Background

This project serves the purpose of providing and showcasing different exercises in order to become more familiar with PL/SQL and other Oracle SQL skills. The dataset consists of my (Trevor) youtube user data. The different exercises used queries, procedure, queries, cursors, triggers, aggregate functions, nested queries, and subqueries. In addition, this project aims to apply SQL skills learned in class lectures and lab to a real world situation, in this case, YouTube data. The goal is to utilize skills learned in the classroom and apply it into a real world experience showing the role SQL plays in real world scenarios.

Database Description

(Column names in red.)

TABLE 1: MY_COMMENTS

COLUMNS: URL (VARCHAR2), TEXT(VARCHAR2)

TABLE 2: SUBSCRIPTIONS

COLUMNS: CHANNEL_ID (VARCHAR2), CHANNEL_URL (VARCHAR2),

CHANNEL_TITLE(VARCHAR2)

TABLE 3: SEARCH_HISTORY

COLUMNS: TITLE (VARCHAR2), TITLE_URL (VARCHAR2), DATE_TIME (VARCHAR2)

TABLE 4: WATCH_HISTORY

COLUMNS: TITLE (VARCHAR2), TITLE_URL(VARCHAR2),

CHANNEL_NAME(VARCHAR2), VIDEO_URL(VARCHAR2), DATE_TIME(VARCHAR2)

There are no relationships like primary keys or foreign keys between any of the tables.

Questions and Solutions

1. What youtube channel did the user view the most in this dataset?
    a. **Business Value:** This is important for generating YouTube algorithms and suggested videos/content on the user's YouTube recommendations page. Seeing what YouTube channels that the user watched the most gives YouTube a good prediction of recommended videos in the future that the user would be interested in.
    b. **Answer:**
    c. **Code:**

```
CREATE OR REPLACE PROCEDURE QUESTION1 (

 date1 IN varchar2, date2 IN varchar2

)

AS

v_channel varchar2(128);

BEGIN

SELECT STATS_MODE(CHANNEL_NAME) INTO v_channel FROM WATCH_HISTORY

WHERE TO_DATE(SUBSTR(DATE_TIME, 1, 10), 'YYYY-MM-DD') BETWEEN TO_DATE(date1,
'YYYY-MM-DD') AND TO_DATE(date2, 'YYYY-MM-DD');

DBMS_OUTPUT.PUT_LINE(v_channel);

END;
```

    d. **Results:**
2. What were the top 4 channels viewed in this dataset?
    a. **Business Value:** This is important in showing which channels the user interacts with the most which is helpful in YouTube algorithms when suggesting videos on the recommended page. Whenever a new video within these channels are posted, it helps YouTube recommend it to the user.
    b. **Answer:** Shows the top 4 viewed channels.
    c. **Code:**

```
CREATE OR REPLACE PROCEDURE QUESTION2

AS

CURSOR top4_channel IS

    SELECT COUNT(channel_name), channel_name FROM WATCH_HISTORY
```

```
    GROUP BY channel_name

    ORDER BY count(channel_name) DESC

    FETCH FIRST 4 ROWS ONLY;

BEGIN

DBMS_OUTPUT.PUT_LINE('The top 4 channels visited in this dataset are: ');

for channel IN top4_channel

LOOP

DBMS_OUTPUT.PUT_LINE(channel.channel_name);

END LOOP;

END;
```

d. **Results:**

```
The top 4 channels visited in this dataset are:
XP to Level 3
Cut
Big Ten Network
FloWrestling
```

3. Which subscribed channel, if any, was viewed the most?
    a. **Business Value:** This is important the channel that the user subscribed to and viewed the most which gives a prediction on user interests and can be used to suggest other videos or channels that have the same content.
    b. **Answer:** None of the watched videos this year were subscriptions.
    c. **Code:**

```
CREATE OR REPLACE PROCEDURE QUESTION3

AS

v_channel watch_history.channel_name%type;

BEGIN

    select stats_mode(channel_name) into v_channel FROM WATCH_HISTORY

    WHERE (SELECT count(channel_ID) FROM subscriptions WHERE channel_id =
channel_name) > 0;
```

```
    if (v_channel IS NULL) THEN

    DBMS_OUTPUT.PUT_LINE('The user was not subscribed to any channels they
visited in this dataset.');



 ELSE

    DBMS_OUTPUT.PUT_LINE(v_channel);

    END IF;

END;
```

   d. **Results:** “The user was not subscribed to any channels they visited in this dataset.”

