Metrocar- Funnel Analysis

Introduction

This project aims to analyze the customer funnel of our company. As a ride-sharing app, we need to identify areas for improvement and optimization regularly. Similar to other customer funnels, there will be drop-offs at every stage of the funnel, which is why funnel analysis can help identify areas for improvement and optimization. The stakeholders have asked several business questions that can uncover valuable insights for improving specific areas of the customer funnel, which will be proceeded and explained in this report. We will analyze the company data, build a funnel analysis, and will present recommendations based on insights retrieved from it.

Context

In this project, we used SQL for data analysis and Tableau to visualize our data.
As a reminder, our funnel analysis includes the 5 ordered steps the users should go through our app, with the aim he will get up to the end of the funnel, so we could get our revenue. The steps the user should follow (in this order) are:
Download the Metrocar app→ signup into the app → request a ride→ the driver accepts the user's ride request→ the ride occurs and the driver picks up the user and takes him to the destination→ when the ride is over, the user is charged through the app, and a receipt is sent→ the user is prompted to rate their driver and leave a review of their ride experience.

Dataset structure

The dataset includes 5 tables. Here below is a description of each table and its columns.

- **app_downloads**: contains information about app downloads
  - app_download_key: unique id of an app download
  - platform: ios, android or web
  - download_ts: download timestamp
- **signups**: contains information about new user signups
  - user_id: primary id for a user
  - session_id: id of app download
  - signup_ts: signup timestamp
  - age_range: the age range the user belongs to
- **ride_requests**: contains information about rides
  - ride_id: primary id for a ride
  - user_id: foreign key to user (requester)
  - driver_id: foreign key to driver
  - request_ts: ride request timestamp

    - accept_ts: driver accept timestamp
    - pickup_location: pickup coordinates
    - destination_location: destination coordinates
    - pickup_ts: pickup timestamp
    - dropoff_ts: dropoff timestamp
    - cancel_ts: ride cancel timestamp
- **transactions**: contains information about financial transactions based on completed rides:
    - ride_id: foreign key to ride
    - purchase_amount_usd: purchase amount in USD
    - charge_status: approved, canceled
    - transaction_ts: transaction timestamp
- **reviews**: contains information about driver reviews once rides are completed
    - review_id: primary id of review
    - ride_id: foreign key to ride
    - driver_id: foreign key to driver
    - user_id: foreign key to user (requester)
    - rating: rating from 0 to 5
    - free_response: text response that was given by the user/requester

It is worth noting that during the right request, the user can cancel a ride before the driver arrives, which then we will have data about the cancellation like the drop-off time, but not the pick-up information. In a complicated ride, the case will be the opposite. Where there is no data, the empty place may be null.

In addition, other important notes are about our analysis metrics and analysis level. The results and the recommendation in this report lie on the User-Funnel view, which means the user funnel would be the entire funnel at the user-level granularity and start from the first step (the app_downloads table) in the dataset. This level of granularity is very high and was the reason for choosing to analyze the dataset by it.

More than this, the analysis and the results represent a “percent of previous” approach. This calculation involves measuring the conversion rate as a percentage of the users who proceeded to the current stage of the funnel, relative to the number of users who were at the previous stage. In other words, this metric tracks the progression of users through each stage of the funnel. This helps to track the progression of users through each step and identify potential areas of improvement or drop-offs.

As said, the first step included using SQL to explore the platform data and understand it more deeply. The analysis and the queries were done in the bit.io platform, using PostgreSQL.
You can find the full, raw dataset here: https://bit.io/norbinn_masterschool/metrocar?tab=Data .
All the queries and their results can be found in the appendix.

After this, we moved to develop the Metrocar funnel metrics. This stage was done in Tableau, looking after the user conversion rate and drop-off through the funnel steps, to see what could be weak points in our platform. Segmenting by age range for example, and the user platform (IOS, Android, Web), were also explored in Tableau to see if specific segments can show different trends than others and to understand better what to focus on.
Here in Tableau, we want to keep track of metrics and KPIs over time and find trends in the dataset that can lead to the meaningful inside, with a more aggregated level of data than was done in SQL.

Results
Analysis in SQL and Tableau shows that at the User-Level through the funnel, considering the "percent of previous" approach- we started with a total of **23,608 users that downloaded our app** (on any platform. **Moving to the signup step, the amount went down to 17,623 users, meaning a 74.65% conversion rate between the two first steps.** In the following steps, the conversion rates (the number of users that move to this step from the previous step in the funnel) are:
70.40% between the signups and the ride requests (a drop-off of 29.6% from the previous step)
50.24% between the ride requests and transactions (a drop-off of 49.76% from the previous step).
69.76% between transactions and reviews (a drop-off of 30.24% from the previous step).

