Final Presentation

Edward Zhou, David Ilitzky

Area of Interest

- Factors that influence popularity of a movie
  - Region
  - Directors
  - Genres

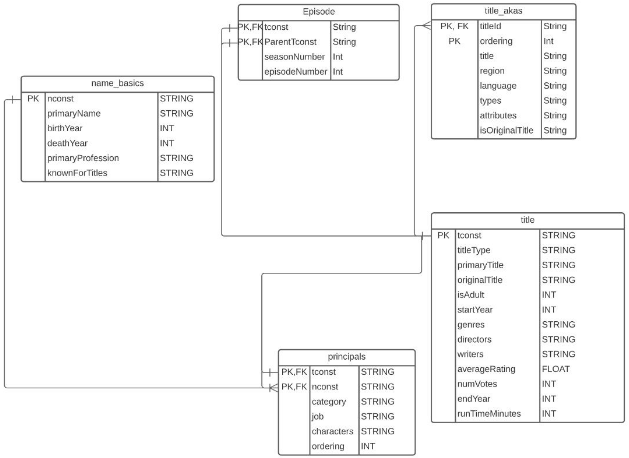
IMDb

Dataset Overview

- IMDb
  - Title.akas
  - title.basics
  - title.crew
  - Title.episode
  - Title.principals
  - Title.ratings
  - name.basics

- Bollywood
  - Bollywood
  - Bollywood_actress
  - bollywood_actor

Staging/Modeled Tables



Episode

PK,FK	tconst	String
PK,FK	ParentTconst	String
	seasonNumber	Int
	episodeNumber	Int

title_akas

PK, FK	titleId	String
PK	ordering	Int
	title	String
	region	String
	language	String
	types	String
	attributes	String
	isOriginalTitle	String

name_basics

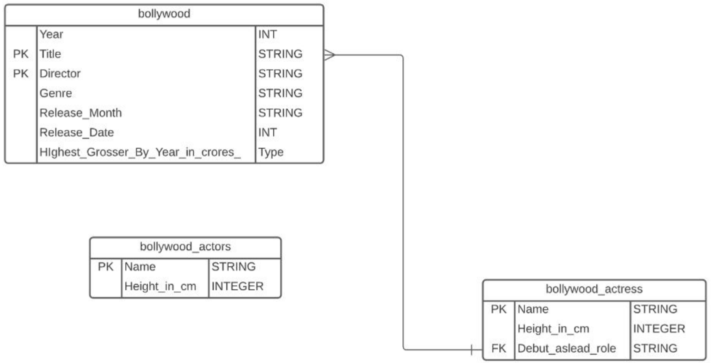
PK	nconst	STRING
	primaryName	STRING
	birthYear	INT
	deathYear	INT
	primaryProfession	STRING
	knownForTitles	STRING

title

PK	tconst	STRING
	titleType	STRING
	primaryTitle	STRING
	originalTitle	STRING
	isAdult	INT
	startYear	INT
	genres	STRING
	directors	STRING
	writers	STRING
	averageRating	FLOAT
	numVotes	INT
	endYear	INT
	runTimeMinutes	INT

principals

PK,FK	tconst	STRING
PK,FK	nconst	STRING
	category	STRING
	job	STRING
	characters	STRING
	ordering	INT



bollywood

	Year	INT
PK	Title	STRING
PK	Director	STRING
	Genre	STRING
	Release_Month	STRING
	Release_Date	INT
	HIghest_Grosser_By_Year_in_crores_	Type

