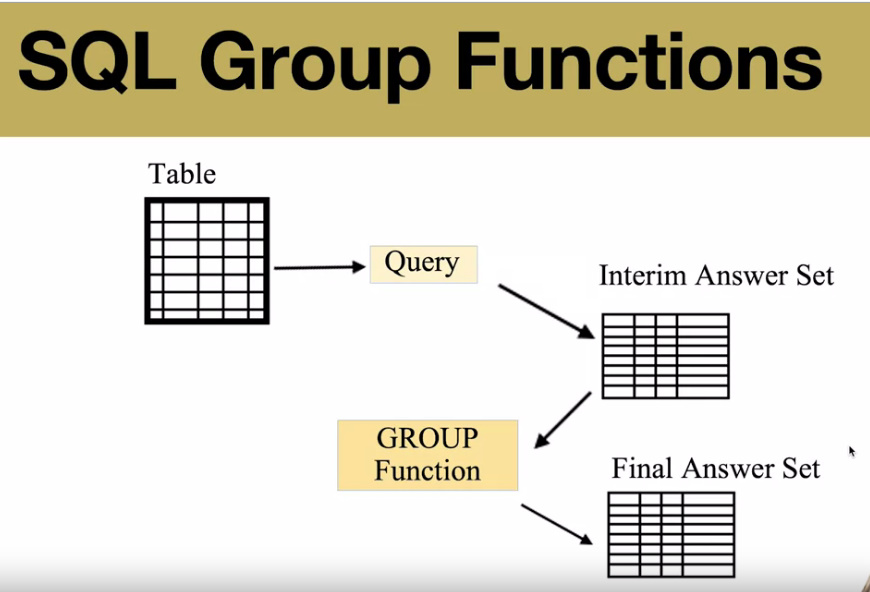
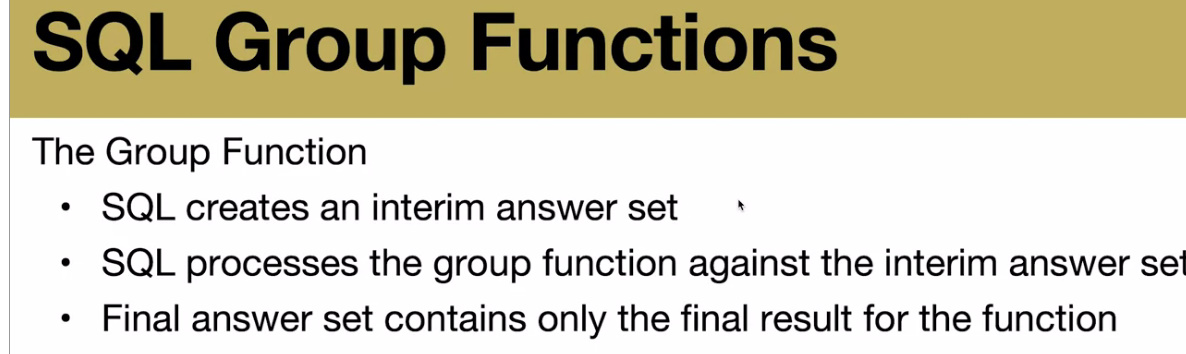
* In Group By, SQL creaza intai un tabel, ce nu il vedem, cu toate tuples din tabel, apoi pe baza la GROUP BY grupeaza valorile si ne returneaza rezultatul final:



* 

**SQL Group function**

* SUM – aduna valorile non null. Lucreaza doar cu numere
* AVG – afla media valorilor non null. Lucreaza doar cu numere.
* COUNT – calculeaza cate linii au o non null valoare. Lucreaza cu orice tip de date. Poate fi folosit \* pentru a calcula pur si simplu cate linii sunt

Putem pune si COUNT(DISTINCT Attribute) pentru a calcula valoile unice doar

* MIN – minim non null value. Lucreaza cu orice tip de date. Daca e text, returneaza cel mai scurt string
* MAX – maxim non null value. Lucreaza cu orice tip de date. Daca e text, returneaza cel mai lung string