At the end of the funnel, we ended with 4,348 users that made up to the bottom of the funnel- out of 6,233 from the previous step (transactions), and out of 23,608 who started our funnel and downloaded the Metrocar app.

More finds were that out of 385,477 users that asked for a ride, only 248,379 were accepted by drivers, and only 223,652 rides were completed. It's good to mention that all of those ride requests represent only 12,406 unique users. The reason is that every user can make more than one ride request.

From the completed rides we successfully collected payments from 212,628 users, which sums up to about 4,251,667 dollars.

Looking even more deeply into the ride requests, considering the three different platforms a user can come to our app, we can see that 112,317 users ask for a ride from Android, 234,693 users used iOS, and 38,467 used the web. The meaning of this finding and the others will be discussed in the “Conclusion and Recommendations” part.

Another interesting found is that an average time of a ride from pick up to drop off is about 52 minutes.

Keeping to Tableau I was able to visualize our aggregated data and also look for answers to the business questions I got, which will be shown in the next chapter. Important points that were taken from the visualizations, which are also related to these questions are:

- the biggest drop-off through the funnel was between ride requests to transactions. The drop-off in this area was almost half of the users from our last step and stands at 49.76%. that means that the conversion rate here is the lowest from all the stages, standing on a similar size- 50.24%. looking at the row numbers, users went down from 12,406 in the ride requests step (unique users), to 6,233 users that did at least one transaction.

- in each of the five steps, most of the users used the ios platform to go through our funnel, with about 60% of the users in each step. In contrast, the least amount of users arrived at our Metrocar app using the web, representing between 9.75% - 10.1% of the users in each step. About 30% of the users in each step use the third possible app, Android.

  In addition, we used Tableau to visualize the last results in a graphic way that can help to see things from the other side- not just looking at each step as its own but also looking at the behavior of each platform (Android, iOS, Web) along the funnel. It was found that the conversion rates in each platform along the funnel are pretty similar in each step:

  > An average of about 74% from download to signup, about 70% conversion rate from signup to ride request, about 50% from ride requests to transactions, and 69% from transactions to reviews (the last step).

- The age range 35-44 is the most common in all the funnel steps, with a little over 40% in each step. The list one is age range 45-54, with about 14% in each step. It's good to mention that some users have a null in the age-range column, and the percentage that is shown here isn't including them in the calculations.

- Talking about the rushing hours of our app is also something important and interesting to look for, which was easy to see in Tableau. It was found that the most traffic in our app for ride requests (step 3 in the funnel) is between 8 AM - 9 AM, and between 4 PM - 5 PM.
  Looking at each day of the week separately didn't show a difference in this trend.

<u>Conclusion and Recommendations</u>

The Metrocar app data is very rich in tables and columns, describing well each of the users and their process in each of the steps. As a first thing, the analysis started by thinking about what types of funnel summaries and visualizations would be necessary to answer the business questions. Through all the analysis we wanted to analyze the funnel at a user-level, and to be able to do this we had to maintain a view of each table at a user-level granularity (for example, keep track of the user id in the signups step. In addition, we used to use the "percent of previous" method which was discussed earlier in this report.
Although the dataset was pretty clean, there was a need to remove the nulls in some cases, and also use different calculation methods along the analysis- in SQL we searched for a more detailed level of the data and performed more granular queries. In Tablaua on the other hand, we wanted to summarize the data at a higher level and therefore used aggregated data.
Through all the analysis we wanted to analyze the funnel at a user-level, and to be able to do this we had to maintain a view of each table at a user-level granularity (for example, keep track of the user id in the signups step. In addition, we used to use the "percent of previous" method which was discussed earlier in this report.

Back to the business questions we have asked to answer, and after analyzing the data and getting the results shown above, we found some important and meaningful insight for our company, that can help us to make even better performance in our ride-sharing app.
<u>Below you can see the business questions. We will answer them one by one:</u>

- **What steps of the funnel should we research and improve? Are there any specific drop-off points preventing users from completing their first ride?**

Looking at the visualization of the funnel, it looks that from all the users that download our app, only ¾ of them also signed up, and from those who signed up, only about 70% of them requested at least one ride. It could be possible that specific platform/s (iOS/android/web) are less convenient for the user to sign up from, nor also to make a ride request after the signup step. Therefore, we checked the movement through those areas for each platform by itself, but couldn't see a difference in the conversion rates for those two steps between the three platforms. due to this finding, there is a possibility that the issue for some users isn't related to the platform the user came from, but something more comprehensive needs to be checked.

