Made By Neeraj Dangwal

Myntra Sales Presentation

Analyzed the Myntra Sales dataset from Kaggle using PostgreSQL to generate insights on brand performance, pricing, discounts, and customer preferences. Applied joins, aggregations, subqueries, CTEs, and window functions to answer business questions such as top-performing brands, most expensive products, highest-rated items, and discount trends.

Data Cleaning & Integrity

- The dataset was thoroughly examined for missing, inconsistent, or erroneous data.
- No missing values were found in any of the columns, ensuring reliable analysis.
- Data types were validated and appropriate tables with constraints were created in PostgreSQL.

How it Works

Step 1 - Data Collection

- Imported the Myntra sales dataset from Kaggle.

Step 2 - Data Preparation

- Cleaned data, handled null values, created PostgreSQL tables

Step 3 - Querying With SQL

- Used joins, aggregations, CTEs & window functions for analysis

Step 4 - Insights & Visualization

- Extracted trends & visualized brand, pricing, and discount patterns

MYNTRA PRODUCT & BRAND PERFORMANCE ANALYSIS

A. Basic Queries

01 Show all brands available in the dataset.

02 Show the first 10 products with their brand name and price.

03 Find the total number of products for each brand.

B. Filtering & Sorting

04 List all products from 'Levis' that have a discount greater than 40%.

05 Find the top 5 most expensive products (after discount).

C. Aggregations

06 Calculate the average price of products for each brand.

07 Find the highest discount percentage for each brand.

D. Subqueries

08 Find all products priced below the average product price.

09 Find the brand(s) with the maximum number of products.

E. Window Functions

10 Rank products within each brand based on ratings.

11 Find the top 3 most-rated products (by number_of_ratings) in each brand.

F. CTEs (Common Table Expressions)

12 Find the best-selling product (highest number_of_ratings) per brand.

13 Find the average discount across all products and list brands with higher-than-average discounts.

G. Business-Oriented Analysis

14 Calculate the total revenue per brand (price × number_of_ratings as a proxy for sales).

15 Find the brand with the highest-rated product overall.

16 Find the percentage of products across all brands that have a rating above 4.

A. BASIC QUERIES

01

```sql
SELECT brand_name
FROM brands;
```

brand_name character varying (255)
WROGN
Flying Machine
Roadster

02

```sql
SELECT b.brand_name, p.price
FROM brands b
JOIN products p ON b.brand_id = p.brand_id
LIMIT 10;
```

brand_name character varying (255)	price numeric (10,2)
WROGN	1374.00
Flying Machine	1829.00
Roadster	974.00
Bene Kleed	873.00

03

```sql
SELECT b.brand_name, COUNT(p.product_id) AS total_products
FROM brands b
JOIN products p ON b.brand_id = p.brand_id
GROUP BY b.brand_name
ORDER BY total_products DESC;
```

brand_name character varying (255)	total_products bigint
WROGN	5303
Flying Machine	5197
United Colors of Benetton	3985
Roadster	3523

B. Filtering & Sorting

04

```sql
SELECT b.brand_name, p.product_id, p.discount_percent
FROM brands b
JOIN products p ON b.brand_id = p.brand_id
WHERE b.brand_name = 'Levis'
  AND p.discount_percent > 0.40;
```

brand_name character vary	product_id integer	discount_percent numeric (5,2)
Levis	5	0.49
Levis	26	0.48
Levis	72	0.55
Levis	122	0.55

05

```sql
SELECT b.brand_name,p.price
FROM products p
JOIN brands b ON p.brand_id = b.brand_id
ORDER BY p.price DESC
LIMIT 5;
```

brand_name character varying (	price numeric (10
Jacob Cohen	54000.00
Jacob Cohen	53062.00
United Colors ...	53062.00

C. Aggregations

06

```sql
SELECT b.brand_name, ROUND(AVG(p.price), 2) AS avg_price
FROM brands b
JOIN products p ON b.brand_id = p.brand_id
GROUP BY b.brand_name
ORDER BY avg_price DESC;
```

brand_name character varying (255)	avg_price numeric
Jacob Cohen	33142.50
Tramarossa	22345.05
Karl Lagerfeld	18487.50
Just Cavalli	18149.00

07

```sql
SELECT b.brand_name, MAX(p.discount_percent) AS max_discount
FROM brands b
JOIN products p ON b.brand_id = p.brand_id
GROUP BY b.brand_name
ORDER BY max_discount DESC;
```

brand_name character varyir	max_discount numeric
FREAKINS	64.00
Red Tape	58.98
United Colo…	52.53
V-Mart	50.05

D. Subqueries

08

```sql
SELECT b.brand_name, p.price
FROM products p
JOIN brands b ON p.brand_id = b.brand_id
WHERE p.price < (SELECT AVG(price) FROM products);
```

brand_name character varyin	price numeric
WROGN	1374.00
Roadster	974.00
Bene Kleed	873.00
Levis	1478.00

