Capstone Project: TPCDS Group 10

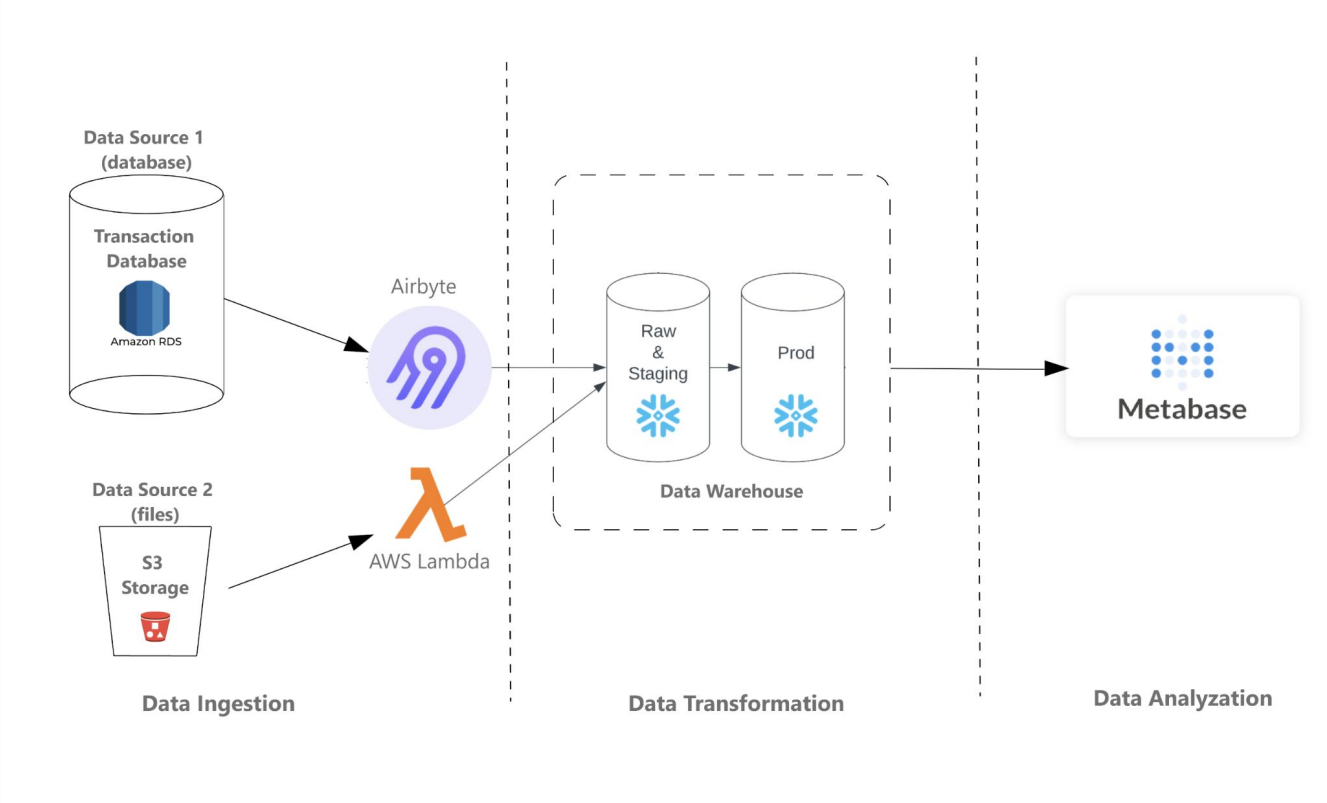
Maram Alshehri
Afnan Alshehri
Munira Alhumaidan

Table of contents

01

Project Overview

Project Overview



02

About Data

About Data

- Dataset Source: TPCDS, a widely recognized dataset designed for database testing with a focus on Retail Sales data.
- Dataset Components:
  - Fact Tables:

    Catalog_Sales, Web_Sales, Inventory

  - Dimension Tables:

    Date_Dim, Customer, Item, Promotion, Customer_Demographics, Call_Center, Customer_Address, Catalog_Page, Warehouse, Time_Dim, Ship_Mode, Household_Demographics, Income_Band, Web_Page, Web_Site: Providing detailed information about various aspects such as customers, warehouses, items, promotions, and more.

