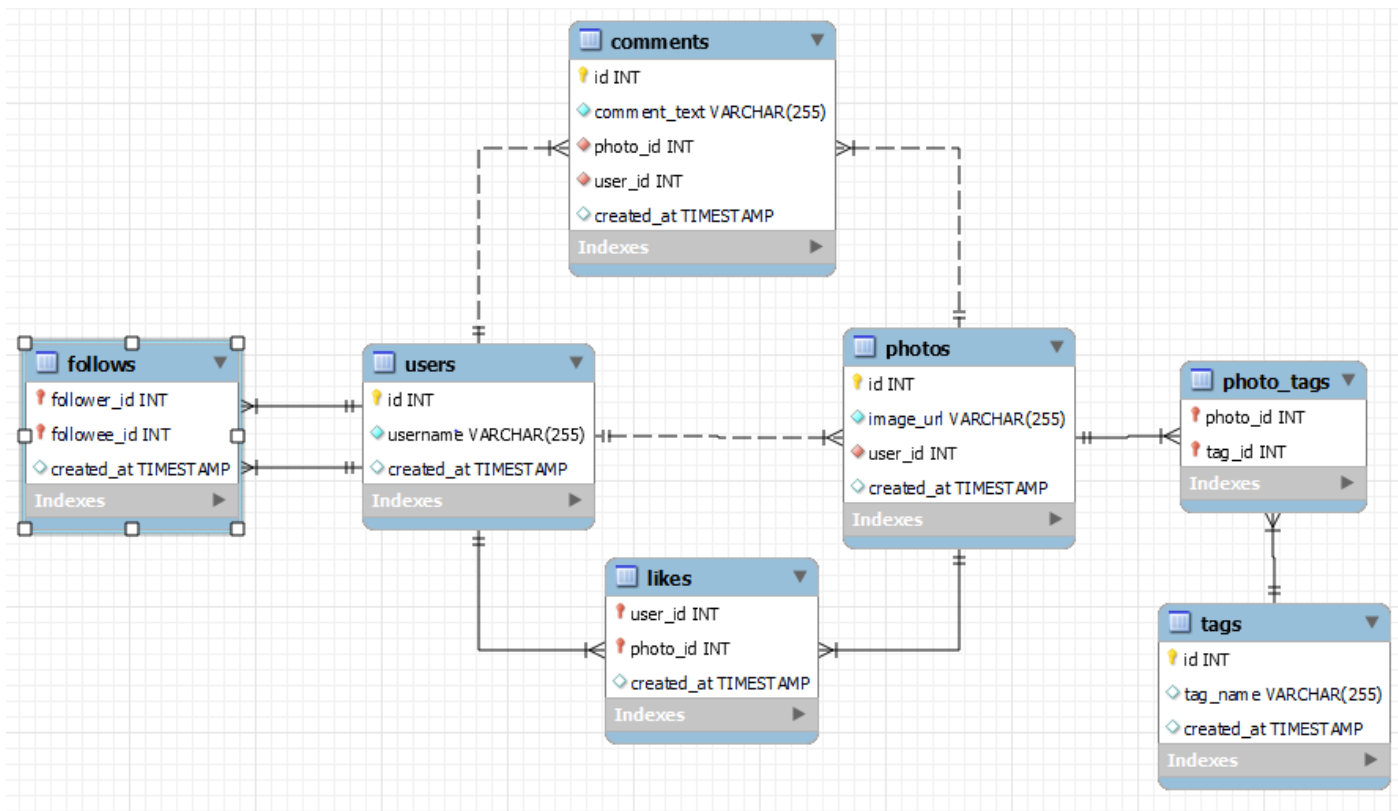


1. Create an ER diagram by using Workbench. Arrange the generated diagram so that no lines or boxes intersect. Refer to the Workbench note about how to create an ER diagram from database.



2. Write a query to find the least popular signup dates for users. The answer should be in Day, that is, Monday, Tuesday, ..., and the number of days. If there is a tie, show all tied days.

