PROJECT II

INSTAGRAM USER ANALYTICS

PROJECT DESCRIPTION

This project deals with the Instagram user data. It basically involves analyzing user interactions and engagement with the Instagram app. The main objective of this project is to draw valuable insights as required by the team. It is the role of the data analyst to extract valuable information from the data resources provided which might help the business to grow and help the team to take the necessary steps ahead for further development.

As per the instructions given it was mentioned to carry out the required tasks using SQL. Therefore I intend to carry out my tasks using MySQL Workbench. Using the Database provided I ran some queries that were related to the tasks and I found some valuable answers to the questions raised.

APPROACH

I have used various statements to run my queries using DML commands and SQL clauses. There were 7 tables in the database. While querying I used Join statements to join the tables which were quite essential for the analysis purposes. A few clauses that were commonly used were ordered by, group by, and limit clauses.

The tasks included two parts A) Marketing Analysis

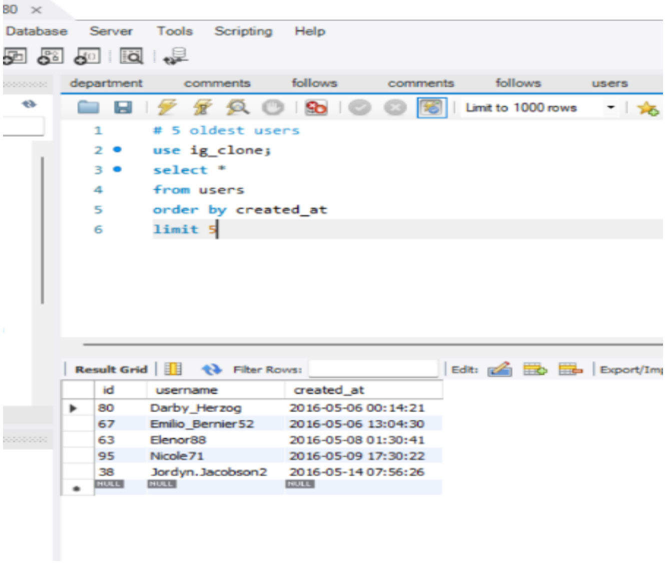
B) Investor Metrics

A 1) Loyal User Reward

The first task was to identify 5 loyal users who have been using the platform for the longest time. The team wanted to reward the most loyal users.

Using the users table from the ig_clone database I found the 5 users who joined the platform in 2016. I used the 'order by' clause to sort the users according to the time they joined and used the 'limit' clause to identify the 5 members.

The SQL input and output is given below:



```
# 5 oldest users
use ig_clone;
select *
from users
order by created_at
limit 5
```

id	username	created_at
80	Darby_Herzog	2016-05-06 00:14:21
67	Emilio_Bernier52	2016-05-06 13:04:30
63	Elenor88	2016-05-08 01:30:41
95	Nicole71	2016-05-09 17:30:22
38	Jordyn.Jacobson2	2016-05-14 07:56:26
NULL	NULL	NULL

In this picture, you can see the 5 users that have been using the platform for the longest time.

The Loyal User Reward. can be rewarded to these 5 users by the team.

A 2) Inactive User Engagement

The second task was to find the users who had never posted a single photo on Instagram. So that the team can send promotional emails to encourage inactive users to start posting.

Using the photos table and users table, I was able to find the inactive users who have never posted a post yet. In order to merge the two tables I used the ‘join’ clause to run the query. While running the query I found there were 26 inactive users who have never posted anything on their feed yet.

The SQL input commands used and the output obtained is given below for reference:

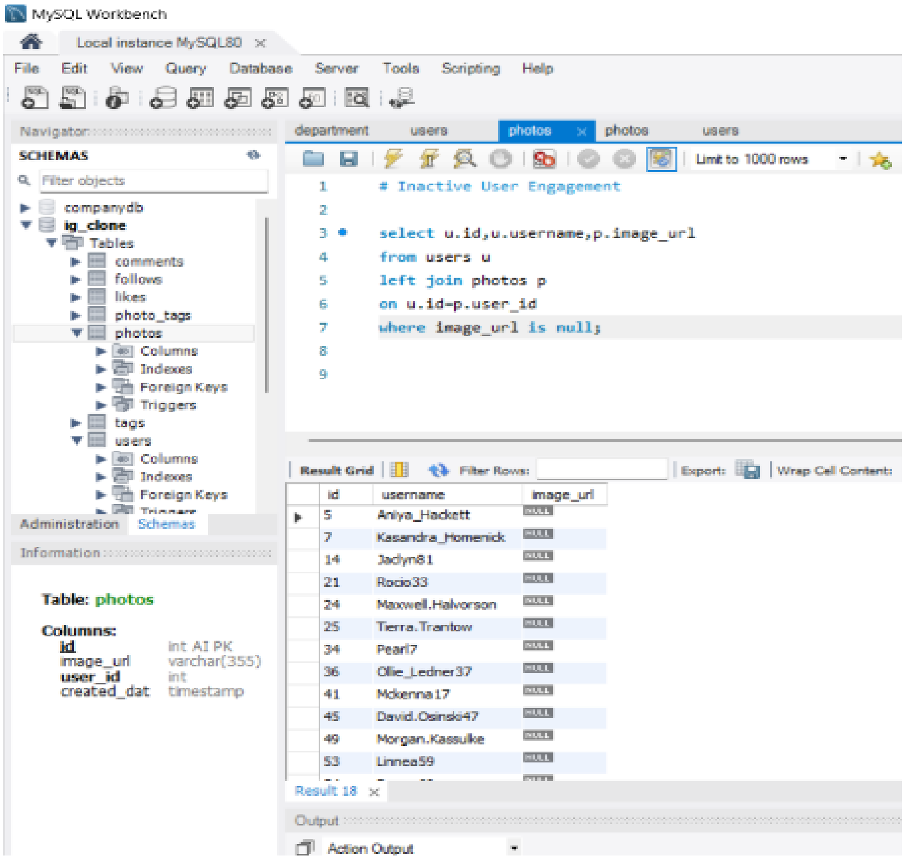


Table: photos

Columns:

id	int AI PK
image_url	varchar(355)
user_id	int
created_dat	timestamp

```sql
# Inactive User Engagement

select u.id,u.username,p.image_url
from users u
left join photos p
on u.id=p.user_id
where image_url is null;
```

id	username	image_url
5	Aniya_Hackett	NULL
7	Kasandra_Homenick	NULL
14	Jaclyn81	NULL
21	Rocio33	NULL
24	Maxwell.Halvorson	NULL
25	Tierra.Trantow	NULL
34	Pearl7	NULL
36	Ollie_Ledner37	NULL
41	Mckenna17	NULL
45	David.Osinski47	NULL
49	Morgan.Kassulke	NULL
53	Linnea59	NULL

This is the list of inactive users who have not posted anything yet.

The team can send promotional emails to these users to encourage them to post.

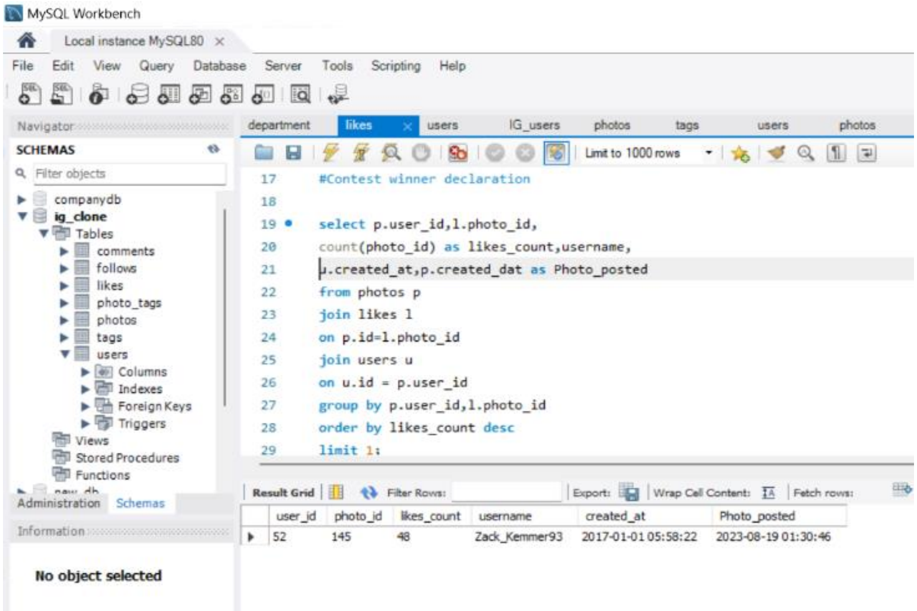
A 3) Contest Winner Declaration

The third task was to determine the winner of the contest and provide their details to the team. The team has organized a contest where the user with the most likes on a single photo wins.

