

# Libreria di funzioni XML per CMS

by Fabrizio Ricciarelli per Eustema S.p.A., maggio 2017

Raccolta di funzioni e stored procedures T-Sql su architetture Microsoft SQL Server 2012+ orientate al governo e alla trasformazione di strutture di dati tabellari in formato XML

# Libreria di funzioni XML per CMS

by Fabrizio Ricciarelli per Eustema S.p.A., maggio 2017

## Scopo

Il presente documento descrive brevemente, e a livello squisitamente tecnico, le caratteristiche intrinseche delle funzioni – nonché delle stored procedures – incluse nella libreria la cui destinazione ultima è il CMS della Intranet INPS (benché sia le funzioni che le SP sono state realizzate secondo uno stile il più generico, versatile e scalabile possibile affinché esse possano essere impiegate in qualunque altro progetto applicando minime variazioni e personalizzazioni).

## Elenco delle funzionalità presenti nella libreria

Gli elenchi che seguono riportano le funzioni presenti in libreria specificandone il tipo di ritorno:

- SVF = Scalar Value Function
- TVF = Table Value Function

Funzioni "primitive"		
Nome funzione	Tipo	Descrizione
<b>dbo.fnGetAllIdentityColumns</b>	TVF	ritorna tutte le proprietà della colonna IDENTITY (quando presente) della tabella specificata. Se invocata con il parametro in ingresso valorizzato a NULL, la presente funzione ritorna tutte le colonne IDENTITY di tutte le tabelle presenti nel database corrente
<b>dbo.fnGetAllTablesByColumnName</b>	TVF	ritorna l'elenco di tutte le tabelle che contengano, come nome di colonna, il valore specificato nel parametro "@columnname". se il parametro sarà valorizzato a null, verrà ritornato un resultset contenente tutte le colonne (con le relative proprietà) di tutte le tabelle presenti nel database corrente
<b>dbo.fnGetColumnDataType</b>	TVF	ritorna un resultset contenente i nomi dei campi di una tabella o di una vista; ritorna, tra i campi del resultset, anche il pattern XML; funziona su tutti i database della IntranetInps (Lavoro, Richieste, etc.)
<b>dbo.fnGetIDColumn</b>	SVF	recupero (quando presente) del primo campo IDENTITY e/o PRIMARY KEY della tabella specificata
<b>dbo.fnGetIDColumnProp</b>	TVF	ritorna una tabella a due colonne riportanti (quando presenti) sia il nome del primo campo IDENTITY e/o PRIMARY KEY della tabella specificata nonché un valore booleano specificante se il campo è di tipo IDENTITY o meno
<b>dbo.fnGetTableFields</b>	SVF	ritorna un elenco CSV con i nomi dei campi di una tabella o di una vista (in combinazione con la precedente funzione <b>dbo.fnSplit</b> , è possibile avere questo elenco in forma tabellare)
<b>dbo.fnGetCSVfieldsList</b>	SVF	ritorna una stringa, composta dinamicamente, contenente l'elenco dei campi - e il loro tipo di dato - presenti nella tabella specificata.

		l'elenco, se non diversamente specificato (quindi con il parametro "@separator" valorizzato a null) vedrà i campi separati tra loro da un punto e virgola.
<b>dbo.fnGetExplicitXmlTableFields</b>	SVF	rappresentazione in formato xml explicit di una qualsiasi tabella secondo specifici livelli di annidamento
<b>dbo.fnGetColumnInfo</b>	SVF	ritorna un resultset contenente i nomi dei campi di una tabella o di una vista; (simile alla funzione <code>dbo.fnGetColumnDataType</code> ma senza il pattern XML); funziona esclusivamente sul DB corrente
<b>dbo.fnGetMostProbableXmlTagDataType</b>	SVF	preleva tutti i tag presenti all'interno di un dato xml (che si può, come indicato negli esempi, prelevare dalla colonna xml di una tabella) e cerca di attribuirvi in modo predittivo, tramite la ricerca di tutte le colonne presenti in tutte le tabelle del database corrente, il tipo di dato più verosimile. L'analisi predittiva è applicabile in tutti quei casi in cui il dato xml sia stato generato partendo dalle colonne delle tabelle presenti nel database sul quale si sta invocando la funzione: in questa situazione è piuttosto verosimile e ricorrente, riuscire ad effettuare un reverse-engineering del dato non-tipizzato, ovvero quello corrispondente ad un tag xml *non legato ad un namespace* fortemente tipizzato. La restituzione del tipo di dato presunto può avvenire relativamente ad un singolo tag o a fronte di tutti tag trovati all'interno del dato xml: se il parametro opzionale "@tagname" viene valorizzato, il suo valore - se trovato all'interno del dato xml - sarà utilizzato per la ricerca del tipo di dato più prossimo relativamente a quel nome di colonna in tutte le tabelle del db corrente. Qualora, invece, il parametro opzionale "@tagname" venga valorizzato a null, saranno presi in esame tutti i tags individuati nel dato xml e per ciascuno di essi verrà effettuata una ricerca per tutte le colonne aventi un nome corrispondente in tutte le tabelle del db corrente
<b>dbo.fnTMPGetStructure</b>	SVF	nell'ambito del microsistema di gestione di dati tabellari temporanei (sistema "tmp"), questa funzione è preposta alla rappresentazione della struttura tabellare della tabella temporanea creata dalla stored procedure "sptmpcreate" e successivamente modificata dalla stored procedure "sptmpaddcolumns"

