**Database project Phase-03 Submission**

**Project Idea: portfolio terrain – A dedicated social media platform for freelance and remote developers**



**Submitted by:**

**Usman Shahid (L1F22BSCS1057) &**

**Ali Hamza Sehole (L1F21BSCS0791)**

**Section: D-8**

**Submitted to:**

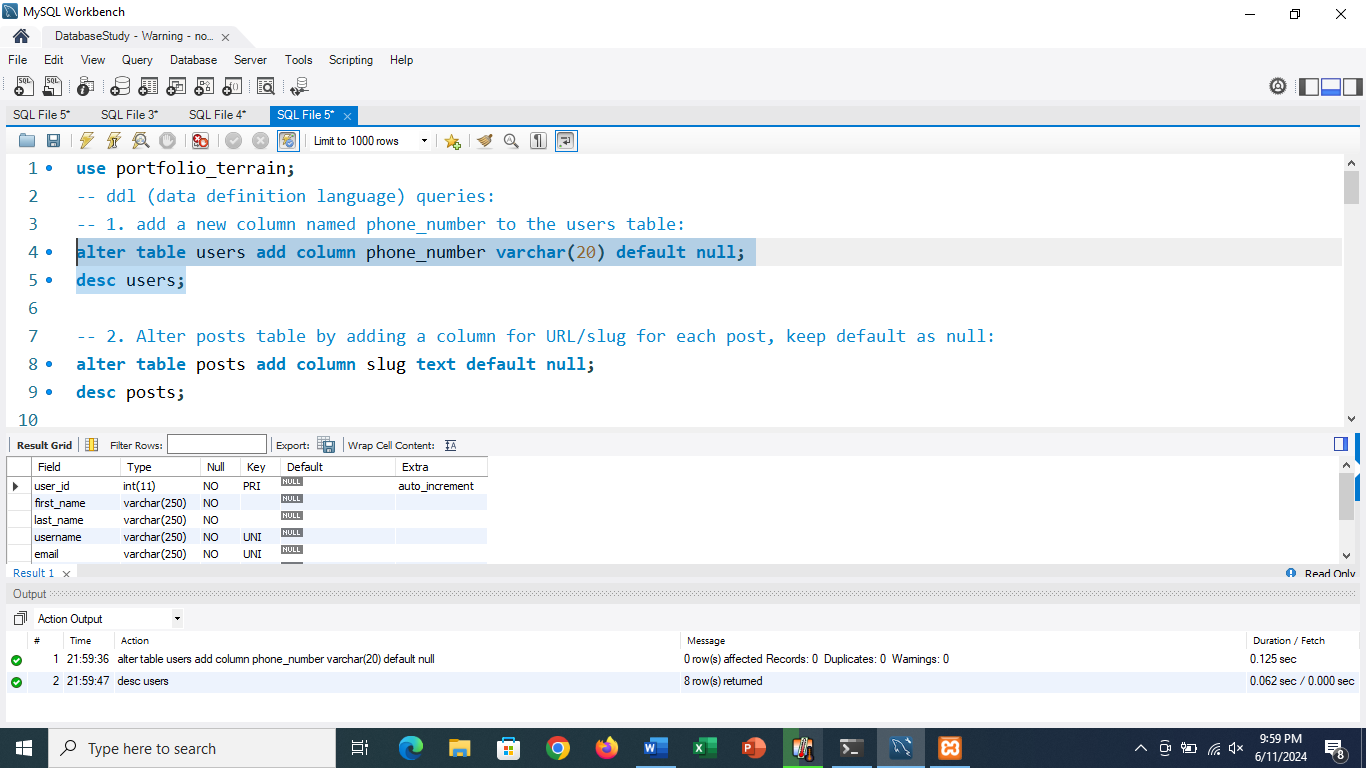
**Sir Afham Nazir**

use portfolio\_terrain;

**DDL (data definition language) queries:**

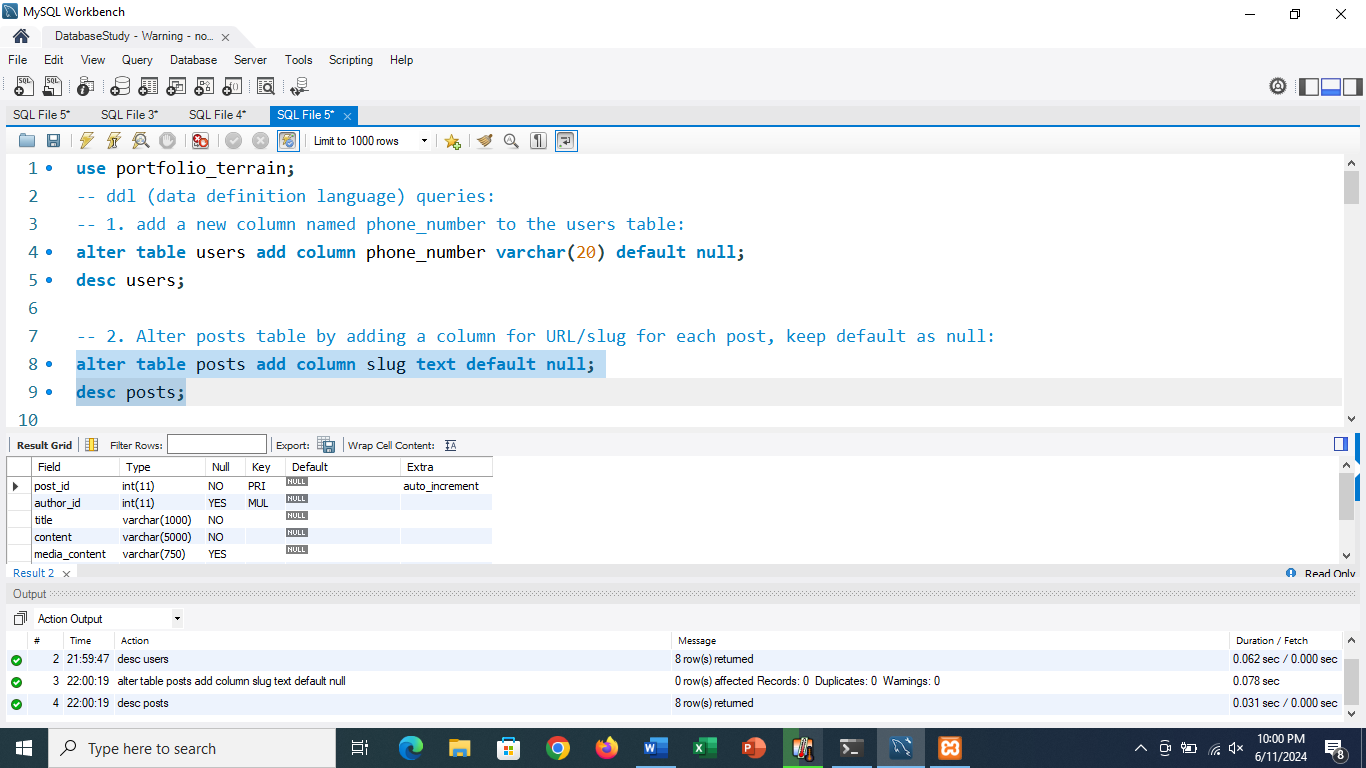
**1. add a new column named phone\_number to the users table:**

