



Artigo

SQL Server - Comandos Básicos, Objetos, Tipos de Dados e Criação de Database

Neste artigo do mini-curso de SQL Server, iremos ver os comandos básicos, descrever os objetos do SQL Server, mostrar a definição e criação de um Database e mostrarmos os tipos de dados

Marcar como lido

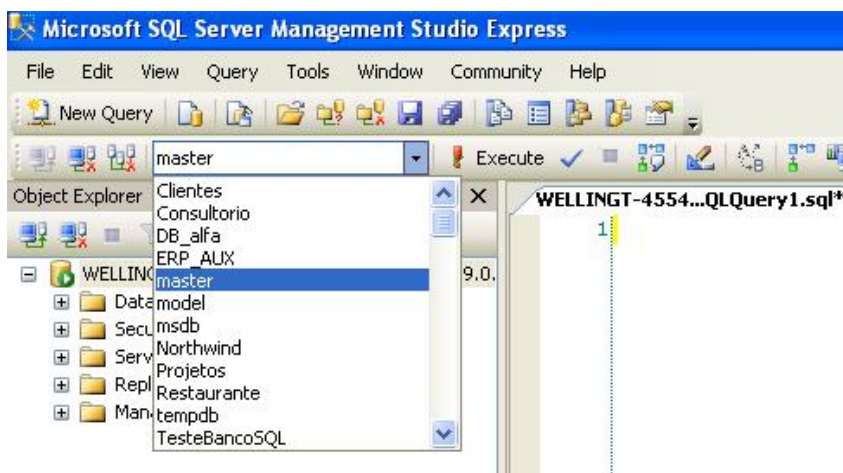


Anotar

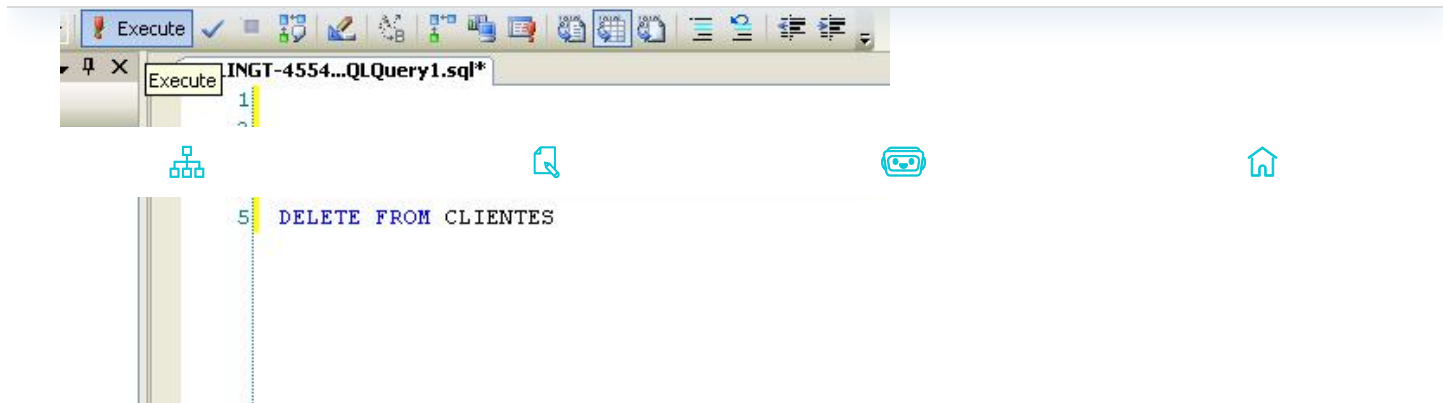


Olá pessoal, voltamos com nosso curso básico de SQL Server. Nesta parte, iremos ver os comandos básicos, descrever os objetos do SQL Server, mostrar a definição e criação de um Database e mostrarmos os tipos de dados. Acompanhem:

Comandos Básicos – Após ter aberto o SQL, clique em **New Query** abaixo do menu superior ou aperte **CTRL+N** para criar uma nova consulta. Abaixo desse botão New Query existe a barra de ferramentas SQL Editor, que possui uma dropdownlist chamada **Available Databases**, por meio da qual podemos escolher o banco de dados que queremos acessar, como mostra a imagem:



Para executar qualquer comando no SQL Server, escreva o determinado comando na tela de consultas do centro da tela e logo acima, na barra de ferramentas, clique no botão **Execute** ou simplesmente aperte **F5**. Já, se você tiver mais do que um comando, como por exemplo a imagem abaixo nos mostra uma instrução **SELECT** e abaixo dela uma **DELETE**, é necessário que selecionemos qual instrução queremos executar



Abaixo podemos notar a janela de resultados quando executamos a instrução acima.

