

Bancos de Dados

Roteiro de Exercícios 1

Versão: 0.4

Data: nov/2024

Objetivos:

- O objetivo fundamental deste material de apoio complementar é fornecer uma oportunidade adicional para o aluno praticar os conceitos estudados em sala de aula.
- Os exercícios estão divididos em ordem crescente de dificuldade e foram distribuídos em seções.
- Para melhor aproveitamento do material fornecido, o aluno deve resolver os exercícios propostos em casa e trazer os resultados obtidos para serem discutidos em sala de aula.

Orientações iniciais:

- Este material foi criado considerando as vendas em uma loja de alguns produtos para alguns clientes realizadas por alguns vendedores.

Modelo conceitual

- O modelo conceitual das entidades envolvidas e seus relacionamentos estão na figura 1:

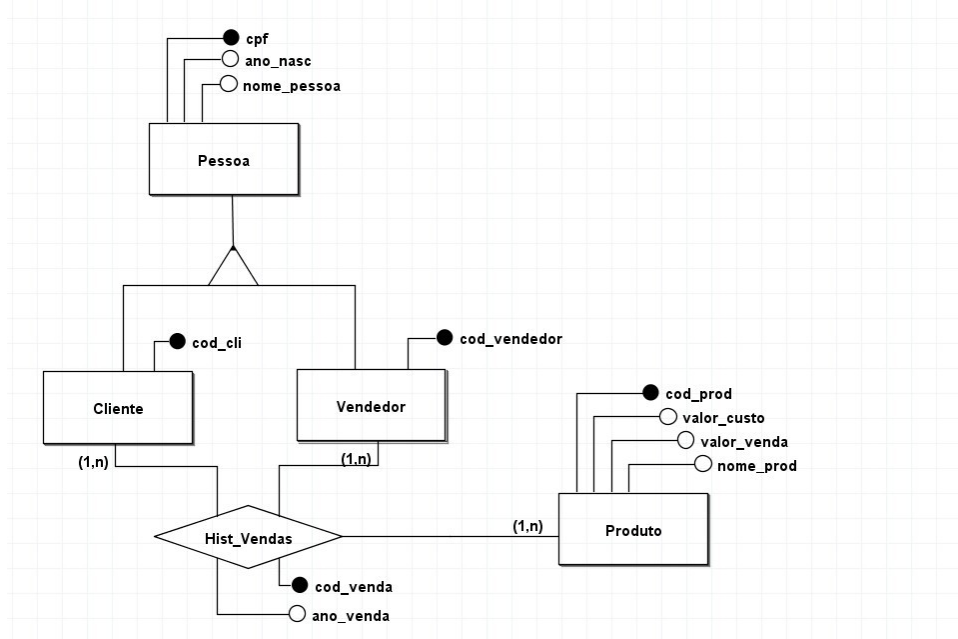


Figura 1: Modelo conceitual usado neste material

Neste diagrama podemos notar que uma pessoa pode ser um cliente ou um vendedor, por isto seus atributos em comum ficam na entidade Pessoa.

Pode-se notar também que muitos vendedores podem vender muitos produtos para muitos clientes, por isto existe um relacionamento ternário com os atributos que são específicos de cada venda realizada, contendo o código e a data da operação.

Modelo lógico

- O modelo lógico destas entidades está representado na figura 2:

