Executive Summary

Travel Tide Rewards Program

TravelTide, a new rising star in the E-booking online travel industry, aims to enhance its marketing strategies to improve customer retention rates and maximise profitability. Through comprehensive review of collected data, the analyst team proposes targeted user segmentation to optimise the proposed marketing perks.

Our team is filtering out the active users by narrowing down the user scope up to date with 04 January 2023 and after with only 7 latest sessions. This filtering resulted in 5998 active users. Out of these active users, there are 695 Male, 5292 females and 11 Others. The database only collected users' data from 2 countries, the United States of America and Canada. 1007 active users are located in Canada with the proportion of 115 Male to 892 Female. While the majority of active users are located in the United States of America with a total users of 4991. The gender proportions of active users are 580 Male to 4400 Female and 11 Others.

Based on the derived active users data, we are able to consolidate that female users are the most active users in gender proportion and most likely to be the higher spender compared to others. The marketing strategies or web design can be tapping into the female consumer segment for a stronger digital presence and therefore heightened up the customer retention experiences.

The current offered perks for active users lack specificity and might not be targeting the aimed customer group and hence it is lower in customer retention rate. Based on the provided data, our team filtered the customer data after 04 January 2023 with most recent 7 sessions only, we managed to identified 5 different user groups:

1. Seniors/ Pensioners Traveller
2. Business Travellers
3. Family Travellers with Children
4. Married couple without Children
5. Last Minute or Opportunistic Travellers

As per our discussion with the Marketing Department Leader in this project, Elena, several marketing perks are suggested:

1. Free hotel meal (breakfast, Half-board, Full-board, etc.)
2. Free checked bag (23 kg, 30kg, 40kg, etc.)
3. No cancellation fees (Hotel booking, Flight booking, Museum tickets, etc.)
4. Exclusive discounts (Early-bird discount, Last minute deals, loyalty discount)
5. One-night free hotel with flight (Transit Hotel, Airport Hotel, short term overnight stay)

To encourage higher participation and booking rate of customers, specific perks should be assigned to the targeted user group. Seniors traveller and most likely to be pensioners are flexible in travel dates and tend to go for a relaxed itinerary or longer stay. This group of customers most likely to be focusing on the seniors-friendly facilities such as accessible escalators or lifts in the higher hotel floors or simplified booking process as not all of the seniors can be tech savvy. Some seniors prefer to get airport transit arranged by the accommodation due to language barrier. Seniors tend to visit eateries or enjoy their meals in the walking distance from their accommodation as exploring further might be difficult for them due to health restrictions. Hence, suggested perks to this group of customers will most likely be free hotel meals which can range from just breakfast only to half board or full board depending on the hotel location. Lodge in natural reserves will be more likely to offer full board to customers as eating out can be a big challenge to non local travellers.

On the other hand, family travellers with children will be highly concerned about child-friendly locations such as theme parks or accessible to child friendly facilities such as baby changing rooms. Parents travelling with children will be spending more of their time taking care of the young child if there are no child care facilities. Children tend to be more impatient and intolerant towards harsh weather change or run down facilities. Unpredictable young children's behaviour can sometimes affect other travellers in the same area such as loud crying and running kids in the dining restaurant. Long dining duration such as buffets in the restaurant can be challenging for parents with young children. Childcare or children's playing areas will be the popular choice for parents with children. The same situation applies for family seating arrangements in the flight where parents prefer to sit with the children and close access to the facilities such as toilet or meal services. Parents usually travel with school children during the school holidays and the overcrowded facilities may cause dissatisfaction among the travellers. Perks assigned to this group of customers should be free additional checked bags. Travelling with children or in big family groups always resulted in bulky luggage. Bulk discount for family booking is a good option for this group of customers.

On the contrary, married couples without children are focusing more on privacy and a quiet environment during vacation. This customer group might be willing to pay more to enjoy private space or adult only facilities without enduring the noisy chaos. Couples without children will be more focused on the personal times spent between themselves and enjoying more in the city, nature or even endless museum trips. This customer group might be doing several cancellations due to unmatched schedules. Travel schedule is more likely to be inflexible if both of the couple are working adults. Perks such as free cancellations and early bird discount complement each other for this group of customers as customers will be more willing to book their trip earlier to save on the early bird discount while enjoying the free cancellations advantage in case the trip needs to be rescheduled later.

