

Trabajo práctico no. 6

XML

Los ejercicios que aparecen en este trabajo práctico deben ser resueltos empleando el SQL Server.

1 – Usando la base de datos de company conteste las siguientes preguntas.

XPath

- a. Obtener el nombre de los empleados.

```
select Doc.query('companyDB//employee/fname')  
from Company;
```

- b. **Obtener el nombre de los departamentos.**

- c. Obtener el nombre de los departamentos que tienen ubicaciones en "Houston".

```
select Doc.query('companyDB//department[locations/location="Houston"]/dname')  
from Company;
```

- d. **Obtener el nombre de los proyectos con ubicación en "Stafford".**

- e. Listar el nombre de los empleados con sueldo inferior a 30,000.

```
select Doc.query('companyDB//employee[salary<30000]/fname')  
from Company;
```

- f. **Escribir el nombre de los proyectos ubicados en una ciudad cuyo nombre inicie con una letra a partir de "N".**

- g. Mostrar el número de los proyectos controlados por el departamento 5.

```
select Doc.query('companyDB//department[@dno=5]/projectsControlled[@pnos]')  
from Company;
```

- h. **Mostrar el nombre de los empleados que están adscritos al departamento 7.**

- i. Obtener el nombre de los empleados que no tienen dependientes.

```
select Doc.query('companyDB//employee[not (dependents)]/fname')  
from Company;
```

- j. Escribir el nombre de los departamentos que tienen una ubicación en "Bellaire" o cuyo gerente fue nombrado en 1999.

```
select Doc.query('companyDB//department[locations/location="Bellaire"  
or contains(manager/startDate,"1999")]/dname')  
from Company;
```

- k. **Escribir el nombre de los empleados que trabajan para el departamento 6 o que tienen un salario superior a los 90,000.**

- l. Para los empleados que trabajan en el departamento 7, listar el nombre de su cónyuge ('Spouse').

```
select Doc.query('companyDB//employee[@worksFor=7]
//dependent[relationship="Spouse"]/dependentName')
from Company;
```
- m. Para los departamentos cuyo nombre inicia con una letra a partir de 'R', obtener el nombre de sus ubicaciones (location) siempre y cuando éstas inicien con una letra a partir de 'P'.

```
select Doc.query('companyDB//department[dname>="R"]//location[.>="P"]')
from Company;
```
- n. Listar el nombre de los departamentos que tienen una ubicación cuyo nombre inicie con una letra a partir de 'S'.

```
select Doc.query('companyDB//locations[location>="S"]/../dname')
from Company;
```
- o. **Mostrar el nombre de los proyectos que tienen empleados que les dedican más de 40 horas de trabajo.**

2 – Usando la base de datos de company conteste las siguientes preguntas.

XQuery

Consultas generales

- a. Escribir el nombre (completo) de los empleados en orden ascendente por apellido.

```
select Doc.query
('for $x in //employee
order by $x/lname
return <resul> {concat($x/fname," ", $x/minit,". ", $x/lname)} </resul>')
from Company;
```

- b. Escribir el nombre (completo) de los empleados para los cuales su nombre inicia con “J”.

```
select Doc.query
('for $x in //employee
where substring($x/fname,1,1) = "J"
return <resul> {concat($x/fname," ", $x/minit,". ", $x/lname)} </resul>')
from Company;
```

- c. Obtener el nombre de los proyectos que tienen trabajadores que le dedican un máximo de 5 horas al proyecto.

```
select Doc.query
('for $x in //project
where $x//worker <= 5
return <resul> { $x/pname } </resul>')
from Company;
```

- d. Escribir el nombre de los departamentos que inicien con una letra a partir de ‘L’.

- e. Mostrar el nombre de los proyectos que inician con la palabra "Product" y que tienen más de dos trabajadores.

```
select Doc.query
('for $x in //project
where contains($x/pname,"Product ") and count($x//worker)>2
return <resul> { $x/pname } </resul>')
from Company;
```

- f. Listar el nombre de los proyectos controlados por un departamento con clave mínima de 6.

- g. Igual al anterior, mostrando también la cantidad de empleados que tienen dichos proyectos.

Nota: usar *cast as xs:string?* para cada elemento escrito entre los tags de <resul>.

Diferencia entre *for* y *let*

- h. Escribir el nombre de los departamentos (usando For).

```
select Doc.query
('for $x in //department
return <resul> {$x//dname} </resul>')
from Company;
```

- i. Escribir el nombre de los departamentos (usando Let).

```
select Doc.query
('let $x := //department
return <resul> {$x//dname} </resul>')
from Company;
```

- j. Obtener el nombre y los datos del manager de cada departamento.

```
select Doc.query
('for $x in //department
let $y := $x/dname, $z := $x/manager
return <resul> {$y, $z} </resul>')
from Company;
```

Uso de las funciones de agregados

- k. Listar el promedio del salario que ganan los empleados.

- l. Mostrar el valor máximo de horas dedicadas a cada proyecto.

Nota: usar *cast as xs:string?* para cada elemento escrito entre los tags de <resul>.

Equijuntas

- m. Listar el nombre de los proyectos y el nombre de los empleados que trabajan en ellos, ordenando alfabéticamente en ambos casos.

```
select Doc.query
('for $p in //project, $e in //employee
where $p//@essn=$e/@ssn
order by $p/pname, $e/fname
return <resul> {$p/pname, $e/fname} </resul>')
from Company;
```

- n. Listar el nombre completo de los empleados y el nombre del departamento al que están adscritos, ordenando alfabéticamente por apellido del empleado.

Varias

- o. Obtener el nombre y el apellido de los empleados que trabajan en el departamento 6 y que contienen una i en su nombre completo.
- p. Mostrar el nombre de los departamentos que tienen más de dos ubicaciones junto con la cuenta de las mismas. **Nota:** se puede usar, además, *let* en la solución.