Executive Summary of TravelTide Project
By Sanchita Neupane

In the fast-paced world of online travel, TravelTide stands out as a dynamic newcomer with the potential to make a significant impact in the industry. Supported by positive customer feedback and validated by industry analysts, TravelTide offers an expansive travel inventory that grants customers access to a wide range of options. The company's focus on exceptional service and its attention to emerging trends position TravelTide as a strong competitor in the competitive e-booking space.

The TravelTide Customer Segmentation Project reflects a strategic collaboration between the Marketing and Data Analytics teams. The goal of this collaboration is to enhance customer retention and engagement through the implementation of a personalized rewards program. This project aims to analyze customer behavior and preferences to design personalized rewards tailored to each customer group's needs and preferences.

Objectives:

As part of the Analytics team, my role is to support Elena, a seasoned expert in marketing and customer retention, in developing and implementing the rewards program. The project's primary objective is to enhance customer retention and engagement by introducing a personalized rewards program.

Methodology:

The analysis involves data exploration, segmentation, and generating recommendations. SQL was used to extract and explore data from the company database, while Tableau was used for data visualization.

Project Workflow:

- Customer Segmentation: Utilized SQL to segment customers into cohorts.
- Metric Calculation: Employed SQL to calculate basic statistics, central tendencies, and index metrics for the segmented customers.
- Demographic Analysis: Investigated the demographics of the customer base.
- Data Visualization: Used Tableau to create data visualizations.
- Conclusions and Recommendations: Summarized the project's objectives, findings, and recommendations based on customer segmentations.

Business Questions:

The report addresses the following questions:

- What are the behaviors and preferences of different customer groups, and how can these be used to generate customized promotions?
- Which perks are suitable for each group of people to increase company revenue?
- What strategies should be applied to each group for promoting the company based on data-driven decisions?

Scope:

The project includes users with more than seven sessions, and it covers data from after the New Year's holiday (January 4, 2023) to the most recent available data. This fixed time frame helps to analyze customer behavior effectively.

Customer Segmentation Approaches:

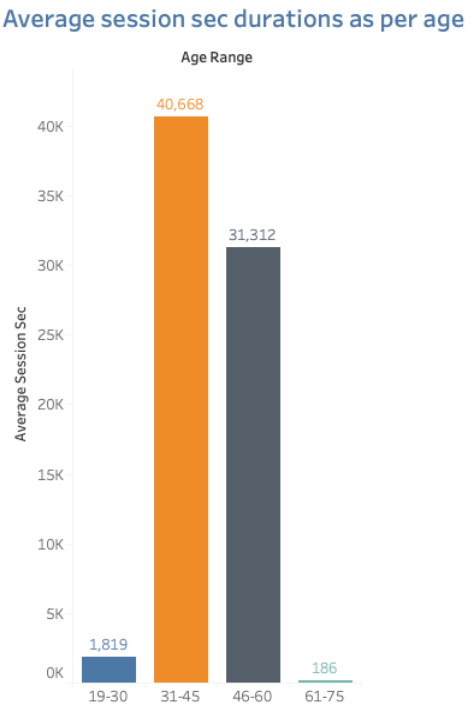
- Free Hotel Meal: Target customers who are interested in discounts with free hotel meals.
- Free Checked Bags: Offer this perk to travelers who carry many bags during their travels, especially those traveling with children.
- No Cancellation Fee: Target customers who frequently cancel trips.
- Exclusive Discount: Provide exclusive discounts to those who actively seek flight discounts.
- Free Hotel Night with Flight: Offer this perk to loyal customers who book both flights and hotels together.

