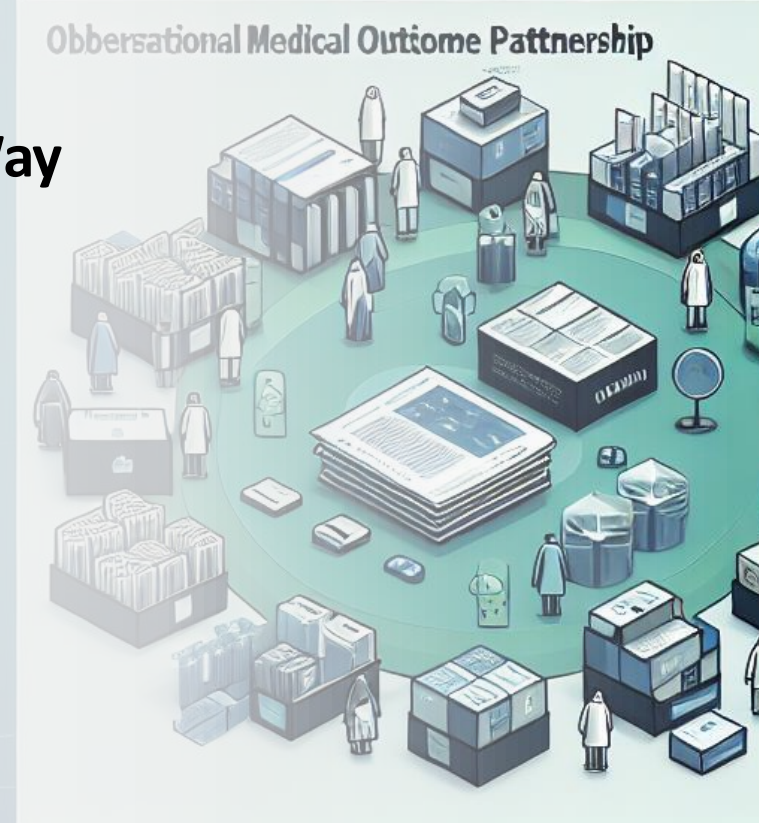
NHS
Lancashire and South Cumbria
Integrated Care Board

Lancashire and South Cumbria
Secure Data Environment

OHDSI/OMOP

The Hard Way is the Easy Way



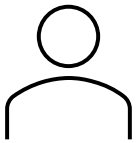
Vishnu V Chandrabalan

OHDSI DevCon 2024

2024-04-26

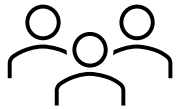
NHS
Lancashire Teaching Hospitals
NHS Foundation Trust

ABOUT



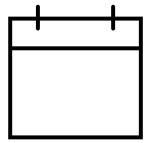
Me

Lancashire Teaching Hospitals
Consultant Surgeon, Head of Data Science
Lancashire & South Cumbria ICB
Director/CCIO - LSC SDE
Lancaster University
Honorary Professor



Us

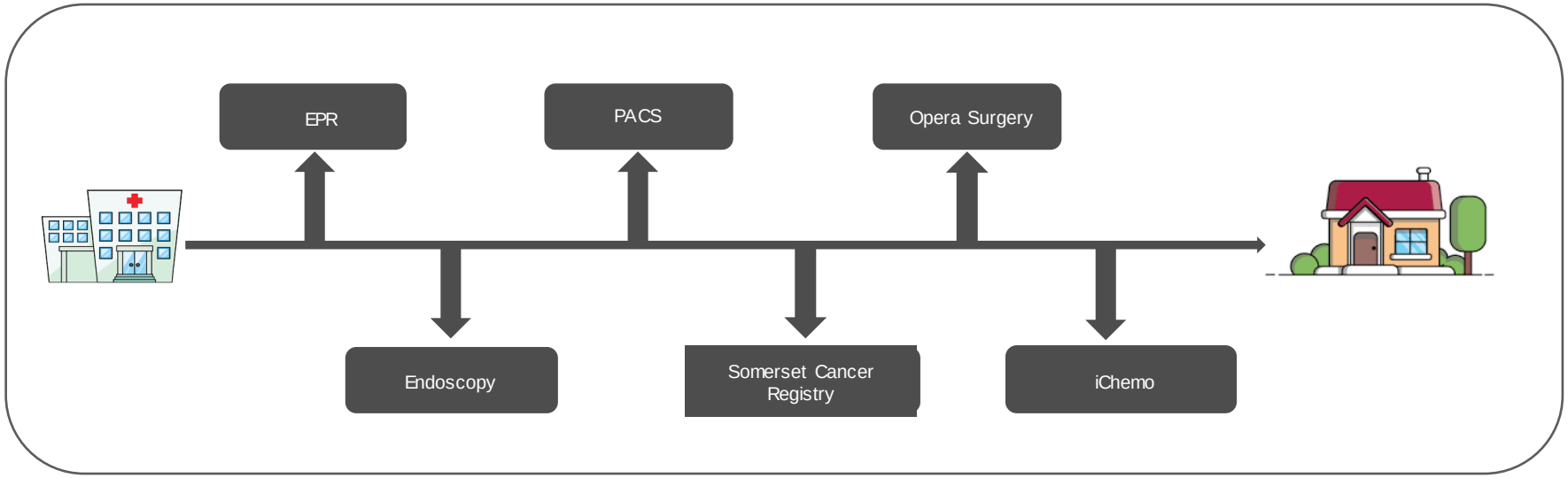
Lancashire Teaching Hospitals
Major trauma centre, 2 sites, Digitally mature(ish)
Lancashire & South Cumbria ICB
5 providers, 1.8M pop, Single-EPR, One-LSC
Part of NWSDE with GM, C&M

