Introducing ISO-21090 Healthcare Datatypes

PostgreSQL Conference Europe 2011
October 18-21, Amsterdam

Yeb Havinga

MGRID Healthcare Datatypes

- Functionality common to many situations
- Increase software developer productivity
- Synergetic with PostgreSQL features
- Dramatic speedups


1
ml/h Σml
58.0
Alarm
Occlusion
Nearly Empty
Finish
Occlusion Limit
10 20 30 50
Select
Silence
Delivery
Power
Pause
Start
Purge
2
ml/h Σml
36.0
Alarm
Occlusion
Nearly Empty
Finish
Occlusion Limit
10 20 30 50
Select
Silence
Delivery
Pause
Start
Purge
SINO MDT
www.sinomdt.com
10
20
50ml
1
2

Calculate infusion rate

- Goal: administer 2 milliunits per minute
- Liquid: oxytocin 10 units per 1000 ml
- Enter on device: ml/hr

FOKKE & SUKKE

ZIJN BLIJ DAT DIE VERPLEEGSTERS NIET KUNNEN REKENEN

Using MGRID

```
t=# DO $$DECLARE
      administer pq := 2m[IU]/min';
      liquid     pq := pq '10 [IU]' / '1000 ml';
    BEGIN
      RAISE INFO '%', convert(administer/liquid, 'ml/h');
    END$$;

INFO:  11.999999999999999928 ml/h
DO
```

Catching errors

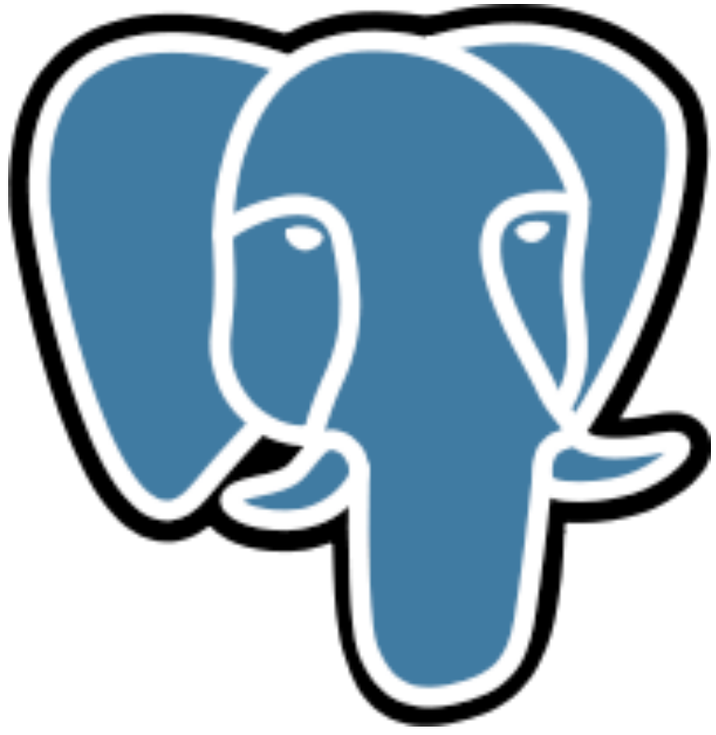
```
t=# DO $$DECLARE
      administer pq := '2m[IU]/min';
      liquid     pq := pq '10 [IU]' / '1000 ml';
    BEGIN
      RAISE INFO '%', convert(liquid/administer, 'ml/h');
    END$$;

ERROR:  cannot convert: unit mismatch between [IU]/ml/m[IU]/min and ml/h
CONTEXT:  PL/pgSQL function "inline_code_block" line 5 at RAISE
```

Healthcare Datatypes specification

- Originally part of HL7v3 standard
- 2008: became ISO 21090 standard

Healthcare Datatypes implementation

- Usually implemented using XML bindings
- MGRID implemented them with PostgreSQL's extensible type system



Healthcare Datatypes

ANY
BL
COLL
BAG
LIST
SET
CEQ
DSET
HXIT
HIST
ED

ST
SC
CD
CO
CS
OID
UUID
RUID
II
TEL
AD

EN
INT
REAL
RTO
PQ
MO
TS
EXPR
QSET
QSU
QSI

QSD
QSP
QSC
IVL
PIVL
EIVL
GTS
UVP
URG

Healthcare Datatypes

CD

IVL

PQ

TS

QSET

About codes

```
'EVN:2.16.840.1.113883.5.1001'::cv
```

```
'EVN'::cv('ActMood')
```

Codesystem

- Vocabulary
- Terminology
- Thesaurus
- Ontology
- Nomenclature
- Enumeration

Search | NCBO BioPortal

http://bioportal.bioontology.org/search

bioportal

BioPortal Browse Search Mappings Recommender Annotator Resource Index Projects

Term Search

Search for a term across multiple ontologies ?