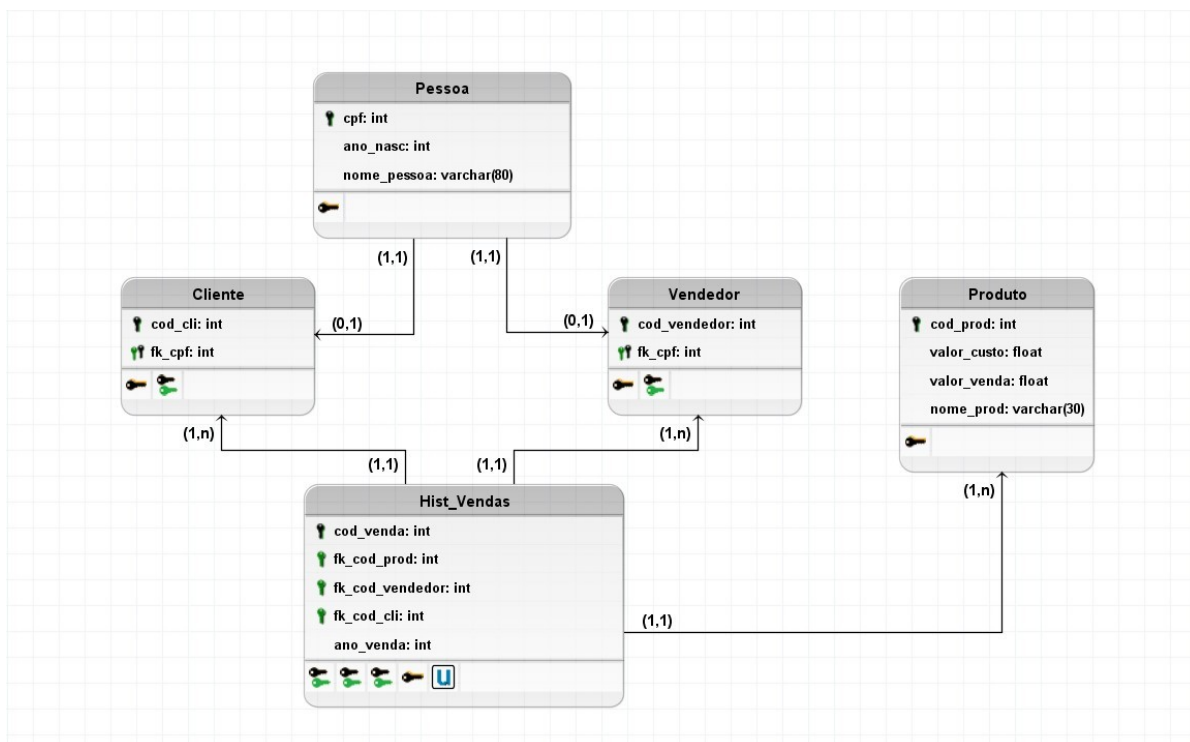


Figura 2: Modelo lógico usado neste material

Um dos propósitos do diagrama lógico é demonstrar as transformações dos relacionamentos em novas tabelas e também explicitar as chaves primárias e estrangeiras

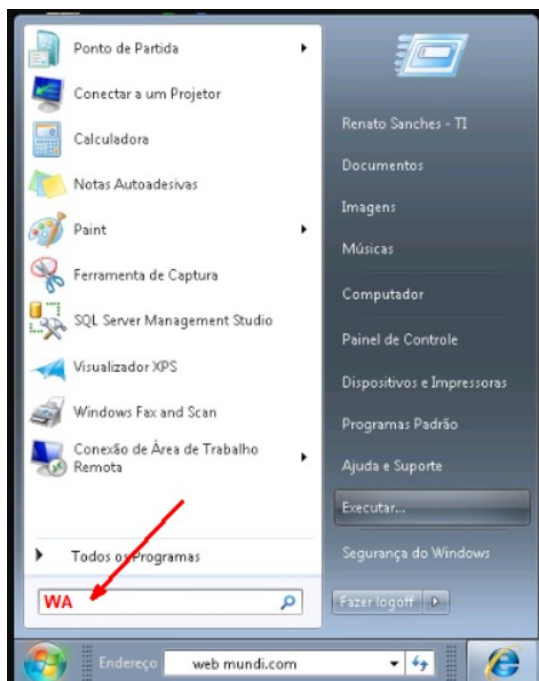
Modelo físico

- O modelo físico, que deve ser usado para a criação das tabelas, chaves e demais elementos está no arquivo [SQL_modelo_fisico.sql](#). Ele foi criado para o Sistema Gerenciador de Bancos de Dados (SGBD) MySQL, na sua versão Maria DB.
- Após a criação das tabelas e demais entidades devem ser inseridos os dados iniciais deste banco de dados. O script com estes dados estão no arquivo [SQL_inserts.sql](#).