 A screenshot of the 'Results' window in SQL Server. It displays a table with two columns: 'IDCLIENTE' and 'NOMECLIENTE'. There are 5 rows of data. The first row is highlighted.

	IDCLIENTE	NOMECLIENTE
1	6	Wellington Balbo
2	7	Charles Pires
3	8	Teste
4	9	Osmar Balbo
5	10	Testando

Podemos usar as teclas **CTRL+R** para ocultar esse resultado se desejarmos.

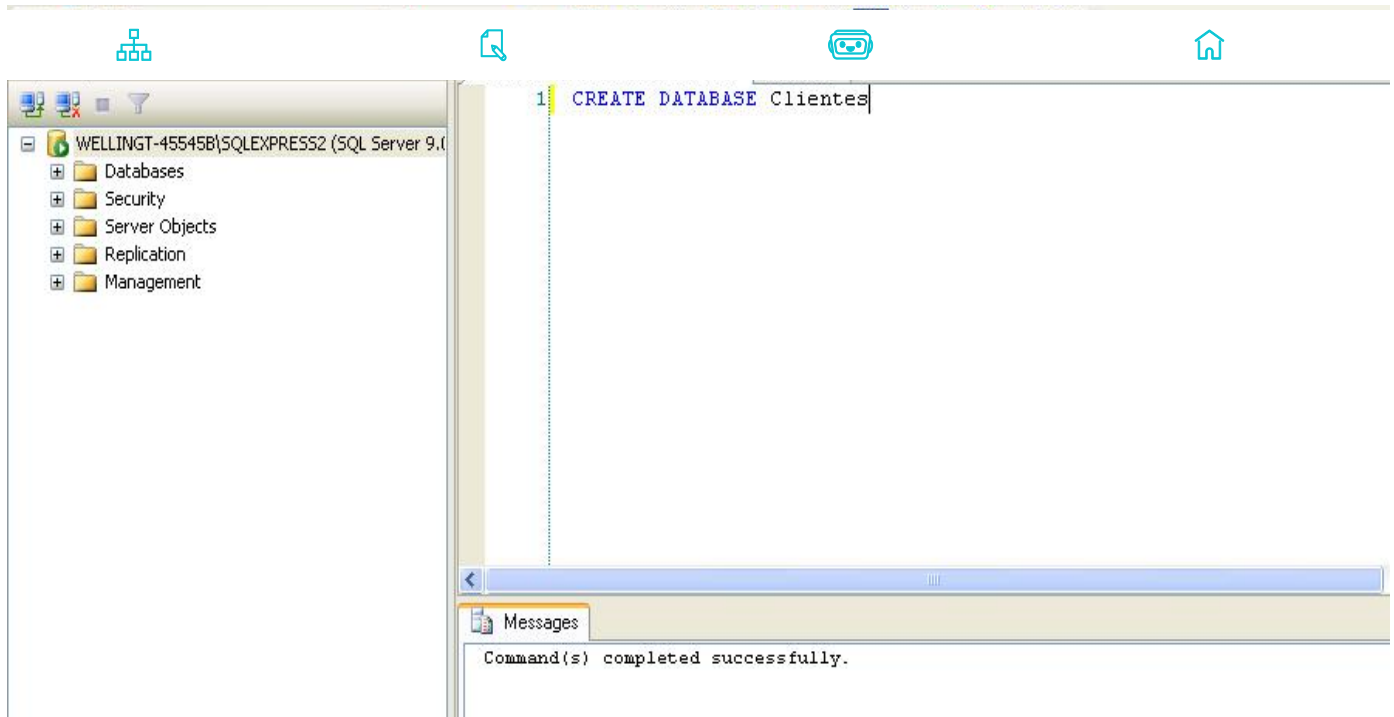
Objetos do SQL Server – A versão 2005 do SQL Server nos oferece diferentes tipos de objetos, como descrevemos abaixo:

