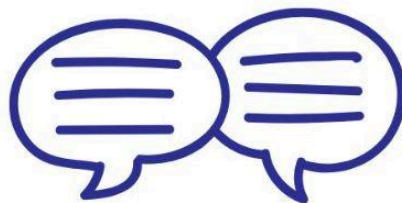




# FACULDADE IMPACTA

SQL - CRIANDO SISTEMAS DE BANCO DE DADOS

NANODEGREE



ALEX SOUSA

SÃO PAULO - 05/2024

## SUMÁRIO

SUMÁRIO.....	1
As linguagens SQL e T-SQL.....	2
SQL.....	2
T-SQL - Transact-SQL.....	2
SQL Server.....	3
Objetos de banco de dados.....	3
Tabelas.....	3
Índices.....	3
CONSTRAINT.....	3
VIEW (Visualização).....	4
PROCEDURE (Procedimento armazenado).....	4
FUNCTION (Função).....	4
TRIGGER (Gatilho).....	4
Usuário e senha padrão do SQL Server.....	4
<b>Ferramentas de gerenciamento.....</b>	<b>4</b>
<b>SSMS - SQL Server Management Studio.....</b>	<b>4</b>
SQL Server Configuration Manager.....	5
Microsoft SQL Server Profiler.....	6
Database Engine Tuning Advisor.....	6
SQL Server Data Tools.....	6
<b>Modelo Físico.....</b>	<b>6</b>
<b>Pontos principais.....</b>	<b>7</b>
<b>Pontos principais - I.....</b>	<b>7</b>
<b>Índices.....</b>	<b>8</b>
Tipos de índices.....	8
Índice clusterizado.....	8
Índice não clusterizado.....	8
Benefícios dos índices.....	8
Quando usar índices.....	8
Cuidados com os índices.....	9
Exemplos de criação de índices.....	9
<b>Pontos principais - II.....</b>	<b>9</b>
<b>Pontos principais II.....</b>	<b>9</b>
<b>Pontos principais III.....</b>	<b>10</b>
<b>Pontos principais IV.....</b>	<b>10</b>
<b>Pontos principais V.....</b>	<b>10</b>

## As linguagens SQL e T-SQL

Toda a manipulação de um banco de dados é feita por meio de uma linguagem específica, com uma exigência sintática rígida, chamada SQL (Structured Query Language). Os fundamentos dessa linguagem estão baseados no conceito de banco de dados relacional.

Essa linguagem foi desenvolvida pela IBM no início da década de 1970 e, posteriormente, foi adotada como linguagem padrão pela ANSI (American National Standards Institute) e pela ISO (International Organization for Standardization), em 1986 e 1987, respectivamente.

A T-SQL (Transact-SQL) é uma implementação da Microsoft para a SQL padrão ANSI. Ela cria opções adicionais para os comandos e também cria novos comandos que permitem o recurso de programação, como os de controle de fluxo, variáveis de memória etc.

Veja um exemplo de comando SQL:

Para realizar uma consulta na tabela de departamento (TB\_DEPARTAMENTO) é utilizado o comando SELECT. Esse comando permite a visualização das informações contidas na referida tabela.

Code:

```
SELECT * FROM TB_DEPARTAMENTO
```

## SQL

SQL - Structured Query Language (Linguagem de Consulta Estruturada), é uma linguagem de domínio específico usada para gerenciar dados em sistemas de gerenciamento de banco de dados relacional (SGBDR).

Imagine o SQL como uma espécie de tradutor que permite você se comunicar com esses bancos de dados.

Aqui estão alguns dos pontos-chave sobre SQL:

- **Consulta:** SQL é principalmente usada para consultar dados armazenados em tabelas. Você pode selecionar dados específicos, filtrar resultados e ordená-los de acordo com a sua necessidade.
- **Padronizada:** SQL é uma linguagem padronizada, o que significa que a maior parte do código funciona em diferentes plataformas de banco de dados com pouca variação.
- **Flexível:** Embora a consulta seja o foco principal, o SQL também permite manipular dados (inserir, atualizar e deletar), criar tabelas e definir regras de acesso aos dados.

Então, porque usamos SQL? Bom, bancos de dados relacionais armazenam informações de maneira organizada, mas sem uma linguagem específica para acessá-los, seria bem complicado extrair esses dados. O SQL facilita esse processo, tornando a comunicação com o banco de dados simples e eficiente.

