



## Базе података 2

– XQuery –

– Задаци –

За дате XML фајлове написати одговарајуће XQuery упите:

1. Дохватите све производе из каталога којима је dept атрибут једнак ACC.
2. Формирати листу производа из каталога којима је dept атрибут једнак ACC.
3. Потребно је исписати сваку ставку нарудзбине у следећем формату:  
`<item num="..." name="..." quan="..."/>`
4. За дату нарудзбину потребно је исписати колико ставки је са ког одељења. Резултат приказати у формату:  
`<department name="..." totQuantity="..."/>`
5. Исписати називе производа из каталога који се налазе на одељењима са називом ACC или WMN.
6. Исписати бројеве од 1 до 3, сваки у засебном тагу oneEval.
7. Исписати све парове бројева где је први елемент из скупа (1,2), а други из скупа (a,b), сваки у засебном тагу oneEval.
8. Исписати бројеве од 1 до 3, резултат треба да се налази у тагу oneEval.
9. Исписати све производе из каталога, који имају број већи од 100, почињу на слово F, имају опције за боју и налазе се на одељењу под називом ACC и WMN.
10. Проверити да ли постоји производ који се налази на одељењу ACC.
11. Проверити да ли се сви производи налазе на одељењу ACC.
12. Проверити да ли постоји производ који се не налази на одељењу ACC.
13. Проверите да ли неки од парова бројева ( $i, j$ ), где је  $i$  из скупа 1,2,3, а  $j$  из скупа 10, 11, имају разлику 7.
14. Исписати све производе из каталога, у формату:  
`<result dept="..." number="..."/>`  
Дупликате је потребно исписати само једном.

15. Исписати све ставке наруџбине у формату:  
`<item num="..." name="..." quan="..."/>`.

16. Исписати све ставке наруџбине у формату:  
`<item num="..." name="..." price="..."/>`

17. Исписати све производе у формату:  
`<product number="..." price="..." />`

За производе који немају цену, потребно је оставити цену празну.

– Решења –

1. 

```
for $prod in doc("catalog.xml")/catalog/product
where $prod/@dept = "ACC"
order by $prod/name
return $prod/name

или

for $prod in doc("catalog.xml")/catalog/product
let $name := $prod/name
where $prod/@dept = "ACC"
order by $name
return $name
```
2. 

```
<ul type="square">{
  for $prod in doc("catalog.xml")/catalog/product
  where $prod/@dept='ACC'
  order by $prod/name
  return <li class="{ $prod/@dept}">{data($prod/name)}</li>
}</ul>
```
3. 

```
for $item in doc("order.xml")//item
let $name := doc("catalog.xml")//product[number = $item/@num]/name
return <item num="{ $item/@num}"
      name="{ $name}"
      quan="{ $item/@quantity}"/>
```
4. 

```
for $i in doc("order.xml")//item
let $d := $i/@dept
group by $d
order by $d
return <department name="{ $d}" totQuantity="{sum($i/@quantity)}"/>
```

5. for \$prod in doc("catalog.xml")//product  
 let \$prodDept := \$prod/@dept  
 where \$prodDept = "ACC" or \$prodDept = "WMN"  
 return \$prod/name  
  
 или  
  
 let \$doc := doc("catalog.xml")  
 for \$prod in \$doc//product  
 let \$prodDept := \$prod/@dept  
 let \$prodName := \$prod/name  
 where \$prodDept = "ACC" or \$prodDept = "WMN"  
 return \$prodName
6. for \$i in 1 to 3  
 return <oneEval>{\$i}</oneEval>
7. for \$i in (1, 2)  
 for \$j in ("a", "b")  
 return <oneEval>i is {\$i} and j is {\$j}</oneEval>  
  
 или  
  
 for \$i in (1, 2), \$j in ("a", "b")  
 return <oneEval>i is {\$i} and j is {\$j}</oneEval>
8. let \$i := (1 to 3)  
 return <oneEval>{\$i}</oneEval>
9. for \$prod in doc("catalog.xml")//product  
 let \$prodDept := \$prod/@dept  
 where \$prod/number > 100  
     and starts-with(\$prod/name, "F")  
     and exists(\$prod/colorChoices)  
     and (\$prodDept = "ACC" or \$prodDept = "WMN")  
 return \$prod
10. some \$dept in doc("catalog.xml")//product/@dept  
 satisfies (\$dept = "ACC")
11. every \$dept in doc("catalog.xml")//product/@dept  
 satisfies (\$dept = "ACC")
12. some \$dept in doc("catalog.xml")//product/@dept  
 satisfies (\$dept != "ACC")

13. some  $\$i$  in (1 to 3),  $\$j$  in (10, 11)  
satisfies  $\$j - \$i = 7$

```
14. let $prods := doc("catalog.xml")//product
    for $d in distinct-values($prods/@dept),
        $n in distinct-values($prods[@dept = $d]/number)
    return <result dept="{ $d}" number="{ $n}"/>
```

```
15. for $item in doc("order.xml")//item,
    $prod in doc("catalog.xml")//product[number = $item/@num]
    return <item num="{ $item/@num}"
        name="{ $prod/name}"
        quan="{ $item/@quantity}"/>
```

или

```
for $item in doc("order.xml")//item,
    $prod in doc("catalog.xml")//product
where $item/@num = $prod/number
return <item num="{ $item/@num}"
    name="{ $prod/name}"
    quan="{ $item/@quantity}"/>
```

```
16. for $item in doc("order.xml")//item,
    $prod in doc("catalog.xml")//product,
    $price in doc("prices.xml")//prod
where $item/@num = $prod/number and $prod/number = $price/@num
return <item num="{ $item/@num}"
    name="{ $prod/name}"
    price="{ $price/price}"/>
```

```
17. for $prod in doc("catalog.xml")//product
    return <product number="{ $prod/number}">{
        attribute price
        {for $price in doc("prices.xml")//prod
            where $prod/number = $price/@num
            return $price/price}
    }</product>
```

или

```
for $prod in doc("catalog.xml")//product
for $price allowing empty
    in doc("prices.xml")//prices/priceList/prod[@num = $prod/number]
return <product number="{ $prod/number}" price="{ $price/price}"/>
```