TITLE : YOUTUBE ON DATA HARVESTING AND WAREHOUSING USING SQL AND STREAMLIT

INTRODUCTION

In the digital age, understanding and analyzing online video content is crucial for marketers, content creators, and researchers. YouTube, being the largest video-sharing platform, offers a wealth of data that can be harnessed for various insights. This report details a project aimed at harvesting YouTube data, warehousing it using SQL databases, and visualizing it through an interactive Streamlit application.

**2. Project Objectives**

The primary objectives of this project are:

* **Data Harvesting:** Collect detailed metrics and information from YouTube, including video statistics, channel data, and user comments.
* **Data Warehousing:** Store the harvested data efficiently in a structured SQL database.
* **Data Visualization:** Develop an interactive dashboard using Streamlit to enable users to explore and analyze the data.

**3. Methodology**

**3.1 Data Harvesting**

**Tools and Technologies:**

* **YouTube Data API:** Utilized to access YouTube data programmatically.
* **Python Libraries:** google-api-python-client for API interactions and pandas for data manipulation.

**Process:**

1. **API Setup:**
   * Registered for API access via the Google Developer Console.
   * Obtained API keys for authentication.
2. **Data Extraction:**
   * Developed Python scripts to query the YouTube Data API.
   * Collected data such as video titles, descriptions, view counts, likes, comments, and channel statistics.
3. **Data Cleaning:**
   * Preprocessed data to address issues such as missing values, duplicates, and inconsistencies.

**3.2 Data Warehousing**

**Tools and Technologies:**

* **SQL Database:** Chose PostgreSQL for its robustness and scalability.
* **SQLAlchemy:** Employed for database interaction and ORM functionality.

**Process:**

1. **Database Design:**
   * Designed a relational schema with tables for videos, channels, comments, and statistics.
2. **Data Insertion:**
   * Created Python scripts to insert cleaned data into the database.
   * Used SQLAlchemy to manage database connections and transactions.
3. **Data Maintenance:**
   * Implemented procedures for updating records and managing new data.

**3.3 Data Visualization**

**Tools and Technologies:**

* **Streamlit:** For building interactive web applications.
* **Visualization Libraries:** matplotlib, seaborn, and plotly for generating charts and graphs.

**Process:**

1. **Streamlit App Development:**
   * Developed a Streamlit application that connects to the PostgreSQL database.
   * Implemented various widgets to allow users to filter and explore data.
2. **UI Components:**
   * Designed interactive elements such as sliders, dropdowns, and buttons.
   * Enabled users to view metrics like subscriber growth, view counts, and engagement rates.
3. **Visualization:**
   * Created visualizations including time series plots, bar charts, and heatmaps.
   * Enabled dynamic updates based on user inputs to facilitate real-time analysis.
4. **Deployment:**
   * Deployed the Streamlit app on a cloud service for accessibility.

**4. Use Cases**

* **Channel Performance Analysis:**
  + Analyze trends in subscriber growth, view counts, and engagement metrics.
  + Identify peak performance periods and evaluate the impact of content strategies.
* **Content Analysis:**
  + Examine the effectiveness of various types of content
  + Analyze audience feedback and sentiment from comments.
* **Comparative Analysis:**
  + Compare metrics across multiple channels or videos.
  + Identify best practices and benchmark against competitors.

**5. Challenges and Considerations**

* **API Quota Limits:**
  + Addressed potential limitations by optimizing API queries and managing usage efficiently.
* **Data Privacy:**
  + Ensured compliance with YouTube’s data policies and relevant privacy regulations.
* **Scalability:**
  + Designed the database schema and Streamlit application to handle large datasets and multiple concurrent users.

**6. Conclusion**

This project effectively demonstrates the integration of data harvesting, warehousing, and visualization techniques to analyze YouTube data. By leveraging SQL databases for structured storage and Streamlit for interactive visualizations, users can gain meaningful insights into video content and channel performance. The solutions provided are scalable, efficient, and tailored to meet the needs of diverse users seeking to understand YouTube metrics.

NEHASRI.R