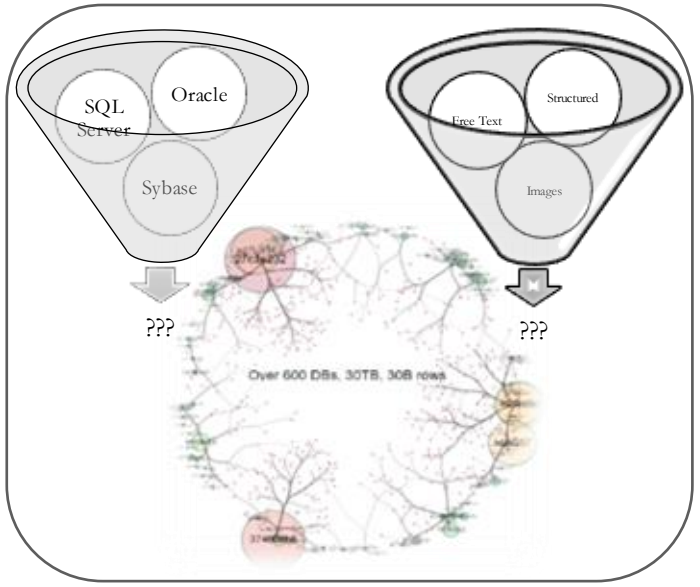


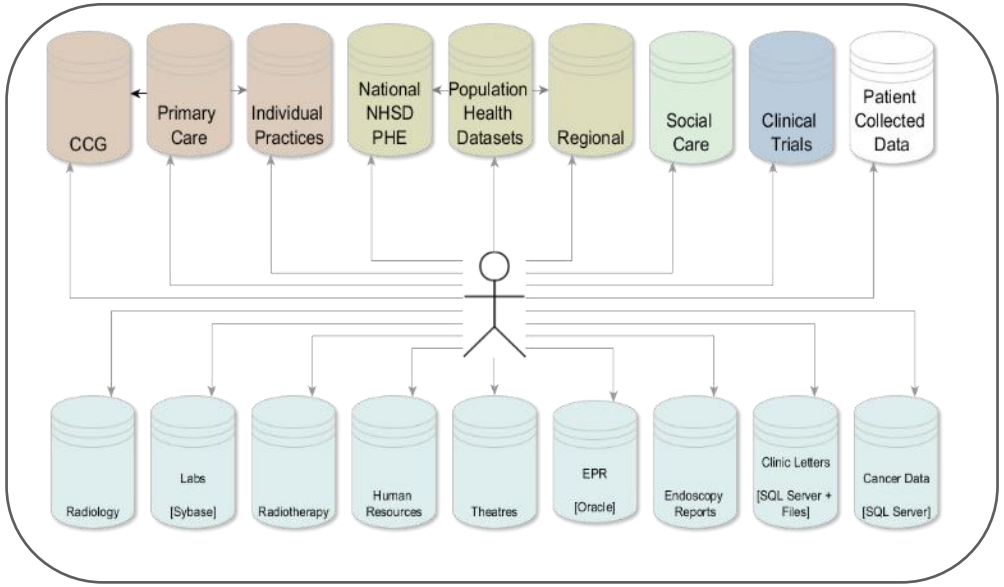
Today

Data Landscape
OMOP Data Engineering
OHDSI Analytics Infrastructure

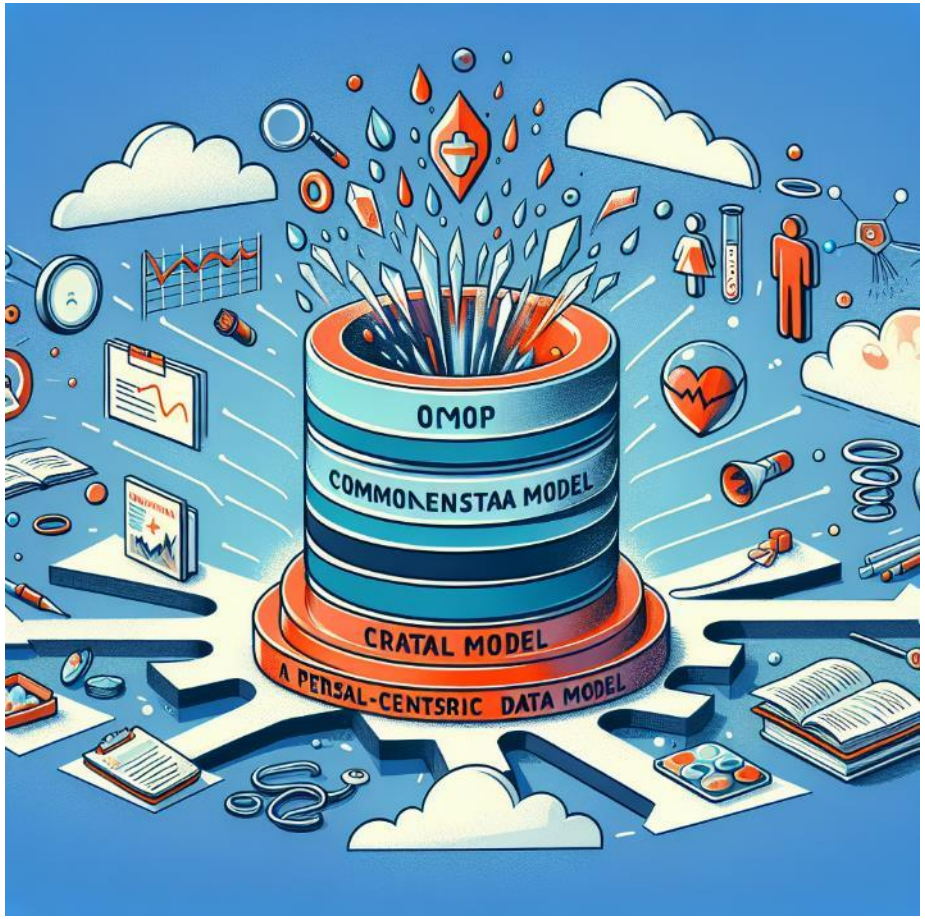
Current Data Landscape



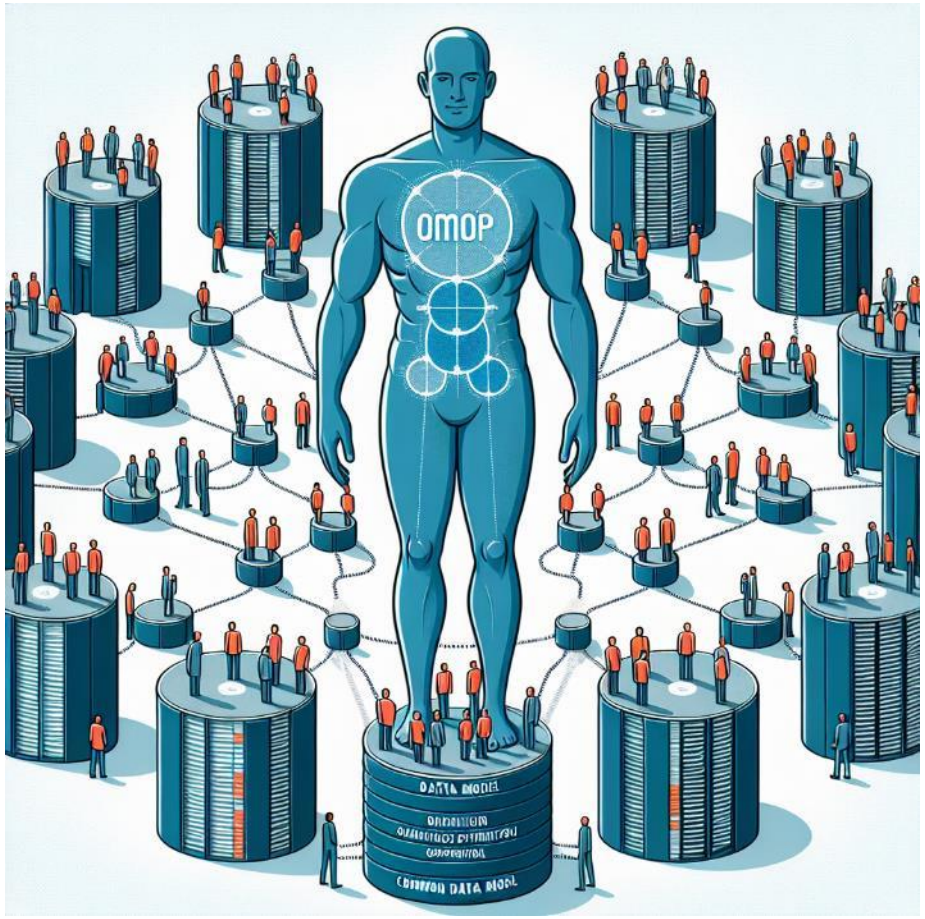




