

1. Altere o comentário Uau! Tooke my breath away!!! para Uau! Took my breath away!!!

```
UPDATE avaliacao
```

```
SET coment = 'Uau! Took my breath away!!!'
```

```
WHERE coment = 'Uau! Tooke my breath away!!!';
```

2. Insira o comentário Super fã dos filmes Tubarão do usuário jaws@gmail.com para o filme Jaws. Ele deu 5 estrelas.

```
INSERT INTO avaliacao (codf, email, nestrelas, coment)
```

```
VALUES (
```

```
(SELECT codf FROM filme WHERE titulo = 'Jaws'),
```

```
'jaws@gmail.com',
```

```
5,
```

```
'Super fã dos filmes Tubarão'
```

```
);
```

3. Execute o seguinte insert: insert into elenco (codf, codp, papel, oscar) values (30,177,5,'N'); Analise a mensagem de erro e conserte o problema para que o comando acima seja re-executado da mesma forma que está escrito

primeiro devemos adicionar uma pessoa para o código 177

```
INSERT INTO pessoa (codp, nome, dtnasc, paisn) VALUES (177, 'whashitson',  
'1990-01-01', 'Brasil');
```

```
INSERT INTO elenco (codf, codp, codpapel, oscar) values (30,177,5,'N');
```

4. Retornar os nomes dos filmes com os nomes dos respectivos diretores

```
SELECT f.titulo, p.nome AS diretor
```

```
FROM filme f
```

```
INNER JOIN pessoa p ON f.codd = p.codp;
```

5. Retornar os nomes dos filmes dirigidos por Alfred

```
SELECT f.titulo
```

```
FROM filme f
```

```
INNER JOIN pessoa p ON f.codd = p.codp
```

```
WHERE p.nome = 'Alfred';
```

6. Retorne os nomes dos atores que já receberam um oscar (oscar='Y').

```
SELECT p.nome AS ator
```

```
FROM elenco e
```

```
INNER JOIN pessoa p ON e.codp = p.codp
```

```
WHERE e.oscar = 'Y';
```

7. Retorne os nomes, anos de lançamento, duração dos filmes bem com os nomes dos atores que fizeram parte do elenco

```
SELECT f.titulo, f.anol, f.dur, p.nome AS ator
FROM filme f
INNER JOIN elenco e ON f.codf = e.codf
INNER JOIN pessoa p ON e.codp = p.codp;
```

8. Retornar os nomes dos filmes que receberam 5 estrelas na avaliação

```
SELECT f.titulo
FROM filme f
INNER JOIN avaliacao a ON f.codf = a.codf
WHERE a.nestrelas = 5;
```

9. Retornar os nomes dos filmes que o ator Steven fez parte do elenco

não existe a pessoa steven

```
SELECT f.titulo
FROM filme f
INNER JOIN elenco e ON f.codf = e.codf
INNER JOIN pessoa p ON e.codp = p.codp
WHERE p.nome = 'Stephen';
```

10. Retornar os nomes de todos os atores cadastros e os filmes que eles foram elenco, caso existirem.

```
SELECT p.nome AS ator, f.titulo AS filme
FROM pessoa p
LEFT JOIN elenco e ON p.codp = e.codp
LEFT JOIN filme f ON e.codf = f.codf;
```

11. Retornar os nomes das pessoas que não participaram de nenhum filme

```
SELECT nome
FROM pessoa
WHERE NOT EXISTS (
    SELECT 1
    FROM elenco
    WHERE pessoa.codp = elenco.codp
);
```

12. Retornar, caso existe, um filme que não possua nenhuma avaliação

```
SELECT titulo
FROM filme
WHERE NOT EXISTS (
  SELECT 1
  FROM avaliacao
  WHERE filme.codf = avaliacao.codf
);
```

13. Retornar o nome e a duração dos filmes que foram lançados nos anos 1960

```
SELECT titulo, dur
FROM filme
WHERE anul >= 1960 AND anul <= 1969;
```

14. Retornar os nomes dos atores que já tiveram o papel de figur em qualquer filme

```
SELECT p.nome AS ator
FROM pessoa p
INNER JOIN elenco e ON p.codp = e.codp
INNER JOIN papel pa ON e.codpapel = pa.codpp
WHERE pa.descr = 'Figurante';
```

15. Retornar quantos comentários o usuário cinefilo@gmail.com fez.

não existe cinefilo@gmail.com mas sim cinefilo@yahoo.com

```
SELECT COUNT(*) AS quantidade_comentarios
FROM avaliacao
WHERE email = 'cinefilo@gmail.com';
```

16. Retornar o país e a quantidade de atores cadastrados nele

```
SELECT paisn AS país, COUNT(*) AS quantidade_atores
FROM pessoa
GROUP BY paisn;
```

17. Retornar a pessoa mais velha cadastrada (pode existir mais de 1)

```
SELECT codp, nome, dtnasc
FROM pessoa
WHERE dtnasc = (SELECT MAX(dtnasc) FROM pessoa);
```

18. Retornar o filme com a menor duração (pode existir mais de 1)

```
SELECT codf, titulo, dur
FROM filme
WHERE dur = (SELECT MIN(dur) FROM filme);
```

19. Retornar o nome do filme e número de atores no elenco

```
SELECT f.titulo AS nome_filme, COUNT(e.codp) AS numero_atores
FROM filme f
LEFT JOIN elenco e ON f.codf = e.codf
GROUP BY f.codf, f.titulo;
```

20. Retornar o nome do filme e a média de número de estrela recebidas

```
SELECT f.titulo AS nome_filme, AVG(a.nestrelas) AS media_estrelas
FROM filme f
LEFT JOIN avaliacao a ON f.codf = a.codf
GROUP BY f.codf, f.titulo;
```

