Module 3 Lab 1: Data Lifecycle Design and Management

Estimated Time: 60–90 minutes
Risk Level: Moderate
Environment Stress: Low (dbt + Postgres only)

Introduction
In this lab, you will operate and analyze a complete data lifecycle using dbt. Rather than creating new models, you will run, observe, and reason about how data moves through lifecycle stages; from raw ingestion to analytics-ready outputs and how governance, quality, and access decisions apply at each stage. This lab emphasizes understanding and interpretation, not configuration or setup.

What You Have Already Done (Labs 1–3)
Before starting M3 Lab 1, you have already:

- Installed and validated PostgreSQL and dbt (M2 Lab 1)
- Explored raw data and documented governance classifications (M2 Lab 2)
- Built and validated dbt models across staging → core → marts (M2 Lab 3)

👉 ***M3 Lab 1 does NOT repeat these tasks.***
Instead, it focuses on lifecycle execution, lineage visualization, and decision documentation.

Learning Outcomes
By completing this lab, you will be able to:

1. Map the full data lifecycle: raw → staging → core → marts
2. Execute a controlled lifecycle build using dbt
3. Generate and interpret a dbt lineage (DAG) graph
4. Explain how lifecycle stages support:
    - Data quality
    - Governance
    - Access control
    - Analytics readiness
5. Document lifecycle decisions using a structured framework

Lab Concept: Data Lifecycle in Practice
A well-designed data lifecycle ensures that:

- Raw data is preserved and auditable
- Staging models standardize and clean data
- Core models enforce business meaning and integrity
- Marts serve specific operational or analytical use cases

dbt makes this lifecycle visible and testable, especially through its lineage graph.

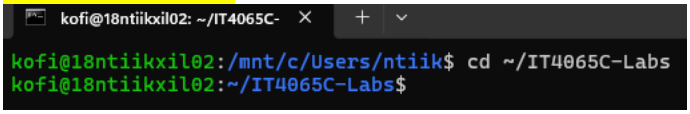
Before You Begin
Make sure that:

- You successfully completed M2 Labs 1–3
- You are working inside your cloned course repository
- You are using the same database credentials configured in Lab 1
- Your PostgreSQL service is running

⚠️ If M2 Lab 3 did not run successfully, fix that first before continuing.

Step 1: Run the Full Lifecycle Pipeline

1. Start your VM
2. Navigate your clone course repo root
   cd ~/IT4065C-Labs



---

3. From the root of the course repository, run
   bash labs/module_3/lab4/_run_all.sh

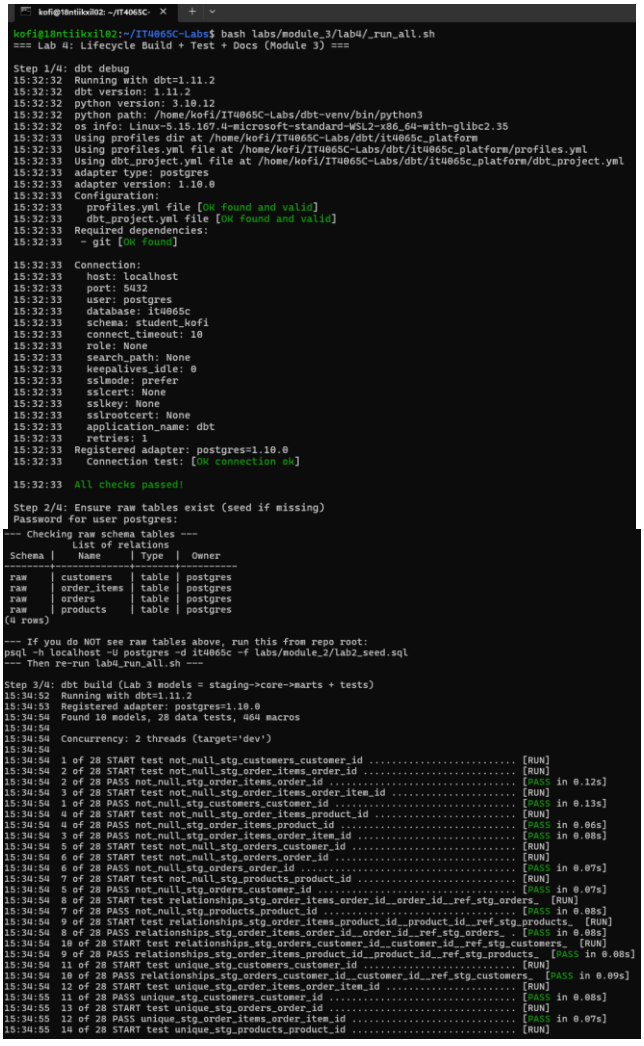
Use the same database credentials configured in M2 Lab 1 (do not change them).

