FINAL PROJECT
DATA ANALYSIS - SQL

By: Endah Rakhmawati

Intensive Bootcamp Data Analysis Batch 18 | MySkill

Overview

Project

SQL enables Data Analysts to access and manipulate data efficiently. With SQL, they can search, filter, calculate, aggregate, sort, group, and merge data easily. Data Analysts can optimize their SQL queries to get results quickly, especially when dealing with large volumes of data.

Dataset

The dataset used is sales data from Tokopedia (not real data). It consists of 4 tables in the period 2021 to 2022.

Dataset

order_detail:

1. id → unique number of order / id_order
2. customer_id → unique number of customer
3. order_date → date when transaction was made
4. sku_id → unique number of product (sku is stock keeping unit)
5. price → price listed on price tag
6. qty_ordered → number of items purchased by customer
7. before_discount → total price value of product (price * qty_ordered)
8. discount_amount → total product discount value
9. after_discount → total price value of product when reduced by discount
10. is_gross → indicates customer has not paid for order
11. is_valid → indicates customer has made payment
12. is_net → indicates transaction is complete
13. payment_id → unique number of payment method

Dataset

sku_detail:

1. id → unique number of the product (can be used for key when joining)
2. sku_name → name of the product
3. base_price → price of goods listed on the price tag / price
4. cogs → cost of goods sold / total cost to sell 1 product
5. category → product category

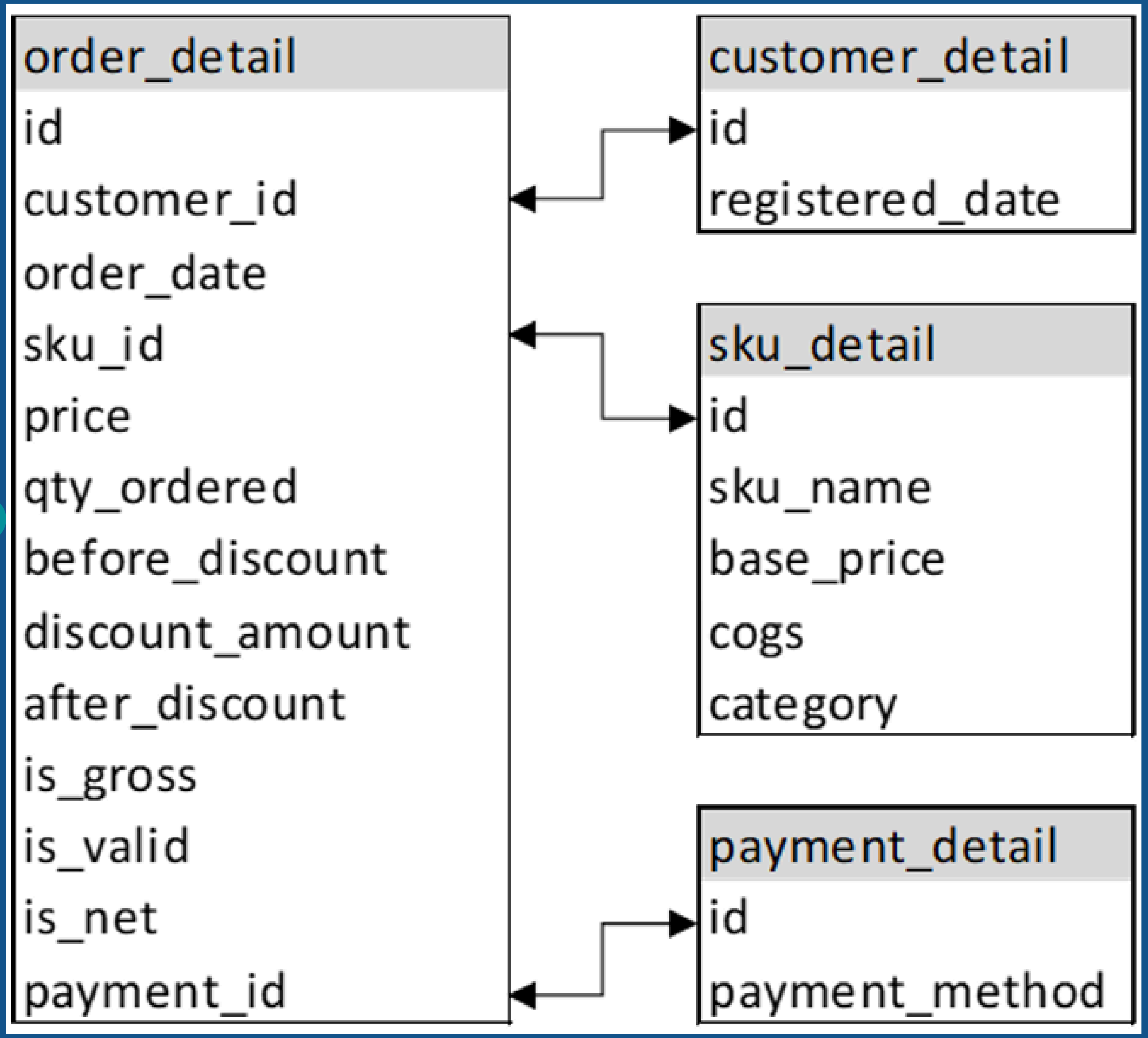
customer_detail:

1. id → unique number of the customer
2. registered_date → date the customer started registering as a member

payment_detail:

1. id → unique number of payment method
2. payment_method → payment method used

Schema



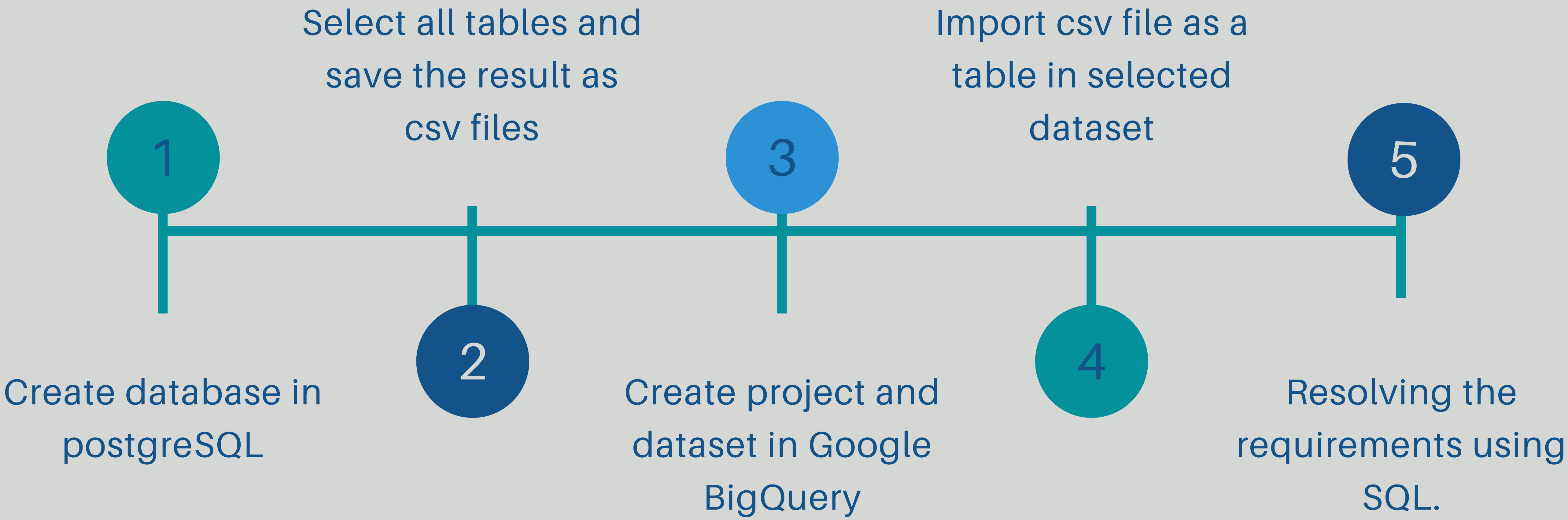
order_detail
id
customer_id
order_date
sku_id
price
qty_ordered
before_discount
discount_amount
after_discount
is_gross
is_valid
is_net
payment_id