- **Metrocar currently supports 3 different platforms: iOS, android, and web. To recommend where to focus our marketing budget for the upcoming year, what insights can we make based on the platform?**

We could see that most of the users that downloaded our app are using an iOS for that (about 60%). This is not surprising since the iOS platform is very popular in our world those days, but what is surprising is that only about 29.5% of the people used Android to download our app. This is interesting because the year 2021 was the year when Samsung actually sold more phones than Apple, after 2 years when Apple selling's were bigger, and it is also known that the Android platform have the biggest market share in the last decade.
To summarize this point, it could be that users who have iOS tend to use those kinds of apps more than people who use Android or the web.
The last platform, which is the web, represents only about 10.5%. This is reasonable because people download an app and used it (especially if it's a ride-sharing app) through the phone itself.

- **What age groups perform best at each stage of our funnel? Which age group(s) likely contain our target customers?**

It was very striking from the visualization to see that the 35-44 age group performed best at each stage of the funnel. This can be taken into account when we advertise our app and for other promotional events.

- **If we want to adopt a price-surging strategy, what does the distribution of ride requests look like throughout the day?**

The best hours to adopt a price-surging are 8:00-9:00 in the mornings, and 4:00-5:00 in the afternoon, where above 8,000 users use our app. Those hours are also reasonable since people go to work and come back home around those hours of the day, and not everyone prefers nor can use a private car or bus as public transportation for example. I suggest increasing the number of available drivers in those hours, so we could supply more from our service and also our users will be satisfied with the waiting time and will not have to wait long. Also, there is no need to do a surge pricing by specific day of the week (for example weekdays VS weekends) since there is no different trend between them.

- **What part of our funnel has the lowest conversion rate? What can we do to improve this part of the funnel?**

We should look better for the area between the ride request to the transaction, where the user completes the ride and should pay for it. This is because this stage has the biggest drop-off along the funnel (49.76%).
We should check what is the reason that half of the customers aren't paying for their ride since this is a very serious matter and we don't earn as we should. I suggest that there is a need to check the payment process in each platform and see if there is any issue (logical, visual, or functional), that leads to the fact that half of the users aren't completing the transaction step.
In addition, there is no difference between the 3 platforms or the different age groups when looking at this stage of the funnel, so the problem shouldn't be specific to one of those segments.

As a last note, it would be nice if there were additional classifications that could be used to segment the data and investigate according to them, such as the gender of the user and the number of people in the car.
Also, information and reviews specifically from the side of the drivers can improve their experience in our company, as well as user experience, which will ultimately contribute to increasing our profit.

Appendix

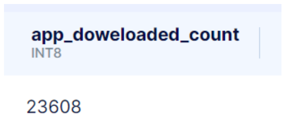
Link to Tableau:

https://public.tableau.com/views/Metrocar-FunnelAnalysis/MetrocarFunnelAnalysis?:language=en-US&publish=yes&:display_count=n&:origin=viz_share_link

SQL Queries:

1. **How many times was the app downloaded?**

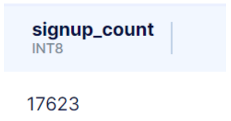
```
SELECT COUNT(*) as app_doweloaded_count
FROM app_downloads;
```



app_doweloaded_count INT8
23608

2. **How many users signed up on the app?**

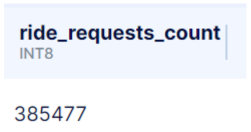
```
SELECT COUNT(DISTINCT user_id) signup_count
FROM signups;
```



signup_count INT8
17623

3. **How many rides were requested through the app?**

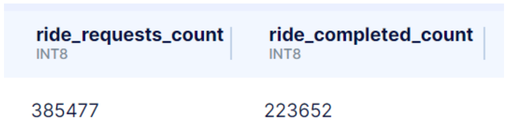
```
SELECT COUNT(DISTINCT ride_id) ride_requests_count
FROM ride_requests;
```



ride_requests_count INT8
385477

4. **How many rides were requested and completed through the app?**

```
SELECT COUNT(DISTINCT ride_id) ride_requests_count,
    COUNT(dropoff_ts) ride_completed_count
FROM ride_requests;
```



