CS3563: DBMS II

Assignment 2

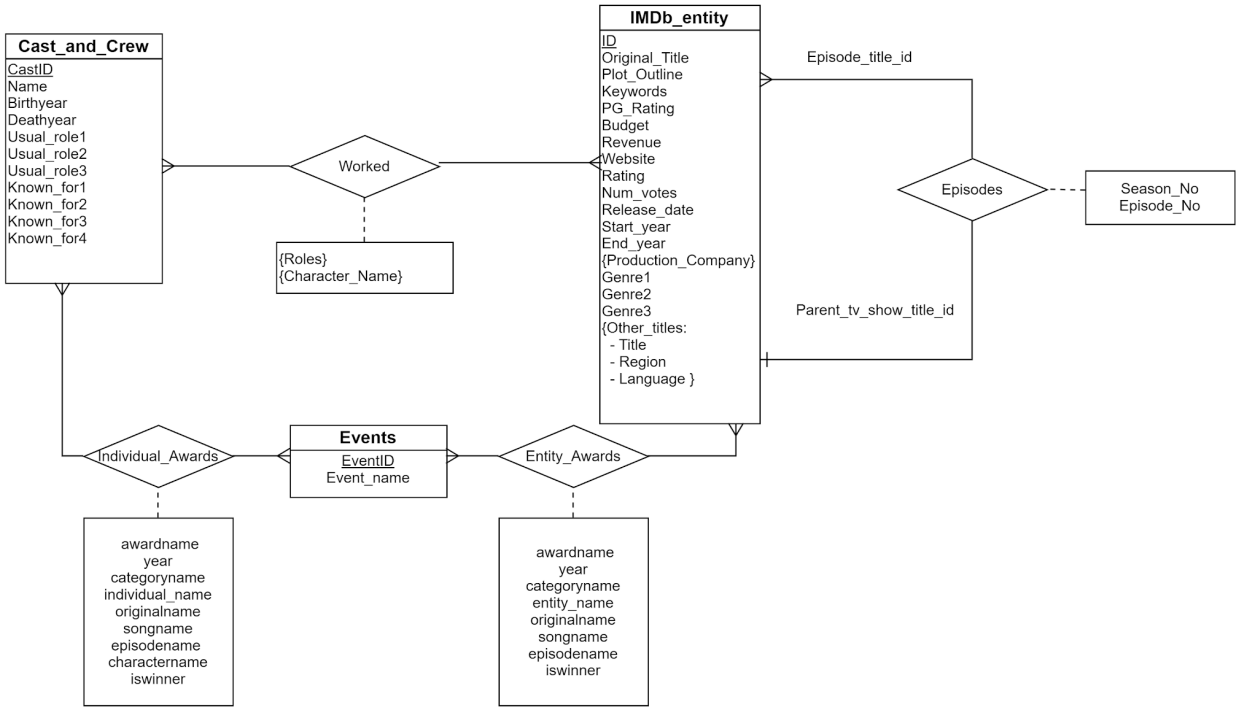
Group 3

Shreyas Havaldar: CS18BTECH11042
Sandeep Kumar: CS18BTECH11041
Devi Aishwarya: ES18BTECH11006
Deepika Palagani: ES18BTECH11002

Idea

Updated ER Diagram:

We have updated the ERD we submitted in Assignment 1 according to the data we were provided in the assignment problem statement and the data we newly acquired via scraping and crawling various websites.



Note: ***The modifications we made ensure that all the tables are filled and we have data for all the attributes, including the ones we added using scraping. This ensures that the***

relational database thus created is usable for further analysis directly. We try to create as few tables as possible to improve the efficiency and speed of the queries. This also ensures that the eventual csv's are minimal in size.

Our plan was to first create a table corresponding to the main entity *imdb_entity* in the relational database, so that all available attributes for each entity available on the net could be crawled and populated in this main table.

The *cast_and_crew* table corresponds to the table containing the information of individuals playing different roles in the different imdb entities, and is fed with all information about the individual.

The events entity has a unique identifier for the ID of the event, for example *Oscars* and then we create a table for it.

Because of the way we scraped the data from different sources we created an awards table and then split it into individual awards and entity awards depending on who was awarded the award in question. We chose this approach because we found a collective award and then we created a table for the two respective relationships.

