RELACIÓN 1

CHRISTIAN MILLÁN SORIA

1. Dado el siguiente código:

```
<html>
  <body>
    <table border="1">
      {
        for $x in doc("libros.xml")/bib/libro
        return <tr><td>{string($x/titulo)}</td></tr>
      }
    </table>
  </body>
</html>
```

a. Explicar qué se hace en el trozo de código seleccionado o sombreado de color. Explícalo con todo detalle.

El código lo que hace es generar una tabla HTML con los títulos de los libros que se encuentran en un archivo XML llamado "libros.xml". Esta tabla se muestra en una página web cuando se carga el archivo HTML que contiene el código.

La tabla tiene un borde que se muestra gracias al atributo border que se le asigna en la etiqueta "< table >". Dentro de la tabla, se van a agregar filas ("< tr >") y celdas ("< td >") por cada libro que se encuentre en el archivo XML.

Para obtener los libros del archivo XML, se utiliza una consulta XQuery. Esta consulta se hace con la cláusula for y selecciona todos los elementos "< libro >" que están dentro del elemento raíz "< bib >" del archivo XML. Por cada libro seleccionado en la consulta, se va a crear una fila en la tabla HTML.

Dentro de cada fila, se crea una celda que va a contener el título del libro. Para obtener el título, se utiliza una expresión XQuery que toma el contenido del elemento "< titulo >" de cada libro seleccionado y lo convierte en una cadena de caracteres con la función string(). Esta cadena de caracteres se agrega dentro de la etiqueta "< td >" para mostrar el título en la celda correspondiente.

En resumen, el código utiliza una consulta XQuery para seleccionar los títulos de los libros en un archivo XML y los muestra en una tabla HTML en una página web. La consulta se hace con la cláusula for y los resultados se insertan en la tabla HTML usando las etiquetas "< tr >" y "< td >".</>

b. ¿Qué resultado obtienes tras la ejecución de la consulta y añadir el resultado al código HTML que lo envuelve?

El resultado de la consulta sería una tabla HTML que mostraría los títulos de los libros contenidos en el archivo XML "libros.xml". Si se agrega el resultado de la consulta al código HTML que lo envuelve, se generaría una página web completa que incluiría la tabla con los títulos de los libros.

El código completo resultante se vería así:

```
<html>
  <body>
    <table border="1">
      <tr><td>TCP/IP Illustrated</td></tr>
      <tr><td>Advanced Programming in the Unix environment</td></tr>
      <tr><td>Data on the Web</td></tr>
      <tr><td>Economics of Technology for Digital TV</td></tr>
    </table>
  </body>
</html>
```

Y si se carga este archivo HTML en un navegador web, se mostraría una página con una tabla que contiene los títulos de los libros del archivo XML "libros.xml".

2. Dado el siguiente código:

a.

```
for $baile in /bailes/baile
where $baile/precio>30
return $baile/nombre
```

1. Explica qué se realiza con estas sentencias.

En la primera línea, se utiliza el comando "for" para establecer una variable $baile que se usará para iterar sobre todos los elementos "baile" del documento XML. La ruta "/bailes/baile" especifica que los elementos "baile" se encuentran en la raíz del documento.

En la segunda línea, se utiliza el comando "where" para establecer una condición. La condición es que el precio del baile ($baile/precio) debe ser mayor que 30.

En la tercera línea, se utiliza el comando "return" para especificar que se debe devolver el nombre de los bailes que cumplen con la condición establecida en la línea anterior. El resultado de esta consulta sería una lista de los nombres de los bailes que cumplen con la condición establecida.

En resumen, estas sentencias son una consulta XQuery que busca el nombre de los bailes cuyo precio es mayor que 30 en un documento XML.

2. Modifica estas sentencias para que desaparezca la cláusula "where" pero donde se haga el control del precio del baile.

Respuesta:

```
for $baile in /bailes/baile[precio>30]/nombre
return $baile
```

b.

```
for $baile in /bailes/baile
where $baile/precio>30 and $baile/precio/@moneda="euro"
return $baile/nombre
```

1. Explica qué realizan estas sentencias.

Estas sentencias realizan una consulta a un documento XML que contiene información sobre bailes y sus precios. La consulta filtra los elementos "baile" que tienen un precio mayor a 30 euros y que la moneda en la que está establecido el precio sea "euro". A continuación, devuelve el nombre de cada uno de los bailes que cumple con estos criterios.