Preparação para os exercícios

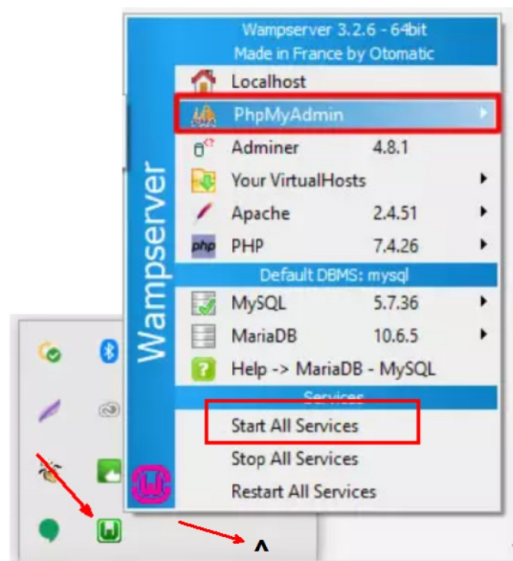
1. O primeiro passo para iniciar este roteiro de exercícios é iniciar o SGBD Maria DB. Para tanto é necessário que ele já esteja devidamente instalado na sua máquina. Em nossos laboratórios é comum encontrar o Maria DB instalado como um componente do WAMP. Então, localize o ícone do SGBD no canto esquerdo do seu computador.

Se ele não estiver visível então pressione ao mesmo tempo as teclas WINDOWS e R e depois digite WA para que apareça o ícone do inicializador.



2. É muito importante certificar que o ícone do SGBD esteja verde. Se ele estiver vermelho ou amarelo ou qualquer outra cor diferente de verde significa que existe algum problema. Localize o ícone do SGBD no canto inferior direito de sua tela. As vezes ele fica visível e em outras vezes ele fica oculto e pode ser exibido clicando no símbolo de circunflexo que fica bem em baixo e à direita da sua tela. Tente clicar em “Start All Services” para contornar o problema. Se desta forma não resolver o problema e o ícone não ficar verde então é possível que seja necessário reinstalar o WAMP todo.

Depois de iniciar o SGBD Maria DB, é necessário disparar um cliente gráfico para ser mais confortável a execução dos comandos. Você pode usar o phpMyAdmin, por exemplo. Este programa também fica no menu que é acionado clicando com o botão direito do mouse no símbolo verde do SGBD.



3. Neste material foi usado o programa HeidiSQL para mostrar os resultados esperados pelas consultas propostas nos exercícios.
4. Se o banco de dados e as tabelas ainda não existem na sua máquina então é necessário executar o script [SQL_modelo_fisico.sql](#) para a criação das estruturas (tabelas, chaves, índices etc.).
5. Depois de criadas as principais estruturas então é necessário inserir os dados básicos para os exercícios. Isto deve ser feito executando o script [SQL_inserts.sql](#) no seu cliente gráfico.

a) Exercícios sobre extrações elementares (SELECT, FROM, WHERE, LIKE):

a.1) Mostre todos os nomes e os valores de todos os produtos cadastrados

Resultado esperado:

produto (5r x 3c)		
nome_prod	valor_custo	valor_venda
prod_1	24	46
prod_2	12	42
prod_3	26	53
prod_4	14	52
prod_5	30	45

Exemplo de resolução:

```
USE aulas;
```

```
SELECT nome_prod, valor_custo, valor_venda
```

```
FROM produto;
```

a.2) Mostre todos os produtos que o valor de custo seja menor que 20,00

Resultado esperado:

produto (2r x 3c)		
nome_prod	valor_custo	valor_venda
prod_2	12	42
prod_4	14	52

a.3) Mostre todos os produtos que a margem (venda - custo) seja maior que 25,00

Resultado esperado:

produto (2r x 4c)			
nome_prod	valor_custo	valor_venda	margem
prod_1	24	46	22
prod_5	30	45	15

1.4) Mostre todos os produtos cujo nome começa com prod.

