Projekt Travel Tide

SQL Query / Beekeeper:

```
-- Elena proposes to only include sessions:
-- After Jan 4 2023 - sessions CHECK
-- From Users with more than 7 Sessions in the selected time frame
-- check in time should be set to 11.00 to calculate the nights correctly
-- Segmented Business, Family, Senior, Single, Couple, Mature Single
-- output showing segment distribution and key metrics
-- add perks to the segments

with sessions_2023 as (
    select *
    from sessions
    where session_start >= '2023-01-05'
),
users_over_seven_sessions as (
    select user_id, count(session_id) as sessions
    from sessions_2023
    group by user_id
    having count(session_id) > 7
),
session_based_table as (
    select
        s.session_id,
        s.user_id,
        s.trip_id,
        u.birthdate,
        u.gender,
        u.married,
        u.has_children,
        u.home_country,
        f.seats,
        f.departure_time,
        f.checked_bags,
        f.base_fare_usd,
        h.nights,
        h.rooms,
        h.hotel_per_room_usd,
        h.check_in_time,
        h.check_out_time,
        f.return_time,
        s.cancellation
    from sessions_2023 s
    inner join users_over_seven_sessions uos on s.user_id = uos.user_id
    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
),
-- check in time should be set to 11.00 to calculate the nights correctly
session_cleaned as (
    select *,
        case when date(check_out_time) < date(check_in_time)
            or date(check_out_time) = date(check_in_time)
            then date(return_time) - date(check_in_time)
            else date(check_out_time) - date(check_in_time)
        end as nights_cleaned,
        max(case when cancellation = true then 1 else 0 end) over (partition by trip_id) as
trip_was_cancelled
    from session_based_table
),
trip_aggregates as (
    select
        user_id, trip_id, birthdate, gender, married, has_children, home_country,
        max(seats) as trip_seats,
        max(nights_cleaned) as trip_nights,
        max(rooms) as trip_rooms,
        max(checked_bags) as trip_bags,
        max(departure_time) as trip_departure,
        max(hotel_per_room_usd) as trip_hotel_price
    from session_cleaned
    where trip_id is not null
    group by user_id, trip_id, birthdate, gender, married, has_children, home_country
),
user_aggregates as (
    select
        user_id, birthdate, gender, married, has_children, home_country,
        count(distinct trip_id) as total_trips,
        avg(trip_nights) as avg_nights,
        avg(trip_seats) as avg_seats,
        avg(trip_rooms) as avg_rooms,
        avg(trip_bags) as avg_bags,
        avg(trip_hotel_price) as avg_hotel_price,

        -- Weekday travel ratio
        case when count(trip_departure) > 0
            then sum(case when extract(dow from trip_departure) between 1 and 5 then 1 else
0 end) * 1.0 / count(trip_departure)
            else 0 end as weekday_ratio,

        -- Summer/holiday travel ratio
        case when count(trip_departure) > 0
            then sum(case when extract(month from trip_departure) in (6,7,8,12) then 1 else 0
end) * 1.0 / count(trip_departure)
            else 0 end as holiday_ratio,
```

```
        -- Age calculation
        extract(year from current_date) - extract(year from birthdate) as current_age

    from trip_aggregates
    group by user_id, birthdate, gender, married, has_children, home_country
),
-- hotel price
hotel_avg as (
    select avg(avg_hotel_price) as overall_avg_hotel_price
    from user_aggregates
    where avg_hotel_price is not null
),
user_segments as (
    select
        ua.*,
        ha.overall_avg_hotel_price,

  -- Business Travelers: Short stays (<3), weekday, single rooms, min bags, seats 1-2
        case
            when avg_nights < 3
                and weekday_ratio >= 0.7
                and avg_rooms <= 1
                and avg_seats between 1 and 2
                and avg_bags <= 1
                and current_age < 65
            then 'Business'

  -- Family Travelers: Have children, multiple seats/rooms, longer stays, summer/holiday
travel
            when has_children = true
                and avg_seats >= 3
                and avg_nights >= 5
                and avg_rooms > 1
                and holiday_ratio >= 0.3
                and current_age < 65
            then 'Family'

