Agregaciones

COMO TE VES EN 10 AÑOS???

Aggregations in SQL

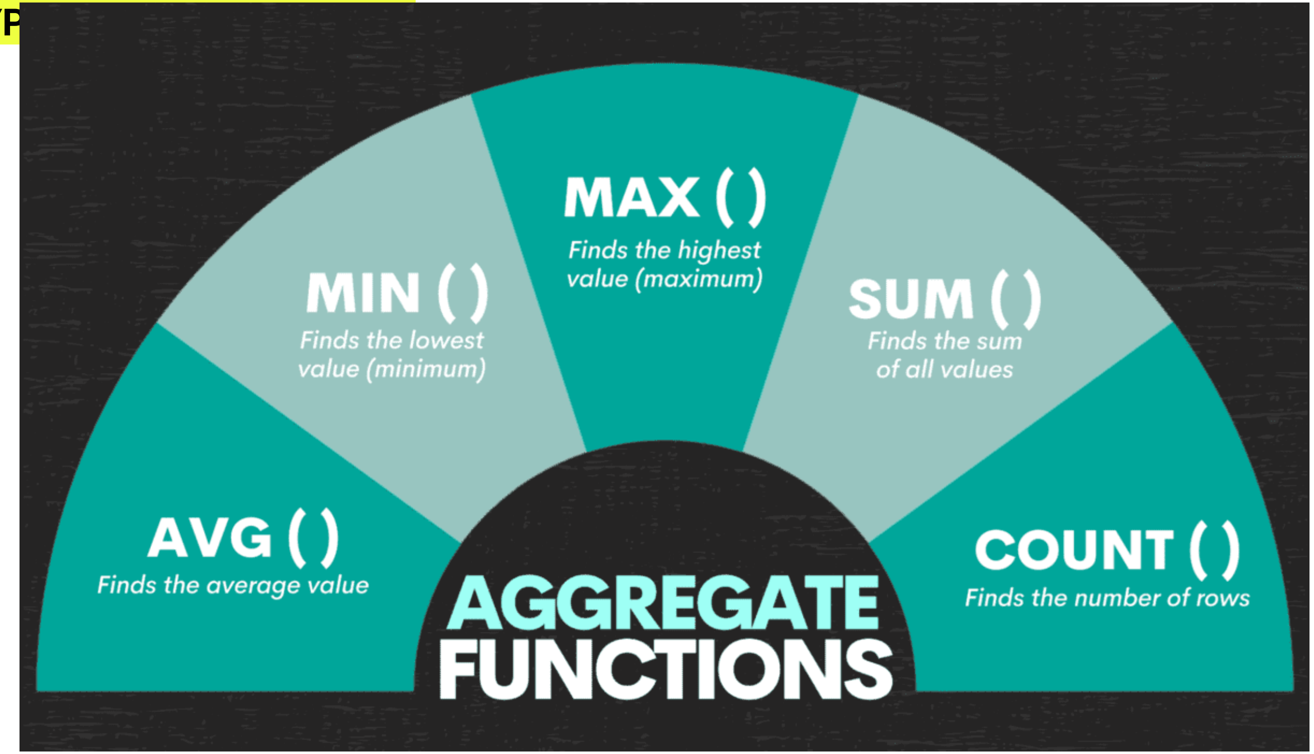
DATA ANALYTICS | IRONHACK

AGGREGATION FUNCTIONS AVAILABLE

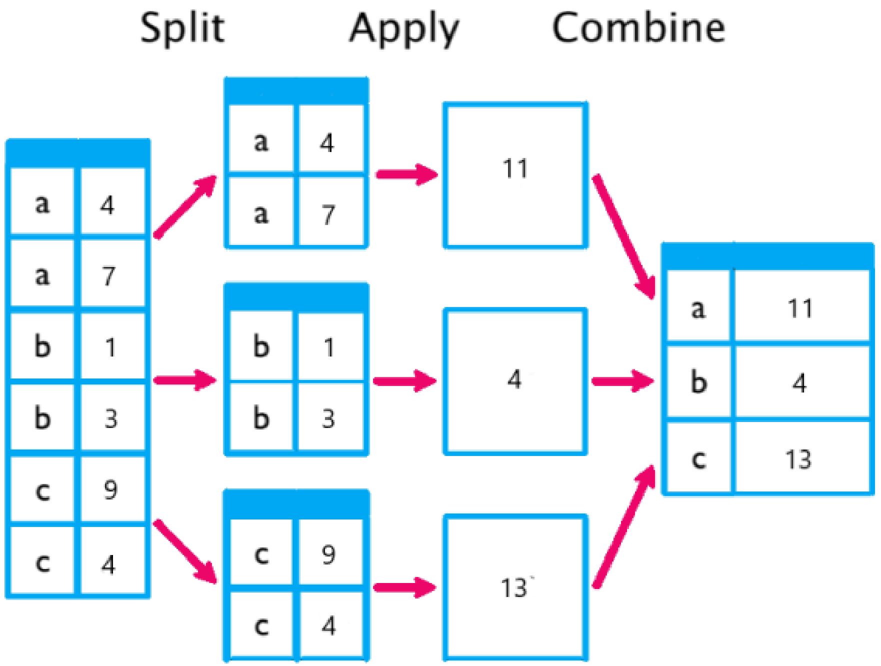
- The functions available are:

    - sum()
    - min()
    - max()
    - avg()
    - count()

