

01

```
for $e in doc("employees.xml")//employee
where $e/lastName != "Smith"
return
  <employee>
    <number>{ data($e/@number) }</number>
    { $e/firstName }
    { $e/lastName }
  </employee>
```

... **where** \$e/lastName/text() != "Smith" ...

... **where** not(\$e/lastName = "Smith") ...

... **where** \$e/lastName ne "Smith" ...

... **where** not(\$e/lastName eq "Smith") ...

02

```
for $e in doc("employees.xml")//employee
let $n := concat($e/firstName, " ", $e/lastName)
return <employee>{ $n }</employee>
```

```
... let $n := concat($e/firstName/text(), " ", $e/lastName/text()) ...
... return element employee { text { $n } }
... return element employee { $n }
... return element { "employee" } { text { $n } }
```

```
for $e in doc("employees.xml")//employee
return <employee>{ concat($e/firstName, " ", $e/lastName) }</employee>
... return <employee>{ $e/firstName/text() } { " " } { $e/lastName/text() }</employee>
... return element employee { text { concat($e/firstName, " ", $e/lastName) } }
... return element employee { text { $e/firstName }, " ", text { $e/lastName } }
```

Incorrect:

```
... return <employee>{ $e/firstName } { " " } { $e/lastName }</employee>
```

03

```
for $e in doc("employees.xml")//employee
where ($e/salary > 2300) and ($e/worksIn != "D1.1")
order by $e/salary, $e/lastName descending
return text { $e/email }
```

```
... where ($e/salary > 2300) and not($e/worksIn = "D1.1") ...
... return text { $e/email/text() }
... return $e/email/text()
```

Incorrect:

```
... return { $e/email/text() }
... return $e/email
```

04

```
for $d in doc("departments.xml")//department
let $c := count($d/descendant::employee)
order by $c
return <department employees="{ $c }">{ $d/@id }{ $d/name/text() }</department>
```

```
... let $c := count($d//employee) ...
```

```
... order by $c ascending ...
```

```
... return
  <department employees="{ $c }" id="{ $d/@id }">
    { $d/name/text() }
  </department>
```

```
... return
  element department {
    attribute employees { $c },
    attribute id { $d/@id },
    text { $d/name/text() }
  }
```

```
... return
  element department {
    attribute employees { $c },
    attribute id { data($d/@id) },
    text { $d/name }
  }
```

Incorrect:

```
... return <department employees="{ $c }">{ $d/@id }{ $d/name }</department>
```

05

```
let $s := doc("departments.xml")//manager
let $x := max(
  for $m in $s
  return count($m/../../employee)
)
for $m in $s
where count($m/../../employee) = $x
return <manager>{ concat($m/firstName, " ", $m/lastName) }</manager>
```

```
let $s := doc("departments.xml")//manager
let $i :=
  for $m in $s
  return
    <item employees="{ count($m/../../employee) }">
      { concat($m/firstName, " ", $m/lastName) }
    </item>
let $x := max($i/@employees)
for $m in $i
where $m/@employees = $x
return <manager>{ $m/text() }</manager>
```

```
let $s := doc("departments.xml")//manager
let $x := max($s/count(../../employee))
for $m in $s
where count($m/../../employee) = $x
return <manager>{ concat($m/firstName, " ", $m/lastName) }</manager>
```

```
let $s := doc("departments.xml")//manager
let $x := max($s/count(../../employee))
for $m in $s/.[count(../../employee) = $x]
where true()
return <manager>{ concat($m/firstName, " ", $m/lastName) }</manager>
```

```
let $s := doc("departments.xml")//manager, $x := max($s/count(../../employee)) ...
```

06

```
<html>
  <head><title>Departments</title></head>
  <body>
    <table>
      {
        for $d in doc("departments.xml")//department
        let $s := $d/ancestor-or-self::department/manager
        let $m := $s[last()]
        order by $d/@id
        return
          <tr>
            <td>{ data($d/@id) }</td>
            <td>{ $d/name/text() }</td>
            <td>{
              if ($s)
                then concat($m/firstName, " ", $m/lastName)
                else <i>Unknown</i>
            }</td>
          </tr>
      }
    </table>
  </body>
</html>
```