Banco de Dados

Módulo Intermediário

STORED PROCEDURES

Procedures

- ***Stored Procedures***
  - ***Conjunto*** *de* ***instruções*** *(****uma*** *ou* ***mais****)* ***SQL****, normalmente* ***agrupadas*** *para* ***executar*** *uma* ***rotina específica***
  - *Eles podem ser* ***criados*** *em* ***qualquer BD definido*** *pelo* ***usuário*** *e de* ***sistema***
  - ***São comparáveis*** *às* ***funções*** *de* ***várias instruções****, mas possuem* ***recursos*** *e* ***flexibilidades*** ***impossíveis*** *dentro de* ***funções***

Procedures

- ***Stored Procedures***
  - ***Vantagens** de usar **stored procedures:***
    - ***Oferecem maior desempenho**, devido ao **código compilado***
    - *São **fáceis** de **manter**, pois as **alterações** são **centralizadas**, em vez de **serem feitas** com **código***
    - *Como as **operações** de **BD** podem ser **efetuadas** dentro dos **stored procedures**, eles **oferecem** um **alto nível** de **segurança***
    - *Em vez do **acesso** ser **concedido** ao **objeto subjacente**, a **permissão** pode ser dada **somente** para o **stored procedure***
    - *Os **stored procedures** criam um **nível** de **abstração** para **permissões**, em vez de se **conceder** ao **usuário direitos** **SELECT**, **INSERT**, **UPDATE** ou **DELETE**, ele pode **receber direitos** **EXECUTE** para um **stored procedure***

Procedures

- ***Stored Procedures***
  - *O **Microsoft SQL Server 2012** tem **quatro tipos** de **stored procedure:***
    - ***Definido** pelo **usuário***
    - ***Sistema***
    - ***Temporário***
    - ***Definido pelo usuário estendido***
  - *Os **stored procedures definidos** pelo **usuário estendidos** foram **substituídos** por **procedimentos** Common Language Runtime (**CLR**)*

Procedures

- ***Stored Procedures***
  - ***Sintaxe (Stored Procedures definidos** pelo **usuário):***

```
CREATE [ OR ALTER ] { PROC | PROCEDURE }
  [nome_do_esquema.] nome_do_procedimento [ ; número ]
  [ { @parâmetro [ nome_do_esquema_do_tipo. ] tipo_de_dado }
  [ VARYING ] [ = padrão ] [ OUT | OUTPUT ] [READONLY]
  ] [,...n]
  [WITH <opção_do_procedimento> [,...n]]
  [ FOR REPLICATION ]
  AS { [ BEGIN ] instrução_sql [;] [ ...n ] [END] }
  [;]
  <opção_do_procedimento> ::=
   [ ENCRYPTION ]
   [ RECOMPILE ]
   [ EXECUTE AS clause ]
```

Procedures

- ***Stored Procedures***
  - ***Sintaxe (Stored Procedures definidos** pelo **usuário):***
    - ***; número:** um **inteiro opcional** usado para **agrupar procedimentos** do **mesmo nome***
    - ***@parâmetro: parâmetro declarado** no **procedimento***
    - ***tipo_de_dado: tipo** de **dado** do **parâmetro** e o **esquema** ao qual o **tipo** de **dado pertence***
    - ***VARYING: especifica** o **conjunto** de **resultados** com **suporte** como um **parâmetro** de **saída***
    - ***padrão:** um **valor padrão** para um **parâmetro***
    - ***OUT | OUTPUT: indica** que o **parâmetro** é um **parâmetro** de **saída***

Procedures

- ***Stored Procedures***
  - ***Sintaxe (Stored Procedures definidos** pelo **usuário):***
    - ***READONLY:** indica que o **parâmetro não** pode ser **atualizado** nem **modificado dentro** do **corpo** do **procedimento***
    - ***RECOMPILE:** indica que o **BD não armazena** em **cache** um **plano** de **consulta** para este **procedimento**, forçando-o a ser **compilado sempre** que for **executado***
    - ***ENCRYPTION:** indica que o **SQL Server** converte o **texto original** da **instrução CREATE PROCEDURE** em um **formato ofuscado***
    - ***EXECUTE AS clause:** especifica o **contexto** de **segurança** no qual o **procedimento** deve ser **executado***

Procedures

- ***Stored Procedures***
  - ***Sintaxe (Stored Procedures definidos** pelo **usuário):***
    - ***FOR REPLICATION**: **especifica** que o **procedimento** é **criado** para **replicação***
    - *{ [**BEGIN**] instrução_sql [;] [...n] [**END**] }**: uma** ou **mais instruções T-SQL** que **abrangem** o **corpo** do **procedimento***

Procedures

- ***Stored Procedures***
  - ***Exemplo (01): - Parte 1***
    - ***Criando** um **stored procedure simples** (PurchaseOrderInformation) que **não espera nenhum parâmetro** e **contém apenas** uma **instrução T-SQL***

```
USE MyAdventureWorks;

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- continua…
```

