Sora Union Data Engineering Task

Data Warehousing & ETL Process

The project provided us with two CSV files in a Google Sheet, in which we are expected to design a data warehouse and create an ETL process.

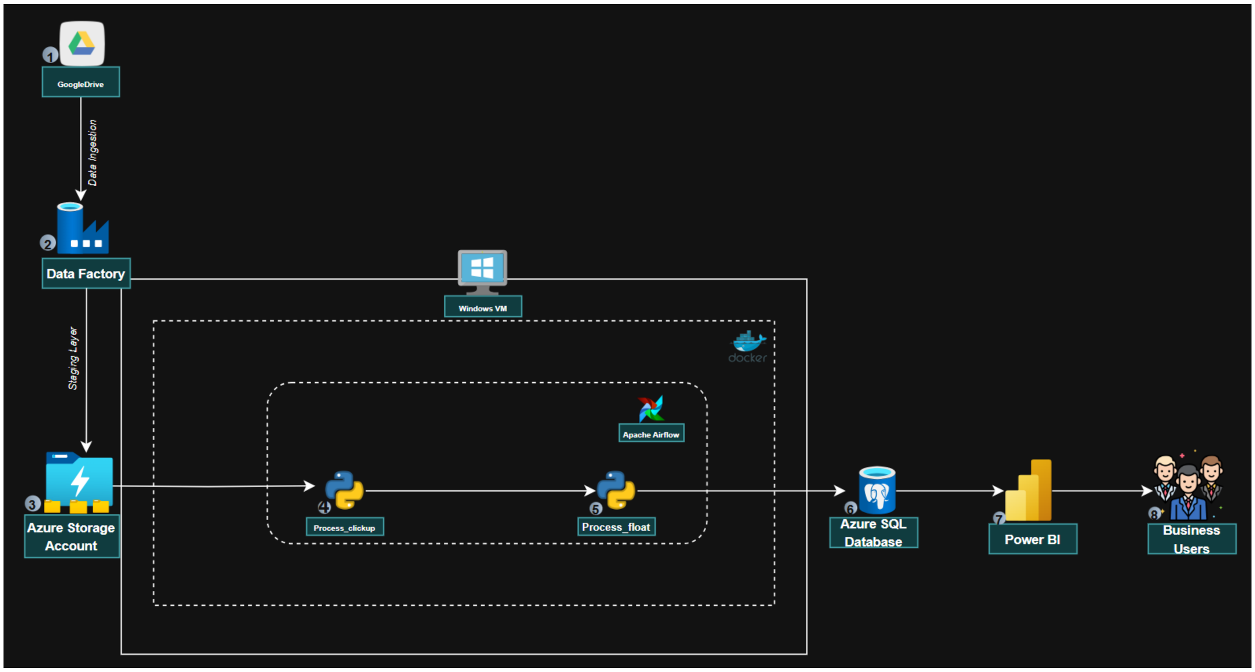
Assumption

Assuming the data comes in a batch process daily to Google Drive and are ingested to a storage account like ***S3 bucket, Google Cloud Storage, or Azure Storage Account (ADLS)*** using ETL tools like Fivetran, AWS Glue, Azure Data Factory or Dataproc***.*** For this project, we will be using the ***Azure Storage Account, data Lake Gen 2*** for storing the ***Float and Clickup*** which will be ingested using the Azure Data Factory.

Since the data gets ingested to the storage account daily, we will be using the concept of getting the latest file (Last Modified Date) in the storage account, performing the necessary transformation and modeling before inserting it into the Azure PostgreSQL Database.

Project Architecture

Based on our early assumption you will notice that data from the Google Drive are ingested to Azure Storage Account (adls) using the Azure Data Factory, we then created a ***DAG*** process in ***Apache Airflow*** for Orchestrating the whole process from ingestion of data to transformation and finally loading into Azure PostgreSQL Database (Flexible Server). This process was set to run at ***7:00 WAT/ GMT+ 1/ UTC +1*** daily



Project Architecture

Data Modeling and Warehouse Design

For this reason, we are going to break it into two parts:

- *Table Creating with Relationship*
- *Entity Relationship Diagram*

Table Creation

Following the Start Schema design approach we are going to break the data into two fact tables and five-dimension tables.

Fact Tables

This contains the incremental tables that will continue to grow on a steady basis. Optimized for analytical queries.

- **fact_time_tracking:** Stores actual time entries from ClickUp
- **fact_allocation:** Stores resource allocation/planning data from Float

Dimension Tables

This is information about the fact table containing necessary information used in the modeling process.

