

# Banco de Dados II

**PROF. HELTON DE ANDRADE SILVA**

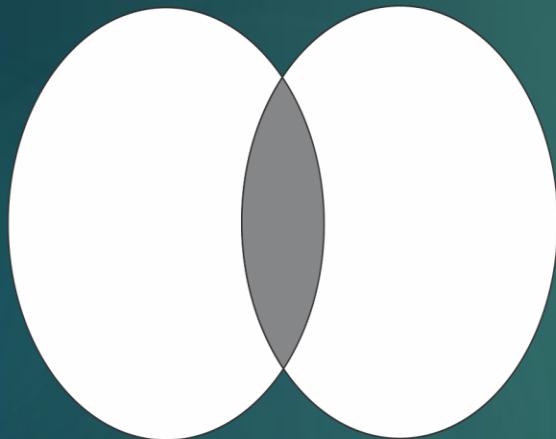
# Associação de Tabelas

- ▶ Para a associação de tabelas, podemos utilizar a cláusula JOIN com a cláusula Where.
- ▶ Para juntar as tabelas como condição de junção, utilizamos uma condição ou grupo de condições.
- ▶ Uma condição que pode ser utilizada é aquela que se baseia nas chaves estrangeiras para ligar as tabelas.
- ▶ Uma característica que deve ser levada em consideração é que as tabelas devem ser associadas em pares, o que não impede que associemos mais de duas tabelas em um só comando.

# Teoria de conjuntos

- ▶ Utilizamos a teoria de conjuntos para entender o funcionamento da cláusula Join.
- ▶ **INNER JOIN:** Retorna apenas os registros que têm correspondência em ambas as tabelas, com base na condição de junção especificada.
- ▶ **LEFT JOIN (ou LEFT OUTER JOIN):** Retorna todos os registros da tabela à esquerda (tabela A), e os registros correspondentes da tabela à direita (tabela B). Se não houver correspondência na tabela à direita, os valores NULL serão retornados.
- ▶ **RIGHT JOIN (ou RIGHT OUTER JOIN):** É o oposto do LEFT JOIN. Retorna todos os registros da tabela à direita (tabela B), e os registros correspondentes da tabela à esquerda (tabela A). Se não houver correspondência na tabela à esquerda, os valores NULL serão retornados.
- ▶ **FULL JOIN (ou FULL OUTER JOIN):** Retorna todos os registros quando há uma correspondência em uma das tabelas. Se não houver correspondência, os valores NULL serão retornados nas colunas da tabela sem correspondência.

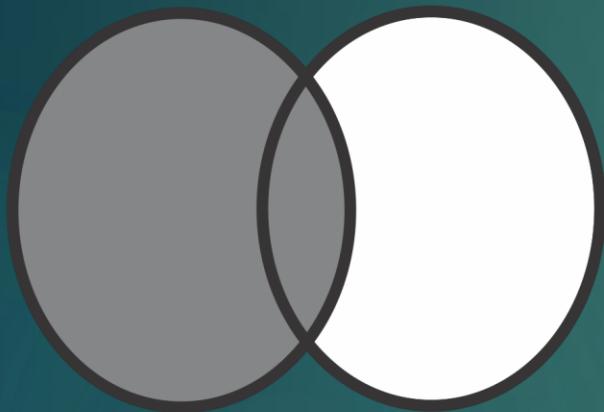
# INNER JOIN



INNER JOIN

- ▶ INNER JOIN: Retorna apenas os registros que têm correspondência em ambas as tabelas, com base na condição de junção especificada.

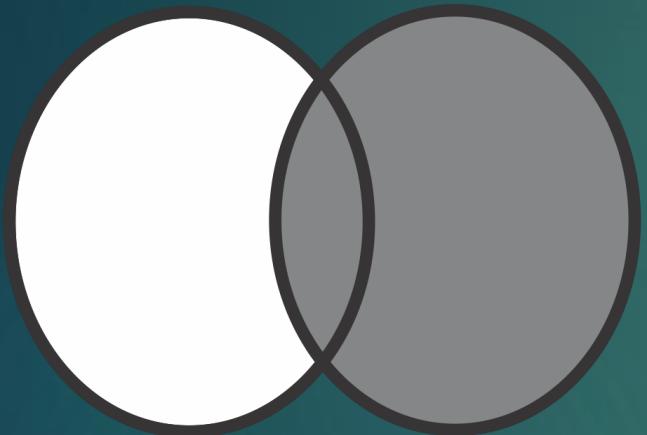
# LEFT JOIN



LEFT JOIN

- ▶ **LEFT JOIN (ou LEFT OUTER JOIN):** Retorna todos os registros da tabela à esquerda (tabela A), e os registros correspondentes da tabela à direita (tabela B). Se não houver correspondência na tabela à direita, os valores NULL serão retornados.

