|  |  |
| --- | --- |
| **TECHNICKÁ UNIVERZITA V KOŠICIACH**  **FAKULTA ELEKTROTECHNIKY A INFORMATIKY** | |
| **Talianske titulky**  **Dokumentácia** | |
|  | |
| **2022** | **Bc. Oleksandr Kashytskyi Bc. Dmytro Kagirov**  **Bc. Jakub Magáč** |

**Obsah**

[Zoznam obrázkov 3](#_Toc122231563)

[Úvod 4](#_Toc122231564)

[1. Extrahovanie dát 5](#_Toc122231565)

[1.1. Postup získavanie titulkov 5](#_Toc122231566)

[1.2. Postup získavanie filmov 7](#_Toc122231567)

[2. Formátovanie údajov 8](#_Toc122231568)

[2.1. Priebeh formátovanie filmov 8](#_Toc122231569)

[2.2. Priebeh formátovanie titulok 9](#_Toc122231570)

[3. Vytvorenie dátového modelu 11](#_Toc122231571)

[4. Ukladanie dát 16](#_Toc122231572)

[5. Informačný balíčky 17](#_Toc122231573)

[5.1. Štatistika titulkov 17](#_Toc122231574)

[5.2. Štatistika filmov 17](#_Toc122231575)

[6. Star schema 18](#_Toc122231576)

[7. Nahrávanie dát do dátového skladu 22](#_Toc122231577)

[8. Vizualizácia dát 25](#_Toc122231578)

Zoznam obrázkov

[Obr. 1 Testovaci sequel prikaz pre databazu 4](#_Toc26632757)  
[Obr. 2 Prevod csv do formátu JSON 8](#_Toc26632757)  
[Obr. 3 Odstránenie nepotrebných filmov](#_Toc26632757) 9  
[Obr. 4 Funkcia rozbalenia archívu 9](#_Toc26632757)  
[Obr. 5 Konverzia .srt v .json](#_Toc26632757) 10  
[Obr. 6 Spojenie všetkých titulok 10](#_Toc26632757)  
[Obr. 7 ER diagram databázy](#_Toc26632757) 11  
[Obr. 8 Logický relačný diagram databázy](#_Toc26632757) 12

[Obr. 9 Vytváranie klasu GenreModel a inicializacnej funkcie](#_Toc26632757) 15

[Obr. 10 Nahrávanie tituliek do databazy](#_Toc26632757) 16

[Obr. 11](#_Toc26632757)  Star schema18

[Obr. 12](#_Toc26632757)  Select pre CS skupinu22

[Obr. 13](#_Toc26632757)  Zlučovanie údajov zo všetkých skupín pre tabuľku movie\_fact22

[Obr. 14](#_Toc26632757)  Zlučovanie údajov zo všetkých skupín pre tabuľku subtitle\_fact23

[Obr. 15](#_Toc26632757)  Nastavenie správneho ID a odstránenie duplikátov23

[Obr. 16](#_Toc26632757)  Implementácia star schémy v Power BI24

[Obr. 17](#_Toc26632757)  Graf priemerného hodnotenia podľa žánru25

Obr. 18 Graf priemerného počtu hlasov podľa žánrov25

Obr. 19 Graf rozpočtu podľa filmu26

Obr. 20 Rozpočet podľa žánru27

Obr. 21 Graf počtu filmov podľa rokov28

Obr. 22 Počet filmov podľa pohlavia režiséra28

Obr. 23 Priemerný počet slov v titulkoch podľa jazyka29

Obr. 24 Priemerný počet slov v titulkoch podľa scenáristu29

Obr. 25 Priemerný rozpočet filmu podľa krajiny30

Obr. 26 Priemerný počet hlasov podľa režiséra30

Úvod

Témou nášho zadania je vytvorenie filmovej databázy s titulkami. Hlavným cieľom je vytvoriť databázový model filmov s titulkami a naplniť ho relevantnými údajmi. Počet filmov nemôže byť nižší ako 1000.

Aktuálny stav prvej časti projektu je zverejnení na gite za linkom:

* <https://github.com/kot-1999/tass_proj>

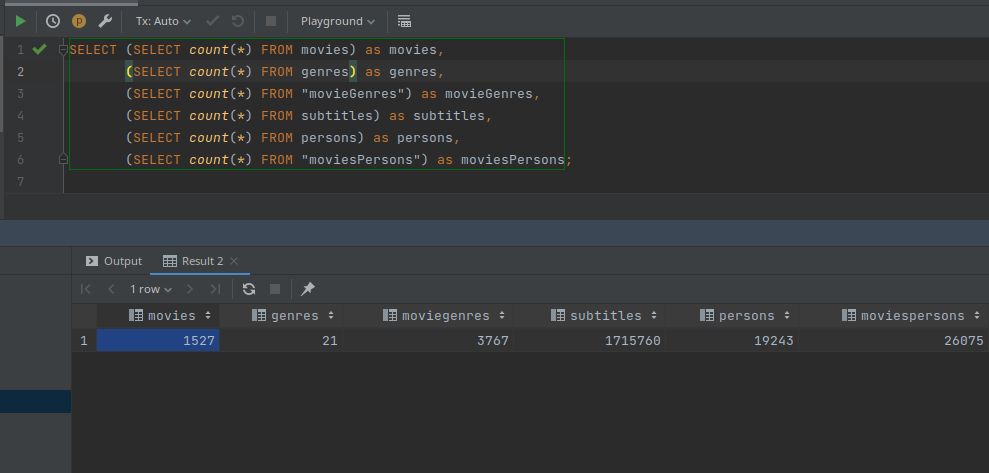
Aktuálny stav druhej časti projektu je zverejnení na gite za linkom:

* <https://github.com/Dmytro27Ind/movies-star-schema>

Samotný backup databázy je ulozeny na google drive a pristupit sa da za nasledujúcim odkazom:

* <https://drive.google.com/file/d/1NbSQsVZDgRwqz3KcG4jCoKcEReaZlpWW/view?usp=sharing>

Pred použitím backup suboru je potrebné vytvoriť prázdnu databázu a spraviť **restore** do nej. Ak restor databázy bol spraveny úspešne, tak uvedený sequel bude mať nasledovný výsledok obr. 1



Obr. 1 Testovaci sequel prikaz pre databazu

1. Extrahovanie dát
   1. Postup získavanie titulkov

Pre zozbieranie titulkov sme použili webovú stránku <https://yts-subs.com/>, ktorá slúži na stiahnutie tituliek. Skript pre zber tituliek sme písali v jazyky Python a pre web scraping bola použitá knižnica Beautiful Soup. Pre posielanie HTTP requestov je použitá knižnica *requests*.

**Zdrojový kód:**

*from* operator *import* sub

*from* bs4 *import* BeautifulSoup

*import* requests, wget, re, time, csv

*import* subprocess

def runcmd(*cmd*, *verbose* = False, \**args*, \*\**kwargs*):

    process = subprocess.Popen(

        cmd,

*stdout* = subprocess.PIPE,

*stderr* = subprocess.PIPE,

*text* = True,

*shell* = True

    )

    std\_out, std\_err = process.communicate()

*if* verbose:

        print(std\_out.strip(), std\_err)

*pass*

*for* page *in* range(123):

*if* page == 0:

        movies\_list\_url = 'https://yts-subs.com/language/italian'.format(page)

*else*:

        movies\_list\_url = 'https://yts-subs.com/language/italian?page={}'.format(page)

    html\_text = requests.get(movies\_list\_url).content

    soup = BeautifulSoup(html\_text.decode('utf-8','ignore'), *features*="lxml", *from\_encoding*="utf-8")

    media\_body = soup.findAll("li", {"class":"media-movie-clickable"})

*for* mb *in* media\_body:

        href = mb.find("a", *href*=re.compile("/movie-imdb/tt"))['href']

        ttf = href.split('/')[2]

        movies\_url = 'https://yts-subs.com{}'.format(href)

        html\_text\_sub = requests.get(movies\_url).content

        soup\_sub = BeautifulSoup(html\_text\_sub.decode('utf-8','ignore'), *features*="lxml", \_ *from\_encoding*="utf-8")

        sub\_href\_el = soup\_sub.findAll("span", *text* = re.compile('Italian'))[0].parent.find\_next('td')

        sub\_url = sub\_href\_el.find("a")['href']

        download = sub\_url.split('/')[2]

        runcmd("wget --output-document={1}.zip --directory-prefix=download\_folder \_ \_ \_ \_ \_\_ \_\_ \_\_ \_\_ \_ https://yifysubtitles.org/subtitle/{0}.zip".format(download, ttf), *verbose* = False)