Procedures

- ***Stored Procedures***
  - ***Exemplo (01): - Parte 2***

```
CREATE OR ALTER PROCEDURE dbo.PurchaseOrderInformation
AS
  BEGIN
    SELECT poh.PurchaseOrderID, pod.PurchaseOrderDetailID,
           poh.OrderDate, poh.TotalDue, pod.ReceivedQty,
           p.Name ProductName
        FROM Purchasing.PurchaseOrderHeader poh
        INNER JOIN Purchasing.PurchaseOrderDetail pod
          ON (poh.PurchaseOrderID = pod.PurchaseOrderID)
        INNER JOIN Production.Product p
          ON (pod.ProductID = p.ProductID)
  END
GO
```

Procedures

- ***Palavra-chave EXECUTE***
  - *Para **executar** um **stored procedure** com **T-SQL**, utilizamos a **palavra-chave EXECUTE***
  - ***Sintaxe:***

```
EXECUTE | EXEC nome_do_procedimento [parâmetro1, parâmetro2, n…]
```

Procedures

- ***Palavra-chave EXECUTE***
  - *Na **versão mais recente** do **SQL Server**, podemos **mudar** os **nomes** e **tipos** de **dados** de **cada coluna** no **conjunto** de **resultados** ou **redefinir** esse **conjunto***
  - ***Sintaxe:***

```
EXECUTE | EXEC nome_do_procedimento [parâmetro1, parâmetro2, n…]
WITH RESULT SETS
(
    ( [definição_da_coluna1, definição_da_coluna2, n…] )
)
```

  - *Para **alterar** o **conjunto** de **resultados**, **execute** o **comando** **EXECUTE** **normalmente**, mas **adicione uma instrução** **WITH RESULT SETS** e, **dentro** dos **parênteses**, forneça uma **definição** para **cada coluna** do **conjunto** de **resultados***

Procedures

- ***Palavra-chave EXECUTE***
  - ***Exemplo (02): - Parte 1***
    - ***Invocando o stored procedure*** *PurchaseOrderInformation*

```
USE MyAdventureWorks;

EXEC dbo.PurchaseOrderInformation;
```

	PurchaseOrderID	PurchaseOrderDetailID	OrderDate	TotalDue	ReceivedQty	ProductName
1	1	1	2005-05-17 00:00:00.000	222,1492	3.00	Adjustable Race
2	2	2	2005-05-17 00:00:00.000	300,6721	3.00	Thin-Jam Hex Nut 9
3	2	3	2005-05-17 00:00:00.000	300,6721	3.00	Thin-Jam Hex Nut 10
4	3	4	2005-05-17 00:00:00.000	9776,2665	550.00	Seat Post
5	4	5	2005-05-17 00:00:00.000	189,0395	2.00	Headset Ball Bearings
6	5	6	2005-05-31 00:00:00.000	22539,0165	550.00	HL Road Rim
7	6	7	2005-05-31 00:00:00.000	16164,0229	468.00	Touring Rim
8	7	8	2005-05-31 00:00:00.000	64847,5328	550.00	LL Crankarm
9	7	9	2005-05-31 00:00:00.000	64847,5328	550.00	ML Crankarm
10	7	10	2005-05-31 00:00:00.000	64847,5328	550.00	HL Crankarm
11	8	11	2005-05-31 00:00:00.000	766,1827	3.00	External Lock Washer 3
12	8	12	2005-05-31 00:00:00.000	766,1827	3.00	External Lock Washer 4
13	8	13	2005-05-31 00:00:00.000	766,1827	3.00	External Lock Washer 9
14	8	14	2005-05-31 00:00:00.000	766,1827	3.00	External Lock Washer 5
15	8	15	2005-05-31 00:00:00.000	766,1827	3.00	External Lock Washer 7
16	9	16	2006-01-14 00:00:00.000	767,0528	3.00	Thin-Jam Lock Nut 9
17	9	17	2006-01-14 00:00:00.000	767,0528	3.00	Thin-Jam Lock Nut 10
18	9	18	2006-01-14 00:00:00.000	767,0528	3.00	Thin-Jam Lock Nut 1
19	9	19	2006-01-14 00:00:00.000	767,0528	3.00	Thin-Jam Lock Nut 2

Procedures

- ***Palavra-chave EXECUTE***
  - ***Exemplo (03): - Parte 1***
    - ***Redefinindo** o **conjunto** de **resultados** usando **EXEC***

```
USE MyAdventureWorks;

EXEC dbo.PurchaseOrderInformation
WITH RESULT SETS
(
  (
        [Purchase Order ID] INT,
        [Purchase Order Detail ID] INT,
        [Order Date] DATETIME,
        [Total Due] MONEY,
        [Received Quantity] FLOAT,
        [Product Name] VARCHAR(50)
  )
)
```

Procedures

- ***Palavra-chave EXECUTE***
  - ***Exemplo (03): - Parte 2***
    - ***Redefinindo** o **conjunto** de **resultados** usando **EXEC***