alter table users add column phone\_number varchar(20) default null;



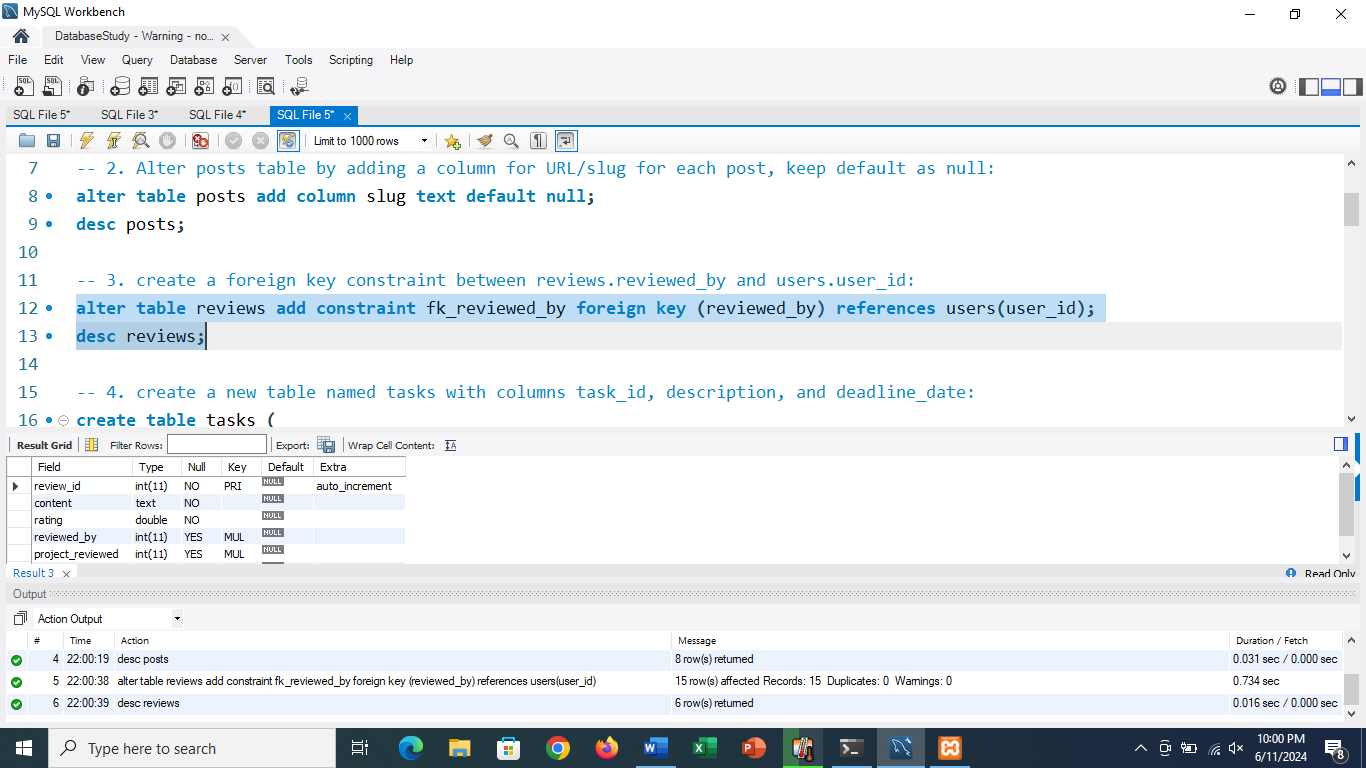
**2. Alter posts table by adding a column for URL/slug for each post, keep default as null:**

alter table posts add column slug text default null;



**3. create a foreign key constraint between reviews.reviewed\_by and users.user\_id:**

alter table reviews add constraint fk\_reviewed\_by foreign key (reviewed\_by) references users(user\_id);



