

Dynamic Mapping Tools Abstract outline

Melanie Philofsky, RN, MS^{1,2*}, Hanan Shorosh^{1*}, Margaret Izzie Clinton², Jue Wang, MFM², Krista Miller, MS, MHA², Michael G. Kahn, MD, PhD², Michelle N. Edelman, PhD², Ian M. Brooks, PhD²

¹Odysseus EPAM, Newtown, PA, USA, ²Health Data Compass, University of Colorado Anschutz Medical Campus

*co-first authors

Background

Health Data Compass, the clinical data warehouse at the University of Colorado Anschutz, uses the OMOP CDM for delivery of research datasets. A weekly ETL process transforms the raw EHR data into the CDM and updates the OHDSI vocabulary tables with the latest codes and mappings. However, when vocabulary tables are updated concepts can change to “non-standard”, or change field / table location, which can cause data to appear to be missing when in fact they have moved to a different field and/or table from where they were originally located. To alert users, we created two tools: the concept mapping stored procedure and the concept mapping table. These tools inform users of the current location of the concept and allow them to find and utilize these and their associated data. The stored procedure also identifies errors in the codes or concept IDs and corrects for common variations in vocabulary input. These tools will allow the OHDSI community to produce reliable evidence more swiftly and accurately.

Methods

Concept mapping stored procedure

The concept mapping stored procedure identifies spelling and other errors in an input table of concept IDs or codes and the associated vocabulary, to be corrected before use in a query. The procedure outputs a table of the input concepts and their current mapped standard concepts and associated domains. Lastly, it cleans the input table by case-correcting concept codes and translating common formatting variations in the input vocabularies by use of the vocabulary translation table.

Initial setup and maintenance of the stored procedure The procedure is designed to run in Google BigQuery (GBQ). The following steps initialize the stored procedure.

1. Create the vocabulary translation table. The table should be named `concept_vocabulary_translation` and contain two string fields named `vocabulary_id_variation` and `vocabulary_id_translation`. The purpose of this table is to translate any input vocabulary variations to their corresponding OMOP standardized vocabulary identifier. An example of an entered row is as follows: `vocabulary_id_variation = "ICD-10-CM"; vocabulary_id_translation = "ICD10CM"`.

2. Ensure the following required OMOP CDM tables exist in GBQ: concept, concept_ancestor, and concept_relationship.

3. Edit the provided SQL code, updating the project id and dataset names to match the current organization's names, then execute the edited code.

Calling the stored procedure

Before calling the procedure, create an input table of concepts to be mapped in GBQ. The input table fields should include an integer field containing concept IDs or two string fields containing codes and vocabularies of the input concepts.

Call the procedure, specifying all required input parameters (e.g. string name of the input table, the string name of the field containing concept IDs, etc.) and a Boolean parameter specifying whether descendent concepts should be returned. Optionally, the name of the output table may be specified in the input/output parameter.

Concept mapping table

The concept mapping table contains all concepts along with their current mapped standard concepts and domains. Since concepts and their associated data can change field and domain / table location when the OHDSI vocabulary tables are updated, this table will inform users where table data are now located.

Initial setup and maintenance of the mapping table

The concept mapping table is designed for GBQ. The following steps should be taken to initialize the table.

1. Ensure the following required OMOP CDM tables exist in GBQ: concept and concept_relationship.

2. Edit the provided SQL statement, updating the project ID and dataset names to match the current organization names. Execute the edited code to materialize the table. To maintain relevant results, the table must be regenerated whenever new concepts are added.

Joining to the mapping table

Before joining to the concept mapping table, the user must first create a GBQ table of clean input concepts to be mapped. The input concepts should be cleaned by running the table through the concept mapping stored procedure and errors corrected before proceeding.

The concept mapping table should be joined to the cleaned input concept table on the concept ID. The concept mapping table may then be joined to a final OMOP table on the standard concept ID or the source concept ID. The example below demonstrates a method of linking the concept mapping table, clean input concept table, and the observation table.

Mapping table join example

```
with concept_mapped as (  
  
  select distinct  
    cm.concept_id,  
    cm.standard_concept_id,  
    cm.source_concept_id  
  from `project.dataset.concept_mapped` cm  
    inner join `project.dataset.clean_concept_ids` c on  
    cm.concept_id = c.concept_id  
  
)  
Select o.* from `project.dataset.observation` o  
  cross join concept_mapped cm  
WHERE (o.observation_concept_id = cm.standard_concept_id or  
o.observation_source_concept_id = cm.source_concept_id)
```

RESULTS

Concept mapping stored procedure

Output table

The concept mapping stored procedure returns a table of mapped standard concepts, current domains, and optionally, mapped descendent concepts of the input concepts. If the input concept is an ICD code, it returns the source concept ID and source concept name. The procedure also returns the user input data for comparison and additional information about the input concepts, including the concept ID, case-corrected code, translated and case-corrected vocabulary, concept name etc.

Error checking

If errors are detected, the concept mapping stored procedure will exit, returning an error message with an error code and description. Additionally, for most errors, an error table is

returned providing more detail. Errors may include incomplete calling of the stored procedure or incorrect input concepts that cannot be mapped.

Concept mapping table

The concept mapping table contains all concepts mapped to standard concepts and current domains. It may be used to join a list of concepts to an OMOP CDM table. Unlike the stored procedure, the concept mapping table does not include the option of returning descendent concepts. If descendent concepts are desired the stored procedure may be used to generate this list and joined to the concept mapping table.

The following issues and changes to the location of data in the CDM are identified by the mapping stored procedure or the concept mapping table.

Issue addressed	Tool	Consequence if not addressed
Typo in the code or concept ID; variable of interest is not found in the concept table	Concept mapping stored procedure	Data are omitted from the concept set, cohort, and study; Data can't be utilized
The concept ID changed from standard to non-standard	Concept mapping stored procedure and concept mapping table	Data are omitted from the concept set, cohort, and study; Data can't be utilized
The concept ID changed domains	Concept mapping stored procedure and concept mapping table	Data are omitted from the concept set, cohort, and study; Data can't be utilized

The following efficiencies are facilitated by the mapping stored procedure.

Efficiency	Efficiency facilitates
The tool corrects anticipated errors and variations in the vocabulary name	The tool prevents an error by correcting for common variations in vocabulary name and spelling.
The tool identifies concepts which have descendent concepts and returns all the descendent concept IDs when the feature is enabled.	The returned descendant concept IDs will each be a row in the output table allowing for their immediate use and future use as an independent concept and not a concept tied to a hierarchy, allowing the data to be mapped to standard concept IDs in the future. None of the descendant concepts will be lost if they are moved out of the hierarchy.

Conclusion

We have received positive feedback from our team who found the inclusion of descendent concepts, changes in field or table location of the data, and identification of erroneous codes and concept IDs helpful in preventing errors of omission when delivering datasets. Future work includes measuring time saved and errors prevented by utilizing the two tools.