# DICA DO MÊS – TEMPORAL TABLE E O CALOR, UMA COMBINAÇÃO MUITO QUENTE

*[22 22America/Sao\_Paulo janeiro 22America/Sao\_Paulo 2019](https://pedrogalvaojunior.wordpress.com/2019/01/22/dica-do-mes-temporal-table-e-o-calor-uma-combinacao-muito-quente/)*

Salve pessoal, bom dia.

Estamos no mês de janeiro, férias, sol, calor, chuvas, e para minha alegria te encontro mais uma vez no meu blog, caso esta seja a sua primeira visita ou acesso, fico mais feliz ainda, seja muito bem vindo.

Este é mais um post da sessão **Dica do Mês**, sessão dedicada a compartilhar bimestralmente dicas, novidades, curiosidades e demais assuntos, conteúdos e informações relacionadas ao Microsoft SQL Server, Banco de Dados e Tecnologias de Banco de Dados.

No post de hoje, quero compartilhar com vocês uma das funcionalidades adicionadas ao Microsoft SQL Server a partir da versão **2016** e que recentemente acabei conhecendo com um pouco mais, como você já pode notar no título deste post, estou fazendo referência as chamadas Temporal Tables (Tabelas Temporais).

Você já conhece? Teve a necessidade de utilizar? Eu particularmente falando conhecia muito pouco sobre este recurso, mas na semana passada neste período de férias tive a ideia de fazer uma brincadeira aqui em casa em conjunto com um termômetro, e justamente através desta brincadeira que utilizei uma temporal table.

Ficou curioso para saber como eu fiz uso dela? Calma, daqui a pouco eu conto mais sobre isso para você.

Pois bem, sem mais delongas, vamos em frente, vou tentar mitigar a sua curiosidade e ao mesmo também satisfazer os meus objetivos. Sendo assim, seja bem vindo ao post – **Dica do Mês – Temporal Table e o Calor, uma combinação muito quente.**

### **INTRODUÇÃO**

A partir da versão 2016 do Microsoft SQL Server, a Microsoft introduziu o suporte para tabelas temporais de sistema baseadas no versionamento de dados como um recurso de banco de dados, sendo este, uma funcionalidade que traz o suporte interno para fornecer informações sobre dados armazenados na tabela em qualquer ponto no tempo, ao invés de apenas os dados que é corretos no momento atual em está na hora.

Esta nova funcionalidade, também é reconhecida e trata como um recurso de banco de dados criado com base nos padrões em **ANSI SQL 2011**.

A partir do momento em que idealizamos fazer uso de uma tabela temporal, estamos criando um novo objeto ou transformando um objeto já existente em nosso banco de dados, em um elemento responsável em manter o histórico completo das alterações de dados ocorridos durante um período de tempo, sendo esta a principal finalidade de uso de uma temporal que é tratada internamento como um repositório de gerenciamento de tempo.

Cada tabela temporal tem duas colunas explicitamente definidas, cada um com um tipo de dados **datetime2** , estas colunas são referidas como colunas de **período,** sendo período colunas usadas exclusivamente pelo sistema de registro prazo de validade para cada linha, sempre que uma linha for modificada. Além dessas colunas de período, uma tabela temporal também contém uma referência a outra tabela, a qual será utilizada como esquema **espelho**.

Por padrão o Microsoft SQL Server utiliza esta tabela para armazenar automaticamente a versão anterior de uma linha cada vez que a mesma na tabela temporal é atualizada ou excluída. Esta tabela adicional é referida como a tabela de **histórico**, enquanto a tabela principal que **armazena versões de linha (real)** atual é ***conhecida como a tabela atual ou simplesmente como a tabela temporal***.

Importante ressaltar que durante a criação do quadro temporal, os usuários podem especificar a existência de uma tabela de histórico (deve ser esquema compatível) ou deixar o sistema criar tabela de histórico padrão.

Agora que já conhecemos um pouco do que é uma Temporal Table, vamos avançar mais um pouco em nossa caminhada, vou apresentar o porque tive a ideia de fazer uso deste recurso.

