

新 DB2 9.5 版 pureXML 加強功能概觀

新 DB2 9.5 版 pureXML 的功能與加強功能

級別：中級

Manoj Sardana (msardana@in.ibm.com)，IBM 系統軟體工程師

2007 年 11 月 1 日

本文介紹 IBM® DB2® 9.5 版 pureXML™ 加強功能，以及 Linux、Unix 與 Windows 的新功能。DB2 第 9 版支援 XML 為原生資料類型，也支援 SQL/XML 與 XQuery 語言以及綱目支援、發佈功能、公用程式的 XML 支援、分解這類其他功能。DB2 9.5 版不僅加強部分功能，還導入一些新功能，提高 XML 處理的效率。本文將循著如何提高 XML 處理效率，還有如何更容易使用的脈絡，來介紹這些加強功能與新功能。

DB2 9 XML 支援概觀

DB2 9 導入原生的 XML 資料類型，透過剖析過的階層式（原生）格式來儲存 XML，而且允許使用者運用 XQuery 與 SQL/XML 語言來查詢資料。DB2 XQuery 表示式使用儲存於 DB2 資料庫的 XML 文件，做為查詢的 XML 來源。而函數 `xmlcolumn` 與 `sqlquery` 則用來連接儲存於資料庫的 XML 值，XQuery 剖析程式並提供了 XML 順序。

除了 XQuery 語言，DB2 9 還提供 SQL/XML 函數，讓您在單一查詢中同時處理 XML 與關聯式資料。SQL/XML 函數 `xmlquery`、`xmltable` 與 `xmlexists` 有助於在 SQL 陳述式內嵌 XQuery。

此外，DB2 9 也支援綱目驗證，導入可向資料庫登錄綱目的新指令與儲存程序，讓這些綱目能夠像資料庫物件般運作。您可以在使用 `xmlvalidate` 函數插入作業前或後，使用這些已登錄的綱目驗證 XML 值。將 XML 分解成關聯式表格時，也可以註釋這些綱目。而 `xmlelement` 與 `xmlattributes` 之類的發佈函數，則可以用來將關聯值轉換成 XML 文件。一些公用程式（匯入、匯出等等）也已更新，可支援 XML 資料。如需更多有關第 9 版 XML 支援的詳細資料，請參閱[資源](#)一節。

DB2 9.5 版 - 最新消息

在處理與使用 XML 資料方面，現有的 DB2 9 功能已相當強大。DB2 9.5 版加強部分現有功能，同時導入其他功能，讓 XML 處理能力更加強大且更有效率。下列功能是本文的說明重點：

- 支援非 Unicode 資料庫中的 XML
- 次文件更新
- 基本表格 (Base-table) 儲存/壓縮
- 相容的 XML 綱目升級
- 驗證觸發程式
- 驗證檢查限制
- XML 抄寫
- XML 聯合
- XML 載入
- sqlquery() 參數
- 友善使用者的發佈功能
- SQL/XML 函數的預設參數傳遞
- XSLT 函數
- XML 分解加強功能
- XML 索引加強功能
- 索引顧問與最佳化工具加強功能
- DB2 資料 Web 服務

下文的程式碼範例大都援引自 DB2 9.5 版的範例資料庫。您可以在 DB2 9.5 版的指令行處理器上執行指令 **db2sampl**，以建立範例資料庫；或者，您也可以透過 **First Step** 來建立。*First Step* 是 DB2 所提供的工具，您可以在安裝 DB2 後執行，也可以稍後再執行，方法是在 Windows 跳至開始> 所有程式 > IBM DB2 > db2 版本名稱 > Set up tools > first steps。

支援非 Unicode 資料庫的 XML

DB2 9 只容許使用者採用 UTF-8 字碼頁格式的 XML 資料來建立資料庫，這意味著即使 XML 文件中的值是 ASCII 格式，它還是必須儲存在 UTF-8 資料庫。DB2 9.5 版已解決這個限制，現在它容許使用者採用各種字碼集格式的 XML 欄來建立資料庫。由於不再有限制，因此即使並非採用 UTF-8 建立的資料庫，使用者也可以透過新增 XML 欄的方式來改變表格，或者建立含 XML 欄的新表格。

以下字碼建立了範例資料庫以及含 XML 欄的範例表格：

清單 1：含預設字碼頁的資料庫

```
db2 CREATE DATABASE sampl edb
db2 CONNECT TO sampl edb
db2 CREATE TABLE record(id INT, record XML)
```

