

# NYC Zoning Tax Lot Database - Refined Workflows

NYC Department of City Planning | Geographic Data Engineering

**Heng Jiang**, Date Engineering Fellow | Carnegie Mellon University, Public Policy and Management – Data Analytics

## Keywords:

data pipelines, database management, quality assurance, process automation, Geographic Information Systems

## Summary:

To refine the building process of the NYC zoning tax lot (ZTL) data product, Heng worked with DCP's Data Engineering team to redesign the SQL infrastructure, code repository, and data pipeline of the ZTL database and implemented the effective data transformation tool – DBT. Utilizing **SQL**, **Python**, and **bash** scripts with the integration of **Docker** containerized environments, **Amazon S3** cloud infrastructure, and **GitHub Actions**, Heng created an automated data product workflow with refined quality assurance features and a more streamlined code base.

# NYC Zoning Tax Lot Database

---

Implementation of DBT and Refined Workflows

NYC Department of City Planning, Data Engineering  
Heng Jiang  
Coding it Forward Fellow of 2024



# Agenda



Introduction to  
Zoning Tax Lots  
(ZTL) and its  
problems



Introduction to  
DBT



Implement  
Directed Acyclic  
Graph (DAG) as  
DBT models



Quality  
Assurance



Takeaways

# About me

---

- Heng Jiang (he/him)
- Coding it Forward Data Engineering Fellow
- Carnegie Mellon MS in Public Policy and Data Analytics Class of 2025
- From China, currently based in Pittsburgh, PA



# Introduction to Zoning Tax Lots

---

# The Zoning Tax Lot Database

- The Database includes the zoning designations and zoning map associated with a specific BBL.
- The Database is updated monthly to reflect rezonings and corrections.



# Data Sources

DCP  
Zoning Amendments

DCP  
Commercial Overlay

DCP  
Zoning Districts

DOF  
Digital Tax Map

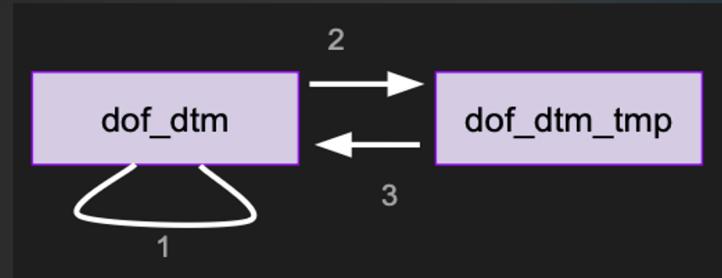
DCP  
Limited Height Districts

DCP  
Zoning Map Index

DCP  
Special Purpose Districts

# Problems with the Old Build Process

1. Repetitive preprocessing steps result in **messy temporary tables**.
2. Difficult to troubleshoot as we **update/mutate data**, making it hard to pinpoint problems.



*Difficult to perform code review and version control with such SQL infrastructure.*

