

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
4º SEMESTRE
JOÃO VICTOR AMORIM

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
Programação e Desenvolvimento de Banco de Dados

São José dos Campos, SP
10/08/2024

JOÃO VICTOR AMORIM

ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
Programação e Desenvolvimento de Banco de Dados

Trabalho apresentado com imagens e comandos SQL.
Orientador Prof. Anderson Emidio de Macedo Goncalves

SUMÁRIO

1. INTRODUÇÃO
2. DESENVOLVIMENTO
3. CONCLUSÃO

1 INTRODUÇÃO

Neste portfolio criamos um Banco de Dados MySQL hospedado no servidor local XAMPP, usamos o SGBD MySQL Workbench para executar as query SQL.

A Modelagem do Banco de Dados foi seguida conforme a proposta feita pelo Diagrama E.R. que estava no Roteiro da aula pratica.

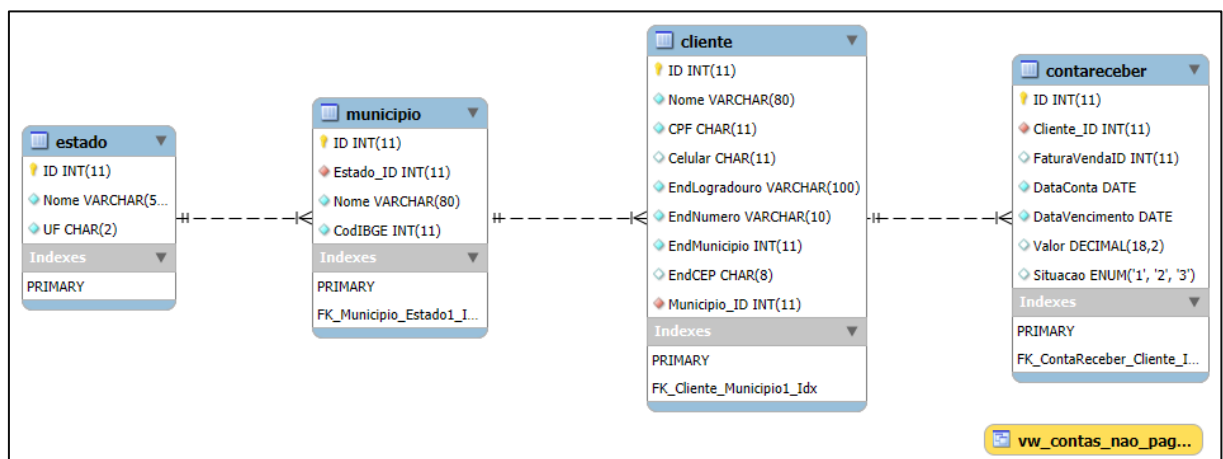
2 DESENVOLVIMENTO

Criei as tabelas na seguinte ordem para ligar as FK Chave Estrangeira:

1. Tabela Estado.
2. Tabela Municipio
3. Tabela Cliente
4. Tabela ContaReceber.

Foi criado cada tabela com sua PK e FK com cada FK nomeada para uma fácil interpretação na manutenção do Database também foi criado os tipos de dados conforme proposto no roteiro. Foi criado uma View para visualizar as contas não pagas com situação 1 = Conta Registrada.

Abaixo está uma foto do D.E.R.



Código da criação de cada tabela:

1. Tabela Estado:

```
CREATE TABLE Estado
(
    ID INT PRIMARY KEY AUTO_INCREMENT,
    Nome VARCHAR(50) NOT NULL,
    UF CHAR(2) NOT NULL
);
```

2. Tabela Municipio:

```

CREATE TABLE Municipio
(
    ID INT PRIMARY KEY AUTO_INCREMENT,
    Estado_ID INT NOT NULL,
    Nome VARCHAR(80) NOT NULL,
    CodIBGE INT NOT NULL,
    CONSTRAINT FK_Municipio_Estado1_Idx FOREIGN KEY
    (Estado_ID)
    REFERENCES Estado (ID)
);

```

3. Tabela Cliente:

```

CREATE TABLE Cliente
(
    ID INT PRIMARY KEY AUTO_INCREMENT,
    Nome VARCHAR(80) NOT NULL,
    CPF CHAR(11) NOT NULL,
    Celular CHAR(11),
    EndLogradouro VARCHAR(100) NOT NULL,
    EndNumero VARCHAR(10) NOT NULL,
    EndMunicipio INT NOT NULL,
    EndCEP CHAR(8),
    Municipio_ID INT NOT NULL,
    CONSTRAINT FK_Cliente_Municipio1_Idx FOREIGN KEY
    (Municipio_ID)
    REFERENCES Municipio (ID)
);

```

4. Tabela ContaReceber:

```

CREATE TABLE ContaReceber
(
  ID INT PRIMARY KEY AUTO_INCREMENT,
  Cliente_ID INT NOT NULL,
  CONSTRAINT FK_ContaReceber_Cliente_Idx FOREIGN KEY
  (Cliente_ID)
  REFERENCES Cliente (ID),
  FaturaVendaID INT,
  DataConta DATE NOT NULL,
  DataVencimento DATE NOT NULL,
  Valor DECIMAL(18,2),
  Situacao ENUM('1', '2', '3')
);

