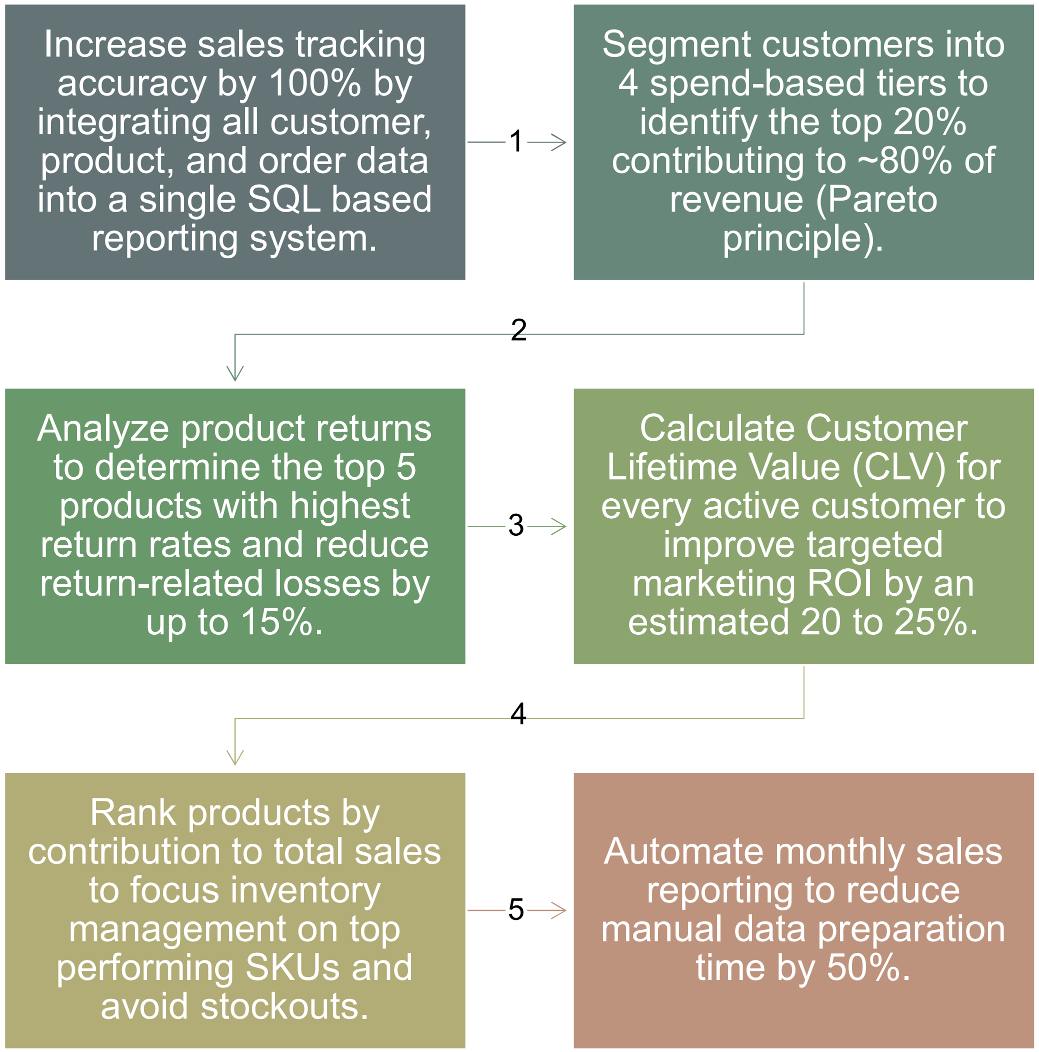
RETAIL ANALYTICS PROJECT

SQL IN ACTION

DERIVING BUSINESS INSIGHTS FROM ORDERS, SALES, CUSTOMERS AND RETURNS DATA USING ADVANCED SQL TECHNIQUES

Gurleen Kaur
Aspiring Data Analyst

PROJECT OBJECTIVES



Increase sales tracking accuracy by 100% by integrating all customer, product, and order data into a single SQL based reporting system.

Segment customers into 4 spend-based tiers to identify the top 20% contributing to ~80% of revenue (Pareto principle).

Analyze product returns to determine the top 5 products with highest return rates and reduce return-related losses by up to 15%.

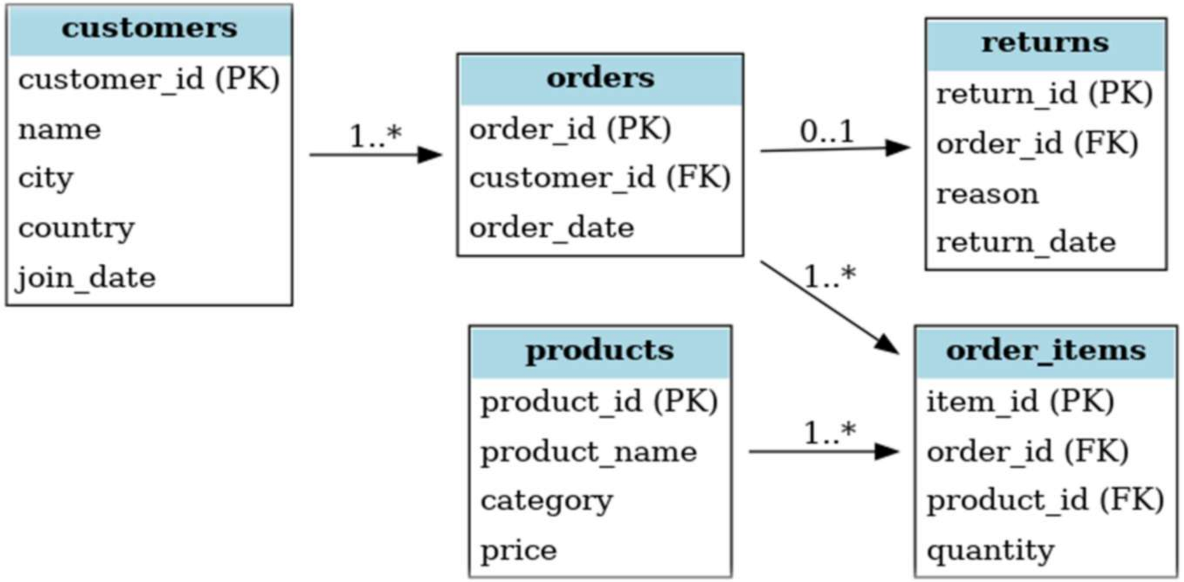
Calculate Customer Lifetime Value (CLV) for every active customer to improve targeted marketing ROI by an estimated 20 to 25%.

Rank products by contribution to total sales to focus inventory management on top performing SKUs and avoid stockouts.

Automate monthly sales reporting to reduce manual data preparation time by 50%.

DATABASE SCHEMA OVERVIEW

ENTITY RELATIONSHIP DIAGRAM (ERD)



customers
customer_id (PK)
name
city
country
join_date

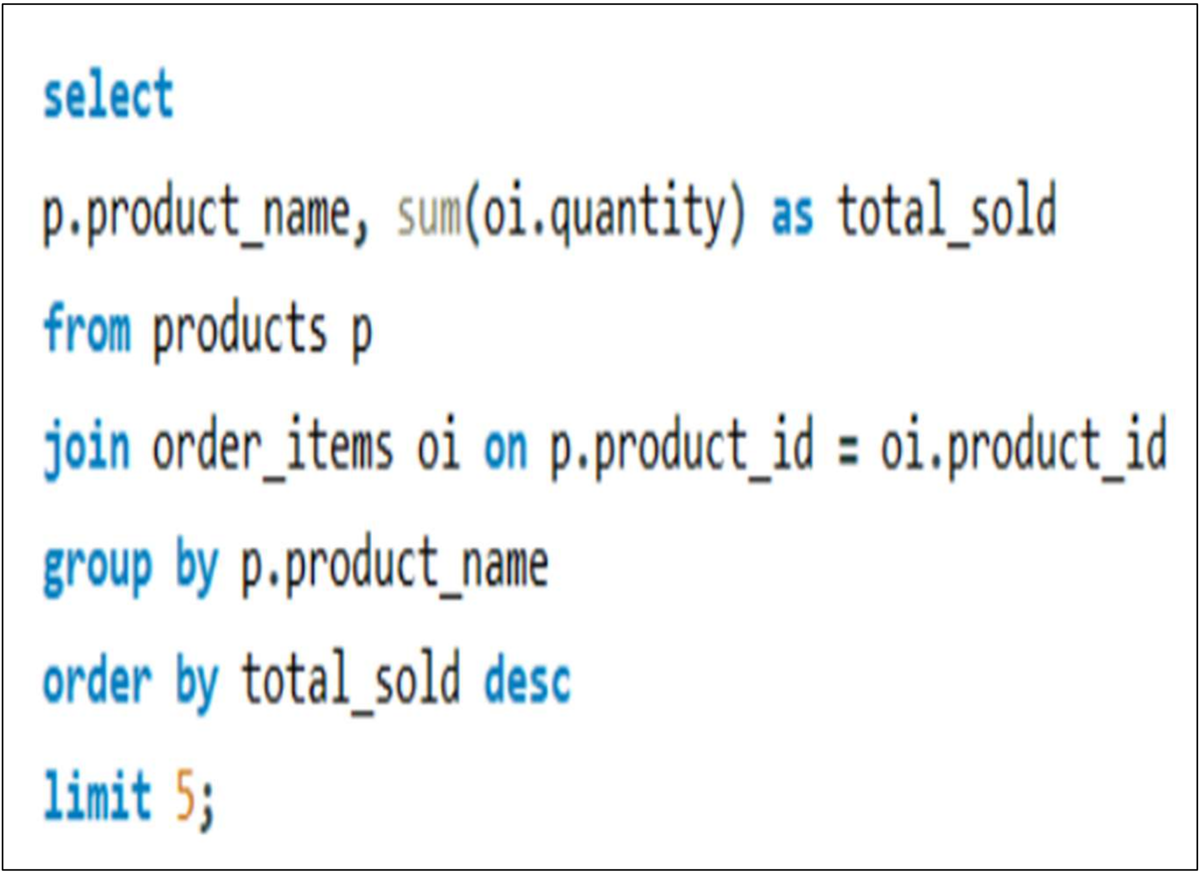
orders
order_id (PK)
customer_id (FK)
order_date

