Write-Audit-Publish Pattern in Modern Data Pipelines

2024-04-05

Tomas Peluritis

Data Engineer @ WIX

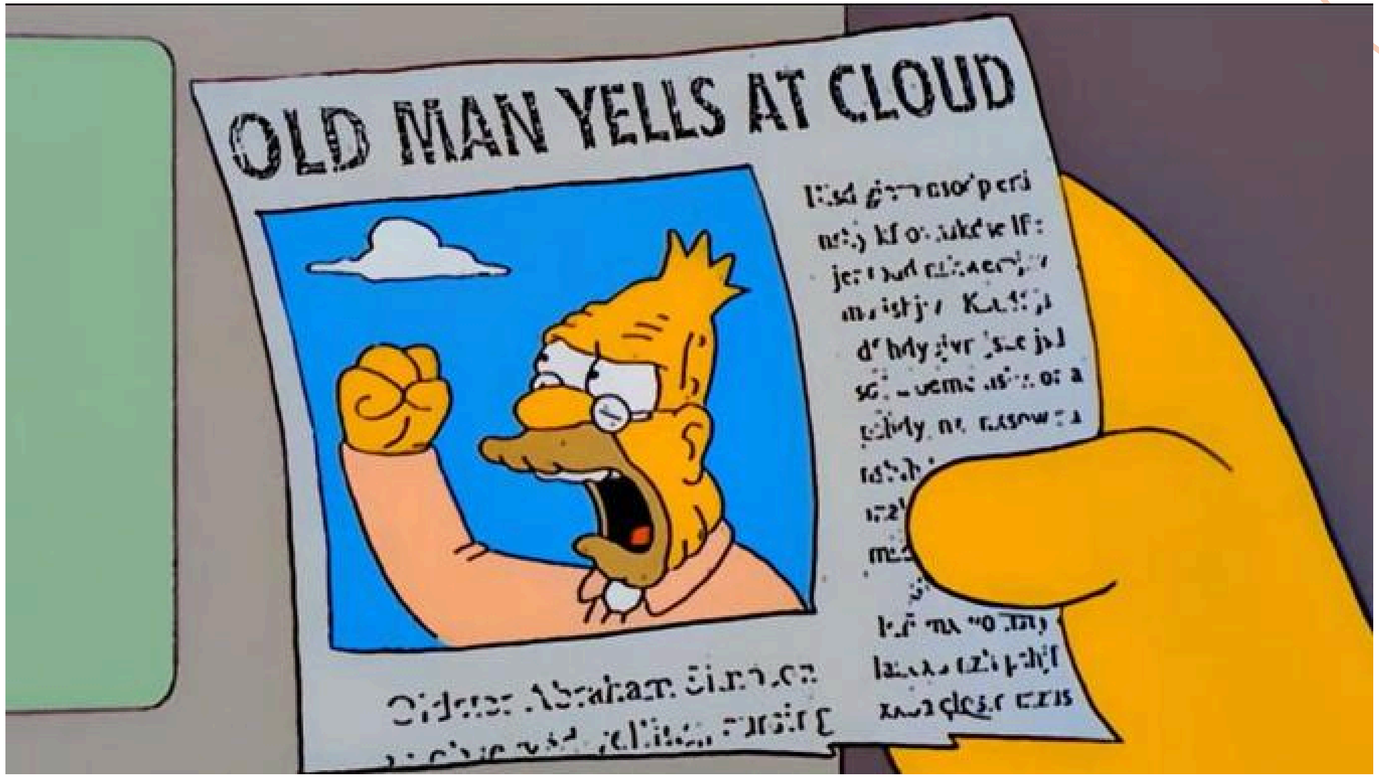
AKA Uncle Data

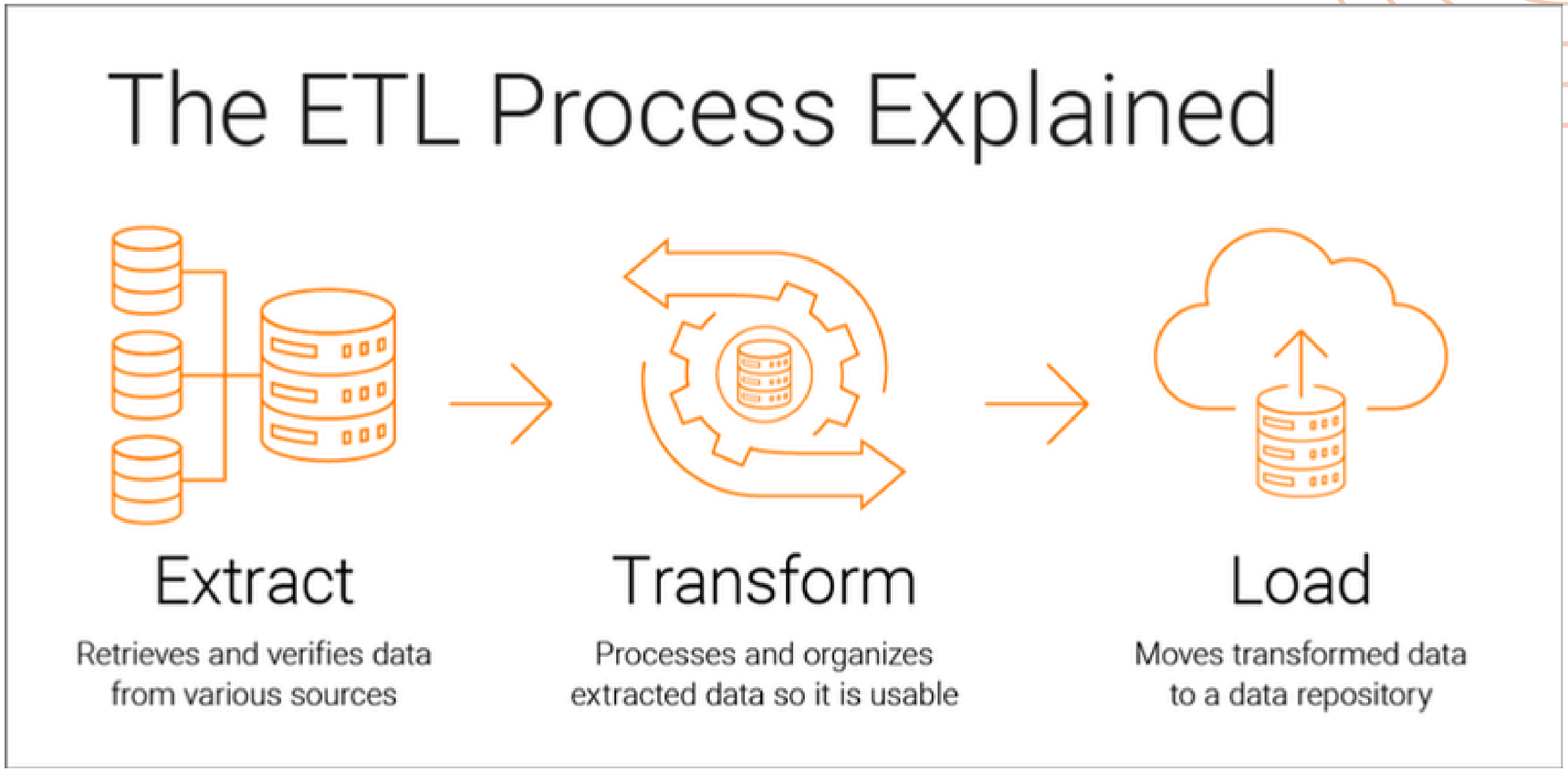
https://www.linkedin.com/in/tomaspeluritis/

https://podcasters.spotify.com/pod/show/duomenu-dede

https://uncledata.substack.com

Back in the early times...


OLD MAN YELLS AT CLOUD


The ETL Process Explained
Extract
Retrieves and verifies data from various sources
Transform
Processes and organizes extracted data so it is usable
Load
Moves transformed data to a data repository

