



Technical Task

Showcase: A&B Gaming

By Jordi Grau

Content

Introduction.....	3
Monthly Cohort Analysis	4
Accumulative Retentions By Currency	4
% Customers and Revenue by length of stay	5
Customers that leave within a 3-month period.....	5
Average % Retention – Revenue by Product	6
Average % Retention – Revenue by Product & Currency.....	7
Other Charts to understand business	8
Summary	9
Technical Overview	10
Initial data loading steps	10
SQL Queries to generate cohort analysis	11
First sale of every customer view.....	11
Retentions view	11
Monthly Cohort Analysis View.....	12
Month Frequency Analysis	14
Customer Value	15

Introduction

A&B Gaming is a SaaS company that offers **monthly subscriptions** to access their online gaming platform.

They offer 3 different plans: **Small, Medium, and Large for 9.95, 13.95 and 17.95**, respectively.

Users can pay in EUR or USD.

Subscriptions are renewed automatically unless the user cancels them. No refunds are allowed.

We got data from subscribers 2019 and the marketing team is asking us to analyse and understand the behaviour of our customers.

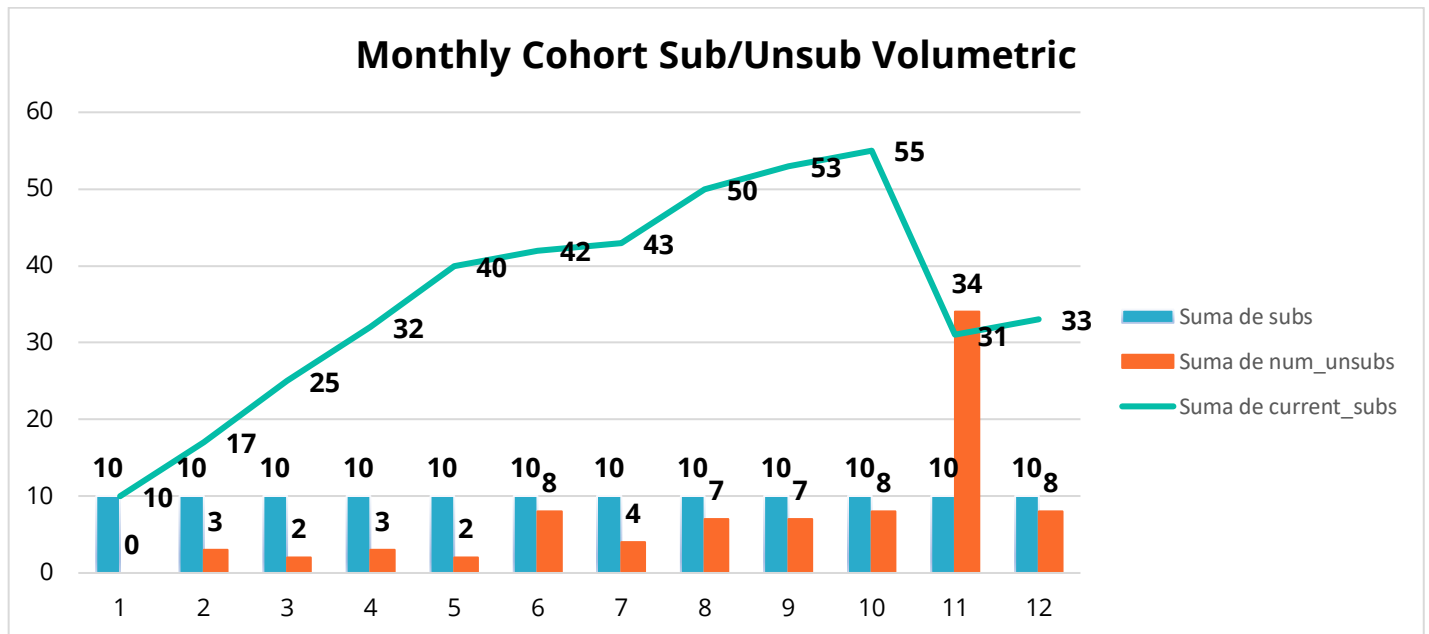
Here is the data structure:

- **Order ID**
- **Account ID**
- **Purchase Date**
- **Product**
- **Amount**
- **Currency**

We need to solve 2 questions here:

- **What products have higher retention?**
- **Differences between users paying in USD vs EUR**

