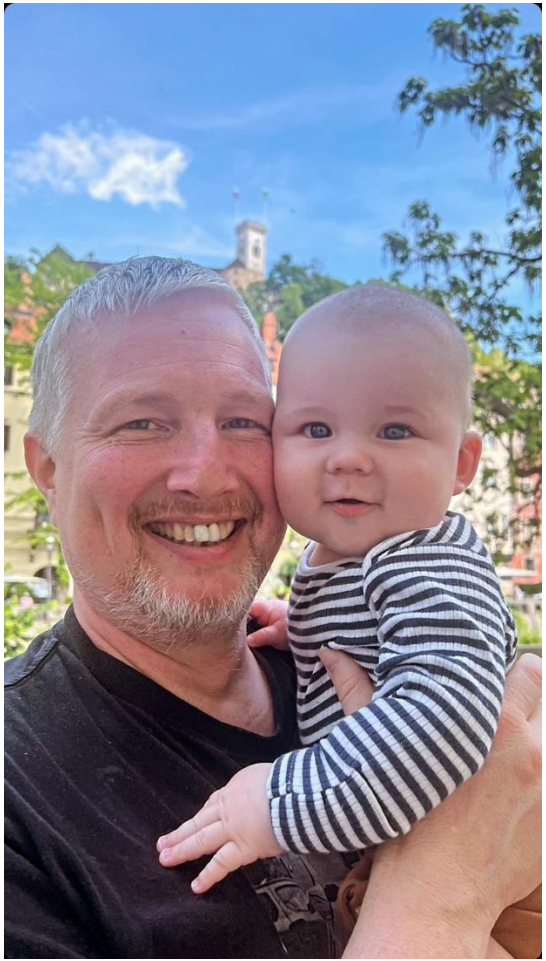
SQL
KONFERENZ
2025



An Introduction to dbt

André Kamman



DATA SATURDAY Holland

@andrekamman.com

Microsoft MVP Most Valuable Professional 2023

'89

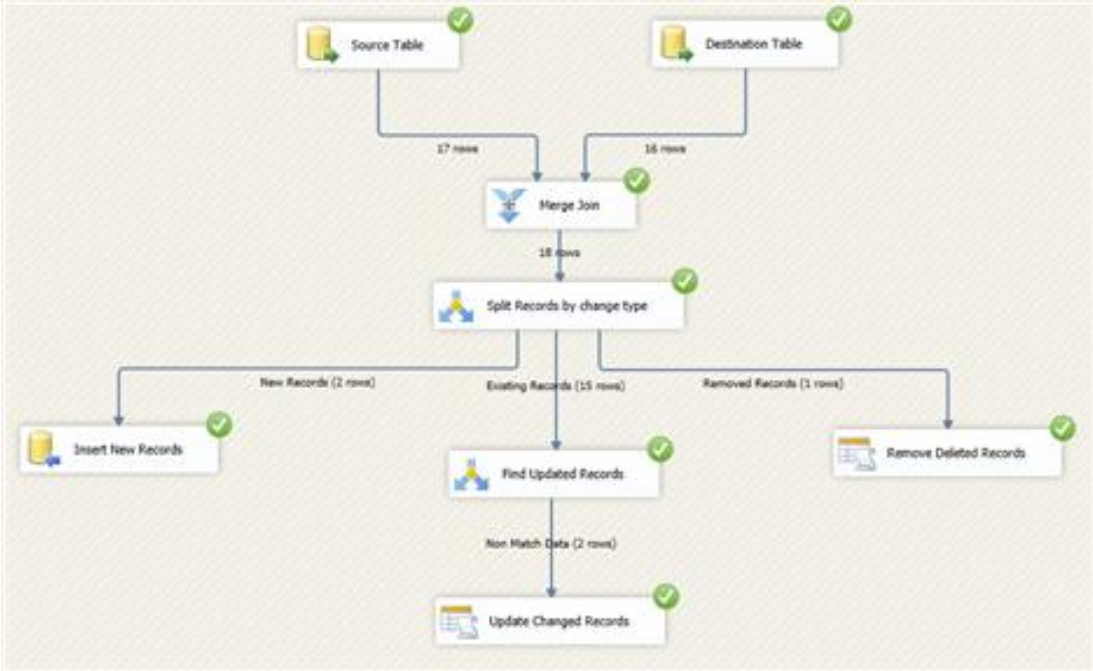
dbt

databricks

MICROSOFT FABRIC

SQL KONFERENZ 2025

Remember this?