Lastly, the last minute or Opportunistic traveller is the traveller that booked the flight or hotel less than 7 days before the travel. This unique group of customers is most likely to be tech savvy, good at comparing and aim to get the best deals from online offers. This customer group is relatively new to the travel industry since the introduction of the online booking system and comparison websites. This group of customers tend to have more sessions login than other users as most of the login is to compare the price of their interesting trip. This group of customers is less inclined to cancel as the travel date is way too close to the booking date. Perks such as free cancellation will not be applicable to this group of customers as most of the booking is made less than a week before the actual departure date and hotel check in date. Some of the booking dates are less than a day before the actual departure or check in date which indicated the customer could be already at the departure location before the booking. Perks applicable to such a group will be more suitable as after the booked trip is performed. As such, free hotel meals for next hotel booking, free hotel stay with the next booked flight or discount for the next booked trip are suitable to attract these customers for next booking.

By segmenting the customers and applying the suitable perks to each group of customers will enhance the customer satisfaction and attract these customers to retain with the company for future booking. This in turn will enhance the competitiveness of TravelTide to be the first choice of customers when booking trips online.

Appendix with SQLs

```
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),


filtered_users AS (
  SELECT user_id, COUNT(*) FROM sessions_2023 s
  GROUP BY user_id
  HAVING COUNT(*) > 7
),


results AS
(SELECT s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end,
s.page_clicks,
 s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount, s.flight_booked, s.hotel_booked, s.cancellation,
 u.birthdate, u.gender, u.married, u.has_children, u.home_country, u.home_city,
u.home_airport, u.home_airport_lat, u.home_airport_lon, u.sign_up_date,
 f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,f.departure_time, f.return_time, f.checked_bags, f.trip_airline,
f.destination_airport_lat, f.destination_airport_lon,f.base_fare_usd,
 h.hotel_name, h.nights, h.rooms,h.check_in_time, h.hotel_per_room_usd AS
hotel_per_room_night_usd
 FROM sessions_2023 s
 LEFT JOIN users u
 ON s.user_id = u.user_id
 LEFT JOIN flights f
 ON s.trip_id = f.trip_id
 LEFT JOIN hotels h
 ON s.trip_id = h.trip_id
 WHERE s.user_id IN (SELECT user_id FROM filtered_users))

 SELECT *
 FROM results
=====================================================================
```

Family travellers/ trips with children

```
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),


filtered_users AS (
  SELECT user_id, COUNT(*) FROM sessions_2023 s
  GROUP BY user_id
  HAVING COUNT(*) > 7
),


results AS
(SELECT s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end,
s.page_clicks,
 s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount, s.flight_booked, s.hotel_booked, s.cancellation,
 u.birthdate, u.gender, u.married, u.has_children, u.home_country, u.home_city,
u.home_airport, u.home_airport_lat, u.home_airport_lon, u.sign_up_date,
 f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,f.departure_time, f.return_time, f.checked_bags, f.trip_airline,
f.destination_airport_lat, f.destination_airport_lon,f.base_fare_usd,
 h.hotel_name, h.nights, h.rooms,h.check_in_time, h.hotel_per_room_usd AS
hotel_per_room_night_usd
 FROM sessions_2023 s
 LEFT JOIN users u
 ON s.user_id = u.user_id
 LEFT JOIN flights f
 ON s.trip_id = f.trip_id
 LEFT JOIN hotels h
 ON s.trip_id = h.trip_id
 WHERE s.user_id IN (SELECT user_id FROM filtered_users))

 SELECT user_id, COUNT(DISTINCT session_id) AS num_sessions,
COUNT(DISTINCT trip_id) AS num_trips, has_children
 FROM results
 GROUP BY user_id, has_children
======================================================================
```

Hotel stay with nights < 0

```
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),
filtered_users AS (
  SELECT user_id, COUNT(*) FROM sessions_2023 s
  GROUP BY user_id
  HAVING COUNT(*) > 7
),


results AS
(SELECT s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end,
s.page_clicks,
 s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount, s.flight_booked, s.hotel_booked, s.cancellation,
 u.birthdate, u.gender, u.married, u.has_children, u.home_country, u.home_city,
u.home_airport, u.home_airport_lat, u.home_airport_lon, u.sign_up_date,
 f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,f.departure_time, f.return_time, f.checked_bags, f.trip_airline,
f.destination_airport_lat, f.destination_airport_lon,f.base_fare_usd,
 h.hotel_name, h.nights, h.rooms,h.check_in_time, h.hotel_per_room_usd AS
hotel_per_room_night_usd
 FROM sessions_2023 s
 LEFT JOIN users u
 ON s.user_id = u.user_id
 LEFT JOIN flights f
 ON s.trip_id = f.trip_id
 LEFT JOIN hotels h
 ON s.trip_id = h.trip_id
 WHERE s.user_id IN (SELECT user_id FROM filtered_users))

 SELECT *
 FROM results
 WHERE nights < 0
=================================================================
```

