Logotipo

O conteúdo gerado por IA pode estar incorreto.

**EAD - Polo Santa Luiza – Vitória – ES**

**DESENVOLVIMENTO FULL STACK**

**RELATÓRIO DE PRÁTICA**

**MUNDO 3 – NÍVEL 2 – MISSÃO PRÁTICA**

**RPG0015 - Vamos manter as informações**

**CARLOS ALTOMARE CATÃO**

**Semestre Letivo: 2025/1 - 3º Período**

**Data: 2025/04**

**PROCEDIMENTO 1 – Criando o Banco de Dados**

**Objetivos da Prática:**

1. Identificar os requisitos de um sistema e transformá-los no modelo adequado.
2. Utilizar ferramentas de modelagem para bases de dados relacionais.
3. Explorar a sintaxe SQL na criação das estruturas do banco (DDL).
4. Explorar a sintaxe SQL na consulta e manipulação de dados (DML)
5. 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.

**Objetivos do Procedimento 1:**

O objetivo do Procedimento 1 é efetuar a modelagem de um Banco de Dados Relacional para um Sistema de Gestão de Operações Comerciais de Compra e Venda de Produtos em transações entre Pessoas Físicas e Jurídicas, levando em consideração os diferentes tipos de relacionamento entre as entidades (como 1:1, 1:N e N:N). A ferramenta de Modelagem será o ***DBDesigner Fork*** e o Banco de Dados adotado será o ***Microsoft SQL Server***.

**Modelagem do Banco de Dados:**

Utilizando a ferramenta *DBDesigner Fork* e baseado nas premissas estabelecidas, foi desenvolvido o Modelo apresentado na figura que se encontra no *link*  .

O arquivo *.xml* referente à imagem acima também se encontra disponibilizado em .

**Criação do Banco de Dados com o Usuário *loja*:**

A sequência a seguir ilustra a criação do Banco de Dados.

Antes da criação, utilizando o usuário *loja.*

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Efetuando a criação do Banco de Dados.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

**Criação das Tabelas, Índices e *Sequence*:**

A seguir pode ser observada a implementação do modelo desenvolvido, com a criação das tabelas, dos índices e da *sequence* utilizada para gerar os códigos de pessoas.

Imagem antes de executar o *script.*

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Imagem após a execução do *script*.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Códigos:**

Os códigos *sql* relativos às operações ilustradas anteriormente se encontram em repositório no GitHub e podem ser acessados pelo link: <https://github.com/CarlosCatao/Mundo_3_Nivel_1_Missao_Pratica/tree/main/Procedimento_1/CadastroPOO>.

A criação do Banco de Dados deve ser, obrigatoriamente, a primeira etapa e deve ser executada, isolada das demais tarefas.

**Resultados:**

Os resultados da execução dos códigos se encontram ilustrados nos parágrafos anteriores.

**Análise e Conclusão:**

No desenvolvimento da prática foram observados inúmeros desafios, desde a instalação dos *Softwares* até a confecção e execução dos *scripts*.

A ferramenta de Modelagem se mostrou muito útil, tanto na construção e montagem do modelo, quanto na agilidade obtida na confecção dos *scripts* de comandos *sql*.

A utilização do **SSMS** (Microsoft SQL Server Management Studio) agiliza as operações junto ao Banco de Dados de forma amigável e permite depurar erros, sejam de codificações ou de permissões, o que impacta muito os trabalhos.

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

As cardinalidades se destinam a expressar a quantidade de entidades que podem se associar a outra em um relacionamento entre tabelas. Em um banco de dados relacional, elas são denominadas como chaves estrangeiras.

* A cardinalidade 1:1 é implementada quando uma entidade pode se associar a outra por meio de uma única chave primária em ambas as tabelas. No modelo desenvolvido pode ser observado nos relacionamentos entre *Pessoa x Pessoa\_Fisica*, *Pessoa x Pessoa\_Juridica* e *Pessoa x Usuario*.
* A cardinalidade 1:N ocorre quando uma *linha* em uma tabela pode se associar a várias linhas de uma outra tabela. No modelo implementado pode ser observado nos relacionamentos *Produto x Operacao* e *Pessoa x Operacao*.
* Um relacionamento N:N é modelado quando uma linha em uma tabela pode se associar a várias linhas de uma outra tabela e vice-versa. No modelo em questão, se considerarmos que uma Pessoa pode comprar vários Produtos e um Produto pode ser comprado por várias Pessoas, a tabela *Pessoa\_Produto* representa esta associação e constitui um exemplo de relacionamento N:N.

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

Nos bancos de dados relacionais, a *herança* é geralmente representada por um dos seguintes modelos:

1. **Modelo de Tabela Única** (Single Table Inheritance): Toda a hierarquia de herança é armazenada em uma única tabela, com um campo que identifica o tipo de entidade. Este modelo simplifica a estrutura, mas casos as classes herdeiras possuam muitos atributos diferentes poderá resultar em muitos campos ***nulos***.
2. **Modelo de Tabela por Classe** (Class Table Inheritance): Cada classe (tanto a superclasse quanto as subclasses) é representada por uma tabela separada. A tabela da subclasse contém apenas os campos **adicionais** da subclasse, e a chave primária é a chave estrangeira para a tabela da superclasse. Este modelo foi adotado na presente Modelagem (*Pessoa x Pessoa\_Fisica x Pessoa\_Juridica*).
3. **Modelo de Tabela por Conjunto** (Concrete Table Inheritance): Cada classe é representada por uma tabela própria e a tabela da subclasse contém todos os campos, inclusive os campos da superclasse. Neste caso pode ocorrer uma ***redundância de dados***.

Conforme exposto o mais recomendável é aplicar o **Modelo de Tabela por Classe** de acordo com a Modelagem aplicada nesta prática.

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

O **SQL Server Management Studio (SSMS)** oferece diversas funcionalidades para a melhoria da produtividade:

1. **IntelliSense**:

Oferece sugestões automáticas de código enquanto você escreve, o que ajuda a reduzir erros e acelera o desenvolvimento de consultas SQL.

1. **Gerenciamento de Banco de Dados Visual**:

Permite que você gerencie de forma intuitiva objetos do banco de dados (tabelas, índices, *views*, etc.) através da sua interface gráfica.

1. **Geração de Scripts**:

O **SSMS** permite gerar scripts SQL automaticamente para a criação ou modificação de objetos de banco de dados, o que acelera a implementação de mudanças.

1. **Execução de Consultas em Lote**:

Suporta a execução de várias consultas simultaneamente, o que aumenta a eficiência quando se executa scripts longos ou na realização de manutenção em larga escala.

1. **Planos de Execução**:

O SSMS permite analisar planos de execução de consultas, o que ajuda a identificar gargalos de desempenho e a otimizar o banco de dados.