MySql Tool CTG

Sistema manipulação de banco de dados MySQL

Ajuda / Tutorial

suporte@toolctg.com.br

www.toolctg.com.br

Sumário

- O que é um banco de dados ?

Banco de dados é um sistema que reúne e mantém organizado uma série de informações relacionadas a um determinado assunto. Um bom exemplo de banco de dados é uma lista telefônica, nela observamos todos os dados referentes a uma pessoa, organizados de acordo com as necessidades do usuário.

- Para que serve um banco de dados ?

Um banco de dados serve para armazenar dados que se relacionam, são criados de acordo com a necessidade de cada sistema. É usado por quase todos os programas e é responsável por armazenar os dados que o usuário necessita para uso posterior.

FUNÇÕES INTERNAS DO BANCO DE DADOS

- Definições e Tipos de dados

Os tipos de dados básicos para os atributos incluem numéricos, cadeia de caracteres (string), cadeia de *bits*, booleanos, data e horário.

Numéricos: Tipos de dados numéricos que englobam os números inteiros de vários tamanhos(INTERGER,TINYINT,SMALLINT,MEDIUMINT,BIGINT) e os números pontos flutuantes (reais) de várias precisões (FLOAT, REAL ou DOUBLE).

Os números formatados podem ser declarados pelo uso do NUMERIC (i,j), em que i é o número total de dígitos decimais, e j refere-se ao número de dígitos depois do ponto decimal.

TIPO	TAMANHO	N° COM SINAIS	N° SEM SINAIS
TINYINT	1 byte	-128 a 127	0 a 255
SMALLINT	2 bytes	-32.768 a 32.767	0 a 65.535
MEDIUMINT	3 bytes	-8.388.608 a 8.388.607	0 a 16.777.215
INTERGER	4 bytes	-2.147.683.648 a 2.147.683.647	0 a 4.294.967.295
BIGINT	8 bytes	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	0 a 18.446.744.073.709.551.615
NUMERIC	Variável	Ex : numeric (3) Numeric(5,2)	999.999,999
FLOAT	4 bytes		
DOUBLE	8 bytes		
REAL	8 bytes		

- Cadeia de caracteres:

CHAR(n) - Utiliza-se este tipo de dados quando houver a necessidade de fazer uso de sequências de caracteres de tamanho fixo que estejam limitadas até 255 caracteres de comprimento. O parâmetro tamanho determina o valor máximo em caracteres que pode conter a sequência.

VARCHAR(n) - Utiliza-se este tipo de dados quando houver a necessidade de fazer uso de sequências de caracteres de tamanho variável que estejam limitadas até 255 caracteres de comprimento. A diferença entre esse tipo e o CHAR é que, neste caso, os espaços em branco excedentes do lado direito da sequência de caracteres não utilizados são automaticamente desprezados. O parâmetro tamanho determina o valor máximo em caracteres que pode conter a sequência.

TEXT - Armazena cadeias de caracteres de qualquer comprimento, e não requer um limite superior explicitamente declarado de seu tamanho.

TIPO	TAMANHO EXIGIDO
CHAR (m)	M bytes, 1<= M <= 255
VARCHAR (m)	L+1 bytes, onde L<= M e 1 <= M <=255
TEXT	L + 2 bytes, onde L < 2^16

Date e Time

O tipo *DATE*(data) contém dez posições e seus comportamentos são YEAR (ano), MONTH (mês) e DAY (dia) no formato YYYY:MM:SS.

O tipo *TIME*(hora) tem oito posições com os componentes HOUR(hora), MINUTE(minuto), e SECOND(segundo) no formato HH:MM:SS

TIPO	TAMANHO EXIGIDO	EXEMPLO
DATE	3 bytes	2010-01-20
TIME	3 bytes	15:16:50

- O que são tabelas nos banco de dados?