customer_detail
id
registered_date

sku_detail
id
sku_name
base_price
cogs
category

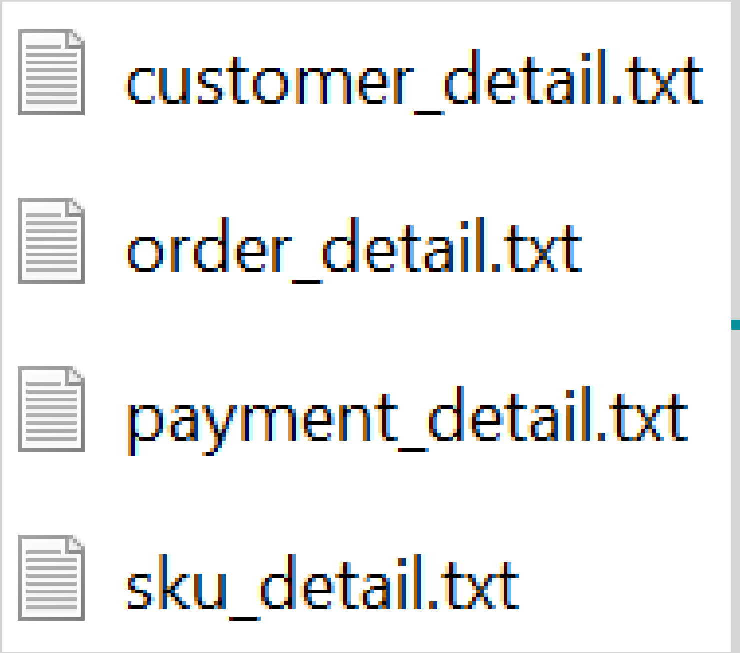
payment_detail
id
payment_method

Data Preparation



1. Create database in postgreSQL
2. Select all tables and save the result as csv files
3. Create project and dataset in Google BigQuery
4. Import csv file as a table in selected dataset
5. Resolving the requirements using SQL.

Data Preparation



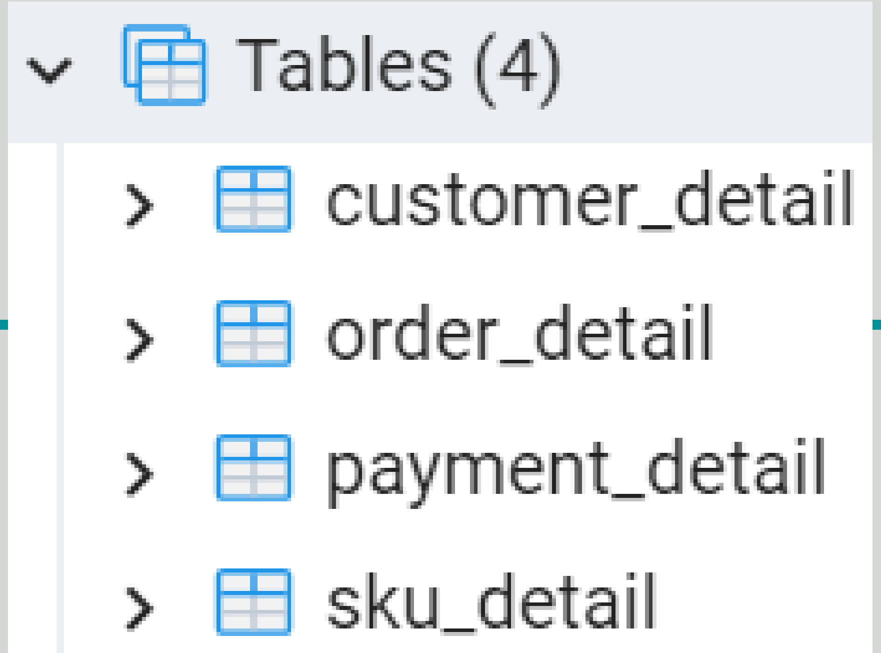
customer_detail.txt

order_detail.txt

payment_detail.txt

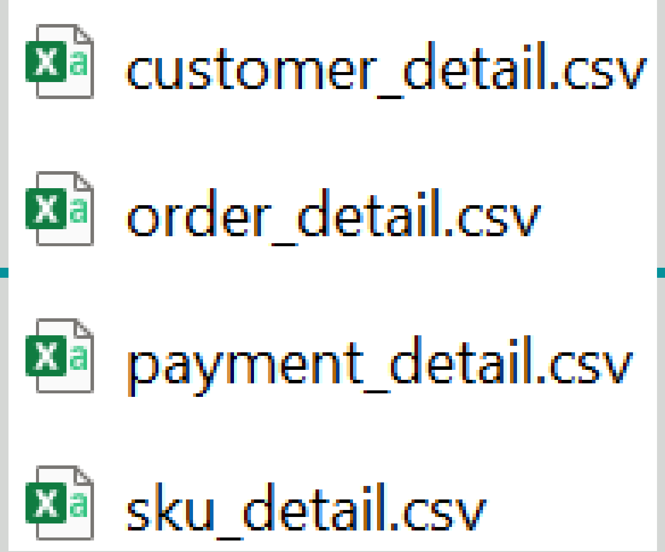
sku_detail.txt

PostgreSQL



Tables (4)

