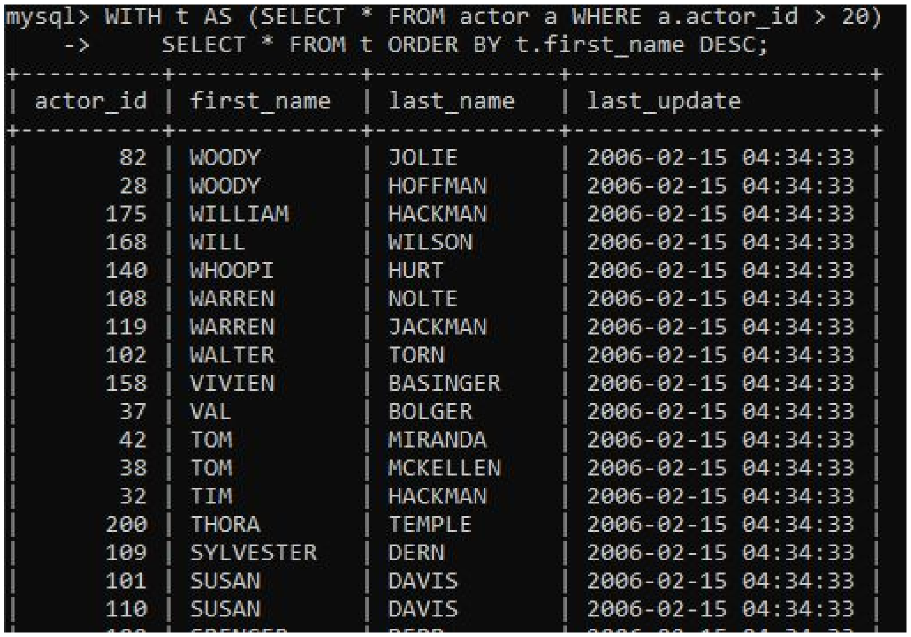
Kevin Hance
CPSC 321 - DBMS
Due 11/14/2019
HW8

READING ASSIGNMENT:

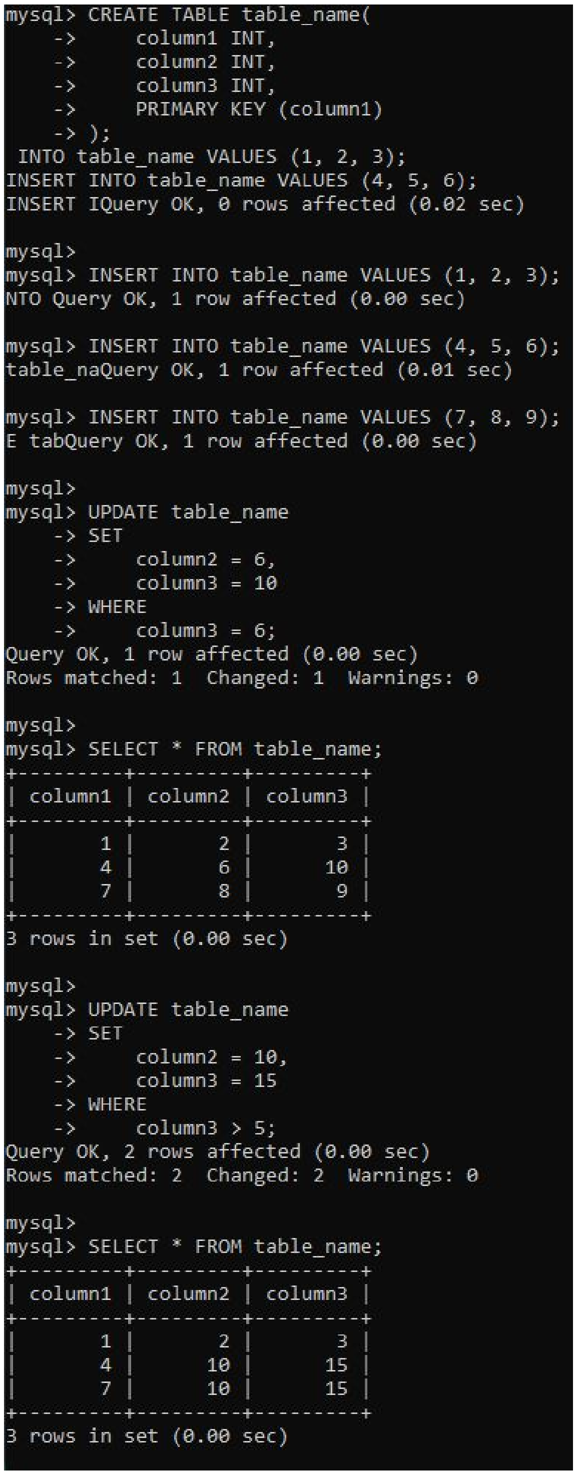
1. A scalar subquery is a subquery that returns only a scalar value representing a single attribute from a single tuple. As an example, this can be used if the subquery uses an aggregate without a GROUP BY clause, resulting in a single scalar value being returned.
2. WITH t AS (SELECT * FROM actor a WHERE a.actor_id > 20)
   SELECT * FROM t ORDER BY t.first_name DESC;



