

**Generar un esquema XSD que sirva para describir en XML dicha base de datos relacional.
Asociar el documento XML correspondiente a dicha base de datos.**

// Genero un documento XML llamado clase.xml

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

// Ahora genero un esquema XSD al que le he llamado para variar clase.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation xml:lang="Es-es">
      Base de datos de un instituto (Tarea 6 LMSGI)
      <fecha>2017-06-09</fecha>
      <autor>Adrián Álvarez Lois</autor>
    </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="clase" type="claseDatos"/>
  <xsd:complexType name="claseDatos">
    <xsd:sequence>
      <xsd:element name="alumnos" type="alumnosDatos" minOccurs="1" maxOccurs="1"/>
      <xsd:element name="asignaturas" type="asignaturasDatos" minOccurs="1"
maxOccurs="1"/>
      <xsd:element name="notas" type="notasDatos" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="alumnosDatos">
    <xsd:sequence>
      <xsd:element name="alumno" type="alumnoDato" minOccurs="1"
maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="alumnoDato">
    <xsd:sequence>
      <xsd:element name="apenom" type="xsd:string"/>
      <xsd:element name="direc" type="xsd:string"/>
      <xsd:element name="pobla" type="xsd:string"/>
      <xsd:element name="telef" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:decimal">
            <xsd:minExclusive value="6000000000"/>
            <xsd:maxExclusive value="9999999999"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="cod" type="codAlumnos" use="required"/>
  </xsd:complexType>
  <xsd:simpleType name="codAlumnos">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[a-z]([0-9]){8}"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:complexType name="asignaturasDatos">
    <xsd:sequence>
      <xsd:element name="asignatura" minOccurs="1" maxOccurs="unbounded"
type="asignaturaDato"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="asignaturaDato">
    <xsd:attribute name="cod" type="codAsignatura" use="required"/>
    <xsd:attribute name="nombre" type="nombreAsignatura" use="required"/>
  </xsd:complexType>
  <xsd:simpleType name="codAsignatura">
    <xsd:restriction base="xsd:string">
      <xsd:pattern value="[a-z][0-9]"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="nombreAsignatura">
    <xsd:restriction base="xsd:string">
```

```

    <xsd:pattern value="([A-Z])+"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="notasDatos">
  <xsd:sequence>
    <xsd:element name="nota" minOccurs="1" maxOccurs="unbounded" type="notaDato"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="notaDato">

  <xsd:attribute name="alum" type="codAlumnos" use="required"/>
  <xsd:attribute name="asig" type="codAsignatura" use="required"/>
  <xsd:attribute name="calificacion" type="calAsignatura" use="required"/>

</xsd:complexType>
<xsd:simpleType name="calAsignatura">
  <xsd:restriction base="xsd:decimal">
    <xsd:minInclusive value="0"/>
    <xsd:maxInclusive value="10"/>
  </xsd:restriction>
</xsd:simpleType>
</xsd:schema>

```

Una vez realizados estos documentos realizar las siguientes consultas XQuery sobre los datos:

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

```

for $a in doc("clase.xml")//alumnos/alumno
where $a/@cod union $a/../../../../notas/nota/@alum
return
<alumno>{ data($a/apenom) }</alumno>

```

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

```

for $a in doc("clase.xml")//notas/nota
for $b in doc("clase.xml")//asignaturas/asignatura
where $a/@alum = "n43483437" and $b/@cod = $a/@asig
return
<alumno>{ data($b/@nombre),"---",data($a/@calificacion) }</alumno>

```

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

```

for $a in doc("clase.xml")//alumnos/alumno
let $b := index-of($a/apenom,',')
order by $a/apenom descending
return
<alumno>{ data(substring-after($a/apenom,',')), "---", data($a/telef) }</alumno>

```

4 // ¿Cuántos módulos hay?

