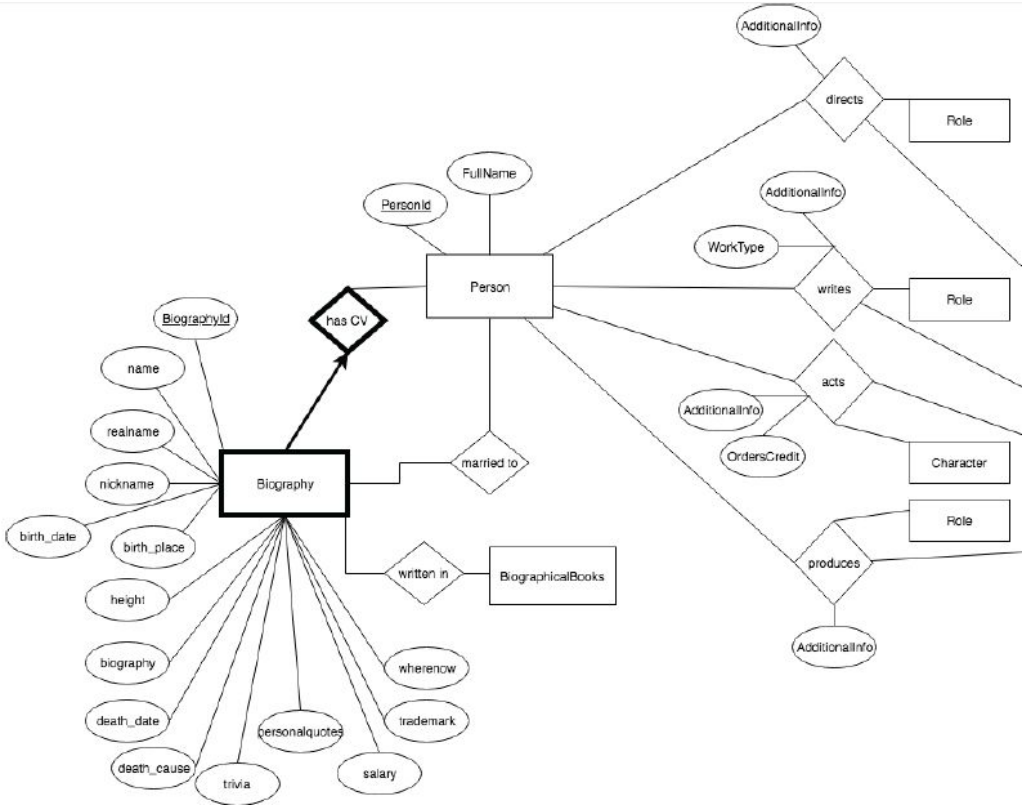
Database Project

Kirusanth Poopalasingam, Florian Reetz, Hyun Jii Cho

IMDB Movies Dataset

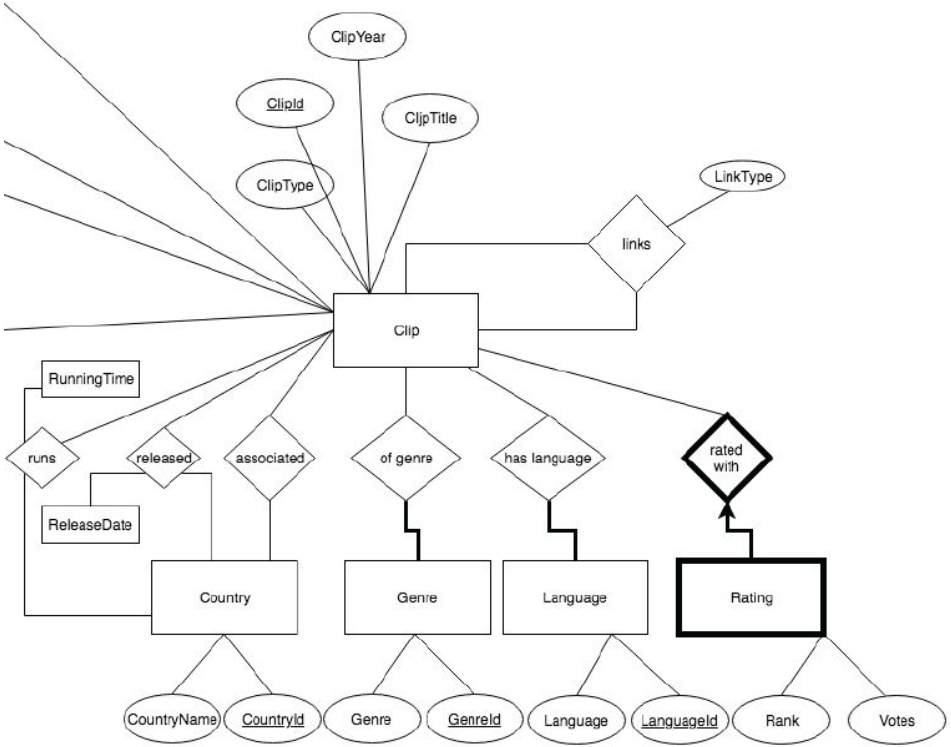
- Internet Movie Database
  - An online database of information on films, tv programs, video games, internet streams, including information about cast, production, biographies, plots summaries, trivia, and fan reviews/ratings
- Goal: Design and create a database
  - E-R Schema to Relational Model
  - Loading data into DBMS
  - Querying the database
  - Creating a user interface which interacts with the database

