#### 

#### 

#### **Mobil Game**

Group No: 5

Şamil Özpınar , 2019205159

Onur Ayaz, 2020100048

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### **1.Description of the Project**

#### **Brief description:** This project involves analyzing game data for user behavior, engagement, and purchasing patterns within a mobile game.

#### **Dataset**: The dataset consists of multiple tables, including install, session, revenue, cost, level, level\_attempt, and player\_action, which capture key user interactions, in-game progress, and costs associated with advertising.

#### **Reason for choosing the project:** We enjoy playing mobile games, and this project allows us to explore how different game dynamics influence player behavior and in-app purchases.

### 

### 

### 

### 

### 

### **2. About the Dataset**

#### **2.1 Schema of the Dataset**

The dataset is a simulated relational database representing user behavior and monetization in a mobile game.We created our own synthetic data. It includes 500 users from 5 different countries and 4 networks, covering a 30-day gameplay period. Segmentation into 4 different engagement levels helps classify users by their level of interaction with the game. This will allow us to perform meaningful analysis.

It includes the following seven CSV tables.

* Install: Contains user installation data (e.g., user ID, platform,country,network).
* Session: Logs user sessions, including session start and end times, user actions, etc.
* Revenue: Captures revenue data, including purchases, ad revenue, and in-app purchases.
* Ua\_cost: Contains data related to user acquisition cost by campaign.
* Level: Defines the structure of the game, including levels, difficulty, and player progress.
* Level\_attempt: Logs user attempts and completions for each level.
* Player\_action: Captures detailed user actions such as level completion, purchase behavior.

#### 

#### **2.2 Datatypes of the Dataset**

#### Each table includes specific columns with appropriate data types, primarily suited for SQL Server.

#### **install**

* user\_id: Integer
* platform, network, country: nvarchar(50)
* event\_time: date

#### **revenue**

* revenue\_id, user\_id: Integer
* revenue: Float
* package\_type: nvarchar(50)
* event\_time: date

#### **ua\_cost**

* cost\_id, user\_id: Integer
* ua\_cost: Float
* country,campaign\_type,network: nvarchar(50)
* event\_time: date

#### **level**

* level\_id: Integer
* difficulty\_rate: Float

#### **level\_attempt**

* attempt\_id, user\_id, session\_id, level\_id, attempt\_number: Integer
* status: nvarchar(50)
* event\_time: date

#### **player\_action**

* user\_id, attempt\_id: Integer
* action\_type: varchar(50)
* event\_time: date

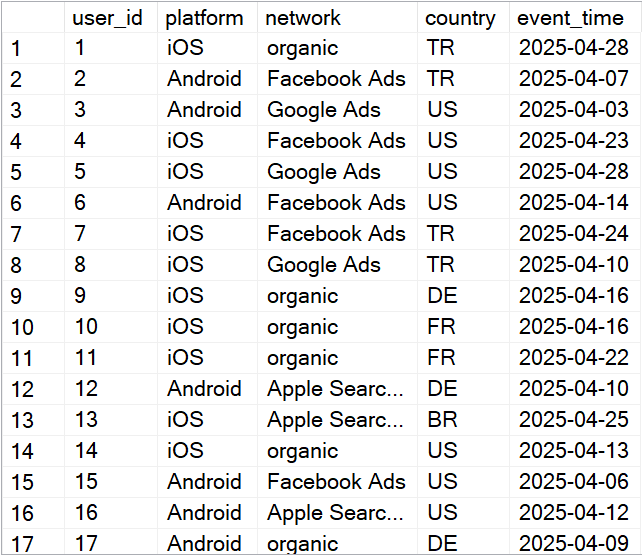
#### **session**

* session\_id, user\_id: Integer
* time\_spent: Integer
* event\_time, session\_day: datatime2(7)

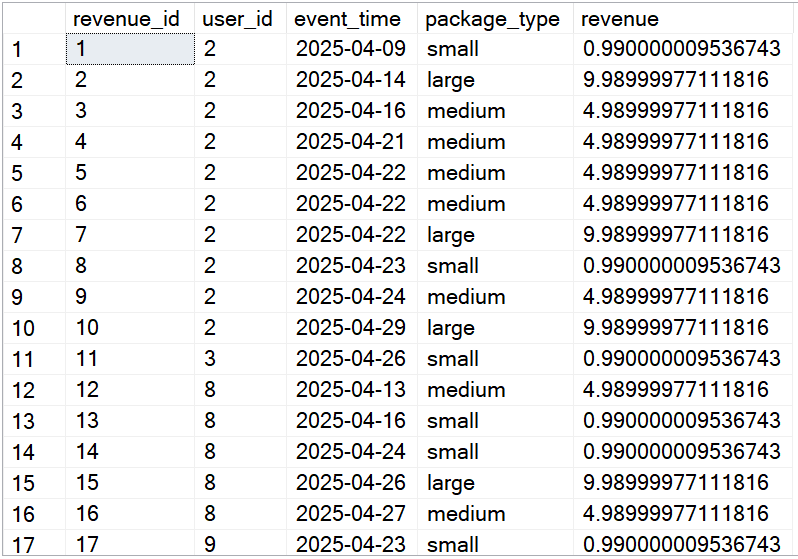
**2.3 Description of the Dataset**

Each table is described below with first 5 rows:

* install (500 rows)



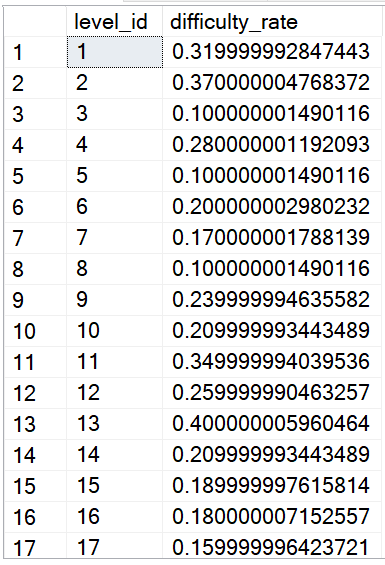
* revenue (874 rows)



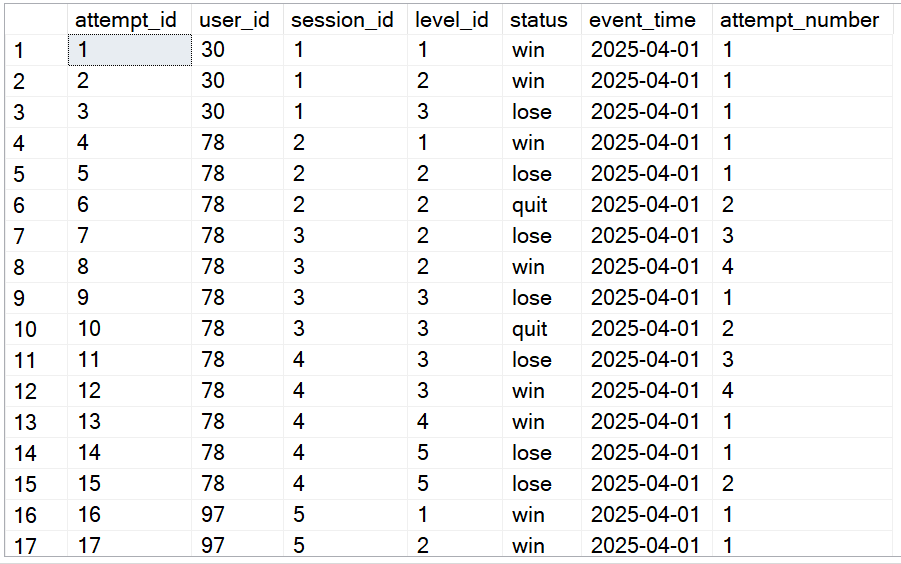
* ua\_cost (500 rows)



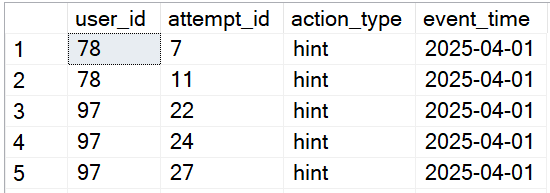
* level (200 rows)



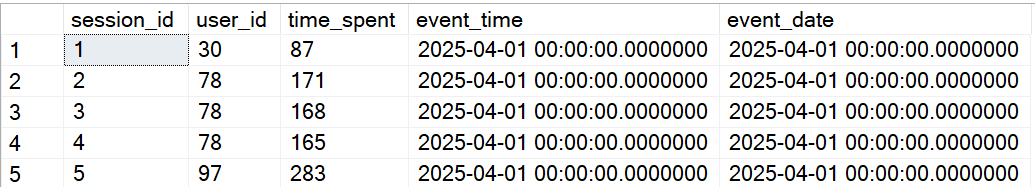
* level\_attempt (71,471 rows)



* player\_action (21,304 rows)

