dbt Labs

# dbt @ Cinco de Trino

5 May 2022

# Jeremy Cohen

*he/him*

Product Manager, dbt Core

Head of Office,
dbt Labs - Marseille

**@jerco** in dbt Community Slack

# Goals of this talk

Answer these questions:

- What is dbt?
- What do we mean by “data transformation”?
- Why are modularity + testing so important?

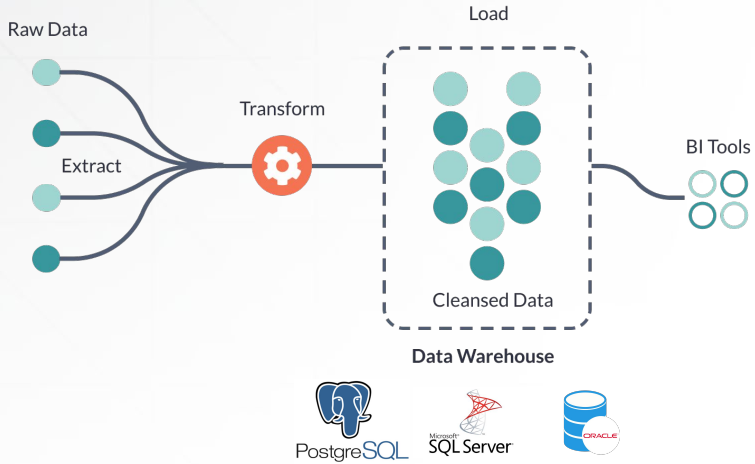
Demo dbt + Trino (Galaxy):

- Most important features
- Using TPC-H data

# How we got here

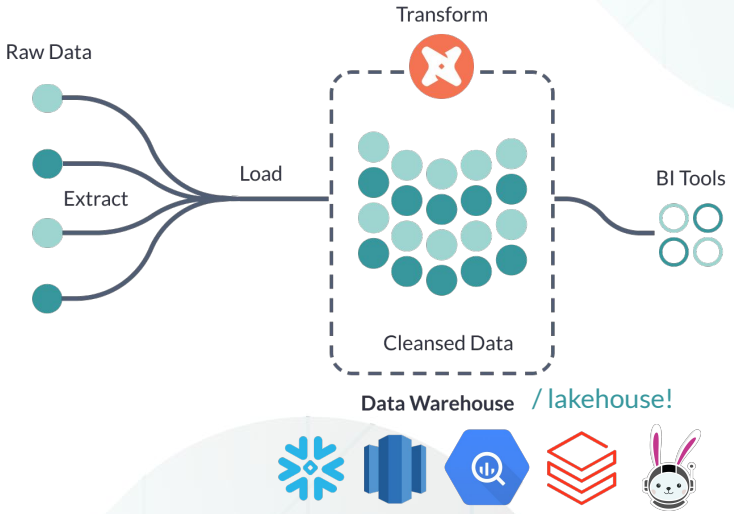
# Cloud warehousing made it cheaper to transform in place