次文件更新

DB2 9.5 版可讓使用者為儲存於資料庫的 XML 文件更新部分內容，導入使用四個更新表示式（`insert`、`delete`、`replace` 與 `rename`）的 XQuery 轉換表示式，來修改 XML 文件片段。轉換表示式為 XQuery 語言的一部分，因此可用於 XQuery 表示式。在 DB2 9.5 版，更新表示式只能用於轉換表示式，而轉換表示式則含有下列子句：

- **COPY**：轉換表示式的 `copy` 子句連結了變數與來源 XML 值。更新表示式會在查詢中進一步處理這個 `copy`。
- **MODIFY**：`modify` 子句會按照更新表示式來修改複製的 XML。MODIFY 子句可以使用多個更新表示式。
- **RETURN**：`return` 子句會傳回修改值。

四個更新表示式說明如下：

1. `insert` 表示式用來將新的 XML 節點插入現有的 XML 文件。您可以指定 XML 文件中的插入點。
2. `replace` 表示式用來更新特定節點的特定值。
3. `delete` 表示式用來刪除 XML 文件的特定節點。
4. `rename` 表示式則用來重新命名節點。

由於轉換表示式為 XQuery 語言的一部分，因此可以用於採用 `xmlquery` 函數的 SQL 陳述式，也可以用於用來更新 XML 值的更新陳述式。

清單 2 中的程式碼，將更新範例資料庫中客戶表格的 `info` 欄，會更新 XML 文件，使 `CID` 屬性與 `cid` 關聯欄的值相符。

清單 2：含 UPDATE 表格的轉換表示式

```

UPDATE CUSTOMER
SET info =
  XMLQUERY(' transform
            copy $po := $INFO
            modify
              do replace value of $po/customerinfo/@Cid with $CID
            return $po'
            passing info as "INFO", cid as "CID")
WHERE cid=1000

```

如果表格中的 XML 欄存在 XML 驗證檢查限制，則使用者手動更新或建立觸發程式更新之前，可能要先驗證新的 XML 值。

以下程式碼範例會從採購單表格移除項目，並且提供修改過的文件做為查詢結果。

清單 3：轉換表示式

```

xquery
transform
copy
  $po := db2-fn:sql query(' select porder from purchaseorder where
                          custid = 1002 and orderdate="2006-02-18"')
modify
  do delete $po/ PurchaseOrder/item[partid = "100-201-01"]
return $po

```

範例 xupdate.db2 提供各種不同的轉換表示式範例。這個範例位於 `sqlib/samples/xml/xquery/clp` 目錄。

基本表格列儲存/壓縮

在 DB2 9 中，XML 資料的儲存位置和關聯式資料不同，其儲存位置為 XML 資料區 (XDA)。DB2 9 會將全部 XML 文件儲存於這個儲存位置，也就是說，一起存取 XML 值與關聯式資料時，需要更多的 I/O。如果 XML 文件不大，而且儲存關聯值後頁面還有足夠的空間，就可以同時容納 XML 值，在相同頁面儲存 XML 有效能上的優點，例如：

- 壓縮：XML 資料與關聯式資料一起儲存時，您可以使用導入 DB2 9 的壓縮技術來執行壓縮。相較於關聯式資料，XML 值的檔案大小較大，壓縮後可節省不少空間。
- 查詢效能：列入 XML 資料後，基本表格會比平時更大，因為 XML 資料與關聯式資料儲存於同一個位置。如果存取 XML 資料和存取表格中其他關聯式資料的頻率相當，查詢資料時即可發揮優點。

DB2 9.5 版導入 XML 資料的基本表格列儲存功能。這意味著，如果每一列的關聯式資料與 XML 資料並未超出頁面大小，就可以在相同的實體頁面同時儲存 XML 資料與關聯式資料。唯有記錄的大小總計並未超出頁面大小時，才能執行 XML 資料的基本表格列儲存。假如大小超過，則一樣會將 XML 資料儲存於 XML 儲存位置 (XDA)。DB2 的頁面大小上限為 32KB，因此 XML 值行內長度的長度上限為 32KB。如果文件內部樹狀結構表示法的大小小於指定行內長度，則即可立即列入。清單 4 中的程式碼，建立了含 XML 資料基本表格列儲存的表格：

清單 4：XML 資料的基本表格儲存

```
db2 CREATE TABLE emp1(id INT, info XML INLINE LENGTH 1024)
```

