

Examen 2º Parcial- INF 317- Pregunta 1

Enunciado:

Mediante el uso del Agente genere una réplica de una base de datos del repositorio de Microsoft, de al menos dos tablas.

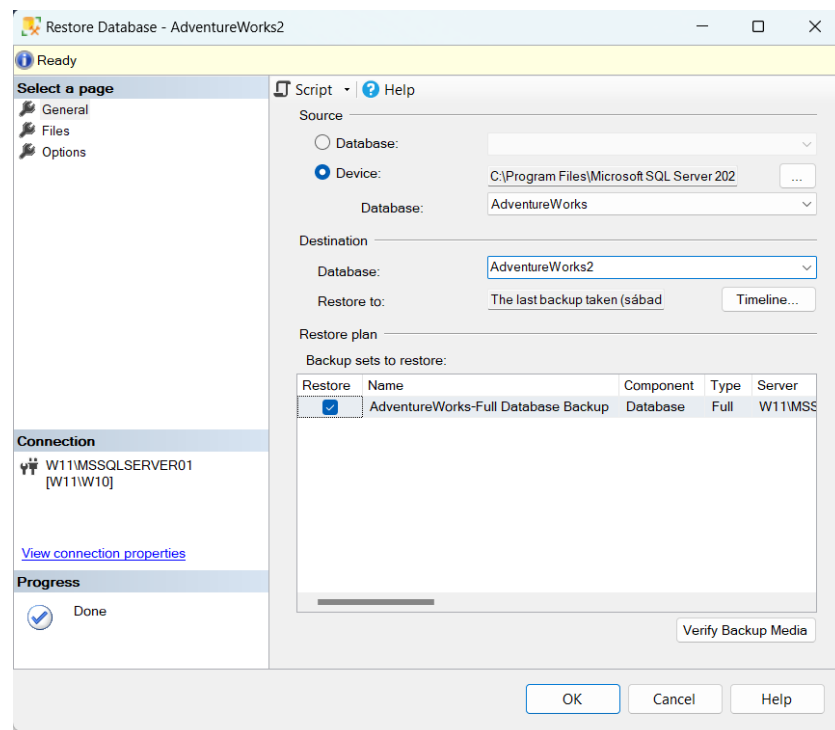
Descarga de Base de Datos y restauración

Descargamos una base de datos ejemplo del repositorio de Microsoft, en este caso Adventure Works 22:

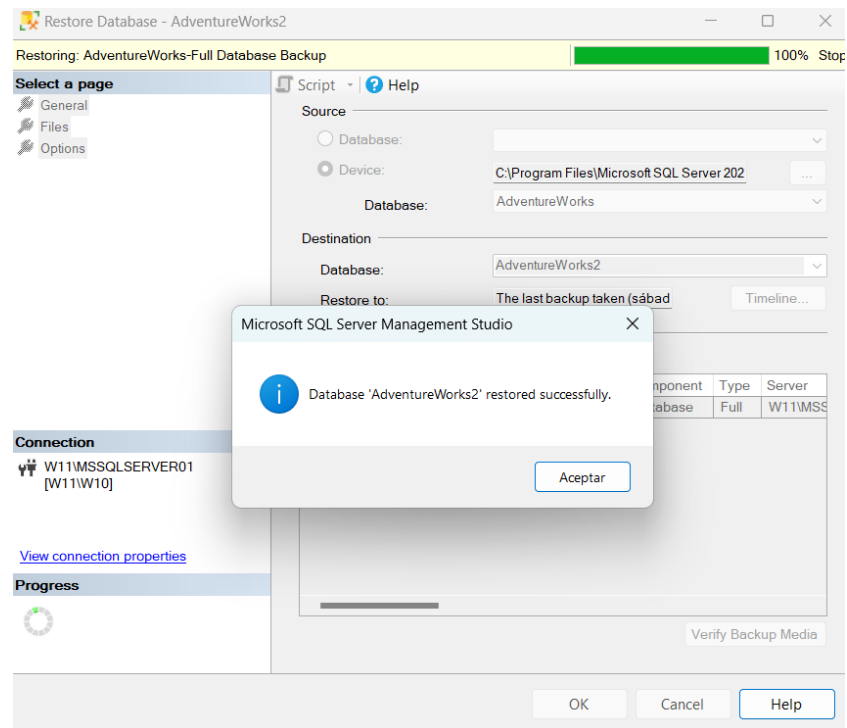
<https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms>