TYP


MAX ( )
Finds the highest value (maximum)
MIN ( )
Finds the lowest value (minimum)
SUM ( )
Finds the sum of all values
AVG ( )
Finds the average value
COUNT ( )
Finds the number of rows
AGGREGATE FUNCTIONS

WHAT IS AN AGGREGATION???

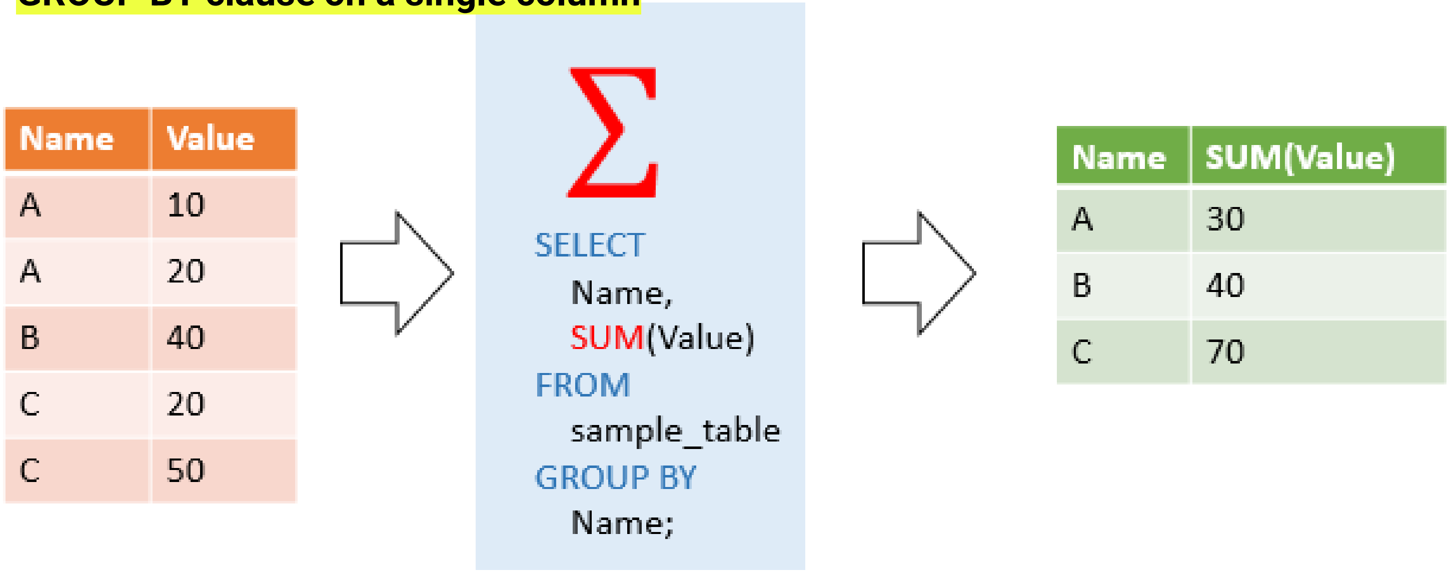


WHAT IS AGGREGATION?

