

# MVP de Engenharia de Dados

Roberto Harkovsky da Cunha

IMDb Database

## 1 Definição do Problema

O IMDb (um acrônimo para Internet Movie Database) é um banco de dados on-line de informações relacionadas a filmes, séries de televisão, podcasts, vídeos caseiros, videogames e streaming de conteúdo on-line - incluindo elenco, equipe de produção e biografias pessoais, resumos de enredos, curiosidades, classificações e análises críticas e de fãs. Como complemento aos dados, o IMDb oferece uma escala de classificação que permite aos usuários votar e avaliar os filmes em uma escala de um a dez.

Neste escopo, objetivo deste projeto é o de realizar uma análise dos títulos publicados (filmes, seriados de TV) e responder as seguintes questões:

- Qual a popularidade dos filmes de James Bond
- Qual a pontuação dos filmes de James Bond
- Quais são os gêneros de filmes mais populares
- Quais os gêneros com as melhores pontuações
- Quais os 10 filmes com maior popularidade de Steven Spielberg
- Quais os 10 filmes com maior pontuação de Steven Spielberg
- Quais os 10 diretores de filmes com maiores médias de pontuação com mais de 5 filmes realizados?
- Quais os são 10 diretores de filmes mais populares?
- Qual é o tempo de execução típico para filmes de cada gênero?
- Qual a pontuação média por gênero de filme, entre os anos de 2020 e 2022?

## 2 Visão Geral do Projeto

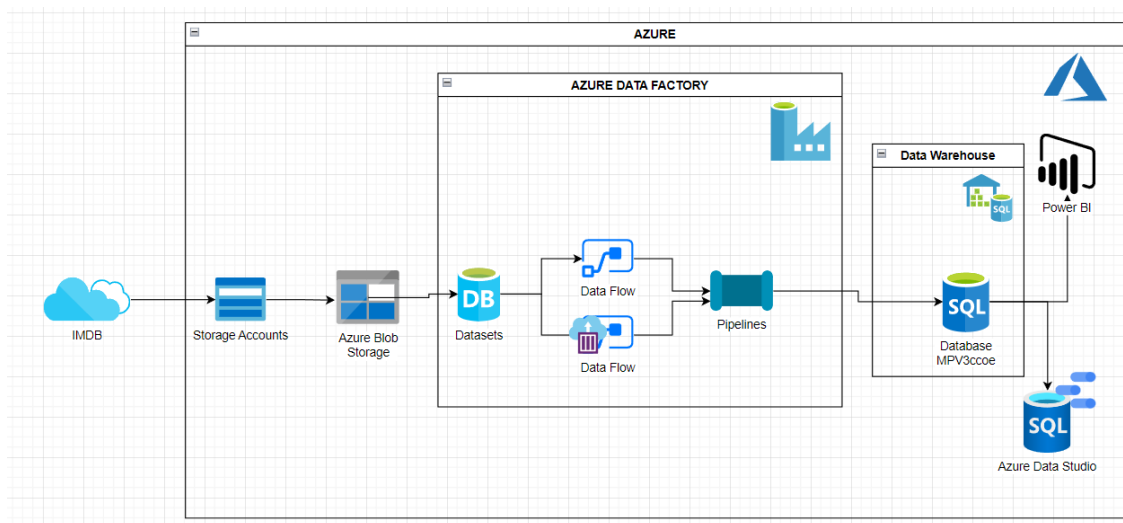
Utilizaremos neste projeto a nuvem Azure da Microsoft, e seus serviços. O desenvolvimento do projeto foi composto pelas seguintes etapas:

- Fonte de dados (Data Sourcing)
- Modelagem de dados

- Ingestão de dados
  - Extração
  - Transformação de dados
  - Carga de dados
- Resultados e Visualização de dados

A visão geral das etapas está apresentada na figura a seguir.

*Figura 1 - Visão geral do Processo de Ingestão dos dados numa base SQL*



Na etapa de fonte de dados foram utilizados arquivos públicos do portal do IMDB. Utilizamos nesta etapa os serviços de repositórios de dados do Azure para armazenar estes dados originais.

Para a etapa de ingestão de dados, foi utilizado o serviço da plataforma Azure Data Factory para realização das etapas de ETL (extração, transformação e carga dos dados), que foram ingeridas num data Warehouse simulado com um base SQL do Azure.

Finalmente, na etapa de resultados e Visualização foi feita utilizando o Azure Data Studio para geração das consultas e o PowerBI da Microsoft para geração dos respectivos gráficos.

Todas as etapas do projeto serão detalhadas nos itens a seguir.

### 3 Fonte de dados (Data Sourcing)

Para este projeto foi utilizado os dados oriundos do IMDb, que consiste em 7 arquivos compactados, com valores separados por tabulação (\*.tsv), que estão disponíveis para download em <https://datasets.imdbws.com/>.

Os arquivos selecionados e baixados são os seguintes:

- name.basics.tsv.gz
- title.akas.tsv.gz
- title.basics.tsv.gz
- title.crew.tsv.gz
- title.episode.tsv.gz
- title.principals.tsv.gz
- title.ratings.tsv.gz

Algumas informações adicionais sobre os dados disponíveis no IMDb:

- Os dados são atualizados diariamente, embora os dados utilizados neste projeto tenham sido obtidos em 16/09/2023.
- Cada um desses arquivos compactados com valores separados por tabulação (TSV) formatados no conjunto de caracteres UTF-8.
- A primeira linha de cada arquivo contém cabeçalhos que descrevem o que há em cada coluna. Um “\N” é usado para indicar que um campo específico está faltando ou tem um valor NULL para esse título ou nome.

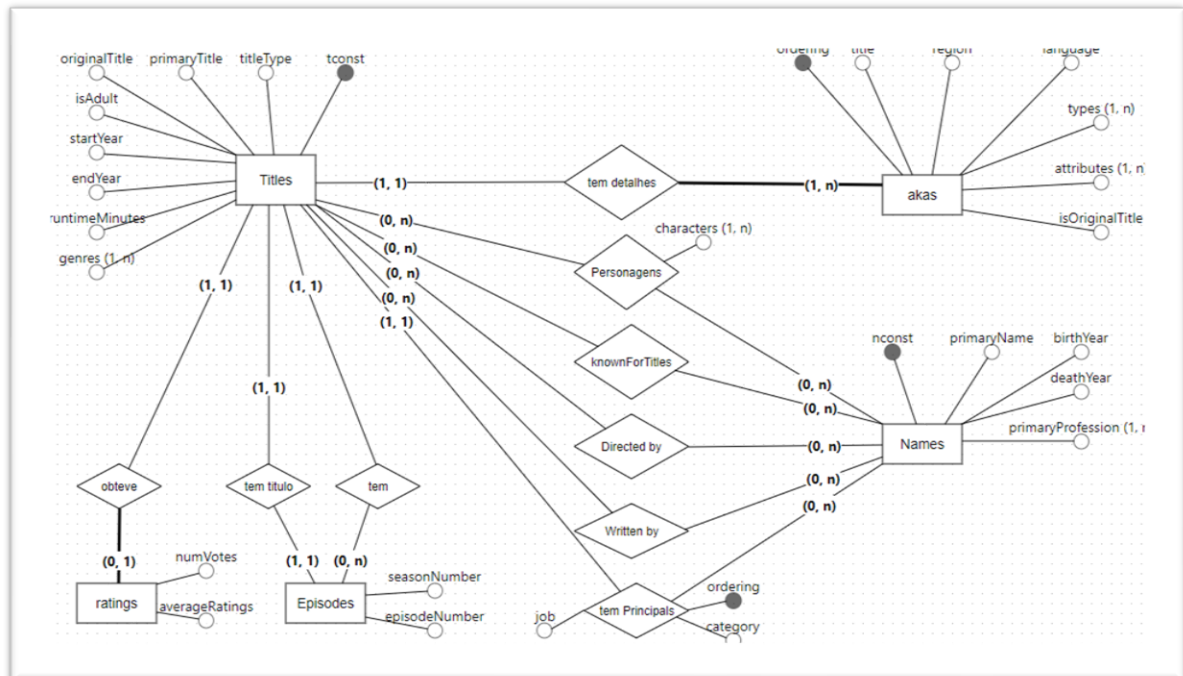
### 4 Modelagem de dados

O objetivo principal do projeto é responder perguntas ligados aos fatos Pontuação (“IMDBRating”), Popularidade (“NumVotes”) e tempo de execução (“Runtime”), segundo as dimensões tempo (ano), diretor, escritor, gênero do filme, linguagem, personagens e episódios, bem como outras questões relacionadas.

## 4.1 Modelo conceitual

O modelo conceitual para este projeto está apresentado a seguir:

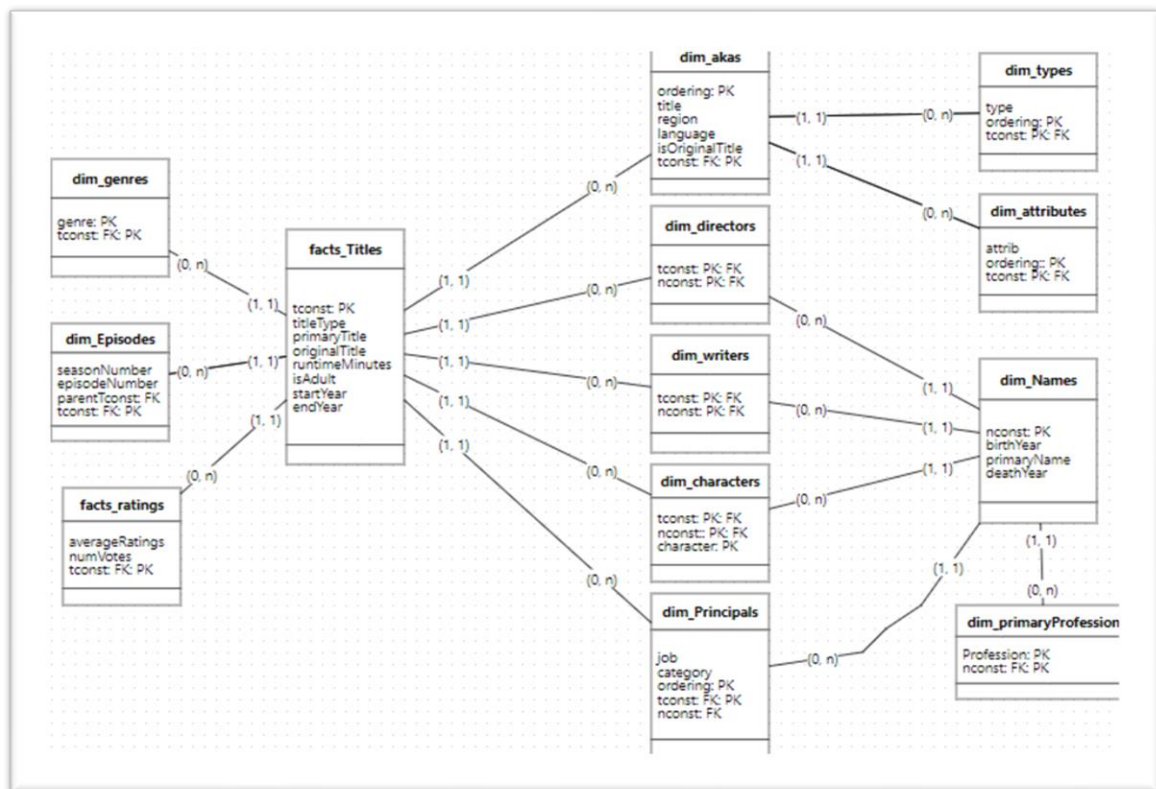
Figura 2 - modelo conceitual do projeto



## 4.2 Modelo lógico (relacional)

Já o modelo lógico foi idealizado como um modelo de dados em Esquema Snowflake, como se fosse fazer parte de um Data Warehouse esta apresentado na figura a seguir

Figura 3 - modelo lógico snowflake do projeto



Este modelo lógico será utilizado como esquema de saída da transformação dos dados.

### 4.3 Catálogo de Dados

A descrição detalhada dos dados de entrada, oriundo do portal do IMDb e dos dados produzidos e ingeridos no SQL (como um DW) seguem nos itens.

#### 4.3.1 Fonte de Dados (IMDb)

Os arquivos fonte estão no formato tsv e o detalhamento do conteúdo de cada foi obtido diretamente do site do IMDb <https://developer.imdb.com/non-commercial-datasets/>. Importante notar que a sequência “\N” representa valores nulos.

Os itens a seguir apresentam a descrição dos campos disponibilizados.

##### 4.3.1.1 *name.basics.tsv.gz*

Contém as seguintes informações para nomes da equipe/atores:

Coluna	Descrição
<b>nconst (string)</b>	Identificador alfanumérico exclusivo do nome/pessoa.
<b>PrimaryName (string)</b>	Nome pelo qual a pessoa é creditada com mais frequência.
<b>birthYear</b>	Valores inteiros no formato AAAA.
<b>deathYear</b>	Valores inteiros no formato no formato AAAA, se aplicável, caso contrário, “\N”.
<b>primaryProfession (matriz de strings)</b>	As 3 principais profissões da pessoa.
<b>knownForTitles (matriz de tconsts)</b>	Títulos pelos quais a pessoa é conhecida.

#### 4.3.1.2 *title.basics.tsv.gz*

Contém as seguintes informações para filmes:

Coluna	Descrição
<b>tconst (string)</b>	Identificador alfanumérico exclusivo do título.
<b>titleType (string)</b>	O tipo/formato do título (por exemplo, filme, curta, série de TV, episódio de TV, vídeo).
<b>primaryTitle (string)</b>	O título mais popular; o título usado pelos cineastas em materiais promocionais no momento do lançamento.
<b>originalTitle (string)</b>	Título original, no idioma original.
<b>isAdult (booleano)</b>	0: título não adulto; 1: título adulto.
<b>startYear (YYYY)</b>	Representa o ano de lançamento de um título. No caso de séries de tv, é o ano de início da série.
<b>endYear (YYYY)</b>	Representa o ano final da série de TV. "N" para todos os outros tipos de títulos.
<b>runtimeMinutes</b>	Tempo de execução principal do título, em minutos.
<b>genres (array de strings)</b>	Inclui até três gêneros associados ao título.