Depict the OMOP common data model as a person-centric data model that shatters data silos and makes research easier.

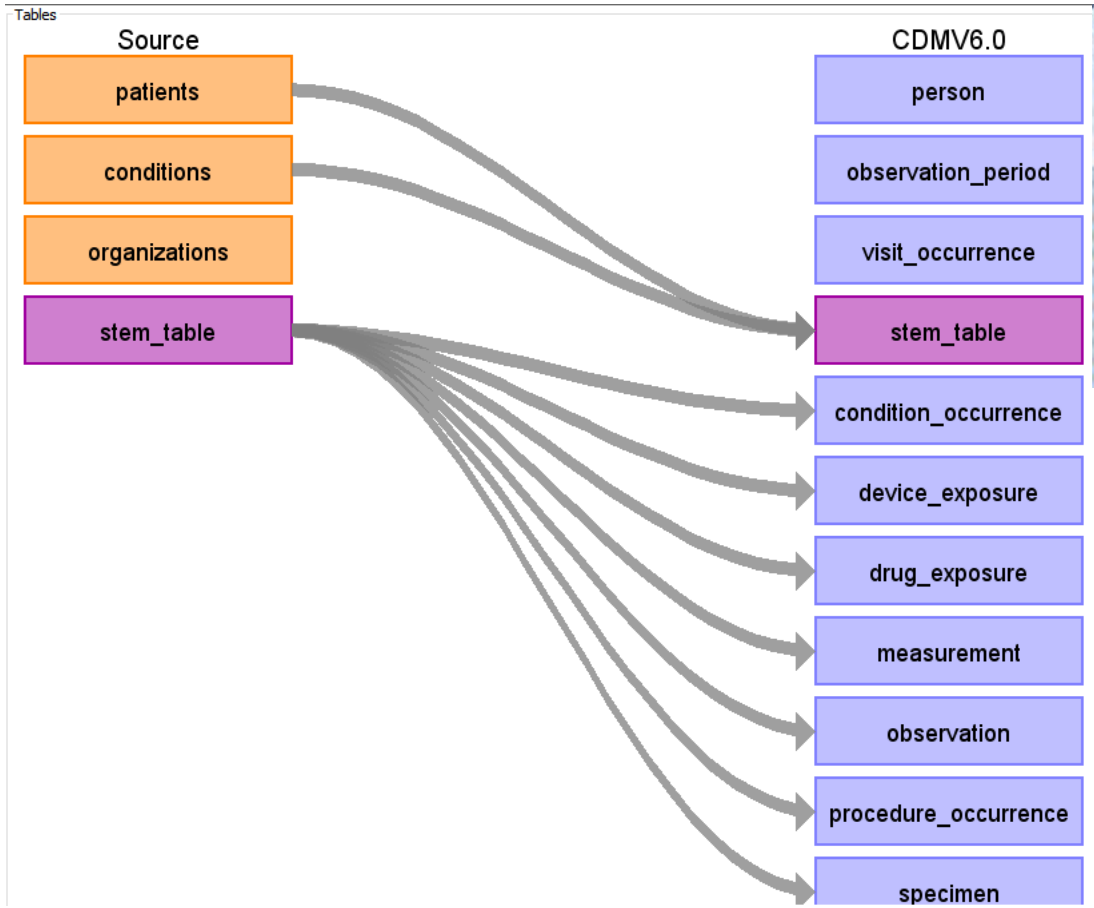


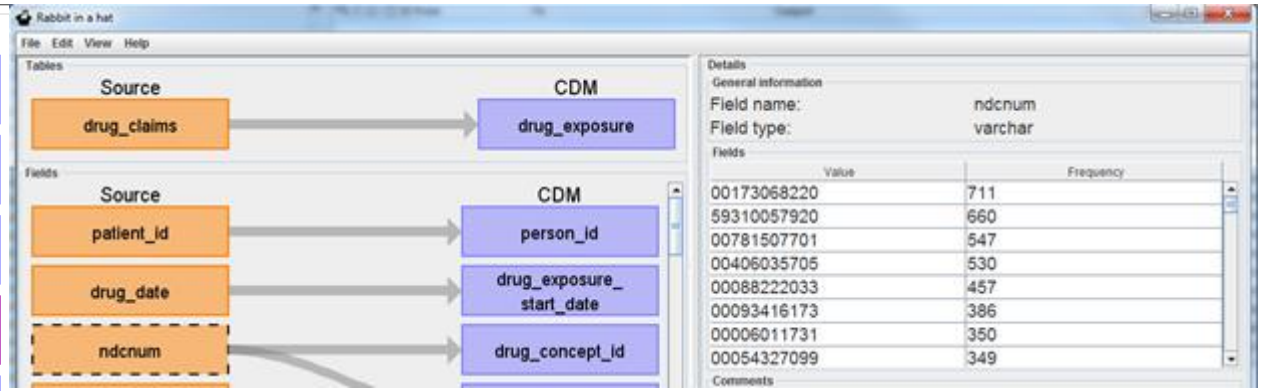
Depict the OMOP common data model as a person-centric data model that brings together data from multiple data silos.



