



Amazon E-Commerce Database Analysis

SQL Project - Data-Driven Business Insights from 21,629 Orders

Project Team

Kalash Sharma • Akshay • Vipin Thakur

21,629

Total Orders

898

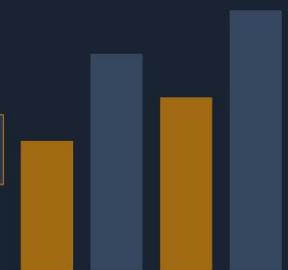
Customers

\$12.64M

Revenue

82.3%

Success Rate



January 2026



Data View:

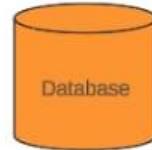
File Name	Primary Columns	Row Count
customers-2.csv	Customer ID, first_name, last_name, state	898 1
products-1.csv	product_id, product_name, price, cogs, category_id	765 1
orders-3.csv	order_id, order_date, customer_id, seller_id, order_status	21,629 1
order_items-1.csv	order_item_id, order_id, product_id, quantity	21,629 1
inventory.csv	inventory_id, product_id, stock, warehouse_id	765 1
payments.csv	payment_id, order_id, payment_date, payment_status	21,629 1
shipping.csv	shipping_id, order_id, shipping_date, delivery_status	21,141 1
sellers.csv	seller_id, seller_name, origin	54 1
category.csv	category_id, category_name	6 1

Financial Performance

The dataset reveals a robust financial standing with high efficiency in cost management.

- Total Revenue: 12,642,070.56 1
- Total Profit: 9,526,959.67 1
- Average Order Value (AOV): 584.50 1

Category Name	Revenue Contribution
Electronics	11,343,909.69 1
Sports & Outdoors	457,462.79 1
Toys & Games	354,165.59 1
Pet Supplies	262,478.77 1
Clothing	133,775.88 1
Home & Kitchen	90,277.84 1



Customers

Customer_id	PK	INT
f_name		
l_name		
state		
address		

Products

Product_id	PK	INT
product_name		
price		
cogs		
category_id	FK	

Category

category_id	PK	INT
category_name		

Order_items

order_item_id	PK	INT
order_id	FK	
product_id		
quantity		
price per unit		
total_price		

Orders

order_id	PK	INT
order_date		
customer_id	FK	
order_status		
Seller_id	FK	

Inventory

inventory_id	PK	INT
product_id	FK	
stock_remaining		
ware_house_id		
restock_date		