#### 4.3.1.3 *title.akas.tsv.gz*

Contém as seguintes informações extras para filmes:

Coluna	Descrição
<b>titleId (string)</b>	Um tconst que é um identificador alfanumérico exclusivo do título.
<b>ordenação (inteiro)</b>	Um número para identificar exclusivamente as linhas para um determinado titleId.
<b>title (string)</b>	O título localizado.
<b>region (string)</b>	A região para esta versão do título.
<b>language (string)</b>	O idioma do título.

<b>types (array)</b>	Conjunto enumerado de atributos para este título alternativo. Um ou mais dos seguintes: "alternativo", "dvd", "festival", "tv", "vídeo", "trabalho", "original", "imdbDisplay".
<b>attributes (array)</b>	Termos adicionais para descrever este título alternativo, não enumerados.
<b>isOriginalTitle (booleano)</b>	0: título não original; 1: título original.

#### 4.3.1.4 *Title.crew.tsv.gz*

Contém informações do diretor e escritor de todos os títulos da IMDb. Os campos incluem:

Coluna	Descrição
<b>tconst (string)</b>	Identificador alfanumérico exclusivo do título.
<b>directors (array de nconsts)</b>	Diretor(es) do título determinado.
<b>writers (array de nconsts)</b>	Escritor(es) do(s) título(s) fornecido(s).

#### 4.3.1.5 *title.episode.tsv.gz*

Contém as informações do episódio de TV. Os campos incluem:

Coluna	Descrição
<b>tconst (string)</b>	Identificador alfanumérico do episódio.
<b>parentTconst (string)</b>	Identificador alfanumérico da série de TV pai.
<b>seasonNumber (inteiro)</b>	Número da temporada à qual o episódio pertence.
<b>EpisodeNumber (inteiro)</b>	Número do episódio do título da série de TV.



#### 4.3.1.6 *title.principais.tsv.gz*

Contém o elenco/equipe principal dos títulos:

Coluna	Descrição
<b>tconst (string)</b>	Identificador alfanumérico exclusivo do título.
<b>ordering (inteiro)</b>	Um número inteiro positivo para identificar exclusivamente as linhas para um determinado titleid.
<b>nconst (string)</b>	Identificador alfanumérico exclusivo do nome/pessoa.
<b>category (string)</b>	A categoria do trabalho em que a pessoa estava.
<b>job (string)</b>	O cargo específico, se aplicável, caso contrário, “\N”.
<b>characters (string)</b>	O nome do personagem interpretado, se aplicável, caso contrário “\N”

#### 4.3.1.7 *title.ratings.tsv.gz*

Contém a classificação da IMDb e informações de votos para títulos:

Coluna	Descrição
<b>tconst (string)</b>	Identificador alfanumérico exclusivo do título.
<b>AverageRating</b>	Número inteiro positivo com a média ponderada de todas as avaliações individuais dos usuários.
<b>numVotes</b>	Número inteiro positivo com os votos que o título recebeu.

### 4.3.2 Dados Ingeridos (DW SQL)

A descrição dos dados produzidos e ingeridos no SQL Database segue abaixo:

---

#### Dicionário de Dados

NOME	TIPO	TAMANHO	DESCRIÇÃO	CATEGORIAS
<b>INFORMAÇÕES DOS TÍTULOS - facts_title</b>				
tconst	String	10	Identificador alfanumérico exclusivo do título.	formato 'ttXXXXXXX'
titleType	String	30	O tipo/formato do título	[filme, curta, série de TV, episódio de TV, vídeo]
PrimaryTitle	String	até 256	O título mais popular; o título usado pelos cineastas em materiais promocionais no momento do lançamento.	Nome do título
OriginalTitle	String	até 256	Título original, no idioma original.	Nome título na língua original
isAdult	int	Boolean	Especifica se o título é de conteúdo adulto	0: título não adulto; 1: título adulto.
starYear	int	YYYY	representa o ano de lançamento de um título. No caso de séries de TV, é o ano de início da série	valores inteiros positivos de 1874 a 2031 (obs.: valores acima do ano atual podem ser erros ou previsões de lançamento de títulos)
endYear	int	YYYY	Representa o ano final da série de TV. Vaor NULO para todos os outros tipos de títulos.	Valores inteiros positivos de 1906 a 2030 (obs.: valores acima do ano atual podem ser erros ou previsões de lançamento de títulos)
runtimeMinutes	int		Tempo de execução principal do título, em minutos	Valores inteiro de 1 a 54321 (obs.: há valores null indicando dados faltantes e 0 que indica valores errados)
<b>INFORMAÇÕES DE POPULARIDADE E PONTUAÇÃO - facts_ratings</b>				
tconst	String	10	Identificador alfanumérico exclusivo do título.	formato 'ttXXXXXXX'
AverageRating	float		média ponderada de todas as avaliações individuais dos usuários.	valor de 1.0 a 10.0
numVotes	int		número de votos que o título recebeu (Popularidade do título;)	valores de 1 a 2797408
<b>INFORMAÇÕES NOME/IDADE DO ELENCO/ EQUIPE DO TITULO - dim_names</b>				
nconst	String	10	Identificador alfanumérico exclusivo de uma pessoa da equipe	formato 'nmXXXXXXX'
PrimaryName	String		Nome pelo qual a pessoa é creditada com mais frequência.	Nome de pessoa
birthYear	int	YYYY	Ano de nascimento da pessoa	Valores inteiros, positivos, maiores que 1800 e menores que 2023
deathYear	int	YYYY	Ano de falecimento da pessoa	Valores inteiros, positivos, maiores que 1800 e menores que 2023; aplicável se a pessoa já faleceu, caso contrário, NULL.

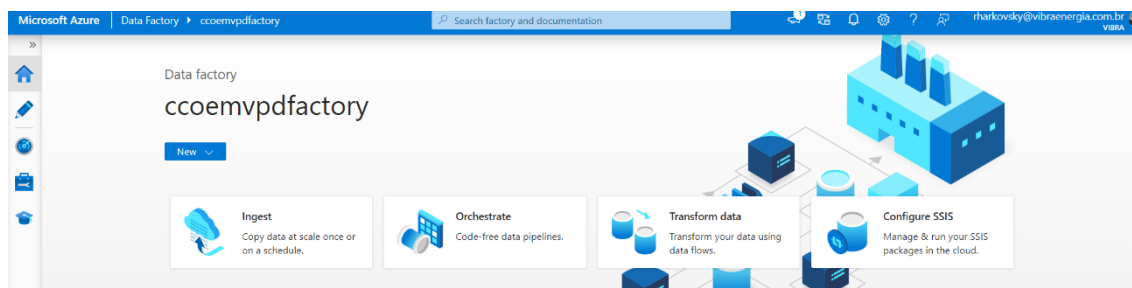
INFORMAÇÕES DOS PERSONAGENS ENCONTRADOS NOS TITULOS - dim_characters				
tconst	String	10	Identificador alfanumérico exclusivo do título.	formato 'ttXXXXXXX'
nconst	String	10	Identificador alfanumérico exclusivo de uma pessoa da equipe	formato 'nmXXXXXXX'
characters	String	100	O nome do personagem interpretado	Nome do personagem, caso ator, caso contrário NULL
INFORMAÇÕES DOS COMPLEMENTARES PARA EPISODIOS DOS TITULOS - dim_episodes				
tconst	String	10	Identificador alfanumérico exclusivo do Episódio.	formato 'ttXXXXXXX'
parentTconst	String	10	Identificador alfanumérico exclusivo do título da série.	formato 'ttXXXXXXX'
seasonNumber	int		Número da temporada à qual o episódio pertence.	numérico inteiro de 1 a 2020
EpisodeNumber	int		Número do episódio do título da série de TV.	numérico inteiro de 0 a 91334
INFORMAÇÕES DOS DIRETORES DOS TITULOS - dim_directors				
tconst	String	10	Identificador alfanumérico exclusivo do título que o diretor fez.	formato 'ttXXXXXXX'
nconst	String	10	Identificador alfanumérico exclusivo do diretor	formato 'nmXXXXXXX'
INFORMAÇÕES DOS ESCRITORES DOS TITULOS - dim_writers				
tconst	String	10	Identificador alfanumérico exclusivo do título que foi escrito	formato 'ttXXXXXXX'
nconst	String	10	Identificador alfanumérico exclusivo do roteirista/escritos do título	formato 'nmXXXXXXX'
INFORMAÇÕES DOS GENEROS DOS TITULOS - dim_genres				
tconst	String	10	Identificador alfanumérico exclusivo do título que foi escrito	formato 'ttXXXXXXX'
genres	String	100	O nome gênero associado ao título.	um valor dentro do seguinte conjunto: [Action, Adult, AdventureAnimation, Biography, Comedy, Crime, Documentary, Drama, Family, Fantasy, Film-Noir, Game-Show, History, Horror, Music, Musical, Mystery, News, Reality-TV, Romance, Sci-Fi, Short, Sport, Talk-Show, Thriller, War, Western]
INFORMAÇÕES DO ELENCO/EQUIPE DOS TITULOS - dim_principals				
tconst	String	10	Identificador alfanumérico exclusivo do Episódio.	formato 'ttXXXXXXX'

ordering	int		Um número para identificar exclusivamente as linhas para um determinado título	valor inteiro positivo de 1 a 10
nconst	String	10	Identificador alfanumérico exclusivo de uma pessoa da equipe	formato 'nmXXXXXXX'
category	String		A categoria do trabalho que a pessoa realizou no título.	Um valor do conjunto [actor, actress, archive_footage, archive_sound, cinematographer, composer, director, editor, producer, production_designer, self, writer]
job	String		O cargo específico no título, se aplicável, caso contrário, NULL	Um nome de um cargo ou o valor Nulo
<b>INFORMAÇÕES DA PROFISSÃO DAS PESSOAS - dim_professions</b>				
nconst	String	10	Identificador alfanumérico exclusivo de uma pessoa	formato 'ttXXXXXXX'
profession	String	30	Profissão da pessoa	um valor do conjunto [actor, actress, animation_department, art_department, art_director, assistant, assistant_director, camera_department, casting_department, casting_director, choreographer, cinematographer, composer, costume_department, costume_designer, director, editor, editorial_department, electrical_department, executive, legal, location_management, make_up_department, manager, miscellaneous, music_artist, music_department, podcaster, producer, production_department, production_designer, production_manager, publicist, script_department, set_decorator, sound_department, soundtrack, special_effects, stunts, talent_agent, transportation_department, visual_effects, writer]

## 5 Ingestão - ETL

Como comentado no tópico “Visão geral”, para o processo de ingestão de dados foi realizado um ETL utilizando o Azure Data Factory (ADF). Para tal fim foi criada uma instância do ADF chamada “ccoempdfactory”, na qual será realizada a orquestração do ETL deste projeto.

Figura 4 - Azure Data factory do projeto



### 5.1 Extração de dados

A primeira etapa foi a criação de um repositório no Azure de onde os arquivos serão ingeridos originalmente. O repositório foi criado por meio do serviço StorageAccount, nomeado “ccoempstorage” conforme a figura:

Figura 5 - Etapa de extração de Dados

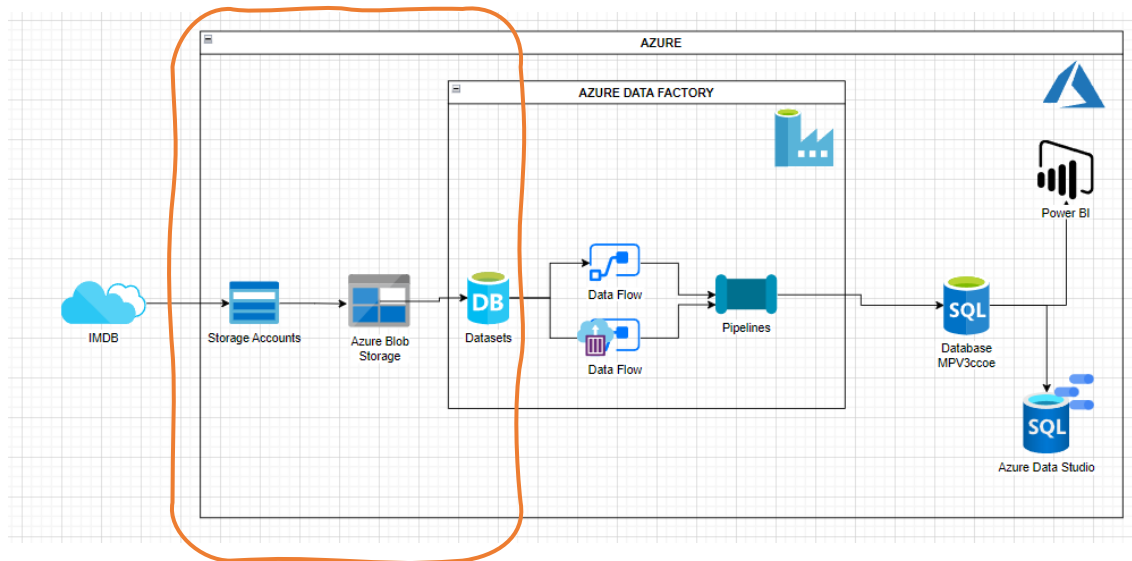
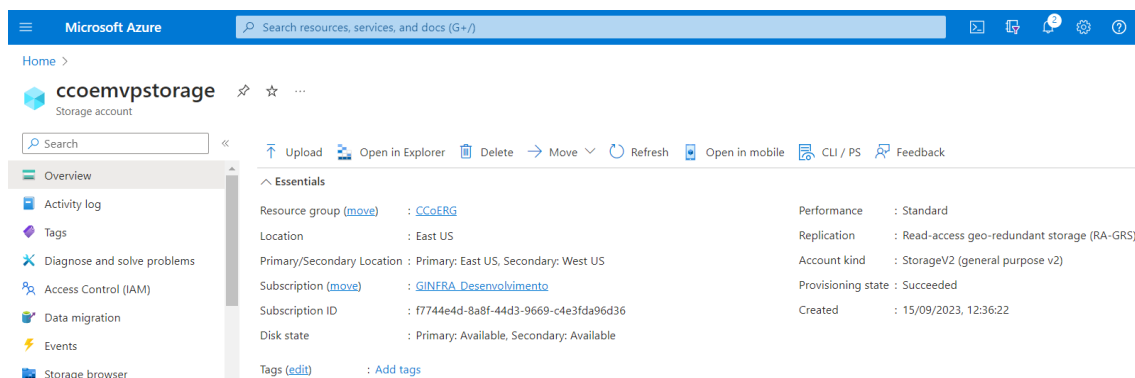


