# Building A Simple Data Stack

## Project Brief

This project demonstrate how to implement a modern data stack, build data pipelines, machine learning and reporting capabilities using a variety of solutions.

BRIAN GWAYI
Independent Data Lead & Engineer

# First Things First !!!

## Five Key Questions

I. Where is our data? Source
II. Where do we consolidate our data? Storage
III. How will we get it there? Ingestion
IV. How will we clean it up? Transformation
V. How will we analyze it? Reporting

BRIAN GWAYI
Independent Data Lead & Engineer

# Data Stack Architecture Design

Where is our data?

PostgreSQL

MySQL

How will we get it there?

Airflow

+

Python

Where do we consolidate our data?

Amazon Redshift

Transformation

dbt

Raw

Model

Model

How do we analyze it?

tableau

looker

Power BI

# Ultimate End Goal

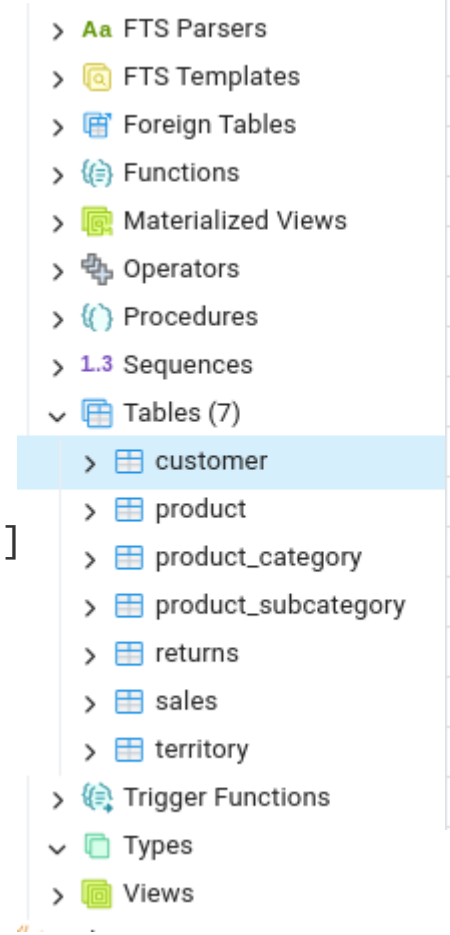
**Data + Insights + Action**
**= Actionable Insights**

| Data | Insight | Action |
| --- | --- | --- |
| **What** happened/ will happen? | **Why** did it happened/ will it happen? | **What** do we do? |

# Where

## is Our Data?

```
Source : PostgresSQL
Schema : Public
Database Name: adw_db
Tables Count : 7
Tables : [customer,
          product,
          product_category,
          return,
          sales,
          territory,
          product_subcategory]
```




Data Output | Messages | Notifications

| | orderdate date | stockdate date | ordernumber character varying (255) | productkey integer | customerkey integer |
|---|---|---|---|---|---|
| 1 | 2022-01-01 | 2021-12-13 | SO61285 | 529 | 23791 |
| 2 | 2022-01-01 | 2021-09-24 | SO61285 | 214 | 23791 |
| 3 | 2022-01-01 | 2021-09-04 | SO61285 | 540 | 23791 |
| 4 | 2022-01-01 | 2021-09-28 | SO61301 | 529 | 16747 |
| 5 | 2022-01-01 | 2021-10-21 | SO61301 | 377 | 16747 |
| 6 | 2022-01-01 | 2021-10-23 | SO61301 | 540 | 16747 |
| 7 | 2022-01-01 | 2021-09-04 | SO61269 | 215 | 11792 |
| 8 | 2022-01-01 | 2021-10-21 | SO61269 | 229 | 11792 |
| 9 | 2022-01-01 | 2021-10-24 | SO61286 | 528 | 11530 |
| 10 | 2022-01-01 | 2021-09-27 | SO61286 | 536 | 11530 |
| 11 | 2022-01-01 | 2021-10-23 | SO61298 | 530 | 18155 |
| 12 | 2022-01-01 | 2021-12-02 | SO61298 | 214 | 18155 |
| 13 | 2022-01-01 | 2021-12-15 | SO61298 | 223 | 18155 |
| 14 | 2022-01-01 | 2021-10-01 | SO61310 | 538 | 13541 |
| 15 | 2022-01-01 | 2021-11-08 | SO61310 | 584 | 13541 |

