--Funnels from Downloads to Unique Users and From Requested Rides to Approved Transactions (Pivot)

```
SELECT
   COALESCE(platform, 'Total') AS platform,
   COUNT(distinct d.app_download_key) AS download_count,
   COUNT(distinct s.session_id) AS signup_count,
   COUNT(distinct r.user_id) as unique_users,
   COUNT(r.request_ts) AS request_count,
   COUNT(r.accept_ts) AS accepted_request_count,
   COUNT(r.dropoff_ts) AS completed_rides,
   COALESCE(SUM(CASE WHEN t.charge_status = 'Approved' THEN 1 ELSE 0 END), 0)
AS actual_payment,
   ROUND(COALESCE(SUM(CASE WHEN t.charge_status = 'Approved' THEN 1 ELSE 0
END), 0) * 100.0 / COUNT(r.dropoff_ts), 1) AS collected_payment_ratio,
   ROUND(COUNT(distinct s.session_id) * 100.0 / COUNT(distinct
d.app_download_key), 1) AS signups_perc_of_downl,
   ROUND(COUNT(distinct r.user_id) * 100.0 / COUNT(distinct d.app_download_key), 1)
AS unique_users_perc_of_downl,
   ROUND(COUNT(distinct r.user_id) * 100.0 / COUNT(distinct s.session_id), 1) AS
unique_users_perc_of_signups,
   ROUND(COUNT(r.accept_ts) * 100.0 / COUNT(r.request_ts), 1) AS
accepted_req_share,
   ROUND(COUNT(r.dropoff_ts) * 100.0 / COUNT(r.request_ts), 1) AS
completed_rides_share,
   ROUND(COALESCE(SUM(CASE WHEN t.charge_status = 'Approved' THEN 1 ELSE 0
END), 0) * 100.0 / COUNT(r.request_ts), 1) AS paid_rides_share
FROM
   app_downloads AS d
LEFT JOIN
   signups AS s ON d.app_download_key = s.session_id
LEFT JOIN
```

```
   ride_requests as r ON s.user_id = r.user_id
LEFT JOIN
   transactions as t ON r.ride_id = t.ride_id
WHERE
   platform IN ('ios', 'android', 'web')
GROUP BY
   GROUPING SETS ((platform), ())
ORDER BY
   download_count DESC;
```

--Pivot by age group

```
SELECT
   COALESCE(s.age_range, 'Total') AS age_range,
   COUNT(distinct d.app_download_key) AS download_count,
   COUNT(distinct s.session_id) AS signup_count,
   COUNT(distinct r.user_id) as unique_users,
   COUNT(r.request_ts) AS request_count,
   COUNT(r.accept_ts) AS accepted_request_count,
   COUNT(r.dropoff_ts) AS completed_rides,
   COALESCE(SUM(CASE WHEN t.charge_status = 'Approved' THEN 1 ELSE 0 END), 0)
AS actual_payment,
   ROUND(
      COALESCE(SUM(CASE WHEN t.charge_status = 'Approved' THEN 1 ELSE 0 END), 0)
* 100.0
      / NULLIF(COUNT(r.dropoff_ts), 0), 1
   ) AS collected_payment_ratio,
   ROUND(
      COUNT(distinct s.session_id) * 100.0
      / NULLIF(COUNT(distinct d.app_download_key), 0), 1
```

```
  ) AS signups_perc_of_downl,
  ROUND(
    COUNT(distinct r.user_id) * 100.0
    / NULLIF(COUNT(distinct d.app_download_key), 0), 1
  ) AS unique_users_perc_of_downl,
  ROUND(
    COUNT(distinct r.user_id) * 100.0
    / NULLIF(COUNT(distinct s.session_id), 0), 1
  ) AS unique_users_perc_of_signups,
  ROUND(
    COUNT(r.accept_ts) * 100.0
    / NULLIF(COUNT(r.request_ts), 0), 1
  ) AS accepted_req_share,
  ROUND(
    COUNT(r.dropoff_ts) * 100.0
    / NULLIF(COUNT(r.request_ts), 0), 1
  ) AS completed_rides_share,
  ROUND(
    COALESCE(SUM(CASE WHEN t.charge_status = 'Approved' THEN 1 ELSE 0 END), 0)
* 100.0
    / NULLIF(COUNT(r.request_ts), 0), 1
  ) AS paid_rides_share
FROM
  app_downloads AS d
LEFT JOIN
  signups AS s ON d.app_download_key = s.session_id
LEFT JOIN
  ride_requests as r ON s.user_id = r.user_id
```

```
LEFT JOIN
  transactions as t ON r.ride_id = t.ride_id
WHERE
  platform IN ('ios', 'android', 'web')
GROUP BY
  GROUPING SETS ((age_range), ())
ORDER BY
  download_count DESC;

-- Downloads by platform
SELECT
  COALESCE(platform, 'Total') AS platform,
  COUNT(*) AS download_count,
  ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM app_downloads
                WHERE platform IN ('ios', 'android', 'web')), 1) AS percentage_share
FROM
  app_downloads
WHERE
  platform IN ('ios', 'android', 'web')
GROUP BY
  GROUPING SETS ((platform), ())
ORDER BY
                download_count DESC;
```