Monthly Cohort Analysis

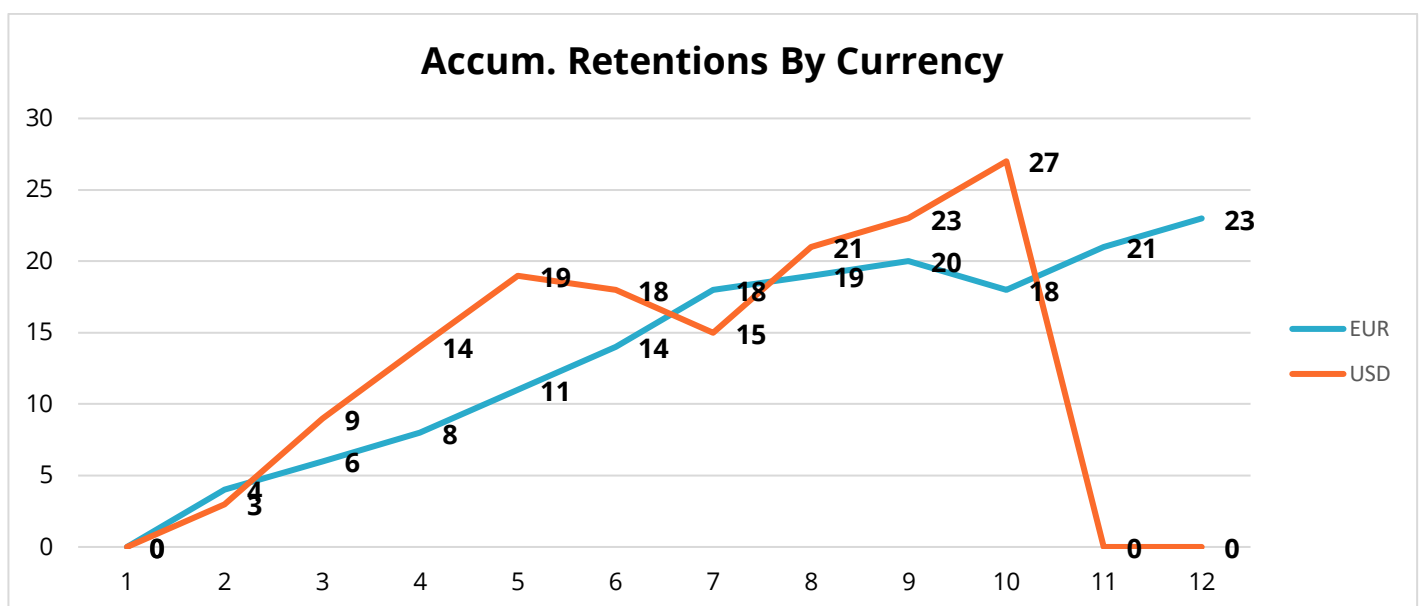


In this chart we have the number of subscribers, unsubscribes, and the current/active users in the A&B Gaming online platform by month.

In the first place, we can see that the number of subscribers is higher than unsubscribes but, in November we see that active users drop down significantly, from 55 to 31.

Accumulative Retentions By Currency

To get further insight on this event, we drilled down at currency level.



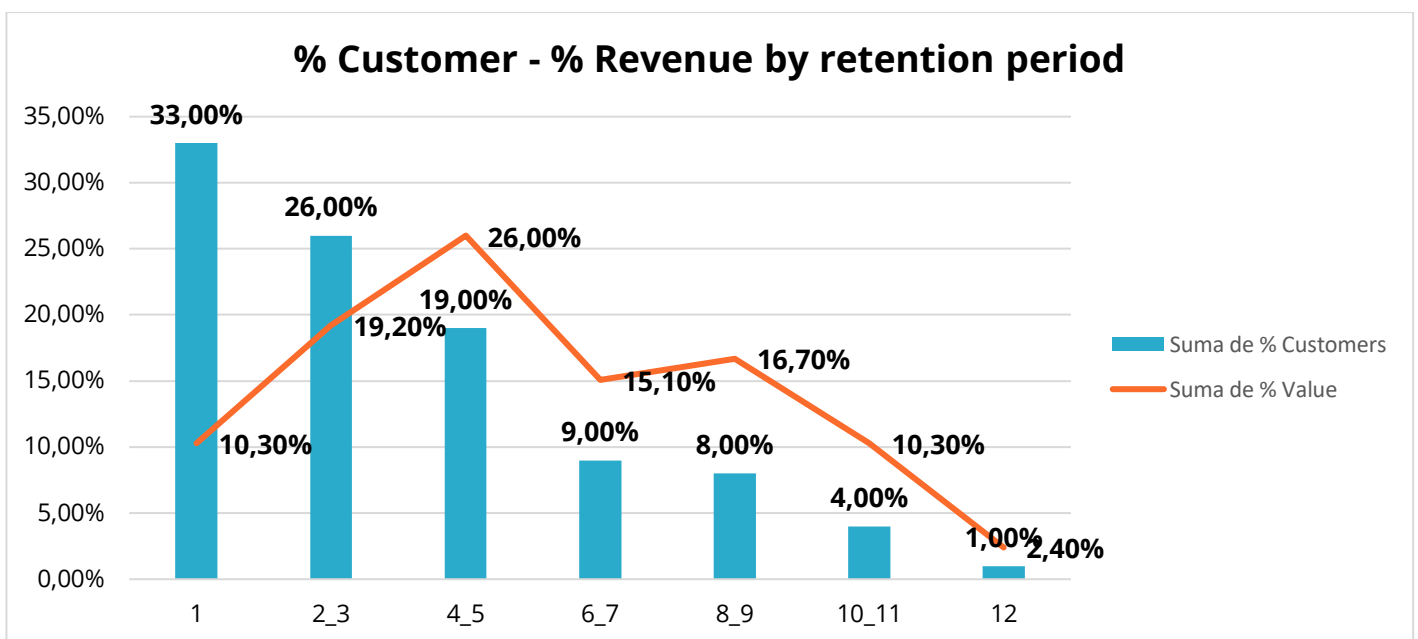
In this chart we can see that the drop in November is **caused by users paying in USD**.