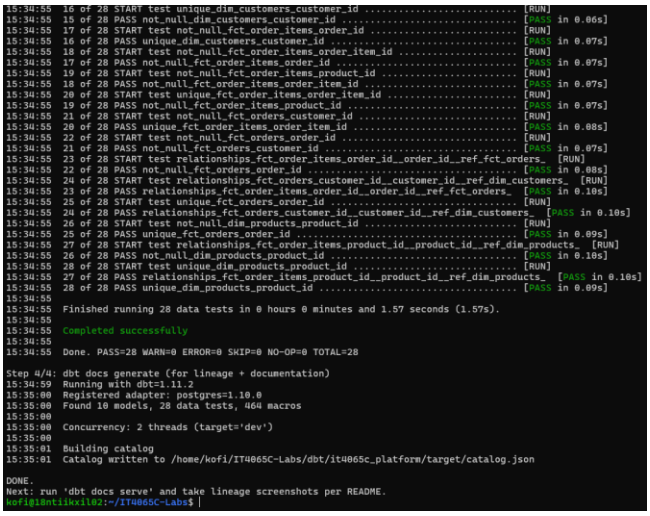
This script will:

1. Validate your dbt environment
2. Confirm raw data availability
3. Run all Lab 3 dbt models (**staging → core → marts**)
4. Execute dbt tests
5. Generate dbt documentation for lineage analysis

⏳ This may take a few minutes.

If the script completes without errors, you are ready to continue. A successful run shows display "Completed Successfully", see images below.



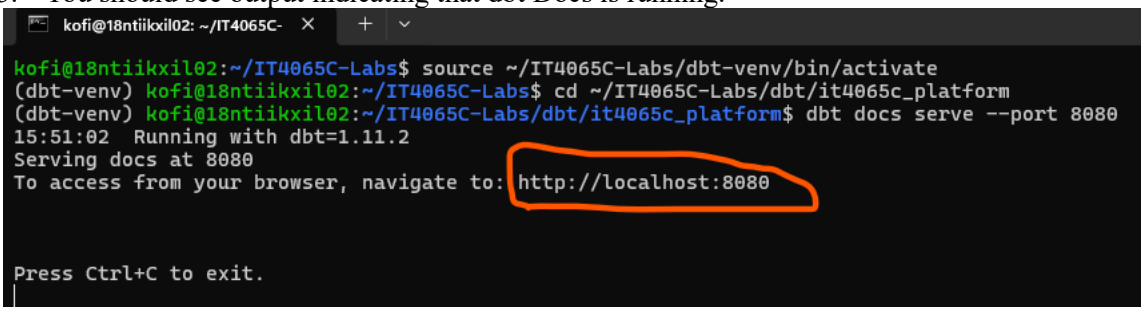


Step 2: Launch dbt Docs and View Lineage

In this step, you will launch dbt Docs and confirm that the data lineage graph is available.

Step 2.1: Start dbt Docs

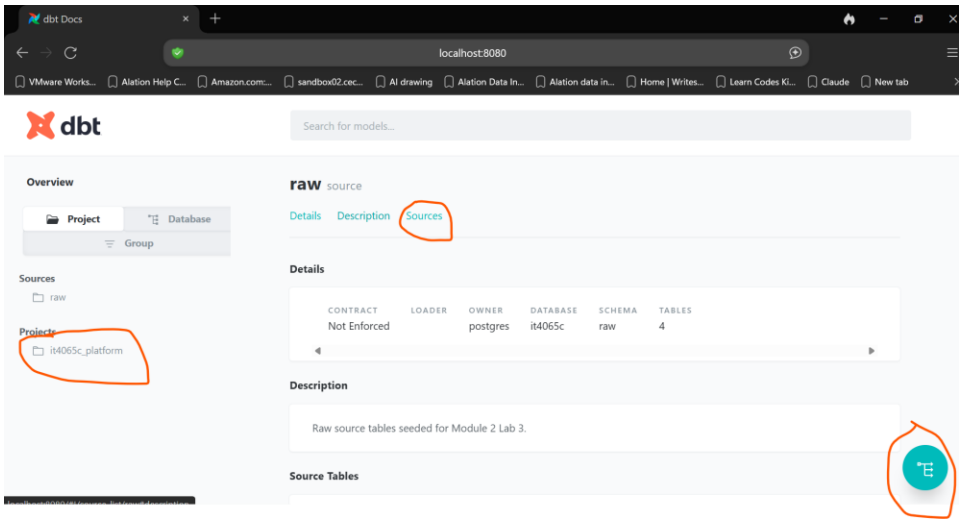
1. Navigate to the repository root directory.
2. Run the following commands in order:
   source ~/IT4065C-Labs/dbt-venv/bin/activate
   cd ~/IT4065C-Labs/dbt/it4065c_platform
   dbt docs serve --port 8080
3. You should see output indicating that dbt Docs is running.



