John Doe • 2nd

Analytics engineer | Author

1d •••

I write 5x more YAML nowadays than Python 🥲

Like · 1 | Reply · 1 Reply

Jane Doe

Data Platform Engineering

1d •••

YAML >>> Python

Like | Reply

```yaml
models:
  - name: pizzas
    description: "Assortment of 100 pizzas"
    config:
      materialized: table
    meta:
      owner: "John Doe"
    columns:
      - name: id
        data_type: integer
        description: "Pizza ID"
        constraints:
          - type: primary_key
      - name: dough
        data_type: varchar
      - name: base
        data_type: varchar
      - name: cheese
        data_type: varchar
      - name: extra_1
        data_type: varchar
```

Matthieu Caneill

~~ML engineer~~

Software / Analytics

🇫🇷 🥐

Pixel

Coding assistant

Picnic

Online supermarket

Groceries supply-chain

Lots of data

The rise of the YAML engineer

Takeaways

1. **Describe** the desired state, don't compute it
2. **YAML** is ubiquitous in data systems
3. Track metadata **in git** and build on top

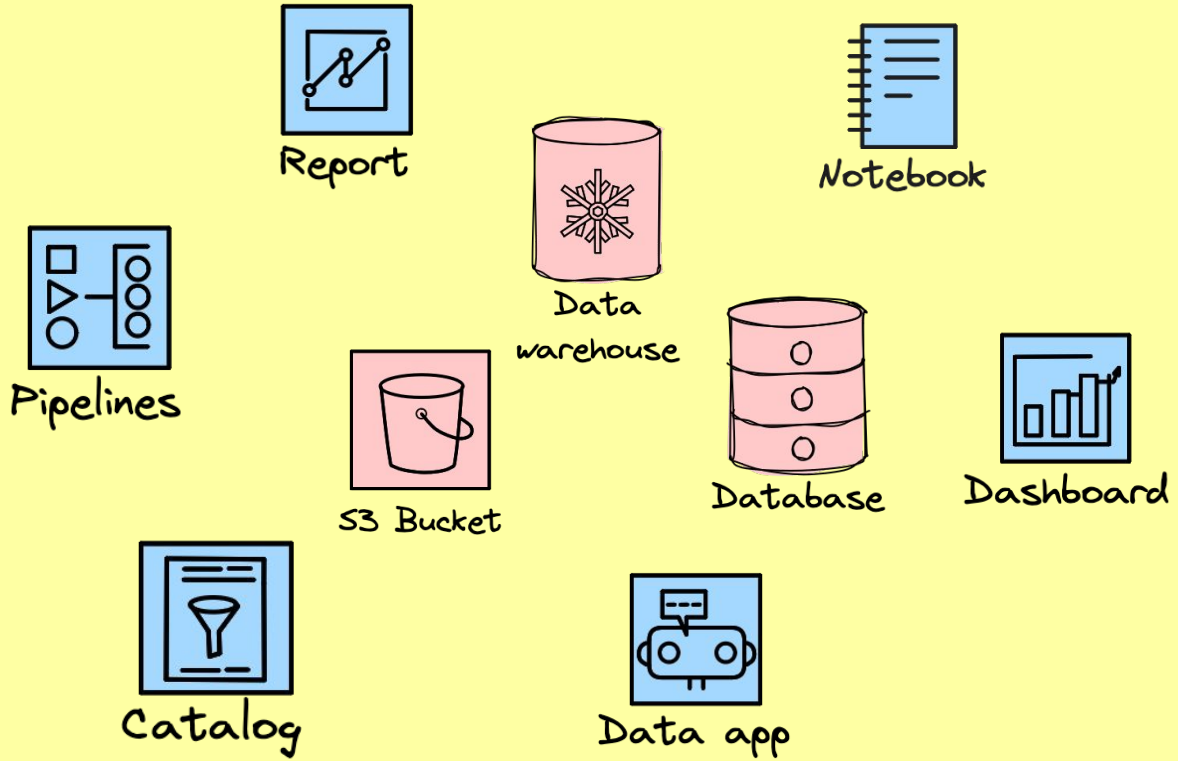
Outline

1. Data, logic, & state
2. The declarative **data stack**
3. **GitOps** & good practices

Data,
Logic,
& State

```
create table pizzas (
  id int primary key,
  cheese varchar
)
```

```
import pandas

def transform(df):
   output = ...
```

