



UNIVERSIDADE ESTÁCIO DE SÁ

Relatório de Atividade Prática | Nivel 2 | Parte 2

IDELMAR FERNANDO DE SOUZA

SANTA ROSA
2024

1. Atividade Prática Nível 2, Parte 2 – Banco Relacional

Objetivo da Prática

Tem como objetivo desenvolver um banco relacional para consultas em SQL;

2. A seguir as consultas usadas:

PESSOA:

```
CREATE TABLE pessoa (
    idpessoa INT IDENTITY(1,1) PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    telefone VARCHAR(20),
    email VARCHAR(40),
    tipo VARCHAR(1) NOT NULL,
    estado INT,
    cidade INT,
    bairro INT,
    logradouro INT,
    cep INT,
);

ALTER TABLE pessoa
ADD CONSTRAINT FK_pessoa_id_est
FOREIGN KEY (estado) REFERENCES estado (idestado);

-----

ALTER TABLE pessoa
ADD CONSTRAINT FK_pessoa_id_cid
FOREIGN KEY(cidade) REFERENCES cidade (idcidade);

-----

ALTER TABLE pessoa
ADD CONSTRAINT FK_pessoa_id_bai
FOREIGN KEY(bairro) REFERENCES bairro (idbairro);

-----

ALTER TABLE pessoa
ADD CONSTRAINT FK_pessoa_pes_log
FOREIGN KEY(logradouro) REFERENCES logradouro (idlogradouro);

-----

ALTER TABLE pessoa
ADD CONSTRAINT FK_pessoa_pes_cep
FOREIGN KEY(cep) REFERENCES cep (idcep);

-----

INSERT INTO pessoa (nome,telefone,email,tipo,estado,cidade,      bairro,
logradouro, cep)
VALUES
('JJC', '35127845','fruteirajjc@gmail.com','J',21,3,3,3,3),
('Joao', '998965214','joaopedro@gmail.com','F',21,1,1,1,1),
('Ana Paula', '999111222', 'anapaula@gmail.com','F',21,1,1,1,1),
('Carlos', '888222333', 'carlos@bol.com','F',21,1,2,2,2);
```

PESSOA FISICA:

```
CREATE TABLE pessoafisica (  
    idpessoafisica INT IDENTITY(1,1) PRIMARY KEY,  
    idpessoa INT NOT NULL,  
    cpf VARCHAR(20) NOT NULL,  
);  
  
ALTER TABLE pessoafisica  
ADD CONSTRAINT FK_pessoafisica_id_pes  
FOREIGN KEY (idpessoa) REFERENCES pessoa (idpessoa);  
  
INSERT INTO dbo.pessoafisica (cpf,idpessoa)  
VALUES  
    ('99999999954',2),  
    ('23476511555',3),  
    ('12344486954',4);
```

PESSOA JURIDICA:

```
CREATE TABLE pessoajuridica (  
    idpessoajuridica INT IDENTITY(1,1) PRIMARY KEY,  
    idpessoa INT,  
    cnpj VARCHAR(20) NOT NULL,  
);  
  
ALTER TABLE pessoajuridica  
ADD CONSTRAINT FK_pessoajuridica_id_pes  
FOREIGN KEY (idpessoa) REFERENCES pessoa (idpessoa);  
  
INSERT INTO dbo.pessoajuridica (cnpj,idpessoa)  
VALUES ('12345678000190',1)
```

PRODUTO:

```
CREATE TABLE produto (  
    idproduto INT IDENTITY(1,1) PRIMARY KEY,  
    produto VARCHAR(20) NOT NULL ,  
    estoque INT NOT NULL,  
    preco Decimal (10,2) NOT NULL,  
);  
  
INSERT INTO produto (produto, estoque, preco)  
VALUES  
    ('banana', 100, 5.00),  
    ('maçã', 400, 2.00),  
    ('laranja', 800, 4.00);
```

USUARIO:

```
CREATE TABLE usuario (  
    idusuario INT IDENTITY(1,1) PRIMARY KEY,  
    login VARCHAR(50) NOT NULL,  
    senha VARCHAR(25) NOT NULL,  
);
```

```
INSERT INTO usuario (login, senha)  
VALUES ('op1', 'op1'),  
      ('op2', 'op2'),  
      ('op3', 'op3');
```

