Rashedul Islam
Portfolio.

Online Retail Sales Performance Analysis

Contact:
66.rashedul.islam@gmail.com

About Project

This project focuses on analyzing a retail sales database using SQL.

- Dataset: Customers, Orders, Order_Items, Products
- Goal: Extract business insights through queries at 3 levels – Basic, Intermediate & Advanced
- Key Insights: Total revenue, top-selling products & categories, high-value customers, order trends, and customer lifetime value
- Skills Used: SQL joins, aggregations, window functions, and business-focused analysis

Key Metrices

```sql
-- How many total customers are in the database?

SELECT
    COUNT(*) AS total_customer
FROM
    customers;
```

Result Grid

	total_customer
▶	100

```sql
-- What is the total number of orders placed?

SELECT
    COUNT(*) AS total_orders
FROM
    orders;
```

Result Grid

	total_orders
▶	500

Key Metrices

```sql
-- How many products are available in the inventory?

SELECT
    COUNT(*) AS total_product
FROM
    products;
```

Result Grid

	total_product
▶	50

```sql
-- What is the total revenue from completed orders?

SELECT
    ROUND(SUM(order_items.quantity * order_items.unit_price),
            2) AS total_revenue
FROM
    order_items
        JOIN
    orders ON order_items.order_id = orders.order_id
WHERE
    orders.status = 'Completed';
```

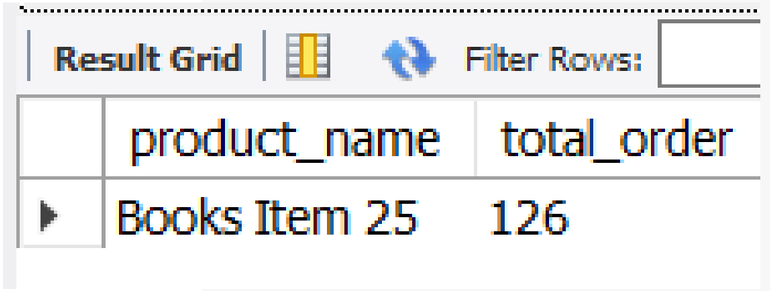
Result Grid

	total_revenue
▶	710888.11

Highest Selling Product

```sql
-- Which product has been sold the most (by quantity)?

SELECT
    products.product_name,
    SUM(order_items.quantity) AS total_order
FROM
    products
        JOIN
    order_items ON products.product_id = order_items.product_id
GROUP BY products.product_name
ORDER BY total_order DESC
LIMIT 1;
```



Top 3 Product Categories

```sql
-- List the top 3 most popular product categories (by total quantity sold).

SELECT
    products.category,
    SUM(order_items.quantity) AS total_order
FROM
    products
        JOIN
    order_items ON products.product_id = order_items.product_id
GROUP BY products.category
ORDER BY total_order DESC
LIMIT 3;
```

Result Grid | Filter Rows:

category	total_order
Books	1151
Electronics	1105
Clothing	833

Country with Highest number of Customers

```sql
-- Which country has the highest number of customers?

SELECT
    customers.country, COUNT(customers.country) AS num
FROM
    customers
GROUP BY customers.country
ORDER BY num DESC
LIMIT 1;
```

Result Grid | Filte

	country	num
▶	Denmark	2

Cancelled Order Lists

```sql
-- List all orders that were cancelled.

SELECT
    customers.name,
    customers.email,
    products.product_name,
    orders.order_date,
    orders.status
FROM
    orders
        JOIN
    customers ON orders.customer_id = customers.customer_id
        JOIN
    order_items ON orders.order_id = order_items.order_id
        JOIN
    products ON order_items.product_id = products.product_id
WHERE
    orders.status = 'Cancelled';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	email	product_name	order_date	status
Tiffany Barnes	brobinson@johnson-rogers.biz	Clothing Item 26	2024-07-31 00:00:00	Cancelled
Timothy Hancock	vjohnson@patton-jenkins.com	Books Item 25	2025-06-28 00:00:00	Cancelled
Timothy Hancock	vjohnson@patton-jenkins.com	Home & Kitchen Item 7	2025-06-28 00:00:00	Cancelled
Timothy Hancock	vjohnson@patton-jenkins.com	Home & Kitchen Item 31	2025-06-28 00:00:00	Cancelled
Timothy Hancock	vjohnson@patton-jenkins.com	Books Item 40	2025-06-28 00:00:00	Cancelled
Timothy Hancock	vjohnson@patton-jenkins.com	Books Item 40	2025-06-28 00:00:00	Cancelled
Bobby Hall	oramirez@martin-kelly.com	Books Item 47	2024-10-05 00:00:00	Cancelled
Bobby Hall	oramirez@martin-kelly.com	Books Item 28	2024-10-05 00:00:00	Cancelled
Bobby Hall	oramirez@martin-kelly.com	Electronics Item 15	2024-10-05 00:00:00	Cancelled
Bobby Hall	oramirez@martin-kelly.com	Clothing Item 22	2024-10-05 00:00:00	Cancelled
Bobby Hall	oramirez@martin-kelly.com	Clothing Item 26	2024-10-05 00:00:00	Cancelled

Result 2

Output

Total Revenue By Product Category

