

# Banco de Dados : Views

Prof. Márcio Funes



## Plano de aula

Introdução às Views

Criação de Views

Exercícios

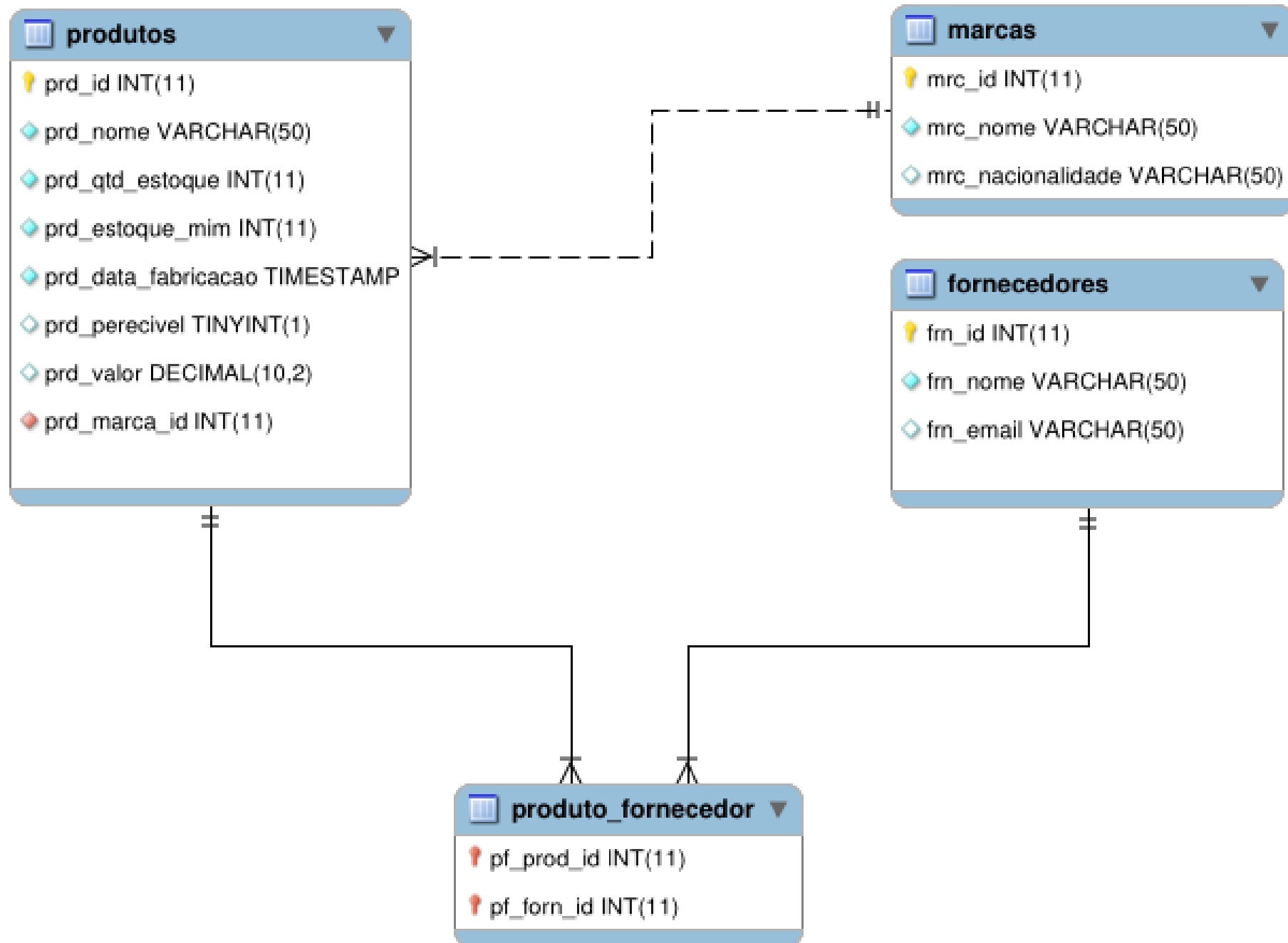
# Introdução

## O contexto

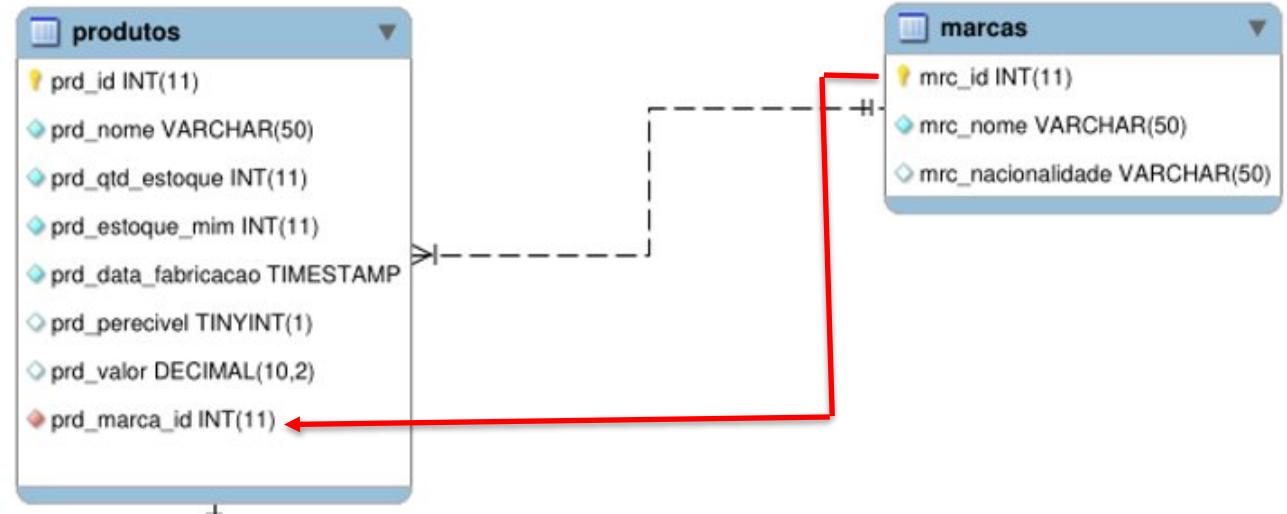
Um SGBDR deve pode conter várias tabelas que se relacionam. Com isso, algumas consultas podem se tornar **complexas e extensas**.

A linguagem **SQL** fornece um mecanismo para criar **visões dos dados**, como um forma de visualizá-los por uma certa perspectiva e com isso para facilitar o processo de consultas

Observe o modelo e script a seguir ...

