DB2 9 的 pureXML：要用哪一種方法來查詢 XML 資料？

級別：中級

Matthias Nicola (mnicola@us.ibm.com)，DB2/XML 效能，IBM 矽谷實驗室
Fatma Ozcan (fozcan@almaden.ibm.com)，研究人員，IBM 艾瑪登研究中心

2006 年 6 月 8 日
更新日期：2007 年 8 月 28 日

DB2 9 已推出 pureXML® 支援，表示 XML 資料可以用固有的階層格式來儲存及查詢。爲了查詢 XML 資料，DB2 提供了兩種語言：SQL/XML 及 XQuery。您可分別使用 XQuery 及 SQL，但也可使用 SQL 內建的 XQuery，或 XQuery 內建的 SQL。如此一來，在查詢 XML 資料時就有很多彈性選擇，每種方法適用於特定情況。本文將說明這些方法、其個別優缺點，以及針對不同需求選擇適當方法的準則。本文已針對 DB2 9.5 進行更新。

前言

DB2 的純 XML 支援可提供有效且多用途的功能來管理 XML 資料。DB2 會以固有階層格式儲存及處理 XML，以免在將 XML 儲存成 CLOB 格式的文字或對映對到關聯式表格時，造成效能及彈性限制。有別於只能儲存 XML 的資料庫，DB2 V9 還能完美整合單一資料庫中的關聯式資料及 XML 資料，甚至是表格中的單一行。這種彈性反映在語言支援上，即可存取關聯式資料、XML 資料，或同時存取兩者。您可透過下列四種方法之一來查詢 XML：

- 純 SQL（沒有 XQuery）
- SQL/XML，即 SQL 內建的 XQuery
- XQuery 作爲單獨語言（沒有 SQL）
- 內建 SQL 的 XQuery

如需使用 XQuery 及 SQL/XML 來查詢 XML 資料的介紹，請參閱稍早在 developerWorks 發表的文章 "Query DB2 XML Data with SQL" 及 "Query DB2 XML data with XQuery"。我們假設您熟悉這兩篇文章所介紹的概念。請注意，XPath 是 XQuery 的子語言，所以每當我們提到 XQuery 的時候，其實也隱含地包括了 XPath。如果您有用過 DB2 XML Extender 中的 XSLT 樣式表或位置路徑，就可能知道 XPath。很多時候，XPath 就足以擷取 XML 值或表示 XML

述詞，所以就算您還不熟悉 XQuery 的其他特性，也可開始使用 XPath。

DB2 可讓您善用這些方法提高生產力，並且根據應用程式需求調整查詢方式。本文要探討的問題如下：

- 這四種方法的主要特性及優缺點爲何？
- 在什麼情況下應使用何種方法？

讓我們先簡介概要，然後再深入探討每種方法，瞭解其細節及特定範例。

摘要及準則

您可以用純 XQuery、SQL/XML 或內建 SQL 的 XQuery 來表示許多查詢。在某些情況下，您可能會覺得用其中一種方法來表示查詢邏輯，會比其他方法來得直接。一般而言，查詢 XML 資料的適當方法，必須視個別情況來選擇，要考量應用程式的需求及特性。不過，大致可以彙整出下列準則。

- **沒有 XQuery 或 XPath 的純 SQL**：只有擷取完整文件及進行插入、刪除及更新整份文件等作業時才有效。同時，文件的選擇必須根據相同表格中的非 XML 欄位。
- **附帶 XQuery 的 SQL/XML 或 SQL 陳述式中內建的 XPath**：可提供最廣泛的功能，限制最少。您可以在 XML 欄位表示述詞、擷取文件片段、將參數標記傳到 XQuery 表示式、使用全文檢索、進行 SQL 層次聚集及分組，同時還可用彈性的方式整合及連結 XML 和關聯式資料。這種方法適用於大部分應用程式。即使現在不需要用到全部這些優點，您還是可以選用這種方法，方便日後進行擴充。
- **XQuery**：是功能強大的查詢語言，專門用來查詢 XML 資料。因此，如果您的應用程式只需要查詢及操作 XML 資料，而不涉及任何關聯式資料的話，就適合用這個方法。這種語言有時比較簡單且直接。此外，如果您要將純 XML 資料庫移轉到 DB2 9，而且已經有 XQuery，可能會想繼續使用純 XQuery。
- **內建 SQL 的 XQuery**：如果您要善用關聯式述詞、索引及全文檢索，從 XML 欄位預先過濾文件，然後再輸入 XQuery，就適合採用此方法。XQuery 中的內建 SQL 也可以讓您在 XML 欄位執行外部函數。但是，如果您需要以分組和聚集的方式執行資料分析查詢，可能比較適合使用 SQL/XML。