- Is the process of computing something for a given column (ie)

  - By gender
  - By Age

GROUP BY clause on a single column

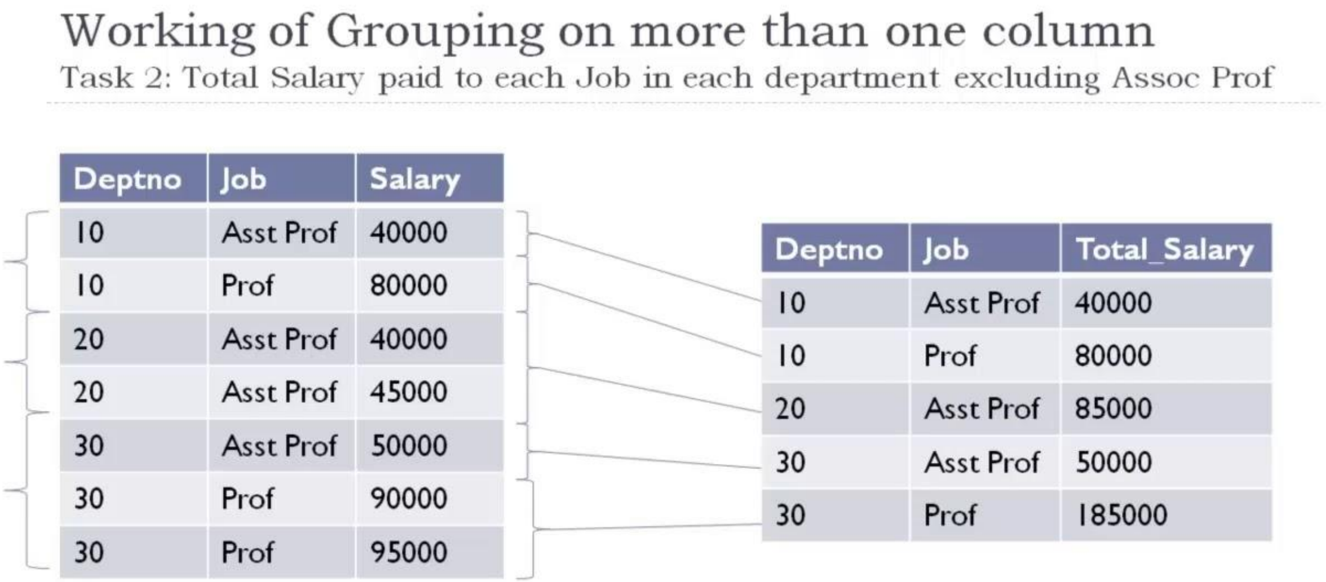


Name	Value
A	10
A	20
B	40
C	20
C	50

```
SELECT
    Name,
    SUM(Value)
FROM
    sample_table
GROUP BY
    Name;
```

Name	SUM(Value)
A	30
B	40
C	70

GROUP BY clause on multiple columns



Working of Grouping on more than one column

Task 2: Total Salary paid to each Job in each department excluding Assoc Prof

Deptno	Job	Salary
10	Asst Prof	40000
10	Prof	80000
20	Asst Prof	40000
20	Asst Prof	45000
30	Asst Prof	50000
30	Prof	90000
30	Prof	95000

Deptno	Job	Total_Salary
10	Asst Prof	40000
10	Prof	80000
20	Asst Prof	85000
30	Asst Prof	50000
30	Prof	185000

AGGREGATING VALUES

- To compute aggregated values by group the syntax is:

