SEQUENCE in SQL Server 2012

#### di [Sergio Govoni](http://mvp.microsoft.com/profiles/Sergio.Govoni) - Microsoft MVP ([blog](http://www.ugiss.org/sgovoni/))

* 1. 

*Ottobre, 2012*

Introduzione

Questo articolo fornisce una overview dell’oggetto SEQUENCE che implementato con SQL Server 2012 è in grado di generare una sequenza di valori numerici in funzione delle specifiche definite in fase di creazione. A differenza di una colonna di tipo IDENTITY, un oggetto SEQUENCE **non è associato** ad alcuna tabella; le eventuali relazioni con una o più tabelle del DB dovranno essere eventualmente gestite dall’applicazione.

Utilizzi Tipici

Un oggetto SEQUENCE può essere utilizzato in sostituzione di una colonna IDENTITY, ad esempio, quando:

* + L’applicazione richiede un numero (valore) prima che un comando di INSERT sia eseguito
  + L’applicazione richiede che sia condivisa una singola sequenza di valori numerici tra più tabelle o tra più colonne all’interno della stessa tabella
  + L’applicazione deve resettare la sequenza (valore corrente) quando si raggiunge un determinato numero. Ad esempio, dopo aver assegnato ad ogni nuova riga di un ordine di vendita valori compresi tra 1 e 20, le (eventuali) successive righe inserite dovranno ricevere ancora valori numerici compresi tra 1 e 20, e così via…
  + L’applicazione richiede una sequenza di valori ordinati in funzione di un’altra colonna

Limitazioni

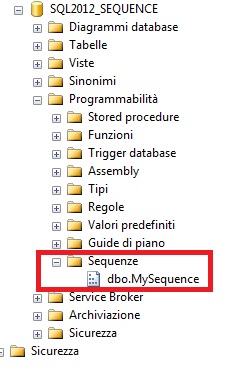
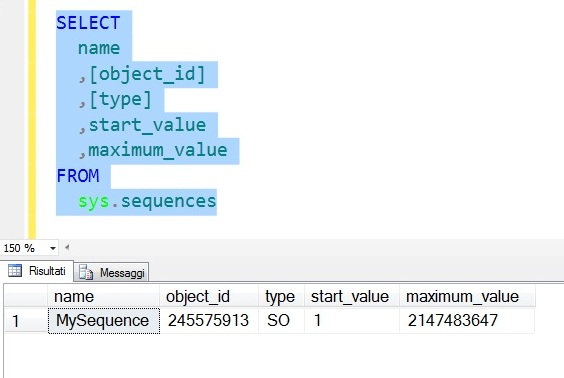
* + Diversamente da una colonna IDENTITY, i cui valori tipicamente non possono essere modificati; i valori di una sequenza non sono protetti by design da eventuali UPDATE eseguiti ad esempio dopo aver inserito tali valori in un tabella
  + Non ci sono vincoli automatici di univocità sui valori di una sequenza
  + Gli oggetti di tipo SEQUENCE, dopo aver generato i valori coerentemente con la loro definizione, non sono in grado di controllare come tali valori saranno utilizzati
  + I valori di una sequenza, inseriti ad esempio in una tabella, possono non essere contigui per effetto di eventuali ROLLBACK

L’oggetto SEQUENCE

Per creare un oggetto SEQUENCE è necessario utilizzare il nuovo statement [CREATE SEQUENCE](http://msdn.microsoft.com/en-us/library/ff878091(v=SQL.110).aspx) come riportato nel seguente frammento di codice in linguaggio T-SQL:

* 1. USE [master];
  2. GO
  3. IF EXISTS(SELECT \* FROM sys.databases WHERE NAME = 'SQL2012\_SEQUENCE')
  4. BEGIN
  5. ALTER DATABASE [SQL2012\_SEQUENCE]
  6. SET SINGLE\_USER WITH ROLLBACK IMMEDIATE;
  7. DROP DATABASE [SSDENALI\_SEQUENCE];
  8. END
  9. GO
  10. -- Create database
  11. CREATE DATABASE [SQL2012\_SEQUENCE];
  12. GO
  13. -- Change database context
  14. USE [SQL2012\_SEQUENCE];
  15. -- Create new SEQUENCE object (drop if exists)
  16. IF OBJECT\_ID('dbo.MySequence', 'SO') IS NOT NULL
  17. DROP SEQUENCE dbo.MySequence;
  18. GO
  19. CREATE SEQUENCE dbo.MySequence
  20. AS
  21. INT
  22. MINVALUE 1
  23. NO MAXVALUE
  24. START WITH 1;
  25. GO