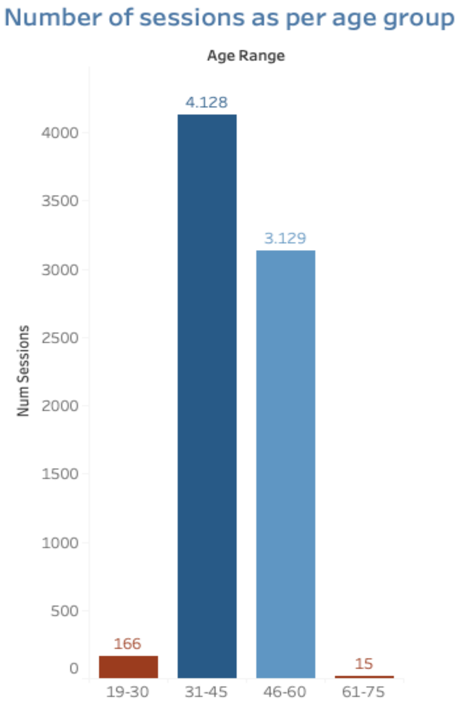
Metrics Used in the Analysis:

- Session Intensity Index: Measures average session duration and the number of sessions.
- Hotel Hunter Index: Measures average dollars saved per night and the number of hotel bookings.
- Bargain Hunter Index: Measures average dollars saved per kilometer and the number of flight bookings.

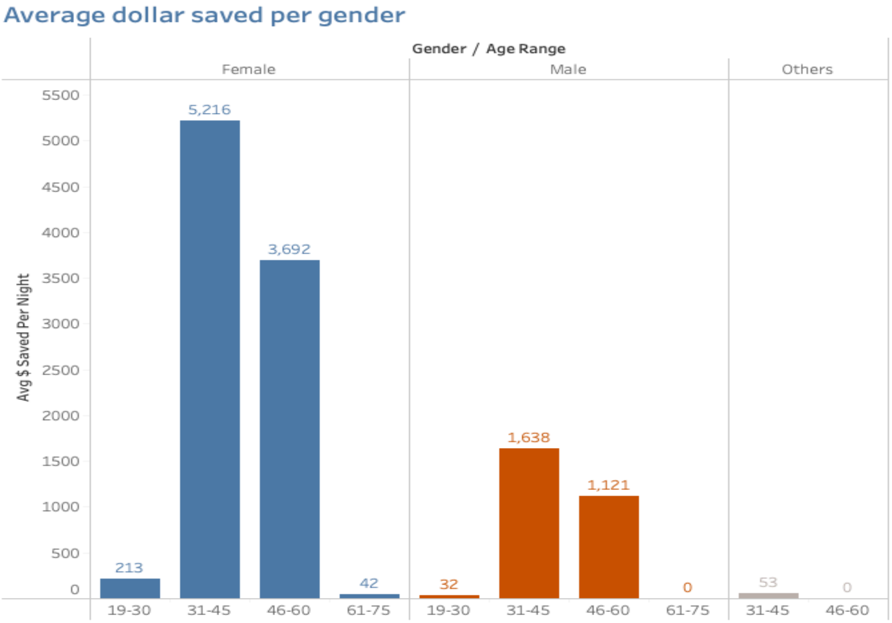
This analysis provides insights into customer habits and preferences, offering recommendations for customized promotions and personalized rewards programs to increase customer engagement and retention.

Demographic Analysis:

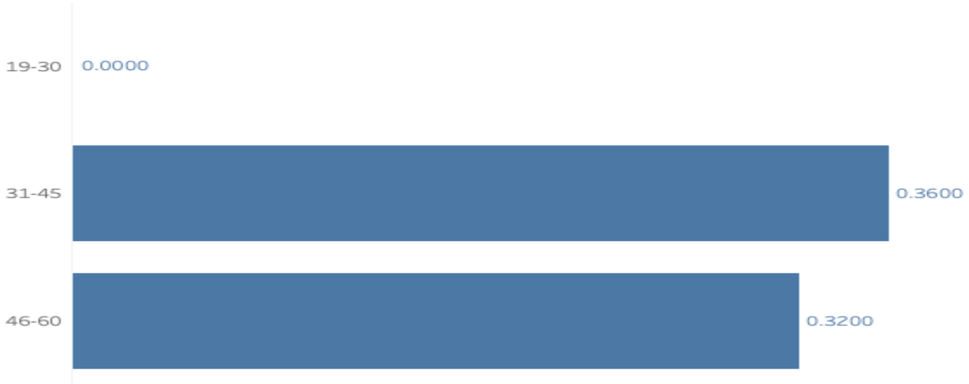


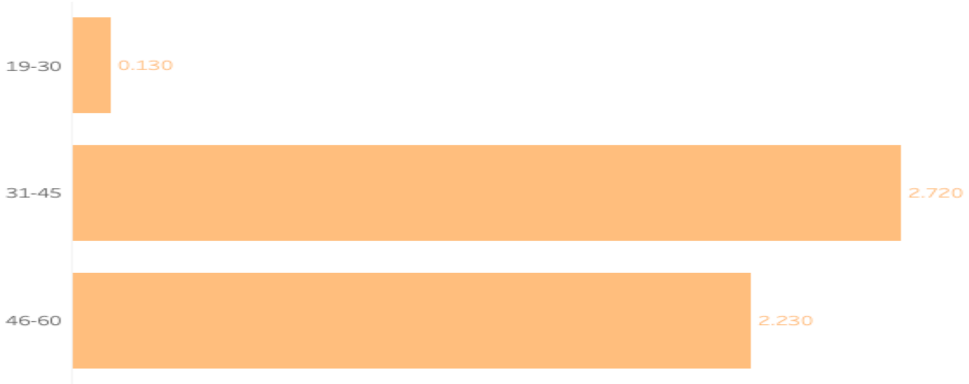


Based on the data, we observe that the age group 31-60 are the most active users of the TravelTide app.

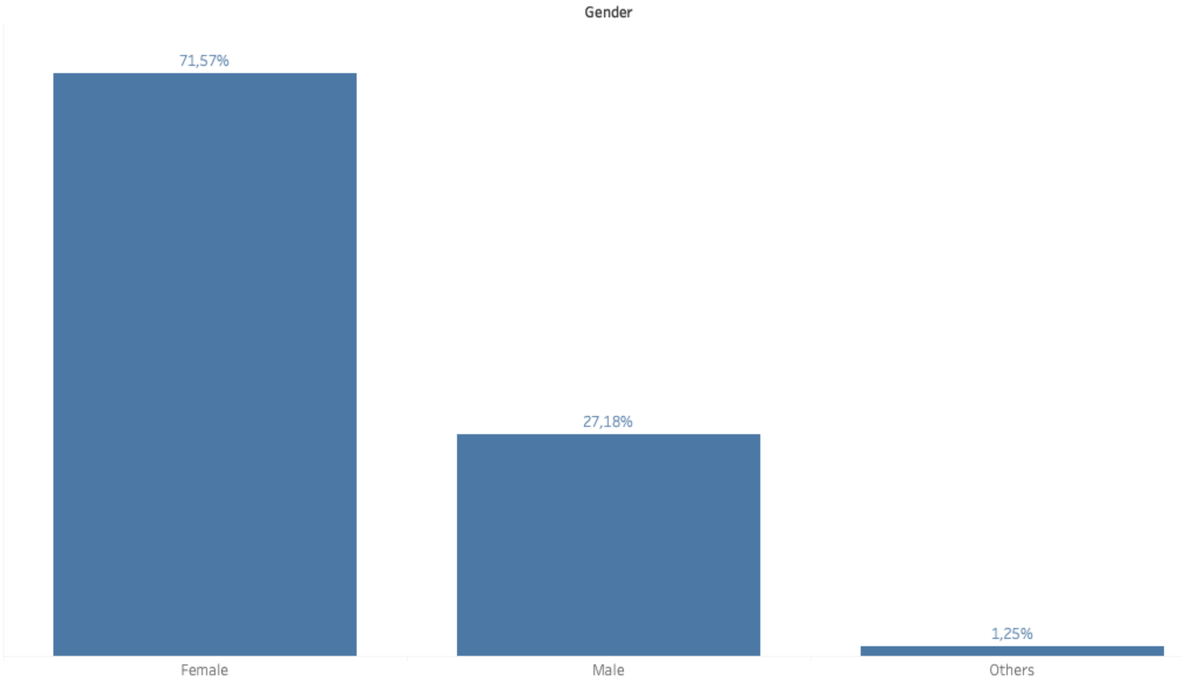


Female users in the 31-60 age range are particularly focused on saving money on hotel bookings.

