BIG DATA ENGINEERING

AT 1

Data Lakehouse using Snowflake

Rajat Singh

14330387

1. Introduction

The project focusses on ingesting YouTube trending data from Microsoft azure blob storage to create a data lake house in Snowflake. Also, this data is further merged, cleaned and store to answer some critical business questions.

2. Datasets

In this assignment, two types of files were given to ingest into Snowflake and data analysis i.e., YouTube Trending Dataset and YouTube Categories Dataset. Both datasets had 11 separate files, one for each country (Brazil, Canada, Germany, France, Great Britain, India, Japan, Korea, Mexico, Russia, and United States).

2.1 YouTube Trending Dataset

The YouTube Trending Dataset has 11 csv files separate files, one file for each country. The dataset has daily records for Top trending videos from each countries from 2020-08-12 to 2022-01-28.

The list of data attributes and their datatypes is given in the table below.

DATA ATTRIBUTES	DATA TYPES
VIDEO_ID	varchar
TITLE	varchar
PUBLISHEDAT	timestamp
CHANNELID	varchar
CHANNELTITLE	varchar
CATEGORYID	int
TRENDING_DATE	date
VIEW_COUNT	double
LIKES	double
DISLIKES	double
COMMENT_COUNT	double
COMMENTS_DISABLED	boolean

Table 1. Data Dictionary of Given Youtube Trending Dataset

2.2 YouTube Categories Dataset

Similar to the trending Dataset, Categories dataset also has 11 files but these are unstructured json files, not csv files. These files have data in json format for 29 categories. The categories id could be different for different countries; hence each country has its own file.

The most important data attributes for categories datasets are: -

1. category_id : int

2. category_title : varchar

3. Architecture

Architectural work flow of the project uses Microsoft Azure Blob Storage to store csv and json files that are later ingested into Snowflake for Data processing before using it to solve the business problem.

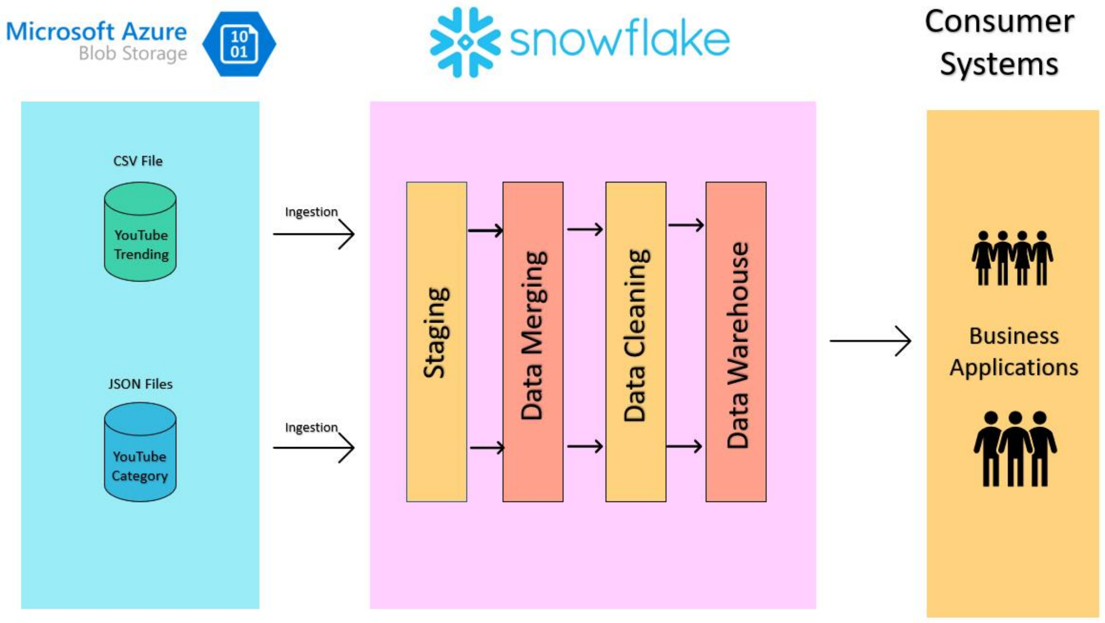


Figure 1. Architecture of the Project

3.1 Microsoft Azure Blob Storage

Azure Blob Storage is a scalable and secured data management solution by Microsoft that helps to create data lakes. In the storage account, a new container was created to store all the json and csv files.

