

XML v prostredí SQL Servera 2008

Pálffy Juraj · Informačné technológie, Študentské práce

14.05.2010



XML je „de facto“ štandardom pre uloženie, spracovanie a vymieňanie informácií medzi aplikáciami. Práca prezentuje využitie XML z pohľadu relačných databáz. Popisuje dátový typ xml a oboznámi čitateľa s najnovšími postupami, metódami a voľbami v prostredí MS SQL Servera. Upozorňuje na dodatočné konfigurácie a prípadné chyby pri realizácii nových technológií. Znárodňuje vývoj integrácií XML od MS SQL Server 2000 až do MS SQL Server 2008.

1. Úvod

Žijeme v dobe integrácií informačných a technických systémov. Treba sa len trochu zamyslieť nad tým, aké každodenné elektronické prístroje používame. Určite nájdeme aspoň jedno multifunkčné zariadenie, ktoré nosí v sebe viacero funkcií (napr.: mobil, multifunkčná tlačiareň, osobný počítač, HiFi sústava, DVD prehrávač, atď.). Možno, že v prvom prehľade nevidíme, ale náš malý mobil, predstavuje vrchol vedecko-technického vývoja ľudstva. V ňom sa zhromažďuje úsilie vývojárov z rôznych vedeckých oblastí (bez úplnosti napr.: sociológovia, matematici, softvéroví, elektronickí, telekomunikační inžinieri, atď.). Z pohľadu softvérového inžinierstva tiež môžeme pozorovať podobnú integráciu.

Kvôli bezpečnosti a efektívnosti organizácie sa začínajú používať virtuálne servery a virtuálne klientske počítače. Ak používame elektronickú poštu z Google známu pod názvom Gmail, máme k dispozícii veľké množstvo elektronických služieb (napr.: Gmail chat, Calendar, Google Code, Google Webmaster Tools a ostatné). Pri integrácií takýchto informačných systémov sa určite stretne s XML-om (napr.: pri virtuálnych serveroch VMWare a VirtualBox, konfigurácie sú uložené do XML súboru, Gmail používa technológiu AJAX, ktorý pracuje na základe asynchronného JavaScriptu a XMLu).

Ako sme už spomenuli, hlavnou podstatou XML je podpora vzájomnej integrácie informačných systémov. Mnohé súčasné web aplikácie používajú databázové systémy, ktoré v menšej, alebo vo väčšej miere podporujú XML. XML uľahčuje prenesenie údajov medzi databázami nezávisle od platformy. Kvôli tomu, že bez spomenutia základných vlastností XMLu by sme sa nemohli zaoberať s našou problematikou, ktorou je spolupráca XML a relačných dát, v ďalšej kapitole si spravíme stručný úvod do XMLu.

2. Úvod do XML

XML bol navrhnutý tak, aby poskytol jednoduchý, rýchly a flexibilný formát textu. Pôvodne bol odvodený od jazyka SGML (Standardized Generalized Markup Language, ISO8879, v roku 1986) a od 10. Februára 1998 sa stal odporúčením W3C (World Wide Web Consortium). XML (eXtensible Markup Language) odporúčenia udržiava pracovná skupina zameraná na XML (XML Core Working Group) pod lupou W3C s poslaním vývoja spolupráce schopných web technológií. V tejto časti si ukážeme dôležité vlastnosti XML z hľadiska relačnej databázy.

2.1. Definícia XML údajov

XML údaje sa skladajú z niekoľkých typov prvkov:

- DTD (Document Type Definition) - je špeciálna štruktúra na popis deklarácií entít obsahujúca informácie o validácii štruktúry (SQL Server 2008 nepodporuje DTD validáciu namiesto toho Microsoft doporučuje využívať XML schémy, ktoré si znázorníme neskôr).
- XML elementy - sú kontajnery pre obsah typu CDATA (Charakter DATA) v XML dokumentoch. Každý XML element je párovaný, skladá sa zo začiatkových a koncových značiek, ktoré si môžu zapuzdriť ďalšie elementy alebo údaje. XML elementy poskytujú kostru pre XML dáta (napr.: <album> údaje </album>).
- XML atribúty - majú úzku súvislosť s elementmi, lebo elementy môžu obsahovať atribúty vo svojich začiatkových značkách. Atribúty poskytujú prídavný kontext, obsah a metaúdaje pre naše XML dokumenty (napr.: <album id="číslo" autor="Meno" nazov="Názov"></album>).
- XML komentáre - umožňujú pridať informácie užitočné pre človeka zaručujú dokumentáciu XML údajov (napr.: <!-- Komentár -->).
- XML-om spracované inštrukcie - ich významnou úlohou je poskytovanie metaúdajov pre programy ktoré ich spracúvajú (napr.: <?xml version="1.0" encoding="UTF-16"?>).
- XML referencie charakterov - sú konštrukcie, ktoré nám dovoľujú pridať špeciálne predeklarované znaky do XML údajov. Poznáme dva typy: character entity references (napr.: znak < = <) a numeric character references (napr.: znak < v decimálnom tvare < a v hexadecimálnom tvare <).

