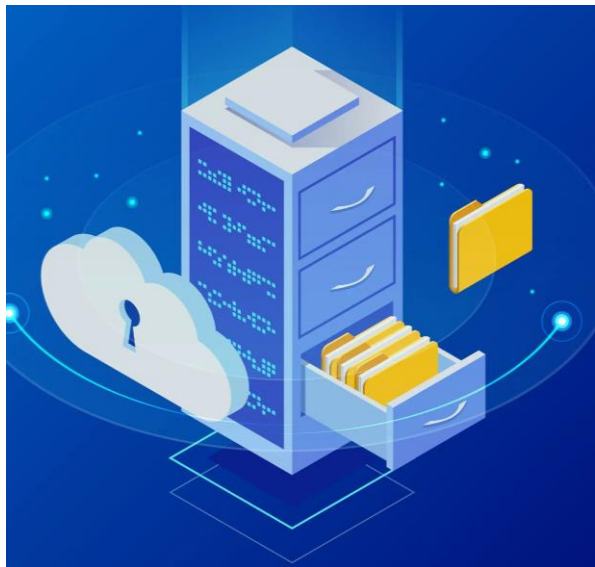


Aonde você quer chegar?
Vai com a



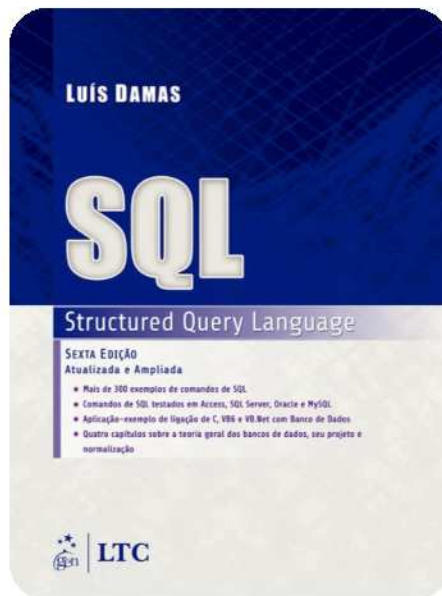


Disciplina: Banco de Dados
Prof. Maurício P. de Freitas MSc.

