



Nível 2: Vamos Manter as Informações?

Rodrigo de O. Alarcon - 202204482321

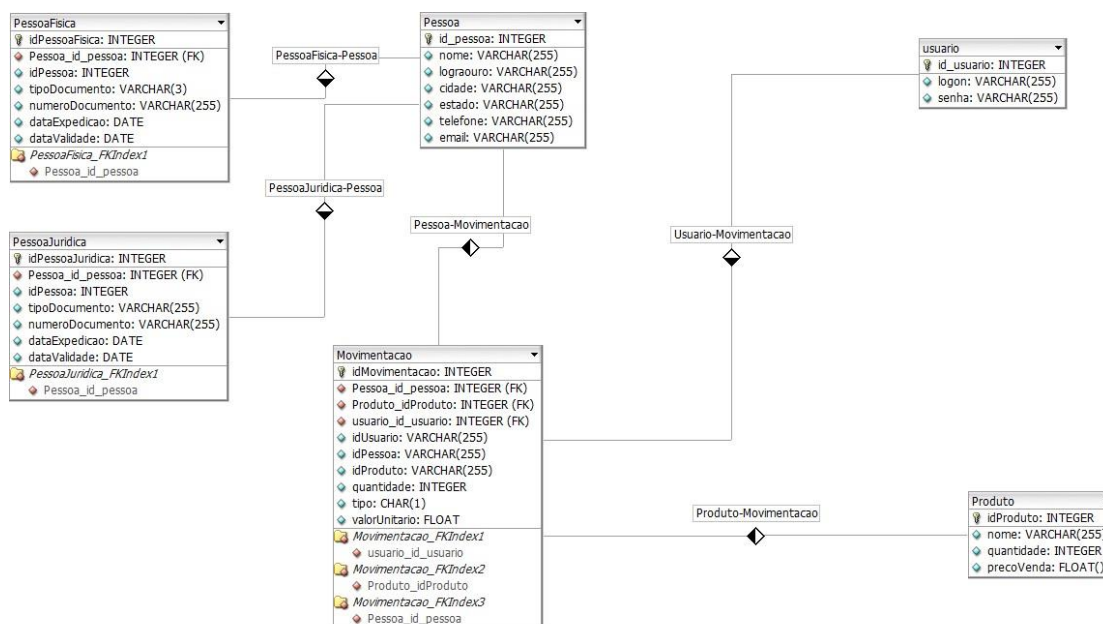
Vila Mariana

Vamos Manter as Informações? – 2022.2 – 3º Semestre

Objetivo da Prática

1º Procedimento – Criando o Banco de Dados

Inserir neste campo, **de forma organizada**, todos os códigos do roteiro do 1º Procedimento da Atividade Prática, os resultados da execução do código e a Análise e Conclusão:



```
use Loja;
go

-- Cria a tabela Usuario no banco Loja
CREATE TABLE Usuario (id_user int IDENTITY(1,1) PRIMARY KEY,
logon varchar(3) not null,
senha varchar(3) not null,
);

-- Inserindo dados na tabela de Usuário
INSERT INTO Usuario(logon, senha)
VALUES ('op2', 'op2')

select * from Usuario;
```

Resultados		Mensagens	
	id_user	logon	senha
1	1	op1	op1
2	2	op2	op2

```

-- Cria a tabela Produto no banco Loja
CREATE TABLE Produto (id_produto int IDENTITY(1,1) PRIMARY KEY,
nome varchar(255) not null,
quantidade integer not null,
preco_venda float
);

-- Inserindo dados na tabela de Produto
INSERT INTO Produto(nome, quantidade, preco_venda)
VALUES ('Limão', 152, 3.88)

DELETE FROM Produto WHERE nome='Maçã';

ALTER TABLE Produto
DROP COLUMN preco_venda;

ALTER TABLE Produto
ADD preco_venda float;

UPDATE Produto
SET preco_venda = 5.25
WHERE nome = 'Banana';

select * from Produto;