---

4. Open the URL displayed in your browser (usually http://localhost:8080).
5. Leave this terminal window running while you work in the browser.

At this point, the dbt Docs interface should open successfully.



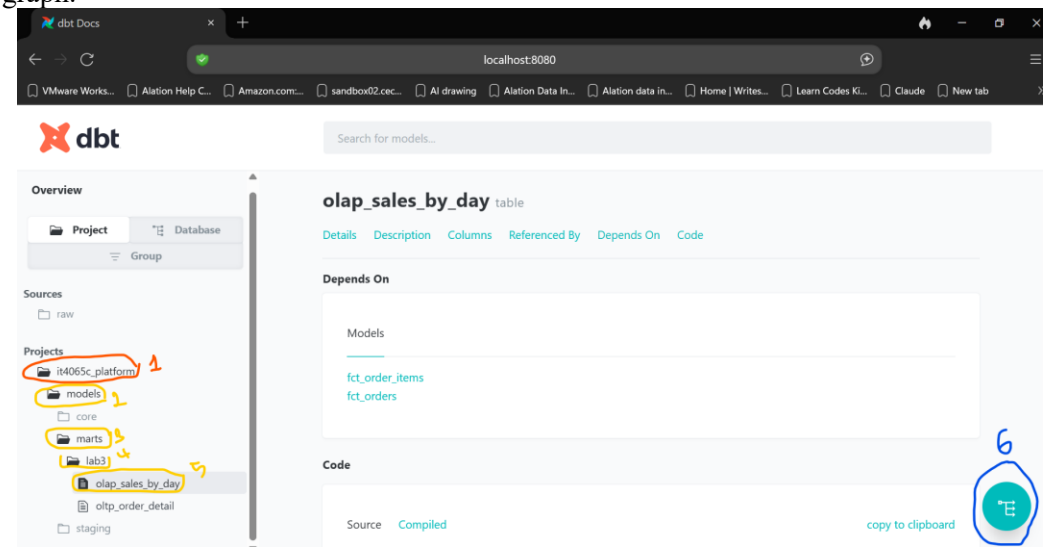
Step 3: Explore and Interpret the Data Lineage Graph

In this step, you will use the lineage graph (DAG) to understand how data flows through the lifecycle.

Step 3.1: Open the Lineage Graph

Inside the dbt Docs interface:

1. Click **Project** in the left sidebar.
2. Locate any model under models/marts/lab3 (for example, olap_sales_by_day).
3. From this model page, click the green graph icon in the bottom-right corner to open the lineage graph.



You should now see a lineage graph showing data flowing from *raw* → *staging* → *core* → *marts*.

This initial lineage view shows a model's direct dependencies only. You will explore upstream lifecycle layers in the next steps.

Step 3.2: Verify Mart Dependencies Using the Lineage Graph

1. In dbt Docs, open the model oltp_order_detail.
2. Click the lineage graph icon (bottom-right).
3. Observe how the model depends on:
    - Core fact and dimension models
    - Upstream staging models
    - Raw source tables

📸 **Screenshot 1:** Lifecycle via oltp_order_detail
Must show:

- oltp_order_detail
- At least one upstream fact or dimension model
- At least one upstream staging model
- At least one upstream raw source

This screenshot demonstrates a complete lifecycle path.