Azure Resource	*Name*
Storage Account	rajatsinghuts
Storage - Container	bde-at-1

Like a directory in a file system, a container organises a group of blobs. An infinite number of containers may be included in a storage account, and an infinite number of blobs may be kept in a container.

3.2 Snowflake

Snowflake’s Data Cloud is powered by an advanced data platform provided as Software-as-a-Service (SaaS). Snowflake enables data storage, processing, and analytic solutions that are faster, easier to use, and far more flexible than traditional offerings (Snowflake, n.d).

Snowflake was used to create to perform CRUD operations on the data that was stored in Microsoft Blob. As Snowflake supports json file along with csv files, useful information was also extracted json files and converted it into tables.

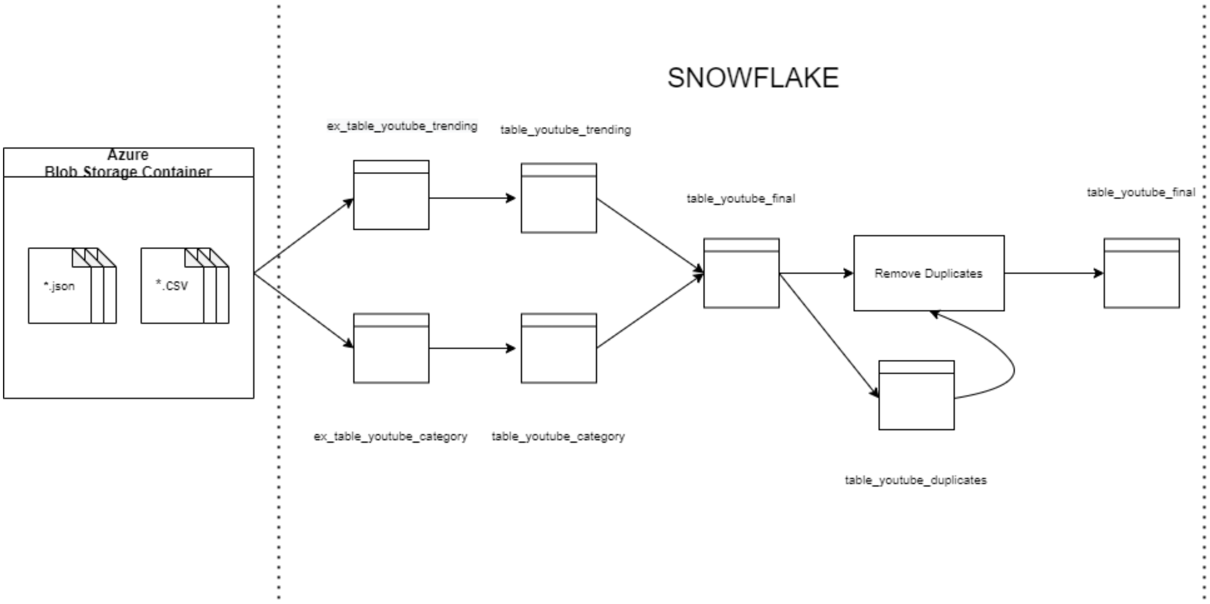


Figure 2. Data Flow Chart

The following steps were taken for data transformation in Snowflake.

1. Staging
   All of the files in the blob storage was staged in Snowflake. External tables were created with was further used to create new table in our database.
2. Data Merging
   The tables table_youtube_trending and table_youtube_category was merged with each other on country and category_id columns such that we got all the rows from table_youtube_trending.
3. Data Cleaning
   The duplicate rows entries were removed from the table ‘table_youtube_final’.
4. Data Warehousing
   The final table table_youtube_final is stored in our database for further use.