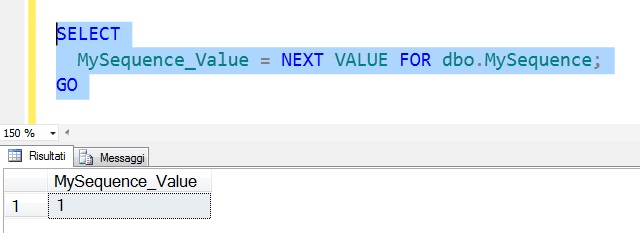
L’oggetto dbo.MySequence, appena creato sul database di esempio SQL2012\_SEQUENCE, è consultabile sia attraverso l’Object Explorer di SQL Server Management Studio (SSMS), sia attraverso la vista di sistema [sys.sequences](http://msdn.microsoft.com/en-us/library/ff877934.aspx), come illustrato rispettivamente in figura 1 e in figura 2.

* 1. 
  2. Figura 1 – Oggetto dbo.MySequence in SSMS (Object Explorer)
  3. 
  4. Figura 2 – Query sulla vista di sistema sys.sequences

Il seguente frammento di codice T-SQL rappresenta un esempio di utilizzo della funzione [NEXT VALUE FOR](http://msdn.microsoft.com/en-us/library/ff878370(v=SQL.110).aspx) per l’oggetto SEQUENCE, in particolare il comando restituirà il prossimo valore delle sequenza numerica dbo.MySequence, che sarà uguale a uno in quanto questa occorrenza rappresenta la prima chiamata alla funzione NEXT VALUE FOR per la sequenza dbo.MySequence:

* 1. SELECT
  2. MySequence\_Value = NEXT VALUE FOR dbo.MySequence;
  3. GO

L’output è illustrato in figura 3:

* 1. 
  2. Figura 3 – Utilizzo della funzione NEXT VALUE FOR sulla sequenza dbo.MySequence

Osserviamo che per creare ed utilizzare l’oggetto dbo.MySequence non è stato necessario disporre di una tabella, non ci siamo neppure preoccupati di salvare su disco le informazioni relative alla sequenza (al momento, il database di esempio SQL2012\_SEQUENCE non contiene user-table).

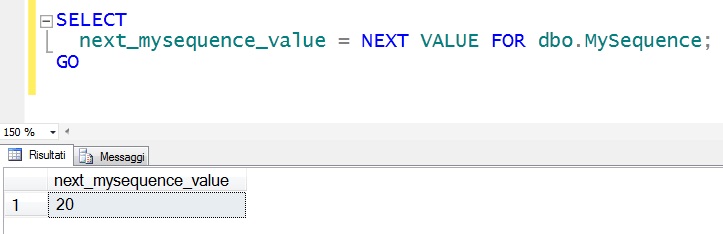
Un problema tipico delle colonne IDENTITY è quello che si riferisce al reset dei valori della sequenza. Per resettare i valori di una sequenza numerica di un oggetto SEQUENCE si potrà utilizzare l’opzione RESTART WITH come riportata il seguente comando T-SQL:

* 1. ALTER SEQUENCE dbo.MySequence
  2. RESTART WITH 20;

I prossimi valori restituiti, da dbo.MySequence, partiranno dal valore 20:

* 1. SELECT
  2. next\_mysequence\_value = NEXT VALUE FOR dbo.MySequence;
  3. GO

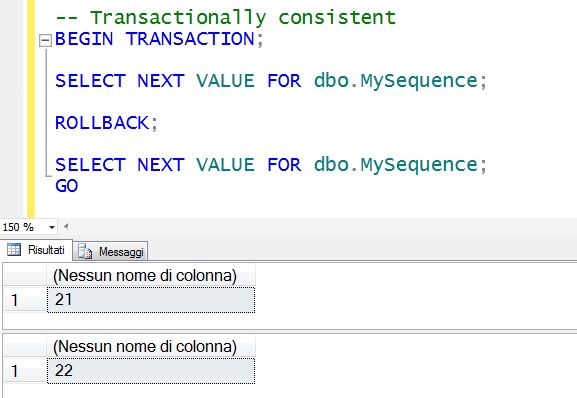
L’output è illustrato in figura 4:

* 1. 
  2. Figura 4 – Valori restituiti dalla sequenza dbo.MySequence dopo il comando RESTART WITH

Ritengo sia importante segnalare che gli oggetti SEQUENCE non assicurano la transazionalità dei valori e quindi la continuità degli stessi ad esempio quando una transazione viene respinta (ROLLBACK) oppure quando più tabelle condividono valori provenienti dalla stessa sequenza. Nel prossimo esempio dimostreremo che i valori della sequenza dbo.MySequence rimangono “consumati” anche dopo il ROLLBACK del contesto transazionale in cui sono stati generati.