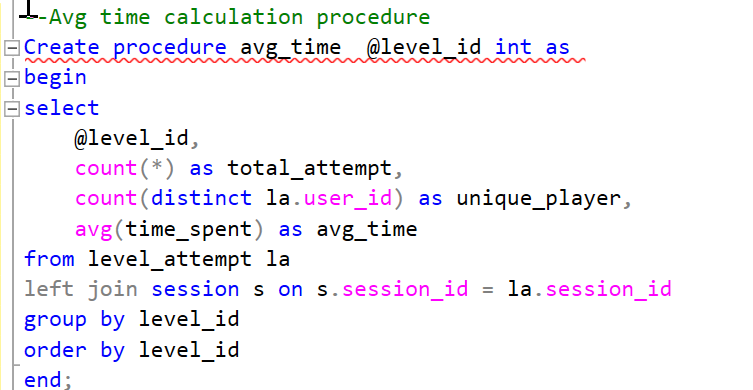


* session (11,862 rows)

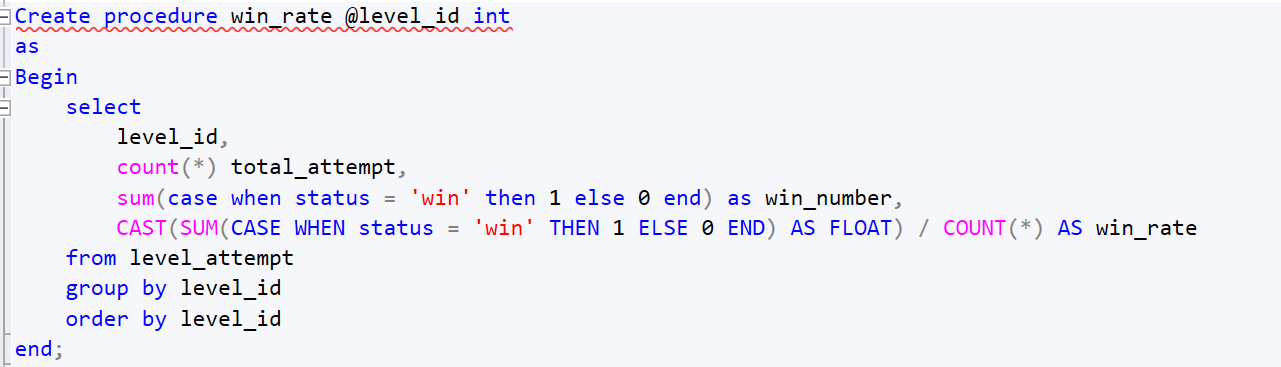


**2.4 view and procedure**

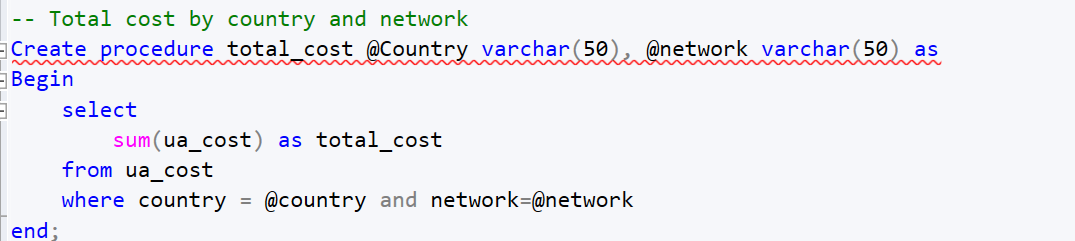
This stored procedure is designed to calculate the average time spent on a level, based on user attempts recorded in the game.

****

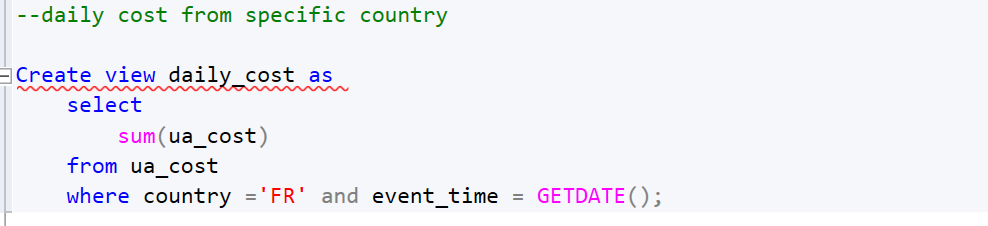
This procedure calculates the win rate for each level in the game, helping to evaluate level difficulty and player performance.

****

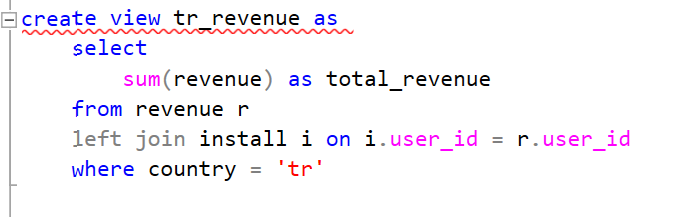
This procedure calculates the total user acquisition (UA) cost for a specific combination of country and network.

****

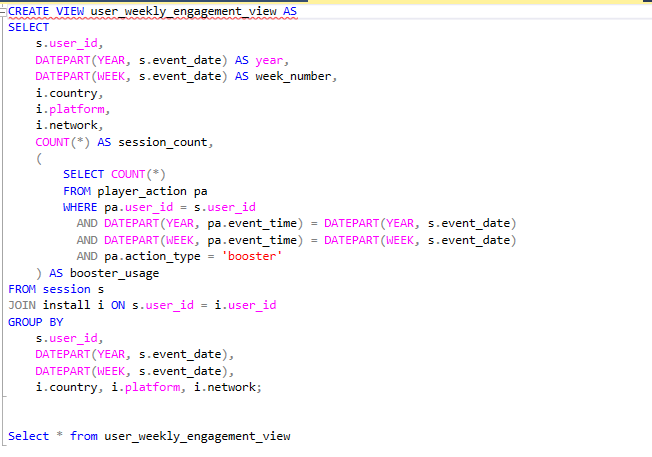
This view computes the daily total user acquisition cost (UA cost) specifically for France (FR) on the current date.

****

This view calculates the total revenue generated from users in Turkey (country = 'tr') by joining the revenue and install tables.

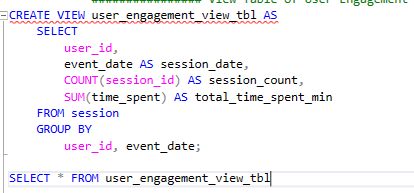
****

**—- View**

****

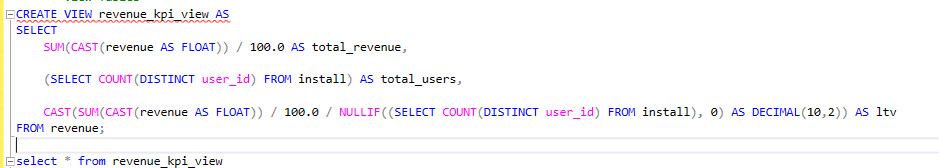
The user\_weekly\_engagement\_view summarizes each user's weekly activity by calculating their total number of sessions and booster uses per week. It groups the data by year and week number, and includes segmentation fields like country, platform, and network from the install table. Booster usage is filtered from the player\_action table for actions labeled 'booster', and matched by week and year. This view helps track user engagement trends across different segments on a weekly basis.

**— view2**

****

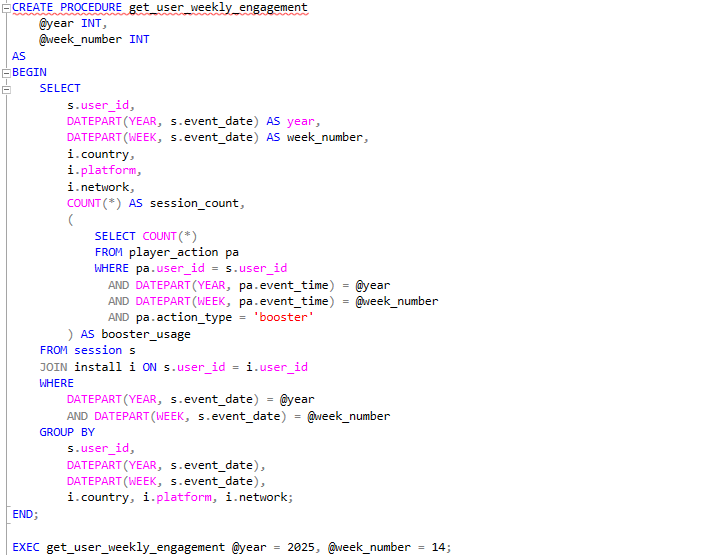
The user\_engagement\_view\_tbl provides a daily summary of each user's activity. For every user and date, it shows how many sessions they started (session\_count) and the total time they spent in the app (total\_time\_spent\_min). This view helps track individual user engagement over time and is useful for analyzing usage trends, daily activity levels, and user retention behavior.

