



Estácio

Vamos Manter as Informações?

Jonathan da Silva Araujo - 202205178111

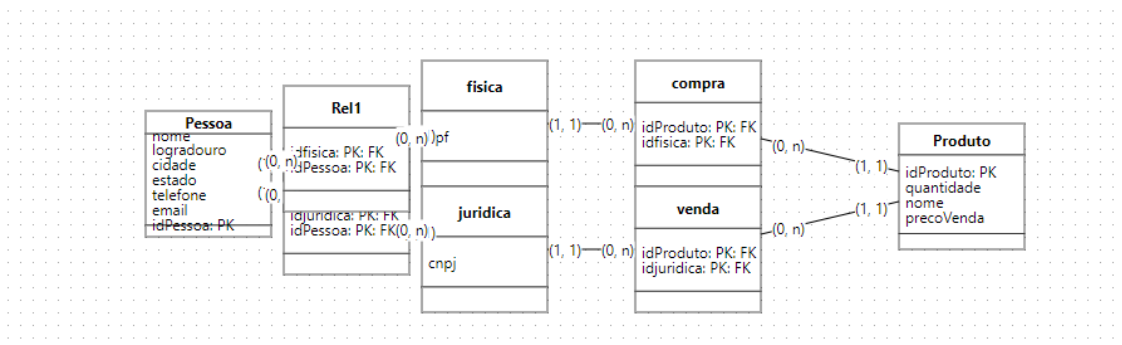
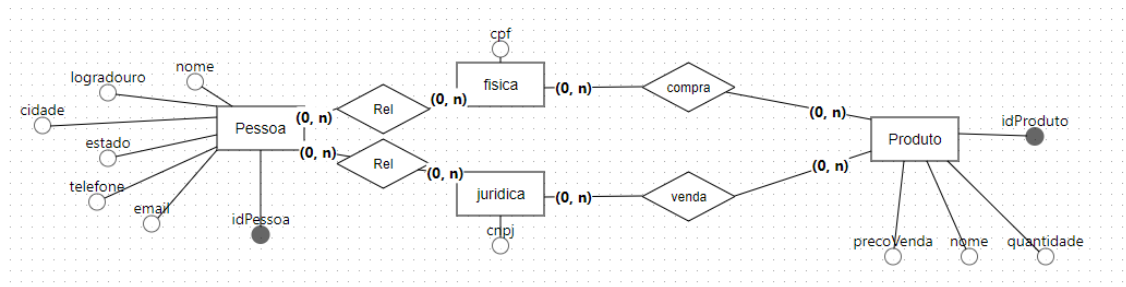
Campus Nova Iguaçu

Vamos manter as informações? (RPG0015) – 9001 – 2023.2 (3º Semestre)

Objetivo da Prática

O projeto consiste em realizar a modelagem da base de dados para um sistema simples e implementá-la a um banco de dados simples, utilizando a sintaxe SQL na criação das estruturas do banco além de utilizar também para consulta e manipulação de dados.

1º Procedimento – Criando o Banco de Dados



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

As diferentes cardinalidades são implementadas usando chaves primárias e chaves estrangeiras. Para implementar um relacionamento 1X1 basta combinar as duas entidades em uma única tabela. Para implementar um relacionamento 1XN pode-se adicionar uma chave estrangeira à tabela que tem cardinalidade máxima N. Para implementar um relacionamento NXN cria-se uma nova entidade para que seja armazenado os dados do relacionamento.

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

O tipo de relacionamento deve ser o especialização/generalização, onde uma entidade é subtítulo da outra entidade.

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

De diversas maneiras, tais como: automatização de tarefas de gerenciamento de bd, criação de scripts, consultas, fornece uma visão geral do banco de dados entre outros.

2º Procedimento – Alimentando a Base

C: > Users > jonat > Desktop > Projetos > Estácio > Mundo 3 > Missão Prática 2 > MissaoPratica2-CRIACAO_BD.sql

You, há 2 semanas | 1 author (You)