returns
return_id (PK)
order_id (FK)
reason
return_date

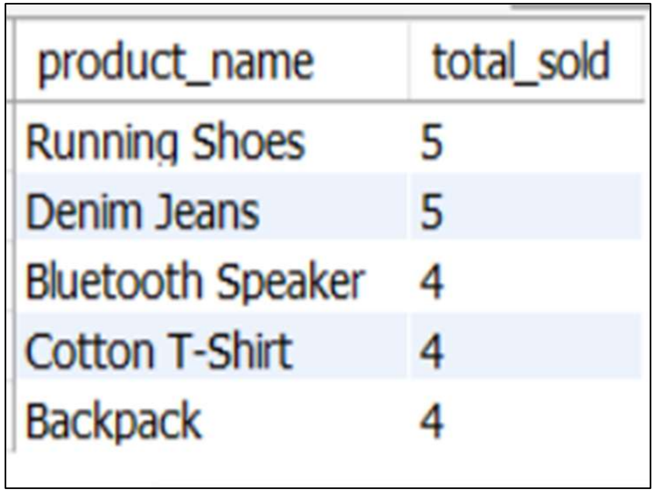
products
product_id (PK)
product_name
category
price

order_items
item_id (PK)
order_id (FK)
product_id (FK)
quantity

Top 5 Best-Selling Products – June 2024



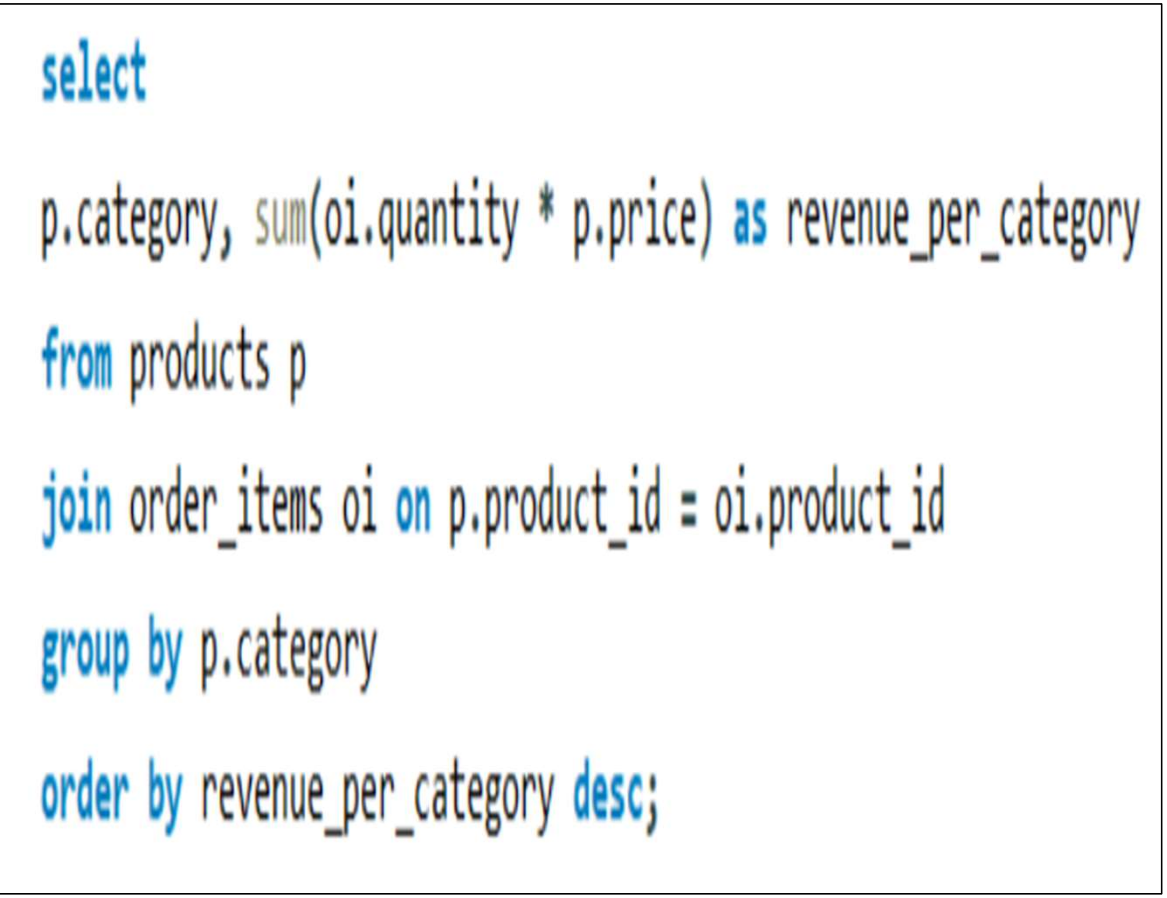
```
select
p.product_name, sum(oi.quantity) as total_sold
from products p
join order_items oi on p.product_id = oi.product_id
group by p.product_name
order by total_sold desc
limit 5;
```



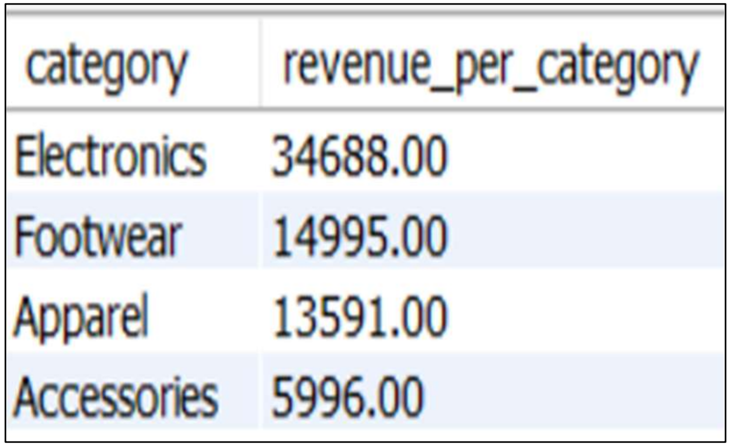
product_name	total_sold
Running Shoes	5
Denim Jeans	5
Bluetooth Speaker	4
Cotton T-Shirt	4
Backpack	4

- Top 5 products driving bulk of sales volume in June
- Key focus for inventory & marketing push
- Strong indicators for demand forecasting

