

# DSA5104 Practice - XML

Zhao Peiduo

AY2025/26 Sem1

## Questions

Questions covering basic concepts in XQuery generated by ChatGPT for practice. Please pull the sample data file from Github page as well.

1. List all unique department names.
2. Find all level attributes of course with more than 3 credits.
3. List the id of instructors with salary above 100000.
4. Join instructors and departments by dept\_name and show instructor name and building. The result should be in format "instructor\_name-building", for instance "Instructor\_0 - Curie".
5. Find course id(s) taught by Instructor\_5. Return the text content only instead of its wrapping element.
6. Count how many courses each department offers. The result should be in format "department:number of courses, for instance "Physics: 6". Sort the result in descending order.
7. Find the maximum salary in each department. The result should be in format "department:maximum salary, for instance "Physics: 136896". Sort the result in descending order.
8. List courses and the number of instructors teaching each course. The result should be in format "course.id:count, for instance "COM-139: 3". Sort the result in descending order.
9. Find departments with total salary expenditure over 300000. The result should be in format "dept\_name:total.expenditure, for instance "Biology: 432551".
10. List instructors who do not teach any course.
11. List courses that have no assigned instructor.
12. Define a function `local:getSalaryStatus($salary)` that returns "High" if salary is higher 100000 else "Normal". Use this function to list all instructors in the format "Instructor\_X: High".
13. Nested query: instructors teaching a course in the Biology department.
14. List courses whose title contains '5'.
15. Return instructors and the number of courses they teach.
16. Sort courses by dept then title.
17. Group instructors by building. Count how many instructors belong to each building. Return results in format "Building:Count".

# Answers

## 1. Query:

```
for $d in /university/department
return distinct-values($d/dept_name)
```

## Output:

```
Comp. Sci.
Biology
Mathematics
Physics
History
Chemistry
```

## 2. Query:

```
for $c in /university_gpt/course[credits > 3]
return $c/@level
```

## Output:

```
level="pg"
level="pg"
level="ug"
level="ug"
level="ug"
level="ug"
level="ug"
level="ug"
level="ug"
level="ug"
level="ug"
```

## 3. Query:

```
for $i in /university/instructor
where $i/salary > 100000
return $i/name
```

## Output:

```
Instructor_4
Instructor_7
Instructor_9
Instructor_11
Instructor_12
Instructor_13
Instructor_14
Instructor_18
Instructor_19
Instructor_23
```

4. Query:

```
for $i in /university/instructor
let $d := /university/department[dept_name = $i/dept_name]
return concat($i/name, " - ", $d/building)
```

Output:

```
Instructor_0 - Curie
Instructor_1 - Einstein
Instructor_2 - Heritage
Instructor_3 - Watson
Instructor_4 - Taylor
Instructor_5 - Heritage
Instructor_6 - Heritage
Instructor_7 - Taylor
Instructor_8 - Curie
Instructor_9 - Heritage
```

5. Query:

```
for
  $t in /university/teaches,
  $i in /university/instructor
where $i/name = "Instructor_5" and $t/IID = $i/IID
return $t/course_id/text()
```

Output:

```
HIS-119
```

6. Query:

```
for $d in /university/department
let $count := count(/university/course[dept_name = $d/dept_name])
order by $count descending
return concat($d/dept_name, ":", $count)
```

Output:

```
Mathematics: 11
History: 9
Physics: 6
Chemistry: 6
Comp. Sci.: 5
Biology: 3
```

7. Query:

```
for $d in /university/department
let $max_salary := max(/university/instructor[dept_name = $d/dept_name]/salary)
order by $max_salary descending
return concat($d/dept_name, ":", $max_salary)
```

**Output:**

```
Mathematics: 145505
History: 142058
Physics: 136896
Biology: 136834
Comp. Sci.: 129596
Chemistry: 114089
```

**8. Query:**

```
for $c in /university/course
let $n := count(/university/teaches[course_id = $c/course_id])
order by $n descending
return concat($c/course_id, ": ", $n)
```

