Using dbt—a free and open-source software framework—to transform data into OMOP CDM in the ETL process

Thanapat Pitchayarat, Gun Pinyo, Watcharaporn Tanchotsrinon, Somkid Khamsrimuang, Chalita Issarasittiphap, Chaiyanun Bootnumpech, Noppon Siangchin, Kanphitcha Promma, Nattachai Bovornmongkolsak, Prapat Suriyaphol, Natthawut Adulyanukosol

Siriraj Informatics and Data Innovation Center (SiData+),
Faculty of Medicine Siriraj Hospital, Mahidol University, Bangkok, Thailand

Background

The conversion of medical data into the Observational Medical Outcomes Partnership Common Data Model (OMOP CDM) format requires a managed data engineering pipeline commonly referred to as the extract, transform, and load process or ETL process. Usually, the extraction of data from the sources and the loading into the target databases are more straightforward than the transformation step. The main transformation tasks in a typical OMOP CDM conversion include combining data from multiple sources, changing the original data models to match the OMOP CDM, retrieving the concept IDs of source values, and mapping the source concept IDs to the standard IDs. These tasks are usually executed with SQL scripts. However, the complexity of these scripts may grow rapidly beyond manageable. In order to promote the maintainability of the ETL process along with gaining other desirable features, Siriraj Hospital leverages dbt to transform our data into the OMOP CDM format. dbt™ (shortened from data build tool) is a free and open-source software (FOSS) framework available at https://www.getdbt.com and its core Python library is available at https://github.com/dbt-labs/dbt-core.[1] In this article, we present the conversion of data into OMOP CDM with dbt and accompanying tools at Siriraj Hospital, and how dbt facilitates our data transformation process, which could potentially be applicable to other institutions.

Methods

Siriraj Hospital is an academic health center of the Faculty of Medicine Siriraj Hospital, Mahidol University in Thailand. As of 2022, the hospital has 2,100 beds with approximately 4 million outpatient department visits and 86,000 admissions annually. The hospital has a data lake and a data warehouse in operations for several years, but recently began the OMOP CDM conversion in late 2021.

The data lake at Siriraj Hospital is on an on-premise Microsoft SQL Server database that loads in hospital data from multiple sources nightly. By using Apache Spark, we extract a portion of this loaded data for the OMOP CDM conversion. The conversion begins with the standard mapping specification using OHDSI WhiteRabbit[2], Rabbit-In-A-Hat[3] and Usagi[4] with internal data specialists and medical domain experts. From the specifications, our two data engineers write SQL transformation scripts following the format required by dbt to transform the data structures and map concept codes. All OMOP CDM tables are first materialized on a database in the Development environment. The OMOP CDM-ed data subsequently enters QA and Production environments. Overall, the process is an ELTL, where the first L is the data load into the data lake and the second L is the data load into the production data warehouse. Each step of the ELTL process is containerized with Docker. The whole ELTL process is orchestrated by Apache Airflow, as summarized in **Figure 1**. All codes are version controlled on GitHub.

