



Junções

07 – Junções

Sand Onofre

Sand.Onofre@FaculdadeImpacta.com.br

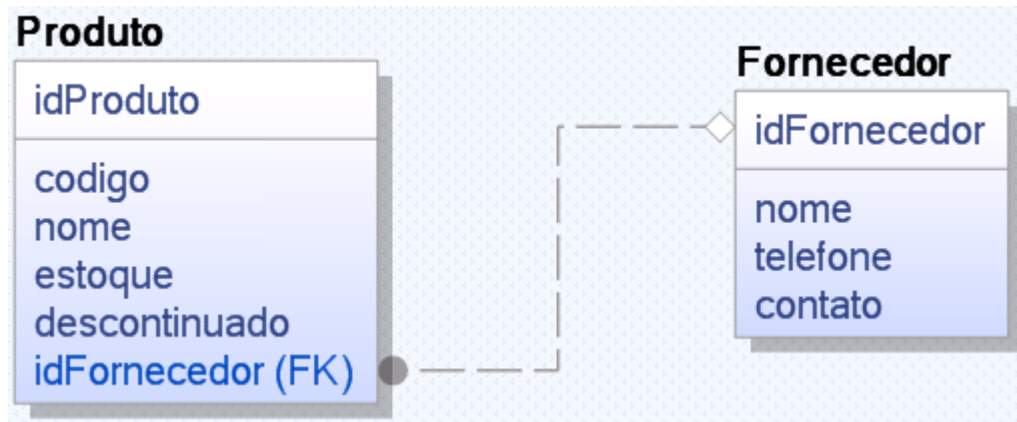
Sumário

- Data Query Language – DQL
 - Join
 - Left Join
 - Right Join
 - Full Join
 - Join com mais de duas tabelas
- Exemplos de Fixação
- Exercícios

Join

Em Bancos de Dados Relacionais, as entidades são projetadas para serem relacionadas umas com as outras. Tarefas de unir informações entre diversos objetos (*tables, views, functions, ...*) são muito comuns e necessárias. Estas junções precisam ser organizadas de forma a obtermos os dados necessários para apresentar aos usuários.

As junções entre as entidades são feitas através da relação de um ou mais atributos entre elas. Por exemplo, no DER abaixo a tabela Produto se relaciona com a tabela Fornecedor através das colunas idFornecedor.



Join

No relacionamento mostrado, dado um determinado produto, através do relacionamento entre a coluna idFornecedor, conseguimos mapear as informações do fornecedor daquele produto.

Para chegarmos a estas informações, escrevemos o SELECT da seguinte forma:

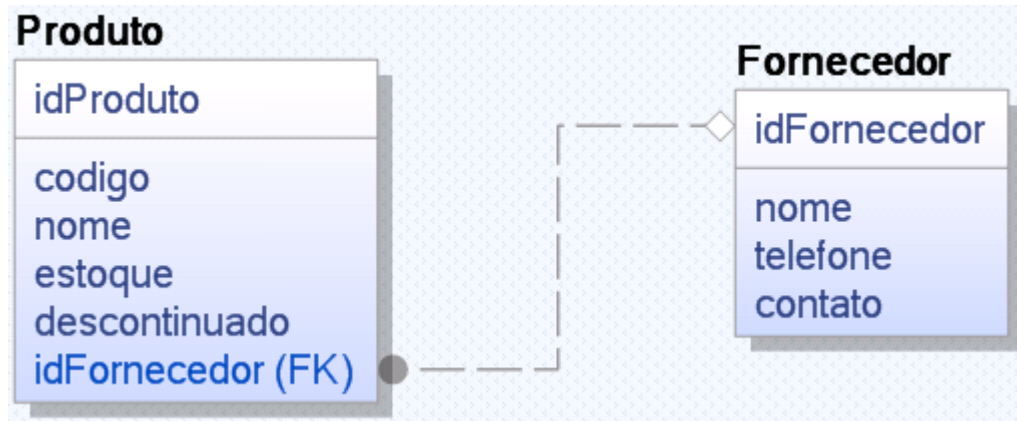
```
SELECT  <tabela 1>.<coluna 1>, ..., <tabela 1>.<coluna n>
        , <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
FROM    <tabela 1> JOIN <tabela 2>
        ON  <tabela 1>.<coluna chave> = <tabela 2>.<coluna chave>
```

A cláusula JOIN faz com que o banco de dados retorne informações das tabelas envolvidas, onde a expressão na cláusula ON for atendida, no caso acima, onde a coluna chave da tabela 1 seja igual a coluna chave da tabela 2.

