

# SQL

## Estendida e funções de agregação

---

Carina F. Dorneles  
dorneles@inf.ufsc.br

# Funções de agregação de valores

- AVG()
    - Calcula a **média** dos valores **não nulos** de uma coluna. O domínio deve ser **numérico**
  - MAX() e MIN()
    - Retorna o valor **mais alto** e **mais baixo** (respectivamente) de uma coluna. Domínio deve ser **numérico**
  - SUM()
    - **Soma** os valores **não nulos** de uma coluna. O domínio deve ser numérico
  - COUNT()
    - **Conta** o número de linhas **não nulas** de uma coluna.
-

## Exemplos

```
SELECT avg(salario)
```

```
FROM pessoa
```

```
SELECT sum(salario)
```

```
FROM pessoa
```

```
SELECT min(salario)
```

```
FROM pessoa
```

```
SELECT max(salario)
```

```
FROM pessoa
```

```
SELECT count(salario)
```

```
FROM pessoa
```

```
SELECT count (distinct salario)
```

```
FROM  pessoa
```

### ***Pessoa***

<b><u>Codigo</u></b>	<b><i>Nome</i></b>	<b><i>Salário</i></b>
1	Ana	100
2	Paulo	200
3	João	NULL
4	Antônio	100

## Exemplos

SELECT avg(salario)

FROM pessoa →

<b><i>avg(salario)</i></b>
133.3333333

SELECT sum(salario)

FROM pessoa →

<b><i>sum(salario)</i></b>
400

SELECT min(salario)

FROM pessoa →

<b><i>min(salario)</i></b>
100

### ***Pessoa***

<b><u>Codigo</u></b>	<b><i>Nome</i></b>	<b><i>Salário</i></b>
1	Ana	100
2	Paulo	200
3	João	NULL
4	Antônio	100

SELECT max(salario)

FROM pessoa →

<b><i>max(salario)</i></b>
200

SELECT count(salario)

FROM pessoa →

<b><i>count(salario)</i></b>
3

SELECT count (distinct salario)

FROM pessoa →

<b><i>count(salario)</i></b>
2

## Exemplos – função de agregação

- Tabela consulta (data, hora, codPac, CodMed, CodConvenio)

- Obter a **quantidade** de consultas **já realizadas**

```
SELECT COUNT(*)
```

```
FROM consulta
```

- Obter a **quantidade** de consultas **que já tiveram convênio**

```
SELECT COUNT(codConv)
```

```
FROM consulta
```

---

# Exemplos – função de agregação

- Obter a **média de idade** dos paciente

```
SELECT AVG(idade)
```

```
FROM paciente
```

- Obter a **idade mínima** dos pacientes cadastrados

```
SELECT MIN(idade)
```

```
FROM paciente
```

- Obter o **valor total** das consultas entre os dias `2005-10-21` e `2005-12-21`

```
SELECT SUM(valor)
```

```
FROM consulta
```

```
WHERE data BETWEEN `2005-10-21` AND `2005-12-21`
```

---

## Exemplos – função de agregação

- Obter a **média de idade** dos paciente

```
SELECT AVG(idade)
```

```
FROM paciente
```

- Obter a **idade mínima** dos paciente

```
SELECT MIN(idade)
```

```
FROM paciente
```

- Obter o **valor total** das consultas entre os dias

```
SELECT SUM(valor)
```

```
FROM consulta
```

```
WHERE data BETWEEN '2005-10-21' AND '2005-12-21'
```

Observar que o **SELECT** possui apenas a função de agregação, sem nenhum atributo a mais... Quando há **atributo**, é diferente...

# SQL

**Estendida: ORDER BY, GROUP BY,  
HAVING**

---

Carina F. Dorneles  
dorneles@inf.ufsc.br



## Modelo estendido de consulta

SELECT <Lista de colunas, função agregação>

FROM <Lista de tabelas>

[WHERE <Condição>]

[GROUP BY <Lista de colunas>]

[HAVING <critério>]

[ORDER BY <Lista de colunas>]

---

## ORDER BY

SELECT <Lista de colunas, função agregação>

FROM <Lista de tabelas>

[WHERE <Condição>]

[ORDER BY <Lista de colunas>]

---

# ORDER BY

- Ordena as colunas por ordem crescente (*default*)

SELECT nome, email

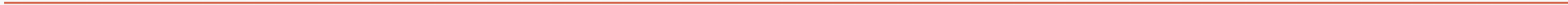
FROM medico

WHERE email LIKE '%gmail%'

ORDER BY nome **ASC**



*default!*



# ORDER BY - exemplos

```
SELECT nome, email  
FROM medico m  
WHERE email LIKE '%eee%'  
ORDER BY nome DESC /*decrescente*/
```

```
SELECT nome, fone, CRM  
FROM medico  
WHERE email LIKE '%eee%'  
ORDER BY m.dtanasc
```

- *default!*  
- não precisa estar no  
SELECT

# GROUP BY

SELECT <Lista de colunas, função agregação>

FROM <Lista de tabelas>

[WHERE <Condição>]

[GROUP BY <Lista de colunas>]

---

# GROUP BY

- Usado **junto** com as **funções de agregação** para **agrupar** o resultado por uma ou mais colunas

- Exemplo:

- Obter a **quantidade** de consultas

```
SELECT count(*)
```

```
FROM consulta
```

- Obter a **quantidade** de consultas **por data**

```
SELECT data, count(*)
```

```
FROM consulta
```

```
GROUP BY data
```

---

# GROUP BY

- Todos os atributos que acompanham a função de agregação devem estar no GROUP BY
  - Exemplo

```
SELECT data, hora, count(*)  
FROM consulta  
GROUP BY data, hora
```

---

# HAVING

SELECT <Lista de colunas, função agregação>

FROM <Lista de tabelas>

[WHERE <Condição>]

[GROUP BY <Lista de colunas>]

[HAVING <critério>]



Precisa do  
GROUP BY

---



# HAVING

- Aplicada sobre o agrupamento feito pelo GROUP BY
  - Usada porque no WHERE não é possível testar função de agregação
-

# HAVING - exemplo

- Obter a **quantidade** de consultas **por data**

```
SELECT data, count(*)
```

```
FROM consulta
```

```
GROUP BY DATA
```

- Obter a quantidade de consultas por data, desde que sejam mais de 10 consultas

```
SELECT data, count(*)
```

```
FROM consulta
```

```
GROUP BY DATA
```

```
HAVING count(*) >10
```

---

# HAVING - exemplo

- Obter a **quantidade** de consultas **por data**

```
SELECT data, count(*)
```

```
FROM consulta
```

```
GROUP BY DATA
```

- Obter a **quantidade** de consultas **por data**, **desde que** sejam mais de 10 consultas

```
SELECT data, count(*)
```

```
FROM consulta
```

```
GROUP BY DATA
```

```
HAVING count(*) >10
```

---

## → CUIDAR!!!!

- Sempre que houver necessidade de **fazer um teste nos valores** retornados pelas funções de agregação
    - usar **HAVING!**
    - **nunca** fazer no **WHERE**
-

## → CUIDAR!!!!

- Sempre que houver necessidade de **fazer um teste nos valores** retornados pelas funções de agregação
  - usar **HAVING**!
  - **nunca** fazer no **WHERE**



```
SELECT data, count(*)  
FROM consulta  
GROUP BY data  
HAVING count(*) > 10
```



```
SELECT data, count(^)  
FROM consulta  
WHERE count(*) > 10  
GROUP BY data
```