OPTIMIZING MARKETING STRATEGIES: A MARKETING CAMPAIGN ANALYSIS

INTRODUCTION

Summary

In today's competitive landscape, marketing success hinges on data-driven decisions. This report tells the story of how we analyzed a Marketing Campaign Dataset using SQL to uncover actionable insights. By exploring key metrics like impressions, clicks, ROI, and conversions, we identified which campaigns, channels, and locations delivered the best results—empowering businesses to optimize budgets and strategies.

Problem Statement

Marketing campaigns are costly, and their effectiveness is often unclear. Businesses struggle with questions like:

- Which campaigns are reaching the widest audience?
- Where should budgets be allocated for maximum ROI?
- Which channels or locations drive the most conversions?

This analysis tackles these challenges, transforming raw data into a roadmap for smarter marketing decisions.

Outline of the Report

- **The Data**: A glance at the dataset and its role in solving the problem.
- **The Method**: How SQL and data cleaning paved the way for insights.
- **The Analysis**: Key discoveries about campaigns, channels, and ROI.
- **The Results**: Actionable takeaways to refine future strategies

DATA: UNDERSTANDING THE CAMPAIGN LANDSCAPE

Dataset Overview

The dataset contains performance metrics for diverse marketing campaigns, including:

- **Core Metrics**: Impressions, clicks, conversions, ROI.
- **Campaign Context**: Target audience, channels used, location, budget.

For full column descriptions, see Appendix.

Why It Matters

This data lets us compare campaigns objectively, answering critical questions about cost-effectiveness, audience engagement, and geographic impact.

METHODS: FROM RAW DATA TO INSIGHTS

Tools & Process

- **Preprocessing**: Python cleaned the data (e.g., standardizing dates, fixing decimal inconsistencies).
- **SQL Analysis**: Queries in PostgreSQL identified trends (e.g., top-performing campaigns, ROI leaders).

Key Steps

- **Data Cleaning**: Ensured accuracy by handling missing values and formatting issues.
- **Database Setup**: Created a PostgreSQL database to store and query the dataset.
- **Query Execution**: Extracted insights like CTR, cost-per-conversion, and regional performance.

Technical details (e.g., SQL schemas, full queries) are in the Appendix.

ANALYSIS: UNCOVERING THE STORY IN THE DATA

We explored the dataset through 8 key SQL queries (See Appendix), transforming raw metrics into strategic insights. Here's how the story unfolded:

1. **Campaign Reach & Visibility**
   - **Total Impressions by Campaign**: Identified campaigns with the broadest reach (e.g., Campaign #X led with 1.2M impressions).
   - **Top Locations**: New York, Miami, and Chicago dominated impressions, suggesting untapped potential in these markets.
   - **CTR Performance**: 160,000+ campaigns exceeded the 5% CTR benchmark, with email campaigns outperforming Facebook.
2. **ROI & Cost Efficiency**
   - **ROI Champion**: NexGen Systems emerged as the top performer with an 8.0 ROI.
   - **Cost Per Conversion**: Alpha Solutions' campaign achieved conversions at just $34 each, setting a benchmark for efficiency.
3. Audience & Channel Insights
   - **Engagement Trends**: Men (18–24) showed the highest engagement, while Women (36–44) lagged.
   - **Channel Effectiveness**: Email drove 62% of conversions, outperforming social media and other channels.

RESULTS & INSIGHTS: THE BIG PICTURE

1. **Location Matters**:
   - New York, Miami, and Chicago accounted for 662 million impressions—prioritize these regions for future campaigns.
2. **ROI & Efficiency Wins**:
   - NexGen Systems' 8.0 ROI highlights the value of targeted messaging.
   - Alpha Solutions' $34 cost-per-conversion model offers a blueprint for budget optimization.
3. **Audience & Channel Gaps**:
   - Men (18–24): High engagement suggests opportunities for loyalty programs.
   - Email Marketing: Dominated conversions—double down on personalized email strategies.
   - Facebook Underperformance: Requires creative/content reevaluation.
4. **CTR Success:**
   - A 9.98% overall CTR indicates strong audience relevance—sustain this through A/B testing.

