Linguagens de Banco de Dados

SQL- DDL

Prof. Raimundo Claudio Vasconcelos

Linguagem de Definição de Dados SQL - DDL

SQL - DDL permite a especificação de:

- - esquema de cada tabela
- - domínio de valores associados a cada atributo.
- - regras de integridade.
- - conjunto de índices para cada tabela.
- - informações de segurança (permissões de acesso)
- - estruturas de armazenamento físico das tabelas no disco.

Linguagem de Definição de Dados SQL - DDL

- Criação de bases de dados:

```
Create database <nome_bd>
        [ ON [PRIMARY] arquivo1 {, arquivo2} ...]
        [ LOG ON arquivo3 {, arquivo4} ...]
```

- Pode-se usar especificações default ou definir dados dos arquivos.
- arquivo1, ... representam especificações de arquivos (nome lógico, nome físico, tamanho, ...)
- Primary especifica o arquivo que conterá as tabelas do sistema
- LOG ON arquivos de log.

Linguagem de Definição de Dados SQL - DDL

- Exemplo:

```
CREATE DATABASE db_exemplo;
```

  - Ou

```
CREATE DATABASE db_BDII
ON (NAME = db_BDII_dat
      FILENAME = 'C:\SQL2005\DATA\db_BDII.mdf',
     SIZE = 10,
      MAXSIZE = 100,
      FILEGROWTH = 5)
 LOG ON (NAME = db_BDII_log
      FILENAME = 'C:\SQL2005\DATA\db_BDII.ldf',
     SIZE = 40,
      MAXSIZE = 100,
      FILEGROWTH = 10)
```

OBS: unidades default = MB, pode-se usar os sufixos TB, MB e KB.

Linguagem de Definição de Dados SQL - DDL

- Para mudar a base de dados em uso digite:
  use nome_bd

- Criação de tabelas:

Create table <nome_tabela>
(A1 D1 <[regra de integridade>],
A2 D2,
.
.
.
[<regra de integridade>]);

Linguagem de Definição de Dados SQL - DDL

- **tipos de dados numéricos:**
- int – representa valores inteiros, os quais podem ser armazenados em 4 bytes. int é a forma abreviada para integer.
- smallint – valores inteiros armazenados em 2 bytes
- tinyint – valores inteiros não negativos armazenados em 1 byte (0 a 255).
- bigint – valores inteiros armazenados em 8 bytes
- decimal(p,[s]) – valores com ponto flutuante. Tamanho total de dígitos é p e s é a quantidade de casas decimais
- numeric(p,[s]) – o mesmo que decimal
- real – aproximadamente de 2,23E -308 a 1,79E 308
- float[(p)] – semelhante a real. Se p<25 precisão simples (4 bytes), caso contrário, precisão dupla (8 bytes).
- money – valores monetários armazenados em 8 bytes
- smallmoney – 4 bytes

Linguagem de Definição de Dados SQL - DDL

- **tipos de dados string:**

- char[(n)] – conjunto de n caracteres, onde o valor máximo de n é 8000. Se n for omitido, o comprimento será de 1.
- varchar[(n)] - conjunto de n caracteres (tamanho variável). Armazenado o tamanho real da string.
- nchar[(n)] – o mesmo que char, apenas que cada caracter é armazenado em 2 bytes. Total de caracteres máximo é 4000.
- nvarchar[(n)] – mesmo que varchar com 2 bytes para cada caracter.
- text[(n)] – define uma string de comprimento fixo de até 2GB.
- ntext[(n)] – strings de comprimento variável onde cada caracter possui 2 bytes.

Linguagem de Definição de Dados SQL - DDL

- **tipos de dados binários:**
- binary[(n)] – especifica uma string de bits de comprimento fixo com exatamente n bytes. (até 8000)
- varbinary[(n)] – string de comprimento variável com até n bytes.
- image[(n)] – especifica uma string de comprimento fixo com valores praticamente ilimitados (2^{31} bytes).
- bit – especifica valores booleanos (false, true, NULL).

