|  |
| --- |
| **Shipping** |
| Shipping\_ID |
| Status |
| Customer\_ID |

Verify the accuracy, completeness, and reliability of source data.

**Datasets**

|  |
| --- |
| **Customer** |
| Customer\_ID |
| First |
| Last |
| Age |
| Country |

|  |
| --- |
| **Order** |
| Order\_ID |
| Item |
| Amount |
| Customer\_ID |

The datasets are in different formats, hence loaded all to a python dataframe to perform initial analysis before loading it to SQL table

**Analysis on the datasets:**

* **Customers**
  + Data seems accurate, **no duplicate** records found

SELECT

Customer\_id

,First

,Last

,count(\*)

FROM [dbo].[PEI\_Customer]

group by Customer\_id,First,Last

having count(\*) > 1

* No customer is in 2 different Country
* Customer data seems **OK** to proceed
* **Order**
  + No Duplicates at Order id level
  + Same Items are purchased by Customers in different orders

SELECT [Item]

,[Customer\_ID]

,COUNT(1)

FROM [DataAnalytics].[dbo].[PEI\_Order]

GROUP BY [Item],[Customer\_ID]

HAVING COUNT(1)>1

* + **Date column is missing**. Hence, **cannot** perform any time variant analysis.
* **Shipping**
  + **Order ID** missing in the shipping table to tag the shipping status of the order.
  + **Date column is missing**. Hence, cannot see the progress of the order status.
  + Data seems **incomplete** without the above 2 columns.

All Customer IDs in Shipping and Orders are **Valid** ids present in Customer table.

SELECT Customer\_ID FROM PEI\_Order

WHERE Customer\_ID NOT IN (SELECT Customer\_ID FROM PEI\_Customer)

UNION

SELECT Customer\_ID FROM PEI\_Shipping

WHERE Customer\_ID NOT IN (SELECT Customer\_ID FROM PEI\_Customer)

* 0 Rows returned

However, there are entries in Shipping for Customer\_id which **do not** have any records in order table and vice versa.

**For example:**

SELECT

C.Customer\_ID

,O.Order\_ID

,O.Item

,S.[Status]

FROM dbo.PEI\_Customer C

LEFT JOIN PEI\_Order O

ON C.Customer\_ID = O.Customer\_ID

LEFT JOIN dbo.PEI\_Shipping S

ON C.Customer\_ID = S.Customer\_ID

WHERE C.Customer\_ID = 185

OR C.Customer\_ID = 250

Based on your findings, define and outline the requirements for anticipated datasets, detailing the necessary data components.

**Requirements on existing datasets.**

**Customer**

Data is complete

Additional details like address can be added.

**Order**

**Order Date, Quantity** needs to be added.

**Shipping**

Shipping date needs to be mapped

Order id needs to be added with referential integrity to orders table.

Progress of the shipment can be recorded in the shipping table. With a “Shipment update date”

Anticipated Datasets:

Calendar

Region

Product

Develop the data models to effectively organise and structure the information and provide a detailed mapping of existing data flows, focussing on the areas of concern.

**Data flow**

File sources -> Python / SSIS -> SQL tables

Additional date dimension to be created using table valued functions in SQL to generate a series of dates and then use the DATE functions to create the rest of the columns.

Distinct values of Country from the customer table can be used to create a region table which can be used to join to orders.

Order table need more information like Order\_date, Quantity and Country details.

Shipping data needs mapping to order number to identify the status of the order. Order\_ID should be foreign key in the table to ensure referential integrity is maintained.

This table should be a SCD table to maintain the status of the order and update the date when the status changes.

**Proposed Data Model**

Region

Calendar

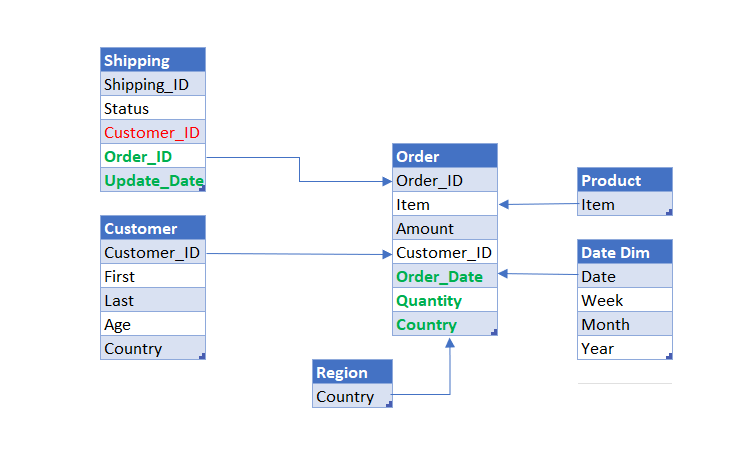
Customer

Shipping

Orders

Product

A Star schema with Orders fact table and Customer, Region, Product and a date Dimension tables. Shipping can be added as a dimension table of SCD Type 1 or Type 2 depending on the requirement.