Restauramos el archivo .bak descargado en SQL siguiendo los pasos:

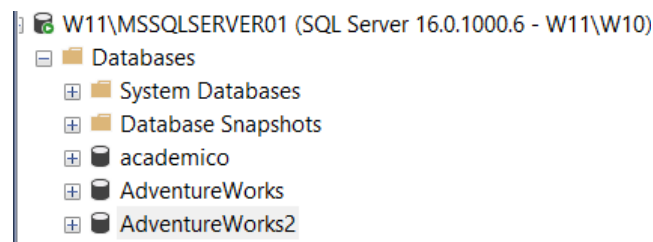
Clic derecho en el Base de datos > Restaurar Base de Datos. En la ventana seleccionar Dispositivo e ingresar a la dirección del archivo descargado anteriormente. Luego presionar OK:



Se repite el procedimiento para restaurar una segunda copia de la base de datos AdventureWorks2 que será la base replica de la base original:



Las bases restauradas son AdventureWorks y AdventureWorks2:



Donde la base de datos AdventureWorks2 replica los cambios de AdventureWorks

Configuración de agente y creación del trabajo

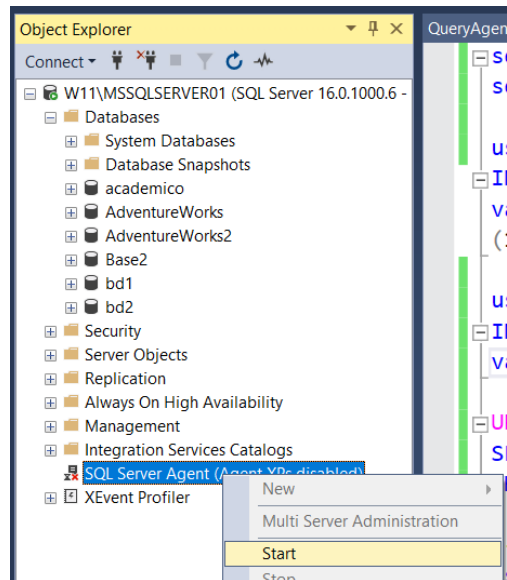
Se procede a crear un trabajo, que realizará la siguiente tarea:

Genera una réplica de la base AdventureWorks en la base AdventureWorks2, de las tablas:

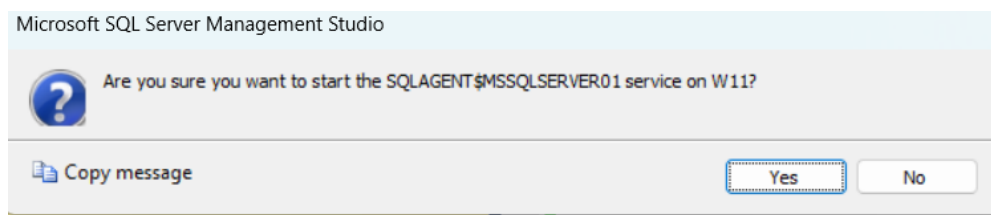
Purchasing.Vendor y **Sales.SpecialOffer**

En la base AdventureWorks se adicionan filas en caso de que la base AdventureWorks posea nuevas filas y realiza cambios en las filas que cambiaron de valor en alguno de sus campos.