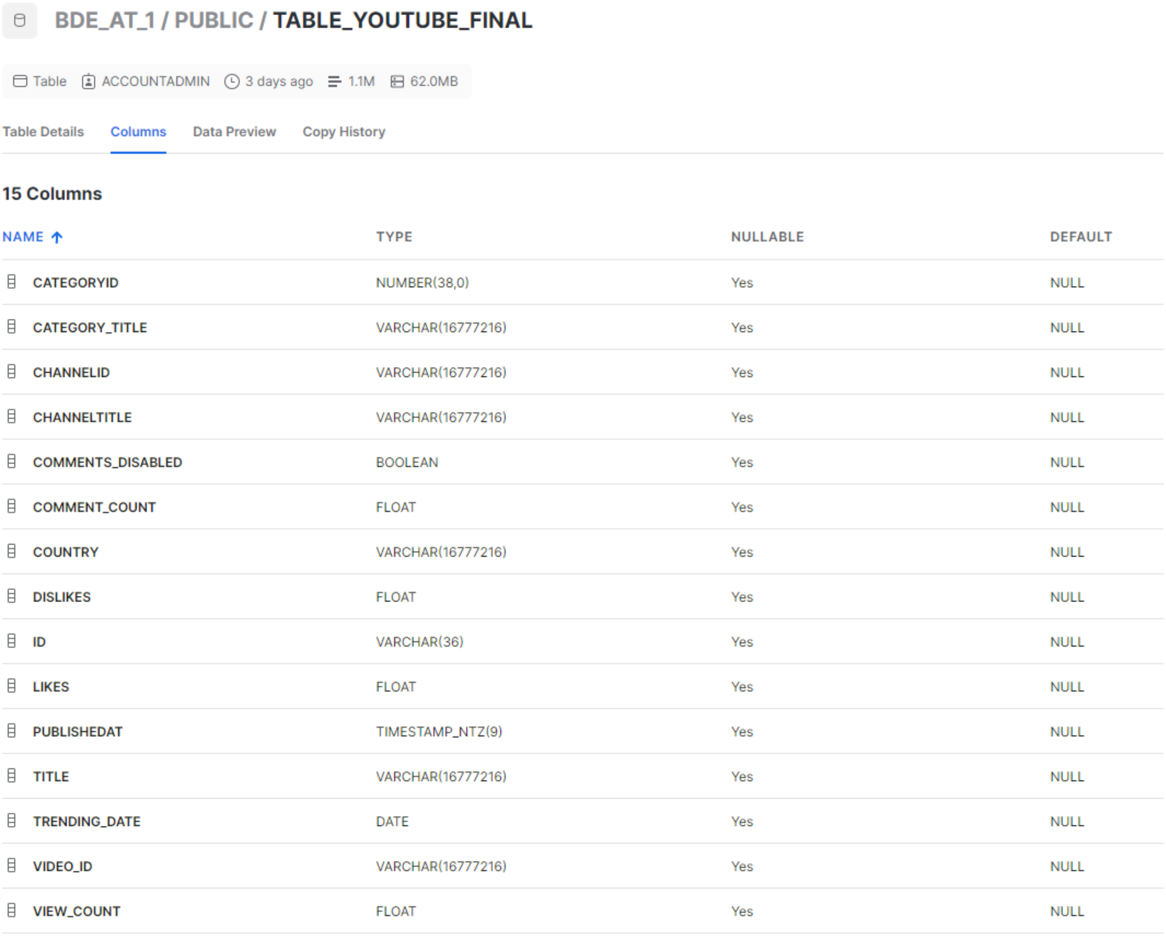
3.3 Consumer Systems

The data stored in the database after processing could be used by various business applications, and data visualisation tools to answer critical business questions.

