DATA+AI SUMMIT 2022

dbt + Machine Learning

What makes a great baton pass?

Sung Won Chung
Senior Solutions Architect, dbt Labs

ORGANIZED BY databricks

Who is this person?

Why should I listen to him?

I am Sung

Senior Solutions Architect @ dbt Labs

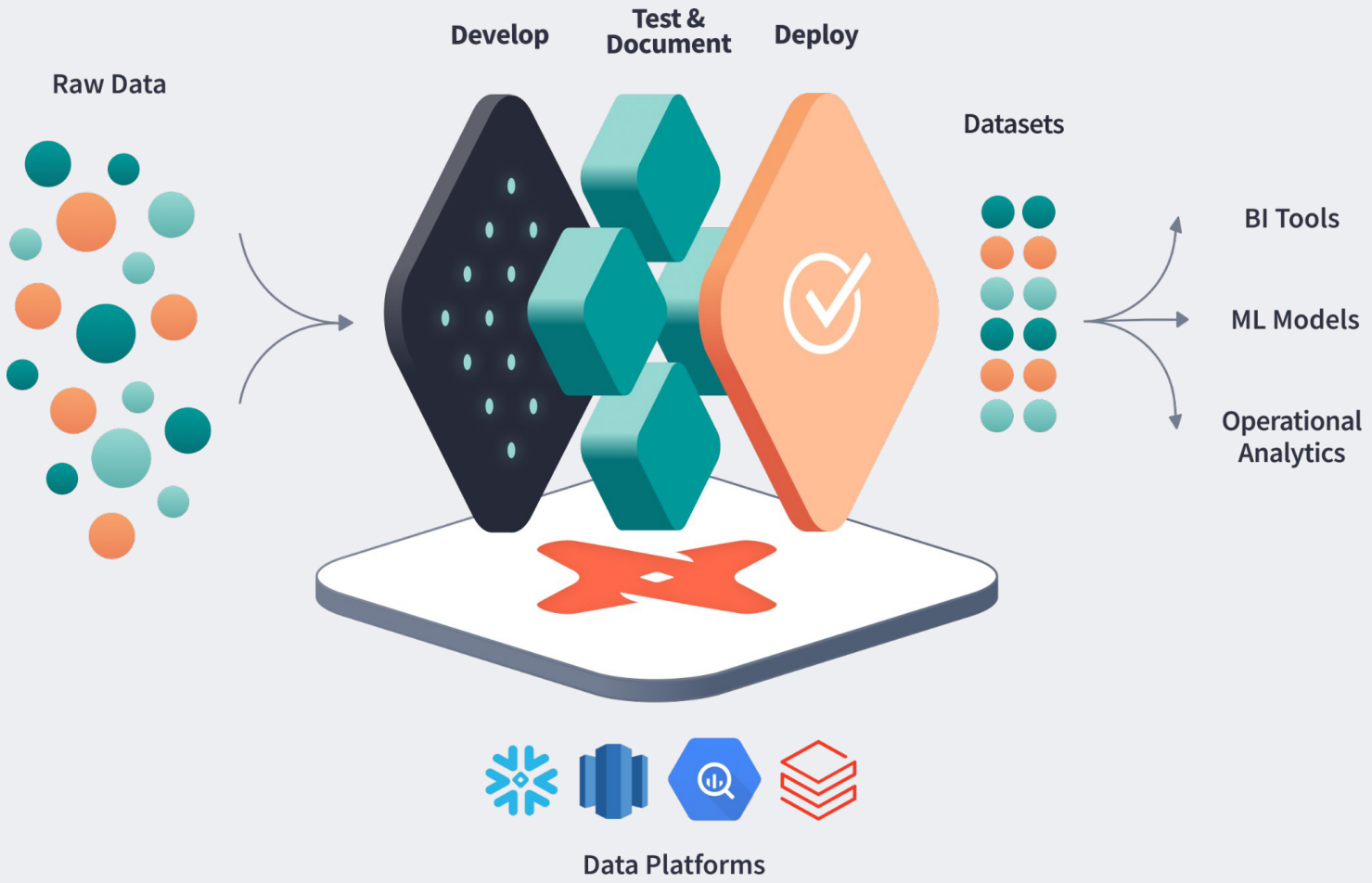
Fun fact-check out my spotify: here

Agenda

- What is dbt?
- Example Baton Pass
- Imagine the Future Together
- What do we do next?