**4. create a new table named tasks with columns task\_id, description, and deadline\_date:**

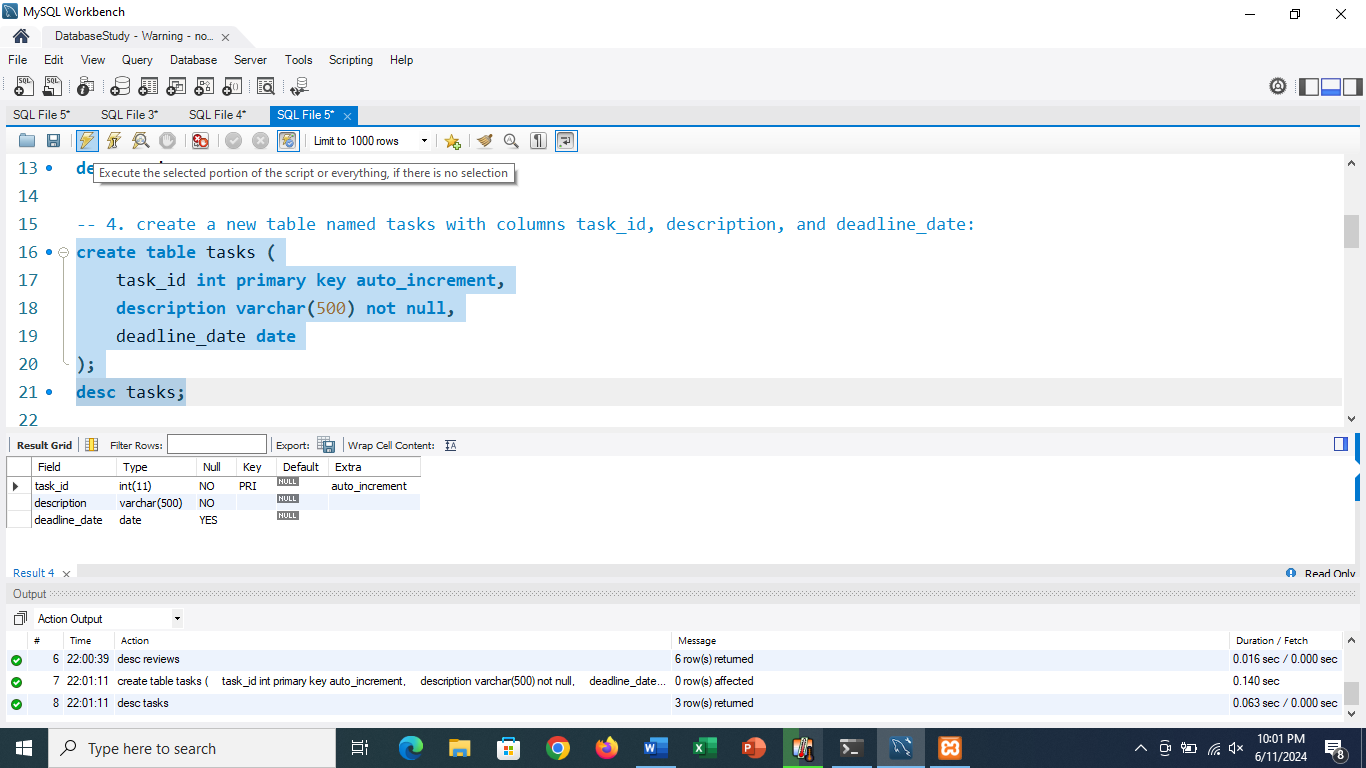
create table tasks (

task\_id int primary key auto\_increment,

description varchar(500) not null,

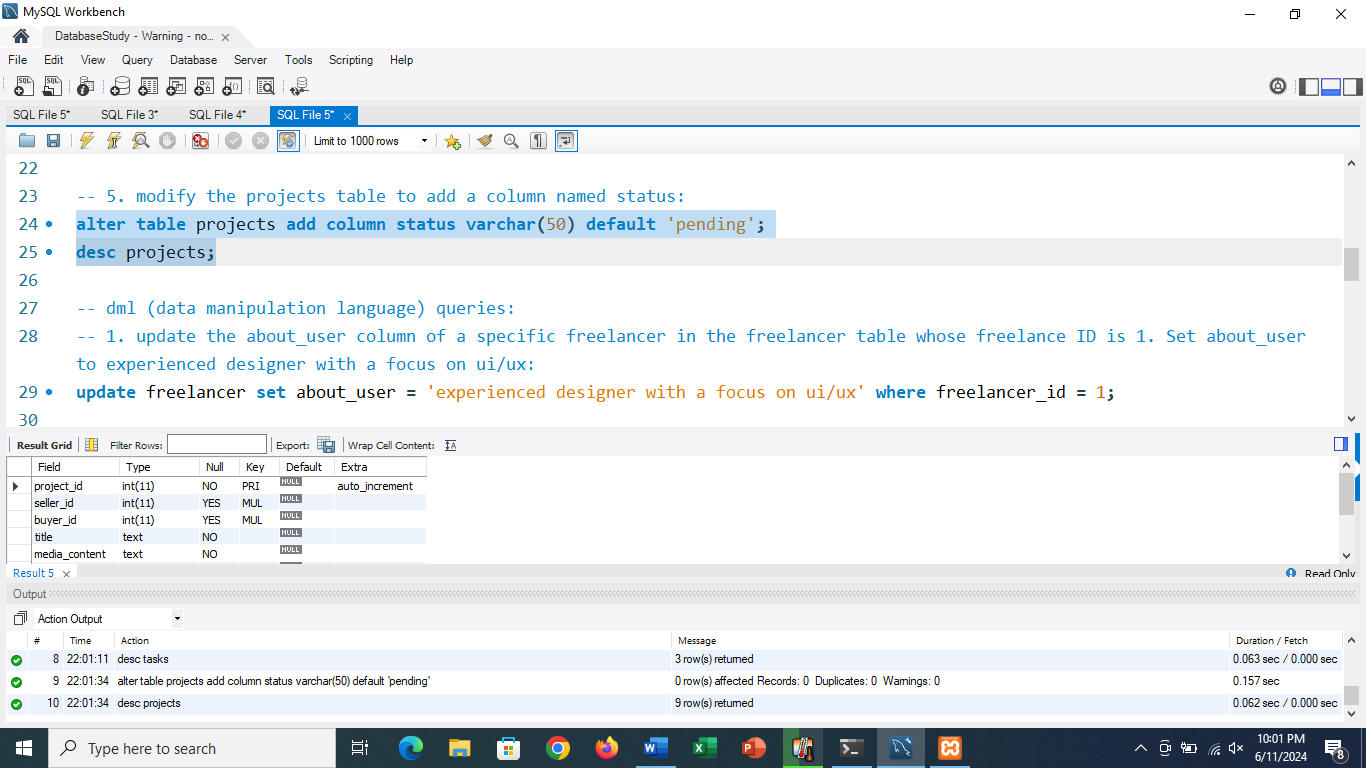
deadline\_date date

);



