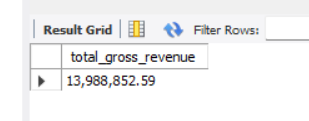
1.

### 1. Total Gross Revenue

SELECT FORMAT(SUM(quantity \* unit\_price), 2) AS total\_gross\_revenue

FROM fact\_transaction;



2.

###2. Total Net Revenue (after discount)

SELECT FORMAT(SUM(quantity \* unit\_price \* (1 - discount\_perc / 100)), 2) AS total\_net\_revenue

FROM fact\_transaction;

A screenshot of a computer

AI-generated content may be incorrect.

3.

###3. Total Profit

SELECT FORMAT(

SUM(

(quantity \* unit\_price \* (1 - discount\_perc / 100)) -

(quantity \* unit\_manufacturing\_cost + quantity \* unit\_shipping\_cost)

), 2

) AS total\_profit

FROM fact\_transaction;

A screenshot of a computer

AI-generated content may be incorrect.

4. ### 4. Orders by Customer Segment

SELECT customer\_segment, COUNT(DISTINCT order\_id) AS total\_orders

FROM fact\_transaction

GROUP BY customer\_segment;

A screenshot of a computer

AI-generated content may be incorrect.

5.

### 5. Top 5 Best-Selling Products by Quantity

SELECT product\_id, SUM(quantity) AS total\_quantity

FROM fact\_transaction

GROUP BY product\_id

ORDER BY total\_quantity DESC

LIMIT 5;

A screenshot of a computer

AI-generated content may be incorrect.

6.

SELECT

DATE\_FORMAT(order\_date, '%Y-%m') AS month,

FORMAT(SUM(quantity \* unit\_price), 2) AS gross\_revenue

FROM fact\_transaction

WHERE YEAR(order\_date) = 2015

GROUP BY month

ORDER BY month;

A screenshot of a computer

AI-generated content may be incorrect.

7.

### 7. Sub-Category with Highest Profit Margin

SELECT sub\_category,

SUM((unit\_price \* (1 - discount\_perc / 100)) - unit\_manufacturing\_cost - unit\_shipping\_cost) /

SUM(unit\_price \* (1 - discount\_perc / 100)) AS profit\_margin

FROM fact\_transaction

GROUP BY sub\_category

ORDER BY profit\_margin DESC

LIMIT 1;

A screenshot of a computer

AI-generated content may be incorrect.

8.

### 8. Continents with Most Product Returns (as % of total sold)

SELECT dl.continent,

COUNT(DISTINCT r.`Order ID`) / COUNT(DISTINCT f.order\_id) \* 100 AS return\_rate\_pct

FROM fact\_transaction f

JOIN dim\_location dl ON f.city = dl.city

LEFT JOIN return\_table r ON f.order\_id = r.`Order ID`

GROUP BY dl.continent

ORDER BY return\_rate\_pct DESC;

A screenshot of a computer

AI-generated content may be incorrect.

9.

### 9. Products with Negative Profit

SELECT product\_id,

SUM(

(quantity \* unit\_price \* (1 - discount\_perc / 100)) -

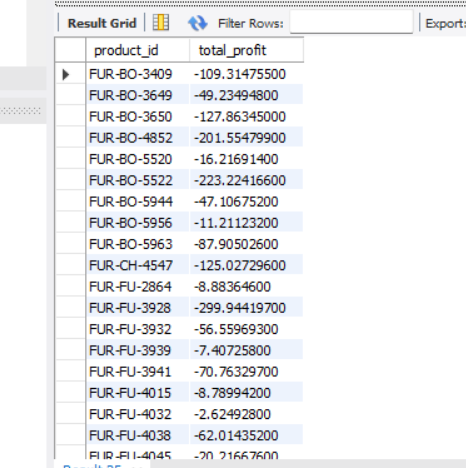
(quantity \* unit\_manufacturing\_cost + quantity \* unit\_shipping\_cost)

) AS total\_profit

FROM fact\_transaction

GROUP BY product\_id

HAVING total\_profit < 0;



10.

### 10. Correlation Between Discount and Order Volume

SELECT

discount\_perc,

ROUND(AVG(unit\_price), 2) AS avg\_unit\_price,

SUM(quantity) AS order\_volume

FROM fact\_transaction

GROUP BY discount\_perc

ORDER BY discount\_perc

LIMIT 25;

A screenshot of a data

AI-generated content may be incorrect.

11.

### 11. Return Rate by Product Category

SELECT category,

COUNT(DISTINCT r.`Order ID`) / COUNT(DISTINCT f.order\_id) \* 100 AS return\_rate\_pct

FROM fact\_transaction f

LEFT JOIN return\_table r ON f.order\_id = r.`Order ID`

GROUP BY category;

A screenshot of a computer

AI-generated content may be incorrect.

12.