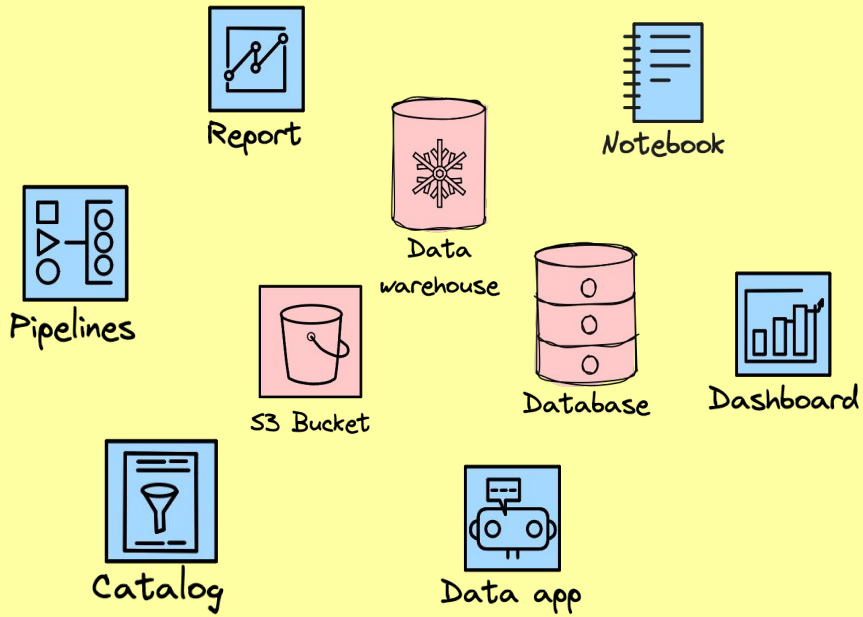


The Code

```
create table pizzas (
  id int primary key,
  cheese varchar
)
```

```
import pandas

def transform(df):
   output = ...
```

The Cloud



The Code

```
create table pizzas (
  id int primary key,
  cheese varchar
)
```

```
import pandas

def transform(df):
   output =  ...
```

How many pizzas do we have with pepperoni?

Imperative Python

```
counter = 0

for pizza in pizzas:
    if pizza.extra_1 == "pepperoni":
        counter += 1

return counter
```

Declarative SQL

```
select count(*)
from pizzas
where extra_1 = 'pepperoni'
```

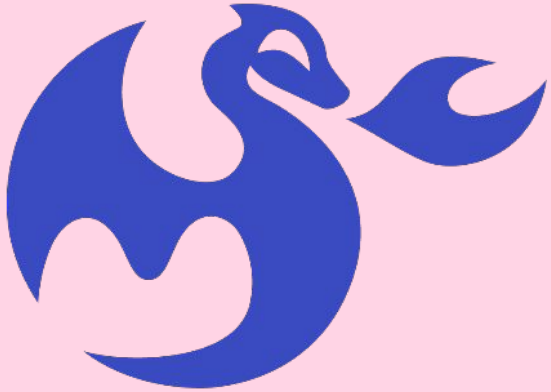
```sql
select count(*)
from pizzas
where extra_1 = 'pepperoni'
```

```python
counter = 0

for pizza in pizzas:
    if pizza.extra_1 == "pepperoni":
        counter += 1

return counter
```

Capture
new pizza data
from JSON and CSV
to the database

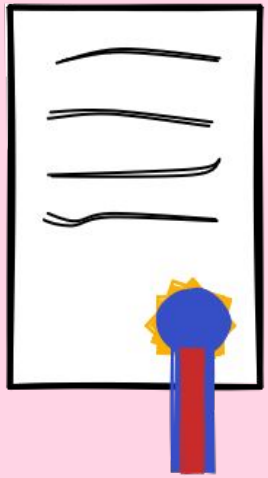
```yaml
plugins:
  extractors:
    - name: tap-csv
      config:
        files:
          - file: data/pizzas.csv
            keys:
              - id
    - name: tap-json
      config:
        files:
          - file: other_data/pizzas.json
            keys:
              - id
  loaders:
    - name: target-postgres
      config:
        host: localhost
        port: 5432
        user: postgres
        dbname: pizzas
```