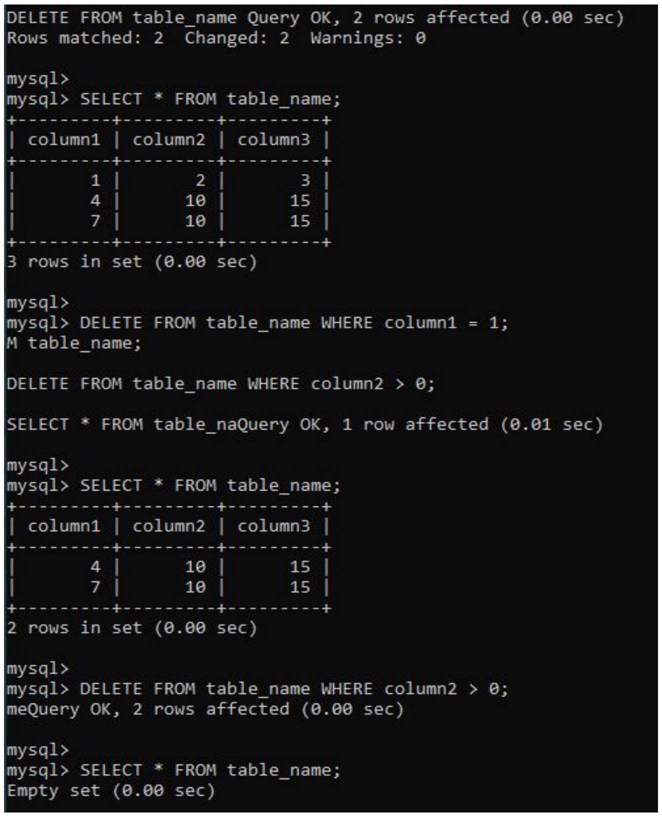
```
mysql> WITH t AS (SELECT * FROM actor a WHERE a.actor_id > 20)
    ->     SELECT * FROM t ORDER BY t.first_name DESC;
+----------+------------+------------+---------------------+
| actor_id | first_name | last_name  | last_update         |
+----------+------------+------------+---------------------+
|       82 | WOODY      | JOLIE      | 2006-02-15 04:34:33 |
|       28 | WOODY      | HOFFMAN    | 2006-02-15 04:34:33 |
|      175 | WILLIAM    | HACKMAN    | 2006-02-15 04:34:33 |
|      168 | WILL       | WILSON     | 2006-02-15 04:34:33 |
|      140 | WHOOPI     | HURT       | 2006-02-15 04:34:33 |
|      108 | WARREN     | NOLTE      | 2006-02-15 04:34:33 |
|      119 | WARREN     | JACKMAN    | 2006-02-15 04:34:33 |
|      102 | WALTER     | TORN       | 2006-02-15 04:34:33 |
|      158 | VIVIEN     | BASINGER   | 2006-02-15 04:34:33 |
|       37 | VAL        | BOLGER     | 2006-02-15 04:34:33 |
|       42 | TOM        | MIRANDA    | 2006-02-15 04:34:33 |
|       38 | TOM        | MCKELLEN   | 2006-02-15 04:34:33 |
|       32 | TIM        | HACKMAN    | 2006-02-15 04:34:33 |
|      200 | THORA      | TEMPLE     | 2006-02-15 04:34:33 |
|      109 | SYLVESTER  | DERN       | 2006-02-15 04:34:33 |
|      101 | SUSAN      | DAVIS      | 2006-02-15 04:34:33 |
|      110 | SUSAN      | DAVIS      | 2006-02-15 04:34:33 |
```

   Query returns a table of all actors with an actor_id greater than 10, ordered in descending alphabetical order by first_name.

3. Table creation and UPDATE statement: (code is in hw8.sql SQL document)

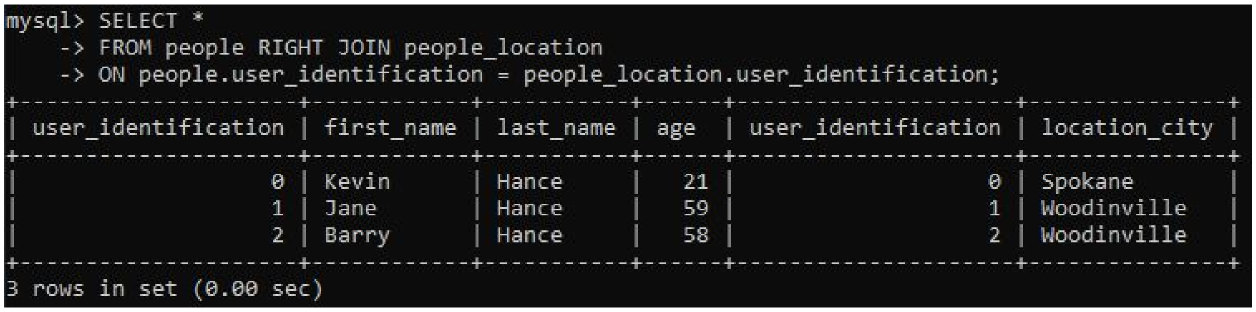


```
mysql> CREATE TABLE table_name(
    ->     column1 INT,
    ->     column2 INT,
    ->     column3 INT,
    ->     PRIMARY KEY (column1)
    -> );
 INTO table_name VALUES (1, 2, 3);
INSERT INTO table_name VALUES (4, 5, 6);
INSERT IQuery OK, 0 rows affected (0.02 sec)

mysql>
mysql> INSERT INTO table_name VALUES (1, 2, 3);
NTO Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO table_name VALUES (4, 5, 6);
table_naQuery OK, 1 row affected (0.01 sec)

mysql> INSERT INTO table_name VALUES (7, 8, 9);
E tabQuery OK, 1 row affected (0.00 sec)

mysql>
mysql> UPDATE table_name
    -> SET
    ->     column2 = 6,
    ->     column3 = 10
    -> WHERE
    ->     column3 = 6;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
mysql> SELECT * FROM table_name;
+---------+---------+---------+
| column1 | column2 | column3 |
+---------+---------+---------+
|       1 |       2 |       3 |
|       4 |       6 |      10 |
|       7 |       8 |       9 |
+---------+---------+---------+
3 rows in set (0.00 sec)

mysql>
mysql> UPDATE table_name
    -> SET
    ->     column2 = 10,
    ->     column3 = 15
    -> WHERE
    ->     column3 > 5;
Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql>
mysql> SELECT * FROM table_name;
+---------+---------+---------+
| column1 | column2 | column3 |
+---------+---------+---------+
|       1 |       2 |       3 |
|       4 |      10 |      15 |
|       7 |      10 |      15 |
+---------+---------+---------+
3 rows in set (0.00 sec)
```



```
DELETE FROM table_name Query OK, 2 rows affected (0.00 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql>
mysql> SELECT * FROM table_name;
+---------+---------+---------+
| column1 | column2 | column3 |
+---------+---------+---------+
|       1 |       2 |       3 |
|       4 |      10 |      15 |
|       7 |      10 |      15 |
+---------+---------+---------+
3 rows in set (0.00 sec)

mysql>
mysql> DELETE FROM table_name WHERE column1 = 1;
M table_name;

DELETE FROM table_name WHERE column2 > 0;

SELECT * FROM table_naQuery OK, 1 row affected (0.01 sec)

mysql>
mysql> SELECT * FROM table_name;
+---------+---------+---------+
| column1 | column2 | column3 |
+---------+---------+---------+
|       4 |      10 |      15 |
|       7 |      10 |      15 |
+---------+---------+---------+
2 rows in set (0.00 sec)

mysql>
mysql> DELETE FROM table_name WHERE column2 > 0;
meQuery OK, 2 rows affected (0.00 sec)

mysql>
mysql> SELECT * FROM table_name;
Empty set (0.00 sec)
```