Booking cancellation

```
SELECT *
FROM flights f
INNER JOIN sessions s
ON f.trip_id = s. trip_id
WHERE f.seats = 0 AND s.cancellation = FALSE
AND s.trip_id NOT IN
(SELECT s.trip_id
FROM sessions s
WHERE s.trip_id IN
(SELECT s.trip_id
FROM flights f
INNER JOIN sessions s
ON f.trip_id = s.trip_id
WHERE f.seats = 0 AND s.cancellation = FALSE)
AND s.cancellation = TRUE)
```

==

1. Add in the cohort according to extracted data

==

1) modify the `results` CTE to include user-level aggregations
2) Examples:

   -Number of trips (`trip_id` count)

   -Total amount spent (`total_spent`)

   -Number of cancelled trips

   -Average trip duration

   -User demographics (e.g., age, gender)

   —---

- Query the `users` table to get a breakdown of users by gender, marital status, and whether they have children.

```
WITH sessions_2023 AS (

  SELECT *

  FROM sessions s

  WHERE s.session_start >= '2023-01-04'

),
filtered_users AS (

  SELECT user_id, COUNT(*) FROM sessions_2023 s

  GROUP BY user_id

  HAVING COUNT(*) > 7

),




results AS

(SELECT s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end,
s.page_clicks,

 s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount, s.flight_booked, s.hotel_booked, s.cancellation,

 u.birthdate, u.gender, u.married, u.has_children, u.home_country,
u.home_city, u.home_airport, u.home_airport_lat, u.home_airport_lon,
u.sign_up_date,

 f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,f.departure_time, f.return_time, f.checked_bags,
f.trip_airline, f.destination_airport_lat,
f.destination_airport_lon,f.base_fare_usd,

 h.hotel_name, h.nights, h.rooms,h.check_in_time, h.hotel_per_room_usd AS
hotel_per_room_night_usd

 FROM sessions_2023 s

 LEFT JOIN users u

 ON s.user_id = u.user_id

 LEFT JOIN flights f

 ON s.trip_id = f.trip_id
 LEFT JOIN hotels h

 ON s.trip_id = h.trip_id

 WHERE s.user_id IN (SELECT user_id FROM filtered_users))



 SELECT user_id, COUNT(DISTINCT session_id) AS num_sessions,
COUNT(DISTINCT trip_id) AS num_trips, gender, married, has_children

 FROM results

 GROUP BY user_id, gender, married, has_children
```

- What is the distribution of the user's birth year? Do you spot any irregularity there? What is special about birth year 2006? How would you calculate the age based on birth date?

  User with birth year 2006 = 43360

```
WITH sessions_2023 AS (

  SELECT *

  FROM sessions s

  WHERE s.session_start >= '2023-01-04'

),



filtered_users AS (

  SELECT user_id, COUNT(*) FROM sessions_2023 s

  GROUP BY user_id
  HAVING COUNT(*) > 7

),




results AS

(SELECT s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end,
s.page_clicks,

 s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount, s.flight_booked, s.hotel_booked, s.cancellation,

 u.birthdate, u.gender, u.married, u.has_children, u.home_country,
u.home_city, u.home_airport, u.home_airport_lat, u.home_airport_lon,
u.sign_up_date,

 f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,f.departure_time, f.return_time, f.checked_bags,
f.trip_airline, f.destination_airport_lat,
f.destination_airport_lon,f.base_fare_usd,

 h.hotel_name, h.nights, h.rooms,h.check_in_time, h.hotel_per_room_usd AS
hotel_per_room_night_usd

 FROM sessions_2023 s

 LEFT JOIN users u

 ON s.user_id = u.user_id

 LEFT JOIN flights f

 ON s.trip_id = f.trip_id

 LEFT JOIN hotels h

 ON s.trip_id = h.trip_id

 WHERE s.user_id IN (SELECT user_id FROM filtered_users))
```

```sql
SELECT
  EXTRACT(YEAR FROM birthdate) AS birth_year,
  COUNT(*) AS user_count
FROM users
GROUP BY birth_year
ORDER BY birth_year;
```

```sql
SELECT
  COUNT(*) AS user_count_2006
FROM users
WHERE EXTRACT(YEAR FROM birthdate) = 2006;
```

```sql
SELECT
  user_id, birthdate,
  DATE_PART('year', AGE(birthdate)) AS age
FROM users;
```

