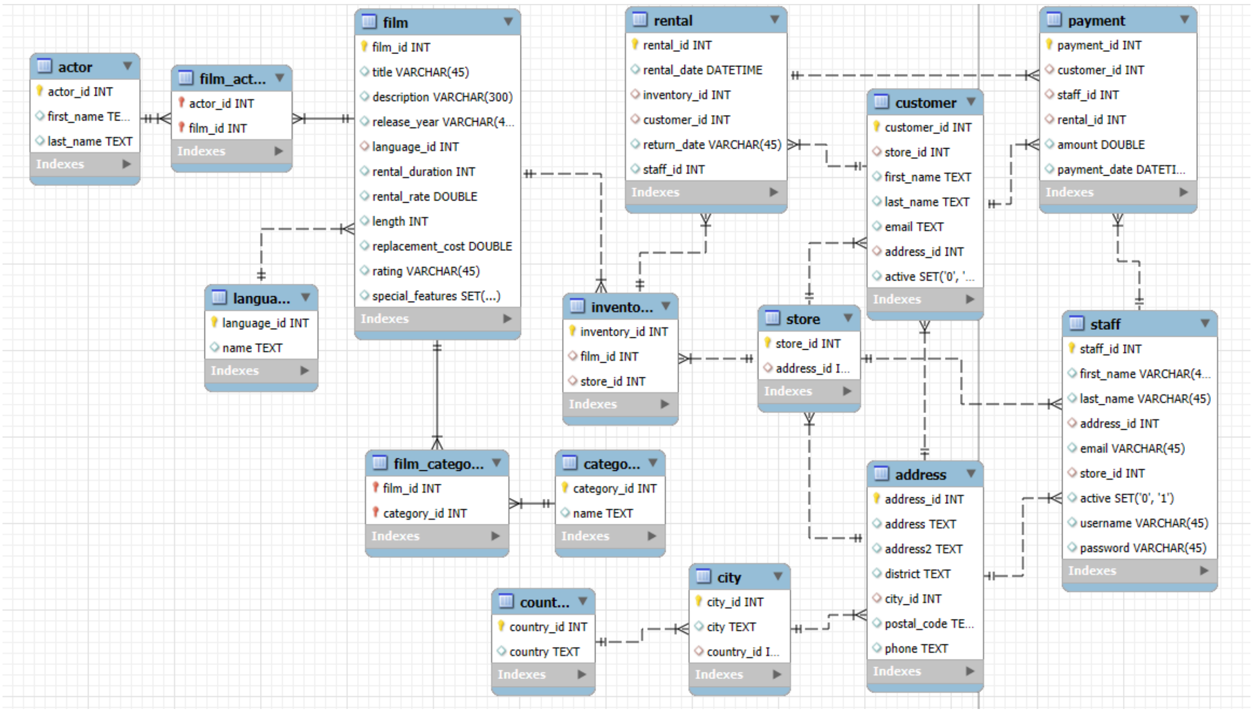
Title: DB Assignment 4

Name: James Letts

Date: 11/04/2025

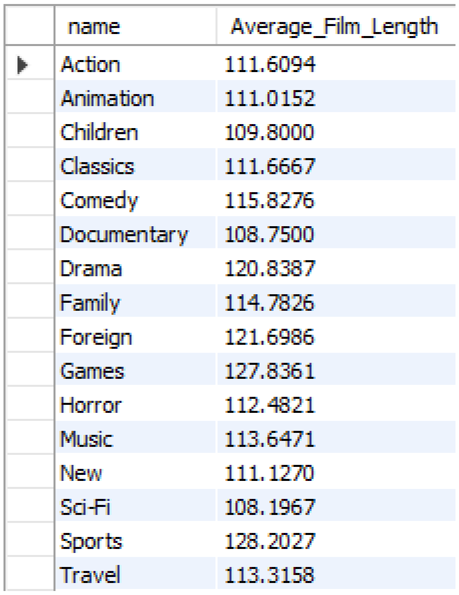
ERD Diagram:



1) **Query 1:** What is the average length of films in each category? List the results in alphabetic order of categories.

    a) Solution Explanation: A fairly simple query solves this one. Simply select distinct category names alongside the average film length from film using an aggregate. After that you simply need to join the tables together with category >> film_category >> film using joins and Group By and Order By the category name.

    b) Solution Result:

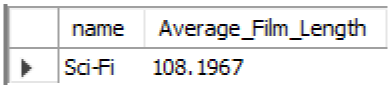


name	Average_Film_Length
Action	111.6094
Animation	111.0152
Children	109.8000
Classics	111.6667
Comedy	115.8276
Documentary	108.7500
Drama	120.8387
Family	114.7826
Foreign	121.6986
Games	127.8361
Horror	112.4821
Music	113.6471
New	111.1270
Sci-Fi	108.1967
Sports	128.2027
Travel	113.3158

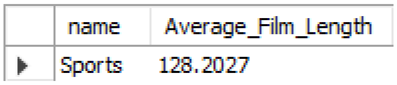
2) **Query 2:** Which categories have the longest and shortest average film lengths?

    a) Solution Explanation: To answer this question in the easiest manner, just use two queries that do opposite things. Both queries are nearly the same as query 1, except for the Order By being by the average film length, one ascending the other descending, and Limiting the results by 1.

