Completed by Nick Gundobin

110805377

Report on Horror Movies Database.

Building Horror Movies database was a true labor of passion for Horror movies as genre and just learning experience. It was as exhausting as it was rewarding. This particular database would be of a great interest to someone similar interest or using it for the websites like UpcomingHorrorMovies.com. For this project, MySQL database was used and Python programming language to manipulate, scrape and clean the data by using different libraries like Pandas, Numpy, BeautifulSoup etc. This database has a little over than 1000 entries with movie names, directors, actors, release years and a lot more. In order to see a real picture of this database, we will consider ER model with entities and their relationships.

1. Entities with their domains and Primary and Foreign Keys

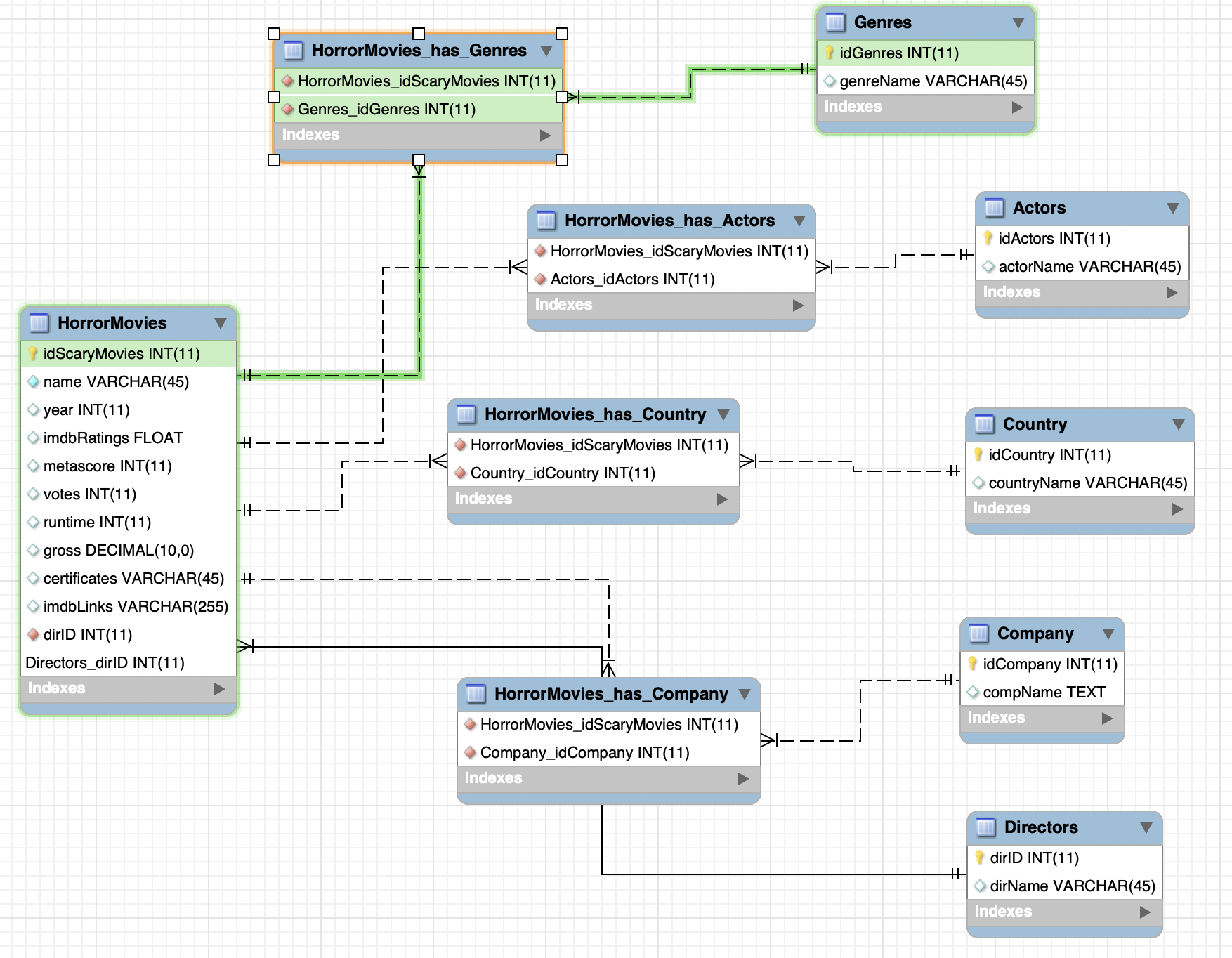
A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

