ASSIGNMENT: AUTOMATED REPORTING SYSTEM DESIGN

Group-3:
Glenn Onyango, Jishnuraj, Kajal Sharma, Rajat Mohite, Suraj Shivaraj

OBJECTIVE:

The primary goal of this project is to develop an automated reporting solution that extracts, analyses, and reports the previous day's marketing spending and key performance metrics from Google BigQuery, and delivers the insights to a Microsoft Teams channel. This solution aims to streamline the reporting process, ensuring that accurate and timely data is available to the marketing team and stakeholders.

Benefits of Automating the Reporting Process:

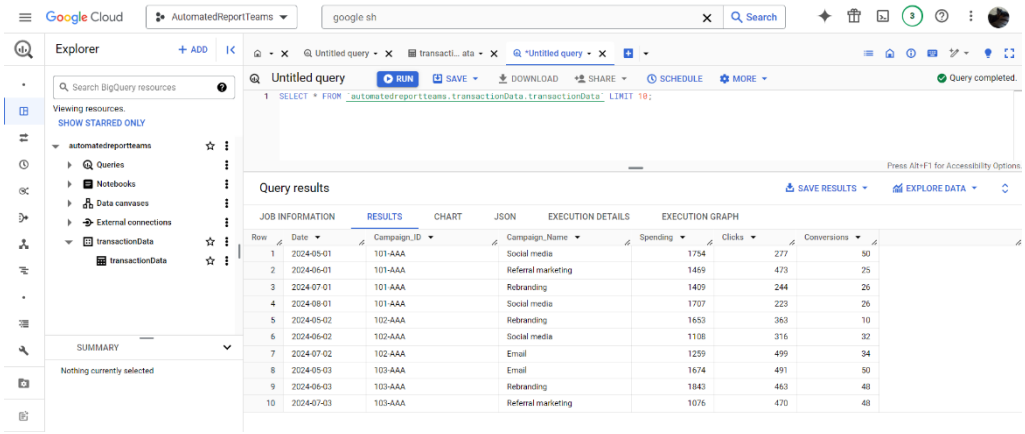
1. **Time Efficiency**: Reduces the manual effort required to gather, analyse, and report on marketing data.
2. **Accuracy and Consistency**: Minimizes the risk of human error in data extraction and report generation.
3. **Real-Time Insights**: Provides the marketing team with up-to-date information, enabling quicker responses to performance trends and potential issues.
4. **Cost Efficiency**: Reduces the need for manual labour and potential overtime costs associated with manual report generation.

STEPS TO ACHIEVE THE PROJECT:

DATA EXTRACTION:

1. **Create a New Project in BigQuery Studio**:
    - Begin by logging into the Google Cloud Console and navigate to BigQuery.
    - Create a new project or select an existing project to organize and manage your BigQuery resources effectively.
2. **Configure IAM and Admin Settings**:
    - Access the IAM & Admin section from the Google Cloud Console.
    - Create a new service account to manage the BigQuery access.
    - Assign the appropriate roles to the service account, such as "Owner" or any other role that suits your organization's requirements.
3. **Generate a Service Account Key**:
    - Once the service account is created, generate a key for it.
    - This key will be a JSON file containing all the necessary credentials for accessing BigQuery programmatically from Google Sheets and Python scripts.
    - Download and securely store this JSON key file as it will be used in subsequent steps to authenticate and connect to BigQuery.

4. **Load the Dataset into BigQuery**:
    - If the dataset is available on your local machine, upload it to BigQuery.
    - Navigate to the BigQuery console and select the project you created and select the dataset.
    - Then create a table name and choose the appropriate source format (e.g., CSV if your data is in CSV format).
    - Follow the prompts to upload and load the dataset into the BigQuery table.
5. **Verify the Data with a Simple SQL Query**:
    - To ensure that the data has been correctly uploaded and is accessible, write a simple SQL query.
    - Execute the query to check the contents of the table and verify that the correct data has been loaded.



DATA CLEANING:

Cleaning the data in BigQuery Studio involves several crucial steps to ensure data quality, including handling missing values and ensuring consistency. Here are few examples which can be worked on the data.

Identify Missing Values: Use SQL queries to detect missing values in your dataset. For example:

```
SELECT *
FROM `automatedreportteams.transactionData.transactionData`
WHERE Spending IS NULL;
```

Remove or Impute Missing Values:

- **Removal**: If the missing values are few and not critical, you can choose to remove these row
- **Imputation**: For critical columns, impute missing values using appropriate methods such as mean, median, or a specific value.