  SELECT function(column) FROM db.table
  GROUP BY column;

```
select avg(amount) from bank.loan
group by status;

select avg(amount) as Average, status from bank.loan
group by status
order by Average asc;
```

GROUP BY MORE THAN ONE COLUMN

- We can make aggregations according to several columns (ie):

```
SELECT * FROM db.table_name
GROUP BY col1, col2, col3,...
```

EXAMPLES

```sql
select round(avg(amount),2) - round(avg(payments),2) as "Avg Balance", status, duration
from bank.loan
group by status, duration
order by status, duration;
```

```sql
select round(avg(amount),2) - round(avg(payments),2) as "Avg Balance", status, duration
from bank.loan
group by status, duration
order by duration, status;
```

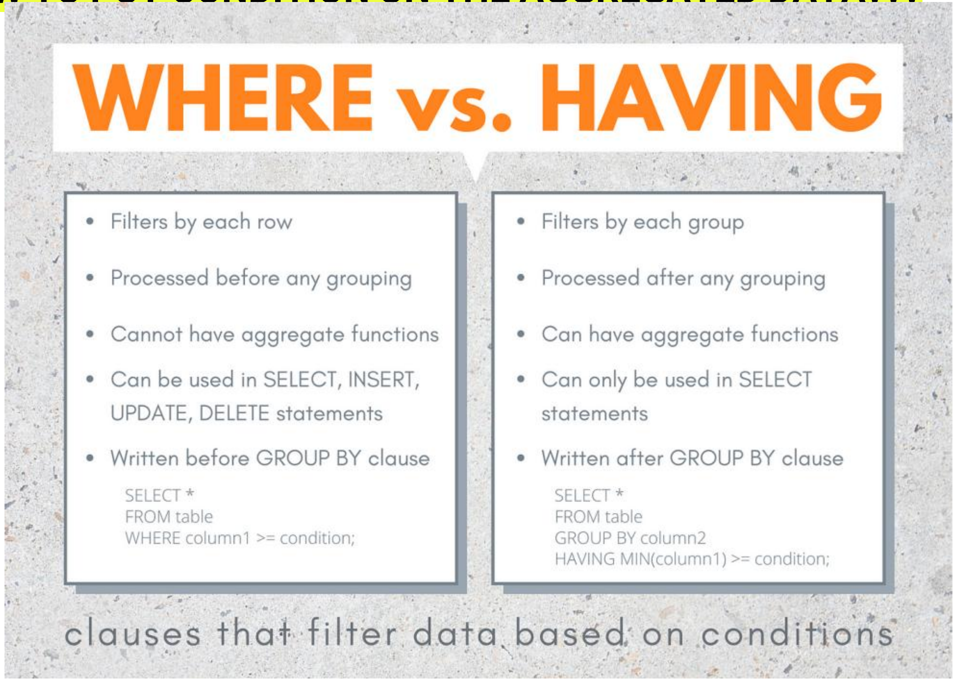
```sql
select type, operation, k_symbol, round(avg(balance),2)
from bank.trans
group by type, operation, k_symbol;
```

```sql
select type, operation, k_symbol, round(avg(balance),2)
from bank.trans
group by type, operation, k_symbol
order by type, operation, k_symbol;
```

WHERE vs HAVING

DATA ANALYTICS | IRONHACK

HOW TO PUT CONDITION ON THE AGGREGATED DATA???



WHERE vs. HAVING

- Filters by each row
- Processed before any grouping
- Cannot have aggregate functions
- Can be used in SELECT, INSERT, UPDATE, DELETE statements
- Written before GROUP BY clause

```
SELECT *
FROM table
WHERE column1 >= condition;
```