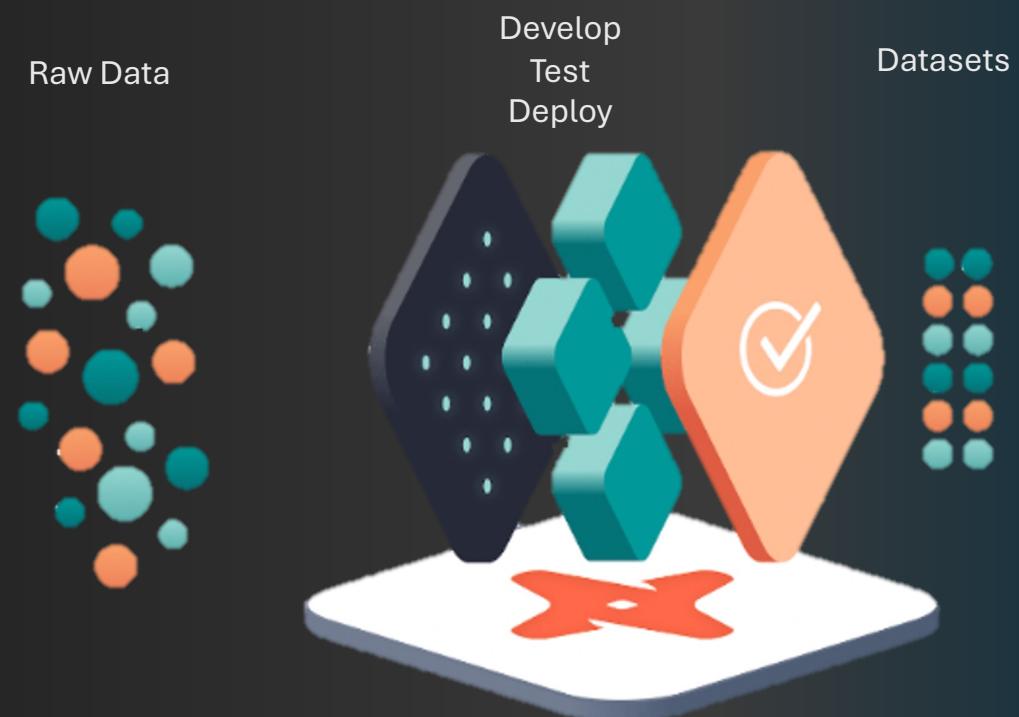
# Introduction to DBT

---

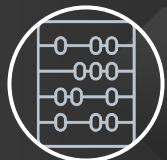


# What is DBT?

- A framework of how we store SQL queries and files.
- A command line tool to run a series of SQL files and achieve data transformations (with a bunch of handy features that come with it).



# Why DBT?



Modular Data Transformations



Automatic Dependency Management



Tests Before Deployments



Macros and Jinja Templating



Dev. Env. and Git Friendly

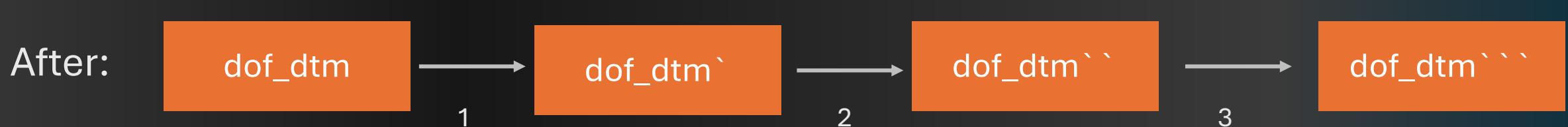
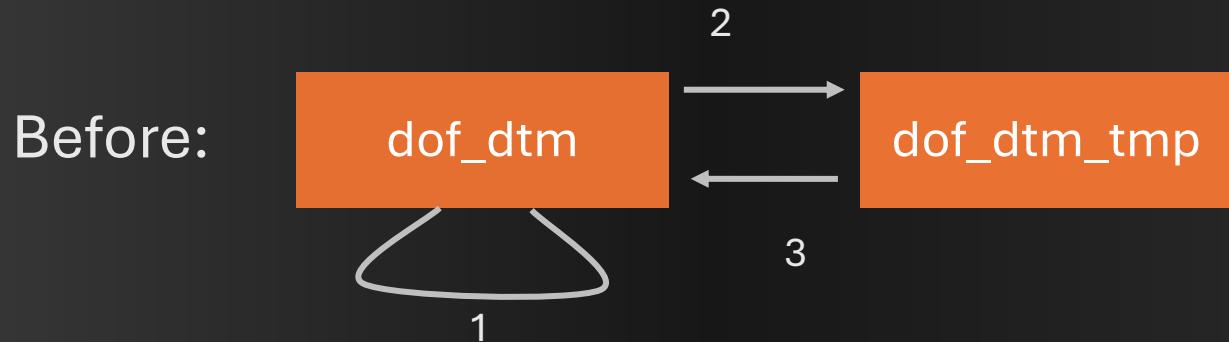