See Appendix for more

RECOMMENDATIONS: FROM INSIGHTS TO ACTION

1. **Amplify High-Performing Campaigns**
   - Replicate NexGen Systems' strategies (high ROI) and Alpha Solutions' cost-efficiency tactics.
   - Allocate 30% of budget to New York, Miami, and Chicago for maximum visibility.
2. **Revitalize Underperforming Segments**
   - Redesign campaigns targeting Women (36–44) using A/B-tested creatives.
   - Audit Facebook ad content (e.g., visuals, CTAs) to match email's conversion success.
3. **Leverage Email Dominance**
   - Invest in dynamic email personalization tools to boost conversions further.
   - Test hybrid campaigns (e.g., email + social retargeting).

4. **Monitor & Iterate**
    - Track Men (18–24) engagement monthly to maintain momentum.
    - Set quarterly CTR benchmarks (target: 12% by 2025).

APPENDIX

Column Descriptions

Below us a detailed description of the columns in the dataset:

- **Campaign_ID**: A unique identifier assigned to each campaign.
- **Company**: The company responsible for the campaign, representing a mix of fictional brands.
- **Campaign_Type**: The type of campaign employed such as email, social media, influencer, display or search.
- **Target_Audience**: The specific audience segment targeted by the campaign, including demographics like women aged 25-34, men aged 18-24 or all age groups.
- **Duration**: The length of the campaign in days.
- **Channels_Used**: The marketing channels utilized, such as email, social media, YouTube, websites or Google Ads.
- **Conversion_Rate**: The percentage of leads or impressions that converted into desired actions, including campaign effectiveness.
- **Acquisition_Cost**: The cost incurred by the company to acquire customers, presented in monetary format.
- **ROI**: Return On Investment, representing the profitability and success of the campaign.
- **Location**: The geographical area where the campaign took place, covering major cities such as New York, Los Angeles, Chicago, Houston, or Miami,
- **Date**: The specific date on which the campaign was executed helping track trends over time.
- **Clicks**: The number of clicks generated by the campaign, representing user engagement.
- **Impressions**: The total number of times the campaign was displayed or viewed by the target audience.
- **Engagement_Score**: A score from 1-10 that quantifies how much engagement a campaign received.
- **Customer_Segment**: The specific customer segment or audience category the campaign was designed for (e.g. tech enthusiasts, fashionistas, health-conscious individuals, food lovers, or outdoor adventurers).

Technical Details

Data Collection & Tools Used

The dataset used in this analysis Marketing Campaign Data was provided in a structured format and stored in a CSV format. The following tools were used:

- **Jupyter Notebook**: For data preprocessing ad cleaning.
- **PostgresSQL**: The relational database system used for data storage and querying
- **pgAdmin**: The graphical interface for managing and querying the database
- **Python**: Used for preprocessing, cleaning and crating cleaned version of data for analysis.

1. **Data Preprocessing**
   Before performing SQL analysis, the raw dataset underwent preprocessing in Python using Jupyter-Notebook. The following steps were taken to ensure data quality and consistency:
   a) Handling Missing Values: Checked for NULL values in all columns. The dataset presented no NULLs.
   b) Data Type Conversions:
      - Ensured numerical fields were correctly formatted as floats or numeric types
      - Standardized date values in Date column to ensure uniform Date formats and converted to DateTime data type.
      - Standardized The ROI columns which had inconsistent decmal places to enforce a uniform two-decimal format to maintain consistency.
2. **Exported Clean Dataset**: Exported cleaned and preprocessed dataset in csv format ready to be imported into pgAdmin to be used to create table for analysis.
3. **Database Creation in PostgresSQL**

After data preprocessing, the cleaned dataset was importe into PPostgresSQL for structure querying and analysis. The following steps outline the database creation process un pgAdmin.