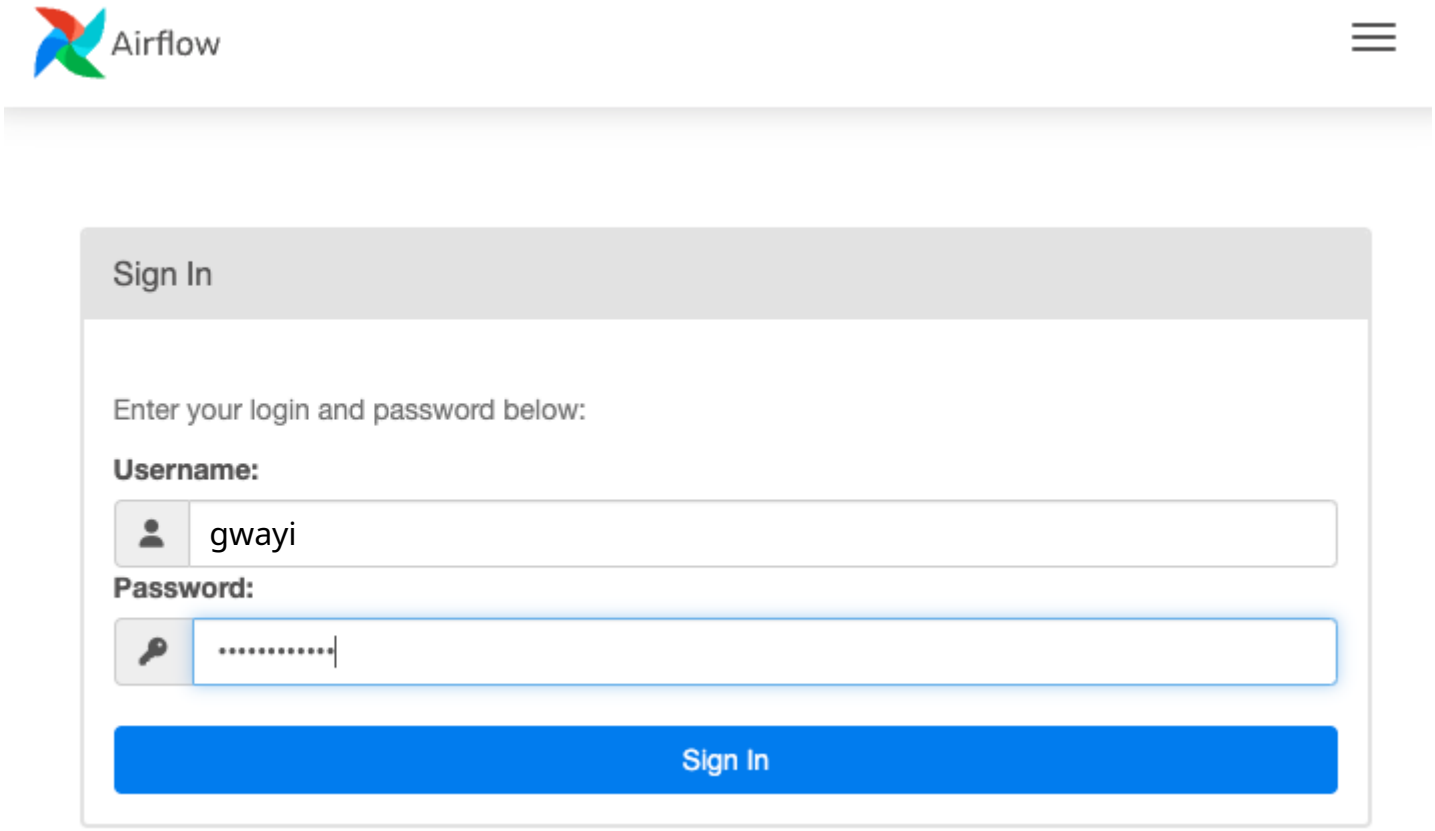
# How do we ingest Our Data?

```
Ingestion : python script
Orchestration : Apache Airflow
```

## Apache Airflow Setup

```
Terminal

$ python3 -m venv airflow-env
$ source airflow-env/bin/activate
$ export AIRFLOW_HOME=~/airflow
$ pip install apache-airflow
$ airflow db init
$ airflow webserver -p 8080
$ airflow sheduler
```

