

## YOUTUBE ANALYSIS

### DATA CLEANING AND CREATING A VIEW

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor displays a SQL query that selects the top 1000 rows from the 'youtube\_uk\_data' table, ordered by the number of subscribers. The query is as follows:

```
SELECT TOP (1000) [Column_1]
FROM [youtube_data].[dbo].[youtube_uk_data]
```

The results pane shows the following data:

Column_1	NOMBRE	SEGUIDORES	TP	PAIS	TEMA_DE_INFLUENCIA	ALCANCE_POTENCIAL	GUARDAR	INVITAR_A_LA_CAMPANA	channel_name
1	NoCopyrightSounds @UC_aEa8K-EQJ3D6gOa7HcyNg	32.9M	-	Reino Unido	Música	9.9M	NULL	Ver Perfil	NoCopyrightSounds
2	DanTDM @UCSS0z6ChmwoF7v5adqqx9w	26.3M	-	Reino Unido	NULL	7.9M	NULL	Ver Perfil	DanTDM
3	KSI @UCVfOyBpEzLjyGQ5gxG	24M	-	Reino Unido	NULL	7.2M	NULL	Ver Perfil	KSI
4	Mister Max @UC_8PA0X0mrl_gpa77S1Aagg	23M	0.10%	Reino Unido	NULL	6.9M	NULL	Ver Perfil	Mister Max
5	Jelly @UC002mkupLWecyDafoclH0A	22.9M	-	Reino Unido	NULL	6.9M	NULL	Ver Perfil	Jelly
6	Miss Katy @UCaahVvAlzdflyeT_Gg	22.6M	-	Reino Unido	NULL	6.8M	NULL	Ver Perfil	Miss Katy
7	Dua Lipa @UC-JK2Rv8c13OCkXGdUQ	21.1M	-	Reino Unido	NULL	6.3M	NULL	Ver Perfil	Dua Lipa
8	Dan Rhodes @UC8D1L2vEAg_V0JSxMBDgA	17.7M	0.30%	Reino Unido	NULL	5.3M	NULL	Ver Perfil	Dan Rhodes
9	Al-A @UCYVinkwSX7aARULg1p7nLw	17.7M	-	Reino Unido	NULL	5.3M	NULL	Ver Perfil	Al-A
10	Gaby and Alex @UCHu6RtoQ0TV-HBQVYGSVHhQ	17M	-	Reino Unido	NULL	5.1M	NULL	Ver Perfil	Gaby and Alex
11	Sidemen @UCDogdK7N7HqZQ95aEwkdMw	17M	2.80%	Reino Unido	NULL	5.1M	NULL	Ver Perfil	Sidemen
12	Queen Official @UCMhD4pUgGgPzUmmYRQ	16.3M	-	Reino Unido	NULL	4.9M	NULL	Ver Perfil	Queen Official
13	Little Mix @UCVqWqC4Lvp7z8hzARa	16.1M	-	Reino Unido	NULL	4.8M	NULL	Ver Perfil	Little Mix
14	JJ Olatunji @UCGmmsW623G1r-Chmo5RB4Yw	15.9M	-	Reino Unido	NULL	4.8M	NULL	Ver Perfil	JJ Olatunji
15	F2Freestylers - Ultimate Soccer Skills Channel @UCK...	14M	0.20%	Reino Unido	Celebridad	4.2M	NULL	Ver Perfil	F2Freestylers - UL...
16	BDP - Music @UK15-BD-BN-MED-M79v7KsDA	13.9M	-	Reino Unido	Música	4.1M	NULL	Ver Perfil	BDP - Music

The status bar at the bottom indicates that the query was executed successfully, returning 100 rows.

The screenshot shows the Microsoft SQL Server Management Studio interface with two queries displayed in the query editor.