- customer_detail
- order_detail
- payment_detail
- sku_detail



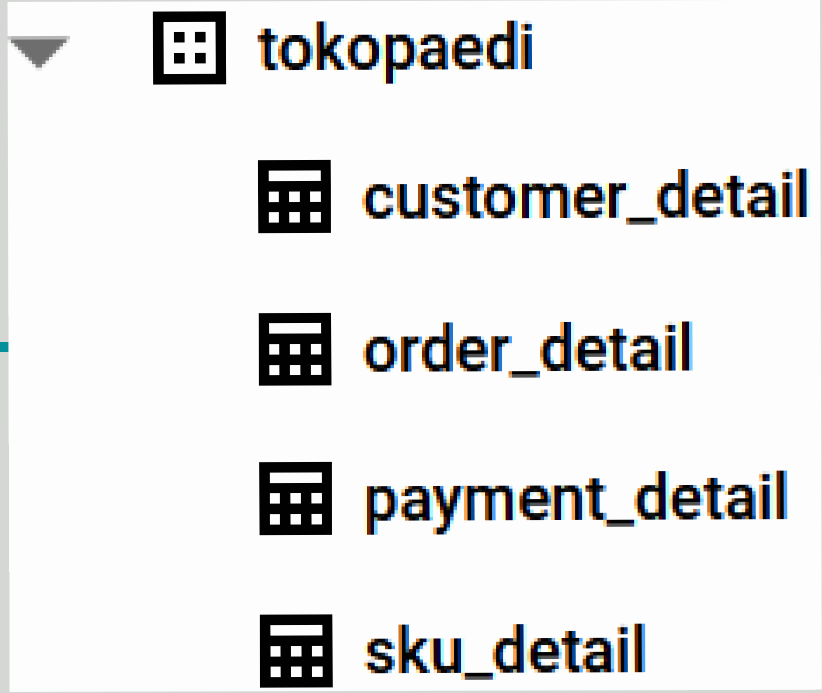
customer_detail.csv

order_detail.csv

payment_detail.csv

sku_detail.csv

Google BigQuery



tokopaedi

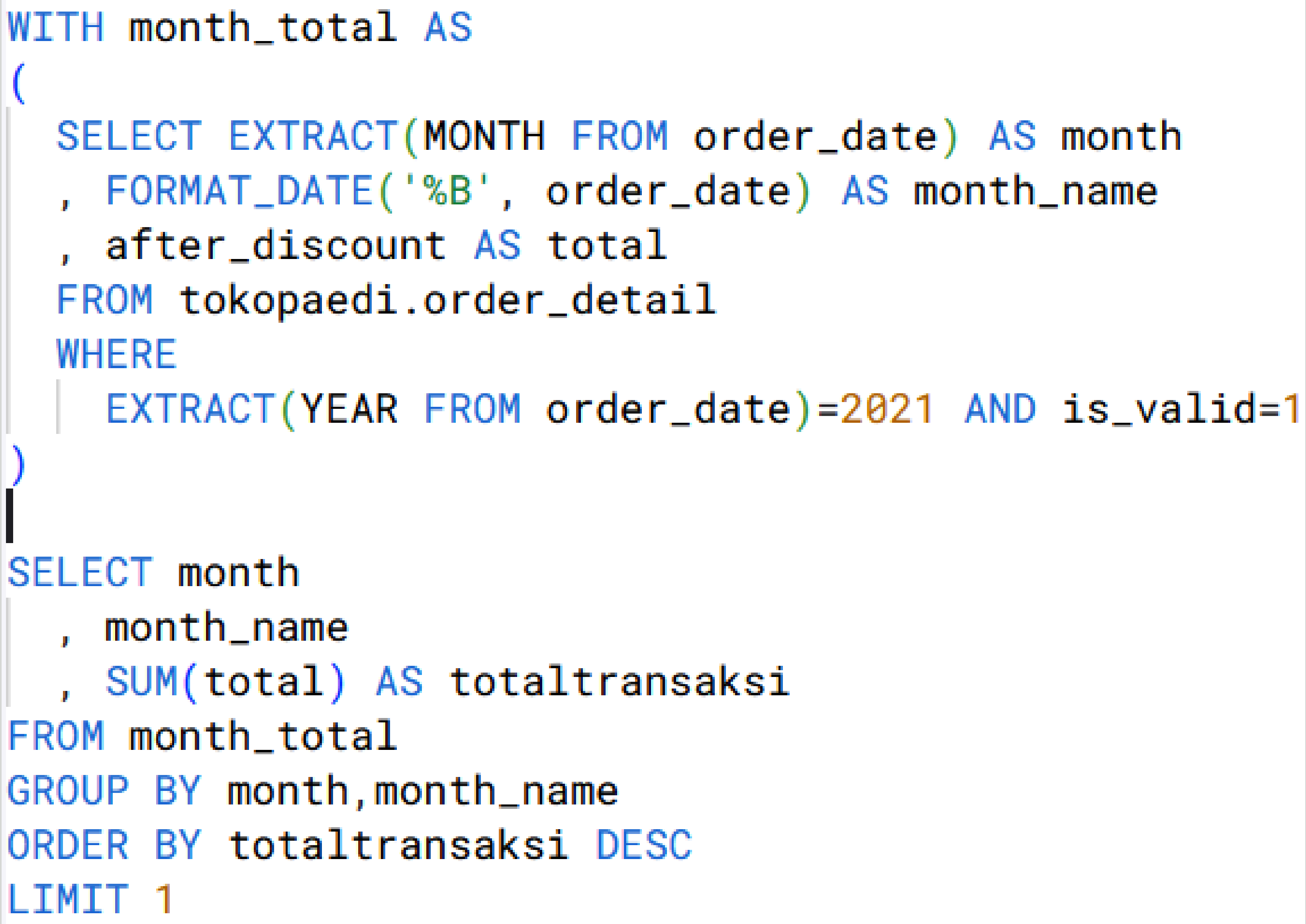
- customer_detail
- order_detail
- payment_detail
- sku_detail

Case Study

01

During the transactions that occurred during 2021, in which month did the total transaction value (after_discount) be the largest? Use is_valid = 1 to filter transaction data.

Query 01



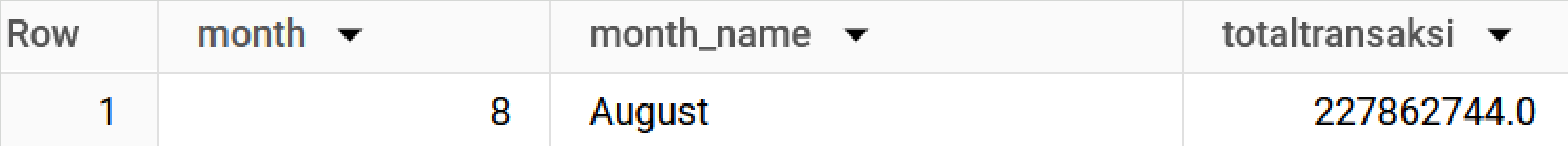
```
WITH month_total AS
(
  SELECT EXTRACT(MONTH FROM order_date) AS month
  , FORMAT_DATE('%B', order_date) AS month_name
  , after_discount AS total
  FROM tokopaedi.order_detail
  WHERE
    EXTRACT(YEAR FROM order_date)=2021 AND is_valid=1
)

SELECT month
  , month_name
  , SUM(total) AS totaltransaksi
FROM month_total
GROUP BY month,month_name
ORDER BY totaltransaksi DESC
LIMIT 1
```

CTE (month_total)

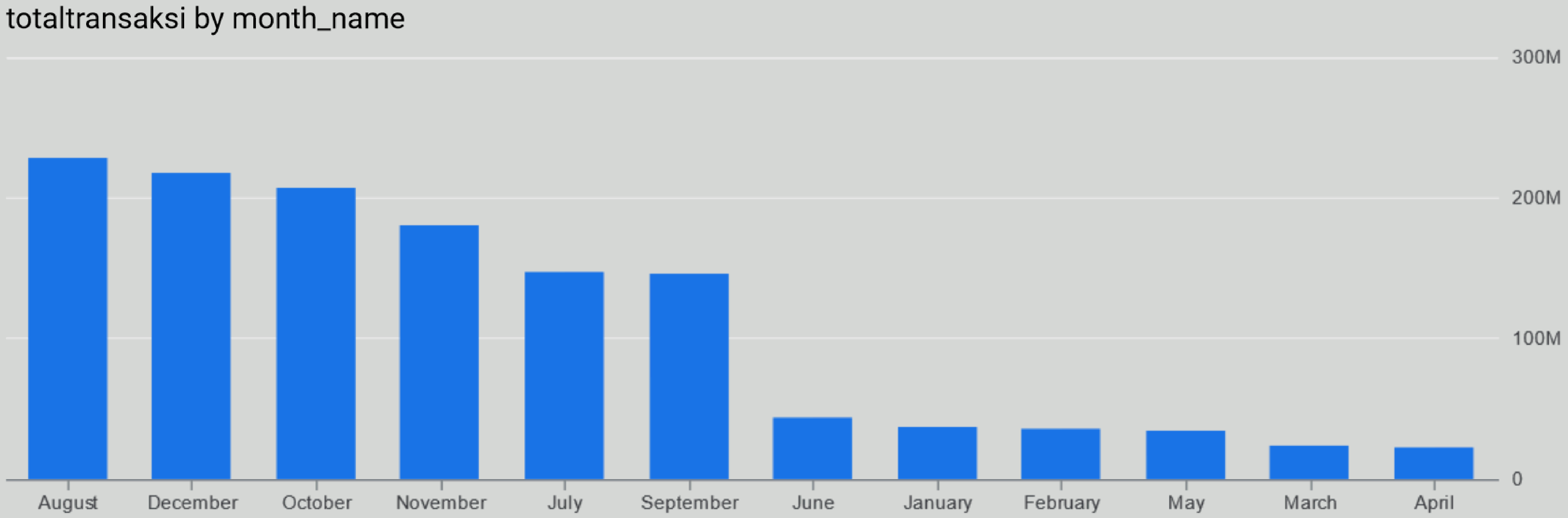
- select 'month' extracted from order_date, function FORMAT_DATE(%B, order_date) to get 'month name', after_discount as transaction value, named total
- filters : year=2021 (extracted from order_date), is_valid=1

- selecting from CTE : month, month name, summation of total per month, named totaltransaksi
- sort descending by totaltransaksi
- LIMIT 1 : show only 1 row

Result 01



Row	month	month_name	totaltransaksi
1	8	August	227862744.0



Insights

01

01 The highest transaction value in 2021 was in August at 227,862,744 dollars

02 Average transaction value is high in the 3rd and 4th quarters: July, August, Sept, Oct, Nov, Dec

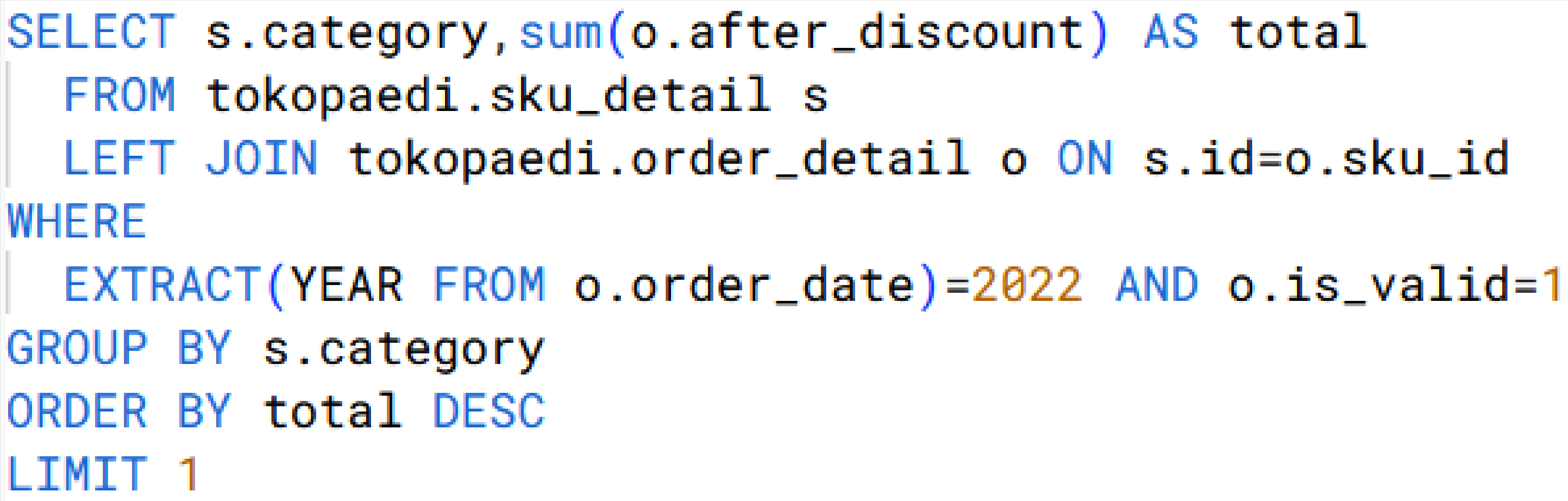
03 Identify best-sellers in the third and fourth quarters and make data-driven decisions to maximize profits

Case Study

02

During transactions in 2022, which category generated the largest transaction value? Use is_valid = 1 to filter transaction data.

