3) User Story

Build "Order" Subject Area (Core Dims + Order Fact) in Warehouse

Summary

This project focuses on building a curated data model for the Orders domain to support downstream analytics. The model includes three conformed dimensions (**Customer, Product, Country**) and one fact table (**Order**). It also specifies documented transformations from source to target, data quality rules, and a simple incremental loading approach.

Background / Context

We receive three flat files from Sales/Ops:

- **Customer**: Customer_ID, First, Last, Age, Country
- **Order**: Order_ID, Item, Amount, Customer_ID
- **Shipping**: Shipping_ID, Status, Customer_ID (customer-grain, duplicates per customer, no Order_ID)

For this project, the focus is exclusively on the **Orders** domain (facts + core dims). Shipping data will be modeled separately.

In Scope

- Curated schemas/tables for **d_country**, **d_product**, **d_customer**, and **f_order**.
- Deterministic transformations from source to target, including data cleansing for names and basic validations.
- A basic incremental loading strategy based on a CDC (Change Data Capture) timestamp.
- QA checks for row counts, foreign key integrity, and domain rules.

Business Requirements

- To support country-level and customer-level aggregations for spend, transactions, and product mix.
- To enable analysis of product popularity by country and age band.
- To create stable, conformed dimensions for consistent slicing and dicing across various analyses.

Target Data Model

Fact: f_order (order-line grain)

- **Grain**: One row represents a single order line (each record in the Order feed).
- **Keys**: order_id (PK), customer_id (FK), product_id (FK), country_id (FK).
- **Measures**: amount (numeric, must be > 0), quantity (default is 1 until a source quantity is provided).
- **Notes**: The country_id is stamped at load time from the customer's record to conform with the d_country dimension.

Dim: d_customer (customer grain)

- **Grain**: One row per customer (customer_id from the source).
- **Attributes**: first_name, last_name, age, country_id (FK).
- **Cleansing**: Name hygiene (map "leetspeak," strip disallowed symbols, trim, and collapse spaces). Age is validated to be between 10 and 100.

Dim: d_product (product grain)

- **Grain**: One row per distinct Item from the orders data.
- **Attributes**: product_name (from Item).
- **Placeholders**: category, unit_price (nullable, for future use).

Dim: d_country (country grain)

- **Grain**: One row per distinct country.
- **Attributes**: country_name (must be unique).

Shipping: Shipping data is **not** joined to the order fact in this project. A separate customer-grain status structure will be designed.

Source → Target Requirements

- **d_country**: Populated with a distinct list of countries from Customer.Country.
- **d_product**: Populated with a distinct list of items from Order.Item.
- **d_customer**:
  - customer_id is taken directly from the source.
  - **Name hygiene**: Map 0→o, 1→i, 3→e, 4→a, 5→s, 7→t, @→a, !→i. Remove other punctuation except for space, apostrophe, and hyphen. Trim and collapse internal spaces.
  - country_id is populated via a lookup on d_country using country_name.
- **f_order**:
  - order_id is taken from the source (and must be unique).
  - customer_id comes from the source and must resolve to an existing record in d_customer.
  - product_id is populated via a lookup on d_product using product_name.
  - country_id is copied from the resolved customer’s country_id in d_customer.

    - amount must be **> 0**.
    - quantity is set to 1.

Data Quality & Validation Rules

Integrity

- **PK uniqueness**: order_id and customer_id must be unique.
- **FK coverage**: Every f_order.customer_id, product_id, and country_id must resolve to a valid record in its respective dimension table.

Domains

- amount must be greater than 0.
- Names must conform to the post-cleaning whitelist of characters (letters, space, apostrophe, hyphen).

Row Parity

- The f_order row count should equal the number of valid orders in the source after business rule filtering.
- The d_customer row count should equal the number of customers who pass the name and age rules.

Join Stability

- Joining f_order to all three dimensions should not change the row count of f_order.

Load Strategy

- **Initial Load**: Perform a full load of all dimension tables first, followed by an upsert/merge for f_order to prevent duplicates.
- **Incremental (CDC time if present)**: Use a source "updated at" or watermark timestamp to pull only the changed rows. Perform upserts on the dimensions by natural key (customer_id, product_name, country_name) and on the fact table by order_id.

Acceptance Criteria

1. **Tables exist** in the curated schema: d_country, d_product, d_customer, f_order.
2. **Row counts** are as expected:
    - f_order count equals valid orders (after amount > 0 rule).
    - d_customer count equals customers passing all validation rules.
3. **No foreign key violations** occur when joining the fact table to its dimensions.
4. **Domain checks pass** (no negative/zero amounts; names are clean).

5. **Join stability**: COUNT(f_order) equals COUNT(f_order ⋈ d_customer ⋈ d_product ⋈ d_country).
6. An **incremental run** successfully updates only the delta rows.

Dependencies / Assumptions

- Access to raw data files in a landing zone (e.g., mdsdb).
- If CDC timestamps are unavailable, a simple, idempotent upsert is an acceptable substitute given the small data volume.
- There is no existing product catalog or price master table to leverage yet.

Risks / Notes

- "Item" is the only product identifier. If the naming changes upstream, it could create new, unintended product rows. A product master should be introduced in the future.
- The shipping data is **customer-grain**; it **must not** be joined directly to the f_order table.

Deliverables

- Curated tables ready for analytics: xdsdb.d_country, xdsdb.d_product, xdsdb.d_customer, xdsdb.f_order.

Test Scenarios

- **PK/Unique**: Test that inserting a duplicate order_id results in an update or is rejected, rather than creating a new row.
- **FK Coverage**: Intentionally remove a country from the dimension and confirm that the fact load flags an exception.
- **Name Cleaning**: Test that inputs like N!cole, L@rry, R0bert, Al1cia are correctly cleaned.
- **Amount Rule**: Inject an amount=0 or negative value and confirm it results in an exception.