# Implement DBT Models

---

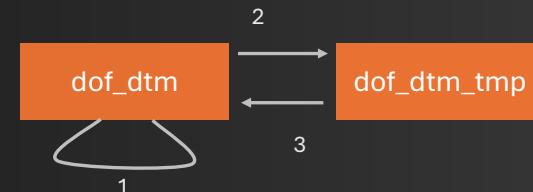


# Directed Acyclic Graph



# Directed Acyclic Graph (DAG) for DBT

- DBT does not support updates or cyclical references



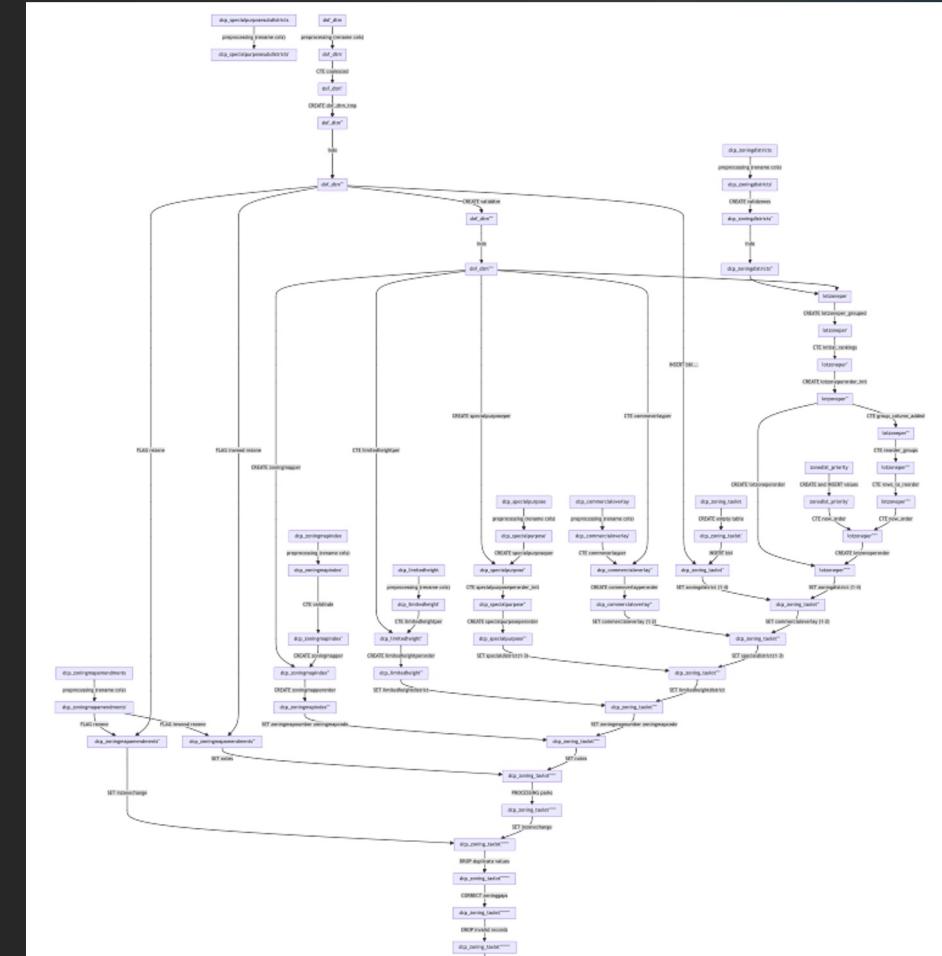
- Move Forward Only!

- We need to restructure the repo and the codes for DBT deployment



# DAG for the Original Transformation

- What our original process looks like assuming no cyclical references and moving forward only.
- Missing Structure



