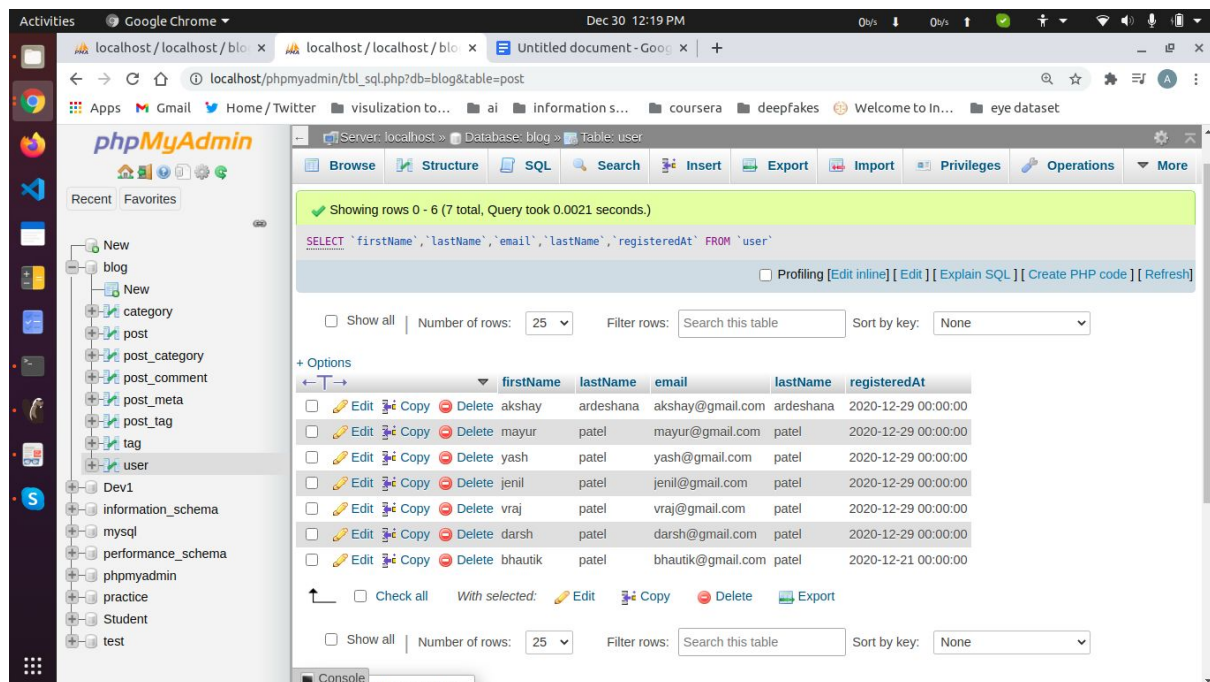
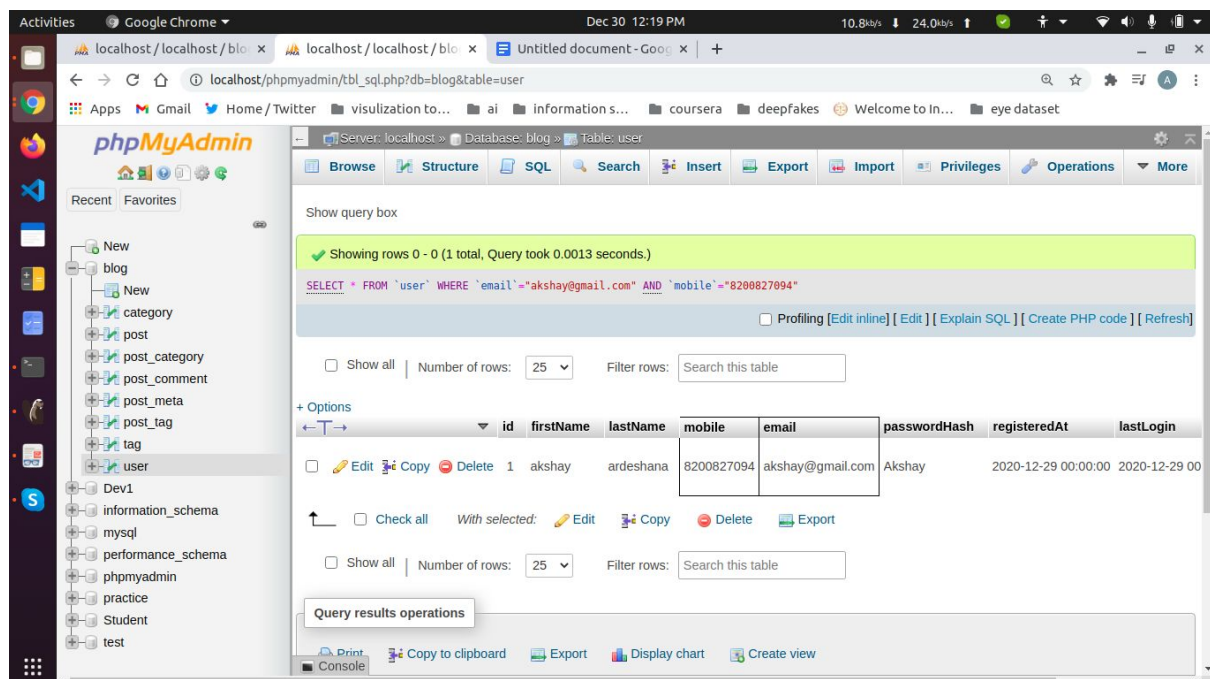


- 1)Get all the users with columns firstname, lastname, email, phone and registered at.
SELECT `firstName`,`lastName`,`email`,`lastName`,`registeredAt` FROM `user`



- 2)Get the user with a specific email and phone number.
SELECT * FROM `user` WHERE `email`="akshay@gmail.com" AND
`mobile`="8200827094"



- 3)Get all the users registered before 2 days. (Add the such records in table if those are not available)

`SELECT * FROM user WHERE `registeredAt` >= CURDATE() AND `registeredAt` < CURDATE() + INTERVAL 2 DAY`

4) Get all the users with more than 2 published posts available.

