```
WITH params AS (
  SELECT
    1::int  AS min_sessions,
    (CURRENT_DATE - INTERVAL '6 months') AS signup_cutoff,
    500::int AS clicks_cap,
    100::numeric AS pct_cap
),

-- 1) Sessions je User (für Kohortenauswahl)
session_rollup AS (
  SELECT
    s.user_id,
    COUNT(*)             AS total_sessions,
    MIN(s.session_start) AS first_session,
    MAX(s.session_end)   AS last_session
  FROM sessions s
  GROUP BY s.user_id
),

-- 2) Kohorte: aktive Nutzer vor Stichtag mit min. X Sessions
cohort_users AS (
  SELECT
    u.user_id,
    u.gender,
    u.married,
    u.has_children,
    u.home_country,
    u.home_city,
    u.home_airport,
    u.sign_up_date,
    (DATE_PART('year', AGE(CURRENT_DATE, u.sign_up_date)) * 12
     + DATE_PART('month', AGE(CURRENT_DATE, u.sign_up_date)))::int AS
customer_age_months,
    r.total_sessions,
    r.first_session,
    r.last_session
  FROM users u
  JOIN session_rollup r ON r.user_id = u.user_id
  CROSS JOIN params p
  WHERE u.sign_up_date <= p.signup_cutoff
    AND r.total_sessions >= p.min_sessions
),

-- 3) Sessions bereinigen (nur Kohorte, Dauer, Caps)
tt_sessions_clean AS (
  SELECT
    s.*,
    EXTRACT(EPOCH FROM (s.session_end - s.session_start))/60.0 AS session_minutes,
    LEAST(s.page_clicks, p.clicks_cap) AS page_clicks_capped,
    GREATEST(0, LEAST(p.pct_cap, COALESCE(s.flight_discount_amount, 0))) AS
flight_discount_pct,
    GREATEST(0, LEAST(p.pct_cap, COALESCE(s.hotel_discount_amount, 0)))  AS
hotel_discount_pct
  FROM sessions s
  JOIN cohort_users cu ON cu.user_id = s.user_id
  CROSS JOIN params p
  WHERE s.page_clicks >= 2
    AND s.session_end >= s.session_start
),

-- 4) Hotels bereinigen (nur valide Nächte/Rooms/Preis)
hotel_clean AS (
  SELECT
    h.trip_id,
    h.hotel_name,
    h.nights,
    h.rooms,
    h.check_in_time,
    h.check_out_time,
    h.hotel_per_room_usd,
    (h.nights * h.rooms * h.hotel_per_room_usd) AS hotel_total_usd
  FROM hotels h
  WHERE h.nights > 0 AND h.rooms > 0 AND h.hotel_per_room_usd > 0
),

-- 5) Flüge bereinigen (nur valide Preise)
flight_clean AS (
  SELECT
    f.trip_id,
    f.checked_bags,
    f.trip_airline,
    f.base_fare_usd
  FROM flights f
  WHERE f.base_fare_usd > 0
),

-- 6) Buchungs-/Umsatz-Aggregate pro User
book_rollup AS (
  SELECT
    s.user_id,
    COUNT(DISTINCT CASE WHEN fc.trip_id IS NOT NULL THEN fc.trip_id END) AS
flights_cnt,
    COUNT(DISTINCT CASE WHEN hc.trip_id IS NOT NULL THEN hc.trip_id END) AS
hotels_cnt,
    SUM(fc.base_fare_usd)                                    AS flight_revenue_usd,
    SUM(hc.hotel_total_usd)                                  AS hotel_revenue_usd,
    AVG(NULLIF(fc.checked_bags, 0))                         AS avg_checked_bags,
    AVG(hc.nights)                                          AS avg_nights
  FROM tt_sessions_clean s
  LEFT JOIN flight_clean fc ON fc.trip_id = s.trip_id
  LEFT JOIN hotel_clean  hc ON hc.trip_id = s.trip_id
  GROUP BY s.user_id
),

-- 7) Session-Features pro User
session_features AS (
  SELECT
    s.user_id,
    COUNT(*)                              AS sessions_cnt,
    COUNT(DISTINCT DATE(s.session_start)) AS active_days,
    AVG(s.page_clicks_capped)             AS avg_clicks,
    SUM(s.page_clicks_capped)             AS total_clicks,
    AVG(s.session_minutes)                AS avg_session_min,
    SUM((s.flight_booked::int))           AS flight_bookings,
    SUM((s.hotel_booked::int))            AS hotel_bookings,
    AVG((s.cancellation::int))            AS cancellation_rate,
    AVG((s.flight_discount::int))         AS p_seen_flight_discount,
    AVG((s.hotel_discount::int))          AS p_seen_hotel_discount
  FROM tt_sessions_clean s
  GROUP BY s.user_id
)

-- 8) Finale User-Tabelle
SELECT
  cu.user_id,
  cu.gender,
  cu.married,
  cu.has_children,
  cu.home_country,
  cu.home_city,
  cu.home_airport,
  cu.sign_up_date,
  cu.customer_age_months,
  cu.total_sessions             AS total_sessions_lifetime,
  cu.first_session,
  cu.last_session,

  sf.sessions_cnt,
  sf.active_days,
  sf.avg_clicks,
  sf.total_clicks,
  sf.avg_session_min,
  sf.flight_bookings,
  sf.hotel_bookings,
  sf.cancellation_rate,
  sf.p_seen_flight_discount,
  sf.p_seen_hotel_discount,

  br.flights_cnt,
  br.hotels_cnt,
  br.flight_revenue_usd,
  br.hotel_revenue_usd,
  br.avg_checked_bags,
  br.avg_nights

FROM cohort_users cu
LEFT JOIN session_features sf ON sf.user_id = cu.user_id
LEFT JOIN book_rollup     br ON br.user_id = cu.user_id;
```