

Data Cleaning Steps



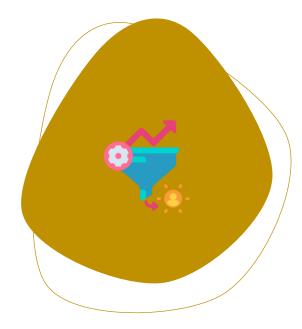
Validation of Data Integrity

Conducted checks for missing values, duplicates, and negative entries. No such anomalies were found, confirming data integrity.



Standardization of Date Format

Converted inconsistent date formats to YYYY-MM-DD.



Converting
Acquisition Cost to
Numeric Format

Removed dollar signs and commas from acquisition_cost and converted the data to numeric for aggregation.

Calculate Total Impressions 8 for Each Campaign Identify the Campaign with 2 the Highest ROI Analytical Queries Find the Top 3 Locations 6 and with the Most Impressions **Findings** 4

Rank Channels by Total Conversions

Find Campaigns with CTR Above a Threshold

Find the Most Cost-Effective Campaign

Calculate Average Engagement Score by Target Audience

Calculate the Overall CTR (Click-Through Rate)



Calculate Total Impressions for Each Campaign.

Expected Output: A table with campaign_id and total impressions.

```
SELECT
    campaign_id,
    SUM(impressions) AS total_impressions
FROM
    marketing_campaign
GROUP BY
    campaign_id
-- I needed to cast the campaign_id as integer so as to be able to order it in numerical order
ORDER BY
CAST
    (campaign_id AS INT);
```

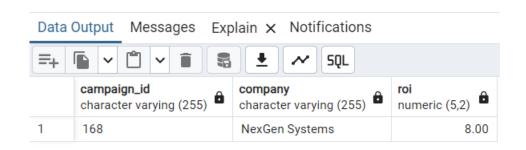
Data Output Messages Explain × Notifications					
= + [<u>▼</u> ~ SQL			
	campaign_id character varying (255)	total_impressions bigint			
1	1	1922			
2	2	7523			
3	3	7698			
4	4	1820			
5	5	4201			
6	6	1643			
7	7	8749			
8	8	7854			
9	9	1754			
10	10	3856			
11	11	6628			
12	12	8948			
13	13	8817			
14	14 2201				
15	15	8470			
16	16	1512			
17	17	2488			
18	18	9264			
19	19	5798			
20	20	3677			
21	21	1561			
Total rows: 1000 of 200005		Query complete 00:00:00.384			



Identify the Campaign with the Highest ROI.

Expected Output: A single row with campaign_id, company, and roi.

```
SELECT
    campaign_id,
    company,
    roi
FROM
    marketing_campaign
ORDER BY
    roi DESC
LIMIT 1;
```

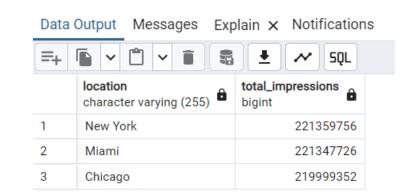




Find the Top 3 Locations with the Most Impressions.

Expected Output: A table with location and total impressions.

```
SELECT
location,
SUM (impressions) AS total_impressions
FROM
marketing_campaign
GROUP BY
location
ORDER BY
total_impressions DESC
LIMIT 3;
```





Calculate Average Engagement Score by Target Audience.

Expected Output: A table with target_audience and averagegengagementscore.

SELECT target_audience, AVG (engagement_score) AS average_enagagement_score FROM marketing_campaign GROUP BY target_audience;

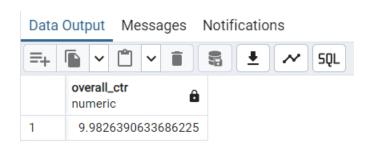
Data	Output Messages No	Messages Notifications	
=+		<u> </u>	
	target_audience character varying (255)	average_enagagement_score numeric	
1	All Ages	5.4868693935683766	
2	Men 18-24	5.5150152760873345	
3	Men 25-34	5.4919798121127324	
4	Women 25-34	5.4927398595456477	
5	Women 35-44	5.4865702479338843	



Calculate the Overall CTR (Click-Through Rate).

Expected Output: A single value for the overall CTR.

