* ClinicMain.xq

|  |  |
| --- | --- |
| **Function Summary** | |
| Treatments | **getPatientTreatments(String id, ClinicBD)**             For a given patient, returns the treatments |
| Drugs | **getPatientDrugs** (**String id, ClinicBD**)            For a given patient, returns the DrugsInfo |
| DB | **getTreatmentInfo** (**ClinicBD**)            Returns treatmentInfo in the DB |
| DB | **getProviderInfo** (**ClinicBD**)            Return ProviderInfo, for each provider, return patients that the provider is currently treating. |
| DB | **getDrugInfo**(**ClinicBD**)  For each Drugs, return the patients undergoing treatment with that drug, and the provider providing this drug. |

Test:

1. **getPatientTreatments**

Input:

let $clinic :=doc("ClinicSample.xml")/c:Clinic

return local:getPatientTreatments('10002',$clinic)

Output: see ResultOfgetPatientTreatments.xml

1. **getPatientDrugs**

Input:

let $clinic :=doc("ClinicSample.xml")/c:Clinic

return local:getPatientDrugs('10002',$clinic)

Output: see ResultOfgetPatientDrugs.xml

1. **getTreatmentInfo**

Input:

let $clinic :=doc("ClinicSample.xml")/c:Clinic

return local:getTreatmentInfo($clinic)

Output: see ResultOfgetTreatmentInfo.xml

1. **getProviderInfo**

Input:

**let** $clinic :=doc("ClinicSample.xml")/c:Clinic

**return** local:getProviderInfo($clinic)

Output: see ResultOfgetProviderInfo.xml

1. **getDrugInfo**

This task is actually to do an outer join.

Patient <-> Drug <-> provider

Input:

**let** $clinic :=doc("ClinicSample.xml")/c:Clinic

**return** g:getDrugInfo($clinic)

Output: see ResultOfgetDrugInfo.xml

# Problems encountered:

## Ugly namespace in the result

<?xml version="1.0" encoding="UTF-8"?>

<ps:treatmentInfo xmlns:ps="http://www.example.org/xquery/clinic"><ps:treatment><pat:patient-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">10001</pat:patient-id><trmt:provider-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">201</trmt:provider-id><trmt:diagnosis xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Cancer</trmt:diagnosis><trmt:radiology xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<trmt:date>2001-09-26</trmt:date>

<trmt:date>2001-10-26</trmt:date>

<trmt:date>2001-11-26</trmt:date>

<trmt:date>2001-12-26</trmt:date>

</trmt:radiology></ps:treatment><ps:treatment><pat:patient-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">10001</pat:patient-id><trmt:provider-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">202</trmt:provider-id><trmt:diagnosis xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Cancer</trmt:diagnosis><trmt:surgery xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<trmt:date>2001-12-28</trmt:date>

</trmt:surgery></ps:treatment><ps:treatment><pat:patient-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">10002</pat:patient-id><trmt:provider-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">201</trmt:provider-id><trmt:diagnosis xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">HIV</trmt:diagnosis><trmt:drug-treatment xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<trmt:name>B</trmt:name>

<trmt:dosage>1.5</trmt:dosage>

</trmt:drug-treatment></ps:treatment></ps:treatmentInfo>

Then I ask a question:

Just to discuss and get a better understanding:

I found that zorba will add namespace automatically into the result. However, when I use for loop, the namespaces were added repeatedly, which is ugly.

EX.

<?xml version="1.0" encoding="UTF-8"?>

<treatmentInfo><treatment><pat:patient-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" xmlns:prov="http://www.example.org/schemas/clinic/Provider" xmlns:tns="http://www.example.org/schemas/clinic" xmlns:trmt="http://www.example.org/schemas/clinic/Treatment" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">10001</pat:patient-id><trmt:provider-id xmlns:cid="http://www.example.org/schemas/clinic/ClinicIds" xmlns:pat="http://www.example.org/schemas/clinic/Patient" ...

I found a solution on stackoverflow: To avoid those namespace, just rebuild tags and only extract data (using data() or text()) from source .xml DB.

http://stackoverflow.com/questions/5850504/how-can-i-fix-ugly-xquery-output

For situation like:

<trmt:provider-id>201</trmt:provider-id>

just rebuild it as:

<provider-id>  { $provider-id/text() }</provider-id>

However, a new problem emergeed: How to deal with the the type substitution since there are multiple possibilities for <treatment>?

It is hard to rebuild an appropriate tag for those different formats.

<pat:treatment>

 <trmt:provider-id>201</trmt:provider-id>

 <trmt:diagnosis>Cancer</trmt:diagnosis>

 <trmt:radiology>

 <trmt:date>2001-09-26</trmt:date>

 <trmt:date>2001-10-26</trmt:date>

 <trmt:date>2001-11-26</trmt:date>

 <trmt:date>2001-12-26</trmt:date>

 </trmt:radiology>

 </pat:treatment>

 <pat:treatment>

 <trmt:provider-id>202</trmt:provider-id>

 <trmt:diagnosis>Cancer</trmt:diagnosis>

 <trmt:surgery>

 <trmt:date>2001-12-28</trmt:date>

 </trmt:surgery>

 </pat:treatment>

 </pat:treatments>

Answer from professor:

You could use typeswitch in XQuery to define a result element tag that depends on the treatment element tag.

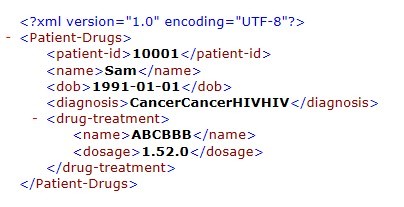
**Here are articles about typeswitch**

<http://blog.msbbc.co.uk/2010/06/marklogic-xquery-typeswitching-based-on.html>

[**http://www.xquery.com/tutorials/guided-tour/types-in-xquery.html**](http://www.xquery.com/tutorials/guided-tour/types-in-xquery.html)

Problem 2:

Get strange result when invoking local:getPatientDrugs('10001',$clinic)



**So, just make it clear, according to the schema, a patient could have multiple drug treatments.**

**Follow this rule to fix** local:getPatientDrugs()