Mastering DBT
Your Complete Training Journey Begins Here

Foundational Skills for dbt Mastery

Command Line Basics

Necessary for navigating and executing commands in the terminal.

Version Control

Helpful for managing and tracking changes in dbt projects.

Data Warehouse Understanding

Crucial for comprehending data storage and processing.

VS Code

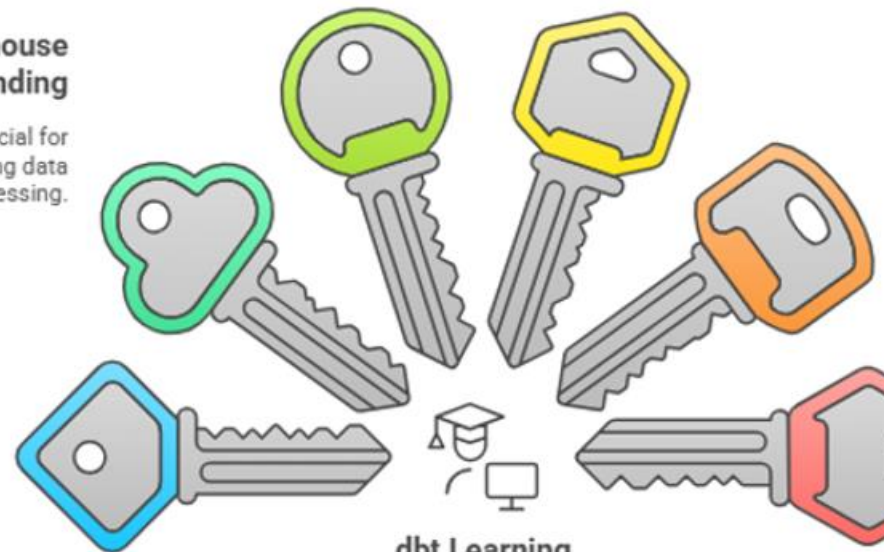
A powerful editor for writing and managing dbt code.

SQL Knowledge

Essential for querying and manipulating data in dbt.

Snowflake

A cloud-based platform for data storage and processing.



dbt Learning

What is dbt ?

- **DATA BUILD TOOL, COMMONLY KNOWN AS DBT**
- Data Build Tool (dbt) is an open-source analytics engineering tool
- Focuses on the 'Transform' step in ELT (Extract, Load, Transform) processes
- Founded in 2016 by Fishtown Analytics (later rebranded as dbt Labs)
- It enables data teams to transform raw data into clean, analytics-ready datasets directly within the data warehouse using SQL.
- DBT has quickly become a widely adopted tool in modern data pipeline development
- Unlike traditional ETL tools, DBT performs data transformations directly within the data warehouse using SQL.

Why Use dbt?

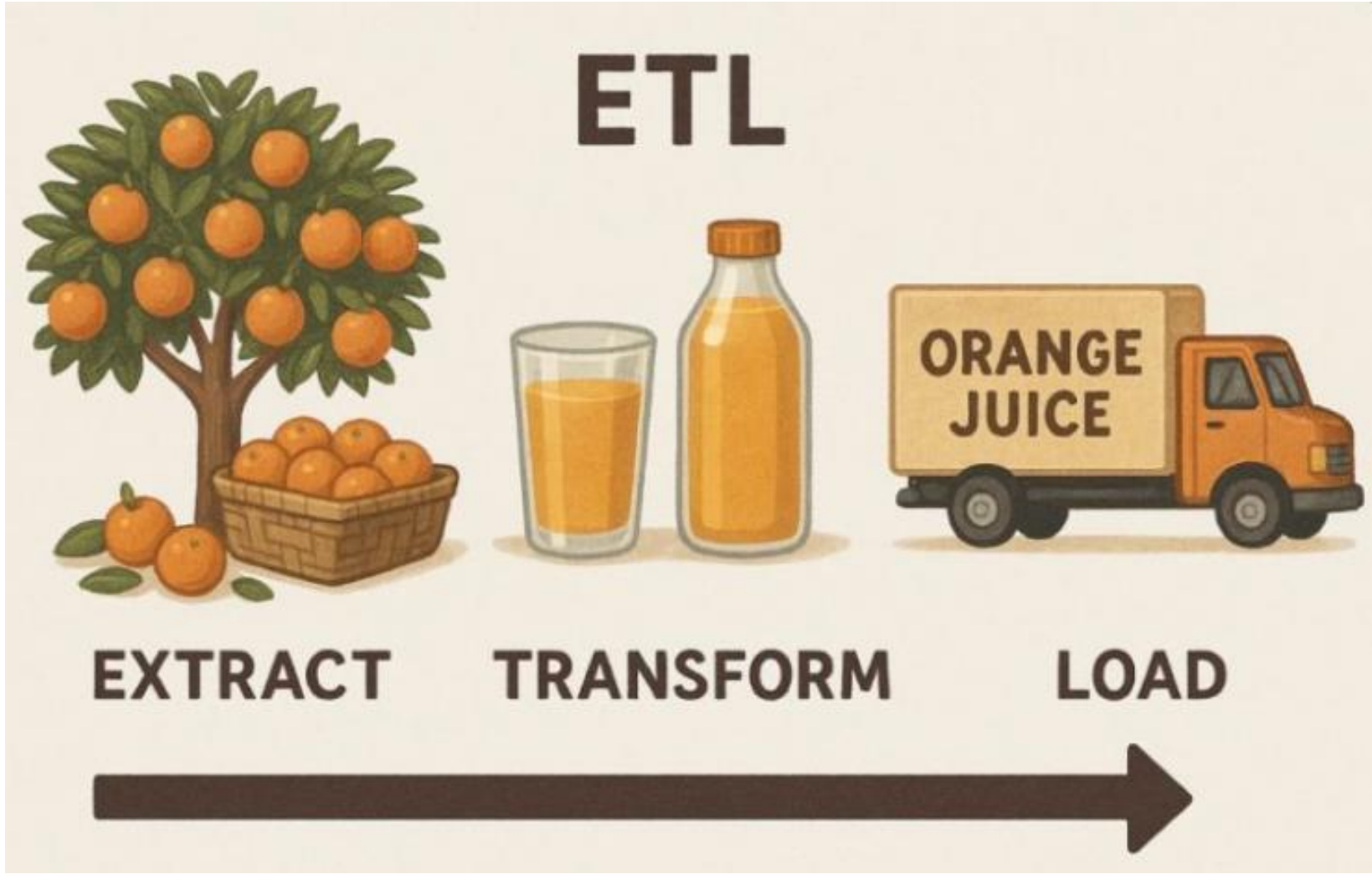
🔁 **Version Control :**Tracks all changes with Git-like “undo” and “history” for your data code.