## Legacy E-T-L




- High storage and compute costs
- Disjointed analytics workflows

## Modern E-L-T




- Cloud architecture; SQL-first
- Elastic storage & compute

# The dbt viewpoint: **Build data like developers build applications**

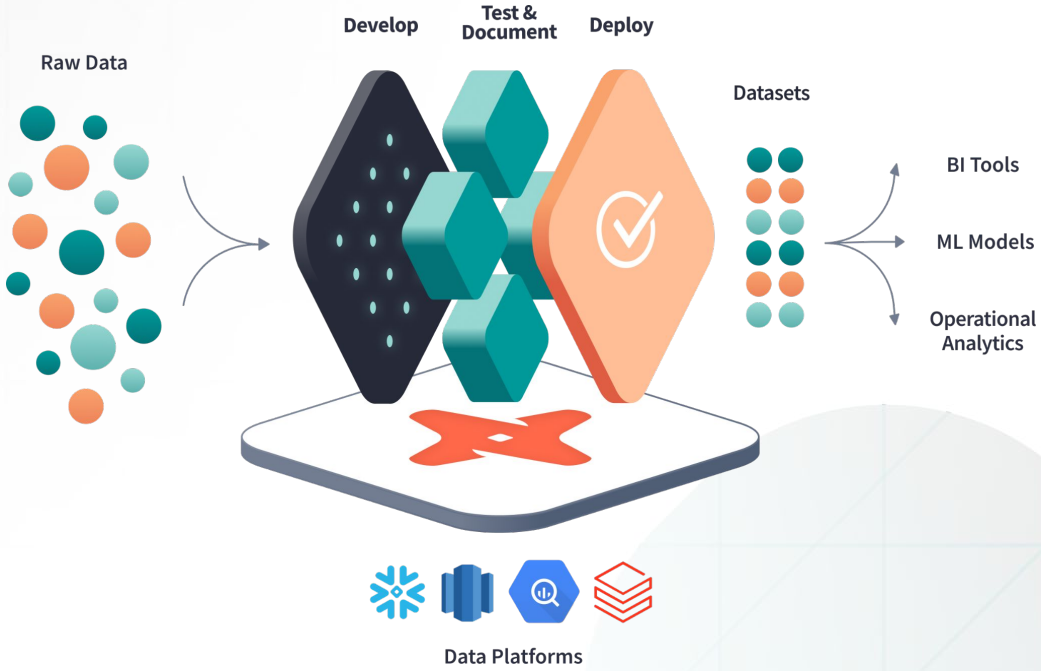
## Unite on SQL

SQL is used by every cloud data warehouse, and known by every data team.

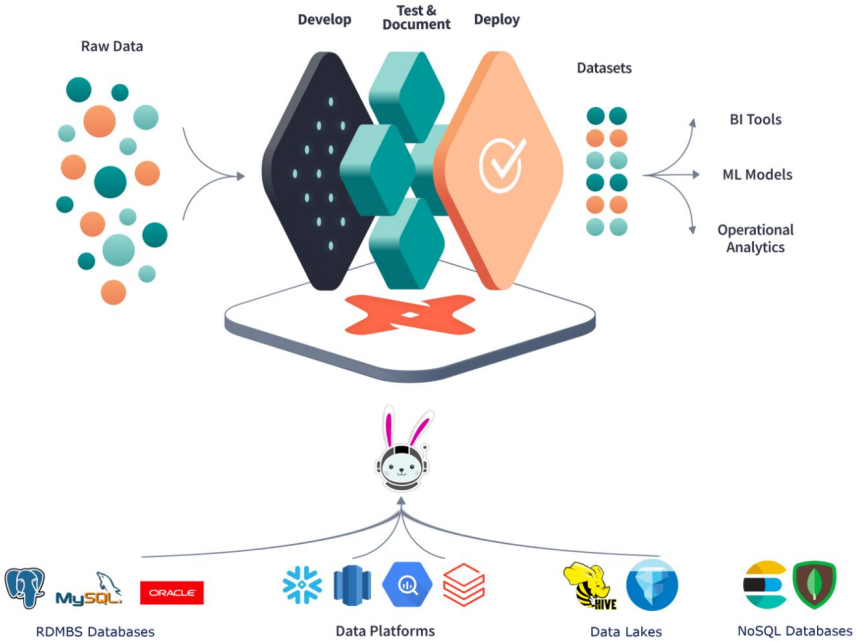
## Work like engineers

Modularity, testing, CI, & documentation promote speed and reliability.

dbt Labs


Develop
Test & Document
Deploy
Raw Data
Datasets
BI Tools
ML Models
Operational Analytics
Data Platforms

Our architecture uses following components:




*Thanks to Brian Olsen from Trino for making this image, and to Michiel De Smet for posting it in a very cool blog post!*

