

TravelTide - SQL Code for Customer Segmentation

-- Zuerst alle Daten nach dem 2023-01-04 selektieren

WITH sessions_2023 AS (

SELECT *

FROM sessions

WHERE session_start >= '2023-01-05' -- Selektiere alle Spalten und Zeilen aus Sessions für nach dem 04.01.2023

),

-- Selektieren nach Usern, die mehr als 7 Sessions haben

over_seven_sessions AS (

SELECT

user_id,

COUNT (*) AS session_count

FROM sessions_2023

GROUP BY user_id

HAVING COUNT (*) > 7 -- Selektiere alle User, die mehr als 7 Sessions hatten

),

/* Verknüpfe alle Spalten von allen Tabellen flights, hotels, users mit sessions, um nur eine Tabelle

zu erhalten */

prep_sessions_based AS (

SELECT

s.session_id,

s.user_id,

s.trip_id,

s.session_start,

s.session_end,

s.page_clicks,

s.flight_discount::integer AS flight_discount, -- Für Berechnungen als Ganzzahl
umgewandelt

s.flight_discount_amount,

s.hotel_discount::integer AS hotel_discount, -- Für Berechnungen als Ganzzahl
umgewandelt

s.hotel_discount_amount,

s.flight_booked::integer AS flight_booked, -- Für Berechnungen als Ganzzahl
umgewandelt

```

        s.hotel_booked::integer AS hotel_booked, -- Für Berechnungen als Ganzzahl
umgewandelt
        s.cancellation:: integer AS cancellation, -- Für Berechnungen als Ganzzahl
umgewandelt
        u.birthdate,
        u.gender,
        u.married::integer AS married, -- Für Berechnungen als Ganzzahl umgewandelt
        u.has_children::integer AS has_children, -- Für Berechnungen als Ganzzahl
umgewandelt
        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.check_out_time,
        h.hotel_per_room_usd,

```

/* check-in-time sollte auf 11:00 Uhr vom Vortag gesetzt werden, um die Nächte korrekt zu kalkulieren.

check_in_time kann nicht nach check-out-time sein wie in manchen Zeilen vorhanden. Führt zur

Verzerrung und auch zu negativen Nächten. Für leichtere Berechnung Timestamp in Date umwandeln */

```

CASE WHEN date (h.check_out_time) < date (check_in_time)
      OR date (h.check_out_time) = date(check_in_time) THEN date (f.return_time)
      - date (h.check_in_time)
ELSE date(check_out_time) - date (check_in_time) END AS cleaned_nights,

```

/* Korrektur der Cancellations. Zu jeder Trip-ID, die storniert wurde, gehört in dem Datensatz auch

eine Trip-ID, die nicht storniert wurde - also Buchung des Trips und dann mit einer anderen Session die Stornierung. Wir wollen dann nur die Stornierung sehen und nicht mehr als gebucht. Max (0, 1) nimmt von beiden Werten nur den höheren Wert. Daher wurde cancellation in Integer umgewandelt */

```
MAX (CASE WHEN s.cancellation = 'true' THEN 1 ELSE 0 END) OVER (PARTITION BY  
s.trip_id) AS final_trip_cancellation_status
```

```
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 over_seven_sessions)
```

```
),
```

```
-- 47436 Zeilen
```

