**DATA ANALYSIS PROJECT**

**OBJECTIVE:**

The Head of Marketing wants to find out who the top YouTubers are in 2024 to decide on which YouTubers would be best to run marketing campaigns throughout the rest of the year.

Objective is to create a dashboard that provides insights into the top UK YouTubers in 2024 that includes their:

* subscriber count
* total views
* total videos
* engagement metrics

This dashboard should allow to identify the top performing channels based on metrics like subscriber base and average views which in return will help the marketing team make informed decisions about which YouTubers to collaborate with for their marketing campaigns.

**TOOLS USED:**

Microsoft Excel: to explore and analyze the data

SQL Server: for cleaning, testing and analyzing the data

Power BI: Visualizing the data via interactive dashboards

**GENERAL APPROACH IN CREATING THE SOLUTION:**

1. Get the data (from kaggle)
2. Explore the data in Excel
3. Load the data into SQL Server
4. Clean the data with SQL
5. Test the data with SQL
6. Visualize the data in Power BI
7. Generate the findings based on the insights
8. Write the final recommendation and documentation required

**Creating the Database in SQL and organizing the data**

We create the youtuber\_db database and switched to it to use it using the USE command

Then we import the excel data file in the database

Then we:

1. Remove unnecessary columns by only selecting the ones we need

2. Extract the YouTube Channel names from the first columns

3. Rename the column names

SQL code is given below

/\*

a Data cleaning steps

1. Remove unnecessary columns by only selecting the ones we need

2. Extract the Youtube Channel names from the first columns

3. Rename the column names

\*/

/\* step 1 \*/

/\*

SELECT

NOMBRE,

total\_subscribers,

total\_videos,

total\_views

FROM

top\_uk\_youtubers \_2024

\*/

-- this will display only the mentioned columns

/\* step 2 \*/

-- Charindex function

-- contains 2 arg, first the targetted char, 2nd the whole string

-- SELECT CHARINDEX('nice','i went to a nice beach') as pos\_of\_n\_in\_nice

-- this will show 13 as n is in the 13th position

-- we need to find the @ in all the names then we can extract whats on the left side to get the names

-- SELECT CHARINDEX('@', NOMBRE), NOMBRE FROM top\_uk\_youtubers\_data\_2024

-- this will show the position of the @ from the names in NOMBRE column

-- Substring

-- this takes 3 arg, first the column from where data will be taken, 2nd the starting point, 3rd the ending point

-- SELECT SUBSTRING(NOMBRE, 1, CHARINDEX('@', NOMBRE) -1) FROM top\_uk\_youtubers\_data\_2024

-- here the -1 will elimnate the @ symbol coming, bt this is not enuf we need to cast this into string type in sql server and for that

Then we create a view that will only show the data that power BI will use

CREATE VIEW view\_uk\_youtubers\_2024 AS

-- this creates a SQL view that will only view the data that power BI will use

-- this will be availabe under the views section under the database folders

SELECT

CAST(SUBSTRING(NOMBRE, 1, CHARINDEX('@', NOMBRE) -1) as varchar(100)) as channel\_name,

total\_subscribers,

total\_videos,

total\_views

FROM top\_uk\_youtubers \_2024

-- the hundred implies that the string limit is 100 in the channel names, we cant go over that and we named the column and selected the other columns as well

Then we will run data quality checks:

1. The data needs to be 100 records of Youtube Channels (row count checks)

2. The data needs 4 fields (column count test)

3. The channel name column must be string format, and the other data need to be numerical data type (data type check)

4. Each record must be unique in the dataset (duplicate count check)

/\*

Data Quality Checks

1. The data needs to be 100 records of Youtube Channels (row count checks)

2. The data needs 4 fields (column count test)

3. The channel name column must be string format, and the other data need to be numerical data type (data type check)

4. Each record must be unique in the dataset (duplicate count check)

check 1: Row Count - 100 (passed)

check 2: Column count - 4 (passed)