Podľa toho akú formu zobrazenia XML údajov používame, hovoríme o atribútovo orientovanej alebo o elementovo orientovanej forme. V nasledovnej tabuľke Tab. 1. zhrnieme výhody a nevýhody týchto dvoch reprezentačných foriem.

Tab. 1. Výhody a nevýhody atribútovo/elementovo orientovanej formy zobrazenia.

Atribútovo orientovaná
Výhody:
• zmena poradia údajov v dokumente • jednoduché vybranie elementov a dôsledok použitia ID atribútov • ľahší záznam atribútov • výhodnejšie mapovanie v databáze • dostaneme menšie dokumenty
Nevýhody:
• ťažšie transformácie kvôli adresáciám atribútov

Elementovo orientovaná
Výhody:
• pohodlnejšie transformácie
Nevýhody:
• zmena poradia údajov v dokumente • mierne ťažší záznam elementov • obťažné mapovanie databázy • dostaneme väčšie dokumenty

Z tabuľky 1. vyplýva, že atribútovo orientovanú formu zobrazenia môžeme výhodnejšie využiť pre naše XML dáta. Vo všeobecnosti platí pravidlo, že neexistuje najlepšie zobrazenie. Ale existuje pre daný problém vyhovujúce zobrazenie. Pre tých, ktorí sa chcú hlbšie oboznámiť s touto problematikou doporučujem do pozornosti slovenskú literatúru [8].

2.2. Správne naformátovaný a validný XML

XML dokumenty, ktoré spĺňajú organizáciou W3C doporučený XML dokumentový formát sú správne naformátované XML dokumenty. Správne naformátované XML dokumenty môžu obsahovať elementy, atribúty a text. Pre validáciu nášho správne naformátovaného XML dokumentu potrebujeme deskriptor dokumentu (DTD alebo XML schému).

Dokument deskriptor je nič menej ako formálne vyjadrenie pravidiel, ako sa napr. obsah a štruktúra môže (alebo musí) objaviť na zvolenom XML dokumente (napr.: v obchodnom akte „dodávateľská faktúra“ má vopred danú štruktúru a typy údajov, ktoré si určujú obsah položky). SQL Server obsahuje údajový typ xml. Ten dokáže obsluhovať dve formy XML údajov. XML údaje reprezentované ako fragment, alebo správne naformátovaný dokument. Aby XML údaje boli validné, musia splniť nasledovné kritériá:

- XML dáta musia obsahovať aspoň 1 alebo viac elementov.
- XML dáta nemôžu obsahovať viac ako 1 koreňový element.
- Každý element musí byť správne vnorený.

Štruktúra a obsah XML údajov môže byť vynútená priradeným xml inštancií k zbierke XML schém (XML Schema Collection). Zbierka XML schém obsahuje XML schémy, ktoré sú definované s W3C XML schématickým odporúčaním (W3C XML Schema Recommendation). XML schémy zabezpečujú pružný a silný prostriedok pre obmedzenie XML údajov.

Kedy by sme mali voliť XML formát namiesto iných možností? V tejto podkapitole hľadáme odpoveď na túto otázku. Oboznámime čitateľa s faktormi, ktoré pomôžu pri rozhodnutí zvoliť si XML reprezentáciu údajov. Hlavné dôvody voľby XML formátu sú:

- Nezávislosť od platformy a prenositeľnosť údajového formátu.
- Pozícia elementov poskytuje dodatočnú súvislosť (napr.: hierarchické umiestnenie údajov, kde pozícia elementov poskytuje prídavný kontext).
- Dopyty a aktualizácie údajov na základe jej štruktúr.