* 1. -- Transactionally consistent
  2. BEGIN TRANSACTION;
  3. SELECT NEXT VALUE FOR dbo.MySequence;
  4. ROLLBACK;
  5. SELECT NEXT VALUE FOR dbo.MySequence;
  6. GO

La figura 5 illustra l’output del precedente frammento di codice T-SQL.

* 1. 
  2. Figura 5 - Valori della sequenza dbo.MySequence in un contesto transazionale

Vediamo ora uno degli utilizzi più tipici per un oggetto SEQUENCE, nel prossimo esempio verrà creata la tabella dbo.Orders in cui abbiamo evitato di utilizzare la proprietà IDENTITY per la colonna OrderID che rappresenta la Primary Key della tabella. La generazione dei valori univoci, da assegnare alla colonna OrderID della tabella dbo.Orders, è gestita attraverso la sequenza dbo.SO\_OrderID.

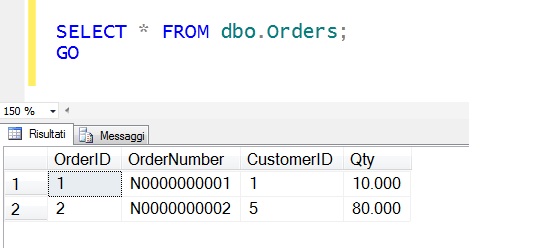
E’ importante osservare l’utilizzo della funzione NEXT VALUE FOR nello statement INSERT INTO usato per popolare la tabella.

* 1. -- Using of sequence instead of IDENTITY
  2. -- Create a table (drop if exists)
  3. IF OBJECT\_ID('dbo.Orders', 'U') IS NOT NULL
  4. DROP TABLE dbo.Orders;
  5. GO
  6. CREATE TABLE dbo.Orders
  7. (
  8. OrderID INT NOT NULL PRIMARY KEY
  9. ,OrderNumber VARCHAR(20) NOT NULL
  10. ,CustomerID INT NOT NULL
  11. ,Qty DECIMAL(8, 3) NOT NULL
  12. );
  13. GO
  14. -- Create a sequence (drop if exists)
  15. IF OBJECT\_ID('dbo.SO\_OrderID', 'SO') IS NOT NULL
  16. DROP SEQUENCE dbo.SO\_OrderID;
  17. GO
  18. CREATE SEQUENCE dbo.SO\_OrderID
  19. AS
  20. INT
  21. START WITH 1
  22. INCREMENT BY 1;
  23. GO
  24. -- Insert some records
  25. INSERT INTO dbo.Orders
  26. (
  27. OrderID
  28. ,OrderNumber
  29. ,CustomerID
  30. ,Qty
  31. )
  32. VALUES
  33. (
  34. NEXT VALUE FOR dbo.SO\_OrderID
  35. ,'N0000000001'
  36. ,1
  37. ,10
  38. ),
  39. (
  40. NEXT VALUE FOR dbo.SO\_OrderID
  41. ,'N0000000002'
  42. ,5
  43. ,80
  44. );
  45. GO

Visualizziamo ora il contenuto della tabella dbo.Orders, ci aspettiamo due record, e per ognuno ci aspettiamo che la colonna OrderID sia stata valorizzata correttamente.

* 1. -- Query the table
  2. SELECT \* FROM dbo.Orders;
  3. GO

L’output ottenuto è illustrato in figura 6.

* 1. 
  2. Figura 6 – Contenuto della tabella dbo.Orders

Conclusioni

SEQUENCE è un oggetto definito dall’utente, legato soltanto allo schema, in grado di generare una sequenza di valori numerici in accordo con la sua definizione. Attraverso questa overview è possibile valutare l’utilizzo di SEQUENCE nell’implementazione di un database SQL Server.

Per maggiori informazioni potete consultare la documentazione in linea [Sequence Numbers](http://msdn.microsoft.com/en-us/library/ff878058%28v=SQL.110%29.aspx).

Risorse correlate

Webcast in italiano: [What is Coming Next in T-SQL “Denali”](http://www.microsoft.com/italy/beit/Generic.aspx?video=What-is-Coming-Next-in-T39SQL-%22Denali%22)

#### di [Sergio Govoni](http://mvp.microsoft.com/profiles/Sergio.Govoni) - Microsoft MVP ([blog](http://www.ugiss.org/sgovoni/))

* 1. [*Altri articoli di Sergio Govoni nella Libreria*](http://sxp.microsoft.com/feeds/3.0/msdntn/TA_MSDN_ITA?contenttype=Article&author=Sergio%20Govoni) [](http://sxp.microsoft.com/feeds/3.0/msdntn/TA_MSDN_ITA?contenttype=Article&author=Sergio%20Govoni)