Aula 08 – 05/09/2024
DQL – Linguagem de Pesquisa de Dados



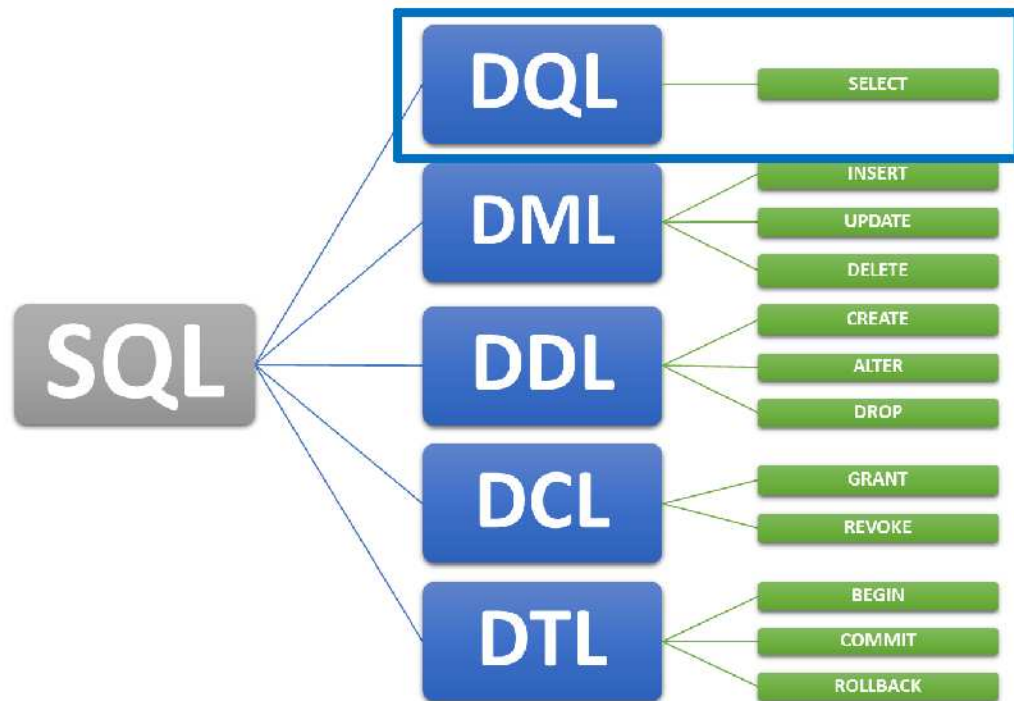
Linguagem SQL



Livros

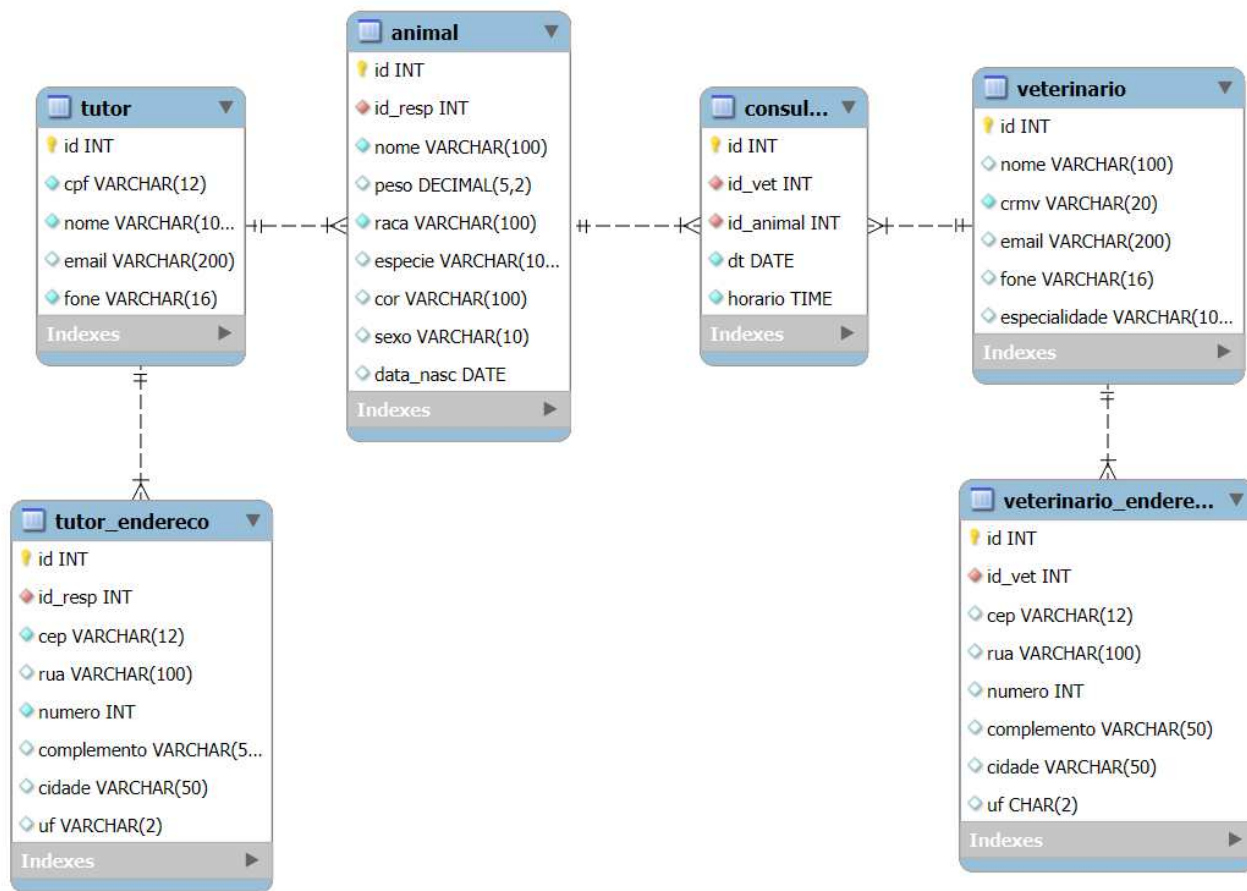
DQL – Linguagem de Consulta de Dados





Conceito

- ❑ DQL → Data query language.
- ❑ Pode-se buscar dados de uma ou mais tabelas relacionadas dentro do banco de dados.
- ❑ Uma das tarefas mais comuns em bancos de dados.
- ❑ Saber criá-las da melhor maneira é muito importante para o desempenho do BD e de aplicações.



□ Modelo lógico, utilizado nos exemplos

Select

- ❑ Principais recursos: ordenação de registros, funções de agregação, junções, entre outros.
- ❑ `SELECT * FROM nome_tabela;`
 - ❑ `SELECT`: identifica que colunas;
 - ❑ `*` = todas;
 - ❑ `FROM`: identifica qual tabela.
- ❑ **Exemplo:**
 1. `SELECT * FROM pet`
 2. `SELECT * FROM responsavel`

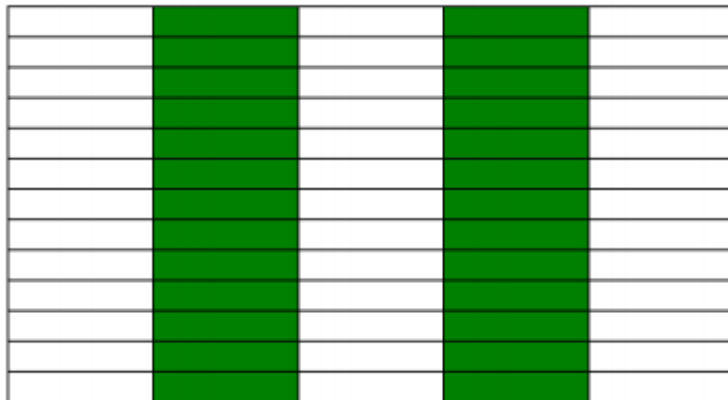
Select - Seleção

- ❑ `SELECT * FROM nome_tabela WHERE condicoes;`

- ❑ **Exemplo:**
 1. `SELECT * FROM Animal
WHERE raca != 'pincher' AND data_nasc > '2019-01-01';`