/* Vereinfachung für anschließende Metriken auf User-Ebene über Bedingungen. So erhalte mit der entsprechenden Bedingung für das, was ich berechnen möchte nur den Alias und muss die Bedingung nicht in die Funktion mehr integrieren und welche Metriken für meine Features des Segments eventuell wichtig sind. */

```
sessions_based_cleaned AS (
```

```
SELECT
```

```
*,
```

```
    CASE WHEN final_trip_cancellation_status = 0 THEN trip_id END AS  
    not_cancelled_trip, -- Nicht stornierte Buchungen
```

```
    CASE WHEN final_trip_cancellation_status = 1 THEN trip_id END AS cancelled_trip,  
-- Stornierte Buchungen
```

```
    CASE WHEN trip_id IS NOT NULL AND final_trip_cancellation_status = 0 THEN  
    EXTRACT (DAY FROM (departure_time - session_start))  
    END AS days_to_trip, -- Wieviele Tage wurde vor Antritt des Trips gebucht. Trip-Id  
    muss vorhanden und nicht storniert sein sein. Aus dem Timestamp wird der Tag  
    herausgefiltert und die Differenz zwischen Session-Start und Abflugtag berechnet.
```

```
    EXTRACT (EPOCH FROM (session_end - session_start)) AS seconds_per_session,  
-- Berechnung der Dauer pro Session in Sekunden
```

CASE WHEN final_trip_cancellation_status = 0 THEN page_clicks else 0 END AS
page_clicks_not_cancelled, -- Page_clicks für die nicht Sessions, wo der Trip nicht
storniert wurde

CASE WHEN final_trip_cancellation_status = 0 AND (flight_booked = 1 AND
hotel_booked = 1) THEN 1 ELSE 0 END -- Hotel und Flug gebucht wurde
AS flight_hotel_booked,

CASE WHEN hotel_discount = 1 OR flight_discount = 1 THEN trip_id END AS
booking_with_discount, -- Hotel- oder ein Flugdiscount für nicht stornierte
Buchungen gegeben wurde

CASE WHEN final_trip_cancellation_status = 0 THEN EXTRACT (EPOCH FROM
(session_end - session_start)) END AS seconds_per_non_cancelled_session,
-- Berechnung der Dauer der Session in Sekunden für nicht stornierte Buchungen

EXTRACT (YEAR FROM AGE ('2023-06-01', birthdate)) AS age,
-- Umrechnung in Jahre als Alter

CASE WHEN final_trip_cancellation_status = 0 THEN haversine_distance
(destination_airport_lat, home_airport_lat, destination_airport_lon,
home_airport_lon) END AS distance_km,
-- Berechnung der Distanz vom Abflugsflughafen zum Ankunftsflughafen in km

CASE WHEN final_trip_cancellation_status = 0 THEN hotel_per_room_usd::dec *
rooms * cleaned_nights *
COALESCE ((1 - hotel_discount_amount), 1) END hotel_costs,
-- Berechnung der Hotelkosten pro Nacht und Zimmer für nicht stornierte Buchungen abzgl.
des Discounts unter Berücksichtigung, dass der erste Wert, der nicht NULL ist
zurückgegeben wird.

CASE WHEN final_trip_cancellation_status = 0 THEN base_fare_usd::dec *
COALESCE ((1 - flight_discount_amount), 1)
END AS flight_costs,
-- Berechnung der Flugkosten für nicht stornierte Buchungen abzgl. des Discounts unter
Berücksichtigung, dass der erste Wert, der nicht NULL ist zurückgegeben wird.

CASE WHEN final_trip_cancellation_status = 0 THEN session_id END AS
sessions_not_cancelled,
-- Nur Session_ID von Buchungen, die nicht storniert wurden

```

CASE WHEN final_trip_cancellation_status = 0 AND EXTRACT (MONTH FROM
departure_time) IN (3, 4, 7, 8, 12) THEN 1
  WHEN final_trip_cancellation_status = 0 AND EXTRACT (MONTH FROM
departure_time) IN (1,2, 5, 6, 9, 10, 11)
  THEN 0 ELSE NULL END AS vacation_time

```

-- Anzeige von Zeiten, die typischerweise Urlaubs-/Ferienzeiten sind oder nicht unter Berücksichtigung nur von den Monaten