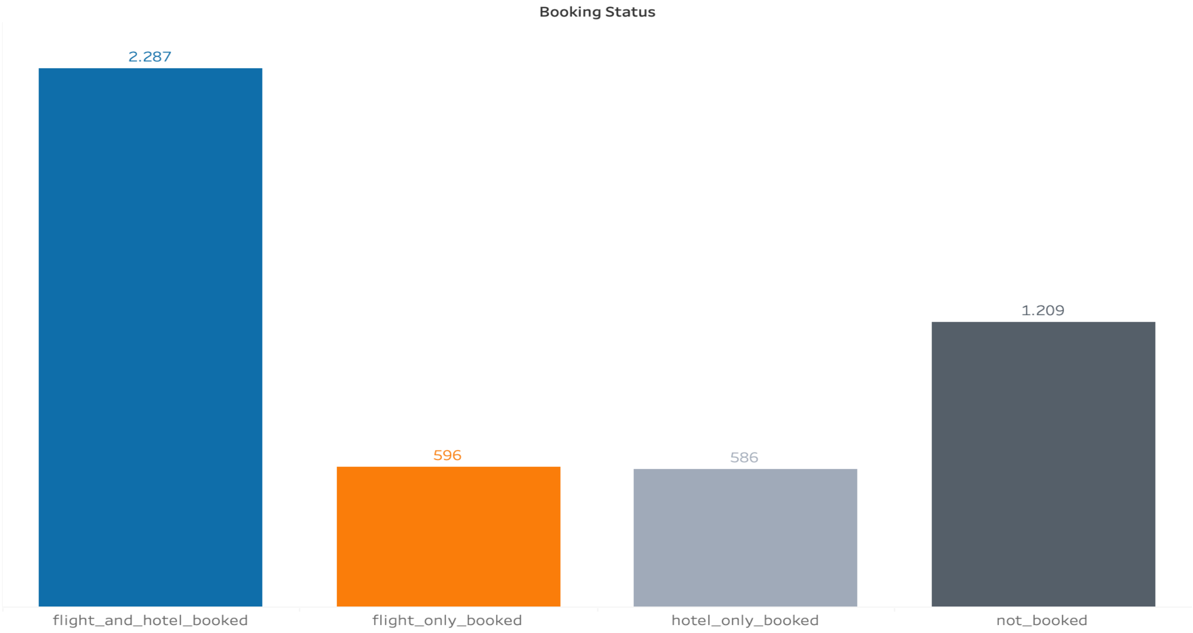

Average $ saved per km by married people having children


Average $ per km saved by umarried people having no children

Single individuals who are not married and do not have children tend to seek out flight discounts more frequently, showing a keen interest in finding the best prices.


Cancelled trip as per gender

Regarding trip cancellations, the majority of cancellations (71.57%) were made by female customers.



Flight and hotel booked as per users

Customers who book both hotel and flight simultaneously are identified as loyal and prestigious customers of the company. They prioritize the overall service quality rather than solely focusing on discounts.

Key Findings

1. Demographic Analysis: The most active searchers on the app are people aged 31-60, with female users more focused on saving money from hotel bookings. Single individuals without children are particularly interested in flight discounts, continuously hunting for cheaper prices.
2. Perks Allocation:
   - Free Hotel Meal: Appeals to customers looking for discounts.
   - Free Checked Bags: Attracts travelers with children or multiple bags.
   - No Cancellation Fee: Targets frequent trip cancellers.
   - Exclusive Discount: Draws active flight discount seekers.
   - Free Hotel Night with Flight: Attracts loyal customers who book flights and hotels together.
3. Metric Analysis: Metrics such as the Hotel Hunter Index and Bargain Hunter Index provide insights into customer habits, revealing preferences for saving money on hotels and flights.