E-R Schema (Part I)



- Biography is modelled as a weak entity (we cannot have a Biography for a non-existing Person)
- Role and Characters are modelled as entities to create ternary relations between person clip role/character

E-R Schema (Part II)



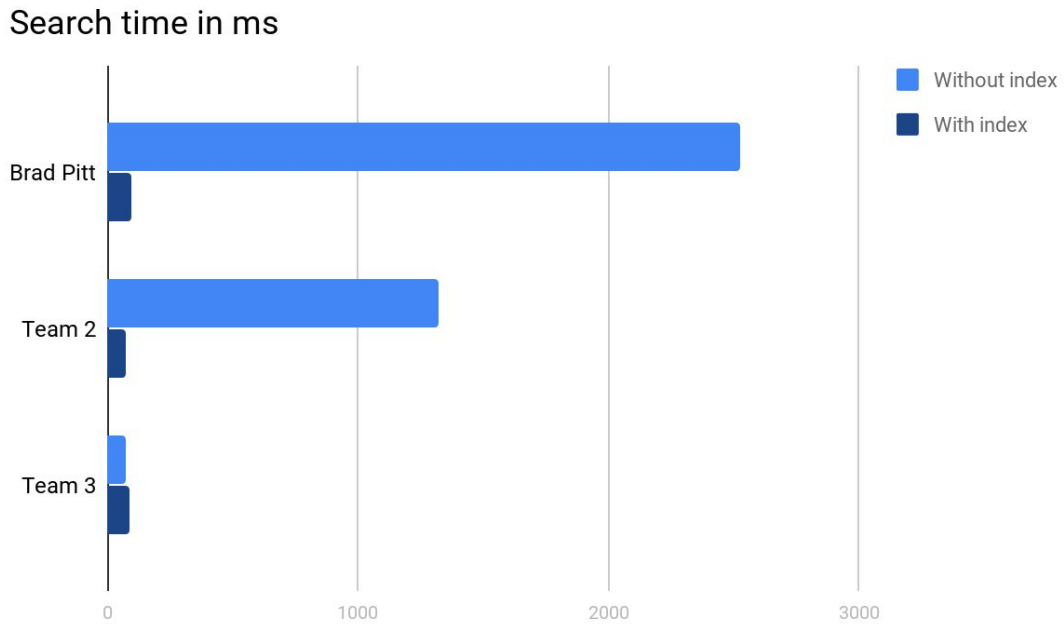
- Rating is modelled as a weak entity since it is only associated to a single Clip
- Each language and each genre is associated with at least one clip (total participation)

Working with the data

- Tried to use Oracle DBMS and SQLite
- We used PostgreSQL as DBMS.
- Data was cleaned and sorted using the python library “pandas”. Most expensive: splitting entries into tuples. The data is exported to CSV and imported into the DB directly using the “COPY FROM” command.
    - Examples:
        - We assume that spouses with no marital status stated are married or unknown, so we mark them as “married/unknown”. When there is no children number stated, we assume that it is “0/unknown”.
        - For languages, we lowercase all languages and remove all parentheses. In addition, we assume that “English”, “English version”, and “English version only” are all clips in English so we clean the data to remove “version” and “only”, for all applicable languages

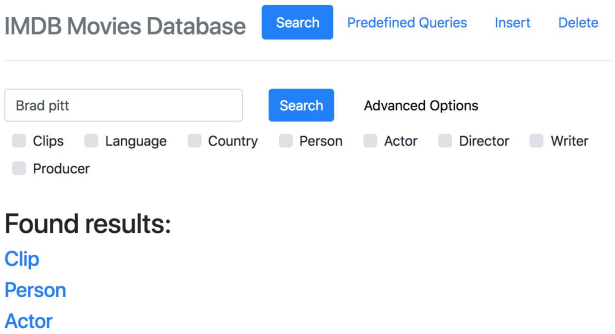
Query optimization

- Added trigram index to improve user interface performance



User Interface

- Bootstrap 4
- Python Flask application
- Simple server generated HTML with some JS
- Minimal interface to show SQL backend



Lessons learned

- Switch database earlier if it’s too slow
- Data analysis before ER-Model makes sense. It helps with possible constraints.
- Plan enough time for data cleaning, search for fast algorithms, write clean data first to CSV and import then
- Correct index can have huge performance impact (>1h => 44s)
- Composite IDs are harder to work with than artificial IDs (for the user interface)

User Interface Demo