FROM prep_sessions_based

),

/* Auf User-Ebene werden jetzt die Metriken, die später für die Feature-Berechnungen benötigt werden, berechnet und um eine Vorab-Ansicht über die Daten zu erhalten. So können die neu kreierten Spalten aus der vorigen CTE einfach benutzt werden. Nur Werte für eine Zeile pro User und Berechnungen auf User-Ebene. */

user_based_metric AS (

```

SELECT
  user_id,

```

-- Trip basierte Metriken

```

COUNT (DISTINCT trip_id) AS total_trips, -- Summe der Trips/Buchungen
COUNT (DISTINCT not_cancelled_trip) AS num_not_cancelled_trips, -- Anzahl der nicht
stornierten Buchungen
COUNT (DISTINCT cancelled_trip) AS num_cancelled_trips, -- Anzahl der stornierten
Buchungen
COUNT (DISTINCT booking_with_discount) AS total_booking_with_discount, -- Anzahl der
Hotel- oder Flugbuchungen mit Discount
SUM (distance_km) AS total_distance_km, -- Summe der km vom Abflug- und
Ankunftsflughafen
SUM (seats) AS total_seats, -- Summe der Sitze
SUM (checked_bags) AS total_checked_bags, -- Summe der eingetragenen Gepäckstücke
SUM (vacation_time) AS num_vacation_time, -- Summe der Buchungen in der Urlaubszeit
SUM (flight_costs) AS total_flight_costs, -- Summe der Flugkosten
SUM (flight_discount_amount) AS total_flight_discount_amount, -- Summe der
Flugdiscounts
SUM (hotel_costs) AS total_hotel_costs, -- Summe der Hotelkosten
SUM (hotel_discount_amount) AS total_hotel_discount_amount, -- Summe der
Hoteldiscounts
SUM (cleaned_nights) AS total_cleaned_nights, -- Summe der bereinigten
Hotelübernachtungen
SUM (flight_hotel_booked) AS total_flight_hotel_booked, -- Summe der gemeinsamen
Hotel- und Flugbuchungen

```

-- Sessions basierte Metriken

```
SUM (page_clicks) AS total_page_clicks,  
COUNT (session_id) AS total_sessions,  
    COUNT (sessions_not_cancelled) AS total_sessions_not_cancelled,  
SUM (seconds_per_non_cancelled_session) AS total_sec_sessions,  
SUM (days_to_trip) AS total_days_to_trip,
```

-- Berechnungen von married, has_children und age, um diese in der Auflistung in GROUP BY zu vermeiden

```
MAX (CASE WHEN married = 1 THEN 1 ELSE 0 END) AS married,  
MAX (CASE WHEN has_children = 1 THEN 1 ELSE 0 END) AS has_children,  
MAX (age) AS age
```

```
FROM sessions_based_cleaned  
GROUP BY user_id
```

),

-- 5782 Zeilen

/*

-- Feature Engineering --

*/

user_based_features AS (

```
SELECT  
    user_id,
```

-- Trip features

```
total_trips, -- Summe der Trips/Buchungen bleibt  
num_not_cancelled_trips, -- bleibt bestehen  
num_cancelled_trips, -- bleibt zum Vergleichen bestehen  
num_cancelled_trips::dec / NULLIF (total_trips, 0) AS cancellation_quote, -- Berechnung  
des Durchschnittswerts von Stornierungen aller Buchungen pro User. NULLIF, da es  
0-Werte gibt und nicht durch 0 teilen darf  
total_booking_with_discount, -- bleibt bestehen zum Vergleich bestehen  
total_booking_with_discount::dec / NULLIF (total_trips, 0) AS discount_quote,  
--Berechnung des Durchschnittswerts der Buchungen mit Discount für alle Buchungen pro  
User
```

total_distance_km / NULLIF (total_trips, 0) AS avg_distance_km, -- Durchschnittsdistanz in km für alle Buchungen pro User

total_seats, -- bleibt bestehen zum Vergleich, da es bei der Durchschnittsberechnung zu 0-Werten führt, wenn z.B. eine Buchung mit Flug 1 Sitz hat und die zweite Buchung nur Hotel gebucht wurde. Es wird nicht 0.5 angezeigt, sondern 0