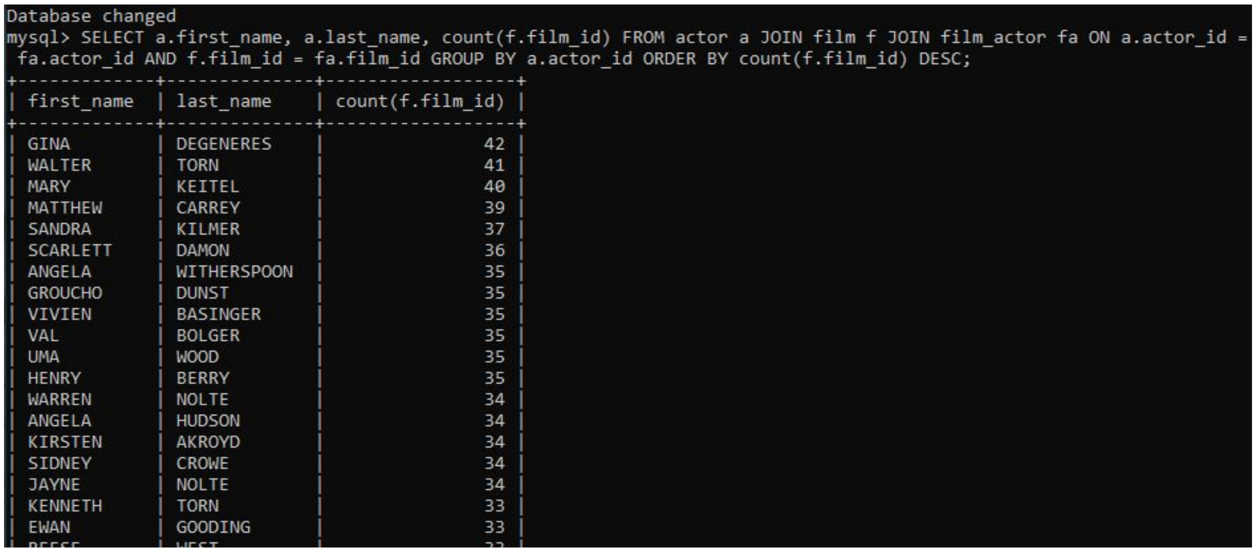
4. SELECT *
   FROM people RIGHT JOIN people_location
   ON people.user_identification = people_location.user_identification;



```
mysql> SELECT *
    -> FROM people RIGHT JOIN people_location
    -> ON people.user_identification = people_location.user_identification;
+---------------------+------------+-----------+------+---------------------+---------------+
| user_identification | first_name | last_name | age  | user_identification | location_city |
+---------------------+------------+-----------+------+---------------------+---------------+
|                   0 | Kevin      | Hance     |   21 |                   0 | Spokane       |
|                   1 | Jane       | Hance     |   59 |                   1 | Woodinville   |
|                   2 | Barry      | Hance     |   58 |                   2 | Woodinville   |
+---------------------+------------+-----------+------+---------------------+---------------+
3 rows in set (0.00 sec)
```

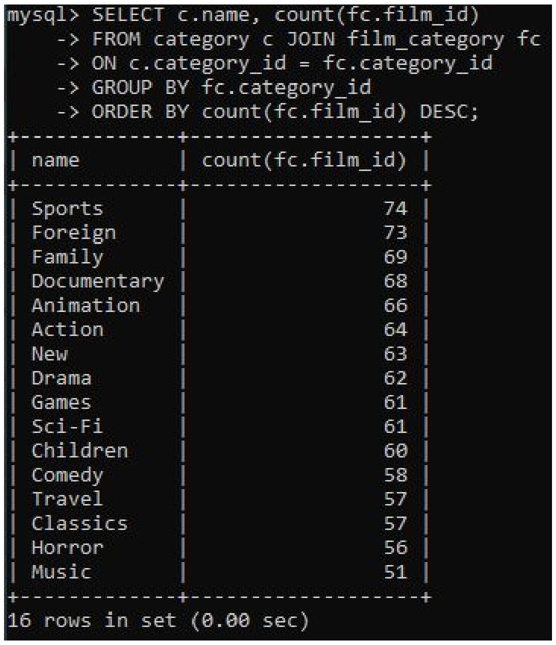
TECHNICAL WORK:

1. SELECT a.first_name, a.last_name, count(f.film_id)
   FROM actor a JOIN film f JOIN film_actor fa
   ON a.actor_id = fa.actor_id AND f.film_id = fa.film_id
   GROUP BY a.actor_id
   ORDER BY count(f.film_id) DESC;



