

MANIPULAÇÃO DE DATAS

Prof. Ewerton

Datas

No MySQL, você pode trabalhar com datas usando o tipo de dados DATE, TIME, DATETIME, TIMESTAMP e YEAR. Esses tipos de dados permitem armazenar e manipular informações relacionadas a datas e horas. Aqui estão algumas informações sobre o uso de datas no MySQL:

1. Tipo de dados DATE: Armazena uma data no formato 'YYYY-MM-DD'. Por exemplo, '2023-05-17'.

2. Tipo de dados TIME: Armazena um horário no formato 'HH:MM:SS'. Por exemplo, '09:30:00'.
3. Tipo de dados DATETIME: Armazena uma data e um horário combinados no formato 'YYYY-MM-DD HH:MM:SS'. Por exemplo, '2023-05-17 09:30:00'.
4. Tipo de dados TIMESTAMP: Armazena uma data e um horário combinados, mas tem uma faixa de valores diferente do DATETIME. O TIMESTAMP também pode ser usado para registrar automaticamente a data e hora da inserção ou atualização de um registro, usando a função CURRENT_TIMESTAMP.
5. Tipo de dados YEAR: Armazena um ano de quatro dígitos no formato 'YYYY'. Por exemplo, '2023'.

Exemplos de uso de datas no MySQL

Inserindo uma data em uma tabela

```
INSERT INTO tabela (data) VALUES ('2023-05-17');
```

Selecionando registros com base em uma data:

```
SELECT * FROM tabela WHERE data = '2023-05-17';
```

Extraindo partes de uma data usando funções

```
SELECT YEAR(data) AS ano, MONTH(data) AS mes, DAY(data) AS dia FROM tabela;
```

Realizando operações aritméticas com datas:

```
SELECT DATE_ADD(data, INTERVAL 1 DAY) AS data_nova FROM tabela;
```

Hora

No MySQL, você pode trabalhar com horas utilizando o tipo de dado TIME e uma variedade de funções e operadores específicos para manipular valores de tempo.

O tipo de dados TIME: Armazena um horário no formato 'HH:MM:SS'. Por exemplo, '09:30:00'.

A função CURTIME() ou CURRENT_TIME() retorna o horário do servidor.

Formatar horas na saída:

O MySQL também oferece a função `TIME_FORMAT` e `DATE_FORMAT()` que permite formatar valores de tempo de acordo com um padrão específico. Por exemplo, para exibir a hora no formato "HH:MM:SS", você pode usar o seguinte comando:

```
SELECT CURTIME() AS HORA, TIME_FORMAT(CURTIME(), '%h - %i - %s') AS HORA_FORMATADA;
```

HORA	HORA_FORMATADA
14:24:27	02 - 24 - 27

Observações: o parâmetro “%” é obrigatório antes de informar os caracteres de formato. Abaixo seguem alguns formatos aceitos:

Especificação	Descrição
%h	Hora (01..12)
%H	Hora (00..23)
%i	Minutos, numérico (00..59)
%s	Segundos (00..59)
%r	Tempo, 12 horas (hh:mm:ss seguido por AM ou Pm)
%T	Horário, 24 horas (hh:mm:ss)

Realizar cálculos com horas

O MySQL fornece várias funções para realizar cálculos com valores de hora. Algumas das funções úteis são:

- `TIME_TO_SEC()`: Converte um valor de tempo para segundos.
- `SEC_TO_TIME()`: Converte um valor de segundos para o formato de hora.
- `TIMEDIFF()`: Calcula a diferença entre dois valores de tempo.
- `ADDDTIME()`: Adiciona um intervalo de tempo a um valor de tempo.
- `SUBTIME()`: Subtrai um intervalo de tempo de um valor de tempo.

Por exemplo, para calcular a diferença entre duas horas na tabela horarios, você pode usar a função TIMEDIFF() da seguinte forma:

```
SELECT id, TIMEDIFF(hora1, hora2) AS diferenca  
FROM horarios;
```

CAST

A função CAST() do MySQL é usada para converter um valor de um tipo de dados para outro tipo de dados específico. A função CAST() aceita dois parâmetros que são o valor a ser convertido e o tipo de dados para o qual o valor precisa ser convertido. Os tipos de dados nos quais um determinado valor pode ser convertido são:

- BINARY[(N)]
- CHAR[(N)]
- DATE
- DATETIME
- DECIMAL[(M[,D])]
- SIGNED [INTEGER]
- TIME
- UNSIGNED [INTEGER]

Por exemplo, se você quiser converter uma string em um número inteiro, você pode usar a seguinte consulta:

```
SELECT CAST('42' AS UNSIGNED);
```

Isso retornará o número inteiro 42. Se você quiser converter um número em uma string, você pode usar a seguinte consulta:

```
SELECT CAST(42 AS CHAR);
```

Isso retornará a string '42'.

SIGNED e UNSIGNED

Em MySQL, os tipos de dados SIGNED e UNSIGNED são usados para armazenar números inteiros com ou sem sinal . O tipo de dados SIGNED pode armazenar números inteiros positivos e negativos, enquanto o tipo de dados UNSIGNED pode armazenar apenas números inteiros positivos. O tamanho do número que pode ser armazenado em um campo SIGNED ou UNSIGNED depende do número de bits que o campo usa. Por exemplo, um campo TINYINT SIGNED usa 8 bits e pode armazenar números inteiros entre -128 e 127, enquanto um campo TINYINT UNSIGNED também usa 8 bits, mas pode armazenar apenas números inteiros entre 0 e 255.

Se você não especificar se um campo deve ser SIGNED ou UNSIGNED, o MySQL assumirá que o campo é SIGNED. Para especificar que um campo deve ser UNSIGNED, você deve usar a palavra-chave UNSIGNED após o tipo de dados. Por exemplo:

```
CREATE TABLE example (  
    id INT UNSIGNED NOT NULL AUTO_INCREMENT,  
    name VARCHAR(255) NOT NULL,  
    PRIMARY KEY (id)  
);
```

Isso criará uma tabela chamada "example" com um campo "id" que é um número inteiro sem sinal.

FUNÇÕES MYSQL

Pesquise para que servem e de exemplos de uso das funções:

- CONCAT
- FORMAT
- LCASE/LOWER
- UCSE/UPPER