Prática 6

1. Selecionar o nome das categorias que começam pela letra 'S'. (2)

```
SELECT name
FROM category
WHERE name LIKE 's%';
```

2. Selecionar o primeiro e último nome de todos os atores, ordenados alfabeticamente pelo último nome.(200)

```
SELECT first_name, last_name
FROM actor
ORDER BY last_name;
```

3. Selecionar o nome de todos os filmes alugados pelo cliente número 258.(24)

```
SELECT DISTINCT title
FROM ((film NATURAL JOIN inventory) NATURAL JOIN rental)
WHERE customer_id = 258;
```

4. Selecionar o nome dos filmes em que participa o ator 'HARRISON BALE'. (28)

```
SELECT title
FROM (film NATURAL JOIN (actor NATURAL JOIN film_actor))
WHERE first_name LIKE 'Harrison' AND last_name LIKE 'Bale';
```

5. Selecionar o nome dos clientes que têm atualmente DVDs alugados. (159)

```
SELECT first_name, last_name
FROM customer
WHERE customer_id IN (SELECT customer_id
                      FROM rental
                      WHERE return_date IS NULL);
```

6. Selecionar o primeiro e último nome de cada empregado e o primeiro e último nome do respetivo gerente. (2)

```
SELECT E.first_name AS Employee_FName, E.last_name AS Employee_LName, M.first_name
AS Manager_FName, M.last_name AS Manager_LName
FROM staff AS E, staff as M
WHERE E.store_id IN (SELECT store_id
                     FROM store
                     WHERE manager = M.staff_id);
```

7. Selecionar o primeiro e último nome dos atores que participam em filmes da categoria 'Drama' (note que existem duas actrizes com o nome 'SUSAN DAVIS'). (162)

```
SELECT first_name, last_name
FROM actor
WHERE actor_id IN (SELECT actor_id
                   FROM film_actor
                   WHERE film_id IN (SELECT film_id
                                     FROM film
                                     WHERE category_id IN (SELECT category_id
                                                           FROM category
                                                           WHERE name  LIKE 'Drama')));
```

8. Inserir um novo DVD para o filme 'EYES DRIVING' na loja número 1.

```
INSERT INTO inventory(film_id, store_id)
SELECT film_id, store_id
FROM (film NATURAL JOIN inventory)
WHERE title LIKE 'EYES DRIVING' AND store_id = 1
GROUP BY title;
```

9. Eliminar todos os pagamentos efetuados pela cliente 'LISA ANDERSON'.

```
DELETE FROM payment
WHERE customer_id IN (SELECT customer_id
        FROM customer
        WHERE first_name LIKE 'lisa' AND last_name LIKE 'anderson');
```

10. Atualizar o telefone da cliente 'KAREN JACKSON' para '351212212212'.

```
UPDATE customer SET phone = '351212212212'
WHERE first_name LIKE 'Karen' AND last_name LIKE 'Jackson';
```

Pratica 7

1- Selecionar o primeiro e último nome de todos os clientes e de todos os empregados. (601)

```
(SELECT first_name, last_name FROM customer)
UNION
(SELECT first_name, last_name FROM staff);
```

2- Selecionar os nomes dos filmes com maior duração. (10)

```
SELECT title
FROM film
WHERE length=(SELECT MAX(length) FROM film);
```

3- Selecionar o nome dos filmes que nunca foram alugados pelo cliente 'LEE HAWKS' mas que já foram alugados por outros clientes. (932)

```
SELECT title
FROM film
WHERE film_id IN (SELECT DISTINCT film_id FROM inventory NATURAL JOIN rental)
    AND
    film_id NOT IN (SELECT DISTINCT film_id
            FROM (inventory NATURAL JOIN rental)
            WHERE customer_id IN (SELECT customer_id
                    FROM customer
                    WHERE first_name LIKE "LEE" AND last_name LIKE "HAWKS"));
```

4- Selecionar o nome dos filmes em que não participa nenhum ator.(3)

```
SELECT title FROM film
WHERE film_id NOT IN (SELECT film_id FROM film_actor);
```

5- Selecionar o nome dos filmes, com existência de DVDs, que nunca foram alugados.(0)

```
SELECT title FROM (film NATURAL JOIN inventory)
WHERE film_id NOT IN (SELECT film_id FROM (inventory NATURAL JOIN rental));
```