```

Resultados		Mensagens		
	id_produto	nome	quantidade	preco_venda
1	4	Banana	100	5,75
2	5	Laranja	500	2,25
3	6	Manga	800	4,1
4	7	Mexirica	150	4,88
5	8	Uva	250	9,25
6	9	Ameixa	800	6,1
7	10	Abacaxi	85	5,88
8	11	Bacuri	250	15,25
9	12	Gravi...	100	25,1
10	13	Melan...	65	2,88
11	14	Melao	77	8,25
12	16	Limão	152	3,88

```
-- Cria a tabela Pessoa no banco Loja
CREATE TABLE Pessoa (id_pessoa int IDENTITY(1,1) PRIMARY KEY,
nome_pessoa varchar(255) not null,
logradouro varchar(255) null,
cidade varchar(255) null,
estado char(2) null,
telefone varchar(11),
email varchar(255)
);
GO

INSERT INTO Pessoa (nome_pessoa, logradouro, cidade, estado, telefone, email)
VALUES ('Wallace Tavares', 'Rua Ana Augusta', 'São Paulo', 'SP', '11991052555',
'w.tavares@gmail.com'),
('Ricardo Alarcon', 'Rua Dr Cristiano', 'São Paulo', 'SP', '11991054554',
'r.alarcon@gmail.com')

INSERT INTO Pessoa (nome_pessoa, logradouro, cidade, estado, telefone, email)
VALUES ('Tintas Carrão', 'Rua Roxa', 'São Paulo', 'SP', '1127814777',
'tintas@outlook.com'),
('Pneus Trindade', 'Rua Dom Pedro Silva', 'São Paulo', 'SP', '1127814777',
'p.trindade@gmail.com')

select * from Pessoa;
```

	id_pessoa	nome_pessoa	logradouro	cidade	estado	telefone	email
1	1	Rodrigo Alarcon	Rua Dom Pedro Silva	São Paulo	SP	11991051556	oliveira.alarcon@outlook.com
2	2	Pietro Rdorigo Kiraly Alarcon	Rua Dom Pedro Silva	São Paulo	SP	11991051556	pietro.alarcon@gmail.com
3	3	Cynthia Vanessa	Rua Dom Pedro Silva	São Paulo	SP	11991051556	cynthia.kiraly@gmail.com
4	5	Alarcon Designer	Rua Terra Roxa	São Paulo	SP	1127814777	alarconDesigner@outlook.com
5	6	PB Pinturas	Rua Dom Pedro Silva	São Paulo	SP	1127814777	pb.pinturas@gmail.com
6	7	Wallace Tavares	Rua Ana Augusta	São Paulo	SP	11991052555	w.tavares@gmail.com
7	8	Ricardo Alarcon	Rua Dr Cristiano	São Paulo	SP	11991054554	r.alarcon@gmail.com
8	9	Tintas Carrão	Rua Roxa	São Paulo	SP	1127814777	tintas@outlook.com
9	10	Pneus Trindade	Rua Dom Pedro Silva	São Paulo	SP	1127814777	p.trindade@gmail.com

```
-- Cria a tabela Pessoa Física no banco Loja
CREATE TABLE Pessoa_Fisica (
id_pessoa_fisica int IDENTITY(1,1) not null,
id_pessoa int not null,
tipo_documento varchar(20) not null,
numero_documento varchar(50) not null,
data_expedicao date not null,
data_validade date null,
CONSTRAINT PK_Pessoa_Documento PRIMARY KEY (id_pessoa_fisica),
CONSTRAINT FK_Pessoa_Documento FOREIGN KEY (id_pessoa)
REFERENCES Pessoa(id_pessoa)
);
GO

INSERT INTO Pessoa_Fisica (id_pessoa, tipo_documento, numero_documento,
data_expedicao, data_validade)
VALUES (7, 'CPF', '155094775-40', '2003/05/23', ''),
(8, 'CPF', '033055785-60', '2013/05/23', '');

select * from Pessoa_Fisica;
```