*DAG visual that approximates the original build process*

# Modularize/Group the Transformation

Source/Seed Tables

Staging Tables

Intermediate  
Tables

Product Tables

Load the source tables or  
create seed tables

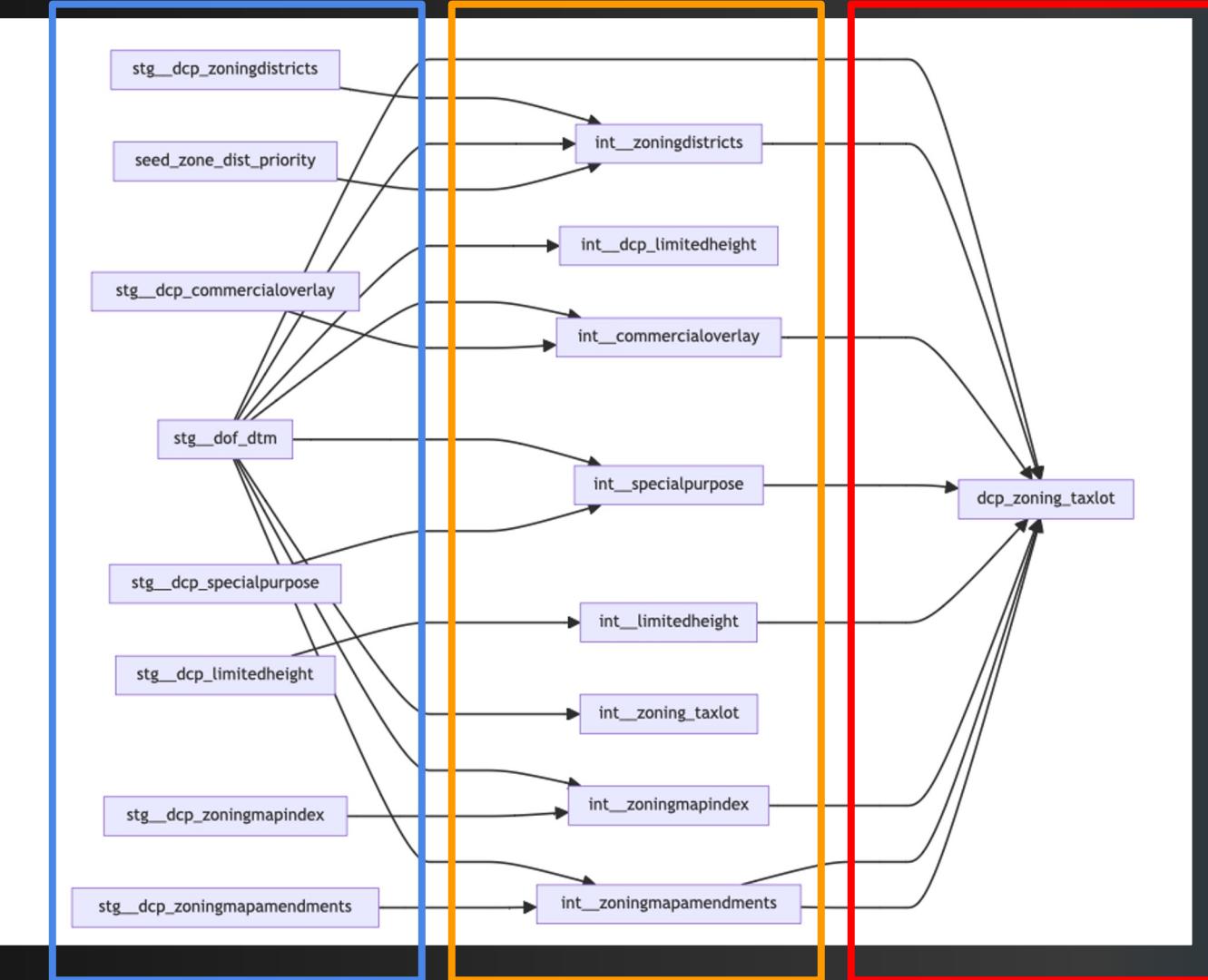
Perform preprocessing  
such as rename  
columns, drop columns,  
etc.

Major transformations  
such as joins and  
filtering across tables.

Final steps to tune the  
output, such as rename  
columns, then our  
product tables will be  
ready to go!

```
✓ zoningtaxlots
  > bash
  > dbt_packages
  > logs
  > macros
  ✓ models
    ✓ intermediate
      ! _int_models.yml
      int_commercialoverlay.sql
      int_inwoodrezoning.sql
      int_inzonechange.sql
      int_limitedheight.sql
      int_specialpurpose.sql
      int_validdtm.sql
      int_zoningdistricts.sql
      int_zoningmapindex.sql
      int_zoningtaxlots.sql
    ✓ product
      > qaqc
      zoningtaxlot_db.sql
    ✓ staging
      ! _staging_models.yml
      stg_dcp_commercialoverlay.sql
      stg_dcp_limitedheight.sql
      stg_dcp_specialpurpose.sql
      stg_dcp_zoningdistricts.sql
      stg_dcp_zoningmapamendments.sql
      stg_dcp_zoningmapindex.sql
      stg_dof_dtm.sql
      ! _sources.yml
    > seeds
```

## Staging -> Intermediate -> Product



# Quality Assurance

---



# Modularized Tests

Run tests against source, staging, intermediate, and product tables **as we build them** and **address discrepancies right away**.

```
1 of 6 START seed file hj_ztl_action.specialdistrict_priority ..... [RUN]
1 of 6 OK loaded seed file hj_ztl_action.specialdistrict_priority ..... [CREATE 10 in 0.45s]
2 of 6 START seed file hj_ztl_action.zonedist_priority ..... [RUN]
2 of 6 OK loaded seed file hj_ztl_action.zonedist_priority ..... [CREATE 166 in 0.21s]
3 of 6 START test not_null_zonedist_priority_priority ..... [RUN]
3 of 6 PASS not_null_zonedist_priority_priority ..... [PASS in 0.30s]
4 of 6 START test not_null_zonedist_priority_zonedist ..... [RUN]
4 of 6 PASS not_null_zonedist_priority_zonedist ..... [PASS in 0.19s]
5 of 6 START test unique_zonedist_priority_priority ..... [RUN]
5 of 6 PASS unique_zonedist_priority_priority ..... [PASS in 0.19s]
6 of 6 START test unique_zonedist_priority_zonedist ..... [RUN]
6 of 6 PASS unique_zonedist_priority_zonedist ..... [PASS in 0.19s]
```