What is dbt?

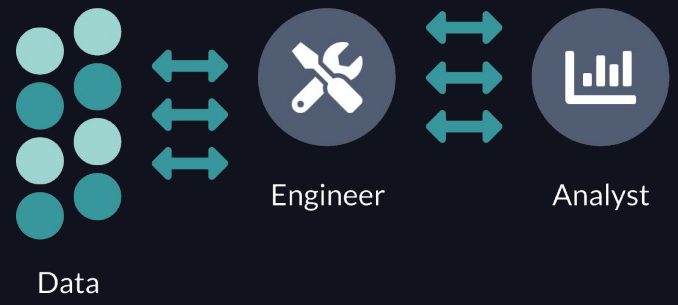
Develop, test, and deploy data products in Databricks



What is dbt?

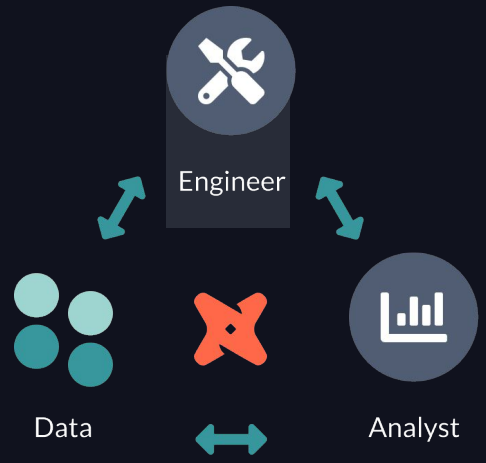
dbt solves the workflow problem in analytics

without dbt



- Engineers become bottlenecks
- Processes aren't repeatable
- No version control or governance

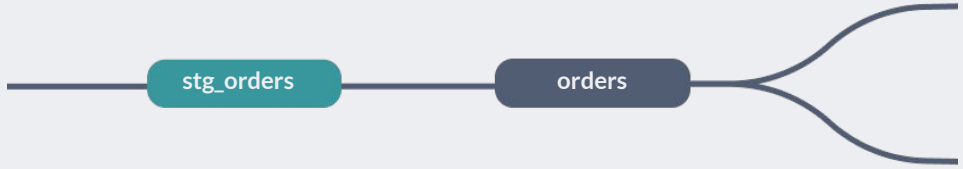
with dbt



- SQL makes analytics **collaborative**
- Modularity increases **speed**
- Testing, lineage, + access control **reduce risk**

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

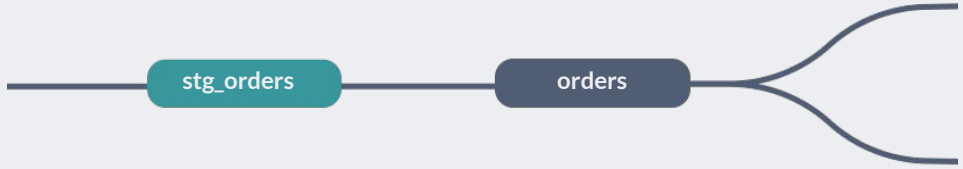
```
select *

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

Runs in the warehouse