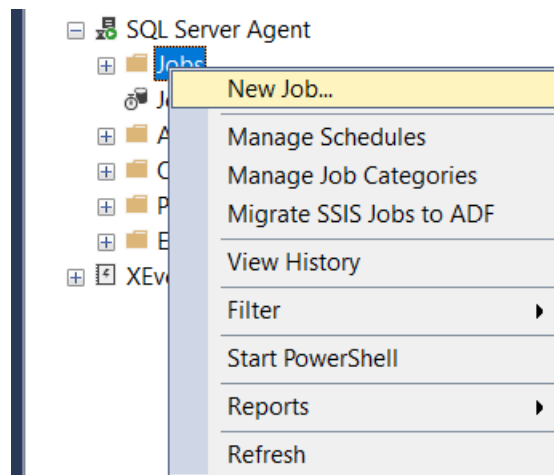
Primero inicializamos el **agente**:



En la ventana emergente presionamos Yes:



En la carpeta Jobs, hacemos clic derecho, y creamos un nuevo trabajo (New Job):



En la nueva ventana llenamos los datos de nombre y descripción:

New Job

Select a page

- General
- Steps
- Schedules
- Alerts
- Notifications
- Targets

Script Help

Name:

Owner:

Category:

Description:

Descripcion:
Genera una replica de la base AdventureWorks en la base AdventureWorks2, de las tablas: Purchasing.Vendor y Sales.SpecialOffer. En la base AdventureWorks se adicionan filas en caso de que la base AdventureWorks posea nuevas filas y realiza cambios en las filas que cambiaron de valor en alguno de sus campos.

Connection

Server:
W11\MSSQLSERVER01

Connection:
W11\W10

[View connection properties](#)

Progress

Ready

Enabled

OK Cancel

En la pestaña 'Steps' hacemos clic en new y llenamos los datos de nombre y el script:

New Job Step

Select a page

- General
- Advanced

Script Help

Step name: actualizarTablas

Type: Transact-SQL script (T-SQL)

Run as:

Database: master

Command:

Open... Select All Copy Paste Parse

```
-- Tabla Purchasing.Vendor
-- | Adicionar |
-- Adiciona filas a la base AdventureWorks2 que estan en la base original pero no estan en la
segunda
insert into AdventureWorks2.Purchasing.Vendor
select *
from AdventureWorks.Purchasing.Vendor
where BusinessEntityID not in (
    select BusinessEntityID
    from AdventureWorks2.Purchasing.Vendor
)
-- |Actualizar valores|
-- Si hay algun cambio en cualquier campos de alguna fila en la tabla publicador
-- se replican los cambios en el subscritor

declare @contador int, @limite int, @cambio int, @campos nvarchar(600),
        @idOrig int,
        @accountNumberOg nvarchar(20), @accountNum
        @nameOg nvarchar(50), @name
        @creditRatingOg int, @creditRatingI
        @preferredVendorStatusOg bit, @preferredVendorStatusI
        @activeFlagOg bit, @activeFlagI
```

Previous Next

OK Cancel

Connection

Server: W11\MSSQLSERVER01

Connection: W11\W10

[View connection properties](#)

