**Data Analyst Task**

Data Sources: Customer.xls, Orders.csv, Shipping.json

Objectives:

**Step 1: Process to check accuracy, completeness and reliability of the data sources**

import pandas as pd

import openpyxl as op

df\_shipping = pd.read\_json(‘DataAnalystTask/Shipping.json')

df\_order = pd.read\_csv('DataAnalystTask/Order.csv')

df\_customer = pd.read\_excel('DataAnalystTask/Customer.xls', sheet\_name='atkoe-u250m')

# --- Step 1: Completeness ---

# Check NULL values in any of the fields

print('Order missing values:\n', df\_order.isnull().sum())

print('Customer missing values:\n', df\_customer.isnull().sum())

print('Shipping missing values:\n', df\_shipping.isnull().sum())

#--- Step 2: Accuracy ---

# Amounts should be positive

print('Orders with non-positive amount: ', df\_order[df\_order['Amount'] <= 0].count())

# Check foreign key consistency

print('Orders with missing customer info: ', df\_order[~df\_order['Customer\_ID'].isin(df\_customer['Customer\_ID'])].count())

print('Shipping with missing customer info: ', df\_shipping[~df\_shipping['Customer\_ID'].isin(df\_customer['Customer\_ID'])].count())

#--- Step 3: Reliability ---

# Check for duplicate records

print('Orders with duplicate values: ',df\_order.duplicated().sum())

print('Customer with duplicate values: ',df\_customer.duplicated().sum())

print('Shipping with duplicate values: ',df\_shipping.duplicated().sum())

**Step 2: Create Data Model**

A simple star schema is created to relate the tables.

* DimCustomer has 1🡪Many relationships to FactOrder as 1 customer can place multiple orders.
* DimCustomer has 1🡪Many relationships to FactShipping as 1 customer can have multiple orders under shipment based on the orders placed.

A diagram of a data flow

Description automatically generated with medium confidence

**Step 3: Technical story to implemented by a Data & QA Engineer**

*Data Engineer User Story:*

As a Data Engineer, I want to ingest customer data from the source file Customer.xls into DimCustomer table so it is available for downstream cleansing and reporting process.

Target DDL:

CREATE TABLE DimCustomer (

customer\_id VARCHAR (50),

first\_name VARCHAR (100),

last\_name VARCHAR (100),

age INT,

country VARCHAR (100),

created\_date DATETIME DEFAULT GETDATE ()

)

Transformation logic:

* Direct pull from source to target.
* Rename source columns first as first\_name and last as last\_name before mapping
* Add housekeeping column – created\_date

*QA Engineer User Story:*

As a QA Engineer, I want to validate the completeness, accuracy and reliability of the data loaded into target system.

Acceptance Criteria:

* Row counts must match between source and target.
* Check for NULLs in mandatory fields - customer\_id
* Check for duplicate rows
* Verify transformation rules

**Business Reporting Requirements**

1. *The total amount spent and the country for the Pending delivery status for each country.*

SELECT country, SUM(Amount) AS total\_amount\_spent

FROM fact\_order o

LEFT JOIN dim\_customer c ON o.Customer\_ID = c.Customer\_ID

LEFT JOIN fact\_shipping s ON o.Customer\_ID = s.Customer\_ID

WHERE Status = 'Pending'

GROUP BY Country;

1. *The total number of transactions, total quantity sold, and total amount spent for each customer, along with the product details.*

WITH customer\_item\_agg AS (

SELECT

customer\_id,

item,

COUNT(\*) AS total\_transactions,

COUNT(\*) AS total\_quantity\_sold,

SUM(amount) AS total\_amount\_spent

FROM fact\_order

GROUP BY Customer\_ID, Item

)

SELECT

c.customer\_id,

c.First AS first\_name,

c.Last AS last\_name,

cia.item,

cia.total\_transactions,

cia.total\_quantity\_sold,

cia.total\_amount\_spent

FROM customer\_item\_agg cia

JOIN dim\_customer c ON c.Customer\_ID = cia.Customer\_ID

ORDER BY c.Customer\_ID, cia.Item;

1. *The maximum product purchased for each country.*

SELECT country, item

FROM (

SELECT c.country, o.item, COUNT(o.customer\_id) AS total\_quantity,

DENSE\_RANK() OVER (PARTITION BY c.country ORDER BY COUNT(o.customer\_id) DESC) AS rn

FROM fact\_order o

JOIN dim\_customer c ON o.customer\_id = c.customer\_id

GROUP BY c.country, o.item

) a WHERE rn = 1;

1. *The most purchased product based on the age category less than 30 and above 30.*

WITH customer\_age AS (

SELECT

c.customer\_id,

CASE WHEN c.age < 30 THEN '<30' ELSE '>=30' END AS age\_category,

o.item

FROM fact\_order o

JOIN dim\_customer c ON o.customer\_id = c.customer\_id

),

product\_quantity AS (

SELECT

age\_category,

item,

COUNT(\*) AS total\_quantity\_sold

FROM customer\_age

GROUP BY age\_category, item

),

ranked\_products AS (

SELECT

\*,

ROW\_NUMBER() OVER (PARTITION BY age\_category ORDER BY total\_quantity\_sold DESC) AS rn

FROM product\_quantity

)

SELECT age\_category, item, total\_quantity\_sold

FROM ranked\_products

WHERE rn = 1;

1. *The country that had minimum transactions and sales amount*

SELECT country, total\_quantity

FROM (

SELECT c.country, o.item, SUM(o.amount) AS total\_quantity,

ROW\_NUMBER() OVER (PARTITION BY c.country ORDER BY SUM(o.amount)) AS rn

FROM fact\_order o

JOIN dim\_customer c ON o.customer\_id = c.customer\_id

GROUP BY c.country, o.item

) a WHERE rn = 1;