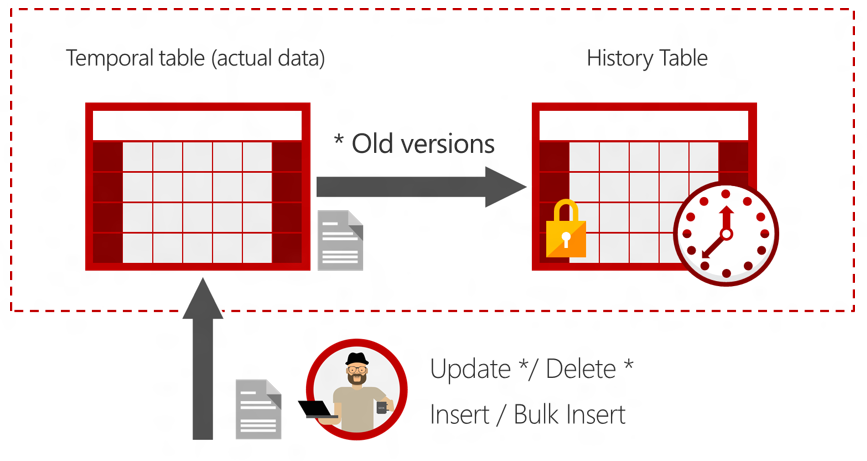
### **SEU FUNCIONAMENTO**

Como já destacado anteriormente o sistema de controle de versão de uma tabela temporal é implementado através do uso de um par de tabelas, uma tabela atual e uma tabela de histórico. Dentro de cada uma destas tabelas, as seguintes duas colunas adicionais **datetime2** são usadas para definir o período de validade para cada linha:

* **Coluna de início de período:** O sistema registra a hora de início para a linha nesta coluna, denotado tipicamente como a coluna de **SysStartTime** .
* **Coluna de fim do período:** O sistema registra a hora final para a linha nesta coluna, normalmente indicado na coluna **SysEndTime** .

A tabela atual contém o valor atual para cada linha. A tabela de histórico contém cada valor anterior para cada linha, se for o caso, e a hora de início e hora de término para o período para o qual foi válido.

A *Figura 1* apresentada abaixo, ilustra de forma simples o funcionamento do sistema de controle dos dados aplicado a partir do uso de uma tabela temporal:



*Figura 1 – Funcionamento do sistema de controle de uma tabela temporal.*

Este sistema de controle de versionamento dos dados é realizado sempre as instruções: **Insert, Update, Delete ou Merge** venham a ser realizadas de forma individual ou simultânea.

### **PORQUE UTILIZAR UMA TEMPORAL TABLE**

Uma das coisas que eu aprendi a gostar no decorrer da minha carreira na área de tecnologia é a importância e as possibilidades de mudanças que um mesmo dado pode apresentar no decorrer de um período de tempo, este é um dos meus maiores prazeres entender o quanto aquele dado a uma minuto atrás agora já é outro dado e podem me trazer representar novas informações e conhecimentos.

Desta forma, ao analisarmos uma temporal table podemos também reconhecer ou fazer uso da mesma como uma ***Slowly Changing Dimension*** (Dimensão com mudanças lentas ou mudanças lentas em uma dimensão), o que vai nos possibilitar criar uma visão dos nossos dados com base uma período ou determinada data.

Uma outra funcionalidade que pode ser aplicada a uma temporal table se relacionada a controles de auditoria mais propriamente falando de auditoria de dados, normalmente as fontes de dados reais são dinâmicas e se tornam voláteis ao longo do tempo, para uma empresas isso pode influenciar diretamente em suas decisões as quais dependem de percepções que os analistas podem começar a identificar a partir da evolução ou mudanças de dados.

Já sabemos o porque escolhi fazer uso de uma temporal table, agora vou apresentar o cenário que me permitiu aplicar este recurso com base na minha ideia.

### **MINHA IDEIA**

Estamos visando uma forte onda de calor em praticamente todo o Brasil, algo que muitos brasileiros adoram eu sinceramente não sou um destes brasileiros, pois eu não suporto estas altas temperaturas.