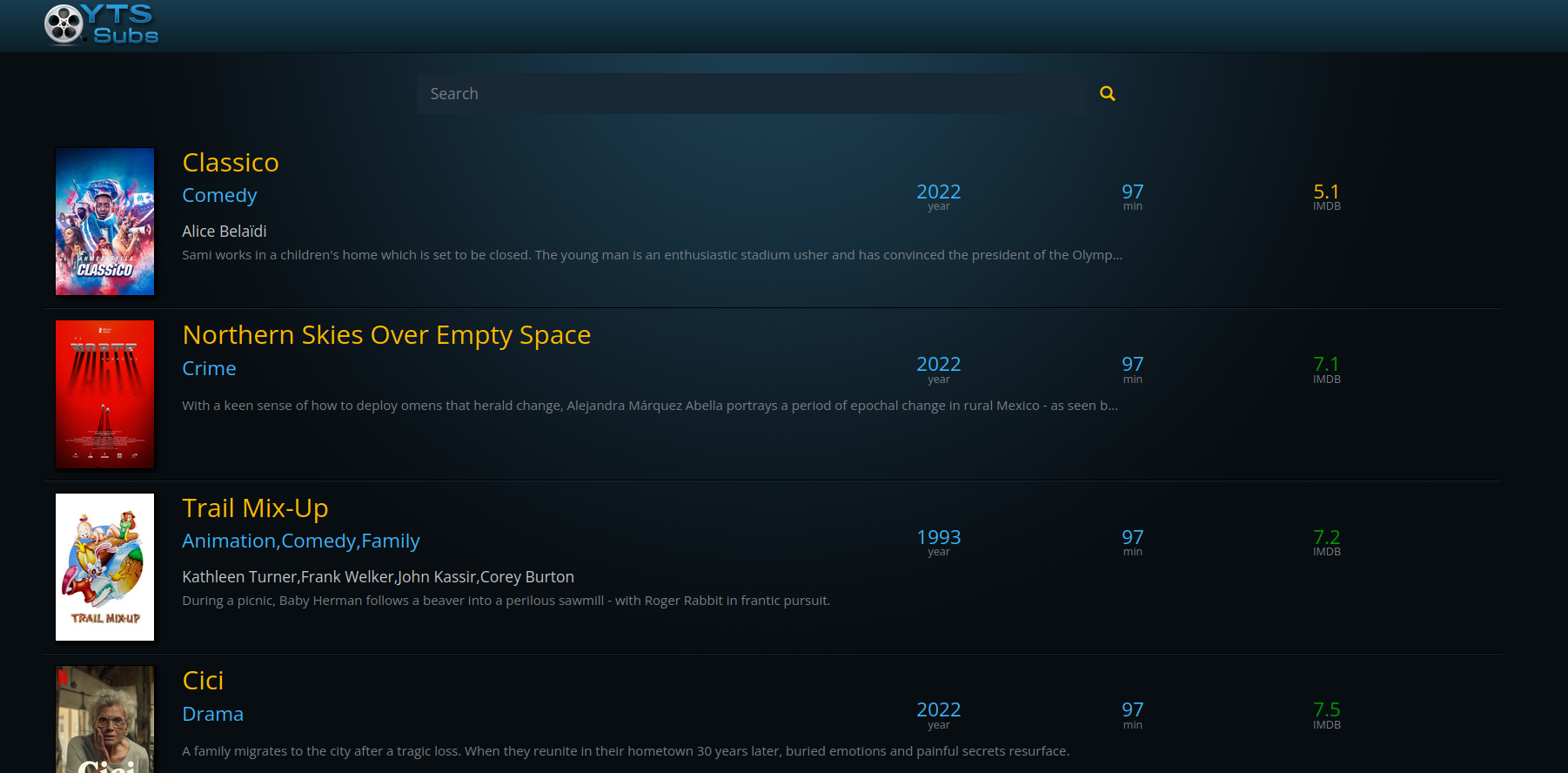
**Použité funkcie:  
runcmd():** Funkcia slúži pre spúšťanie systémovývh príkazov z Pythonovského kódu. Použili sme ju pre spustenie programu wget, ktorý je použitý pre sťahovanie zip súborov, obasahujúcich titulky.

**Beautiful Soup:**  
Knižnica Beautiful soup je knižnica slúžiaca pre vyťahovanie dát z HTML a XML súborov.

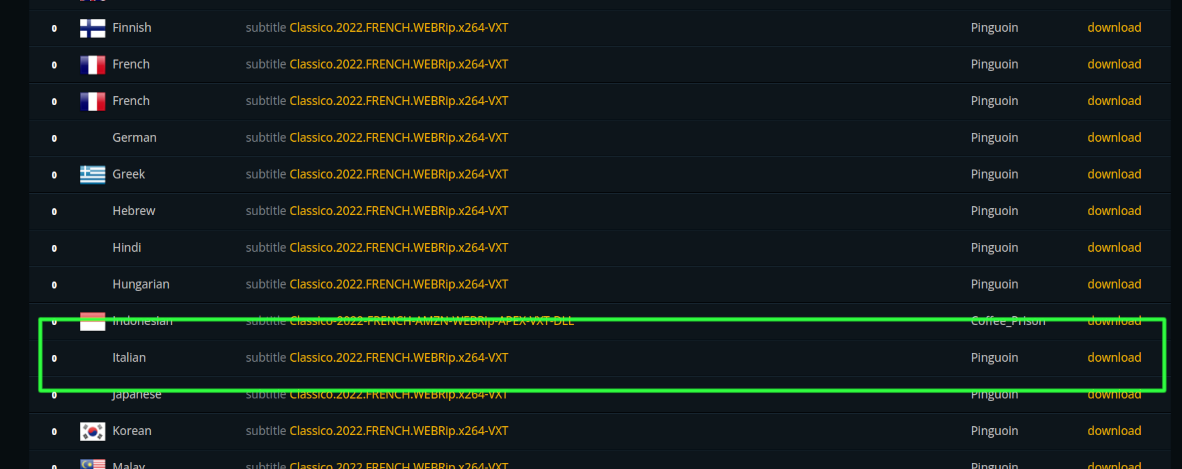
**funkcie Beautiful Soup**.findAll() - vyhľadá všetky zadefinové dáta z HTML .find() - nájde prvý prvok, ktorý zodpovedá prvému prvku.find\_next() - nájde další element, ktorý je poslaný ako parameter

**Spôsob sťahovania titulkov:**

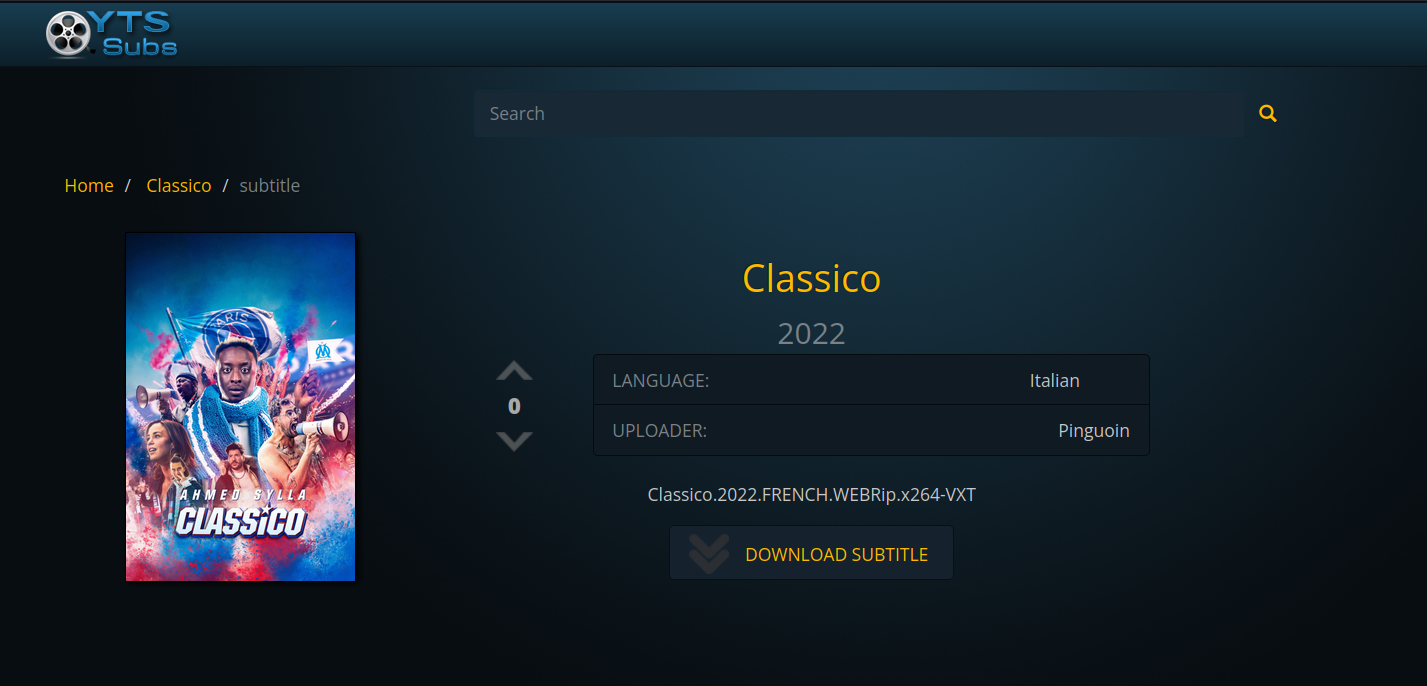
1. Prechádzame linky zo zoznamu, <https://yts-subs.com/language/italian>. Filmy sú vyfiltorvané, zobrazuju sa len filmy, ktoré majú talianské titulky.