La consulta comienza con un "for" que recorre todos los elementos "baile" que se encuentran en la ruta "/bailes/baile". La variable $baile toma el valor de cada uno de los elementos "baile" que se recorren.

A continuación, se utiliza la cláusula "where" para filtrar los elementos "baile" que cumplan con la condición de tener un precio mayor a 30 euros y que la moneda en la que está establecido el precio sea "euro". Para hacer esto, se utiliza la sintaxis "$baile/precio>30" para indicar que el precio del baile debe ser mayor a 30 euros y se utiliza la sintaxis "$baile/precio/@moneda="euro"" para indicar que la moneda en la que está establecido el precio debe ser "euro". El operador "and" se utiliza para indicar que ambas condiciones deben cumplirse para que un elemento "baile" se considere válido.

Por último, se utiliza la cláusula "return" para devolver el nombre de cada uno de los elementos "baile" que cumplen con los criterios establecidos. Para hacer esto, se utiliza la sintaxis "$baile/nombre" para indicar que se debe devolver el contenido del elemento "nombre" que se encuentra dentro del elemento "baile" que cumple con las condiciones.

2. Modifica estas sentencias para que desaparezca la cláusula "where" pero donde se haga el control del precio del baile.

Respuesta:

```
for $baile in /bailes/baile[precio>30 and precio/@moneda="euro"]
return $baile/nombre
```

c.

```
for $baile in /bailes/baile
order by $baile/sala
return
  <baile>
    {$baile/profesor}
    {$baile/sala}
  </baile>
```

1. Explica qué realizan estas sentencias.

Estas sentencias realizan una consulta a un documento XML que contiene información sobre bailes. La consulta devuelve una lista de elementos "baile" que incluyen información sobre el profesor que imparte el baile y la sala en la que se lleva a cabo, ordenados por el nombre de la sala.

La consulta comienza con la cláusula "for", que establece la variable $baile para iterar a través de todos los elementos "baile" que se encuentran en el documento XML. Dentro del ciclo "for", se crea un elemento "baile" que incluye información sobre el profesor y la sala en la que se lleva a cabo el baile. Para agregar esta información, se utiliza la sintaxis "{$baile/profesor}" y "{$baile/sala}", que accede al contenido de los elementos "profesor" y "sala" que están dentro del elemento "baile".

La cláusula "order by" se utiliza para ordenar los elementos "baile" por el nombre de la sala. En este caso, se utiliza la sintaxis "$baile/sala" para acceder al contenido del elemento "sala" que se encuentra dentro del elemento "baile" y ordenar los resultados en función de este valor.

En resumen, estas sentencias realizan una consulta que devuelve una lista de elementos "baile" que incluyen información sobre el profesor y la sala en la que se lleva a cabo el baile, ordenados por el nombre de la sala.

2. Modifica estas sentencias para que se produzcan criterios de búsqueda por un determinado profesor y por una determinada sala.

Respuesta:

```
for $baile in /bailes/baile
where $baile/profesor="nombre_profesor" and $baile/sala="nombre_sala"
order by $baile/sala
return
  <baile>
    {$baile/profesor}
    {$baile/sala}
  </baile>
```

3. Realiza un pequeño programa en XQUERY que proporcione el siguiente resultado:

```
<test>1 2 3 4 5</test>
<test>1 2 3 4 5</test>
<test>1 2 3 4 5</test>
```

Respuesta:

```
for $i in (1 to 3)
return <test>{(1 to 5)}</test>
```

Explicación:

Se utiliza la cláusula "for" para iterar tres veces, asignando a la variable $i los valores del 1 al 3. Se utiliza la sintaxis de paréntesis para crear una secuencia que va del 1 al 5, y se inserta esta secuencia dentro del elemento "test" utilizando llaves. El resultado es una secuencia de tres elementos "test", cada uno con los valores del 1 al 5 dentro.