09

```sql
SELECT brand_name
FROM (
    SELECT b.brand_name, COUNT(*) AS total_products
    FROM brands b
    JOIN products p ON b.brand_id = p.brand_id
    GROUP BY b.brand_name
) sub
WHERE total_products = (
    SELECT MAX(total_products)
    FROM (
        SELECT COUNT(*) AS total_products
        FROM products
        GROUP BY brand_id
    ) t
);
```

brand_name character varying (255)
WROGN

E. Window Function

10

```sql
SELECT b.brand_name, p.pants_description, p.ratings,
       RANK() OVER (PARTITION BY b.brand_name ORDER BY p.ratings DESC) AS rank_in_brand
FROM products p
JOIN brands b ON p.brand_id = b.brand_id;
```

brand_name character varying (2	pants_description text	ratings numeric	rank_in_brand bigint
7 For All Mankind	Men Slim Fit Jeans	4.60	1
7 For All Mankind	Men Slim Fit Mid-Rise Jeans	4.60	1
7 For All Mankind	Men Slim Fit Jeans	4.60	1
7 For All Mankind	Men Skinny Fit Cotton Jeans	4.30	4

E. Window Function

11

```sql
SELECT *
FROM (
    SELECT b.brand_name, p.pants_description, p.number_of_ratings,
           ROW_NUMBER() OVER (PARTITION BY b.brand_name ORDER BY p.number_of_ratings DESC) AS rn
    FROM products p
    JOIN brands b ON p.brand_id = b.brand_id
) sub
WHERE rn <= 3;
```

brand_name character varying (25	pants_description text	number_of_ratings numeric (10,2)	rn bigint
7 For All Mankind	Men Slim Fit Cotton Jeans	130.00	1
7 For All Mankind	Men Slim Fit Jeans	130.00	2
7 For All Mankind	Men Mid-Rise Slim Fit Jeans	63.00	3
7OUNCE	Men Slim Fit Cotton Jeans	264.00	1

F. CTEs (Common Table Expressions)

12

```sql
WITH brand_best AS (
    SELECT b.brand_name, p.pants_description, p.number_of_ratings,
           ROW_NUMBER() OVER (PARTITION BY b.brand_name ORDER BY p.number_of_ratings DESC) AS rn
    FROM products p
    JOIN brands b ON p.brand_id = b.brand_id
)
SELECT brand_name, pants_description, number_of_ratings
FROM brand_best
WHERE rn = 1;
```

brand_name character varying (2	pants_description text	number_of_ratings numeric (10,2)
7 For All Mankind	Men Slim Fit Cotton Jeans	130.00
7OUNCE	Men Slim Fit Cotton Jeans	264.00
AD By Arvind	Men Skinny Fit Jeans	130.00
ADBUCKS	Men Relaxed Fit Cargo Jeans	130.00

F. CTEs (Common Table Expressions)

13

```sql
WITH avg_discount AS (
    SELECT ROUND(AVG(discount_percent),2) AS overall_avg
    FROM products
)
SELECT b.brand_name,ROUND(AVG(p.discount_percent),2) AS brand_avg_discount
FROM products p
JOIN brands b ON p.brand_id = b.brand_id
GROUP BY b.brand_name
HAVING AVG(p.discount_percent) > (SELECT overall_avg FROM avg_discount);
```

brand_name character varying (2!	brand_avg_discount numeric
Calvin Klein Jeans	2.21
ASHTOM	20.15
mode de base	26.24
MYLANA	16.10

G. Business-Oriented Analysis

14

```sql
SELECT b.brand_name, SUM(p.price * p.number_of_ratings) AS total_revenue
FROM products p
JOIN brands b ON p.brand_id = b.brand_id
GROUP BY b.brand_name
ORDER BY total_revenue DESC
LIMIT 5;
```

brand_name character varyin	total_revenue numeric
Flying Mach…	697591862.0000
United Color…	643474540.0000
WROGN	605762331.0000
Levis	472492909.0000

15

```sql
SELECT b.brand_name, p.pants_description, p.ratings
FROM products p
JOIN brands b ON p.brand_id = b.brand_id
ORDER BY p.ratings DESC
LIMIT 1;
```

brand_name character vary	pants_description text	ratings numeric
Roadster	Washed Effect…	5.00

16

```sql
SELECT
   (COUNT(CASE WHEN ratings > 4 THEN 1 END) * 100.0 / COUNT(*)) AS percent_high_rated
FROM products;
```

percent_high_rated numeric
49.7812739831158

KEY INSIGHTS

- Dataset covers 417 brands and 52K+ products.
- Discounts average around 40–60%, strongly influencing sales.
- Top brands by product count: WROGN, Flying Machine, UCB, Roadster, Mufti.
- **Top by customer ratings: Nike, BAWSE., Happy2r, INKD.**
- Revenue leaders: Flying Machine, UCB, WROGN, Levis, Roadster.

CONCLUSION

- SQL analysis revealed top-performing brands, discount patterns, and product trends.
- Discounts boost sales, but high ratings indicate stronger long-term value.
- Focus on quality-driven brands and optimize discount strategies for sustainable growth.

This project demonstrated how SQL can be leveraged to extract actionable insights from Myntra’s sales data. We identified top brands, discount patterns, and product performance metrics. The analysis shows that while discounts drive visibility, customer ratings are a stronger indicator of long-term brand success. Going forward, Myntra can focus on balancing discounts with quality-driven marketing and explore advanced analytics for customer segmentation and demand forecasting.