source: https://radacad.com/insert-update-and-delete-destination-table-with-ssis

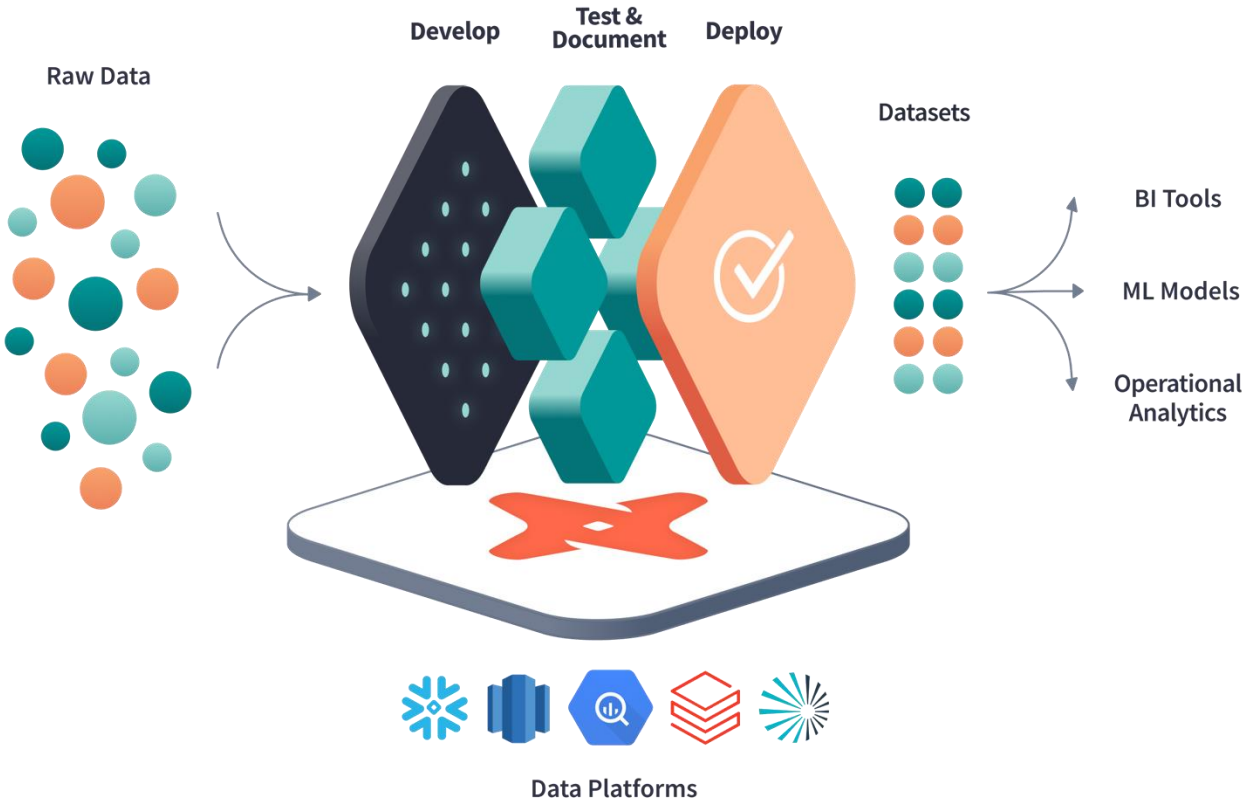


How about this?

DeployTargetTables.biml

```
<#@ template tier="2" #>
<Biml xmlns="http://schemas.varigence.com/biml.xsd">
  <Packages>
    <Package Name="MasterTableDeploy" ConstraintMode="Parallel">
      <Tasks>
        <# foreach (var table in RootNode.Tables) { #>
        <ExecuteSQL Name="Create <#=table.Name#>" ConnectionName="Target">
          <DirectInput><#=table.GetDropAndCreateDdl()#></DirectInput>
        </ExecuteSQL>
        <# } #>
      </Tasks>
    </Package>
  </Packages>
</Biml>
```

source: https://varigence.com/bimlexpress



source: https://getdbt.com

Start using dbt now

Start with a forever-free Developer account. New accounts include access to a 14-day free trial of the Team plan.

