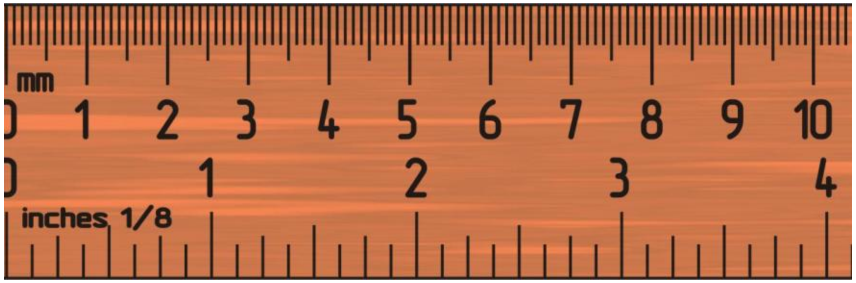
Observations, Concepts and Encounters

A programmer's guide

EHSDI
eBuzima

What's an obs?

- *Obs* is short for *observation*
- It can be measurement or a question, e.g.
  - What is a person's HIV status?
  - Are they pregnant?
  - What is their height?
  - What is their weight?



The flat table approach

- Many EMR's start as spreadsheets, using a flat table to hold observations of patients, e.g.

Patient	Height	Weight	Pregnant	HIV
Ben	160	70	No	No
Rita	150	60	No	No
Rowan	170	80	No	No
Christian	160	75	No	No

Observations

Patients

Flat table problems

- Table gets unmanageably big as we keep adding new observations
- Gets really complicated when we want multiple records of the same observation type, e.g. weight change over time
- Wasted space when columns aren't used

Adding time

- Adding time information to the observations means we can record change over time without creating new columns, e.g.

PATIENTS

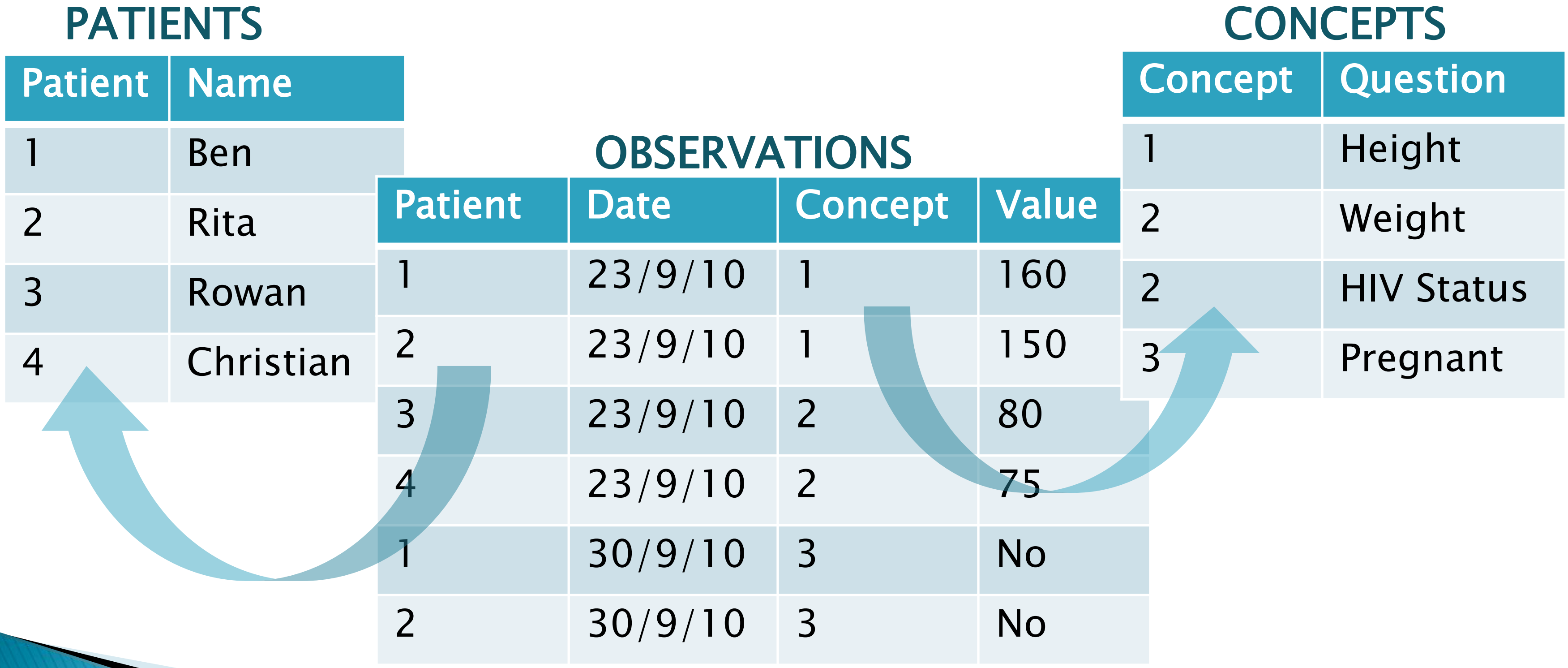
Patient	Name
1	Ben
2	Rita
3	Rowan
4	Christian