無論您在一個陳述式中選擇什麼樣的 SQL 和 XQuery 組合，DB2 都會使用單

一混合式編譯器，針對整個查詢產生單一執行計畫並最佳化，而不影響查詢的執行效能。

下表彙總了四種 XML 資料查詢方法的個別優點。

表 1：摘要

	純 SQL	SQL/XML	SQL/XML 純 XQuery	內建 SQL/XML 的 XQuery
XML 述詞	-	+ +	+ +	+ +
關聯式述詞	+ +	+ +	-	+
XML 及關聯式述詞	-	+ +	-	++
連結 XML 與關聯式	-	++	-	++
連結 XML 與 XML	-	+	++	++
轉換 XML 資料	-	o	++	++
插入、更新、刪除	+ +	++	-	-
參數標記	+	++	-	-
全文檢索	+	++	-	++
XML 聚集和群組	-	++	o	o
函數呼叫	++	++	-	++

在上表中，"-" 表示該語言不支援某功能；"+" 則指支援該功能但可能有更有效或更方便的方法；"++" 意指該語言非常適合用來表示該功能，最後 "o" 表示雖然可用來表示該功能，但有點不便或效率不彰。

現在，讓我們來定義一些資料及表格範本，以便進一步瞭解具體的查詢範例。

表格及資料範本

爲方便討論各種 XML 查詢方法，我們會使用下列三個表格。部門表格有兩個欄位，分別命名爲 unitID 及 deptdoc。部門表格的每一列代表虛構公司的一個部門。unitID 欄位表示包含部門的事業單位（一個事業單位可包含數個部門），而 deptdoc 欄位則包含一份 XML 文件，列出該部門員工。專案表格只有一個名爲 projectDoc 的欄位，資料類型爲 XML。projectDoc 表格的每一列包含一份 XML 文件，說明特定專案。一個專案可能跟多個部門有關。爲了說明混合關聯式與 XML 查詢以及連結，我們也提供了純關聯式表格單位，列出每個事業單位的名稱、管理者等。一個事業單位可能有多個部門。

```
create table dept( unitID char(8), deptdoc xml)
```

unitID	deptdoc
WWPR	<dept deptID="PR27"> <employee id="901"> <name>Jim Qu</name> <phone>408 555 1212</phone> </employee> <employee id="902"> <name>Peter Pan</name> <office>216</office> </employee> </dept>
WWPR	<dept deptID="V15"> <employee id="673"> <name>Matt Foreman</name> <phone>416 891 7301</phone> <poffice>216</office> </employee> <description>This dept supports sales world wide</description> </dept>
S-USE	…
…	…

```
create table project(projectDoc xml)
```

projectDOC
<project ID="P0001"> <name>Hello World</name> <deptID>PR27</deptID> <manager>Peter Pan</manager> </project>
<project ID="P0009"> <name>New Horizon</name>

```
  <deptID>PR27</deptID>
  <deptID>V15</deptID>
  <manager>Matt Foreman</manager>
  <description>This project is brand new</description>
</project>
```

```
```

unitID	name	manager	...
WWPR	Worldwide Marketing	Jim Qu	...
S-USE	Sales US East Coast	Tom Jones	...
...	...	...	...

純 SQL

您可以使用沒有 XPath 或 XQuery 的純 SQL，來讀取不含 XML 述詞的完整文件。只要應用程式可根據相同表格上的關聯式述詞來識別 XML 文件，以擷取完整文件，這就比較方便、簡單。例如，「查詢一」會擷取事業單位 "WWPR" 的所有部門文件：

查詢一

```
select deptdoc
from dept
where unitID = 'WWPR';
```

同樣地，「查詢二」會傳回由 Jim Qu 管理的所有部門的 XML 資料。

查詢二

```
select deptdoc
from dept d, unit u
where d.unitID = u.unitID and u.manager= 'Jim Qu';
```

最明顯的缺點就是您不能表示 XML 資料本身的述詞，或只是擷取 XML 文件的片段。在本範例中，純 SQL 無法只選取部門 PR27，並且只傳回員工姓名。

如果查詢不需要 XML 欄位的述詞，而且一律傳回完整的 XML 文件，則純 SQL 或許就能滿足您的需求。在此情況下，您可以將 XML 儲存爲 VARCHAR 或 CLOB 欄位，提高插入及擷取完整文件的作業效能。

即使不使用 DB2 V9 的任何 SQL/XML 或 XQuery 支援，純 SQL 還是可以讓您在查詢中列出全文檢索條件。有了 DB2 Net Search Extender，您可建立 XML 欄位的全文索引，以支援文字搜尋，範圍從基本關鍵字搜尋到進階搜尋（37 種語言的字根還原、辭彙集及模糊搜尋）。您也可以用路徑表示式，將文字搜尋限制在特定文件章節。根據文字索引查閱，「查詢三」會傳回凡是 /dept/description 之下，內含 "sales" 字串的所有部門文件：

