



Aula 09

Prof: Henrique Augusto Maltauro

Implementar Banco de Dados Para WEB

SQL

SQL

O **SQL** (**Structured Query Language, Linguagem de Consulta Estruturada**) é uma linguagem padrão para manipulação de registros em **bancos de dados relacionais**.

É uma linguagem declarativa de script que apresenta sintaxe simples e fácil de ser aprendida até por pessoas iniciantes na área da programação.

SQL

O **SQL** surgiu a partir de um estudo realizado na década de 70 pelo laboratório da **IBM**. Esse estudo tinha como um de seus objetivos elaborar uma linguagem que fosse capaz de simplificar o gerenciamento de informações em **bancos de dados relacionais**.

Ali surgiu a especificação original do **SQL**, que foi feita pelos cientistas da computação, **Donald Chamberlim** e **Raymond Boyce**.

SQL

O sucesso da linguagem foi imediato e rapidamente diversos “dialetos” começaram a ser desenvolvidos, o que gerou a necessidade do **ANSI (American National Standards Institute)** padronizar o seu uso.

Assim, o **SQL** continua sendo usado em grande parte dos **bancos de dados relacionais** até os dias atuais. Mas, apesar da padronização da linguagem, ainda existem algumas variações utilizadas por diferentes **SGBDs**.

SQL

O **SQL Server**, utiliza o **T-SQL (Transact-SQL)** que é a extensão do **SQL** pertencente à **Microsoft**.

O **T-SQL** expande o padrão **SQL** para incluir programação processual, variáveis locais, várias funções de suporte para processamento de strings, processamento de datas, matemática, etc.

SQL

O **SQL** é usado para fazer qualquer tipo de manipulação dos registros de um **banco de dados**.

Ou seja, ele serve para criar, inserir, atualizar, excluir e consultar as informações armazenadas na base, além de outras diversas funções mais complexas.

SQL

- Vantagens

Padronização: como já dito, o **SQL** foi padronizado pelo **ANSI**. Por isso, ela oferece uma ampla documentação e, aprendendo a linguagem, você vai conseguir lidar facilmente com diferentes **bancos de dados** que a implementam.

Rápido Acesso aos Dados: o **SQL** permite a recuperação de múltiplos registros com um único comando, tudo feito de forma rápida e eficiente.

SQL

- Vantagens

Portabilidade: o SQL pode ser usado em diversas aplicações para diferentes plataformas.

Múltiplas Visões de Dados: com o SQL é possível definir diferentes visualizações da estrutura do banco para diferentes pessoas usuárias.

Linguagem Cliente/Servidor: com o SQL é possível implementar arquitetura cliente-servidor.

SQL

- Desvantagens

Dificuldade com a Interface: para quem não está acostumado, a interface de um **banco de dados SQL** pode ser mais difícil de lidar.

Custo: o custo operacional de alguns serviços oferecidos por bancos que utilizam **SQL** é caro e dificulta o acesso dos desenvolvedores.

Preços do SQL Server 2022¹

Edições	Aberta, sem restrições (USD\$)	Modelo de licença	Disponibilidade do canal
Grandes empresas	USD\$ 15.123 ^[2]	Pacote de 2 núcleos	Licenciamento por volume, hospedagem
Standard – por núcleo	USD\$ 3.945 ^[2]	Pacote de 2 núcleos	Licenciamento por volume, hospedagem
Standard – servidor	USD\$ 989 ^[2]	Server ^[3]	Licenciamento por volume, hospedagem
Standard – CAL	USD\$ 230	CAL	Licenciamento por volume, hospedagem
Desenvolvedor	Grátis	Por usuário	Baixe gratuitamente
Web	Consulte seu parceiro de hospedagem para saber o preço	Não aplicável	Somente hospedagem
Expresso	Grátis	Não aplicável	Baixe gratuitamente

SQL

- Subconjuntos

O **SQL** possui uma organização estrutural muito bem definida que divide seus comandos em 5 **subconjuntos** diferentes. Cada um desses **subconjuntos** apresenta objetivos específicos.

SQL

- Subconjuntos

Esses subconjuntos são:

→ DQL

→ DML

→ DDL

→ DCL

→ DTL

SQL

- Subconjuntos: DQL

O DQL (Data Query Language, Linguagem de Consulta de Dados) é o subconjunto do SQL que define os comandos mais populares da linguagem.

Esses comandos são essenciais para que possamos consultar os dados que armazenamos em nosso banco.

SQL

- Subconjuntos: DML

O DML (Data Manipulation Language, Linguagem de Manipulação de Dados) é o subconjunto do SQL que define os comandos usados para manipular os dados armazenados em um banco.

Esse é um dos conjuntos mais utilizados, pois ele fornece operadores que nos permitem inserir, excluir e alterar os registros de uma tabela, por exemplo.

SQL

- Subconjuntos: DDL

O DDL (Data Definition Language, Linguagem de Definição de Dados) é o subconjunto do SQL que apresenta comandos usados para gerenciar as estruturas do banco de dados.

Com ele, podemos criar, atualizar e remover objetos da base, como tabelas e índices.

SQL

- Subconjuntos: **DCL**

O **DCL** (Data Control Language, Linguagem de Controle de Dados) é o **subconjunto** do **SQL** dos comandos utilizados para controlar o acesso aos dados da base.

A um primeiro momento, eu não vou passar esses comandos, mas se der tempo eu passo eles.

