```
WITH
ride_requesters AS (
  SELECT 1 AS funnel_step, 'ride requested' AS funnel, COUNT(ride_id) AS rides
  FROM ride_requests),

ride_accepted AS (
  SELECT 2 AS funnel_step, 'ride accepted' AS funnel, COUNT(ride_id) AS rides
  FROM ride_requests
  WHERE accept_ts IS NOT NULL),

completed_ride AS (
  SELECT 3 AS funnel_step, 'ride completed' AS funnel, COUNT(rr.ride_id) AS
rides
  FROM ride_requests AS rr
  INNER JOIN transactions AS ts
  ON rr.ride_id = ts.ride_id
  WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL),

successful_payment AS (
  SELECT 4 AS funnel_step, 'ride paid' AS funnel, COUNT(rr.ride_id) AS rides
  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 5 AS funnel_step, 'ride reviewed' AS funnel, COUNT(rv.ride_id) AS
rides
  FROM ride_requests AS rr
  INNER JOIN transactions AS ts ON rr.ride_id = ts.ride_id
  INNER JOIN reviews AS rv ON ts.ride_id = rv.ride_id
  WHERE rr.accept_ts IS NOT NULL AND rr.dropoff_ts IS NOT NULL AND
ts.charge_status = 'Approved'),

funnel AS (
  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 rides DESC)

SELECT
  funnel_step AS funnel_step,
  funnel AS funnel_name,
  rides AS Rides,
  COALESCE((LAG(rides, 1) OVER(ORDER BY rides DESC)-rides), 0) AS
Drop,
  COALESCE(ROUND(((LAG(rides, 1) OVER(ORDER BY rides DESC)-
rides)::numeric/LAG(rides, 1) OVER(ORDER BY rides DESC)::numeric)*100,
2), 0) AS Drop_Percentage
FROM funnel;
```