🧩 **Modularity :** Breaks big SQL queries into small, reusable models—easy to build and manage.

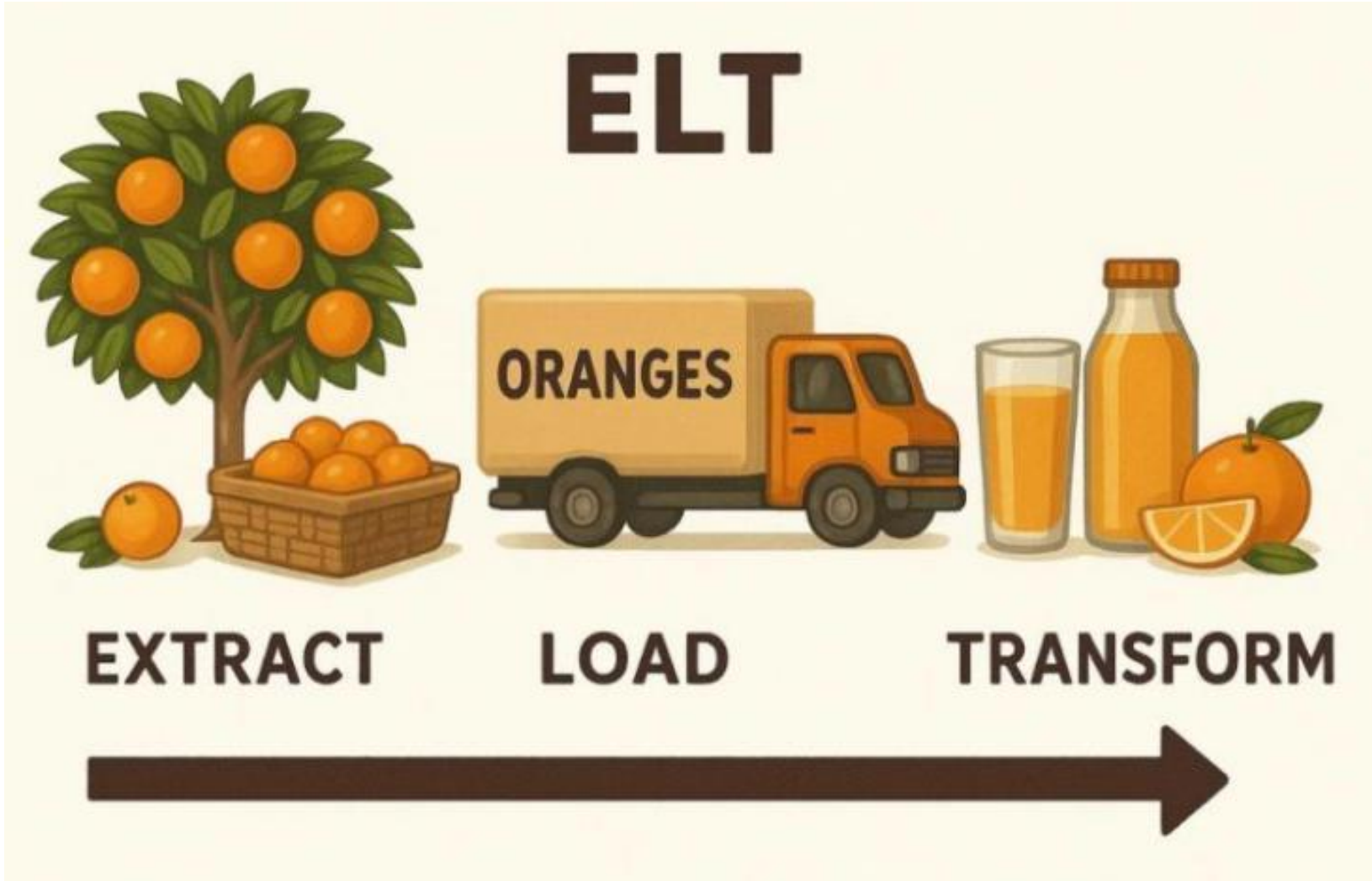
📚 **Documentation :**Creates clear, automatic docs—so everyone understands what’s happening.

✅ **Testing :**Built-in checks make sure your data is clean and correct.

⏱️ **Orchestration :**Run your dbt jobs on a schedule using dbt Cloud or tools like Airflow.


ETL
ORANGE JUICE
EXTRACT
TRANSFORM
LOAD

dbt



ETL vs ELT { what is the real difference ❓ }

Basic definition :
ETL : Extract data from main source, Transform it in a staging location, Load it in a warehouse
ELT : Extract data from main source, Load raw data into warehouse, transform using computing power of the warehouse.

📌 ETL use cases :
- Legacy systems
- On-premise databases
- Systems with limited transformation capabilities

📦 ETL tools : Informatica, Talend, Apache NiFi, SSIS

📌 ELT use cases :
- Cloud-native architectures, leverages warehouse scalability
- Big data platforms (Snowflake, Redshift, BigQuery). Easier to manage raw + transformed versions together

