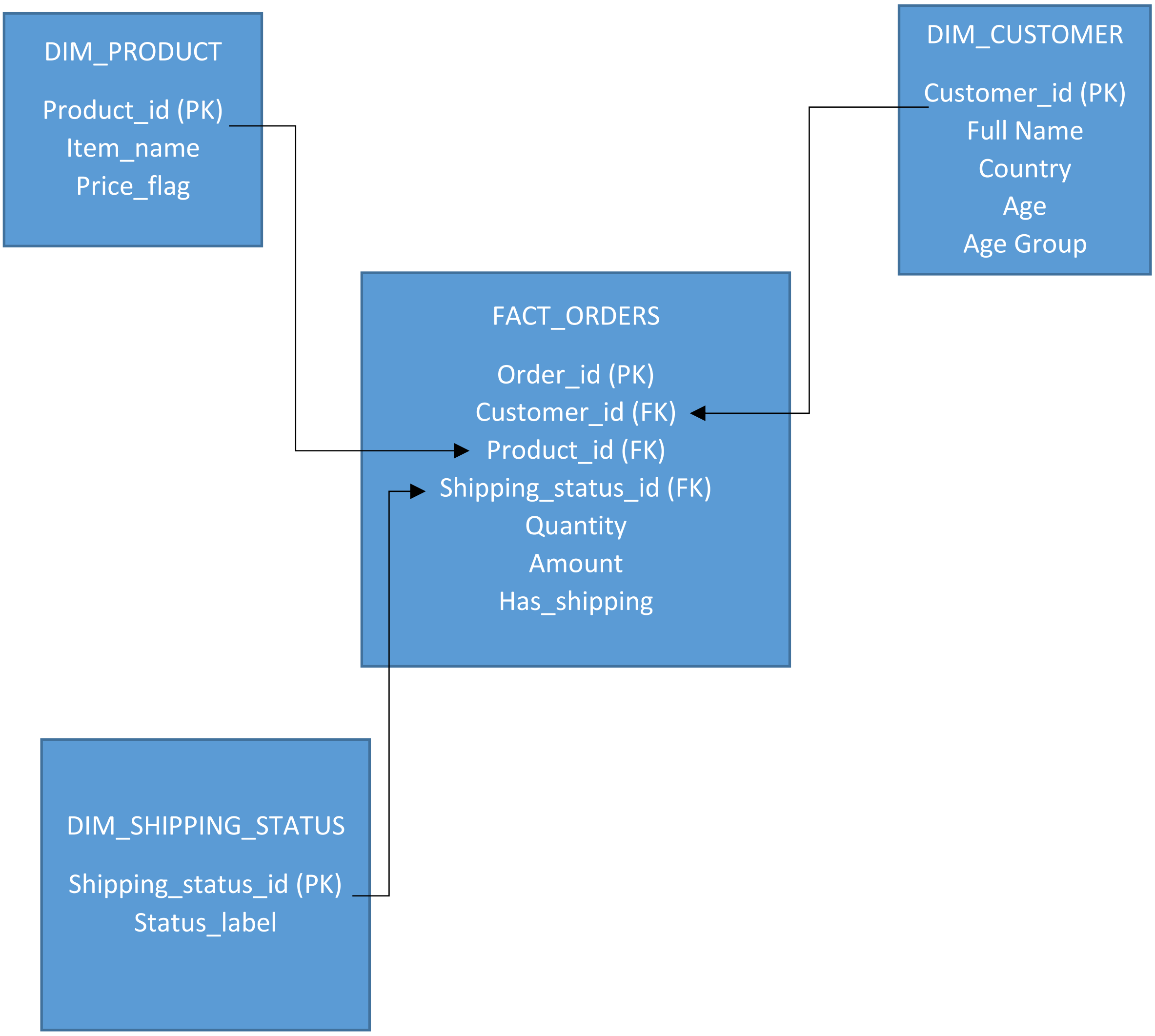
Model Design :



DIM_PRODUCT

Product_id (PK)
Item_name
Price_flag

DIM_CUSTOMER

Customer_id (PK)
Full Name
Country
Age
Age Group

FACT_ORDERS

Order_id (PK)
Customer_id (FK)
Product_id (FK)
Shipping_status_id (FK)
Quantity
Amount
Has_shipping

DIM_SHIPPING_STATUS

Shipping_status_id (PK)
Status_label

FACT_ORDERS

Field	**Type**	**Description**
Order_ID	PK	Unique identifier per order
Customer_ID	FK	Linked to DIM_CUSTOMER
Product_ID	FK	Linked to DIM_PRODUCT
Shipping_Status_ID	FK	Linked to DIM_SHIPPING_STATUS
Amount	Float	Amount paid
Quantity	Integer	Set as 1
Has_shipping	Boolean	True if there is matching record in shipping.json

