BambaSwap KPI’s Jan to Oct 2024

1. **Customer Acquisition Rate**

Customer Acquisition Rate (**CAR**) measure how many **new** customers a BambaSwap has gained over a period relative to its **existing customer base**. In our case, we calculate CAR **month by month** from **January to October** using historical data.

**Step-by-Step Breakdown of the CAR Calculation**

**Identify Returning Customers**

* We define "returning customers" as those who have appeared at least **3 times before the current month** (January to September).
* This ensures we only consider customers who have shown **consistent engagement**.
* This filters out casual or one-time customers who aren't truly "returning."

**Identify New Customers Per Month**

* A **new customer in month m** is a customer who appears in m **but was NOT a returning customer**.
* If a customer in ‘m’ doesn’t exist in ‘returning\_customers’, they are **new**.

**Compute CAR for Each Month**

* The standard CAR formula is:

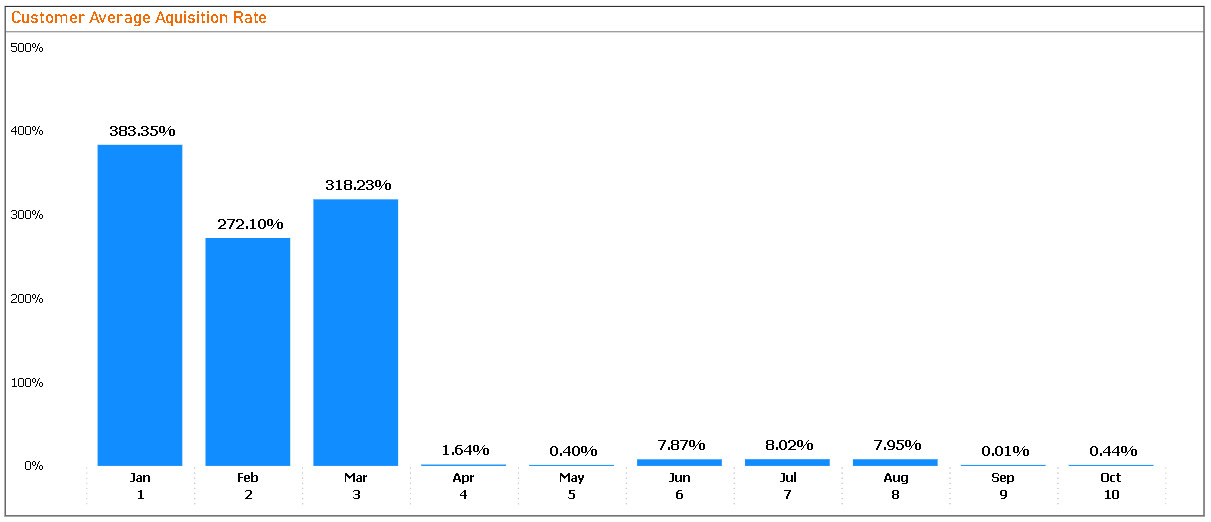
* **Numerator:** COUNT(n.phone\_number) → Total new customers in the month.
* **Denominator:** COUNT(r.phone\_number) → Total returning customers before the month.  
  **Avoiding NULLs:**
  + GREATEST(COUNT(r.phone\_number), 1): Ensures denominator is **never zero**.
  + COALESCE(..., 0): Ensures final CAR is **never NULL**, replacing NULLs with.

**Sorting the Results by Month**

* Since we already have a month\_number column, we **order by month** so that results appear in a proper timeline:
* Ensures CAR is presented **chronologically** from January to October.

**Summary**

| **Step** | **Purpose** |
| --- | --- |
| **1️. Identify Returning Customers** | Filters out one-time customers & defines the base for comparison. |
| **2️. Identify New Customers Per Month** | Finds customers appearing for the first time in that month. |
| **3️. Compute CAR** | Uses the CAR formula and ensures no NULL values. |
| **4️. Sort by Month** | Displays the results in the correct order. |



1. **Customer Conversion Rate**

Customer Conversion Rate (CCR) measures how many BambaSwap existing customers (who have used the service before) return in a given month. It helps BambaSwap understand customer retention—how well Bambaswap keeps their customers engaged over time.

**How is CCR Different from CAR?**

* **Customer Acquisition Rate (CAR)** measures how many new customers are added in a month.
* **Customer Conversion Rate (CCR)** measures how many of those **who were customers in previous months** come back in the current month.