ride_requests_count INT8	ride_completed_count INT8
385477	223652

5. **How many rides were requested and how many unique users requested a ride?**

```
SELECT COUNT(DISTINCT ride_id) as ride_requests_count,
    COUNT(DISTINCT user_id) as distinct_user_requested
FROM ride_requests;
```

ride_requests_count INT8	distinct_user_requested INT8
385477	12406

6. **What is the average time of a ride from pick up to drop off?**

```
SELECT AVG(dropoff_ts-pickup_ts) as avg_time_ride
FROM ride_requests;
```

avg_time_ride INTERVAL
52 minutes 36.738773 seconds

7. **How many rides were accepted by a driver?**

```
SELECT COUNT(accept_ts) as driver_accept_count
FROM ride_requests;
```

driver_accept_count INT8
248379

8. **How many rides did we successfully collect payments and how much was collected?**

```
SELECT COUNT(ride_id) as success_payments,
    ROUND(SUM(purchase_amount_usd)::NUMERIC, 3) as total_payments_amount
FROM transactions
WHERE charge_status = 'Approved'
```

success_payments INT8	total_payments_amount NUMERIC
212628	4251667.610

9. **How many ride requests happened on each platform?**

```
SELECT platform,
    COUNT(request_ts) as ride_requests_count
FROM app_downloads ad
JOIN signups s
ON ad.app_download_key = s.session_id
JOIN ride_requests ride_r
ON s.user_id = ride_r.user_id
GROUP BY platform;
```

platform TEXT	ride_requests_count INT8
android	112317
ios	234693
web	38467

10. **What is the drop-off from users signing up to users requesting a ride?**

```
11. WITH t AS(
    SELECT COUNT(DISTINCT s.user_id)::NUMERIC as signup_count,
        COUNT(DISTINCT ride_r.user_id)::NUMERIC as requests_count
    FROM signups s
    LEFT JOIN ride_requests ride_r
    ON s.user_id = ride_r.user_id
)
SELECT signup_count,
    requests_count,
    ROUND(((signup_count-requests_count)/signup_count)*100, 3) as
signup_to_request_dropoff
FROM t;

```

signup_count NUMERIC	requests_count NUMERIC	signup_to_request_dropoff NUMERIC
17623	12406	29.603

11. **How many unique users <u>completed</u> a ride through the Metrocar app?**

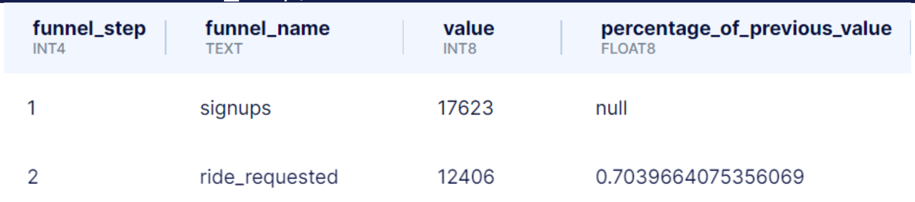
```
WITH user_ride_status AS (
    SELECT
        user_id,
        MAX(
            CASE
                WHEN dropoff_ts IS NOT NULL
                THEN 1
                ELSE 0
            END
        ) AS ride_completed
    FROM ride_requests
    GROUP BY user_id
)
SELECT
    COUNT(*) AS total_users_ride_requested,
    SUM(ride_completed) AS total_users_ride_completed
FROM user_ride_status;
```

total_users_ride_requested INT8	total_users_ride_completed INT8
12406	6233

12. **Of the users that signed up on the app, what percentage of these users requested a ride?**

```
WITH user_ride_status AS (
    SELECT
        user_id
    FROM ride_requests
    GROUP BY user_id
),
totals AS (
```