DATA TYPES

check 3: Channel name = VARCHAR (passed)

total subscribers, views, videos = INTEGER

check 4: Duplicate Count = 0 (passed)

\*/

-- ROW COUNT CHECK

SELECT

COUNT(\*) as no\_of\_rows

FROM

view\_uk\_youtubers\_2024

-- this will give 100

-- COLUMN COUNT CHECK

SELECT

COUNT(\*) as column\_count

FROM

INFORMATION\_SCHEMA.COLUMNS

-- this basically knows all information about the database, columns rows etc etc, and the .column will check the column specific

WHERE

Table\_name = 'view\_uk\_youtubers\_2024'

-- this will give 4

-- DATA TYPE CHECK

SELECT

COLUMN\_NAME, -- this names we got when ww ran the prev function with only: \* in place of: COUNT(\*) as column\_count

DATA\_TYPE

FROM

INFORMATION\_SCHEMA.COLUMNS

WHERE

TABLE\_NAME = 'view\_uk\_youtubers\_2024'

-- this will view the data types

-- DUPLICATE CHECK

SELECT

channel\_name,

COUNT(\*) as duplicate\_count

FROM

view\_uk\_youtubers\_2024

GROUP BY

channel\_name

HAVING

COUNT(\*) > 1 -- having pairing more than 1

-- this returns nothing which means no duplicate

**Creating the Power BI Dashboard**

Then we open Power BI and start by creating a blank report

After that we go to “Get Data” to get data from “Sequel Server” (we need to input the server’s name that we will get from ssms)

We then select the view table that we created and LOAD that into power bi

Now we officially have the sql view in our power bi

Creating the DAX code:

Start by creating a new measure

Measure is something that summarizes columns and contents

DAX MEASURE

We create all the variable that we need in the new measure command section

Command :

Total Subscribers (M) =

VAR million = 1000000

VAR sumofSubscribers = SUM(view\_uk\_youtubers\_2024[total\_subscribers])

VAR totalSubscribers = DIVIDE(sumofSubscribers, million)

RETURN totalSubscribers

TOTAL VIEWS (B)

Command :

Total Views (B) =

VAR billion = 1000000000

VAR sumofTotalViews = SUM(view\_uk\_youtubers\_2024[total\_views])

VAR totalViews = DIVIDE(sumofTotalViews, billion)

RETURN totalViews

TOTAL VIDEOS

Command :

Total Videos =

VAR totalVideos = SUM(view\_uk\_youtubers\_2024[total\_videos])

RETURN totalVideos

AVERAGE VIEWS PER VIDEOS (M)  
Command :

Average Views per Videos =

VAR sumofTotalViews = SUM(view\_uk\_youtubers\_2024[total\_views])

VAR sumofTotalVideos = SUM(view\_uk\_youtubers\_2024[total\_videos])

VAR avgOfViewsPerVideos = DIVIDE(sumofTotalViews, sumofTotalVideos, BLANK())

VAR finalAvgPerVideos = DIVIDE(avgOfViewsPerVideos, 1000000, BLANK())

RETURN finalAvgPerVideos

SUBSCRIBER ENGAGEMENT RATE  
Command :

Subscriber Engagement Rate =

VAR sumOfTotalSubscribers = SUM(view\_uk\_youtubers\_2024[total\_subscribers])

VAR sumOfTotalVideos = SUM(view\_uk\_youtubers\_2024[total\_videos])

VAR subscriberEngRate = DIVIDE(sumOfTotalSubscribers, sumOfTotalVideos)

RETURN subscriberEngRate

VIEW PER SUBSCRIBER

Command :

Views per Subscribers =

VAR sumOfTotalViews = SUM(view\_uk\_youtubers\_2024[total\_views])

VAR sumOfTotalSubscribers = SUM(view\_uk\_youtubers\_2024[total\_subscribers])

VAR viewPerSubscriber = DIVIDE(sumOfTotalViews, sumOfTotalSubscribers)

