

Introdução ao SQL/Criando Tabelas

Podemos criar uma tabela qualquer dentro de um banco de dados. A sintaxe básica para criarmos é:

```
CREATE TABLE nome_tabela
( nome_campo_1
tipo_1, nome_campo_2
tipo_2,
... nome_campo_n
tipo_n,
PRIMARY KEY ( campo_x,...));
```

CREATE TABLE é o comando para criação da tabela e deve ser seguida pelo nome que daremos à tabela. Dentro do comando, devemos definir os nomes dos campos de acordo com a conveniência do banco de dados, e determinar o tipo de dado que poderá ser incluído neste campo. Na seção abaixo Tipo de Dados estão especificados os tipos mais comuns encontrados nos SGBDs. PRIMARY KEY define a chave primária da tabela, isto é, o campo que serve como chave da tabela e que não pode ser repetido.

Se desejamos que um campo seja de preenchimento obrigatório, devemos inserir NOT NULL na frente do campo determinado.

```
CREATE TABLE nome_tabela
( nome_campo_1 tipo_1 NOT
NULL, nome_campo_2 tipo_2,
... nome_campo_n tipo_n,
PRIMARY KEY(campo_x,...));
```

Se desejamos que um campo seja de auto-incremento, devemos inserir AUTO_INCREMENT na frente do campo determinado. Isto pode ser utilizado por exemplo, para automatizar um código que sirva de chave primária de uma tabela:

```
CREATE TABLE nome_tabela
( nome_campo_1 tipo_1 NOT NULL
AUTO_INCREMENT, nome_campo_2 tipo_2,
...
nome_campo_n tipo_n,
PRIMARY KEY (campo_x,...));
```

Tipos de Dados

Tipos de dados definem os tipos de informação que podem ser inseridos em um campo. Somente dados do mesmo tipo do campo poderão ser inseridos. Os tipos suportados por um banco de dados podem variar de SGBD para SGBD, mas descrevemos a seguir os principais tipos encontrados na maior parte dos gerenciadores:

Tipo	Abreviação genérica	Descrição
INTEIRO	INT ou INTEGER	Números inteiros. Existem outros tipos de números inteiros de acordo com o gerenciador utilizado.
REAL	Decimal(10,2) Dez casas a esquerda e 2 casas a direita	Números reais. Permitem armazenar todo tipo de número não inteiro. Existem outros tipos de números reais de acordo com o gerenciador utilizado.
CARACTER	CHAR (NÚMERO)	Caracter. Permite a inclusão de dados alfanuméricos com tamanho pré-definido. O número de caracteres é definido entre os parênteses.
CARACTER VARIÁVEL	VARCHAR (NÚMERO)	Caracter. Permite a inclusão de dados alfanuméricos. O número de caracteres é definido entre os parênteses, porém o número de bytes utilizados pelo campo é de acordo com o dado inserido.
DATA	DATE	Data. Existem diversos tipos de datas de acordo com o gerenciador utilizado.
BOOLEANO	BOOLEAN	Armazena um bit de informação, utilizado para verdadeiro ou falso.

Exemplo

Como exemplo do uso do comando CREATE TABLE, imaginemos a necessidade de uma tabela que deva possuir os dados dos clientes de uma loja.

```
CREATE TABLE Cliente
(
Codigo INT NOT NULL AUTO_INCREMENT,
Nome VARCHAR (30) NOT NULL, Endereco
VARCHAR (40),
PRIMARY KEY (Codigo) );
```

Neste comando, criaremos uma tabela chamada Cliente. Esta tabela contém três campos: o primeiro campo é o Código do cliente. Este campo será utilizado como chave primária de forma que não poderá se repetir nunca. Desta forma o campo deve ser sempre preenchido (NOT NULL), é numérico do tipo inteiro (INT) e deve autoincrementar de acordo com o número de clientes que for incluído. O campo Nome é do tipo VARCHAR (30), ou seja, aceita dados alfa-numéricos com até 30 caracteres. No entanto se um nome

for inserido com menos de 30 caracteres, o número de bytes consumidos pelo campo será de acordo com o nome inserido. O campo Endereco é do mesmo modo que o campo nome. A instrução PRIMARY KEY define qual dos campos será a chave primária e não pode ser repetido, sendo o diferenciador entre os diversos clientes que sejam inseridos nesta tabela.

Describe

Após criarmos uma tabela, elas podem ser manipuladas de diversas formas, inserindo, alterando ou excluindo dados. Podemos visualizar a estrutura e campos de uma tabela criada utilizando o comando DESCRIBE ou DESC:

```
DESCRIBE Nome_Tabela; Ou:
```

```
DESC Nome_Tabela;
```