total_seats / NULLIF (total_trips, 0) AS avg_seats, -- Anzahl der Sitze im Durchschnitt für alle Buchungen pro User

total_checked_bags / NULLIF (total_trips, 0) AS avg_checked_bags, -- Anzahl der durchschnittlichen Gepäckstücke für alle Buchungen pro User

total_hotel_costs / NULLIF (total_trips, 0) AS avg_hotel_costs, -- Durchschnittliche Hotelkosten für alle Buchungen pro User

total_hotel_discount_amount / NULLIF (total_trips, 0) AS avg_hotel_discount, -- Durchschnittlicher Hotel-Discountbetrag für alle Buchungen pro User

total_cleaned_nights / NULLIF (total_trips, 0) AS avg_cleaned_nights, -- Durchschnittliche bereinigte Hotelübernachtungen für alle Buchungen pro User

num_vacation_time / NULLIF (total_trips, 0) AS avg_vacation_time, -- Durchschnittliche Urlaubszeit für alle Buchungen

total_flight_costs / NULLIF (total_trips, 0) AS avg_flight_costs, -- Durchschnittlich Flugkosten für alle Buchungen

total_flight_discount_amount / NULLIF (total_trips, 0) AS avg_flight_discount, -- Durchschnittlicher Flug-Discountbetrag für alle Buchungen pro User

total_flight_hotel_booked::dec / NULLIF (total_trips, 0)::dec AS avg_flight_hotel_booked, -- Durchschnittliche gebuchte Flüge und Hotels gemeinsam für alle Buchungen pro User

-- Session based features

total_sessions, -- Summe der Sessions pro User

total_sessions_not_cancelled / NULLIF (total_sessions, 0) AS

avg_sessions_not_cancelled, -- Durchschnittliche nicht stornierter Sessions von allen Sessions pro User

total_page_clicks / NULLIF (total_sessions, 0) AS avg_page_clicks, -- Durchschnittliche Clicks per Session pro User

total_trips / NULLIF (total_sessions_not_cancelled, 0)::dec AS booking_rate, -- Durchschnittliche Buchungen für alle nicht stornierten Buchungen per Session pro User

total_sec_sessions / total_sessions AS avg_sec_per_session, -- Durchschnittliche Dauer pro Session in Sekunden für alle Sessions pro User

total_days_to_trip / NULLIF (total_sessions_not_cancelled, 0) AS avg_days_to_trip, -- Durchschnittliche Anzahl der Tage von den nicht stornierten Buchungen der Sessions bis zum Abflugsag

-- Personal features

married, -- Status: verheiratet übernommen

has_children, -- Status: hat Kinder übernommen

age, -- Das Alter des Users übernommen

CASE WHEN age >=16 AND age <= 21 THEN 1 ELSE 0 END AS young_customers, -- Kategorisiert in eine Altersklasse Junge Kunden

```
CASE WHEN age > 21 AND age < 55 THEN 1 ELSE 0 END AS middle_age_customers,  
-- Kategorisiert in eine Altersklasse Mittleres Alter des Kunden
```

```
CASE WHEN age >= 55 THEN 1 ELSE 0 END AS senior_customers  
-- Kategorisiert in die Altersklasse Senior des Kunden
```

```
FROM user_based_metric
```

```
),  
/*
```

```
-- Segmentation --
```

Um Bias Annahmen zu vermeiden und einer ausgesuchten Kennzahl anzugeben, werden
Perzentile fortlaufend berechnet.