6- Selecionar o nome dos clientes que alugaram filmes de todas as categorias.

```
SELECT first_name, last_name
FROM customer AS C
WHERE NOT EXISTS (SELECT *
        FROM category AS K
        WHERE NOT EXISTS (SELECT *
                FROM (film NATURAL JOIN inventory) NATURAL JOIN rental
                WHERE C.customer_id = customer_id AND K.category_id = category_id));
```

7- Selecionar o nome dos atores que já participaram em filmes de todas as categorias.

```
SELECT A.first_name, A.last_name
FROM actor AS A
WHERE NOT EXISTS (SELECT *
        FROM category as K
        WHERE NOT EXISTS (SELECT *
                FROM (film_actor NATURAL JOIN film)
                WHERE K.category_id = category_id AND A.actor_id = actor_id));
```

Pratica 8

1 - Selecionar, por cidade, o número de empregados que trabalham em lojas dessa cidade. (2)

```
SELECT L.city, COUNT(*)
FROM (staff AS S JOIN store AS L ON L.store_id = S.store_id)
GROUP BY L.city;
```

2 - Selecionar para cada filme o número de DVDs que dele existem. (1000)

```
SELECT title, COUNT(*)
FROM (film NATURAL JOIN inventory)
GROUP BY film_id
UNION
SELECT title, 0
FROM (film AS F LEFT OUTER JOIN inventory AS I ON F.film_id=I.film_id)
WHERE I.inventory_id IS NULL;
```

3 - Selecionar o nome dos filmes em que participa o ator 'HARRISON BALE' que já tiveram mais do que 20 alugueres. (8)

```
SELECT title
FROM film AS F
WHERE
 ((SELECT COUNT(*)
   FROM (inventory NATURAL JOIN rental)
   WHERE F.film_id=film_id) > 20)
 AND
 (F.film_id IN (SELECT film_id
        FROM (actor NATURAL JOIN film_actor)
        WHERE first_name LIKE 'HARRISON' AND last_name LIKE 'BALE'));
```

4 - Selecionar o nome dos filmes dos quais há mais DVDs. (72)

```
SELECT title, COUNT(*)
FROM (film NATURAL JOIN inventory)
GROUP BY film_id
HAVING COUNT(*) = (SELECT MAX(C)
            FROM (SELECT COUNT(*) AS C
                FROM (film NATURAL JOIN inventory)
                GROUP BY film_id) AS AUX);
```

5 - Selecionar o nome dos clientes que já fizeram devoluções tardias, de forma ordenada decrescente no número total de devoluções tardias. (599)

```
SELECT first_name, last_name, COUNT
FROM (SELECT customer_id, COUNT(*) AS COUNT
    FROM film NATURAL JOIN (inventory NATURAL JOIN rental)
    WHERE (return_date IS NOT NULL) AND (DATEDIFF(return_date,rent_date) >
rental_duration)
    GROUP BY customer_id
    ORDER BY COUNT DESC) AS AUX NATURAL JOIN customer;
```

6 - Selecionar, por nome de categoria, o número médio de atores por filme dessa categoria. (16)

```
SELECT name, AVG(numActors) AS MediaActores
FROM (SELECT film_id, title, name, category_id, COUNT(*) AS numActors
    FROM ((film NATURAL JOIN category) NATURAL JOIN film_actor)
    GROUP BY film_id) AS AUX
GROUP BY category_id;
```

7 - Selecionar os pares de atores que mais vezes contracenaram juntos.

```
SELECT act_A.first_name AS Afn, act_A.last_name AS Aln, act_B.first_name AS Bfn,
act_B.last_name AS Aln
 FROM (  SELECT P.actor_id AS A, S.actor_id AS B
    FROM film_actor AS P, film_actor AS S
    WHERE   P.film_id = S.film_id AND
       P.actor_id < S.actor_id
    GROUP BY P.actor_id, S.actor_id
    HAVING COUNT(*) = (
     SELECT MAX(n_cont) AS n_cont_max
     FROM (  SELECT P.actor_id AS A, S.actor_id AS B, COUNT(*) n_cont
        FROM film_actor AS P, film_actor AS S
        WHERE   P.film_id = S.film_id AND
           P.actor_id < S.actor_id
        GROUP BY P.actor_id, S.actor_id) AS CONS_A)) AS CONS_B,
    actor AS act_A, actor AS act_B
 WHERE act_A.actor_id = A AND act_B.actor_id = B;
```

