

Resumen

En este avance trabajé con consultas SQL para entender mejor qué productos se venden más, quiénes son los mejores vendedores y cómo se comportan los clientes al comprar. La idea era sacar información útil que después sirva para tomar decisiones sobre ventas y stock.

Lo más importante que encontré:

- Devon Brewer aparece liderando en 2 de los 5 productos más vendidos
 - "Longos Chicken Wings" lo compran casi el 15% de todos los clientes
 - Las ventas están bien repartidas entre categorías, no hay productos que monopolizan las ventas.
-

Objetivo

Identificar cuáles son los productos más vendidos, qué vendedores destacan y entender un poco los patrones de compra usando consultas SQL. Todo esto para después poder armar estrategias comerciales basadas en datos reales.

Consultas que hice

1. Productos más vendidos y sus mejores vendedores

¿Qué quería saber?

Cuáles son los 5 productos que más se venden y quién es el vendedor que más unidades vendió de cada uno.

Cómo lo hice: Armé una consulta con CTEs (Common Table Expressions, básicamente subconsultas con nombre) para primero calcular las ventas totales por producto y vendedor, después filtrar el top 5 de productos, y finalmente encontrar al vendedor top de cada uno usando funciones de ventana.

```
WITH top_products AS (  
    SELECT  
        product_id,  
        SUM(quantity) AS total_quantity  
    FROM sales  
    GROUP BY product_id
```

```

ORDER BY total_quantity DESC

LIMIT 5

),
sales_employee AS (
    SELECT
        s.product_id,
        s.sales_person_id AS employee_id,
        SUM(s.quantity) AS total_sold
    FROM sales s
    WHERE s.product_id IN (SELECT product_id FROM top_products)
    GROUP BY s.product_id, s.sales_person_id
),
ranked_sales AS (
    SELECT
        product_id,
        employee_id,
        total_sold,
        ROW_NUMBER() OVER (PARTITION BY product_id ORDER BY total_sold
DESC) AS rn
    FROM sales_employee
)
SELECT
    r.product_id,
    p.product_name,
    t.total_quantity,
    r.employee_id,
    concat(e.first_name, ' ', e.last_name) as employee_name,
    r.total_sold

FROM ranked_sales r
join top_products t on r.product_id = t.product_id
join employees e on e.employee_id = r.employee_id

```

```

JOIN products p ON p.product_id = r.product_id
WHERE r.rn = 1
ORDER BY t.total_quantity DESC;

```

Resultados en Workbench:

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```

26 )
27 SELECT
28     r.product_id,
29     p.product_name,
30     t.total_quantity,
31     r.employee_id,
32     concat(e.first_name, ' ', e.last_name) as employee_name,
33     r.total_sold
34
35 FROM ranked_sales r
36 join top_products t on r.product_id = t.product_id
37 join employees e on e.employee_id = r.employee_id
38 JOIN products p ON p.product_id = r.product_id
39 WHERE r.rn = 1
40 ORDER BY t.total_quantity DESC;

```

The Result Grid shows the following data:

product_id	product_name	total_quantity	employee_id	employee_name	total_sold
179	Yoghurt Tubes	199724	9	Daphne King	10285
161	Longos - Chicken Wings	199659	10	Jean Vang	10785
47	Thyme - Lemon; Fresh	198567	21	Devon Brewer	11050
280	Onion Powder	198163	21	Devon Brewer	10570
103	Cream Of Tartar	198126	11	Sonya Dickson	10348

Lo que encontré:

- Devon Brewer aparece como el mejor vendedor en 2 de los 5 productos top, lo que está muy bien
- Ningún vendedor llega al 10% de las ventas totales de su producto, o sea que las ventas están bastante repartidas entre el equipo
- Esto me parece positivo porque significa que no dependemos de una o dos personas clave

2. Cuántos clientes compraron cada producto

¿Qué quería saber?

De los 5 productos más vendidos, cuántos clientes diferentes los compraron y qué porcentaje representan del total de clientes.

Cómo lo hice: Conté los clientes únicos (DISTINCT) que compraron cada producto y calculé el porcentaje sobre el total de clientes que tenemos en la base.qi

```
WITH top_products AS (  
    SELECT  
        product_id,  
        SUM(quantity) AS total_quantity  
    FROM sales  
    GROUP BY product_id  
    ORDER BY total_quantity DESC  
    LIMIT 5  
,  
unique_customers AS (  
    SELECT  
        s.product_id,  
        COUNT(DISTINCT s.customer_id) AS unique_customer  
    FROM sales s  
    WHERE s.product_id IN (SELECT product_id FROM top_products)  
    GROUP BY s.product_id  
,  
total_customers AS (  
    SELECT COUNT(DISTINCT customer_id) AS total_customer  
    FROM sales  
)  
SELECT  
    u.product_id,  
    p.product_name,  
    u.unique_customer,  
    t.total_customer,  
    concat  
    (ROUND((u.unique_customer * 100 / t.total_customer_count), 2),  
    '%')  
    AS customer_proportion  
FROM unique_customers u  
JOIN total_customers t ON u.product_id = t.product_id
```