Progress

Ready

El contenido del script de la replicación es el siguiente:

```
-- Tabla Purchasing.Vendor
-- | Adicionar |
-- Adiciona filas a la base AdventureWorks2 que estan en la base original pero no estan en la
segunda
insert into AdventureWorks2.Purchasing.Vendor
select *
from AdventureWorks.Purchasing.Vendor
where BusinessEntityID not in (
    select BusinessEntityID
    from AdventureWorks2.Purchasing.Vendor
)
-- |Actualizar valores|
-- Si hay algun cambio en cualquier campos de alguna fila en la tabla publicador
```

```
-- se replican los cambios en el subscriptor
```

```
declare @contador int, @limite int, @cambio int, @campos nvarchar(600),
        @idOrig int, @idRepl int,
        @accountNumberOg nvarchar(20), @accountNumberRe nvarchar(20),
        @nameOg nvarchar(50), @nameRe nvarchar(50),
        @creditRatingOg int, @creditRatingRe int,
        @preferredVendorStatusOg bit, @preferredVendorStatusRe bit,
        @activeFlagOg bit, @activeFlagRe bit,
        @purchasingWebServiceURLOg nvarchar(1024), @purchasingWebServiceURLRe
nvarchar(1024),
        @modifiedDateOg datetime, @modifiedDateRe datetime,
        @sql nvarchar(2000)
-- Se inicializa el contador en el ID de menor valor
select @contador = min(BusinessEntityID) from AdventureWorks.Purchasing.Vendor
-- Se inicializa el limite que es el ID de mayor valor
select @limite = max(BusinessEntityID) from AdventureWorks.Purchasing.Vendor
-- Iteramos mientras el contador no supere el limite
while @contador <= @limite
begin
    -- se selecciona la fila que tiene id igual al contador en ambas tablas
    select @idOrig = BusinessEntityID, @accountNumberOg = AccountNumber, @nameOg = Name,
           @creditRatingOg = CreditRating, @PreferredVendorStatusOg =
PreferredvndorStatus,
           @ActiveFlagOg = ActiveFlag, @purchasingWebServiceURLOg =
PurchasingWebServiceURL,
           @modifiedDateOg = ModifiedDate
    from AdventureWorks.Purchasing.Vendor where BusinessEntityID=@contador

    select @idRepl = BusinessEntityID, @accountNumberRe = AccountNumber, @nameRe = Name,
           @creditRatingRe = CreditRating, @PreferredVendorStatusRe =
PreferredvndorStatus,
           @ActiveFlagRe = ActiveFlag, @purchasingWebServiceURLRe =
PurchasingWebServiceURL,
           @modifiedDateRe = ModifiedDate
    from AdventureWorks2.Purchasing.Vendor where BusinessEntityID=@contador
    -- Se verifica si ambos id's existen en las tabla original y en el replicador
    if @idOrig = @contador and @idRepl = @contador -- se encontraron ambos registros
    begin
        set @cambio = 0
        set @campos = ''
        -- Se verifican los campos de las filas de ambas tablas, si son distintos los
cambios
        -- necesarios se guardan en la variable @campos
        if @accountNumberOg != @accountNumberRe
        begin
            set @cambio = @cambio + 1
            set @campos = @campos + ' AccountNumber = ''' + @accountNumberOg + ''' '
        end
        if @nameOg != @nameRe
        begin
            set @cambio = @cambio + 1
            if @campos != '' begin set @campos = @campos + ','end
            set @campos = @campos + ' Name = ''' + @nameOg + ''' '
        end
        if @creditRatingOg != @creditRatingRe
        begin
            set @cambio = @cambio + 1
            if @campos != '' begin set @campos = @campos + ','end
            set @campos = @campos + ' CreditRating = ''' + CAST(@creditRatingOg AS
nvarchar(20)) + ''' '
        end
    end
end
```

```

end
if @preferredVendorStatusOg != @preferredVendorStatusRe
begin
    set @cambio = @cambio + 1
    if @campos != '' begin set @campos = @campos + ','end
    set @campos = @campos + ' PreferredVendorStatus = '' +
CAST(@preferredVendorStatusOg AS nvarchar(5)) + '' '
end
if @activeFlagOg != @activeFlagRe
begin
    set @cambio = @cambio + 1
    if @campos != '' begin set @campos = @campos + ','end
    set @campos = @campos + ' ActiveFlag = '' + CAST(@activeFlagOg AS
nvarchar(5)) + '' '
end
if @purchasingWebServiceURLLog != @purchasingWebServiceURLRe
or (@purchasingWebServiceURLLog is not NULL and @purchasingWebServiceURLRe is
NULL)
begin
    set @cambio = @cambio + 1
    if @campos != '' begin set @campos = @campos + ','end
    set @campos = @campos + ' PurchasingWebServiceURL = '' +
@purchasingWebServiceURLLog + '' '
end
-- Verificación extra para el campos que puede tener valores nulos
if (@purchasingWebServiceURLLog is NULL and @purchasingWebServiceURLRe is not
null)
begin
    set @cambio = @cambio + 1
    if @campos != '' begin set @campos = @campos + ','end
    set @campos = @campos + ' PurchasingWebServiceURL = NULL'
end

if @modifiedDateOg != @modifiedDateRe
begin
    set @cambio = @cambio + 1
    if @campos != '' begin set @campos = @campos + ','end
    set @campos = @campos + ' ModifiedDate = '' + CAST(@modifiedDateOg AS
nvarchar(70)) + '' '
end
-- Si algun campos es distinto realizar los cambios en el replicador
if @cambio > 0
begin
    /*
    print 'update AdventureWorks2.Purchasing.Vendor set ' + @campos +
    ' where BusinessIdentityID = ' + CAST(@contador AS nvarchar(200))
    */
    set @sql='update AdventureWorks2.Purchasing.Vendor set ' + @campos +
    ' where BusinessEntityID = ' + CAST(@contador AS nvarchar(200))
    exec sp_executesql @sql
end
end
set @contador = @contador + 1
end
--
#####
-- Tabla Sales.SpecialOffer
-- | Adicionar |
insert into AdventureWorks2.Sales.SpecialOffer
select Description, DiscountPct, Type, Category, StartDate, EndDate, MinQty, MaxQty, RowGuid,
ModifiedDate