Para tentar de alguma maneira aprender algo de novo com este calor e tentando se distrair dentro das possibilidades, pensei em ter uma noção do quanto a temperatura aqui na minha casa localizada na cidade de São Roque interior do estado de São Paulo muda no decorrer de um período de tempo, sendo justamente esta a minha ideia de utilizar uma temporal table, talvez esta não tenha sido a melhor ideia ou até mesmo o melhor cenário para uso, mas entendo que pode ser uma possibilidade dentre as mais variadas possíveis.

Seguindo em frente e avançando mais um pouco, chegou a hora de colocar em prática a minha ideia, para isso vamos construir um simples cenário para fazer uso da Temporal Table.

### **NOSSO AMBIENTE**

Como de costume vamos utilizar um ambiente isolado dos demais bancos de dados que você possa conter, desta maneira nosso cenário será constituído dos seguintes elementos:

* Banco de Dados: **DatabaseTemporalTabel**;
* Tabela Atual: **TemporalTableTemperatura**;
* Tabela Historico: **TemporalTableTemperaturaHistorico**;
* Colunas Temporais: **DataHoraInicial** e **DataHoraFinal**; e
* Period For System formado por: **DataHoraInicial** e **DataHoraFinal.**

## CRIANDO O AMBIENTE

Através do ***Bloco de Código 1*** apresentado abaixo, vamos realizar a criação dos respectivos elementos destacados anteriormente:

**— Bloco de Código 1 —**

**— Criando o Banco de Dados —**

Create Database DatabaseTemporalTable

Go

**— Acessando o Banco de Dados —**

Use DatabaseTemporalTable

Go

**— Criando a Tabela TemporalTableTemperatura —**

Create Table TemporalTableTemperatura

(Codigo Int Identity(1,1) Primary Key Clustered,

Local Char(10) Default ‘Minha Casa’,

Cidade Char(9) Default ‘São Roque’,

DataAtual Date Default GetDate(),

HoraAtual Time Default GetDate(),

Temperatura TinyInt,

**DataHoraInicial** Datetime2 (0) *GENERATED ALWAYS AS ROW START*,

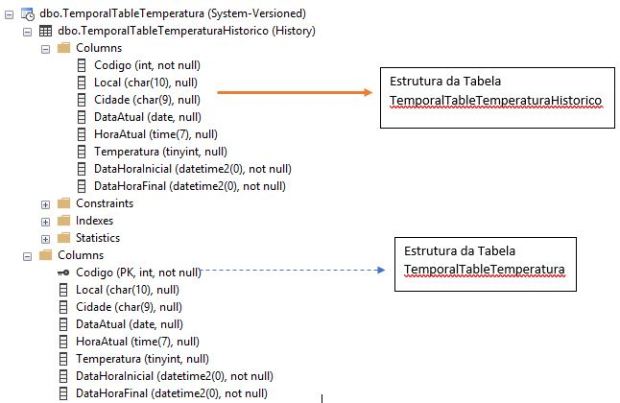
**DataHoraFinal** Datetime2 (0) *GENERATED ALWAYS AS ROW END*,

PERIOD FOR SYSTEM\_TIME (DataHoraInicial, DataHoraFinal))

WITH (SYSTEM\_VERSIONING = ON (HISTORY\_TABLE = dbo.TemporalTableTemperaturaHistorico))

Go

A *Figura 2* apresentada abaixo, ilustra a estrutura da tabela TemporalTableTemperatura e sua tabela espelho TemporalTableTemperaturaHistorico:



*Figura 2 – Tabelas TemporalTableTemperatura e TemporalTableTemperaturaHistorico.*

**Observações**

1 – Para que o Microsoft SQL Server reconheça uma tabela como Temporal Table as colunas temporais devem ser formadas pelo tipo de dados DateTime2 e logo após a declaração do seu tipo de dados informar as instruções:

* **Generated Always as Row Start** – Valor gerado sempre no início da linha; e
* **Generated Always as Row End** – Valor gerado sempre no final da linha.

2 – O controle do período dos valores é feito através da instrução **PERIOD FOR SYSTEM\_TIME,** declarada obrigatoriamente no final da construção da tabela, formada pelas colunas que recebem os valores **DateTime2**.

