|  |
| --- |
|  |
| Ejemplo triggers |
|  |
| *15.AndadosIpurchase-upurchase.sql* |

Ejemplo triggers

# Ejemplo1

### Desencadenadores DML iPurchaseOrderDetail y uPurchaseOrderHeader

Cuando se insertan datos en la tabla PurchaseOrderDetail, el desencadenador DML iPurchaseOrderDetail también inserta parte de los datos insertados en la tabla Production.TransactionHistory. Puesto que varios elementos de línea de PurchaseOrderDetail forman un pedido de compra en la tabla PurchaseOrderHeader, el campo Subtotal de PurchaseOrderHeader debe actualizarse cuando se produzca una inserción en PurchaseOrderDetail. El desencadenador iPurchaseOrderDetail agrega el campo LineTotal de PurchaseOrderDetail para un determinado pedido de venta y establece el campo Subtotal de PurchaseOrderHeader con dicha agregación.

Recuerde que cuando el desencadenador DML realiza una actualización en Purchasing.PurchaseOrderHeader, provoca la activación del desencadenador de actualización uPurchaseOrderHeader que existe en la tabla Purchasing.PurchaseOrderHeader. Para obtener más información acerca de los desencadenadores que provocan la activación de otros desencadenadores, vea [Usar desencadenadores anidados](http://msdn.microsoft.com/es-es/site/ms190739).

El desencadenador DML **uPurchaseOrderDetail** lleva a cabo las mismas acciones en las tablas **TransactionHistory** y **PurchaseOrderHeader** que **iPurchaseOrderDetail**, pero lo hace cuando se actualiza la tabla **PurchaseOrderDetail** y no cuando recibe una inserción.

Los desencadenadores DML **iWorkOrder** y **uWorkOrder** de la tabla **Production.WorkOrder** sólo insertan datos en la tabla **TransactionHistory**.

# Tabla PurchaseOrderDetail (AdventureWorks)

Contiene los productos que deben comprarse con cada pedido de compra. Un pedido de compra puede incluir pedidos para varios productos. La información general, o padre, para cada pedido de compra se almacena en la tabla [PurchaseOrderHeader](http://msdn.microsoft.com/es-es/library/ms124469(v=sql.100).aspx). Cada producto pedido, o hijo, se almacena en la tabla **PurchaseOrderDetail**.

La tabla **PurchaseOrderDetail** se encuentra en el esquema **Purchasing**.

|  |  |  |  |
| --- | --- | --- | --- |
| Columna | Tipo de datos | Nulabilidad | Descripción |
| **PurchaseOrderID** | int | No NULL | Clave principal. Clave externa para **PurchaseOrderHeader.PurchaseOrderID**. |
| **PurchaseOrderDetailID** | int | No NULL | Clave principal. Número secuencial que se utiliza para garantizar la unicidad de los datos. |
| **DueDate** | datetime | No NULL | Fecha en que se espera recibir el producto del proveedor. |
| **OrderQty** | smallint | No NULL | Cantidad pedida. |
| **ProductID** | int | No NULL | Número de identificación del producto pedido. Clave externa para **Product.ProductID**. |
| **UnitPrice** | money | No NULL | Precio unitario del producto. |
| **LineTotal** | Se calcula de este modo: **OrderQty** \* **UnitPrice** | No NULL | Subtotal del costo de los productos. |
| **ReceivedQty** | decimal (8,2) | No NULL | Cantidad real recibida del proveedor. |
| **RejectedQty** | decimal (8,2) | No NULL | Cantidad rechazada durante la inspección. |
| **StockedQty** | Se calcula de este modo: **ReceiveQty** - **RejectedQty** | No NULL | Cantidad aceptada en el inventario. |
| **ModifiedDate** | datetime | No NULL | Fecha y hora de la última actualización de la fila. |

# Tabla PurchaseOrderHeader (AdventureWorks)

La tabla **PurchaseOrderHeader** se encuentra en el esquema **Purchasing**.