25 Quartile, Werte unter dem Median und 75 Quartile, Werte oberhalb des Median und
Outliers unterhalb bzw. oberhalb des Median-Werts einer

Metrik mit einzubeziehen. Dafür muss eine eigene CTE vor der Anwendung als Merkmal zur
Bestimmung der Segmente für die Berechnung der Perzentile eingefügt werden. */

```
segmented_thresholds AS (
```

```
SELECT
```

```
    PERCENTILE_CONT (0.5) WITHIN GROUP (ORDER BY avg_page_clicks) AS  
p50_avg_page_clicks, -- Median-Wert von dem Durchschnittlichen Wert der Clicks per  
Session
```

```
    PERCENTILE_CONT (0.75) WITHIN GROUP (ORDER BY avg_hotel_costs) AS  
p75_avg_hotel_costs_cont, -- 3 Quartil der durchschnittlichen Hotelkosten
```

```
    PERCENTILE_CONT (0.25) WITHIN GROUP (ORDER BY avg_hotel_costs) AS  
p25_avg_hotel_costs_cont, -- 1 Quartil der der durchschnittlichen Hotelkosten
```

```
    PERCENTILE_CONT (0.75) WITHIN GROUP (ORDER BY avg_flight_costs) AS  
p75_avg_flight_costs_cont, -- 3 Quartil der durchschnittlichen Flugkosten
```

```
    PERCENTILE_CONT (0.25) WITHIN GROUP (ORDER BY avg_flight_costs) AS  
p25_avg_flight_costs_cont -- 1 Quartil der der durchschnittlichen Flugkosten
```

```
FROM user_based_features
```

```
WHERE avg_hotel_costs IS NOT NULL AND avg_flight_costs IS NOT NULL
```

```
-- Filter: Nur für vorhandene Hotel- und Flug-Kosten und Page-Clicks berechnen
```

```
AND avg_page_clicks IS NOT NULL
```

```
),
```

```
users_segmented AS (
```


SELECT

```
*,
p50_avg_page_clicks,
p75_avg_hotel_costs_cont,
p25_avg_hotel_costs_cont,
p75_avg_flight_costs_cont,
p25_avg_flight_costs_cont,
```

/* Alle User zusammen 5782 User */

- Kundensegmentierung in Window-Shopper, Hesitant, Bargain-Shopper, Family (Gruppenreisen), Package-Tour, Geschäftsreisende,
- Long-Distance-Customer, Luxury, Last-minute-Booker und Andere (Rest, die nicht zu einem der genannten Segmente zuzuordnen ist)
- Überwiegend haben fast alle Kunden Langstrecken-Flüge mit mehr als 8000 km, überwiegend über 10000 km durchschnittlich, bis auf Last-minute-Booker und Luxury, die überwiegend nur Hotel gebucht haben.
- Daher wurde an anderen Merkmalen gruppiert, ob Single reist oder mehr eingetragene Gepäckstücke gab oder überdurchschnittlich bzw. unterdurchschnittlich Hotelkosten oder Flugkosten bezahlt wurde oder
- die Abreise nach Buchung sehr kurzfristig erfolgt war.

- 1. Window-Shopper: viele Clicks, keine Buchungen

- 2. Hesitant: generell keine Buchungen

- 3. Bargain-Shopper: Buchungen nicht 0, Discount-Quote nicht 0 und größer als 0.75 oder
-- Hotel oder Flug kleiner bis gleich 1 Quartil von Hotel- oder Flugkosten und die Buchung größer gleich 2 Tage vor Abflug erfolgt ist.
-- (größer gleich 2 Tage vor Abflug könnte auch als Last-minute-Booker zugeordnet werden). Dennoch wird eine Abgrenzung benötigt.)

- 4. Familie/Gruppenreise: Seats >=3, has_children TRUE, Age 21 - 55, Urlaubszeiten = Ferienzeiten,
-- Nächte >=7, eventuell Buchung > 21 Tage vor Abflug, mindestens 2 Gepäckstücke

- 5. Geschäftsreisende: Sitze 1 oder weniger (Der User hat eventuell einen zweiten Trip mit nur Hotel
-- gebucht und keinen Flug), Maximal 1 Gepäckstück, Discount-Quote nicht Null und kleiner als 0.25 und
-- durchschnittlich größer gleich 3 Tage Buchung vor Abflug. Alter 25-63
-- Auch hier gibt es Überschneidungen mit Last-minute-Booker, da einige User sehr kurzfristig buchen und ein paar Differenzen vom Alter her gibt und jünger oder älter sind.