NEGOCIAÇÃO:

```
CREATE TABLE negociacao (  
    idnegociacao INT IDENTITY(1,1) PRIMARY KEY,  
    tipo VARCHAR(10) NOT NULL,  
    valorUnitario NUMERIC (10,2) NOT NULL,  
    usuario INT,  
    produto INT,  
    pessoa INT,  
    quantidade INT  
);
```

```
-----  
ALTER TABLE negociacao  
ADD CONSTRAINT FK_negociacao_id_usu  
FOREIGN KEY (usuario) REFERENCES usuario (idusuario);
```

```
-----  
ALTER TABLE negociacao  
ADD CONSTRAINT FK_negociacao_id_pro  
FOREIGN KEY (produto) REFERENCES produto (idproduto);
```

```
-----  
ALTER TABLE negociacao  
ADD CONSTRAINT FK_negociacao_id_pes  
FOREIGN KEY (pessoa) REFERENCES pessoa (idpessoa);  
-----
```

```
INSERT INTO negociacao  
(tipo, valorUnitario, usuario, produto, pessoa, quantidade)  
VALUES  
    ('V', 4, 1, 1, 2, 20),  
    ('V', 2, 1, 3, 2, 24),  
    ('V', 3, 2, 3, 3, 14),  
    ('C', 51, 2, 4, 15),  
    ('C', 4, 1, 2, 4, 27);
```

ESTADO:

```
CREATE TABLE estado (  
    idestado INT IDENTITY(1,1) PRIMARY KEY,  
    estado VARCHAR(50) NULL,  
    sigla VARCHAR(2) NULL,  
);
```

```

INSERT INTO estado (estado, sigla)
VALUES
    ('Acre', 'AC'),
    ('Alagoas', 'AL'),
    ('Amapá', 'AP'),
    ('Amazonas', 'AM'),
    ('Bahia', 'BA'),
    ('Ceará', 'CE'),
    ('Distrito Federal', 'DF'),
    ('Espírito Santo', 'ES'),
    ('Goiás', 'GO'),
    ('Maranhão', 'MA'),
    ('Mato Grosso', 'MT'),
    ('Mato Grosso do Sul', 'MS'),
    ('Minas Gerais', 'MG'),
    ('Pará', 'PA'),
    ('Paraíba', 'PB'),
    ('Paraná', 'PR'),
    ('Pernambuco', 'PE'),
    ('Piauá', 'PI'),
    ('Rio de Janeiro', 'RJ'),
    ('Rio Grande do Norte', 'RN'),
    ('Rio Grande do Sul', 'RS'),
    ('Rorônia', 'RO'),
    ('Roraima', 'RR'),
    ('Santa Catarina', 'SC'),
    ('São Paulo', 'SP'),
    ('Sergipe', 'SE'),
    ('Tocantins', 'TO');

```

CIDADE:

```

CREATE TABLE cidade (
    idcidade INT IDENTITY(1,1) PRIMARY KEY,
    cidade VARCHAR(50) NULL,
    estado INT
);

ALTER TABLE cidade
ADD CONSTRAINT FK_cidade_id_est
FOREIGN KEY(estado) REFERENCES estado(idestado)

INSERT INTO cidade (cidade,estado)
VALUES
    ('Santa Rosa',1),
    ('Alegrete',1),
    ('Torres',1);

```

BAIRRO:

```

CREATE TABLE bairro (
    idbairro INT IDENTITY(1,1) PRIMARY KEY,
    bairro VARCHAR(50) NULL,
    cidade INT
);

ALTER TABLE bairro

```

```
ADD CONSTRAINT FK_bairro_id_cid
FOREIGN KEY(cidade) REFERENCES cidade(idcidade)
```

```
INSERT INTO bairro (bairro,cidade)
VALUES ('Cruzeiro',1),
       ('Centro',1),
       ('Centro',3);
```

LOGRADOURO:

```
CREATE TABLE logradouro (
    idlogradouro INT IDENTITY(1,1) PRIMARY KEY,
    logradouro VARCHAR(50) NULL,
    bairro INT
);
```

```
ALTER TABLE logradouro
ADD CONSTRAINT FK_logradouro_id_bai
FOREIGN KEY(bairro) REFERENCES bairro (idbairro)
```

```
INSERT INTO logradouro (logradouro,bairro)
VALUES ('Av. Expedicionario Weber',2),
       ('Rua Bela Uniao',1),
       ('Rua Coronel Pacheco',3);
```

CEP:

```
CREATE TABLE cep (
    idcep INT IDENTITY(1,1) PRIMARY KEY,
    cep INT NOT NULL,
    logradouro INT
);
```

```
ALTER TABLE cep
ADD CONSTRAINT FK_cep_id_log
FOREIGN KEY(logradouro) REFERENCES logradouro (idlogradouro)
```

```
INSERT INTO cep (cep,logradouro)
VALUES ('98900000',1),
       ('98785200',2),
       ('95560975',3);
```

