What is a dbt model?

INTRODUCTION TO DBT

Mike Metzger
Data Engineer

What is a data model?

- Conceptual, with different definitions depending on context
- Represents the logical meaning of data
- How the data and its components relate
- Helps users collaborate

Species	# of legs	Venomous
Cheetah	4	No
Duck	2	No
Platypus	4	Yes
Rattlesnake	0	Yes

What is a model in dbt?

- Represents the various transformations
- Typically written in SQL
  - Newer versions can use Python
- Usually a `SELECT` query
- Each model represented by a text file with .sql extension

Simple dbt model

1. Create a directory in the `models` directory
2. Create a `.sql` file in above directory
3. Add the SQL statement to the newly created file
4. Run `dbt run` to materialize the model

```
bash> mkdir models/order
bash> touch models/order/customer_orders.
```

```
select first_name,
       last_name,
       shipping_address,
       item_quantity
from source_table
```

```
bash> dbt run
```

Reading from Parquet

- Parquet?
  - Columnar binary file format
  - Used by many tools to efficiently store data
    - Apache Spark
    - Apache Arrow
    - DuckDB

- DuckDB can read Parquet files directly
  - read_parquet
    - `SELECT * FROM read_parquet('filename.parquet')`
  - Or simply the filename in single quotes
    - `SELECT * FROM 'filename.parquet'`

Let's practice!

INTRODUCTION TO DBT

Updating dbt models

INTRODUCTION TO DBT

Mike Metzger
Data Engineer

Why update?

- Iterative work
- Fixing bugs with queries / models
- Migrating to different sources / destinations

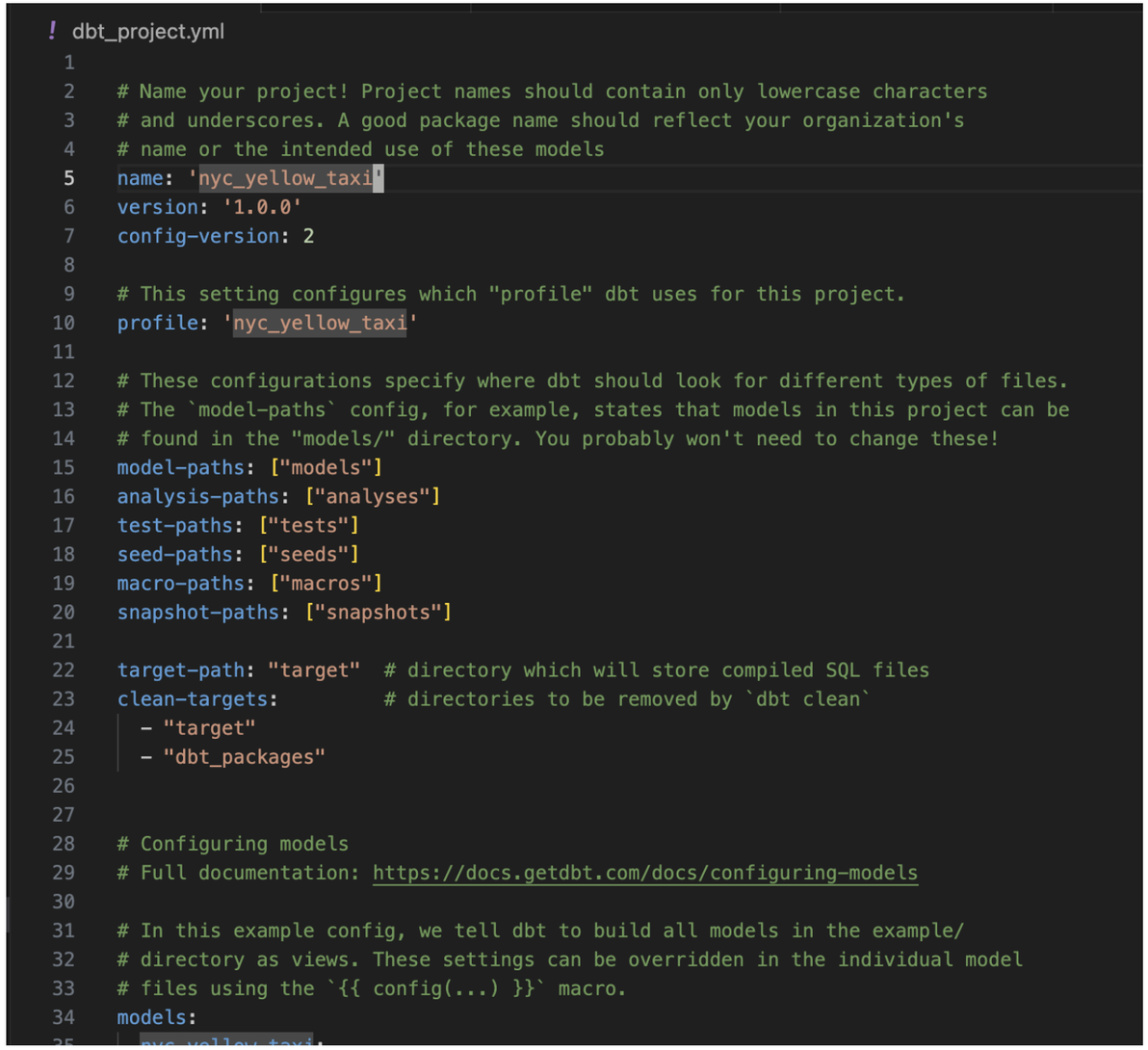
[1] Photo by Caspar Camille Rubin on Unsplash