Query 02

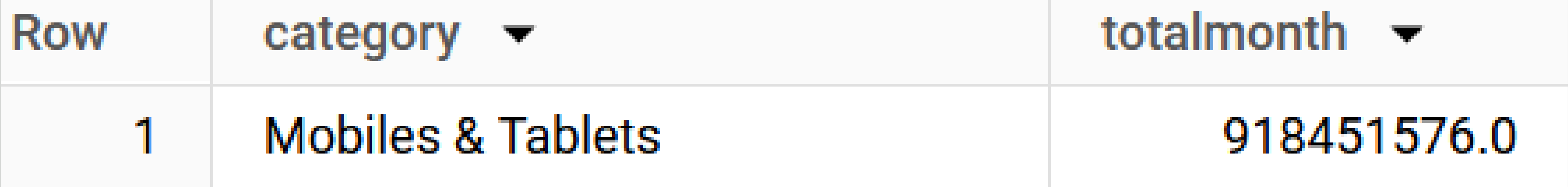


```
SELECT s.category,sum(o.after_discount) AS total
  FROM tokopaedi.sku_detail s
  LEFT JOIN tokopaedi.order_detail o ON s.id=o.sku_id
WHERE
  EXTRACT(YEAR FROM o.order_date)=2022 AND o.is_valid=1
GROUP BY s.category
ORDER BY total DESC
LIMIT 1
```

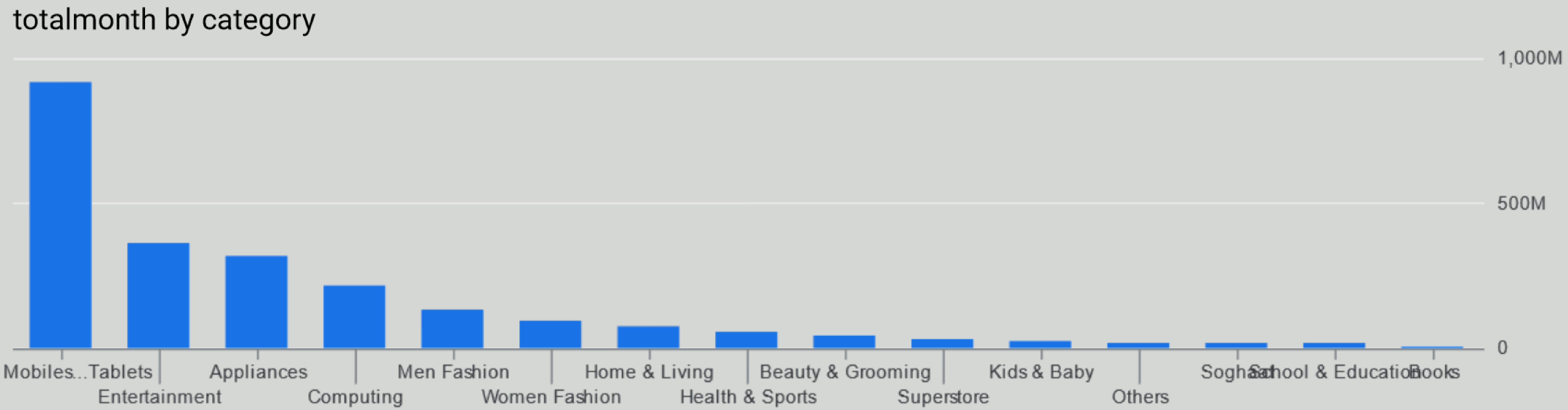
- select from sku_detail : category, summation of after_discount as transaction value per category, named total
- LEFT JOIN with order_detail on which sku_id=id (based on category in sku_detail)
- filters : year=2022 (extracted from order_date), is_valid=1
- sort descending by total
- LIMIT 1 : show only 1 row

Result

02



Row	category	totalmonth
1	Mobiles & Tablets	918451576.0



totalmonth by category

Insights

02

01 During 2022, the highest transaction value was sales in the mobile and tablets category of 918,451,576 dollars.

02 In 2022, high demand tends to be for technology or digital products such as mobile and tablets, computer or laptop, other entertainment stuff.

03 Identify best-sellers of technology product and make data-driven decisions to maximize profits.

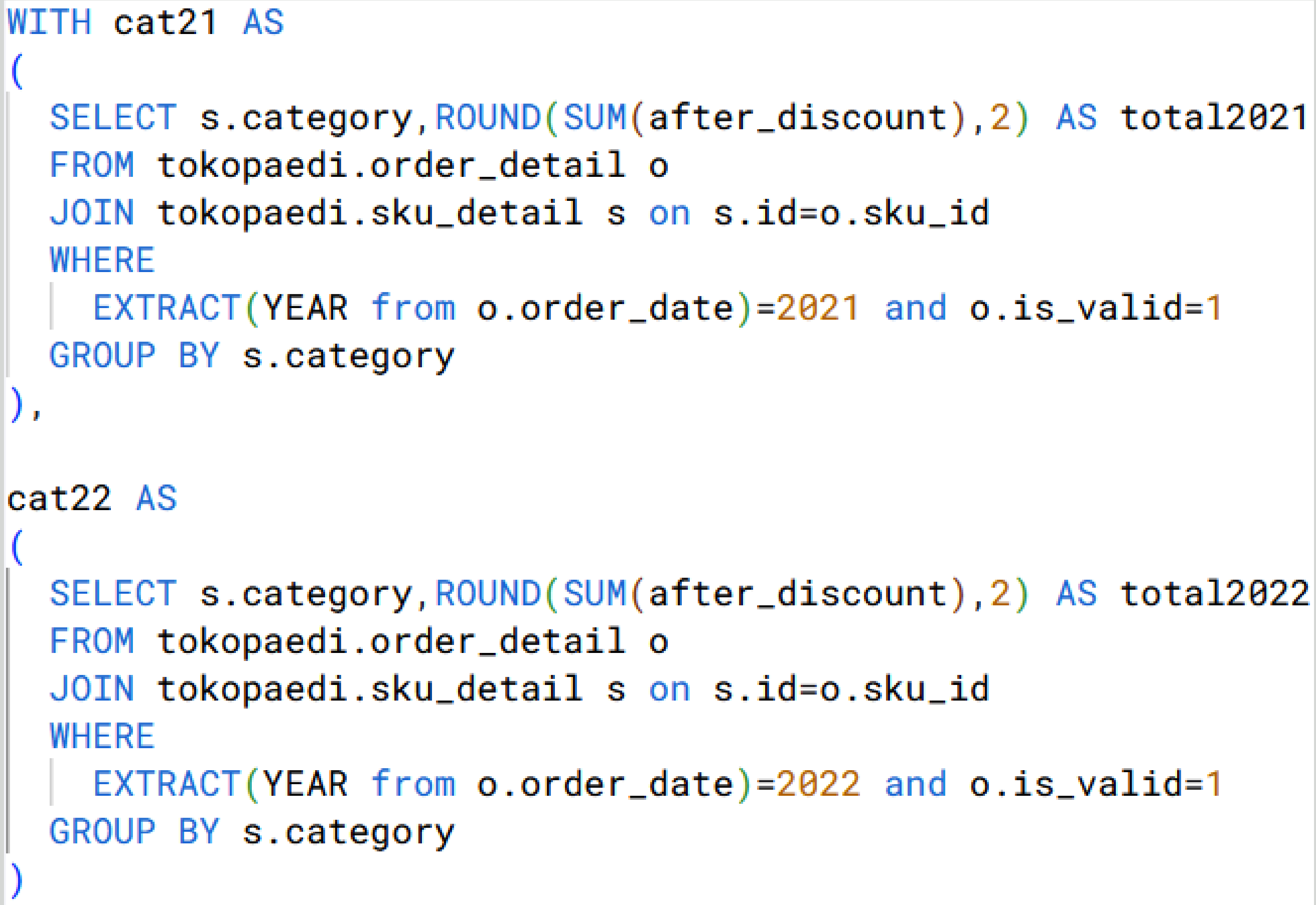
Case Study

03

Compare the transaction value of each category in 2021 with 2022. State which categories experienced an increase and which categories experienced a decrease in transaction value from 2021 to 2022. Use is_valid = 1 to filter transaction data.

