Name: Nguyen Anh Tai

## FINAL EXAMINATION

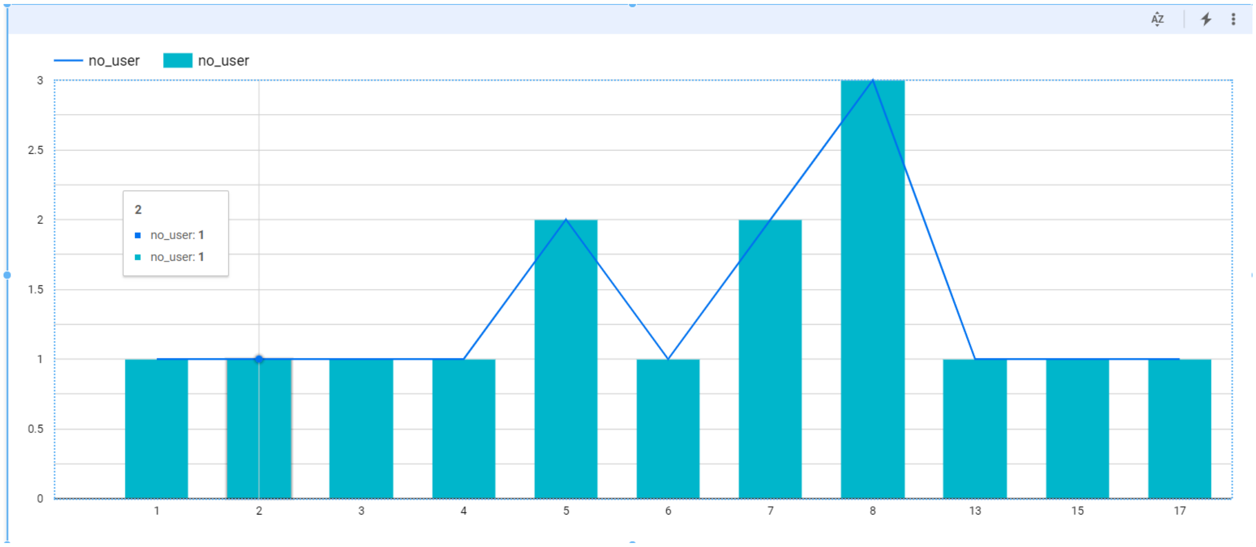
```
-- Question 1: Viết câu SQL để trả về chi phí khuyến mãi của mỗi
tỉnh thành.
SELECT Province AS province, SUM(Voucher_Amount) AS
promotion_Cost
FROM`vef-bi-course.dummy_data.user_profile` a
INNER JOIN `vef-bi-course.dummy_data.transactions` b ON
a.User_ID = b.User_id
INNER JOIN `vef-bi-course.dummy_data.promotions` c ON b.Tid =
c.TID
GROUP BY (Province)
ORDER BY promotion_Cost DESC;

-- Question 2: Viết câu SQL trả về danh sách user_id, user_name,
và province tương ứng của các users chưa bao giờ sử dụng dịch vụ
Grab Food
WITH grabfood AS
(
    SELECT DISTINCT User_id AS id
    FROM `vef-bi-course.dummy_data.transactions`
    WHERE Service_group = 'Grab Food'
)
SELECT a.*
FROM `vef-bi-course.dummy_data.user_profile` a
LEFT JOIN grabfood b ON a.User_ID = b.id
WHERE b.id IS NULL;
```

**Question 3:**

a. First build a histogram of user for the last 27 days to choose the appropriate metric:

```
WITH user_active AS(
        SELECT member_id, COUNT(DISTINCT DATE(submit_date)) AS date_count
    FROM `vef-bi-course.final_test.booking_tracker`
          WHERE DATE(submit_date) BETWEEN '2019-10-01' AND '2019-10-31'
    GROUP BY member_id
)
SELECT date_count, COUNT(member_id) AS no_user
FROM user_active
GROUP BY date_count
ORDER BY 1
```



→ From this, the majority of users are active within 5 to 8 days a month → Apply the metric for month → monthly retention

