**describe customer;**

5 columns with NULL and DEFAULT constraint enabled.

**SELECT min(age) AS min\_age, max(age) AS max\_age, avg(age) AS avg\_age FROM customer;**

min\_age : 18

max\_age : 80

avg\_age : 47.5760

**# checking if customer\_id has duplicates or is null**

**select customer\_id from customer**

**group by customer\_id**

**having count(customer\_id) >1 or customer\_id is null;**

unique values with no nulls

select \* from customer

where age is null

or first is null

or last is null

or country is null

**# Count the number of customers from each country**

SELECT country, COUNT(\*) AS customer\_count FROM Customer GROUP BY country;

**Describe orders**

4 columns with NULL and DEFAULT constraint enabled.

**Checking for duplicates**

select order\_id from orders

group by order\_id

having count(order\_id) >1 or order\_id is null;

SELECT customer\_id, COUNT(\*) AS order\_count FROM Orders GROUP BY customer\_id;

**Checking Nulls**

select \* from orders

where Order\_ID is null

or item is null

no null item and order\_id

SELECT status, COUNT(\*) AS status\_count FROM Shipping GROUP BY status;

SELECT customer\_id, COUNT(\*) AS shipment\_count FROM Shipping GROUP BY customer\_id;

SELECT \* FROM Shipping WHERE status = 'Pending';

**Checking accuracy of data:**

select count(c.customer\_id) from customer c

left join orders o on

c.customer\_id =o.customer\_id

left join shipping s on

c.customer\_id =s.customer\_id

where s.customer\_id is null

and order\_id is not null

**#94 customers have orders but shipping details are missing.**

select \* from customer c

left join orders o on

c.customer\_id =o.customer\_id

left join shipping s on

c.customer\_id =s.customer\_id

where s.customer\_id is null

and order\_id is null

**#35 customers don't have orders**

select status,count(shipping\_id) as cnt\_of\_orders from shipping

group by status

**pending: 150**

**Delivered : 100**

purchase cost

select avg(amount) as avg\_order\_amt,min(amount) AS lowest\_order\_amt,max(amount) AS maximum\_order\_amt from orders

**avg\_order\_amt : 2130.0000**

**lowest\_order\_amt : 200**

**maximum\_order\_amt : 12000**

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

1. Shipping table has missing data, depending on order id.

2. Order id should be the foreign key of the shipping table.

3. No way to calculate the number of items sold per order. (create new product table with product price) or order table should have quantity column

- **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.**

We should have a snowflake schema for the data model, rather than the star schema which is being used now.

**Areas of concern** :

* Product table is missing.
* In orders table,
* **order status** column should be there : if it is pending, cancelled, fulfilled, returned or delivered.
* Order date column should be there.
* In shipping table, order\_id column should be there to know what order is shipped out of multiple orders placed by customer.

**Prepare a story with technical specifications for one part of the data model for a data engineer to include technical specifications and transformations. This story should give enough information for a Data Engineer to build table(s) and for a QA engineer to test it.**

**1. Business Context & Goal**

This story focuses on establishing the foundational data model for customer order fulfillment. The goal is to provide a unified and consistent view of customer information, their associated orders, and the shipping details for those orders. This data will be used for analytical reporting, customer segmentation, order tracking, and future data science initiatives.

**2. High-Level Requirements**

Ingest raw customer, order, and shipping data from the source systems.

Cleanse and standardize the data.

Establish clear relationships between customer, order, and shipping entities.

Ensure data quality and integrity.

Provide a robust and scalable data model for downstream consumption.

**3.1. Source Systems**

Customer Data: CRM\_DB.Customers (PostgreSQL) - Daily full load from CRM.

Order Data: ECOM\_DB.Orders (MySQL) - Daily incremental load based on last\_updated\_timestamp.

Shipping Data: SHIPPING\_API.Shipments (REST API) - Daily batch pull.

**3.2. Target Database & Schema**

Database: Analytics\_Warehouse (Snowflake)

Schema: DWH\_CORE

**3.3. Table Specifications & Transformations**

The following tables will be created in the DWH\_CORE schema.

**Table : Shipping**

This table will contain distinct shipping addresses associated with orders.

**Purpose**: To normalize and store unique shipping address details, linked to orders.

**Type:** Dimension Table

**Update Strategy:** Type 0 (Fixed) or Type 1 (Overwriting) based on unique address key. If an address changes, it's considered a new address.

**Grainularity:** One row per unique combination of address attributes.

