



Banco de Dados

06 – Normalização de Banco de Dados

Prof. André Ulisses
andre.ulisses@edu.sc.senai.br

DQL: Linguagem de Consulta de Dados – Principais comandos.

Select – Consultando dados

- Simples
- Junções Internas
 - INNER JOIN
- Junções Externas
 - LEFT OUTER JOIN
 - RIGHT OUTER JOIN
 - FULL OUTER JOIN
- Junções Cruzadas
 - CROSS JOIN
- Junções Naturais
 - NATURAL JOIN
- Auto Junção

Consultas Joins

Nos Bancos de Dados relacionais a relação entre tabelas é o ponto fundamental da sua existência. Por isso muitas informações são acessadas através de junções entre tabelas.

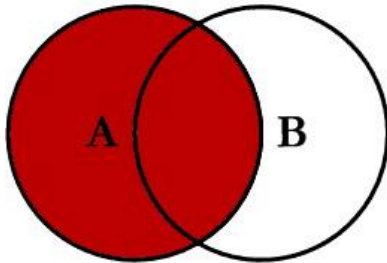
Os relacionamentos são organizados através das chaves estrangeiras (FK).

Atenção: Tem-se sempre a preferencia ao juntar duas tabelas à partir dos campos utilizados na foreign key existente entre as tabelas.

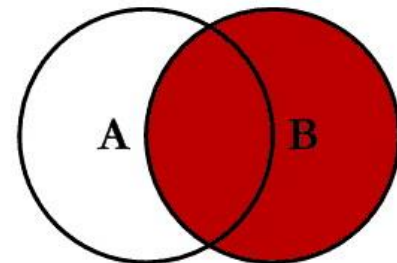
Junções

As Junções propriamente ditas já se caracterizam por trazerem somente as informações **verdadeiras**. Tendo como ponto fundamental para isso a **condição de junção**, que é a condição na cláusula **WHERE** ou na cláusula **ON**, que representa efetivamente a junção entre as duas tabelas. Esta **condição de junção** é sempre baseada em **chaves estrangeiras**.

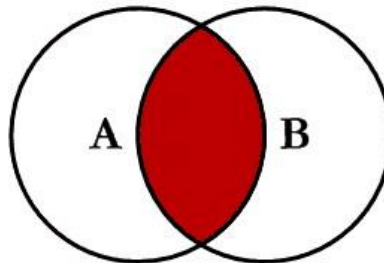
SQL JOINS



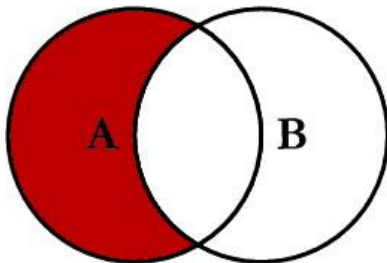
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



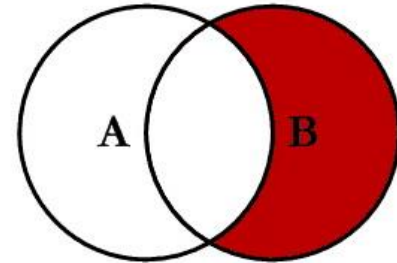
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



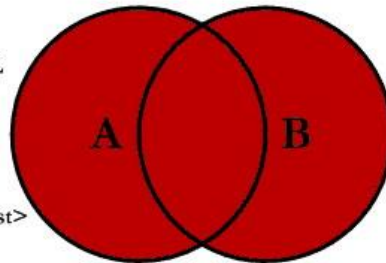
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



