

LMSGI06-Solución a la TAREA

ALMACENAMIENTO DE INFORMACIÓN.

Dado el fichero LMSGI06_TARE_R01.xml generar un esquema XSD que sirva para describir en XML dicha base de datos relacional.

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">

  <!-- Documentacion del esquema -->

  <xs:annotation>
    <xs:documentation xml:lang="es-es">
      Materiales para formación e-Learning
      <modulo>Lenguajes de marcas y sistemas de gestión de información.</modulo>
      <fecha_creacion> 2011</fecha_creacion>
      <autor> Nuky La Bruji</autor>
    </xs:documentation>
  </xs:annotation>

  <!-- Definicion del vocabulario -->

  <xs:element name="clase">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="alumnos"/>
        <xs:element ref="asignaturas"/>
        <xs:element ref="notas"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Tabla alumnos -->

  <xs:element name="alumnos">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="alumno"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!-- Definicion de cada fila -->

  <xs:element name="alumno">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="apenom"/>
        <xs:element ref="direc"/>
        <xs:element ref="pobla"/>
        <xs:element ref="telef"/>
      </xs:sequence>
      <xs:attribute name="cod" use="required" type="xs:ID"/>
    </xs:complexType>
  </xs:element>

  <!-- Definicion de cada columna -->

  <xs:element name="apenom">
    <xs:simpleType>
```

```

    <xs:restriction base="xs:string">
      <xs:maxLength value="30"></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="direc">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="30"></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="pobla">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="15"></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<xs:element name="telef">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="10"></xs:maxLength>
    </xs:restriction>
  </xs:simpleType>
</xs:element>

<!-- Tabla asignaturas -->

<xs:element name="asignaturas">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" ref="asignatura"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!-- Definicion de cada fila -->

<xs:element name="asignatura">
  <xs:complexType>

    <!-- Definicion de cada columna -->

    <xs:attribute name="cod" use="required" type="xs:ID"/>

    <xs:attribute name="nombre" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="5"></xs:maxLength>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
</xs:element>

<!-- Tabla notas -->

<xs:element name="notas">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element maxOccurs="unbounded" ref="nota"/>
  </xs:sequence>
</xs:complexType>
</xs:element>

<!-- Definicion de cada fila -->

<xs:element name="nota">
  <xs:complexType>

    <!-- Definicion de cada columna -->

    <xs:attribute name="alum" use="required" type="xs:IDREF"/>
    <xs:attribute name="asig" use="required" type="xs:IDREF"/>
    <xs:attribute name="calificacion">
      <xs:simpleType>
        <xs:restriction base="xs:decimal">
          <xs:totalDigits value="3"/>
          <xs:fractionDigits value="1"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>

  </xs:complexType>
</xs:element>
</xs:schema>

```

Crear el documento XML correspondiente a dicha base de datos.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<clase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LMSGI06_SOLT_R02_esquema.xsd">

  <alumnos>
    <alumno cod="n12344345">
      <apenom>Alcalde García, Luis</apenom>
      <direc>Las Manos, 24</direc>
      <pobla>Lamadrid</pobla>
      <telef>942756645</telef>
    </alumno>

    <alumno cod="n43483437">
      <apenom>González Pérez, Olga</apenom>
      <direc>Miraflor 28 - 3A</direc>
      <pobla>Torres</pobla>
      <telef>942564355</telef>
    </alumno>

    <alumno cod="n88234942">
      <apenom>Fernández Díaz, María</apenom>
      <direc>Luisa Fernanda 53</direc>
      <pobla>Miera</pobla>
      <telef>942346945</telef>
    </alumno>
  </alumnos>

  <asignaturas>
    <asignatura cod="a1" nombre="FH"/>
    <asignatura cod="a2" nombre="FOL"/>
    <asignatura cod="a3" nombre="ISO"/>
  </asignaturas>
</clase>

```

```

<asignatura cod="a4" nombre="LMSGI"/>
<asignatura cod="a5" nombre="PAR"/>
<asignatura cod="a6" nombre="GBD"/>
</asignaturas>

<notas>
<nota alum="n12344345" asig="a1" calificacion="4"/>
<nota alum="n43483437" asig="a1" calificacion="5"/>
<nota alum="n88234942" asig="a1" calificacion="8"/>
<nota alum="n12344345" asig="a2" calificacion="10"/>
<nota alum="n43483437" asig="a2" calificacion="7"/>
<nota alum="n88234942" asig="a2" calificacion="6"/>
<nota alum="n12344345" asig="a3" calificacion="3"/>
<nota alum="n88234942" asig="a3" calificacion="6"/>
<nota alum="n12344345" asig="a4" calificacion="8"/>
<nota alum="n43483437" asig="a4" calificacion="4"/>
<nota alum="n12344345" asig="a5" calificacion="6"/>
<nota alum="n12344345" asig="a6" calificacion="9"/>
</notas>

</clase>

```

Una vez realizado este documento realizar las siguientes consultas XQuery sobre los datos:

1. Obtener el nombre de todos los alumnos matriculados en algún módulo.

```

for $a in doc("LMSGI06_SOLT_R01_datos.xml")//alumno
return
<alumno>
{ $a/apenom }
</alumno>

```

2. Obtener las calificaciones del alumno de código "n43483437" en cada módulo.

```

for $n in doc("LMSGI06_SOLT_R01_datos.xml")//nota
where $n/@alum = "n43483437"
return
<nota>
{ $n/@asig } { $n/@calificacion }
</nota>

```

3. Obtener el nombre y el teléfono de cada alumno ordenado por apellidos de forma descendente.

```

for $a in doc("LMSGI06_SOLT_R01_datos.xml")//alumno
order by $a/apenom descending
return
<telefonos>
{ $a/apenom } { $a/telef }
</telefonos>

```

4. ¿Cuántos módulos hay?

```

count(doc("LMSGI06_SOLT_R01_datos.xml")//asignatura)

```

5. Obtener los nombres de los alumnos matriculados en LMSGI y sus notas ordenado por notas.

```

for $a in doc("LMSGI06_SOLT_R01_datos.xml")//alumno
for $n in doc("LMSGI06_SOLT_R01_datos.xml")//nota
for $m in doc("LMSGI06_SOLT_R01_datos.xml")//asignatura

where $a/@cod = $n/@alum and $n/@asig = $m/@cod and $m/@nombre="LMSGI"

```

```

order by $n/@calificacion
return
<notas_LMSGI>
  {$n/@calificacion}
  { $a/apenom}
</notas_LMSGI>

```

6. Obtener los nombres y las calificaciones de los matriculados en FH que han aprobado.

```

for $a in doc("LMSGI06_SOLT_R01_datos.xml")//alumno
for $m in doc("LMSGI06_SOLT_R01_datos.xml")//asignatura
for $n in doc("LMSGI06_SOLT_R01_datos.xml")//nota

where $a/@cod = $n/@alum and $n/@asig = $m/@cod and $m/@nombre="FH" and $n/@calificacion >
4
order by $n/@calificacion
return
<notas_FH>
  {$n/@calificacion}
  { $a/apenom}
</notas_FH>

```