Fitness Product Sales Analysis and Forecast

Leila Zhumabaevna Mamatalieva
Kwang Sub Ahn (Mike)
Ngoc Phuoc Nguyen (Patrick)

Objective

To analyze the performance of fitness products across Amazon and Walmart using **transactional data, promotional event impacts, and inventory status.**

The project will focus on **sales trends, promotional effectiveness, and inventory forecasting** to improve decision-making for future campaigns and stock management.

Key Questions

1. **Sales Contribution by Category**

   How much does each product category contribute to total sales?

2. **Sales Trends Over Time**

   What is the month-over-month growth rate for each category?

   How has each category performed over time?

3. **Channel Effectiveness**

   How do different advertising (media) and promotion channels impact sales?

4. **Top 10 Best-Selling Subcategories**

   Which subcategories have the highest number of orders?

5. **Product Launches Overview**

   How many products have we launched so far?

Dataset Overview: Key Metrics

- **RS (Real Sales):** The number of units sold to customers (= shipped_units).
- **Promotion:** Discount costs on the product's retail price (= coupon + vc_promo + vm_promo).
- **Ads:** Advertising costs for search campaigns on the platform (= sbv + sp + sd + sb + dsp).
- **Frozen Time:** A product requires 3 months to complete; therefore, "frozen time" is the period during which no additional inventory can be ordered (3 months from the current month).
- **Levels of Product:** Category (Product Type) >> Subcategory (Product Group) >> SKU (Product Code).
- **Holiday Month:** High-sales months on the platform.
- **GMV (Gross Merchandise Value):** Revenue from the product (= price * real sales).
- **Ordered Date:** The date when the customer placed the order on the app.
- **Shipped Date:** The date when the customer received the product.
- **Incoming:** Expected stock arrivals.
- **Spend:** Expenditure.

- **Data Source**

  YES4ALL: SALES SPORTS & OUTDOOR DEPARTMENT

- **Key Tables**

transactions: sales, ad performance, transactional records

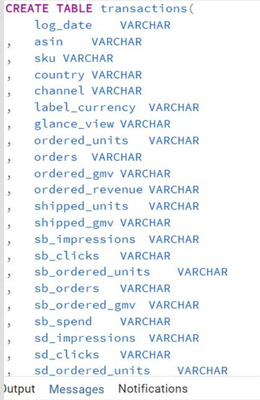
product_list: Yes4All product catalog

Inventory: Stock levels, incoming inventory schedule

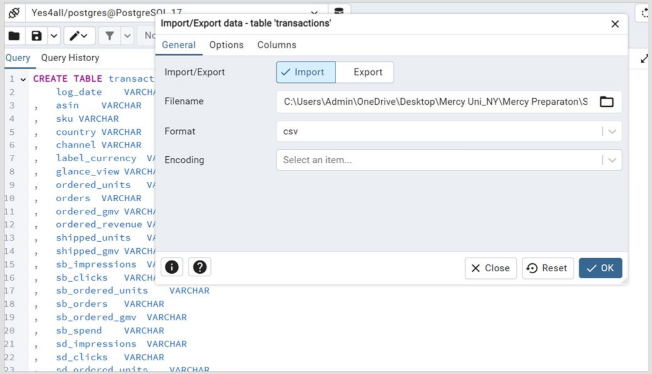
Projection: Sales forecast, cost estimation

Even_list: Sales event schedule

Databases:



```
CREATE TABLE transactions(
    log_date    VARCHAR
,   asin    VARCHAR
,   sku VARCHAR
,   country VARCHAR
,   channel VARCHAR
,   label_currency  VARCHAR
,   glance_view VARCHAR
,   ordered_units   VARCHAR
,   orders  VARCHAR
,   ordered_gmv VARCHAR
,   ordered_revenue VARCHAR
,   shipped_units   VARCHAR
,   shipped_gmv VARCHAR
,   sb_impressions  VARCHAR
,   sb_clicks   VARCHAR
,   sb_ordered_units    VARCHAR
,   sb_orders   VARCHAR
,   sb_ordered_gmv  VARCHAR
,   sb_spend    VARCHAR
,   sd_impressions  VARCHAR
,   sd_clicks   VARCHAR
,   sd_ordered_units    VARCHAR
```



```
CREATE TABLE transact
    log_date    VARCH
,   asin    VARCHAR
,   sku VARCHAR
,   country VARCHAR
,   channel VARCHAR
,   label_currency  V
,   glance_view VARCH
,   ordered_units   V
,   orders  VARCHAR
,   ordered_gmv VARCH
,   ordered_revenue V
,   shipped_units   V
,   shipped_gmv VARCH
,   sb_impressions  V
,   sb_clicks   VARCH
,   sb_ordered_units    VARCHAR
,   sb_orders   VARCHAR
,   sb_ordered_gmv  VARCHAR
,   sb_spend    VARCHAR
,   sd_impressions  VARCHAR
,   sd_clicks   VARCHAR
,   sd_ordered_units    VARCHAR
```

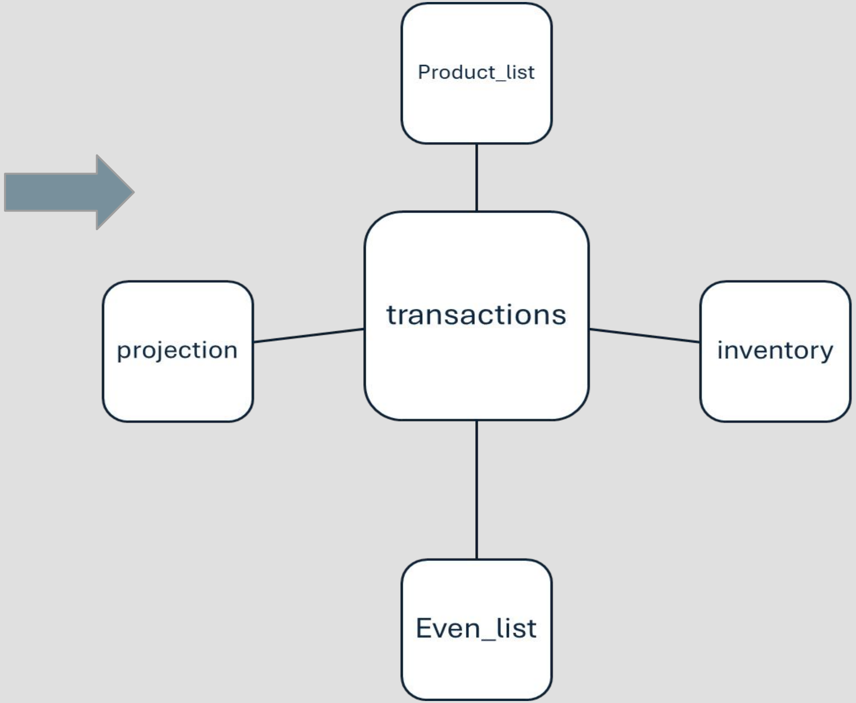
Import/Export data - table 'transactions'

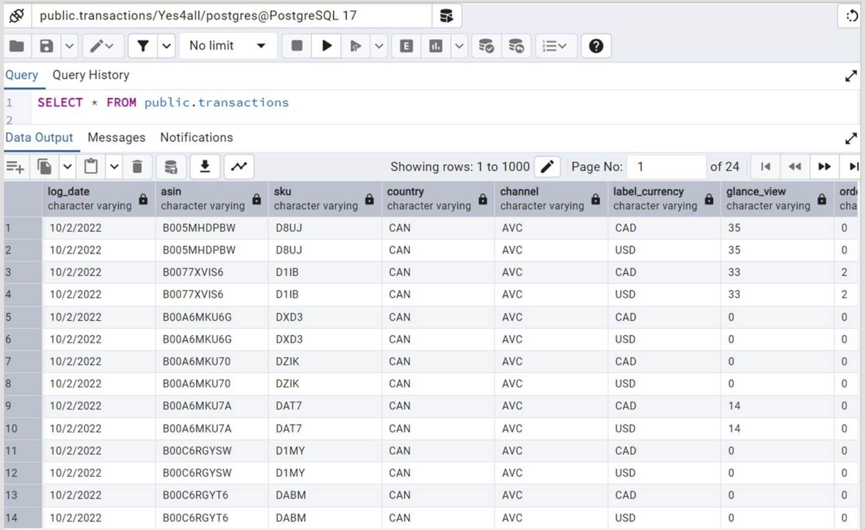
Import/Export: ☑ Import | Export

Filename: C:\Users\Admin\OneDrive\Desktop\Mercy Uni_NY\Mercy Preparaton\S

Format: csv

Encoding: Select an item...



