

DBT

Shani Cohen

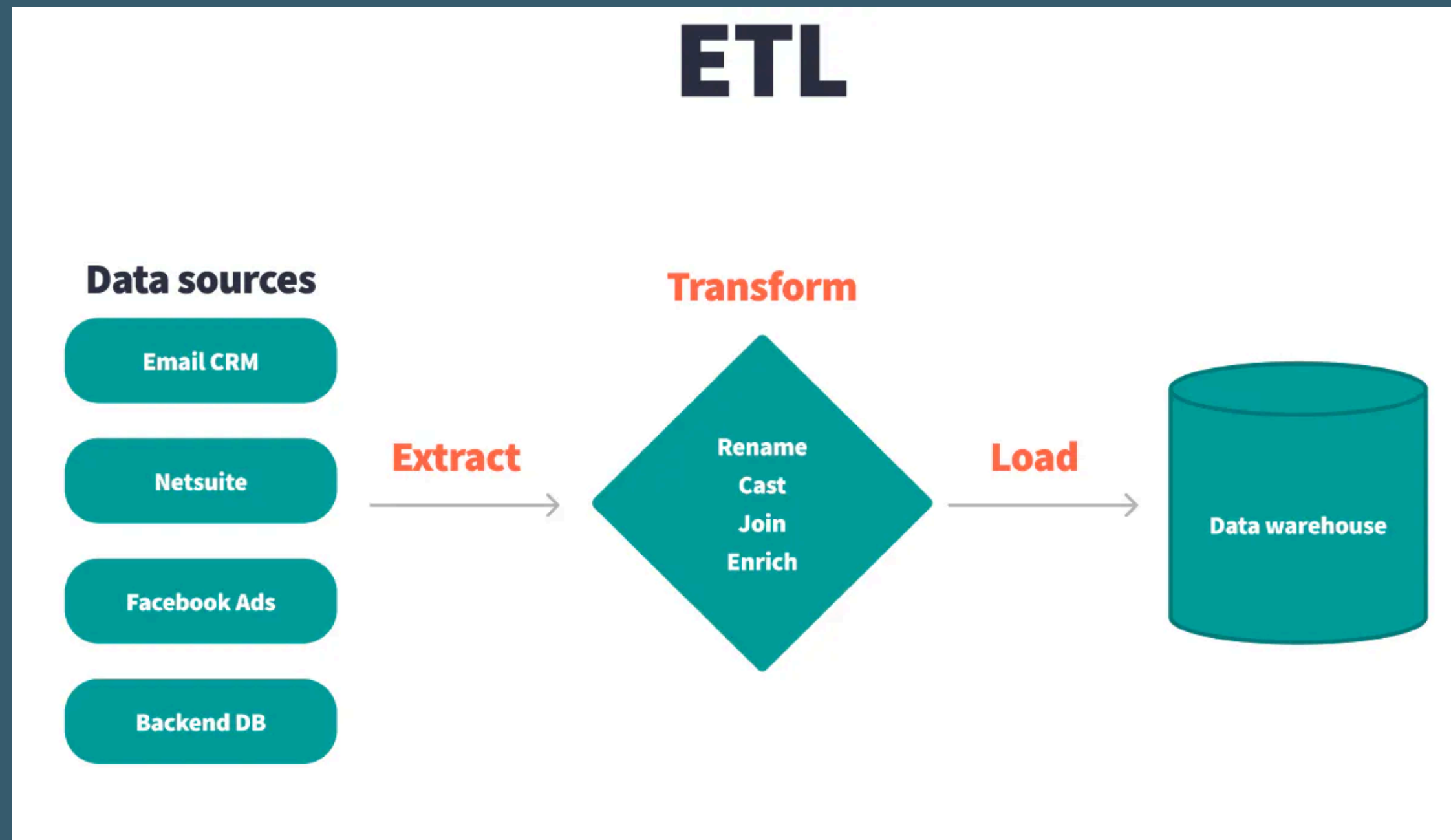
**What, exactly, is
dbt?**

dbt Labs raises \$222M in Series D funding at \$4.2B valuation led by Altimeter with participation from Databricks and Snowflake