A tabela é um conjunto de dados dispostos em número finito de colunas e número ilimitado de linhas. As colunas são consideradas os campos da tabela, e caracterizam os tipos de dados que deverão constar na tabela (nome, telefone, e-mail, endereço etc). O número de linhas são as combinações de valores dos campos da tabela, preenchendo as colunas de forma organizada (João,5555-5555,joão@hotmail.com,etc).

Toda tabela deve ter uma chave primária, que é o campo que não pode repetir em nenhuma linha (um código interno criado para diferenciar os dados, CPF, RG, etc).

Exemplo :

```
create table tbl_contatos(

codigo int auto_increment,

nome varchar (255) not null,

telefone bigint,

email varchar(255),

primary key (codigo)

);
```

- O que é uma Procedure?

Conhecida como Stored Procedure é uma coleção de comandos em SQL para facilitar o manuseio do banco de dados. Otimiza tarefas repetitivas, aceita parâmetros de entrada e pode retornar um valor (para indicar aceitação ou falha na execução). O procedimento armazenado pode reduzir o tráfego na rede, melhorar a performance, criar mecanismos de segurança, etc.

Exemplo:

CREATE PROCEDURE *proc_nome* ([*parametros*,...])

[*caracteristicas*]

[BEGIN]

código;

[END]

proc_nome: seu procedimento armazenado deve ter um nome, para, quando for chamado podermos usá-lo.

tipos de parametros: existem três tipos de parâmetros em uma Procedure no MySql:

- IN → este é um parâmetro de entrada, ou seja, um parâmetro cujo valor será utilizado no interior do procedimento para produzir algum resultado.

- OUT → este parâmetro retorna algo de dentro do procedimento para o lado externo, colocando os valores manipulados disponíveis na memória ou no conjunto de resultados.

INOUT → faz os dois trabalhos ao mesmo tempo.

parametros: nesta parte da procedure, informamos os parâmetros da seguinte forma: [IN | OUT | INOUT] nome_parametro tipo_dado.

caracteristicas: as características que o procedimento pode apresentar. Questões de segurança, se são determinísticas ou não, qual é a linguagem que estamos usando e se nossa procedure modificará dados no banco de dados são algumas das características que podemos definir neste item.

código: onde são definidos os comandos SQL que farão alguma manipulação e/ou defenderão alguma lógica, podendo retornar ou não algum resultado.

- O que é uma Trigger?

Trigger é um recurso de programação executado sempre que o evento associado ocorrer, é um tipo especial de procedure, que é executado sempre que há uma tentativa de modificar os dados de uma tabela que é protegida por ele.

É muito utilizada para ajudar a manter a consistência dos dados ou para propagar alterações em um determinado dado de uma tabela para outras. Um bom exemplo é um trigger criado para controle de quem alterou a tabela, nesse caso, quando a alteração for efetuada, a trigger é "disparada" e grava em uma tabela de histórico de alteração, o usuário e data/hora da alteração.

CREATE TRIGGER *tgr_nome* ON *tbl_nome*

[*caracteristicas*] [*ativação*] (ex: AFTER INSERT)

AS

[*código*]

*tgr_nome***:** sua trigger armazenada deve ter um nome, para, quando for chamado podermos usá-lo.

*tbl_nome***:** a tabela que será "protegida" pela trigger, assim que ocorrer uma mudança nessa tabela, a trigger será disparada.

*caracteristicas***:** as triggers são definidas para BEFORE (antes) ou AFTER (depois)

As triggers BEFORE(antes) disparam antes de as modificações da instrução serem aplicadas, e antes de qualquer restrição ser aplicada.

As triggers AFTER(depois) disparam após todas as restrições terem sido satisfeitas, e após todas as alterações terem sido aplicadas à tabela de destino.

ativação: a ativação da trigger está associada a um evento, sendo eles:

- Insert
- Update
- Delete

código: onde será colocado o código SQL, que fará a manipulação no banco.

Tutorial do programa

- Como criar uma conexão

- Como criar um banco de dados

- Como criar uma tabela

- Como inserir dados em uma tabela

- Como criar Trigger e Procedure

- Como fazer Backup