```sql
-- Calculate total revenue by product category.

SELECT
    products.category,
    ROUND(SUM(order_items.quantity * order_items.unit_price),
            2) AS total_revenue
FROM
    products
        JOIN
    order_items ON products.product_id = order_items.product_id
GROUP BY products.category;
```

Result Grid | Filter Rows:

category	total_revenue
Home & Kitchen	122419.61
Clothing	241844.57
Books	370027.91
Electronics	227517.54

Top 10 Customers

```sql
-- Find the top 10 customers by total revenue.

SELECT
    customers.name,
    ROUND(SUM(order_items.quantity * order_items.unit_price),
            2) AS total_revenue
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
        JOIN
    order_items ON orders.order_id = order_items.order_id
GROUP BY customers.name
ORDER BY total_revenue DESC
LIMIT 10;
```

Result Grid | Filter Rows:

name	total_revenue
Lori Garcia	20968.66
Jill Moore	20786.22
Michael Martinez	19113.2
Casey Stone	19106.68
Valerie Gray	18406.23
Carolyn Miller	17022.6
David Campos	16467.72
Timothy Duncan	16084.27
Kelly Campbell	16067.44
Jeremy Young	16000.41

AOV per Country

```sql
-- Determine the average order value (AOV) per country.

SELECT
    customers.country,
    ROUND(AVG(order_items.quantity * order_items.unit_price),
            2) AS avg_order_value
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
        JOIN
    order_items ON orders.order_id = order_items.order_id
GROUP BY customers.country
ORDER BY avg_order_value DESC;
```

Result Grid | Filter Rows:

country	avg_order_value
New Zealand	1021.16
Reunion	910.19
Timor-Leste	891.12
Angola	867.3
Djibouti	848.2
Kyrgyz Republic	840.37
Equatorial Guinea	830.37
Hungary	810.98
Estonia	803.9
Nepal	803.16
Falkland Islands (...	799.47
Tokelau	794.12
Vanuatu	786.31
Tonga	779.97

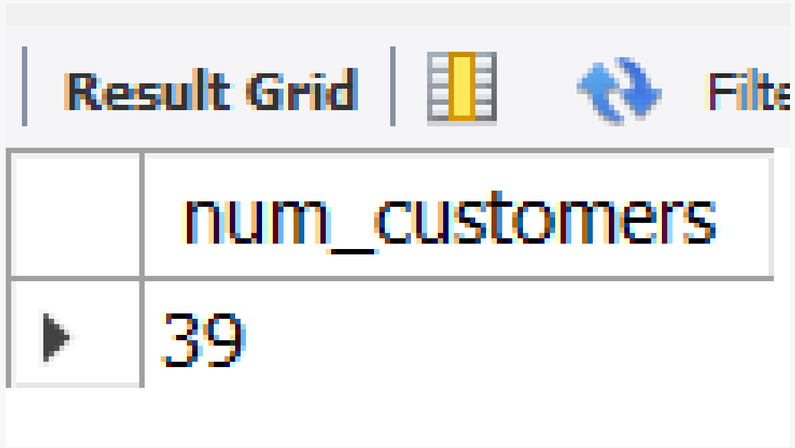
Result 1

Output

Customers with more than 5 orders

```sql
-- How many customers have placed more than 5 orders?

SELECT COUNT(*) AS num_customers
FROM (
    SELECT
        customers.customer_id
    FROM
        customers
    JOIN
        orders ON customers.customer_id = orders.customer_id
    GROUP BY
        customers.customer_id
    HAVING
        COUNT(DISTINCT orders.order_id) > 5
) AS sub;
```



	num_customers
▶	39

Monthly Order Count

```sql
-- Find the month-wise order count trend for the last 12 months.