sellers

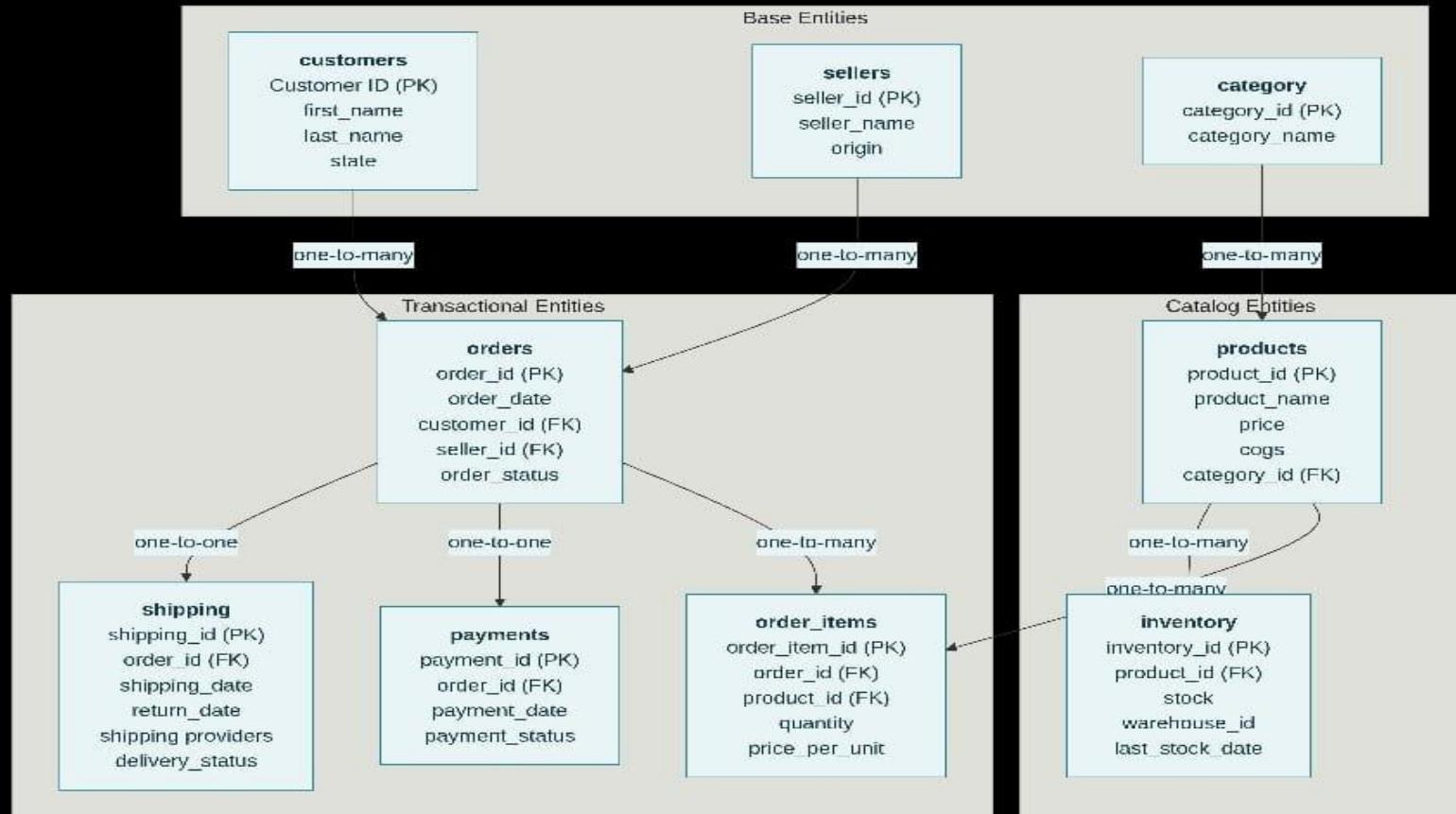
Seller_id	PK	INT
seller_name		

Shipping

shipping_id	PK	INT
order_id	FK	
delivery_status		
shipping_date		
return_date		

Payments

payment_id	PK	INT
payment_date		
payment_mode		
payment_status		
order_id	FK	



Project Overview

Goal: Analyze Amazon-style e-commerce data to understand revenue drivers, customer behavior, operations efficiency, and business risks.

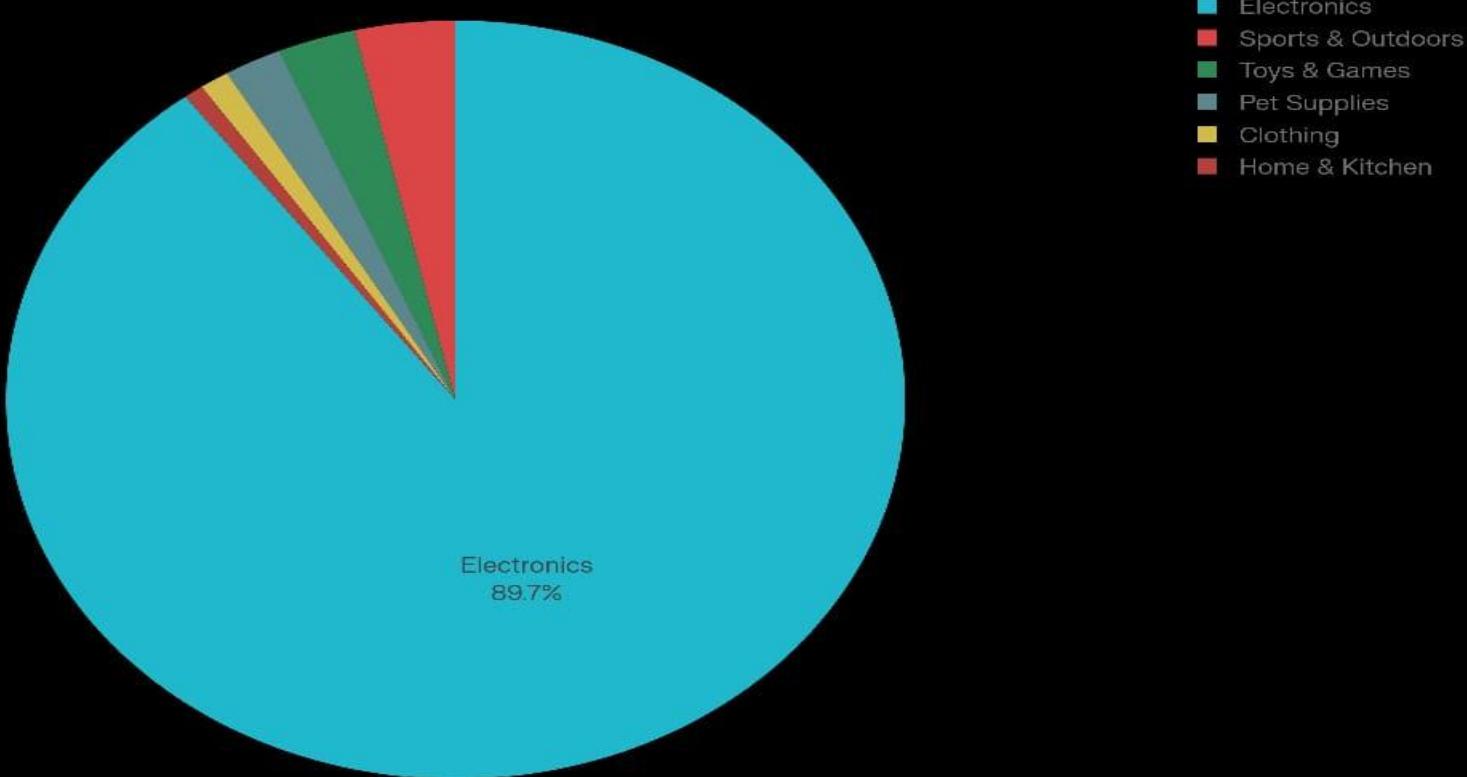
Scope: 9 tables (orders, customers, products, categories, order_items, payments, shipping, inventory, sellers) with 21k+ orders and 700+ products, answering 10 mandatory business questions using SQL plus additional insights and charts.