Join

No exemplo Produto x Fornecedor, para trazer informações das duas tabelas podemos escrever a seguinte query:

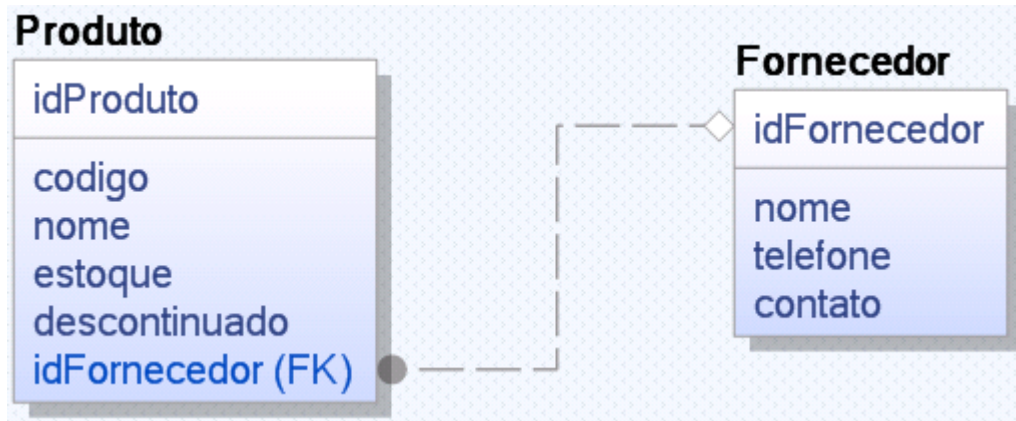
```
SELECT Produto.Codigo, Produto.Nome, Produto.Estoque
       , Fornecedor.Nome, Fornecedor.Contato, Fornecedor.Telefone
FROM   Produto JOIN Fornecedor
       ON Produto.idFornecedor = Fornecedor.idFornecedor
```



Join

A mesma consulta poderia ser simplificada e/ou melhorada no seu entendimento adotando a utilização de apelidos para tabelas e colunas. O SELECT abaixo é exatamente o mesmo do slide anterior, utilizando *Aliases* para tabelas e colunas.

```
SELECT  P.Codigo, P.Nome AS 'Nome do Produto', P.Estoque
        , F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone
FROM    Produto AS P JOIN  Fornecedor AS F
        ON  P.idFornecedor = F.idFornecedor
```



Join

Para o exemplo de dados abaixo:

Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor
1	XT890A	Asus Zenfone	5	0	3
2	RQ765B	iPhone	0	1	9
3	WD528B	Moto X	3	0	2
4	TF897A	Xperia	7	0	1
5	RF212B	Moto Maxx	2	0	2

Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

```

SELECT  P.Codigo, P.Nome AS 'Nome do Produto', P.Estoque
        , F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone
FROM    Produto AS P JOIN  Fornecedor AS F
        ON    P.idFornecedor = F.idFornecedor
  
```

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120
WD528B	Moto X	3	Motorola	Cristina	7987-9900
TF897A	Xperia	7	Sony	Allan	8498-8732
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900

Left Join

Também podemos alterar as características destas junções para mostrarmos não somente os dados que são encontrados entre os relacionamentos, mas “forçar” que registros, mesmo não possuindo relacionamento, sejam apresentados mesmo assim.

Observando a mesma modelagem mostrada anteriormente, para gerar uma lista com TODOS os produtos, mesmo que não haja relação entre as duas tabelas, podemos substituir a cláusula JOIN por LEFT JOIN.

Ou seja, para gerar uma consulta que apresente OBRIGATORIAMENTE todos os produtos, mesmo aqueles onde não encontramos relação na tabela de fornecedores, poderíamos utilizar a cláusula LEFT JOIN, que faz com que todos os registros do objeto (tabela) do lado ESQUERDO da consulta seja mostrada.