To understand this situation, we would need to get more data for a further deep dive:

- Was any product discontinued? Did the price schema change?
- Satisfaction Survey to those customers that left in November.
- Check competitors new or updated features / products
- Etc.

% Customers and Revenue by length of stay

To understand how this business works and how its customers behave, we build the next chart to see the percentage of customers based on their retention period.

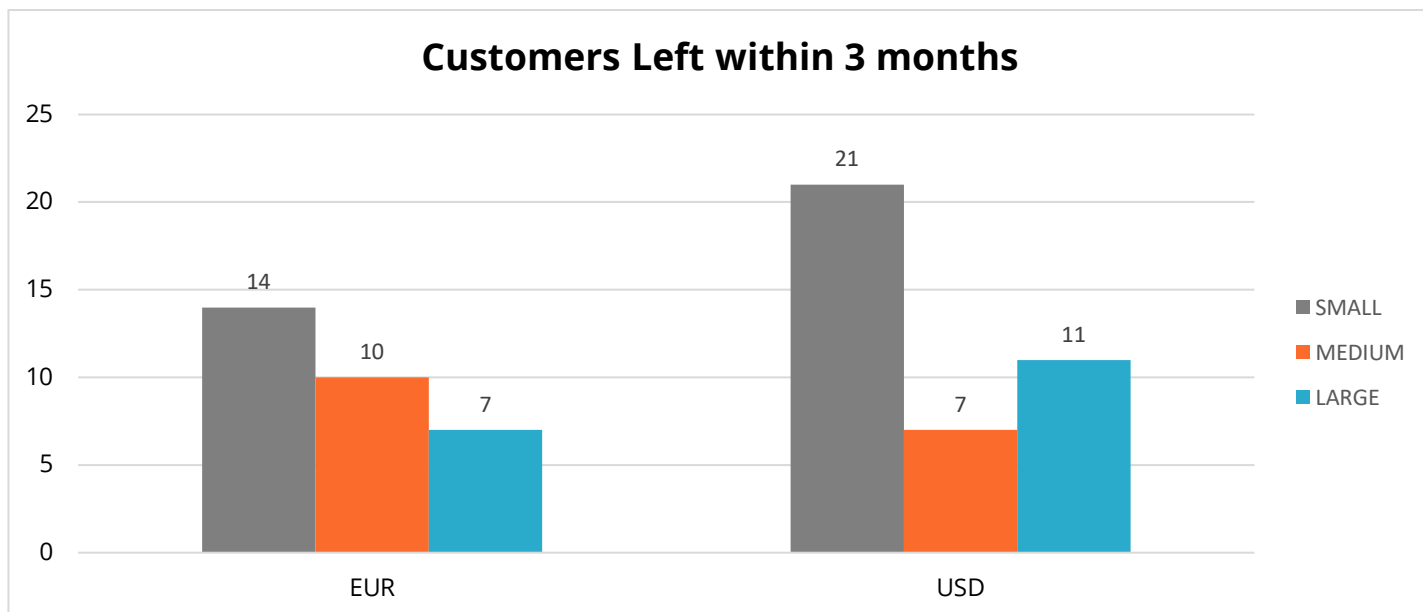


As we can see, we have a challenge with the acquisition efforts as **33% of our new customers don't renew their subscription after their first month, and almost 60% of the total new customers don't last longer than 3 months.**

A good insight from this chart is that **22% of our customer, who have been active from 6 to 12 months represent the 44,50% of our 2019 revenue.** We should analyse these customers to check their **problems and needs** to maybe **build our Buyer Persona** and **adapt our product and communications** to get more retention of our customers.