OBSERVATIONS

Patient	*Date*	Height	Weight	Pregnant	HIV
1	*23/9/10*	160	70	No	No
2	*23/9/10*	150	60	No	No
3	*23/9/10*	170	80	No	No
4	*23/9/10*	160	75	No	No
1	*30/9/10*	160	100	No	No
2	*30/9/10*	140	60	No	No

Adding concepts

- Each observation becomes a *concept* and a *value*… like a question and an answer



PATIENTS

Patient	Name
1	Ben
2	Rita
3	Rowan
4	Christian

OBSERVATIONS

Patient	Date	Concept	Value
1	23/9/10	1	160
2	23/9/10	1	150
3	23/9/10	2	80
4	23/9/10	2	75
1	30/9/10	3	No
2	30/9/10	3	No

CONCEPTS

Concept	Question
1	Height
2	Weight
2	HIV Status
3	Pregnant

Concept IDs

- If different EMRs are going to share data, then there needs to be a way of standardizing concepts
  - Is the concept called "HEIGHT" in one EMR the same as the concept called "HEIGHT" in another?
- We use an ID value which can be made standard across different EMRs…

Concept	ConceptId	Question
1	5090	Height
2	1001	Weight
2	2050	HIV Status
3	4000	Pregnant

Internal database id

External standardized id

Concepts in code

- OpenMRS defines a `Concept` class
- To get a specific concept use the `ConceptService`, e.g.

```
Concept height = Context.getConceptService().getConcept(5090);
```

Concept id for height

Concept management

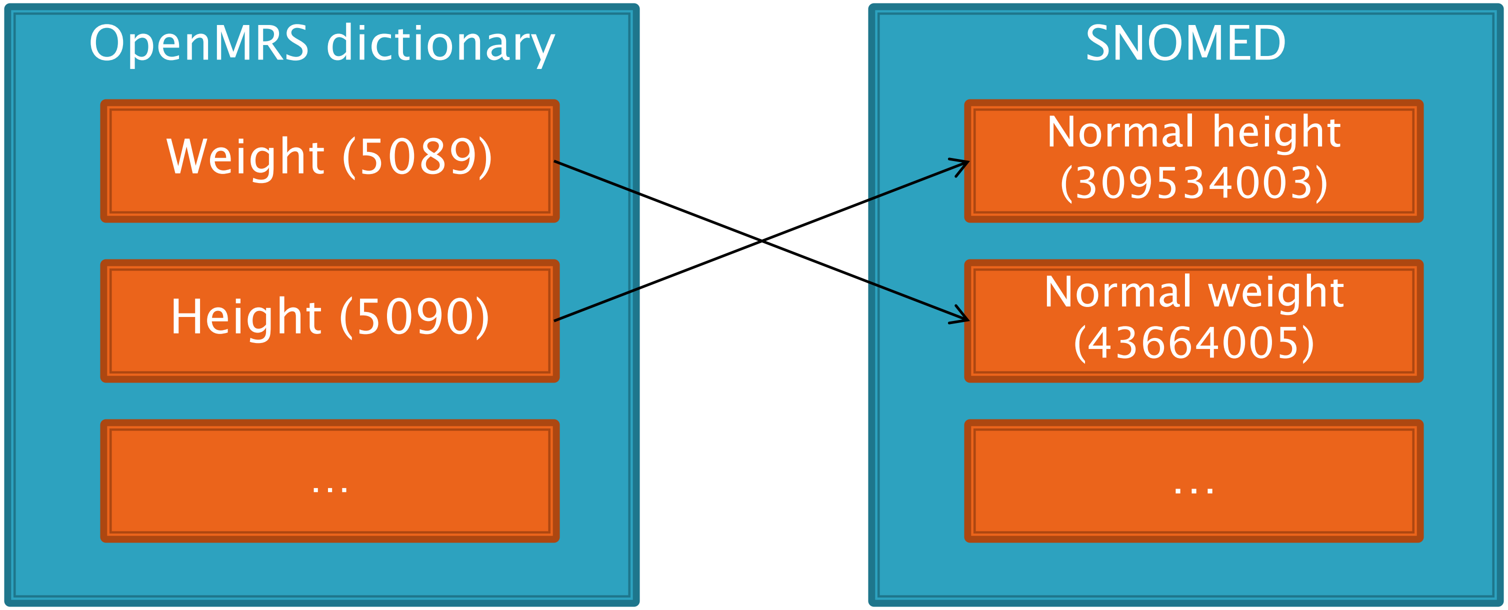
- It's important to keep concepts standardized across different OpenMRS installations
  - Means we can compare data from different installations
- Concepts should be managed, e.g.
  - Partners In Health maintains a concept dictionary in Boston to keep all their sites using the same concepts
  - If you need a new concept – ASK FIRST

