

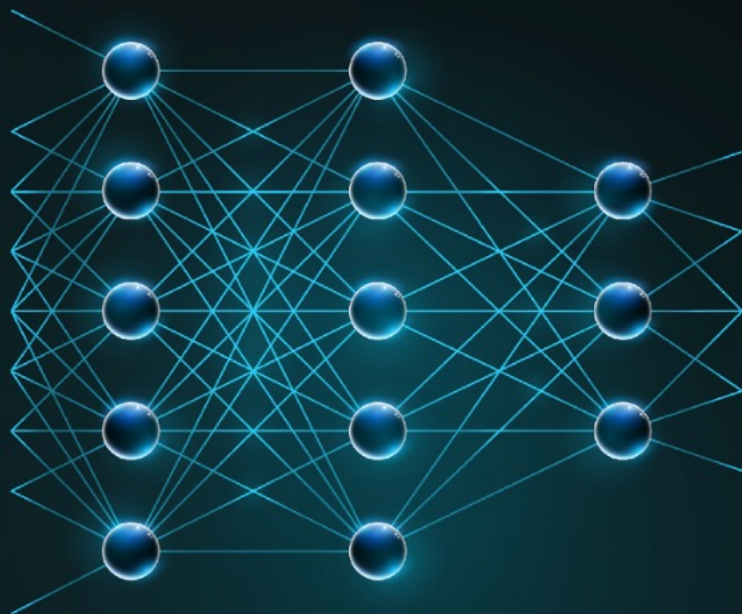


DATA
SCIENCE
SUMMIT

MACHINE
LEARNING
EDITION

DataOps – Is there a cure for the pains of data science team?

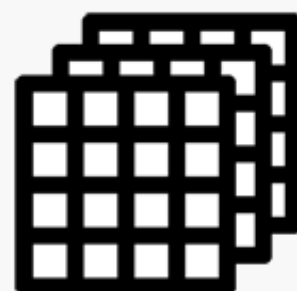
Paweł Bogumiło
BI & Analytics Engineer @RTB House



DATASET

HP: 

100 / 100



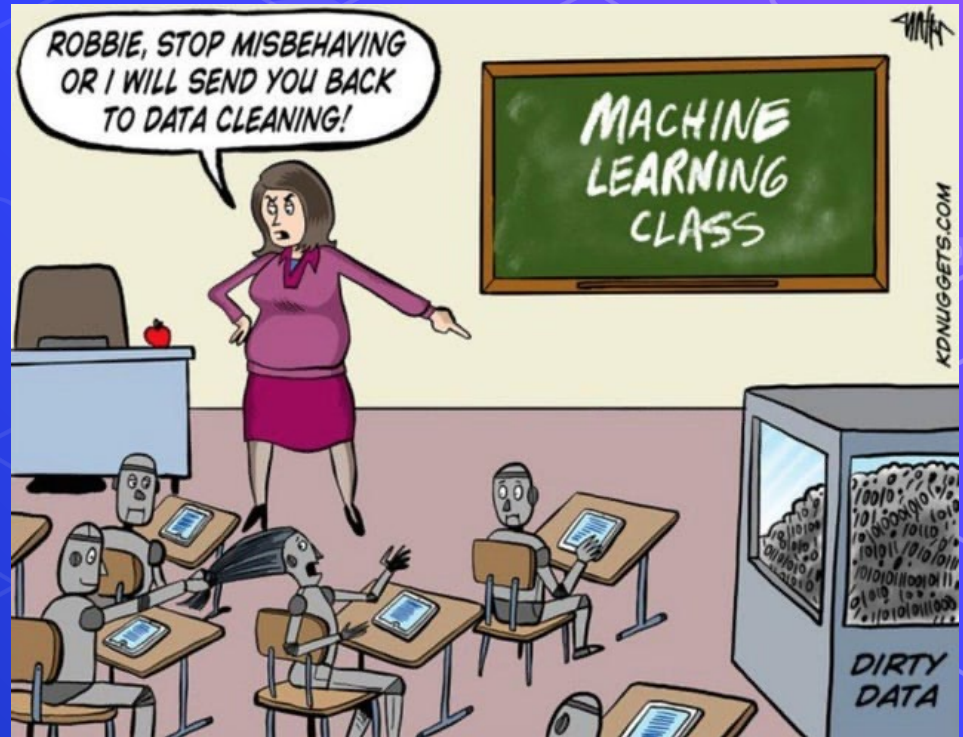
YOU

HP: 

100 / 100

A wild DATASET
appeared!

