

DBMS Project

Team : 3 Amigos

March 20, 2023

1 Raw Data Set

We initially began with the raw data set that we obtained from IMDB. The size of this data set is approximately 7 GB. We obtained 7 tables in .tsv files with the following schema description.

title_akas	title_basics	title_crew	title_episode
title_id varchar	tconst varchar	tconst varchar	tconst varchar
ordering int	title_type varchar	directors varchar[]	parent_tconst varchar
title varchar	primary_title varchar	writers varchar[]	season_number int
region varchar	original_title varchar		episode_number int
language varchar	is_adult boolean		
types varchar[]	start_year int		
attributes varchar[]	end_year varchar		
is_original_title boolean	runtime_minutes int		
	genres varchar[]		

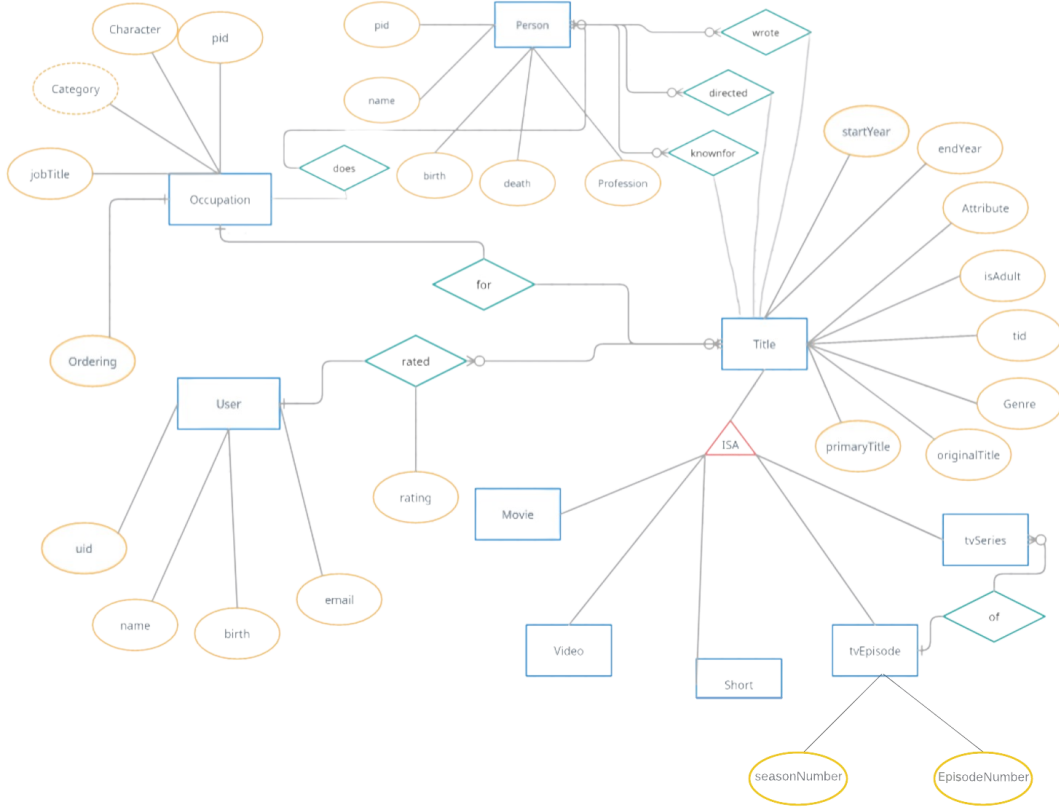
title_principals	title_ratings	name_basics
tconst varchar	tconst varchar	nconst varchar
ordering int	average_rating decimal(3,1)	primary_name varchar
nconst varchar	num_votes int	birth_year varchar
category varchar		death_year varchar
job varchar		primary_profession varchar[]
characters varchar		known_for_titles varchar[]



It is clear that the raw data that we have here has lots of redundancies, unwanted tables, non-atomic data types etc. We first drew an ER diagram for the database that we plan to build for the application. We went on to remove some unwanted data, i.e. tables that we do not require for the application we are meaning to build. We added some new tables for our implementation purpose, like tables containing users and rating information. Finally, we performed FD preserving normalization which has been described in section 3 below.