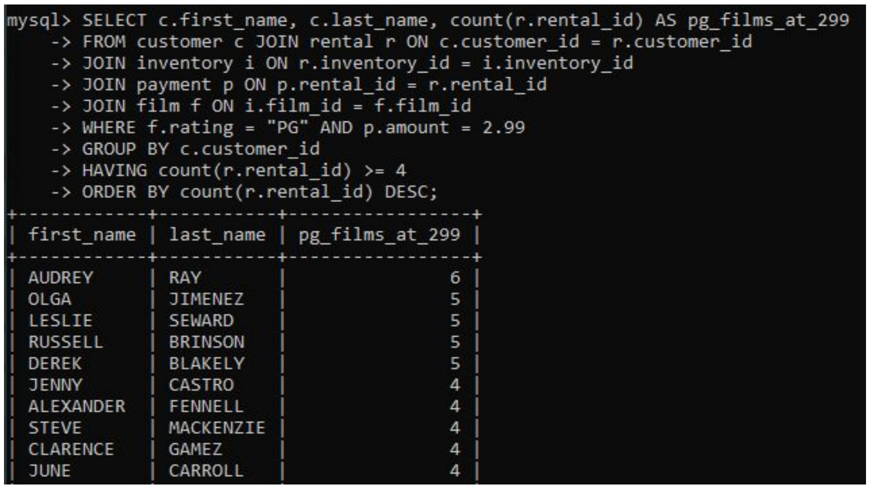
```
Database changed
mysql> SELECT a.first_name, a.last_name, count(f.film_id) FROM actor a JOIN film f JOIN film_actor fa ON a.actor_id =
 fa.actor_id AND f.film_id = fa.film_id GROUP BY a.actor_id ORDER BY count(f.film_id) DESC;
+------------+-------------+------------------+
| first_name | last_name   | count(f.film_id) |
+------------+-------------+------------------+
| GINA       | DEGENERES   |               42 |
| WALTER     | TORN        |               41 |
| MARY       | KEITEL      |               40 |
| MATTHEW    | CARREY      |               39 |
| SANDRA     | KILMER      |               37 |
| SCARLETT   | DAMON       |               36 |
| ANGELA     | WITHERSPOON |               35 |
| GROUCHO    | DUNST       |               35 |
| VIVIEN     | BASINGER    |               35 |
| VAL        | BOLGER      |               35 |
| UMA        | WOOD        |               35 |
| HENRY      | BERRY       |               35 |
| WARREN     | NOLTE       |               34 |
| ANGELA     | HUDSON      |               34 |
| KIRSTEN    | AKROYD      |               34 |
| SIDNEY     | CROWE       |               34 |
| JAYNE      | NOLTE       |               34 |
| KENNETH    | TORN        |               33 |
| EWAN       | GOODING     |               33 |
```

2. SELECT c.name, count(fc.film_id)
   FROM category c JOIN film_category fc
   ON c.category_id = fc.category_id
   GROUP BY fc.category_id
   ORDER BY count(fc.film_id) DESC;



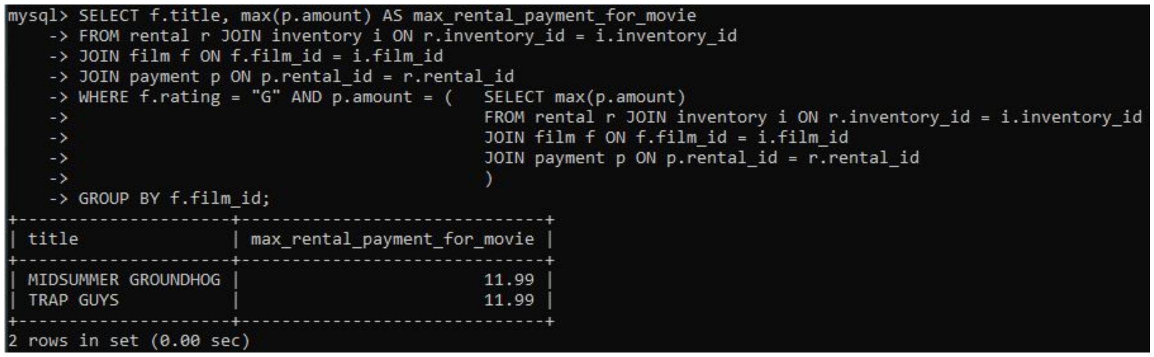
```
mysql> SELECT c.name, count(fc.film_id)
    -> FROM category c JOIN film_category fc
    -> ON c.category_id = fc.category_id
    -> GROUP BY fc.category_id
    -> ORDER BY count(fc.film_id) DESC;
+-------------+------------------+
| name        | count(fc.film_id) |
+-------------+------------------+
| Sports      |               74 |
| Foreign     |               73 |
| Family      |               69 |
| Documentary |               68 |
| Animation   |               66 |
| Action      |               64 |
| New         |               63 |
| Drama       |               62 |
| Games       |               61 |
| Sci-Fi      |               61 |
| Children    |               60 |
| Comedy      |               58 |
| Travel      |               57 |
| Classics    |               57 |
| Horror      |               56 |
| Music       |               51 |
+-------------+------------------+
16 rows in set (0.00 sec)
```

3. SELECT c.first_name, c.last_name, count(r.rental_id) AS pg_films_at_299
   FROM customer c JOIN rental r ON c.customer_id = r.customer_id
   JOIN inventory i ON r.inventory_id = i.inventory_id
   JOIN payment p ON p.rental_id = r.rental_id
   JOIN film f ON i.film_id = f.film_id
   WHERE f.rating = "PG" AND p.amount = 2.99
   GROUP BY c.customer_id
   HAVING count(r.rental_id) >= 4
   ORDER BY count(r.rental_id) DESC;