ETL vs ELT

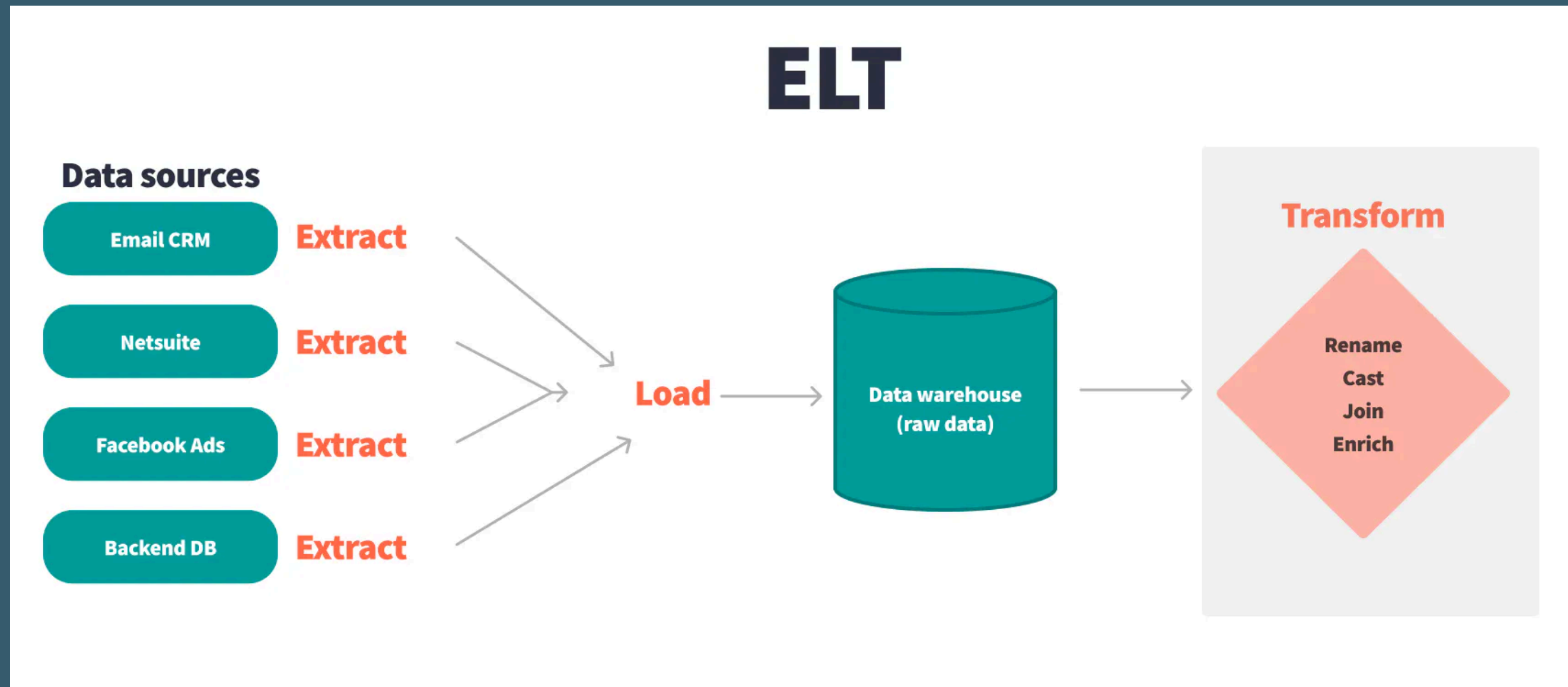
ETL

Extract Transform Load

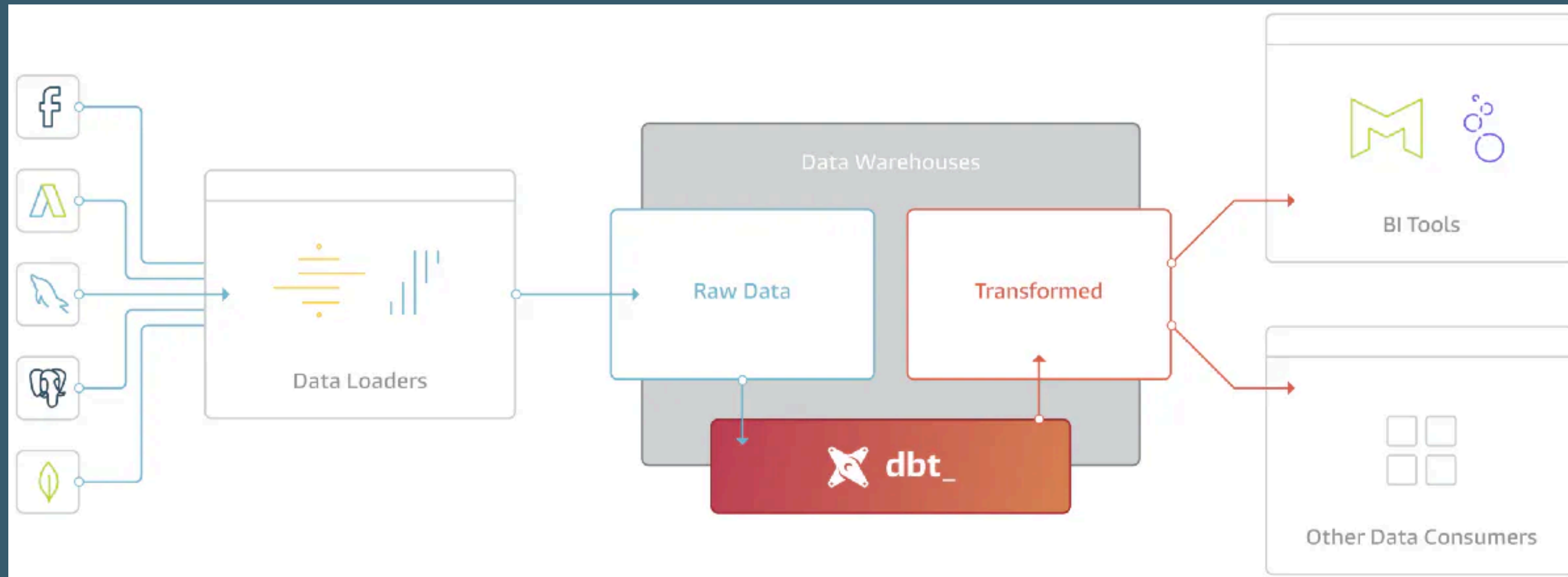


ELT

Extract Load Transform



DBT and the modern BI stack



神社 Jinja


```
import jinja2

# loading the environment
environment = jinja2.Environment()

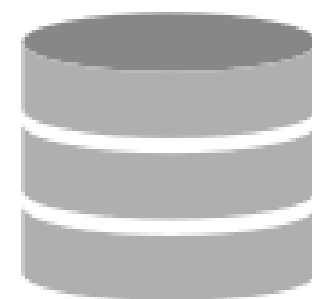
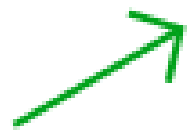
# loading the template
template = environment.from_string("Hello, {{ name }}!")

# rendering the template and storing the resultant text in variable output
rendered = template.render(name="World")

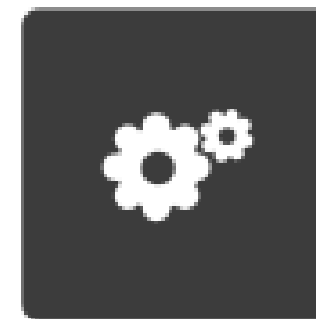
# printing the output on screen
print(rendered)
```



Jinja
Template



Data



Jinja
Rendering
Engine



Output
text

Delimiters

- `{{ }}` for expressions.
- `{# #}` for comments (even multiline) inside the template.
- `{% %}` for jinja statements (like loops, etc.)

Decisions

```
% if <condition> %} <if block>  
{% elif <condition2> %} <elif block>  
<%else%> <else block>  
<% endif %>
```

Exercises

<https://github.com/shanicohen1902/tikal-data-engineering-workshop>



WHY DUckDB?

- Simple
- Feature Reach
- Free
- Fast

DuckDB CLI

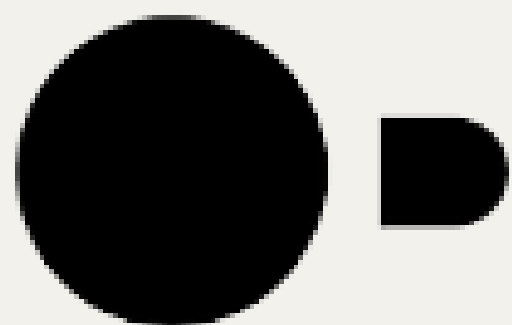
```
$ duckdb #In memory db
```

```
$ duckdb my_database.duckdb # persistent db
```