Select - Seleção

- SELECT **lista_colunas** FROM nome_tabela;



- Exemplo:**
 - SELECT nome, raca, peso FROM Animal;

Like

- ❑ É usada para realizar buscas por padrões específicos em colunas de texto.
- ❑ Ela utiliza caracteres curinga para representar um ou mais caracteres.
 - ❑ Os curingas mais comuns são:
 - ❑ %: Representa zero ou mais caracteres.
 - ❑ _: Representa um único caractere.
- ❑ Exemplos:
 - ❑ `SELECT * FROM Animal WHERE nome LIKE 'B%';`
 - ❑ `SELECT * FROM Animal WHERE nome LIKE '%bb%';`
 - ❑ `SELECT * FROM Animal WHERE nome LIKE '_a%';`

Distinct

- ❑ Trazer informações únicas.
 - ❑ Removendo duplicados.
- ❑ DISTINCT:
 - ❑ `SELECT DISTINCT raza FROM Animal;`

Order By

- ❑ Organiza os resultados de acordo com uma ou mais colunas da tabela.
- ❑ Pode-se definir a ordem dos resultados:
 - ❑ Crescente (ASC) → padrão;
 - ❑ Decrescente (DESC).
- ❑ **Exemplo:**
 1. `SELECT nome, raca, peso FROM Animal ORDER BY raca, peso;`
 2. `SELECT nome, raca, peso FROM Animal ORDER BY raca, peso DESC;`