About Data

- Data Storage:
  - Postgres DB (AWS RDS): Stores all tables except the Inventory table; refreshed daily with the latest sales data using ETL processes.
  - S3 Bucket: Houses the Inventory table; a new file with the latest data is added daily, reflecting data typically registered at the end of each week (one entry per item per warehouse).

03

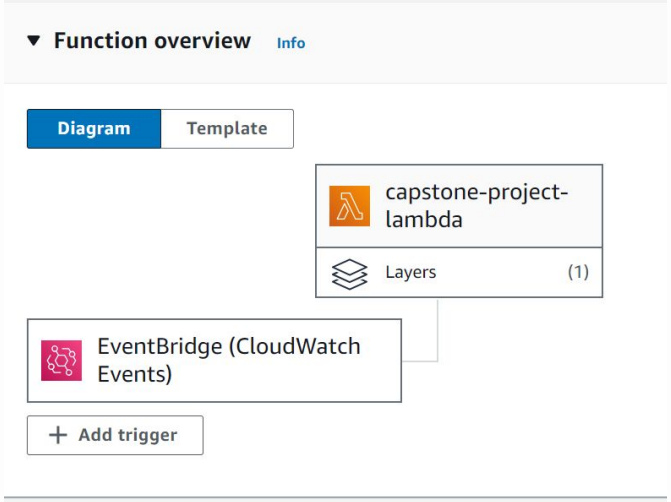
Data Ingestion

Data Ingestion from AWS S3 Bucket

- Transfer inventory data from the AWS S3 bucket to the Snowflake data warehouse.
- AWS S3 Bucket: containing inventory data.
- Data Transfer Method: AWS Lambda:

  Facilitates the connection to the AWS S3 bucket.

  Extracts the inventory.csv file from the S3 bucket and transfers it to the Snowflake data warehouse.



AWS Lambda

lambda_function × Execution results ×

```python
import os
import requests
import snowflake.connector as sf


def lambda_handler(event, context):

    url = 'https://de-materials-tpcds.s3.ca-central-1.amazonaws.com/inventory.csv'
    destination_folder = '/tmp'
    file_name = 'inventory.csv'
    local_file_path = '/tmp/inventory.csv'

    # Snowflake connection parameters
    account = 'TCNREJL-IKB46695'
    warehouse = 'COMPUTE_WH'
    database = 'tpcds'
    schema = 'raw'
    table = 'inventory'
    user = 'TPCDS_user'
    password = '/
    role='accountadmin'
    stage_name = 'inv_Stage'

    # Download the data from the API endpoint
    response = requests.get(url)
    response.raise_for_status()
```