SELECT

ship\_mode,

FORMAT(

SUM(

(quantity \* unit\_price \* (1 - discount\_perc / 100)) -

(quantity \* unit\_manufacturing\_cost + quantity \* unit\_shipping\_cost)

), 2

) AS total\_profit

FROM fact\_transaction

GROUP BY ship\_mode

ORDER BY

SUM(

(quantity \* unit\_price \* (1 - discount\_perc / 100)) -

(quantity \* unit\_manufacturing\_cost + quantity \* unit\_shipping\_cost)

) DESC

LIMIT 1;

A screenshot of a computer

AI-generated content may be incorrect.

13.

SELECT

CONCAT(

ROUND(

(SUM(CASE

WHEN LOWER(TRIM(REPLACE(REPLACE(order\_priority, '\r', ''), '\n', ''))) = 'high' THEN 1

ELSE 0

END) / COUNT(\*)) \* 100, 2

),

'%'

) AS high\_priority\_pct

FROM fact\_transaction;

A screenshot of a computer

AI-generated content may be incorrect.

14.

SELECT

city,

FORMAT(

SUM(quantity \* unit\_price) / COUNT(DISTINCT order\_id), 2

) AS revenue\_per\_order

FROM fact\_transaction

GROUP BY city

ORDER BY

SUM(quantity \* unit\_price) / COUNT(DISTINCT order\_id) DESC

LIMIT 1;

A screenshot of a computer

AI-generated content may be incorrect.

15.

### 15. Average Profit per Customer Segment

SELECT customer\_segment,

AVG(

(quantity \* unit\_price \* (1 - discount\_perc / 100)) -

(quantity \* unit\_manufacturing\_cost + quantity \* unit\_shipping\_cost)

) AS avg\_profit

FROM fact\_transaction

GROUP BY customer\_segment;

A screenshot of a computer

AI-generated content may be incorrect.

16.   
SELECT

category,

YEAR(order\_date) AS year,

FORMAT(

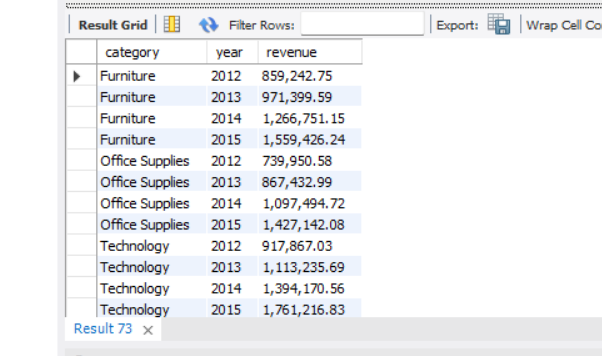
SUM(quantity \* unit\_price \* (1 - discount\_perc / 100)), 2

) AS revenue

FROM fact\_transaction

GROUP BY category, year

ORDER BY category, year;



17.  
### 17. Products Frequently Purchased Together

SELECT a.product\_id AS product\_a, b.product\_id AS product\_b, COUNT(\*) AS times\_bought\_together

FROM (

SELECT DISTINCT order\_id, product\_id

FROM fact\_transaction

) a

JOIN (

SELECT DISTINCT order\_id, product\_id

FROM fact\_transaction

) b

ON a.order\_id = b.order\_id AND a.product\_id < b.product\_id

GROUP BY product\_a, product\_b

ORDER BY times\_bought\_together DESC;

A screenshot of a computer

AI-generated content may be incorrect.

18.

SELECT

CONCAT(

ROUND(

(COUNT(CASE WHEN product\_count > 1 THEN 1 END) / COUNT(\*)) \* 100, 2

),

'%'

) AS multi\_product\_order\_pct

FROM (

SELECT order\_id, COUNT(DISTINCT product\_id) AS product\_count

FROM fact\_transaction

GROUP BY order\_id

) t;

A screenshot of a computer

AI-generated content may be incorrect.

19.

### 19. Orders with Abnormally High Shipping Costs

SELECT order\_id,

SUM(unit\_shipping\_cost \* quantity) AS total\_shipping\_cost

FROM fact\_transaction

GROUP BY order\_id

HAVING total\_shipping\_cost > (

SELECT AVG(unit\_shipping\_cost \* quantity) + 2 \* STDDEV(unit\_shipping\_cost \* quantity)

FROM fact\_transaction

);

A screenshot of a data

AI-generated content may be incorrect.

20.

SELECT

value\_segment,

COUNT(\*) AS order\_count,

ROUND( COUNT(\*) / (SELECT COUNT(DISTINCT order\_id) FROM fact\_transaction) \* 100, 2) AS percentage\_of\_orders

FROM (

SELECT order\_id,

SUM(quantity \* unit\_price \* (1 - discount\_perc / 100)) AS total\_value,

CASE

WHEN SUM(quantity \* unit\_price \* (1 - discount\_perc / 100)) < 100 THEN 'Low'

WHEN SUM(quantity \* unit\_price \* (1 - discount\_perc / 100)) < 500 THEN 'Medium'

ELSE 'High'

END AS value\_segment

FROM fact\_transaction

GROUP BY order\_id

) t

GROUP BY value\_segment

ORDER BY FIELD(value\_segment, 'Low', 'Medium', 'High');

A screenshot of a computer

AI-generated content may be incorrect.