	Purchase Order ID	Purchase Order Detail ID	Order Date	Total Due	Received Quantity	Product Name
1	1	1	2005-05-17 00:00:00.000	222,1492	3	Adjustable Race
2	2	2	2005-05-17 00:00:00.000	300,6721	3	Thin-Jam Hex Nut 9
3	2	3	2005-05-17 00:00:00.000	300,6721	3	Thin-Jam Hex Nut 10
4	3	4	2005-05-17 00:00:00.000	9776,2665	550	Seat Post
5	4	5	2005-05-17 00:00:00.000	189,0395	2	Headset Ball Bearings
6	5	6	2005-05-31 00:00:00.000	22539,0165	550	HL Road Rim
7	6	7	2005-05-31 00:00:00.000	16164,0229	468	Touring Rim
8	7	8	2005-05-31 00:00:00.000	64847,5328	550	LL Crankarm
9	7	9	2005-05-31 00:00:00.000	64847,5328	550	ML Crankarm
10	7	10	2005-05-31 00:00:00.000	64847,5328	550	HL Crankarm
11	8	11	2005-05-31 00:00:00.000	766,1827	3	External Lock Washer 3
12	8	12	2005-05-31 00:00:00.000	766,1827	3	External Lock Washer 4
13	8	13	2005-05-31 00:00:00.000	766,1827	3	External Lock Washer 9
14	8	14	2005-05-31 00:00:00.000	766,1827	3	External Lock Washer 5
15	8	15	2005-05-31 00:00:00.000	766,1827	3	External Lock Washer 7
16	9	16	2006-01-14 00:00:00.000	767,0528	3	Thin-Jam Lock Nut 9
17	9	17	2006-01-14 00:00:00.000	767,0528	3	Thin-Jam Lock Nut 10
18	9	18	2006-01-14 00:00:00.000	767,0528	3	Thin-Jam Lock Nut 1
19	9	19	2006-01-14 00:00:00.000	767,0528	3	Thin-Jam Lock Nut 2

Procedures

- ***Parametrize***
  - ***Semelhantes*** *às* ***funções****, os* ***stored procedures*** *podem* ***incluir parâmetros*** *como* ***parte*** *de* ***seu código***
  - *A* ***criação*** *de um* ***stored procedure*** *com* ***parâmetros*** *permite aos* ***programas chamadores*** *passar* ***valores*** *para o* ***procedimento***
  - *Os* ***parâmetros*** *dos* ***stored procedures*** *são* ***diferentes*** *dos* ***parâmetros*** *das* ***funções****, pois é possível* ***especificar*** *a* ***direção****, se é* ***um parâmetro*** *de* ***entrada*** *ou de* ***saída***
  - ***Podemos especificar*** *se o* ***parâmetro aceitará*** *um* ***valor*** *(****entrada****) ou se* ***retornará*** *um* ***valor*** *(****saída****)*

Procedures

- ***Parametrize***
  - *Os **parâmetros OUTPUT** podem ser **usados** para **atribuir** um **valor produzido** por um **stored procedure** diretamente a **uma variável** ou a **um aplicativo** no **contexto** de **execução***
  - *Os **stored procedures** podem ter **parâmetros opcionais** e **padrão**, **semelhantes** às **funções***
  - *Caso um **stored procedure** contenha um **parâmetro padrão**, ao **contrário** do que acontece com uma **função**, **não** será **obrigatório fornecer** a **palavra-chave DEFAULT***

Procedures

- ***Parametrize***
  - ***Recomenda-se especificar cada nome** de **parâmetro** e **atribuir** um **valor** a **cada um** (garante que o valor correto seja atribuído ao parâmetro apropriado)*

```
EXEC [dbo].[PurchaseOrderInformation] @parameter1 = 1,
                @parameter2 = default,
                @parameter3 = null
```

Procedures

- ***Parametrize***
  - ***Mudando** a **direção** de uma **parâmetro** para **OUTPUT**, ou seja, **em vez** de **atribuir um valor** ao **parâmetro** quando o **stored procedure** for **executado, será retornado um valor** e ele **poderá** ser **acessado** por meio **desse parâmetro***

```
USE MyAdventureWorks;
GO
-- Cria proc com param OUTPUT
CREATE PROC dbo.SampleOutput
  @Parameter2 INT OUTPUT
AS
  SELECT @Parameter2 = 10
```

É criado um stored procedure que tem um único parâmetro de saída.
O parâmetro de saída inclui a palavra-chave OUTPUT. O stored procedure contém apenas uma instrução T-SQL que atribui 10 ao parâmetro.

