Title: YouTube Data Ecosystem.

Scope & Objective: We'll design a comprehensive RDMS to model YouTube's data intricacies and manage retrieval efficiently, followed by translating it to a Cassandra database with adjustments for large-scale handling and a query-first design. A performance analysis will then highlight the differences and advantages between the two databases in terms of efficiency, scalability, and maintenance.

Translation process: Given Cassandra's architecture, our schema design will be guided by our anticipated query patterns and specific use cases. This approach, which diverges from traditional normalization techniques found in relational databases, is aimed at optimizing our database for high-speed reads and writes at scale. It aligns with our commitment to a "Query First" methodology in database design.

Data Reference:
We are using a combination of some dummy data and some original Youtube data from below source:
- https://data.world/popculture/donald-glovers-this-is-america-youtube-comments

The data we have outlined will primarily serve as the foundation upon which we will build and populate our relational database schema. In an effort to closely mirror a real-world application, we have decided to draw inspiration from the YouTube data ecosystem. This will involve using a mix of dummy data for general table structure, such as Videos and Users, as well as incorporating actual data excerpts, such as comments from YouTube videos, to add authenticity to our database. Where necessary, we intend to augment our data set by leveraging the YouTube Public APIs to source additional content.

Use Cases we are considering while designing Cassandra schema:
- Listing videos uploaded by a specific user
- Fetching comments for a specific video
- Retrieving comments made by a specific user

Attached: YouTube Data Ecosystem ERD

YouTube Data Ecosystem

videolikes	
PK	user_id INT NOT NULL
PK	video_id INT NOT NULL
PK	like_id SERIAL
	like_date TIMESTAMP NOT NULL
	FK (video_id)
	FK (user_id)

videocategories	
PK	category_id INT NOT NULL
PK	video_id INT NOT NULL
	FOREIGN KEY (video_id)
	FOREIGN KEY (category_id)

users	
PK	user_id SERIAL
	username TEXT NOT NULL
	email TEXT NOT NULL
	password_hash TEXT NOT NULL
	join_date TIMESTAMP NOT NULL
	last_login TIMESTAMP

videos	
PK	user_id INT NOT NULL
PK	video_id SERIAL
	title TEXT NOT NULL
	description TEXT
	upload_date TIMESTAMP NOT NULL
	duration INT NOT NULL
	views INT DEFAULT 0
	status VARCHAR(50)
	FK (user_id)

categories	
PK	category_id SERIAL
	category_name TEXT NOT NULL UNIQUE

comments	
PK	user_id INT NOT NULL
PK	video_id INT NOT NULL
PK	comment_id SERIAL
	comment_text TEXT NOT NULL
	comment_date TIMESTAMP NOT NULL
	FK (video_id)
	FK (user_id)

videoviews	
PK	user_id INT
PK	video_id INT NOT NULL
PK	view_id SERIAL
	view_date TIMESTAMP NOT NULL
	FOREIGN KEY (video_id)
	FOREIGN KEY (user_id)

subscriptions	
PK	subscription_id SERIAL
	subscriber_id INT NOT NULL
	subscribed_to_id INT NOT NULL
	subscription_date TIMESTAMP WITH TIME ZONE N
	FOREIGN KEY (subscriber_id) REFERENCES users
	FOREIGN KEY (subscribed_to_id) REFERENCES us