Part A

Section 1

- What is the dataset?

My dataset is about movies. It contains basic detailed information about movies (one artificial id, name, release year, the genre, country, the main actor, the production company, the number of viewers, the box office), the details of the main actor (one artificial id, the name, the home country, number of movies filmed, gender), the information about the genre (one artificial id , the genre name, one example movie, number of movies in the genre), information of the production company (one artificial id, company name, number of movies produced, the origin country), with other information about the country (one artificial id, the country name, population, abbreviation, the capital city, continent name).

- The sources of the data:

My information comes from web searched websites: Worldometers, The spread sheet guru, The numbers, Washington's Top News. The rest of the information is generated from AI tools like ChatGPT.

Country:

Worldometers: https://www.worldometers.info/geography/alphabetical-list-of-countries/

Thespreadsheetguru: https://www.thespreadsheetguru.com/list-countries-capitals-abbreviations/

Genre:

The numbers: https://www.the-numbers.com/market/genres

Washington's Top News: https://wtop.com/wp-content/uploads/2018/09/Best-Movies-in-Every-Genre.pdf

Company:

ChatGPT: https://chat.openai.com/c/2129cd35-5d41-43bb-8b20-ba69a9494115

Actor:

ChatGPT: https://chat.openai.com/c/f4ec2311-5b5b-4caa-83d2-9119bce0ec98

Movie:

ChatGPT: https://chat.openai.com/c/9a661af1-032c-4fe0-8a65-412c43108e82

- The license information:

I can’t find the license information on the websites. But ChatGPT said “As an AI language model, I don't own the generated data or have the authority to grant licenses for it. The generated table is based on fictional movie data created for the purpose of this interaction. You are free to use, modify, and distribute the generated data for personal or commercial purposes without any restrictions.”. My data is from Worldometers, Thespreadsheetguru, The numbers, Washington's Top News and ChatGPT. I am using the data educationally and you are not publishing anything to the public.

- What have the data been used for in the past:

In the past this data is used for statistics. Jannik(2023) used the data to write an statistics article to analyze the movie industry in 2023 and “help to identify trends and changes in the industry, as well as provide an understanding of the impact of certain films and filmmakers on the industry.”

Reference:

Lindner, J. (2023, December 16). Must-know movie statistics [recent analysis] • gitnux. GITNUX. https://gitnux.org/movie-statistics/

(I am sorry but this is the closest one I can find)

- Provide details of how you generated any simulated part of your data:

Country: (country_id, country_name, population, abbreviation, capital_city, continent_name) there are 195 rows. No foreign keys. No AI tools used.

Genre: (genre_id, genre_name, example_movie, num_movies) there are 15 rows. No foreign keys. No AI tools used.

Company: (company_id, company_name, num_movie, origin_country) there are 53 rows. There is one foreign key origin_country, I provide ChatGPT with the constraint, foreign key and form then asked for 50 to 60 values. Then I removed the repeating ones.

Actor: (actor_id, actor_name, home_country, num_movie, gender) there are 80 rows. There is one foreign key home_country, I provide ChatGPT with the constraint, foreign key and form then asked for 70 to 80 non-repeating values.

Movie: (movie_id, movie_name, movie_year, movie_genre, movie_country, movie_main_actor, production_company, movie_viewers, box_office) there are 137 rows. There are 4 foreign keys movie_genre, movie_country, movie_main_actor, production_company. I provide ChatGPT with the constraint, foreign keys and form then asked for 200 to 300 non-repeating values. Then I removed the repeating ones.

- What do you plan to do with the dataset:

With this data we will be able to explore the geographical presence of the movie industry, and research actors to find out which genre they are most talented and which company are they associated with most of the time.

- Example questions:

What is the genre with the highest average box office (most popular)?

What are the top 3 countries with the most movies?

Which company produces movies with highest average box office?

Section 2

My database have 5 tables: movie, actor, company, genre, country.

Country: (country_id, country_name, population, abbreviation, capital_city, continent_name) there are 6 columns 195 rows. The primary key is an artificial key country_id, which is an int. This key determines the rest of the columns and can not be NULL. There is no foreign key in the table. The other attributes includes the name of the country(country_name), which is varchar (100), the population of the country which is an int, the abbreviation of the country with only 3 chars which is varchar (3), the capital city of the country which is varchar (100), and the continent name which is varchar (100).

Genre: (genre_id, genre_name, example_movie, num_movies) there are 4 columns 15 rows. The primary key is an artificial key genre_id, which is an int. This key determines the rest of the columns and can not be NULL. There is no foreign key in the table. The other attributes include the name of the genre(genre_name), which is varchar (100), one example movie, which is varchar (100), and the number of movies in the genre which is an int.