4. Database Details and Data Dictionary

Type	Name
Database	BDE_AT_1
STORAGE INTEGRATION	AZURE_BDE_AT_1
STAGE	STAGE_BDE_AT_1
FILE FORMAT	FILE_FORMAT_CSV
EXTERNAL TABLE	EX_TABLE_YOUTUBE_TRENDING
EXTERNAL TABLE	EX_TABLE_YOUTUBE_CATEGORY
TABLE	TABLE_YOUTUBE_TRENDING
TABLE	TABLE_YOUTUBE_CATEGORY
TABLE	TABLE_YOUTUBE_FINAL
TABLE	TABLE_YOUTUBE_DUPLICATES

Table 2. Database Details



BDE_AT_1 / PUBLIC / TABLE_YOUTUBE_FINAL

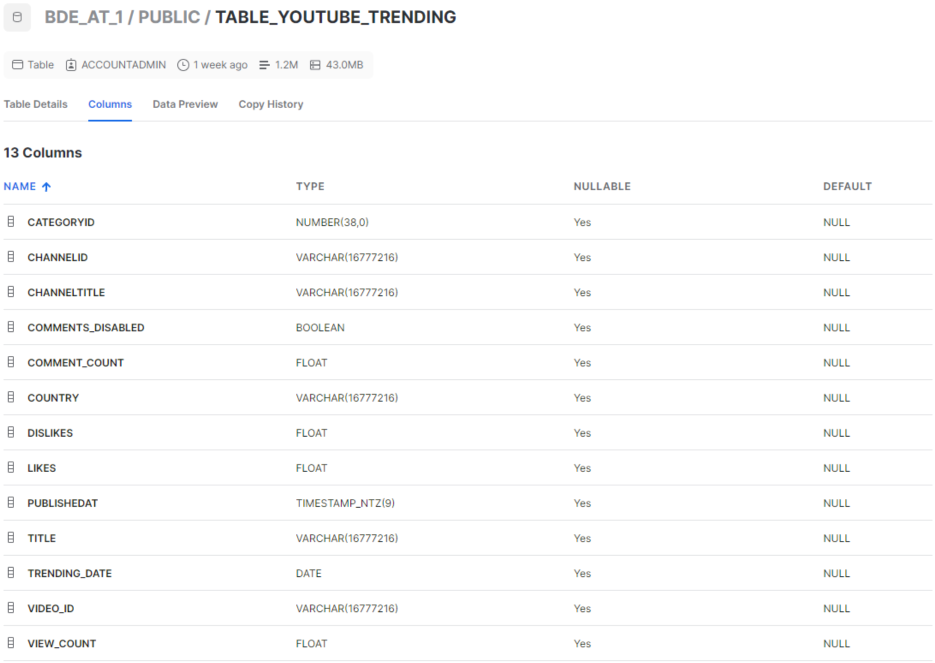
Table ACCOUNTADMIN 3 days ago 1.1M 62.0MB

Table Details Columns Data Preview Copy History

15 Columns

NAME ↑	TYPE	NULLABLE	DEFAULT
CATEGORYID	NUMBER(38,0)	Yes	NULL
CATEGORY_TITLE	VARCHAR(16777216)	Yes	NULL
CHANNELID	VARCHAR(16777216)	Yes	NULL
CHANNELTITLE	VARCHAR(16777216)	Yes	NULL
COMMENTS_DISABLED	BOOLEAN	Yes	NULL
COMMENT_COUNT	FLOAT	Yes	NULL
COUNTRY	VARCHAR(16777216)	Yes	NULL
DISLIKES	FLOAT	Yes	NULL
ID	VARCHAR(36)	Yes	NULL
LIKES	FLOAT	Yes	NULL
PUBLISHEDAT	TIMESTAMP_NTZ(9)	Yes	NULL
TITLE	VARCHAR(16777216)	Yes	NULL
TRENDING_DATE	DATE	Yes	NULL
VIDEO_ID	VARCHAR(16777216)	Yes	NULL
VIEW_COUNT	FLOAT	Yes	NULL