Apache Airflow Webserver UI



# How do we

## ingest Our Data?

```
Ingestion : python script
Orchestration : Apache Airflow
```

### ELT Python Script

```
pip install google-cloud-bigquery
pip install --upgrade snowflake-connector-python

# importing libraries

from airflow.decorators import dag, task
from datetime import datetime, timedelta
from google.cloud import bigquery
import pandas as pd
import psycopg2
```

### Insatiate a DAG

```
args{
    ”owner”:”gwayi”,
    "retries": 1,
    "retry_delay":timedelta(minutes=5)
    }

@dag(
    default_arguments = args
    Schedule=timedelta(minutes=30),
    start_date=datetime(2024, 7, 29),
    catchup=False,
    tags=['Team B']
    )
```

# How do we

## ingest Our Data?

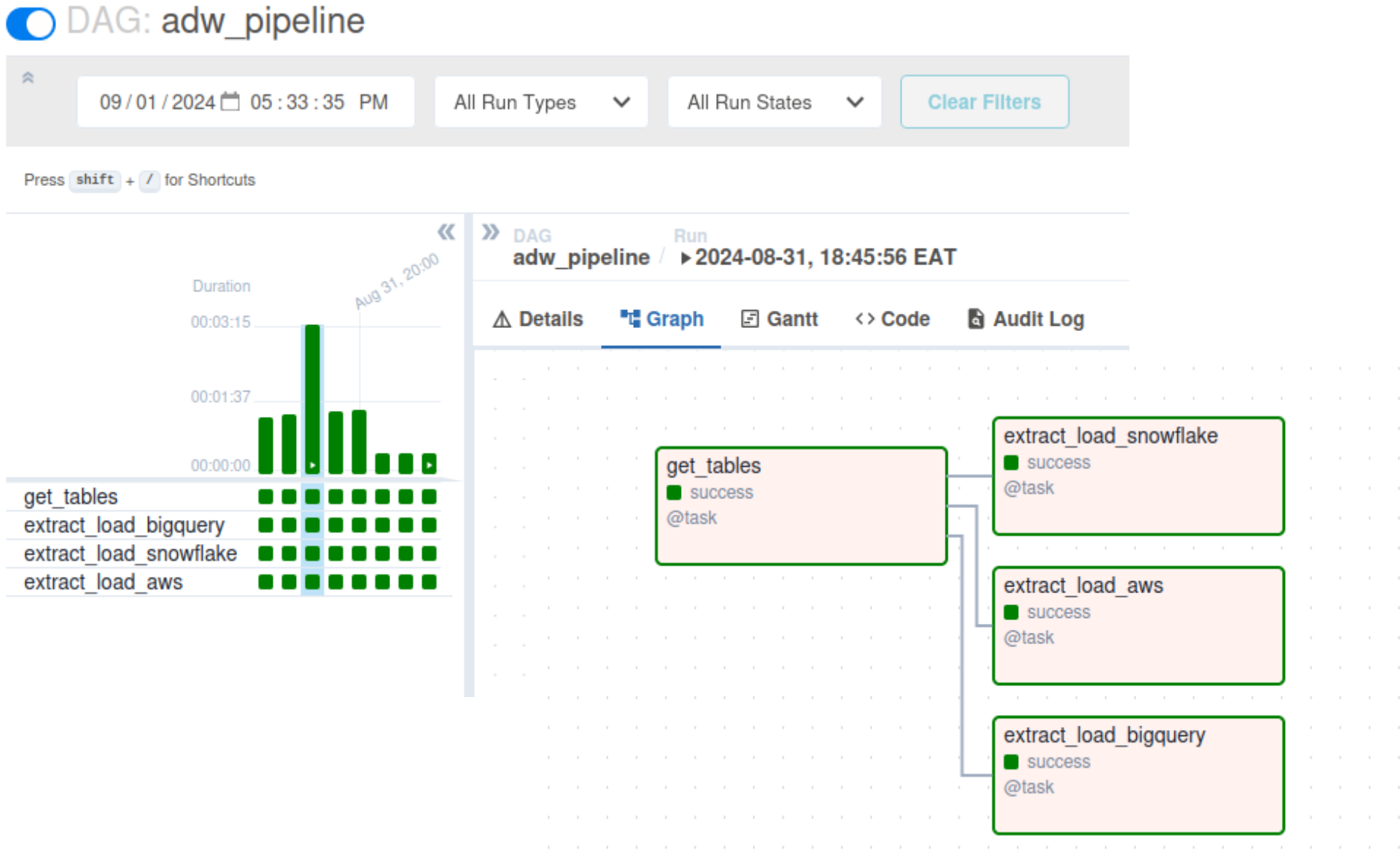
Ingestion : Python Script
Orchestration : Apache Airflow