`select * from post group by authorId having count(*) > 1`

So is of :- `select * from post group by authorId having count(*) >= 1`

The screenshot shows the phpMyAdmin interface with the 'post' table selected. The query box contains the SQL: `select * from post group by authorId having count(*) >= 1`. The result shows 5 rows (0-4) with columns: id, authorId, parentId, title, metaTitle, slug, summary, published, createdAt, and updatedAt. The data is as follows:

id	authorId	parentId	title	metaTitle	slug	summary	published	createdAt	updatedAt
1	1	1	this is my first post	this is the metatitle regarding post	slug entry in the post	this is the summary	1	2020-12-29 15:43:00	2020-12-29 15:43:00
2	2	2	this is title in mayur post	this is meta title in mayur post	slug url in mayur post	this is summary in mayur post	1	2020-12-29 16:16:41	2020-12-29 16:16:41
3	3	3	this is title in yash post	this is meta title in yash post	slug url in yash post	this is summary in yash post	1	2020-12-29 16:16:41	2020-12-29 16:16:41

5) Get all the users who created posts from the last 2 days.

`SELECT * from user where id IN (SELECT authorId FROM post WHERE `createdAt` >= CURDATE() AND `createdAt` < CURDATE() + INTERVAL 2 DAY)`

6) Get all the users who do not have a post available.

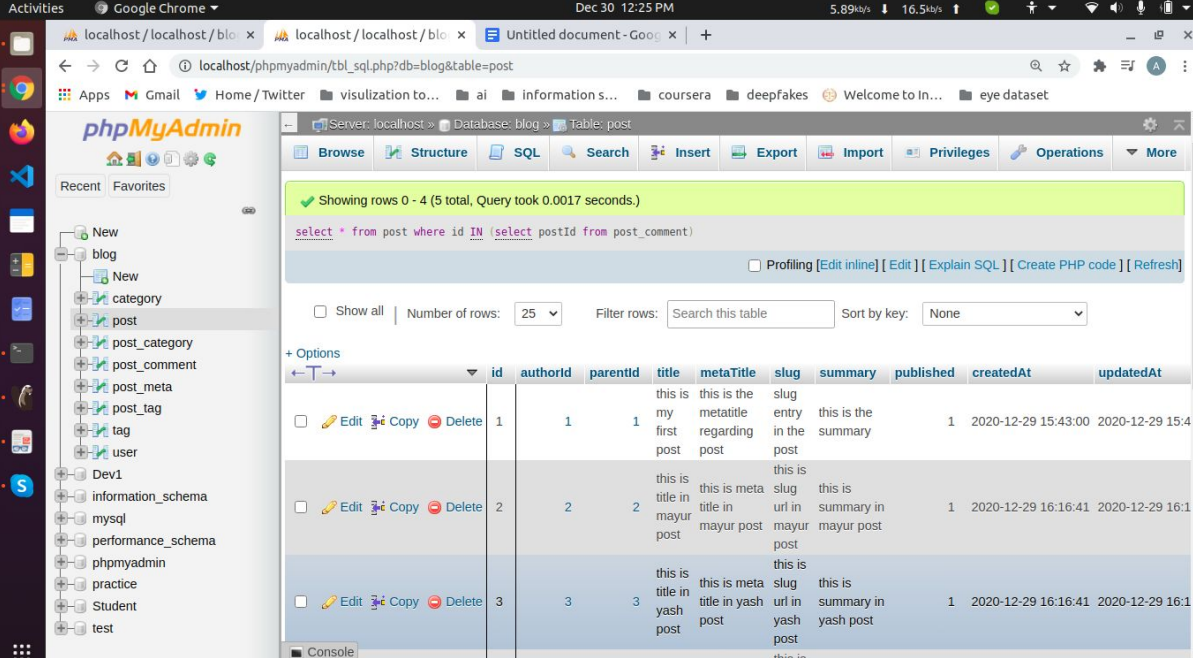
`SELECT * from user where id NOT IN (SELECT authorId FROM post)`

The screenshot shows the phpMyAdmin interface with the 'user' table selected. The query box contains the SQL: `SELECT * from user where id NOT IN (SELECT authorId FROM post)`. The result shows 2 rows (0-1) with columns: id, firstName, lastName, mobile, email, passwordHash, registeredAt, lastLogin, intro, and profile. The data is as follows:

id	firstName	lastName	mobile	email	passwordHash	registeredAt	lastLogin	intro	profile
6	darsh	patel	8200927194	darsh@gmail.com	darsh	2020-12-29 00:00:00	2020-12-29 00:00:00	hi this is darsh	this is darsh profile
7	bhautik	patel	8500627094	bhautik@gmail.com	bhautik	2020-12-21 00:00:00	2020-12-29 00:00:00	hi this is bhautik	this is bhautik profile

7) Get all the users where any of their posts have at least one post comment. Also list out the users who do not have any post comment for their any post.

