



# SQL Avançado

Fundamentos de Bancos de Dados  
Prof.<sup>a</sup> Lívia Almada

# Tópicos

- Comparação com valores NULL
- GROUP BY
- Funções de agregação
- HAVING

# Consultas de recuperação SQL mais complexas

- Recursos adicionais que permitem que os usuários especifiquem recuperações mais complexas do banco de dados:
  - Consultas aninhadas, tabelas de junção, junções externas, funções agregadas e agrupamento

# Comparações envolvendo NULL

- Significados de NULL
  - **Valor desconhecido**
  - **Valor indisponível ou retido**
  - **Atributo não aplicável**
- Cada valor NULL individual é considerado diferente de qualquer outro valor NULL

# Comparações envolvendo NULL

- A SQL permite consultas que verificam se o valor de um atributo é NULL
  - IS ou IS NOT NULL

**Consulta 18.** Recuperar os nomes de todos os funcionários que não possuem supervisores.

```
C18:      SELECT  Pnome, Unome
          FROM    FUNCIONARIO
          WHERE   Cpf_supervisor IS NULL;
```

# Funções de agregação em SQL

- São usadas para resumir informações de várias tuplas em uma síntese de tupla única.
- **Agrupamento**
  - Cria subgrupos de tuplas antes do resumo

# Funções de agregação em SQL

- Funções de agregação embutidas
  - **COUNT:** retorna o número de tuplas ou de valores
  - **SUM:** retorna a soma dos valores
  - **MAX:** retorna o valor máximo
  - **MIN:** retorna o valor mínimo
  - **AVG:** retorna a média dos valores

# Funções de agregação em SQL

## SINTAXE:

**nome-da-função (ALL | DISTINCT nome-coluna)|  
count(\*)**

- Essas funções podem ser usadas na cláusula SELECT ou em uma cláusula HAVING



# Funções de agregação em SQL

```
SELECT COUNT(*)  
FROM FUNCIONARIO;
```

- Valores NULL são descartados quando as funções de agregação são aplicadas a determinada coluna (atributo)

# Funções de agregação em SQL

Selecionar a soma dos salários, o valor máximo e mínimo e a média.

```
SELECT SUM(salario), MAX(salario),  
        MIN(salario), AVG(salario)  
FROM FUNCIONARIO;
```

# Funções de agregação em SQL

Quantos funcionários trabalham no departamento de Pesquisa?

```
SELECT COUNT(*)  
FROM FUNCIONARIO, DEPARTAMENTO  
WHERE drn = dnumero AND  
      dnome = 'PESQUISA'
```

# Funções de agregação em SQL

Quantos gerentes distintos existem?

```
SELECT COUNT(distinct cpfgerente)  
FROM FUNCIONARIO;
```

# Funções de agregação em SQL

Exercícios:

- 1- Encontre o número de empregados lotados no departamento de Informática ;
- 2 - Encontre o valor total da folha de pagamento;
- 3 - Encontre o valor do salário médio pago pela empresa.

# Agrupamento: **GROUP BY**

- **Particiona** a relação em subconjuntos de tuplas
  - Baseado no **atributo(s) de agrupamento**
  - Aplicar a função a cada grupo desse tipo independentemente
- Cláusula **GROUP BY**
  - Especifica os atributos de agrupamento

## Agrupamento: GROUP BY

```
SELECT cod_depto, Count(*), AVG (Salario)
FROM Professor
GROUP BY cod_depto;
```

# Agrupamento: GROUP BY

Pnome	Minicial	Unome	Cpf	...	Salario	Cpf_supervisor	Dnr		Dnr	Count (*)	Avg (Salario)
João	B	Silva	12345678966		30.000	33344555587	5	}	5	4	33.250
Fernando	T	Wong	33344555587		40.000	88866555576	5		4	3	31.000
Ronaldo	K	Lima	66688444476		38.000	33344555587	5		1	1	55.000
Joice	A	Leite	45345345376	...	25.000	33344555587	5	}			
Alice	J	Zelaya	99988777767		25.000	98765432168	4				
Jennifer	S	Souza	98765432168		43.000	88866555576	4				
André	V	Pereira	98798798733		25.000	98765432168	4	}			
Jorge	E	Brito	88866555576		55.000	NULL	1				

**Resultado da consulta**

## Agrupamento das tuplas pelo Dnr

- As tuplas da tabela Professor são divididas em grupos: um grupo correspondente a cada departamento (cada valor de cod\_depto).
- As funções COUNT e AVG são aplicadas a cada grupo de tuplas.