Customers that leave within a 3-month period

As a next step, we focused on the 1-3 months lifetime to check if there is enough information to change our strategy.

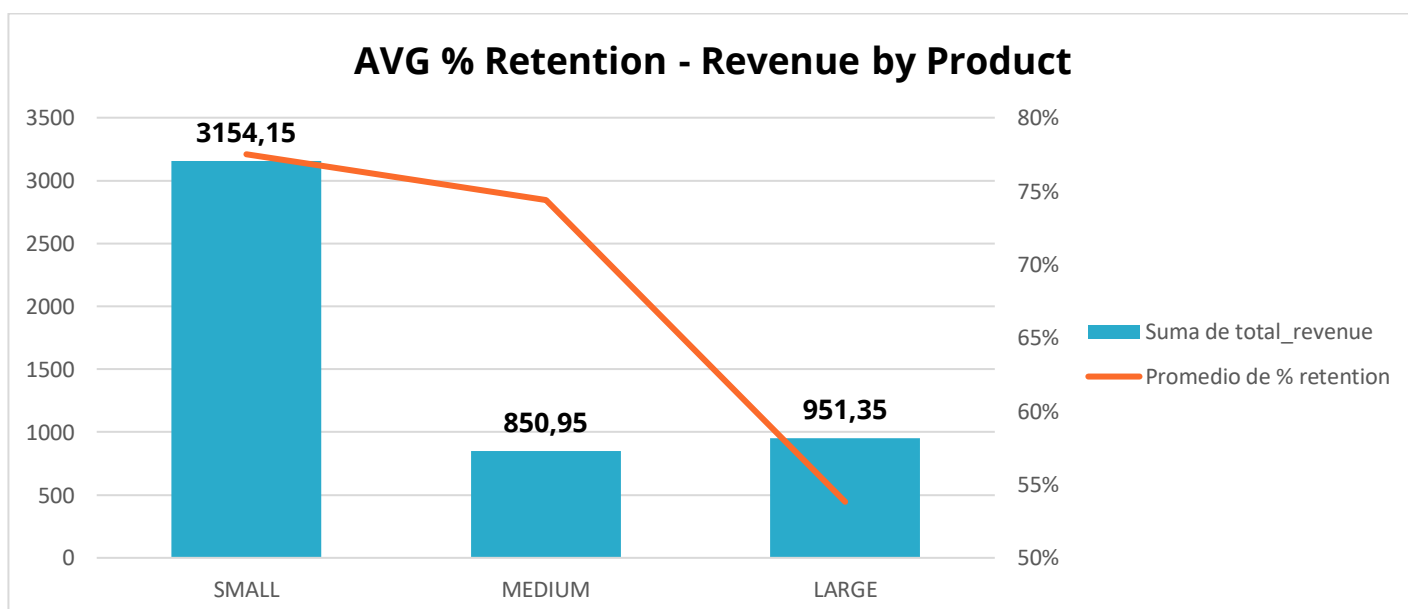


In the first place we can see differences between EUR and USD and differences by product.

Most of the customers who left within the 3-month period were USD paying users with a Small product type of subscription.

Average % Retention – Revenue by Product

By now we can say that USD paying users have less retention but let us check globally how it is.