=> `select * from post where id IN (select postId from post_comment);`



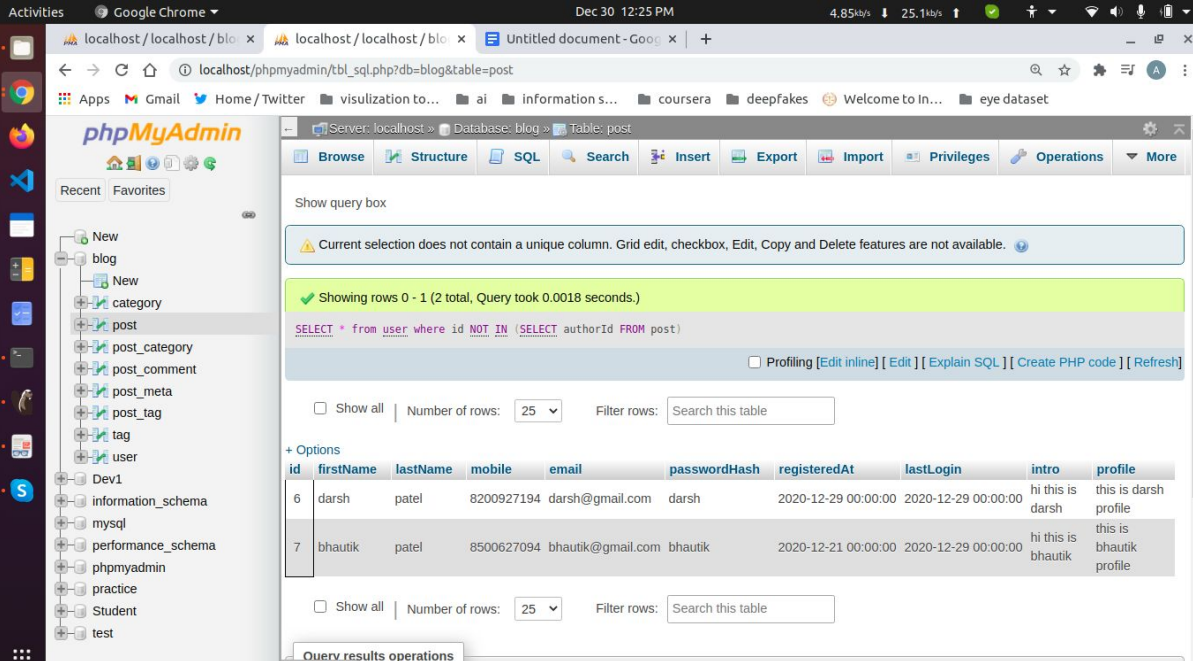
Showing rows 0 - 4 (5 total, Query took 0.0017 seconds.)

```
select * from post where id IN (select postId from post_comment)
```

Number of rows: 25 | Filter rows: Search this table | Sort by key: None

	id	authorId	parentId	title	metaTitle	slug	summary	published	createdAt	updatedAt
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	1	1	this is my first post	this is the metatitle regarding post	slug entry in the post	this is the summary	1	2020-12-29 15:43:00	2020-12-29 15:43:00
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	2	2	this is title in mayur post	this is meta title in mayur post	slug url in mayur post	this is summary in mayur post	1	2020-12-29 16:16:41	2020-12-29 16:16:41
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	3	3	this is title in yash post	this is meta title in yash post	slug url in yash post	this is summary in yash post	1	2020-12-29 16:16:41	2020-12-29 16:16:41

=> `select * from post where id NOT IN (select postId from post_comment);`



Showing rows 0 - 1 (2 total, Query took 0.0018 seconds.)

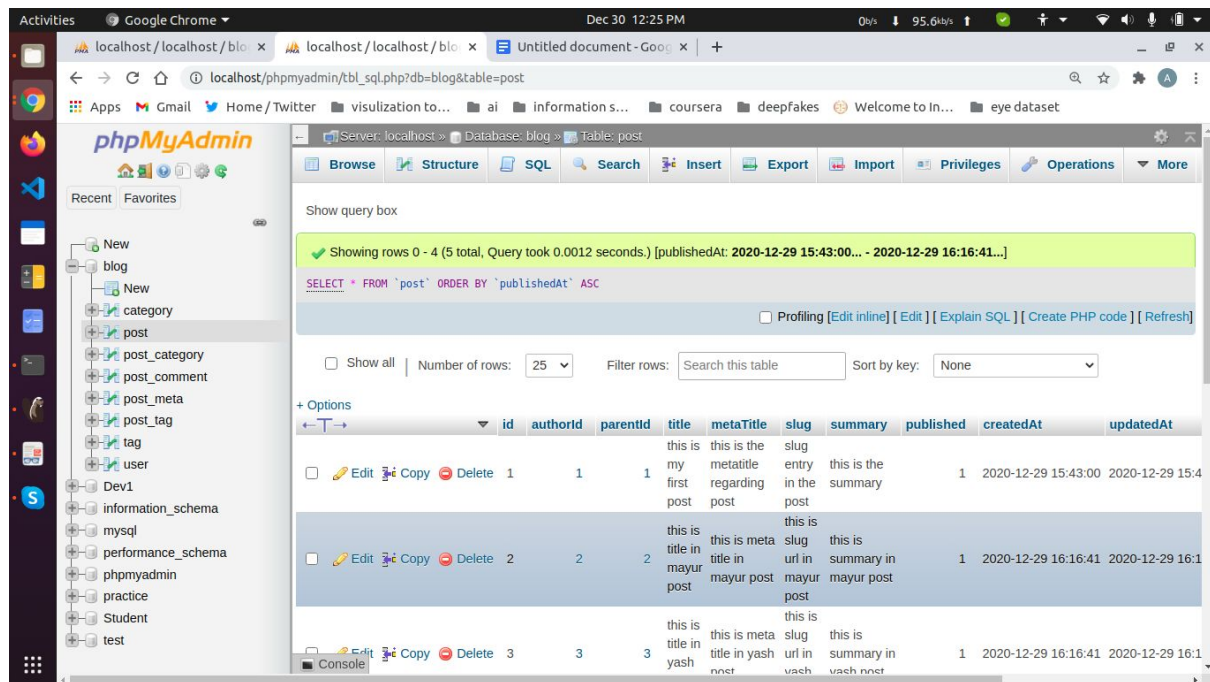
```
SELECT * from user where id NOT IN (SELECT authorId FROM post)
```