1. All Relationships





There are 6 total entities and 5 relationships where 4 are many-to-many: movies with actors, movies with companies, movies with countries, movies with genres and one-to-many is movies with director. In this model, one director made multiple movies but only one movie made by one director. It is a high unlikely to have more than one director per movie unless it is TV-show. HorrorMovies contains Foreign key from Director to connect the tables. Company, Country, Actors, Genres and HorrorMovies have third entity in order to join their tables with foreign keys in those entities.



1. The GUI was created by means of Flask and runs on the local browser. To run it, go to Terminal and run command ‘python hm\_database.py’. Then right after open any browser and paste this command into url ‘http://127.0.0.1:5000/’.

A screenshot of a cell phone

Description automatically generated

After successfully completing those steps, you should see above screen image. There are 3 options one can select: for those who feels comfortable with SQL you can do search directly by entering it into the box, Interesting Queries, Forms.

A screenshot of a cell phone

Description automatically generated

**Interesting Queries:**

There are 5 different queries, one can discover on the below screen.

A screenshot of a social media post

Description automatically generated

1. SELECT name,Directors.dirName, year, metascore,runtime, countryName

FROM HorrorMovies,Directors,HorrorMovies\_Country, Country

WHERE HorrorMovies.dirID = Directors.dirID AND

HorrorMovies.idScaryMovies = HorrorMovies\_Country.idScaryMovies

AND Country.idCountry = HorrorMovies\_Country.idCountry

AND year = year(CURRENT\_DATE)

AND metascore IS NOT NULL

ORDER BY name;

This particular query join 3 tables and give you information on the horror movies that came out this year.

A screenshot of a cell phone

Description automatically generated

1. SELECT name, dirName, imdbRatings, year

FROM HorrorMovies,Directors

WHERE HorrorMovies.dirID = Directors.dirID

ORDER BY imdbRatings DESC, year

LIMIT 10;

This query will show you information on 10 best rated movies of all time. It joins 2 tables.

A screenshot of a cell phone

Description automatically generated

WITH maxRate AS(SELECT year, MAX(imdbRatings) AS maxRate

FROM HorrorMovies

GROUP BY year)

SELECT HorrorMovies.year,name, HorrorMovies.imdbRatings

FROM HorrorMovies, maxRate

WHERE HorrorMovies.imdbRatings = maxRate.maxRate AND

HorrorMovies.year = maxRate.year

ORDER BY year DESC;

This query will give you information on the best rated movies by year starting from early 1900 up to this year.

A screenshot of a cell phone

Description automatically generated

1. SELECT name, year, imdbRatings, dirName, countryName,

concat('$ ', format(gross, 2)) AS Revenue

FROM HorrorMovies, Country, HorrorMovies\_Country, Directors

WHERE HorrorMovies.idScaryMovies = HorrorMovies\_Country.idScaryMovies

AND Country.idCountry = HorrorMovies\_Country.idCountry

AND Directors.dirID = HorrorMovies.dirID

ORDER BY gross DESC

LIMIT 10;

This query will inform you on 10 highest grossing movies of all the time and it joins 3 tables in total.

A screenshot of a cell phone

Description automatically generated

1. SELECT dirName AS Director, COUNT(name) AS 'Movie Count'

FROM HorrorMovies, Directors

WHERE HorrorMovies.dirID = Directors.dirID

GROUP BY dirName

ORDER BY COUNT(name) DESC;

This quiry will show you number of movies made by a particular director. It will join 2 tables.

A screenshot of a cell phone

Description automatically generated

**Forms:**

**A screenshot of a cell phone

Description automatically generated**

**There are 2 options one can choose: Search movies and Insert Movie.**

A screenshot of a cell phone

Description automatically generated

One can search by movie name, actor or director by inputing the name of the movie or pattern if one can’t remember the movie it will give all found results. Let us search for “Dawn of the Dead”. The query will return the following results:

A screenshot of a cell phone

Description automatically generated

Finally, to insert movie, one can enter info in the following form and submit to database.

A screenshot of a cell phone

Description automatically generated

Those forms involve complex queries that have more than 3 joined tables.