# Open source at its Core — Cloud for the full experience

**dbt Core:** Open source standard for data transformation, testing, documentation

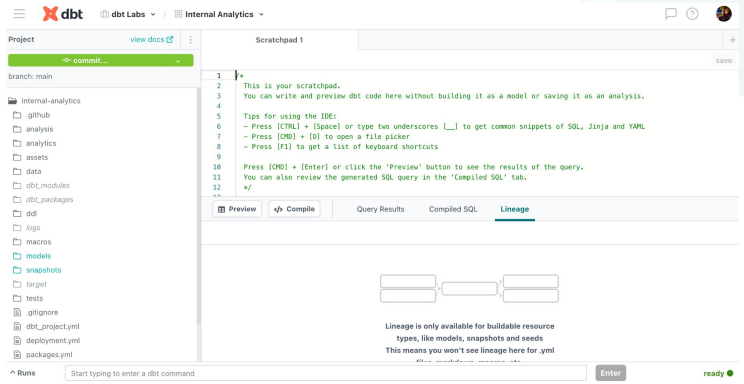
```
14:16:48 ~/dbt-tutorial (master) $ dbt run
Running with dbt=0.15.0
Found 2 models, 4 tests, 0 snapshots, 0 analyses, 133 macros, 0 operations, 0 seed files, 0 sources

14:16:53 | Concurrency: 1 threads (target='dev')
14:16:53 |
14:16:53 | 1 of 2 START table model dbt_claire.my_first_dbt_model.............. [RUN]
14:16:56 | 1 of 2 OK created table model dbt_claire.my_first_dbt_model......... [CREATE TABLE (2) in 3.79s]
14:16:56 | 2 of 2 START view model dbt_claire.my_second_dbt_model.............. [RUN]
14:16:57 | 2 of 2 OK created view model dbt_claire.my_second_dbt_model......... [CREATE VIEW in 0.68s]
14:16:57 |
14:16:57 | Finished running 1 table model, 1 view model in 5.38s.

Completed successfully
```

- Open Source: Apache 2.0
- Includes core SQL compilation logic, Jinja templating, database adapters
- Interface via the CLI

**dbt Cloud:** A fully-managed SaaS experience




- User auth / SSO
- Full IDE to develop and test code
- Simplified Git flow
- Customer Support
- Orchestrate your jobs
- Logging and Alerting
- Integrated documentation
- Metadata API for interoperability