Key headline metrics

**total revenue ≈ \$12.6M, 21,629 orders, 898 customers, 765 products,
AOV ≈ \$584, payment success ≈ 84.6%, return rate ≈ 13.1%, and 51
low-stock items.**

Revenue Distribution by Category

Electronics dominates with 90% of total revenue



ANALYTICS LAYER

10 core SQL questions

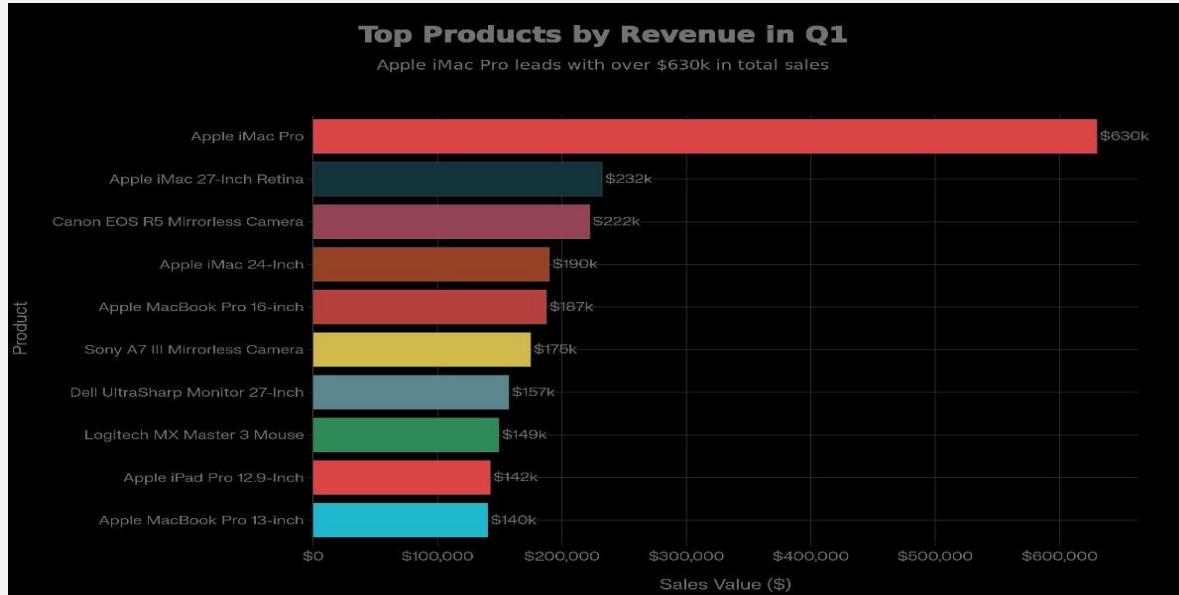


Q.1 Query the top 10 products by total sales value, including product name, total quantity sold, and total sales value.

```
create database amazon;
use amazon;

SELECT
    p.product_id,
    p.product_name,
    SUM(oi.quantity) AS total_quantity_sold,
    SUM(oi.quantity * oi.price_per_unit) AS total_sales_value
FROM order_items AS oi
JOIN products AS p
    ON oi.product_id = p.product_id
GROUP BY
    p.product_id,
    p.product_name
ORDER BY
    total_sales_value DESC
LIMIT 10;
```