- **dim_client:** Client information
- **dim_project:** Project details with client relationships
- **dim_employee:** Employee information
- **dim_role:** Role definitions
- **dim_date:** Date dimension for time-based analysis

Data Integrity

For Data Integrity we ensured following the best practice by creating references and using the indexing approach for query optimization.

a) Referential Integrity:

- Foreign key constraints on all dimension references
- Ensures data consistency across the warehouse

b) Indexing Strategy:

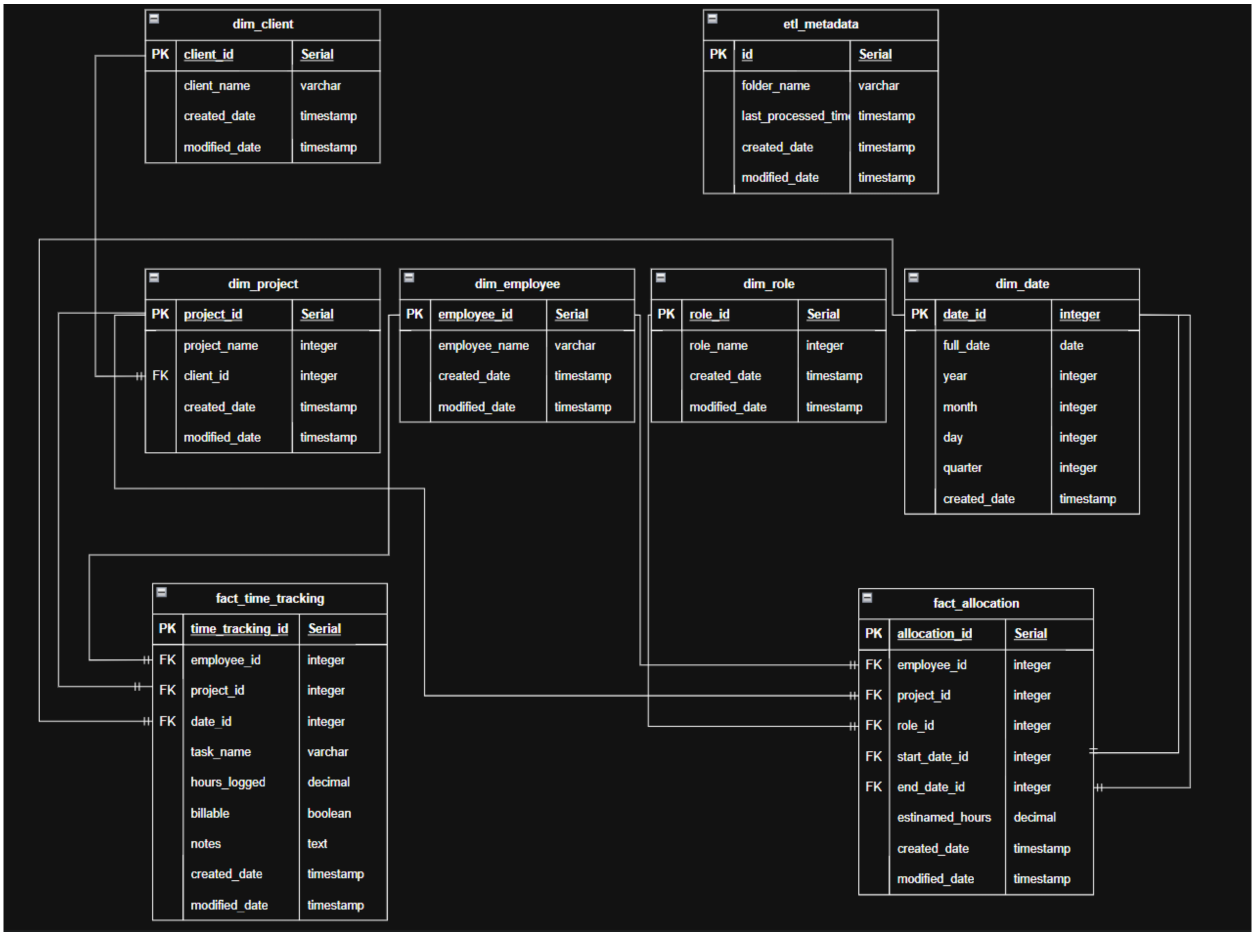
- Primary keys on all tables
- Foreign key indexes for joining optimization
- Composite index on allocation dates
- Created_date and modified_date tracking

c) ETL Metadata

These are standard data used in keeping records and changes.

Entity Relationship Diagram

Entity-relationship (ER) modeling is a visual approach to data modeling used to represent the structure of a database. It is used to identify the "things" (entities) in a system and how they relate to each other.