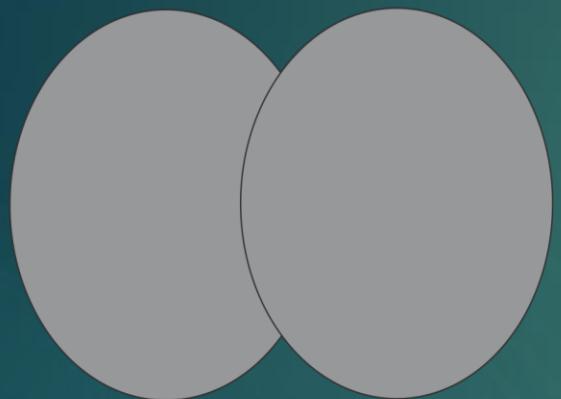
# RIGHT JOIN



RIGHT JOIN

- ▶ **RIGHT JOIN (ou RIGHT OUTER JOIN):** É o oposto do LEFT JOIN. Retorna todos os registros da tabela à direita (tabela B), e os registros correspondentes da tabela à esquerda (tabela A). Se não houver correspondência na tabela à esquerda, os valores NULL serão retornados.

# FULL JOIN



**FULL JOIN**

- ▶ **FULL JOIN (ou FULL OUTER JOIN):** Retorna todos os registros quando há uma correspondência em uma das tabelas. Se não houver correspondência, os valores NULL serão retornados nas colunas da tabela sem correspondência.

Para exercitarmos os comandos  
vamos utilizar as seguintes tabelas

Cod_Cargo (PK)	Nome_cargo (chave única)
1	Presidente
2	Gerente
3	Supervisor
4	Revisor
5	Redator

Cod_Func (PK)	Cod_cargo (Fk)	Nome_func	Salário_func	RG_func
1	5	Luiz Pereira	3000,00	27.291.905
2	5	Antonio Almeida	3000,00	15.970.247
3	3	Donizete Ribeiro	2800,00	27.151.908
4	3	Gabriela Moura	4700,00	255.279.145
5	2	Emilio Duarte	5000,00	17.278.246
6	1	Carolina Ferreira	9000,00	18.154.578

# Associando tabelas

Se compararmos a criação da tabela **Cargo** e **Funcionário**, podemos notar um aspecto em comum, a coluna **Cod\_cargo** está presente em ambas. Contudo recebe funções diferentes em cada tabela:

Na Tabela **CARGO**, **Cod\_cargo** é chave primária;

Na tabela **FUNCIONÁRIO**, **Cod\_cargo** é chave estrangeira;

- ▶ Para que sejam exibidos os dados relacionados das duas tabelas, de maneira que sejam apresentados todos os cargos que possuem funcionários e todos os funcionários que desempenham algum cargo, podemos utilizar a cláusula **INNER JOIN** ou a cláusula **WHERE**.

### ▶ CÓDIGO COM INNER JOIN

```
Select CARGO.NOME_CARGO, Funcionario.Nome_Func  
From cargo Inner Join Funcionario  
On Cargo.Cod_cargo=Funcionario.Cod_Cargo;
```

### CÓDIGO COM WHERE

```
Select cargo.Nome_cargo, Funcionario.Nome_func  
From cargo, Funcionario  
Where Cargo.Cod_cargo=Funcionario.Cod_cargo
```

# LEFT JOIN

VEJAMOS A SEGUINTE TABELA

NOME_CARGO	NOME_FUNC
REDATOR	LUIZ PEREIRA
REDATOR	ANTONIO ALMEIDA
SUPERVISOR	DONIZETE RIBEIRO
SUPERVISOR	GABRIELA MOURA
GERENTE	EMILIO DUARTE
PRESIDENTE	CAROLINA FERREIRA
REVISOR	

# COMANDO LEFT JOIN

```
Select CARGO.NOME_CARGO, Funcionario.Nome_Func  
From cargo LEFT outer JOIN Funcionario  
On Cargo.Cod_cargo=Funcionario.Cod_Cargo;
```