查詢三

```
select deptdoc
from dept
where CONTAINS(deptdoc,'SECTION("/dept/description") "sales" ')=1;
```

SQL/XML（SQL 中內建的 XQuery/XPath）

SQL/XML 是 SQL 語言標準的一部分，可定義新的 XML 資料類型，以及查詢、建構、驗證及轉換 XML 資料的函數。DB2 V8 已提供許多 SQL/XML 發佈函數，所以使用者可使用 XMLELEMENT、XMLATTRIBUTE、XMLFOREST、XMLAGG 及其他函數，從關聯式資料建構 XML。

DB2 9 新增了 SQL/XML 查詢函數，包括 XMLQUERY、XMLTABLE 及 XMLEXISTS 述詞。這些函數可讓使用者在 SQL 陳述式中內建 XQuery 或簡單的 XPath 表示式。

如「查詢 4」所述，XMLQUERY 函數通常用於 select 子句，以便從 XML 欄位中擷取 XML 片段，而 XMLEXISTS 通常用於 where 子句，以表示 XML 資料的述詞。

查詢四

```
select unitID, XMLQUERY('for $e in $d/dept/employee return $e/name/text()'
                                        passing  d.deptdoc as "d")
from dept d
where  unitID LIKE 'WW%' and
      XMLEXISTS('$d/dept[@deptID = "V15"]' passing d.deptdoc as "d");
```

此查詢範例使用 XMLEXISTS 來選取 "WW" 部門 V15，並運用 XMLQUERY 來傳回該部門文件的所有員工姓名。結果如下：

WWPR	Matt Foreman

此查詢也展示您可以透過整合方式，使用 SQL/XML 來查詢 XML 及關聯式資料。select 子句會從關聯式及 XML 欄位擷取資料，而 where 子句則包含關聯式及 XML 述詞。

在 DB2 9.5，您可進一步簡化此查詢，即刪除 XMLEXISTS 及 XMLQUERY 函數中的 "passing" 子句。如果只是將 XML 欄位 "deptdoc" 傳入 XQuery，您可以用 XQuery 表示式的 $DEPTDOC 直接參照該欄位，不需要 "passing" 子句。請參閱「查詢五」。

查詢五

```
select unitID, XMLQUERY('for $e in $DEPTDOC/dept/employee return
$e/name/text()')
from dept d
where  unitID LIKE 'WW%' and
      XMLEXISTS('$DEPTDOC/dept[@deptID = "V15"]');
```

我們可以使用 XMLTABLE 函數來表示相同的查詢，如「查詢六」所示。在此格式中，我們會指定條件來限制輸入資料，並擷取想要的輸出值。在「查詢六」中，XMLTABLE 函數的 XQuery 表示式會找出任職於部門 V15 的員工，而 COLUMNS 子句的路徑表示式 ("name/text()") 會傳回他們的姓名。其輸出結果跟「查詢四」和「查詢五」一樣。

查詢六

```
```

```
select d.unitID, T.name
from dept d, XMLTABLE('$d/dept[@deptID="V15"]/employee' passing d.deptdoc as
"d"
                    COLUMNS
                      name varchar(50) path 'name/text()'  ) as T
where unitID LIKE 'WW%';
```

SQL/XML 的優點

SQL/XML 方法具有下列優點：

- 如果您已經有 SQL 應用程式，偶爾需要新增一些 XML 功能，則適合使用 SQL/XML。
- 如果您喜歡使用 SQL，並且 SQL 是您和團隊最熟悉的語言，您希望繼續以 SQL 作爲主要語言，那麼就適合使用 SQL/XML。
- 如果您的查詢要同時從關聯式欄位及 XML 欄位傳回資料，則適合使用 SQL/XML。
- 如果您的查詢需要全文檢索條件，如「查詢三」所示，則適合使用 SQL/XML。
- 如果要以集合方式傳回結果，而遺漏的 XML 元素則以空值表示，則適合使用 SQL/XML。
- 如果要使用參數標記，因爲 DB2 V9 XQuery 並不支援外部參數，則適合使用 SQL/XML。XMLQUERY、XMLTABLE 及 XMLEXISTS 中的傳遞機制可讓您將 SQL 參數標記當作變項 ($x) 傳入內建的 XQuery 表示式：

  查詢七

  ```
  select deptdoc
  from dept
  where XMLEXISTS('$d/dept[@deptID =  $x]'
                              passing deptdoc as "d", cast(? as varchar(8)) as
  "x");
  ```

- 對於需要整合關聯式及 XML 資料的應用程式，則適合使用 SQL/XML。它提供最簡便方法，連結 XML 資料及關聯式資料。下列範例會選取所有部門的事業單位 ID，而這些部門都包含一個員工是事業單位表格中的管理者。此查詢會連結關聯值 (unit.manager) 和 XML

