



Aula 10

Prof: Henrique Augusto Maltauro

Implementar Banco de Dados Para WEB

SQL

- Tipos de Dados: Numéricos

Os tipos de dados numéricos no SQL Server são:

- bit
- tinyint
- smallint
- int
- bigint
- numeric

SQL

- Tipos de Dados: Numéricos (bit)

O **bit** é um tipo de dado de **número inteiro** que pode armazenar **0 ou 1**.

SQL

- Tipos de Dados: Numéricos (tinyint)

O **tinyint** é um tipo de dado de **número inteiro** que pode armazenar números do **0 ao 255**.

SQL

- Tipos de Dados: Numéricos (smallint)

O **smallint** é um tipo de dado de **número inteiro** que pode armazenar números do **-32.768** ao **32.767**.

SQL

- Tipos de Dados: Numéricos (int)

O **int** é um tipo de dado de **número inteiro** que pode armazenar números do **-2.147.483.648** ao **2.147.483.647**.

SQL

- Tipos de Dados: Numéricos (bigint)

O **bigint** é um tipo de dado de **número inteiro** que pode armazenar números do **-9.223.372.036.854.775.808** a **9.223.372.036.854.775.807**.

SQL

- Tipos de Dados: Numéricos (numeric)

O **numeric** é um tipo de dado de **número real** que pode armazenar números do $-10^{38} + 1$ ao $10^{38} - 1$. Ele possui dois **parâmetros**, a **precisão** e a **escala**.

O **parâmetro da precisão**, vai determinar a quantidade máxima de números.

SQL

- Tipos de Dados: Numéricos (numeric)

Enquanto o **parâmetro da escala**, vai determinar o quanto dessa quantidade máxima de números, estarão depois da vírgula.

Assim, a definição **numeric(5, 2)** vai permitir um número da seguinte forma: **000,00**.

SQL

- Tipos de Dados: Data e Hora

Os tipos de dados de data e hora no SQL Server são:

- date
- smalldatetime
- time
- datetimeoffset
- datetime
- datetime2

SQL

- Tipos de Dados: Data e Hora (date)

O **date** é um tipo de dado de data e hora, que armazena **apenas a data** com valores de **01/01/0001** até **31/12/9999**.

SQL

- Tipos de Dados: Data e Hora (time)

O **time** é um tipo de dado de data e hora, que armazena **apenas a hora**, com valores de **00:00:00** até **23:59:59.99999999**, com precisão de **100 nanosegundos**.

SQL

- Tipos de Dados: Data e Hora (datetime)

O **datetime** é um tipo de dado de **data e hora**, com valores de **01/12/1753 00:00:00 até 31/12/9999 23:59:59.997**, com precisão de **3,33 milisegundos**.

SQL

- Tipos de Dados: Data e Hora (datetime2)

O **datetime2** é um tipo de dado de **data e hora**, com valores de **01/01/0001 00:00:00** até **31/12/9999 23:59:59.9999999**, com precisão de **100 nanosegundos**.

SQL

- Tipos de Dados: Data e Hora (datetimeoffset)

O **datetimeoffset** é um tipo de dado de **data e hora**, que armazena valores iguais ao **datetime2**, porém com a adição de um **fuso horário**.

SQL

Conforme já vimos nas aulas passadas, nós precisamos usar o **DDL** para manipular a estrutura do nosso banco de dados.

E o **DDL** possui alguns **comandos** que utilizamos para manipular essa nossa estrutura, entre os quais, nesse primeiro momento, vamos nos atentar em três deles.

SQL

Esses **comandos** são:

→ **CREATE**

→ **ALTER**

→ **DROP**

SQL

- DQL: CREATE

O comando CREATE é responsável por criar os elementos da estrutura do nosso banco de dados.

SQL

- DQL: ALTER

O comando ALTER é responsável por alterar os elementos da estrutura do nosso banco de dados.

SQL

- DQL: DROP

O comando DROP é responsável por remover os elementos da estrutura do nosso banco de dados.

SQL

Todos esses três **comandos**, possuem uma série de **subcomandos**, que nos permite manipular diversos elementos dentro do nosso **banco de dados**.

Por hora, vamos nos atentar apenas na manipulação de tabelas com o **subcomando TABLE**, e conforme formos avançando vamos vendo outros **subcomandos**.

SQL

- DQL: CREATE (TABLE)

O comando **CREATE** em conjunto com o subcomando **TABLE** vai criar uma tabela na estrutura do **banco de dados**.

Logo após o **CREATE TABLE** é necessário informar o nome da tabela, o qual **não pode conter espaços e nem caracteres especiais**.

SQL

- **DQL: CREATE (TABLE)**

Depois de informado o nome da tabela, é informado as definições da tabela, bem como os seus campos, configurações de **PK** e **FK**, e etc.

Todas essas definições devem estar dentro de **um único parênteses**, e **separadas por vírgulas**.

SQL

- **DQL: CREATE (TABLE)**

Para criar as definições dos campos da tabela, é necessário informar primeiro o nome do campo, o qual não pode conter espaços nem caracteres especiais, seguido do tipo de dado que aquele campo corresponde.

SQL

- **DQL: CREATE (TABLE)**

Em seguida na definição do campo, vem a definição de nulidade do campo.

Para campos que aceitam valores nulos, se utiliza **NULL**, e para campos que não aceitam valores nulos, se utilizam **NOT NULL**.

SQL

- DQL: CREATE (TABLE)

```
CREATE TABLE PESSOAS (  
    ID INT NOT NULL,  
    NOME VARCHAR(60) NOT NULL  
)
```

SQL

- DQL: CREATE (TABLE)

Para definirmos uma **PK** em um ou mais campos da tabela, basta que no lugar da configuração de nulidade, seja usado o subcomando **PRIMARY KEY**.

SQL

- DQL: CREATE (TABLE)

```
CREATE TABLE PESSOAS (  
    ID INT PRIMARY KEY,  
    NOME VARCHAR(60) NOT NULL  
)
```

Exercício

Exercício

gg.gg/SenacBD10

github.com/hmaltaurodev/slides