4. Has the user received or placed more Youtube comments?
   a. **Business Value:** This is important in analyzing, predicting, and understanding the user’s behavior when using YouTube. In addition, it keeps track of the comments history of the user for logging purposes.
   b. **Code:**

```
CREATE OR REPLACE PROCEDURE QUESTION4

AS

count_comments number;

count_replies number;

BEGIN

SELECT COUNT(TEXT) INTO count_comments

FROM MY_COMMENTS

WHERE TEXT = 'comment';

SELECT COUNT(TEXT) INTO count_replies

FROM MY_COMMENTS

WHERE TEXT = 'replied';

IF (count_replies > count_comments) THEN
```

```
DBMS_OUTPUT.PUT_LINE('The user has received more comments than they have
placed.');

DBMS_OUTPUT.PUT_LINE('Placed: ' || count_comments || ' | Received: ' ||
count_replies);



ELSIF (count_comments > count_replies) THEN

DBMS_OUTPUT.PUT_LINE('The user has "place more comments than they have
received.');

DBMS_OUTPUT.PUT_LINE('Placed: ' || count_comments || ' | Received: ' ||
count_replies);

ELSE

DBMS_OUTPUT.PUT_LINE('The user has placed the same amount of comments as they
have received.');

DBMS_OUTPUT.PUT_LINE('Placed: ' || count_comments || ' | Received: ' ||
count_replies);

END IF;

END;
```

   c. **Results:** "The user has received more comments than they have placed."

5. What percent of the watched videos is the user subscribed to?
   a. **Business Value:** Finding this percent can help Youtube determine which users have a low subscribe percentage. This can be useful when youtube wants to send reminders to users to subscribe to the channels they are watching. This would increase retention on Youtube and those specific channels.
   b. **Answer**: This shows an extremely low percentage of the watched videos the user is also subscribed to.
   c. **Code**:

```
 create or replace procedure percentage_watched_subs
as
 num_of_subbed number:=0;
 videos_watched number:=0;
 percentage_count number:=0;
 begin
  select count(channel_name) into num_of_subbed from
watch_history where channel_name in
 (select channel_id from subscriptions);
```

```
select count(channel_name) into videos_watched from
watch_history;
 percentage_count:=to_char(num_of_subbed / videos_watched,
'FM9999999990.000');

dbms_output.put_line('The user is subscribed to '||
percentage_count ||'% of his watched videos');
 End;
```

d. **Results**

The user is subscribed to 0% of his watched videos

6. How many watched videos have also been commented on?
   a. **Business Value:** Youtube employees wanting to investigate comment engagement can look at this answer to learn how many videos are being watched and commented on
   b. **Answer:** The results show that no comments were made on any watched videos.
   c. **Code**:

```
create or replace procedure question6
as
counter number:=0;

begin

select count(title_url) into counter from watch_history
where title_url in (select url from my_comments);

if counter=0 then
    dbms_output.put_line('There are no videos commented
that have also been watched');

else
    dbms_output.put_line('There are '||counter || ' videos
that have been commented on and watched');

end if;
end;
```

   d. Results

There are no videos commented that have also been watched

7. What is the last date a youtube video was watched?
    a. **Business Value:** This can be useful to youtube so they can keep users consistently engaged with the platform. Youtube could use our answer to check if the user hasn't watched a video in a week. A notification could be sent out telling the user to watch Youtube if it has been a week.
    b. **Answer:** This showed the user's last watched video was on April 20, 2023.
    c. **Code**:

```
create or replace procedure last_daywatched
as

 current_date date;
 convert_date date;
 tracker date;
 cursor all_dates is
    select substr(date_time, 0, 10) as each_date from
watch_history;

 begin
  select SYSDATE into current_date  from dual;
    tracker:=to_date('0001-01-01', 'YYYY-MM-DD');

for p_dates in all_dates loop
        convert_date:=to_date(p_dates.each_date,
'YYYY-MM-DD');
    if convert_date > tracker and convert_date <
current_date then
        tracker:=convert_date;
    end if;
    end loop;
    dbms_output.put_line('The last day that a youtube
video was watched was on ' || tracker);
 end;
```

    d. **Results**
    The last day that a youtube video was watched was on 20-APR-23