**— view 3**

****

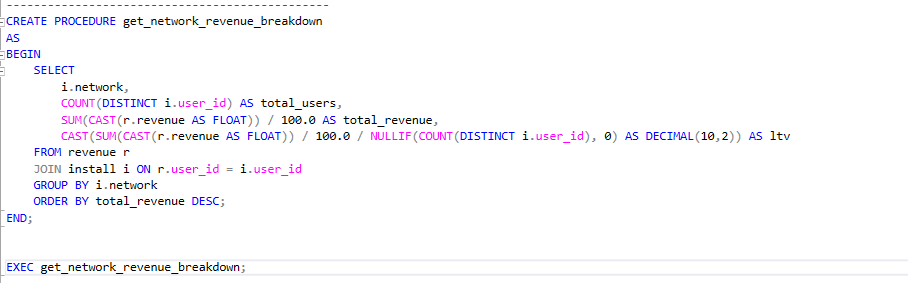
The revenue\_kpi\_view provides a high-level summary of key monetization metrics, including total revenue, total number of users, and lifetime value (LTV). It calculates the total revenue by summing all in-game purchases (adjusted from cents to dollars), counts the total installs to determine user base size, and divides revenue by user count to compute LTV. This view is designed to simplify reporting and is especially useful for dashboards and performance tracking in Power BI or other analytics tools.

**—- procedure**

****

The get\_user\_weekly\_engagement procedure returns weekly engagement data for users, filtered by a given year and week number. For each user, it calculates the total number of sessions (session\_count) and booster usages (booster\_usage) during that week. It also includes segmentation fields such as country, platform, and network from the install table. This procedure helps analyze user activity and feature interaction (e.g., booster usage) on a weekly basis and is especially useful for comparing engagement across different user segments and time periods.

**—-- procedure 2**

****

The get\_network\_revenue\_breakdown procedure provides a breakdown of revenue performance by marketing network. For each network (e.g., Facebook, Google Ads), it returns the total number of users, total revenue (converted from cents to dollars), and the average revenue per user (LTV). It joins the install and revenue tables by user ID to connect revenue with acquisition source, making it useful for evaluating the effectiveness and return on investment (ROI) of each user acquisition channel. Results are ordered by total revenue in descending order for easy comparison.

**— procedure 3**

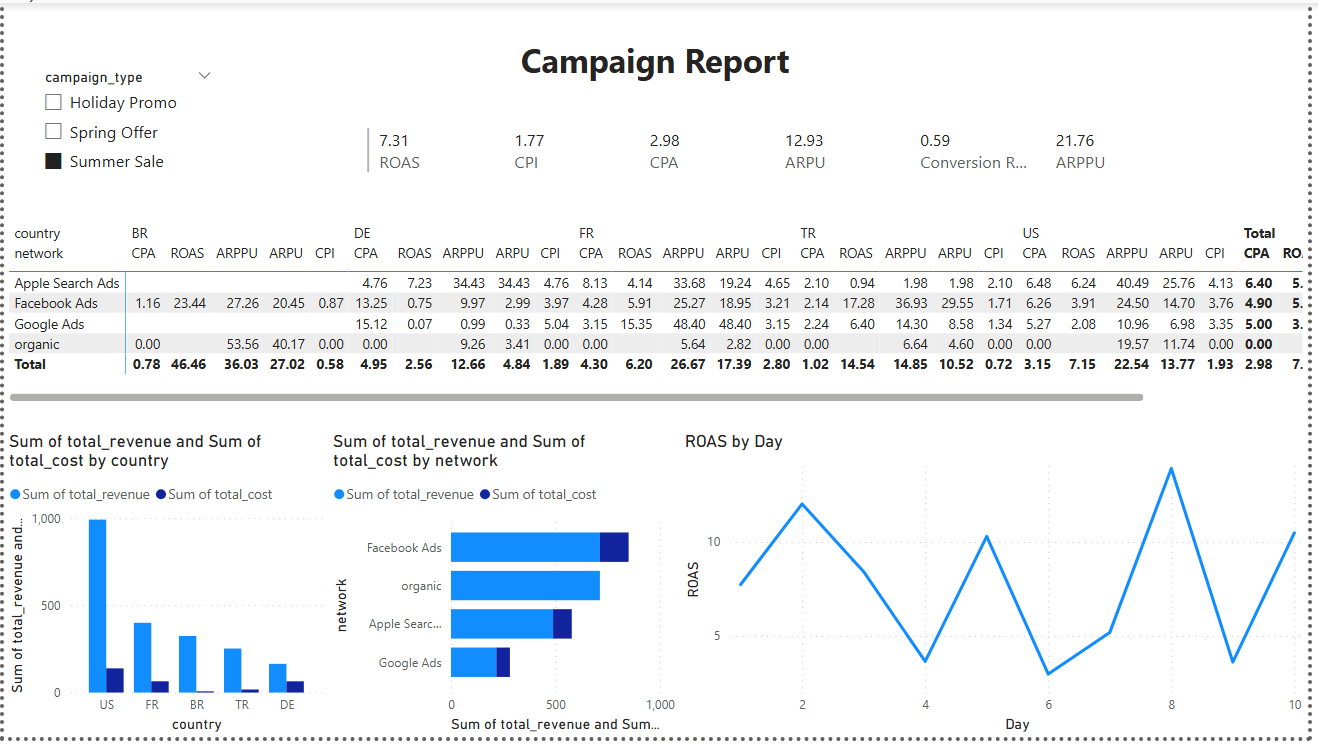
****

The get\_daily\_revenue\_summary procedure calculates daily revenue and ARPDAU (Average Revenue Per Daily Active User) for a given date range. It sums the daily revenue from the revenue table (adjusted to dollars) and counts daily active users (DAU) from the session table. By dividing revenue by DAU for each day, it computes the ARPDAU. This procedure helps monitor monetization trends over time and is useful for identifying peak performance days or evaluating the impact of marketing campaigns.

**3.Descriptions of Reports**

**3.1 Campaign Report**

The **"Campaign Report"** is an analytical Power BI dashboard that provides insights into the performance of different ad campaigns across countries and ad networks. It includes summary KPIs, detailed breakdowns, and visualizations to support marketing and UA (User Acquisition) strategy evaluation.



I used the ua\_cost table and revenue table.

ua\_cost column: user\_id, revenue, network,country, campaign\_type, event\_time

revenue columns:user\_id, revenue

**Derived/Calculated Metrics** (from Power BI)

* + **CPA**: DIVIDE(SUM(CampaignMetrics[total\_cost]), SUM(CampaignMetrics[paying\_users]))
  + **CPI**:DIVIDE(SUM(CampaignMetrics[total\_cost]), SUM(CampaignMetrics[total\_users\_acquired]))
  + **ROAS**: DIVIDE(SUM(CampaignMetrics[total\_revenue]), SUM(CampaignMetrics[total\_cost]))
  + **ARPU**: DIVIDE(SUM(CampaignMetrics[total\_revenue]), SUM(CampaignMetrics[total\_users\_acquired]))
  + **ARPPU**: DIVIDE(SUM(CampaignMetrics[total\_revenue]), SUM(CampaignMetrics[paying\_users]))
  + **Conversion Rate**: DIVIDE(SUM(CampaignMetrics[paying\_users]), SUM(CampaignMetrics[total\_users\_acquired]))

#### 

#### 

#### **Visualizations Used**

#### **1. KPI Cards**

* Metrics: ROAS, CPI, CPA, ARPU, Conversion Rate, ARPPU
* These are summarized across all campaigns to give an at-a-glance view.

#### **2. Matrix Table**

* Rows: Ad Networks (e.g., Apple Search Ads, Facebook Ads, Google Ads, Organic)
* Columns: Countries (BR, DE, FR, TR, US)
* Values: CPA, ROAS, ARPPU, ARPU, CPI for each combination
* This allows detailed comparison of campaign efficiency across countries and networks.

#### **3. Clustered Bar Charts**

* **Chart 1**: Sum of Total Revenue vs Sum of Total Cost by Country
* **Chart 2**: Sum of Total Revenue vs Sum of Total Cost by Network
* These help identify the most profitable regions and marketing channels.

#### **4. Line Chart**

* **ROAS by Day**: Time-series line chart showing how ROAS evolves day by day
* Helps detect performance trends or seasonal variations

#### **5. Metric Slicer**

* Allows users to dynamically filter the report by **metric type**.