使用 `INLINE` 選項指定，將 XML 資料與關聯式資料一起儲存。可以在同一個位置找到所有資料時，有助於提取 XML 資料的查詢。另一方面，如果是存取非 XML 資料的查詢，可能需要更多的 I/O 才能尋找關聯式資料。

表格只有一欄是 XML 類型，而且 XML 文件的最大大小並未超出頁面大小時，最適合使用 XML 資料基本表格列儲存。

相容的 XML 綱目升級

爲了提高彈性並且提供最佳的綱目升級，DB2 9.5 版爲 XML 綱目導入更新功能。如果先前已登錄的綱目與新綱目相容，則可以使用新綱目來更新已登錄的綱

目。假如使用較舊綱目進行驗證的 XML 文件，仍可以用新綱目進行驗證，則表示新舊綱目之間是相容的。

例如，通過舊綱目驗證的 XML 文件，增加選用性新元素後仍然有效，因為新元素的本質屬於選用性。新的 XML 文件可具備這個選用元素，還可使用新綱目進行驗證。綱目更新後，舊文件仍有效，因此您無須執行任何動作。如果綱目不相容，則綱目更新就會失敗，並且保留舊綱目的註釋與 ID。

若要更新綱目，必須導入 XSR_UPDATE 儲存程序。儲存程序會檢查相容性，並且只會在綱目相容時，才會更新綱目。若要更新綱目，使用者必須分別登錄綱目，並且呼叫 XSR_UPDATE 儲存程序。舊綱目順利更新後，可由使用者選擇保留或捨棄。

接下來，以客戶表格中的 info 欄為例。info 欄擁有含以下定義的 addr 元素。（您可以在 `sqllib/samples/db2sampl` 目錄找到完整綱目）。

清單 5：舊的 XML 綱目定義

```
<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema"
            name="addr" minOccurs="1" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="street" type="xs:string" minOccurs="1" />
      <xs:element name="city" type="xs:string" minOccurs="1" />
      <xs:element name="prov-state" type="xs:string" minOccurs="1" />
      <xs:element name="pcode-zip" type="xs:string" minOccurs="1" />
    </xs:sequence>
    <xs:attribute name="country" type="xs:string" />
  </xs:complexType>
</xs:element>
```

未來，使用者希望採用選用元素 `HouseNo`。為了更新已登錄的綱目以保持 ID 不變，您必須先登錄含額外元素的新綱目。新的 `addr` 元素其定義如下：

清單 6：新的 XML 綱目定義

```

<xs:element xmlns:xs="http://www.w3.org/2001/XMLSchema"
            name="addr" minOccurs="1" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="HouseNo" type="xs:string" minOccurs="0" />
      <xs:element name="street" type="xs:string" minOccurs="1" />
      <xs:element name="city" type="xs:string" minOccurs="1" />
      <xs:element name="prov-state" type="xs:string" minOccurs="1" />
      <xs:element name="pcode-zip" type="xs:string" minOccurs="1" />
    </xs:sequence>
    <xs:attribute name="country" type="xs:string" />
  </xs:complexType>
</xs:element>

```

完成登錄之後，就可以使用以下儲存程序，將現有綱目更新成新綱目：

清單 7：使用 XSR_UPDATE 更新綱目

```
db2 call XSR_UPDATE('RSHEMA','CUSTOMER','RSHEMA','CUSTOMER1',0)
```

最後的引數值 0，表示完成更新後不應捨棄新綱目。如果該值並非設定為零，則完成更新作業後會捨棄新綱目。

範例程式碼 *xsupdate.db2* 示範相容的 XML 綱目升級。您可以在 `sqlib/samples/xml/clp` 目錄中找到這個範例。

驗證觸發程式支援

爲了提高應用程式的彈性，並且提供使用者傳入 XML 文件自動驗證功能，DB2 9.5 版延伸了前觸發程式的 XML 支援。前觸發程式是使用 **BEFORE** 選項所建立，而且於 **inster/update/delete** 作業之前執行。在前觸發程式，新變數中可以參照 XML 值。觸發程式的動作可以在新值套用 **xmlvalidate** 函數。觸發程式的 **WHEN** 子句則可依任何指定的綱目，檢查新值是否已驗證。您也可以使用 **WHEN** 條件中的 **IS VALIDATED** 或 **IS NOT VALIDATED ACCORDING TO XMLSCHEMA** 子句來檢查。視 **WHEN** 條件的輸出而定，您可決定要驗證 XML

值或設定新值。目前，新的 XML 類型轉移變數中，只允許使用 `xmlvalidate` 函數。這個觸發程式一旦建立之後，每當表格出現插入時就會自動啟動與執行，因此如果在插入陳述式中並未驗證 XML 值，即可使用這個觸發程式來進行驗證。

