

Esercizi XQuery

Disheng Qiu
disheng.qiu@gmail.com

Luca Rossi
luca.rossi.917@gmail.com

Evaluate XQuery

Online evaluator:

<http://videlibri.sourceforge.net/cgi-bin/xidelcgi>

Commercial editors (free 30-days trial):

<http://www.oxygenxml.com/>

```
<?xml version="1.0" encoding="UTF-8"?>
<studenti>
  <studente matricola="281283">
    <corsostudi>Ingegneria Informatica</corsostudi>
    <anagrafica>
      <cognome>Rossi</cognome>
      <nome>Luca</nome>
      <eta>24</eta>
    </anagrafica>
    <esami>
      <esame corso="Sistemi Informativi">
        <data>
          <aa>2012</aa>
          <mm>5</mm>
          <gg>29</gg>
        </data>
        <voto>29</voto>
      </esame>
      <esame corso="Analisi">
        <data>
          <aa>2012</aa>
          <mm>4</mm>
          <gg>28</gg>
        </data>
        <voto>29</voto>
      </esame>
    </esami>
  </studente>
  <studente matricola="281374">
    ...
  </studente>
</studenti>
```

XML Instance

Problemi

- I. Per ogni studente di “Ingegneria Informatica”, un elemento carriera contenente i dati dello studente, l’elenco di tutti gli esami ordinati per data, e il voto.

Soluzioni

```
xquery version "1.0";
for $s in fn:doc("studenti.xml")//studente
where $s/corsostudi = "Ingegneria Informatica"
return
  <carriera>
  <info-studente matricola="{ $s/@matricola }">
    { $s/anagrafica/node() }
  </info-studente>
  {
    for $e in $s/esami/esame
    order by $e/data, $e/voto
    return ($e)
  }
```

Problemi

1. Per ogni studente di “Ingegneria Informatica”, un elemento carriera contenente i dati dello studente, l’elenco di tutti gli esami ordinati per data, e il voto.
2. Per ogni corso, un elemento risultati contenente il nome del corso come attribute, e l’elenco di tutti gli studenti che hanno sostenuto l’esame con il relativo voto.

Soluzioni

```
xquery version "1.0";
for $c in distinct-values(//esame/@corso)
return
  <risultati corso="{ $c }">
    {for $s in //studente
     where $s//esame/@corso = $c
     return
      <studente matricola="{ $s/@matricola }">
        { $s/anagrafica/node() }
        { $s//esame[@corso=$c]/voto }
      </studente>
    }
  </risultati>
```

Problemi

1. Per ogni studente di “Ingegneria Informatica”, un elemento carriera contenente i dati dello studente, l’elenco di tutti gli esami ordinati per data, e il voto.
2. Per ogni corso, un elemento risultati contenente il nome del corso come attribute, e l’elenco di tutti gli studenti che hanno sostenuto l’esame con il relativo voto.
3. Per ogni corso, un elemento statistiche contenente il nome del corso, il numero di studenti che hanno sostenuto l’esame e il voto medio.

Soluzioni

```
xquery version "1.0";
for $c in distinct-values(//esame/@corso)
return
  <statistiche>
    <corso>
      {$c}
    </corso>
    <studenti-promossi>
      {count(//studente[.//esame/@corso = $c ])}
    </studenti-promossi>
    <voto-medio>
      {avg(//studente//esame[@corso = $c ]/voto)}
    </voto-medio>
  </statistiche>
```

Problemi

1. Per ogni studente di “Ingegneria Informatica”, un elemento carriera contenente i dati dello studente, l’elenco di tutti gli esami ordinati per data, e il voto.
2. Per ogni corso, un elemento risultati contenente il nome del corso come attribute, e l’elenco di tutti gli studenti che hanno sostenuto l’esame con il relativo voto.
3. Per ogni corso, un elemento statistiche contenente il nome del corso, il numero di studenti che hanno sostenuto l’esame e il voto medio.
4. Coppie di studenti che hanno sostenuto esattamente gli stessi esami.

Soluzioni

```
xquery version "1.0";  
for $s1 in //studente  
for $s2 in //studente
```

```
let $e1 := for $a in $s1//@corso  
order by $a  
return $a
```

```
let $e2 := for $a in $s2//@corso  
order by $a  
return $a
```

```
where $s2 >> $s1 and deep-equal($e1, $e2)  
return
```

```
  <coppia-studente>  
    <studente>  
      {$s1/anagrafica/node()}  
    </studente>  
    <studente>  
      {$s2/anagrafica/node()}  
    </studente>  
  </coppia-studente>
```

Problemi

1. Per ogni studente di “Ingegneria Informatica”, un elemento carriera contenente i dati dello studente, l’elenco di tutti gli esami ordinati per data, e il voto.
2. Per ogni corso, un elemento risultati contenente il nome del corso come attribute, e l’elenco di tutti gli studenti che hanno sostenuto l’esame con il relativo voto.
3. Per ogni corso, un elemento statistiche contenente il nome del corso, il numero di studenti che hanno sostenuto l’esame e il voto medio.
4. Coppie di studenti che hanno sostenuto esattamente gli stessi esami.
5. Una funzione utente che, data la matricola di uno studente, restituisce un elenco degli esami con data e voto, raggruppati per anni.

Soluzioni

```
xquery version "1.0";
declare namespace xs = "http://www.w3.org/2001/XMLSchema";
declare function utente($mat as xs:string) as node()*
{
  for $aa in distinct-values(//studente[@matricola=$mat]//aa)
  return
    <esami-anno anno="{ $aa }">
      {for $esa in //studente[@matricola=$mat]//esame[.//aa = $aa]
      return
        <esame>
          { $esa/voto }
          { $esa/data }
        </esame>
      }
    </esami-anno>
};
<utente matricola="281283">
{ utente("281283") }
</utente>
```