3 – Ao declarar o nome da tabela a ser utilizada para o versionamento dos dados, é obrigatório informar o nome do ower ou schema a qual esta tabela irá pertencer, caso isso não seja feito o Microsoft SQL Server retornará a seguinte mensagem de erro:

**Msg 13539, Level 15, State 1, Line 18**

**Setting SYSTEM\_VERSIONING to ON failed because history table ‘TemporalTableTemperaturaHistorico2 is not specified in two-part name format.**

4 – Ao informar a tabela que será utilizada para o versionamento dos dados o Database Engine realiza automaticamente a criação desta tabela histórico caso a mesma não exista.

Ótimo estamos no caminho certo, nosso próximo passo será abastecer a tabela TemporalTableTemperatura com dados iniciais e na sequência proporcionar alterações nestes mesmos dados iniciais para que o Database Engine faça uso da nossa Temporal Table registrando na Tabela TemperalTableTemperaturaHistorico todas as manipulações realizadas.

Para isso vamos utilizar o ***Bloco de Código 2*** declarado abaixo:

**— Bloco de Código 2 —**

**— Inserindo Dados na Tabela TemporalTableTemperatura —**

Insert Into TemporalTableTemperatura (Temperatura)

Values (25)

Go

**— Gerando um Delay de 20 segundos —**

WAITFOR DELAY ’00:00:20′

Go

**— Atualizando os dados na Tabela TemporalTableTemperatura —**

Update TemporalTableTemperatura

Set Temperatura = 26,

HoraAtual = GetDate()

Go

**— Gerando um novo Delay de 40 segundos —**

WAITFOR DELAY ’00:00:40′

Go

**— Atualizando os dados na Tabela TemporalTableTemperatura —**

Update TemporalTableTemperatura

Set Temperatura = 27,

HoraAtual = GetDate()

Go

**— Gerando um novo Delay de 1 minuto e 20 segundos —**

WAITFOR DELAY ’00:01:20′

Go

**— Atualizando os dados na Tabela TemporalTableTemperatura —**

Update TemporalTableTemperatura

Set Temperatura = 27,

HoraAtual = GetDate()

Go

Até aqui tudo tranquilo, realizamos o processo de inserção de dados iniciais na tabela TemporalTableTemperatura e na sequência através do comando WaitFor forçamos a ocorrência de alguns delays (atrasos) de tempo para simular o aumento da temperatura como se fosse um termômetro realizando uma nova marcação, com isso, já temos neste momento um pequena porção de dados a serem consultados.

Vamos então executar o ***Bloco de Código 3*** a seguir para identificar as possíveis maneiras de se consultar os dados armazenados em nossa temporal table:

**— Bloco de Código 3 —**

— Consultando dados na Tabela TemporalTableTemperatura —

Select \* From TemporalTableTemperatura

Go

Após realizarmos o Select declarado acima teremos um retorno de dados similar ao apresentado na *Figura 3 abaixo:*

*[Figura 3 – Posição atual de dados armazenados na tabela TemporalTableTemperatura.](https://pedrogalvaojunior.files.wordpress.com/2019/01/temporaltable-figura-2.jpg)*

Observe que a coluna Temperatura apresenta o valor **27**, número informado no último update realizado, a coluna **DataHoraInicial** apresentando o valor que representa o início da realização da última manipulação aplicada a tabela, no caso o comando Update e a coluna **DataHoraFinal** vai apresentar o valor final que representa o encerramento do período de controle de versionamento dos dados com o **valor fixo e padrão 9999-12-31 23:59:59**.

Pois bem, mas se quisermos então identificar ao longo do tempo todas as manipulação que podem ter ocorrido em nossa tabela temporal? É ai que entra em ação nossa tabela de espelho, nossa tabela **TemporalTableTemperaturaHistorico**, a qual é responsável em armazenar e controlar todo versionamento e alterações que venham a ser realizadas em nossa Temporal Table.