Database design decisions:

- Imdb_entities table:
    - We reduced the multi-valued attribute genres to 3 separate attributes genre1, genre2 and genre3 as each IMDb title has only contains the top 3 genres and thus felt that it would be more efficient to have them as separate attributes than making it a multivalued attribute and making a whole new table for it.
    - We merged the ratings table into this table because it is a one-one relation and we can add ratings as an attribute instead of having a new table for it.
- Cast_and_Crew table
    - We made multi-valued attribute usual_roles to 3 attributes usual_role1, usual_role2, usual_role3 because every person has only top 3 usual roles, so we thought that it would be better to make them as separate attributes than making it a multivalued attribute and making a whole new table for it.

- We made multi-valued attribute known_for to 4 attributes known_for1, known_for2, known_for3 because every title has only top 4 titles they are known for, so we thought that it would be better to make them as separate attributes than making it a multivalued attribute and making a whole new table for it.

List of tables and succinct description for each

Imdb_entities:

Attributes:

- ID:
  - **Description:** The primary ID of the title (primary key)
  - **datatype**: VARCHAR with max length 255
- Primary_title:
  - **Description:** The usual title used for advertising
  - **datatype**: Text
- Original_title:
  - **Description:** The title of the film in original language
  - **datatype**: Text
- Title_type:
  - **Description:** Type of title, for example: short film, documentary, tv series, movies, episode
  - **datatype**: VARCHAR with max length 50
- PG_Rating:
  - **Description:** Says if the title is adult rated or not
  - **datatype**: boolean
- StartYear:
  - **Description:** The release year of a movie/start year of a TV Series
  - **datatype:** INTEGER
- EndYear:
  - **Description:** End year of a TV Series
  - **datatype:** INTEGER
- Runtime:

    - **Description:** Primary runtime of titles in minutes
    - **datatype:** INTEGER
- Genre1:
    - **Description:** Top genre of the movie
    - **datatype**: VARCHAR with max length 255
- Genre2:
    - **Description:** Second top genre of the movie
    - **datatype**: VARCHAR with max length 255
- Genre3:
    - **Description:** Third top genre of the movie
    - **datatype**: VARCHAR with max length 255
- Avg_rating:
    - **Description:** Average rating of the movie based on user ratings
    - **datatype**: FLOAT
- Num_votes:
    - **Description:** Number of people who rated the movie
    - **datatype**: INTEGER
- Budget:
    - **Description:** Budget of the movie/TV series
    - **datatype**: INTEGER
- Revenue:
    - **Description:** Revenue of the movie
    - **datatype**: INTEGER
- Website:
    - **Description:** Refers to the website the entity can be watched on if any
    - **datatype**: Text
- Keywords:
    - **Description:** Keywords in movie/series/episode plot
    - **datatype**: Text
- Plot_outline:
    - **Description:** Overview of movie/TV Series
    - **datatype**: Text
- Production_companies:

    - **Description:** Companies that produces the title
    - **datatype**: Text
- Release_date:
    - **Description:** Release date of the title
    - **datatype**: Text

Cast_and_Crew:

Attributes:

- CastID:
    - **Description:** Unique ID given to each person
    - **datatype**: VARCHAR with max length 255
- Name
    - **Description:** Name of the person
    - **datatype:** VARCHAR(255)
- BirthYear
    - **Description:** Birth year of the person
    - **datatype:** SMALLINT
- DeathYear
    - **Description:** Death year of the person
    - **datatype:** SMALLINT
- Usual_Role1
    - **Description:** One of the top roles the person is known for
    - **datatype:** TEXT
- Usual_Role2
    - **Description:** One of the top roles the person is known for
    - **datatype:** TEXT
- Usual_Role3
    - **Description:** One of the top roles the person is known for
    - **datatype:** TEXT
- Known_for1
    - **Description:** One of the top titles the person is known for (foreign key referencing Imdb_entities)

    - **datatype:** TEXT
- Known_for2
    - **Description:** One of the top titles the person is known for (foreign key referencing Imdb_entities)
    - **datatype:** TEXT
- Known_for3
    - **Description:** One of the top titles the person is known for (foreign key referencing Imdb_entities)
    - **datatype:** TEXT
- Known_for4
    - **Description:** One of the top titles the person is known for (foreign key referencing Imdb_entities)
    - **datatype:** TEXT

Worked_in:

Attributes:

- ID:
    - **Description:** Title ID of the movie (foreign key referencing Imdb_entities)
    - **datatype:** VARCHAR Of max length 255
- CastID:
    - **Description:** ID of the person who acted in the movie (foreign key referencing Cast_and_Crew)
    - **datatype:** VARCHAR Of max length 255
- Role:
    - **Description:** The role the person played in the movie
    - **datatype:** VARCHAR Of max length 255

Other_titles:

Attributes:

- ID:
    - **Description:** Title ID of the original movie (foreign key referencing Imdb_entities)

    - **datatype:** VARCHAR Of max length 255
- Title:
    - **Description:** Title of the other movie
    - **datatype:** TEXT
- Region:
    - **Description:** Region of other movie
    - **datatype:** VARCHAR Of max length 4
- Language:
    - **Description:** Language of other the movie
    - **datatype:** VARCHAR Of max length 4
- Is_original_title:
    - **Description:** It's 1 if it's the original movie
    - **datatype:** BOOLEAN

Characters:

Attributes:

- ID:
    - **Description:** Title ID of the movie (foreign key referencing Imdb_entities)
    - **datatype:** VARCHAR Of max length 255
- CastID:
    - **Description:** Cast ID of the person (foreign key referencing Cast_and_Crew)
    - **datatype:** VARCHAR Of max length 255
- Character_Name:
    - **Description:** Name of the the person's character in the movie
    - **datatype:** Text

Episodes:

Attributes:

- Episode_title_ID:
    - **Description:** Title ID of the episode (foreign key referencing Imdb_entities)
    - **datatype:** VARCHAR Of max length 255
- parent_tv_show_id:

    - **Description:** Title ID of the TV show the episode belongs to (foreign key referencing Imdb_entities)
    - **datatype:** VARCHAR Of max length 255
- Season_number:
    - **Description:** Season number of the episode
    - **datatype:** INTEGER
- Episode_Number:
    - **Description:** Episode number of the episode
    - **datatype:** INTEGER

Individual_awards:

Attributes:

- eventid:
    - **Description:** ID of the event (foreign key referencing events)
    - **datatype:** VARCHAR Of max length 255
- Eventname:
    - **Description:** name of the event
    - **datatype:** Text
- Awardname:
    - **Description:** name of the award
    - **datatype:** Text
- year:
    - **Description:** Year of the event
    - **datatype:** SMALLINT
- Categoryname:
    - **Description:** name of the category
    - **datatype:** Text
- Name:
    - **Description:** The name they used for their career
    - **Datatype:** Text
- Originalname:
    - **Description:** In case people use aliases, we store their actual name here

    - **Datatype:** Text
- Songname:
    - **Description:** name of the song
    - **Datatype:** Text
- Episodename:
    - **Description:** name of the the episode
    - **Datatype:** Text
- Charactername:
    - **Description:** name of the character
    - **Datatype:** Text
- Iswinner:
    - **Description:** if award has been won or not
    - **Datatype:** Boolean
- Isperson:
    - **Description:** is a person
    - **Datatype:** Boolean
- Istitle:
    - **Description:** is a movie, episode or series title
    - **Datatype:** Boolean
- Individual_id:
    - **Description:** id of cast and crew member (foreign key referencing cast_and_crew)
    - **datatype:** VARCHAR Of max length 255

entitity_awards:

Attributes:

- Eventid:
    - **Description:** ID of the event (foreign key referencing events)
    - **Datatype:** VARCHAR of max length 255
- eventname :
    - **Description:** name of the event
    - **Datatype:** Text

- Awardname:
  - **Description:** name of the award
  - **Datatype:** Text
- Year:
  - **Description:** year of the event
  - **Datatype:** SMALLINT
- Categoryname:
  - **Description:** name of the award category
  - **Datatype:** Text
- Name:
  - **Description:** The name they used for their career
  - **Datatype:** Text
- Originalname:
  - **Description:** In case people use aliases, we store their actual name here
  - **Datatype:** Text
- Songname:
  - **Description:** name of the song
  - **Datatype:** Text
- Episodename:
  - **Description:** name of the episode
  - **Datatype:** Text
- Charactername:
  - **Description:** name of the character
  - **Datatype:** Text
- Iswinner:
  - **Description:** if award has been won or not
  - **datatype:** Boolean
- Isperson:
  - **Description:** is a person
  - **datatype:** Boolean
- Istitle:
  - **Description:** is a movie, episode or series title
  - **Datatype:** boolean

