

Avaliação 06

Nome: Júlia Maria Freitas Bezerra

Turma: P4- Informática

Data: 04/10/2023

A partir do Banco de Dados definido no Modelo Entidade Relacionamento anexo fazer as seguintes atividades:

1) Inserir na tabela TB_CLIENTES os seguintes dados:

NOME_CLI	ENDERECO
José Maria Alves	Av João Pessoa 2081
Maria Conceição Tavares	Rua Waldery Uchoa 4
João Cosme Fonseca	Rua Padre Franscisco Pinto 790

Observação: A chave primaria é ID com auto-incremento

```
sqlite> create table TB_CLIENTES (id_cliente integer primary key autoincrement, nome_cli text not null, endereco text not null);
sqlite> insert into TB_CLIENTES (nome_cli, endereco) values ("José Maria Alves", " Av João Pessoa 2081"), ("Maria Conceição Tavares", "Rua Waldery Uchoa 4"), ("João Cosme Fonseca", "Rua Padre Franscisco Pinto 790");
sqlite> select * from TB_CLIENTES;
1|José Maria Alves| Av João Pessoa 2081
2|Maria Conceição Tavares|Rua Waldery Uchoa 4
3|João Cosme Fonseca|Rua Padre Franscisco Pinto 790
sqlite>
```

2) Inserir na Tabela TB_VENDEDORES os seguintes dados:

NOME

Luciano Arruda Cavalcante

Joana Alves Pessoa

Mercia Bessa Santos

Antonio de Padua Lopes

Observação: A chave primaria é ID com auto-incremento

```
sqlite> create table TB_VENDEDORES (id_vendedor integer primary key autoincrement, nome_vendedor text not null);
sqlite> insert into TB_Vendedores (nome_vendedor) values ("Luciano Arruda Cavalcante"), ("Joana Alves Pessoa"), ("Mercia Bessa Santos"), ("Antonio de Padua Lopes");
sqlite> select * from TB_VENDEDORES;
1|Luciano Arruda Cavalcante
2|Joana Alves Pessoa
3|Mercia Bessa Santos
4|Antonio de Padua Lopes
```

3) Inserir na Tabela TB_PRODUTOS os seguintes dados:

CODIGO	NOME	PRECO_UNITARIO
100	Arroz Tio João	6.00
150	Feijão Cariquinha	5.50
200	Macarrão Fortaleza	3.50
250	Oleo de Soja	4.00
300	Manteiga Betania 500g	8.00
350	Queijo Ricota Betania	7.00

Observação: A chave primaria é ID com auto-incremento

```
sqlite> create table TB_PRODUTOS (id_prod integer primary key autoincrement, codigo integer, nome_prod text not null, preco_unitario real);
```

```
sqlite> insert into TB_PRODUTOS (codigo, nome_prod, preco_unitario) VALUES (100, "Arroz Tio João", 6.0), (150, "Feijão Cariquinha", 5.5), (200, "Macarrão Fortaleza", 3.5), (250, "Óleo de Soja", 4.0), (300, "Manteiga Betânia 500g", 8.0), (350, "Queijo Ricota Betânia", 7.0);
sqlite> select * from TB_PRODUTOS;
1|100|Arroz Tio João|6.0
2|150|Feijão Cariquinha|5.5
3|200|Macarrão Fortaleza|3.5
4|250|Óleo de Soja|4.0
5|300|Manteiga Betânia 500g|8.0
6|350|Queijo Ricota Betânia|7.0
sqlite>
```

4) Inserir na Tabela TB_NOTAS_FISCAIS os seguintes dados:

COD_CLI	COD_VEND	NUM_NF	SERIE_NF
1	1	100	A
3	2	101	A
2	3	102	A
4	4	103	A
2	1	104	A
1	3	105	A
3	2	106	A
4	4	107	A

Observação: A chave primaria é NUM_NF

```
sqlite> create table TB_NOTAS_FISCAIS (cod_cliente integer, cod_venda integer, num_nf integer primary key, serie_nf text);
sqlite> insert into TB_NOTAS_FISCAIS (cod_cliente, cod_venda, num_nf, serie_nf) values (1,1,100, "A"), (3,2,101,"A"), (2,3,102,"A"),(4,4,103,"A"), (2,1,104,"A"), (1,3,105,"A"), (3,2,106,"A"), (4,4,107,"A");
sqlite> select * from TB_NOTAS_FISCAIS;
1|1|100|A
3|2|101|A
2|3|102|A
4|4|103|A
2|1|104|A
1|3|105|A
3|2|106|A
4|4|107|A
```

5) Inserir na Tabela TB_ITENS_NOTAS_FISCAIS os seguintes dados:

NUM_NF	COD_PRO	QTD
100	100	5
100	150	4
100	200	4
101	250	8
101	300	4
102	100	6
102	250	8
103	300	4
103	350	4
104	150	10
104	100	12
106	150	10
106	200	10
107	100	10

107	150	10
107	200	10

Observação: A chave primaria é (NUM_NF, COD_PROD)

COD_PROD é chave estrangeira em TB_ITENS_NOTAS_FISCAIS e
chave primária em TB_PRODUTOS.

```
sqlite> create table TB_ITENS_NOTAS_FISCAIS (num_nf integer, COD_PROD integer, qtd integer, primary key (num_nf, COD_PROD), foreign key (COD_PROD) REFERENCES TB_PRODUTOS (id_prod));;
sqlite> insert into TB_ITENS_NOTAS_FISCAIS (num_nf, COD_PROD, qtd) values (
100, 100, 5), (100,150,4), (100, 200,4), (101,250,8), (101, 300, 4), (102,
100, 6), (102, 250, 8), (103, 300, 4), (103, 350, 4), (104, 150, 10), (104,
100, 12), (106, 150, 10), (106, 200, 10), (107, 100, 10), (107, 150, 10),
(107,200,10);
sqlite> select * from TB_ITENS_NOTAS_FISCAIS;
100|100|5
100|150|4
100|200|4
101|250|8
101|300|4
102|100|6
102|250|8
103|300|4
103|350|4
104|150|10
104|100|12
106|150|10
106|200|10
107|100|10
107|150|10
107|200|10
```

→ Schema

```
sqlite> .schema
CREATE TABLE TB_CLIENTES (id_cliente integer primary key autoincrement, nome_cli text not null, endereco text not null);
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE TB_VENDEDORES (id_vendedor integer primary key autoincrement, nome_vendedor text not null);
CREATE TABLE TB_PRODUTOS (id_prod integer primary key autoincrement, codigo integer, nome_prod text not null, preco_unitario real);
CREATE TABLE TB_NOTAS_FISCAIS (cod_cliente integer, cod_venda integer, num_nf integer primary key, serie_nf text);
CREATE TABLE TB_ITENS_NOTAS_FISCAIS (num_nf integer, COD_PROD integer, qtd integer, primary key (num_nf, COD_PROD), foreign key (COD_PROD) REFERENCES TB_PRODUTOS (id_prod));
```

DB BROWSER A SEGUIR

EVIDENCIAS:

- 1) Print de execução dos comandos com posterior `Select *` de cada tabela; **(JÁ COLOCADO)**
- 2) Print do DBBROWSER ou Replit da Estrutura das Tabelas após o create.

→ DB Browser

