1. **DQ Checks on Source Data**

* Integrity Checks

Primary Key Uniqueness

-- Check for duplicate Customer\_ID

SELECT customer\_id, COUNT(\*)

FROM Customer

GROUP BY customer\_id

HAVING COUNT(\*) > 1;

-- Check for duplicate order\_id

SELECT order\_id, COUNT(\*)

FROM Orders

GROUP BY order\_id

HAVING COUNT(\*) > 1;

-- Check for duplicate shipping\_id

SELECT shipping\_id, COUNT(\*)

FROM Shipping

GROUP BY shipping\_id

HAVING COUNT(\*) > 1;

* Foreign Key Integrity/ Referential Integrity

--- Check if any Order belongs to Customers not present in Customers table

SELECT o.\*

FROM Orders o

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

WHERE c.Customer\_ID IS NULL;

--- Check if one Order\_ID belongs to different Customer\_ID

SELECT Order\_ID,COUNT(DISTINCT Customer\_ID) AS distinct\_customers

FROM Orders

GROUP BY Order\_ID

HAVING COUNT(DISTINCT Customer\_ID) > 1;

--- check if shipments belongs to non existent customers

SELECT s.\*

FROM Shipping s

LEFT JOIN Customer c ON s.Customer\_ID = c.Customer\_ID

WHERE c.Customer\_ID IS NULL;

* Completeness Checks

--- Completeness check for NULL or missing values

SELECT

  SUM(CASE WHEN First IS NULL OR TRIM(First) = '' THEN 1 ELSE 0 END) AS Missing\_First,

  SUM(CASE WHEN Last IS NULL OR TRIM(Last) = '' THEN 1 ELSE 0 END) AS Missing\_Last,

  SUM(CASE WHEN Age IS NULL THEN 1 ELSE 0 END) AS Missing\_Age,

  SUM(CASE WHEN Country IS NULL OR TRIM(Country) = '' THEN 1 ELSE 0 END) AS Missing\_Country

FROM Customer;

* Accuracy/Consistency Checks

-- Special characters in First/Last Name

SELECT Customer\_ID, First, Last

FROM Customer

WHERE First LIKE '%[^A-Za-z ]%' OR Last LIKE '%[^A-Za-z ]%';

* A screenshot of a computer

  AI-generated content may be incorrect.

--- Age limit

SELECT \*

FROM Customer

WHERE Age not between 18 and 100;

-- Unexpected values in Country

SELECT DISTINCT Country

FROM Customer

WHERE Country NOT IN ('USA', 'UK', 'UAE') or len(country)>3;

--- Check consistency in prices across products

select distinct item,amount

from Orders

order by 1;

A screenshot of a computer

AI-generated content may be incorrect.

--- Consistency Check on Status

SELECT \*

FROM Shipping

WHERE Status NOT IN ('Pending', 'Delivered');

* Business logic checks

---160/250 customers have placed orders

select distinct C.Customer\_ID from Customer C

left join Orders o on o.Customer\_ID=C.Customer\_ID

where O.Customer\_ID is not null

A screenshot of a computer program

AI-generated content may be incorrect.

-----61 Customers with orders but not shipping info

select count(DISTINCT C.Customer\_ID) from Customer C

left join Orders o on o.Customer\_ID=C.Customer\_ID

left join Shipping s on s.Customer\_ID=C.Customer\_id

where O.Customer\_ID is not  null and s.customer\_id is null

A screenshot of a computer

AI-generated content may be incorrect.

---------55 customers have shipments but no orders

select count(DISTINCT C.Customer\_ID) from Customer C

left join Orders o on o.Customer\_ID=C.Customer\_ID

left join Shipping s on s.Customer\_ID=C.Customer\_id

where O.Customer\_ID is null and s.customer\_id is not null

A screen shot of a computer program

AI-generated content may be incorrect.

---- duplicate order transactions with different shipping\_id

select O.Customer\_ID,O.Order\_ID,O.Item,count(\*) from Customer C

left join Orders o on o.Customer\_ID=C.Customer\_ID

left join Shipping s on s.Customer\_ID=C.Customer\_id

where O.Customer\_ID is not null and s.customer\_id is not null

group by O.Customer\_ID,O.Order\_ID,O.Item

having count(\*)>1

A screenshot of a computer program

AI-generated content may be incorrect.