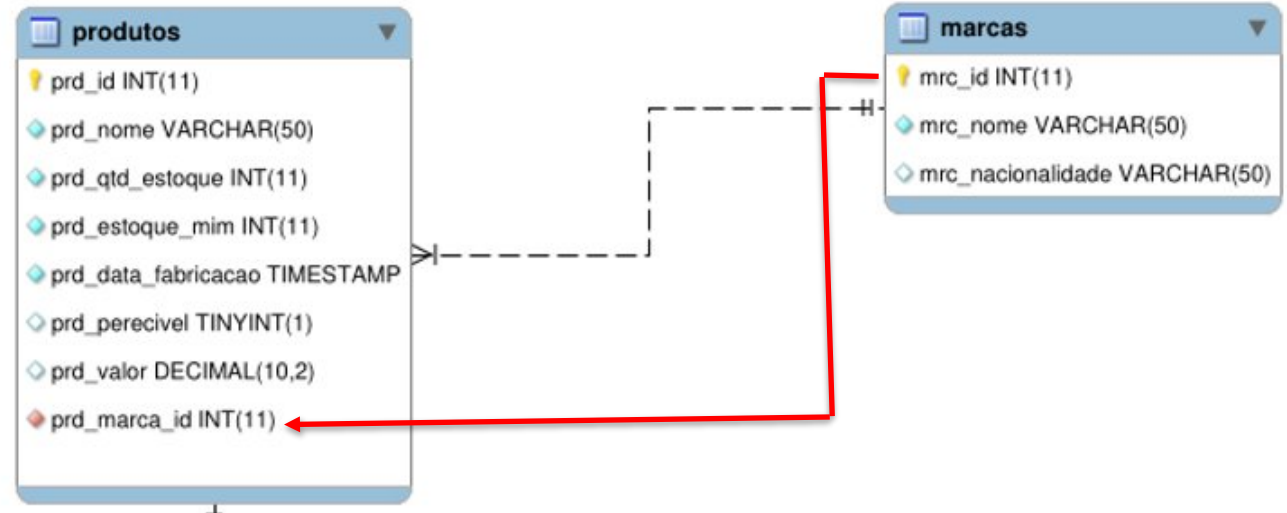


# Modelo Lógico -> Modelo Físico



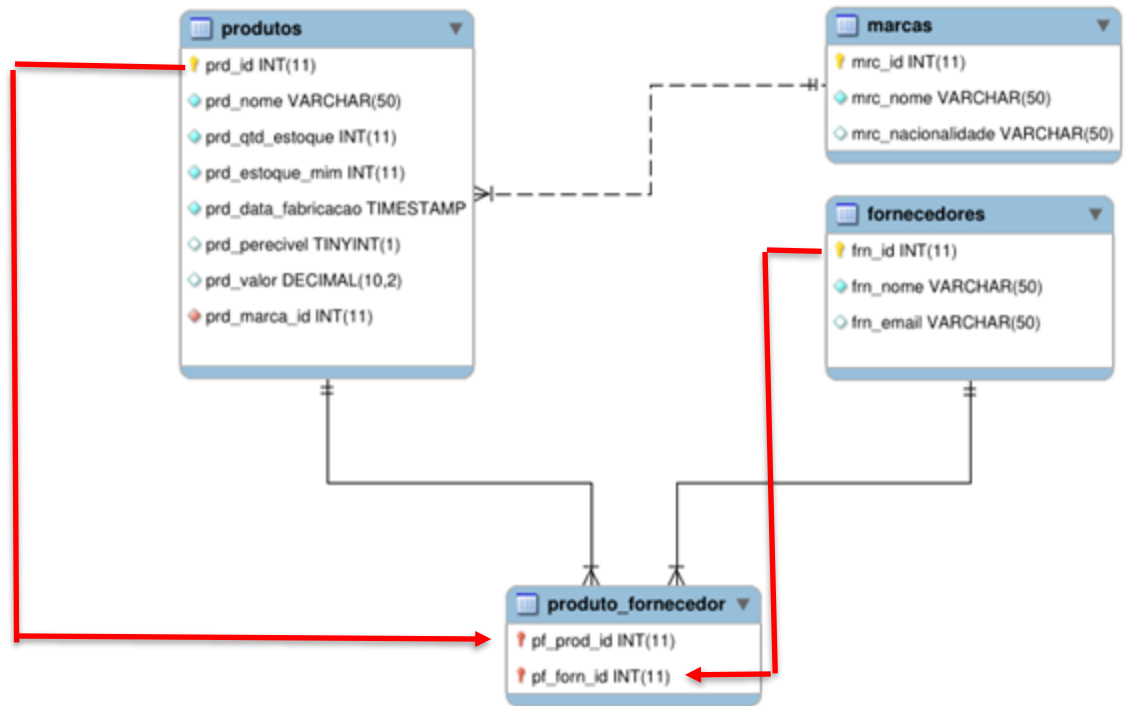
```
create table marcas (  
    marca_id          int          auto_increment    primary key,  
    marca_nome        varchar(50)  not null,  
    marca_origem      varchar(50)  
);
```

