

TABLE OF CONTENT

1. PROJECT OVERVIEW	2
2. INTRODUCTION	2
3. DATA INGESTION AND TRANSFORMATION	3
4. INGESTING JSON DATA	7
5. COMBINING DATA	9
6. DATA CLEANING	10
7. DATA ANALYSING	16
8. BUSINESS QUESTION	20
9. CONCLUSION	22

1. PROJECT OVERVIEW:

For this project, I used Snowflake's Data Lakehouse architecture to analyse YouTube trending and category data across multiple countries, including the United States, Great Britain, Germany, Canada, France, Brazil, Mexico, South Korea, Japan, and India. The dataset included trending video data in CSV format and category information in JSON format. The trending data captured key details such as video titles, channel names, view counts, likes, dislikes, comments, and trending dates, offering insights into popular content and viewing habits across different regions. The category data provided a comprehensive classification of content, with titles and unique IDs for various video formats.

The dataset was initially uploaded to Azure cloud storage and then ingested as external tables into Snowflake. I transformed this data into structured internal tables, ensuring correct data types and integrity. Trending and category data were merged using unique identifiers generated with the `UUID_STRING()` method to create the final table. Data cleaning involved handling duplicates, filling missing values, and ensuring consistency. The analysis focused on identifying top-performing content by region, examining trends in video popularity, and analyzing category preferences over time. This approach offered valuable insights into global digital content consumption trends and highlighted opportunities for optimizing YouTube content strategy.

2. INTRODUCTION

The primary goal of this project is to leverage Snowflake's Data Lakehouse architecture to analyse trending and category data on YouTube from various countries. Utilising databases with trending video data and category specifics from the United States, India, Germany, and Japan, among other nations, our goal was to identify regional variations in video popularity and viewer preferences. The procedure comprised transferring the data to cloud storage on Azure, ingesting it as external tables into Snowflake, transforming the data, and carrying out comprehensive analysis. The results shed light on worldwide content trends, explain what makes videos popular on YouTube, and give content producers tactical advice.

3. DATA INGESTION AND TRANSFORMATION:

The dataset, consisting of YouTube trending and category data, was initially downloaded from provided links and subsequently uploaded to Microsoft Azure Blob Storage. Azure Blob Storage was chosen for its scalable and secure data management capabilities, which facilitated smooth integration with Snowflake for further processing.

Name ↑↓	Type ↑↓	Kind ↑↓	Resource group ↑↓	Location ↑↓	Subscription ↑↓
bdas1	Storage account	StorageV2	Data_Base_1	North Europe	Azure subscription 1

Storage accounts

Default Directory

+ Create

Restore

...

Filter for any field...

Name ↑↓

bdas1

...

bdas1 | Containers

Storage account

Search

+ Container

Change access level

Restore containers

Refresh

Delete

Give feedback

Overview

Activity log

Tags

Diagnose and solve problems

Access Control (IAM)

Data migration

Events

Storage browser

Storage Mover

Partner solutions

Favorites

Containers

Search containers by prefix

Show deleted containers

Name	Last modified	Anonymous access level	Lease state
<input type="checkbox"/> \$logs	28/8/2024, 2:41:41 pm	Private	Available
<input checked="" type="checkbox"/> bd-as-1	28/8/2024, 5:18:07 pm	Private	Available
<input type="checkbox"/> bd-as-1j	29/8/2024, 9:28:39 pm	Private	Available

Upload

Change access level

Refresh

Delete

Change tier

Acquire lease

Break lease

View snapshots

Create snapshot

Give feedback

<input type="checkbox"/>	BR_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	BR_youtube_trending_data.csv	28/8/2024, 5:15:52 pm	Hot (Inferred)	Block blob	45.01 MiB	Available
<input type="checkbox"/>	CA_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	CA_youtube_trending_data.csv	28/8/2024, 5:16:12 pm	Hot (Inferred)	Block blob	43.67 MiB	Available
<input type="checkbox"/>	DE_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	DE_youtube_trending_data.csv	28/8/2024, 5:15:53 pm	Hot (Inferred)	Block blob	44.95 MiB	Available
<input type="checkbox"/>	FR_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	FR_youtube_trending_data.csv	28/8/2024, 5:16:30 pm	Hot (Inferred)	Block blob	43.64 MiB	Available
<input type="checkbox"/>	GB_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	GB_youtube_trending_data.csv	28/8/2024, 5:15:44 pm	Hot (Inferred)	Block blob	43.85 MiB	Available
<input type="checkbox"/>	IN_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	IN_youtube_trending_data.csv	28/8/2024, 5:17:14 pm	Hot (Inferred)	Block blob	49.27 MiB	Available
<input type="checkbox"/>	JP_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	JP_youtube_trending_data.csv	28/8/2024, 5:17:32 pm	Hot (Inferred)	Block blob	56.47 MiB	Available
<input type="checkbox"/>	KR_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	KR_youtube_trending_data.csv	28/8/2024, 5:17:18 pm	Hot (Inferred)	Block blob	53.64 MiB	Available
<input type="checkbox"/>	MX_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	9.86 KiB	Available
<input type="checkbox"/>	MX_youtube_trending_data.csv	28/8/2024, 5:17:30 pm	Hot (Inferred)	Block blob	45.39 MiB	Available
<input type="checkbox"/>	US_category_id.json	30/8/2024, 12:12:33 ...	Hot (Inferred)	Block blob	10.18 KiB	Available
<input type="checkbox"/>	US_youtube_trending_data.csv	28/8/2024, 5:17:53 pm	Hot (Inferred)	Block blob	43.53 MiB	Available

The datasets were accurately loaded from Azure Blob Storage into Snowflake, transformed into the required formats, and combined into a comprehensive final table. This setup provides a robust foundation for subsequent data analysis tasks. The use of Azure Blob Storage facilitated efficient data management, while Snowflake’s powerful features enabled effective data integration and transformation

3.1 DATABASE AND STAGE CREATION:

An essential first step in effectively organising and managing datasets in Snowflake is to create a new database. Tables, schemas, and other database objects are logically contained within databases in Snowflake. It aids in data structure that promotes security, efficiency in querying, and data integrity. For this project, a new snowflake database named 'assignment' was created

```
/* Creating a DataBase called Assignment1 */
```

```
CREATE OR REPLACE DATABASE assignment_1;  
USE DATABASE assignment_1;
```

