

第十四章 XML與資料庫

- 目的
- XML的語法
- DTD
- Schema
- XML文件處理
- 資料庫管理系統對XML的支援
 - ↳ XML轉入
 - ↳ XML轉出

©黃三益2007
資料庫的核心理論與實務第三版

14-1

目的

- XML的前身是HTML
- 都是標籤語言
- HTML的標籤大體上是為了呈現（Presentation）的目的，而不是用來標示內容

```
<html>
<B>黃三益</B>教的<l>資料庫</l>
</HTML>
```

 - 黃三益教的資料庫

```
<HTML>
Compaq N610C laptop <br>
<B>Spec</B>: Pentium 4-M 1.8 G CPU, 256 DDR, 30G HDD, 8X DVD<br>
<B>Price</B>: NT$70,000
</HTML>
```

 - Compaq N610C laptop
Spec: Pentium 4-M 1.8 G CPU, 256 DDR, 30G HDD, 8X DVD
Price: NT\$70,000

©黃三益2007
資料庫的核心理論與實務第三版

14-2

目的 (Cont.)

- XML的目的是標示內容，可以加入自訂的適當標籤

```
<laptop>
<name> Compaq N610C laptop</name>
<spec>
  <cpu> Pentium 4-M 1.8 G</cpu>
  <memory>256 DDR</memory>
  <hdd>30G</hdd>
  <dvd>8X</dvd>
</spec>
<price> NT$70,000</price>
<warranty term=" 6 months" restriction=" effective only in Taiwan" />
</laptop>
```

- 這些標籤無法事先定義，而必須隨著應用範圍的不同做不同的制訂
- XML（可擴充的標示語言）就是為滿足這樣的需求而誕生
- 已經變成B-C（Business to Customer）和B-B（Business to Business）電子商務的標準

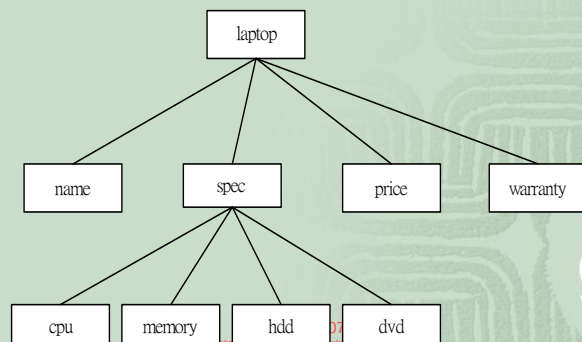
©黃三益2007
資料庫的核心理論與實務第三版

14-3

XML的語法

元素和屬性

- XML文件的資料組織方式是階層式的
- 每一個資料敘述稱為一個元素（Element）
- 一個元素包含一個包含啟始（**start**）標籤、內容和一個結束（**end**）標籤（參考[範例XML](#)）



14-4

XML的語法 (Cont.)

- 一個元素可以有一些附屬屬性和屬性值，這些屬性和屬性值是設定在啟始標籤裡

```
<laptop>  
<name> Compaq N610C laptop</name>  
<spec>  
  <cpu> Pentium 4-M 1.8 G</cpu>  
  <memory>256 DDR</memory>  
  <hdd>30G</hdd>  
  <dvd>8X</dvd>  
</spec>  
<price> NT$70,000</price>  
<warranty term=" 6 months" restriction=" effective only in Taiwan" />  
</laptop>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-5

XML的語法 (Cont.)

- 保留字元

☞ 五個保留字：**&**，**<**，**>**，**'**，和 **"**

☞ 需要用這些字時，表達方式如下：

- **&** : **&**;
- **<** : **<**;
- **>** : **>**;
- **'** : **'**;
- **"** : **"**;

☞ 範例：“ $A[i] > 5 \ \& \ A[i+1] < 10$ ”

```
<equation>  
A[i] &gt; 5 & A[i+1] &lt; 10  
</equation>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-6

XML的語法 (Cont.)