|  |  |  |  |
| --- | --- | --- | --- |
| Columna | Tipo de datos | Nulabilidad | Descripción |
| **PurchaseOrderID** | int | No NULL | Clave principal. |
| **RevisionNumber** | tinyint | No NULL | Número incremental para realizar el seguimiento de los cambios efectuados en el pedido de compra a lo largo del tiempo. |
| **Status** | tinyint | No NULL | Estado actual del pedido. 1 = Pendiente  2 = Aprobado  3 = Rechazado  4 = Finalizado |
| **EmployeeID** | int | No NULL | Empleado que ha creado el pedido de compra. Clave externa para **Employee.EmployeeID**. |
| **VendorID** | int | No NULL | Proveedor con el que se realiza el pedido de compra. Clave externa para **Vendor.VendorID**. |
| **ShipMethodID** | int | No NULL | Método de envío. Clave externa para **ShipMethod.ShipMethodID**. |
| **OrderDate** | datetime | No NULL | Fecha de creación del pedido de compra. |
| **ShipDate** | datetime | No NULL | Fecha estimada de envío del proveedor. |
| **SubTotal** | money | No NULL | Subtotal del pedido de compra. |
| **TaxAmt** | money | No NULL | Importe de los impuestos. |
| **Freight** | money | No NULL | Costo de envío. |
| **TotalDue** | Se calcula como **SubTotal** + **TaxAmt** + **Freight** | No NULL | Total debido al proveedor. |
| **ModifiedDate** | datetime | No NULL | Fecha y hora de la última actualización de la fila. |

# Tabla TransactionHistory (AdventureWorks)

Contiene una fila de cada transacción de pedido de compra, pedido de venta o pedido de trabajo para el año actual.

La tabla **TransactionHistory** se encuentra en el esquema **Production**.

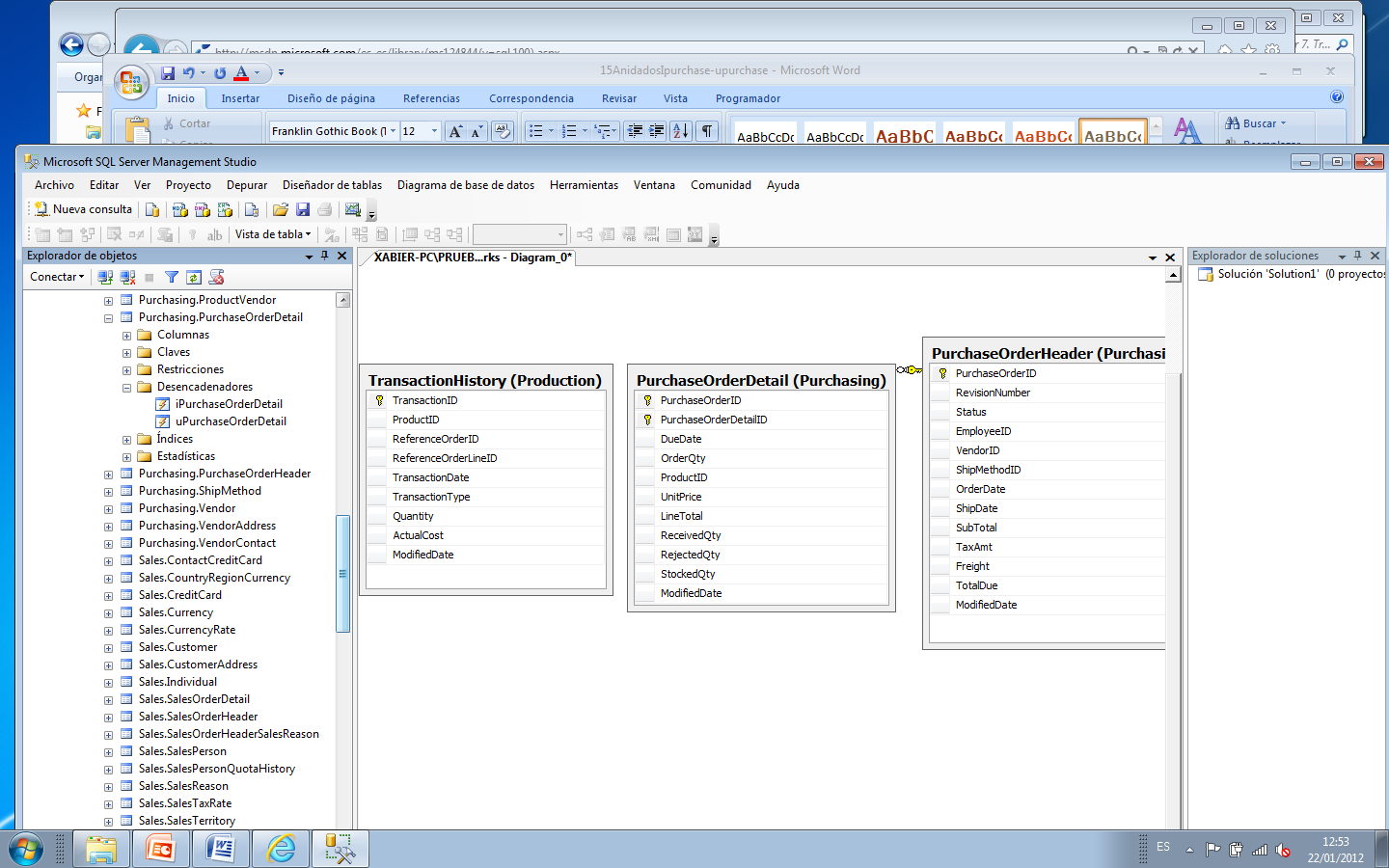
|  |  |  |  |
| --- | --- | --- | --- |
| Columna | Tipo de datos | Nulabilidad | Descripción |
| **TransactionID** | int | No NULL | Clave principal. |
| **ProductID** | int | No NULL | Número de identificación del producto. Clave externa para **Product.ProductID**. |
| **ReferenceOrderID** | int | No NULL | Número de identificación del pedido de compra, pedido de venta o pedido de trabajo. |
| **ReferenceOrderLineNumber** | int | No NULL | Número de línea asociada con el pedido de compra, pedido de venta o pedido de trabajo. |
| **TransactionDate** | datetime | No NULL | Fecha y hora de la transacción. |
| **TransactionType** | nchar(1) | No NULL | Tipo de transacción.  P = Pedido de compra  S = Pedido de venta  W = Pedido de trabajo |
| **Quantity** | int | No NULL | Cantidad de productos. |
| **ActualCost** | money | No NULL | Costo del producto. |
| **ModifiedDate** | datetime | No NULL | Fecha y hora de la última actualización de la fila. |

