Minor Project — Data Set Options & Table Matrix (Final Submission)

Title: Data Set Options and Table Matrix — YouTube Trending Videos
Author: Syed Ahmed Ali
Program: Intrainz Edutech — Data Analyst Program
Submitted on: October, 2025

Abstract

This report presents the complete design and implementation plan for structuring the dataset used in the *YouTube Trending Video Analytics* project. The dataset, containing **32,458 snapshot rows**, captures trending video data from multiple countries, representing **16,307 unique videos**, **1,425 unique channels**, and **17 distinct categories**. The project objective is to transform the raw CSV dataset (all_trending_videos.csv) into a normalized relational model suitable for analytics and reporting. This report explains dataset characteristics, storage options, normalization principles, SQL schema design, table matrix mapping, and key analytical queries. Using a relational database system (MySQL), the report demonstrates how redundant data can be reduced while enabling efficient analytical queries for trends, category insights, and engagement metrics.

Dataset Overview

- **Source file:** all_trending_videos.csv (from GitHub repository).
- **Rows (snapshots):** 32,458
- **Columns:** video_id, trending_date, title, channel_title, category_id, publish_time, tags, views, likes, dislikes, comment_count, thumbnail_link, comments_disabled, ratings_disabled, video_error_or_removed, description, country
- **Unique videos:** 16,307
- **Unique channels:** 1,425
- **Unique categories:** 17
- **Missing values:** description has 526 missing entries; all other fields are complete.

Dataset Characteristics:

- Each row represents a trending snapshot for a video_id on a specific trending_date.
- Metadata such as title, channel_title, and publish_time are repeated across multiple rows for the same video.
- tags are stored as a pipe-separated string (e.g., tag1|tag2|tag3).

The dataset structure indicates redundancy and motivates the need for a normalized schema to eliminate duplication while maintaining referential integrity.

Data Set Options and Recommendation

Option 1 — Flat CSV

- **Advantages:** Easy to handle, requires no database.
- **Disadvantages:** High redundancy, slow for aggregation queries, and limited flexibility for analytics.

Option 2 — Relational Database (MySQL)

- **Advantages:** Enforces structure, supports queries, indexing, and referential integrity. Ideal for analytical queries.
- **Disadvantages:** Requires initial setup and data import.

Option 3 — Data Warehouse (Parquet/BigQuery)

- **Advantages:** Scalable for very large datasets; columnar compression improves performance.
- **Disadvantages:** Overhead for setup; unnecessary for current dataset size.

Recommendation: Use **MySQL** for the Minor Project, as it balances structure, query power, and simplicity for ~32k records.

Proposed Relational Model (ER) & SQL Schema

Entity Relationships:

- **Channels → Videos:** One channel uploads many videos.
- **Videos → Video Stats:** One video has many trending snapshots.
- **Videos → Tags:** Many-to-many relationship through video_tags.
- **Videos → Categories:** One category can contain many videos.

Normalized Tables:

- channels: Stores unique uploader details.
- categories: Stores category IDs and names.
- videos: Holds metadata for each unique video.
- video_stats: Tracks daily trending stats for each video.
- tags: Stores all unique tag strings.
- video_tags: Junction table linking videos and tags.

SQL Schema (MySQL)

```
CREATE TABLE channels (
  channel_id INT PRIMARY KEY,
  channel_title VARCHAR(255) UNIQUE,
  channel_url VARCHAR(512) NULL
);

CREATE TABLE categories (
  category_id INT PRIMARY KEY,
  category_name VARCHAR(255) NULL
);

CREATE TABLE videos (
  video_id VARCHAR(64) PRIMARY KEY,
  channel_id INT,
  title TEXT,
  publish_time DATETIME NULL,
  category_id INT NULL,
  thumbnail_link VARCHAR(512),
  description TEXT,
  country VARCHAR(100),
  FOREIGN KEY (channel_id) REFERENCES channels(channel_id),
  FOREIGN KEY (category_id) REFERENCES categories(category_id)
);

CREATE TABLE video_stats (
  id INT AUTO_INCREMENT PRIMARY KEY,
  video_id VARCHAR(64),
  trending_date DATE,
  views BIGINT,
  likes BIGINT,
```

```
  dislikes BIGINT,
  comment_count BIGINT,
  comments_disabled TINYINT(1),
  ratings_disabled TINYINT(1),
  video_error_or_removed TINYINT(1),
  FOREIGN KEY (video_id) REFERENCES videos(video_id)
);

CREATE TABLE tags (
  tag_id INT PRIMARY KEY,
  tag_text VARCHAR(255) UNIQUE
);

CREATE TABLE video_tags (
  video_id VARCHAR(64),
  tag_id INT,
  PRIMARY KEY (video_id, tag_id),
  FOREIGN KEY (video_id) REFERENCES videos(video_id),
  FOREIGN KEY (tag_id) REFERENCES tags(tag_id)
);
```