“ What data scientists spend 80% of their time on?



Who remember that?

Harvard
Business
Review



DATA

Data Scientist: The Sexiest Job of the 21st Century

by Thomas H. Davenport and D.J. Patil

FROM THE OCTOBER 2012 ISSUE

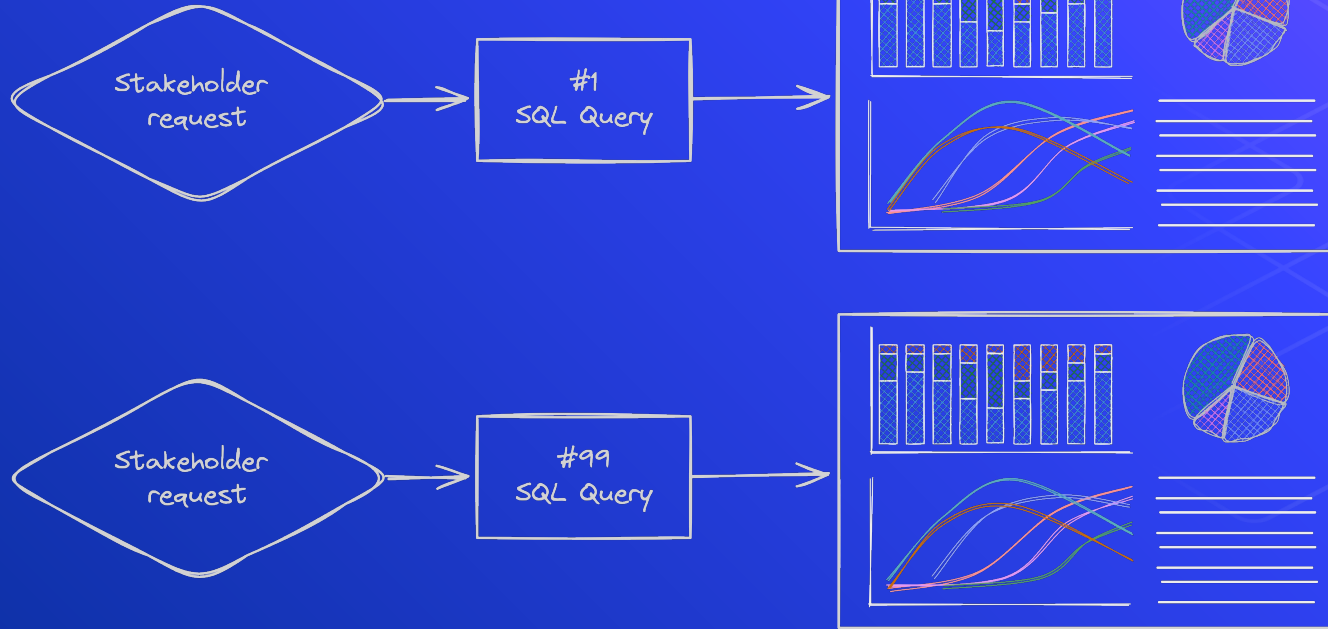
When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't seeking out connections with the people who were already on the site at the rate executives had expected. Something was apparently missing in the social experience. As one LinkedIn manager put it, "It was like arriving at a conference reception and realizing you don't know

// Per Forrester Research, **60%** of the data and analytics decision-makers surveyed said they are not very confident in their analytics insights. **Only 10%** responded that their organizations sufficiently manage the quality of data and analytics. **Just 16%** believe they perform well in producing accurate models.