Concept standards

- There are several existing standards for medical concepts, used by other EMR systems, e.g.
  - **SNOMED** (**S**ystematized **No**menclature of **Med**icine – – Clinical **T**erms). You can browse its concepts at http://terminology.vetmed.vt.edu/SCT/menu.cfm
  - **ICD-10** (International Classification of Diseases – Revision 10)

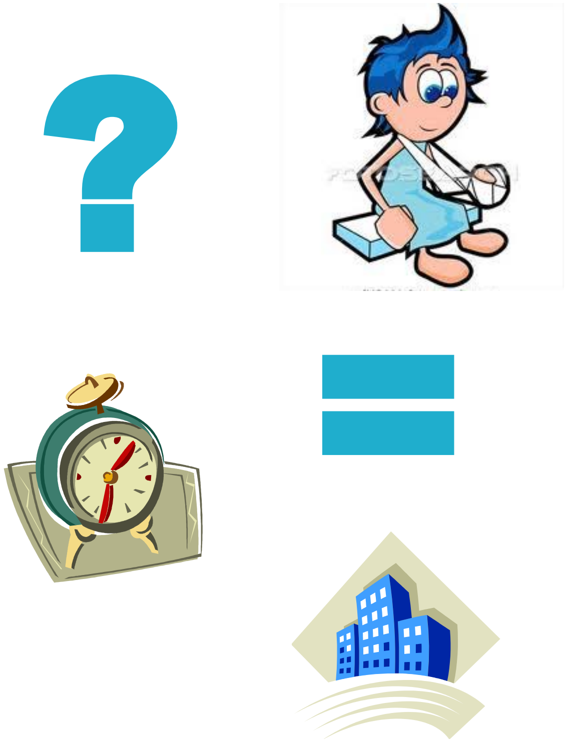
Concept mapping

- OpenMRS concepts can be mapped to concepts defined by standards such as SNOMED, e.g.



Creating observations

- For a single observation we want to store the following:
  - Patient
  - Concept
  - Value
  - Time/date
  - Location



Obs values

- Different concepts will require different types of answers, e.g.
  - "Height" and "weight" concepts will require a numeric value
  - "Pregnant" concept might require "Yes" or "No" answers (coded values)
  - "Comment" (i.e. from doctor) needs to be stored as text
- How can one field store all of these?

