

## DB2 9 的 pureXML：要用哪一種方法查詢 XML 資料？

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

原始文章刊載於:

<http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0606nicola/index.html>

DB2 9 推出了 pureXML® 支援，也就是以階層格式來儲存及查詢 XML 資料。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，也可使用內含 XQuery 的 SQL，或內含 SQL 的 XQuery。如此一來，在查詢 XML 資料時，就有很大的彈性及選擇，每種選項在特定情況下都相當實用。本文將說明這些選項、個別優缺點，以及依照需求選擇適當方法的準則。

### 前言

DB2 的純 XML 支援提供了有效且多用途的功能，方便管理 XML 資料。DB2 是以其固有階層格式儲存及處理 XML，以免在將 XML 儲成 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。和只能儲存 XML 的資料庫不同，DB2 V9 還能緊密整合單一資料庫中的關聯式資料及 XML 資料，甚至是表格的其中一行。這種彈性也反映在語言支援上，您可存取關聯式資料、XML 資料，或同時進行存取。您可透過下列任一方法查詢 XML：

- 純 SQL（不使用 XQuery）
- SQL/XML，即內含 XQuery 的 SQL
- 以 XQuery 作爲單獨語言（不使用 SQL）
- 內含 SQL 的 XQuery

如需初步瞭解如何使用 XQuery 及 SQL/XML 來查詢 XML 資料，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"。我們假設您對這兩篇文章中的概念都很熟悉。請注意，XPath 是 XQuery 的子語言，所以只要提到 XQuery 的時候，也等於間接提到 XPath。如果您使用過 DB2 XML Extender 中的 XSLT 樣式表或位置路徑，大概就會知道 XPath。很多時候，光使用 XPath 就足以擷取 XML 值或表示 XML 述詞，所以就算還不熟悉 XQuery 的其他特性，您也可從 XPath 著手。

DB2 可讓您充分利用這些選項，發揮出最大的生產力，並且根據應用程式需求調整查詢方式。本文探討的問題如下：

- 這四種選項的主要特性及優缺點為何？
- 不同情況應使用何種方法？
- 讓我們先以摘要大略說明，然後再深入探討每個選項，瞭解其細節及特定範例。

## 摘要及準則

您可以使用純 XQuery、SQL/XML 或內含 SQL 的 XQuery 來表示許多查詢。在某些情況下，您可能會覺得使用某選項來表示查詢邏輯，會比其他選項來得直觀。一般而言，您必須視個別情況來選擇「適當」方法來查詢 XML 資料，並要考量應用程式的需求及特性。不過，我們可以彙整出下列準則。

- **不含 XQuery 或 XPath 的純 SQL**：只適用於完整文件的擷取及作業，如插入、刪除及更新整份文件。同時，還必須根據相同表格中的非 XML 直欄選擇文件。
- **附帶 XQuery 的 SQL/XML 或 SQL 陳述式中內含的 XPath**：提供最廣泛的功能，且限制最少。您可以在 XML 直欄表示述詞、擷取文件片段、將參數標記傳到 XQuery 表示式、使用全文檢索、在 SQL 層次聚集及分組，同時還可用彈性的方式整合及合併 XML 和關聯式資料。這種方法適用於大部分應用程式。即使您現在用不到這些優點，可能還是想選用這種方法，以便日後進行擴充。
- **XQuery**：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，而不牽涉任何關聯式資料的話，這會是理想的選擇。這種語言有時還較為簡單且直觀。此外，如果您要將純 XML 資料庫移轉到 DB2 9，而且已經有 XQueries，您可能會比較想繼續使用純 XQuery。
- **內含 SQL 的 XQuery**：如果您要充分利用關聯式述詞、索引及全文檢索，並從 XML 直欄預先過濾文件，然後再輸入 XQuery，便適合使用此方法。XQuery 中的內含 SQL 也可以讓您在 XML 直欄執行外部函數，但如果需要以分組和聚集方式分析查詢資料，可能會想用 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，針對整個查詢產生及最佳化單一執行計畫，並不影響查詢的執行效能。