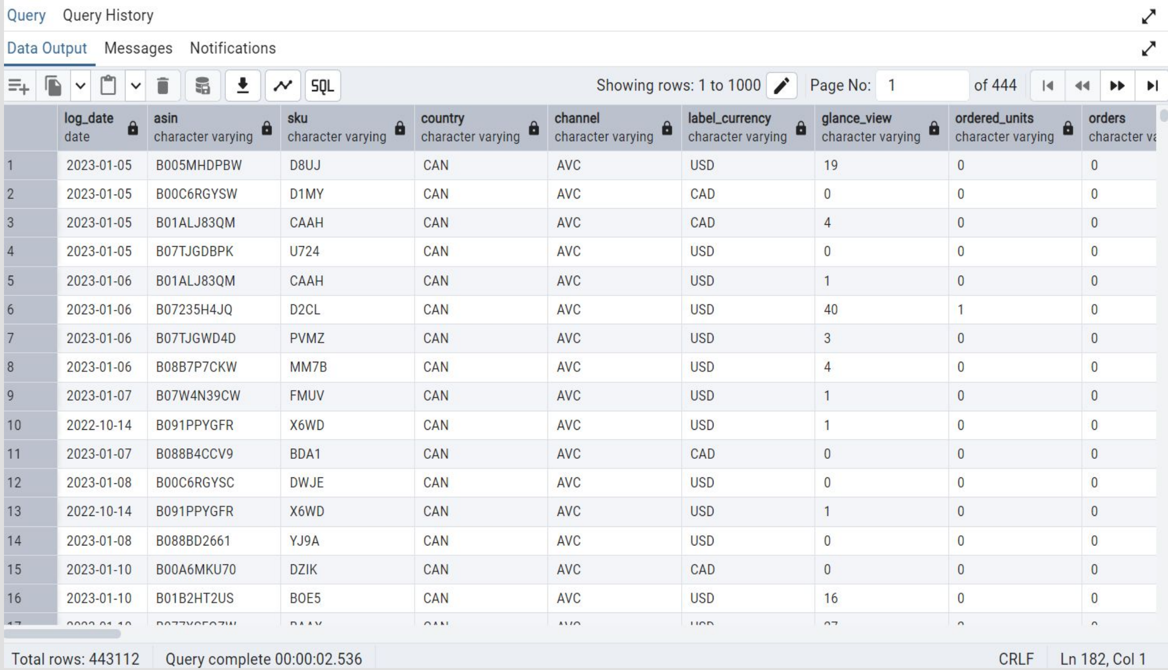


```
SELECT * FROM public.transactions
```

Showing rows: 1 to 1000 Page No: 1 of 24

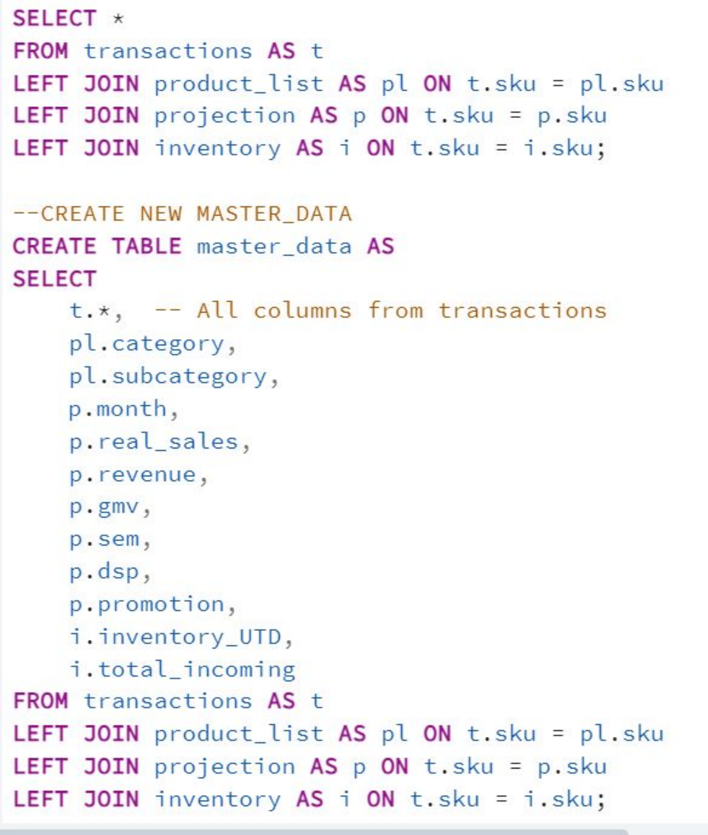
	log_date character varying	asin character varying	sku character varying	country character varying	channel character varying	label_currency character varying	glance_view character varying	ord cha
1	10/2/2022	B005MHDPBW	D8UJ	CAN	AVC	CAD	35	0
2	10/2/2022	B005MHDPBW	D8UJ	CAN	AVC	USD	35	0
3	10/2/2022	B0077XVIS6	D1IB	CAN	AVC	CAD	33	2
4	10/2/2022	B0077XVIS6	D1IB	CAN	AVC	USD	33	2
5	10/2/2022	B00A6MKU6G	DXD3	CAN	AVC	CAD	0	0
6	10/2/2022	B00A6MKU6G	DXD3	CAN	AVC	USD	0	0
7	10/2/2022	B00A6MKU70	DZIK	CAN	AVC	CAD	0	0
8	10/2/2022	B00A6MKU70	DZIK	CAN	AVC	USD	0	0
9	10/2/2022	B00A6MKU7A	DAT7	CAN	AVC	CAD	14	0
10	10/2/2022	B00A6MKU7A	DAT7	CAN	AVC	USD	14	0
11	10/2/2022	B00C6RGYSW	D1MY	CAN	AVC	CAD	0	0
12	10/2/2022	B00C6RGYSW	D1MY	CAN	AVC	USD	0	0
13	10/2/2022	B00C6RGYT6	DABM	CAN	AVC	CAD	0	0
14	10/2/2022	B00C6RGYT6	DABM	CAN	AVC	USD	0	0

