CITP Portfolio Project Source data

Complex IT Systems – Practice

September 19, 2025

The data provided for the CITP project portfolio includes a subset of the Internet Movie Database IMDb's publicly available dataset, a supplementary dataset collected from the Open Movie Database OMDb, as well as a preliminary inverted word index covering these data. Find description and instructions on how to access and load the data into your own database below.

The IMDb dataset

The IMDB dataset is available in the file `imdb.backup`[1] that you can find on Moodle. It is a subset with around 160000 of the movies in IMDb´s publicly available dataset[2]. To create a database on your local PostgreSQL database server and load the data into that new database, you can simply issue the following two psql commands in your computers command line interface (assuming that your current directory is where the file is located)[3]:

```
psql -U postgres -c "create database imdb"
psql -U postgres -d imdb -f imdb.backup
```

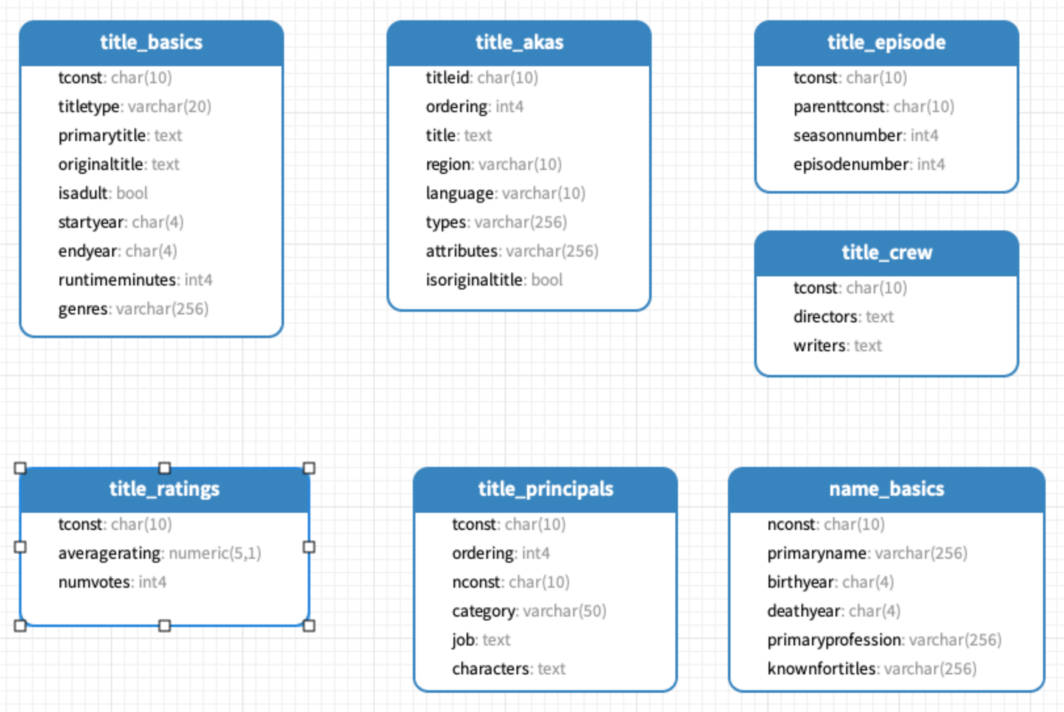
The result of loading the `imdb.backup` data in this way will be a new database, called `imdb` with the 7 tables shown in figure 1. The schema / table structure corresponds closely to the 7 tsv-files (tab-separated-values files) in IMDb's dataset. Observe that a movie (and an episode in a series) is in IMDb's terminology called a title. To prepare, we created a database with 7 tables, loaded data from the 7 tsv-files into these, reduced the content to cover around 160.000 titles and dumped the reduced database to the file `imdb.backup`. The content of the reduced dataset is fine for our purposes and is easier to work with than the full dataset. A database that just adopts the structure of the 7 files as relational database schema is not in good shape. It calls for a thorough redesign. During the reduction of the data, we started with the title_basics and aimed at reducing the other tables correspondingly, so that for instance all titles included in title_ratings are also included in title_basics. However, during this process we did

[1] A database backup file (dump), that can be used as input with psql to create a copy of the database, just as an sql file