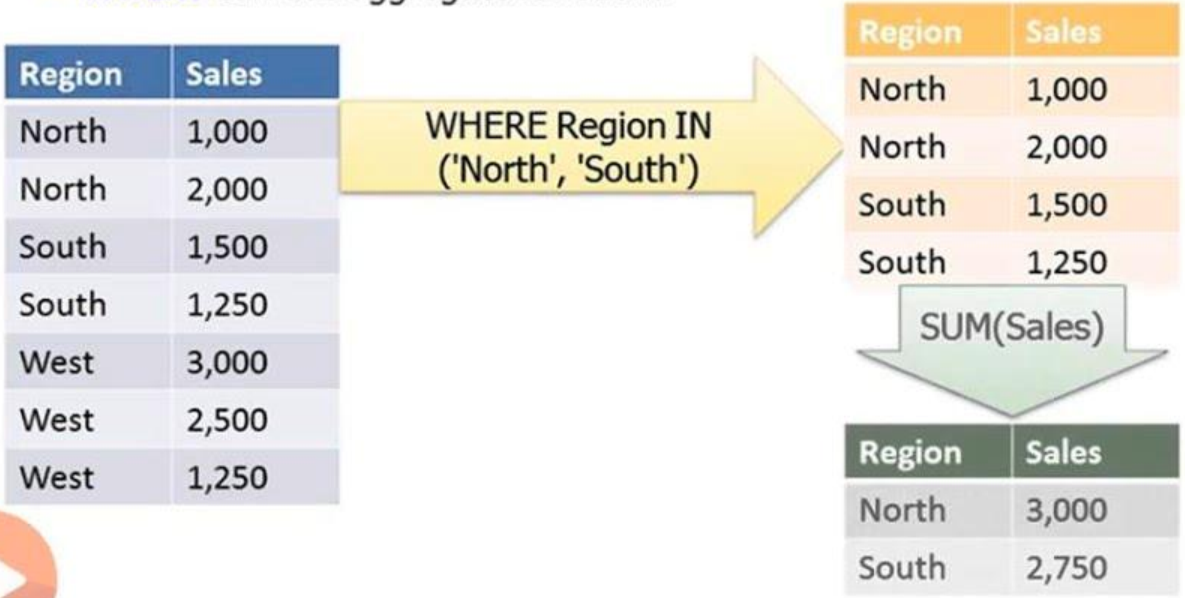
- Filters by each group
- Processed after any grouping
- Can have aggregate functions
- Can only be used in SELECT statements
- Written after GROUP BY clause

```
SELECT *
FROM table
GROUP BY column2
HAVING MIN(column1) >= condition;
```

clauses that filter data based on conditions

HAVING vs. WHERE

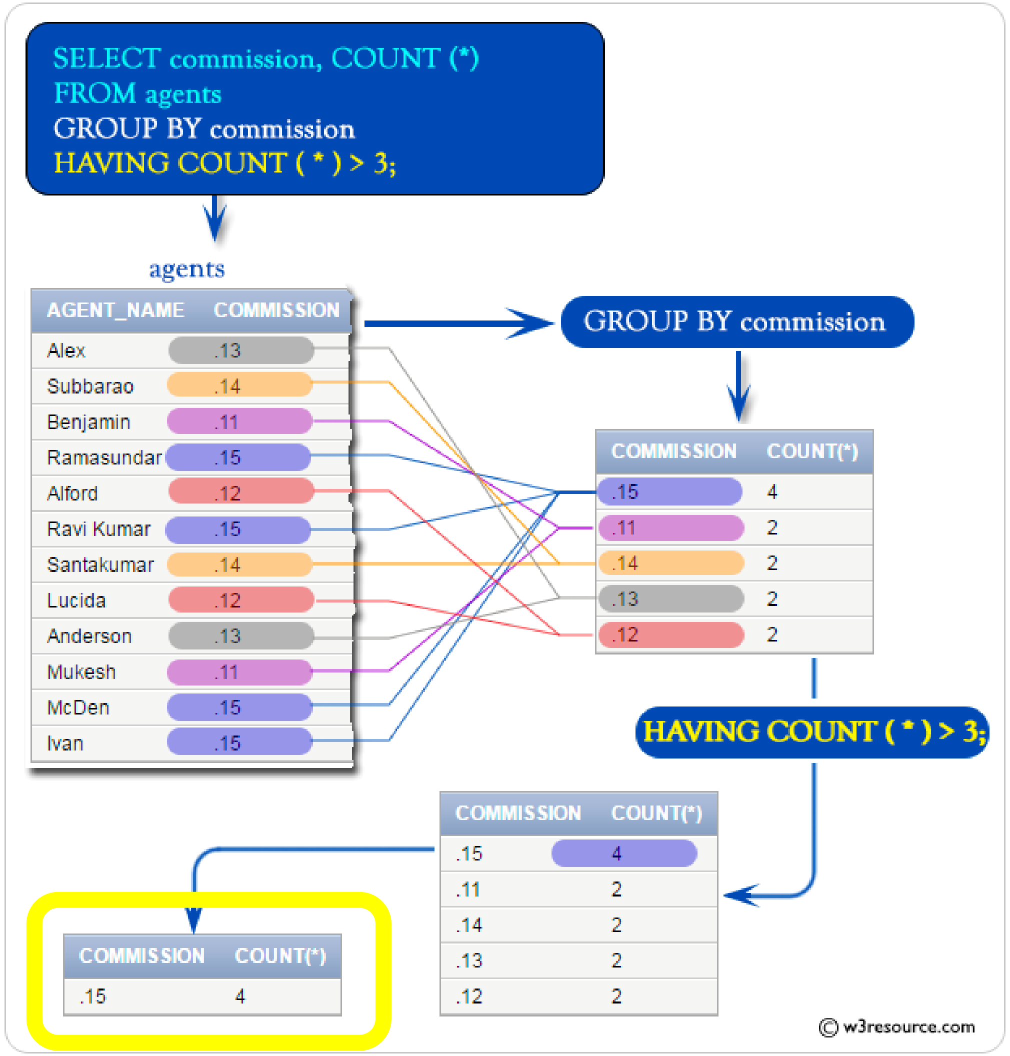
- `WHERE` filters rows
- `HAVING` filters groups
  - `HAVING` can use aggregate functions