值 (//employee/name)：

查詢八

```
select u.unitID
from dept d, unit u
where XMLEXISTS('$d//employee[name = $m]'
                                    passing d.deptdoc as "d", u.manager as "m");
```

爲了完成這項連結動作，我們將事業單位管理者傳入 XMLEXISTS 述詞，使得實際合併條件爲 XQuery 述詞。反之，我們可以從 XML 文件將員工姓名擷取到 SQL 環境定義，使合併條件爲 SQL 述詞：

查詢九

```
select u.unitID
from dept d, unit u
where u.manager = XMLCAST(XMLQUERY('$d//employee/name '
                                          passing d.deptdoc as "d") as
char(20));
```

「查詢八」通常會比「查詢九」好用，因爲 XMLCAST 函數必須是單一輸入值。我們的範例就不適用，因爲部門有多位員工。帶有 XMLCAST 的「查詢九」十分適合用於每份文件只發生一次的的 XML 值連結，因爲它可以對 unit.manager 使用關聯式索引。「查詢八」則無法使用該索引，因爲連結條件不是關聯式述詞，而是 XQuery 述詞。

- SQL/XML 適用於分組和聚集 XML。XQuery 語言並不提供明確分組 (group-by) 建構。雖然在 XQuery 中可以使用自行連結 (self-join) 來表示分組和聚集，但相當不便。例如，讓我們來計算以分公司分組的員工人數，即每個分公司有多少員工。「查詢十」顯示如何使用純 XQuery 來執行這個查詢。「查詢十」的 db2-fn:xmlcolumn 函數可存取 DB2 中的 XML 資料。它加入 XML 欄位的名稱作爲引數，然後傳回該欄位所儲存的一連串 XML 值。使用 SQL/XML 函數（如 XMLTABLE 或 XMLQUERY）擷取 XML 欄位中的資料項目比較方便，然後再用熟悉的 SQL 概念來表示分組和聚集。「查詢十一」會傳回跟「查詢十」相同的邏輯結果，但使用的是 XMLTABLE 及 SQL Group By 子句。

查詢十

```
XQUERY
for $o in
distinct-values(db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept/employee/office)
let $emps :=
db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept/employee[office/text()=$o]
return
           <result><office>{$o}</office><cnt>{count($emps)}</cnt></result>;

Result:
<result><office>216</office><cnt>2</cnt></result>
```

查詢十一

```
select X.office, count(X.emp)
from dept, XMLTABLE ('$d/dept/employee' passing deptdoc as "d"
  COLUMNS
    emp        VARCHAR(30)     PATH  'name',
    office     INTEGER         PATH  'office ') as X
GROUP BY X.office;

Result:
216   2
-     1
```

在「查詢十一」中，XMLTABLE 函數會從每份有 "emp" 及 "office" 欄位的表格文件中，擷取 /dept/employee/name 及 /dept/employee/office。針對該表格使用 SQL 分組及聚集函數，通常會比使用純 XQuery 產生相同結果更有效率。

請注意，「查詢十」會多出一列，因爲 SQL 的分組函數還會針對 NULL 值產生一個群組，因爲在我們的範例中，有一位員工沒有分公司資訊。「查詢九」並沒有爲該員工產生一列，因爲 for 迴圈會迭代正確無誤的分公司值，但不含遺漏分公司資訊的值。

SQL/XML 的缺點

- SQL/XML 不一定是轉換 XML 文件最好的方式。單獨使用 XQuery 可能更適合，且如果只是處理 XML 資料，也會比較直接。不過，DB2 9.5 也有提供 XSLTRANSFORM 函數，可用於 SQL 陳述式，轉換含 XSLT 樣式表的 XML 文件。尤其是，XMLQUERY 函數可以當作 XSLTRANSFORM 函數中的引數。
- 您可能會發現，用純 XQuery 來表示兩個 XML 欄位（或兩個 XML 值）之間的連結，會比使用 SQL/XML 更直接。例如，「查詢十二」會連結部門及專案表格的 XML 欄位，傳回負責任一專案的員工。

  查詢十二

```
select XMLQUERY('$d/dept/employee' passing d.deptdoc as "d")
from dept d, project p
where XMLEXISTS('$d/dept[@deptID=$p/project/deptID]'
              passing d.deptdoc as "d", p.projectDoc as "p");
```

  在「查詢十二」，我們將所有部門及專案文件傳入 XQuery 的 XMLEXISTS，並在當中表示連結條件。您可能會發現用純 XQuery（查詢十四）來撰寫這種連結會比較容易。