twin

Select ontologies to search

Type here to select ontologies or leave blank to use all

select from list

Search

TERM NAME		MATCHED IN	
Fraternal twin brother	details | visualize	Preferred Nam	
Fraternal twin sister	details | visualize	Preferred Nam	
Identical twin brother	details | visualize	Preferred Nam	
Identical twin sister	details | visualize	Preferred Nam	
Twin brother	details | visualize	Preferred Nam	
Twin sister	details | visualize	Preferred Nam	
THT2_SCHPO	details | visualize	Property	
TLOV1_ARATH	details | visualize	Property	
Fetofetal Transfusion	details | visualize	Synonym	
Twin pregnancy	details | visualize	Preferred Nam	
Twin pregnancy, delivered	details | visualize	Preferred Nam	
Twin liveborn born in hospital	details | visualize	Synonym	
Twin pregnancy	details | visualize	Preferred Name	CORE Subset of SNOMED CT
dizygotic twin	details | visualize	Preferred Name	CRISP Thesaurus, 2006
monozygotic twin	details | visualize	Preferred Name	CRISP Thesaurus, 2006

Select Multiple Ontologies

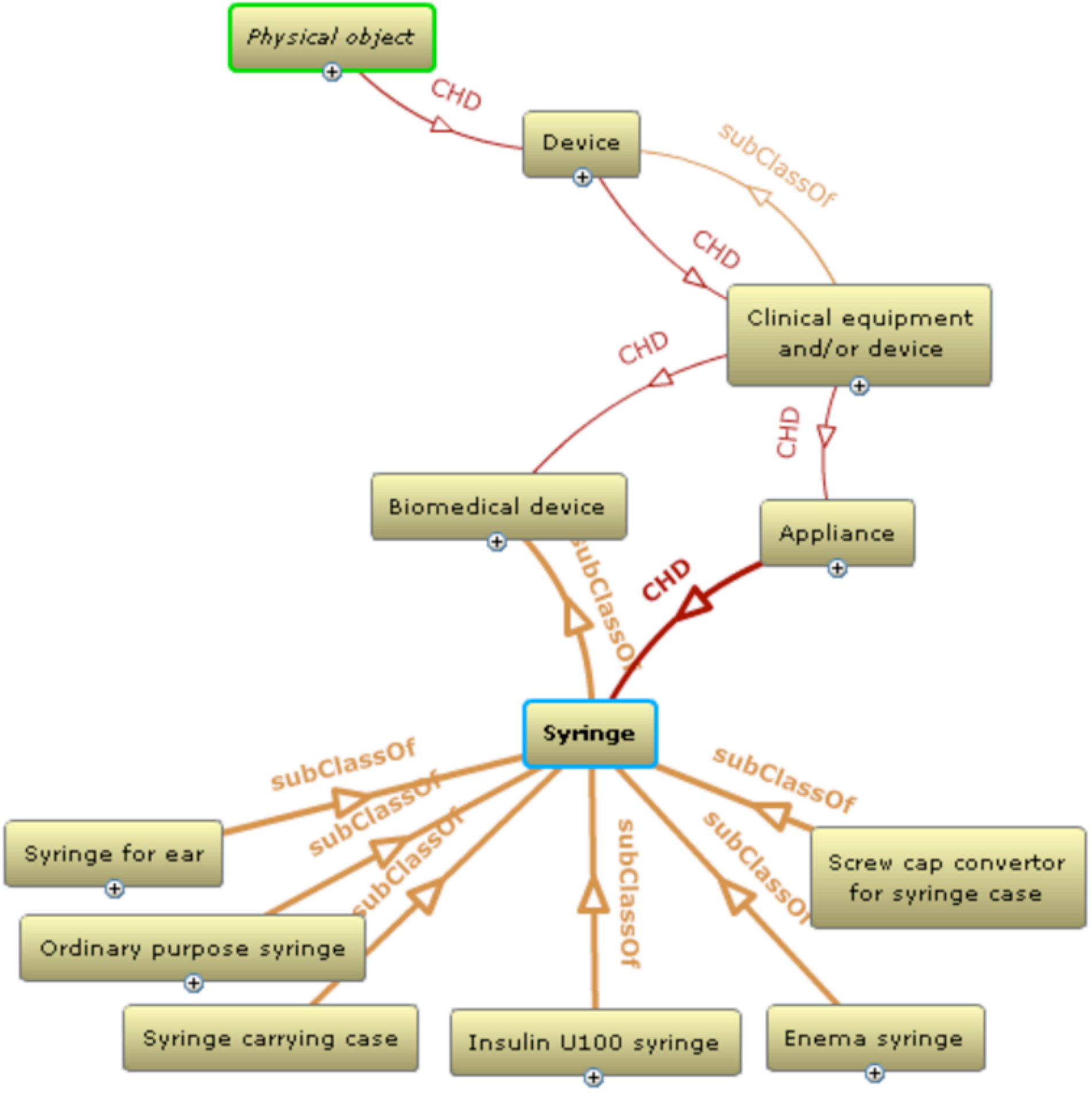
Search all ontology names

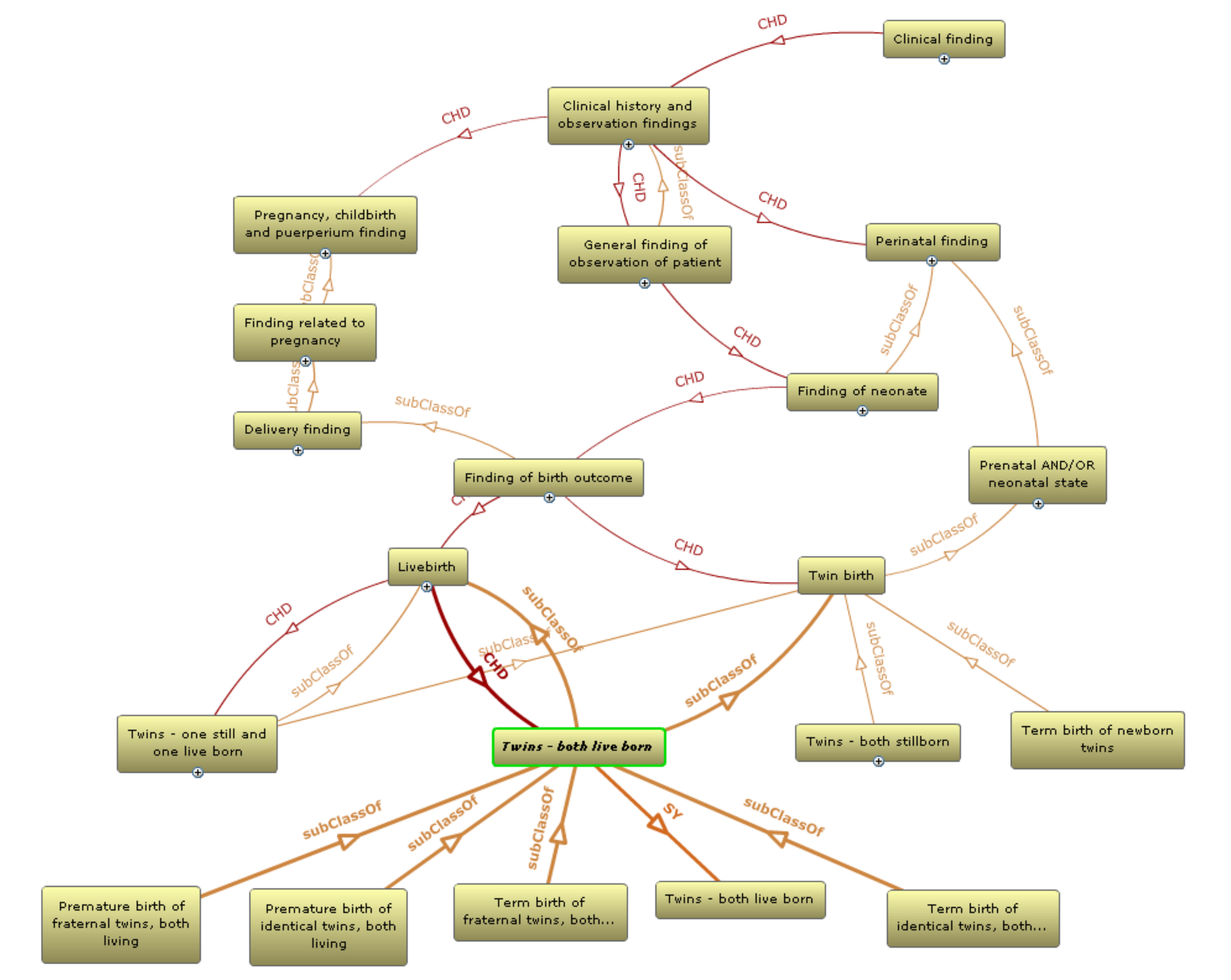
Groups Categories

- ABA Adult Mouse Brain (ABA)
- Adverse Event Ontology (AEO)
- Adverse Event Reporting ontology (AERO)
- African Traditional Medicine (ATMO)
- AI/RHEUM (AIR)