File Edit Find View Go Tools Window Test Deploy

Go to Any

lambda_function. × Execution results ×

Environment

lam

▾ Execution results

Test Event Name

test

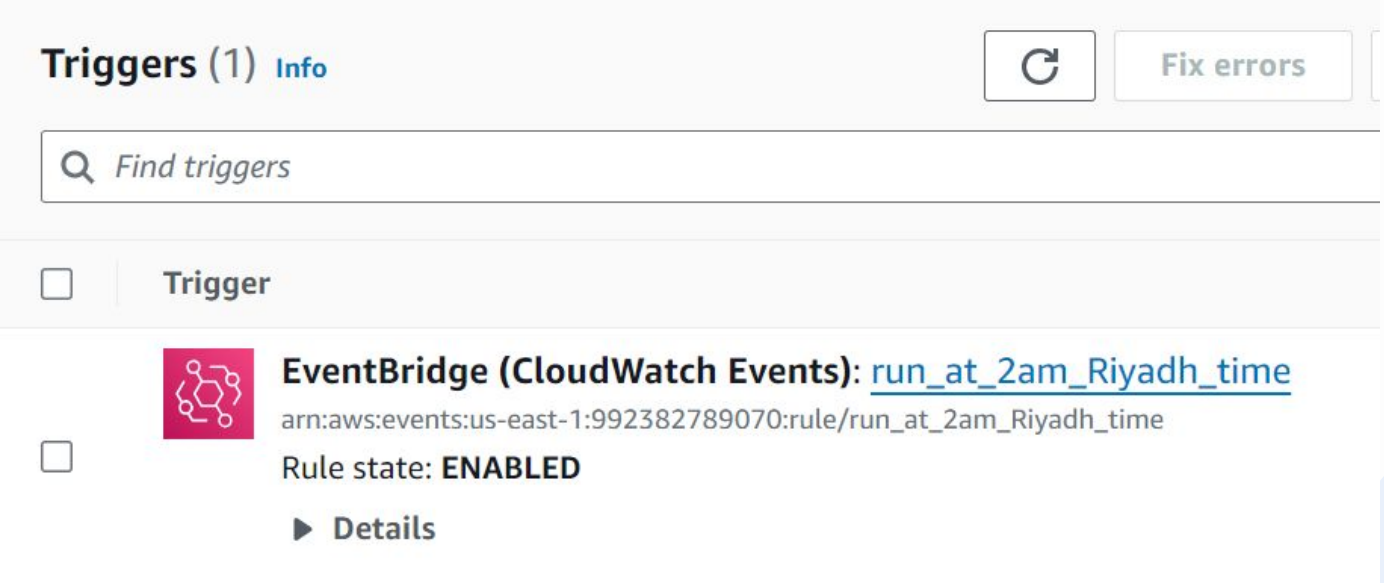
Response


```
{
  "statusCode": 200,
  "body": "File downloaded and uploaded to Snowflake successfully."
}
```


Function Logs

```
4
2452016,17612,5,74
2452016,17614,5,635
2452016,17617,5,648
2452016,17618,5,221
2452016,17620,5,157
2452016,17623,5,895
2452016,17624,5,380
2452016,17626,5,88
2452016,17629,5,521
2452016,17630,5,263
2452016,17632,5,558
2452016,17635,5,719
2452016,17636,5,768
2452016,17638,5,946
2452016,17641,5,722
2452016,17642,5,319
```

EventBridge (CloudWatch Events)



EventBridge (CloudWatch Events)

Log streams (8)

Delete | Create log stream | Search all log streams

Filter log streams or try prefix search

Exact match | Show expired | Info

1

Log stream	Last event time
2024/04/17/[$LATEST]1ea808d37f3b47b2987d755140e87278	2024-04-18 02:01:55 (UTC+03:00)
2024/04/16/[$LATEST]3ae672f3cc75422b993eea98941ec8aa	2024-04-17 02:01:54 (UTC+03:00)
2024/04/16/[$LATEST]47148c04aa2d4a5b943e1f11faeedfa0	2024-04-16 22:25:48 (UTC+03:00)

2024-04-18T02:00:36.162+03:00	2452016,17995,5,906
2024-04-18T02:00:36.162+03:00	2452016,17996,5,810
2024-04-18T02:00:36.162+03:00	2452016,17998,5,3
2024-04-18T02:01:55.234+03:00	File uploaded to Snowflake successfully.
2024-04-18T02:01:55.259+03:00	END RequestId: 8e17cfc8-0d9c-43d2-bcb7-8590420d8a74
2024-04-18T02:01:55.259+03:00	REPORT RequestId: 8e17cfc8-0d9c-43d2-bcb7-8590420d8a74 Duration: 87499.92 ms Billed Duration: 87500 ms Memory Size: 1000 MB …

Data Ingestion from Postgres Database

- Transfer data from the Postgres database on AWS RDS to the Snowflake data warehouse.
- Postgres Database: Hosted on AWS RDS.

All tables except the Inventory table, containing data about sales, customers, and other dimensions.

- Data Transfer Method: Airbyte
  Establishes a connection to the raw schema of the Postgres database.
  Transfers all tables from the Postgres database to the Snowflake data warehouse.

