1. Transactions:

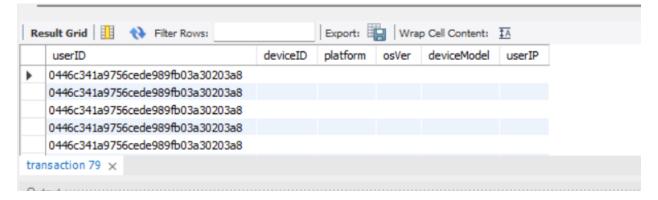
- Query on the same users are using the same device fingerprint or not

```
-- check if many users are using the same device fingerprint
select userID,
    count(*) as txn_count,
    sum(case when userChargeAmount = 0 then 1 else 0 end) as zero_amount_txns,
    min(reqDate) as first_txn,
    max(reqDate) as last_txn
from transaction
group by userID
having txn_count >= 5 and (zero_amount_txns >= 3);
```

- 8 users made more than 5 actions or made no payment in a short periods
- Next action: we will check whether they used the same IP or device model or not?

```
select userID, deviceID, platform, osVer, deviceModel, userIP
from transaction

where userID in ('0446c341a9756cede989fb03a30203a8', 'acd6b370d43843dc12aab97dfa39583e','4219acfb75a28586e9d86f727badefe1',
'e0c979e16e7cfleedf92846ccc1bc789','9a1256c59ef6c5b0fde46b6024c9d119',
'477db45ec864a1bfb73a7c08572b297b','d48fe5ad11427ce4422316efe57af8bd','db6b865ee81111bbbae46e3233db71f3');
```



-> 53 rows returns

0446c341a9756cede989fb03a30203a8 - 10 rows

4219acfb75a28586e9d86f727badefe1 - 5rows

477db45ec864a1bfb73a7c08572b297b - 7 rows

9a1256c59ef6c5b0fde46b6024c9d119 - 9 rows

Acd6b370d43843dc12aab97dfa39583e - 5 rows

D48fe5ad11427ce4422316efe57af8bd - 5 rows

E0c979e16e7cf1eedf92846ccc1bc789 - 7 rows

The results reveals that: same deviceID (669083EAA6629B91B3932B6E5F12DD59)x2 with the userIP (171.248.159.136) across the multiple users

→ several users logging in or transacting from the same deviceIP

Notice that the userID such as 0446c341a9756cede989fb03a30203a8, 477db45ec864a1bfb73a7c08572b297b, 9a1256c59ef6c5b0fde46b6024c9d119, E0c979e16e7cf1eedf92846ccc1bc789 are having signs of bot-driven abuse because of being repeated many times.

Besides, it is recorded that almost **deviceID column** has missing or empty values across multiple accounts, meaning that:

- the system couldn't track the device
- the bot can be intentionally hiding device data

Next step: we will find deviceFingerprint shared across multiple accounts?

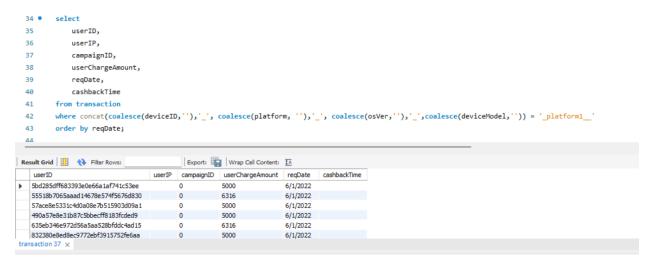
So why deviceFingerprint? – Because bot often reuse the same device environment to perform like creating fake accounts, claiming promotions repeatedly or triggering actions quickly. Otherwise, normal people often have their own unique phone/device or at least different combinations of device properties.

```
-- find deviceFingerprints shared across multiple accounts
22
23 • SELECT
        CONCAT(COALESCE(deviceID, ''), '_', COALESCE(platform, ''), '_', COALESCE(osVer, ''), '_', COALESCE(deviceModel, '')) AS deviceFingerprint,
          COUNT(DISTINCT userID) AS user_count,
25
          COUNT(*) AS total_txns,
26
        MIN(cashbackTime) AS first_cashback,
27
28
          MAX(cashbackTime) AS last_cashback
29
     GROUP BY deviceFingerprint
30
31
     HAVING user_count >= 3
      ORDER BY user_count DESC;
33
Export: Wrap Cell Content: IA
  deviceFingerprint user_count total_txns first_cashback last_cashback
 _platform1_ 169 252
____ 157 259 6/1/2022 6/1/2022
 _platform3__ 4
```

If more than 3 users share the same exact deviceFingerprint and cashback behavior, they are highly likely automated.

