Marketing Analytics Business Case

Harsh Wardhan Singh

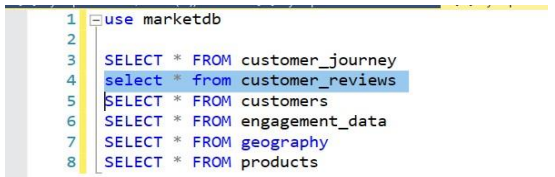
Day -1

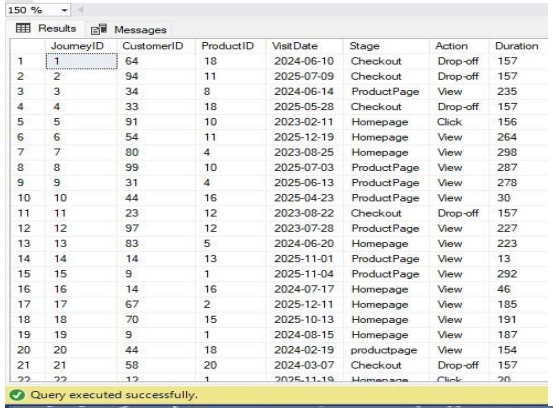
These tables give me a solid foundation:

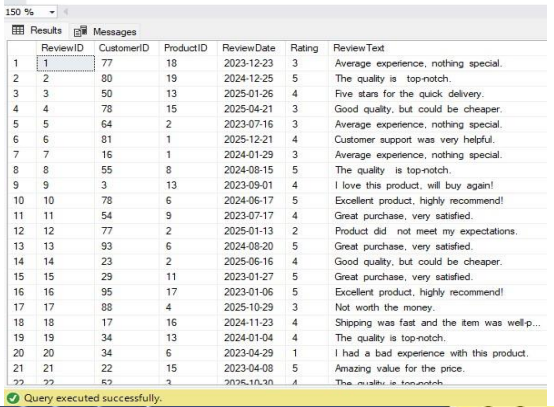
customer_journey: shows how users move through different marketing stages.

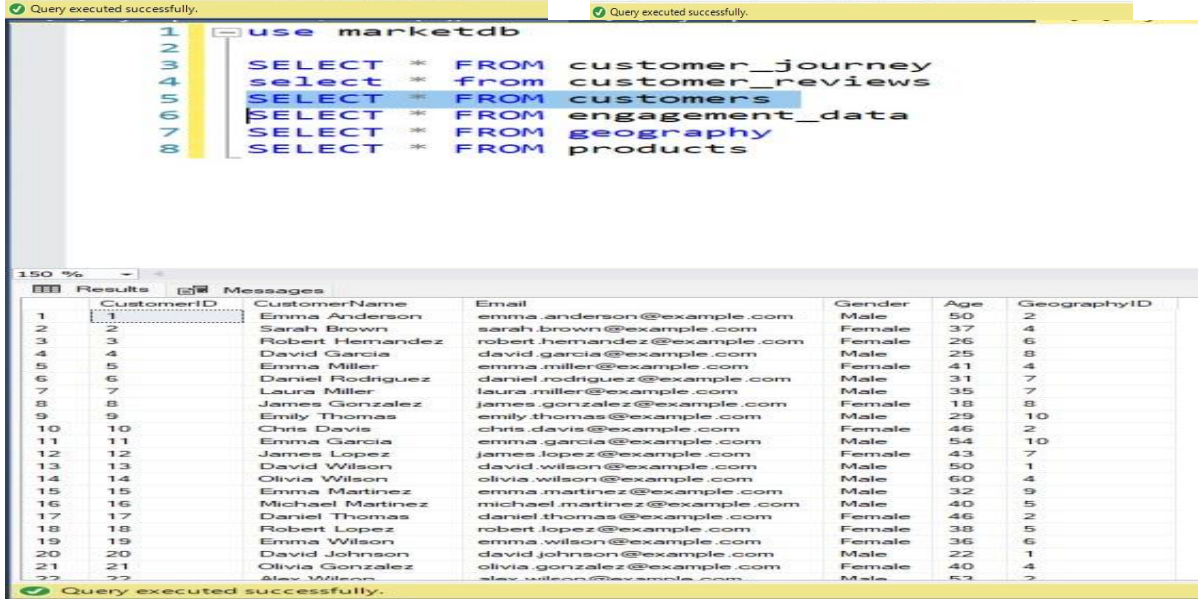
customer_reviews: contains feedback, which might reveal pain points.

customers: gives context like demographics or geography.