- Amino Acid (amino-acid)
- Amphibian gross anatomy (AAO)
- Amphibian taxonomy (ATO)
- Anatomical Entity Ontology (AEO)
- Animal natural history and life history (ADW)
- apollo-akesios (apollo)
- Ascomycete phenotype ontology (APO)
- Basic Formal Ontology (BFO)
- Basic Vertebrate Anatomy (basic-vertebrate-gross-anatomy)
- Bilateria anatomy (BILA)
- BioAssay Ontology (BAO)
- Bioinformatics data, formats, identifiers, operations and topics (EDAM)
- Biological imaging methods (FBbi)
- Biomedical Resource Ontology (BRO)
- BioPAX (BP)
- BioPortal Metadata (BPMetadata)
- BioTop (BT)
- BIRNLex (birnlex)
- Bleeding History Phenotype (BHO)
- Body System (bodysystem)
- Bone Dysplasia Ontology (BDO)
- Breast Cancer Grading Ontology (BCGO)
- Breast tissue cell lines (MCBCC)
- BRENDA tissue / enzyme source (BTO)
- Brucellosis Ontology (IDOBRU)
- C. elegans development (WBls)
- C. elegans gross anatomy (WBbt)

SNOMED-CT

- almost 400,000 concepts
- over 1M relations
- hierarchical


Physical object
CHD
Device
subClassOf
CHD
Clinical equipment and/or device
CHD
CHD
Biomedical device
subClassOf
CHD
Appliance
Syringe
subClassOf
subClassOf
subClassOf
subClassOf
subClassOf
subClassOf
Syringe for ear
Ordinary purpose syringe
Syringe carrying case
Insulin U100 syringe
Enema syringe
Screw cap convertor for syringe case