**How I have CCR Calculated in this SQL Query?**

**Step-by-Step Breakdown of the Query**

We calculate CCR by:

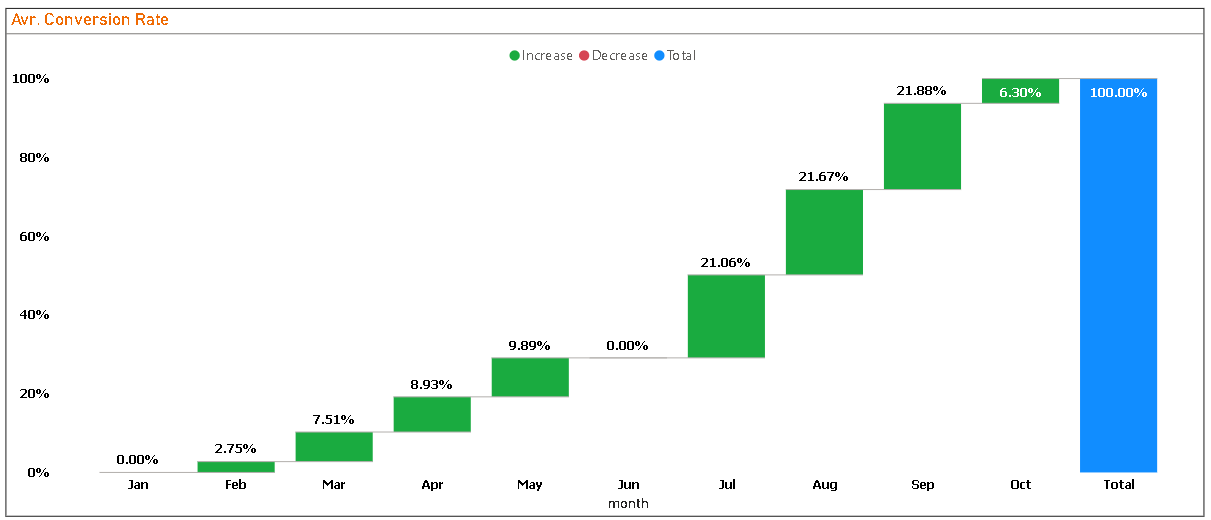
1. **Identifying all BambaSwap unique customers per month.**
   * This is stored in monthly\_customers, which tells us which customers appeared in which month.
2. **Finding customers who appeared in any previous month before month ‘m’.**
   * In previous\_customers, we count how many times a customer appeared in months **before** m.
   * If past\_occurrences > 0, it means the customer **was seen in a previous month**.
3. **Identifying returning customers.**
   * The returning\_customers table filters out customers who **reappear after being seen in any previous month**.
4. **Computing CCR using the formula:**
   * **Numerator:** COUNT(rc.phone\_number) → The number of returning customers.
   * **Denominator:** COUNT(pc.phone\_number) → The total number of customers who were seen before m.
   * GREATEST(COUNT(pc.phone\_number), 1): Ensures we **never divide by zero** to avoid errors.
   * COALESCE(..., 0): Ensures if there are **no returning customers, CCR is 0% instead of NULL**.

**What Insights Does This CCR Provide?**

* **A High CCR (Near 100%)**  
  → **Great retention!** Most customers who have used the service before are coming back.
* **A Low CCR (Close to 0%)**  
  → **Poor retention!** Customers are **not** returning after their first experience.
* **Fluctuating CCR Trends**  
  → Could indicate **seasonal customer behavior** or an issue with **customer satisfaction**.

**Summary**

| **Step** | **Purpose** |
| --- | --- |
| **Identify unique customers per month (monthly\_customers)** | Get a list of all customers in each month. |
| **Find past occurrences (previous\_customers)** | Identify customers who appeared before each month. |
| **Find returning customers (returning\_customers)** | Extract customers who returned after appearing in previous months. |
| **Compute CCR** | Measure how many previous customers returned in the current month. |



**What is Monthly Recurring Revenue (MRR)**

**Monthly Recurring Revenue (MRR)** is a key financial metric that measures the predictable and recurring revenue BambaSwap earns each month from **returning customers**.