Airbyte

Connections

+ New connection

NAME	SOURCE NAME	DESTINATION NAME	FREQUENCY	LAST SYNC	ENABLED
DE-RDS → Snowflake	Postgres - DE-RDS	Snowflake - Snowflake	Cron	17 minutes ago	

Snowflake

Databases Worksheets

Pinned (0)

No pinned objects

Search objects

- RAW
  - Tables
    - CALL_CENTER
    - CATALOG_PAGE
    - CATALOG_SALES
    - CUSTOMER
    - CUSTOMER_ADDRESS
    - CUSTOMER_DEMOGRAPHI...
    - DATE_DIM
    - HOUSEHOLD_DEMOGRAP...
    - INCOME_BAND

Databases Worksheets

Pinned (0)

No pinned objects

Search objects

- ITEM
- PROMOTION
- SHIP_MODE
- TIME_DIM
- WAREHOUSE
- WEB_PAGE
- WEB_SALES
- WEB_SITE
- _AIRBYTE_RAW_CALL_CEN...
- _AIRBYTE_RAW_CATALOG_...
- _AIRBYTE_RAW_CATALOG_...
- _AIRBYTE_RAW_CUSTOMER

03

Data Transformation

Data Transformation

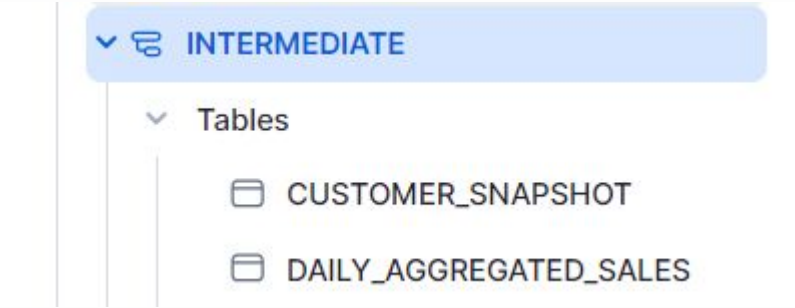
This involves reshaping tables from their original structure to the desired format. Throughout this phase, tasks include:

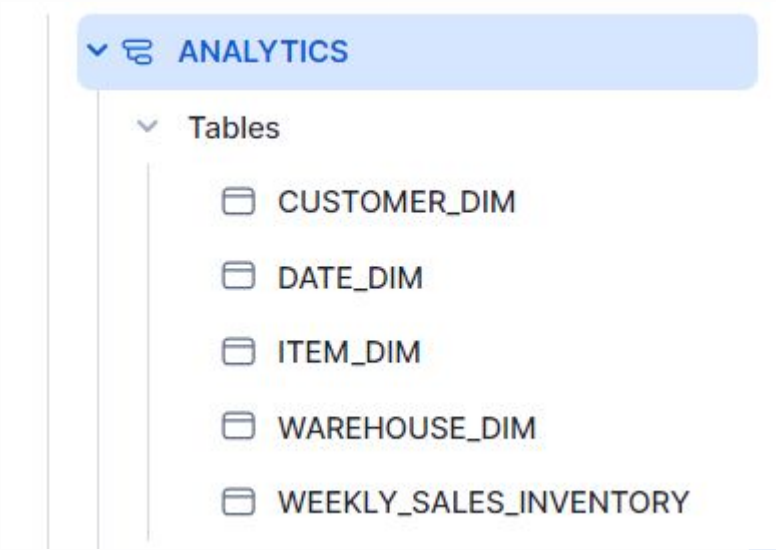
- Creating a data model
- Developing ETL scripts
- Establishing a schedule for the data loading process

Business Requirements

- sum_qty_wk: The sum of sales_quantity for this week.
- sum_amt_wk: The sum of sales_amount for this week.
- sum_profit_wk: The sum of net_profit for this week.
- avg_qty_dy: The average daily sales_quantity for this week (= sum_qty_wk/7).
- inv_on_hand_qty_wk: The item's inventory on hand at the end of each week in all warehouses (= The inventory on hand at the end of this week).
- wks_sply: Weeks of supply, an estimated metric to see how many weeks the inventory can supply the sales (inv_on_hand_qty_wk/sum_qty_wk).
- low_stock_flg_wk: Low stock weekly flag. For example, if there is a single day where (avg_qty_dy > 0 && (avg_qty_dy > inventory_on_hand_qty_wk)) in the week, then mark this week's flag as True.or the data loading process