Update workflow

1. Check out from source control
   - `git clone dbt_project`
2. Find the model in question
3. Update query contents
4. Regenerate with `dbt run`
   - Or `dbt run -f` (Force full refresh)
5. Check changes back to source control

YAML files

- Some updates may require changes to YAML / `.yml` files
- Typically would require changes in:
  - `dbt_project.yml`
  - `model_properties.yml`



dbt_project.yml

- Contains mostly contents related to full project
  - Project name / version
  - Directory locations
- Model materialization settings (global)
- One `dbt_project.yml` file per project

model_properties.yml

- Contain settings that reference models
  - Description
  - Documentation details
  - Much more
- Can actually be named anything (with `.yml`) in models/ subdirectory
- Can have as many files as needed

models > ! model_properties.yml

```yaml
version: 2

models:
  - name: taxi_rides_raw
    description: Initial import of the NYC Yellow Taxi trip data from Parquet source
    access: public
  - name: avg_fare_per_day
    description: The average ride amount spent per day
    access: public
```

Let's practice!

INTRODUCTION TO DBT

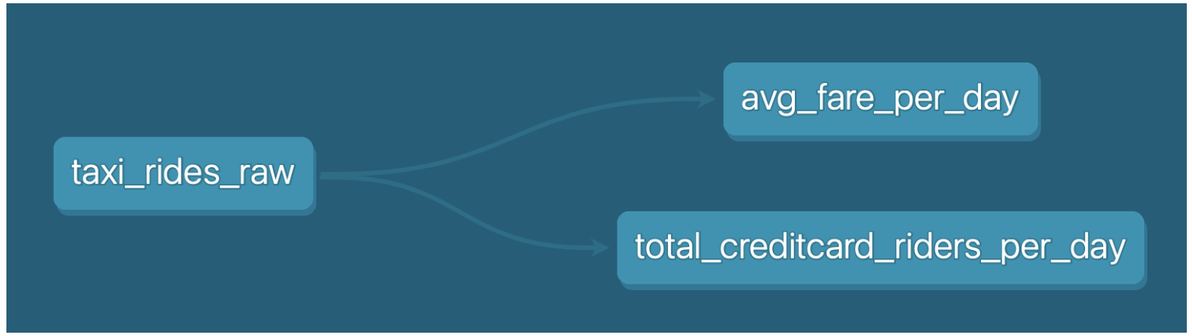
Hierarchical models in dbt

INTRODUCTION TO DBT

Mike Metzger
Data Engineer

What is a hierarchy in dbt?

- Represents the dependencies between models
- Also known as a directed acyclic graph (DAG), or lineage graph
  - Note this being specific to dbt, rather than general DAG
- Allows models to be built / updated according to dependencies



How are hierarchies defined?

- Built using the Jinja template language in the model file(s)
- Most often using the `ref` function
- Replace table name with `{{ ref('model_name') }}` in SQL
- (Re-)Materialize table with `dbt run`
- `ref` templates are substituted with actual table names

```
SELECT
  first_name, last_name
FROM taxi_rides_raw
```

```
SELECT
  first_name, last_name
FROM {{ ref('taxi_rides_raw') }}
```

Jinja templating language

- Simple text based templating language
- Used in many tools beyond just dbt
- `{{ ... }}` represents a template substitution
- dbt has many Jinja functions available for use in projects
- Allows for more dynamic usage of dbt
- Some of the many Jinja functions available
  - `ref`
  - `config`
  - `docs`
  - Many more!

Let's practice!