Since it’s a contest that involves finding the most liked photo, I used the photos table, likes table, and also the users table from the database because it was mentioned to give further details about the winner. I joined these tables using the join statement and also used a few clauses to sort them and group them accordingly. Finally using the limit clause I found the most liked photo and the user.

The winner of the contest is Zack_Kemmer93(user_id:52) who posted his photo(photo_id:145) on 2023-08-19. He got 48 likes on his photo. He created his account on 2017-01-01.

The queries I ran to find the winner are given below:



```
#Contest winner declaration

select p.user_id,l.photo_id,
count(photo_id) as likes_count,username,
u.created_at,p.created_dat as Photo_posted
from photos p
join likes l
on p.id=l.photo_id
join users u
on u.id = p.user_id
group by p.user_id,l.photo_id
order by likes_count desc
limit 1;
```

user_id	photo_id	likes_count	username	created_at	Photo_posted
52	145	48	Zack_Kemmer93	2017-01-01 05:58:22	2023-08-19 01:30:46

The winner's details can be obtained from this query.

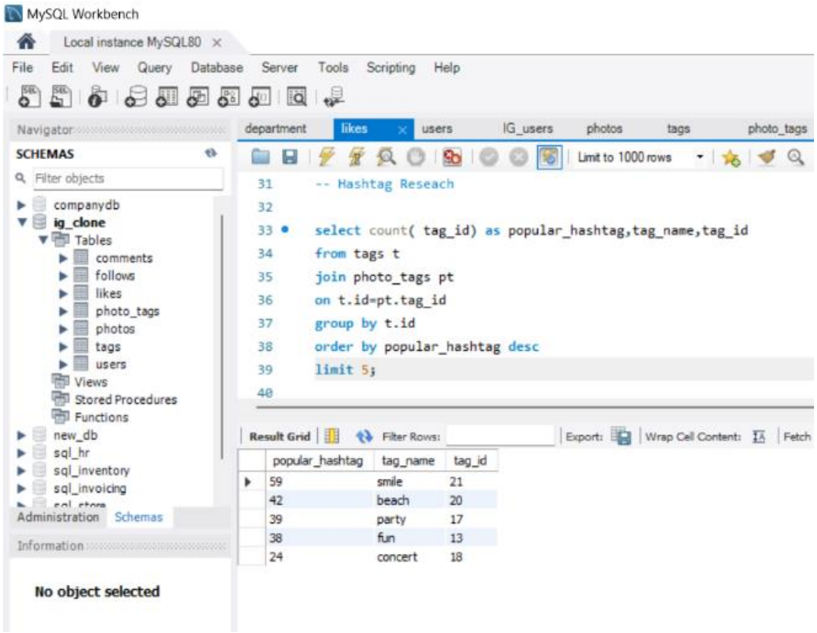
The team can declare Zack_kimmer93 as

A4) Hashtag Research

The fourth task was to identify the top five most commonly used hashtags on the platform. So that a partner brand can use those hashtags in their post to reach most people.

In order to find the most popular hashtag, I used the tags table and photo_tags table and combined them using the join clause. Then sorted them, grouped them accordingly, and also used a limit clause to identify the top 5 popular hashtags.

After running the following queries, I found the most used hashtags



```
-- Hashtag Reseach

select count( tag_id) as popular_hashtag,tag_name,tag_id
from tags t
join photo_tags pt
on t.id=pt.tag_id
group by t.id
order by popular_hashtag desc
limit 5;
```

popular_hashtag	tag_name	tag_id
59	smile	21
42	beach	20
39	party	17
38	fun	13
24	concert	18

Hence these are the top 5 most commonly used hashtags in the platform.

The partner brand can use these 5 hashtags in their post to reach the most people.