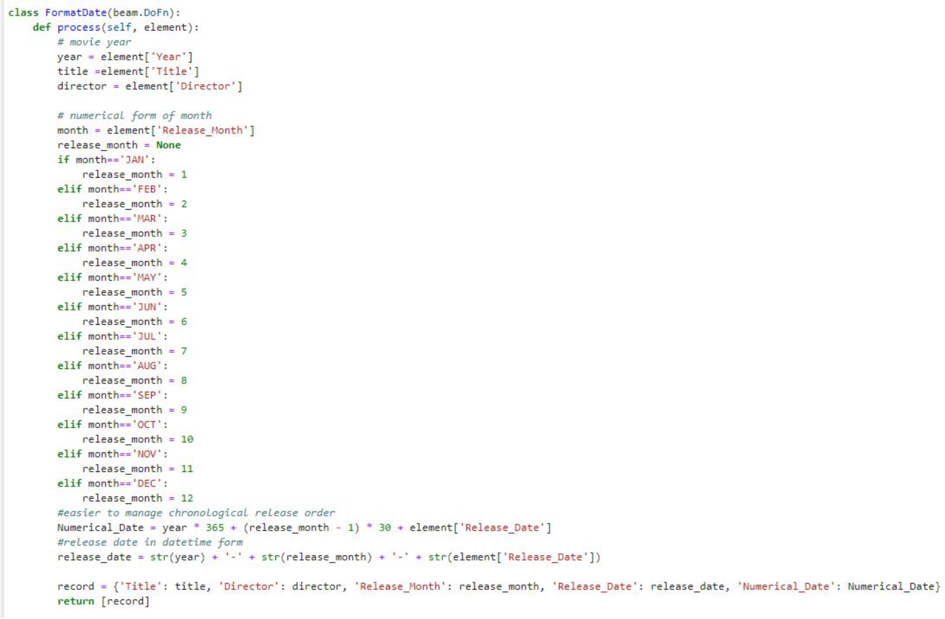
bollywood_actors

PK	Name	STRING
	Height_in_cm	INTEGER

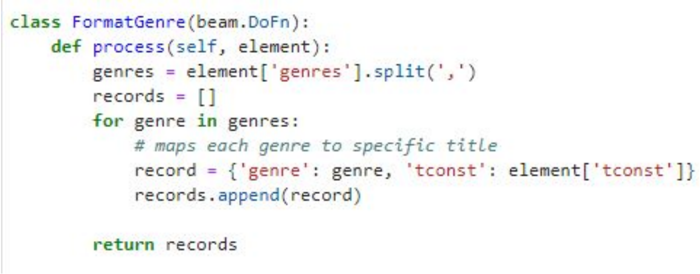
bollywood_actress

PK	Name	STRING
	Height_in_cm	INTEGER
FK	Debut_aslead_role	STRING

Beam Pipelines

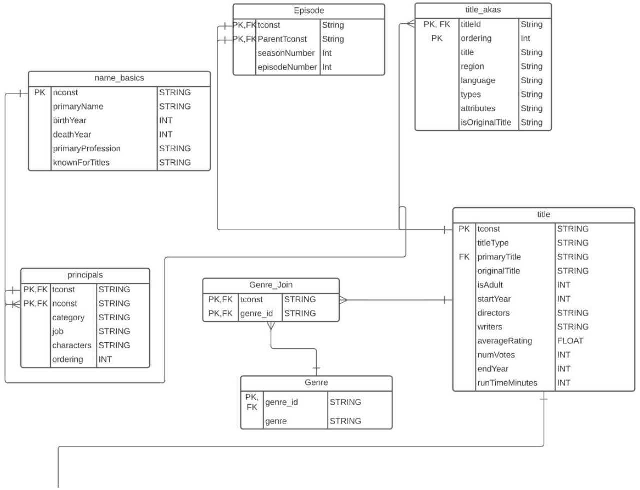


```
class FormatDate(beam.DoFn):
    def process(self, element):
        # movie year
        year = element['Year']
        title =element['Title']
        director = element['Director']

        # numerical form of month
        month = element['Release_Month']
        release_month = None
        if month=='JAN':
            release_month = 1
        elif month=='FEB':
            release_month = 2
        elif month=='MAR':
            release_month = 3
        elif month=='APR':
            release_month = 4
        elif month=='MAY':
            release_month = 5
        elif month=='JUN':
            release_month = 6
        elif month=='JUL':
            release_month = 7
        elif month=='AUG':
            release_month = 8
        elif month=='SEP':
            release_month = 9
        elif month=='OCT':
            release_month = 10
        elif month=='NOV':
            release_month = 11
        elif month=='DEC':
            release_month = 12
        #easier to manage chronological release order
        Numerical_Date = year * 365 + (release_month - 1) * 30 + element['Release_Date']
        #release date in datetime form
        release_date = str(year) + '-' + str(release_month) + '-' + str(element['Release_Date'])

        record = {'Title': title, 'Director': director, 'Release_Month': release_month, 'Release_Date': release_date, 'Numerical_Date': Numerical_Date}
        return [record]
```



```
class FormatGenre(beam.DoFn):
    def process(self, element):
        genres = element['genres'].split(',')
        records = []
        for genre in genres:
            # maps each genre to specific title
            record = {'genre': genre, 'tconst': element['tconst']}
            records.append(record)

        return records
```

