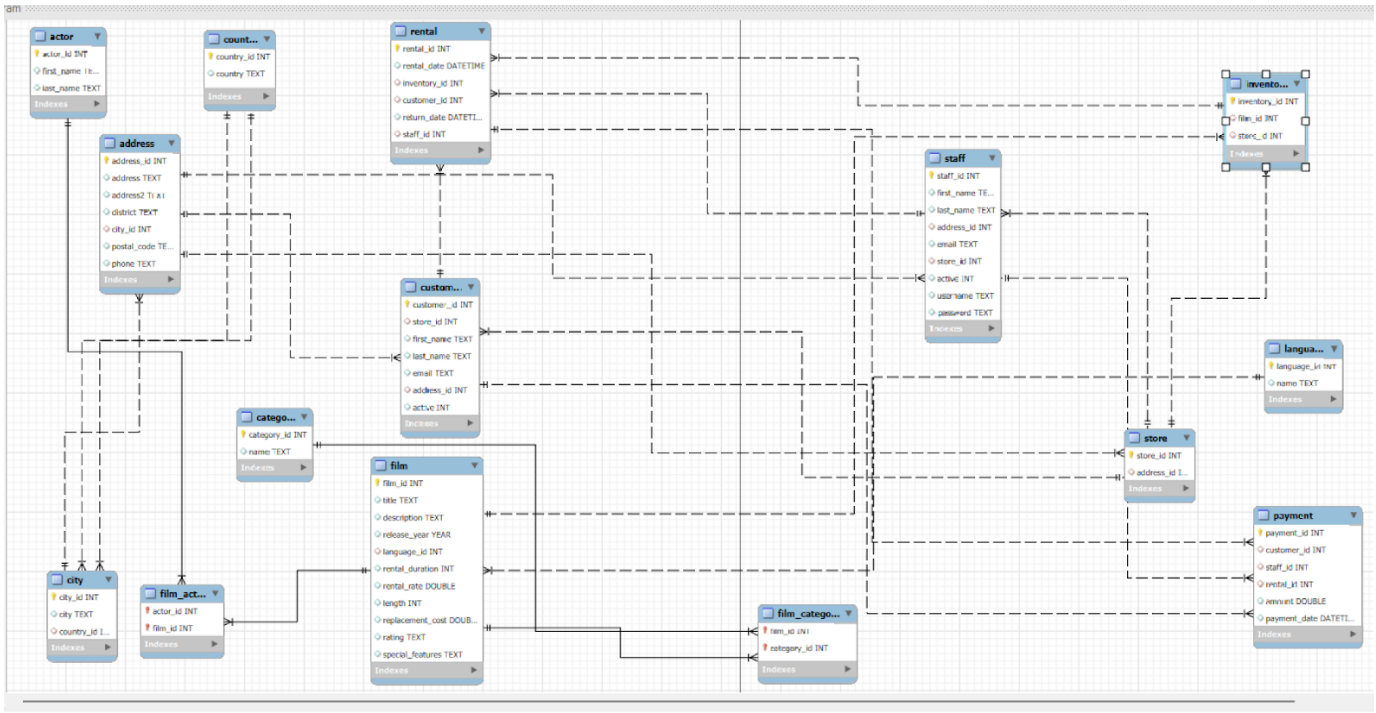
Title: DB Assignment 4

Name: Daniela Cruz Perez

Date: 11/4/2025

ER Diagram:



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

Explanation: The query inner joins the tables category, film_category, and film. It aggregates the average film length for each category. Finally, the query is ordered by category name, alphabetically by the order clause.

```
SQL
select category.name, avg(film.length) as average_length
```

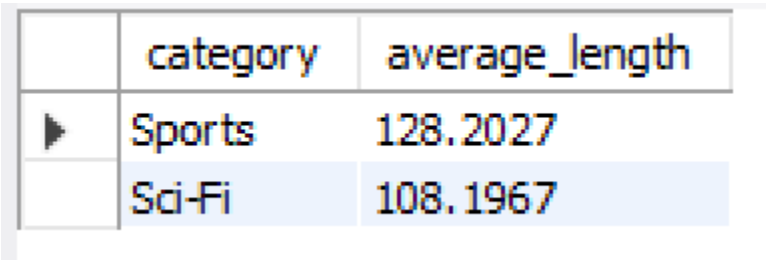
```
from category join film_category on category.category_id =
film_category.category_id
join film on film.film_id = film_category.film_id
group by category.name
order by category.name;
```

name	average_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

Problem 2: Which categories have the longest and shortest average film lengths?
Explanation: The query uses 3 CTEs, the first, cat_avg , first initialize the average film length for each category. The CTE Max_l selects the highest length in cat_avg and Min_l selects the lowest length in cat_avg. In the main query, there are two selects unionized, one where the cat_avg average_length is set to the max_l which also retrieves the film with this attribute and the other for the min_l.

```
SQL
With cat_avg as  (
select category.name as category, avg(film.length) as average_length
from category join film_category on category.category_id =
film_category.category_id
join film on film.film_id = film_category.film_id
```

```
group by category.name
order by category.name
),
Max_l as (
select max(average_length) as highest_legnth
from cat_avg
),
Min_l as (
select min(average_length) as lowest_length
from cat_avg
)
select ca.category, ca.average_length
from cat_avg ca
join Max_l ml on ca.average_length = ml.highest_legnth
union
select ca.category, ca.average_length
from cat_avg ca
join Min_l minl on ca.average_length = minl.lowest_length;
```



	category	average_length
▶	Sports	128.2027
	Sci-Fi	108.1967

Problem 3: Which customers have rented action but not comedy or classic movies?
Explanation: The query first selects the customers first and last name and ids (to ensure distinct customers) who have rented an action movie from the table inner join by rental, inventory, film, film_category, and category using a where clause. The query then excludes those in the subquery who have rented comedy or classic movies by the except clause.