O próximo passo consiste na execução do ***Bloco de Código 4***, o qual vai nos permitir consumir os dados temporais armazenados em nossa tabela **TemporalTableTemperaturaHistorico:**

**— Bloco de Código 4 —**

**— Consultando dados Temporais, obtendo todas as manipulações realizadas —**

Select \* From TemporalTableTemperatura

For **System\_Time All — Apresenta todas as manipulações realizadas**

Go

*Figura 4 – Todas as manipulações realizadas na tabela TemporalTableTemperatura armazenadas de forma espealhada na tabela histórico TemporalTableTemperaturaHistorico.*

Nota que a coluna **DataHoraFinal** apresenta na linha 1 o valor fixo e padrão **9999-12-31**23:59:59, mas no decorrer das demais linhas, de acordo com as operações realizadas os valores foram sendo atualizados, como podemos comprovar na linha 7 a qual apresenta o valor **2019-01-22 12:59:42**.

Já estamos praticamente no final desta caminhada, nosso últimos passos consistem em realizar outras formas de consultar dados temporais, através das instruções:

* **For System\_Time as Of;**
* **For System\_Time From ” To ”;**
* **For System\_Time Between ” And ”; e**
* **For System\_Time Contained In ().**

Para realizar estas consultamos, vamos executar o ***Bloco de Código 5*** apresentando abaixo:

**— Bloco de Código 5 —**

**— Conhecendo outras formas de consultar dados temporais —**

Select \* From TemporalTableTemperatura

For System\_Time as Of ‘2019-01-22 12:33:56’

Go

Select \* From TemporalTableTemperatura

For System\_Time From ‘2019-01-22 12:33:56’ To ‘2019-01-22 12:48:36’

Go

Select \* From TemporalTableTemperatura

For System\_Time Between ‘2019-01-22 12:48:36’ And ‘2019-01-22 12:58:22’

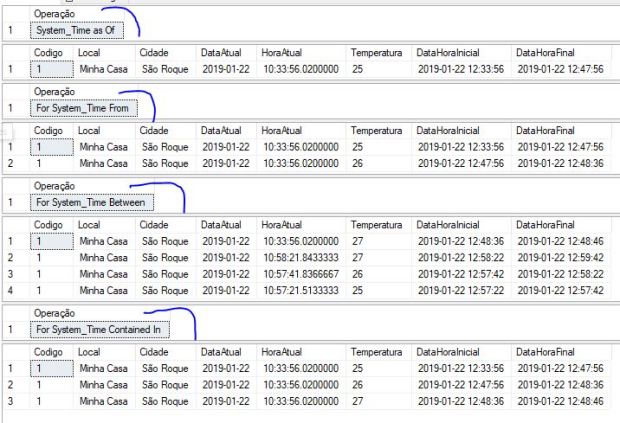
Order By Temperatura Desc

Go

Select \* From TemporalTableTemperatura

For System\_Time Contained In (‘2019-01-22 12:33:00′ ,’2019-01-22 12:55:00’)

Go

A *Figura 5* a seguir apresentado o resultado tornado após a execução do *Bloco de Código 5*declarado acima:*Figura 5 – Resultados obtidos após a execução de cada comando select declarado no Bloco de Código 5.*

Praticamente términos, mas quero finalizar este post com uma pequena amostra do quanto uma tabela temporal pode ser útil, imagine se excluirmos todos os dados da nossa tabela TemporalTableTemperatura.

**O que aconteceria com os dados em nossa tabela espelho:**

**1 – Os dados seriam excluídos também?**

**2 – Os dados são mantidos?**

**3 – A tabela espelho será excluída?**

**4 – Não podemos remover dados em tabelas que utilizam versionamento de dados?**

Bom, vou deixar o ***Bloco de Código 6*** declarado abaixo, mas a respostas para esta pergunta você que vai descobrir e posteriormente publicar seu comentário aqui neste post:

**— Bloco de Código 6 —**

**— Excluíndo os dados cadastrados na Tabela TemporalTableTemperatura —**

Delete From TemporalTableTemperatura

Go

**— Consultando dados na Tabela TemporalTableTemperaturaHistorico —**

Select Local, Cidade, DataAtual, HoraAtual, Temperatura

From TemporalTableTemperaturaHistorico

Go