



Revisão - Tipos de dados

Transcrição

[0:00] Além de conhecer as entidades que fazem parte de um banco MySQL, uma informação muito importante que você também deve saber são os tipos de dados que o MySQL trabalha.

[0:12] Todos os bancos de dados relacionais - seja ele MySQL, Oracle ou SQL Server, da Microsoft - eles possuem tipos de campos muito parecidos, mas os nomes e as referências desses campos podem mudar de um campo para outro, e aí antes da modelagem, antes de você projetar o seu banco, é legal que você saiba que tipos de dados você tem à disposição para saber como você vai usar na sua modelagem.

[0:42] Vamos começar falando um pouquinho sobre os campos de tipos inteiros.

[0:46] Número inteiros são número que não têm casas decimais, e aí nós temos, em uma SQL, cinco tipos de número inteiros.

[0:54] Eles vão desde o Tinyint, Smallint, Mediumint, Int e Bigint.

[1:08] A diferença desses campos é o número de bytes que cada um armazena dentro do banco.

[1:16] Essa informação inclusive é importante, porque se você tem um número muito pequeno e você cria um campo do tipo bigint, embora você saiba que aquele número inteiro, quando você for gravar na tua tabela, ele não vai pass

de um valor específico, vai ser sempre um valor muito baixo, você vai estar gastando oito bytes para armazenar um número que não vai chegar a um byte.

[1:39] Então, é legal você saber os tipos de campos e o tamanho que eles vão armazenar para você projetar bancos de dados mais otimizados, ok?

[1:53] Claro, então, que esses campos possuem o menor e o maior valor possíveis.

[2:01] O Tinyint vai de -128 a +127; o Smallint vai de -32.768 a 32.767, só que quando a gente cria um campo numérico a gente tem uma propriedade no MySQL chamado Unsigned.

[2:24] O unsigned significa o seguinte: se eu digo que um campo é unsigned, ele não vai ter negativo, ele só vai ter número positivo, e aí o menos do número negativo é uma coisa que ocupa espaço no armazenamento do campo, então, quando eu digo que o número não tem número negativo, o range, ou seja, o espaço que eu tenho armazenado para gravar números positivos aumenta.

[2:55] Então, quando o número é unsigned, o tinyint, que o limite máximo seria 127 passa a ser 255.

[3:05] O smallint, que o valor mínimo era 32.767, passa a ser 65.535.

[3:15] Então há um aumento desse valor do espaço reservado para armazenagem do número inteiro, e isso se aplica ao mediumint, ao int e ao bigint, ok?

[3:33] Agora vamos falar dos números decimais.

[3:37] Nós temos o Float e o Double.

[3:40] A diferença deles é a precisão: uma usa a precisão simples e a outra usa a precisão dupla.

[3:48] Uma ocupa 4 bytes do meu banco e o outro ocupa 8 bytes.

[3:55] O precisão tem muito a ver com a forma com que o número é armazenado, então, às vezes o número, internamente, ele é armazenado com muitas casas decimais, mas a gente, depois, quando visualiza ele, ele faz uma aproximação através desse ponto flutuante no valor decimal que ele realmente tem.

[4:14] Então, se eu digo que um campo, por exemplo, é float de 7 dígitos com 4 casas decimais, ou seja, o primeiro parâmetro do float é o número de dígitos totais do número incluindo os decimais, e o que eu tenho depois da vírgula é só o que eu vou ter na parte decimal.

[4:34] Então, se eu digo que o float é 7 de 4, um número, por exemplo, 999,00009, quando eu armazenar, ele será armazenado como 999,0001, ou seja, ele vai arredondar esse número e armazenar dentro do espaço que foi determinado.

[5:02] Outros números decimais mas que não são de pontos flutuantes é o Decimal ou Numeric, pode ser um ou outro, eles são iguais.

[5:10] Eles armazenam número de até 65 dígitos, incluindo as casas decimais, e esses números são fixos, esse tamanho é fixo, ou seja, quando eu coloco o número lá ele vai ocupar o número de casas decimais até o finalzinho do que o número ocupa, ok?