Modeled Tables after Beam Transforms



Episode

PK,FK	tconst	String
PK,FK	ParentTconst	String
	seasonNumber	Int
	episodeNumber	Int

title_akas

PK, FK	titleId	String
PK	ordering	Int
	title	String
	region	String
	language	String
	types	String
	attributes	String
	isOriginalTitle	String

name_basics

PK	nconst	STRING
	primaryName	STRING
	birthYear	INT
	deathYear	INT
	primaryProfession	STRING
	knownForTitles	STRING

title

PK	tconst	STRING
	titleType	STRING
FK	primaryTitle	STRING
	originalTitle	STRING
	isAdult	INT
	startYear	INT
	directors	STRING
	writers	STRING
	averageRating	FLOAT
	numVotes	INT
	endYear	INT
	runTimeMinutes	INT

principals

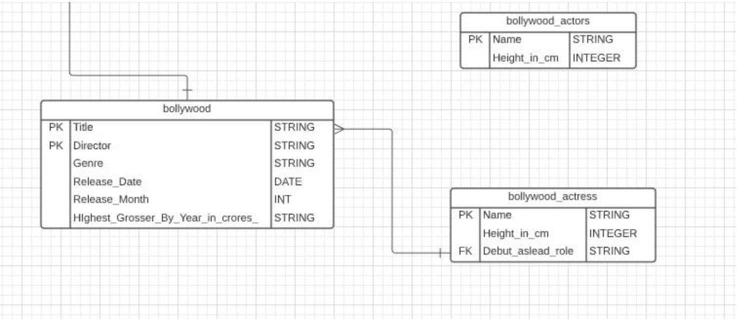
PK,FK	tconst	STRING
PK,FK	nconst	STRING
	category	STRING
	job	STRING
	characters	STRING
	ordering	INT

Genre_Join

PK,FK	tconst	STRING
PK,FK	genre_id	STRING

Genre

PK, FK	genre_id	STRING
	genre	STRING



bollywood_actors

PK	Name	STRING
	Height_in_cm	INTEGER

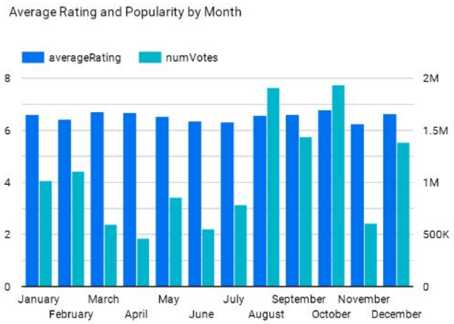
bollywood

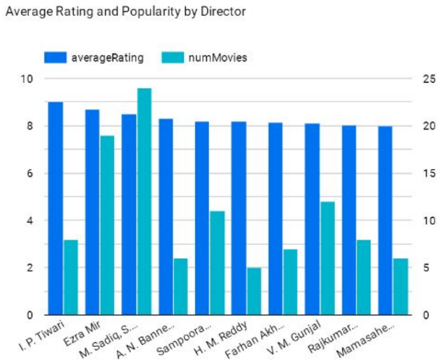
PK	Title	STRING
PK	Director	STRING
	Genre	STRING
	Release_Date	DATE
	Release_Month	INT
	HIghest_Grosser_By_Year_in_crores_	STRING