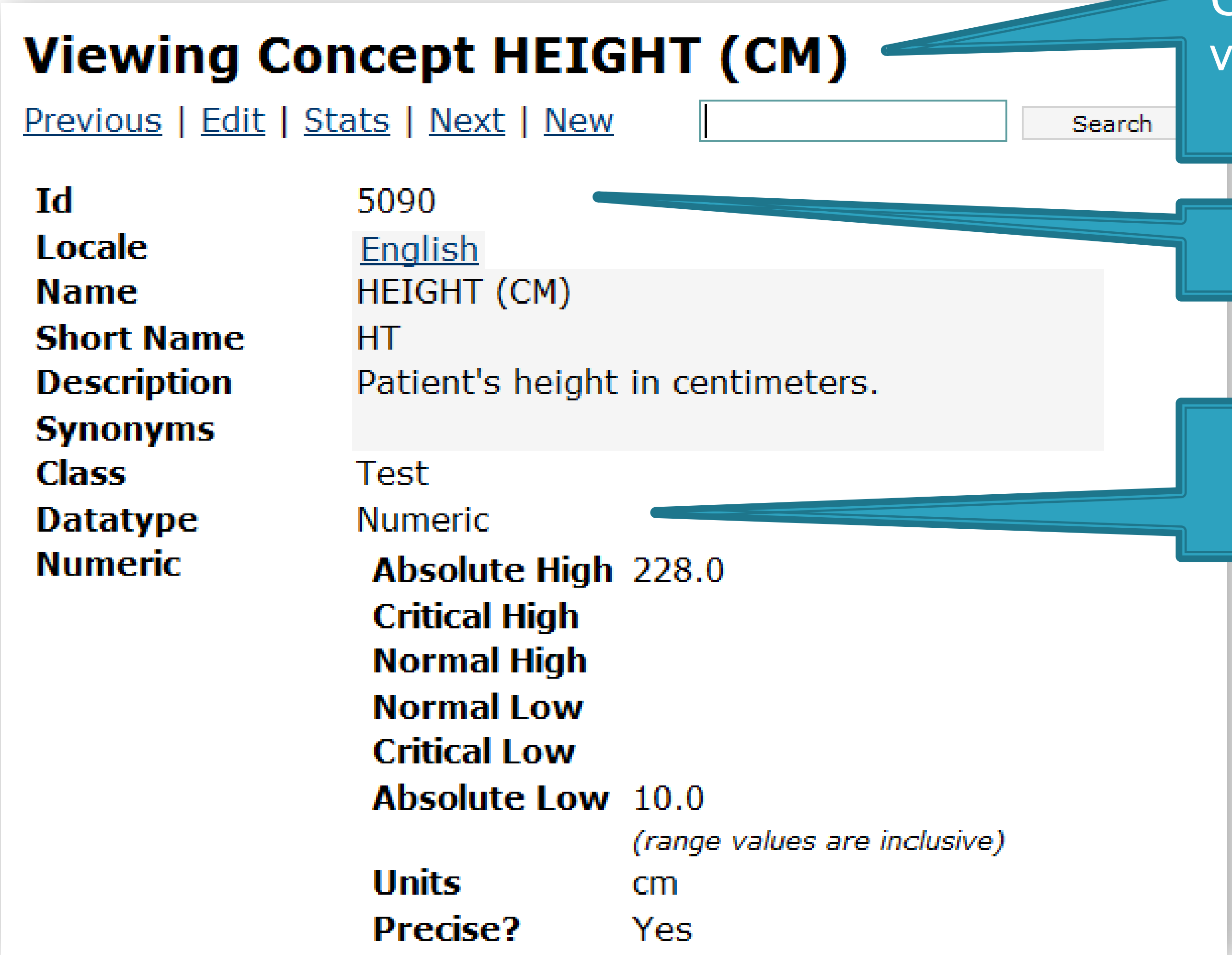
Obs

- Look inside the `Obs` class and you'll see all these fields…

```
protected Concept valueCoded;
protected ConceptName valueCodedName;
protected Drug valueDrug;
protected Integer valueGroupId;
protected Date valueDatetime;
protected Double valueNumeric;
protected String valueModifier;
protected String valueText;
protected String valueComplex;
```

Each of these has a getter and setter

Example: numeric



Viewing Concept HEIGHT (CM)

Id	5090	
Locale	English	
Name	HEIGHT (CM)	
Short Name	HT	
Description	Patient's height in centimeters.	
Synonyms		
Class	Test	
Datatype	Numeric	
Numeric	**Absolute High**	228.0
	Critical High	
	Normal High	
	Normal Low	
	Critical Low	
	Absolute Low	10.0 *(range values are inclusive)*
	Units	cm
	Precise?	Yes