8. In the month of February of 2023, how many total YouTube videos did the user watch?

- **Business Value:** This requires the use of aggregate functions to count and find the sum of the number of YouTube videos watched within a specific time period, in our case, the month of February of 2023. This is useful for predicting the statistics and YoutTube usage of the user in between a certain time period which can be used for predicting future watch counts and usage trends.
- **Answer Description:** This answer shows the number of YouTube videos that the user watched during the month of February in 2023. The answer shows that during the month of February in 2023, the user watched a total of 1468 YouTube videos.
- **Code:**

```
 CREATE OR REPLACE PROCEDURE Question8(v_output out number)
AS
BEGIN
  SELECT count(*) into v_output
  FROM watch_history
  WHERE date_time LIKE '2023-02%';

  dbms_output.put_line(v_output);
END;
```

- **Results:**

```
COUNT(*)
----------
      1468
```

9. Which YouTube channel was watched the most in 2023?
    - **Business Value:** This requires an understanding of using a subquery, aggregate function count, and aggregate max to find the YouTube channel that the user watched the most. This is important for generating YouTube algorithms and suggested videos/content on the user's YouTube recommendations page.
    - **Answer Description:** This answer shows the YouTube channel that was watched the most during the year of 2023. It shows that the channel name of 'XP to Level 3' was watched the most.
    - **Code:**

```
CREATE OR REPLACE PROCEDURE Question9(v_output out varchar2)
AS
BEGIN
  SELECT channel_name into v_output
  FROM watch_history
  WHERE date_time LIKE '2023%'
  GROUP BY channel_name
  HAVING count(*) = (
    SELECT max(count(*))
    FROM watch_history
```

```
    WHERE date_time LIKE '2023%'
    GROUP BY channel_name
  );
  dbms_output.put_line(v_output);
END;
```

- **Results:**
  CHANNEL_NAME

  XP to Level 3

10. Up until now, how many videos have the user watched whose title contained the word 'GTA'?

- **Business Value:** This requires an understanding of using aggregate functions to find all the videos that the user watched containing the word ‘GTA’. This is useful to track user watching habits and predict and suggest video recommendations for the user.
- **Answer Description:** This answer gives the total count of videos that the user watched containing the topic ‘GTA’. The total number of videos watched by the user containing the topic ‘GTA’ is 63.
- **Code:**

```
CREATE OR REPLACE PROCEDURE Question10(v_output out number)
AS
BEGIN
  SELECT count(*) into v_output
  FROM watch_history
  WHERE title LIKE '%GTA%';

  dbms_output.put_line(v_output);
END;
```

- **Results:**
  COUNT(*)

  63

Team Members and Contributions

Our team consists of three members: Nina Li, Nebey Gebreslassie, and Trevor Vannatta. For our CNIT 372 project, each member of the team made individual contributions. Our team occasionally meets to discuss the execution of this project from milestone to milestone. We made sure that everyone understood what was going on from milestone to milestone so that no

one is left behind on the work. For milestone 1, all members of the team discussed and agreed on a project topic. Nebey and Nina worked on writing the milestone 1 description which talks about the insights derived from our YouTube data and why they are important in addition to designing the database. Trevor worked on outlining the tables and describing their column names, data types, and relationships. For milestone 2, we had to generate questions in addition to explaining their importance and business value. We split up the questions with two members writing three questions each and one member who will write four questions. Trevor worked on three questions and the explanations. Nina worked on three questions/explanations and wrote the teamwork description. Nebey worked on the remaining four questions for the milestone, in addition to their explanations. Finally, for milestone 3, Trevor worked on extracting the YouTube data from the account, so all members can import the file into SQL. Trevor also worked on four questions. Nebey made a Github repository for the group to store the milestone files and worked on three questions. Nina worked on the remaining three questions in addition to writing contributions of each member and the team.