

Comando e prática SQL

```
CREATE TABLE Curso ( idCurso INT NOT NULL, Curso VARCHAR(45) NOT NULL, PRIMARY KEY (idCurso))
CREATE TABLE Curso_Funcionario ( idCursofuncionario INT NOT NULL, idCurso INT NOT NULL, idFuncionario INT NOT NULL, oferta DATE, aprovado CHAR, PRIMARY KEY (idCursofuncionario), CONSTRAINT FOREIGN KEY (idFuncionario) REFERENCES Funcionario (idFuncionario), CONSTRAINT FOREIGN KEY (idCurso) REFERENCES Curso (idCurso))
INSERT INTO curso_funcionario VALUES (1,3,123,'2018-05-30','S'), (2,3,152,'2018-05-30','N'), (3,3,222,'2018-05-30','S'), (4,1,123,'2018-10-20','S'), (5,1,152,'
```

Aqui, estamos criando duas tabelas: Curso e Curso_Funcionario, e inserindo alguns dados na tabela Curso_Funcionario.

```
SELECT DATE_FORMAT('2018-03-31', '%e %M %Y')
SELECT DATE_FORMAT('2018-03-29', '%d/%m/%Y')
SELECT TIME_FORMAT('15:40:00', '%Hh %im %ss')
SELECT CURRENT_DATE()
SELECT CURRENT_TIME()
SELECT NOW()
SELECT DATEDIFF(CURRENT_DATE(), '2018-02-08')
```

SELECT DATE_FORMAT('2018-03-31', '%e %M %Y');

Isso é como pedir para um mágico transformar uma data em palavras bonitas! Ele pega a data '31 de março de 2018' e a mostra de um jeito especial: ele escreve o dia, o mês e o ano em um formato mais legal, como '31 março 2018'.

SELECT DATE_FORMAT('2018-03-29', '%d/%m/%Y');

Agora, o mágico está fazendo o mesmo truque, mas de um jeito um pouco diferente. Ele pega a data '29 de março de 2018' e a escreve de uma forma diferente, como '29/03/2018'.

SELECT TIME_FORMAT('15:40:00', '%Hh %im %ss');

Aqui, o mágico está transformando uma hora em palavras legais! Ele pega o horário '15:40:00' e o mostra de uma forma legal, como '15h 40min 00seg'.

SELECT CURRENT_DATE();

Isso é como olhar para um relógio e ver a data de hoje! Ele mostra a data atual, tipo '13 de junho de 2024'.

SELECT CURRENT_TIME();

Agora, o mágico está olhando para o relógio para ver a hora atual! Ele mostra o horário exato agora, como '15:00:00'.

SELECT NOW():

Aqui, o mágico está realmente juntando a data e a hora atual em uma única mágica! Ele diz algo como '13 de junho de 2024, às 15:00:00'.

SELECT DATEDIFF(CURRENT_DATE(), '2018-02-08');

Isso é como fazer uma subtração com datas! Ele está vendo quantos dias se passaram desde 8 de fevereiro de 2018 até hoje. É tipo contar quantos dias tem entre duas datas.

MySQL: Funções DATE_FORMAT e TIME_FORMAT

```
SELECT DATE_FORMAT('2018-03-31', '%e %M %Y'); -- 31 março 2018
```

```
SELECT DATE_FORMAT('2018-03-29', '%d/%m/%Y'); -- 29/03/2018
```

```
SELECT TIME_FORMAT('15:40:00', '%Hh %im %ss'); -- 15h 40m 00s
```

```
SELECT CURRENT_DATE(); -- 2018-11-29
```

```
SELECT CURRENT_TIME(); -- hora atual
```

```
SELECT NOW(); -- 2018-11-29 15:40:00
```

```
SELECT DATEDIFF(CURRENT_DATE(), '2018-02-08'); -- valor em dias
```

```
SELECT * FROM curso_funcionario where oferta between '2018-01-01' and '2018-07-31'
```

```
SELECT * FROM curso_funcionario where oferta > '2018-07-31'
```

```
SELECT * FROM curso_funcionario WHERE oferta LIKE '%2018-11%'
```

SELECT * FROM curso_funcionario where oferta between '2018-01-01' and '2018-07-31':

Esse código está selecionando todas as linhas da tabela curso_funcionario onde a data de oferta está entre 1 de janeiro de 2018 e 31 de julho de 2018. Ou seja, ele está buscando os cursos oferecidos dentro desse período de tempo.