1. Skrip prejde na podstránku filmu a vyťiahne z podstránky link kde je prístupný link pre stiahnutie titulky.

1. Po prejdení na podstránku, ktorej link sme získali z predchádzajúceho kroku, vyscrapjueme link kde je prístupný zip súbor. Tento link je potom poslaný do funkcie *runcmd()*, kde sa použije program wget pre stiahnutie. Ako ďalší argument sa pošle do funkcie imdb\_id, ktoré sme získali z parametru URL adresy. IDčko je tatkiež použité ako názov, sťahovaného ZIP súboru.



* 1. Postup získavanie filmov

Zakladnym zdrojom pre filmy bola datova sada z vebovej stranky [Data World](https://data.world/mahe432/movies). Bola stiahnutá datova sada “**IMDb movies.csv**“. Dana datova sada obsahuje 22 stĺpce a 81 273 riadkov. Celkovo bolo pouzito 1527 filmov ktorý mali titulky.

1. Formátovanie údajov

Po stiahnutí súboru údajov s 80 000+ filmov a ~2500 archívov s titulkami, bolo potrebne tieto údaje naformátovať, aby sme ich následne mohli nahrať do našej databázy. Na spustenie všetkých skriptov na formátovanie a spracovanie dát je možne použiť príkaz:

$ node movies\_to\_json.js && node generate\_id\_array.js && node extract\_subtitles.js && node subtitles\_to\_json.js && node all\_subtitles\_json.js && node all\_movies\_with\_sub.js

Balíčky, ktoré boli použité:

* convert-csv-to-json - <https://www.npmjs.com/package/convert-csv-to-json>
* extract-zip - <https://www.npmjs.com/package/extract-zip>
* parse-srt - <https://www.npmjs.com/package/parse-srt>
  1. Priebeh formátovanie filmov

Je potrebne previesť súbor s filmami IMDb\_movies.csv do formátu JSON ale na to potrebné zmeniť oddeľovač, ktorý bol použitý v tomto súbore, na iný. Preto sme zmenili **,** na **|** pomocou online služby <https://wtools.io/change-csv-column-separator>. Tento krok bol potrebný, pretože v pôvodnom súbore IMDb\_movies.csv sa čiarka používa nielen na oddelenie stĺpcov, ale aj na oddelenie prvkov v poli.

Skript movies\_to\_json.js konvertuje filmý do formátu JSON a uloží ich do súboru IMDb\_movies.json. Na prevod csv do formátu JSON sa použiva balík convert-csv-to-json. Kód skriptu je možné vidieť na obrázku 2.

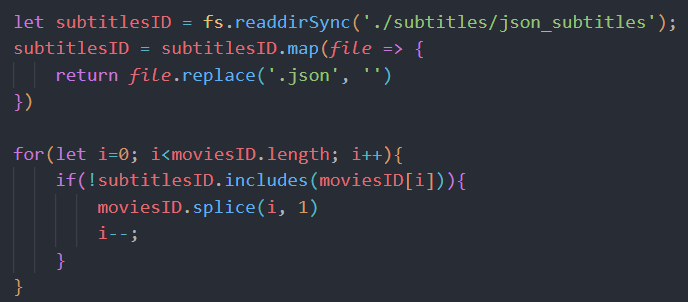
Obr. 2 Prevod csv do formátu JSON



Pomocou skriptu generate\_id\_array.js, sa všetky id. čísla filmov uložia do samostatného súboru, aby s nimi bolo ľahšie pracovať.

Po naformátovaní všetkých tituliek je možne odfiltrovať filmy a nechať len tie, ku ktorým máme titulky, preto že nemá zmysel nahrávať do databázy film, ku ktorému nie sú titulky. Dá sa to urobiť pomocou skriptu all\_movies\_with\_sub.js, ktorý vytvorí súbor all\_movies.json so všetkými filmami, ktoré majú titulky. Tento skript najprv prečíta priečinok, v ktorom sa nachádzajú titulky, a uloží ich id. čísla v premennej subtitlesID. Súbory tituliek, ktoré sa nachádzajú v tomto priečinku, majú názvy, ktoré obsahujú ich id. čísla, preto stačí len prečítať názvy súborov, ktoré sú v tomto priečinku. Ďalej z poľa moviesID odstránia tie ID, ktoré nie je v subtitlesID. ID. čísla tituliek sú rovnaké ako pri filmoch, ku ktorým tieto titulky patria, preto týmto spôsobom je možne odfiltrovať tie filmy, pre ktoré existuje titulky. Ako je to implementované v kóde je možne vidieť na obrázku 3.

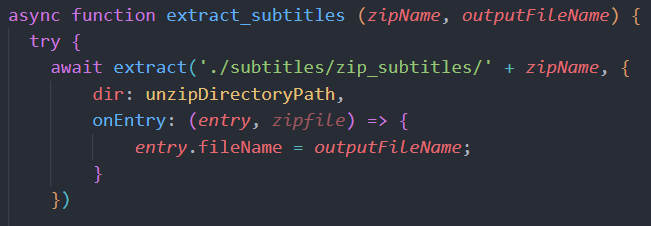
Obr. 3 Odstránenie nepotrebných filmov



* 1. Priebeh formátovanie titulok

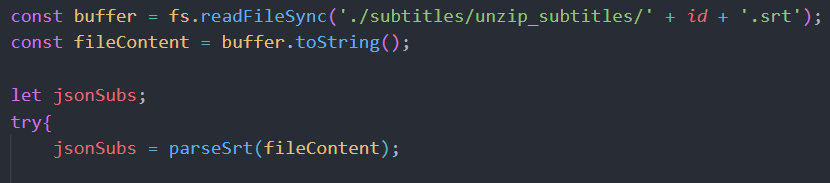
Keďže všetky titulky sú v samostatných archívoch, na rozbalenie bol použitý balík extract-zip pomocou ktorého sme rozbalili každý archív a nastavili nový názov pre titulky. Názov titulok je id číslo filmu, ku ktorému patria tieto titulky. Na obrázku 4 je možne vidieť funkciu extract\_subtitles (zipName, outputFileName), ktorá berie ako vstup názov archívu a názov, ktorý bude priradený titulkom (v našom prípade je to id číslo filmu).

Obr. 4 Funkcia rozbalenia archívu



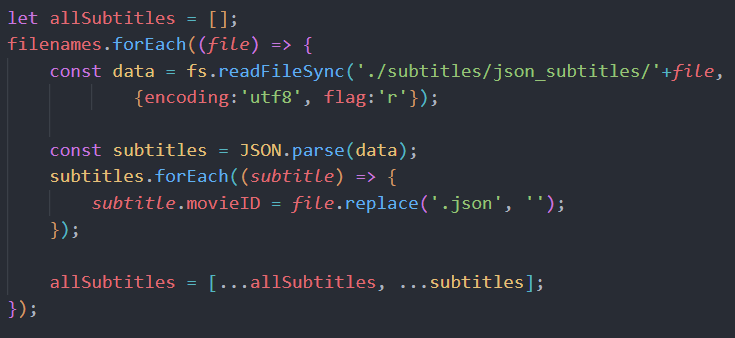
Na konverziu .srt súboru, ktorý obsahuje titulky, do formátu JSON, je potrebné použiť skript subtitles\_to\_json.js. Tento skript prečíta priečinok, v ktorom sú umiestnené všetky titulky, a prevedie ich do JSON formatu pomocou balíka parse-srt. Na obrázku 5 je možné vidieť, ako prebieha konverzia pomocou funkcie parseSrt().

