

# XML desde SQL y PL/SQL

## IDENTIFICACIÓN

<b>Proyecto</b>	DBconnector – XML desde SQL y PL/SQL
<b>Nombre del Documento</b>	DOCxmlDesdePLSQL.odt
<b>Autor</b>	Juan Luis Serradilla Amarilla
<b>Versión Actual</b>	1.0
<b>Fecha de la Versión</b>	7 de mayo de 2008

## RESUMEN

Esto es un "how-to" de cómo trabajar con XML desde SQL o PL/SQL (los ejemplos están probados en nuestra BD de desarrollo, que es 9i).

## VERSIONES

<b>Versión</b>	<b>Fecha</b>	<b>Autor</b>	<b>Descripción</b>
1.0	07/05/2008	Juan Luis Serradilla Amarilla	Versión inicial

## Índice de contenido

1.Introducción.....	3
2.Obtener los datos de una tabla en formato XML.....	3
2.1.Desde PL/SQL con DBMS_XMLGEN.....	3
2.2.Desde SQL usando funciones de SQLX.....	3
3.Almacenar datos XML en una tabla y leerlos.....	4
4.Crear un índice sobre datos XML.....	4
5.Referencias.....	4

## 1. Introducción.

---

Oracle, ya en su versión 8i, empezó a introducir capacidades de procesamiento XML en su motor de base de datos.

Esto es una Nota Técnica, más bien un "how-to", de cómo trabajar con XML desde SQL o PL/SQL (los ejemplos están probados en nuestra BD de desarrollo, que es 9i).

## 2. Obtener los datos de una tabla en formato XML.

---

### 2.1. Desde PL/SQL con DBMS\_XMLGEN.

```
SQL>
DECLARE
  Ctx      DBMS_XMLGEN.ctxHandle;   -- Var's to convert SQL output to XML
  xml      clob;
  emp_no   NUMBER := 7369;
  xmlc     varchar2(4000);          -- Var's required to convert lob to varchar
  off      integer := 1;
  len      integer := 4000;
BEGIN
  Ctx := DBMS_XMLGEN.newContext('SELECT * FROM emp WHERE empno = '||emp_no);
  DBMS_XMLGen.setRowsetTag(Ctx, 'EMP_TABLE');
  DBMS_XMLGen.setRowTag(Ctx, 'EMP_ROW');
  xml := DBMS_XMLGEN.getXML(Ctx);
  DBMS_XMLGEN.closeContext(Ctx);

  DBMS_LOB.READ(xml, len, off, xmlc);   -- Display first part on screen
  DBMS_OUTPUT.PUT_LINE(xmlc);
END;
```

```
SQL> /
<?xml version="1.0"?>
<EMP_TABLE>
<EMP_ROW>
  <EMPNO>7369</EMPNO>

  <ENAME>SMITH</ENAME>
  <JOB>CLERK</JOB>
  <MGR>7902</MGR>

  <HIREDATE>17-DEC-80</HIREDATE>
  <SAL>800</SAL>
  <DEPTNO>20</DEPTNO>

</EMP_ROW>
</EMP_TABLE>
```

PL/SQL procedure successfully completed.

### 2.2. Desde SQL usando funciones de SQLX.

Como XMLelement(), XMLForest(), XMLSequence(), etc.

```
SQL> set long 32000
SQL> SELECT XMLELEMENT("EMP_TABLE",
      (select XMLELEMENT("EMP_ROW",
            XMLFOREST(empno, ename, job, mgr, hiredate, sal, deptno)
            )
      from emp
      where empno = 7369))
from dual;
```

```
XMLELEMENT("EMP_TABLE", (SELECTXMLELEMENT("EMP_ROW", XMLFOREST(EMPNO, ENAME, JOB, MGR
-----
<EMP_TABLE><EMP_ROW><EMPNO>7369</EMPNO><ENAME>SMITH</ENAME><JOB>CLERK</JOB><MGR>
7902</MGR><HIREDATE>17-DEC-80</HIREDATE><SAL>800</SAL><DEPTNO>20</DEPTNO></EMP_R
OW></EMP_TABLE>
```

### 3. Almacenar datos XML en una tabla y leerlos.

---

```
SQL> create table XMLTable (doc_id number, xml_data XMLType);
Table created.
```

```
SQL> insert into XMLTable values (1,
      XMLType('<FAQ-LIST>
            <QUESTION>
              <QUERY>Question 1</QUERY>
              <RESPONSE>Answer goes here.</RESPONSE>
            </QUESTION>
          </FAQ-LIST>'));
1 row created.
```

```
SQL> select extractValue(xml_data, '/FAQ-LIST/QUESTION/RESPONSE') -- XPath expression
      from XMLTable
      where existsNode(xml_data, '/FAQ-LIST/QUESTION[QUERY="Question 1"]') = 1;
```

```
EXTRACTVALUE(XML_DATA, '/FAQ-LIST/QUESTION/RESPONSE')--XPATHEXPRESSION
```

```
-----
Answer goes here.
```

### 4. Crear un índice sobre datos XML.

---

```
SQL> create index XMLTable_ind on XMLTable
      (extractValue(xml_data, '/FAQ-LIST/QUESTION/QUERY') );
```

```
Index created.
```

### 5. Referencias

---

- <http://www.oraFAQ.com/faqxml.htm>
- <http://www.oracle.com/technology/tech/xml/index.html>
- <http://www.oracle.com/technology/pub/articles/quinlan-xml.html>