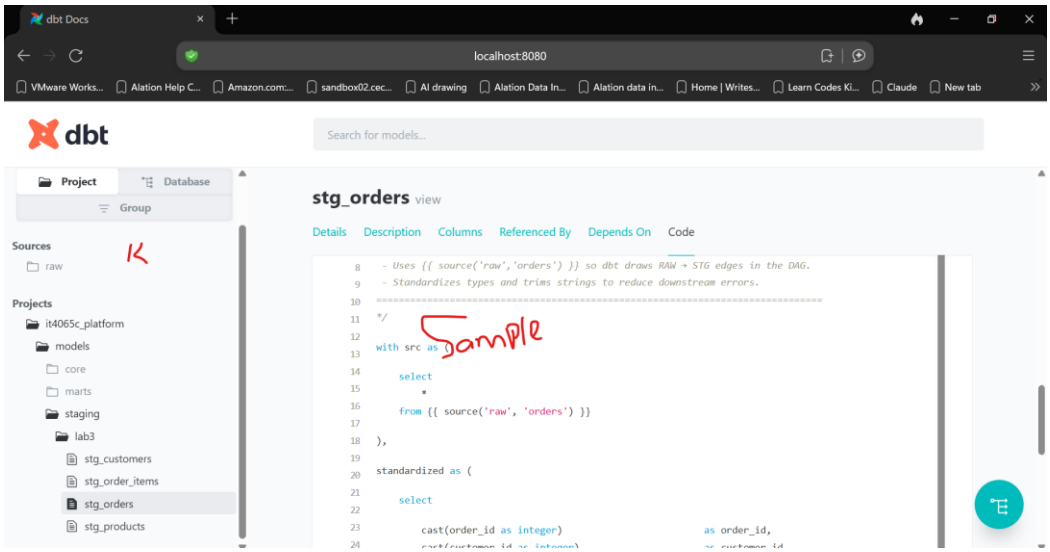
Step 3.3: Confirm Raw → Staging Relationship in Code

1. In dbt Docs, open the it4065c_platform/models/staging/lab3/stg_orders.
2. Click Code → Source.
3. Locate the source reference:

from {{ source('raw', 'orders') }}

📸 **Screenshot 2:** Source verification ({{ source() }})

- Must show:
    - stg_orders
    - Code → Source view
    - The {{ source('raw', 'orders') }} reference



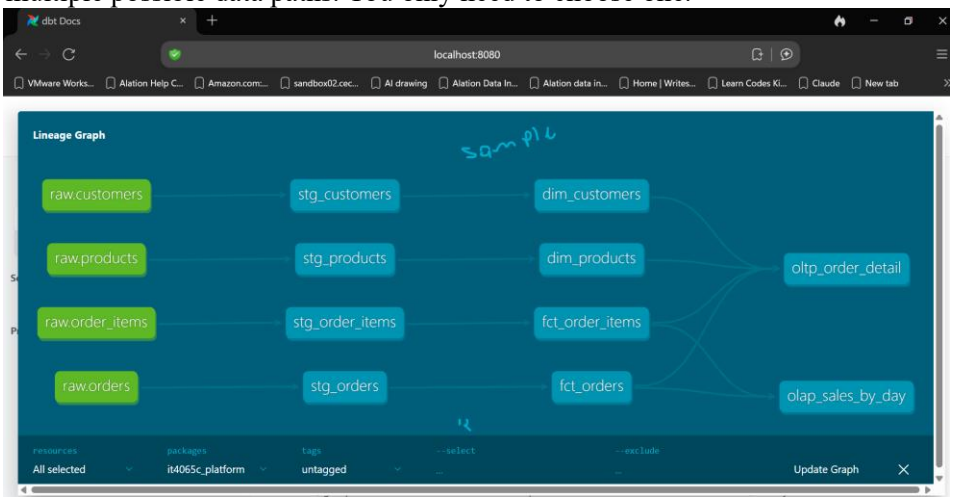
Step 3.4: Write the Data Lifecycle Chain (Short Answer)

In this step, you will use the full lineage graph in dbt Docs to demonstrate your understanding of how data moves through the system.

1. In dbt Docs, click it4065c_platform under project in the left sidebar.
2. Click the lineage graph icon (bottom-right corner) to open the full lineage graph.
3. Observe how data flows from raw sources, through staging and core models, and into mart models.

Screenshot 3: Full project lineage graph

You will see multiple possible data paths. You only need to choose one.



Written Submission

Using the lineage graph, write one complete lifecycle chain that shows how data flows from a raw source to a mart model.

Your lifecycle chain must:

- Start with a raw.* table
- Include one stg_* model
- Include one core model (fct_* or dim_*)
- End with a mart model (olap_* or oltp_*)

Use exact model names as shown in dbt Docs, and use arrows (→) to show data flow.

Example (One Valid Answer)
raw.orders
→ stg_orders
→ fct_orders
→ olap_sales_by_day
Note: You do not need to describe the entire graph.
One correct lifecycle path is sufficient.

Why This Matters
Being able to identify and describe a complete lifecycle chain demonstrates that you understand:

- Where data originates
- How it is transformed
- Where business logic is applied
- Which models are safe for analytics and reporting

This skill is essential for data governance and analytics engineering.