---------38.19% delivery rate of valid orders

select 100.0 \* SUM(CASE WHEN S.Status = 'Delivered' THEN 1 ELSE 0 END) / COUNT(\*)  from Customer C

left join Orders o on o.Customer\_ID=C.Customer\_ID

left join Shipping s on s.Customer\_ID=C.Customer\_id

where O.Customer\_ID is not null and s.customer\_id is not null A screenshot of a computer program

AI-generated content may be incorrect.

**2. Dataset Requirements and Domain Model**

**Source Tables:**

Customer:

| **Field Name** | **Data Type** | **Description** | **Required** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- | --- |
| Customer\_ID | INT | Unique customer identifier | Yes | Primary Key, Not Null | 101 |
| First | varchar 50 | First name of the customer | Yes | Not Null | John |
| Last | varchar 50 | Last name of the customer | Yes | Not Null | Smith |
| Age | INT | Age of the customer | Yes | Between 10 and 100 | 35 |
| Country | varchar 3 | Country of residence | Yes | Not Null | USA |

Order:

| **Field Name** | **Data Type** | **Description** | **Required** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- | --- |
| Order\_ID | INT | Unique order identifier | Yes | Primary Key, Not Null | 5001 |
| Item | STRING | Product or item name | Yes | Not Null | Running Shoes |
| Amount | FLOAT | Total order amount | Yes | Must be greater than 0 | 120.5 |
| Customer\_ID | INT | Link to Customer table | Yes | Foreign Key, Not Null | 101 |
| Order\_Date  (Optional, better if included for better shipment analysis) | DATETIME | Date of order placed | Yes | Not Null | 2025-01-01 |

Shipping:

| **Field Name** | **Data Type** | **Description** | **Required** | **Constraints** | **Example** |
| --- | --- | --- | --- | --- | --- |
| Shipping\_ID | INT | Unique shipping identifier | Yes | Primary Key, Not Null | 9001 |
| Status | STRING | Delivery status | Yes | should be either 'Pending', 'Delivered' | Delivered |
| Customer\_ID | INT | Link to Customer table | Yes | Foreign Key, Not Null | 101 |
| Shipping\_Date  (Optional, better if included for better shipment analysis) | DATETIME | Date of order placed | Yes | Not Null | 2025-01-01 |

**Target Tables:**Create a star schema with Facts and Dimensions with 3NF normalisation to avoid data redundancy and to ensure the data model is future proof ready and allows scaling without any modifications.

| Table | Purpose | Grain |
| --- | --- | --- |
| dim\_customer | Customer Data | One row per customer |
| dim\_product | Product data | One row per unique product (by name) |
| dim\_geography | Country data | One row per unique country |
| dim\_date | Date table | One row per calendar day |
| fact\_sales | Sales transactions | One row per order (unique Order\_ID) |
| fact\_shipping | Shipping status | One row per shipping record |

Dim\_Customer:

| **Column** | **Data Type** | **Description** | **Transform/Constraints** |
| --- | --- | --- | --- |
| customer\_key | INT, PK | Surrogate key (auto) | Identity(1,1) |
| customer\_id | INT, UNIQUE | Business key | from source customer |
| first\_name | VARCHAR(50) | Cleaned first name | trim spaces, remove special chars |
| last\_name | VARCHAR(50) | Cleaned last name | trim spaces, remove special chars |
| full\_name | VARCHAR(101) | first\_name + ' ' + last\_name | Derived field |
| age | INT | Customer age | Check: age between 18 & 100 |
| age\_category | VARCHAR(20) | 'Under 30' or '30 and Above' | Derived: age < 30 = 'Under 30' |
| geography\_id | INT, FK | FK to dim\_geography |  |
| is\_active | BOOL | All set to TRUE unless deactivated |  |
| created\_timestamp | DATETIME | Timestamp of load date |  |
| updated\_timestamp | DATETIME | Timestamp of updated date |  |

Dim\_Product:

| **Column** | **Data Type** | **Description** | **Transform/Constraints** |
| --- | --- | --- | --- |
| product\_id | INT, PK | Surrogate key | Identity (1,1) |
| product\_name | VARCHAR(100) | Business key | from Orders, dedup |
| is\_active | BOOL | TRUE unless deactivated |  |
| created\_timestamp | DATETIME | Timestamp of load date |  |
| updated\_timestamp | DATETIME | Timestamp of updated date |  |