Images created using DALL-E generative AI model on Microsoft Azure OpenAI

OMOP ETL - Rabbit-in-a-Hat





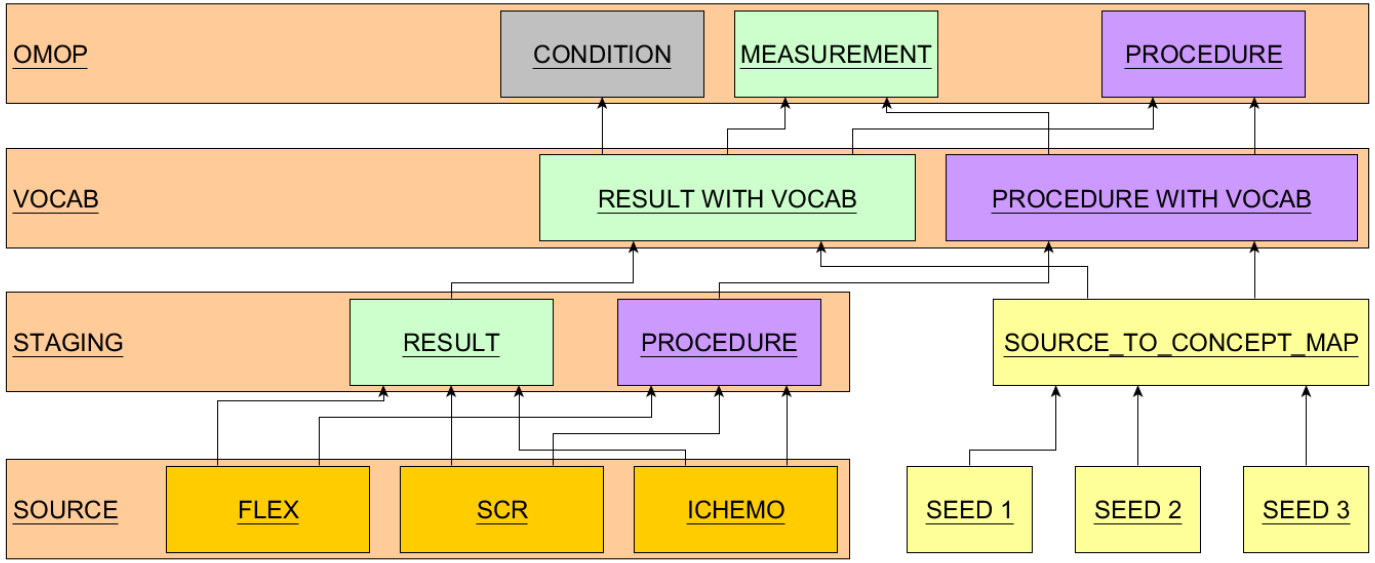
Generating a SQL Skeleton (*v0.9.0*)

The step after documenting your ETL process is to implement it in an ETL framework of your choice. As many implementations involve SQL, Rabbit-In-a-Hat provides a convenience function to export your design to an SQL skeleton. This contains all field to field mappings, with logic/descriptions as comments, as non-functional pseudo-code. This saves you copying names into your SQL code, but still requires you to implement the actual logic. The general format of the skeleton is:

```
INSERT INTO <target_table> (
  <target_fields>
)
SELECT
  <source_fields>
FROM <source_table>
;
```

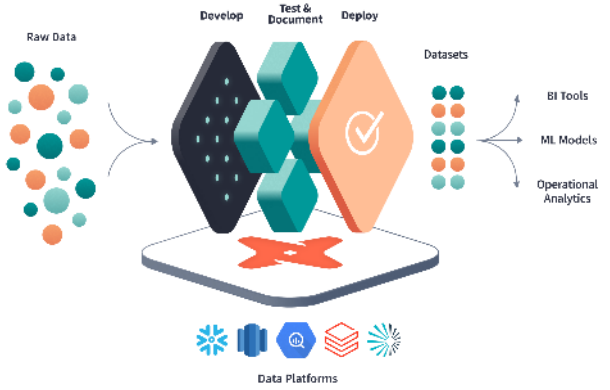

src_flex__ae_procedures
src_flex__stg_procedure
stg_flex__ae_procedures
stg_flex__procedure_occurrence
SOURCE_TO_CONCEPT_MAP
ext__data_opt_out
stg__procedure_occurrence
vocab__procedure_occurrence
PROCEDURE_OCCURRENCE
src_flex__catheter_devices
src_flex__implant_devices
stg_flex__devices
stg_flex__visit_detail
src_flex__visit_detail_id
stg__visit_detail
stg__death
vocab__death
DEATH
stg__device_exposure
vocab__device_exposure
DEVICE_EXPOSURE
vocab__visit_detail
VISIT_DETAIL
vocab__provider
src_src__referrals_map
COST
DRUG_STRENGTH
CDM_SOURCE
EPISODE
NOTE
DOSE_ERA
METADATA
NOTE_NLP
EPISODE_EVENT
OBSERVATION_PERIOD
PAYER_PLAN_PERIOD
src_flex__amp_type_amp_facility
src_flex__amp_facility_most_uses
src_flex__cars_site_gp
ext__postcodes
src_flex__procedure_event
src_flex__cars_site_id
src_flex__cars_site_op
stg_flex__facility_transfer
stg_flex__ae_diagnosis
stg_flex__provider_partial
stg_flex__visit_occurrence
stg__location
stg_ext__care_site
stg_flex__care_site
stg_ui__care_site
stg_flex__condition_occurrence
src_flex__visit_segment
src_flex__ae_diagnosis
src_flex__amp_provider
src_flex__amp_type
stg_flex__amp_type_amp_facility
src_flex__amp_consultant
src_flex__stg_diagnosis
stg_flex__provider
stg_flex__drug_exposure
src_sl__bacteriology_atomic
stg__provider
stg_flex__stg_diagnosis
src_sl__bacteriology_true
cdc__updated_at
src_bi__referrals
src_flex__result
stg_flex__provider_specialty
src_flex__medical_specialty
src_gosport__person
src_gosport__upgosp_procedure
src_gosport__upgosp_diagnosis
src_src__diagnosis
src_src__follow_ups
src_src__investigations_test
src_src__pathology
src_src__procedure_chemotherapy
vocab__source_to_concept_map_raw
vocab__location
src_flex__visit_detail_ae
stg__care_site
src_flex__person
src_src__person
stg__master_patient_index
vocab__source_to_concept_map
LOCATION
vocab__care_site
stg__person
stg__visit_occurrence
CARE_SITE
PERSON
vocab__person
vocab__condition_occurrence
VISIT_OCCURRENCE
vocab__visit_occurrence
fr__measurement_to_observation_ui
stg__condition_occurrence
stg__drug_exposure
PROVIDER
vocab__drug_exposure
DRUG_EXPOSURE
era__drug_target
stg_sl__bacteriology
cdc_sl__bacteriology
cdc_bi__referrals
cdc_flex__result
stg_bi__referrals
stg_flex__result
stg__result
vocab__observation
OBSERVATION
vocab__measurement
MEASUREMENT
fr__mother_to_child
cdc__updated_at_final
FACT_RELATIONSHIP
fr__observation_to_measurement_ui
fr__specimen_to_measurement
stg__specimen
vocab__specimen
SPECIMEN