- Creating the Database (marketing_campaign)
  To store the cleaned dataset, a new PostgresSQL database was vreated using the following SQL command in pgAdmin:

```
-- Database: marketing_campaign
-- DROP DATABASE IF EXISTS marketing_campaign;
CREATE DATABASE marketing_campaign
  WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'en-US'
  LC_CTYPE = 'en-US'
  LOCALE_PROVIDER = 'libc'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1
  IS_TEMPLATE = False;
```

4. **Creating the ‘*campaigndata*’ Table**
   Once the database was set up and a connection finally made to the database, a table structure was defined to start the dataset. The campaigndata table was created using the following SQL script:

```
CREATE TABLE campaigndata (
  Campaign_ID INT PRIMARY KEY,
  Company TEXT,
  Campaign_Type TEXT,
  Target_Audience TEXT,
  Duration TEXT,
  Channel_Used TEXT,
  Conversion_Rate FLOAT,
  Acquisition_Cost MONEY,
  ROI FLOAT,
  Location TEXT,
  Date TIMESTAMP,
  Clicks INT,
  Impressions INT,
  Engagement_Score INT,
  Customer_Segment TEXT
);
```

Explanation of Table Schema

- **Campaign_ID**: Primary key, uniquely identifying each campaign.
- **Company**: Name of the company running the campaign.
- **Campaign_Type**: Specifies the type of marketing campaign (e.g., Social Media, Email, Influencer, etc.).
- **Target_Audience**: Defines the demographic targeted.
- **Duration**: Number of days the campaign lasted.
- **Channel_Used**: The primary channel used for marketing.
- **Conversion_Rate**: Percentage of impressions that converted into desired actions.
- **Acquisition_Cost**: Cost incurred to acquire customers (stored in MONEY format).
- **ROI**: Return on Investment for the campaign.
- **Location**: The geographical area where the campaign was conducted.

- **Date**: Date the campaign was executed (stored as TIMESTAMP).
- **Clicks**: Number of clicks generated by the campaign.
- **Impressions**: Number of times the campaign was displayed to users.
- **Engagement_Score**: Score from 1 to 10 measuring audience interaction.
- **Customer_Segment**: Audience category targeted (e.g., Tech Enthusiasts, Foodies, etc.).

Importing Data into PostgreSQL

To populate the CAmpaigndata table, the cleaned dataset (stored in CSV file) was loaded using the COPY command.

```
COPY campaigndata FROM 'C:\Users\USER\Desktop\marketing_campaign_SQL_data.csv'
DELIMITER ','
CSV HEADER;
```

Verifying the Data Import: After executing the COPY command, the following query was run to verify that the data was successfully imported:

```
SELECT * FROM campaigndata LIMIT 10;
```

If the import was successful, he first 10 rows of the dataset were displayed.

QUERIES

Below are the queries executed along with their objectives:

a) **Calculating Total Impressions for Each Campaign**

Objective: To identify the total number of impressions each campaign received.

```
SELECT campaign_id, impressions AS totalImpressions
FROM campaigndata
GROUP BY campaign_iD
ORDER BY impressions DESC;
```

Purpose: Determines the reach of each campaign based on the impressions.

b) **Identifying the Campaign with the Highest ROI**

Objective:Find the most profitable campaign based on Return On Investment(ROI)

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

Purpose: Helps business identify the most successful campaign in terms of ROI.

c) **Finding the top 3 locations with the Most Impressions**

Objective: Identify the locations where the campaigns were viewed the most.

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

Purpose: Helps understand high-engagement locations for future targeting.

d) **Calculating the Average Engagement Score by Target Audience:**

Objective: Determine the engagement levels for different audience segments.

```
SELECT target_audience,
    AVG(engagement_score::NUMERIC) AS avgengagementscore
FROM campaigndata
GROUP BY target_audience
ORDER BY avgengagementscore DESC;
```

Purpose: Helps marketers refine audience targeting based on engagement.

e) **Calculate Overall Click-Thriugh_Rate (CTR)**

Objective: Measure the effectiveness of campaigns in driving user interactions.

```
SELECT ((SUM(clicks)::NUMERIC)/SUM(impressions) * 100.0) AS overallctr
FROM campaigndata;
```

Purpose: Helps measure user engagement with campaigns.

f) **Finding the Most Cost-Effective Campaign**

Objective: Identify the campaign with the lowest cost per conversion.

```
SELECT campaign_id, company,
            (acquisition_cost::NUMERIC/NULLIF((conversion_rate * clicks), 0)) AS costperconversion
FROM campaigndata
ORDER BY costperconversion ASC
LIMIT 1;
```