Compile all necessary attribute to one 1 master table by join



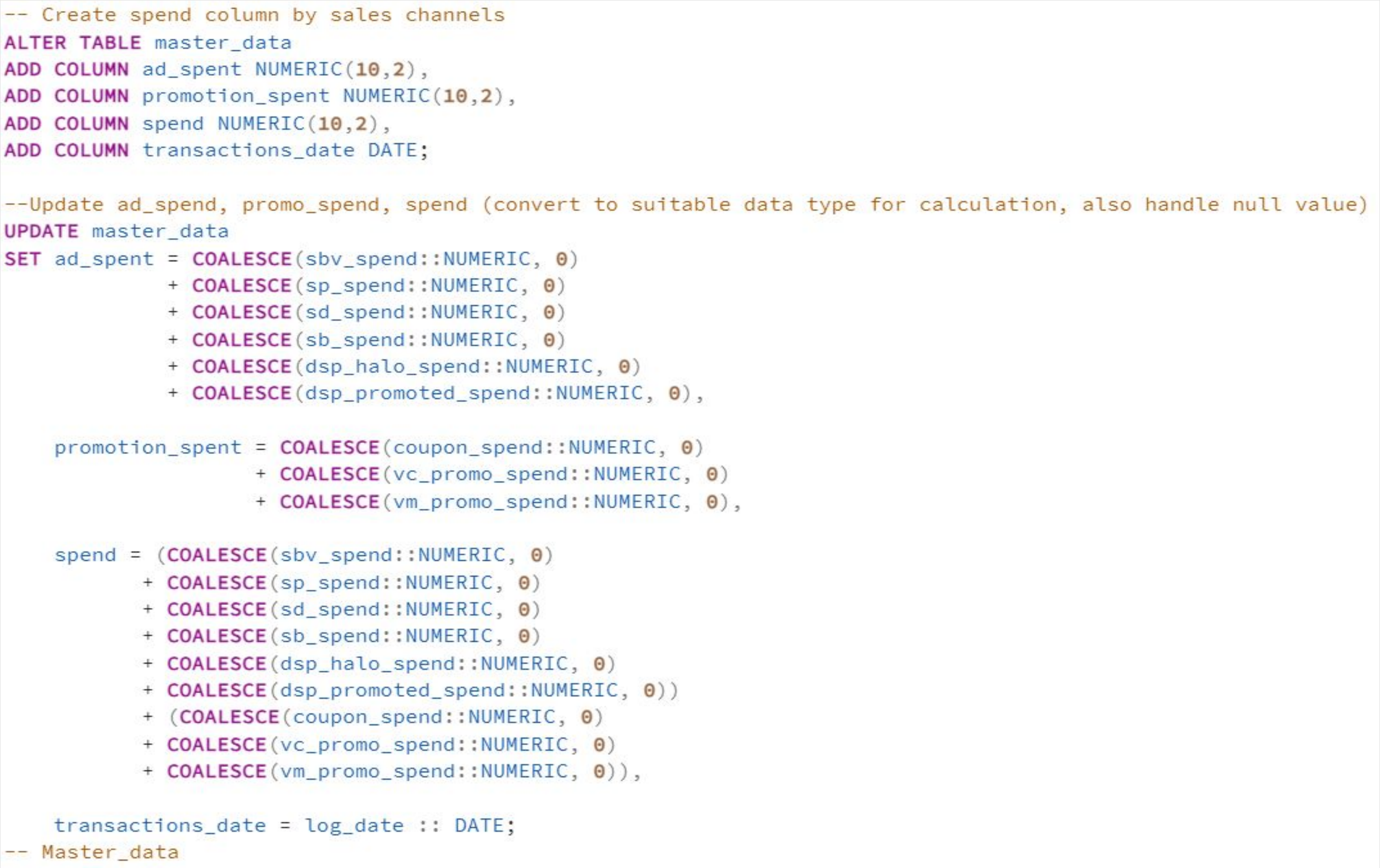
	log_date	asin	sku	country	channel	label_currency	glance_view	ordered_units	orders
	date	character varying	character varying	character varying	character varying	character varying	character varying	character varying	character va
1	2023-01-05	B005MHDPBW	D8UJ	CAN	AVC	USD	19	0	0
2	2023-01-05	B00C6RGYSW	D1MY	CAN	AVC	CAD	0	0	0
3	2023-01-05	B01ALJ83QM	CAAH	CAN	AVC	CAD	4	0	0
4	2023-01-05	B07TJGDBPK	U724	CAN	AVC	USD	0	0	0
5	2023-01-06	B01ALJ83QM	CAAH	CAN	AVC	USD	1	0	0
6	2023-01-06	B07235H4JQ	D2CL	CAN	AVC	USD	40	1	0
7	2023-01-06	B07TJGWD4D	PVMZ	CAN	AVC	USD	3	0	0
8	2023-01-06	B08B7P7CKW	MM7B	CAN	AVC	USD	4	0	0
9	2023-01-07	B07W4N39CW	FMUV	CAN	AVC	USD	1	0	0
10	2022-10-14	B091PPYGFR	X6WD	CAN	AVC	USD	1	0	0
11	2023-01-07	B088B4CCV9	BDA1	CAN	AVC	CAD	0	0	0
12	2023-01-08	B00C6RGYSC	DWJE	CAN	AVC	USD	0	0	0
13	2022-10-14	B091PPYGFR	X6WD	CAN	AVC	USD	1	0	0
14	2023-01-08	B088BD2661	YJ9A	CAN	AVC	USD	0	0	0
15	2023-01-10	B00A6MKU70	DZIK	CAN	AVC	CAD	0	0	0
16	2023-01-10	B01B2HT2US	BOE5	CAN	AVC	USD	16	0	0

Total rows: 443112 Query complete 00:00:02.536 CRLF Ln 182, Col 1



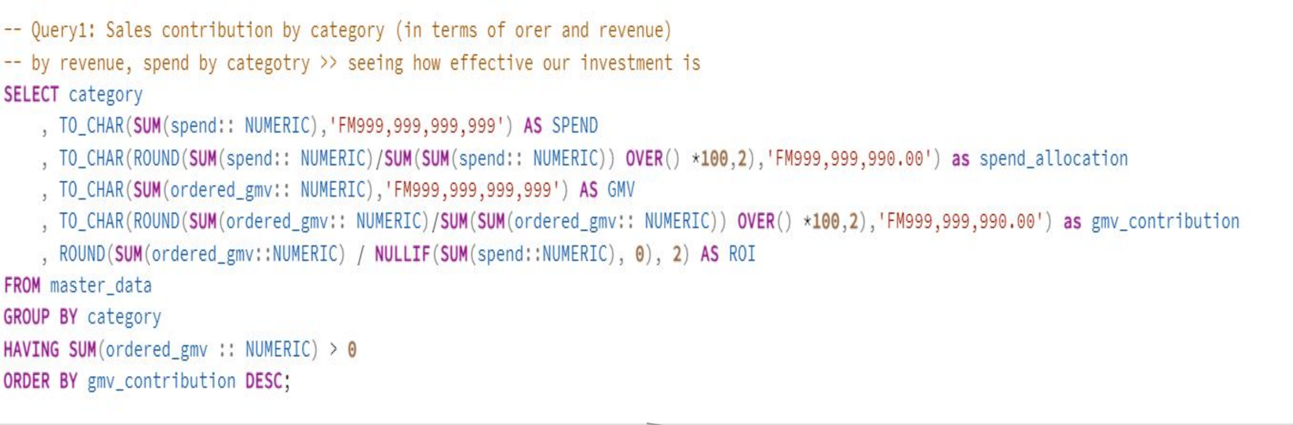
```
SELECT *
FROM transactions AS t
LEFT JOIN product_list AS pl ON t.sku = pl.sku
LEFT JOIN projection AS p ON t.sku = p.sku
LEFT JOIN inventory AS i ON t.sku = i.sku;

--CREATE NEW MASTER_DATA
CREATE TABLE master_data AS
SELECT
    t.*,  -- All columns from transactions
    pl.category,
    pl.subcategory,
    p.month,
    p.real_sales,
    p.revenue,
    p.gmv,
    p.sem,
    p.dsp,
    p.promotion,
    i.inventory_UTD,
    i.total_incoming
FROM transactions AS t
LEFT JOIN product_list AS pl ON t.sku = pl.sku
LEFT JOIN projection AS p ON t.sku = p.sku
LEFT JOIN inventory AS i ON t.sku = i.sku;
```