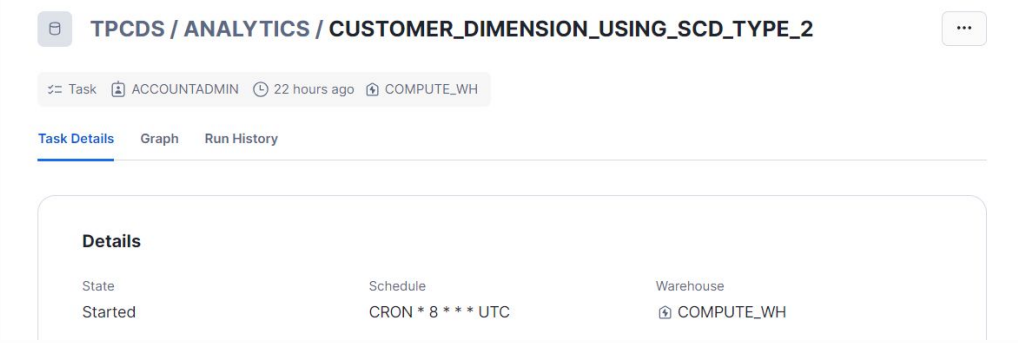
Data Model and ETL

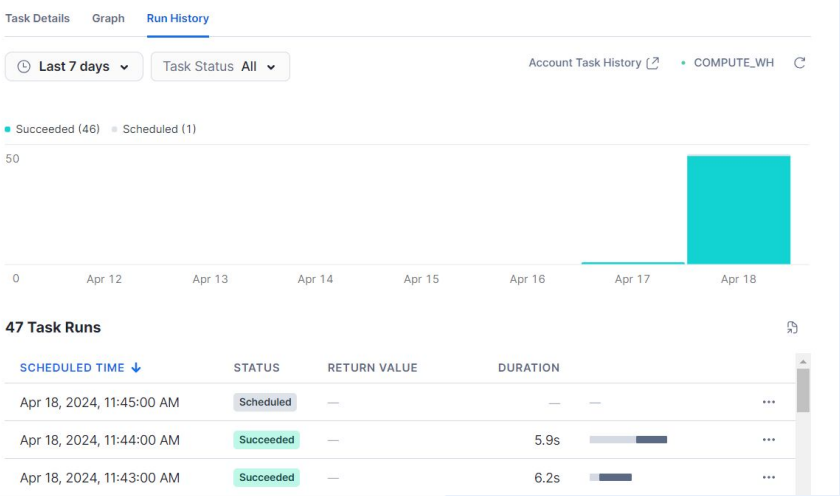




Scheduling

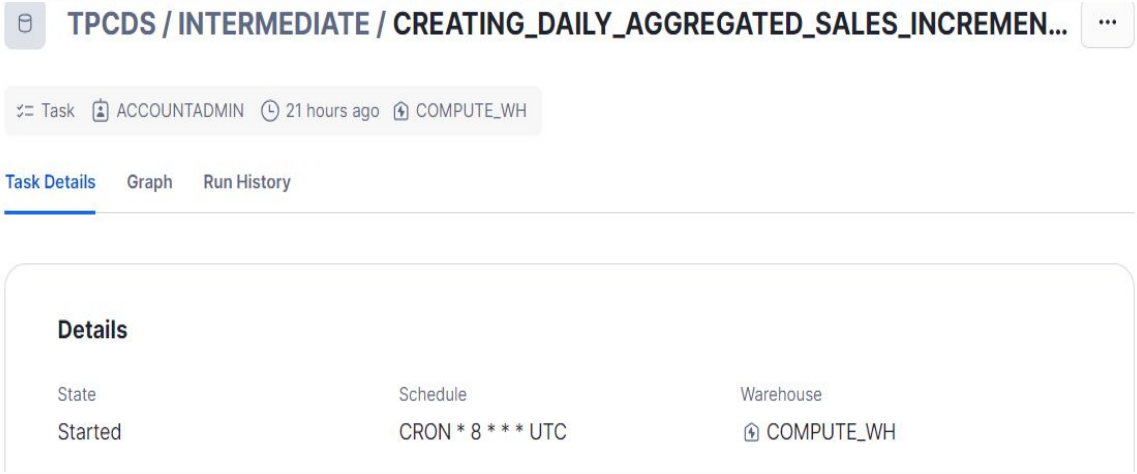
- customer_dimension
  The task is scheduled to run every day at 8:00 UTC using a cron expression.