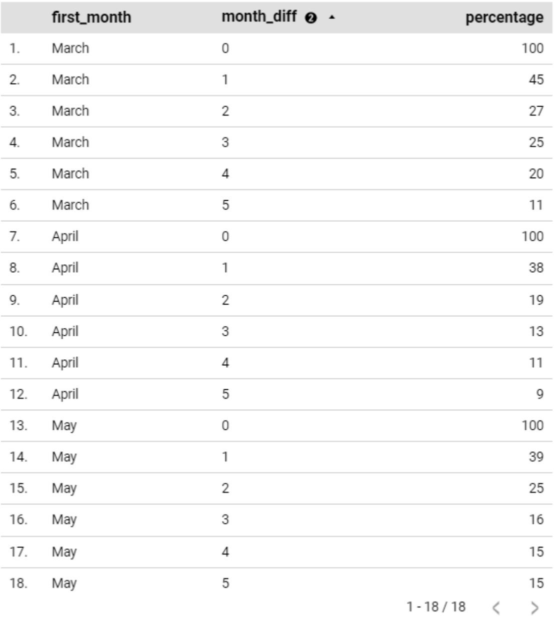
I have also checked for other months and the graph is geared towards 1 to 5 days → monthly retention would be appropriate

```
-- Divide users into cohort month by the first purchasing
with monthly_user AS
```

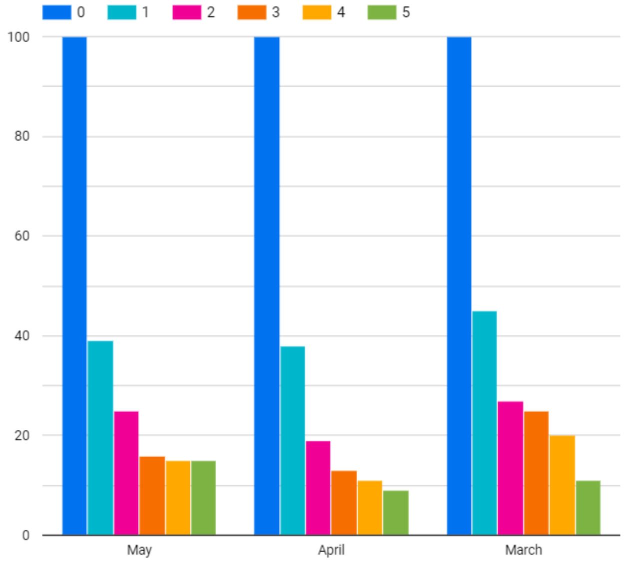
```
(
    SELECT DISTINCT
    member_id,
    EXTRACT(MONTH FROM MIN(DATE(submit_date))) AS first_month
    FROM `vef-bi-course.final_test.booking_tracker`
    GROUP by 1
    ORDER by 1, 2
), cohort_group AS(
    SELECT first_month, COUNT(DISTINCT member_id) AS no_new_user
    FROM monthly_user
    GROUP BY first_month
    ORDER BY first_month
), next_purchasing AS(
    SELECT a.member_id,
      CAST((EXTRACT(MONTH FROM DATE(submit_date)) - first_month)
AS int64) AS month_diff
    FROM `vef-bi-course.final_test.booking_tracker` a
    LEFT JOIN monthly_user b ON a.member_id = b.member_id
    GROUP BY 1,2
), next_combined AS(
    SELECT first_month,
    month_diff,
    COUNT (DISTINCT a.member_id) AS num_user
    FROM next_purchasing a
    LEFT JOIN monthly_user b ON a.member_id = b.member_id
    GROUP BY 1, 2
)
SELECT a.first_month,
CAST (month_diff AS BIGNUMERIC) AS month_diff,
no_new_user,
num_user,
num_user*100/no_new_user AS percentage
FROM next_combined a
LEFT JOIN cohort_group b ON a.first_month = b.first_month
ORDER BY 1,2
```