Concept name is very specific and includes units

Concept id

Numeric value types

Example: numeric

- To create a height observation, create an instance of `Obs` and set its properties…

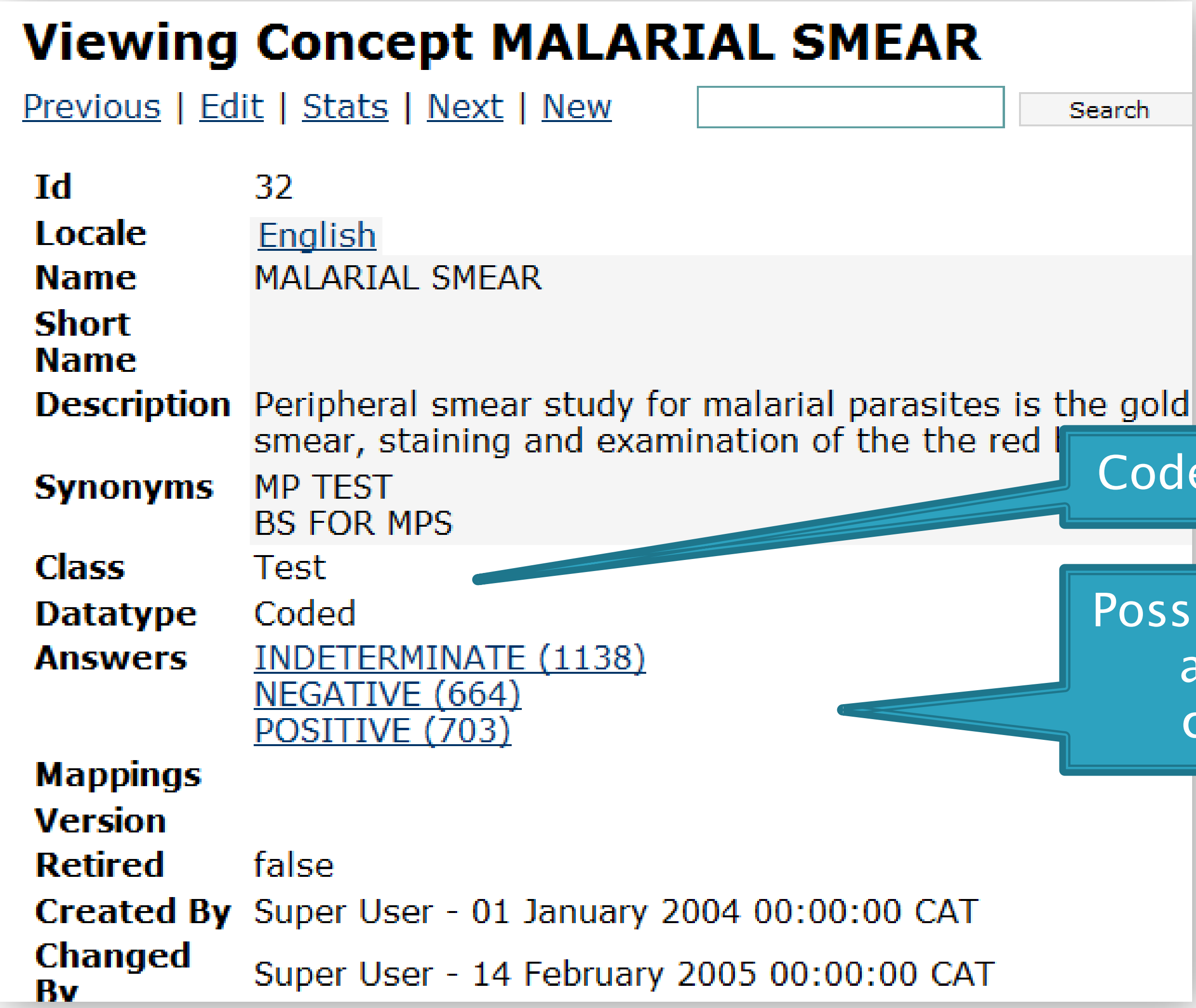
```
Obs obs = new Obs();

obs.setPerson(patient);
obs.setLocation(location);
obs.setObsDatetime(new Date());

obs.setConcept(conceptService.getConcept(5090));
obs.setValueNumeric(150);
```