Resultados		Mensagens				
	id_pessoa_fisica	id_pessoa	tipo_documento	numero_documento	data_expedicao	data_validade
1	1	1	CPF	022094775-40	2003-05-23	1900-01-01
2	2	2	CPF	022094775-40	2003-05-23	1900-01-01
3	3	3	CPF	022055785-50	2013-05-23	1900-01-01
4	6	7	CPF	155094775-40	2003-05-23	1900-01-01
5	7	8	CPF	033055785-60	2013-05-23	1900-01-01

```

-- Cria a tabela Pessoa Jurídica no banco Loja
CREATE TABLE Pessoa_Juridica (
id_pessoa_juridica int IDENTITY(1,1) not null,
id_pessoa int not null,
tipo_documento varchar(20) not null,
numero_documento varchar(50) not null,
data_expedicao date not null,
data_validade date null,
CONSTRAINT PK_Pessoa_Juridica PRIMARY KEY (id_pessoa_juridica),
CONSTRAINT FK_Pessoa_Juridica FOREIGN KEY (id_pessoa)
REFERENCES Pessoa(id_pessoa)
);
GO

INSERT INTO Pessoa_Juridica (id_pessoa, tipo_documento, numero_documento,
data_expedicao, data_validade)
VALUES (9, 'CNPJ', '21263258000144', '2013/05/23', ''),
(10, 'CNPJ', '25263558000199', '2013/05/23', '');

UPDATE Pessoa_Juridica
SET numero_documento = '48686801000999'
WHERE id_pessoa = 10;

select * from Pessoa_Juridica;

```

	id_pessoa_juridica	id_pessoa	tipo_documento	numero_documento	data_expedicao	data_validade
1	1	5	CNPJ	25686801000183	2013-05-23	1900-01-01
2	2	6	CNPJ	18263258000146	2013-05-23	1900-01-01
3	3	9	CNPJ	21263258000144	2013-05-23	1900-01-01
4	4	10	CNPJ	48686801000999	2013-05-23	1900-01-01

```

-- Movimentação de compra e venda
CREATE TABLE Movimentacao (
id_movimento int IDENTITY(1,1) not null,
id_usuario int not null,
id_pessoa int not null,
id_produto int not null,
quantidade integer not null,
tipo char(1),
valor_unitario float
CONSTRAINT PK_Movimentacao PRIMARY KEY (id_movimento),
CONSTRAINT FK_User FOREIGN KEY (id_usuario)
REFERENCES Usuario(id_user),
CONSTRAINT FK_People FOREIGN KEY (id_pessoa)
REFERENCES Pessoa(id_pessoa),
CONSTRAINT FK_Product FOREIGN KEY (id_produto)
REFERENCES Produto(id_produto)
);
GO

INSERT INTO Movimentacao(id_usuario, id_pessoa, id_produto, quantidade, tipo,
valor_unitario)
VALUES (1, 5, 16, 52, 'E', 5.90)

DELETE FROM Movimentacao WHERE id_pessoa=2;

UPDATE Movimentacao
SET id_pessoa = 5
WHERE id_movimento = 2;

select * from Movimentacao;

```

	id_movimento	id_usuario	id_pessoa	id_produto	quantidade	tipo	valor_unitario
1	1	1	1	4	20	S	4,75
2	3	1	3	6	15	S	4,1
3	4	1	6	5	20	E	4,75
4	6	2	5	5	150	E	8,9
5	8	1	5	16	52	E	5,9

```
-- Efetuar as seguintes consultas sobre os dados inseridos:
-- Dados completos de pessoas físicas.
select * from Pessoa_Fisica;
-- Dados completos de pessoas jurídica.
select * from Pessoa_Juridica;

-- Movimentações de entrada, com produto, fornecedor, quantidade, preço unitário e valor total.
select p.nome_pessoa, m.id_produto, m.id_pessoa, m.quantidade, m.valor_unitario,
(m.quantidade * m.valor_unitario) valor_total
from Movimentacao as m
Join Pessoa as p on (m.id_pessoa = p.id_pessoa)
where m.tipo = 'E';
```

