# Agregações:

## GROUP BY

Se não colocado, o ‘grupo’ são todos os registros

SELECT COUNT(\*)

 ,COUNT(v)

 ,COUNT(N2)

 ,COUNT(DISTINCT v)

 FROM N

SELECT N1

 ,COUNT(\*)

 ,COUNT(v)

 ,COUNT(N2)

 ,COUNT(DISTINCT v)

 FROM N

 GROUP BY N1

Funções ‘normais’:

COUNT

AVG

MAX

MIN
SUM

STRING\_AGG

SELECT S\_SEXO\_PESSOA

 ,COUNT(\*)

 ,STRING\_AGG(N\_PESSOA\_PESSOA,',')

FROM PESSOA

WHERE N\_PESSOA\_PESSOA<1000

GROUP BY S\_SEXO\_PESSOA

CHECKSUM\_AGG

CHECKSUM\_AGG(BINARY\_CHECKSUM(\*))

## WHERE / HAVING

WHERE = antes da agregação -> filtra registros de entrada

HAVING = depois da agregação -> filtra registros de saída

## OVER

SELECT Cod

 ,N1

 ,v

 ,SUM(v) OVER (ORDER BY Cod) AS sv

 ,SUM(v) OVER (PARTITION BY N1 ORDER BY Cod) AS svp

 ,SUM(v) OVER (PARTITION BY N1) AS svp2

 ,RANK() OVER (ORDER BY Cod) AS r

 ,ROW\_NUMBER() OVER (ORDER BY Cod) AS rn

 ,COUNT(\*) OVER (ORDER BY Cod) AS c

 ,COUNT(\*) OVER (PARTITION BY N1) AS qtd\_p

 ,RANK() OVER (PARTITION BY N1 ORDER BY Cod) AS cp

 ,RANK() OVER (PARTITION BY N1 ORDER BY Cod) \* 100 / COUNT(\*) OVER (PARTITION BY N1) AS porcentagem\_na\_p

 FROM N

-- ORDER BY Cod desc

* [PARTITION BY](https://learn.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql?view=sql-server-ver16#partition-by) that divides the query result set into partitions.
* [ORDER BY](https://learn.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql?view=sql-server-ver16#order-by) that defines the logical order of the rows within each partition of the result set.
* [ROWS/RANGE](https://learn.microsoft.com/en-us/sql/t-sql/queries/select-over-clause-transact-sql?view=sql-server-ver16#rows-or-range) that limits the rows within the partition by specifying start and end points within the partition. It requires ORDER BY argument and the default value is from the start of partition to the current element if the ORDER BY argument is specified.

SELECT Cod

 ,N1

 ,v

 ,LAST\_VALUE(v) OVER (PARTITION BY N1 ORDER BY Cod)

 ,LAST\_VALUE(v) OVER (PARTITION BY N1 ORDER BY Cod RANGE BETWEEN CURRENT ROW AND UNBOUNDED FOLLOWING)

 FROM N

 ORDER BY Cod

## LAG/LEAD

SELECT AnoMes

 ,COALESCE(LAG(IPCA\_Acu) OVER (ORDER BY AnoMes),1)\*(1+(IPCA/100)) AS IPCA\_Acu

 FROM Z\_IPREJUN\_Indices i

 WHERE IPCA IS NOT NULL