Recommendations:

1. Offer Tailored Perks:
    - Free Hotel Meals: Provide free hotel meals to customers who are primarily focused on discounts. This can enhance their experience and incentivize them to continue booking through TravelTide.
    - Free Checked Bags: Offer free checked bags to cater to travelers with children or those who frequently travel with multiple bags. This will make their travel experience smoother and more convenient.
    - No Cancellation Fee: Target customers who frequently cancel trips with a no-cancellation fee perk. This flexibility will increase their satisfaction and trust in the service.
    - Exclusive Discounts: Extend exclusive discounts to single individuals without children who consistently seek cheaper flight prices. This can encourage them to book more frequently through TravelTide.
    - Free Hotel Night with Flight: Provide loyal customers, who book flights and hotels together, with a free hotel night. These customers are not as focused on discounts, so rewarding their loyalty can strengthen their connection with the brand.
2. Customized Marketing Campaigns:
    - Segmented Campaigns: Create marketing campaigns that specifically target the needs and preferences of each customer segment. Emphasize the perks available to them to drive engagement and retention.
    - Personalized Communication: Send personalized notifications, such as emails or app alerts, that highlight the specific rewards and benefits each customer can receive.
3. Monitor Program Performance:
    - Track Key Metrics: Continuously monitor key performance indicators such as engagement rates, customer satisfaction, and retention rates.
    - Adjust Program as Needed: Use data insights to adjust and refine the rewards program to ensure it aligns with customer preferences and drives the desired outcomes.
4. Customer Feedback and Refinement:
    - Collect Feedback: Regularly gather feedback from customers regarding the rewards program to understand their experiences and preferences.
    - Refine Perks: Use customer feedback to refine and improve the program, making adjustments as necessary to better meet their needs.
5. Reward Loyal Customers:
    - VIP Benefits: Offer special perks and VIP benefits to loyal customers to foster a sense of exclusivity and encourage them to continue booking through TravelTide.

By implementing these recommendations, we can create a more personalized and rewarding experience for customers, leading to higher retention rates and increased revenue for the company.