    b) Solution Result:



name	Average_Film_Length
Sci-Fi	108.1967

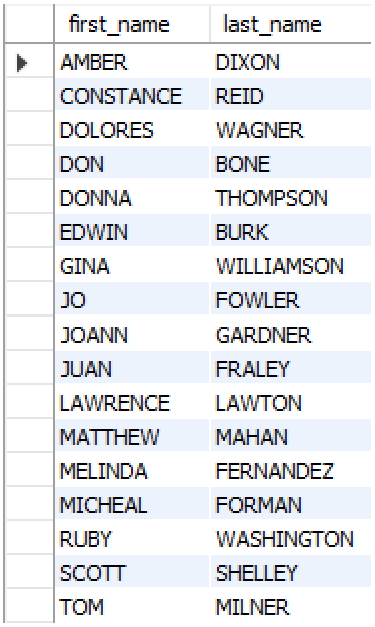


name	Average_Film_Length
Sports	128.2027

3) **Query 3:** Which customers have rented action but not comedy or classic movies?

a) Solution Explanation: This is a long one, but it is still simple enough. Essentially, we need to make 3 queries in 1, with the main query selecting the first and last name of customers and joining the tables to category and filtering the names by choosing the ones that have ordered Comedy. This is then followed by the next two queries within a NOT IN filter, both of which are basically the same but filtered into those who ordered Action and Classic.

b) Solution Result:

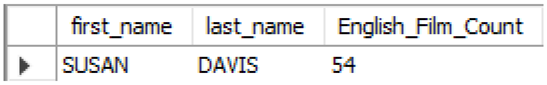


first_name	last_name
AMBER	DIXON
CONSTANCE	REID
DOLORES	WAGNER
DON	BONE
DONNA	THOMPSON
EDWIN	BURK
GINA	WILLIAMSON
JO	FOWLER
JOANN	GARDNER
JUAN	FRALEY
LAWRENCE	LAWTON
MATTHEW	MAHAN
MELINDA	FERNANDEZ
MICHEAL	FORMAN
RUBY	WASHINGTON
SCOTT	SHELLEY
TOM	MILNER

4) **Query 4:** Which actor has appeared in the most English-language movies?

a) Solution Explanation: While I did not take this route, I feel it's important to note that one could simply just check what actor is in the most movies in general, because after checking the data tables, it becomes clear that there is not a single movie in the table that is not credited as being in english. However, the solution is simple even without using this; you Select the actor names along with a count of their filmography. Then it's as easy as joining a language and filtering films by the English id, which is 1. Finally, to only get the actor with the most, you simply order by the COUNT result in descending order and limit the results to 1.

b) Solution Result:

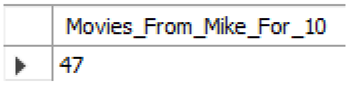


	first_name	last_name	English_Film_Count
▶	SUSAN	DAVIS	54

5) **Query 5:** How many distinct movies were rented for exactly 10 days from the store where Mike works?

a) Solution Explanation: It is now that I am thankful that I realized that the database you share with us was a toned down version of the Sakila database that I already had on my computer so I emulated the data types, because since I made all of the dates the “datetime” type, this query was mindnumbingly easy. You literally just take the count of the films and use the DateDiff Aggregate function to filter it by films that have 10 days in between the rental and return date, with a small statement at the end specifying the staff name as Mike.

b) Solution Result:

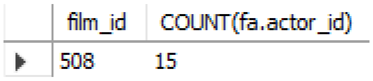


	Movies_From_Mike_For_10
▶	47

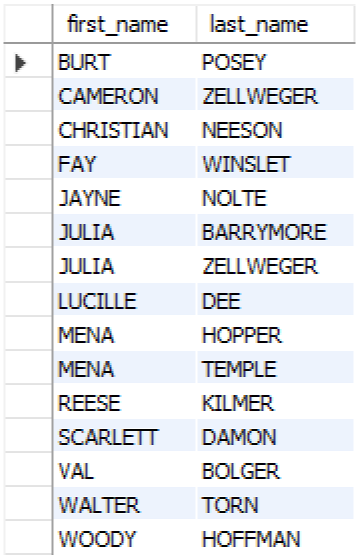
6) **Query 6:** Alphabetically list actors who appeared in the movie with the largest cast of actors.

a) Solution Explanation: This answer is a bit of a stretch, as it technically requires you to type in the answer to a separate query into a new query to get the correct answer. However, with how long the query would be otherwise, I still feel that this is more efficient than the one long query. The first query selects the film_id and actor count of all of the film_ids, sorts them by the actor count descending and limits the results to one, to get the highest one. Then you simply take the film id and use it to filter a list of alphabetically sorted names of actors into only the actors who were in that film. If the number of actors listed is equal to the actor count of the first query, then you did it correctly, which it did, so I did.

b) Solution Results:



film_id	COUNT(fa.actor_id)
508	15



first_name	last_name
BURT	POSEY
CAMERON	ZELLWEGER
CHRISTIAN	NEESON
FAY	WINSLET
JAYNE	NOLTE
JULIA	BARRYMORE
JULIA	ZELLWEGER
LUCILLE	DEE
MENA	HOPPER
MENA	TEMPLE
REESE	KILMER
SCARLETT	DAMON
VAL	BOLGER
WALTER	TORN
WOODY	HOFFMAN