Project ER Diagram

One-to-Many Relationships:

- A client can have multiple projects (1:M)
- A project can have multiple time tracking entries (1:M)
- A project can have multiple allocations (1:M)
- An employee can have multiple time tracking entries (1:M)
- An employee can have multiple allocations (1:M)
- A role can be used in multiple allocations (1:M)
- A date can be referenced by multiple time tracking entries (1:M)
- A date can be the start or end date for multiple allocations (1:M)

Fact Table Relationships:

fact_time_tracking connects to:

- dim_employee
- dim_project
- dim_date

fact_allocation connects to:

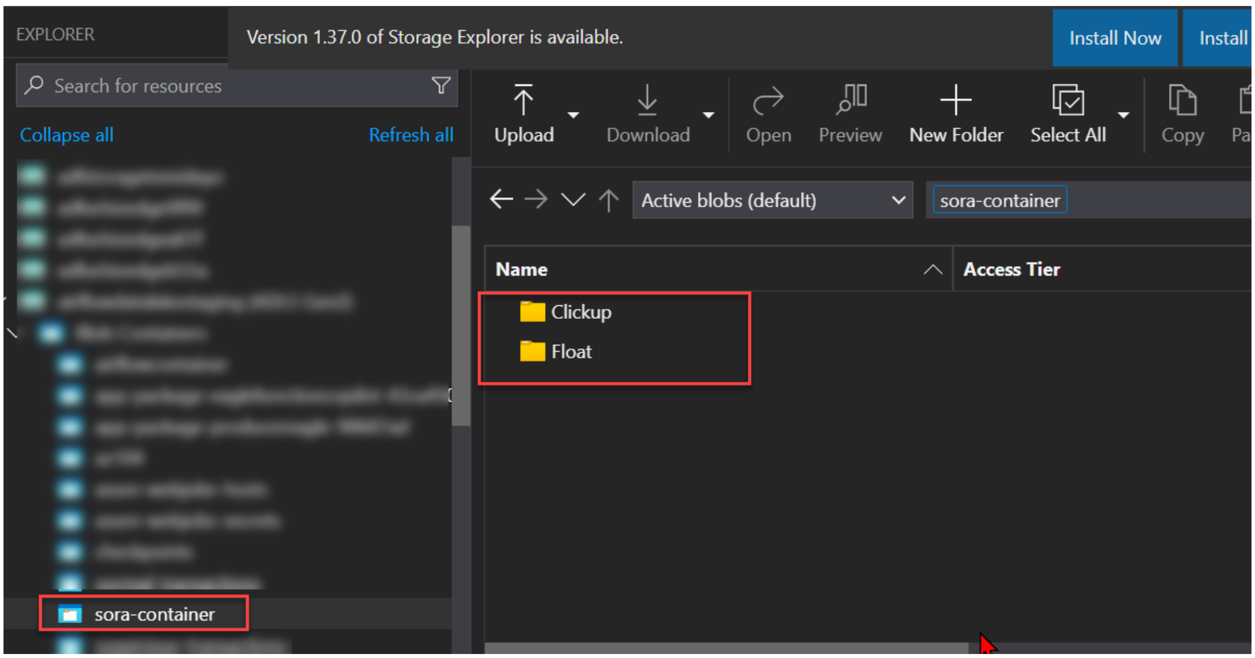
- dim_employee
- dim_project
- dim_role
- dim_date (twice, for start and end dates)

Independent Tables:

- etl_metadata is independent and used for ETL process tracking

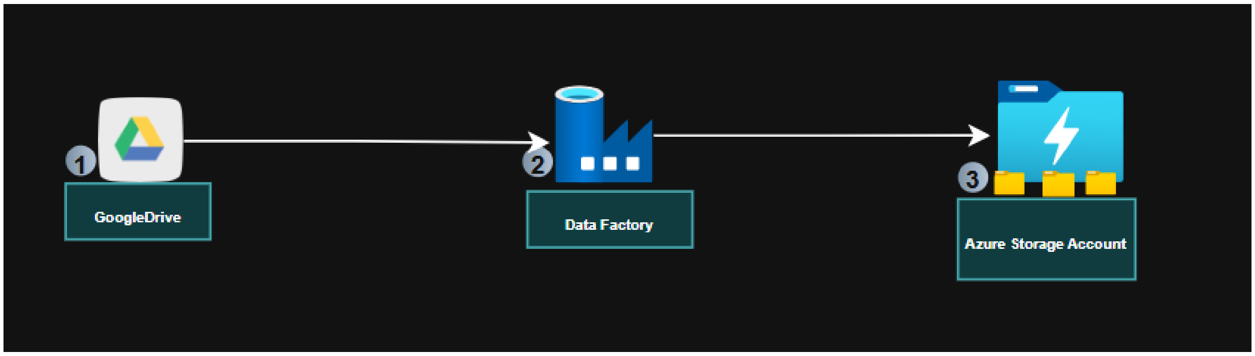
ETL Process and Pipeline Development

This section focuses on the ETL process and pipeline development. Firstly let start by creating two empty folders in our ***container(prefix- adls).*** These are the containers where the ingestion process will start from.



Data Ingestion with Data Factory

At this point we are going to create a pipeline using Azure Data Factory that will ingest the data from Google Drive and store it in our staging layer which is the adls.