### Task I ( Get Tables )

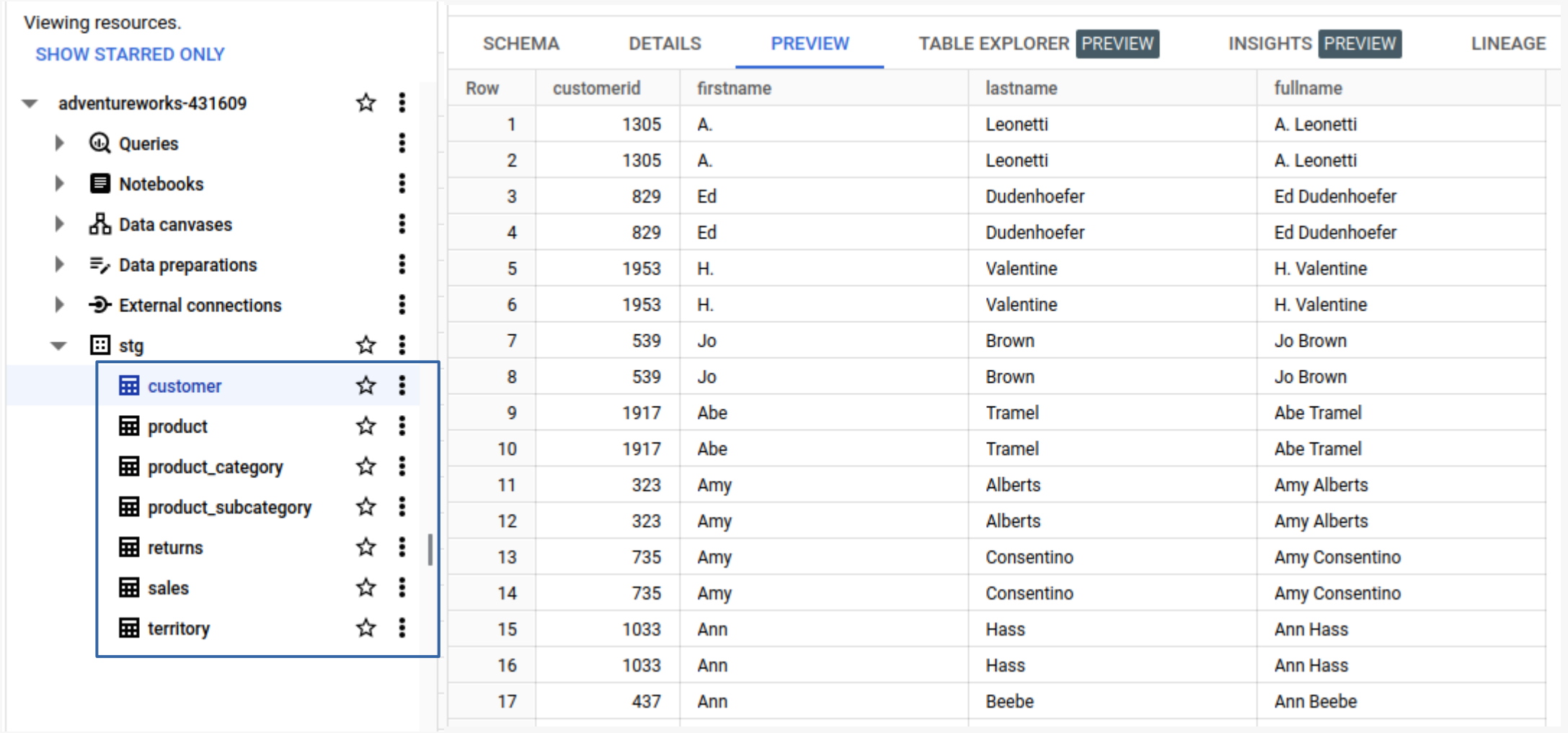
```
@task()
def get_tables():
"""extract list of tables in public schema"""

  try:
    cursor.execute(
         f"""SELECT table_name
         FROM information_schema.tables
         WHERE table_schema = 'public'"""
          )

  tbls = [x[0] for x in cursor.fetchall()]
```