Revenue by Product Category



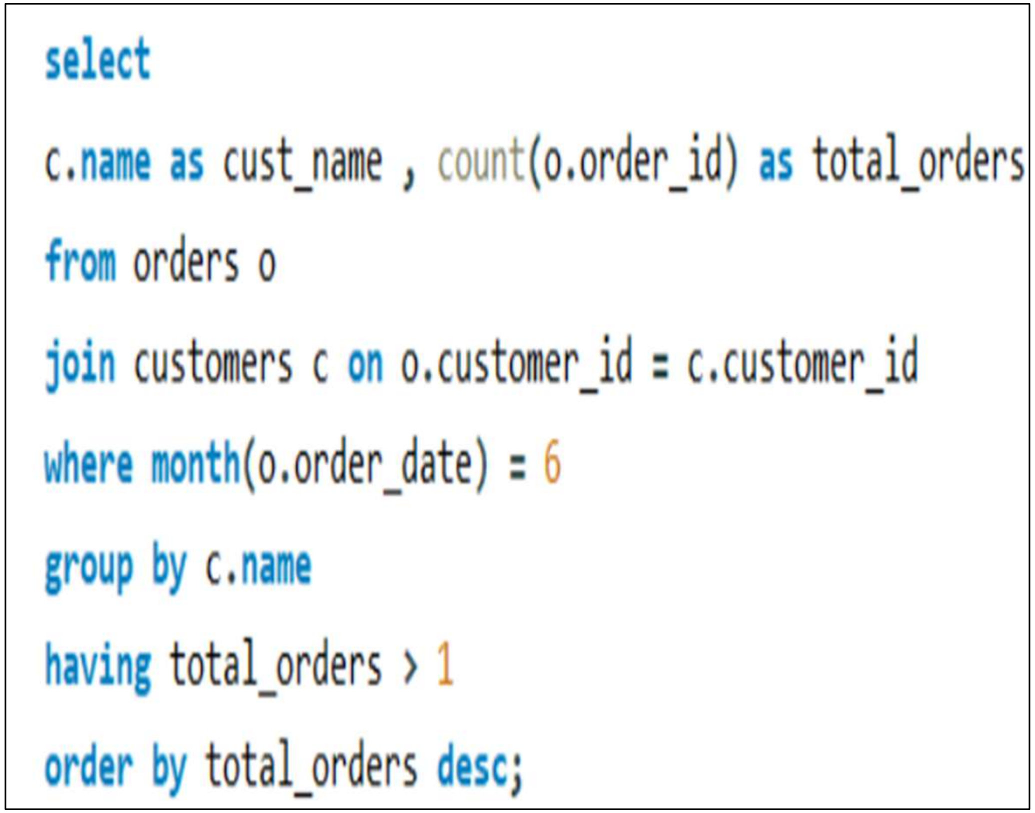
```
select

p.category, sum(oi.quantity * p.price) as revenue_per_category

from products p

join order_items oi on p.product_id = oi.product_id

group by p.category

order by revenue_per_category desc;
```



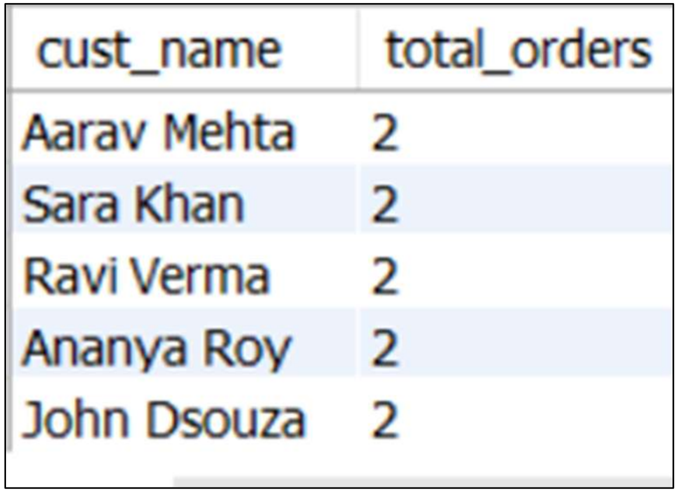
category	revenue_per_category
Electronics	34688.00
Footwear	14995.00
Apparel	13591.00
Accessories	5996.00

- Total revenue snapshot for June sales
- Helps measure business performance for the month.

Customer with more than 1 Orders



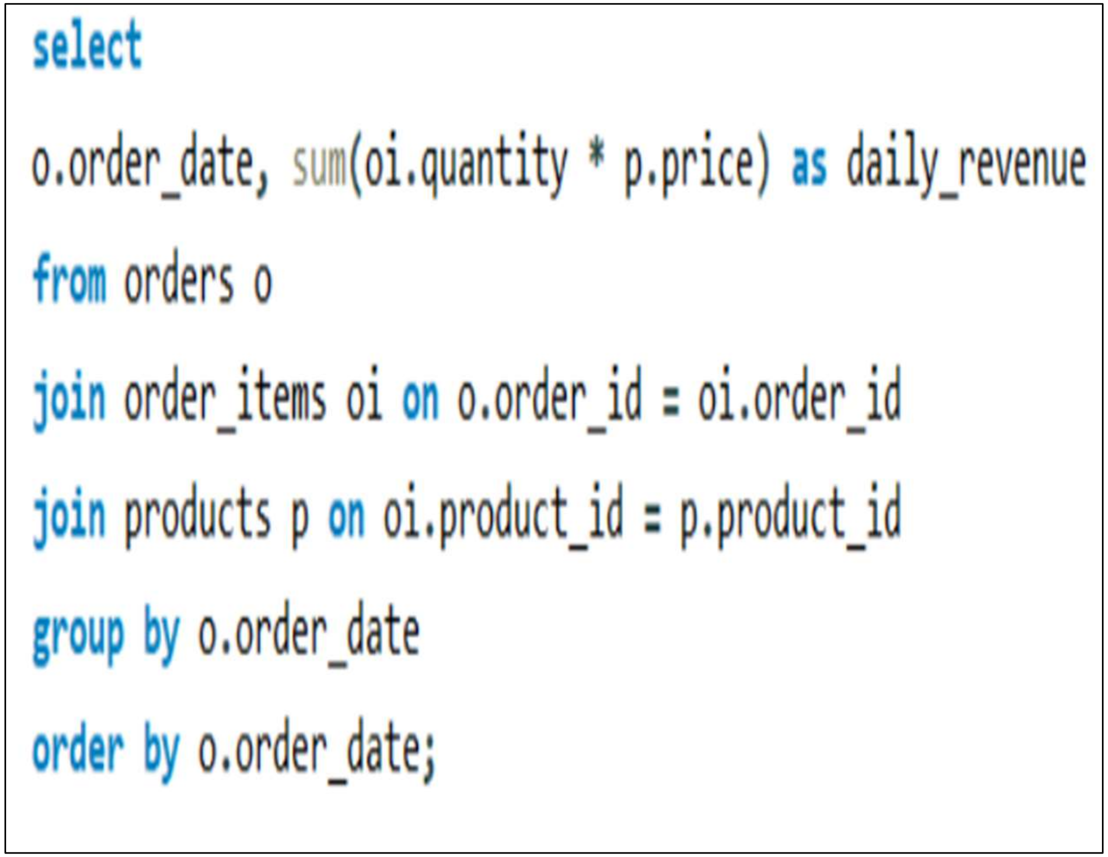
```
select
c.name as cust_name , count(o.order_id) as total_orders
from orders o
join customers c on o.customer_id = c.customer_id
where month(o.order_date) = 6
group by c.name
having total_orders > 1
order by total_orders desc;
```



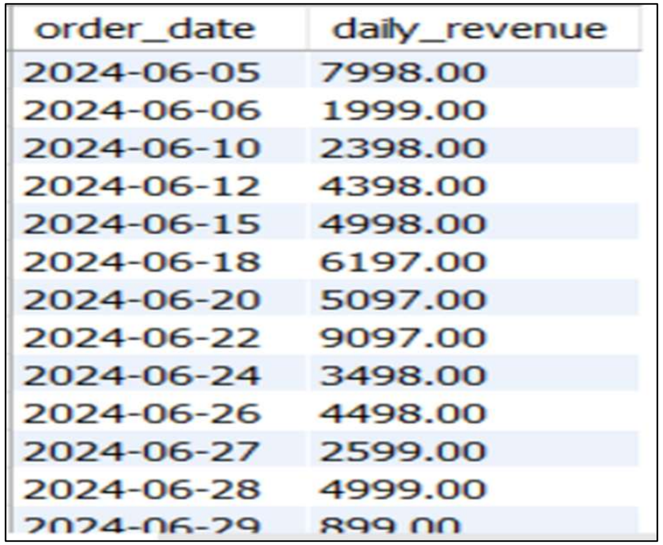
cust_name	total_orders
Aarav Mehta	2
Sara Khan	2
Ravi Verma	2
Ananya Roy	2
John Dsouza	2

- **50%**of customers placed more than one order in June
- These repeat customers contributed to **~61%** of total revenue
- High-value segment for loyalty programs & retention campaigns

