

ELLYPUNK

# Applied

WEEKLY DIGEST / T201

7-13 September 2026



# This week

7-13 September 2026

Available archived editions, in their original date order.

MONDAY / 7 September 2026

## FHIR Validation

3

[Open original online](#)

---

TUESDAY / 8 September 2026

## Qlik Associations: One Meaning per Field Name

8

[Open original online](#)

---



# FHIR VALIDATION

## Catch Bad Clinical Data Early



### TODAY'S OBJECTIVE

Check a FHIR resource before it crosses a trust boundary.

### FOUR LAYERS



Base rules



Agreed profile



Terminology



Business rules

### WHY NOW



Early feedback makes bad data easier to repair.

### IMPORTANT



Validation supports safety. It does not prove clinical correctness.



Workflow • Linux / macOS / Windows



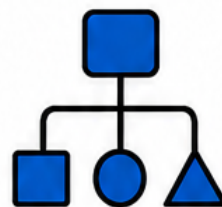


# VALIDATE IN FOUR PASSES

## 1

### STRUCTURE & TYPES

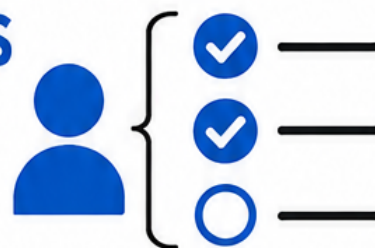
Known elements?  
Correct FHIR data types?



## 2

### CARDINALITY & INVARIANTS

Required items present?  
Min/max and co-occurrence  
rules satisfied?



## 3

### PROFILE & TERMINOLOGY

Right profile? Required local  
fields? Codes in allowed  
value sets?



## 4

### CONTEXTUAL RULES

References resolve? Duplicate  
risk checked? Sender authorised?



Business rules may need separate checks.

The base specification is  
the floor. Your implementation  
guide adds the route.





# MAKE FAILURE USEFUL BEFORE ACCEPTING

PAGE 3



Choose the *FHIR* release, profile, and terminology context.



Validate at chosen control points and record the result.

## READ THE *OperationOutcome*



**severity** — how serious the issue is



**code** — machine-friendly issue type



**expression** — affected element path



**diagnostics** — human-readable detail

## DECIDE BY POLICY



**Fatal** or **error**: reject or quarantine.



**Warning**: review by policy.



**Information**: keep when useful.

## SAFETY GUARDRAIL



Do not log sensitive resource bodies by default.

*Elly:*

“Keep the *reason*,  
not every *patient detail*.”





# VALIDATE A PATIENT UPDATE

## PROBLEM

The proposed Patient says:

```
resourceType: Patient
gender: "femalee"
```

- ! No identifier is present.
- ! The agreed profile requires one identifier.



## ACTION

FHIR API • Linux / macOS / Windows

```
POST [base]/Patient/$validate?profile=[profile URL]
```

Send the proposed Patient as the request body.



## CHECK



- HTTP 200 means the validation operation ran.
- An invalid resource can still return HTTP 200.
- Read the OperationOutcome. Find fatal or error issues and their expression.



## IMPROVEMENT



- Change "femalee" to "female".
- Add the required identifier.
- Validate again. Then run duplicate and authorisation checks.

Transport success is  
**not** validation success.





# SHIP THE **CONTROL**, NOT JUST THE TOOL



## PRACTICAL CHECKLIST

- ✓ Pin the FHIR release and profile version.
- ✓ Validate test fixtures in CI.
- ✓ Validate at chosen trust boundaries.
- ✓ Handle fatal, error, and warning separately.
- ✓ Keep OperationOutcome details useful and safe.
- ✓ Test terminology outages and unknown profiles.

## REMEMBER



Validation cannot prove clinical truth, narrative completeness, or every business rule.



## FINAL TIP FROM ELLY

“Ask what happens to bad data before you switch validation on.”





# QLIK ASSOCIATIONS

→ ONE MEANING PER FIELD NAME ←



## OBJECTIVE



Design a model where every shared field is an **intentional key**.