Funzioni di manipolazione XML		
Nome funzione	Tipo	Descrizione
<b>dbo.fnAddXmlNode</b>	SVF	inserimento di uno o più nodi xml, anche complessi, all'interno di una struttura xml preesistente. Si noti, osservando il codice che compone la presente funzione, che il posizionamento del nuovo nodo può avvenire secondo differenti modalità
<b>fnDeleteXmlNode</b>	SVF	eliminazione di uno o più nodi xml, anche complessi, presenti all'interno di una struttura xml preesistente
<b>fnReplaceXmlNodeName</b>	SVF	rimpiazzo del nome di un nodo xml, presente all'interno di una struttura xml preesistente, con la stringa specificata
<b>dbo.fnGetMonoLevelXml</b>	SVF	ritorna una stringa, composta dinamicamente, corrispondente ad una porzione di statement per la generazione di codice XML al secondo livello di annidamento (utilizzata dalla funzione <code>dbo.fnGetDualLevelXml</code> , non usufruibile a se stante)
<b>dbo.fnGetDualLevelXml</b>	SVF	ritorna una stringa, composta dinamicamente, in grado di creare la corretta sequenza di statements per la generazione di codice XML

		bi-livello con un numero pressoché infinito di aggregazioni sul secondo livello
<b>dbo.fnGetMultiLevelXml</b>	SVF	ritorna una stringa, composta dinamicamente, in grado di creare la corretta sequenza di statements per la generazione di codice XML bi-livello con sole due aggregazioni sul secondo livello
<b>dbo.fnXml2Table</b>	TVF	parsing di qualunque struttura XML, capace di ritornare una nutrita serie di informazioni, in forma tabellare, inerenti la struttura nodale e i campi (tags)
<b>dbo.fnXml2Table_type</b>	TVF	parsing di qualunque struttura xml, capace di ritornare una nutrita serie di informazioni, in forma tabellare, inerenti la struttura nodale, i campi (tags) e loro caratteristiche (datatype, datalenght) *** da rivedere e sistemare ***
<b>dbo.fnCompareXML</b>	SVF	confronto tra due colonne di tipo XML, ritornante un valore booleano 1 quando le colonne sono identiche, 0 quando differenti
<b>dbo.fnGetXQueryPatterns</b>	TVF	estrazione del pattern XQuery di un determinato campo (preventivamente mappato) dalla tabella dedicata XQUERY_RULES in base all'ID o alla descrizione
<b>dbo.fnGetXQueryResultDataTypes</b>	TVF	estrazione del tipo di dato di un determinato campo (preventivamente mappato) dalla tabella dedicata XQUERY_RULES in base all'ID o alla descrizione

