Title Page

Title: DB Assignment 4
Your Name: Ryan Farley
Date: 10.31.24

I had some corruption issues, mysql kept saying errors and bugs, I just wrote queries and had to assume my logic was sound as I could not test 5 / 6.

Constraints:
I used a series of alters to add constraints to the provided tables.

- All tables must have the indicated primary key, foreign key and unique constraints
  - ALTER TABLE payment
  - ADD FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
  - ADD FOREIGN KEY (staff_id) REFERENCES staff(staff_id);
  -
  - ALTER TABLE rental
  - ADD FOREIGN KEY (inventory_id) REFERENCES inventory(inventory_id),
  - ADD FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
  - ADD FOREIGN KEY (staff_id) REFERENCES staff(staff_id);

- Category names come from the set {Animation, Comedy, Family, Foreign, Sci-Fi, Travel, Children, Drama, Horror, Action, Classics, Games, New, Documentary, Sports, Music}
  - ALTER TABLE Category
  - ADD CHECK (name IN ('Animation', 'Comedy', 'Family', 'Foreign', 'Sci-Fi', 'Travel', 'Children', 'Drama', 'Horror', 'Action', 'Classics', 'Games', 'New', 'Documentary', 'Sports', 'Music'));

- A film's special_features attribute comes from the set {Behind the Scenes, Commentaries, Deleted Scenes, Trailers}
  -
  - ALTER TABLE film
  - ADD CHECK (special_features IN ('Behind the Scenes', 'Commentaries', 'Deleted Scenes', 'Trailers'));

- All dates must be valid
  - ALTER TABLE film ADD CONSTRAINT check_release_year CHECK (DATE(CONCAT(release_year, '-01-01')) IS NOT NULL);

- Active is from the set {0,1} where 1 means active and 0 inactive
  - ALTER TABLE customer ADD CONSTRAINT check_active CHECK (active IN (0, 1));

- Rental duration is a positive number of days between 2 and 8
  - ALTER TABLE film ADD CHECK (rental_duration BETWEEN 2 AND 8);

- Rental rate per day is between 0.99 and 6.99
  -
  - ALTER TABLE film ADD CHECK (rental_rate BETWEEN 0.99 AND 6.99);

- Film length is between 30 and 200 minutes
  - ALTER TABLE film ADD CHECK (length BETWEEN 30 AND 200);

- Ratings are {PG, G, NC-17, PG-13, R}
  - ALTER TABLE film ADD CHECK (rating IN ('PG', 'G', 'NC-17', 'PG-13', 'R'));

- Replacement cost is between 5.00 and 100.00
  -
  - ALTER TABLE film ADD CHECK (replacement_cost BETWEEN 5.00 AND 100.00);

- Amount should be >= 0
  - ALTER TABLE payment ADD CHECK (amount >= 0);

questions/queries

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

```
SELECT category.name AS category_name, AVG(film.length) AS average_length
FROM film_category
INNER JOIN film ON film_category.film_id = film.film_id
INNER JOIN category ON film_category.category_id = category.category_id
GROUP BY category.name
ORDER BY category.name;
```

2. Which categories have the longest and shortest average film lengths?

```
WITH CategoryAvgLength AS (
  SELECT category.name AS category_name, AVG(film.length) AS avg_length
  FROM category
  LEFT JOIN film_category ON category.category_id = film_category.category_id
  LEFT JOIN film ON film_category.film_id = film.film_id
  GROUP BY category_name
)
SELECT category_name, avg_length
FROM CategoryAvgLength
WHERE avg_length = (SELECT MAX(avg_length) FROM CategoryAvgLength)
  OR avg_length = (SELECT MIN(avg_length) FROM CategoryAvgLength);
```

3. Which customers have rented action but not comedy or classic movies?

```
SELECT DISTINCT customer.customer_id, customer.first_name, customer.last_name
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.film_id = film_category.film_id
JOIN category ON film_category.category_id = category.category_id
WHERE category.name = 'Action'
AND customer.customer_id NOT IN (
  SELECT DISTINCT customer2.customer_id
  FROM customer AS customer2
  JOIN rental AS rental2 ON customer2.customer_id = rental2.customer_id
  JOIN inventory AS inventory2 ON rental2.inventory_id = inventory2.inventory_id
  JOIN film AS film2 ON inventory2.film_id = film2.film_id
```

```
  JOIN film_category AS film_category2 ON film2.film_id = film_category2.film_id
  JOIN category AS category2 ON film_category2.category_id = category2.category_id
  WHERE category2.name IN ('Comedy', 'Classics')
);
```

4. Which actor has appeared in the most English-language movies?

```
SELECT actor.actor_id, actor.first_name, actor.last_name, COUNT(*) AS movie_count
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
ORDER BY movie_count DESC
LIMIT 1;
```