[2] https://datasets.imdbws.com

[3] If you alternatively (or in combination) prefer to use the remote server on cit.ruc.dk, you can use the following command *psql -h cit.ruc.dk -p 5432 -U citXX -W -f imdb.backup* (replace citXX with your group cit01, cit02, cit03 . . .). On cit.ruc.dk you don't have privilege to create databases, so you must use your default database, citXX.

not change any column values. So, a second problem is: some columns may contain inconsistent data due to dangling references. You can consider to update your database to remove these.



title_basics
tconst: char(10)
titletype: varchar(20)
primarytitle: text
originaltitle: text
isadult: bool
startyear: char(4)
endyear: char(4)
runtimeminutes: int4
genres: varchar(256)

title_akas
titleid: char(10)
ordering: int4
title: text
region: varchar(10)
language: varchar(10)
types: varchar(256)
attributes: varchar(256)
isoriginaltitle: bool

title_episode
tconst: char(10)
parenttconst: char(10)
seasonnumber: int4
episodenumber: int4

title_crew
tconst: char(10)
directors: text
writers: text

title_ratings
tconst: char(10)
averagerating: numeric(5,1)
numvotes: int4

title_principals
tconst: char(10)
ordering: int4
nconst: char(10)
category: varchar(50)
job: text
characters: text

name_basics
nconst: char(10)
primaryname: varchar(256)
birthyear: char(4)
deathyear: char(4)
primaryprofession: varchar(256)
knownfortitles: varchar(256)

Figure 1: Schemas for the 7 IMDb dataset tables included in imdb.backup

IMDb source table columns

The description of the table columns below is a slightly modified version of the description of the tsv-files at the IMDb download page[4].

title_akas ("also known as" for titles)

- titleId (string) – a tconst, an alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleId
- title (string) – the localized title
- region (string) – the region for this version of the title
- language (string) – the language of the title
- types (strings) – Enumerated set of attributes for this alternative title. One or more of the following: "alternative", "dvd", "festival", "tv", "video", "working", "original", "imdbDisplay".
- attributes (strings) – Additional terms to describe this alternative title, not enumerated
- isOriginalTitle (boolean)

[4] https://developer.imdb.com/non-commercial-datasets/

title_basics (Basic information about titles)

- tconst (string) – alphanumeric unique identifier of the title
- titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)
- primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release
- originalTitle (string) – original title, in the original language
- isAdult (boolean) – 0: non-adult title; 1: adult title
- startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year
- endYear (YYYY) – TV Series end year. Only specified for TV Series title types
- runtimeMinutes (integer) – primary runtime of the title, in minutes
- genres (strings) – includes up to three genres associated with the title

title_crew (Director and writer information for all the titles in IMDb)

- tconst (string) – alphanumeric unique identifier of the title
- directors (strings / nconsts) – director(s) of the given title
- writers (strings / nconsts) – writer(s) of the given title

title_episode (TV episode information)

- tconst (string) – alphanumeric identifier of episode
- parentTconst (string) – alphanumeric identifier of the parent TV Series
- seasonNumber (integer) – season number the episode belongs to
- episodeNumber (integer) – episode number of the tconst in the TV series

title_principals (Casting/crew relating titles and names)

- tconst (string) – alphanumeric unique identifier of the title
- ordering (integer) – a number to uniquely identify rows for a given titleId
- nconst (string) – alphanumeric unique identifier of the name/person
- category (string) – the category of the job that the person was in
- job (string) – the specific job title if applicable
- characters (string) – the name of the character played if applicable

title_ratings (IMDb rating and votes information for titles)

- tconst (string) – alphanumeric unique identifier of the title
- averageRating – weighted average of all the individual user ratings
- numVotes – number of votes the title has received

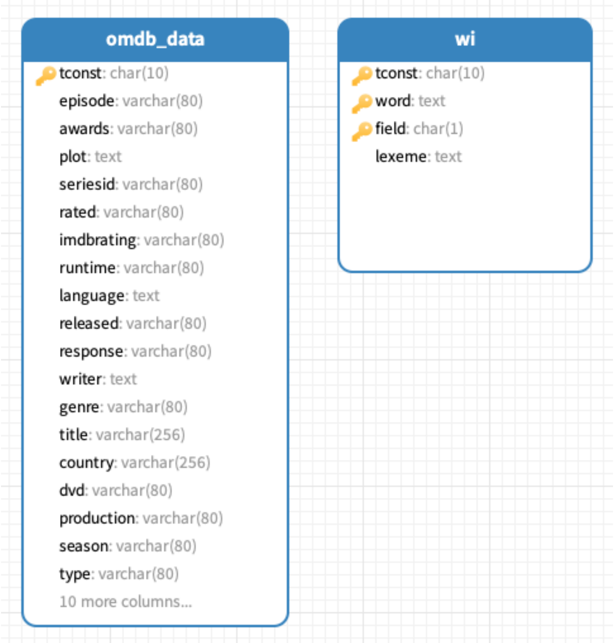
name_basics (Basic info about titles)

