

## INSTAGRAM USER ANALYTICS

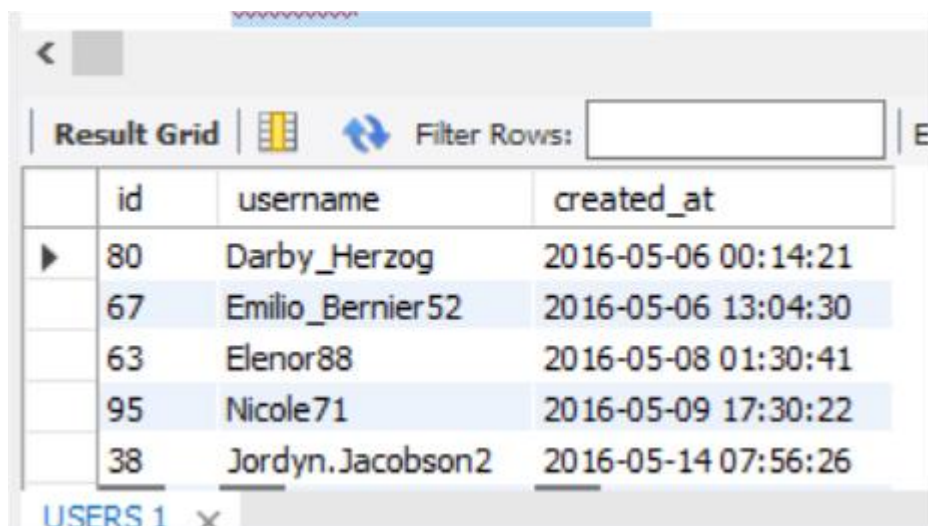
### Marketing Analysis:

#### 1. Loyal user reward:

##### INPUT:

```
select * from users
order by created_at asc
limit 5
```

##### OUTPUT



The screenshot shows a database interface with a 'Result Grid' tab. It displays the results of a SQL query. The grid has columns for 'id', 'username', and 'created\_at'. There are five rows of data, each representing a user. The first row is highlighted with a blue background. Below the grid, there is a tab labeled 'USERS 1' with a close button (X).

	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

USERS 1 X

#### 2. Who Have never Photo on Instagram Inactive Engagement

##### INPUT:

```
select * from users as a
left join photos as b on
a.id = b.user_id and
b.user_id is null
```

##### OUTPUT

Result Grid							
Filter Rows:				Export:	Wrap Cell Content:		
	id	username	created_at	id	image_url	user_id	created_at
▶	1	Kenton_Kirlin	2017-02-16 18:22:11	NULL	NULL	NULL	NULL
	2	Andre_Purdy85	2017-04-02 17:11:21	NULL	NULL	NULL	NULL
	3	Harley_Lind18	2017-02-21 11:12:33	NULL	NULL	NULL	NULL
	4	Arely_Bogan63	2016-08-13 01:28:43	NULL	NULL	NULL	NULL
	5	Aniya_Hackett	2016-12-07 01:04:39	NULL	NULL	NULL	NULL
	6	Travon.Waters	2017-04-30 13:26:14	NULL	NULL	NULL	NULL
	7	Kassandra_Homenick	2016-12-12 06:50:08	NULL	NULL	NULL	NULL
	8	Tabitha_Schamberger11	2016-08-20 02:19:46	NULL	NULL	NULL	NULL
	9	Gus93	2016-06-24 19:36:31	NULL	NULL	NULL	NULL
	10	Presley_McClure	2016-08-07 16:25:49	NULL	NULL	NULL	NULL

### 3.Declaration winner of contest:

**INPUT:**

```
select * from
(select user_id , count(photo_id)as count from likes
group by user_id
order by count desc) as a
left join users as b on a.user_id = b.id
```