📦 ELT tools : dbt, fivetran, stitch

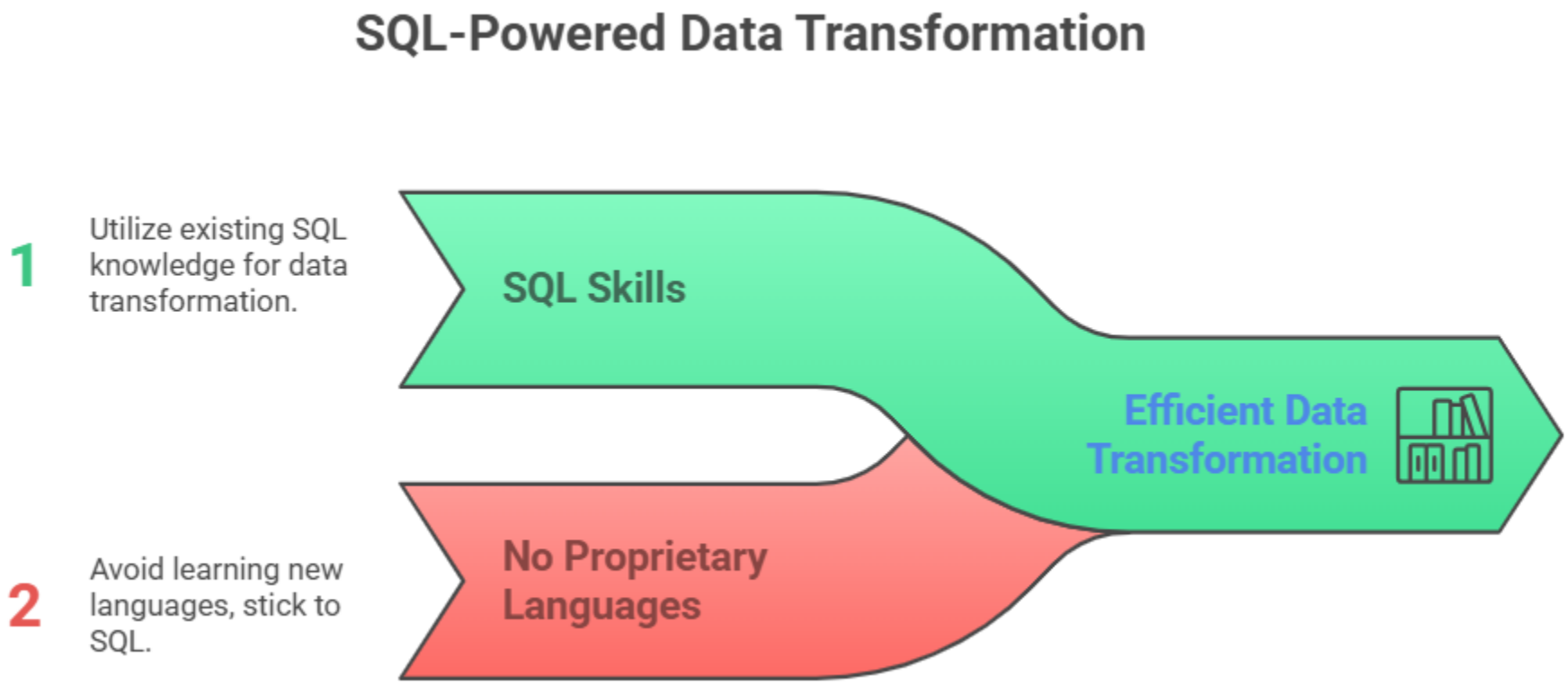
Advantages of dbt :

- **Uses SQL** : You can write transformations using SQL, which many analysts and data engineers already know.
- **Easy to Manage Code** : You can break big tasks into smaller parts (models), making your work easier to understand and reuse.
- **Works with Git** : dbt connects with Git, so teams can work together and keep track of changes.
- **Built-in Testing** : It has tools to check if your data is correct, helping you catch mistakes early.
- **Creates Documentation Automatically**: dbt writes useful documentation for you, so others can easily understand your data models.
- **Shows Data Flow Visually** : It draws diagrams to show how your data moves and connects, which helps when fixing issues or planning changes.
- **Strong Community Support**: Many people use and support dbt, so you can find lots of help and examples online.
- **Keeps History with Snapshots**: It can save older versions of your data so you can see how it changed over time.
- **Supports Seed Files**: You can add small or rarely changing data from CSV files into your models easily.
- **Manages Dependencies**: dbt knows the order to run your models so everything works correctly.

Key Advantages of dbt (Data Build Tool)

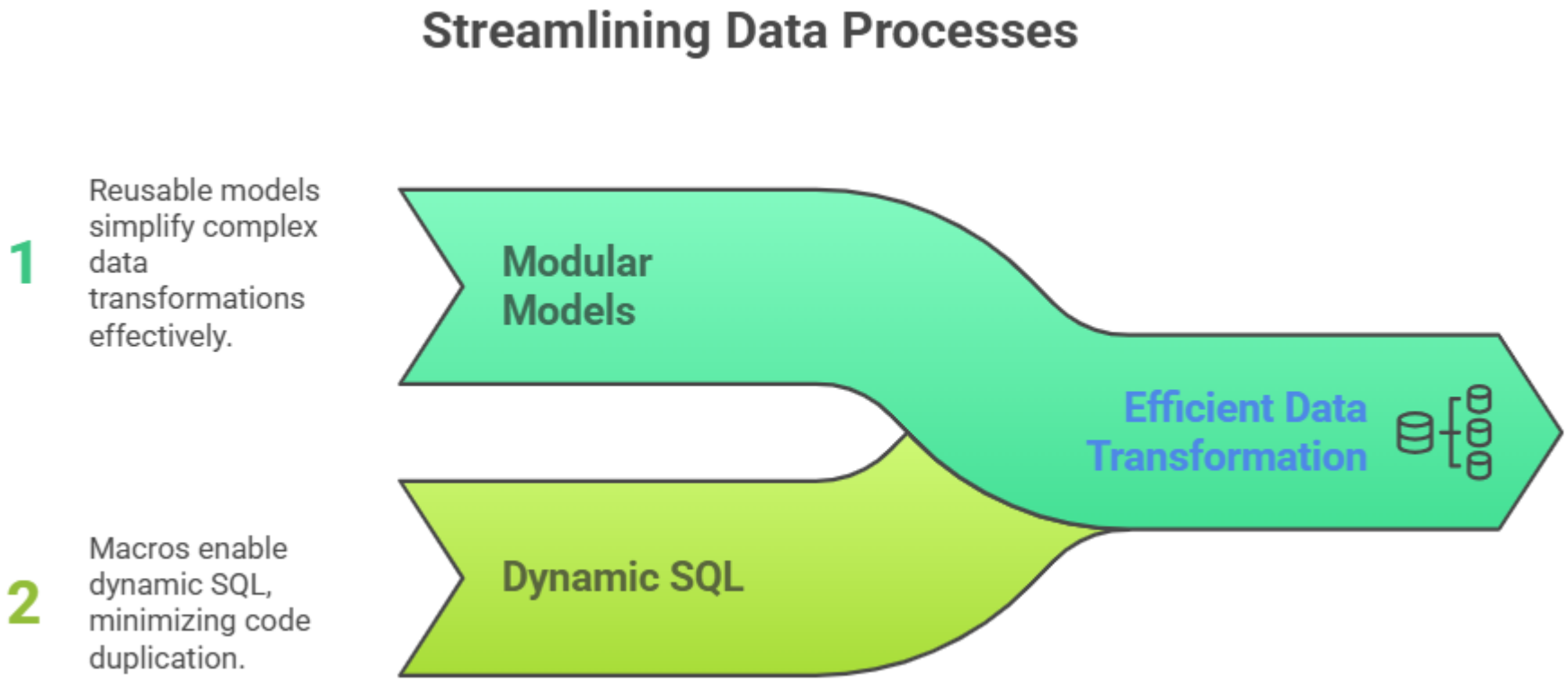
1.SQL-Based Transformations

- Uses standard SQL syntax, making it accessible to data analysts and engineers.
- No need to learn proprietary languages—leverage existing SQL skills.



