

# CODING TEST

This was a coding test that I did using Python and SQL. To see my Python code please go to the Python file.

The questions are as follows:

- 1) Count views, clicks, unique ads and unique campaigns by date.
- 2) Why there was a spike in events on April 5? Which events spiked? For all ads or particular?
- 3) Find top 10 ads by CTR.

Click Through Rate = Total count of clicks/Total count of views

- 4) Are the CTR of ads with video higher than the CTR of ads without video?
- 5) We need to report our revenue by days. What days were the most and the least profitable? We charge for ads if the ad cost type is CPC and the event is click. We also charge when the ad cost type is CPM and the event is view (cost is given per 1000 views).
- 6) What platform is the most popular among clients?
- 7) Are there ads that first were clicked and later viewed?

## 1) Count views, clicks, unique ads and unique campaigns by date.

	date	event_count
1	2019-04-01 00:00:00.0000000	22073
2	2019-04-02 00:00:00.0000000	47117
3	2019-04-03 00:00:00.0000000	59483
4	2019-04-04 00:00:00.0000000	275735
5	2019-04-05 00:00:00.0000000	519707
6	2019-04-06 00:00:00.0000000	75885

Total number of events by days:

```
select date, count(event) as event_count
from dataset
group by date
order by date
```

	date	view_count
1	2019-04-01 00:00:00.0000000	21782
2	2019-04-02 00:00:00.0000000	46572
3	2019-04-03 00:00:00.0000000	59023
4	2019-04-04 00:00:00.0000000	275092
5	2019-04-05 00:00:00.0000000	427386
6	2019-04-06 00:00:00.0000000	60967

Number of views:

```
select date, count(event) as view_count
from dataset
where event = 'view'
group by date
order by date
```

### Number of unique ads:

	date	ad_count
1	2019-04-01 00:00:00.0000000	150
2	2019-04-02 00:00:00.0000000	344
3	2019-04-03 00:00:00.0000000	360
4	2019-04-04 00:00:00.0000000	407
5	2019-04-05 00:00:00.0000000	465
6	2019-04-06 00:00:00.0000000	220

```
select date, count(distinct ad_id) as  
ad_count  
from dataset  
group by date  
order by date
```

### Number of unique campaigns:

	date	campaign_count
1	2019-04-01 00:00:00.0000000	149
2	2019-04-02 00:00:00.0000000	336
3	2019-04-03 00:00:00.0000000	352
4	2019-04-04 00:00:00.0000000	396
5	2019-04-05 00:00:00.0000000	442
6	2019-04-06 00:00:00.0000000	212

```
select date, count(distinct campaign_id)  
as campaign_count  
from dataset  
group by date  
order by date
```

2) Why there was a spike in events on April 5? Which events spiked? For all ads or particular?

date	event_count	view_count	click_count	campaign_count	ad_count
4/1/2019	22073	21782	291	149	150
4/2/2019	47117	46572	545	336	344
4/3/2019	59483	59023	460	352	360
4/4/2019	275735	275092	643	396	407
4/5/2019	519707	427386	92321	442	465
4/6/2019	75885	60967	14918	212	220

	date	campaign_id	ad_id	event	event_count
1	4/5/2019	112260	112583	view	302811
2	4/5/2019	112260	112583	click	91017
3	4/5/2019	107728	107729	view	29724
4	4/5/2019	28142	28142	view	20872
5	4/5/2019	107837	107837	view	8338
6	4/5/2019	38889	38892	view	7986
7	4/5/2019	37720	37720	view	4861
8	4/5/2019	45008	45008	view	3145
9	4/5/2019	46629	46629	view	2302
10	4/5/2019	18424	18425	view	2274
11	4/5/2019	42518	42518	view	1885
12	4/5/2019	22490	22490	view	1440
13	4/5/2019	98280	98325	view	1306
14	4/5/2019	29864	29881	view	1203
15	4/5/2019	44685	44685	view	1114
16	4/5/2019	113251	113350	view	975
17	4/5/2019	111310	111335	view	931
18	4/5/2019	105060	105076	view	910
19	4/5/2019	121308	121309	view	780
20	4/5/2019	39825	39828	view	780