```

let $a := doc("clase.xml")//asignaturas/asignatura
return
<alumno>{ "Existen: ", data(count($a)), "módulos distintos" }</alumno>

```

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

```

for $a in doc("clase.xml")//asignaturas/asignatura,
$b in $a/../../../../notas/nota,
$c in $a/../../../../alumnos/alumno
where $a/@nombre="LMSGI" and $b/@asig = $a/@cod and $c/@cod = $b/@alum
order by $b/@calificacion
return
<alumno>{ "Nombre: ", data($c/apenom), " Calificacion: ",
data($b/@calificacion) }</alumno>

```

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

```
for $a in doc("clase.xml")//asignaturas/asignatura,
$b in $a/../../notas/nota,
$c in $a/../../alumnos/alumno
where $a/@nombre="FH" and $b/@asig = $a/@cod and $c/@cod = $b/@alum
and $b/@calificacion >=5
order by $b/@calificacion
return
<alumno>{ "Nombre: ", data($c/apenom), " Calificacion: ",
data($b/@calificacion) }</alumno>
```

A continuación adjunto una serie de pantallazos del proceso de elaboración de la tarea:

// Elaboración del XML

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <clase xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance" xsd:noNamespaceSchemaLocation="clase.xsd">
3   <alumnos>
4     <alumno cod="n12344345">
5       <apenom>Alcalde García, Luis</apenom>
6       <direc>Las Manos, 24</direc>
7       <pobla>Lanadrid</pobla>
8       <telef>942756645</telef>
9     </alumno>
10    <alumno cod="n43483437">
11      <apenom>González Pérez, Olga</apenom>
12      <direc>Miraflores 28 - 3A</direc>
13      <pobla>Torres</pobla>
14      <telef>942564355</telef>
15    </alumno>
16    <alumno cod="n88234942">
17      <apenom>Fernández Díaz, María</apenom>
18      <direc>Luisa Fernanda 53</direc>
19      <pobla>Miera</pobla>
20      <telef>942346945</telef>
21    </alumno>
22  </alumnos>
23  <asignaturas>
24    <asignatura cod="a1" nombre="FH"/>
25    <asignatura cod="a2" nombre="FOL"/>
26    <asignatura cod="a3" nombre="ISO"/>
27    <asignatura cod="a4" nombre="LMSGI"/>
28    <asignatura cod="a5" nombre="PAR"/>
29    <asignatura cod="a6" nombre="GBD"/>
30  </asignaturas>
31  <notas>
32    <nota alum="n12344345" asig="a1" calificacion="4"/>
33    <nota alum="n43483437" asig="a1" calificacion="5"/>
34    <nota alum="n88234942" asig="a1" calificacion="8"/>
35    <nota alum="n12344345" asig="a2" calificacion="10"/>
36    <nota alum="n43483437" asig="a2" calificacion="7"/>
37    <nota alum="n88234942" asig="a2" calificacion="6"/>
38    <nota alum="n12344345" asig="a3" calificacion="3"/>
39    <nota alum="n88234942" asig="a3" calificacion="6"/>
40    <nota alum="n12344345" asig="a4" calificacion="8"/>
41    <nota alum="n43483437" asig="a4" calificacion="4"/>
42    <nota alum="n12344345" asig="a5" calificacion="6"/>
43    <nota alum="n12344345" asig="a6" calificacion="9"/>
44  </notas>
45 </clase>
46
```

