

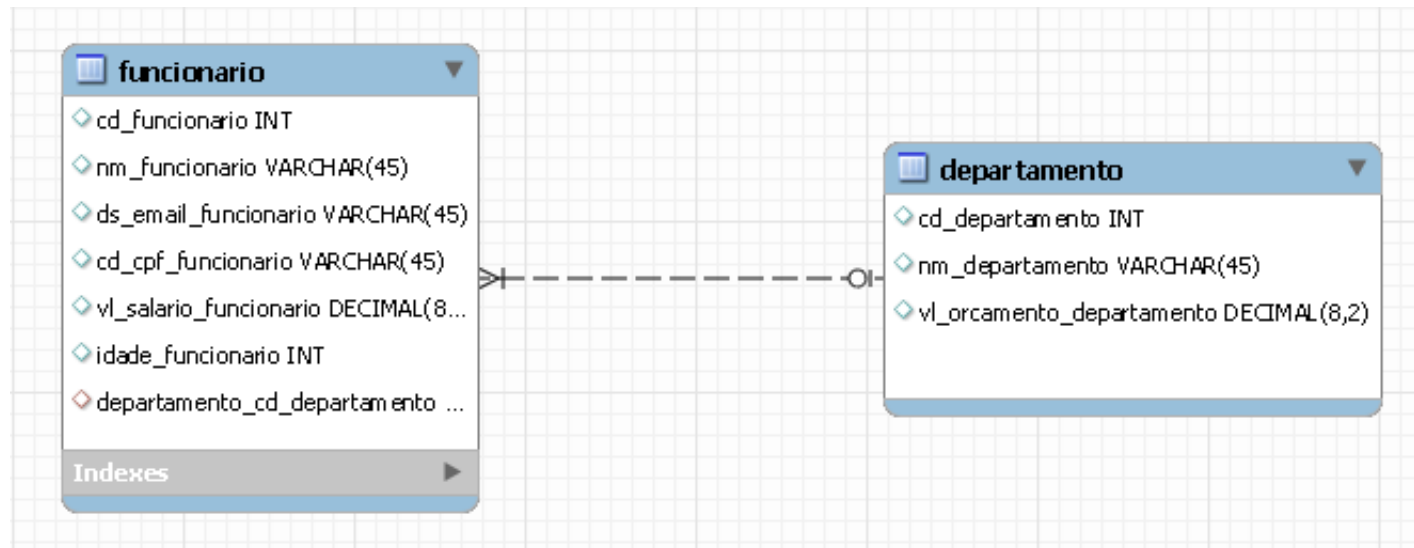
Linguagem de Programação de Banco de Dados

Junções

Carlos Arruda Baltazar

UNIP – Cidade Universitária

- Para o laboratório vamos implementar o seguinte banco de dados:



```
CREATE TABLE departamento  
(  
    cd_departamento INT NOT NULL AUTO_INCREMENT,  
    nm_departamento VARCHAR(45) NOT NULL,  
    vl_orcamento DECIMAL(8,2),    PRIMARY KEY(cd_departamento)  
);
```

```
CREATE TABLE funcionario(  
    cd_funcionario INT NOT NULL AUTO_INCREMENT,  
    nm_funcionario VARCHAR(45) NOT NULL,  
    ds_email_funcionario VARCHAR(45),  
    cd_cpf_funcionario VARCHAR(45) UNIQUE NOT NULL,  
    vl_salario_funcionario DECIMAL(8,2) NOT NULL,  
    idade_funcionario INT,  
    departamento_cd_departamento INT,  
    PRIMARY KEY (cd_funcionario),  
    FOREIGN KEY (departamento_cd_departamento) REFERENCES departamento (cd_departamento)  
);
```

- Vamos efetuar uma pequena revisão sobre as junções com objetivo de embasar os próximos tópicos desta disciplina;
- inicie inserindo os registros abaixo no banco de dados que foi criado no item anterior:

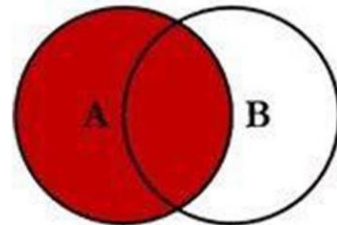
```
INSERT INTO departamento (nm_departamento, vl_orcamento) VALUES ('Qualidade', 200000);  
INSERT INTO departamento (nm_departamento, vl_orcamento) VALUES ('Processos', 300000);  
INSERT INTO departamento (nm_departamento, vl_orcamento) VALUES ('Produto', 400000);
```

```
INSERT INTO funcionario (nm_funcionario, ds_email_funcionario, cd_cpf_funcionario,  
vl_salario_funcionario, idade_funcionario, departamento_cd_departamento) VALUES ('Mario',  
'mario.quinello@docente.unip.br', 22222222222, '20000', 44,1);
```