  -- Senior Travelers: Age 65+, longer stays, higher hotel prices, avoid peak season
            when current_age >= 65
                and avg_nights >= 7
                and coalesce(avg_hotel_price, 0) > coalesce(ha.overall_avg_hotel_price, 0)
                and holiday_ratio <= 0.4
            then 'Senior'

  -- Single: Under 35, single, with or without children, not married
            when current_age < 35
                and married = false
```

```
            then 'Single'

    -- Couple: Married, no children, age 35-64
            when married = true
                and has_children = false
                and current_age between 35 and 64
            then 'Couple'

    -- Mature Single: 35-64, not married
            when current_age between 35 and 64
                and married = false
            then 'Mature Single'

    -- All 65+ who don't meet Senior criteria become Senior anyway
            when current_age >= 65
            then 'Senior'

     -- Remaining under 65 cases
            when current_age < 65
            then case
                when has_children = true then 'Family'
                when married = true then 'Couple'
                else 'Single'
            end
        end as user_segment

      from user_aggregates ua
      cross join hotel_avg ha
      where total_trips >= 2
)
-- output: the segmented table
/*
select
    user_id,
    user_segment,
    current_age,
    gender,
    married,
    has_children,
    home_country,
    total_trips,
    round(avg_nights, 1) as avg_nights_per_trip,
    round(avg_seats, 1) as avg_seats_per_trip,
    round(avg_rooms, 1) as avg_rooms_per_trip,
    round(avg_bags, 1) as avg_bags_per_trip,
    round(avg_hotel_price, 0) as avg_hotel_price,
    round(weekday_ratio, 2) as weekday_travel_ratio,
    round(holiday_ratio, 2) as holiday_travel_ratio
```

```
from user_segments
order by user_segment, total_trips desc;
*/

-- output: showing segment distribution and key metrics
/*
select
    user_segment,
    count(*) as user_count,
    round(count(*) * 100.0 / sum(count(*)) over (), 2) as percentage_of_total,
    round(avg(current_age), 1) as avg_age,
    round(avg(total_trips), 1) as avg_trips,
    round(avg(avg_nights), 1) as avg_nights_per_trip,
    round(avg(avg_seats), 1) as avg_seats,
    round(avg(avg_hotel_price), 0) as avg_hotel_price,
    round(avg(weekday_ratio), 2) as avg_weekday_ratio,
    round(avg(holiday_ratio), 2) as avg_holiday_ratio,
    round(avg(case when gender = 'M' then 1.0 else 0.0 end) * 100, 1) as pct_male,
    round(avg(case when gender = 'F' then 1.0 else 0.0 end) * 100, 1) as pct_female,
    round(avg(case when married = true then 1.0 else 0.0 end) * 100, 1) as pct_married,
    round(avg(case when has_children = true then 1.0 else 0.0 end) * 100, 1) as
pct_with_children
from user_segments
group by user_segment
order by percentage_of_total desc;
*/



-- MAIN QUESTION: All users with their segments and perks

select
    user_id,
    user_segment,
    case
        when user_segment = 'Business' then 'Priority Check-in & Lounge Access'
        when user_segment = 'Family' then 'Kids Fly Free & Family Entertainment Package'
        when user_segment = 'Senior' then 'Senior Discount & Flexible Booking'
        when user_segment = 'Single' then 'Solo Traveler Bonus & City Tour Credits'
        when user_segment = 'Couple' then 'Romantic Upgrade & Couples Spa Package'
        when user_segment = 'Mature Single' then 'Premium Experience & Cultural Tours'
        else 'Standard Benefits'
    end as segment_perk,
    current_age,
    case when gender = 'M' then 'Male'
         when gender = 'F' then 'Female'
         else 'Unknown' end as gender,
```

```
    case when married = true then 'Married' else 'Single' end as marital_status,
    case when has_children = true then 'With Children' else 'No Children' end as
children_status,
    home_country,
    total_trips,
    round(avg_nights, 1) as avg_nights_per_trip,
    round(avg_seats, 1) as avg_seats_per_trip,
    round(avg_rooms, 1) as avg_rooms_per_trip,
    round(avg_bags, 1) as avg_bags_per_trip,
    round(avg_hotel_price, 0) as avg_hotel_price_usd,
    round(weekday_ratio * 100, 1) as weekday_travel_percent,
    round(holiday_ratio * 100, 1) as holiday_travel_percent

from user_segments
order by user_segment, total_trips desc, user_id;
```