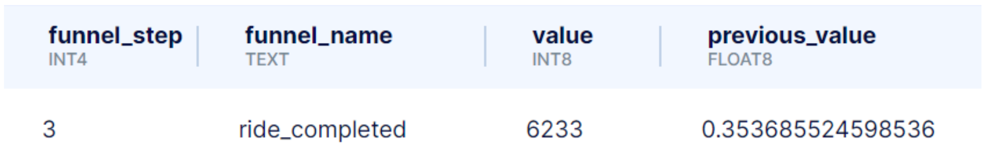
```sql
    SELECT
        COUNT(*) AS total_users_signed_up,
        COUNT(DISTINCT urs.user_id) AS total_users_ride_requested
    FROM signups s
    LEFT JOIN user_ride_status urs ON
        s.user_id = urs.user_id
),
funnel_stages AS (
    SELECT
        1 AS funnel_step,
        'signups' AS funnel_name,
        total_users_signed_up AS value
    FROM totals

    UNION

    SELECT
        2 AS funnel_step,
        'ride_requested' AS funnel_name,
        total_users_ride_requested AS value
    FROM totals
)
SELECT *,
    value::float / LAG(value) OVER (
        ORDER BY funnel_step
    ) AS percentage_of_previous_value
FROM funnel_stages

ORDER BY funnel_step;
```



funnel_step INT4	funnel_name TEXT	value INT8	percentage_of_previous_value FLOAT8
1	signups	17623	null
2	ride_requested	12406	0.7039664075356069

13. **Of the users that signed up on the app, what percentage of these users completed a ride?**

```sql
WITH user_ride_status AS (
    SELECT
        user_id,
        MAX(
            CASE
                WHEN dropoff_ts IS NOT NULL
                THEN 1
                ELSE 0
            END
        ) AS ride_completed
    FROM ride_requests
    GROUP BY user_id
),

totals AS (
    SELECT
        COUNT(*) AS total_users_signed_up,
        COUNT(urs.user_id) AS total_users_ride_requested,
        SUM(ride_completed) AS total_users_ride_completed
    FROM signups s
    LEFT JOIN user_ride_status urs ON
        s.user_id = urs.user_iD
),

funnel_stages AS (
    SELECT
        1 AS funnel_step,
        'signups' AS funnel_name,
        total_users_signed_up AS value
    FROM totals

    UNION

    SELECT
        2 AS funnel_step,
        'ride_requested' AS funnel_name,
        total_users_ride_requested AS value
    FROM totals

    UNION

        SELECT
        3 AS funnel_step,
        'ride_completed' AS funnel_name,
        total_users_ride_completed AS value
    FROM totals

),

funnel_steps AS (
    SELECT *,
    value::float / LAG(value, 2) OVER (
        ORDER BY funnel_step
    ) AS previous_value
FROM funnel_stages

ORDER BY funnel_step
)

SELECT *
FROM funnel_steps
WHERE funnel_name = 'ride_completed';
```



funnel_step INT4	funnel_name TEXT	value INT8	previous_value FLOAT8
3	ride_completed	6233	0.353685524598536

14. **Using the *"percent of <u>previous</u>"* approach, what are the user-level conversion rates for the first 3 stages of the funnel (*app download to signup* and *signup to ride requested*)?**

```
WITH user_ride_status AS (
    SELECT
        user_id
    FROM ride_requests
    GROUP BY user_id
),

user_download_status AS (
    SELECT
        count(distinct app_download_key) as total_users_app_download
    FROM app_downloads

),

totals AS (
    SELECT
        COUNT(*) AS total_users_signed_up,
        COUNT(DISTINCT urs.user_id) AS total_users_ride_requested
    FROM signups s
    LEFT JOIN user_ride_status urs ON
        s.user_id = urs.user_id

),
funnel_stages AS (
    SELECT
        1 AS funnel_step,
        'app_doenloads' AS funnel_name,
        (SELECT
        count(distinct app_download_key) as total_users_app_download
    FROM app_downloads) AS value
    FROM totals

    UNION

    SELECT
        2 AS funnel_step,
        'signups' AS funnel_name,
        total_users_signed_up AS value
    FROM totals
    UNION

    SELECT
        3 AS funnel_step,
        'ride_requested' AS funnel_name,
        total_users_ride_requested AS value
    FROM totals
)
SELECT *,
    value::float / LAG(value) OVER (
        ORDER BY funnel_step
    ) AS previous_value
FROM funnel_stages

ORDER BY funnel_step;
```

funnel_step INT4	funnel_name TEXT	value INT8	previous_value FLOAT8
1	app_doenloads	23608	null
2	signups	17623	0.7464842426296171
3	ride_requested	12406	0.7039664075356069