[5:30] O seu tamanho não é variável, ele já é um tamanho fixo reservado para aquele número, tem uma precisão numérica melhor.

[5:40] O tipo Bit, como o próprio nome diz, é um bit apenas, ok?

[5:46] Mas eu posso ter um campo de bit de 2, de 3, de 4, mas o conteúdo lá dentro vai ser caracteres binários: ou 1 ou 0. Então, se eu digo que é bit de 1, eu posso armazenar ou 1 ou 0, mas se eu digo que o campo é bit de 2 eu posso

armazenar 01, 10, 00 ou 11, ou seja, eu tenho 4 tipos de números binários que eu posso armazenar no bit de 2.

[6:20] Normalmente o bit 1 a gente utiliza para armazenar campos lógicos onde eu tenho alguma coisa que é verdadeira ou falsa, e normalmente a gente aplica o 1 para o verdadeiro e o 0 para o falso.

[6:38] Como eu falei, os campos numéricos possuem atributos, signed ou unsigned, se o número vai ter ou não número negativo.

[6:50] Zerofil, alguns números, geralmente os que não são de ponto flutuante, eu preencho com zeros os espaços antes do número inteiro até chegar ao tamanho do número.

[7:03] Então, um número inteiro que eu armazeno, o número 5, eu vou armazenar o número 0005, ou seja, eu vou colocar 3 zeros na frente de tal maneira que o número de dígitos bata com esse número que eu acrescentei aqui.

[7:21] A gente tem também campos numéricos que são do tipo auto incremento.

[7:26] O auto incremento é um número sequencial que vai ser crescido automaticamente na medida em que eu vou inserindo dados da tabela, e aí além de crescimento desse número sequencial eu posso determinar, pode ser um número que cresça de 1 em 1, de 5 em 5, e assim por diante.

[7:46] O auto incremento ele é muito usado quando eu quero criar chaves primárias randômicas, que não vão se repetir nunca.

[7:54] Randômicas, não, sequenciais.

[7:57] Randômicas a gente estaria falando de coisas aleatórias, mas não, o auto incremento respeita um critério, e aí eu sei que todo insert que eu fizer dentro

dessa tabela, ou seja, toda inclusão de dados, nunca esse campo vai se repetir, porque ele é do tipo auto incremento.

[8:16] Quando a gente tenta armazenar um número maior do que ele suporta, eu digo que o número é um tinyint ou smallint e eu quero gravar um número grande, eu sempre vou ter erro no momento da gravação, que são os erros chamados de out of range.

[8:33] Então, quando você estiver trabalhando com SQL e ver um erro do tipo out of range, já sabe que é alguma coisa que você quer gravar que é maior do que é suportado.

[8:46] A gente tem os campos de data, que são superimportantes no banco de dados, e aí eu posso ter tipos de bancos diferentes.

[8:55] Date é um campo de data que somente vai armazenar ano, mês e dia, e varia do dia 1 de janeiro do ano 1000 até o último dia de dezembro de 9.999.

[9:16] O date time é igual ao date, só que além da data, eu armazeno a hora, então eu posso, também, ter uma precisão maior do dado armazenado, e aqui em cima vocês podem ver o limite inferior e superior.

[9:33] O timestamp é parecidíssimo com o date time, só que ele tem um limite menor e ele utiliza o fuso horário para estar armazenando a informação.

[9:45] Normalmente, a informação vem com o fuso horário de Greenwich, que é o fuso horário padrão do mundo, o fuso zero, e depois eu posso também trabalhar com fusos horários para poder gravar, por exemplo, dados com fuso horário do Brasil, outro com fuso horário dos Estados Unidos, e assim por diante.

[10:07] Aplicações do tipo timestamp normalmente são aplicações em que você envolve, por exemplo, um workflow ou alguma comunicação entre pessoas que estão, por exemplo, em países diferentes.