Number of rows: 25 | Filter rows: Search this table

	id	firstName	lastName	mobile	email	passwordHash	registeredAt	lastLogin	intro	profile
	6	darsh	patel	8200927194	darsh@gmail.com	darsh	2020-12-29 00:00:00	2020-12-29 00:00:00	hi this is darsh	this is darsh profile
	7	bhautik	patel	8500627094	bhautik@gmail.com	bhautik	2020-12-21 00:00:00	2020-12-29 00:00:00	hi this is bhautik	this is bhautik profile

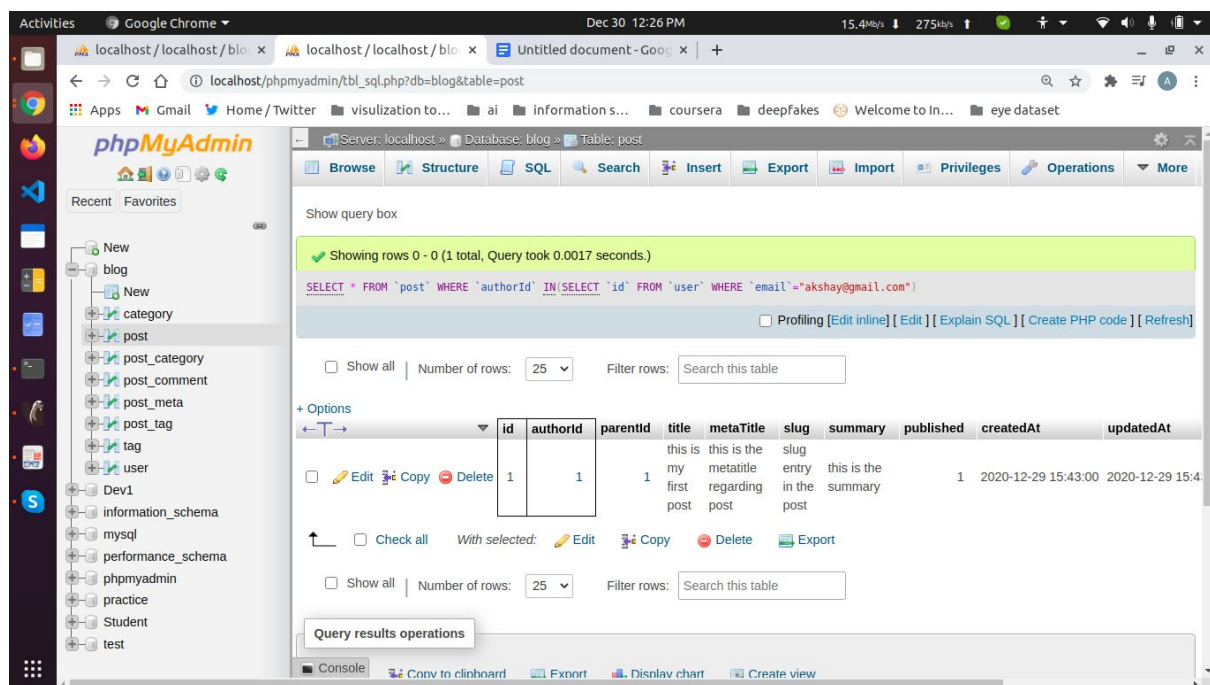
8) List out all the posts which are published and it should be sorted by created date.