Feature engineering for better observation

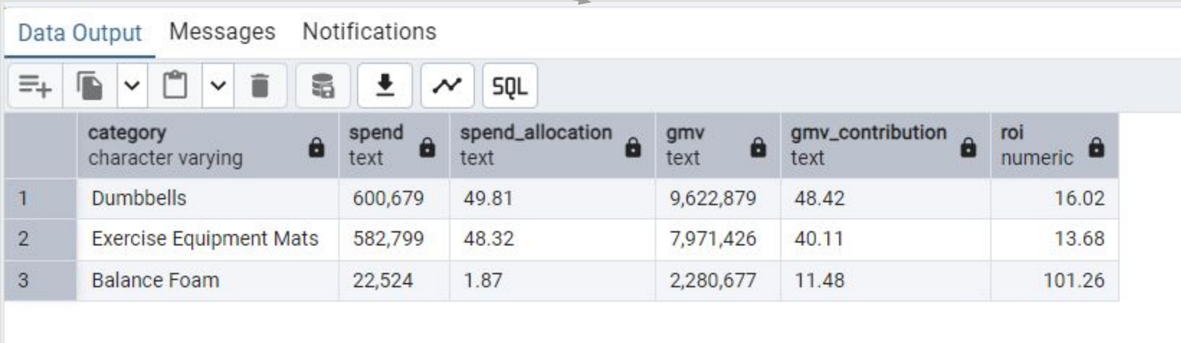


```
-- Create spend column by sales channels
ALTER TABLE master_data
ADD COLUMN ad_spent NUMERIC(10,2),
ADD COLUMN promotion_spent NUMERIC(10,2),
ADD COLUMN spend NUMERIC(10,2),
ADD COLUMN transactions_date DATE;

--Update ad_spend, promo_spend, spend (convert to suitable data type for calculation, also handle null value)
UPDATE master_data
SET ad_spent = COALESCE(sbv_spend::NUMERIC, 0)
             + COALESCE(sp_spend::NUMERIC, 0)
             + COALESCE(sd_spend::NUMERIC, 0)
             + COALESCE(sb_spend::NUMERIC, 0)
             + COALESCE(dsp_halo_spend::NUMERIC, 0)
             + COALESCE(dsp_promoted_spend::NUMERIC, 0),

    promotion_spent = COALESCE(coupon_spend::NUMERIC, 0)
                    + COALESCE(vc_promo_spend::NUMERIC, 0)
                    + COALESCE(vm_promo_spend::NUMERIC, 0),

    spend = (COALESCE(sbv_spend::NUMERIC, 0)
           + COALESCE(sp_spend::NUMERIC, 0)
           + COALESCE(sd_spend::NUMERIC, 0)
           + COALESCE(sb_spend::NUMERIC, 0)
           + COALESCE(dsp_halo_spend::NUMERIC, 0)
           + COALESCE(dsp_promoted_spend::NUMERIC, 0))
           + (COALESCE(coupon_spend::NUMERIC, 0)
           + COALESCE(vc_promo_spend::NUMERIC, 0)
           + COALESCE(vm_promo_spend::NUMERIC, 0)),

    transactions_date = log_date :: DATE;
-- Master_data
```

Question 1: How much does each product category contribute to total sales?



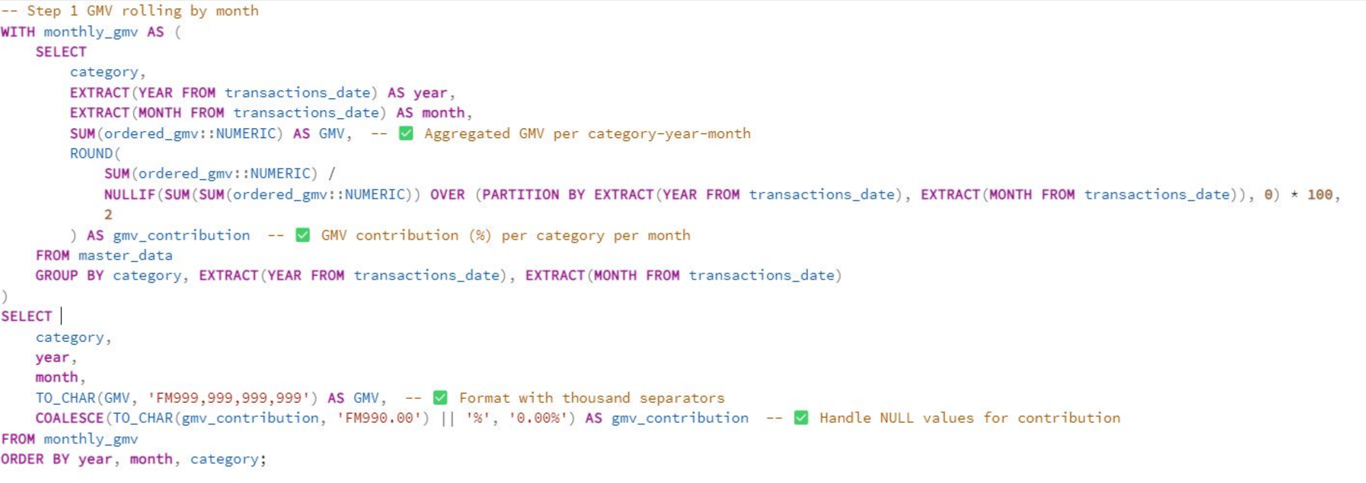
```
-- Query1: Sales contribution by category (in terms of orer and revenue)
-- by revenue, spend by categotry >> seeing how effective our investment is
SELECT category
    , TO_CHAR(SUM(spend:: NUMERIC),'FM999,999,999,999') AS SPEND
    , TO_CHAR(ROUND(SUM(spend:: NUMERIC)/SUM(SUM(spend:: NUMERIC)) OVER() *100,2),'FM999,999,990.00') as spend_allocation
    , TO_CHAR(SUM(ordered_gmv:: NUMERIC),'FM999,999,999,999') AS GMV
    , TO_CHAR(ROUND(SUM(ordered_gmv:: NUMERIC)/SUM(SUM(ordered_gmv:: NUMERIC)) OVER() *100,2),'FM999,999,990.00') as gmv_contribution
    , ROUND(SUM(ordered_gmv::NUMERIC) / NULLIF(SUM(spend::NUMERIC), 0), 2) AS ROI
FROM master_data
GROUP BY category
HAVING SUM(ordered_gmv :: NUMERIC) > 0
ORDER BY gmv_contribution DESC;
```



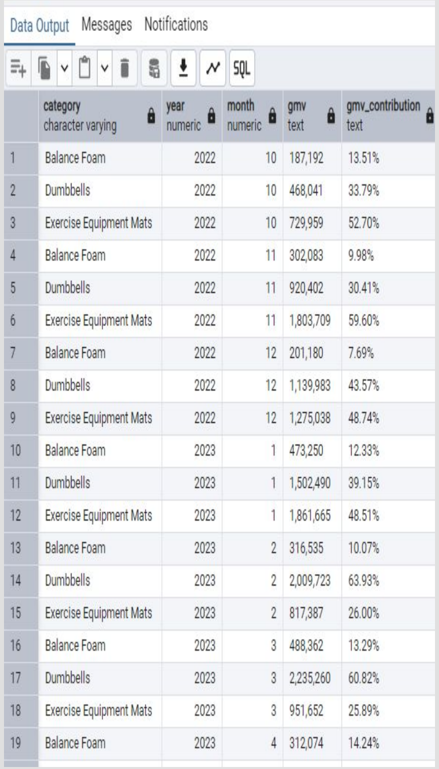
	category character varying	spend text	spend_allocation text	gmv text	gmv_contribution text	roi numeric
1	Dumbbells	600,679	49.81	9,622,879	48.42	16.02
2	Exercise Equipment Mats	582,799	48.32	7,971,426	40.11	13.68
3	Balance Foam	22,524	1.87	2,280,677	11.48	101.26

Question 2: Sales Trends Over Time

Calculate the month-over-month, contribution, growth for each product category & analyze the performance trends for each category over time.



