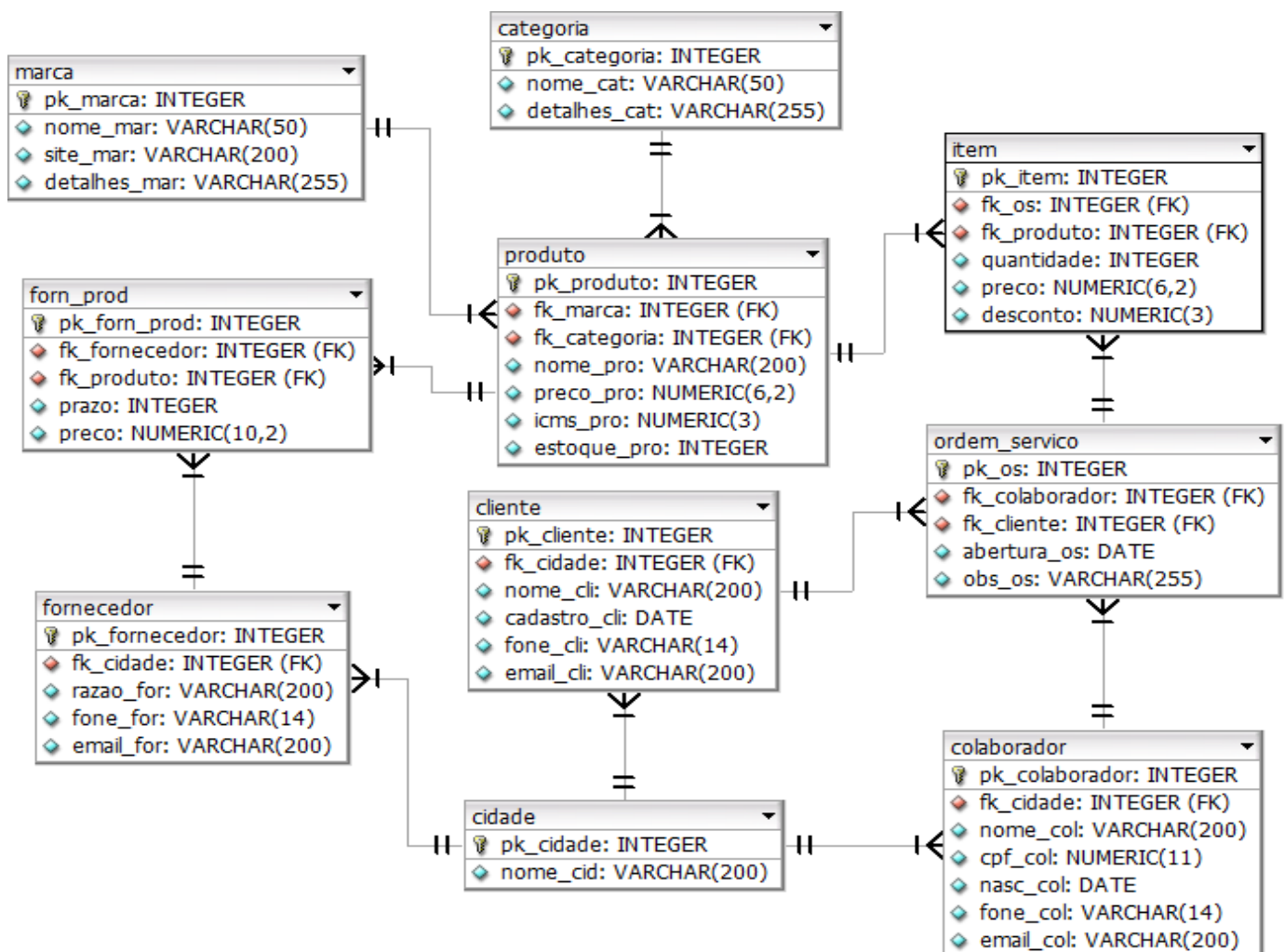


Resumo do comando JOIN:

O comando JOIN no SQL é usado para combinar linhas de duas ou mais tabelas com base em uma condição relacionada entre elas. Essa condição é especificada no bloco WHERE da consulta SQL e geralmente envolve chaves primárias e estrangeiras. O comando JOIN é fundamental em bancos de dados relacionais para consultar dados que estão distribuídos entre diferentes tabelas.

