DATA(BASE) DRIVEN DEVELOPMENT

pgconf 2018 by @niquola

НИКОЛАЙ РЫЖИКОВ

CTO of health-samurai.io @niquola

HEALTH IT > 10 YEARS

- EHR for hospitals
- Care coordination & telemedicine systems
- Government Registries
- BaaS based on FHIR

HEALTH IT

NOT HL || BD

but informational complex domain

COMPLEX DOMAIN?

- more then 300 entities
- more then 2K tables

COMPLEX DOMAIN?

Alphabetical

A-D:

- Account 2
- ActivityDefinition 2
- AllergyIntolerance 3
- AdverseEvent 0
- Appointment 3
- AppointmentResponse 3
- AuditEvent 3
- Basic 1
- Binary 5
- BodySite 1
- Bundle 5
- CapabilityStatement 3
- CarePlan 2
- CareTeam 2
- ChargeItem 0
- Claim 2
- ClaimResponse 2
- ClinicalImpression 0
- CodeSystem 5
- Communication 2
- CommunicationRequest 2
- CompartmentDefinition 1
- Composition 2
- ConceptMap 3
- Condition (aka Problem) 3
- Consent 1
- Contract 1
- Coverage 2
- DataElement 1
- DetectedIssue 1

D-I:

- Device 2
- DeviceComponent 1
- DeviceMetric 1
- DeviceRequest 0
- DeviceUseStatement 0
- DiagnosticReport 3
- DocumentManifest 2
- DocumentReference 3
- EligibilityRequest 2
- EligibilityResponse 2
- Encounter 2
- Endpoint 2
- EnrollmentRequest 0
- EnrollmentResponse 0
- EpisodeOfCare 2
- ExpansionProfile 2
- ExplanationOfBenefit 2
- FamilyMemberHistory 2
- Flag 1
- Goal 2
- GraphDefinition 0
- Group 1
- GuidanceResponse 2
- HealthcareService 2
- ImagingManifest 1
- ImagingStudy 3
- Immunization 3
- ImmunizationRecommendation 1
- ImplementationGuide 1

I-P:

- Library 2
- Linkage 0
- List 1
- Location 3
- Measure 2
- MeasureReport 2
- Media 1
- Medication 3
- MedicationAdministration 2
- MedicationDispense 2
- MedicationRequest 3
- MedicationStatement 3
- MessageDefinition 0
- MessageHeader 3
- NamingSystem 1
- NutritionOrder 2
- Observation 5
- OperationDefinition 4
- OperationOutcome 5
- Organization 3
- Parameters 5
- Patient 5
- PaymentNotice 2
- PaymentReconciliation 2
- Person 2
- PlanDefinition 2
- Practitioner 3
- PractitionerRole 2
- Procedure 3

P-Z:

- ProcedureRequest 3
- ProcessRequest 2
- ProcessResponse 2
- Provenance 3
- Questionnaire 3
- QuestionnaireResponse
- ReferralRequest 1
- RelatedPerson 2
- RequestGroup 2
- ResearchStudy 0
- ResearchSubject 0
- RiskAssessment 1
- Schedule 3
- SearchParameter 3
- Sequence 1
- ServiceDefinition 0
- Slot 3
- Specimen 2
- StructureDefinition 5
- StructureMap 2
- Subscription 3
- Substance 2
- SupplyDelivery 1
- SupplyRequest 1
- Task 2
- TestScript 2
- TestReport 0
- ValueSet 5
- VisionPrescription 1

COMPLEX DOMAIN?