- You may define “customer age” as a period in months since the user signed up to the platform. What is the average “customer age” of TravelTide user?

  Average customer age of Travel Tide user = **18.99789**

```
WITH sessions_2023 AS (

  SELECT *

  FROM sessions s

  WHERE s.session_start >= '2023-01-04'

),

filtered_users AS (

  SELECT user_id, COUNT(*)

  FROM sessions_2023 s

  GROUP BY user_id

  HAVING COUNT(*) > 7

),

results AS (

  SELECT

    s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end, s.page_clicks,

    s.flight_discount, s.flight_discount_amount, s.hotel_discount, s.hotel_discount_amount,

    s.flight_booked, s.hotel_booked, s.cancellation,

    u.birthdate, u.gender, u.married, u.has_children, u.home_country, u.home_city,

    u.home_airport, u.home_airport_lat, u.home_airport_lon, u.sign_up_date,

    f.origin_airport, f.destination, f.destination_airport, f.seats, f.return_flight_booked,
    f.departure_time, f.return_time, f.checked_bags, f.trip_airline,
f.destination_airport_lat, f.destination_airport_lon, f.base_fare_usd,

    h.hotel_name, h.nights, h.rooms, h.check_in_time, h.hotel_per_room_usd
AS hotel_per_room_night_usd

  FROM sessions_2023 s

  LEFT JOIN users u ON s.user_id = u.user_id

  LEFT JOIN flights f ON s.trip_id = f.trip_id

  LEFT JOIN hotels h ON s.trip_id = h.trip_id

  WHERE s.user_id IN (SELECT user_id FROM filtered_users)

),


customer_age AS (

  SELECT

    user_id,

    DATE_PART('year', AGE(CURRENT_DATE, sign_up_date)) * 12 +

    DATE_PART('month', AGE(CURRENT_DATE, sign_up_date)) AS
customer_age_months

  FROM users

)




SELECT

  AVG(customer_age_months) AS avg_customer_age_months
```

```
FROM customer_age;
```

- What are the 10 most popular hotels? Include the information about the average duration of stay and average price before the discount. Do the same for most expensive hotels (top 10), hotels with the longest stays and etc.

  Most popular hotel

  ```
  -- Filter sessions from 2023

  WITH sessions_2023 AS (

    SELECT *

    FROM sessions s

    WHERE s.session_start >= '2023-01-04'

  ),



  -- Filter users with more than 7 sessions

  filtered_users AS (

    SELECT user_id, COUNT(*)

    FROM sessions_2023 s

    GROUP BY user_id

    HAVING COUNT(*) > 7

  ),



  -- Select detailed session data for filtered users
  ```

```
results AS (

  SELECT

    s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end, s.page_clicks,

    s.flight_discount, s.flight_discount_amount, s.hotel_discount, s.hotel_discount_amount,

    s.flight_booked, s.hotel_booked, s.cancellation,

    u.birthdate, u.gender, u.married, u.has_children, u.home_country, u.home_city,

    u.home_airport, u.home_airport_lat, u.home_airport_lon, u.sign_up_date,

    f.origin_airport, f.destination, f.destination_airport, f.seats, f.return_flight_booked,

    f.departure_time, f.return_time, f.checked_bags, f.trip_airline, f.destination_airport_lat, f.destination_airport_lon, f.base_fare_usd,

    h.hotel_name, h.nights, h.rooms, h.check_in_time, h.hotel_per_room_usd AS hotel_per_room_night_usd

  FROM sessions_2023 s

  LEFT JOIN users u ON s.user_id = u.user_id

  LEFT JOIN flights f ON s.trip_id = f.trip_id

  LEFT JOIN hotels h ON s.trip_id = h.trip_id

  WHERE s.user_id IN (SELECT user_id FROM filtered_users)

),



hotel_stats AS (

  SELECT
```

```
    h.hotel_name,
    COUNT(*) AS booking_count,
    AVG(h.nights) AS avg_duration_of_stay,
    AVG(h.hotel_per_room_usd) AS avg_price_before_discount
  FROM results r
  LEFT JOIN hotels h ON r.trip_id = h.trip_id
  GROUP BY h.hotel_name
)

-- Select top 10 most popular hotels
SELECT *
FROM hotel_stats
ORDER BY booking_count DESC
LIMIT 10;
```