**Link visualization:**

https://datastudio.google.com/reporting/961f041c-539e-45a7-9e2d-cf9546962bab



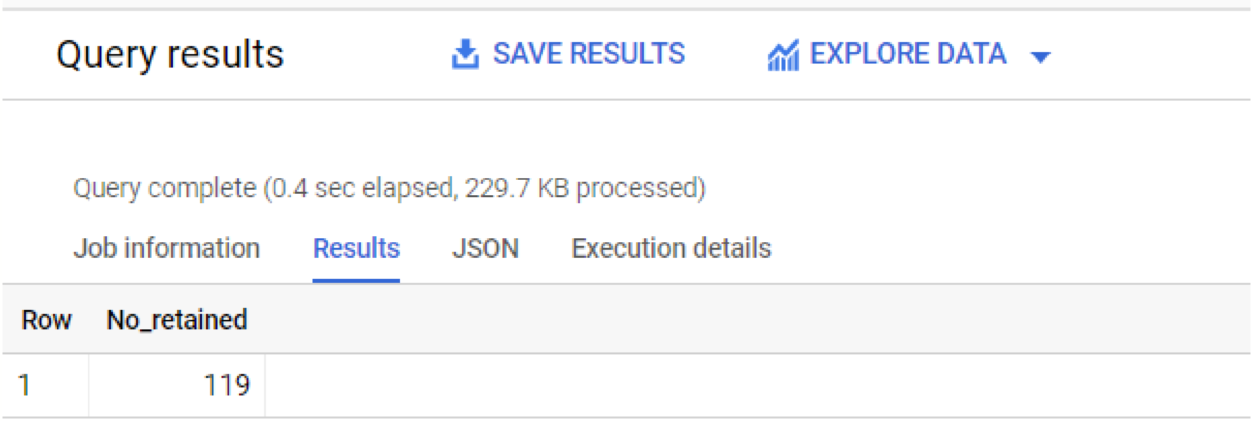
| | first_month | month_diff ❷ ▴ | percentage |
|---|---|---|---|
| 1. | March | 0 | 100 |
| 2. | March | 1 | 45 |
| 3. | March | 2 | 27 |
| 4. | March | 3 | 25 |
| 5. | March | 4 | 20 |
| 6. | March | 5 | 11 |
| 7. | April | 0 | 100 |
| 8. | April | 1 | 38 |
| 9. | April | 2 | 19 |
| 10. | April | 3 | 13 |
| 11. | April | 4 | 11 |
| 12. | April | 5 | 9 |
| 13. | May | 0 | 100 |
| 14. | May | 1 | 39 |
| 15. | May | 2 | 25 |
| 16. | May | 3 | 16 |
| 17. | May | 4 | 15 |
| 18. | May | 5 | 15 |



## b. Count the number of retained user

```
WITH first_date AS(
    SELECT member_id,
    MIN(DATE(submit_date)) AS first_date
    FROM `vef-bi-course.final_test.booking_tracker`
    GROUP BY member_id
), day_differ AS (
    SELECT a.*,
    b.first_date,
    DATE_DIFF(DATE(a.submit_date),first_date, DAY) AS day_diff
    FROM `vef-bi-course.final_test.booking_tracker` a
    LEFT JOIN first_date b ON a.member_id = b.member_id
) SELECT COUNT(DISTINCT member_id) AS No_retained
FROM day_differ
WHERE day_diff >= 30
```

→ There are 119 retained user



Query complete (0.4 sec elapsed, 229.7 KB processed)