### Task II ( Extract_Load )

```
@task()
      def extract|load_bigquery(tbls, conn):
"""loop through tbls then extract & load"""

     client = bigquery.Client()
     job_config = bigquery.LoadJobConfig(
     write_disposition="WRITE_TRUNCATE")

      for tbl in tbls:
      table_id = f"adventureworks-431609.stg.{tbl}"
      sql = f"SELECT * FROM {tbl} WHERE
      updated_at >= {ds}'"
      df = pd.read_sql(sql, conn)

      job = client.load_table_from_dataframe(
      df, table_id, job_config=job_config)
      job.result()

    get_tables = get_tables()
    extract_load = extract_load(get_tables)
```


DAG: adw_pipeline
09 / 01 / 2024 05 : 33 : 35 PM
All Run Types
All Run States
Clear Filters
Press shift + / for Shortcuts
Duration
Aug 31, 20:00
00:03:15
00:01:37
00:00:00
get_tables
extract_load_bigquery
extract_load_snowflake
extract_load_aws
DAG
adw_pipeline
Run
2024-08-31, 18:45:56 EAT
Details
Graph
Gantt
Code
Audit Log
get_tables
success
@task
extract_load_snowflake
success
@task
extract_load_aws
success
@task
extract_load_bigquery
success
@task

# Tables Loaded: BigQuery




| Row | customerid | firstname | lastname | fullname |
|---|---|---|---|---|
| 1 | 1305 | A. | Leonetti | A. Leonetti |
| 2 | 1305 | A. | Leonetti | A. Leonetti |
| 3 | 829 | Ed | Dudenhoefer | Ed Dudenhoefer |
| 4 | 829 | Ed | Dudenhoefer | Ed Dudenhoefer |
| 5 | 1953 | H. | Valentine | H. Valentine |
| 6 | 1953 | H. | Valentine | H. Valentine |
| 7 | 539 | Jo | Brown | Jo Brown |
| 8 | 539 | Jo | Brown | Jo Brown |
| 9 | 1917 | Abe | Tramel | Abe Tramel |
| 10 | 1917 | Abe | Tramel | Abe Tramel |
| 11 | 323 | Amy | Alberts | Amy Alberts |
| 12 | 323 | Amy | Alberts | Amy Alberts |
| 13 | 735 | Amy | Consentino | Amy Consentino |
| 14 | 735 | Amy | Consentino | Amy Consentino |
| 15 | 1033 | Ann | Hass | Ann Hass |
| 16 | 1033 | Ann | Hass | Ann Hass |
| 17 | 437 | Ann | Beebe | Ann Beebe |