Developer

A best-in-class analytics engineering workflow for data teams of one

Free

Forever for one developer seat

Sign up

Features

- Browser-based IDE
- Job scheduling
- Unlimited daily runs
- One project limit
- Logging & alerting
- Data documentation
- Source freshness reporting
- Continuous integration
- Native support for GitHub and GitLab
- US hosted

Team

Collaborate on the analytics engineering workflow

$100

Per developer seat per month

Start free trial

Features

- All features in Developer
- Add up to 8 seats
- One project limit
- 5 read-only seats
- Up to 2 concurrently running jobs
- API access
- Outbound webhooks
- Semantic Layer

Enterprise

Customize your deployment and apply fine-grained controls

Custom

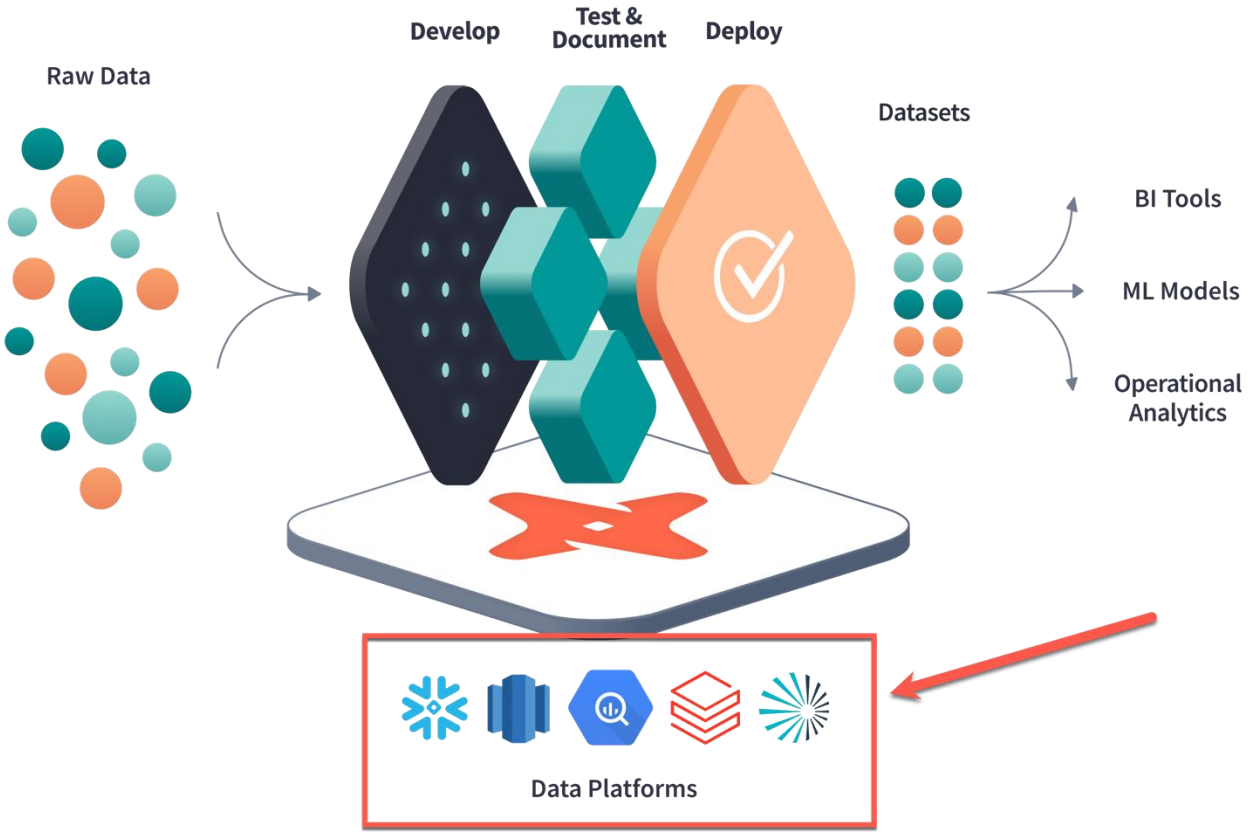
For more security and control

Book a demo

Features

- All features in Team
- Unlimited projects
- Single Sign On (SSO)
- Multiple deployment regions
- Service Level Agreements (SLAs)
- Professional Services
- Role-based ACLs
- Fine-grained Git permissions
- Audit logging
- Native support for GitHub, GitLab, and Azure DevOps