We can see that the ad with ID 112583 had enormous amount of views and clicks, which probably resulted from altered ad settings and budget.

```
select date,campaign_id, ad_id, event, count(event) as event_count
from dataset
where date like '4/5/2019'
group by date, campaign_id, ad_id, event
order by event_count desc, campaign_id, ad_id
```

### 3) Find top 10 ads by CTR.

Click Through Rate = Total count of clicks/Total count of views

	campaign_id	ad_id	event_click	event_view	ctr
1	117157	117164	6	19	0.32
2	112260	112583	105767	351802	0.30
3	42507	42507	3	11	0.27
4	98417	98569	3	16	0.19
5	23595	23599	4	24	0.17
6	46639	46639	44	253	0.17
7	19912	19912	4	25	0.16
8	110363	110414	5	32	0.16
9	45969	45969	2	13	0.15
10	20661	20662	4	26	0.15

```
--top 10 ads by CTR
with view_temp as (
select campaign_id, ad_id, count(event) as event_view
from dataset
where event = 'view'
group by campaign_id, ad_id),

click_temp as (
select campaign_id, ad_id, count(event) as event_click
from dataset
where event = 'click'
group by campaign_id, ad_id)

select top 10 v.campaign_id, v.ad_id, event_click, event_view,
cast(round(event_click*1.000/event_view,2)as decimal(18,2)) as ctr
from view_temp as v
full join click_temp as c
on v.ad_id=c.ad_id
order by ctr desc
```

#### 4) Are the CTR of ads with video higher than the CTR of ads without video?

	campaign_id	ad_id	event_view	has_video	event_click	ctr
1	117353	117353	288	1	4	0.014
2	39715	39715	41	1	1	0.024
3	41059	41059	213	1	4	0.019
4	31491	31491	273	1	2	0.007
5	44949	44949	667	1	1	0.001
6	38376	38376	1841	1	24	0.013
7	31335	31336	163	1	5	0.031
8	43565	43565	60	1	4	0.067
9	23829	23829	82	1	1	0.012
10	26269	26269	763	1	4	0.005
11	43522	43522	64	1	6	0.094
12	43837	43837	15	1	1	0.067
13	37422	37422	41	1	2	0.049
14	45864	45864	47	1	0	0.000
15	45831	45831	81	1	0	0.000
16	39910	39910	93	1	0	0.000
17	119508	119508	47	1	0	0.000
18	33952	33952	451	1	0	0.000
19	38437	38437	27	1	0	0.000
20	45783	45783	31	1	0	0.000

##### With video:

Mean CTR = 0.02014481900764, Median CTR = 0.0097605646386050.

##### Without video:

Mean CTR = 0.01574512278733, Median CTR = 0.0027327504461200.

Overall, the ads with video have better CTR. However, the group sizes (20 with video and 936 without video) do not allow us draw such conclusions.

```

--CTR of ads with video
with view_temp as (
select campaign_id, ad_id, count(event) as event_view, has_video
from dataset
where event = 'view'
group by campaign_id, ad_id, has_video),

click_temp as (
select campaign_id, ad_id, count(event) as event_click, has_video
from dataset
where event = 'click'
group by campaign_id, ad_id, has_video),

final as(
select v.campaign_id, v.ad_id, event_view, v.has_video,
case when event_click is null then 0 else event_click end as event_click
from view_temp as v
full join click_temp as c
on v.ad_id=c.ad_id
where v.has_video = '1')

select *,
cast(round(event_click*1.000/event_view,3) as decimal (18,3)) as ctr
from final

--CTR mean for ads with video
with view_temp as (
select campaign_id, ad_id, count(event) as event_view, has_video
from dataset
where event = 'view'
group by campaign_id, ad_id, has_video),

click_temp as (
select campaign_id, ad_id, count(event) as event_click, has_video
from dataset
where event = 'click'
group by campaign_id, ad_id, has_video),

final as(
select v.campaign_id, v.ad_id, event_view, v.has_video,
case when event_click is null then 0 else event_click end as event_click
from view_temp as v
full join click_temp as c
on v.ad_id=c.ad_id
where v.has_video = '1'),

final2 as(
select *,
(event_click*1.000/event_view) as ctr
from final)

select avg(ctr) as mean
from final2

--CTR Median for ads with video (put this code instead of 'select avg(ctr)...')
select
((select max(ctr) from
(select top 50 percent ctr from final2 order by ctr) as bottomhalf)
+
(select min(ctr) from
(select top 50 percent ctr from final2 order by ctr desc) as tophalf)) / 2 as Median