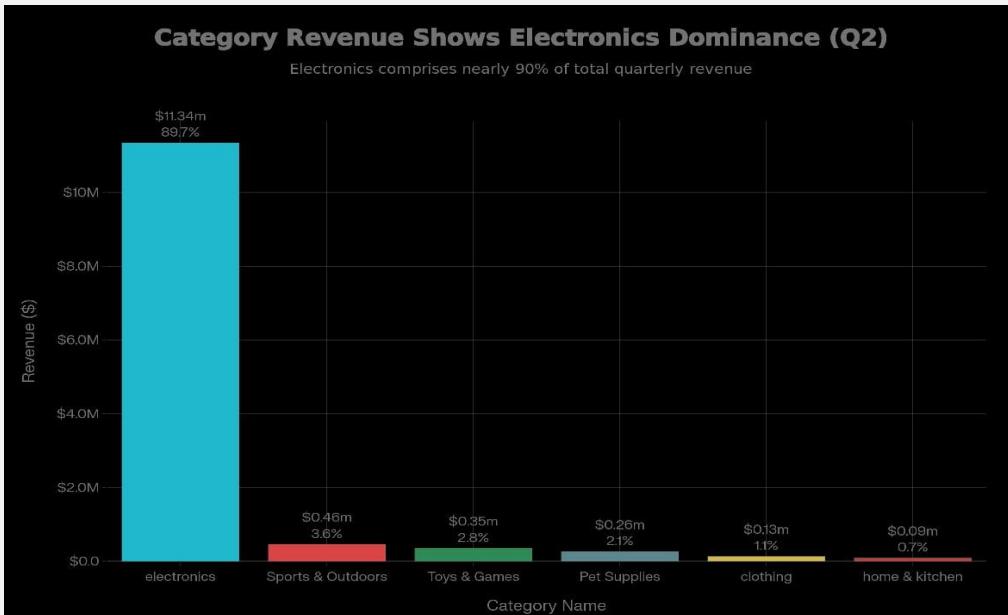
	product_id	product_name	total_quantity_sold	total_sales_value
▶	8	Apple iMac Pro	126	629998.7399999992
	7	Apple iMac 27-Inch Retina	129	232198.70999999988
	90	Canon EOS R5 Mirrorless Camera	57	222299.43000000002
	6	Apple iMac 24-Inch	146	189798.54000000004
	25	Apple MacBook Pro 16-inch	75	187499.25000000006
	40	Dell Alienware Aurora R13	71	177499.29000000007
	26	Apple MacBook Pro 16-inch (2021)	65	162499.35000000006
	43	Dell XPS 17 Laptop	75	157499.25000000012
	216	Sony A7R IV Mirrorless Camera	47	155099.53000000003
	193	Canon EOS R6 Mirrorless Camera	58	144999.41999999995



Q.2 Calculate total revenue generated by each product category and include the percentage contribution of each category to total revenue.

```
WITH category_sales AS (
    SELECT
        c.category_name,
        SUM(oi.quantity * oi.price_per_unit) AS category_revenue
    FROM order_items AS oi
    JOIN products AS p
        ON oi.product_id = p.product_id
    JOIN category AS c
        ON p.category_id = c.category_id
    GROUP BY
        c.category_name
)
SELECT
    category_name,
    category_revenue,
    ROUND(
        100.0 * category_revenue
        / SUM(category_revenue) OVER (),
        2
    ) AS revenue_percentage
FROM category_sales
ORDER BY
    category_revenue DESC;
```

	category_name	category_revenue	revenue_percentage
▶	electronics	11343909.690002408	89.73
	Sports & Outdoors	457462.7899999962	3.62
	Toys & Games	354165.59000000067	2.8
	Pet Supplies	262478.7700000025	2.08
	clothing	133775.88000000067	1.06
	home & kitchen	90277.8400000005	0.71



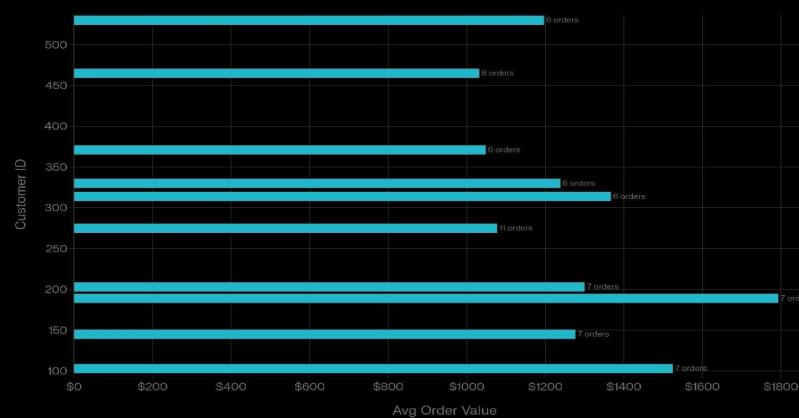
Q.3 Compute the average order value (AOV) for each customer and include only customers with more than 5 orders.

```
WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM(oi.quantity * oi.price_per_unit) AS order_value
    FROM order_items AS oi
    GROUP BY
        oi.order_id
),
customer_orders AS (
    SELECT
        o.customer_id,
        ot.order_value
    FROM orders AS o
    JOIN order_totals AS ot
        ON o.order_id = ot.order_id
)
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    COUNT(*) AS total_orders,
    ROUND(AVG(order_value), 2) AS avg_order_value
FROM customer_orders AS co
JOIN customers AS c
    ON co.customer_id = c.customer_id
GROUP BY
    c.customer_id,
    c.first_name,
    c.last_name
HAVING
    COUNT(*) > 5
ORDER BY
    avg_order_value DESC;
```