Difficult when mapping multiple sources into single OMOP instance



OMOP ELT using dbt

https://omop-lsc.surge.sh/

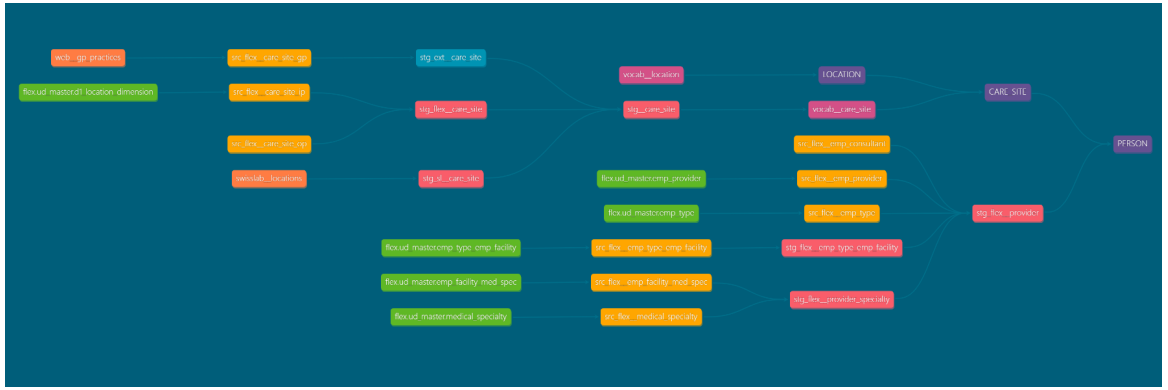


dbt

Data Build Tool

SQL + Jinja + Git

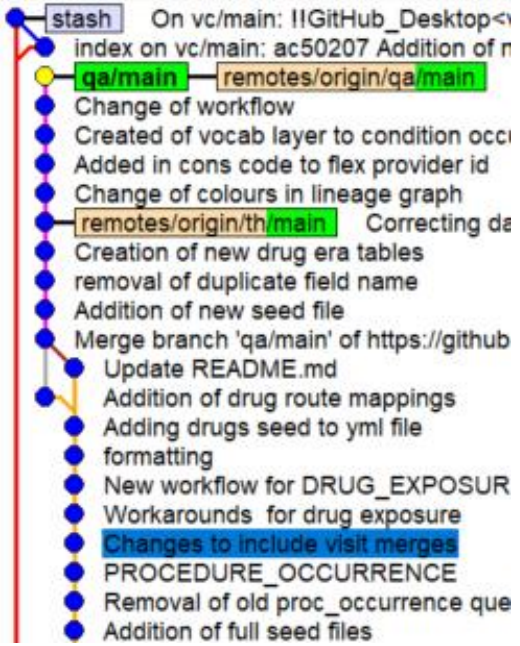
www.getdbt.com



Data Lineage, Self-Documenting, Metadata Generation

Diverse source and target architectures

Shallow Learning Curve

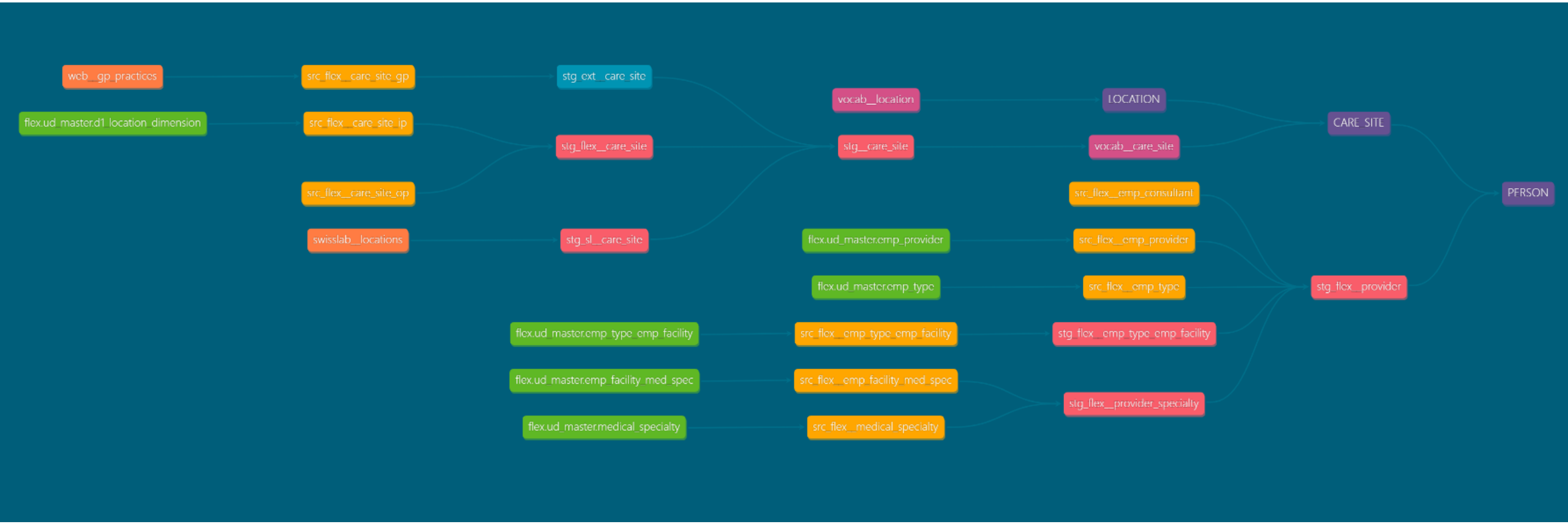


Git + GitHub

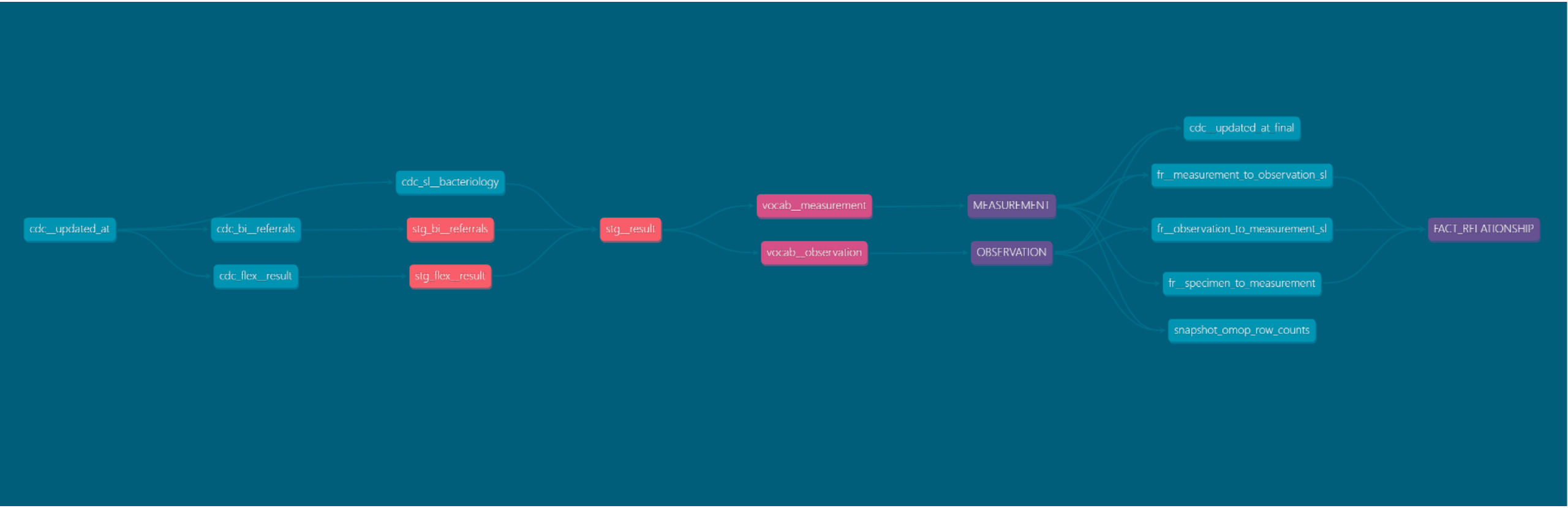
Version Control

CI/CD

Data Lineage - PERSON



Data Lineage – Incremental Refresh



Data Lineage - Vocabulary

Map

Vocabulary from Athena

Usagi for mapping

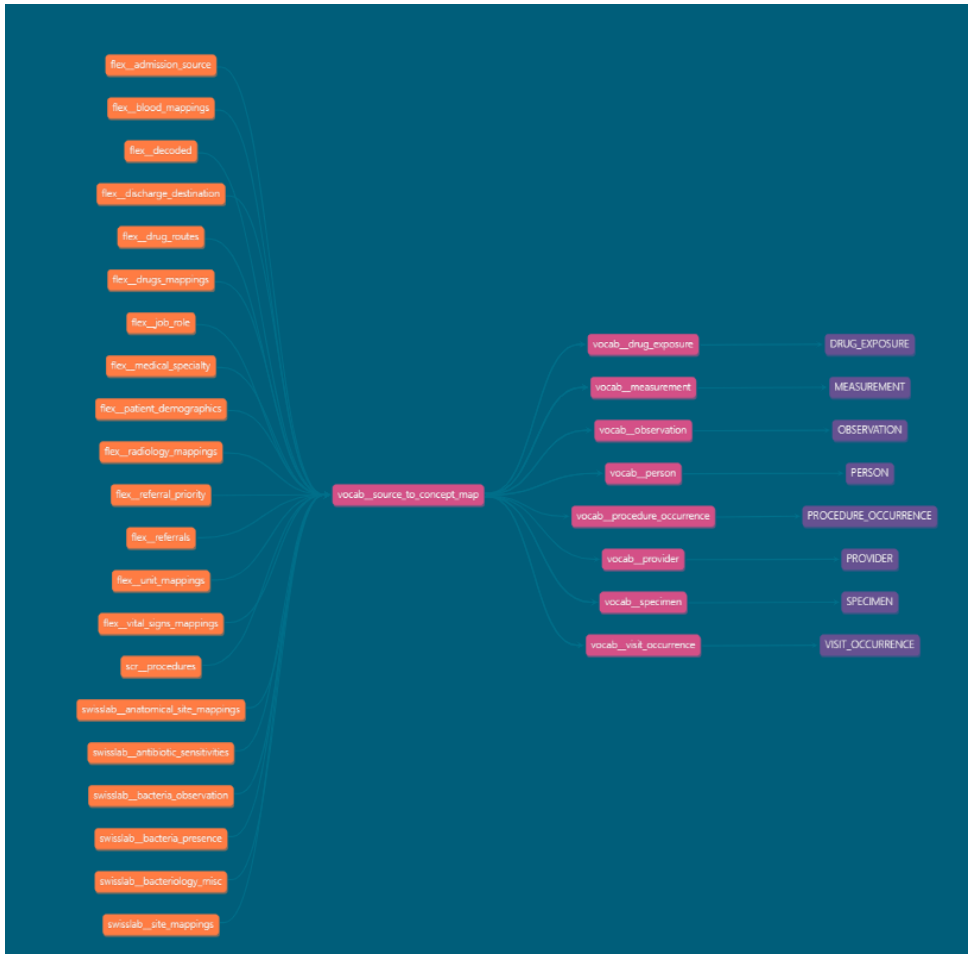
Seed

Multiple source-to-concept CSVs

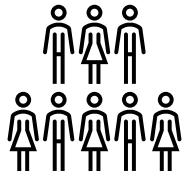
Version-controlled, dbt seeds

ELT

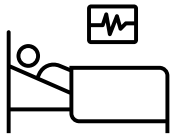
Incorporate into dbt lineage



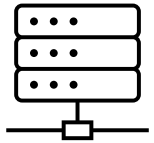
Summary Stats of LTH OMOP



1.8M patients
10.8M OP visits
3.0M ED visits
4.2M IP visits



17.1M Condition occurrences
16.7M Drug exposures
13.0M Procedure occurrences
400M Measurements



Multiple Data Sources
Refreshes every morning
Direct care and Research

HDRUK Team of the Year 2024

OMOP Medallion Architecture on Azure Databricks

Bronze

Raw data ingested from on-prem server

Minimal DQ checks

Focus is on good, efficient, ***incremental*** loading every day with minimal disruption to existing workflows

Silver

Clean, re-identifiable, near real-time data for direct care uses

OMOP with extension tables

Bespoke 'silver products' for results tracking, anti-microbial stewardship, etc.