**5. modify the projects table to add a column named status:**

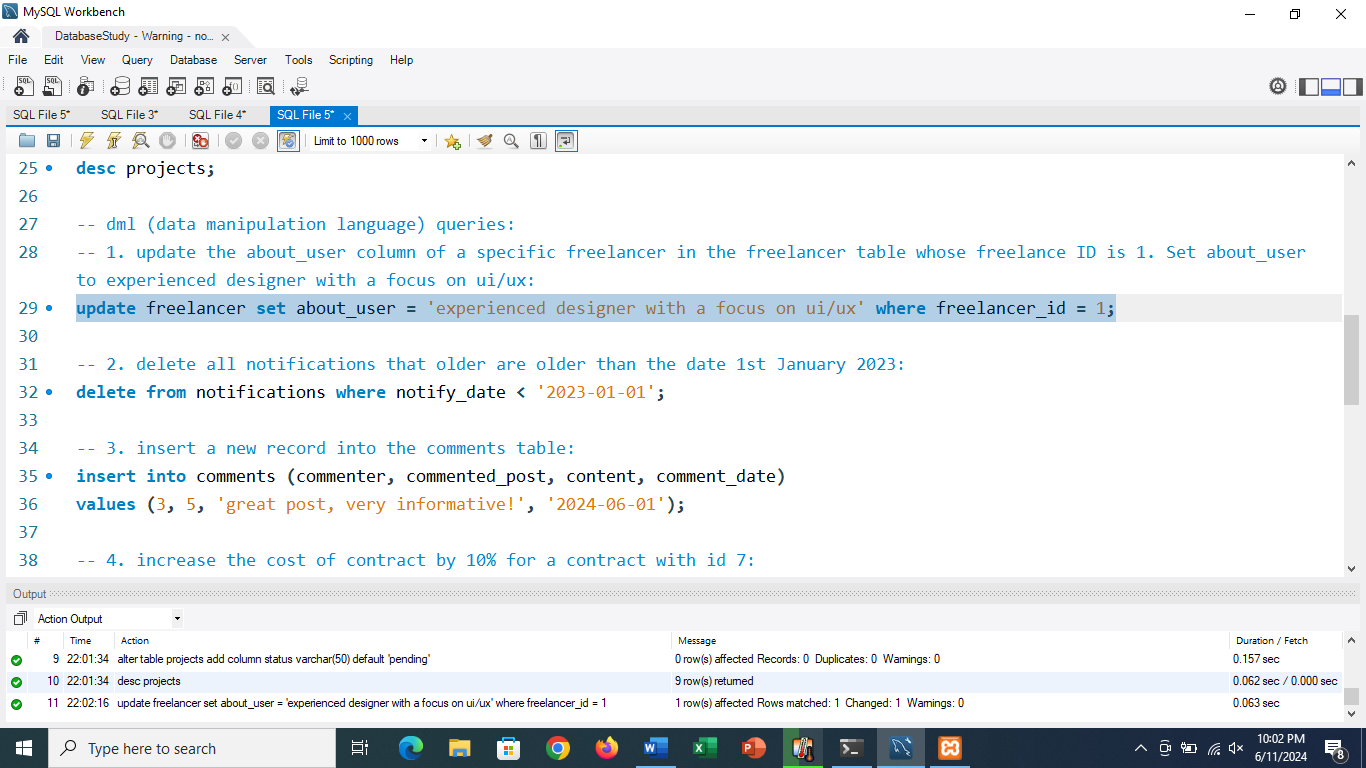
alter table projects add column status varchar(50) default 'pending';



**DML (data manipulation language) queries:**

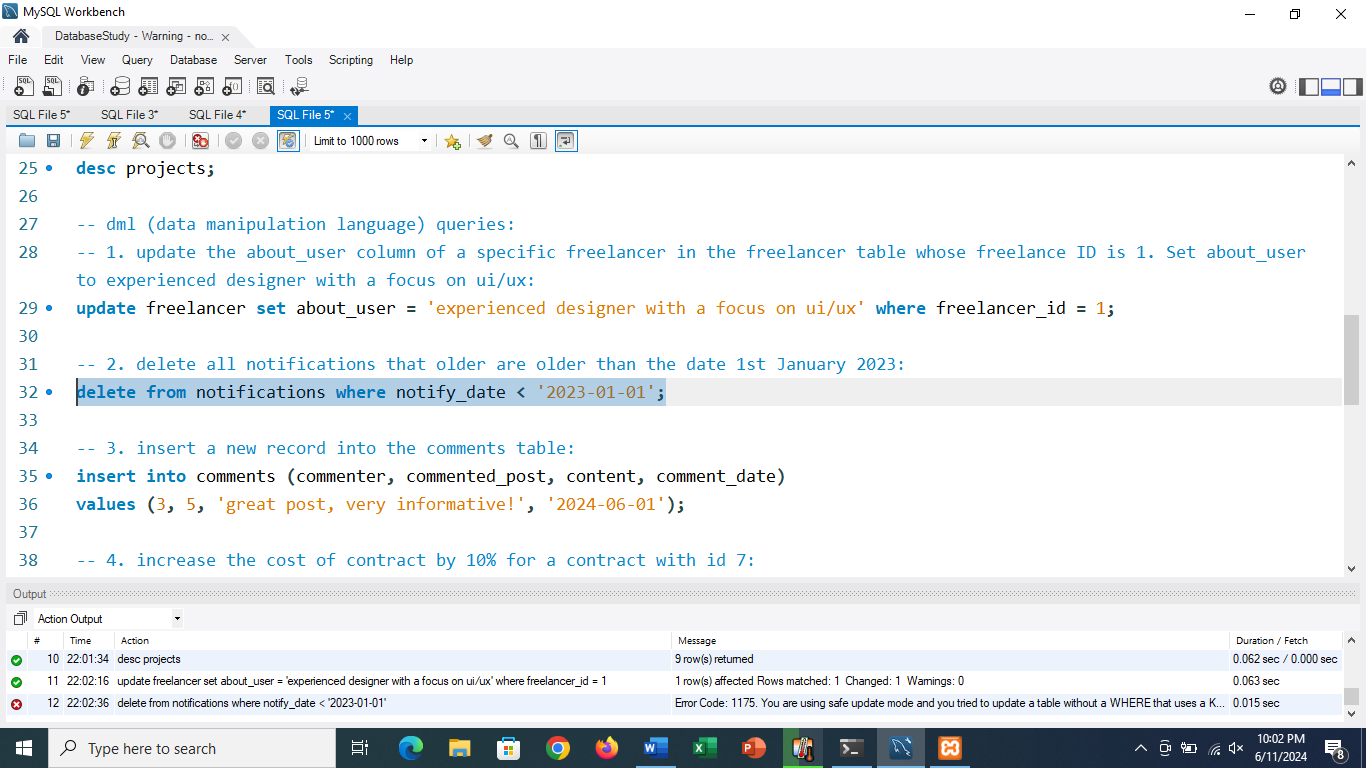
**1. update the about\_user column of a specific freelancer in the freelancer table whose freelance ID is 1. Set about\_user to experienced designer with a focus on ui/ux:**

update freelancer set about\_user = 'experienced designer with a focus on ui/ux' where freelancer\_id = 1;



