



Estácio

Campus: Dorival Caymmi

Nome: Filippe Markouizos Duarte -
202308599615

Curso: Desenvolvimento Full-Stack

Disciplina: Vamos manter as
informações!

Turma: 2023.1 **3º Semestre**

Título:

Modelagem e implementação de um banco de dados simples, utilizando como base o SQL Server

Objetivo:

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.

Análise e Conclusão:

Quais as diferenças no uso de sequence e identity?

Sequence: *Independente da tabela. Pode ser compartilhada por várias tabelas.*

Requer a criação de uma sequência separada. Pode ser usado manualmente

para obter valores próximos sem a necessidade de inserção. A sintaxe específica

pode variar entre SGBDs. Pode ser mais flexível em termos de controle e

reutilização, mas também pode ser mais complexo.

Identity: *Pertence a uma coluna específica em uma tabela. A propriedade de*

identidade é definida diretamente na coluna da tabela. Gera automaticamente

valores incrementais, e normalmente não requer intervenção manual. O

conceito é amplamente suportado, mas a sintaxe pode variar. Mais simples e

direto, adequado para casos em que a complexidade adicional não é necessária.

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

As chaves estrangeiras desempenham um papel crucial na manutenção da

consistência e integridade dos dados em bancos de dados relacionais,

proporcionando uma estrutura robusta para garantir que as relações entre as

tabelas sejam mantidas de maneira precisa e confiável.

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

Ambos os modelos são equivalentes em termos de expressividade, ou seja,

qualquer consulta que pode ser expressa em álgebra relacional pode ser

expressa em cálculo relacional, e vice-versa. No entanto, a escolha entre eles

muitas vezes depende da preferência do usuário ou do otimizador de consulta

do sistema de gerenciamento de banco de dados. Em prática, a maioria dos

sistemas de gerenciamento de banco de dados utiliza uma linguagem

semelhante ao SQL, que é baseada em ambos os modelos.

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

Em SQL, o agrupamento em consultas é realizado usando a cláusula GROUP BY.

Essa cláusula é usada em conjunto com funções de agregação para agrupar

linhas em conjuntos com base nos valores de uma ou mais colunas e, em

seguida, aplicar funções agregadas, como SUM, AVG, COUNT, MAX ou MIN, aos

grupos resultantes.

Colunas no SELECT e GROUP BY: Todas as colunas no SELECT que não estão

envolvidas em funções de agregação devem estar presentes na cláusula GROUP

BY. Isso significa que, se você estiver agrupando por uma coluna, ela deve

aparecer na lista GROUP BY, e se você estiver selecionando uma coluna, ela

deve ser agregada ou estar na lista GROUP BY.

Funções de Agregação: As funções de agregação como SUM, AVG, COUNT, MAX

ou MIN são frequentemente usadas em conjunto com a cláusula GROUP BY. Elas

operam nos grupos de linhas criados pela cláusula GROUP BY.

Imagens:

Diagrama de Banco de Dados (DB Model) mostrando as seguintes tabelas e relacionamentos:

- Pessoa**: idPessoa (INTEGER), nome (VARCHAR(255)), telefone (VARCHAR(255)), estado (VARCHAR(255)), email (VARCHAR(255)).
- PessoaFisica**: cpf (VARCHAR(11)), Pessoa_idPessoa (INTEGER (FK)), Pessoa_idPessoa (INTEGER (FK)).
- PessoaJuridica**: cnpj (VARCHAR(14)), Pessoa_idPessoa (INTEGER (FK)), Pessoa_idPessoa (INTEGER (FK)).
- Produto**: idProduto (INTEGER), nome (VARCHAR(255)), quantidade (INTEGER), precoUnitario (NUMERIC).
- ItemMovimentacao**: idMovimento (INTEGER), Usuario_idUsuario (INTEGER (FK)), Produto_idProduto (INTEGER (FK)), Produto_idProduto (INTEGER (FK)), quantidadeItem (INTEGER), tipo (CHAR(1)).
- Usuario**: idUsuario (INTEGER), nome (VARCHAR(255)), senha (VARCHAR(255)).
- Responsavel**: idMovimento (INTEGER), Usuario_idUsuario (INTEGER (FK)).

Relacionamentos: Pessoa (1) para PessoaFisica (N), Pessoa (1) para PessoaJuridica (N), Produto (1) para ItemMovimentacao (N), Usuario (1) para ItemMovimentacao (N), Usuario (1) para Responsavel (N).

SQL Server Connection Properties:

- Logon: Propriedades da Conexão, Always Encrypted, Parâmetros de Conexão Adicionais
- Servidor: Tipo de servidor: Mecanismo de Banco de Dados, Nome do servidor: VIKYNSQLSERVER, Autenticação: Autenticação do Windows, Nome de usuário: VIKYNNiviky, Senha: Lembrar senha (desmarcado).
- Segurança de Conexão: Criptografia: Obrigatório, Certificado do Servidor Confiável (marcado).
- Nome do host no certificado: (vazio).
- Botões: Conectar, Cancelar, Ajuda, Opções <<

Script SQL:

```
CREATE TABLE PessoaJuridica (
    cnpj VARCHAR(14) NOT NULL,
    Pessoa_idPessoa INTEGER NOT NULL,
    PRIMARY KEY(cnpj),
    FOREIGN KEY(Pessoa_idPessoa)
        REFERENCES Pessoa(idPessoa));
```

Status: 90 % Desconectado.