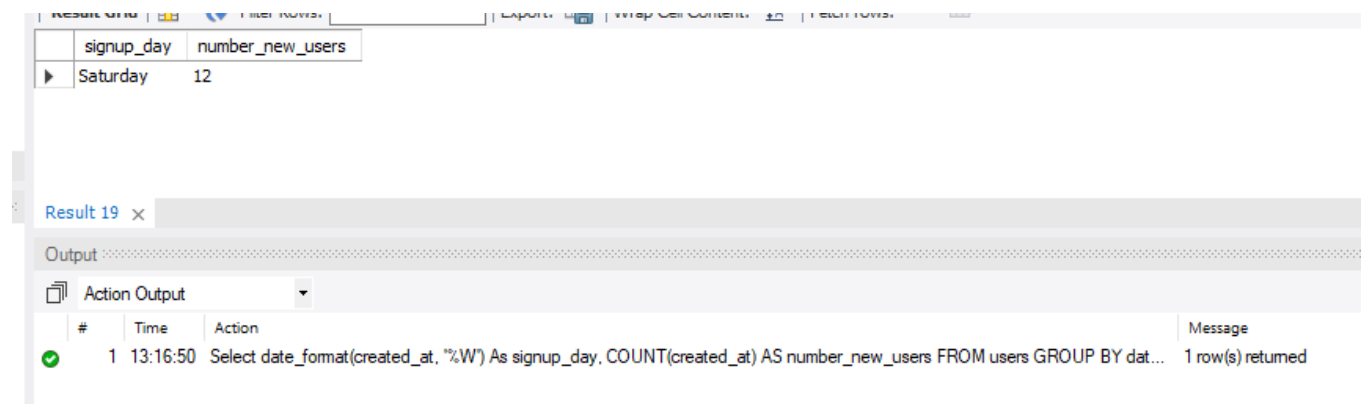
```
SELECT date_format(created_at, '%W') As signup_day, COUNT(created_at) AS number_new_users
```

```
FROM users
```

```
GROUP BY date_format(created_at, '%W')
```

```
ORDER BY COUNT(created_at)
```

```
LIMIT 1
```



	signup_day	number_new_users
▶	Saturday	12

#	Time	Action	Message
✓	1	13:16:50	Select date_format(created_at, \"%W\") As signup_day, COUNT(created_at) AS number_new_users FROM users GROUP BY dat... 1 row(s) returned

Sarah Mentel 3/14/2023

Homework 4

3. Write a query to find the users who have not posted any photos.

SELECT username

FROM users AS u LEFT JOIN photos As p

ON u.id = p.user_id

WHERE p.user_id is null

The screenshot displays a database interface with a list of usernames and a query execution log. The usernames listed are: Aniya_Hackett, Bartholome.Bernhard, Bethany20, Darby_Herzog, David.Osinski47, Duane60, Esmeralda.Mraz57, Esther.Zulauf61, Franco_Keebler64, Hulda.Macejkovic, Jadlyn81, Janelle.Nikolaus81, Jessyca_West, and Julian_Schmidt. Below the list, the query execution log shows the following details:

#	Time	Action	Message
1	13:36:48	Select username FROM users AS u LEFT JOIN photos As p ON u.id = p.user_id WHERE p.user_id is null LIMIT 0, 1000	26 row(s) returned

4. Write a query to find five most popular photos and users who created them. Most popular photos are those being liked most. Show the user id, user name, the photo URL, and the number of likes.

```
SELECT p.user_id, u.username, p.image_url, COUNT(*) AS num_of_likes
```

```
FROM photos AS p
```

```
JOIN likes AS l ON p.id = l.photo_id
```

```
JOIN users AS u ON p.user_id = u.id
```

```
GROUP BY p.image_url, p.user_id, u.username
```

```
ORDER BY num_of_likes DESC
```

```
LIMIT 5
```

	user_id	username	image_url	num_of_likes
▶	52	Zack_Kemmer93	https://jarret.name	48
	46	Malinda_Streich	https://celestine.name	43
	65	Adelle96	https://dorcas.biz	43
	44	Seth46	http://shannon.org	42
	20	Delpha.Kihn	https://dejon.name	41