Clinical finding
CHD
Clinical history and observation findings
CHD
subClassOf
CHD
Pregnancy, childbirth and puerperium finding
General finding of observation of patient
Perinatal finding
subClassOf
Finding related to pregnancy
CHD
subClassOf
subClassOf
Delivery finding
subClassOf
CHD
Finding of neonate
Finding of birth outcome
Prenatal AND/OR neonatal state
CHD
CHD
Livebirth
Twin birth
subClassOf
CHD
subClassOf
subClassOf
subClassOf
subClassOf
subClassOf
Twins - one still and one live born
Twins - both live born
Twins - both stillborn
Term birth of newborn twins
subClassOf
subClassOf
subClassOf
Sy
subClassOf
Premature birth of fraternal twins, both living
Premature birth of identical twins, both living
Term birth of fraternal twins, both...
Twins - both live born
Term birth of identical twins, both...

Coded Value (CV)

- Can filter on codesystem subtrees
  - return all twin births (and subconcepts)
  - implies(code, '28030000|Twin birth')
  - code << '28030000|Twin birth'

Subtree query example

```
t=# WITH tenk AS
      (SELECT cdcode AS concept FROM pg_code
       WHERE cdcsid = 10000
       ORDER BY random()
       LIMIT 100000)
     SELECT concept::cv('SNOMED-CT') INTO t FROM tenk;
SELECT 100000
t=# CREATE INDEX ti ON t USING GIST(concept);
CREATE INDEX
```

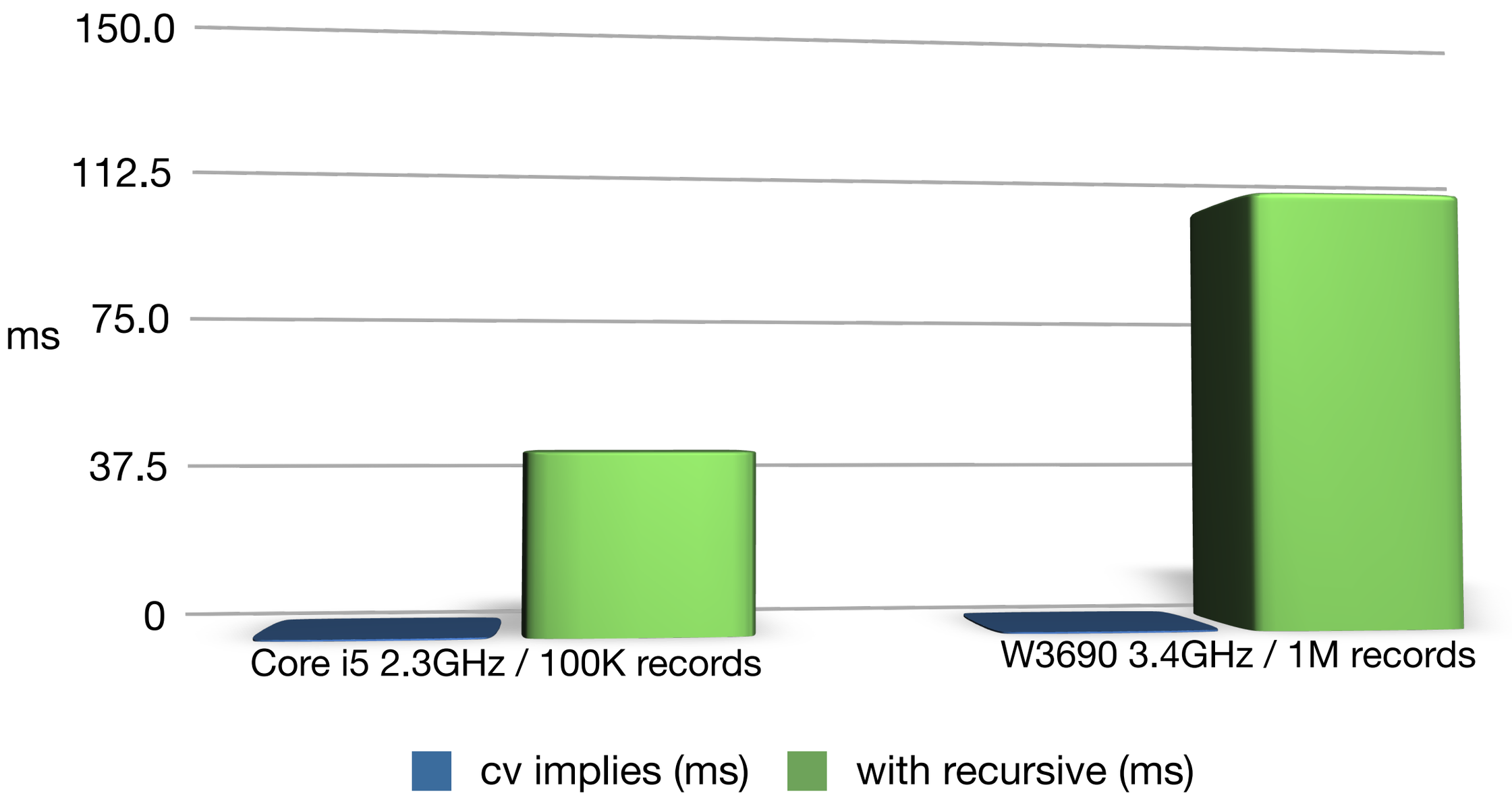
Implies query

```
t=# SELECT displayname(concept) FROM t
    WHERE concept << '28030000|Twin birth'::cv('SNOMED-CT');

                              displayname
-----------------------------------------------------------------
 Premature birth of identical twins, one living, one stillborn
 Term birth of identical twins, both stillborn
 Term birth of identical twins, both living
 Term birth of fraternal twins, both living
 Twin birth
 Term birth of stillborn twins
 Premature birth of identical twins, both stillborn
 Term birth of newborn twins
 Premature birth of fraternal twins, both stillborn
(9 rows)
```