- [10:19] O time ele só armazena a hora, mas não é a hora do relógio, é um valor crescente de hora, que tem os limites ali especificados.
- [10:31] E o campo do tipo year ele armazena só ano, e ele varia de 1.901 a 2.155.
- [10:40] E o campo year pode ser expresso com 2 dígitos ou com 4 dígitos.
- [10:47] Os campos string nós chamamos de dois: varchar e char.
- [10:55] O char tem valor fixo, ou seja, quando a gente diz que o char tem 255 caracteres, se eu gravar um caracter com 1 dígito apenas, por exemplo a letra A, ele vai ocupar os 255 dígitos.
- [11:13] Ele vai gravar o A e 254 espaços em branco.
- [11:18] Já o varchar, não.
- [11:19] O varchar, apesar de eu dizer que há um limite máximo de tamanho, se eu gravo a letra A nesse exemplo e o campo tem 255 dígitos, eu só gravo 1 dígito na base.
- [11:36] Ou seja, o varchar ele é mais inteligente para controlar tamanhos.
- [11:44] Eu tenho também os valores binários, que também podem variar de 0 a 255 caracteres, porém os dados que são gravados são bits, 1, 0, 0, 1 e assim por diante.
- [11:59] Não consigo ver, ali, letras nem números, só o número 1 e o número 0.
- [12:08] Nós temos alguns strings que nós chamamos de strings longos.
- [12:11] O blob é um string desse tipo, e tem alguns tipos: tinyblob, blob, mediumblob, longblob.

[12:20] E o texto, que é o texto longo.

[12:24] O blob tem a ver com binários e o blob com strings.

[12:29] Com texto propriamente dito, claro, como o próprio nome diz.

[12:35] O blob eu uso para gravar coisas binárias, então, por exemplo, eu posso gravar o conteúdo de uma foto, de um arquivo, que é binário, no campo blob.

[12:47] Já no campo texto é só caracteres que respeitam as tabelas ASCIIIS dependendo de cada tabela que o computador suporta, que são tabelas que convertem números em letras.

[13:02] O enum são campos que me dão opções, então, por exemplo, eu digo que o campo é do tipo enum e tenho aqui: x-small, small, medium, large, x-large.

[13:16] Ou seja, quando eu dou insert nesse campo, se eu der um string diferente desse conjunto que está aqui, o banco de dados vai me dar um erro, vai dizer que eu não posso.

[13:27] Esses são os campos do tipo enum.

[13:31] Todos os campos string têm um campo chamado set e collate, que têm a ver com o conjunto de caracteres suportados, com aquelas tabelas ASCIIIS que eu falei há alguns minutos atrás.

[13:45] Quando a informática começou a ser difundida, a conversão de caracteres em números binários para serem armazenados no computador respeitavam somente uma tabela muito simples onde o acento, por exemplo, não era suportado.

[14:03] Eu não tinha o caracter A com acento ou Ç.

[14:07] Aí começaram a surgir outras tabelas de armazenamento de dados que englobava, por exemplo, os caracteres latinos, russos, ou, por exemplo, os símbolos islâmicos.

[14:19] Enfim, há um conjunto de tabelas que suporta outros símbolos além dos que você vê no seu teclado.

[14:27] Você pode, ao criar um campo com o MySQL, especificar que conjunto de caracteres você vai estar utilizando para aquele campo.

[14:39] Também, agora, somente nas versões mais recentes do MYSQL, surgiram campos do tipo espacial.

[14:47] Com o advento do uso de mapas, você poder desenhar campos nos mapas, poder traçar áreas para poder dizer: "essa é a minha área de venda", ou eu traço uma área de um Estado para pintar, dizer qual foi o valor da venda daquele Estado, esses desenhos em mapas a gente pode armazenar em banco de dados através dos campos espaciais.

[15:13] Então, o geometric ele armazena uma figura geométrica, ou seja, tem todas as coordenadas dentro desse campo que fazem com que eu consiga desenhar essa figura dentro de um mapa.

[15:26] O point é um ponto, o line string é uma linha onde eu coloco o ponto origem e um ponto destino e ele traça uma linha até lá.

[15:35] E o polígono ele é um detalhamento mais específico da área, que geralmente é fechada.

[15:44] É o tipo do campo polígono.

[15:48] Ok. Então, eram esses os campos que eu queria estar mostrando para vocês antes de realmente a gente começar a entrar no MySQL e começar a criar tabelas, está legal? Valeu, gente, obrigado.

