

# Histórico de Vendas

## Exibições (VIEWS)

Nome: Matheus Henrique Elias Rosa

RA: 319141144

### Código Limpo

Seu código deverá estar muito bem formatado, indentado e estruturado.

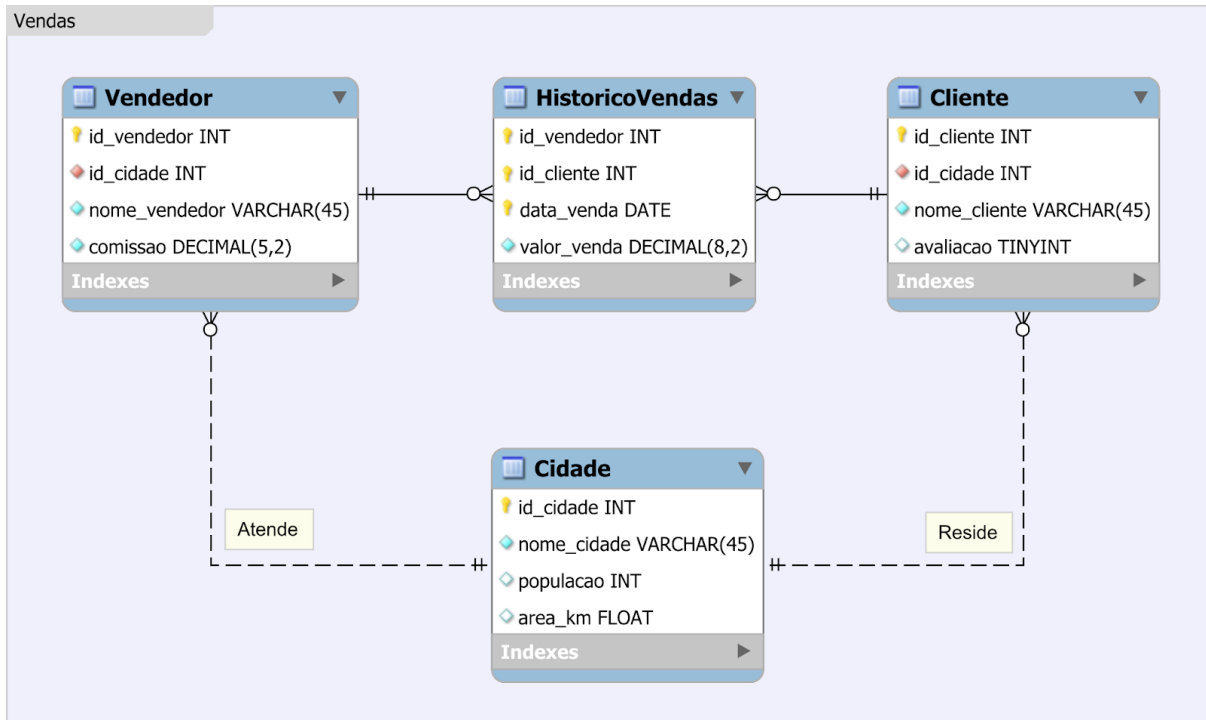
### Importante • Não Serão Aceitas Cópias!

Em hipótese alguma serão aceitas cópias de exercícios ou parte deles, caso seja identificada, será feita a correção de apenas uma das cópias e dividida a nota pela quantidade de cópias identificadas.

terça-feira, 19 de maio de 2020



## 1. Banco de Dados de Histórico de Vendas



### Script de Criação

```

CREATE DATABASE vendas;
USE vendas;

CREATE TABLE Cidade(
    id_cidade INT NOT NULL,
    nome_cidade VARCHAR(45) NOT NULL,
    populacao INT DEFAULT 0,
    area_km FLOAT DEFAULT 0,
    PRIMARY KEY (id_cidade)
);

CREATE TABLE Vendedor(
    id_vendedor INT NOT NULL,
    id_cidade INT NOT NULL,

```

```

    nome_vendedor VARCHAR(45) NOT NULL,
    comissao DECIMAL(5, 2) NOT NULL DEFAULT 0,
    PRIMARY KEY (id_vendedor),
    CONSTRAINT fk_vendedor_cidade
    FOREIGN KEY (id_cidade) REFERENCES Cidade(id_cidade)
);

CREATE TABLE Cliente(
    id_cliente INT NOT NULL,
    id_cidade INT NOT NULL,
    nome_cliente VARCHAR(45) NOT NULL,
    avaliacao TINYINT DEFAULT 0,
    PRIMARY KEY (id_cliente),
    CONSTRAINT fk_cliente_cidade
    FOREIGN KEY (id_cidade) REFERENCES Cidade(id_cidade)
);

CREATE TABLE HistoricoVendas(
    id_vendedor INT NOT NULL,
    id_cliente INT NOT NULL,
    data_venda DATE NOT NULL,
    valor_venda DECIMAL(8, 2) DEFAULT 0,
    PRIMARY KEY (id_vendedor, id_cliente, data_venda),
    CONSTRAINT fk_historico_vendas_vendedor
    FOREIGN KEY (id_vendedor) REFERENCES Vendedor(id_vendedor),
    CONSTRAINT fk_historico_vendas_cliente
    FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente)
);