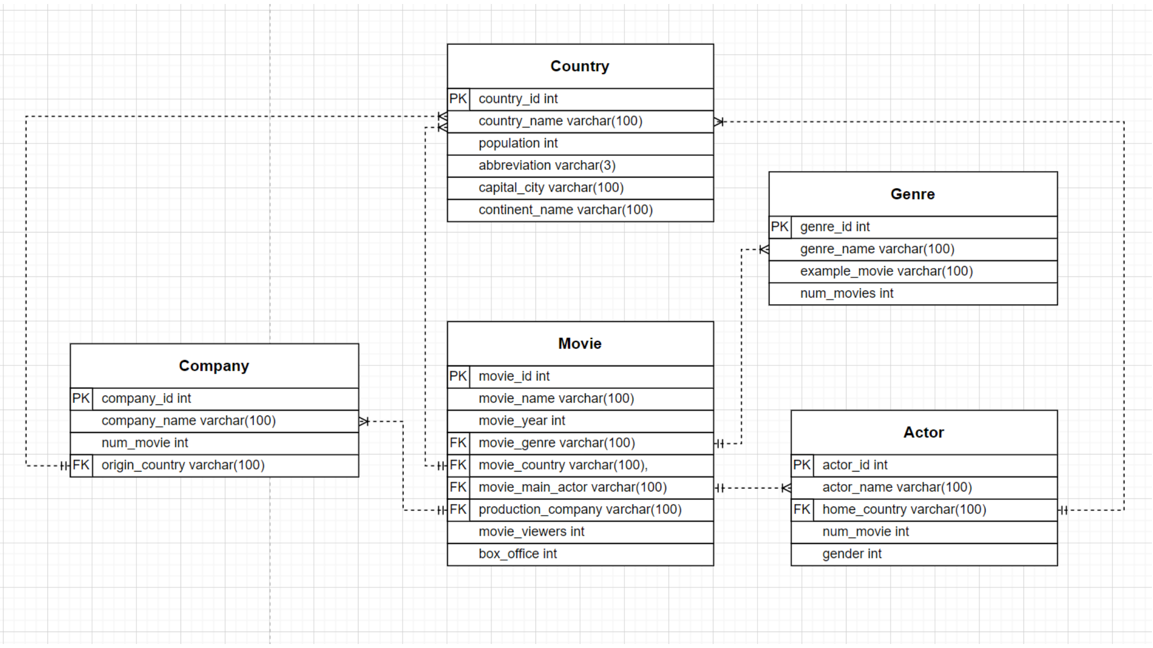
Company: (company_id, company_name, num_movie, origin_country) there are 4 columns 53 rows. The primary key is an artificial key company_id, which is an int. This key determines the rest of the columns and can not be NULL. There is one foreign key in the table, which is origin_country. It references country_name in the country table, which means the country_name determines this key. The other attributes include the name of the company(company_name), which is varchar (100) and the number of movies made by the company which is a int.

Actor: (actor_id, actor_name, home_country, num_movie, gender) there are 5 columns 80 rows. The primary key is an artificial key actor_id, which is an int. This key determines the rest of the columns and can not be NULL. There is one foreign key in the table, which is home_country. It references country_name in the country table, which means the country_name determines this key. The other attributes includes the name of the actor(actor_name), which is varchar (100), the number of movies filmed by the actor which is a int, and the gender of the actor, which is represented by a int, 1 for male and 0 for female.

Movie: (movie_id, movie_name, movie_year, movie_genre, movie_country, movie_main_actor, production_company, movie_viewers, box_office) there are 9 columns 137 rows. The primary key is an artificial key movie_id, which is an int. This key determines the rest of the columns and can not be NULL. There are four foreign keys in the table, they are: movie_genre, It references genre_name in the genre table; movie_country, It references country_name in the country table; movie_main_actor, It references actor_name in the actor table; production_company, It references company_name in the company table. The other attributes includes the name of the movie(movie_name), which is varchar(100), the year of the movie which is a int, the amount of viewers which is a int and the box office which is a int.