Existing patient and location objects

Sets the value as a number

Gets the concept for HEIGHT by its concept ID

Example: coded



Viewing Concept MALARIAL SMEAR

Id	32
Locale	English
Name	MALARIAL SMEAR
Short Name	
Description	Peripheral smear study for malarial parasites is the gold smear, staining and examination of the the red l
Synonyms	MP TEST BS FOR MPS
Class	Test
Datatype	Coded
Answers	INDETERMINATE (1138) NEGATIVE (664) POSITIVE (703)
Mappings	
Version	
Retired	false
Created By	Super User - 01 January 2004 00:00:00 CAT
Changed By	Super User - 14 February 2005 00:00:00 CAT

Coded data type

Possible answers are other concepts

Example: coded

- To create a observation of a coded concept, we need to set the value as another concept

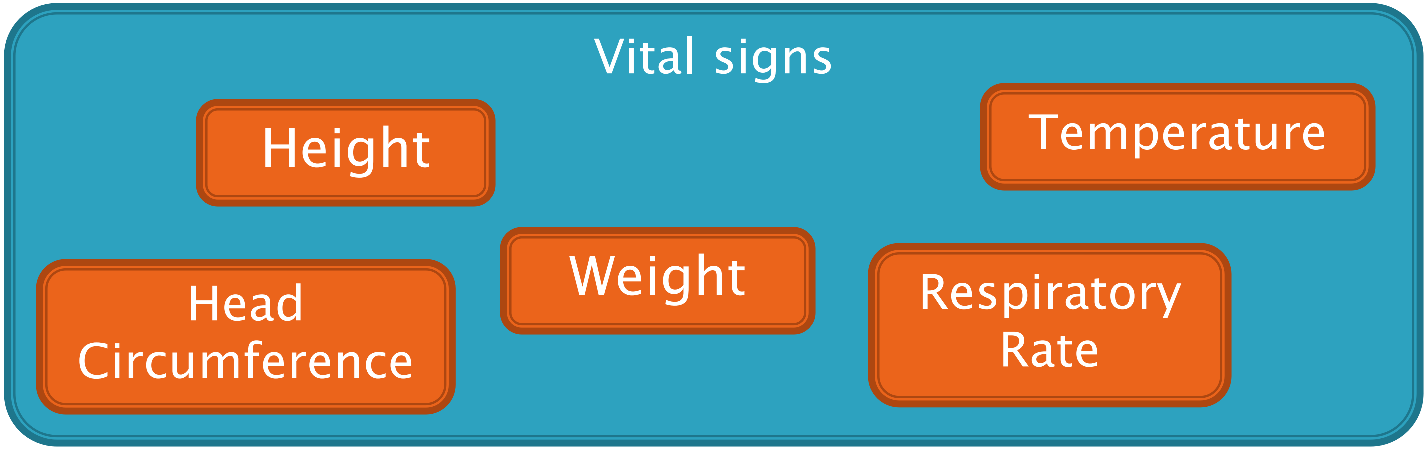
```
Obs obs = new Obs();

obs.setPerson(patient);
obs.setLocation(location);
obs.setObsDatetime(new Date());

obs.setConcept(conceptService.getConcept(32));
obs.setValueCoded(conceptService.getConcept(664));
```