- 也可用<![CDATA[和]]>將其圈起來

```
<?xml version = "1.0" ?>
<book title = "c++ programming" >
<sample>
<![CDATA[
// c++ comment
if (this->getSalary(>30000 && value[0]!=3)
cerr << this ->displayError();
]]>
</sample>
</book>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-7

練習13-1

- 試著解釋以下XML片段的意義

```
<library_books>
<book>
<call_number> 423.21 8744 </call_number>
<title> 如何當個稱職的職場新鮮人</title>
<author>張大頭</author>
<author>李大炮</author>
<publish company = "想像出版社", year = "民93年", place = "台
北" />
<price>400</price>
</book>
</library_books>
```

- Ans:

圖書館有一本書，書名是” 如何當個稱職的職場新鮮人”，索書號
423.21 8744，作者是張大頭和李大炮，由想像出版社於民國93年在台
北發行，定價為400。

©黃三益2007
資料庫的核心理論與實務第三版

14-8

命名空間

- 不同領域的名稱相同的標籤卻可能有不同的意義，XML因此有命名空間（namespace）的概念
- 考慮下例的XML文件裡兩個命名空間：college和kindergarten

```
<?xml version = "1.0" ?>
<directory xmlns:college = "http://www.mis.nsysu.edu.tw/db-
book/college" xmlns:kindergarten =
"http://www.mis.nsysu.edu.tw/db-book/kindergarten" >
<person>
<name>張三</name>
<college:student major = "資訊管理" />
<kindergarten:student years=2/>
</person>
</directory>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-9

DTD

- 最早被使用的XML文件結構定義方式
- 主要是描述元素的內容和屬性
- XML文件的相對應DTD可以宣告在該XML文件裡，稱為內嵌式的DTD

```
<!DOCTYPE example1 [
<!ELEMENT example1 (example2)>
<!ELEMENT example2 (#PCDATA)>
]>
<example1>
<example2>This is a test!</example2>
</example1>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-10

DTD (Cont.)

- 也可將DTD的宣告置於另一個檔案，然後在XML文件裡匯入該DTD檔案。這種方式稱為外部宣告

- ☞ XML文件

```
<?xml version = "1.0"?>
<!-- Using an external subset -->
<!DOCTYPE example1 SYSTEM "test.dtd">
<example1>
<example2>This is a test!</example2>
</example1>
```

- ☞ DTD檔案 (存成test.dtd)

```
<!-- test.dtd -->
<!-- External declarations -->
<!ELEMENT example1 (example2)>
<!ELEMENT example2 ( #PCDATA )>
```



DTD (Cont.)

- 一個XML元素的內容可以包含數個子元素，而這些子元素間的關係可有以下數種：

- ☞ 順序

```
<!ELEMENT family (father, mother, son, daughter)>
```

- ☞ 多選一

```
<!ELEMENT member (teacher | student | alumni)>
```

- ☞ 0-1次

```
<!ELEMENT family (father?, mother?, son, daughter)>
```

- ☞ 0-多次

```
<!ELEMENT family (father?, mother?, son*, daughter*)>
```

- ☞ 1-多次

```
<!ELEMENT family (parent+, son*, daughter*)>
```



DTD (Cont.)

- 元素內容的設定方式
 - ☞ 內容值為空：用EMPTY
<!ELEMENT oven **EMPTY**>
 - 表示oven元素沒有內容（但可以有屬性值）
 - ☞ 內容值為字串：用以下方式設定：
<!ELEMENT test (**#PCDATA**)>
 - ☞ 內容值可能為字串或子元素
<!ELEMENT test1 (**#PCDATA | test2**)>
 - ☞ 內容值不做任何規定：用ANY
<!ELEMENT data **ANY**>

©黃三益2007
資料庫的核心理論與實務第三版

14-13



DTD (Cont.)

- DTD的屬性設定
<!ELEMENT example (#PCDATA)>
<!ATTLIST example **att CDATA #REQUIRED**>
- DTD裡定義了以下的型態
 - ☞ 字串：用CDATA 表示
<!ATTLIST department name **CDATA**>
 - ☞ ID：用ID 表示。ID型態的資料不可以包含空白，斷行，Tab等區隔符號
<!ATTLIST department dno **ID #REQUIRED**>
 - ☞ IDREF：用IDREF 表示。IDREF型態的資料值必須出現在ID裡
<!ATTLIST student dnum **IDREF**>
 - ☞ 選列（enumerated）：設定資料的數個可能值，用”|”分隔
<!ATTLIST student gender (**M|F**) “F” >

©黃三益2007
資料庫的核心理論與實務第三版

14-14