Modelo de Banco de dados



Estrutura do comando JOIN

1) Liste alfabeticamente o código e nome dos clientes de Joinville.

```
SELECT cli.pk_cliente, cli.nome_cli  
FROM cidade cid  
JOIN cliente cli  
ON cid.pk_cidade = cli.fk_cidade  
WHERE cid.nome_cid = 'Joinville'  
ORDER BY cli.nome_cli;
```

→ **SELECT** cli.pk_cliente, cli.nome_cli

-> Você sempre começará, ele servirá para selecionar o que você deseja mostrar, por exemplo no comando ele irá mostrar o pk do cliente e o nome do cliente.

-> Esse cli antes do pk_cliente, é apenas um apelido que será usado, porque usar ele, ele é usado pois se houvesse um campo/atributo com o mesmo nome, ele iria diferenciar ele.

→ **FROM** cidade cid

-> O **FROM** indica onde irá começar a busca sendo também a tabela inicial a ser juntada, nesse exemplo ele irá começar pela tabela cidade, junto com isso ele define o apelido daquela tabela, sendo o apelido cli.

→ **JOIN** cliente cli

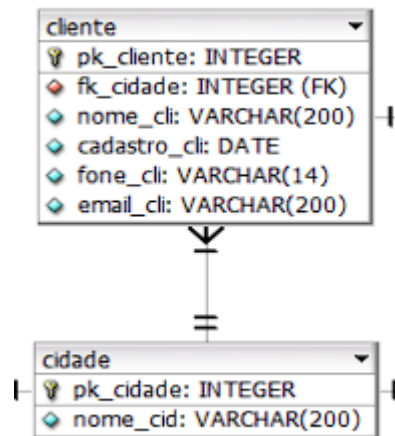
-> Aqui você está definindo a segunda tabela a ser juntada, colocando também um apelido para aquela tabela.

→ **ON** cid.pk_cidade = cli.fk_cidade

-> Aqui irá ocorrer a "junção" das tabelas por meios dos PK e FK, sempre que você quiser fazer a junção de duas tabelas você tem que definir isso.

****IMPORTANTE****

-> Você junta as tabelas por caminhos, como assim, o relacionamento entre elas que você irá juntar, pense nisso como um caminho para chegar na tabela que você quer juntar (se você não entendeu me chama que eu tento explicar pessoalmente).



→ **WHERE** cid.nome_cid = 'Joinville'

-> Aqui vai ocorrer o filtro daquela do que irá aparecer no select, sempre começando com o prefixo **WHERE**, aqui vai aparecer só os clientes que são da cidade Joinville. (Você pode colocar mais de um filtro).

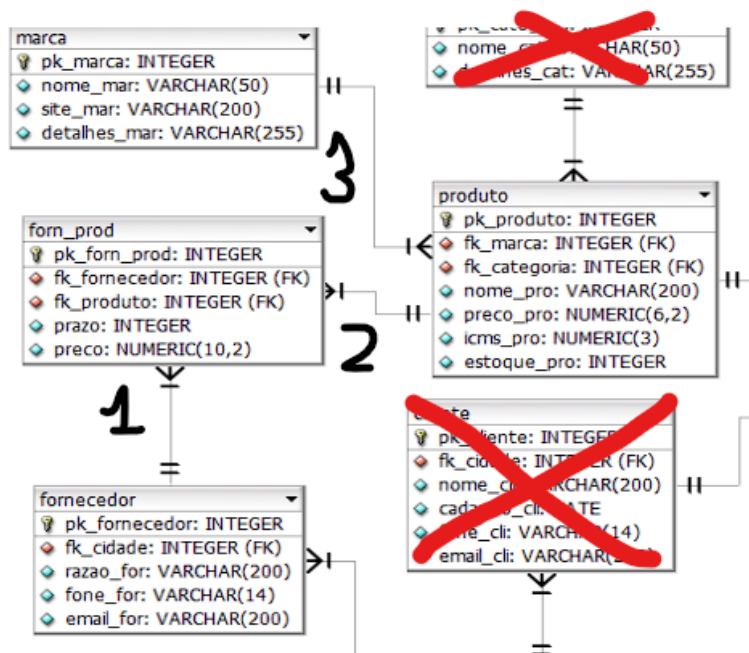
→ **ORDER BY** cli.nome_cli;

-> Aqui é bem simples, vai ordenar em ordem alfabética a pesquisa baseando no nome do cliente.

