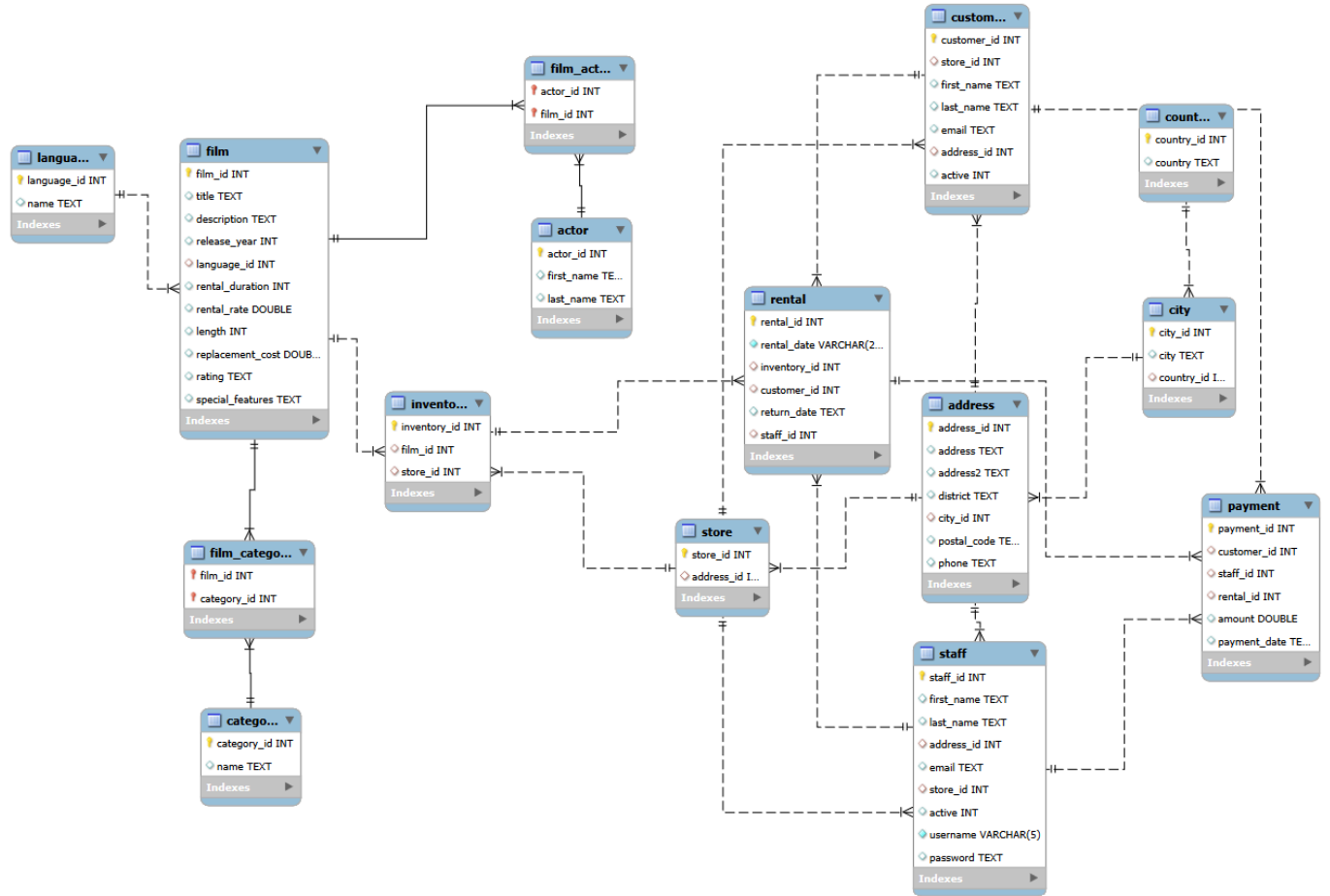


DB Assignment 4

Brennan Duff

October 31, 2024



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

a. Find the average length of films and group by category name.

b. Query:

i. `SELECT c.name, AVG(f.length) AS avg_length`

ii. `FROM film f`

iii. `JOIN film_category fc ON f.film_id = fc.film_id`

iv. `JOIN category c ON fc.category_id = c.category_id`

v. `GROUP BY c.name`

vi. `ORDER BY c.name;`

c. Screenshot:

i.

	name	avg_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

d. Explanation:

i. This query retrieves the average length of films in each category, ordered alphabetically by category name. It joins film with film_category to match films with their categories and then joins with category to get the category names.