source: https://getdbt.com

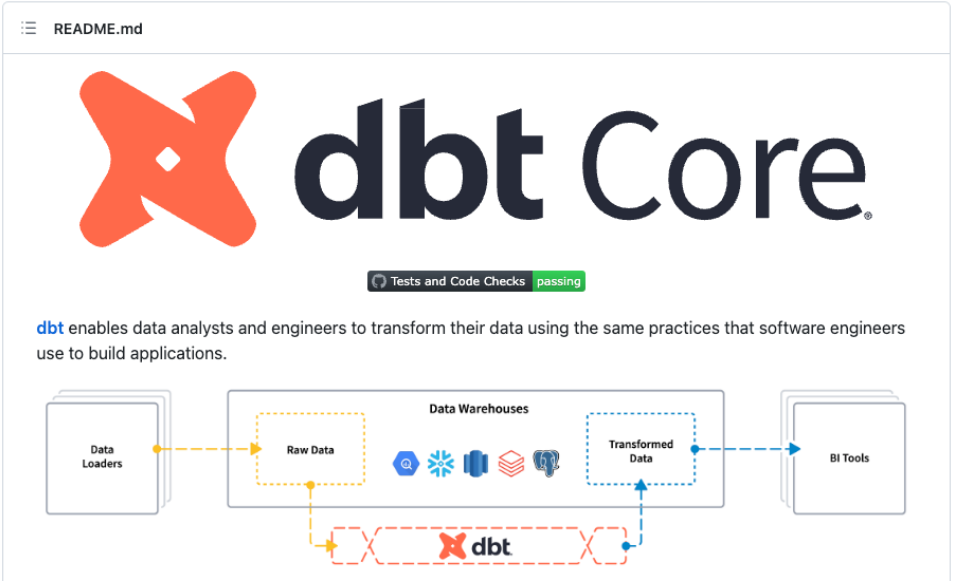

Develop
Test &
Document
Deploy
Raw Data
Datasets
BI Tools
ML Models
Operational
Analytics
Data Platforms

