Sample Database: Movies (ERD and SQL)

This post describes a sample database containing data about movies. It includes:

- An ERD (entity relationship diagram) for the sample movie
- database an explanation of the tables and columns
- A download of sample data to create and populate this
- database an example query on the database

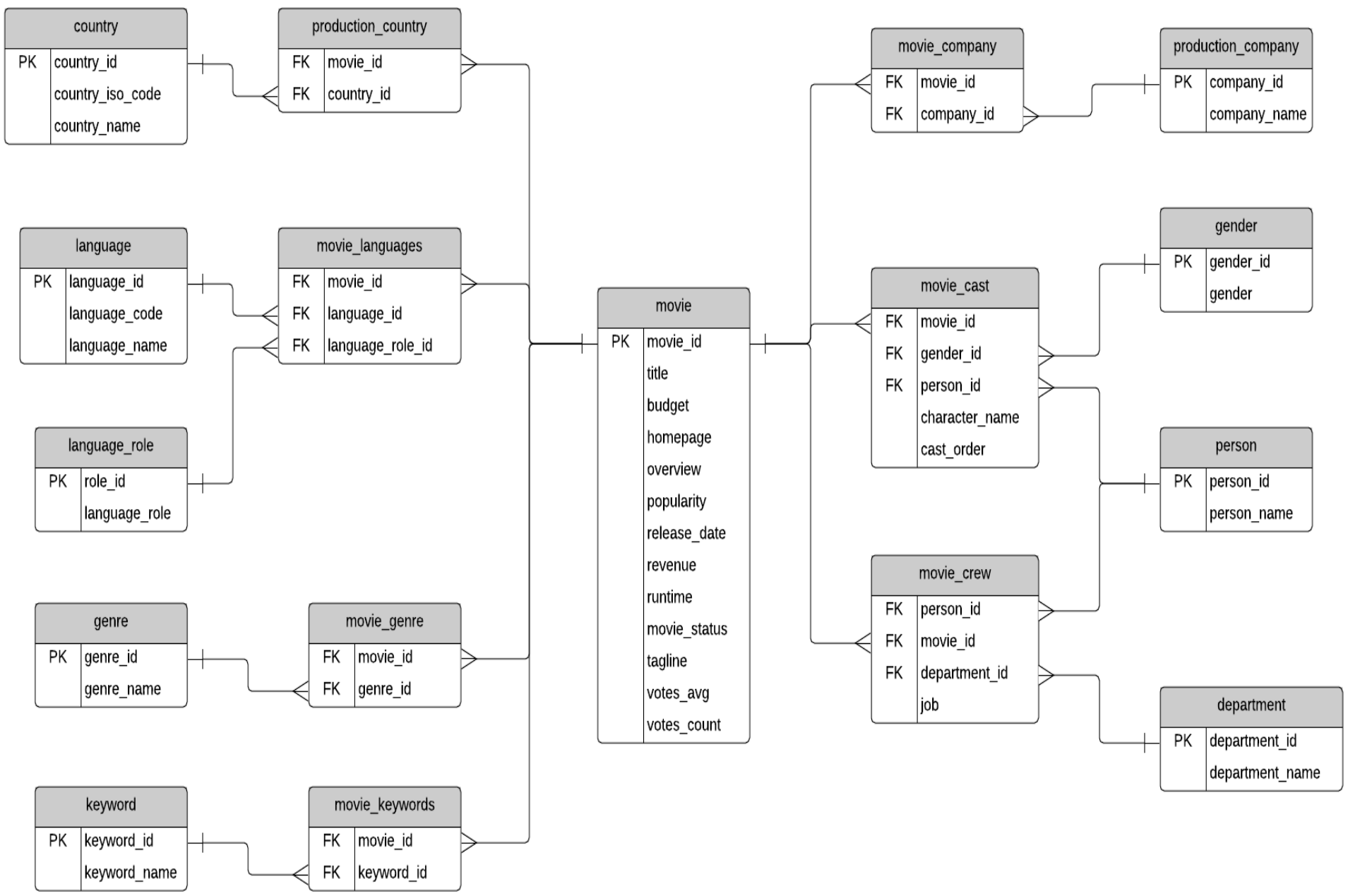
Why is this helpful? Firstly, you can understand more about how a movie database might work.

Also, you can practice SQL against realistic data and write your own queries, both simple (how many movies has Tom Cruise been in?) and complex (which movies have Tom Cruise and Matt Damon both been in?).

So, let's take a look at the database.

Sample Movie Database: ERD

The ERD or database design of the sample movie database is here (open in new tab, or save, to see a larger version):



This database stores information about movies, the cast and crew involved, where the movie was produced and by which company, and other information about movies such as the languages, genres, and keywords.

The sample data was obtained from a free online data source. It contains about 4,800 movies, 104,000 cast and crew, and thousands of metadata records such as languages and keywords.

What do all of these tables and columns mean?

Table Explanations

The **movie** table contains information about each movie. There are text descriptions such as title and overview. Some fields are more obvious than others: revenue (the amount of money the movie made), budget (the amount spent on creating the movie). Other fields are calculated based on data used to create the data source: popularity, votes_avg, and votes_count. The status indicates if the movie is Released, Rumoured, or in Post-Production.

The **country** list contains a list of different countries, and the **movie_country** table contains a record of which countries a movie was filmed in (because some movies are filmed in multiple countries). This is a standard many-to-many table, and you'll find these in a lot of databases.

The same concept applies to the **production_company** table. There is a list of production companies and a many-to-many relationship with movies which is captured in the **movie_company** table.