```

<?xml version = "1.0"?>
<!--A DTD example -->
<!DOCTYPE library [
  <!ELEMENT library (branch+, book+)>
  <!ELEMENT branch (name, address)>
  <!ATTLIST branch branchID ID #REQUIRED>
  <!ELEMENT book (#PCDATA)>
  <!ATTLIST book inBranch IDREF >
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT address (#PCDATA)>
]>
<library>
  <branch branchID = "b1">
    <name>理工圖書館</name>
    <address>理工大道一號</address>
  </branch>
  <branch branchID = "b2">
    <name>文法商圖書館</name>
    <address>文管大道一號</address>
  </branch>
  <book inBranch = "b2">
    如何當個稱職的職場新鮮人
  </book>
  <book inBranch = "b2">
    如何管理資料庫
  </book>
  <book inBranch = "b1">
    JAVA十天速成手冊
  </book>
</library>

```

滿足XML語法的XML文件稱為合格 (Well Formed)

滿足其對應DTD定義的XML文件稱為有效 (Valid)

©黃三益2007
資料庫的核心理論與實務第三版

14-15

練習14-2

- 請寫出練習14-1的DTD
- Ans:

```

<?xml version = "1.0"?>
<!--A DTD example -->
<!DOCTYPE library_books [
  <!ELEMENT library_books (book+)>
  <!ELEMENT book (call_number, title, author+, publish)>
  <!ELEMENT call_number (#PCDATA)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT publish EMPTY>
  <!ATTLIST publish company CDATA>
  <!ATTLIST publish year CDATA>
  <!ATTLIST publish place CDATA>
]>

```

©黃三益2007
資料庫的核心理論與實務第三版

14-16

練習14-3

- 以下XML文件不是一個有效的文件，為什麼？

```
<?xml version = "1.0"?>
<!--A DTD example -->
<!DOCTYPE library [
  <!ELEMENT library (branch+, book+)>
  <!ELEMENT branch (name, address)>
  <!ATTLIST branch branchID ID
    #REQUIRED>
  <!ELEMENT book (#PCDATA)>
  <!ATTLIST book inBranch IDREF>
  <!ELEMENT address ( #PCDATA )>
]>
<library>
  <branch branchID = "b1">
    <name>理工圖書館</name>
    <address>理工大道一號</name>
  </branch>
```

```
<branch branchID = "b2">
  <name>文法商圖書館</name>
  <address>文管大道一號</name>
</branch>
<branch> ← 沒有定義branchID屬性
  <name>總圖書館</name>
</branch>
<book inBranch = "b2"> ← 沒有 address元素
  如何當個稱職的職場新鮮人
</book>
<book inBranch = "b3"> ← b3沒有定義
  如何管理資料庫
</book>
<book inBranch= "b1">
  JAVA十天速成手冊
</book>
</library>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-17

W3C Schema

- W3C所提出解決DTD的兩大缺點：
 - 有限的資料值型態
 - 非XML 語法
- XML Schema就是為解決以上問題而由W3C所提出的XML資料定義方式，一般認為XML Schema會逐步取代DTD成為XML資料定義的新標準。
 - 一個Schema檔的開頭是啟始標籤<schema ...>，結束時是結束標籤</schema>，並在其中宣告一些元素
 - 在Schema啟始標籤裡可以描述該Schema定義的命名空間
 - 元素的宣告可以利用<element name="xxx" type="yyy" minOccurs='a' maxoccurs='b' .../>

©黃三益2007
資料庫的核心理論與實務第三版

14-18

W3C Schema

常用的簡單型態包括

- boolean：0 (false) 或 1 (true)
- byte：一個字元
- string：字串
- date：日期型態
- time：時間型態
- integer：整數
- positiveInteger：正整數
- id：識別一個元素或屬性
- idref：一個對id的參考

©黃三益2007
資料庫的核心理論與實務第三版

14-19

W3C Schema (Cont.)