Dim\_Geography:

| **Column** | **Type** | **Description** | **Transform/Constraints** |
| --- | --- | --- | --- |
| geography\_key | INT, PK | Surrogate key | Identity(1,1) |
| country\_code | VARCHAR(3) | Standardized (e.g. USA) | From Customers, dedup |
| country\_name | VARCHAR(35) | e.g. "United States" | Lookup table if needed |
| is\_active | BOOL | TRUE unless deactivated |  |
| created\_timestamp | DATETIME | Timestamp of load date |  |
| updated\_timestamp | DATETIME | Timestamp of updated date |  |

NOTE: All dimension tables above can be implemented as SCD type 2 by adding a new column as END\_DATE along with the active flag that is present in the table above to track all the changes in the dimensions.

Dim\_Date:

| **Column** | **Type** | **Description** |
| --- | --- | --- |
| date\_key | INT, PK | e.g., YYYYMMDD |
| full\_date | DATE | Calendar date |
| year | INT | Year |
| quarter | INT | Calendar quarter |
| month | INT | Month |
| day | INT | Day of month |

Fact\_Sales:

| **Column** | **Type** | **Description** | **Transform/Constraints** |
| --- | --- | --- | --- |
| sale\_id | BIGINT, PK | Surrogate key (auto) | Identity(1,1) |
| order\_id | INT, UNIQUE | Natural key | from Source Order table |
| customer\_id | BIGINT, FK | Customer dimension | Join by customer\_id dim customer |
| product\_id | BIGINT, FK | Product dimension | Join by producy\_id dim product |
| geography\_id | BIGINT, FK | Geography dimension | Join by geography\_id dim geography |
| order\_date\_key(Optional if included in Source Order) | INT/FK | Date dimension | Join by Date\_Key dim\_date |
| amount | DECIMAL(10,2) | Order value | Always > 0 |
| quantity | INT | Number of items | Default 1 per row |
| created\_timestamp | DATETIME | Timestamp of load date |  |
| updated\_timestamp | DATETIME | Timestamp of updated date |  |

Fact Shipping:

| **Column** | **Type** | **Description** | **Transform/Constraints** |
| --- | --- | --- | --- |
| shipping\_key | INT, PK | Surrogate key | Identity (1,1) |
| shipping\_id | INT | Shipping natural key | From Source Shipping |
| order\_id | INT, FK | Order natural key | Join source Shipping ,dim\_customer and Order to extract, must not be NULL |
| customer\_id | INT, FK | Customer dimension | Join by customer\_id dim customer |
| status | VARCHAR(20) | 'Pending', 'Delivered', etc | Normalize case |
| shipping\_date\_key(Optional) | INT/FK | Ship date for delivery analysis | From Source Shipping |
| created\_timestamp | DATETIME | Timestamp of load date |  |
| updated\_timestamp | DATETIME | Timestamp of updated date |  |

**NOTE: The above two fact tables are implemented with SCD 1 types to simply overwrite any changes.**

**3. USER STORY: Fact Sales Table Implementation**

**Objective:**

Build a fact\_sales table that integrates and transforms transactional data from Orders, Customers sources. The objective is to enable accurate, analytics-ready sales fact table with referential integrity, data quality, and operational reporting capability.

Fact\_Sales:

| **Column** | **Type** | **Description** | **Transform/Constraints** |
| --- | --- | --- | --- |
| sale\_id | BIGINT, PK | Surrogate key (auto) | Identity(1,1) |
| order\_id | INT, UNIQUE | Natural key | from Source Order table |
| customer\_id | BIGINT, FK | Customer dimension | Join by customer\_id dim customer |
| product\_id | BIGINT, FK | Product dimension | Join by product\_id dim product |
| geography\_id | BIGINT, FK | Geography dimension | Join by geography\_id dim geography |
| order\_date\_key(Optional if included in Source Order) | INT/FK | Date dimension | Join by Date\_Key dim\_date |
| amount | DECIMAL(10,2) | Order value | Always > 0 |
| quantity | INT | Number of items | Default 1 per row |
| created\_timestamp | DATETIME | Timestamp of load date |  |
| updated\_timestamp | DATETIME | Timestamp of updated date |  |