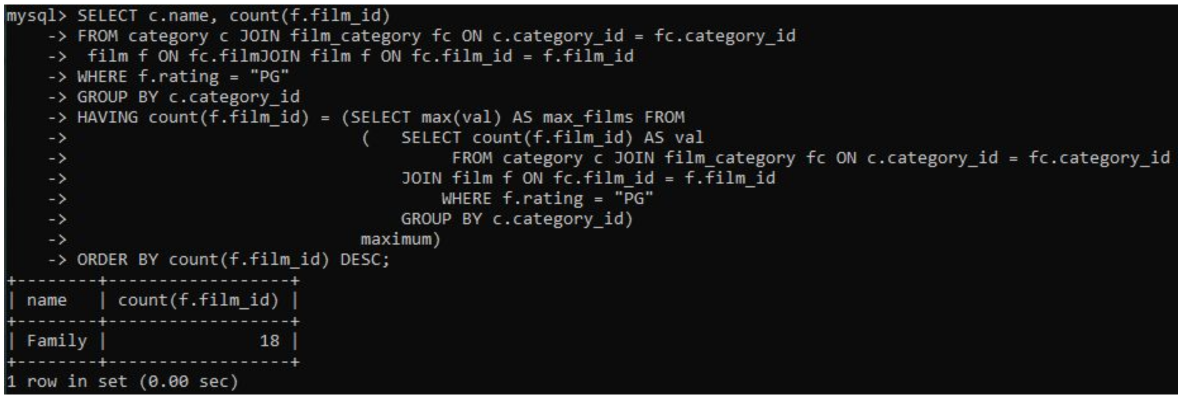
```
mysql> SELECT c.first_name, c.last_name, count(r.rental_id) AS pg_films_at_299
    -> FROM customer c JOIN rental r ON c.customer_id = r.customer_id
    -> JOIN inventory i ON r.inventory_id = i.inventory_id
    -> JOIN payment p ON p.rental_id = r.rental_id
    -> JOIN film f ON i.film_id = f.film_id
    -> WHERE f.rating = "PG" AND p.amount = 2.99
    -> GROUP BY c.customer_id
    -> HAVING count(r.rental_id) >= 4
    -> ORDER BY count(r.rental_id) DESC;
+------------+-----------+-----------------+
| first_name | last_name | pg_films_at_299 |
+------------+-----------+-----------------+
| AUDREY     | RAY       |               6 |
| OLGA       | JIMENEZ   |               5 |
| LESLIE     | SEWARD    |               5 |
| RUSSELL    | BRINSON   |               5 |
| DEREK      | BLAKELY   |               5 |
| JENNY      | CASTRO    |               4 |
| ALEXANDER  | FENNELL   |               4 |
| STEVE      | MACKENZIE |               4 |
| CLARENCE   | GAMEZ     |               4 |
| JUNE       | CARROLL   |               4 |
```

4. SELECT f.title, max(p.amount) AS max_rental_payment_for_movie
   FROM rental r JOIN inventory i ON r.inventory_id = i.inventory_id
   JOIN film f ON f.film_id = i.film_id
   JOIN payment p ON p.rental_id = r.rental_id
   WHERE f.rating = "G" AND p.amount = (SELECT max(p.amount)
   FROM rental r JOIN inventory i ON r.inventory_id = i.inventory_id
   JOIN film f ON f.film_id = i.film_id
   JOIN payment p ON p.rental_id = r.rental_id
)
   GROUP BY f.film_id;



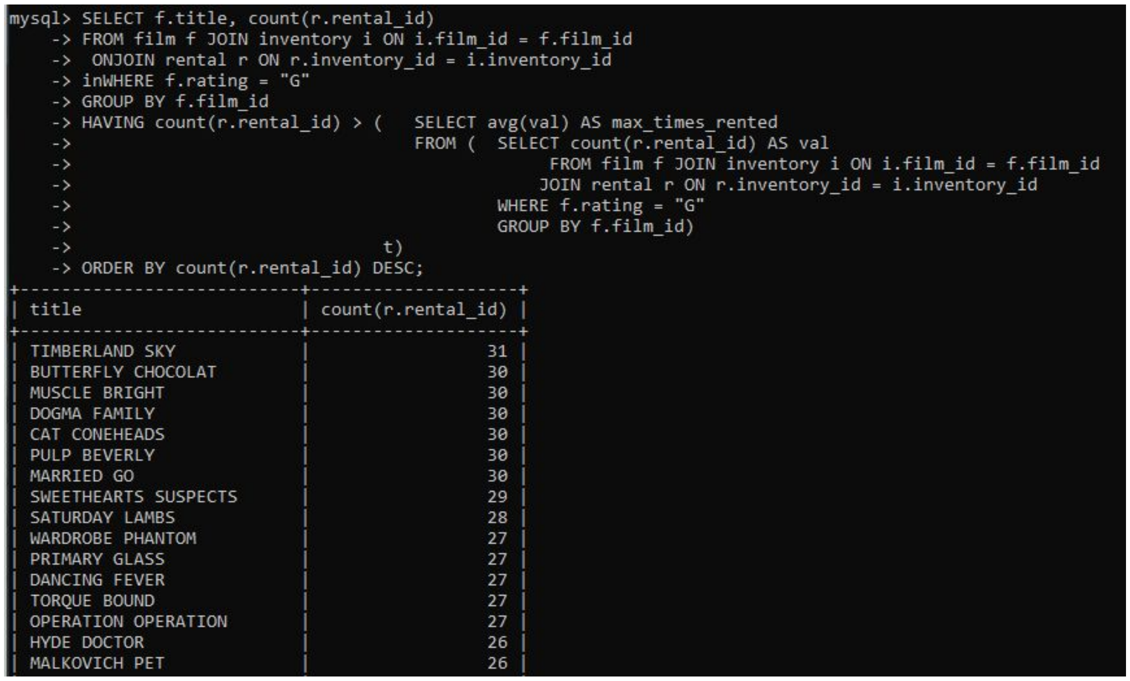
```
mysql> SELECT f.title, max(p.amount) AS max_rental_payment_for_movie
    -> FROM rental r JOIN inventory i ON r.inventory_id = i.inventory_id
    -> JOIN film f ON f.film_id = i.film_id
    -> JOIN payment p ON p.rental_id = r.rental_id
    -> WHERE f.rating = "G" AND p.amount = (   SELECT max(p.amount)
    ->                                         FROM rental r JOIN inventory i ON r.inventory_id = i.inventory_id
    ->                                         JOIN film f ON f.film_id = i.film_id
    ->                                         JOIN payment p ON p.rental_id = r.rental_id
    ->                                         )
    -> GROUP BY f.film_id;
+---------------------+-----------------------------+
| title               | max_rental_payment_for_movie |
+---------------------+-----------------------------+
| MIDSUMMER GROUNDHOG |                       11.99 |
| TRAP GUYS           |                       11.99 |
+---------------------+-----------------------------+
2 rows in set (0.00 sec)
```

5. SELECT c.name, count(f.film_id)
   FROM category c JOIN film_category fc ON c.category_id = fc.category_id
   JOIN film f ON fc.film_id = f.film_id
   WHERE f.rating = "PG"
   GROUP BY c.category_id
   HAVING count(f.film_id) = (SELECT max(val) AS max_films FROM
   (SELECT count(f.film_id) AS val
   FROM category c JOIN film_category fc ON c.category_id = fc.category_id
   JOIN film f ON fc.film_id = f.film_id
   WHERE f.rating = "PG"
   GROUP BY c.category_id)
   maximum)
   ORDER BY count(f.film_id) DESC;



