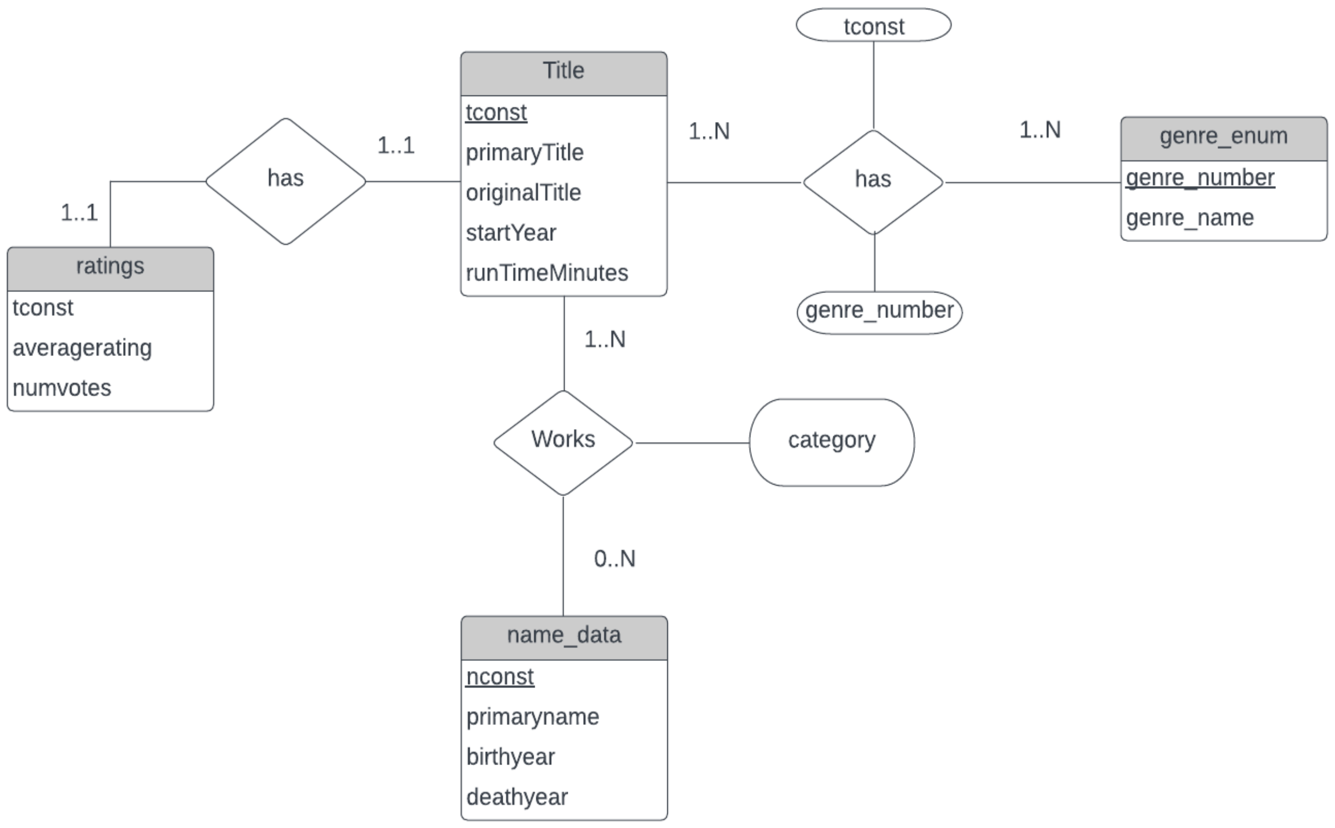
Big Data

Assignment Number 1

Name : Vishal Panchidi

Q1. ER diagram to represent the IMDb data.



Note: ER diagram shown above doesn't match the data files. To keep the ER diagram easy to understand some of the attributes are missing in the ER diagram.

Q2.
Below are the create table SQL commands :

```
Create Table title(
        tconst integer,
        primaryTitle varchar(500),
        originalTitle varchar(500),
        startYear integer,
        runtimeMinutes integer,
        primary key(tconst)
)
```

```
Create Table genre_enum(
        genre_number integer,
        genre_name varchar(500),
        primary key(genre_number)
)

Create Table genre_mapper(
        tconst integer,
        genre_number integer,
        foreign key(tconst)references title(tconst),
        foreign key(genre_number)references genre_enum(genre_number)
)

Create Table resultTitleMapper(
        tconst integer,
        nconst integer,
        category varchar(50),
        foreign key(tconst)references title(tconst),
        foreign key(nconst)references name_data(nconst),
        primary key(tconst,nconst,category)
)

Create Table name_data(
        nconst integer,
        primaryName varchar(500),
        birthYear varchar(4),
        deathYear varchar(4),
        primary key(nconst)
)

Create Table ratings(
        tconst integer,
        averageRating integer,
        numVotes integer,
        foreign key(tconst)references title(tconst)
)
```

Note : All the requirements are taken into consideration while creating the tables. Eg. The primary keys are kept as integers and the string lengths are kept by checking the longest length in the given data. Commands like *titleMapperdf.nconst = titleMapperdf.nconst.str.slice(start=2).astype(int)* are used to remove the two character prefix from all primary and foreign keys.