2 Entity-Relationship Diagram

Here is the entity relationship diagram that we intend to have for our database.



3 FD Preserving Normalisation Steps

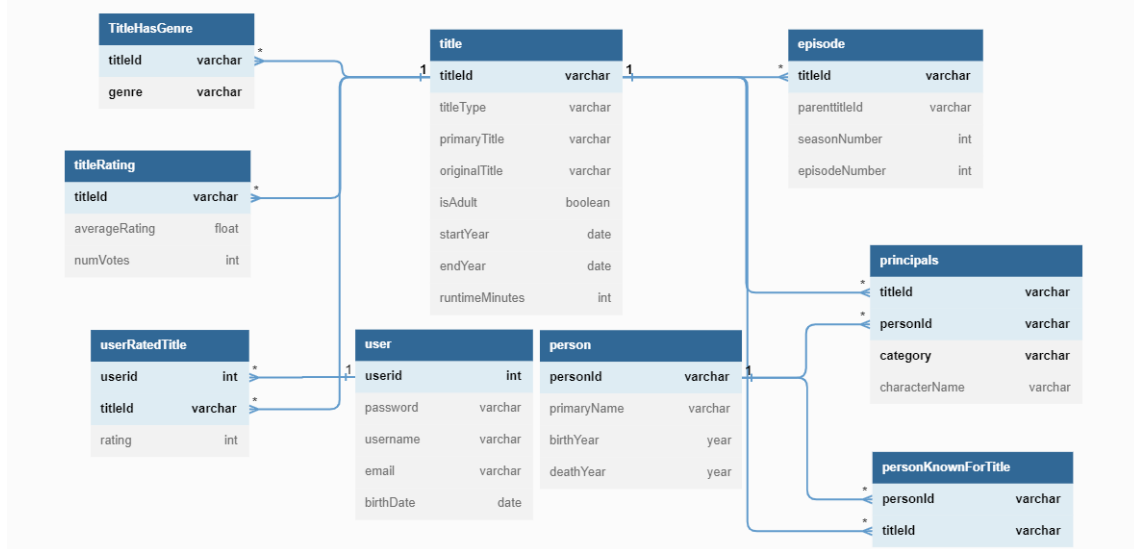
We performed the following normalization steps:

The raw database had multiple data redundancies and non-atomic attributes. To ensure data integrity and coherence in our database design, we used the following normalizations to ensure our database is in Boyce-Codd Normal Form.

- Originally the genre attribute in the 'title' table was a non-atomic varchar array. We normalized 'title' table and separated out the genre attribute to form another relation 'titleHasGenre' which has tuples of the form (title, genre).
- 'knownForTitles' was an array attribute in the original data, which we separated out into 'personKnownForTitle' relation. This relation contains (personId, titleId) tuples.
- We intend to store the ratings given by users to titles in the 'userRatedTitle' relation.

4 Relational Schema

Here is the relational schema for the final database we obtained after doing the normalisation steps. The attributes highlighted in bold are primary keys.



This diagram was made using an online tool DBDiagram.io.

5 Functional Dependencies

Following are the functional dependencies in the tables that we obtained after doing Functional Dependency Preserving Normalization.

5.1 title

$\text{titleid} \rightarrow \text{titleType primaryTitle originalTitle isAdult startYear endYear runtimeMinutes}$

5.2 user

$\text{userid} \rightarrow \text{password username email birthDate}$

5.3 person

$\text{personid} \rightarrow \text{primaryName birthYear deathYear}$

5.4 episode

$\text{titleid} \rightarrow \text{parenttitleid seasonNumber episodeNumber}$

5.5 principals

$\text{titleid personid category} \rightarrow \text{characterName}$

5.6 titleRating

$\text{titleid} \rightarrow \text{averageRating numVotes}$

5.7 userRatedTitle

$\text{userid titleid} \rightarrow \text{rating}$