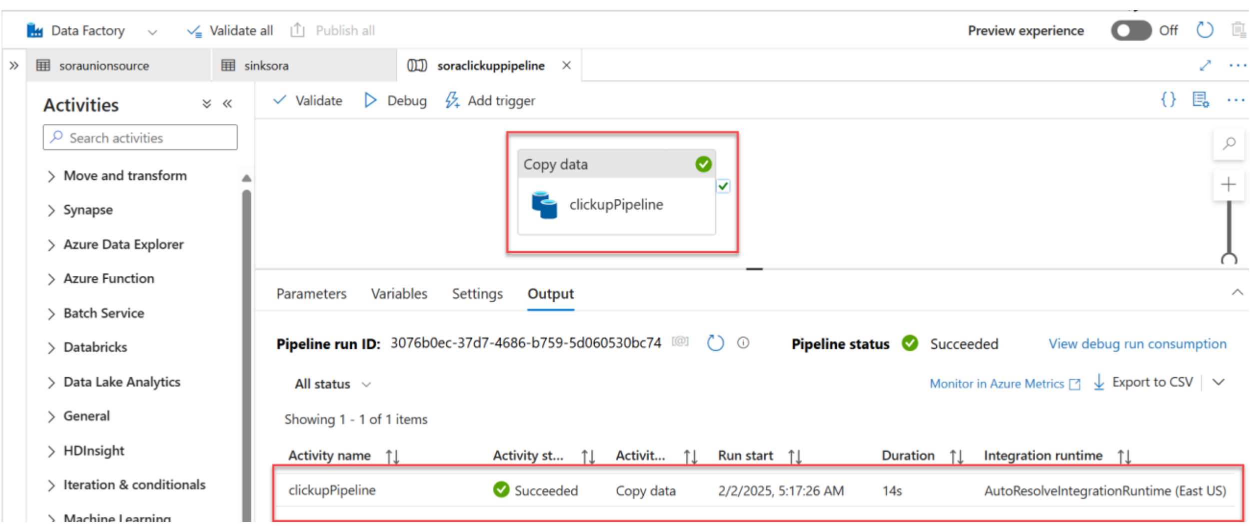
Ingestion Phase

Pipeline Creation

In the Azure Data Factory the following components are created to achieve this.

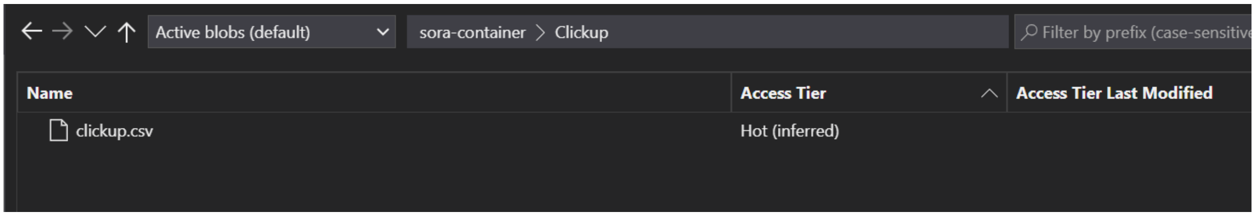
- ***Source Dataset***
- ***Source Linked Service***
- ***Activity (Copy Activity - Google Drive to adls)***
- ***Sink Linked Service***
- ***Sink Dataset***

From the image below you will notice the data copied successfully from Google Drive to ADLS which serves as our storage layer.



Successful Pipeline Run

We can confirm the data in the storage account.



Transformation and Loading Preparation

At this stage we will be developing the transformation logic and creating and orchestration process using Apache Airflow

Pre-requisite

- ***Docker Desktop (Using Windows)***
- ***Virtual Machine/ EC2 Instant (Windows Server)***
- ***WSL - Windows Subsystem for Linux (Applicable for only Windows Users)***
- ***Preferred IDE - VSCode***

Setting Airflow on Docker

By going through the official documentation on Airflow Docker was a straight process. https://airflow.apache.org/docs/apache-airflow/stable/howto/docker-compose/index.html#running-airflow-in-docker

Folder Breakdown

Sora Union/
>> **config**
-- *.env*
-- *config.py*
>> **dags**
-- *etl_dag.py*
-- *etl_process.py*
-- *utils.py*
>> **logs**
>> **plugins**

- **.env:** Environment variables and sensitive configuration. Stores sensitive configuration like:

Azure Storage credentials, PostgreSQL database connection details.