# Relaciones



#### Creación de los triggers

Los triggers ya están creados en la base de datos AW



#### Funcionamiento de los triggers

uPurchaseOrderDetail

Actualizar la tabla transactionHistory.

Actualizar el subtotal de la tabla PurchaseOrderHeader.

iPurchaseOrderDetail

Actualiza el número de revisión del pedido si la actualización no se ha hecho al campo status

uPurchaseOrderHeader

#### Crear el trigger iPurchaseOrderDetail

USE [AdventureWorks]

GO

IF EXISTS (SELECT \* FROM sys.triggers WHERE object\_id = OBJECT\_ID(N'[Purchasing].[iPurchaseOrderDetail]'))

DROP TRIGGER [Purchasing].[iPurchaseOrderDetail]

GO

USE [AdventureWorks]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TRIGGER [Purchasing].[iPurchaseOrderDetail] ON [Purchasing].[PurchaseOrderDetail]

AFTER INSERT AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

--actualizar la tabla transactionHistory

INSERT INTO [Production].[TransactionHistory]

([ProductID]

,[ReferenceOrderID]

,[ReferenceOrderLineID]

,[TransactionType]

,[TransactionDate]

,[Quantity]

,[ActualCost])

SELECT

inserted.[ProductID]

,inserted.[PurchaseOrderID]

,inserted.[PurchaseOrderDetailID]

,'P'

,GETDATE()

,inserted.[OrderQty]

,inserted.[UnitPrice]

FROM inserted

INNER JOIN [Purchasing].[PurchaseOrderHeader]

ON inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID];

--actualizar el subtotal de la tabla purchase order Header

UPDATE [Purchasing].[PurchaseOrderHeader]

SET [Purchasing].[PurchaseOrderHeader].[SubTotal] =

--Sumamos cada una de las nuevas lineas de pedidos de la tabla purchaseOrder detail

(SELECT SUM([Purchasing].[PurchaseOrderDetail].[LineTotal])

FROM [Purchasing].[PurchaseOrderDetail]

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID])

--Esta actualización se realizará solo en las líneas de nueas ordenes que

-–se hayan insertado.

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] IN (SELECT inserted.[PurchaseOrderID] FROM inserted);

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

#### Crear el trigger uPurchaseOrderDetail

IF EXISTS (SELECT \* FROM sys.triggers WHERE object\_id = OBJECT\_ID(N'[Purchasing].[uPurchaseOrderDetail]'))

DROP TRIGGER [Purchasing].[uPurchaseOrderDetail]

GO

USE [AdventureWorks]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TRIGGER [Purchasing].[uPurchaseOrderDetail] ON [Purchasing].[PurchaseOrderDetail]

AFTER UPDATE AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

IF UPDATE([ProductID]) OR UPDATE([OrderQty]) OR UPDATE([UnitPrice])

