Introducing the dbt case study

CASE STUDY: BUILDING E-COMMERCE DATA MODELS WITH DBT

Susan Sun
Freelance Data Scientist

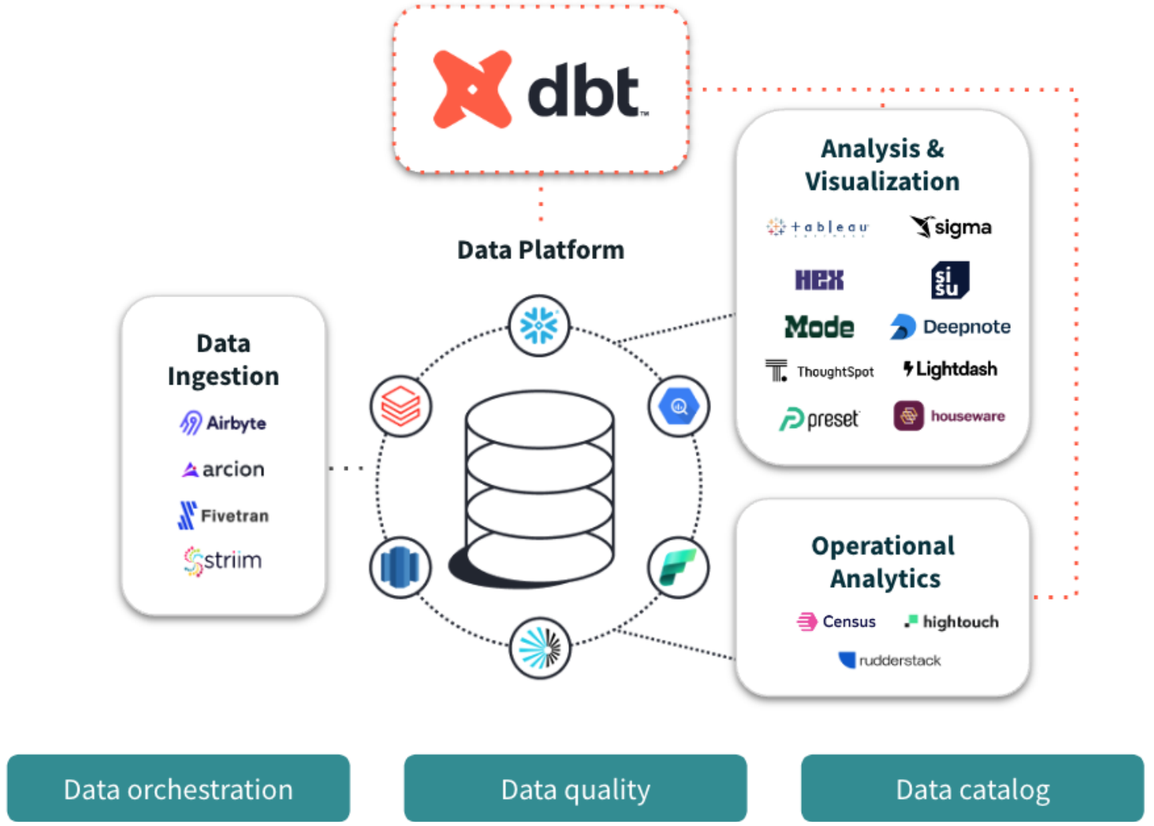
datacamp

Why dbt?

dbt spans many sectors:

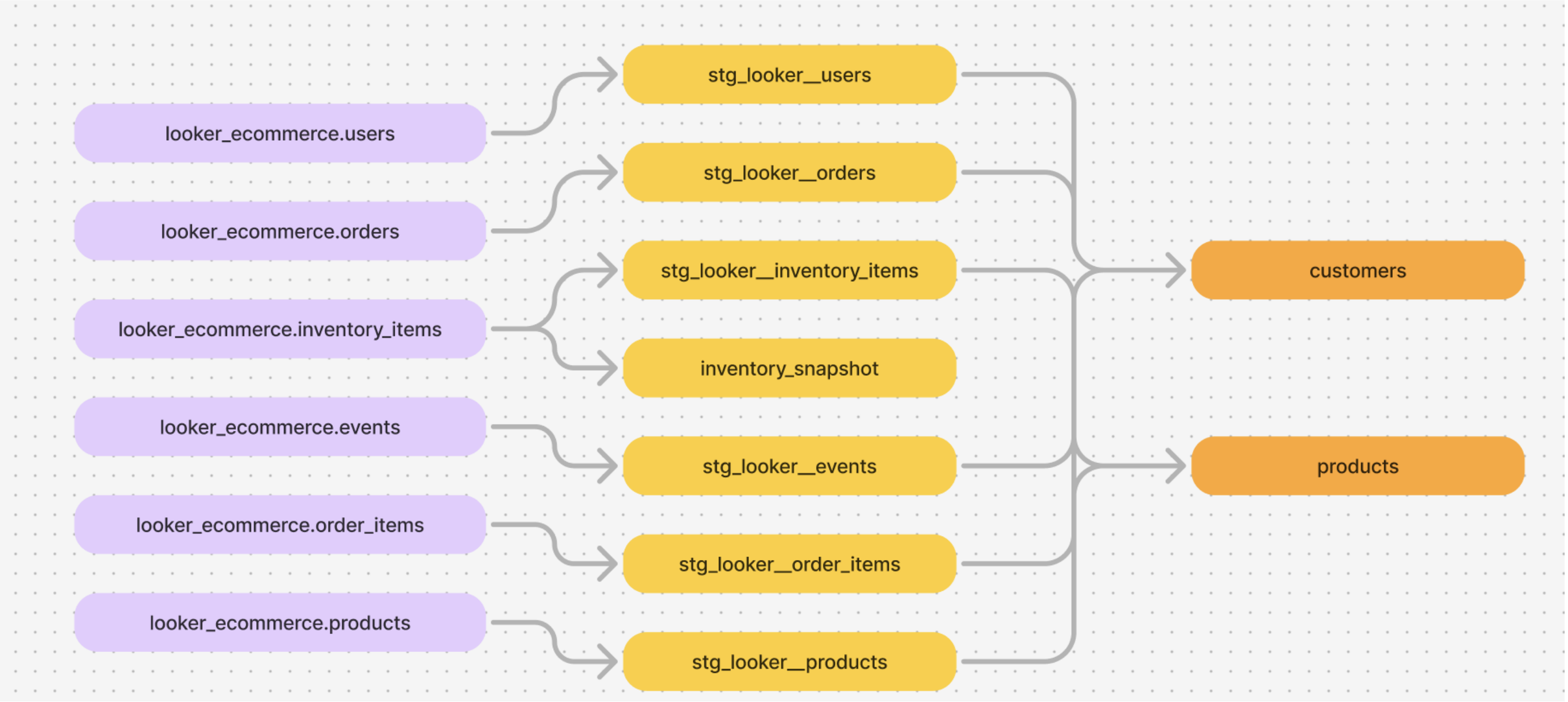
- banking & financial services
- e-commerce
- education
- government & civic tech
- healthcare
- renewable energy
- transportation & logistics
- ... etc

dbt integrates well with the modern stack:



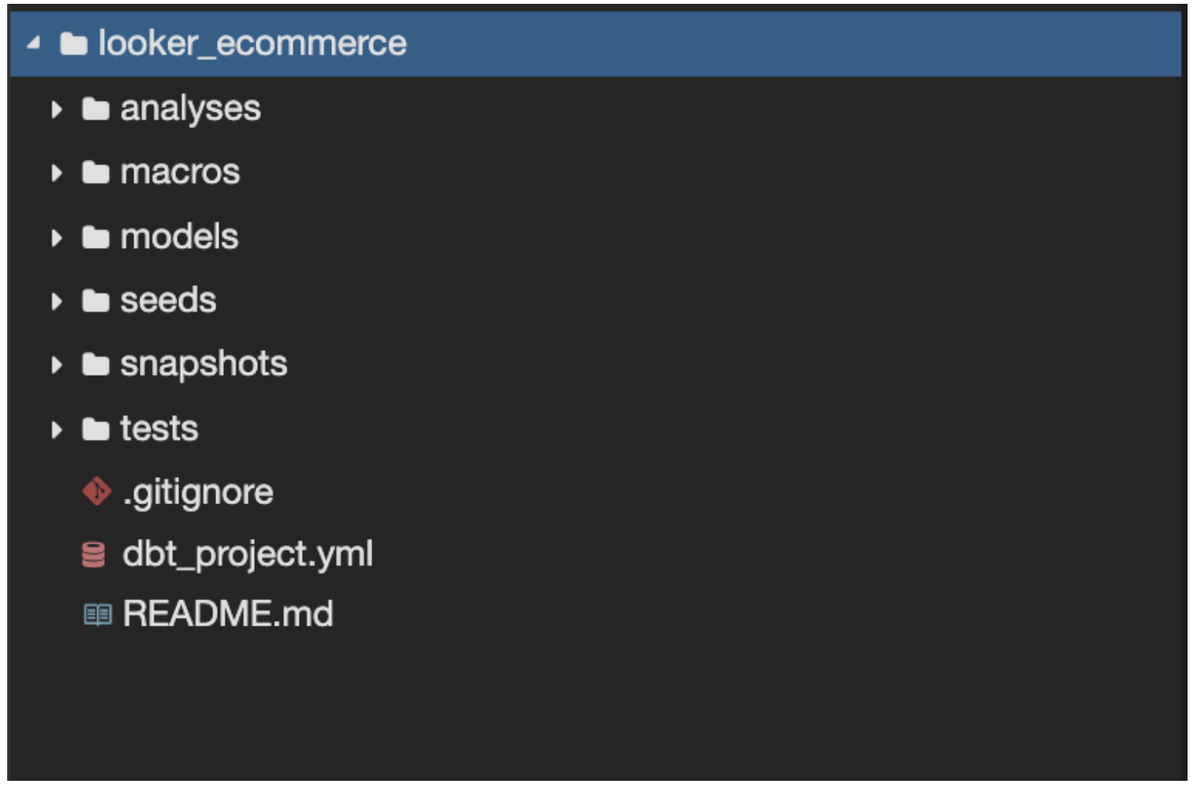
[1] https://docs.getdbt.com/docs/introduction https://www.getdbt.com/case-studies

A preview of what we are building



A preview of what we are building

Sample of dbt project:



Recap of dbt commands:

```
# Checks if/where dbt is installed
which dbt

# Checks the version of dbt installed
dbt --version

# Initializes a new dbt project
dbt init <project_name>

# Short for dbt --help
dbt -h
```

Let's practice!