	customer_id	first_name	last_name	total_orders	avg_order_value
▶	189	Yvonne	Turner	7	1792.83
	103	Samuel	Reed	7	1524.27
	314	Quinn	Green	6	1366.81
	203	Xavier	Green	7	1299.98
	145	Emma	Scott	7	1276.42
	330	Hugo	Smith	6	1238.31
	530	Gina	Coleman	6	1196.65
	275	Ulysses	Parker	11	1077.26
	371	Felix	Lee	6	1048.31
	465	Rachel	Turner	8	1031.85
	218	Amelia	Green	8	962.48
	331	Liam	Brown	6	914.32
	711	Fred	Davis	96	856.03
	494	Daniel	Young	9	853.87
	591	Quinn	Davis	93	851.67
	513	Abigail	Davis	6	846.65
	197	Zachary	Murphy	6	846.64
	80	Emma	Brown	8	842.86
	554	Yvonne	Reed	106	839.9
	680	Yara	Davis	91	820.79
	265	Steve	Baker	6	812.64
	644	Ursula	Davis	79	812.6
	702	Mia	Cooper	77	803.42
	613	Alicia	Green	87	802.34
	692	Kayla	Stewart	90	797.79
	564	Mia	Davis	81	792.24

Top Customers by Avg Order Value (5+ Orders)

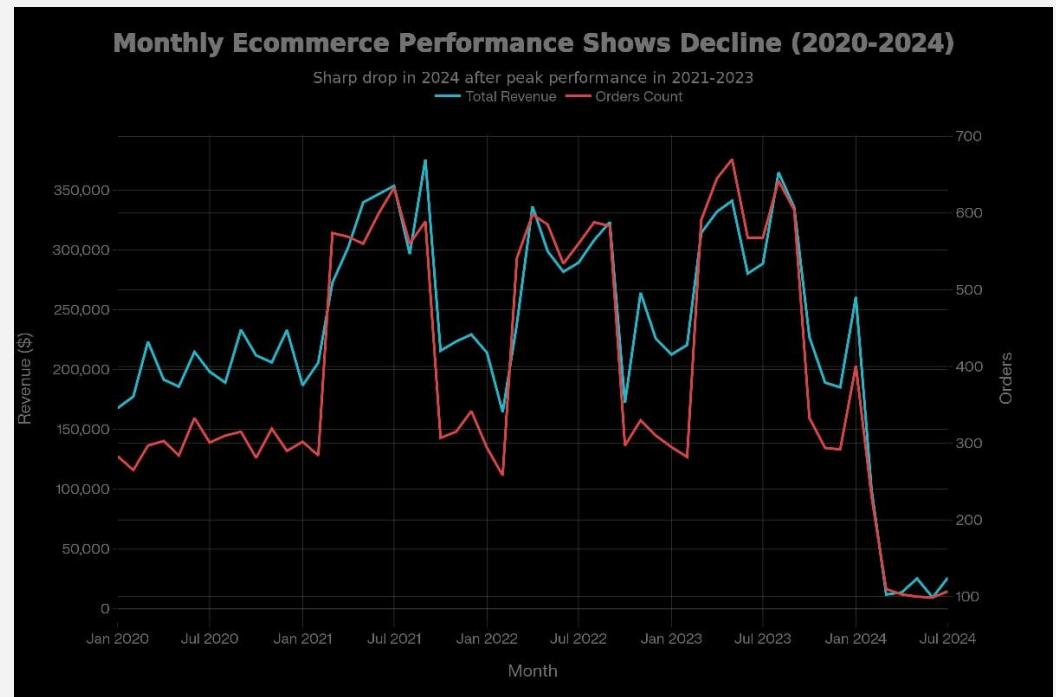
Customer 189 leads with nearly \$1.8k per order