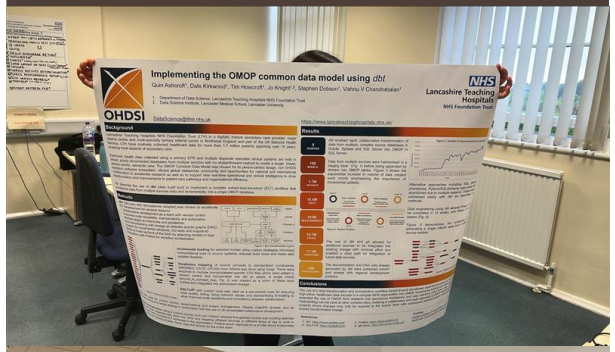
Gold

De-identified, high-quality, research-ready, snapshot for federated OHDSI-style research

REC-approved (subject to DARS and Scientific Advisory Committee review)

OHDSI Collaborator Showcase 2023

https://www.ohdsi.org/2023showcase-20/



Implementing the OMOP common data model using *dbt*

Quin Ashcroft[1], Dale Kirkwood[1], Tim Howcroft[1], Jo Knight[1,2], Stephen Dobson[1], Vishnu V Chandrabalan[1]

1. Department of Data Science, Lancashire Teaching Hospitals NHS Foundation Trust
2. Data Science Institute, Lancaster Medical School, Lancaster University

DataScience@lthtr.nhs.uk

https://www.lancsteachinghospitals.nhs.uk/

Background

Lancashire Teaching Hospitals NHS Foundation Trust (LTH) is a digitally mature secondary care provider, major trauma centre and multi-specialty tertiary referral centre in Northwest England and part of the UK National Health Service. LTH have routinely collected healthcare data for more than 1.7 million patients spanning over 15 years, covering most aspects of secondary care.

Electronic health data collected using a primary EPR and multiple disparate specialist clinical systems are held in siloed, poorly documented databases from multiple vendors with no straightforward method to create a single, linked, person-centric, semantic view. The OMOP Common Data Model was chosen for its person-centric design, rich OHDSI analytics software ecosystem, vibrant global researcher community and opportunities for national and international collaboration to accelerate research as well as to support near real-time operational and clinical intelligence to drive transformation and improvements to patient care pathways and organisational efficiency.

We describe the use of ***dbt*** *(data build tool)* to implement a complex extract-load-transform (ELT) workflow that transforms data from multiple sources daily and incrementally, into a single OMOP database.

Methods

dbt[1] [dbt-core with dbt-sqlserver adapter] was chosen to accelerate ELT development for several reasons:

- Collaborative development as a team with version control
- Improved code reusability, maintainability and automation
- Multiple target architectures and parallelism
- Auto-documenting with lineage as directed acyclic graphs (DAG)
- Support for incremental refreshes, DQ tests and snapshots
- Ability to build sections of the DAG by selecting models or tags
- Integration with Prefect for workflow orchestration

Figure 1. ELT Transformation Layers

Figure 2. Vocabulary DAG Lineage

Incremental loading for selected models using custom strategies minimised computational load on source systems, reduced build times and made daily updates feasible.

Vocabulary mapping of source concepts to standardised vocabularies (SNOMED, ICD10, OPCS4) from Athena was done using Usagi. These were exported to multiple domain/dataset-specific CSV files which were added to version control and incorporated into *dbt* as *seeds*. A single model (*source_to_concept_map, Fig. 2*) was created as a union of these seed models and integrated into downstream lineage.

SQLFluff[2] with custom rules was used as a pre-commit hook for ensuring code quality, linting, fixing common issues and standardising formatting to allow improved code readability and consistency between collaborators.

GitHub was used for version control, issue-tracking and project management. Weekly code/PR reviews, and an internal branching and merge strategy in combination with the use of *dbt* accelerated collaborative development.

Workflow orchestration using a custom solution built with *Prefect*[3] allowed fine-grained control over building selected models and their dependencies within the DAG and targeting different sources at different times of day to work in harmony with other complex ELT workloads in the organisation. Prefect server deployed on a private Azure Kubernetes cluster improved visibility of tasks, flows, logs and failures for the entire team.

Results

5 SOURCES	
106 MODELS	
1.7M PATIENTS	
18.4M VISITS	
141M MEASUREMENTS	
16.7M DRUGS	
17.1M CONDITIONS	
13M PROCEDURES	

dbt enabled rapid, collaborative transformation of data from multiple, complex source databases in Oracle, Sybase and SQL Server into OMOP in SQL Server.

Data from multiple sources were harmonised in a staging layer (*Fig. 1*) before being separated by domain into OMOP tables. *Figure 3* shows the exponential increase in volume of data created each month emphasising the importance of incremental updates.

Figure 3. Number of measurements by month

Figure 4. Project Timeline

Alternative approaches including SQL stored procedures, Python/SQLAlchemy were tried and abandoned due to multiple reasons. These were addressed easily with *dbt* as described in methods.

Data engineering using *dbt* allowed Phase 1 to be completed in 12 weeks, and Phase 2 in 4 weeks *(Fig. 4)*.

Figure 5 demonstrates the complexity of generating a single OMOP table from multiple source models.

The use of *dbt and git* allowed for additional sources to be integrated into existing lineage with minimal effort and enabled a clear path for integration of future data sources.

The documentation and DAG data lineage generated by *dbt* were published online[4] and shared with regional development partners.

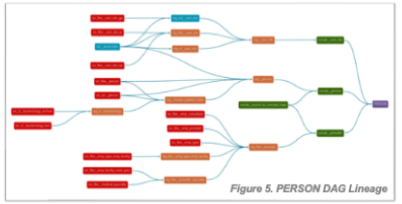


Figure 5. PERSON DAG Lineage

Conclusions

The use of a data transformation and orchestration workflow based around *dbt* allowed rapid harmonisation of multiple, high-value, healthcare data sources in a complex NHS organisation into OMOP. Incremental loading with daily updates extended the use of OMOP from research into operational intelligence and near real-time direct care uses. Similar methodology can be used at other complex sites, enabling a collaborative and open approach to OMOP transformation projects where changes may only be required at the source layer with subsequent transformations done using a shared transformation lineage.

