```
—-- Filter users with more than 7 sessions in 2023 and date after '2023-01-04’—-
WITH session_2023 AS (
  SELECT *
  FROM sessions s
  WHERE s.session_start > '2023-01-04'
),
filtered_users AS (
  SELECT user_id,
        COUNT(*) AS session_count
  FROM session_2023
  GROUP BY user_id
  HAVING COUNT(*) > 7
),
-- Clean data and join all tables—
session_base AS (
  SELECT
    s.session_id, s.user_id, s.trip_id,
    CAST(s.session_start AS DATE) AS session_start_date,
    s.session_start,
    CAST(s.session_end AS DATE) AS session_end_date,
    s.session_end,
    s.flight_discount, s.hotel_discount, s.flight_discount_amount,
s.hotel_discount_amount,
    s.flight_booked, s.hotel_booked, s.page_clicks, 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,
    h.hotel_name,
    CASE WHEN h.nights < 0 THEN 1 ELSE h.nights END AS nights,
    h.rooms,
    CAST(h.check_in_time AS DATE) AS check_in_date,
    CAST(h.check_out_time AS DATE) AS check_out_date,
    h.hotel_per_room_usd AS hotel_night_price_per_room_usd,
    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
  FROM session_2023 s
  LEFT JOIN users AS u ON s.user_id = u.user_id
  LEFT JOIN hotels AS h ON h.trip_id = s.trip_id
  LEFT JOIN flights AS f ON f.trip_id = s.trip_id
  WHERE s.user_id IN (SELECT user_id FROM filtered_users)
),
-- Filter canceled and non-canceled trips—
```

```
cancelled_trips AS (
  SELECT DISTINCT trip_id
  FROM session_base
  WHERE cancellation = TRUE
),

not_cancelled_trips AS (
  SELECT *
  FROM session_base
  WHERE trip_id IS NOT NULL
  AND trip_id NOT IN (SELECT trip_id FROM cancelled_trips)
),
```

—Feature engineerings BROWSING BEHAVIOR—

```
user_session AS (
  SELECT
    user_id,
    COUNT(DISTINCT session_id) AS num_sessions,
    ROUND(SUM(EXTRACT(EPOCH FROM (session_end - session_start))) / 60,
2) AS session_duration_minutes,
    COUNT(DISTINCT session_id) / NULLIF(COUNT(DISTINCT
session_start_date), 0) AS sessions_per_day,
    SUM(page_clicks) AS num_clicks,
    ROUND(SUM(page_clicks) / NULLIF(SUM(EXTRACT(EPOCH FROM
(session_end - session_start))) / 60, 0), 2) AS click_rate,
    COUNT(session_id) AS user_visit_frequency
  FROM session_base
  GROUP BY user_id
),
```