`SELECT * FROM `post` ORDER BY `publishedAt` ASC;`



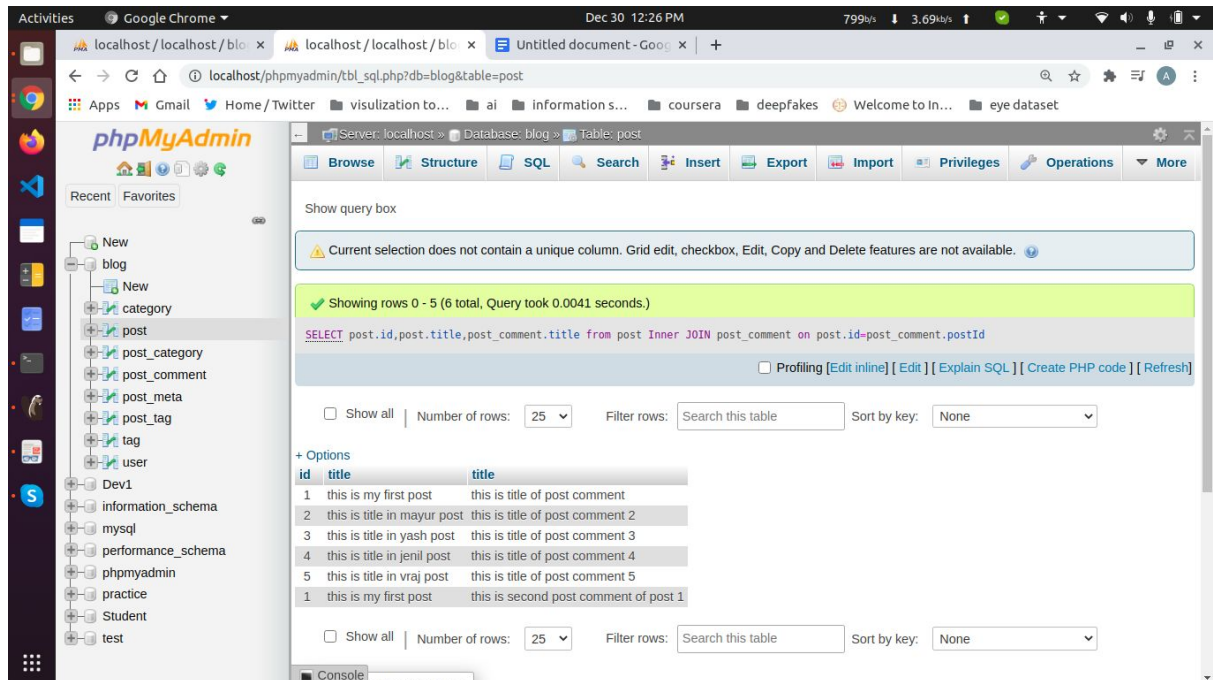
9) List out all the posts for any specific user id. (Try with specific email id as well)

```
SELECT * FROM `post` WHERE `authorId` IN (SELECT `id` FROM `user` WHERE `email`="akshay@gmail.com")
```



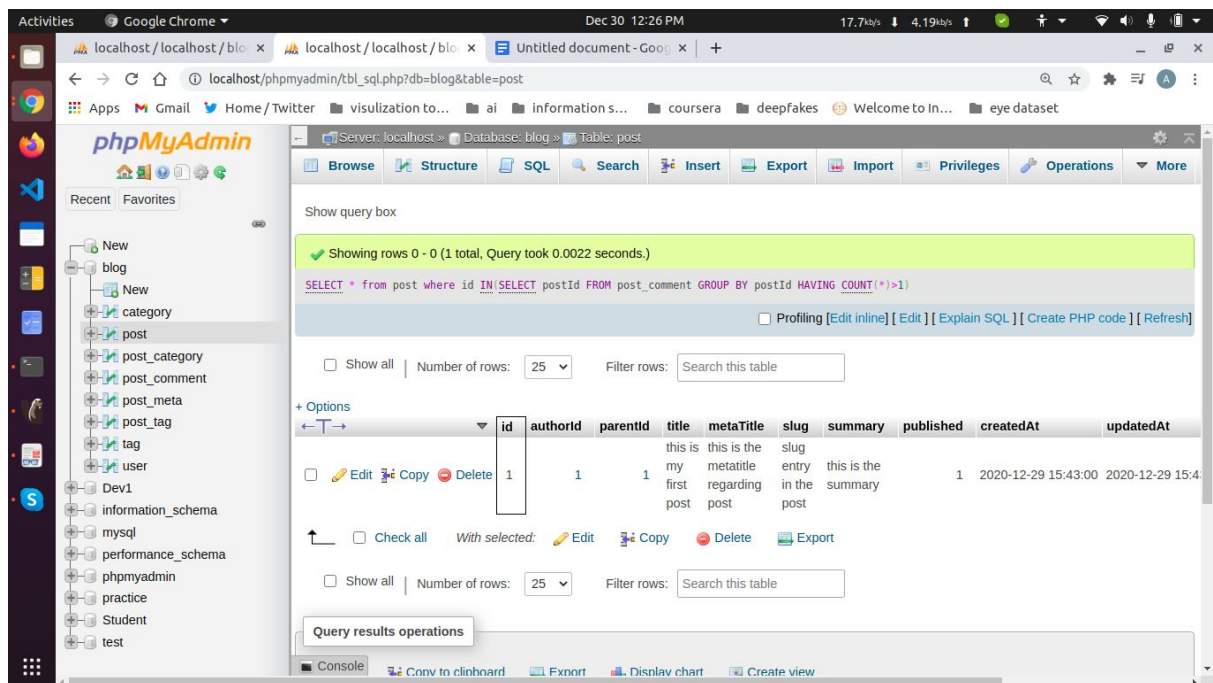
10) Get all the posts with their post comments

```
SELECT post.id, post.title, post_comment.title from post Inner JOIN post_comment on post.id=post_comment.postId;
```