Q.3 Description of files in IMDB dataset:
Title: This is the file which contains movie titles and the details about the movie. Additionally this file also helps us to know if the movie is an adult movie, what is the release year and end year of the movie, runtime of the movie and the genres the movie is associated with.
Name: One of the primary database where all the names of the personals from the movie industry. Their best work is also mentioned in this file.
Crew.: The crew data helps to know the writers and directors associated with the title of the movie.
Ratings: This data helps us to know how well the movie has performed and what the audience is thinking about the movie. The number of people who gave the rating and the average rating of the movie is present in the data.
Principals: Principals dataset helps to understand the main people associated with the movie title along with the ordering entity which helps to know the priority given to an individual in that particular movie.

Q4)
Complete python is attached as *preprocessing,py*
All load commands are in load_data,.txt
The snippet which removes the Adult movies in the code is given below:
title_basicsDF = title_basicsDF[title_basicsDF['isAdult'] == 0]

The data processing is done in python and pandas dataframes have been achieved according to the desired tables designed. Then these dataframes are saved in CSV files which are then loaded into the tables using postgresql's LOAD command.

The foreign keys issues are solved by inner joining the table data with primary key data set which eventually only keeps the foreign keys which are already present as primary keys in the main table. Then the updated table is loaded in the database which faces no issues.

Primary Code snippet used to solve one foreign key issue:
name_title_mapper = title_data_csv.join(name_title_mapper.set_index("tconst"), on="tconst", how="inner")

Here the *title_data_csv* is left joined on *name_title_mapper_csv* which gives all the *tconst* data that is definitely present in both tables.

name_title_mapper = name_title_mapper[['tconst', 'nconst', 'category']]
Later the excessive columns are removed that got added because of the join operation.

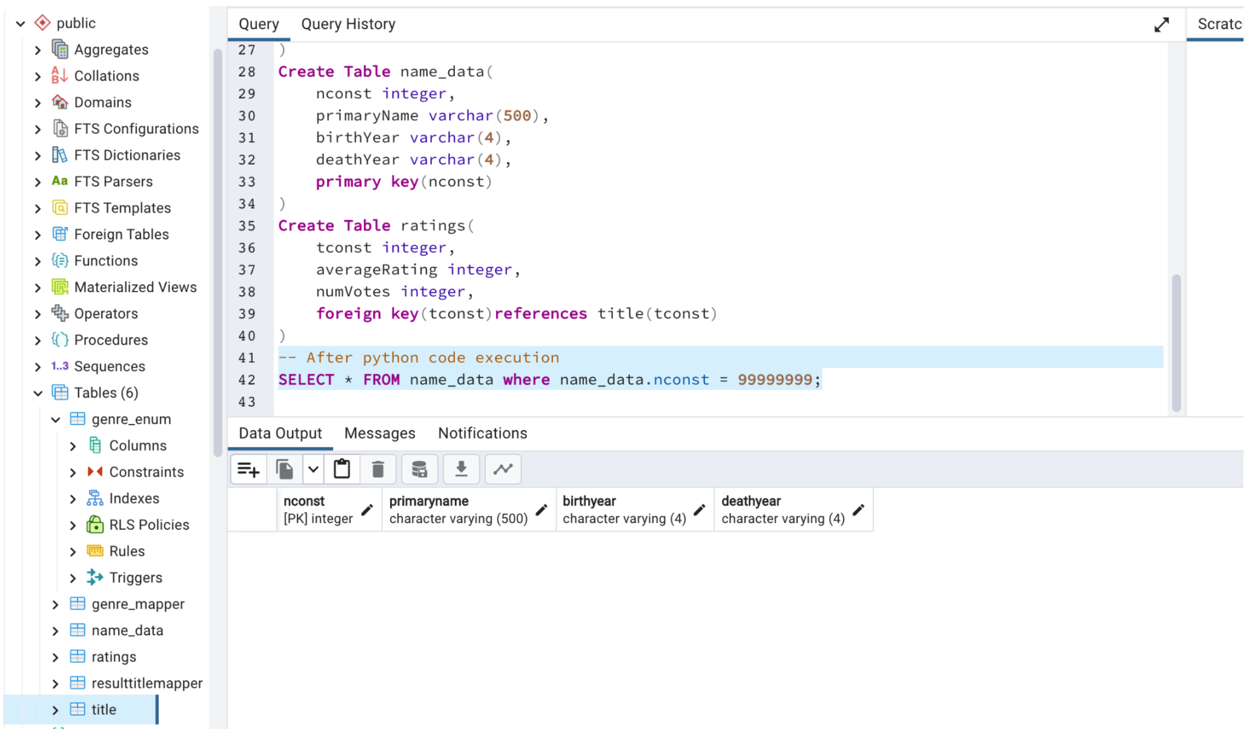
Time taken to load all the processed and clean CSV files in the database was around 20 minutes. Since, the information stored in the data is only focused on actors, actresses, directors, writers and producers and information regarding other job roles have been removed, the duplicates are also removed, the data is loading faster in the database even if it has millions of data

Q5)

Python to database connectivity is done through pyscope2 python library
Complete code is in transaction.py

Error is present in the second line where birth year is 19970.

After the execution of python code which executes the transaction, I am checking the database with the select query if the insert query before the error has executed.



```
)
Create Table name_data(
    nconst integer,
    primaryName varchar(500),
    birthYear varchar(4),
    deathYear varchar(4),
    primary key(nconst)
)
Create Table ratings(
    tconst integer,
    averageRating integer,
    numVotes integer,
    foreign key(tconst)references title(tconst)
)
-- After python code execution
SELECT * FROM name_data where name_data.nconst = 99999999;
```

nconst [PK] integer	primaryname character varying (500)	birthyear character varying (4)	deathyear character varying (4)

As we can see there is no data with the id number.(That is, row #1 should not have been inserted)
That means the transaction code is running successfully.