

Partiendo de universidad.xml obtener utilizando XQUERY las siguientes consultas:

1. Obtener los nombres de los empleados de los departamentos tipo A, cuyo puesto es profesor  

```
for $prof in /universidad/departamento[@tipo='A']/empleado
where $prof/puesto='Profesor'
return $prof/nombre
```
2. Devolver el nombre del departamento encerrado entre las etiquetas tipo A si es de este tipo y de tipo B para cualquier otro.  

```
for $dep in /universidad/departamento
return if ($dep/@tipo='A')
then <tipoA>{data($dep/nombre)}</tipoA>
else <tipoB>{data($dep/nombre)}</tipoB>
```
3. Obtener los nombres de los departamentos y los empleados que tiene entre etiquetas.  

```
for $dep in /universidad/departamento
let $nom:=$dep/empleado
return <depart>{data($dep/nombre)}<emple>{count($nom)}</emple></depart>
```
4. Obtener los nombres de los departamentos, los empleados que tiene y la mitad del salario entre etiquetas.  

```
for $dep in /universidad/departamento
let $emp:=$dep/empleado
let $sal:=$dep/empleado/@salario
return
<depart>{data($dep/nombre)}<emple>{count($emp)}</emple><medsal>{avg($sal)}</medsal></depart>
```

Partiendo de emp.xml y dept.xml obtener utilizando XQUERY las siguientes consultas:

1. Devolver los nombres de los empleados, los directores entre etiqueta director y el resto entre emple.  

```
for $emp in /empleados/empleado
order by $emp/apellido
return if ($emp/oficio='DIRECTOR')
then
<DIRECTOR>{$emp/apellido/text()}</DIRECTOR>
else
<EMPLE>{data($emp/apellido)}</EMPLE>
```
2. Devolver el nombre de los oficios que empiezan por P  

```
for $ofi in /empleados/empleado/oficio
where starts-with(data($ofi), 'P')
return $ofi
```
3. Obtener el nombre de los oficios y los empleados de cada uno de ellos.  

```
for $ofi in distinct-values(/empleados/empleado/oficio)
let $cu:=count(/empleados/empleado[oficio=$ofi])
return concat($ofi, ' = ', $cu)
```
4. Obtener el número de empleados que tiene cada departamento y la media del salario redondeada.  

```
for $dep in distinct-values(/empleados/empleado/dept_no)
let $cu:=count(/empleados/empleado[dept_no=$dep])
let $sala:=round(avg(/empleados/empleado[dept_no=$dep]/salario))
return concat('Departamento: ', $dep, '. Numero de empleados = ', $cu, '. Media del salario = ', $sala)
```

5. Visualizar por cada empleado su apellido, su número de departamento y el nombre del departamento.

```
for $emp in (/empleados/empleado)
let $emple:= $emp/apellido
let $dep:= $emp/dept_no
let $dnom:=(/departamentos/departamento[dept_no=$dep]/dnombre)
return <res>{$emple, $dep, $dnom}</res>
```

6. Obtener por cada departamento, el nombre del departamento, el número de empleados y la media de los salarios.

```
for $dep in /departamentos/departamento
let $d:=$dep/dept_no
let $tot:=sum(/empleados/empleado[dept_no=$d]/salario)
let $cu:=count(/empleados/empleado[dept_no=$d]/emp_no)
return
<resul>{$dep/dnombre}<sumsalario>{$tot}</sumsalario><totalemp>{$cu}</totalemp>
></resul>
```

7. Convertir la salida de la consulta anterior, de manera que el total salario y el total empleados sean atributos de cada departamento.

```
for $dep in /departamentos/departamento
let $d:=$dep/dept_no
let $tot:=sum(/empleados/empleado[dept_no=$d]/salario)
let $cu:=count(/empleados/empleado[dept_no=$d]/emp_no)
return concat('<departamento sumasalarario="',$tot,'" totalemp="',$cu,'">',
data($dep/dnombre), '</departamento>')
```

8. Obtener por cada departamento el nombre del empleado que más gana.

```
for $emp in /empleados/empleado
let $d:=$emp/dept_no, $nom:=$emp/apellido, $sal:=$emp/salario
let $ndep:=(/departamentos/departamento[dept_no=$d]/dnombre)
let $salmax:=max(/empleados/empleado[dept_no=$d]/salario)
return
if ($sal=$salmax)
then
<depar>{data($ndep)}<salmax>{data($sal)}</salmax><emple>{data($nom)}</emple></depar>
else()
```