Pratica 9

1. **Selecionar o primeiro e último nome de cada empregado e o primeiro e último nome do respetivo gerente, ordenado pelo atributo 'staff_id' do empregado.**

```
SELECT E.first_name, E.last_name, B.first_name, B.last_name
FROM staff as E, staff as B
WHERE B.staff_id IN (SELECT manager
                     FROM store
                     WHERE store_id = E.store_id);
```

2. **Selecionar o primeiro nome dos atores que participam no filme 'WYOMING STORM', ordenando alfabeticamente pelo primeiro nome.**

```
SELECT first_name
FROM (actor NATURAL JOIN film_actor)
WHERE film_id in (SELECT film_id
                  FROM film
                  WHERE title LIKE "WYOMING STORM")
ORDER BY first_name;
```

3. **Selecionar o primeiro e último nome dos atores que participam em filmes da categoria 'Drama', ordenado decrescentemente pelo atributo 'actor_id'.**

```
SELECT DISTINCT first_name, last_name
FROM actor
WHERE actor_id IN (SELECT DISTINCT(actor_id)
                   FROM (film_actor NATURAL JOIN (category NATURAL JOIN film))
                   WHERE name LIKE 'DRAMA')
ORDER BY actor_id DESC;
```

4. **Selecionar o primeiro e último nome dos clientes que já alugaram filmes em lojas que ficam em cidades diferentes da cidade onde moram, ordenado pelo atributo 'customer_id' do cliente.**

```
SELECT C.first_name, C.last_name
FROM customer AS C
WHERE C.customer_id IN (SELECT customer_id
                        FROM rental
                        WHERE inventory_id IN (SELECT inventory_id
                                               FROM (store NATURAL JOIN inventory)
                                               WHERE C.city NOT LIKE city))
ORDER BY C.customer_id;
```

5. **Selecionar o primeiro e último nome dos atores que já contracenaram com a actriz 'JULIA ZELLWEGER', ordenado pelo atributo 'actor_id'.**

```
SELECT DISTINCT first_name, last_name
FROM actor
WHERE actor_id IN (SELECT actor_id
                   FROM film_actor
                   WHERE film_id IN (SELECT film_id
                                     FROM (film_actor NATURAL JOIN actor)
                                     WHERE first_name LIKE 'JULIA' AND last_name LIKE 'ZELLWEGER'))
    AND
```

```
    actor_id NOT IN (SELECT actor_id
                FROM actor
                WHERE first_name LIKE 'JULIA' AND last_name LIKE 'ZELLWEGER')
ORDER BY actor_id;
```

6. **Selecionar, por categoria, o nome da categoria e a média da duração dos filmes dessa categoria, ordenado alfabeticamente pelo nome da categoria (considere apenas categorias para as quais existe pelo menos um filme na base de dados).**

```
SELECT C.name, AVG(F.length)
FROM category AS C, film AS F
WHERE C.category_id = F.category_id
GROUP BY C.category_id
ORDER BY C.name
```

7. **Selecionar o nome dos filmes que já tiveram devoluções tardias juntamente com o total dessas devoluções, ordenado decrescentemente no número total de devoluções tardias (use DATEDIFF(data1,data2) para obter a diferença em dias entre duas datas).**

```
SELECT DISTINCT title, COUNT(*)
FROM ((film NATURAL JOIN inventory) NATURAL JOIN rental)
WHERE DATEDIFF(return_date, rent_date)>rental_duration
GROUP BY title
ORDER BY COUNT(*) DESC;
```

8. **Selecionar o nome das categorias para as quais existem mais filmes na base de dados, ordenado alfabeticamente pelo nome da categoria.**

```
SELECT name
FROM (SELECT name, COUNT(*)
    FROM category NATURAL JOIN film
    GROUP BY category_id
        ORDER BY COUNT(*) DESC LIMIT 1) AS AUX;
```