| Row | No_retained |
|---|---|
| 1 | 119 |

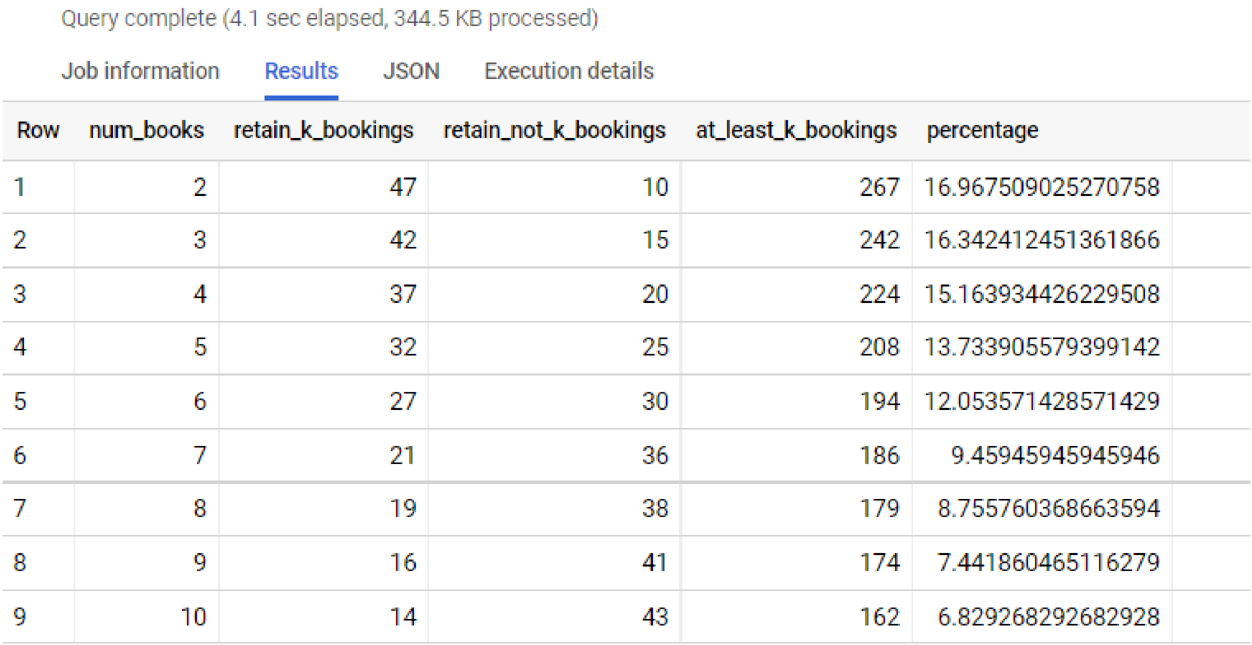
### c. Find the habit moment

Because we assume retained users are the ones who continue to use the service of the companies after 30 days. Therefore, we can start the assumption of aha moment: “X bookings made in the first Y weeks” with Y starting at 5.

First we find the table of number of bookings made in the first 5 weeks

I have the SQL code presented below with the table

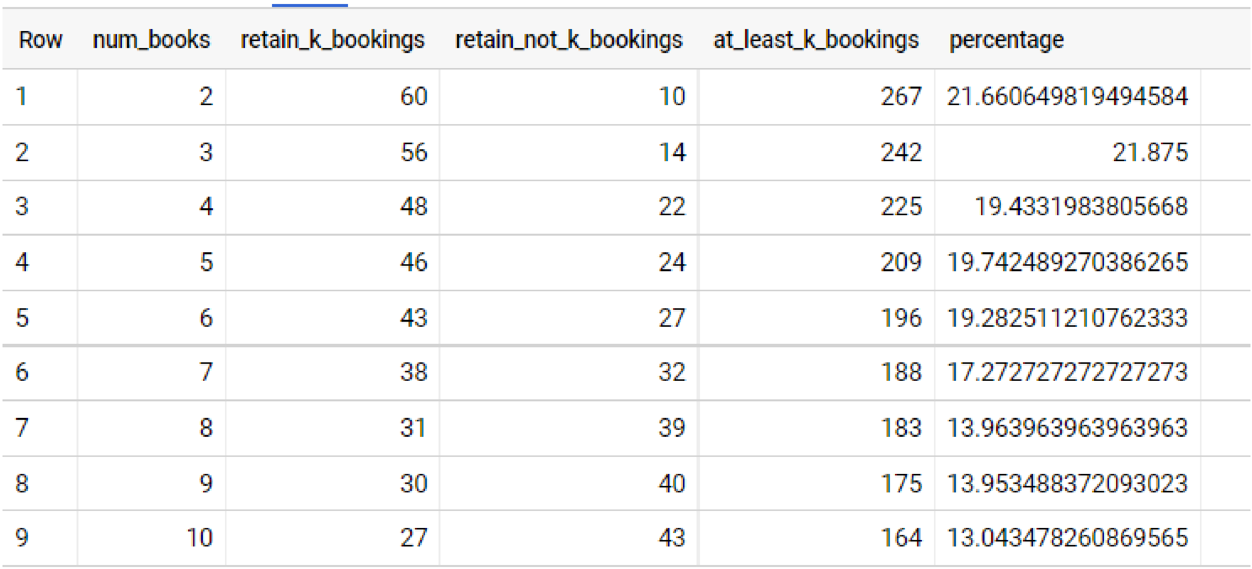
```
-- The number of bookings made by each person during the first 5 weeks since the first bookings
WITH first_date AS(
    SELECT member_id,
    MIN(DATE(submit_date)) AS first_date
    FROM `vef-bi-course.final_test.booking_tracker`
    GROUP BY member_id
), day_differ AS (
    SELECT a.*,
    b.first_date,
    DATE_DIFF(DATE(a.submit_date),first_date, DAY) AS day_diff
    FROM `vef-bi-course.final_test.booking_tracker` a
    LEFT JOIN first_date b ON a.member_id = b.member_id
), no_bookings_y AS(
    SELECT member_id, COUNT(booking_id) AS no_bookings
    FROM day_differ
    WHERE day_diff<=35
    GROUP BY member_id
    ORDER BY 2
), booking_user_count AS(
    SELECT no_bookings, COUNT(member_id) AS member_count
    FROM no_bookings_y
    GROUP BY no_bookings
    ORDER BY 1
), sum_user AS(
    SELECT SUM(member_count) total_user
    FROM booking_user_count
), raw_table AS(
    SELECT *,
    SUM(member_count) OVER(ORDER BY no_bookings) AS cum_sum
    FROM booking_user_count, sum_user
    ORDER BY 1
), lag_table AS (
    SELECT *,
    LAG(cum_sum,1,0) OVER (ORDER BY no_bookings ASC) AS move_1
    FROM raw_table
    ORDER BY 1
), user_at_least_k_bookings AS(
    SELECT no_bookings,
    total_user - move_1 AS at_least_k_bookings
    FROM lag_table
    LIMIT 9 OFFSET 1
),
-- Now we will turn to the number of retained users who make at least X bookings in the first 5 weeks
retained_user_week_limited AS(
    SELECT member_id, COUNT(booking_id) AS num_books
    FROM day_differ
    WHERE day_diff >= 30 AND day_diff<=35
    GROUP BY member_id
    ORDER BY 2
), book_retained_user_count AS(
    SELECT num_books,
    COUNT(member_id) AS no_member
    FROM retained_user_week_limited
    GROUP BY num_books
    ORDER BY 1
), sum_user_retained AS(
    SELECT SUM(no_member) AS sum_retain
    FROM book_retained_user_count
), raw_table_1 AS(
    SELECT *,
    SUM(no_member) OVER (ORDER BY num_books) AS cum_sum
    FROM book_retained_user_count, sum_user_retained
), lag_table_1 AS(
    SELECT *,
    LAG(cum_sum,1,0) OVER (ORDER BY num_books ASC) AS move_1
    FROM raw_table_1
    ORDER BY 1
), retained_at_least_k_bookings AS(
    SELECT num_books,
    sum_retain - move_1 AS retain_k_bookings,
    move_1 AS retain_not_k_bookings
    FROM lag_table_1
    LIMIT 9 OFFSET 1
), combine_table AS(
    SELECT a.*,
    b.at_least_k_bookings
    FROM retained_at_least_k_bookings a
       LEFT JOIN user_at_least_k_bookings b ON a.num_books =
b.no_bookings
)
SELECT *,
(retain_k_bookings)/(retain_not_k_bookings + at_least_k_bookings
)*100 AS percentage
FROM combine_table
```



