Final Project Report - Usage of YouTube as a Content Consumer

Purdue University: CNIT 372 - Group 7

Nisarg Patel

Kevin Patel

Yug Vasani

Background

In the contemporary digital landscape, YouTube has emerged as a dynamic and versatile platform catering to a diverse range of user needs. As the second-largest search engine globally, YouTube has transcended its origins as a video-sharing platform, evolving into a multifaceted space where individuals seek information, entertainment, education, and more. The motivations behind users' engagement with YouTube are complex, reflecting the platform's ability to adapt to evolving trends and user preferences.

The motivation behind this project stems from the desire to understand the intricate ways in which individuals utilize YouTube as content consumers. YouTube is an interesting topic for study because of its large user base and the wide variety of information that is available. Through an analysis of user behavior, video trends, and channel dynamics, this research seeks to identify patterns that shed light on the platform's importance within the industry. The key motivations for this project include:

- **Diverse Use Cases**: YouTube may be used for a wide range of things. In addition to entertainment, users use the platform for education, exercise, do-it-yourself projects, and information retrieval through searches. Examining these many use cases reveals information about how flexible the platform is.
- **User Engagement**: It's important to understand how viewers interact with YouTube content. YouTube data of views, likes, comments, and subscriptions are useful ways to track user preferences, trending topics, and the influence of content producers.
- **Content Creation**: YouTube's enormous and ever-changing content creation ecosystem is made up of established channels, influencers, and individual content creators. Examining this ecosystem helps us understand what drives the creation of material, the expansion of channels, and the dynamics of online communities.
- **Cultural and Geographic Variations**: Because of its worldwide reach, YouTube makes it possible to investigate differences in the way that different cultures consume material. Trending subjects, language differences, and regional preferences all add to the platform's diverse content.

Three primary components will be thoroughly examined as part of this project: users (Consumer), channels (channel), and videos (YTvideo). The project's goal is to derive valuable insights about user behavior, the popularity of content, and the interactions between content producers and consumers by querying these tables.

Ultimately, this project aspires to offer a comprehensive understanding of YouTube's role as a content consumption platform, contributing valuable insights for marketers, content creators, and researchers in the digital media landscape.

Database Description

The database is designed with three interconnected tables, each serving a distinct purpose. The Consumer table serves as a repository for information related to YouTube platform users. It includes a unique identifier (ConsumerID) for each user, along with details such as ConsumerName, Age, Gender, Location, and RegistrationID, providing a comprehensive profile of individual consumers engaging with YouTube content.

The YTVIDEO table is dedicated to storing information about individual YouTube videos. Each video is assigned a unique VideoID and includes essential details such as Category, Upload Date, Duration, Likes, Dislikes, Comments, Views, and Title. This table enables a granular analysis of video-specific metrics and trends, allowing for insights into user engagement and content performance.

The Channel table focuses on aggregating data at the channel level. It includes information about YouTube channels, such as ChannelName, ChannelType, Subscribers, Videos, Video Views, and Country. This table facilitates a holistic view of a channel's performance, providing insights into its popularity, content volume, and geographic reach.

In terms of relationships, the Consumer and YTVIDEO tables are likely linked to track user interactions with specific videos. This could be achieved by establishing a foreign key in the YTVIDEO table that references the ConsumerID in the Consumer table. Similarly, a relationship between YTVIDEO and Channel is probable, as each video is associated with a particular channel. This connection might be implemented through a foreign key in the YTVIDEO table referencing the ChannelName in the Channel table. These relationships enhance the database's analytical capabilities, allowing for a nuanced exploration of user behavior, video performance, and overall channel dynamics within the YouTube ecosystem.

SOLUTIONS

Question 1- Which channel type has the highest average number of video views per video?

