Автоматическое изменение данных таблицы по получении документов XML

Владимир Пржиялковский

Преподаватель технологий Oracle
prz@yandex.ru
http://open-oracle.ru/

Февраль 2013 г.

"Чего ты хочешь отъ меня?"
Нахмурясь, голова вскричала :
"Вот, гостя мне судьба послала!"
А. С. Пушкин. Руслан и Людмила

Реферат

В статье показано, как пользуясь аппаратом событий репозитария XML DB можно добиться автоматического изменения данных таблицы в результате обычного копирования данных XML в репозитарий.

Введение

Основополагающей составной частью XML DB в БД Oracle является репозитарий. По форме он моделирует файловую систему в БД, и содержит в своем составе директории (папки, folders) и файлы, произвольного типа. Доступ к ним осуществляется как программно (встроенный пакет DBMS_XDB), так и по протоколам FTP и HTTP. Последние два способа позволяют работать с содержимым репозитария без наличия на компьютере программного обеспечения Oracle, что, безусловно, облегчает встраивание БД Oracle в общую информационную систему.

Функционально с репозитарием XML DB связан аппарат событий, моделирующий, по сути, триггерные процедуры, способные к автоматическому срабатыванию при выполнении действий с «файловой системой». Так, есть возможность запрограммировать срабатывание нужной нам процедуры при помещении в репозитарий очередного файла. Именно это и будет сделано ниже для решения следующей задачи:

> Пусть в оговоренный каталог репозитария поступает файл с документом XML. Проверить, действительно ли в нем содержатся правильно оформленные данные для добавления в таблицу DEPT из схемы SCOTT, и если это так, добавить строки.

Далее показано, как это сделать.

Создание папки в репозитарии

Создадим рабочую папку для размещения поступающих документов XML:

```
CONNECT scott/tiger

DECLARE b BOOLEAN := FALSE;
BEGIN
  b := DBMS_XDB.CREATEFOLDER ( '/public/app' );
END;
/
```

Позже мы свяжем с этой папкой особенный «конфигурационный файл», который обяжет XML DB автоматически выполнять необходимые нам действия при помещении в эту папку новых файлов и замене старых.

Создание схемы XML для проверки поступающих документов и регистрация схемы в XML DB

Для добавления строк в таблицу DEPT будем брать не все вообще файлы XML, попадающие в папку */public/app*, а только те, чье содержание удовлетворяет «канонической форме» Oracle, используемой для представления данных в таблице, то есть:

```
<ROWSET>
   <ROW><Столбец1>значение</Столбец1><Столбец2>значение</Столбец2>...</ROW>
   <ROW><Столбец1>значение</Столбец1><Столбец2>значение</Столбец2>...</ROW>
   ...
</ROWSET>
```

Построить схему XML канонического представления данных для таблицы DEPT не сложно вручную, однако ради общности, а также страховки от ошибок, ниже будет выполнено автоматическое создание схемы функцией GETXML из пакета DBMS_XMLQUERY. Эта функция способна включить схему в документ XML с данными ответа на запрос SQL. Построим запрос SELECT * FROM dept с ложным условием отбора строк (данные-то нам не нужны), а схему из результата GETXML извлечем с помощью функции XMLQUERY:

```
VARIABLE schema VARCHAR2 ( 4000 )

BEGIN
SELECT
XMLSERIALIZE ( DOCUMENT
 XMLQUERY (
   'declare namespace ns = "http://www.w3.org/2001/XMLSchema";
    /DOCUMENT/ns:schema'
   PASSING t.s
   RETURNING CONTENT
 )
 AS VARCHAR2 ( 4000 )
)
INTO :schema
FROM ( SELECT XMLTYPE ( DBMS_XMLQUERY.GETXML (
                                  'SELECT * FROM dept WHERE 2 = 1'
                                , DBMS_XMLQUERY.SCHEMA ) ) s
       FROM   dual ) t
;
END;
/
```

Результат в SQL*Plus покажет команда PRINT:

```
PRINT schema
```

При необходимости в полученную схему можно внести ручные правки. Так, не помешало бы заменить значение атрибута `minOccurs` у элемента `<xsd:element name="DEPTNO" ...` схемы с `0` на `1`.