Old style with recursive

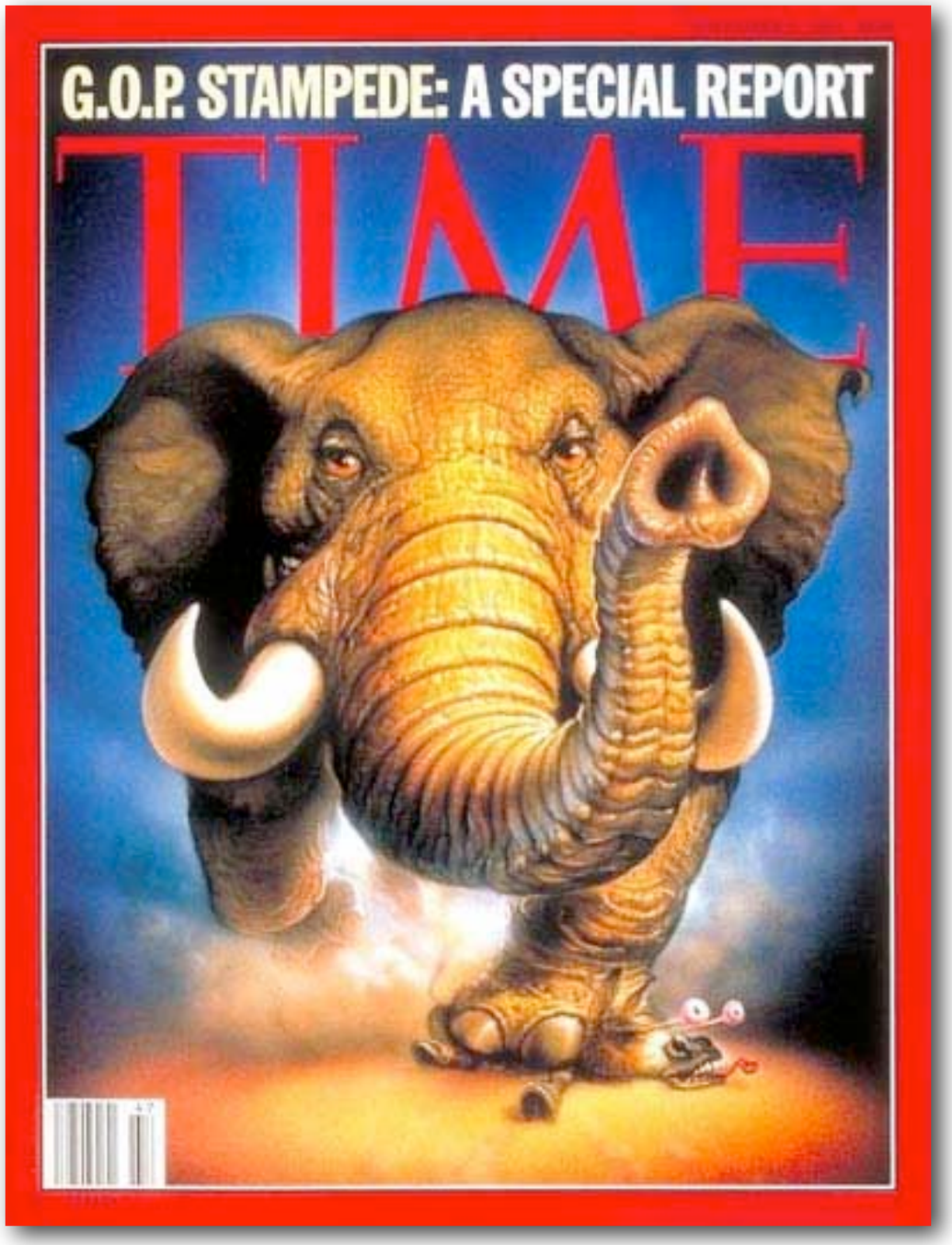
```
t=# WITH RECURSIVE r(n) AS
     (SELECT 28030000::bigint
      UNION ALL
      SELECT conceptid1
      FROM   relationships
      JOIN   r ON (conceptid2=n)
      WHERE  relationshiptype=116680003)
    SELECT displayname(concept)
    FROM   t
    WHERE  code(concept)::bigint IN (SELECT n FROM r);
                        displayname
-----------------------------------------------------------
 Premature birth of identical twins, one living, one stillborn
 Term birth of identical twins, both stillborn
 Term birth of identical twins, both living
 Term birth of fraternal twins, both living
 Twin birth
 Term birth of stillborn twins
 Premature birth of identical twins, both stillborn
 Term birth of newborn twins
 Premature birth of fraternal twins, both stillborn
(9 rows)
```