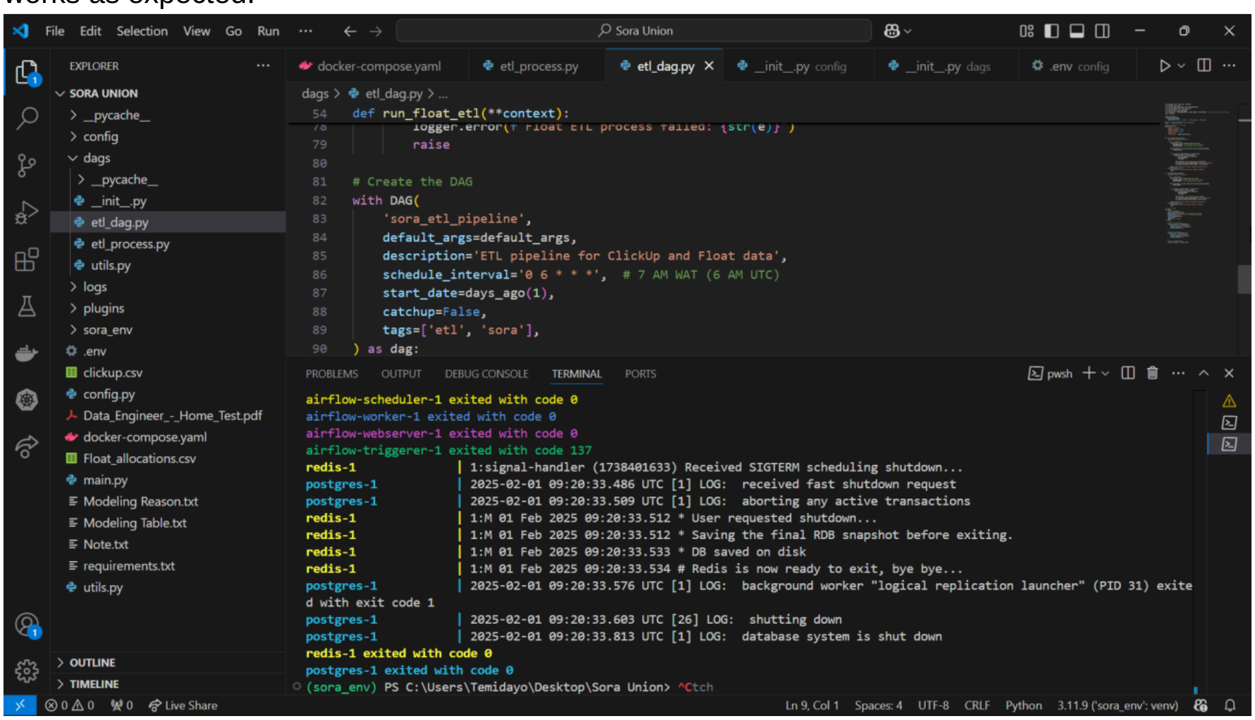
- **config.py:** Configuration loader and settings management. Loads environment variables and defines constants used across the application.
- **etl_dag.py:** Airflow DAG definition and task orchestration.
    - **process_clickup:** Processes ClickUp time tracking data
    - **process_float:** Processes Float resource allocation data
    - Configures scheduling (runs daily at 6 AM UTC)
- **etl_process.py:** Core ETL transformation logic. Contains the ETLProcess class which handles:
    - Dimension table management (clients, projects, employees, roles)
    - Data transformation logic for both ClickUp and Float data
    - Fact table loading for time tracking and resource allocation
    - Incremental loading through metadata tracking
- **utils.py:** Utility classes for Azure and PostgreSQL connections.

Provides utility classes:

- **AzureStorageClient:**
- Handles connections to Azure Data Lake Storage
- Manages file listing and reading
- **PostgresClient:**
- Manages database connections
- Provides methods for querying and data loading