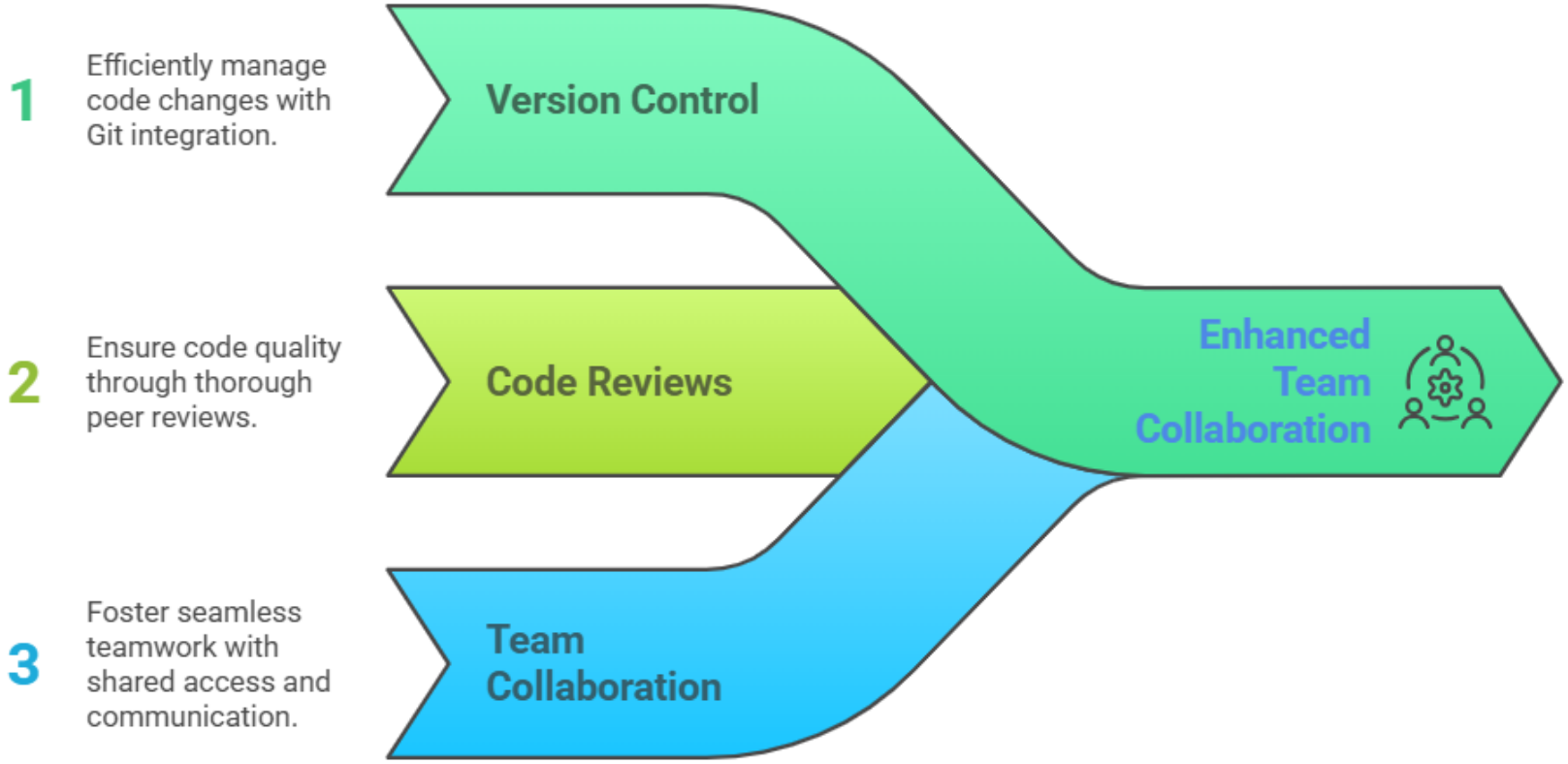
2.Modular & Reusable Code

- Break complex transformations into **small, reusable models** (like functions in programming).
- Supports **macros** (Jinja templating) for dynamic SQL, reducing redundancy.



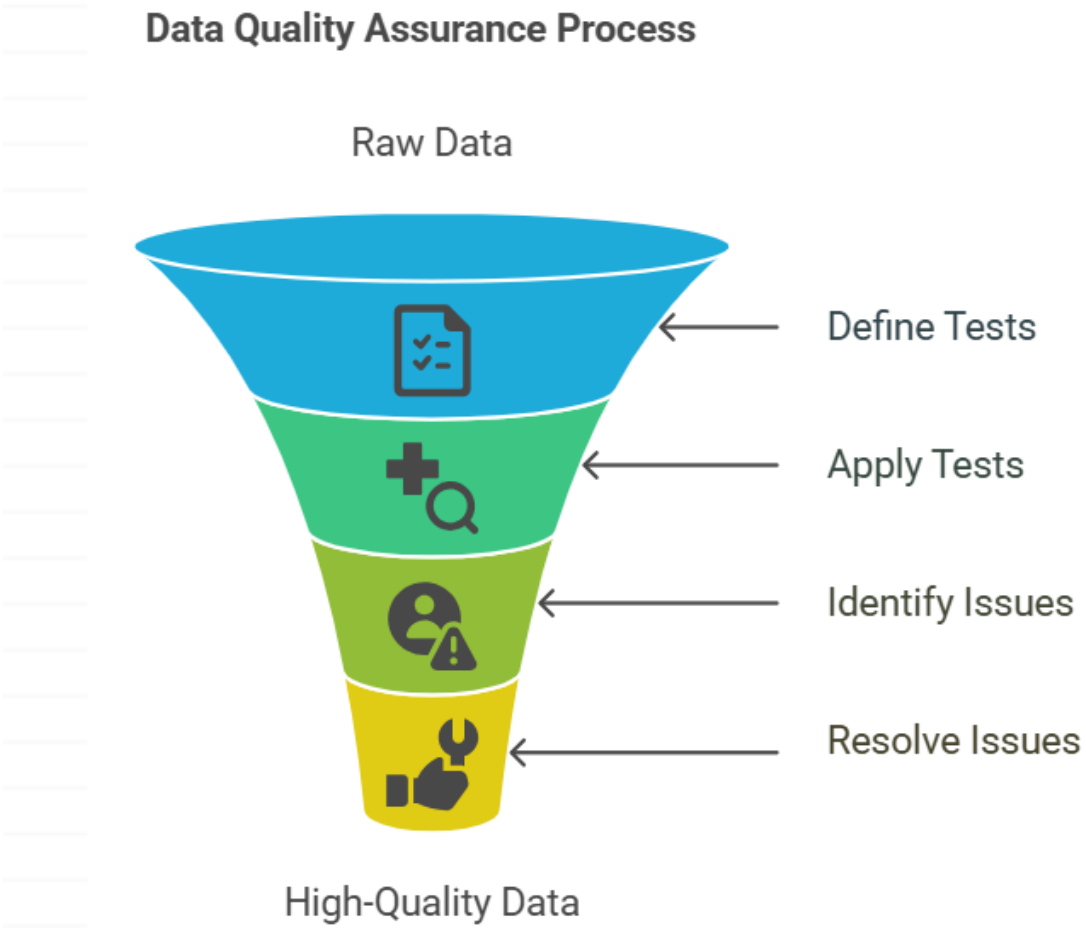
3. Version Control & Collaboration

- Native Git integration for tracking changes, code reviews, and team collaboration.
- Enables CI/CD pipelines for automated testing and deployment.