Ensuring Data Consistency:

- **Standardize Formats**: Ensure that data formats are consistent across columns.

```
4
5  CREATE OR REPLACE TABLE cleaned_table AS
6  SELECT Spending,Clicks,Conversions
7    PARSE_DATE('%Y-%m-%d', Date) AS standardized_date,
8
9  FROM `automatedreportteams.transactionData.transactionData`
```

Handling Outliers:

- Detect outliers using statistical methods such as Z-score or IQR:

```
4
5  SELECT*,
6    (Spending - AVG(Spending) OVER ()) / STDDEV(Spending) OVER () AS z_score
7  FROM `automatedreportteams.transactionData.transactionData`
8  HAVING ABS(z_score) < 3
```

Alternative Data Cleaning Approach: While data cleaning can be efficiently performed within BigQuery, it is also possible to handle this process in Python before updating the dataset in BigQuery. This approach allows for more flexible and extensive data manipulation using Python's robust data processing libraries. Once the data is cleaned in Python, the updated dataset can be uploaded to BigQuery for further analysis.

METRIC CALCULATION:

1. **Total Spending**: The total amount spent on marketing campaigns for the previous day. (Sum of Spending)
2. **Total Clicks**: The total number of clicks generated by the marketing campaigns. (Sum of Clicks)
3. **Total Conversions**: The total number of conversions generated by the marketing campaigns. (Sum of Conversions)
4. **Cost Per Click (CPC)**: The average cost for each click generated by the marketing campaigns. (Total Spending / Total Clicks)
5. **Cost Per Conversion (CPA)**: The average cost for each conversion generated by the marketing campaigns. (Total Spending / Total Conversions)

REPORT CREATION STRUCTURE:

1. **Report Overview**

- **Title**: Marketing Campaign Performance Report
- **Date**: The reporting period (e.g., Previous Day: YYYY-MM-DD)
- **Objective**: A brief statement on the purpose of the report, which is to analyze and assess the effectiveness of marketing campaigns for the specified period.

2. **Key Metrics**
   - Total Marketing Spend, Clicks, Conversions, Cost Per Click (CPC), Cost Per Conversion
3. **Visualizations**
   - **Spend Breakdown**: Pie chart or bar graph showing the distribution of marketing spend across different channels or campaigns.
   - **Conversion Funnel**: Funnel chart depicting the conversion process from impressions to final conversions, highlighting drop-off points.
4. **Detailed Analysis**
   - **Campaign Performance**: A breakdown of performance by individual campaigns, including metrics such as spend, clicks and conversions.
   - **Channel Analysis**: Comparison of different marketing channels (e.g., social media, email, search engine) based on performance metrics.
   - **Audience Segmentation**: Analysis of how different audience segments responded to the campaigns.

6. Key Insights

- **Trends**: Emerging trends or patterns observed in the marketing performance metrics.
- **Recommendations**: Actionable insights and suggestions for optimizing future marketing campaigns based on the data analysis.

7. Conclusion

- **Summary of Findings**: Recap of the main insights and implications for the marketing strategy.
- **Next Steps**: Suggested actions based on the report's findings, such as reallocating budget, adjusting strategies, or further analysis.

AUTOMATING REPORT GENERATION AND DELIVERY TO MICROSOFT TEAMS:

To automate the report generation and delivery process, follow these detailed steps:

1. **SQL Query Execution**: Write and execute the SQL code to extract the relevant data from the BigQuery tables. Ensure that the SQL queries are accurate and tailored to retrieve the necessary performance metrics.

```
previousDay = """
SELECT  date,
        SUM(Spending) as TotalSpending,
        SUM(Clicks) as TotalClicks,
        SUM(Conversions) as TotalConversions,
        ROUND(SUM(Spending)/SUM(Clicks),2) as CostPerClick,
        ROUND(SUM(Spending)/SUM(Conversions),2) as CostPerConversion
FROM `automatedreportteams.transactionData.transactionData`
WHERE Date = DATE_SUB(CURRENT_DATE(), INTERVAL 1 DAY)
GROUP BY Date
;
"""
```

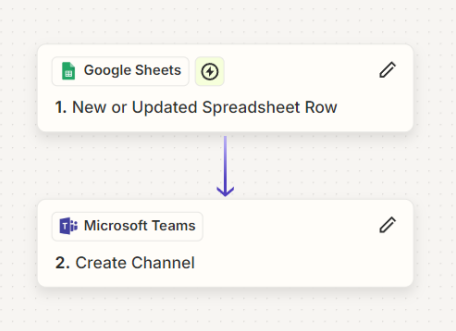
2. **API Configuration**: Enable the Google Drive API and Google Sheets API in the Google Cloud Console. This configuration is essential for interacting with Google Sheets from BigQuery.
3. **Data Extraction and Transformation**: Load the results of the executed SQL query into a pandas DataFrame. This DataFrame will serve as the source of the data to be appended to Google Sheets. Convert the DataFrame to a list format and ensure that date fields are formatted as strings to maintain compatibility with Google Sheets.

```
# Define the scope
scope = ["https://spreadsheets.google.com/feeds", "https://www.googleapis.com/auth/drive"]

# Add credentials to the account
creds = ServiceAccountCredentials.from_json_keyfile_name('automatedreportteams-426703e845d0.json', scope)

# Authorize the clientsheet
client = gspread.authorize(creds)

# Open the spreadsheet
spreadsheet = client.open('AutomatedReportTeams')

#Select first sheet
sheet = spreadsheet.sheet1
```

