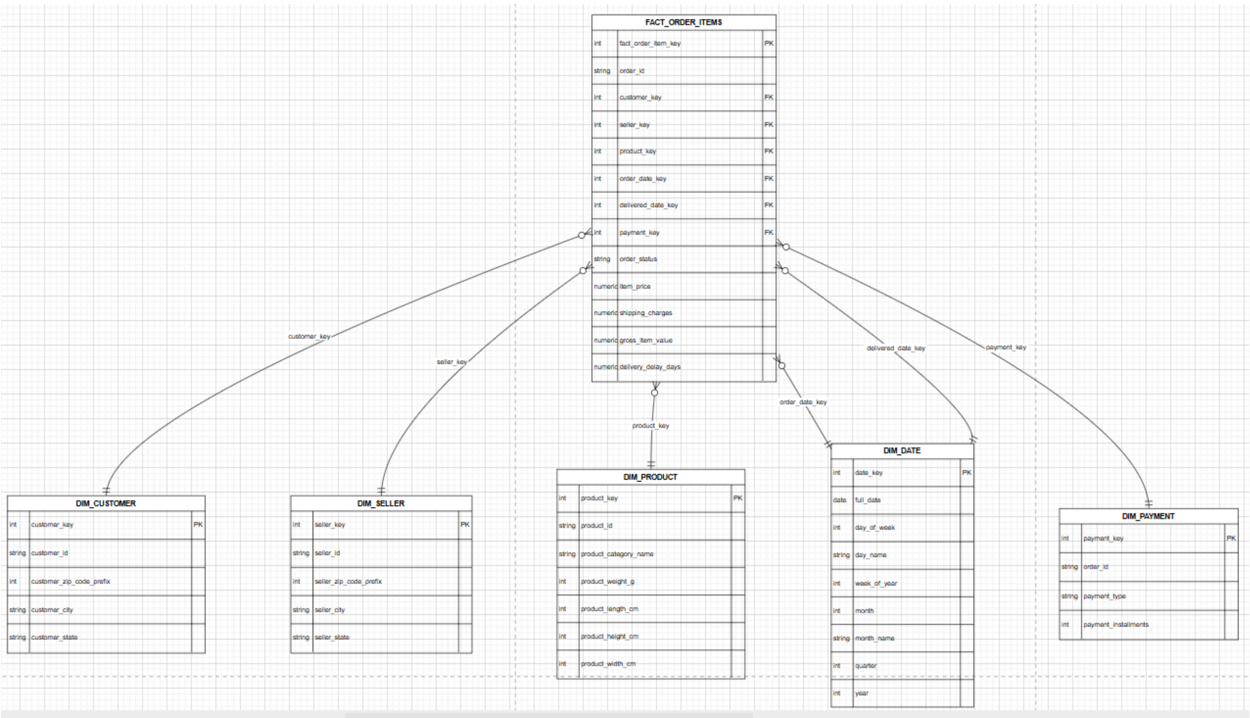
Dimensional Modeling & Star Schema Design



FACT_ORDER_ITEMS		
int	fact_order_item_key	PK
string	order_id	
int	customer_key	FK
int	seller_key	FK
int	product_key	FK
int	order_date_key	FK
int	delivered_date_key	FK
int	payment_key	FK
string	order_status	
numeric	item_price	
numeric	shipping_charges	
numeric	gross_item_value	
numeric	delivery_delay_days	

DIM_CUSTOMER		
int	customer_key	PK
string	customer_id	
int	customer_zip_code_prefix	
string	customer_city	
string	customer_state	

DIM_SELLER		
int	seller_key	PK
string	seller_id	
int	seller_zip_code_prefix	
string	seller_city	
string	seller_state	

DIM_PRODUCT		
int	product_key	PK
string	product_id	
string	product_category_name	
int	product_weight_g	
int	product_length_cm	
int	product_height_cm	
int	product_width_cm	

