

WITH

downloads AS (

SELECT 1 AS step, 'downloaded' AS funnel_step,

COUNT(app_download_key) AS users FROM app_downloads),

signups AS (

SELECT 2 AS step, 'signed up' AS funnel_step, COUNT(DISTINCT user_id) AS users

FROM signups), ride_requesters AS (

SELECT 3 AS step, 'ride requested' AS funnel_step, COUNT(DISTINCT user_id) AS users

FROM ride_requests), ride_accepted AS (

SELECT 4 AS step, 'ride accepted' AS funnel_step, COUNT(DISTINCT user_id) AS users

FROM ride_requests

WHERE accept_ts IS NOT NULL),

completed_ride AS (

SELECT 5 AS step, 'ride completed' AS funnel_step,

COUNT(DISTINCT user_id) AS users

FROM ride_requests AS rr

WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL),

successful_payment AS (

SELECT 6 AS step, 'ride paid' AS funnel_step, COUNT(DISTINCT user_id) AS users

```

FROM ride_requests AS rr
INNER JOIN transactions AS ts
ON rr.ride_id = ts.ride_id
WHERE ts.charge_status = 'Approved'),

reviewed_ride AS (

SELECT 7 AS step, 'reviewed' AS funnel_step, COUNT(DISTINCT user_id) AS
users

FROM reviews),

funnel AS (
SELECT *
FROM downloads
UNION ALL
SELECT *
FROM signups
UNION ALL
SELECT *
FROM ride_requesters UNION ALL
SELECT *
FROM ride_accepted UNION ALL
SELECT *
FROM completed_ride UNION ALL
SELECT *
FROM successful_payment UNION ALL
SELECT *
FROM reviewed_ride

ORDER BY step, users DESC)

SELECT

step AS funnel_step,

funnel_step AS funnel_name,

users AS Users,

```

COALESCE((LAG(users, 1) OVER(ORDER BY step, users DESC)- users), 0) AS Drop,

COALESCE(ROUND(((LAG(users, 1) OVER(ORDER BY step, users DESC)- users)::numeric/LAG(users, 1) OVER(ORDER BY step, users DESC)::numeric)*100, 2), 0) AS Drop_Percentage

FROM funnel;