

EXERCÍCIO 1

a. ALTER TABLE Actors DROP COLUMN salary;

RESPOSTA: Esse comando altera a tabela Actor apagando a coluna 'salary'

b. ALTER TABLE Actors CHANGE gender sex VARCHAR(6);

RESPOSTA: Esse comando altera a tabela Actor, mudando o nome da coluna 'gender' para 'sex' e definindo que no máximo ela vai ter 6 caracteres.

c. ALTER TABLE Actors CHANGE gender gender VARCHAR(255);

RESPOSTA: Esse comando altera a tabela Actor, definindo que a coluna 'gender' passa a ter um limite máximo de 255 caracteres.

**** d.**** Agora, altere a coluna gender da tabela ACTOR para que ele aceite strings com até 100 caracteres.

RESPOSTA:

ALTER TABLE Actors CHANGE gender gender VARCHAR(100);

****EXERCÍCIO 2 ****

a. Escreva uma query que atualize o nome e a data de nascimento do ator com o id **003** .

RESPOSTA:

UPDATE Actor

SET

name = "Moacyr Franco",

birth_date = "1929-10-05"

WHERE id = "003"

b. Escreva uma query que atualize o nome da atriz **Juliana Paes** para **JULIANA PÃES** . Então, escreva outra query para voltar ao nome anterior.

RESPOSTA:

UPDATE Actor

SET name = "Juliana Pães"

WHERE name = "Juliana Paes"

UPDATE Actor

SET name = "Juliana Paes"

WHERE name = “Juliana Pães”

c. Escreva uma query que atualize todas as informações do ator com o id **005** .

RESPOSTA:

UPDATE Actor

SET

name = “Rodrigo Santoro”,

birth_date = “1979-07-20”,

salary = 800000,

gender = “male”

WHERE id = “005”;

d. Escreva uma query em que você tente atualizar um dado da tabela que não existe (com um id inválido, por exemplo).

Teste, anote e explique o resultado.

RESPOSTA:

UPDATE Actor

SET

name = “Rodrigo Santoro”,

birth_date = “1979-07-20”,

salary = 800000,

gender = “male”

WHERE id = “010”;

Ele não deu um erro, ele criou um novo ator na lista usando o que passei de informações para alteração. Acredito que ele considera uma alteração (UPDATE) da tabela a inclusão de uma nova linha.

EXERCÍCIO 3

a. Escreva uma query que apague a atriz com o nome **Fernanda Montenegro** .

RESPOSTA:

DELETE from Actor

WHERE name = “Fernanda Montenegro”;

b. Escreva uma query que apague todos os atores (do gênero **male**) com o salário maior do que R\$1.000.000,00

RESPOSTA:

DELETE from Actor

WHERE

gender = "male", AND

salary >= 1000000;

EXERCÍCIO 4

a. Escreva uma query que pegue o maior salário de todos os atores e atrizes

RESPOSTA:

```
SELECT MAX(salary) FROM Actor;
```

b. Escreva uma query que pegue o menor salário das atrizes

RESPOSTA:

```
SELECT MIN(salary)
FROM Actor WHERE gender = "female";
```

c. Escreva uma query que pegue a quantidade de atrizes

RESPOSTA:

```
SELECT COUNT(*) FROM Actor WHERE gender = "female";
```

d. Escreva uma query que pegue a soma de todos os salários

```
SELECT SUM(salary) FROM Actor
```

EXERCÍCIO 5

a. Releia a última query. Teste-a. Explique o resultado com as suas palavras

```
SELECT COUNT(*), gender
```

```
FROM Actor
```

```
GROUP BY gender;
```

RESPOSTA: Essa query me devolve o numero de mulheres e homens, separados por gênero.

b. Faça uma query que retorne somente o id e o nome dos atores em ordem decrescente alfabética

RESPOSTA:

```
SELECT id, name FROM Actor
```

```
ORDER BY name DESC;
```

c. Faça uma query que retorne todos os atores ordenados pelo salário

RESPOSTA:

```
SELECT * FROM Actor  
ORDER BY salary;
```

d. Faça uma query que retorne os atores com os três maiores salários

RESPOSTA:

```
SELECT * FROM Actor  
ORDER BY salary DESC  
LIMIT 3;
```

e. Faça uma query que retorne a média de salário por gênero

RESPOSTA:

```
SELECT AVG(salary), gender FROM Actor  
GROUP BY gender;
```

EXERCÍCIO 6

a. Altere a tabela de **Movie** e adicione um novo parâmetro: **playing_limit_date** que indique a data limite em que o filme será passado no cinema.

RESPOSTA:

```
ALTER TABLE Movie ADD playing_limit_date DATE;
```

b. Altere a tabela de **Movie** para que o parâmetro **rating** possa aceitar valores não inteiros, como, por exemplo, uma avaliação **8.5**.

RESPOSTA:

```
ALTER TABLE Movie CHANGE rating rating FLOAT;
```

c. Atualize dois filmes de tal forma que tenhamos um que ainda esteja em cartaz e um que já tenha saído

RESPOSTA:

```
UPDATE Movie  
SET playing_limit_date = "2020-12-20"  
WHERE id = "002";
```

```
UPDATE Movie  
SET playing_limit_date = "2020-01-01"  
WHERE id = "001";
```

d. Delete algum dos filmes, mas guarde o id. Tente fazer uma query para atualizar a sinopse desse filme que você acabou de deletar (usando o mesmo id). Anote o resultado e explique.

RESPOSTA:

```
DELETE title, synopsis, release_Date, rating, playing_limit_date FROM Movie WHERE id = "003";
```

(testei mas não deu certo, não entendi como fazer isso)

EXERCÍCIO 7

a. Quantos filmes em cartaz possuem avaliações maiores do que 7.5 ?

RESPOSTA: 3

```
SELECT COUNT(*)
```

```
FROM Movie
```

```
WHERE rating > 7.5;
```

b. Qual a média das avaliações dos filmes?

RESPOSTA: 8,75

```
SELECT AVG(rating)
```

```
FROM Movie;
```

c. Qual a quantidade de filmes em cartaz?

RESPOSTA: 1

```
SELECT COUNT(*)
```

```
FROM Movie
```

```
WHERE playing_limit_date >= "2020-06-09";
```

d. Qual a quantidade de filmes que ainda irão lançar?

RESPOSTA: 0

```
SELECT COUNT(*)
```

```
FROM Movie
```

```
WHERE release_Date >= "2020-06-09";
```

e. Qual a maior nota dos filmes?

RESPOSTA: 10

```
SELECT MAX(rating)
```

```
FROM Movie;
```

f. Qual a menor nota dos filmes?

RESPOSTA: 7

```
SELECT MIN(rating)
FROM Movie;
```

EXERCÍCIO 8

Escreva **uma** query que:

a. Retorne todos os filmes em ordem alfabética

RESPOSTA:

```
SELECT *
FROM Movie
ORDER BY title ASC;
```

b. Retorne os 5 primeiros filmes em ordem decrescente alfabética

RESPOSTA:

```
SELECT *
FROM Movie
ORDER BY title DESC
LIMIT 5;
```

c. Retorne os 3 filmes mais recentes em cartaz

RESPOSTA:

```
SELECT *
FROM Movie
ORDER BY release_Date DESC
LIMIT 3;
```

d. Retorne os 3 filmes melhor avaliados

RESPOSTA:

```
SELECT *
FROM Movie
ORDER BY rating DESC
LIMIT 3;
```