Prepare a story with technical specifications for one part of the data model for a data engineer.

**Order Data Analysis**

**Overview**

In order to extract insights from the data set for Orders data with the provided datasets, it is observed that the data in missing column and key mappings. Additional datasets also required to be brought in to make more informed decisions.

**Goals and Technical Requirements**

**Tools:**

ETL (SSIS/Python)

SQL database

Visualisation (Powerbi/Tableau)

Datasets in multiple formats needs to be transformed and loaded to database. This can be done using any ETL tool or python.

Create a **Calendar** dimension to join all date columns and perform time intelligence analysis.

Use Table values function or looping to create the table

CREATE TABLE dbo.Calendar (

Date DATE NOT NULL,

DayOfWeek VARCHAR(10) NOT NULL,

DayOfMonth INT NOT NULL,

DayOfYear INT NOT NULL,

WeekOfYear INT NOT NULL,

MonthOfYear INT NOT NULL,

QuarterOfYear INT NOT NULL,

Year INT NOT NULL,

IsWeekend BIT NOT NULL

);

DECLARE @StartDate DATE = '2020-01-01';

DECLARE @EndDate DATE = '2025-12-31';

WHILE @StartDate <= @EndDate

BEGIN

INSERT INTO dbo.Calendar (

Date,

DayOfWeek,

DayOfMonth,

DayOfYear,

WeekOfYear,

MonthOfYear,

QuarterOfYear,

Year,

IsWeekend

)

VALUES (

@StartDate,

DATENAME(WEEKDAY, @StartDate),

DAY(@StartDate),

DATEPART(DAYOFYEAR, @StartDate),

DATEPART(WEEK, @StartDate),

MONTH(@StartDate),

DATEPART(QUARTER, @StartDate),

YEAR(@StartDate),

CASE WHEN DATENAME(WEEKDAY, @StartDate) IN ('Saturday', 'Sunday') THEN 1 ELSE 0 END

);

SET @StartDate = DATEADD(DAY, 1, @StartDate);

END;

Create **Product** and **Region** dimensions. If no separate dataset is available, create by Normalising the Orders table.

For the **Shipping** table, create a stored procedure to implement and Type 1 SCD table, which will update the date as when the status is updated.

To achieve this create a staging table for Shipping when a new record for shipping is inserted to staging, its will be checked with the existing record in the Shipping table. If a match is found on the order\_id the new status is updated on the table and the updated data is also changed.

**Order\_id** in the Shipping table should be the foreign key to make sure that only valid orders are getting entered to the Shipping table.

In the **Order** table every single product is considered as 1 order. Also see there are customers purchasing the same product multiple times. Ther is no option to identify if this is the same order or a different one.

Adding and order date along with the Quantity of item purchased will be helpful to achieve this.

**Product** should be a foreign key.

Along with that, adding a foreign key for **Region** also helps is more flexibility for detailed analysis.

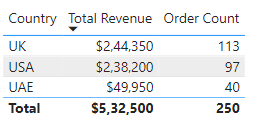
Quantity should be of **integer** data type.

Communicate the findings and insights to stakeholders in a visually comprehensive manner.

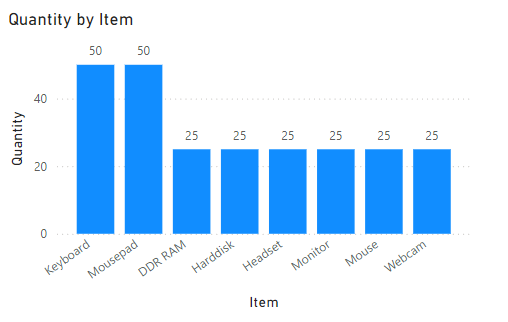
Findings:

From the Orders we data it is noted that there are a total Revenue of **$5,32,500** from **250** orders.

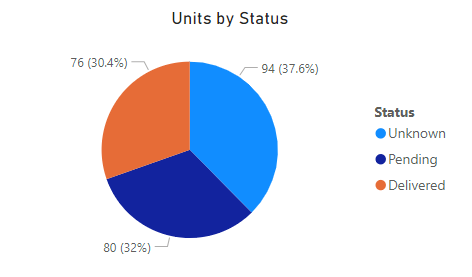
The UK contribute the most to the revenue, followed by USA and UAE.



It is observed that, every order has only 1 product associated to it which says the total units sold is also 250. The split of Product and units sold is as follows



We notice that the shipping status for many orders are unavailable. And also, there are Shipment which does not tagged to any order. Around **37.6%** of all order fall into this category.



To Achieve this, we have assumed that a customer can have only one status at a time, giving priority to delivered orders. This assumption is being done since there is no direct link between orders and shipping using order id.

Out of the all the 250 customers, majority fall above the age group of above 30 yrs (69.2%)

