Analytics Assignment

24th June'2025

Debarati Sarkar

Accuracy, Completeness, and Reliability of Source Data:

Accuracy at Table Level:

Customers Table:

1. Data is unique at a customer id level. So, customer id is the primary key.
2. No duplicate customer id found. No Null customer id found. Primary key rules maintained.
3. All other fields are in expected format.

Shipping Table:

1. Data is unique at Shipping Id level. So, shipping id is the primary key.
2. No duplicate shipping id found. No Null shipping id found. Primary key rules maintained.
3. One customer can have multiple shipping IDs. Hence there are multiple customer Ids for shipping ids. This can be a valid scenario.

Orders Table:

1. Data is unique at Order Id level. So, Order id is the primary key.
2. No duplicate Order id found. No Null order id found. Primary key rules maintained.
3. One customer can have multiple Order IDs. Hence there are multiple customer Ids for order ids. This can be a valid scenario.
4. All other fields are correct format.

Accuracy, Completeness, and Reliability of Source Data:

Completeness and Reliability:

Customers Table:
1. No changes needed in the Customers Table. Fields and Data is complete and reliable.

Shipping Table:
1. No changes needed in the Shipping Table. Fields and Data is complete and reliable.

Orders Table:

1. Few fields needs to be added in this table to achieve the asks from the BRD shared.

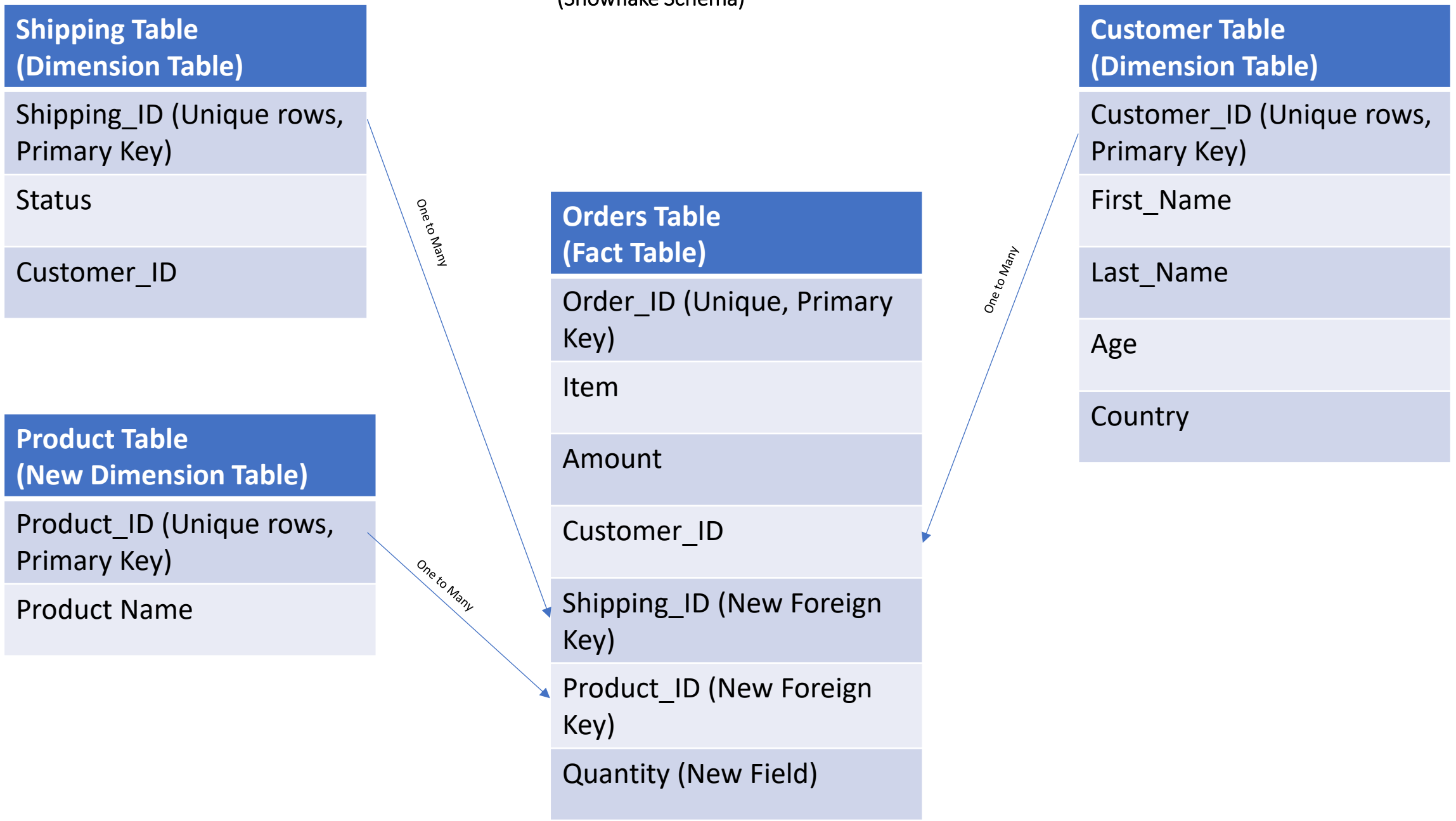
2. As per the BRD, to achieve the result for Point Number 1, we need to join the Orders table and Shipping Table. So, Shipping ID column should be present in the Orders Table which is missing currently.