You, há 2 semanas • Projeto concluído.

```
1 CREATE SEQUENCE id_seq_pessoa START WITH 1;
2 CREATE SEQUENCE id_seq_movimentacoes START WITH 1;
3
4 CREATE TABLE Pessoa
5 (
6     idPessoa INT DEFAULT nextval('id_seq_pessoa') PRIMARY KEY,
7     nome VARCHAR,
8     logradouro VARCHAR,
9     cidade VARCHAR,
10    estado VARCHAR,
11    telefone VARCHAR,
12    email VARCHAR
13 );
14
15 CREATE TABLE produto
16 (
17     idProduto INT PRIMARY KEY,
18     quantidade INT,
19     nome VARCHAR,
20     precoVenda NUMERIC
21 );
22
23 CREATE TABLE Fisica
24 (
25     idPessoaFisica INT DEFAULT nextval('id_seq_pessoa') PRIMARY KEY,
26     nome VARCHAR,
27     logradouro VARCHAR,
28     cidade VARCHAR,
29     estado VARCHAR,
30     telefone VARCHAR,
31     email VARCHAR,
32     idpessoa int,
33     cpf VARCHAR
34 );
35
36 CREATE TABLE Jurídica
37 (
38     idPessoaJuridica INT DEFAULT nextval('id_seq_pessoa') PRIMARY KEY,
```

```

36 CREATE TABLE Juridica
37 (
38     idPessoaJuridica INT DEFAULT nextval('id_seq_pessoa') PRIMARY KEY,
39     idPessoa INT,
40     dadosPessoa VARCHAR,
41     cnpj VARCHAR,
42     FOREIGN KEY (idPessoa) REFERENCES Pessoa(idPessoa)
43 );
44
45 CREATE TABLE Compra
46 (
47     idCompra INT PRIMARY KEY,
48     idProduto INT,
49     idPessoaFisica INT,
50     FOREIGN KEY (idProduto) REFERENCES Produto(idProduto),
51     FOREIGN KEY (idPessoaFisica) REFERENCES fisica(idPessoaFisica)
52 );
53
54 CREATE TABLE Venda
55 (
56     idVenda INT PRIMARY KEY,
57     idProduto INT,
58     idPessoaJuridica INT,
59     FOREIGN KEY (idProduto) REFERENCES Produto(idProduto),
60     FOREIGN KEY (idPessoaJuridica) REFERENCES juridica(idPessoaJuridica)
61 );
62
63 CREATE TABLE usuarios
64 (
65     idUsuario INT,
66     login VARCHAR,
67     senha VARCHAR
68 );
69

```

```

70 CREATE TABLE movimentacoes
71 (
72     idMovimento INT DEFAULT nextval('id_seq_movimentacoes'),
73     idUsuario INT,
74     idPessoa INT,
75     idProduto INT,
76     quantidade INT,
77     tipo VARCHAR,
78     valorUnitario DECIMAL
79 );

```

C: > Users > jonat > Desktop > Projetos > Estácio > Mundo 3 > Missão Prática 2 > MissaoPratica2-SCRIPTS_CONSULTAS.sql
You, há 2 semanas | 1 author (You)

1 --DADOS COMPLETOS DE PESSOAS FÍSICAS: You, há 2 semanas • Projeto concluído.

2 SELECT * FROM fisica;

3

4 --DADOS COMPLETOS DE PESSOAS JURÍDICAS:

5 SELECT * FROM juridica;

6

7 --MOVIMENTAÇÕES DE ENTRADA, COM PRODUTO, QTD, PREÇO UN E VALOR TOTAL:

8 SELECT

9 produto.nome,
10 juridica.nome,
11 movimentacoes.quantidade,
12 movimentacoes.valorunitario,
13 movimentacoes.quantidade * movimentacoes.valorunitario AS valortotal

14 FROM

15 produto,
16 juridica,
17 movimentacoes

18 WHERE

19 movimentacoes.tipo = 'E';

20

21 --MOVIMENTAÇÕES DE SAÍDA, COM PRODUTO, QTD, PREÇO UN E VALOR TOTAL:

22 SELECT

23 produto.nome,
24 juridica.nome,
25 movimentacoes.quantidade,
26 movimentacoes.valorunitario,
27 movimentacoes.quantidade * movimentacoes.valorunitario AS valortotal

28 FROM

29 produto,
30 juridica,
31 movimentacoes

32 WHERE

33 movimentacoes.tipo = 'S';