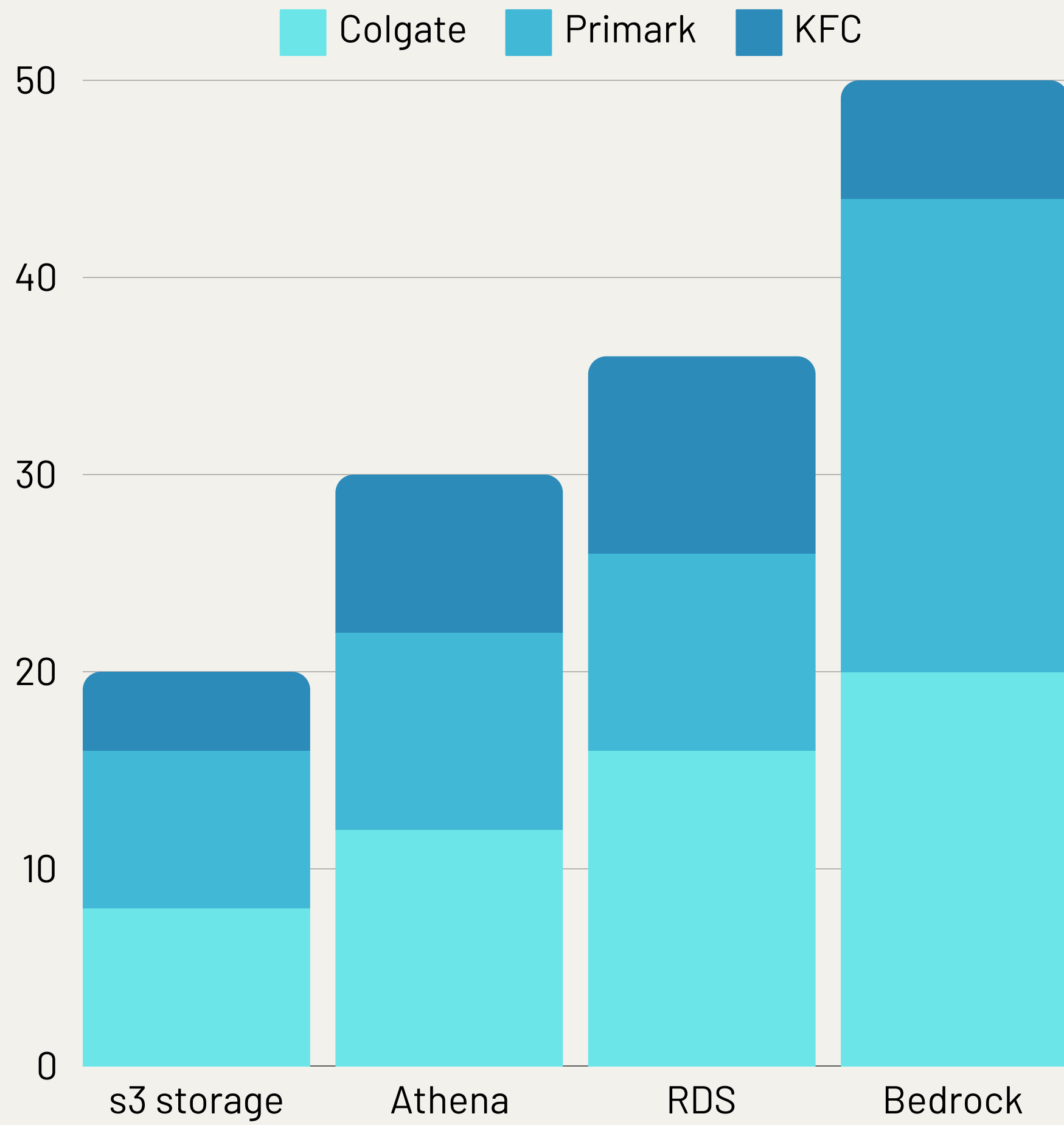
```
SELECT 'quack' AS my_column; # Running sql
```

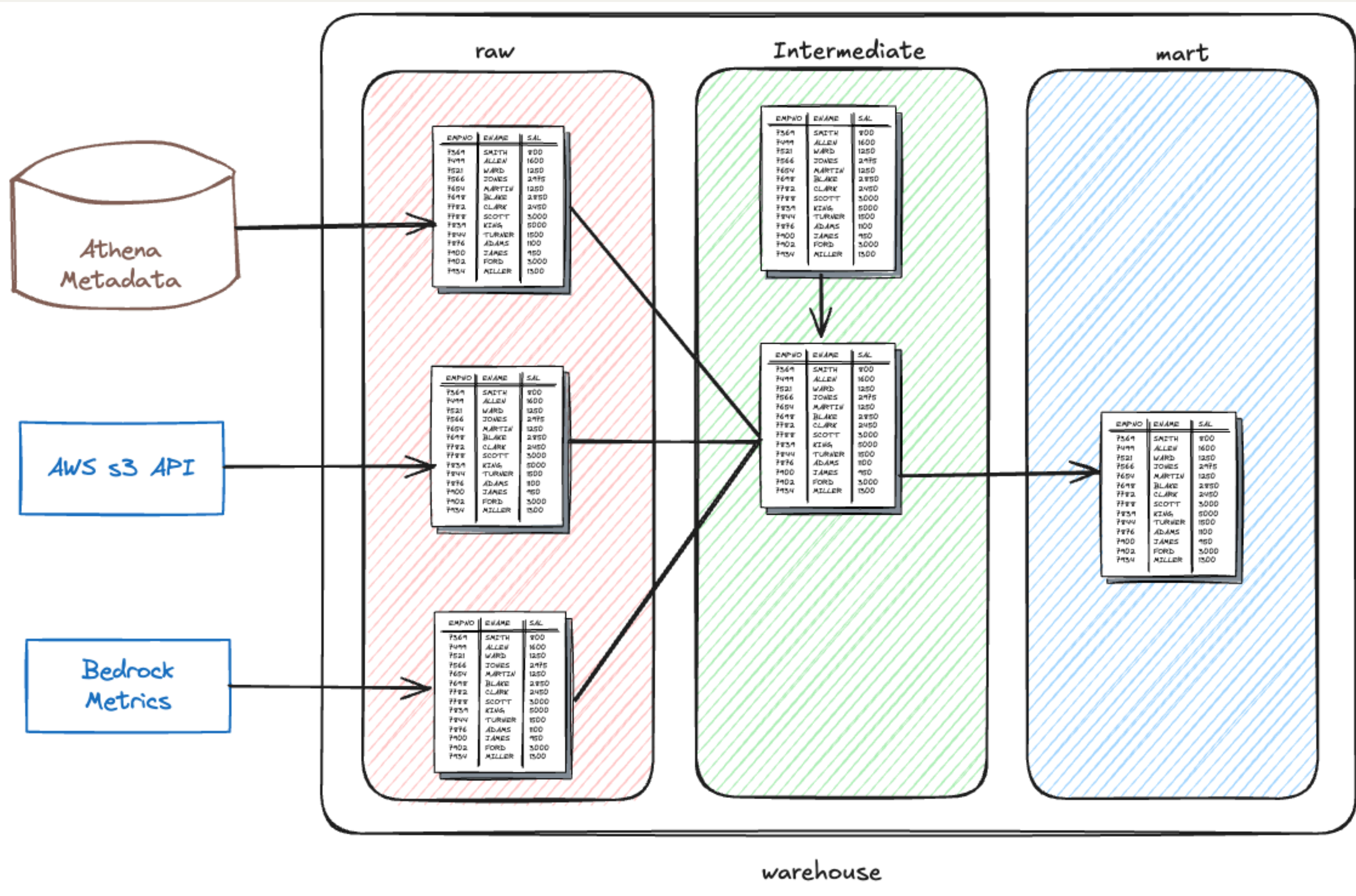


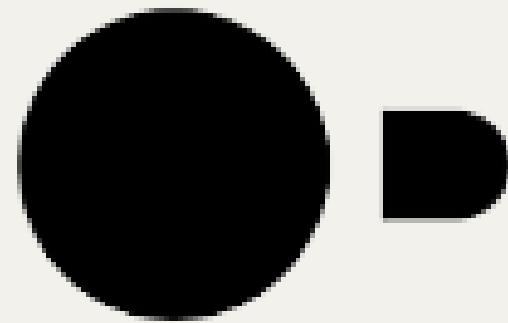

dbt_



DuckDB







dbt_

DuckDB

▼  cost_allocation

>  analyses

>  macros

>  models

>  seeds

>  snapshots

>  tests

 .gitignore

 dbt_project.yml

 readme.md

DBT project structure

Write your first DBT Model

models/customers.sql

```
with customer_orders as (  
    select  
        customer_id,  
        min(order_date) as first_order_date,  
        max(order_date) as most_recent_order_date,  
        count(order_id) as number_of_orders  
  
    from jaffle_shop.orders  
  
    group by 1  
)  
  
select  
    customers.customer_id,  
    customers.first_name,  
    customers.last_name,  
    customer_orders.first_order_date,  
    customer_orders.most_recent_order_date,  
    coalesce(customer_orders.number_of_orders, 0)  
  
from jaffle_shop.customers  
  
left join customer_orders using (customer_id)
```

dbt_project.yml

Every dbt project needs a
dbt_project.yml file

DBT basic commands

```
$ dbt compile # models to sql
```

```
$ dbt run # run :)
```