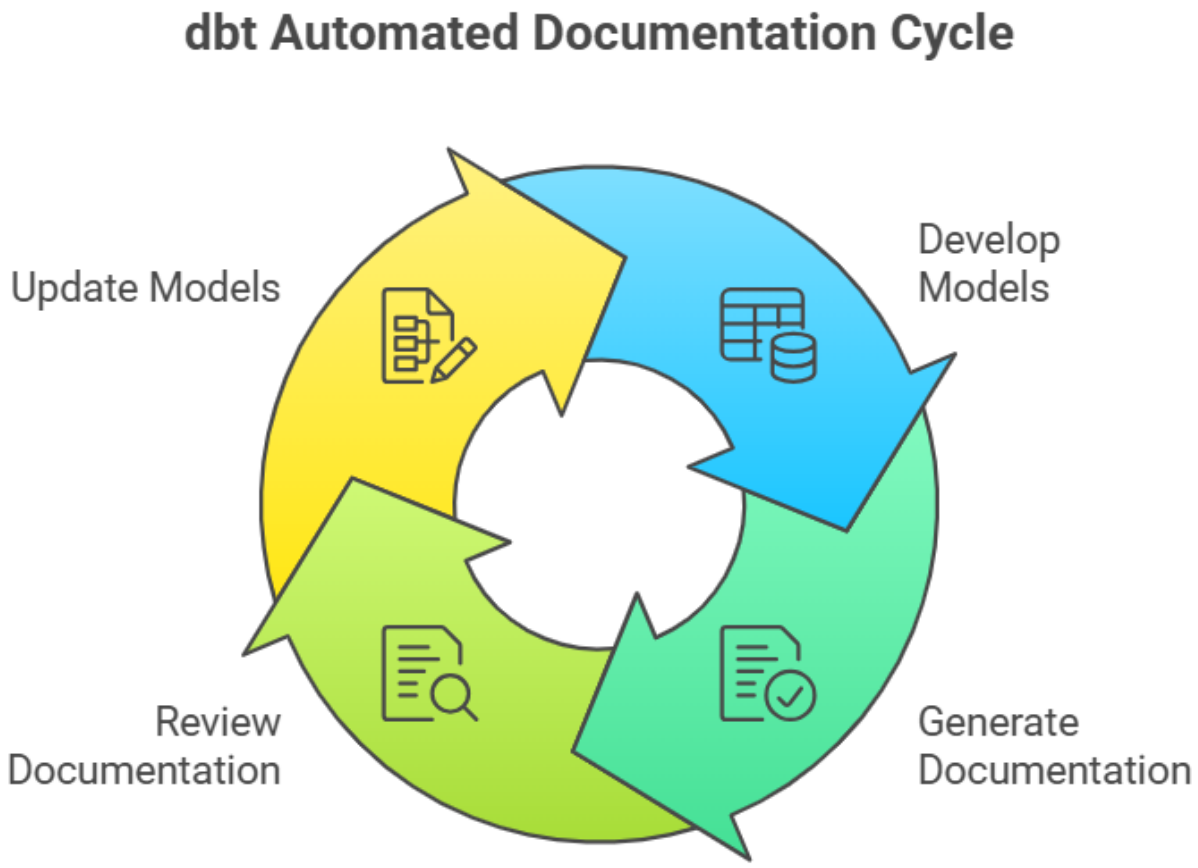
4. Built-In Data Testing

- Define data quality tests (e.g., uniqueness, null checks) directly in models.
- Catch issues early with automated validation.