下表彙總了四種 XML 資料查詢選項的個別優點。

表 1：摘要

	純 SQL	SQL/XML	純 XQuery	內含 SQL/XML 的 XQuery
XML 述詞	-	++	++	++
關聯式述詞	++	++	-	+
XML 及關聯式述詞	-	++	-	++
合併 XML 與關聯式	-	++	-	++

合併 XML 與 XML	-	+	++	++
轉換 XML 資料	-	o	++	++
插入、更新、刪除	++	++	-	-
參數標記	+	++	-	-
全文檢索	+	++	-	++
XML 聚集和分組	-	++	o	o
函數呼叫	++	++	-	++

在上表中，"-" 表示特定語言不支援某功能，"+" 則指支援該功能但可能有更有效或更方便的方法，"++" 意指特定語言非常適合用來表示某功能，最後 "o" 表示雖然可用來表示該功能，但較為不便或效率不彰。

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

為方便討論 XML 查詢選項，我們使用了下列三個表格。部門表格有兩個直欄，分別命名為 `unitID` 及 `deptdoc`。部門表格的每一列即代表虛構公司的一個部門。`unitID` 直欄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` 直欄則包含列出部門員工的 XML 文件。專案表格只有一個名為 `projectDoc` 的直欄，屬 XML 類型。`projectDoc` 表格的每一列均包含說明特定專案的 XML 文件，且一個專案可能跟多個部門有關。為了說明混合關聯式、XML 查詢及合併等功能，我們也提供了純關聯式表格單位，可列出每個事業單位的名稱、管理者等。一個事業單位可能包含多個部門。

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

unitID	deptdoc
WWPR	<pre>&lt;dept deptID="PR27"&gt;   &lt;empl oyee id="901"&gt;     &lt;name&gt;Jim Qu&lt;/name&gt;     &lt;phone&gt;408 555 1212&lt;/phone&gt;   &lt;/empl oyee&gt;   &lt;empl oyee id="902"&gt;     &lt;name&gt;Peter Pan&lt;/name&gt;     &lt;offi ce&gt;216&lt;/offi ce&gt;   &lt;/empl oyee&gt; &lt;/dept&gt;</pre>

WWPR	<pre>&lt;dept deptID="V15"&gt;   &lt;employee id="673"&gt;     &lt;name&gt;Matt Foreman&lt;/name&gt;     &lt;phone&gt;416 891 7301&lt;/phone&gt;     &lt;office&gt;216&lt;/office&gt;   &lt;/employee&gt;   &lt;description&gt;This dept supports sales worldwide&lt;/description&gt; &lt;/dept&gt;</pre>
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，識別出 XML 文件來進行完整文件擷取，不但簡單且又相當方便。比方說，「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 XML 資料。

- 查詢二

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

但缺點也很明顯，您將無法表示 XML 資料本身的述詞，或是只擷取 XML 文件的片段。在本範例中，純 SQL 無法只選取部門 PR27，並僅傳回員工姓名。

如果您的查詢不需要 XML 直欄的述詞，而且只需傳回完整的 XML 文件，則純 SQL 或許就能滿足您的需求。在此情況下，您也可以像 VARCHAR 或 CLOB 直欄一樣儲存 XML，便能在完整文件插入與擷取作業方面，取得效能上的優勢。

即使不使用 DB2 V9 中的任何 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 直欄的全文索引，以支援文字搜尋，範圍從基本關鍵字搜尋到進階搜尋（37 種語言的字形變化、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 /dept/description 之下，包含 "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 中內含 XQuery/XPath)

SQL/XML 是 SQL 語言標準的一部分，可定義新的 XML 資料類型，並可查詢、建構、驗證及轉換 XML 資料的函數。DB2 V8 提供許多 SQL/XML 發佈函數，因此使用者可使用 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，可讓使用者在 SQL 陳述式中內含 XQuery 或簡單的 XPath 表示式。

誠如「查詢 4」所述，XMLQUERY 函數通常用於 select 子句，以便從 XML 直欄中擷取 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
------	--------------