- Údaje sú rozptýlené (zriedkavé) alebo očakávame podstatné zmeny v ich štruktúre.
- Údaje sú polo-štruktúrované (semi-structured), napr.: nemajú fixovanú schému, predpokladané ale-bo nepravidelné zloženie, alebo sú vnorené a rôznorodé.
- Údaje predstavujú jednu obmedzenú hierarchiu.

Hlavné prostriedky SQL Servera 2008 sú určené na manipuláciu a vytvorenie dopytov. V nasledujúcich prípadoch nájdeme ich výhodné využitie:

- Manipulácia alebo zdieľanie XML údajov počas využitia SQL Server transakčných schopností.
- Použitie administratívnej funkcie SQL Servera na zálohovanie, obnovenie alebo vytvorenie kópie (replík) z XML údajov.
- Zabezpečenie údajov, aby XML údaje na strane servera boli správne naformátované a validné.
- Potreba funkčnej spolupráce XML a relačných údajov.
- Použitie výhod XQuery a XPath na strane SQL Servera.
- Optimalizácia dopytov XML údajov použitím indexov a dopytového optimalizátora SQL Servera.

XML je obzvlášť poslušná alternatíva komunikačného formátu nielen pre voľne viazané systémy, ale aj pre staršie systémy komunikujúce s modernými servermi. Poskytuje tiež dokonalý výber pre uloženie údajov v aplikáciach, ktoré požičajú sami seba pre použitie údajov v značkovacom formáte (angl. marked-up format), napr. systém pre správu dokumentov a prezentačné aplikácie. Modelovanie ad hoc polo-štruktúrovaných údajov pre konkrétny prípad je toho príkladom.

Skoro v každom odvetví sa stretávame s mnohými štandardnými vopred definovanými XML aplikáciami. Preto XML je často skvelou voľbou, najmä keď prioritou sú štandardné komunikačné formáty a zdieľanie dát. Doposiaľ sme sa oboznámili so silnými strankami využitia XML. Teraz by sme uviedli niekoľko situácií, keď XML nie je najlepším nástrojom na vykonanie danej práce:

- Údaje sú vysoko štruktúrované a husté.
- Poradie je bezvýznamné z hľadiska údajov.
- Na základe žiadnej štruktúry nevykonávame dopyty.
- Údaje sú najlepšie zobrazené použitím referencií na entity (angl. entity references).

Dodatočne, ak sa nezamýšľame o dopytovaní alebo manipulácii XML údajov na strane SQL Servera, ale len si ich chceme uchovať a opäť ich znova získať, mali by sme použiť dátové typy varchar a nvarchar namiesto dátového typu xml. Najčastejším prípadom je použitie SQL Servera ako úložiska dát pre XML údaje. Potom všetky naše XML dopyty a spracovanie sú vykonávané v strednej vrstve (angl. middle tier) alebo na aplikácii, ktorá je na strane klienta.

Podobne si uložíme XML údaje ako varchar alebo nvarchar ak si chceme ich presene uchovať v tvare kópie „písmeno po písmene“ (angl. character-for-character) alebo pre auditorské účely. To je všeobecný scenár, keď chceme ukladať dáta o digitálnych finančných transakciách. Pritom si potrebujeme udržiavať kontrolu priebehu transakcií, pričom nepotrebujeme manipulovať alebo dopytovať informácie v XML.

Dátový typ xml nie za každých okolností ukladá presnú kópiu „písmeno po písmene“ nášho XML dokumentu (COLES, 2008).

3. Čo nové prináša MS SQL Server 2008?

SQL Server 2008 poskytuje viacero vylepšení nad SQL Serverom 2000 v zmysle podpory XML a niekoľkých doplnkov na SQL Server 2005. Zatiaľ veľká časť špecifickej XML funkcionality je dozadu kompatibilná. Ale niektoré časti boli odmietnuté a vymenené za nové časti s úplne novou funkcionalitou. V nasledovných podkapitolách všeobecne popíšeme hlavné vylepšenia podpory XML.