Obr. 5 Konverzia .srt v .json



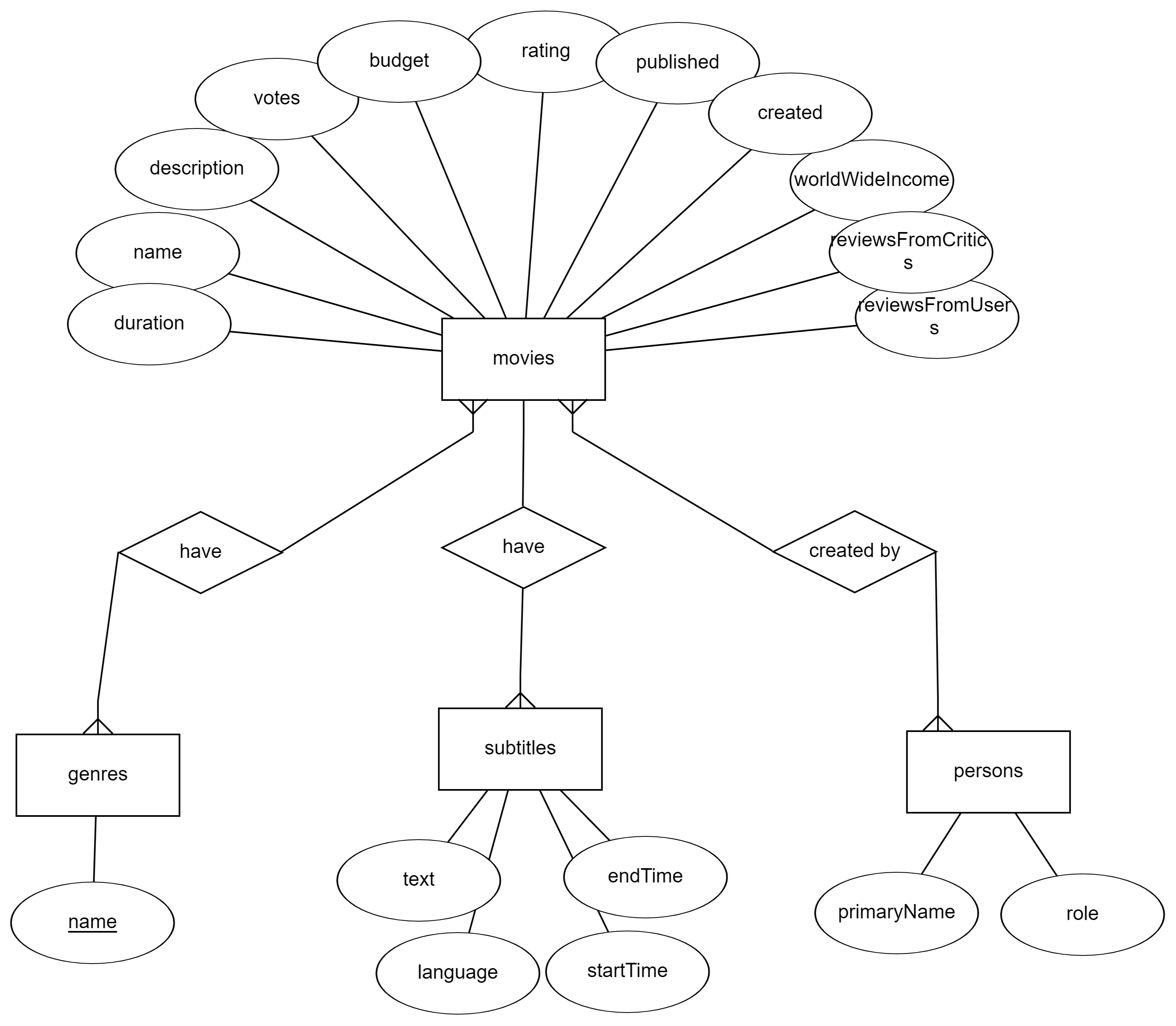
Ďalej pre pridanie id. čísla filma do tituliek a zoskupenie všetkých tituliek do jedného JSON súboru treba použiť skript all\_subtitles\_json.js. Tento skript prečíta každý JSON súbor s titulkami, vezme id číslo filmu z názvu tohto súboru a pridá k objektu, v ktorom sa nachádzajú titulky. Potom tieto titulky zlúči so všetkými predchádzajúcimi titulkami, aby potom bolo možné toto pole objektov uložiť ako jeden JSON súbor so všetkými titulkami. Implementáciu kódu je možne vidieť na obrázku 6.

Obr. 6 Spojenie všetkých titulok



1. Vytvorenie dátového modelu

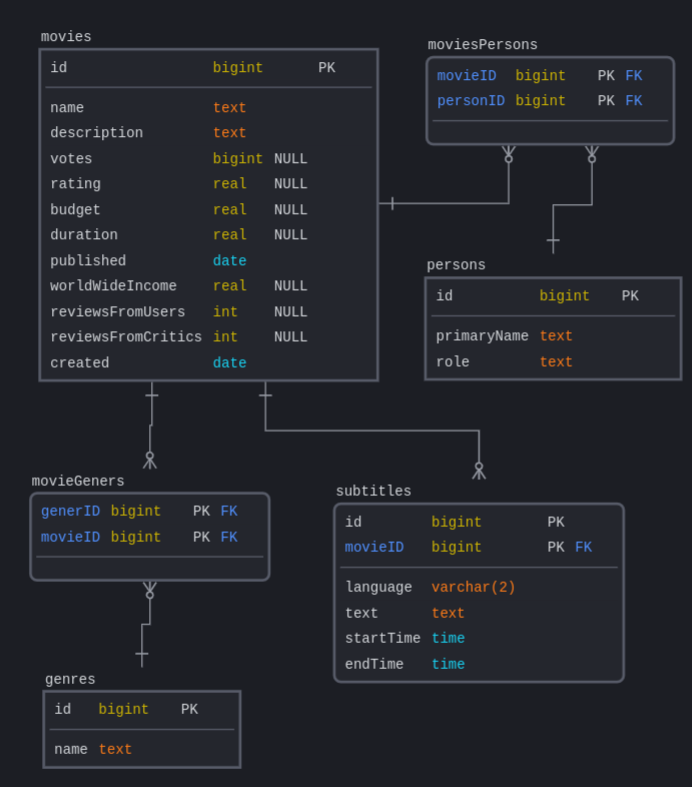
Na obrázku 7 je možne vidieť ER diagram našej databázy.



Obr. 7 ER diagram databázy

V tomto diagrame je možne vidieť 4 entity, ako sú movies, genres, subtitles a persons.

* Vzťah medzi filmami a žánrami – **many-to-many**, pretože film môže mať veľa žánrov a rovnaký žáner môže byť v mnohých filmoch.
* Vzťah medzi filmom a titulkami – **one-to-many**, pretože v jednom filme môže byť veľa titulkov. V entite subtitles sú samostatné repliky.
* Vzťah medzi filmami a osobami, ktoré na tomto filme pracovali – **many-to-many,** pretože na jednom filme môže pracovať veľa ľudí a jeden človek môže pracovať na rôznych filmoch. V entite persons sú herci, režiséri a scenáristi.



Obr. 8 Logický relačný diagram databázy

Na obrázku 8 môžme vidieť logický relačný diagram navrhnutej databázy. Diagram bol vytvorený pomocou online nástroja [SqlDBM](https://app.sqldbm.com/). Obsahuje 4 obyčajné dátové tabuľky a 2 spojovacie. Okrem toho sú tam všetky atribúty, ktoré tabuľky obsahujú a ich typy. Tiež je označené či daný atribút môže nadobúdať NULL hodnotu. Šípky sú typu Crow’s Foot Notation a označujú vzťahy medzi tabuľkami. V danom modeli sú použité vstahy **one-to-many** a **many-to-many**.

movies - je základnou tabuľkou, ktorá obsahuje väčšinu údajov o filme a ma nasledovné atribúty:

* id - unikátny identifikator tabuľky;
* name - názov samotného filmu;
* description - popis ku filmu;
* votes - počet ľudí, ktorí sa zúčastnili hodnotenia filmu;
* rating - samotné hodnotenie filmu, kde 10 je najlepšia známka, 0 je najhoršia;
* budget - reprezentuje cenu natáčania filmu v USD;
* duration - trvalosť filmu v minútach;
* published - rok publikácie filmu;
* worldWideIncome - celosvetový zisk;
* reviewsFromUsers - počet hodnotení od obyčajných používateľov;
* reviewsFromCritics - počet hodnotení od kritikov;
* created - rok vytvarania filmu.

subtitles - tabuľka s titulkami. Ma 2 primárne kľúče id a movieID, ktoré spolu tvoria jeden unikátny kompozitny kľúč.

* id - poradový identifikator tituliek. Prezentuje postupnosť výskytu titulkov vo filme;
* movieID - označuje film ku ktoremu patria titulky;
* language - jazyk v titulkoch;
* text - samotny text tituliek;
* startTime - čas začiatku titulkov;
* endTime - čas ukončenia titulkov.

genres - tabuľka obsahuje žánry filmov, ktoré sa vyskytujú v dátovej sade a obsahuje:

* id – unikátny identifikator tabulky;
* name - nazov samotného žánru;

moviesGeners - spojovacia tabuľka, ktorá spája tabuľky movies a genres tak, že jeden film môže mať viacero žánrov. Obsahuje iba kompozitný kľúč:

* movieID - identifikator tabuľky movies;
* generID - identifikator tabuľky genres.

persons - tabuľka, ktorá obsahuje zoznam ľudí, ktorí brali účasť vo vývoji filmov z použitej dátovej sady. Obsahuje kompozitny kľúč, ktorý pozostáva z atribútov id a role:

* id - unikatny identifikator tabuľky;
* role - definuje rolu človeka a môže nadobúdať nasledujúce hodnoty: DIRECTOR, SCENARIST a ACTOR
* primaryName - preferované meno človeka (meno + priezvisko).

moviesPersons - spojovacia tabuľka, ktorá spája tabuľky movies a persons. Obsahuje iba kompozitny kľúč:

* movieID - identifikator tabuľky movies;
* personID - identifikator tabuľky persons.

