

Domanda 2 (10% per la prova lunga, 20% per la prova breve)

Considerare il documento XML qui sotto e lo schema XSD nella pagina successiva. Individuare gli errori nella definizione dello schema che rendono non valido il documento rispetto allo schema, indicando possibili correzioni allo schema.

```
<films>
  <film>
    <titolo>Spirited Away</titolo>
    <anno>2001</anno>
    <durata>125 min</durata>
    <voto>8.6</voto>
    <generi>Animazione, Avventura, Famiglia</generi>
    <staff>
      <regista>Hayao Miyazaki</regista>
      <sceneggiatore>Hayao Miyazaki</sceneggiatore>
      <attori>
        <attore>Daveigh Chase</attore>
        <attore>Suzanne Pleshette</attore>
        <attore>Miyu Irino</attore>
      </attori>
    </staff>
  </film>
  <film>
    <titolo>Cowboys and Aliens</titolo>
    <anno>2011</anno>
    <durata>119 min</durata>
    <voto>6.2</voto>
    <generi>Azione, Fantascienza, Thriller</generi>
    <staff>
      <regista>John Favreau</regista>
      <attori>
        <attore>Daniel Craig</attore>
        <attore>Harrison Ford</attore>
        <attore>Olivia Wilde</attore>
      </attori>
    </staff>
  </film>
</films>
```

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="films">
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="film"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="film">
    <xsd:sequence>
      <xsd:element name="titolo" type="xsd:string"/>
      <xsd:element name="anno" type="xsd:int"/>
      <xsd:element name="durata" type="xsd:int"/>
      <xsd:element name="voto" type="xsd:string"/>
      <xsd:element name="generi" type="xsd:string"/>
      <xsd:element ref="staff"/>
    </xsd:sequence>
  </xsd:element>

  <xsd:element name="staff">
    <xsd:complexType>
      <xsd:choice>
        <xsd:element name="regista" type="xsd:string"/>
        <xsd:element name="sceneggiatore" type="xsd:string"/>
        <xsd:element name="attori">
          <xsd:complexType>
            <xsd:sequence minOccurs="0" maxOccurs="unbounded">
              <xsd:element name="attore" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

1. Errori

- a. Nell'elemento film non c'è complexType prima di sequence
- b. La durata è un int invece di una string
- c. Lo sceneggiatore deve essere opzionale
- d. Dentro attori c'è una choice al posto della sequence

XSD CORRETTO:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="films">
    <xsd:complexType>
      <xsd:sequence minOccurs="0" maxOccurs="unbounded">
        <xsd:element ref="film"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="film">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="titolo" type="xsd:string"/>
        <xsd:element name="anno" type="xsd:int"/>
        <xsd:element name="durata" type="xsd:string"/>
        <xsd:element name="voto" type="xsd:string"/>
        <xsd:element name="generi" type="xsd:string"/>
        <xsd:element ref="staff"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="staff">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="registra" type="xsd:string"/>
        <xsd:element name="sceneggiatore" type="xsd:string"
minOccurs="0"/>
        <xsd:element name="attori">
          <xsd:complexType>
            <xsd:sequence minOccurs="0" maxOccurs="unbounded">
              <xsd:element name="attore" type="xsd:string"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

</xsd:schema>
```

Domanda 3 (25% per la prova lunga, 50% per la prova breve)

Con riferimento a documenti come quello mostrato in precedenza (supponendolo memorizzato nel file `esame.xml`), e ripetuto qui sotto per comodità, rispondere alle interrogazioni mostrate nella pagina successiva

```
<films>
  <film>
    <titolo>Spirited Away</titolo>
    <anno>2001</anno>
    <durata>125 min</durata>
    <voto>8.6</voto>
    <generi>Animazione, Avventura, Famiglia</generi>
    <staff>
      <regista>Hayao Miyazaki</regista>
      <sceneggiatore>Hayao Miyazaki</sceneggiatore>
      <attori>
        <attore>Daveigh Chase</attore>
        <attore>Suzanne Pleshette</attore>
        <attore>Miyu Irino</attore>
      </attori>
    </staff>
  </film>
  <film>
    <titolo>Cowboys and Aliens</titolo>
    <anno>2011</anno>
    <durata>119 min</durata>
    <voto>6.2</voto>
    <generi>Azione, Fantascienza, Thriller</generi>
    <staff>
      <regista>John Favreau</regista>
      <attori>
        <attore>Daniel Craig</attore>
        <attore>Harrison Ford</attore>
        <attore>Olivia Wilde</attore>
      </attori>
    </staff>
  </film>
</films>
```

Si ricorda la disponibilità delle seguenti funzioni:

- `count()` che restituisce il numero di nodi dell'argomento
- `contains()` che ha due argomenti e verifica se la stringa del primo argomento contiene la stringa del secondo argomento

- (a) Con XPath, trovare i film che hanno stesso regista e sceneggiatore (il regista è uguale allo sceneggiatore)

```
//film[.//regista=../sceneggiatore]
```

```
//film[./staff/regista=./staff/sceneggiatore]
```

- (b) Con XPath, trovare il numero di film presenti nel documento che abbiano “Animazione” come uno dei generi

```
count(//film[contains(../generi/text(),"Animazione")])
```

- (c) Con XQuery, stampare i film del 2001 raggruppati per voto

```
xquery version "1.0";  
for $r in distinct-values(fn:doc("exa.xml")//film[anno=2001]/voto/text())  
return  
  <voto valore="{ $r }">  
    { for $f in fn:doc("exa.xml")//film[anno=2001 and ./voto/text()=$r]  
      return  
        $f  
    }  
  </voto>
```

- (d) Con XQuery, stampare coppie di film con lo stesso regista

```
xquery version "1.0";  
for $f1 in distinct-values(fn:doc("exa.xml")//film)  
for $f2 in distinct-values(fn:doc("exa.xml")//film)  
where $f1 >> $f2 and $f2/regista/text()=$f1//regista/text()  
return  
  <coppiaFilm>  
    { $f1 } { $f2 }  
  </coppiaFilm >
```

Con XPath, trovare i film per i quali il regista è anche uno degli attori

```
//film[.//regista=//attore]
```

```
//film[./staff/regista=./staff/attore]
```

- (a) Con XPath, trovare il numero di film che hanno uno sceneggiatore il cui nome contiene Hayao

```
count(//film[contains(./sceneggiatore/text(),"Hayao")])
```

- (b) Con XQuery , stampare i registi e per ogni regista i film dell'anno 2001

```
xquery version "1.0";
for $r in distinct-values(fn:doc("exa.xml")//regista)
return
  <regista nome="{ $r }">
    { for $f in fn:doc("exa.xml")//film[anno=2001 and ./regista/text()=$r]
      return $f
    }
  </regista>
```

- (c) Con XQuery, stampare coppie di film con lo stesso voto

```
xquery version "1.0";
for $f1 in distinct-values(fn:doc("exa.xml")//film)
for $f2 in distinct-values(fn:doc("exa.xml")//film)
where $f1 >> $f2 and $f2/voto/text()=$f1/voto/text()
return
  <coppiaFilm>
    { $f1 } { $f2 }
  </coppiaFilm >
```