Query 1 (SQLQuery5.sql - D...4BOD\Sushma (55)):

```
select
NOMBRE,
total_subscribers,
total_views,
total_videos
from youtube_uk_data
```

Query 2 (SQLQuery1.sql - D...4BOD\Sushma (52))\*:

```
select
cast(substring(NOMBRE, 1, charindex('@', NOMBRE)-1) as varchar(100)) as channel_name
from youtube_uk_data
```

## Checking the data

```
SQLQuery5.sql - D...4BOD\Sushma (55))    SQLQuery1.sql - D...4BOD\Sushma (52))*  
--1. row count check  
SELECT TOP (1000) [channel_name]  
    , [total_subscribers]  
    , [total_videos]  
    , [total_views]  
FROM [youtube_datab].[dbo].[view_uk_youtube]  
  
--2. column count check  
SELECT count(*) AS no_of_rows  
FROM view_uk_youtube  
  
SELECT count(*) AS column_count  
FROM  
    INFORMATION_SCHEMA.COLUMNS  
WHERE  
    TABLE_NAME = 'view_uk_youtube'
```

--3. data type check

```
SELECT  
    COLUMN_NAME,  
    DATA_TYPE  
FROM  
    INFORMATION_SCHEMA.COLUMNS  
WHERE  
    TABLE_NAME = 'view_uk_youtube'
```

--4. Duplicates check

```
SELECT channel_name,  
    COUNT(*) AS duplicate_count  
FROM view_uk_youtube  
GROUP BY channel_name  
HAVING COUNT(*) > 1
```

## ROI ANALYSIS FOR THE CAMPAIGN :

Conversion rate is 2%

Production cost is \$5 per unit

Cost for the campaign is \$13000

## MOST VIEWS

SQLQuery7.sql - D...4BOD\Sushma (64))*					
SQLQuery2.sql - D...4BOD\Sushma (82))					
SQLQuery4.sql - D...4BOD\Sushma (69))*					
<pre>---- MOST VIEWS  DECLARE @conversionrate FLOAT= 0.02; --conversion rate is 2% DECLARE @productcost MONEY = 5.0; --Production cost is \$5 per unit DECLARE @campaigncost MONEY = 130000.0; --Cost of the campaign is \$50000  WITH ChannelData AS (     SELECT         channel_name,         total_views,         total_videos,         ROUND(CAST(total_views AS FLOAT) / total_videos, -4) AS avg_views_per_video     FROM view_uk_youtube ) SELECT     channel_name,     avg_views_per_video,     (avg_views_per_video * @conversionrate) AS potential_units_sold_per_video,     (avg_views_per_video * @conversionrate * @productcost) AS potential_revenue_per_video,     (avg_views_per_video * @conversionrate * @productcost) - @campaigncost AS net_profit FROM     ChannelData  WHERE     channel_name IN ('Mister Max', 'DanTDM', 'Dan Rhodes') ORDER BY     net_profit DESC;</pre>					
100 %					
Results Messages					
channel_name	avg_views_per_video	potential_units_sold_per_video	potential_revenue_per_video	net_profit	
1 Mister Max	14060000	281200	1406000	1276000	
2 Dan Rhodes	11150000	223000	1115000	985000	
3 DanTDM	5340000	106800	534000	404000	

## TOP SUBSCRIBERS

SQLQuery7.sql - D...4BOD\Sushma (64))*					
SQLQuery2.sql - D...4BOD\Sushma (82))					
SQLQuery4.sql - D...4BOD\Sushma (69))*					
<pre>--top subscribers  DECLARE @conversionRate FLOAT = 0.02; -- The conversion rate @ 2% DECLARE @productCost FLOAT = 5.0; -- The product cost @ \$5 DECLARE @campaignCost FLOAT = 13000.0; -- The campaign cost  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     FROM         view_uk_youtube ) 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  WHERE     channel_name in ('NoCopyrightSounds', 'DanTDM', 'Dan Rhodes') ORDER BY     net_profit DESC</pre>					
100 %					
Results Messages					
channel_name	rounded_avg_views_per_video	potential_units_sold_per_video	potential_revenue_per_video	net_profit	
1 Dan Rhodes	11150000	223000	1115000	1102000	
2 NoCopyrightSounds	6920000	138400	692000	679000	
3 DanTDM	5340000	106800	534000	521000	

## Most videos

```
SQLQuery7.sql - D...4BOD\Sushma (64))*  SQLQuery2.sql - D...4BOD\Sushma (82))  SQLQuery4.sql - D...4BOD\Sushma (69))*
```

```
--youtubers with most video

DECLARE @conversionRate FLOAT = 0.02;           -- The conversion rate @ 2%
DECLARE @productCost FLOAT = 5.0;               -- The product cost @ $5
DECLARE @campaignCostPerVideo FLOAT = 13000.0;   -- The campaign cost per video
DECLARE @numberOfVideos INT = 11;               -- The number of videos (11)

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
    FROM
        view_uk_youtube)
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) - (@campaignCostPerVideo * @numberOfVideos)) AS net_profit
FROM
    ChannelData
WHERE channel_name IN ('GRM Daily', 'Man City', 'YOGSCAST Lewis & Simon ')
ORDER BY
    net_profit DESC;
```