Pre vytvorenie databázy boli použité nasledujúce technológie:

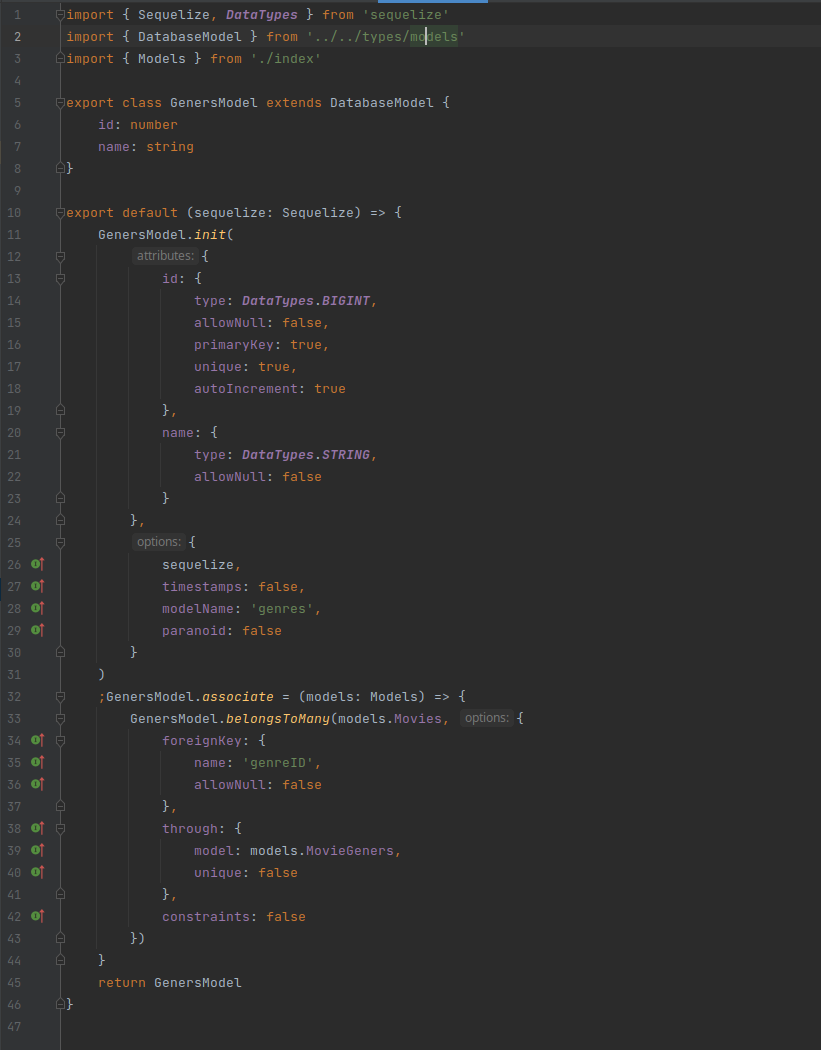
* **Docker** a **Docker-Compose** - tento nástroj bol použitý na vytvorenie postgres servera na lokalnom virtuálnom počítači. Všetky zmeny a aktuálny stav servera sa ukladá do priečinku ./docker root kategórie projektu;
* **NodeJS** - back-endové prostredie JavaScript poslúžilo na vytváranie aplikácie na strane servera;
* **Sequelize** - je moderný TypeScript a Node.js ORM ktorý umožní vytvárať, napĺňať a komunikovať s databázou;
* **TypeScript** - bezplatný a otvorený programovací jazyk vyvinutý a udržiavaný spoločnosťou Microsoft. Bol použití ako základný jazyk pre prácu s databázou. Jediným rozdielom od JavaScript je typizácia premenných.
* ts-config - určuje koreňové súbory a možnosti kompilátora potrebné na kompiláciu projektu a jeho časti.

Databáza sa automaticky vytvára pri spustení backendoveho servera pomocou príkazu npm run debug*.* Pred spustením je potrebné spustiť virtuálny stroj s postgres serverom pomocou príkazu docker-compose up*.* Detailnejší popis použitia je na GitHub repozitari projektu.

Pomocou použitých nástrojov, proces vytvárania databázových modelov bol nasledovný:

* Vytvoriť inštanciu objektu sequelize s pripojením na postgres;
* Rozšírením triedy DatabaseModel vytvoriť modely pre každú tabuľku obsahujúce všetky atribúty.
* Vytvoriť funkcie inicializácie pre každý model, kde sa zadefinuju typy a asociácie;
* Inicializovat všetky modely do objektu modelsa exportovať ho*.*

Tento proces je možné vidieť na obrázku 9.



Obr. 9 Vytváranie klasu GenreModel a inicializacnej funkcie

1. Ukladanie dát

Pre ukládanie udajov bol použitý seed script. Ten sa spúšťa pomocou príkazu npm run seed*.* Spustenie potrebuje len raz. Pre prácu scriptu je potrebné spustit virtualny stroj s postgres serverom pomocou prikazu docker-compose up*.*

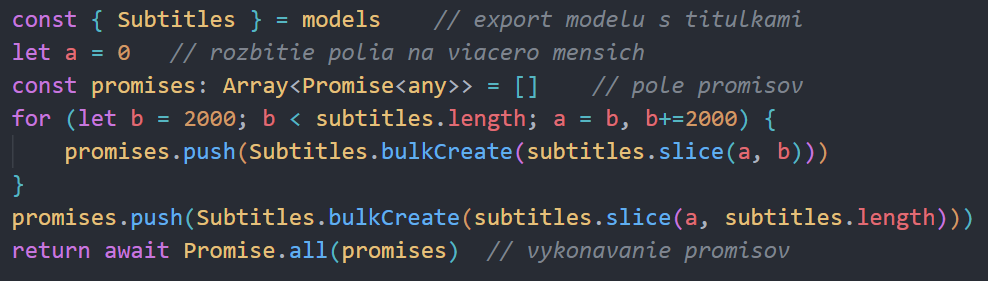
Pre seedy na kompilaciu sa pouziva ts-config.database.json subor. On kompiluje postupne každý subor ktorý sa nachádza v ./src/db/seeders . Aby sa vyhol chybám počas nahrávania dát, seed každej tabuľky prebieha zvlášť v určitom súbore podľa poradia zadefinovaného v prvých dvoch písmenách názvu súborov (napr. 03-filename.ts).

Subory 01-file.ts az 06-file.ts nahrávajú dáta priamo do databázy. Predspracovanie dát a ich konvertácia to tvaru vhodného na nahratie do databázy prebieha v súbore 00-convertData.ts. Okrem toho, počas nahrávania dát bolo zistené že niektoré filmy majú duplicitné údaje, napr. doch rovnakých režisérov alebo hercov. Odstránenie duplicity tiež prebieha v tomto súbore. Na konci exportujeme polia objektov (code 4.1) vhodné pre použitie metodou .bulkCreate od sequelize.

export { movies, subtitles, genres, movieGenres, persons, moviesPersons }

code 4.1 eksport poli objektov na ukladanie do databázy

Metoda .bulkCreate naraz postupne nahráva všetky objekty z polí ako riadky do databázy. Jediným problemom bolo nahrať všetky titulky, kedže ich je viac ako milión. To sa podarilo vyriešiť tak, že celé pole s titulkami sa rozbilo na viacero menších s dĺžkou max 2000 objektov a asynchrónne nahrávanie pomocou Promise.all() urýchlilo nahrávanie dát do databázy (Obr. 10). Takýto prístup skrátil nahrávanie dát do databázy s niekoľkých desiatkov minut do jednej minúty.