Result 35 ×

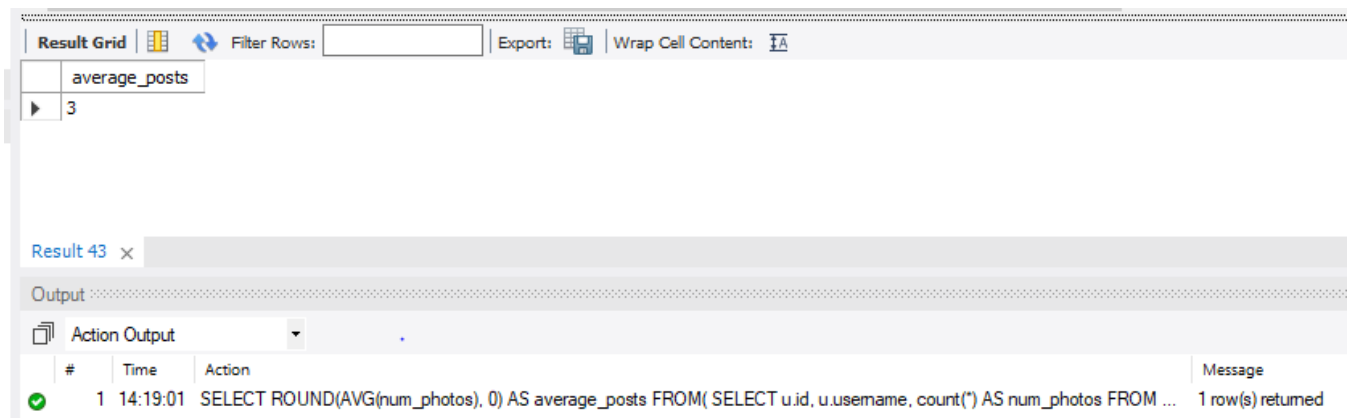
Output

Action Output

#	Time	Action	Message
✓ 1	14:00:53	SELECT p.user_id, u.username, p.image_url, COUNT(*) AS num_of_likes FROM photos AS p JOIN likes AS l ON p.id = l.photo_i...	5 row(s) returned

5. Write a query to find the average number of photos posted by users who posted at least one photo.

```
SELECT ROUND(AVG(num_photos), 0) AS average_posts
FROM(
SELECT u.id, u.username, count(*) AS num_photos
FROM users AS u
JOIN photos AS p ON u.id =p.user_id
GROUP BY u.id, u.username
HAVING num_photos >= 1
) AS COUNT
```



The screenshot shows a database query interface. At the top, there's a toolbar with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. Below this, a table displays the query result. The table has two columns: 'average_posts' and a value '3'. Below the table, there's a section labeled 'Result 43' with a close button. Underneath, there's an 'Output' section with a dropdown menu set to 'Action Output'. At the bottom, there's a log table with columns '#', 'Time', 'Action', and 'Message'. The log shows a successful execution of the query at 14:19:01, returning 1 row(s).

	average_posts
▶	3

#	Time	Action	Message
✓ 1	14:19:01	SELECT ROUND(AVG(num_photos), 0) AS average_posts FROM(SELECT u.id, u.username, count(*) AS num_photos FROM ...	1 row(s) returned

6. Write a query to find the most followed users and the most following users. The answer should like the following.

SELECT COUNT(follower_id) AS follower_count, followee_id, users.username

FROM follows

JOIN users ON follows.followee_id = users.id

GROUP BY followee_id

ORDER BY follower_count DESC

The screenshot shows a database interface with a 'Result Grid' tab. The query results are displayed in a table with columns: follower_count, followee_id, and username. The results are ordered by follower_count in descending order. The first 8 rows are highlighted in blue. Below the table, there is a 'Result 123' label and an 'Output' section. The 'Output' section shows the query text and a message indicating that 100 row(s) were returned.

	follower_count	followee_id	username
▶	77	88	Clint27
	77	68	Franco_Keebler64
	77	29	Jaime53
	77	80	Darby_Herzog
	77	74	Hulda.Macejkovic
	77	90	Esmeralda.Mraz57
	77	45	David.Osinski47

Result 123 ×

Output

Action Output

#	Time	Action	Message
✓ 1	19:26:09	SELECT COUNT(follower_id) AS follower_count, followee_id, users.username FROM follows JOIN users ON follows.followee_id = users.id GROUP BY followee_id ORDER BY follower_count DESC	100 row(s) returned

SELECT COUNT(followee_id) AS followee_count, follower_id, users.username

FROM follows

JOIN users ON follows.follower_id = users.id

GROUP BY follower_id

ORDER BY followee_count DESC

Sarah Mentel 3/14/2023

Homework 4



The screenshot shows a database interface with a 'Result Grid' at the top. It contains a table with three columns: 'followee_count', 'follower_id', and 'username'. There are 7 rows of data. Below the grid, there is an 'Output' section with a tab labeled 'Action Output'. It shows a single log entry with a green checkmark, indicating a successful query execution. The log entry includes the time '19:27:13' and the SQL query: 'SELECT COUNT(followee_id) AS followee_count, follower_id, users.username FROM follows JOIN users ON follows.follower_id ...'. The message at the end of the log entry states '77 row(s) returned'.

