

# Concept Best Practices

- [Concept creation](#)
  - [Avoid Duplicates! Avoid Duplicates!](#)
  - [Concept Naming](#)
  - [Describing Concepts](#)
  - [Classes and datatypes](#)
  - [Design for Re-use](#)
- [Modeling diagnoses](#)
  - [Flat by diagnosis \(add coded answers for each diagnosis\)](#)
  - [Fully abstract \(one concept for diagnosis and a 2nd for answer to multiple choice question\)](#)
  - [Abstract, but use existing DIAGNOSIS concept](#)
  - [Flat by answer \(one concept per answer for questionnaire\)](#)
- [Additional information](#)

## Concept creation

Since OpenMRS 1.7, duplicate concept names are not allowed for the same locale. Validation checks for this and will prevent addition of new duplicates or saving a concept with a duplicate name. It is still possible to have duplicate names for different locales (ie. Leprosy could be the concept name for 'en' and 'gb' locale).

## Avoid Duplicates! Avoid Duplicates!

When creating a new concept, use the "similar concepts" resources and/or search your dictionary carefully to be certain that there is not already a concept that meets your needs.

- Create synonyms when you create a concept, to avoid re-creating a duplicate of that concept in the future. This helps avoid duplicate concepts with different phrasings.

## Concept Naming

- Only use alphanumeric characters (letters and numbers) and simple punctuation sparingly – e.g., comma (","), parentheses ("(" and ")"), and forward slash ("/").
  - Descriptions may contain non-alphanumeric characters
- Sentence-case is recommended. This is either all lowercase or uppercase first letter (ie. 'Blue suede shoes').
  - Avoid using all uppercase.
  - Make exceptions for well known names (ie. EKG, HIV, pH)
- For drug names, some people use [Tall Man Lettering](#) to uppercase certain parts of drug names that are confused by providers, since this has been proven to reduce medication errors.
- Be consistent. Having a consistent naming convention for concepts is a must.
- Add standard SNOMED, ICD10, or RxNORM maps to concepts. Do this as part of creating the concept.
  - Consistency in naming helps users predict concept names and makes the dictionary much easier to manage.
- Always work with your end users to ensure the your concept names make sense to them and match their workflows.
- Avoid changing names of concepts that have been used to store clinical data. After the point of data collection, concepts shouldn't be changed as you're then altering retrospective data. It is fine to continue to add synonyms to concepts, where applicable.

## Describing Concepts

Every concept should have a description (at least within the system locale) that unambiguously describes the concept and, ideally, explains how it is intended to be used. For example, for a concept like "NUMBER OF CHILDREN", the description could clarify the concept with a statement like "A question asked of the patient and representing the number of their biological children, whether or not the children are alive, living at home, or living elsewhere."

## Classes and datatypes

- Don't create boolean concepts for things like "Cough for last 3 weeks" = Yes/No. Instead have Symptom Present (coded), Symptom Absent (coded), and "Cough for last 3 weeks" (as an answer)  
Consider importing concepts (e.g. from CIEL/MVP) when suitable <<https://wiki.openmrs.org/x/ww4JAg>>

## Design for Re-use

Well-designed concepts can be used in several different contexts (e.g., multiple, different forms).

- Avoid making your symptoms & diagnoses concepts boolean; rather, use datatype N/A for these and use them as answers (not questions). This promotes re-use of these concepts.
  - POOR DESIGN: a *boolean* "HIV" concept answered yes/no based on whether or not the patient has HIV.
  - GOOD DESIGN: a *coded* "DIAGNOSES" or "PROBLEM ADDED" concept that is answered with the concept HIV (the HIV concept has datatype N/A)

# Modeling diagnoses

Here are some notes/conversations related to concept modeling that may be helpful resources when confronting modeling issues.

Hamish wrote:

If I create a concept "malaria" say a boolean and I want to also have "Malaria diagnosis date" how do I link these two items so that it is clear that they refer to the same item?

Also as Darius and I discussed today what if Carole our epidemiologist friend says she wants all the diagnoses for her data collection form to have true, false, unknown and "no data available". Should we just create hundreds of coded concepts for this or formalize it as a standard construct? If we formalize it does that help us to handle such data efficiently in the same analysis and data extraction tools that normally take a boolean?

I would use DIAGNOSIS or PATIENT REPORTED DIAGNOSIS or similar generically and then record the diagnoses as answers (e.g. MALARIA) for most cases. That gives you the choice of recording the date of diagnosis into the obs date (probably simpler) or creating a DATE OF DIAGNOSIS observation to link dates to diagnoses through an obs\_group\_id.

If someone wants multiple choice answers for 1..n diagnosis, then you are collecting two observations in each case: a multiple choice answer and the diagnosis for which the answer applies. There are several ways to skin this.

## Flat by diagnosis (add coded answers for each diagnosis)

MALARIA with coded answers TRUE, FALSE, UNKNOWN, NO DATA AVAILABLE

PRO  
simple, MALARIA/etc could still be used as answers

CON  
not scalable, ties you to one model of answers to diagnoses and these must be replicated/managed for **every** diagnosis in the system.

## Fully abstract (one concept for diagnosis and a 2nd for answer to multiple choice question)

INQUIRED DIAGNOSIS as coded answered by diagnosis  
DIAGNOSIS STATUS as coded with coded answers TRUE, FALSE, UNKNOWN, NO DATA AVAILABLE  
Tied together with obs\_group\_id

PRO  
easily scalable

CON  
needs tools/knowledge to convert back to one complex data point, diagnoses stored in a questionnaire-specific manner

## Abstract, but use existing DIAGNOSIS concept

INQUIRED DIAGNOSIS as coded answered by diagnosis  
DIAGNOSIS STATUS as coded with coded answers FALSE, UNKNOWN, NO DATA AVAILABLE  
Linked with obs\_group\_id

Store TRUE answers as DIAGNOSIS with coded answer MALARIA, ASTHMA, etc

PRO  
re-use of DIAGNOSIS, so you can still search DIAGNOSIS concept for all known diagnoses

CON  
application must know to treat TRUE answers differently

## Flat by answer (one concept per answer for questionnaire)

DIAGNOSIS  
DIAGNOSIS DENIED  
DIAGNOSIS STATUS UNKNOWN  
DIAGNOSIS DATA NOT AVAILABLE  
each is coded and is answered with the diagnosis

CAROLE1 QUESTIONNAIRE DIAGNOSES OPTIONS as a concept\_set of the above concepts

PRO  
scalable, could re-use DIAGNOSIS concept or have a separate concept for diagnosis = true on questionnaire

CON

requires search for multiple concepts (could be facilitated by concept\_set)

## Additional information

- The [MVP-CIEL concept dictionary](#) is a valuable resource (either as a starting dictionary or to look for best practices)
- [Handling Missing Information](#) is a guide for designing better forms and concepts so as to avoid missing information.
- [Notes](#) or [recording](#) from the 2012-Sep-12 [OpenMRS University](#) forum, where concept management experts discussed best practices.