Left Join

Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor
1	XT890A	Asus Zenfone	5	0	3
2	RQ765B	iPhone	0	1	9
3	WD528B	Moto X	3	0	2
4	TF897A	Xperia	7	0	1
5	RF212B	Moto Maxx	2	0	2

Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

```

SELECT  P.Codigo, P.Nome AS 'Nome do Produto', P.Estoque
        , F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone
FROM    Produto AS P LEFT JOIN Fornecedor AS F
        ON  P.idFornecedor = F.idFornecedor
    
```

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120
RQ765B	iPhone	0	NULL	NULL	NULL
WD528B	Moto X	3	Motorola	Cristina	7987-9900
TF897A	Xperia	7	Sony	Allan	8498-8732
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900

Right Join

Podemos fazer o inverso, ou seja, gerar uma lista com OBRIGATORIAMENTE todos os fornecedores, mesmo aqueles onde não encontramos relação na tabela de produtos, bastando alterar para cláusula RIGHT JOIN, que faz com que todos os registros do lado DIREITO da consulta seja mostrada.

O controle do modo que queremos o join é realizado apenas pelo posicionamento da tabela ao lado direito ou esquerdo do SELECT ou simplesmente trocando a cláusula de RIGHT JOIN para LEFT JOIN.

Por exemplo, sejam as tabelas TB1 e TB2, relacionadas pelas colunas C1 e C2, se quisermos gerar uma lista com TODOS os registros da tabela TB1, os SELECTs abaixo geram essa lista:

SELECT ... FROM TB1 LEFT JOIN TB2 ON TB1.C1 = TB2.C2

OU

SELECT ... FROM TB2 RIGHT JOIN TB1 ON TB2.C2 = TB1.C1

Simples assim !

Right Join

Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor
1	XT890A	Asus Zenfone	5	0	3
2	RQ765B	iPhone	0	1	9
3	WD528B	Moto X	3	0	2
4	TF897A	Xperia	7	0	1
5	RF212B	Moto Maxx	2	0	2

Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

```

SELECT  P.Codigo, P.Nome AS 'Nome do Produto', P.Estoque
        , F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone
FROM    Produto AS P RIGHT JOIN Fornecedor AS F
        ON  P.idFornecedor = F.idFornecedor
    
```

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120
WD528B	Moto X	3	Motorola	Cristina	7987-9900
TF897A	Xperia	7	Sony	Allan	8498-8732
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900
NULL	NULL	NULL	Nokia	Fábio	6755-5656

Join com mais de duas tabelas

Conforme visto, junções entre tabelas são operações triviais em bancos de dados relacionais. Estender o mesmo conceito a mais de duas tabelas não é diferente. No dia-a-dia faremos junções entre três, quatro, cinco ou mais objetos.

Apesar da complexidade aumentar, se mantermos uma lógica em mente, não teremos problemas em estender o mesmo conceito para junções com dez ou mais objetos.

A ordem que colocarmos as tabelas ou as cláusulas JOIN, LEFT JOIN, RIGHT JOIN, ocasionam diferenças na geração dos resultados e conforme já visto, devemos estar atentos a isso.

Mas se tratarmos a consulta, envolvendo três ou mais tabelas como conjunto de resultados, não teremos maiores problemas na construção das queries.

Join com mais de duas tabelas

Mantenha em mente cada junção entre uma tabela e outra, como um conjunto de resultados e faça as junções através deles. No script abaixo perceba que, independente das junções que tivermos, se tratarmos a cada duas tabelas como um conjunto e a próxima junção como um novo conjunto o entendimento será mais fácil.