Figure 3. Data Dictionary TABLE_YOUTUBE_FINAL



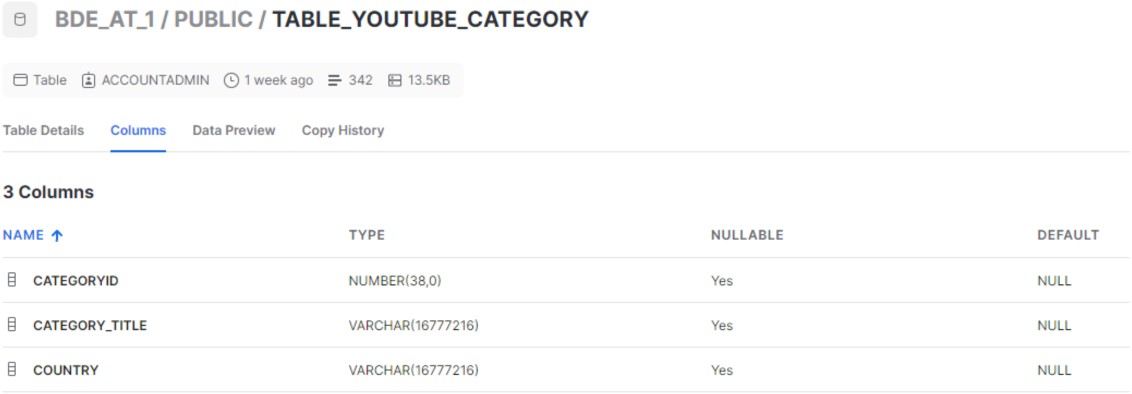
BDE_AT_1 / PUBLIC / **TABLE_YOUTUBE_TRENDING**

Table | ACCOUNTADMIN | 1 week ago | 1.2M | 43.0MB

13 Columns

NAME ↑	TYPE	NULLABLE	DEFAULT
CATEGORYID	NUMBER(38,0)	Yes	NULL
CHANNELID	VARCHAR(16777216)	Yes	NULL
CHANNELTITLE	VARCHAR(16777216)	Yes	NULL
COMMENTS_DISABLED	BOOLEAN	Yes	NULL
COMMENT_COUNT	FLOAT	Yes	NULL
COUNTRY	VARCHAR(16777216)	Yes	NULL
DISLIKES	FLOAT	Yes	NULL
LIKES	FLOAT	Yes	NULL
PUBLISHEDAT	TIMESTAMP_NTZ(9)	Yes	NULL
TITLE	VARCHAR(16777216)	Yes	NULL
TRENDING_DATE	DATE	Yes	NULL
VIDEO_ID	VARCHAR(16777216)	Yes	NULL
VIEW_COUNT	FLOAT	Yes	NULL

