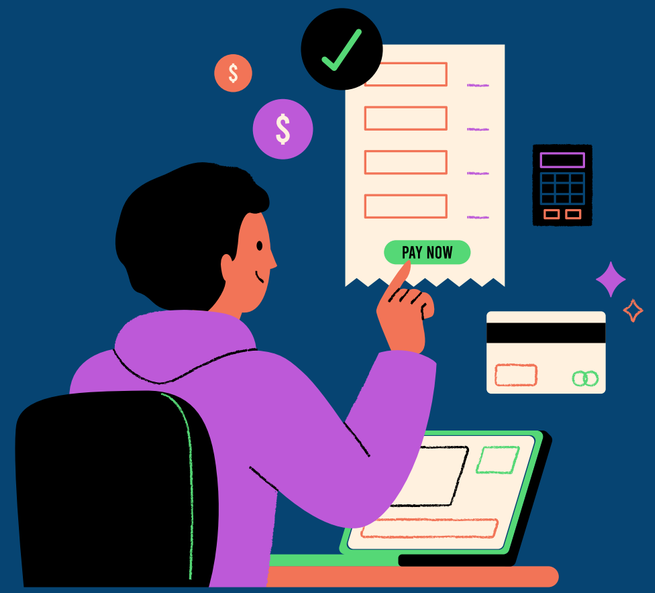
# PRODUCT & CUSTOMER ANALYSIS

## Data Analysis Project Using SQL




## Project Scope:

Derive insights to improve sales, customer engagement, and product strategy

Anas Ghouri

https://github.com/anas-ghouri

# 🎯 Project Objectives:

Product-Level Insights

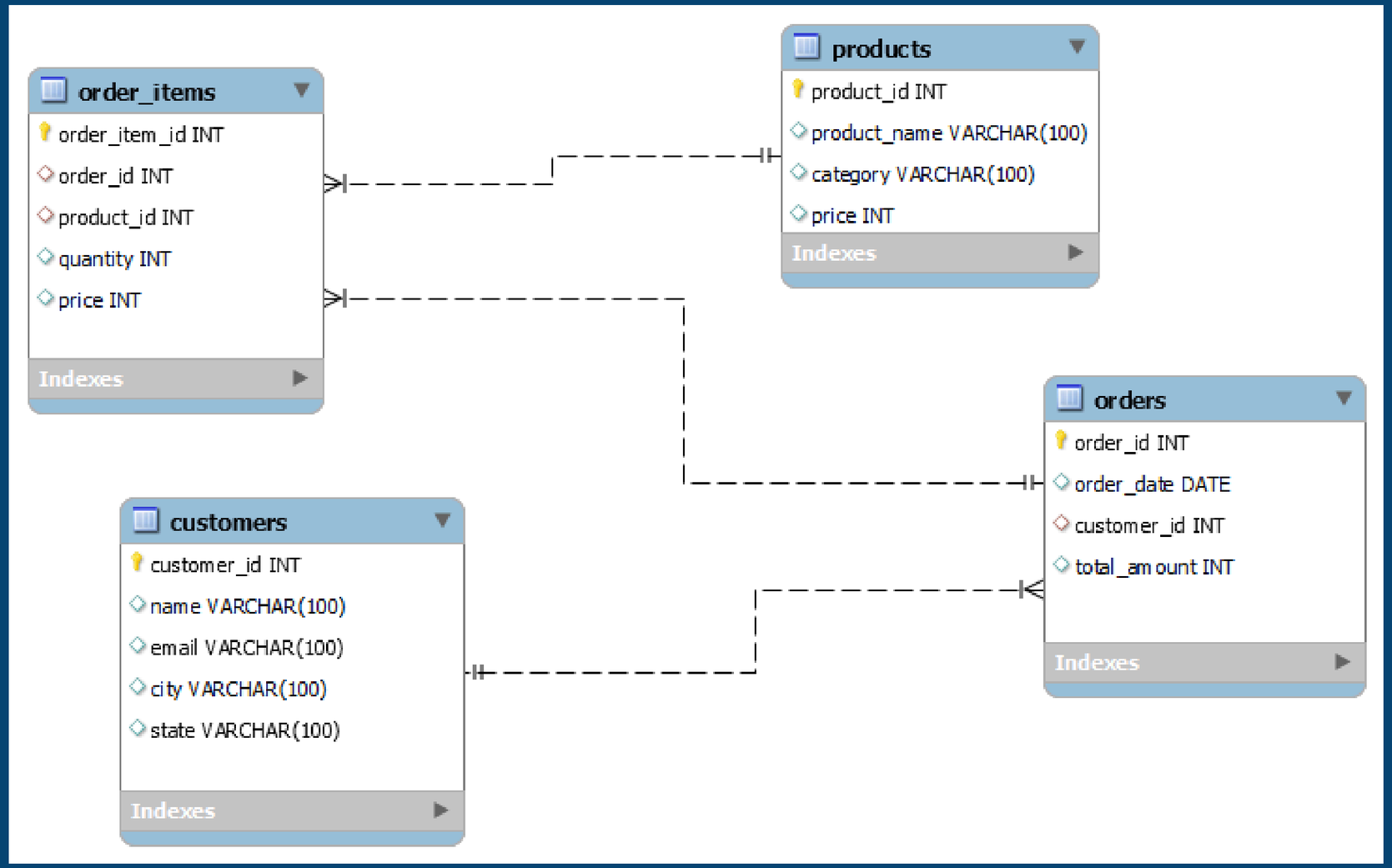
Customer Segmentation

Advanced KPIs (Customer Retention, Lifetime Value)

# Project Overview

Analyzed top-selling products, sales trends, customer segmentation, customer retention and overall business performance using SQL. Shared insights to optimize marketing and inventory strategies.

