

Sakila Database Exercise

1. Which actors have the first name 'Scarlett'?

```
mysql> select *from actor where first_name="Scarlett";
+-----+-----+-----+-----+
| actor_id | first_name | last_name | last_update |
+-----+-----+-----+-----+
|      81 | SCARLETT  | DAMON    | 2006-02-15 04:34:33 |
|     124 | SCARLETT  | BENING    | 2006-02-15 04:34:33 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

2. Which actors have the last name 'Johansson'?

```
mysql> select *from actor where last_name="Johansson";
+-----+-----+-----+-----+
| actor_id | first_name | last_name | last_update |
+-----+-----+-----+-----+
|      8   | MATTHEW    | JOHANSSON | 2006-02-15 04:34:33 |
|     64   | RAY        | JOHANSSON | 2006-02-15 04:34:33 |
|    146   | ALBERT     | JOHANSSON | 2006-02-15 04:34:33 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

3. How many distinct actors last names are there?

```
mysql> select count(distinct last_name) as Number_of_distinct_la
st_name from actor;
+-----+
| Number_of_distinct_last_name |
+-----+
|                121          |
+-----+
1 row in set (0.00 sec)
```

4. Which last names are not repeated?

```
mysql> select last_name from actor group by last_name having count(last_name)=1;
```

last_name
ASTAIRE
BACALL
BALE
BALL
BARRYMORE
BASINGER
BERGEN
BERGMAN
BIRCH
BLOOM
BRIDGES
BULLOCK
CARREY
CHAPLIN
CLOSE
COSTNER
CROWE
CRUISE
CRUZ
DAMON
DAY-LEWIS
DERN
DREYFUSS
DUNST
GABLE
GIBSON
GOLDBERG
GRANT
HAWKE
HESTON
HOPE
HUDSON
HUNT
HURT
JOLIE
JOVOVICH
LEIGH
LOLLOBRIGIDA
MALDEN
MANSFIELD
MARX
MCDORMAND
MIRANDA
NICHOLSON
PESCI
PFEIFFER
PHOENIX
PINKETT
PITT
POSEY
PRESLEY
REYNOLDS
RYDER
SINATRA
SOBIESKI
STALLONE
SUVARI
SWANK
TAUTOU
TOMEI
VOIGHT
WALKEN
WAYNE
WILSON
WITHERSPOON
WRAY

```
66 rows in set (0.00 sec)
```

5. Which last names appear more than once?

```
mysql> select last_name from actor group by last_name having count(last_name)>1;
```

last_name
AKROYD
ALLEN
BAILEY
BENING
BERRY
BOLGER
BRODY
CAGE
CHASE
CRAWFORD
CRONYN
DAVIS
DEAN
DEE
DEGENERES
DENCH
DEPP
DUKAKIS
FAWCETT
GARLAND
GOODING
GUINNESS
HACKMAN
HARRIS
HOFFMAN
HOPKINS
HOPPER
JACKMAN
JOHANSSON
KEITEL
KILMER
MCCONAUGHEY
MCKELLEN
MCQUEEN
MONROE
MOSTEL
NEESON
NOLTE
OLIVIER
PALTROW
PECK
PENN
SILVERSTONE
STREEP
TANDY
TEMPLE
TORN
TRACY
WAHLBERG
WEST
WILLIAMS
WILLIS
WINSLET
WOOD
ZELLWEGER

55 rows in set (0.00 sec)

7. Which actor has appeared in the most films?

```
mysql> SELECT actor.actor_id, actor.first_name, actor.last_name, film_actor.film_count
-> FROM actor
-> JOIN (
->     SELECT actor_id, count(actor_id) AS film_count
->     FROM film_actor
->     GROUP BY actor_id
-> ) film_actor ON actor.actor_id = film_actor.actor_id
-> ORDER BY film_actor.film_count DESC
-> LIMIT 1;
+-----+-----+-----+-----+
| actor_id | first_name | last_name | film_count |
+-----+-----+-----+-----+
|      107 | GINA      | DEGENERES |          42 |
+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

8. Is 'Academy Dinosaur' available for rent from Store 1?

```
mysql> SELECT film.film_id, film.title, store.store_id, inventory.inventory_id
-> FROM film
-> INNER JOIN inventory ON film.film_id = inventory.film_id
-> INNER JOIN store ON inventory.store_id = store.store_id
-> WHERE (film.title = 'Academy Dinosaur' AND store.store_id = 1);
+-----+-----+-----+-----+
| film_id | title          | store_id | inventory_id |
+-----+-----+-----+-----+
|      1  | ACADEMY DINOSAUR |      1  |          1  |
|      1  | ACADEMY DINOSAUR |      1  |          2  |
|      1  | ACADEMY DINOSAUR |      1  |          3  |
|      1  | ACADEMY DINOSAUR |      1  |          4  |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

10. When is 'Academy Dinosaur' due?

```
mysql> SELECT rental_date, rental_date + interval(SELECT rental_duration FROM film WHERE film_id = 1) DAY AS due_date
-> FROM rental
-> WHERE rental_id = (SELECT rental_id FROM rental ORDER BY rental_id DESC LIMIT 1);
+-----+-----+
| rental_date | due_date |
+-----+-----+
| 2005-08-23 22:50:12 | 2005-08-29 22:50:12 |
+-----+-----+
1 row in set (0.00 sec)
```

11. What is that average running time of all the films in the sakila DB?

```
mysql> SELECT avg(length) AS average_running_time_of_films FROM film;
+-----+
| average_running_time_of_films |
+-----+
| 115.2720 |
+-----+
1 row in set (0.00 sec)
```

12. What is the average running time of films by category?

```
mysql> SELECT category.name, avg(length) AS average_running_time
-> FROM film
-> JOIN film_category USING (film_id)
-> JOIN category USING (category_id)
-> GROUP BY category.name
-> HAVING avg(length) > (SELECT avg(length) FROM film)
-> ORDER BY avg(length) DESC;
+-----+-----+
| name | average_running_time |
+-----+-----+
| Sports | 128.2027 |
| Games | 127.8361 |
| Foreign | 121.6986 |
| Drama | 120.8387 |
| Comedy | 115.8276 |
+-----+-----+
5 rows in set (0.01 sec)
```

13. Why does this query return the empty set?

```
select * from film natural join inventory;
```

Natural join will only work if the same named columns values are same.

In this case, the table film and inventory have different columns and values.