CONSULTAS:

```
--dados pessoa fisica
SELECT pes.nome,
pef.cpf,pes.telefone,pes.email,est.estado,cid.cidade,bai.bairro,log.logradouro,cep.cep
FROM pessoafisica pef
INNER JOIN dbo.pessoa pes on (pes.idpessoa = pef.idpessoa and pes.tipo = 'F')
LEFT JOIN dbo.estado est on (est.idestado = pes.estado)
LEFT JOIN dbo.cidade cid on (cid.idcidade = pes.cidade)
LEFT JOIN dbo.bairro bai on (bai.idbairro = pes.bairro)
LEFT JOIN dbo.logradouro log on (log.idlogradouro = pes.logradouro)
LEFT JOIN dbo.cep cep on (cep.idcep = pes.cep)
--dados pessoa juridica
```

```

SELECT pes.nome,
pej.cnpj,pes.telefone,pes.email,est.estado,cid.cidade,bai.bairro,log.lograd
ouro,cep.cep
FROM pessoajuridica pej
    INNER JOIN dbo.pessoa pes on (pes.idpessoa = pej.idpessoa and pes.tipo =
'J')
    LEFT JOIN dbo.estado est on (est.idestado = pes.estado)
    LEFT JOIN dbo.cidade cid on (cid.idcidade = pes.cidade)
    LEFT JOIN dbo.bairro bai on (bai.idbairro = pes.bairro)
    LEFT JOIN dbo.logradouro log on (log.idlogradouro = pes.logradouro)
    LEFT JOIN dbo.cep cep on (cep.idcep = pes.cep)
--negociações compra
SELECT neg.tipo,prod.produto,usu.login as comprador,pes.nome as
fornecedor,neg.quantidade,neg.valorUnitario,
(neg.quantidade * neg.valorUnitario) as valor_total
FROM dbo.negociacao neg
INNER JOIN dbo.usuario usu on (usu.idusuario = neg.usuario)
INNER JOIN dbo.pessoa pes on (pes.idpessoa = neg.pessoa)
INNER JOIN produto prod on (prod.idproduto = neg.produto)
WHERE neg.tipo = 'C'

--negociações venda
SELECT neg.tipo,prod.produto,usu.login as comprador,pes.nome as
fornecedor,neg.quantidade,neg.valorUnitario,
(neg.quantidade * neg.valorUnitario) as valor_total
FROM dbo.negociacao neg
INNER JOIN dbo.usuario usu on (usu.idusuario = neg.usuario)
INNER JOIN dbo.pessoa pes on (pes.idpessoa = neg.pessoa)
INNER JOIN produto prod on (prod.idproduto = neg.produto)
WHERE neg.tipo = 'V'

--valor total compras por produto
SELECT prod.produto,sum(neg.quantidade * neg.valorUnitario) as
total_compras
FROM dbo.negociacao neg
INNER JOIN produto prod on (prod.idproduto = neg.produto)
WHERE neg.tipo = 'C'
GROUP by prod.produto

--valor total de vendas por produto
SELECT prod.produto,sum(neg.quantidade * neg.valorUnitario) as
total_compras
FROM dbo.negociacao neg
INNER JOIN produto prod on (prod.idproduto = neg.produto)
WHERE neg.tipo = 'V'
GROUP by prod.produto

--usuario que nao compraram
SELECT usu.login
FROM usuario usu
WHERE NOT EXISTS
    (SELECT neg.idnegociacao from negociacao neg where neg.usuario =
usu.idusuario and neg.tipo = 'C')

--valor total compra agrupado por usuario
SELECT usu.login,(neg.quantidade * neg.valorUnitario) as valor_total
FROM dbo.usuario usu
INNER JOIN dbo.negociacao neg on (neg.usuario = usu.idusuario)
WHERE neg.tipo = 'C'

```

```

GROUP BY usu.login,neg.quantidade,neg.valorUnitario

--valor total venda agrupado por usuario
SELECT usu.login,(neg.quantidade * neg.valorUnitario) as valor_total
from dbo.usuario usu
INNER JOIN dbo.negociacao neg on (neg.usuario = usu.idusuario)
WHERE neg.tipo = 'V'
GROUP BY usu.login,neg.quantidade,neg.valorUnitario

--Valor médio de venda por produto
SELECT prod.produto,ROUND(SUM(neg.valorUnitario * neg.quantidade) /
SUM(neg.quantidade),2) AS media_venda
FROM negociacao neg
INNER JOIN produto prod on (prod.idproduto = neg.produto)
WHERE neg.tipo = 'V'
GROUP BY prod.produto;

```

3. Conclusão:

Quais as diferenças no uso de sequence e identity?

A sequência gera automaticamente, porém com maior flexibilidade que o identify , por exemplo se definir qual incremento usar, também são mais escaláveis. Já o identify gera automaticamente um valor único e autoincremento e tem ligação direta com a tabela referente.

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

São as chaves estrangeiras que garantem o relacionamento das tabelas e a integridade dos dados pois os atributos estão interligados;

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

Os operadores mais usados são o 'select', 'join','union', tendo também o 'project','intersect' e o 'cross union',no calculo relacional temos o 'project','join', 'union','exists','in' e "not in";

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

O agrupamento é feito usando o 'group by' e deve ser escolhido obrigatoriamente qual coluna deseja agrupar, ele é solicitado ao usar na consulta, o "sum","count","max","min" entre outros.