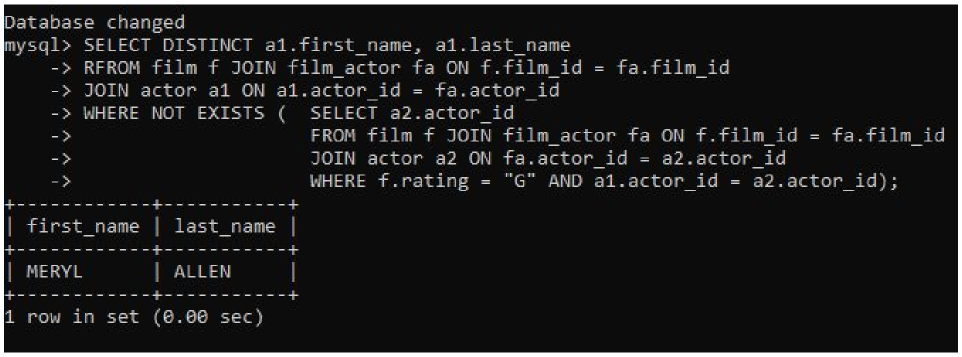
```
mysql> SELECT c.name, count(f.film_id)
    -> FROM category c JOIN film_category fc ON c.category_id = fc.category_id
    ->  film f ON fc.filmJOIN film f ON fc.film_id = f.film_id
    -> WHERE f.rating = "PG"
    -> GROUP BY c.category_id
    -> HAVING count(f.film_id) = (SELECT max(val) AS max_films FROM
    ->                             (   SELECT count(f.film_id) AS val
    ->                                     FROM category c JOIN film_category fc ON c.category_id = fc.category_id
    ->                                 JOIN film f ON fc.film_id = f.film_id
    ->                                     WHERE f.rating = "PG"
    ->                                 GROUP BY c.category_id)
    ->                             maximum)
    -> ORDER BY count(f.film_id) DESC;
+--------+------------------+
| name   | count(f.film_id) |
+--------+------------------+
| Family |               18 |
+--------+------------------+
1 row in set (0.00 sec)
```

6.
```
SELECT f.title, count(r.rental_id)
FROM film f JOIN inventory i ON i.film_id = f.film_id
JOIN rental r ON r.inventory_id = i.inventory_id
WHERE f.rating = "G"
GROUP BY f.film_id
HAVING count(r.rental_id) > (   SELECT avg(val) AS max_times_rented
                                FROM (  SELECT count(r.rental_id) AS val
                                        FROM film f JOIN inventory i ON i.film_id = f.film_id
                                        JOIN rental r ON r.inventory_id = i.inventory_id
                                        WHERE f.rating = "G"
                                        GROUP BY f.film_id)
                             t)
ORDER BY count(r.rental_id) DESC;
```



```
mysql> SELECT f.title, count(r.rental_id)
    -> FROM film f JOIN inventory i ON i.film_id = f.film_id
    ->  ONJOIN rental r ON r.inventory_id = i.inventory_id
    -> inWHERE f.rating = "G"
    -> GROUP BY f.film_id
    -> HAVING count(r.rental_id) > (   SELECT avg(val) AS max_times_rented
    ->                                 FROM (  SELECT count(r.rental_id) AS val
    ->                                              FROM film f JOIN inventory i ON i.film_id = f.film_id
    ->                                             JOIN rental r ON r.inventory_id = i.inventory_id
    ->                                         WHERE f.rating = "G"
    ->                                         GROUP BY f.film_id)
    ->                              t)
    -> ORDER BY count(r.rental_id) DESC;
+----------------------------+--------------------+
| title                      | count(r.rental_id) |
+----------------------------+--------------------+
| TIMBERLAND SKY             |                 31 |
| BUTTERFLY CHOCOLAT         |                 30 |
| MUSCLE BRIGHT              |                 30 |
| DOGMA FAMILY               |                 30 |
| CAT CONEHEADS              |                 30 |
| PULP BEVERLY               |                 30 |
| MARRIED GO                 |                 30 |
| SWEETHEARTS SUSPECTS       |                 29 |
| SATURDAY LAMBS             |                 28 |
| WARDROBE PHANTOM           |                 27 |
| PRIMARY GLASS              |                 27 |
| DANCING FEVER              |                 27 |
| TORQUE BOUND               |                 27 |
| OPERATION OPERATION        |                 27 |
| HYDE DOCTOR                |                 26 |
| MALKOVICH PET              |                 26 |
```