```

## Dados

```

-- -----
-- Dados da tabela Cidade --
-- -----

INSERT INTO Cidade VALUES (1, 'Belo Horizonte', 2512070, 331.4);
INSERT INTO Cidade VALUES (2, 'Contagem', 663855, 195.3);
INSERT INTO Cidade VALUES (3, 'Betim', 439340, 343.7);

```

```
INSERT INTO Cidade VALUES (4, 'Ribeirão das Neves', 334858, 155.5);
INSERT INTO Cidade VALUES (5, 'Sete Lagoas', 239639, 537.6);
INSERT INTO Cidade VALUES (6, 'Santa Luzia', 219134, 235.3);
INSERT INTO Cidade VALUES (7, 'Ibirité', 180204, 72.6);
INSERT INTO Cidade VALUES (8, 'Sabará', 136344, 302.2);
INSERT INTO Cidade VALUES (9, 'Vespasiano', 127601, 71.2);
INSERT INTO Cidade VALUES (10, 'Nova Lima', 94889, 429.0);
```

```
-- -----
-- Dados da tabela Vendedor --
-- -----
```

```
INSERT INTO Vendedor VALUES (1, 6, 'Bernardo', 0.10);
INSERT INTO Vendedor VALUES (2, 1, 'Giovana', 0.25);
INSERT INTO Vendedor VALUES (3, 8, 'Nilton', 0.15);
INSERT INTO Vendedor VALUES (4, 1, 'Paulo', 0.25);
INSERT INTO Vendedor VALUES (5, 2, 'Vanessa', 0.20);
```

```
-- -----
-- Dados da tabela Cliente --
-- -----
```

```
INSERT INTO Cliente VALUES (1, 1, 'Danilo', 3);
INSERT INTO Cliente VALUES (2, 4, 'Elen', 2);
INSERT INTO Cliente VALUES (3, 2, 'Caetano', 4);
INSERT INTO Cliente VALUES (4, 1, 'Gerson', 5);
INSERT INTO Cliente VALUES (5, 2, 'Vinícius', 4);
INSERT INTO Cliente VALUES (6, 9, 'Lucas', 1);
INSERT INTO Cliente VALUES (7, 7, 'Clarisse', 3);
INSERT INTO Cliente VALUES (8, 1, 'Davi', 5);
INSERT INTO Cliente VALUES (9, 2, 'Rafaela', 4);
INSERT INTO Cliente VALUES (10, 3, 'José', 3);
```

```
-- -----
-- Dados da tabela Histórico Vendas --
-- -----
```

```
INSERT INTO HistoricoVendas VALUES (2, 1, '2019-08-15', 501.10);
INSERT INTO HistoricoVendas VALUES (2, 4, '2019-08-16', 645.30);
INSERT INTO HistoricoVendas VALUES (2, 1, '2019-09-02', 580.00);
INSERT INTO HistoricoVendas VALUES (2, 4, '2019-09-06', 600.25);
INSERT INTO HistoricoVendas VALUES (2, 1, '2019-09-15', 406.24);
```

```
INSERT INTO HistoricoVendas VALUES (2, 1, '2019-09-17', 560.00);
INSERT INTO HistoricoVendas VALUES (3, 7, '2019-10-09', 560.30);
INSERT INTO HistoricoVendas VALUES (3, 7, '2019-10-28', 250.45);
INSERT INTO HistoricoVendas VALUES (4, 1, '2019-10-16', 576.00);
INSERT INTO HistoricoVendas VALUES (4, 1, '2019-10-18', 343.86);
INSERT INTO HistoricoVendas VALUES (4, 8, '2019-10-27', 675.00);
INSERT INTO HistoricoVendas VALUES (5, 9, '2019-08-15', 600.00);
INSERT INTO HistoricoVendas VALUES (5, 9, '2019-08-22', 520.25);
INSERT INTO HistoricoVendas VALUES (5, 5, '2019-10-14', 505.10);
INSERT INTO HistoricoVendas VALUES (5, 5, '2019-10-19', 480.25);
```

### Dados da Tabela Cidade

id_cidade	nome_cidade	populacao	area_km
1	Belo Horizonte	2512070	331.4
2	Contagem	663855	195.3
3	Betim	439340	343.7
4	Ribeirão das Neves	334858	155.5
5	Sete Lagoas	239639	537.6
6	Santa Luzia	219134	235.3
7	Ibirité	180204	72.6
8	Sabará	136344	302.2
9	Vespasiano	127601	71.2
10	Nova Lima	94889	429

### Dados da Tabela Vendedor

id_vendedor	id_cidade	nome_vendedor	comissao
1	6	Bernardo	0.10

2	1	Giovana	0.25
3	7	Nilton	0.15
4	1	Paulo	0.25
5	2	Vanessa	0.20

### Dados da Tabela Cliente

id_cliente	id_cidade	nome_cliente	avaliacao
1	1	Danilo	3
2	4	Elen	2
3	2	Caetano	4
4	1	Gerson	5
5	2	Vinícius	4
6	9	Lucas	1
7	7	Clarisse	3
8	1	Davi	5
9	2	Rafaela	4
10	3	José	3