Figura 6 - storage account para repositório dos arquivos fontes

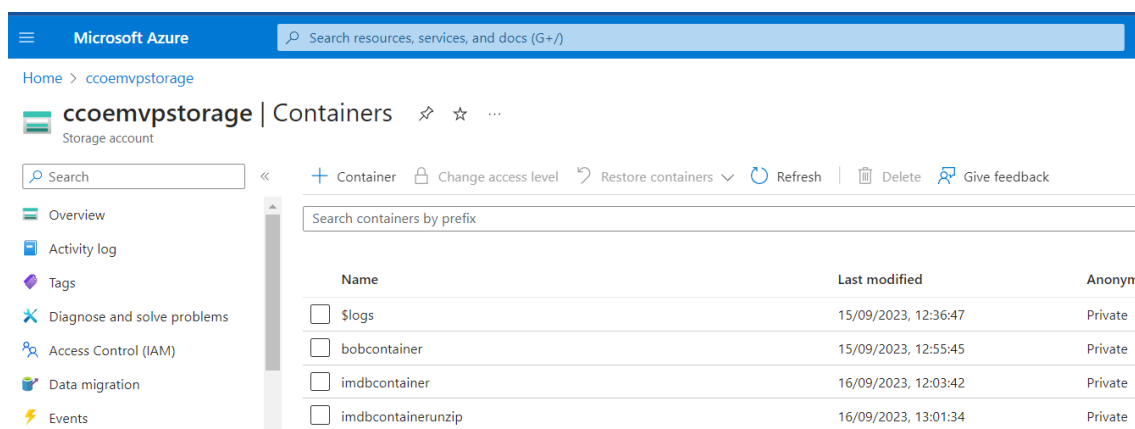


A seguir foram criados 2 containers: “imdbcontainer”, que contém os arquivos IMDB originais compactados, e o container “imdbcontainerunzip” que contém a versão descompactada dos arquivos.

A seguir foram criados 2 containers: “imdbcontainer” e “imdbcontainerunzip”.

Os arquivos originais compactados (extensão .gz) foram descompactados e carregados no storage Account “cchoempstorage” da seguinte forma: a versão compactada foi carregada no container “imdbcontainer”, e a versão descompactada foi carregada no container “imdbcontainerunzip”.

Figura 7 - Container com os arquivos fonte



A figura a seguir evidencia a criação e o conteúdo do container “imdbcontainerunzip”.

Figura 8 - Container com arquivos IMDB descompactados

Home > ccoempvstorage | Containers >

**imdbcontainerunzip** ...

Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots

Overview

Diagnose and solve problems

Access Control (IAM)

Settings

Shared access tokens

Access policy

Properties

Metadata

Authentication method: Access key (Switch to Azure AD User Account)

Location: imdbcontainerunzip

Search blobs by prefix (case-sensitive)

Add filter

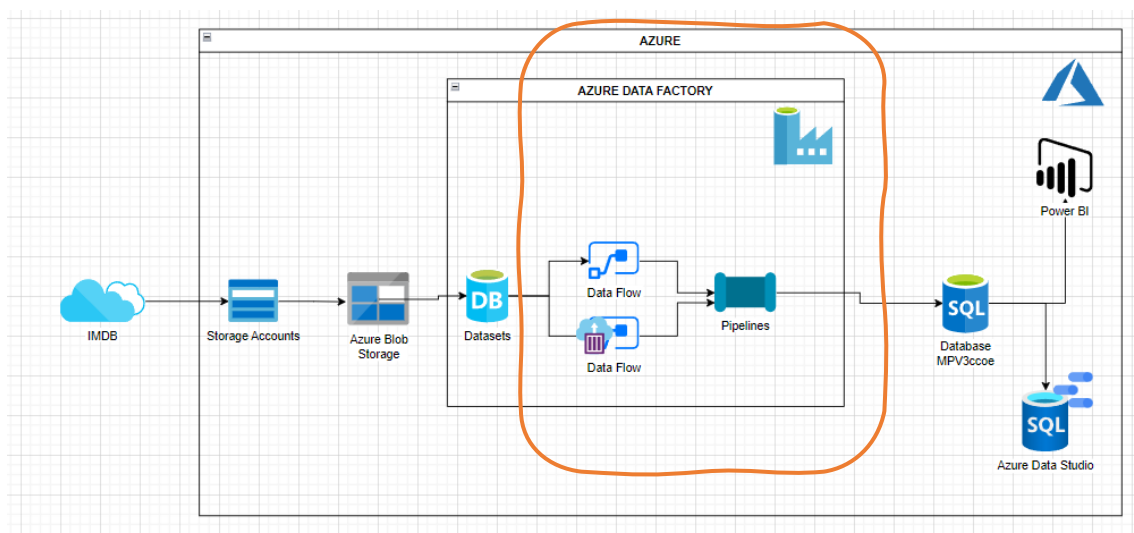
Name	Modified	Access tier	Archive status	Blob type
<input type="checkbox"/> name.basics.tsv	16/09/2023, 13:08:40	Hot (Inferred)		Block blob
<input type="checkbox"/> title.akas.tsv	16/09/2023, 13:10:47	Hot (Inferred)		Block blob
<input type="checkbox"/> title.basics.tsv	16/09/2023, 13:07:48	Hot (Inferred)		Block blob
<input type="checkbox"/> title.crew.tsv	16/09/2023, 13:04:26	Hot (Inferred)		Block blob
<input type="checkbox"/> title.episode.tsv	16/09/2023, 13:03:29	Hot (Inferred)		Block blob
<input type="checkbox"/> title.principals.tsv	16/09/2023, 13:11:53	Hot (Inferred)		Block blob
<input type="checkbox"/> title.ratings.tsv	16/09/2023, 13:03:56	Hot (Inferred)		Block blob

Para este projeto utilizaremos apenas os arquivos descompactados oriundos deste container.

## 5.2 Transformação de dados

A etapa de transformação dos dados está representada na figura. Ele envolve basicamente a criação de fluxos de transformação de dados (Dataflow) que são agrupados e executados em uma estrutura chamada pipeline.

Figura 9 - Etapa de ETL



Para as transformações necessárias, foram criados 2 dataflows no processo de transformação dos dados: um para as tabelas de dimensões

chamado de “dataflow\_dim” e um segundo chamado de “dataflow\_fact” para a tabela de fatos.

Para seleção dos atributos para as novas tabelas foi utilizado a técnica de projeção das colunas por meio do componente "Select".

Já o tratamento dados multivalorados , como mostrado na modelagem, foi o de criar tabelas específicas par cada um deles. Neste caso foram utilizados os componentes “derived column” e "flatten".

Para garantir que não ocorrência de campos nulos, foi utilizado o componente de filtragem de conteúdo "Filter".

O tratamento dos dados para cada fluxo está detalhado a seguir.

### 5.2.1 “dataflow\_fact”

O Dataflow “dataflow\_fact” para geração da tabela de fatos está apresentado abaixo:

Figura 10 - Fluxo Dataflow Facts



Para geração do dataflow de fatos, foram utilizadas como fonte as tabelas titleBasics e titleRatings (1)

Em cada uma delas foram projetadas as seguintes colunas (2)

- titleRatings (tconst, averageRating, numVotes)
- titleBasics (tconst, primarytite, originaltitle, isAdult, startYear, endYear, runtimeMinutes)



Figura 11 - Datasources dataflow facts - Fatos



A saída do fluxo é feita pelo componente “Sink” (3).

Um exemplo de preview de saída da tabela de fatos “fact\_title”, com as configurações adotadas no componente “Select” (projeção de campos) está apresentado na figura a seguir:

Figura 12 - Preview de saída da tabela facts\_title no ADF

dataflow\_facts\_v1

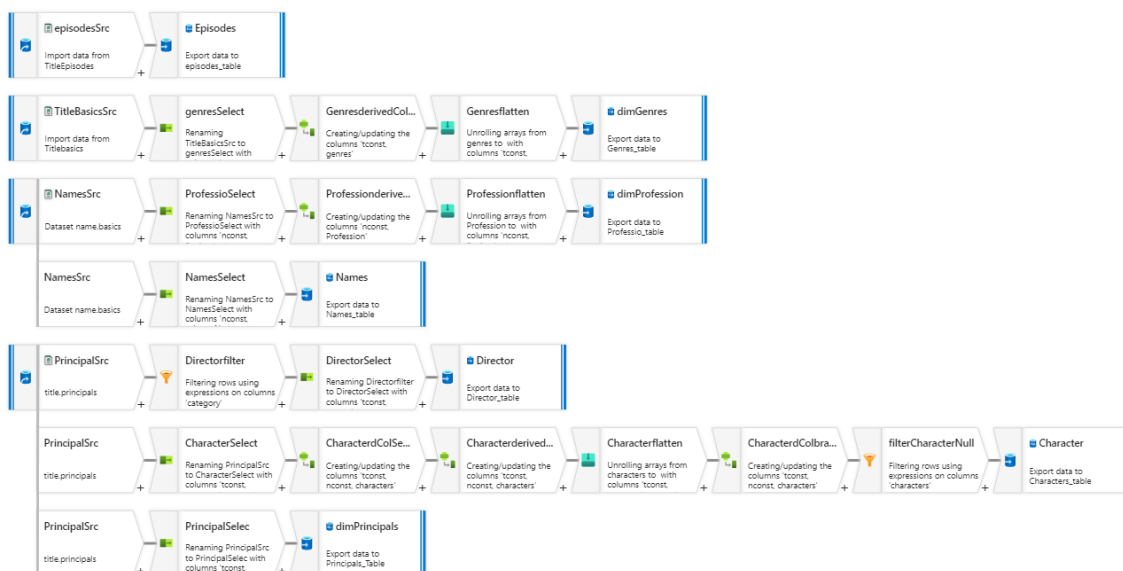
Validate Data flow debug Debug Settings

tconst	titleType	primaryTitle	originalTitle	isAdult	startYear	endYear	runtimeMinutes
tt0000001	short	Carmencita	Carmencita	×	1894	NULL	1
tt0000002	short	Le clown et ses chiens	Le clown et ses chiens	×	1892	NULL	5
tt0000003	short	Pauvre Pierrot	Pauvre Pierrot	×	1892	NULL	4
tt0000004	short	Un bon bock	Un bon bock	×	1892	NULL	12
tt0000005	short	Blacksmith Scene	Blacksmith Scene	×	1893	NULL	1
tt0000006	short	Chinese Opium Den	Chinese Opium Den	×	1894	NULL	1
tt0000007	short	Corbett and Courtne...	Corbett and Courtne...	×	1894	NULL	1
tt0000008	short	Edison Kinetoscopic ...	Edison Kinetoscopic ...	×	1894	NULL	1

### 5.2.2 “dataflow\_dim”

A estrutura do Dataflow “dataflow\_dim” para geração das tabelas de dimensão do projeto está apresentada na figura a seguir.

Figura 13 - Dataflow de dimensões



Para geração do dataflow de dimensões, foram utilizadas como fonte as tabelas titleBasics , titleEpisodes, names, titleprincipals (1).

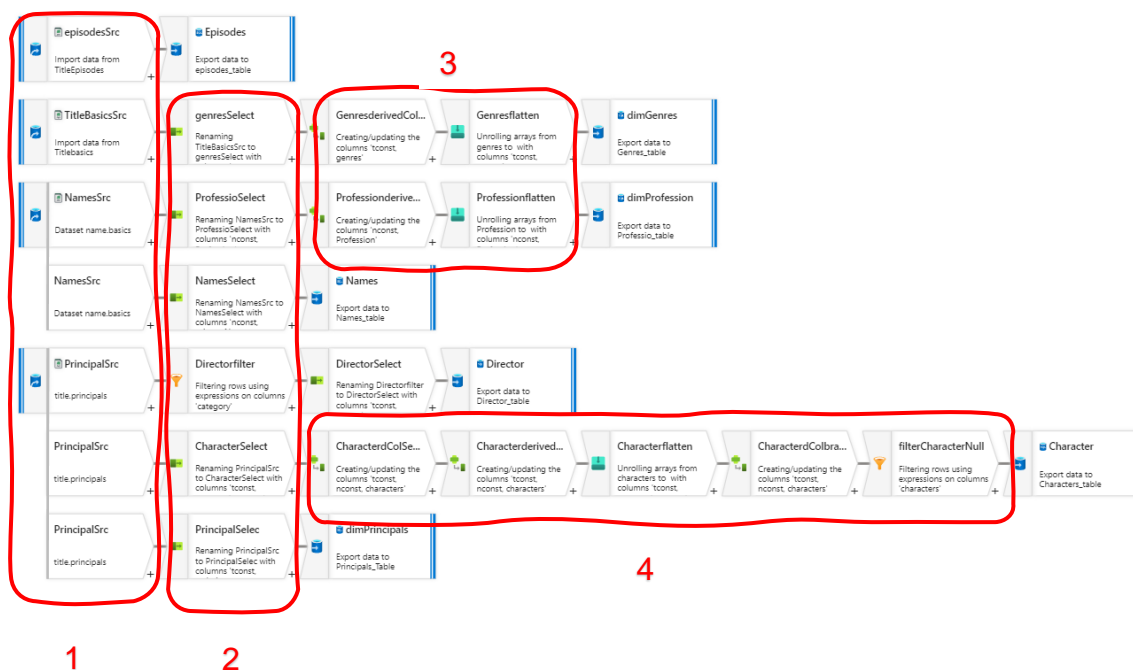
As seguintes transformações foram aplicadas: (2)