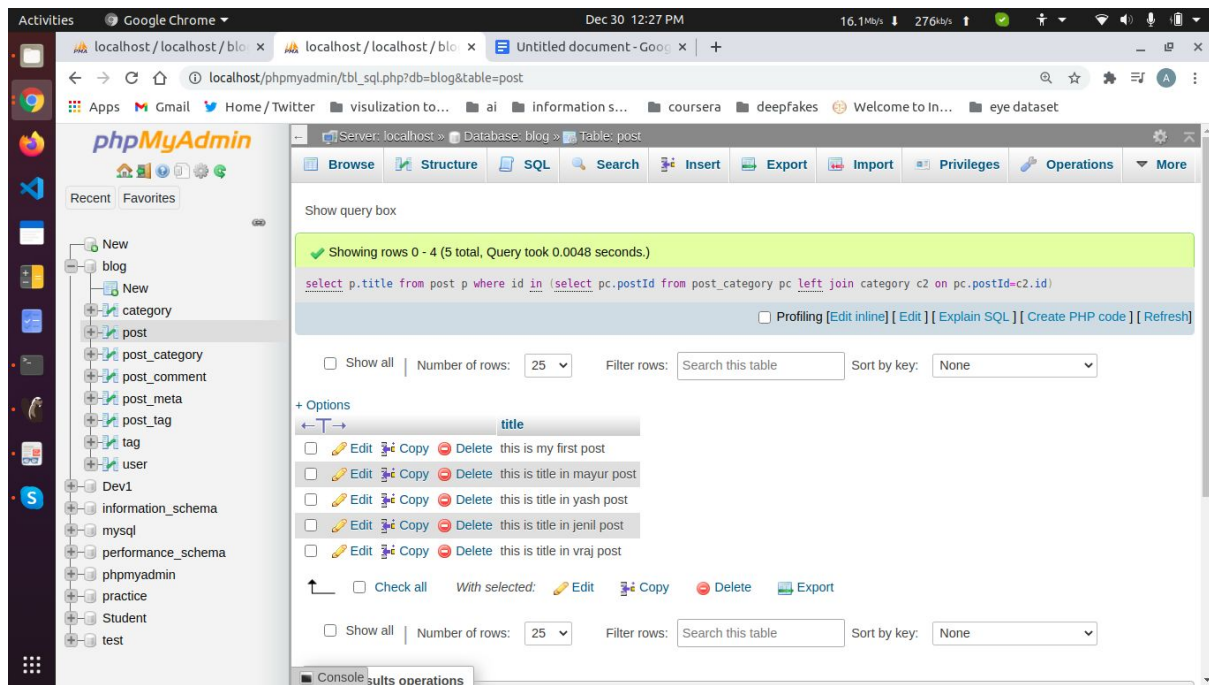
11) Get all the posts with minimum 2 post comments.

SELECT * from post where id IN(SELECT postId FROM post_comment GROUP BY postId HAVING COUNT(*)>1)

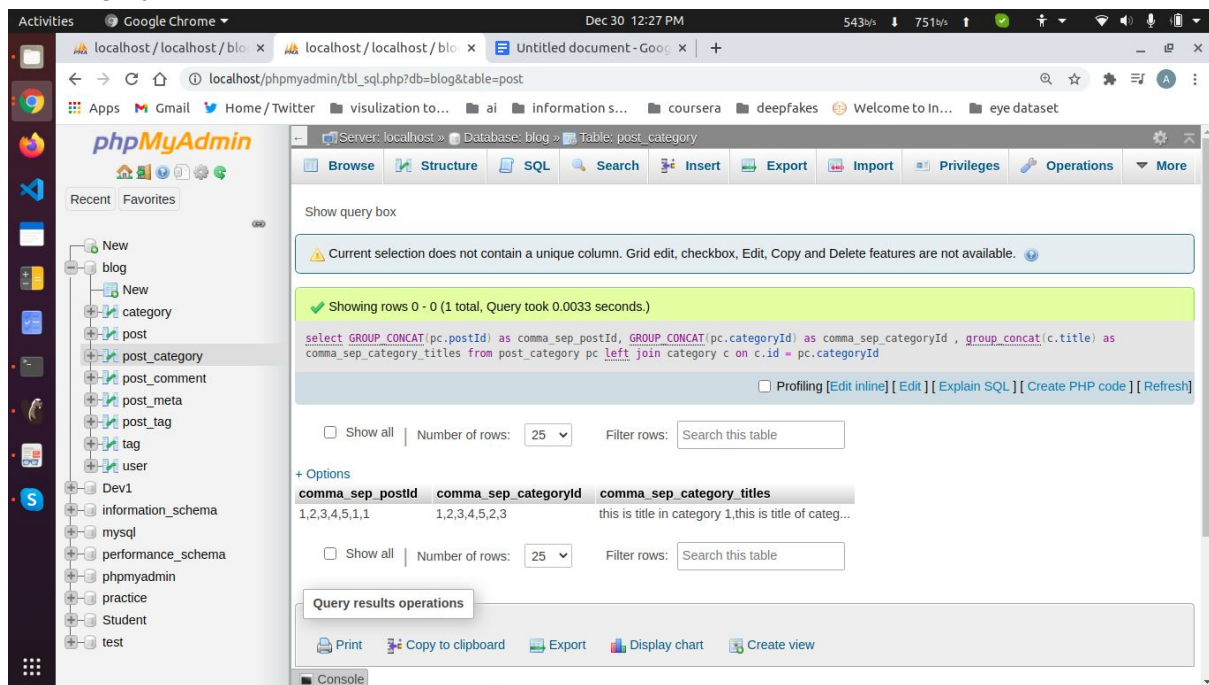


12) Get all the posts with their categories. Category columns should be displayed in result with comma separated category names if the same post is assigned TO multiple categories.