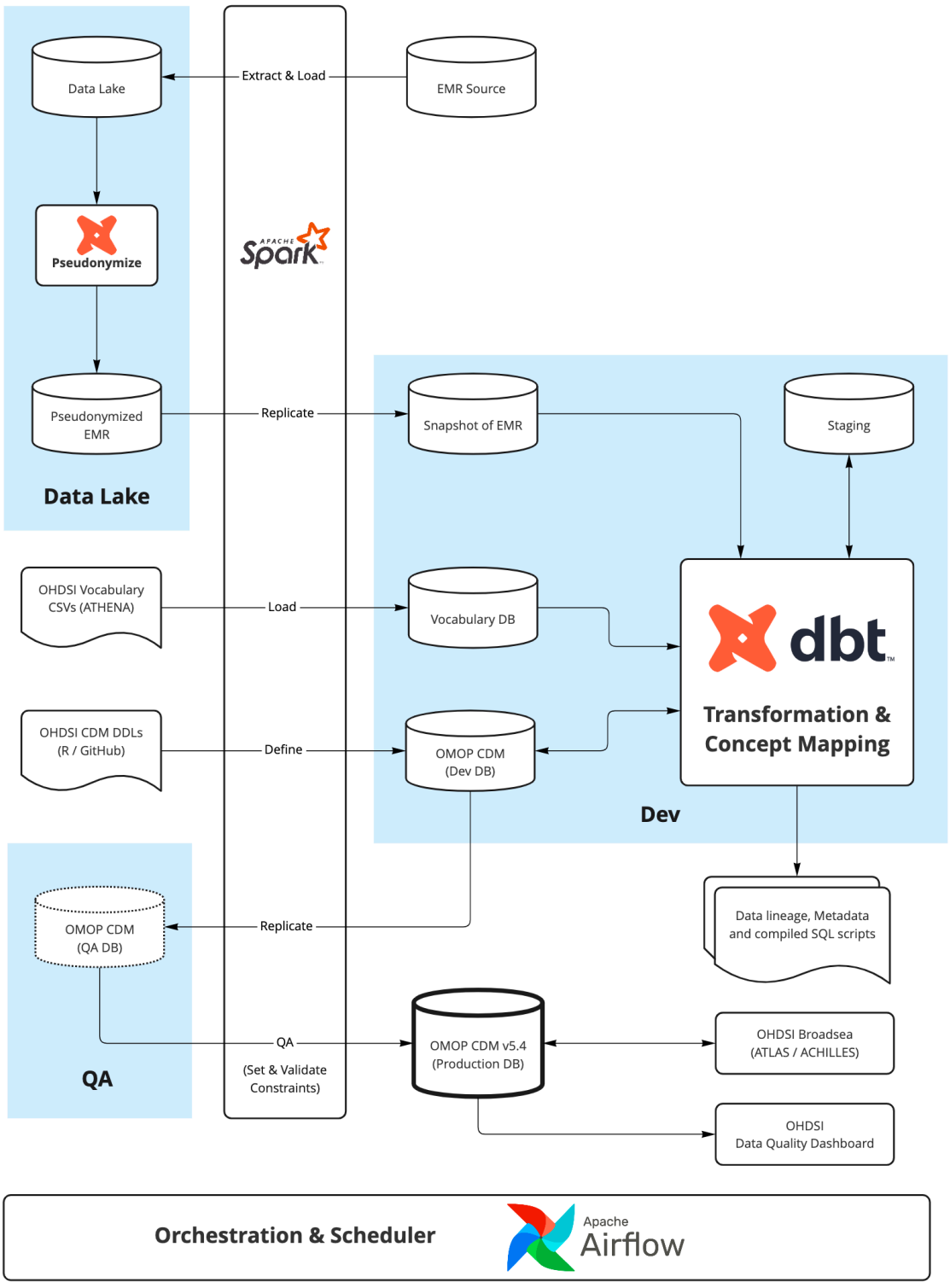

Figure 1: Simplified architectural diagram of the OMOP CDM conversion pipeline at Siriraj Hospital.

Results

The core library of dbt is a Python package that supplements traditional SQL scripts with Pythonic Jinja templating,[5] as exemplified in **Figure 2**. dbt comes with a command-line interface with commands that compile SQL scripts and execute the code on the connected database engines.

With the Jinja templating, any frequently used SQL command can be packaged as a modular macro that can take parameters similar to a Python function. In addition, the Jinja tags enable data lineage tracking that can be visualized on an interactive web application generated by dbt command, as shown in **Figure 3**. The web application referred to as dbt documentation also presents metadata, such as table & field descriptions, data testing conditions, upstream and downstream tables. The metadata are partly generated automatically and can be added manually as YAML files.

To verify data quality, dbt can run automated tests during transformation execution or on demand. It comes with basic tests, such as uniqueness, accepted values, null values, and freshness of the data. More advanced test cases can be added as custom SQL scripts or supplemented by other Python libraries, such as Great Expectations.[6,7] dbt then produces reports of failed tests. Upon failure to pass tests, we can choose to continue or selectively abort the transformation pipeline. This testing functionality provides a valuable complement to OHDSI ACHILLES and Data Quality Dashboard.

Given the popularity of dbt in the enterprise analytics space[8], there are many tools that can be integrated with dbt, namely Airflow for data pipeline orchestration[9] and DataHub for data catalog.[10]

(a)

```sql
-- dbt_project/models/cdm/PERSON.sql

SELECT
  person.patient_id AS person_id,
  gender_concept.concept_id AS gender_concept_id,
  -- ...
  race_concept.concept_id AS race_concept_id,
  -- ...
  -- the rest of SELECT statement omitted for brevity
  -- please refer to OMOP CDM PERSON table for CDM fields
FROM {{ ref('stg__person') }} AS person
{{ map_concept(cdm_table='person', concept_code_field='gender_concept_code',
               vocabulary_id='gender') }}
{{ map_concept(cdm_table='person', concept_code_field='race_concept_code',
               vocabulary_id='race') }}
```

(b)

```sql
-- dbt_project/macros/map_concept.sql

{%- macro map_concept(cdm_table="", concept_code_field="", vocabulary_id="") -%}

LEFT JOIN {{ source('vocab', 'concept') }} AS {{vocabulary_id}}_concept
ON {{cdm_table}}.{{concept_code_field}} = {{vocabulary_id}}_concept.concept_code
  AND {{vocabulary_id}}_concept.vocabulary_id = '{{vocabulary_id}}'
  AND {{vocabulary_id}}_concept.standard_concept = 'S'

{%- endmacro -%}
```

Figure 2: Simplified SQL snippets (a) to create the CDM PERSON table with data from a staging table joined with the vocabulary concept tables via macros (b) to set a macro template for concept mapping. These SQL snippets with Jinja tags are to be compiled and submitted to the database engine by dbt.