# How do we transform Our Data?

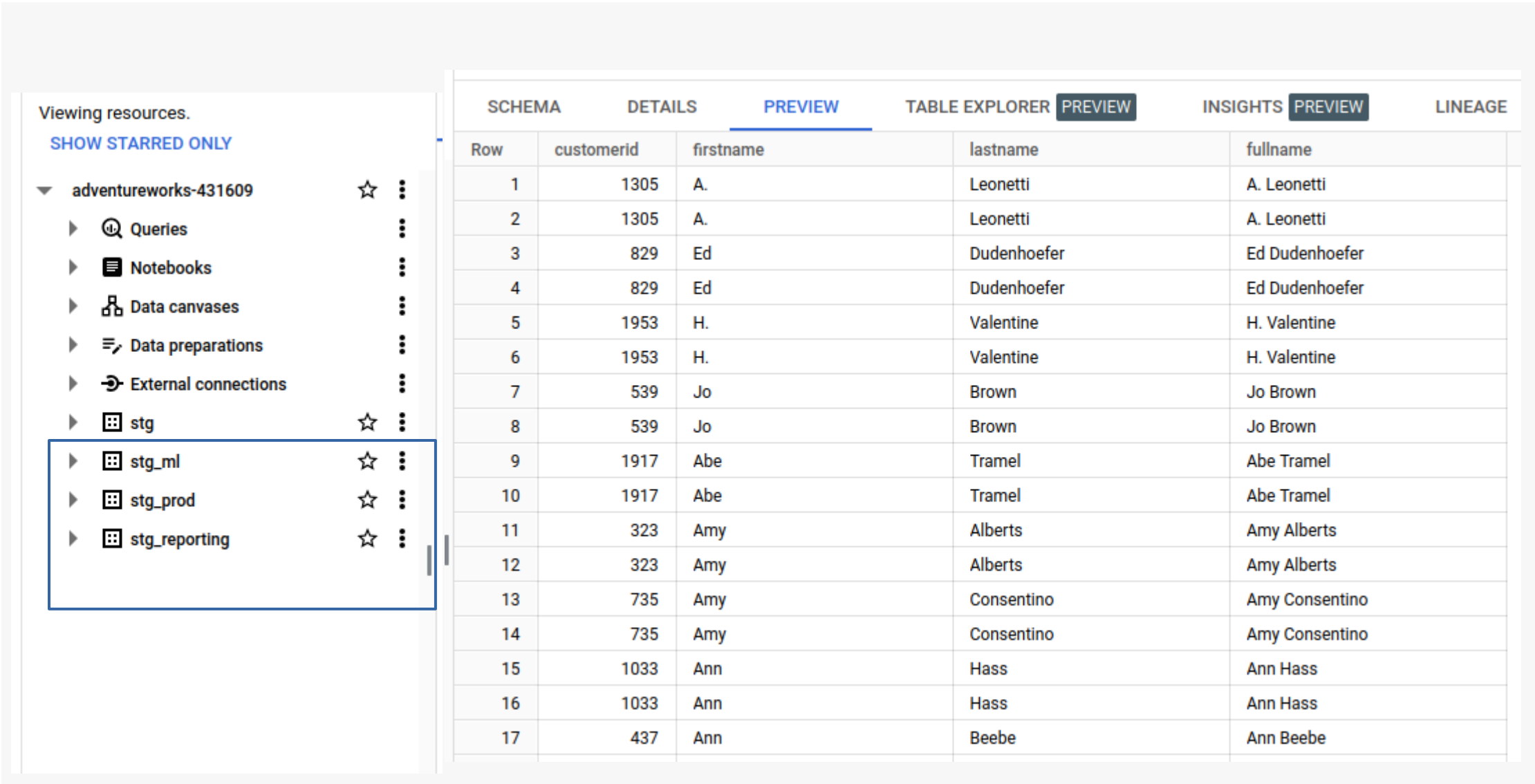
```
Transformation : dbt
Orchestration : Apache Airflow
Models 3 : [production,
            machine learning,
            Reporting]
```

## dbt Setup

```
pip install google-cloud-bigquery
pip install --upgrade snowflake-connector-python
```

## Insatiate a DAG

```
args
```




| Row | customerid | firstname | lastname | fullname |
|---|---|---|---|---|
| 1 | 1305 | A. | Leonetti | A. Leonetti |
| 2 | 1305 | A. | Leonetti | A. Leonetti |
| 3 | 829 | Ed | Dudenhoefer | Ed Dudenhoefer |
| 4 | 829 | Ed | Dudenhoefer | Ed Dudenhoefer |
| 5 | 1953 | H. | Valentine | H. Valentine |
| 6 | 1953 | H. | Valentine | H. Valentine |
| 7 | 539 | Jo | Brown | Jo Brown |
| 8 | 539 | Jo | Brown | Jo Brown |
| 9 | 1917 | Abe | Tramel | Abe Tramel |
| 10 | 1917 | Abe | Tramel | Abe Tramel |
| 11 | 323 | Amy | Alberts | Amy Alberts |
| 12 | 323 | Amy | Alberts | Amy Alberts |
| 13 | 735 | Amy | Consentino | Amy Consentino |
| 14 | 735 | Amy | Consentino | Amy Consentino |
| 15 | 1033 | Ann | Hass | Ann Hass |
| 16 | 1033 | Ann | Hass | Ann Hass |
| 17 | 437 | Ann | Beebe | Ann Beebe |

# PROJECTS

01

**Storage/Data Warehouse**
Implementing Data Warehouse Solutions
Google BigQuery | Snowflake | AWS Redshift | Oracle ADW

02

**Ingestion**
Developing Data Pipelines
Python | Airflow | Airbyte | dagster | Prefect

03

**Transformation**
Setting up dbt
Building Models

04

**Reporting**
Looker | Tableau | Power BI

05

**Machine Learning**
Building ML Models