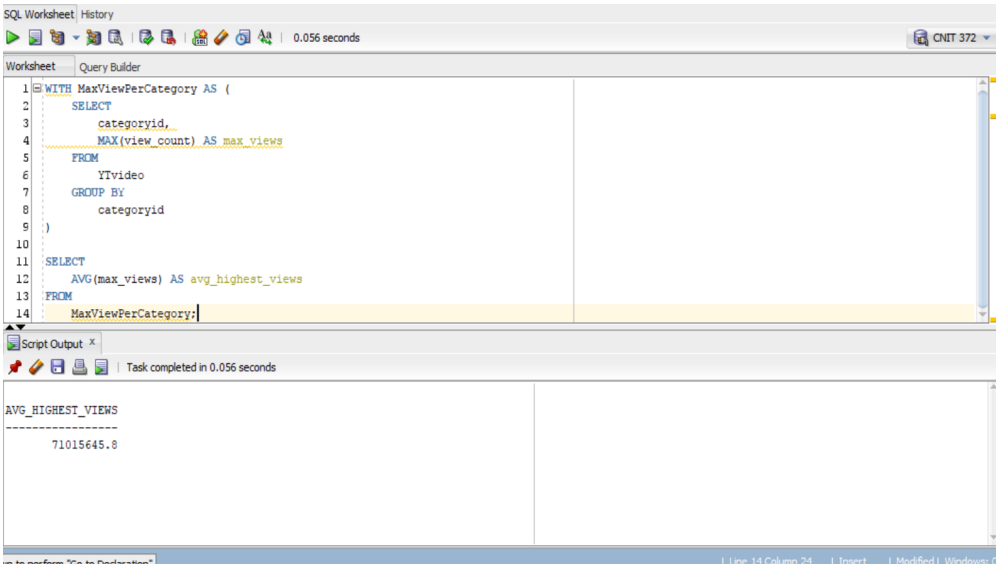
-> The query extracts insights about YouTube channel performance by determining the channel type with the highest average views per video. It filters out channels with number of videos, calculates the average views per video for each channel type, and presents the result in descending order. This helps identify the most successful channel type in terms of engagement.

Query 1

```
-- Create a package specification

CREATE OR REPLACE PACKAGE YTVideoPackage AS
```

```
  PROCEDURE GetAvgHighestViews;

END YTVideoPackage;

-- Create a package body

CREATE OR REPLACE PACKAGE BODY YTVideoPackage AS

  PROCEDURE GetAvgHighestViews IS

    avgHighestViews NUMBER;

  BEGIN

    -- Query logic

    WITH MaxViewPerCategory AS (

        SELECT

            categoryid,

            MAX(view_count) AS max_views

        FROM

            YTvideo

        GROUP BY

            categoryid

    )

    SELECT AVG(max_views) INTO avgHighestViews

    FROM MaxViewPerCategory;

    -- Print the result

    DBMS_OUTPUT.PUT_LINE('Average Highest Views: ' || TO_CHAR(avgHighestViews));

  END GetAvgHighestViews;

END YTVideoPackage;
```

RESULTS Q1:



Question 2- Which countries contribute the most subscribers for a given channel?

