

# SQL – Funções Agregadas

- São funções que tomam uma coleção (um conjunto ou subconjunto) de valores como entrada, retornando um valor simples.
- **AVG** - Média (Average)
- **MIN** - Mínimo (Minimum)
- **MAX** - Máximo (Maximum)
- **SUM** - Total (Soma)
- **COUNT** - Contagem (Count)

# SQL – Funções Agregadas

- A entrada para SUM e AVG precisa ser um conjunto de números.

```
SELECT AVG (atributo_numerico)  
FROM tabela  
WHERE condição
```


```
SELECT SUM (atributo_numerico)  
FROM tabela  
WHERE condição
```

```
SELECT COUNT (*)  
FROM tabela
```

# SQL – Funções Agregadas

```
1  /*SOMA*/
2  SELECT SUM(vin_salario)
3  FROM vinculos;
4
5  /*MÉDIA*/
6  SELECT AVG(vin_salario)
7  FROM vinculos;
8
9  /*CONTADOR*/
10 SELECT COUNT(vin_salario)
11 FROM vinculos;
12
```

Conta a  
quantidade de  
tuplas (linhas)



# SQL – Funções Agregadas

- Podemos utilizar **funções agregadas** em um **grupo de conjuntos** de tuplas, utilizando **GROUP BY**

Tem que aparecer aqui

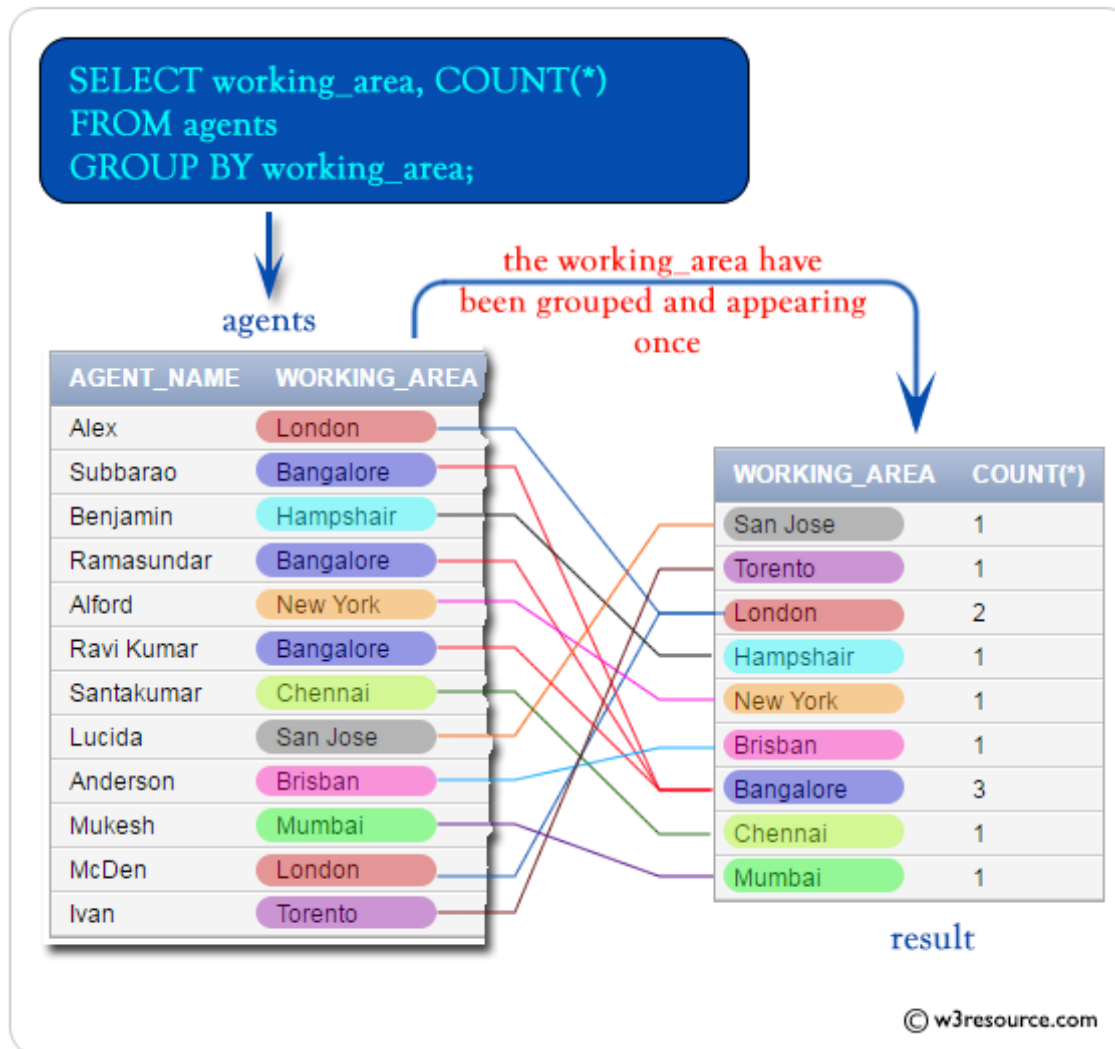
```
SELECT atributo1, AVG (atributo2)  
FROM tabela  
GROUP BY atributo1
```

Agrupar  
colunas da  
tabela

```
1  /*AGRUPA O SALÁRIO DOS MÉDICOS POR CLÍNICA*/  
2  SELECT c.cli_nome, SUM(vin_salario)  
3  FROM clinicas c JOIN vinculos v ON c.cli_iden = v.vin_cli_iden  
4  GROUP BY c.cli_nome;
```

- Obs. Atributo apareceu no Select com função agregada deve-se usar o GROUP BY.