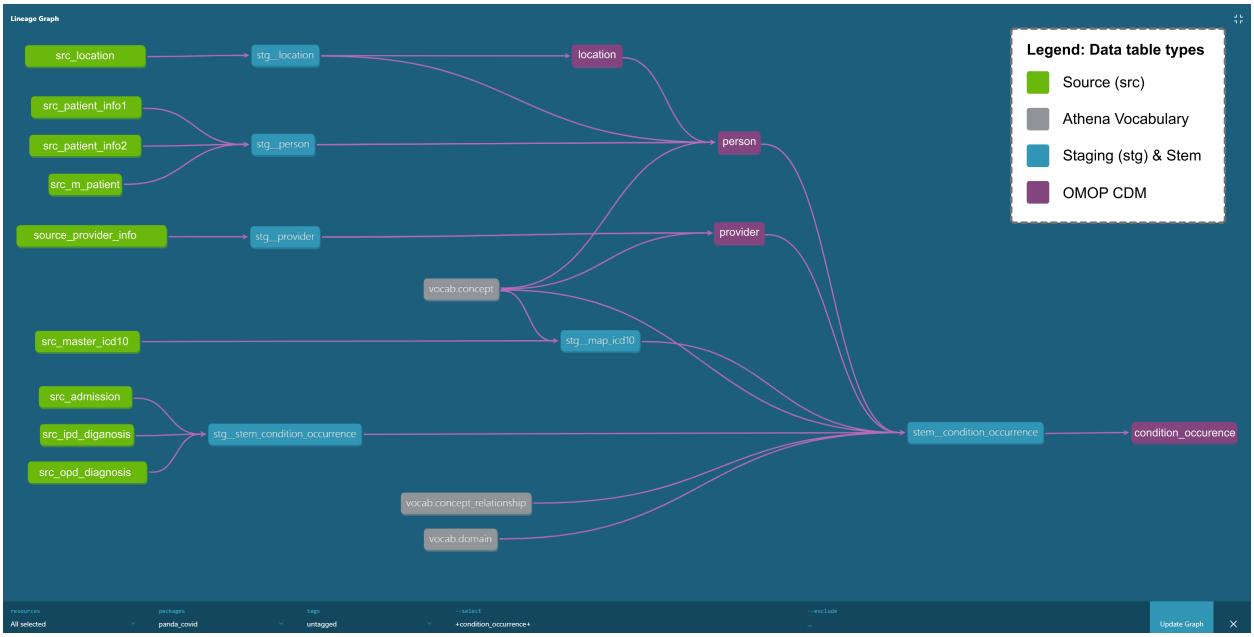

Figure 3: Table data lineage automatically generated by dbt. Each node represents a table or a view of data. Each linking edge represents a data flow from the source(s) to its destination(s), with data transformation in between. Each of the data transformation step is programmed as an SQL SELECT script, as exemplified in **Figure 2(a)**.

Conclusion

dbt is a promising free and open-source software framework that massively facilitates the data conversion process into OMOP CDM. dbt programmatically manages the SQL transformation scripts in the ETL process, and consequently enhances the maintainability of the data pipeline. Its auto-generated documentation from the code surfaces metadata details about each transformation step and visualizes data lineage with all transformation steps. dbt macros ease transformation code reuse and could enable sharing common code with the community. dbt supports automated tests on data conversion, combined with the documentation, improves data quality and data provenance tracking. Data engineers with proficiency in SQL and Python could learn dbt in a few days and probably take a few weeks to implement dbt in the pipeline. In conclusion, we offer a suggestion that dbt can facilitate the data transformation steps in the ETL process of OMOP CDM conversion at any institutions.

References/Citations

1.dbt Labs, Inc.. dbt-core [Internet]. 2022. Available from: https://github.com/dbt-labs/dbt-core

2.OHDSI. WhiteRabbit [Internet]. 2022. Available from: https://github.com/OHDSI/WhiteRabbit

3.OHDSI. Rabbit-in-a-Hat [Internet]. 2022. Available from: http://ohdsi.github.io/WhiteRabbit/RabbitInAHat.html

4.OHDSI. Usagi [Internet]. 2022. Available from: https://github.com/OHDSI/Usagi

5.The Pallets Projects. Jinja [Internet]. 2022. Available from: https://palletsprojects.com/p/jinja/

6.Superconductive Health, Inc.. Welcome to great expectations [Internet]. 2022. Available from: https://greatexpectations.io/

7.Calogica. dbt_expectations [Internet]. 2022. Available from: https://hub.getdbt.com/calogica/dbt_expectations/0.1.2/

8.dbt Labs, Inc.. Success Stories [Internet]. 2022. Available from: https://www.getdbt.com/success-stories/

9.Apache Software Foundation. dbt Cloud Operators [Internet]. 2022. Available from: https://airflow.apache.org/docs/apache-airflow-providers-dbt-cloud/stable/operators.html

10.DataHub Project. dbt [Internet]. 2022. Available from: https://datahubproject.io/docs/generated/ingestion/sources/dbt/

Disclaimer