# Data Model


order_items
order_item_id INT
order_id INT
product_id INT
quantity INT
price INT
Indexes
products
product_id INT
product_name VARCHAR(100)
category VARCHAR(100)
price INT
Indexes
orders
order_id INT
order_date DATE
customer_id INT
total_amount INT
Indexes
customers
customer_id INT
name VARCHAR(100)
email VARCHAR(100)
city VARCHAR(100)
state VARCHAR(100)
Indexes

# What is the Average Order Value (AOV)?

```sql
SELECT ROUND(SUM(total_amount) / COUNT(order_id), 2) AS avg_order_value
FROM orders;
```

Result Grid

| | Avg_order_value |
|---|---|
| ▶ | 8980.0000 |

# What is the Monthly Revenue Trend?

```sql
SELECT
    DATE_FORMAT(order_date, '%m') AS month,
    SUM(total_amount) AS monthly_revenue
FROM orders
GROUP BY DATE_FORMAT(order_date, '%m')
ORDER BY month;
```

Result Grid | Filter Rows:

| | month | monthly_revenue |
|---|---|---|
| ▶ | 02 | 33700 |
| | 04 | 15000 |
| | 05 | 19700 |
| | 06 | 8300 |
| | 07 | 4900 |
| | 09 | 1000 |
| | 10 | 5500 |
| | 11 | 26200 |
| | 12 | 20400 |

# Top 5 most frequently purchased products?

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

| product_name | total_quantity_sold |
|---|---|
| Wireless Mouse | 34 |
| Notebook | 26 |
| Backpack | 19 |
| Ball Pen Pack | 18 |
| Bluetooth Speaker | 17 |

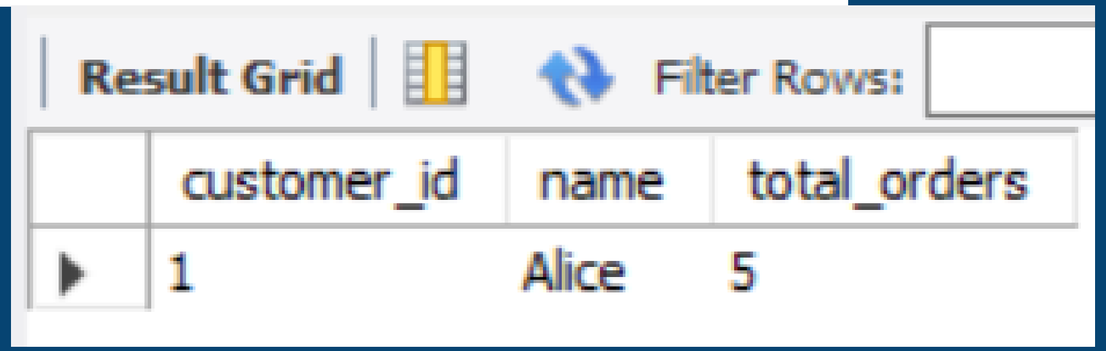
# Find the top best-selling product per category by quantity

```sql
with ranked_products as(
    SELECT
        p.category,
        p.product_name,
        SUM(oi.quantity) AS total_quantity,
        rank() OVER (PARTITION BY p.category ORDER BY SUM(oi.quantity) DESC) AS rank_in_category
    FROM products p
    JOIN order_items oi ON p.product_id = oi.product_id
    GROUP BY p.category, p.product_name
)
SELECT category, product_name, total_quantity
FROM ranked_products
WHERE rank_in_category = 1;
```

| category | product_name | total_quantity |
|---|---|---|
| Accessories | Backpack | 19 |
| Electronics | Wireless Mouse | 34 |
| Stationery | Notebook | 26 |

# Which customers have placed the most orders?

```sql
SELECT c.customer_id, c.name, COUNT(o.order_id) AS total_orders
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name
ORDER BY total_orders DESC
limit 1;
```




| customer_id | name | total_orders |
|---|---|---|
| 1 | Alice | 5 |

# Which customers have generated the highest total revenue?

```sql
select c.customer_id, c.name, sum(o.total_amount) as total_spend
from customers c
join orders o
on c.customer_id=o.customer_id
group by c.customer_id, c.name
order by total_spend desc
limit 1;
```