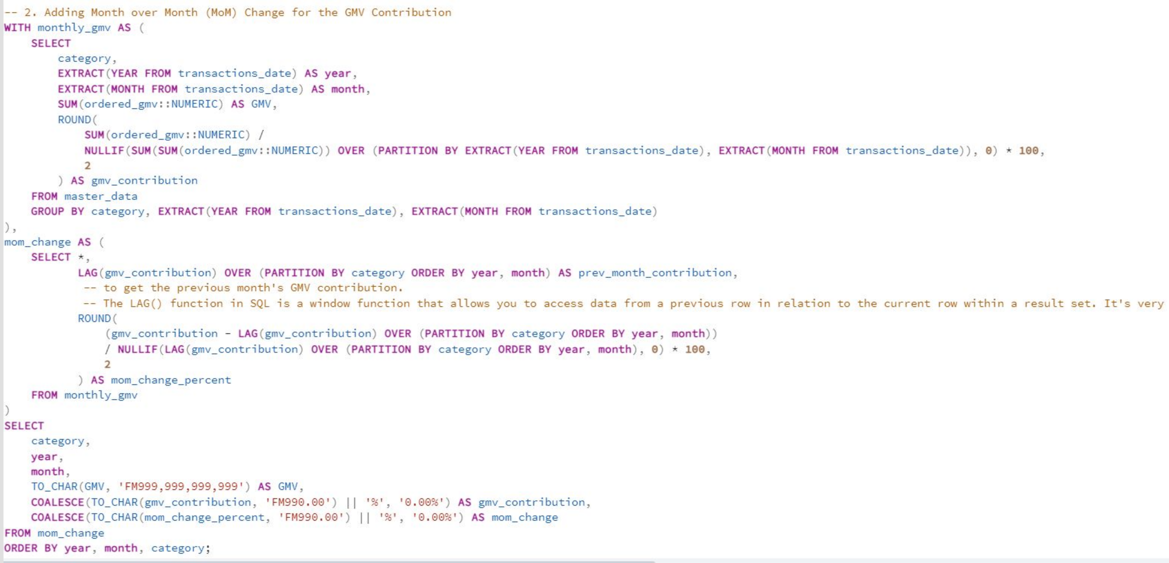
```sql
-- Step 1 GMV rolling by month
WITH monthly_gmv AS (
    SELECT
        category,
        EXTRACT(YEAR FROM transactions_date) AS year,
        EXTRACT(MONTH FROM transactions_date) AS month,
        SUM(ordered_gmv::NUMERIC) AS GMV,  -- ✅ Aggregated GMV per category-year-month
        ROUND(
            SUM(ordered_gmv::NUMERIC) /
            NULLIF(SUM(SUM(ordered_gmv::NUMERIC)) OVER (PARTITION BY EXTRACT(YEAR FROM transactions_date), EXTRACT(MONTH FROM transactions_date)), 0) * 100,
            2
        ) AS gmv_contribution  -- ✅ GMV contribution (%) per category per month
    FROM master_data
    GROUP BY category, EXTRACT(YEAR FROM transactions_date), EXTRACT(MONTH FROM transactions_date)
)
SELECT
    category,
    year,
    month,
    TO_CHAR(GMV, 'FM999,999,999,999') AS GMV,  -- ✅ Format with thousand separators
    COALESCE(TO_CHAR(gmv_contribution, 'FM990.00') || '%', '0.00%') AS gmv_contribution  -- ✅ Handle NULL values for contribution
FROM monthly_gmv
ORDER BY year, month, category;
```



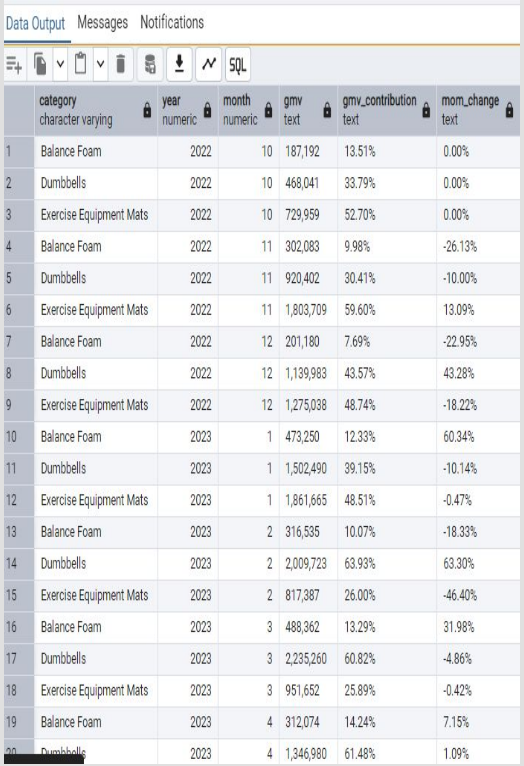
	category character varying	year numeric	month numeric	gmv text	gmv_contribution text
1	Balance Foam	2022	10	187,192	13.51%
2	Dumbbells	2022	10	468,041	33.79%
3	Exercise Equipment Mats	2022	10	729,959	52.70%
4	Balance Foam	2022	11	302,083	9.98%
5	Dumbbells	2022	11	920,402	30.41%
6	Exercise Equipment Mats	2022	11	1,803,709	59.60%
7	Balance Foam	2022	12	201,180	7.69%
8	Dumbbells	2022	12	1,139,983	43.57%
9	Exercise Equipment Mats	2022	12	1,275,038	48.74%
10	Balance Foam	2023	1	473,250	12.33%
11	Dumbbells	2023	1	1,502,490	39.15%
12	Exercise Equipment Mats	2023	1	1,861,665	48.51%
13	Balance Foam	2023	2	316,535	10.07%
14	Dumbbells	2023	2	2,009,723	63.93%
15	Exercise Equipment Mats	2023	2	817,387	26.00%
16	Balance Foam	2023	3	488,362	13.29%
17	Dumbbells	2023	3	2,235,260	60.82%
18	Exercise Equipment Mats	2023	3	951,652	25.89%
19	Balance Foam	2023	4	312,074	14.24%

Question 2: Sales Trends Over Time

Calculate the month-over-month, contribution, growth for each product category & analyze the performance trends for each category over time.

