosite atat pentru numere cat si pentru caractere sau date calendaristice.