Query

03



```sql
WITH cat21 AS
(
  SELECT s.category,ROUND(SUM(after_discount),2) AS total2021
  FROM tokopaedi.order_detail o
  JOIN tokopaedi.sku_detail s on s.id=o.sku_id
  WHERE
    EXTRACT(YEAR from o.order_date)=2021 and o.is_valid=1
  GROUP BY s.category
),

cat22 AS
(
  SELECT s.category,ROUND(SUM(after_discount),2) AS total2022
  FROM tokopaedi.order_detail o
  JOIN tokopaedi.sku_detail s on s.id=o.sku_id
  WHERE
    EXTRACT(YEAR from o.order_date)=2022 and o.is_valid=1
  GROUP BY s.category
)
```

CTE (cat21)

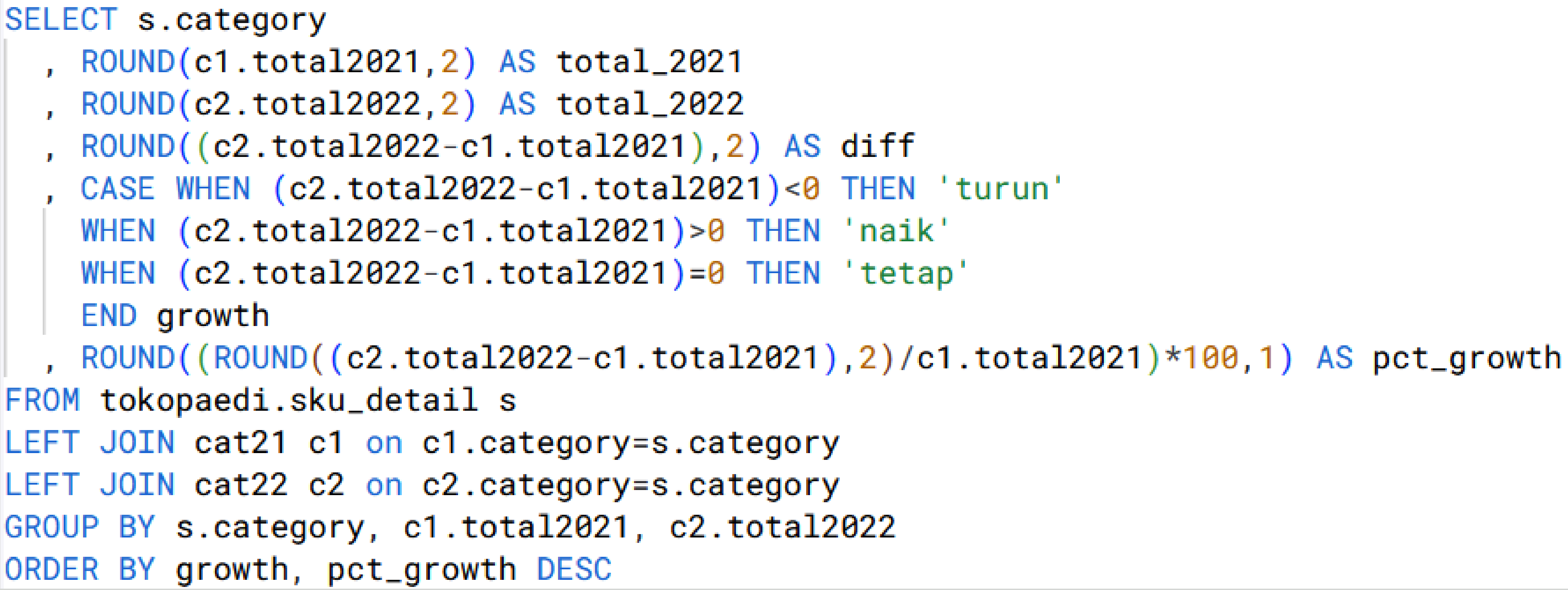
- JOIN sku_detail with order_detail on which sku_id of order_detail=id of sku_detail
- select category, summation of after_discount as transaction value per category, named total2021
- ROUND 2 digits after the comma of total2021
- filters : year=2021 (extracted from order_date), is_valid=1

CTE (cat22)

- JOIN sku_detail with order_detail on which sku_id of order_detail=id of sku_detail
- select category, summation of after_discount as transaction value per category, named total2022
- ROUND 2 digits after the comma of total2022
- filters : year=2022 (extracted from order_date), is_valid=1

Query

03

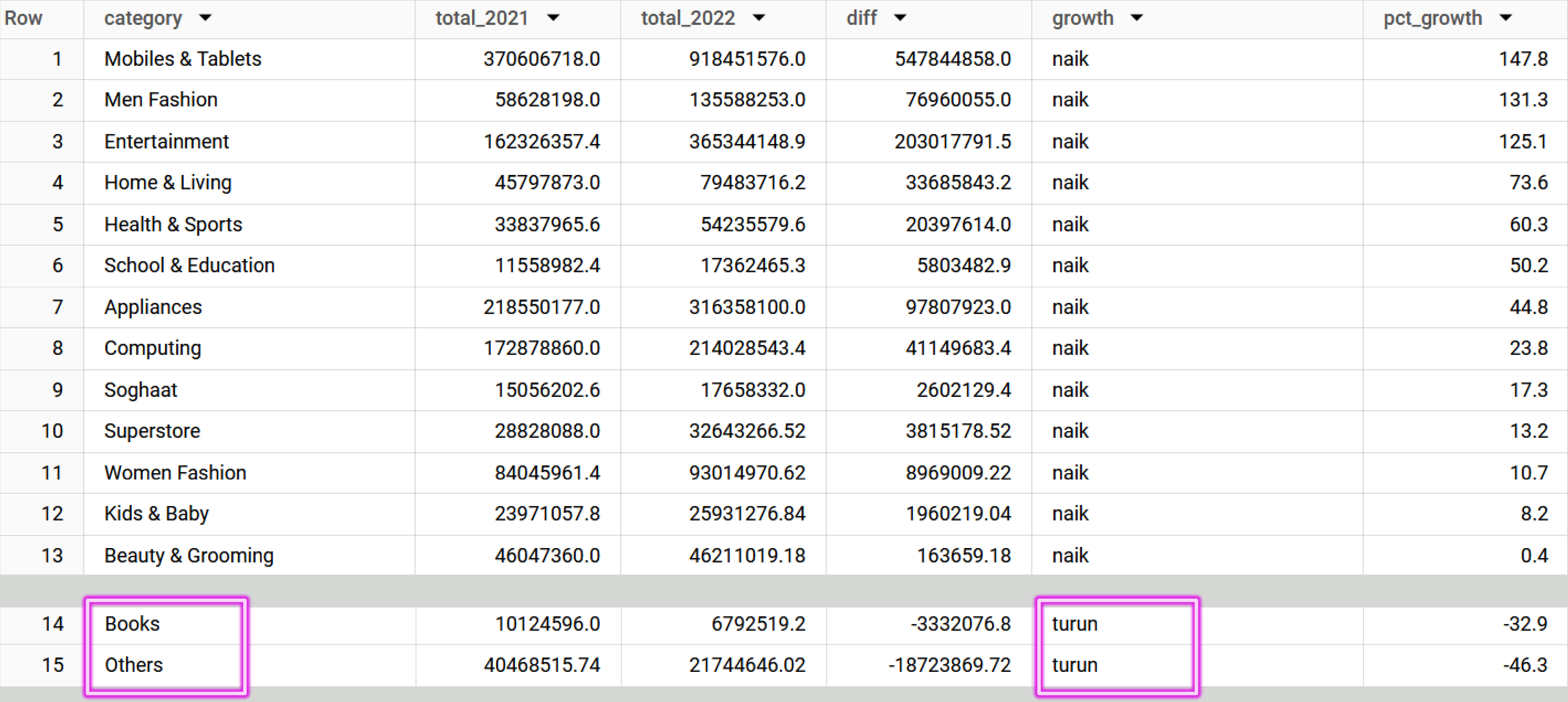


```sql
SELECT s.category
  , ROUND(c1.total2021,2) AS total_2021
  , ROUND(c2.total2022,2) AS total_2022
  , ROUND((c2.total2022-c1.total2021),2) AS diff
  , CASE WHEN (c2.total2022-c1.total2021)<0 THEN 'turun'
    WHEN (c2.total2022-c1.total2021)>0 THEN 'naik'
    WHEN (c2.total2022-c1.total2021)=0 THEN 'tetap'
    END growth
  , ROUND((ROUND((c2.total2022-c1.total2021),2)/c1.total2021)*100,1) AS pct_growth
FROM tokopaedi.sku_detail s
LEFT JOIN cat21 c1 on c1.category=s.category
LEFT JOIN cat22 c2 on c2.category=s.category
GROUP BY s.category, c1.total2021, c2.total2022
ORDER BY growth, pct_growth DESC
```