--- Time stamps funnel

-- in min

```
SELECT
   AVG(EXTRACT(EPOCH FROM (s.signup_ts - d.download_ts)) / 60)::float AS
average_time_from_DNL_to_signup,
   AVG(EXTRACT(EPOCH FROM (r.request_ts - s.signup_ts)) / 60)::float AS
average_time_from_signup_to_request,
   AVG(EXTRACT(EPOCH FROM (t.transaction_ts - r.request_ts)) / 60)::float AS
average_time_from_request_to_payment,
   AVG(EXTRACT(EPOCH FROM (t.transaction_ts - d.download_ts)) / 60)::float AS
average_time_from_DNL_to_payment,
   AVG(EXTRACT(EPOCH FROM (t.transaction_ts - s.signup_ts)) / 60)::float AS
average_time_from_DNL_to_payment
FROM
   app_downloads AS d
LEFT JOIN
   signups AS s ON d.app_download_key = s.session_id
LEFT JOIN
   ride_requests as r ON s.user_id = r.user_id
LEFT JOIN
   transactions as t ON r.ride_id = t.ride_id
```

-- in hours

```
SELECT
   AVG(EXTRACT(EPOCH FROM (s.signup_ts - d.download_ts)) / 3600)::float AS
average_time_from_DNL_to_signup,
   AVG(EXTRACT(EPOCH FROM (r.request_ts - s.signup_ts)) / 3600)::float AS
average_time_from_signup_to_request,
   AVG(EXTRACT(EPOCH FROM (t.transaction_ts - r.request_ts)) / 3600)::float AS
average_time_from_request_to_payment,
```

```
  AVG(EXTRACT(EPOCH FROM (t.transaction_ts - d.download_ts)) / 3600)::float AS
average_time_from_DNL_to_payment,
  AVG(EXTRACT(EPOCH FROM (t.transaction_ts - s.signup_ts)) / 3600)::float AS
average_time_from_DNL_to_payment
FROM
  app_downloads AS d
LEFT JOIN
  signups AS s ON d.app_download_key = s.session_id
LEFT JOIN
  ride_requests as r ON s.user_id = r.user_id
LEFT JOIN
  transactions as t ON r.ride_id = t.ride_id;

-- in days
SELECT
  AVG(EXTRACT(EPOCH FROM (s.signup_ts - d.download_ts)) / 86400)::float AS
average_time_from_DNL_to_signup,
  AVG(EXTRACT(EPOCH FROM (r.request_ts - s.signup_ts)) / 86400)::float AS
average_time_from_signup_to_request,
  AVG(EXTRACT(EPOCH FROM (t.transaction_ts - r.request_ts)) / 86400)::float AS
average_time_from_request_to_payment,
  AVG(EXTRACT(EPOCH FROM (t.transaction_ts - d.download_ts)) / 86400)::float AS
average_time_from_DNL_to_payment
FROM
  app_downloads AS d
LEFT JOIN
  signups AS s ON d.app_download_key = s.session_id
LEFT JOIN
  ride_requests as r ON s.user_id = r.user_id
LEFT JOIN
```

```
   transactions as t ON r.ride_id = t.ride_id;


--- Ride timing (min)

SELECT

   AVG(EXTRACT(EPOCH FROM (r.accept_ts - r.request_ts)) / 60)::float AS
average_time_from_request_to_accepting,

   AVG(EXTRACT(EPOCH FROM (r.pickup_ts - r.accept_ts)) / 60)::float AS
average_time_from_accepting_to_pickup,

   AVG(EXTRACT(EPOCH FROM (r.dropoff_ts - r.pickup_ts)) / 60)::float AS
average_ride_duration_minutes,

   AVG(EXTRACT(EPOCH FROM (t.transaction_ts - r.dropoff_ts)) / 60)::float AS
average_time_from_ride_to_payment

FROM

   ride_requests as r

LEFT JOIN

   transactions as t ON r.ride_id = t.ride_id


--- revenue & paid_users

SELECT

                COUNT(DISTINCT r.user_id) as paid_users,

   SUM(t.purchase_amount_usd) as total_revenue,

                SUM(t.purchase_amount_usd)/count(distinct r.user_id) as
revenue_per_paid_user,

   COUNT(DISTINCT t.ride_id)/count(distinct r.user_id) as rides_per_paid_user,

   AVG(t.purchase_amount_usd) as average_ride_payment

FROM ride_requests as r

LEFT JOIN transactions as t

ON r.ride_id = t.ride_id

WHERE t.charge_status = 'Approved'
```

```
-- Paid users & "happy" paid users
SELECT
   COUNT(DISTINCT r.user_id) AS paid_users,
   COUNT(DISTINCT r.user_id) - COUNT(DISTINCT re.user_id) AS non_reviewed_users,
   COUNT(DISTINCT re.user_id) AS reviewed_users
FROM
   ride_requests AS r
LEFT JOIN
   reviews AS re ON r.user_id = re.user_id
LEFT JOIN
   transactions AS t ON r.ride_id = t.ride_id
WHERE
   t.charge_status = 'Approved';
```