此查詢也突顯如何透過整合方式，使用 SQL/XML 來查詢 XML 及關聯式資料。select 子句會從關聯式及 XML 直欄擷取資料，而 where 子句則包含關聯式及 XML 述詞。我們可以使用 XMLTABLE 函數來表示相同的查詢，如「查詢五」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「查詢五」中，XMLTABLE 函數的 XQuery 表示式會找出在部門 V15 工作的員工，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」一樣。

- 查詢五

```
select d.unitID, T.name
from dept d, XMLTABLE(' for $emp in $d/dept[@deptID="V15"]/employee
                      return $emp'
                      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 方法的優點如下：

- 如果您已經有 SQL 應用程式，而且需要逐步新增一些 XML 功能，SQL/XML 會是理想的方法。

- 如果您是 SQL 的忠實支持者，且由於您和團隊最熟悉 SQL 語言，因此希望繼續使用 SQL 作為主要語言，SQL/XML 會是理想的選擇。
- 如果您的查詢要同時從關聯式直欄及 XML 直欄傳回資料，便適合使用 SQL/XML。
- 如果您的查詢需要全文檢索條件，如「查詢三」所示，便適合使用 SQL/XML。
- 如果您希望以集合方式傳回結果，而遺漏的 XML 元素則以空值表示，便適合使用 SQL/XML。
- 如果您要使用參數標記，SQL/XML 會是理想的選擇，因為 DB2 V9 XQuery 並不支援外部參數。XMLQUERY、XMLTABLE 及 XMLEXISTS 中的傳遞機制，可讓您將 SQL 參數標記當作變項 (\$x) 傳入內含的 XQuery 表示式：

#### ● 查詢六

```
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，SQL/XML 會是理想的選擇，讓您可以最方便的方法，來合併 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，其中一個員工是事業單位表格中的管理者。此查詢會合併關聯值 (unit.manager) 和 XML 值 (/employee/name)：

#### ● 查詢七

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

為了完成這項合併動作，我們將事業單位管理者傳入 XMLEXISTS 述詞，因此實際的合併條件是 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，因此合併條件就是 SQL 述詞：

#### ● 查詢八

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

由於 XMLCAST 函數必須是單一輸入值，因此「查詢七」通常會比「查詢八」好用。我們的範例就不適用，因為部門有多個員工。帶有 XMLCAST 的「查詢八」對合併 XML 值來說十分有用，由於這種合併可以使用 unit.manager 的關聯式索引，因此每份文件只會發生

一次。「查詢七」則無法使用該索引，因為合併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 語言並未提供明確分組 (group-by) 建構。雖然在 XQuery 中可以使用自行合併來表示分組和聚集，但相當不便。舉個例子，如要計算以辦事處分組的員工，即每個辦事處的員工人數。「查詢九」顯示如何使用純 XQuery 來執行此項查詢。「查詢九」的 db2-fn:xmlcolumn 函數可存取 DB2 中的 XML 資料。此函數會加入 XML 直欄的名稱作為引數，然後傳回該直欄所儲存的一連串 XML 值。使用 XMLTABLE 或 XMLQUERY 等 SQL/XML 函數來擷取 XML 直欄中的資料項目比較容易，然後再用熟悉的 SQL 概念來表示分組和聚集。「查詢十」會傳回跟「查詢九」相同的邏輯結果，但使用的是 XMLTABLE 及 SQL Group By 子句。

- 查詢九

```
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

```
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」中，XMLTABLE 函數會從每份有 "emp" 及 "office" 直欄的表格文件中，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，因為 SQL 的分組函數還會針對 NULL 值產生一個群組，而在我們的範例中，有一個員工是並沒有辦事處資訊。「查詢九」並沒有為該員工產生一行，因為 for 迴圈會疊代正確無誤的辦事處值（不含遺漏辦事處資訊的值）。

## SQL/XML 的缺點

- 若要將 XML 文件轉換成另一 XML 文件，SQL/XML 未必是最好的選擇。使用獨立式的 XQuery 往往會比較適當，且如果只需處理 XML 資料時也會比較直觀。
- 用 SQL/XML 表示兩個 XML 直欄之間的合併，或更常見的兩個 XML 值，這種作法可能較難處理，用純 XQuery 則會比較有效率且直觀。例如，「查詢十一」會合併部門及專案表格的 XML 直欄，以傳回負責任一專案的員工。