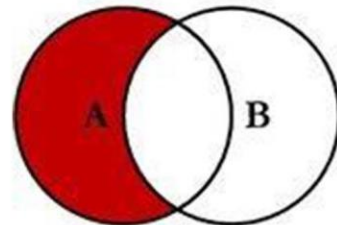
```
INSERT INTO funcionario (nm_funcionario, ds_email_funcionario, cd_cpf_funcionario,  
vl_salario_funcionario, idade_funcionario, departamento_cd_departamento) VALUES ('Mario',  
'mario.quinello@docente.unip.br', 33333333333, '20000', 44,Null);
```

```
INSERT INTO funcionario (nm_funcionario, ds_email_funcionario, cd_cpf_funcionario,  
vl_salario_funcionario, idade_funcionario, departamento_cd_departamento) VALUES ('Mario',  
'mario.quinello@docente.unip.br', 44444444444, '20000', 44,2);
```

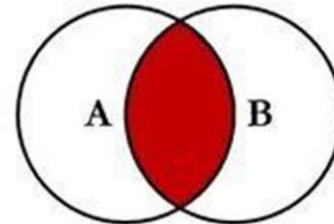
SQL JOINS



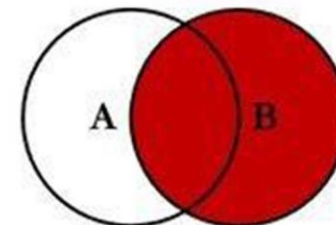
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



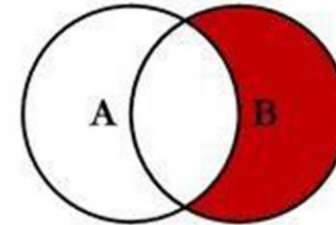
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



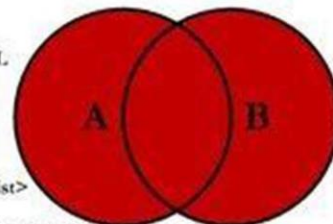
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



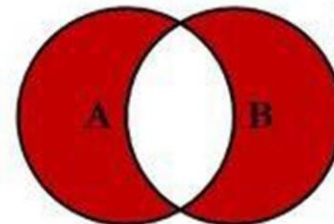
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
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
```

© C.L. Moffatt, 2008

- Agora que já temos informação em nosso banco de dados, vamos iniciar aplicando o JOIN para a Junção Interna:
 - Liste toda informação referente aos funcionários cadastrados e seus respectivos departamentos; desconsidere os registros que não possuem departamento associado.

```
SELECT f.*, c.nm_campus FROM funcionarios f JOIN departamento d  
ON f.cd_departamento = d.cd_departamentos;
```


- Agora vamos repetir a consulta, porém usando o LEFT JOIN; esta junção retorna todos os registros da tabela funcionário, incluindo aqueles que não possuem relacionamento com a tabela departamento:
 - Liste toda informação referente aos funcionários cadastrados e seus respectivos departamentos; inclua nesta consulta os registros que não possuem departamento associado.

```
SELECT f.* FROM funcionario f LEFT JOIN departamento d  
ON f.cd_departamento = d.cd_departamento;
```

Com base no modelo utilizado neste tutorial, apresente o script/comando SQL para aplicar as seguintes junções:

1. “Efetuar uma consulta que retorne o nome e departamento do funcionário; exibir o código do departamento e o código dos funcionários, somente dos funcionário que estão filiados a um departamento.”
2. “Efetuar uma consulta que retorne o nome e departamento do funcionário; este resultado deve apresentar somente os funcionários do Desenvolvimento.”
3. “Efetuar uma consulta que retorne apenas os departamentos que não possuem funcionários associados.”

```
SELECT f.nm_funcionario, d.nm_departamento, f.cd_funcionario,  
d.cd_departamento FROM funcionario f  
INNER JOIN departamento d  
ON f.departamento_cd_departamento = d.cd_departamento;
```

```
SELECT f.nm_funcionario, d.nm_departamento  
FROM funcionario f INNER JOIN departamento d  
ON f.departamento_cd_departamento = d.cd_departamento AND  
   d.nm_departamento = 'Desenvolvimento';
```

```
SELECT d.nm_departamento  
FROM funcionario f RIGHT JOIN departamento d  
ON f.departamento_cd_departamento = d.cd_departamento  
WHERE f.departamento_cd_departamento IS NULL;
```

OBRIGADO