### Funzioni "ausiliarie" e di varia utilità

Nome funzione	Tipo	Descrizione
<b>dbo.fnCercaNelCodice</b>	TVF	Ritorna tutti gli oggetti programmatici (stored procedures e funzioni) nel cui contenuto sia presente la stringa specificata
<b>dbo.fnCountStringOccurrences</b>	SVF	ritorna il numero di occorrenze presenti nella stringa specificata corrispondenti al pattern indicato
<b>dbo.fnGetAllFunctions</b>	TVF	ritorna l'elenco delle funzioni presenti nel db corrente, corredate del loro codice t-sql e la loro tipologia.
<b>dbo.fnLeftPart</b>	SVF	ritorna la porzione di stringa che si trova alla sinistra del delimitatore specificato
<b>dbo.fnMiddlePart</b>	SVF	ritorna la porzione di stringa che si trova in mezzo a due occorrenze del delimitatore specificato
<b>dbo.fnPurge</b>	SVF	rimozione di delimitatori e parentesi quadre dai nomi completi di tabelle, viste, stored procedures e funzioni, ritornando solo il nome dell'oggetto
<b>dbo.fnRightPart</b>	SVF	ritorna la porzione di stringa che si trova alla destra del delimitatore specificato
<b>dbo.fnSplit</b>	TVF	trasformazione di un elenco CSV (Comma Separated Values) in una tabella dove ciascuna riga contiene ciascun elemento dell'elenco
<b>dbo.fnTrimCommas</b>	TVF	rimuove qualunque punteggiatura che si trovi all'inizio e/o alla fine della stringa passata. Tra i caratteri che saranno rimossi <u>è compreso</u> il ritorno-carrello
<b>dbo.fnTrimSeparator</b>	TVF	rimuove ogni occorrenza del carattere specificato nel parametro "@separator" che si trovi all'inizio e/o alla fine della stringa passata. Tra i caratteri che saranno rimossi <b>non è compreso</b> il ritorno-carrello

### Funzioni specifiche per CMS

Nome funzione	Tipo	Descrizione
<b>dbo.fnGetBannersXml</b>	SVF	tutte le funzioni rappresentate qui sulla sinistra, estraggono la struttura xml dalle relative tabelle predisposte per recepire i dati (baners, gallerie, links, liste, news, etc). La modalità di estrazione varia a seconda della valorizzazione del parametro "@id_versione": se questo è valorizzato a null
<b>dbo.fnGetGallerieXml</b>	SVF	
<b>dbo.fnGetImmaginiInGalleriaXml</b>	SVF	
<b>dbo.fnGetInfoPaginaXml</b>	SVF	
<b>dbo.fnGetLinksInGalleriaXml</b>	SVF	

<b>dbo.fnGetLinksInListeXml</b>	SVF	i dati saranno prelevati dalle tabelle "di base" e loro eventuali ancillari; in caso contrario i dati saranno prelevati dalla tabella di versionamento "vsn_nometabella" e l'output sarà formattato in modo che esso sia aderente al modello xsd del portale statico.
<b>dbo.fnGetListeXml</b>	SVF	
<b>dbo.fnGetNewsXml</b>	SVF	
<b>dbo.fnGetTemplateCategoriaXml</b>	SVF	
<b>dbo.fnGetTestoConImmagineXml</b>	SVF	
<b>dbo.fnGetTestoDoppioXml</b>	SVF	
<b>dbo.fnGetTestoSempliceXml</b>	SVF	Fanno eccezione le funzioni il cui nome inizia per "fnGetValueFromXml...": queste funzioni sono state generate automaticamente dalla stored procedure "dbo.spgensqlcodeforxmlget" le cui istruzioni circa il richiamo ed il funzionamento sono riportate in testa alla sp medesima. Scopo principale di queste funzioni è quello di consentire l'accesso ai valori inclusi all'interno dei tag presenti in una determinata colonna xml e poter trattare questi ultimi come se fossero accessibili direttamente, ovvero come se fossero delle colonne di più alto livello (e non, quindi, incapsulate in un contenitore quale la colonna xml nella quale sono invece contenute). Funzioni di questo tipo, ovvero generate una tantum dinamicamente dalla summenzionata stored procedure, hanno a corredo un numero di parametri variabile, strettamente dipendente dal numero di colonne *non xml* componenti la tabella cui si riferiscono. Tali parametri - quasi tutti opzionali - servono quasi sempre a titolo di criterio di filtro, consentono cioè di restringere il set di risultati che la funzione dovrà ritornare: solo il primo e l'ultimo parametro della funzione (ovvero "@xmlpath" e "@separator") sono fissi e comuni a tutte le funzioni di questo tipo ("fngetvaluefromxml..."), il terzo parametro ("@idvsncontenutonews" in questo caso specifico) sarà sempre corrispondente al campo id primario della tabella. In tutte le funzioni di questo tipo, tra l'altro, non è praticamente *mai* presente una colonna di tipo xml in quanto quest'ultima, presente invece nella tabella cui la funzione si riferisce, è utilizzata internamente dalla funzione per svolgere tutte le operazioni di estrazione e manipolazione dei dati. Queste funzioni possono ritornare i dati contenuti nella colonna xml delle loro tabelle di riferimento in diversi modi: - possono essere ritornati valori singoli, relativi al contenuto di un singolo tag, avvalendosi anche (ma non obbligatoriamente) di criteri di filtro applicati alle colonne *non* xml della tabella di riferimento - possono essere ritornati interi set di dati, in forma tabellare, laddove un determinato tag si presenti più volte allo stesso livello di annidamento - valori singoli o tabellari possono essere impiegati *direttamente* all'interno di join, avvalendosi o meno dei vari criteri di filtro applicati alle colonne non xml - può essere ritornato, ovviamente, il contenuto dell'intera colonna xml così come è stata memorizzata in fase di inserimento o aggiornamento
<b>dbo.fnGetValueFromXmlContenutoNews</b>	SVF	
<b>dbo.fnGetValueFromXmlLink</b>	SVF	Altra eccezione è data dalla funzione fnGetPagineXml la quale racchiude in sé, in una sorta di wrapper, tutte le precedenti funzioni allo scopo di produrre una stringa XML riguardante un'intera Pagina del CMS. Come per le
<b>dbo.fnGetPagineXml</b>	SVF	