Table Matrix (Field → Table Mapping)

Raw Field	Destination Table	Column Name	SQL Type	Description
video_id	videos	video_id	VARCHAR(64)	Unique identifier for each video
title	videos	title	TEXT	Video title
channel_title	channels	channel_title	VARCHAR(255)	Name of channel
category_id	categories	category_id	INT	Foreign key to categories table

Raw Field	Destination Table	Column Name	SQL Type	Description
publish_time	videos	publish_time	DATETIME	Video publish timestamp
trending_date	video_stats	trending_date	DATE	Trending snapshot date
views	video_stats	views	BIGINT	Number of views
likes	video_stats	likes	BIGINT	Number of likes
dislikes	video_stats	dislikes	BIGINT	Number of dislikes
comment_count	video_stats	comment_count	BIGINT	Total comments
tags	tags / video_tags	tag_text	VARCHAR(255)	Keywords associated with the video
thumbnail_link	videos	thumbnail_link	VARCHAR(512)	Thumbnail URL
comments_disabled	video_stats	comments_disabled	TINYINT(1)	Boolean flag
ratings_disabled	video_stats	ratings_disabled	TINYINT(1)	Boolean flag
video_error_or_removed	video_stats	video_error_or_removed	TINYINT(1)	Boolean flag
description	videos	description	TEXT	Video description
country	videos	country	VARCHAR(100)	Country of origin

Sample Rows

video_id	trending_date	title	channel_title	views	likes	dislikes	comment_count	country
kzwfHumJyYc	2017-11-14	Sharry Mann: Cute Munda	Lokdhun Punjabi	1,096,327	33,966	798	882	IN
zUZ1z7FwLc8	2017-11-14	पीरियड्स के समय...	HJ NEWS	590,101	735	904	0	IN
10L1hZ9qa58	2017-11-14	Stylish Star Allu Arjun	TFPC	473,988	2,011	243	149	IN

video_id	trending_date	title	channel_title	views	likes	dislikes	comment_count	country
N1vE8iiEg64	2017-11-14	@ ChaySam Eruma Saani	Tamil vs English	1,242,680	70,353	1,624	2,684	IN
kJzGH0PVQHQ	2017-11-14	Why Samantha became Emotional...	Filmylooks	464,015	492	293	66	IN

Sample Analytical Queries & Results

1. Top Categories by Snapshots

SELECT category_id, COUNT(*) AS snapshot_count

FROM videos v JOIN video_stats vs ON v.video_id = vs.video_id

GROUP BY category_id

ORDER BY snapshot_count DESC LIMIT 10;

Result: Category 24 dominates with 14,511 snapshots, followed by 25 (4,649) and 10 (3,171).

2. Average Views by Publish Hour

SELECT HOUR(publish_time) AS publish_hour, COUNT(*) AS cnt

FROM videos

WHERE publish_time IS NOT NULL

GROUP BY publish_hour

ORDER BY cnt DESC LIMIT 10;

Result: Most videos were published around 12–14 hours of the day (UTC), matching global peak hours.

3. Longest-Trending Videos

SELECT v.video_id, v.title, COUNT(DISTINCT vs.trending_date) AS trending_days

FROM video_stats vs JOIN videos v ON vs.video_id = v.video_id

GROUP BY v.video_id, v.title

ORDER BY trending_days DESC LIMIT 10;

Result: Shows top videos trending for the longest duration, highlighting persistent popularity.

4. Top Channels by Number of Trending Videos

```
SELECT c.channel_title, COUNT(DISTINCT v.video_id) AS num_videos
FROM videos v JOIN channels c ON v.channel_id = c.channel_id
GROUP BY c.channel_title
ORDER BY num_videos DESC LIMIT 10;
```

Result: Identifies highly consistent content creators with frequent trending videos.

Conclusion

This Minor Project demonstrates the transformation of a raw dataset into a clean, normalized database suitable for analytics. The relational model supports efficient querying, eliminates redundancy, and aligns perfectly with the analytical needs of the YouTube Trending Video Analytics project. Through normalization, data integrity and scalability are achieved, forming a strong foundation for the Major Project (Charts Report).