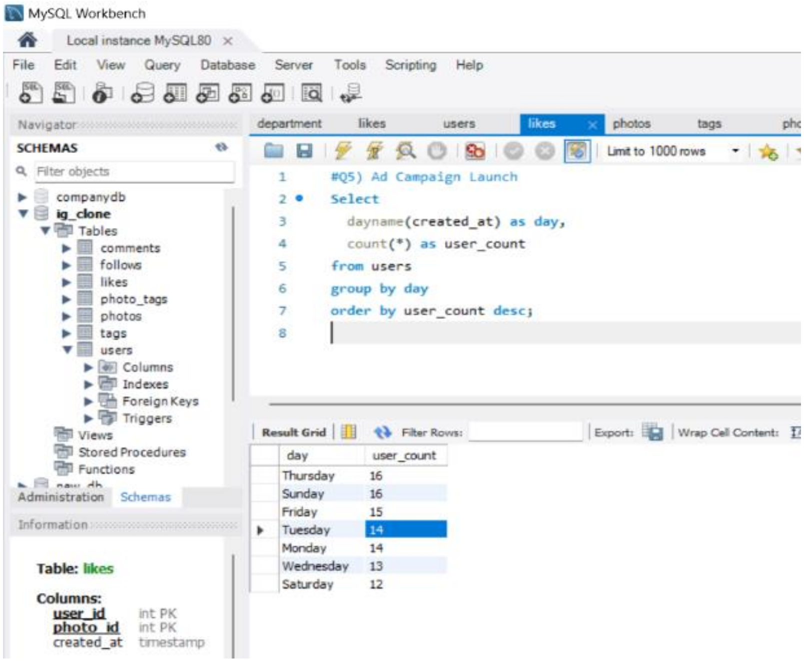
A5) Ad Campaign Launch

The fifth task was to determine the day of the week when most users registered on Instagram. So that the team can launch the ads on the best days of the week to reach more users.

The users table consists of username, id, and the created_at columns, therefore we can make use of this table to identify the day of the week when most users register on the platform. Since the team wants to know the name of the day, I used the dayname() function to extract information from the created_at column as it was in timestamp format. Followed by 'group by' and 'order by' clauses were also used to sort and arrange the days.

Among 100 users, Thursdays and Sundays show the highest count in registration. 16 users registered on these days respectively. Saturday showed the lowest count in registration with only 12 users.

These are the statements used to run the above query:



```
#Q5) Ad Campaign Launch
Select
  dayname(created_at) as day,
  count(*) as user_count
from users
group by day
order by user_count desc;
```

day	user_count
Thursday	16
Sunday	16
Friday	15
Tuesday	14
Monday	14
Wednesday	13
Saturday	12

Table: likes

Columns:
user_id int PK
photo_id int PK
created_at timestamp

Thursdays and Saturdays show the highest count.

The team can launch ads on either of these days to reach most users.

B) Investor Metrics

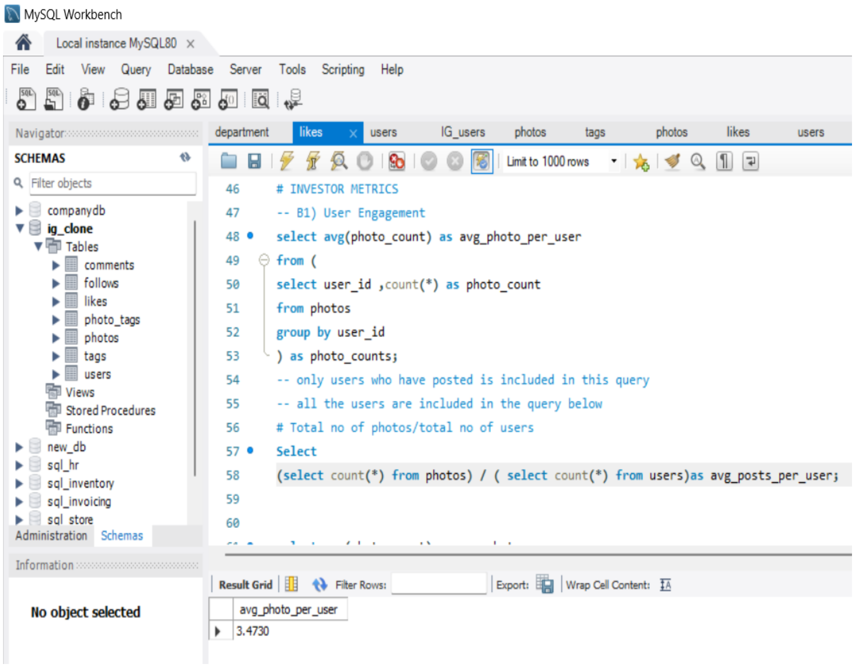
B1) User Engagement

The next task was to calculate the average no of posts per user on Instagram and also provide (the total number of photos / total number of users). Thereby the investors can know if the users are still active and posting on Instagram or if they are making fewer posts.

i)In the first part, only the users who posts are considered thereby I only used the photos table. A subquery was created to find the average no of users who posted. Using the subquery created we can use AVG() function to find the average number of posts per users.