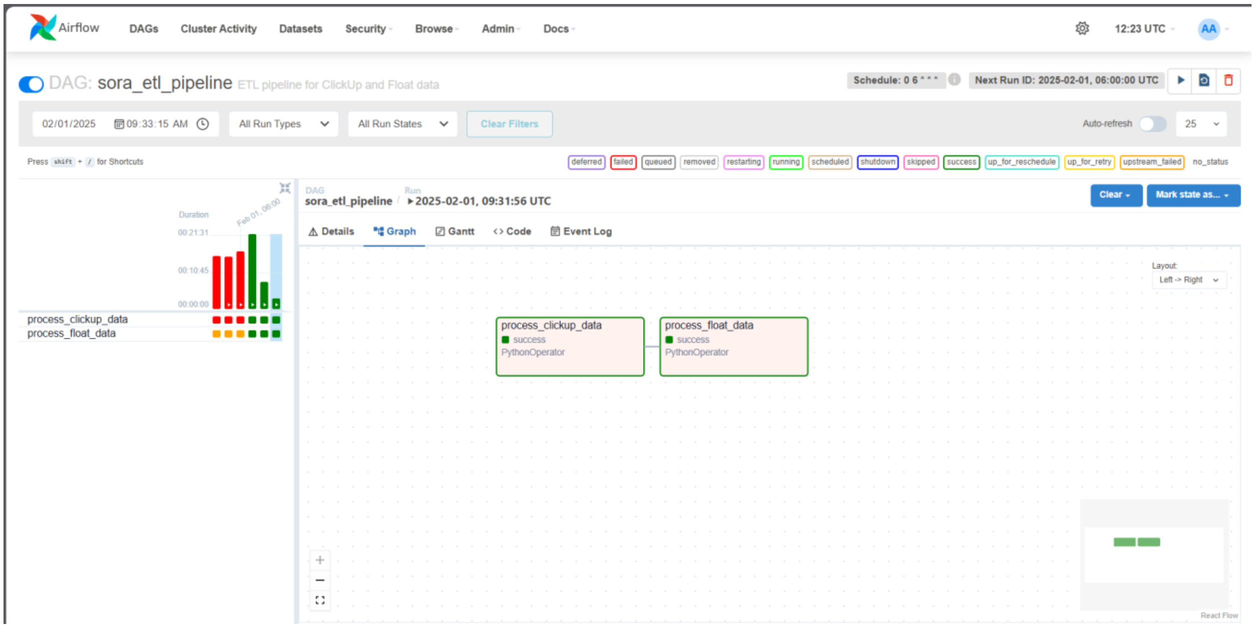
Airflow Orchestration

After successfully setting up your code you can run it on the Airflow Webserver UI and see it works as expected.



```
docker-compose up
```

Process_clickup_data >> process_float_data



Airflow Orchestration

Test and Validate ETL Process

After successfully running the entire ETL process we can test it by using the SELECT statement in PostgreSQL Database.

```sql
select * from datamart.dim_client
```

```
select * from datamart.dim_client
limit 100
```

dim_client 1

select * from datamart.dim_client limit ' | Enter a SQL expression to filter results (use Ctrl+Space)

	client_id	client_name	created_date	modified_date
1	1	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417
2	2	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417
3	3	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417
4	4	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417
5	5	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417
6	6	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417
7	7	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417
8	8	Client 1	2025-02-01 09:32:15.417	2025-02-01 09:32:15.417

select * from datamart.dim_date

```
select * from datamart.dim_date
```

_date 1

ct * from datamart.dim_date | Enter a SQL expression to filter results (use Ctrl+Space)

date_id	full_date	year	month	day	quarter	created_date
20,230,703	2023-07-03	2,023	7	3	3	2025-02-01 09:32:11.223
20,230,704	2023-07-04	2,023	7	4	3	2025-02-01 09:32:11.223
20,230,705	2023-07-05	2,023	7	5	3	2025-02-01 09:32:11.223
20,230,706	2023-07-06	2,023	7	6	3	2025-02-01 09:32:11.223
20,230,707	2023-07-07	2,023	7	7	3	2025-02-01 09:32:11.223
20,230,708	2023-07-08	2,023	7	8	3	2025-02-01 09:32:11.223
20,230,709	2023-07-09	2,023	7	9	3	2025-02-01 09:32:11.223
20,230,710	2023-07-10	2,023	7	10	3	2025-02-01 09:32:11.223
20,230,711	2023-07-11	2,023	7	11	3	2025-02-01 09:32:11.223
20,230,712	2023-07-12	2,023	7	12	3	2025-02-01 09:32:11.223
20,230,713	2023-07-13	2,023	7	13	3	2025-02-01 09:32:11.223
20,230,714	2023-07-14	2,023	7	14	3	2025-02-01 09:32:11.223
20,230,715	2023-07-15	2,023	7	15	3	2025-02-01 09:32:11.223
20,230,716	2023-07-16	2,023	7	16	3	2025-02-01 09:32:11.223

select * from datamart.dim_employee

```sql
select * from datamart.dim_employee
```

_employee 1 ×

ct * from datamart.dim_employee | Enter a SQL expression to filter results (use Ctrl+Space)

employee_id	employee_name	created_date	modified_date
1	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
2	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
3	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
4	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
5	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
6	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
7	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
8	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
9	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
10	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
11	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
12	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
13	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
14	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664
15	Isabella Rodriguez	2025-02-01 09:32:31.664	2025-02-01 09:32:31.664

select * from datamart.dim_project

```sql
select * from datamart.dim_project
```

dim_project 1 ×

select * from datamart.dim_project | Enter a SQL expression to filter results (use Ctrl+Space)

	project_id	project_name	client_id	created_date	modified_date
1	1	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
2	2	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
3	3	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
4	4	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
5	5	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
6	6	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
7	7	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
8	8	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
9	9	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
10	10	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
11	11	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
12	12	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
13	13	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
14	14	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037
15	15	Website Development	1	2025-02-01 09:32:21.037	2025-02-01 09:32:21.037

select * from datamart.dim_role

```
select * from datamart.dim_role
```

dim_role 1

select * from datamart.dim_role | Enter a SQL expression to filter results (use Ctrl+Space)

	role_id	role_name	created_date	modified_date
1	1	Product Designer	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
2	2	Design Manager	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
3	3	Front End Engineer	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
4	4	QA Engineer	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
5	5	Project Manager	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
6	6	Brand Designer	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
7	7	Design Manager	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
8	8	Project Manager	2025-02-01 09:33:06.922	2025-02-01 09:33:06.922
9	9	Localization Specialist UK	2025-02-01 09:33:06.923	2025-02-01 09:33:06.923
10	10	Brand Designer	2025-02-01 09:33:06.923	2025-02-01 09:33:06.923
11	11	Design Manager	2025-02-01 09:33:06.923	2025-02-01 09:33:06.923
12	12	Brand Designer	2025-02-01 09:33:06.923	2025-02-01 09:33:06.923
13	13	Project Manager	2025-02-01 09:33:06.923	2025-02-01 09:33:06.923

```
select * from datamart.fact_allocation
```