DIM_CUSTOMER

Field	**Type**	**Description**

Customer_ID	Integer	From source
Full_name	String	Concat(First,Last) from customer
Age	Integer	From source
Age_group	String	Derived : “<30”, “30+”
Country	String	From source

DIM_PRODUCT

Field	**Type**	**Description**
Product_ID	Integer	System generated key
Item	String	Distinct item From Orders
Price_flag	Boolean	TRUE if multiple prices Exist

DIM_SHIPPING_STATUS

Field	**Type**	**Description**
Shipping_Status_ID	Integer	System generated key
Status_label	String	“Pending”, “Delivered”

Story for Data Engineers:

Title : Build End-to-End Dimensional Model for Customer Order Data

As a Senior Data Analyst,
I want to ingest, cleanse, and model data from Customer.xls, Order.csv, and Shipping.json into a refined dimensional model consisting of 3 dimension tables and 1 fact table
so that business users can access high-quality, analysis-ready data that supports reporting on customer behavior, product sales, and delivery performance.

Tables :
1. DIM_CUSTOMER

-Filter invalid ages (< 0 or > 100)
-Remove NULL Customer_IDs
-Derive Full_Name = Concat (First, last) from customer
    Age_Group = case when age < 30 then ‘<30’ else ‘30+’ end

2. DIM_PRODUCT

- Deduplicate items
- Flag products with inconsistent pricing (e.g., Mousepad = 200, 250)

3. DIM_SHIPPING_STATUS

- Extract unique status values from Shipping.json

4. FACT_ORDER

-Join Order.csv to Customer.csv using Customer_ID
- Left join to Shipping.json (via Customer_ID)
-Flag Has_Shipping = FALSE where Shipping record is missing

Creation order:
- Create DIM_ tables first
- Create FACT_ORDER referencing ids
- Load in dependency order to ensure FK resolution

Load Frequency: One-time load for now (can be scheduled daily in production)
Add this timestamp as last field in each table.

Acceptance Criteria

- DIM_CUSTOMER, DIM_PRODUCT, DIM_SHIPPING_STATUS, and FACT_ORDERS tables are created with correct schema.
- All tables include Refresh_Timestamp
- Invalid or null Customer_ID, Age, and Country are excluded or flagged in DIM_CUSTOMER.
- Age_Group is derived correctly (<30, 30+).
- DIM_PRODUCT deduplicates items and sets Price_Flag = TRUE where multiple prices exist.
- Has_Shipping flag in FACT_ORDERS is set to FALSE if no matching shipping record.
- Referential integrity is maintained:
  Customer_ID in FACT_ORDERS exists in DIM_CUSTOMER
  Product_ID in FACT_ORDERS exists in DIM_PRODUCT
  Shipping_Status_ID in FACT_ORDERS exists in DIM_SHIPPING_STATUS
- Orders without shipping are included (orphan shipping records are excluded).
- Row counts match expected values after cleaning.

Story for QA Team:

As a Data Analyst
I want to ensure that the dimensional model built by the Data Engineering team, including all fact and dimension tables, meets accuracy, completeness, and referential integrity standards
so that the reporting and analytics layer is powered by reliable, trusted data that aligns with business logic and source definitions.

Tables to Validate

DIM_CUSTOMER
DIM_PRODUCT
DIM_SHIPPING_STATUS
FACT_ORDER

Test Scenarios:

Test	Table	Expected Results
keys are unique	All Dim tables	No duplicates in keys/ids
Columns not null	All	No nulls in critical fields
Data types match spec	All	Int, string, float etc. as defined
FACT_ORDER.Customer_id exists in DIM_CUSTOMER	Fact_order	Should be present
FACT_ORDER.Product_id exists in DIM_PRODUCT	Fact_order	Should be present
FACT_ORDER.Shipping_Status_ID exists in DIM_SHIPPING_STATUS	Fact_order	Should be present
Age group logic (<30, 30+)	DIM_CUSTOMER	Correctly labeled based on Age
Has_Shipping = TRUE only when shipping exists	FACT_ORDER	Derived correctly from joined data
Pricing conflicts flagged	DIM_PRODUCT	Products with multiple prices marked TRUE
Duplicate Order_IDs in source		Exclude or deduplicate
Orphan shipping records (no order)		Excluded or flagged
Orders with no shipping		Has_Shipping = FALSE
NULL Age or Country in DIM_CUSTOMER		Row excluded or marked invalid
Inconsistent Item name casing		Normalized in `DIM_PRODUCT`

Acceptance Criteria:

- No referential integrity breaks
- No critical NULLs
- Business rules consistently applied
- Edge cases (e.g., duplicate IDs, inconsistent prices) flagged or excluded
- Row counts, sum of financials between raw → refined match expectation