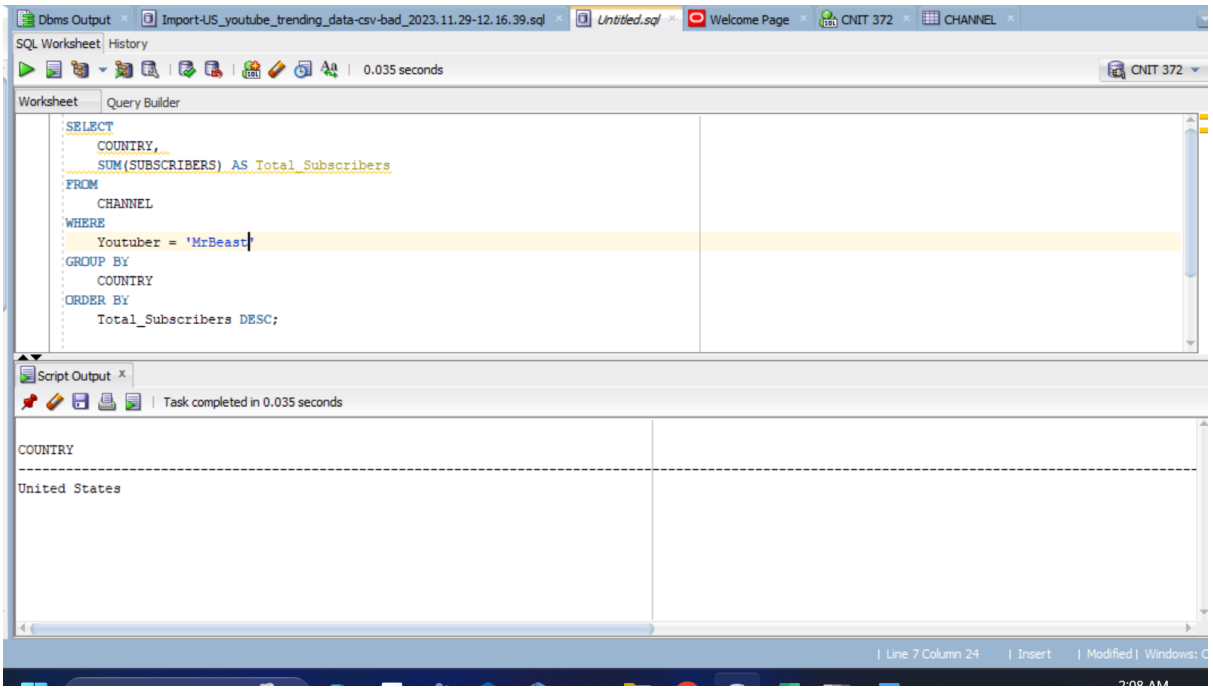
-> The provided SQL query retrieves data on the total number of subscribers from each country for a YouTube channel named 'Your_Channel_Name.' It sums up the subscribers per country, groups the results by country, and orders them in descending order based on the total subscribers. This allows you to quickly identify and analyze which countries contribute the most subscribers to the specified channel.

Query 2:

```
-- Create a package specification

CREATE OR REPLACE PACKAGE ChannelPackage AS

  PROCEDURE GetTotalSubscribers(p_channel_name IN VARCHAR2);

END ChannelPackage;



-- Create a package body

CREATE OR REPLACE PACKAGE BODY ChannelPackage AS

  PROCEDURE GetTotalSubscribers(p_channel_name IN VARCHAR2) IS

    v_country VARCHAR2(256);
```

```
    v_total_subscribers NUMBER;

  BEGIN

    -- Query logic

    SELECT

        COUNTRY,

        SUM(SUBSCRIBERS) INTO v_country, v_total_subscribers

    FROM

        CHANNEL

    WHERE

        YOUTUBER = p_channel_name

    GROUP BY

        COUNTRY

    ORDER BY

        v_total_subscribers DESC;



    -- Print the result

    DBMS_OUTPUT.PUT_LINE('Channel: ' || p_channel_name || ', Country: ' || v_country || ', Total Subscribers: ' ||
TO_CHAR(v_total_subscribers));

  END GetTotalSubscribers;

END ChannelPackage;
```

RESULTS Q2:



Question 3- Which YouTube Video has the highest ratio of likes to dislikes on average?