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

a. Find the min and max of average length and group by category name.

b. Query:

- i. WITH avg_lengths AS (
- ii. SELECT c.name, AVG(f.length) AS avg_length
- iii. FROM film f
- iv. JOIN film_category fc ON f.film_id = fc.film_id
- v. JOIN category c ON fc.category_id = c.category_id
- vi. GROUP BY c.name)
- vii. SELECT name, avg_length
- viii. FROM avg_lengths
- ix. WHERE avg_length = (SELECT MAX(avg_length) FROM
avg_lengths)
- x. OR avg_length = (SELECT MIN(avg_length) FROM
avg_lengths);

c. Screenshot:

i.

	name	avg_length
▶	Sci-Fi	108.1967
	Sports	128.2027

d. Explanation:

- i. This query retrieves the categories with the longest and shortest average film lengths. It first calculates the average length for each category by joining film with film_category to link films with categories, then joins with category to get category names. Finally,

it selects the categories with the maximum and minimum average lengths.

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

- a. Find customer names that have rented action movies and exclude ones that rented either comedy or classic movies.

b. Query:

```
i.   SELECT DISTINCT cu.customer_id, cu.first_name, cu.last_name
ii.  FROM customer cu
iii. JOIN rental r ON cu.customer_id = r.customer_id
iv.  JOIN inventory i ON r.inventory_id = i.inventory_id
v.   JOIN film_category fc ON i.film_id = fc.film_id
vi.  JOIN category c ON fc.category_id = c.category_id
vii. WHERE c.name = 'Action'
viii. AND cu.customer_id NOT IN (
ix.   SELECT cu.customer_id
x.    FROM customer cu
xi.   JOIN rental r ON cu.customer_id = r.customer_id
xii.  JOIN inventory i ON r.inventory_id = i.inventory_id
xiii. JOIN film_category fc ON i.film_id = fc.film_id
xiv.  JOIN category c ON fc.category_id = c.category_id
xv.   WHERE c.name IN ('Comedy', 'Classic')
xvi. );
```

c. Screenshot:

i.

customer_id	first_name	last_name
2	PATRICIA	JOHNSON
8	SUSAN	WILSON
11	LISA	ANDERSON
17	DONNA	THOMPSON
43	CHRISTINE	ROBERTS
49	JOYCE	EDWARDS
60	MILDRED	BAILEY
69	JUDY	GRAY
71	KATHY	JAMES
73	BEVERLY	BROOKS
83	LOUISE	JENKINS
90	RUBY	WASHINGTON
111	CARMEN	OWENS
115	WENDY	HARRISON
123	SHANNON	FREEMAN
124	SHEILA	WELLS
126	ELLEN	SIMPSON
136	ANITA	MORALES
139	AMBER	DIXON
142	APRIL	BURNS
164	JOANN	GARDNER
171	DOLORES	WAGNER
178	MARION	SNYDER
183	IDA	ANDREWS
185	ROBERTA	HARPER
203	TARA	RYAN
212	WILMA	RICHARDS
213	GINA	WILLIAMSON
217	AGNES	BISHOP
223	MELINDA	FERNANDEZ
232	CONSTANCE	REID
237	TANYA	GILBERT
238	NELLIE	GARRETT
239	MINNIE	ROMERO
242	GLENDA	FRAZIER
246	MARIAN	MENDOZA
250	JO	FOWLER
257	MARSHA	DOUGLAS
262	PATSY	DAVIDSON
266	NORA	HERRERA

282	JENNY	CASTRO
283	FELICIA	SUTTON
288	BOBBIE	CRAIG
292	MISTY	LAMBERT
300	JOHN	FARNSWORTH
304	DAVID	ROYAL
323	MATTHEW	MAHAN
330	SCOTT	SHELLEY
336	JOSHUA	MARK
341	PETER	MENARD
350	JUAN	FRALEY
355	TERRY	GRISSOM
356	GERALD	FULTZ
360	RALPH	MADRIGAL
361	LAWRENCE	LAWTON
370	WAYNE	TRUONG
382	VICTOR	BARKLEY
386	TODD	TAN
397	JIMMY	SCHRADER
399	DANNY	ISOM
405	LEONARD	SCHOFIELD
407	DALE	RATCLIFF
419	CHAD	CARBONE
423	ALFRED	CASILLAS
425	FRANCIS	SIKES
426	BRADLEY	MOTLEY
432	EDWIN	BURK
433	DON	BONE
439	ALEXANDER	FENNELL
440	BERNARD	COLBY
445	MICHEAL	FORMAN
450	JAY	ROBB
451	JIM	REA
452	TOM	MILNER
456	RONNIE	RICKETTS
463	DARRELL	POWER
475	PEDRO	CHESTNUT
482	MAURICE	CRAWLEY
487	HECTOR	POINDEXTER
493	BRENT	HARKINS