100 %

Results Messages

	channel_name	rounded_avg_views_per_video	potential_units_sold_per_video	potential_revenue_per_video	net_profit
1	YOGSCAST Lewis & Simon	710000	14200	71000	-72000
2	GRM Daily	510000	10200	51000	-92000
3	Man City	240000	4800	24000	-119000

## Youtubers with most views

```
SQLQuery7.sql - D...4BOD\Sushma (64))*  SQLQuery2.sql - D...4BOD\Sushma (82))  SQLQuery4.sql - D...4BOD\Sushma (69))*
```

```
--youtubers with most views

DECLARE @conversionRate FLOAT = 0.02;           -- The conversion rate @ 2%
DECLARE @productCost MONEY = 5.0;               -- The product cost @ $5
DECLARE @campaignCost MONEY = 13000.0;         -- The campaign cost @ $13000

WITH ChannelData AS (
    SELECT
        channel_name,
        total_views,
        total_videos,
        ROUND(CAST(total_views AS FLOAT) / total_videos, -4) AS avg_views_per_video
    FROM
        view_uk_youtube)
SELECT
    channel_name,
    avg_views_per_video,
    (avg_views_per_video * @conversionRate) AS potential_units_sold_per_video,
    (avg_views_per_video * @conversionRate * @productCost) AS potential_revenue_per_video,
    (avg_views_per_video * @conversionRate * @productCost) - @campaignCost AS net_profit
FROM
    ChannelData
WHERE
    channel_name IN ('Mister Max', 'DanTDM', 'Dan Rhodes')
ORDER BY
    net_profit DESC;
```

100 %

Results Messages

	channel_name	avg_views_per_video	potential_units_sold_per_video	potential_revenue_per_video	net_profit
1	Mister Max	14060000	281200	1406000	1393000
2	Dan Rhodes	11150000	223000	1115000	1102000
3	DanTDM	5340000	106800	534000	521000

## POWERBI MEASURES

1.

---

```
1 avg views per video =  
2 VAR sumoftotalviews = SUM(view_uk_youtube[total_views])  
3 VAR sumoftotalvideos = SUM(view_uk_youtube[total_videos])  
4 VAR avgviewspervideo = DIVIDE(sumoftotalviews, sumoftotalvideos, BLANK())  
5 VAR finalavgviewspervideo = divide(avgviewspervideo, 1000000, BLANK())  
6  
7 Return finalavgviewspervideo
```

2.

```
1 Subscriber Engagement rate =  
2 VAR sumoftotalsubscriber = SUM(view_uk_youtube[total_subscribers])  
3 VAR sumoftotalvideos = SUM(view_uk_youtube[total_videos])  
4 VAR subscriberengrate = DIVIDE(sumoftotalsubscriber, sumoftotalvideos, BLANK())  
5  
6 RETURN subscriberengrate
```

3.

```
1 Total Subscriber (M) =  
2 var million = 1000000  
3 var SumofSubscriber = sum(view_uk_youtube[total_subscribers])  
4 var totalsubscriber = divide(SumofSubscriber,million)  
5  
6 return totalsubscriber
```

4.

```
1 total videos =  
2 VAR totalVideos = SUM(view_uk_youtube[total_videos])  
3  
4 RETURN totalVideos
```

5.

---

```
1 Total Views(B) =  
2 var billion = 1000000000  
3 var sumoftotalviews = SUM(view_uk_youtube[total_views])  
4 var totalviews = divide(sumoftotalviews, billion)  
5  
6 return totalviews
```

6.

---

```
1 views per subscriber =  
2 VAR sumoftotalviews = SUM(view_uk_youtube[total_views])  
3 VAR sumoftotalsubscribers = SUM(view_uk_youtube[total_subscribers])  
4 VAR viewspersubscriber = DIVIDE(sumoftotalsubscribers, sumoftotalsubscribers, BLANK())  
5  
6 RETURN viewspersubscriber
```