**Toate ignora NULL**

**Round**

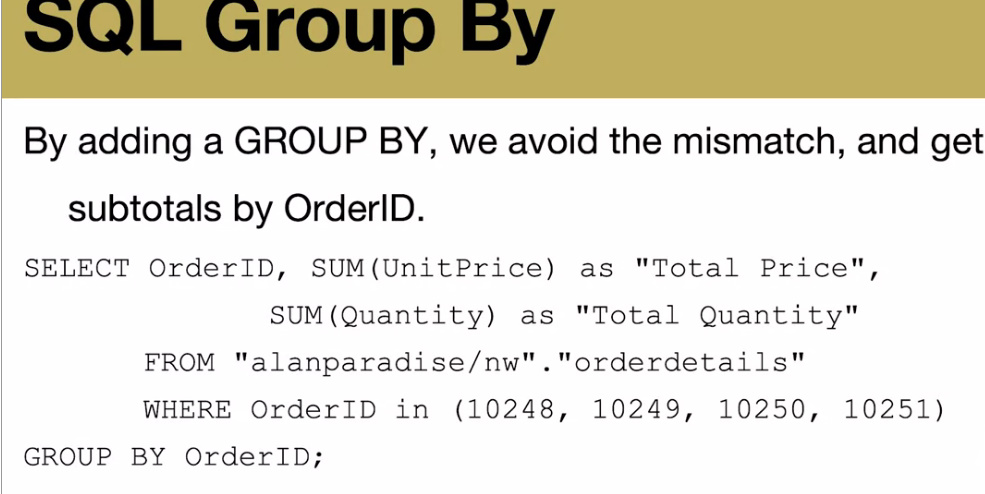
* AVG() returneaza adesea un double cu foarte multe cifre dupa virgula. Dar, ProstgreSQL ofera o functie de a il **aproxima**:
* **TO\_CHAR(AVG(),’999.999’)** – folosim cifre de 9 pentru a arata formatul. Deci, aici spunem ca vrem 4 cifre pana la virgula si 4 dupa, dar putem pune oricate, important sa fie de 9

**Having vs Where cu Group**

* **Where** – acesta va influenta datele din interim answer, adica sa nu uitam ca cand folosim Group, query intai va selecta tuples din tabel, creand un tabel virtual, si abea apoi va da acest tabel instructiunii Group ca sa grupeze si sa creeze un alt tabel final, pe care sa il vedem. Ei bine, WHERE va influenta anume tabelul virtual, adica el va decide ce date sa fie selectate din baza de date ca sa fie date apoi lui Group
* **Having** – el decide care date vor fi afisate din tabelul final in care deja s-au facut gruparile cu Group.

**Group by**

* Group By creaza subtotaluri
* Aceasta ii spune lui SQL sa execute group function asupra la o submultime de linii din tabelul interimar(cel ce nu il vedem inca), si sa ofere un total pentru fiecare submultime.
* GROUP BY – el ii spune lui SQL care atribut anume e cel dupa care trebuie sa se faca gruparea. El se va asigura ca pentru fiecare valoare unica a unui atribut pus, sa se creeze o submultime



Deci, aici aratam ca se vor crea submultimi, pe baza la valorile lui OrderID, unice evident, si se va face suma pentru fiecare ID din OrderID care se gaseste prin tabel. Putem avea oricate functii de grupare, important sa nu fie mai multe atribute de cat avem in GROUP BY, caci daca punem mai multe atribute, dar la GROUP BY doar unul, nu e clar ce valoare sa se afiseze pentru celalalt, asa cum exista multe tupluri cu valori diferite.

* **Orice atribut din SELECT trebuie sa fie o group function sau un atribut pus in GROUP BY. ASTA E REGULA!!!**

**!!!!Atentie, cand definim o noua coloana, cu AS, unde folosim functii de group, nu vom folosi in Having numele acelei coloane, ci vom folosi functia:**

SELECT employeeid, count(orderid) AS "Orders"

          FROM "alanparadise/nw"."orders"



       GROUP BY employeeid

       HAVING count(orderid) > 100

       ORDER BY 2 desc;



SELECT employeeid, count(orderid) AS "Orders"

          FROM "alanparadise/nw"."orders"

       GROUP BY employeeid

       HAVING "Orders" > 100



       ORDER BY 2 desc;