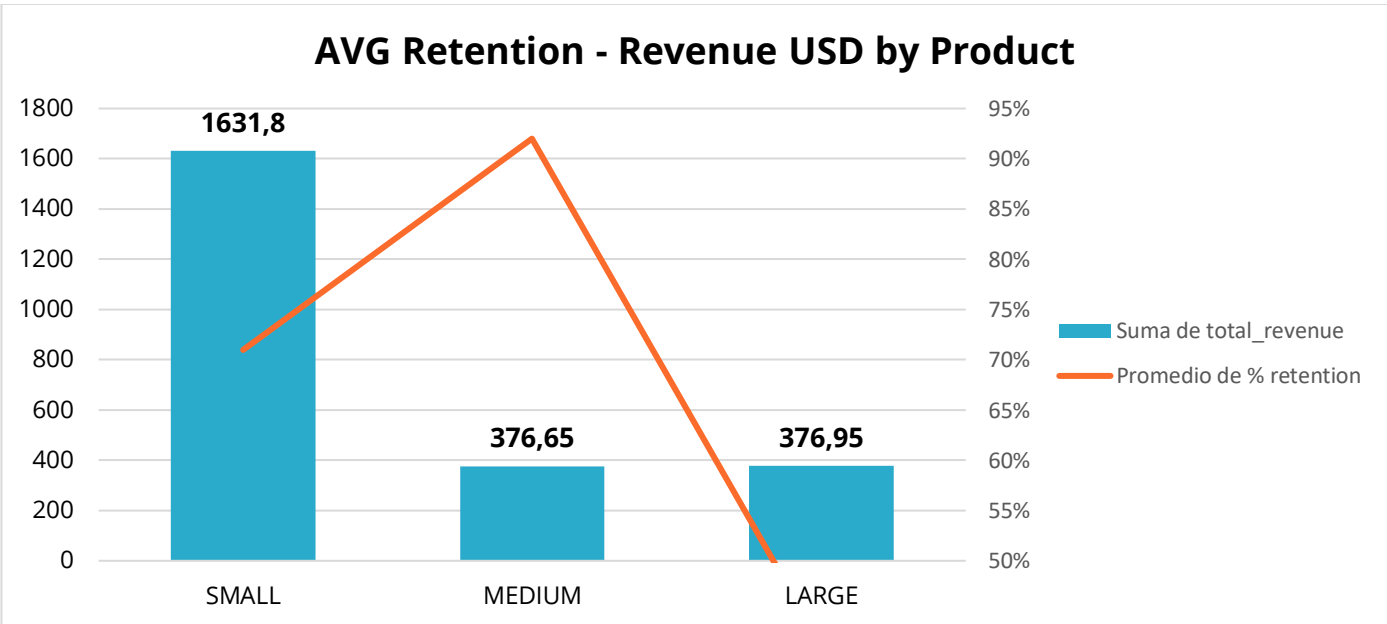
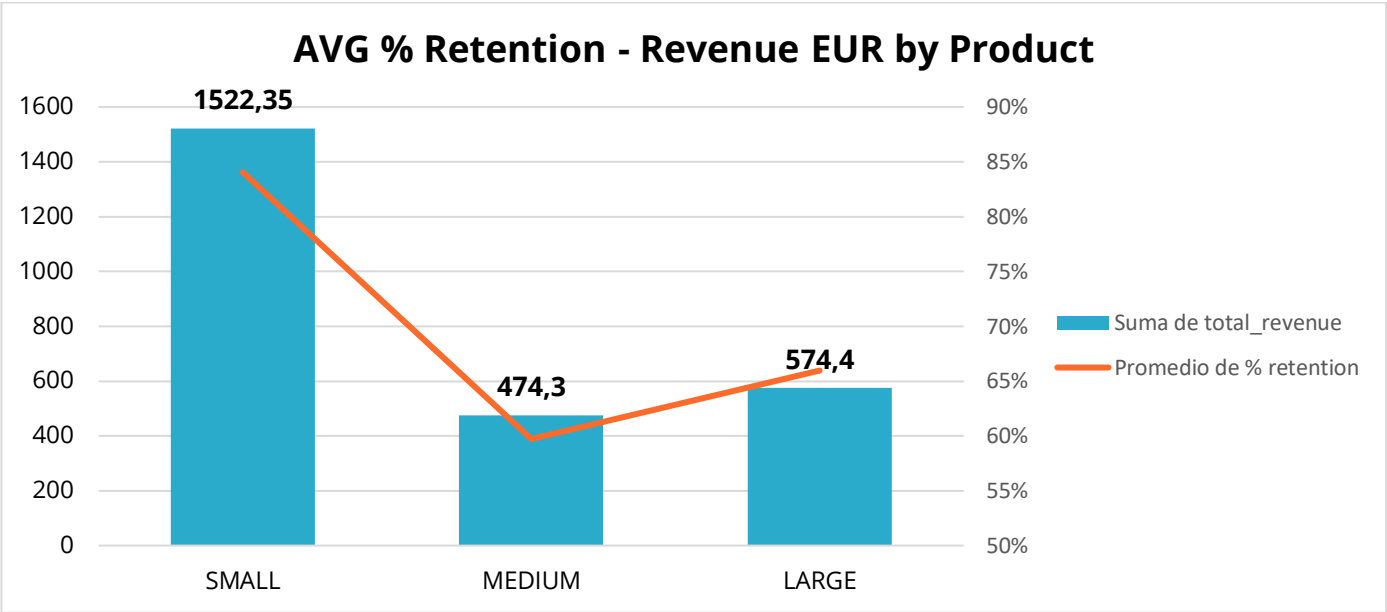
As we can see, **the product with the highest retention (77,5%) is the Small subscription and has the biggest contribution to our revenue. Medium subscription has almost the same**

retention (74,3%) but no as much revenue as the small subscription; this means that there are less customers who bought the medium subscription and were retained for several months.

Large subscription has the lowest retention (53,8%).

Average % Retention – Revenue by Product & Currency

We drilled down by Product and both currencies to see if we spot differences.



