

# INNER, CROSS, LEFT, RIGHT E FULL JOINS

Prof. Me. Sérgio Carlos Portari Júnior  
Banco de Dados II

# Joins




- ▶ As junções entre duas ou mais tabelas podem ser realizadas através de:
  - a. Cross join
  - b. Inner join
  - c. Left outer join
  - d. Right outer join
  - e. Outer Full join

# Joins




- ▶ As questões são duas:
  - i. O que são cada uma dessas junções (joins)?
  - ii. Como usar?
- ▶ Para definir e exemplificar as junções acima citadas vamos criar as tabelas a seguir:

# Joins

**FUNCIONARIO**

 Matricula	INTEGER	NOT NULL
 NomeFuncionario	VARCHAR(80)	NOT NULL
 CodCargo (FK)	CHAR(2)	NULL

**CARGO**

 CodCargo	CHAR(2)	NOT NULL
 NomeCargo	VARCHAR(50)	NOT NULL
 ValorCargo	MONEY(10,0)	NOT NULL

```
CREATE TABLE Cargo (  
Codcargo char(2) not null,  
Nomecargo varchar(50)  
null,  
Valorcargo double null,  
primary key (codcargo))  
Engine=InnoDB;
```

```
CREATE TABLE Funcionario (  
Matricula int not null,  
NomeFuncionario varchar(50)  
not null,  
Codcargo char(2) null,  
primary key (Matricula),  
foreign key (Codcargo)  
references cargo  
(Codcargo)) Engine=InnoDB;
```

# Joins

- ▶ Insira os seguintes registros:

TODAS AS LINHAS....bTestes (sa (75))

```
SELECT * FROM CARGO AS C --> Apelidamos a tabelas Cargo de C neste artigo
SELECT * FROM FUNCIONARIO AS F --> Apelidamos funcionário de F neste artigo
```

Results Messages

	CodCargo	NomeCargo	ValorCargo
1	C1	CAIXA	800,00
2	C2	VENDEDOR	1200,00
3	C3	GERENTE	2400,00

**CARGO AS C**

	Matricula	NomeFuncionario	CodCargo
1	100	JOÃO	C1
2	110	MARIA	C2
3	120	CARLOS	C1
4	130	TADEU	NULL

**FUNCIONARIO AS F**

# Cross Join

- ▶ Quando queremos juntar duas ou mais tabelas por cruzamento. Ou seja, para cada linha da tabela FUNCIONARIO queremos todos os CARGOS ou vice-versa.

```
SELECT F.NomeFuncionario, C.NomeCargo  
FROM CARGO AS C  
CROSS JOIN FUNCIONARIO AS F
```

# Inner Join

- ▶ Quando queremos juntar duas ou mais tabelas por coincidência.
- ▶ Para cada linha da tabela FUNCINARIO queremos o CARGO correspondente que internamente (INNER), em seus valores de atributos, coincidam.
- ▶ No caso de FUNIONÁRIO e CARGO os atributos internos coincidentes são codigoCargo na tabela CARGO e codigoCargo na tabela FUNCIONARIO.

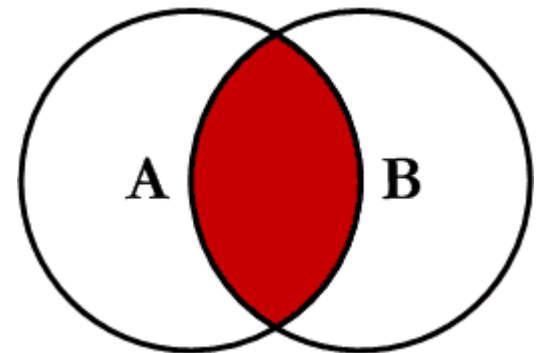
# Inner Join

- ▶ Lembrem que `codigoCargo` é chave primária da tabela `CARGO` e chave estrangeira na tabela `FUNCIONARIO`.
- ▶ Para efetivarmos a junção das duas tabelas se fará necessário ligar (ON) as duas tabelas por seus atributos internos (INNER) coincidentes.



