Background:

We as a group decided to create the project on utilizing Youtube data and creating useful information from said Youtube data. Youtube is a site that we all use on a daily basis and we were wondering if we could extract useful data out of it. For example, if we can look at our Youtube data and figure out what kind of videos are viewed the most or what video do we keep going back to.

Database Description:

We have four tables in our database

- Viewer
  - UserID: Unique identifier for each viewer
  - Name: Name of the viewer
  - Subscribers: Number of subscriber the viewer has
  - TimeWatched: Amount of time the view has spent watching videos
  - JoinDate: Date the viewer joined Youtube
- Channel
  - ChannelID: Unique identifier for each channel
  - ChannelName: Name of the channel
  - SubscriberCount: Number of subscribers the channel has
  - MonetizationStatus: The monetization status of the channel
  - CopyrightClaims: Number of copyright claims the channel received
- Video
  - VideoID: Unique identifier for each video
  - ChannelID: ID of the channel that uploaded the video
  - Title: Title of the video
  - Views: Number of views the video has
  - Likes: Number of likes the video has
  - DateUploaded: The date the video was uploaded
  - Genre: Genre of the video
- Viewer_Video
  - UserID: The ID of the viewer
  - VideoID: The ID of the video
  - WatchTime: Time viewer has watched the video
  - WatchDate: The date the viewer watched the video
  - CommentHistory: Comment history of the video

Questions and Solutions:

1. What genre of videos are viewed the most? This question would help a content creator determine what kinds of content are best for growing their channel. It would also help commercial entities to create ads that reach a greater audience.
    a. The procedure created what genre of video has how many views. This will display how many views each genre got from most to least. From the output of the procedure it can be seen that Music has the most views and Sports have the lead views.

    b.
```
create or replace procedure get_genre_views
is
  genre_name  video.genre%type;
  total_views number;
begin
  for genre_rec in (select genre, sum(views) as totalviews
                    from video
                    group by genre
                    order by totalviews desc)
  loop
    genre_name := genre_rec.genre;
    total_views := genre_rec.totalviews;
    dbms_output.put_line(genre_name || ': ' || total_views);
  end loop;
end;

exec genre_views_proc;
```

    c. Results:
    Music: 1815885195
    Entertainment: 221214698
    Educational: 36461305
    Informative: 8502164
    Commentary: 4382261
    Sports: 998283

2. What is the average subscriber count for each tier of Monetization status?
    a. This procedure listed the average subscriber count for each of the tiers of Monetization status (3 tiers). This will display the average of the 3 tiers of

Monetization status, rounded up to the nearest whole number. We can see that Good has the most with Limited behind it and Not Monetized in last.

b. create or replace procedure avgsubcountbymstatus
is
cursor c_channels is
select monetizationstatus, round(avg(subscribercount))
from channel
group by monetizationstatus
order by monetizationstatus;
v_monstatus channel.monetizationstatus%type;
v_avgsubcount number;
begin
open c_channels;
loop
fetch c_channels into v_monstatus, v_avgsubcount;
exit when c_channels%notfound;
dbms_output.put_line('average subscriber count for '|| v_monstatus || ' is: '
|| v_avgsubcount);
end loop;
close c_channels;
end;

execute avgsubcountbymstatus;

c. Results:
Average Subscriber Count for Good is: 19135639
Average Subscriber Count for Limited is: 53796
Average Subscriber Count for Not Monetized is: 3418

3. What is the user's favorite channel? A user can subscribe to many channels, but the number of videos they watch from these channels might make one channel more desirable.
   a. The function returns the favorite channel of the user. It will return the highest number of watched channels from that user. From the output of the function, user's 'user-ho7zh7gf2q' favorite channel is Mr. Beast.

   b. create or replace function get_favorite_channel(
      p_user_id in viewer_video.userid%type
)

```
return channel.channelname%type
is
   v_favorite_channel channel.channelname%type;
begin
   select channelname
   into v_favorite_channel
   from (
      select c.channelname, count(*) as video_count
      from viewer_video v
      join video vd on v.videoid = vd.videoid
      join channel c on vd.channelid = c.channelid
      where v.userid = p_user_id
      group by c.channelname
      order by video_count desc
   )
   where rownum = 1;

   return v_favorite_channel;
end;

select get_favorite_channel('user-ho7zh7gf2q') as favorite_channel from dual;
```

c. Results:
```
FAVORITE_CHANNEL
---------------------------------
MrBeast
```

4. If a creator were to have bots removed from their channel, their subscriber count needs to change. The creator would like to know if their subscriber count increased so a trigger could be used for this.