- LEFT JOIN sku_detail with CTE cat21 and cat22 on which category of sku_detail=category of both cat21 and cat22
- select category from sku_detail, total2021, total2022, subtraction of total2022 and total2021 to show the difference value per category named diff, using conditional statement CASE WHEN to mention the diff whether turun/naik/tetap named growth, calculate the diff as percentage named pct_growth, ROUND 2 digits after the comma of pct_growth
- sort descending by growth and pct_growth

Result

03



Row	category	total_2021	total_2022	diff	growth	pct_growth
1	Mobiles & Tablets	370606718.0	918451576.0	547844858.0	naik	147.8
2	Men Fashion	58628198.0	135588253.0	76960055.0	naik	131.3
3	Entertainment	162326357.4	365344148.9	203017791.5	naik	125.1
4	Home & Living	45797873.0	79483716.2	33685843.2	naik	73.6
5	Health & Sports	33837965.6	54235579.6	20397614.0	naik	60.3
6	School & Education	11558982.4	17362465.3	5803482.9	naik	50.2
7	Appliances	218550177.0	316358100.0	97807923.0	naik	44.8
8	Computing	172878860.0	214028543.4	41149683.4	naik	23.8
9	Soghaat	15056202.6	17658332.0	2602129.4	naik	17.3
10	Superstore	28828088.0	32643266.52	3815178.52	naik	13.2
11	Women Fashion	84045961.4	93014970.62	8969009.22	naik	10.7
12	Kids & Baby	23971057.8	25931276.84	1960219.04	naik	8.2
13	Beauty & Grooming	46047360.0	46211019.18	163659.18	naik	0.4
14	Books	10124596.0	6792519.2	-3332076.8	turun	-32.9
15	Others	40468515.74	21744646.02	-18723869.72	turun	-46.3

Insights

03

01 There are 13 categories that experienced an increase in transaction value and 2 categories that experienced a decrease in transaction value from 2021 to 2022.

02 The most significant increase in sales is Mobile and Tablets category by 147,8%. The smallest increase in sales is Beauty & Grooming, which only increased by 0,4%.

03 Categories that experienced a decrease in sales are Books and Others. The one whose sales fell the most is Others which fell by almost half or 46,3%.

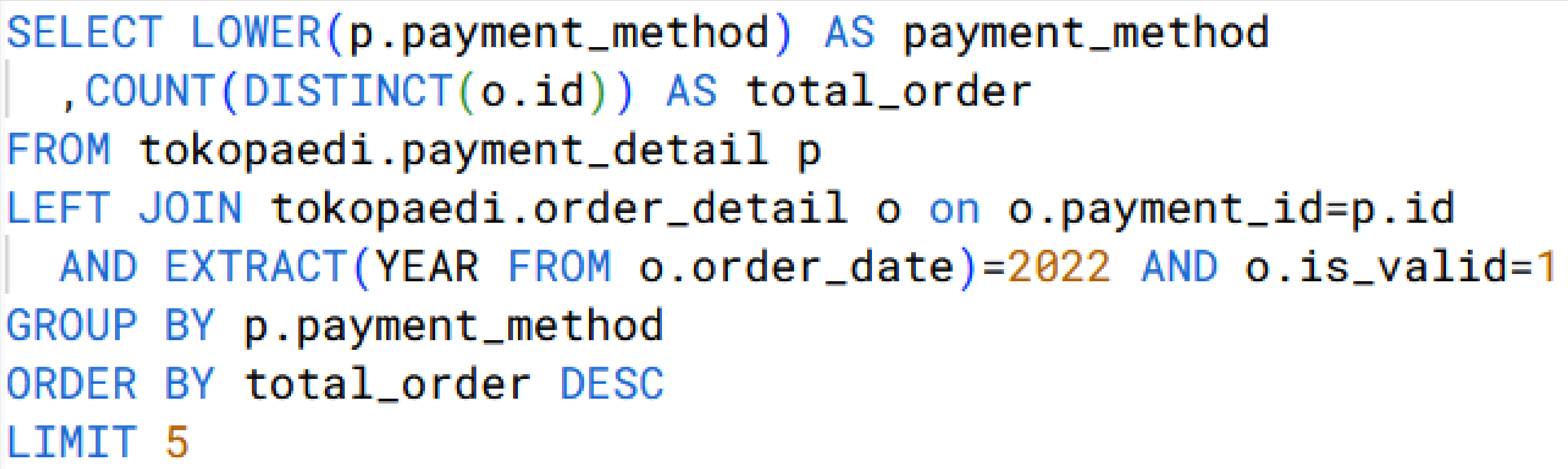
04 Increasing sales next year focus on understanding customer needs, refining product offerings, refining the customer experience, and leveraging data-driven insights. Are the Books and Others categories still worth selling in store?

Case Study

04

Show the top 5 most popular payment methods used during 2022 (based on total unique orders). Use is_valid = 1 to filter transaction data.

Query 04

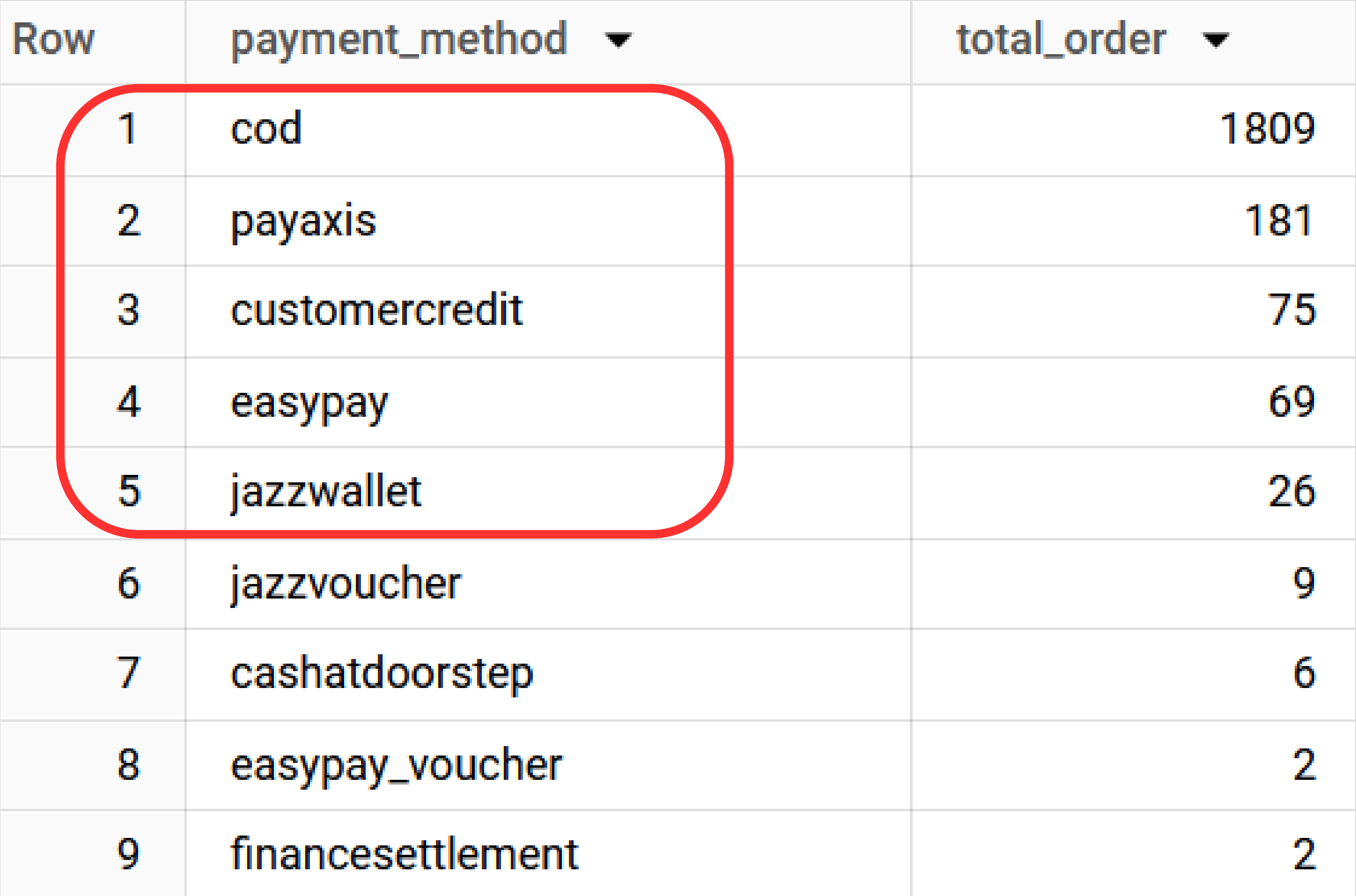


```sql
SELECT LOWER(p.payment_method) AS payment_method
    ,COUNT(DISTINCT(o.id)) AS total_order
FROM tokopaedi.payment_detail p
LEFT JOIN tokopaedi.order_detail o on o.payment_id=p.id
    AND EXTRACT(YEAR FROM o.order_date)=2022 AND o.is_valid=1
GROUP BY p.payment_method
ORDER BY total_order DESC
LIMIT 5
```