SELECT
 SUM(clicks) * 100.0 / SUM(impressions) AS overall_CTR
FROM marketing_campaign;



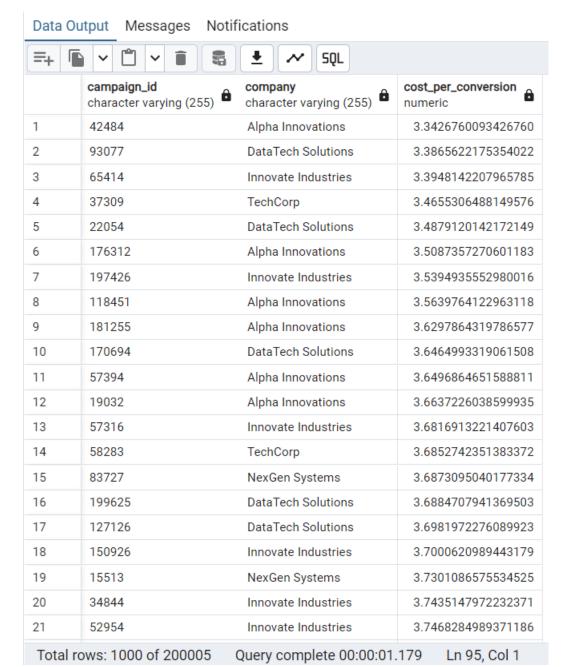


Find the Most Cost-Effective Campaign.

Expected Output: A table with campaign_id, company, and costperconversion.

SELECT

```
campaign_id,
  company,
  acquisition_cost/(conversion_rate * impressions) AS cost_per_conversion
FROM
  marketing_campaign
ORDER BY
  cost_per_conversion ASC;
```





Find Campaigns with CTR Above a Threshold.

Expected Output: A table with campaign_id, company, and ctr.

```
SELECT
    campaign_id,
    company,
    ctr
FROM
SELECT
    campaign_id,
    company,
    SUM(clicks) * 100.0 / SUM(impressions) AS ctr
FROM
    marketing_campaign
GROUP BY
    campaign_id,
    company
WHERE
    ctr > 50
ORDER BY
    ctr
DESC;
```

Data	Data Output Messages Notifications					
=+	=+					
	campaign_id character varying (255)	company character varying (255)	ctr numeric			
1	122375	TechCorp	99.2023928215353938			
2	26330	Alpha Innovations	99.2000000000000000			
3	133972	Innovate Industries	99.0049751243781095			
4	121860	Alpha Innovations	99.0009990009990010			
5	171192	Alpha Innovations	98.4047856430707876			
6	65535	DataTech Solutions	98.3218163869693978			
7	77443	Innovate Industries	98.1280788177339901			
8	173975	NexGen Systems	97.2194637537239325			
9	67651	DataTech Solutions	97.0472440944881890			
10	14262	TechCorp	96.9902912621359223			
11	63985	NexGen Systems	96.9518190757128810			
12	165410	Alpha Innovations	96.9184890656063618			
13	84190	Innovate Industries	96.4071856287425150			
14	180495	Alpha Innovations	96.2818003913894325			
15	34234	Alpha Innovations	96.1500493583415597			
16	152721	NexGen Systems	96.1127308066083576			
17	78109	Innovate Industries	96.0386473429951691			
18	84625	TechCorp	95.9921798631476051			
19	60669	Alpha Innovations	95.9381044487427466			
20	142773	TechCorp	95.6777996070726916			
Total rows: 1000 of 6166						



Rank Channels by Total Conversions.

Expected Output: A table with channel_used and total conversions.

```
SELECT
    channel_used,
    SUM (conversion_rate * impressions) AS total_conversions,
    RANK () OVER (ORDER BY SUM (conversion_rate * impressions) DESC) AS rank
FROM
    marketing_campaign
GROUP BY
    channel_used;
```

Data Output Messages Notifications						
=+ L ~ L ~ E L ~ SQL						
	channel_used character varying (255)	total_conversions numeric	rank bigint			
1	Google Ads	14804993.60	1			
2	Email	14796855.85	2			
3	Website	14762165.57	3			
4	YouTube	14661901.97	4			
5	Instagram	14657395.58	5			
6	Facebook	14451723.86	6			



THANK YOU -