15. **Using the *“percent of <u>top</u>”* approach, what are the user-level conversion rates for the first 3 stages of the funnel (*app download to signup* and *signup to ride requested*)?**

```
WITH user_ride_status AS (
    SELECT
        user_id
    FROM ride_requests
    GROUP BY user_id
),

user_download_status AS (
    SELECT
        count(distinct app_download_key) as total_users_app_download
    FROM app_downloads

),

totals AS (
    SELECT
        COUNT(*) AS total_users_signed_up,
        COUNT(DISTINCT urs.user_id) AS total_users_ride_requested
    FROM signups s
    LEFT JOIN user_ride_status urs ON
        s.user_id = urs.user_id

),
funnel_stages AS (
    SELECT
```

```
        1 AS funnel_step,
        'app_doenloads' AS funnel_name,
        (SELECT
        count(distinct app_download_key) as total_users_app_download
    FROM app_downloads) AS value
    FROM totals

    UNION

    SELECT
        2 AS funnel_step,
        'signups' AS funnel_name,
        total_users_signed_up AS value
    FROM totals

    UNION

    SELECT
        3 AS funnel_step,
        'ride_requested' AS funnel_name,
        total_users_ride_requested AS value
    FROM totals
)
SELECT *,
    value::float / first_value(value) OVER (
        ORDER BY funnel_step
    ) AS previous_value
FROM funnel_stages

ORDER BY funnel_step;
```

```
1          app_doenloads      23608       1

2          signups            17623       0.7464842426296171

3          ride_requested     12406       0.5254998305659099
```

16. **Using the *"percent of __previous__"* approach, what are the user-level conversion rates for the following 3 stages of the funnel? 1. signup, 2. ride requested, 3. ride completed**

```
WITH user_ride_status AS (
    SELECT
        user_id,
        MAX(
            CASE
                WHEN dropoff_ts IS NOT NULL
                THEN 1
                ELSE 0
            END
        ) AS ride_completed
    FROM ride_requests
    GROUP BY user_id
),

totals AS (
    SELECT
        COUNT(*) AS total_users_signed_up,
        COUNT(urs.user_id) AS total_users_ride_requested,
        SUM(ride_completed) AS total_users_ride_completed
    FROM signups s
    LEFT JOIN user_ride_status urs ON
        s.user_id = urs.user_iD
),

funnel_stages AS (
    SELECT
        1 AS funnel_step,
        'signups' AS funnel_name,
        total_users_signed_up AS value
    FROM totals

    UNION

    SELECT
        2 AS funnel_step,
        'ride_requested' AS funnel_name,
        total_users_ride_requested AS value
    FROM totals

    UNION

        SELECT
        3 AS funnel_step,
        'ride_completed' AS funnel_name,
        total_users_ride_completed AS value
    FROM totals

)
SELECT *,
    value::float / LAG(value) OVER (
        ORDER BY funnel_step
    ) AS previous_value
FROM funnel_stages

ORDER BY funnel_step;
```

funnel_step INT4	funnel_name TEXT	value INT8	previous_value FLOAT8
1	signups	17623	null
2	ride_requested	12406	0.7039664075356069
3	ride_completed	6233	0.5024181847493149

17. **Using the *“percent of <u>Top</u>”* approach, what are the user-level conversion rates for the following 3 stages of the funnel? 1. signup, 2. ride requested, 3. ride completed**

```
WITH user_ride_status AS (
    SELECT
        user_id,
        MAX(
            CASE
                WHEN dropoff_ts IS NOT NULL
                THEN 1
                ELSE 0
            END
        ) AS ride_completed
    FROM ride_requests
    GROUP BY user_id
),

totals AS (
    SELECT
        COUNT(*) AS total_users_signed_up,
        COUNT(urs.user_id) AS total_users_ride_requested,
        SUM(ride_completed) AS total_users_ride_completed
    FROM signups s
    LEFT JOIN user_ride_status urs ON
        s.user_id = urs.user_iD
),

funnel_stages AS (
    SELECT
        1 AS funnel_step,
        'signups' AS funnel_name,
        total_users_signed_up AS value
    FROM totals

    UNION

    SELECT
        2 AS funnel_step,
        'ride_requested' AS funnel_name,
        total_users_ride_requested AS value
    FROM totals

    UNION

        SELECT
        3 AS funnel_step,
        'ride_completed' AS funnel_name,
        total_users_ride_completed AS value
    FROM totals

)
```

```
SELECT *,
    value::float / FIRST_VALUE(value) OVER (
        ORDER BY funnel_step
    ) AS previous_value
FROM funnel_stages

ORDER BY funnel_step;
```

funnel_step	funnel_name	value	previous_value
INT4	TEXT	INT8	FLOAT8
1	signups	17623	1
2	ride_requested	12406	0.7039664075356069
3	ride_completed	6233	0.353685524598536