**SQL Server Para análise de dados – Diário de bordo de aplicações**

Inserindo banco de dados do curso

- Foi feito uma restauração de um arquivo de backup para efeito de abertura de arquivos que vão ser usados par ao curso. Nesse momento apliquei os seguintes passos:

1 – Abri o SQL Server MSSMS e dentro da minha instância (LEO\_PROJECT):

* 1. – Propriedades;
     1. – Configurações de Banco de Dados;
        1. - Backup (Copiei caminho da pasta)

1. – Copiei o arquivo de backup para a pasta de backup do SQL Server;
2. – Lá na parte esquerda em ‘Pesquisador de Objetos’, com o botão direito em ‘Banco de Dados’, há a opção ‘Restaurar Bando de Dados’
   1. – Selecionar o arquivo recém incluído e clicar em restaurar.

Feito isso, o sistema já carregou o arquivo de BD ‘AdventureWorksDW2019’ contendo 30 tabelas que irei usar durante o treinamento.

Primeiros comandos no SQL Server

USE [Nome do BD] Toda vez que usamos esse comando é selecionado o BD

GO

Uso do comando USE para selecionar o BD correspondente para fazer a consulta.


Interface gráfica do usuário, Aplicativo, Tabela

O conteúdo gerado por IA pode estar incorreto. SELECT \*

FROM dbo.DimProduct;

Aqui é uma consulta simples usando o comando SELECT para selecionar uma instância;

(\*) serve para selecionar todas as colunas;

FROM qual tabela vamos usar;

dbo.DimProduct - aqui é nome da tabela.

Uma vez colocado o BD no SQL Server, se estiver logado com o Azure Data Studio (ADS) vai aparecer lá também espelhado.  
No Docker é preciso copiar o arquivo BD para dentro do container do Docker.

###############################################################

# 🚢 Passo a Passo - Restaurar Banco AdventureWorks no Docker

# Autor: Leonardo Gildo

# Objetivo: Migrar BD do SQL Server local -> Docker (SQL Server)

###############################################################

# ==============================================

# 1. Criar BACKUP no SQL Server (instância local)

# ==============================================

-- Execute no SSMS (Windows local)

BACKUP DATABASE [AdventureWorksDW2019]

TO DISK = 'C:\Backup\AdventureWorksDW2019.bak'

WITH INIT, FORMAT;

# ==============================================

# 2. Copiar o .bak para o container Docker

# ==============================================

# No PowerShell:

# (ajuste o nome do container caso não seja sql2022)

# Criar a pasta de backup dentro do container (se não existir)

docker exec -it sql2022 mkdir -p /var/opt/mssql/backup

# Copiar o .bak do Windows para dentro do container

docker cp "C:\Backup\AdventureWorksDW2019.bak" sql2022:/var/opt/mssql/backup/

# Confirmar se o arquivo chegou ao container

docker exec -it sql2022 ls -lh /var/opt/mssql/backup/

# ==============================================

# 3. Descobrir nomes lógicos do .bak

# ==============================================

-- Execute no SSMS/ADS conectado ao Docker (localhost,1433)

RESTORE FILELISTONLY

FROM DISK = '/var/opt/mssql/backup/AdventureWorksDW2019.bak';

-- Isso vai mostrar algo como:

-- LogicalName: AdventureWorksDW2017

-- LogicalName: AdventureWorksDW2017\_log

# ==============================================

# 4. Restaurar o banco dentro do Docker

# ==============================================

-- Use os nomes lógicos descobertos no passo anterior

-- Aqui forçamos a restaurar como AdventureWorksDW2019

RESTORE DATABASE [AdventureWorksDW2019]

FROM DISK = '/var/opt/mssql/backup/AdventureWorksDW2019.bak'

WITH MOVE 'AdventureWorksDW2017' TO '/var/opt/mssql/data/AdventureWorksDW2019.mdf',

MOVE 'AdventureWorksDW2017\_log' TO '/var/opt/mssql/data/AdventureWorksDW2019\_log.ldf',

REPLACE;

# ==============================================

# 5. Validar

# ==============================================

-- Verificar no SSMS/ADS (Docker) se o banco aparece em Databases

SELECT name, create\_date

FROM sys.databases

WHERE name = 'AdventureWorksDW2019';

Passo a passo para deploy no GitHub 👍

**✅ 1. Primeiro Deploy para um Repositório Novo**

1. Acesse a pasta do projeto:
2. cd "C:\caminho\do\projeto"
3. Inicialize o Git no diretório:
4. git init
5. Adicione todos os arquivos:
6. git add .
7. Faça o primeiro commit:
8. git commit -m "Primeiro commit - Estrutura inicial"
9. Conecte ao repositório remoto:
10. git remote add origin https://github.com/usuario/repositorio.git
11. Defina o branch principal como **main**:
12. git branch -M main
13. Envie os arquivos para o GitHub:
14. git push -u origin main

**✅ 2. Atualizar um Repositório Já Existente**

1. Acesse a pasta do projeto:
2. cd "C:\caminho\do\projeto"
3. Verifique o status dos arquivos:
4. git status
5. Adicione os arquivos alterados:
6. git add .
7. Faça o commit das mudanças:
8. git commit -m "Mensagem descritiva da alteração"
9. Envie as alterações para o GitHub:
10. git push

ALGUMAS DICAS PARA AGILIZAR NA UTILIZAÇÃO DO SQL

- Uma boa prática é escrever o ‘FROM’ antes de digitar as colunas para que se for trabalhar especificadamente as colunas no código SQL, a prática fica melhor.

- Se o nome da tabela estiver na tela, podemos selecionar o nome da tabela e apertar ALT+F1. Com isso, vai aparecer listadas todas as coluna da referida tabela com tipos (domínios) e demais características.

- Sintaxe de comando para mostrar as colunas da tabela e outras características da tabela.

EXEC sp\_columns [nome\_da\_tabela]

- Com o botão de propriedades na tela em branco do SQL abre-se um menu de contexto para escolher algumas opções de ferramentas. Uma delas é ‘PROJETAR CONSULTA NO EDITOR’. Aqui é possível escolher as colunas clicando na coluna que desejar para a consulta.