Procedures

- ***Parametrize***
  - ***Mudando** a **direção** de uma **parâmetro** para **OUTPUT**, ou seja, **em vez** de **atribuir um valor** ao **parâmetro** quando o **stored procedure** for **executado, será retornado um valor** e ele **poderá** ser **acessado** por meio **desse parâmetro***

```
-- Executa proc com param OUTPUT
DECLARE @HoldParameter2 INT
EXEC dbo.SampleOutput
  @HoldParameter2 OUTPUT
SELECT @HoldParameter2
```

É declarada uma variável para armazenar o valor do parâmetro de saída. A palavra-chave EXEC é usada para executar o stored procedure. Além disso, a variável declarada é especificada com a palavra-chave OUTPUT. Por último, é executada uma instrução SELECT para exibir o valor da saída.

Procedures

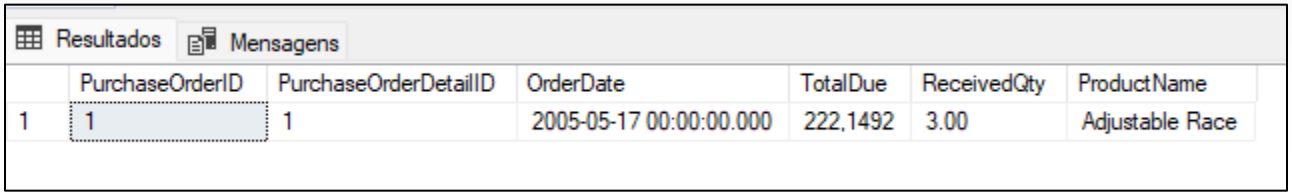
- ***Parametrize***
  - ***Exemplo (04): - Parte 1***
    - ***Alterando** o **stored procedure** para **incluir parâmetros***

```
ALTER PROCEDURE [dbo].[PurchaseOrderInformation]
  @EmployeeID INT,
  @OrderYear INT = 2005
AS
BEGIN
    SELECT poh.PurchaseOrderID, pod.PurchaseOrderDetailID,
           poh.OrderDate, poh.TotalDue, pod.ReceivedQty,
           p.Name ProductName
    FROM Purchasing.PurchaseOrderHeader poh
    INNER JOIN Purchasing.PurchaseOrderDetail pod
          ON (poh.PurchaseOrderID = pod.PurchaseOrderID)
    INNER JOIN Production.Product p ON (pod.ProductID = p.ProductID)
    WHERE poh.EmployeeID = @EmployeeID
        AND YEAR(poh.OrderDate) = @OrderYear
END
```

Procedures

- ***Parametrize***
  - ***Exemplo (04): - Parte 2***
    - ***Invocando** o **stored procedure** e, passando **apenas** um **parâmetro***

```
USE MyAdventureWorks;
EXEC [dbo].[PurchaseOrderInformation]
        @EmployeeID = 258;
```


Resultados Mensagens

	PurchaseOrderID	PurchaseOrderDetailID	OrderDate	TotalDue	ReceivedQty	ProductName
1	1	1	2005-05-17 00:00:00.000	222,1492	3.00	Adjustable Race

Procedures

- ***Parametrize***
  - ***Exemplo (04): - Parte 3***
    - *O **valor padrão** para o **parâmetro** OrderYear foi **sobrescrito** com 2006*

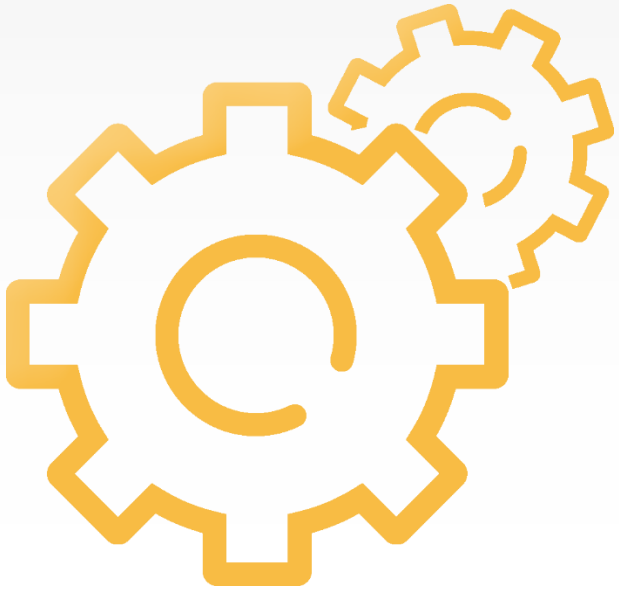
```
USE MyAdventureWorks;
EXEC [dbo].[PurchaseOrderInformation]
        @EmployeeID = 258,
        @OrderYear = 2006;
```