The 'stage_assignment' stage was created to link snowflake with azure blob storage. Allowing snowflake to access the data files directly from the cloud storage.

```
CREATE OR REPLACE STAGE stage_assignment  
URL='azure://bdas1.blob.core.windows.net/bd-as-1'  
CREDENTIALS=(AZURE_SAS_TOKEN='?sv=2022-11-  
02&ss=b&srt=co&sp=rwdlaciylfx&se=2024-12-30T14:19:42Z&st=2024-  
08-  
28T07:19:42Z&spr=https&sig=Fq9YilGA6fq18X2w9dnJJoAiTn732LYAvqGO  
mPrpmP4%3D')  
.
```

3.2 INGESTING CSV DATA:

FILE FORMAT: Defining a file format in Snowflake is a critical step when ingesting data from external sources. This process ensures that Snowflake correctly interprets and processes the data according to its structure and characteristics.

```
--WE going to create a file format skipping the header, adding  
delimiters and type as csv to create a external table  
CREATE OR REPLACE FILE FORMAT file_format_csv  
TYPE = 'CSV'  
FIELD_DELIMITER = ','  
SKIP_HEADER = 1  
NULL_IF = ('\N', 'NULL', 'NUL', '')  
FIELD_OPTIONALLY_ENCLOSED_BY = ''  
;
```

A file format for CSV was defined to correctly interpret the structure of the data files.

3.3 EXTERNAL TABLE CREATION:

An external table 'ex_table_youtube_trending' was created to read the csv files from azure blob storage based on the file format we created.

--Created a external table using the file format we created

```
CREATE OR REPLACE EXTERNAL TABLE ex_table_youtube_trending
WITH LOCATION = @stage_assignment
FILE_FORMAT = file_format_csv
PATTERN = '[A-Z]{2}_youtube_trending_data.csv'; -- patten will
help us to bring all the csv file from the storage.
```

	VALUE
1	{ "c1": "uq5LCIQN3cE", "c10": "105756", "c11": "139946", "c2": "안녕하세요 보컬입니다", "c3": "2020-08-09T09:32:48Z", "c4": "UCu9BctGIEr73LXZsKmqW", "c5": "보컬 BK", "c6": "24", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
2	{ "c1": "I-ZbZCHsHD0", "c10": "494", "c11": "3339", "c2": "부라토스의 계획 [송영명 프리퀀]", "c3": "2020-08-12T09:00:08Z", "c4": "UCRuSxVv4iqTK5kCh90ntAgA", "c5": "송영명", "c6": "1", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
3	{ "c1": "9d7JNUjBoss", "c10": "68898", "c11": "50688", "c2": "평생 반성하면서 살겠습니다.", "c3": "2020-08-10T09:54:13Z", "c4": "UCMVC92EOs9yDJG5JS-CMesQ", "c5": "양광 YangPang", "c6": "22", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
4	{ "c1": "3pLL3-sMVg", "c10": "1798", "c11": "8751", "c2": "안녕하세요 파워를입니다.", "c3": "2020-08-11T15:00:58Z", "c4": "UCkQCwnkQfgSuPTTnw_Y7v7w", "c5": "파워를 Quaddurup", "c6": "24", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
5	{ "c1": "zrsBjYukE8s", "c10": "9763", "c11": "23405", "c2": "박진영 (J.Y. Park) When We Disco (Duet with 선미) M/V", "c3": "2020-08-11T09:00:13Z", "c4": "UCaO6TYIC8U5ttz6zhTrZgg", "c5": "JYP Entertainment", "c6": "10", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
6	{ "c1": "jbGRowa5tIk", "c10": "15176", "c11": "31040", "c2": "ITZY 'Not Shy' M/V TEASER", "c3": "2020-08-11T15:00:13Z", "c4": "UCaO6TYIC8U5ttz6zhTrZgg", "c5": "JYP Entertainment", "c6": "10", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
7	{ "c1": "X-TPOQEYRGo", "c10": "393", "c11": "834", "c2": "흑인 마동석이 출연하는 마트를 털러온 강도들에게 벌어진 일 ㅋㅋ", "c3": "2020-08-10T09:37:33Z", "c4": "UCpCiIDf9UrfRqte55FHWIYQ", "c5": "드림텔러(DreamTeller)", "c6": "24", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
8	{ "c1": "REUA4roJndU", "c10": "13337", "c11": "18286", "c2": "진심으로 사과드립니다. 죄송합니다.", "c3": "2020-08-11T14:00:01Z", "c4": "UCwZTeeKyb1hT6sg8tOaZIA", "c5": "일다TV", "c6": "24", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
9	{ "c1": "7FOI40uehC4", "c10": "338", "c11": "5697", "c2": "집에 혼자 있을 때 하는 짓ㅋㅋㅋㅋㅋㅋ [핑포발발]", "c3": "2020-08-12T09:00:02Z", "c4": "UCCJkwrmlhQWksFV-sQo4Qw", "c5": "핑포발발Minggo", "c6": "22", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
10	{ "c1": "0dsnm3n6ZdM", "c10": "103", "c11": "2324", "c2": "정윤호가 무려 6시간 공들여 만든 세기의 발명품 [발명왕] Ep.1", "c3": "2020-08-11T09:30:00Z", "c4": "UC0S0PwEH3idvemsDvKaYgGA", "c5": "달라스튜디오", "c6": "24", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
11	{ "c1": "rFwZqtPc-Ss", "c10": "344", "c11": "1622", "c2": "[책이벤트] 진짜 인물루머서로 사는 법 황태환 @비글부부 Bgeul Bubu 하준파파 에이치유지 대표 치유 회복 사랑 컴패션 세바시 1217회", "c3": "2020-08-10T09:00:12Z", "c4": "UCaO6TYIC8U5ttz6zhTrZgg", "c5": "JYP Entertainment", "c6": "10", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
12	{ "c1": "7Y8Vv_KH7l", "c10": "1309", "c11": "1166", "c2": "🔴 SUB [물커] 누가봐도 강도인데 전혀 의심을 안하는 직원들 본다면!!!! 열 커피를 제대로 터졌음 ㅋㅋㅋㅋㅋ - [동네농들 HOODBOYZ]", "c3": "2020-08-09T11:30:01Z", "c4": "UCaO6TYIC8U5ttz6zhTrZgg", "c5": "JYP Entertainment", "c6": "10", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
13	{ "c1": "rRaBKB9gDSQ", "c10": "2863", "c11": "16283", "c2": "뒷광고 논란에 대한 해명 및 진할 말씀이 있습니다", "c3": "2020-08-09T16:12:59Z", "c4": "UCBLoXzDldCnpbM_7uyG0_Tg", "c5": "HONG SOUND", "c6": "24", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }
14	{ "c1": "TadJcNUIdTA", "c10": "378", "c11": "236", "c2": "세계 1위 재벌의 손자가 납치당하자 벌어진 충격적인 일.ㄷㄷ(심화)", "c3": "2020-08-11T07:00:04Z", "c4": "UCor2S3SBmivrx9Q4oLUZHw", "c5": "movie trio 무비트리", "c6": "24", "c7": "2020-08-12T00:00:00Z", "c8": "2020-08-12T00:00:00Z", "c9": "2020-08-12T00:00:00Z" }

The PATTERN function specifies which files to include based on their names, using regular expressions to filter data files.