SQL
```
select customer.first_name, customer.last_name, customer.customer_id
from customer join rental on customer.customer_id = rental.customer_id
            join inventory on rental.inventory_id = inventory.inventory_id
            join film on inventory.film_id = film.film_id
            join film_category on film_category.film_id = film.film_id
            join category on film_category.category_id = category.category_id
where category.name = 'Action'
```

```
except
select distinct customer.first_name, customer.last_name, customer.customer_id
from customer join rental on customer.customer_id = rental.customer_id
            join inventory on rental.inventory_id = inventory.inventory_id
            join film on inventory.film_id = film.film_id
            join film_category on film_category.film_id = film.film_id
            join category on film_category.category_id = category.category_id
where category.name = 'Comedy' or category.name = 'Classics';
```

first_name	last_name	customer_id
LAWRENCE	LAWTON	361
MATTHEW	MAHAN	323
TOM	MILNER	452
JO	FOWLER	250
SCOTT	SHELLEY	330
EDWIN	BURK	432
JOANN	GARDNER	164
DONNA	THOMPSON	17
DON	BONE	433
JUAN	FRALEY	350
DOLORES	WAGNER	171
MICHEAL	FORMAN	445
AMBER	DIXON	139
MELINDA	FERNANDEZ	223
CONSTANCE	REID	232
RUBY	WASHINGTON	90
GINA	WILLIAMSON	213

Problem 4: Which actor has appeared in the most English-language movies?
The query is joined by the tables actor, film_actor, film, and language. The query selects the actors id (their distinctive value), name, and counts how many films each actor has been in filtered by the language is set to english by the where clause, stored as number_of english films. The table is then ordered by count of films in descending order limited to 1.

```
SQL
select actor.actor_id, first_name, actor.last_name, count(film.film_id) as
number_of_English_films
```

```
from actor join film_actor on actor.actor_id = film_actor.actor_id
            join film on film_actor.film_id = film.film_id
            join language on film.language_id = language.language_id
where language.name = 'English'
group by actor.actor_id, actor.first_name, actor.last_name
order by count(film.film_id) desc
limit 1;
```

	actor_id	first_name	last_name	number_of_English_films
▶	107	GINA	DEGENERES	42

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

Explanation: This query is joined by tables film, inventory, rental, staff, and store. The query selects the distinct count of films filtered by the rental duration is equal to 10, this is aggregated by the return date and rental date using the datediff(). It is also filtered by the staff name Mike Hillyer using the where clause .

```
SQL
select distinct count(film.title)
number_of_films_rented_for_10_days_where_Mike_works
from film join inventory on film.film_id = inventory.film_id
join rental on inventory.inventory_id = rental.inventory_id
join staff on rental.staff_id = staff.staff_id
join store on store.store_id = staff.store_id
where DATEDIFF(rental.return_date,rental.rental_date) = 10 and staff.first_name
= 'Mike' and staff.last_name = 'Hillyer'
group by store.store_id;
```

	number_of_films_rented_for_10_days_where_Mike_works
▶	49

Problem 6: Alphabetically list actors who appeared in the movie with the largest cast of actors.

Explanation: The query is sectioned into two CTEs, each selecting from a table joined by actor, film_actor, and film. The first CTE, film_and_highest_cast, finds the film with the highest number

of actors by counting the number of film actors IDsin each movie. The greatest number is found using the having clause where the count is greater than or equal to the other counts. The second CTE, actors_in_film, selects the actors' first and last names for each film. The final query, then selects the movie found in the first CTE and its corresponding actors from the second CTE through joining them by film title. The final query, orders the names alphabetically by the actors' first name.

```
SQL
With film_and_highest_cast as  (
select film.title as movie_with_largest_cast, count(film_actor.actor_id) as
number_of_actors
from actor join film_actor on actor.actor_id = film_actor.actor_id
            join film on film_actor.film_id = film.film_id
group by movie_with_largest_cast
having count(film_actor.actor_id) >= all (
    select count(film_actor.actor_id)
    from actor join film_actor on actor.actor_id = film_actor.actor_id
    join film on film_actor.film_id = film.film_id
    group by film.title
)
),
actors_in_film as (
select film.title as title, actor.first_name as first_name, actor.last_name as
last_name
from actor join film_actor on actor.actor_id = film_actor.actor_id
join film on film_actor.film_id = film.film_id
group by film.title, actor.first_name, actor.last_name
)
select fc.movie_with_largest_cast, ac.first_name, ac.last_name
from film_and_highest_cast fc join actors_in_film ac on
fc.movie_with_largest_cast = ac.title
order by ac.first_name;
```

movie_with_largest_cast	first_name	last_name	actor_id
LAMBS CINCINATTI	BURT	POSEY	75
LAMBS CINCINATTI	CAMERON	ZELLWEGER	111
LAMBS CINCINATTI	CHRISTIAN	NEESON	61
LAMBS CINCINATTI	FAY	WINSLET	147
LAMBS CINCINATTI	JAYNE	NOLTE	150
LAMBS CINCINATTI	JULIA	BARRYMORE	47
LAMBS CINCINATTI	JULIA	ZELLWEGER	186
LAMBS CINCINATTI	LUCILLE	DEE	138
LAMBS CINCINATTI	MENA	TEMPLE	53
LAMBS CINCINATTI	MENA	HOPPER	170
LAMBS CINCINATTI	REESE	KILMER	45
LAMBS CINCINATTI	SCARLETT	DAMON	81
LAMBS CINCINATTI	VAL	BOLGER	37
LAMBS CINCINATTI	WALTER	TORN	102
LAMBS CINCINATTI	WOODY	HOFFMAN	28