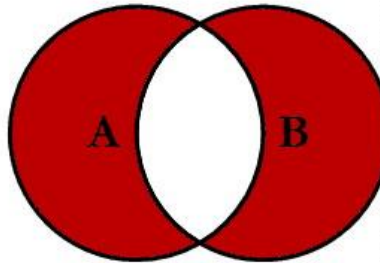
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL
```

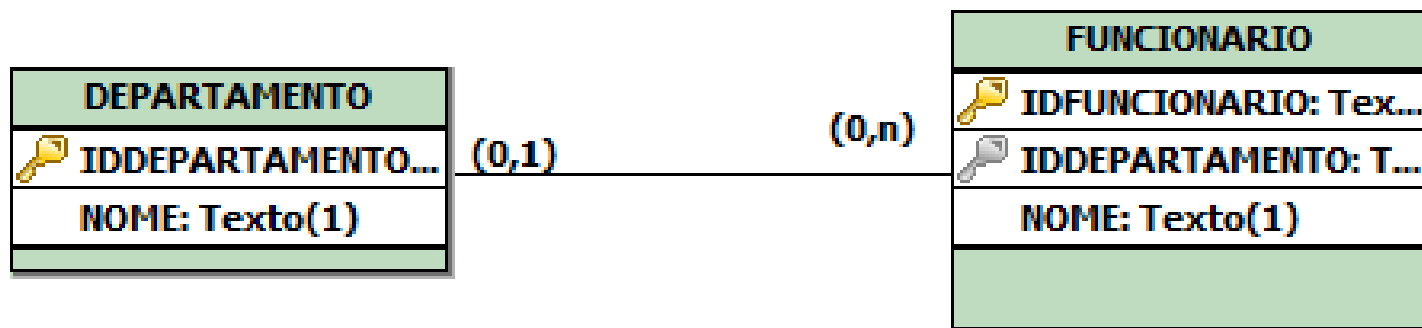
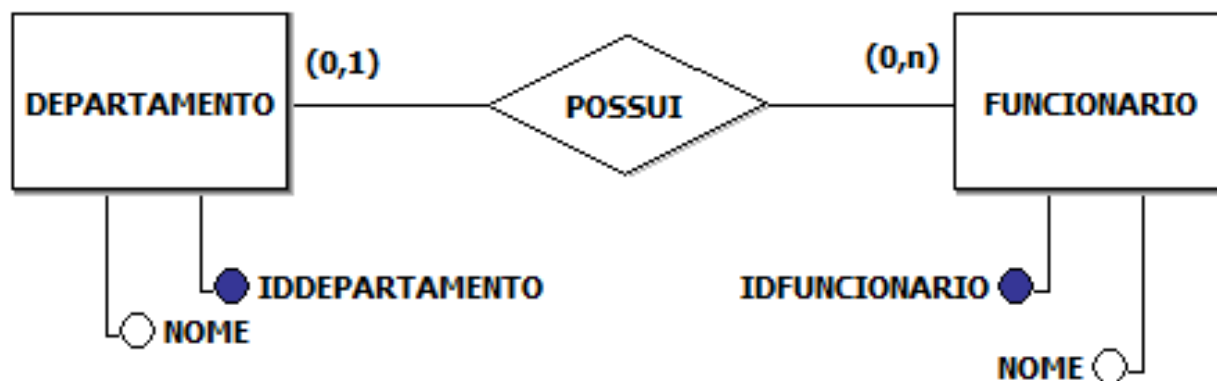
Consultas Joins

Quando precisamos acessar as informações entre duas tabelas, o SQL nos fornece um recurso para acesso simultâneo a tabelas que encontram-se relacionadas. Esse recurso é identificado como junção (Join) ou produto cartesiano.

Uma junção de tabelas cria uma pseudo-tabela derivada de duas ou mais tabelas de acordo com as regras especificadas, e que são parecidas com as regras da teoria dos conjuntos

Atenção: Tem-se preferência ao juntar duas tabelas à partir dos campos utilizados na **foreign key** existente entre as tabelas.

SQL - CONSULTA DE DADOS