XML Schema也允許你自行建立複合型態

- 複合型態的宣告語法是在啟始標籤<complexType>和結束標籤</complexType>中間列出其所包含的元素和屬性
- 常見的元素組成方式有以下三種：
 - ☞ all：表示元素的出現次序不限，但每個元素不可出現超過一次
 - ☞ sequence：表示元素要按次序出現
 - ☞ choice：表示可選擇性的出現一些元素

©黃三益2007
資料庫的核心理論與實務第三版

14-20

W3C Schema (Cont.)

- 元素也可以用啟始標籤<element ...>和結束標籤</element>，並在中間設定其（複合）型態的方式來宣告
- 屬性是用標籤<attribute name = " xxx", type="yyy" ...>來宣告

W3C Schema (Cont.)

```
<?xml version = "1.0" ?>  
<schema targetNamespace="http://www.mis.nsysu.edu.tw/db-book/lib/"  
  xmlns=http://www.w3.org/2001/XMLSchema  
  xmlns:lib=" http://www.mis.nsysu.edu.tw/db-book/lib/">  
  <element name = "library" >  
    <complexType>  
      <sequence>  
        <element ref= "branch" maxOccurs= "unbounded" />  
        <element name = "book" type= "lib:bookType" minOccurs= "0"  
          maxOccurs= "unbounded" />  
      </sequence>  
    </complexType>  
  </element>
```

本Schema的目標命名空間
預設命名空間
本Schema的命名空間
Library包括數個branch和數個book

W3C Schema (Cont.)

```
<element name = "branch" >
  <complexType>
    <sequence>
      <attribute name = "branchID" type = "id" use= "required" />
      <element name = "name" type = "string" />
      <element name = "address" type = "string" />
    </sequence>
  </complexType>
</element>
```

```
<complexType name = "bookType" >
  <simpleContent>
    <extension base="string">
      <attribute name = "inBranch" type = "idref" />
      <attribute name = "price" type= "positiveInteger" />
    </extension>
  </simpleContent>
</complexType>
</schema>
```

定義branch元素

定義bookType型態

©黃三益2007
資料庫的核心理論與實務第三版

14-23

有效的XML文件

```
<?xml version = "1.0"
<library
  xmlns="http://www.mis.nsysu.edu.tw/db-book/lib/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.mis.nsysu.edu.tw/db-book/lib/library-schema.xsd"
  <branch branchID = "b1">
    <name>理工圖書館</name>
    <address>理工大道一號</address>
  </branch>
  <branch branchID = "b2">
    <name>文法商圖書館</name>
    <address>文管大道一號</address>
  </branch>
  <branch branchID = "b3">
    <name>總圖書館</name>
    <address>大學路一號</address>
  </branch>
  <book inBranch = "b2" price = 400>
    如何當個稱職的職場新鮮人
  </book>
  <book inBranch = "b3" price = 500>
    如何管理資料庫
  </book>
  <book inBranch = "b1">
    JAVA十天速成手冊
  </book>
</library>
```

預設命名空間

Schema位置

©黃三益2007
資料庫的核心理論與實務第三版

14-24

W3C Schema (Cont.)

- 在許多應用裡，為了簡化起見，轉成的XML檔裡只有元素而沒有屬性。
- 考慮以下的另一個W3C Schema檔案（檔案名稱為library-schema2.xsd）

```
<?xml version = "1.0" ?>
  <schema targetNamespace="http://www.mis.nsysu.edu.tw/DB-book/"
    xmlns="http://www.w3.org/2001/XMLSchema" >
    <element name = "library" >
      <complexType>
        <sequence>
          <element name = "branch" type = "branchType"
            maxOccurs= "unbounded" />
          <element name = "book" type= "bookType" minOccurs= "0"
            maxOccurs= "unbounded" />
        </sequence>
      </complexType>
    </element>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-25

W3C Schema (Cont.)