```

```

from AdventureWorks.Sales.SpecialOffer
where SpecialOfferID not in (
    select SpecialOfferID
    from AdventureWorks2.Sales.SpecialOffer
)
-- |Actualizar valores|
declare @contador2 int, @limite2 int, @cambio2 int, @campos2 nvarchar(600),
        @idOfferOrig int, @idOfferRepl int,
        @descriptionOg nvarchar(255), @descriptionRe nvarchar(255),
        @discountPctOg smallmoney, @discountPctRe smallmoney,
        @typeOg nvarchar(50), @typeRe nvarchar(50),
        @CategoryOg nvarchar(50), @CategoryRe nvarchar(50),
        @startDateOg datetime, @startDateRe datetime,
        @endDateOg datetime, @endDateRe datetime,
        @minQtyOg int, @minQtyRe int,
        @maxQtyOg int, @maxQtyRe int,
        @rowguidOg uniqueidentifier, @rowguidRe uniqueidentifier,
        @modifiedDateOffOg datetime, @modifiedDateOffRe datetime,
        @sql2 nvarchar(2000)
-- Se inicializa el contador2 en el ID de menor valor
select @contador2 = min(SpecialOfferID) from AdventureWorks.Sales.SpecialOffer
-- Se inicializa el limite2 que es el ID de mayor valor
select @limite2 = max(SpecialOfferID) from AdventureWorks.Sales.SpecialOffer
-- Iteramos mientras el contador2 no supere el limite2
while @contador2 <= @limite2
begin
    -- se selecciona la fila que tiene id igual al contador2 en ambas tablas
    select @idOfferOrig = SpecialOfferID, @descriptionOg = Description, @discountPctOg =
DiscountPct,
        @typeOg = Type, @CategoryOg = Category, @startDateOg = StartDate,
@endDateOg = EndDate,
        @minQtyOg = MinQty, @maxQtyOg = MaxQty, @rowguidOg = rowguid,
@modifiedDateOffOg = ModifiedDate
    from AdventureWorks.Sales.SpecialOffer where SpecialOfferID=@contador2

    select @idOfferRepl = SpecialOfferID, @descriptionRe = Description, @discountPctRe =
DiscountPct,
        @typeRe = Type, @CategoryRe = Category, @startDateRe = StartDate,
@endDateRe = EndDate,
        @minQtyRe = MinQty, @maxQtyRe = MaxQty, @rowguidRe = rowguid,
@modifiedDateOffRe = ModifiedDate
    from AdventureWorks2.Sales.SpecialOffer where SpecialOfferID=@contador2
    -- Se verifica si ambos id's existen en las tabla original y en el replicador
    if @idOfferOrig = @contador2 and @idOfferRepl = @contador2 -- se encontraron ambos
registros
    begin
        set @cambio2 = 0
        set @campos2 = ''
        -- Se verifican los campos2 de las filas de ambas tablas, si son distintos los
cambios
        -- necesarios se guardan en la variable @campos2
        if @descriptionOg != @descriptionRe
        begin
            set @cambio2 = @cambio2 + 1
            set @campos2 = @campos2 + ' Description = '' + @descriptionOg + '' '
        end
        if @discountPctOg != @discountPctRe
        begin
            set @cambio2 = @cambio2 + 1
            if @campos2 != '' begin set @campos2 = @campos2 + ', 'end