3.1. Dátový typ xml

Dátový typ xml vznikol z SQL/XML 2003 štandardu, ktorý integruje XML s SQL. Na začiatkoch SQL Servera poskytol veľmi obmedzenú podporu pre uchovanie, manažovanie a manipuláciu XML údajov. SQL Server 2000 implementoval svoje XML schopnosti cez FOR XML klauzuly a dátové operácie typu LOB (Large Object) s kombináciou špeciálnych systémových procedúr. SQL Server 2005 predstavil dátový typ xml kvôli tomu, aby s jeho pomocou prevádzal ukladanie a manipuláciu XML údajov. Dátový typ xml zostal jedným z naj-dôležitejších charakteristických znakov XML v prostredí SQL Servera 2008. Dátový typ xml podporuje uloženie XML dokumentov a ich fragmentov, ktoré môžu byť validované s XML schémami (angl. typed XML).

Dátový typ xml možno používať na deklaráciu: stĺpcov v tabulke, T-SQL (Transact-SQL – je rozšírená verzia SQL jazyka, vyvinutá firmami Microsoft a Sybase.) premenných, parametrov a pre návratovú hodnotu funkcií. Údaje si ľahko dokážeme premeniť do xml dátového typu a tiež aj naspäť. Navyše dátový typ xml vezme so sebou súbor metód, vhodných pre dopyty, manipuláciu s XML údajmi a shredding. Shredding je rozbaľovanie XML dokumentu a jeho vloženie do jednej alebo viacerých tabuliek s použitím cudzieho kľúča na reprezentáciu relácie medzi rodičom a potomkom existujúcim v XML dokumente.

3.2. Zbierka XML schém

Zbierka XML schémy sa postará o schválenie našich dokumentov pre správny formát a platnosť (validitu). SQL Server 2008 podporuje na strane servera tvorbu zbierky XML schém. Zbierka XML schém používa XQuery/XDM (XDM – XPath Data Model) pre definíciu štruktúry a obmedzenia obsahu hodnôt v dátovom type xml (resp. v XML údajoch). XML schémy podporujú oveľa efektívnejšiu metódu validácie XML dokumentov ako od nich staršie DTD metódy. Oproti DTD pri interpretácii XML schém používame jazyk XML. Hlavné výhody XML schém voči DTD sú tieto:

- DTD nevie kontrolovať údaje pre daný element/atribút (napr.: formát textu).
- DTD určí 10 dátových typov, XML schémy 44 a možnosť vytvorenia vlastných.
- Deklarácie v DTD majú globálny charakter nezohľadňujúci kontext elementov.
- DTD používa inú syntax ako XML. Tým pridáva ďalšiu zložitosť a spomaľuje spracovanie dokumentov.

Použitím XSD (XML Schema Definition) si dokážame vytvárať zložitejšiu zbierku XML

schém.

3.3. Indexy typu XML

SQL Server XML model, v každom prípade dopytu a manipulácie XML údajov konvertuje údaje na relačnú formu v procese známom ako shredding. Tento proces môže byť časovo náročný najmä keď pracujeme s rozsiahlym XML súborom. Časovo náročné sú také prípady, keď vznikajú veľké a rozsiahle dopyty pýtajúce si od SQL Servera veľké kvantum údajov typu inšancií xml. SQL Server 2008 podporuje indexovanie stĺpcov dátového typu xml. Indexovanie xml stĺpcov napomáha SQL optimalizátorovi výrazným vylepšením vykonávania dopytov nad uloženými XML údajmi v databáze.

Zvýšenie výkonnosti dopytov sa uskutočňuje budovaním indexov našich XML údajov pri ich konvertovaní na relačný formát (angl. preshredding). Indexovanie XML v preshreddingovom procese odstráni krok shreddingu počas dopytu alebo pri manipulácii s XML údajmi. Z týchto skutočností vyplývajú omnoho rýchlejšie a menej intenzívnejšie využitie zdrojov pri XML dopytových operáciách. K T-SQL sa pridali nové DML vyhlásenia, aby spravili relatívne jednoduchý manažment indexácie XML.

3.4. FOR XML a XPath

SQL Server zahrňuje aj vylepšenú verziu FOR XML klauzuly. Vývojári Microsoftu ho tesnejšie integrovali s novým dátovým typom xml vrátane s dodatočnou voľbou pri generovaní validovaných xml (angl. xml-typed) výsledkov. K výsledkom FOR XML si môžeme priradiť premenné dátového typu xml s ďalšou podporou pre vnorenie FOR XML dopytov. Na strane SQL Servera 2000 FOR XML klauzula bola limitovaná na najvyššiu úroveň výberu SELECT. FOR XML PATH metóda prenesená z SQL Servera 2005 je zdokonalenie staršej FOR XML EXPLICIT metódy.