The average no of posts per user is 3.4730.

The following queries were used to find the average no of posts per user.



```sql
# INVESTOR METRICS
-- B1) User Engagement
select avg(photo_count) as avg_photo_per_user
from (
select user_id ,count(*) as photo_count
from photos
group by user_id
) as photo_counts;
-- only users who have posted is included in this query
-- all the users are included in the query below
# Total no of photos/total no of users
Select
(select count(*) from photos) / ( select count(*) from users)as avg_posts_per_user;
```

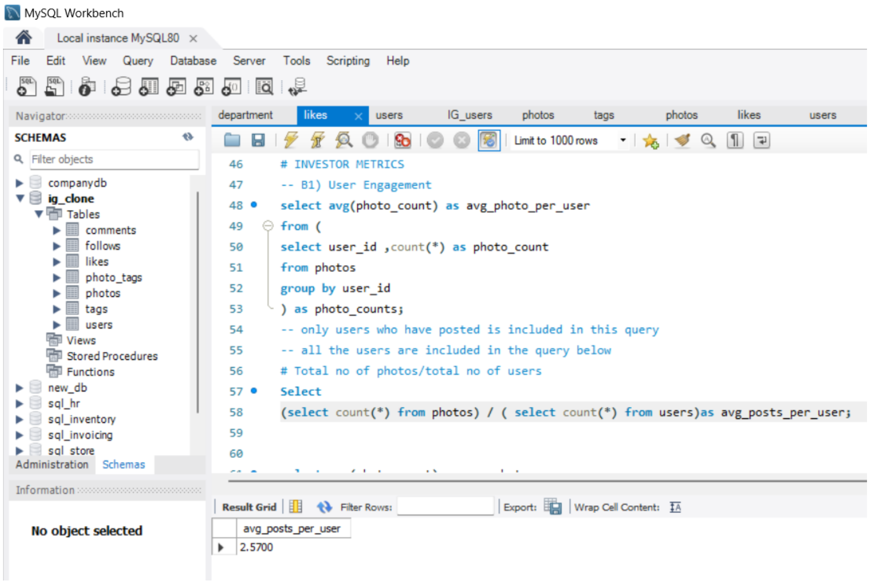
```
avg_photo_per_user
3.4730
```

Note that only users who have posted photos are considered.

ii)The second part of the task was to find the total no of photos/ total no of users

In this part all the users are considered so we have to use both photos table and users table.

The output of the above query is 2.57



```sql
# INVESTOR METRICS
-- B1) User Engagement
select avg(photo_count) as avg_photo_per_user
from (
select user_id ,count(*) as photo_count
from photos
group by user_id
) as photo_counts;
-- only users who have posted is included in this query
-- all the users are included in the query below
# Total no of photos/total no of users
Select
(select count(*) from photos) / ( select count(*) from users)as avg_posts_per_user;
```

```
avg_posts_per_user
2.5700
```

Note that in this query all the users are considered.(i.e those who haven't posted yet are also included)