	nome_pessoa	id_produto	id_pessoa	quantidade	valor_unitario	valor_total
1	PB Pinturas	5	6	20	4,75	95
2	Alarcon Designer	5	5	150	8,9	1335
3	Alarcon Designer	16	5	52	5,9	306,8

```
-- Movimentações de entrada, com produto, cliente, quantidade, preço unitário e valor total.
select p.nome_pessoa, m.id_produto, m.id_pessoa, m.quantidade, m.valor_unitario,
(m.quantidade * m.valor_unitario) valor_total
from Movimentacao as m
Join Pessoa as p on (m.id_pessoa = p.id_pessoa)
where m.tipo = 'S';
```

	nome_pessoa	id_produto	id_pessoa	quantidade	valor_unitario	valor_total
1	Rodrigo Alarcon	4	1	20	4,75	95
2	Cynthia Vanessa	6	3	15	4,1	61,5

```
-- Movimentações somatória de quantidade por produto agrupado.
select id_produto, SUM(quantidade) Quantidade, SUM(quantidade * valor_unitario)
QuantidadeValor
from Movimentacao
group by id_produto;
```

	id_produto	Quantidade	QuantidadeValor
1	4	20	95
2	5	170	1430
3	6	15	61,5
4	16	52	306,8


```
-- Valor médio do produto por fornecedor
select id_produto, SUM(quantidade * valor_unitario) / SUM(quantidade) as
MediaCompra
from Movimentacao
where tipo = 'E'
group by id_produto;
```

	id_produto	MediaCompra
1	5	8,41176470588235
2	16	5,9

```
-- Valor médio do produto por cliente
select id_produto, SUM(quantidade * valor_unitario) / SUM(quantidade) as
MediaVenda
from Movimentacao as m
where tipo = 'S'
group by id_produto;
```

	id_produto	MediaVenda
1	4	4,75
2	6	4,1

- a) Como são implementadas as diferentes cardinalidades, basicamente 1X1, 1XN ou NxN, em um banco de dados relacional?

Cardinalidade 1 x 1 significa que um relacionamento único de registros entre tabelas. Aqui podemos dizer que o relacionamento é obrigatório.

Cardinalidade 1 x N significa que para cada registro de uma tabela podem existir um ou vários registros na tabela relacionada.

Cardinalidade N x X significa que múltiplos registros se relacionam com múltiplos registros.

- b) Que tipo de relacionamento deve ser utilizado para representar o uso de herança em bancos de dados relacionais?

1 x N -> Um para muitos

- c) Como o SQL Server Management Studio permite a melhoria da produtividade nas tarefas relacionadas ao gerenciamento do banco de dados?

A produtividade se dá com o atalho das teclas ctrl + espaço, botão direito do mouse para criar tabelas, seleções etc.

2º Procedimento – Alimentando a Base

Inserir neste campo, de forma organizada, todos os códigos do roteiro do 2º Procedimento da Atividade Prática, os resultados da execução do código e a Análise e Conclusão:

a) Quais as diferenças no uso de sequence e identity?

Sequence é criado separadamente como uma função interna que calcula a sequência dos campos chaves. Já o identity é atribuído diretamente no código de criação da estrutura do Banco de Dados.

b) Qual a importância das chaves estrangeiras para a consistência do banco?

Só pode-se cadastrar uma chave estrangeira se houver antes um cadastro da chave primária que faz referência a forenkey.

c) Quais operadores do SQL pertencem à álgebra relacional e quais são definidos no cálculo relacional?

Álgebra Relacional -> Seleção, Projeção, Renomeação, União, Intersecção, Diferença,

Cálculo Relacional -> Tuplas, Quantificadores, Domínios

d) Como é feito o agrupamento em consultas, e qual requisito é obrigatório?

Pelo comando SUM e obrigatoriamente o Group By

Observe que os tópicos acima seguem exatamente o que está na Atividade Prática exigida.

Conclusão

Podemos avaliar e mensurar a importância de se utilizar um SGBD para manter e manipular os dados que são parte essencial para qualquer sistema e ou aplicativo. Aprender a importância dos relacionamentos entre as tabelas com suas chaves primárias(primaryKey) e estrangeiras(forenkey).