# Inner Join

```
SELECT F.NomeFuncionario, C.NomeCargo  
FROM CARGO AS C  
INNER JOIN FUNCIONARIO AS F  
ON (F.CodCargo = C.CodCargo);
```



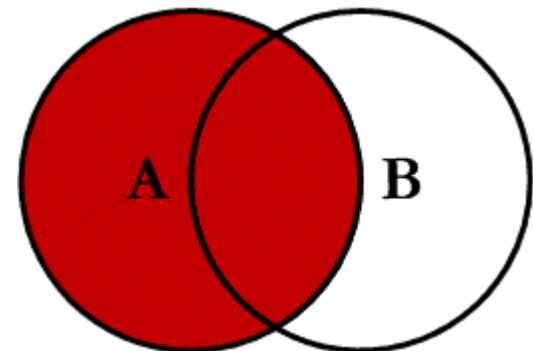
# Left Outer Join

- ▶ Se desejarmos listar todos os funcionários com seus respectivos cargos, incluindo os funcionários sem cargos, a exemplo de Tadeu, poderíamos usar todo o poder da junção INNER JOIN adicionando ainda OUTER (EXTERNOS / OUTROS) Funcionários que não fazem parte do INNER JOIN.
- ▶ Justamente àqueles sem cargos, como Tadeu. Podemos conseguir esse feito com a junção FUNCIONARIO/CARGO através da declaração **FUNCIONARIO OUTER LEFT JOIN CARGO**, que promove a junção interna (INNER) de todos os funcionários a cargos e lista ainda outros (EXTERNOS/OUTER) não associados.

# Left Outer Join

- ▶ Uma observação importante é que a ordem da ligação (ON) não faz diferença, ou seja: “ON (F.codCargo = C.codCargo)” é exatamente igual a “ON (C.codCargo = F.codCargo)”

```
SELECT F.nomeFuncionario,C.nomeCargo  
FROM FUNCIONARIO AS F  
LEFT OUTER JOIN CARGO AS C ON  
( C.codCargo = F.codCargo );
```



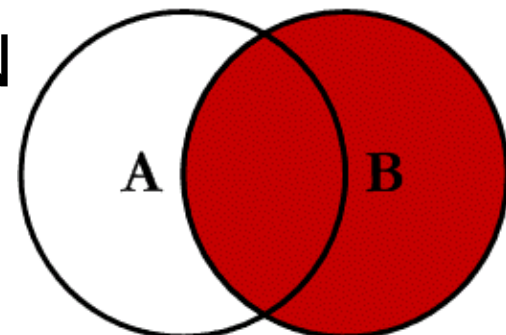
# Right Outer Join

- ▶ Observando a tabela CARGO notamos que o cargo GERENTE, com código C3, não é referenciado/associado por/a nenhum funcionário na tabela FUNCIONARIO.
- ▶ Se desejarmos listar todos os CARGOS e seus respectivos FUNCIONARIOS, incluindo os CARGOS sem FUNCIONÁRIOS, poderíamos usar a junção **RIGHT OUTER JOIN**.

# Right Outer Join

- ▶ Uma observação importante é que a ordem da ligação (ON) não faz diferença, ou seja: “ON (F.codCargo = C.codCargo)” é exatamente igual a “ON (C.codCargo = F.codCargo)”

```
SELECT F.nomeFuncionario, C.nomeCargo  
FROM FUNCIONARIO AS F  
RIGHT OUTER JOIN CARGO AS C ON  
( F.codCargo = C.codCargo );
```

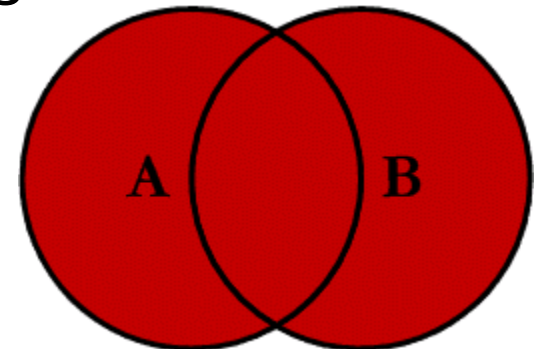


# Union (Outer Full Join)

- ▶ Aqui juntamos o resultado das junções (JOIN) internas(INNER), a listagem de todas as outras linhas não associadas, tanto do lado direito (RIGHT) da junção como do lado ESQUEDO (LEFT).

```
SELECT F.nomeFuncionario,C.nomeCargo  
FROM FUNCIONARIO AS F  
LEFT OUTER JOIN CARGO AS C ON  
( C.codCargo = F.codCargo )  
UNION
```

```
SELECT F.nomeFuncionario, C.nomeCargo  
FROM FUNCIONARIO AS F  
RIGHT OUTER JOIN CARGO AS C ON  
( F.codCargo = C.codCargo );
```



# Exercícios

- ▶ Amanhã.....