



Universidade Federal de Sergipe
Departamento de Sistemas de Informação
Itatech Group Jr – Softwares Itabaiana

Site: www.itatechjr.com.br

E-mail: contato@itatechjr.com.br



Linguagem SQL (Parte III)

Introdução a Banco de Dados

André Vinicius R. P. Nascimento
andreviniciusnascimento@gmail.com



Conteúdo

- Aliases para Tabelas
- Junções
- Junções e Filtros
- Junções e Nulos



Aliases para Tabelas

- Na linguagem SQL, podemos nomear as tabelas na cláusula SELECT
- Isso vai ser muito útil quando precisarmos identificar em qual tabela encontra-se determinado atributo.

```
SELECT f.matricula, f.nome  
FROM TB_FUNCIONARIO as f
```

Ou

```
SELECT f.matricula, f.nome  
FROM TB_FUNCIONARIO f
```



Junções

- Considere as tabelas

```
CREATE TABLE TB_FUNCIONARIO (  
    matricula int NOT NULL PRIMARY KEY,  
    nm_funcionario varchar (50) NOT NULL,  
    cd_departamento int  
)
```

```
CREATE TABLE TB_DEPARTAMENTO (  
    cd_departamento int NOT NULL PRIMARY KEY,  
    nm_departamento varchar(40) NOT NULL  
)
```



Junções

- As tabelas TB_FUNCIONARIO e TB_DEPARTAMENTO possuem um relacionamento que está representado pelo atributo cd_departamento na tabela TB_FUNCIONARIO.

```
CREATE TABLE TB_FUNCIONARIO (  
    matricula int NOT NULL PRIMARY KEY,  
    nm_funcionario varchar (50) NOT NULL,  
    cd_departamento int  
)
```



Junções

```
CREATE TABLE TB_FUNCIONARIO (  
    matricula int NOT NULL PRIMARY KEY,  
    nm_funcionario varchar (50) NOT NULL,  
    cd_departamento int  
)
```

- O atributo `cd_departamento` representa o departamento no qual o funcionário está lotado.



Junções

- Como podemos listar todos os funcionários e o nome dos seus respectivos departamentos ?

```
SELECT matricula, nm_funcionario,  
       cd_departamento  
FROM   TB_FUNCIONARIO
```

- Utilizar a tabela TB_FUNCIONARIO não é suficiente pois a mesma não traz o nome do departamento.



Junções

- Precisamos juntar (relacionar) as informações das tabelas TB_FUNCIONARIO e TB_DEPARTAMENTO.
- Como isso é feito na linguagem SQL ?
 - Selecionando informações das duas tabelas em um mesmo comando SQL e informando ao SGBD como essas tabelas estão relacionadas.



Junções

```
SELECT f.matricula,  
       f.nm_funcionario,  
       d.nm_departamento  
FROM TB_FUNCIONARIO f JOIN  
     TB_DEPARTAMENTO d  
On (f.cd_departamento =  
    d.cd_departamento)
```



Junções

- JOIN é sinônimo para INNER JOIN

```
SELECT f.matricula,  
       f.nm_funcionario,  
       d.nm_departamento  
FROM TB_FUNCIONARIO f INNER JOIN  
     TB_DEPARTAMENTO d  
On (f.cd_departamento =  
    d.cd_departamento)
```



Junções

- Podemos também utilizar filtros

```
SELECT f.matricula,  
       f.nm_funcionario,  
       d.nm_departamento  
FROM TB_FUNCIONARIO f JOIN  
     TB_DEPARTAMENTO d  
  ON (f.cd_departamento =  
      d.cd_departamento)  
WHERE d.nm_departamento = 'TI'
```



Junções e nulos

- O que acontece na consulta abaixo se algum funcionário não estiver lotado em algum departamento ?

```
SELECT f.matricula,  
       f.nm_funcionario,  
       d.nm_departamento  
FROM TB_FUNCIONARIO f INNER JOIN  
     TB_DEPARTAMENTO d  
On (f.cd_departamento =  
    d.cd_departamento)
```



Junções e nulos

- Como podemos listar todos os funcionários, independente da existência de uma lotação ?
 - Utilizando o conceito de LEFT OUTER JOIN

```
SELECT f.matricula,  
       f.nm_funcionario,  
       d.nm_departamento  
FROM TB_FUNCIONARIO f LEFT OUTER JOIN  
     TB_DEPARTAMENTO d  
On (f.cd_departamento =  
    d.cd_departamento)
```



Junções e nulos

- Como podemos listar todos os departamentos, independente da existência de um funcionário ?
 - Utilizando o conceito de RIGHT OUTER JOIN

```
SELECT f.matricula,  
       f.nm_funcionario,  
       d.nm_departamento  
FROM TB_FUNCIONARIO f RIGHT OUTER JOIN  
     TB_DEPARTAMENTO d  
On (f.cd_departamento =  
    d.cd_departamento)
```