INTRODUCTION TO DBT

Model troubleshooting

INTRODUCTION TO DBT

Mike Metzger
Data Engineer

datacamp

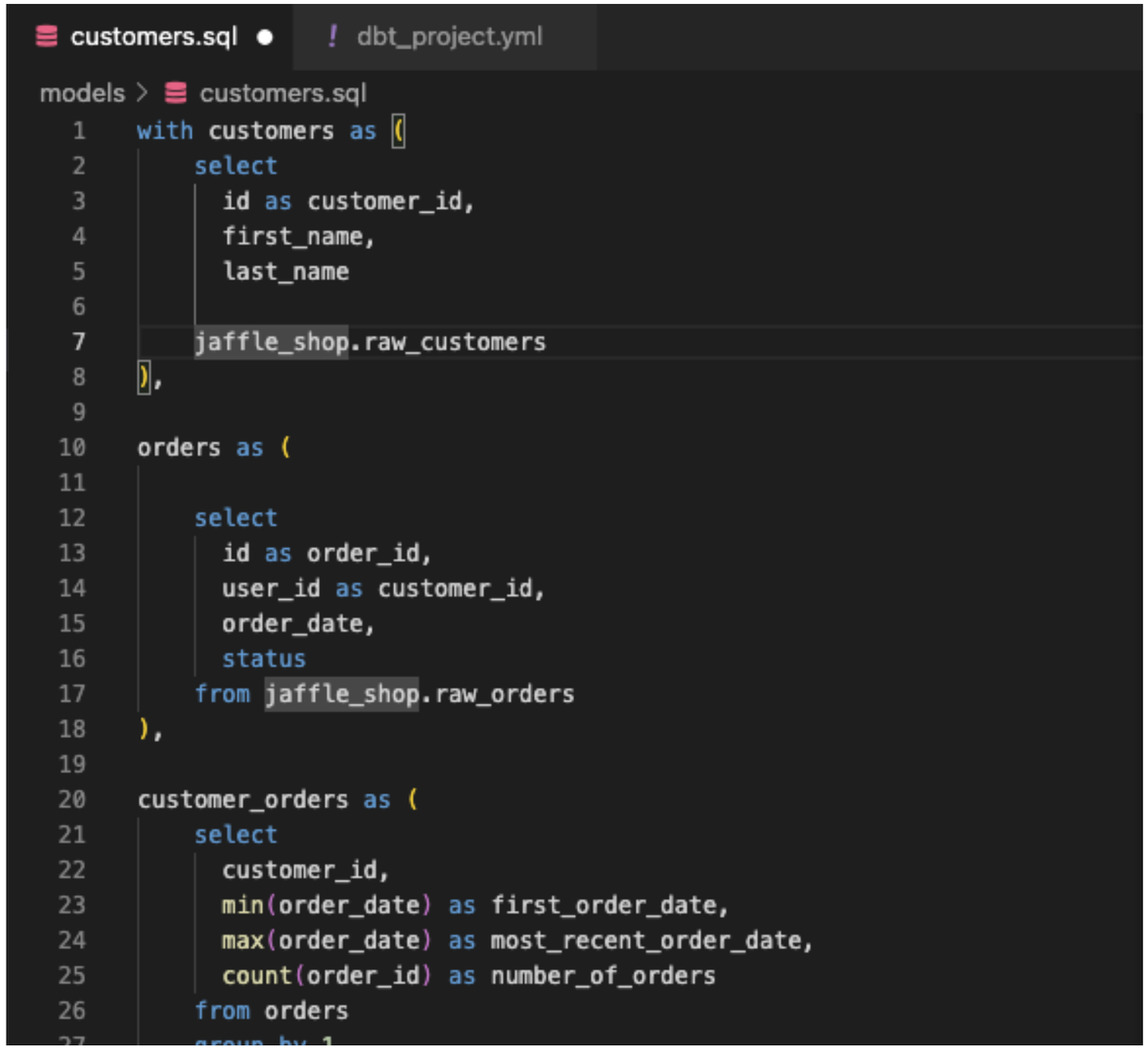
Common model issues

- Query errors
  - Syntax errors
  - Logic errors
- Invalid references

[1] Photo byahmad gunnaivi on Unsplash

Query errors

- Misspellings / syntax issues
- Non-standard SQL
- Common SQL logic issues
  - Not grouping by all non-aggregated columns
  - Incorrect CTEs