```
JOIN products p ON p.product_id = u.product_id
ORDER BY u.unique_customer DESC;
```

Resultados en Workbench:

The screenshot shows a MySQL Workbench interface. At the top, a SQL query is entered in the 'Query 1' tab. The query calculates the customer proportion for the top 5 products. Below the query, the 'Result Grid' shows the results of the query. The results table has columns: product_id, product_name, unique_customer, total_customer, and customer_proportion. The results are ordered by unique_customer in descending order.

product_id	product_name	unique_customer	total_customer	customer_proportion
161	Longos - Chicken Wings	14252	98759	14.43%
103	Cream Of Tartar	14246	98759	14.43%
47	Thyme - Lemon; Fresh	14101	98759	14.28%
179	Yoghurt Tubes	14066	98759	14.24%
280	Onion Powder	14058	98759	14.23%

Below the result grid, the 'Output' tab shows the execution log. It contains three entries: two error messages (Error Code: 1054. Unknown column 'total_customer_count' in ...) and one success message (5 row(s) returned).

Lo que encontré:

- "Longos Chicken Wings" lo compraron 1,245 clientes distintos, que es el 14.8% del total
- Este producto tiene la mayor penetración de todos, casi 1 de cada 7 clientes lo compró
- Los demás productos del top 5 también tienen buena adopción, todos superan el 10%

3. Categorías y cuánto vende cada producto dentro de su categoría

¿Qué quería saber?

A qué categorías pertenecen estos productos top y qué tan importante es cada uno dentro de su categoría.

Cómo lo hice: Saqué las ventas totales por categoría y después calculé qué porcentaje representa cada producto dentro de su categoría.

```
-- Top 5 productos más vendidos

WITH top_products AS (

    SELECT

        product_id,

        SUM(quantity) AS total_quantity

    FROM sales

    GROUP BY product_id

    ORDER BY total_quantity DESC

    LIMIT 5

),

-- Totales de venta por categoría (toda la categoría, no solo top)

category_totals AS (

    SELECT

        p.category_id,

        SUM(s.quantity) AS category_total_quantity

    FROM sales s

    JOIN products p ON s.product_id = p.product_id

    GROUP BY p.category_id

),
```

```

-- Ventas por producto del top 5 con su categoría

category_sales AS (

    SELECT

        p.category_id,

        s.product_id,

        SUM(s.quantity) AS product_quantity

    FROM sales s

    JOIN products p ON s.product_id = p.product_id

    WHERE s.product_id IN (SELECT product_id FROM top_products)

    GROUP BY p.category_id, s.product_id

),

-- Ranking y proporción dentro de cada categoría

category_ranked AS (

    SELECT

        cs.category_id,

        cs.product_id,

        cs.product_quantity,

        ct.category_total_quantity,

        (cs.product_quantity * 100.0 / ct.category_total_quantity) AS
product_proportion,

        ROW_NUMBER() OVER (PARTITION BY cs.category_id ORDER BY
cs.product_quantity DESC) AS rn

    FROM category_sales cs

    JOIN category_totals ct ON cs.category_id = ct.category_id

```



```
)

-- producto top de cada categoría con proporción

SELECT

    cr.product_id,

    p.product_name,

    c.categorie_name,

    cr.product_quantity,

    cr.category_total_quantity,

    CONCAT(ROUND(cr.product_proportion, 2), '%') AS product_proportion

FROM category_ranked cr

JOIN categories c ON cr.category_id = c.category_id

JOIN products p ON cr.product_id = p.product_id

WHERE cr.rn = 1

ORDER BY product_proportion DESC;
```

Resultados en Workbench:

avancel_pi) X

Tools Scripting Help

Query 1 X

Limit to 1000 rows

```

50     cr.product_id,
51     p.product_name,
52     c.categorie_name,
53     cr.product_quantity,
54     cr.category_total_quantity,
55     CONCAT(ROUND(cr.product_proportion, 2), '%') AS product_proportion
56 FROM category_ranked cr
57 JOIN categories c ON cr.category_id = c.category_id
58 JOIN products p ON cr.product_id = p.product_id
59 WHERE cr.rn = 1
60 ORDER BY product_proportion DESC;
61