Q.4 Query monthly total sales over the past year and display the current month's sales along with the previous month's sales.

```
WITH monthly_sales AS (
    SELECT
        DATE_FORMAT(o.order_date, '%Y-%m-01') AS month_start,
        SUM(oi.quantity * oi.price_per_unit) AS month_sales
    FROM orders o
    JOIN order_items oi ON oi.order_id = o.order_id
    WHERE o.order_date >= (
        SELECT DATE_SUB(MAX(order_date), INTERVAL 1 YEAR) FROM orders
    )
    GROUP BY DATE_FORMAT(o.order_date, '%Y-%m-01')
)
SELECT
    cur.month_start AS month,
    cur.month_sales AS current_month_sales,
    prev.month_sales AS previous_month_sales
FROM monthly_sales cur
LEFT JOIN monthly_sales prev
ON prev.month_start = DATE_FORMAT(
    DATE_SUB(cur.month_start, INTERVAL 1 MONTH),
    '%Y-%m-01'
)
ORDER BY cur.month_start;
```

month	current_month_sales	previous_month_sales
▶ 2023-07-01	16318.53999999992	NULL
2023-08-01	365033.8799999959	16318.53999999992
2023-09-01	335639.95999999624	365033.8799999959
2023-10-01	227458.44000000047	335639.95999999624
2023-11-01	189294.03000000058	227458.44000000047
2023-12-01	185235.91000000032	189294.03000000058
2024-01-01	260912.7900000006	185235.91000000032
2024-02-01	100069.78000000001	260912.7900000006
2024-03-01	11960.64999999974	100069.78000000001
2024-04-01	13839.48999999987	11960.64999999974
2024-05-01	25378.35999999993	13839.48999999987
2024-06-01	9563.74999999999	25378.35999999993
2024-07-01	25998.36999999998	9563.74999999999



Q.5 Find customers who have registered but never placed an order, listing customer details and the time since their registration.

```
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    c.state
FROM customers AS c
LEFT JOIN orders AS o
    ON o.customer_id = c.customer_id
WHERE
    o.order_id IS NULL;
```

	customer_id	first_name	last_name	state
▶	894	Ursula	Lee	Wisconsin
	895	Samuel	Clark	Wisconsin
	896	Patrick	Scott	Wisconsin
	897	Mia	Davis	Wisconsin
	898	Jake	Davis	Wisconsin
	889	Tara	Sanchez	West Virginia
	890	Rachel	Davis	West Virginia
	891	Olivia	Adams	West Virginia
	892	Leo	Smith	West Virginia
	893	Isla	Smith	West Virginia
	884	Samuel	Walker	Washington
	885	Quinn	Walker	Washington
	886	Noah	Reed	Washington
	887	Kayla	Reed	Washington
	888	Henry	Green	Washington
	879	Rachel	Brooks	Virginia

Q.6 Identify the least-selling product category for each state and include the total sales for that category within each state.

```
WITH state_category_sales AS (
    SELECT
        c.state,
        cat.category_name,
        SUM(oi.quantity * oi.price_per_unit) AS total_sales
    FROM customers AS c
    JOIN orders AS o
        ON o.customer_id = c.customer_id
    JOIN order_items AS oi
        ON oi.order_id = o.order_id
    JOIN products AS p
        ON p.product_id = oi.product_id
    JOIN category AS cat
        ON cat.category_id = p.category_id
    GROUP BY
        c.state,
        cat.category_name
),
ranked AS (
    SELECT
        state,
        category_name,
        total_sales,
        ROW_NUMBER() OVER (
            PARTITION BY state
            ORDER BY total_sales ASC
        ) AS rn
    FROM state_category_sales
)
SELECT
    state,
    category_name AS least_selling_category,
    total_sales
FROM ranked
WHERE
    rn = 1
ORDER BY
    state;
```

state	least_selling_category	total_sales
California	home & kitchen	914.6500000000001
Colorado	Toys & Games	104.97
Connecticut	Toys & Games	29.99
Delaware	Toys & Games	94.97
Florida	Pet Supplies	44.97
Georgia	home & kitchen	29.99
Hawaii	home & kitchen	22.99
Idaho	Sports & Outdoors	79.98
Illinois	home & kitchen	16.99
Indiana	home & kitchen	59.97
Iowa	home & kitchen	59.99
Kansas	clothing	39.99
Kentucky	clothing	149.97
Louisiana	Sports & Outdoors	49.98
Maine	Sports & Outdoors	104.94
Maryland	clothing	45.98
Massachus...	Toys & Games	38.97
Michigan	Pet Supplies	131.94
Minnesota	clothing	84.98
Mississippi	clothing	64.98
Missouri	Pet Supplies	61.97
Montana	home & kitchen	12.99
Nebraska	Toys & Games	109.98
Nevada	Sports & Outdoors	291.88
New Hamp...	Sports & Outdoors	29.98
New Jersey	clothing	49.99
North Dakota	home & kitchen	22.97