Obr. 10 Nahrávanie tituliek do databazy

1. Informačný balíčky
   1. Štatistika titulkov

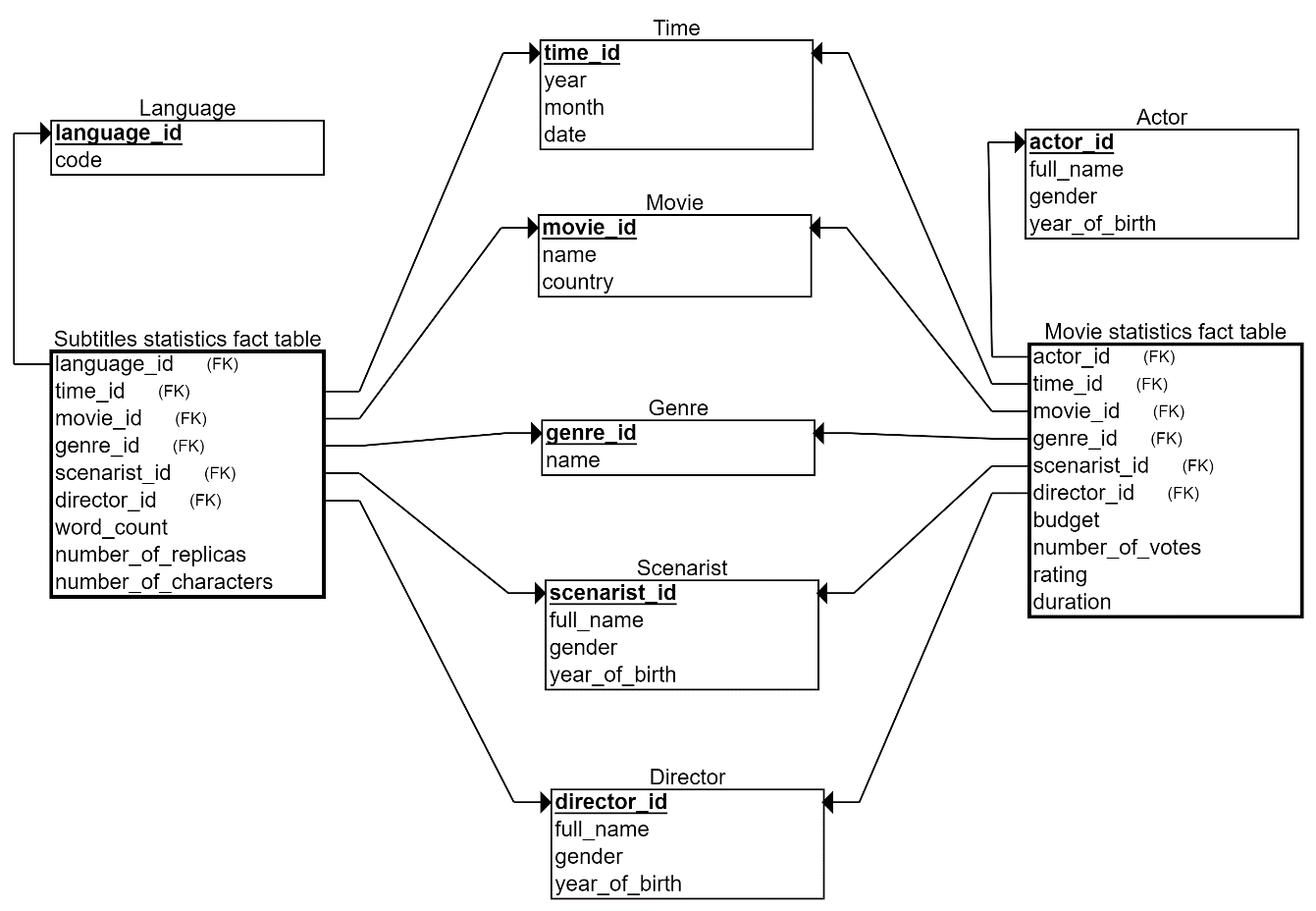
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time** | **Language** | **Movie** | **Genre** | **Director** | **Scenarist** |
| year | code | name | name | full name | full name |
| month |  | country |  | gender | gender |
| date |  |  |  | year of birth | year of birth |
|  |  |  |  |  |  |
| **Fakty**: word count, number of replicas, number of characters | | | | | |

* Dimenzia **Time** znamená dátum, kedy film vyšiel a má tri atribúty: rok, mesiac a deň.
* Dimenzia **Language** znamená jazyk titulkov k filmu a má jeden atribút: kód ktorý môže byť jednou z nasledujúcich hodnôt: 'it', 'cs', 'en', 'sk' alebo 'de'.
* Dimenzia **Movie** má dva atribúty: názov filmu a krajinu.
* Dimenzia **Genre** má len jeden atribút a to je názov žánru.
* Dimenzia **Director** má tri atribúty: celé meno, pohlavie a rok narodenia.
* Dimenzia **Scenarist** má tri atribúty: celé meno, pohlavie a rok narodenia.
  1. Štatistika filmov

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Time** | **Movie** | **Genre** | **Director** | **Scenarist** | **Actor** |
| year | name | name | full name | full name | full name |
| month | country |  | gender | gender | gender |
| date |  |  | year of birth | year of birth | year of birth |
|  |  |  |  |  |  |
| **Fakty**: budget, number of votes, rating, duration | | | | | |

* Dimenzia **Time** znamená dátum, kedy film vyšiel a má tri atribúty: rok, mesiac a deň.
* Dimenzia **Movie** má dva atribúty: názov filmu a krajinu.
* Dimenzia **Genre** má len jeden atribút a to je názov žánru.
* Dimenzia **Director** má tri atribúty: celé meno, pohlavie a rok narodenia.
* Dimenzia **Scenarist** má tri atribúty: celé meno, pohlavie a rok narodenia.
* Dimenzia **Actor** má tri atribúty: celé meno, pohlavie a rok narodenia.

1. Star schema



Obr. 11 Star schema

Na obrázku 11 je môžne vidieť našu star schému. Máme dve tabuľky faktov. Jedna tabuľka pre titulky a jedna pre filmy. Takmer všetky dimenzie sú spoločné okrem **Language** a **Actor.** Pre faktovu tabuľku titulkov sme nepridali dimenziu hercov (Actor), pretože počet slov nezávisí od herca, ale môže závisieť od scenárista a režiséra. A pre tabuľku faktov o filmoch sme nepridali dimenziu jazyka titulkov (Language), pretože rozpočet, počet hlasov, hodnotenie a trvanie filmu nezávisí od jazyka, v ktorom sú titulky vyrobené.

Typy údajov pre atribúty:

|  |  |
| --- | --- |
| **Language** | |
| language\_id | integer |
| code | integer |

|  |  |
| --- | --- |
| **Time** | |
| time\_id | integer |
| year | numeric |
| month | numeric |
| date | numeric |

|  |  |
| --- | --- |
| **Movie** | |
| movie\_id | integer |
| name | text |
| country | text |

|  |  |
| --- | --- |
| **Genre** | |
| genre\_id | integer |
| name | text |

|  |  |
| --- | --- |
| **Scenarist** | |
| scenarist\_id | integer |
| full\_name | text |
| gender | char |
| year\_of\_birth | numeric |

|  |  |
| --- | --- |
| **Director** | |
| director\_id | integer |
| full\_name | text |
| gender | char |
| year\_of\_birth | numeric |

|  |  |
| --- | --- |
| **Actor** | |
| actor\_id | integer |
| full\_name | text |
| gender | char |
| year\_of\_birth | numeric |

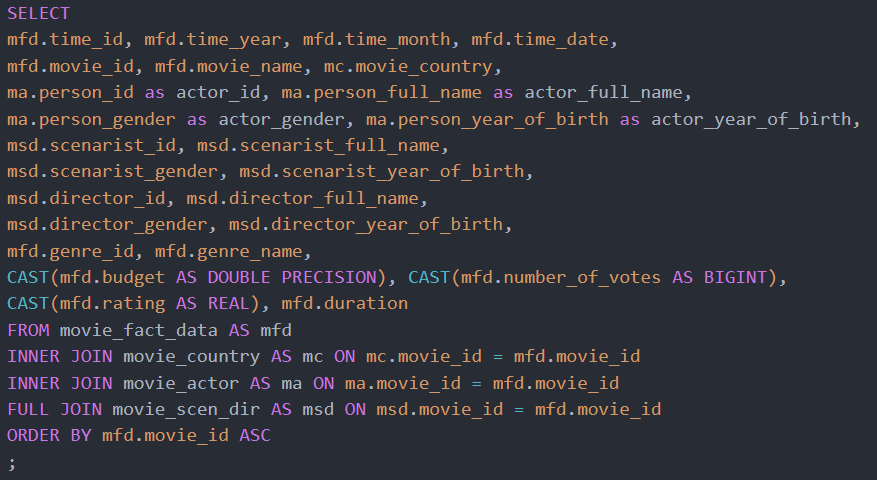
|  |  |
| --- | --- |
| **Subtitles fact table** | |
| language\_id | integer |
| time\_id | integer |
| movie\_id | integer |
| genre\_id | integer |
| scenarist\_id | integer |
| director\_id | integer |
| word\_count | bigint |
| number\_of\_replicas | bigint |
| number\_of\_characters | bigint |

|  |  |
| --- | --- |
| **Movie fact table** | |
| actor\_id | integer |
| time\_id | integer |
| movie\_id | integer |
| genre\_id | integer |
| scenarist\_id | integer |
| director\_id | integer |
| budget | double precision |
| number\_of\_votes | bigint |
| rating | real |
| duration | integer |

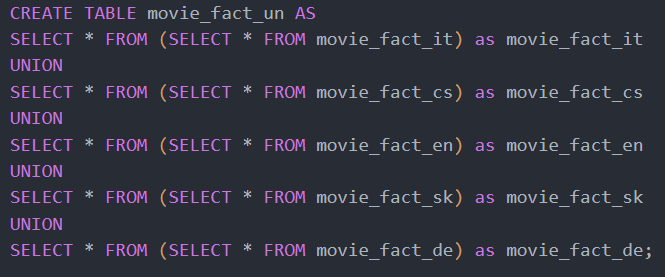
1. Nahrávanie dát do dátového skladu

Pre každú skupinu sme napísali selecty, ktoré vrátili všetky potrebné údaje pre náš dátovy sklad v jednej tabuľke a ak požadované údaje neboli dostupné, sme nastavili hodnotu na null. Všetky tieto SQL súbory pre každú skupinu sú dostupné na git na odkaze [github.com/Dmytro27Ind/movies-star-schema](https://github.com/Dmytro27Ind/movies-star-schema) . Na obrázku 12 je môžne vidieť select pre českú skupinu, ktorý vracia všetky potrebné údaje pre faktovu tabuľku **movie\_fact.**

Obr. 12 Select pre CS skupinu

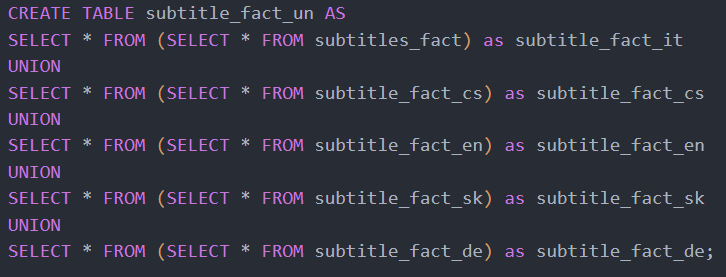


Keď sme mali údaje pre tabuľky faktov z každej skupiny, museli sme ich zlúčiť. Na obrázku 13 je môžne vidieť ako sme to urobili pre tabuľku **movie\_fact**.