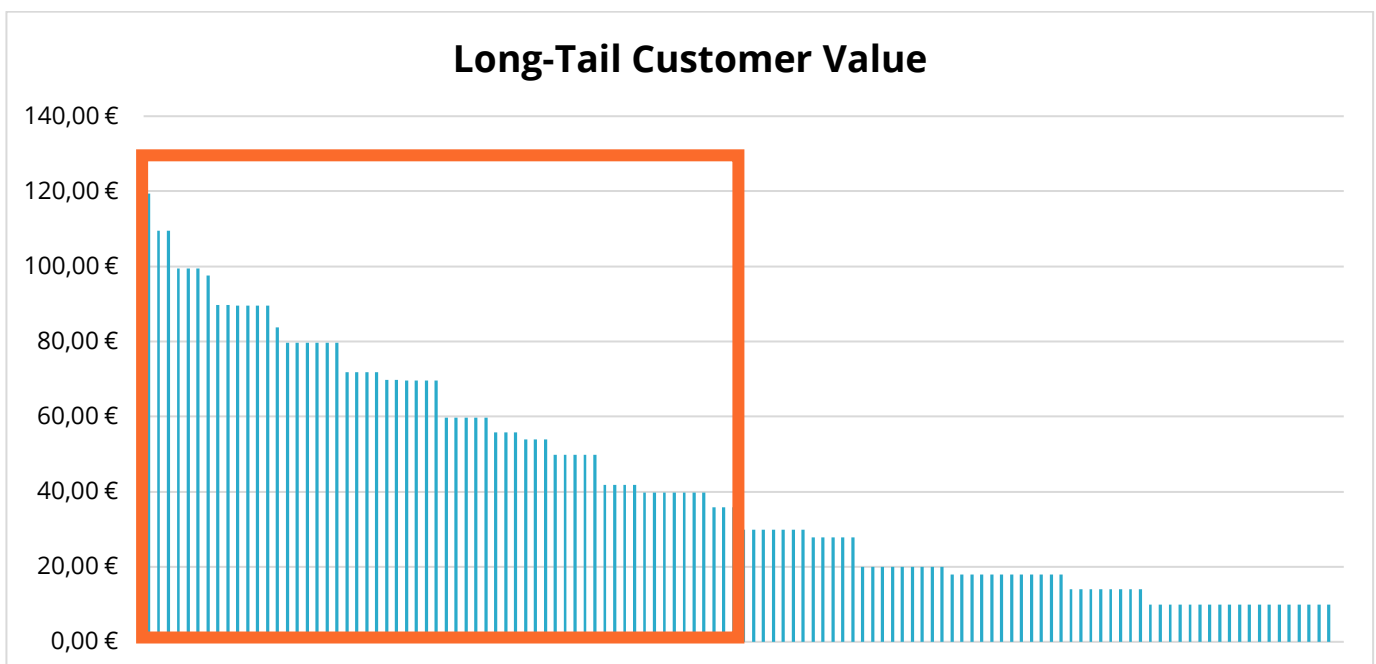
Here we have a huge difference between EUR – USD by the Small subscription. EUR Small has a better retention (84%) than USD Small (70,9%) but it has more sales than EUR.

We know that medium has a good retention too, as we can see in the “**AVG % Retention – Revenue by Product**” chart, now we can see why. **USD users have less revenue but a higher retention (92%) than EUR users (59,7%).**

Finally, we spot **another big difference between EUR – USD**. Users paying in **EUR have more revenue and retention (65,9%)** than users paying in USD, who have **41,6% retention**.

Other Charts to understand business

We have checked the Customer value and charted it to see how the curve is and if it has the long tail.



As we can see, the distribution of our customer's revenue doesn't follow a pronounced curve yet **50% of our customers represent almost 80% of our revenue (78,46%)**. That is a good insight to check differences between that 50% and the other 50% that only has the 20% of the revenue in order to differentiate investment priorities on retention activities.

Summary

- In November there is a major drop of active users from 55 to 31. **Most of the decline is caused by USD paying users.**
- **Customer retention is higher in EUR than USD.**
- 33% of our customers have been active for just a month and 59% only from 1 to 3 months. The product that has been bought the most by these customers is the Small subscription.
- 22% of our customers who have been active from 6 to 12 months represent the 44,5% of our revenue.
- **The product that has the biggest retention is the Small subscription (77,5%) followed by the medium subscription (74,3%) although it is the product with the smallest total revenue.**
- **Users paying in EUR prefer Small and Large subscription and users paying in USD prefer medium subscription.**
- **Users paying in USD represent more revenue than EUR, but they are retained for a shorter period (70,9%).**
- **Users paying in USD tried the large product subscription but have remained a customer for a small period (41,6%) than EUR users (65,9%).**
- Finally, the top 50% of our customers represent the 78,46% of our total revenue.

Technical Overview

Initial data loading steps

We have a csv with data from customers' subscriptions in 2019.

