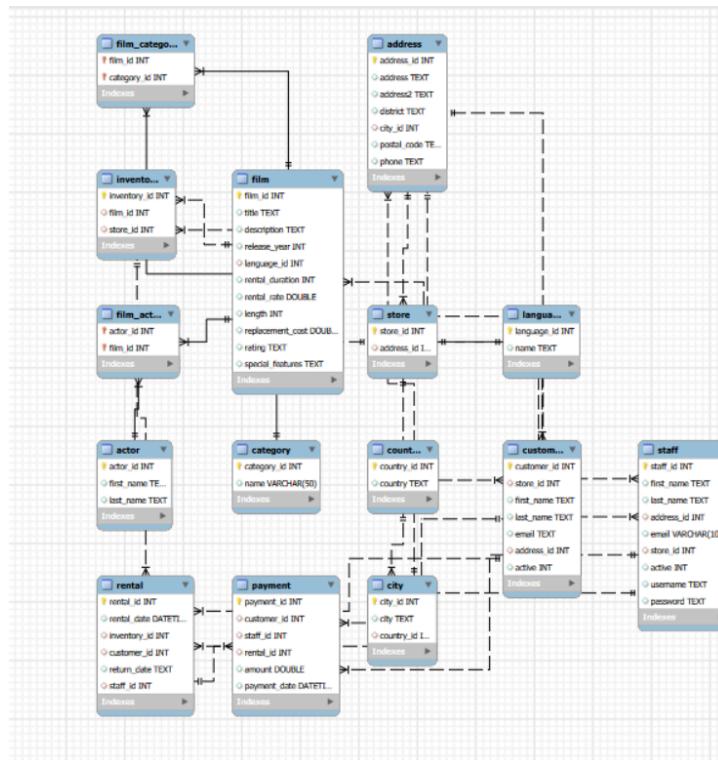


Title Page

- **Title:** DB Assignment 4
- **Your Name:** Joseph Pepe
- **Date:** Nov. 4rd, 2025

This assignment uses a complex schema consisting of over a dozen tables that is meant to store data about movies and rentals. Below is the EER diagram for the table.



Query 1: We start by selecting the name and the average film length. From there, We need to join from film -> category, then we simply group by category and ordering by category will get us the categories in alphabetical order.

	name	avg_length
▶	Action	111.6094
	Animation	111.0152
	Children	109.8000
	Classics	111.6667
	Comedy	115.8276

Query 2: Begin by selecting category name and use the function avg to get the average length of films. We will start in the film table then need to move from film -> film_category -> category. Finally, we will group by category since that is what the problem says and we can order by the name without including anything else since it auto orders by ascending. Now this query is a little different - we need the highest and lowest average length. One of the ways to do this is to make two separate queries and make a union of them so that they display together, which is what I do.

	name	avg_length
▶	Sci-Fi	108.1967
	Sports	128.2027

Query 3: This query has a lot going on. First we want to select the customer's name, since that is what the problem wants this to return. Next, we have to join from customer -> rental -> inventory -> film -> film_category -> category and we want to make sure the category is action, so we include that in the where clause. Next, we want to make sure that we are disallowing the comedy and classics, so we say 'not' then open a subquery and basically put in the exact same query but we select classics and comedy, which will disallow those categories from the main query.

	first_name	last_name
▶	DONNA	THOMPSON
	RUBY	WASHINGTON
	AMBER	DIXON
	JOANN	GARDNER
	DOLORES	WAGNER

Query 4: Which actor has appeared in the most amount of English lang. Films
 First, we select the actor name, and we also want to select the count of the language id, because when we are done culling out things this will tell us the total number of english language films the actor was in. Next, we have to join from actor -> film_actor -> film ->language, and include a where to make sure the language is english. Since we only want one actor we can just order by desc and limit it to one.

	first_name	last_name	num
▶	GINA	DEGENERES	42

Query 5: We solve this by only selecting the count of distinct film IDs, since we want the movies to be unique. Next we join from inventory -> rental to staff and add a where to make sure that the staff is Mike. We don't want this to be the only parameter though, so we add an and and make sure the difference between the rental and return is exactly 10 days.

	distinct_movies
▶	47

Query 6: First we select the actor first and last name. Next we join to go from actor -> film_actor. Next we need to open a subquery to get the film with the largest amount of actors, and in the main query we can list all the actors by name since we are only looking at actors from the largest film.

	first_name	last_name
▶	JULIA	BARRYMORE
	VAL	BOLGER
	SCARLETT	DAMON
	LUCILLE	DEE
	WOODY	HOFFMAN