- **Database** – Os objetos do sistema são criados dentro de uma estrutura lógica que corresponde ao objeto Database, iremos falar dele mais adiante.
- **Table** – Os dados do sistema são incluídos neste objeto de duas dimensões, que é formado por linhas e colunas.
- **Constraint, Default e Rule** – Consistem em regras usadas para implementar a consistência e a integridade dos dados. Mais sobre elas em artigos futuros.
- **Data Type e User Defined Data Type** – Os dados são armazenados no disco sob um formato representado pelo datatype. Um datatype deverá ser atribuído a cada coluna de uma tabela.
- **View** – Este objeto nos oferece uma visualização lógica dos dados de uma tabela, de modo que diversas aplicações possam compartilhá-la.
- **Index** – São objetos responsáveis pela otimização de acesso aos dados de uma tabela, com o objetivo de agilizar determinadas pesquisas de dados.
- **Procedure** – Neste objeto, encontramos um bloco de comandos **Transact-SQL**, responsável por uma determinada tarefa. Sua lógica pode ser compartilhada por várias aplicações. Muito utilizado atualmente em praticamente todo tipo de empresa.
- **Trigger** – Como a Procedure, este objeto também possui um bloco de comandos Transact-SQL. É criado sobre uma tabela e ativado no momento da execução dos comandos **UPDATE**, **INSERT** e/ou **DELETE**.
- **Function** – Neste objeto, temos um bloco de comandos Transact-SQL responsável por uma determinada tarefa. Como a Procedure, sua lógica pode ser compartilhada por N aplicações. Importante dizer que uma função SEMPRE retornará um valor.

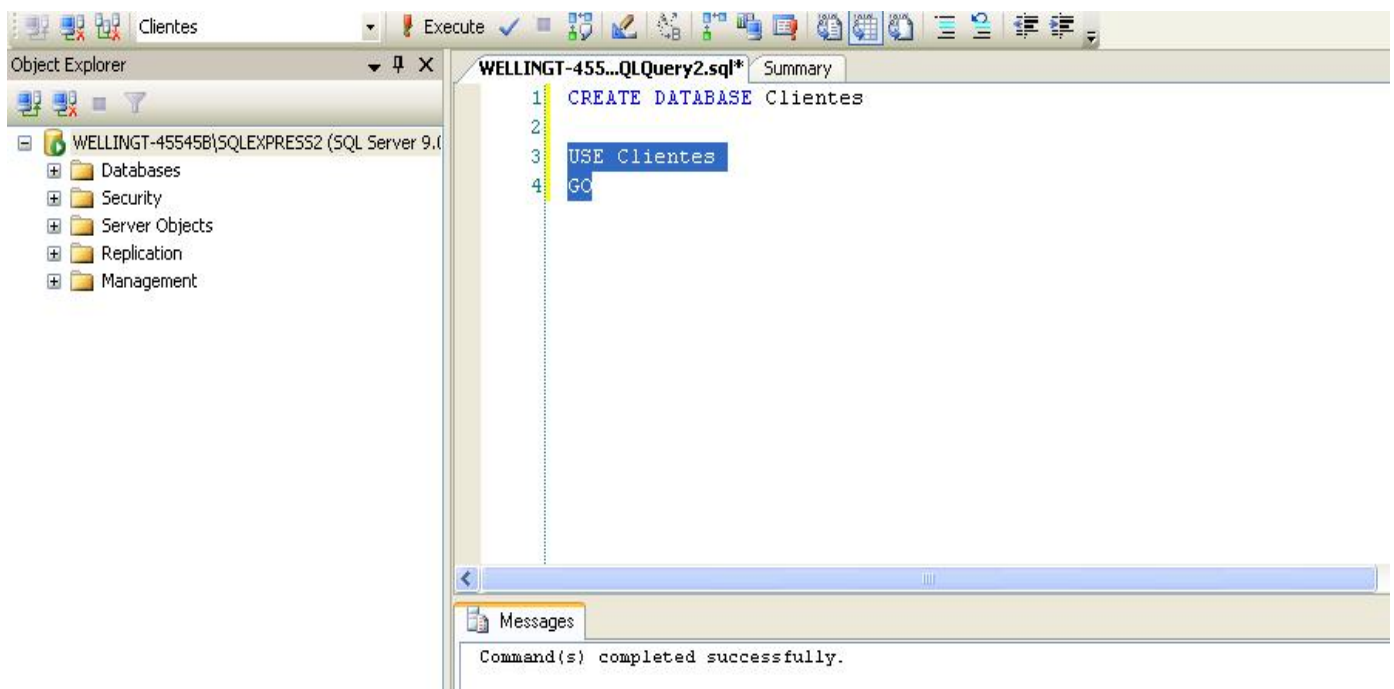
Obs: Os objetos **Procedure**, **Trigger** e **Function** são processados rapidamente, pois seu código tende a ficar compilado na memória. Isso acontece porque estes objetos são executados no servidor de dados.