```
CREATE TABLE DEPARTAMENTO (  
    IDDEPARTAMENTO INT NOT NULL PRIMARY KEY,  
    NOME VARCHAR(100)  
);
```

```
CREATE TABLE FUNCIONARIO (  
    IDFUNCIONARIO INT NOT NULL PRIMARY KEY,  
    IDDEPARTAMENTO INT NOT NULL,  
    NOME VARCHAR(100),  
    FOREIGN KEY (IDDEPARTAMENTO)  
        REFERENCES DEPARTAMENTO (IDDEPARTAMENTO)  
);
```


Consultas Joins

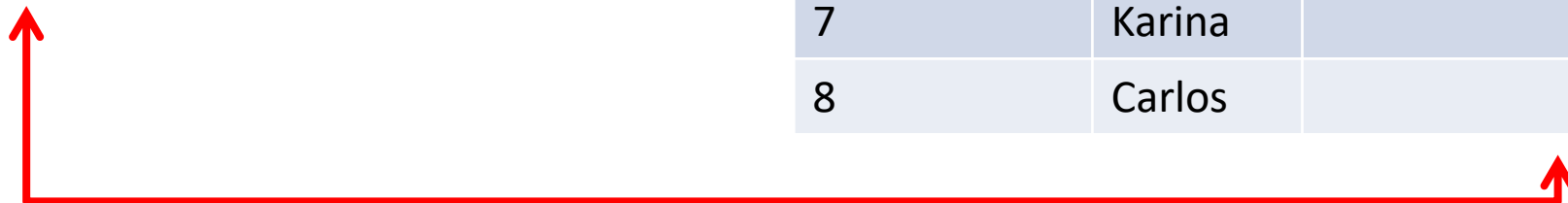
Exemplo:

Tabela **Departamento**

IDDEPARTAMENTO	NOME
1	Administração
2	Marketing
3	Financeiro
4	Cobrança
5	Logística
6	manutenção

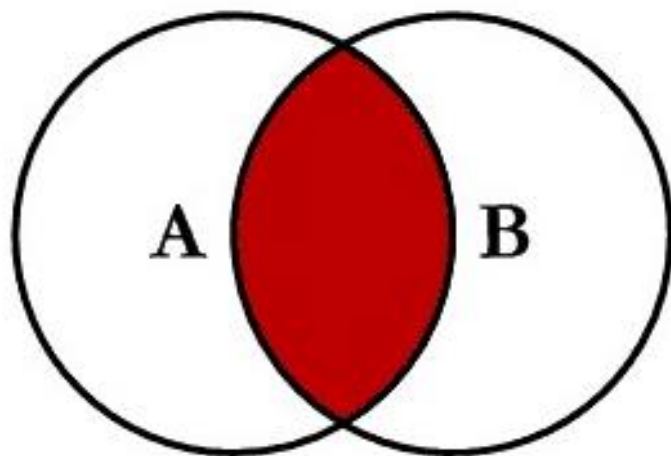
Tabela **Funcionário**

IDFUNCIONARIO	NOME	IDDEPARTAMENTO
1	João	1
2	Maria	2
3	Marcia	2
4	Ana	3
5	Lucas	3
6	Antonio	5
7	Karina	
8	Carlos	



Inner Join

As junções INNER JOIN se caracterizam por uma seleção que retorna apenas os dados que atendem a **condição de junção**



```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```

Inner Join

As junções INNER JOIN se caracterizam por uma seleção que retorna apenas os dados que atendem a **condição de junção**

$$A = \{2, 4, 6\}$$

$$B = \{4, 6, 8\}$$

$$A * B = \{ (2,4), (2,6), (2,8), (4,4), (4,6), (4,8), (6,4), (6,6), (6,8) \}$$

$$A \text{ INNER JOIN } B = \{ (4,4), (6,6) \}$$

Inner Joins

Exemplo:

Tabela **Departamento**

Iddepartamento	nome
1	Administração
2	Marketing
3	Financeiro
4	Cobrança
5	Logística
6	manutenção

Tabela **Funcionário**

Idfuncionario	Nome	iddepartamento
1	João	1
2	Maria	2
3	Marcia	2
4	Ana	3
5	Lucas	3
6	Antonio	5
7	Karina	
8	Carlos	


Consultas Joins – Inner Join

```
SELECT
    DEPARTAMENTO.IDDEPARTAMENTO
    , DEPARTAMENTO.NOME
    , FUNCIONARIO.IDFUNCIONARIO
    , FUNCIONARIO.NOME
FROM
    DEPARTAMENTO
INNER JOIN FUNCIONARIO ON
    DEPARTAMENTO.IDDEPARTAMENTO = FUNCIONARIO.IDDEPARTAMENTO;
```

Iddepartamento	Nome	Idfuncionario	nome
1	Administração	1	João
2	Marketing	2	Maria
2	Marketing	3	Marcia
3	Financeiro	4	Ana
3	Financeiro	5	Lucas
5	Logística	6	Antonio