- 查詢十一

```
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「查詢十一」，我們將所有部門及專案文件傳入 XQuery 的 XMLEXISTS，並在此表示合併條件。您會發現，用純 XQuery（查詢十三）撰寫這種合併比較容易且有效率。

## XQuery 作為單獨語言

先回過頭來想想：何謂 XQuery？為什麼需要用到 XQuery？SQL 是專門用於關聯式資料模型的查詢語言，而 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) 且未經排序，而 XML 資料經過排序、呈巢狀結構、選用類型，且往往屬非正規及半結構化。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、操作及疊代一連串 XML 項目，並建構新的 XML 資料。

和 SQL 一樣，IBM 已擴充所有 DB2 主要應用程式設計介面 (API)，來支援 XQuery 作為第一類語言，其中包括 CLI/ODBC、Embedded SQL、JDBC 及 .NET。因此，DB2 指令行處理器也支援 XQuery，您可以直接送出 XQueries，不必修改，但必須以 XQUERY 關鍵字開頭，以通知 DB2 使用 XQuery 剖析器，如以下範例所示：

- 查詢十二

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

「查詢十二」會疊代每份部門文件的每個 "dept" 元素，然後傳回在部門 PR27 工作的員工姓名。

## XQuery 的優點

- 對於不需（或不想）使用 SQL 或關聯式結構的純 XML 應用程式，便適合使用 XQuery。
- XQuery 適合用來將純 XML 資料庫移轉到 DB2 V9。只要稍微變更現有 XQueries，通常即可在 DB2 中執行。例如，XQuery 的輸入資料來自 DB2 中的 db2-fn:xmlcolumn() 函數，而其他資料庫可能稱之為 collection()。在這種情況下，只要簡單的重新命名即可。
- 如果查詢結果需要嵌入（並傳回）新建的 XML 文件（有別於資料庫中的其他文件），XQuery 會是理想的選擇。
- XQuery 非常適合用來表示兩個 XML 文件之間的合併，也適用於合併 XML 值。例如，您可以使用「查詢十三」的 XQuery，以更直接、更有效的方式來表示「查詢十一」中的合併。

### ● 查詢十三

```
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，則無法利用 DB2 Net Search Extender (NSE) 所提供的全文檢索功能，必須使用 SQL 才能進行全文檢索。
- 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 並無法以參數標記來呼叫獨立式 XQuery。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。比方說，在「查詢十二」中，您可能想用問號 (?) 作為 SQL 式的參數標記，而不是文字值 "PR27"，但這樣就會變成無效查詢。

在「查詢六」中，SQL/XML 可讓您將 SQL 參數標記當作變項，傳入內含的 XQuery 表示式。SQL/XML 查詢的 SELECT 子句通常有 XMLQUERY 函數，以擷取 XML 文件的組成部分，而 WHERE 子句的 XMLEXISTS 述詞則過濾合格文件。如果您想以單一 FLWOR

表示式來表示整個查詢邏輯，不使用兩個獨立 XQuery 呼叫（一個以 XMLQUERY，另一個以 XMLEXISTS），而且還要使用參數標記，請考慮將「查詢十二」改寫成「查詢十四」。

- 查詢十四

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

「查詢十四」是 SQL/XML 陳述式，因為這就是 SQL fullselect VALUES 子句。VALUES 子句會針對結果表格中的每個直欄指定表示式，以傳回值表格。「查詢十四」的結果表只有一個 XML 類型的列和直欄，而 XMLQUERY 函數則會產生輸出直欄的值。所有員工姓名會以單一 XML 值傳給用戶端。「查詢十四」的 FLOWR 表示式幾乎跟「查詢十二」一樣，除了「查詢十四」包括外部變項 (\$z) 之外，此變項會當作參數傳入 XMLQUERY 函數。

