40/40

```
/*************************************************************************
 * NAME: Samuel Sovi
 * CLASS: CPSC 321 Section 1
 * DATE: 11/22/22
 * HOMEWORK: HW-6
 * DESCRIPTION: Queries practice
 *************************************************************************/


-- TODO: Implement the queries for HW6 below. Include comments as
--       appropriate.


-- Quesiton 1:

\! echo 'Question 1:';
SELECT COUNT(*), MIN(length), MAX(length), AVG(length), COUNT(DISTINCT special_f
eatures)
FROM film;


-- Question 2:

\! echo 'Question 2:';
SELECT rating, COUNT(*), ROUND(AVG(length), 2)
FROM film
GROUP BY rating
ORDER BY AVG(length) DESC;

-- Question 3:

\! echo 'Question 3:';
SELECT f.rating, COUNT(*)
FROM film f JOIN film_actor fa on (f.film_id = fa.film_id)
GROUP BY f.rating
ORDER BY COUNT(*) DESC;

-- Question 4:

\! echo 'Question 4:';
SELECT c.name, COUNT(*), ROUND(MIN(f.rental_rate), 2), ROUND(MAX(f.rental_rate),
 2), ROUND(AVG(f.rental_rate), 2), ROUND(MIN(f.replacement_cost), 2), ROUND(MAX(
f.replacement_cost), 2), ROUND(AVG(f.replacement_cost), 2)
FROM film f JOIN film_category fc ON (f.film_id = fc.film_id) JOIN category c ON
 (c.category_id = fc.category_id)
GROUP BY c.category_id
ORDER BY c.name ASC;

-- Question 5:

\! echo 'Question 5:';
SELECT f.rating, COUNT(*)
FROM rental r JOIN inventory i ON (r.inventory_id = i.inventory_id) JOIN film f
ON (i.film_id = f.film_id) JOIN film_category fc ON (f.film_id = fc.film_id) JOI
N category c ON (c.category_id = fc.category_id)
WHERE c.name = "Horror" AND i.store_id = 1
GROUP BY f.rating
ORDER BY COUNT(*) DESC;
```

```
-- Question 6:

\! echo 'Question 6:';
SELECT f.title, f.rating, COUNT(*)
FROM rental r JOIN inventory i ON (r.inventory_id = i.inventory_id) JOIN film f
ON (i.film_id = f.film_id) JOIN film_category fc ON (f.film_id = fc.film_id) JOI
N category c ON (c.category_id = fc.category_id)
WHERE c.name = "Horror"
GROUP BY f.film_id
ORDER BY COUNT(*) DESC, f.rating DESC -- note: the f.rating in ORDER BY is here
to match the example
LIMIT 15;


-- Question 7:

\! echo 'Question 7:';
SELECT f.title, COUNT(*)
FROM rental r JOIN inventory i ON (r.inventory_id = i.inventory_id) JOIN film f
ON (i.film_id = f.film_id) JOIN film_category fc ON (f.film_id = fc.film_id) JOI
N category c ON (c.category_id = fc.category_id)
WHERE f.rating = "G" AND c.name = "Action"
GROUP BY f.film_id
HAVING COUNT(*) >= 15
ORDER BY COUNT(*) DESC;


-- Question 8:

\! echo 'Question 8:';
SELECT a.first_name, a.last_name, COUNT(*)
FROM film f JOIN film_category fc ON (f.film_id = fc.film_id) JOIN category c ON
 (c.category_id = fc.category_id) JOIN film_actor fa ON (f.film_id = fa.film_id)
 JOIN actor a ON (a.actor_id = fa.actor_id)
WHERE c.name = "Horror"
GROUP BY a.actor_id
HAVING COUNT(*) >= 4
ORDER BY COUNT(*) DESC, a.last_name, a.first_name;

-- Question 9:

\! echo 'Question 9:';
SELECT c.first_name, c.last_name, COUNT(*)
FROM rental r JOIN inventory i ON (r.inventory_id = i.inventory_id) JOIN film f
ON (i.film_id = f.film_id) JOIN customer c ON (r.customer_id = c.customer_id)
WHERE f.rating = "PG"
GROUP BY c.customer_id
HAVING COUNT(*) > 10
ORDER BY COUNT(*) DESC, c.last_name, c.first_name;

-- Question 10:

\! echo 'Question 10:';
\! echo '';
\! echo 'Purpose:';
\! echo 'Find number of movie rentals, number of distinct customers, total reven
ue generated and average replacment cost of each movie category ordered by most
revenue to least';
\! echo '';
```

```
SELECT c.name, ROUND(SUM(p.amount), 2), COUNT(*), COUNT(DISTINCT cu.customer_id)
, ROUND(AVG(f.replacement_cost), 2)
FROM payment p JOIN rental r ON (p.rental_id = r.rental_id) JOIN inventory i ON
(r.inventory_id = i.inventory_id) JOIN film f ON (i.film_id = f.film_id) JOIN fi
lm_category fc ON (f.film_id = fc.film_id) JOIN category c ON (c.category_id = f
c.category_id) JOIN customer cu ON (r.customer_id = cu.customer_id)
GROUP BY c.category_id
ORDER BY SUM(p.amount) DESC;

\! echo '';
\! echo 'Why this is interesting:';
\! echo 'As we can see, rental count does not necessarily correspond with the to
tal revenue generated from each rental';
\! echo 'This is because some genres have better sale prices than others and can
 generate more money per rental';
\! echo 'This query would be of interest to a marketer because it shows the tota
l revenue as well as the number of sales and number of unique customers';
\! echo 'This allows a shop to stock up on movies based on whichever of these me
trics meets their desires more';
\! echo 'They have the option of being more customer-oriented or focusing more o
n what generates the most revenue';
\! echo 'Furthermore, they can see the replacement cost to gauge how worth it wo
uld be to replace a stolen film';
```

