Comandos de SQL da DDL

Prof. Anésio Freire

Dividem-se em três tipos

- Definição de Dados – DDL ou Data Definiton Language
- Manipulação de Dados – DML ou Data Manipulation Language
- Controle de Dados – DCL ou Data Control Language

Comandos da Sub-linguagem DDL

- Conjunto de instruções SQL destinadas a criação de objetos numa base de dados:
  - DOMÍNIOS
  - TABELAS
  - ÍNDICES
  - CHAVES
  - VISTAS
  - RESTRIÇÕES

- Para a criação desses objetos a DDL oferece três comandos;
  - CREATE
  - ALTER
  - DROP

Create Database

Sintaxe CREATE DATABASE nome_bancodedados

```
[ON {
[PRIMARY] (NAME = nome_lógico_arquivo,
        FILENAME = 'caminho_e_nome_arquivo'
        [, SIZE = tamanho]
        [, MAXSIZE = tamanho_máximo]
        [, FILEGROWTH = taxa_crescimento]
        }[,...n]
]
[LOG ON
      {
      (NAME = nome_lógico_arquivo,
      FILENAME = 'caminho_e_nome_arquivo'
      [, SIZE = tamanho])
      }[,..n]
]
```

Onde:

nome_bancodedados é o nome do banco de dados que se deseja criar. **nome_logico_arquivo** é um nome usado para referenciar o arquivo em quaisquer comandos SQL executados depois que o banco de dados tiver sido criado.

PRIMARY: Esta opção especifica o grupo de arquivos primário. O grupo de arquivos primário deve conter todas as tabelas de sistema para o banco de dados. Um banco de dados só pode ter um grupo de arquivo PRIMARY. Se não for especificado algum, o primeiro listado será o primário.

FILENAME: Aqui deve-se especificar o caminho e nome do arquivo que você está criando. O arquivo deve estar localizado na mesma máquina que o servidor SQL Server. Ele pode estar em uma unidade de disco diferente contanto que esteja na mesma máquina.
SIZE: Especifica o tamanho em megabytes que você quer alocar para o seu banco de dados. O valor mínimo é 1MB, e o padrão é 3MB para arquivos de dados, e 1MB para arquivos de log.
MAXSIZE: Esta opção lhe permite especificar o tamanho máximo até o qual seu arquivo pode crescer. O padrão permite que seu arquivo cresça até que o disco esteja cheio.
FILEGROWTH: Especifica a taxa de crescimento do arquivo. Este ajuste não pode exceder a configuração de MAXSIZE. Um valor de 0 indica que não é permitido crescimento. O padrão é 10 por cento, significando que a cada vez que o arquivo cresce, será alocado um espaço adicional de 10 por cento para ele.
Na opção **LOG ON** se aplicam as mesmas definições acima, exceto pelo fato de não ser o arquivo de dados, mas sim o arquivo de log de transações que estará sendo criado. Caso LOG ON seja omitido, é criado um único arquivo de log com um nome gerado pelo sistema e um tamanho que seja 25 por cento da soma dos tamanhos de todos os arquivos de dados para o banco de dados. Nota:

Como exemplo, vamos criar um banco de dados, entrando com o seguinte código SQL no Query Analyzer.

```
CREATE DATABASE Exemplo
ON
PRIMARY (NAME=Exemplo2_data,
FILENAME = 'C:\MSSQL7\DATA\Exemplo2.mdf',
SIZE = 10MB,
MAXSIZE = 15MB,
FILEGROWTH = 25%)
LOG ON
(NAME = Exemplo2_log,
FILENAME = 'C:\MSSQL7\DATA\Exemplo2.ldf',
SIZE = 4MB,
MAXSIZE = 6MB,
FILEGROWTH = 2MB)
```

Comandos de SQL da DDL

Prof. Anésio Freire

Create Table

primary key: indica os atributos que formam a chave primária;
unique key: indica os atributos que formam a chave candidata;
foreign key: indica os atributos que formam a chave estrangeira e o nome da relação referida pela chave estrangeira

Sintaxe:

CREATE TABLE <nome-tabela>
(<nome-coluna> , <tipo-do-dado> [NOT NULL]
[NOT NULL WITH DEFAULT])
PRIMARY KEY (nome-coluna-chave)
FOREIGN KEY (nome-coluna-chave-estrangeira) **REFERENCES**
(nome-tabela-pai) **ON DELETE** [RESTRICT]
[CASCADE]
[SET NULL]

onde:

a) nome-tabela - Representa o nome da tabela que será criada.
b) nome-coluna - Representa o nome da coluna que será criada.
c) tipo-do-dado - Cláusula que define o tipo e tamanho dos campos
d) NOT NULL - Exige o preenchimento do campo
e) NOT NULL WITH DEFAULT - Preenche o campo com valores pré-definidos, de acordo com o tipo do campo, caso não seja especificado o seu conteúdo no momento da inclusão do registro. Os valores pré definidos são:
e.1) Campos numéricos - Valor zero.
e.2) Campos alfanuméricos - Caracter branco.
e.3) Campo formato Date - Data corrente.
e.4) Campo formato Time - Horário no momento da operação.
f) PRIMARY KEY (nome-coluna-chave) - Definir para o banco de dados a coluna que será a chave primária da tabela
g) FOREIGN KEY (nome-coluna-chave-estrangeira) REFERENCES (nome-tabela-pai) - Definir para o banco de dados as colunas que são chaves estrangeiras, ou seja, os campos que são chaves primárias de outras tabelas. Na opção REFERENCES deve ser especificado a tabela na qual a coluna é a chave primária.
h) ON DELETE - Esta opção especifica os procedimentos que devem ser feitos pelo SGBD quando houver uma exclusão de um registro na tabela pai quando existe um registro correspondente nas tabelas filhas. As opções disponíveis são:
h.1) RESTRICT - Opção default. Esta opção não permite a exclusão na tabela pai de um registro cuja chave primária exista em alguma tabela filha.
h.2) CASCADE - Esta opção realiza a exclusão em todas as tabelas filhas que possua o valor da chave que será excluída na tabela pai.
h.3) SET NULL - Esta opção atribui o valor NULO nas colunas das tabelas filhas que contenha o valor da chave que será excluída na tabela pai.

Tipos de dados mais comuns:

1) Numéricos:
- Smallint - Armazena valores numéricos, em dois bytes binários, compreendidos entre o intervalo -32768 a +32767.

- Integer - Armazena valores numéricos, em quatro bytes binários, compreendidos entre o intervalo -2147483648 a +2147483647
- Decimal(n,m) - Armazena valores numéricos com no máximo 15 dígitos. Nesta opção deve ser definida a quantidade de dígitos inteiros (n) e casas decimais (m) existentes no campo.

2) Alfanuméricos:
- Varchar (n) - Definir um campo alfanumérico de até n caracteres, onde n deve ser menor ou igual a 254 caracteres.
- Char (n) - Definir um campo alfanumérico de n caracteres, onde n deve ser menor ou igual a 254 caracteres.
- Long Varchar - Definir um campo alfanuméricos de comprimento maior que 254 caracteres.

3) Campo Date - Definir um campo que irá armazenar datas.
4) Campo Time - Definir um campo que irá armazenamento de horário.

Tipos de Dados

Para dados	**Tipo**	**Tamanho**
Caractere	char(n), varchar(n)	até n bytes
Numérico exato	decimal(p,e) ou numeric(p,e)	-depende-
Numérico aprox.	float, real	8, 4 bytes
Numérico inteiro	int, smallint, tinyint	4, 2, 1 byte
Monetário	money, smallmoney	8, 4 bytes
Data e hora	datetime, smalldatetime	8, 4 bytes
Binário	binary(n), varbinary(n)	n bytes
Texto e imagens	text, image	-variável-
Outros	bit, timestamp	1 bit, 8 bytes

Para dados contendo caracteres, char(n) armazena um número fixo de caracteres. Por exemplo, uma coluna do tipo char(30) tem sempre 30 caracteres. Se forem informados menos, o restante é completado com espaços. Já o tipo varchar(n) armazena uma quantidade variável de caracteres, até o máximo informado.
Os tipos "numéricos exatos", decimal e numeric, permitem armazenar dados exatos, sem perdas devidas a arredondamento. Ao usar esses tipos, você pode especificar uma precisão, que indica quantos dígitos podem ser usados no total e uma escala, que indica quantos dígitos podem ser usados à direita do ponto. Por exemplo, decimal(9,2) permite guardar 7 dígitos antes do ponto decimal e 2 após, num total de 9, assim o maior valor possível é 9999999,99.
Os tipos "numéricos inexatos", float e real, armazenam dados numéricos, mas nem sempre mantém a precisão suficiente para armazenar corretamente números de vários dígitos.
O tipo money é usado para valores monetários, ocupando 8 bytes em disco e permitindo valores entre -922.337.203.685.477,5808 e 922.337.203.685.477,5807 (922 trilhões). O tipo smallmoney permite valores entre - 214.748,3648 e +214.748,3647 (214 mil) e ocupa 4 bytes em disco.
Dos tipos inteiros, int usa 32 bits (4 bytes), permitindo armazenar até +/-2.147.483.647, smallint usa 16 bits (2 bytes) permitindo +/-32767 e tinyint usa 8 bits (1 byte), permitindo números não-negativos de 0 a 255.