以下程式碼為 DDL 陳述式，可以建立客戶表格，並且在表格建立定義的觸發程式。每當表格出現插入作業時，就會啟動這個觸發程式。如果在使用 `xmlvalidate` 函數的 `insert` 陳述式中並未驗證 XML 文件，這個觸發程式就會在執行插入作業之前先驗證 XML 文件。這個範例假設表格客戶不存在，但資料庫中已登錄客戶綱目。

清單 8：表格中定義的觸發程式

```
CREATE TABLE Customer ( Cid          BIGINT NOT NULL,
                        info          XML,
                        History       XML,
                        CONSTRAINT PK_CUSTOMER PRIMARY KEY (Cid))

CREATE TRIGGER Trigger_customer NO CASCADE BEFORE INSERT ON customer
  REFERENCING NEW AS n
  FOR EACH ROW MODE db2sql
  WHEN (n.info IS NOT VALIDATED ACCORDING TO XMLSCHEMA ID customer)
  BEGIN ATOMIC
    SET n.info = XMLVALIDATE(n.info ACCORDING TO XMLSCHEMA ID customer);
  END@
```

範例 `xmltrig.db2` 提供不同的實務練習與作業，皆可在觸發程式內執行，以指派新值並且驗證 XML 值。您可以在 `sqlib/samples/xml/clp` 目錄中找到這個範例。

XML 驗證檢查限制

檢查限制是一種限制類型，建立表格時可以置於表格欄。只要限制有效，DB2 就允許執行插入作業，反之則會插入失敗。

DB2 9.5 版支援 XML 值的檢查限制。使用者可以使用檢查限制，在 XML 欄強制執行驗證。類似於前觸發程式，您也可以檢查限制中使用 `IS VALIDATED ACCORDING TO XMLSCHEMA` 子句來強制執行驗證。唯一不同

之處在於，這個限制只會檢查驗證條件，不會實際執行驗證。因此，使用者需要在 `insert` 陳述式中使用 `xmlvalidate` 明確驗證 XML 值，或者使用前觸發程式執行自動驗證。根據檢查限制中指定的綱目，XML 值有效時，插入作業才會成功。

前觸發程式與表格 XML 值的檢查限制可確保 XML 值符合指定綱目。雖然只要發行插入作業，前觸發程式就會自動驗證，檢查限制會強制使用者確實使用 `xmlvalidate` 函數。這兩者可以搭配使用，以確保 XML 值的完整性。

清單 9 中的程式碼改變清單 8 所建立的表格客戶，以在表格放置檢查限制：

清單 9：檢查限制

```
db2 ALTER TABLE customer ADD CONSTRAINT check_info
      CHECK(info IS VALIDATED ACCORDING TO XMLSCHEMA ID customer)
```

以上所建立的檢查限制，一律會根據客戶綱目驗證文件。如果觸發程式不存在，使用者就需要使用 `xmlvalidate` 函數來明確驗證值。

範例程式碼 `xmlcheckconstraint.db2` 示範，使用者如何在結構相同且含檢查限制的不同表格建立視圖，並且依綱目實作表格切割。

XML 抄寫支援

DB2 9.5 版可將 XML 資料抄寫至其他支援 XML 資料的資料庫。您可以使用 WebSphere® Replication Server 9.5 版或 WebSphere Data Event Publisher 9.5 版進行抄寫。WebSphere Replication Server 可將資料抄寫至支援 XML 資料類型的聯合目標，或者將 XML 對映至 CLOB/BLOB 欄。

XML 無異於其他關聯欄，會在交易訊息中執行資料抄寫，因此所抄寫的 XML 大小受限於交易訊息長度上限。假如是大型資料，可插入位置保留元文件取代原始文件。另外，也可以在異常狀況表插入異常狀況。

執行抄寫時無法抄寫 XML 綱目登錄，也無法驗證 XML 資料。

XML 聯合支援

WebSphere Federation Server 9.1 版支援 pureXML，因此可以用來整合本端與遠端的 XML 儲存資料。來自不同資料庫的 XML 資料可以當成本端資料檢視，而查詢資料時，則可以套用 DB2 XQuery 與 SQL/XML。您可以在遠端聯合資料庫建立視圖，將資料當成序列化字串檢視，之後則可以在 WebSphere Federation Server，將資料再次剖析成 XML 值。現在，使用為該視圖所建立的暱稱，DB2 即可透過 SQL/XML 與 XQuery 語言查詢這個資料。

