

# Data & Informatie antwoorden op voorbeeldvragen voor Toets 4 SQL/XML, SPARQL, RESTXQ, ‘De wiskunde erachter’ en Security

## Opgave 1: SQL/XML

### Vraag (a)

```
select XMLELEMENT(NAME director, person.name)
from directs, person
WHERE directs.pid=person.pid
```

### Vraag (b)

```
select XMLELEMENT(NAME directors,
XMLAGG(XMLELEMENT(NAME director, person.name)))
from directs, person
WHERE directs.pid=person.pid
```

### Vraag (c)

```
select
XMLELEMENT(NAME director,
  XMLELEMENT(NAME name, person.name),
  XMLAGG(XMLELEMENT(NAME movie, movie.name)))
from directs, person, movie
WHERE directs.pid=person.pid
AND directs.mid=movie.mid
GROUP BY person.pid
```

## Opgave 2: SPARQL

### Vraag (a)

Door ‘?beatrix a:name "Beatrix of Netherlands //”’ matcht ‘?beatrix’ alleen maar met de individual Beatrix. Hierdoor matcht ‘?family’ dmv ‘?beatrix a:spouseIn ?family’ alleen maar met de familie van Beatrix. So far so good. Echter, ‘?husband’ matcht met twee personen in het patroon ‘?husband a:spouseIn ?family’, namelijk Claus en Beatrix zelf. Dat de variabele ‘?husband’ heet, maakt niet uit. En er staat ook nergens dat ‘?husband’ iemand anders moet zijn dan ‘?beatrix’. Als gevolg daarvan zijn er dus ook twee mogelijke matches voor ‘?n’.

### Vraag (b)

```
SELECT ?t ?ft WHERE
{
  ?b a:year "2000" .
  ?b a:title ?t .
  OPTIONAL { ?fb f:original ?b . ?fb f:titre ?ft }
}
```

## Opgave 3: RESTXQ

De functie-declaratie moet veranderen in:

```
function page:hello($a as xs:string, $b as xs:string)
```

omdat de argumenten nu strings zijn.

Tussen de ‘{’ en ‘}’ kun je willekeurige XQuery’s zetten. Dus het is simpelweg een kwestie van de ‘<p>...</p>’ regel veranderen in

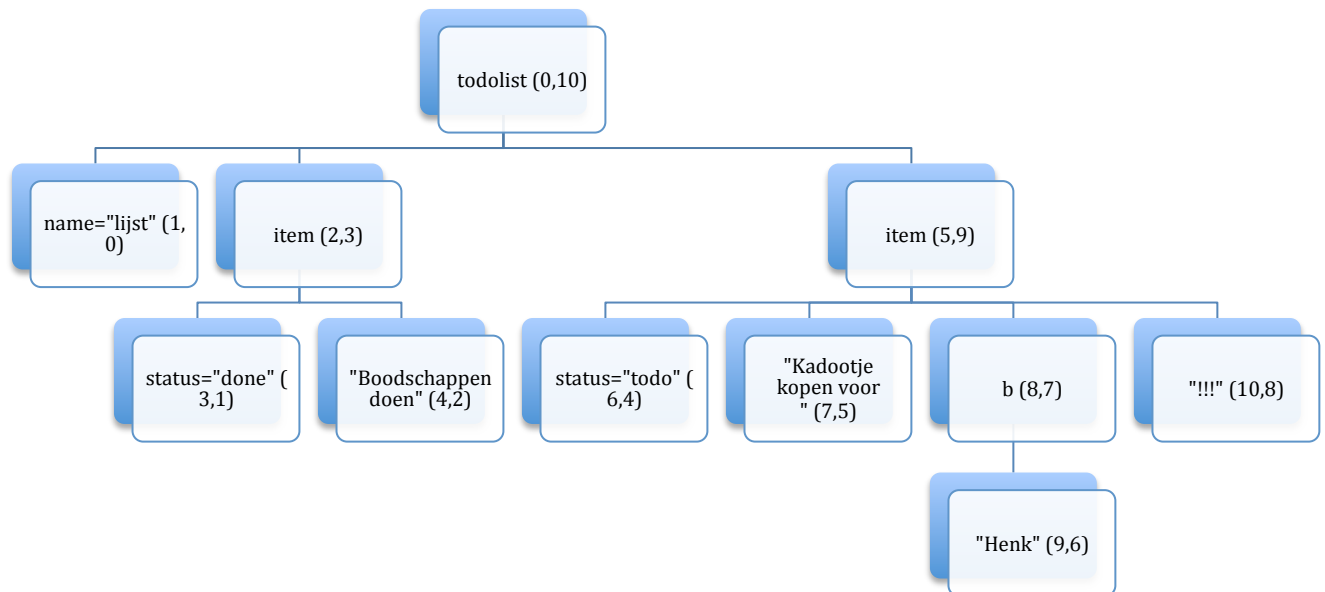
```
<p>De som van {$a} en {$b} is
{ //getal[@naam=$a] + //getal[@naam=$b] = { //getal[$naam=$a] + //getal[$naam=$b] }
.</p>
```

