Project Title: YouTube Trending Video Analysis Using MySQL

## Objective:

The objective of this project is to clean and analyze YouTube trending video data from India using MySQL. The key goals include:

- Identifying and removing bad/missing/duplicate data.
- Creating formatted and meaningful columns.
- Performing business-focused queries such as top channels, engagement analysis, and category-wise insights.

### **Tools Used:**

- MySQL Workbench & MySQL CLI
- Kaggle Dataset (INvideos.csv & IN\_category\_id.json)
- Manual Category Mapping from JSON

## **Dataset Overview**

- Source: Kaggle YouTube Trending Dataset
- Files Used: INvideos.csv, IN\_category\_id.json
- Original Rows: 37346
- **Columns:** 16
- Key columns: video\_id, title, channel\_title, views, likes, dislikes, comment\_count, publish\_time, tags, category\_id

# **Data Import Steps**

CREATE DATABASE YouTubeAnalytics;

USE YouTubeAnalytics;

CREATE TABLE INvideos (

```
video_id VARCHAR(20),
  trending_date DATE,
  title TEXT,
  channel_title VARCHAR(100),
  category_id INT,
  publish_time DATETIME,
  tags TEXT,
  views BIGINT,
  likes BIGINT,
  dislikes BIGINT,
  comment_count BIGINT,
  thumbnail_link TEXT,
  comments_disabled BOOLEAN,
  ratings_disabled BOOLEAN,
  video_error_or_removed BOOLEAN,
  description TEXT
);
LOAD DATA LOCAL INFILE 'E:/Youtube_Analytics/data/INvideos_utf8.csv'
INTO TABLE INvideos
CHARACTER SET utf8mb4
FIELDS TERMINATED BY ','
ENCLOSED BY ""
LINES TERMINATED BY '\r\n'
IGNORE 1 ROWS;
```

## **Data Cleaning Steps**

#### -- Date Conversion

```
ALTER TABLE INvideos ADD formatted_trending_date DATE;
```

**UPDATE INvideos** 

SET formatted\_trending\_date = STR\_TO\_DATE(trending\_date, '\%y.\%d.\%m')

WHERE trending\_date REGEXP '^[0-9]{2}\\.[0-9]{2}\\.[0-9]{2}\\;

### -- Publish Time Conversion

ALTER TABLE INvideos ADD formatted\_publish\_time DATETIME;

**UPDATE INvideos** 

SET formatted\_publish\_time = STR\_TO\_DATE(SUBSTRING\_INDEX(publish\_time, '.', 1), '%Y-%m-%dT%H:%i:%s')

WHERE publish\_time LIKE '%T%';

### -- Convert Boolean Texts

```
ALTER TABLE INvideos
```

```
ADD is_comments_disabled BOOLEAN,
```

ADD is\_ratings\_disabled BOOLEAN,

ADD is\_video\_error\_removed BOOLEAN;

```
UPDATE INvideos SET is_comments_disabled = (comments_disabled = 'TRUE');
```

UPDATE INvideos SET is\_ratings\_disabled = (ratings\_disabled = 'TRUE');

UPDATE INvideos SET is\_video\_error\_removed = (video\_error\_or\_removed = 'TRUE');

```
CREATE TABLE INcategories (
category_id INT PRIMARY KEY,
category_name VARCHAR(100)
```

);

### **INSERT INTO INcategories VALUES**

- (1, 'Film & Animation'), (2, 'Autos & Vehicles'), (10, 'Music'),
- (15, 'Pets & Animals'), (17, 'Sports'), (19, 'Travel & Events'),
- (20, 'Gaming'), (22, 'People & Blogs'), (23, 'Comedy'),
- (24, 'Entertainment'), (25, 'News & Politics'), (26, 'Howto & Style'),
- (27, 'Education'), (28, 'Science & Technology'), (29, 'Nonprofits & Activism');

### -- Index for Speed

CREATE INDEX idx\_video\_date ON INvideos(video\_id, formatted\_trending\_date);

### -- Initial Row Count

SELECT COUNT(\*) FROM INvideos;



