```
-- Ratings Count

SELECT COUNT(review_id) AS Count_Of_Reviews, rating
FROM reviews
GROUP BY rating;

-- Age Range Across PlatformsSELECT COUNT(DISTINCT user_id),
age_range,platform
FROM metrocar
GROUP BY age_range, platform

— —Count of Failed Transactions and Dollars

SELECT charge_status, COUNT(charge_status)  AS num_of_transactions,
SUM(purchase_amount_usd) AS ammount_of_usd
FROM transactions
GROUP BY charge_status;

-- Full Query
WITH download AS
(SELECT platform, COUNT(DISTINCT app_download_key) AS Download_Count
 FROM metrocar
 GROUP BY platform
 ),

signup AS
(SELECT platform, COUNT(DISTINCT user_id) AS Signup_Count
 FROM metrocar
 GROUP BY platform
 ),

ride_requested AS
(SELECT platform, COUNT(DISTINCT user_id) AS Requested_Count
 FROM metrocar
 WHERE request_ts IS NOT NULL
 GROUP BY platform
  ),

ride_accepted AS
(SELECT platform, COUNT(DISTINCT user_id) AS Accepted_Count
 FROM metrocar
 WHERE accept_ts IS NOT NULL
 GROUP BY platform
  ),
ride_completed AS
(SELECT platform, COUNT(DISTINCT user_id) AS Completed_Count
 FROM metrocar
 WHERE dropoff_ts IS NOT NULL
 GROUP BY platform
  ),

payment_completed AS
(SELECT platform, COUNT(DISTINCT user_id) AS Payment_Completed_Count
 FROM metrocar
 WHERE charge_status = 'Approved'
 GROUP BY platform
  ),

review_completed AS
(SELECT platform, COUNT(DISTINCT user_id) AS Review_Completed_Count
 FROM metrocar
 WHERE review_id IS NOT NULL
 GROUP BY platform
  ),

totals AS (
    SELECT
        s.platform,
        COALESCE(SUM(Download_Count), 0) AS Download_Count,
        COALESCE(SUM(Signup_Count), 0) AS Signup_Count,
        COALESCE(SUM(Requested_Count), 0) AS Requested_Count,
        COALESCE(SUM(Accepted_Count), 0) AS Accepted_Count,
        COALESCE(SUM(Completed_Count), 0) AS Completed_Count,
        COALESCE(SUM(Payment_Completed_Count), 0) AS
Payment_Completed_Count,
        COALESCE(SUM(Review_Completed_Count), 0) AS
Review_Completed_Count

    FROM download d
    FULL JOIN signup s ON d.platform = s.platform
    FULL JOIN ride_requested rr ON s.platform = rr.platform
    FULL JOIN ride_accepted ra ON s.platform = ra.platform
    FULL JOIN ride_completed rc ON s.platform = rc.platform
    FULL JOIN payment_completed pc ON s.platform = pc.platform
    FULL JOIN review_completed rec ON s.platform = rec.platform
    GROUP BY s.platform

),
funnel_stages AS (
    SELECT

        1 AS funnel_step,
        'downloaded' AS funnel_name,
        platform,
        Download_Count AS user_count
    FROM totals

    UNION

    SELECT

        2 AS funnel_step,
        'signed_up' AS funnel_name,
        platform,
        Signup_Count AS user_count
    FROM totals

    UNION

    SELECT

        3 AS funnel_step,
        'ride_requested' AS funnel_name,
        platform,
        Requested_Count AS user_count
    FROM totals

    UNION

    SELECT

        4 AS funnel_step,
        'ride_accepted' AS funnel_name,
        platform,
        Accepted_Count AS user_count
    FROM totals

    UNION

    SELECT
        5 AS funnel_step,
        'ride_completed' AS funnel_name,
        platform,
        Completed_Count AS user_count
        FROM totals

        UNION

        SELECT
            6 AS funnel_step,
            'payment_completed' AS funnel_name,
            platform,
            Payment_Completed_Count AS user_count

        FROM totals

        UNION

        SELECT
            7 AS funnel_step,
            'review_completed' AS funnel_name,
            platform,
            Review_Completed_Count AS user_count
        FROM totals
    )
    -- The '*' brings everthing from each select statement in the funnel_stages table
    SELECT *,
            LAG(user_count)OVER (
            PARTITION BY platform
            ORDER BY funnel_step, platform) AS user_cnt_prev_step,
        ROUND((user_count::float / LAG(user_count) OVER (
            PARTITION BY platform
            ORDER BY funnel_step, platform)
            )::numeric,3) AS per_users_from_prev_step,
        ROUND((user_count::float / first_value(user_count) OVER (
            PARTITION BY platform
            ORDER BY funnel_step, platform)
            )::numeric,3) AS per_of_total
    FROM funnel_stages
    ORDER BY funnel_step,platform;
```