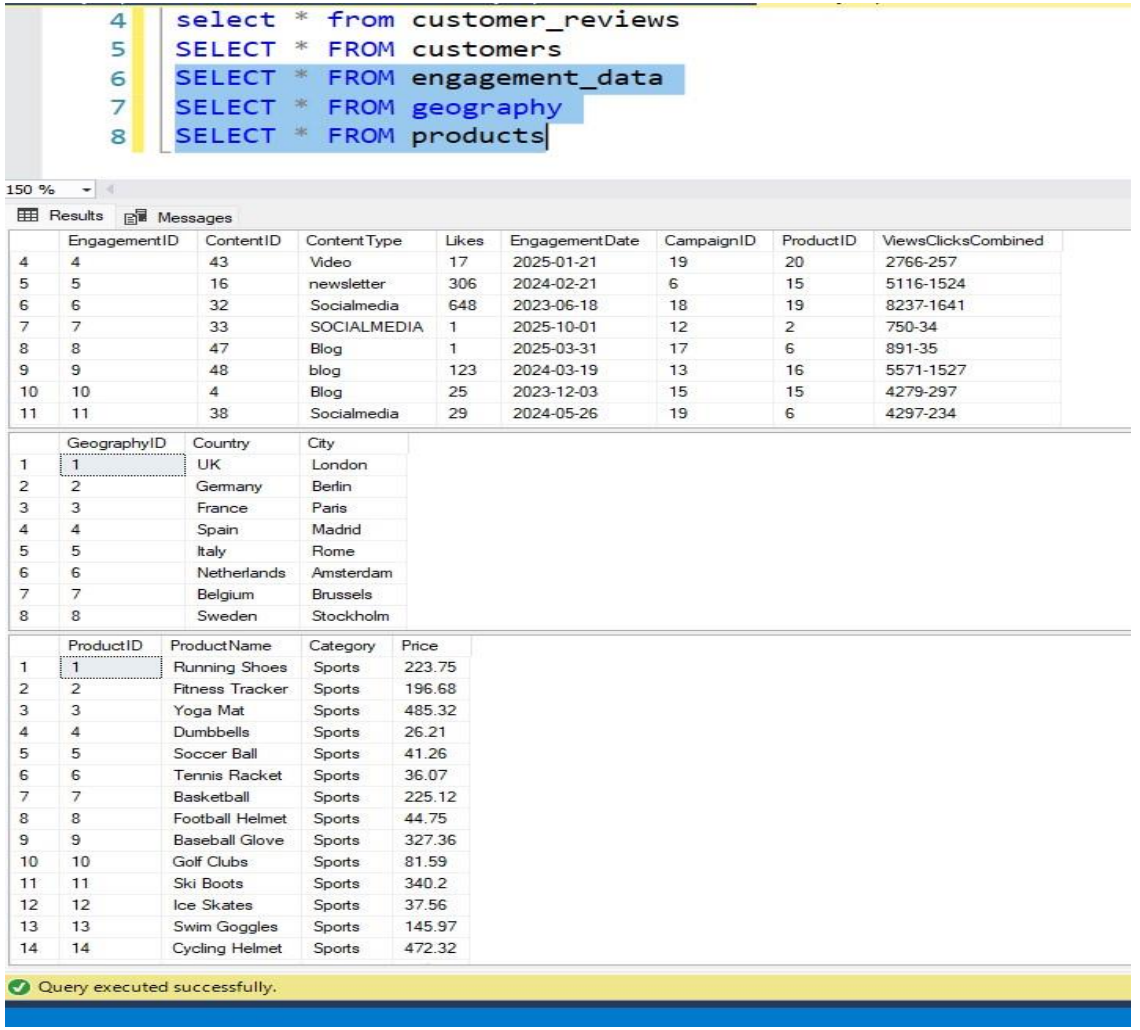
Day -1

Engagement_data: This tracks customer interactions like views, clicks, likes — perfect for measuring engagement levels.

Geography: Gives me location data to analyze engagement by country or city.

Products: Helps me understand what types of products are linked to higher or lower engagement.