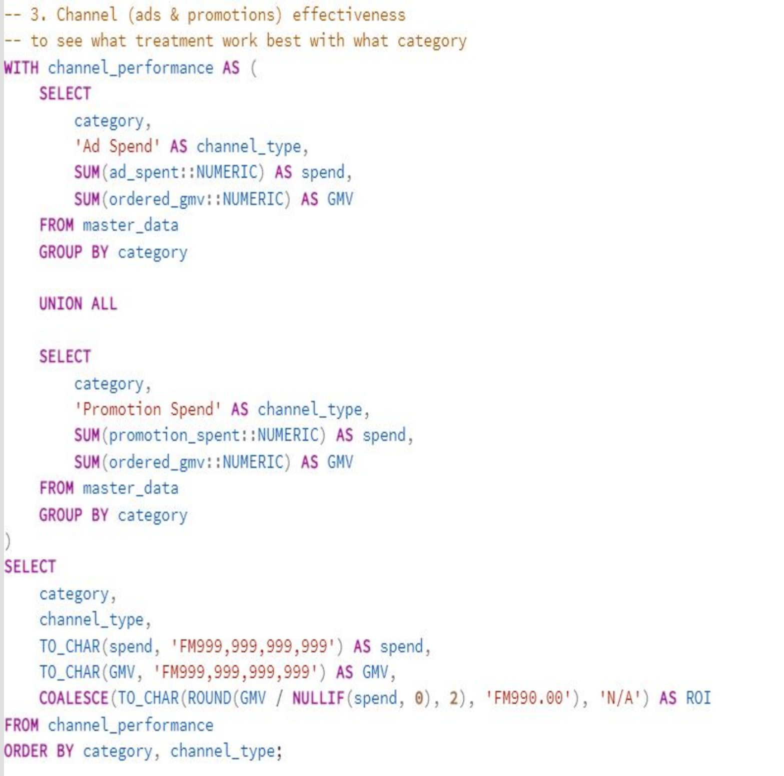


```sql
-- 2. Adding Month over Month (MoM) Change for the GMV Contribution
WITH monthly_gmv AS (
    SELECT
        category,
        EXTRACT(YEAR FROM transactions_date) AS year,
        EXTRACT(MONTH FROM transactions_date) AS month,
        SUM(ordered_gmv::NUMERIC) AS GMV,
        ROUND(
            SUM(ordered_gmv::NUMERIC) /
            NULLIF(SUM(SUM(ordered_gmv::NUMERIC)) OVER (PARTITION BY EXTRACT(YEAR FROM transactions_date), EXTRACT(MONTH FROM transactions_date)), 0) * 100,
            2
        ) AS gmv_contribution
    FROM master_data
    GROUP BY category, EXTRACT(YEAR FROM transactions_date), EXTRACT(MONTH FROM transactions_date)
),
mom_change AS (
    SELECT *,
           LAG(gmv_contribution) OVER (PARTITION BY category ORDER BY year, month) AS prev_month_contribution,
            -- to get the previous month's GMV contribution.
            -- The LAG() function in SQL is a window function that allows you to access data from a previous row in relation to the current row within a result set. It's very
           ROUND(
               (gmv_contribution - LAG(gmv_contribution) OVER (PARTITION BY category ORDER BY year, month))
               / NULLIF(LAG(gmv_contribution) OVER (PARTITION BY category ORDER BY year, month), 0) * 100,
               2
           ) AS mom_change_percent
    FROM monthly_gmv
)
SELECT
    category,
    year,
    month,
    TO_CHAR(GMV, 'FM999,999,999,999') AS GMV,
    COALESCE(TO_CHAR(gmv_contribution, 'FM990.00') || '%', '0.00%') AS gmv_contribution,
    COALESCE(TO_CHAR(mom_change_percent, 'FM990.00') || '%', '0.00%') AS mom_change
FROM mom_change
ORDER BY year, month, category;
```

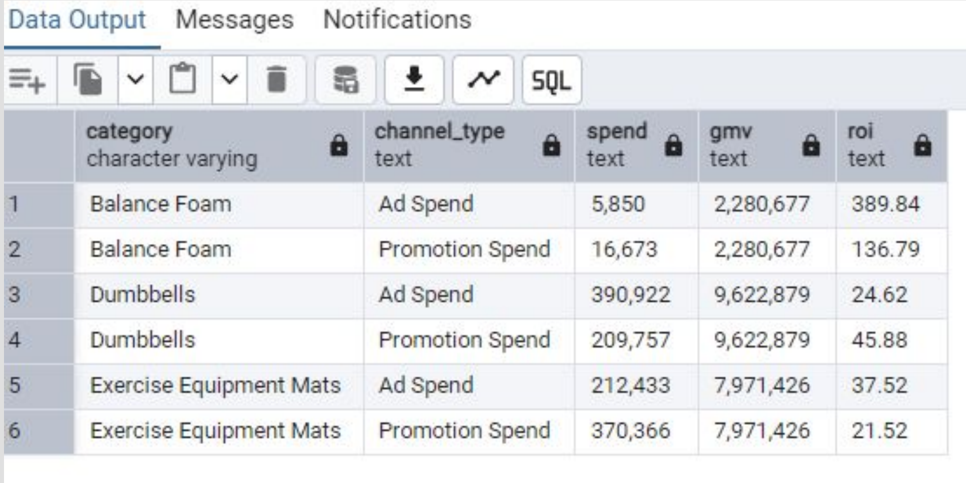


	category character varying	year numeric	month numeric	gmv text	gmv_contribution text	mom_change text
1	Balance Foam	2022	10	187,192	13.51%	0.00%
2	Dumbbells	2022	10	468,041	33.79%	0.00%
3	Exercise Equipment Mats	2022	10	729,959	52.70%	0.00%
4	Balance Foam	2022	11	302,083	9.98%	-26.13%
5	Dumbbells	2022	11	920,402	30.41%	-10.00%
6	Exercise Equipment Mats	2022	11	1,803,709	59.60%	13.09%
7	Balance Foam	2022	12	201,180	7.69%	-22.95%
8	Dumbbells	2022	12	1,139,983	43.57%	43.28%
9	Exercise Equipment Mats	2022	12	1,275,038	48.74%	-18.22%
10	Balance Foam	2023	1	473,250	12.33%	60.34%
11	Dumbbells	2023	1	1,502,490	39.15%	-10.14%
12	Exercise Equipment Mats	2023	1	1,861,665	48.51%	-0.47%
13	Balance Foam	2023	2	316,535	10.07%	-18.33%
14	Dumbbells	2023	2	2,009,723	63.93%	63.30%
15	Exercise Equipment Mats	2023	2	817,387	26.00%	-46.40%
16	Balance Foam	2023	3	488,362	13.29%	31.98%
17	Dumbbells	2023	3	2,235,260	60.82%	-4.86%
18	Exercise Equipment Mats	2023	3	951,652	25.89%	-0.42%
19	Balance Foam	2023	4	312,074	14.24%	7.15%
20	Dumbbells	2023	4	1,346,980	61.48%	1.09%

Question 3: Investment Effectiveness

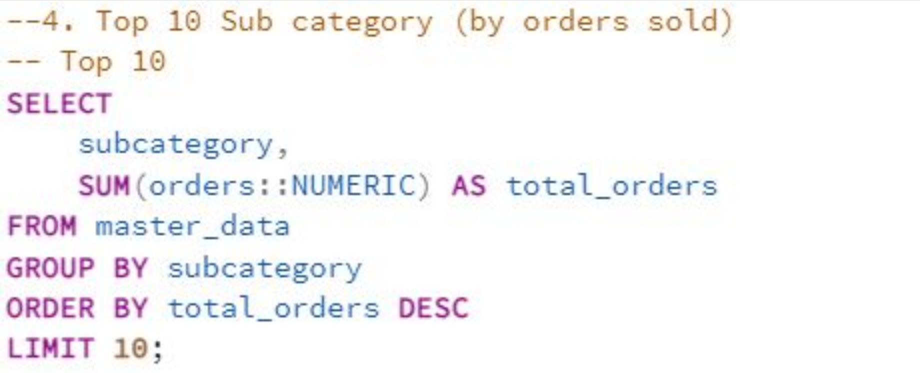


```
-- 3. Channel (ads & promotions) effectiveness
-- to see what treatment work best with what category
WITH channel_performance AS (
    SELECT
        category,
        'Ad Spend' AS channel_type,
        SUM(ad_spent::NUMERIC) AS spend,
        SUM(ordered_gmv::NUMERIC) AS GMV
    FROM master_data
    GROUP BY category

    UNION ALL

    SELECT
        category,
        'Promotion Spend' AS channel_type,
        SUM(promotion_spent::NUMERIC) AS spend,
        SUM(ordered_gmv::NUMERIC) AS GMV
    FROM master_data
    GROUP BY category
)
SELECT
    category,
    channel_type,
    TO_CHAR(spend, 'FM999,999,999,999') AS spend,
    TO_CHAR(GMV, 'FM999,999,999,999') AS GMV,
    COALESCE(TO_CHAR(ROUND(GMV / NULLIF(spend, 0), 2), 'FM990.00'), 'N/A') AS ROI
FROM channel_performance
ORDER BY category, channel_type;
```