Source

Database Constraints

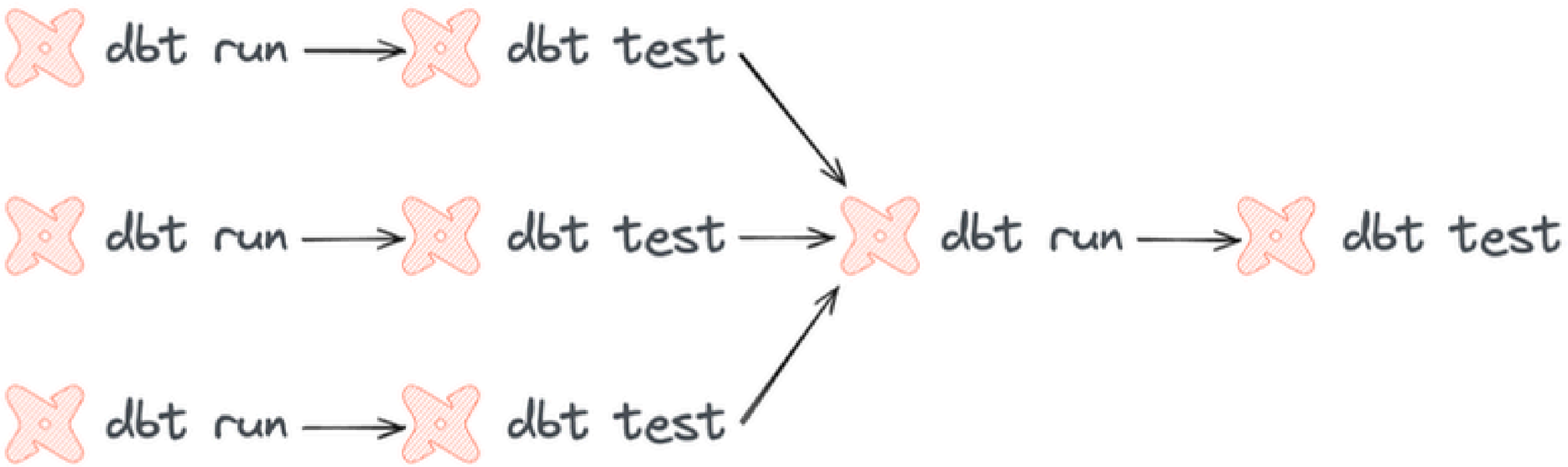
Entity integrity - Primary key, Unique, Not Null, etc.

Domain Integrity - Check and Default constraints

Referential integrity - Foreign key

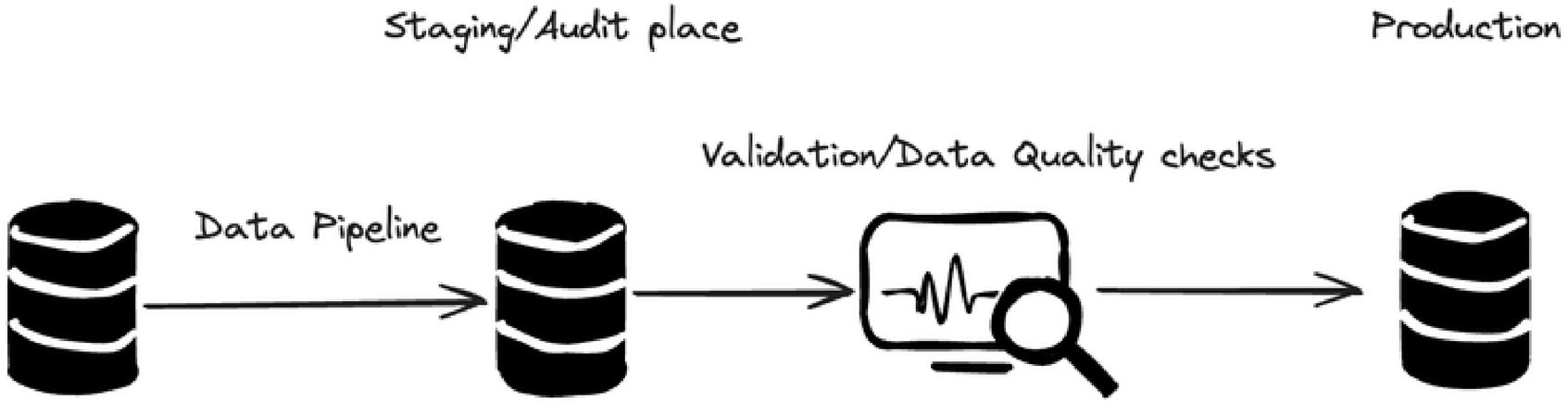
User-defined Integrity - Indexes, Stored Procedures, triggers

Data Pipelines Currently


dbt run
dbt test
dbt run
dbt test
dbt run
dbt test
dbt run
dbt test

Problems

What's Write-Audit-Publish?