3.4 SCHEMA ADJUSTMENT:

In Snowflake, correcting the external table schema involves adjusting data types and extracting relevant information to ensure that the data is accurately represented and accessible.

```
CREATE OR REPLACE EXTERNAL TABLE ex_table_youtube_trending
(
  Videoid STRING as (value:c1::STRING),
  TITLE STRING as (value:c2::STRING),
  PUBLISHEDAT DATE as (value:c3::DATE),
  CHANNELID STRING as (value:c4::STRING),
  CHANNELTITLE STRING as (value:c5::STRING),
  CATEGORYID INT as (value:c6::INT),
  TRENDING_DATE DATE as (value:c7::DATE),
  VIEW_COUNT INT as (value:c8::INT),
  LIKES INT as (value:c9::INT),
  DISLIKES INT as (value:c10::INT),
  COMMENT_COUNT INT as (value:c11::INT),
  COUNTRY STRING AS SUBSTRING(METADATA$FILENAME, 1, 2)
)
WITH LOCATION = @stage_assignment
FILE_FORMAT = file_format_csv
PATTERN = '[A-Z]{2}_youtube_trending_data.csv';;
```

	VALUE	VIDEOID	TITLE	PUBLISHEDAT	CHANNELID	CHANNELTITLE	CATEGORYID	TRENDING_DATE	VIEW_CC
1	{ "c1": "J78aPJ3VYnS",	J78aPJ3VYnS	I left youtube for a mo	2020-08-11	UCYzPXprvISY-Sf0g4v	jacksepticeye	24	2020-08-12	203
2	{ "c1": "9nidKH8cM38	9nidKH8cM38	TAXI CAB SLAYER KILL	2020-08-11	UCFmbX7frWZfuWdjaI	Eleanor Neale	27	2020-08-12	23
3	{ "c1": "M9Pmf9AB4M	M9Pmf9AB4Mo	Apex Legends Storie	2020-08-11	UC0ZV6M2THA81QT9	Apex Legends	20	2020-08-12	238
4	{ "c1": "kgUV1MaD_M8	kgUV1MaD_M8	Nines - Clout (Official	2020-08-10	UCvDkzrj8ZPIBqRd8fb	Nines	24	2020-08-12	61
5	{ "c1": "49Z6Mv4_WC	49Z6Mv4_WCA	i don't know what im d	2020-08-11	UCTinbF-Q-FVthAQrFi	CaseyNeistat	22	2020-08-12	94
6	{ "c1": "ua4QMFOATcc	ua4QMFOATco	CGP Grey was WRONG	2020-08-11	UC2C_jShtL725nvbm1	CGP Grey	27	2020-08-12	105
7	{ "c1": "x-KbnJ9fvJc",	x-KbnJ9fvJc	Kya Baat Aa : Karan Au	2020-08-11	UCm9SZAIO3Rev9sFw	Rehaan Records	10	2020-08-12	1130
8	{ "c1": "3C66w5Z0ixs"	3C66w5Z0ixs	I ASKED HER TO BE M	2020-08-11	UCvtRTOMP2TqYqu51	Brawadis	22	2020-08-12	151
9	{ "c1": "ZNf6MbO_AHc	ZNf6MbO_AHo	Popek ft. Dr Alban - It	2020-08-12	UC8Mh9UmrfaQPEcyc	Kr6lAlbanITV	24	2020-08-12	27
10	{ "c1": "VIUo6yapDbc"	VIUo6yapDbc	Ultimate DIY Home Mc	2020-08-11	UCDVPcEbVLQgLZXOf	Mr. Kate	26	2020-08-12	112
11	{ "c1": "pAEa5BnAUgC	pAEa5BnAUgC	Champions League pri	2020-08-11	UC2-0sEOYbQFuaURc	OneFootball English	17	2020-08-12	21
12	{ "c1": "cSiKfmgRtIs	cSiKfmgRtIs	Sensible Transfers: Ar	2020-08-12	UCGYNGmyhZ_kwBF	Tifo Football	17	2020-08-12	12
13	{ "c1": "w-hVKXmib1c"	w-hVKXmib1c	I've Seen Something K	2020-08-10	UCHhfSXoD06gSgpOv	Behzinga	20	2020-08-12	87
14	{ "c1": "xTor7oiQu2M"	xTor7oiQu2M	FUT 21 Official Trailer	2020-08-10	UCovaxd5LQSuP4Chk	EA SPORTS FIFA	20	2020-08-12	121

The external table schema was corrected to match the data types and using substring function we extract the country information from the file name And included in the table.

3.5 CREATE INTERNAL TABLE:

An internal table 'table_youtube_trending' was created and all the data was moved from external table to internal. Since we are allowed to modify the

internal table, The VALUE column was dropped since it was no longer needed.

```
CREATE OR REPLACE TABLE table_youtube_trending AS
SELECT *
FROM ex_table_youtube_trending
;

ALTER TABLE table_youtube_trending DROP COLUMN VALUE ;
SELECT*
FROM table_youtube_trending;
```