		precedenti funzioni, la modalità di produzione dell'XML varierà a seconda della valorizzazione del parametro "@id_versione": se questo è valorizzato a null i dati saranno prelevati dalle tabelle "di base" e loro eventuali ancillari; in caso contrario i dati saranno prelevati dalla tabella di versionamento "vsn_pagina" e l'output sarà formattato in modo che esso sia aderente al modello xsd del portale statico.
--	--	--

Stored procedures	
Nome SP	Descrizione
<b>spGenXmlFromTable</b>	ritorna, alla stessa stregua delle due "sorelle maggiori" spGetDualLevelXml e spGetMultiLevelXml il codice XML, popolato con i dati, risultante dalla trasformazione in tags di tutti i campi della tabella specificata, filtrati per il criterio indicato
<b>spGetAllXmlNodeVaues</b>	estrae, in forma tabellare, tutti i valori presenti in una struttura XML utilizzando, come nomi di colonna, i tags contenuti nella struttura medesima; dipende direttamente dalla funzione fnXml2Table
<b>spGetDualLevelXml</b>	esegue gli statements generati dalla funzione dipendente dbo.fnGetDualLevelXml; il valore ritornato sarà una tabella contenente un singolo campo il cui tipo è varchar(MAX) e il contenuto è il codice XML frutto dell'elaborazione
<b>spGetMultiLevelXml</b>	esegue gli statements generati dalla funzione dipendente dbo.fnGetMonoLevelXml; il valore ritornato sarà una tabella contenente un singolo campo il cui tipo è varchar(MAX) e il contenuto è il codice XML frutto dell'elaborazione
<b>spGetSingleNodeValue</b>	estrazione del valore popolato in prossimità del nodo XML specificato
<b>spGetXmlFieldValues</b>	ritorna, in forma tabellare, tutti i valori popolati nei corrispondenti nodi contenuti nella colonna di tipo XML presente in una tabella, in una vista, o nel resultset derivante da un'operazione di JOIN tra due oggetti (tabelle e/o viste) specificati
<b>spGetXmlStructure</b>	ritorna la struttura completa (alberatura) di tutti i nodi, a qualsiasi livello di annidamento, della colonna di tipo XML presente nella tabella/vista specificata
<b>spVSN_Link</b>	versionamento dei Links, nonché dei loro elementi ancillari, del CMS; la SP incorpora le opportune metodiche atte ad evitare la creazione di elementi duplicati (grazie all'impiego della funzione dbo.fnCompareXML)

## Alcune specifiche, parametri in ingresso e valori di ritorno

(Per le specifiche in dettaglio di tutte funzionalità espresse in questo documento si rimanda al sorgente T-SQL di ciascuna funzione/stored procedure, tutte ampiamente commentate)

Oggetto	Tipo/valore ritornato	Parametri in ingresso	Esempi di invocazione
<b>dbo.fnSplit</b>	<i>Resultset:</i> <i>Item nvarchar(4000)</i>	@List nvarchar(MAX) @Delimiter nvarchar(255) = NULL	SELECT * FROM dbo.fnSplit('Pippo, Pluto, Paperino', ',')  Risultato:

			<b>Item</b> <hr/> - Pippo Pluto Paperino
<b>dbo.fnLeftPart</b>	<i>varchar(MAX)</i>	@str varchar(MAX) @what varchar(MAX)	SELECT dbo.fnLeftPart('Intranetins_Lav oro].[dbo].[TestoConImmagine]', '.') AS LEFTPART  Risultato: LEFTPART <hr/> - [Intranetins_Lavoro]
<b>dbo.fnMiddlePart</b>	<i>varchar(MAX)</i>	@str varchar(MAX) @what varchar(MAX)	SELECT dbo.fnMiddlePart('Intranetins_L avoro].[dbo].[TestoConImmagine]', ',.') AS MIDDLEPART  Risultato: MIDDLEPART <hr/> - [dbo]
<b>dbo.fnRightPart</b>	<i>varchar(MAX)</i>	@str varchar(MAX) @what varchar(MAX)	SELECT dbo.fnRightPart('Intranetins_Lav oro].[dbo].[TestoConImmagine]', '.') AS RIGHTPART  Risultato: RIGHTPART <hr/> - [TestoConImmagine]
<b>dbo.fnPurge</b>	<i>varchar(128)</i>	@tableName varchar(MAX) = NULL	SELECT dbo.fnPurge ('Intranetins_Lavoro].[dbo].[Link ]') AS Purged  Risultato: Purged <hr/> - Link
<b>dbo.fnGetTableFields</b>	<i>varchar(MAX)</i>	@tableName varchar(128) = NULL	SELECT dbo.fnGetTableFields('IntranetIn ps_Richieste).[dbo].[VSN_NewsInP age]') AS fieldsList  Risultato: fieldsList <hr/> - Id_VsnNewsInPage,Data,XmlNews InPage,Autore,IdContenutoNews, Titolo,DataNews,NewsInPage,IdA ggregato
<b>dbo.fnGetColumnDataTy pe</b>	<i>Resultset:</i> <i>fieldName</i> <i>varchar(128)</i> <i>,fieldType</i> <i>varchar(128)</i> <i>,fieldLenght int</i> <i>,fieldPrecision int</i> <i>,fieldScale int</i> <i>,xmlPattern</i> <i>varchar(MAX)</i>	@objectName varchar(MAX) = NULL	SELECT * FROM fnGetColumnDataType('vToolNew sHomePage')  Risultato: fieldName      fieldType fieldLenght      fieldPrecision fieldScale      xmlPattern <hr/> Id_GruppoNT    int      4      10 0      Id_GruppoNT = C.value('//Id_GruppoNT[1]',int'), NomeGruppo    varchar 100    0 0      NomeGruppo =

			C.value('//NomeGruppo[1]','varchar(100)'), ID_Link int 4 10 0 ID_Link = C.value('//ID_Link[1]','int'),
<b>dbo.fnGetIDColumn</b>	<i>varchar(128)</i>	@tableName varchar(128)=NULL	SELECT dbo.fnGetIDcolumn('Pagine') AS ID  Risultato ID ----- - Id_Pagina
<b>dbo.fnGetIDColumnProp</b>	<i>Resultset: ColumnName varchar(128), IsIdentity bit</i>	@tableName varchar(128)=NULL	SELECT * FROM dbo.fnGetIDcolumnProp('Pagine')  Risultato: ColumnName IsIdentity ----- ---- Id_Pagina 0
<b>dbo.fnCountStringOccurrences</b>	<i>Int</i>	@string varchar(MAX), @charToCount char(1)	SELECT dbo.fnCountStringOccurrences('[I ntranetInps].[dbo].[KeyWord_Link] , ' ') AS Occorrenze  Risultato Occorrenze ----- - 2
<b>dbo.fnGetMonoLevelXml</b>	<i>varchar(MAX)</i>	@level2TableName varchar(128) = NULL, @level2Alias varchar(128) = NULL, @commonIDfieldName varchar(128) = NULL, @commonIDfieldValue int, @useElementTag BIT = NULL	SELECT dbo.fnGenMonoLevelXml('VX_Gru ppi', NULL 'Id_Link', 24577, 1) AS XmlField  Risultato XmlField ----- - ,(SELECT Id_GruppoNT ,NomeGruppo ,ID_Link FROM VX_Gruppi WITH(NOLOCK) WHERE Id_Link = 24577 FOR XML PATH('element'), ROOT('Gruppi'), TYPE)