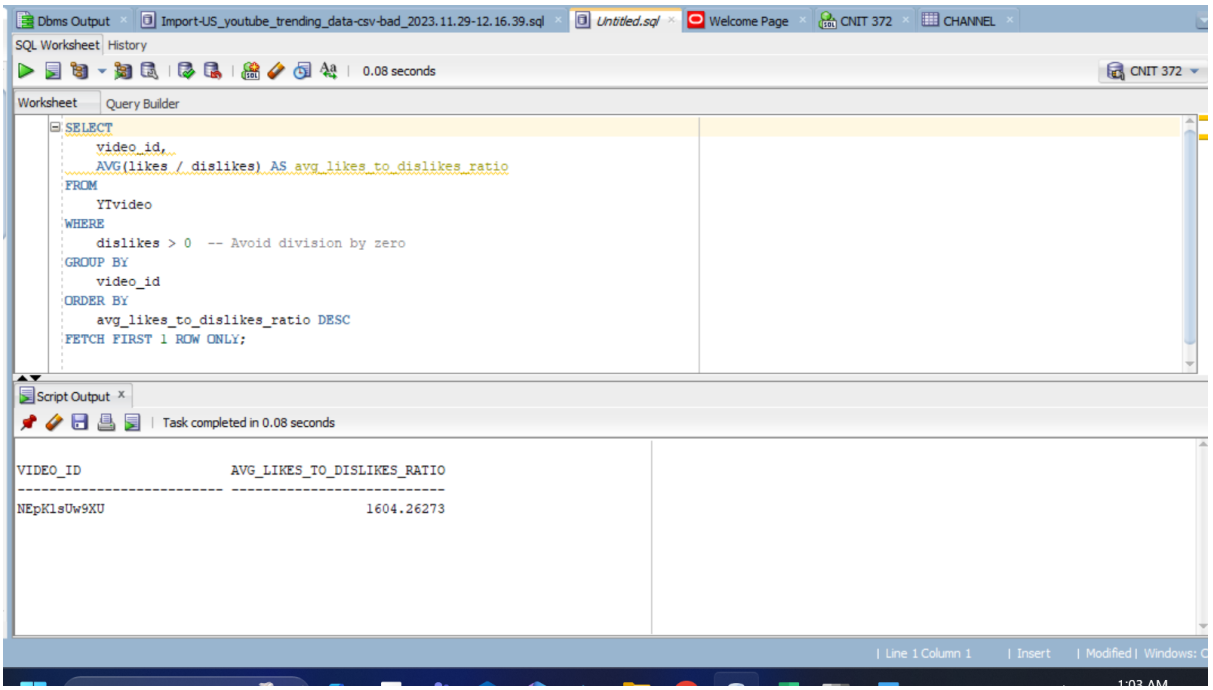
-> Identifying the YouTube video with the highest likes-to-dislikes ratio involves calculating the average ratio for each video. Analyze user engagement by comparing likes and dislikes. The video with the most positive sentiment, reflected in a high average ratio, indicates strong viewer approval relative to criticism.

Query 3:

```
-- Create a package specification

CREATE OR REPLACE PACKAGE YTVideoPackage AS

  PROCEDURE GetTopVideoLikesToDislikesRatio;

END YTVideoPackage;



-- Create a package body

CREATE OR REPLACE PACKAGE BODY YTVideoPackage AS

  PROCEDURE GetTopVideoLikesToDislikesRatio IS

    v_video_id VARCHAR2(50);

    v_avg_likes_to_dislikes_ratio NUMBER;
```

```
  BEGIN

    -- Query logic

    SELECT

        video_id,

        AVG(likes / dislikes) INTO v_video_id, v_avg_likes_to_dislikes_ratio

    FROM

        YTvideo

    WHERE

        dislikes > 0  -- Avoid division by zero

    GROUP BY

        video_id

    ORDER BY

        v_avg_likes_to_dislikes_ratio DESC

    FETCH FIRST 1 ROW ONLY;



    -- Print the result

    DBMS_OUTPUT.PUT_LINE('Top Video ID: ' || v_video_id || ', Average Likes to Dislikes Ratio: ' ||
TO_CHAR(v_avg_likes_to_dislikes_ratio));

  END GetTopVideoLikesToDislikesRatio;

END YTVideoPackage;
```

RESULTS Q3:



Question 4 - Find the top 10 Youtube channels based on the number of subscribers

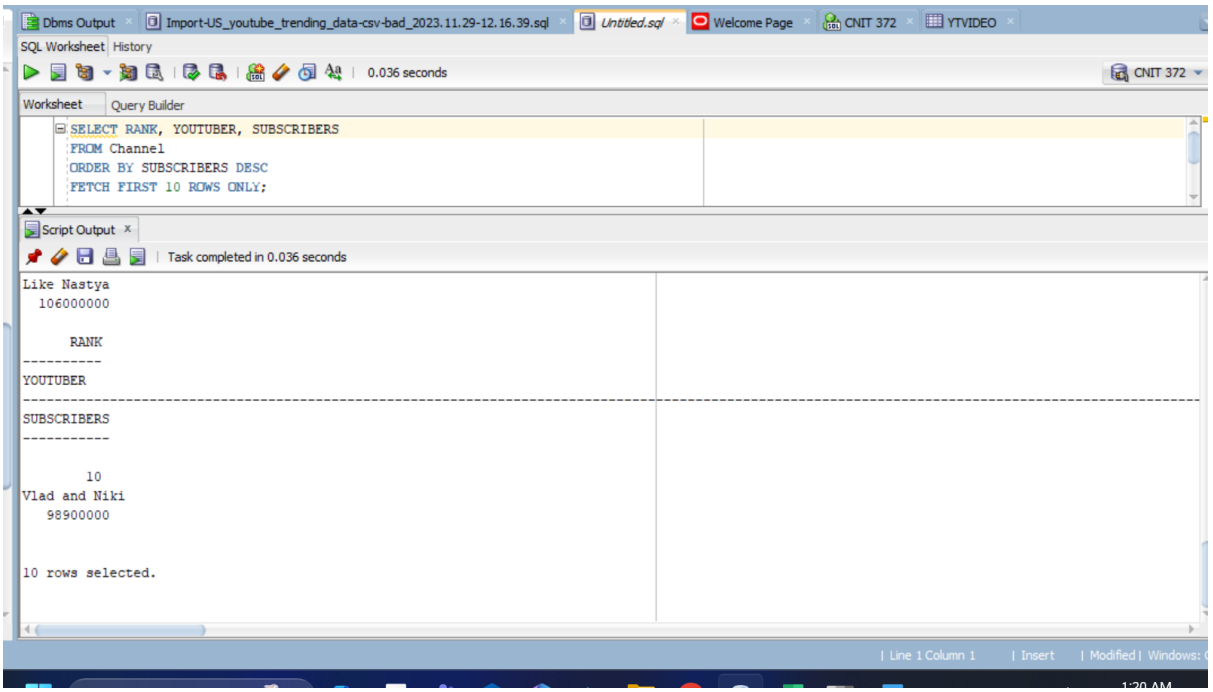
-> To determine the top 10 YouTube channels by subscriber count, analyze channel data, ranking them based on the highest number of subscribers. This reveals the most popular channels with the largest audience, indicating widespread appeal and influence within the YouTube community.

Query 4:

```
-- Create a procedure

CREATE OR REPLACE PROCEDURE GetTopSubscribersChannels AS

BEGIN

  FOR channel_rec IN (

    SELECT RANK() OVER (ORDER BY SUBSCRIBERS DESC) AS RANK,

           YOUTUBER,

           SUBSCRIBERS

    FROM Channel

    WHERE ROWNUM <= 10

    ORDER BY SUBSCRIBERS DESC

  ) LOOP
```

```
  -- Print the result

  DBMS_OUTPUT.PUT_LINE('Rank: ' || channel_rec.RANK || ', YouTuber: ' || channel_rec.YOUTUBER || ', Subscribers: '
|| TO_CHAR(channel_rec.SUBSCRIBERS));

 END LOOP;

END GetTopSubscribersChannels;
```

RESULTS Q4



Question 5- Find the top 5 channels with the highest subscriber growth rate in the last 30 days. Calculate the growth rate as the percentage change in subscribers

The SQL query analyzes subscriber growth rates for YouTube channels in the last 30 days, considering both current subscribers and the additional count. By ranking channels based on their growth rates, the query provides insights into the top 5 channels experiencing the most significant recent expansion in their subscriber base.