Daily Sales Trend for June



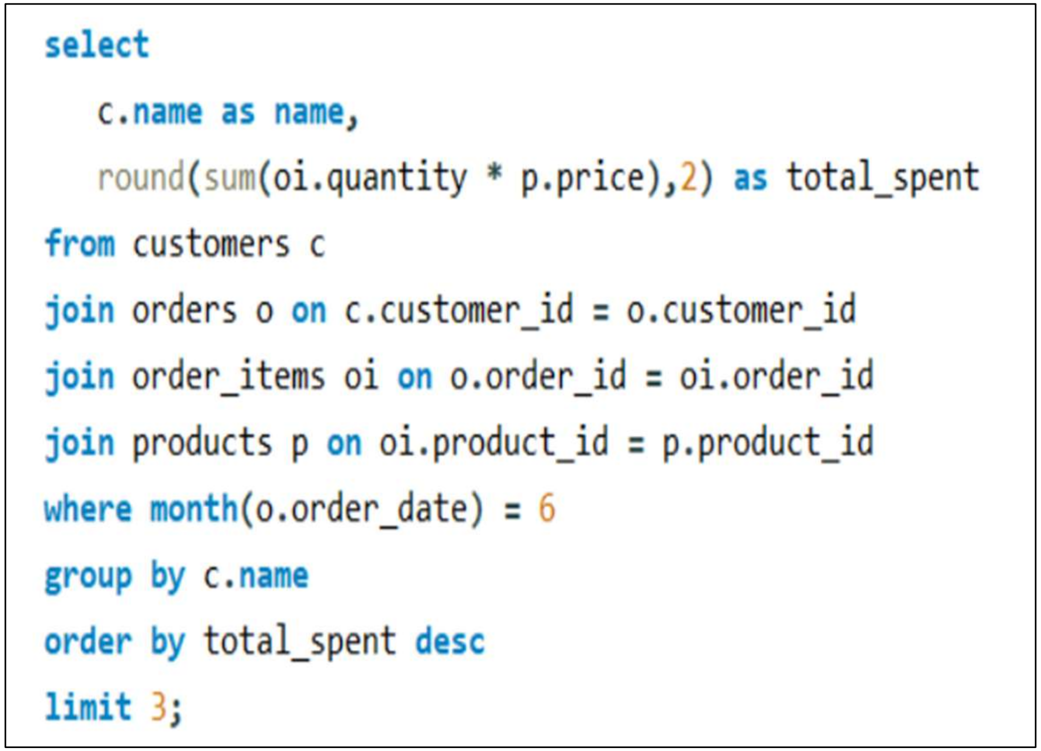
```
select

o.order_date, sum(oi.quantity * p.price) as daily_revenue

from orders o

join order_items oi on o.order_id = oi.order_id

join products p on oi.product_id = p.product_id

group by o.order_date

order by o.order_date;
```



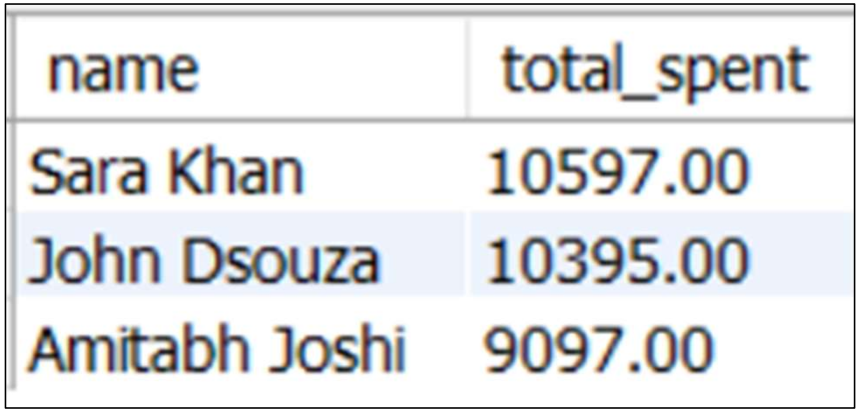
order_date	daily_revenue
2024-06-05	7998.00
2024-06-06	1999.00
2024-06-10	2398.00
2024-06-12	4398.00
2024-06-15	4998.00
2024-06-18	6197.00
2024-06-20	5097.00
2024-06-22	9097.00
2024-06-24	3498.00
2024-06-26	4498.00
2024-06-27	2599.00
2024-06-28	4999.00
2024-06-29	899.00

- **Peak sales day**: June 19th with **₹9097revenue**
- **Lowest sales day**: June 29th with only **₹899** revenue
- Clear **weekday vs weekend** pattern observed in sales performance

Top 3 Customers by Spending



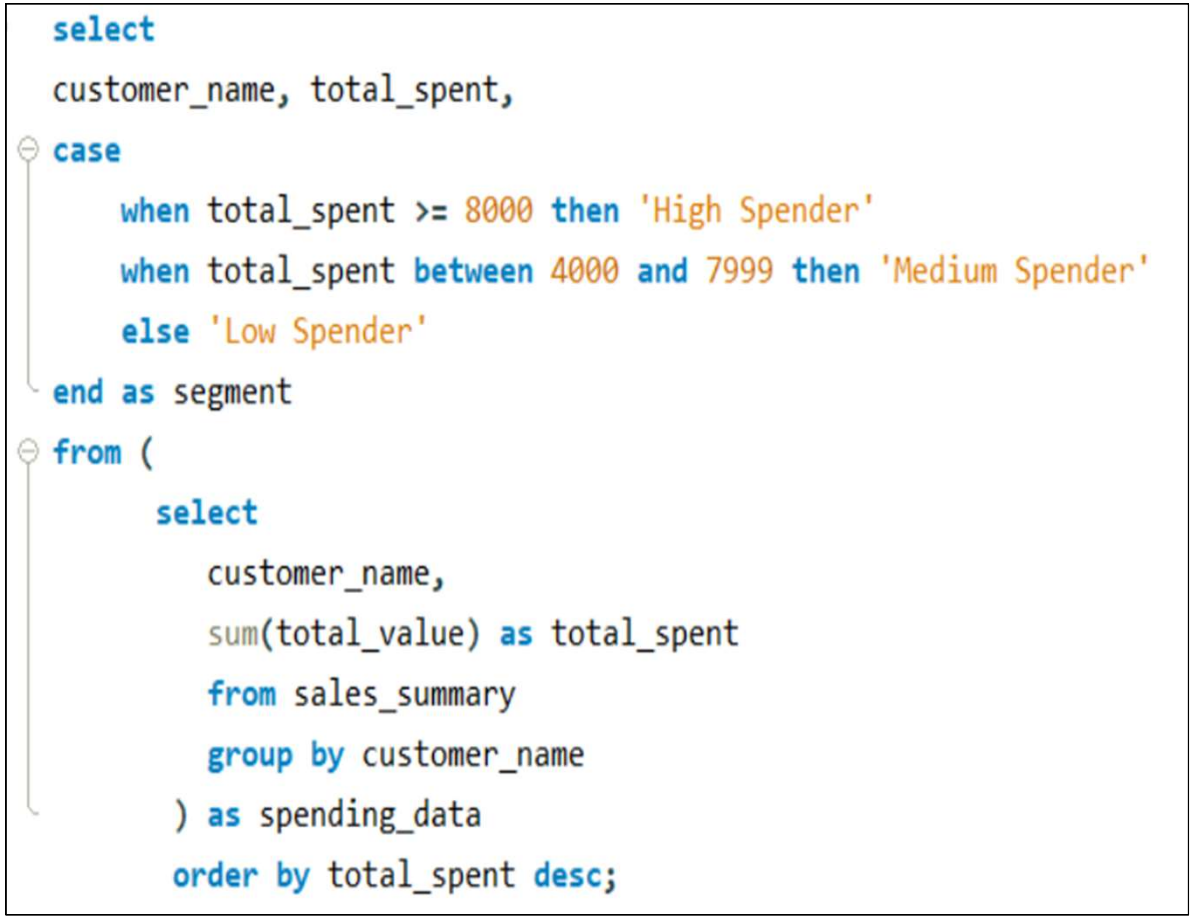
```
select
   c.name as name,
   round(sum(oi.quantity * p.price),2) as total_spent
from customers c
join orders o on c.customer_id = o.customer_id
join order_items oi on o.order_id = oi.order_id
join products p on oi.product_id = p.product_id
where month(o.order_date) = 6
group by c.name
order by total_spent desc
limit 3;
```



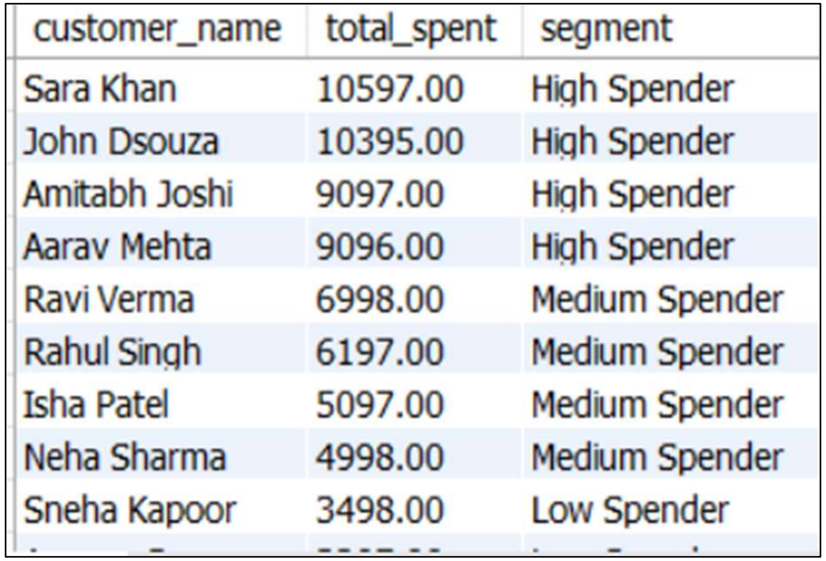
name	total_spent
Sara Khan	10597.00
John Dsouza	10395.00
Amitabh Joshi	9097.00

- **Top spenders** drive majority of revenue.
- Focus on **loyalty rewards and retention**.