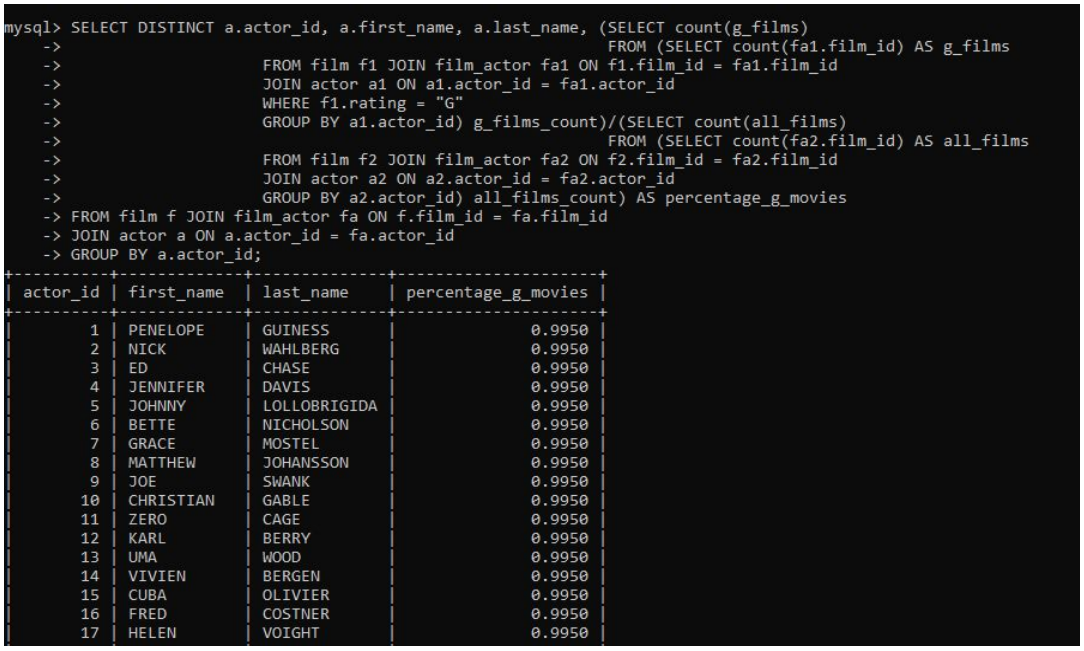
7. SELECT DISTINCT a1.first_name, a1.last_name
   FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
   JOIN actor a1 ON a1.actor_id = fa.actor_id
   WHERE NOT EXISTS (SELECT a2.actor_id
                       FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
                       JOIN actor a2 ON fa.actor_id = a2.actor_id
                       WHERE f.rating = "G" AND a1.actor_id = a2.actor_id);



```
Database changed
mysql> SELECT DISTINCT a1.first_name, a1.last_name
    -> RFROM film f JOIN film_actor fa ON f.film_id = fa.film_id
    -> JOIN actor a1 ON a1.actor_id = fa.actor_id
    -> WHERE NOT EXISTS (  SELECT a2.actor_id
    ->                     FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
    ->                     JOIN actor a2 ON fa.actor_id = a2.actor_id
    ->                     WHERE f.rating = "G" AND a1.actor_id = a2.actor_id);
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| MERYL      | ALLEN     |
+------------+-----------+
1 row in set (0.00 sec)
```

8. I was unsure how to approach this problem.

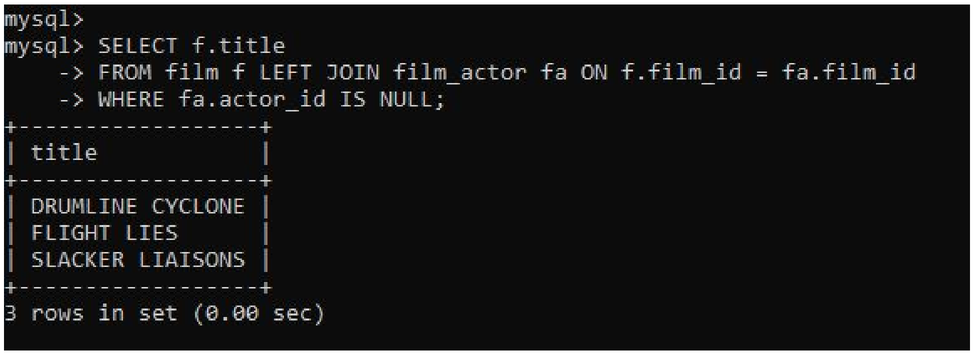
9. SELECT DISTINCT a.actor_id, a.first_name, a.last_name, (SELECT count(g_films)
```
                FROM (SELECT count(fa1.film_id) AS g_films
                FROM film f1 JOIN film_actor fa1 ON f1.film_id = fa1.film_id
                JOIN actor a1 ON a1.actor_id = fa1.actor_id
                WHERE f1.rating = "G"
                GROUP BY a1.actor_id) g_films_count)/(SELECT count(all_films)
                        FROM (SELECT count(fa2.film_id) AS all_films
                FROM film f2 JOIN film_actor fa2 ON f2.film_id = fa2.film_id
                JOIN actor a2 ON a2.actor_id = fa2.actor_id
                GROUP BY a2.actor_id) all_films_count) AS percentage_g_movies
FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
JOIN actor a ON a.actor_id = fa.actor_id
GROUP BY a.actor_id;
```



```
mysql> SELECT DISTINCT a.actor_id, a.first_name, a.last_name, (SELECT count(g_films)
    ->                                                                FROM (SELECT count(fa1.film_id) AS g_films
    ->                      FROM film f1 JOIN film_actor fa1 ON f1.film_id = fa1.film_id
    ->                      JOIN actor a1 ON a1.actor_id = fa1.actor_id
    ->                      WHERE f1.rating = "G"
    ->                      GROUP BY a1.actor_id) g_films_count)/(SELECT count(all_films)
    ->                                                                FROM (SELECT count(fa2.film_id) AS all_films
    ->                      FROM film f2 JOIN film_actor fa2 ON f2.film_id = fa2.film_id
    ->                      JOIN actor a2 ON a2.actor_id = fa2.actor_id
    ->                      GROUP BY a2.actor_id) all_films_count) AS percentage_g_movies
    -> FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
    -> JOIN actor a ON a.actor_id = fa.actor_id
    -> GROUP BY a.actor_id;
+----------+------------+--------------+---------------------+
| actor_id | first_name | last_name    | percentage_g_movies |
+----------+------------+--------------+---------------------+
|        1 | PENELOPE   | GUINESS      |              0.9950 |
|        2 | NICK       | WAHLBERG     |              0.9950 |
|        3 | ED         | CHASE        |              0.9950 |
|        4 | JENNIFER   | DAVIS        |              0.9950 |
|        5 | JOHNNY     | LOLLOBRIGIDA |              0.9950 |
|        6 | BETTE      | NICHOLSON    |              0.9950 |
|        7 | GRACE      | MOSTEL       |              0.9950 |
|        8 | MATTHEW    | JOHANSSON    |              0.9950 |
|        9 | JOE        | SWANK        |              0.9950 |
|       10 | CHRISTIAN  | GABLE        |              0.9950 |
|       11 | ZERO       | CAGE         |              0.9950 |
|       12 | KARL       | BERRY        |              0.9950 |
|       13 | UMA        | WOOD         |              0.9950 |
|       14 | VIVIEN     | BERGEN       |              0.9950 |
|       15 | CUBA       | OLIVIER      |              0.9950 |
|       16 | FRED       | COSTNER      |              0.9950 |
|       17 | HELEN      | VOIGHT       |              0.9950 |
```