- Tabela titleBasics: projetado o campo multivalorado “genre” para criação de uma tabela específica de gêneros de filmes;
- Tabela Names: projetadas as colunas “nconst”, “primaryName”, “birthDate”, “deathDate” para criação de uma tabela de apoio de nomes de pessoal;
- Tabela Names foi ainda projetado o campo multivalorado “profession” para criação de uma tabela específica de profissões na produção dos filmes.
- Da Tabela titlePrincipals foi projetado e transformado o campo multivalorado “characters” para criação de uma tabela específica de personagens de filmes;
- Da Tabela titlePrincipals foi projetado e transformado o campo “Director”, filtrando as linhas com a categoria de trabalho (campo “category”) como

o valor “Director” para criação de uma tabela específica de diretores de filmes;

- Da Tabela titlePrincipals foram projetados os campos “tconst”, “ordering”, “nconst”, “category” e “job” para uso nas consultas;
- A tabela Episodes não sofreu o processo de projeção.
- Para os campos multivalorados “genre” e “profession”, foram feitas ainda diversas transformações para derivar novas tabelas. (3)
- Já para a tabela de personagens (character) foram feitas diversas transformações e limpas as possíveis ocorrências de nulos (null). (4)

Figura 14 - Dataflow\_dim - dimensões

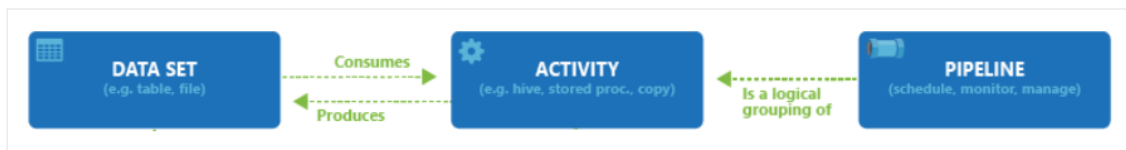


### 5.2.3 Pipelines

Um pipeline é um agrupamento lógico de atividades que juntas executam uma tarefa, que ingerem dados de um dataset e produzem novos dados. As atividades disponíveis são atividade de cópia, atividade de fluxo de dados

Desta forma as atividades de um pipeline definem as ações a serem executadas nos seus dados. No caso deste projeto, foram usadas atividades de fluxo de dados.

Figura 15 – visão geral do funcionamento de Pipelines (fonte:Microsoft)



Neste projeto foram criados 2 pipelines contendo cada um dos fluxos principais dataflow\_facts ou dataflow\_dim. Um exemplo de execução de um deles está apresentado abaixo:

Figura 16 - exemplo de execução - Pipeline facts

**Pipeline run ID:** 05be433d-f4e7-4c86-9c03-3ef1fb44f970 **Pipeline status:** Succeeded

Data flow activity for this debug run will start as soon as the data flow debug session is ready.

All status

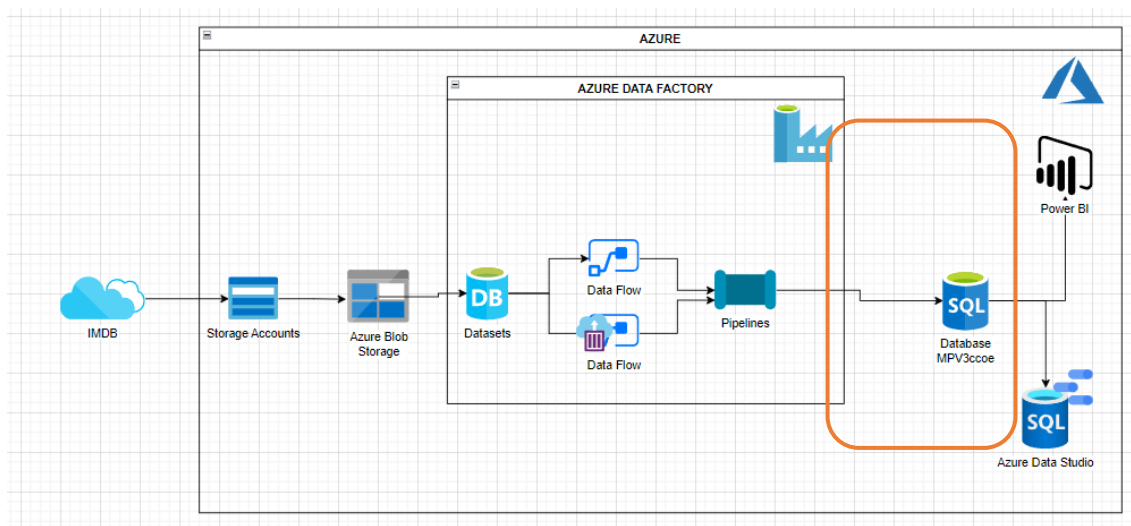
Showing 1 - 1 of 1 items

Activity name	Activity status	Activity type	Run start	Duration	Integration runtime	User property
dataflow_facts	Succeeded	Data flow	9/17/2023, 9:48:29 PM	4m 22s	AutoResolveIntegration	

### 5.3 Carga de dados

A etapa final do processo de ETL é a carga dos dados gerados em um repositório. Neste projeto o repositório é uma base SQL no AZURE.

Figura 17 - Etapa de carga de dados



Contudo o processo de carga exige a existência de um servidor e um banco de dados. Assim, o processo de carga de dados envolveu 3 fases distintas:

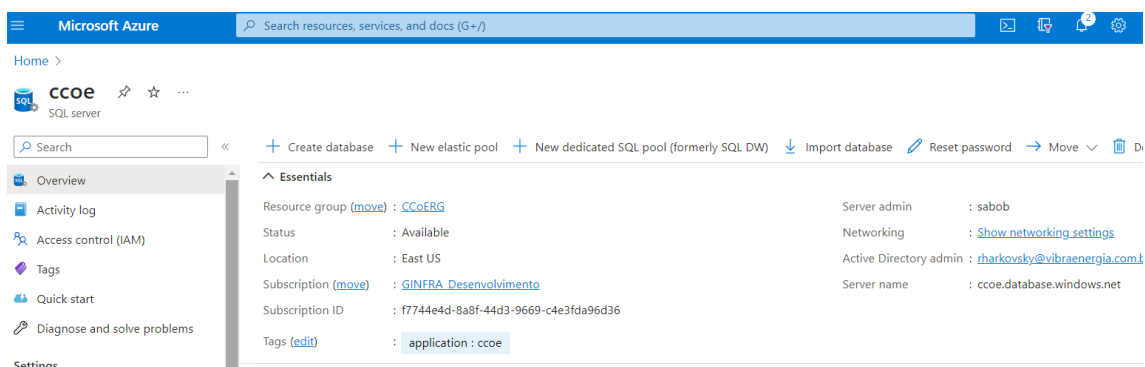
- Criação do database e esquemas no Azure
- Carga das tabelas (saída do ADF)
- Inclusão das restrições de chaves nas tabelas

Estas etapas estão descritas nos itens a seguir.

### 5.3.1 Criação do Database e Esquemas no Azure

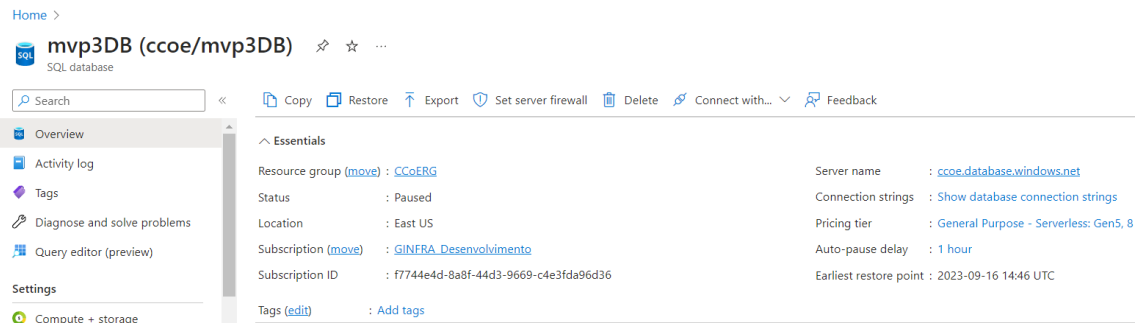
Para receber os dados oriundos do ETL foi criado no Azure um servidor de banco de dados SQL *serverless* chamado de “ccoe”, como evidenciado na figura a seguir.

Figura 18 - servidor de banco de dados “ccoe” no Azure



Em seguida, uma base de dados chamada de “mvp3DB” foi criada neste servidor (vide figura), para onde foram direcionados os dados de saída do modelo.

Figura 19 - Database no servidor "ccoe"



A próxima etapa consistiu em, a partir da modelagem realizada no item 4, proceder a criação propriamente dita das tabelas com as devidas restrições de chave primária e estrangeira. O script da figura foi elaborado e aplicado ao database, resultando na criação das tabelas.

Figura 20 - Script de criação das tabelas

```
CREATE TABLE title_facts
(
  tconst varchar(10) PRIMARY KEY,
  titleType varchar(30),
  primaryTitle varchar(max),
  originalTitle varchar(max),
  isAdult INT,
  startYear INT,
  endYear INT,
  runtimeMinutes INT,
);
CREATE TABLE dim_names
(
  nconst varchar(10),
  primaryName varchar(150),
  birthYear INT,
  deathYear INT,
  CONSTRAINT pk_names PRIMARY KEY (nconst)
);
CREATE TABLE dim_Profession
(
  nconst varchar(10),
  profession varchar(30) ,
  CONSTRAINT pk_profession PRIMARY KEY (nconst, profession)
);
```

```

CREATE TABLE dim_principals
(
    tconst varchar(10),
    ordering INT,
    nconst varchar(10),
    job varchar(max),
    category varchar(60),
    CONSTRAINT pk_princpals PRIMARY KEY (tconst,ordering)
)

CREATE TABLE dim_episodes
(
    tconst varchar(10) PRIMARY KEY,
    parentTconst varchar(10),
    seasonNumber INT,
    episodeNumber INT,
);

CREATE TABLE [dbo].[dim_directors]
(
    tconst varchar(10),
    nconst varchar(10),
    CONSTRAINT pk_Director PRIMARY KEY (tconst, nconst)
);

CREATE TABLE dim_Characters
(
    tconst varchar(10),
    nconst varchar(10),
    characters varchar(30),
    CONSTRAINT pk_Character PRIMARY KEY (tconst, nconst, characters)
);

CREATE TABLE dim_genres
(
    tconst varchar(10),
    genre varchar(30),
    CONSTRAINT pk_genres PRIMARY KEY (tconst, genre)
);

CREATE TABLE facts_ratings
(
    tconst varchar(10) PRIMARY KEY,
    averageRating DECIMAL (5,1),
    numVotes INT
);

CREATE TABLE dim_akas
(
    tconst varchar(10) ,
    ordering INT,
    title varchar(max),
    region varchar(10),
    language varchar(5),
    isOriginalTitle INT,
    CONSTRAINT pk_akas PRIMARY KEY (tconst, ordering)
);

ALTER TABLE dim_episodes ADD FOREIGN KEY(parentTconst) REFERENCES facts_title (parentTconst)
ALTER TABLE dim_episodes ADD FOREIGN KEY(tconst) REFERENCES facts_title (tconst)

ALTER TABLE dim_principals ADD FOREIGN KEY(tconst) REFERENCES facts_title (tconst)
ALTER TABLE dim_principals ADD FOREIGN KEY(nconst) REFERENCES dim_names (nconst)

ALTER TABLE dim_directors FOREIGN KEY(tconst) REFERENCES facts_title (tconst)
ALTER TABLE dim_directors FOREIGN KEY(nconst) REFERENCES dim_names (nconst)

```

```
ALTER TABLE dim_Profession ADD FOREIGN KEY(nconst) REFERENCES facts_title (nconst)
ALTER TABLE dim_genres ADD FOREIGN KEY(genre) REFERENCES facts_title (genre)
ALTER TABLE dim_genres ADD FOREIGN KEY(tconst) REFERENCES facts_title (tconst)

ALTER TABLE dim_Characters ADD FOREIGN KEY(tconst) REFERENCES facts_title (tconst)
ALTER TABLE dim_Characters ADD FOREIGN KEY(nconst:) REFERENCES dim_names (nconst)
ALTER TABLE facts_ratingsADD FOREIGN KEY(tconst) REFERENCES facts_title (tconst)
```

### 5.3.2 Carga das tabelas (saída ADF)

A saída do processo de extração e transformação dos dados foram tabelas SQL sem restrições, que foram armazenados num banco de dados SQL no Azure. Para tal, o componente “sink” do fluxo de transformação é responsável por apontar para o SQL server/database e carregar os dados na respectiva tabela. Nas figuras do item de “transformação” anteriormente apresentados, ele é o último componente como nome dim\*.