Resultado esperado:

produto (5r x 3c)		
nome_prod	valor_custo	valor_venda
prod_1	24	46
prod_2	12	42
prod_3	26	53
prod_4	14	52
prod_5	30	45

Exemplo de resolução:

```
USE aulas;  
SELECT nome_prod, valor_custo, valor_venda  
FROM produto  
WHERE nome_prod LIKE 'prod%';
```

b) Extrações usando mais de uma tabela (JOIN)

b.1) Mostre o código, nome e CPF dos clientes

Resultado #1 (5r x 3c)			
cod_cli		nome_cliente	cpf_cliente
1		pessoa_101	101
2		pessoa_102	102
3		pessoa_103	103
4		pessoa_104	104
5		pessoa_105	105

Exemplo de resolução:

```
USE aulas;
```

```
SELECT c.cod_cli, p.nome_pessoa nome_cliente, p.cpf cpf_cliente
```

```
FROM pessoa p, cliente c
```

```
WHERE p.cpf = c.fk_cpf
```

b.2) Mostre todos os clientes que o nome termine com 102

Resultado #1 (1r x 3c)			
cod_cli		nome_cliente	cpf_cliente
2		pessoa_102	102

Exemplo de resposta:

```
USE aulas;
```

```
SELECT c.cod_cli, p.nome_pessoa nome_cliente, p.cpf cpf_cliente
```

```
FROM pessoa p, cliente c
```

```
WHERE p.nome_pessoa LIKE '%102' AND p.cpf = c.fk_cpf
```

b.3) Mostre os vendedores que o nome termine com 2

Resultado #1 (1r x 3c)			
cod_vendedor		nome_vendedor	cpf_vendedor
2		Pessoa_202	202

b.4) Mostre os vendedores que o código seja 3 ou 5

Resultado #1 (2r × 3c)		
cod_vendedor	nome_vendedor	cpf_vendedor
3	Pessoa_203	203
5	Pessoa_205	205

b.5) Mostre nome, ano de nascimento e idade em 2022 de todos os clientes

pessoa (5r × 3c)		
nome_cliente	ano_nasc	idade_em_2022
pessoa_101	2.007	15
pessoa_102	2.002	20
pessoa_103	1.986	36
pessoa_104	1.980	42
pessoa_105	1.969	53

Exemplo de resolução:

```
USE aulas;
```

```
SELECT p.nome_pessoa nome_cliente, p.ano_nasc, (2022 - p.ano_nasc) idade_em_2022
FROM pessoa p, cliente c
WHERE p.cpf = c.fk_cpf
```

b.6) Mostre todos os vendedores com idade maior que 40 anos em 2022

pessoa (2r × 3c)		
nome_vendedor	ano_nasc	idade_em_2022
Pessoa_201	1.973	49
Pessoa_205	1.981	41

c) Extrações com contagem e ordenação (COUNT e ORDER BY)

c.1) Mostre o código, nome e CPF dos clientes

pessoa (5r × 3c)		
nome_cliente	ano_nasc	idade_em_2022
pessoa_101	2.007	15
pessoa_102	2.002	20
pessoa_103	1.986	36
pessoa_104	1.980	42
pessoa_105	1.969	53

Exemplo de resolução:

```
USE aulas;
```

```
SELECT p.nome_pessoa nome_cliente, p.ano_nasc, (2022 - p.ano_nasc) idade_em_2022
```

```
FROM pessoa p, cliente c
```

```
WHERE p.cpf = c.fk_cpf
```