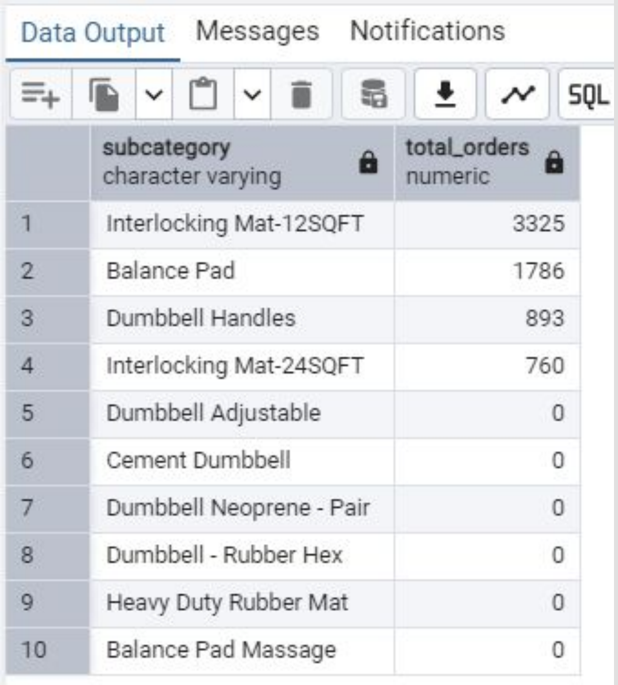


	category character varying	channel_type text	spend text	gmv text	roi text
1	Balance Foam	Ad Spend	5,850	2,280,677	389.84
2	Balance Foam	Promotion Spend	16,673	2,280,677	136.79
3	Dumbbells	Ad Spend	390,922	9,622,879	24.62
4	Dumbbells	Promotion Spend	209,757	9,622,879	45.88
5	Exercise Equipment Mats	Ad Spend	212,433	7,971,426	37.52
6	Exercise Equipment Mats	Promotion Spend	370,366	7,971,426	21.52

Question 4: Top 10 Best-Selling Subcategories

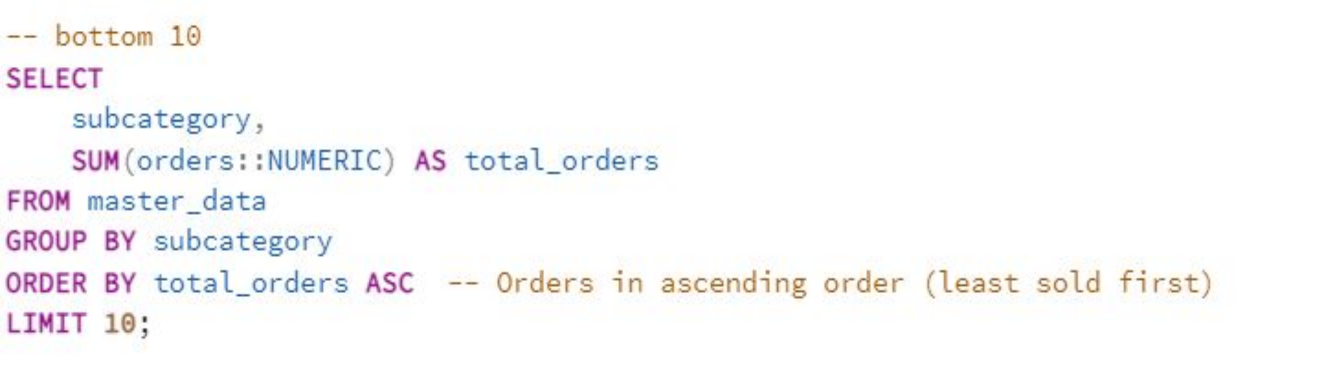


```
--4. Top 10 Sub category (by orders sold)
-- Top 10
SELECT
    subcategory,
    SUM(orders::NUMERIC) AS total_orders
FROM master_data
GROUP BY subcategory
ORDER BY total_orders DESC
LIMIT 10;
```

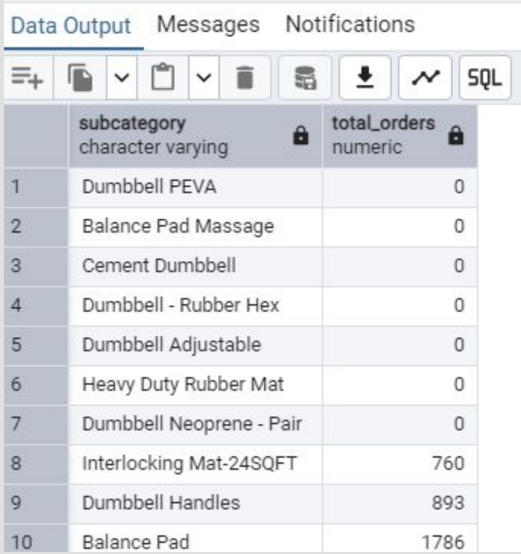


	subcategory character varying	total_orders numeric
1	Interlocking Mat-12SQFT	3325
2	Balance Pad	1786
3	Dumbbell Handles	893
4	Interlocking Mat-24SQFT	760
5	Dumbbell Adjustable	0
6	Cement Dumbbell	0
7	Dumbbell Neoprene - Pair	0
8	Dumbbell - Rubber Hex	0
9	Heavy Duty Rubber Mat	0
10	Balance Pad Massage	0

Question 4: Bottom 10 Selling Subcategories

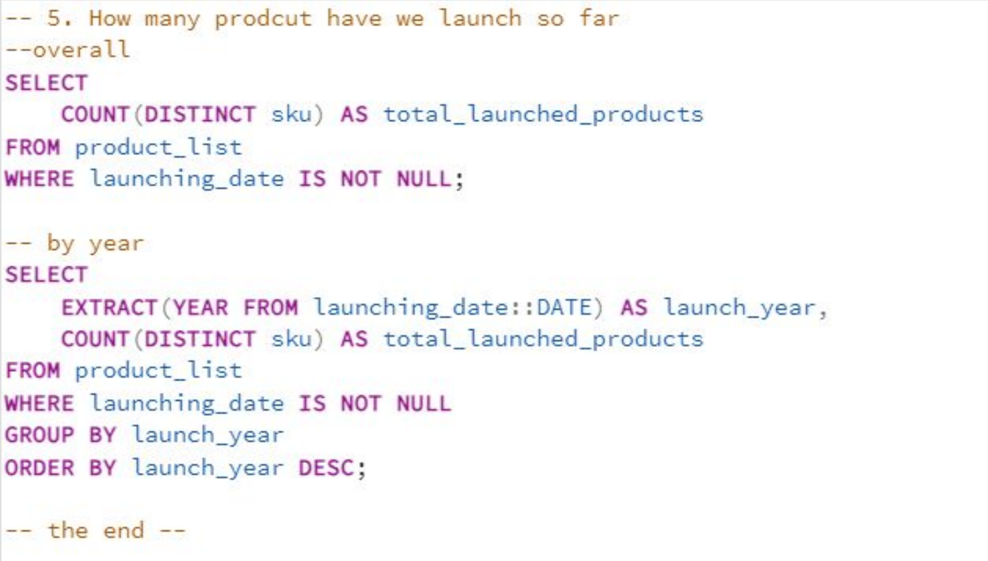


```sql
-- bottom 10
SELECT
    subcategory,
    SUM(orders::NUMERIC) AS total_orders
FROM master_data
GROUP BY subcategory
ORDER BY total_orders ASC  -- Orders in ascending order (least sold first)
LIMIT 10;
```

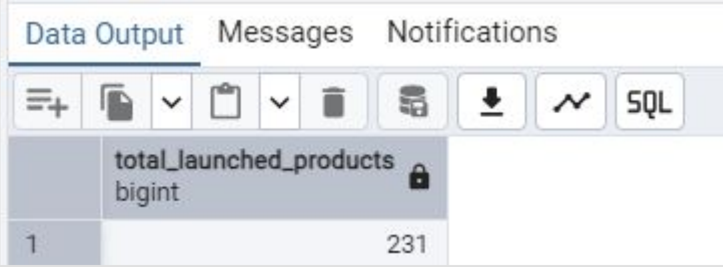


	subcategory character varying	total_orders numeric
1	Dumbbell PEVA	0
2	Balance Pad Massage	0
3	Cement Dumbbell	0
4	Dumbbell - Rubber Hex	0
5	Dumbbell Adjustable	0
6	Heavy Duty Rubber Mat	0
7	Dumbbell Neoprene - Pair	0
8	Interlocking Mat-24SQFT	760
9	Dumbbell Handles	893
10	Balance Pad	1786