	VIDEOID	TITLE	PUBLISHEDAT	CHANNELID	CHANNELTITLE	CATEGORYID	TRENDING_DATE
1	J78aPJ3VyNs	I left youtube for a month and THIS is what happened.	2020-08-11	UCYzPXprvISY-Sf0g4vX-m6g	jacksepticeye	24	2020-08-12
2	9nidKH8cM38	TAXI CAB SLAYER KILLS 'TO KNOW HOW IT FEELS'	2020-08-11	UCFmbX7frWZfuWdjAML0babA	Eleanor Neale	27	2020-08-12
3	M9Pmf9AB4Mo	Apex Legends Stories from the Outlands - "The Endorsement"	2020-08-11	UC0ZV6M2THA81QT9hrVWUG3A	Apex Legends	20	2020-08-12
4	kgUV1MaD_M8	Nines - Clout (Official Video)	2020-08-10	UCvDkzrj8ZPiBgRd6fidxhTw	Nines	24	2020-08-12
5	49Z6Mv4_WCA	i don't know what im doing anymore	2020-08-11	UCtinbF-Q-fVthA0qrFQTgXQ	CaseyNeistat	22	2020-08-12
6	ua4QMFQATco	CGP Grey was WRONG	2020-08-11	UC2C_jShtL725hvbmlarSV9w	CGP Grey	27	2020-08-12
7	x-KbnJ9fvJc	Kya Baat Aa : Karan Aujla (Official Video) Tania Sukh Sanghera D	2020-08-11	UCm9SZA03Rev9sFwoCdZlg	Rehaan Records	10	2020-08-12
8	3C86w5Z0ixs	I ASKED HER TO BE MY GIRLFRIEND...	2020-08-11	UCvtRTOMP2TqYqu51xNrqAzg	Brawadis	22	2020-08-12
9	ZNfeMb0_AHo	Popek ft. Dr Alban - It's My Life (prod. Claysteer)	2020-08-12	UC8Mh9UmrtaQPecybdWvQsOg	KrdiAlbanITV	24	2020-08-12
10	VIUo6yapDbc	Ultimate DIY Home Movie Theater for The LaBrant Family!	2020-08-11	UCDVPcEbVLQglZX0R6jp34A	Mr. Kate	26	2020-08-12
11	pAe5BnAUgQ	Champions League preview: Messi v Boateng part 2! ► OneFootb	2020-08-11	UC2-0sEOYbQFuaURd_AU6Krg	OneFootball English	17	2020-08-12
12	cSlrkfmgR1s	Sensible Transfers: Arsenal	2020-08-12	UCGYYNgmyhZ_kw8F_lqqXdAQ	Tifo Football	17	2020-08-12
13	w-hvKXmib1c	I've Seen Something KSI Hasn't Shown You...	2020-08-10	UCHhfSXoD66gSgpOvLH4wrRw	Behzinga	20	2020-08-12
14	xTor7oiQu2M	FUT 21 Official Trailer	2020-08-10	UCovaxdSLQSuP4ChkxK0onZQ	EA SPORTS FIFA	20	2020-08-12

4. INGESTING JSON DATA:

4.1 CREATING AN EXTERNAL TABLE:

An 'ex_table_youtube_category' external table is created to store and query data directly from Json files located on the azure blob storage specified by the 'stage_assignment' stage. We mentioned file format as Json. It specifies the data files are in Json format. The PATTERN clause uses a regular expression to filter and match only specific files in the storage location. The pattern'. _category_id[.]Json' targets all JSON files whose filenames end with _category_id. json.

```
CREATE OR REPLACE EXTERNAL TABLE ex_table_youtube_category
WITH LOCATION = @stage_assignment
FILE_FORMAT = (TYPE=JSON)
PATTERN = '.*_category_id[.]json';
```

	VALUE
1	{ "etag": "kBCy3l9KLHHU79W4lp5196LDpt", "items": [{ "etag": "lFWa37JGcqZs-jZeAyFGkbeh6bc", "id": "1", "kind": "youtube#videoCategory", "snippet": { "assignable": true, "channelId": "UCBR8-f

4.2 CREATING THE INTERNAL TABLE BY USING LATERAL FLATTEN:

By extracting and converting JSON data from an external table called `ex_table_youtube_category`, the SQL statement produces a table called `table_youtube_category`.

`Split_part` extracts the country code from the file name from the file and allow us to include in the table.

The lateral flatten function is used to expand the nested json arrays. It enables each item within the array to become a separate row in the output. This function is crucial for handling json data structures, allowing you to extract fields like 'id' and 'snippet:title' for each category in normalized format.

```
CREATE OR REPLACE TABLE table_youtube_category AS
SELECT
    split_part(metadata$filename, '_', 1) as country,
    l.value:id::string as CATEGORYID,
    l.value:snippet:title::string as CATEGORY_TITLE
FROM ex_table_youtube_category, LATERAL FLATTEN(value:items) l;
```

	COUNTRY	CATEGORYID	CATEGORY_TITLE
1	KR	1	Film & Animation
2	KR	2	Autos & Vehicles
3	KR	10	Music
4	KR	15	Pets & Animals
5	KR	17	Sports
6	KR	18	Short Movies
7	KR	19	Travel & Events
8	KR	20	Gaming
9	KR	21	Videoblogging
10	KR	22	People & Blogs
11	KR	23	Comedy
12	KR	24	Entertainment
13	KR	25	News & Politics
14	KR	26	Howto & Style

5.COMBINING DATA:

The final internal table `table_youtube_final` was created by merging `table_youtube_trending` with `table_youtube_category` using a LEFT JOIN. A unique identifier was added to each record using `UUID_STRING()`. It generates generating a distinct UUID for each row

```
CREATE OR REPLACE TABLE table_youtube_final AS
SELECT
    UUID_STRING() AS id, --UUID_STRING() function is executed for each row in the table, generating a distinct UUID for each row
    t.VideoId AS VIDEO_ID,
    t.TITLE AS TITLE,
    t.PUBLISHEDAT AS PUBLISHEDATE,
    t.CHANNELID AS CHANNEL_ID,
    t.CHANNELTITLE AS CHANNEL_TITLE,
    t.CATEGORYID AS CATEGORY_ID,
    c.CATEGORY_TITLE AS CATEGORY_TITLE,
    t.TRENDING_DATE AS TRENDING_DATE,
    t.VIEW_COUNT AS VIEW_COUNT,
    t.LIKES AS LIKES,
    t.DISLIKES AS DISLIKES,
    t.COMMENT_COUNT AS COMMENT_COUNT,
    t.COUNTRY AS COUNTRY
FROM table_youtube_trending t
LEFT JOIN table_youtube_category c
    ON t.COUNTRY = c.COUNTRY AND t.CATEGORYID = c.CATEGORYID;

SELECT*FROM table_youtube_final;

--ROW COUNT
SELECT COUNT(*)
FROM table_youtube_final;
```