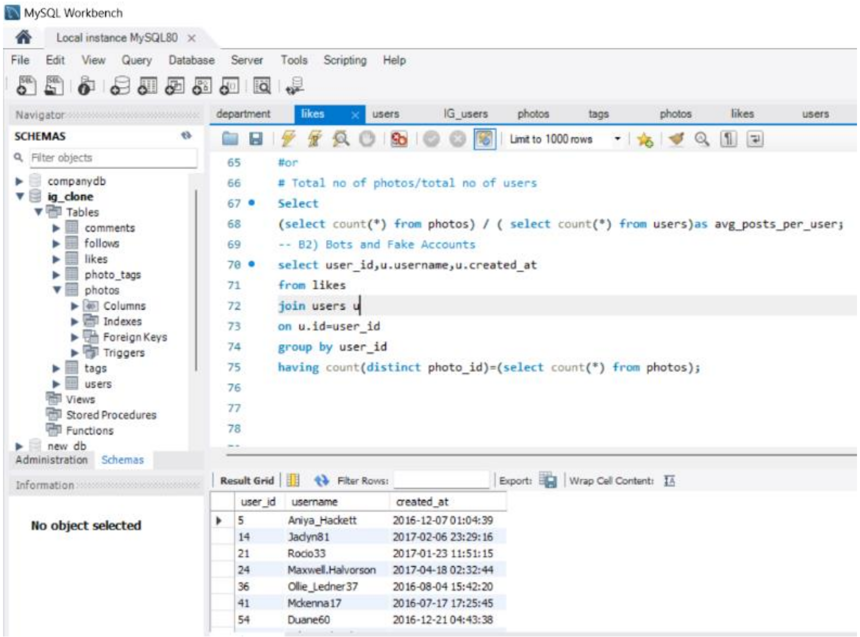
B2) Bots and Fake Accounts

The final task is to identify potential bots i.e. users who have liked every single photo on the site. Thereby the investors can know whether the platform is crowded with dummy or fake accounts.

Three tables from the database were used to run the above query. The likes table, users table, and photos tables were used. Using group by and having clauses, the potential bots were able to be identified. The having clause helps to return the row where the condition is true.

There were 13 potential bots on the platform that liked every single photo on the site.

The following queries were used to find the potential bots in the platform:



```
#or
# Total no of photos/total no of users
Select
(select count(*) from photos) / ( select count(*) from users)as avg_posts_per_user;
-- B2) Bots and Fake Accounts
select user_id,u.username,u.created_at
from likes
join users u
on u.id=user_id
group by user_id
having count(distinct photo_id)=(select count(*) from photos);
```

user_id	username	created_at
5	Aniya_Hackett	2016-12-07 01:04:39
14	Jaclyn81	2017-02-06 23:29:16
21	Rocio33	2017-01-23 11:51:15
24	Maxwell.Halvorson	2017-04-18 02:32:44
36	Ollie_Ledner37	2016-08-04 15:42:20
41	Mckenna17	2016-07-17 17:25:45
54	Duane60	2016-12-21 04:43:38

If you run these queries one can find 13 potential bots in the platform.

The investors can find the details of the bots from these queries.

TECH-STACK USED

I used MySQL workbench version 8.0.34 build 3263449 CE(64 bits) for this project. The reason I used MySQL workbench for my project is because I was more comfortable using this software. I practiced my SQL queries mostly in this software. MySQL workbench is user-friendly and is also open-source.

INSIGHTS

While working on this project I got to learn more about SQL querying. Each and every question was challenging in its own way. To explore the database and drawing relevant information from the database was quite interesting. Each time you run a query you learn a bit more about SQL. Though this was my first SQL project, I enjoyed working on this project. The SQL sessions really helped me to get through this project. Indeed it was a different experience to put the knowledge you gained to put in use.

The Instagram User Analytics database has 7 tables. Each table was significant in its own way and contained data that was valuable for analysis. Using some basic SQL clauses we could run our queries and find resourceful information from the data that will benefit the team. The team can further use these valuable insights for their further development and progress. One valuable insight I found while querying is the presence of bots. Identifying bots is crucial because it has the influence to misinterpret the output we obtain. For eg, if we want to find the most liked photo, the presence of these bots might mislead us. Other significant inferences obtained were mostly task related queries. They include the presence of inactive users who have not used, the most used hashtags and the days in which most users register on this platform. Thereby the team can take required actions using these valuable sources of information. The team can encourage more people to be active on this platform by conducting more contests so that many inactive users try to participate in this contest.

RESULT

I was successfully able to complete all the tasks and that is so far one of my biggest achievements. More importantly, the knowledge and experience I gained while performing these tasks have benefitted me and encouraged me to learn more about SQL. Learning SQL is one thing but putting what you learned in use is a different and challenging experience.

Definitely, it has a positive impact on me, as it helped me to explore more about SQL applications on real-life datasets. Gaining practical experience is as important as learning SQL sessions. I am glad that I could use the knowledge I gained through the SQL sessions to put to practical use and derive insightful queries.

END

Submitted by,

S Nandana.