Databases – Como dito anteriormente, dentro do objeto Database são criados os objetos que fazem parte

Por exemplo, vamos dizer que quero criar um database chamado Clientes. Para isso, abro o SQL Server, crio uma nova query, digito a instrução **CREATE DATABASE** Clientes e aperto **F5** ou clico em **Execute**. Veja abaixo:

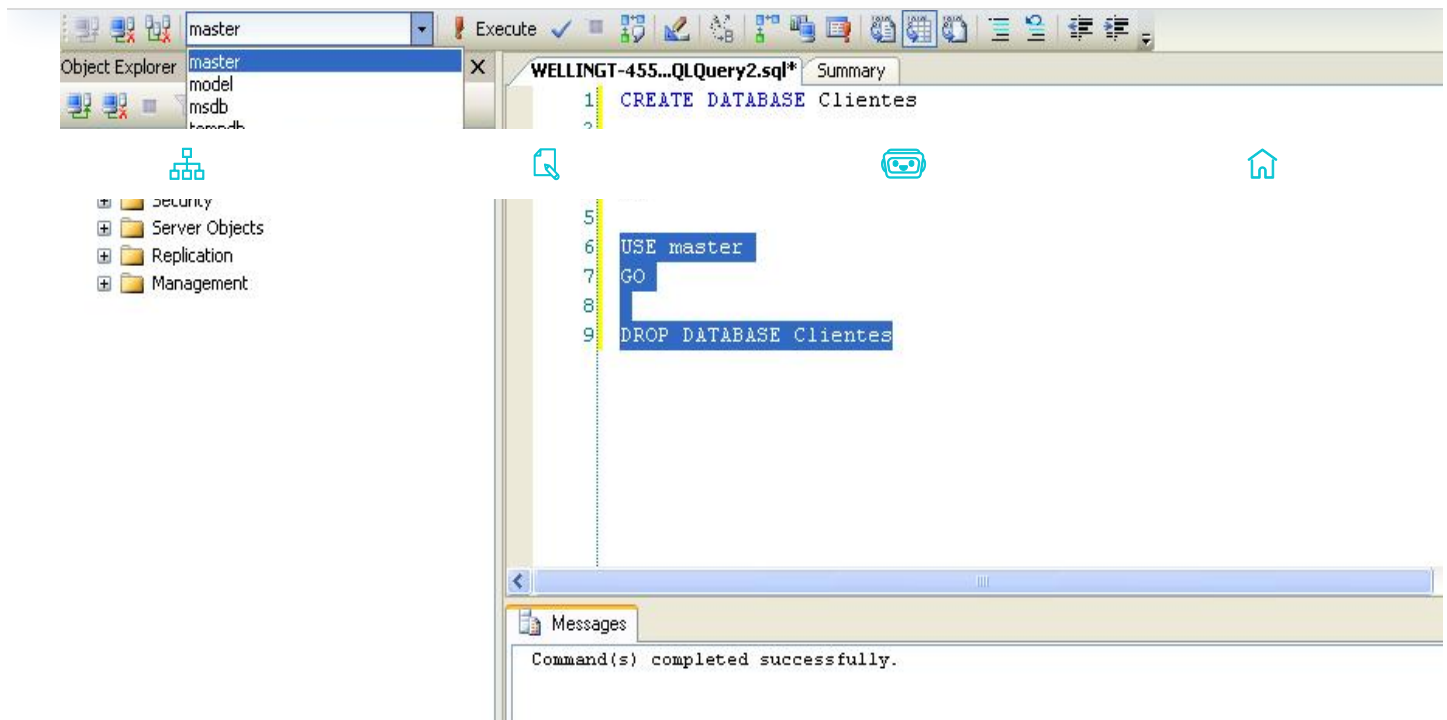


Como há de se esperar, assim que criamos um database, não há tabelas no mesmo, muito menos registros. Mais, antes que pensemos em inserir tabelas (e registros), precisamos informar ao SQL Server qual database iremos utilizar. Para isso, usamos a instrução **USE nomedodatabase**. Confira abaixo:



Logo abaixo da minha instrução usei a palavra **GO**, que nada mais é que uma simples maneira de dizer ao SQL para seguir na respectiva instrução. Esse comando não é obrigatório.

Da mesma forma, se desejarmos excluir o database recém criado, devemos usar a instrução **DROP DATABASE**. Apenas ressaltando que, se estivermos dentro do database que iremos excluir, o SQL nos impede de fazermos isso. Para resolver isso, simplesmente saia do database e vá para o master por exemplo, como mostra a imagem a seguir:



Podemos notar que o database foi mesmo excluído, pois vemos acima no DropDownList **Available Databases**.

Tipos de Dados – Os tipos de dados são classificados em diferentes categorias e permitem N formatos. Abaixo uma descrição de cada categoria e de cada tipo de dado do SQL Server 2005:

- Baseados em Caracteres:

- **Char(n)** – Trata-se de um datatype que aceita como valor qualquer dígito, sendo que o espaço ocupado no disco é de um dígito por caractere. É possível utilizar até 8 mil dígitos.