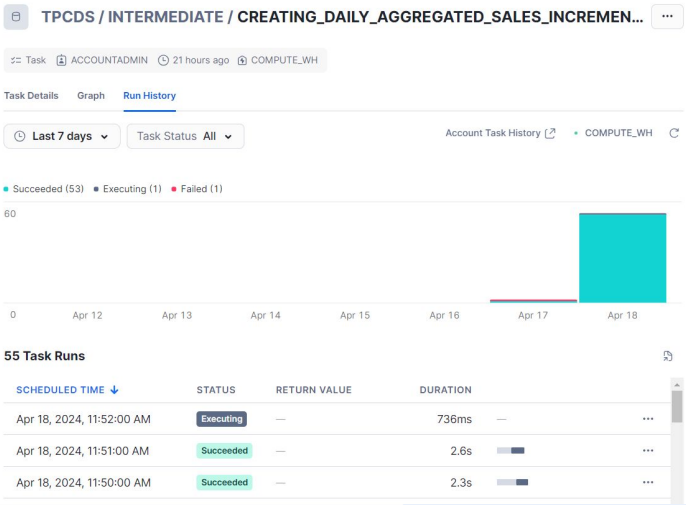




Scheduling

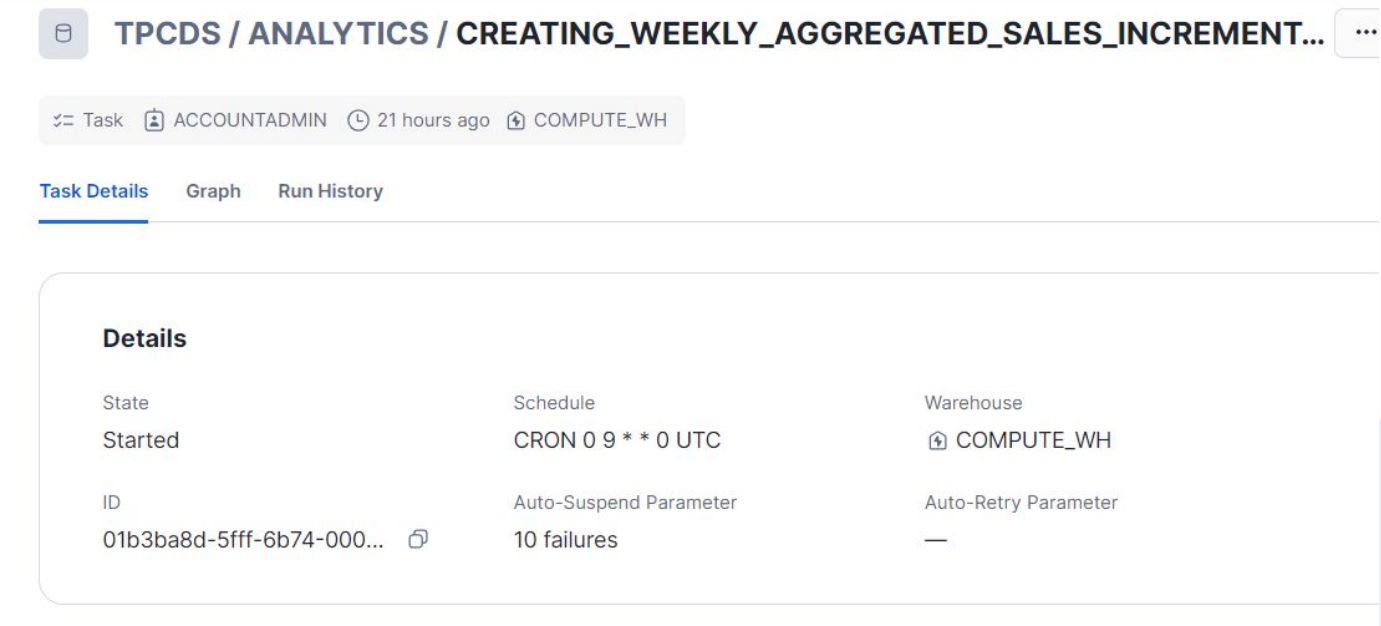
- DAILY_AGGREGATED_SALES





Scheduling

- WEEKLY_SALES_INVENTORY
  The task is scheduled to run every Sunday at 9:00 UTC using a cron expression.



04

Data Visualization

Data Visualization

Metabase

BI (Business Intelligence) tool that allows users to explore and analyze data from Snowflake with advantages such as:

- Easy Data Exploration: Emphasize how users can drag-and-drop to create visualizations like charts and graphs without writing code.

- Collaborative Features: sharing dashboards and collaborating with colleagues on data analysis.

- Customization: customize dashboards and write custom SQL queries

Data Visualization

TPCDS

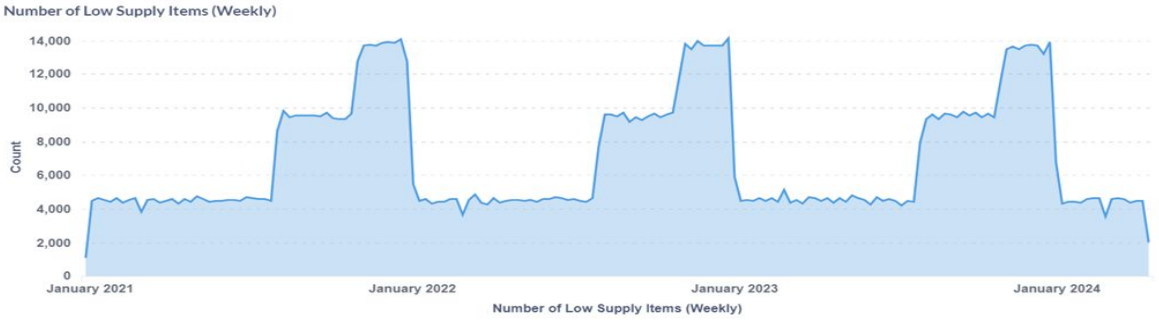
Number Of Items: 2

Warehouse: 1

Weekly Inventory Shortages

Warehouse Sk	Sold Wk Sk → Cal Dt: Week	Low Stock Flg Wk	Item Sk
1	December 27, 2020 – January 2, 2021	true	56
1	December 27, 2020 – January 2, 2021	true	262
1	December 27, 2020 – January 2, 2021	true	272
1	December 27, 2020 – January 2, 2021	true	442
1	December 27, 2020 – January 2, 2021	true	478
1	December 27, 2020 – January 2, 2021	true	512

Rows 1-6 of first 2000



Data Visualization

Top selling items

CAL_WK	ITEM_SK	SUM_QTY_WK	SUM_AMT_WK	RANKING
November 5, 2023	3,865	243	24,601.2	1
November 5, 2023	8,947	98	23,901.22	2
April 30, 2023	1,561	97	18,946.04	1
April 30, 2023	17,513	99	18,785.25	2
August 7, 2022	17,692	98	26,733.42	1

Rows 1-5 of 346

Bottom selling items

CAL_WK	ITEM_SK	SUM_QTY_WK	SUM_AMT_WK	RANKING
November 5, 2023	1,907	3	0	1
November 5, 2023	9,355	5	0	2
April 30, 2023	9,542	7	0	1
April 30, 2023	9,380	12	0	2
August 7, 2022	17,506	8	0	1

Rows 1-5 of 363

For Future

Improving the dashboard by providing more information about the data, therefore; it can indeed enhance both business requirements and data modeling.

Thanks!