Staging/Audit place
Production
Validation/Data Quality checks
Data Pipeline

Potential solutions

DIY in DataFrames

Snowflake - Zero Copy Clone

Apache Iceberg - Branches

DIY in DataFrames

Write-Audit-Publish by Dagster

```python
import pandas as pd
import glob
import duckdb

# Collect all CSV files in the directory
files = glob.glob('sales_data_*.csv')

# Buffer to temporarily store data from each file
data_frames = []

for file in files:
    df = pd.read_csv(file)
    data_frames.append(df)

# Combine all data into a single DataFrame
combined_data = pd.concat(data_frames,
ignore_index=True)
```

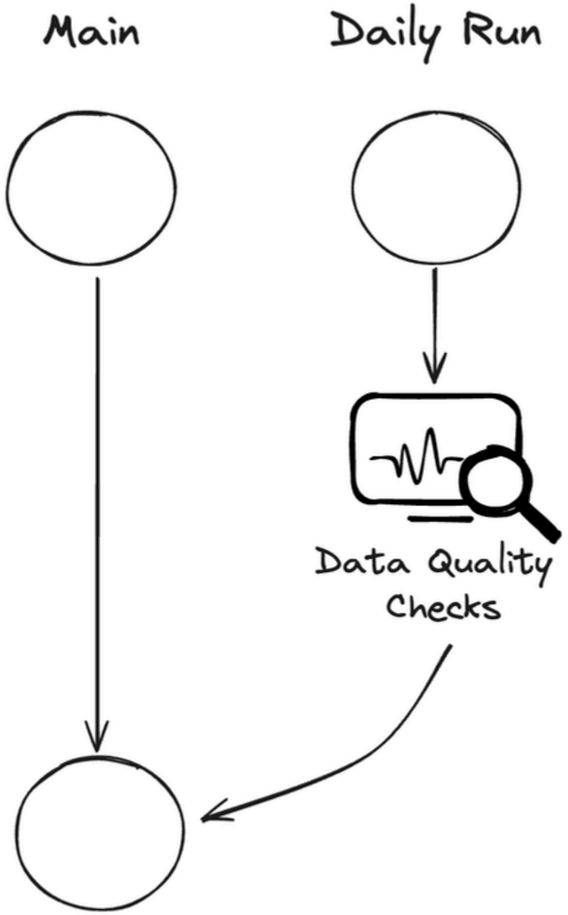
```
# Filter out rows with missing 'sales_amount'
clean_data = combined_data.dropna(subset=
['sales_amount'])
# Filter out rows with negative 'sales_amount'
clean_data = clean_data[clean_data['sales_amount'] >= 0]
```

```python
# Connect to DuckDB
conn = duckdb.connect(database=':memory:',
read_only=False)
if not clean_data.empty:
    clean_data.to_sql('cleaned_sales_data',
                      conn,
                      if_exists='replace',
                      index=False)
    print(f"Inserted {len(clean_data)} rows into DuckDB.")
else:
    print("No data passed the audit.")
```

Snowflake - Zero Copy Clone


dbt run
dbt test
snowflake COPY

Apache Iceberg: Branches


Main
Daily Run
Data Quality
Checks

```
spark.sql("""
          ALTER TABLE staging.aggregated_taxi_per_vendor_date
          SET TBLPROPERTIES ('write.wap.enabled'='true')""")
          )


spark.sql("""
          ALTER TABLE staging.aggregated_taxi_per_vendor_date
          CREATE BRANCH manual_hot_fix""")
          )

spark.conf.set('spark.wap.branch', 'manual_hot_fix')

spark.sql("""delete from staging.aggregated_taxi_per_vendor_date
          where vendor_id = 6""")

spark.sql("""CALL prod.system.fast_forward('prod.staging.aggregated_taxi_per_vendor_date', 'main', 'manual_hot_fix');""")
```

To WAP or not to WAP?

- **Works with any data quality tool**
- **The complete table state is available**
- **Humans are in the loop**
- **Validate multiple tables**

- **DIY in the majority of tooling**

- **Very little knowledge of what's WAP in the industry**

Questions?

Tomas Peluritis

Data Engineer @ WIX

AKA Uncle Data

https://www.linkedin.com/in/tomaspeluritis/

https://podcasters.spotify.com/pod/show/duomenu-dede

https://uncledata.substack.com