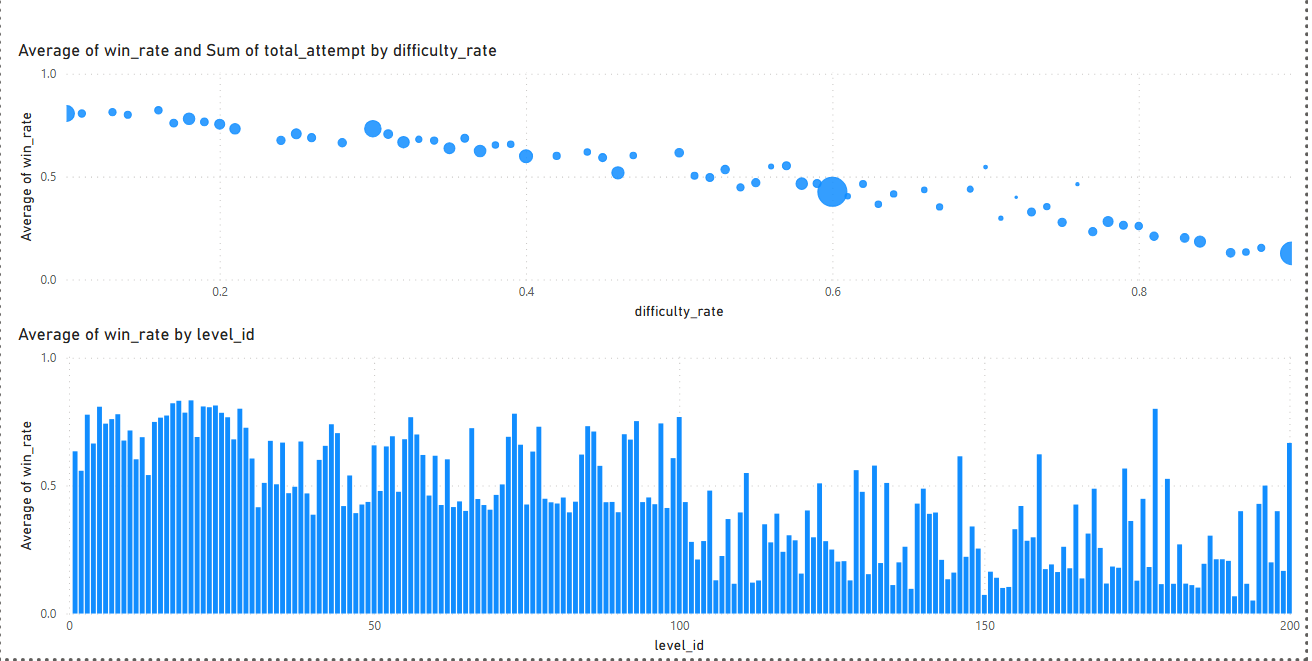
**3.2 Level Report**

This dashboard analyzes how level design (difficulty, progression) affects player performance and behavior. level, level\_attempt,session tables are used.

level table columns: level\_id, difficulty\_rate

level\_attempt table columns: level\_id, status,event\_time, attempt\_number, sessin\_id

session table columns: session\_id, time\_spent



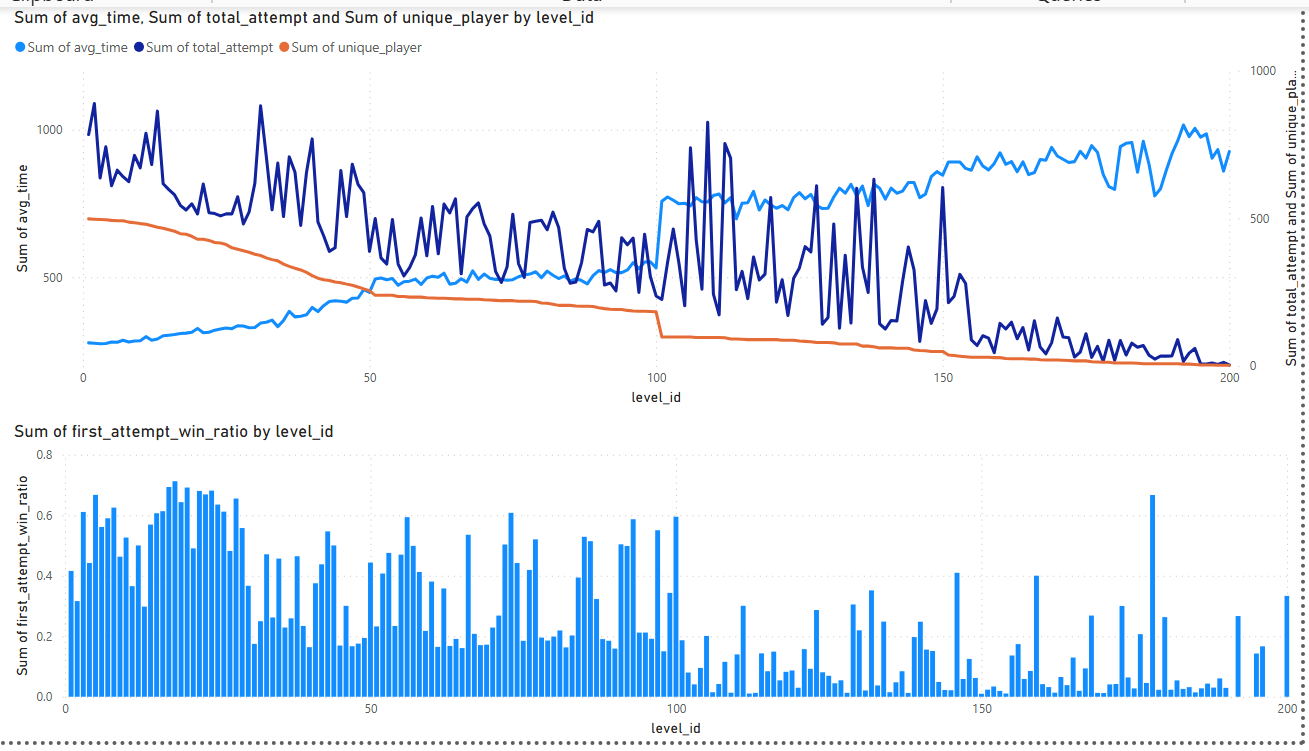
#### **Visualizations Used**

### **1. Scatter Chart with Bubble Size (Top Left)**

* **X-axis**: difficulty\_rate
* **Y-axis**: win\_rate
* **Size**: total\_attempt
* **Purpose**: Shows how level difficulty impacts win rates. Larger bubbles indicate levels with more total attempts, providing context on popularity or frequency. A downward trend suggests that higher difficulty correlates with lower win rates.

### **2. Clustered Column Chart (Top Right)**

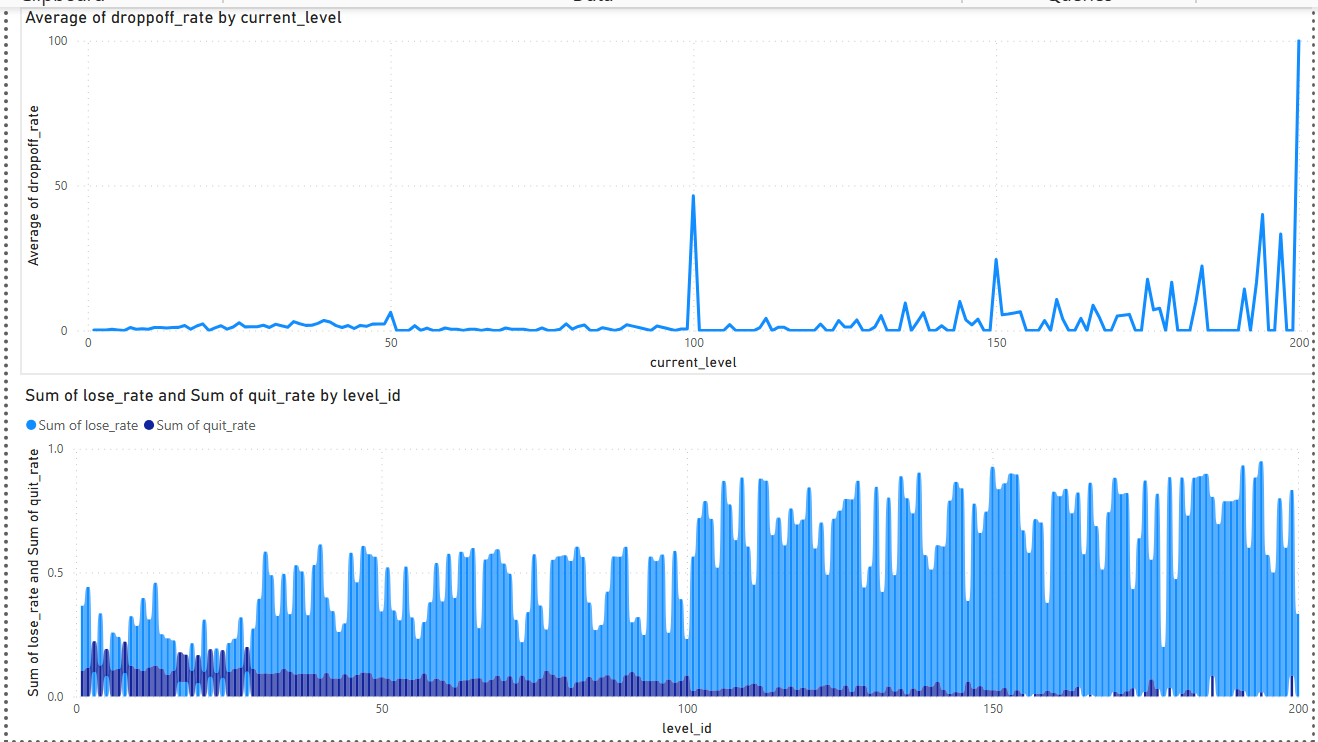
* **X-axis**: level\_id
* **Y-axis**: win\_rate
* **Purpose**: Displays win rate distribution across all game levels. Helps identify outlier levels with extremely high or low success rates, which may need balance adjustments.



#### **Visualizations Used**

### **Combined Line & Clustered Column Chart (Middle Section)**

* **Line Charts**:  
  + avg\_time (average time spent)
  + total\_attempt (number of attempts)
  + unique\_player (unique users)
* **Column Chart**:  
  + first\_attempt\_win\_ratio
* **X-axis**: level\_id
* **Purpose**: Analyzes time engagement and difficulty. Identifies levels where players spend too long, retry often, or succeed on the first try — valuable for UX balancing.



### **4. Line Chart (Bottom Left)**

* **X-axis**: current\_level
* **Y-axis**: dropoff\_rate
* **Purpose**: Highlights user drop-off behavior. Sharp peaks may indicate frustrating or broken levels where users quit the game.