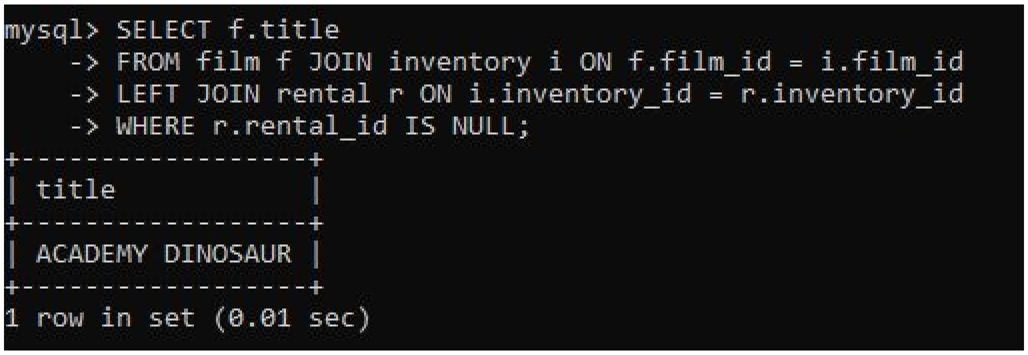
I don’t think I got query 9 right, as the same percentage was returned for every value.

10. SELECT f.title
```
FROM film f LEFT JOIN film_actor fa ON f.film_id = fa.film_id
WHERE fa.actor_id IS NULL;
```



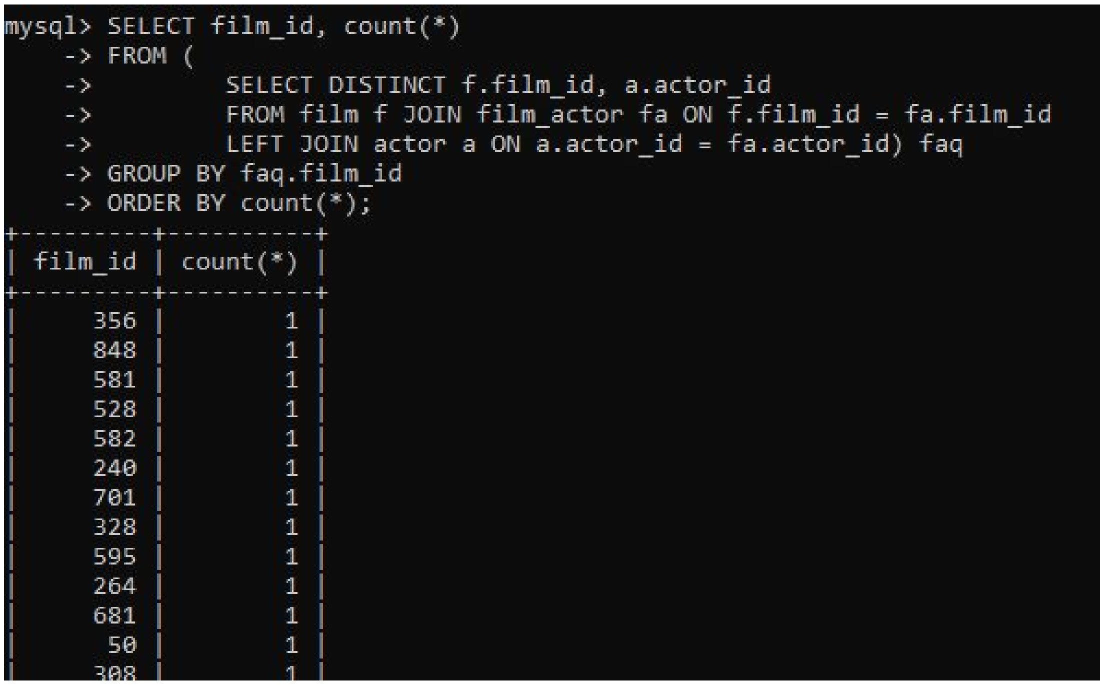
```
mysql>
mysql> SELECT f.title
    -> FROM film f LEFT JOIN film_actor fa ON f.film_id = fa.film_id
    -> WHERE fa.actor_id IS NULL;
+-----------------+
| title           |
+-----------------+
| DRUMLINE CYCLONE |
| FLIGHT LIES     |
| SLACKER LIAISONS |
+-----------------+
3 rows in set (0.00 sec)
```

11. SELECT f.title
    FROM film f JOIN inventory i ON f.film_id = i.film_id
    LEFT JOIN rental r ON i.inventory_id = r.inventory_id
    WHERE r.rental_id IS NULL;



```
mysql> SELECT f.title
    -> FROM film f JOIN inventory i ON f.film_id = i.film_id
    -> LEFT JOIN rental r ON i.inventory_id = r.inventory_id
    -> WHERE r.rental_id IS NULL;
+------------------+
| title            |
+------------------+
| ACADEMY DINOSAUR |
+------------------+
1 row in set (0.01 sec)
```

12. SELECT film_id, count(*)
    FROM (
        SELECT DISTINCT f.film_id, a.actor_id
        FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
        LEFT JOIN actor a ON a.actor_id = fa.actor_id) faq
    GROUP BY faq.film_id
    ORDER BY count(*);



```
mysql> SELECT film_id, count(*)
    -> FROM (
    ->         SELECT DISTINCT f.film_id, a.actor_id
    ->         FROM film f JOIN film_actor fa ON f.film_id = fa.film_id
    ->         LEFT JOIN actor a ON a.actor_id = fa.actor_id) faq
    -> GROUP BY faq.film_id
    -> ORDER BY count(*);
+---------+----------+
| film_id | count(*) |
+---------+----------+
|     356 |        1 |
|     848 |        1 |
|     581 |        1 |
|     528 |        1 |
|     582 |        1 |
|     240 |        1 |
|     701 |        1 |
|     328 |        1 |
|     595 |        1 |
|     264 |        1 |
|     681 |        1 |
|      50 |        1 |
|     308 |        1 |
```

I don’t think I did this one entirely correctly either because I got no films that had zero actors in them. However, I believe the rest of the data returned from my query is accurate to the number of actors featured in each film.