	PurchaseOrderID	PurchaseOrderDetailID	OrderDate	TotalDue	ReceivedQty	ProductName
1	11	24	2006-01-14 00:00:00.000	553,8221	3.00	Lock Nut 5
2	11	25	2006-01-14 00:00:00.000	553,8221	3.00	Lock Nut 6
3	11	26	2006-01-14 00:00:00.000	553,8221	3.00	Lock Nut 16
4	11	27	2006-01-14 00:00:00.000	553,8221	3.00	Lock Nut 17
5	31	77	2006-02-08 00:00:00.000	157,3647	3.00	Keyed Washer
6	41	95	2006-02-16 00:00:00.000	24880,9811	550.00	HL Mountain Seat/Saddle
7	51	116	2006-02-20 00:00:00.000	108,2513	3.00	LL Nipple
8	51	117	2006-02-20 00:00:00.000	108,2513	3.00	HL Nipple
9	61	136	2006-02-24 00:00:00.000	560,3312	3.00	External Lock Washer 3
10	61	137	2006-02-24 00:00:00.000	560,3312	3.00	External Lock Washer 4
11	61	138	2006-02-24 00:00:00.000	560,3312	3.00	External Lock Washer 9
12	61	139	2006-02-24 00:00:00.000	560,3312	3.00	External Lock Washer 5
13	71	161	2006-02-25 00:00:00.000	2214,0703	3.00	Chainring Bolts
14	71	162	2006-02-25 00:00:00.000	2214,0703	3.00	Chainring Nut
15	71	163	2006-02-25 00:00:00.000	2214,0703	60.00	Chainring

Procedures

- ***Excluindo** um **Stored Procedure***
  - ***Existem duas maneiras** de **remover** um **stored procedure** de um **BD**: com SSMS ou com T-SQL*
  - ***Sintaxe:***

```
DROP PROCEDURE nome_do_esquema.nome_do_procedimento
```



Triggers de Manipulação de Dados

Triggers

- ***Triggers*** *de* ***Manipulação*** *de* ***Dados***
  - *São* ***conjuntos*** *de* ***instruções T-SQL*** *que* ***executam*** *uma* ***ação específica***
  - *São* ***considerados*** *um* ***tipo especial*** *de* ***stored procedure***
  - *Ao* ***contrário*** *dos* ***stored procedures****, os* ***triggers*** *são* ***executados somente*** *quando um* ***usuário*** *ou* ***aplicativo*** *tenta* ***modificar dados utilizando DML***
  - ***DML*** *inclui* ***execuções*** *de* ***instruções*** ***INSERT****,* ***UPDATE*** *e* ***DELETE*** *em* ***views*** *e* ***tabelas***

Triggers

- ***Tipos** de **Triggers***
  - *Existem muitos **tipos** de **triggers:***
    - ***AFTER***
    - ***INSTEAD OF***
    - ***CLR** (**C**ommon **L**anguage **R**untime)*
  - *O **trigger** **AFTER** é **executado** **após** um **evento DML** e é a **ação padrão** para um **trigger novo**, caso **não** seja **especificado** um **tipo** (mais utilizado)*

Triggers

- ***Tipos** de **Triggers***
  - ***Suponha** que um **aplicativo execute** uma **instrução** **INSERT** que **adiciona** uma **única linha** a **uma tabela** que **contém** um **trigger AFTER***
  - ***Uma vez concluída** a **inserção**, o **código dentro** do **trigger** será **executado**, mas dentro da **mesma transação** da **inserção**, o **código dentro** do **trigger será executado**, mas **dentro** da **mesma transação** de **inserção** que **disparou** o **trigger***

Triggers

- ***Tipos** de **Triggers***
  - ***Motivo** pelo qual os **triggers** podem ser **executados** com **integridade relacional**, todavia, pode **reduzir** o **desempenho***
  - *A **transação original** **não** será **executada** nem **retrocederá** **até** que o **trigger** também seja **efetivado** ou **retroceda***
  - *Se o **INSERT** for **executado** em **uma tabela**, o **trigger** **INSTEAD OF** será **executado** no **lugar** de **INSERT**, ou seja, o **INSERT** **não** será **executado** e o **código** dentro do **trigger** **DML** será **executado** em seu lugar, como parte da **transação** que **disparou** o **trigger***

Triggers

- ***Triggers***
  - ***Sintaxe:***

```
CREATE [ OR ALTER ] TRIGGER [ nome_do_esquema. ] nome_do_trigger
ON { tabela | view }
[ WITH <opção_de_trigger_dml> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ NOT FOR REPLICATION ]
AS { instrução_sql  [ ; ] [ ,...n ] | EXTERNAL NAME
<especificador_de_método [ ; ] > }
<opção_de_trigger> ::=
    [ ENCRYPTION ]
    [ EXECUTE AS Clause ]
  <especificador_de_método> ::=
    nome_do_assembly.nome_da_classe.nome_do_método
```

Triggers

- ***Triggers***
  - ***Sintaxe:***
    - ***OR ALTER: altera** o **gatilho somente** se **ele já existir***
    - ***nome_do_esquema: nome** do **esquema** ao qual o **gatilho DML pertence***
    - ***nome_do_trigger: nome** do **gatilho***
    - ***tabela | view: a **tabela** ou a **exibição** em que o **gatilho DML** é **executado***
    - ***FOR | AFTER: especifica** que o **gatilho DML** é **disparado apenas** quando **todas** as **operações especificadas** na **instrução SQL** de **gatilho** foram **iniciadas** com **êxito***

