

XQuery

Basi di dati 2

Disheng Qiu
disheng.qiu@gmail.com

Luca Rossi
luca.rossi.917@gmail.com

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 data, un elemento esami contenente la data e l'elenco di tutti gli esami tenuti quel giorno, senza ripetizioni.
3. Per ogni corso, un elemento risultati contenente il nome del corso come attribute l'elenco di tutti gli studenti che hanno sostenuto l'esame con il relativo voto.
4. Per ogni corso, un elemento statistiche contenente il nome del corso, il numero di studenti che hanno sostenuto l'esame e il voto medio.
5. Coppie di studenti che hanno sostenuto esattamente gli stessi esami.
6. Una funzione utente che, data la matricola di uno studente, restituisce un elenco degli esami con data e voto, raggruppati per anni.

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.

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)
    }
  </carriera>
```

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 data, un elemento esami contenente la data e l'elenco di tutti gli esami tenuti quel giorno, senza ripetizioni. to l'esame e il voto medio.

Soluzioni

```
xquery version "1.0";
declare namespace xs = "http://www.w3.org/2001/XMLSchema";
for $d in fn:doc("studenti.xml")//data
where not($d/preceding::data[./gg = $d/gg and ./mm=$d/mm and ./aa=$d/aa])
order by xs:integer($d/aa),
xs:integer($d/mm),
xs:integer($d/gg)
return
  <esami>
    {$d}
    {for $e in distinct-values(fn:doc("studenti.xml")//esame[./data[./gg=$d/gg and ./
mm=$d/mm and ./aa=$d/aa]]/@corso)
      return
        <esame>
          {$e}
        </esame>
    }
  </esami>
```

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 data, un elemento esami contenente la data e l'elenco di tutti gli esami tenuti quel giorno, senza ripetizioni.
3. Per ogni corso, un elemento risultati contenente il nome del corso come attribute 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(fn:doc("studenti.xml")//esame/@corso)
return
  <risultati corso="{ $c }">
    {for $s in fn:doc("studenti.xml")//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 data, un elemento esami contenente la data e l'elenco di tutti gli esami tenuti quel giorno, senza ripetizioni.
3. Per ogni corso, un elemento risultati contenente il nome del corso come attribute l'elenco di tutti gli studenti che hanno sostenuto l'esame con il relativo voto.
4. 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 $c in distinct-values(fn:doc("studenti.xml")//esame/@corso)
return
  <statistiche>
    <corso>
      {$c}
    </corso>
    <studenti-promossi>
      {count(fn:doc("studenti.xml")//studente[.//esame/@corso = $c ])}
    </studenti-promossi>
    <voto-medio>
      {avg(fn:doc("studenti.xml")//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 data, un elemento esami contenente la data e l'elenco di tutti gli esami tenuti quel giorno, senza ripetizioni.
3. Per ogni corso, un elemento risultati contenente il nome del corso come attribute l'elenco di tutti gli studenti che hanno sostenuto l'esame con il relativo voto.
4. Per ogni corso, un elemento statistiche contenente il nome del corso, il numero di studenti che hanno sostenuto l'esame e il voto medio.
5. Coppie di studenti che hanno sostenuto esattamente gli stessi esami.

Soluzioni

```
xquery version "1.0";  
for $s1 in fn:doc("studenti.xml")//studente  
for $s2 in fn:doc("studenti.xml")//studente
```

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

```
let $e2 := for $a in $s2//esame  
order by $a/@corso  
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 data, un elemento esami contenente la data e l'elenco di tutti gli esami tenuti quel giorno, senza ripetizioni.
3. Per ogni corso, un elemento risultati contenente il nome del corso come attribute l'elenco di tutti gli studenti che hanno sostenuto l'esame con il relativo voto.
4. Per ogni corso, un elemento statistiche contenente il nome del corso, il numero di studenti che hanno sostenuto l'esame e il voto medio.
5. Coppie di studenti che hanno sostenuto esattamente gli stessi esami.
6. 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 local:utente($mat as xs:integer) as node()*
{
  for $aa in distinct-values(fn:doc("studenti.xml")//studente[@matricola=$mat]//aa)
  return
    <esami-anno anno="{ $aa }">
      {for $esa in fn:doc("studenti.xml")//studente[@matricola=$mat]//esame[.//aa = $aa]
      return
        <esame>
          { $esa/voto }
          { $esa/data }
        </esame>
      }
    </esami-anno>
};
<utente matricola="430910">
{ local:utente("430910") }
</utente>
```