of beter een paar let's introduceren als volgt:

```
function page:hello($a as xs:string, $b as xs:string)
{
  let $getala = //getal[$naam=$a]
  let $getalb = //getal[$naam=$b]
  return
  <html>
  <h1>Optelling</h1>
  <p>De som van { $a } en { $b } is { $getala } + { $getalb } = { $getala + $getalb }.</p>
  </html>
};
```

## Opgave 4: 'De wiskunder erachter'

### Vraag (a)



pre-order rank en post-order rank tussen haakjes.

**NB:** Een attribuut is één node. Een element met tekst zoals “<b>Henk</b>” is twee nodes!

### Vraag (b)

Een ancestor node van dit element moet een pre-order rank hebben die kleiner is dan 14 en een post-order rank groter dan 5.

### Vraag (c)

$$P('zege', d_1) = 1/5$$

$$P('zege', d_2) = 0/5 = 0$$

$$P('zege', d_3) = 1/7$$

### Vraag (d)

Dit wordt direct verdisconteerd in  $P(D)$ . Het is namelijk een eigenschap van het document zelf en onafhankelijk van de query. Indirect is het daarmee ook verdisconteerd in  $P(D|T_1, \dots, T_n)$ .

### Vraag (e)

(v) is de beste keuze. Hiermee dwing je af dat een object/persoon én een man is, én in een relatie 'heeftKind' staat met een object/persoon van type Kind, én dan alle andere objecten/personen waarmee het/hij in relatie 'heeftKind' staat ook van type Kind moeten zijn.

(i) zoek ik ook goed rekenen. Het laat toe dat de man in een relatie 'heeftKind' staat van een ander type dan Kind.

### Vraag (f)

Stel we kiezen (i). Een voorbeeld is een kinderloze kunstenaar die zijn create beschouwt als zijn kind ... dwz er is dan een 'heeftKind' relatie (we voldoen dus aan de beschrijving), maar met een object van een ander type dan kind; dat maakt 'm nog geen vader ... beetje vergezocht, maar toch.

Stel we kiezen (iv). Een vrouw met of zonder kinderen voldoet hier aan.

## Opgave 5: Security

### Vraag (a)

A certificate consists of identity information, along with the algorithms used for encryption, hashing and signatures. The public keys of the owner are also given in the certificate. The certificates also contain the date information during which the certificate is valid. An expired certificate can be misused by an attacker if that computer does not check the dates properly, for example pretending to be someone else.

### Vraag (b)

That hash function can generate  $2^{80}$  different hash values. This number is large enough to produce hash values for the whole car parts. However, in order to prevent collusions, a larger size of has values, such as 128, 196 or 259 bits, can be used.

### Vraag (c)

- (i) `assert(z < 3);`  
`z < 0` is not needed since `z` is unsigned and cannot obtain negative values.
- (ii) symbolic values: `X == x, Y == y, Z1 == X + 5, Z2 == Z1 * 2`  
path predicate: `Z2 <= 4 & Z2 > 2`, the second part is needed to trigger the buffer overflow.
- (iii) We require: `Z2 <= 4 & Z2 > 2`, thus either `Z2 = 3`, or `Z2 = 4`.  
`Z2 = 3` gives `Z1 = 1.5`, which is impossible. `Z2 = 4` gives `Z1 = 2`, and therefore `X = 2 - 5 = (overflow) 253`. `Y` can be any value.

### Vraag (d)

- (i) For example add a loop `3a,b,c` after `2`. which runs:
  - 3. For all downloaded files:
    - a. Feed pdf file to program
    - b. Count amount of newly covered code
    - c. If this is 0, filter out the pdf filePerform the remaining steps only for the non-filtered pdfs.
- (ii) This (drastically) improves the fuzzing speed. The new filtered set covers the same code as the non-filtered one, but is potentially much smaller. This way, more time can be spent to fuzz different parts of the code.  
A downside is that we possibly removed pdfs that executed the same code but

would results in a system crash with minor modification. Code coverage is only a heuristic.