RETURN viewPerSubscriber

**Formatting the DASHBOARD**

Now we are officially ready to create the dashboard

We will start be adding the title by adding a text box and center it

Then we add the table, under the visual tab on the rhs theres a table option, use this to create the table and then add the measures that we need, after that go into the format tab on the rhs and do all the necessary stuffs to make the table look presentable

After this is done, we need to add the tree map

Add the tree map from the rhs and add the specifics that we want to add like the channel name and the total views, bt we need only the top 20, 10 or whatever the number maybe so for that, so again on the rhs under the filter , I the channel name select the filter and use the top n filter and add the 20 so filter out only the top 20 channels and in the by value area drag and drop the total views from the list of measures and then finally apply filter  
and under the tooltips bar on the bottom rhs add the fields from the list of measure that we want to view when we hover over the channel name in the tree map  
  
Then we add the score card from the visualisation on the rhs. Add the average views per videos on the card from the list of measures, like this create 3 cards for subscribers’ engagement rate and views per subscriber

After this we add the horizontal Bar Chart

Similarly add this from the visualization tab from rhs, and add channel name and total subscriber measures in the bar chart and again filtering it to top 10 and in the by value option add the total subscribers measure

Final Formatting will be done next, like formatting some column name only for the visual, change the range in the bar chart, for the tree map use the fx option in the colour section to add gradient and minimalistic touch to the tree chart

**Analysis**

Then we prepare a Excel Sheet to do the analysis

First, we do the analysis manually on Excel and then on SQL separately

**Excel**

Tips:

While performing cell calculation if we select a cell and press f4 then during all the calculation the cell with f4 will be locked for all and will not change

In the difference column we apply conditional formatting to the values, i.e. if there is a difference between the SQL and the EXCEL value, the cell will get coloured

This is done from the conditional formatting option in the home tab

**SQL**

/\*

Steps to perfrom analysis

1. Define the variables

2. Create a CTE that rounds the avg views per video

3. Select column appropriate for our analysis

4. Filter the result by YT channels with the highest subscriber basis

5. Order by net\_profit (high to low)

\*/

-- 1.

declare @conversionRate float = 0.02; -- conversion rate at 2%

declare @productCost float = 5.0; -- product cost is $5

declare @campaignCost float = 50000.0; --campaign cost at $50000

-- 2.

with ChannelData as (

select

channel\_name,

total\_views,

total\_videos,

Round ((cast (total\_views as float) / total\_videos), -4) as rounded\_avg\_views\_per\_video,

-- when we cast we needed to declare the value as float for calculation else it get stored as a string

-- and we rouned it to the nearest ten thousand

(total\_views / total\_videos) as original\_avg\_views\_per\_video

-- these datas will be not rounded

from

youtube\_db.dbo.view\_uk\_youtubers\_2024

)

-- select \* from ChannelData

-- this will views all filtered data

-- 3.

select

channel\_name,

rounded\_avg\_views\_per\_video,

(rounded\_avg\_views\_per\_video \* @conversionRate) as potential\_units\_sold\_per\_video,

(rounded\_avg\_views\_per\_video \* @conversionRate \* @productCost) as Potential\_revenue\_per\_video,

(rounded\_avg\_views\_per\_video \* @conversionRate \* @productCost) - @campaignCost as net\_profit

from

ChannelData

-- 4.

-- filter by youtube channel names

where

channel\_name in ('NoCopyrightSounds' , 'DanTDM' , 'Dan Rhodes')

-- 5.

order by

net\_profit desc

**Conclusion**

Now we have all the data that we need from SQL after the analysis

We now copy the data into the excel analysis sheet to see if there is any difference between the manual and the SQL data analysis, if there is no difference we can say our SQL analysis is correct.

**Recommendation**

After that we provide our own recommendation on which Youtuber the Marketing Team should invest on to get the maximum profit. Recommendation is given in the Excel Sheet