	ID	VIDEO_ID	TITLE	PUBLISHEDATE	CHANNEL_ID	CHAN
1	0b87e4dd-c8d6-4f3d-b1fe-17a2fb5bc47d	7rlwxSPUCQk	ON EST POSITIF AU COVID-19 (coronavirus)	2020-08-11	UCpWaR3gNAQGsX48cIIQC0qw	Tibo
2	60a1e41f-dfc8-48c8-a064-3bd2ee894b1b	AcBd_RH9JSw	PASSER UNE NUIT DANS LA PISCINE DE LA VILLA AVEC @Michou !	2020-08-11	UCUj7mwOyySFzZUkq4H29nug	LeBo
3	0f7c3e9a-2766-452c-bbc8-694e2537a268	JVm8P6kKgD0	FRANGLISH - My Salsa feat. Tory Lanez (clip officiel)	2020-08-12	UCnwfco0-LGEg8y9bEQTaSgQ	Franç
4	c024c526-a472-40a0-a26e-8a4619798fd2	JfgeD7xWy-w	L'ÉTÉ LE PLUS ECLATÉ DE MA LIFE	2020-08-11	UCMJ2VQ3-8zpyeA17dU0T-Bg	Fahd
5	1b807a99-f410-4d78-92c4-ab1455ed67ad	P3lKBxra3a8	JE L'ESSAYE ENFIN ! DUCATI HYPERMOTARD	2020-08-12	UC-uy71hColMpQ_2rs-kY6w	KIKAI
6	ea3acc67-595b-4ae9-83bf-67943498d4f8	AwdfloSJrtA	MÉLI-MÉLO... le 3.	2020-08-09	UCDPK_MTu3uTUFJXRvTJcEw	Mcfly
7	77ea8e4a-ce8d-4b23-a9b8-6c784b8c51fe	J9ITZ20-ITc	Inoxtag attend Léa devant la Villa... Il n'a toujours pas compris... #7	2020-08-10	UCNqG4mP3Ds5OUGJp8lUOcW	Valou
8	b589d8d2-bf05-4bce-8875-d7c3ee9d8efa	zatMgEXo4Ko	LDC - MES PRONOS DU FINAL 8 !	2020-08-12	UCopwMeeGavRDZbo3EFB7NUg	Bruce
9	af2a93a9-d05a-46df-b10f-4960b09a3256	56HGTfZabxA	Inoxtag - Funkinox ft. Kazzey (Clip officiel)	2020-08-09	UCL9aTJb0ur4s0vxcppAopEw	Inoxi
10	408b9608-a07d-457c-90df-3fe5caba190a	n0rGxQw5ics	MICHO - DANS LE CLUB (PARODIE FORTNITE) SAISON 3 CHAPITRE	2020-08-12	UCHGhSbkm9JS_u1thVCeGpiw	Kebo
11	134c625d-297d-4f75-b274-3be86cea9b60	DTs12U36x54	VOS SOIRÉES QUI ONT FAILLI (TRÈS) MAL FINIR ! (Feat Amixem)	2020-08-09	UCow2lGnug1l3Xazkr5jM_Q	JOYC
12	d723cc9b-95d3-4b46-b613-1baf913499d	JbGR0wa5tlk	ITZY "Not Shy" M/V TEASER	2020-08-11	UCa06TYiC8USltz62hTzgg	JYP E
13	58934716-0baf-430b-b590-a86a050e2f23	emuvKXZ0-f4	MON HOUSE TOUR ! (Je vous présente ma maison)	2020-08-11	UCTHkQBKXqsytdNccmKl8pGw	Lytha
14	6c23fa3c-7dc7-4842-ac7b-52f9c766b95b	F14CK7vniUo	le meilleur anniversaire de toute sa vie	2020-08-11	UCxDSG8fSVisYtE.0a6YJ7Q	Esile

The row count of `table_youtube_final` was verified to ensure completeness, resulting in 2,667,041 rows.

	COUNT(*)
1	2667041

6.DATA CLEANING:

This report describes the procedures and queries that were used in Snowflake's 'table_youtube_final' and 'table_youtube_category' tables to organise, clean, and deduplicate data. Ensuring data consistency and integrity was the goal because these are essential for precise analysis of trends in YouTube videos.

1. In table_youtube_category, Which category_title Has Duplicates if We Don't Take Into Account the categoryid(Return Only a Single Row)?

```
-- 1: In "table_youtube_category" which category_title has  
duplicates if we don't take into account the categoryid (return  
only a single row)?
```

```
SELECT CATEGORY_TITLE FROM table_youtube_category  
GROUP BY CATEGORY_TITLE  
HAVING  
    COUNT(DISTINCT CATEGORYID) > 1  
LIMIT 1;
```

This query identifies category_title entries that have multiple categoryid values, ignoring the category id's uniqueness. By grouping by CATEGORY_TITLE and using HAVING COUNT, this query returns any title associated with more than one category_id, highlighting potential inconsistencies in how categories are labeled across different datasets or countries.

	CATEGORY_TITLE
1	Comedy

2. In table_youtube_category, Which category_title Only Appears in One Country?

```
-- 2: In "table_youtube_category" which category_title only  
appears in one country?
```

```
SELECT  
    CATEGORY_TITLE  
FROM  
    table_youtube_category  
GROUP BY  
    CATEGORY_TITLE  
HAVING  
    COUNT(DISTINCT COUNTRY) = 1;
```

This query aims to identify category_title entries that are unique to a single country. By grouping by CATEGORY_TITLE and filtering with HAVING COUNT(DISTINCT COUNTRY) = 1, it pinpoints titles that appear exclusively within one country's dataset. This information is useful for understanding regional specificity in content categorization.

	CATEGORY_TITLE
1	Nonprofits & Activism

3. In table_youtube_final, what is the categoryid of the Missing category_titles?

```
-- 3: In "table_youtube_final", what is the categoryid of the  
missing category_titles?
```

```
SELECT CATEGORY_ID  
FROM table_youtube_final  
WHERE CATEGORY_TITLE IS NULL
```

This query selects the CATEGORY_ID from records where CATEGORY_TITLE is NULL in table_youtube_final. Identifying these entries is crucial for data completeness, as missing category_titles may lead to gaps in data analysis or misinterpretation of trends.

