

— Maxwell Acha's Metrocar SQL Queries

--1: Customer Funnel Analysis

```
WITH user_funnel AS(
--app downloads
SELECT
    1 AS funnel_step,
    'app downloads' AS step_name,
    COUNT(*) AS user_count
FROM app_downloads
--signups
UNION
SELECT
    2 AS funnel_step,
    'signups' AS step_name,
    COUNT(*) AS user_count
FROM signups
--ride requests
UNION
SELECT
    3 AS funnel_step,
    'ride requests' AS step_name,
    COUNT(DISTINCT user_id) AS user_count
FROM ride_requests
--ride accepted
UNION
SELECT
    4 AS funnel_step,
    'ride accepted' AS step_name,
    COUNT(DISTINCT user_id) AS user_count
FROM ride_requests
WHERE accept_ts IS NOT NULL
-- ride completed
UNION
SELECT
    5 AS funnel_step,
    'ride completed' AS step_name,
    COUNT(DISTINCT user_id) AS user_count
FROM ride_requests
WHERE dropoff_ts IS NOT NULL
```

```

--payments done
UNION
SELECT
    6 AS funnel_step,
    'payment completed' AS step_name,
    COUNT(DISTINCT r.user_id) AS user_count
FROM transactions tt
LEFT JOIN ride_requests r
    ON tt.ride_id = r.ride_id
WHERE tt.charge_status = 'Approved'
-- reviews
UNION
SELECT
    7 AS funnel_step,
    'reviews' AS step_name,
    COUNT(DISTINCT user_id) AS user_count
FROM reviews
),
--resume and add drop_off
final_tab AS(
    SELECT *
FROM user_funnel
ORDER BY funnel_step
)
SELECT
    *,
    LAG(user_count,1) OVER() AS drop_off,
    ROUND((1.0 - user_count::numeric/LAG(user_count,1) OVER()),2) AS drop_off_rate,
    ROUND((user_count::numeric/LAG(user_count,1)OVER()),2) AS conversion_rate
FROM final_tab;

```

--2: Ride Funnel Analysis

```

WITH ride_funnel AS(
    -- requests
    SELECT
        1 AS funnel_step,
        'ride requests' AS step_name,
        COUNT(DISTINCT ride_id) AS ride_count

```

```

FROM ride_requests
WHERE request_ts IS NOT NULL
UNION
-- acceptance
SELECT
    2 AS funnel_step,
    'ride accepted' AS step_name,
    COUNT(DISTINCT ride_id) AS ride_count
FROM ride_requests
WHERE accept_ts IS NOT NULL
UNION
-- pickUp
SELECT
    3 AS funnel_step,
    'ride pickup' AS step_name,
    COUNT(DISTINCT ride_id) AS ride_count
FROM ride_requests
WHERE pickup_ts IS NOT NULL
UNION
-- dropoff
SELECT
    4 AS funnel_step,
    'ride dropoff' AS step_name,
    COUNT(DISTINCT ride_id) AS ride_count
FROM ride_requests
WHERE dropoff_ts IS NOT NULL
UNION
-- payments
SELECT
    5 AS funnel_step,
    'ride payment completed' AS step_name,
    COUNT(DISTINCT r.ride_id) AS ride_count
FROM transactions tt
LEFT JOIN ride_requests r
    ON tt.ride_id = r.ride_id
WHERE tt.charge_status = 'Approved'
UNION
-- reviews
SELECT
    6 AS funnel_step,
    'ride reviews' AS step_name,
    COUNT(DISTINCT ride_id) AS ride_count
FROM reviews

```

```

),
--resume and add drop_off
final_tab AS(
  SELECT *
FROM ride_funnel
ORDER BY funnel_step
)
SELECT
  *,
  LAG(ride_count,1) OVER() AS drop_off,
  (ROUND((1.0 - ride_count::numeric/LAG(ride_count,1) OVER()),2)*100) AS
drop_off_rate_percentage,
  (ROUND((ride_count::numeric/LAG(ride_count,1)OVER()),2)*100) AS
conversion_rate_percentage
FROM final_tab;

```

-- Query Table Tableau

--1: App Download and SignUps

```

SELECT
  ad.app_download_key,
  ad.platform,
  ad.download_ts,
  s.user_id,
  s.signup_ts,
  s.age_range
FROM app_downloads ad
LEFT JOIN signups s ON ad.app_download_key = s.session_id;

```

--2: Ride Requests and Payments

```

SELECT
  r.user_id,
  r.ride_id,
  r.driver_id,

```

```
r.request_ts,  
r.accept_ts,  
r.pickup_location,  
r.pickup_ts,  
r.dropoff_location,  
r.dropoff_ts,  
r.cancel_ts,  
tt.purchase_amount_usd,  
tt.charge_status  
FROM ride_requests r  
INNER JOIN transactions tt ON r.ride_id = tt.ride_id;
```

--3: Reviews

```
SELECT  
  r.ride_id,  
  r.driver_id,  
  r.user_id,  
  rr.rating  
FROM ride_requests r  
LEFT JOIN reviews rr ON r.ride_id = rr.ride_id;
```

--4: Metrocar Service Waiting Time

```
SELECT  
  r.ride_id,  
  Round((EXTRACT(EPOCH FROM (r.accept_ts - r.request_ts)) / 60),2) AS  
diff_Request_Accept,  
  Round((EXTRACT(EPOCH FROM (r.pickup_ts - r.accept_ts)) / 60),2) AS diff_Accept_Pickup,  
  Round((EXTRACT(EPOCH FROM (r.cancel_ts - r.accept_ts)) / 60),2) AS diff_Accept_Cancel,  
  Round((EXTRACT(EPOCH FROM (tt.transaction_ts - r.dropoff_ts)) / 60),2) AS  
diff_DropOff_Pay  
FROM ride_requests r  
LEFT JOIN transactions tt ON r.ride_id = tt.ride_id;
```