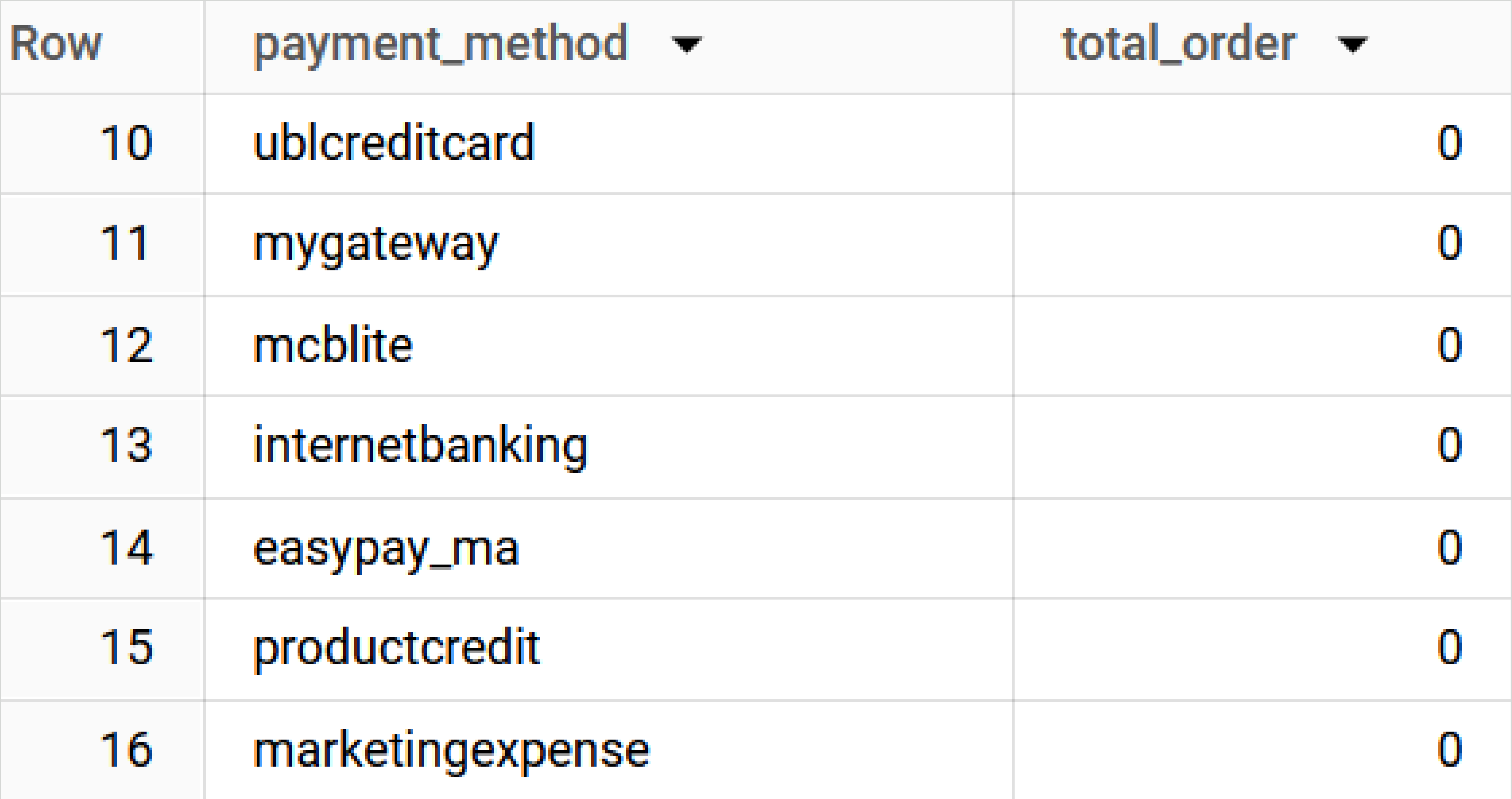
- select from payment_detail : lower string of payment_method, COUNT(DISTINCT id) to sum up order id of transaction per payment_method, named total_order
- LEFT JOIN with order_detail on which payment_id=id and filtering the transactions in order_detail during 2022, is_valid=1 (based on payment_method in payment_detail)
- sort descending by total_order
- LIMIT 5 : show only 5 rows

Result

04



Row	payment_method	total_order
1	cod	1809
2	payaxis	181
3	customercredit	75
4	easypay	69
5	jazzwallet	26
6	jazzvoucher	9
7	cashatdoorstep	6
8	easypay_voucher	2
9	financesettlement	2



Row	payment_method	total_order
10	ublcreditcard	0
11	mygateway	0
12	mcblite	0
13	internetbanking	0
14	easypay_ma	0
15	productcredit	0
16	marketingexpense	0

Insights

04

01 Out of 16 payment_methods, there are 9 payment_methods used for valid transaction in 2022

02 COD is the most widely used payment_method, there are 1809 transactions paid using the COD method

03 Top 5 payment_methods include: cod, payaxis, customercredit, easypay, jazzwallet

04 Evaluating user experience using cashless methods such as payaxis, customercredit, easypay, jazzwallet. Are there certain factors that cause buyers to prefer COD?

Case Study

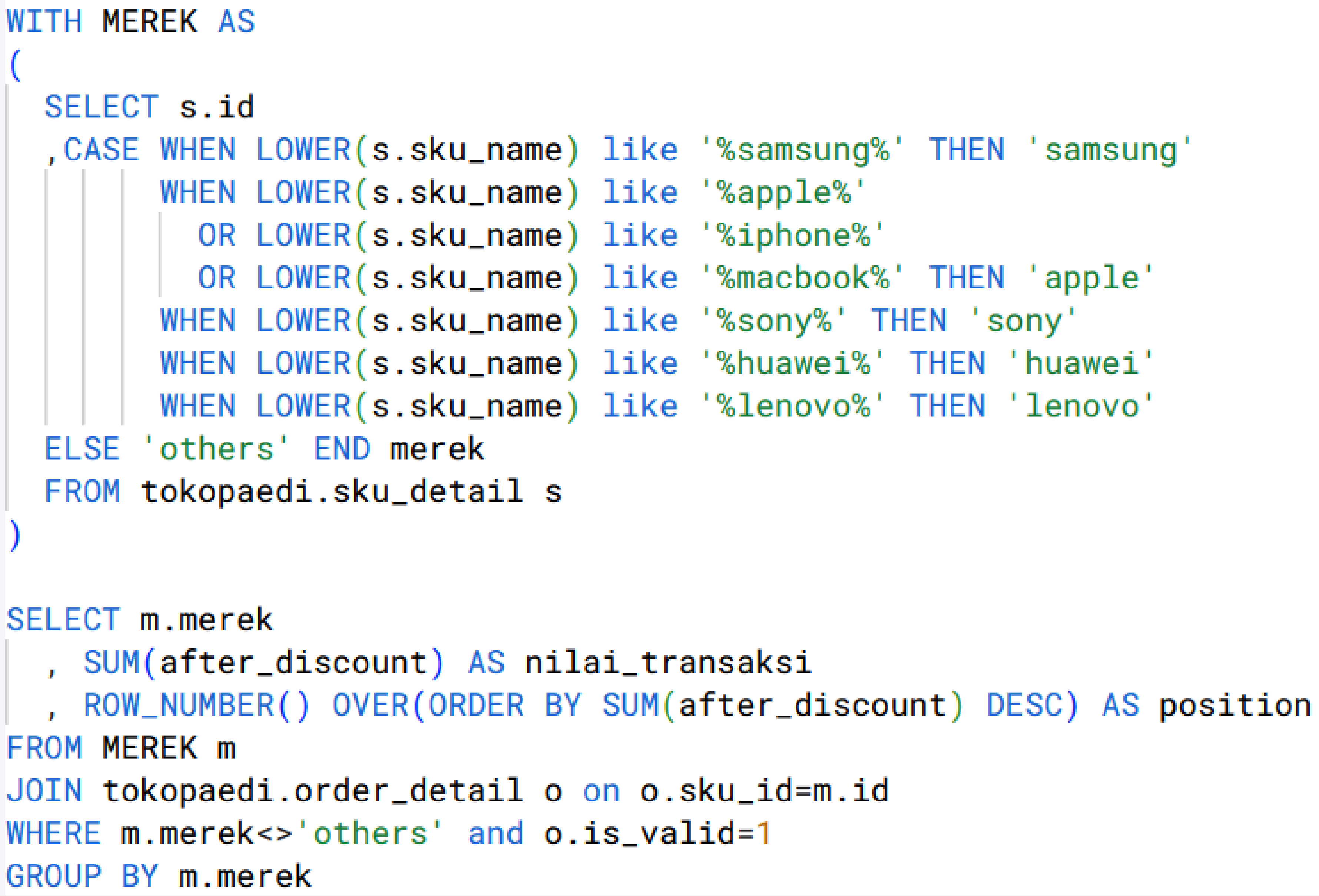
05

Sort these 5 products by their transaction value.

