--Sales & Revenue Analysis

--What is the total revenue generated per product?

SELECT

p.product\_name,

sum(s.total\_price) as total\_revenue

FROM products p

JOIN sales s ON p.product\_id = s.product\_id

GROUP BY p.product\_name

ORDER BY total\_revenue DESC;

--Which store generated the highest revenue in 2025?

SELECT

st.store\_name,

sum(s.total\_price) AS total\_revenue,

strftime('%Y', s.sale\_date) AS year

FROM sales s

JOIN stores st ON s.store\_id = st.store\_id

WHERE strftime('%Y', s.sale\_date) = '2025'

GROUP BY st.store\_name

ORDER BY total\_revenue DESC

LIMIT 1;

--Which month had the highest total sales across all stores?

SELECT

st.store\_name,

sum(s.total\_price) AS total\_revenue,

strftime('%Y-%m', s.sale\_date) AS sale\_month

FROM stores st

JOIN sales s ON st.store\_id = s.store\_id

GROUP BY sale\_month

ORDER BY total\_revenue DESC

LIMIT 1;

--What is the average order value per store?

SELECT

st.store\_name,

count(s.sale\_id) AS total\_sales,

round(sum(s.total\_price), 2) AS total\_revenue,

round(sum(s.total\_price) \*1.0/count(s.sale\_id),2) AS avg\_order\_value

FROM stores st

JOIN sales s ON st.store\_id = s.store\_id

GROUP BY st.store\_name

ORDER BY avg\_order\_value DESC;

--Product Insights

--Which are the top 5 best-selling products by quantity sold?

SELECT

p.product\_name,

sum(s.quantity\_sold) AS total\_quantity

FROM products p

JOIN sales s ON p.product\_id = s.product\_id

GROUP BY p.product\_name

ORDER BY total\_quantity DESC

LIMIT 5;

--List products that were never sold.

SELECT

p.product\_name

FROM products p

LEFT JOIN sales s ON p.product\_id = s.product\_id

WHERE s.sale\_id IS NULL;

--Which products have a high price but low sales volume?

WITH highest\_price AS (

SELECT product\_id, product\_name, price

FROM products

GROUP BY product\_name

ORDER BY price DESC

LIMIT 10)

SELECT

hp.product\_name,

hp.price,

sum(s.quantity\_sold) AS total\_quantity\_sold

FROM highest\_price hp

JOIN sales s ON hp.product\_id = s.product\_id

GROUP BY hp.product\_name

ORDER BY s.total\_price ASC

LIMIT 10;

-- Customer Behavior

--Which customer made the highest number of purchases?

SELECT

c.name,

count(s.sale\_id) AS total\_orders,

sum(s.total\_price) AS total\_amount

FROM customers c

JOIN sales s ON c.customer\_id = s.customer\_id

GROUP BY c.name

ORDER BY total\_orders DESC

LIMIT 1;

--Which customer spent the most in total?

SELECT

c.name,

sum(s.total\_price) AS total\_amount

FROM customers c

JOIN sales s ON c.customer\_id = s.customer\_id

GROUP BY c.name

ORDER BY total\_amount DESC

LIMIT 1;

--Find customers who made purchases in every quarter of 2025.

WITH customer\_quarter AS (

SELECT

c.customer\_id,

c.name,

CASE

WHEN strftime('%m', s.sale\_date) BETWEEN '01' AND '03' THEN 'Q1'

WHEN strftime('%m', s.sale\_date) BETWEEN '04' AND '06' THEN 'Q2'

WHEN strftime('%m', s.sale\_date) BETWEEN '07' AND '09' THEN 'Q3'

WHEN strftime('%m', s.sale\_date) BETWEEN '10' AND '12' THEN 'Q4'

END AS quarter

FROM customers c

JOIN sales s ON s.customer\_id = c.customer\_id

WHERE strftime('%Y', s.sale\_date ) = '2025'

)

SELECT

name,

count(distinct quarter) AS quarter\_covered

FROM customer\_quarter

GROUP BY name

HAVING quarter\_covered = 4;

-- Store Performance

--Which store had the most diverse product sales (sold the most different products)?