// Elaboración del XSD

```

40 <xsd:restriction base="xsd:string">
41 <xsd:pattern value="[a-z]([0-9]){8}" />
42 </xsd:restriction>
43 </xsd:simpleType>
44 <xsd:complexType name="asignaturasDatos">
45 <xsd:sequence>
46 <xsd:element name="asignatura" minOccurs="1" maxOccurs="unbounded" type="asignaturaDato"/>
47 </xsd:sequence>
48 </xsd:complexType>
49 <xsd:complexType name="asignaturaDato">
50 <xsd:attribute name="cod" type="codAsignatura" use="required"/>
51 <xsd:attribute name="nombre" type="nombreAsignatura" use="required"/>
52 </xsd:complexType>
53 <xsd:simpleType name="codAsignatura">
54 <xsd:restriction base="xsd:string">
55 <xsd:pattern value="[a-z][0-9]" />
56 </xsd:restriction>
57 </xsd:simpleType>
58 <xsd:simpleType name="nombreAsignatura">
59 <xsd:restriction base="xsd:string">
60 <xsd:pattern value="([A-Z])+" />
61 </xsd:restriction>
62 </xsd:simpleType>
63 <xsd:complexType name="notasDatos">
64 <xsd:sequence>
65 <xsd:element name="nota" minOccurs="1" maxOccurs="unbounded" type="notaDato"/>
66 </xsd:sequence>
67 </xsd:complexType>
68 <xsd:complexType name="notaDato">
69 <xsd:sequence>
70 <xsd:attribute name="alum" type="codAlumnos" use="required"/>
71 <xsd:attribute name="asig" type="codAsignatura" use="required"/>
72 <xsd:attribute name="calificacion" type="calAsignatura" use="required"/>
73 </xsd:sequence>
74 </xsd:complexType>
75 <xsd:simpleType name="calAsignatura">
76 <xsd:restriction base="xsd:decimal">
77 <xsd:minInclusive value="0"/>
78 <xsd:maxInclusive value="10"/>
79 </xsd:restriction>
80 </xsd:simpleType>
81 </xsd:schema>
82

```

// Elaboración de los Xquery

1)

```

1 (: 1. Obtener el nombre de todos los alumnos matriculados en algún módulo :)
2 for $a in doc("clase.xml")//alumnos/alumno
3 where $a/@cod union $a/../../notas/nota/@alum
4 return
5 <alumno>{ data($a/apenom) }</alumno>
6

```

2)

```

1 (: 2. Obtener las calificaciones del alumno de código "n43483437" en cada módulo :)
2 for $a in doc("clase.xml")//notas/nota
3 for $b in doc("clase.xml")//asignaturas/asignatura
4 where $a/@alum = "n43483437" and $b/@cod = $a/@asig
5 return
6 <alumno>{ data($b/@nombre), "---", data($a/@calificacion) }</alumno>
7

```

3)

```

1 |(: 3.  Obtener el nombre y el teléfono de cada alumno ordenado por apellidos de forma descendente. :)
2 |for $a in doc("clase.xml")//alumnos/alumno
3 |let $b := index-of($a/apenom, '')
4 |order by $a/apenom descending
5 |return
6 |<alumno>{ data(substring-after($a/apenom, ''), "---", data($a/telef) )}</alumno>
7 |

```

4)

```

1 |(: 4.  ¿Cuántos módulos hay? :)
2 |let $a := doc("clase.xml")//asignaturas/asignatura
3 |return
4 |<alumno>{ "Existen: ", data(count($a)), "módulos distintos" }</alumno>
5 |

```

5)

```

1 |(: 5.  Obtener los nombres de los alumnos matriculados en LMSGI y sus notas, ordenado por notas. :)
2 |for $a in doc("clase.xml")//asignaturas/asignatura,
3 |$b in $a/../../notas/nota,
4 |$c in $a/../../alumnos/alumno
5 |where $a/@nombre="LMSGI" and $b/@asig = $a/@cod and $c/@cod = $b/@alum
6 |order by $b/@calificacion
7 |return
8 |<alumno>{ "Nombre: ", data($c/apenom), " Calificacion: ", data($b/@calificacion) }</alumno>
9 |

```

6)

```

1 |(: 6.  Obtener los nombres y las calificaciones de los matriculados en FH que han aprobado. :)
2 |for $a in doc("clase.xml")//asignaturas/asignatura,
3 |$b in $a/../../notas/nota,
4 |$c in $a/../../alumnos/alumno
5 |where $a/@nombre="FH" and $b/@asig = $a/@cod and $c/@cod = $b/@alum and $b/@calificacion >=5
6 |order by $b/@calificacion
7 |return
8 |<alumno>{ "Nombre: ", data($c/apenom), " Calificacion: ", data($b/@calificacion) }</alumno>
9 |

```