**Output:**

```
MAT-102: 4
PHY-109: 3
CHE-136: 3
COM-139: 3
PHY-101: 2
CHE-106: 2
COM-128: 2
HIS-133: 2
MAT-138: 2
HIS-105: 1
```

**9. Query:**

```
for $d in distinct-values(/university/instructor/dept_name)
let $total := sum(/university/instructor[dept_name=$d]/salary)
where $total > 300000
return concat($d, ": ", $total)
```

**Output:**

```
Chemistry: 494646
History: 813268
Biology: 432551
Comp. Sci.: 447957
Mathematics: 548290
```

**10. Query:**

```
for $i in doc("uni_gpt.xml")/university/instructor
where not(doc("uni_gpt.xml")/university/teaches[IID = $i/IID])
return $i/name
```

**Output:**

```

<name>Instructor_2</name>
<name>Instructor_4</name>
<name>Instructor_7</name>
<name>Instructor_12</name>
<name>Instructor_13</name>
<name>Instructor_14</name>
<name>Instructor_16</name>
<name>Instructor_17</name>

```

#### 11. Query:

```

for $c in doc("uni_gpt.xml")/university/course
where not(doc("uni_gpt.xml")/university/teaches[course_id = $c/course_id])
return $c/course_id

```

#### Output:

```

<course_id>PHY-100</course_id>
<course_id>PHY-103</course_id>
<course_id>CHE-104</course_id>
<course_id>MAT-108</course_id>
<course_id>MAT-110</course_id>
<course_id>BIO-111</course_id>
<course_id>MAT-114</course_id>
<course_id>MAT-116</course_id>
<course_id>MAT-117</course_id>
<course_id>COM-120</course_id>

```

#### 12. Query:

```

declare function local:getSalaryStatus($s as xs:integer) as xs:string {
  if ($s > 100000) then "High" else "Normal"
};

for $i in /university/instructor
return concat($i/name, ": ", local:getSalaryStatus($i/salary))

```

#### Output:

```

"Instructor_25: High"
"Instructor_0: Normal"
"Instructor_1: Normal"
"Instructor_2: Normal"
"Instructor_3: Normal"
"Instructor_4: High"
"Instructor_5: Normal"
"Instructor_6: Normal"
"Instructor_7: High"
"Instructor_8: Normal"

```

#### 13. Query:

```

for $i in /university/instructor
where some $t in /university/teaches satisfies ($t/IID=$i/IID)

```

```
and /university/course[course_id=$t/course_id]/dept_name="Biology")
return $i/name
```

**Output:**

```
<name>Instructor_20</name>
<name>Instructor_21</name>
```

**14. Query:**

```
for $c in /university/course
where contains($c/title, "5")
return $c/course_id
```

**Output:**

```
HIS-105
COM-115
MAT-125
PHY-135
```

**15. Query:**

```
for $i in /university/instructor
let $n := count(/university/teaches[IID=$i/IID])
return concat($i/name, ": ", $n)
```

**Output:**

```
Instructor_0: 1
Instructor_1: 3
Instructor_2: 0
Instructor_3: 2
Instructor_4: 0
Instructor_5: 1
Instructor_6: 1
Instructor_7: 0
Instructor_8: 2
Instructor_9: 2
```

**16. Query:**

```
for $c in /university/course
order by $c/dept_name, $c/title
return concat($c/dept_name, " - ", $c/title)
```

**Output:**

```
Biology - Course Title 11
Biology - Course Title 13
Biology - Course Title 23
Chemistry - Course Title 18
```

Chemistry - Course Title 22  
Chemistry - Course Title 31  
Chemistry - Course Title 36  
Chemistry - Course Title 4  
Chemistry - Course Title 6  
Comp. Sci. - Course Title 15

**17. Query:**

```
let $depts := /university/department
for $b in distinct-values($depts/building)
let $deptNames := $depts[building = $b]/dept_name
let $count :=
    count(/university/instructor[dept_name = $deptNames])
return concat($b, ": ", $count)
```

**Output:**

"Taylor: 4"  
"Watson: 4"  
"Newton: 6"  
"Einstein: 3"  
"Heritage: 8"  
"Curie: 6"