您可以使用 `db2 xmlvalidate` 函數來驗證來自不同聯合來源的 XML 資料，方法和驗證本端 XML 值完全相同。

如需如何使用 WebSphere Federation server for pureXML 的相關資訊，請參閱 [資源](#) 一節。

載入支援

針對在表格中填入 XML 值，DB2 9 主要支援兩種方式，分別是使用 `insert` 陳述式將 XML 值插入表格，以及使用匯入公用程式將大量資料匯入表格。

DB2 9.5 版支援延伸後，可支援載入公用程式。XML 資料匯入功能所支援的選項，大部分也支援載入功能。您可以使用 `XML FROM` 子句來指定 XML 資料的路徑，並且透過 `XMLVALIDATE USING` 子句，在載入期間驗證 XML 資料。至於其他的選項則包括 `XDS`、`SCHEMA` 與 `SCHEMALOCATION HINTS`。指定 `XDS` 選項時，可以使用 `DEFAULT`、`IGNORE` 與 `MAP` 子句。這些選項的意義和匯入選項相同。您可以使用檔案類型修飾元 `XMLCHAR` 與 `XMLGRAPHIC` 來指定資料的字碼頁。在資料檔案，會使用 XML 資料指定元 (`XDS`) 來指定 XML 值。載入的重新啟動行為不會改變，會透過掃描全部 XML 文件來重建索引。

範例程式碼 `xmlload.db2` 示範，DB2 9.5 版中適用於 XML 資料的載入選項。您可以在 `sqllib/samples/xml/clp` 目錄中找到範例。

XSLT 支援

DB2 9.5 版提供一個功能，可在資料庫本身使用 XSL 轉換來處理 XML 文件，套用 XSLT 樣式表，即可將 XML 文件轉換成 HTML 格式。因此，DB2 9.5 版導入 `xslttransform` 函數，也支援使用參數的樣式表。`xslttransform` 函數可以為 XML 文件套用 XSLT 樣式表，這個樣式表以 XML 文件形式儲存於資料庫表格欄。這樣一來，使用者可以從資料庫擷取已轉換的 XML 文件，然後直接在 Web

呈現。

現在，假設您具備這類 XML 文件：

清單 10：XML 文件

```
<?xml version="1.0"?>
  <products xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <product pid="100-201-01">
      <description>
        <name>Ice Scraper, Windshield 4 inch</name>
        <details>Basic Ice Scraper 4 inches wide, foam handle</details>
        <price>3.99</price>
      </description>
    </product>
  </products>
```

以及此對應的 XSLT 樣式表：

清單 11：XSLT 樣式表

```
<?xml version="1.0" encoding="UTF-8"?><xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="products">
    <html >
    <head/>
    <body>
      <table border="1">
        <th>
          <tr>
            <td width="80">product ID</td>
            <td width="200">product name</td>
            <td width="200">price</td>
            <td width="50">details</td>
          </tr>
        </th>
      <xsl:apply-templates/>
```

```

</table>
</body>
</html >
</xsl:template>
<xsl:template match="product">
<tr>
<td><xsl:value-of select="@pid" /></td>
<td><xsl:value-of select="./description/name" /></td>
<td><xsl:value-of select="./description/price" /></td>
<td><xsl:value-of select="./description/details" /></td>
</tr>
</xsl:template>
</xsl:stylesheet>

```

這些文件必須儲存於表格，也可以當成參數傳遞。將值當成參數傳遞時，這些文件必須是格式正確的 XML 文件，參數的資料類型為 XML、VARCHAR、CLOB 或 BLOB。假設文件與樣式表皆儲存於表格，則您可以使用以下的陳述式來轉換 XML 文件：

附註：這個範例假設，表格中所儲存的文件名為 *xslt*，XML 文件的欄名為 *xmldoc*，而 XSL 文件為 *xsl doc*。

清單 12：XSLTransform 表示式

```

SELECT XSLTRANSFORM (description USING stylesheet AS CLOB (10M))
FROM product_details

```

這個查詢會輸出 HTML 文件，透過瀏覽器即可檢視文件，該 HTML 輸出如清單 13 所示：

清單 13：XSLTransform 表示式的 HTML 輸出

```

<html >
<head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>