### -- Check NULL / Empty

SELECT COUNT(\*) FROM INvideos WHERE description IS NULL OR description = ";



SELECT COUNT(\*) FROM INvideos WHERE tags IS NULL OR tags = ";



SELECT COUNT(\*) FROM INvideos WHERE title IS NULL OR title = ";



## -- Fill Missing Text

UPDATE INvideos SET description = 'No Description' WHERE description IS NULL OR description = ";

UPDATE INvideos SET tags = 'No Tags' WHERE tags IS NULL OR tags = ";

UPDATE INvideos SET title = 'No Title' WHERE title IS NULL OR title = ";

### -- Remove Invalid Rows

SELECT COUNT(\*) FROM INvideos

WHERE views =  $0 \text{ OR likes} = 0 \text{ OR dislikes} = 0 \text{ OR comment\_count} = 0$ ;



**DELETE FROM INvideos** 

WHERE views = 0 AND likes = 0 AND comment\_count = 0;

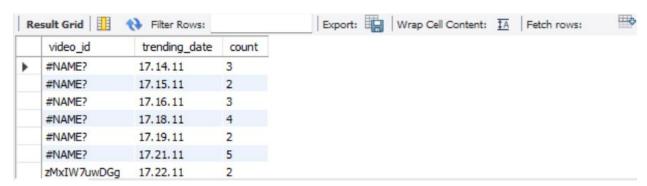
## -- Check Duplicates

SELECT video\_id, trending\_date, COUNT(\*) AS count

FROM INvideos

GROUP BY video\_id, trending\_date

HAVING count > 1;



## -- Deduplication

```
CREATE TABLE INvideos_deduped AS

SELECT * FROM INvideos

WHERE (video_id, trending_date) IN (

SELECT video_id, MIN(trending_date)

FROM INvideos

WHERE trending_date > '1000-01-01'

GROUP BY video_id, trending_date
```

DROP TABLE INvideos;

);

RENAME TABLE INvideos\_deduped TO INvideos;

### -- Final Cleaned Row Count

SELECT COUNT(\*) AS final\_cleaned\_rows FROM INvideos;



#### -- Most Viewed Videos

SELECT title, channel\_title, views

FROM INvideos

ORDER BY views DESC

LIMIT 10;



#### -- Most Liked

SELECT title, likes, views

FROM INvideos

ORDER BY likes DESC

LIMIT 10;



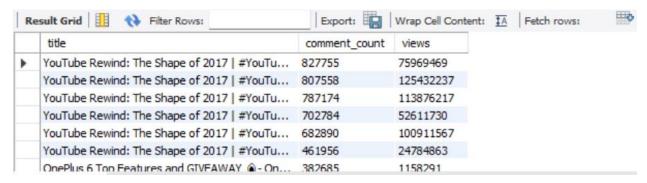
#### -- Most Commented

SELECT title, comment\_count, views

FROM INvideos

ORDER BY comment\_count DESC

LIMIT 10;



## -- Top Channels by Views

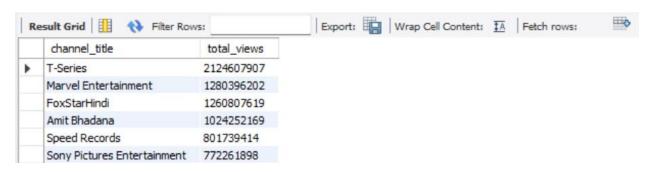
SELECT channel\_title, SUM(views) AS total\_views

FROM INvideos

GROUP BY channel\_title

ORDER BY total\_views DESC

LIMIT 10;



### -- Engagement Ratio

SELECT title, ROUND((likes / views) \* 100, 2) AS like\_ratio, views, likes

FROM INvideos

WHERE views > 100000

ORDER BY like\_ratio DESC

LIMIT 10;



### -- Average Views by Category Name

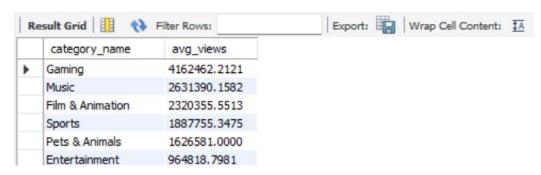
SELECT c.category\_name, AVG(v.views) AS avg\_views