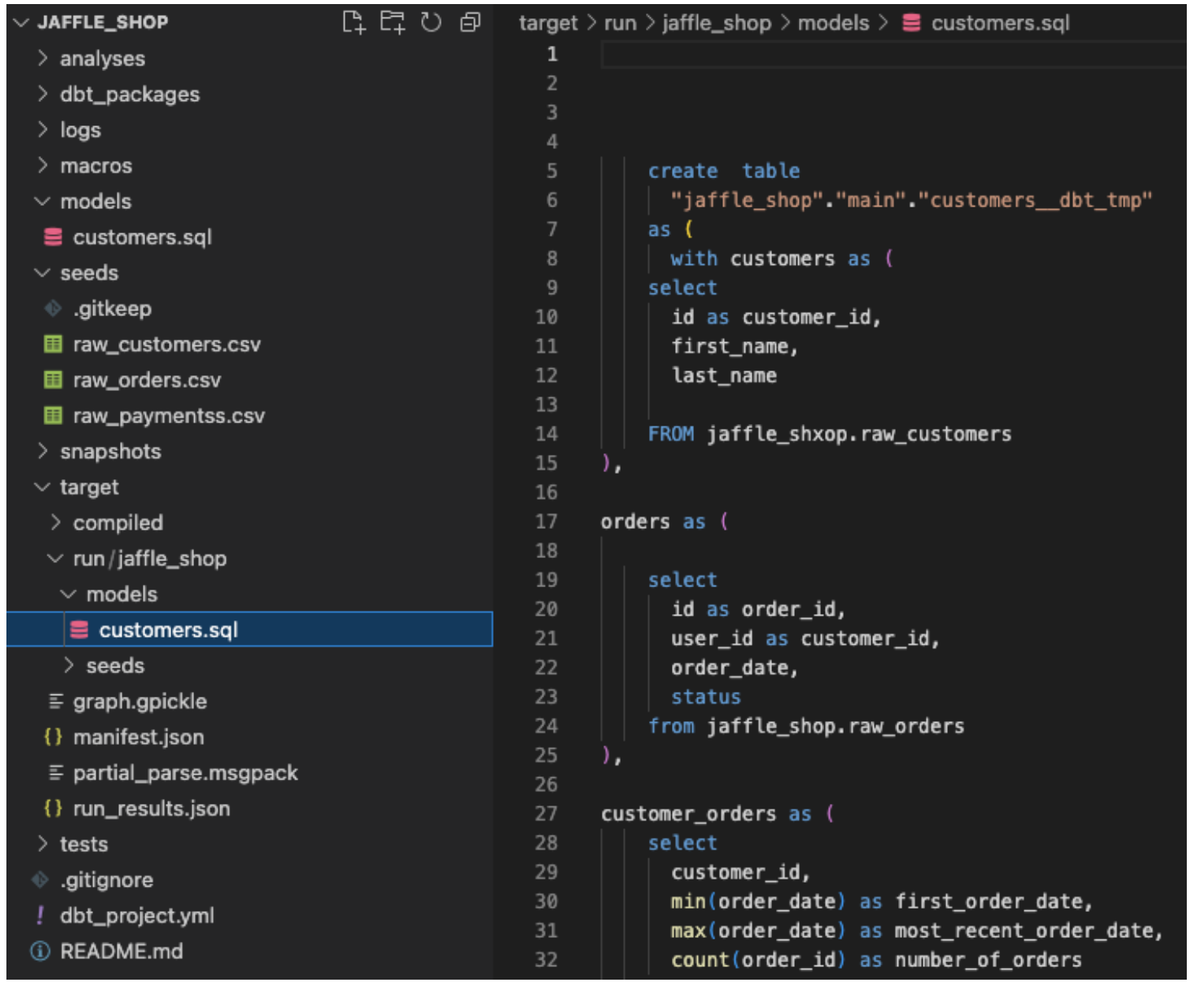
Invalid references

- Table / view named differently than expected
  - Context name
  - Preceding underscore or periods
- Referencing objects that have yet to be created

```
15:46:21  Running with dbt=1.4.5
15:46:21  Found 1 model, 0 tests, 0 snapshots, 0 analyses, 296 macros, 0 operations, 3 seed files, 0 sources, 0 exposures, 0 metrics
15:46:21
15:46:21  Concurrency: 1 threads (target='dev')
15:46:21
15:46:21  1 of 1 START sql table model main.customers ..................................... [RUN]
15:46:21  1 of 1 ERROR creating sql table model main.customers ............................ [ERROR in 0.03s]
15:46:21
15:46:21  Finished running 1 table model in 0 hours 0 minutes and 0.07 seconds (0.07s).
15:46:21
15:46:21  Completed with 1 error and 0 warnings:
15:46:21
15:46:21  Runtime Error in model customers (models/customers.sql)
15:46:21    Catalog Error: Table with name raw_customers does not exist!
15:46:21    Did you mean "jaffle_shop.raw_customers"?
15:46:21
15:46:21  Done. PASS=0 WARN=0 ERROR=1 SKIP=0 TOTAL=1
```

Troubleshooting methods

- `dbt run`
- View logs
  - `logs/dbt.log`
  - `run_results.json`
- Viewing generated SQL
  - Running generated SQL manually
- Verify fixes



Let's practice!

INTRODUCTION TO DBT