**2. delete all notifications that older are older than the date 1st January 2023:**

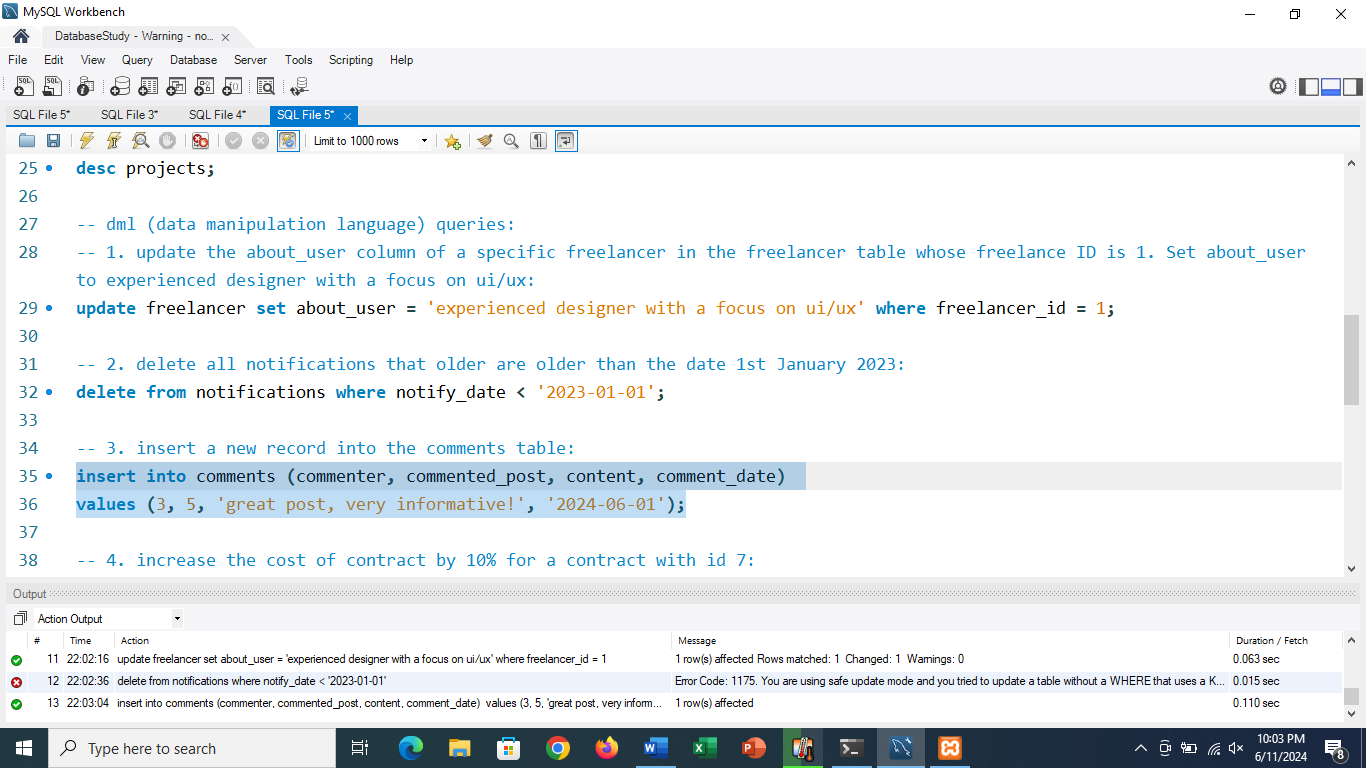
delete from notifications where notify\_date < '2023-01-01';



**3. insert a new record into the comments table:**

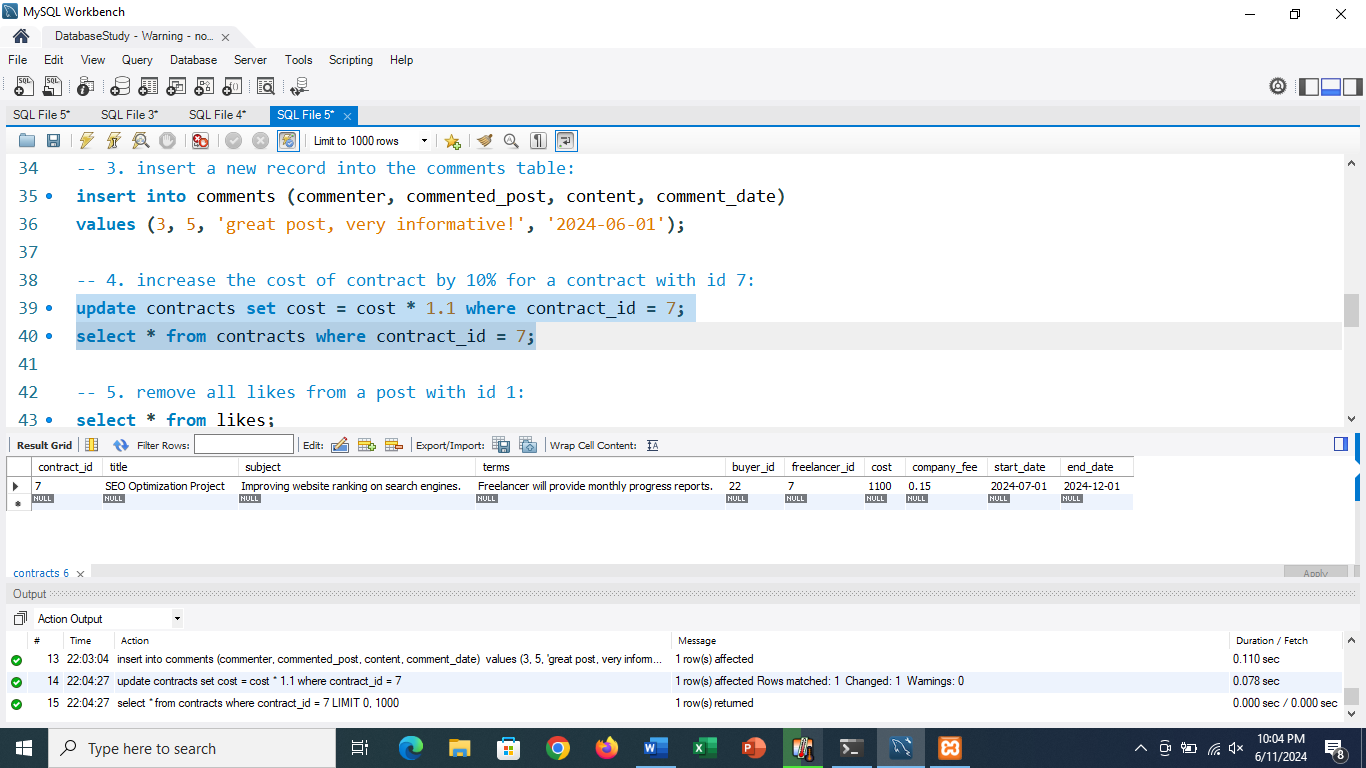
insert into comments (commenter, commented\_post, content, comment\_date)

values (3, 5, 'great post, very informative!', '2024-06-01');