Antes de criar um dataset, é preciso criar um serviço para vincular o repositório de armazenamento de dados ao ADF. Assim, o componente SINK implementa este serviço de conexão, ou *linked service*, com a base de dados do servidor ccoe. A configuração deste *linked service* segue abaixo:

*Figura 21 - linked server coma base de dados no servidor ccoe*



## Edit linked service


 Azure SQL Database [Learn more](#) 

Name \*

AzureSqlDbMVP3

Description

Database MVP3

Connect via integration runtime \* 

AutoResolveIntegrationRuntime

Connection string

Azure Key Vault

Account selection method 

☐ From Azure subscription ☒ Enter manually

Fully qualified domain name \*

ccoe.database.windows.net

Database name \*

mvp3DB

Authentication type \*

SQL authentication

User name \*

sabob


Password

Azure Key Vault

Password \*

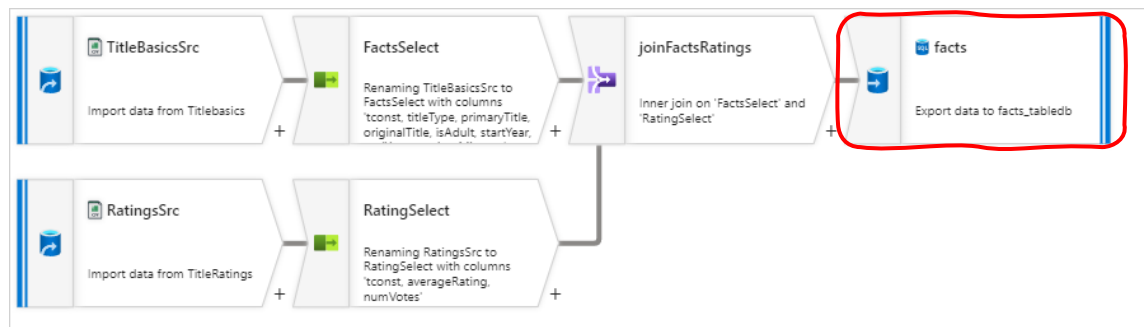
Apply

Cancel

 Test connection

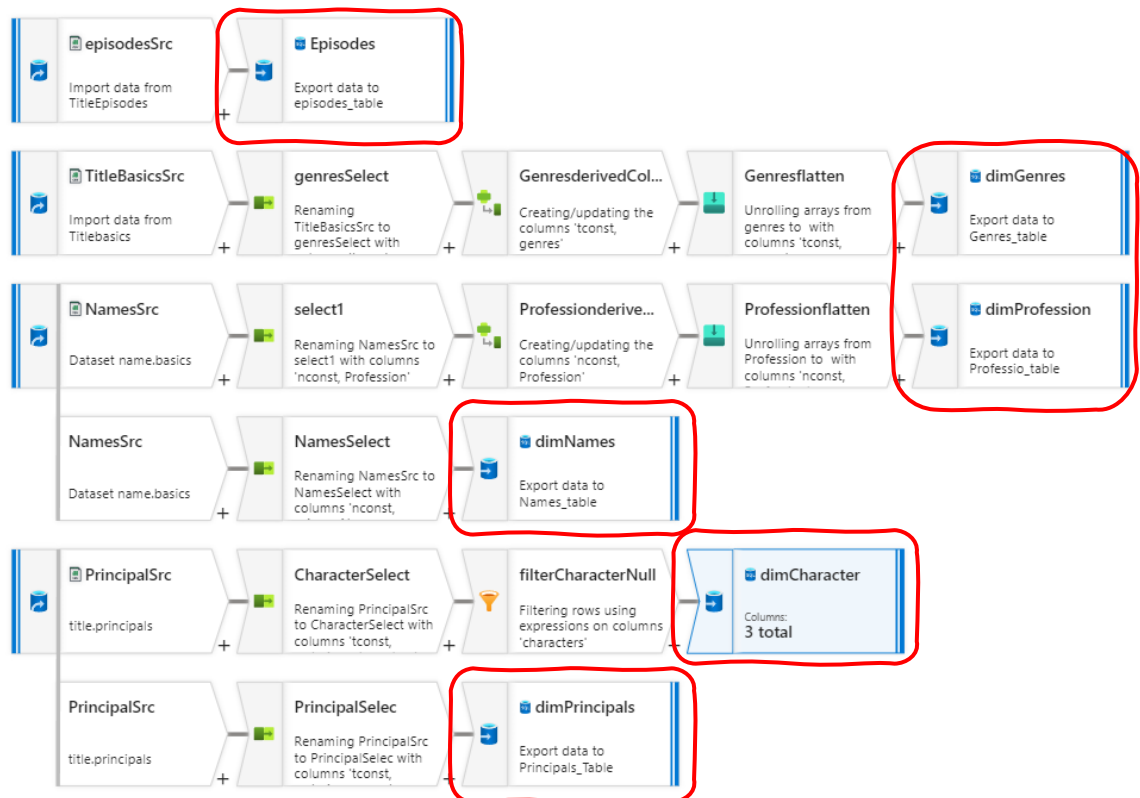
Para o fluxo de criação da tabela de fatos com o componente “Sink” está apresentado na figura, com nome “facts”.

Figura 22 - componente de saída da transformação SINK – tabela de fatos



Já para o fluxo de criação das tabelas de dimensão, temos vários sinks (um para cada tabela gerada) conforme mostrado na figura:

Figura 23 - componente de saída da transformação SINK – tabela de dimensões



### 5.3.3 Inclusão das restrições

A terceira etapa consistiu em carregar os dados das tabelas geradas pelo ADF (sem restrições) nas tabelas SQL criadas com as devidas restrições de chave Primária e estrangeira. Para isto foi utilizado o AZURE DATA STUDIO e comandos “INSERT INTO” tendo como origem as tabelas oriundas do ADF, renomeadas para “\*\_old”. O script utilizado está mostrado na figura a seguir.

Figura 24 - Script INSERT INTO de carga final das tabelas

```
--INSERT INTO
INSERT INTO [dbo].[dim_title_facts]
SELECT * FROM [dbo].[dim_title_facts_old];

INSERT INTO [dbo].[dim_dim_names]
SELECT * FROM [dbo].[dim_dim_names_old];

INSERT INTO [dbo].[dim_Profession]
SELECT * FROM [dbo].[dim_Profession_old];

INSERT INTO [dbo].[dim_principals]
SELECT * FROM [dbo].[dim_principals_old];

INSERT INTO [dbo].[dim_episodes]
SELECT * FROM [dbo].[dim_episodes_old];

INSERT INTO [dbo].[dim_directors]
SELECT * FROM [dbo].[dim_directors_old];

INSERT INTO [dbo].[dim_episodes]
SELECT * FROM [dbo].[dim_episodes_old];

INSERT INTO [dbo].[dim_Characters]
SELECT * FROM [dbo].[dim_Characters_old];

INSERT INTO [dbo].[dim_genres]
SELECT * FROM [dbo].[dim_dim_genres_old];

INSERT INTO [dbo].[title_Ratings]
SELECT * FROM [dbo].[title_Ratings_old];
```

## 6 Análise

### 6.1 Qualidade dos dados

Numa análise de qualidade dos dados obtidos no IMDb, foi notado que há dados faltantes, principalmente nos arquivos fontes **name.basics.tsv.gz** e **title.basics.tsv.gz**. Esses dados faltantes causam problemas quando do processo de ETL dos dados tentamos impor certas restrições de chave estrangeira.

#### 6.1.1 Problemas no conjunto de dados

A tabela `dim_akas` (oriunda de “`title.akas.tsv.gz`”) possui títulos que não existem em `facts_title` (oriundas de “`title.basics.tsv.gz`”). O problema se propaga ao definir a chave estrangeira para as tabelas geradas via ETL a partir de “`title.akas.tsv.gz`”, “`dim_attributes`” e “`dim_types`”. Também foi observado a falta

de dados de linguagem (campos “Language”) e outras informações (valores NULLs)

RunCancelDisconnectChangeDatabase:.mvp3DBEstimated PlanEnable Actual Plan

ParseEnable SQLCMDTo Notebook

1SELECT top (10) \*

2FROM dim\_akas

3WHERE tconst not in (SELECT tconst FROM facts\_title)

	tconst	ordering	title	region	language	isOriginalTitl
1	tt0021453	1	Tapping Toes	US	NULL	NULL
2	tt0023019	1	Hollywood on Parade	US	NULL	NULL
3	tt0024677	1	Tom's in Town	US	NULL	NULL
4	tt0036165	1	Missing Men	US	NULL	NULL
5	tt0038098	1	Son of the Prairie	US	NULL	NULL
6	tt0046142	1	One Came Home	US	NULL	NULL
7	tt0052041	1	Over She Goes	US	NULL	0
8	tt0052206	1	Smoke Jumpers	US	NULL	0
9	tt0063739	1	Une cigarette pour un ingénu	FR	NULL	NULL
10	tt0066616	1	L'échappé fabuleux	FR	NULL	NULL

Esta observação vale também para a tabela gerada “dim\_Episodes”, oriundas do dataset title.episode.tsv.gz.

RunCancelDisconnectChangeDatabase:.mvp3DBEstimated Plan

ParseEnable SQLCMDTo Notebook

1SELECT DISTINCT parentTconst

2FROM dim\_episodes

3WHERE parentTconst NOT IN (SELECT tconst from facts\_title)

	parentTconst
1	tt12146052
2	tt12146082
3	tt12153004
4	tt27502188

A mesma situação de dados ausentes acontece com as tabelas derivadas do campo “category” de dim\_names (origem: “name.basics.tsv.gz”), que deram origem as tabelas “dim\_directors”.

▶ Run

❏ Cancel

🔌 Disconnect

🔄 Change

Database: mvp3DB

✓ Parse

📄 Enable SQLCMD

📄 To Notebook

```
1 SELECT tconst
2 FROM dim_director
3 WHERE tconst NOT IN (SELECT tconst from facts_title)
```

Results

Messages

	tconst	▼
1	tt1124380	
2	tt7495268	
3	tt28994664	
4	tt7110514	
5	tt8186272	
6	tt8186272	
7	tt8186272	
8	tt8034630	
9	tt8034630	
10	tt1661214	
11	tt1490018	
12	tt28995232	

A tabela dim\_principals (origem: “title.principals.tsv.gz”) tem nomes que não aparecem em dim\_names (origem: “name.basics.tsv.gz”), ou seja, há códigos de pessoas em title.principals.tsv.gz que não tem nome e outras informações associadas.

▶ Run
❏ Cancel
🔌 Disconnect
🔄 Change

Database: mvp3DB

✓ Parse
📄 Enable SQLCMD
📖 To Notebook

```

1  SELECT TOP(10) nconst
2  FROM dim_principals
3  WHERE nconst NOT IN (SELECT nconst FROM dim_names)

```

Results

Messages

	nconst
1	nm2007716
2	nm2007716
3	nm2007716
4	nm2007716
5	nm2007716
6	nm2007716
7	nm2007716
8	nm2007716
9	nm2007716
10	nm2007716

▶ Run
❏ Cancel
🔌 Disconnect
🔄 Change

Database: mvp3DB

✓ Parse
📄 Enable SQLCMD
📖 To Notebook

```

1  SELECT * from dim_names
2  where nconst='nm2007716'

```

Results

Messages

nconst	primaryName	birthYear	deathYear
--------	-------------	-----------	-----------

Isso vale para a tabela gerada a partir do campo “characters”, a partir da tabela origem “title.principals.tsv.gz”, “dim\_characters”, que não tem correlação com facts\_tile.

▶ Run

❏ Cancel

🔌 Disconnect

🔄 Change

Database: mvp3DB

✓ Parse

📄 Enable SQLCMD

📓 To Notebook

1

SELECT top(1000) \*

2

FROM dim\_Characters

3

WHERE tconst NOT IN (SELECT tconst from facts\_title)

ResultsMessages

	tconst	nconst	characters
1	tt10378764	nm0083109	Pig Won't
2	tt10378764	nm0501306	Pig Will
3	tt10378764	nm0889120	Huckle
4	tt10378764	nm1517496	Bike Shop Bob
5	tt10378764	nm1517496	Goldbug
6	tt10378764	nm1846310	Lowly
7	tt10378764	nm3706393	Sally
8	tt10636886	nm10744505	Girl's Voice
9	tt10636886	nm9368183	Ben
1...	tt10808862	nm0917381	The Major
1...	tt10808862	nm4976391	Vanity Luther-Song
1...	tt10832286	nm11255457	Swat Team Member

▶ Run

❏ Cancel

🔌 Disconnect

🔄 Change

Database: mvp3DB

✓ Parse

📄 Enable SQLCMD

📓 To Notebook

1

SELECT \* from facts\_title

2

where tconst='tt10378764'

ResultsMessages

tconst	titleType	primaryTitle	originalTitle	isAdult	startYear
--------	-----------	--------------	---------------	---------	-----------

Na tabela dim\_names, nos campos “birthYear” e “DeathYear” existem muitos valores faltantes (Null), alguns sem sentido (negativos), e outros em domínios errados (1, 4, 12 etc...).

Run

Cancel

Disconnect

Change

Database:

mvp3DB

Estimated Plan

Enable Actual I

Parse

Enable SQLCMD

To Notebook

1

select PrimaryName, birthYear, deathYear, deathYear-birthYear

2

from dim\_names

3

Where birthYear < 1800

4

order by birthYear

Results

Messages

	PrimaryName	birthYear	deathYear	(No column name)
1	Xavier Castano	1	NULL	NULL
2	Lucio Anneo Seneca	4	65	61
3	Megan Liz Smith	12	NULL	NULL
4	Paul Walsh	21	NULL	NULL
5	Flavius Josephus	37	95	58
6	Plutarch	46	122	76
7	Titus Livius	59	17	-42
8	Pliny the Younger	61	113	52
9	Suetonius	69	140	71
10	Appian	95	165	70
11	Cassius Dio	163	235	72
12	Augustinus	354	430	76

Ln 3, Col 21

Spaces: 4

UTF-8

CRLF

720 rows

MSSQL





00:00:00

ccoe.database.windows.net : mvp3DB

Existem ainda valores nulos no campo runtimeMinutes da tabela facts\_title.




Query 1 ✕

 Run ☐ Cancel query  Save query  Export data as ▾  Show only Editor

```
1 SELECT count(*) AS ValoresNulosExecução
2 from [dbo].[facts_title]
3 where runtimeMinutes is NULL
```

Results Messages

 Search to filter items...