Order By

- SELECT nome, cpf FROM responsavel ORDER BY nome;

	nome	fone
►	Márcia Luna Duarte	(63) 2980-8765
	Benício Meyer Azevedo	(63) 99931-8289
	Ana Beatriz Albergaria Bochimpani Trindade	(87) 2743-5198
	Thiago Edson das Neves	(85) 3635-5560
	Luna Cecília Alves	(67) 2738-7166

	nome	fone
►	Ana Beatriz Albergaria Bochimpani Trindade	(87) 2743-5198
	Benício Meyer Azevedo	(63) 99931-8289
	Luna Cecília Alves	(67) 2738-7166
	Márcia Luna Duarte	(63) 2980-8765
	Thiago Edson das Neves	(85) 3635-5560

Order By

- ❑ Especifica o número de linhas que devem ser retornadas no resultado de uma consulta.
 - ❑ `SELECT * FROM alunos LIMIT 0, 3;`

	uf	cidade
▶	RJ	Campos dos Goytacazes
	SP	São Paulo
	SP	Diadema
	SP	São Paulo
	SP	São Paulo
	SP	São Paulo
	SP	Araçatuba
	CE	Fortaleza
	RS	Rio Grande
	PI	Teresina

	uf	cidade
▶	RJ	Campos dos Goytacazes
	SP	São Paulo
	SP	Diadema

	uf	cidade
▶	SP	São Paulo
	SP	Araçatuba
	CE	Fortaleza

LIMIT 0, 3

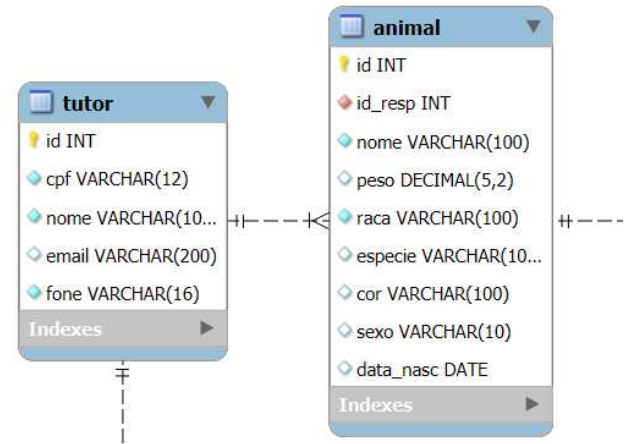
LIMIT 5, 3

❑ Exemplo:

1. `SELECT * FROM Veterinario LIMIT 0,3 ;`
2. `SELECT * FROM Veterinario LIMIT 5,3 ;`
3. `SELECT especialidade, nome FROM Veterinario ORDER BY especialidade LIMIT 3,6 ;`

Junção

- As tabelas devem ser associadas em pares, embora seja possível usar um único comando para juntar várias tabelas.
- Uma das formas mais usadas é a associação da chave primária da primeira tabela com a chave estrangeira da segunda.



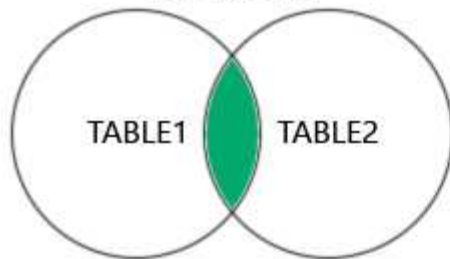
Junção – Usando Where

- Uma das formas de efetuar junções entre tabelas, é com o uso da clausula WHERE.