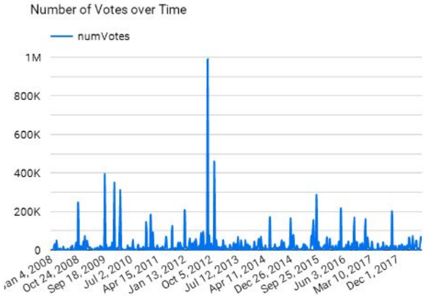
bollywood_actress

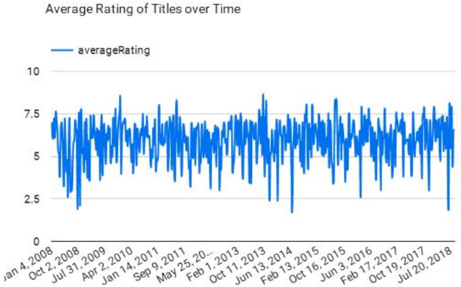
PK	Name	STRING
	Height_in_cm	INTEGER
FK	Debut_aslead_role	STRING

Cross-Dataset Queries


Average Rating and Popularity by Month

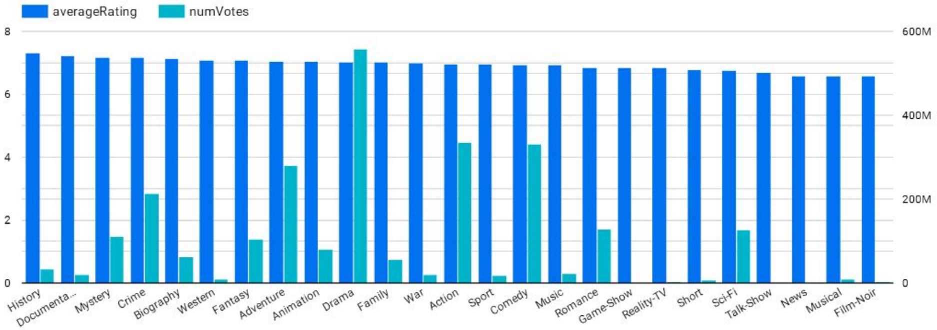

Average Rating and Popularity by Director


Number of Votes over Time

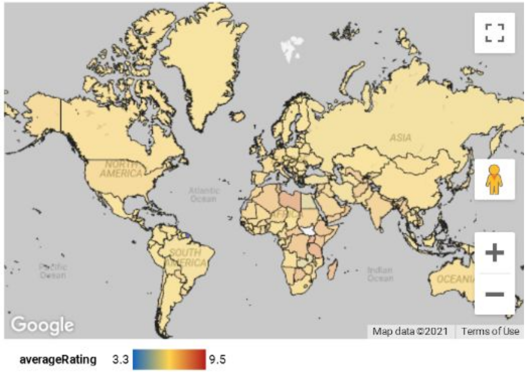

Average Rating of Titles over Time

Data Visualization

Average Rating and Popularity by Genre



Average Rating of Titles per Region



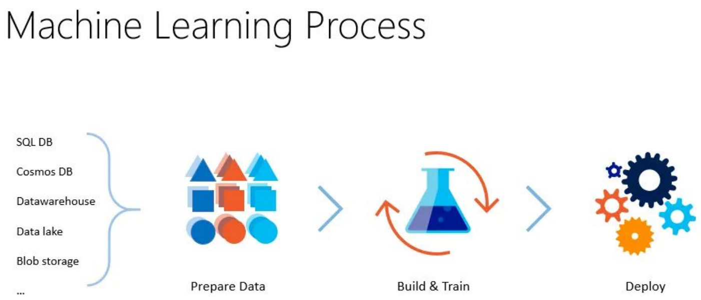
Challenges and Future Improvements

Challenges:

- Constructing the ParDo Transformations (mainly genre tables)
- Maintaining Referential Integrity and Unique Primary Key in every modified table

Future Improvements:

- Explore other factors that influence movie ratings:
    - Explore specific actors connected to certain genres
    - Explore more complicated (machine learning) models to predict the ratings


Machine Learning Process
SQL DB
Cosmos DB
Datawarehouse
Data lake
Blob storage
...
Prepare Data
Build & Train
Deploy