## 內含 SQL 的 XQuery

單用 XQuery 即可讓您存取 XML 資料，但只限 XML 資料。如果您只要處理 XML 資料，這個方法非常理想，但如果您的應用程式需要整合存取 XML 及關聯式資料，光靠 XQuery 無法發揮兩個語言及資料模型的完整功能。若使用本文稍早提到的 SQL/XML，在 SQL 中內含 XQuery，即可完成此作業。反之，在 XQuery 中內含 SQL 也可以有其他用途。除了上一節所提到的優缺點，下列各項也很重要：

## 內含 SQL 的 XQuery 優點

- 如果您只要根據關聯式直欄的條件處理 XML 文件子集，內含 SQL 的 XQuery 會是理想的選擇。此選項能只將 XQuery 套用到 XML 直欄中的 XML 文件子集，相當有用。特別是，您可以使用關聯式述詞，將輸入限制為特定 XQuery。為此，DB2 提供了另一個輸入函數 db2-fn:sqlquery，可呼叫 XQuery 中的 SQL 查詢。此函數採用 SQL SELECT 陳述式，然後傳回 XML 直欄作為輸出。例如，「查詢十五」不考慮 XML 直欄 deptdoc 中的所有文件，但包含一個內含 SQL 的陳述式，可透過合併套用述詞的事業單位表，預先過濾 XML 文件：

- 查詢十五

```
XQUERY  
for $emp in db2-fn:sql query("select deptdoc  
                             from dept d, unit u
```

```

where d.uni tID=u.uni tID and
u.manager = 'Jim Qu' )/dept/empl oyee
where $emp/offi ce = 216
return $emp/name;

```

在兩個表格中的 **unitID** 直欄，以及事業單位表中管理者直欄的一般關聯式索引，可加速內含 **SQL** 的查詢處理。

- 內含 **SQL** 的 **XQuery** 可讓您利用全文檢索，因為您可在內含 **SQL** 陳述式的 **WHERE** 子句中使用文字搜尋函數 "contains"。在「查詢十六」中，只要部門表格中的 /dept/description 出現 "sales" 一字，**XQuery** 內的 **SQL** 陳述式便會選取文件。全文索引會快速找出這些 **XML** 文件，然後輸入 **FLWOR** 表示式，以便從這些文件中擷取所有員工姓名。根據我們的表格範本，「查詢十七」會傳回相同結果，但卻使用了 **SQL/XML** 表示法。

#### ● 查詢十六

```

XQUERY
for $emp in db2-fn:sql query("
    select deptdoc from dept
    where CONTAINS(deptdoc, 'SECTION("/dept/description"
""sales" ')=1
    )//employee
return $emp/name;

```

#### ● 查詢十七

```

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

```

- 對於需要整合關聯式及 **XML** 資料的應用程式來說，內含 **SQL** 的 **XQuery** 就很有用。您可以整合查詢 **XML** 資料及關聯式資料，而 **SQL/XML** 也是一樣。但您可能會覺得，如要合併 **XML** 及關聯式值，使用帶有 **XMLEXISTS** 函數的 **SQL/XML** 會比較容易。比較查詢十八及查詢十九「查詢十八」會傳回部門員工中包含事業單位管理者的 **deptID**。內含 **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。「查詢十九」以 **SQL/XML** 表示法表示相同的合併。

#### ● 查詢十八

```

XQUERY
for $m in db2-fn:sql query('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);
```

- 查詢十九

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

- 誠如「查詢十八」所示，內含 SQL 的 XQuery 適合用來將關聯式資料傳入 XQuery，讓您可以合併 XML 及關聯式資料。「查詢二十」會建構包含事業單位及部門資訊的結果文件。部門資訊是一份 XML 文件，擷取自 XML 直欄 deptdoc；事業單位資訊則來自關聯式表格單位。內含 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，即直欄 unitID、姓名及管理者的，其值則是來自事業單位表的關聯式直欄。