**OUTPUT:**

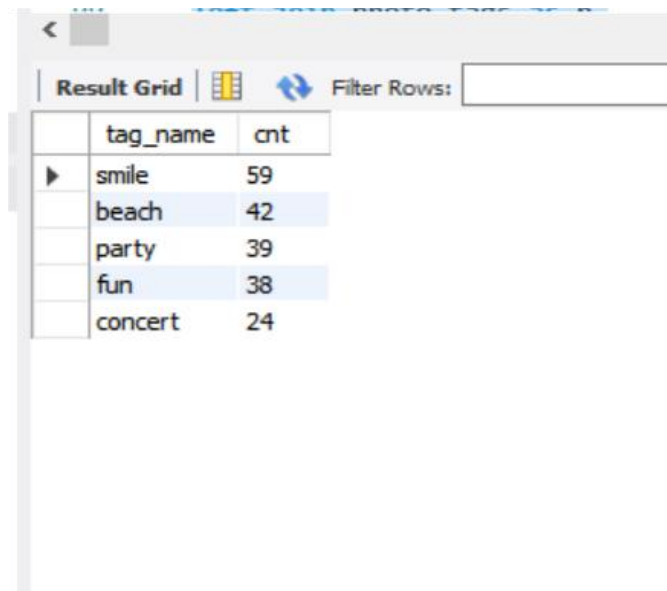
Result Grid					
Filter Rows:					
Export:					
Wrap Cell Content:					
	user_id	count	id	username	created_at
▶	21	257	21	Rocio33	2017-01-23 11:51:15
	71	257	71	Nia_Haag	2016-05-14 15:38:50
	5	257	5	Aniya_Hackett	2016-12-07 01:04:39
	66	257	66	Mike.Auer39	2016-07-01 17:36:15
	41	257	41	Mckenna17	2016-07-17 17:25:45
	14	257	14	Jadyn81	2017-02-06 23:29:16
	57	257	57	Julien_Schmidt	2017-02-02 23:12:48
	24	257	24	Maxwell.Halvorson	2017-04-18 02:32:44
	76	257	76	Janelle.Nikolaus81	2016-07-21 09:26:09
	75	257	75	Leslie67	2016-09-21 05:14:01
	54	257	54	Duane60	2016-12-21 04:43:38
	91	257	91	Bethany20	2016-06-03 23:31:53
	26	257	26	Ollie.Ledner37	2016-08-04 15:42:20

#### 4. Hashtag Research

INPUT:

```
select a.tag_name , count(b.tag_id) as cnt from tags as a
left join photo_tags as b
on a.id = b.tag_id
group by a.tag_name
order by cnt desc limit 5
```

OUTPUT:



The screenshot shows a database query result grid with the following data:

	tag_name	cnt
▶	smile	59
	beach	42
	party	39
	fun	38
	concert	24

#### 5. Ad Campaign Launch

INPUT:

```
select week(created_at) as wk ,
count(week(created_at))as cnt from users
group by wk
order by cnt desc
```

OUTPUT:

Result Grid			Filter Rows:
	wk	cnt	
▶	18	5	
	6	5	
	40	4	
	13	4	
	27	4	
	19	4	
	14	3	
	34	3	
	23	3	
	35	3	
	4	3	
	22	3	
	11	2	

Result 5 x

## 6. Investor Metrics: Bots & Fake Accounts:

INPUT:

```
select b.user_id , count(a.id) from users as a
left join photos as b
on a.id = b.user_id
group by b.user_id
```

OUTPUT:

Result Grid			Filter Rows:
	user_id	count(a.id)	
▶	1	5	
	2	4	
	3	4	
	4	3	
	NULL	26	
	6	5	
	8	4	
	9	4	
	10	3	
	11	5	
	12	4	
	13	5	
	14	4	

Result 6 ×

select count(\*) from bot\_count

where cnt\_likes = '257' ;

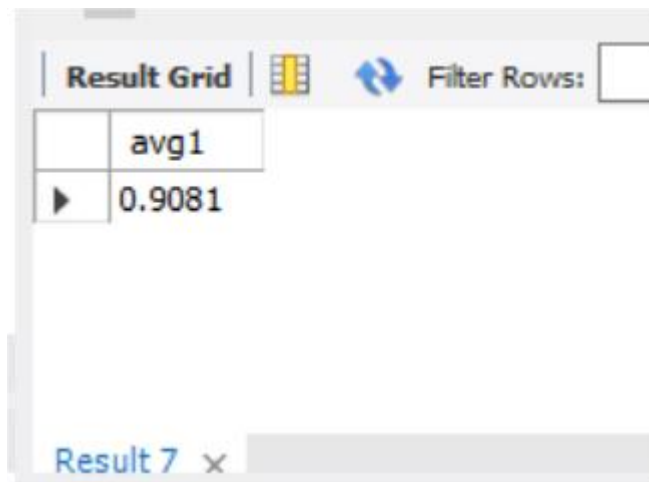
Result Grid		Filter
	count(*)	
▶	13	

Result 2 ×

7.

```
select count(b.image_url) / count(a.id) as avg1 from users as a  
left join photos as b  
on a.id = b.user_id
```

OUTPUT:



The screenshot shows a 'Result Grid' window with a single row of data. The column header is 'avg1' and the value is '0.9081'. The window also includes a 'Filter Rows' button and a tab labeled 'Result 7'.

	avg1
▶	0.9081