First, we have created an environment to build our database. Then, we have started to build the database and the table which contains the data from the csv.

```
create database qustodio;

use qustodio;

drop table if exists qustodio.csv_import_cohort;

CREATE TABLE csv_import_cohort (
  order_id VARCHAR(30) NOT NULL COLLATE 'utf8_unicode_ci',
  account_id VARCHAR(15) NOT NULL COLLATE 'utf8_unicode_ci',
  purchase_date DATE NOT NULL,
  product VARCHAR(15) NOT NULL COLLATE 'utf8_unicode_ci',
  amount DECIMAL(10,2) NOT NULL,
  currency VARCHAR(50) NOT NULL DEFAULT '' COLLATE
'utf8_unicode_ci',
  PRIMARY KEY (order_id),
  INDEX account_id (account_id),
  INDEX purchase_date (purchase_date),
  INDEX product (product),
  INDEX currency (currency)
)
COLLATE='utf8_unicode_ci'
ENGINE=InnoDB
;
```

The primary key will be always the order_id because that is the variable that differentiates every row.

To build our monthly cohort analysis we need joins by many variables, so these are the variables which we indexed the table:

- **Account_id**
- **Purchase_date**
- **Product**
- **Currency**

To upload the data into the table, we used the import wizard in workbench and assigned every column of csv to every variable in the table.

SQL Queries to generate cohort analysis

First sale of every customer view

First, we created a view that contains the **first sale of every customer to differentiate a sale of a renewal**.

```
DROP view if EXISTS qustodio.v_initial_purchases;

CREATE view qustodio.v_initial_purchases AS
(
    SELECT
        t1.account_id,
        t1.order_id,
        t1.purchase_date,
        t1.product,
        t1.currency
    FROM qustodio.csv_import_cohort AS t1
    JOIN
        (
            SELECT
                account_id,
                MIN(purchase_date) AS first_Date
            FROM qustodio.csv_import_cohort
            GROUP BY account_id
        ) AS t2 ON t1.account_id = t2.account_id AND
        t1.purchase_date = t2.first_Date);
```

We have selected every variable of our first table and we have used a subquery to perform the first purchase_date of every customer to select the first sale of every one of them and joined the by these 2 variables (**account_id** and **purchase_id**).

Retentions view

As we created the **v_initial_purchases view**, we selected all the data without the initial purchases of our customers **with a subquery in a where condition** to create a view with just the renewals of our customers' subscriptions.

```
DROP VIEW if EXISTS qustodio.v_retentions;

create view qustodio.v_retentions AS
(
    SELECT
        MONTH(tt0.purchase_date) AS month_retained,
        tt0.product as product,
        tt0.currency AS currency,
        tt0.order_id AS order_id,
        tt0.account_id as account_id
    FROM qustodio.csv_import_cohort tt0
    WHERE tt0.order_id NOT IN
        (
            SELECT
                t0.order_id
            FROM qustodio.v_initial_purchases t0
        )
    GROUP BY month_retained, product, currency, order_id
);
```

Monthly Cohort Analysis View

We unified both previously views to have sales and retentions in the same structure.

We joined v_initial_purchases with v_retentions by product, month of the purchase date and currency because we need these variables in our view to not just compare subscriptions and unsubscribes but currency and product differences too.

```
DROP VIEW if EXISTS v_monthly_purchases_retentions;

CREATE VIEW v_monthly_purchases_retentions AS
(
SELECT
    MONTH(t1.purchase_date) AS month_num,
    MONTHNAME(purchase_date) AS month_sub,
    t1.product,
    t1.currency,
    COUNT(DISTINCT t1.account_id) AS num_subs,
    COUNT(DISTINCT tt2.order_id) AS num_retentions,
    ((COUNT(DISTINCT t1.account_id))+(COUNT(DISTINCT tt2.order_id)))
AS current_subs
FROM v_initial_purchases t1
LEFT JOIN v_retentions tt2
    ON t1.product = tt2.product
    AND MONTH(t1.purchase_date) = tt2.month_retained
    AND t1.currency = tt2.currency
GROUP BY month_num, MONTHNAME(purchase_date), t1.product, t1.currency
ORDER BY month_num, product DESC
);
```

Here we have the number of sales, the retention by month, product and currency and the current number of active users by these variables.