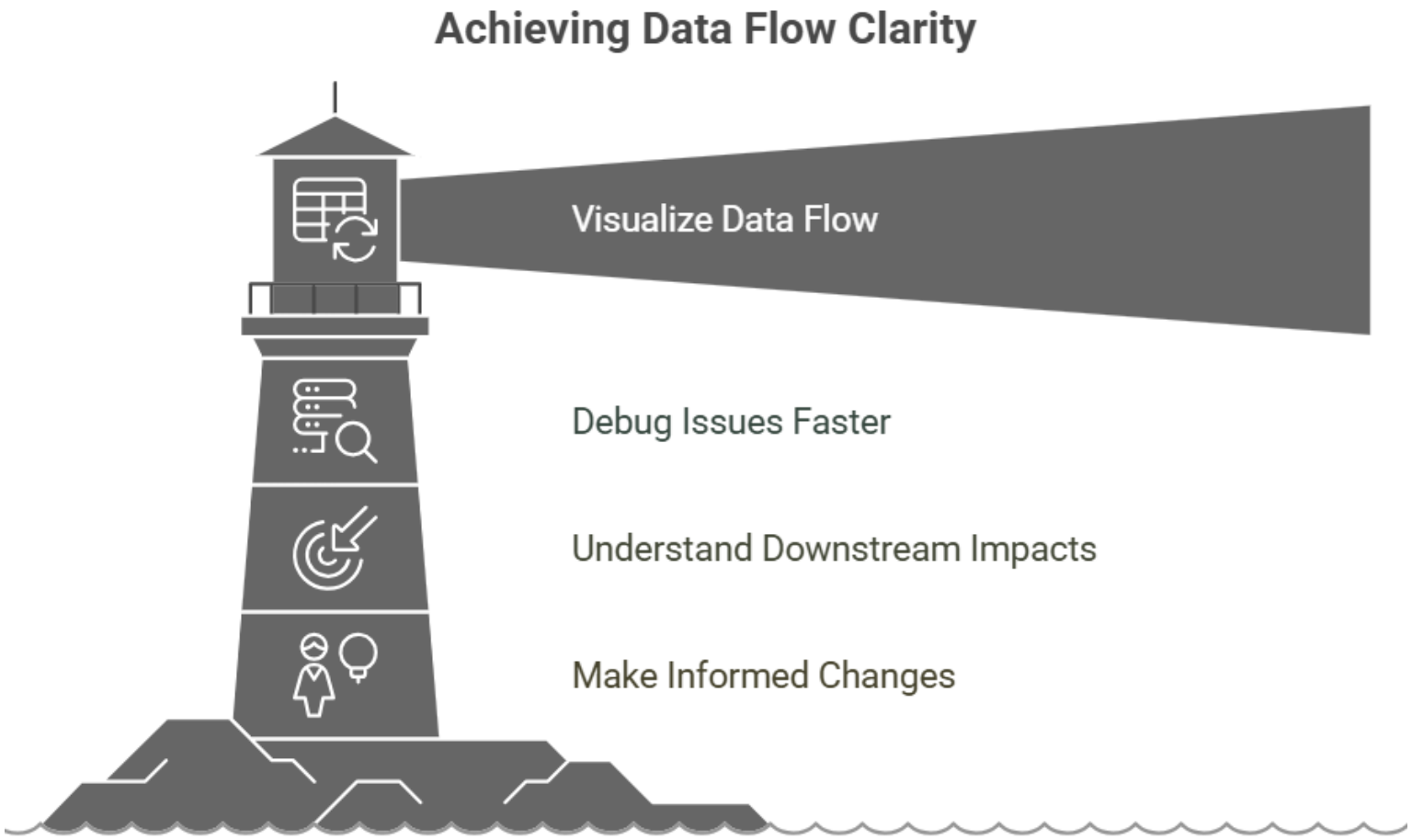
5. Automated Documentation

- Self-documenting models with auto-generated docs.
- Centralized metadata for tables, columns, and dependencies.



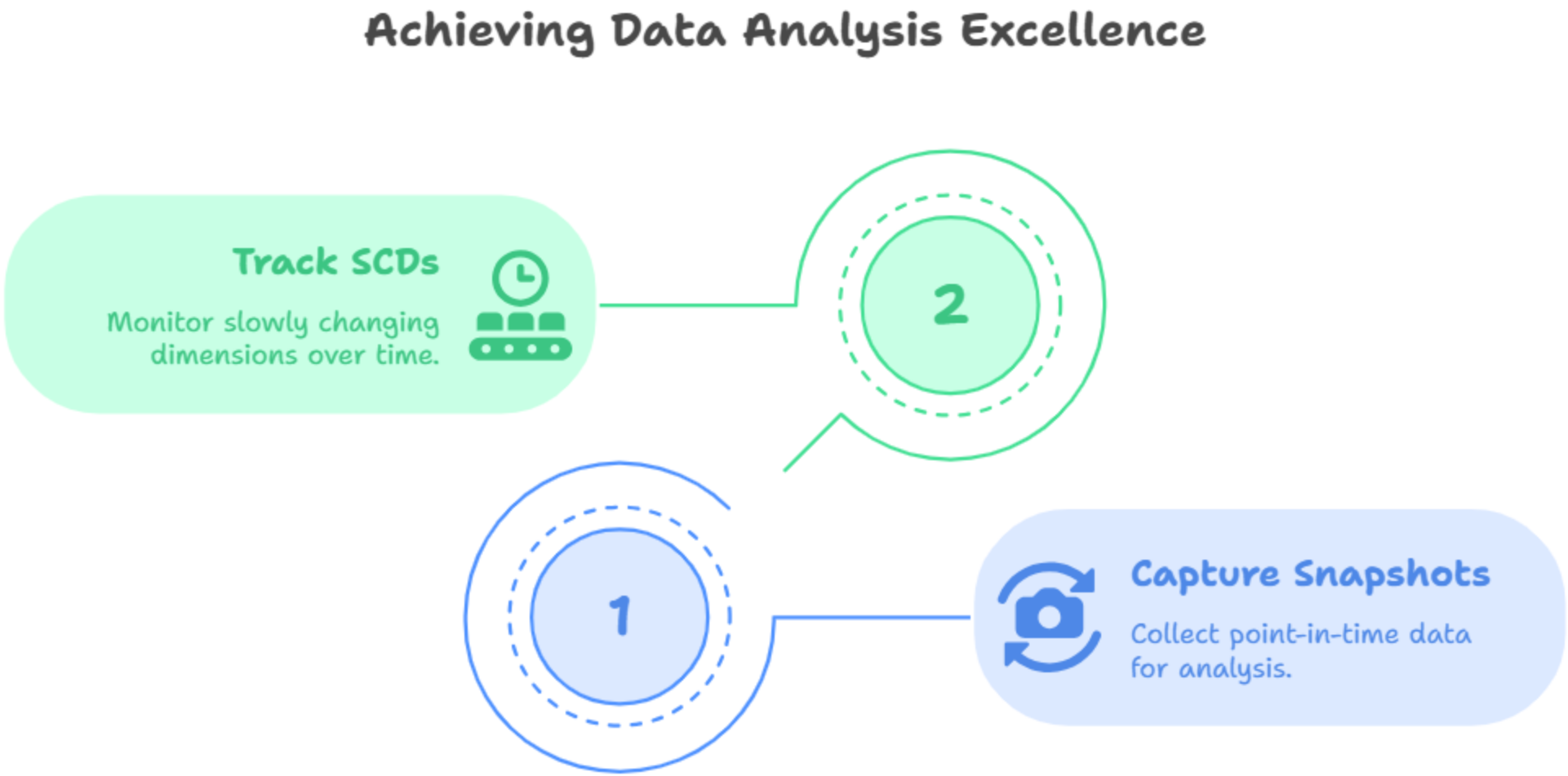
6. Data Lineage & Impact Analysis

- Visualize data flow across models to debug issues faster.
- Understand downstream impacts before making changes



