

## E-Commerce Category Performance Analysis (YOY 2024 vs 2023) (End-to-End SQL + Excel Weekly Commercial Performance Reporting Project)

### Executive Summary

This project replicates a real-world weekly **Category Performance Reporting System** that I owned and delivered in my previous role as a **Junior Data Analyst**. The purpose of the report was to provide commercial leadership with weekly visibility into sales performance, profitability, contribution trends and category movement across major cities in India.

The analysis evaluates weekly performance across **35 cities** and **13 product categories** over **2023 vs 2024**, helping identify growth opportunities, category momentum, and margin performance essential for planning and inventory decisions.

This project demonstrates my ability to translate raw transactional data into strategic insights and build reporting systems that support high-impact commercial decision-making.

### Business Problem

Retail businesses operate in highly competitive environments where assortment, pricing strategy and regional performance insights are critical. Leadership teams require weekly sales and profitability tracking to:

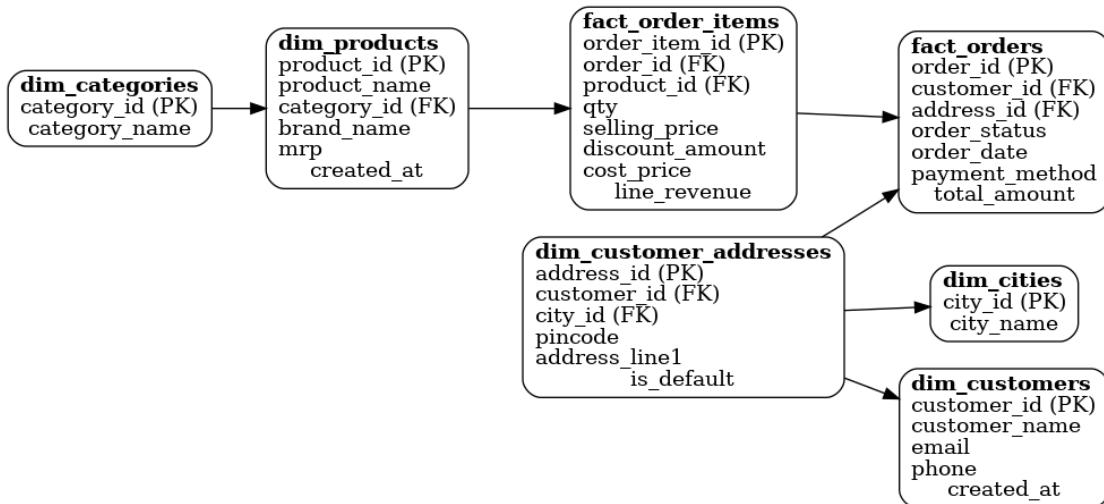
- Identify top-performing and underperforming product categories
- Prioritize focus markets and high-opportunity growth areas
- Guide commercial decisions such as pricing, range expansion, and promotions
- Support revenue forecasting and financial planning

In my previous role, I delivered this weekly performance report every **Monday morning**, based on SQL extracted transactional data and summarized through Excel dashboards for senior leadership review.

## Data Architecture & ERD

The project follows a **star-schema** design similar to real ecommerce database modeling.

Table	Description
dim_categories	Product category reference
dim_cities	City master
dim_products	Product details & category mapping
dim_customers	Customer profile
dim_customer_addresses	Address mapping to city
fact_orders	Order-level commercial information
fact_order_items	Transactional item-level details



To maintain confidentiality and avoid using any proprietary information, I created a synthetic dataset that mimics realistic ecommerce sales behavior, volume variation and seasonality.

## SQL Transformation & Metrics

### 1. Base CTE: order\_items\_enriched

The purpose of this CTE is to prepare a **clean, analysis-ready dataset** by joining all required fact and dimension tables into a single view.

In this stage, I:

- Begin with the transactional fact table **fact\_order\_items**, which contains one row per product per order.
- Join supporting dimensional tables to enrich the dataset:
  - **dim\_products** → adds product\_name and category\_id
  - **dim\_categories** → adds category\_name
  - **fact\_orders** → provides order\_date to enable time-based aggregation
  - **dim\_customer\_addresses** → links each order to a location
  - **dim\_cities** → adds the final city\_name

### Key calculated fields

Field	Logic	Purpose
actual_sell_price	selling_price * qty – discount_amount	Net revenue after discount
costP	cost_price	Cost associated with that line item
week, year, order_date	extracted using DATEPART()	Enables weekly and yearly trend reporting

This CTE produces a **rich joined dataset** that contains:

- Category
- City
- Order timing (week and year)
- Revenue & cost at the order-item level