- 查詢二十

```
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」的輸出結果如下：

```
<result>
  <units><unit>
    <UNITID> WWPR </UNITID>
    <NAME> World Wide Marketing</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>
```

- 內含 SQL 的 XQuery 是理想的選擇，因為內含 SQL 陳述式可讓您呼叫使用者定義函數 (UDF)。許多 IT 組織都廣泛使用 UDF 來封裝重要的商業邏輯，並且簡化應用程式開發需求。由於純 XQuery 無法呼叫 UDF，所以此功能就顯得很重要。

## 內含 SQL 的 XQuery 缺點

- 雖然 db2-fn:sqlquery 可以在 XQuery 中內含 SQL 陳述式，但目前還不能在內含 SQL 陳述式中加入參數標記。
- 目前，db2-fn:sqlquery 並不允許從 XQuery 傳送參數到 SQL。因此，並不如 SQL/XML 那樣強大，無法合併 XML 及關聯式資料。例如，內含 SQL 的 XQuery 無法像 SQL/XML 一樣，用來表示關聯式與 XML 直欄之間的合併。

## XML 查詢結果

請務必注意，使用不同方式撰寫特定查詢，DB2 可能會提供不同格式的查詢結果。例如，純 XQuery 會以結果集方式傳回項目（如元素或文件片段），且每一列一個項目，即使多個項目均是來自資料庫的同一文件（列）。另一方面，使用 XMLQUERY 時，SQL/XML 可能會在一列中傳回多個項目，而非使用個別列。依據不同的應用程式，只有某些情況才需要使用此種方式，其他時候則不需要。請參考以下範例。

「查詢二十一」及「查詢二十二」都要求部門文件的員工姓名。「查詢二十一」是以 SQL/XML 表示，「查詢二十二」則以 XQuery 撰寫。

- 查詢二十一

```
XMLQUERY db2-fn: xml col umn("DEPT. DEPTDOC")/dept/employee/name;
```

- 查詢二十二

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

「查詢二十一」會針對每份部門文件傳回一列，而每列包括在該部門工作的員工姓名。

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

應用程式通常比較容易處理「查詢二十二」的結果，即一次一個 XML 值。不過，在此情況下，您並不知道哪一個名稱是來自相同的部門文件。「查詢二十一」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，因為需要先分割這些結果列，才能個別存取名稱。如果應用程式使用 XML 剖析器來處理 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，因為文件的格式錯誤（缺乏單一根元素）。若要解決此問題，您可將 XMLELEMENT 建構子新增到「查詢二十一」來加入單一根元素，如「查詢二十三」所示：

- 查詢二十三

```
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，讓每個結果列都是格式正確 XML 文件：

<employees><name>Jim Qu </name><name>Peter Pan </name></employees>
<employees><name>Matt Foreman</name></employees>
....

- 查詢二十四

```
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 V9 提供了一系列豐富的 XML 資料查詢選項，您可根據應用程式的需求及特性，選擇最適用的選項。如果您需要整合關聯式及 XML 資料，那麼多數情況下，SQL/XML 是最佳選擇，更何況，SQL/XML 還可以針對 XML 資料使用參數標記。如果您用的是純 XML 應用程式，獨立式 XQuery 就是功能強大的選項，而且可以用內含 SQL 加強，以進行全文檢索及呼叫 UDF。本文探討的範例可協助您制定明智的決策，讓您使用本文的查詢模式，著手撰寫用於 XML 資料的查詢。

這些一般準則旨在介紹您真正需要的查詢知識，不至於太過艱澀。例如，內含 SQL 陳

述式的 XQuery，其 SQL 陳述式當然可能再包含內含 XQuery，以此類推。但根據我們的經驗顯示，在表示所需的查詢邏輯時，納入兩種語言的層級通常不需超過一層。因此，我們建議，SQL 中只使用一個層級的內含 XQuery，或者是 XQuery 中只使用一個層級內含 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。

## 作者簡介

	<p>Nicola 博士是 IBM 矽谷實驗室的 XML 資料庫效能技術主管，工作重點在 DB2 XML 效能的各個面向，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、導入及最佳化 XML 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。也曾針對分散式及抄寫資料庫，進行相關研究及產業專案達四年之久。1999 年，從德國阿亨 (Aachen) 技術大學取得電腦科學博士學位。</p>
	<p>Ozcan 博士從 2001 年開始便是 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電機工程學系的電腦科學博士學位，研究興趣包括 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。</p>