OMOP ETL Design for Transparency, Quality, and Maintainability

Natasha Flowers, Jose Posada, Hannah Morgan-Cooper,
Farnoosh Sheikhi, Shikha Kothari, Alvaro Alvarez, Priya Desai

OHDSI Community Call, 10 November 2026
Community Contribution Honoree: Data Standards

Stanford MEDICINE | Technology & Digital Solutions
STANFORD HEALTH CARE & SCHOOL OF MEDICINE

Why did we rewrite our entire ETL?

- Inconsistencies, repeated code, lack of readability in existing pipeline after years of additions/bug fixes/porting from CWL -> WDL -> DBT
- Time-consuming to debug or add new features
- Hard to add tests to cover all the different workflows
- New tools available (DBT) to take advantage of

Principles for New ETL

- Each step has a clear purpose
  - The name of the table for that step reflects & reinforces that purpose
- The workflow is consistent across all tables
  - If a step is not needed for a given table, maintain the naming convention for later steps
- Transparency is valued over ‘minimum necessary’
  - Temp fields are kept up until the final step

Steps

Step 1: Prep the source data

Prepare the source data, including cleaning, aggregation, stacking, etc.

Step 2: Map to concept IDs

Map to OMOP vocabulary source concept IDs and then standard concept IDs

Step 3: Route records based on domain

Route clinical event records to the appropriate OMOP table based on domain

Step 4: Add primary & foreign keys

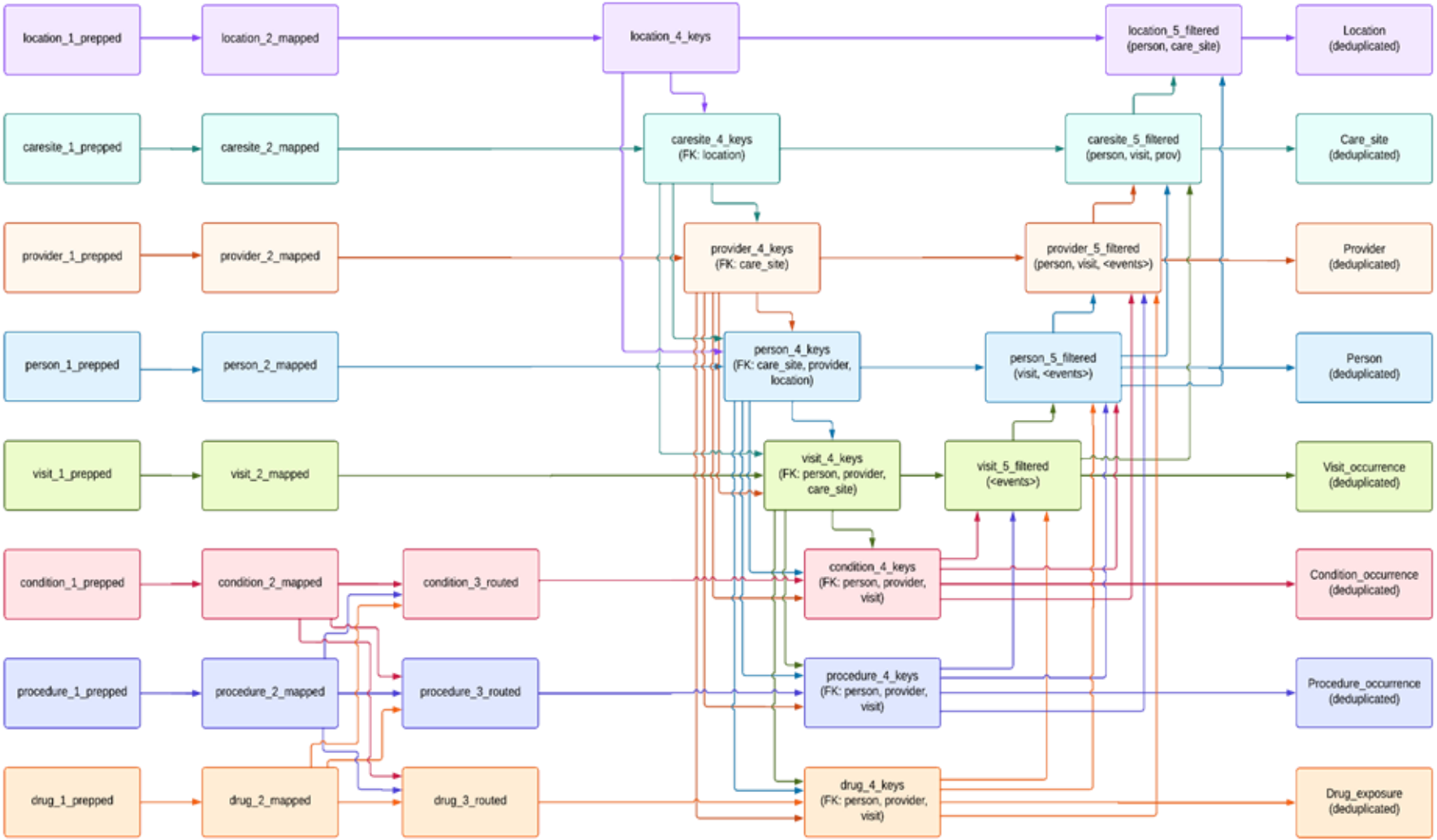
Define a primary key and link to other tables to add foreign keys

Step 5: Filter unneeded records

Filter out extraneous records (if needed - ex. removing care_site_ids with no visits)

Step 6: Select final rows & columns

Select the final OMOP fields & deduplicate to distinct rows


location_1_prepped
location_2_mapped
location_4_keys
location_5_filtered (person, care_site)
Location (deduplicated)
caresite_1_prepped
caresite_2_mapped
caresite_4_keys (FK: location)
caresite_5_filtered (person, visit, prov)
Care_site (deduplicated)
provider_1_prepped
provider_2_mapped
provider_4_keys (FK: care_site)
provider_5_filtered (person, visit, <events>)
Provider (deduplicated)
person_1_prepped
person_2_mapped
person_4_keys (FK: care_site, provider, location)
person_5_filtered (visit, <events>)
Person (deduplicated)
visit_1_prepped
visit_2_mapped
visit_4_keys (FK: person, provider, care_site)
visit_5_filtered (<events>)
Visit_occurrence (deduplicated)
condition_1_prepped
condition_2_mapped
condition_3_routed
condition_4_keys (FK: person, provider, visit)
Condition_occurrence (deduplicated)
procedure_1_prepped
procedure_2_mapped
procedure_3_routed
procedure_4_keys (FK: person, provider, visit)
Procedure_occurrence (deduplicated)
drug_1_prepped
drug_2_mapped
drug_3_routed
drug_4_keys (FK: person, provider, visit)
Drug_exposure (deduplicated)

Why so many steps?

- Improves efficiency of maintenance & development by making it obvious where the relevant code lives
- Allows for testing on intermediate tables for issues that are hard or impossible to see in final table
- Helps with readability of code (for any entity that can do pattern-matching…)

It was worth it!

- Users have benefitted from increased data quality & transparency, as well as speed of adding new features
- Developers have benefitted from clear starting point for debugging or new feature requests

We recommend this kind of structured approach to anyone thinking about updating their existing ETL or writing a new OMOP ETL.

Thank you!

Stanford MEDICINE | Technology & Digital Solutions
STANFORD HEALTH CARE & SCHOOL OF MEDICINE

Questions?

Poster/Abstract Link:

https://www.ohdsi.org/2026showcase-95/