Query complete (4.1 sec elapsed, 344.5 KB processed)

| Row | num_books | retain_k_bookings | retain_not_k_bookings | at_least_k_bookings | percentage |
|---|---|---|---|---|---|
| 1 | 2 | 47 | 10 | 267 | 16.967509025270758 |
| 2 | 3 | 42 | 15 | 242 | 16.342412451361866 |
| 3 | 4 | 37 | 20 | 224 | 15.163934426229508 |
| 4 | 5 | 32 | 25 | 208 | 13.733905579399142 |
| 5 | 6 | 27 | 30 | 194 | 12.053571428571429 |
| 6 | 7 | 21 | 36 | 186 | 9.45945945945946 |
| 7 | 8 | 19 | 38 | 179 | 8.755760368663594 |
| 8 | 9 | 16 | 41 | 174 | 7.441860465116279 |
| 9 | 10 | 14 | 43 | 162 | 6.829268292682928 |

→ From this, 2 is the number which has the highest coverage When I replace the figure 35 with 42 which is equivalent to 6 weeks, the figure is presented like below:



| Row | num_books | retain_k_bookings | retain_not_k_bookings | at_least_k_bookings | percentage |
|---|---|---|---|---|---|
| 1 | 2 | 60 | 10 | 267 | 21.660649819494584 |
| 2 | 3 | 56 | 14 | 242 | 21.875 |
| 3 | 4 | 48 | 22 | 225 | 19.4331983805668 |
| 4 | 5 | 46 | 24 | 209 | 19.742489270386265 |
| 5 | 6 | 43 | 27 | 196 | 19.282511210762333 |
| 6 | 7 | 38 | 32 | 188 | 17.272727272727273 |
| 7 | 8 | 31 | 39 | 183 | 13.963963963963963 |
| 8 | 9 | 30 | 40 | 175 | 13.953488372093023 |
| 9 | 10 | 27 | 43 | 164 | 13.043478260869565 |

→ 2 and 3 will have a similar highest coverage. But we can notice that the coverage percentage also increases for other number of bookings, which can somehow imply that by choosing 2 (2 bookings in the first 5 weeks), the users are more likely make more bookings

*Therefore, I conclude that the habit moment is: making 2 bookings in 5 weeks*