BEGIN

INSERT INTO [Production].[TransactionHistory]

([ProductID]

,[ReferenceOrderID]

,[ReferenceOrderLineID]

,[TransactionType]

,[TransactionDate]

,[Quantity]

,[ActualCost])

SELECT

inserted.[ProductID]

,inserted.[PurchaseOrderID]

,inserted.[PurchaseOrderDetailID]

,'P'

,GETDATE()

,inserted.[OrderQty]

,inserted.[UnitPrice]

FROM inserted

INNER JOIN [Purchasing].[PurchaseOrderDetail]

ON inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID];

UPDATE [Purchasing].[PurchaseOrderHeader]

SET [Purchasing].[PurchaseOrderHeader].[SubTotal] =

(SELECT SUM([Purchasing].[PurchaseOrderDetail].[LineTotal])

FROM [Purchasing].[PurchaseOrderDetail]

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID]

= [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID])

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID]

IN (SELECT inserted.[PurchaseOrderID] FROM inserted);

UPDATE [Purchasing].[PurchaseOrderDetail]

SET [Purchasing].[PurchaseOrderDetail].[ModifiedDate] = GETDATE()

FROM inserted

WHERE inserted.[PurchaseOrderID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderID]

AND inserted.[PurchaseOrderDetailID] = [Purchasing].[PurchaseOrderDetail].[PurchaseOrderDetailID];

END;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

#### Trigger encadenado uPurchaseOrderHeader de la tabla PurchaseOrderHeader

USE [AdventureWorks]

GO

IF EXISTS (SELECT \* FROM sys.triggers WHERE object\_id = OBJECT\_ID(N'[Purchasing].[uPurchaseOrderHeader]'))

DROP TRIGGER [Purchasing].[uPurchaseOrderHeader]

GO

USE [AdventureWorks]

GO

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TRIGGER [Purchasing].[uPurchaseOrderHeader] ON [Purchasing].[PurchaseOrderHeader]

AFTER UPDATE AS

BEGIN

DECLARE @Count int;

SET @Count = @@ROWCOUNT;

IF @Count = 0

RETURN;

SET NOCOUNT ON;

BEGIN TRY

IF NOT UPDATE([Status])

BEGIN

--Actualiza el número de revisión del pedido

UPDATE [Purchasing].[PurchaseOrderHeader]

SET [Purchasing].[PurchaseOrderHeader].[RevisionNumber] =

[Purchasing].[PurchaseOrderHeader].[RevisionNumber] + 1

WHERE [Purchasing].[PurchaseOrderHeader].[PurchaseOrderID] IN

(SELECT inserted.[PurchaseOrderID] FROM inserted);

END;

END TRY

BEGIN CATCH

EXECUTE [dbo].[uspPrintError];

IF @@TRANCOUNT > 0

BEGIN

ROLLBACK TRANSACTION;

END

EXECUTE [dbo].[uspLogError];

END CATCH;

END;

GO

#### Test del trigger iPurchaseOrderDetail

select \* from Purchasing.PurchaseOrderDetail

where PurchaseOrderDetail.PurchaseOrderID=717

go

select \* from Purchasing.PurchaseOrderHeader

where PurchaseOrderHeader.PurchaseOrderID=717

go

INSERT INTO Purchasing.PurchaseOrderDetail

(PurchaseOrderID, ProductID, OrderQty, UnitPrice,ReceivedQty,RejectedQty,DueDate )

VALUES (717,474,3,43,3.00,0.00,'2003.10.12')

go

select \* from Purchasing.PurchaseOrderDetail

where PurchaseOrderDetail.PurchaseOrderID=717

go

select \* from Purchasing.PurchaseOrderHeader

where PurchaseOrderHeader.PurchaseOrderID=717

go

#### Resultado

#### 

#### Deshacer el ejemplo:

Para poder deshacer el ejemplo debería ser suficiente con la eliminación de las líneas de pedido insertadas esto no es así ya que no hay un trigger que quite del subtotal al borrar una línea de pedido.