### **5. Ribbon (Bottom Right)**

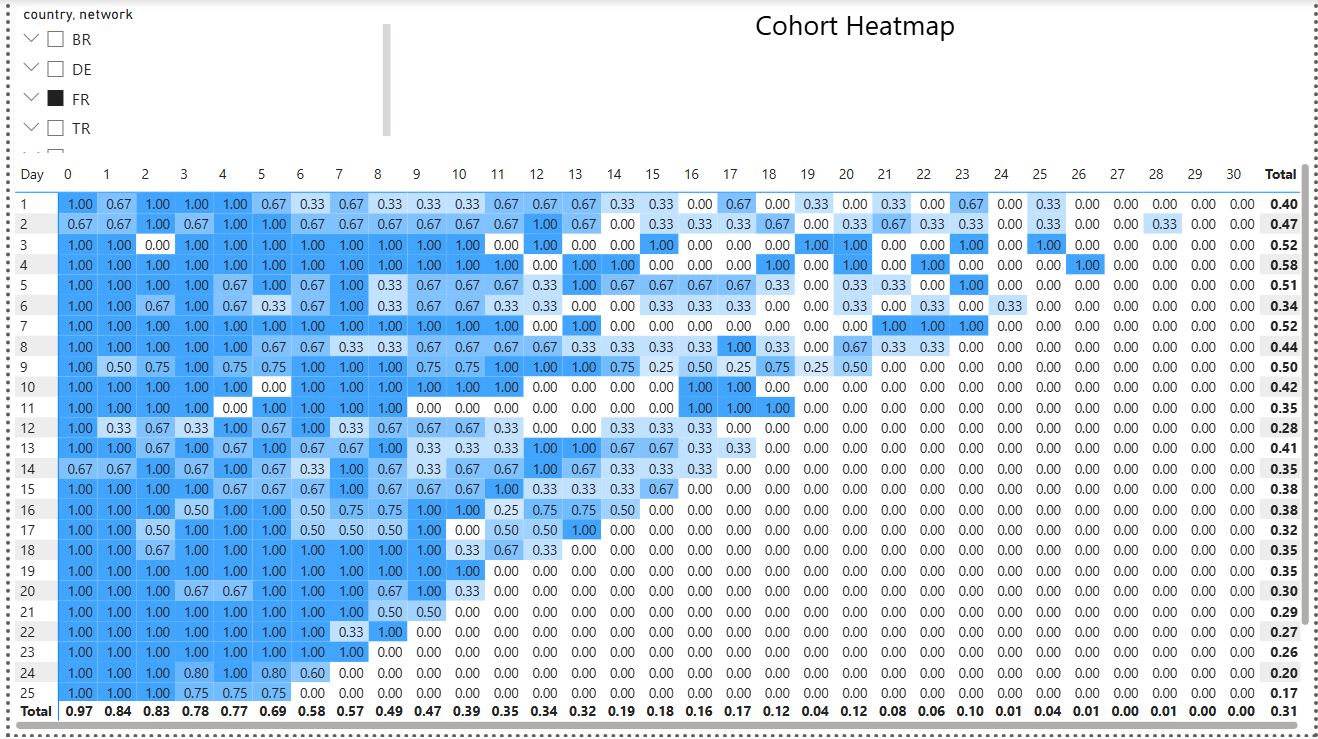
* **X-axis**: level\_id
* **Y-axis**:  
  + lose\_rate
  + quit\_rate
* **Purpose**: Compares the proportion of users who lose versus those who quit entirely at each level. Useful to distinguish between challenge-induced retries vs. actual churn.

**3.3 Retention Report**

Install session tables used.

install table columns: user\_id,event\_time, country, network

session table columns: user\_id, event\_time

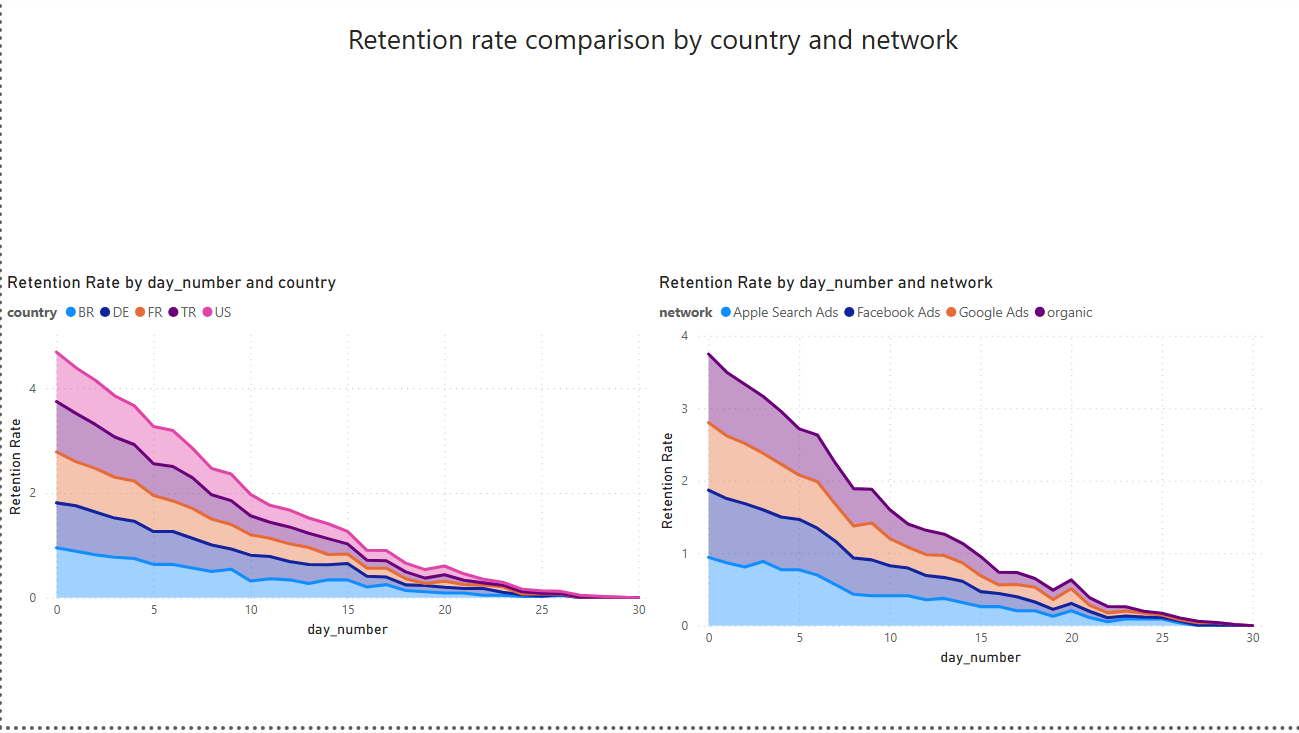


### **1. Matrix Table (Top Section – Cohort Heatmap)**

* **Title**: *Cohort Heatmap*
* **X-axis**: Day (0 to 30)
* **Y-axis**: Install date
* **Values**: retention rate (calculated as retained users / total users per cohort)
* **Slicer Filters**:  
  + country
  + network
* **Purpose**:  
   This matrix heatmap shows user retention behavior across 30 days after install for different cohorts. Darker shades indicate higher retention. It helps identify how long users remain active and compare behavior across cohorts, countries, or campaigns using the slicers.

### **2. Slicer Visual (Top Left of Heatmap)**

* **Fields**: country, network
* **Type**: slicer
* **Purpose**:  
   Allows dynamic filtering of the cohort heatmap to isolate specific segments (e.g., French users acquired via Facebook Ads). This enables comparative and targeted cohort analysis.



### **3. Stacked Area Chart (Bottom Left)**

* **Title**: *Retention Rate by day\_number and country*
* **X-axis**: day\_number (0 to 30)
* **Y-axis**: retention rate
* **Legend**: country (BR, DE, FR, TR, US)
* **Purpose**:  
   Shows retention rate trends across different countries. Helps identify which regions retain users longer and can guide geo-specific optimizations or marketing strategies.

### **4. Stacked Area Chart (Bottom Right)**

* **Title**: *Retention Rate by day\_number and network*
* **X-axis**: day\_number (0 to 30)
* **Y-axis**: retention rate
* **Legend**: network (Apple Search Ads, Facebook Ads, Google Ads, organic)
* **Purpose**:  
   Visualizes how different acquisition channels affect retention. For instance, organic users might show stronger long-term retention than paid campaigns. Useful for ROI and UA channel performance analysis.