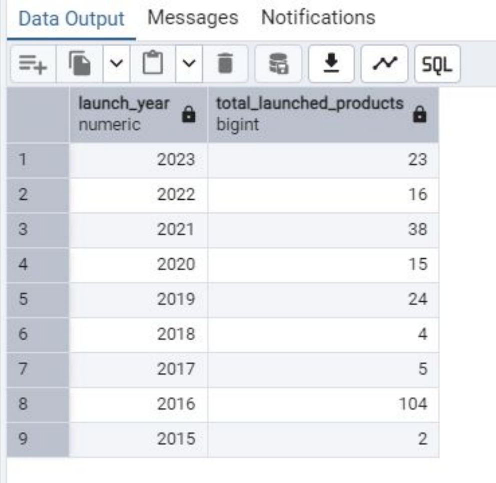
Question 5: How many products have we launched so far?



```sql
-- 5. How many prodcut have we launch so far
--overall
SELECT
    COUNT(DISTINCT sku) AS total_launched_products
FROM product_list
WHERE launching_date IS NOT NULL;

-- by year
SELECT
    EXTRACT(YEAR FROM launching_date::DATE) AS launch_year,
    COUNT(DISTINCT sku) AS total_launched_products
FROM product_list
WHERE launching_date IS NOT NULL
GROUP BY launch_year
ORDER BY launch_year DESC;

-- the end --
```

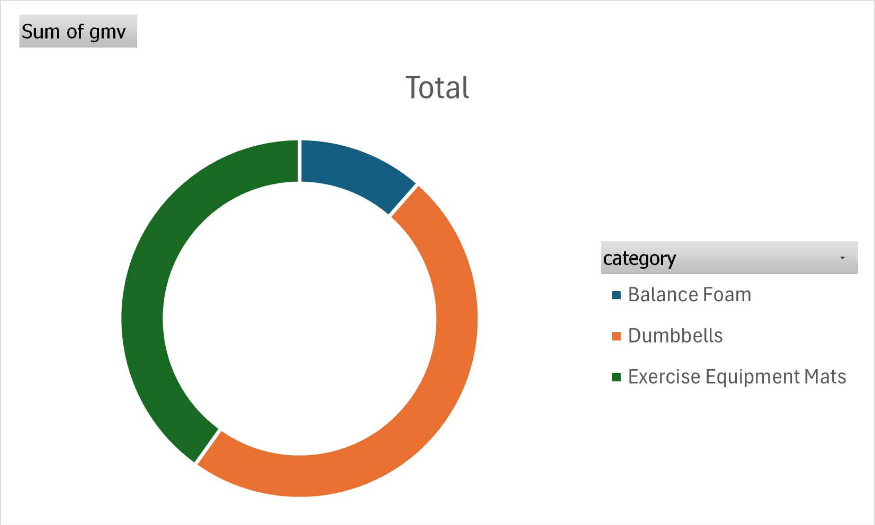


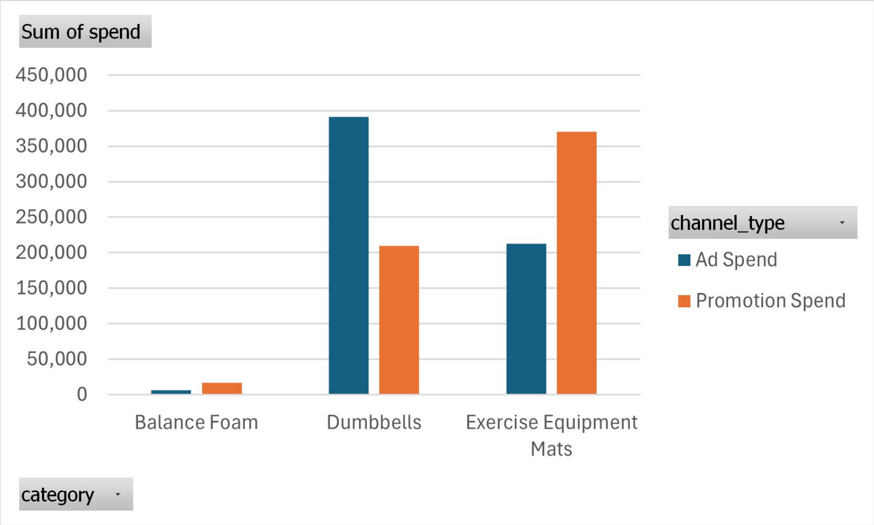
	total_launched_products bigint
1	231



	launch_year numeric	total_launched_products bigint
1	2023	23
2	2022	16
3	2021	38
4	2020	15
5	2019	24
6	2018	4
7	2017	5
8	2016	104
9	2015	2

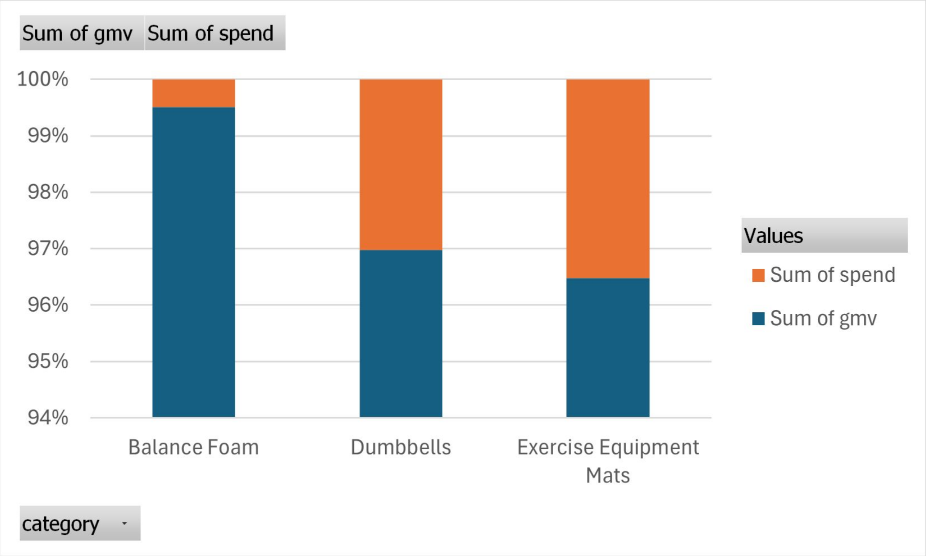
Visualization Output - Spending & Earnings





- Majority of the spending goes into Dumbbells and Exercise Mats.
- Dumbbells and Exercise Mats comprise the majority of GMV

Visualization Output - Spending v Earnings



- Based on spending v earnings, the balance foam shows the best ROI
- Dumbbells and the Exercise Mats yield reasonable ROI
- What are the next steps the business can take to maximize profits?

Business Insight

- **Focus on high ROI product**
  - the Balance Foam.
- **Evaluate cost-intensive products**
  - In this case, the two other products are costly in terms of ROI, but we need more information to analyze and present a solution
  - Is the company spending too much unnecessary spending into an already saturated market?
- **Portofolio Balance**
  - Are there more products that the company can test in the market to boost sales and profits?

Conclusion

- This project bridges the gap between raw data and strategic business decisions, enhancing the company’s ability to respond proactively to market demands.
- With more available data, we can forecast and conduct more sophisticated analysis

References

YES4ALL TRADING SERVICES COMPANY LIMITED (YES4ALL): **Commercial Analyst Test (2024)**YES4ALL TRADING SERVICES COMPANY LIMITED (YES4ALL): **Commercial Analyst Test (2024)**

Thank You