```



```

        set @campos2 = @campos2 + ' DiscountPct = ''' + CAST(@discountPctOg AS
nvarchar(20)) + ''' '
    end
    if @typeOg != @typeRe
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' Type = ''' + @typeOg + ''' '
    end
    if @categoryOg != @categoryRe
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' Category = ''' + @categoryOg + ''' '
    end
    if @startDateOg != @startDateRe
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' StartDate = ''' + CAST(@startDateOg AS
nvarchar(30)) + ''' '
    end
    if @endDateOg != @endDateRe
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' EndDate = ''' + CAST(@endDateOg AS
nvarchar(30)) + ''' '
    end
    if @minQtyOg != @minQtyRe
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' MinQty = ''' + CAST(@minQtyOg AS
nvarchar(50)) + ''' '
    end
    if @maxQtyOg != @maxQtyRe or (@maxQtyOg is not null and @maxQtyRe is null)
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' MaxQty = ''' + CAST(@maxQtyOg AS
nvarchar(50)) + ''' '
    end
    -- Verificación extra para el campos2 que puede tener valores nulos
    if (@maxQtyOg is null and @maxQtyRe is not null)
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' MaxQty = NULL '
    end
    if @rowguidOg != @rowguidRe
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end
        set @campos2 = @campos2 + ' rowguid = ''' + CAST(@rowguidOg AS
nvarchar(100)) + ''' '
    end
    if @modifiedDateOffOg != @modifiedDateOffRe
    begin
        set @cambio2 = @cambio2 + 1
        if @campos2 != '' begin set @campos2 = @campos2 + ','end

```

```
        set @campos2 = @campos2 + ' ModifiedDate = '' + CAST(@modifiedDateOff0g
AS nvarchar(50)) + '' '
    end
    -- Si algun campos2 es distinto realizar los cambios en el replicador
    if @cambio2 > 0
    begin
        /*
        print 'update AdventureWorks2.Sales.SpecialOffer set ' + @campos2 +
        ' where SpecialOfferID = ' + CAST(@contador2 AS nvarchar(200))
        */
        set @sql2='update AdventureWorks2.Sales.SpecialOffer set ' + @campos2 +
        ' where SpecialOfferID = ' + CAST(@contador2 AS nvarchar(200))
        exec sp_executesql @sql2
    end
end
set @contador2 = @contador2 + 1
end
```

En la pestaña 'Schedule' hacemos clic en new y llenamos los datos de nombre y especificamos cual será la frecuencia en la que se ejecutará el agente, que en este caso será cada día y cada 5 minutos:

New Job Schedule

Name: actualizarTablaSchedule Jobs in Schedule

Schedule type: Recurring ☒ Enabled

One-time occurrence

Date: 13/12/2023 **Time:** 11:47:57

Frequency

Occurs: Daily

Recurs every: 1 day(s)

Daily frequency

☐ Occurs once at: 0:00:00

☒ Occurs every: 5 minute(s)