O tipo datetime armazena valores contendo a data e hora, com precisão de 1/300 de segundo, entre 1º de janeiro de 1753 e 31 de dezembro de 9999 (o século é sempre armazenado). O tipo smalldatetime ocupa menos espaço e armazena datas e horas de 1º de janeiro de 1900 até 6 de junho de 2079, com precisão de 1 minuto.
Tipos binários são usados para dados que o SQL Server não interpreta, por exemplo, o conteúdo de um arquivo binário.
O tipo text é usado para colunas com dados "memo", ou seja, com texto de tamanho variável; O tipo image armazena imagens, também de tamanho variável.
O tipo bit armazena valor 1 ou 0. Uma coluna do tipo timestamp não pode ser alterada pelo usuário. Ela é definida automaticamente com a data e hora atual quando a linha é inserida ou atualizada.

Exemplo:

```
CREATE TABLE Cliente1
(
CodCliente int NOT NULL,
Nome varchar(50),
CPF varchar(11) NULL,
DataCadastro datetime NOT NULL DEFAULT (getdate()),
Cidade varchar(20) NULL,
UF char(2) NULL,
País varchar(20) DEFAULT ('Brasil')
```

Restrições

Restrições de domínio podem ser implementadas através da clausula CHECK, seguida de expressão booleana delimitada por parênteses. Para a montagem de expressão booleana poderemos utilizar operadores de comparação (>, <, =, >=, <=, <>) e predicados como LIKE e IN.

```
CREATE TABLE Cliente(
        Codigo  INTEGER NOT NULL CHECK(Codigo > 0),
        Nome    VARCHAR(40) NOT NULL,
        Fone    VARCHAR(20),
        Fax     VARCHAR(20),
        Renda   MONEY DEFAULT 0 NOT NULL CHECK(Renda >= 0)
      )
```

Alter Table

Alterar a estrutura de uma tabela(arquivo) acrescentando, alterando, retirando e alterando nomes, formatos das colunas e a integridade referencial definidas em uma determinada tabela.
Sintaxe:

ALTER TABLE <nome-tabela>

DROP <nome-coluna>

ADD <nome-coluna> <tipo-do-dado> [NOT NULL]
[NOT NULL WITH DEFAULT]

ADD PRIMARY KEY <nome-coluna>

DROP PRIMARY KEY <nome-coluna>

ADD FOREIGN KEY (nome-coluna-chave-estrangeira) **REFERENCES**
(nome-tabela-pai) **ON DELETE** [RESTRICT]
[CASCADE]
[SET NULL]

DROP FOREIGN KEY (nome-coluna-chave-estrangeira) **REFERENCES**
(nome-tabela-pai)

Em SQL Server o comando para alterar o nome da uma tabela ou nome de um campo dever ser executada o procedimento:

Exec sp_rename 'tabela.coluna', 'coluna_nova' -- Altera o nome da coluna

Exec sp_rename 'tabela', 'tabela_nova' -- Altera o nome da tabela

Drop Table

Excluir a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão excluídos todos dados, estrutura e índices de acessos que estejam a ela associados.

Sintaxe:

DROP TABLE <nome-tabela>

Create Index

Criar uma estrutura de índice de acesso para uma determinada coluna em uma tabela. Um índice de acesso permite um acesso mais rápido aos dados em uma operação de seleção. Os índices podem ser criados a partir de um ou mais campos de uma tabela.

Sintaxe:

CREATE [UNIQUE] INDEX <nome-índice>
ON <nome-tabela> (<nome-coluna> [ASC], [<nome-coluna> [ASC]])
[DESC] [DESC]

onde:

a) nome-índice - Representa o nome da estrutura de índice que será criada.

b) nome-tabela - Representa o nome da tabela que contem a coluna na qual será criada o índice de acesso.

c) nome-coluna - Representa o nome da coluna que será criada.

d) Opção ASC/DESC - Representa a criação do índice ordenada crescentemente (ASC) ou decrescentemente (DESC).

Drop Index

Excluir uma estrutura de índice de acesso para uma determinada coluna em uma tabela.

Sintaxe:

DROP INDEX <nome-índice>