	followee_count	follower_id	username
▶	99	65	Adelle96
	99	31	Aiyana_Hoeger
	99	99	Alek_Watsica
	99	13	Alexandro35
	99	84	Alysa22
	99	2	Andre_Purdy85
	99	5	Aniya_Hackett

Result 125 x

Output

Action Output

#	Time	Action	Message
✓ 1	19:27:13	SELECT COUNT(followee_id) AS followee_count, follower_id, users.username FROM follows JOIN users ON follows.follower_id ...	77 row(s) returned

7. Write a query to find the user who has posted most photos, and the photos' ids and the tag names. Show the user name, photo id, and tag name.

```
SELECT u.username, pt.photo_id, t.tag_name
FROM users AS u
JOIN photos AS p ON u.id = p.user_id
JOIN photo_tags AS pt ON p.id = pt.photo_id
JOIN tags AS t ON pt.tag_id = t.id
WHERE username =
(SELECT u.username
FROM users AS u
JOIN photos AS p ON u.id = p.user_id
GROUP BY username
ORDER BY COUNT(*) DESC
LIMIT 1)
GROUP BY u.username, pt.photo_id, t.tag_name;
```

Sarah Mentel 3/14/2023

Homework 4

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
username	photo_id	tag_name				
Eveline95	63	foodie				
Eveline95	64	lol				
Eveline95	64	happy				
Eveline95	64	fun				
Eveline95	64	smile				
Eveline95	65	food				
Eveline95	65	delicious				
Eveline95	66	hair				
Eveline95	66	party				
Eveline95	66	...				

Result 115 x

Output

Action Output

#	Time	Action	Message
✓ 1	19:11:18	SELECT u.username, pt.photo_id, t.tag_name FROM users AS u JOIN photos AS p ON u.id = p.user_id JOIN photo_tags AS ...	24 row(s) returned

8. Write a query to find the five most popular tag names and the total number of tags for each tag name. Show the tag id, tag name, and the number of tags.

SELECT id, tag_name, COUNT(id) as tag_count

FROM tags

JOIN photo_tags AS pt ON tags.id = pt.tag_id

GROUP BY id

ORDER BY COUNT(id) DESC

LIMIT 5

Result Grid				Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
id	tag_name	tag_count					
21	smile	59					
20	beach	42					
17	party	39					
13	fun	38					
18	concert	24					

Result 82 x

Output

Action Output

#	Time	Action	Message
✓ 1	18:37:53	SELECT id, tag_name, COUNT(id) as tag_count FROM tags JOIN photo_tags AS pt ON tags.id = pt.tag_id GROUP BY id ORD...	5 row(s) returned

Sarah Mentel 3/14/2023

Homework 4

9. Bots (or fake user ids) can be created and the bots like every photo being posted. Write a query to find such bots. Show the bot user names and user ids.

```
SELECT user_id, u.username
```

```
FROM likes
```

```
JOIN users AS u ON likes.user_id = u.id
```

```
GROUP BY user_id
```

```
HAVING COUNT(*) = (SELECT COUNT(*) FROM photos);
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	user_id	username
▶	5	Aniya_Hackett
	14	Jadyn81
	21	Rocio33
	24	Maxwell.Halvorson
	36	Ollie_Ledner37
	41	Mckenna17
	54	Duane60
	57	Julien_Schmidt
	66	Mike.Auer39
	71	Nia_Haag
	75	Leslie67

Result 87

×

Output

Action Output

#	Time	Action	Message
✓ 1	18:45:04	SELECT user_id, u.username FROM likes JOIN users AS u ON likes.user_id = u.id GROUP BY user_id HAVING COUNT(*) = (S...	13 row(s) returned

Sarah Mentel 3/14/2023

Homework 4

10. With the bot accounts removed, show the result of #6.