500	REGINALD	KINDER
511	CHESTER	BENNER
516	ELMER	NOE
527	CORY	MEEHAN
529	ERIK	GUILLEN
534	CHRISTIAN	JUNG
545	JULIO	NOLAND
585	PERRY	SWAFFORD
587	SERGIO	STANFIELD
588	MARION	OCAMPO
590	SETH	HANNON

d. Explanation:

- i. This query retrieves the names of customers who have rented action movies but have not rented comedy or classic movies. It joins customer with rental and inventory to match customers with their rentals, then joins with film_category to filter by category. An additional subquery is used to exclude customers who rented movies in the comedy or classic categories.

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

a. Find the actor that is in the most movies with the English Language.

b. Query:

- i. `SELECT a.actor_id, a.first_name, a.last_name, COUNT(*) AS
film_count`
- ii. `FROM actor a`
- iii. `JOIN film_actor fa ON a.actor_id = fa.actor_id`
- iv. `JOIN film f ON fa.film_id = f.film_id`
- v. `JOIN language l ON f.language_id = l.language_id`
- vi. `WHERE l.name = 'English'`
- vii. `GROUP BY a.actor_id, a.first_name, a.last_name`
- viii. `ORDER BY film_count DESC`
- ix. `LIMIT 1;`

c. Screenshot:

i.

	actor_id	first_name	last_name	film_count
▶	107	GINA	DEGENERES	42

d. Explanation:

- i. This query retrieves the name of the actor who has appeared in the most English-language movies. It joins actor with film_actor to match actors with their films, then joins with film and language to filter by English-language films. The result is grouped and ordered to find the actor with the highest count of English-language movies.

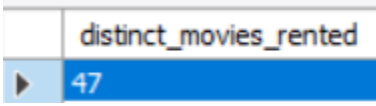
5. How many distinct movies were rented for exactly 10 days from the store where Mike works?

a. Find the total number of distinct movies rented for 10 days where Mike works. Note: the query invalidates restraint 6.

b. Query:

- i. `SELECT COUNT(DISTINCT i.film_id) AS distinct_movies_rented`
- ii. `FROM rental r`
- iii. `JOIN inventory i ON r.inventory_id = i.inventory_id`
- iv. `JOIN staff s ON r.staff_id = s.staff_id`
- v. `WHERE s.first_name = 'Mike'`
- vi. `AND DATEDIFF(r.return_date, r.rental_date) = 10;`

c. Screenshot:

i. 

distinct_movies_rented
47

d. Explanation:

- i. This query retrieves the number of distinct movies rented for exactly 10 days from the store where Mike works. It joins rental with inventory to match rentals with films, then joins with staff to filter by rentals handled by Mike. It also calculates the rental duration by subtracting rental_date from return_date.

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

a. Find the largest cast of actors in a movie, then list them alphabetically.

b. Query:

```
i.   WITH movie_cast_count AS (  
ii.      SELECT fa.film_id, COUNT(fa.actor_id) AS actor_count  
iii.     FROM film_actor fa  
iv.      GROUP BY fa.film_id  
v.     ),  
vi.   largest_cast_movie AS (  
vii.    SELECT film_id  
viii.   FROM movie_cast_count  
ix.     WHERE actor_count = (SELECT MAX(actor_count) FROM  
      movie_cast_count)  
x.     )  
xi.    SELECT a.first_name, a.last_name  
xii.   FROM actor a  
xiii.  JOIN film_actor fa ON a.actor_id = fa.actor_id  
xiv.   JOIN largest_cast_movie lcm ON fa.film_id = lcm.film_id  
xv.    ORDER BY a.first_name, a.last_name;
```

c. Screenshot:

	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

i.

d. Explanation:

- i. This query retrieves the names of actors who appeared in the movie with the largest cast, ordered alphabetically. It first calculates the number of actors per movie using film_actor, then identifies the movie with the largest cast. Finally, it joins actor with film_actor to list the actors in that movie, sorted alphabetically by name.