Implies vs with recursive



Benefits of CV

- Implies query shows 250X speedup compared to recursive subquery
- Increased productivity


G.O.P. STAMPEDE: A SPECIAL REPORT
TIME

Timestamp (TS)

- PostgreSQL has timestamp, do we need it?
- TS holds precision
- Conversion from and to Interval of Time (IVL_TS)

TS and IVL_TS

```
t=# SELECT '20080929'::ts;
     ts
----------
 20080929
(1 row)
```

```
t=# SELECT '20080929'::ts::ivl_ts;
       ivl_ts
---------------------
 [20080929;20080930[
(1 row)

t=# SELECT '200809291115'::ts::ivl_ts;
           ivl_ts
-----------------------------
 [200809291115;200809291116[
(1 row)
```

IVL_TS literal forms

Interval	`[20080929;20080930[`	29 sept 2008
Comparator	`<2009`	before 2009
Centerwidth	`200809291115 [300s]`	300s with 11:15 as center
Width	`[10d]`	10 days
Center	`20080929`	some interval with 20080929 as center
Any	`?20080929?`	some interval with 20080929 in it
Hull	`20080929..20080930`	29 and 30 sept 2008

IVL_TS Example

```
t=# CREATE TABLE t (a ivl_ts);
CREATE TABLE
t=# INSERT INTO t VALUES ('20080929'::ts), ('2007'::ts),
('<20111019');
INSERT 0 3
t=# SELECT * FROM t;
              a
-----------------------------
 [20080929;20080930[
 [2007;2008[
 ]NullFlavor.NINF;20111019[
(3 rows)
t=# SELECT * FROM t WHERE a && '<2008';
              a
-----------------------------
 [2007;2008[
 ]NullFlavor.NINF;20111019[
(2 rows)
```

Continuous set of time (QSET_TS)

QSET_TS is the closure of IVL_TS under +,-

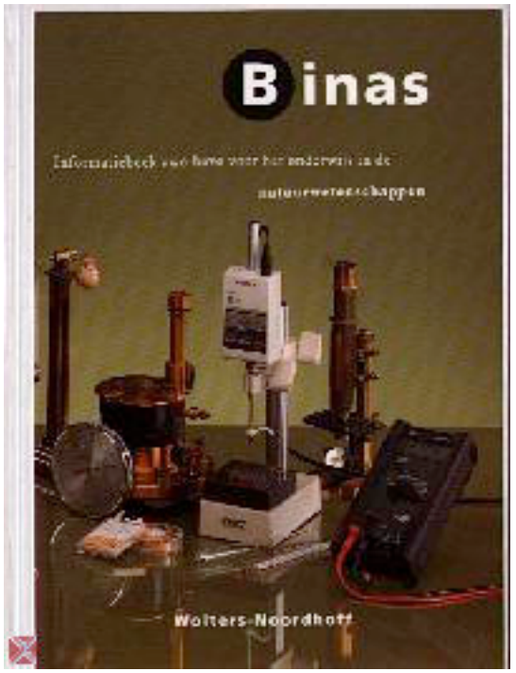
QSET_TS Example

```
t=# SELECT * FROM medication;
 name |    effectivetime
------+---------------------
 Pete | [20100316;20100514]
 Pete | [20100420;20100701]
 Pete | [20101220;20110119]
 John | [20100516;20100614]
 John | [20100620;20100801]
 John | [20101220;20110119]
(6 rows)

t=# SELECT name,
           canonical('2010'::ts::ivl_ts - SUM(effectivetime)) AS nomeds
FROM medication
GROUP BY name;
 name |                                nomeds
------+-----------------------------------------------------------
 John | [20100101;20100516[;]20100614;20100620[;]20100801;20101220[
 Pete | [20100101;20100316[;]20100701;20101220[
(2 rows)
```

TS, IVL_TS, QSET_TS

A complete solution for temporal data and queries

Physical quantities



Physical Quantity (PQ)

- Result of (calculations on) observations or measurements
- Value
- Unit

UCUM

- Unified Code for Units of Measure
- 294 units
- 24 prefixes
- unitsofmeasure.org

Voedingswaarde	per 100 ml	per glas*
Energie	193 kJ / 46 kcal	386 kJ / 92 kcal
Eiwit	3,5 g	7,0 g
Koolhydraten	4,6 g	9,2 g
waarvan suikers	4,6 g	9,2 g
Vet	1,5 g	3,0 g
waarvan: verzadigd vet	1,0 g	2,0 g
enkelvoudig onverzadigd vet	0,3 g	0,6 g
meervoudig onverzadigd vet	0,0 g	0,0 g
Voedingsvezel	0,0 g	0,0 g
Natrium	0,04 g	0,08 g
Calcium	128 mg (16%)[1]	256 mg (32%)[1]
Vitamine B12	0,58 µg (23%)[1]	1,16 µg (46%)[1]

* Een glas halfvolle melk is 200 ml.
** Op basis van 2000 kcal per dag.