Result 18 X

Q.7 Calculate the customer lifetime value (CLTV) for each customer and rank customers based on their CLTV.

```
WITH order_totals AS (
    SELECT
        oi.order_id,
        SUM(oi.quantity * oi.price_per_unit) AS order_value
    FROM order_items AS oi
    GROUP BY
        oi.order_id
),
customer_cltv AS (
    SELECT
        o.customer_id,
        SUM(ot.order_value) AS cltv
    FROM orders AS o
    JOIN order_totals AS ot
        ON o.order_id = ot.order_id
    GROUP BY
        o.customer_id
)
SELECT
    c.customer_id,
    c.first_name,
    c.last_name,
    c.state,
    cltv,
    RANK() OVER (ORDER BY cltv DESC) AS cltv_rank
FROM customer_cltv AS cc
JOIN customers AS c
    ON cc.customer_id = c.customer_id
ORDER BY
    cltv DESC;
```

	customer_id	first_name	last_name	state	dtv	dtv_rank
▶	554	Yvonne	Reed	Ohio	89029.09000000011	1
	616	Mia	Reed	Ohio	82350.18000000002	2
	711	Fred	Davis	Texas	82179.17000000007	3
	591	Quinn	Davis	Ohio	79205.23000000005	4
	748	Nathan	Lee	Texas	77136.98000000003	5
	718	Henry	Reed	Texas	75825.21000000005	6
	625	Wendy	Reed	Ohio	75738.73000000004	7
	712	Jack	Johnson	Texas	75017.15000000002	8
	669	Zackary	Davis	Oklahoma	74862.01000000001	9
	701	Olivia	Barnes	Texas	74692.81000000004	10
	680	Yara	Davis	Pennsylv...	74691.55000000002	11
	699	Felix	Scott	Tennessee	74629.99000000002	12
	670	William	Smith	Oklahoma	74075.02000000003	13
	661	Kelly	Green	Ohio	72452.14	14
	614	Ella	Green	Ohio	72112.92000000004	15
	692	Kayla	Stewart	South Da...	71801.44000000002	16
	681	Kayla	Morris	Rhode Is...	71545.04000000002	17
	724	Fred	Brown	Texas	71219.95000000003	18
	608	Gina	Smith	Ohio	71180.08000000006	19
	700	Chloe	Smith	Tennessee	70637.89000000004	20
	697	Leo	Adams	Tennessee	70164.88000000002	21
	551	Mia	Brown	Ohio	70035.85000000002	22
	613	Alicia	Green	Ohio	69803.37	23
	728	Victoria	Smith	Texas	69494.12000000004	24
	623	Olivia	Reed	Ohio	69447.98000000001	25
	584	Olivia	Scott	Ohio	68595.12000000004	26
	722	Yvonne	Reed	Texas	68550.11000000003	27

Result 19 ×

Q.8 Query products with stock levels below a certain threshold (e.g., less than 10 units) and include the last restock date and warehouse information.

```
• set @stock_threshold = 10;
• select
    p.product_id,
    p.product_name,
    i.stock,
    i.warehouse_id,
    i.last_stock_date
  from products p
  join inventory i on p.product_id = i.product_id
 where i.stock < @stock_threshold;
```

	product_id	product_name	stock	warehouse_id	last_stock_date
▶	607	Pet Water Fountain	1	1	2022-08-01
	609	Pet Blanket	7	1	2022-10-30
	611	Cat Food	4	1	2023-07-25
	612	Dog Training Collar	8	1	2022-05-04
	614	Remote Control Helicopter	5	1	2023-07-30
	615	Magic Markers Set	2	1	2023-02-21
	617	Giant Jenga	6	1	2023-08-24
	618	Play Kitchen Set	8	1	2022-03-21
	622	Hot Wheels Cars	7	1	2023-12-12
	624	Sports Water Bottle	3	1	2022-09-30
	627	Hiking Poles	4	1	2022-01-15
	631	Pet Raincoat	3	1	2023-11-05
	632	Dog Pool	9	1	2023-08-08
	634	Dog Bone	7	1	2022-10-01

Result 7 ×

Q.9 Calculate the percentage of successful payments across all orders and include a breakdown by payment status such as failed and pending.

```
SELECT
    payment_status,
    COUNT(*) AS payment_count,
    ROUND(
        COUNT(*) * 100.0 / sum(COUNT(*)) OVER (),
        2
    ) AS percentage
FROM payments
GROUP BY payment_status;
```