Spolu so štandardne zabudovanou podporou pre XPath (jazyk, ktorý slúži na adresovanie časti XML dokumentu, navrhnutý s účelom, aby XSLT a Xpointer vedeli využívať jeho výhody.) výrazy, FOR XML PATH vytvára XML v explicitných štruktúrach omnoho jednoduchšie, ako tomu bolo možné pre SQL Server 2000. FOR XML RAW metódy boli tiež rozvinuté s týmito prídavnými vlastnosťami:

- Premenenie predvoleného elementu pod menom row.
- Explicitnou špecifikáciou koreňového uzla.
- Získanie našich dát v elementovo-orientovanom formáte.

FOR XML AUTO a FOR XML EXPLICIT sú tiež vylepšené pomocou prídavných volieb a nastavení. Niektoré voľby boli celkom odmietnuté, a na ich miesto niekoľko ďalších volieb bolo pridaných k FOR XML klauzule od vzniku SQL Server 2000. Spomenieme si dve také voľby, ELEMENTS XSINIL a XMLSCHEMA. ELEMENTS XSINIL generuje NULL-ové hodnoty pre výslednú množinu (skupinu). XMLSCHEMA generuje vnútorné (angl. inline) definície XML schémy (XSD – XML Schema Definition) na našom XML rezultáte. V nasledovnej tabuľke Tab. 2. si môžete prečítať jednotlivé metódy a voľby FOR XML.

Tab. 2. Zhrnutie voľby a metódy FOR XML klauzuly. (COLES, 2008)

	FOR XML AUTO	FOR XML RAW	FOR XML PATH	FOR XML EXPLICIT
XMLDATA	X	X		X
XMLSCHEMA	X	X		
ELEMENTS XSINIL	X	X	X	
ELEMENTS ABSENT	X	X	X	
BINARY BASE64	X	X	X	X
TYPE	X	X	X	X
ROOT	X	X	X	X
(‘MenoElementu’)		X	X	

Voľba XMLDATA bola odmietnutá, preto sa odporúča namiesto nej používať voľbu XMLSCHEMA.

3.5. XQuery a podpora XML DML

Pred ukončením SQL/XML:2003 štandardu, SQL štandardizačná komunita začala pracovať na nasledujúcom vydaní, do ktorej zahrnula XQuery 1.0. V dôsledku SQL/XML:2003 štandardu vznikla podpora vnorených XQuery vo vnútri SQL funkcií. XQuery nahradí programovacie jazyky pre strednú vrstvu (middle-ware) a jazyky pre vývoj web aplikácií. XQuery nahradí zložité Java alebo C++ programy so zopár programovými riadkami. W3C vyvíjal XQuery v mene „urob viac, pracuj menej“. Technologicky XQuery 1.0 je rozšírením jazyka XPath 2.0. Tie dva jazyky majú:

- Konzistentný jazykový popis.
- Syntax (grammar).
- Spoločný dátový model (XDM – XQuery/XPath Data Model).

Navyše, ktorýkoľvek výraz vyhovujúci XPath 2.0 zaručuje vrátenie rovnakého výsledku v XQuery. Vďaka dátovému typu xml, XQuery podporuje nasledovné rozšírenia:

- Metódy query(), value(), exist(), nodes() a modify().
- XQuery dopyty.
- XML Shredding.
- Manipuláciu s našimi XML údajmi pomocou XML DML (XML Data Manipulating Language – programovacím spôsobom modifikuje XML údaje).

Následkom implementácie Xquery sa SQL Server 2008 stal mocnou podmnožinou W3C XML Query Language špecifikácií. Preto má SQL Server nasledujúce funkcie:

- Podporu XPath výrazov
- FOR-LET-WHERE-ORDER-BY-RETURN (FLWOR) výrazy
- Štandardné funkcie a operátory
- XML DML rozšírenia

Samostatne XQuery neposkytuje mechanizmus pre vykonávanie základných manipulácií s XML údajmi, ako napr.: aktualizáciu, ich vloženie a odstránenie. SQL Server poskytuje rozšírenú XQuery, ktorá nám poskytuje tieto schopnosti vo forme

XML DML. XML DML používame pomocou metódy modify() dátového typu xml. Metóda modify() používa nasledovné tri výrazy:

- insert
- delete
- replace value of

Tieto výrazy sú podobné odpovedajúcemu SQL DML výrazu.