It serves as the **foundation** for downstream aggregation, dashboarding and YOY comparison work.

```

WITH order_items_enriched AS (
    SELECT
        fo.order_item_id,
        fo.order_id,
        fo.product_id,
        fo.qty,
        fo.selling_price,
        fo.discount_amount,
        (fo.selling_price * fo.qty - fo.discount_amount) AS actual_sell_price,
        fo.cost_price AS costP,
        DATEPART(WEEK, fo.order_date) AS week,
        DATEPART(YEAR, fo.order_date) AS year,
        dc.category_name,
        dc.address_id,
        dc.city_name
    FROM fact_order_items fo
    LEFT JOIN dim_products dp
        ON fo.product_id = dp.product_id
    LEFT JOIN dim_categories dc
        ON dp.category_id = dc.category_id
    LEFT JOIN dim_customer_addresses da
        ON fo.order_id = da.order_id
    LEFT JOIN dim_cities dmc
        ON da.address_id = dmc.address_id
    LEFT JOIN dim_cities dmc
        ON da.city_id = dmc.city_id
    WHERE DATEPART(WEEK, fo.order_date) = 2024
)

```

## 2. Weekly Aggregated Performance Summary

Based on the unified CTE dataset, the final query performs an aggregation at **Category × City × Week** level.

In this stage, I:

- **Group by**
  - category\_name
  - city\_name
  - Week
- **Aggregate key performance metrics**

Metric	Logic	Purpose
SUM(qty)	total_qty	Total units sold
SUM(actual_sell_price)	sale_nett	Total revenue after discount
SUM(costP)	cost_nett	Total cost of goods sold
SUM(selling_price - costP)	profit	Profit contribution

The result of this query is a **weekly performance table** with the below output

```

-- Weekly aggregated performance summary
SELECT
    category_name,
    city_name,
    week,
    SUM(qty) AS total_qty,
    SUM(actual_sell_price) AS sale_nett,
    SUM(costP) AS cost_nett,
    SUM(selling_price - costP) AS profit
FROM order_items_enriched
GROUP BY
    category_name,
    city_name,
    week
ORDER BY
    week,
    category_name,
    city_name;

```

category_name	city_name	week	total_qty	sale_nett	cost_nett	profit
Automotive & Industrial Tools	Ahmedabad	1	4	4690.3200000000	790.5200000000	476.1800000000
Automotive & Industrial Tools	Anritsar	1	7	61115.5300000000	15256.9800000000	8258.8200000000
Automotive & Industrial Tools	Bangalore	1	8	55084.2900000000	26915.8900000000	12482.8000000000
Automotive & Industrial Tools	Bhopal	1	9	53016.8800000000	22883.2500000000	8641.3000000000
Automotive & Industrial Tools	Bhubaneshwar	1	4	26826.2100000000	17579.8400000000	9939.8100000000
Automotive & Industrial Tools	Chandigarh	1	9	129282.0000000000	30583.5000000000	15792.7400000000
Automotive & Industrial Tools	Chennai	1	36	364867.6000000000	88685.4000000000	41924.8400000000
Automotive & Industrial Tools	Coimbatore	1	17	121663.0200000000	43097.6200000000	16923.3300000000
Automotive & Industrial Tools	Dehradun	1	5	27765.2000000000	9106.9100000000	5615.2800000000
Automotive & Industrial Tools	Delhi	1	18	164658.4600000000	33935.8700000000	14972.9600000000
Automotive & Industrial Tools	Gurgaon	1	16	157465.2700000000	27333.2500000000	13285.7300000000
Automotive & Industrial Tools	Guwahati	1	14	162667.7400000000	22586.4400000000	12396.1000000000

This is the exact structure I export to Excel to do my analysis.

## Dashboard Output & Analysis Performed

The main goal of the project was to conduct a detailed, multi-dimensional performance review to identify key trends, top performers, and areas for improvement.

- **Year-over-Year (YOY) Performance Analysis:** The core objective is to compare sales and profit performance metrics (**Quantity, Net Revenue, Net Cost, and Profit**) between 2024 and 2023.
- **Performance by Category:** The report classifies and ranks products by **Category** to determine which groups are driving the most value, notably via an **ABC Analysis** (A, B, C classifications).
- **Time-Series Trend Analysis:** Performance is tracked on a **weekly** basis to identify short-term trends, seasonality, or the impact of specific events across the entire business and within specific categories.
- **Geographical Performance Review:** The data is broken down by **City** and aggregated into **Areas/Regions** (e.g., North, South, East, West) to evaluate regional success and compare different markets.

## Business Insights (from the excel report)

Insight	Business Impact
<b>Mobiles &amp; Tablets and Electronics contribute ~80% of revenue</b>	Commercial dependency risk and major strategic focus area
<b>Bangalore is the top performing city with highest profit contribution</b>	Prioritize this market for promotions & premium assortment
<b>Footwear and Beauty categories declining YOY</b>	Opportunity for pricing or assortment optimization
<b>ABC analysis shows 3 categories drive 85% of sales</b>	Inventory prioritization and marketing allocation

---