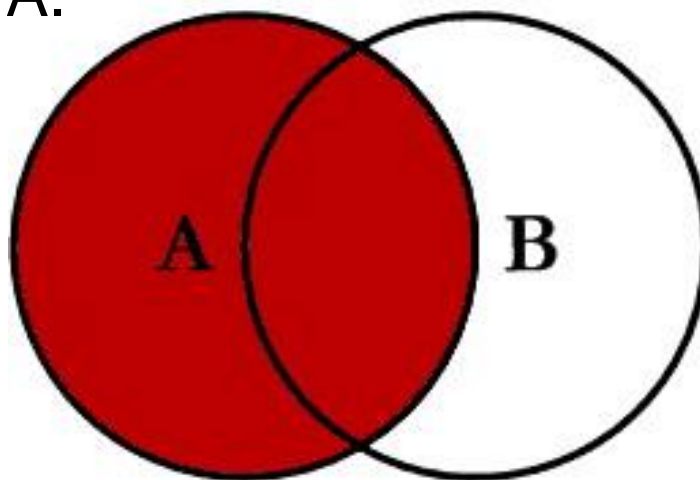
```
CREATE TABLE DEPARTAMENTO (  
    IDDEPARTAMENTO INT NOT NULL PRIMARY KEY,  
    NOME VARCHAR(100)  
);  
CREATE TABLE FUNCIONARIO (  
    IDFUNCIONARIO INT NOT NULL PRIMARY KEY,  
    IDDEPARTAMENTO INT NOT NULL,  
    NOME VARCHAR(100),  
    FOREIGN KEY (IDDEPARTAMENTO) REFERENCES DEPARTAMENTO (IDDEPARTAMENTO)  
);
```

```
SELECT  
    DEPARTAMENTO.IDDEPARTAMENTO  
    , DEPARTAMENTO.NOME  
    , FUNCIONARIO.IDFUNCIONARIO  
    , FUNCIONARIO.NOME  
FROM  
    DEPARTAMENTO  
    INNER JOIN FUNCIONARIO ON  
    DEPARTAMENTO.IDDEPARTAMENTO = FUNCIONARIO.IDDEPARTAMENTO  
;
```



Left Join

As junções externas LEFT JOIN se caracterizam por uma seleção que retorna todos os dados da tabela da ESQUERDA (left) e apenas os dados que atendem a **condição de junção** da tabela da DIREITA.



```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```

Left Join

As junções externas LEFT OUTER JOIN se caracterizam por uma seleção que retorna todos os dados da tabela da ESQUERDA (left) e apenas os dados que atendem a **condição de junção** da tabela da DIREITA.

$$A = \{2, 4, 6\}$$

$$B = \{4, 6, 8\}$$

$$A * B = \{ (2,4), (2,6), (2,8), (4,4), (4,6), (4,8), (6,4), (6,6), (6,8) \}$$

$$A \text{ LEFT OUTER JOIN } B = \{ (2,\text{null}), (4,4), (6,6) \}$$

Left Joins

Exemplo:

Tabela **Departamento**

Iddepartamento	nome
1	Administração
2	Marketing
3	Financeiro
4	Cobrança
5	Logística
6	manutenção

Tabela **Funcionário**

Idfuncionario	Nome	iddepartamento
1	João	1
2	Maria	2
3	Marcia	2
4	Ana	3
5	Lucas	3
6	Antonio	5
7	Karina	
8	Carlos	

Consultas Joins – Left Join


```
SELECT DEPARTAMENTO.IDDEPARTAMENTO, DEPARTAMENTO.NOME  
    , FUNCIONARIO.IDFUNCIONARIO, FUNCIONARIO.NOME  
FROM DEPARTAMENTO LEFT JOIN FUNCIONARIO ON  
    DEPARTAMENTO.IDDEPARTAMENTO = FUNCIONARIO.IDDEPARTAMENTO;
```

Iddepartamento	Nome	Idfuncionario	nome
1	Administração	1	João
2	Marketing	2	Maria
2	Marketing	3	Marcia
3	Financeiro	4	Ana
3	Financeiro	5	Lucas
4	Cobrança		
5	Logística	6	Antonio
6	Manutenção		

SQL - CONSULTA DE DADOS

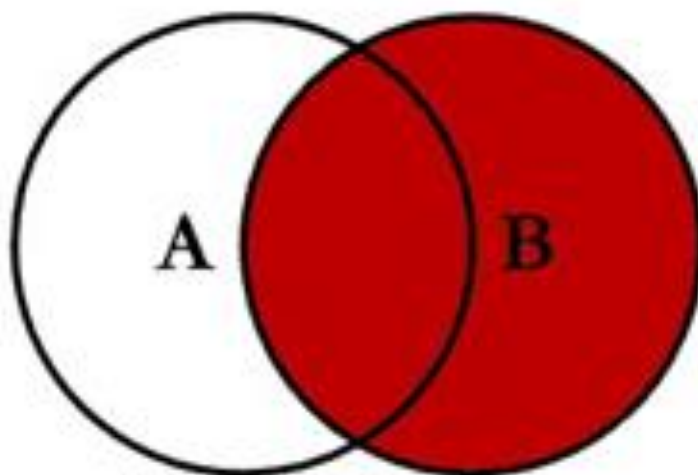
```
CREATE TABLE DEPARTAMENTO (  
    IDDEPARTAMENTO INT NOT NULL PRIMARY KEY,  
    NOME VARCHAR(100)  
);  
CREATE TABLE FUNCIONARIO (  
    IDFUNCIONARIO INT NOT NULL PRIMARY KEY,  
    IDDEPARTAMENTO INT NOT NULL,  
    NOME VARCHAR(100),  
    FOREIGN KEY (IDDEPARTAMENTO) REFERENCES DEPARTAMENTO (IDDEPARTAMENTO)  
);
```