```

```

<body>
  <table border="1">
    <thead>
      <tr>
        <th>
          <tr>
            <td width="80">product ID</td>
            <td width="200">product name</td>
            <td width="200">price</td>
            <td width="50">details</td>
          </tr>
        </th>
      <tr>
        <td>100-201-01</td>
        <td>Ice Scraper, Windshield 4 inch</td>
        <td>3.99</td>
        <td>Basic Ice Scraper 4 inches wide, foam handle</td>
      </tr>
    </thead>
  </table>
</body>
</html >

```

使用者也可以將 XML 文件與 XSLT 樣式表分別儲存於不同表格，然後合併表格，將單一 XSLT 樣式表套用至多個 XML 值。

發佈函數

發佈函數可用來將關聯式資料轉換為 XML 值。DB2 9 導入的 SQL/XML 支援，DB2 9.5 版已予以加強並簡化。xmlelement 這類的 DB2 9 SQL/XML 函數需要提供全部 XML 元素的名稱、屬性與其他節點，但這些元素的值不是衍生自表格的關聯欄，就是以明確的方式提供。有時候使用者需要產生 XML 值，但不需要元素名稱。

DB2 9.5 版導入 xmlrow 與 xmlgroup 等新函數，以延伸現有發佈函數。這些函數會從表格欄衍生 XML 元素的名稱與值，其中 xmlrow 輸出是以 XML 呈現的列值順序，而 xmlgroup 函數則會將全部的值集中至根節點。

以下是包含員工地址詳細資料的員工表格範例，記錄如下。

ID	姓名	街名	城市名	州名	國名
1	manoj	sector14	gurgaon	haryana	india

以下查詢將 `xmlrow` 與 `xmlgroup` 函數套用至這一系列。

清單 14：新的發佈函數

```
db2 SELECT XMLROW(id, name, street, city, state, country) FROM EMPLOYEE
<row>
  <ID>1</ID>
  <NAME>manoj </NAME>
  <STREET>sector14</STREET>
  <CI TY>gurgaon</CI TY>
  <STATE>haryana</STATE>
  <COUNTRY>i ndi a</COUNTRY>
</row>

db2 SELECT XMLGROUP(id, name, street, ci ty, state, country) FROM EMPLOYEE
<rowset>
  <row>
    <ID>1</ID>
    <NAME>manoj </NAME>
    <STREET>sector14</STREET>
    <CI TY>gurgaon</CI TY>
    <STATE>haryana</STATE>
    <COUNTRY>i ndi a</COUNTRY>
  </row>
</rowset>
```

在 **DB2 9** 中，若要獲得相同結果，您必須明確提供元素名稱，還必須將 `xmlelement` 套用至每一欄值。

範例程式碼 `xmlintegrate.db2` 針對這些函數提供更多更複雜的範例，同時還與 **V9.1** 版的相同查詢進行比較。您可以在 `sqlib/samples/xml/clp` 目錄中找到這個範例。

傳遞至 `sqlquery` 函數的參數

在 DB2 9，`sqlquery` 函數可用來在 XQuery 表示式內嵌 SQL 陳述式，這個函數取字串值做為輸出，是有效的 SQL 完整 `Select` 陳述式。在 DB2 9，從 XQuery 陳述式無法傳遞參數至這個函數，

DB2 9.5 版則加強此函數，導入全新參數函數，可取整數值為輸出。現在，`sqlquery` 函數可取多個引數為輸入，其中第一個引數為代表完整 `Select` 的字串值，後面接參數值。而 `sqlquery` 函數的第一個字串引數可以包含參數函數，其中在第一個強制字串引數之後，會以傳遞至這個函數的引數取代。傳遞至這個參數函數的整數值，會告知 `sqlquery` 函數呼叫中引數的位置，然後由這個函數呼叫取而代之。例如，`parameter(1)` 告知剖析器，使用字串之後的第一個引數來取代這個參數，引數的類型應該與完整 `Select` 預期的值類型相同，使用 `cast` 函數即可將值強制轉型為適當的類型。

以下是使用範例資料庫客戶表格的範例，範例資料庫可以透過 `db2sampl` 指令或 `First Step` 建立。`First Step` 是 DB2 所提供的工具，您可以在安裝 DB2 後執行，也可以稍後再執行，方法是在 Windows 跳至開始> 所有程式 > IBM DB2 > db2 版本名稱 > Set up tools > first steps。

客戶表格具備做為關聯欄的 `cid` 欄，以及代表客戶 ID 的鍵值。`info XML` 欄則具備再次代表客戶 ID 的屬性 `Cid`。如果資料一致，則屬性 `Cid` 值應該與特定列的 `cid` 欄值相同，而以下查詢會查看值一致的列數。`Cid` 屬性的值會傳遞至 `sqlquery` 函數，以比較它與關聯的 `cid` 值。