Quando aplicamos JOIN entre as tabelas, a ordem em que estão, não irão afetar os resultados, assim podemos inverter as ordens das tabelas que o efeito será o mesmo:

```
SELECT <tabela 1>.<coluna 1>, ..., <tabela 1>.<coluna n>
      , <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
      , <tabela 3>.<coluna 1>, ..., <tabela 3>.<coluna n>
FROM   <tabela 1> JOIN <tabela 2>
      ON  <tabela 1>.<coluna chave> = <tabela 2>.<coluna chave>
      JOIN <tabela 2>
      ON  <tabela 2>.<coluna chave> = <tabela 3>.<coluna chave>
```

Join com mais de duas tabelas

Se alternarmos os joins entre JOIN, LEFT JOIN ou RIGHT JOIN devemos ter cuidado, pois o resultado final será afetado. Veja o exemplo abaixo e perceba como há diferença entre as junções:

1º. Conjunto de Resultados



```
SELECT <tabela 1>.<coluna 1>, ..., <tabela 1>.<coluna n>
      , <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
      , <tabela 3>.<coluna 1>, ..., <tabela 3>.<coluna n>
```

```
FROM <tabela 1> LEFT JOIN <tabela 2>
      ON <tabela 1>.<coluna chave> = <tabela 2>.<coluna chave>
      JOIN <tabela 3>
      ON <tabela 1>.<coluna chave> = <tabela 3>.<coluna chave>
```

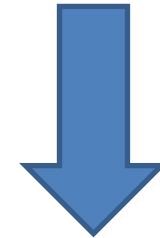


2º. Conjunto de Resultados

Join com mais de duas tabelas

Que é totalmente diferente se invertermos os joins conforme mostrado abaixo:

1º. Conjunto de Resultados



```
SELECT <tabela 1>.<coluna 1>, ..., <tabela 1>.<coluna n>
      , <tabela 2>.<coluna 1>, ..., <tabela 2>.<coluna n>
      , <tabela 3>.<coluna 1>, ..., <tabela 3>.<coluna n>
```

```
FROM <tabela 1> JOIN <tabela 2>
      ON <tabela 1>.<coluna chave> = <tabela 2>.<coluna chave>
      LEFT JOIN <tabela 3>
      ON <tabela 1>.<coluna chave> = <tabela 3>.<coluna chave>
```



2º. Conjunto de Resultados

Join com mais de duas tabelas

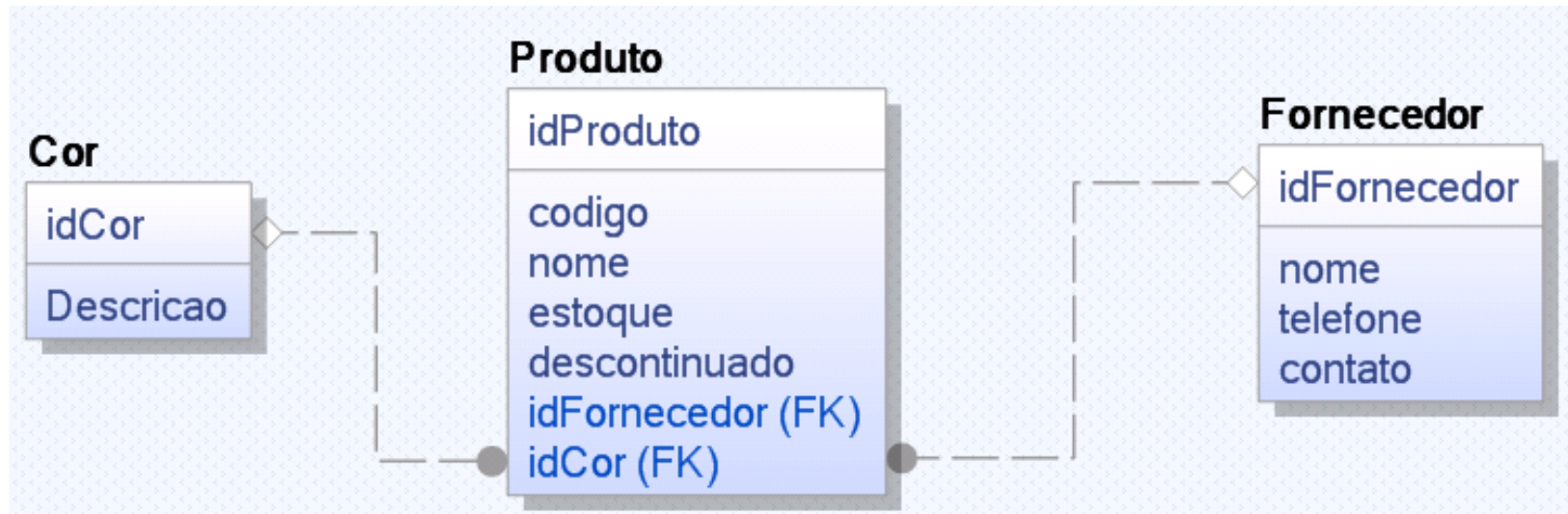
A mesma diferença será estendida se trocarmos pela cláusula RIGHT JOIN, ou seja, quando temos apenas a cláusula JOIN, não precisamos nos preocupar com a ordem em que as tabelas são posicionadas.