- 6. Langstrecken-Reisende (Heimreisende): Weiter als 8000 km, maximal durchschnittlich 2 Sitze, durchschnittlich maximal 2 Gepäckstücke

- 7. Luxus: Flug teuer größer gleich 3. Quartil oder Hotel größer gleich 3. Quartil
- 8. Last-minute-Booker: Kunden, die kurzfristig buchen (maximal 2 Tage vor Reiseantritt), sehr
 -- oft nur ein Hotel und kein Flug gebucht
- 9. Alle anderen, die nicht zu den oben genannten Kundensegmenten passen.

/* CASE

WHEN booking_rate = 0 AND avg_page_clicks >= p50_avg_page_clicks THEN
 'Window-Shopper' -- 42 User, kann mit Zögerer zusammengefasst werden */

WHEN booking_rate = 0 THEN 'Hesitant' -- 404 User + 42 Window-Shopper = 446 User

WHEN booking_rate > 0 AND ((discount_quote IS NOT NULL AND discount_quote > 0.75)
 OR (avg_hotel_costs <= p25_avg_hotel_costs_cont
 OR avg_flight_costs <= p25_avg_flight_costs_cont)) AND avg_days_to_trip >= 2 THEN
 'Bargain-Shopper'

- 1060 User, in Verhältnis zu den anderen Segmenten die meisten Stornierungen
- und die meisten User haben mehr als 1 Trip zwischen 2-7 Trips.

WHEN avg_seats = 1 AND avg_checked_bags <=1 AND (discount_quote IS NOT NULL
 AND discount_quote < 0.25)
 OR (avg_hotel_costs > p25_avg_hotel_costs_cont AND avg_flight_costs >
 p25_avg_flight_costs_cont)
 AND avg_days_to_trip >= 3 AND age BETWEEN 25 AND 63 THEN 'Business' -- 1897
 User, sehr wenig Stornierungen und die meisten haben zwischen 2-8 Trips. Darunter sind
 auch Last-minute-Booker, weniger als 3 Tage vor Abflug buchen.

-- WHEN /*avg_distance_km > 4500 AND avg_cleaned_nights >= 7 AND
 avg_days_to_trip >=7 AND (avg_flight_hotel_booked > 0 AND
 -- avg_hotel_costs > p25_avg_hotel_costs_cont AND avg_flight_costs >
 p25_avg_flight_costs_cont) THEN 'Package-Tour' -- Nur 4 User,
 -- kann mit Long-Distance-Customer oder einem anderen Segment zusammengefasst
 werden

WHEN avg_distance_km > 8000 AND avg_seats <= 2 AND avg_checked_bags <=2
 THEN 'Long-Distance-Customer'
 -- 1730 User, wenig Stornierungen und die meisteisten 2-4 Trips

-- WHEN avg_seats >=3 AND has_children = 1 AND age BETWEEN 21 AND 64 --
 AND avg_vacation_time = 1
 -- AND avg_cleaned_nights >= 3 AND avg_checked_bags >=2 THEN 'Family' -- 15 User,
 die alle einen Trip storniert haben !!!
 -- Daher auf andere Segmente aufgeteilt.