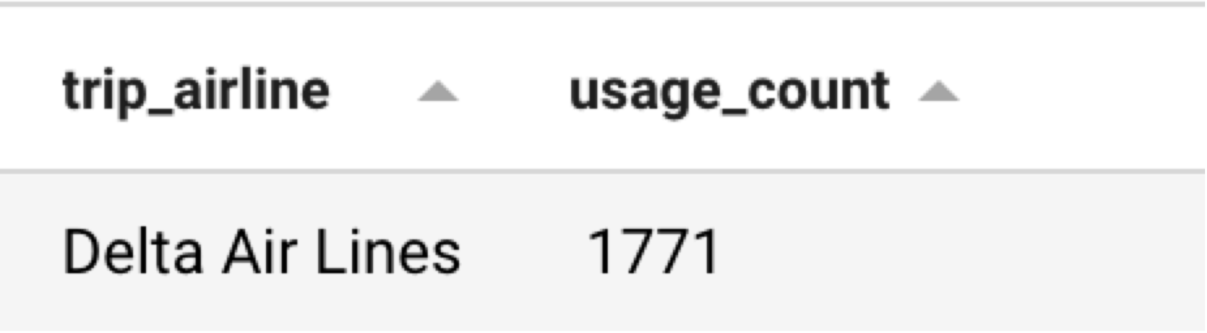
Most expensive hotel

```
-- Filter sessions from 2023
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),
-- Filter users with more than 7 sessions
filtered_users AS (
  SELECT user_id, COUNT(*)
  FROM sessions_2023 s
  GROUP BY user_id
  HAVING COUNT(*) > 7
),


-- Select detailed session data for filtered users
results AS (
  SELECT
    s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end, s.page_clicks,
    s.flight_discount, s.flight_discount_amount, s.hotel_discount, s.hotel_discount_amount,
    s.flight_booked, s.hotel_booked, s.cancellation,
    u.birthdate, u.gender, u.married, u.has_children, u.home_country, u.home_city,
    u.home_airport, u.home_airport_lat, u.home_airport_lon, u.sign_up_date,
    f.origin_airport, f.destination, f.destination_airport, f.seats, f.return_flight_booked,
    f.departure_time, f.return_time, f.checked_bags, f.trip_airline, f.destination_airport_lat, f.destination_airport_lon, f.base_fare_usd,
    h.hotel_name, h.nights, h.rooms, h.check_in_time, h.hotel_per_room_usd AS hotel_per_room_night_usd
  FROM sessions_2023 s
  LEFT JOIN users u ON s.user_id = u.user_id
  LEFT JOIN flights f ON s.trip_id = f.trip_id
  LEFT JOIN hotels h ON s.trip_id = h.trip_id
  WHERE s.user_id IN (SELECT user_id FROM filtered_users)
),

hotel_stats AS (
  SELECT
    h.hotel_name,
    COUNT(*) AS booking_count,
    AVG(h.nights) AS avg_duration_of_stay,
    AVG(h.hotel_per_room_usd) AS avg_price_before_discount
  FROM results r
  LEFT JOIN hotels h ON r.trip_id = h.trip_id
  GROUP BY h.hotel_name
)

-- Select top 10 most expensive hotels
SELECT *
```

```
FROM hotel_stats
ORDER BY avg_price_before_discount DESC
LIMIT 10;
```