- **Varchar(n)** – Também aceita como valor qualquer dígito e o espaço ocupado em disco é de um dígito por caractere. Permite usar também no máximo 8 mil dígitos. A diferença pro Char, é que o Varchar geralmente é usado quando não sei o tamanho fixo de um campo.
- **Text** – Qualquer dígito pode ser usado neste datatype, sendo ocupado 1 byte por caractere, o equivalente a **2.147.483.647** bytes.

- Baseados em Caracteres Unicode:

- **Nchar(n)** – Neste datatype, pode usar qualquer dígito, sendo ocupados 2 bytes a cada caractere. É possível usar até 8 mil bytes.
- **Nvarchar(n)** – Igual ao tipo anterior, com a única diferença que uso esse tipo quando não sei o tamanho fixo de um campo. 2 bytes são ocupados a cada caractere. É possível usar até **8 mil** bytes.
- **Ntext** – Também aceita qualquer dígito, 2 bytes são ocupados a cada caractere. Podem ser usados até **1.073.741.823** bytes.

- Baseados em Numéricos Inteiros:

- **Bigint** – Aceita valores entre -2^{63} e $2^{63}-1$, sendo que esse datatype ocupa 8 bytes.
- **Int** – Os valores aceitos aqui variam entre -2^{31} a $2^{31}-1$. Ocupa 4 bytes.
- **Smallint** – Aceita valores entre -32768 até 32767 e ocupa 2 bytes.
- **Tinyint** – Os valores aceitos aqui variam entre 0 e 255, ocupa apenas 1 byte.

Bit – É um tipo de dado inteiro (conhecido também como booleano), cujo valor pode corresponder a

- Baseados em Numéricos Exatos:



precisão e de 10 a 19, o espaço ocupado é de 9 bytes, já se a precisão for de 20 a 28, o espaço ocupado é de 13 bytes, e se a precisão for de 29 a 38, o espaço ocupado é de 17 bytes.

- **Numérico(P,S)** – Considerado um sinônimo do datatype decimal, o numérico também permite valores entre -10^{38-1} e 10^{38-1} e o espaço ocupado é o mesmo do anterior.