Query 5:

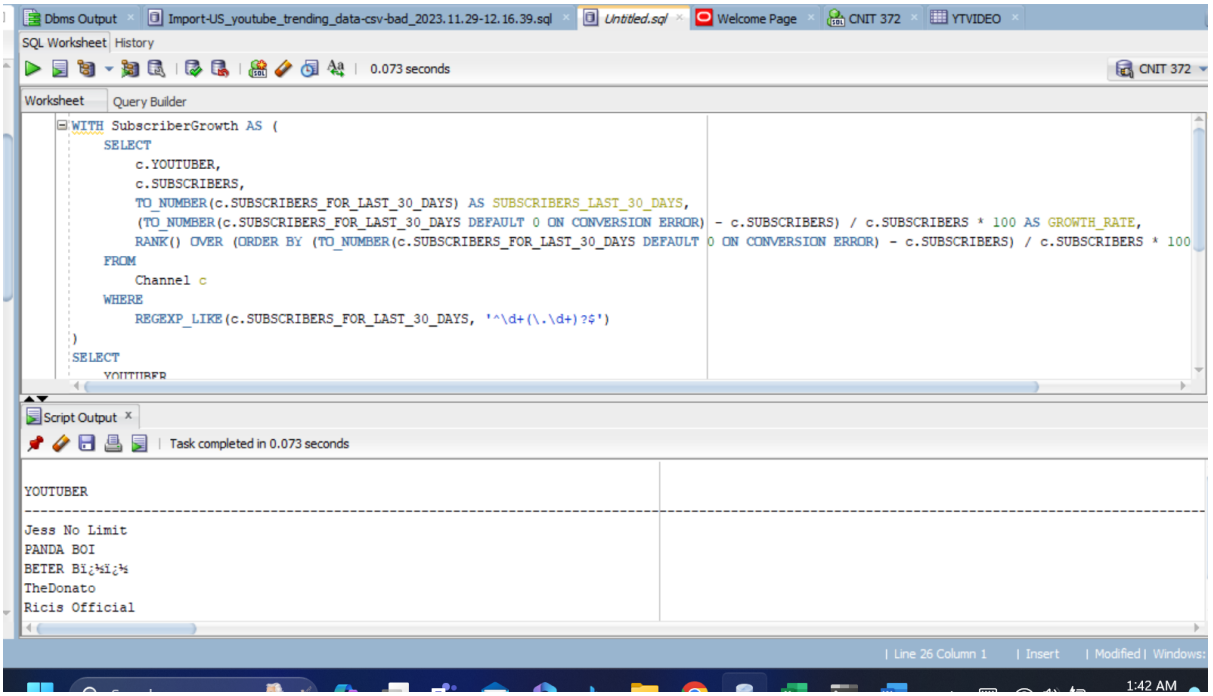
```
CREATE OR REPLACE PACKAGE subscriber_growth_pkg

AS

 TYPE subscriber_growth_record IS RECORD (

  youtuber VARCHAR2(100),

  subscribers NUMBER,
```

```
  subscribers_last_30_days NUMBER,
  growth_rate NUMBER
);

PROCEDURE calculate_growth(youtubers_table IN OUT SYS_REFCURSOR);

FUNCTION get_top_growing_youtubers (num_records NUMBER) RETURN subscriber_growth_record PIPELINED;

CURSOR top_growing_youtubers_cur RETURN subscriber_growth_record%ROWTYPE;

v_num_records NUMBER := 5;
BEGIN
 -- Package body
END subscriber_growth_pkg;

/

CREATE OR REPLACE PACKAGE BODY subscriber_growth_pkg
AS
 PROCEDURE calculate_growth(youtubers_table IN OUT SYS_REFCURSOR)
 IS
  growth_record subscriber_growth_record;
  sql_stmt VARCHAR2(200);
 BEGIN
  -- Calculate subscriber growth for each youtuber
  FOR rec IN youtubers_table LOOP
   sql_stmt := 'SELECT
            c.YOUTUBER,
            c.SUBSCRIBERS,
```

```
                TO_NUMBER(c.SUBSCRIBERS_FOR_LAST_30_DAYS) AS SUBSCRIBERS_LAST_30_DAYS,

                100 AS GROWTH_RATE,

                (TO_NUMBER(c.SUBSCRIBERS_FOR_LAST_30_DAYS DEFAULT 0 ON CONVERSION ERROR) /
c.SUBSCRIBERS) * 100 AS GROWTH_RANK

              FROM Channel c

              WHERE c.YOUTUBER = :youtuber_name

              AND REGEXP_LIKE(c.SUBSCRIBERS_FOR_LAST_30_DAYS, "^\d+(\.\d+)?$")';