After we have created this view, we wanted to perform a self-join to have even more details about data from 2019. So, we have created a view of product lookup to check products prices.

```
create v_prod_lookup AS
(
    SELECT
        t1.product AS product,
        t1.amount AS amount
    FROM csv_import_cohort t1
    GROUP BY t1.product
);
```

After we have created this view, we self-joined v_monthly_purchases_retentions to compare variables from current and previous row by month, product and currency.

```
SELECT
    t1.month_num,
    t1.month_sub,
    t1.currency,
    t1.product,
    (t1.num_subs) as subs,
    t1.num_subs*t3.amount AS subs_value,
    IFNULL((case
        when (t2.current_subs)!=t1.num_retentions) then
            (t2.current_subs-t1.num_retentions)
        ELSE null END), 0)AS num_unsubs,
    IFNULL((case
        when (t2.current_subs)!=t1.num_retentions) then
            (t2.current_subs-t1.num_retentions)
        ELSE null END )*(t3.amount), 0) AS value_unsubs,
    t1.num_retentions,
    t1.current_subs,
    (t1.num_retentions/IFNULL(t2.current_subs, 1)) AS '% retention'
FROM v_monthly_purchases_retentions t1
left outer JOIN v_monthly_purchases_retentions t2
    ON t1.product = t2.product
    AND t1.currency = t2.currency
    AND t1.month_num = t2.month_num+1
JOIN v_prod_lookup t3 ON t1.product = t3.product
GROUP BY t1.month_num, t1.month_sub, t1.product, t1.currency
ORDER BY t1.product DESC, t1.currency, t1.month_num;
```

First, we have self-joined by product currency and month_num but plus 1, so we get the previous variable record in the current row and compare them. So, with this self-join we can compare number of retentions of the main view with the current subscribers from the second view (previous row) to get the number of unsubscribes and the amount they represent to us.

We have done it get the % retention of every month by product and currency too.

With this select we have built our monthly cohort chart.

Month Frequency Analysis

We have analysed our customers by the number of months they have been active to spot differences at our customers distribution.

```
SELECT
    (case
        when freq=12 then '12'
        when freq>=10 and freq<12 then '10_11'
        when freq>=8 and freq<10 then '8_9'
        when freq>=6 and freq<8 then '6_7'
        when freq>=4 and freq<6 then '4_5'
        when freq>=2 and freq<4 then '2_3'
        when freq= 1 then '1' END) AS Frequency_Band,
    count(distinct t1.account_id) AS Sum_Customers,
    ROUND((count(distinct t1.account_id))/(SELECT COUNT(DISTINCT
        tt2.account_id) FROM csv_import_cohort tt2), 2)
        AS '% Customers',
    ROUND((SUM(t1.amount)/COUNT(t1.order_id)), 2)
        AS Average_Purchase_Value,
    ROUND((COUNT(t1.order_id)/COUNT(DISTINCT t1.account_id)), 2)
        AS Average_Purchase_Frequency_Rate,
    ROUND((SUM(t1.amount)/COUNT(DISTINCT t1.account_id)), 2)
        AS Average_Revenue_x_User,
    SUM(t1.amount) AS total_value,
    ROUND(((SUM(t1.amount))/(SELECT SUM(tt3.amount) FROM
        csv_import_cohort tt3)), 3) AS '% Value'
    from csv_import_cohort t1
    INNER JOIN
        (
            SELECT t1.account_id, COUNT(t1.account_id) AS freq
            FROM csv_import_cohort t1
            GROUP BY t1.account_id
        ) t3 ON t1.account_id = t3.account_id
    GROUP BY (case
        when freq=12 then '12'
        when freq>=10 and freq<12 then '10_11'
        when freq>=8 and freq<10 then '8_9'
        when freq>=6 and freq<8 then '6_7'
        when freq>=4 and freq<6 then '4_5'
        when freq>=2 and freq<4 then '2_3'
        when freq= 1 then '1' END);
```

Here we have the main table we have created at first to get the number of months every customer have been active and we called it 'Freq' (subquery) and grouped by it, so we have here the number and % of customers and revenue and % revenue for each frequency.

As we have seen the chart of our month frequency, we wanted to analyse customers who have left within the first 3 months. So, we set a where condition of 1, 2 and 3 frequency months.

```
SELECT
    t3.freq AS Frequency,
    t1.product,
    t1.currency AS currency,
    t1.account_id,
    t1.amount
from csv_import_cohort t1
INNER JOIN
    (
        SELECT t2.account_id, COUNT(t2.account_id) AS freq
        FROM csv_import_cohort t2
        GROUP BY t2.account_id
    ) t3 ON t1.account_id = t3.account_id
WHERE t3.freq IN (1,2,3)
GROUP BY t3.freq, t1.product, t1.currency, t1.account_id;
```

Customer Value

To perform the Customers' Value Long-Tail, we have selected the account_id and the total amount they spent ordered by the same total amount.

```
SELECT
    t1.account_id,
    SUM(t1.amount) AS total_value
FROM csv_import_cohort t1
GROUP BY t1.account_id
ORDER BY SUM(t1.amount) desc;
```