Gets the concept for MALARIA SMEAR by its concept ID

Gets the concept for NEGATIVE

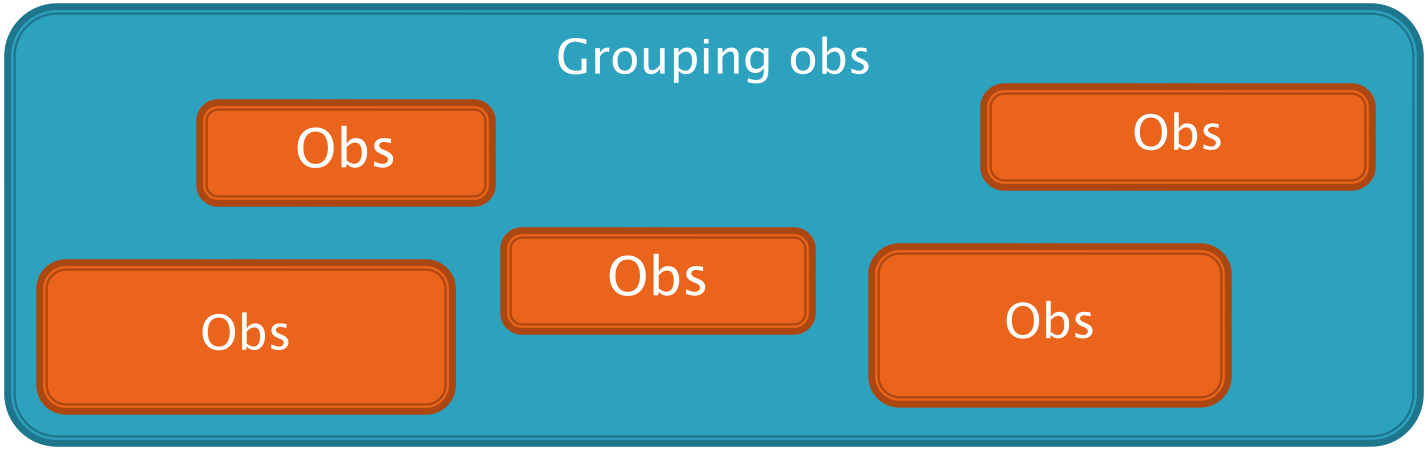
Convenience sets

- Some concepts will often be used together,
  - E.g. when a patient visits the a doctor they often record "vital signs" such as height, weight, blood pressure, temperature
- For convenience these can be grouped into a set of concepts, called a ConvSet