Section 3

- Internal schema


Country
PK country_id int
country_name varchar(100)
population int
abbreviation varchar(3)
capital_city varchar(100)
continent_name varchar(100)
Genre
PK genre_id int
genre_name varchar(100)
example_movie varchar(100)
num_movies int
Company
PK company_id int
company_name varchar(100)
num_movie int
FK origin_country varchar(100)
Movie
PK movie_id int
movie_name varchar(100)
movie_year int
FK movie_genre varchar(100)
FK movie_country varchar(100),
FK movie_main_actor varchar(100)
FK production_company varchar(100)
movie_viewers int
box_office int
Actor
PK actor_id int
actor_name varchar(100)
FK home_country varchar(100)
num_movie int
gender int

- Dependency diagrams

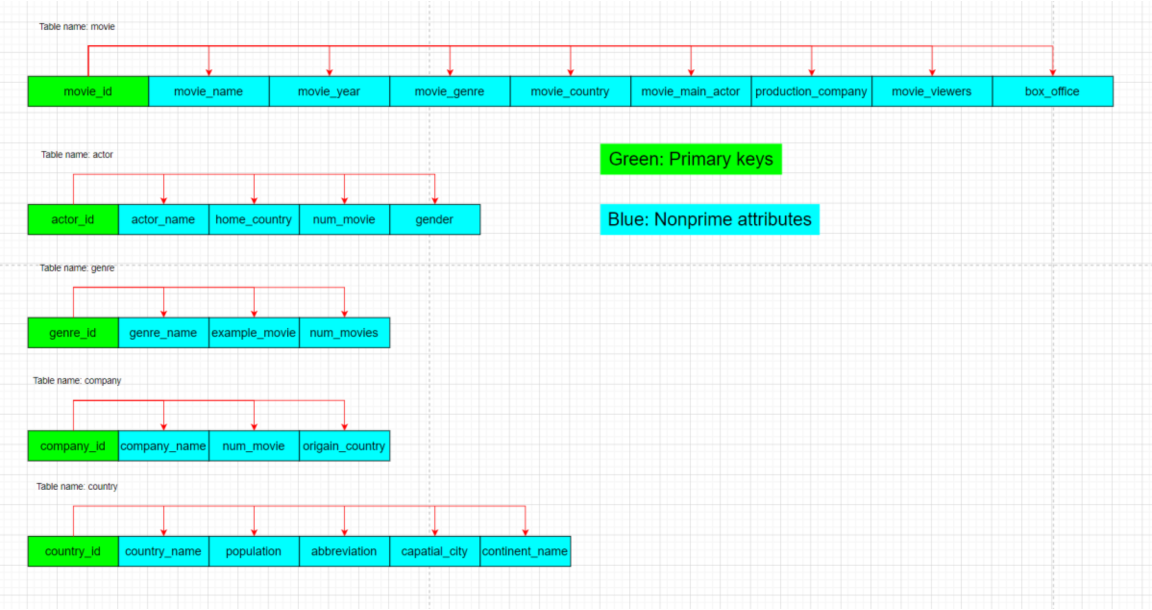


Table name: movie

movie_id	movie_name	movie_year	movie_genre	movie_country	movie_main_actor	production_company	movie_viewers	box_office

Table name: actor

actor_id	actor_name	home_country	num_movie	gender

Table name: genre

genre_id	genre_name	example_movie	num_movies

Table name: company

company_id	company_name	num_movie	origain_country

Table name: country

country_id	country_name	population	abbreviation	capatial_city	continent_name

Green: Primary keys

Blue: Nonprime attributes

- Explanation of why these tables are in 3NF

Country: (country_id, country_name, population, abbreviation, capital_city, continent_name) The primary key for this table is the artificial key country_id, and the primary key directly determines all the other attributes in the table. country_name also uniquely identify all the rows in the table, but is not as determined as the artificial key. There is no transitive dependency or partial dependencies in the table. Hence it is in the 3NF.

Genre: (genre_id, genre_name, example_movie, num_movies) The primary key for this table is the artificial key genre_id, and the primary key directly determines all the other attributes in the table. genre_name also uniquely identify all the rows in the table, but is not as determined as the artificial key. There is no transitive dependency or partial dependencies in the table. Hence it is in the 3NF.

Company: (company_id, company_name, num_movie, origin_country) The primary key for this table is the artificial key company_id, and the primary key directly determines all the other attributes in the table. company_name also uniquely identify all the rows in the table, but is not as determined as the artificial key. There is no transitive dependency or partial dependencies in the table. Hence it is in the 3NF.

Actor: (actor_id, actor_name, home_country, num_movie, gender) The primary key for this table is the artificial key actor_id, and the primary key directly determines all the other attributes in the table. actor_name also uniquely identify all the rows in the table, but is not as determined as the artificial key. There is no transitive dependency or partial dependencies in the table. Hence it is in the 3NF.

Movie: (movie_id, movie_name, movie_year, movie_genre, movie_country, movie_main_actor, production_company, movie_viewers, box_office) The primary key for this table is the artificial key movie_id, and the primary key directly determines all the other attributes in the table. movie_name also uniquely identify all the rows in the table, but is not as determined as the artificial key. There is no transitive dependency or partial dependencies in the table. Hence it is in the 3NF.

Hence this database is in the 3NF. Because there is no multivalued attributes, and repeated columns, and there is no partial dependencies and transitive dependencies.