Table of Contents

1. Data: Tables & Columns

1. To create a data table

Use this prompt:

create a date table that has the dates from 03-01-2025 (March 1st, 2025)

to 05-31-2025 (May 31st, 2025)

2. To create an exchange rate table

Use this prompt:

Create an exchange_rate table for the period from March 1st, 2025, to May 17th, 2025. Use the following API request sample to fetch historical exchange rates:

response = requests.get(f'https://openexchangerates.org/api/historical/2024-10

21.json?app_id=<Replace your app id here>&symbols=INR')

Replace the date for each in the URL with each date within the specified range.

3. Data cleaning and summarizing required data in one table **(fact_summary)**

Use this prompt:

Create a Python code that:

1. Load data from:

- fact_order_line table
- dim_products sheet (skip first row, use second row as headers)
- dim_customers sheet (skip first row, use second row as headers)
- Exchange Rate sheet (skip first row, use second row as headers)

2. Cleans the data:

- Convert product_id and customer_id to numeric
- Strip whitespace from IDs - Remove rows with NULL IDs
- Convert IDs to integers
- Convert dates to datetime

3. Merges the tables:

- Orders with products using product_id

- Result with customers using customer_id

- Result with exchange rates using order_placement_date

4. Calculates total amounts:

- For USD currency: price_USD * USD_INR_Rate * order_qty

- For INR currency: price_INR * order_qty

5. Clean final output:

- Drop intermediate columns: price_USD, price_INR, currency, Date, USD_INR_Rate

- Keep only essential columns: order details, IDs, dates, quantities, delivery information, and total_amount (in INR)

4. New Columns **(fact_summary)**

1. **backorder_qty** = order_qty - delivery_qty
2. **order_cycle_time_days** = actual_delivery_date - order_placement_date
3. **delivery_delay_days** = actual_delivery_date - agreed_delivery_date
4. **in_full_percent** = delivery_qty / order_qty * 100
5. **on_time_flag** = 1 if actual_delivery_date <= agreed_delivery_date else 0
6. **category_demand_variability** = Std Dev of order_qty per week (needs grouping)
7. **week_of_year** = Extract from dim_date_table using order_placement_date
8. **lead_time_variability =** STDDEV(order_cycle_time_days)

2. Business Question: Exercise 1

1. Creating Business KPIs

Use this **Prompt:**

Create the following KPIs

1. Total Order Lines
2. Line Fill Rate
3. Volume Fill Rate
4. Total Orders
5. On Time Delivery %
6. In Full Delivery %
7. On Time In Full %
8. Order Cycle Time
9. Perfect Order Rate
10. Backorder Rate
11. Demand Variability
12. Inventory Turnover Rate (if inventory data available)
13. Delivery Reliability Index
14. Lead Time Variability
15. Average Unit Fill % per Order

2. Top Customers

Use this Prompt:

Create the following KPIs

Show me top 5 customers based on order value and their OTIF %, IF %, OT %.

Also add the customer name, customer ID and city in the table

3. Top Customers (India)

Use this Prompt:

Show me top 5 customers in India based on order value and their OTIF %, IF %, OT %.

Also add the customer name, customer ID and city in the table

3. Business Question: Exercise 2

1. Quantify the revenue loss attributed to undelivered orders. Revenue Loss from Undelivered Orders (in Millions):

Prompt:

Using fact_summary, calculate revenue loss per customer as:

1. revenue_loss = (order_qty - delivery_qty) * unit price (converted to INR). If currency is USD, use exchange rate from exchange_rate sheet based on order_placement_dat
2. Sum total revenue value (order_qty * unit price in INR)
3. Show output with:
    - customer_id, customer_name, city, currency, total_order_value_INR, total_revenue_loss_INR, %_revenue_loss
4. Sort customers by total_revenue_loss_INR descending

2. Identify customers with the most significant On-Time, In Full (OTIF) discrepancies

Prompt Title: Customers with OTIF Gaps

Using fact_order_line and dim_target_orders, generate a table of customers with largest OTIF discrepancies:

1. Calculate actual_otif% = % of order lines where On Time In Full = 1
2. Join with dim_target_orders on customer_id
3. Calculate otif_gap% = actual_otif% - otif_target%
4. Include: customer_id, customer_name, city, actual_otif%, otif_target%, otif_gap%
5. Sort by otif_gap% ascending (most negative gaps first)
6. Show Top 10 customers