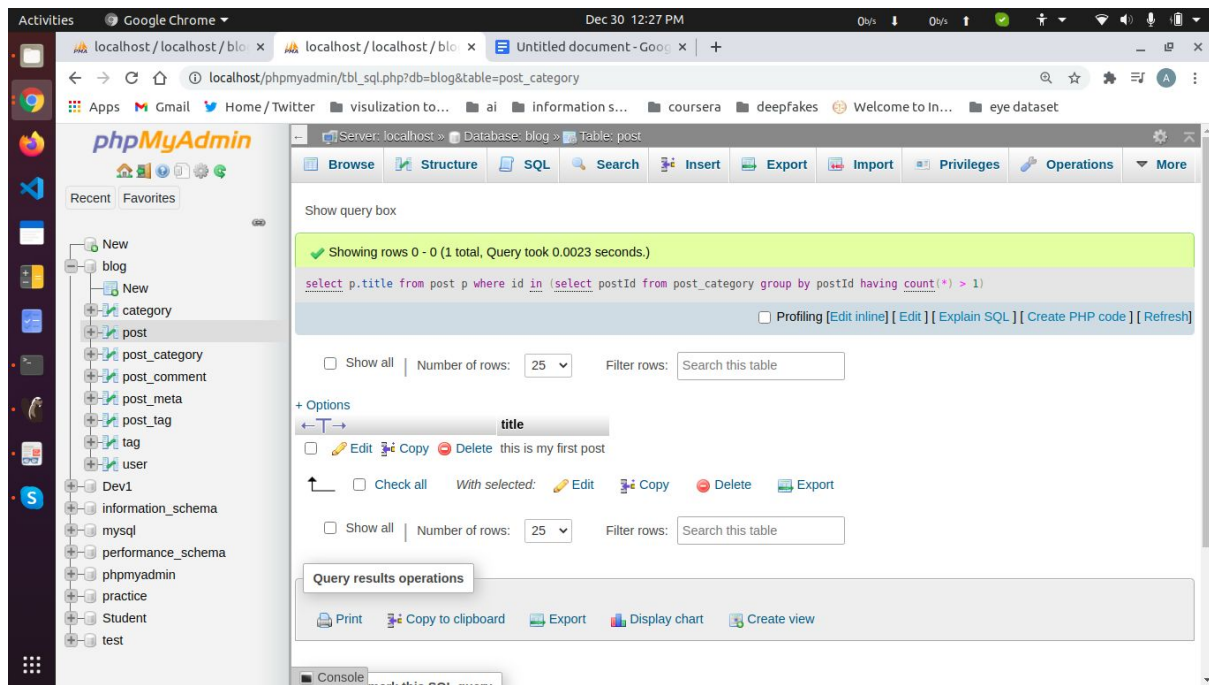
=> select p.title from post p where id in (select pc.postId from post_category pc left join category c2 on pc.postId=c2.id);



=>select GROUP_CONCAT(pc.postId) as comma_sep_postId,
GROUP_CONCAT(pc.categoryId) as comma_sep_categoryId , group_concat(c.title) as
comma_sep_category_titles from post_category pc left join category c on c.id =
pc.categoryId ;