Name	Flags	Card.	Type	Description & Constraints
MedicationAdministration	I		DomainResource	Administration of medication to a patient + *Reason not given is only permitted if NotGiven is true* + *Reason given is only permitted if NotGiven is false* Elements defined in Ancestors: id, meta, implicitRules, language, text, contained, extension, modifierExtension
identifier		0..*	Identifier	External identifier
definition	Σ	0..*	Reference(PlanDefinition | ActivityDefinition)	Instantiates protocol or definition
partOf	Σ	0..*	Reference(MedicationAdministration | Procedure)	Part of referenced event
status	?! Σ	1..1	code	in-progress | on-hold | completed | entered-in-error | stopped | unknown MedicationAdministrationStatus (Required)
category		0..1	CodeableConcept	Type of medication usage MedicationAdministrationCategory (Preferred)
medication[x]	Σ	1..1		What was administered SNOMED CT Medication Codes (Example)
medicationCodeableConcept			CodeableConcept	
medicationReference			Reference(Medication)	
subject	Σ	1..1	Reference(Patient | Group)	Who received medication
context		0..1	Reference(Encounter | EpisodeOfCare)	Encounter or Episode of Care administered as part of
supportingInformation		0..*	Reference(Any)	Additional information to support administration
effective[x]	Σ	1..1		Start and end time of administration
effectiveDateTime			dateTime	
effectivePeriod			Period	
performer	Σ	0..*	BackboneElement	Who administered substance
actor	Σ	1..1	Reference(Practitioner | Patient | RelatedPerson | Device)	Individual who was performing
onBehalfOf		0..1	Reference(Organization)	Organization organization was acting for
notGiven	?! Σ	0..1	boolean	True if medication not administered
reasonNotGiven	I	0..*	CodeableConcept	Reason administration not performed SNOMED CT Reason Medication Not Given Codes (Example)
reasonCode	I	0..*	CodeableConcept	Reason administration performed Reason Medication Given Codes (Example)
reasonReference		0..*	Reference(Condition | Observation)	Condition or Observation that supports why the medication was administered
prescription		0..1	Reference(MedicationRequest)	Request administration performed against
device		0..*	Reference(Device)	Device used to administer
note		0..*	Annotation	Information about the administration
dosage	I	0..1	BackboneElement	Details of how medication was taken + *SHALL have at least one of dosage.dose or dosage.rate[x]*
text		0..1	string	Free text dosage instructions e.g. SIG
site		0..1	CodeableConcept	Body site administered to SNOMED CT Anatomical Structure for Administration Site Codes (Example)
route		0..1	CodeableConcept	Path of substance into body SNOMED CT Route Codes (Example)
method		0..1	CodeableConcept	How drug was administered SNOMED CT Administration Method Codes (Example)
dose		0..1	SimpleQuantity	Amount of medication per dose
rate[x]		0..1		Dose quantity per unit of time
rateRatio			Ratio	
rateQuantity			SimpleQuantity	
eventHistory		0..*	Reference(Provenance)	A list of events of interest in the lifecycle

COMPLEX DOMAIN?

Name	Flags	Card.	Type	Description & Constraints
Identifier	Σ		Element	An identifier intended for computation Elements defined in Ancestors: id, extension
use	?! Σ	0..1	code	usual | official | temp | secondary (If known) IdentifierUse (Required)
type	Σ	0..1	CodeableConcept	Description of identifier Identifier Type Codes (Extensible)
system	Σ	0..1	uri	The namespace for the identifier value
value	Σ	0..1	string	The value that is unique
period	Σ	0..1	Period	Time period when id is/was valid for use
assigner	Σ	0..1	Reference(Organization)	Organization that issued id (may be just text)