單獨使用 XQuery 語言

讓我們先回過頭來問：何謂 XQuery？爲什麼需要 XQuery？就像 SQL 是專門用於關聯式資料模型的查詢語言，XQuery 則是專門用來查詢 XML 資料的語言。由於 XML 資料跟關聯式資料差異相當大，我們需要專用的語言來有效處理 XML 資料。關聯式資料的結構是扁平的、強制類型 (strongly typed) 且未排序的，而 XML 資料是有排序、呈巢狀結構、選用類型，且往往是非正規及半結構化。SQL 無法處理這種資料結構，但 XQuery 則專門處理這種資料。具體而言，XQuery 不僅可以導覽 XML 文件樹狀結構並擷取 XML 片段，還可以納入表示式，來建立、操作及迭代一連串 XML 項目並建構新的 XML 資料。

IBM 已擴充所有 DB2 主要應用程式設計介面 (API)，來支援 XQuery 作爲第一級 (first-class) 語言，就如 SQL。其中包括 CLI/ODBC、內建 SQL、JDBC 及 .NET。因此，DB2 指令行處理器也支援 XQuery。您可以直接送出 XQuery，不必修改，但必須以 XQUERY 關鍵字開頭，以通知 DB2 使用 XQuery 剖析器，如以下範列所示：

查詢十三

```
XQUERY
for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
where $dept/@deptID="PR27"
return $dept/employee/name;
```

「查詢十三」會迭代每份部門文件的每個 "dept" 元素，然後傳回部門 PR27 的各員工姓名。

XQuery 的優點

- XQuery 適用於不需使用或不想使用 SQL 或關聯式結構的純 XML 應用程式。
- XQuery 適用於將純 XML 資料庫移轉到 DB2 V9。現有 XQuery 通常只要稍微變更即可在 DB2 中執行。例如，XQuery 的輸入資料源自 DB2 中的 db2-fn:xmlcolumn() 函數，而其他資料庫可能稱之爲 collection()。在這種情況下，只要簡單的重新命名即可。
- 如果要將查詢結果嵌入（並傳回）新建的 XML 文件（有別於資料庫中的文件），則適合使用 XQuery。
- XQuery 非常適合用來表示兩份 XML 文件之間的連結，以及連結 XML 值。例如，您可以使用「查詢十四」的 XQuery，以更直接、有效的方式來表示「查詢十二」中的連結。

  查詢十四

```
XQUERY
for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
   for $proj in db2-fn:xmlcolumn("PROJECT.PROJECTDOC")/project
where $dept/@deptID = $proj/deptID
return $dept/employee;
```

XQuery 的缺點

- 若使用純 XQuery，則無法使用 DB2 Net Search Extender (NSE) 提供的全文檢索功能。全文檢索需要用到 SQL。
- 純 XQuery 不允許您呼叫 SQL 使用者定義函數 (UDF) 或以 C 或

Java 撰寫的外部 UDF。

- 目前，DB2 並沒有提供以參數標記來呼叫純 XQuery 的方法。若要將參數傳給 XQuery，您必須使用 SQL/XML 將參數標記 ("?") 轉換成命名變項。例如，在「查詢十三」中，您可能想用問號 (?) 作爲 SQL 式的參數標記，而不用文字值 "PR27"。但這樣就會變成無效查詢。

  在「查詢七」中，SQL/XML 可讓您將 SQL 參數標記當作變項傳入內建的 XQuery 表示式。SQL/XML 查詢的 SELECT 子句通常有 XMLQUERY 函數，擷取 XML 文件片段，而 WHERE 子句的 XMLEXISTS 述詞則過濾符合文件。如果您想以單一 FLWOR 表示式來表示整個查詢邏輯，而不用兩個獨立的 XQuery 呼叫（一個以 XMLQUERY，另一個以 XMLEXISTS），而且還要使用參數標記，則可以將「查詢十三」改寫成「查詢十五」：

  查詢十五

```
values( XMLQUERY(
                ' for $dept in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
                 where $dept/@deptID = $z
                 return $dept/employee/name'
           passing cast(? as varchar(8)) as "z" ) );
```

  「查詢十五」是 SQL/XML 陳述式，因爲它只是 SQL fullselect 的 VALUES 子句。VALUES 子句會針對結果表格中的每個欄位指定一個表示式，傳回值表格。「查詢十五」的結果表只有一個 XML 類型的列和欄位，XMLQUERY 函數則會產生輸出欄位的值。所有員工姓名會以單一 XML 值傳給用戶端。除了「查詢十五」包括外部變項 ($z) 之外，其 FLOWR 表示式幾乎跟「查詢十三」完全一樣，$z 變項會當作參數傳入 XMLQUERY 函數。

內建 SQL 的 XQuery

純 XQuery 可存取 XML 資料，但只限 XML 資料。這個方法適合處理純 XML 資料，但如果應用程式需要存取 XML 及關聯式資料，發揮兩個語言及資料模型的完整功能，便不能只用 XQuery。使用本文稍早提到的 SQL/XML，在 SQL 中內建 XQuery，可達到此目標。反之，在 XQuery 中內建 SQL 也可以提供其他用途。除了上一節所提到的優缺點，下列各項也很重要：