ValoresNulosExecução
7120757

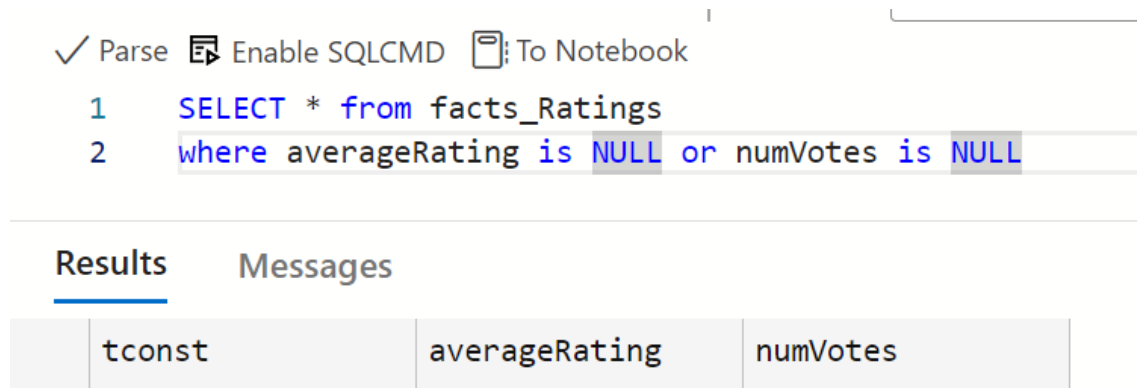
### 6.1.2 Soluções adotadas

Como observado no item anterior, a base de dados está bastante “suja”, com diversos dados importantes faltando, e outros com valores fora de um domínio.

Uma possível solução para os dados ausentes poderia ser realizar uma “mineração de dados”, tanto no próprio no site da IMDb, como em outros sites na internet buscando preencher os gaps encontrados. Outra possibilidade seria realizar uma análise exploratória mais detalhada, substituindo valores pela sua mediana, ou até eliminando aquelas com compôs NULL. Contudo isto foge do objetivo deste projeto.

Assim, para superarmos estas questões e podermos responder as perguntas objetivo deste projeto, realizamos algumas atividades de limpeza, como comandos que desabilitaram o bloqueio da criação das chaves estrangeiras para dados com problemas, transformando erros bloqueantes em “warnings”, e permitindo assim a carga nas tabelas. Checamos também a existência das chaves para responder as perguntas de forma adequada.

Com relação as perguntas, como mantivemos o arquivo de facts\_ratings, que não contém nulos (vide figura a seguir), em separado, não houve influencia de dados faltantes pois a operação de junção (INNER JOIN) faz o filtro.



The screenshot shows the Azure Data Studio interface. At the top, there are icons for 'Parse', 'Enable SQLCMD', and 'To Notebook'. Below these, a SQL query is entered in a text editor:

```
1 SELECT * from facts_Ratings
2 where averageRating is NULL or numVotes is NULL
```

Below the query editor, there are two tabs: 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with the following columns:

tconst	averageRating	numVotes
--------	---------------	----------

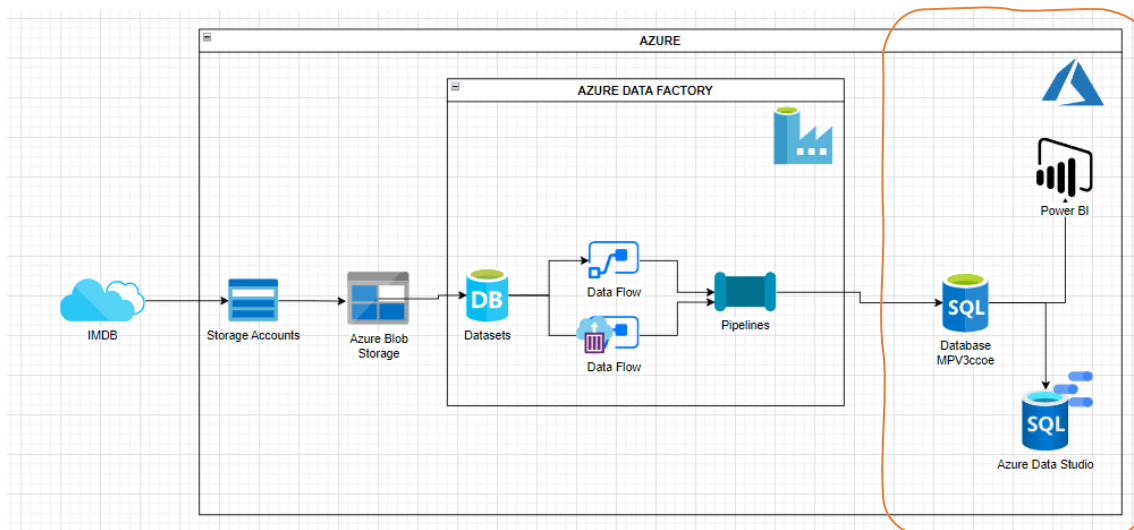
Com relação ao tempo de execução dos filmes, na qual aparecem nulos, como utilizamos a função AVG elas foram descartadas do cálculo.

## 6.2 Solução do Problema

Para obtenção das respostas as perguntas propostas, foram utilizadas 2 ferramentas: Azure Data Studio, para gerar as tabelas de resposta e Power BI, para geração dos gráficos. Ambos os serviços foram conectados diretamente no Azure SQL Database.

Nas respostas abaixo consideramos “popularidade” como número de votos do filme (quanto maior o número de votos, mais popular é o filme) e “pontuação” como o valor do IMDB obtido (0 a 10).

*Figura 25 - Azure data Studio e Power BI*



Consideramos ainda que para termos uma pontuação razoável um mínimo de 10.000 votos precisa ser obtido.

## 6.2.1 Qual a popularidade dos filmes de James Bond

### Azure Data Studio

Run Cancel Disconnect Change Database: mvp3DB Estimated Plan Enable Actual Plan

Parse Enable SQLCMD To Notebook

```

1 SELECT f.primaryTitle, SUM(r.numVotes) as Popularidade
2 FROM facts_title f INNER JOIN dim_Characters c on f.tconst=c.tconst
3     INNER JOIN facts_ratings r ON r.tconst= c.tconst
4 WHERE c.characters like 'James Bond' and f.titleType='movie' and r.numVotes>10000
5 GROUP by f.primaryTitle
6 ORDER BY Popularidade DESC

```

Results Messages

	primaryTitle	Popularidade
1	Skyfall	716762
2	Casino Royale	678667
3	Quantum of Solace	462211
4	Spectre	456015
5	No Time to Die	428417
6	GoldenEye	265240
7	Die Another Day	225289
8	The World Is Not Enough	206468
9	Tomorrow Never Dies	201028
10	Goldfinger	197809
11	Dr. No	174824

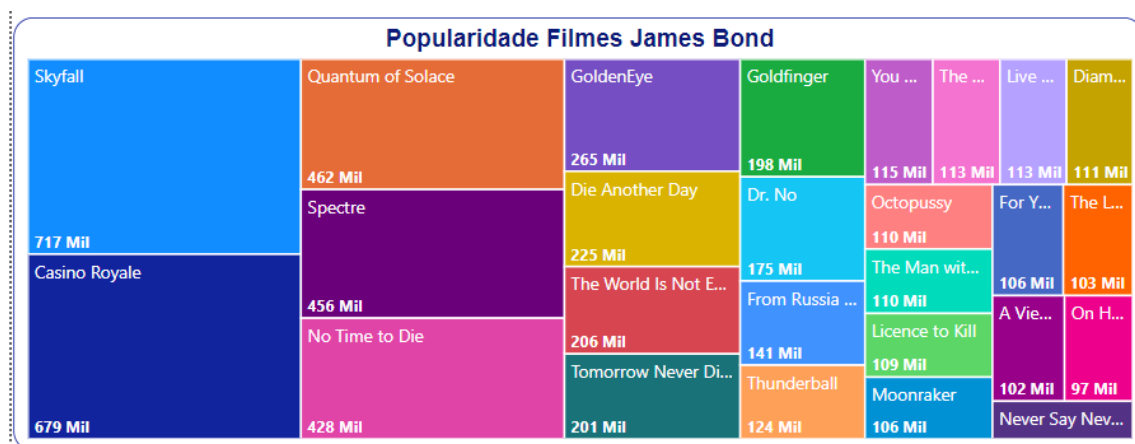
Ln 6, Col 27 Spaces: 4 UTF-8 CRLF 26 rows MSSQL 00:00:00 ccoe.database.windows.net : mvp3DB

Resultado completo:

primaryTitle	Popularidade
Skyfall	716762

Casino Royale	678667
Quantum of Solace	462211
Spectre	456015
No Time to Die	428417
GoldenEye	265240
Die Another Day	225289
The World Is Not Enough	206468
Tomorrow Never Dies	201028
Goldfinger	197809
Dr. No	174824
From Russia with Love	141276
Thunderball	123931
You Only Live Twice	114603
The Spy Who Loved Me	113449
Live and Let Die	112628
Diamonds Are Forever	111339
Octopussy	110418
The Man with the Golden Gun	110394
Licence to Kill	109378
Moonraker	105995
For Your Eyes Only	105698
The Living Daylights	103325
A View to a Kill	102306
On Her Majesty's Secret Service	96554
Never Say Never Again	71231

Visual (powerBI)



6.2.2 Qual a pontuação dos filmes de James Bond

Azure Data Studio

Run

Cancel

Disconnect

Change

Database:

mvp3DB

Estimated Plan

Enable Actual

Parse

Enable SQLCMD

To Notebook

```

1  SELECT f.primaryTitle, r.averageRating as Pontuacao
2  FROM facts_title f INNER JOIN dim_Characters c ON f.tconst=c.tconst
3  INNER JOIN facts_ratings r ON r.tconst= c.tconst
4  WHERE c.characters LIKE 'James Bond' and f.titleType='movie' and r.numVotes>10000
5  ORDER BY Pontuacao Desc

```

Results

Messages

	primaryTitle	Pontuacao
1	Casino Royale	8.0
2	Skyfall	7.8
3	Goldfinger	7.7
4	From Russia with Love	7.3
5	No Time to Die	7.3
6	Dr. No	7.2
7	GoldenEye	7.2
8	The Spy Who Loved Me	7.0
9	Thunderball	6.9
10	You Only Live Twice	6.8
11	Spectre	6.8

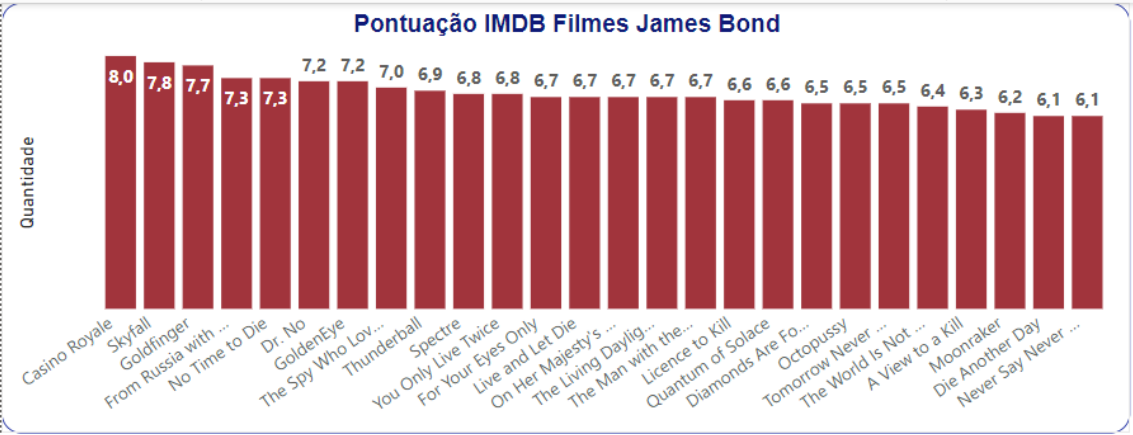
Ln 5, Col 24 Spaces: 4 UTF-8 CRLF 26 rows MSSQL 00:00:01 ccoe.database.windows.net : mvp3DB

## Resultado Completo

primaryTitle	Pontuacao
Casino Royale	8.0
Skyfall	7.8
Goldfinger	7.7
From Russia with Love	7.3
No Time to Die	7.3
GoldenEye	7.2
Dr. No	7.2
The Spy Who Loved Me	7.0
Thunderball	6.9
You Only Live Twice	6.8
Spectre	6.8
The Living Daylights	6.7
For Your Eyes Only	6.7
On Her Majesty's Secret Service	6.7
Live and Let Die	6.7
The Man with the Golden Gun	6.7
Quantum of Solace	6.6
Licence to Kill	6.6
Diamonds Are Forever	6.5
Octopussy	6.5
Tomorrow Never Dies	6.5
The World Is Not Enough	6.4

<b>A View to a Kill</b>	6.3
<b>Moonraker</b>	6.2
<b>Die Another Day</b>	6.1
<b>Never Say Never Again</b>	6.1

### Visual (power BI)



### 6.2.3 Quais são os gêneros de filmes mais populares

#### Azure Data Studio

Run Cancel Disconnect Change
Database: mvp3DB
Estimated Plan Enable Actual P

☒ Parse
☒ Enable SQLCMD
☐ To Notebook

```

1 SELECT g.genre, SUM(r.numVotes) As Popularidade
2 FROM dim_genres g INNER JOIN facts_ratings r on g.tconst=r.tconst
3 GROUP BY g.genre
4 ORDER BY Popularidade DESC

```

Results

Messages

	genre	Popularidade
1	Drama	727666892
2	Action	452560351
3	Comedy	425886436
4	Adventure	361059088
5	Crime	281332739
6	Thriller	204725179
7	Romance	158374971
8	Sci-Fi	155602172
9	Mystery	154278735
10	Horror	126988249
11	Fantasy	126255289
12	Animation	115494690