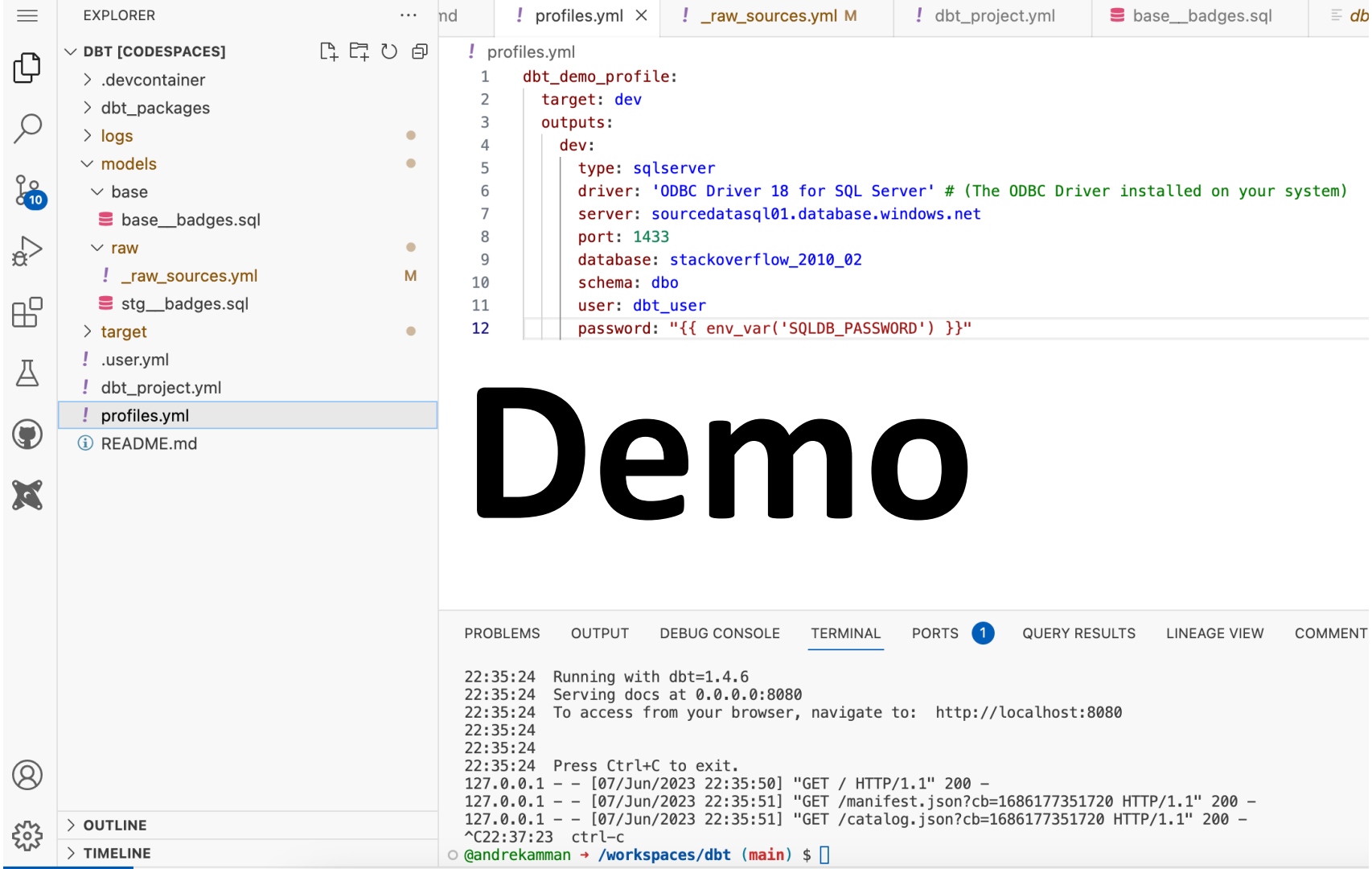
ELT

ELT

What?

- dbt is open source
- Transforms
- Tests
- Documentation




EXPLORER
DBT [CODESPACES]
.devcontainer
dbt_packages
logs
models
base
base__badges.sql
raw
_raw_sources.yml
M
stg__badges.sql
target
.user.yml
dbt_project.yml
profiles.yml
README.md
OUTLINE
TIMELINE
profiles.yml
_raw_sources.yml M
dbt_project.yml
base__badges.sql
profiles.yml
1 dbt_demo_profile:
2 target: dev
3 outputs:
4 dev:
5 type: sqlserver
6 driver: 'ODBC Driver 18 for SQL Server' # (The ODBC Driver installed on your system)
7 server: sourcedatasql01.database.windows.net
8 port: 1433
9 database: stackoverflow_2010_02
10 schema: dbo
11 user: dbt_user
12 password: "{{ env_var('SQLDB_PASSWORD') }}"
Demo
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
1
QUERY RESULTS
LINEAGE VIEW
COMMENT
22:35:24 Running with dbt=1.4.6
22:35:24 Serving docs at 0.0.0.0:8080
22:35:24 To access from your browser, navigate to: http://localhost:8080
22:35:24
22:35:24
22:35:24 Press Ctrl+C to exit.
127.0.0.1 - - [07/Jun/2023 22:35:50] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2023 22:35:51] "GET /manifest.json?cb=1686177351720 HTTP/1.1" 200 -
127.0.0.1 - - [07/Jun/2023 22:35:51] "GET /catalog.json?cb=1686177351720 HTTP/1.1" 200 -
^C22:37:23 ctrl-c
@andrekamman → /workspaces/dbt (main) $

docs,source,init,clean,debug
,deps,list,ls,build,snapshot,
run,compile,parse,test,seed,
run-operation

Ephemeral

```
models:
    stackoverflow:
    raw:
    +materialized: view
    base:
    +materialized: table
```

id	status	updated_at
1	pending	2024-01-01

id	status	updated_at
1	shipped	2024-01-02

Snapshot

* I had to get used to that name as well, it’s basically SCD II

id	status	updated_at	dbt_valid_from	dbt_valid_to
1	pending	2024-01-01	2024-01-01	2024-01-02
1	shipped	2024-01-02	2024-01-02	`null`

```
snapshots:
  - name: orders_snapshot
    relation: source('jaffle_shop', 'orders')
    config:
      schema: snapshots
      database: analytics
      unique_key: id
      strategy: timestamp
      updated_at: updated_at
      dbt_valid_to_current: "to_date('9999-12-31')"
```

```
{% snapshot orders_snapshot_timestamp %}

    {{
        config(
          target_schema='snapshots',
          strategy='timestamp',
          unique_key='id',
          updated_at='updated_at',
        )
    }}

    select * from {{ source('jaffle_shop', 'orders') }}

{% endsnapshot %}
```

```
$ dbt snapshot
Running with dbt=1.9.0

15:07:36 | Concurrency: 8 threads (target='dev')
15:07:36 |
15:07:36 | 1 of 1 START snapshot snapshots.orders_snapshot..... [RUN]
15:07:36 | 1 of 1 OK snapshot snapshots.orders_snapshot.....[SELECT 3 in 1.82s]
15:07:36 |
15:07:36 | Finished running 1 snapshots in 0.68s.

Completed successfully

Done. PASS=2 ERROR=0 SKIP=0 TOTAL=1
```

