**Employee Management -MarkLogic Assignment**

1. **Create Document for each employee and add document to employee collection**

**Query:**

xquery version "1.0-ml";

let $doc:=

<employees>   
<employee>   
<empId>EMP001</empId>   
<name>John</name>   
<dob>1996-09-21</dob>   
<designation>Backend Developer</designation>   
<department>Dev</department>   
<department>RnD</department>   
<salary>10000</salary>   
<joiningdttm>2017-07-17T10:10:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP002</empId>   
<name>Mike</name>   
<dob>2001-02-11</dob>   
<designation>UI Developer</designation>   
<department>Dev</department>   
<department>RnD</department>   
<salary>20000</salary>   
<joiningdttm>2018-08-20T02:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP003</empId>   
<name>Clare</name>   
<dob>1992-07-11</dob>   
<designation>DBA</designation>   
<department>Dev</department>   
<department>RnD</department>   
<salary>30000</salary>   
<joiningdttm>2001-05-21T08:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP004</empId>   
<name>Marker</name>   
<dob>1995-09-21</dob>   
<designation>Backend Developer</designation>   
<department>Dev</department>   
<department>RnD</department>   
<salary>40000</salary>   
<joiningdttm>2011-04-01T07:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP005</empId>   
<name>Wade</name>   
<dob>1888-09-21</dob>   
<designation>AWS Admin</designation>   
<department>Ops</department>   
<department>RnD</department>   
<salary>90000</salary>   
<joiningdttm>2001-02-22T12:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP006</empId>   
<name>Laura</name>   
<dob>2005-12-01</dob>   
<designation>QA Manager</designation>   
<department>QA</department>   
<department>RnD</department>   
<salary></salary>   
<joiningdttm>2017-04-06T11:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP007</empId>   
<name>Anne</name>   
<dob>1996-09-07</dob>   
<designation>Backend Developer</designation>   
<department>Dev</department>   
<department>Service</department>   
<salary>60000</salary>   
<joiningdttm>2011-04-13T19:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP008</empId>   
<name>Gray</name>   
<dob>1995-07-16</dob>   
<designation>Backend Developer</designation>   
<department>Dev</department>   
<department>Service</department>   
<salary>10000</salary>   
<joiningdttm>2014-02-17T11:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP009</empId>   
<name>Robert</name>   
<dob>2001-03-31</dob>   
<designation>UI Developer</designation>   
<department>Dev</department>   
<department>Service</department>   
<salary>30000</salary>   
<joiningdttm>2021-08-16T13:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP010</empId>   
<name>Robert1</name>   
<dob>2002-03-31</dob>   
<designation>UI Developer</designation>   
<department>Dev</department>   
<department>Service</department>   
<salary></salary>   
<joiningdttm>2018-04-09T07:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP0011</empId>   
<name>Jack</name>   
<dob>1997-04-01</dob>   
<designation>GCP Admin</designation>   
<department>Ops</department>   
<department>Service</department>   
<salary>60000</salary>   
<joiningdttm>2011-07-21T10:12:13:232Z</joiningdttm>   
</employee>

<employee>   
<empId>EMP0012</empId>   
<name>May</name>   
<dob>1992-07-10</dob>   
<designation>GCP Admin</designation>   
<department>Ops</department>   
<department>RnD</department>   
<salary>40000</salary>   
<joiningdttm>2011-07-21T12:13:17:232Z</joiningdttm>   
</employee>

</employees>

for $employee at $counter in $doc//employee

return xdmp:document-insert("/" || $employee/empId/text(),

$employee,

<options xmlns="xdmp:document-insert">

<collections>

<collection>employee</collection>)

</collections>

</options>)

1. **Get the employee id grouping By Salary into following format**

**Query:**

xquery version "1.0-ml";

element result{

for $salary in distinct-values(collection("employee")/employee/salary)

return element object{element salary{$salary},collection("employee")[.//salary eq $salary]//empId}}

1. **Get Employee Id , Name , Salary, and date of joining is greater than 2010-07-21**

**Query:**

xquery version "1.0-ml";

for $employee in collection("employee")/employee

where $employee/joiningdttm ge "2020-07-21"

return element employee{$employee/empId,$employee/name,$employee/salary,$employee/joiningdttm}

1. **Get Employee whose Salary is Greater than 15000 in below format with tax 10%**

**Query:**

xquery version "1.0-ml";

element result{

for $employee in collection("employee")/employee

where $employee/salary gt "15000"

return element object{$employee/empId,$employee/name,$employee/salary,element tax{$employee/salary div 100 \* 10}}}}</result>

1. **Get all employees who are in Ops department but not in Service Department**

**Query:**

xquery version "1.0-ml";

for $employee in collection("employee")/employee

where $employee/department[1] eq "Ops" and $employee/department[2] ne "Service"

return $employee

1. **Retrieve all the employees whose age is greater than 27 years**

**Query:**

xquery version "1.0-ml";

for $employee in collection("employee")/employee

let $age := xs:integer(fn:days-from-duration(fn:current-date() - xs:date($employee/dob/text())) div 365.25)

where $age gt xs:integer("27")

return $employee

1. **Retrieve all the employees who are born in November or December in below format**

**Query:**

xquery version "1.0-ml";

element result{

for $employee in collection("employee")/employee

let $date := xdmp:month-name-from-date(xs:date($employee/dob))

where $date eq "December" or $date eq "November"

return element object{$employee/empId,$employee/name,element bornMonth{$date}}

}

1. **Retrieve All employees from Ops Department[1] and group by Department[2]**

**Query:**

xquery version "1.0-ml";

element result{

for $dep in distinct-values(collection("employee")/employee/department[2])

return element object {element department{$dep}, collection("employee")[.//department[1] eq "Ops" and .//department[2] eq $dep ]}}

1. **Write a Function which return Interval For given input start datetime, end datetime and interval duration**

**Query:**

xquery version "1.0-ml";

declare function local:interval($start as xs:dateTime,$end as xs:dateTime,$interval as xs:dayTimeDuration,$att as xs:integer)

{

if($start ne $end) then

let $startDate := $start + $interval

return

(

<interval attr="{$att}">

<startTime>{$start}</startTime>

<endTime>{$startDate - xs:dayTimeDuration("PT1M")}</endTime>

</interval>,

local:interval($startDate,$end,$interval,$att+1))

else

(<interval attr="{$att}">

<startTime>{$start}</startTime>

<endTime>{$end}</endTime>

</interval>)

};

<IntervalDetails>{

let $startDate := xs:dateTime("2008-02-01T10:00:00")

let $endDate := xs:dateTime("2008-02-02T10:00:00")

return (local:interval($startDate,$endDate,xs:dayTimeDuration("PT1H"),1))

}</IntervalDetails>