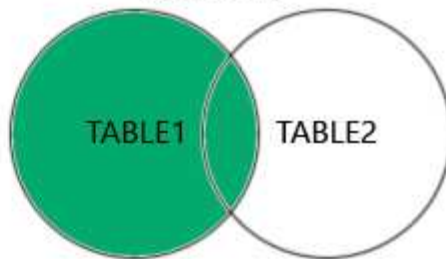
```
SELECT t.nome, t.email, a.nome, a.raca  
FROM Tutor t, animal a  
WHERE t.id = a.id_resp;
```

Junção usando JOIN

INNER JOIN



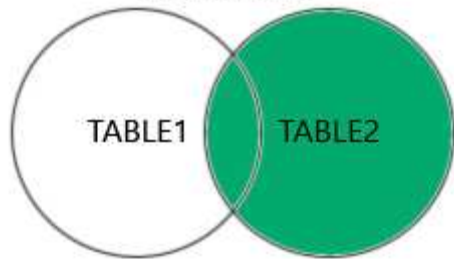
LEFT JOIN



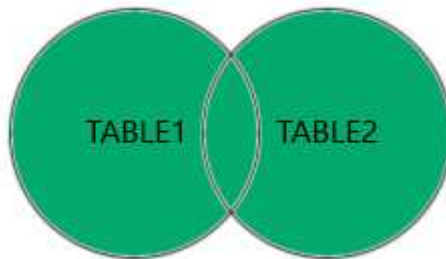
INNER JOIN: Retorna registros que possuem valores correspondentes em ambas as tabelas.

LEFT JOIN: Retorna todos os registros da tabela esquerda e os registros correspondentes da tabela direita.

RIGHT JOIN



CROSS JOIN



RIGHT JOIN: Retorna todos os registros da tabela direita e os registros correspondentes da tabela esquerda.

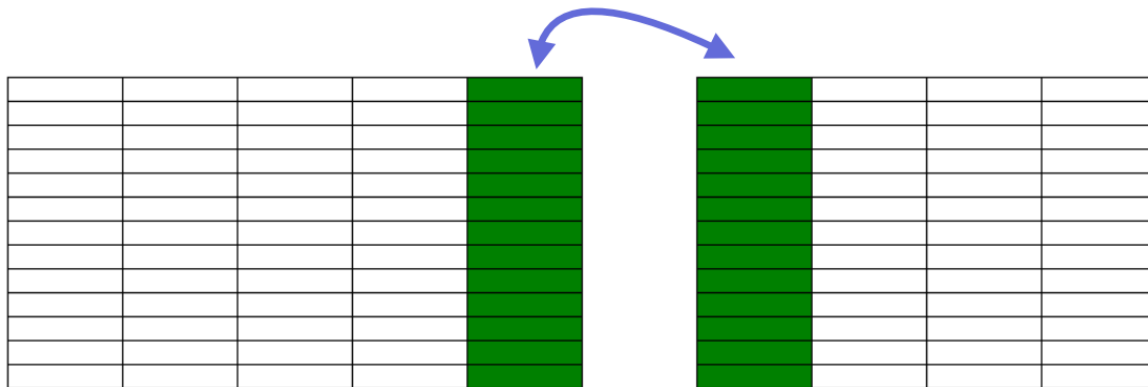
CROSS JOIN: Retorna todos os registros de ambas as tabelas

Junção – Usando Join

- ❑ INNER JOIN: Retorna registros que possuem valores correspondentes em ambas as tabelas
- ❑ LEFT JOIN: Returns all records from the left table, and the matched records from the right table
- ❑ RIGHT JOIN: Retorna todos os registros da tabela direita e os registros correspondentes da tabela esquerda
- ❑ CROSS JOIN: Retorna todos os registros de ambas as tabelas. Se as tabelas possuírem um relacionamento, o resultado será o mesmo que INNER JOIN.

