

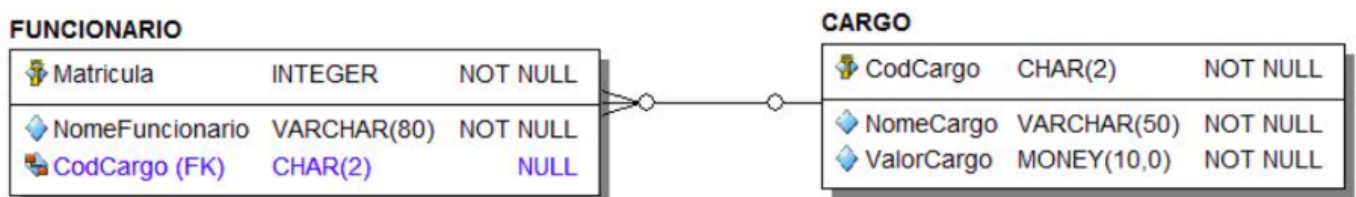
Estudo sobre Junções em SQL

Em um SGBD é comum que a informação desejada pela pessoa que utiliza a base de dados esteja dispersa em várias tabelas. Com isso, é preciso agrupar essas informações entre essas diversas tabelas de modo que a tabela resultante contenha todas as informações requisitadas.

Esses agrupamentos, no contexto de banco de dados, são as junções. Existem diversos tipos de junções, como: CROSS JOIN, INNER JOIN, LEFT OUTER JOIN, RIGHT OUTER JOIN, OUTER FULL JOIN. Será utilizado um exemplo de relação para retratar o funcionamento de cada tipo de junção.

Exemplo:

A seguinte relação entre as entidades FUNCIONARIO e CARGO permite que um funcionário possua nenhum ou vários cargos e um cargo pode ser de nenhum ou vários funcionários:



Abaixo temos um conjunto de tuplas:

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

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

Junções:

1. CROSS JOIN

Essa junção é um produto cartesiano entre as tabelas. Isto é, cada tupla à esquerda irá se agrupar com todas as tuplas da tabela à direita. Para ter acesso aos funcionários e seus cargos será necessário realizar uma junção. Uma junção do tipo CROSS JOIN entre as duas tabelas teria o seguinte resultado:

SELECT NomeFuncionario, NomeCargo FROM CARGO CROSS JOIN FUNCIONARIO;

	NomeFuncionario	NomeCargo
1	JOÃO	CAIXA
2	JOÃO	VENDEDOR
3	JOÃO	GERENTE
4	MARIA	CAIXA
5	MARIA	VENDEDOR
6	MARIA	GERENTE
7	CARLOS	CAIXA
8	CARLOS	VENDEDOR
9	CARLOS	GERENTE

Podemos notar que cada nome de funcionário da tabela FUNCIONARIO foi associado com um nome de cada carga da tabela CARGO, correspondendo ao produto cartesiano das tabelas.

2. INNER JOIN

Essa junção, diferentemente da CROSS JOIN, não gerará uma combinação de todos com todos. A junção INNER JOIN irá agrupar as tuplas que atenderem a um restrição estabelecida em seu comando.

Para poder agrupar o nome do funcionário da tabela FUNCIONARIO com seu respectivo nome do cargo da tabela CARGO, seria necessário realizar o seguinte comando:

**SELECT NomeFuncionario, NomeCargo FROM CARGO INNER JOIN FUNCIONARIO ON
(CARGO.CodCargo = FUNCIONARIO.CodCargo);**

	NomeFuncionario	NomeCargo
1	JOÃO	CAIXA
2	MARIA	VENDEDOR
3	CARLOS	CAIXA

A última parte do comando acima especifica a restrição de junção, isto é, neste caso, ele irá selecionar o nome do funcionário e o nome do cargo cujo código do cargo seja igual nas duas tabelas. Um detalhe que pode ser observado, a tupla que descreve o funcionário Tadeu não era referenciada na tabela CARGO, por isso não atendeu à restrição de junção e não foi selecionada na nova tabela criada.

3. LEFT OUTER JOIN

Essa junção, assim como a anterior, irá especificar uma restrição, porém essa restrição será somente para a tabela da direita do comando. Ou seja, o resultado dessa junção irá conter todos os dados da tabela da esquerda e os dados das tuplas da tabela da direita que obedecerem a restrição de junção.

**SELECT NomeFuncionario, NomeCargo FROM FUNCIONARIO LEFT OUTER JOIN CARGO
ON (FUNCIONARIO.CodCargo = CARGO.CodCargo);**

	nomeFuncionario	nomeCargo
1	JOÃO	CAIXA
2	MARIA	VENDEDOR
3	CARLOS	CAIXA
4	TADEU	NULL

Como resultado, a tupla do funcionário Tadeu, mesmo não atendendo à restrição da junção, aparece na tabela resultante, pois estava na tabela à esquerda. Já a tupla que detalha o cargo de Gerente, na tabela à direita (CARGO), por não atender a restrição, não aparece na tabela resultante.

4. RIGHT OUTER JOIN

Essa junção possui o mesmo funcionamento da anterior, a diferença está em qual tabela a restrição é aplicada, que neste caso, é na tabela à esquerda. Ou seja, quando esse tipo de junção é utilizado, a tabela resultante conterá todas as tuplas da tabela à direita e os dados das tuplas da tabela da esquerda que obedecerem a restrição de junção.

**SELECT NomeFuncionario, NomeCargo FROM FUNCIONARIO RIGHT OUTER JOIN
CARGO ON (FUNCIONARIO.CodCargo = CARGO.CodCargo);**

	nomeFuncionario	nomeCargo
1	JOÃO	CAIXA
2	CARLOS	CAIXA
3	MARIA	VENDEDOR
4	NULL	GERENTE

Nesta tabela resultante, é possível notar que agora o cargo de Gerente aparece, porém o funcionário Tadeu não, pois não atende à restrição de junção.

5. OUTER FULL JOIN

Essa junção é se diferencia das duas anteriores pois agora não terá um lado de respeitará a restrição. A tabela resultante irá trazer todos os dados selecionados de todas as tuplas, porém as tuplas que não possuem relação, ou seja, não obedecem à restrição de junção, irão receber o valor NULL.

**SELECT NomeFuncionario, NomeCargo FROM FUNCIONARIO OUTER FULL JOIN CARGO
ON (FUNCIONARIO.CodCargo = CARGO.CodCargo);**

	nomeFuncionario	nomeCargo
1	JOÃO	CAIXA
2	MARIA	VENDEDOR
3	CARLOS	CAIXA
4	TADEU	NULL
5	NULL	GERENTE

Podemos notar que agora todas as tuplas de FUNCIONARIO e CARGO estão descritas na nova tabela, porém na tupla resultante do funcionário Tadeu, foi atribuído o valor NULL para o nome do cargo. A mesma situação acontece com a tupla resultante do cargo Gerente.