Ampliación de Bases de Datos

Práctica 3

Año 2014/2015 Grado en Ing. Informática Fdi - UCM

Sweden, twelve points - La Suède, douze points

Entretenido y casposo a partes iguales

Fecha de entrega: 24 de mayo de 2014¹.

Porcentaje en la calificación total de prácticas: 30 % (es decir, 9 % de la nota final de la asignatura).

Esta práctica se divide en cuatro apartados, de los cuales son obligatorios los dos primeros: diseño de la base de datos XML/DTD y realización de consultas en *XQuery*. Con ellos se puede obtener la calificación máxima de APTO (4/10). Para obtener una nota superior se deberán implementar las partes opcionales de la práctica. La calificación máxima varía según los apartados de la práctica implementados:

- Apartados 1 y 2: Calificación máxima de 4/10 (0.36 puntos en la nota final de la asignatura).
- Apartados 1,2 y 3: Calificación máxima de 9/10 (0.81 puntos en la nota final).
- Apartados 1,2,3 y 4: Calificación máxima de 10/10 (0.90 puntos en la nota final).

Se valorarán, principalmente:

- El correcto diseño de la base de datos XML. Será de especial importancia la ausencia de redundancias en la representación de la información.
- La corrección de los resultados devueltos por las consultas XQuery.
- En el caso de realizar la aplicación Java (opcional), se tendrá en cuenta la correcta estructuración de la aplicación y división de responsabilidades entre los distintos componentes.

Venerado por unos, aborrecido por otros, e ignorado por la mayoría. El festival de la canción de Eurovisión es un concurso televisivo anual en el que participan artistas (y algún que otro friki) que representan (y, de vez en cuando, abochornan) a los países miembros de la Unión Europea de Radiodifusión (UER).

El objetivo de la práctica consiste en diseñar una base de datos XML que permita almacenar la información de los festivales (presentes y pasados), de los artistas y países participantes, de las canciones intepretadas, y de las votaciones realizadas por cada uno de los países al resto de canciones. Aunque el concurso se desarrolla cada año en varias fases (dos semifinales y una final) nosotros sólo representaremos la información correspondiente a la final de cada uno de ellos.

Así pues, consideramos dos entidades principales:

Por un lado, tenemos a los artistas. Para cada uno de ellos almacenamos un nombre artístico y, opcionalmente, una descripción y una URL externa a un fichero de imagen que contenga la imagen de un artista. Ten en cuenta que algunos artistas pueden participar varias veces en distintas ediciones del concurso, incluso representando a países distintos cada vez.

¹Curiosamente, un día después del festival de este año.

Por otro lado, almacenaremos información sobre las distintas ediciones del concurso. Para cada una de ellas almacenamos el año, la ciudad y país donde tuvo lugar, la lista de países participantes, y la lista de votos emitidos. Para cada país que participa se indicará el artista representante, la canción interpretada, y el número de orden de de la actuación dentro de la gala. Con respecto a los votos, un país puede otorgar votos a otros países, cada uno de ellos con una puntuación.

1 Diseño de base de datos XML y DTD

1. Crea un fichero Eurovision.xml con la estructura mencionada arriba, e introduce algunos datos de ejemplo. Los datos de ejemplo han de contener, como mínimo, cuatro artistas y dos ediciones del concurso. Para cada edición se han de introducir al menos cinco países participantes y cada uno de estos países ha de otorgar como mínimo dos votos (con distinta puntuación) a dos países distintos.

Importante: Toda la información (artistas y ediciones) ha de estar contenida en un *único* archivo XML, cuyo nombre ha de ser el indicado anteriormente.

2. Diseña la DTD correspondiente a la base de datos y asóciala al documento XML creado anteriormente. Comprueba que el documento XML es válido con respecto a la DTD utilizando *Eclipse* u otra herramienta similar.

2 Consultas en XQuery

Una vez diseñada la base de datos, subiremos el archivo Eurovision.xml al SGBD de *eXist*² y realizaremos consultas en *XQuery* sobre esta base de datos. Cada apartado de esta parte de la práctica debe guardarse en un fichero .xquery separado.

Importante: no crees ninguna carpeta o colección dentro del SGBD de *eXist*. Tanto el fichero XML como los ficheros de consulta han de estar almacenados en el directorio raíz de la base de datos.

- 1. En primer lugar, escribe una consulta que obtenga los años de las ediciones del festival almacenadas en la BD. Guarda esta consulta en un fichero Eurovision1.xquery.
- 2. A continuación escribe en Eurovision2. xquery una consulta que devuelva la clasificación total de los participantes en el festival en un determinado año (por ejemplo, 2014). Antes de ello declara una variable global en el fichero . xquery del siguiente modo:

```
declare variable $anyo as xs:integer := 2014;
```

Tras la declaración escribe la consulta propiamente dicha, que deberá obtener, para cada participante de la edición \$anyo: el país correspondiente, el nombre de canción, el nombre de artista y la puntuación total obtenida. Cada resultado de la consulta debe mostrarse en un elemento <clasificacion> con el siguiente formato:

²En los laboratorios, el nombre de usuario del administrador es admin y la contraseña es eXist.