**3.4 Cost and Revenue Report**

revenue and ua\_cost tables used.

revenue table columns: user\_id, revenue

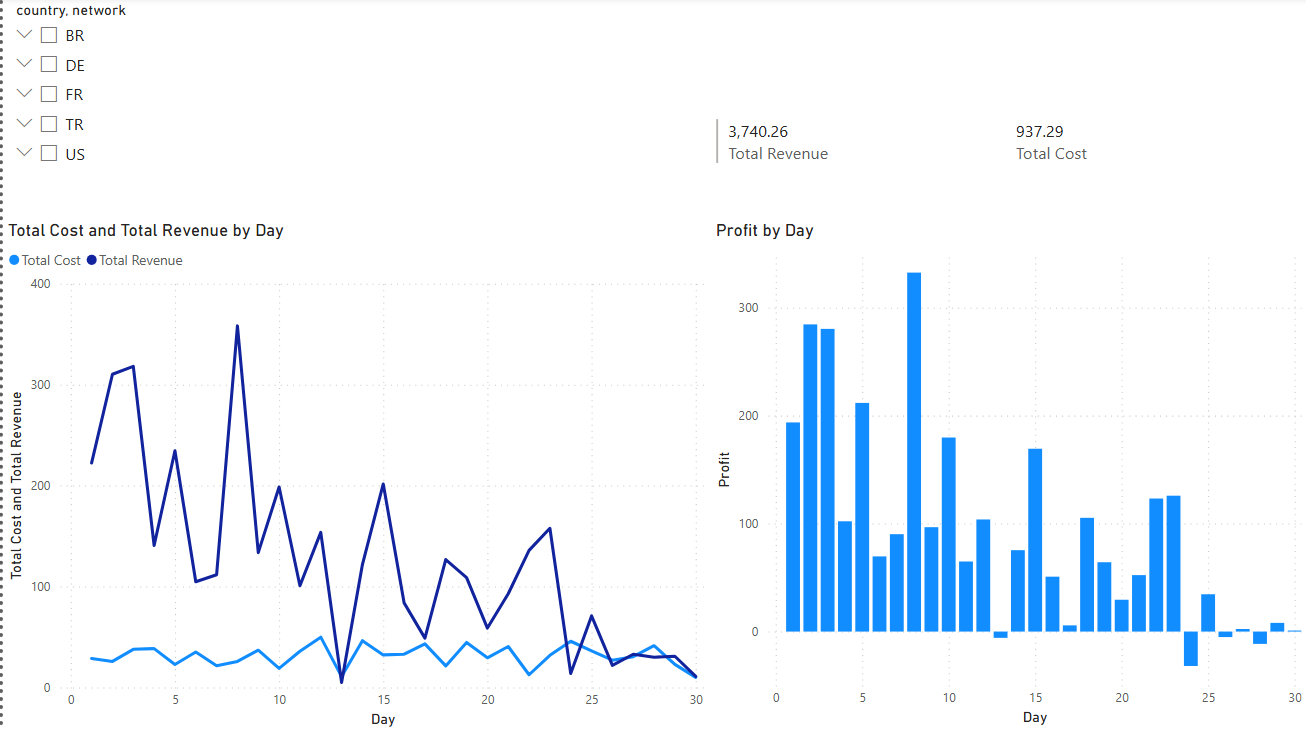
ua\_cost table columns: user\_id,campaign\_type, event\_time,network,country, ua\_cost

### **1. Slicer (Top Left of Each Page)**

* **Fields**: country, network
* **Purpose**:  
   Enables filtering across all charts to segment performance by specific regions and acquisition channels. Essential for comparative analysis.

### **2. Multi-Row Card Visuals**

* **Metrics**: Total Revenue, Total Cost, CPI, ROI
* **Purpose**:  
   Displays key financial KPIs in a compact, easily readable format. Automatically updates based on slicer filters.



### 

### 

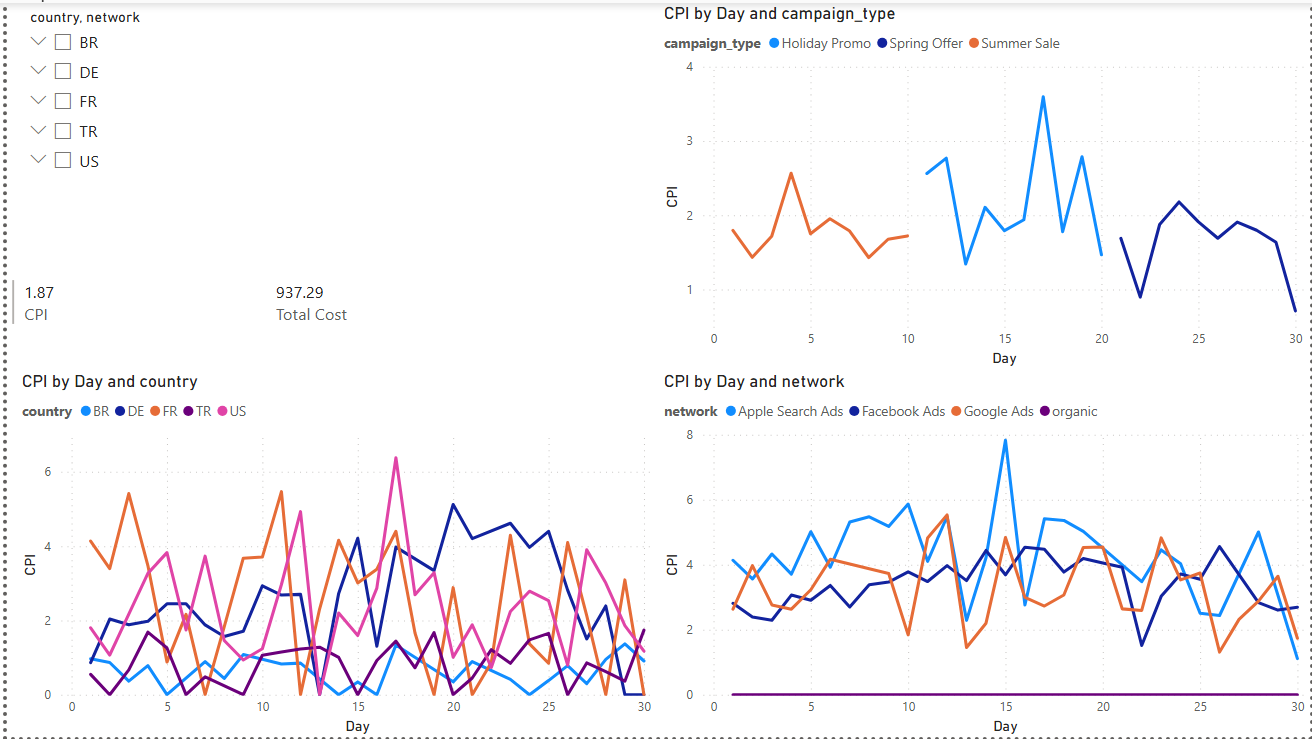
### 

### **3. Line Chart – Total Cost and Total Revenue by Day**

* **X-axis**: Day
* **Y-axis**: Total Cost, Total Revenue
* **Purpose**:  
   Visualizes daily revenue vs. cost trends. Helps identify profitable and unprofitable periods.

### **4. Clustered Column Chart – Profit by Day**

* **X-axis**: Day
* **Y-axis**: Profit (Revenue - Cost)
* **Purpose**:  
   Quickly identifies days with net positive or negative profit. Supports budgeting and campaign timing decisions.



### 

### **5. Line Chart – CPI by Day and Campaign Type**

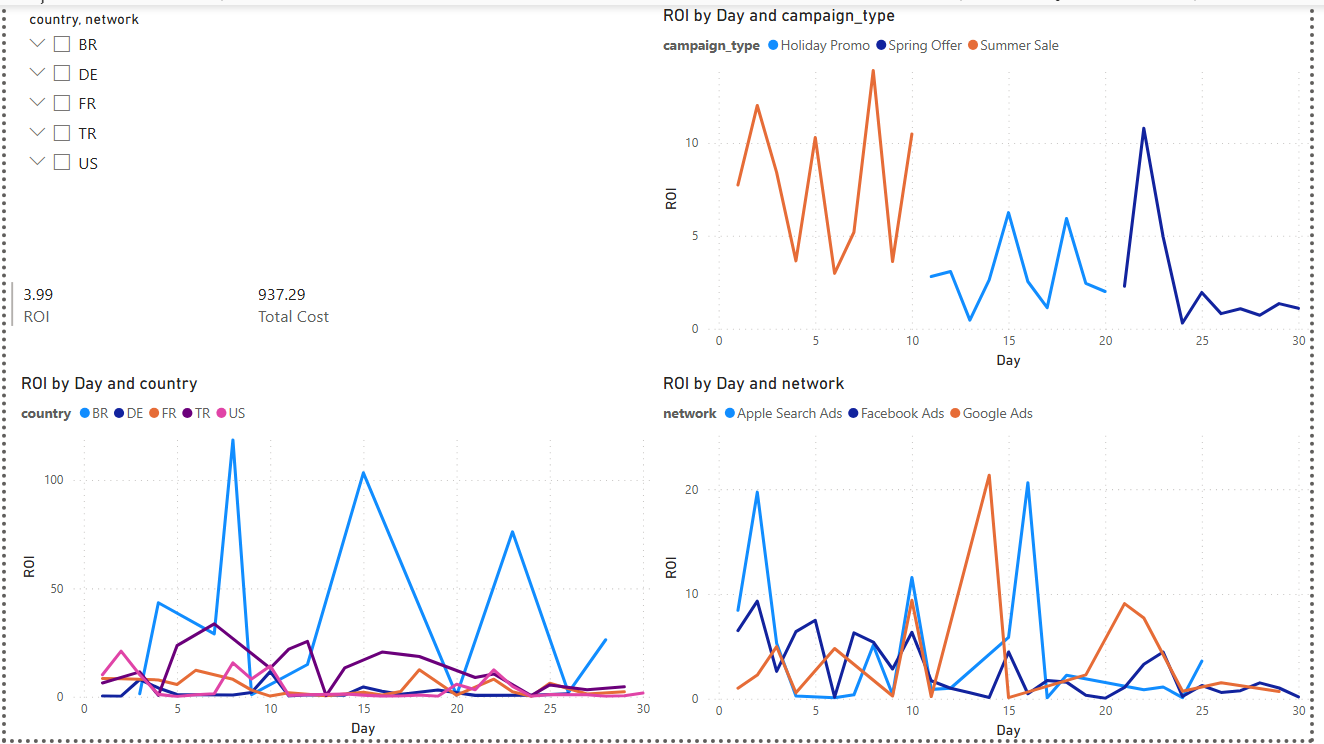
* **X-axis**: Day
* **Y-axis**: CPI
* **Legend**: campaign\_type (Holiday Promo, Spring Offer, Summer Sale)
* **Purpose**:  
   Evaluates which campaign types are more cost-efficient per install across time.