Mais de um caminho

4) Liste alfabeticamente apenas o nome dos fornecedores que vendem produtos da marca Makita.

```
SELECT fo.razao_for
FROM marca m
JOIN produto p
ON m.pk_marca = p.fk_marca
JOIN forn_prod fp
ON p.pk_produto = fp.fk_produto
JOIN fornecedor fo
ON fo.pk_fornecedor = fp.fk_fornecedor
WHERE (m.nome_mar = 'Makita')
ORDER BY fo.razao_for;
```



→ No enunciado ele pede o nome dos fornecedores que vendem produtos da marca Makita, para tu conseguir mostrar os fornecedores, precisa passar pela TABELA produto e pela TABELA marca, é como PODEM VER elas não estão sendo ligadas diretamente, por isso que ter que fazer mais "caminhos".

→ Existem 3 ligações com 4 tabelas, então vai ter que ser feito 3 JOIN, pense em como um caminho de rato (meus exemplos são os melhores KKK).

```

FROM marca m
JOIN produto p
ON m.pk_marca = p.fk_marca
JOIN forn_prod fp
ON p.pk_produto = fp.fk_produto
JOIN fornecedor fo
ON fo.pk_fornecedor = fp.fk_fornecedor
  
```

→ O "caminho" vai começar na tabela marca, e assim indo até a tabela fornecedor, sempre terminando a ligação com a junção do PK com FK.

Mais de um filtro

3) Liste o código e o nome dos produtos da marca Bosch ou Makita com preços entre 50 e 100 reais.

```
SELECT p.pk_produto, p.nome_pro  
FROM marca m  
JOIN produto p  
ON m.pk_marca = p.fk_marca  
WHERE (m.nome_mar = 'Bosch' OR m.nome_mar = 'Makita')  
AND (p.preco_pro >= 50 AND p.preco_pro <= 100);
```

→ No enunciado ele pede dois filtros para a pesquisa, os, que os produtos sejam das marcas Bosch ou Makita **É** tenham um valor entre 50 e 100 reais.

→ O primeiro filtro você sempre irá começar com **WHERE**, é agora deixará a lógica do filtro entre (), se não deixa vai dar **ERRADO**.

→ O segundo filtro agora você vai começar com o **AND**, é deixando a lógica entre () também.

→ Se o enunciado pedir mais de 2 filtros, você irá repetir os passos, começando o 3 filtro com **AND** e colocando a lógica entre ().

Tipos de filtros

Alguns tipos de filtros que podem aparecer.

Filtro data

8) Liste o número de todas as ordens de serviço **emitidas em 2000** que contenham produtos com preço faturado acima de 500 reais.

```
WHERE (os.abertura_os >= '2000-01-01' AND os.abertura_os <= '2000-12-31')
```

Ou

10) Liste alfabeticamente o nome dos colaboradores que **nasceram entre 1980 e 1990;**

```
WHERE (nasc_col >= '1980-01-01' AND nasc_col <= '1990-12-31')
```

Filtro cidade

14) Liste alfabeticamente a razão social e e-mail dos **fornecedores de Sao Paulo, Rio de Janeiro ou Natal. Liste inclusive, o nome da cidade.**

WHERE (cid.nome_cid = 'Sao Paulo' **OR** cid.nome_cid = 'Rio de Janeiro' **OR** cid.nome_cid = 'Natal')

Filtro não cidade

19) Liste alfabeticamente o nome dos clientes que **não residam em Joinville, Sao Paulo, Blumenau ou Rio de Janeiro;**

WHERE (cid.nome_cid != 'Joinville' **AND** cid.nome_cid != 'Sao Paulo' **AND** cid.nome_cid != 'Blumenau' **AND** cid.nome_cid != 'Rio de Janeiro')

!= → Significa Diferente de

Filtro Simples

7) Liste alfabeticamente o nome do produto, sua marca e categoria, **mas apenas os da marca Gong ou Makita que sejam da categoria motosserras.**

WHERE (mar.nome_mar = 'Gong' **OR** mar.nome_mar = 'Makita')

AND (cat.pk_categoria = 8)

Espero que tenha ajudado