Purpose: Helps businesses optimize ad spending

g) **Finding Campaigns with CTR Above 5%**

Objective: Identify high-performing campaigns that achieved a CTR > 5%

```
WITH CampaignCTR AS (
  SELECT campaign_id,
      company,
      ROUND(((NULLIF(clicks::NUMERIC, 0) /NULLIF(impressions,0)) * 100),2) AS ctr
  FROM campaigndata
)
SELECT campaign_id, company, ctr
FROM CampaignCTR
WHERE ctr > 5
ORDER BY ctr DESC;
```

Purpose: Helps focus on high engagement campaigns.

h) **Ranking Channels by Total Conversions.**

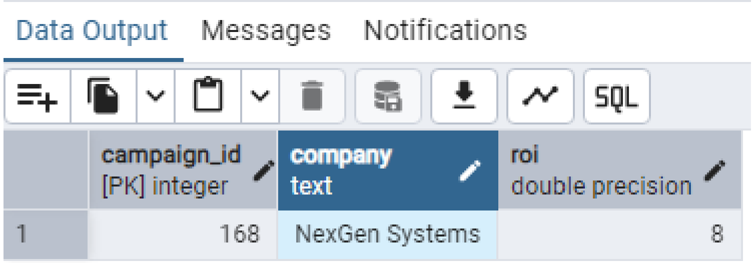
Objective: Determine which marketing channels generated the most conversions.

```
WITH CHannelConversions AS (
    SELECT channel_used,
                    SUM(conversion_rate * clicks) AS totalconversions
    FROM campaigndata
    GROUP BY channel_used
)
SELECT channel_used,
    totalconversions
FROM ChannelConversions
ORDER BY totalconversions DESC;
```

Purpose: Helps businesses prioritize the most effective channels for future campaigns.

RESULTS & INSIGHTS INTEPRETATION

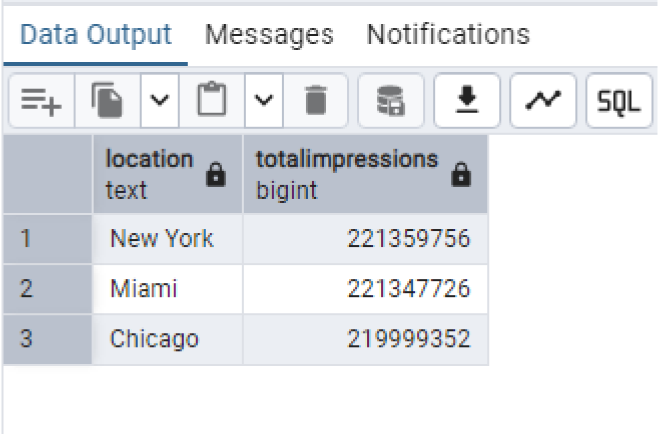
1. **Top ROI Performer:** NexGen Systems achieved the highest ROI of **8.0**.



	campaign_id [PK] integer	company text	roi double precision
1	168	NexGen Systems	8

2. **Highest Impressions by State:**

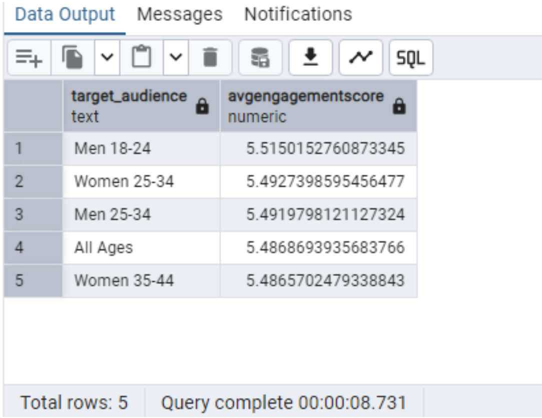
- **New York: 221,359,756** impressions
- **Miami: 221,347,726** impressions
- **Chicago: 219,999,352** impressions



	location text	totalimpressions bigint
1	New York	221359756
2	Miami	221347726
3	Chicago	219999352