Mas se em nossas junções tivermos qualquer tipo de LEFT JOIN ou RIGHT JOIN, temos que ter atenção no posicionamento e ordem em que as tabelas estarão, assim como cada cláusula de junção.

A forma mais fácil de projetarmos o resultado final é combinando as junções por conjunto de resultado, ou seja, obter o primeiro conjunto de resultado através da junção entre duas tabelas. A partir deste conjunto de resultados, fazer nova junção com outra tabela, obtendo assim o segundo conjunto de resultados e assim por diante.

Exemplos de Fixação

Para ilustrarmos ainda mais o conceito de junção entre mais de duas tabelas, veja o seguinte DER:



Exemplos de Fixação

Gerando a extração com JOIN entre as tabelas:

Cor		Produto					
idCor	descricao	idProduto	codigo	nome	estoque	descontinuado	idFornecedor
1	Branco	1	XT890A	Asus Zenfone	5	0	3
2	Preto	2	RQ765B	iPhone	0	1	9
3	Azul	3	WD528B	Moto X	3	0	2
4	Vermelho	4	TF897A	Xperia	7	0	1
5	Amarelo	5	RF212B	Moto Maxx	2	0	2

Fornecedor			
idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

```

SELECT P.Codigo, P.Nome AS 'Nome do Produto', P.Estoque
, F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone
, C.Descricao AS 'Cor'
FROM Produto AS P JOIN Cor AS C ON P.idCor = C.idCor
JOIN Fornecedor AS F ON C.idFornecedor = F.idFornecedor

```

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco

[illegible]

Exemplos de Fixação

Trocando o RIGHT pelo LEFT o resultado é alterado:

Cor		Produto						
idCor	descricao	idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	Branco	1	XT890A	Asus Zenfone	5	0	3	4
2	Preto	2	RQ765B	iPhone	0	1	9	4
3	Azul	3	WD528B	Moto X	3	0	2	7
4	Vermelho	4	TF897A	Xperia	7	0	1	1
5	Amarelo	5	RF212B	Moto Maxx	2	0	2	1

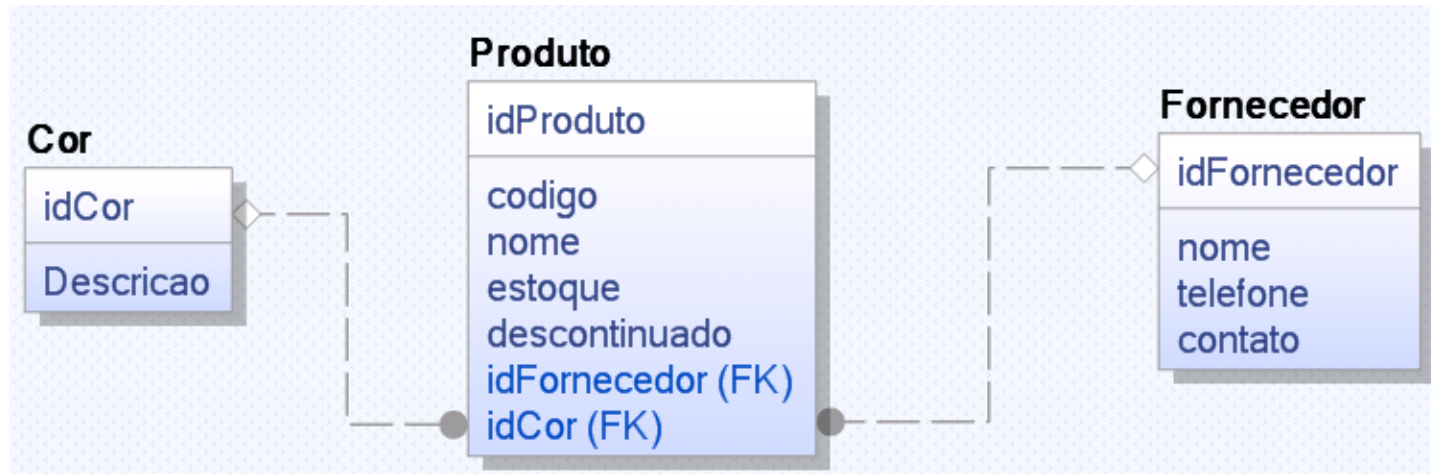
Fornecedor			
idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

```

SELECT  P.Codigo, P.Nome AS 'Nome do Produto', P.Estoque
        , F.Nome AS 'Nome do Fornecedor', F.Contato, F.Telefone
        , C.Descricao AS 'Cor'
FROM      Produto AS P JOIN  Fornecedor AS F  ON  P.idFornecedor = F.idFornecedor
        LEFT JOIN  Cor AS C   ON C.idCor = P.idCor
    
```