34

```
35 --VALOR TOTAL DAS ENTRADAS AGRUPADAS POR PRODUTOS:
36 SELECT
37     movimentacoes.idproduto,
38     SUM(movimentacoes.quantidade) AS total_quantidade,
39     SUM(movimentacoes.valorunitario * movimentacoes.quantidade) AS total_valortotal
40 FROM
41     movimentacoes
42 WHERE
43     movimentacoes.tipo = 'E'
44 GROUP BY
45     movimentacoes.idproduto;
46
47
48 --VALOR TOTAL DAS SAÍDAS AGRUPADAS POR PRODUTOS:
49 SELECT
50     movimentacoes.idproduto,
51     SUM(movimentacoes.quantidade) AS total_quantidade,
52     SUM(movimentacoes.valorunitario * movimentacoes.quantidade) AS total_valortotal
53 FROM
54     movimentacoes
55 WHERE
56     movimentacoes.tipo = 'S'
57 GROUP BY
58     movimentacoes.idproduto;
59
60 -- OPERADORES QUE NÃO EFETUARAM MOVIMENTAÇÕES DE COMPRA:
61 SELECT
62     fisica.idpessoafisica,
63     fisica.nome
64 FROM
65     fisica
66 LEFT JOIN
67     movimentacoes ON fisica.idpessoafisica = movimentacoes.idpessoa
68 WHERE
69     movimentacoes.idpessoa IS NULL;
```

```

71 --VALOR TOTAL DE ENTRADA, AGRUPADO POR OPERADOR:
72 SELECT
73     movimentacoes.idusuario,
74     SUM(movimentacoes.quantidade) AS total_quantidade,
75     SUM(movimentacoes.valorunitario * movimentacoes.quantidade) AS total_valortotal
76 FROM
77     movimentacoes
78 WHERE
79     movimentacoes.tipo = 'E'
80 GROUP BY
81     movimentacoes.idusuario;
82 --VALOR TOTAL DE SAÍDA, AGRUPADO POR OPERADOR:
83 SELECT
84     movimentacoes.idusuario,
85     SUM(movimentacoes.quantidade) AS total_quantidade,
86     SUM(movimentacoes.valorunitario * movimentacoes.quantidade) AS total_valortotal
87 FROM
88     movimentacoes
89 WHERE
90     movimentacoes.tipo = 'S'
91 GROUP BY
92     movimentacoes.idusuario;
93 --VALOR MÉDIO DE VENDA POR PRODUTO:
94 SELECT
95     movimentacoes.idproduto,
96     SUM(movimentacoes.quantidade) AS total_quantidade,
97     SUM(movimentacoes.valorunitario * movimentacoes.quantidade) AS total_valortotal,
98     ROUND(SUM(movimentacoes.valorunitario * movimentacoes.quantidade) / SUM(movimentacoes.quantidade), 2) AS media_valores
99 FROM
100     movimentacoes
101 WHERE
102     movimentacoes.tipo = 'S'
103 GROUP BY
104     movimentacoes.idproduto;
105

```

```

C: > Users > jonat > Desktop > Projetos > Estácio > Mundo 3 > Missão Prática 2 > MissaoPratica2-ATUALIZA_PJ.sql
You, há 2 semanas | 1 author (You)
1  UPDATE juridica
2  SET nome = pessoa.nome,
3      logradouro = pessoa.logradouro,
4      cidade = pessoa.cidade,
5      estado = pessoa.estado,
6      telefone = pessoa.telefone,
7      email = pessoa.email
8  FROM pessoa
9  WHERE juridica.idpessoa = pessoa.idpessoa;

```

```

C: > Users > jonat > Desktop > Projetos > Estácio > Mundo 3 > Missão Prática 2 > MissaoPratica2-ATUALIZA_PF.sql
You, há 2 semanas | 1 author (You)
1  UPDATE fisica
2  SET nome = pessoa.nome,
3      logradouro = pessoa.logradouro,
4      cidade = pessoa.cidade,
5      estado = pessoa.estado,
6      telefone = pessoa.telefone,
7      email = pessoa.email
8  FROM pessoa
9  WHERE fisica.idpessoa = pessoa.idpessoa;

```

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

O recurso **sequence** pode ser compartilhado por várias tabelas, os valores são gerados em memória e podem ser reiniciados com um período de tempo pré-estipulado. Já o recurso **identity** é uma propriedade empregada somente a uma

coluna, os valores não podem ser reiniciados e são gerados em disco, o que pode afetar negativamente o desempenho.

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

Uma vez que as chaves estrangeiras são usadas para estabelecer relacionamentos entre tabelas, elas garantem que os dados relacionados sejam consistentes.

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

Os operadores do SQL que pertencem à álgebra relacional são: **SELECT, FROM, WHERE, JOIN, GROUP BY, HAVING, ORDER BY**. Já os que são definidos no cálculo relacional são: **DISTINCT, UNION, INTERSECT, EXCEPT**.

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

É feito usando a cláusula **GROUP BY**, a qual especifica as colunas pelas quais os registros serão agrupados. O requisito obrigatório é que as colunas (especificadas na cláusula **GROUP BY**) sejam colunas de chave primária ou de agregação.

Conclusão

O referido trabalho teve por finalidade realizar a modelagem do banco de dados com o intuito de criar um modelo de dados o qual representa os dados necessários para um sistema de informação, sendo este uma representação abstrata do banco de dados.

Após a modelagem realizada, as tabelas criadas foram alimentadas com dados básicos do sistema ao incluir dados nas tabelas e, após isso, foi efetuado consultas sobre os dados inseridos constatando o correto funcionamento do banco de dados.