Чтобы использовать эту схему для проверки документов способом, о котором речь идет ниже, ее требуется зарегистрировать в репозитарии. Пример регистрации под именем *http://app/deptCanonical.xsd*:

```
BEGIN
DBMS_XMLSCHEMA.REGISTERSCHEMA (
  schemaurl => 'http://app/deptCanonical.xsd'
, schemadoc => :schema
);
END;
/
```

Аппарат событий репозитария XML DB

Аппарат событий репозитария XML DB очерчивается следующими понятиями:

- *Событие* (event) — описывает действия с ресурсами (файлами и каталогами), и может быть вида Create, Delete, Update, Render (чтение), Lock, LinkIn (установление ссылки) и так далее (полный перечень в документации по Oracle).
- *Слушатель событий* (event listener) — пакет на PL/SQL или же класс Java, процедурам или же методам которых СУБД автоматически передает управление при возникновении определенных событий, и с помощью которых разработчик программирует желаемые действия.
- *Обработчик событий* (event handler) — процедура PL/SQL или метод Java, обрабатывающие события типа «pre событие» и «post событие», по аналогии с обычными триггерными процедурами типа BEFORE и AFTER.
- *Файл конфигурации ресурса* (resource configuration file) — файл XML, который задает привязку слушателя событий с определенным ресурсом репозитария.

Для решения нашей задачи потребуется написать процедуры-обработчики события «появление файла», создать файл конфигурации и связать его с папкой */public/app*.

Программирование процедур-обработчиков поступления данных XML

С созданием файла в репозитарии связаны три события (Create, LinkIn и LinkTo), но по определенным причинам программировать будем обработку LinkIn. Для общности добавим обработку Update, которую сведем к действиям по LinkIn. Для обработчиков событий создается пакет, в котором имена процедур выбираются по определенным правилам: handle[pre | post]*имя_события*.

```
CREATE OR REPLACE PACKAGE app_event_pkg AS
  PROCEDURE handlepostupdate  ( eventobject DBMS_XEVENT.XDBREPOSITORYEVENT );
  PROCEDURE handlepostlinkin  ( eventobject DBMS_XEVENT.XDBREPOSITORYEVENT );
END;
/

CREATE OR REPLACE PACKAGE BODY app_event_pkg AS
 PROCEDURE handlepostupdate ( eventobject DBMS_XEVENT.XDBREPOSITORYEVENT ) AS
    BEGIN
      handlepostlinkin ( eventobject );
    END;

 PROCEDURE handlepostlinkin ( eventobject DBMS_XEVENT.XDBREPOSITORYEVENT ) AS
      xdbpathobj DBMS_XEVENT.XDBPATH;
      respath    VARCHAR2 ( 1000 );
      restype    VARCHAR2 ( 100 );
      xcontents  XMLTYPE;
    BEGIN
      xdbpathobj := DBMS_XEVENT.GETPATH ( eventobject );
      respath    := DBMS_XEVENT.GETNAME ( xdbpathobj );

      SELECT                                -- узнать из PATH_VIEW тип добавленного файла
       XMLCAST (
        XMLQUERY (
          'declare namespace
                      ns = "http://xmlns.oracle.com/xdb/XDBResource.xsd";
           /ns:Resource/ns:ContentType'
          PASSING r.res RETURNING CONTENT
        )
        AS VARCHAR2 ( 100 )
      )
      INTO restype
      FROM PATH_VIEW r WHERE r.path = respath
      ;

      IF restype = 'text/xml' THEN       -- если файл типа XML, проверяем содержимое

         SELECT                          -- извлекаем содержимое
            XMLTYPE ( XMLSERIALIZE ( DOCUMENT
            XMLQUERY (
               'declare namespace
```

```
                        ns = "http://xmlns.oracle.com/xdb/XDBResource.xsd";
              /ns:Resource/ns:Contents/*'
             PASSING r.res RETURNING CONTENT
          ) ) )
        INTO   xcontents
        FROM   PATH_VIEW r WHERE r.path = respath
        ;

        -- проверяем содержимое на соответствие схеме XML
        IF  xcontents.ISSCHEMAVALID ( 'http://app/deptCanonical.xsd' ) = 1
        THEN
           INSERT INTO dept            -- вставляем строки в DEPT
             SELECT                    -- превращаем данные XML в табличный вид
                depts.deptno, depts.dname, depts.loc
             FROM
                XMLTABLE (
                     '/ROWSET/ROW'
                     PASSING xcontents
                     COLUMNS deptno NUMBER    ( 2 ) PATH 'DEPTNO'
                           , dname  VARCHAR2 ( 14 ) PATH 'DNAME'
                           , loc    VARCHAR2 ( 13 ) PATH 'LOC'
                ) AS depts
           ;
        END IF;
      END IF;
    END;
END;
/
```

Обращение к описанию и содержимому файла в репозитарии делается через системную таблицу (view) PATH_VIEW.

Выборка содержимого в переменную XCONTENTS выглядит неуклюже: результат XMLQUERY сериализуется в CLOB, и после этого переводится в XMLTYPE, хотя, вроде бы, сам имеет тип XML. Такое нагромождение преобразований является вынужденной реакцией на несколько туманную реализацию XMLTYPE фирмой Oracle. Вызвано ли это тем, что функция XQuery способна возвращать не только документы XML, но и значения, или же чем-нибудь еще, не ясно.