	CATEGORY_ID
1	29
2	29
3	29
4	29
5	29
6	29
7	29
8	29
9	29
10	29
11	29
12	29
13	29
14	29

4.Update the table_youtube_final to Replace the NULL Values in category_title with the Answer from the Previous Question

--4.Update the table_youtube_final to replace the NULL values in category_title with the answer from the previous question.

```
UPDATE table_youtube_final
SET category_title = (
    SELECT CATEGORY_TITLE FROM table_youtube_category
    GROUP BY CATEGORY_TITLE
    HAVING COUNT(DISTINCT COUNTRY)=1
)
WHERE category_title IS NULL;
```

This update operation replaces NULL values in category_title in the table_youtube_final table. The subquery identifies unique CATEGORY_TITLE entries that appear in only one country from table_youtube_category. Filling these gaps ensures that all records are fully populated, maintaining the dataset's integrity and completeness.

	number of rows updated	numb
1	1563	

5. In table_youtube_final, Which Video Doesn't Have a channeltitle (Return Only the Title)?

--5: In "table_youtube_final", which video doesn't have a channeltitle (return only the title)?

```
SELECT TITLE
FROM table_youtube_final
WHERE CHANNEL_TITLE IS NULL;
```

This query identifies videos that lack a CHANNEL_TITLE, returning only their titles. Detecting these gaps is essential for data completeness and ensuring every video is properly attributed to a channel, which is important for analysing channel-specific trends.

	TITLE
1	Kala Official Teaser Tovino Thomas Rohith V S Juvis Productions Adventure Company

6. Delete from table_youtube_final, Any Record with video_id = "#NAME?"

--6:Delete from "table_youtube_final", any record with video_id = "#NAME?"

```
DELETE FROM table_youtube_final
WHERE video_id = '#NAME?';
```

This delete statement removes records with video_id values set to #NAME?. These entries likely result from data import errors or placeholders and can skew analysis if left unaddressed. Cleaning these records ensures the dataset's accuracy.

	number of rows deleted
1	32081

7. Create a New Table Called table_youtube_duplicates Containing Only the “Bad” Duplicates Using the ROW_NUMBER() Function

--7.Create a new table called “table_youtube_duplicates” containing only the “bad” duplicates by using the row_number() function

```
CREATE OR REPLACE TABLE table_youtube_duplicates AS
SELECT *
FROM (
    SELECT *,
           ROW_NUMBER() OVER (PARTITION BY video_id, country,
trending_date ORDER BY view_count desc) AS row_num
    FROM table_youtube_final
) AS duplicates
WHERE row_num > 1;

select*from table_youtube_duplicates;
```

from table_youtube_final. The ROW_NUMBER() function is used to rank each record within partitions defined by video_id, country, and trending_date. The ORDER BY view_count DESC clause ensures that the highest view count gets a row_num of 1, marking others as duplicates. Rows with row_num > 1 are considered duplicates and stored in the new table.

	ID	VIDEO_ID	TITLE	PUBLISHEDDATE	CHANN
11	a0307c40-30c4-4ce0-9e0a-4232a1c77213	mnpvrt1dgvtA	【激怒争論】地獄禁止! 京境アヤシクレスがスリル海賊トキドナダにwwwwww	2021-06-09	UCpYV
12	6490d82d-988b-489f-af17-3b2593211fc7	0y3CKfYPcW4	【#スバちょこルーナ】地獄くじ引き? ! 足つぽダンス? ! ? かかってこいや! ! ! !	2021-06-03	UCvzGI
13	74cf0946-99d7-413e-a09e-b21be02214ff	Cj2f296Juuw	Mitten im Lockdown plötzlich Tourette: Wie klappt eine Rückkehr in die Normalität?	2021-06-10	UCLoW
14	1a6c21e7-f120-41b0-abaf-51c7e6a5621c	iIWJ04CvBS8	MARIDO ADIVINHANDO PREÇOS DE MAQUIAGEM PARTE 2 Dearo e Manu	2021-06-05	UCQ4J
15	9efc5d44-b314-4ef5-99d3-c1a24d923213	ihHZFuW-9bg	Sath Mai - Emiway Bantai X Swaailina X Meme Machine [Music Video]	2021-06-06	UCuvel
16	3f5a94e0-d97b-489a-adbf-3f1eae7cdce	JiqXNzazKnU	30 weds 21 Web Series Kurravaadu Lyrical Video Song Girl Formula Chai Bisket	2021-06-04	UCswa
17	a5a65aec-2fc7-436d-8d13-c85d9e9e4d46	h3OXnPWn8ts	Krasser Business-Plan! 🙌 #389 Krass Schule	2021-06-09	UCKJG1
18	2c43a11b-d88c-4842-84b4-3ca6b8ef261f	GYcSgA5bw4	PARAGUAI 0 X 2 BRASIL MELHORES MOMENTOS ELIMINATÓRIAS DA COPA ge	2021-06-09	UCgCK
19	695d8619-df24-4b2f-9cc3-824451abb407	cEzMQhy7Ltg	네 남자의 코스플레이	2021-06-01	UCyozk
20	cc98aa5c-5752-4e30-bacc-a3bfead8c972	TCMY7yA3IOk	We Spent 24 Hrs. Locked In Our Office (No Internet)	2021-06-02	UCpi8T
21	a6666fe1-de6b-4e1a-be14-ac961ed9597a	QGLz_1W3Br0	TWICE(트와이스) - Alcohol-Free (Music Bank) KBS WORLD TV 210611	2021-06-11	UC5BM
22	7aacc966-a408-4a8a-h46a-788276b0a03b	C100TC-BRII1	I Trilled Youtubers that I not Fortnite SEASON 7 aarful	2021-06-02	UC18Rd

8. Delete the Duplicates in table_youtube_final Using table_youtube_duplicates

```
--8 Delete records from table_youtube_final that are found in table_youtube_duplicates

DELETE FROM table_youtube_final
WHERE id IN (
    SELECT id
    FROM table_youtube_duplicates);
```

This delete statement removes duplicate entries from table_youtube_final by matching records against table_youtube_duplicates. Using the id field ensures that only unwanted duplicates are deleted while retaining the most accurate records for each unique combination of video_id, country, and trending_date.

	number of rows deleted
1	37466

9. Count the Number of Rows in table_youtube_final and Check That It Is Equal to 2,597,494 Rows

```
--9 .Count the number of rows in "table_youtube_final" and
check that it is equal to 2,597,494 rows. rows.
```

```
SELECT COUNT(*)
FROM table_youtube_final;
```

This final check ensures the table_youtube_final table contains the expected 2,597,494 rows after the cleaning process. Verifying the row count confirms that all duplicates have been correctly managed and that no unintended deletions occurred, ensuring the dataset's integrity and readiness for further analysis.