```
create table dev.orders as (


select *

from 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

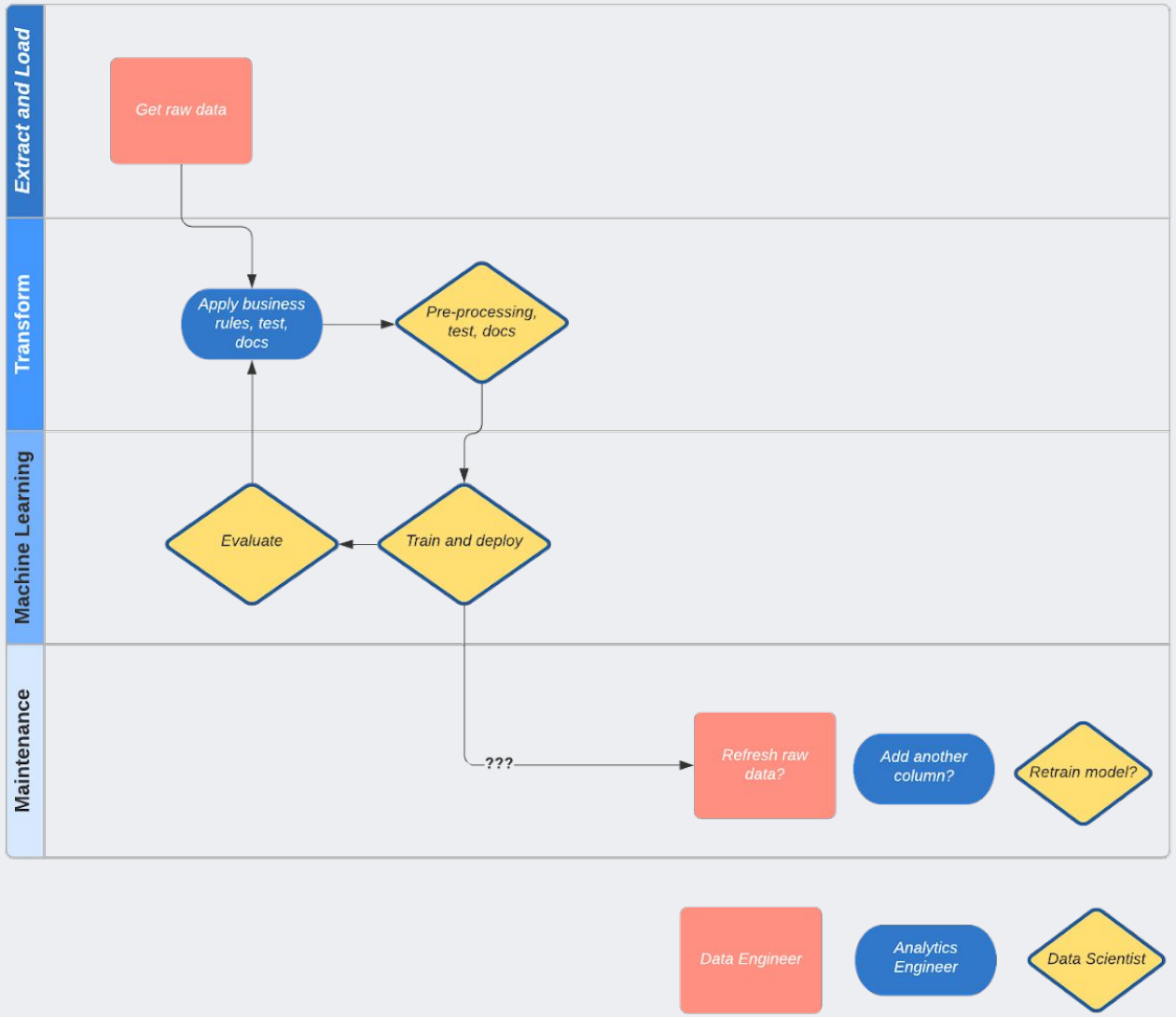
```
select *

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

Runs in the warehouse