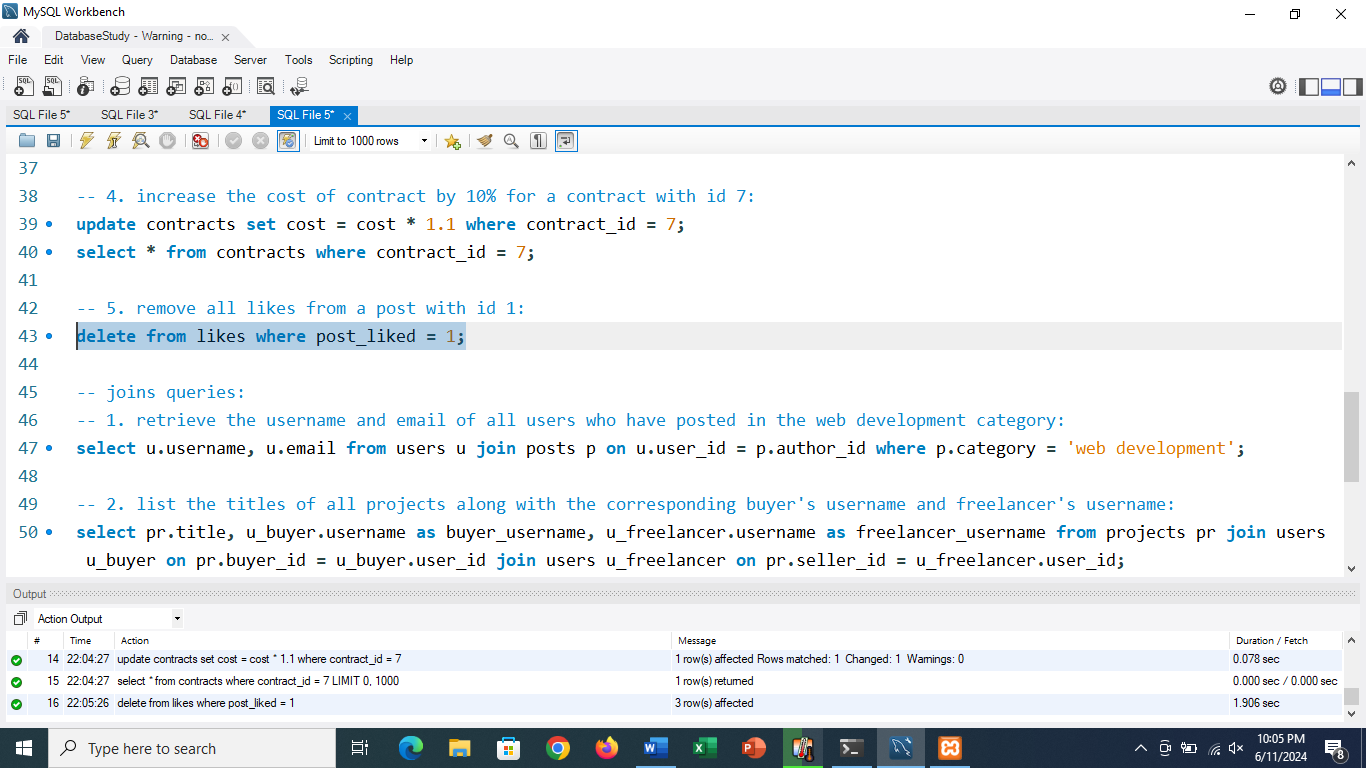
**4. increase the cost of contract by 10% for a contract with id 7:**

update contracts set cost = cost \* 1.1 where contract\_id = 7;



**5. remove all likes from a post with id 1:**

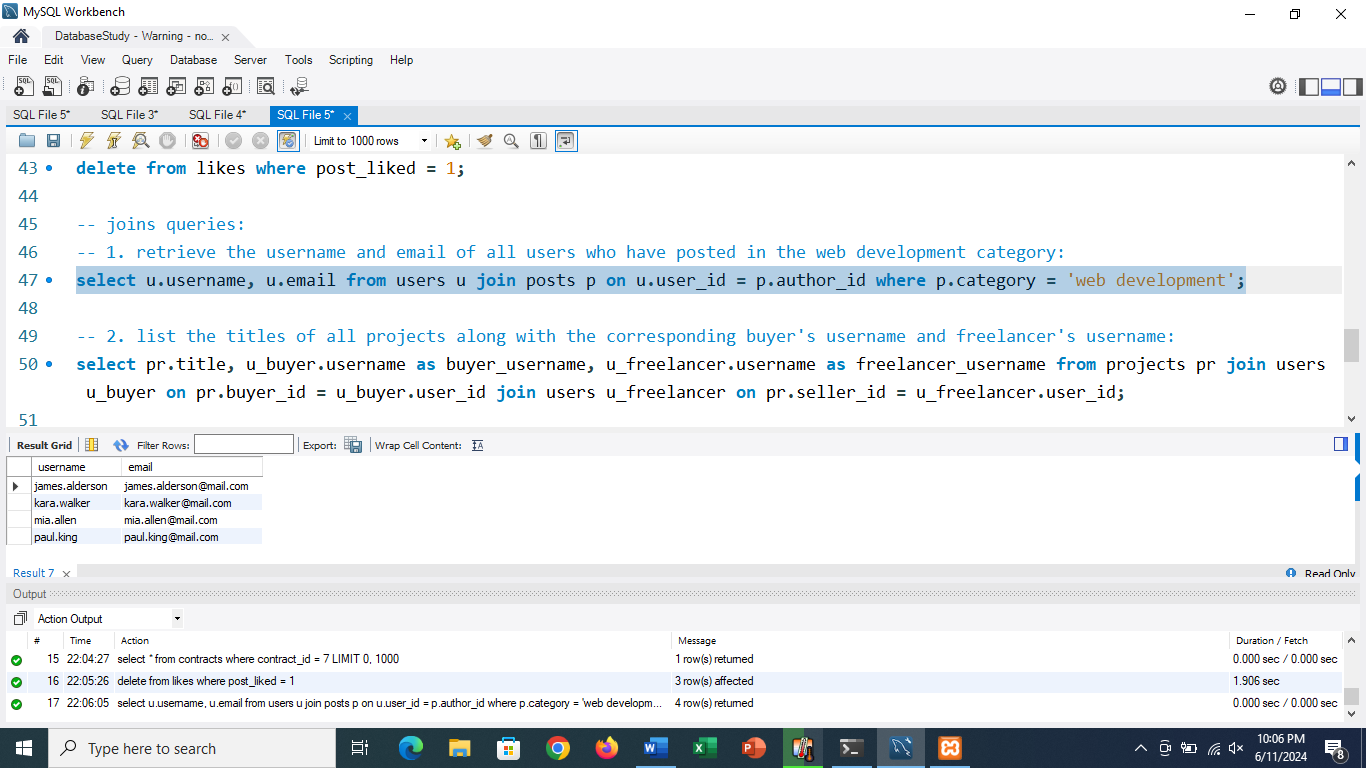
delete from likes where post\_liked = 1;



