Section 2 - SQL

Data source:

Use the cleaned dataset from the data wrangling section. For this section only, assume the dataset provided has data from 2016 onwards stored in the same format.

Context:

1. “Platform hard churn” is defined as someone who has not made any transaction (any service) in the past 6 months or more (6 months <= inactive)

2. “Platform soft churn” is defined as someone who has not made any transaction in more than 1 month but below 6 months (1 month <= inactive < 6 months)

3. “Product hard churn” and “Product soft churn” are defined very similarly, but instead of any transaction, it is on a specific service_type. Meaning GoFood hard churn users are those who have not completed GoFood transactions in the past 6 months

Data set description

customer_id (integer): customer identifier, which is unique at customer level
driver_id (integer): customer identifier, which is unique at customer level
order_no (string): unique identifier at order level
booking_time (timestamp): the timestamp when booking completed
service_type (string): type of service that customer booked
actual_gmv(string): the amount of money that customer spent on particular booking

Questions:

1. Calculate average spending of GO-SEND users throughout every month of 2020

```
# generate months of year
With month_year as (
Select “2020-01-01” as dt, 1 as num
union all
Select dateadd(month, 1, dt), num + 1
from month_year
where num < 12),

# calculate month wise customers, total amount, average spend
month_spend as (
Select dateadd(month, datediff(month, 0, booking_time), 0) as “booking_month”,
count(distinct customer_id) as “no_of_customers”, sum(actual_gmv) as total_amount,
avg(actual_gmv) as “average_spend”
from table
where lower(service_type) like ‘%go-send%” and date(booking_time) between ‘2020-01-01’
and ‘2020-12-31’
group by 1)
```

```
Select dt, average_spend, coalesce(average_spend_per_customer,0) as
“average_spend_per_customer”
from
(Select booking_month, average_spend, coalesce(total_amount / no_of_customers, 0) *
100 as average_spend_per_customer
from month_spend) as a
left join month_year as b on a.booking_month = b.dt)
order by 1;
```

2. Calculate number of “platform hard churn” users and “platform soft churn” users for each month
in 2020.
For example:
platform hard churn in February 2020 means how many users who have not made a booking in the past 6 months or more, i.e those who last transacted july 2019 or later (jan 2020, dec 2019, nov 2019, oct 2019, sept 2019, aug 2019 → no transaction in 6 months)

```
# generate months of year
With month_year as (
Select “2020-01-01” as dt, 0 as num
union all
Select dateadd(month, 1, dt), num + 1
from month_year
where num < 12),

# get customer last booking month based on each month in 2020
bookings as (
Select distinct dt, customer_id, max(booking_month) as “booking_month”,
from
(Select distinct dt, customer_id, dateadd(month, datediff(month, 0, booking_time), 0) as
“booking_month”
from table
where date(booking_time) =< ‘2020-12-31’) as a
cross join month_year as b
where booking_month <= dt)
group by 1,2),

# differentiate platform hard and soft churn users
churn_users as (
Select dt, customer_id, booking_month,
case when month_diff >= 6 then ‘platform hard churn’
when month_diff between 1 and 5 then ‘platform_soft_churn’
end as user_type
from
(Select customer_id, booking_month, datediff(month, booking_month, dt) as month_diff
from bookings) as a
cross join month_year as b)

Select dt, “platform hard churn”, “platform_soft_churn”
from
(Select dt, user_type, coalesce(count(distinct customer_id),0) as cnt
from churn_users
```

```
group by 1,2)
pivot
(sum(cnt) for user_type in ('platform hard churn', 'platform_soft_churn') as t
order by 1;
```

3. What is the reactivation rate (those who transacted after labelled churn) of "platform hard churn" users throughout each month of 2020? Identify which product helps in reactivation the most in each month.

```
# get customer recent booking month and service_type based on each month in 2020
bookings as (
Select distinct dt, customer_id, min(booking_month) as "booking_month",
from
(Select distinct a.customer_id, dateadd(month, datediff(month, 0, a.booking_time), 0) as
"booking_month"
from table as a
inner join churn_users as b
on a.customer_id = b.customer
where date(a.booking_time) >= '2020-01-01' and b.user_type = 'platform hard churn') as e
cross join month_year as c
where booking_month >= dt
group by 1,2),

# get reactive users and their service type
reactive_users as (
Select distinct dt, c.customer_id, d.service_type, c.booking_month, user_type
from
(Select dt, customer_id, booking_month,
case when month_diff >= 0 then "reactive"
else "active" as user_type
from
(Select customer_id, booking_month, datediff(month, dt, booking_month) as month_diff
from bookings) as a
cross join month_year as b) as c
inner join table as d
on c.customer_id = d.customer_id and month(c.booking_month) =
month(d.booking_month)),

# reactive users: most frequent service type
freq_servicetype as (
Select dt, service_type, dense_rank() over(partition by dt order by freq_service desc) as rn
from
(Select dt, service_type, count(distinct customer_id) as freq_service
from reactive_users
group by 1,2))

Select e.dt,  "platform hard churn", "reactive", coalesce("reactive" / "platform hard churn", 0)
* 100 as reactivation_rate, f.service_type
from
(Select dt, count(c.customer_id) as "platform hard churn", count(r.customer_id) as "reactive"
from churn_users as c
left join
(Select * from
reactive_users where user_type = 'reactive') as r
on c.customer_id = r.customer_id and c.dt = r.dt)
group by 1) as e
```

```
left join freq_servicetype as f
on f.dt = e.dt
where f.rn = 1
order by 1;
```