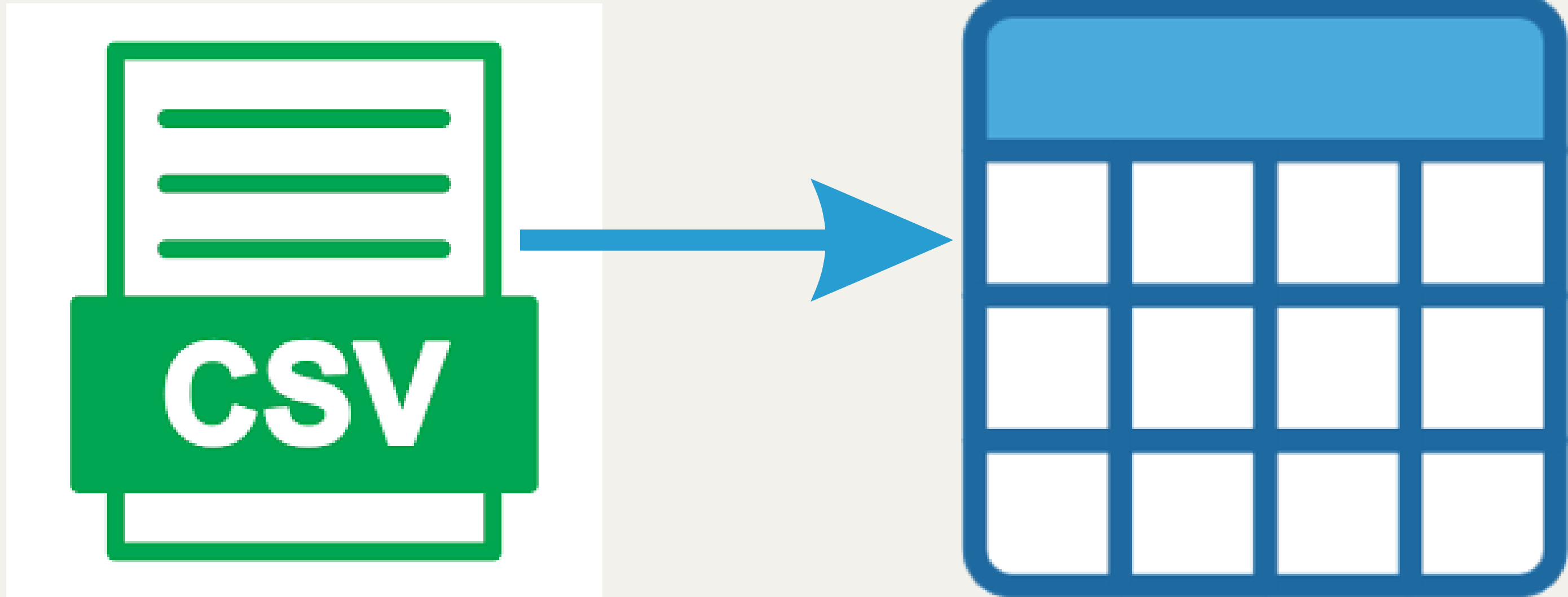
```
$ dbt test # test :)
```


Compile



manifest.json

Seeds



Seeds

```
$ dbt seed # csv to table
```

```
$ dbt seed --full-refresh
```

Exercise 1: DBT Seeds

Instructions:

1. Run the command `dbt seed` to load the data into your dbt project.
2. Verify the data is loaded correctly by querying the tables in your database.

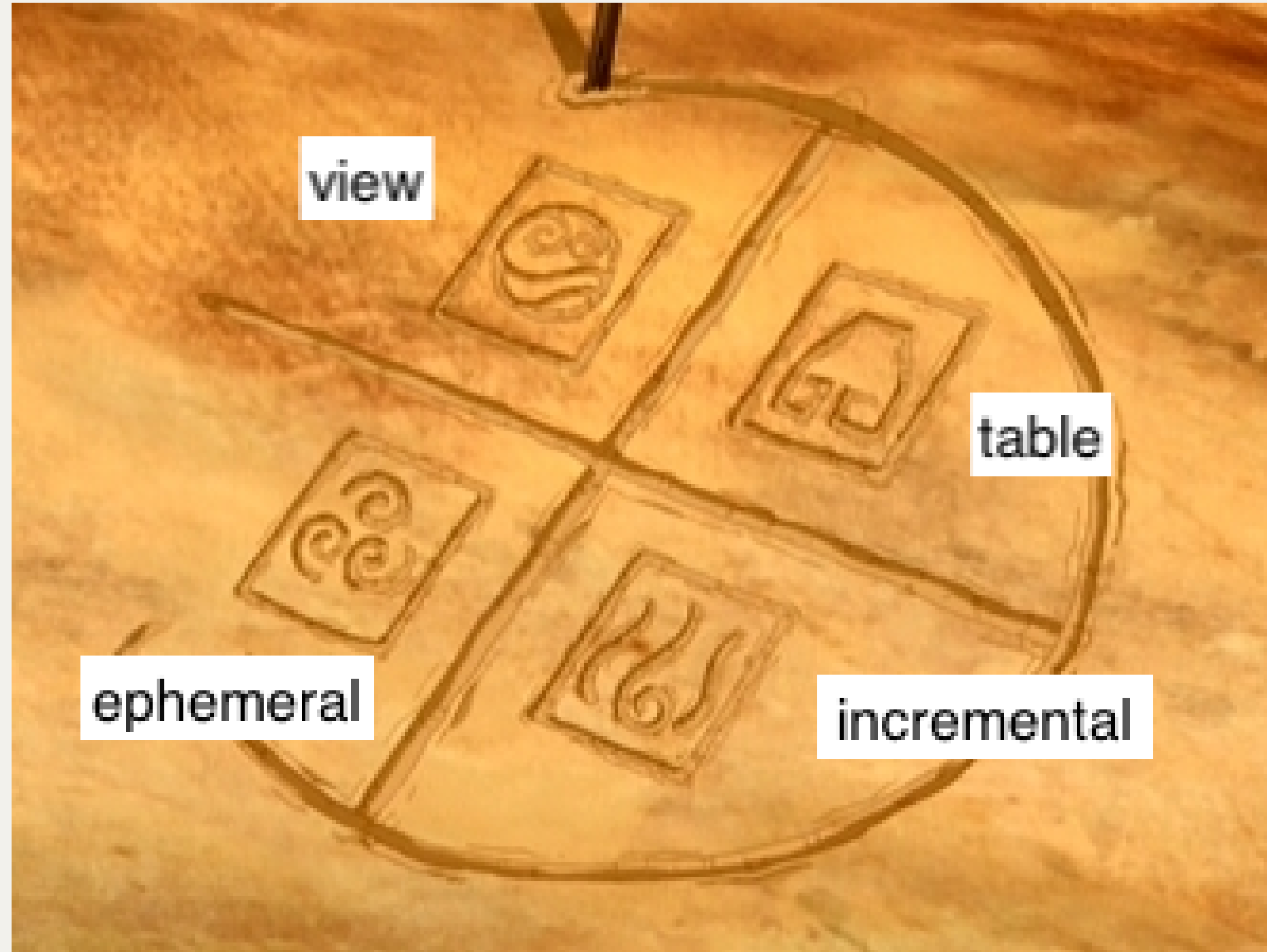
Answer the following questions using DuckDb cli:

Which tables created in the db?