- **tipos para data e hora:**
- datetime – especifica uma data e hora com cada valor armazenado como um inteiro de 4 bytes.
- smalldatetime - especifica uma data e hora com cada valor armazenado como um inteiro de 2 bytes.

Linguagem de Definição de Dados SQL - DDL

- Tipos de regras de integridade:
  - - not null: especificadas junto com a definição do atributo, torna o campo obrigatório.
  - - primary key (A1,Am,Ak) : usada para definir a chave primária. Se a chave primária for formada apenas por um campo, ela pode ser definida na frente do campo, caso contrário deverá ser definida após a definição de todos os campos especificados.
  - - unique (A1,Am,Ak) : usada para definir a chaves candidatas (conjunto de atributos (um ou mais) que não se repetem na tabela)
  - - check (condição)-> permite a verificação de valores válidos.
  - - foreign key (Ai) references tabela (atributo): especificação de chave estrangeira. Deve ser fornecido o nome do(s) atributo(s) que compõe(m) a chave estrangeira e a tabela onde ele(s) são chave primária com os respectivo(s) nome(s) nesta tabela.

Linguagem de Definição de Dados SQL - DDL

- **Atribuição de nomes às regras de integridade através da clausula constraint:**
- **Exemplo:**
  constraint pk_exemplo primary key (A1, A2, …, An),

- Eliminação de tabelas:

- drop table <nome_tabela>;

Linguagem de Definição de Dados SQL - DDL

- Alteração da estrutura das tabelas:

- Alter table <nome_tabela> add (Ai Di) ou
  - alter column (Ai Di) ou
  - drop column A;
  - onde: - add - permite adicionar novas colunas a uma tabela. Se há dados cadastrados na tabela, a nova coluna será preenchida com null, portanto, neste caso, o atributo não pode ser definido como not null.
  - - drop column - permite eliminar colunas,
  - - alter column - permite alterar domínio de atributos existentes:
  - - aumentar o tamanho de coluna de caracteres ou números;
  - - aumentar ou diminuir o número de casas decimais;
  - - diminuir o tamanho de colunas de caracteres ou números apenas se a coluna estiver vazia.

Linguagem de Definição de Dados SQL - DDL

- Exemplo: criação das seguintes tabelas

  Agencias (**cod_ag**, nome, cidade);

  Clientes (**cod_cli**, nome, endereco, cidade);

  Contas (**cod_ag**, **n_conta**, cod_cli, saldo);

  Emprestimos (**cod_ag**, **n_empr**, cod_cli, valor);

Linguagem de Definição de Dados SQL - DDL

```
Create table  agencias
  (cod_ag   numeric (4) not null primary key,
        nome varchar (50) not null unique,
        cidade varchar (50)
        );

Create table  clientes
  (cod_cli  numeric (7) not null primary key,
        nome varchar (50) not null,
        endereco varchar (100),
        cidade varchar (50)
        );
```

Linguagem de Definição de Dados SQL - DDL

```
Create table  contas
   (       cod_ag   numeric (4) not null,
           n_conta numeric (7) not null,
           cod_cli  numeric (7) not null,
           saldo money,
           constraint ck_contas_saldo check (saldo >=0),
           constraint pk_contas primary key (cod_ag, n_conta),
           constraint fk_contas_cod_ag foreign key (cod_ag)
references agencias (cod_ag),
           constraint fk_contas_cod_cli foreign key (cod_cli)
references clientes (cod_cli)
   );
```

Linguagem de Definição de Dados SQL - DDL

```
Create table  emprestimos
   (        cod_ag   numeric (4) not null,
            n_empr numeric (7) not null,
            cod_cli  numeric (7) not null,
            valor money,
            constraint ck_empr_valor check (valor >=0),
            constraint pk_empr primary key (cod_ag, n_empr),
            constraint fk_empr_cod_ag foreign key (cod_ag)
references agencias (cod_ag),
            constraint fk_empr_cod_cli foreign key (cod_cli)
references clientes (cod_cli)
   );
```