   a. This trigger will activate when a change is made to the subscriber count of a channel. It is triggered by a change to the VIEWER_VIDEO table. It will trigger after an insert or a delete for every row change. If a row was added, it would have a different output than if a row was removed.

   b.
```
create or replace trigger updatesubcount
after insert or delete on viewer_video
for each row
begin
```

```
 if inserting then
        update channel
        set subscribercount = subscribercount + 1
        where channelid = (select channelid from video where videoid =
:new.videoid);
        dbms_output.put_line('subscriber count was increased by 1');
 else
        update channel
        set subscribercount = subscribercount - 1
        where channelid = (select channelid from video where videoid =
:old.videoid);
        dbms_output.put_line('subscriber count was reduced by one');
 end if;
end;
```

```
insert into viewer_video values ('lohghugt953','9ne33fpquw8', to_timestamp('09:45:00',
'hh:mi:ss'), to_date('04-04-2023', 'mm-dd-yyyy'),'great video man, worth a subscription');
```

c. Results:
   Subscriber count was increased by 1

5. What videos have been watched within a given time range? Youtube may use time data to recommend videos in accordance with a certain season or time frame.
   a. The procedure created below will first create a cursor that includes videos watched by a user within a given time frame. Then the procedure will print out the title of any video that matches the requirements. If there are no videos that meet the requirements, the procedure's results will be blank.

   b.
```
create or replace procedure cnit372watchtime
(vvuserid viewer_video.userid%type, startdate viewer_video.watchdate%type,
enddate viewer_video.watchdate%type)
as
   cursor vidswatched
   is
      select v.title from video v inner join viewer_video vv
      on v.videoid = vv.videoid
      where vv.userid like vvuserid and vv.watchdate between startdate and
enddate;

   currentvid vidswatched%rowtype;
```

```
begin
   open vidswatched;

   fetch vidswatched into currentvid;

   dbms_output.put_line('videos watched within this time frame:');

   while vidswatched%found loop
      dbms_output.put_line(currentvid.title);
      fetch vidswatched into currentvid;

   end loop;

   close vidswatched;

end;

exec cnit372watchtime('user-ho7zh7gf2q', to_date('01-jan-2020', 'dd-mon-yyyy'),
to_date('17-apr-2023', 'dd-mon-yyyy'));
```

c. Results:
Videos watched within this time frame:
$1 vs $500,000 Plane Ticket!
iPhone 14 Pro Max - 25+ Tips + Tricks!

6. Based on watch history, what kinds of videos does a user regularly watch? Youtube uses a user's video preferences to make recommendations of other videos that are the same type.
   a. The procedure created below will first create a cursor with a count of all videos a user has watched, organized by genre. The procedure will then print out all the videos within the cursor.

   b.
```
create or replace procedure cnit372genrehistory
(vvuserid viewer_video.userid%type)
as
   cursor vidswatched
   is
      select v.genre, count(v.videoid) as count from video v inner join
viewer_video vv
      on v.videoid = vv.videoid
      where userid like vvuserid
```

```
        group by v.genre;

    currentvid vidswatched%rowtype;
begin
    open vidswatched;

    fetch vidswatched into currentvid;

    dbms_output.put(rpad('video genre', 25, ' '));
    dbms_output.put_line('video count');

    while vidswatched%found loop
        dbms_output.put(rpad(currentvid.genre, 25, ' '));
        dbms_output.put_line(currentvid.count);
        fetch vidswatched into currentvid;

    end loop;

    close vidswatched;

end;

exec cnit372genrehistory('user-ho7zh7gf2q');
```

c. Results:

Video genre	Video count
Entertainment	3
Informative	1

7. What videos does the user return to? One user might watch a video more than once, and knowing which videos a user likes the most will help Youtube recommended videos similar to it.
    a. The function created will return data on what videos are watched and how many times it was watched. The function requires user id to be entered. When user id ‘'user-ho7zh7gf2q'’ was entered it returned a video that was watched twice. The output has been formatted to fit better on the report.