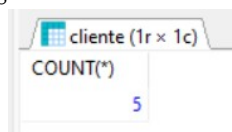
```
ORDER BY idade_em_2022
```

c.2) Mostre os vendedores ordenados pela idade



nome_vendedor	ano_nasc	idade_em_2022
Pessoa_202	1.994	28
Pessoa_203	1.991	31
Pessoa_204	1.987	35
Pessoa_205	1.981	41
Pessoa_201	1.973	49

c.3) Mostre a quantidade de clientes



cliente (1r x 1c)
COUNT(*)
5

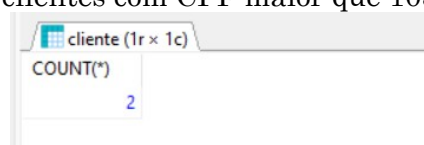
Exemplo de resolução:

```
USE aulas;
```

```
SELECT COUNT(*)
```

```
FROM cliente c
```

c.4) Mostre a quantidade de clientes com CPF maior que 103



cliente (1r x 1c)
COUNT(*)
2

c.5) Mostre todos os produtos em ordem decrescente de valor_custo

produto (5r x 4c)				
cod_prod	nome_prod	valor_custo	valor_venda	
5	prod_5	30	45	
3	prod_3	26	53	
1	prod_1	24	46	
4	prod_4	14	52	
2	prod_2	12	42	

Exemplo de resolução:

```
USE aulas;
SELECT cod_prod, nome_prod, valor_custo, valor_venda
FROM produto pr
ORDER BY valor_custo DESC
```

c.6) Mostre todos os produtos em ordem decrescente de margem (valor_venda - valor_custo)

produto (5r x 5c)					
cod_prod	nome_prod	valor_custo	valor_venda	margem	
4	prod_4	14	52	38	
2	prod_2	12	42	30	
3	prod_3	26	53	27	
1	prod_1	24	46	22	
5	prod_5	30	45	15	

d) Extrações com 3 ou mais tabelas

d.1) Mostre o código da venda, nome do produto, nome do cliente e nome do vendedor dos produtos vendidos em 2022

Resultado #1 (5r x 5c)				
cod_venda	ano_venda	nome_prod	cliente	vendedor
6	2.022	prod_1	pessoa_104	Pessoa_204
9	2.022	prod_4	pessoa_103	Pessoa_201
11	2.022	prod_3	pessoa_105	Pessoa_204
17	2.022	prod_4	pessoa_105	Pessoa_201
18	2.022	prod_3	pessoa_101	Pessoa_201

Exemplo de resolução:

```
USE aulas;
SELECT h.cod_venda, h.ano_venda, pr.nome_prod, pc.nome_pessoa cliente, pv.nome_pessoa vendedor
FROM pessoa pc, cliente c, pessoa pv, vendedor v, produto pr, hist_vendas h
USE aulas;
```

```
WHERE h.ano_venda = 2022
      AND h.fk_cod_prod = pr.cod_prod
      AND h.fk_cod_vendedor = v.cod_vendedor AND v.fk_cpf = pv.cpf
      AND h.fk_cod_cli = c.cod_cli AND c.fk_cpf = pc.cpf
ORDER BY h.cod_venda
```

d.2) Mostre o código da venda, nome do produto, nome do cliente, nome do vendedor e margem dos produtos vendidos em 2022 com margem superior a 30,00

Resultado #1 (2r x 6c)					
cod_venda	ano_venda	nome_prod	cliente	vendedor	margem
9	2.022	prod_4	pessoa_103	Pessoa_201	38
17	2.022	prod_4	pessoa_105	Pessoa_201	38

d.3) Mostre o código da venda, nome do produto, nome do cliente, nome do vendedor e margem dos produtos vendidos em 2022 pelo vendedor Pessoa_204

Resultado #1 (2r x 6c)					
cod_venda	ano_venda	nome_prod	cliente	vendedor	margem
6	2.022	prod_1	pessoa_104	Pessoa_204	22
11	2.022	prod_3	pessoa_105	Pessoa_204	27

Agradecimentos:

Agradecemos a colaboração e atenção dos amigos que contribuíram para a elaboração deste material de apoio.

Toda e qualquer contribuição será sempre muito bem-vinda.