Starting at: 0:00:00 **Ending at:** 23:59:59

Duration

Start date: 13/12/2023 ☐ **End date:** 13/12/2023

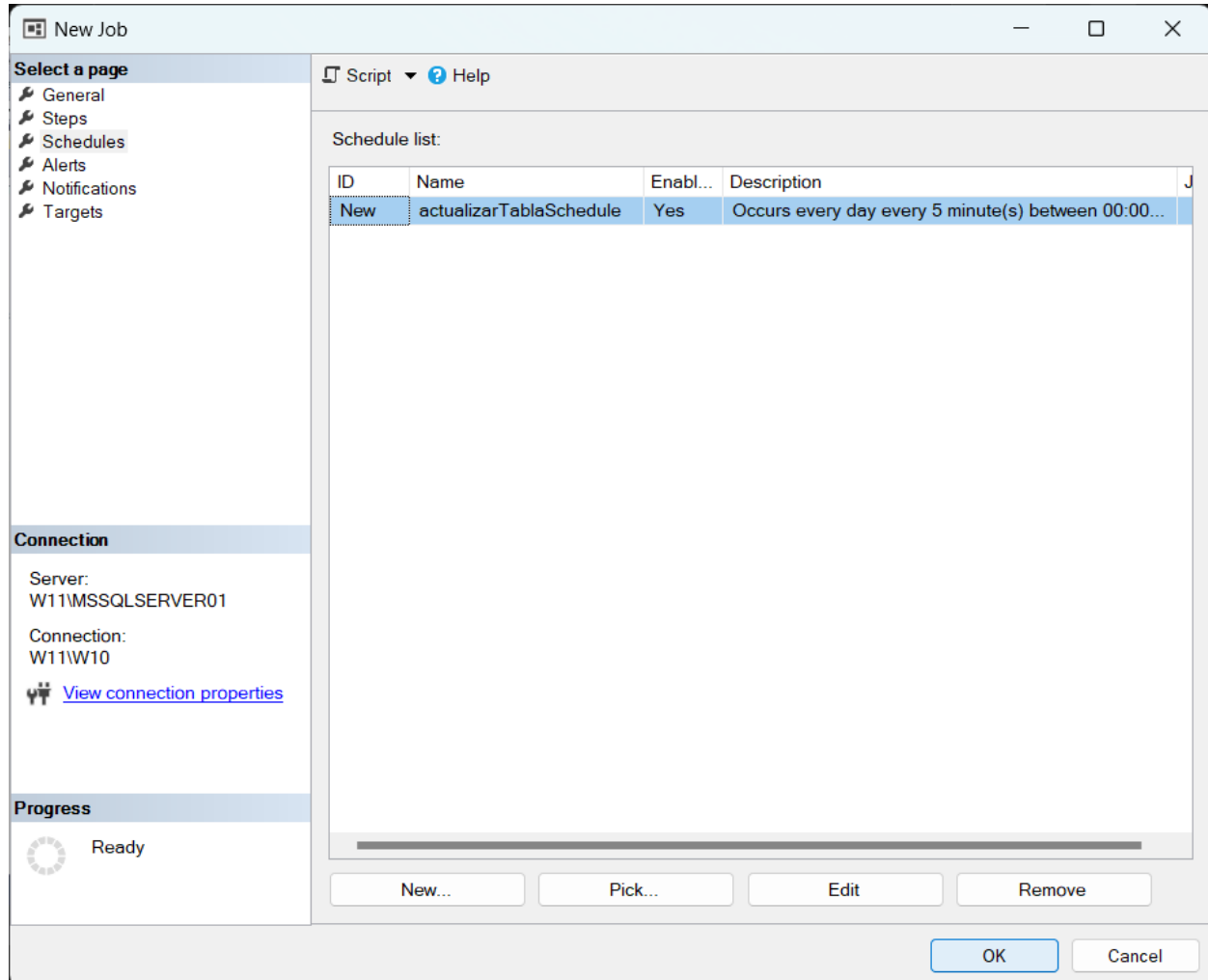
☒ No end date:

Summary

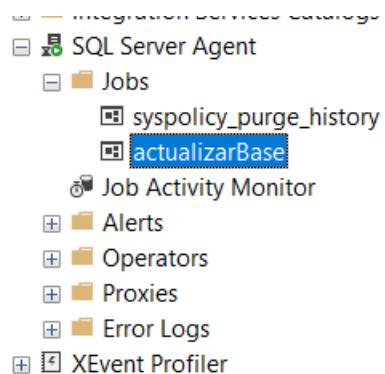
Description: Occurs every day every 5 minute(s) between 0:00:00 and 23:59:59. Schedule will be used starting on 13/12/2023.

OK Cancel Help

Presionamos OK para crear el agente:



El trabajo es creado:



Ahora cada vez que se agreguen filas a las tablas Sales.SpecialOffer o Purchasing.Vendor de la base de datos AdventureWorks, o cuando se modifiquen campos en las filas de esta base la segunda base de datos AdventureWorks2 replicará esos cambios.