3.6. SQLCLR

SQLCLR (SQL Common Language Runtime) jazyk vznikol pre potrebu podpory na strane servera XSLT (Extensible Stylesheet Language Transformations) a aby umožnil prístup k .NET funkcionalite. Týmto vlastnosťami sa snaží SQLCLR pokrýť dieru, ktorú T-SQL a dátový typ xml nevyplnili. Aby sme mohli využiť funkcie SQLCLR v prostredí SQL Servera 2008 treba si ho dodatočne nakonfigurovať nielen na strane servera ale aj na strane klienta. .NET pre XSLT transformácie používa menný priestor System.Xml.Xsl a dátový typ SqlXml, ktorý je analogický dátovému typu xml pre SQL Server 2008. SQLCLR používa dve verejné funkcie:

- fn_XsltTransform - pre SQL Server umožňuje použitie štýlového listu pre uložené alebo generované XML údaje. Výsledok potom preniesie do aplikačnej/používateľskej vrstvy
- p_XsltTransformToFile - uloží výsledok do výstupného súboru. Poskytuje pre SQL Server generovanie XHTML alebo iných XML súborov uložených alebo generovaných údajov

SQLCLR po prvýkrát bol implementovaný pre SQL Server 2005. Preto, pre tých ktorých zaujala táto problematika doporučujem k pozornosti literatúru Pro SQL Server 2005 Assemblies od Robin Dewson a Julian Skinner.

3.7. SQLXML

Najskôr sa v SQL Server 2000 používalo SQLXML pre integráciu podpory XML. Cieľom SQLXML bol návrh technológie, ktorý poskytne „mostík“ medzi XML a relačnými údajmi. Väčšia časť funkcionality SQLXML je ukrytá v dátových typoch xml, T-SQL a .NET. Microsoft ohlásil plán odstránenia SQLXML z inštalácií v produkčnej verzii SQL Servera 2008. Mnohé aplikácie pracujú na základe SQLXML a preto žiadajú pokračujúcu podporu tejto technológie. SQLXML funkcionalita je stále užitočná pre špecifické požiadavky napr.: vzdialený webový prístup k SQL Serveru a pre následnú manipuláciu dátami. Veľká časť technológií SQLXML bola nahradená novšími funkcionalitami a je ich rozšírením z SQL Servera 2005. Hlavnými nástrojmi SQLXML sú:

- Dopytovanie relačných údajov v XML formáte (XPath a xml)
- Manipulácia s dátami cez Updategram
- Bulk load XML údajov do relačného formátu

Z metodického hľadiska treba spomenúť, že SQLXML pracuje na základe OLEDB/COM (Object Linking and Embedding Database / Component Object Model). Preto pracuje menej efektívne ako iné riešenia napr.: .NET cez „managed wrapper classes“. Skutočnosť, že SQLXML stále ešte existuje nám pomôže pri práci so špecializovanými

relačnými dopytmi a manipuláciami pracujúcimi na základe XML-u.

3.8. LINQ

LINQ (Language-Integrated Query) je nový .NET Framework (dostupný z verzie 3.5) so svojou výbavou poskytujúci štandardný súbor operátorov pre modifikáciu, dopyt a manipuláciu dát bez ohľadu na ich zdroj. LINQ pre XML (XLinQ) poskytuje mnoho nástrojov založených na XML dopytoch a je manipulačným nástrojom, ktorý je založený na C# a Visual Basic jazykoch. XLinQ obsahuje rýchly a malý XML API (Application Programming Interface). Výhody XLinQ oproti tradičných XML technológiám zamerané na dopyty a manipulácie sú nasledovné:

- LINQ API je priamo dostupný z C# a VB
- Umožňuje nám vykonať štandardný súbor dopytových operácií pre ľubovoľnú aplikáciu podporovanú dátovým zdrojom (XML údaje, SQL databázy a .NET objekty)
- Používa známu syntax pre SQL vývojárov

Snáď najdôležitejším aspektom LINQ je to, že celkom obyčajná opytovacia syntax a sémantika môže byť použitá pre kombináciu a manipuláciu dát z viacerých zdrojov.