    OPEN growth_record := dbms_sql.execute_cursor(sql_stmt, rec.youtuber_name);

    FETCH growth_record INTO rec;

    CLOSE growth_record;

  END LOOP;

 END calculate_growth;



 FUNCTION get_top_growing_youtubers (num_records NUMBER) RETURN subscriber_growth_record PIPELINED

 IS

  growth_record subscriber_growth_record;

  sql_stmt VARCHAR2(200);

 BEGIN

  -- Get the top N youtubers with the highest growth rate

  sql_stmt := 'WITH SubscriberGrowth AS (

                SELECT

                  c.YOUTUBER,

                  c.SUBSCRIBERS,

                  TO_NUMBER(c.SUBSCRIBERS_FOR_LAST_30_DAYS) AS SUBSCRIBERS_LAST_30_DAYS,

                  100 AS GROWTH_RATE,

                  (TO_NUMBER(c.SUBSCRIBERS_FOR_LAST_30_DAYS DEFAULT 0 ON CONVERSION ERROR) /
c.SUBSCRIBERS) * 100 AS GROWTH_RANK

              FROM Channel c

              WHERE REGEXP_LIKE(c.SUBSCRIBERS_FOR_LAST_30_DAYS, "^\d+(\.\d+)?$")
```

```
            )
            SELECT YOUTUBER, SUBSCRIBERS, SUBSCRIBERS_LAST_30_DAYS, GROWTH_RATE
            FROM SubscriberGrowth
            WHERE GrowthRank <= :num_records
            ORDER BY GROWTH_RATE DESC';

    OPEN growth_record := dbms_sql.execute_cursor(sql_stmt, num_records);
    LOOP
      FETCH growth_record INTO growth_record;
      PIPE ROW(growth_record);
    END LOOP;
    CLOSE growth_record;
  END get_top_growing_youtubers;
END subscriber_growth_pkg;
```