	COUNT(*)
1	2597494

7. DATA ANALYSIS:

1. What are the 3 Most Viewed Videos for Each Country in the “Gaming” Category for trending_date = '2024-04-01'?

--1. What are the 3 most viewed videos for each country in the "Gaming" category for the trending_date = '2024-04-01'.
Order the result by country and the rank

```
SELECT country, title, channel_title, view_count, RANK() OVER
(PARTITION BY country ORDER BY view_count DESC) AS RK
FROM
    table_youtube_final
WHERE
    category_title = 'Gaming'
    AND trending_date = '2024-04-01'
QUALIFY
    RK <= 3
ORDER BY
    country,
    RK;
```

The query identifies the top three most viewed videos in the "Gaming" category for each country on April 1, 2024. It uses the RANK() window function partitioned by country and ordered by view_count in descending order to assign a rank to each video. The QUALIFY clause ensures only the top three videos (ranked 1 to 3) for each country are included in the results. This helps highlight the most popular gaming content across different regions on a specific date.

2. For Each Country, Count the Number of Distinct Videos with a Title Containing the Word "BTS"

--2. For each country, count the number of distinct video with a title containing the word "BTS" and order the result by count in a descending order

```
SELECT country, COUNT(DISTINCT VIDEO_ID) AS Number_Count
FROM table_youtube_final
WHERE title LIKE '%BTS%'
GROUP BY country
ORDER BY Number_Count DESC;
```

This query counts the number of unique videos containing the word "BTS" (case insensitive) in their title for each country. The COUNT(DISTINCT VIDEO_ID) function ensures that each video is counted only once, even if it appears multiple times. The results are then ordered in descending order based on the count, highlighting countries with the most "BTS"-related content.

3. For Each Country, Year, and Month (Combined into a Single Column), Identify the Most Viewed Video and Its likes_ratio

--3. For each country, year and month (in a single column), which video is the most viewed and what is its likes_ratio (defined as the percentage of Likes against view_count) truncated to 2 decimals. Order the result by year_month and country.

```
WITH VideoStats AS (
    SELECT
        country,
        EXTRACT(YEAR FROM trending_date) AS year,
        EXTRACT(MONTH FROM trending_date) AS month,
        video_id,
        title,
        channel_title,
        category_title,
        view_count,
        likes,
        dislikes,
        CASE
            WHEN view_count = 0 THEN 0
            ELSE ROUND((likes::NUMERIC / view_count) * 100, 2)
        END AS likes_ratio,
        RANK() OVER (PARTITION BY country, EXTRACT(YEAR FROM trending_date), EXTRACT(MONTH FROM trending_date) ORDER BY view_count DESC) AS
        video_rank
    FROM
        table_youtube_final
)
SELECT
    country,
    TO_DATE(year || '-' || TO_CHAR(month, 'FM00') || '-01', 'YYYY-MM-DD') AS year_month,
    title,
    channel_title AS CHANNELTITLE,
    category_title,
    view_count,
    likes_ratio
FROM
    VideoStats
WHERE
    video_rank = 1
    AND year = 2024
ORDER BY
    year_month,
    country;
```

The query calculates the most viewed video and its likes_ratio (likes as a percentage of view count) for each country, year, and month for 2024. Using a WITH clause, it extracts the year and month from trending_date, computes the likes_ratio, and ranks videos by view count. The final selection filters for the top-ranked (most viewed) video for each month and country, providing insight into top-performing videos and their viewer engagement.

	COUNTRY	YEAR_MONTH	TITLE	CHANNELTITLE	CATEGORY_TITLE	VIEW_COUNT	LIKES_RATIO
1	BR	2024-01-01	Survive 100 Days Trapped, Win \$500,000	MrBeast	Entertainment	139504939	3.20
2	CA	2024-01-01	Still Here Season 2024 Cinematic - League of Legends (ft. Forts, Tiffany Aris, and 2W	League of Legends	Gaming	104159411	1.69
3	DE	2024-01-01	Still Here Season 2024 Cinematic - League of Legends (ft. Forts, Tiffany Aris, and 2W	League of Legends	Gaming	104159411	1.69
4	FR	2024-01-01	Still Here Season 2024 Cinematic - League of Legends (ft. Forts, Tiffany Aris, and 2W	League of Legends	Gaming	104159411	1.69
5	GB	2024-01-01	Still Here Season 2024 Cinematic - League of Legends (ft. Forts, Tiffany Aris, and 2W	League of Legends	Gaming	104159411	1.69
6	IN	2024-01-01	Protect \$500,000 Keep It!	MrBeast	Entertainment	85458562	4.21
7	JP	2024-01-01	Survive 100 Days Trapped, Win \$500,000	MrBeast	Entertainment	137639799	3.22
8	KR	2024-01-01	Survive 100 Days Trapped, Win \$500,000	MrBeast	Entertainment	143955997	3.16
9	MX	2024-01-01	Survive 100 Days Trapped, Win \$500,000	MrBeast	Entertainment	137639799	3.22
10	US	2024-01-01	Grand Theft Auto VI Trailer 1	Rockstar Games	Gaming	166323421	6.73
11	BR	2024-02-01	Face Your Biggest Fear To Win \$800,000	MrBeast	Entertainment	126846652	3.54
12	CA	2024-02-01	Face Your Biggest Fear To Win \$800,000	MrBeast	Entertainment	119170728	3.66
13	DE	2024-02-01	Face Your Biggest Fear To Win \$800,000	MrBeast	Entertainment	114978689	3.72

4. For Each Country, Which category_title Has the Most Distinct Videos Before 2022?

```

-- 4. For each country, which category_title has the most distinct videos and what is its percentage (12 decimals) out of the total distinct number
of videos of that country? Only look at the data from before 2022. Order the result by category_title and country

WITH VideoCategories AS (
    SELECT
        country,
        category_title,
        COUNT(DISTINCT video_id) AS total_category_video
    FROM
        table_youtube_final
    WHERE
        EXTRACT(YEAR FROM trending_date) >= 2022
    GROUP BY
        country,
        category_title
),
TotalCountryVideos AS (
    SELECT
        country,
        COUNT(DISTINCT video_id) AS total_country_video
    FROM
        table_youtube_final
    WHERE
        EXTRACT(YEAR FROM trending_date) >= 2022
    GROUP BY
        country
)

SELECT
    vc.country AS COUNTRY,
    vc.category_title AS CATEGORY_TITLE,
    vc.total_category_video AS TOTAL_CATEGORY_VIDEO,
    tc.total_country_video AS TOTAL_COUNTRY_VIDEO,
    ROUND((vc.total_category_video::NUMERIC / tc.total_country_video) * 100, 2) AS PERCENTAGE
FROM
    VideoCategories vc
JOIN
    TotalCountryVideos tc
ON
    vc.country = tc.country
WHERE
    (vc.country, vc.total_category_video) IN (
        SELECT
            country,
            MAX(total_category_video)
        FROM
            VideoCategories
        GROUP BY
            country
    )
ORDER BY
    vc.category_title,
    vc.country;

--5. Which channeltitle has produced the most distinct videos and what is this number?
WITH ChannelVideoCounts AS (
    SELECT
        channel_title,
        COUNT(DISTINCT video_id) AS distinct_video_count
    FROM
        table_youtube_final
    GROUP BY
        channel_title
)

SELECT
    channel_title,
    distinct_video_count
FROM
    ChannelVideoCounts
WHERE
    distinct_video_count = (SELECT MAX(distinct_video_count) FROM ChannelVideoCounts);