## tables:

- name: dcp\_commercialoverlay
  - columns:
    - name: ogc\_fid
      - tests:
        - not\_null
        - unique
    - name: wkb\_geometry
      - tests:
        - not\_null
    - name: overlay
      - tests:
        - not\_null

# Modularized Tests

Source/Seed Tables

Staging Tables

Intermediate  
Tables

Product Tables

Tests:

- Not Null ✓
- Unique ✓
- ...

Tests:

- Not Null ✓
- Unique ✓
- ...

Tests:

- Not Null ✓
- Unique ✓
- ...

Tests:

- Not Null ✓
- Unique ✓
- ...

# Automatic Dependency Management

DBT automatically detects dependencies across queries (models) and determines the orders of execution, decreasing the chances of errors.

intermediate
! _int_models.yml
int__commercialoverlay.sql
int__inwoodrezoning.sql
int__inzonechange.sql
int__limitedheight.sql
int__specialpurpose.sql
int__validdtm.sql
int__zoningdistricts.sql
int__zoningmapindex.sql
int__zoningtaxlots.sql

# Macros and Jinja Templating

Macros are **analogous to Functions**.

Jinja Templates replace repetitive queries with, for example, **for-loops**.

Jinja and Macros could be used interactively to make queries more precise and free of errors.

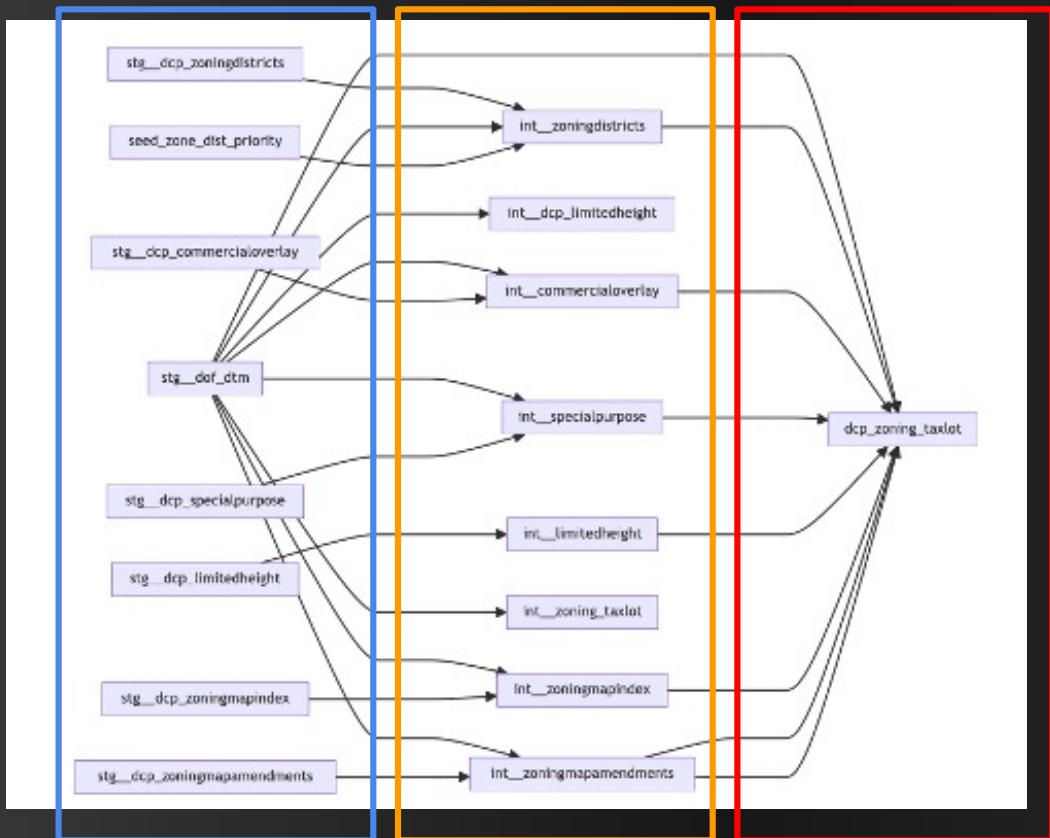
```
{% macro union_newvalues(left='new_version', right='prev_version', cols=[]) %}
    {% for col in cols %}
        SELECT
            '{{col}}' AS field,
            COUNT(*) AS count
        FROM {{left}} AS a
        INNER JOIN {{right}} AS b
        ON a.bbl=b.bbl
        WHERE a.{{col}} IS NOT NULL AND b.{{col}} IS NULL
        {% if not loop.last %}
            UNION ALL
        {% endif %}
    {% endfor %}
{% endmacro %}
```