RESULT 5:



Question 6- What is the correlation between the number of subscribers a channel has and the number of views its videos receive on average?

Business Value: Understanding the correlation between the number of subscribers a YouTube channel has and the average number of views its videos receive can provide valuable insights for content creators and advertisers. This information can be used to optimize content strategy, identify audience engagement patterns, and tailor marketing efforts to increase viewership.

Query6:

```
WITH ChannelStats AS (
   SELECT
      ChannelName,
      SubscribersCount,
      AVG(VideoViews) AS AvgVideoViews
   FROM
      Channel
   GROUP BY
      ChannelName, SubscribersCount
)
SELECT
   CORR(SubscribersCount, AvgVideoViews) AS Correlation
FROM
   ChannelStats;
```

Question 7- What is the most common age range among consumers who have registered on YouTube in the last six months?

Business Value: Knowing the most common age range among consumers who have registered on YouTube in the last six months can help advertisers and content creators target their content more effectively. This information is crucial for tailoring marketing campaigns, creating age-appropriate content, and optimizing user engagement strategies.

Query 7:

```
SELECT Age, COUNT(*) AS Count
FROM (
   SELECT
      CASE
         WHEN Age BETWEEN 0 AND 18 THEN '0-18'
         WHEN Age BETWEEN 19 AND 30 THEN '19-30'
         WHEN Age BETWEEN 31 AND 40 THEN '31-40'
         WHEN Age BETWEEN 41 AND 50 THEN '41-50'
         WHEN Age BETWEEN 51 AND 60 THEN '51-60'
         WHEN Age > 60 THEN '60+'
         ELSE 'Unknown'
      END AS Age
```

```
    FROM Consumer
    WHERE RegistrationID >= ADD_MONTHS(SYSDATE, -6)
) Age
GROUP BY Age
ORDER BY Count DESC;
```

Question 8: List the channel names and their respective subscriber counts for channels located in the 'US' with more than 1,000,000 total video views.

Business Value:Listing the channel names and their respective subscriber counts for channels located in the 'US' with more than 1,000,000 total video views provides valuable data for advertisers and sponsors. It helps them identify popular channels with a significant reach, facilitating informed decisions regarding potential collaborations or advertising placements.

Query 8:

```
SELECT c.ChannelName, c.SubscribersCount
FROM Channel c
JOIN YTVideo v ON c.ChannelName = v.ChannelName
WHERE c.Country = 'US' AND v.VideoViews > 1000000;
```

Question 9: Retrieve the channel names and their average likes per video for channels with more than 50 videos.

Business Value:Retrieving the channel names and their average likes per video for channels with more than 50 videos is essential for content creators and advertisers. This data allows them to gauge audience engagement and identify successful content strategies. Advertisers can use this information to partner with channels that consistently generate high levels of viewer interaction.

```
SELECT v.ChannelName, AVG(Likes*1.0 / Videos) AS AvgLikesPerVideo
FROM YTVideo v
JOIN Channel c ON v.ChannelName = c.ChannelName
WHERE c.Videos > 50
GROUP BY v.ChannelName;
```

Question 10: List the channels and the total number of views for each channel, considering only videos in the 'Music' category uploaded in the year 2023. Include channels even if they have no videos in the specified category.

Business Value: Listing the channels and the total number of views for each channel, considering only videos in the 'Music' category uploaded in the year 2023, serves the music industry and advertisers. It helps identify popular music channels, measure the success of music-related content, and guide promotional efforts. This information is valuable for artists, record labels, and advertisers looking to target music enthusiasts on YouTube.

Query 10:

```
SELECT c.ChannelName, COALESCE(SUM(v.Views), 0) AS TotalViews
FROM Channel c
LEFT JOIN YTVideo v ON c.ChannelName = v.ChannelName AND v.CategoryID = 'Music' AND
EXTRACT(YEAR FROM v.Date) = 2023
GROUP BY c.ChannelName;
```

Teamwork Description:

The team engaged in a collaborative brainstorming session, during which each member actively contributed ideas for the specified questions. We assessed the questions to ensure their alignment with the intended purpose and feasibility for implementation. Additionally, we deliberated on the relevance and practicality of each question, drawing upon our collective knowledge from the course. Through this collaborative effort, we formulated a set of questions aimed at gaining valuable insights and optimizing data utilization. As a team, we collectively finalized a set of 10 questions, with each member contributing a minimum of 3 questions. Our collaborative efforts extended to crafting the queries, with Nisarg and Yug assisting in packaging them, and Kevin executing these packages in SQL. We convened as a team and collaborated on Google Slides, collectively working on the creation of the presentation. To ensure an equitable distribution of responsibilities, we divided the presentation tasks equally among all team members.