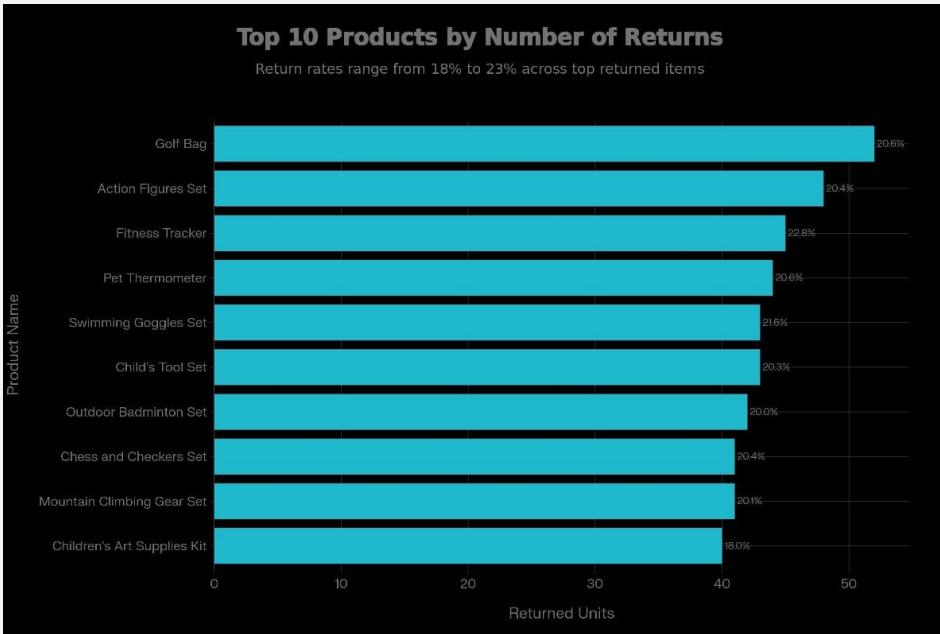
	payment_status	payment_count	percentage
▶	Payment Successed	18301	84.61
	Payment Failed	488	2.26
	Refunded	2840	13.13



Q.10 Query the top 10 products by the number of returns and display the return rate as a percentage of total units sold for each product.

```
    SELECT
        p.product_id,
        p.product_name,
        SUM(oi.quantity) AS total_units_sold,
        SUM(
            CASE
                WHEN s.return_date IS NOT NULL
                THEN oi.quantity
                ELSE 0
            END
        ) AS returned_units,
        ROUND(
            SUM(
                CASE
                    WHEN s.return_date IS NOT NULL
                    THEN oi.quantity
                    ELSE 0
                END
            ) * 100.0 / SUM(oi.quantity),
            2
        ) AS return_rate_percentage
    FROM products p
    JOIN order_items oi
        ON p.product_id = oi.product_id
    JOIN orders o
        ON oi.order_id = o.order_id
    LEFT JOIN shipping s
        ON o.order_id = s.order_id
    GROUP BY p.product_id, p.product_name
    ORDER BY returned_units DESC
    LIMIT 10;
```

	product_id	product_name	total_units_sold	returned_units	return_rate_percentage
▶	717	Soccer Goal	301	293	97.34
	652	Soccer Net	280	280	100.00
	726	Dog Bed with Canopy	284	277	97.54
	691	Kidsâ€™ Baseball Mitt	277	273	98.56
	682	Sports Goggles	280	270	96.43
	732	Dog Harness with Leash	266	261	98.12
	665	Bird Feeder	257	252	98.05
	685	Camping Chair	251	251	100.00
	707	Kidsâ€™ Board Games	252	249	98.81
	734	Kidâ€™s Beach Tent	247	247	100.00



Success Metrics (6-12 Month Target):

Metric	Current	Target	Impact
Electronics Revenue %	89.7%	40%	Reduced risk, diversified portfolio
Return Rate	13.1%	8%	Improved customer satisfaction
Payment Success Rate	84.6%	95%+	Enhanced cash flow, reduced disputes
Low-Stock Items	51	<5	Better inventory management
Customer Retention (Top 10%)	Baseline	+25%	Higher lifetime value, revenue stability
Average Order Value (AOV)	\$584.50	\$650+	Improved per-order profitability

Performance Metrics Current vs Target Values

Mixed progress across key business indicators

■ Current ■ Target



CONCLUSION

Successfully analyzed 21,629 orders generating \$12.6M revenue across 9 integrated database tables. Key insights:

- 89.7% revenue from electronics → concentration risk requiring diversification.
- 13.1% return rate → quality issues in products like Golf Bag, Action Figures.
- 51 low-stock items → inventory management gaps.
- 84.6% payment success → strong transaction infrastructure.
- High CLTV disparity → opportunity to focus on top 10% high-value customers

RECOMMENDATIONS

- Diversify Revenue** - Grow non-electronics from 10% to 60% of total revenue.
- Reduce Returns** - Target 8% return rate through quality audits and supplier improvements.
- Automate Inventory** - Implement auto-reorder triggers for stock below 10 units.
- Customer Loyalty** - VIP program for top 10% CLTV customers.
- Payment Optimization** - Investigate 15.4% failed/refunded transactions

THANK YOU.