HW-6 Samuel Sovi

Question 1:

COUNT(*)	MIN(length)	MAX(length)	AVG(length)	COUNT(DISTINCT special_features)
1000	46	185	115.2720	15

1 row in set (0.000 sec)

Question 2:

rating	COUNT(*)	ROUND(AVG(length), 2)
PG-13	223	120.44
R	195	118.66
NC-17	210	113.23
PG	194	112.01
G	178	111.05

5 rows in set (0.000 sec)

Question 3:

rating	COUNT(*)
PG-13	1184
PG	1143
NC-17	1128
R	1031
G	976

5 rows in set (0.000 sec)

Question 4:

name	COUNT(*)	ROUND(MIN(f.rental_rate), 2)	ROUND(MAX(f.rental_rate), 2)	ROUND(AVG(f.rental_rate), 2)	ROUND(MIN(f.replacement_cost), 2)	ROUND(MAX(f.replacement_cost), 2)	ROUND(AVG(f.replacement_cost), 2)
Action	64	0.99	4.99	2.65	9.99	29.99	20.91
Animation	66	0.99	4.99	2.81	9.99	29.99	20.13
Children	60	0.99	4.99	2.89	9.99	29.99	20.06
Classics	57	0.99	4.99	2.74	10.99	29.99	21.01
Comedy	58	0.99	4.99	3.16	9.99	28.99	19.02
Documentary	68	0.99	4.99	2.67	9.99	29.99	19.62
Drama	62	0.99	4.99	3.02	9.99	29.99	21.09
Family	69	0.99	4.99	2.76	9.99	29.99	19.73
Foreign	73	0.99	4.99	3.10	9.99	29.99	18.65
Games	61	0.99	4.99	3.25	9.99	29.99	20.29
Horror	56	0.99	4.99	3.03	10.99	29.99	19.87
Music	51	0.99	4.99	2.95	10.99	29.99	19.44
New	63	0.99	4.99	3.12	9.99	29.99	19.42
Sci-Fi	61	0.99	4.99	3.22	9.99	29.99	21.15
Sports	74	0.99	4.99	3.13	9.99	29.99	20.40
Travel	57	0.99	4.99	3.24	9.99	29.99	19.03

16 rows in set (0.000 sec)

Question 5:

rating	COUNT(*)
R	119
PG-13	102
PG	71
G	64
NC-17	30

5 rows in set (0.000 sec)