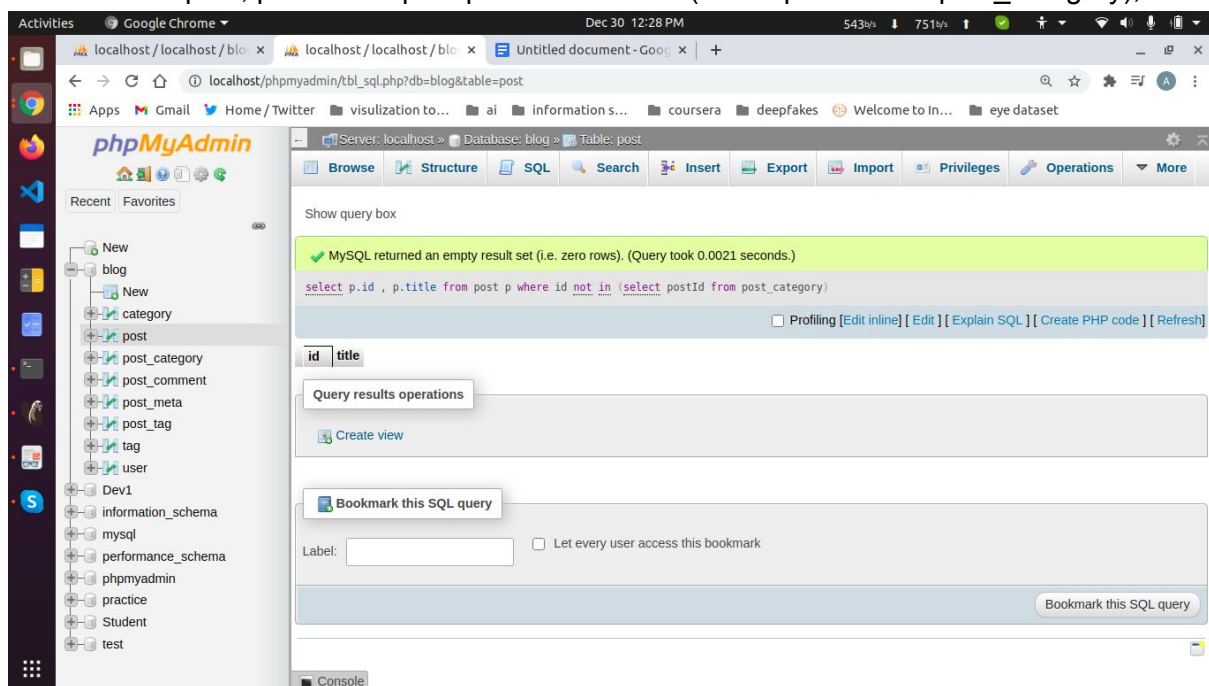


=>select p.title from post p where id in (select postId from post_category group by
postId having count(*) > 1);



13) Get all the posts which do not have any category assigned.

`select p.id , p.title from post p where id not in (select postId from post_category);`



14) List out all the categories with their post count. Only category and post count needs to be listed.

`SELECT `categoryId`,COUNT(*) FROM post_category GROUP BY `categoryId`;`

The screenshot shows the phpMyAdmin interface with the 'post_category' table selected. The query executed is: `SELECT 'categoryId', COUNT(*) FROM post_category GROUP BY 'categoryId'`. The result shows 5 rows, each representing a category and the count of posts assigned to it.

categoryid	COUNT(*)
1	1
2	2
3	2
4	1
5	1

15) List out all the categories which are having at least one post assigned.

`SELECT DISTINCT category.id, category.title, post_category.postId from category Inner JOIN post_category on category.id=post_category.postId`

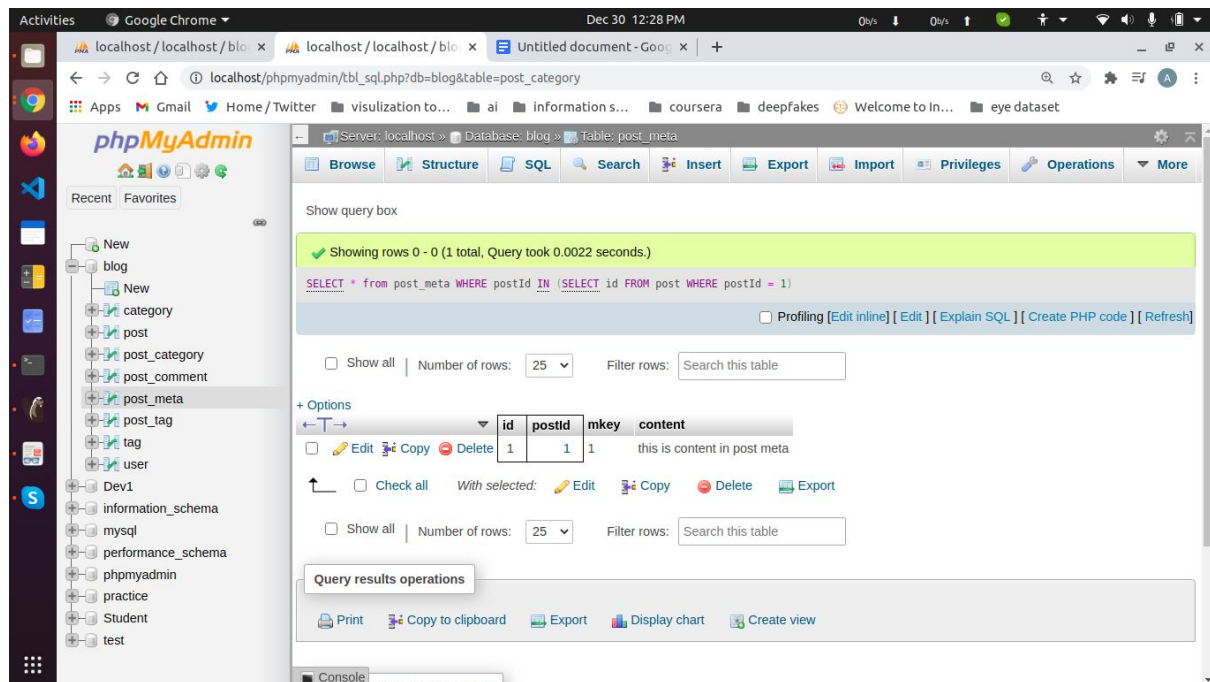
The screenshot shows the phpMyAdmin interface with the same query executed: `SELECT DISTINCT category.id, category.title, post_category.postId from category Inner JOIN post_category on category.id=post_category.postId`. The result shows 5 rows, each representing a category and the count of posts assigned to it.

id	title	postId
1	this is title in category 1	1
2	this is title of category 2	2
3	this is title of category 3	3
4	this is title of category 4	4
5	this is title of category 5	5

16) Get all the posts for the specific category. Category name and post title should be listed.

17) List out all the meta for the specific post.

`SELECT * from post_meta WHERE postId IN (SELECT id FROM post WHERE postId = 1);`



18) Get the user name who is having a specific meta key assigned to their posts.

`select firstName from user where id in (select id from post where id in (select postId from post_meta where mkey is not null));`

16 & 18 NOT able to understand