This step is all about understanding what data I have before jumping into deeper analysis.



```
4  select * from customer_reviews
5  SELECT * FROM customers
6  SELECT * FROM engagement_data
7  SELECT * FROM geography
8  SELECT * FROM products
```

150 %

Results | Messages

	EngagementID	ContentID	ContentType	Likes	EngagementDate	CampaignID	ProductID	ViewsClicksCombined
4	4	43	Video	17	2025-01-21	19	20	2766-257
5	5	16	newsletter	306	2024-02-21	6	15	5116-1524
6	6	32	Socialmedia	648	2023-06-18	18	19	8237-1641
7	7	33	SOCIALMEDIA	1	2025-10-01	12	2	750-34
8	8	47	Blog	1	2025-03-31	17	6	891-35
9	9	48	blog	123	2024-03-19	13	16	5571-1527
10	10	4	Blog	25	2023-12-03	15	15	4279-297
11	11	38	Socialmedia	29	2024-05-26	19	6	4297-234

	GeographyID	Country	City
1	1	UK	London
2	2	Germany	Berlin
3	3	France	Paris
4	4	Spain	Madrid
5	5	Italy	Rome
6	6	Netherlands	Amsterdam
7	7	Belgium	Brussels
8	8	Sweden	Stockholm

	ProductID	ProductName	Category	Price
1	1	Running Shoes	Sports	223.75
2	2	Fitness Tracker	Sports	196.68
3	3	Yoga Mat	Sports	485.32
4	4	Dumbbells	Sports	26.21
5	5	Soccer Ball	Sports	41.26
6	6	Tennis Racket	Sports	36.07
7	7	Basketball	Sports	225.12
8	8	Football Helmet	Sports	44.75
9	9	Baseball Glove	Sports	327.36
10	10	Golf Clubs	Sports	81.59
11	11	Ski Boots	Sports	340.2
12	12	Ice Skates	Sports	37.56
13	13	Swim Goggles	Sports	145.97
14	14	Cycling Helmet	Sports	472.32

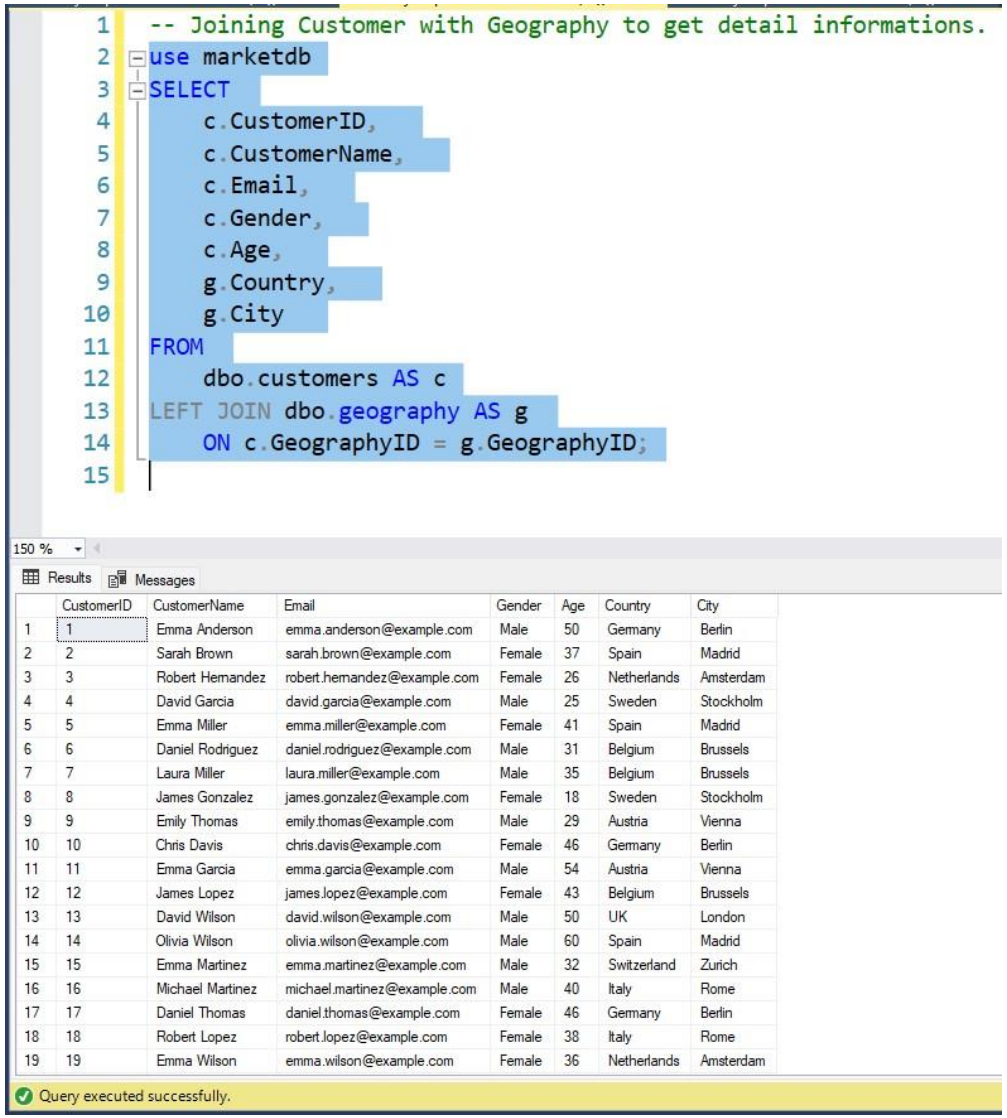
Query executed successfully.