The Declarative Pizza Stack



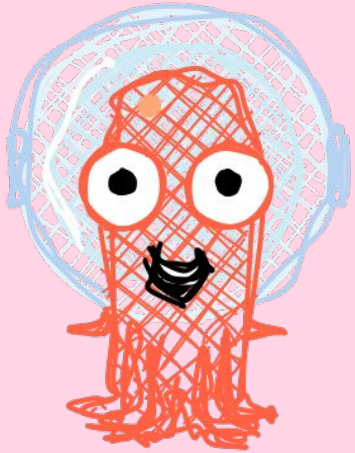
Meltano

```yaml
plugins:
  extractors:
    - name: tap-csv
      config:
        files:
          - file: data/pizzas.csv
            keys:
              - id
    - name: tap-json
      config:
        files:
          - file: other_data/pizzas.json
            keys:
              - id
  loaders:
    - name: target-postgres
      config:
        host: localhost
        port: 5432
        user: postgres
        dbname: pizzas
```



data contract

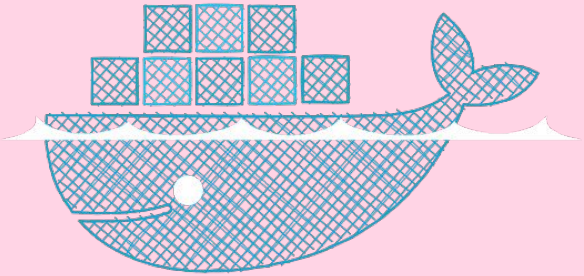
```yaml
dataContractSpecification: 0.9.3
id: urn:datacontract:checkout:orders-latest
info:
  title: Raw pizza data
  version: 1.0.0
  description: Raw data about pizzas
  owner: Pizza production team
tags:
  - uncooked
servers:
  production:
    type: s3
    environment: prod
    location: s3://raw-pizza-data/data/{date}/pizzas.json
    format: json
models:
  pizzas:
    type: file
    fields:
      pizza_id:
        $ref: '#/definitions/pizza_id'
        required: true
        unique: true
        primary: true
...
```




```yaml
kind: Workflow
spec:
  entrypoint: etl
  templates:
  - name: etl
    dag:
      tasks:
      - name: extract_ingredients
        template: extract
        arguments:
          parameters: [{name: cheese, value: mozzarella}]
      - name: cook
        template: transform
        depends: extract
        arguments:
          parameters: [{name: temperature, value: 240}]
      - name: serve
        template: load
        depends: cook
        arguments:
          parameters: [{name: cuts, value: 8}]
```


```yaml
models:
  - name: pizzas
    description: "Assortment of 100 pizzas"
    config:
      schema: analytics
      materialized: table
      grants:
        select: ["pizza_analyst"]
      tags: ["food"]
    meta:
      owner: "John Doe"
    columns:
      - name: id
        data_type: integer
        description: "Pizza ID"
        constraints:
          - type: primary_key
        data_tests:
          - unique
          - not_null
      - name: dough
        ...
```


```yaml
models:
  - name: pizzas
    columns:
      ...
      - name: extra_1
        data_type: varchar
        meta:
          dimension:
            colors:
              pepperoni: "green"
              artichoke: "red"
              mushroom: "brown"
              olives: "yellow"
```



docker
compose

```yaml
services:

  db:
    image: postgres:13
    environment:
      POSTGRES_USER: pizzaiolo
      POSTGRES_PASSWORD: hawaii
      POSTGRES_DB: pizzas
    volumes:
      - data:/var/lib/postgresql/data

  extract:
    image: meltano:3.5
    volumes:
      - ./meltano.yml:/opt/meltano/

  transform:
    image: dbt-labs/dbt-postgres:1.8
    volumes:
      - ./dbt:/usr/app

  viz:
    image: lightdash/lightdash:latest
    ports:
      - "8080:8080"

  ...
```

GitOps
& good practices
& existential
questions

Is this a good idea?

git

pizzas.yml

```
pizzas:
    - id
    - extra_1
+   - extra_2
```

dashboard_pizzas.yml

```
pizza_chart:
    - pizzas.id
    - pizzas.cheese
+ groups:
+   - pizzas.cheese
```

CI/CD

1. Compute delta

2. Apply commands to reconcile state

```
alter tables pizzas
add column extra_2;
update pizzas...
```

```
POST
/api/dashboards/pizzas
-d "{type: pizza_chart,
labels: [id, cheese],
group_by: cheese}"
```

data platforms

id	extra_1	extra_2
41	pepperoni	olives
42	chili	egg
43	pineapple	ham

Why am I paying you
to write YAML?

Credits

- **Maxime Beauchemin**
  The Rise of the Data Engineer
- **Picnic**
  We're hiring YAML engineers!
  https://jobs.picnic.app

  We love declarativeness
  https://dbt-score.picnic.tech

Matthieu Caneill
https://matthieu.io