#### Test del trigger uPurchaseOrderDetail

select \* from Purchasing.PurchaseOrderDetail

where PurchaseOrderDetail.PurchaseOrderID=717

go

select \* from Purchasing.PurchaseOrderHeader

where PurchaseOrderHeader.PurchaseOrderID=717

Update Purchasing.PurchaseOrderDetail

SET OrderQty=4

Where PurchaseOrderDetailID=1598

select \* from Purchasing.PurchaseOrderDetail

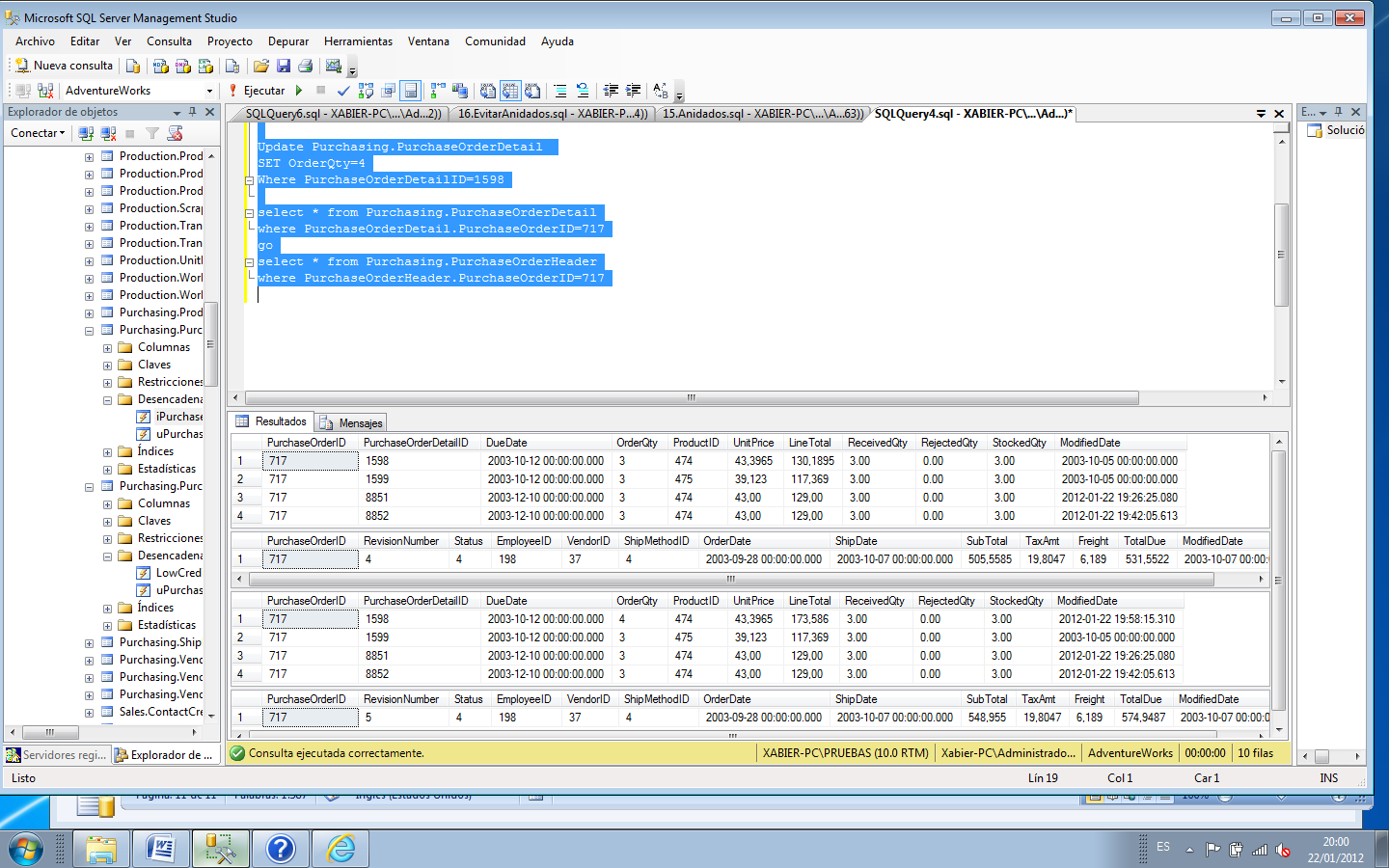
where PurchaseOrderDetail.PurchaseOrderID=717

go

select \* from Purchasing.PurchaseOrderHeader

where PurchaseOrderHeader.PurchaseOrderID=717

#### Resultado



#### Deshacer el ejemplo:

Update Purchasing.PurchaseOrderDetail

SET OrderQty=3

Where PurchaseOrderDetailID=1598

# Más información

|  |  |
| --- | --- |
| C/ Miracruz, 10 (Bº de Gros) 20001 Donostia  Telf.: 943 275819  email: [seim@centroseim.com](mailto:seim@centroseim.com) |  |

Logotipo SEIM