Longest stayed Hotel

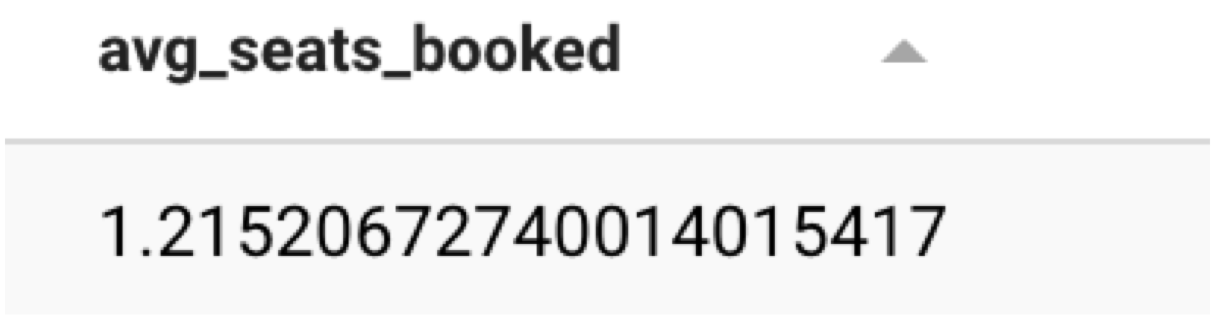
```
-- Filter sessions from 2023
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),

-- Filter users with more than 7 sessions
filtered_users AS (
  SELECT user_id, COUNT(*)
  FROM sessions_2023 s
  GROUP BY user_id
  HAVING COUNT(*) > 7
),

-- Select detailed session data for filtered users
results AS (
  SELECT

    s.session_id, s.user_id, s.trip_id, s.session_start, s.session_end,
s.page_clicks,

    s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount,

    s.flight_booked, s.hotel_booked, s.cancellation,

    u.birthdate, u.gender, u.married, u.has_children, u.home_country,
u.home_city,

    u.home_airport, u.home_airport_lat, u.home_airport_lon, u.sign_up_date,

    f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,

    f.departure_time, f.return_time, f.checked_bags, f.trip_airline,
f.destination_airport_lat, f.destination_airport_lon, f.base_fare_usd,

    h.hotel_name, h.nights, h.rooms, h.check_in_time, h.hotel_per_room_usd
AS hotel_per_room_night_usd

  FROM sessions_2023 s

  LEFT JOIN users u ON s.user_id = u.user_id

  LEFT JOIN flights f ON s.trip_id = f.trip_id

  LEFT JOIN hotels h ON s.trip_id = h.trip_id

  WHERE s.user_id IN (SELECT user_id FROM filtered_users)

),



hotel_stats AS (

  SELECT

    h.hotel_name,
    COUNT(*) AS booking_count,

    AVG(h.nights) AS avg_duration_of_stay,

    AVG(h.hotel_per_room_usd) AS avg_price_before_discount

  FROM results r

  LEFT JOIN hotels h ON r.trip_id = h.trip_id

  GROUP BY h.hotel_name

)


-- Select top 10 most longest stay hotel

SELECT *

FROM hotel_stats

ORDER BY avg_duration_of_stay DESC

LIMIT 10;
```