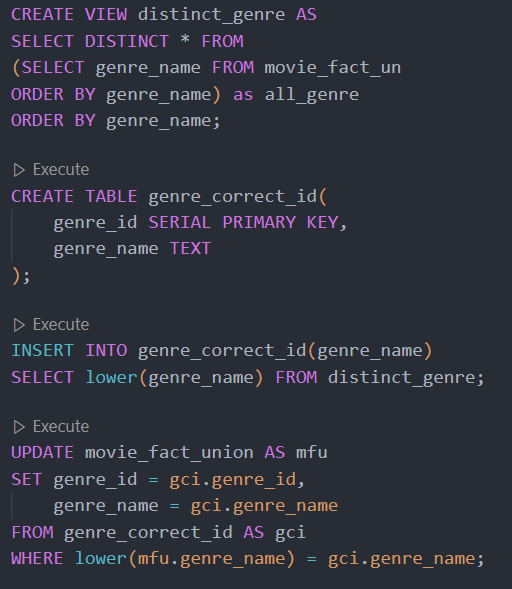
Obr. 13 Zlučovanie údajov zo všetkých skupín pre tabuľku movie\_fact

To isté sme urobili aj pre faktovu tabuľku titulkov **subtitles\_fact**. Na obrázku 14 je to môžne vidieť.



Obr. 14 Zlučovanie údajov zo všetkých skupín pre tabuľku subtitle\_fact

Po zlúčení všetkých skupín bolo potrebné nastaviť správne ID, keďže sa mohli opakovať v rôznych skupinách a odstrániť duplikáty. Na obrázku 15 môžete vidieť, ako to robíme pre žánre. Rovnakým spôsobom sme to robili pre filmy, hercov, scenáristov a režisérov.



Obr. 15 Nastavenie správneho ID a odstránenie duplikátov

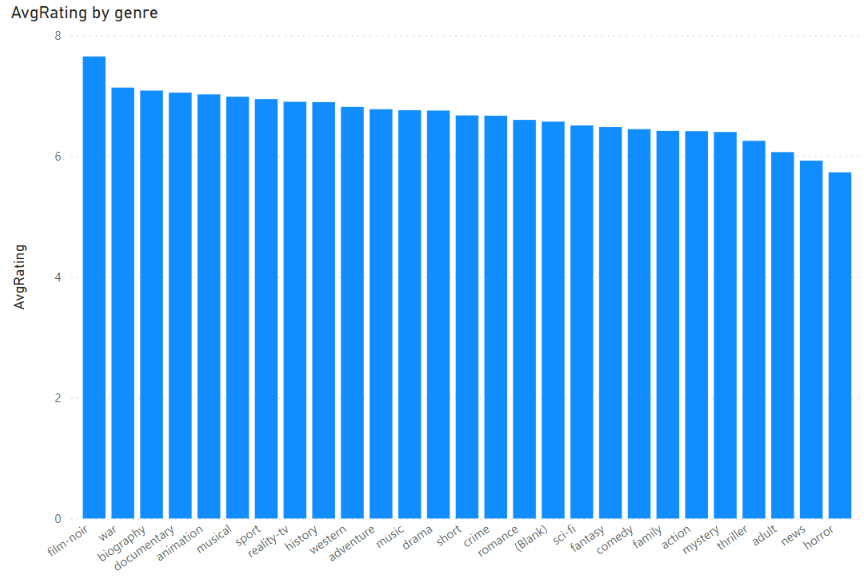
Ďalej sme importovali dáta do **Power BI** a rozdelili ich do samostatných dimenzí a odstránili duplikáty v údajoch pomocou funkcií **Power BI.** Vzťah medzi dimenziami a tabuľkami faktov je jeden k mnohým. Ako to vyzerá je možné vidieť na obrázku 16.



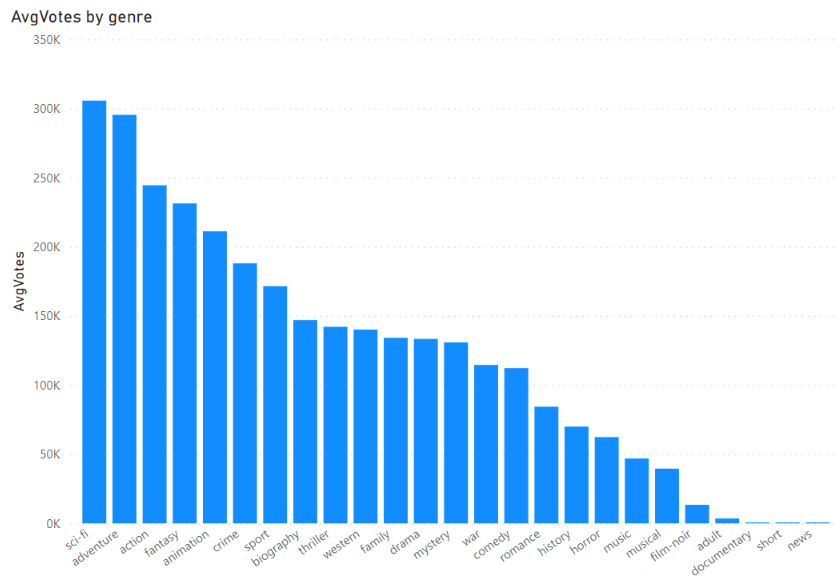
Obr. 16 Implementácia star schémy v Power BI

1. Vizualizácia dát

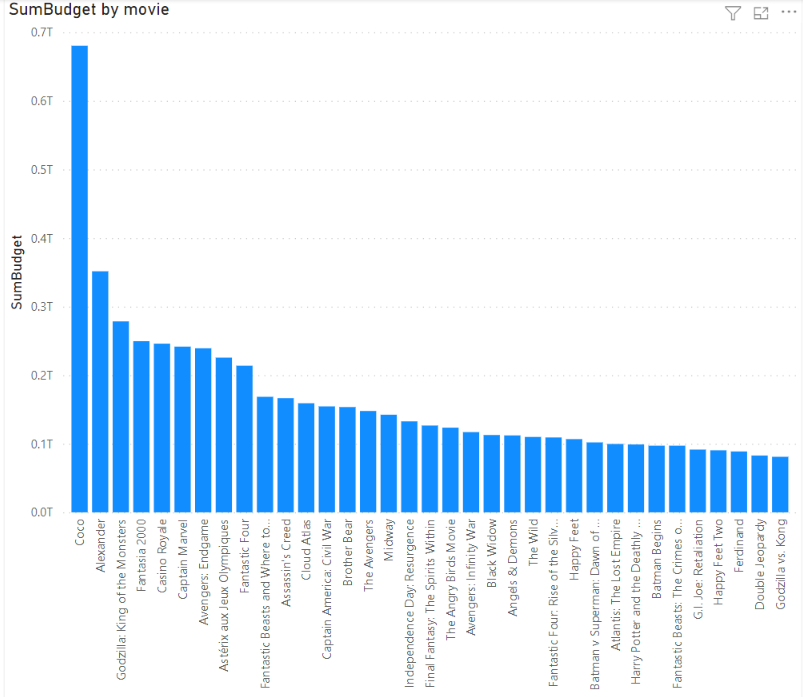
S pomocou **Power BI** je môžne veľmi jednoducho a rýchlo zostaviť ľubovoľné grafy alebo vykonávať výpočty bez potreby písania SQL dotazu.