* It excludes **one-time payments** and focuses only on **customers who continue to pay over time**.
* **For our dataset**, MRR tracks revenue from **customers who have paid before (in any previous month) and are paying again.**

**How MRR is Different from Total Revenue?**

| **Metric** | **What It Tracks** |
| --- | --- |
| **Total Revenue** | All payments made in a given month, including first-time customers. |
| **MRR** | Only revenue from **returning customers** (customers who have paid before). |

**Why This Matters:**  
MRR is crucial for understanding **customer retention and revenue stability**. If MRR is growing, it means the BambaSwap business is retaining and monetizing existing customers well.

To calculate **MRR**, the query follows these logical steps:

**Formula for MRR**

The general formula for MRR is:

Where:

* **Recurring Customers** = Customers who have paid in **any previous month** and return in month m.
* **Revenue** = The sum of bs\_revenue from all returning customers in month m.

**Step 1️: Identify Monthly Revenue**

* The **monthly\_revenue** CTE groups transactions by phone\_number and month\_number, summing up bs\_revenue for each customer per month.
* This gives **total revenue per customer per month**.

**Why?** This allows us to calculate revenue per customer for each month.

**Step 2️: Identify Customers Who Paid in Any Previous Month**

* The **previous\_customers** CTE checks if a customer has made **at least one past payment before the current month**.
* If they have paid **before**, they are considered **recurring customers**.

**Why?**

* This ensures that we **don’t count new customers**, only those who have paid before.
* If a customer paid in **March and April**, their **April revenue counts as MRR**, but their **March revenue doesn’t** (because it was their first payment).

**Step 3️: Handle Skipped Months (Fixing the 0 MRR Issue)**

* The **recurring\_customers** CTE also includes customers who paid **in the future** (in case they skipped a month).
* Example:
  + A customer paid in **January**, skipped **February and March**, then paid again in **April**.
  + **Without this fix, April’s MRR would have been 0**, but this step ensures they are counted as **recurring revenue**.

**Why?**

* This ensures that we **capture all recurring customers**, even if they **skipped months** before coming back.

**Step 4️: Compute MRR Per Month**

* The final SELECT statement calculates **total revenue from recurring customers per month**.

**Why?**

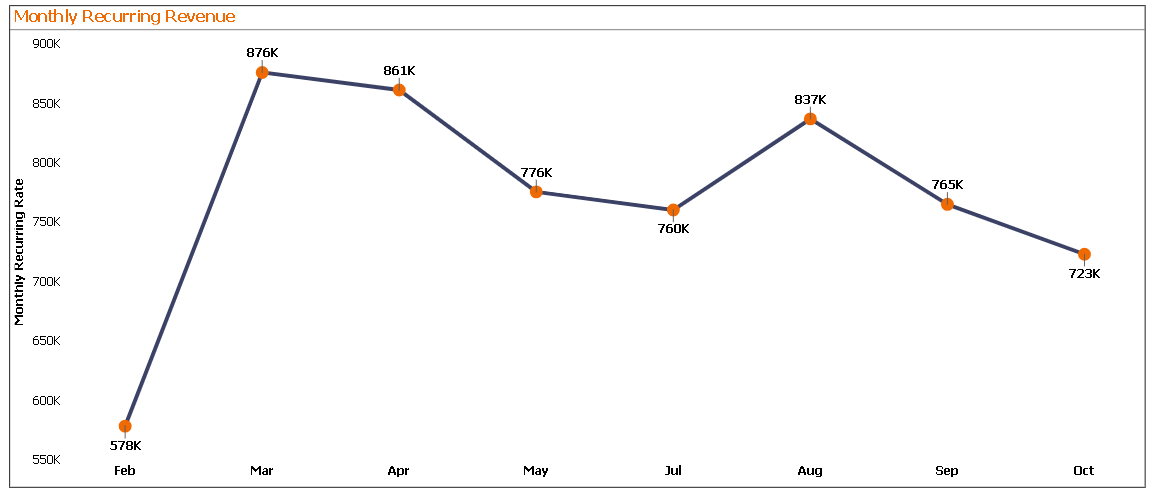
* It **only sums revenue from returning customers** (not new ones).
* **COALESCE(..., 0)** ensures that if no recurring customers exist in a month, MRR is **0 instead of NULL`**.