Incremental

```
{{ config(
    materialized='incremental',
    file_format='delta', # or 'hudi'
    unique_key='user_id',
    incremental_strategy='merge'
) }}

with new_events as (
    select * from {{ ref('events') }}

    {% if is_incremental() %}
    where date_day >= date_add(current_date, -1)
    {% endif %}
)
```

* This is basically scd 1

https://docs.getdbt.com/docs/build/incremental-strategy

jinja

```
{% set payment_methods = ["bank_transfer", "credit_card", "gift_card"] %}

select
    order_id,
    {% for payment_method in payment_methods %}
    sum(case when payment_method = '{{payment_method}}' then amount end) as
{{payment_method}}_amount,
    {% endfor %}
    sum(amount) as total_amount
from app_data.payments
group by 1
```

macros

```
{% macro cents_to_dollars(column_name, scale=2) %}
    (({{ column_name }} / 100)::numeric(16, {{ scale }})
{% endmacro %}
```

```
select
  id as payment_id,
  {{ cents_to_dollars('amount') }} as amount_usd,
  ...
from app_data.payments
```

Slim CI

```
dbt build
    --select "state:modified+1"
    --state "ci_main"
    --defer
```

sqlfluff

```
== [/Users/andre/jaffle_shop_duckdb/target/run/jaffle_shop/models/staging/schema.yml/not_null_stg_payments_payment_id.sql] FAIL
L:  2 | P:  1 | LT02 | Expected indent of 4 spaces. [layout.indent]
L:  3 | P:  1 | LT02 | Expected indent of 4 spaces. [layout.indent]
L:  4 | P:  1 | LT02 | Expected indent of 4 spaces. [layout.indent]
L:  5 | P:  1 | LT02 | Line should not be indented. [layout.indent]
L:  6 | P:  1 | LT01 | Unnecessary trailing whitespace.
                     | [layout.spacing]
L:  7 | P:  1 | LT01 | Unnecessary trailing whitespace.
                     | [layout.spacing]
L:  8 | P:  1 | LT01 | Unnecessary trailing whitespace.
                     | [layout.spacing]
L: 12 | P:  1 | LT02 | Expected indent of 4 spaces. [layout.indent]
L: 13 | P:  1 | LT02 | Expected indent of 4 spaces. [layout.indent]
L: 14 | P:  1 | LT02 | Expected indent of 4 spaces. [layout.indent]
L: 18 | P:  1 | LT01 | Unnecessary trailing whitespace.
                     | [layout.spacing]
L: 19 | P:  1 | LT02 | Line should not be indented. [layout.indent]
L: 19 | P:  7 | AL01 | Implicit/explicit aliasing of table.
                     | [aliasing.table]
L: 19 | P: 24 | LT12 | Files must end with a single trailing newline.
                     | [layout.end_of_file]
```

