

OMOP ETL Design for Transparency, Quality, and Maintainability

Natasha Flowers^{1,2}, Jose Posada^{1,2}, Hannah Morgan-Cooper^{1,2}, Farnoosh Sheikhi^{1,2}, Shikha Kothari^{1,2}, Alvaro Alvarez^{1,2}, Priya Desai^{1,2}

¹ Research Technology, Technology & Digital Solutions, Stanford Health Care, ² Stanford School of Medicine

Background

The process of converting electronic health records from their original format into the OMOP CDM can be very complex and involve different considerations for every site and source system¹. However, there are certain core steps that are common across all implementations, and having a clear understanding of these stages can simplify the development process and provide a more robust final pipeline. Also, most existing data quality checks for OMOP exclusively evaluate the final OMOP dataset^{2, 3}, which is important and useful, but can miss certain issues that arise during the intermediate stages of the ETL process itself. Structuring the ETL into well-defined phases reduces the likelihood of errors, improves maintainability, and provides opportunities for testing and validation throughout the process. At Stanford, we recently rewrote our OMOP ETL around these principles and would like to share our learnings and recommendations from the process for other OMOP sites, whether they are developing an OMOP pipeline for the first time or are hoping to improve upon an existing OMOP pipeline.

Methods

The primary goal of this pipeline design is to ensure that each stage of the ETL process has a clear purpose. This approach helps initial developers to understand and implement the required ETL transformation, and also makes future enhancements, maintenance, troubleshooting and bugfixing more efficient. This workflow is performed for each of the OMOP tables, although some tables will skip certain steps (e.g. the person table does not have any domain routing), and not all sites may require the filtering steps (e.g. Stanford requires a visit have an associated event in order for it to be included, but that may not be required at other institutions).

Stanford's EHR data is extracted from Epic Clarity and hosted on Google Cloud Platform. Our ETL is written in DBT and Google BigQuery SQL, and our testing utilizes DBT's testing framework for both data tests and unit tests. Our pipeline design decisions took into account DBT's dependency modeling, so we have a strict forward dependency structure (no inserts or updates of existing tables), and we ensure the early steps can be run in parallel by pushing all cross-table dependencies to later steps.

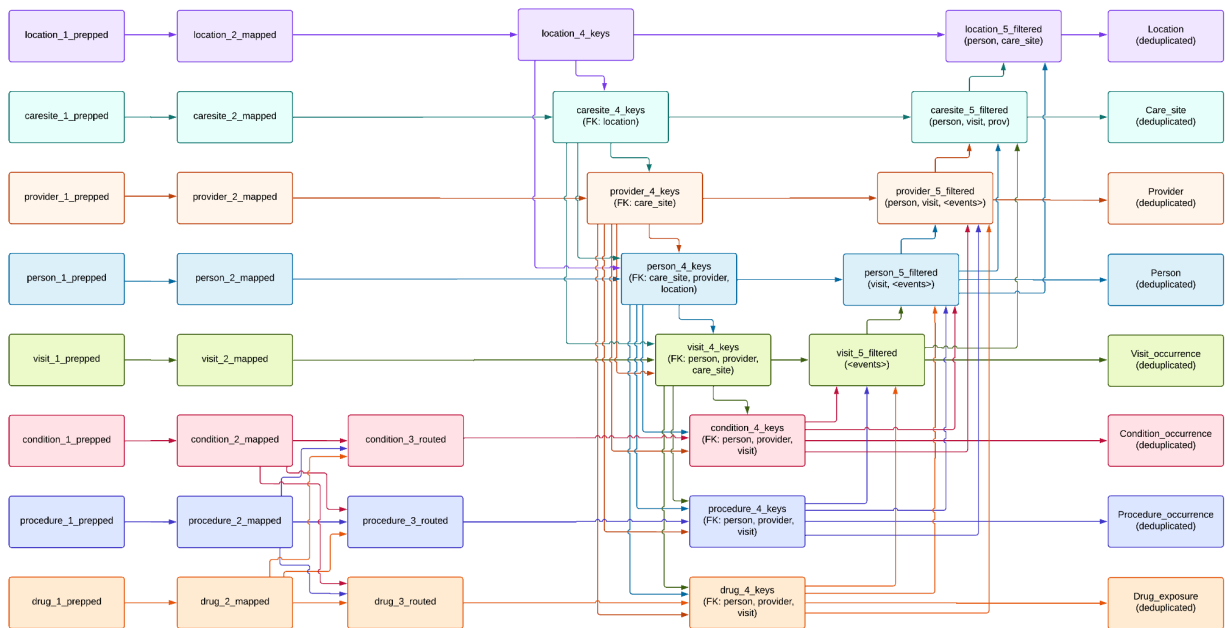


Fig.1 Suggested workflow diagram for an OMOP ETL. Each block represents a table in the ETL; blocks that are in the same vertical column can all be executed in parallel. A selection of core tables are included to demonstrate the concept, and the workflow would be extended to the remaining tables in practice. Not all steps are required for each individual table. Filtering steps shown are based on Stanford's internal inclusion criteria and may not be applicable for all instances.