Appendix

SQL Queries

```
SELECT * FROM flights
LEFT JOIN hotels ON flights.trip_id=hotels.trip_id
LEFT JOIN sessions ON hotels.trip_id=sessions.trip_id
LEFT JOIN users ON sessions.user_id=users.user_id
WHERE session_start >= '2023-01-04' AND
users.user_id IN
  (SELECT user_id FROM sessions WHERE cancellation is FALSE
   GROUP BY 1
   HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7 )
;

WITH sessions_filtered AS (
SELECT * FROM sessions
WHERE session_start >= '2023-01-04' AND
user_id IN
  (SELECT user_id FROM sessions WHERE cancellation is FALSE
   GROUP BY 1
   HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7 )
),
status_table AS (SELECT users.user_id,CASE WHEN has_children = true AND married = true THEN 'Married_with_children'
                                                  WHEN has_children = false AND married = false THEN 'Single'
            WHEN has_children = true AND married = false THEN 'Single_parent'
            ELSE 'Not_specified'
       END AS Status
FROM users
LEFT JOIN sessions_filtered ON users.user_id=sessions_filtered.user_id)

SELECT COUNT(DISTINCT user_id) AS total_users,
       Status
FROM status_table
GROUP BY Status
ORDER BY 1 DESC;

WITH sessions_filtered AS (
SELECT * FROM sessions
WHERE session_start > '2023-01-04'
AND user_id IN
  (SELECT user_id FROM sessions WHERE cancellation is FALSE
   GROUP BY 1
   HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7 )
),
age_table AS (SELECT *,
                                                   EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) AS age
FROM users
LEFT JOIN sessions_filtered ON
users.user_id=sessions_filtered.user_id
)

SELECT CASE WHEN age BETWEEN 16 AND 30 THEN '19-30'
                                                                      WHEN age BETWEEN 31 AND 45 THEN '31-45'
            WHEN age BETWEEN 46 AND 60 THEN '46-60'
            WHEN age BETWEEN 61 AND 75 THEN '61-75'
            WHEN age BETWEEN 76 AND 90 THEN '76-90'
            ELSE 'above 90'
       END AS age_range,
       COUNT(*) AS total_users
```

```
FROM age_table
GROUP BY 1
ORDER BY 2 DESC
;

WITH sessions_filtered AS (SELECT user_id
FROM sessions
WHERE session_start >= '2023-01-04'
AND user_id IN
 (SELECT user_id FROM sessions WHERE cancellation is FALSE
  GROUP BY 1
  HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7 )
),
customer_table AS(SELECT users.user_id,
   session_end - session_start AS duration_session,
   users.sign_up_date,
   CASE
      WHEN EXTRACT(DAY FROM users.sign_up_date) = 1 THEN 'new_customer'
      ELSE 'regular_customer'
   END AS customer_nature
FROM
   sessions
LEFT JOIN
   users ON sessions.user_id = users.user_id
LEFT JOIN
                        sessions_filtered ON sessions.user_id=sessions_filtered.user_id
GROUP BY 1,2)

SELECT COUNT(DISTINCT user_id) AS total_customer,
                                    customer_nature
FROM customer_table
GROUP BY 2
;

WITH customer_table AS(SELECT users.user_id,
                                    SUM(CASE WHEN EXTRACT(YEAR FROM session_start) = 2021 THEN 1 ELSE 0 END) AS
users_2021,
     SUM(CASE WHEN EXTRACT(YEAR FROM session_start) = 2022 THEN 1 ELSE 0 END) AS users_2022,
                                    SUM(CASE WHEN EXTRACT(YEAR FROM session_start) = 2023 THEN 1 ELSE 0 END) AS
users_2023
FROM users
LEFT JOIN sessions ON
users.user_id=sessions.user_id
WHERE cancellation = false
GROUP BY 1)

SELECT  COUNT(*) AS loyal_customers
FROM customer_table
WHERE users_2021 > 0 AND users_2022 > 0 AND users_2023 > 0
;


WITH customer_table AS(SELECT users.user_id,
                                    SUM(CASE WHEN EXTRACT(YEAR FROM session_start) = 2021 THEN 1 ELSE 0 END) AS
users_2021,
     SUM(CASE WHEN EXTRACT(YEAR FROM session_start) = 2022 THEN 1 ELSE 0 END) AS users_2022,
                                    SUM(CASE WHEN EXTRACT(YEAR FROM session_start) = 2023 THEN 1 ELSE 0 END) AS
users_2023
FROM users
LEFT JOIN sessions ON
users.user_id=sessions.user_id
WHERE cancellation = false
GROUP BY 1)

SELECT customer_table.user_id,
                                    SUM(base_fare_usd) AS total_revenue,
     CASE WHEN EXTRACT(YEAR FROM departure_time) = 2021 THEN '2021'
                                   WHEN EXTRACT(YEAR FROM departure_time) = 2022 THEN '2022'