Customer Segmentation By Spending



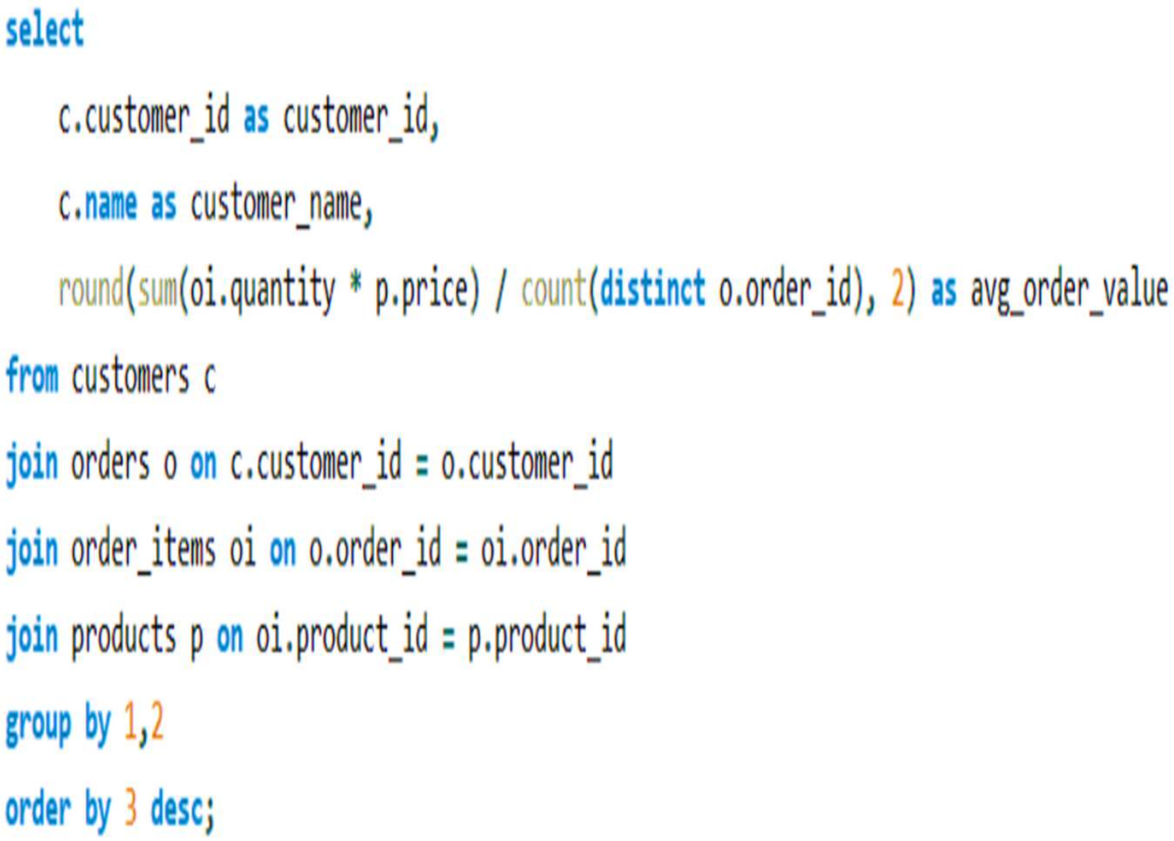
```
select
customer_name, total_spent,
case
    when total_spent >= 8000 then 'High Spender'
    when total_spent between 4000 and 7999 then 'Medium Spender'
    else 'Low Spender'
end as segment
from (
       select
          customer_name,
          sum(total_value) as total_spent
          from sales_summary
          group by customer_name
       ) as spending_data
     order by total_spent desc;
```



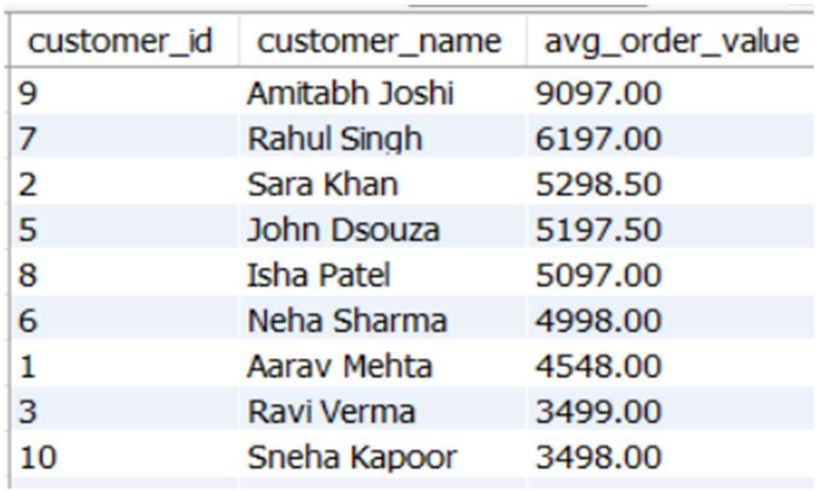
customer_name	total_spent	segment
Sara Khan	10597.00	High Spender
John Dsouza	10395.00	High Spender
Amitabh Joshi	9097.00	High Spender
Aarav Mehta	9096.00	High Spender
Ravi Verma	6998.00	Medium Spender
Rahul Singh	6197.00	Medium Spender
Isha Patel	5097.00	Medium Spender
Neha Sharma	4998.00	Medium Spender
Sneha Kapoor	3498.00	Low Spender

- **High** Spenders – **56.5%** Revenue
- **Medium** Spenders – **33.5%** Revenue
- **Low** Spenders – **10%** revenue

Average Order Value per Customer



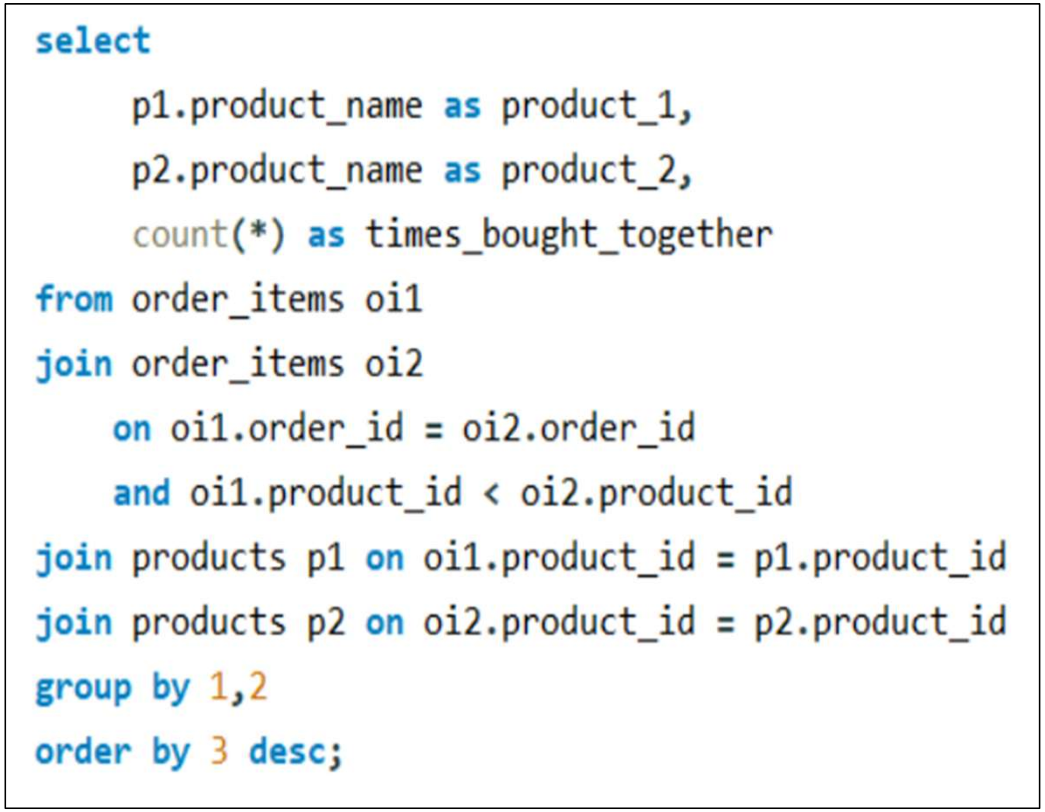
```
select
    c.customer_id as customer_id,
    c.name as customer_name,
    round(sum(oi.quantity * p.price) / count(distinct o.order_id), 2) as avg_order_value
from customers c
join orders o on c.customer_id = o.customer_id
join order_items oi on o.order_id = oi.order_id
join products p on oi.product_id = p.product_id
group by 1,2
order by 3 desc;
```