From the result, we can see that 'platform 1' and '-----' are mismatching, it can be bot behavior. However, we need to confirm on some information of each platform.

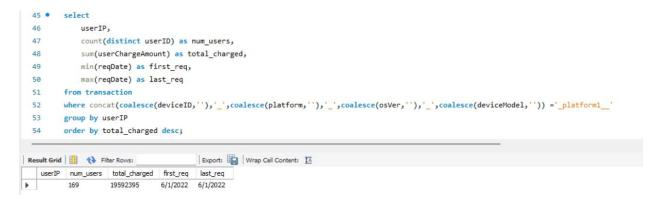
****Platform 1:



→ 252 rows returned

From the result, *userIP*, *campaignID* and *cashbackTime* are almost blank, implying that information is hidden due to bot or automation.

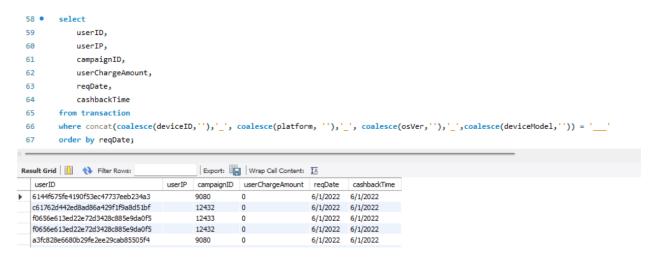
Normally, useIP is a line of numbers (eg: 14.191.228.77) instead of 0/blank, meaning that there are surely signs of bot-driven abuse.



Abnormal behavior in *platform 1* also reveals through the amount deducted from transactions. In this case, unknown *userIP* supports the motion that bot is screaming the device and many campaignID and stole a quite large amount of **19592395** (**169/252 users**).

Moreover, within a day, these accounts were created so that the total charged are huge for these quite new accounts.

****Platform '-----':



→ 259 rows

Like platform 1, userIP is missing, userChargeAmount is recorded of 0, but campaignID are identical at some transactions.

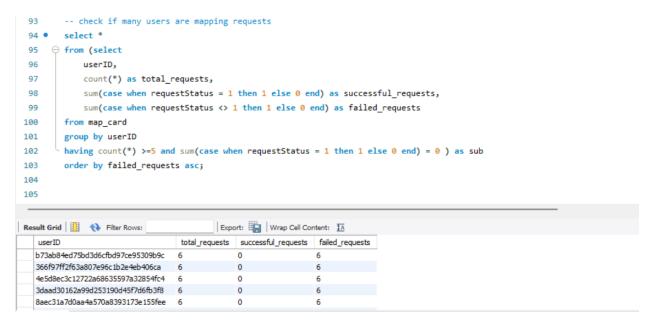
We can suspect the bot due to:

- Lack of userIP is abnormal.
- userChargeAmount = 0, it can be that bot is testing or triggering spam the campaigns
- campaignID is repeatedly, it can be that bot is repeating a campaign to gain more rewards



2. Map card

Case 1: the same users repeated mapping requests



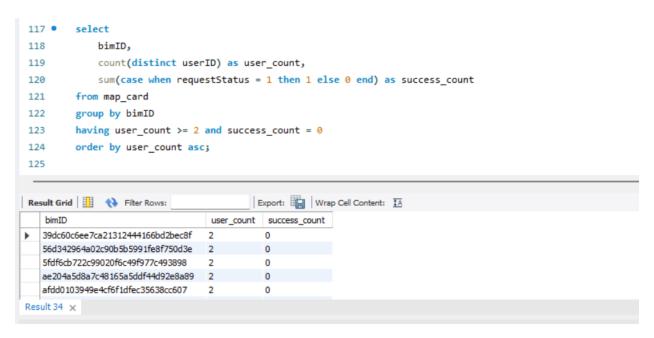
→ 60 rows

Case 2: many users are requesting mapping in many times per day

```
-- check if many users are requesting mapping in many times per day
106 • select
107
         userID.
         reqDate,
         count(*) as total_requests,
110
         sum(case when requestStatus = 1 then 1 else 0 end) as failed_requests
111
      from map_card
      group by userID, reqDate
113
       having count(*) >= 3 and sum(case when requestStatus = 1 then 1 else 0 end) = 0
      order by failed_requests asc;
Export: Wrap Cell Content: IA
 userID reqDate total_requests failed_requests
```