```
create table prod.orders as (

select *

from prod.stg_orders
where is_deleted = false

);
```

Example Baton Pass Today

Where is the breakdown happening?



“Who makes data AND machine learning pipelines maintainable over time when things break?”

“How do we get people to use our work?"

Probably You

ML Engineer & Analytics Engineer

Imagine the Future: Together

How do we address the breakdown?

Questions our Industry is Wrestling Through

Do we need a better developer notebook?

Should we increase the SQL surface area to build ML models?

Should we leave that to non-SQL interfaces(Python/Scala/etc.)?

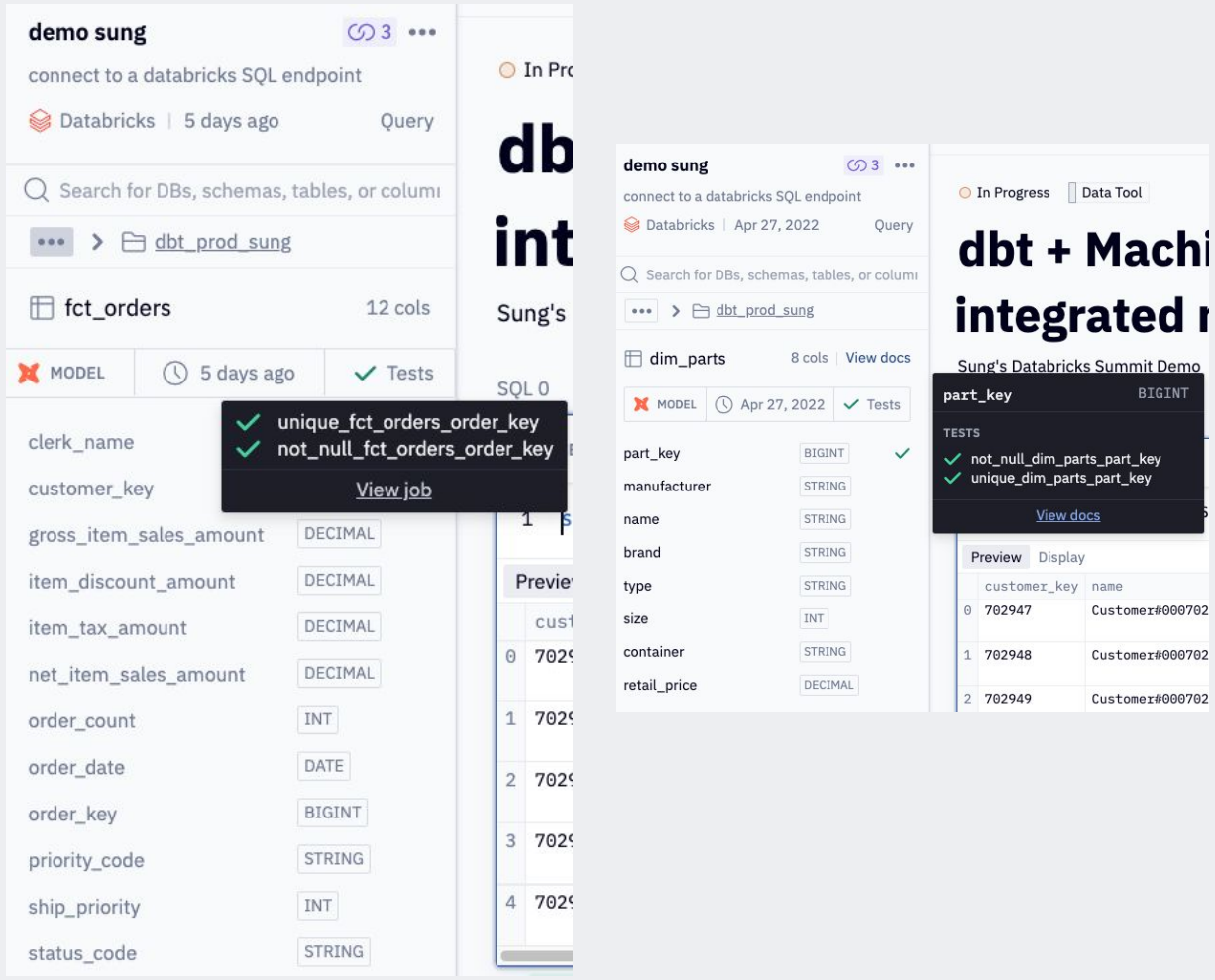
Does this have to be an either/or future?

A better dbt-integrated notebook experience

Keep evolving what most practitioners are doing today

- I know at a glance if data quality is driving errors

Recorded Demo: here



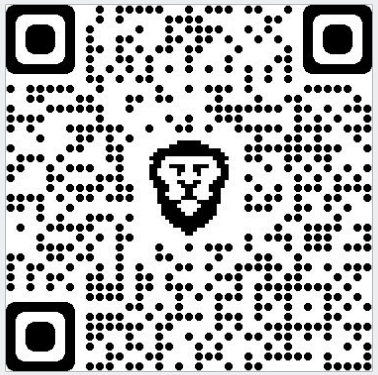
Machine Learning comes to SQL

Unite ML and Analytics Engineers on a common interface

SQL is our entrypoint together

Integrate machine learning configs within my dbt project

Recorded Demo: here



For example, the following is an example of including your configuration in a schema.yml file:

```yaml
models:
  - name: customer_churn
    description: "historic customer churn information"
    meta:
      continual:
        type: "Model"
        index: "ID"
        target: "churn"
```

However, we could similarly define this in the customer_churn.sql file:

```
{{
  config(
    meta = {
      continual: {
        'type': 'Model',
        'index': 'ID',
        'target': 'churn',
      }
    }
  )
}}