--5: User Funnel Data

```

WITH total_data AS (
  SELECT
    ad.app_download_key AS downloads,
    ad.platform,
    date_trunc('day', ad.download_ts) AS download_date,
    si.user_id AS user_signed_up,
    si.age_range,
    rr.user_id AS user_requested_ride,
    rr.ride_id AS ride_requested,
    CASE WHEN rr.request_ts IS NOT NULL THEN rr.user_id END AS user_request,
    CASE WHEN rr.accept_ts IS NOT NULL THEN rr.user_id END AS user_accepted,
    CASE WHEN rr.accept_ts IS NOT NULL THEN rr.ride_id END AS rides_accepted,
    CASE WHEN rr.dropoff_ts IS NOT NULL THEN rr.user_id END AS user_completed,
    CASE WHEN rr.dropoff_ts IS NOT NULL THEN rr.ride_id END AS rides_completed,
    CASE WHEN tr.charge_status = 'Approved' THEN rr.user_id END AS user_paid,
    CASE WHEN tr.charge_status = 'Approved' THEN tr.ride_id END AS rides_paid,
    re.ride_id AS ride_reviewed,
    re.user_id AS user_reviewed
  FROM app_downloads AS ad
  LEFT JOIN signups AS si ON ad.app_download_key = si.session_id
  LEFT JOIN ride_requests AS rr ON si.user_id = rr.user_id
  LEFT JOIN transactions AS tr ON rr.ride_id = tr.ride_id
  LEFT JOIN reviews AS re ON rr.ride_id = re.ride_id)
SELECT
  0 AS funnel_step,
  'Downloads' AS funnel_name,
  platform,
  age_range,
  download_date,
  COUNT(DISTINCT downloads) AS user_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
  1 AS funnel_step,
  'Sign Ups' AS funnel_name,
  platform,
  age_range,
  download_date,
  COUNT(DISTINCT user_signed_up) AS user_count
FROM total_data
GROUP BY platform, age_range, download_date

```

```

UNION
SELECT
    2 AS funnel_step,
    'Ride Requested' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT user_request) AS user_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    3 AS funnel_step,
    'Ride Accepted' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT user_accepted) AS user_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    4 AS funnel_step,
    'Ride Completed' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT user_completed) AS user_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    5 AS funnel_step,
    'Payments' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT user_paid) AS user_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    6 AS funnel_step,

```

```

'Reviews' AS funnel_name,
platform,
age_range,
download_date,
COUNT(DISTINCT user_reviwed) AS user_count
FROM total_data
GROUP BY platform, age_range, download_date;

```

--6: Ride Funnel Data

```

WITH total_data AS (
    SELECT
        ad.app_download_key AS downloads,
        ad.platform,
        date_trunc('day', ad.download_ts) AS download_date,
        si.user_id AS user_signed_up,
        si.age_range,
        rr.user_id AS user_requested_ride,
        rr.ride_id AS ride_requested,
        CASE WHEN rr.request_ts IS NOT NULL THEN rr.user_id END AS user_request,
        CASE WHEN rr.accept_ts IS NOT NULL THEN rr.user_id END AS user_accepted,
        CASE WHEN rr.accept_ts IS NOT NULL THEN rr.ride_id END AS rides_accepted,
        CASE WHEN rr.pickup_ts IS NOT NULL THEN rr.user_id END AS user_started,
        CASE WHEN rr.pickup_ts IS NOT NULL THEN rr.ride_id END AS rides_started,
        CASE WHEN rr.dropoff_ts IS NOT NULL THEN rr.user_id END AS user_completed,
        CASE WHEN rr.dropoff_ts IS NOT NULL THEN rr.ride_id END AS rides_completed,
        CASE WHEN tr.charge_status = 'Approved' THEN rr.user_id END AS user_paid,
        CASE WHEN tr.charge_status = 'Approved' THEN tr.ride_id END AS rides_paid,
        re.ride_id AS ride_reviewed,
        re.user_id AS user_reviwed
    FROM app_downloads AS ad
        LEFT JOIN signups AS si ON ad.app_download_key = si.session_id
        LEFT JOIN ride_requests AS rr ON si.user_id = rr.user_id
        LEFT JOIN transactions AS tr ON rr.ride_id = tr.ride_id
        LEFT JOIN reviews AS re ON rr.ride_id = re.ride_id)
SELECT
    1 AS funnel_step,
    'Ride Requested' AS funnel_name,
    platform,
    age_range,

```



```

        download_date,
        COUNT(DISTINCT ride_requested) AS ride_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    2 AS funnel_step,
    'Ride Accepted' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT rides_accepted) AS ride_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    3 AS funnel_step,
    'Ride Started' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT rides_started) AS ride_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    4 AS funnel_step,
    'Ride Completed' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT rides_completed) AS ride_count
FROM total_data
GROUP BY platform, age_range, download_date
UNION
SELECT
    5 AS funnel_step,
    'Ride Payments' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT rides_paid) AS ride_count
FROM total_data

```

```
GROUP BY platform, age_range, download_date
UNION
SELECT
    6 AS funnel_step,
    'Ride Reviews' AS funnel_name,
    platform,
    age_range,
    download_date,
    COUNT(DISTINCT ride_reviewed) AS ride_count
FROM total_data
GROUP BY platform, age_range, download_date;
```