CASE STUDY: BUILDING E-COMMERCE DATA MODELS WITH DBT

Setting up the dbt project and loading data

CASE STUDY: BUILDING E-COMMERCE DATA MODELS WITH DBT

Susan Sun
Freelance Data Scientist

Review: dbt set up and initialization

Install dbt

```
pip install dbt
```

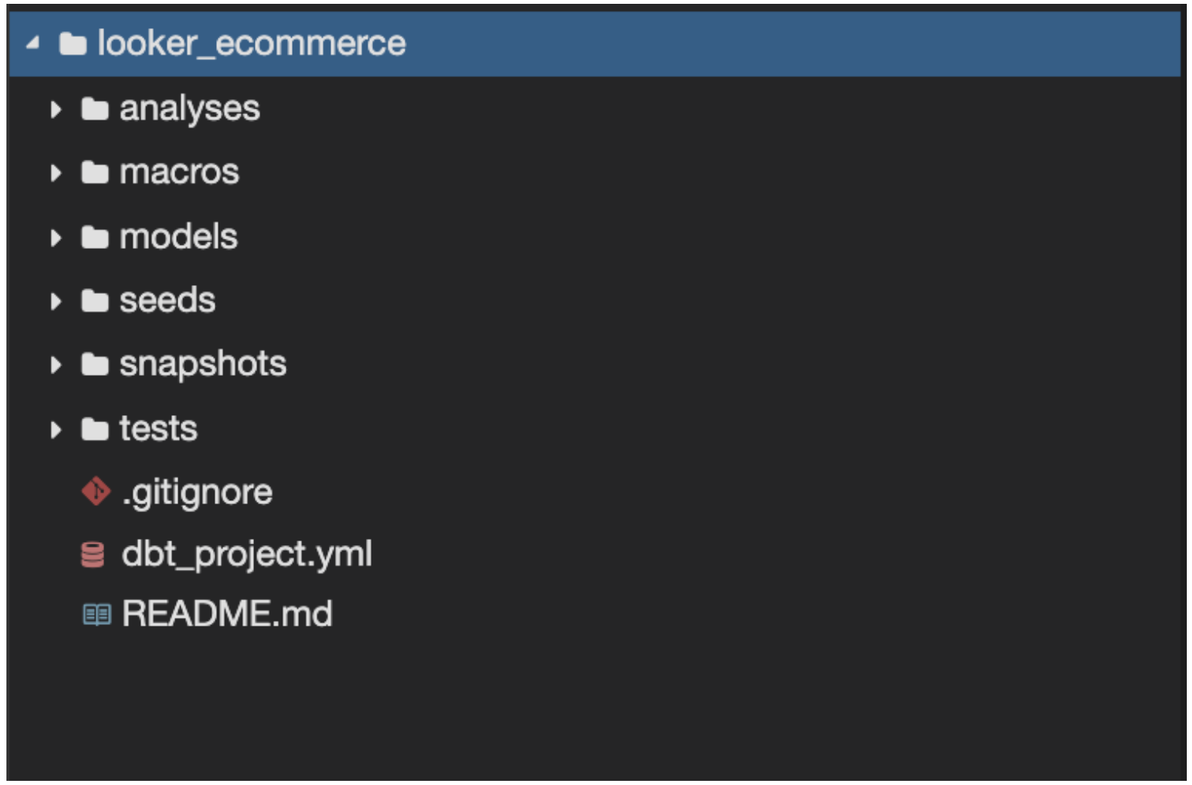
Initialize dbt project `looker_ecommerce`

```
dbt init looker_ecommerce
```

Verify set up success

```
cd looker_ecommerce
dbt debug
```

dbt auto-generated file directory:



Getting familiar with the data: distribution centers

- `distribution_centers.csv` is small and static, with only 10 rows
- A sample of the raw `distribution_centers.csv` data file

```
id,name,latitude,longitude
1,Memphis TN,35.1174,-89.9711
2,Chicago IL,41.8369,-87.6847
...
10,Savannah GA,32.0167,-81.1167
```

- **id**: Unique identifier for each distribution center
- **name**: Name of the distribution center
- **latitude**: Latitude coordinate of the distribution center
- **longitude**: Longitude coordinate of the distribution center

Getting familiar with the data: orders

`orders.csv` is large and constantly updating, with 125,000 rows and 9 columns

- **order_id:** Unique ID for each order item
- **user_id:** ID of the user who placed an order
- **status:** Status of the order
- **gender:** Gender of the user
- **created_at:** Order created at timestamp
- **returned_at:** Order returned at timestamp
- **shipped_at:** Order shipped at timestamp
- **delivered_at:** Order delivered at timestamp
- **num_of_items:** Number of items in each order