SELECT * FROM curso_funcionario where oferta > '2018-07-31':

Este código está selecionando todas as linhas da tabela curso_funcionario onde a data de oferta é posterior a 31 de julho de 2018. Em outras palavras, ele está buscando cursos oferecidos depois dessa data.

SELECT * FROM curso_funcionario WHERE oferta LIKE '%2018-11%':

Aqui, estamos selecionando todas as linhas da tabela curso_funcionario onde a data de oferta contém "2018-11". Basicamente, ele está buscando cursos oferecidos especificamente no mês de novembro de 2018.

```

SELECT COUNT(*), idFuncionario FROM curso_funcionario GROUP BY (idfuncionario)
SELECT COUNT(*), idFuncionario FROM curso_funcionario GROUP BY (idfuncionario)
HAVING idfuncionario=123
SELECT COUNT(*), idFuncionario FROM curso_funcionario GROUP BY (idfuncionario)
HAVING idfuncionario=123

```

SELECT COUNT(*), idfuncionario FROM curso_funcionario GROUP BY (idfuncionario):

Imagine que temos uma lista de pessoas que fizeram cursos. Este código está contando quantos cursos cada pessoa fez. Ele agrupa os cursos por pessoa (idfuncionario) e conta quantos cursos cada pessoa fez.

SELECT COUNT(*), idFuncionario FROM curso_funcionario GROUP BY (idFuncionario) HAVING idfuncionario=123:

Agora, se queremos saber quantos cursos uma pessoa específica fez, digamos a pessoa com o ID 123. Este código conta quantos cursos a pessoa com o ID 123 fez. Ele agrupa os cursos por pessoa (idFuncionario) e conta quantos cursos essa pessoa fez, mas apenas para a pessoa com o ID 123.

```

SELECT * FROM curso_funcionario limit 2
SELECT * FROM funcionario limit 2,8
SELECT * FROM curso_funcionario limit 5
SELECT * FROM funcionario limit 5

```

Aqui, estamos limitando o número de linhas retornadas em consultas às tabelas Curso_Funcionario e Funcionario.

```

SELECT f.Nome,f.sobrenome, d.dNome FROM Funcionario f INNER JOIN Departamento d ON
f.idDepartamento= d.idDepartamento
SELECT * FROM Departamento d LEFT JOIN Funcionario f ON d.idDepartamento =
f.idDepartamento

```

Imagine que temos duas listas. Uma lista tem nomes de pessoas que gostam de comer doces, e a outra lista tem nomes de diferentes tipos de doces, como bolo, sorvete, e chocolate.

Agora, queremos saber qual pessoa gosta de qual tipo de doce, então precisamos juntar essas duas listas para descobrir.

1. Primeiro, olhamos para a lista de pessoas e vemos o nome de uma pessoa.
2. Em seguida, olhamos para a lista de tipos de doces e perguntamos "Que tipo de doce essa pessoa gosta?".
3. Então, encontramos o tipo de doce preferido dessa pessoa.
4. Repetimos isso para cada pessoa na lista.

O SQL é como um ajudante que faz isso automaticamente para nós. Ele olha para as informações das pessoas e dos tipos de doces e nos mostra em uma lista quem gosta de qual tipo de doce. Assim, podemos descobrir facilmente quem gosta de bolo, quem prefere sorvete, e assim por diante.