```

Abaixo o código do script inserir.sql:

-- Inserindo dados na tabela Estado

```

INSERT INTO Estado
(Nome, UF)
VALUES
('São Paulo', 'SP');

```

```

INSERT INTO Estado
(Nome, UF)
VALUES
('Rio de Janeiro', 'RJ');

```

```

INSERT INTO Estado
(Nome, UF)
VALUES
('Rio Grande do Sul', 'RS');

```

-- Inserindo dados na Tabela Municipio

```
INSERT INTO Municipio
(Estado_ID, Nome, CodIBGE)
VALUES
(1, 'São José dos Campos', 3549904);
```

```
INSERT INTO Municipio
(Estado_ID, Nome, CodIBGE)
VALUES
(3, 'Gramado', 4309100);
```

```
INSERT INTO Municipio
(Estado_ID, Nome, CodIBGE)
VALUES
(2, 'Resende', 3304201);
```

-- Inserindo dados na Tabela Cliente

```
INSERT INTO Cliente
(Nome,
CPF,
Celular,
EndLogradouro,
EndNumero,
EndMunicipio,
EndCEP,
Municipio_ID)
VALUES
('Luiz Gomes de Souza',
'24367876534',
'12998547865',
```



```
        'R. Santo Expedito',  
        '735',  
3549904,  
        2260040,  
        2);
```

```
INSERT INTO Cliente  
(Nome,  
CPF,  
        Celular,  
        EndLogradouro,  
EndNumero,  
        EndMunicipio,  
        EndCEP,  
        Municipio_ID)
```

```
VALUES  
(('Paulo Roberto Oliveira',  
  '86734578698',  
  '54997864534',  
    'R Vinte e Dois',  
    '674',  
4309100,  
    13356765,  
    3);
```

```
INSERT INTO Cliente  
(Nome,  
CPF,  
        Celular,  
        EndLogradouro,  
EndNumero,  
        EndMunicipio,  
        EndCEP,  
        Municipio_ID)
```

VALUES

```
('João Pedro da Silva',  
'78945378623',  
'24998346789',  
    'R. das Palmeiras',  
    '845',  
3304201,  
    12004567,  
    4);
```

-- Inserindo dados na Tabela ContaReceber

INSERT INTO ContaReceber

```
(Cliente_ID,  
    DataConta,  
    DataVencimento,  
    Valor,  
    Situacao)
```

VALUES

```
(2,  
    '2023-05-24',  
    '2023-06-05',  
159.99,  
    3);
```

INSERT INTO ContaReceber

```
(Cliente_ID,  
    DataConta,  
    DataVencimento,  
    Valor,  
    Situacao)
```

VALUES

```
(3,
```

```
        '2023-12-22',  
        '2024-01-15',  
672.99,  
1);
```

```
INSERT INTO ContaReceber  
(Cliente_ID,  
    DataConta,  
    DataVencimento,  
    Valor,  
    Situacao)  
VALUES  
(1,  
    '2024-02-08',  
    '2024-03-11',  
3799.99,  
1);
```

Foram inseridos três registro por cada tabela, dois desses registros estavam com compras não pagas, situação 1.

Abaixo script de consulta.sql onde contem a criação da VIEW:

-- Criando a VIEW

```
CREATE OR REPLACE VIEW VW_Contas_Nao_Pagas
AS
SELECT      ct.ID AS ID_Conta,
cl.Nome AS Nome_Cliente,
            ct.CPF AS CPF_Cliente,
            ct.DataVencimento AS DataVencimento_Conta,
            ct.Valor AS Valor_Conta
FROM        ContaReceber ct
JOIN        Cliente cl ON ct.Cliente_ID = cl.ID
WHERE       Situacao = 1;
```

-- Consultando a VIEW

```
SELECT      *
FROM        VW_Contas_Nao_Pagas;
```

Abaixo print da consulta da VIEW:

Result Grid					
		Filter Rows:		Export:	Wrap Cell Content:
	ID_Conta	Nome_Cliente	CPF_Cliente	DataVencimento_Conta	Valor_Conta
▶	3	Luiz Gomes de Souza	24367876534	2024-03-11	3799.99
	2	João Pedro da Silva	78945378623	2024-01-15	672.99

Abaixo print da consulta da Tabela Estado:

Result Grid			
		Filter Rows:	
	ID	Nome	UF
▶	1	São Paulo	SP
	2	Rio de Janeiro	RJ
	3	Rio Grande do Sul	RS
⬇	NULL	NULL	NULL

Abaixo print da consulta da Tabela Municipio:

Result Grid				
		Filter Rows:		Edit:
	ID	Estado_ID	Nome	CodIBGE
▶	2	1	São José dos Campos	3549904
	3	3	Gramado	4309100
	4	2	Resende	3304201
⬇	NULL	NULL	NULL	NULL

Abaixo print da consulta da Tabela Cliente:

3 CONCLUSÃO

Modelagem do Banco de Dados criado conforme o D.E.R. proposto no roteiro da aula pratica, tabelas criadas com sucesso e feito os testes.

REFERÊNCIAS

Curso SQL: [Banco de Dados Oracle Completo: SQL+PLSQL+Modelagem | Udemey](#)