Problem Statement

Polo Ralph Lauren receives sales and return data from multiple channels — **POS**, **E-commerce**, and **Third-Party partners** — in separate raw CSV files.
These files:

- Arrive in **different currencies** and are stored in Amazon S3
- Have to be **reconciled** to produce a unified view of sales
- Need to be **kept updated automatically** without manual loading

The business requires:

- A **single fact table** combining all channels
- **Amounts standardized to USD** for consistent reporting
- **Refund transactions identified** for tracking returns
- **Automated daily updates** when new files arrive in S3

Solution Implemented (AWS + Snowflake ELT)

1️⃣ Data Storage & Ingestion

- Created an **S3 bucket** `omni-recon-data` with folders:

```
raw/pos/
raw/ecommerce/
raw/thirdparty/
raw/reference/
```

- Uploaded channel-specific CSV files + FX rate file to their respective folders.
- Created a **Snowflake storage integration** with AWS IAM role & external stage `@omni_stage` pointing to the S3 bucket.

2️⃣ Automatic Loading (Snowpipe)

- Created **4 Snowpipes**:
  - `pos_pipe` → loads from `raw/pos/` → `stage_pos_sales`
  - `ecommerce_pipe` → loads from `raw/ecommerce/` → `stage_ecommerce_sales`
  - `thirdparty_pipe` → loads from `raw/thirdparty/` → `stage_thirdparty_sales`
  - `fx_rates_pipe` → loads from `raw/reference/` → `stage_fx_rates`
- Configured them to **auto-ingest** new files from S3 into staging tables.

3️⃣ Staging Table Structures

All sales staging tables (`POS` , `E-commerce` , `Third-Party`):

```
transaction_id STRING
channel STRING
store_id STRING
product_id STRING
currency STRING
amount_local FLOAT
refund_flag BOOLEAN
quantity INT
datetime_local TIMESTAMP
```

FX rates staging table:

```
date DATE
currency STRING
exchange_rate FLOAT
```

4️⃣ Transformations Implemented

The transformation logic inside Snowflake creates the `sales_reconciliation_fact` table by:

1. **Merging all sales channels**

   ```
   SELECT * FROM stage_pos_sales UNION ALL SELECT * FROM stage_ecommerce_sales
   UNION ALL SELECT * FROM stage_thirdparty_sales
   ```

2. **Joining with FX rates** to standardize amounts in USD

   ```
   ROUND(s.amount_local / f.exchange_rate, 2) AS amount_usd
   ```

3. **Retaining refund_flag** to identify returns

   ```
   s.refund_flag
   ```

4. **Extracting date dimensions** for reporting

   ```
   YEAR(s.datetime_local) AS year, MONTH(s.datetime_local) AS month
   ```

5. **Deduplicating on load** (incremental)

   ```
   WHERE NOT EXISTS ( SELECT 1 FROM sales_reconciliation_fact fact WHERE
   fact.transaction_id = s.transaction_id AND fact.channel = s.channel )
   ```

5️⃣ Automation (Snowflake Task)

- Created a **daily scheduled Snowflake Task** `refresh_sales_fact`:
  - Runs at **2 AM UTC**.
  - Inserts only **new transactions** from staging tables into `sales_reconciliation_fact`.
  - Leaves existing rows untouched → **incremental loads**, not full rebuilds.

6️⃣ Current Workflow

1. **New file lands in S3** → Snowpipe detects and loads into staging table.
2. **Daily at 2 AM UTC** → Task runs transformation SQL → Inserts only new rows into fact table.
3. `sales_reconciliation_fact` always contains **all channels, all currencies converted to USD, refunds tracked, with Year/Month fields** for easy reporting.