Los elementos <clasificacion> del resultado deben estar ordenados en order decreciente de puntuación.

3. Por último, escribe una consulta que devuelva un fragmento XHTML con las actuaciones de una determinada edición. En primer lugar debe mostrarse el nombre de la ciudad y el país donde tuvo lugar el concurso. Tras esto debe mostrarse, para cada actuación: el país, el nombre del artista, el nombre de la canción, la descripción del artista, la imagen del artista, y la lista de países que votaron a esa actuación.

Para realizar esta consulta define, en un nuevo fichero Eurovision3.xquery una variable \$anyo del mismo modo que en el apartado anterior. A continuación escribe la consulta *XQuery*, que muestre la información requerida con el siguiente formato:

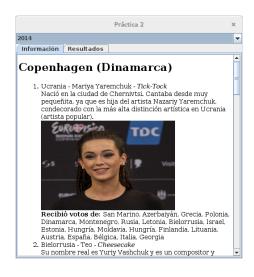
La lista de participantes debe estar ordenada ascendentemente por número de orden de actuación en el concurso.

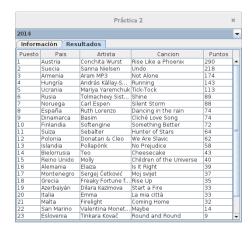
Indicación: Utiliza las funciones string-join y number.

Dado que la URL con la imagen es opcional, en aquellos participantes cuyos artistas no tengan este este atributo la etiqueta contendrá una imagen vacía. Esto puede arreglarse mostrando la etiqueta sólo en aquellos casos en los que el artista tenga asociada una imagen. Consulta la estructura if-then-else de XQuery para más información. No es obligatorio el uso de esta estructura en la práctica; la imagen puede quedar del siguiente modo: .

3 Acceso a la BD de *eXist* mediante *Java*

En esta parte de la práctica se creará una aplicación Java que ejecute las consultas anteriores utilizando la API de XQJ. El funcionamiento de la aplicación es el siguiente: se abrirá una ventana que contenga un JComboBox en la parte superior y un JTabbedPane en el centro con dos pestañas.





(b) Vista de ranking

(a) Vista de información

Figura 1: Interfaz de la aplicación

La primera de ellas debe mostrar, para el año seleccionado en el JComboBox un componente JTextPane que muestre la información devuelta por la consulta Eurovision3.xquery (ver Figura 1a). La segunda pestaña debe contener un JTable con la puntuación total de cada participante de dicho año (ver Figura 1b).

Para comenzar parte de la práctica, crea un nuevo proyecto de *Maven* y guarda los archivos .xquery del apartado anterior en la carpeta src/main/resources. Cuando sea necesario, la aplicación deberá leer estos ficheros y ejecutar las consultas contenidas en ellos.

También tienes que añadir al proyecto de *Maven* la dependencia a las librerías de XQJ. Añade el siguiente código al fichero pom.xml, dentro de la etiqueta ct>:

Las dos últimas consultas realizadas en el apartado anterior utilizaban una variable \$anyo, que se declaraba previamente. A continuación convertiremos estas consultas en paramétricas, para

que el valor de esta variable sea establecido por la aplicación *Java*, en lugar de aparecer directamente en el fichero . xquery. Para ello modifica la declaración de la variable \$anyo en los ficheros Eurovision2.xquery y Eurovision3.xquery del siguiente modo:

```
declare variable $anyo as xs:integer external;
```

Para ejecutar una consulta paramétrica desde una aplicación Java se realizan los siguientes pasos:

a) Apertura del archivo:

```
InputStream is = getClass().getResourceAsStream("/nombre_fichero.xquery")
```

b) Creación de la conexión a partir del XQDataSource, y creación de un objeto XQPreparedExpression a partir de la conexión:

```
XQConnection con = xqds.getConnection();
XQPreparedExpression exp = con.prepareExpression(is);
```

c) Asignación del valor de la variable externa (esto es, parámetro de la consulta):

```
exp.bindInt(new QName("anyo"), valor_del_parámetro, null);
```

d) Ejecución de la consulta:

```
XQResultSequence rs = exp.executeQuery();
```

La variable xqds contiene una referencia al XQDataSource, que debe haberse inicializado previamente (una sola vez en toda la aplicación):

```
XQDataSource xqs = new ExistXQDataSource();
xqs.setProperty("serverName", "localhost");
xqs.setProperty("port", "8080");
xqs.setProperty("user", "admin");
xqs.setProperty("password", "eXist");
```

Con respecto al componente JTextPane de la interfaz, debes llamar al método setContentType del mismo, pasándole como parámetro la cadena "text/html" para que pueda reconocer el resultado de la consulta Eurovision3. xquery como código HTML.

4 Creación del XML Schema

Por último, diseña el *XML Schema* correspondiente a la base de datos XML y, utilizando *Eclipse*, comprueba que esta última se adapta a las especificaciones del esquema. Puedes diseñar dicho esquema a mano o mediante una herramienta auxiliar (editor de esquemas de *Eclipse*, *XMLSpy*, etc.).