內建 SQL 的 XQuery 之優點

- 如果您只要根據關聯式欄位的條件處理 XML 文件片段，就適合使用內建 SQL 的 XQuery。您可能會想將 XQuery 只套用到 XML 欄位中的 XML 文件片段。特別是，您可以使用關聯式述詞，限制特定 XQuery 的輸入項目。爲此，DB2 提供了另一個輸入函數 db2-fn:sqlquery，可呼叫 XQuery 中的 SQL 查詢。此函數使用 SQL SELECT 陳述式，然後傳回 XML 欄位作爲輸出。例如，「查詢十六」不處理 XML 欄位 deptdoc 中的所有文件，而是使用一個內建 SQL 陳述式，透過連結套用述詞的事業單位表，預先過濾 XML 文件：

  查詢十六

```
XQUERY
for $emp in db2-fn:sqlquery("select deptdoc
                             from dept d, unit u
                             where d.unitID=u.unitID and
                             u.manager = 'Jim Qu'")/dept/employee
where $emp/office = 216
return $emp/name;
```

  兩表的 unitID 欄位及事業單位表中管理者欄位的一般關聯式索引，可加速內建 SQL 的查詢處理。

- 內建 SQL 的 XQuery 可讓您使用全文檢索，因爲您可在內建 SQL 陳述式的 WHERE 子句中使用文字搜尋函數 "contains"。在「查詢十七」中，XQuery 內的 SQL 陳述式會從部門表格中選取文件，只要 "sales" 一字有出現在 /dept/description 任何地方的文件都會選取。全文索引會快速找出這些 XML 文件，然後輸入 FLWOR 表示式，從這些文件中擷取所有員工姓名。以我們的範本表格來說，「查詢十八」會傳回相同結果，但使用的是 SQL/XML 表示法。

  查詢十七

```
XQUERY
for $emp in db2-fn:sqlquery("
                select deptdoc from dept
```

```
                where CONTAINS(deptdoc, 'SECTION(""/dept/description"")
""sales"" ')=1


")//employee
return $emp/name;
```

查詢十八

```
select XMLQUERY('$d//employee/name' passing deptdoc as "d")
from dept
where CONTAINS(deptdoc,'SECTION("/dept/description") "sales" ')=1;
```

- 內建 SQL 的 XQuery 很適合用於需要整合關聯式及 XML 資料的應用程式。您可以合併查詢 XML 資料及關聯式資料。SQL/XML 也一樣。但您可能會覺得，使用帶有 XMLEXISTS 函數的 SQL/XML 來連結 XML 及關聯式數值會比較容易。比較「查詢十九」及「查詢二十」。「查詢十九」會傳回部門員工中有事業單位管理者的 deptID。內建 SQL 陳述式會將事業單位表中的管理者姓名編成 XML 類型（在此範例是 XML 字串），然後傳入 XQuery。XQuery where 子句包含合併條件，以比較管理者姓名及員工姓名元素。「查詢二十」以 SQL/XML 表示法表示相同的連結。

查詢十九

```
XQUERY
for $m in db2-fn:sqlquery('select XMLCAST(u.manager as XML) from unit u')
for $d in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept
where $d/employee/name = $m
return $d/data(@deptID);
```

查詢二十

```
select XMLQUERY('$d/dept/data(@deptID)' passing  d.deptdoc as "d")
from dept d, unit u
where XMLEXISTS('$d/dept/employee[name = $m]'
                                passing d.deptdoc as "d", u.manager as "m");
```

此外，您也可以想想，如果要傳回符合的員工姓名，而不是 deptID，要如何修改「查詢十九」及「查詢二十」。「查詢十九」很容易修改。只要直接把傳回子句改爲：return $m。在「查詢二十」，直覺的想法是將 XMLQUERY 函數中的路徑變更爲：XMLQUERY('$d/dept/employee/name' passing d.deptdoc as "d")。不過，這樣會傳回指定部門的所有員工姓名，而不只是管理者的姓名。若只要從符合的部門文件中擷取管理者姓名，XMLQUERY 函數必須加入類似「查詢二十」 XMLEXISTS 的姓名述詞，即 XMLQUERY('$d/dept/employee/name[.= $m]' passing d.deptdoc as "d", u.manager as "m")。差別在於 XMLEXISTS 使用述詞來查詢整份文件，但 XMLQUERY 函數則套用述詞在符合 XMLEXISTS 的特定文件中尋找特定姓名。