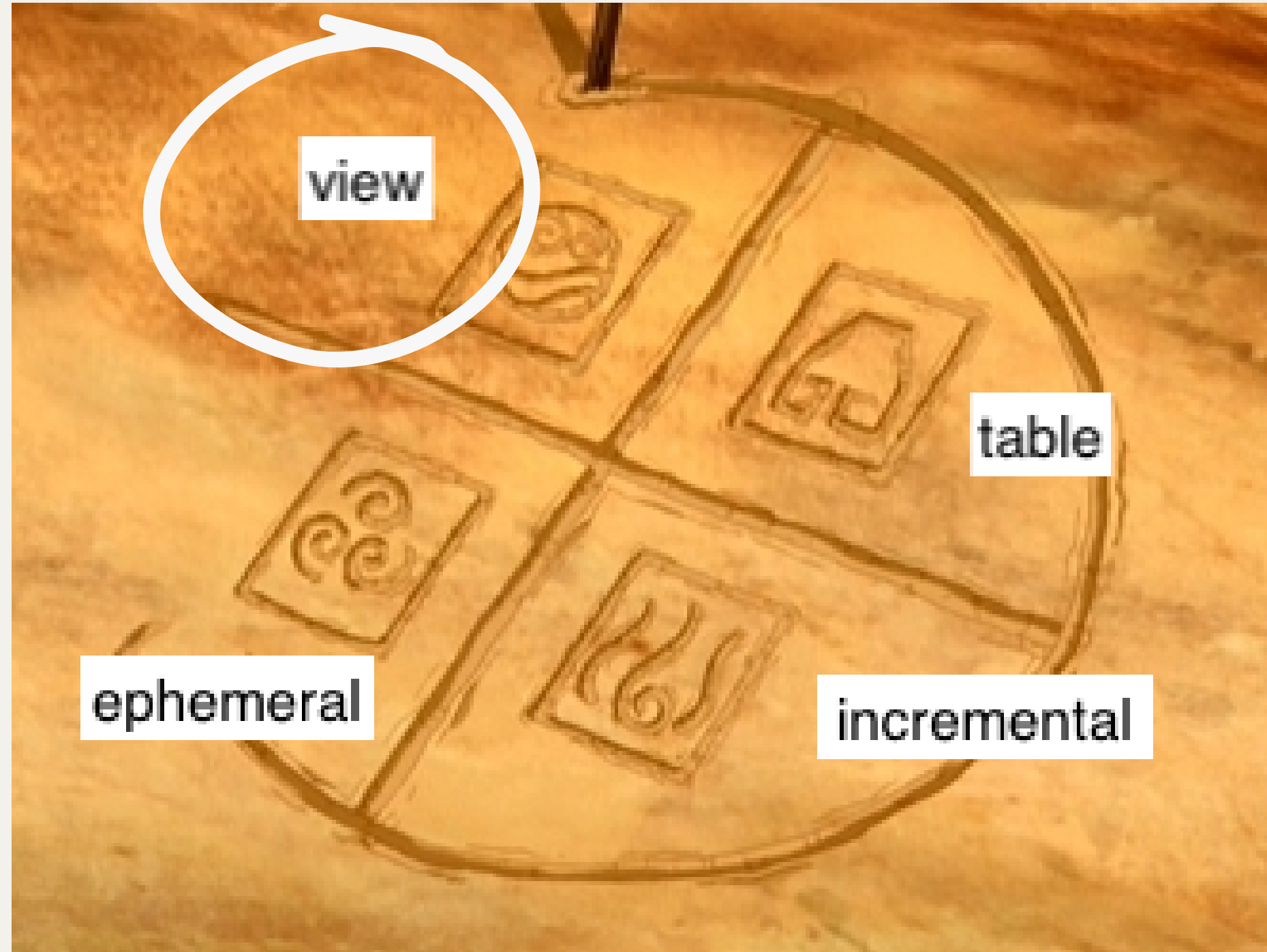
How many lines in each table?



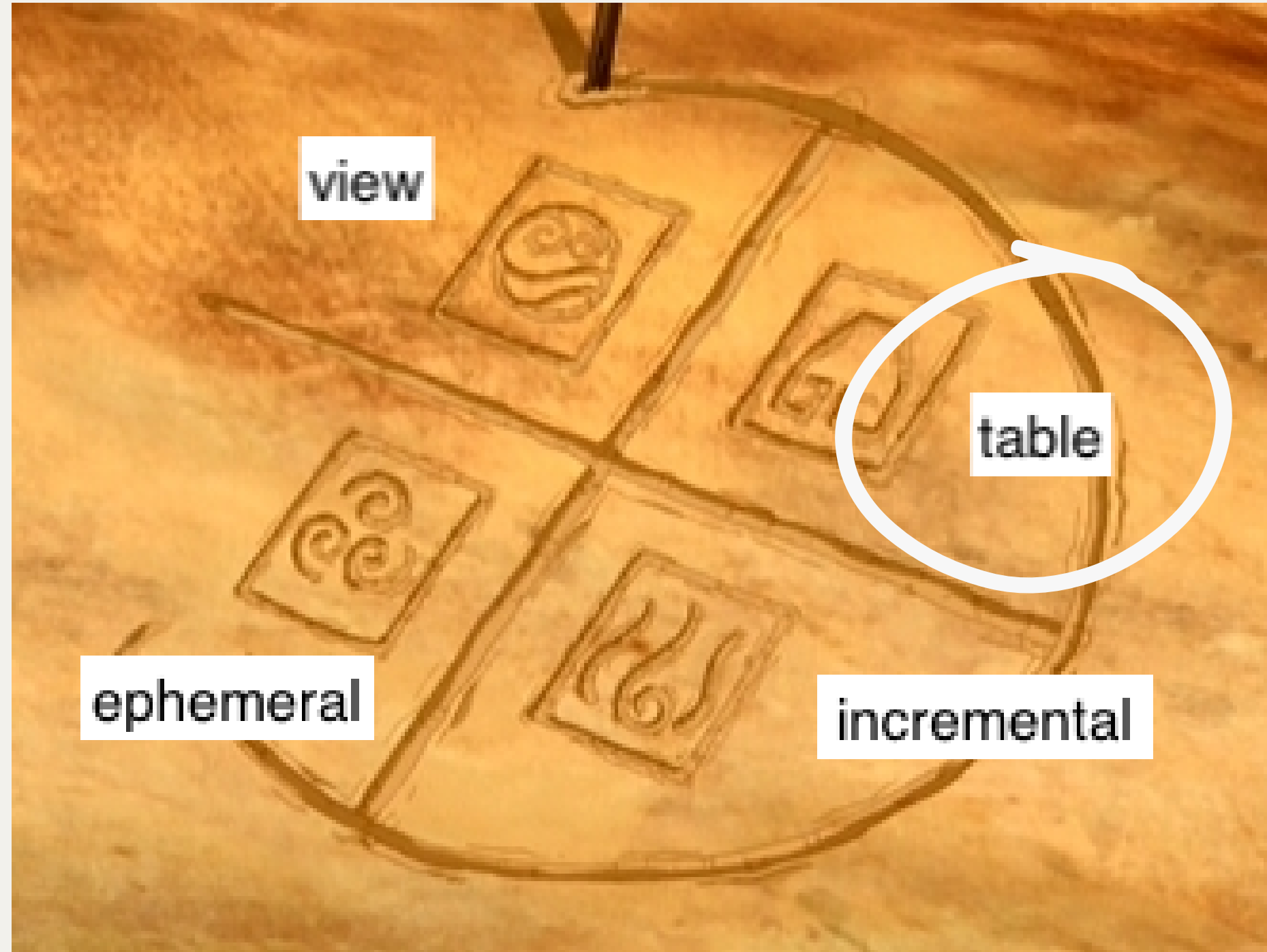
Exploring DBT Materialization Strategies