<https://xkcd.com/1838/>

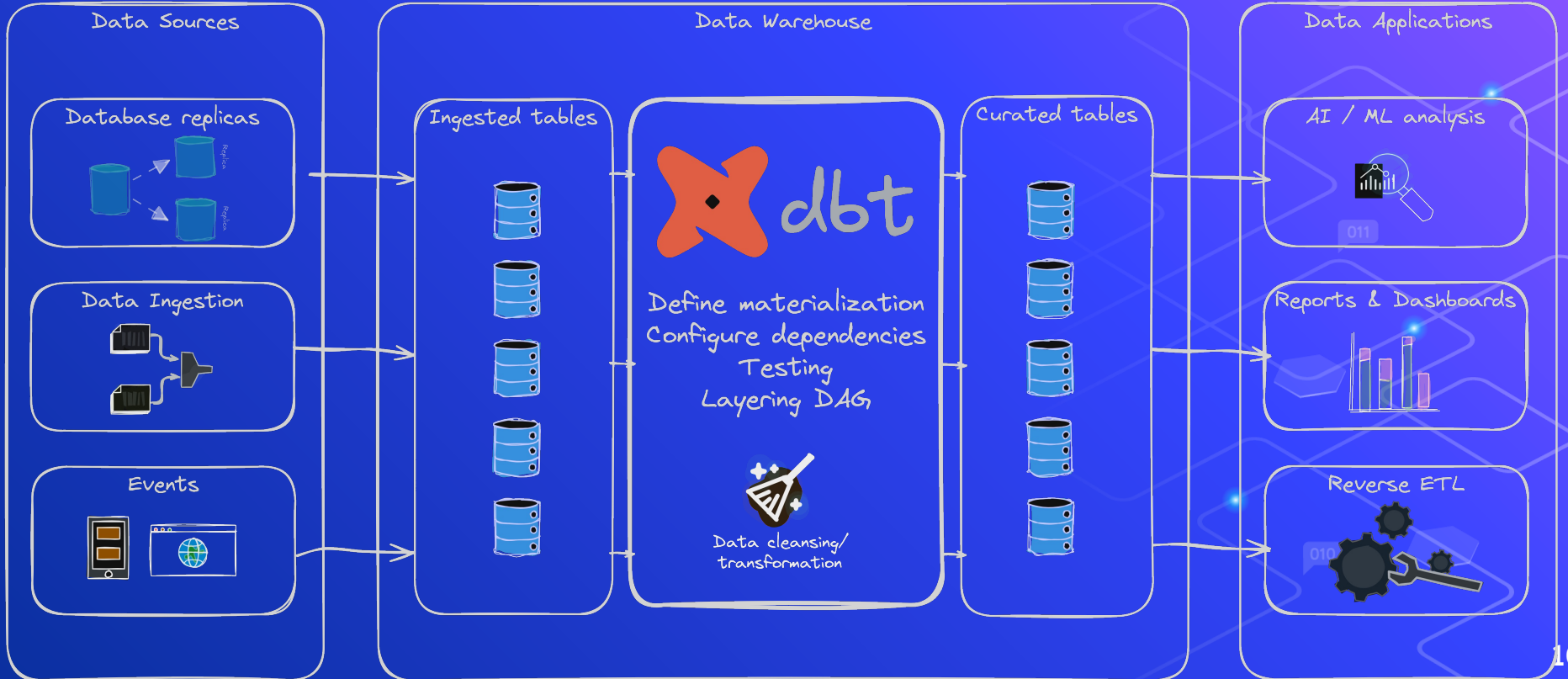
This is not DRY



❖ The schedule

- 1 Best SoftEng practices in data?
- 2 'dbt run' for free?
- 3 Blue-Green-Deployment vs. DWH
- 4 Differences in CI/CD process
between Snowflake & BigQuery
- 5 Landcape of dbt-enabled products

EL[T]: T means transformation



dbt for DataOps?

- ⬡ Testing both code and data
- ⬡ Versioning
- ⬡ Proper management of branches and PRs
- ⬡ Automation of the above
- ⬡ Multiple environments
- ⬡ Containerization
- ⬡ Parameterization of the process

Transformation flow: Types of data models

transform

mart

stage



```
version: 2
```

```
sources:
```

```
- name: warehouse
  description: Data from application database brought in by an ETL process.
```

```
tables:
```

```
- name: customers
  columns:
  - name: customer_id
  tests:
    - not_null
```

```
- name: orders
  columns:
  - name: order_id
  tests:
    - not_null
    - unique
```

```
- name: cust_id
  tests:
    - relationships:
        to: source('warehouse', 'customers')
        field: customer_id
```

```
- name: state
```

```
src_backend.yml
```

```
{{
```

```
  config(
    materialized = "table"
  )
}}
```

```
with source as (
  select *
  from {{ source('warehouse', 'customers') }}
```

```
),
renamed as (
  select customer_id,
         zipcode,
         city,
         state_code,
         datetime_created::TIMESTAMP c.datetime_created,
         datetime_updated::TIMESTAMP c.datetime_updated,
         dbt_valid_from,
         dbt_valid_to
  from source
```

```
select * from renamed
```

```
stg_customer.sql
```

```
{{
```

```
with customers as (
  select *
  from {{ ref('stg_customers') }}
```

```
),
state as (
  select *
  from {{ ref('stg_state') }}
```

```
)
select c.customer_id,
       c.zipcode,
       c.city,
       c.state_code,
       s.state_name,
       c.datetime_created,
       c.datetime_updated,
       c.dbt_valid_from::TIMESTAMP as valid_from,
       CASE
         WHEN c.dbt_valid_to IS NULL THEN '9999-12-31'::TIMESTAMP
         ELSE c.dbt_valid_to::TIMESTAMP
       END as valid_to
from customers c
join state s on c.state_code = s.state_code
```

```
dim_customer.sql
```

```
with customers as (
  select *
  from {{ ref('stg_customers') }}
```

```
),
state as (
  select *
  from {{ ref('stg_state') }}
```

```
)
select c.customer_id,
       c.zipcode,
       c.city,
       c.state_code,
       s.state_name,
       c.datetime_created,
       c.datetime_updated,
       c.dbt_valid_from::TIMESTAMP as valid_from,
       CASE
         WHEN c.dbt_valid_to IS NULL THEN '9999-12-31'::TIMESTAMP
         ELSE c.dbt_valid_to::TIMESTAMP
       END as valid_to
from customers c
join state s on c.state_code = s.state_code
```