Getting familiar with the data: orders

A sample of the raw `orders.csv` data file:

```
order_id,user_id,status,gender,created_at,returned_at,shipped_at,delivered_at,num_of_items

88616,70663,Returned,F,2024-01-15 18:12:00+00:00,2024-01-21 08:54:00+00:00,2024-01-18
05:01:00+00:00,2024-01-18 23:54:00+00:00,2

88641,70659,Returned,F,2021-03-18 17:45:00+00:00,2021-03-27 02:56:00+00:00,2021-03-21
17:08:00+00:00,2021-03-25 03:07:00+00:00,4

....
```

Setting up raw source and seed data sources

Distribution center raw data file:

- Data characteristics:
  - Small, flat file (csv)
  - Slow changing data
- dbt load method:
  - `dbt seed`
  - Load as a one-time file

Orders raw data file:

- Data characteristics:
  - Large dataset
  - Fast changing data
- dbt load method:
  - `dbt source`
  - Connector to DuckDB

Setting up raw sources and seed data sources

`dbt init` auto-generates the basic file structure:

```
looker_ecommerce/
  macros/
  models/
  seeds/
  snapshots/
  tests/
  dbt_project.yml
```

Setting up raw source and seed data sources

Load `distribution_center` as `seed`:

```
looker_ecommerce/
  macros/
  models/
    stg_looker__distribution_centers.sql
  seeds/
    looker__distribution_centers.csv
  snapshots/
  tests/
  dbt_project.yml
```

In `stg_looker__distribution_centers.sql`:

```
SELECT
  id,
  name,
  latitude,
  longitude
FROM
 {{ref('looker__distribution_centers')}}
```

Setting up raw source and seed data sources

Load `orders` as `source`:

```
looker_ecommerce/
  macros/
  models/
    stg_looker__orders.sql
  seeds/
  snapshots/
  tests/
  dbt_project.yml
```

In `stg_looker__orders.sql`:

```
SELECT *
FROM
{{source('looker_ecommerce', 'orders')}}
```

Documenting sources and staging models

To document `sources`:

```
looker_ecommerce/
  macros/
  models/
    _looker__sources.yml
  seeds/
  snapshots/
  tests/
  dbt_project.yml
```

In `_looker__sources.yml`:

```
version: 2


sources:
  - name: looker_ecommerce
    tables:
      - name: orders
```

Documenting sources and staging model

- Sample `_looker__models.yml` file under the same `models` directory:

```
version: 2

models:
  - name: stg_looker__distribution_centers
    description: Distribution center name and location

  - name: stg_looker__orders
    description: Order information such as order status
```

Sources, seeds, models, and yaml

```
looker_ecommerce/
  macros/
  models/
    _looker__models.yml
    _looker__sources.yml
    stg_looker__distribution_centers.sql
    stg_looker__orders.sql
  seeds/
    looker__distribution_centers.csv
  snapshots/
  tests/
  dbt_project.yml
```

Review: dbt subcommands

Loads csv files into as seed files

```
dbt seed
```

Creates or updates all models in dbt project

```
dbt run
```

Creates or updates the specified model

```
dbt run --select model
```

Runs all tests in dbt project

```
dbt test
```

Runs tests with the specified model

```
dbt test --select model
```

Combines `dbt run` and `dbt test` in one!

```
dbt build
dbt build --select model
```

Review: best practice guides

models naming structure:

- uses double underscore to separate data source name from model name
- `<data_source>__<model_name>.sql`

e.g.

- `stg_looker__distribution_centers.sql`
- `stg_looker__orders.sql`

yaml file naming structure:

- starts with single underscore
- uses double underscore to separate data source name from artifact name
- `_<data_source>__<artifact_type>.yml`

e.g.

- `_looker__models.yml`
- `_looker__sources.yml`

[1] https://docs.getdbt.com/best-practices

Let's practice!

CASE STUDY: BUILDING E-COMMERCE DATA MODELS WITH DBT