SQL queries used in Metrocar database

**ACCEPT TO PICK UP (BY TIME):**

with time\_group\_pickup as (

select

case

when extract(epoch from (pickup\_ts - accept\_ts)) < 180 then '0-3'

else

floor(extract(epoch from (pickup\_ts - accept\_ts)) / 180) \* 3 || '-' ||

(floor(extract(epoch from (pickup\_ts - accept\_ts)) / 180) \* 3 + 3)::text

end as time\_group, user\_id,

row\_number() over (partition by user\_id order by accept\_ts) as user\_row

from ride\_requests

where

extract(epoch from (pickup\_ts - accept\_ts)) is not null

)

select time\_group ,

count(distinct user\_id) as user\_count

from time\_group\_pickup

where user\_row = 1

group by 1

order by 1

**Funnel by Age:**

with

download as (  
select 'download' as funnel ,count(distinct ap.app\_download\_key) , su.age\_range   
from app\_downloads as ap  
left join signups as su  
on ap.app\_download\_key = su.session\_id  
group by 3) ,

signup as (  
select 'signup' as funnel, count(distinct user\_id) , age\_range  
from signups  
group by 3),  
  
ride\_request as (  
select 'ride\_request' as funnel ,count(distinct rr.user\_id) , su.age\_range  
 from ride\_requests as rr  
left join signups as su  
on rr.user\_id = su.user\_id  
where request\_ts is not null  
 group by 3),

ride\_accept as (   
 select 'ride\_accept' as funnel ,count(distinct rr.user\_id) , su.age\_range  
 from ride\_requests as rr  
 left join signups as su  
on rr.user\_id = su.user\_id  
 where accept\_ts is not null  
 group by 3),

ride\_complete as (   
select 'ride\_complete' as funnel, count(distinct rr.user\_id) , su.age\_range  
from ride\_requests as rr  
left join signups as su  
on rr.user\_id = su.user\_id  
where dropoff\_ts is not null   
group by 3),

funnel as (  
select \* from download  
union all  
select \* from signup  
union all  
select \* from ride\_request  
union all  
select \* from ride\_accept  
union all  
select \* from ride\_complete)

select \* from funnel

**FUNNEL BY PLATFORM:**

With

downloads as (  
select 'download' as funnel, platform, count(distinct app\_download\_key) as user  
from metrocar  
group by 2 ),

signup as (  
select 'signup' as funnel, platform, count(distinct user\_id) as users  
from metrocar  
group by 2),

ride\_requesters as (

select 'ride\_requesters' as funnel, platform, count(distinct user\_id) as users

from metrocar

where request\_ts is not null

group by 2),

ride\_accepted as (

select 'ride\_accepted' as funnel, platform, count(distinct user\_id) as users

from metrocar

where accept\_ts is not null

group by 2),

ride\_complete as (

select 'ride\_complete' as funnel, platform, count(distinct user\_id) as users

from metrocar

where dropoff\_ts is not null

group by 2),

ride\_paid as (

select 'ride\_paid' as funnel, platform, count(distinct user\_id) as users

from metrocar

where charge\_status = 'Approved'

group by 2),

funnel as (

select \*

from downloads

union all

select \*

from signup

union all

select \*

from ride\_requesters

union all

select \*

from ride\_accepted

union all

select \*

from ride\_complete

union all

select \*  
 from ride\_paid)

select funnel, users, platform

from funnel

order by 3,

case funnel

when 'download' then 1

when 'signup' then 2

when 'ride\_requesters' then 3

when 'ride\_accepted' then 4

when 'ride\_complete' then 5

when 'ride\_paid' then 6

else 7

end,

1

**User Funnel + % drop:**

with downloads as (

select 'download' as funnel, count(distinct app\_download\_key) as users

from metrocar   
),

signup as (

select 'signup' as funnel, count(distinct user\_id) as users

from metrocar  
),

ride\_requesters as (

select 'ride\_requesters' as funnel, count(distinct user\_id) as users

from metrocar

where request\_ts is not null  
),

ride\_accepted as (

select 'ride\_accepted' as funnel, count(distinct user\_id) as users

from metrocar

where accept\_ts is not null  
),

ride\_complete as (

select 'ride\_complete' as funnel, count(distinct user\_id) as users

from metrocar

where dropoff\_ts is not null  
),

ride\_paid as (

select 'ride\_paid' as funnel, count(distinct user\_id) as users

from metrocar

where charge\_status = 'Approved'  
),

reviewed\_ride as (

select 'reviewed' as funnel, count(distinct user\_id) as users

from reviews  
),

funnel as (

select \*

from downloads

union all

select \*

from signup

union all

select \*

from ride\_requesters

union all

select \*

from ride\_accepted

union all

select \*

from ride\_complete

union all

select \*

from ride\_paid

union all

select \*

from reviewed\_ride  
)

select funnel, users,

coalesce((lag(users, 1)

over(order by users desc)-users),0) as drop,

coalesce(round(((lag(users, 1)

over(order by users desc)-users)::numeric/lag(users, 1)

over(order by users desc)::numeric)\*100, 2),0) as drop\_pct

from funnel

**RIDE FUNNEL + % DROP:**

with

ride\_requesters as (

select 'ride requested' as funnel, count(distinct ride\_id) as rides

from metrocar),

ride\_accepted as (

select 'ride accepted' as funnel, count(distinct ride\_id) as rides

from metrocar

where accept\_ts is not null),

complete\_ride as (

select 'ride completed' as funnel, count(distinct ride\_id) as rides

from metrocar

where accept\_ts is not null and dropoff\_ts is not null ),

successful\_payment as (

select 'ride paid' as funnel, count(distinct ride\_id) as rides

from metrocar

where charge\_status = 'Approved'),

reviewed\_ride as (

select '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 complete\_ride

union all

select \*

from successful\_payment

union all

select \*

from reviewed\_ride

order by rides desc

)

select

funnel,

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\_pct

from funnel

order by

case funnel

when 'ride requested' then 1

when 'ride accepted' then 2

when 'ride completed' then 3

when 'ride paid' then 4

when 'ride reviewed' then 5

else 7

end