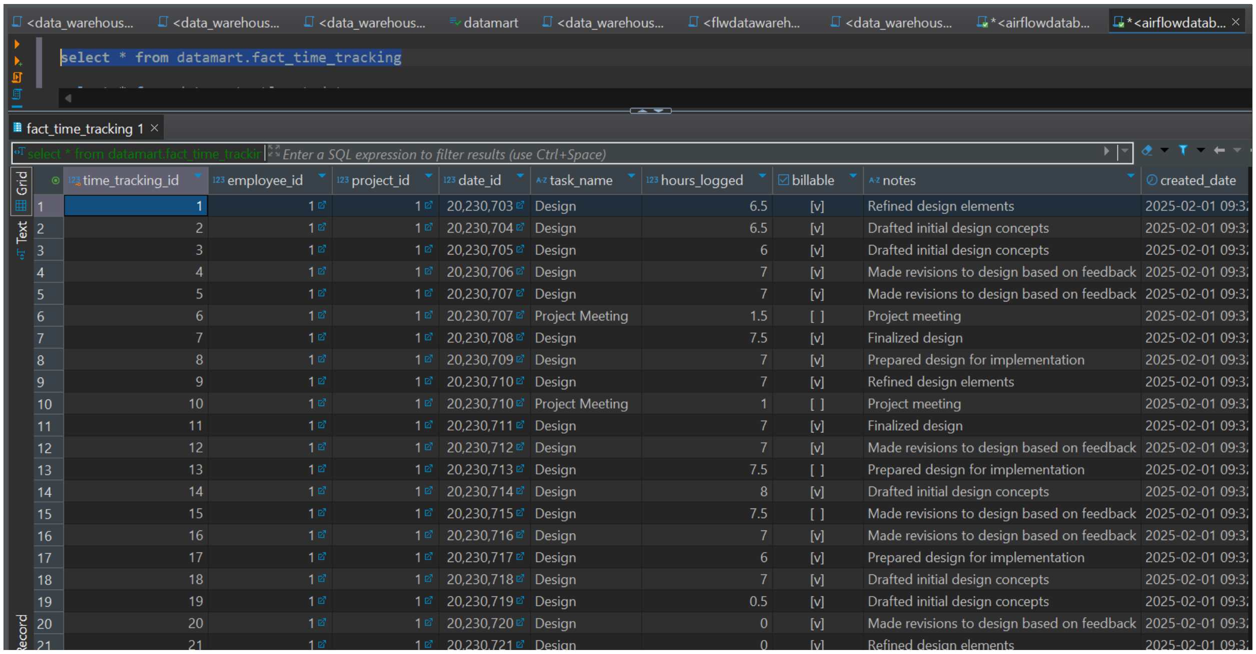
```
select * from datamart.fact_allocation
```

ct_allocation 1

lect * from datamart.fact_allocation | Enter a SQL expression to filter results (use Ctrl+Space)

	allocation_id	employee_id	project_id	role_id	start_date_id	end_date_id	estimated_hours	created_date
1	1	1	1	1	20,230,703	20,230,724	112	2025-02-01 09
2	2	26	1	2	20,230,703	20,230,724	24	2025-02-01 09
3	3	50	1	3	20,230,731	20,230,828	189	2025-02-01 09
4	4	83	1	4	20,230,821	20,230,904	77	2025-02-01 09
5	5	100	1	5	20,230,703	20,230,904	92	2025-02-01 09
6	6	169	169	6	20,230,703	20,230,724	112	2025-02-01 09
7	7	26	169	2	20,230,703	20,230,724	32	2025-02-01 09
8	8	100	169	5	20,230,703	20,230,724	24	2025-02-01 09
9	9	242	242	9	20,230,710	20,230,814	182	2025-02-01 09
10	10	169	242	6	20,230,724	20,230,828	182	2025-02-01 09
11	11	26	242	2	20,230,724	20,230,828	52	2025-02-01 09
12	12	358	242	6	20,230,724	20,230,828	0	2025-02-01 09
13	13	100	242	5	20,230,710	20,230,828	36	2025-02-01 09

```
select * from datamart.fact_time_tracking
```