FROM INvideos v

JOIN INcategories c ON v.category\_id = c.category\_id

GROUP BY c.category\_name

ORDER BY avg\_views DESC;



## -- Category ID-wise Views

SELECT category\_id, AVG(views) AS avg\_views

FROM INvideos

GROUP BY category\_id

ORDER BY avg\_views DESC;



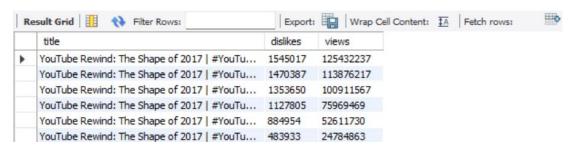
### -- Most Disliked

SELECT title, dislikes, views

FROM INvideos

ORDER BY dislikes DESC

LIMIT 10;



## -- Channels with Most Trending Videos

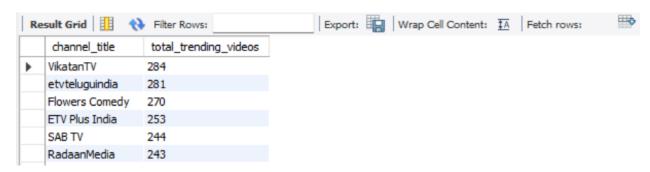
SELECT channel\_title, COUNT(\*) AS total\_trending\_videos

FROM INvideos

GROUP BY channel\_title

ORDER BY total\_trending\_videos DESC

LIMIT 10;

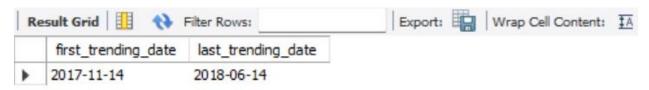


## -- Date Range

SELECT MIN(formatted\_trending\_date) AS first\_trending\_date,

MAX(formatted\_trending\_date) AS last\_trending\_date

## FROM INvideos;



## -- Monthly Trend

SELECT MONTH(formatted\_trending\_date) AS month, COUNT(\*) AS total\_trending

FROM INvideos

GROUP BY month

ORDER BY month;



### -- Upload Timing Trends

SELECT HOUR(formatted\_publish\_time) AS hour, COUNT(\*) AS uploads

FROM INvideos

GROUP BY hour

ORDER BY uploads DESC;

R	esult Gri	d 📗 🙌	Filter Rows:	Export:	Wrap Cell Content:	<u>‡A</u>
	hour	uploads				
•	14	2837				
	12	2782				
	13	2631				
	6	2322				
	11	2295				
	5	2184				

## Summary

- Original Rows: 37346
- Final Cleaned Rows: ~35,598 (after null, invalid and duplicate removals)
- Most Viewed Videos in India: YouTube Rewind: The Shape of 2017
- Top Channels by Total Views: T-Series views -416339525
- Most Liked Videos in India: YouTube Rewind: The Shape of 2017, likes 2912710
- Engagement Rate: Likes vs. Views: OnePlus 6 Top Features and GIVEAWAY, views 1049339, likes 355742
- Average Views by Category Name: Cate\_name Gaming, Avg\_views 36270875385
- Category-wise Average Views: Cate\_id 20, Avg\_views 36270875385
- Videos That Got Disliked Most: YouTube Rewind: The Shape of 2017, Dislike 1545017
- Top Channels by Number of Trending Videos: Channel etvteluguindia, video\_num 66
- Videos with Most Comments: YouTube Rewind: The Shape of 2017, Comments 807558
- First and Last Trending Dates: first 2017-01-12, last 2018 -12-06
- Monthly Trend of Trending Videos: (jan)- 471, (feb)- 438

# Conclusion

This project helped in understanding real-world data cleaning, MySQL commands, and analysis using structured query language. It demonstrated:

- Data wrangling capability
- Error resolution (e.g. file import, date parsing)
- Business query framing skills