# dbt's development framework promotes **4 key outcomes**

**Collaboration** | dbt code is sql-based and self-documenting; everyone can **work together**

**Velocity** | Focus on analytics, not infrastructure and **ship data products 3x faster**

**Quality** | Test and work from the same assumptions to **ensure alignment**

**Governance** | Standardize processes and control access to **simplify compliance**

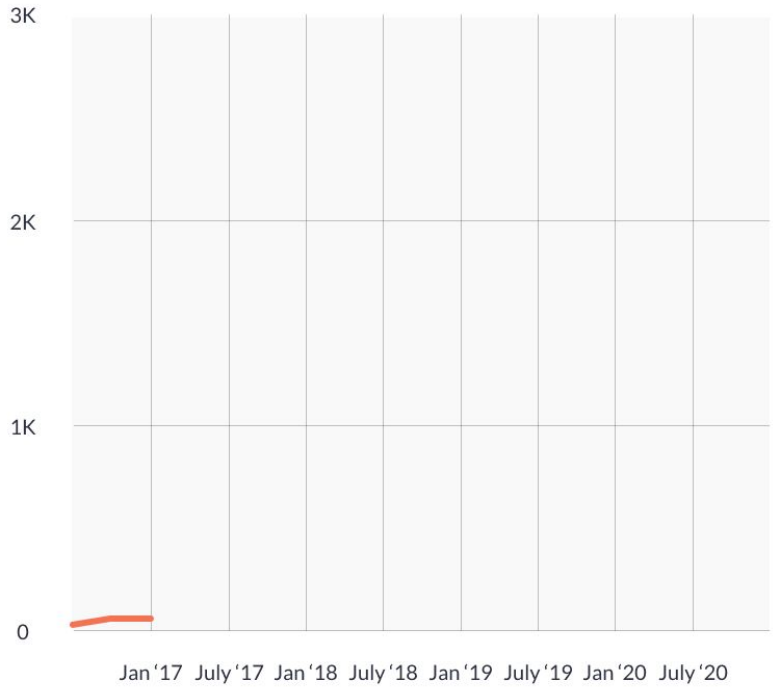

# dbt momentum

# dbt adoption is growing quickly, and organically

**11,000+**
companies using dbt

**1,800+**
dbt Cloud customers

**28,000+**
in the dbt Slack community



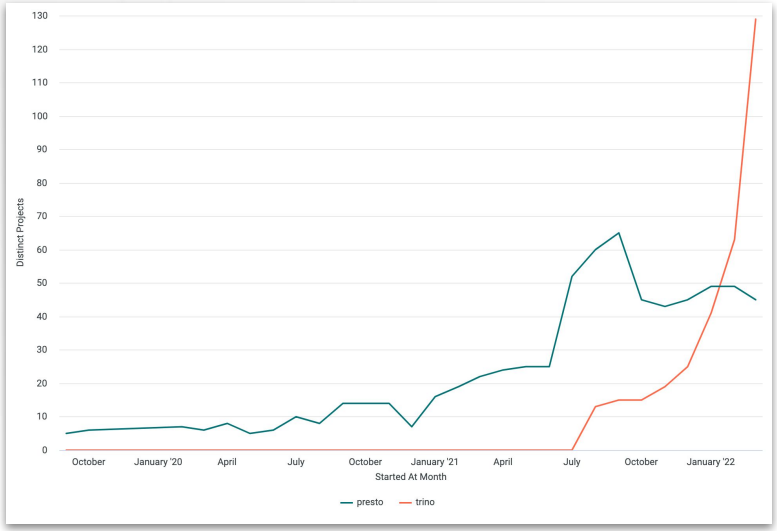
# dbt + Trino, too!

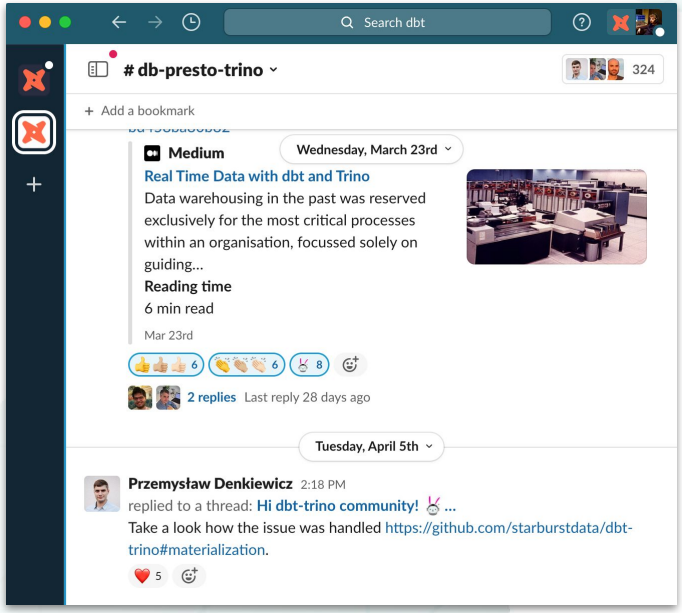
starburstdata / **dbt-trino** Public

The Trino (https://trino.io/) adapter plugin for dbt (https://getdbt.com)

Apache-2.0 License

**42** stars **14** forks



# How it Works

# A centralized environment for collaborative development

## Develop

- IDE or CLI
- Modular SQL
- No DDL/DML
- Pre-built packages

## Document

- Dependency management
- Auto-generate DAG
- Auto-updated docs

## Test

- Generic tests
- Data value testing
- Pre-packaged tests for complex logic

## Deploy

- Job scheduling
- CI/CD
- Version control
- Logging & alerting

# A centralized environment for collaborative development

## Develop

- IDE or CLI
- Modular SQL
- No DDL/DML
- Pre-built packages

# **Develop faster** with SELECT statements

- Express business logic in **SQL**
- **Repeatable builds**
- Includes several **materializations**
  - Table
  - View
  - Incremental

```
-- orders.sql


select *
from analytics.dev.stg_orders
where is_deleted = false
```

**Runs in the warehouse**

```
create table analytics.dev.orders as (

select *
from analytics.dev.stg_orders
where is_deleted = false

);
```