3. **Engagement Score Trends:**

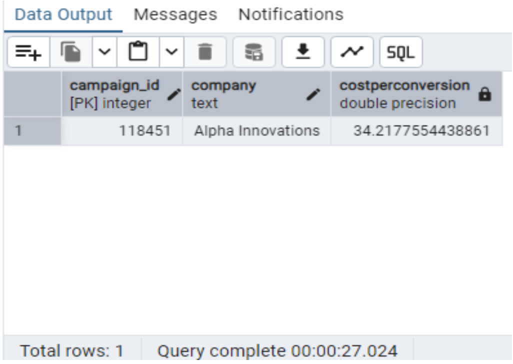
- Highest Average Engagement Score: Men (18-24)
- Lowest Average Engagement Score: Women (36-44).



	target_audience text	avgengagementscore numeric
1	Men 18-24	5.5150152760873345
2	Women 25-34	5.4927398595456477
3	Men 25-34	5.4919798121127324
4	All Ages	5.4868693935683766
5	Women 35-44	5.4865702479338843

Total rows: 5 Query complete 00:00:08.731

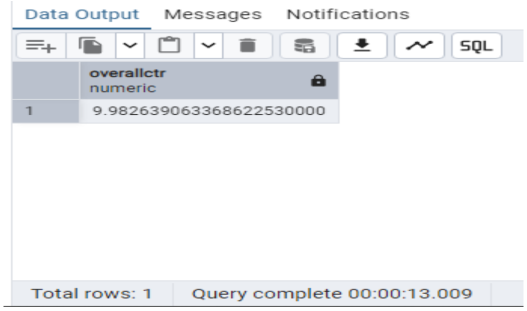
4. **Overall Click-Through Rate (CTR): Approximately 9.98%** across all campaigns.



	campaign_id [PK] integer	company text	costperconversion double precision
1	118451	Alpha Innovations	34.2177554438861

Total rows: 1 Query complete 00:00:27.024

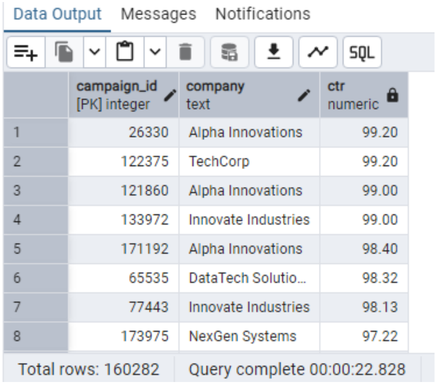
5. **Most Cost-Effective Campaign:** Alpha Solutions ran the most optimized campaign with a **cost-per-conversion of $34**.conversion.



	overallctr numeric
1	9.982639063368622530000

Total rows: 1 Query complete 00:00:13.009

6. **High-Performing Campaigns:** Over **160,000 campaigns** surpassed the **5% CTR threshold.**Email was the

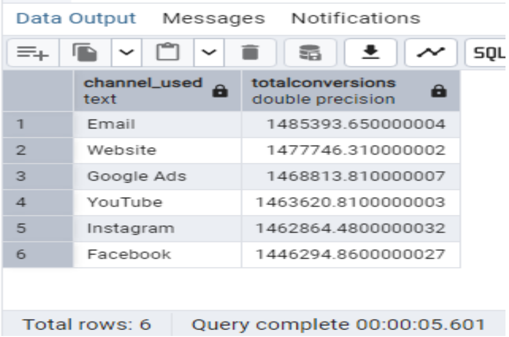


	campaign_id [PK] integer	company text	ctr numeric
1	26330	Alpha Innovations	99.20
2	122375	TechCorp	99.20
3	121860	Alpha Innovations	99.00
4	133972	Innovate Industries	99.00
5	171192	Alpha Innovations	98.40
6	65535	DataTech Solutio...	98.32
7	77443	Innovate Industries	98.13
8	173975	NexGen Systems	97.22

Total rows: 160282 Query complete 00:00:22.828

7. **Conversion by Channel:**

- **Highest Conversions: Email**
- **Lowest Conversions: Facebook**



	channel_used text	totalconversions double precision
1	Email	1485393.650000004
2	Website	1477746.310000002
3	Google Ads	1468813.810000007
4	YouTube	1463620.8100000003
5	Instagram	1462864.4800000032
6	Facebook	1446294.8600000027

Total rows: 6 Query complete 00:00:05.601

○ Click to view python script and SQL queries