DIM_DATE		
int	date_key	PK
date	full_date	
int	day_of_week	
string	day_name	
int	week_of_year	
int	month	
string	month_name	
int	quarter	
int	year	

DIM_PAYMENT		
int	payment_key	PK
string	order_id	
string	payment_type	
int	payment_installments	

Introduction to the Star Schema

To support the analytical workload and reporting beyond what was possible on the OLTP, a dimensional model was designed using a Star Schema approach. This schema rearranges the transactional dataset into a format that best supports aggregations, the analysis of trends, and business intelligence queries. The model places a central fact table at the core of the schema, surrounded by several descriptive dimension tables that store contextual information relative to customers, sellers, products, dates, and payments.

Defining the Grain of the Fact Table

The grain of the fact table, FACT_ORDER_ITEMS, is at the level of an individual order item line; that is, each row in this table records one product that was bought as part of a given order. This grain was selected because this way analysts can drill down to the most detailed level of activity, while still being supported at higher-level aggregations like order level, seller level, customer level, product level, and date level summaries. All foreign keys and measures are precisely consistent with this grain.

Purpose of Each Dimension Table

Each dimension table provides descriptive attributes to enhance analytical queries.

• DIM_CUSTOMER stores demographic and geographic information about customers, and also supports segmenting customer data by city, state, or ZIP code.

• DIM_SELLER: It contains location and identity information about each seller. Therefore, it can be used to compare the performance of sellers across different marketplaces.

• DIM_PRODUCT describes product characteristics such as category and physical dimensions to support category-level and product-level sales analysis.

•DIM_DATE standardizes all relevant timestamps, such as order date and delivery date, into a single reusable date dimension supporting analysis by day, month, quarter, or year.

• DIM_PAYMENT: This stores the attributes related to payment methods and installment behavior, enabling the analysis of payment patterns and financial metrics.

Note on Improving the Seller Dimension

In the OLTP schema, the sellers table contains only a single attribute, seller_id. This level of detail is enough to maintain transactional integrity but is quite insufficient for analytical workflows. In dimensional modeling, each dimension table should supply rich descriptive attributes that allow meaningful slicing, grouping, and filtering. Therefore, the seller dimension was intentionally expanded to include additional analytical attributes such as seller ZIP prefix, seller city, and seller state. These attributes can be derived during ETL from the transactional dataset and greatly enhance the analytical capability by enabling analysts to evaluate the performance of sellers geographically, compare regional efficiency in delivery, and model marketplace behavior across different locations. It is a usual and accepted standard in star schema design, where dimensions are expected to hold descriptive context not always present in the OLTP source.

Measures and Metrics in the Fact Table

The fact table stores all the numerical values and metrics needed for meaningful analysis: item price, shipping charges, total item value, delivery delay days. Since all measures are at the chosen grain, analysts will easily aggregate data to report on seller performance, customer behavior, summarization of category performance, or time-based trends. This structure ensures analytical queries will run efficiently and do not repeatedly join the OLTP tables.

Mapping OLTP Schema to the Dimensional Model

The dimension model was derived directly from the OLTP tables: orders, order_items, products, customers, sellers, and payments. Natural keys from these source tables were transformed into surrogate keys in the dimension tables. Facts were assembled by joining transactional tables along their foreign keys, and timestamps were mapped to the shared date dimension. This separation of concerns reduces redundancy and improves query performance by avoiding multiple complex joins.

Analytical Benefits of the Star Schema

This star schema enables fast, flexible, and intuitive analysis, reducing join complexity and aligning data with typical business questions. Analysts can evaluate seller performance, category revenue, customer segmentation, and delivery efficiency in a substantially improved query performance manner. The schema also supports adding new dimensions or facts in the future without restructuring the whole database.

ETL and Potential dbt Integration

Although this phase does not implement it, the schema is structured to be compatible with ETL pipelines, including dbt. dbt models can be used to transform OLTP tables into dimension and fact models, apply data quality checks, document lineage, and automate refresh schedules. This positions the star schema to serve as the foundation for a scalable analytics workflow.