

21. Liste os filmes que nunca foram alugados.

-- Condição extra: Inclua o custo de substituição desses filmes.

```
SELECT * FROM aluguel          -- [inventario_id]
SELECT * FROM inventario       -- [inventario_id],[filme_id]
SELECT * FROM filme            -- [filme_id],titulo , custo de sub
-- resolvendo
SELECT DISTINCT filme.titulo, filme.custo_de_substituicao
FROM aluguel
LEFT JOIN inventario ON inventario.inventario_id = aluguel.inventario_id
JOIN filme ON filme.filme_id = inventario.filme_id
WHERE aluguel.aluguel_id IS NULL;
```

22. Encontre os clientes que realizaram aluguéis consecutivos no mesmo dia.

-- Condição extra: Exiba o total de aluguéis consecutivos por cliente.

```
SELECT * FROM aluguel          -- DATE(data), [cliente_id], COUNT(alugel id)
SELECT * FROM cliente          -- [cliente_id], CONCAT
-- resolvendo
SELECT CONCAT(cliente.primeiro_nome,' ',cliente.ultimo_nome) AS nome, DATE(aluguel.data_de_aluguel) AS
dia, COUNT(aluguel.aluguel_id) AS quantidade
FROM aluguel
JOIN cliente ON cliente.cliente_id = aluguel.cliente_id
GROUP BY dia, aluguel.cliente_id
HAVING quantidade>1           -- essa resolução ficou linda pode falar skaks
ORDER BY nome, dia;
```

23. Liste as lojas que possuem mais de 500 itens em inventário.

-- Condição extra: Inclua o total de itens por loja.

```
SELECT * FROM inventario
-- resolvendo
SELECT loja_id, COUNT(inventario_id) AS quantidade
FROM inventario
GROUP BY loja_id
HAVING quantidade > 500;
```

24. Encontre os 5 filmes com maior receita gerada por pagamentos.

-- Condição extra: Exiba o total de receita gerada por cada filme.

```
SELECT * FROM pagamento        -- SUM(valor), [aluguel_id]
SELECT * FROM aluguel          -- [inventario_id],[aluguel_id]
SELECT * FROM inventario       -- [inventario_id],[filme_id]
SELECT * FROM filme            -- [filme_id],titulo
-- resolvendo
SELECT filme.titulo, SUM(pagamento.valor) AS receita
FROM pagamento
JOIN aluguel ON aluguel.aluguel_id = pagamento.aluguel_id
JOIN inventario ON inventario.inventario_id = aluguel.inventario_id
JOIN filme ON filme.filme_id = inventario.filme_id
GROUP BY filme.filme_id
ORDER BY receita DESC
LIMIT 5;
```

25. Liste os clientes que realizaram aluguéis de filmes de categorias diferentes no mesmo dia.

-- Condição extra: Inclua o nome das categorias.

```
SELECT * FROM aluguel          -- dia, [inventario_id],[cliente_id]
SELECT * FROM inventario       -- [inventario_id],[filme_id]
SELECT * FROM filme_categoria -- [filme_id], (DISTINCT categoria_id)
SELECT * FROM categoria        -- [categoria_id], nome
SELECT * FROM cliente          -- [cliente_id], CONCAT(nome)
# acho q vou ter q fazer com subconsulta em, vamos ver
-- resolvendo
SELECT dia, cliente, GROUP_CONCAT(DISTINCT categoria ORDER BY categoria SEPARATOR ', ') AS
categorias
FROM (SELECT
        DATE(aluguel.data_de_aluguel) AS dia ,
        categoria.nome AS categoria,
        CONCAT(cliente.primeiro_nome,' ',cliente.ultimo_nome) AS cliente
FROM aluguel
JOIN inventario ON inventario.inventario_id = aluguel.inventario_id
```

```

        JOIN filme_categoria ON filme_categoria.filme_id = inventario.filme_id
        JOIN categoria ON categoria.categoria_id = filme_categoria.categoria_id
        JOIN cliente ON cliente.cliente_id = aluguel.cliente_id) AS sub
GROUP BY dia, cliente
HAVING COUNT(DISTINCT categoria)>1;          -- fiquei orgulhoso dessa

### 26. Liste as categorias com mais de 20 filmes.
-- Condição extra: Inclua o total de filmes por categoria.
SELECT * FROM filme_categoria                -- count(filme_id) , [categoria_id]
SELECT * FROM categoria                      -- nome , [categoria_id]
-- resolvendo
SELECT categoria.nome AS categoria, COUNT(filme_categoria.filme_id) AS quantidade_filme
FROM filme_categoria
JOIN categoria ON categoria.categoria_id = filme_categoria.categoria_id
GROUP BY categoria
HAVING quantidade_filme > 20;

### 27. Qual foi o dia com maior número de aluguéis realizados?
-- Condição extra: Exiba o total de aluguéis realizados nesse dia.
SELECT * FROM aluguel
-- resolvendo
SELECT CONCAT(DATE(data_de_aluguel),' foi o dia com mais aluguéis') AS dia
FROM aluguel
GROUP BY dia
ORDER BY COUNT(aluguel_id)
LIMIT 1;

### 28. Liste os atores que atuaram em filmes com classificação "PG-13" e "R".
-- Condição extra: Exiba o total de filmes por ator.
SELECT * FROM filme                        -- [filme_id] , [classificacao]
SELECT * FROM filme_ator                  -- [filme_id] , [ator_id]
SELECT * FROM ator                        -- NOME, [ator_id]
-- resolvendo
SELECT CONCAT(ator.primeiro_nome,' ',ator.ultimo_nome) AS ator , filme.classificacao, COUNT(filme.filme_id)
AS quantidade
FROM filme
JOIN filme_ator ON filme_ator.filme_id = filme.filme_id
JOIN ator ON ator.ator_id = filme_ator.ator_id
WHERE filme.classificacao IN ('PG-13','R')
GROUP BY filme.classificacao, filme_ator.ator_id;

### 29. Encontre as cidades que possuem clientes ativos e inativos.
-- Condição extra: Exiba o total de clientes por status em cada cidade.
SELECT * FROM cliente                    -- [endereco_id], *ativo*, COUNT(cliente_id)
SELECT * FROM endereco                  -- [endereco_id], [cidade_id]
SELECT * FROM cidade                    -- [cidade_id], cidade
-- resolvendo
SELECT cidade.cidade,COUNT(CASE WHEN cliente.ativo = 1 THEN 1 END) AS ativos,COUNT(CASE WHEN
cliente.ativo = 0 THEN 1 END) AS inativos
FROM cliente
JOIN endereco ON endereco.endereco_id = cliente.endereco_id
JOIN cidade ON cidade.cidade_id = endereco.cidade_id
GROUP BY cidade.cidade
ORDER BY ativos DESC;

### 30. Liste os filmes que possuem duração maior que a média e são da categoria "Drama".
-- Condição extra: Exiba o título e a duração dos filmes.
SELECT * FROM filme                      -- titulo, [filme_id], ((duracao))
SELECT * FROM filme_categoria            -- [filme_id],[categoria_id]
SELECT * FROM categoria                  -- [categoria_id], ((nome = Drama))
-- resolvendo
SELECT filme.titulo, filme.duracao_do_filme AS duração
FROM filme
JOIN filme_categoria ON filme_categoria.filme_id = filme.filme_id
JOIN categoria ON categoria.categoria_id = filme_categoria.categoria_id
WHERE
        filme.duracao_do_filme > (SELECT AVG(filme.duracao_do_filme) FROM filme)
        AND categoria.nome = 'Drama'
ORDER BY duração DESC;

```