## OPENING RULE

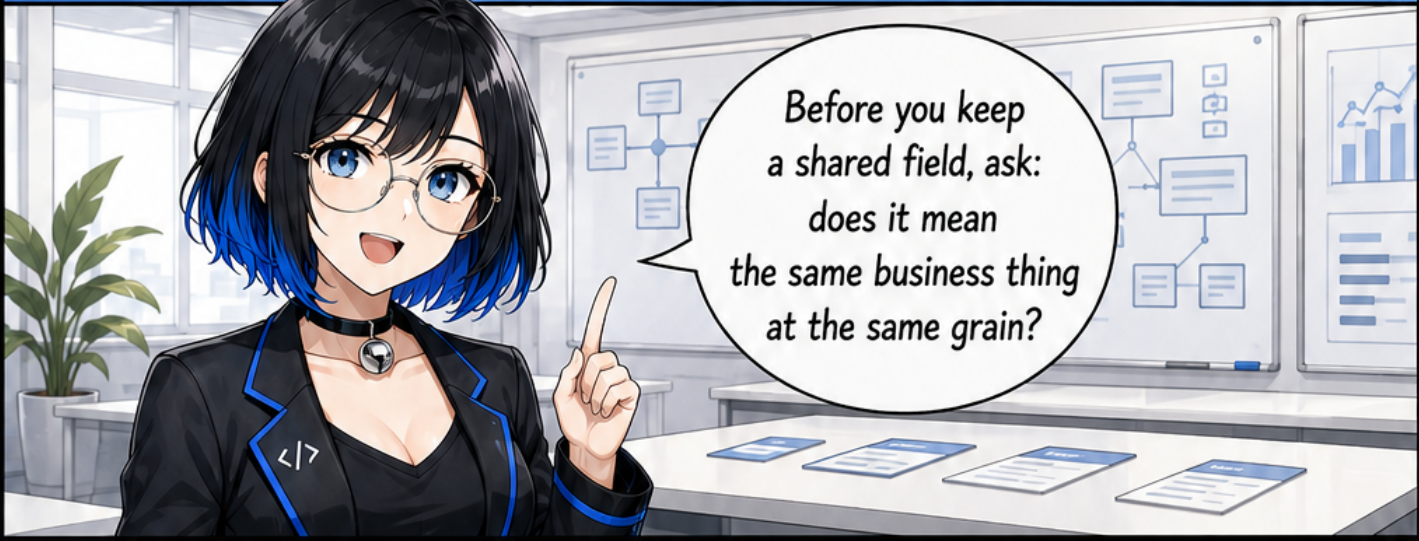
Same name = **same business meaning** and the **same grain**.





# HOW ASSOCIATIONS TURN INTO WARNINGS

PAGE 2



## 1 ONE SHARED FIELD

Qlik links the tables on that field name.



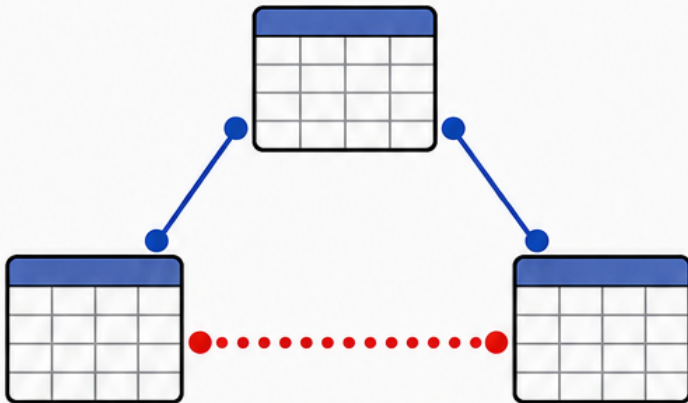
## 2 TWO OR MORE SHARED FIELDS

Qlik creates a synthetic key for the composite relationship.



## 3 MORE THAN ONE PATH

The model has a circular reference.





# FOUR RULES FOR A CLEAN MODEL



## 1 CHOOSE THE INTENDED KEY



Keep only fields that should drive selections between tables.

## 2 RENAME DIFFERENT ROLES

Date **AS** OrderDate  
Date **AS** ShipDate



Qlik Cloud load script —  
Linux / macOS / Windows

## 3 REMOVE REDUNDANT LINKS



Do not keep a second shared field merely because both tables contain it.

## 4 INSPECT AFTER EVERY RELOAD



Review unexpected \$Syn fields and dotted circular links in the internal table view.



## GUARDRAIL

*A synthetic key can be real.  
Unexpected, multiple, or dependent  
synthetic keys demand review.*





# WORKED EXAMPLE

## ONE LINK, NOT THREE

*Assumption: each shipment belongs to one OrderID.*

### PROBLEM

**Orders:** OrderID | CustomerID | Date

**Shipments:** OrderID | CustomerID | Date

Three shared names create a composite relationship.  
Date has two different meanings.

### ACTION

**Orders:** OrderID | CustomerID | *OrderDate*

**Shipments:** OrderID | *ShipDate*

- 1 Keep *OrderID* as the intended link.
- 2 Rename *Date* to *OrderDate* and *ShipDate*.
- 3 Remove or role-rename the redundant shipment *CustomerID*.

### CHECK

**Orders:** OrderID | CustomerID | *OrderDate*

**Shipments:** OrderID | *ShipDate*

- ✓ Reload: no unexpected synthetic key; one link on OrderID.
- ✓ Select *OrderID* = 1001: the matching order and shipment respond together.

### IMPROVEMENT



The model now states the relationship *deliberately*.



# THE ASSOCIATION CHECKLIST

PAGE 5

- 1  Same name means the same business meaning and grain. 
- 2  Count shared fields between every table pair. 
- 3  Rename role-specific dates, names, and IDs. 
- 4  Reload and inspect the internal table view. 
- 5  Test selections from both sides of every intended link. 



## RED FLAGS

Unexpected \$Syn, synthetic keys built from synthetic keys, or dotted circular links.



## FINAL TIP FROM ELLY

In Qlik, field names are **joins you can see** — make each one **deliberate**.