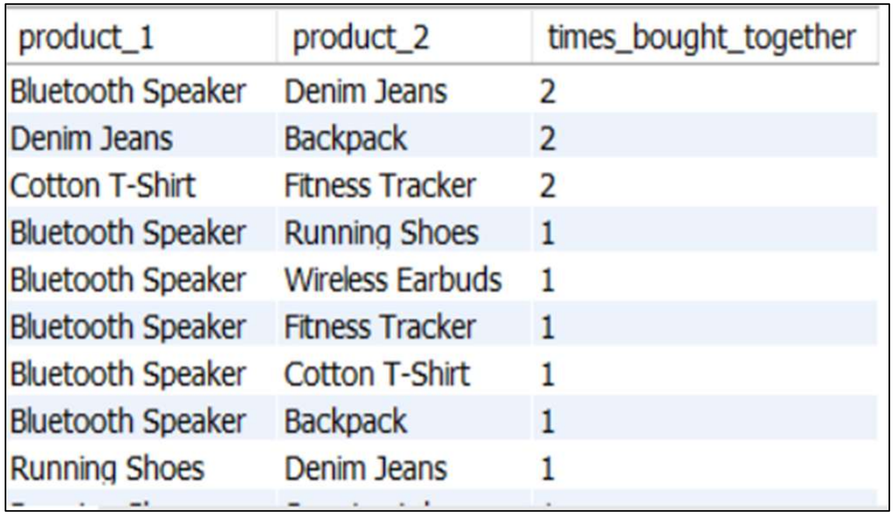
customer_id	customer_name	avg_order_value
9	Amitabh Joshi	9097.00
7	Rahul Singh	6197.00
2	Sara Khan	5298.50
5	John Dsouza	5197.50
8	Isha Patel	5097.00
6	Neha Sharma	4998.00
1	Aarav Mehta	4548.00
3	Ravi Verma	3499.00
10	Sneha Kapoor	3498.00

- **Highest AOV** – ₹9097
- **Lowest AOV** – ₹1648.5
- **Overall Average AOV** - ₹4888

Product Pair Analysis



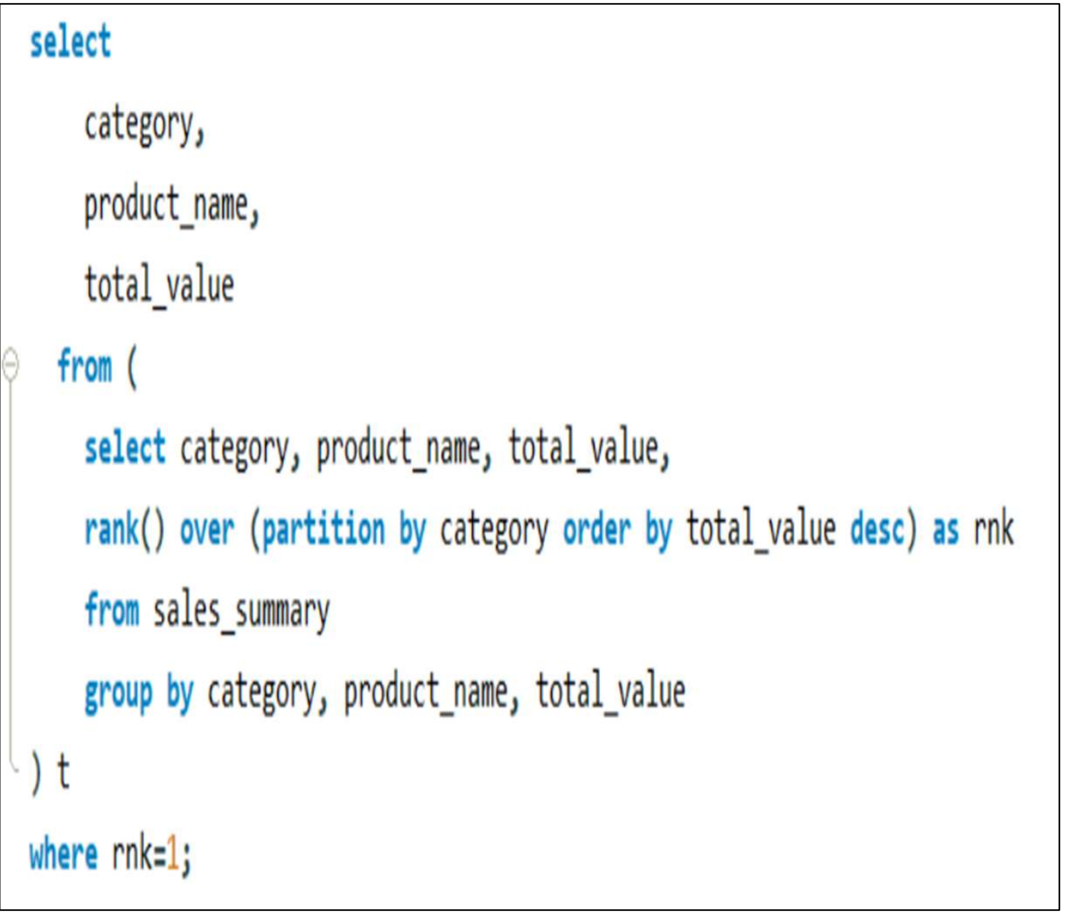
```
select
     p1.product_name as product_1,
     p2.product_name as product_2,
     count(*) as times_bought_together
from order_items oi1
join order_items oi2
    on oi1.order_id = oi2.order_id
    and oi1.product_id < oi2.product_id
join products p1 on oi1.product_id = p1.product_id
join products p2 on oi2.product_id = p2.product_id
group by 1,2
order by 3 desc;
```



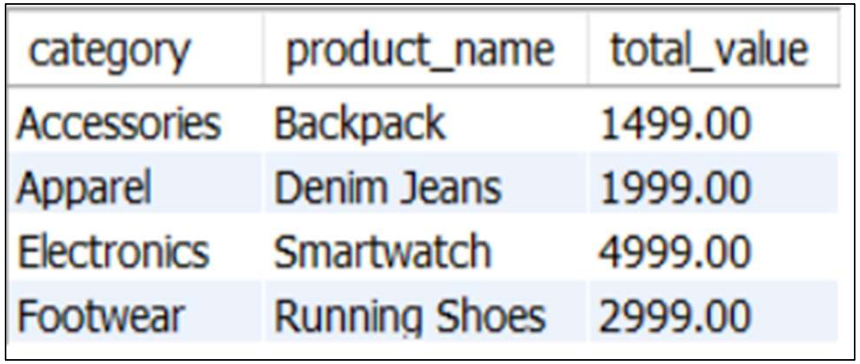
product_1	product_2	times_bought_together
Bluetooth Speaker	Denim Jeans	2
Denim Jeans	Backpack	2
Cotton T-Shirt	Fitness Tracker	2
Bluetooth Speaker	Running Shoes	1
Bluetooth Speaker	Wireless Earbuds	1
Bluetooth Speaker	Fitness Tracker	1
Bluetooth Speaker	Cotton T-Shirt	1
Bluetooth Speaker	Backpack	1
Running Shoes	Denim Jeans	1

- **5** frequent product pairs identified.
- Supports **cross-sell & bundling** strategies

Top Products per Category



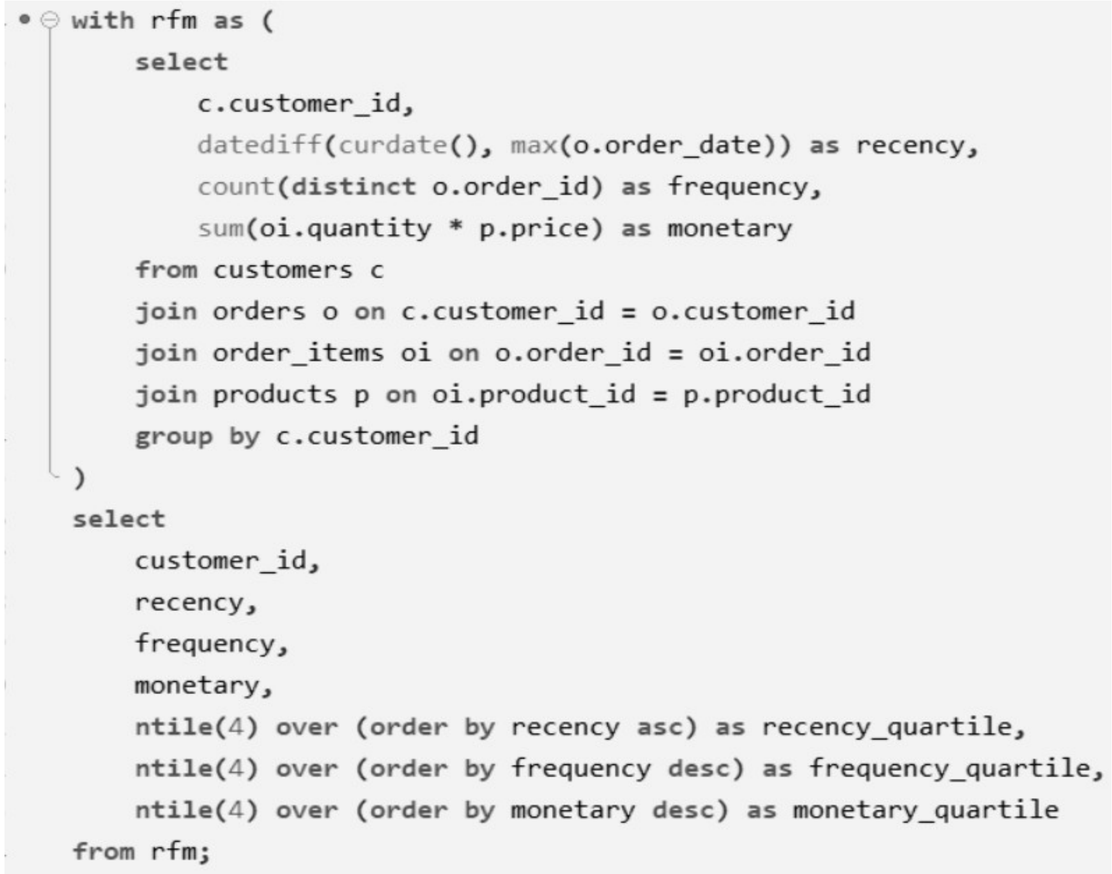
```
select
    category,
    product_name,
    total_value
  from (
    select category, product_name, total_value,
    rank() over (partition by category order by total_value desc) as rnk
    from sales_summary
    group by category, product_name, total_value
) t
where rnk=1;
```