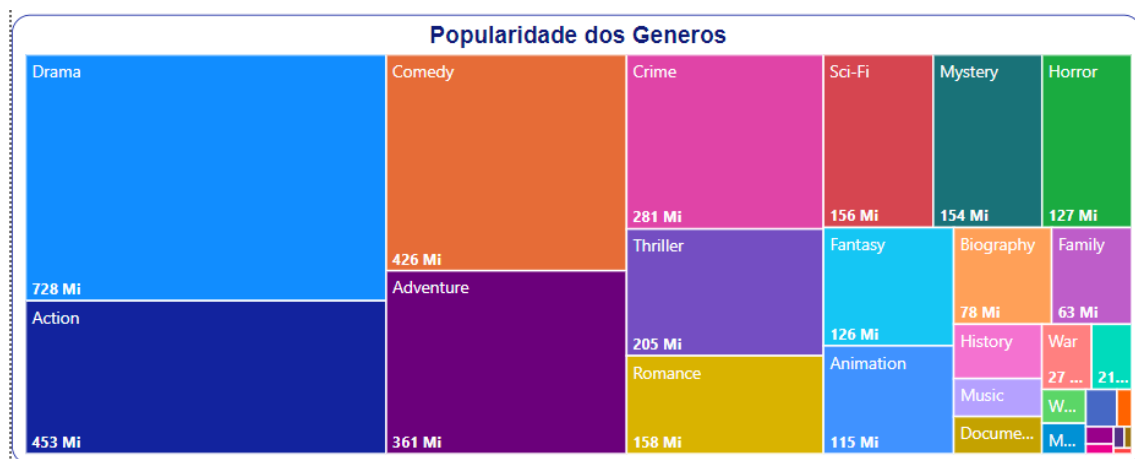
Ln 4, Col 22 Spaces: 4 UTF-8 CRLF 28 rows MSSQL 00:00:02 ccoe.database.windows.net : mvp3DB

#### Resultado completo

genre	Popularidade
Drama	727666892

Action	452560351
Comedy	425886436
Adventure	361059088
Crime	281332739
Thriller	204725179
Romance	158374971
Sci-Fi	155602172
Mystery	154278735
Horror	126988249
Fantasy	126255289
Animation	115494690
Biography	77677171
Family	63451833
History	39278113
Music	27457243
Documentary	26997430
War	26625783
Sport	21396309
Western	12207588
Musical	11015312
Short	9511853
Reality-TV	4474116
Film-Noir	3934127
Talk-Show	2241707
Game-Show	1863926
News	1299681
Adult	849689

## Visual PowerBI



6.2.4 Quais os gêneros com as melhores pontuações (acima de 6.0)

Azure Data Studio

Run Cancel Disconnect Change Database:.mvp3DB Estimated Plan Enable Actual F

Parse Enable SQLCMD To Notebook

```

1 SELECT g.genre, CAST(AVG(r.averageRating) AS DECIMAL(5,2)) As MediaPontuação
2 FROM dim_genres g INNER JOIN facts_ratings r on g.tconst=r.tconst
3 WHERE r.numVotes>10000
4 GROUP BY g.genre
5 HAVING (AVG(r.averageRating)>6)
6 ORDER BY MediaPontuação DESC
  
```

Results Messages

	genre	MediaPontuação
1	News	8.06
2	Film-Noir	7.67
3	Talk-Show	7.67
4	Game-Show	7.59
5	Short	7.55
6	Documentary	7.54
7	Animation	7.52
8	History	7.31
9	War	7.29
10	Biography	7.22
11	Western	7.17

Ln 6, Col 29 Spaces: 4 UTF-8 CRLF 27 rows MSSQL 00:00:00 ccoe.database.windows.net :.mvp3DB

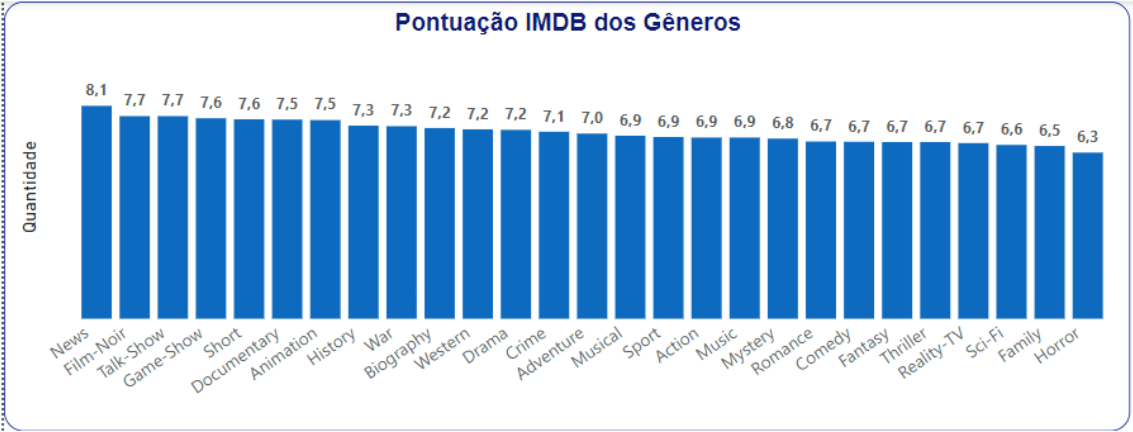
Resultado completo

genre	MediaPontuação
News	8.06
Talk-Show	7.67
Film-Noir	7.67
Game-Show	7.59
Short	7.55
Documentary	7.54
Animation	7.52
History	7.31
War	7.29
Biography	7.22
Western	7.17
Drama	7.15
Crime	7.08
Adventure	7.01
Musical	6.93
Sport	6.88
Music	6.86
Action	6.86
Mystery	6.82
Romance	6.71
Comedy	6.70
Fantasy	6.69



Thriller	6.69
Reality-TV	6.65
Sci-Fi	6.58
Family	6.54
Horror	6.29

Visual (powerBI)



### 6.2.5 Quais os 10 filmes com maior popularidade de Steven Spielberg

Azure data Studio

Run Cancel Disconnect Change
Database: mvp3DB
Estimated Plan Enable Actual

☒ Parse
☒ Enable SQLCMD
☐ To Notebook

```

1 SELECT top (10) f.primaryTitle, n.primaryName, r.numvotes as Popularidade
2 FROM facts_ratings r INNER JOIN facts_title f on f.tconst=r.tconst
3     INNER JOIN dim_director d on d.tconst=f.tconst
4     INNER JOIN dim_names n ON n.nconst=d.nconst
5 WHERE n.primaryName='Steven Spielberg'
6 ORDER BY r.numVotes desc

```

**Results**
Messages

	primaryTitle	primaryName	Popularidade
1	Saving Private Ryan	Steven Spielberg	1448909
2	Schindler's List	Steven Spielberg	1406454
3	Catch Me If You Can	Steven Spielberg	1047055
4	Jurassic Park	Steven Spielberg	1033739
5	Indiana Jones and the Raiders of the Lost Ark	Steven Spielberg	1012521
6	Indiana Jones and the Last Crusade	Steven Spielberg	791216
7	Jaws	Steven Spielberg	639216
8	Minority Report	Steven Spielberg	571395
9	Indiana Jones and the Temple of Doom	Steven Spielberg	522033
10	Indiana Jones and the Kingdom of the Crystal Skull	Steven Spielberg	480879

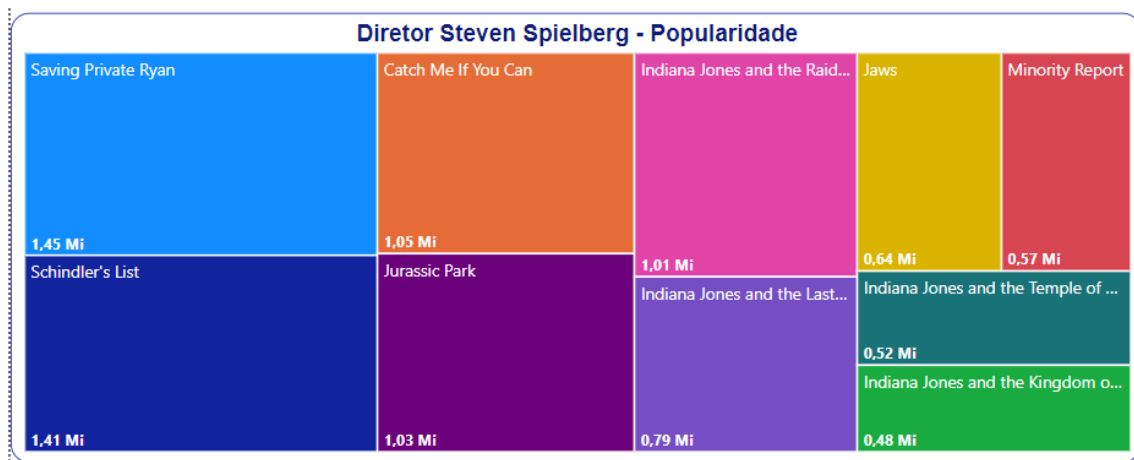
Ln 6, Col 25
Spaces: 4
UTF-8
CRLF
10 rows
MSSQL
00:00:04
ccoe.database.windows.net : mvp3DB

Resultado Completo

primaryTitle	primaryName	Popularidade
Saving Private Ryan	Steven Spielberg	1448909

Schindler's List	Steven Spielberg	1406454
Catch Me If You Can	Steven Spielberg	1047055
Jurassic Park	Steven Spielberg	1033739
Indiana Jones and the Raiders of the Lost Ark	Steven Spielberg	1012521
Indiana Jones and the Last Crusade	Steven Spielberg	791216
Jaws	Steven Spielberg	639216
Minority Report	Steven Spielberg	571395
Indiana Jones and the Temple of Doom	Steven Spielberg	522033
Indiana Jones and the Kingdom of the Crystal Skull	Steven Spielberg	480879

## Visual



## 6.2.6 Quais os 10 filmes com maior pontuação de Steven Spielberg

### Azure Data Studio

Run Cancel Disconnect Change Database:.mvp3DB Estimated Plan Enable Actual PI  
Parse Enable SQLCMD To Notebook

```
1 SELECT TOP (10) f.primaryTitle, n.primaryName, r.averagerating as Pontuacao
2 FROM facts_ratings r INNER JOIN facts_title f ON f.tconst=r.tconst
3     INNER JOIN dim_director d ON d.tconst=f.tconst
4     INNER JOIN dim_names n ON n.nconst=d.nconst
5 WHERE n.primaryName='Steven Spielberg'
6 ORDER BY r.averagerating DESC
```

Results Messages

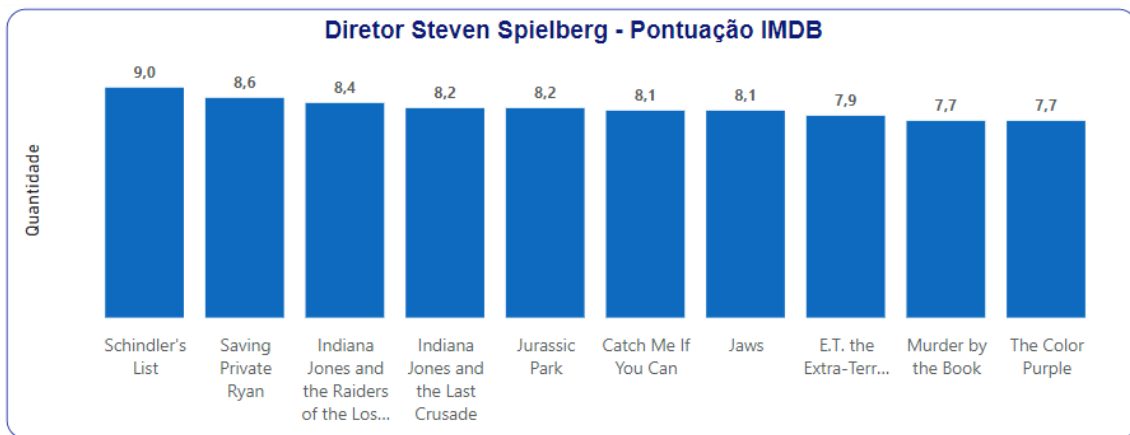
	primaryTitle	primaryName	Pontuacao
1	Schindler's List	Steven Spielberg	9.0
2	Saving Private Ryan	Steven Spielberg	8.6
3	Indiana Jones and the Raiders of the Lost Ark	Steven Spielberg	8.4
4	Indiana Jones and the Last Crusade	Steven Spielberg	8.2
5	Jurassic Park	Steven Spielberg	8.2
6	Jaws	Steven Spielberg	8.1
7	Catch Me If You Can	Steven Spielberg	8.1
8	E.T. the Extra-Terrestrial	Steven Spielberg	7.9
9	Murder by the Book	Steven Spielberg	7.7
10	The Color Purple	Steven Spielberg	7.7

Ln 6, Col 30 Spaces: 4 UTF-8 CRLF 10 rows MSSQL 00:00:01 ccoe.database.windows.net :.mvp3DB

Resultado completo

primaryTitle	primaryName	Pontuacao
Schindler's List	Steven Spielberg	9.0
Saving Private Ryan	Steven Spielberg	8.6
Indiana Jones and the Raiders of the Lost Ark	Steven Spielberg	8.4
Indiana Jones and the Last Crusade	Steven Spielberg	8.2
Jurassic Park	Steven Spielberg	8.2
Catch Me If You Can	Steven Spielberg	8.1
Jaws	Steven Spielberg	8.1
E.T. the Extra-Terrestrial	Steven Spielberg	7.9
Murder by the Book	Steven Spielberg	7.7
The Color Purple	Steven Spielberg	7.7

Visual



6.2.7 Quais os 10 diretores de filmes com maiores médias de pontuação com mais de 5 filmes realizados?

### Azure data Studio

Run Cancel Disconnect Change Database:.mvp3DB Estimated Plan Enable Actual I

✓ Parse Enable SQLCMD To Notebook

```

1 SELECT top (10) n.primaryName, CAST(AVG(r.averagerating) AS DECIMAL(10,2)) as MediaPontuaçãoDiretor
2 FROM facts_ratings r INNER JOIN facts_title f ON f.tconst=r.tconst
3     INNER JOIN dim_director d ON d.tconst=f.tconst
4     INNER JOIN dim_names n ON n.nconst=d.nconst
5 WHERE NumVotes >10000 and f.titleType='movie'
6 GROUP BY n.primaryName
7 HAVING (COUNT(*) > 5)
8 ORDER BY MediaPontuaçãoDiretor DESC