Figure 4. Data Dictionary TABLE_YOUTUBE_TRENDING



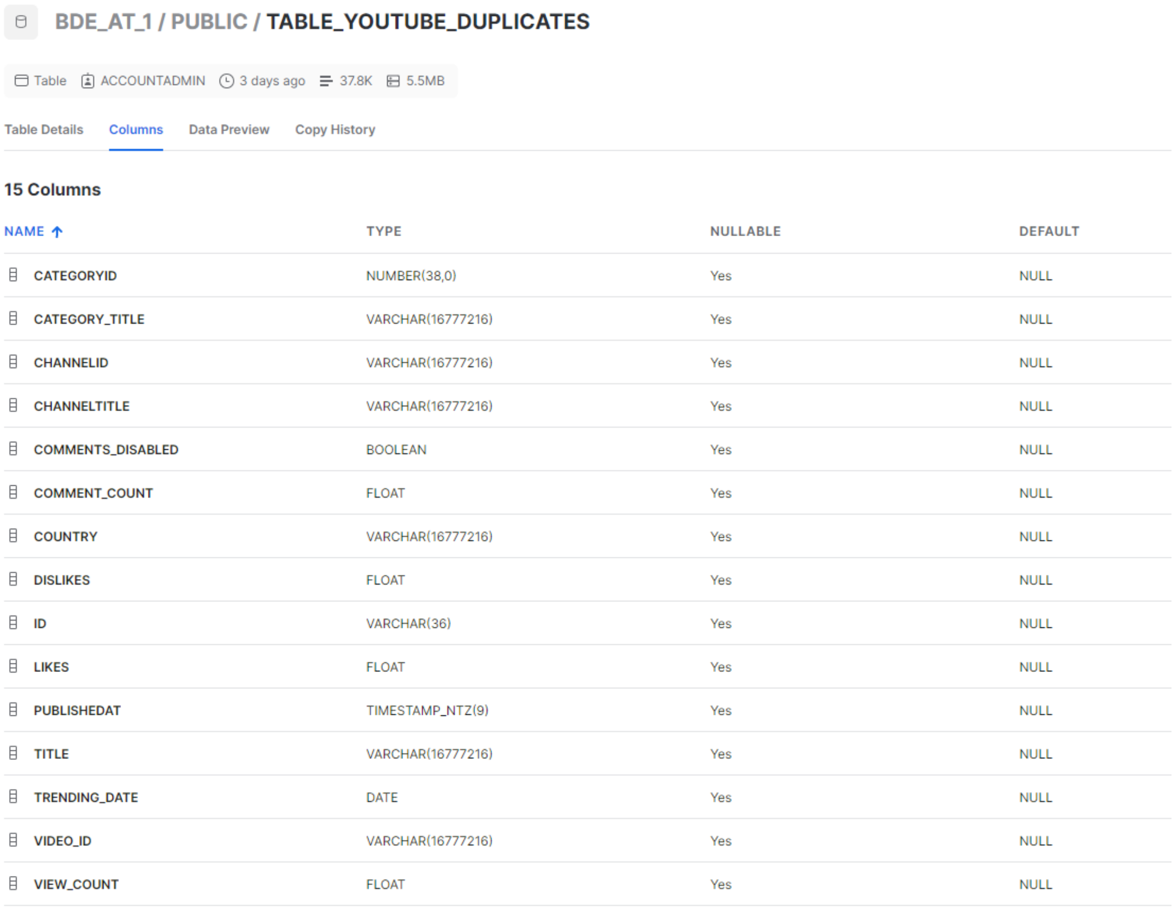
BDE_AT_1 / PUBLIC / **TABLE_YOUTUBE_CATEGORY**

Table | ACCOUNTADMIN | 1 week ago | 342 | 13.5KB

3 Columns

NAME ↑	TYPE	NULLABLE	DEFAULT
CATEGORYID	NUMBER(38,0)	Yes	NULL
CATEGORY_TITLE	VARCHAR(16777216)	Yes	NULL
COUNTRY	VARCHAR(16777216)	Yes	NULL

Figure 5. Data Dictionary TABLE_YOUTUBE_CATEGORY



BDE_AT_1 / PUBLIC / **TABLE_YOUTUBE_DUPLICATES**

Table · ACCOUNTADMIN · 3 days ago · 37.8K · 5.5MB

Table Details · Columns · Data Preview · Copy History

15 Columns

NAME ↑	TYPE	NULLABLE	DEFAULT
CATEGORYID	NUMBER(38,0)	Yes	NULL
CATEGORY_TITLE	VARCHAR(16777216)	Yes	NULL
CHANNELID	VARCHAR(16777216)	Yes	NULL
CHANNELTITLE	VARCHAR(16777216)	Yes	NULL
COMMENTS_DISABLED	BOOLEAN	Yes	NULL
COMMENT_COUNT	FLOAT	Yes	NULL
COUNTRY	VARCHAR(16777216)	Yes	NULL
DISLIKES	FLOAT	Yes	NULL
ID	VARCHAR(36)	Yes	NULL
LIKES	FLOAT	Yes	NULL
PUBLISHEDAT	TIMESTAMP_NTZ(9)	Yes	NULL
TITLE	VARCHAR(16777216)	Yes	NULL
TRENDING_DATE	DATE	Yes	NULL
VIDEO_ID	VARCHAR(16777216)	Yes	NULL
VIEW_COUNT	FLOAT	Yes	NULL

Figure 6. Data Dictionary TABLE_YOUTUBE_DUPLICATES

5. SQL Queries

All the queries are already commented, yet some of the queries are complex to read through even after comments. In this section, we shall discuss the most complex query in the project.

