|  |  |  |
| --- | --- | --- |
| Lenguaje de Marcas | | |
| Conceptual | Fecha: 17-02-16 | XML |

**ALUMNO**

1.- ¿Qué hace esta orden?

SELECT ProductModelID, Name

FROM Production.ProductModel

WHERE ProductModelID=31 or ProductModelID=156

FOR XML RAW, ELEMENTS;

Genera una salida con el ID del Modelo y un contenido XML con formato de etiquetas ELEMENT (no atributos) para aquellos productos que tengan como código uno de los dos valores (31o 156)

2.- Explica qué hace este código y cuál es el objetivo(s) final(es).

DECLARE @i\_documento int

DECLARE @documento varchar(1000)

SET @documento ='

<ROOT>

<authors><au\_id>172-32-1176</au\_id><au\_lname>White</au\_lname>

<au\_fname>Johnson</au\_fname><title>book1</title>

<phone>408 496-7223</phone><address>10932 Bigge Rd.</address>

<city>Menlo Park</city><state>CA</state><zip>94025</zip>

<contract>1</contract></authors>

<authors><au\_id>213-46-8915</au\_id><au\_lname>Green</au\_lname

<au\_fname>Marjorie</au\_fname><title>book2</title>

<phone>415 986-7020</phone><address>309 63rd St.

#411</address><city>Oakland</city><state>CA</state>

<zip>94618</zip>

<contract>1</contract></authors>

</ROOT>'

EXEC sp\_xml\_preparedocument @i\_documento OUTPUT, @documento

Declara dos variables:

* Una nos permitirá almacenar el contenido XML en formato texto (@**documento**).
* La otra variable (@**i\_documento**) nos servirá para almacenar la dirección de comienzo del contenido XML binario que se generará.

La sentencia SET … recoge el contenido en formato texto del XML almacenándolo en la variable @doc.

La sentencia EXEC sp\_xml\_preparedocument genera en la RAM el arbol binario correspondiente al contenido almacenado en la variable @documento dejando la dirección de comienzo del árbol en la variable @i\_documento.

Los objetivos finales pueden ser varios:

* Introducir dicho contenido XML binario (lo que tenemos en la RAM) dentro de una columna XML de una tabla.
* Explorar su contenido con sentencias o con programas que tengan la capacidad de hacerlo.
* …

3.- Expresa las diferencias que ves entre tomar la decisión de almacenar un contenido en formato XML en una columna de tipo **varchar** o de tipo **xml**.

* Mejor gestión del tamaño (a favor del tipo XML ya que se almacena en formato binario).
* Se garantiza que la entrada de información siempre esté correctamente formada porque si no es así, no entra.
* Podemos si es del tipo XML validar la entrada contra un esquema de colección.
* Podemos hacer consultas dentro del contenido de la columna con XQuery.

4.- ¿Qué diferencia existe entre una columna xml tipada y una columna xml no tipada?

* Una columna tipada está asociada a un esquema de colección, permitiendo con ello poder validar la entrada de la información de acuerdo a un patrón (.xsd)
* Una columna no tipada no está asociada a un esquema de colección, no permitiendo con ello poder validar la entrada de la información de acuerdo a un patrón (.xsd)

5.- ¿Qué pasos hay que llevar a cabo para crear una columna xml tipada?

* Debemos generar (creando desde cero o a partir de un archivo XML) un fichero .XSD que contenga el patrón de validación.
* Debemos crear un esquema de colección en la base de datos basado en el fichero anterior.
* Crear una columna de una tabla con tipo XML asociando dicha columna al esquema de colección creado anteriormente.

**VALORACIÓN**: 2 PUNTOS POR PREGUNTA