Task 3 - Modern Tooling

==========

Why DBT over SSIS

==========

1. Easy Testing and Validations :

DBT supports built-in tests like not_null/unique test - in .yml.

It supports - custom data quality checks - singular tests

2. DBT ELT vs SSIS ETL :

DBT follows ELT (Extract-Load-Transform) model.

It uses warehouse compute power - which execute all transformations directly within the DWH

This ensures better scalability and performance.

3. Easier to Deploy, Version control and Automate :

DBT integrates easily with Airflow - for orchestration/job scheduling, GitHub - for version controlling

4. DBT is Open Source and Cloud Specific :

DBT-Core is open source and designed specifically for cloud data warehouses

5. Data Lineage :

DBT generates data lineage for both tables and individual columns, making it easy to trace how data flows through the pipeline.

This helps developers quickly identify the original source of any column used in downstream models/data marts,

which also makes debugging easier.

6. Code Reusability :

DBT ref() makes the code reusable especially in layered models (Bronze → Silver → Gold) with clear dependencies.

==========

NOTE:

I've worked with both dbt Core integrated with Airflow, as well as dbt Cloud.

Here's a comparison of the advantages of using dbt Core with Airflow versus dbt Cloud

Below are the key advantages and differences between both approaches:

==========

Option 1 : DBT-Core for ELT with Airflow for orchestration / scheduling

****

1. Flexible and Cost Effective

Both DBT Core and Airflow are open-source, which offers full control on the tools without licensing costs.

They are flexible to deply in cloud.

2. DBT ELT Layering

dbt Core enables clean, version-controlled SQL transformations directly on DWH. Layering in DBT helps in

-> Breaks complex logic into small, reusable models

-> Reduces duplication by allowing shared logic across downstream models.

3. Easy Scheduling

Airflow helps to schedule and chain dbt runs with full control - which include retry , check for dependancy runs etc.

4. Easy Git Integration

Both Airflow and dbt Core can be easily integrated with Git - for version controlling.

Airflow can also be integrated with Slack or PagerDuty to notify whenever there is failure in pipeline.

==========

Option 2 : DBT-Cloud

****

1. Built-in Scheduler

No need for external tools like Airflow — schedule dbt runs directly using DBT Cron Jobs.

2. User Friendly Web UI

Easily develop, run, and monitor dbt models without using the command line.

3. Centralized Monitoring

All runs, logs, and errors cab be tracket in one place.

4. Seamless Git Integration

We can push, merge, and deploy into GitHub/GitLab directly through the dbt Cloud interface.

==========

Proposed Migration from SSIS to DBT

****

1. Choose a Cloud DWH Platform and move the raw data into DWH

Use a Scripts / Cloud Data Transfer Services / ETL Tools to load source data from SSIS into cloud DWH warehouse.

2. Convert SSIS logic into DBT models

Design the DBT model structure and rewrite all the transformation logic in SQL into DBT layers.

3. Add Respective data Tests

Use dbt's built-in features to check data quality and describe each model and column.

4. Schedule and run dbt

Use dbt Cloud or Airflow to run dbt models.

==========

Risks in Changing Position Data Processing

****

1. Incorrect Financial Calculations

Changes in current data logic could lead to incorrect totals.

Instead of directly midifying the existing pipeline, build a new pipeline with the logic changes,

which correct the known existing issues and monitor the % discrepancies with the existing pipeline.

2. Broken Reports and Dashboards

There are chances that the downstream report data may break if these is any change in upstream source.

To prevent this, introduce the new logic fields into the report,

explain the end-users regarding the discrepancies old vs new - and eventually get rid of the old-incorrect fields.

3. Reconciliation Issues

Differences between old (SSIS) and new (dbt) outputs may cause confusion or inconsistencies.

Always good to maintain both old and new pipelines, until all the issues are addressed,

communicated to stake-holders and everything is stabilized on the new set-up.

4. Stakeholder Trust & Compliance Risk

Any discrepancies in financial data may impact stakeholder and sometimes create audit concerns.

To avoid such cases -

-> Do not disturb the old data

-> Lift and shift the existing data as it is into DBT

-> Implement the logic changes to incoming data - post clear communication with stake-holders.

==========

Tools Used for Implementation

****

1. Database: SQLite

2. Jupyter Notebook

3. DBT Core