- nconst (string) – alphanumeric unique identifier of the name/person
- primaryName (string) – name by which the person is most often credited
- birthYear – in YYYY format

- deathYear – in YYYY format if applicable
- primaryProfession (strings) – the top-3 professions of the person
- knownForTitles (strings/tconsts) – titles the person is known for

The OMDb dataset

IMDb does not include all their data in the public dataset, but some of what they don't disclose can be found elsewhere. We will use supplementary data downloaded from the OMDb API[5]. This data is partly overlapping with what can be found in the IMDb dataset, but some useful unique additions are included. To simplify the inclusion of the OMDb data in your relational database, the data has been transformed and made available as a single table **omdb_data** that can be imported (see figure 2). When building your own database, you can select and include columns of your own preference from this table. Only two are required: poster and plot providing respectively a link to the main poster for the movie and a detailed description of the plot for the title. **Observe**: Don't just make a table in your database that you call **omdb_data**. Consider identifying a suitable modification to your database schema to incorporate the additional data from OMDb.



omdb_data

tconst: char(10)
episode: varchar(80)
awards: varchar(80)
plot: text
seriesid: varchar(80)
rated: varchar(80)
imdbrating: varchar(80)
runtime: varchar(80)
language: text
released: varchar(80)
response: varchar(80)
writer: text
genre: varchar(80)
title: varchar(256)
country: varchar(256)
dvd: varchar(80)
production: varchar(80)
season: varchar(80)
type: varchar(80)
10 more columns...

wi

tconst: char(10)
word: text
field: char(1)
lexeme: text

Figure 2: The source tables available in omdb_data.backup and wi.backup

omdb_data (supplementary data from OMD)

- poster - link to the main poster
- plot – a detailed description of the plot for the title
- . . . and many more.

[5] http://www.omdbapi.com

The data is provided as a PostgreSQL database backup file `omdb_data.backup` and the **omdb_data** table will be imported into your imdb database when you issue the command : `psql -U postgres -d imdb -f omdb_data.backup`

Notice that tconst is a column in **title_basics** as well as in **omdb_data**, so you can use this to combine the data. Many titles in **title_basics** don't have a corresponding row in **omdb_data** (data for these titles is not available in OMDb).

An inverted word index for textual data

To support selected queries on textual data a so-called inverted index is provided. The inverted index is a word index that can be used to lookup a word and retrieve the documents (typically movies) that this word appears in. The word index data is provided in a single table **wi** shown in figure 2 and described below. The indexed columns are primarytitle, plot, characters and primaryname (indicated with the letters 't', 'p', 'c' and 'n' respectively. wi – inverted word index on columns primarytitle, plot, characters and primaryname

wi (word index)

- tconst (string) – alphanumeric unique identifier of the title
- word – the word
- field – the column where the word occurs, values are 't', 'p', 'c' and 'n' that indicates primarytitle, plot, characters and primaryname respectively
- lexeme – a lexeme derived for the word. Will be null for fields 'c' and 'n'.

Most important in this table are the columns tconst and word and it's fine just to use these. The field column can be used if you need to be selective regarding what columns to match when using the inverted index. Maybe you want to remove all the 'c' and 'n' words because you find them disturbing. You could also consider replacing them with "words" that are full primarynames / full character names. The lexeme column can be used to provide a search where different forms of words (inflexions) are harmonized by the use of a separate lexeme index. A lexeme is a string, just like a word, but it provides a kind of normalized form, where different inflexion forms, such as aim, aime, aimed, aiming and aims, normalize to a single lexeme, like aim. To import the **wi** table into your own imdb database, download the file `wi.backup` and run the command : `psql -U postgres -d imdb -f wi.backup`