Join

- SELECT t1.status, t2.id FROM nome_tabela1 **t1** INNER JOIN nome_tabela2 **t2** ON (t2.id_tabela1 = t1.id);



- Exemplo:

```
SELECT r.nome, r.email, a.nome, a.raca
FROM responsavel r
JOIN Animal a ON (r.id = a.id_resp);
```


Join

- ❑ **Junção usando LEFT JOIN**

```
SELECT a.nome, a.raca, c.dt, c.horario  
      FROM Animal a  
      LEFT JOIN consulta c ON (c.id_animal = a.id);
```

- ❑ **Junção usando RIGHT JOIN**

```
SELECT a.nome, a.raca, c.dt, c.horario  
      FROM Animal a  
      RIGHT JOIN consulta c ON (c.id_animal = a.id);
```

- ❑ **Junção usando CROSS JOIN**

```
SELECT a.nome, a.raca, c.dt, c.horario  
      FROM Animal a  
      CROSS JOIN consulta c ON (c.id_animal = a.id);
```

Join

❑ **Junção de múltiplas tabelas:**

```
SELECT c.dt, c.horario, a.nome, r.nome, r.fone, v.nome, v.especialidade  
FROM Consulta c  
JOIN veterinario v ON (c.id_vet = v.id)  
JOIN animal a ON (c.id_animal = a.id)  
JOIN responsavel r ON (a.id_resp = r.id)  
ORDER BY dt, horario;
```

Union

- ❑ Combina os resultados de duas ou mais queries em um único result set, retornando todas as linhas pertencentes a todas as queries envolvidas na execução.
 - ❑ O número e a ordem das colunas precisam ser idênticos em todas as queries e os data types precisam ser compatíveis.
- ❑ Existem dois tipos: UNION e UNION ALL.

Union

- ❑ Combina o resultado de execução das duas queries e elimina as linhas duplicadas.
- ❑ `SELECT t.nome, t.email, a.nome, a.raca FROM Tutor t
LEFT JOIN Animal a ON (t.id = a.id_resp)`

UNION

`SELECT t.nome, t.email, a.nome, a.raca FROM Tutor t
RIGHT JOIN Animal a ON (t.id = a.id_resp);`

Union All

- ❑ Combina o resultado de execução das duas queries, entretanto, não elimina as linhas duplicadas.
- ❑ `SELECT * FROM Animal WHERE especie = 'cachorro' UNION ALL SELECT * FROM Animal WHERE peso < 10;`

Funções de Agregação

- ❑ MAX:
 - ❑ `SELECT MAX(peso) FROM Animal;`
 - ❑ `SELECT raza FROM Animal WHERE peso = (SELECT MAX(peso) FROM Animal);`
- ❑ MIN:
 - ❑ `SELECT MIN(peso) FROM Animal;`
 - ❑ `SELECT raza FROM Animal WHERE peso = (SELECT MIN(peso) FROM Animal);`

Funções de Agregação

- ❑ SUM:
 - ❑ `SELECT SUM(peso) FROM Animal;`
- ❑ AVG:
 - ❑ `SELECT AVG(peso) FROM Animal;`
- ❑ COUNT:
 - ❑ `SELECT raza, COUNT(raza) FROM Animal
GROUP BY raza;`

Group By

- ❑ DISTINCT != GROUP BY.
- ❑ É usado em conjunto com as funções de agregação.
- ❑ Agrupa linhas baseado em semelhanças entre elas.
 - ❑ `SELECT raza, COUNT(raza) FROM Animal GROUP BY raza;`

Having

- Podemos usar a cláusula HAVING em conjunto com GROUP BY para filtrar os resultado que serão submetidos a agregação.
- ```
SELECT raza, COUNT(raza)
FROM Animal
GROUP BY raza
HAVING peso < 3.0;
```

## REFERÊNCIAS

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. Sistemas de gerenciamento de banco de dados. São Paulo, SP: McGraw-Hill Interamericana do Brasil, 2008. xxvii, 884 p.

NAVATHE, Shamkant B.; ELMASRI, Ramez. Sistemas de banco de dados. Sham, Addison. Ribeirão Preto SP, 2005.





“Sucesso é o  
acúmulo de  
pequenos esforços,  
repetidos dia e noite.”

Robert Collier

**OBRIGADO E  
BONS ESTUDOS!**





# UniCesumar

EDUCAÇÃO PRESENCIAL E A DISTÂNCIA