Zie voor meer informatie: www.superunie.nl/merken
[1] Percentage o.b.v. de Aanbevolen Dagelijkse Hoeveelheid

Ten minste houdbaar tot: zie bovenzijde verpakking,
mits gekoeld bewaard (max. 7°C).

Astronomical Unit

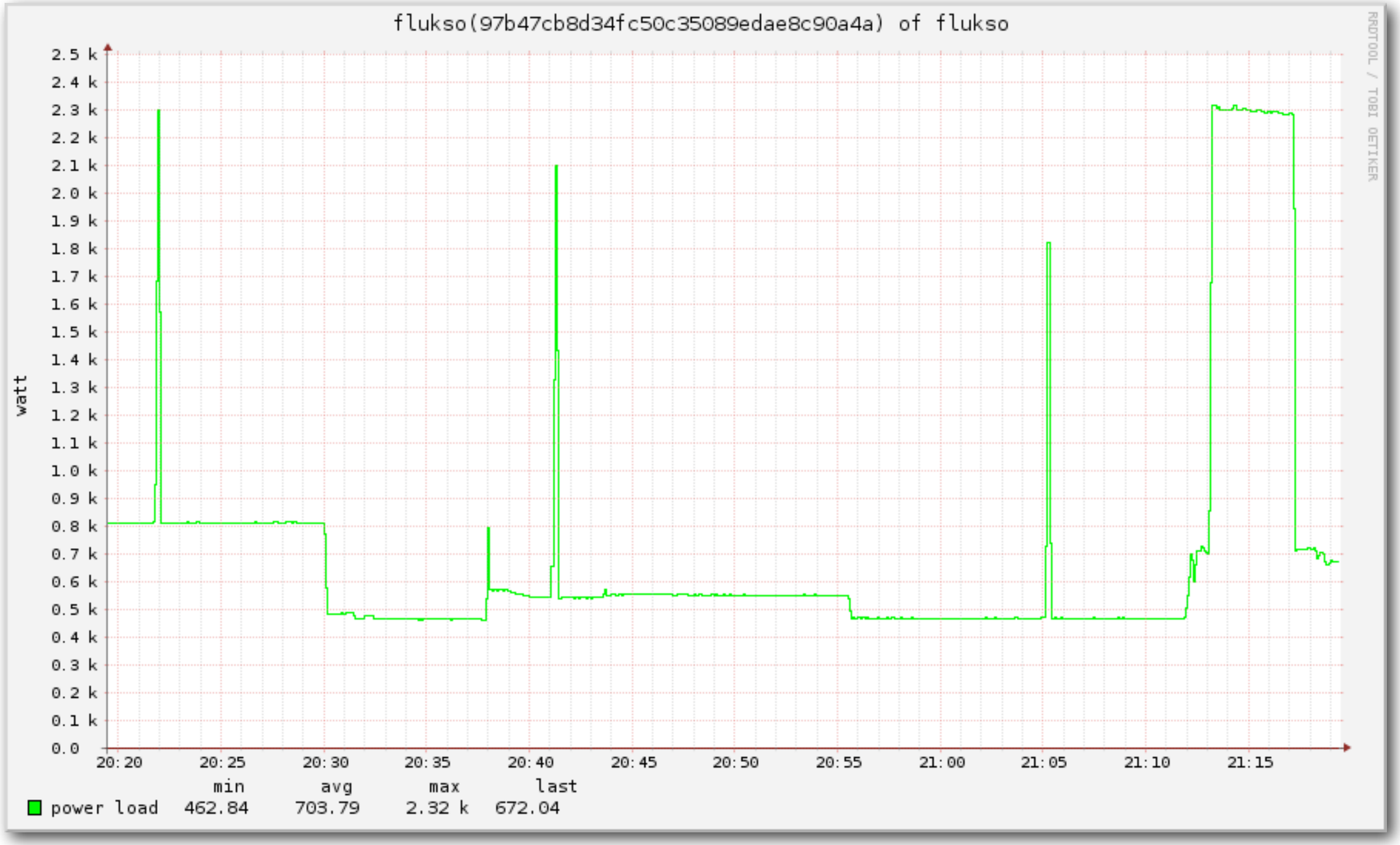
```
t=# SELECT convert(pq '1 AU' / '[c]', 'min');
        convert
------------------------
 8.3167463967689274 min
(1 row)
```