         WHEN EXTRACT(YEAR FROM departure_time) = 2023 THEN '2023'
     END AS years,
    flight_discount_amount * base_fare_usd AS discount_amount

FROM customer_table
LEFT JOIN sessions ON customer_table.user_id=sessions.user_id
LEFT JOIN flights ON sessions.trip_id=flights.trip_id
WHERE users_2021 > 0 AND users_2022 > 0 AND users_2023 > 0
GROUP BY 1,3,4
;

WITH sessions_filtered AS (
SELECT * FROM sessions
WHERE session_start > '2023-01-04'
AND user_id IN
(SELECT user_id FROM sessions WHERE cancellation is FALSE
GROUP BY 1
HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 5 )
)

SELECT CASE WHEN flight_booked = true AND hotel_booked = true THEN 'flight_and_hotel_booked'
                                                         WHEN flight_booked = true AND hotel_booked = false THEN
'flight_only_booked'
       WHEN flight_booked = false AND hotel_booked = true THEN 'hotel_only_booked'
                            ELSE 'not_booked'
    END AS booking_status,
    COUNT(DISTINCT user_id) AS total_users
FROM sessions_filtered
GROUP BY 1
;

WITH sessions_filtered AS (
SELECT * FROM sessions
WHERE session_start > '2023-01-04'
AND user_id IN
(SELECT user_id FROM sessions WHERE cancellation is FALSE
GROUP BY 1
HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7 )
)

SELECT CASE WHEN flight_booked = true AND hotel_booked = true THEN 'flight_and_hotel_booked'
                                                         WHEN flight_booked = true AND hotel_booked = false THEN
'flight_only_booked'
       WHEN flight_booked = false AND hotel_booked = true THEN 'hotel_only_booked'
                            ELSE 'not_booked'
    END AS booking_status,
    COUNT(DISTINCT user_id) AS total_users
FROM sessions_filtered
GROUP BY 1
;

WITH sessions_filtered AS (
SELECT * FROM sessions
WHERE session_start > '2023-01-04'
AND user_id IN
(SELECT user_id FROM sessions WHERE cancellation is FALSE
GROUP BY 1
HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7 )
)


SELECT COUNT(DISTINCT user_id) AS total_users,
    SUM(CASE WHEN (flight_discount is true AND hotel_discount is true) THEN 1 ELSE 0 END)AS discount_oriented_customer,
    SUM(CASE WHEN (flight_discount is false AND hotel_discount is false) THEN 1 ELSE 0 END)AS
not_discount_oriented_customer

FROM sessions
WHERE session_start > '2023-01-04'
AND user_id IN
(SELECT user_id FROM sessions WHERE cancellation is FALSE
GROUP BY 1
HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 5)
```

```
;

SELECT COUNT(DISTINCT user_id) AS total_users,
    SUM(CASE WHEN (hotel_discount is true) THEN 1 ELSE 0 END)AS discount_oriented_customer,
    SUM(CASE WHEN (hotel_discount is false) THEN 1 ELSE 0 END)AS not_discount_oriented_customer

FROM sessions
WHERE session_start > '2023-01-04'
AND user_id IN
(SELECT user_id FROM sessions WHERE cancellation is FALSE
GROUP BY 1
HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7
)
;

SELECT COUNT(sessions.trip_id) AS total_users,
                              SUM(CASE WHEN (hotel_discount is true) THEN 1 ELSE 0 END)AS discount_oriented_customer,
    SUM(CASE WHEN (hotel_discount is false) THEN 1 ELSE 0 END)AS not_discount_oriented_customer

FROM sessions
WHERE session_start > '2023-01-04' AND sessions.trip_id IS NOT NULL
AND user_id IN
(SELECT user_id FROM sessions WHERE cancellation is FALSE
GROUP BY 1
HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7)
;

WITH sessions_filtered AS (
 SELECT * FROM sessions
 WHERE session_start > '2023-01-04'
 AND user_id IN
 (SELECT user_id FROM sessions WHERE cancellation IS FALSE
  GROUP BY 1
  HAVING SUM(CASE WHEN (flight_booked IS TRUE OR hotel_booked IS TRUE) THEN 1 ELSE 0 END) > 7 )
 )

SELECT count(DISTINCT user_id) FROM sessions_filtered


--Session Intensity Index:
WITH sessions_filtered AS (
 SELECT * FROM sessions
 WHERE session_start > '2023-01-04'
 AND user_id IN
 (SELECT user_id FROM sessions WHERE cancellation is FALSE
  GROUP BY 1
  HAVING SUM(CASE WHEN (flight_booked IS TRUE OR hotel_booked IS TRUE) THEN 1 ELSE 0 END) > 7)
 )

SELECT s.user_id,
                              CASE WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate)
BETWEEN 16 AND 30 THEN '19-30'
                                          WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate)