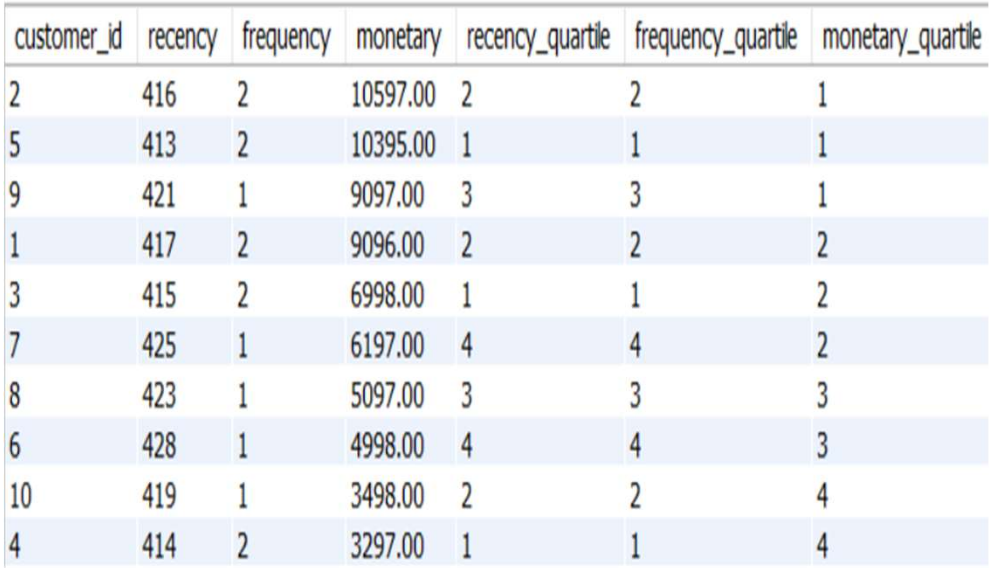
category	product_name	total_value
Accessories	Backpack	1499.00
Apparel	Denim Jeans	1999.00
Electronics	Smartwatch	4999.00
Footwear	Running Shoes	2999.00

- Electronics **Smartwatch dominates.**
- **Denim Jeans** – The Core **Fashion Driver**.
- **Footwear** -> Expands **Lifestyle** Campaigns

RFM Segmentation

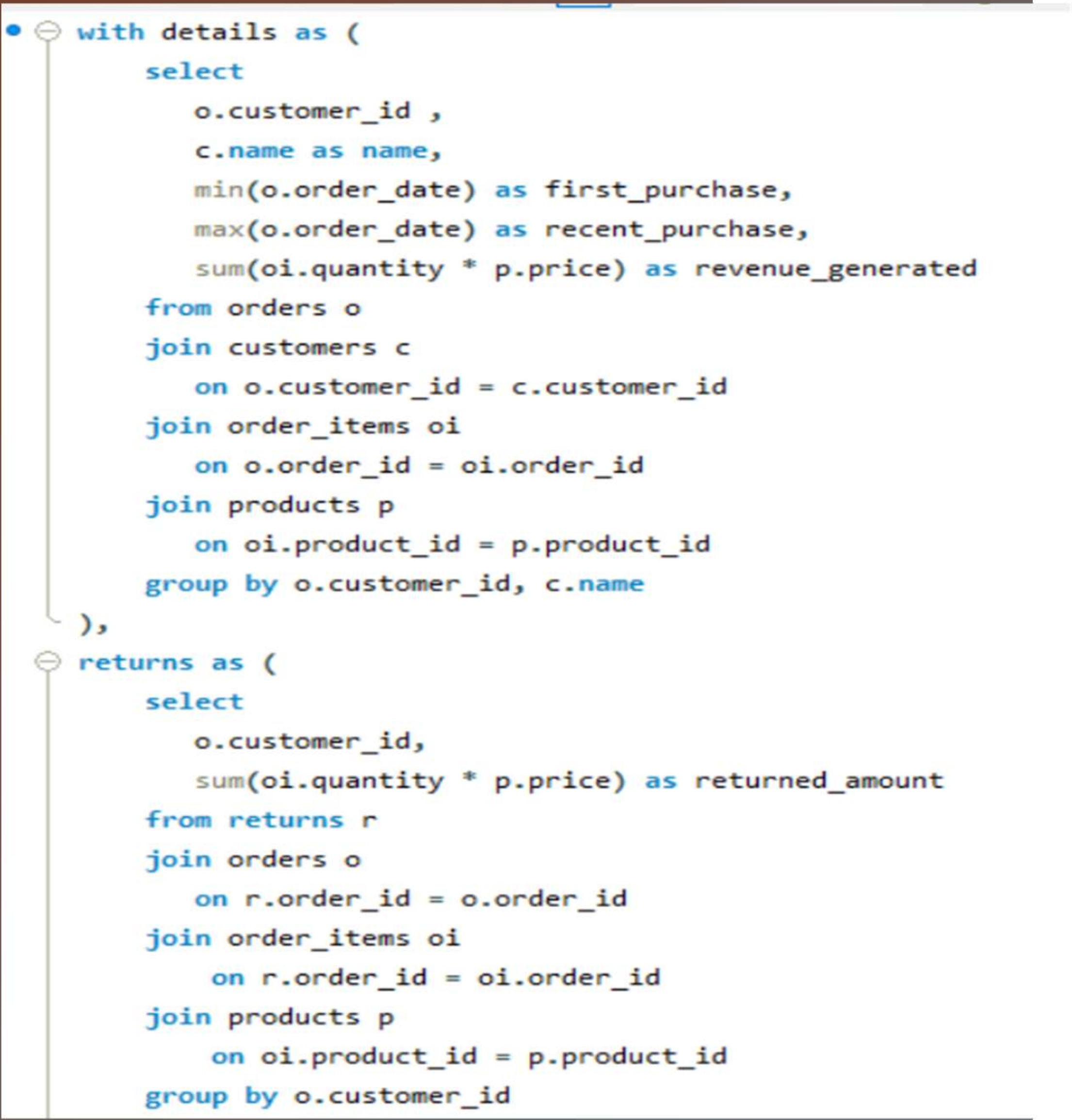


```
with rfm as (
    select
        c.customer_id,
        datediff(curdate(), max(o.order_date)) as recency,
        count(distinct o.order_id) as frequency,
        sum(oi.quantity * p.price) as monetary
    from customers c
    join orders o on c.customer_id = o.customer_id
    join order_items oi on o.order_id = oi.order_id
    join products p on oi.product_id = p.product_id
    group by c.customer_id
)
select
    customer_id,
    recency,
    frequency,
    monetary,
    ntile(4) over (order by recency asc) as recency_quartile,
    ntile(4) over (order by frequency desc) as frequency_quartile,
    ntile(4) over (order by monetary desc) as monetary_quartile
from rfm;
```



customer_id	recency	frequency	monetary	recency_quartile	frequency_quartile	monetary_quartile
2	416	2	10597.00	2	2	1
5	413	2	10395.00	1	1	1
9	421	1	9097.00	3	3	1
1	417	2	9096.00	2	2	2
3	415	2	6998.00	1	1	2
7	425	1	6197.00	4	4	2
8	423	1	5097.00	3	3	3
6	428	1	4998.00	4	4	3
10	419	1	3498.00	2	2	4
4	414	2	3297.00	1	1	4

- **1 High value** Customer 413.
- **3 Mid-Tier** (recent buyers, spending well but not consistent).
- **6 Low-Value** (low frequency and spend).