4. **Google Sheets Integration**: Use the JSON key file downloaded from BigQuery to authenticate and access Google Sheets via the Google Sheets API. Append the transformed data from the DataFrame to a Google Sheet as a new row. This step ensures that the report data is updated and available for subsequent processing.

	A	B	C	D	E	F
1	date	TotalSpending	TotalClicks	talConversions	CostPerClick	PerConversion
2	2024-07-31	1798	439	46	4.1	39.09
3						

5. **Zapier Integration**: Configure Zapier to automate the report delivery process. Create a Zap that includes a Trigger and an Action:
    - **Trigger**: Set Google Sheets as the trigger app. Configure the trigger to activate when a new row is appended to the Google Sheet.
    - **Action**: Set Microsoft Teams as the action app. Configure the action to post the updated report data to a designated channel in Microsoft Teams.

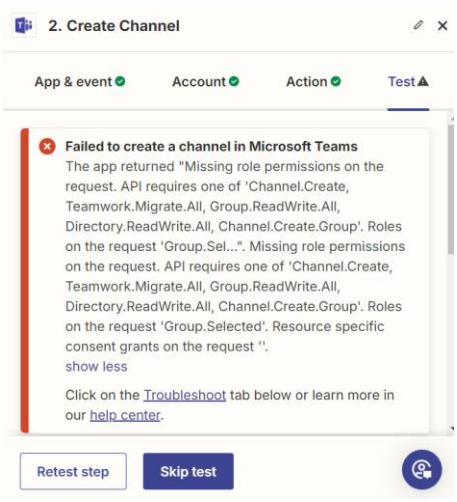


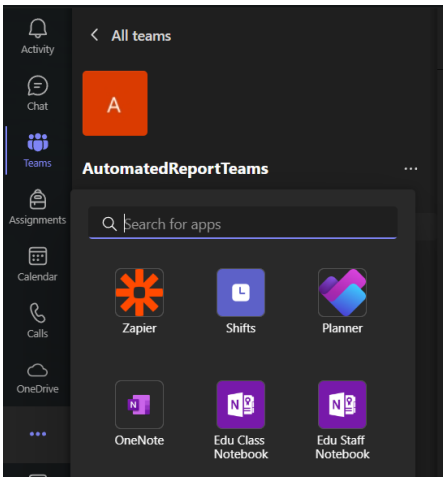
6. **Google Sheets and Zapier Authorization**: Grant Zapier access to Google Sheets and BigQuery by using the JSON key file. This authorization ensures that Zapier can interact with the data sources and execute the desired actions.
7. **Zapier Configuration**: In Zapier, select the appropriate Google Sheet and specify the columns to be monitored for changes.

    Configure the Microsoft Teams action by selecting the target team and channel where the report will be delivered.

8. **Azure Permissions and Teams Integration**: Ensure that the necessary permissions are granted in Azure Active Directory for the Microsoft Teams integration. This may involve assigning appropriate roles and consent for the required permissions.

   Install the Zapier application in Microsoft Teams if not already present, to facilitate the integration and ensure seamless communication between Zapier and Teams.





IMPLEMENTATION PLAN:

Step	Description	Hours	Resources
Data Extraction	Extract data from BigQuery	2 Hrs	BigQuery, Python
Data Cleaning	Clean and preprocess data	6 Hrs	BigQuery, Python
Metric Calculation	Calculate performance metrics	2 Hrs	Python
Report Creation	Generate report with metrics and visualizations	8 Hrs	Matplotlib, Report Generation Tools
Automation and Delivery	Automate and send report to Teams	8 Hrs	Python, Microsoft Teams API

CONCLUSION:

The automated reporting solution integrates BigQuery, Google Sheets, and Microsoft Teams through Zapier to streamline the extraction, transformation, and delivery of marketing performance data. By automating this process, the solution ensures timely and accurate updates of key metrics, significantly reducing manual effort and errors. The result is a more efficient workflow that provides stakeholders with real-time insights, enhances data visibility, and supports informed decision-making, ultimately leading to optimized marketing strategies and improved return on investment.