# RIGHT JOIN

- ▶ VAMOS SUPOR QUE, TENHAMOS TROCADO A POSIÇÃO DAS TABELAS. DIANTE DESSA SITUAÇÃO, AINDA PRETENDEMOS OBTER OS MESMO DADOS OBTIDOS COM TAL COMANDO. PARA ISSO, SERÁ NECESSÁRIO UTILIZAR A CLÁUSULA RIGHT OUTER JOIN, POR MEIO DA QUAL CONSEGUIREMOS TANTO OS DADOS RELACIONADOS ENTRE AS TABELAS CARGO E FUNCIONÁRIOS, COMO OS DADOS NÃO-RELACIONADOS LOCALIZADOS NA TABELA À DIREITA DA CLÁUSULA JOIN.

# COMANDO RIGTH OUTER JOIN

```
Select CARGO.NOME_CARGO, Funcionario.Nome_Func  
From cargo right outer JOIN Funcionario  
On Cargo.Cod_cargo=Funcionario.Cod_Cargo;
```

# Vejamos a tabela a seguir

Cod_Func (PK)	Cod_cargo (Fk)	Nome_func	Salário_fu nc	RG_func
1	5	Luiz Pereira	3000,00	27.291.905
2	5	Antonio Almeida	3000,00	15.970.247
3	3	Donizete Ribeiro	2800,00	27.151.908
4	3	Gabriela Moura	4700,00	255.279.145
5	2	Emilio Duarte	5000,00	17.278.246
6	1	Carolina Ferreira	9000,00	18.154.578
7	0	<b>Jefferson Carvalho</b>	<b>2000.00</b>	<b>15.754.159</b>

- Mesmo que o cargo do novo funcionário não esteja registrado na tabela Cargo, é possível exibir este e todos os outros nomes de funcionários, como os dados apresentados a seguir:

NOME_CARGO	NOME_FUNC
REDATOR	LUIZ PEREIRA
REDATOR	ANTONIO ALMEIDA
SUPERVISOR	DONIZETE RIBEIRO
SUPERVISOR	GABRIELA MOURA
GERENTE	EMILIO DUARTE
PRESIDENTE	CAROLINA FERREIRA
	JEFFERSON CARVALHO

# Associando múltiplas tabelas

O usuário é capaz de associar diversas tabelas. É exatamente a respeito dessa capacidade que trataremos neste tópico, onde iremos considerar as tabelas Cargo, Funcionário e Dependente

# Tabela dependente

Cod_dep (PK)	Cod_func (FK)	Nome_dep
1	1	Mariana Leme
2	1	Camila Leme
3	1	Josival Leme
4	2	Clovis Almeida
5	2	Durval Almeida
6	5	Fabiana Duarte
7	5	Joana Duarte

# Dados da 3 tabelas

- ▶ Primeiramente vamos gerar dados relacionados as tabelas, cargo, Funcionario e Dependente, lembrando que nem todos os funcionários possuem dependentes.

```
select funcionario.nome_func, cargo.nome_cargo,  
dependente.nome_dep  
from cargo inner join funcionários  
on cargo.cod_cargo=funcionario.cod_cargo  
inner join dependente  
on funcionario.cod_func=dependente.cod_func
```

# Retornando funcionário que não possuem dependentes

```
select funcionario.nome_func, cargo.nome_cargo,  
dependente.nome_dep from cargo inner join funcionario on  
cargo.cod_cargo=funcionario.cod_cargo  
left join dependente on  
funcionario.cod_func=dependente.cod_func
```

Retornando todos os cargos  
independente se estes tiverem  
dependentes, e todos os funcionários,  
inclusive aqueles que não possuem  
dependentes.

```
select funcionario.nome_func, cargo.nome_cargo,  
dependente.nome_dep  
from cargo left join funcionario  
on cargo.cod_cargo=funcionario.cod_cargo  
left join dependente  
on funcionario.cod_func=dependente.cod_func
```

# Exercícios complementares

- ▶ 1) Quais são as cláusulas utilizadas para fazer a associação de tabelas?
- ▶ 2) Qual a função da cláusula INNER JOIN?
- ▶ 3) Qual a cláusula gera os dados relacionados entre duas tabelas e os dados não relacionados localizados na tabela posicionada à direita do Join?
- ▶ 4) Quais comandos utilizamos com JOIN para alterar ou remover informações de uma tabela, tendo como base os dados de uma segunda tabela e estando elas inter-relacionadas?