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

```
SELECT COUNT(DISTINCT rental.inventory_id) AS distinct_movie_count
FROM rental
JOIN staff ON rental.staff_id = staff.staff_id
WHERE rental.return_date = DATE_ADD(rental.rental_date, INTERVAL 10 DAY)
AND staff.first_name = 'Mike';
```

This one worked

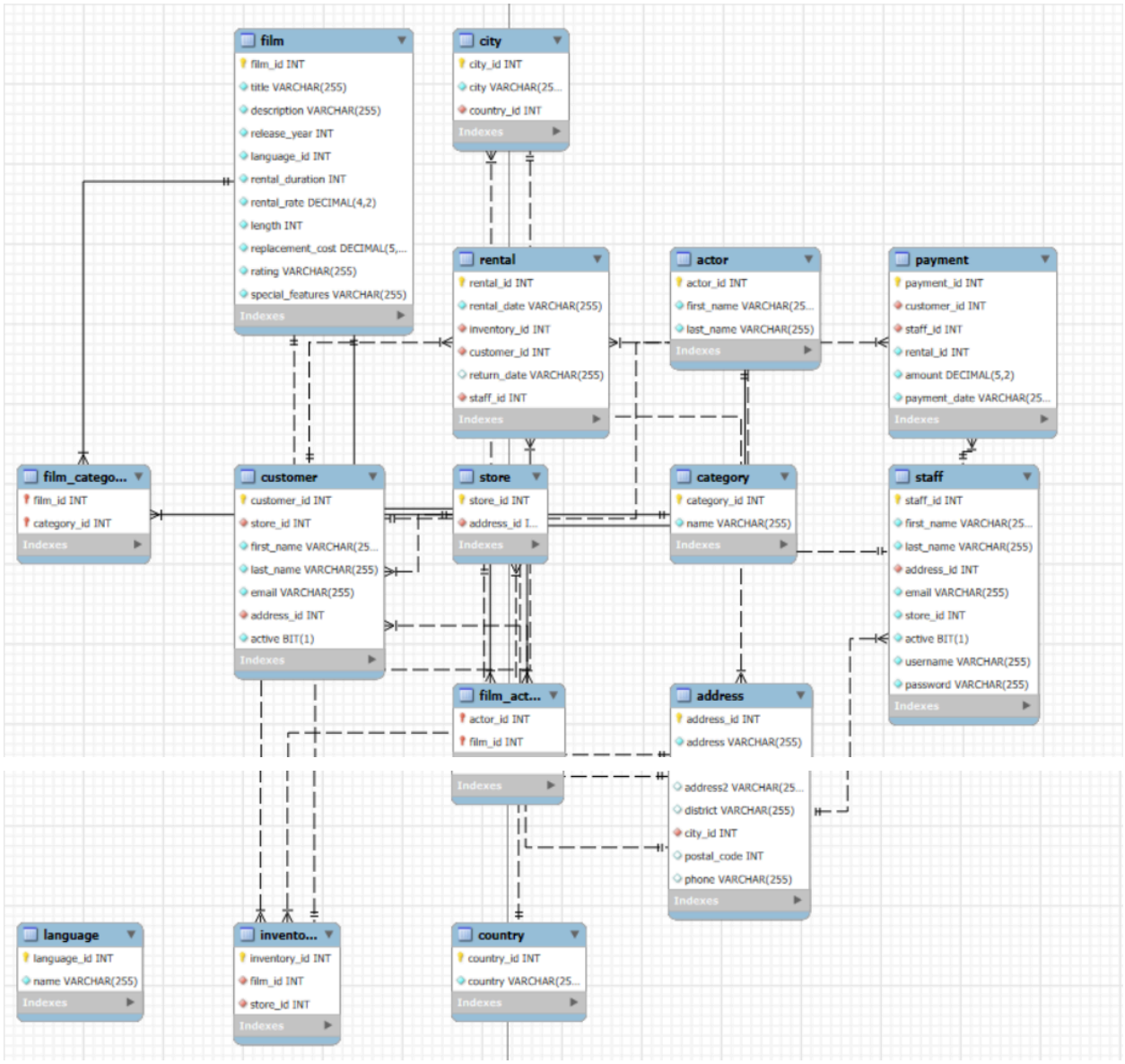
distinct_movie_count
0

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

```
WITH ActorMovieCast AS (
  SELECT f.film_id, a.actor_id, a.first_name, a.last_name
  FROM film f
```

```
    JOIN film_actor fa ON f.film_id = fa.film_id
    JOIN actor a ON fa.actor_id = a.actor_id
)
SELECT amc.actor_id, amc.first_name, amc.last_name
FROM ActorMovieCast amc
WHERE amc.film_id = (
    SELECT film_id
    FROM (
        SELECT film_id, COUNT(actor_id) AS cast_count
        FROM film_actor
        GROUP BY film_id
        ORDER BY cast_count DESC
        LIMIT 1
    ) AS largest_cast
)
ORDER BY amc.first_name, amc.last_name;
```

Model



2 screenshots used