Region	Sales
North	1,000
North	2,000
South	1,500
South	1,250
West	3,000
West	2,500
West	1,250

Region	Sales
North	1,000
North	2,000
South	1,500
South	1,250

Region	Sales
North	3,000
South	2,750



```
SELECT commission, COUNT (*)
FROM agents
GROUP BY commission
HAVING COUNT ( * ) > 3;
```

AGENT_NAME	COMMISSION
Alex	.13
Subbarao	.14
Benjamin	.11
Ramasundar	.15
Alford	.12
Ravi Kumar	.15
Santakumar	.14
Lucida	.12
Anderson	.13
Mukesh	.11
McDen	.15
Ivan	.15

COMMISSION	COUNT(*)
.15	4
.11	2
.14	2
.13	2
.12	2

COMMISSION	COUNT(*)
.15	4
.11	2
.14	2
.13	2
.12	2

COMMISSION	COUNT(*)
.15	4

WHERE CLAUSE

- As you know, this clause has to be used **before** the aggregation with GROUP BY

```
select type, operation, k_symbol, round(avg(balance),2) as Average
from bank.trans
where k_symbol <> '' and k_symbol <> ' ' and  operation <> ''
group by type, operation, k_symbol
```

HAVING | What? Why?

- HAVING clause is equivalent to WHERE **BUT** to check conditions **AFTER** GROUP BY.

```
select type, operation, k_symbol, round(avg(balance),2) as Average
from bank.trans
where k_symbol <> '' and k_symbol <> ' ' and  operation <> ''
group by type, operation, k_symbol
having Average > 30000
order by type, operation, k_symbol;
```

HAVING | What? Why?

- HAVING clause is equivalent to WHERE **BUT** to check conditions **AFTER** GROUP BY.

```
select type, operation, k_symbol, round(avg(balance),2) as Average
from bank.trans
where k_symbol <> '' and k_symbol <> ' ' and  operation <> ''
group by type, operation, k_symbol
having Average > 30000
order by type, operation, k_symbol;
```

```
select round(avg(amount),2) - round(avg(payments),2) as Avg_Balance, status,
duration
from bank.loan
group by status, duration
having Avg_Balance > 100000
order by duration, status;
```

IRON
HACK

THANKS !