3. Determine product categories that exhibit low 'In Full' delivery rates.

Prompt :

Using fact_order_line and dim_product, generate a table showing product categories with lowest In Full delivery rates:

1. Join fact_order_line with dim_product on product_id
2. Group by category
3. For each category, calculate:
    a. total_order_qty = sum(order_qty)

b. total_delivery_qty = sum(delivery_qty)

c. in_full% = total_delivery_qty / total_order_qty * 100

4. Sort by in_full% ascending

4. Calculate the average delay time for late deliveries.

Prompt Title: Average Delay in Late Deliveries

Using fact_order_line (and dim_product for category), generate delay analysis:

1. Filter rows where actual_delivery_date > agreed_delivery_date
2. Create a new column:

   delay_days = actual_delivery_date - agreed_delivery_date (in days)

Calculate:

average_delay_days overall

average_delay_days grouped by:

customer_id, customer_name (join with dim_customer)

category (join with dim_product)

Also include count of late_orders

Display:

Table 1: overall average delay

Table 2: customer-wise delays

Table 3: category-wise delays

5. Identify product categories with the lowest 'In Full' delivery rates. What could this indicate about supply chain bottlenecks?

Prompt :

1. Look at product delivery data from:
   - Orders table
   - Product table
   - Date table
2. Find the 5 product categories with the lowest delivery success (based on how much was delivered vs ordered).
3. For each week and category, calculate:

- Delivery success percentage (in-full %)
- Total quantity ordered
- Number of orders
- Average quantity per order

4. Add an insight label for each week:
   - If delivery % is below 60% and order volume is high → label as “Potential Supply Issue”
   - If delivery % changes a lot from one week to the next → label as “Unstable Fulfillment”
   - If none of the above → label as “Stable”

4. Business Question: Exercise 3

1. Creating Business KPIs :

Prompt :

1. In-Full %
2. On-Time Delivery %
3. OTIF %
4. Order Cycle Time (days)
5. Average Delivery Delay (days)
6. Lead Time Variability
7. Category Demand Variability
8. Total Backorder Quantity

1. Which product categories are facing the highest fulfilment and delivery challenges?

Prompt :

"Filter the most recent 4 weeks. Aggregate by category: total backorder_qty, average in_full_percent, and average delivery_delay_days. Return the five categories with (a) the highest backorder_qty, (b) in_full_percent < 90, and (c) avg delivery_delay_days > 2. Include counts of total orders and suggest probable root causes."

2. Are any customers consistently missing OTIF targets?

Prompt:

"For each customer_id, compute weekly OTIF % (on_time_flag = 1 AND in_full_percent = 100). Identify customers whose 4-week rolling average OTIF % < 85. Return customer name, city, weekly OTIF %, and variance from 85 % threshold."

3. Where is lead-time variability threatening service levels?

Prompt :

"Group by category and calculate lead_time_variability (stddev of order_cycle_time_days) for the past quarter. Flag categories where variability > 3 days. Show avg order_cycle_time_days alongside variability."

4. Which weeks experienced the worst demand swings and did that impact service?

Prompt :

“Aggregate order_qty by week_of_year and category. Compute week-over-week % change. Highlight weeks with demand increase > 30 % or decrease < -20 %. For those weeks, show corresponding in_full_percent and on_time_delivery %.”

5. Do long delivery delays correlate with larger backorders?

Prompt :

“Using last 8 weeks of data, build a scatter of avg_delivery_delay_days vs. avg_backorder_qty at the customer–category level. Return points in the upper-right quadrant (delay > 3 days AND backorder_qty > 100 units) and list them in a table.”

6. Which customers are improving—or worsening—in order cycle time?

Prompt :

“Calculate 3-week rolling average of order_cycle_time_days per customer. Compute the slope of the trend line. List top 5 customers with the steepest positive slope (worsening) and top 5 with steepest negative slope (improving).”

7. Are metro and non-metro customers served differently?

Prompt :

“Tag each city as ‘Metro’ or ‘Non-Metro’. For the past two months, compare Metro vs. Non-Metro on (a) avg in_full_percent, (b) avg on_time_delivery %, (c) avg order_cycle_time_days, and (d) total backorder_qty. Highlight the metric with the largest performance gap.”