Step 1: Prep the source data

- Perform all initial transformations (joins, filtering, combining, etc) to the source data
- Retain source primary keys and original field values for later testing/comparison
- Resulting table should have all `_source_value` fields, dates/datetimes, etc (everything besides concept IDs and OMOP primary/foreign keys)
- Test/review:
 - Create unit tests for any complex transformations that are applied to the source data, such as regex to extract numbers or operators from lab results, unit conversions, or hierarchical selection of the final value
 - Review the original vs transformed fields on this table to identify discrepancies

Step 2: Map the source data

- Add every concept ID (source and standard) to the source data from step 1 via joins to the concept/concept_relationship tables or hardcoding
- Retain the source fields as well as the final concept names and domains for ease of verification and domain routing
- Test/review:

- Verify unique combination of source record primary keys to source concept ID - avoid introducing accidental duplication from multiple source concept codes matching the source data (additional rows will still be added from 1:many source:standard mappings)
- Review mappings by comparing source values to concept names; check percentage unmapped by source and/or vocabulary

Step 3: Domain routing

- Move records to their final table based on the domain of the main concept ID for that row
- Retain unmapped concepts (i.e. concept_id = 0) in their starting domain table
- Test/review:
 - Table row counts can change significantly due to domain-routing
 - Verify rows aren't being dropped unexpectedly (i.e. domain-routed out of one table but not into another table)

Step 4: Add primary & foreign keys

- Create tables in order based on the necessary FKs; location, care_site, provider, person, visit_occurrence, then all clinical event tables
- Add what WILL become the primary key, one per each unique combination of fields in the final table (more than one row in the temp table may have the same 'primary key' for now - this makes joins from the source data easier)
- Test/review:
 - Ensure no extra rows compared to the previous step (foreign key joins should be 1:1)
 - Review records where there is a source FK ID but no final FK ID (for example, a source encounter ID is present but no OMOP visit_occurrence_id is populated)

Step 5: Filtering (if applicable)

- Apply any desired filtering criteria based on presence in other OMOP tables
 - Lookup tables (Stanford applies to care_site, location, provider): only include entities in these tables that are referenced in another OMOP table, rather than including all that are present in the source data
 - Qualifying events (Stanford applies to person & visit_occurrence): require that the person or visit have a linked clinical event in order to be included
- Test/review:
 - Check row count drops across filtered tables
 - Verify that only entities referenced in other tables are allowed to remain in the table

Step 6: Final selection of rows & columns (includes deduplication)

- Select only one record for each unique primary key, resulting in deduplication across multiple 'identical' source records
- Select only the final fields that are to be present in the OMOP table, dropping any temp fields that are not needed for the final table

- Test/review:
 - Conformance to final OMOP CDM schema, including PK uniqueness and FK relationships, required fields, standard concepts, domain validity, etc.

Results

At Stanford, this ETL design exposed issues in our prior transformation process that were not readily apparent, and also significantly improved pipeline maintainability by providing a clear entry point for additions or modifications. One key feature of this approach is that the final temp tables provide a direct map between the source data and the final OMOP rows, including all source records that map to the same primary key. This traceability is extremely valuable for speeding up investigations and debugging.

The clear separation of the transformation stages enables multiple steps to be executed in parallel across many tables, as dependencies (joins between tables) only happen at later stages. The explicit lineage makes it easier for developers who are working on a small change to one part of the pipeline to assess the impact of their changes and identify which tables they will need to re-execute as part of their testing.

Conclusion

We have found that this structured approach to the OMOP ETL has significantly improved our data quality, made the transformation process more transparent to ourselves and other stakeholders, and simplified the ongoing maintenance that is necessary for any OMOP instance¹. We hope that other sites with an OMOP ETL or those who are thinking about converting their source data to OMOP can benefit from adopting a similar approach.

References

1. OHDSI. Extract Transform Load. The Book of OHDSI [Internet]. OHDSI; n.d. Available from:
<https://ohdsi.github.io/TheBookOfOhdsi/ExtractTransformLoad.html#ExtractTransformLoad>. Accessed June 8, 2026.
2. Sadowski, Katy. 501-Sadowski-dqdbt-Katy-Sadowski.pdf [Internet]. OHDSI; 2025. Available from:
<https://www.ohdsi.org/wp-content/uploads/2025/10/501-Sadowski-dqdbt-Katy-Sadowski.pdf>. Accessed June 8, 2026.
3. Clair, Andrew. 2-Plenary-1-OMOP-DQ-Clair-Andrew.pdf [Internet]. OHDSI; 2019. Available from:
<https://www.ohdsi.org/wp-content/uploads/2019/09/2-Plenary-1-OMOP-DQ-Clair-Andrew.pdf>. Accessed June 8, 2026.