Fivetran + dbt Labs

dbt State

Build what's changed, skip what hasn't.

What is dbt State?

Only build when data or code has changed with dbt State. Your dbt project checks warehouse metadata and model SQL to determine any downstream changes and builds, skips, clones, or defers each run accordingly. In production, dbt State leads to lower warehouse compute on average and prevents breakage of your model runs. In development, analytics engineers are able to iterate faster with minimal compute cost. Whether you're orchestrating in the dbt platform, running dbt Core locally, or using an external orchestrator, dbt State optimizes every model run.

Immediate optimization and results

- Improves your team's workflows and speeds up performance
- Works every time a dbt model is executed
- Ensures you only run the parts of your model that actually need to run on changed data

Automatic state awareness

Checks warehouse metadata and model SQL to determine if results would change, and decides whether to:

- **run** the node
- **skip** it
- **clone** existing state
- **auto-defer** to production when appropriate

Lower warehouse compute costs

- Runs jobs faster for less uptime on your warehouse
- Prevents runaway dbt compute costs as data volumes and project sizes increase
- Optimizes expensive jobs and only runs when changes are detected

Why dbt State?

Managing your dbt estate can be challenging to scale as your project grows. Every dbt run gets more expensive, slower, and harder to reason through. dbt State brings state awareness to both dbt Core and the dbt platform. It allows users to optimize their workflows without manual workarounds such as adding scripts or process orchestration to their dbt project – most approaches are brittle, incomplete, and require constant maintenance. dbt State solves these issues with out-of-the-box state management, improving speed and lowering costs.

Learn more at www.getdbt.com/product/dbt-state

dbt State is compatible with dbt Core or natively within the dbt platform

QUICKSTART

dbt Core

- Python versions: 3.10-3.13
- dbt versions: 1.7 - 2.0
- Warehouses: Snowflake, Databricks, BigQuery
- pip

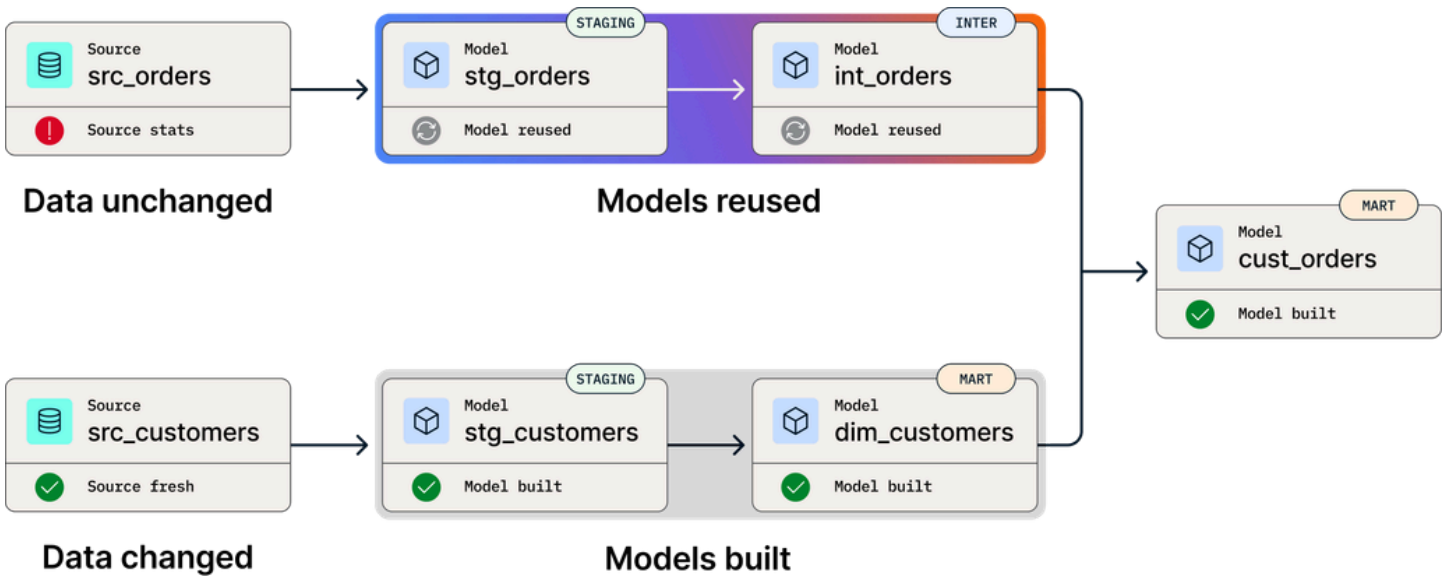
- Install dbt State
  - On Core 1.7-1.11: run pip install dbt-state and the first dbt run or dbt build will prompt you create a standalone account to start a trial
  - On 1.12-2.0 (including Fusion): run dbt login to create a standalone account to start a trial
- dbt State is now enabled on every dbt run or dbt build
- User-level configuration is set in ~/.dbt/user_settings.yml

dbt platform

- Account settings, set up dbt State

"Fusion and dbt State have changed what our team spends its time on. Declaring SLAs at the model level and letting dbt State decide what actually needs to run has already delivered 25–30% model reuse and roughly 15% in Snowflake savings on our pilot project. It's the first orchestration layer that's let us stop managing runs and start managing outcomes."

`Alvin Chai, Fanatics - Senior Analytics Engineer`



Learn more at www.getdbt.com/product/dbt-state