**What Insights Can We Get From This MRR Calculation?**

**If MRR is growing** → More customers are **returning and paying again** → **Good retention & revenue stability** ✅  
**If MRR is declining** → Fewer customers are **returning** → **Possible churn issue** ⚠️  
**If MRR fluctuates** → Indicates **seasonality** or inconsistent customer retention trends 📊  
**If MRR is 0 in a month with transactions** → That month only had **new customers** (not returning ones).

| **Step** | **Purpose** |
| --- | --- |
| **Calculate Total Revenue per Customer per Month (monthly\_revenue)** | Get revenue for each customer per month. |
| **Identify Customers Who Paid Before (previous\_customers)** | Find all customers who have paid in the past. |
| **Handle Skipped-Month Customers (recurring\_customers)** | Ensure customers who skipped months but returned are counted. |
| **Compute Monthly Recurring Revenue (MRR)** | Sum revenue for recurring customers per month. |

**Summary of the MRR Calculation**



**What is Monthly Retention Rate?**

**Monthly Retention Rate (MRR)** is a key customer success metric that measures the **percentage of BambaSwap existing customers who continue paying month after month**.

* It tells us **how well the bambaswap retains customers over time**.
* A high retention rate indicates **strong customer loyalty**, while a low rate suggests **customer churn** (customers are leaving).

2️. Monthly Retention Rate Formula

Where:

* **Returning Customers in Month m** → Bambaswap Customers who paid in m-1 and continue paying in m.
* **Total Customers in Month m-1** → All unique customers who paid in the previous month.

**Step 1️: Identify All Paying Customers Per Month**

* The monthly\_customers CTE extracts all **distinct** customers who made a payment or transacted in each month both in conversions and buying airtime.
* This ensures we **only count active customers** who transacted.

**Why?**

* This acts as our **base dataset** for counting **total customers per month**.

**Step 2️: Count Total Customers in Each Month**

* The previous\_month\_customers CTE counts **how many unique customers transacted in each month**.
* This helps establish a **baseline number of customers for retention calculations**.

**Why?**

* This gives us **a reference for how many customers existed in the previous month**.
* The denominator for the **Retention Rate formula** comes from this dataset.

**Step 3️: Count Customers Who Returned in the Next Month**

* The returning\_customers CTE checks how many **customers from m-1 continued paying in m**.
* This helps identify **loyal returning customers**.

**Why?**

* This step **tracks retention month over month**.
* **Uses LEFT JOIN** to ensure all customers from m-1 are checked for reappearance in m.
* The numerator for the **Retention Rate formula** comes from this dataset.

**Step 4️: Calculate Monthly Retention Rate**

* The final SELECT statement computes the **percentage of customers from m-1 who continued paying in m**.

**Why?**

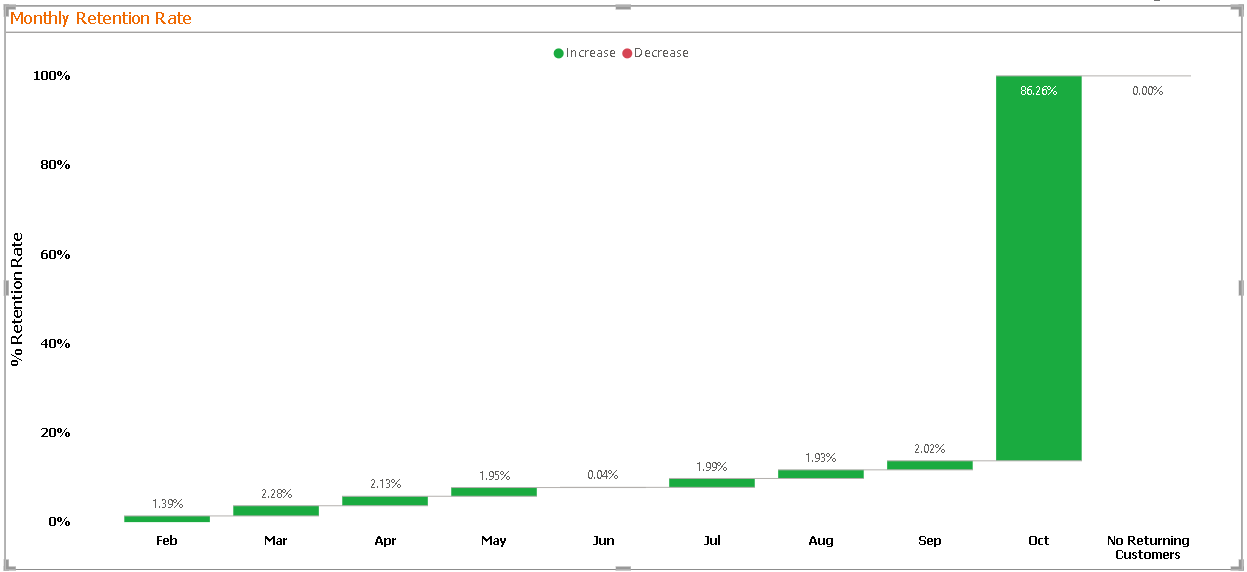
* **Retention rate formula applied** → (returning customers / total previous month customers) \* 100.
* **COALESCE(rc.current\_month, 'No Returning Customers')** → Handles cases where **no customers returned** (avoids NULL values).
* **GREATEST(pm.total\_previous\_customers, 1)** → Prevents division by zero errors.
* **LEFT JOIN ensures months with no returning customers still appear** instead of being skipped.

