**Why We Used Fact and Dimensional Modeling**

In this project, we adopted **fact and dimensional modeling** to structure the Olist e-commerce dataset in a way that supports **scalable, efficient, and business-friendly analytics**.

**What is Dimensional Modeling?**

Dimensional modeling is a data modeling technique commonly used in data warehouses. It organizes data into:

* **Fact Tables**: These store quantitative, measurable metrics (e.g., sales, revenue, shipping cost).
* **Dimension Tables**: These provide context to the facts (e.g., customer details, product categories, dates, seller regions).

This structure supports **OLAP (Online Analytical Processing)** and allows business users to explore data through **intuitive slicing and dicing**.

**Use Case in Olist E-Commerce Data**

We modeled the Olist dataset using:

* A central **fact table** (fact\_order\_summary) containing key metrics like item price, freight value, and delivery performance.
* Several **dimension tables**:
  + dim\_customer – who bought the item
  + dim\_seller – who sold it
  + dim\_product – what was sold
  + dim\_date – when it was purchased

Explanation BELOW:

**fact\_orders — *One Row per Order***

**Purpose**: Acts as the primary grain for order-level analysis (e.g., volume, status, customer behavior).

**Steps & Joins**:

* Joins with stg\_orders, dim\_customer, and dim\_date to include:
  + customer\_id, order\_status, and a surrogate key for the purchase date (order\_purchase\_date\_sk).
* No aggregation here; each row represents **one distinct order**.
* Enables analysis such as:
  + *How many orders were placed per customer?*
  + *What is the distribution of order statuses over time?*

**fact\_order\_items — *Line-Item Level***

**Purpose**: Supports detailed analysis at the item level (e.g., pricing, logistics, seller performance).

**Steps & Joins**:

* Joins:
  + With fact\_orders to inherit order\_id and ensure referential integrity.
  + With dim\_product and dim\_seller for product and seller details.
* No explicit aggregation; each row is **one product line in one order**.
* Includes metrics like item\_price, freight\_value, and is\_hazardously\_heavy.
* Enables:
  + *Unit economics analysis (revenue vs. shipping cost)*.
  + *Seller-level delivery performance or product movement*.

**fact\_payments — *Payment-Level Aggregation***

**Purpose**: Tracks financial inflows per order.

**Steps & Joins**:

* Joins with fact\_orders on order\_id.
* Aggregates payment records per order to calculate:
  + payment\_value = **total paid** for each order, across installments or multiple methods.
* Enables:
  + *Revenue calculation*.
  + *Payment type analysis (e.g., how many orders were paid in installments vs upfront)*.

**fact\_reviews — *Customer Feedback Insights***

**Purpose**: Captures satisfaction and response timing metrics.

**Steps & Joins**:

* Joins with fact\_orders using order\_id.
* Includes:
  + review\_score: Numeric score from 1 to 5.
  + Derived fields like response\_time\_days for how long after delivery the review was posted.
* Aggregations (if present) include:
  + Average review score by product, seller, or customer segment.
* Enables:
  + *Sentiment analysis* and *quality tracking* across sellers or products.

**👤 dim\_customer — *Customer Attributes & Geography***

**Purpose**: To provide unique, standardized information about customers for demographic and location-based insights.

**Steps & Structure**:

* One row per customer\_id.
* Standardizes:
  + customer\_city and customer\_state using trimming and casing functions.
  + ZIP codes with zip\_prefix\_int (numeric for joins) and zip\_prefix (zero-padded string for display).
* Includes customer\_unique\_id for cross-session tracking (important for long-term user behavior analysis).

**Enables**:

* Grouping orders by customer region or city.
* Segmenting buyers for behavioral analysis (e.g., repeat customers by ZIP prefix).
* Accurate geographic joins with geolocation data.

**dim\_geolocation — *Zip Code-Level Spatial Mapping***

**Purpose**: Links ZIP prefixes to geographic coordinates for mapping and regional analytics.

**Steps & Structure**:

* One row per ZIP code prefix.
* Includes:
  + geolocation\_lat and geolocation\_lng (latitude and longitude centroids).
* Validated with NOT NULL and UNIQUE tests on all fields.

**Enables**:

* Spatial joins to compute delivery distances.
* Geographic aggregation of customers and sellers.
* Regional dashboards and choropleth maps.

**dim\_product — *Product Category Lookup***

**Purpose**: Standardizes product identifiers and their associated categories.

**Steps & Structure**:

* One row per product (product\_sk), acting as the product surrogate key.
* Category column provides a high-level classification of items.

**Enables**:

* Aggregation of revenue or order volume by product category.
* Identification of top-performing or underperforming product types.
* Analysis of category-level trends and seasonal performance.

**dim\_seller — *Seller Location & Geography***

**Purpose**: Stores seller identifiers and geographic information for supply-side analysis.

**Steps & Structure**:

* One row per seller, linked via seller\_sk.
* ZIP prefix is used to:
  + Join with dim\_geolocation for region-level mapping.
  + Enable performance comparisons across cities and states.

**Enables**:

* Ranking sellers by delivery success or review scores.
* Mapping seller distribution across Brazil.
* Analyzing seller contribution to freight costs or order delays.

**dim\_date — *Calendar for Time-Based Analysis***

**Purpose**: Acts as the calendar table for joining and filtering by various date granularities.

**Steps & Structure**:

* One row per date, providing:
  + date\_sk (surrogate key),
  + Day, month, quarter, and year breakdowns.
* Used to partition fact tables by order\_purchase\_date, enabling efficient querying.

**Enables**:

* Time series analysis (daily, monthly, quarterly sales).
* Seasonal trend evaluation.
* Comparing performance across time windows.