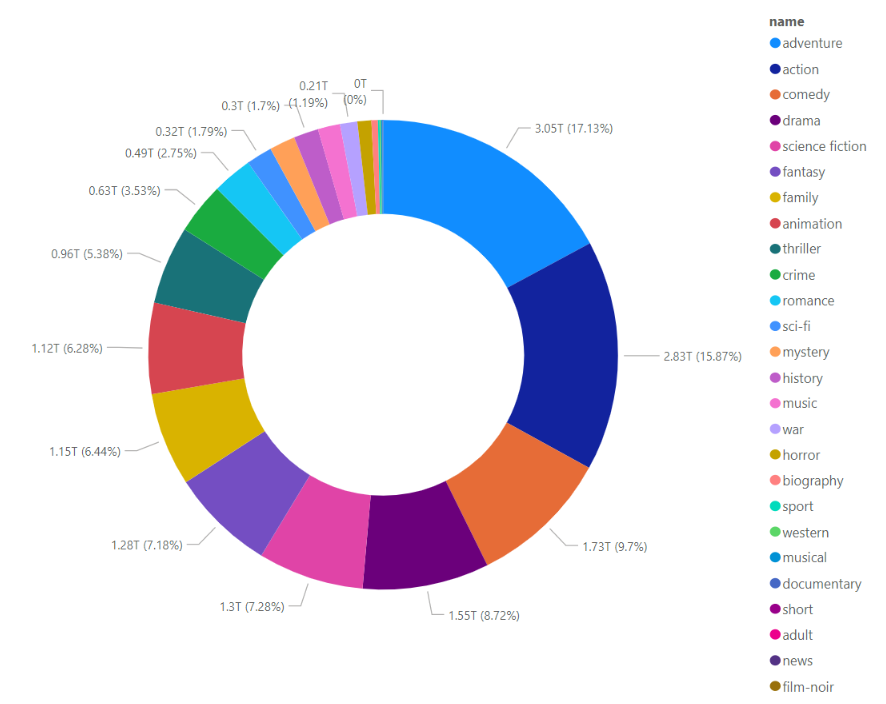
Obr. 17 Graf priemerného hodnotenia podľa žánru



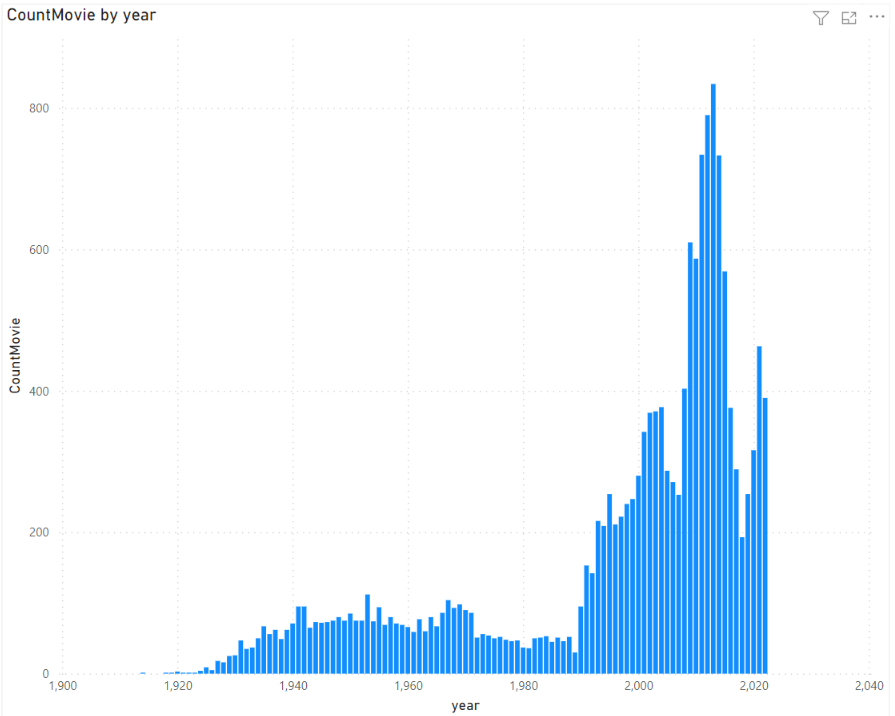
Obr. 18 Graf priemerného počtu hlasov podľa žánrov



Obr. 19 Graf rozpočtu podľa filmu



Obr. 20 Rozpočet podľa žánru

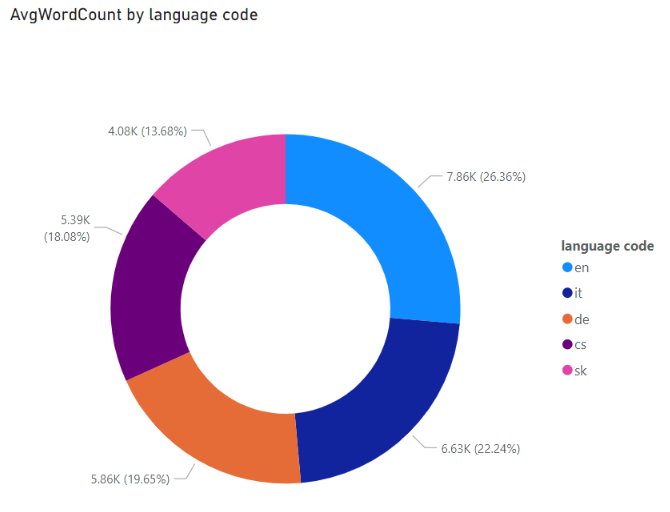


Obr. 21 Graf počtu filmov podľa rokov

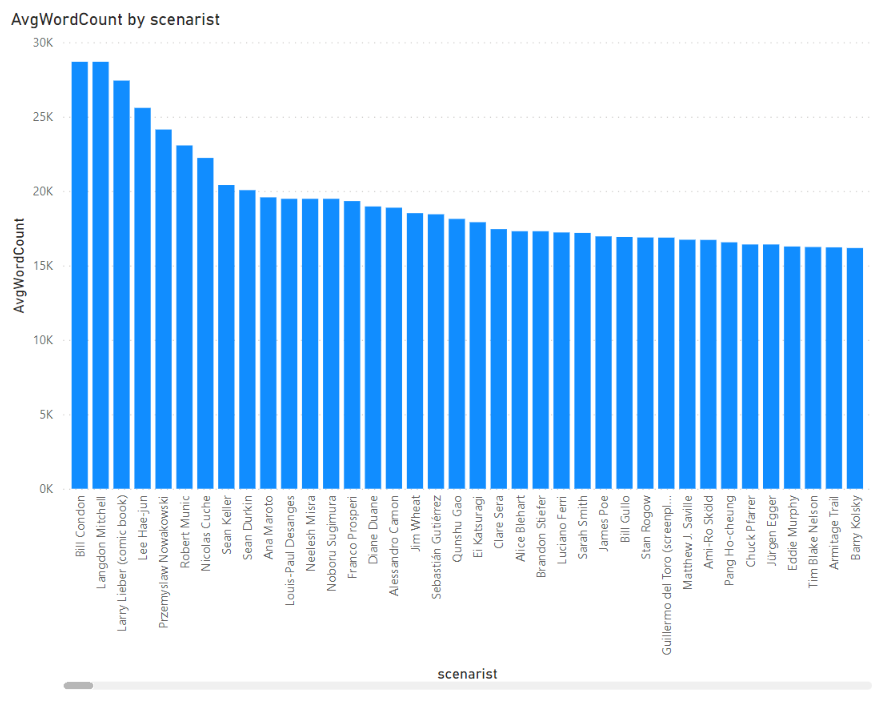
Obr. 22 Počet filmov podľa pohlavia režiséra



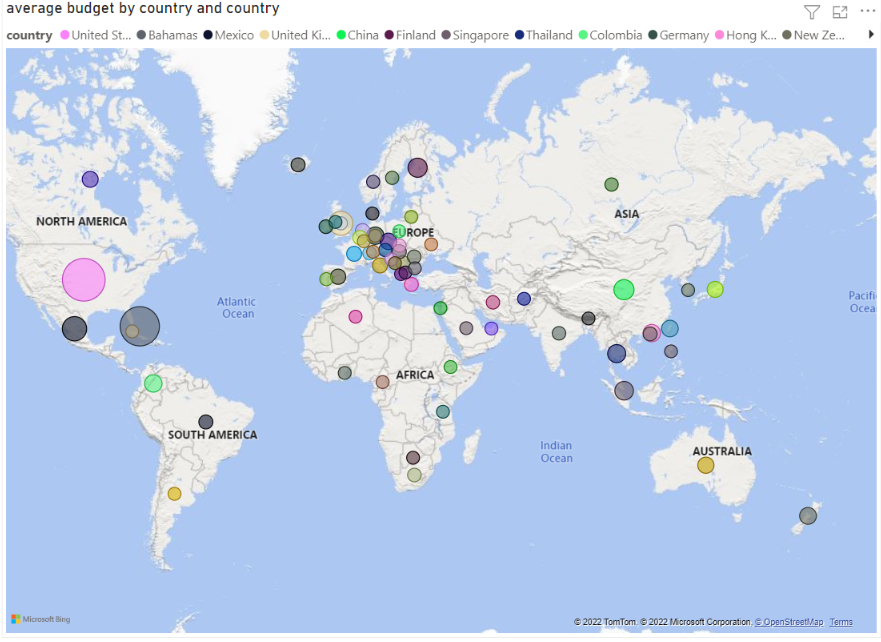
Obr. 23 Priemerný počet slov v titulkoch podľa jazyka



Obr. 24 Priemerný počet slov v titulkoch podľa scenáristu



Obr. 25 Priemerný rozpočet filmu podľa krajiny



Obr. 26 Priemerný počet hlasov podľa režiséra