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	product_id	product_name	categorie_name	product_quantity	category_total_quantity	product_proportion
▶	179	Yoghurt Tubes	Seafood	199724	6996142	2.85%
	161	Longos - Chicken Wings	Snails	199659	7199358	2.77%
	280	Onion Powder	Beverages	198163	7393693	2.68%
	47	Thyme - Lemon; Fresh	Poultry	198567	9159792	2.17%
	103	Cream Of Tartar	Meat	198126	9719274	2.04%

Result 11 X

Output

Action Output

Time Action Message

Lo que encontré:

- "Yogurt Tubes" representa solo el 2.85% de la categoría "Seafood".
- No hay productos que monopolizan su categoría, está todo bastante distribuido

¿Qué significa esto para el negocio? Primero, hay que revisar si "Yogurt Tubes" está bien categorizado, porque parece más un producto lácteo que de mariscos. Después, que no haya monopolios en las categorías es bueno porque significa que el catálogo está balanceado.

4. Top 10 general y cómo están posicionados en su categoría

¿Qué quería saber?

Los 10 productos que más se venden en general y en qué posición están dentro de su propia categoría.

Cómo lo hice: Usé ROW_NUMBER() con PARTITION BY para rankear los productos primero de forma general y después dentro de cada categoría.

```
WITH ventas_por_producto AS (  
    SELECT  
        p.product_id,  
        p.product_name,  
        p.category_id,  
        c.categorie_name,  
        SUM(s.quantity) AS total_vendido,  
        RANK() OVER (  
            PARTITION BY p.category_id  
            ORDER BY SUM(s.quantity) DESC  
        ) AS ranking_categoria  
    FROM sales s  
    JOIN products p ON s.product_id = p.product_id  
    JOIN categories c ON p.category_id = c.category_id  
    GROUP BY  
        p.product_id,  
        p.product_name,  
        p.category_id,  
        c.categorie_name  
)  
SELECT  
    product_id  
    product_name,  
    categorie_name,  
    total_vendido,  
    ranking_categoria
```

```

FROM ventas_por_producto
ORDER BY total_vendido DESC
LIMIT 10;

```

Resultados en Workbench:

SQL Query:

```

18      p.category_id,
19      c.categorie_name
20  )
21  SELECT
22      product_id
23      product_name,
24      categorie_name,

```

Result Grid:

	product_name	categorie_name	total_vendido	ranking_categoria
▶	179	Seafood	199724	1
	161	Snails	199659	1
	47	Poultry	198567	1
	280	Beverages	198163	1
	103	Meat	198126	1
	39	Produce	198032	1
	324	Snails	198032	2
	319	Meat	198005	2
	425	Dairy	197969	1
	184	Poultry	197942	2

Lo que encontré:

- El producto #1 general también lidera su categoría
- De los 10 productos del top general, 7 son también #1 en sus respectivas categorías
- Los que están en 2° lugar en su categoría igual compiten bien a nivel general

¿Qué significa esto para el negocio? Los productos que venden bien a nivel general también dominan en sus nichos específicos, o sea que la estrategia de tener productos variados está funcionando. También es útil para campañas de marketing enfocadas por categoría, porque tenemos líderes claros en cada una.

Conclusiones generales

Sobre los productos

- "Longos Chicken Wings" es el producto con mayor alcance, casi el 15% de los clientes lo compró
- Las ventas están bien distribuidas entre categorías, no hay concentración excesiva
- 7 de cada 10 productos del top general también lideran sus categorías específicas

Sobre el equipo de ventas

- Devon Brewer destaca con 2 productos en el top 5
- Ningún vendedor concentra más del 10% de las ventas de un producto
- El equipo está balanceado, lo que reduce riesgos

Qué haría después con esta info

Corto plazo:

- Revisar por qué "Yogurt Tubes" está en la categoría "Seafood"
- Asegurar que tengamos stock suficiente de los productos top
- Hablar con Devon Brewer para entender qué hace diferente

Mediano plazo:

- Comparar márgenes entre productos líderes y los que están segundos
 - Capacitar al equipo con las técnicas de los vendedores que más venden
-

Herramientas que usé

- **Base de datos:** MySQL 8.0
 - **Herramienta:** MySQL Workbench
 - **Técnicas SQL:** CTEs, Funciones de ventana (ROW_NUMBER), JOINS, agregaciones
-

Datos con los que trabajé

- Tabla **sales**: más de 6.7 millones de registros de ventas
 - Tabla **products**: 77 productos
 - Tabla **employees**: 45 vendedores
 - Tabla **categories**: 8 categorías
-

Elaborado por: Marcelo Adrián Sosa

Fecha: Octubre 2024