Business Question

For each country, which category_title has the most distinct videos and what is its percentage (2 decimals) out of the total distinct number of videos of that country? Order the result by category_title and country.

SQL Query

SELECT D_BY_COUNTRY.COUNTRY, CATEGORY_TITLE, TOTAL_CATEGORY_VIDEO, TOTAL_COUNTRY_VIDEO, TOTAL_CATEGORY_VIDEO/TOTAL_COUNTRY_VIDEO*100 AS PERCENTAGE

FROM

(SELECT COUNTRY, COUNT(DISTINCT TITLE) AS TOTAL_COUNTRY_VIDEO

```
FROM TABLE_YOUTUBE_FINAL
GROUP BY COUNTRY) D_BY_COUNTRY
INNER JOIN
(SELECT * FROM
(SELECT COUNTRY, CATEGORY_TITLE, TOTAL_CATEGORY_VIDEO, RANK() OVER(PARTITION BY COUNTRY ORDER BY TOTAL_CATEGORY_VIDEO DESC) AS RK
FROM (SELECT COUNTRY, CATEGORY_TITLE, COUNT(DISTINCT TITLE) AS TOTAL_CATEGORY_VIDEO
FROM TABLE_YOUTUBE_FINAL
GROUP BY COUNTRY, CATEGORY_TITLE))
WHERE RK = 1) D_BY_CATEGORY
ON D_BY_COUNTRY.COUNTRY = D_BY_CATEGORY.COUNTRY
ORDER BY CATEGORY_TITLE, COUNTRY;
```

Explanation

The query could be broken down into two parts. The first parts determine the count of distinct video titles by country, whereas the second part determines the count of distinct video titles by country and category. Since, we only need the most viewed video, rank function is used in conjunction with partition function to determine most viewed video for the second part. Snowflake doesn’t allow us to use a where clause when a query contains a window function like rank. So, the rank query was nested with another query to find rank 1 for each category.

These two parts were joined each other using country attributes to display the result in a formatted way.

6. Business Problem

The business is interested to determine if they were to launch a new youtube channel, which category except Music and Entertainment would appear on the top of youtube trend.

To Answer this question, the sum of views grouped by each category was queried from table_youtube_final. The results were as shown below.

CATEGORY_TITLE	**SUM_OF_VIEWS**
Music	739900339874.00
Entertainment	519486675387.00
Gaming	195817817788.00
People & Blogs	187418325929.00
Sports	133651789176.00
Comedy	108493038540.00
Science & Technology	61917522815.00
Film & Animation	52847803701.00
Howto & Style	34743662369.00
News & Politics	34670377110.00
Education	28278127210.00
Autos & Vehicles	15972078815.00
Pets & Animals	5699093234.00
Travel & Events	5670394522.00
Nonprofits & Activism	2770972607.00

The table indicates that after music and entertainment, gaming has the greatest number of views followed by People & Blog category.

But this result could be biased due to popularity of one category in one region. Therefore, we need to answer which of these categories are most popular in all the countries. So, we calculated how many times a category appears in TOP 2 categories of a country.

CATEGORY_TITLE	**SUM_OF_VIEWS**
People & Blogs	9
Gaming	8
Comedy	2
Sports	3

The results indicate that People & Blog is more popular in different countries than gaming.

So, we can conclude that if business wants to create a new youtube, it should create in People and Blog category.

7. References

Snowflake. n.d. Snowflake Documentation.https://docs.snowflake.com/en/user-guide/intro-key-concepts.html