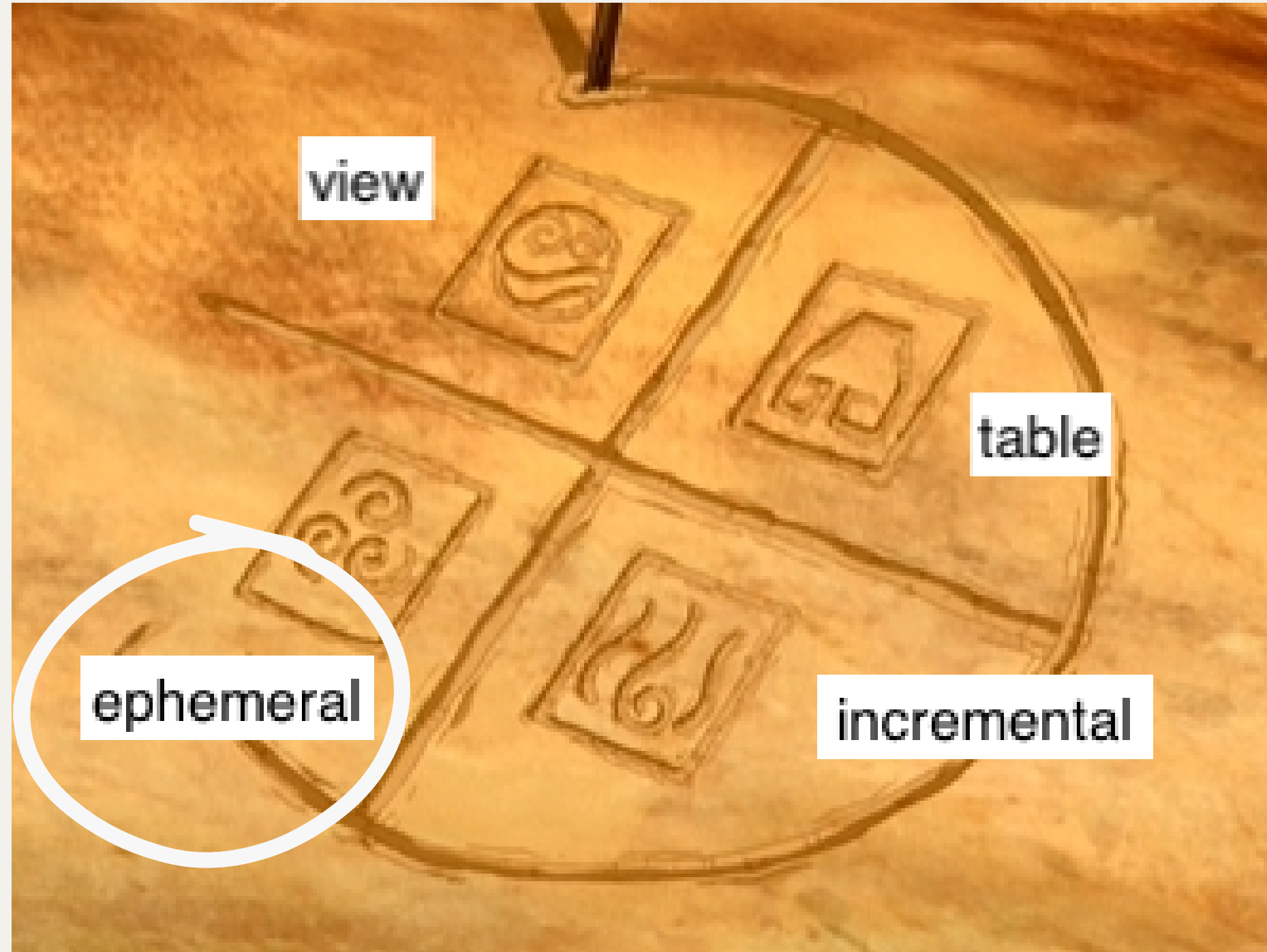
Exploring DBT Materialization Strategies



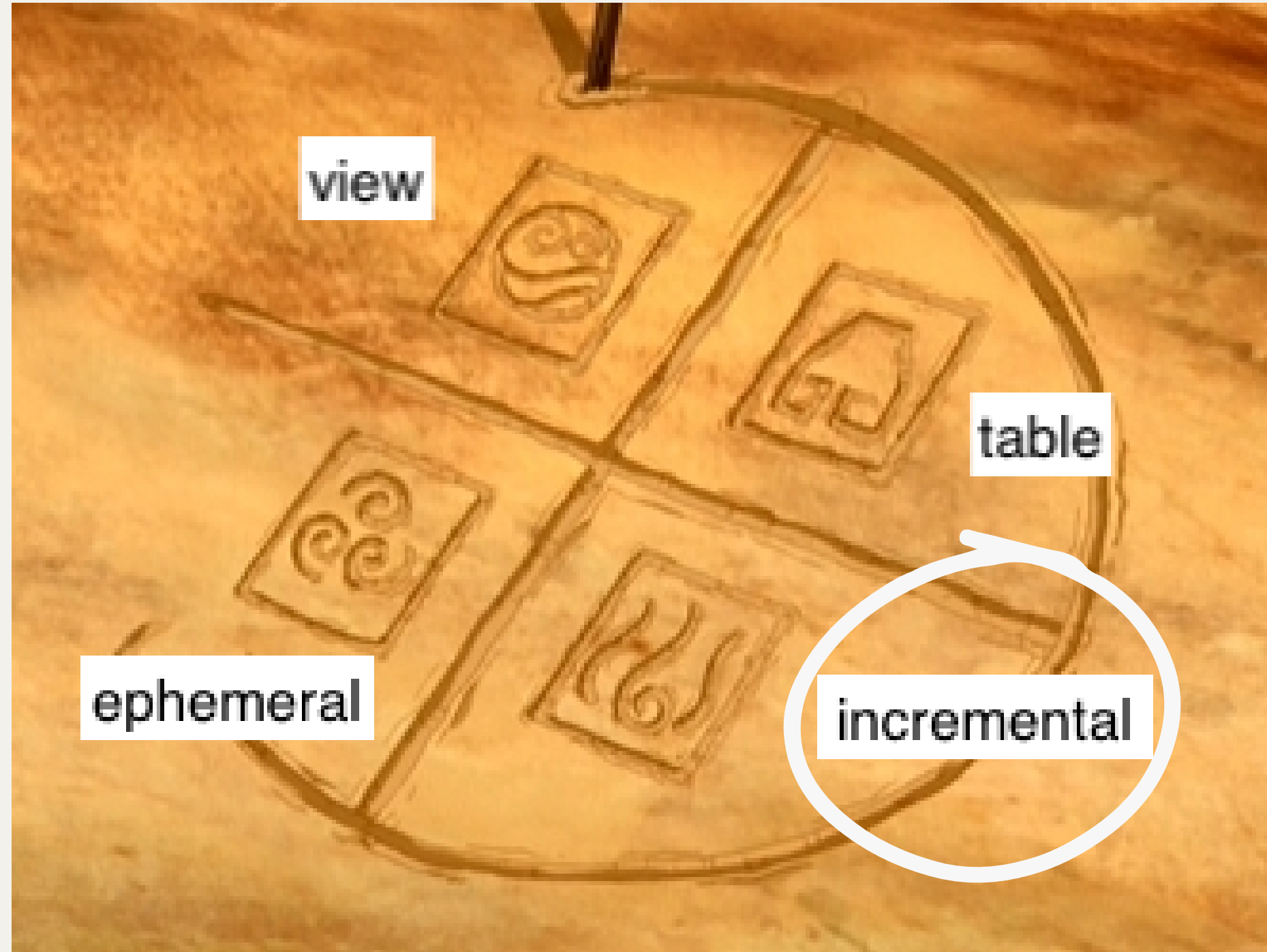
Exploring DBT Materialization Strategies



