



Microsoft®  
**SQL Server®**



## Alex Sander Resende de Deus

A 25 anos ensinando programação a jovens e adultos.

Apaixonado por tecnologia é atualmente coordenador de cursos na ETEC Albert Einstein. Na FIAP atua como professor na FIAP School, lecionando C#, SQLServer e Desenvolvimento Mobile

## Aula 05

# Chaves Estrangeiras Relacionamentos

---



- □ ●
- • • + • □

Chave estrangeira, ou **Foreign Key (FK)**, ou ainda chave externa é a chave que permite a referência a registros oriundos de outras tabelas. Ou seja, é o campo ou conjunto de campos que compõem a chave primária de uma outra tabela.

A utilização da chave estrangeira possibilita a implementação da integridade de dados diretamente no banco de dados, conhecida como integridade referencial.

Uma chave estrangeira é a representação de um relacionamento entre tabelas.

- 
- 
- + + •
- • • •
- • • • •

# Praticando no SQLServer



```
create database relacionamentos
```

```
use relacionamentos
```

```
create table cargos(  
idCargo int primary key identity (1,1),  
nomeCargo varchar(20) not null)
```

```
create table funcionarios (  
id int primary key identity(1,1),  
nome varchar(50) not null,  
idCargo int
```

```
constraint fk_funcionario_cargo foreign key (idCargo)  
references cargos(idCargo)  
)
```

— □ ●  
 • • • + • □  
 insert into cargos  
 values ('Gerente')

• ● +  
 • insert into cargos  
 values ('Vendedor')

insert into cargos  
 values ('Caixa')

insert into funcionarios  
 values ('José',1)

insert into funcionarios  
 values ('Maria',2)

insert into funcionarios  
 values ('Juliano',2)

insert into funcionarios  
 values ('Ronaldo',2)

insert into funcionarios  
 values ('Clarice',null)

•  
 □  
 + + •  
 • • • •  
 □ • • • •

# Junção de Tabelas





# 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 funcionarios.nome,cargos.nomeCargo  
from funcionarios cross join cargos
```

# INNER JOIN

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

Note que idCargo 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.

```
select funcionarios.nome,cargos.nomeCargo  
from funcionarios inner join cargos on funcionarios.idCargo=cargos.idCargo
```

# LEFT OUTER JOIN

- Se desejarmos listar todos os funcionários com seus respectivos cargos, incluindo os funcionários sem cargos 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.

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.

```
select funcionarios.nome,cargos.nomeCargo
from funcionarios left outer join cargos on funcionarios.idCargo=cargos.idCargo
```

# RIGHT OUTER JOIN

Se desejarmos listar todos os CARGOS e seus respectivos FUNCIONARIOS, incluindo os CARGOS sem FUNCIONÁRIOS, poderíamos usar a junção RIGTH OUTER JOIN

```
select funcionarios.nome,cargos.nomeCargo  
from funcionarios right outer join cargos on funcionarios.idCargo=cargos.idCargo
```

# FULL OUTER JOIN

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

```
select funcionarios.nome,cargos.nomeCargo  
from funcionarios full outer join cargos on funcionarios.idCargo=cargos.idCargo
```



# Momento Hands On



Imagine um site onde os jogadores podem criar seus personagens e conquistar itens para incrementar estes personagens. Baseado nisto, crie as tabelas capazes de armazenar estas informações.

Cadastre alguns personagens e itens.

Relacione personagens a itens e crie os selects:

- Selecione todos os personagens e seus itens
- Selecione um personagem específico e seus itens
- Selecione um item e todos os personagens que o possuam.

O Departamento de Trânsito de sua cidade precisa de um sistema que controle as multas aplicadas aos veículos. Crie um banco de dados capaz de armazenar os tipos de multas, veículos e seus proprietários bem como as infrações aplicadas aos veículos.

Crie também os selects:

- Ver todas as multas aplicadas a um veículo qualquer;
- Ver todos os veículos que foram autuados por estacionamento proibido (ou outra infração a sua escolha).



# OBRIGADO



[profalex.deus@fiap.com.br](mailto:profalex.deus@fiap.com.br)



[linkedin.com/in/alexanderresende](https://linkedin.com/in/alexanderresende)

**FIAP** MBA<sup>+</sup>

Copyright © 2019 | Professor (a) Nome do Professor  
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente  
proibido sem consentimento formal, por escrito, do professor/autor.

FIAP