SQL

- Subconjuntos: DTL

O **DTL** (Data Transaction Language, Linguagem de Transação de Dados) é o **subconjunto** do **SQL** que define comandos que utilizamos quando é necessário gerenciar transações feitas no banco.

Isso significa que eles permitem iniciar, confirmar e desfazer determinadas alterações.

Novamente, a um primeiro momento, eu não vou passar esses comandos, mas se der tempo eu passo eles.

SQL

Para iniciarmos então com o nosso **SQL**, precisamos criar o nosso **banco de dados**, e criar as nossas tabelas com os comandos de **DDL**.

Mas antes disso, precisamos saber quais os **tipos de dados** que o **SQL Server** utiliza, para que com isso nós possamos criar as tabelas.

SQL

Para isso, eu quero que vocês pesquisem os tipos de dados do **SQL Server**.

Cada um de vocês deve escolher 3 tipos de dados que o **SQL Server** utiliza e tentar entender ele.

TODOS, SEM EXCEÇÃO, irão falar o que entenderam dos tipos de dados escolhidos.

SQL

- Tipos de Dados

Os tipos de dados são utilizados para definir os tipos de dados dos campos de cada tabela do banco de dados.

SQL

- **Tipos de Dados**

Os tipos de dados do **SQL Server** podem ser divididos em:

→ **Tipos String**

→ **Tipos Numéricos**

→ **Tipos Data e Hora**

SQL

- Tipos de Dados: String

Os tipos de dados de **string** no **SQL Server** são:

- | | |
|------------|-------------|
| → char | → ntext |
| → varchar | → binary |
| → text | → varbinary |
| → nchar | → image |
| → nvarchar | |

SQL

- Tipos de Dados: String (char)

O **char** é uma **sequência de caracteres** de até **8.000 caracteres**, e de tamanho fixo, ou seja, ele sempre vai ocupar o espaço máximo de caracteres na memória.

Ele recebe um **parâmetro** que define o tamanho de bytes que ele ocupa dentro do **banco de dados**. Por coincidência, o número de bytes é o mesmo do tamanho de caracteres.

SQL

- Tipos de Dados: String (varchar)

O **varchar** é uma **sequência de caracteres** de até **8.000 caracteres**, e de tamanho variável, ou seja, ele vai ocupar na memória apenas a quantidade de caracteres que a informação recebeu.

Ele recebe um **parâmetro** que define o tamanho de bytes que ele ocupa dentro do **banco de dados**. Por coincidência, o número de bytes é o mesmo do tamanho de caracteres.

SQL

- Tipos de Dados: **String (varchar)**

O **varchar** também pode receber como parâmetro a **palavra-chave max**, fazendo que o campo da tabela armazene até **1.073.741.824** caracteres.

SQL

- Tipos de Dados: **String (text)**

O **text** é uma **sequência de caracteres** de até **2GB de caracteres**, e de tamanho variável, ou seja, ele vai ocupar na memória apenas a quantidade de caracteres que a informação recebeu.

SQL

- Tipos de Dados: String (nchar)

O **nchar** é uma **sequência de caracteres unicode** de até **4.000 caracteres**, e de tamanho fixo, ou seja, ele sempre vai ocupar o espaço máximo de caracteres na memória.

Ele recebe um **parâmetro** que define o tamanho de bytes que ele ocupa dentro do **banco de dados**. Por coincidência, o número de bytes é o mesmo do tamanho de caracteres.

SQL

- Tipos de Dados: String (nvarchar)

O **nvarchar** é uma **sequência de caracteres unicode** de até **4.000 caracteres**, e de tamanho variável, ou seja, ele vai ocupar na memória apenas a quantidade de caracteres que a informação recebeu.

Ele recebe um **parâmetro** que define o tamanho de bytes que ele ocupa dentro do **banco de dados**. Por coincidência, o número de bytes é o mesmo do tamanho de caracteres.

SQL

- Tipos de Dados: **String (nvarchar)**

O **nvarchar** também pode receber como **parâmetro** a **palavra-chave max**, fazendo que o campo da tabela armazene até **536.870.912** caracteres.

SQL

- Tipos de Dados: **String (ntext)**

O **ntext** é uma **sequência de caracteres unicode** de até **2GB de caracteres**, e de tamanho variável, ou seja, ele vai ocupar na memória apenas a quantidade de caracteres que a informação recebeu.

SQL

- Tipos de Dados: String (binary)

O **binary** é uma **sequência de binários** de até **8.000 binários**, e de tamanho fixo, ou seja, ele sempre vai ocupar o espaço máximo de binários na memória.

Ele recebe um **parâmetro** que define o tamanho de bytes que ele ocupa dentro do **banco de dados**. Por coincidência, o número de bytes é o mesmo do tamanho de binários.

SQL

- Tipos de Dados: String (varbinary)

O **varbinary** é uma **sequência de binários** de até **8.000 binários**, e de tamanho variável, ou seja, ele vai ocupar na memória apenas a quantidade de binários que a informação recebeu.

O varbinary também pode receber como **parâmetro** a **palavra-chave max**, fazendo que o campo da tabela armazene até **2GB de binários**.

SQL

- Tipos de Dados: **String (image)**

O **image** é uma **sequência de binários** de até **2GB de binários**, e de tamanho variável, ou seja, ele vai ocupar na memória apenas a quantidade de caracteres que a informação recebeu.