**joins queries:**

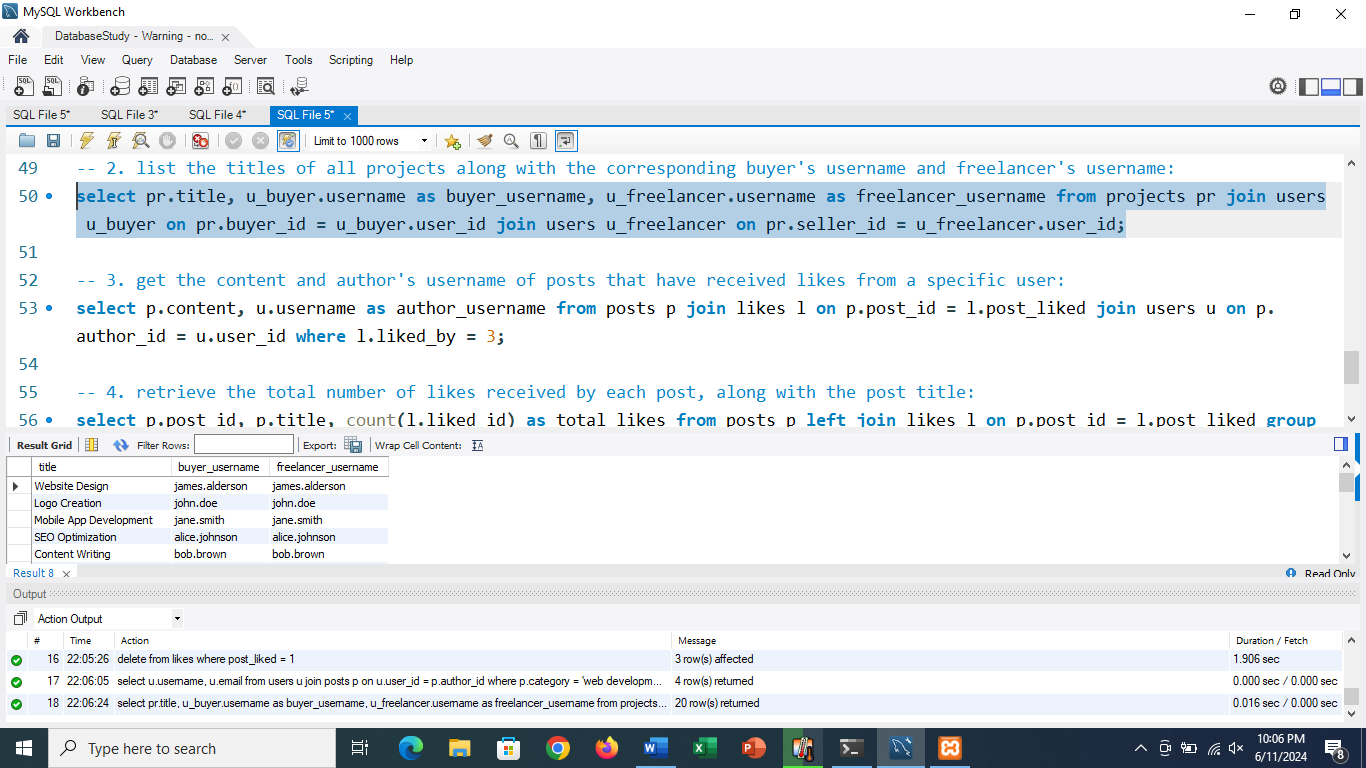
**1. retrieve the username and email of all users who have posted in the web development category:**

select u.username, u.email from users u join posts p on u.user\_id = p.author\_id where p.category = 'web development';