SELECT
    DATE_FORMAT(order_date, '%b %Y') AS month_name,
    COUNT(*) AS order_count
FROM
    orders
WHERE
    order_date >= DATE_SUB(CURDATE(), INTERVAL 12 MONTH)
GROUP BY
    DATE_FORMAT(order_date, '%b %Y')
ORDER BY
    MIN(order_date);
```

Result Grid | Filter Rows:

month_name	order_count
Sep 2024	23
Oct 2024	28
Nov 2024	37
Dec 2024	47
Jan 2025	54
Feb 2025	31
Mar 2025	53
Apr 2025	45
May 2025	41
Jun 2025	36
Jul 2025	30

Total unit sold per product

```sql
-- Show total quantity of items sold per product (ordered descending).

SELECT
    products.product_name, SUM(order_items.quantity) AS quantity
FROM
    products
        JOIN
    order_items ON products.product_id = order_items.product_id
GROUP BY products.product_name
ORDER BY quantity DESC;
```

Result Grid | Filter Rows:

product_name	quantity
Books Item 25	126
Electronics Item 6	93
Books Item 21	92
Electronics Item 23	91
Books Item 20	91
Home & Kitchen Item 14	90
Electronics Item 37	89
Clothing Item 26	88
Clothing Item 24	88
Electronics Item 50	88
Clothing Item 19	88
Home & Kitchen Item 7	85

Result 1

Output

First-order in last 30 days

```sql
-- List customers who placed their first order in the last 30 days.

SELECT
    customers.name, MIN(orders.order_date) AS first_order
FROM
    customers
        JOIN
    orders ON customers.customer_id = orders.customer_id
GROUP BY customers.customer_id , customers.name
HAVING first_order >= CURDATE() - INTERVAL 30 DAY
ORDER BY first_order DESC;
```

Result Grid | Filter

	name	first_order

There is no customers who placed their first order in last 30 days

Completed vs Cancelled vs Returned

```sql
-- Show the number of completed vs cancelled vs returned orders by month.

SELECT
    orders.status, COUNT(*) AS num
FROM
    orders
GROUP BY orders.status;
```

Result Grid | Filter

status	num
Completed	371
Cancelled	72
Returned	57

Customer Lifetime Value (CLV)

```sql
-- Customer Lifetime Value (CLV) = total revenue per customer

SELECT
    c.customer_id,
    ROUND(SUM(oi.quantity * oi.unit_price), 2) AS customer_lifetime_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
GROUP BY c.customer_id
ORDER BY customer_lifetime_value DESC;
```

Result Grid | Filter Rows:

customer_id	customer_lifetime_value
53	20968.66
43	20786.22
75	19113.2
71	19106.68
4	18406.23
85	17022.6
90	16467.72
17	16084.27

Result 1

Top 3 revenue-generating products per category

```sql
-- Top 3 revenue-generating products per category

SELECT *
FROM (
    SELECT
        p.category,
        p.product_name,
        round(SUM(oi.quantity * oi.unit_price), 2) AS total_revenue,
        ROW_NUMBER() OVER (PARTITION BY p.category ORDER BY SUM(oi.quantity * oi.unit_price) DESC) AS rn
    FROM products p
    JOIN order_items oi ON p.product_id = oi.product_id
    GROUP BY p.category, p.product_name
) ranked
WHERE rn <= 3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

category	product_name	total_revenue	rn
Books	Books Item 25	62353.62	1
Books	Books Item 18	39420.8	2
Books	Books Item 8	32529.6	3
Clothing	Clothing Item 19	38243.04	1
Clothing	Clothing Item 29	37510.08	2
Clothing	Clothing Item 26	30400.48	3
Electronics	Electronics Item 23	28070.77	1
Electronics	Electronics Item 50	26706.24	2
Electronics	Electronics Item 6	26505.93	3
Home & Kitchen	Home & Kitchen Item 31	26165.91	1
Home & Kitchen	Home & Kitchen Item 14	24547.5	2
Home & Kitchen	Home & Kitchen Item 39	20535.32	3