trip_airline	usage_count
Delta Air Lines	1771

```
WITH sessions_2023 AS (

 SELECT *

FROM sessions s
```

```
WHERE s.session_start >= '2023-01-04'
),

filtered_users AS (
  SELECT user_id
 FROM sessions_2023
  GROUP BY user_id
 HAVING COUNT(*) > 7
),
combined_data AS (
SELECT s.session_id, s.user_id, s.trip_id, s.session_start, s.flight_booked, f.trip_airline
FROM sessions_2023 s
LEFT JOIN flights f ON s.trip_id = f.trip_id
WHERE s.user_id IN (SELECT user_id FROM filtered_users)
),

last_six_months_data AS (
SELECT *

FROM combined_data

WHERE session_start >= (SELECT MAX(session_start) FROM combined_data) -
INTERVAL '6 MONTH'

AND flight_booked = TRUE

)



-- Final query to get the most used airline

SELECT trip_airline, COUNT(*) AS usage_count

FROM last_six_months_data

GROUP BY trip_airline

ORDER BY usage_count DESC

LIMIT 1;
```



```
avg_seats_booked
1.21520672740014015417
```

```
                    WITH sessions_2023 AS (

                     SELECT *
```

```
  FROM sessions s

  WHERE s.session_start >= '2023-01-04'

),



filtered_users AS (

  SELECT user_id

  FROM sessions_2023

  GROUP BY user_id

  HAVING COUNT(*) > 7

),



combined_data AS (

  SELECT

    s.session_id,

    s.user_id,

    s.trip_id,

    s.session_start,

    s.flight_booked,
    f.seats
  FROM sessions_2023 s
  LEFT JOIN flights f ON s.trip_id = f.trip_id
  WHERE s.user_id IN (SELECT user_id FROM filtered_users)
    AND s.flight_booked = TRUE
),

average_seats AS (
  SELECT AVG(seats) AS avg_seats_booked
  FROM combined_data
  WHERE seats IS NOT NULL
)

-- Final result
SELECT avg_seats_booked
FROM average_seats;
```

```
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),

filtered_users AS (
  SELECT user_id
  FROM sessions_2023
  GROUP BY user_id
  HAVING COUNT(*) > 7
),

combined_data AS (
  SELECT
    s.session_id,
    s.user_id,
    f.origin_airport,
    f.destination_airport,
    f.base_fare_usd,
    f.departure_time,
    CASE
      WHEN EXTRACT(MONTH FROM f.departure_time) IN (12, 1, 2) THEN 'Winter'
      WHEN EXTRACT(MONTH FROM f.departure_time) IN (3, 4, 5) THEN 'Spring'
      WHEN EXTRACT(MONTH FROM f.departure_time) IN (6, 7, 8) THEN 'Summer'
      WHEN EXTRACT(MONTH FROM f.departure_time) IN (9, 10, 11) THEN 'Fall'
    END AS season
  FROM sessions_2023 s
  JOIN flights f ON s.trip_id = f.trip_id
  WHERE s.user_id IN (SELECT user_id FROM filtered_users)
    AND f.base_fare_usd IS NOT NULL
),

route_season_stats AS (
  SELECT
    origin_airport,
```

```sql
    destination_airport,

    season,

    COUNT(*) AS num_flights,

    AVG(base_fare_usd) AS avg_price,

    STDDEV(base_fare_usd) AS price_stddev

  FROM combined_data

  GROUP BY origin_airport, destination_airport, season

)



-- Final selection

SELECT

  origin_airport,

  destination_airport,

  season,

  num_flights,

  avg_price,

  price_stddev,
  (price_stddev / NULLIF(avg_price, 0)) * 100 AS coefficient_of_variation -- Coefficient of Variation as a percentage

FROM route_season_stats

ORDER BY origin_airport, destination_airport, season;
```

Grouping sessions ID into user ID

```
WITH sessions_2023 AS (

  SELECT *

  FROM sessions s

  WHERE s.session_start >= '2023-01-04'

),


filtered_users AS (

  SELECT user_id

  FROM sessions_2023

  GROUP BY user_id

  HAVING COUNT(*) > 7

),


results AS (

  SELECT
```

```
s.session_id,
s.user_id,
s.trip_id,
s.session_start,
s.session_end,
s.page_clicks,
s.flight_discount,
s.flight_discount_amount,
s.hotel_discount,
s.hotel_discount_amount,
s.flight_booked,
s.hotel_booked,
s.cancellation,
u.birthdate,
u.gender,
u.married,
u.has_children,
u.home_country,
u.home_city,
u.home_airport,
u.home_airport_lat,
u.home_airport_lon,
  u.sign_up_date,
  f.origin_airport,
  f.destination,
  f.destination_airport,
  f.seats,
  f.return_flight_booked,
  f.departure_time,
  f.return_time,
  f.checked_bags,
  f.trip_airline,
  f.destination_airport_lat,
  f.destination_airport_lon,
  f.base_fare_usd,
  h.hotel_name,
  h.nights,
  h.rooms,
  h.check_in_time,
  h.hotel_per_room_usd AS hotel_per_room_night_usd
FROM sessions_2023 s
LEFT JOIN users u ON s.user_id = u.user_id
LEFT JOIN flights f ON s.trip_id = f.trip_id
LEFT JOIN hotels h ON s.trip_id = h.trip_id
```

```
  WHERE s.user_id IN (SELECT user_id FROM filtered_users)
)

SELECT
  user_id,
  STRING_AGG(session_id, ', ') AS session_ids,
  COUNT(session_id) AS session_count,
  AVG(page_clicks) AS avg_page_clicks,
  SUM(flight_discount_amount) AS total_flight_discount_amount,
  SUM(hotel_discount_amount) AS total_hotel_discount_amount,
  SUM(CASE WHEN flight_booked THEN 1 ELSE 0 END) AS flights_booked,
  SUM(CASE WHEN hotel_booked THEN 1 ELSE 0 END) AS hotels_booked,
  SUM(CASE WHEN cancellation THEN 1 ELSE 0 END) AS cancellations
FROM results
GROUP BY user_id
ORDER BY user_id;
```

