# Retail Sales Performance Tracker

# Technical Documentation

## 1. Project Overview

This document provides an in-depth technical explanation of the Retail Sales Performance Tracker, which is an end-to-end analytics pipeline that ingests retail data, transforms it for analysis, and visualizes KPIs through an interactive dashboard. It includes engineering decisions, setup steps, configuration files, code structure, testing logic, CI/CD workflow, and encountered issues. The solution simulates how modern data teams build business intelligence systems using modular, cloud-friendly, and open-source tools.

## 2. Project Goals

- Ingest real retail sales data from a CSV file
- Load the cleaned data into a cloud-based warehouse (Snowflake)
- Build staging and fact models using dbt
- Aggregate sales, profit, and order metrics by time and category
- Apply time-series transformations (rolling average, YTD)
- Visualize business metrics in a Streamlit dashboard
- Add CI/CD validation using GitHub Actions

## 3. Dataset Summary

- Source: Kaggle – Superstore Sales Dataset
- Structure: Tabular CSV with ~10,000 rows
- Fields include:
    - Order ID, Customer ID, Order Date, Ship Date
    - Product Name, Category, Sub-Category
    - Region, State, Country, City, Postal Code
    - Sales, Profit, Quantity, Discount

This data is highly relatable and relevant for demonstrating retail business analytics.

## 4. Technology Stack and Why Each Was Chosen

| Component | Tool/Service | Justification |
|---|---|---|
| Data Warehouse | Snowflake | Scalable, supports SQL-based modeling, free trial credits |
| Data Transformation | dbt (data build tool) | Enables modular, tested SQL pipelines with documentation |
| Scripting | Python | Industry standard for automation and ingestion |

| | | |
|---|---|---|
| Dashboarding | Streamlit | Easy-to-build, interactive, Python-native dashboards |
| Version Control | Git + GitHub | Standard practice for team collaboration |
| CI/CD Automation | GitHub Actions | Cloud-native pipeline automation without cost |

## 5. File and Folder Structure

```
Retail-Sales-Performance-Tracker/
├── data/                    # Raw CSV file (excluded from Git)
├── dbt_project/             # dbt config, models, schema.yml
├── diagrams/                # Dashboard preview screenshots
├── scripts/                 # Data upload Python script
├── streamlit_dashboard/     # Dashboard + Snowflake connector
├── .github/workflows/       # CI pipeline YAML for dbt run/test
├── .env                     # Snowflake credentials (ignored)
├── .gitignore
├── README.md
└── requirements.txt
```

## 6. Data Ingestion: Python to Snowflake

### File: `scripts/upload_to_snowflake.py`

**Workflow:**

1. Read the CSV into a pandas DataFrame
2. Standardize column names (e.g., `Order ID` → `order_id`)
3. Connect to Snowflake using `snowflake.connector`
4. Write the DataFrame into `RAW.RAW_SUPERSTORE` using `write_pandas()`

**Code Highlights:**

```
from snowflake.connector.pandas_tools import write_pandas
...
df.columns = df.columns.str.strip().str.lower().str.replace(" ", "_")
write_pandas(conn, df, table_name='RAW_SUPERSTORE')
```

**.env Example:**

```
SNOWFLAKE_USER=your_username
SNOWFLAKE_PASSWORD=your_password
SNOWFLAKE_ACCOUNT=goyjbep-ib68545
SNOWFLAKE_WAREHOUSE=DEMO_WH
SNOWFLAKE_DATABASE=SUPERSTORE_DB
SNOWFLAKE_SCHEMA=RAW
```

### Issues & Resolutions

- `UnicodeDecodeError` - Used `encoding='ISO-8859-1'`
- `invalid identifier` - Removed quotes and renamed columns cleanly
- `pyarrow` version mismatch - Installed compatible version or downgraded

## 7. dbt Modeling (Transformations)

### Models Built

| File | Description |
|---|---|
| `stg_orders.sql` | Clean staging of raw orders |
| `fct_sales.sql` | Aggregated metrics by category and region |
| `fct_sales_by_month.sql` | Monthly breakdown |
| `fct_sales_rolling_avg.sql` | 3-month rolling avg using `window functions` |
| `fct_sales_by_month_ytd.sql` | YTD sales and profit cumulative model |

### Testing via `schema.yml`

```
- name: fct_sales
  columns:
    - name: category
      tests:
        - not_null
    - name: total_sales
      tests:
        - not_null
```

### Documentation & Lineage

```
dbt docs generate
dbt docs serve --port 9000
```

- Visualizes table dependencies
- Describes models and test coverage

### Issues & Resolutions

- dbt not detecting profile - Recreated `~/.dbt/profiles.yml`
- dbt Cloud blocking CLI - Switched to `dbt-snowflake` via pip
- Port 8080 busy - Used `--port 9000`

## 8. Streamlit Dashboard

### Main File: `streamlit_dashboard/app.py`

**Features Implemented:**

- **KPI Cards**: Total Sales, Total Profit, Total Orders (via `st.metric()`)
- **Sales by Category**: Bar chart using `groupby()`
- **Rolling Average**: 3-month average via `fct_sales_rolling_avg`
- **YTD Sales**: Cumulative line chart from `fct_sales_by_month_ytd`