Result Grid | Filter Rows:

| customer_id | name | total_spend |
|---|---|---|
| 2 | Bob | 35300 |

# Total revenue each customer has contributed

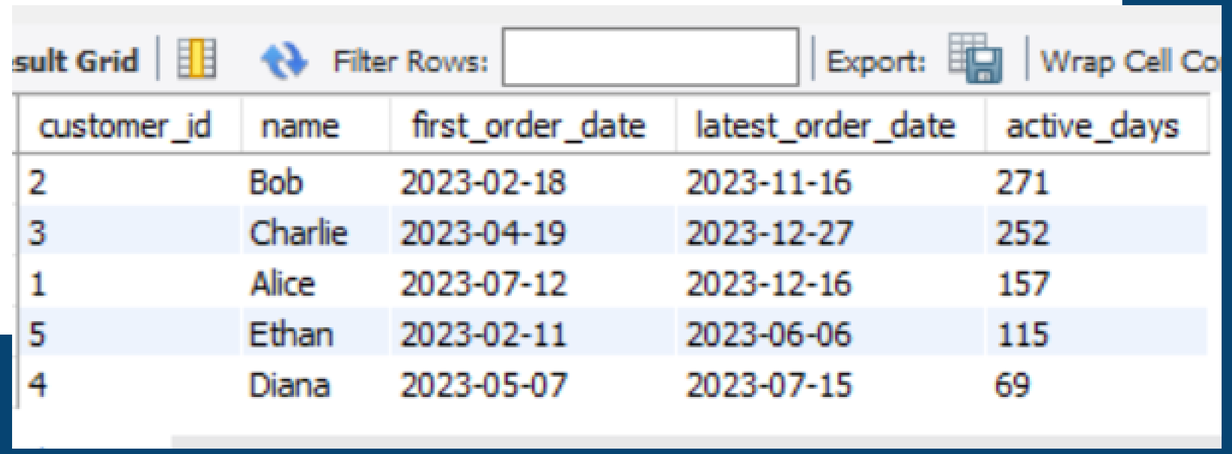
```sql
-- (Total revenue each customer has contributed to date)
SELECT c.customer_id, c.name, SUM(o.total_amount) AS lifetime_value
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
GROUP BY c.customer_id, c.name
ORDER BY lifetime_value DESC;
```

esult Grid | Filter Rows:

| customer_id | name | lifetime_value |
|---|---|---|
| 2 | Bob | 35300 |
| 1 | Alice | 29100 |
| 3 | Charlie | 28800 |
| 5 | Ethan | 21200 |
| 4 | Diana | 20300 |

# Find the first and latest order date of each customer, also find active days of each customer

```sql
select c.customer_id, c.name,
min(o.order_date) as first_order_date,
max(o.order_date) as latest_order_date,
DATEDIFF(MAX(o.order_date), MIN(o.order_date)) AS active_days
from customers c
join orders o
on c.customer_id=o.customer_id
group by c.customer_id, c.name
order by active_days desc;
```



sult Grid | Filter Rows: | Export: | Wrap Cell Co

| customer_id | name | first_order_date | latest_order_date | active_days |
|---|---|---|---|---|
| 2 | Bob | 2023-02-18 | 2023-11-16 | 271 |
| 3 | Charlie | 2023-04-19 | 2023-12-27 | 252 |
| 1 | Alice | 2023-07-12 | 2023-12-16 | 157 |
| 5 | Ethan | 2023-02-11 | 2023-06-06 | 115 |
| 4 | Diana | 2023-05-07 | 2023-07-15 | 69 |

# Segment customers based on lifetime value:

```sql
select c.customer_id, c.name, sum(o.total_amount) as lifetime_value,
case
    when sum(o.total_amount) > 30000 then 'Gold'
    when sum(o.total_amount) between 25000 and 30000 then 'Silver'
    else 'Bronze'
end as customer_tier
from customers c
join orders o
on c.customer_id=o.customer_id
group by c.customer_id, c.name
order by lifetime_value desc;
```

| customer_id | name | lifetime_value | customer_tier |
|---|---|---|---|
| 2 | Bob | 35300 | Gold |
| 1 | Alice | 29100 | Silver |
| 3 | Charlie | 28800 | Silver |
| 5 | Ethan | 21200 | Bronze |
| 4 | Diana | 20300 | Bronze |

# Customer Retention Analysis (Repeat Customers by Month)

```sql
WITH orders_with_months AS (
    SELECT
        customer_id,
        DATE_FORMAT(order_date, '%Y-%m') AS order_month,
        MIN(DATE_FORMAT(order_date, '%Y-%m')) OVER (PARTITION BY customer_id) AS first_order_month
    FROM orders
)
SELECT
    order_month,
    COUNT(DISTINCT customer_id) AS repeat_customers
FROM orders_with_months
WHERE order_month > first_order_month
GROUP BY order_month
ORDER BY order_month;
```

| order_month | repeat_customers |
|---|---|
| 2023-06 | 1 |
| 2023-07 | 1 |
| 2023-09 | 1 |
| 2023-10 | 1 |
| 2023-11 | 2 |
| 2023-12 | 2 |

# THANK YOU!