**What Insights Can We Get From This Query?**

**If Retention Rate is high** → Customers **continue using the service**, indicating strong loyalty.  
**If Retention Rate is dropping** → **Churn rate is increasing**, meaning customers are **leaving**.  
**If Retention Rate is 0% for a month** → No customers from m-1 returned in m.  
**Seasonal Retention Trends** → If retention fluctuates, it could be due to **seasonal demand or product changes**.

**Summary of the Retention Rate Calculation**

| **Step** | **Purpose** |
| --- | --- |
| **Extract Monthly Customers (monthly\_customers)** | Identify all paying customers each month. |
| **Count Total Customers Per Month (previous\_month\_customers)** | Establish the baseline for retention calculations. |
| **Count Returning Customers (returning\_customers)** | Identify customers who continue paying. |
| **Compute Retention Rate** | Apply formula and ensure missing data is handled properly. |



**What is Churn Rate?**

**Churn Rate** measures the percentage of customers who **stop paying** from one month to the next. It is the opposite of **Retention Rate**.

* If **Retention Rate** is **high**, Churn Rate is **low** → Business is retaining customers well.
* If **Churn Rate** is **high**, it means many customers are **leaving** the service.

Formula for Churn Rate

Where:

* **Churned Customers in Month m** → Customers who paid in m-1 but did **not** pay in m.
* **Total Customers in Month m-1** → All unique customers who paid in m-1.

**Step 1️: Identify All Paying Customers Per Month**

* monthly\_customers extracts **distinct** customers who made a payment (bs\_revenue > 0) in each month.
* This ensures we **only count active customers** who have generated revenue.

**Why?**

* This dataset is used to count **total customers per month**.

**Step 2️: Count Total Customers Per Month (previous\_month\_customers)**

* This step counts **how many unique customers paid in each month**.
* This acts as the **baseline for churn calculations**.

**Why?**

* This gives us the **total number of customers before the churn calculation**.

**Step 3️: Count Returning Customers (returning\_customers)**

* This step **counts how many customers from m-1 continued paying in m**.

**Why?**

* This dataset is used to calculate **how many customers stayed**.
* The **number of customers who churned** will be calculated by **subtracting this count from the total previous customers**.