```
<complexType name = "branchType" >
  <sequence>
    <element name = "branchID" type = "string" />
    <element name = "name" type = "string" />
    <element name = "address" type = "string" />
  </sequence>
</complexType>
<complexType name = "bookType" >
  <sequence>
    <element name = "title" type = "string" />
    <element name = "inBranch" type = "string" />
    <element name = "price" type= "positiveInteger" minOccurs= "0"
      maxOccurs= "1" />
  </sequence>
</complexType>
</schema>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-26

Valid XML using W3C-Schema

```
<?xml version = "1.0"
< library
  xmlns="http://www.mis.nsysu.edu.tw/db-book/lib/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance "
  xsi:schemaLocation = "http://www.mis.nsysu.edu.tw/DB-book/library-schema2.xsd" >
<branch>
  <branchID>b1</branchID>
  <name>理工圖書館</name>
  <address>理工大道一號</address>
</branch>
<branch >
  <branchID>b2</branchID>
  <address>文法商圖書館</address>
  <address>文管大道一號</address>
</branch>
<book >
  <title>如何當個稱職的職場新鮮人</title>
  <inBranch>b2</inBranch>
  <unitPrice>400</unitPrice>
</book>
<book >
  <title>如何管理資料庫</title>
  <inBranch>b2</inBranch>
  <unitPrice>500</unitPrice>
</book>
<book >
  <title>JAVA十天速成手冊</title>
  <inBranch>b1</inBranch>
</book>
</library>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-27

練習 14-4

- 請寫出[練習 14-1](#)的schema

Ans:

```
<?xml version = "1.0" ?>
  <schema targetNamespace="http://www.mis.nsysu.edu.tw/db-book/lib-books/"
  xmlns:lib-books="http://www.mis.nsysu.edu.tw/db-book/lib-books/"
  xmlns="http://www.w3.org/2001/XMLSchema">
  <element name = "library_books" >
  <complexType>
  <sequence>
  <element name = "book" type= "lib-books:bookType"
  maxOccurs= "unbounded" />
  </sequence>
  </complexType>
  </element>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-28

練習14-4 (Cont.)

```
<complexType name= "bookType" >
  <sequence>
    <element name = "call_number" type = "string" />
    <element name = "title" type = "string" />
    <element name = "author" type = "string" maxOccurs= "unbounded" />
    <element name = "publish" type = "lib-books:publishType" />
    <element name = "price" type = "positiveInteger" />
  </sequence>
</complexType>
</element>
<complexType name= "publishType" >
  <sequence>
    <attribute name = "company" type = "string" />
    <attribute name = "year" type = "string" />
    <attribute name = "place" type = "string" />
  </sequence>
</complexType>
</schema>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-29



XML文件的處理

- 多種模式，各提供適當的API來存取XML檔案資料
 - ☞ DOM (Document Object Model)
 - 一個XML檔案由剖析器(Parser)讀入後，在電腦記憶體裡形成一個樹狀結構
 - ☞ SAX (Simple API for XML)
 - SAX並不試圖將整個XML檔案載入，而只提供一些API來抓取XML裡的部分資料
 - ☞ XPath
 - 提供了一種簡單的查詢方式
 - ☞ XQuery
 - 類似XPath，不過語法較為複雜，可以做較複雜的條件搜尋
 - ☞ XSL (Extensible Stylesheet Language)
 - 設定XML文件的格式 (Style)，以便轉成另一個XML檔案、HTML檔案、或其他格式的檔案

©黃三益2007
資料庫的核心理論與實務第三版

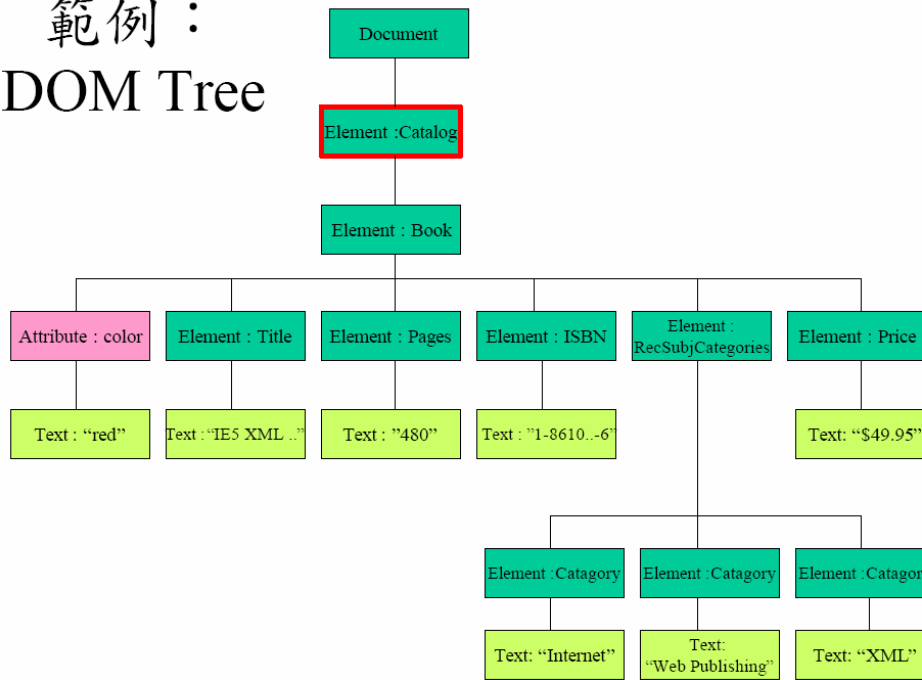
14-30



範例：XML文件