清單 15：傳遞參數至 `sqlquery` 函數

```
xquery declare default element namespace "http://posample.org";
for $i in db2-fn:xmlcolumn("CUSTOMER.INFO")/customerinfo/@Cid
for $j in db2-fn:sqlquery("select info from customer where cid=parameter(1)", $i)
return
<out>{$i} </out>
```

這個查詢將傳回所有資料一致的客戶 ID。

範例程式碼 `xqueryparam.db2` 提供傳遞一個與多個參數至 `sqlquery` 函數的範例。您可以在 `sqlib/samples/xml/clp` 目錄中找到這個範例。

現有函數 `XMLQuery`、`XMLtable` 與 `XMlexists` 的預設傳遞行爲

在 DB2 9，函數 `xmlquery`、`xmltable` 與 `xmlexists` 是用來在 SQL 陳述式內嵌 `Xquery` 陳述式。您可以使用這些函數的 `PASSING` 子句，將參數從 SQL 陳述式傳遞至這些函數。

在 DB2 9，如果相同的 SQL 陳述式中出現多個這類函數，則每一個都需要個別的 `PASSING` 子句，有時查詢結構會因此顯得複雜而龐大。DB2 9.5 版延伸這些函數後可使用預設傳遞機制，現在可以在這些函數中，使用欄名為 `Xquery` 的變數名稱。未使用明確的 `PASSING` 子句時，依預設 DB2 會傳遞變數的同一個欄，如此一來，查詢的規模縮小而且容易理解。以下程式碼提供根據範例資料庫表格的範例，查詢會提取 `Robert Shoemaker` 這位客戶採購單的第一個項目。

清單 16：SQL/XML 函數的預設傳遞行爲

```
SELECT XMLQUERY('declare default element namespace "http://posample.org";
                $PORDER/PurchaseOrder/item[1]' )
FROM purchaseorder AS p, customer AS c
WHERE XMLEXISTS('declare default element namespace "http://posample.org";
                $INFO/customerinfo[name="Robert Shoemaker" and @Cid = $CUSTID]')
```

如果是 `SELECT` 子句的 `xmlquery` 函數，依預設，會傳遞採購單表格的 `porder` 欄。同樣地，依預設也會傳遞 `xmlexists` 函數、客戶表格的 `info` 欄與 `custid` 欄。這些變數的名稱務必使用大寫字母，因為 `xquery` 是區分大小寫的語言，而且關聯欄名稱一律以大寫字母儲存。

XML 驗證限制

DB2 9.5 版加強用於 `SELECT` 陳述式的 `IS VALIDATED` 子句，加入 `ACCORDING TO XML SCHEMA ID`。現在，使用者可以提供多個綱目，並且只選取根據這些綱目進行驗證的 XML 值。DB2 9.5 版還以 XML 表示式取代欄為運算元，大幅提高彈性。下例僅從使用客戶綱目驗證的客戶表格選取文件。

清單 17：SELECT 陳述式中的 XML 驗證限制

```
db2 SELECT info FROM customer
      WHERE info IS VALIDATED ACCORDING TO XMLSCHEMA ID customer
```

註解的 XML 綱目分解

DB2 9 支援 XML 綱目註釋，因此資料可分解成關聯式表格，但遞迴綱目則屬例外。DB2 9.5 版則解除這個限制，現在即使是遞迴的綱目，使用者仍然可以註釋與分解資料。

DB2 9.5 版為延伸分解功能，提供了插入順序，希望將資料分解成多個擁有外部索引鍵關聯的表格時，這個功能就很重要。因此，最先應填入主要表格，以維持參照限制。您可以使用下列註釋指定插入順序：

清單 18：提供插入順序的綱目註釋

```
<db2-xdb: rowSetOperationOrder
xml ns: db2-xdb="http://www.ibm.com/xml ns/prod/db2/xdb1">
  <db2-xdb: order>
    <db2-xdb: rowSet>CUSTOMER</db2-xdb: rowSet>
    <db2-xdb: rowSet>PURCHASEORDER</db2-xdb: rowSet>
  </db2-xdb: order>
</db2-xdb: rowSetOperationOrder>
```

範例程式碼 `recxmldecomp.db2` 與 `xmldecomposition.db2`，分別針對註釋與分解遞迴綱目以及插入表格順序，提供了範例。您可以在 `sqlib/samples/xml/clp` 目錄中找到這個範例。