1. Samsung
2. Apple
3. Sony
4. Huawei
5. Lenovo

Use is_valid = 1 to filter transaction data.

Query 05



```sql
WITH MEREK AS
(
  SELECT s.id
  ,CASE WHEN LOWER(s.sku_name) like '%samsung%' THEN 'samsung'
        WHEN LOWER(s.sku_name) like '%apple%'
          OR LOWER(s.sku_name) like '%iphone%'
          OR LOWER(s.sku_name) like '%macbook%' THEN 'apple'
        WHEN LOWER(s.sku_name) like '%sony%' THEN 'sony'
        WHEN LOWER(s.sku_name) like '%huawei%' THEN 'huawei'
        WHEN LOWER(s.sku_name) like '%lenovo%' THEN 'lenovo'
  ELSE 'others' END merek
  FROM tokopaedi.sku_detail s
)

SELECT m.merek
  , SUM(after_discount) AS nilai_transaksi
  , ROW_NUMBER() OVER(ORDER BY SUM(after_discount) DESC) AS position
FROM MEREK m
JOIN tokopaedi.order_detail o on o.sku_id=m.id
WHERE m.merek<>'others' and o.is_valid=1
GROUP BY m.merek
```

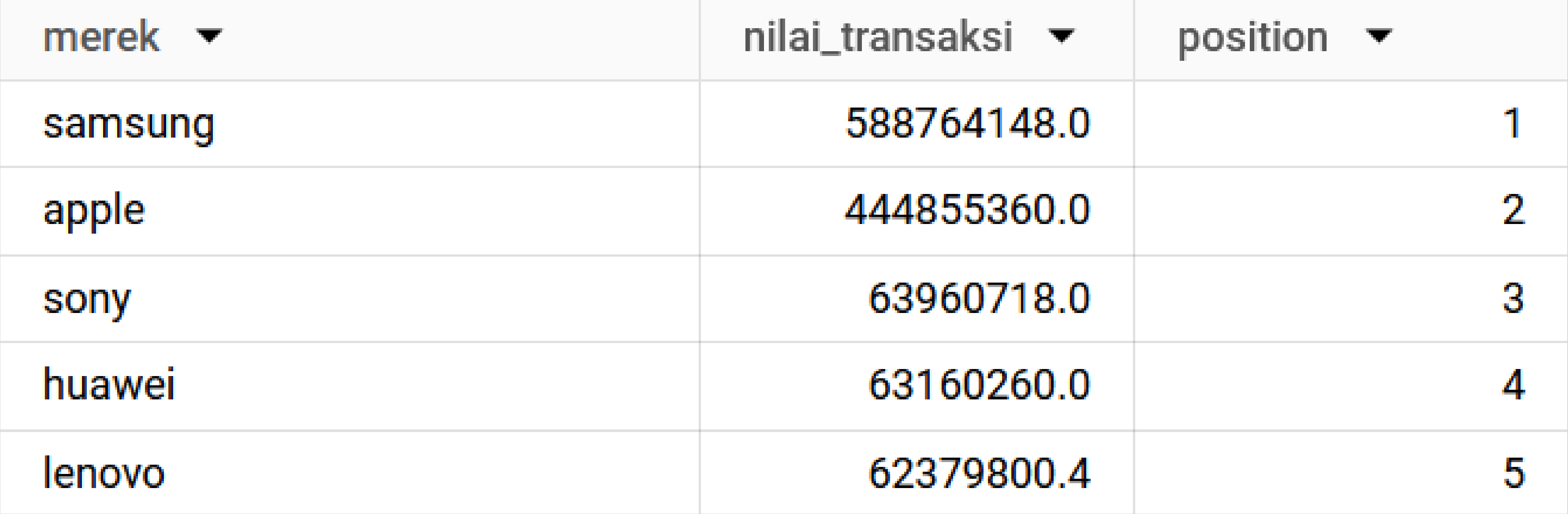
CTE (MEREK)

- select from sku_detail : id, function LOWER() to lower string of sku_name, using conditional statement CASE WHEN to seek some words stored in sku_name whether the sku branded samsung/apple/sony/huawei/lenovo/others named 'merek'

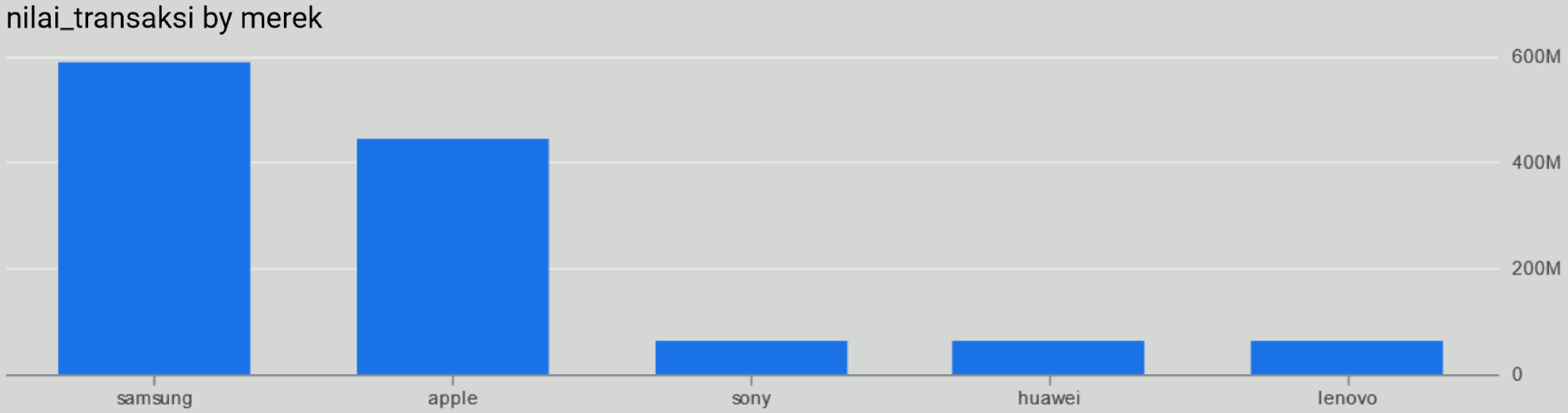
- CTE joined with order_detail on which sku_id=id
- selecting : merek, summation after_discount as transaction value per merek, named nilai_transaksi, labeling the row with number sorted descending by nilai_transaksi to show the position value of merek
- filters : merek excluded 'others', is_valid=1

Result

05



merek ▾	nilai_transaksi ▾	position ▾
samsung	588764148.0	1
apple	444855360.0	2
sony	63960718.0	3
huawei	63160260.0	4
lenovo	62379800.4	5



Insights

05

01 For 2 years, from 2021 to 2022, the Samsung brand dominated sales by 588,764,148 dollars. With quality that is not inferior to the Apple brand, Samsung's price is more affordable than Apple's, making it a major buying attraction.

02 Lenovo became the brand that obtained the smallest transaction value by 62,379,800.4 dollars. Although Sony, Huawei, and Lenovo achieved comparable transaction values among themselves, their numbers still fall significantly below those of Samsung and Apple.

03 This gap highlights the competitive challenge faced by these companies in reaching the market dominance and consumer loyalty that Samsung and Apple currently enjoy. However, with strategic product development and innovative marketing, they have the potential to capture a larger share of the market.

THANK YOU!

Connect with me

endahen12@gmail.com

www.linkedin.com/in/endahrakhmawati