- DB2 9.5 可讓您將參數從 XQuery 傳到 db2-fn:sqlquery 函數中的內建 SQL 陳述式。許多情況都適用這種方式。例如，「查詢二十一」所表示的連結跟「查詢十九」及「查詢二十」相同。"for" 子句會迭代員工姓名，而 "where" 子句會檢查單位表格中是否有相同的管理者姓名。db2-fn:sqlquery 函數會將 $n 視爲其他引數。select 陳述式求值時，函數 "parameter(1)" 會使用 $n 值。可以使用多項類似的參數。如果 db2-fn:sqlquery 函數傳回空白序列，XQuery "where" 子句的值就是 False 值；若函數傳回的不是空白序列，則子句值爲 True。這是由於 XQuery 的存在語義。因此，SELECT 子句也可以是 select XMLCAST（"yes" 表示 XML），因爲重點是值存不存在，而不是值本身爲何。

查詢二十一

```
XQUERY
for $n in db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept/employee/name
where db2-fn:sqlquery('select XMLCAST(manager as XML) from unit
                   where manager = parameter(1)', $n)
return $n/../../data(@deptID);
```

如果「查詢十九」和「查詢二十」已經可以達到目的，爲什麼還要用其他方式表示連結呢？「查詢二十一」的連結述詞屬於 SQL 層次，即可讓 DB2 對 unit.manager 使用關聯式索引。「查詢十九」及「查詢二十」則以 XML 層次表示連結條件，所以可對 /dept/employee/name 使用 XML 索引。這跟上述「查詢八」及「查詢九」的比較類似。

- 如「查詢十九」所示，內建 SQL 的 XQuery 可將關聯式資料傳入 XQuery。這樣可以結合 XML 及關聯式資料。「查詢二十二」會建構一個結果文件，包含事業單位及部門資訊。部門資訊是從 XML 欄位 deptdoc 擷取出來的 XML 文件。事業單位資訊則來自關聯式表格單位。內建 SQL 陳述式會使用 SQL/XML 發佈函數來建構 XML 元素 "Unit"，其中有三個子元素，它們的值是得自事業單位表的關聯式欄位，即欄位 unitID、姓名及管理者。

查詢二十二

```
XQUERY
   let $d := db2-fn:sqlquery("select deptdoc from dept where unitID = 'WWPR'
")
   let $u := db2-fn:sqlquery("select XMLELEMENT(NAME ""Unit"",
                                          XMLFOREST(unitID, name, manager))
                                    from unit where unitID = 'WWPR' " )
   return <result>
                      <units>{$u}</units>
                      <department>{$d}</department>
            </result>;
```

「查詢二十二」的輸出結果如下：

```
  <result>
      <units><unit>
               <UNITID>WWPR</UNITID>
               <NAME>World Wide Markeing</NAME>
               <MANAGER>Jim Qu</MANAGER>
             </unit>
             <unit>
               <UNITID>WWPR</UNITID>
               <NAME> ... </NAME>
               <MANAGER> ... </MANAGER>
            </unit>
                                ....
             </units>
```

```
      <department>
         <dept deptID="PR27">
             <employee id="901">
             <name>Jim Qu</name>
             <phone>408 555 1212</phone>
           </employee>
           <employee id="902">
             <name>Peter Pan</name>
             <office>216</office>
           </employee>
         </dept>
      </department>
  </result>
```

- 內建 SQL 的 XQuery 非常適合查詢，因爲內建 SQL 陳述式可呼叫使用者自訂函數 (UDF)。許多 IT 單位都廣泛使用 UDF 來封裝重要的業務邏輯，並且簡化應用程式開發需求。這種方式很重要，因爲純 XQuery 無法呼叫 UDF。

內建 SQL 的 XQuery 之缺點

- 雖然 db2-fn:sqlquery 可以在 XQuery 中內建 SQL 陳述式，但目前還不能在內建 SQL 陳述式中加入參數標記。
- 在 DB2 9 中，db2-fn:sqlquery 不容許將值從 XQuery 傳入 SQL。但 DB2 9.5 已解除這項限制，如「查詢二十一」所示。

XML 查詢結果

但是要注意，您撰寫特定查詢的方式不同，DB2 查詢結果的格式可能也會不同。例如，純 XQuery 傳回項目（如元素或文件片段）至結果集時，每列一個項目，即使有多個項目來自資料庫的同一文件（列）。相對的，使用 XMLQUERY 時，SQL/XML 可能會將多個項目傳回至同一列，而不是分別傳回至多個列。某些情況可能比較適合這種方式，某些情況則不適合，端視特定應用程式而定。請參考以下範例。

「查詢二十三」及「查詢二十四」都要求部門文件的員工姓名。「查詢二十三」是以 SQL/XML 表示，「查詢二十四」則以 XQuery 撰寫。

查詢二十三

```
select XMLQUERY('$d/dept/employee/name' passing deptdoc as "d")
from dept;
```