- Imdb_id:
    - **Description:** id of imdb entity (foreign key referencing Imdb_entities)
    - **Datatype:** VARCHAR of max length 255

production_companies:

Attributes:

- Id:
    - **Description:** id of the imdb_entity/title (foreign key referencing Imdb_entities)
    - **Datatype:** VARCHAR of max length 255
- Company_name:
    - **Description:** name of the production company
    - **Datatype:** Text

events:

Attributes:

- Eventid:
    - **Description:** Unique ID of an event (primary key)
    - **Datatype:** VARCHAR of max length 255
- Eventname:
    - **Description:** Name of the event
    - **Datatype:** Text

Appendix

Types of modifications to tsv files from IMDb website:

- **title.basics.tsv**: Since there are utmost 3 genres for a movie, we split multivalued attribute genres into 3 attributes Genre1, Genre2, Genre3. In the preprocessing step, we divided the single column genres into 3 column Genre1, Genre2, Genre3.
- **name.basics.tsv**: Since there are at most 3 primary roles for a person and a person is known for at most 4 titles, we split these columns into 3 and 4 columns respectively in preprocessing.

- **title.principals.tsv**: One of the columns (job) here does not contain any useful information, so we dropped the column. And since Characters is a multi-valued attribute, we made a separate tsv file (Characters.tsv) for it which contains, title_id, name_id and character of a person in that movie. Since it's multi-valued, we split each row into multiple rows and made another tsv file Worked_in.tsv which contains the rest of the attributes/columns.
- **title.akas.tsv**: This file contains 2 columns (types, attributes) that do not contain any useful information and are NULL for more than 95% percent of the entries and are not present in our ER diagram, so we removed them.
- We did not modify title.episodes.tsv and title.ratings.tsv and we did not use title.crew.tsv as this information is already present in title.principals.tsv and we thought it would be redundant to use it twice.

Preprocessing the Scraped data

1. Creating reformatted *tsv's* using the provided data

   Please refer to the code file *scraping_preprocessing.py* for the code for dividing a column with multivalued attributes into multiple columns; dividing an entry with multivalued attributes into multiple rows so that each row contains exactly one entry of a multivalued attribute; and for dividing a file into multiple files so that each of them contain only a partial number of columns. Please refer to the comments for more details. It produces the required *.tsv* files which we then copy into our tables in the relational database using the COPY command. Run the code in parts as the code requires a lot of RAM and takes very long to run as the datasets are very large.

2. Awards information scraping and creation of corresponding .csv file using relevant columns

```
import numpy as np
import pandas as pd
import os
import gzip
import shutil

def production_companies(file):
    out_file = file[['imdb_id','production_companies']]
    out_file = out_file.dropna()
    out_file = out_file.assign(production_companies=out_file.production_companies.str.
                split('|')).explode('production_companies').reset_index(drop=True)
    out_file.to_csv('production_companies.tsv',index=False,na_rep=r'\N',sep='\t')

def event_table(file):
    out_file = file[['eventId','eventName']]
    out_file = out_file.dropna()
    out_file = out_file.drop_duplicates(subset = "eventId", keep = 'first')
    out_file.to_csv('events.tsv',index=False,na_rep=r'\N',sep='\t')

data_path = '/content/' #path to  tsv files
events  = pd.read_csv(os.path.join(data_path,'awards.csv'),
        dtype = {'eventId':'str', 'eventName':'str'}, sep=',',na_values='\\N')
events_table(events)
del events

companies  = pd.read_csv(os.path.join(data_path,'imdb-movies.csv'),
    dtype = {'imdb_id':'str', 'production_companies':'str'},
     sep=',',na_values='\\N')
production_companies(companies)
del companies
```

3. Events and Production Companies preprocessing and creation of final *.tsv* files

```
# import pandas with shortcut 'pd'
import pandas as pd

# read_csv function which is used to read the required CSV file
data = pd.read_csv('awards.csv')

# display
print("Original 'input.csv' CSV Data: \n")
print(data.head(10))

# drop function which is used in removing or deleting rows or columns from
the CSV files
data.drop(['occurrence','winAnnouncementTime','nomeneeNote','isPrimary','is
Secondary','isCompany','notes'], inplace=True, axis=1)

# display
print("\nCSV Data after deleting the column 'year':\n")
print(data.head(10))
data.to_csv('awards.csv', index=False)
```