XML 索引加強功能

DB2 9 導入 XML 索引，可以在儲存於資料庫 XML 文件的特定節點建立 XML 索引。索引的資料類型可以是 `VARCHAR`、`DOUBLE`、`DATE` 或 `TIMESTAMP`。

如果索引的資料類型與 XML 文件的元素類型不符，DB2 會插入 XML 值，但不會為此特定的 XML 值建立索引。

DB2 9.5 版另外為 XML 索引導入子句 REJECT INVALID VALUES。如果使用這個子句建立索引，而且索引的資料類型與所插入的 XML 文件元素的資料類型不符，插入動作將會失敗。另一方面，如果插入值之後才建立索引，而且資料類型不相符，則索引建立將會失敗。

這是 DB2 9.5 版的預設行為，而且可以使用 IGNORE INVALID VALUES 子句明確指定。

下例將使用選項 REJECT INVALID VALUES，在客戶表格 Cid 屬性建立索引。

清單 19：XML 索引

```
db2 CREATE INDEX index1 ON customer(info)
      GENERATE KEY USING XMLPATTERN
      'declare default element namespace "http://posample.org";
      /customerinfo/@cid' as SQL DOUBLE REJECT INVALID VALUES
```

索引顧問與最佳化工具加強功能

索引顧問可以用來同時取得 XML 索引與關聯式資料索引的相關建議，如果同時使用 XML 索引與關聯式資料索引，可大幅提升使用者的效能。DB2 9.5 的最佳化工具使用兩類索引來最佳化查詢，並協助選擇執行查詢的最佳計劃。

DB2 Data Web 服務

DB2 9.5 版的 XML 資料也可以揭露成爲 Web 服務，適用於使用 Data Web 服務的資料庫操作作業。Data Web Services (DES) 公開插入、更新、選取與儲存程序這類 XML 作業爲 Web 服務。若要使用這些 Web 服務，可透過 SOAP over HTTP、POST 與 GET 方法，例如，透過 Web 瀏覽器或自訂用戶端。現有資料庫工具已整合的 Eclipse 型工具皆支援 Data Web 服務。

控制中心也更新了，以便處理 XML 資料。

總結

DB2 9 導入 XML 為新的資料類型，同時提供可處理 XML 值的基礎架構。另外還提供查詢 XML 文件、登錄綱目與驗證 XML 文件，使用 SQL/XML 在 SQL 與 XQuery 之間進行互動這類基本功能。DB2 9.5 版加強了現行功能，並且還提供更多功能，提高了處理 XML 資料的效率。

鳴謝

感謝 Matthias Nicola、Susan Malaya、Cindy Sirocco 與 Henrik Looser 為本文提供寶貴的技術性審核意見。另外，還要感謝 Mel 進行 ID 審核。

資源

Learn

- [What's new in DB2 Viper](#): Discover what's new in DB2 Viper
- [Query DB2 XML data with XQuery](#): Use XQuery to query your XML data
- [Get off to a fast start with DB2 Viper](#): Master the new XML features in DB2 Viper
- [pureXML in DB2 9: Which way to query your XML data?](#): See how to query your XML data
- [XML in a non-Unicode database](#): Find tips on how XML can be stored in non-Unicode databases
- [15 best practices for pureXML performance in DB2 9](#): View tips and tricks to manage your XML data with the best performance
- [Industry Standard Demos and downloads](#): Get a quick start to work with DB2 and industry standards XML schema
- [Update XML in DB2 9.5](#): Learn about XML update features in DB2 V9.5
- [Use WebSphere Federation Server Version 9.1 to integrate XML data](#): Learn about WebSphere Federation Server Version 9.1 to integrate XML data
- Browse the [technology bookstore](#) for books on these and other technical topics.

Get products and technologies

- Download [IBM product evaluation versions](#) and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

Discuss

- [Participate in the discussion forum.](#)
- Check out [developerWorks blogs](#) and get involved in the [developerWorks community](#).

作者簡介

Manoj 是 IBM 印度軟體實驗室 DB2 範例開發小組的應用程式開發人員，專精 DB2 pureXML 與 Java 應用程式開發。他是通過 IBM 認證的 DB2 9 應用程式開發人員、資料庫管理者，以及 XML 與相關技術的解決方案開發人員。他曾經撰寫過 DB2 Express-C 應用程式開發方面的 IBM 紅皮書。擔任目前職務前，他曾是功能驗證小組成員，負責測試 DB2 的公用程式與死鎖監控功能。