Packages

Featured Packages



dbt_utils



codegen



dbt_project_evaluator



audit_helper



dbt_external_tables



dbt_expectations

Tests

```
version: 2

models:
  - name: orders
    columns:
      - name: order_id
        tests:
          - unique
          - not_null
```

```
version: 2

models:
  - name: orders
    columns:
      - name: order_id
        tests:
          - unique
          - not_null
      - name: status
        tests:
          - accepted_values:
              values: ['placed', 'shipped', 'completed', 'returned']
      - name: customer_id
        tests:
          - relationships:
              to: ref('customers')
              field: id
```

```
-- Refunds have a negative amount, so the total amount should always be >= 0.
-- Therefore return records where total_amount < 0 to make the test fail.
select
    order_id,
    sum(amount) as total_amount
from {{ ref('fct_payments') }}
group by 1
having total_amount < 0
```

```
version: 2
data_tests:
  - name: assert_total_payment_amount_is_positive
    description: >
      Refunds have a negative amount, so the total amount should always be >= 0.
      Therefore return records where total amount < 0 to make the test fail.
```

```
$ dbt test

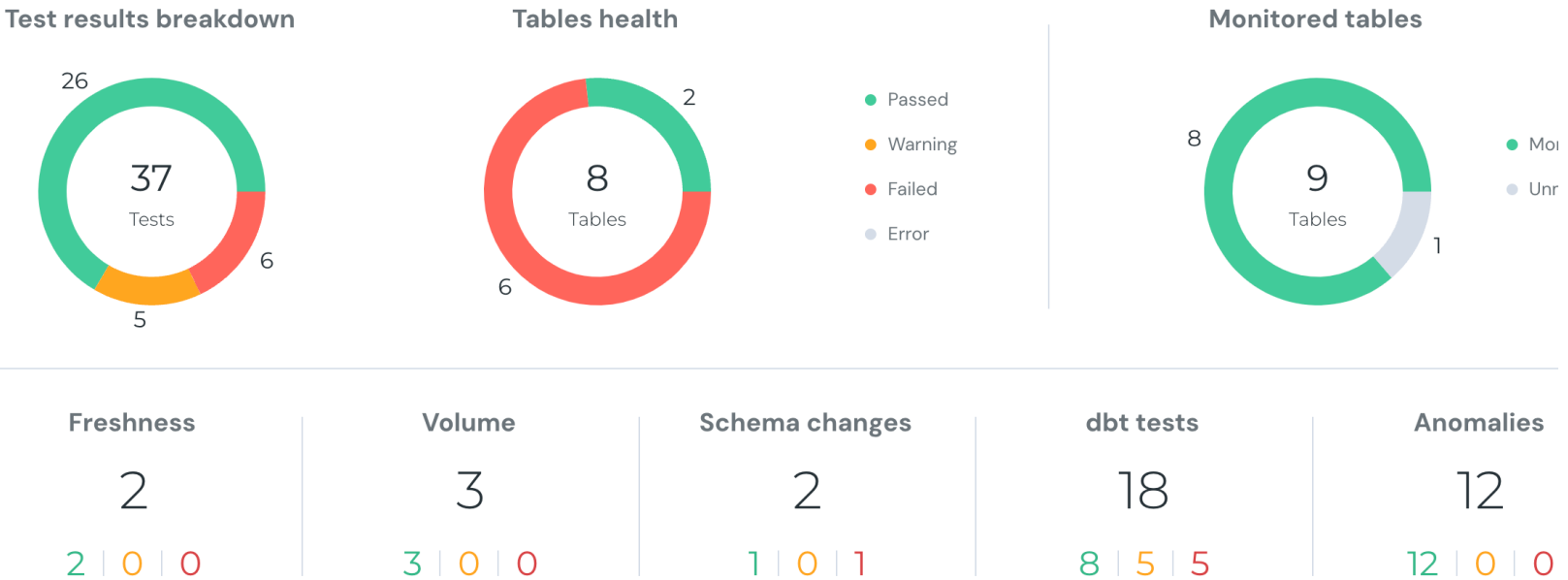
Found 3 models, 2 tests, 0 snapshots, 0 analyses, 130 macros, 0 operations, 0 seed files, 0
sources

17:31:05 | Concurrency: 1 threads (target='learn')
17:31:05 |
17:31:05 | 1 of 2 START test not_null_order_order_id..... [RUN]
17:31:06 | 1 of 2 PASS not_null_order_order_id..... [PASS in 0.99s]
17:31:06 | 2 of 2 START test unique_order_order_id..... [RUN]
17:31:07 | 2 of 2 PASS unique_order_order_id..... [PASS in 0.79s]
17:31:07 |
17:31:07 | Finished running 2 tests in 7.17s.

Completed successfully

Done. PASS=2 WARN=0 ERROR=0 SKIP=0 TOTAL=2
```

Package: Elementary
* dbt packages which exposes metadata and runtime result tables
* elementary cli to collect data for self hosting of report
* paid SaaS service with additional features



dbt-fabric

- Warehouse only
- No dbt-utils package



Biggest advantage

* imho

Forces Engineering practices, automated lineage, documentation, complex engine to run everything in the right order, etc.

Meanwhile

Analysts write one language: SQL
(sort of, there is python jinja in the mix and you can write python models if you really need to)

Analysts can do their thing, talk to stakeholders, and make sh*t happen, without having to ask engineers to get things into production for them !!!



Thank You!

Please provide feedback