Triggers

- ***Triggers***
  - ***Sintaxe:***
    - ***INSTEAD OF*: especifica** *que o* **gatilho DML** *será* **iniciado** *em vez da* **instrução SQL** *de* **gatilho**, **substituindo** *as* **ações** *das* **instruções** *de* **gatilho**
    - ***{ [DELETE] [,] [INSERT] [,] [UPDATE] }*: especifica** *as* **instruções** *de* **modificação** *de* **dados** *que, quando* **disparadas** *nessa* **tabela** *ou* **visão**, *ativam o* **gatilho DML**
    - ***NOT FOR REPLICATION*: indica** *que o* **gatilho não deve** *ser* **executado** *quando* **um agente** *de* **replicação modifica** *a* **tabela envolvida** *no* **gatilho**

Triggers

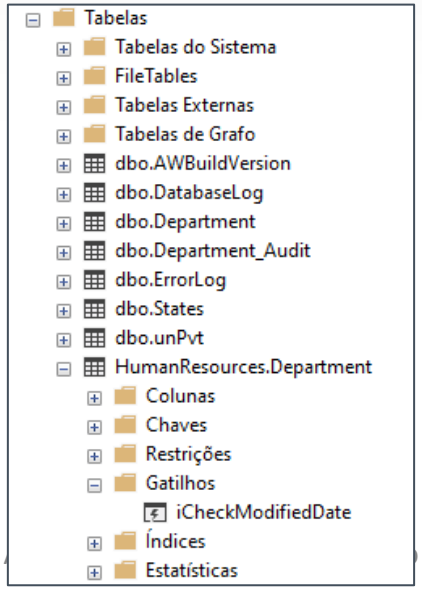
- ***Triggers***
  - ***Sintaxe:***
    - ***ENCRYPTION: ofusca** o **texto** da **instrução CREATE TRIGGER***
    - ***EXECUTE AS: especifica** o **contexto** de **segurança** no qual o **gatilho** é **executado***

Triggers

- ***Triggers***
  - ***Observação:***
    - *As **restrições**, incluindo **operações** de **foreign key** em **cascata**, são **verificadas antes** da **execução** de **qualquer tipo** de **trigger***
    - *Se for um **trigger INSTEAD OF**, a **restrição** será **verificada após** a **conclusão** do **trigger***
    - *Se for um **trigger AFTER**, a **restrição** será **verificada antes** da **execução** do **trigger***
    - ***Independentemente** do **tipo** de **trigger**, quando o **evento** do **trigger** **retroceder**, **causará** uma **violação** de **restrição***
    - *Para **triggers INSTEAD OF**, o **evento inteiro** é **desfeito** ou **retrocedido** e, no **caso** de **triggers FOR**, **nada** é **executado***

Triggers

- ***Criando** um **Trigger***
  - *Os **trigger** são **criados** em **tabelas** e **views***
  - ***Mesmo** sendo normalmente **considerados** **tipos especiais** de **stored procedures**, **não** será **possível visualizar** uma **pasta** para esse **tipo** de **objeto** na **seção** Programmability no Object Explorer para um **BD***



Triggers

- ***Criando** um **Trigger***
  - ***Exemplo (01): - Parte 1***
    - ***Criando** um **trigger** que **verifica** a **data** de **modificação**, **garantindo** que, **durante** a **inserção** de **um novo departamento**, a **data** de **modificação** seja o **dia atual**. Se **não for**, a **linha será atualizada**, **configurando** ModifiedDate com a **data** e **hora atual***

```
USE MyAdventureWorks;

CREATE OR ALTER TRIGGER HumanResources.iCheckModifiedDate
ON HumanResources.Department
 FOR INSERT
 AS
  BEGIN
   DECLARE @modifieddate DATETIME, @DepartmentID INT
   SELECT @modifieddate = modifieddate, @DepartmentID = departmentid
   FROM inserted;
-- continua
```

Triggers

- ***Criando** um **Trigger***
  - ***Exemplo (01): - Parte 2***
    - ***Criando** um **trigger** que **verifica** a **data** de **modificação**, **garantindo** que, **durante** a **inserção** de **um novo departamento**, a **data** de **modificação** seja o **dia atual**. Se **não** **for**, a **linha será atualizada**, **configurando** ModifiedDate com a **data** e **hora atual***

```
    IF (DATEDIFF(Day, @modifieddate, GETDATE()) > 0)
     BEGIN
       UPDATE HumanResources.Department
       SET ModifiedDate = GETDATE()
       WHERE DepartmentID = @DepartmentID
     END
END
```

Triggers

- ***Criando** um **Trigger***
  - ***Exemplo (01): - Parte 3***
    - ***Testando** o **funcionamento** do **trigger***

```
USE MyAdventureWorks;

INSERT INTO HumanResources.Department
VALUES
('Executive Marketing', 'Executive General and Administration',
 '02/12/2011');

SELECT *
FROM HumanResources.Department;
```