```
SELECT  
    DEPARTAMENTO.IDDEPARTAMENTO  
    , DEPARTAMENTO.NOME  
    , FUNCIONARIO.IDFUNCIONARIO  
    , FUNCIONARIO.NOME  
FROM  
    DEPARTAMENTO  
LEFT JOIN FUNCIONARIO ON  
    DEPARTAMENTO.IDDEPARTAMENTO = FUNCIONARIO.IDDEPARTAMENTO  
;
```



Right Join

As junções externas RIGHT OUTER JOIN se caracterizam por uma seleção que retorna todos os dados da tabela da DIREITA (right) e apenas os dados que atendem a **condição de junção** da tabela da ESQUERDA.



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```

Right Join

As junções externas RIGHT JOIN se caracterizam por uma seleção que retorna todos os dados da tabela da DIREITA (right) e apenas os dados que atendem a **condição de junção** da tabela da ESQUERDA.

$A = \{2, 4, 6\}$

$B = \{4, 6, 8\}$

$A * B = \{ (2,4), (2,6), (2,8), (4,4), (4,6), (4,8), (6,4), (6,6), (6,8) \}$

$A \text{ RIGHT OUTER JOIN } B = \{ (4,4), (6,6), (\text{null},8) \}$

Right Joins

Exemplo:

Tabela **Departamento**

Iddepartamento	nome
1	Administração
2	Marketing
3	Financeiro
4	Cobrança
5	Logística
6	manutenção

Tabela **Funcionário**

Idfuncionario	Nome	iddepartamento
1	João	1
2	Maria	2
3	Marcia	2
4	Ana	3
5	Lucas	3
6	Antonio	5
7	Karina	
8	Carlos	

Consultas Joins – Right Join

```
SELECT DEPARTAMENTO.IDDEPARTAMENTO, DEPARTAMENTO.NOME,  
FUNCIONARIO.IDFUNCIONARIO, FUNCIONARIO.NOME  
FROM DEPARTAMENTO RIGHT JOIN FUNCIONARIO ON  
DEPARTAMENTO.IDDEPARTAMENTO = FUNCIONARIO.IDDEPARTAMENTO;
```

Iddepartamento	Nome	Idfuncionario	nome
1	Administração	1	João
2	Marketing	2	Maria
2	Marketing	3	Marcia
3	Financeiro	4	Ana
3	Financeiro	5	Lucas
4	Logística	6	Antonio
		7	Karina
		8	Carlos

```
CREATE TABLE DEPARTAMENTO (  
    IDDEPARTAMENTO INT NOT NULL PRIMARY KEY,  
    NOME VARCHAR(100)  
);  
CREATE TABLE FUNCIONARIO (  
    IDFUNCIONARIO INT NOT NULL PRIMARY KEY,  
    IDDEPARTAMENTO INT NOT NULL,  
    NOME VARCHAR(100),  
    FOREIGN KEY (IDDEPARTAMENTO) REFERENCES DEPARTAMENTO (IDDEPARTAMENTO)  
);
```

```
SELECT  
    DEPARTAMENTO.IDDEPARTAMENTO  
    , DEPARTAMENTO.NOME  
    , FUNCIONARIO.IDFUNCIONARIO  
    , FUNCIONARIO.NOME  
FROM  
    DEPARTAMENTO  
    RIGHT JOIN FUNCIONARIO ON  
    DEPARTAMENTO.IDDEPARTAMENTO = FUNCIONARIO.IDDEPARTAMENTO  
;
```

