

Zerar Tabelas de Vendas

Limpe as tabelas com vendas de testes

04/01/2010 por Mario SAM

A execução de testes é fundamental para qualquer sistema, e inevitavelmente sempre que instalamos uma versão do Magento realizamos diversos testes, entre eles o de vendas.

E após realizar diversos testes de vendas, de executar os ajustes e melhorias, chega o momento de realizar os testes para produção, e seria muito interessante limpar os dados de vendas para que pudéssemos reiniciar nossos testes.

Como o Magento não oferece a opção de apagar registros de vendas, precisamos acessar o banco de dados e executar alguns comandos em SQL.

Existem diversas maneiras de acessar sua base de dados, verifique com sua hospedagem as formas de acesso, geralmente eles oferecem um sistema chamado *phpMyAdmin*. Ou via *SSH* para acessar o prompt de comando do seu MySQL.

Neste exemplo usaremos o comando **TRUNCATE**, que remove todas as linhas da tabela de uma só vez. É a mesma coisa que o comando **DELETE** só que sem a opção de **WHERE** (*condições*). O **TRUNCATE** é mais rápido e utiliza menos recursos do sistema e dos logs de transações.

Depois de apagar os registros usamos o comando **ALTER TABLE** para alterar a sequência de identificação (ID). Sempre que inserimos um novo registro na tabela ele cria um identificador único, o que faremos é apenas reiniciar esse identificador.

O comando SQL para zerar as tabelas de vendas é:

```
SET FOREIGN_KEY_CHECKS=0;

TRUNCATE `sales_order`;

TRUNCATE `sales_order_datetime`;

TRUNCATE `sales_order_decimal`;

TRUNCATE `sales_order_entity`;

TRUNCATE `sales_order_entity_datetime`;

TRUNCATE `sales_order_entity_decimal`;

TRUNCATE `sales_order_entity_int`;

TRUNCATE `sales_order_entity_text`;
```

```
TRUNCATE `sales_order_entity_varchar`;

TRUNCATE `sales_order_int`;

TRUNCATE `sales_order_text`;

TRUNCATE `sales_order_varchar`;

TRUNCATE `sales_flat_quote`;

TRUNCATE `sales_flat_quote_address`;

TRUNCATE `sales_flat_quote_address_item`;

TRUNCATE `sales_flat_quote_item`;

TRUNCATE `sales_flat_quote_item_option`;

TRUNCATE `sales_flat_order_item`;

ALTER TABLE `sales_order` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_datetime` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_decimal` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_entity` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_entity_datetime` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_entity_decimal` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_entity_int` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_entity_text` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_entity_varchar` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_int` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_text` AUTO_INCREMENT=1;

ALTER TABLE `sales_order_varchar` AUTO_INCREMENT=1;

ALTER TABLE `sales_flat_quote` AUTO_INCREMENT=1;

ALTER TABLE `sales_flat_quote_address` AUTO_INCREMENT=1;

ALTER TABLE `sales_flat_quote_address_item` AUTO_INCREMENT=1;
```

```
ALTER TABLE `sales_flat_quote_item` AUTO_INCREMENT=1;

ALTER TABLE `sales_flat_quote_item_option` AUTO_INCREMENT=1;

ALTER TABLE `sales_flat_order_item` AUTO_INCREMENT=1;

SET FOREIGN_KEY_CHECKS=1;
```

Se você nunca ouviu falar em SQL, não altere as linhas do exemplo acima de jeito nenhum. Agora se você tem domínio sobre SQL pode até alterar o comando **TRUNCATE** para **DELETE** e usar a cláusula **WHERE** para deletar registros específicos, sem apagar a tabela inteira.

Qualquer dúvida ou problema utilize o [fórum](#).

Sucesso!