查詢二十四

```
XQUERY db2-fn:xmlcolumn("DEPT.DEPTDOC")/dept/employee/name;
```

「查詢二十三」會傳回部門員工姓名，每份部門文件一列。

```
<name>Jim Qu</name> <name>Peter Pan </name>
<name>Matt Foreman</name>
```

相對的，「查詢二十四」也會傳回員工姓名，但每位員工自成一列：

```
<name>Jim Qu</name>
<name>Peter Pan</name>
<name>Matt Foreman</name>
```

應用程式通常比較容易處理「查詢二十四」的結果，即一次一個 XML 值。不過，在此情況下，您就不知道哪些員工姓名源自相同的部門文件。「查詢二十三」的輸出結果會保留此資訊，但應用程式可能較難處理其結果，因爲結果列可能需要進一步分割，才能存取個別員工姓名。如果應用程式使用 XML Parser 來吸收 DB2 的每個 XML 結果列，剖析器會拒絕「查詢二十一」的第一個結果列，因爲此文件不是形式完整的文件（缺乏單一根元素）。若要解決此問題，您可將 XMLELEMENT 建構者新增到「查詢二十三」來加入單一根元素，如「查詢二十五」所示：

查詢二十五

```
Select XMLELEMENT(name "employees",
   XMLQUERY('$d/dept/employee/name'   passing d.deptdoc as "d"))
from dept d;
```

這會改變查詢結果，使每個結果列都是形式完整的 XML 文件：

```
<employees><name>Jim Qu </name><name>Peter Pan
</name></employees>
```

```
<employees><name>Matt Foreman</name></employees>
....
```

回顧「查詢十五」，它是使用 SQL VALUES 子句及 XMLQUERY 函數，將參數傳入 XQuery。但是，「查詢十五」的結果將所有員工姓名傳回至單一列。如果您想要以個別列呈現每個員工姓名，同時需要使用參數標記，則可透過「查詢二十六」所示的 XMLTABLE 函數來達到此目的：

查詢二十六

```
select X.*
from dept d, XMLTABLE('for $dept in $d/dept
                    where $dept/@deptID = $z
                    return $dept/employee/name'
                    passing d.deptdoc as "d", cast(? as varchar(10)) as "z"
          COLUMNS
                    "name"   XML  PATH '.') as X ;
```

總結

DB2 pureXML 提供了一系列豐富的 XML 資料查詢方法。您可根據應用程式的需求及特性選擇最適合自己的方法。如果要整合關聯式及 XML 資料，則多數情況下，SQL/XML 是最佳選擇。特別是，SQL/XML 可以針對 XML 資料使用參數標記。如果您用的是只限 XML 資料的應用程式，則純 XQuery 就是功能強大的方法，而且可以用內建 SQL 加強，進行全文檢索及呼叫 UDF。本文探討的範例可協助您做出正確的決策。您可修改本文的查詢範例，撰寫適合您的 XML 資料查詢。

一般而言，您無須將查詢寫得過於複雜，只要能夠滿足所需即可。例如，XQuery 可內建 SQL 陳述式，其 SQL 陳述式又可再內建 XQuery，依此類推。但根據我們的經驗，一般所需的查詢邏輯，沒有必要使用兩種語言形成一層以上的複雜巢狀結構。因此，我們建議，SQL 只內建一層 XQuery，或者是 XQuery 只內建一層 SQL。

致謝

感謝 Don Chamberlin、Bert van der Linden、Cindy Saracco、Jan-Eike Michels 及 Hardeep Singh 協助審閱本文。

下載

說明	名稱	檔案大小	下載方法
本文的查詢 Script	DDLandQueries.txt	9KB	HTTP

→ 下載方法的相關資訊

作者簡介

Nicola 博士是 IBM 矽谷實驗室的 XML 資料庫效能技術主任。他主要是負責 DB2 XML 效能的所有層面，包括 DB2 中的 XQuery、SQL/XML 及所有原生 XML 特性。Nicola 博士與 DB2 XML 開發團隊，以及使用 XML 的客戶及商業夥伴密切合作，協助他們設計、導入及 XML 最佳化解決方案。加入 IBM 之前，Nicola 博士曾任職於 Informix Software，負責資料倉儲效能。有四年的時間，他也曾進行分散式及備份資料庫的相關研究與產業專案。1999 年，取得德國阿亨 (Aachen) 技術大學電腦科學博士學位。

Ozcan 博士自 2001 年開始擔任 IBM 艾瑪登研究中心的研究人員。她是 DB2 XML 編譯器團隊的成員，也是 XML 查詢語言、XQuery 語意及重寫最佳化的技術領導者。2001 年，她取得美國馬里蘭大學帕克分校電機工程學系電腦科學博士學位。她的研究專題包括 XML 查詢語言及查詢最佳化、異質資訊系統整合，以及軟體代理程式。她也是 ACM SIGMOD 的會員。