**6. Line Chart – CPI by Day and Country**

* **X-axis**: Day
* **Y-axis**: CPI
* **Legend**: country
* **Purpose**:  
   Highlights regional cost variation. Aids in targeting countries with lower user acquisition costs.

### **7. Line Chart – CPI by Day and Network**

* **X-axis**: Day
* **Y-axis**: CPI
* **Legend**: network
* **Purpose**:  
   Helps compare how different marketing channels perform in terms of CPI over time.



### **8. Line Chart – ROI by Day and Campaign Type**

* **X-axis**: Day
* **Y-axis**: ROI
* **Legend**: campaign\_type
* **Purpose**:  
   Determines which campaigns provide the best return. Spikes can indicate high-performing campaigns or days.

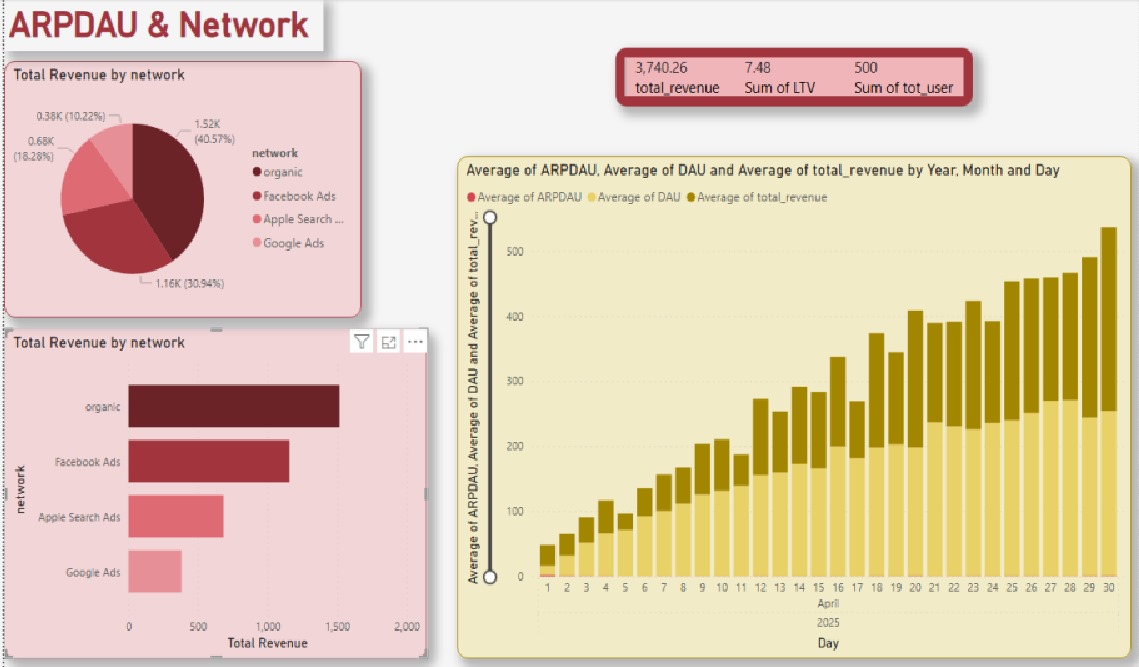
### **9. Line Chart – ROI by Day and Country**

* **X-axis**: Day
* **Y-axis**: ROI
* **Legend**: country
* **Purpose**:  
   Tracks which markets generate the highest return on ad spend.

### **10. Line Chart – ROI by Day and Network**

* **X-axis**: Day
* **Y-axis**: ROI
* **Legend**: network
* **Purpose**:  
   Compares the profitability of each advertising channel over time.

**3.5 Revenue Analysis**

****

## **1. KPI Card Section (Top Right)**

* **Title:** Revenue KPIs
* **Fields Used:**
  + revenue.revenue
  + install.user\_id
* **Values:**
  + total\_revenue = SUM(revenue)/100
  + total\_users = COUNT(DISTINCT user\_id)
  + ltv = total\_revenue / total\_users
* **Purpose:** To present key monetization metrics in a compact form, including total revenue, lifetime value (LTV), and total user count.

## **2. Pie Chart (Top Left)**

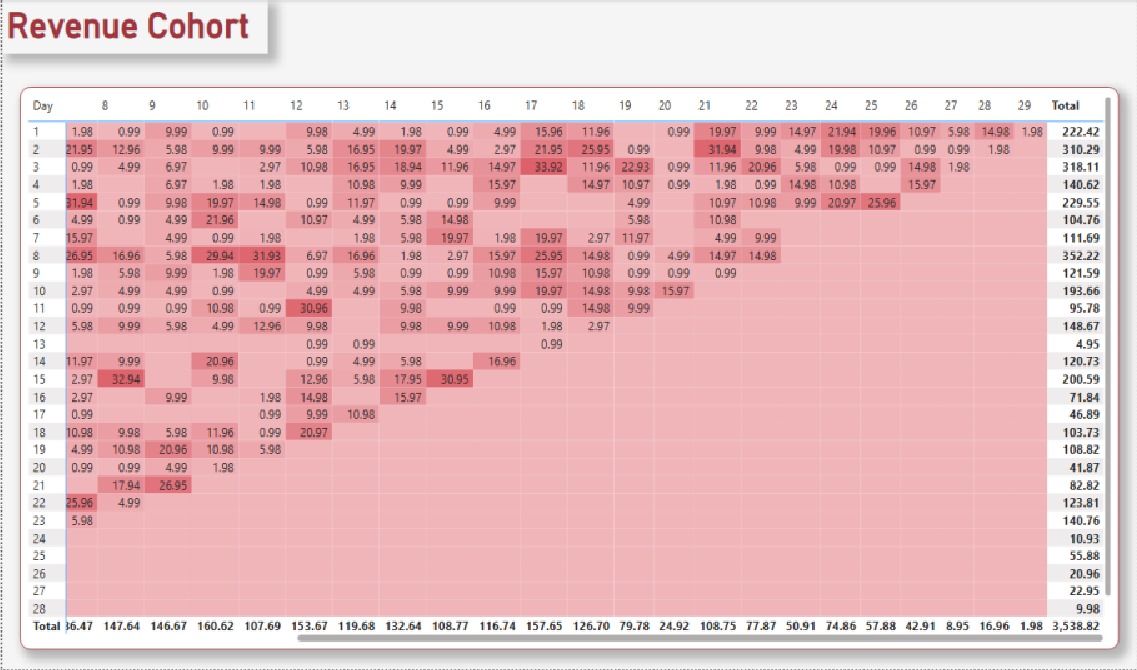
* **Title:** Total Revenue by Network
* **Table Used:** revenue, install
* **Fields Used:**
  + install.network, revenue.revenue
* **Visual Type:** Pie Chart
* **Value:** SUM(revenue)
* **Legend:** network
* **Purpose:** To show the distribution of total revenue across different acquisition networks. It helps identify the most valuable channels by share of income.

## **3. Bar Chart (Bottom Left)**

* **Title:** Total Revenue by Network (Bar)
* **Table Used:** revenue, install
* **Fields Used:**
  + install.network, revenue.revenue
* **Visual Type:** Clustered Bar Chart
* **X-axis:** Total Revenue
* **Y-axis:** Network
* **Purpose:** Same as the pie chart, but optimized for precise value comparison between acquisition channels.

## **4. Column Chart (Right Side)**

* **Title:** Average of ARPDAU, DAU, and Revenue (Daily)
* **Table Used:** revenue, session
* **Fields Used:**
  + revenue.event\_time, revenue.revenue,
  + session.event\_date, session.user\_id
* **Visual Type:** Clustered Column Chart
* **X-axis:** Day (from event\_time)
* **Y-axis (multiple):**
  + Average(ARPDAU)
  + Average(DAU)
  + Average(total\_revenue)
* **Purpose:** To show how revenue and daily user activity evolved over time. Useful for understanding monetization trends and the correlation between DAU and revenue performance.