PQ

Physical quantities are all around us


Quooker


FLUKSO


flukso(97b47cb8d34fc50c35089edae8c90a4a) of flukso
RRDTOOL / TOBI OETIKER
watt
2.5 k
2.4 k
2.3 k
2.2 k
2.1 k
2.0 k
1.9 k
1.8 k
1.7 k
1.6 k
1.5 k
1.4 k
1.3 k
1.2 k
1.1 k
1.0 k
0.9 k
0.8 k
0.7 k
0.6 k
0.5 k
0.4 k
0.3 k
0.2 k
0.1 k
0.0
20:20
20:25
20:30
20:35
20:40
20:45
20:50
20:55
21:00
21:05
21:10
21:15
min avg max last
power load 462.84 703.79 2.32 k 672.04

```
DO $$DECLARE
-- 900 ml water was tapped from the Quooker. Assumed is that though
-- the water is heated to 108 Cel under pressure, once it is in the
-- teapot it is 100 Cel.
water_volume_tapped   pq := '900 ml';
water_start_temp      pq := convert('20 Cel', 'K');
water_end_temp        pq := convert('108 Cel', 'K');
-- http://en.wikipedia.org/wiki/Density#Density_of_water_.28at_1_atm.29
density_water_at_100c pq := '958.4 kg/m3';
water_mass            pq := water_volume_tapped * density_water_at_100c;
water_d_temp          pq := water_end_temp - water_start_temp;
-- The small calorie or gram calorie (symbol: cal) approximates
-- the energy needed to increase the temperature of 1 gram of water
-- by 1 C. This is about 4.2 joules.
-- http://en.wikipedia.org/wiki/Calorie
energy_predicted      pq := water_mass * '1 cal / g.K' * water_d_temp;
-- actual energy
energy_used           pq := pq '240 s' * '1600W';
BEGIN
      RAISE INFO 'predicted energy use %', convert(energy_predicted, 'J');
      RAISE INFO 'actual energy use %', convert(energy_used, 'J');
      RAISE INFO 'efficiency = %', convert(energy_predicted / energy_used,
'%');
END$$;

t=# \i q.sql
psql:/home/yeb/q.sql:25: INFO:  predicted energy use 317587.69152 J
psql:/home/yeb/q.sql:25: INFO:  actual energy use 384000 J
psql:/home/yeb/q.sql:25: INFO:  efficiency = 82.705128 %
DO
```

```
t=# CREATE DOMAIN pq_weight AS pq CHECK (compares(VALUE, 'kg'));
CREATE DOMAIN
t=# CREATE DOMAIN pq_height AS pq CHECK (compares(VALUE, 'm'));
CREATE DOMAIN
t=# CREATE TABLE patients (name text, height pq_height, bodyweight
pq_weight);
CREATE TABLE
t=# INSERT INTO patients VALUES
 ('philip',    '56 cm',                        '2900 g'),
 ('bob',       '1.89 m',                       '80 kg'),
 ('amy',    pq '5 [ft_i]' + '10 [in_i]', pq '10 [stone_av]' + '4 [lb_av]');
INSERT 0 3
t=# CREATE OR REPLACE FUNCTION bmi(pq_weight, pq_height)
    RETURNS pq
    AS $$
    SELECT convert($1 / $2^2, 'kg/m2');
    $$ LANGUAGE SQL;
CREATE FUNCTION
t=# SELECT name, bmi(bodyweight,height) AS bmi FROM patients;
  name   |           bmi
---------+--------------------------
 philip | 9.2474489795918367 kg/m2
 bob    | 22.395789591556787 kg/m2
 amy    | 20.661636626130395 kg/m2
(3 rows)
```

Finally

Summary

- Healthcare Datatypes extend PostgreSQL expressiveness
- Increases productivity
- Synergetic with PostgreSQL's set operations, triggers, constraints and pl/pgsql
- Products run faster

MGRID Availability '11

- Based on PostgreSQL core 9.x
- Linux 32 and 64 bits
- Microsoft Windows 32 bits
- Mac OS X Lion (in progress)

If you liked this talk, please rate it at

http://2011.pgconf.eu/feedback

backup slides

Code checking

```
t=# CREATE TABLE a (code cv('ActMood'));
CREATE TABLE
t=# INSERT INTO a VALUES ('wrong code');
ERROR:  code wrong code not found in any valueset bound to
conceptdomain ActMood
t=# INSERT INTO a VALUES ('EVN');
INSERT 0 1
t=# SELECT displayname(code), codesystem(code) FROM a;
    displayname     |         codesystem
--------------------+----------------------------
 event (occurrence) | 2.16.840.1.113883.5.1001
(1 row)
```

In an Act class instance where the Act.code attribute is a SNOMED CT expression, the expression SHOULD represent a type of [363787002 | observable entity], [<<129125009 | procedure with explicit context] or [<<272379006 | event]

```
t=# CREATE TABLE act (class cv('ActClass'), code cv);
CREATE TABLE
t=# ALTER TABLE act ADD CONSTRAINT act_code_snomedct CHECK (
    (codesystem(code) != '2.16.840.1.113883.6.96') OR
      (code << '363787002'::cv('SNOMED-CT') OR
       code << '129125009'::cv('SNOMED-CT') OR
       code << '272379006'::cv('SNOMED-CT')));
ALTER TABLE
t=# INSERT INTO act VALUES ('ACT','224166006'::cv('SNOMED-CT'));
ERROR:  new row for relation "act" violates check constraint "act_code_snomedct"
t=# INSERT INTO act VALUES ('ACT','3974003'::cv('SNOMED-CT'));
INSERT 0 1
t=# SELECT code(class), displayname(code) FROM act;
code |         displayname
------+-----------------------------
ACT  | Contact with sharp leaves
(1 row)
```

More benefits of CV

- Database can check code validity faster than application layer
- Software engineer completes job faster

Thanks

- SNOMED-CT graphs created with http://informatics.mayo.edu/adephont/index.php/CodingScheme:SNOMED_CT
- Power usage graph created with http://wpd.home.xs4all.nl/symon/
- Syringe pump example from http://www.southernhealth.org.au/icms_docs/1208_Induction_of_labour_and_augmentation_oxytocin_Syntocinoninfusion.pdf