```
<Catalog>
  <Book color="red">
    <Title>IE5 XML Programmer's Reference</Title>
    <Pages>481</Pages>
    <ISBN>1-861001-57-6</ISBN>
    <RecSubjCategories>
      <Category>Internet</Category>
      <Category>Web Publishing</Category>
      <Category>XML</Category>
    </RecSubjCategories>
    <Price>49.99</Price>
  </Book>
</Catalog>
```

範例： DOM Tree



XQuery

- XQuery是建置在XPath之上以提供更強大的搜尋功能
- 範例

```
FOR $r IN doc("library.xml")/branch[branchID]
LET $b:=doc("library.xml")/book[inBranch=$r]
WHERE count($b) > 100000
RETURN
  <big-branch>{<branchID>{$r}</branchID>,
    <bookcount>{count($b)}</bookcount>}
  </big-branch>
```

©黃三益2007
資料庫的核心理論與實務第三版

14-33

資料庫管理系統對XML的支援

- XML的專屬DBMS 仍很少見
- 絕大多數DBMS所提供的功能僅限於XML文件的轉入和轉出
- 以Oracle為例

©黃三益2007
資料庫的核心理論與實務第三版

14-34

XML轉入DBMS的資料表

- 最簡單的方式是將一個XML文件內容當成一個欄位值
 - ☞ 產生一個資料表來儲存所有XML文件
CREATE TABLE xml_doc (
docname VARCHAR2(100) PRIMARY KEY,
doc CLOB,
insertdate DATE DEFAULT SYSDATE);
 - ☞ 利用輔助程式 (insertXML, 是一支PL/SQL程式) 將XML文件新增成該資料表的一筆記錄
 - ☞ 利用輔助程式來檢查該XML文件

©黃三益2007
資料庫的核心理論與實務第三版

14-35

XML轉入DBMS的資料表(Cont.)

- 第二種方式是將一個XML文件建成一個資料表，並將裡頭的標籤視為欄位
 - ☞ 產生一個資料表來儲存一個XML文件。
CREATE TABLE empleados (empid NUMBER PRIMARY KEY,
empname VARCHAR2(30),
empjob VARCHAR2(30),
empsal NUMBER);
 - ☞ 利用輔助程式 (insertXML, 是一支PL/SQL程式) 將XML文件匯入該資料表

©黃三益2007
資料庫的核心理論與實務第三版

14-36

DBMS的資料表轉出XML文件

- 先準備成一個巢狀結構的資料表以便匯出

```
create or replace type nested_type as object (c1 number(5), c2
varchar2(30));
create or replace type nested_type_table as table of nested_type;
create table nested_sample (
    field1 number(5),
    field2 nested_type_table
) nested table field2 store as nested_table;
```
- 呼叫輔助程式 (XmlReadFromDb, 是一支java程式) 來讀取該資料表並輸出成一XML檔