The **languages** table has a list of languages, and the movie_languages captures a list of languages in a movie. The difference with this structure is the addition of a **language_role** table. This language_role table contains two records: Original and Spoken. A movie can have an original language (e.g. English), but many Spoken languages. This is captured in the **movie_languages** table along with a role.

Genres define which category a movie fits into, such as Comedy or Horror. A movie can have multiple genres, which is why the **movie_genres** table exists.

The same concept applies to **keywords**, but there are a lot more keywords than genres. I'm not sure what qualifies as a keyword, but you can explore the data and take a look. Some examples as "paris", "gunslinger", or "saving the world".

The cast and crew section of the database is a little more complicated. Actors, actresses, and crew members are all people, playing different roles in a movie. Rather than have separate lists of names for crew and cast, this database contains a table called **person**, which has each person's name.

The **movie_cast** table contains records of each person in a movie as a cast member. It has their character name, along with the cast_order, which I believe indicates that lower numbers appear higher on the cast list.

The **movie_cast** table also links to the gender table, to indicate the gender of each character. The gender is linked to the movie_cast table rather than the person table to cater for characters which may be a different gender than the person, or characters of unknown gender. This means that there is no gender table linked to the person table, but that's because of the sample data.

The **movie_crew** table follows a similar concept and stores all crew members for all movies. Each crew member has a job, which is part of a **department** (e.g. Camera).

Sample Query

With the sample data in the database, let's take a look at some of the data in the movie table. This query shows the movie title, budget, and other attributes of the movie, sorted by the movies with the highest revenue.

```
SELECT
       title,
       budget,
       release_date,
       revenue,
       runtime,
       vote_average
FROM movie
ORDER BY revenue DESC;
```

Results (top 15 rows only):

title	budget	release_date	revenue	runtime	vote_average
Avatar	237000000	2009-12-10	2787965087	162	7.2
Titanic	200000000	1997-11-18	1845034188	194	7.5
The Avengers	220000000	2012-04-25	1519557910	143	7.4
Jurassic World	150000000	2015-06-09	1513528810	124	6.5
Furious 7	190000000	2015-04-01	1506249360	137	7.3
Avengers: Age of Ultron	280000000	2015-04-22	1405403694	141	7.3
Frozen	150000000	2013-11-27	1274219009	102	7.3
Iron Man 3	200000000	2013-04-18	1215439994	130	6.8
Minions	74000000	2015-06-17	1156730962	91	6.4
Captain America: Civil War	250000000	2016-04-27	1153304495	147	7.1
Transformers: Dark of the Moon	195000000	2011-06-28	1123746996	154	6.1
The Lord of the Rings: The Return of the King	94000000	2003-12-01	1118888979	201	8.1
Skyfall	200000000	2012-10-25	1108561013	143	6.9
Transformers: Age of Extinction	210000000	2014-06-25	1091405097	165	5.8

Lab: SQL Queries

Setup

Test by writing SQL expression for the following queries:

1. How many movies are there?
2. How many movies have titles starting with 'Star Trek'?
3. How many actors are there?
4. Which movie has smallest cast?
5. Which movie has largest cast?
6. Which actor has been featured in the highest number of movies?

1) Inner join

Write SQL expressions for the following queries.

1. Obtain the name of actors in the cast list for the film 'Die Hard'.
2. List the title of films in which 'Kevin Costner' has appeared
3. List the title of films where 'Kevin Costner' has appeared - but not in the lead star role. (Lead star role is defined by casting.order = 1.)
4. List the title of films together with their stars for all 1967 films.
5. List the title of 1970 films by order of cast list size.
6. List the name of actors featured in at least 13 films released after 1990.

2) Outer join

Write SQL expressions for the following queries.

1. List the film title and the lead actor name for all 1960 films. Be sure that movie 'Psycho' is in the list.
2. List the name of actors whose name starts 'U' and the number of films that the actor played. Include actors with no film.
3. For every actor, list the number of movies starring him/her as the first person on the casting. Include all actor with names starting with 'Z' in the result.

3) Self join

Write SQL expressions for the following queries.

1. List movie pairs with common titles. Provide movie ids and years to explain the pairs.
2. List the name pair of actors who have worked together in more than 2 films. (The name of actors starts with 'A'.)
3. List the title pair of movies which have the same lead star actor. (Restrict result to movies with names starting with 'V'.)

4) **Nested queries(I)**

Write SQL expressions for the following queries.

1. Which were the busiest years for 'Al Pacino'.
2. List title and lead actor name for of films starring 'Jesse Ventura' .
3. Obtain the name of actors in who have had at least 15 lead star roles.
4. List the name of all actors who have worked with 'James Dean'.

5) **Nested queries (II) using exists, not exists**

Write SQL expressions for the following queries.

1. List actors who have worked with 'Al Pacino'.
2. List the name of actors who don't play in any movie 'Mel Gibson' has appeared in.(Restrict result to the actors whose name begin with 'Z'.)
3. List names of actors with lead star role in a 1940s movie.

6) **Queries using set operators, union, intersect and minus**

Write SQL expressions for the following queries.

1. List titles of movies both 'Tom Hanks' and 'Meg Ryan' have appeared in.
2. List the title of movies featuring 'Harrison Ford' but not 'Anthony Daniels' not.
3. List titles of movies starting with 'Die Hard' or starring 'Bruce Willis'. Ensure there the list has no duplicates.

7) **Relational Division**

Write SQL expressions for the following queries.

1. List names of actors who acted in every movie with titles starting with 'Die Hard'.
2. List the name of actors who acted in all of films featuring 'Jesse Ventura'.
3. List the title of movies which feature all of actors featured in 'Wallace & Gromit: A Grand Day Out' and 'Wallace & Gromit: The Wrong Trousers' movies.