References

1. *dbt*: https://www.getdbt.com/
2. SQLFluff: https://sqlfluff.com/
3. Prefect: https://www.prefect.io/
4. *dbt* docs: https://omop-lsc.surge.sh/

Funding

EHDEN-HDRUK 7th Data Partner Call
NIHR NW Clinical Research Network

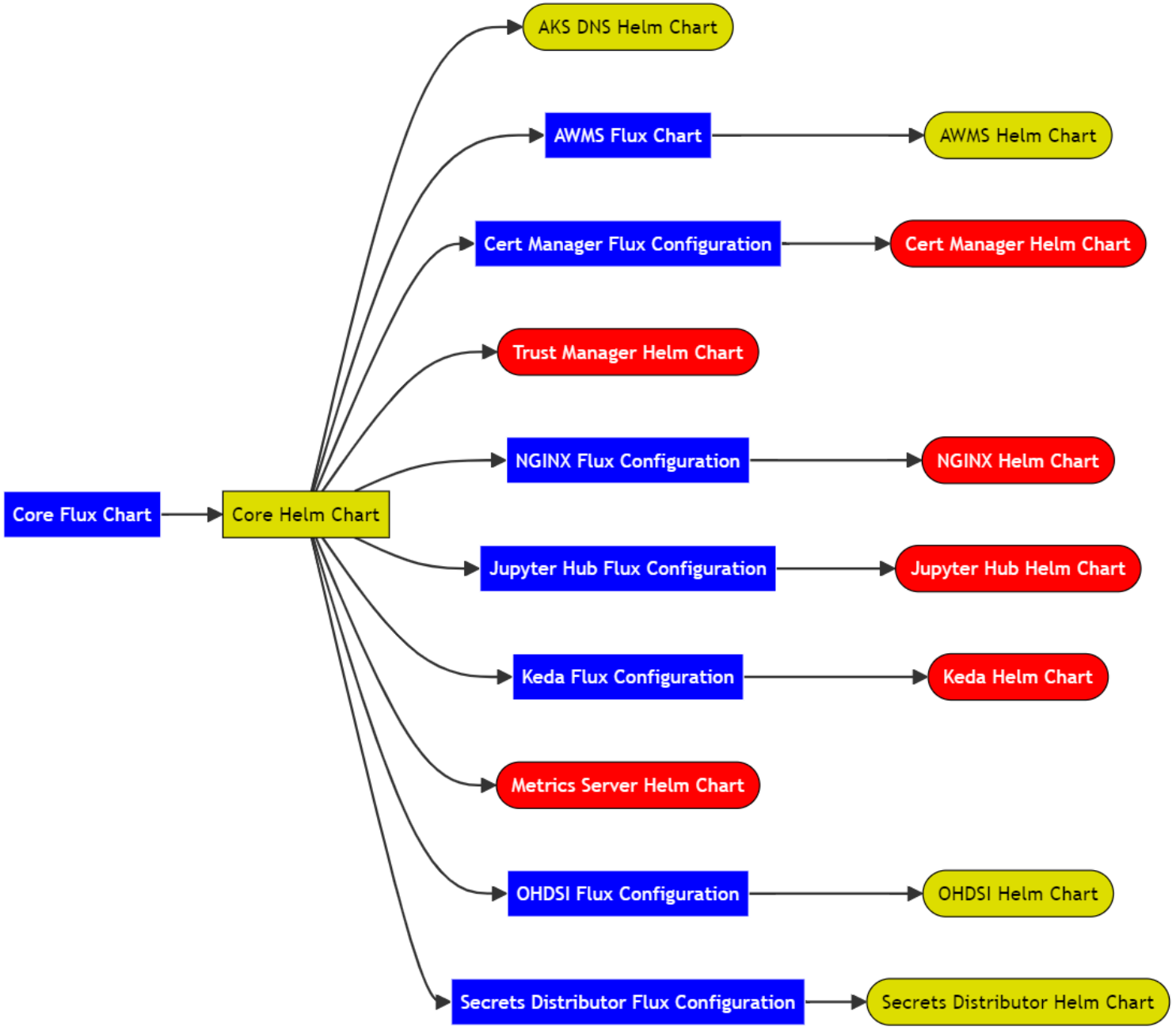
OHDSI on Kubernetes

Current Setup:

- OHDSI on Azure Kubernetes
- Part of an ecosystem of tools
- Open-source Helm Chart and Flux Configurations
- github.com/lsc-sde/iac-helm-ohdsi
- github.com/lsc-sde/iac-flux-ohdsi

In-Progress:

- OHDSI – Databricks integration



Digital Workforce and Research Strategy

The #data_doesn't_save_lives and #lsc_is_a_cool_place_to_work strategy

Research Technology Engineering

Cloud Technical Architects
Kubernetes Engineer
Research Software Engineer
Cloud training pathways with Microsoft
Technology Lead for NW Secure Data Environment

Data Science and Data Engineering

OMOP Analytics Engineer (NIHR CRN funded)
Clinical Scientist

Data science student placements x 9 over 3 years
HEE-funded PHM fellows x 2
NHSX Intern
EPSRC-funded neurology informatics data scientist
Computer vision research
Pharma-funding for nurse researchers in diabetes (TBC)

Career Development

MSc in Healthcare data science (Lancaster University)
PhD studentships x 2 ((Lancaster University, HSST)
EPSRC Collaboration with UoM for RSEs
Centre for Doctoral Training – King's College
New Skills: Python, R, dbt, git + GitHub, Docker, NLP, Solr, OMOP, DevOps, Databricks

Knowledge Transfer Partnerships

Microsoft/Phoenix/Adatis/Kubernetes SME
Oxford Summer School for OHDSI/OMOP training
HDRUK Alliance, SDE Community of Practice groups

Research/Infrastructure Funding

23/24: SDE/EHDEN/HDRUK: £1.41MILLION
24/25: SDE - £1.2M-£1.5M; UKRI £5000
24/26: EPSRC £250K

DALL-E Prompt: Digital workforce strategy that focuses on upskilling, career development pathways and academic research

The Team

- **Quin Ashcroft**, Lead Data Scientist and OMOP Analytics Engineer
- **Tim Howcroft**, Clinical Scientist
- **Dale Kirkwood**, ED Trainee, PPIE Lead - LSC SDE
- **Shaun Turner, Mike Harding,** Cloud Engineers
- **Jo Knight**, Professor of Data Science, Lancaster University
- **Paul Brown, Kina Bennett,** Research and Development
- **Louise Acheson**, Information Governance Lead
- **Saeed Umar, Paul Woodhouse,** Technical Services
- **Stephen Dobson,** Chief Information Officer, LTH