Open source means free, right?

⬡ OSS orchestrator

⬡ Modern Data Tool

⬡ Serverless CI / CD



Blue-Green Deployment

- develop
- build on a copy of prod
- swap (schemas)

Source: martinfowler.com/bliki/BlueGreenDeployment.html

Characteristic of your DWH is crucial for right CI/CD setup



⬡ ZCC

```
{%- macro clone_db() -%}
  {% set sql %}
    DROP DATABASE IF EXISTS ci_pbogumilo;
    CREATE DATABASE ci_pbogumilo CLONE development;
    GRANT ALL ON DATABASE ci_pbogumilo TO ROLE transformer;
  {% endset %}
  {% do run_query(sql) %}
{%- endmacro -%}

{%- macro drop_db() -%}
  {% set sql %}
    DROP DATABASE IF EXISTS ci_pbogumilo;
  {% endset %}
  {% do run_query(sql) %}
{%- endmacro -%}
```

Source: [macro by Jeremy Yeo](#)



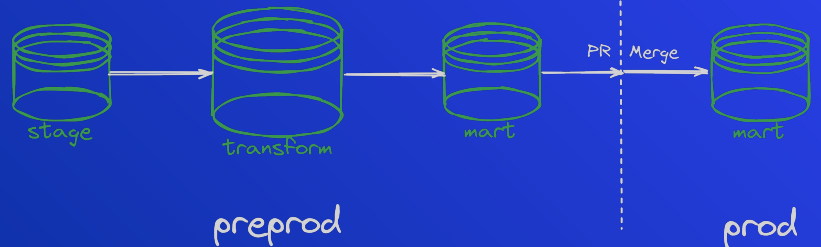
- ⬡ No schema/dataset rename
- ⬡ No database swapping