Day -1

Connecting the Dots with Joins.

I started joining tables to enrich the customer data with more context.

I used a simple **LEFT JOIN** to connect **customers** with **geography** and pull in location details like country and city



```sql
-- Joining Customer with Geography to get detail informations.
use marketdb
SELECT
    c.CustomerID,
    c.CustomerName,
    c.Email,
    c.Gender,
    c.Age,
    g.Country,
    g.City
FROM
    dbo.customers AS c
LEFT JOIN dbo.geography AS g
    ON c.GeographyID = g.GeographyID;
```

150 %

Results | Messages

	CustomerID	CustomerName	Email	Gender	Age	Country	City
1	1	Emma Anderson	emma.anderson@example.com	Male	50	Germany	Berlin
2	2	Sarah Brown	sarah.brown@example.com	Female	37	Spain	Madrid
3	3	Robert Hernandez	robert.hernandez@example.com	Female	26	Netherlands	Amsterdam
4	4	David Garcia	david.garcia@example.com	Male	25	Sweden	Stockholm
5	5	Emma Miller	emma.miller@example.com	Female	41	Spain	Madrid
6	6	Daniel Rodriguez	daniel.rodriguez@example.com	Male	31	Belgium	Brussels
7	7	Laura Miller	laura.miller@example.com	Male	35	Belgium	Brussels
8	8	James Gonzalez	james.gonzalez@example.com	Female	18	Sweden	Stockholm
9	9	Emily Thomas	emily.thomas@example.com	Male	29	Austria	Vienna
10	10	Chris Davis	chris.davis@example.com	Female	46	Germany	Berlin
11	11	Emma Garcia	emma.garcia@example.com	Male	54	Austria	Vienna
12	12	James Lopez	james.lopez@example.com	Female	43	Belgium	Brussels
13	13	David Wilson	david.wilson@example.com	Male	50	UK	London
14	14	Olivia Wilson	olivia.wilson@example.com	Male	60	Spain	Madrid
15	15	Emma Martinez	emma.martinez@example.com	Male	32	Switzerland	Zurich
16	16	Michael Martinez	michael.martinez@example.com	Male	40	Italy	Rome
17	17	Daniel Thomas	daniel.thomas@example.com	Female	46	Germany	Berlin
18	18	Robert Lopez	robert.lopez@example.com	Female	38	Italy	Rome
19	19	Emma Wilson	emma.wilson@example.com	Female	36	Netherlands	Amsterdam

Query executed successfully.

Day -1

Categorizing Products by Price :-

I took a closer look at the product data and decided to segment products based on their pricing.

I used SQL's **CASE WHEN** statement to do this. It works kind of like an "if-else" in programming — helping you create new categories or labels based on specific conditions.

This kind of logic makes raw numerical data a lot more interpretable and useful in business analysis.

```sql
use marketdb

-- Categorize the Product on basis of Price

SELECT * FROM products

SELECT
    PRODUCTID,
    PRODUCTNAME,
    PRICE,

    CASE
        WHEN PRICE < 50 THEN 'Low Price'
        WHEN PRICE BETWEEN 50 AND 200 THEN 'Medium Price'
        ELSE 'High Price'

    END AS PriceCategory

FROM products
```

150 %

Results | Messages

	PRODUCTID	PRODUCTNAME	PRICE	PriceCategory
1	1	Running Shoes	223.75	High Price
2	2	Fitness Tracker	196.68	Medium Price
3	3	Yoga Mat	485.32	High Price
4	4	Dumbbells	26.21	Low Price
5	5	Soccer Ball	41.26	Low Price
6	6	Tennis Racket	36.07	Low Price
7	7	Basketball	225.12	High Price
8	8	Football Helmet	44.75	Low Price
9	9	Baseball Glove	327.36	High Price
10	10	Golf Clubs	81.59	Medium Price
11	11	Ski Boots	340.2	High Price
12	12	Ice Skates	37.56	Low Price
13	13	Swim Goggles	145.97	Medium Price
14	14	Cycling Helmet	472.32	High Price
15	15	Climbing Rope	410.17	High Price
16	16	Kayak	259.4	High Price

Query executed successfully.

Thank You!