## T-SQL - Transact-SQL

T-SQL, ou Transact-SQL, é uma extensão da linguagem SQL (Structured Query Language) criada pela Microsoft (e também pela Sybase). É basicamente a linguagem que você usa para se comunicar com bancos de dados Microsoft SQL Server.

Enquanto o SQL padrão foca principalmente na consulta de dados, o T-SQL oferece funcionalidades bem mais amplas. Pense nele como um SQL com "superpoderes". Aqui estão algumas das coisas que o T-SQL permite fazer além do que o SQL básico oferece:

- **Programação Procedural:** Você pode escrever programas T-SQL que incluem blocos de código, loops e condicionais, permitindo automatizar tarefas complexas no banco de dados.
- **Variáveis Locais:** Armazene valores temporários durante a execução de um script T-SQL, tornando seu código mais flexível.
- **Funções Extras:** T-SQL oferece um conjunto expandido de funções para manipular strings, datas, realizar cálculos e muito mais.
- **Alterações em comandos DELETE e UPDATE:** T-SQL modifica como os comandos para apagar e atualizar dados funcionam no banco de dados.

Resumindo, T-SQL é a ferramenta principal para interagir com o SQL Server. Se você precisa gerenciar um banco de dados SQL Server, vai precisar aprender T-SQL.

## SQL Server

O SQL Server é uma plataforma de banco de dados utilizada para armazenar dados e processá-los, tanto em um formato relacional quanto em documentos XML. Também é utilizada em aplicações de comércio eletrônico e atua como uma plataforma inteligente de negócios para integração e análise de dados, bem como de soluções.

Para essas tarefas, o SQL Server faz uso da linguagem T-SQL para gerenciar bancos de dados relacionais, que contém, além da SQL, comandos de linguagem procedural.

### Objetos de banco de dados

Os objetos que fazem parte de um sistema de banco de dados do SQL Server são criados dentro do objeto DATABASE, que é uma estrutura lógica formada por dois tipos de arquivo: um arquivo responsável por armazenar os dados e outro por armazenar as transações realizadas. Veja a seguir alguns objetos de banco de dados do SQL Server.

### Tabelas

Os dados são armazenados em objetos de duas dimensões denominados tabelas (tables), formadas por linhas e colunas. As tabelas contêm todos os dados de um banco de dados e são a principal forma para coleção de dados.

### Índices

Quando realizamos uma consulta de dados, o SQL Server 2016 faz uso dos índices (index) para buscar, de forma fácil e rápida, informações específicas em uma tabela ou VIEW indexada.

### CONSTRAINT

São objetos cuja finalidade é estabelecer regras de integridade e consistência nas colunas das tabelas de um banco de dados. São cinco os tipos de CONSTRAINT oferecidos pelo SQL Server: PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK e DEFAULT.

## **VIEW (Visualização)**

Definimos uma VIEW (visualização) como uma tabela virtual composta por linhas e colunas de dados, os quais são provenientes de tabelas referenciadas em uma consulta que define essa tabela.

Esse objeto oferece uma visualização lógica dos dados de uma tabela, de modo que diversas aplicações possam compartilhá-la.

Essas linhas e colunas são geradas de forma dinâmica no momento em que é feita uma referência a uma VIEW.

## **PROCEDURE (Procedimento armazenado)**

Nesse objeto, encontramos um bloco de comandos T-SQL, responsável por uma determinada tarefa. Sua lógica pode ser compartilhada por diversas aplicações. A execução de uma procedure é realizada no servidor de dados. Por isso, seu processamento ocorre de forma rápida, visto que seu código tende a ficar compilado na memória.

## **FUNCTION (Função)**

Nesse objeto, encontramos um bloco de comandos T-SQL responsável por uma determinada tarefa, isto é, a função (FUNCTION) executa um procedimento e retorna um valor. Sua lógica pode ser compartilhada por diversas aplicações.

## **TRIGGER (Gatilho)**

Esse objeto também possui um bloco de comandos T-SQL. O TRIGGER é criado sobre uma tabela e ativado automaticamente no momento da execução dos comandos UPDATE, INSERT ou DELETE.

Quando atualizamos, inserimos ou excluímos dados em uma tabela, o TRIGGER automaticamente grava em uma tabela temporária os dados do registro atualizado, inserido ou excluído.

