A deep understanding of the data schema is fundamental to any successful data project.

Providing this context will prevent employees from treating the data as just abstract numbers and will enable them to ask better questions and build more meaningful features.

Here is a detailed, table-by-table and column-by-column description of the Olist dataset, formatted for easy explanation.

Comprehensive Data Dictionary for the Olist E-commerce Project

Introduction to the Data Schema

Imagine you are a data scientist/data analyst at Olist. The company doesn't store all its information in one giant spreadsheet. Instead, it uses a **relational database**. This means data is organized into multiple tables, each dedicated to a specific entity (like customers, orders, or products). These tables are linked together by **keys**. This structure is efficient, reduces data duplication, and maintains data integrity.

Our first major task in this project is to use SQL to "join" these tables back together to create a single, comprehensive view of each customer's journey.

Why are we using multiple tables?

- **Efficiency:** You don't need to repeat a customer's address and name for every single item they buy. You store the customer's information once in a customer’s table and just link to it.
- **Scalability:** The business can add new products without changing the orders table, or add new payment methods without changing the customers table.
- **Clarity:** Each table represents a clear, logical concept (an order, a payment, a review).

Here is a breakdown of the key tables we will use and what each column represents.

1. The olist_customers_dataset Table

This table holds information about the customer, but it's slightly tricky. It's best to think of it as information related to a *specific order's delivery destination*.

Column Name	Data Type	Description & Purpose	Why We Care
customer_id	Text	**Primary Key (for an order)**. This ID is generated for *each new order*. If a single person makes 3 separate orders, they will have 3 different customer_ids. This is the key we use to link to the orders table.	Crucial for joining with orders.
customer_unique_id	Text	**The True Customer Identifier**. This ID is unique to each individual person. If a person makes 3 orders, they will have 3 different customer_ids but only **one** customer_unique_id.	**This is the most important ID for our project.** We will be aggregating all data to this level to analyze a customer's lifetime behavior.
customer_zip_code_prefix	Integer	The first 5 digits of the customer's zip code for the delivery address.	Can be used for geographic analysis (e.g., "Do customers from certain regions churn more?").
customer_city	Text	The city of the delivery address.	Also for geographic analysis.
customer_state	Text	The state of the delivery address (e.g., 'SP' for São Paulo).	A key feature for segmenting customers by region.

2. The olist_orders_dataset Table

This is the central table that records every order placed on the platform.

Column Name	Data Type	Description & Purpose	Why We Care
order_id	Text	**Primary Key**. A unique identifier for each order. This is the central key that links payments, reviews, and items to a specific order.	Essential for connecting almost all other tables together.
customer_id	Text	**Foreign Key**. Links to the customer_id in the customers table.	This is the bridge between an order and the customer who placed it.
order_status	Text	The status of the order (e.g., delivered, shipped, canceled).	Crucial for data cleaning. For our churn analysis, we should probably only focus on delivered orders to ensure we are analyzing completed transactions.
order_purchase_timestamp	Timestamp	The exact date and time when the customer made the purchase.	**Extremely important.** We will use this to calculate customer tenure, recency, frequency, and our final churn definition.
order_approved_at	Timestamp	Timestamp of payment approval.	Can be used to calculate processing time.
order_delivered_carrier_date	Timestamp	Timestamp when the order was handed to the logistics partner (carrier).	Can be used to analyze shipping efficiency.
order_delivered_customer_date	Timestamp	The actual delivery date of the order to the customer.	Can be compared with order_estimated_delivery_date to see if deliveries are on time. Late deliveries likely lead to churn.
order_estimated_delivery_date	Timestamp	The estimated delivery date shown to the customer at the time of purchase.	A key part of the customer experience.

3. The olist_order_items_dataset Table

This table details the contents of each order. An order can contain multiple items, so there can be multiple rows for a single order_id.

Column Name	Data Type	Description & Purpose	Why We Care
order_id	Text	**Foreign Key**. Links to the order_id in the orders table.	Connects specific products to their parent order.
order_item_id	Integer	A sequential number for each item within the same order (1, 2, 3...).	Allows us to count the number of items per order.
product_id	Text	**Foreign Key**. Identifier for the product. Can be used to link to a (missing) products table.	Allows for product-level analysis (e.g., "Do customers who buy electronics churn more?").
seller_id	Text	**Foreign Key**. Identifier for the seller of that product.	Allows for seller-level analysis.
shipping_limit_date	Timestamp	The deadline for the seller to ship the item.	Can be used to measure seller performance.
price	Float	The price of that single item.	**Core metric.** We will sum this to get the total value of an order and a customer's lifetime value.
freight_value	Float	The shipping cost for that single item.	**Core metric.** This is part of the total cost to the customer. High freight costs might be a reason for churn. We should add this to price.

4. The olist_order_payments_dataset Table

This table contains information about how an order was paid for. An order can have multiple payment methods (e.g., part with a credit card, part with a voucher).

Column Name	Data Type	Description & Purpose	Why We Care
order_id	Text	**Foreign Key**. Links to the order_id in the orders table.	Connects payment information to the order.
payment_sequential	Integer	For orders with multiple payment methods, this indicates the sequence (1, 2...).	Useful for understanding complex payment behaviors.
payment_type	Text	The method of payment (e.g., credit_card, boleto, voucher).	A great categorical feature. Do customers who use credit cards behave differently from those who use boleto (a popular Brazilian cash payment system)?
payment_installments	Integer	The number of payment installments (for credit card payments).	Can be a feature. Do customers who pay in many installments have a different churn profile?
payment_value	Float	The transaction value for this specific payment method.	We can sum this up per order_id and it should match the sum of price + freight_value from the order_items table. It's a good way to verify data consistency.

5. The olist_order_reviews_dataset Table

This table contains customer reviews for each order.

Column Name	Data Type	Description & Purpose	Why We Care
review_id	Text	A unique identifier for the review.	-
order_id	Text	**Foreign Key**. Links to the order_id in the orders table.	Connects the review to the order it belongs to.
review_score	Integer	A score from 1 (very unsatisfied) to 5 (very satisfied) given by the customer.	**This is one of the most powerful predictors of churn.** Low scores are a huge red flag. We will likely calculate the average review score for each customer.
review_comment_title	Text	The title of the review comment (optional).	Can be used for Natural Language Processing (NLP).
review_comment_message	Text	The text content of the review (optional).	A goldmine for NLP. We could perform sentiment analysis on this text to get a more nuanced view of customer satisfaction than just the 1-5 score.
review_creation_date	Timestamp	The date the review was sent by the customer.	-
review_answer_timestamp	Timestamp	The date the review was answered by the seller.	We can calculate the response time, which is another measure of customer service quality.