BETWEEN 31 AND 45 THEN '31-45'
       WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 46 AND 60 THEN
'46-60'
       WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 61 AND 75 THEN
'61-75'
       WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 76 AND 90 THEN
'76-90'
       ELSE 'above 90'
    END AS age_range,
    gender,
    married,
    has_children,
    home_country,
    home_city,
                              COUNT(*) AS num_sessions, AVG(DATE_PART('second', session_end - session_start)) AS
average_session_sec
FROM sessions_filtered s
```

```
LEFT JOIN users u ON u.user_id = s.user_id
GROUP BY 1,2,3,4,5,6,7
ORDER BY 2 DESC


--Hotel Hunter Index ( average $ saved per night per user):
WITH sessions_filtered AS (
  SELECT * FROM sessions
  WHERE session_start > '2023-01-04'
  AND user_id IN
  (SELECT user_id FROM sessions WHERE cancellation IS FALSE
   GROUP BY 1
   HAVING SUM(CASE WHEN (flight_booked IS TRUE OR hotel_booked IS TRUE) THEN 1 ELSE 0 END) > 7 )
  )

SELECT s.user_id,
       CASE WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 16 AND 30 THEN
'19-30'
                                                                    WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate)
BETWEEN 31 AND 45 THEN '31-45'
            WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 46 AND 60 THEN
'46-60'
            WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 61 AND 75 THEN
'61-75'
            WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 76 AND 90 THEN
'76-90'
            ELSE 'above 90'
       END AS age_range,
       gender,
       married,
       has_children,
       home_country,
       home_city,
                                                  COUNT(*) AS num_hotel_bookings, ROUND(COALESCE(AVG(hotel_discount_amount *
hotel_per_room_usd),0),2) AS avg_$_saved_per_night
FROM hotels h
JOIN sessions_filtered s ON h.trip_id = s.trip_id
JOIN users u ON u.user_id = s.user_id
GROUP BY 1,2,3,4,5,6,7
ORDER BY 3 DESC


--Bargain Hunter Index and average bags per flight.
-- haversine function to calculate distance between two airports given their coordinates.
WITH sessions_filtered AS (
  SELECT * FROM sessions
  WHERE session_start > '2023-01-04'
  AND user_id IN
  (SELECT user_id FROM sessions WHERE cancellation IS FALSE
   GROUP BY 1
   HAVING SUM(CASE WHEN (flight_booked IS TRUE OR hotel_booked IS TRUE) THEN 1 ELSE 0 END) > 7 )
  )
SELECT s.user_id,
       CASE WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 16 AND 30 THEN
'19-30'
                                                                    WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate)
BETWEEN 31 AND 45 THEN '31-45'
            WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 46 AND 60 THEN
'46-60'
            WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 61 AND 75 THEN
'61-75'
            WHEN EXTRACT(YEAR FROM CURRENT_DATE) - EXTRACT(YEAR FROM birthdate) BETWEEN 76 AND 90 THEN
'76-90'
            ELSE 'above 90'
       END AS age_range,
       gender,
       married,
       has_children,
       home_country,
       home_city,
    COUNT(*) AS num_flight_bookings, ROUND(AVG(checked_bags),1) AS avg_bags,
ROUND(AVG(coalesce(flight_discount_amount,0) * base_fare_usd /
haversine_distance(u.home_airport_lat, u.home_airport_lon, f.destination_airport_lat, f.destination_airport_lon) )::numeric,2) AS
avg_$_saved_per_km
FROM flights f
JOIN sessions_filtered s ON f.trip_id = s.trip_id
JOIN users u ON u.user_id = s.user_id
GROUP BY 1,2,3,4,5,6,7
ORDER BY 4 DESC


WITH sessions_filtered AS (
 SELECT * FROM sessions
 WHERE session_start > '2023-01-04'
 AND user_id IN
 (SELECT user_id FROM sessions WHERE cancellation is FALSE
  GROUP BY 1
  HAVING SUM(CASE WHEN (flight_booked is TRUE or hotel_booked is TRUE) THEN 1 ELSE 0 END) > 7 )
 )

SELECT COUNT(users.user_id) AS total_users,
                              CASE WHEN EXTRACT(MONTH FROM departure_time) IN (12,1,2) THEN 'Winter'
    WHEN EXTRACT(MONTH FROM departure_time) IN (3,4,5) THEN 'Spring'
    WHEN EXTRACT(MONTH FROM departure_time) IN (6,7,8) THEN 'Summer'
    ELSE 'Autumn'
  END AS Seasons
FROM flights
JOIN sessions_filtered ON flights.trip_id=sessions_filtered.trip_id
JOIN users ON sessions_filtered.user_id=users.user_id
GROUP BY 2;
```