```

5) We need to report our revenue by days. What days were the most and the least profitable?  
 We charge for ads if the ad cost type is CPC and the event is click.  
 We also charge when the ad cost type is CPM and the event is view (Cost is given per 1000 views).

	date	sum_cost
1	2019-04-05 00:00:00.0000000	96123.1216993383
2	2019-04-04 00:00:00.0000000	54988.8011000789
3	2019-04-03 00:00:00.0000000	14145.9569000007
4	2019-04-06 00:00:00.0000000	13346.9308000065
5	2019-04-02 00:00:00.0000000	13285.7901999994
6	2019-04-01 00:00:00.0000000	6655.70719999992

We can see that April 5 was the most profitable day (96 123) and April 1 - the least profitable (6 655).

```
with cost_per_click as (
select date, sum(ad_cost) as sum_cost_click
from dataset
where event = 'click' and ad_cost_type = 'CPC'
group by date
),

cost_per_mile as (
select date, sum(ad_cost/1000) as sum_cost_view
from dataset
where event = 'view' and ad_cost_type = 'CPM'
group by date)

select c.date,
(sum_cost_click+sum_cost_view) as sum_cost
from cost_per_click as c
inner join cost_per_mile as m
on c.date = m.date
order by sum_cost desc
```

6) What platform is the most popular among clients?

	platform	view_count	percentage
1	android	445722	50.03
2	ios	267117	29.99
3	web	177983	19.98

Android accounts for 50% of ad views. It's the most popular platform.

```
select platform, count(event) as view_count,
cast(round(count(event)*100.0/(select count(event)
from dataset
where event = 'view'),2) as decimal(18,2)) as percentage
from dataset
where event = 'view'
group by platform
order by view_count desc
```

## 7) Are there ads that first were clicked and later viewed?

	time	user_id	ad_id
1	4/1/2019 21:20	107184	107798
2	4/1/2019 21:31	114880	114886
3	3/31/2019 22:30	115808	115825
4	4/1/2019 16:38	117351	117364
5	4/1/2019 0:55	120412	120431
6	3/31/2019 21:17	120417	120536
7	4/1/2019 15:57	120793	120796
8	3/31/2019 23:33	14606	26204
9	4/4/2019 21:18	1575	18681
10	4/1/2019 10:08	1630	19223
11	4/4/2019 21:10	21688	33033
12	4/3/2019 22:21	22708	36758
13	4/3/2019 21:02	23401	38224
14	4/1/2019 10:59	26105	41500
15	4/3/2019 21:09	31917	44283
16	4/4/2019 21:54	33560	44766
17	4/1/2019 12:40	36207	45418
18	4/1/2019 21:01	42724	46639
19	4/4/2019 21:05	6134	23599
20	4/4/2019 0:09	98413	98569

Overall, 20 ads were first clicked. In the table above you can see the ad\_id, user\_id and time when it was first clicked.

```
--add column 'action' with value 1 to all the rows
with data as (
select date, time, user_id, ad_id, event,
case when coalesce(user_id, 1e10) >= 0 then 1
     else 0
end as action
from dataset
group by date, time, user_id, ad_id, event),

/*if user_id & ad_id repeats, sum up the values in 'action'(cumsum); if either user_id or
ad_id is different, restart summing the values in 'action'. Call the column
'action_number'*/
number as (
select date, time, user_id, ad_id, event,
sum(action) over (partition by user_id, ad_id order by time) as action_number
from data)

--leave the rows with 'click' as their first action
select time, user_id, ad_id
from number
where action_number = '1' and event = 'click'
```