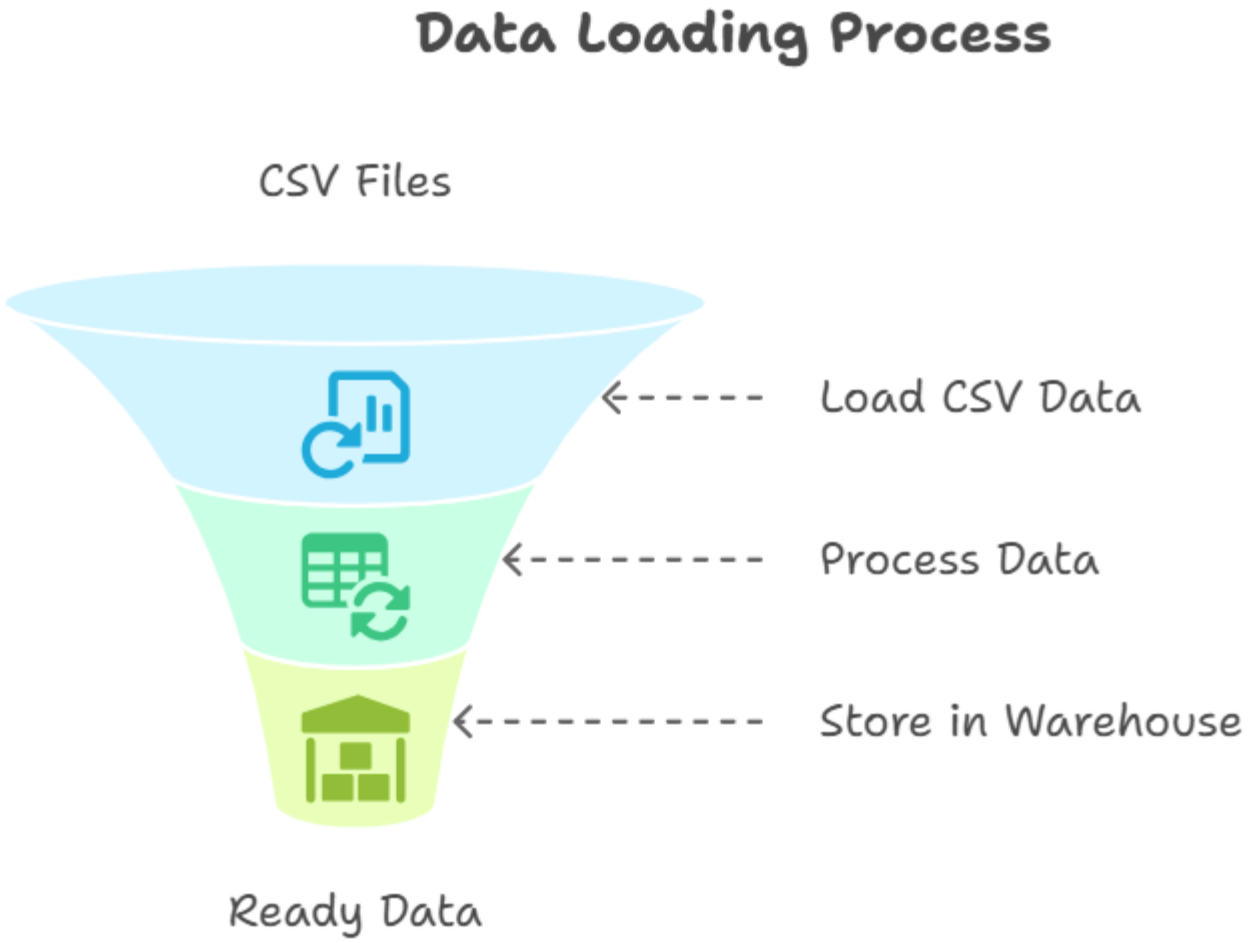
7.Historical Data Tracking (Snapshots)

- Capture point-in-time data with snapshots for trend analysis.
- Track slowly changing dimensions (SCDs) effortlessly.



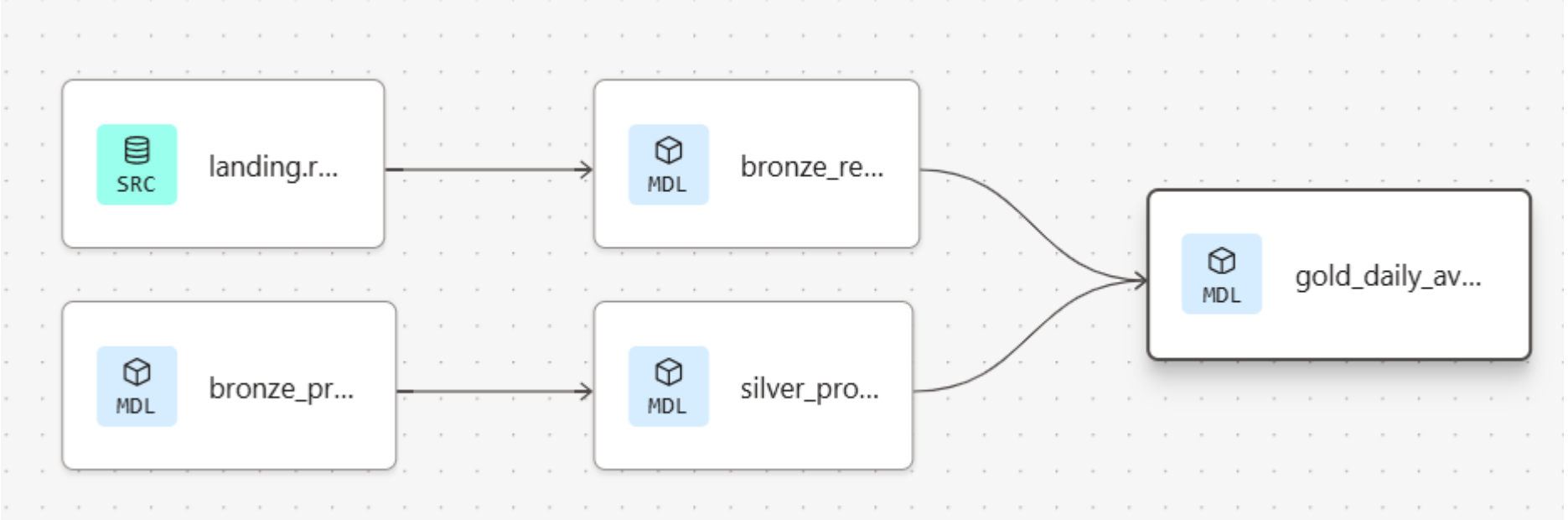
8.Seed Files for Static Data

- Easily load CSV files (e.g., country codes, mapping tables) into your data warehouse.



9. Smart Dependency Management

- dbt automatically resolves and executes models in the **correct order**.
- Parallel execution for faster processing.



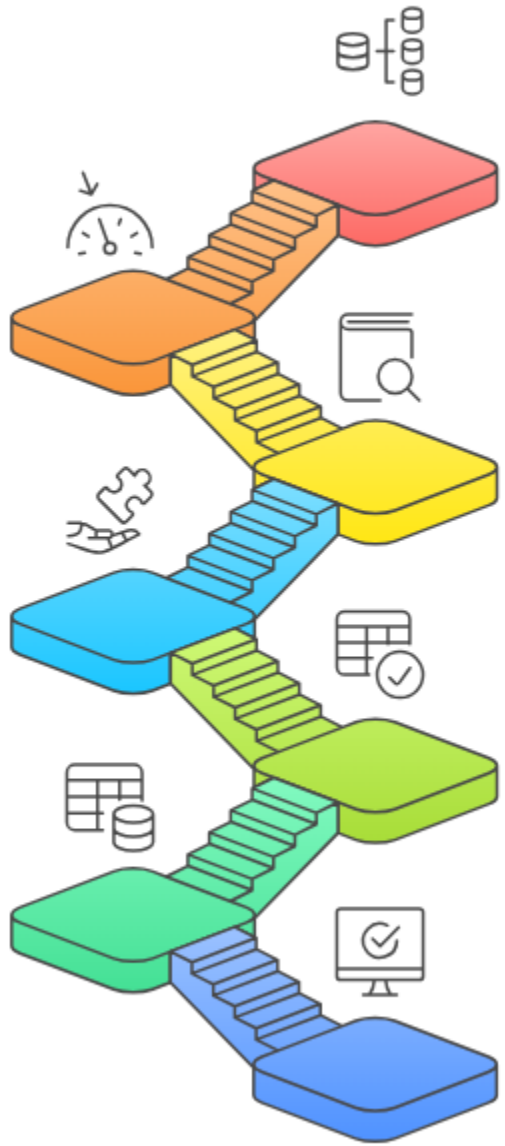
10. Strong Open-Source Community

- Active community support.
- Extensive library of third-party packages (dbt Hub).