SELECT ...
```

Make dbt integrate with multi-lingual support

Baton pass across languages

A ref statement would mean the same thing to both my ML engineer and me

We would work in the same dbt project for the entire workflow, not just part of it

Recorded Demo: here

We start by importing fal into your project

```
from fal import FalDbt
```

Then instantiate a new FalDbt project with the dbt project information:

```
faldbt = FalDbt(profiles_dir="~/.dbt", project_dir="../my_project")

faldbt.list_sources()
# [['results', 'ticket_data_sentiment_analysis']]

faldbt.list_models()
# {
#   'zendesk_ticket_metrics': <RunStatus.Success: 'success'>,
#   'stg_o3values': <RunStatus.Success: 'success'>,
#   'stg_zendesk_ticket_data': <RunStatus.Success: 'success'>,
#   'stg_counties': <RunStatus.Success: 'success'>
# }
```

Reference these objects as you would in a regular fal script, from the `faldbt` object:

```
sentiments = faldbt.source('results', 'ticket_data_sentiment_analysis')
# pandas.DataFrame
tickets = faldbt.ref('stg_zendesk_ticket_data')
# pandas.DataFrame
```

What outcomes matter?

Let's not miss the forest for the trees

When something goes wrong, it's clearer where to start solving

People are excited about baton passing work back and forth, less playing hot potato