# Modelo Lógico -> Modelo Físico



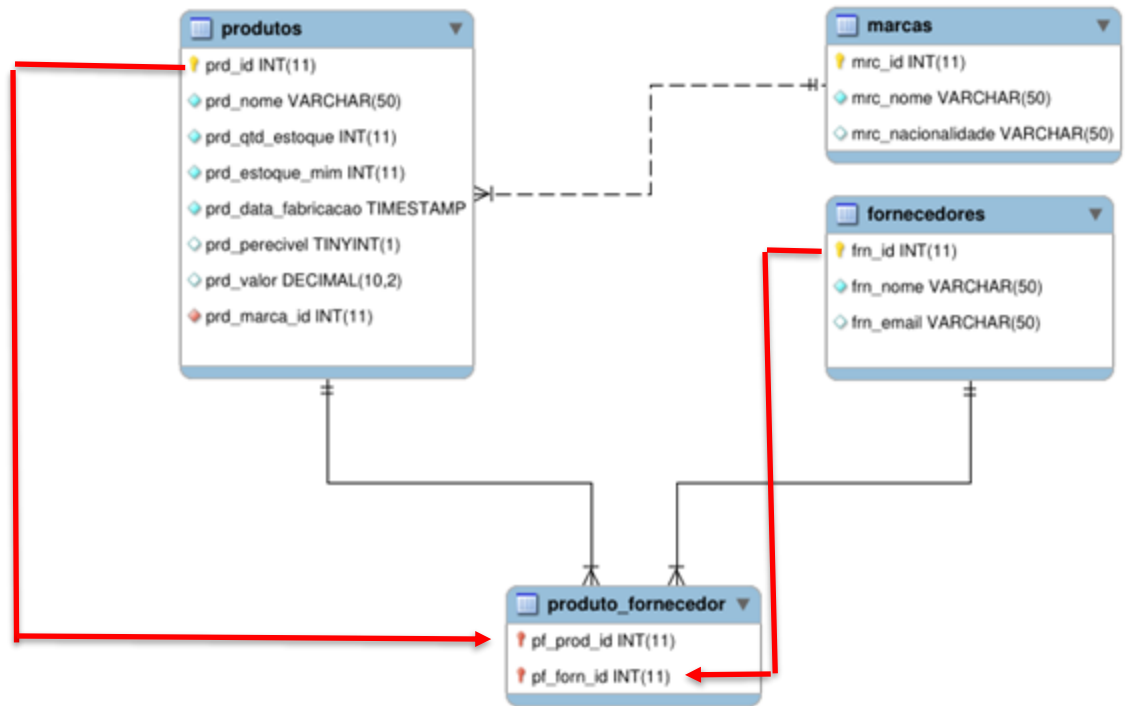
```
create table produtos (  
    prod_id            int            auto_increment    primary key,  
    prod_nome          varchar(50)    not null,  
    prod_qtd_estoque   int            not null          default 0,  
    prod_estoque_mim   int            not null          default 0,  
    prod_data_fabricacao timestamp    default now(),  
    prod_perecivel     boolean,  
    prod_valor         decimal(10,2),  
  
    marca_id           int,  
    constraint fk_marcas foreign key(marca_id) references marcas(marca_id)  
);
```

# O contexto



```
create table fornecedores (  
    forn_id          int          auto_increment primary key,  
    forn_nome        varchar(50)  not null,  
    forn_email       varchar(50)  
);
```

## O contexto



```
create table produto_fornecedor (  
    prod_id          int          not null    references produtos(prod_id),  
    forn_id          int          not null    references fornecedores(forn_id),  
  
    primary key (prod_id, forn_id),  
    constraint fk_produto foreign key(prod_id) references produtos(prod_id),  
    constraint fk_fornecedor foreign key(forn_id) references fornecedores(forn_id)  
);
```



## O contexto

Considere a necessidade de criar um relatório que mostra os 10 produtos mais caros ordenados pelo preço em ordem decrescente.

Nesse caso, o seguinte SELECT seleciona os dados requeridos.

```
select prod_id, prod_nome, prod_valor  
from produtos  
order by prod_valor desc  
limit 10;
```

## Criando Views (Visões)

## Criando Views (Visões)

Podemos entender uma **view** como uma tabela virtual baseada no resultado de um comando **select**.

O comando create view é usado para criar visões no banco de dados.