Why Teams Choose dbt

- ✅ **Faster development** (SQL + modular design)
- ✅ **Better data governance** (testing, docs, lineage)
- ✅ **Scalable transformations** (reusable logic)
- ✅ **Works with modern data stacks** (Snowflake, Big Query, Redshift, etc.)

Mastering dbt for Scalable Data Projects



7
dbt Scalability
Structure dbt projects to scale with the company

6
Performance Optimization
Improve performance and reduce costs in your warehouse

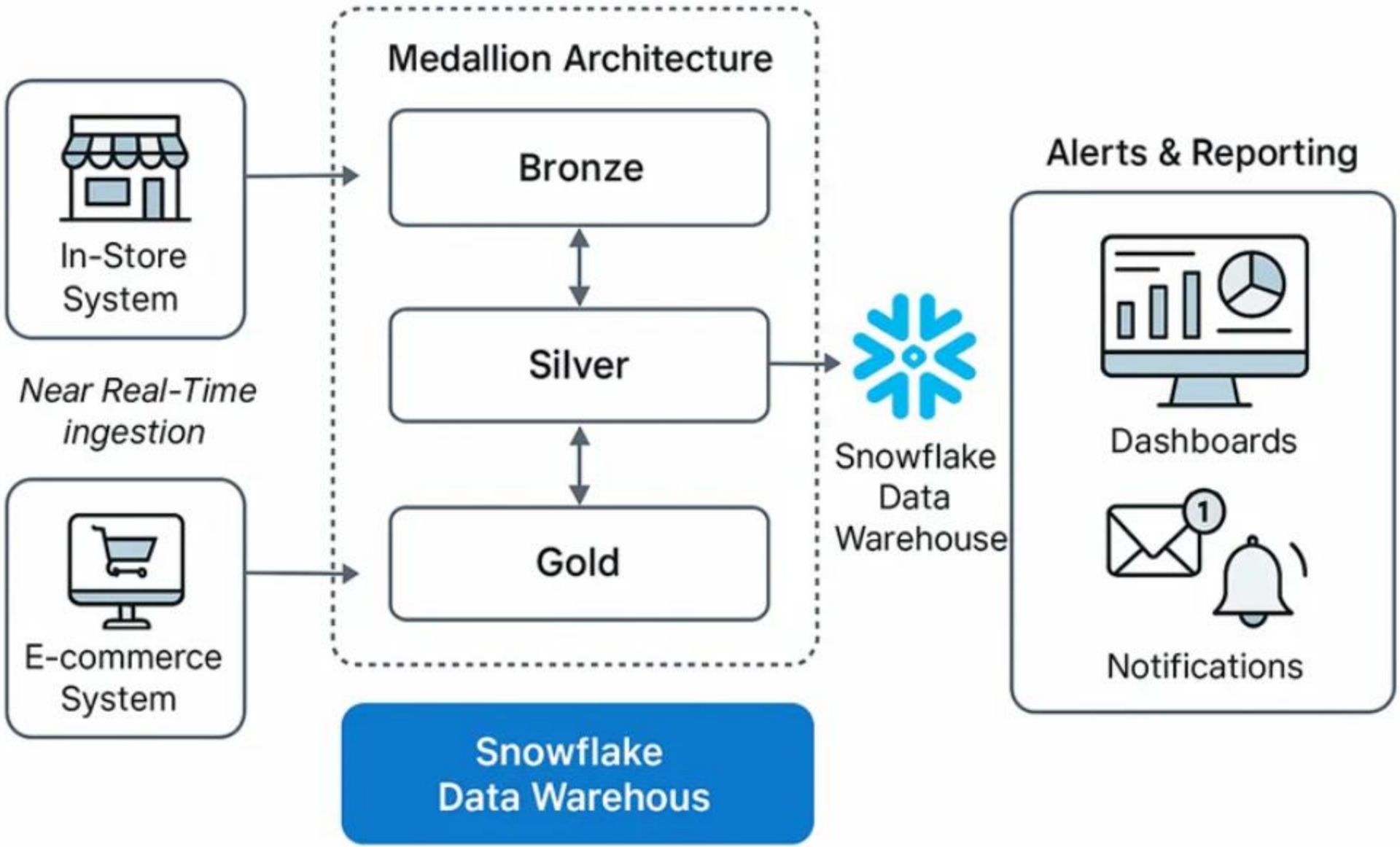
5
Best Practices
Learn best practices for scalable dbt projects

4
Macros & Reuse
Create and reuse code using macros and variables

3
Data Testing
Implement data tests and freshness checks

2
Data Modeling
Build efficient SQL models and understand materializations

1
Installation & Setup
Learn to install and configure dbt core


Medallion Architecture
In-Store System
Near Real-Time ingestion
E-commerce System
Bronze
Silver
Gold
Snowflake Data Warehouse
Alerts & Reporting
Dashboards
Notifications
Snowflake Data Warehous

	Snowflake	dbt
♻️ Modularity & Reusability	✅ Yes – using UDFs and Stored Procedures (preferred for developers)	✅ Supports Jinja macros, modular SQL models, and reusable packages
🧪 Automated Testing & Data Quality	❌ Requires manual validation scripting and no built-in test framework	✅ Built-in tests (unique, not null, relationships) and support for custom logic
📘 Documentation & Data Lineage	❌ Manual documentation; lineage is unclear across many SQL objects	✅ Auto-generated documentation and DAG-based lineage available in dbt Cloud and Core
⏰ Data Freshness Monitoring	❌ No native freshness check on views or tables	✅ Sources can be monitored for freshness with built-in alerting
⚙️ Change Management & CI/CD	❌ Requires custom DevOps scripts for version control, testing, and deployment.	✅ Provides native Git integration, environment management, and built-in CI/CD pipelines.

Thanks