# Agrupamento: GROUP BY

Observe que a cláusula SELECT contém apenas o atributo de agrupamento e as funções de agregação a serem aplicadas nos grupos.

Pnome	Minicial	Unome	Cpf	...	Salario	Cpf_supervisor	Dnr
João	B	Silva	12345678966		30.000	33344555587	5
Fernando	T	Wong	33344555587		40.000	88866555576	5
Ronaldo	K	Lima	66688444476		38.000	33344555587	5
Joice	A	Leite	45345345376	...	25.000	33344555587	5
Alice	J	Zelaya	99988777767		25.000	98765432168	4
Jennifer	S	Souza	98765432168		43.000	88866555576	4
André	V	Pereira	98798798733		25.000	98765432168	4
Jorge	E	Brito	88866555576		55.000	NULL	1

  

Dnr	Count (*)	Avg (Salario)
5	4	33.250
4	3	31.000
1	1	55.000

**Resultado da consulta**

**Agrupamento das tuplas pelo Dnr**

## Agrupamento: GROUP BY

Exemplo: Listar a quantidade de empregados por departamento.

```
SELECT dnumero, count(*) as quantidade_empregados  
FROM empregados  
GROUP BY dnumero;
```

## Agrupamento: **GROUP BY**

O que faz esta consulta?

```
SELECT dnumero, MAX(salario)
FROM empregados
GROUP BY dnumero;
```

## Agrupamento: GROUP BY

O que faz esta consulta?

```
SELECT dnumero, COUNT(*)  
FROM empregados  
WHERE salario > 2500  
GROUP BY dnumero;
```

## Agrupamento: GROUP BY

- Se houver NULLs no atributo de agrupamento
  - Um grupo separado é criado para todas as tuplas com um valor NULL no atributo de agrupamento.
- Somente as funções de agregação podem **aparecer no SELECT** e **não aparecer no GROUP BY**.

## Agrupamento: GROUP BY

```
SELECT dnumero, dnome, COUNT(*)  
FROM empregados  
WHERE salario > 2500  
GROUP BY dnumero;
```

dnome **não** pode aparecer na projeção! Por quê?

# HAVING

- Cláusula **HAVING**

- Oferece uma condição sobre a informação de resumo.
- Aplica condições sobre cada grupo.
- A cláusula HAVING deve ser utilizada em conjunto com o GROUP BY
- Somente os grupos que satisfazem a condição do HAVING são recuperados.

# HAVING

- Exemplo

```
SELECT d.dnro, d.nome  
FROM Departamentos d, Empregados e  
WHERE d.dnro = e.dnumero  
GROUP BY d.dnro, d.nome  
HAVING COUNT(*) > 10;
```



# HAVING

- Tanto o GROUP BY quanto o HAVING são aplicados somente após a cláusula WHERE, após as condições de junção e demais condições.
- Primeiro, o SGBD seleciona as tuplas do WHERE e depois aplica HAVING

# Discussão e resumo das consultas em SQL

```
SELECT <lista atributo e função>  
FROM <lista tabela>  
[ WHERE <condição> ]  
[ GROUP BY <atributo(s) de agrupamento> ]  
[ HAVING <condição de grupo> ]  
[ ORDER BY <lista atributos> ];
```

## Exercícios

**Professor** (nome, siape, categoria, salario, cod\_depto)

**Disciplina** (nome, cod\_disciplina, carga\_horária)

**Departamento** (nome\_depto, numero\_depto)

**Ministra** (siape, cod\_disciplina, periodo)

- 1) Recupere a média de salário dos professores que são Adjunto 1.
- 2) Recupere a quantidade de disciplinas com carga horária de 60h.
- 3) Recupere o nome dos professores juntamente com o nome das disciplinas que eles ministraram em 2013.2.
- 4) Para cada departamento, recupere o numero do depto e o valor do maior salário do professor Adjunto I.
- 5) Retorne a quantidade de professores do departamento D001 que ministram a disciplina IF973