Documentation for this format

DDD AGGREGATE

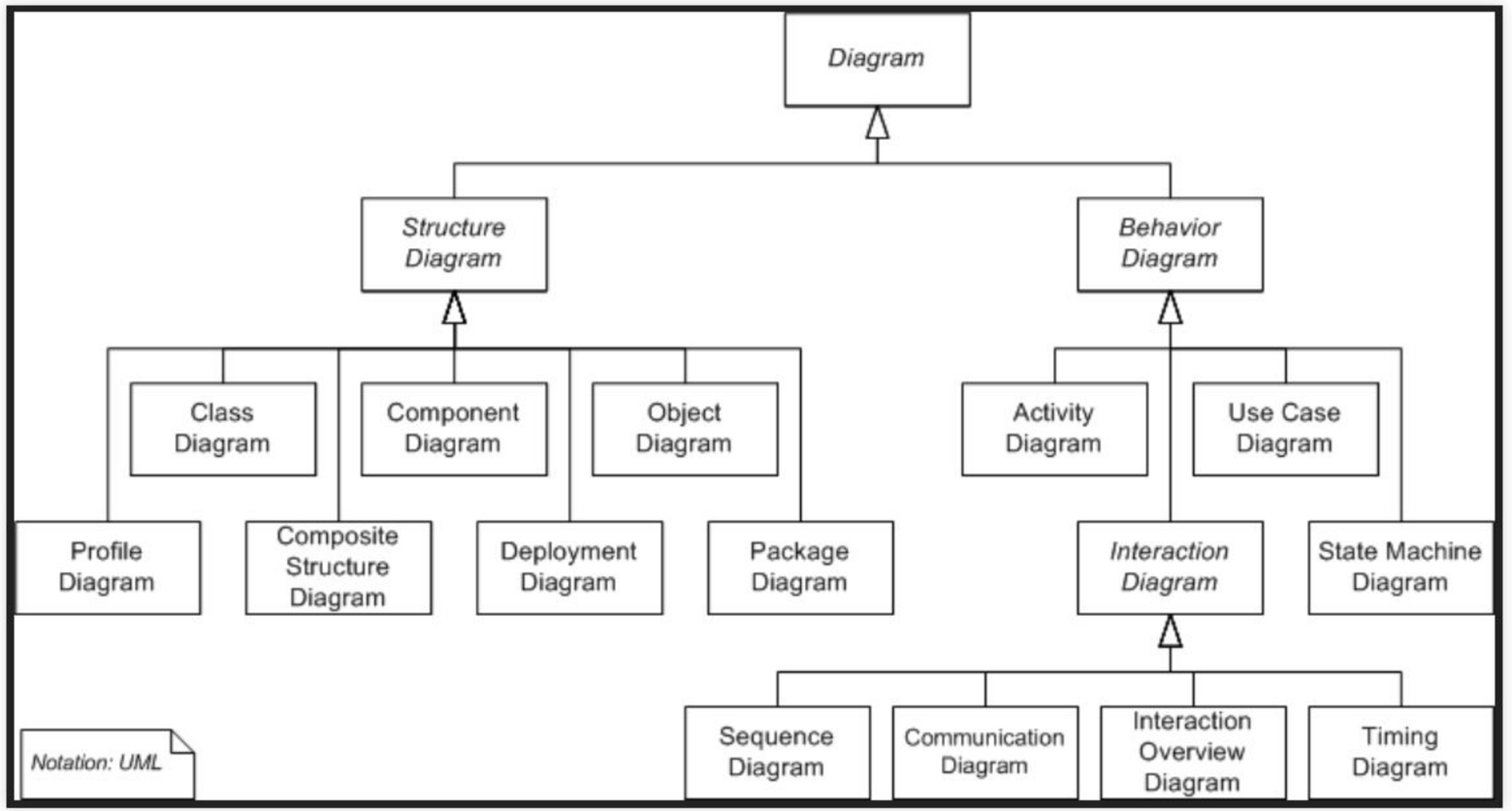
- cognitive load
- expensive
  joins

json(b) as solution

JSON(B) TECHNICAL PERSPECTIVE

- joins are expensive
- de-normalization on steroids

VARIABILITY / POLYMORPHIC



aka product catalog

VARIABILITY / POLYMORPHIC

- tables inheritance
- nullable columns
- jsonb !

OPEN SYSTEMS & EXTENSIBILITY

- integrations
- intensive vs extensive
- semantic web
- jsonb is a new EAV
- jsonb is open

WHAT DO WE PAY?

- 20% for access (vs column access)
- 1-3 times size (zson, jsonbc, compression)
- no statistic
- less concurrency
- more vacuum

Everything is trade-off

NO-SCHEMA

- constraint jsquery
- app layer validation (json-schema)
- attribute based schema (aka rdf/owl)

everything is trade-off do not let schema dominate semantic

OUR DB SCHEMA

```
create table patient (
  id text primary key,
  ts timestamp,
  resource jsonb
);
```

- json-schema generated from schema
- could be compiled into jsquery

HOW TO SEARCH?

```
SELECT  * FROM patient
 WHERE resource->>'gender' = 'male'
   AND resource#>>'{address,city}' = 'New York'
```

we pay x10% for this

SPEED IT UP!

```
CREATE EXTENSION pg_trgm;
CREATE INDEX patient_name_idx
ON patient
USING gin ((resource#>>'{name,given}') gin_trgm_ops,
           (resource#>>'{name,family}') gin_trgm_ops);
--
... WHERE resource#>>'{name,given}' ilike 'davi%' ...
```

CHANGE IT (||, -, #-)

```
update interboxmessage
set resource = resource ||
 jsonb_build_object(
    'status', 'pending',
    'details',
    jsonb_build_object(
      'encounter-date',
      hl7_date_to_iso(
        split_part(split_part(resource->>'message', 'FT1|', 2) , '|'
      )
    ) || resource->'details'
 )
where resource#>>'{type}' = 'HL7v2';
```

waiting for `resource['key']['key'] = val`

ADVANCED JSON MANIPULATION & PLV8

```
CREATE FUNCTION plv8_test(keys text[], obj jsonb)
RETURNS json AS $$
  var o = {};
  for(var i=0; i<keys.length; i++){
      //.....
  }
  return {result: ....}
$$ LANGUAGE plv8 IMMUTABLE STRICT;
```