Average seats booked

```
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),

filtered_users AS (
  SELECT user_id
  FROM sessions_2023
  GROUP BY user_id
  HAVING COUNT(*) > 7
),

combined_data AS (
  SELECT
    s.session_id,
    s.user_id,
    s.trip_id,
    s.session_start,
    s.flight_booked,
    f.seats
  FROM sessions_2023 s
  LEFT JOIN flights f ON s.trip_id = f.trip_id
  WHERE s.user_id IN (SELECT user_id FROM filtered_users)
    AND s.flight_booked = TRUE
```

```
),

average_seats AS (
  SELECT AVG(seats) AS avg_seats_booked
  FROM combined_data
  WHERE seats IS NOT NULL
)

-- Final result
SELECT avg_seats_booked
FROM average_seats;
```

Hotel stay night <0

```
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),


filtered_users AS (
  SELECT user_id, COUNT(*) FROM sessions_2023 s
  GROUP BY user_id

  HAVING COUNT(*) > 7

),




results AS

(SELECT s.session_id, s.user_id, s.trip_id, s.session_start,
s.session_end, s.page_clicks,

 s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount, s.flight_booked, s.hotel_booked,
s.cancellation,

 u.birthdate, u.gender, u.married, u.has_children, u.home_country,
u.home_city, u.home_airport, u.home_airport_lat, u.home_airport_lon,
u.sign_up_date,

 f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,f.departure_time, f.return_time, f.checked_bags,
f.trip_airline, f.destination_airport_lat,
f.destination_airport_lon,f.base_fare_usd,

 h.hotel_name, h.nights, h.rooms,h.check_in_time,
h.hotel_per_room_usd AS hotel_per_room_night_usd

 FROM sessions_2023 s

 LEFT JOIN users u

 ON s.user_id = u.user_id

 LEFT JOIN flights f

 ON s.trip_id = f.trip_id

 LEFT JOIN hotels h
 ON s.trip_id = h.trip_id
 WHERE s.user_id IN (SELECT user_id FROM filtered_users))


 SELECT *
 FROM results
 WHERE nights < 0
```

User ID with children

```
WITH sessions_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start >= '2023-01-04'
),



filtered_users AS (
  SELECT user_id, COUNT(*) FROM sessions_2023 s
  GROUP BY user_id
  HAVING COUNT(*) > 7
),



results AS
```

```
(SELECT s.session_id, s.user_id, s.trip_id, s.session_start,
s.session_end, s.page_clicks,

 s.flight_discount, s.flight_discount_amount, s.hotel_discount,
s.hotel_discount_amount, s.flight_booked, s.hotel_booked,
s.cancellation,

 u.birthdate, u.gender, u.married, u.has_children, u.home_country,
u.home_city, u.home_airport, u.home_airport_lat, u.home_airport_lon,
u.sign_up_date,

 f.origin_airport, f.destination, f.destination_airport, f.seats,
f.return_flight_booked,f.departure_time, f.return_time, f.checked_bags,
f.trip_airline, f.destination_airport_lat,
f.destination_airport_lon,f.base_fare_usd,

 h.hotel_name, h.nights, h.rooms,h.check_in_time,
h.hotel_per_room_usd AS hotel_per_room_night_usd

 FROM sessions_2023 s

 LEFT JOIN users u

 ON s.user_id = u.user_id

 LEFT JOIN flights f

 ON s.trip_id = f.trip_id

 LEFT JOIN hotels h

 ON s.trip_id = h.trip_id

 WHERE s.user_id IN (SELECT user_id FROM filtered_users))



 SELECT user_id, COUNT(DISTINCT session_id) AS num_sessions,
COUNT(DISTINCT trip_id) AS num_trips, has_children

 FROM results

 GROUP BY user_id, has_children
```

Cancelled trip

```
SELECT *
FROM flights f
INNER JOIN sessions s
ON f.trip_id = s. trip_id
WHERE f.seats = 0 AND s.cancellation = FALSE
AND s.trip_id NOT IN
(SELECT s.trip_id
FROM sessions s
WHERE s.trip_id IN
(SELECT s.trip_id
FROM flights f
INNER JOIN sessions s
ON f.trip_id = s.trip_id
WHERE f.seats = 0 AND s.cancellation = FALSE)
AND s.cancellation = TRUE)
```