**Step 4️: Compute Monthly Churn Rate**

* **Final calculation:**

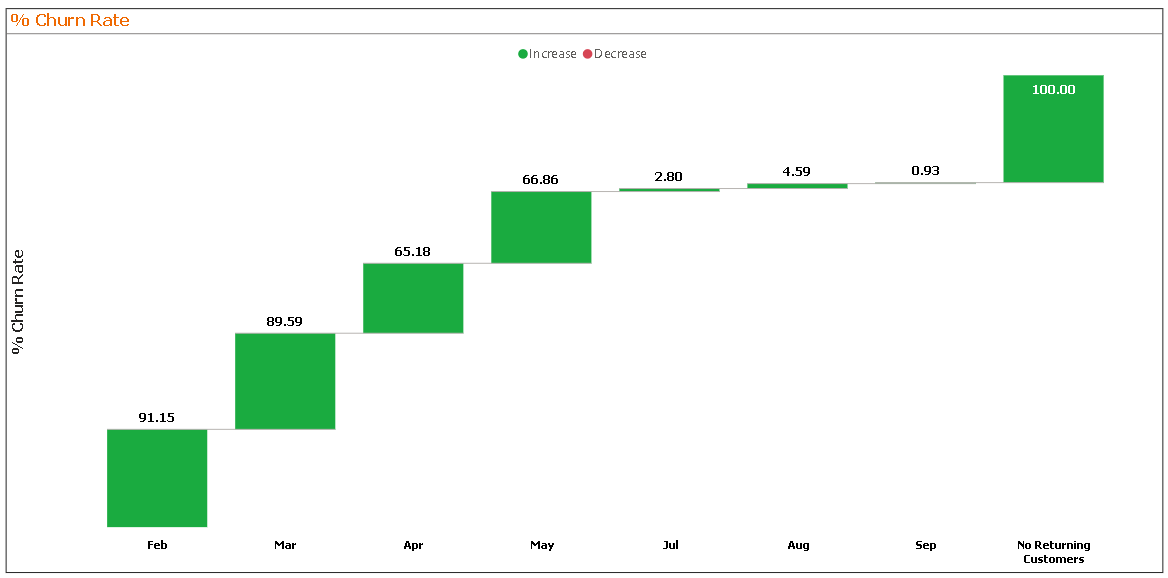
**Why?**

* **Subtracts returning customers from total previous customers** to get the **number of churned customers**.
* **Uses GREATEST(..., 1)** to avoid division by zero errors.
* **COALESCE(..., 0)** ensures that if there are no churned customers, the churn rate is 0% instead of NULL.

**What Insights Can We Get?**

**If Churn Rate is high** → Many customers are leaving → **Possible retention issues** ⚠️  
**If Churn Rate is low** → Customers are **staying & continuing to pay** ✅  
**If Churn Rate fluctuates** → Could indicate **seasonal trends or business model changes** 📊  
**If Churn Rate is 0% for a month** → No customers from m-1 left (great retention) 🔥

| **Step** | **Purpose** |
| --- | --- |
| **Extract Monthly Customers (monthly\_customers)** | Identify all paying customers each month. |
| **Count Total Customers Per Month (previous\_month\_customers)** | Establish the baseline for churn calculations. |
| **Count Returning Customers (returning\_customers)** | Identify customers who continue paying. |
| **Compute Churn Rate** | Apply formula and ensure missing data is handled properly. |



**Customer Lifetime Value (CLV) Calculation from bambaswap\_combined\_jan\_oct\_2024**

**1️What is Customer Lifetime Value (CLV)?**

**Customer Lifetime Value (CLV)** estimates the total revenue a business can expect from a **single customer** over their entire relationship with the company.

* **Higher CLV** means customers stay **longer and spend more**.
* **Lower CLV** suggests **short customer relationships or low spending**.

Formula for CLV Per Month

Where:

**ARPU (Average Revenue Per User)** =

**Churn Rate (Churn\_Rate\_m)** (as calculated earlier)

**If Churn Rate is 0, we set CLV to NULL or a high fixed value (e.g., 12 months)** to avoid division by zero.

**Explanation**

**Step 1️: Calculate Monthly Revenue & ARPU (monthly\_revenue)**

* **Find total revenue (SUM(bs\_revenue)) per month**
* **Count distinct customers to get ARPU**

ARPUm =

**Step 2️: Calculate Monthly Churn Rate (churn\_data)**

* Uses **same logic as the churn rate query**

\text{Churn Rate}\_m = \frac{\text{Total Customers in m-1} - \text{Returning Customers in m}}{\text{Total Customers in m-1}} ]

* If churn rate is **0**, CLV will be set to NULL to avoid division by zero.

Step 3️: Compute CLV Per Month

**Uses CASE WHEN churn\_rate = 0 THEN NULL** to avoid division errors.

**Ensures meaningful CLV values even in months with perfect retention**.

**What Insights Can We Get?**

1️ **If CLV is high** → Customers stay for **longer periods** and spend more. ✅  
2️ **If CLV is low** → Customers **leave quickly**, leading to **poor profitability**. ⚠️  
3️ **If CLV fluctuates** → **Seasonal changes or pricing changes** affecting customer lifetime. 📊

**Summary of CLV Calculation**

| **Step** | **Purpose** |
| --- | --- |
| **Extract Monthly Revenue (monthly\_revenue)** | Calculate total revenue & ARPU per month. |
| **Calculate Churn Rate (churn\_data)** | Compute how many customers stop paying each month. |
| **Compute CLV per Month** | Uses ARPU and churn rate to estimate customer lifetime value. |

**Key Takeaways**

✅ **CLV helps measure long-term customer value**.  
✅ **Combining CLV with churn rate & MRR gives a full picture of revenue stability**.  
✅ **Used for marketing budgets, customer acquisition strategies & financial planning**.