Pre využívanie rozšírenia jazyka C# pre LINQ potrebujeme Visual Studio 2008.

4. Záver

V našej práci sme poskytli širší prehľad o XML technológiách, ktoré nám priniesla súčasná verzia Microsoft SQL Server 2008. SQL Server 2008 nám priniesol významné obohatenie svojich schopností nad predošlými verziami. Postupnou integráciou spracovaní XML so MS SQL Serverom sa stal silným hráčom na trhu XML databáz. Ako sme videli v predchádzajúcich sekciách, Microsoft sa snaží svoju relačnú databázu úzko udržať kompatibilnú s SQL/XML štandardom¹.

Preto podľa nášho názoru môžeme predpokladať, že tomu bude tak aj v budúcnosti. Do pozornosti vývojárov by sme radi doporučili, aby sa vyhli vývoju aplikácií na báze HTTP SOAP endpoints. Hlavným dôvodom je najmä to, že Microsoft ho plánuje odmietnuť z nasledovnej SQL Server verzie (o tom hovorí chybová hláška, ktorú uvádzame nižšie) pri vytvorení novej HTTP SOAP endpoint-tu na strane SQL Servera 2008. Čitateľ sa bude môcť podrobnejšie oboznámiť s informáciami o tejto problematike rozšírenej o Javu v diplomovej práci, na ktorej pracuje prvý z autorov tohto článku.

Creating and altering SOAP endpoints will be removed in a future version of SQL Server. Avoid using this feature in new development work, and plan to modify applications that currently use it.

5. Literatúra

1. BENZ, B. - DURANT, J. R. 2003. XML Programming Bible. Indianapolis : Wiley Publishing, 2003. ISBN 0764538292
2. CLARK, J. - DEROSE, S. 1999. XML Path Language. [online]. W3C [World Wide Web Consortium], 16. november 1999. [Citované 9. marca 2009]. Dostupné na:

<http://www.w3.org/TR/xpath/>

3. COLES, M. 2008. Pro SQL Server 2008 XML. USA : Apress, 2008. ISBN-13 9781590599839
4. DYKES, L. – TITTEL, E. 2005. XML For Dummies® , 4th Edition. Indianapolis : Wiley Publishing, 2005. ISBN-13 9780764588457
5. EISENBERG, A. – MELTON, J. 2005. Advancements in SQL/XML. [online]. ACM [Association for Computing Machinery], September 2004. [Citované 27. marca 2009]. Dostupné na: <http://portal.acm.org/>. ISSN 01635808
6. Extensible Markup Language. 1996. W3C [World Wide Web Consortium], aktualizácia 5. apríla 2009 22:04:58. [Citované 6. apríl 2009]. Dostupné na: <http://www.w3.org/XML/>
7. KRISHNAPRASAD, M. – LIU, Z.H. – MANIKUTTY, A. – WARNER, J. W. – ARORA, V. 2005. Towards an industrial strength SQL/XML Infrastructure. [online]. IEEE [Institute of Electrical and Electronics Engineers], Proceedings of the 21st ICDE [International Conference on Data Engineering], 18. apríla 2005. [Citované 12. marca 2009]. Dostupné na: <http://www.ieee.org/>. Digital Object Identifier 10.1109/ICDE.2005.144
8. PÁLFY, J. 2008. Spracovanie XML dokumentácie v MS SQL. Bakalárska práca, FEI STU Bratislava, 2008.
9. XML Core Working Group. 1997. [online]. W3C [World Wide Web Consortium], aktualizácia 9. február 2009 14:19:37. [Citované 7. marca 2009]. Dostupné na: <http://www.w3.org/XML/Core/>
10. XML Shredding in SQL Server 2005. 2005. [online]. Windows Live, 18. mája 2005. [Citované 18. marec 2009]. Dostupné na: <http://home.live.com/>

Spoluautorom článku je Frank Schindler

¹V čase písania práce pán Jim Melton pracoval na verzii ISO/IEC 9075-14:2008, ktorá bola revidovaná pre medzinárodný štandard. Bližšie informácie na stránke <http://www.iso.org>.