# Takeaways

---

# DAG Before and After

Staging -> Intermediate -> Product



```
zoningtaxlots
  bash
  dbt_packages
  logs
  macros
  models
    intermediate
      ! _int_models.yml
      int_commercialoverlay.sql
      int_inwoodrezoning.sql
      int_inzonechange.sql
      int_limitedheight.sql
      int_specialpurpose.sql
      int_validdtm.sql
      int_zoningdistricts.sql
      int_zoningmapindex.sql
      int_zoningtaxlots.sql
    product
      qaqc
      zoningtaxlot_db.sql
    staging
      ! _staging_models.yml
      stg_dcp_commercialoverlay.sql
      stg_dcp_limitedheight.sql
      stg_dcp_specialpurpose.sql
      stg_dcp_zoningdistricts.sql
      stg_dcp_zoningmapamendments.sql
      stg_dcp_zoningmapindex.sql
      stg_dof_dtm.sql
    !_sources.yml
    seeds
```

**400**  
Lines of Code

Cleaner and more streamlined code

Less Code

# Insights

**Quality Assurance** is vital for data that serve the city and the public.

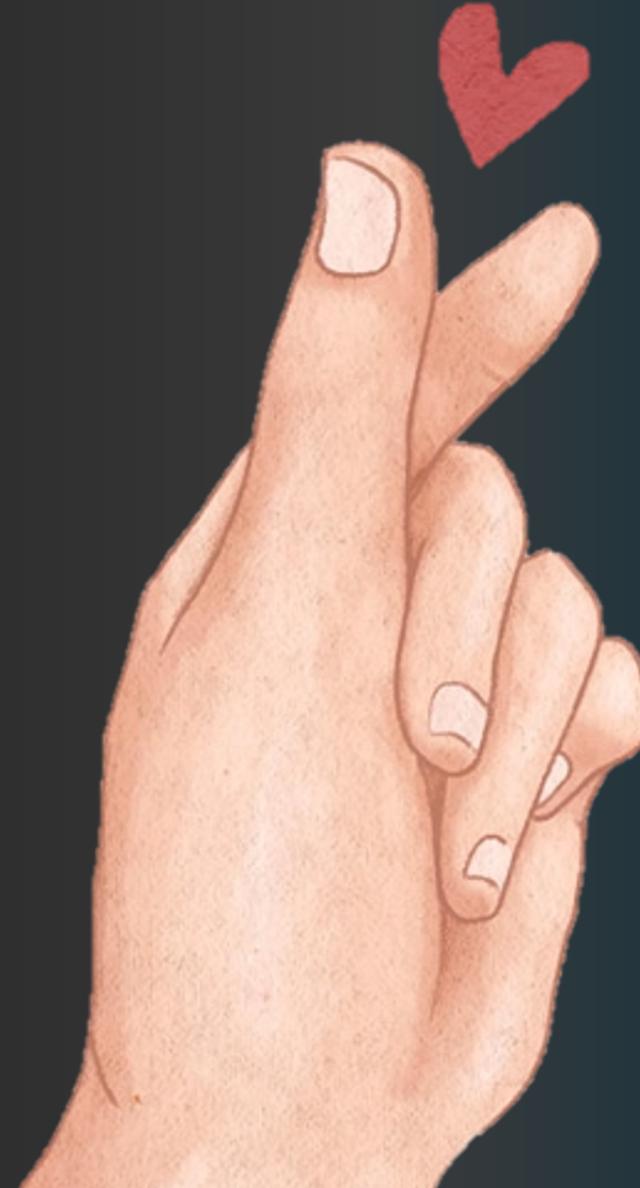
Also supported by **informative documentation and metadata**.

And it is a **long-term, continuous effort...**



# Special Thanks to DCP !!!

- GDE Senior Director **Amanda**
- My supervisor **Finn**
- **Sasha, Alex, and Damon** from the DE team



# Thank You



coding it forward >

NYC Department of City Planning, Data Engineering  
Heng Jiang  
Coding it Forward Fellow of 2024