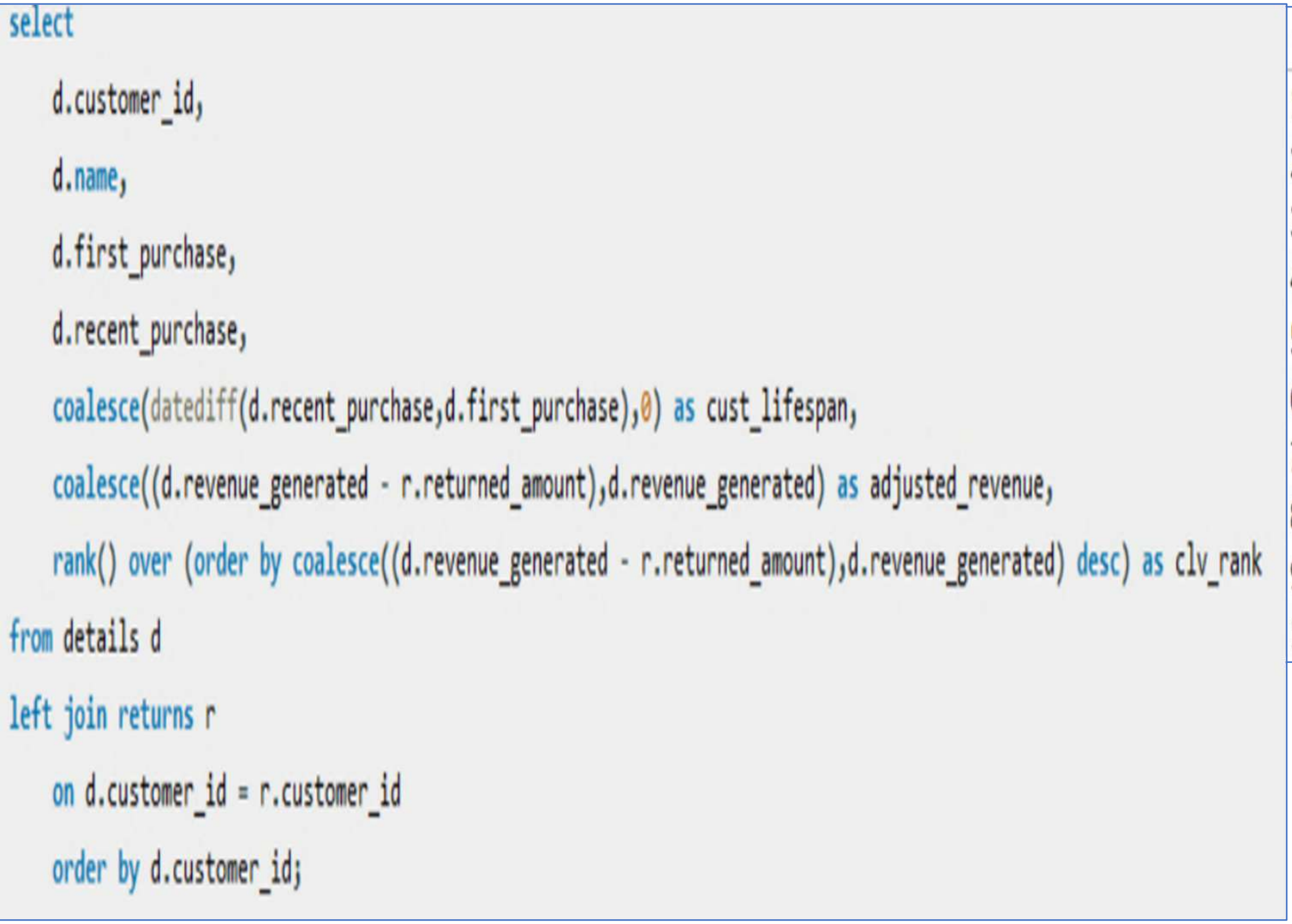


```
with details as (
    select
       o.customer_id ,
       c.name as name,
       min(o.order_date) as first_purchase,
       max(o.order_date) as recent_purchase,
       sum(oi.quantity * p.price) as revenue_generated
    from orders o
    join customers c
       on o.customer_id = c.customer_id
    join order_items oi
       on o.order_id = oi.order_id
    join products p
       on oi.product_id = p.product_id
    group by o.customer_id, c.name
),
returns as (
    select
       o.customer_id,
       sum(oi.quantity * p.price) as returned_amount
    from returns r
    join orders o
       on r.order_id = o.order_id
    join order_items oi
        on r.order_id = oi.order_id
    join products p
        on oi.product_id = p.product_id
    group by o.customer_id
```

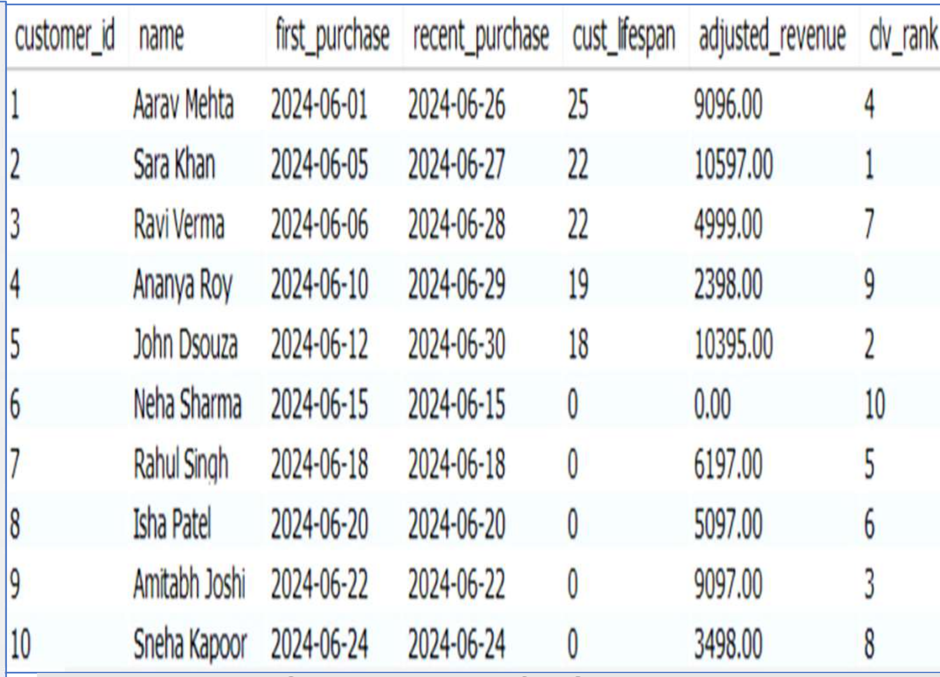
CLV Data Preparation : Step 1

- Aggregates each customer's **total revenue**.
- Captures **purchase history** from their orders.
- Prepares base data for **CLV calculation**.

Select + Output : Step 2



```
select
    d.customer_id,
    d.name,
    d.first_purchase,
    d.recent_purchase,
    coalesce(datediff(d.recent_purchase,d.first_purchase),0) as cust_lifespan,
    coalesce((d.revenue_generated - r.returned_amount),d.revenue_generated) as adjusted_revenue,
    rank() over (order by coalesce((d.revenue_generated - r.returned_amount),d.revenue_generated) desc) as clv_rank
from details d
left join returns r
    on d.customer_id = r.customer_id
    order by d.customer_id;
```



customer_id	name	first_purchase	recent_purchase	cust_lifespan	adjusted_revenue	clv_rank
1	Aarav Mehta	2024-06-01	2024-06-26	25	9096.00	4
2	Sara Khan	2024-06-05	2024-06-27	22	10597.00	1
3	Ravi Verma	2024-06-06	2024-06-28	22	4999.00	7
4	Ananya Roy	2024-06-10	2024-06-29	19	2398.00	9
5	John Dsouza	2024-06-12	2024-06-30	18	10395.00	2
6	Neha Sharma	2024-06-15	2024-06-15	0	0.00	10
7	Rahul Singh	2024-06-18	2024-06-18	0	6197.00	5
8	Isha Patel	2024-06-20	2024-06-20	0	5097.00	6
9	Amitabh Joshi	2024-06-22	2024-06-22	0	9097.00	3
10	Sneha Kapoor	2024-06-24	2024-06-24	0	3498.00	8

- Average Customer Life Span : 18-25 days.
- Top 3 Customers contribute ~50% of total CLV.
- Targets loyal high-value buyers.

BUSINESS INSIGHTS – JUNE 2024

- ❑ **₹60K+ net revenue captured** across June transactions.
- ❑ **Top 3 products & categories** drive **~50% of sales volume**.
- ❑ **30% revenue** contributed by just **2 high-value customers**.
- ❑ **11% revenue loss** identified due to product returns.
- ❑ **Customer segmentation (RFM)** built for **targeted marketing & retention**.
- ❑ **Actionable KPIs (CLV, AOV, Return Rate)** directly mapped to business growth.
- ❑ Strong **loyalty potential** : **50% Repeat buyer rate**.
- ❑ **Top 2 Products** : Running Shoes and Denim Jeans contribute **45% to Total Sales**.