

Centro Universitário Estácio do Ceará - Campus Centro

Curso: Desenvolvimento Full Stack

Disciplina: Nível 1: Iniciando o Caminho Pelo Java

Número da Turma: RPG0015

Semestre Letivo: 3

Integrantes: Fernando Rocha Fonteles Filho

Repositorio Git: https://github.com/fernandorff/EstacioFullStack_Mundo3_Nivel2

Título da Prática: Procedimento | Criando o Banco de Dados

Objetivos da Prática:

- Identificar os requisitos de um sistema e transformá-los no modelo adequado.
- Utilizar ferramentas de modelagem para bases de dados relacionais.
- Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
- Explorar a sintaxe SQL na consulta e manipulação de dados (DML)
- No final do exercício, o aluno terá vivenciado a experiência de modelar a base de dados para um sistema simples, além de implementá-la, através da sintaxe SQL, na plataforma do SQL Server.

Códigos solicitados neste roteiro de aula:

- Esquema de criação das tabelas do banco de dados

```

CREATE TABLE pessoa (
    id_pessoa INTEGER IDENTITY NOT NULL PRIMARY KEY,
    nome VARCHAR(255) NULL,
    logradouro VARCHAR(255) NULL,
    cidade VARCHAR(255) NULL,
    estado CHAR(2) NULL,
    telefone VARCHAR(11) NULL,
    email VARCHAR(255) NULL
);

CREATE TABLE pessoa_fisica (
    id_pessoa_fisica INTEGER IDENTITY NOT NULL PRIMARY KEY,
    pessoa_id_pessoa INTEGER NOT NULL CHECK (pessoa_id_pessoa > 0),
    cpf VARCHAR(11) NULL
);

CREATE TABLE pessoa_juridica (
    id_pessoa_juridica INTEGER IDENTITY NOT NULL PRIMARY KEY,
    pessoa_id_pessoa INTEGER NOT NULL CHECK (pessoa_id_pessoa > 0),
    cnpj VARCHAR(20) NULL
);

CREATE TABLE movimento (
    id_movimento INTEGER IDENTITY NOT NULL PRIMARY KEY,
    pessoa_id_pessoa INTEGER NOT NULL CHECK (pessoa_id_pessoa > 0),
    produto_id_produto INTEGER NOT NULL CHECK (produto_id_produto > 0),
    usuario_id_usuario INTEGER NOT NULL CHECK (usuario_id_usuario > 0),
    quantidade INTEGER NULL CHECK (quantidade > 0),
    tipo CHAR(1) NULL,
    valor_unitario NUMERIC(10, 2) NULL
);

CREATE TABLE produto (
    id_produto INTEGER IDENTITY NOT NULL PRIMARY KEY,
    nome VARCHAR(255) NULL,
    quantidade INTEGER NULL CHECK (quantidade > 0),
    precoVenda NUMERIC(10, 2) NULL
);

CREATE TABLE usuario (
    id_usuario INTEGER IDENTITY NOT NULL PRIMARY KEY,
    login VARCHAR(25) NULL,
    senha VARCHAR(25) NULL
);

CREATE INDEX pessoa_fisica_FKIndex1 ON pessoa_fisica(pessoa_id_pessoa);

CREATE INDEX pessoa_juridica_FKIndex1 ON pessoa_juridica(pessoa_id_pessoa);

CREATE INDEX movimento_FKIndex1 ON movimento(usuario_id_usuario);
CREATE INDEX movimento_FKIndex2 ON movimento(produto_id_produto);
CREATE INDEX movimento_FKIndex3 ON movimento(pessoa_id_pessoa);

```

Análise e Conclusão

1. Quais as diferenças no uso de sequence e identity?

A principal diferença entre eles é que sequence é um objeto de banco de dados, enquanto identity é uma propriedade de uma coluna.

Sequence

- É um objeto de banco de dados que armazena um valor sequencial.
- Pode ser usado em qualquer coluna, não apenas em colunas de tipo inteiro.
- Pode ser usado para gerar valores sequenciais para várias colunas.
- Pode ser reinicializado ou revertido.

Identity

- É uma propriedade de uma coluna que permite que o banco de dados gere valores sequenciais para essa coluna.
- Só pode ser usado em colunas de tipo inteiro.
- Só pode ser usado para gerar valores sequenciais para uma coluna.
- Não pode ser reinicializado ou revertido.

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

As chaves estrangeiras são importantes para a consistência do banco de dados porque ajudam a garantir que os dados nas tabelas relacionadas sejam consistentes. Isso é feito exigindo que o valor de uma chave estrangeira corresponda ao valor da chave primária da tabela relacionada.

3. 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
- PROJECT
- JOIN:
- UNION
- INTERSECT
- EXCEPT

Os operadores do SQL que pertencem ao cálculo relacional são:

- COUNT
- MAX
- MIN
- SUM
- AVG

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

O agrupamento em consultas SQL é feito usando a cláusula GROUP BY. A cláusula GROUP BY especifica as colunas que serão usadas para agrupar os resultados da consulta.

O requisito obrigatório para o agrupamento é que a cláusula GROUP BY contenha pelo menos uma coluna. Se a cláusula GROUP BY não conter nenhuma coluna, a consulta retornará uma tabela com uma única linha e uma única coluna, contendo o valor da função de agregação na linha.