# **Develop faster** without having to think about run order

- Run the same code in dev, test and prod– **the correct schema is resolved for you**
- **Dependencies built automatically** so you can focus on modeling, not run order


```
-- orders.sql


select *
from {{ ref('stg_orders') }}
where is_deleted = false
```

**Runs in the warehouse**

```
create table analytics.dev.orders as (

select *
from analytics.dev.stg_orders
where is_deleted = false

);
```

## **Develop faster** without having to think about run order

- Run the same code in dev, test and prod– **the correct schema is resolved for you**
- **Dependencies built automatically** so you can focus on modeling, not run order


`-- orders.sql`

```
select *
from {{ ref('stg_orders') }}
where is_deleted = false
```

**Runs in the warehouse**

```
create table analytics.prod.orders as (

select *
from analytics.prod.stg_orders
where is_deleted = false

);
```

# A centralized environment for collaborative development

## Develop

- IDE or CLI
- Modular SQL
- No DDL/DML
- Pre-built packages

## Document

- Dependency management
- Auto-generate DAG
- Auto-updated docs

# Maintain shared understanding with auto-updating lineage

# A centralized environment for collaborative development

## Develop

- IDE or CLI
- Modular SQL
- No DDL/DML
- Pre-built packages

## Document

- Dependency management
- Auto-generate DAG
- Auto-updated docs

## Test

- Generic tests
- Data value testing
- Pre-packaged tests for complex logic

# **Preserve quality** by testing in-line

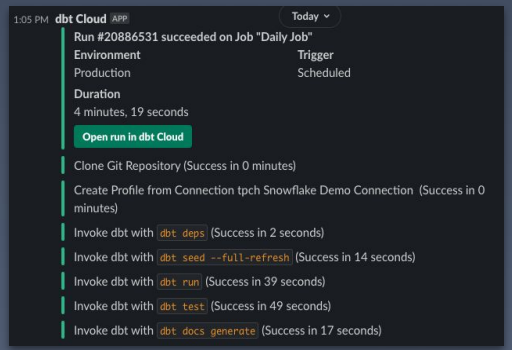
- Test **assumptions** about data, and the **validity** of transformations
- **Custom + out of the box** tests including:
  - Uniqueness
  - Null values
  - Certain values
  - Is a valid foreign key to another table
- Learn about issues before stakeholders with fail/warn alerting

schema.yml

```yaml
columns:
  - name: order_id
    tests:
      - unique
      - not null
  - name: status
    tests:
    - accepted_values:
        values: ['placed', 'shipped', 'completed']
```

tests/assert_payment_amount_is_positive.sql

```sql
select
    order_id,
    sum(amount) as total_amount
from {{ ref('fct_payments' )}}
group by 1
having not(total_amount >= 0)
```



# A centralized environment for collaborative development

## Develop

- IDE or CLI
- Modular SQL
- No DDL/DML
- Pre-built packages

## Document

- Dependency management
- Auto-generate DAG
- Auto-updated docs

## Test

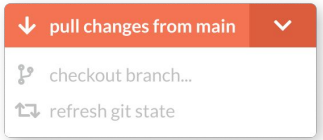
- Schema tests
- Data value testing
- Pre-packaged tests for complex logic

## Deploy

- Job scheduling
- CI/CD
- Version control
- Logging & alerting

# Deploy seamlessly with version control and CI/CD

## Version Control

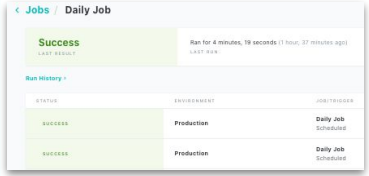



Integrate with git provider of choice

## Continuous Integration


Minimize wasteful runs by testing only changes

## Continuous Deployment




Job scheduling and alerting

**Logging & Alerting**

# Demo

# Demo highlights

- Auto-generated documentation
- Local + dbt Cloud development
- Modular definitions + testing
- Macros + packages to level up SQL
- Easy access to data sources *wherever* they live

# Not included in the demo

- CI/CD in dbt Cloud
- Metadata produced by dbt
- Incremental processing at scale, e.g. powered by Trino’s Delta Lake connector

(lots more!)


# Thank you!