—Feature engineering TRAVEL BEHAVIOR—

```
  SELECT
    user_id,
    COUNT(DISTINCT trip_id) AS num_trips,
    SUM(
      CASE
        WHEN flight_booked = TRUE AND return_flight_booked = TRUE THEN 2
        WHEN flight_booked = TRUE THEN 1
        ELSE 0
      END
    ) AS num_booked_flights,
    ROUND(AVG(nights), 0) AS avg_hotel_stay_nights,
    ROUND(AVG(EXTRACT(DAY FROM departure_time - session_end_date)), 2)
AS time_after_booking,
    COALESCE(
    SUM(hotel_night_price_per_room_usd * nights * rooms * (1 -
COALESCE(hotel_discount_amount, 0))), 0) AS money_spent_hotel,
    ROUND(AVG(haversine_distance(home_airport_lat, home_airport_lon,
destination_airport_lat, destination_airport_lon)::NUMERIC), 2) AS
```

```
avg_km_flown,
    ROUND(AVG(EXTRACT(DAY FROM departure_time - session_end_date)), 2)
AS avg_days_before_flight,
    SUM(base_fare_usd) AS total_spent_flights,
    COUNT(DISTINCT destination) AS unique_destinations,
    SUM(CASE WHEN flight_discount_amount > 0 OR hotel_discount_amount >
0 THEN 1 ELSE 0 END)
    / COUNT(session_id) AS discount_usage_rate
  FROM not_cancelled_trips
  GROUP BY user_id
),
-- Final grouping with separate columns for features—
final_grouping AS (
  SELECT
    us.*,
    ut.*,
    EXTRACT(YEAR FROM AGE(u.birthdate)) AS age,
    u.gender, u.married, u.has_children, u.home_country, u.home_city,
u.home_airport,

    -- Engagement Segmentation
    CASE
        WHEN sessions_per_day >= 1 AND num_clicks >= 50 THEN 'Highly
Engaged'
        WHEN sessions_per_day BETWEEN 0.5 AND 1 THEN 'Moderately
Engaged'
        ELSE 'Low Engagement'
    END AS engagement_segment,

    -- Spending Segmentation
    CASE
        WHEN total_spent_flights + money_spent_hotel >= 5000 THEN 'Premium
Traveler'
        WHEN total_spent_flights + money_spent_hotel BETWEEN 1000 AND
4999 THEN 'Mid Traveler'
        ELSE 'Budget Traveler'
    END AS spending_segment,

    -- Travel Frequency
    CASE
        WHEN num_trips >= 5 THEN 'Frequent Traveler'
        WHEN num_trips BETWEEN 2 AND 4 THEN 'Occasional Traveler'
        ELSE 'Rare Traveler'
    END AS travel_frequency_segment,

    -- Distance-Based Segmentation
    CASE
```

```
        WHEN avg_km_flown >= 5000 THEN 'Long-Distance Traveler'
        WHEN avg_km_flown BETWEEN 1000 AND 4999 THEN 'Mid-Distance
Traveler'
        ELSE 'Short-Distance Traveler'
    END AS travel_distance_segment,

    -- Discount Segmentation
    CASE
        WHEN discount_usage_rate >= 0.5 THEN 'Discount Seeker'
        WHEN discount_usage_rate BETWEEN 0.2 AND 0.5 THEN 'Occasional
Discount User'
        ELSE 'Full-Price Buyer'
    END AS discount_behavior_segment,

    -- Family vs. Business vs. Regular Traveler Grouping
    CASE
        WHEN has_children = TRUE AND num_trips >= 2 THEN 'Family Traveler'
        WHEN num_booked_flights > 5 AND total_spent_flights > 2000 THEN
'Business Traveler'
        ELSE 'General Traveler'
    END AS traveler_type,

    -- Age-Based Segmentation
    CASE
        WHEN EXTRACT(YEAR FROM AGE(u.birthdate)) < 20 THEN 'Young
Traveler'
        WHEN EXTRACT(YEAR FROM AGE(u.birthdate)) BETWEEN 20 AND 40
THEN 'Young Adult Traveler'
        WHEN EXTRACT(YEAR FROM AGE(u.birthdate)) BETWEEN 41 AND 60
THEN 'Mature Traveler'
        ELSE 'Senior Traveler'
    END AS age_group,

    -- Gender-Based Segmentation
    CASE
        WHEN gender = 'M' THEN 'Male Traveler'
        WHEN gender = 'F' THEN 'Female Traveler'
        ELSE 'Other Traveler'
    END AS gender_group,

    -- Marital Status-Based Segmentation
    CASE
        WHEN married = TRUE THEN 'Married Traveler'
        ELSE 'Single Traveler'
    END AS marital_status_group

  FROM user_session us
```

```
  LEFT JOIN users u ON us.user_id = u.user_id
  LEFT JOIN user_trip ut ON ut.user_id = us.user_id
),
```

—and final the perks—

```
perks AS (
  SELECT *,
    -- Perks assignment based on multiple conditions
    CASE
      -- Highly Engaged & Premium Traveler
      WHEN engagement_segment = 'Highly Engaged' AND spending_segment =
'Premium Traveler'
        THEN 'Free Lounge Access & Complimentary Seat Upgrades'

      -- Highly Engaged & Mid Traveler
      WHEN engagement_segment = 'Highly Engaged' AND spending_segment =
'Mid Traveler'
        THEN 'Priority Check-in & One Free Checked Bag'

      -- Moderately Engaged & Premium Traveler
      WHEN engagement_segment = 'Moderately Engaged' AND
spending_segment = 'Premium Traveler'
        THEN 'Exclusive Travel Discounts & Special Offers'

      -- Frequent Traveler & Long Distance
      WHEN travel_frequency_segment = 'Frequent Traveler' AND
travel_distance_segment = 'Long-Distance Traveler'
        THEN 'Complimentary Airport Lounge Access'

      -- Discount Seeker
      WHEN discount_behavior_segment = 'Discount Seeker' THEN 'Early Access
to Sales'

      -- Special perks based on age, children status, and travel behavior
      WHEN num_trips = 0 THEN '30% off first travel'
      WHEN age < 20 AND has_children = TRUE THEN 'Free child ticket'
      WHEN age < 20 AND has_children = FALSE THEN 'Discount at special
events'
      WHEN age > 60 THEN 'Meal voucher'
      WHEN age >= 20 AND age <= 60 AND has_children = TRUE AND
avg_km_flown > 2 THEN 'Free child ticket'
      WHEN age >= 20 AND age <= 60 AND num_booked_flights > 9 THEN 'Free
meal'

      -- Default Perk
      ELSE '10% off next trip'
    END AS perks
  FROM final_grouping
```

```
)

SELECT * FROM perks;
```