    b.
```
create or replace function get_returned_videos(user_id in varchar2)
return sys_refcursor
is
```

```
    returned_videos sys_refcursor;
begin
    open returned_videos for
        select v.title, count(*) as returncount
        from video v
        join viewer_video vv on v.videoid = vv.videoid
        where vv.userid = user_id
        group by v.videoid, v.title
        having count(*) > 1
        order by returncount desc;

    return returned_videos;
end;

select get_returned_videos('user-ho7zh7gf2q') from dual;
```

c. Results:

TITLE	RETURNCOUNT
INSIDIOUS: THE RED DOOR – Official Trailer (HD)	2

8. A potential from an outside source may want to see the monetization status of some CHANNELS in case they want to pay them for an ad. How could this be used in a function/procedure to list the status of every channel and their monetization status?

a. The procedure was created to list all channels who have a monetization status of good. The outputs will include the channel name and their monetization status.

b.
```
create or replace procedure channelmonetization as
begin
for channel in (select channelname, monetizationstatus from channel order by
monetizationstatus)
 loop
dbms_output.put_line('channel: ' || channel.channelname || ', monetization status: '
|| channel.monetizationstatus);
end loop;
end;

execute channelmonetization;
```

c. Results:
Channel Name: MrBeast 2, Monetization Status: Good
Channel Name: Beast Philanthropy, Monetization Status: Good
Channel Name: Bozeman Science, Monetization Status: Good
Channel Name: MrBeast Gaming, Monetization Status: Good
Channel Name: ERB, Monetization Status: Good
Channel Name: Beast Reacts, Monetization Status: Good
Channel Name: Epic Rap Battles of History - Topic, Monetization Status: Good
Channel Name: MrBeast, Monetization Status: Good
Channel Name: ZZ Top, Monetization Status: Good
Channel Name: ERB2, Monetization Status: Good
Channel Name: The Beatles, Monetization Status: Good
Channel Name: Elton John, Monetization Status: Good
Channel Name: Queen Official, Monetization Status: Good
Channel Name: MrBeast ?? ???????, Monetization Status: Good
Channel Name: Rick Astley, Monetization Status: Good
Channel Name: Billy Joel, Monetization Status: Good
Channel Name: ImagineDragons, Monetization Status: Good

9. What is the ratio of likes to views? Is there a correlation on the type of video that may lead to a larger ratio of likes?
    a. The procedure was created to see if the type of video has correlation with the ratio of likes. The output is formatted so that the highest like to view ratio is first. The output displays that commentary has the highest like to view ratio while sports have the lowest like to view ratio

    b.
```
create or replace procedure get_genre_ratio as
begin
  for rec in (
    select genre, round(avg(likes/views), 4) as likeviewratio
    from video
    group by genre
    order by likeviewratio desc
  )
  loop
    dbms_output.put_line('genre: ' || rec.genre || ', like/view ratio: ' ||
rec.likeviewratio);
  end loop;
end;
```

execute get_genre_ratio;

c. Results:
Genre: Commentary, Like/View Ratio: .0336
Genre: Entertainment, Like/View Ratio: .0197
Genre: Informative, Like/View Ratio: .0165
Genre: Music, Like/View Ratio: .0116
Genre: Educational, Like/View Ratio: .0094
Genre: Sports, Like/View Ratio: .0088

10. Are there types of videos that cause a viewer to leave more comments? Is there a correlation on the type of video that may cause more comments to be left?
    a. The procedure was created to see if there was a correlation with type of video and comment. The result shows the highest number of comments to the lowest. The output shows that entertainment has the highest number of comments while informative has the lowest number of comments.

    b.
```
create or replace procedure count_comments_by_genre is
begin
  for row in (
    select genre, count(*) as comment_count
    from video v
    join viewer_video vv on v.videoid = vv.videoid
    where commenthistory is not null
    group by genre
    order by comment_count desc
  ) loop
    dbms_output.put_line(row.genre || ': ' || row.comment_count);
  end loop;
end;

execute count_comments_by_genre;
```

    c. Results:
Entertainment: 3
Educational: 2
Informative: 1

Team:

Dennis Park: Background, Database description, Solutions: 1,3,7,9,10

Logan Mabry: Table Creation, Data Collection, Insert Statements, Solutions 2, 4, 8

Jason Proffit: Github Creation and Management, Solutions 5,6,