WHEN avg_hotel_costs >= p75_avg_hotel_costs_cont OR avg_flight_costs >=

```
p75_avg_flight_costs_cont THEN 'Luxury'
```

```
-- 321 User, wenig Stornierungen, aber mehr User mit nur 1 Trip und ein paar User mit 4-5 Trips. Überwiegend wurde nur Hotel gebucht und Distanz unter 8000 km.
```

```
WHEN (avg_days_to_trip IS NULL OR avg_days_to_trip <=2) OR (avg_flight_costs IS NOT NULL OR avg_hotel_costs IS NOT NULL) THEN 'Last-minute-Booker'
```

```
-- 328 User, die meisten haben nur ein Hotel gebucht und es gibt mehr Stornierungen. Die meisten haben zwischen 1-3 Trips gebucht, ein paar 4-5 Trips, siehe Luxury-Segment  
ELSE 'Other'
```

```
END AS reward_segment
```

```
FROM user_based_features
```

```
CROSS JOIN segmented_thresholds -- CROSS JOIN, da segmented_thresholds nur eine Zeile hat und die eigene CTE für die Perzentile mit der Tabelle users_segmented zu verbinden
```

```
),
```

```
/*
```

```
-----
```

```
-- Segmented Users with Perks --
```

```
-----
```

```
-- Es werden die jeweiligen 5 Perks den einzelnen Kundensegmenten zugeordnet, die wohl für das Kundensegment nach Visualisierung über Tableau Public Software am meisten zutreffend ist. */
```

```
segmented_users_with_perks AS (
```

```
SELECT
```

```
    *,
```

```
    CASE
```

```
        WHEN reward_segment = 'Luxury' THEN 'free hotel meal' -- Das Segment hat überwiegend Hotelbuchungen mit überdurchschnittlichen Hotelkosten, einige hohe Anzahl an Hotelübernachtungen. Das Segment ist nicht so groß. Einige haben auch einen hohen Discount.
```

```
        WHEN reward_segment = 'Long-Distance-Customer' THEN 'free checked bag' -- Das Segment hat durchschnittlich am meisten am meisten Checked Bags
```

```
        WHEN reward_segment = 'Bargain-Shopper' THEN 'No cancellation fees' -- Das Segment hat die höchste Anzahl an Stornierungen. Die User haben viele eine hohe Discount-Quote und haben nicht ganz so kurzfristig gebucht sowie sehr günstige Flüge/Hotels. Daher könnten die Cancellation-Fees am interessantesten sein, aber exklusiver Discount oder kostenlose Hotelmahlzeit auch.
```

WHEN reward_segment = 'Last-minute-Booker' THEN 'exclusive discounts' --
Das Segment hat hohe Discounts aber eine kostenlose Cancellation-Fee könnte aufgrund der Kurzfristigkeit nicht so interessant sein.

WHEN reward_segment = 'Business' THEN '1 night free hotel with flight' --
Das Segment hat die meisten Trips inklusive Flug pro User und viele auch eine Hotelübernachtung und nur eine Reisender. Daher könnte eine kostenlose Hotelübernachtung, wenn man einen Flug gebucht hat, am interessantesten sein.

WHEN reward_segment = 'Hesitant' THEN 'exclusive Discounts' -- Das Segment könnte exclusive Discounts eventuell ansprechen, um doch eine Buchung auszuführen

ELSE 'No specific perk'
END AS assigned_perk
FROM users_segmented
)

-- Alle neu berechneten Spalten aufführen, um eine neue endgültige Tabelle für einen einzelnen User mit dem jeweiligen Kundensegment und dem zugeteilten Perk zu erhalten. */

SELECT
user_id,
reward_segment,
assigned_perk,
--total_trips,
--num_not_cancelled_trips,
--num_cancelled_trips,
--cancellation_quote,
--total_booking_with_discount,
discount_quote,
avg_distance_km,
--total_seats,
--avg_seats,
--avg_checked_bags,
avg_hotel_costs,
--avg_hotel_discount,
avg_cleaned_nights,
--avg_vacation_time,
avg_flight_costs,
--avg_flight_discount,
--total_sessions,
--avg_page_clicks,
--booking_rate,
--avg_sec_per_session,
avg_days_to_trip
--married,
--has_children,
--age,
--young_customers,
--middle_age_customers,

--senior_customers

FROM segmented_users_with_perks

ORDER BY reward_segment, user_id -- Sortiert nach Kundensegment und User_ID

-- 5782 Zeilen und damit eindeutige Nutzer, darauf verteilt 1897 Business-Kunden,