Question 6:

title	rating	COUNT(*)
PULP BEVERLY	G	30
FAMILY SWEET	R	29
SWARM GOLD	PG-13	27
STREETCAR INTENTIONS	R	25
SLEEPING SUSPECTS	PG-13	24
ARACHNOPHOBIA ROLLERCOASTER	PG-13	24
LOLA AGENT	PG	24
FREDDY STORM	NC-17	23
YENTL IDAHO	R	23
KARATE MOON	PG-13	23
PATTON INTERVIEW	PG	23
CARRIE BUNCH	PG	23
AFFAIR PREJUDICE	G	23
HIGH ENCINO	R	22
SINNERS ATLANTIS	PG-13	22

15 rows in set (0.000 sec)

Question 7:

title	COUNT(*)
PRIMARY GLASS	27
HILLS NEIGHBORS	25
EASY GLADIATOR	23
TRUMAN CRAZY	23
EXCITEMENT EVE	21
BRIDE INTRIGUE	19
BAREFOOT MANCHURIAN	18
DARN FORRESTER	18
WATERFRONT DELIVERANCE	17
MIDNIGHT WESTWARD	17
RINGS HEARTBREAKERS	15
LUST LOCK	15

12 rows in set (0.000 sec)

Question 8:

first_name	last_name	COUNT(*)
JULIA	MCQUEEN	7
TOM	MIRANDA	6
VIVIEN	BERGEN	5
HENRY	BERRY	5
KARL	BERRY	4
JOHNNY	CAGE	4
RIP	CRAWFORD	4
PENELOPE	CRONYN	4
JUDE	CRUISE	4
ED	GUINESS	4
WHOOPI	HURT	4
GRETA	MALDEN	4
PENELOPE	PINKETT	4
BURT	POSEY	4
KENNETH	TORN	4
WALTER	TORN	4

16 rows in set (0.000 sec)

Question 9:

first_name	last_name	COUNT(*)
DUSTIN	GILLETTE	13
TAMMY	SANDERS	13
JUNE	CARROLL	12
ELEANOR	HUNT	12
BILLY	POULIN	12
TERRANCE	ROUSH	12
CLARA	SHAW	12
JULIA	FLORES	11
ELSIE	KELLEY	11
BECKY	MILES	11
SUE	PETERS	11
SALLY	PIERCE	11
LESLIE	SEWARD	11

13 rows in set (0.000 sec)

Question 10:

Purpose:
Find number of movie rentals, number of distinct customers, total revenue generated and average replacment cost of each movie category ordered by most revenue to least

name	ROUND(SUM(p.amount), 2)	COUNT(*)	COUNT(DISTINCT cu.customer_id)	ROUND(AVG(f.replacement_cost), 2)
Sports	5314.21	1179	519	20.56
Sci-Fi	4756.98	1101	507	21.19
Animation	4656.30	1166	500	20.28
Drama	4587.39	1060	501	21.63
Comedy	4383.58	941	495	19.02
Action	4375.85	1112	510	21.18
New	4351.62	940	468	19.73
Games	4281.33	969	474	20.74
Foreign	4270.67	1033	493	18.62
Family	4226.07	1096	501	20.01
Documentary	4217.52	1050	483	20.73
Horror	3722.54	846	451	19.62
Children	3655.55	945	482	20.06
Classics	3639.59	939	468	20.96
Travel	3549.64	837	442	19.16
Music	3417.72	830	447	19.19

16 rows in set (0.000 sec)

Why this is interesting:
As we can see, rental count does not necessarily correspond with the total revenue generated from each rental
This is because some genres have better sale prices than others and can generate more money per rental
This query would be of interest to a marketer because it shows the total revenue as well as the number of sales and number of unique customers
This allows a shop to stock up on movies based on whichever of these metrics meets their desires more
They have the option of being more customer-oriented or focusing more on what generates the most revenue
furthermore, they can see the replacement cost to gauge how much it worth it would be to replace a stolen film

Purpose:
Find number of movie rentals, number of distinct customers, total revenue generated and average replacment cost of each movie category ordered by most revenue to least

Why this is interesting:
As we can see, rental count does not necessarily correspond with the total revenue generated from each rental. This is because some genres have better sale prices than others and can generate more money per rental. This query would be of interest to a marketer because it shows the total revenue as well as the number of sales and number of unique customers. This allows a shop to stock up on movies based on whichever of these metrics meets their desires more. They have the option of being more customer-oriented or focusing more on what generates the most revenue. Furthermore, they can see the replacement cost to gauge how worth it would be to replace a stolen film.