Exploring DBT Materialization Strategies



Exploring DBT Materialization Strategies



DBT Materialization – Incremental

cost_allocation.sql

```
{{ config(  
    materialized='incremental',  
    unique_key=['service_id', 'date']  
) }}
```

DBT Materialization – Incremental

dbt_project.yml

models:

cost_allocation:

enrichment:

+materialized: view

Config indicated by + and applies to all files under
models/enrichment/

DBT Materialization – Incremental

How can we ensure incremental update using sql?

DBT Materialization – Incremental

```
{% if is_incremental() %}  
-- this filter will only be applied on an incremental run  
where event_time > (select max(event_time) from {{ this }})  
{% endif %}
```

```
# is_incremental() macro
```

DBT Materialization – Incremental

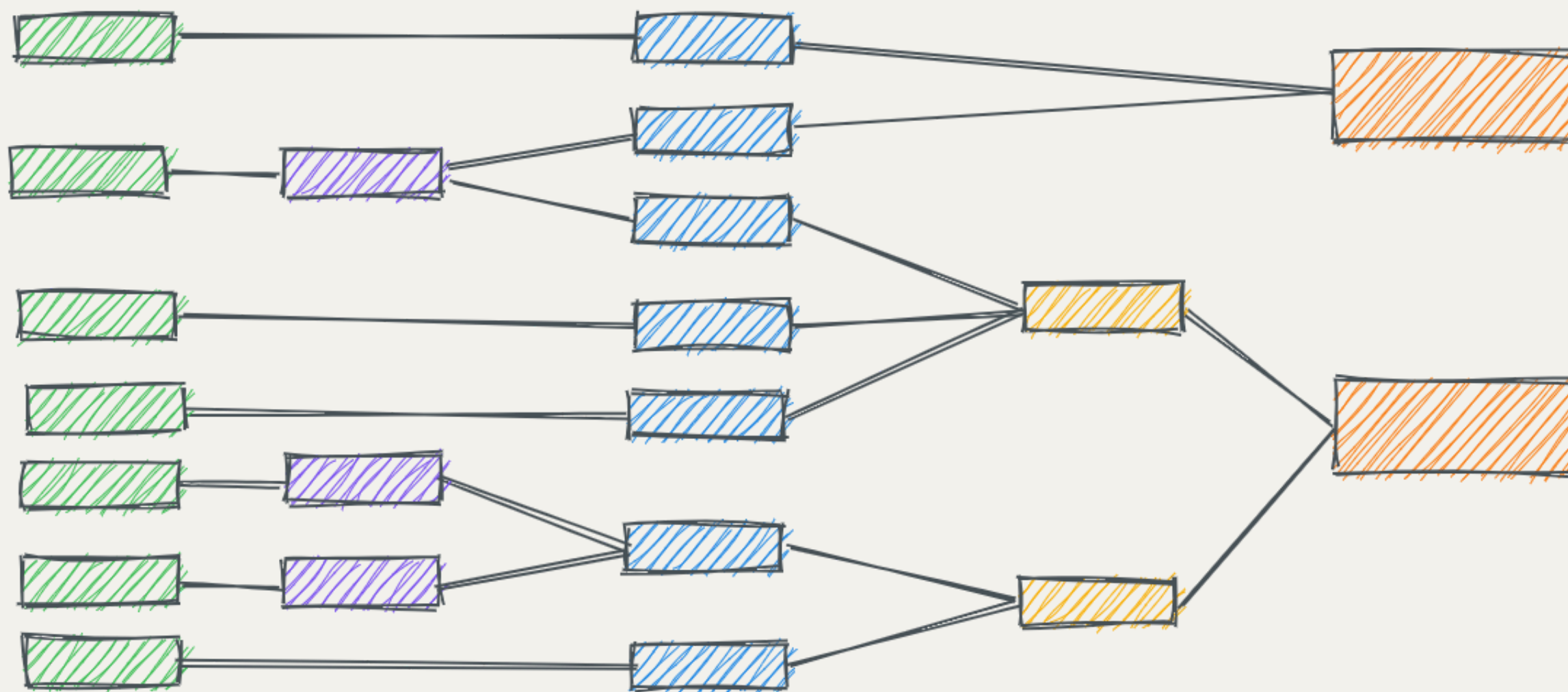
```
$ dbt run select cost_allocation_incremental --full-fresh
```

Exercise

Exercise Instructions:

1. Create a model 'cost_allocation_incremental' that aggregates monthly total usage costs **incrementally**
2. Think - how would you test that kind of model?