```

This query identifies the category_title with the most distinct videos for each country using data from before 2022. It calculates the number of distinct videos per category and country, then computes the percentage of each category's videos out of the total videos for that country. This percentage is truncated to 12 decimal places. The results help understand content diversity and dominant categories in different countries.

	COUNTRY	CATEGORY_TITLE	TOTAL_CATEGORY_VIDEO	TOTAL_COUNTRY_VIDEO	PERCENTAGE
1	BR	Entertainment	5417	23760	22.80
2	DE	Entertainment	7709	30719	25.10
3	FR	Entertainment	7548	32849	22.98
4	GB	Entertainment	5643	27855	20.26
5	IN	Entertainment	21281	50250	42.35
6	JP	Entertainment	5658	17627	32.10
7	KR	Entertainment	5122	15175	33.75
8	MX	Entertainment	4195	17532	23.93
9	CA	Gaming	6594	30869	21.36
10	US	Gaming	6226	28799	21.62

5. Which channel_title Has Produced the Most Distinct Videos?

```
--5. Which channeltitle has produced the most distinct videos and what is this number?
WITH ChannelVideoCounts AS (
  SELECT
    channel_title,
    COUNT(DISTINCT video_id) AS distinct_video_count
  FROM
    table_youtube_final
  GROUP BY
    channel_title
)
SELECT
  channel_title,
  distinct_video_count
FROM
  ChannelVideoCounts
WHERE
  distinct_video_count = (SELECT MAX(distinct_video_count) FROM ChannelVideoCounts);
```

The query finds the channel_title that has produced the most distinct videos across the entire dataset. It counts the number of unique videos per channel and then selects the channel with the maximum count. This analysis helps identify the most prolific content creators on YouTube, indicating channels with broad content production.

	CHANNEL_TITLE	DISTINCT_VIDEO_COUNT
1	Vijay Television	2049

8. BUSINESS QUESTION:

The query first creates a common table expression (CTE) named VideoStats to rank videos by view_count for each country, year, and month. It excludes videos in the "Music" and "

```
-- THE GOAL IS TO FIND OUT WHICH CATEGORIES FREQUENTLY HAVE TRENDING VIDEOS. THEN IT WILL EASY FOR ME TO FIGURE OUT WHICH CATEGORY VIDEO I CAN MAKE
THAT WILL APPEAR ON TRENDING LIST OT NOT
WITH VideoStats AS (
  SELECT
    country,
    category_title,
    RANK() OVER (PARTITION BY country, EXTRACT(YEAR FROM trending_date), EXTRACT(MONTH FROM trending_date) ORDER BY view_count DESC) AS
TRENDING_CT_RK
  FROM
    table_youtube_final
  WHERE NOT
    category_title IN ('Music', 'Entertainment') -- (EXCLUDING MUSIC AND ENTERTAINMENT)
)
SELECT
  country,
  category_title,
  COUNT(*) AS TRENDING_CT_RK
FROM
  VideoStats
WHERE
  TRENDING_CT_RK = 1
GROUP BY
  country,
  category_title
ORDER BY
  country,
  TRENDING_CT_RK DESC;

-- ACCORTING TO THE DATA, I WILL CREATE A YOUTUBE CAHNNEL AND MAKE A VIDEO IN GAMING CATEGORY. IT HAS A HIGH CHANCE OF APPEAR IN TOP TREND IN YOUTUBE.
```

Entertainment" categories to focus on other content types. The RANK() window function is applied to assign a rank to each video based on its view count within its respective country and time frame.

In the main query, only videos ranked first (TRENDING_CT_RK = 1) are considered, representing the top trending videos for each country and period. The results are then grouped by country and category_title, and ordered by the number of times a category has reached the top trending spot (TRENDING_CT_RK).

Insights and Strategy

Based on the analysis, the "Gaming" category frequently appears in the top trending videos in the US, making it a strong candidate for a new YouTube channel. This suggests that focusing on creating gaming content could increase the likelihood of trending. However, this strategy may not work universally in every country, as cultural differences and content preferences vary. For example, other countries might have different popular categories, such as "Sports" or "News." Therefore, while "Gaming" is a strong contender in the US, localized content strategies should be developed for success in other regions.

PART_4SQL										
	BR	CA	DE	FR	GB	IN	JP	KR	MX	US
IR	45									
CA		45								
DE			45							
FR				45						
GB					45					
IN						45				
JP							45			
KR								45		
MX									45	
US										45

	COUNTRY	CATEGORY_TITLE	TRENDING_CT_RK
1	BR	Gaming	12
2	BR	Science & Technology	10
3	BR	People & Blogs	8
4	BR	Sports	8
5	BR	Film & Animation	2
6	BR	Comedy	1
7	BR	Autos & Vehicles	1
8	BR	Education	1
9	BR	Travel & Events	1
10	BR	Howto & Style	1
11	CA	Science & Technology	10
12	CA	Gaming	7
13	CA	People & Blogs	7

8.CONCLUION:

In conclusion, this project effectively utilized Snowflake's Data Lakehouse architecture to analyze YouTube trending and category data across various countries, providing a comprehensive understanding of global content consumption patterns. By carefully ingesting, transforming, and cleaning the data, we ensured its integrity and accuracy, enabling robust analysis. The insights gained from this project reveal significant regional differences in content preferences and highlight the potential for optimizing content strategies on YouTube. Focusing on high-performing categories, such as Gaming or Sports (excluding Music and Entertainment), can increase the likelihood of creating trending videos. Overall, this analysis underscores the importance of data-driven decision-making in digital content creation and strategy development.