Transformation Logics:

* Sale\_id: Generate this field as identity column which will be a PK
* Order\_id: join source order table with dim\_customer, extract the order\_id for all matching records
* customer\_id: join source order table with dim\_customer, extract the customer\_id for all matching records
* product\_id: Join source order table with dim\_customer, for all matching records join with dim\_product on product\_name to extract product\_id
* geography\_id: Join source Order table with dim\_customer, for all matching records join with dim\_geography on country\_code to extract geography\_id
* order\_date\_key (optional): Join source Order table with dim\_customer, for all matching records join with dim\_date on full\_date to extract order\_date\_key
* Amount: Join source order table with dim\_customer, for all matching records join extract amount
* Quantity: Since one row has information of individual product from an order it is defaulted to 1
* created\_timestamp: Timestamp when the transaction was first loaded to the system
* updated\_timestamp: Timestamp when the transaction was modified.

**QA Test Cases**

* No NULLs in order\_id, customer\_id,product\_id,geography\_id
* No duplicate order\_id
* One to one relationship between order\_id and customer\_id
* One to one relationship between order\_id and product\_id
* One to one relationship between geography\_id and customer\_id
* All amounts > 0 and rounded to 2 decimal places.
* All source orders present, counts reconcilation
* referential integrity checked on all FK with source tables.

**Business Reporting Requirements:**

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

| **Objective** | Analyze total spending across countries where orders are yet to be delivered. |
| --- | --- |
| **Metrics** | SUM(Amount) |
| **Grouping** | By Country |
| **Filter** | Shipping\_Status = 'Pending' |
| **Required Fields** | Country\_code, Amount, Shipping\_Status |
| **Tables** | Sales Fact, Dim Customer, Fact Shipping,Dim Geography |
| **Transformation Notes** | 1. Join sales and shipping fact on customer\_id and Order\_id 2. join with sales and Dim\_geography on geography\_id to get Country\_code 3. Filter for Shipping status="Pending" and group by Country |

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

| **Objective** | Get a full customer-level breakdown of their order behavior |
| --- | --- |
| **Metrics** | COUNT(Order\_ID), SUM(Quantity), SUM(Amount) |
| **Grouping** | Customer\_ID, First\_Name, Last\_Name, Item |
| **Required Fields** | Order\_ID, Customer\_ID,First\_Name,Last\_Name Item, Quantity, Amount |
| **Tables** | Fact Sales, Dim Customer |
| **Transformation Notes** | 1.Join Fact sales with Dim customer on customer\_id  2.Join Sales with Dim\_product  3.Group by Customer\_ID, First\_Name, Last\_Name, Item |

1. the maximum product purchased for each country.

| **Objective** | Identify the product with the highest total quantity purchased in each country |
| --- | --- |
| **Metric** | SUM(Quantity) |
| **Grouping** | By Country\_code, Item |
| **Required Fields** | Country\_code, Item, Quantity |
| **Source Tables** | Sales Fact, dim\_geography,dim\_product |
| **Transformation Notes** | 1. Join sales fact with dim geography to get country\_code 2. join sales with dim product on product\_id to get Item names 3. Use rank function to find the highest product sold in each country |

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

| **Objective** | Compare product popularity between young and mature customers |
| --- | --- |
| **Metric** | SUM(Quantity) |
| **Grouping** | By Age\_Category, Item |
| **Required Fields** | Age, Item, Quantity |
| **Tables** | Sales Fact, Dim product,Dim customer |
| **Transformation Notes** | 1. Join sales fact with dim customer on customer id 2.Join sales fact with dim product on product id 3.Group by Age\_category and Item |

1. the country that had minimum transactions and sales amount.

| **Objective** | Identify the least engaged countries based on transaction volume and sales |
| --- | --- |
| **Metric** | COUNT(Order\_ID), SUM(Amount) |
| **Grouping** | By Country\_code |
| **Required Fields** | Country\_code, Order\_ID, Amount |
| **Tables** | Sales Fact, Dim product,Dim customer |
| **Transformation Notes** | 1.Join sales fact with dim geography on geography id 2. 3. Use rank function to find the lowest orders count and Sum of Amount for each country |