**ABORDAGEM**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Neste repo vamos abordar a Arquitetura para um SQL Server baseado nos dados da MovieLens.

Criar modelo de dados que comporte:

Filmes (Movies)

Gêneros (Genres)

Locações (Rent)

Usuários (Users)

Cópias para locação (Copy)

Avaliações (Ratings)

Você deverá implementar as seguintes funcionalidades em SQL

Alugar um filme

Devolver um filme

Verificar filmes com retorno em atraso

Verificar filmes que foram entregues com atraso

Dado um filme, verificar quais cópias físicas estão disponíveis para locação

Criar cópia de um filme no banco

**TABELAS**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A normalização das tabelas se dá da seguinte forma:

A tabela filmes vai receber os dados do CSV movies.

A tabela gênero vai precisar receber os gêneros dos filmes do CSV movies.

Vamos precisar de uma tabela para linkar os gêneros com os respectivos filmes por questões de normalização.

A tabela Ratings vai receber os dados do CSV ratings.

A tabela Copy vai receber as cópias dos filmes basedo no CSV tags

A tabela Users vai receber os dados dos usuários baseados do CSV users.

Vamos precisar de uma tabela com os respectivos dados dos filmes alugados, o dia em que foi alugado, sua expectativa de devolução e a real devolução, por questões também de normalização.

Diagrama das tabelas:

Diagrama

Descrição gerada automaticamente

**AZURE DATA FACTORY**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

O objetivo deste caso de uso é utilizar as ferramentas da Azure para resolução e desenvolvimento, criando um SQL Server, fazer o ETL e Sink dos dados através do Azure Data Factory.

**Provisionar:**

Resource Group.

Azure SQL Server

Azure Data Factory

Storage Account (Formato DataLake)

Azure Key Vault (Opcional para guardar as Chaves)

No ADF devemos configura-lo com os linked services que formos usar, por exemplo para um SQL Server devemos configurar um Linked Service que vai integrar o Data Factory com o SQL Server, sendo assim podemos tanto ler os arquivos que estão no servidor como também persistir os dados através das pipelines.

ETL/Sink do SQL Server:

É muito importante avaliar os dados para saber como podemos importar eles para no SQL Server de maneira que a normalização faça sentido.

Observamos que no movies.csv os dados estão da seguinte forma:

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Sendo assim vamos usar um pipeline de cópia utilizando os dados do movieid e title respectivamente para a tabela Movies em nosso banco, assim como da tabela Ratings vamos pegar somente as colunas necessárias e do Users também (analisar dados CSV).

Com nosso banco criado e nosso SQL Server devemos copiar os dados com um pipeline:

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Este Pipeline vai copiar os dados para as respectivas tabelas.

Movies, Ratings e Users.

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Utilizando o DataFlow:

Como observado no nosso csv os gêneros estão todos no csv de movies e não temos eles de forma distinta, além deles estarem concatenados com uma string “|”.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Sendo assim vamos usar o Data Flow para dar um split, baseado nessa string “|” e depois agrupar e contar quantos filmes de cada gênero temos, assim teremos eles de forma distinta além de uma contagem que pode ser útil no futuro.

Filtrando Generos Distintos:



Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Uma vez com os gêneros distintos filtrados vamos criar um pipeline para debugar o Data Flow e inserir os dados no banco. A partir desta tabela vamos ter o index de cada gênero, sendo assim teremos como passar os dados para a tabela de Genres\_Link e assim ter uma normalização adequada.

Debugando DataFlow:

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Genres\_Linked ETL.

Texto

Descrição gerada automaticamente

Linkando Generos:

Diagrama

Descrição gerada automaticamente

Vamos utilizar o Split da mesma maneira que no passo acima e também o Flatern, porém com base nos dados vindos de nossa tabela com os respectivos ids dos gêneros distintos vamos fazer um join para que os dados sejam inseridos no nosso banco.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

SELECT MOVIE\_ID, TITLE, GENRES, GENRE\_ID

FROM Genres\_Link AS G

JOIN Movies AS M

ON M.DATASET\_ID = G.MOVIE\_ID

JOIN Genres AS GE

ON GE.ID = G.GENRE\_ID

ORDER BY G.MOVIE\_ID;

Com a query podemos ver que os dados estão linkados corretamente entre gêneros e filmes.

Interface gráfica do usuário, Aplicativo, Tabela

Descrição gerada automaticamente

Transformando CSV de tags em Copy:

Em termos de aprendizado e também para termos os dados em nossa tabela vamos fazer uma transformação no tags.csv.

Texto

Descrição gerada automaticamente

Nós vamos utilizar o movieId como a cópia por exemplo o movieId 60756 tem três dados na nossa imagem ou seja serão três cópias respectivamente deste filme, e a coluna timestamp será quando o cópia foi criada.

Uma imagem contendo Aplicativo

Descrição gerada automaticamente

toTimestamp(toInteger(toString(byName('timestamp')))\*1000l,'YYYY-MM-DD hh:mm:ss')

Cast coluna timestamp para data.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Resumidamente o andamento do Data Flow para filtrar e dar sink através das pipelines são estes.

Agora vamos demonstrar como executar as Store Procedures através dos pipelines.

Lembrando que as Procedures já tem que estar criadas no nosso SQL Server.

Todos os Scripts de criação das Tabelas, Querys, Procedures e Views Estão no repositório Git.

Interface gráfica do usuário

Descrição gerada automaticamente

Parâmetros para serem passados para as Procedures:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente

Assim chegamos ao fim, podemos com o Dataflow fazer uma carga de dados e rodar eles como uma "esteira" de ingestão de dados, supondo que esses dados cheguem de um link de atualização por exemplo poderiamos criar triggers para sempre fazer a ingestão destes dados atualizados!!!