**1. Create Document with URI “/emails” and insert the above document.**

**Query:**

xquery version "1.0-ml";

xdmp:document-insert(

"/emails.xml",

<emails>

<email>

<to>Vishal</to>

<from>Sonu</from>

<heading>Hello</heading>

<body>Hello brother, how are you!</body>

</email>

<email>

<from>Jack</from>

<heading>Birth day wish</heading>

<body>Happy birth day Tom!</body>

</email>

<email>

<to>James</to>

<from>Jacalin</from>

<heading>Morning walk</heading>

<body>Please start morning walk to stay fit!</body>

</email>

<email>

<to>Kartik</to>

<from>Kumar</from>

<heading>Health Tips</heading>

<body>Smoking is injurious to health!</body>

</email>

<email>

<to>Sonu</to>

<from>Vishal</from

<heading>Hello</heading>

<body>Hello brother I’m Good, how are you?</body>

</email>

<email>

<to>James</to>

<from>Sonu</from>

<heading>Birth day wish</heading>

<body>Happy birth day James!</body>

</email>

</emails>

)

--------------------------------------------------------------------------------------------------------------------------------------

**2. For Each Email in Doc “/emails” create two Documents one for one for user who sent the email and one for who received the email.**

**Add to email with URI {/Outbox/from/ heading }, to the collections “email” and “outbox” and**

**add from email with URI{/Inbox/to/ heading}, to the collection’s “inbox” and “email”.**

**Query:**

for $emails at $counter in fn:doc("/emails.xml")//email

return xdmp:document-insert(fn:string-join((fn:distinct-values(fn:string-join(("/Outbox/","/"),xs:string($emails//to))),""),

fn:replace($emails//heading/string()," ","\_")),fn:doc("/emails.xml")//email[$counter],

(xdmp:default-permissions()),("Email","Outbox")),

for $emails at $counter in fn:doc("/emails.xml")//email

return xdmp:document-insert(fn:string-join(( fn:distinct-values(fn:string-join(("/Inbox/","/"),xs:string($emails//from))),""),

fn:replace($emails//heading/string()," ","\_")),fn:doc("/emails.xml")//email[$counter],

(xdmp:default-permissions()),("Email","Inbox"))

--------------------------------------------------------------------------------------------------------------------------------------

**3. for each User retrieve the sent email and received email**

**Query:**

xquery version "1.0-ml";

element result{

let $inbox := collection("Email")//email

for $user in fn:distinct-values(($inbox//to/string() ,$inbox//from/string()))

return element user{element username{$user} ,

element emails {element inbox{collection("Inbox")//email[//to/string() = $user ]},

element outbox{collection("Outbox")//email[//from/string()= $user]}}}}

-----------------------------------------------------------------------------------------------------------------------------------

**4. retrieve all the emails and group by from user**

**Query:**

xquery version "1.0-ml";

let $inbox := fn:distinct-values(collection("Email")//from)

for $s in $inbox

return element result{element user{element username{$s} ,

element emails {collection("Email")//email[from/string()= $s]}}}

--------------------------------------------------------------------------------------------------------------------------------------

**5. retrieve all the emails and group by heading**

**Query:**

xquery version "1.0-ml";

let $inbox := fn:distinct-values(fn:doc("/emails.xml")//heading)

for $s in $inbox

return element result{element heading{$s} ,

element emails {fn:doc("/emails.xml")//emails/email[heading/string()= $s]}}

**6. Retrieve all the emails related to James and sonu.**

**Query:**

xquery version "1.0-ml";

for $email in collection("Email")

return

if($email//email[to = "Sonu" or from = "Sonu"] or $email//email[to = "James" or from = "james"])

then $email

else()

**7. Return the username who received two or lesser than two emails.**

**Query:**

xquery version "1.0-ml";

for $name in fn:distinct-values(collection("Inbox")//from)

let $count := fn:count(collection("Inbox")//from[//from/string() =$name])

where $count<=2

return element name {$name}

--------------------------------------------------------------------------------------------------------------------------------------

**8. Retrieve all birthday wish Emails.**

**Query:**

xquery version "1.0-ml";

for $wish in collection("Inbox")

let $d := fn:lower-case($wish//heading/string())

return

if(fn:matches($d,fn:lower-case("Birth Day wish")))

then $wish

else()

-------------------------------------------------------------------------------------------------------------------------------------

**9. Write a Function which return Interval For given input start datetime, end datetime and interval duration(Day, Hours)**

**Query:**

xquery version "1.0-ml";

declare function local:displayInterval($startDateTime ,$endDateTime,$interval, $attr)

{

if($startDateTime le $endDateTime)

then(

element IntervalDetails{ element interval{attribute attr{$attr}, element Start{$startDateTime}, element End{$startDateTime+$interval}}},

local:displayInterval($startDateTime+$interval+$interval,$endDateTime,$interval,$attr+1))

else ()

};

local:displayInterval(xs:dateTime("2022-08-24T04:40:00"),xs:dateTime("2022-08-28T16:20:00"),xs:dayTimeDuration("P1DT1H20M"), 1)