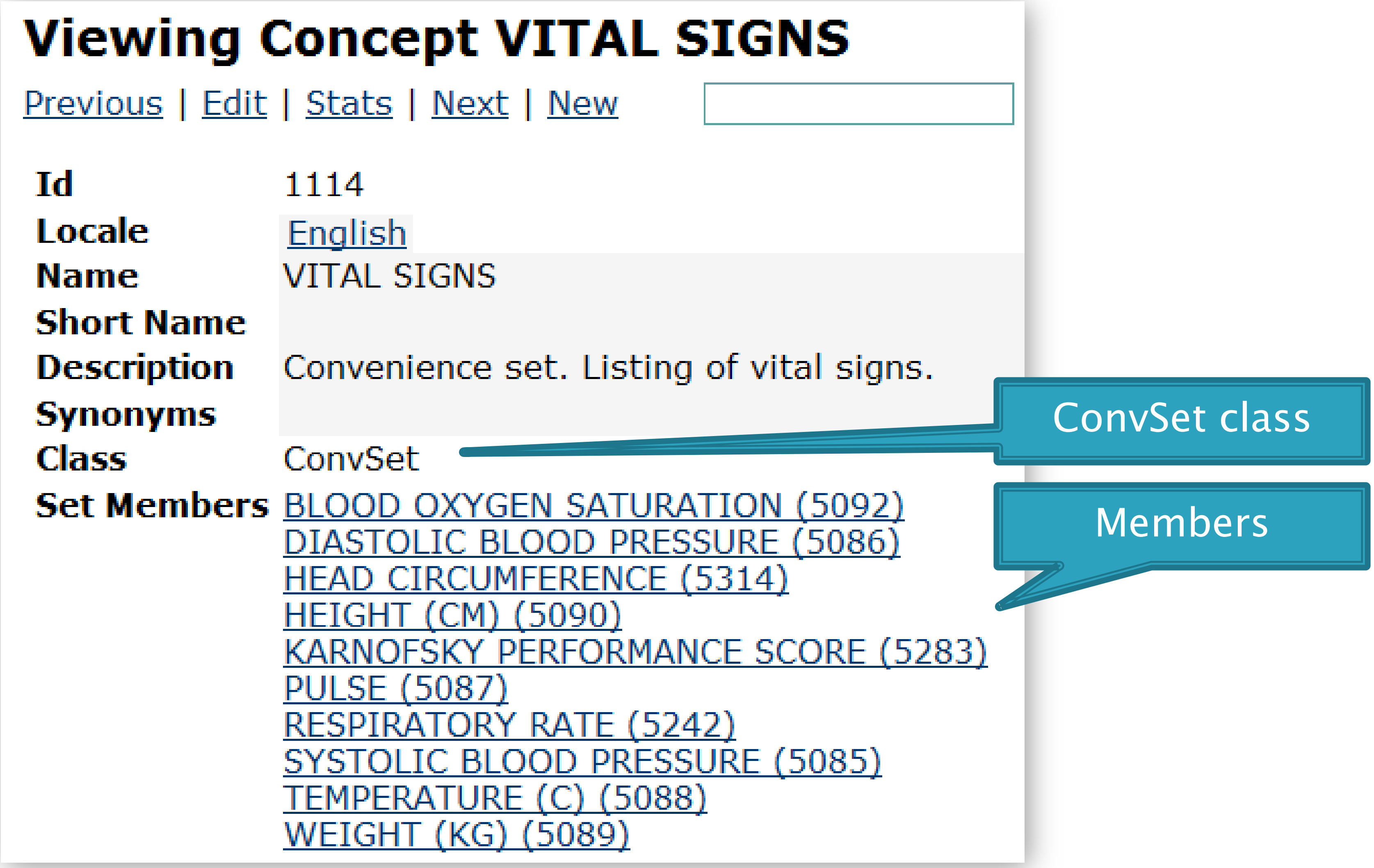


Observing convenience sets

- To create an observation of a ConvSet we need to create an observation grouping
- We add observations for each concept to the grouping as members…



Example: ConvSet



Viewing Concept VITAL SIGNS

Previous | Edit | Stats | Next | New

Id	1114
Locale	English
Name	VITAL SIGNS
Short Name	
Description	Convenience set. Listing of vital signs.
Synonyms	
Class	ConvSet
Set Members	BLOOD OXYGEN SATURATION (5092) DIASTOLIC BLOOD PRESSURE (5086) HEAD CIRCUMFERENCE (5314) HEIGHT (CM) (5090) KARNOFSKY PERFORMANCE SCORE (5283) PULSE (5087) RESPIRATORY RATE (5242) SYSTOLIC BLOOD PRESSURE (5085) TEMPERATURE (C) (5088) WEIGHT (KG) (5089)

ConvSet class

Members

Example: ConvSet

- We first create an obs to be the grouping obs, and then we add the other obs to that, e.g.

```
Obs groupingObs = new Obs();
Obs valueObs = new Obs();

groupingObs.setConcept(conceptService.getConcept(1114)));
groupingObs.setObsDatetime(observationDate);
groupingObs.setPerson(patient);
groupingObs.setLocation(location);

valueObs.setPerson(patient);
valueObs.setLocation(location);
valueObs.setObsDatetime(observationDate);
valueObs.setConcept(conceptService.getConcept(5090));
valueObs.setValueNumeric(23);

groupingObs.addGroupMember(valueObs);
```

Concept id for vital signs

Concept id for height

Height obs added as member

Obs service examples

- To create a new obs, use `saveObs`...

```
Obs obs = new Obs();
obs.setLocation(...);
...
Context.getObsService().saveObs(obs);
```

- To get all the obs for a specific person and concept, use...

```
Patient patient = ...
Concept concept = Context.getConceptService().getConcept(5090);

List<Obs> obs = Context.getObsService().
     getObservationsByPersonAndConcept(patient, concept);
```

Encounters

- Observations typically come from an event such as:
  - The patient visits a clinician
  - A form is submitted which records observations
- An event such as this is called an **Encounter**
- It describes a formal interaction between a provider and a patient
- Encounters can have many observations

Encounters

- Thus for a single encounter we want to store the following:
  - Patient
  - Provider
  - Time/date
  - Location
  - Observations

Encounters

- Encounters are created like observations, with a location, date, and patient
- We must also specify a provider, and add some observations

```
Encounter encounter = new Encounter();

encounter.setEncounterDatetime(date);
encounter.setLocation(location);
encounter.setPatient(patient);

encounter.setProvider(provider);

encounter.addObs(obs1);
encounter.addObs(obs2);
```

Encounter service examples

- To create a new encounter, use...

```
Encounter encounter = new Encounter();
...
Context.getEncounterService().saveEncounter(encounter);
```

- To get all the encounters for a specific patient, use...

```
Patient patient = ...

List<Encounter> encounters = Context.getEncounterService().
     getEncountersByPatient(patient);
```

References

- http://wiki.openmrs.org/display/archive/Concept+Dictionary+Guidelines