## **Usuário e senha padrão do SQL Server**

Embora o SQL Server possua um usuário e senha padrão, é altamente desaconselhável utilizá-los em um ambiente de produção por motivos de segurança:

- Conta "sa" com privilégios de administrador.
- A senha padrão para a conta "sa" é " " (duas aspas simples).

## **Ferramentas de gerenciamento**

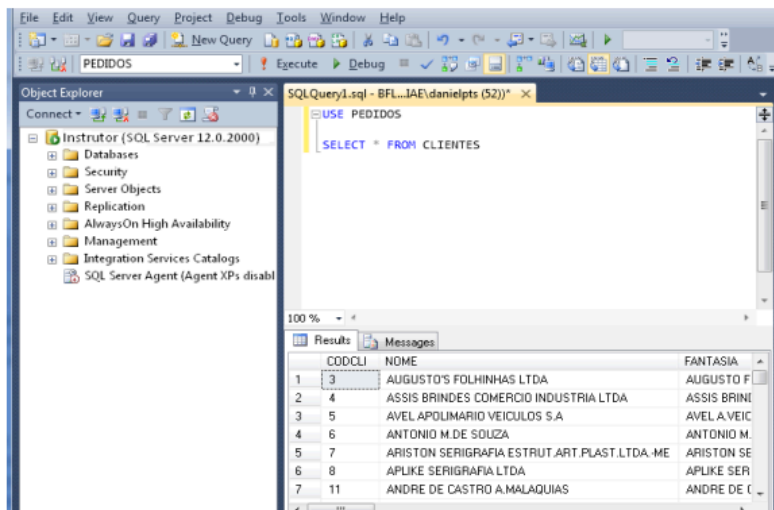
A seguir, descreveremos as funcionalidades oferecidas pelas ferramentas de gerenciamento disponíveis no SQL Server e que trabalham associadas aos componentes descritos anteriormente.

### **SSMS - SQL Server Management Studio**

O SSMS, ou SQL Server Management Studio, é um software da Microsoft usado para configurar, gerenciar e administrar todos os componentes do Microsoft SQL Server. Ele foi lançado pela primeira vez em 2005 como sucessor do Enterprise Manager, presente no SQL Server 2000 e versões anteriores.

O SSMS é uma ferramenta essencial para administradores de banco de dados e desenvolvedores que trabalham com SQL Server. Ele oferece uma interface amigável e intuitiva, facilitando a navegação e a execução de tarefas.

É um aplicativo usado para gerenciar bancos de dados e que permite criar, alterar e excluir objetos no banco de dados:



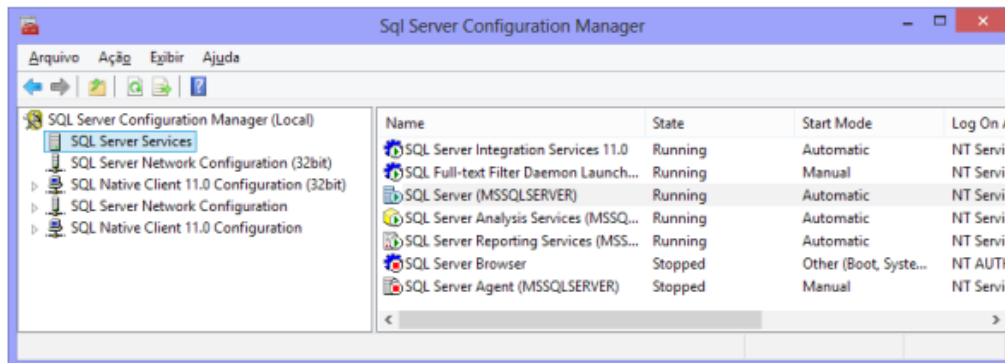
Aqui estão alguns dos recursos do SSMS:

- **Explorador de Objetos:** Permite navegar por todos os objetos do servidor SQL Server, como bancos de dados, tabelas, visualizações e procedimentos armazenados.
- **Editor de Consultas:** Uma janela onde você pode escrever e executar consultas Transact-SQL (T-SQL) para gerenciar dados no banco de dados.
- **Suporte para Script:** SSMS permite escrever scripts T-SQL complexos para automatizar tarefas de banco de dados.
- **Monitoramento:** Fornece ferramentas para monitorar o desempenho do servidor SQL Server e identificar gargalos.