3. As per the BRD, to achieve the result for Point Number 2, we need to have a Products Table as a separate dimension table.
This dimension table needs to be joined with the Orders table on Product ID. So, a Product ID column is needed in the Orders table as well. Lineage diagram has been shown in the next slide.

4. Also to achieve result for Point Number 2 (total quantity sold), we need a new field 'Quantity' in the Orders table.

Product Table: A New Product Table needs to be created

Entity Relationship Diagram (One to Many Relationship)
(Snowflake Schema)



Technical Requirement For Data Engineering Team

Objective: Track Key Sales metrics at Customer, Country and Product Level

Source Tables: Source Tables and Relevant Columns

Orders Table (FACT)	Customer Table(DIMENSION)	Shipping Table(DIMENSION)	Product Table(DIMENSION)
Order_ID INT Item NVARCHAR(50) Amount INT Customer_ID INT Product_ID INT Shipping_ID INT Quantity INT	Customer_ID INT First_Name NVARCHAR(50) Last_Name NVARCHAR(50) Age INT Country NVARCHAR(50)	Shipping_ID INT Status NVARCHAR(50) Customer_ID INT	Product_ID INT Product_Name NVARCHAR(50)
Order_ID is the PRIMARY_KEY Customer_ID, Product_ID and Shippind_ID are Foreign Keys to join with Primary Keys from Dimension Tables	Customer_ID is the PRIMARY_KEY	Shipping_ID is the PRIMARY_KEY	Product_ID is the PRIMARY_KEY

Technical Requirements:

1. Data Type has been documented for each field.
2. Also, Primary and Foreign Keys have been specified. No null values and duplicate values should be present for Primary Keys.

Technical Requirement For Data Engineering Team

Validation Rules:

1. DataType and Metadata of the tables should be maintained as suggested in previous slide.
2. No null values and duplicate values should be present for Primary Keys.
3. Alert should be triggered in case any changes in metadata

Target Table Location:

1. Target should be in analytics catalog and test database and schema.
2. Data should be loaded as tables in SQL server database.