### Dados da Tabela Histórico de Vendas

id_vendedor	id_cliente	data_venda	valor_venda
2	1	2019-08-15	501.10
2	4	2019-08-16	645.30
2	1	2019-09-02	580.00

---

2	4	2019-09-06	600.25
2	1	2019-09-15	406.24
2	1	2019-09-17	560.00
3	7	2019-10-09	560.30
3	7	2019-10-28	250.45
4	1	2019-10-16	576.00
4	1	2019-10-18	343.86
4	8	2019-10-27	675.00
5	9	2019-08-15	600.00
5	9	2019-08-22	520.25
5	5	2019-10-14	505.10

## 2. Exercícios Sobre Exibições (**VIEWS**)

1. Crie uma exibição (**VIEW**) chamada **avaliacao\_clientes** que mostre todos os dados de todos os clientes em ordem decrescente por avaliação.

```
CREATE VIEW avaliacao_clientes AS
SELECT * FROM cliente
ORDER BY avaliacao DESC;
```

2. Crie uma exibição (**VIEW**) chamada **vendedores\_bh**, que mostre o id e nome dos vendedores de Belo Horizonte em ordem alfabética.

```
CREATE VIEW vendedores_bh AS
SELECT id_vendedor, nome_vendedor
FROM vendedor
JOIN cidade ON cidade.id_cidade = vendedor.id_cidade
WHERE cidade.nome_cidade = 'Belo Horizonte'
ORDER BY nome_vendedor ASC;
```

3. Crie uma exibição (**VIEW**) chamada **sumario\_vendas**, que mostre em ordem decrescente a quantidade de vendas feita por cada vendedor.

```
CREATE VIEW sumario_vendas AS
SELECT DISTINCT vendedor.id_vendedor, nome_vendedor, count(*)
FROM vendedor
INNER JOIN HistoricoVendas ON vendedor.id_vendedor =
HistoricoVendas.id_vendedor
GROUP BY nome_vendedor
ORDER BY count(*) DESC;
```



4. Crie uma exibição (**VIEW**) chamada **lucro\_vendas**, que retorne em ordem decrescente o nome do vendedor e o seu lucro, com base na comissão e valor da venda.

```
CREATE VIEW lucro_vendas AS
SELECT DISTINCT nome_vendedor, SUM(valor_venda * comissao)
FROM vendedor
INNER JOIN HistoricoVendas ON vendedor.id_vendedor =
HistoricoVendas.id_vendedor
group by nome_vendedor
ORDER BY nome_vendedor and sum(valor_venda * comissao) DESC;
```

5. Crie a view **qtde\_vendas** que retorne a quantidade de vendas no mês de outubro?

```
CREATE VIEW qtde_vendas AS
SELECT COUNT(*)
FROM historico vendas
WHERE data_venda >= '2019-10-01' AND data_venda <= '2019-10-31'
ORDER BY data_venda asc;
```

6. Atualize a exibição exibição (**VIEW**) **avaliacao\_cliente** criada na questão 1, para que ela mostre apenas o nome e a avaliação do cliente em ordem decrescente.

```
CREATE OR REPLACE VIEW avaliacao_clientes AS
SELECT nome_cliente, avaliacao
FROM cliente
ORDER BY avaliacao DESC;
```

7. Crie a exibição **planejamento\_vendas\_cidade** que mostre para cada um dos vendedores cadastrados no banco de dados as possíveis cidades que ele pode atuar.

```
CREATE VIEW planejamento_vendas_cidade AS
SELECT nome_vendedor, nome_cidade
FROM vendedor
JOIN cidade ON cidade.id_cidade = vendedor.id_cidade
ORDER BY nome_vendedor asc;
```

8. Crie a **VIEW vendedores\_sem\_clientes** que mostre o nome dos vendedores que ainda não atenderam nenhum cliente em ordem alfabética.

```
CREATE VIEW vendedores_sem_clientes AS
SELECT nome_vendedor
FROM vendedor
WHERE NOT EXISTS(SELECT * FROM historicovendas WHERE
historicovendas.id_vendedor = vendedor.id_vendedor)
ORDER BY nome_vendedor ASC;
```

9. Escreva um comando SQL para apagar as seguintes **VIEWS**: **qtde\_vendas** (Questão 5) e **vendedores\_bh** (Questão 2).

```
DROP VIEW qtde_vendas, vendedores_bh;
```

10. Crie uma exibição que liste os nomes dos clientes que estão abaixo da média de avaliação, em ordem decrescente por avaliação e alfabética por nome.

```
CREATE VIEW clientes_abaixo_da_media_da_avaliacao AS  
SELECT nome_cliente, avaliacao  
FROM cliente  
WHERE avaliacao <= ANY (SELECT AVG(avaliacao) FROM cliente)  
ORDER BY avaliacao AND nome_cliente DESC;
```