→ No sign

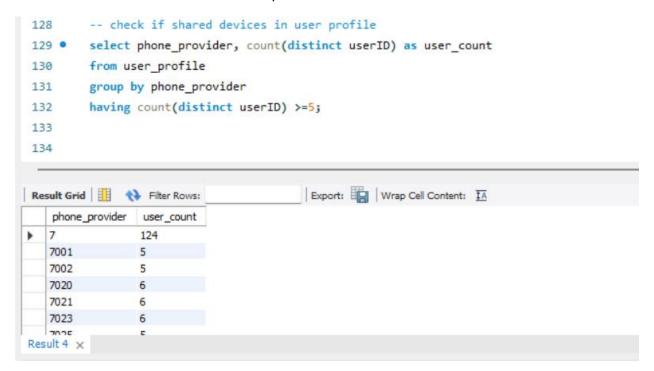
Case 3: Analyze according to bimID



Many users use the same bimID but failed, we observe that there are 8 cases of being attacked

3. User profile

We will detect shared devices in user profile



- → Bot-driven possibility
- → We will find out which userID? (in file excel)

```
134 • ⊖ with filtered_provider as (
135
             select phone_provider, count(distinct userID) as user_count
136
             from user_profile
             group by phone provider
137
             having count(distinct userID) >5
139
        ),
140
        shared_device as
             (select deviceID
141
142
             from transaction
             group by deviceID
             having count(distinct userID)>1
144
        ),
145
       user_shared_device as (
 146
 147
             select up.userID, up.phone_provider, count(distinct t.deviceID) as share_device_count
 148
             from user_profile up
             join transaction t on up.userID = t.userID
 149
 150
             join shared_device sd on t.deviceID = sd.deviceID
             join filtered_provider fp on up.phone_provider = fp.phone_provider
 151
 152
              group by up.userID, up.phone provider
 153
 154
         select userID, phone_provider, share_device_count
 155
         from user shared device
 156
         order by phone_provider, userID;
  Result Grid Filter Rows:
                                             Export: Wrap Cell Content: IA
      userID
              phone_provider
                              share_device_count
  Result 26 ×
```

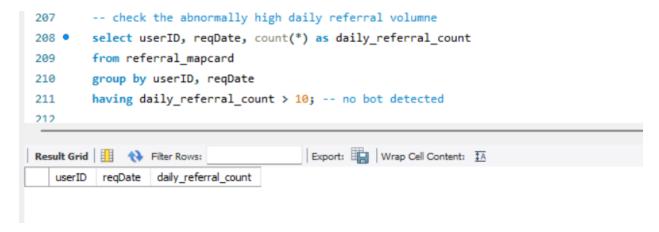
For deviceID, no sign of sharing.

4. Referral_mapping:

In the referrap_mapping, we will observe the userID and refereeId in 2 aspects of bot-driven abuse signs:

Case 1: Abnormally high daily referral volume

→ to find users sending unusual number of referrals in a single day



(positive) - no sign of bot/automation

Case 2: Daily referral streaks

Bot may send referrals every single day – unlike normal users

```
176
        -- check if bot may send referral every single day
177 •
        select userID, count(distinct reqDate) as active_days
        from referral_mapcard
178
        group by userID
179
180
        having active_days > 3;
181
                                         Export: Wrap Cell Content: IA
userID
                                 active_days
 0446c341a9756cede989fb03a30203a8
  2cf86e0ca892dc6712b65541e0523f85
  5b9008b34e646b27d4f839a5806301f8
  810ab3f18ace731b6d81d6f96323dea8 4
  f45574b03d02356238cbe06268144535
  fe4ce77fee80c0e298faec18c7e3d83a
```

These referrals are possible of being bot-driven abuse, however, based on the *active_days* we can observe more about user *0446** and *2cf8** because of the high possibility.

5. Transfer

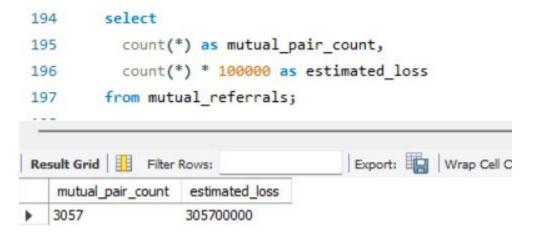
First of all, we will observe the transfer among senders/receivers