Modular Data Modeling



DBT Docs

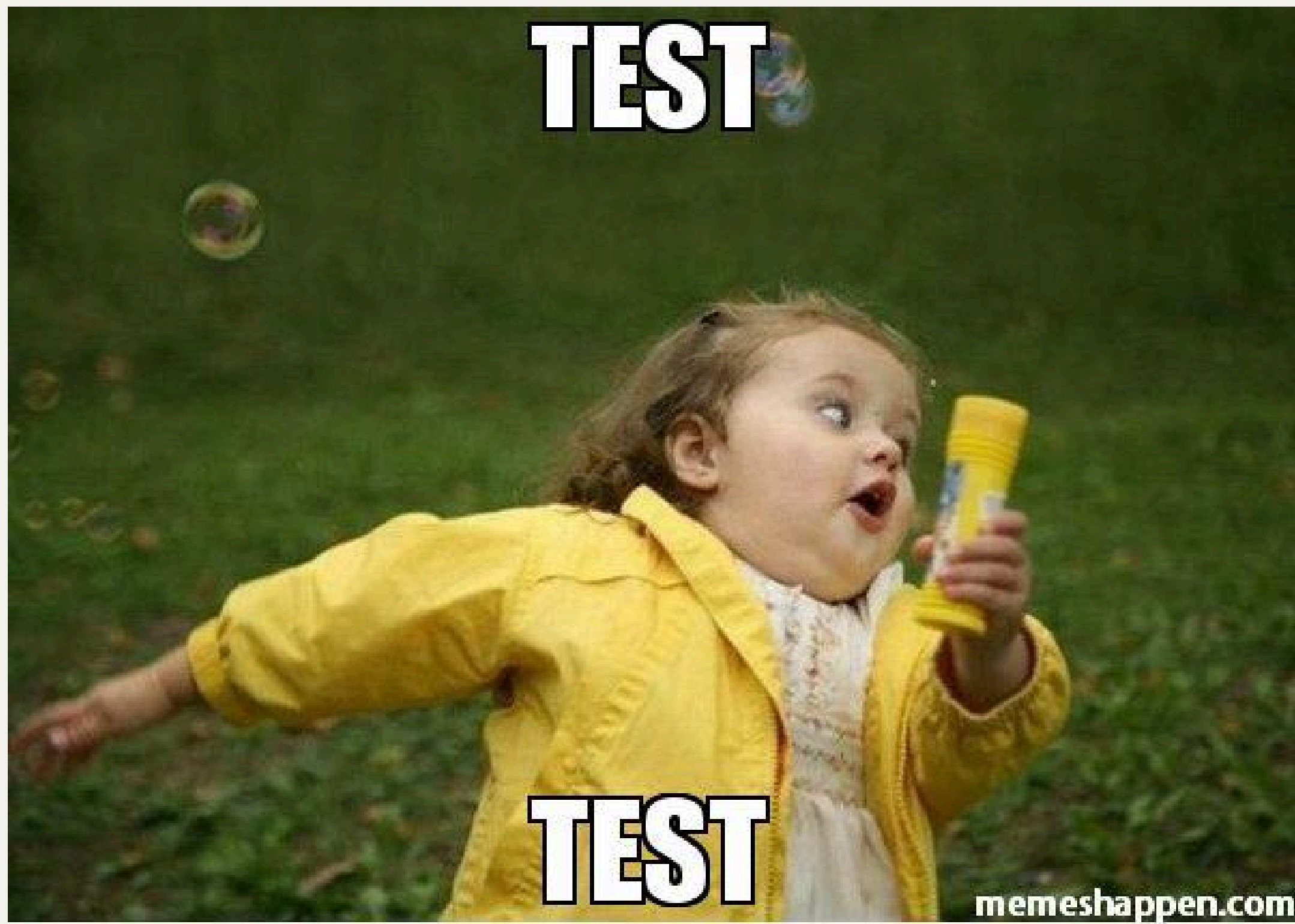
<https://models.opensource.observer/>

Exercise

Exercise Instructions:

1. Add descriptions to your models and columns in the schema file.
2. Run dbt docs generate.
3. Serve the documentation using dbt docs serve.

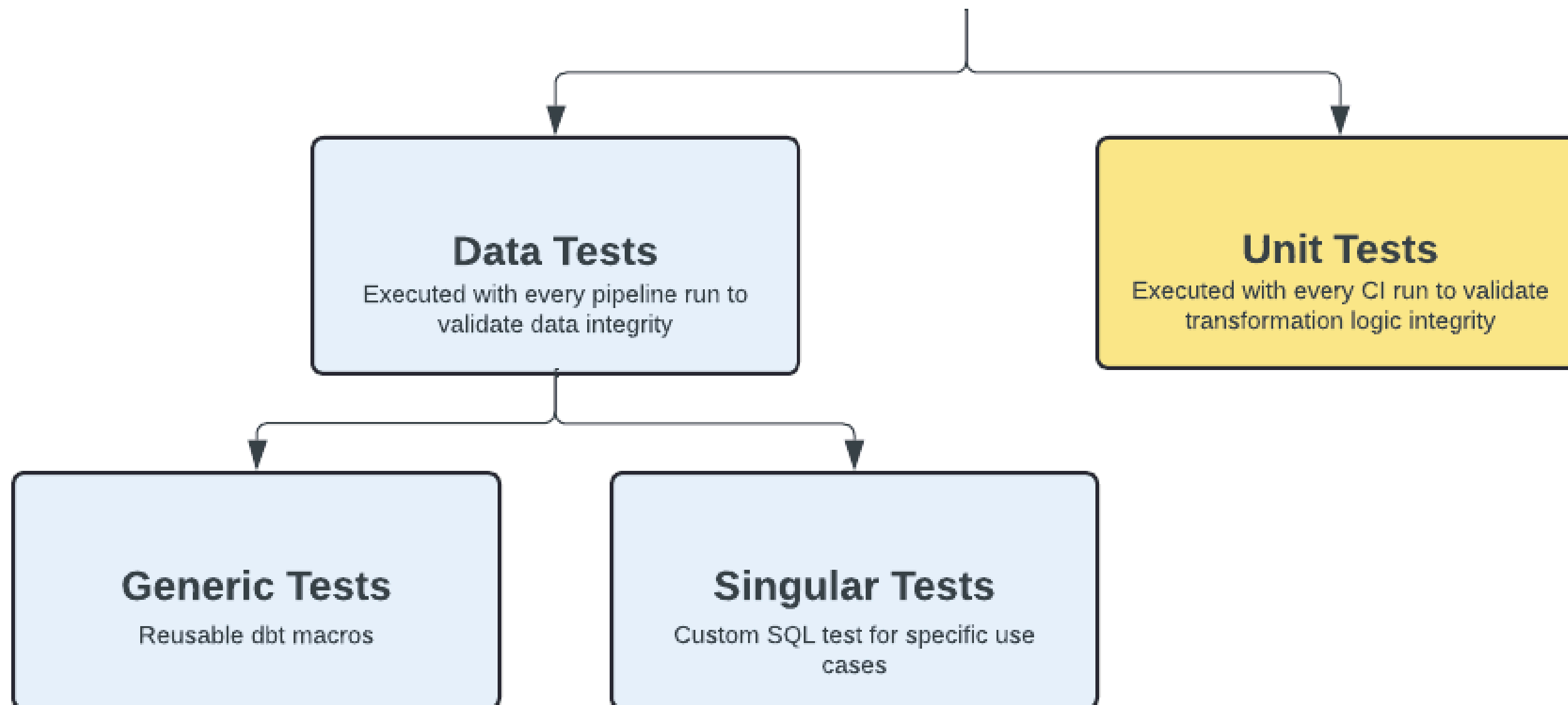
TEST



TEST

memeshappen.com

DBT test types



Exercise

Exercise Instructions:

1. Create tests for `cost_allocation_incremental` to check for valid `resource_type` values.
2. Run `dbt test` to validate your data.

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

Jinja Macros

Exercise

Create a macro that convert a string to upper case

1. Use this macro in your model to add a new column –
'upper_case_service_name' that contains service_name
value in upper case
2. Document your macro and provide examples of how to use it
in your models.

Packages

<https://hub.getdbt.com>

Quizzzzzzzz

What is the primary purpose of dbt?

- A) Data visualization
- B) Data transformation
- C) Data storage
- D) Data ingestion

What is the primary benefit of using dbt for data transformation?

- A) Improved data visualization
- B) Faster data ingestion
- C) Easier collaboration and version control
- D) More efficient data storage
- E) Reduced data processing costs

Which materialization strategy would you use to only process new records in a model?

- A) Table
- B) View
- C) Incremental
- D) Ephemeral

What does the Jinja templating language allow you to do in dbt?

- A) Write Python code
- B) Create dynamic SQL queries
- C) Generate documentation
- D) Load CSV files

WHY DBT?

- Open Source
- SQL
- Data Platform Agnostic
- Software development cycle features
 - VC, testing, documataationgeneration and data lineage

The END