Ferramenta que possui ambiente integrado que permite acessar, configurar, gerenciar, administrar e desenvolver todos os componentes do SQL Server.

### SQL Server Configuration Manager

Permite visualizar, alterar e configurar os serviços dos componentes do SQL Server.



## Microsoft SQL Server Profiler

Essa ferramenta permite capturar e salvar dados de cada evento em um arquivo ou tabela para análise posterior.

## Database Engine Tuning Advisor

Analisa o desempenho das operações e sugere opções para sua melhora.

## SQL Server Data Tools

Possui uma interface que integra os componentes Business Intelligence, Analysis Services, Reporting Services e Integration Services.

## Modelo Físico

O modelo físico de dados pode ser obtido por meio do diagrama lógico de dados e está associado ao software de gerenciamento de banco de dados, neste caso, o SQL Server 2016.

Vejamos as definições:

- **Tabelas (entidades):** Local de armazenamento das informações;
- **Campos (atributos):** Características da tabela;
- **Chave primária:** Campo único que define a exclusividade da linha. Toda chave primária possui as seguintes características:
  - Valores únicos;
  - Não permite valores nulos;
  - A tabela será ordenada pela chave primária (índice clusterizado).
- **Relacionamento:** Relação entre tabelas através de um ou mais campos. As relações podem ser:
  - 1 para 1;
  - 1 para N;
  - N para N.

Vejamos, a seguir, um exemplo de modelo físico de dados:



## Pontos principais

Atente para os tópicos a seguir. Eles devem ser estudados com muita atenção, pois representam os pontos mais importantes da leitura.

- Um banco de dados armazena informações e seu principal objeto são as tabelas;
- É fundamental o design de um banco de dados para que possua um bom desempenho;
- Os modelos de design de um banco de dados são: modelo descritivo, modelo conceitual, modelo lógico e modelo físico;
- Uma tabela precisa ter uma coluna que identifica de forma única cada uma de suas linhas. Essa coluna é chamada de chave primária;
- Normalização é o processo de organizar dados e eliminar informações redundantes de um banco de dados. Envolve a tarefa de criar as tabelas, bem como definir relacionamentos. O relacionamento entre as tabelas é criado de acordo com regras que visam à proteção dos dados e à eliminação de dados repetidos. Essas regras são denominadas normal forms, ou formas normais;
- A linguagem T-SQL (Transact-SQL) é baseada na linguagem SQL ANSI, desenvolvida pela IBM na década de 1970;
- Os principais objetos de um banco de dados são: tabelas, índices, CONSTRAINT, VIEW, PROCEDURE, FUNCTION e TRIGGER;
- O SQL Server Management Studio (SSMS) é a principal ferramenta para gerenciamento de bancos de dados.

## Pontos principais - I

Atente para os tópicos a seguir. Eles devem ser estudados com muita atenção, pois representam os pontos mais importantes da leitura.

- Os objetos que fazem parte de um sistema são criados dentro de um objeto denominado database, ou seja, uma estrutura lógica formada por dois tipos de arquivo: um responsável pelo armazenamento de dados e outro que armazena as transações feitas. Para que um banco de dados seja criado no SQL Server, é necessário utilizar a instrução CREATE DATABASE;
- Os dados de um sistema são armazenados em objetos denominados tabelas (tables). Cada uma das colunas de uma tabela refere-se a um atributo associado a uma determinada entidade. A instrução CREATE TABLE deve ser utilizada para criar tabelas dentro de bancos de dados já existentes;



- Cada elemento, como uma coluna, uma variável ou uma expressão, possui um tipo de dado. O tipo de dado especifica o tipo de valor que o objeto pode armazenar, como números inteiros, texto, data e hora etc.;
- Normalmente, as tabelas possuem uma coluna contendo valores capazes de identificar uma linha de forma exclusiva. Essa coluna recebe o nome de chave primária, cuja finalidade é assegurar a integridade dos dados da tabela;
- As constraints são objetos utilizados com a finalidade de definir regras referentes à integridade e à consistência nas colunas das tabelas que fazem parte de um sistema de banco de dados;
- Para assegurar a integridade dos dados de uma tabela, o SQL Server oferece cinco tipos diferentes de constraints: PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK e DEFAULT;
- Cada uma das constraints possui regras de utilização. Uma coluna que é definida como chave primária, por exemplo, não pode aceitar valores nulos. Em cada tabela, pode haver somente uma constraint de chave primária;
- Podemos criar constraints com o uso de CREATE TABLE, ALTER TABLE ou graficamente (a partir da interface do SQL Server Management Studio).