Refresh Timing:

Refresh should happen daily.

```sql
---Accuracy Checks--
select count(customer_id),count(distinct customer_id) from dbo.Customer
select * from dbo.Customer where customer_id is null
---
select count(Order_ID),count(distinct Order_ID) from dbo.Orders
select * from dbo.Orders where order_id is null
select count(customer_id),count(distinct customer_id) from dbo.Orders
---
select count(shipping_id),count(distinct shipping_id)  from dbo.Shipping
select count(customer_id),count(distinct customer_id)  from dbo.Shipping
select * from dbo.Shipping where shipping_id is null
```

SQL Queries

```sql
---the total amount spent and the country for the Pending delivery status for each
country.

SELECT
    c.country,
    SUM(o.amount) AS total_amount_spent
FROM dbo.Orders o
JOIN dbo.Customer c ON o.customer_id = c.customer_id
JOIN dbo.Shipping s ON o.shipping_id = s.shipping_id
WHERE s.status = 'Pending'
GROUP BY c.country;
```

```sql
-----the total number of transactions for each customer, along with the product details.

SELECT
    c.customer_id,
    p.product_id,
    p.product_name,
    COUNT(o.order_id) AS total_transactions
FROM dbo.Orders o
JOIN dbo.Customer c ON o.customer_id = c.customer_id
JOIN dbo.Products p ON o.product_id = p.product_id
GROUP BY c.customer_id, p.product_id, p.product_name;
```

SQL Queries

```sql
-----the total quantity sold for each customer, along with the product details.

SELECT
    c.customer_id,
    p.product_id,
    p.product_name,
    SUM(o.quantity) AS total_quantity_sold
FROM dbo.Orders o
JOIN dbo.Customer c ON o.customer_id = c.customer_id
JOIN Products p ON o.product_id = p.product_id
GROUP BY c.customer_id, p.product_id, p.product_name;
```

```sql
---------the total amount spent for each customer, along with the product details.

SELECT
    c.customer_id,
    p.product_id,
    p.product_name,
    SUM(o.amount) AS total_amount_sold
FROM dbo.Orders o
JOIN dbo.Customer c ON o.customer_id = c.customer_id
JOIN Products p ON o.product_id = p.product_id
GROUP BY c.customer_id, p.product_id, p.product_name;
```

SQL Queries

```sql
----- the maximum product purchased for each country.

SELECT
    country,
    product_id,
    product_name,
    total_purchases
FROM (
    SELECT
        c.country,
        p.product_id,
        p.product_name,
        COUNT(o.order_id) AS total_purchases,
        RANK() OVER (PARTITION BY c.country ORDER BY COUNT(o.order_id) DESC) AS rank_per_country
    FROM dbo.Orders o
    JOIN dbo.Customer c ON o.customer_id = c.customer_id
    JOIN Products p ON o.product_id = p.product_id
    GROUP BY c.country, p.product_id, p.product_name
) ranked_products
WHERE rank_per_country = 1;
```

```sql
-----the most purchased product based on the age category less than 30 and above 30.
SELECT
    age_category,
    product_id,
    product_name,
    total_purchases
FROM (
    SELECT
        CASE
            WHEN c.age < 30 THEN 'Under 30'
            ELSE '30 and above'
        END AS age_category,
        p.product_id,
        p.product_name,
        COUNT(o.order_id) AS total_purchases,
        RANK() OVER (
            PARTITION BY
                CASE WHEN c.age < 30 THEN 'Under 30' ELSE '30 and above' END
            ORDER BY COUNT(o.order_id) DESC
        ) AS rank_within_age_group
    FROM Orders o
    JOIN Customers c ON o.customer_id = c.customer_id
    JOIN Products p ON o.product_id = p.product_id
    GROUP BY
        CASE WHEN c.age < 30 THEN 'Under 30' ELSE '30 and above' END,
        p.product_id,
        p.product_name
) ranked_products
WHERE rank_within_age_group = 1;
```

SQL Queries

```sql
-------the country that had minimum transactions and sales amount.

SELECT country, num_transactions, total_sales
FROM (
    SELECT
        c.country,
        COUNT(o.order_id) AS num_transactions,
        SUM(o.amount) AS total_sales,
        RANK() OVER (ORDER BY COUNT(o.order_id), SUM(o.amount)) AS rnk
    FROM dbo.Orders o
    JOIN dbo.Customer c ON o.customer_id = c.customer_id
    GROUP BY c.country
) ranked
WHERE rnk = 1;
```

THANK YOU