Characteristic of your DWH is crucial for right CI/CD setup



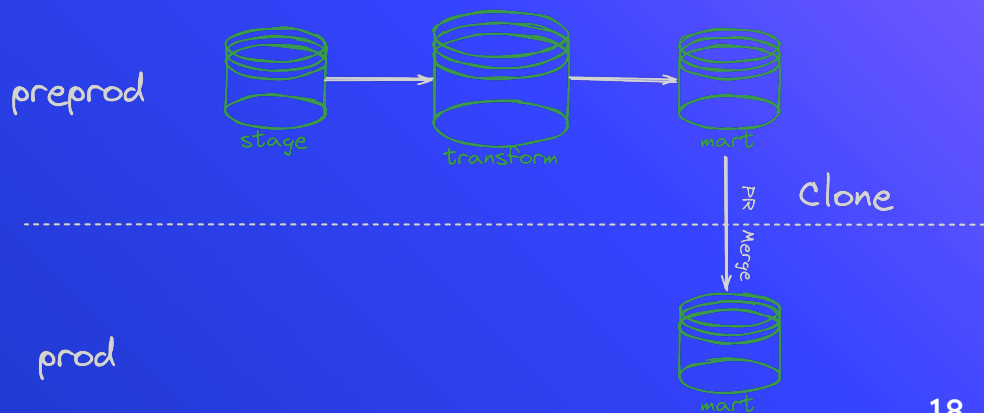
⬡ WAP

Write-Audit



⬡ WAC

Write-Audit



CI/CD with GH Actions Workflows

```
pr.yml

name: On PR

on:
  pull_request:
    branches:
      - main

jobs:
  on_pr:
    name: PR
    runs-on: ubuntu-latest
    permissions:
      contents: 'read'
      id-token: 'write'

    steps:
      - uses: actions/checkout@v4

      - uses: actions/setup-python@v4
        with:
          python-version: '3.9.x'

      - uses: 'google-github-actions/auth@v2'
        with:
          credentials_json: ${ secrets.DBT_GOOGLLE_BIGQUERY_KEYFILE_CI }}

      - run: |
        pip install dbt-bigquery
        dbt deps

      - run: |
        dbt run -t preprod

      - run: |
        dbt test -t preprod
```

```
push.yml

name: On Push

on:
  push:
    branches:
      - main

jobs:
  deploy:
    name: Push to Prod
    runs-on: ubuntu-latest
    permissions:
      contents: 'read'
      id-token: 'write'

    steps:
      - uses: actions/checkout@v4

      - uses: actions/setup-python@v4
        with:
          python-version: '3.9.x'

      - uses: 'google-github-actions/auth@v2'
        with:
          credentials_json: ${ secrets.DBT_GOOGLLE_BIGQUERY_KEYFILE_PROD }}

      - run: |
        pip install dbt-bigquery
        dbt deps

      - run: |
        dbt run -t prod

      - run: |
        dbt test -t prod
```

```
push.yml

name: On Schedule

on:
  schedule:
    - cron: '20 2 * * 1-5'
  workflow_dispatch:
    inputs:
      on_target:
        description: Choose the target to be used
        default: 'preprod'
        options:
          - preprod
          - prod
        required: true
        type: choice

jobs:
  on_preprod:
    if: contains(fromJSON(['preprod', '']), github.event.inputs.on_target)
    runs-on: ubuntu-latest
    timeout-minutes: 30

    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v4
        with:
          python-version: '3.9.x'
      - uses: 'google-github-actions/auth@v2'
        with:
          credentials_json: ${ secrets.DBT_GOOGLLE_BIGQUERY_KEYFILE_CI }}
      - run: |
        pip install dbt-bigquery
        dbt deps
      - run: |
        dbt run -t preprod

  on_prod:
    if: contains(fromJSON(['prod', '']), github.event.inputs.on_target)
    runs-on: ubuntu-latest
    timeout-minutes: 30

    steps:
      - uses: actions/checkout@v4
      - uses: actions/setup-python@v4
        with:
          python-version: '3.9.x'
      - uses: 'google-github-actions/auth@v2'
        with:
          credentials_json: ${ secrets.DBT_GOOGLLE_BIGQUERY_KEYFILE_CI }}
      - run: |
        pip install dbt-bigquery
        dbt deps
      - run: |
        dbt run -t prod -m mart+
```

Built-in lineage and documentation

Source: github.com/bogumilo/boostrank



Tools?



What next?

Download notes: ai.bogumilo.co/dbt.pdf

dbt for Machine Learning

- [dbt_ml_preprocessing](#) package for light feature engineering
- [Article on ELT-ML pipeline](#) inside dbt and how ML team could benefit
- [Meetup talk video](#) by Sam Swift, VP of Data & AI at Bowery Farming
- [Data Science use cases breakdown](#) directly from the dbt Labs

dbt + Airflow

- [Cosmos package](#) for creating Airflow tasks/jobs from dbt project
- [dbt+Airflow](#) by Astronomer authors of the package

dbt + Dagster

- [Dagster + dbt](#) - official article introducing to dbt support

Write-Audit-Publish

- [WAP: dbt + BigQuery](#) - intro video
- [WAP: dbt + BigQuery](#) - PDF from the above presentation
- [Conf talk referencing WAP strategy](#) by employees video intro by Michelle Ufford, Staff Engineer at Netflix

dbt-core technicalities

- [analytics engineering handbook](#) by Hify Labs
- [Talk on custom CI setup for dbt](#) with automated reports a'la terraform plan in GitHub Actions
- [Article on User Defined Functions](#) managed inside dbt project and included in lineage

other

- [DataOps Manifesto](#) that you can sign on
- [awesome-dbt](#) collection on GitHub

Companies deploying and utilizing dbt

Company	Location	Technologies and Tools	Founding Year	Clients	Website
Rittman Analytics	London, UK	dbt, Looker, Snowflake, Google BigQuery, Segment,	2017	Rittman Analytics Clients	rittmananalytics.com

Thanks!

LI: [/in/pawelbogumilo](https://www.linkedin.com/in/pawelbogumilo)

E: pawel@bogumilo.co



QR for feedback

