

Article

[José Pereira](#) · Aug 28, 2020 5m read

[Open Exchange](#)

An example on finding inconsistencies in FHIR data

As I said in the [previous article](#), I started to learn about FHIR for the contest, and I'd like to share an update in my application: detection of inconsistencies in FHIR data.

Note: This document it's also available in [GitHub](#).

A useful usage for FHIR unified schema is search for inconsistencies. As suggested by [@Qi Li](#), one example could be find patients with records for diabetes medication, however, without diabetes condition.

Another example is search conditions with findings (like diabetes, for instance) and a record for resources denoting "no known problems", e.g., SNOMED CT: "160245001 |No current problems or disability (situation)|", as discussed in [HL7 FHIR documentation](#).

So, in order to deal with them, I setup two SQL KPIs for querying FHIR SQL schema and return resources which match to these kinds of inconsistencies. For input, I manually changed some files which are imported when the container is created.

For diabetes medication inconsistency, I choose a patient without diabetes conditions and insert manually a diabetes medication (file [Mikel238Dickinson68862644b68-f18b-46e0-86f9-56b3cb2f6737.json](#)). I found the medication code by searching for "snomed ct insulin".

```
"medicationCodeableConcept": {  
  "coding": [  
    {  
      "system": "http://snomed.info/sct",  
      "code": "325072002",  
      "display": "Insulin Aspart (substance)"  
    }  
  ],  
  "text": "Lantus 100 unit/ml injectable solution"
```

For "no known problems" inconsistencies, I changed the file [Adina377Corkery305cb12851a-2ebd-4c15-88a9-5bee0f308afc.json](#), and add a "160245001 |No current problems or disability|" record in an pre-existing "190905008 |Cystic Fibrosis|" condition.

```
"coding": [  
  {  
    "system": "http://snomed.info/sct",  
    "code": "160245001",  
    "display": "No current problems or disability"  
  },  
  {  
    "system": "http://snomed.info/sct",  
    "code": "190905008",  
    "display": "Cystic Fibrosis"  
  }  
],
```

Then, the two SQL shown below were setup. The first one gets diabetes medication inconsistencies; the second, "no known problems" inconsistencies:

```
-- patients with diabetes medication but no diabetes condition  
  
SELECT  
  
  (  
  
    SELECT GetProp(GetJSON(GetAtJSON(GetJSON(GetJSON(ResourceString, 'medicationCodeableConcept'), 'coding'),0), 'display'), 'display') FROM HSFHIRI0001R.Rsrc r where r.Key = m.Key  
  
  ) "Medication (ingredient)",  
  
  encounter "Encounter",  
  
  authoredon "Encounter date"  
  
FROM HSFHIRI0001S.MedicationRequest m  
  
WHERE  
  
  -- data inserted for last 7 days  
  
  lastUpdated BETWEEN DATEADD('dd', -7, CURRENT_TIMESTAMP) and CURRENT_DATE
```

-- diabetes medication

```
AND $LISTGET(m.code, 1) %inlist $LISTBUILD('96367001','420837001','421367005','420609005','325072002','427292001','414515005','411529005','411530000','412210000','426313003','422346007')
```

-- no diabetes condition

```
AND 0 = (
```

```
  SELECT count(*) FROM HSFHIRI0001S.Condition c
```

```
  WHERE
```

-- diabetes conditions

```
  $LISTGET(c.code, 1) %inlist $LISTBUILD('73211009','733072002','530558861000132104','609568004','609569007','105401000119101','199223000','703136005','46635009','44054006','111552007','716362006','123763000','722206009','8801005')
```

```
  AND m.patient = c.patient
```

```
)
```

-- conditions with "no known problems" no properly

```
SELECT
```

```
  ",
```

```
  encounter "Encounter",
```

```
  onsetDate "Encounter date"
```

```
FROM HSFHIRI0001S.Condition
```

```
WHERE
```

-- data inserted for last 7 days

```
  lastUpdated BETWEEN DATEADD('dd', -7, CURRENT_TIMESTAMP) and CURRENTDATE
```

-- 160245001 = no known problems

```
AND $FIND(code, 160245001) > 0
```

-- more than 1 condition code recorded (each code recorded generates 2 entries into code list - code and its description)

```
AND $LISTLENGTH(code) > 2
```

I found codes for diabetes conditions and medications in the CDC site, [here](#) and [here](#).

These SQL feed up the KPIs [DiabetesMedicationInconsistence](#) and [NoKnowProblemsInconsistence](#), respectively.

Finally, I setup a dashboard where this information is displayed. Note that two inconsistencies issue were detected - one for each of them.

The screenshot shows a web application interface for InterSystems IRIS. The top navigation bar includes 'Menu', 'Home', 'Sobre', 'Salvar', 'Logout', 'Usuário: _SYSTEM', 'Licenciado para: InterSystems IRIS Community', and 'InterSystems IRIS'. The main content area is titled 'User/Inconsistencies' and 'Last 7 days inconsistencies'. On the left, there is a 'Filtros' sidebar. The main area displays two summary cards: 'Diabetes medication inconsistencies' with a count of 1, and 'Not know problems inconsistencies' with a count of 1. Below these are two detailed views:

- Diabetes medication inconsistencies (details)**

	Encounter	Encounter date
Insulin Aspart (substance)	Encounter/2174	2015-10-28T22:46:55Z
- Not know problems inconsistencies (details)**

	Encounter	Encounter date
	Encounter/91	2019-05-03T11:57:24Z

[#Contest #FHIR #InterSystems IRIS for Health](#)
[Check the related application on InterSystems Open Exchange](#)

Source URL: <https://community.intersystems.com/post/example-finding-inconsistencies-fhir-data>