- Baseados em Numéricos Aproximados:

- **Float[(n)]** – O mesmo que **double precision** quando o valor de **n** é 53, este datatype aceita valores entre $-1.79E + 308$ e $1.79E + 308$. O espaço ocupado varia de acordo com o valor de **n**. Se esse valor estiver entre 1 e 24, a precisão será de 7 dígitos, sendo que o espaço ocupado será de 4 bytes. Se o valor de **n** estiver entre 25 e 53, sua precisão será de 15 dígitos, assim sendo o espaço ocupado será de 8 bytes.

- **Real** – Este datatype é similar ao **float(n)** quando o valor de **n** é 24. Os valores aceitos variam entre $-3.40E + 38$ e $3.40E + 38$. Esse datatype ocupa 4 bytes.

- Baseados em Valores Numéricos Monetários:

- **Money** – Este datatype aceita valores entre -2^{63} e $2^{63}-1$, sendo que 8 bytes são ocupados.

- **Smallmoney** – É possível usar valores entre -2^{31} e $2^{31}-1$, sendo que 4 bytes são ocupados.

- Baseados em Data e Hora:

- **Datetime** – Permite o uso de valores entre **1/1/1753** e **31/12/9999**. Este datatype ocupa 8 bytes e sua precisão atinge **3.33** milissegundos.

- **Smalldatetime** – Aceita o uso de valores entre 1/1/1900 e 06/06/2079, sendo que sua precisão é de 1 minuto e ocupa 4 bytes em disco.

- Baseados em Binários:

- **Binary[(n)]** – Este datatype representa os dados que serão usados no formato binário. O espaço ocupado é de **n+4** bytes, sendo que **n** pode variar entre 1 e 8000 bytes.

- **Varbinary[(n)]** – Aqui também é usado o formato binário, o espaço ocupado e a variação de **n** é igual ao anterior.

- **Image** – O formato binário também é usado aqui, sendo que o espaço ocupado é de $2^{31}-1$ bytes ou **2.147.483.647**.

- Baseados em Tipos de Dados Especiais:

- **Uniqueidentifier** – O formato hexadecimal é usado para o armazenamento de dados binários, sendo que este datatype ocupa 16 bytes.

- **Timestamp** – Um valor binário é gerado pelo SQL Server, sendo que esse datatype ocupa 8 bytes.

- **Bit** – Este datatype pode apresentar **0**, **1** ou **NULL**, como valor, sendo ocupado 1 byte. Também utilizado como um tipo de dado **int**.

- **Sql_Variant** – Os valores aqui podem ser de qualquer datatype, sendo que é possível armazenar até 8016 bytes.

- **Cursor** – Datatype usado somente quando trabalhamos com variáveis.



- **Xml** – Por este datatype, podemos armazenar fragmentos de documentos XML em um banco de dados SQL. Estes fragmentos correspondem à instâncias XML que não possuem um determinado elemento de nível superior. Essas instâncias são armazenadas quando criamos variáveis e colunas com datatype XML. O espaço máximo ocupado deve ser de 2GB.

Desta forma, terminamos aqui mais uma parte de nossa série de artigos, ou melhor dizendo, de nosso mini curso básico de SQL Server. Na próxima parte, iremos ver as tabelas do sistema, como crias tabelas e começar a falar dos comandos **DML**, como o **INSERT** e **SELECT**. Aguardem!

Quaisquer dúvidas, postem.

Abraços.

Até o próximo artigo!

Marcar como lido



Anotar



Por Wellington

Em 2010

RECEBA NOSSAS NOVIDADES

Informe o seu e-mail

Receber Newsletter

Suporte ao aluno - Deixe a sua dúvida.

DEV MEDIA





Plataforma para Programadores

Compre pelo WhatsApp:

 (21) 96759-5390

Revistas

Fale conosco

Trabalhe conosco

Assinatura para empresas



[Hospedagem web por Porta 80 Web Hosting.](#)

Av. Ayrton Senna 3000, Shopping Via Parque, grupo 3087 - Barra da Tijuca - Rio de Janeiro - RJ



6