## Índices

Os índices em SQL são estruturas de dados que servem para acelerar a recuperação de dados em tabelas. São como os índices de um livro, que te permitem encontrar rapidamente a página que você procura sem precisar ler o livro inteiro.

### Tipos de índices

Os índices podem ser categorizados em dois tipos principais: clusterizados e não clusterizados.

#### Índice clusterizado

Um índice único que ordena fisicamente os dados da tabela. Cada tabela pode ter apenas um índice clusterizado.

#### Índice não clusterizado

Um índice que não ordena fisicamente os dados da tabela. Uma tabela pode ter vários índices não clusterizados.

### Benefícios dos índices

- Aceleram as consultas: Permitem que o SQL Server encontre rapidamente os dados que você procura sem precisar ler toda a tabela.
- Melhoram o desempenho: Reduzem o tempo necessário para executar consultas complexas.
- Permitem consultas mais eficientes: Otimizam o uso de recursos do servidor.

### Quando usar índices

- Em colunas frequentemente usadas em consultas: WHERE, JOIN, ORDER BY, GROUP BY.
- Em colunas com muitos valores distintos: Aumentam a seletividade do índice.
- Em colunas com valores que mudam frequentemente: Permitem que o SQL Server mantenha os dados atualizados.

## Cuidados com os índices

- Aumentam o espaço em disco utilizado: O índice precisa armazenar uma cópia dos dados da coluna.
- Podem diminuir o desempenho de INSERTs, UPDATEs e DELETEs: O índice precisa ser atualizado a cada modificação na coluna.
- Devem ser criados com cuidado: Índices desnecessários podem prejudicar o desempenho.

## Exemplos de criação de índices

Code:

-- Criar um índice clusterizado na coluna "Nome"

```
CREATE TABLE Clientes (  
  ClientId INT PRIMARY KEY,  
  Nome VARCHAR(255) NOT NULL,  
  ...  
) WITH (CLUSTERED INDEX (Nome));
```

-- Criar um índice não clusterizado na coluna "DataNascimento"

```
CREATE INDEX idx_DataNascimento ON Clientes (DataNascimento);
```

## Pontos principais - II

Atente para os tópicos a seguir. Eles devem ser estudados com muita atenção, pois representam os pontos mais importantes da leitura.

- Para acrescentar novas linhas de dados em uma tabela, utilizamos o comando INSERT;
- No INSERT posicional não é necessário mencionar o nome dos campos, porém será obrigatório incluir todos os campos, exceto o campo autonumerável. Já no INSERT declarativo, os campos deverão ser informados no comando;
- Podemos restringir a quantidade de inserções do comando por meio da cláusula TOP;
- Para auditar e mostrar quais dados foram inseridos, utilizamos a cláusula OUTPUT;
- Os dados pertencentes a múltiplas linhas de uma tabela podem ser alterados por meio do comando UPDATE;
- O comando DELETE deve ser utilizado quando desejamos excluir os dados de uma tabela;
- Transações são unidades de programação capazes de manter a consistência e a integridade dos dados. Devemos considerar uma transação como uma coleção de operações que executa uma função lógica única em uma aplicação de banco de dados;
- Todas as alterações de dados realizadas durante a transação são submetidas e tornam-se parte permanente do banco de dados caso a transação seja executada com êxito. No entanto, caso a transação não seja finalizada com êxito por conta de erros, são excluídas quaisquer alterações feitas sobre os dados.

## Pontos principais II

Atente para os tópicos a seguir. Eles devem ser estudados com muita atenção, pois representam os pontos mais importantes da leitura.

- Na linguagem SQL, o principal comando utilizado para a realização de consultas é o SELECT. Pertencente à categoria DML (Data Manipulation Language), esse comando é utilizado para consultar todos os dados de uma fonte de dados ou apenas uma parte específica deles;
- Às vezes, é necessário que o resultado da consulta de dados seja fornecido em uma ordem específica, de acordo com um determinado critério. Para isso, contamos com opções e cláusulas. Uma delas é a cláusula ORDER BY, que considera certa ordem para retornar dados de consulta;
- A cláusula WHERE é utilizada para definir critérios com o objetivo de filtrar o resultado de uma consulta. As condições definidas nessa cláusula podem ter diferentes propósitos, tais como a comparação de dados na fonte de dados, a verificação de dados de determinadas colunas e o teste de colunas nulas ou valores nulos;
- UNION permite a união de duas ou mais consultas em uma única;
- Uma forma de podermos comparar consultas é a utilização dos comandos INTERSECT e EXCEPT.