### Helper File: `snowflake_connector.py`

- Uses `load_dotenv()` to read `.env`
- Secure Snowflake connection
- Queries raw/fact/rolling/ytd tables separately

### Issues & Fixes

- `NoneType has no attribute find` - Fixed `.env` not loading
- Chart missing - Ensured Streamlit reloads after changes

## 9. GitHub Actions: CI/CD Workflow

### File: `.github/workflows/dbt_ci.yml`

**Trigger**: Manual (`workflow_dispatch`)

**Workflow Steps:**

```
- Setup Python 3.10
- Install dbt-snowflake
- Load DBT_PROFILE from GitHub Secrets
- Run dbt debug, dbt run, dbt test
```

**GitHub Secret Used:** `DBT_PROFILE` → full contents of `profiles.yml`

### Issues & Fixes

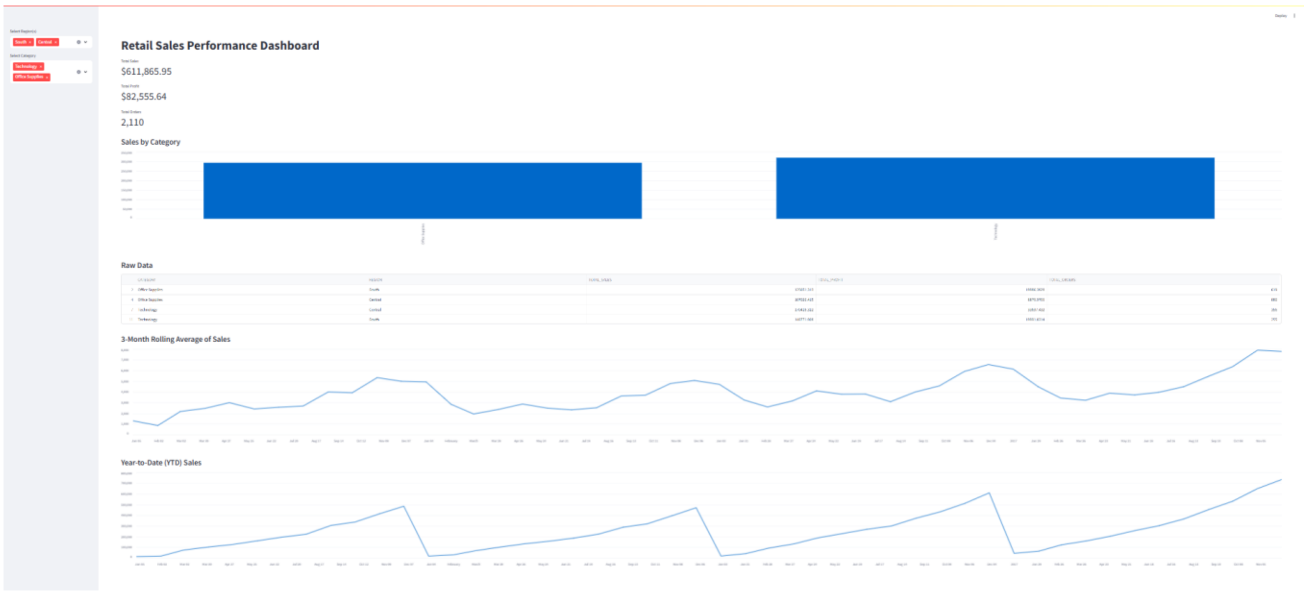
- `push rejected` - Pulled remote first: `git pull origin main --allow-unrelated-histories`
- Profile secret not found - Re-pasted clean YAML into GitHub secrets

## 10. Dashboard Preview

Visuals Displayed:

- KPI cards
- Bar chart: Sales by Category

- Line chart: 3-Month Rolling Avg
- Line chart: YTD Sales



Retail Sales Performance Dashboard

Total Sales
$611,865.95

Total Profit
$82,555.64

Total Orders
2,110

Sales by Category

Raw Data

| | CATEGORY | REGION | TOTAL_SALES | TOTAL_PROFIT | TOTAL_ORDERS |
|---|---|---|---|---|---|
| [illegible] | Office Supplies | South | [illegible] | [illegible] | [illegible] |
| [illegible] | Office Supplies | Central | [illegible] | [illegible] | [illegible] |
| [illegible] | Technology | Central | [illegible] | [illegible] | [illegible] |
| [illegible] | Technology | South | [illegible] | [illegible] | [illegible] |

3-Month Rolling Average of Sales

Year-to-Date (YTD) Sales

## 11. Testing Strategy

### dbt Tests

- Defined in `schema.yml`
- Test types: `not_null`, `unique`
- Covers primary keys, aggregates, and time columns

### Manual Checks

- Compared outputs between models
- Validated counts in Streamlit vs Snowflake

**Repository**: Retail-Sales-Performance-Tracker