**Create database Instagram\_Project;**

**use Instagram\_Project;**

Create table users(

Id integer autoincrement primary key,

User\_name varchar(30) not null,

created\_at timestamp default current\_date());

Create table photos(

ID integer autoincrement primary key,

image\_url varchar(400) not null,

user\_id integer not null,

Created\_date timestamp default current\_date(),

foreign key (user\_id) references users (id) );

Create table comments(

ID integer autoincrement primary key,

comment\_text varchar(400),

User\_id integer not null,

photo\_id integer not null,

created\_at timestamp default current\_date(),

foreign key (User\_id) references users (id),

foreign key (photo\_id) references photos(id));

create table likes(

user\_id integer not null,

repetating as per number of likes

photo\_id integer not null,

created\_at timestamp default current\_date(),

foreign key (user\_id) references users(id),

foreign key (photo\_id) references Photos(Id),

primary key (user\_id, photo\_id));

create table follows(

follower\_id integer not null,

followee\_id integer not null,

created\_at timestamp default current\_date(),

foreign key (follower\_id) references users(id),

foreign key (followee\_id) references users(id),

primary key (follower\_id,followee\_id)) ;

Create table tags(

ID integer autoincrement primary key,

tag\_name varchar(333) unique not null,

create\_at timestamp default current\_date());

create table photo\_tags(

photo\_id integer not null,

tag\_id integer not null,

foreign key (photo\_id) references photos(id),

foreign key (tag\_id) references tags(id),

primary key (photo\_id,tag\_id));

select \* from users;

select \* from photos;

select \* from follows;

select \* from comments;

select \* from likes;

select \* from tags;

select \* from photo\_tags;

**Questions are as below,**

**/\*We want to reward our users who have been around the longest.**

**Find the 5 oldest users.\*/**

Create table Oldest\_users as (

select \* from users order by created\_at limit 5);

**/\*What day of the week do most users register on?**

**We need to figure out when to schedule an ad campgain\*/**

Create table Most\_id\_created\_date as (

select dayname(created\_at) as day, count(\*) as total from users group by day order by total desc limit 2);

**/\*We want to target our inactive users with an email campaign.**

**Find the users who have never posted a photo\*/**

Create table Users\_never\_posted\_photo as(

select u.User\_name,p.id from users u left outer join photos p on u.id=p.user\_id where p.user\_id is null);

**/\*We're running a new contest to see who can get the most likes on a single photo (we can get image with more likes as below).**

**WHO WON??!!\*/**

;

Create table photos\_with\_more\_likes as (

select

photos.image\_url, count(likes.\*) as total\_likes from likes left outer join photos on photos.id = likes.photo\_id join users

on users.id = likes.user\_id group by photos.image\_url order by total\_likes desc limit 1);

**/\*Our Investors want to know...**

**How many times does the average user post?\*/**

Create table avg\_post\_by\_users as (

select round((select count(\*) from photos)/(select count(\*) from users)) as avg\_users); -- not sure why ,2

**--user ranking by postings higher to lower**

;

Create table Most\_posted\_user as (

select users.user\_name,count(photos.created\_date) as total\_post from users left outer join photos on users.id = photos.user\_id

group by users.user\_name order by total\_post desc);

**/\*total numbers of users who have posted at least one time \*/**

;

Create table Users\_posted\_only\_once as(

with sample\_table as (select users.user\_name,count(photos.image\_url) as total\_posts from users left outer join photos on users.id = photos.user\_id group by users.user\_name

order by total\_posts)

select \* from sample\_table where total\_posts = 1);

**/\*A brand wants to know which hashtags to use in a post**

**What are the top 5 most commonly used hashtags? \*/**

Create table most\_popular\_hashtag as (

select tags.tag\_name, count (tags.tag\_name) as count\_of\_tag from tags right outer join photo\_tags on photo\_tags.tag\_id = tags.id

group by tags.tag\_name

order by count\_of\_tag desc

limit 5);

**/\*We have a small problem with bots on our site...**

**Find users who have liked every single photo on the site\*/**

;

Create table Person\_who\_liked\_allphotos as(

select users.user\_name, count(likes.photo\_id) as total\_likes from users left outer join likes on users.id = likes.user\_id group by users.user\_name

having total\_likes =(select count(\*) from photos))

;

**/\*We also have a problem with celebrities**

**Find users who have never commented on a photo\*/**

Create table Users\_with\_nocomments as (

select users.user\_name, comments.comment\_text from users left outer join comments on users.id = comments.user\_id

where comments.comment\_text is null);

**Find the percentage of our users who have either never commented on a photo or have commented on every photo\*/**

;

Create table Percentage\_of\_commenters\_and\_non\_commenters as (

with sample\_data as (

select \* from

(with No\_comment as(

with Table\_A as (select users.user\_name,comments.comment\_text as comment\_text from users left outer join comments on users.id = comments.user\_id)

select comment\_text from Table\_A HAVING comment\_text IS NULL)

select count(\*) as total\_number\_of\_users\_without\_comments from No\_comment) as tablea

join

(with all\_comment as (

with tableB as (select users.user\_name as user\_name,comments.comment\_text as comment\_text from users left outer join comments on users.id = comments.user\_id)

select user\_name,count(comment\_text) as count\_of\_com from tableB group by user\_name

having count\_of\_com = (select count(\*) from photos))

select count(\*) as Total\_users\_commented\_on\_every\_photo from all\_comment) as tableb)

select (TOTAL\_NUMBER\_OF\_USERS\_WITHOUT\_COMMENTS/(select count(\*) from users)\*100) as Non\_commenters\_in\_percentage,

(TOTAL\_USERS\_COMMENTED\_ON\_EVERY\_PHOTO/(select count(\*) from users)\*100) as commenters\_in\_percentage from sample\_data)

;

**/\*Find users who have ever commented on a photo\*/**

Create table users\_never\_commented as (

with No\_comment\_table as

(with sample\_data as

(select users.user\_name as user\_name,comments.comment\_text as comment\_text from users left outer join comments on users.id = comments.user\_id )

select \* from sample\_data having comment\_text is null)

select count(\*) as user\_never\_commented from No\_comment\_table);

**Find the percentage of our users who have either never commented on a photo or have commented on photos before\*/**

;

Create table percentage\_of\_commemted\_before\_and\_never as (

with comment\_and\_non\_commenters\_table as

(

select \* from

(with commenters\_table as

(select users.user\_name as user\_name,comments.comment\_text as comment\_text from users right outer join comments on users.id = comments.user\_id)

select count (distinct(user\_name)) as commenters from commenters\_table) as tablea

join

(with non\_commenters\_table as

(with commenters\_table as

(select users.user\_name as user\_name,comments.comment\_text as comment\_text from users left outer join comments on users.id = comments.user\_id)

select \* from commenters\_table having comment\_text is null)

select count(\*) as non\_commenters from non\_commenters\_table) as tableb)

select (COMMENTERS/(select count (\*) from users)\*100) as percentage\_of\_commenters,

(non\_commenters/(select count(\*) from users)\*100) as percentage\_of\_non\_commenters from comment\_and\_non\_commenters\_table);