Triggers

- ***Criando** um **Trigger***
  - ***Exemplo (01): - Parte 4***
    - ***Testando** o **funcionamento** do **trigger***

Resultados | Mensagens

	DepartmentID	Name	GroupName	ModifiedDate
12	12	Document Control	Quality Assurance	2002-06-01 00:00:00.000
13	13	Quality Assurance	Quality Assurance	2002-06-01 00:00:00.000
14	14	Facilities and Maintenance	Executive General and Administration	2002-06-01 00:00:00.000
15	15	Shipping and Receiving	Inventory Management	2002-06-01 00:00:00.000
16	16	Executive	Executive General and Administration	2002-06-01 00:00:00.000
17	17	Payroll	Executive General and Administration	2012-06-12 00:00:00.000
18	18	International Marketing	Sales and Marketing	2012-05-26 00:00:00.000
19	19	International Sales Europe	Sales and Marketing	2012-05-26 00:00:00.000
20	20	Media Control	Quality Assurance	2012-05-26 00:00:00.000
21	21	International Sales USA	Sales and Marketing	2012-05-26 00:00:00.000
22	25	International Marketing Ou...	Sales and Marketing	2012-05-26 00:00:00.000
23	26	Executive Marketing	Executive General and Administration	2022-11-20 19:03:55.563

Triggers

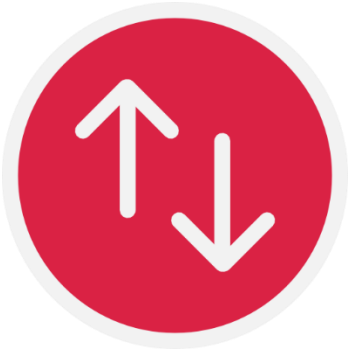
- ***Criando** um **Trigger***
  - ***Observação:***
    - *No **código** do **trigger**, especificamente a **instrução** **SELECT**, é **referenciada** uma **tabela lógica chamada** **inserted***
    - ***Existem duas tabelas** desse **tipo** (**inserted** e **deleted**) que **apenas** estão **disponíveis** no **contexto** do **trigger***
    - ***Não** é **possível modificar** a **estrutura** ou o **conteúdo** dessas **tabelas***
    - *A **tabela** **inserted** **armazena uma cópia** de **uma** ou **mais linhas** que **foram inseridas** ou **uma cópia** dos **novos valores** de **linhas** que **foram atualizadas***
    - ***Durante** uma **atualização**, a **tabela** **inserted** **armazena** os **dados novos** ou **atualizados***

Triggers

- ***Criando** um **Trigger***
  - ***Observação:***
    - *A **tabela deleted armazena** os **dados anteriores** a uma **atualização** e **uma** ou **mais linhas** que **foram excluídas** em uma **instrução DELETE***

Triggers

- ***Alterando** um **Trigger***
  - ***Assim** como **todos** os **outros objetos programáveis** no **SQL Server**, um **trigger apenas** pode ser **modificado** com **T-SQL***
  - ***Pode surgir** uma **situação** na qual **precisamos alterar** a **lógica dentro** do **trigger***
  - *Ao **invés** de **remover** e **recriar** o **trigger**, podemos usar a **palavra-chave ALTER** para **modificar** o **trigger rapidamente***



Triggers

- ***Alterando** um **Trigger***
  - ***Exemplo (02): - Parte 1***
    - ***Procedendo** com a **alteração** do **trigger** iCheckModifiedDate*

```
USE MyAdventureWorks;

CREATE OR ALTER TRIGGER HumanResources.iCheckModifiedDate
ON HumanResources.Department
 FOR INSERT
 AS
  BEGIN
   DECLARE @modifieddate DATETIME, @DepartmentID INT
   SELECT @modifieddate = modifieddate, @DepartmentID = departmentid
   FROM inserted;
-- continua
```

Triggers

- ***Alterando** um **Trigger***
  - ***Exemplo (02): - Parte 2***
    - ***Procedendo** com a **alteração** do **trigger** iCheckModifiedDate*

```
    IF (DATEDIFF(Day, @modifieddate, GETDATE()) > 0)
     BEGIN
       UPDATE HumanResources.Department
       SET ModifiedDate = DATEADD(Day, -1, GETDATE())
       WHERE DepartmentID = @DepartmentID
     END
END
```

Triggers

- ***Alterando** um **Trigger***
  - ***Exemplo (02): - Parte 3***
    - ***Testando** o **funcionamento** do **trigger***

```
USE MyAdventureWorks;

INSERT INTO HumanResources.Department
VALUES
('Executive Purchasing', 'Executive General and Administration',
 '02/12/2011');

SELECT *
FROM HumanResources.Department;
```

Triggers

- ***Alterando** um **Trigger***
  - ***Exemplo (02): - Parte 4***
    - ***Testando** o **funcionamento** do **trigger***