SELECT

st.store\_name,

count(DISTINCT p.product\_id) AS product\_variety,

count(s.sale\_id) AS times\_sales

FROM sales s

JOIN stores st ON s.store\_id = st.store\_id

JOIN products p ON s.product\_id = p.product\_id

GROUP BY st.store\_name

ORDER BY product\_variety DESC

LIMIT 1;

--Find the peak sales day for each store.

with peak\_sale\_day AS (

SELECT

st.store\_name,

date(s.sale\_date) as sale\_day,

sum(s.total\_price) as total\_sales

FROM sales s

JOIN stores st on s.store\_id = st.store\_id

group by st.store\_name, sale\_day

),

ranked\_sale AS (

SELECT \*,

rank() over (PARTITION by store\_name ORDER BY total\_sales DESC) AS rank

FROM peak\_sale\_day

)

SELECT

store\_name,

sale\_day,

total\_sales

FROM ranked\_sale

WHERE rank = 1

--Which store had the highest average order value?

SELECT

st.store\_name,

count(s.sale\_id) AS total\_sales,

round(avg(s.total\_price), 2) AS avg\_order\_value

FROM stores st

JOIN sales s ON s.store\_id = st.store\_id

GROUP BY st.store\_name

ORDER BY avg\_order\_value DESC

LIMIT 1;

--Trends & Comparisons

--Show monthly revenue trends over 2024–2025.

SELECT

strftime('%Y', sale\_date) AS year,

strftime('%m', sale\_date) AS month,

sum(total\_price) AS monthly\_revenue

FROM sales

WHERE strftime('%Y', sale\_date) IN ('2024', '2025')

GROUP BY year, month

ORDER by year, month

--Compare average revenue per store in 2024 vs 2025.

SELECT

st.store\_name,

strftime('%Y', s.sale\_date) AS year,

round(avg(s.total\_price), 2) AS avg\_revenue

FROM sales s

JOIN stores st on s.store\_id = st.store\_id

WHERE strftime('%Y', s.sale\_date) IN ('2024','2025')

GROUP BY st.store\_name

ORDER BY year;

--What is the trend of total units sold per month?

SELECT

strftime('%Y - %m', sale\_date) AS year\_month,

sum(quantity\_sold) AS total\_quantity,

sum(total\_price) AS total\_revenue

FROM sales s

GROUP BY year\_month

ORDER BY year\_month;

--Advanced Challenge

--Which products were consistently top-selling (in top 3) every month?

WITH monthly\_product\_sales AS (

SELECT

p.product\_id,

p.product\_name,

strftime('%Y-%m', s.sale\_date) AS year\_month,

SUM(s.quantity\_sold) AS total\_sold

FROM sales s

JOIN products p ON s.product\_id = p.product\_id

GROUP BY p.product\_id, year\_month

),

ranked\_sales AS (

SELECT \*,

DENSE\_RANK() OVER (

PARTITION BY year\_month

ORDER BY total\_sold DESC

) AS rank

FROM monthly\_product\_sales

)

SELECT product\_name

FROM ranked\_sales

WHERE rank <= 3

GROUP BY product\_id

HAVING COUNT(DISTINCT year\_month) = (SELECT COUNT(DISTINCT strftime('%Y-%m', sale\_date)) FROM sales);

--Find products whose sales dropped 3 months in a row.

WITH monthly\_sales AS (

SELECT

p.product\_id,

p.product\_name,

strftime('%Y-%m', s.sale\_date) AS year\_month,

SUM(s.total\_price) AS monthly\_sales

FROM sales s

JOIN products p ON p.product\_id = s.product\_id

GROUP BY p.product\_id, year\_month

),

sales\_with\_lags AS (

SELECT

\*,

LAG(monthly\_sales, 1) OVER (PARTITION BY product\_id ORDER BY year\_month) AS prev\_1,

LAG(monthly\_sales, 2) OVER (PARTITION BY product\_id ORDER BY year\_month) AS prev\_2

FROM monthly\_sales

),

dropped\_3\_months AS (

SELECT

product\_name,

year\_month,

monthly\_sales,

prev\_1,

prev\_2

FROM sales\_with\_lags

WHERE monthly\_sales < prev\_1

AND prev\_1 < prev\_2

)

SELECT DISTINCT product\_name

FROM dropped\_3\_months;

--Find customers who only purchased from one store

SELECT

c.customer\_id,

c.name,

COUNT(DISTINCT s.store\_id) AS store\_count

FROM customers c

JOIN sales s ON c.customer\_id = s.customer\_id

GROUP BY c.customer\_id, c.name

HAVING COUNT(DISTINCT s.store\_id) = 1;