Codigo	Nome do Produto	Estoque	Nome do Fornecedor	Contato	Telefone	Cor
XT890A	Asus Zenfone	5	Asus	Felipe	5476-1120	Vermelho
WD528B	Moto X	3	Motorola	Cristina	7987-9900	NULL
TF897A	Xperia	7	Sony	Allan	8498-8732	Branco
RF212B	Moto Maxx	2	Motorola	Cristina	7987-9900	Branco

Exercícios



1. Crie o modelo acima com as seguintes características:
 - Assuma que as tabelas terão somente 10 cores, 50 produtos e 20 fornecedores
 - Colunas Ids serão primary keys e devem ser auto-numeradas
 - Coluna descontinuado possui apenas valores 0 e 1. Já estoque, poderemos ter até 5.000 unidades
 - Colunas de nome e de descrição terão até 50 caracteres, já a coluna de código terá exatos 6 caracteres
 - Crie todas as chaves primárias, estrangeiras e chaves únicas para qualquer coluna de descrição, de nome e de código do produto

Exercícios

Cor

idCor	descricao
1	Branco
2	Preto
3	Azul
4	Vermelho
5	Amarelo

Produto

idProduto	codigo	nome	estoque	descontinuado	idFornecedor	idCor
1	XT890A	Asus Zenfone	5	0	3	4
2	RQ765B	Nokia	21	1	4	4
3	WD528B	Moto X	3	0	2	5
4	TF897A	Xperia	7	1	1	1
5	RF212B	Moto Maxx	2	0	2	1

Fornecedor

idFornecedor	nome	telefone	contato
1	Sony	8498-8732	Allan
2	Motorola	7987-9900	Cristina
3	Asus	5476-1120	Felipe
4	Nokia	6755-5656	Fábio

2. Preencha as informações exatamente como estão acima.
3. Repita as inserções dos mesmos produtos, variando as cores e estoque, de forma que tenhamos ao menos **15 produtos** no final das inserções

Exercícios

4. Traga o código do produto, nome do produto, nome do fornecedor, estoque para todos os produtos onde forem encontrado relação. Coloque nomes de colunas apropriados o correto entendimento da informação.
5. Baseado na Query 4, retorne as mesmas informações para produtos descontinuados.
6. Baseado na Query 4, retorne as mesmas informações para produtos com estoque entre 7 e 21 unidades, que ainda tenham produção.
7. Adicione a Query 4 o nome da cor de cada produto.
8. Baseado na Query 7, traga apenas produtos de cores BRANCO, PRETO ou VERMELHO (utilize a coluna Descrição e não idCor)
9. Baseado na Query 7, traga quaisquer produtos onde a cor não seja AZUL (utilize a coluna **Descrição** e não idCor)
10. Baseado na Query 7, adicione a coluna contato do fornecedor e retorne apenas contatos que iniciem pela letra F (utilizar o predicado LIKE).
11. Baseado na Query 10, adicione a coluna telefone do fornecedor e retorne apenas contatos que terminem com dígito 0 (utilizar o predicado LIKE).



Obrigado !

Sand Onofre
Sand.Onofre@FaculdadeImpacta.com.br