### Pontos principais III

Atente para os tópicos a seguir. Eles devem ser estudados com muita atenção, pois representam os pontos mais importantes da leitura.

- A associação de tabelas pode ser realizada, por exemplo, para converter em informação os dados encontrados em duas ou mais tabelas. As tabelas podem ser combinadas por meio de uma condição ou um grupo de condições de junção;
- É importante ressaltar que as tabelas devem ser associadas em pares, embora seja possível utilizar um único comando para combinar várias tabelas. Um procedimento muito comum é a associação da chave primária da primeira tabela com a chave estrangeira da segunda tabela;
- JOIN é uma cláusula que permite a associação entre várias tabelas, com base na relação existente entre elas. Por meio dessa cláusula, os dados de uma tabela são utilizados para selecionar dados pertencentes à outra tabela;
- Há diversos tipos de JOIN: INNER JOIN, OUTER JOIN (LEFT JOIN, RIGHT JOIN e FULL JOIN) e CROSS JOIN

### Pontos principais IV

Atente para os tópicos a seguir. Eles devem ser estudados com muita atenção, pois representam os pontos mais importantes da leitura.

- O uso da instrução UPDATE em uma subquery permite atualizar linhas de uma tabela com informações provenientes de outra tabela;
- Subqueries podem ser utilizadas com o intuito de remover dados de uma tabela. Basta definirmos a cláusula WHERE da instrução DELETE como uma subquery para excluir linhas de uma tabela base conforme os dados armazenados em uma outra tabela;
- Podemos utilizar uma associação (JOIN) em tabelas para determinar quais colunas serão atualizadas por meio de UPDATE;
- Os dados de tabelas associadas podem ser eliminados por meio da cláusula JOIN com o comando DELETE.

### Pontos principais V

Atente para os tópicos a seguir. Eles devem ser estudados com muita atenção, pois representam os pontos mais importantes da leitura.

- Uma consulta aninhada em uma instrução SELECT, INSERT, DELETE ou UPDATE é denominada subquery (subconsulta). As subqueries são também referidas como queries internas. Já a instrução em que está inserida a subquery pode ser chamada de query externa;
- Vejamos algumas das diversas características das subqueries: podem ser escalares (retornam apenas uma linha) ou tabulares (retornam linhas e colunas). Elas, que podem ser incluídas dentro de outras subqueries, devem estar entre parênteses, o que as diferenciará da consulta principal;
- Uma subquery retornará uma lista de zero ou mais valores caso tenha sido introduzida com a utilização de IN ou NOT IN. O resultado, então, será utilizado pela query externa;
- O sinal de igualdade (=) pode ser utilizado para inserir subqueries;
- Quando uma subquery possui referência a uma ou mais colunas da query externa, ela é chamada de subquery correlacionada. Trata-se de uma subquery repetitiva, pois é executada uma vez para cada linha da query externa. Desta forma, os valores das subqueries correlacionadas dependem da query externa, o que significa que, para construir uma subquery desse tipo, será necessário criar tanto a query interna como a externa;
- Ao comparar subqueries e associações (JOINS), é possível constatar que as associações são mais indicadas para verificação de existência, pois apresentam desempenho melhor nesses casos;
- A visualização do aninhamento e da organização da query é mais simples em uma subquery, enquanto que, em uma consulta com diversas associações, a visualização pode ser complicada;
- Tabelas temporárias são armazenadas no tempdb e excluídas automaticamente após terem sido utilizadas;
- As tabelas temporárias locais apresentam, antes do nome, o símbolo # e são visíveis somente durante a conexão atual. Quando o usuário desconecta-se da instância do SQL Server, ela é excluída. Já as tabelas temporárias globais apresentam, antes do nome, dois símbolos ## e são visíveis para todos os usuários. Uma tabela desse tipo será excluída apenas quando todos os usuários que a referenciam se desconectarem da instância do SQL Server;
- A escolha da utilização de tabelas temporárias ou subqueries dependerá de cada situação e de aspectos como desempenho do sistema e até mesmo das preferências pessoais de cada usuário.