Подобно телам обычных триггерных процедур, в телах обработчиков событий не разрешено использовать команды управления транзакциями; отсюда отсутствие COMMIT. Однако, и далее это будет видно, команда COMMIT будет отработана неявно. Другие ограничения перечислены в документации по Oracle XML DB.

Нельзя не отметить, что в жизни программирование обработчиков обязано включать обработку ошибок. Здесь этого не сделано намеренно, чтобы отчетливее представить главную идею программы.

Подготовка файла конфигурации

Файл конфигурации создается опять-таки по определенным правилам, общий характер которых ясен из следующего примера:

```
DECLARE b BOOLEAN := FALSE;
BEGIN
  b := DBMS_XDB.CREATEFOLDER ( '/public/resconfig' );

  b := DBMS_XDB.CREATERESOURCE (
         '/public/resconfig/app-resconfig1.xml'
       , '<ResConfig
            xmlns    ="http://xmlns.oracle.com/xdb/XDBResConfig.xsd"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation=
                          "http://xmlns.oracle.com/xdb/XDBResConfig.xsd
                           http://xmlns.oracle.com/xdb/XDBResConfig.xsd">
          <event-listeners>
```

```
              <listener>
                <description>Category application</description>
                <schema>SCOTT</schema>
                <source>APP_EVENT_PKG</source>
                <language>PL/SQL</language>
                <events>
                  <Post-LinkIn/>
                  <Post-Update/>
                </events>
              </listener>
            </event-listeners>
            <defaultChildConfig>
              <configuration>
                <path>/public/resconfig/app-resconfig1.xml</path>
              </configuration>
            </defaultChildConfig>
          </ResConfig>'
       , 'http://xmlns.oracle.com/xdb/XDBResConfig.xsd'
       , 'ResConfig'
       );
END;
/
```

Цифра 1 в названии файла *app-resconfig1.xml* призвана напомнить, что файлов конфигурации можно создать несколько, по мере надобности. Теперь свяжем с файл конфигурации рабочим каталогом */public/app*:

```
DECLARE b BOOLEAN := FALSE;
BEGIN
  DBMS_RESCONFIG.APPENDRESCONFIG (
        '/public/app'
      , '/public/resconfig/app-resconfig1.xml'
      , DBMS_RESCONFIG.APPEND_RECURSIVE
  );
END;
/
```

Проверка автоматического занесения данных в таблицу

Подготовим файл *newdepts.xml* с данными для добавления в DEPT:

```
<ROWSET>
   <ROW>
      <DEPTNO>50</DEPTNO>
      <DNAME>MARKETING</DNAME>
      <LOC>MOSCOW</LOC>
   </ROW>
   <ROW>
      <DEPTNO>60</DEPTNO>
      <DNAME>JANITARY</DNAME>
      <LOC>LENINGRAD</LOC>
   </ROW>
 </ROWSET>
```

Добавление файла выполним с помощью FTP. Для этого порт FTP СУБД Oracle должен быть заранее открыт. Сделать это (или переустановить номер порта под новые потребности) можно так:

```
SQL> CONNECT / AS SYSDBA

SQL> EXECUTE DBMS_XDB.SETFTPPORT ( 2100 )
```

Для краткости текста выполним занесение в репозитарий файла пакетным образом. Подготовим файл *putfile.ftp* с командами для программы *ftp*:

```
open localhost 2100
user scott tiger
cd public/app
put newdepts.xml
quit
```

Копируем файл *newdepts.xml* в репозитарий XML DB:

```
>ftp -i -v -n -s:putfile.ftp
```

Проверка содержимого таблицы:

```
>sqlplus scott/tiger

SQL*Plus: Release 11.2.0.1.0 Production on Чт Фев 14 18:46:14 2013

Copyright (c) 1982, 2010, Oracle.  All rights reserved.

Присоединен к:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production

SQL> SELECT * FROM dept;

    DEPTNO DNAME          LOC
---------- -------------- -------------
        10 ACCOUNTING     NEW YORK
        20 RESEARCH       DALLAS
        30 SALES          CHICAGO
        40 OPERATIONS     BOSTON
        50 MARKETING      MOSCOW
        60 JANITARY       LENINGRAD
```

Обратите внимание на то, что автоматическое добавление строк в таблицу действительно оказалось закреплено командой COMMIT.

Файл с данными для добавления в таблицу DEPT с равным успехом можно занести в XML DB программой FAR, компьютерной мышкой, пользуясь соединением по WebDAV, или же программой CREATERESOURCE из пакета DBMS_XDB. Последний способ не самый интересный, поскольку подразумевает обычное соединение с Oracle, а в этом случае добавление строк можно сделать напрямую.