Sintaxe

```
CREATE VIEW nome_view AS  
SELECT ...
```

## Criando Views (Visões)

Podemos entender uma **view** como uma tabela virtual baseada no resultado de um comando **select**.

O comando `create view` é usado para criar visões no banco de dados.

Exemplo

```
create view Top10MaisCaros as
select prod_id, prod_nome, prod_valor
from produtos
order by prod_valor desc
limit 10;
```

## Criando Views (Visões)

Uma vez criada, ela pode ser utilizada em consultas como uma tabela, por exemplo:

```
select * from Top10MaisCaros;
```

	prod_id	prod_nome	prod_valor
►	7	ERP	937.50
	11	IOS	756.50
	38	maça	716.80
	33	mesa	632.50
	29	Placa de Vídeo	612.50
	26	Módulo de memória	612.00
	37	pera	612.00
	10	Windows	532.00
	25	Módulo de memória	512.50
	39	banana	512.00

## Criando Views (Visões)

Crie uma visão onde os produtos e as marcas são integrados, você pode criar a seguinte view:

```
create or replace view produtos_marcas as
select
    prod_nome 'Nome do Produto',
    marca_nome 'Marca',
    prod_valor 'Preço',
    prod_qtd_estoque 'Estoque',
    case
        when prod_perecivel = false then 'NÃO'
        when prod_perecivel = true then 'SIM'
    end
    'Perecivel'
from
    produtos left join marcas
        on produtos.marca_id = marcas.marca_id
order by prod_nome;
```

## Criando Views (Visões)

Uma vez criada, ela pode ser utilizada em consultas como uma tabela, por exemplo:

```
select * from produtos_marcas where Preço < 100;
```

	Nome do Produto	Marca	Preço	Estoque	Perecível
►	borracha	Faber Castel	4.20	2907	NÃO
	borracha	Labra	2.00	5408	NÃO
	caderno	Faber Castel	22.50	7004	NÃO
	caneta	Faber Castel	11.00	8030	NÃO
	CD	Multilaser	12.50	8080	NÃO
	Fonte de Energia	HP	12.50	4054	NÃO
	lapis	Faber Castel	2.50	4502	NÃO
	lapis	Labra	14.00	8800	NÃO

## Entendendo as Views

- Uma **view** contém linhas e colunas exatamente como uma tabela real.
- Os campos de uma **view** são normalmente campos de uma ou mais tabelas reais.
- As linhas e colunas da **view** são extraídas de outras e tabelas. Portanto, **não ocupam espaço** no banco de dados.
- Uma view pode conter cláusulas como **WHERE, JOIN, ORDER BY** e em seguida usar essa visualização **como uma tabela simples**.
- Uma **view sempre apresenta os dados atualizados**. Os dados são **selecionados a cada execução**. Apenas a definição dos comandos é mantida.



**Para saber mais...**

<https://www.devmedia.com.br/conceitos-e-criacao-de-views-no-sql-server/22390>

[https://www.w3schools.com/sql/sql\\_view.asp](https://www.w3schools.com/sql/sql_view.asp)

<https://www.geeksforgeeks.org/sql-views/>

## Exercícios

## Exercícios

- Crie uma view que mostra todos os produtos e suas respectivas marcas.
- Crie uma view que mostra todos os produtos e seus respectivos fornecedores.
- Crie uma view que mostra todos os produtos e seus respectivos fornecedores e marcas.
- Crie uma view que mostra todos os produtos com estoque abaixo do mínimo.

## Exercícios

- Crie uma view que mostra todos os produtos e suas respectivas marcas.
- Crie uma view que mostra todos os produtos e seus respectivos fornecedores.
- Crie uma view que mostra todos os produtos e seus respectivos fornecedores e marcas.
- Crie uma view que mostra todos os produtos com estoque abaixo do mínimo.
- Adicione o campo data de validade. Insira novos produtos com essa informação.
- Crie uma view que mostra todos os produtos e suas respectivas marcas com validade vencida.
- (Desafio) Selecionar os produtos com preço acima da média.