Resultados | Mensagens

	DepartmentID	Name	GroupName	ModifiedDate
14	14	Facilities and Maintenance	Executive General and Administration	2002-06-01 00:00:00.000
15	15	Shipping and Receiving	Inventory Management	2002-06-01 00:00:00.000
16	16	Executive	Executive General and Administration	2002-06-01 00:00:00.000
17	17	Payroll	Executive General and Administration	2012-06-12 00:00:00.000
18	18	International Marketing	Sales and Marketing	2012-05-26 00:00:00.000
19	19	International Sales Europe	Sales and Marketing	2012-05-26 00:00:00.000
20	20	Media Control	Quality Assurance	2012-05-26 00:00:00.000
21	21	International Sales USA	Sales and Marketing	2012-05-26 00:00:00.000
22	25	International Marketing Ou...	Sales and Marketing	2012-05-26 00:00:00.000
23	26	Executive Marketing	Executive General and Administration	2022-11-20 19:03:55.563
24	27	Executive Purchasing	Executive General and Administration	2022-11-19 19:37:48.417

Triggers

- ***Removendo** um **Trigger***
  - *Para **remover triggers** é **possível usar T-SQL** ou **SSMS***
  - ***Exemplo (03):***
    - ***Removendo** o **trigger** iCheckModifiedDate com **T-SQL***

```
DROP TRIGGER HumanResources.iCheckModifiedDate;
```

Triggers

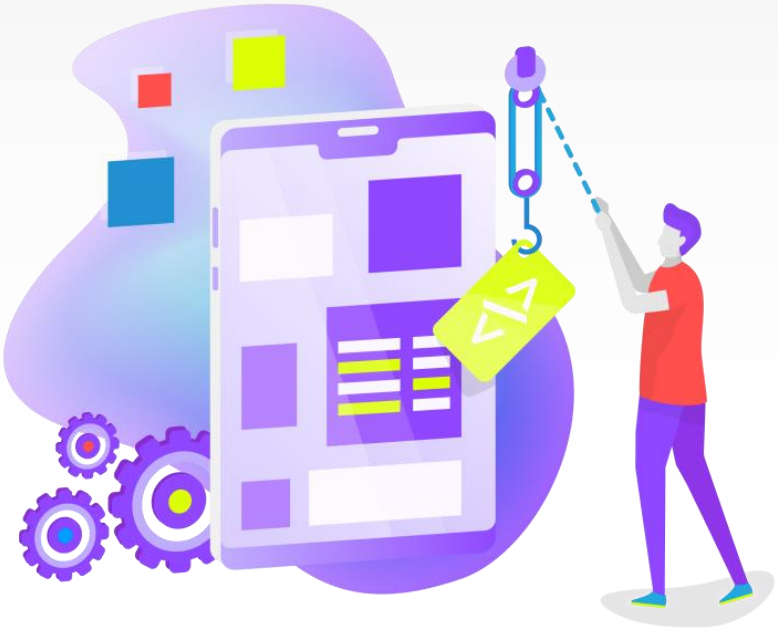
- ***Habite** e **Desabilite Triggers***
  - ***Em alguns casos, não desejamos excluir** um **trigger**, mas **impedir** sua **execução durante** uma **operação DML** grande ou para **propósitos** de **teste***
  - *O **SQL Server oferece** a **capacidade** de **desabilitar** um **trigger** e, uma vez feito isso, **podemos habilitá-lo novamente***
  - ***Exemplo (04): - Parte 1***
    - ***Desabilitando** e **habilitando** o **trigger** iCheckModifiedDate*

```
USE MyAdventureWorks;
-- Desabilita um trigger com T-SQL
DISABLE TRIGGER HumanResources.iCheckModifiedDate
   ON HumanResources.Department;
```

Triggers

- ***Habite** e **Desabilite Triggers***
  - ***Em alguns casos, não desejamos excluir** um **trigger**, mas **impedir** sua **execução durante** uma **operação DML** grande ou para **propósitos** de **teste***
  - *O **SQL Server oferece** a **capacidade** de **desabilitar** um **trigger** e, uma vez feito isso, **podemos habilitá-lo novamente***
  - ***Exemplo (04): - Parte 2***
    - ***Desabilitando** e **habilitando** o **trigger** iCheckModifiedDate*

```
USE MyAdventureWorks;
-- Habilita um trigger com T-SQL
ENABLE TRIGGER HumanResources.iCheckModifiedDate
   ON HumanResources.Department;
```



EXERCÍCIOS

Referências

Noble, E.; Pro T-SQL 2019 Toward Speed, Scalability, and Standardization for SQL Server Developers. Apress, 2020.

Ben-Gan, I.; Microsoft SQL Server 2012 T-SQL Fundamentals. Pearson Education. 2012.

Lahoud, P.; Lopes, P.; T-SQL Querying: A guide to developing efficient and elegant T-SQL code. Packt Publishing. 2019.