# SQL – Funções Agregadas



# SQL – Funções Agregadas

Conta

```
1 SELECT c.cat_nome, count(*)
2 FROM filmes f JOIN filmes_categorias fc ON fc.fca_fil_iden = f.fil_iden
3 JOIN categorias c ON c.cat_iden = fc.fca_cat_iden
4 GROUP BY c.cat_nome
```

Grupo

filme character varying (255)	categoria character varying (25)
Godzilla Resurgence	Ação
The Hobbit: An Unexpected Journey	Aventura
The Fast and the Furious	Ação
The Curious Case of Benjamin Button	Drama
X-Men: First Class	Ação
The Hunger Games: Mockingjay - Part 2	Aventura
The Sorcerers Apprentice	Ação
Poseidon	Ação

5

1

2

# SQL – Funções Agregadas

- Definir condições a grupos podemos utilizar a cláusula **HAVING**.
  - Predicados da cláusula **HAVING** são aplicadas depois da formação de grupos.

```
SELECT atributo1, AVG (atributo2)
FROM tabela
GROUP BY atributo1
HAVING AVG (atributo2) > 1200
```

**Obs.** Having são where dos grupos, ou seja, Restringir as linhas oriundas do Group By

<https://www.w3resource.com/sql/aggregate-functions/Max-having.php>

```
SELECT cust_city, cust_country,
MAX(outstanding_amt)
FROM customer
GROUP BY cust_country, cust_city
HAVING MAX(outstanding_amt)>10000;
```

CUST_CITY	CUST_COUNTRY	OUTSTAND
Bangalore	India	12000
London	UK	4000
New York	USA	6000
New York	USA	6000
Bangalore	India	8000
London	UK	6000
London	UK	11000
New York	USA	3000
Brisban	Australia	5000
Brisban	Australia	7000
Chennai	India	8000
Mumbai	India	11000
Chennai	India	9000

customer

```
SELECT cust_city, cust_country,
MAX(outstanding_amt)
FROM customer
GROUP BY cust_country, cust_city;
```

CUST_CITY	CUST_CO	MAX(OUTSTA
Bangalore	India	12000
Brisban	Australia	7000
Chennai	India	11000
Hampshair	UK	5000
London	UK	11000
Mumbai	India	12000
New York	USA	6000
San Jose	USA	3000
Toronto	Canada	11000

```
HAVING MAX(outstanding_amt)>10000;
```

CUST_CITY	CUST_CO	MAX(OUTSTA
Bangalore	India	12000
Brisban	Australia	7000
Chennai	India	11000
Hampshair	UK	5000
London	UK	11000
Mumbai	India	12000
New York	USA	6000
San Jose	USA	3000
Toronto	Canada	11000

CUST_CITY	CUST_COUNTRY	MAX(OUTSTA
Bangalore	India	12000
Chennai	India	11000
London	UK	11000
Mumbai	India	12000
Toronto	Canada	11000

<https://www.w3resource.com/sql/aggregate-functions/Max-having.php>



# SQL – Funções Agregadas

- Definir condições a grupos podemos utilizar a cláusula **HAVING**.
  - Predicados da cláusula **HAVING** são aplicadas depois da formação de grupos.

```
1  /*RETORNAR OS MÉDICOS QUE POSSUEM MAIS DE UM ACOMPANHAMENTO*/
2  SELECT count(*), m.med_nome
3  FROM medicos m JOIN acompanhamentos a ON m.med_iden = a.acom_med_iden
4         JOIN pacientes p ON a.acom_pac_iden = p.pac_iden
5  GROUP BY m.med_nome
6  HAVING COUNT(*) > 1
7
```

# SQL – Funções Agregadas

- Se uma cláusula **WHERE** e uma cláusula **HAVING** aparecem na mesma consulta, o predicado que aparece em **WHERE** é aplicado primeiro
- Tuplas que satisfazem a cláusula **WHERE** são, agrupados por meio da cláusula **GROUP BY**. A cláusula **HAVING** aplica ao grupo.

# SQL – Funções Agregadas

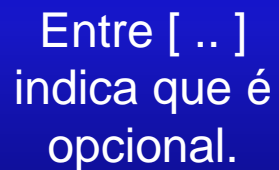
## Diferenças

- **Where:** seleciona as linhas antes dos grupos e funções agregadas sejam computadas (Ou seja, seleciona as linhas que poderão ser computadas)
- **Having:** Seleciona linhas do grupo após o cálculo das funções agregadas.

# SQL – LIMIT

Podemos selecionar a **quantidade exata de linhas** que vão ser retornadas a partir de um SELECT.

```
SELECT coluna1, coluna2,...  
FROM tabela  
[LIMIT { quantidade | ALL }]  
[OFFSET quantidade_pular_inicio]
```



Entre [ .. ]  
indica que é  
opcional.