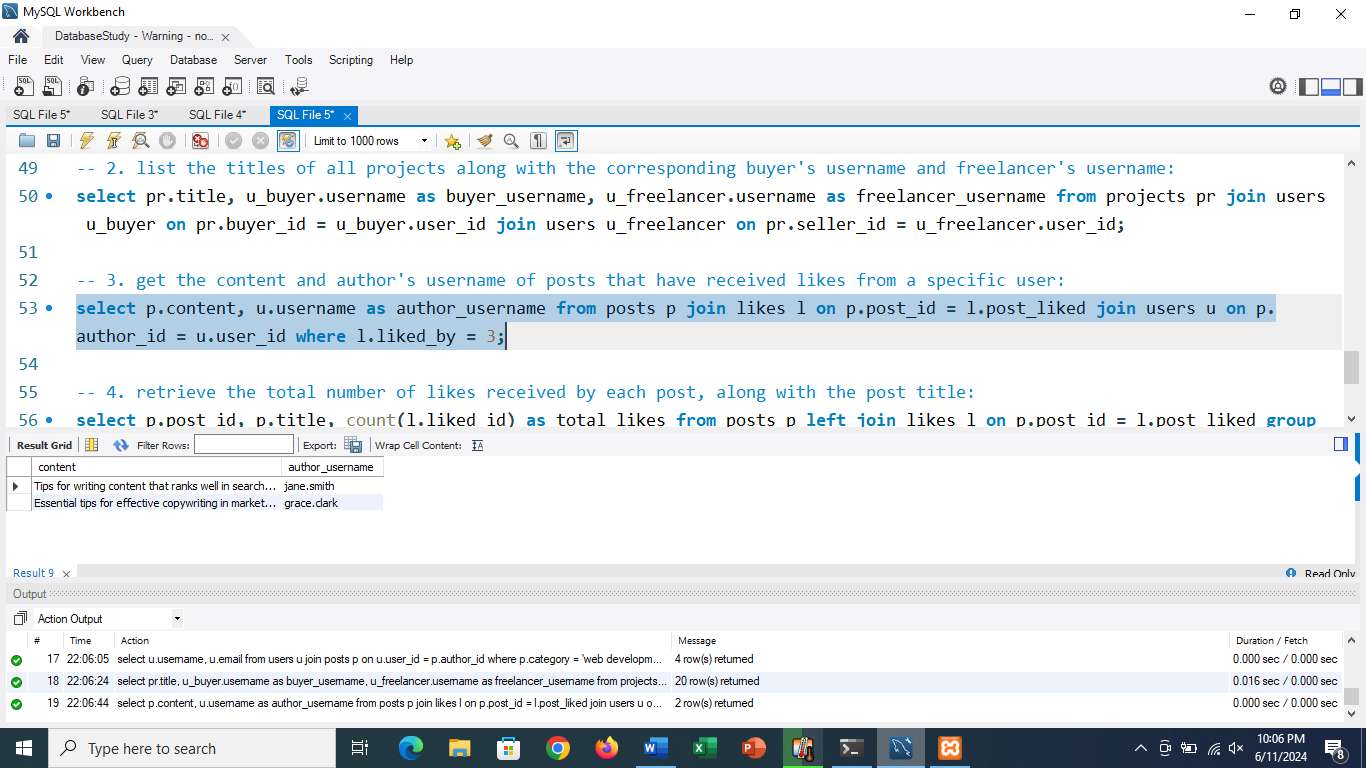
**2. list the titles of all projects along with the corresponding buyer's username and freelancer's username:**

select pr.title, u\_buyer.username as buyer\_username, u\_freelancer.username as freelancer\_username from projects pr join users u\_buyer on pr.buyer\_id = u\_buyer.user\_id join users u\_freelancer on pr.seller\_id = u\_freelancer.user\_id;



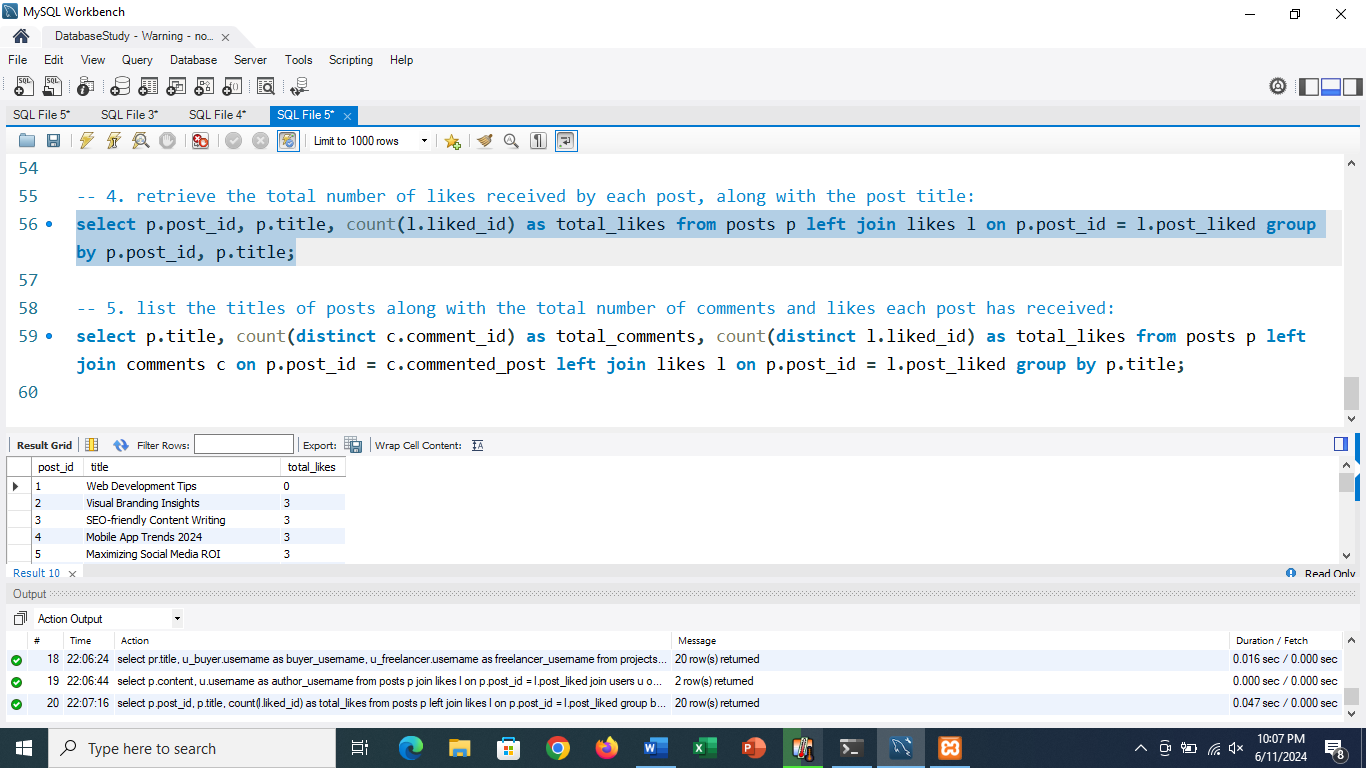
**3. get the content and author's username of posts that have received likes from a specific user:**

select p.content, u.username as author\_username from posts p join likes l on p.post\_id = l.post\_liked join users u on p.author\_id = u.user\_id where l.liked\_by = 3;



**4. retrieve the total number of likes received by each post, along with the post title:**

select p.post\_id, p.title, count(l.liked\_id) as total\_likes from posts p left join likes l on p.post\_id = l.post\_liked group by p.post\_id, p.title;



**5. list the titles of posts along with the total number of comments and likes each post has received:**

select p.title, count(distinct c.comment\_id) as total\_comments, count(distinct l.liked\_id) as total\_likes from posts p left join comments c on p.post\_id = c.commented\_post left join likes l on p.post\_id = l.post\_liked group by p.title;