→ 40 rows returns

Noticeably sender 256f*, 4ad9*, 583b*, d11e*

```
-- count unique mutual pairs and estimate loss from loop patterns
182 ● ⊖ with mutual referrals as (
183
          select
184
            least(a.sender, a.receiver) as user1,
185
            greatest(a.sender, a.receiver) as user2
          from transfer a
186
          join transfer b
187
            on a.sender = b.receiver
188
           and a.receiver = b.sender
189
190
          group by
191
            least(a.sender, a.receiver),
            greatest(a.sender, a.receiver)
192
       )
193
```

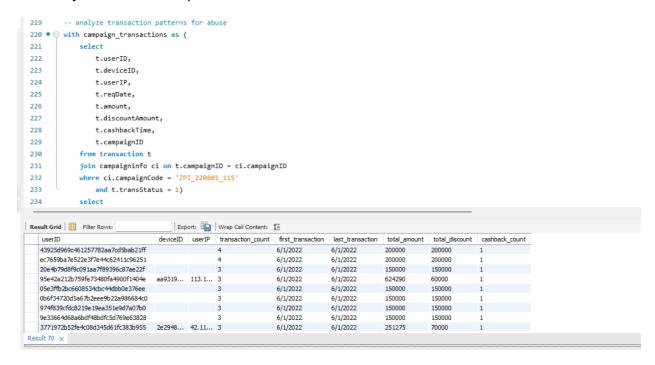


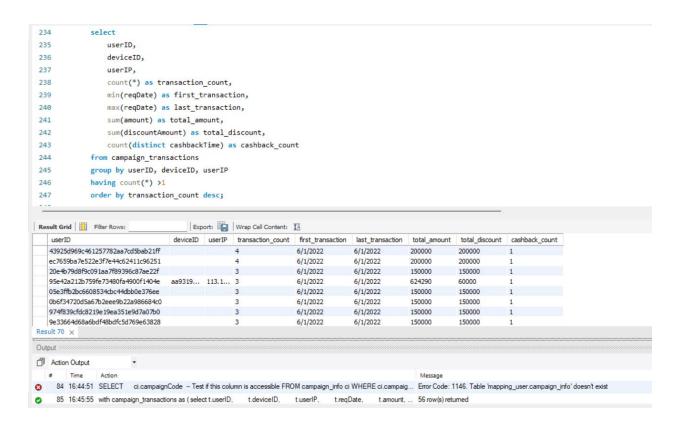
Since we detect the loop patterns, the estimated_loss is 305,700,000.

6. Campaign

Here, we will analyze based on campaign code ZPI_220801_115 and information from transactions:

1- Analyze transactions patterns for abuse





Then we will calculate total amount for suspended abusive transactions

```
250 ● ⊝ with campaign_transactions as (
251
             select
252
                 t.userID,
253
                 t.deviceID,
254
                 t.userIP,
255
                 t.reqDate,
256
                 t.amount,
257
                 t.discountAmount,
                 t.cashbackTime,
258
                 t.campaignID
259
260
             from transaction t
             join campaigninfo ci on t.campaignID = ci.campaignID
261
262
            where ci.campaignCode = 'ZPI_220802_115'
263
             and t.transStatus = 1
```

```
1),
264
265
      suspected abuse as (
266
             select
                 userID,
267
                 count(*) as transaction amount,
268
                 coalesce(sum(amount),0) as total_amount,
269
                 coalesce(sum(discountAmount),0) as total discount
270
             from transaction
271
             group by userID
272
             having count(*) > 10)
273
274
        select
275
             sum(transaction_amount) as total_abusive_transactions,
             sum(total_amount) as total_abusive_amount,
276
             sum(total_discount) as total_abusive_discount
277
        from suspected_abuse;
278
279
280
                                      Export: Wrap Cell Content: IA
Result Grid Filter Rows:
   total_abusive_transactions
                         total_abusive_amount
                                          total_abusive_discount
                        472000
 28
```

While running the campaign, there are 28 transactions having sign of being abusive, the estimated loss is 472,000.

```
280
         -- 3. check deviceID and IP patterns
281 •
         select
282
             deviceID,
283
             userIP,
284
             count(distinct userID) as unique_users,
             count(*) as transaction_count
285
         from transaction t
286
         join campaigninfo ci on t.campaignID = ci.campaignID
287
         where ci.campaignCode = 'ZPI 220801 115'
288
             and transStatus = 1
289
         group by deviceID, userIP
290
         having count(distinct userID) > 1
291
         order by transaction count desc;
292
293
                                          Export: Wrap Cell Content: IA
Result Grid Filter Rows:
   deviceID userIP
                  unique_users
                               transaction_count
٠
                   127
                               211
```