MORE SOPHISTICATED SEARCH

```
{
  //...
  contacts:  [
    {system: "phone", value: "+178...."},
    {system: "email", value: "niq...."},
  ]
  //...
}
```

JSQUERY

```
WHERE
 resource @@ $JSQ$
    contacts.#. ( system = "phone" and value = "+14...") '
 $JSQ$
```

https://github.com/postgrespro/jsquery/

JSQUERY IDX

- optclasses
  - jsonb_path_value_ops
  - jsonb_value_path_ops
- hints 'x = 1 AND y /-- *index* / > 0'

one index - many queries

JSQUERY FOR SCHEMA

```
CREATE TABLE js (
    id serial,
    data jsonb,
    CHECK (data @@ '
        name IS STRING AND
        similar_ids.#: IS NUMERIC AND
        points.#:(x IS NUMERIC AND y IS NUMERIC)'::jsquery));
```

https://github.com/postgrespro/jsquery/

JSQUERY LIMITATIONS

- few json datatypes (string, bool, numeric)
- operators

JSQUERY FILTERING????

Feodor i did not find :(

JSON-KNIFE

```
SELECT knife_extract_max_timestamptz(
  $JSONB$
    {"a":{"b": [{"s": "?", "c": "1980-02-05"},
                {"s": "??", "c": "1979-02-05"}]}}
  $JSONB$,
  '[["a","b",{"s": "?"}, "c"]]'::jsonb
);
-------------------------------------
-- Tue Feb 05 23:59:59.999999 1980 PST
```

SQL/JSON STANDARD 2017

JSONB_AGG

```
jsonb_build_object(
  'query1', (SELECT json_agg(row_to_json(p.*)) FROM patient),
  'query2', .........
)
```

JSONB_OBJECT_AGG

```
with data(color_id, language, name) as (
values (1, 'de', 'blau'), (1, 'en', 'blue'), ...)
select color_id, jsonb_object_agg(language, name)
from data group by 1;

 color_id |              jsonb_object_agg
----------+-----------------------------------------
        1 | {"de": "blau", "en": "blue", "fr": "bleu"}
(1 row)
```

GRAPHSQL (GRAPHQL)

```
WITH parents as (....),
  children as (.......),
  children as (.......)
  SELECT json_agg/row_to_json/build_json_object
    FROM parents
```

BETTER JSON?

PGDN AKA EDN

```
{:name "John"
 :lastname "Black"
 :roles #{"user", "admin"}
 :last-visited #inst "2013-04-12T20:20:50.52Z"
 :id #uuid "b371b600-b175-11e2-9e96-0800200c9a66"}
```

FUNCTIONAL-RELATIONAL PROGRAMMING

OUT OF THE TAR PIT

MY APP

```
Request => Transform => DB => Query => Response
```

NOORM

- SQL is more declarative
- lack of composition

HONEYSQL

```
{:select [:f.* :b.baz :c.quux [:b.bla "bla-bla"]
          (sql/call :now) (sql/raw "@x := 10")]
 :modifiers [:distinct]
 :from [[:foo :f] [:baz :b]]
 :join [:draq [:= :f.b :draq.x]]
 :left-join [[:clod :c] [:= :f.a :c.d]]
 :right-join [:bock [:= :bock.z :c.e]]
 :where [:or
         [:and [:= :f.a "bort"] [:not= :b.baz (sql/param :param
         [:< 1 2 3]
         [:in :f.e [1 (sql/param :param2) 3]]
         [:between :f.e 10 20]]
 :group-by [:f.a]
 :having [:< 0 :f.e]
```

data notation for sql / composability

REACTIVE SYSTEMS / UI

LOGICAL REPLICATION

WAL2JSON => KAFKA

```
...
{
  "kind": "insert",
  "schema": "public",
  "table": "table_with_pk",
  "columnnames": ["a", "b", "c"],
  "columntypes": ["int4", "varchar", "timestamp"],
  "columnvalues": [1, "Backup and Restore", "2015-08-27 16:46:35.81
}
,{
  "kind": "insert",
  "schema": "public",
  "table": "table_with_pk",
  "columnnames": ["a", "b", "c"],
```

ASYNC/JDBC-LESS

JVM PG ADAPTER

- based on Netty
- 1K SLOC
- explicit prepared query
- only text