Pruebas

Al hacer algunos cambios en la tabla **Purchasing.Vendor** de la base de datos AdventureWorks

```
use AdventureWorks
INSERT INTO Purchasing.Vendor
values (2001,'CARD00031','Tarjeta de credito 4', 1, 1, 1, NULL, '2023-10-12');
UPDATE AdventureWorks.Purchasing.Vendor
SET AccountNumber = 'CARD00099'
WHERE BusinessEntityID = 2000;
```

Luego de aproximadamente 5 minutos los cambios se replican en la segunda base de datos:

<pre>select * from AdventureWorks.Purchasing.Vendor select * from AdventureWorks2.Purchasing.Vendor</pre>								
133 %								
Results Messages								
	BusinessEntityID	AccountNumber	Name	CreditRating	PreferredVendorStatus	ActiveFlag	PurchasingWebServiceURL	ModifiedDate
102	1694	COMPETE0002	Compete, Inc.	1	1	1	NULL	2011-12-24 00:00:00.000
103	1696	CHICAGO0001	Chicago City Saddles	1	1	1	NULL	2011-12-24 00:00:00.000
104	1698	BUSINESS0001	Business Equipment Center	2	1	1	NULL	2011-12-23 00:00:00.000
105	1990	CARD00001	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
106	1991	CARD00002	Tarjeta de credito 2	2	1	1	NULL	2023-10-13 00:00:00.000
107	1995	CARD00077	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
108	1996	CARD00023	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
109	1997	CARD00024	Tarjeta de credito 2	2	1	1	NULL	2023-10-13 00:00:00.000
110	1998	CARD00028	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
111	1999	CARD00029	Tarjeta de credito 2	2	1	1	NULL	2023-10-13 00:00:00.000
112	2000	CARD00099	Tarjeta de credito 4	1	1	1	NULL	2023-10-12 00:00:00.000
113	2001	CARD00031	Tarjeta de credito 4	1	1	1	NULL	2023-10-12 00:00:00.000
96	1682	PREMIER0001	Premier Sport, Inc.	1	1	1	NULL	2012-02-02 00:00:00.000
97	1684	PROFESSIO0001	Professional Athletic Consul...	1	1	1	NULL	2012-02-02 00:00:00.000
98	1686	PROSPOR0001	Pro Sport Industries	1	0	1	NULL	2012-02-02 00:00:00.000
99	1688	WOODFIT0001	Wood Fitness	1	1	1	NULL	2012-02-18 00:00:00.000
100	1690	BLOOMING0001	Bloomington Multisport	1	1	1	NULL	2011-12-23 00:00:00.000
101	1692	CARLSON0001	Carlson Specialties	2	1	1	NULL	2011-12-24 00:00:00.000
102	1694	COMPETE0002	Compete, Inc.	1	1	1	NULL	2011-12-24 00:00:00.000
103	1696	CHICAGO0001	Chicago City Saddles	1	1	1	NULL	2011-12-24 00:00:00.000
104	1698	BUSINESS0001	Business Equipment Center	2	1	1	NULL	2011-12-23 00:00:00.000
105	1990	CARD00001	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
106	1991	CARD00002	Tarjeta de credito 2	2	1	1	NULL	2023-10-13 00:00:00.000
107	1995	CARD00077	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
108	1996	CARD00023	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
109	1997	CARD00024	Tarjeta de credito 2	2	1	1	NULL	2023-10-13 00:00:00.000
110	1998	CARD00028	Tarjeta de credito 1	1	1	1	NULL	2023-10-12 00:00:00.000
111	1999	CARD00029	Tarjeta de credito 2	2	1	1	NULL	2023-10-13 00:00:00.000
112	2000	CARD00099	Tarjeta de credito 4	1	1	1	NULL	2023-10-12 00:00:00.000
113	2001	CARD00031	Tarjeta de credito 4	1	1	1	NULL	2023-10-12 00:00:00.000