```

**Results** Messages

	primaryName	MediaPontuaçãoDiretor	NumFilmes
1	Ertem Egilmez	8.74	7
2	Sergio Leone	8.22	6
3	Christopher Nolan	8.20	12
4	Akira Kurosawa	8.06	17
5	Andrei Tarkovsky	8.00	7
6	Hayao Miyazaki	7.96	11
7	Frank Capra	7.94	8
8	Quentin Tarantino	7.94	14
9	Fritz Lang	7.93	7

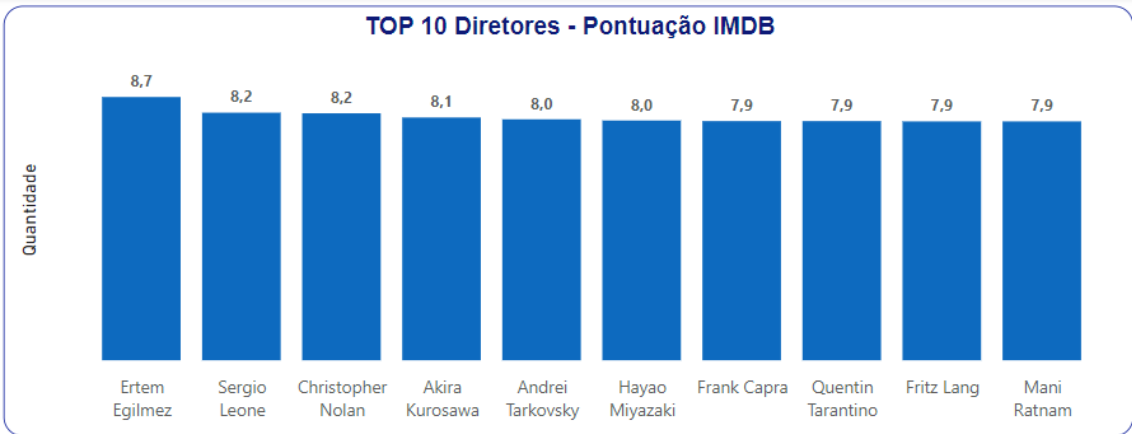
Ln 8, Col 36 Spaces: 4 UTF-8 CRLF 10 rows MSSQL 00:00:00 ccoe.database.windows.net : mvp3DB

### Resultado completo

primaryName	MediaPontuaçãoDiretor	NumFilmes
Ertem Egilmez	8.74	7
Sergio Leone	8.22	6
Christopher Nolan	8.20	12
Akira Kurosawa	8.06	17

<b>Andrei Tarkovsky</b>	8.00	7
<b>Hayao Miyazaki</b>	7.96	11
<b>Quentin Tarantino</b>	7.94	14
<b>Frank Capra</b>	7.94	8
<b>Fritz Lang</b>	7.93	7
<b>Mani Ratnam</b>	7.93	8

## Visual



## 6.2.8 Quais os são 10 diretores de filmes mais populares?

### Azure Data Studio

Run Cancel Disconnect Change Database: mvp3DB Estimated Plan Enable Actual F

Parse Enable SQLCMD To Notebook

```

1 SELECT TOP (10) n.primaryName, SUM(r.Numvotes) as PopularidadeDiretor, count(*) as NumFilmes
2 FROM facts_ratings r INNER JOIN facts_title f ON f.tconst=r.tconst
3     INNER JOIN dim_director d ON d.tconst=f.tconst
4     INNER JOIN dim_names n ON n.nconst=d.nconst
5 WHERE NumVotes >10000 and f.titleType='movie'
6 GROUP BY n.primaryName
7 HAVING (COUNT(*) > 5)
8 ORDER BY PopularidadeDiretor DESC

```

**Results** Messages

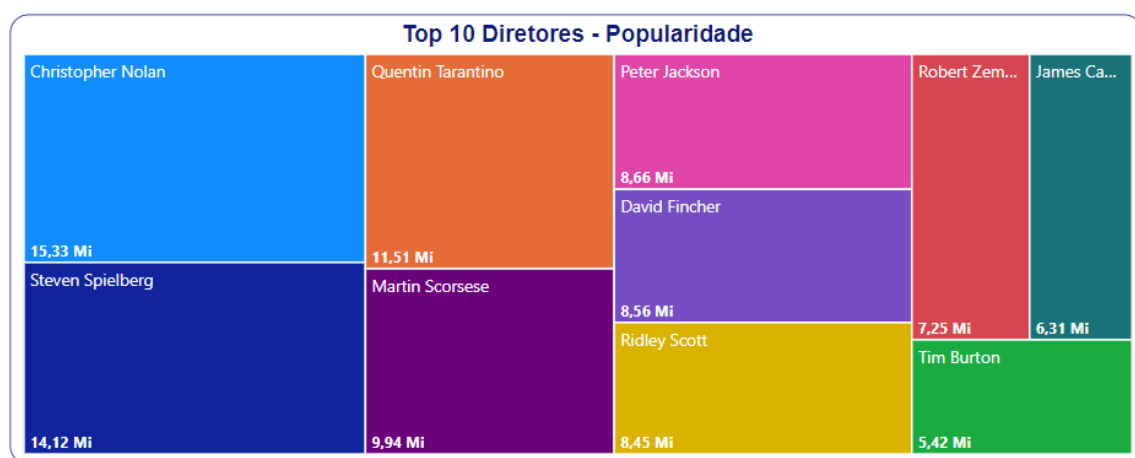
	primaryName	PopularidadeDiretor	NumFilmes
1	Christopher Nolan	15330770	12
2	Steven Spielberg	14115069	34
3	Quentin Tarantino	11505168	14
4	Martin Scorsese	9940049	28
5	Peter Jackson	8663631	14
6	David Fincher	8556649	11
7	Ridley Scott	8445499	27
8	Robert Zemeckis	7252067	20
9	James Cameron	6306824	8

Ln 8, Col 34 Spaces: 4 UTF-8 CRLF 10 rows MSSQL 00:00:00 ccoe.database.windows.net : mvp3DB

### Resultado completo

primaryName	PopularidadeDiretor	NumFilmes
Christopher Nolan	15330770	12
Steven Spielberg	14115069	34
Quentin Tarantino	11505168	14
Martin Scorsese	9940049	28
Peter Jackson	8663631	14
David Fincher	8556649	11
Ridley Scott	8445499	27
Robert Zemeckis	7252067	20
James Cameron	6306824	8
Tim Burton	5422941	19

## Visual



6.2.9 Qual é o tempo de execução típico para filmes de cada gênero?

Azure Data Studios

Run

Cancel

Disconnect

Change

Database: mvp3DB

Estimated Plan

Enable Actual F

Parse

Enable SQLCMD

To Notebook

1

SELECT g.genre, AVG(f.runtimeMinutes) AS MediaMinutos

2

FROM dim\_genres g INNER JOIN facts\_title f ON g.tconst=f.tconst

3

WHERE f.titleType='movie'

4

GROUP BY g.genre

5

ORDER BY g.genre

Results

Messages

	genre	MediaMinutos
1	Action	100
2	Adult	80
3	Adventure	92
4	Animation	81
5	Biography	88
6	Comedy	92
7	Crime	94
8	Documentary	78
9	Drama	96
10	Family	91
11	Fantasy	93

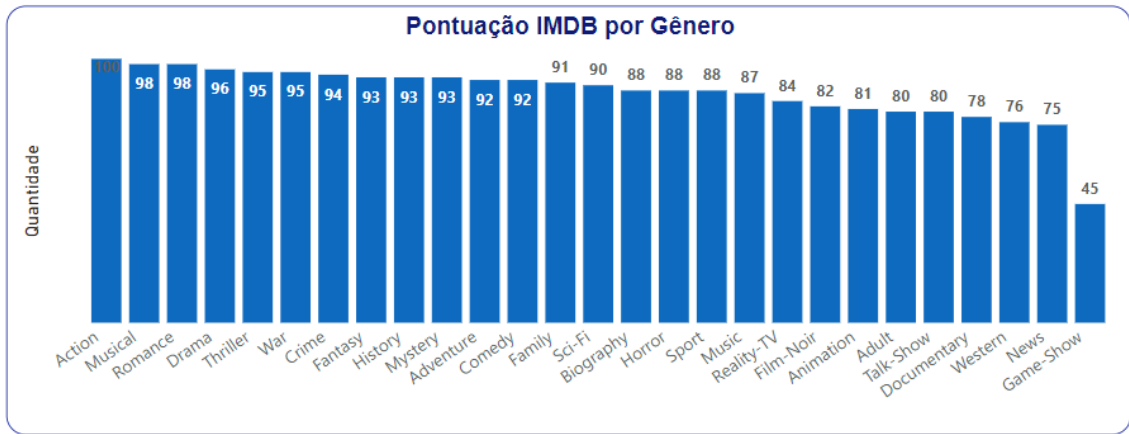
Ln 5, Col 17   Spaces: 4   UTF-8   CRLF   28 rows   MSSQL   00:00:02   ccoe.database.windows.net : mvp3DB

Resultado completo

genre	MediaMinutos
Action	100
Adult	80
Adventure	92
Animation	81
Biography	88
Comedy	92
Crime	94
Documentary	78
Drama	96
Family	91
Fantasy	93
Film-Noir	82
Game-Show	45
History	93
Horror	88
Music	87
Musical	98
Mystery	93
News	75
Reality-TV	84
Romance	98
Sci-Fi	90

<b>Sport</b>	88
<b>Talk-Show</b>	80
<b>Thriller</b>	95
<b>War</b>	95
<b>Western</b>	76

# Visual



6.2.10Qual a pontuação média por gênero de filme, entre os anos de 2020 e 2022?

## Azure Data Studio

Run
Cancel
Disconnect
Change

Database: mvp3DB

Estimated Plan
Enable Actual I

Parse
Enable SQLCMD
To Notebook

```

1 SELECT g.genre, f.startYear, COUNT(*) as FilmesporGenero, avg(r.averageRating) AS PontuaçãoMedia
2 from dim_genres g inner join facts_title f on g.tconst=f.tconst
3     INNER JOIN facts_Ratings R ON r.tconst=f.tconst
4 where f.titleType='movie' and f.startYear BETWEEN 2020 and 2022
5 GROUP BY g.genre, f.startYear
6 Order by g.genre, f.startYear ASC

```

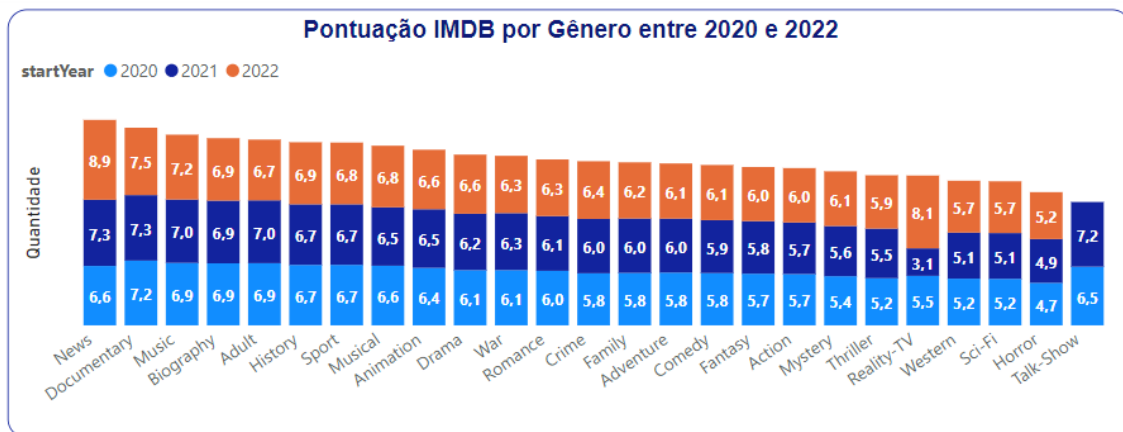
Results
Messages

	genre	startYear	FilmesporGenero	PontuaçãoMedia
1	Action	2020	640	5.682187
2	Action	2021	843	5.710557
3	Action	2022	892	6.030044
4	Adult	2020	7	6.885714
5	Adult	2021	3	6.966666
6	Adult	2022	7	6.728571
7	Adventure	2020	387	5.843927
8	Adventure	2021	359	5.980501
9	Adventure	2022	414	6.111352
10	Animation	2020	216	6.368981

Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
74 rows
MSSQL
00:00:00
ccoe.database.windows.net : mvp3DB



## Visual



## 7 Conclusão

O objetivo deste projeto foi o de realizar uma análise dos títulos (filmes, seriados de TV) publicados e responder a diversas perguntas mais comuns sobre o mercado de mídia, utilizando as informações disponibilizadas pela plataforma IMDB.

A execução deste projeto envolveu a realização das seguintes atividades:

- Entender os dados no database disponibilizado pelo IMDb;
- Modelar o banco de dados usando a técnica o modelo Entidade-Relacionamento (ER) e diagramas de esquema lógico relacional;
- Projetar um banco de dados relacional para ingestão dos dados;
- Criar um servidor relacional AZURE SQL Server, e um banco de dados relacional com as devidas restrições;
- Criar um repositório de dados no Azure e armazenar os dados fonte oriundos do IMDb;
- Realizar um ETL, utilizando o Azure Data Factory, onde extraímos os dados dos arquivos tsv (separados por tabulações), e transformando-os, e carregando-os em tabelas normalizadas e reestruturadas, segundo a modelagem prevista;
- Carregar os dados gerados nas tabelas com restrições de chave primária e estrangeira, e;

- Responder as questões colocadas utilizando o Azure Data Studio e Visualizar as respostas utilizando POWER BI.

Este projeto mostrou apenas algumas possibilidades do que pode ser feito com esses dados do IMDb. Com estes dados poderíamos realizar diversas outras análises. Uma adição interessante a estes dados seria a inclusão dos dados das bilheterias obtidas pelos títulos.

Por fim, uma possível extensão do uso destes dados seria investigar mais detalhadamente as tendências, realizando análises estatísticas e possivelmente até usando alguns algoritmos de aprendizado de máquina.