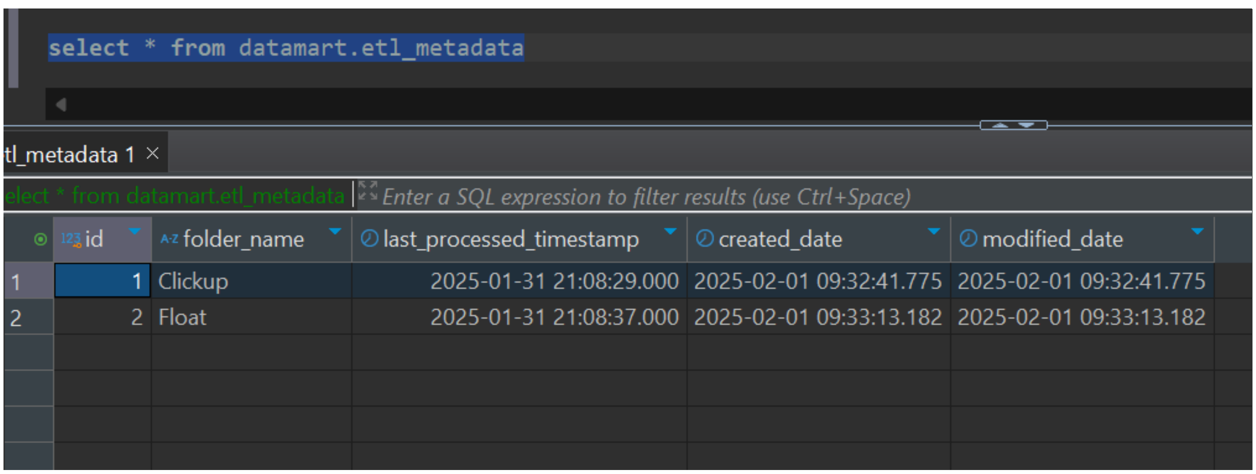


select * from datamart.fact_time_tracking

fact_time_tracking 1

	time_tracking_id	employee_id	project_id	date_id	task_name	hours_logged	billable	notes	created_date
1	1	1	1	20,230,703	Design	6.5	[v]	Refined design elements	2025-02-01 09:3
2	2	1	1	20,230,704	Design	6.5	[v]	Drafted initial design concepts	2025-02-01 09:3
3	3	1	1	20,230,705	Design	6	[v]	Drafted initial design concepts	2025-02-01 09:3
4	4	1	1	20,230,706	Design	7	[v]	Made revisions to design based on feedback	2025-02-01 09:3
5	5	1	1	20,230,707	Design	7	[v]	Made revisions to design based on feedback	2025-02-01 09:3
6	6	1	1	20,230,707	Project Meeting	1.5	[ ]	Project meeting	2025-02-01 09:3
7	7	1	1	20,230,708	Design	7.5	[v]	Finalized design	2025-02-01 09:3
8	8	1	1	20,230,709	Design	7	[v]	Prepared design for implementation	2025-02-01 09:3
9	9	1	1	20,230,710	Design	7	[v]	Refined design elements	2025-02-01 09:3
10	10	1	1	20,230,710	Project Meeting	1	[ ]	Project meeting	2025-02-01 09:3
11	11	1	1	20,230,711	Design	7	[v]	Finalized design	2025-02-01 09:3
12	12	1	1	20,230,712	Design	7	[v]	Made revisions to design based on feedback	2025-02-01 09:3
13	13	1	1	20,230,713	Design	7.5	[ ]	Prepared design for implementation	2025-02-01 09:3
14	14	1	1	20,230,714	Design	8	[v]	Drafted initial design concepts	2025-02-01 09:3
15	15	1	1	20,230,715	Design	7.5	[ ]	Made revisions to design based on feedback	2025-02-01 09:3
16	16	1	1	20,230,716	Design	7	[v]	Made revisions to design based on feedback	2025-02-01 09:3
17	17	1	1	20,230,717	Design	6	[v]	Prepared design for implementation	2025-02-01 09:3
18	18	1	1	20,230,718	Design	7	[v]	Drafted initial design concepts	2025-02-01 09:3
19	19	1	1	20,230,719	Design	0.5	[v]	Drafted initial design concepts	2025-02-01 09:3
20	20	1	1	20,230,720	Design	0	[v]	Made revisions to design based on feedback	2025-02-01 09:3
21	21	1	1	20,230,721	Design	0	[v]	Refined design elements	2025-02-01 09:3

```
select * from datamart.etl_metadata
```



select * from datamart.etl_metadata

tl_metadata 1

	id	folder_name	last_processed_timestamp	created_date	modified_date
1	1	Clickup	2025-01-31 21:08:29.000	2025-02-01 09:32:41.775	2025-02-01 09:32:41.775
2	2	Float	2025-01-31 21:08:37.000	2025-02-01 09:33:13.182	2025-02-01 09:33:13.182

Conclusion

To get a high level of accuracy the following process where followed, having designed the model, had a high level of data quality change and improved the query by optimizing using index on certain columns.