```
SELECT COUNT(follower_id) AS follower_count, followee_id, users.username
FROM follows
JOIN users ON follows.followee_id = users.id
WHERE followee_id AND follower_id NOT IN (SELECT user_id
FROM likes
JOIN users AS u ON likes.user_id = u.id
GROUP BY user_id
HAVING COUNT(*) = (SELECT COUNT(*) FROM photos))
GROUP BY followee_id
ORDER BY follower_count DESC;
```

Result Grid			
Filter Rows:		Export:	Wrap Cell Content:
follower_count	followee_id	username	
64	91	Bethany20	
64	64	Florence99	
64	45	David.Osinski47	
64	86	Delfina_VonRueden68	
64	68	Franco_Keebler64	
64	29	Jaime53	
64	5	Aniya_Hackett	

Result 128 x

Output

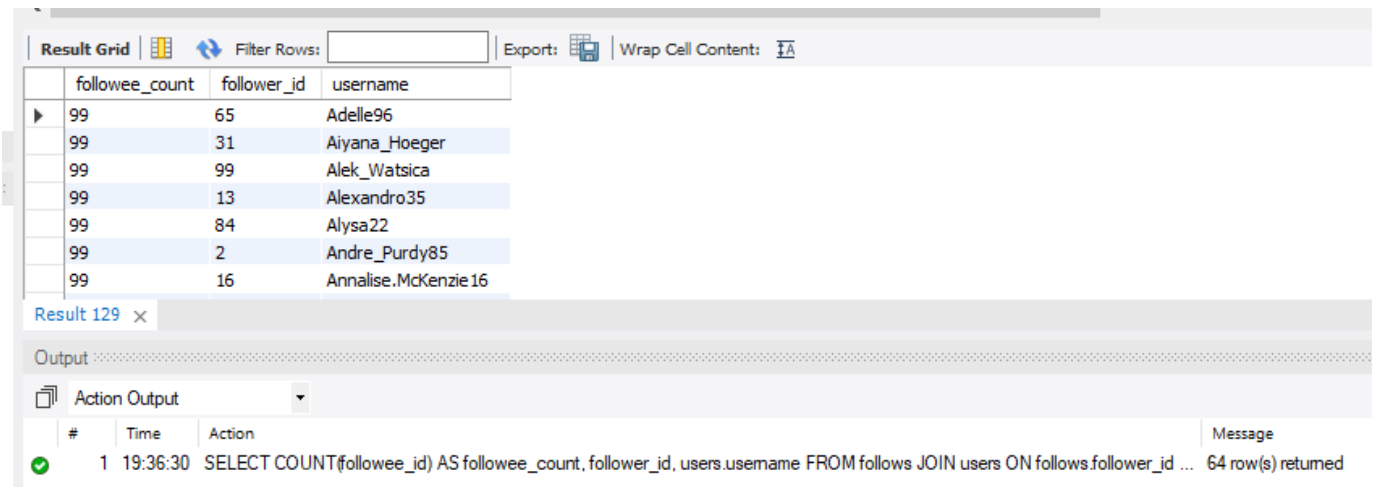
Action Output

#	Time	Action	Message
1	19:35:39	SELECT COUNT(follower_id) AS follower_count, followee_id, users.username FROM follows JOIN users ON follows.followee_id...	100 row(s) returned

Sarah Mentel 3/14/2023

Homework 4

```
SELECT COUNT(followee_id) AS followee_count, follower_id, users.username
FROM follows
JOIN users ON follows.follower_id = users.id
WHERE followee_id AND follower_id NOT IN (SELECT user_id
FROM likes
JOIN users AS u ON likes.user_id = u.id
GROUP BY user_id
HAVING COUNT(*) = (SELECT COUNT(*) FROM photos))
GROUP BY follower_id
ORDER BY followee_count DESC;
```



The screenshot shows a database query result grid with the following columns: followee_count, follower_id, and username. The results are ordered by followee_count in descending order. The first row shows a followee_count of 99 for follower_id 65 (Adelle96). The second row shows a followee_count of 99 for follower_id 31 (Aiyana_Hoeger). The third row shows a followee_count of 99 for follower_id 99 (Alek_Watsica). The fourth row shows a followee_count of 99 for follower_id 13 (Alexandro35). The fifth row shows a followee_count of 99 for follower_id 84 (Alysa22). The sixth row shows a followee_count of 99 for follower_id 2 (Andre_Purdy85). The seventh row shows a followee_count of 99 for follower_id 16 (Annalise.McKenzie16).

	followee_count	follower_id	username
▶	99	65	Adelle96
	99	31	Aiyana_Hoeger
	99	99	Alek_Watsica
	99	13	Alexandro35
	99	84	Alysa22
	99	2	Andre_Purdy85
	99	16	Annalise.McKenzie16

Result 129 x

Output

Action Output

#	Time	Action	Message
✓ 1	19:36:30	SELECT COUNT(followee_id) AS followee_count, follower_id, users.username FROM follows JOIN users ON follows.follower_id ...	64 row(s) returned

All the query problems should be written with one query. Subqueries or Unions are allowed, but multiple queries will not get full credits.

Provide all query answers in TEXT and the result in images in Word file. For the ER diagram, capture the image and put in in Word file. Submit the Word file.