Percentage contribution of each category to overall revenue

```sql
-- Percentage contribution of each category to overall revenue

SELECT
    p.category,
    ROUND(SUM(oi.quantity * oi.unit_price), 2) AS category_revenue,
    ROUND(SUM(oi.quantity * oi.unit_price) * 100.0 / (SELECT
                    SUM(quantity * unit_price)
                FROM
                    order_items),
            2) AS pct_contribution
FROM
    products p
        JOIN
    order_items oi ON p.product_id = oi.product_id
GROUP BY p.category;
```

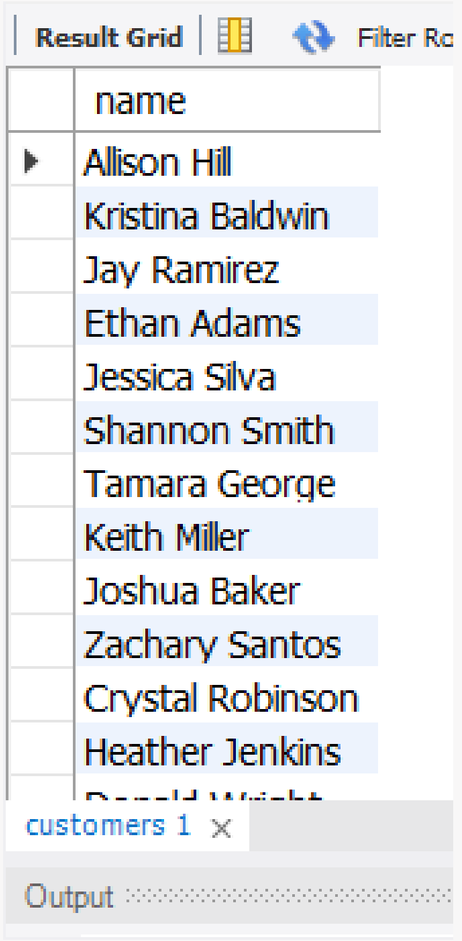
Result Grid | Filter Rows: | Export: | Wra

category	category_revenue	pct_contribution
Home & Kitchen	122419.61	12.73
Clothing	241844.57	25.14
Books	370027.91	38.47
Electronics	227517.54	23.66

Customers who have not placed any orders in the last 3 months

```sql
-- Customers who have not placed any orders in the last 3 months

SELECT
    c.name
FROM
    customers c
WHERE
    c.customer_id NOT IN (SELECT DISTINCT
            o.customer_id
        FROM
            orders o
        WHERE
            o.order_date >= DATE_SUB(CURDATE(), INTERVAL 3 MONTH));
```



Cumulative revenue trend month-by-month

```sql
-- Cumulative revenue trend month-by-month

SELECT
    DATE_FORMAT(o.order_date, '%Y-%m') AS month,
    round(SUM(oi.quantity * oi.unit_price), 2) AS monthly_revenue,
    round(SUM(SUM(oi.quantity * oi.unit_price))
    OVER (ORDER BY DATE_FORMAT(o.order_date, '%Y-%m')), 2) AS cumulati
FROM orders o
JOIN order_items oi ON o.order_id = oi.order_id
GROUP BY DATE_FORMAT(o.order_date, '%Y-%m')
ORDER BY month;
```

Result Grid | Filter Rows: | Export:

month	monthly_revenue	cumulative_revenue
2024-07	345.46	345.46
2024-08	81016.51	81361.97
2024-09	104971.83	186333.8
2024-10	72093.69	258427.49
2024-11	76331.5	334758.99
2024-12	96555.83	431314.82
2025-01	95163.24	526478.06
2025-02	49412.08	575890.14
2025-03	83042.39	658932.53
2025-04	92762.06	751694.59
2025-05	87506.19	839200.78
2025-06	61944.14	901144.92
2025-07	60664.71	961809.63

Orders with more than 3 different products

```sql
-- Orders with more than 3 different products

SELECT
    o.order_id,
    COUNT(DISTINCT oi.product_id) AS distinct_products
FROM
    orders o
        JOIN
    order_items oi ON o.order_id = oi.order_id
GROUP BY o.order_id
HAVING COUNT(DISTINCT oi.product_id) > 3;
```

Result Grid | Filter Rows:

order_id	distinct_products
4	4
8	4
9	4
10	4
12	5
20	5
24	5
31	5
32	5
36	4
37	5

Result 1

Rashedul Islam
Portfolio.

The End
Thank You.

Contact:
66.rashedul.islam@gmail.com