People have deep pride in a predictive analytics workflow that works

Everyone gets to work on more interesting problems than clunkily fixing a machine learning pipeline across 5 browser tabs

“And most importantly, people use our collective gosh darn work to make real decisions”

Probably You

ML Engineer & Analytics Engineer

What's next?

You tell me

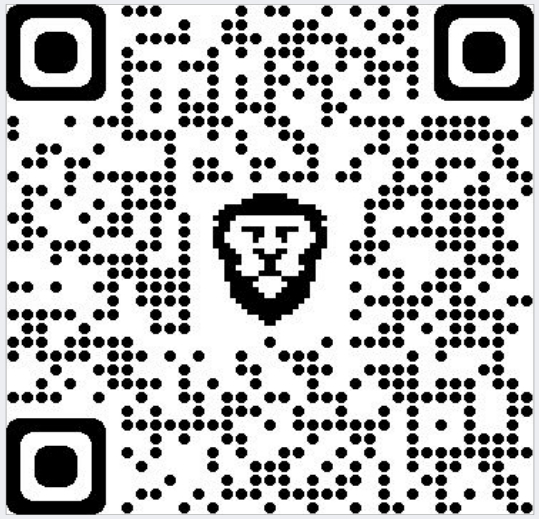
I'm less interested in the tools than understanding if this problem is one we can agree is painful enough to solve in the first place.

- Is this your story?
- Which future excites you?
- Which scares you?
- Do you care?

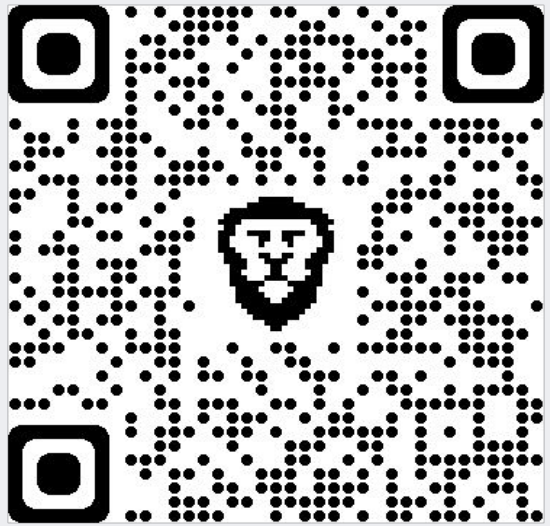
Fun Videos to Watch!

Bookmark these gems ;)

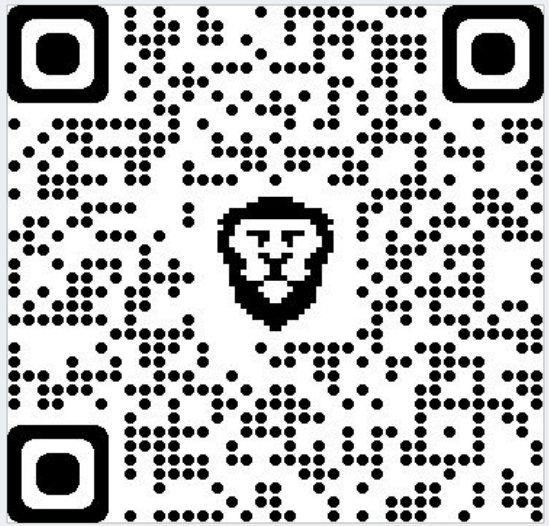
dbt Cloud + Airflow Tutorial



dbt Cloud + Airflow Tutorial: Rerun from point of failure



dbt Cloud Fresh Rebuilds Tutorial



DATA+AI
SUMMIT 2022

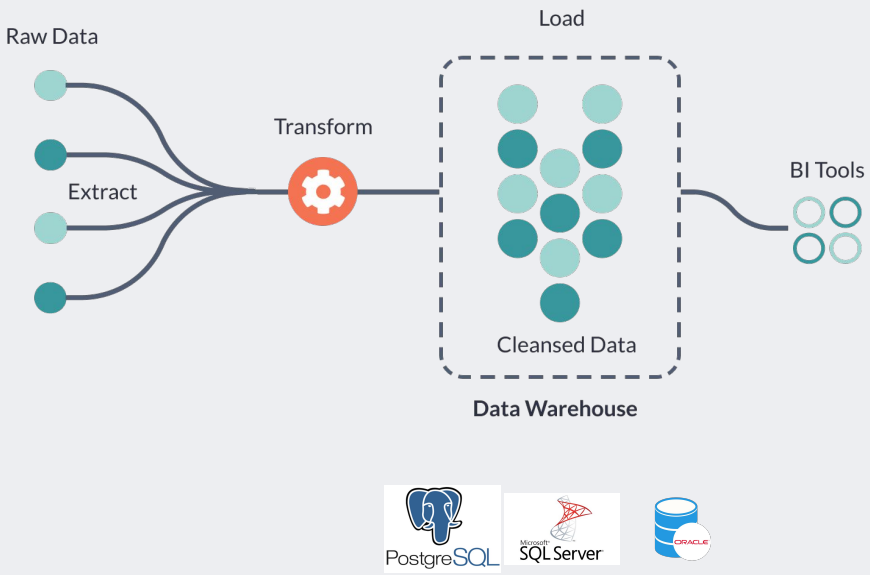
Thank you

Sung Won Chung
Senior Solutions Architect, dbt Labs

What is dbt?

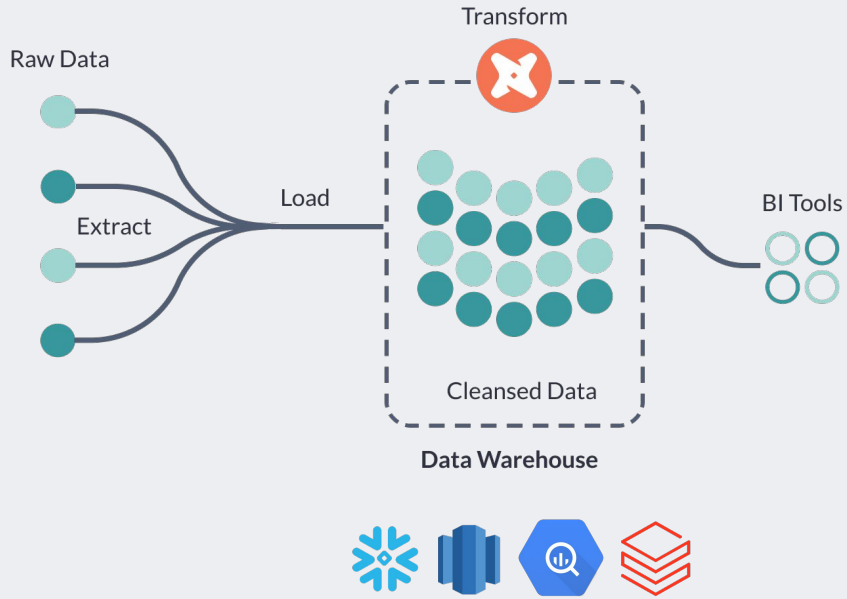
Evolution of the modern data stack

Legacy E-T-L



- High storage and compute costs
- Disjointed analytics workflows

Modern E-L-T



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