****

## **Revenue Cohort – Matrix Table (Heatmap)**

* **Title:** Revenue Cohort Heatmap
* **Visual Type:** Matrix table with conditional formatting

### **1. Fields Used:**

* **Tables:**
  + install
  + revenue
* **Columns:**
  + install.event\_time (as install\_date)
  + revenue.event\_time (as revenue\_date)
  + revenue.revenue
* **Custom Field:**
  + day\_number = DATEDIFF(DAY, install\_date, revenue\_date)
  + Filtered between day 0–30

**2. Matrix Configuration:**

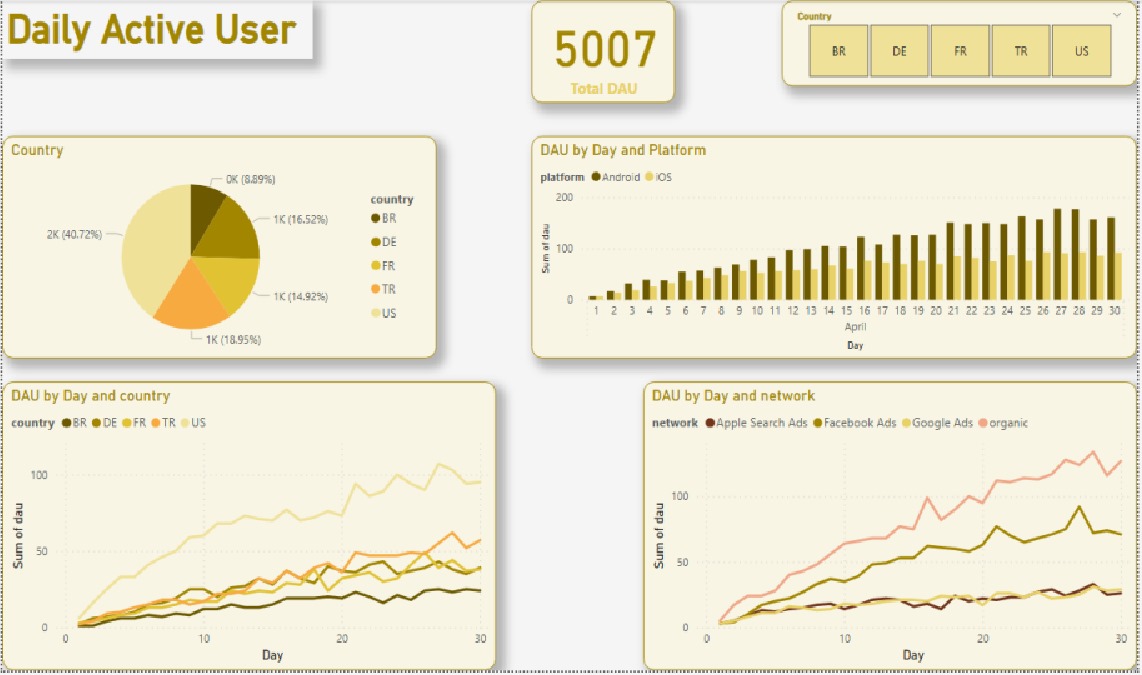
* **X-axis (Columns):** day\_number (0 to 30)
* **Y-axis (Rows):** install\_date (daily cohort)
* **Values:** SUM(revenue)
* **Conditional Formatting:** → Darker red = higher revenue  
   → Lighter = lower/zero revenue

### **Purpose:**

This matrix shows how much revenue each install cohort (based on the day they started) generates across the following 30 days. It helps visualize long-term user value and monetization patterns by cohort. Spikes in rows or columns help identify successful campaigns or strong day-X monetization strategies. The final column summarizes total revenue per cohort, while the final row shows revenue by day across all cohorts.

**3.6 User Engagement Analysis**

**Daily Active User Dashboard**



### **1. Card Visual (Top Center)**

* **Title:** Total DAU
* **Visual Type:** Card
* **Table:** session
* **Field:** DISTINCT COUNT(user\_id)
* **Purpose:** Shows the total number of unique daily active users across the dataset.

### **2. Pie Chart (Top Left)**

* **Title:** DAU by Country
* **Visual Type:** Pie Chart
* **Table:** session + install
* **Fields:**
  + install.country
  + COUNT(DISTINCT session.user\_id)
* **Purpose:** To show the distribution of daily active users by country and identify the most engaged user bases geographically.

### **3. Column Chart (Top Right)**

* **Title:** DAU by Day and Platform
* **Visual Type:** Clustered Column Chart
* **X-axis:** event\_date
* **Y-axis:** Sum of DAU
* **Legend:** platform
* **Tables:** session, install
* **Purpose:** Compare user engagement across platforms (iOS vs Android) over time.

### **4. Line Chart (Bottom Left)**

* **Title:** DAU by Day and Country
* **Visual Type:** Stacked Line Chart
* **X-axis:** Day
* **Y-axis:** DAU
* **Legend:** country
* **Tables:** session, install
* **Purpose:** To observe daily engagement growth by country and detect regional differences.

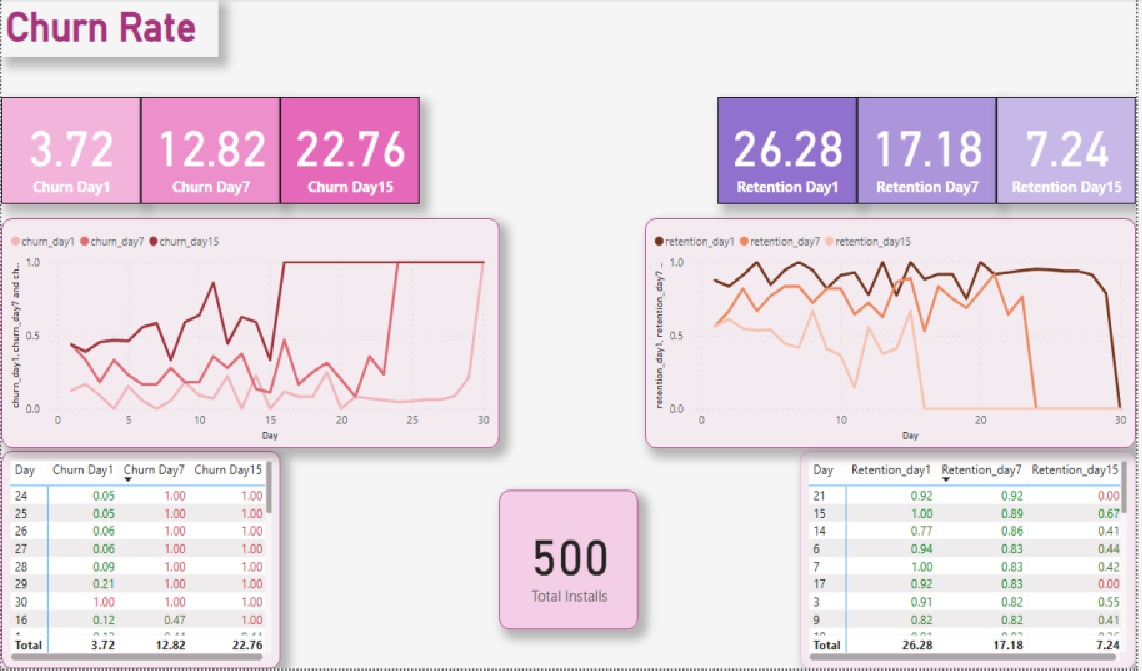
### **5. Line Chart (Bottom Right)**

* **Title:** DAU by Day and Network
* **Visual Type:** Stacked Line Chart
* **X-axis:** Day
* **Y-axis:** DAU
* **Legend:** network
* **Tables:** session, install
* **Purpose:** Show DAU performance by acquisition channel (e.g., organic vs. Facebook Ads), useful for UA team comparisons.

### **6.Slicer (Top Right)**

* **Fields:** country
* **Type:** Slicer
* **Purpose:** Enables focused comparison across countries by filtering the entire dashboard view.

## **Retention & Churn Analysis Dashboard**



### **1.KPI Cards (Top Section)**

* **Visual Type:** Card
* **Fields:**
  + Retention Day1, 7, 15
  + Churn Day1, 7, 15
  + Total Installs
* **Table Used:** View or procedure output (retention summary)
* **Purpose:** These cards provide a quick overview of short-term user loyalty and loss. They’re essential for tracking user behavior within the first few days post-installation.

### **2. Line Chart (Left Center)**

* **Title:** Churn Rate Over Time
* **Visual Type:** Line Chart
* **X-axis:** Day
* **Y-axis:** Churn Rate
* **Legend:** churn\_day1, churn\_day7, churn\_day15
* **Purpose:** This chart visualizes user drop-off trends over time. It helps understand when users are most likely to stop returning and highlights areas needing retention improvement.

### **3.Line Chart (Right Center)**

* **Title:** Retention Rate Over Time
* **Visual Type:** Line Chart
* **X-axis:** Day
* **Y-axis:** Retention Rate
* **Legend:** retention\_day1, retention\_day7, retention\_day15
* **Purpose:** Shows how user loyalty changes over time. Identifies which days users are most likely to return and stay engaged.

### **4. Table (Bottom Left)**

* **Title:** Churn Rate Table
* **Visual Type:** Table
* **Columns:** Day, Churn Day1, Churn Day7, Churn Day15
* **Purpose:** Provides exact churn percentages per cohort day for precision reporting and deeper comparison.

### **5.Table (Bottom Right)**

* **Title:** Retention Rate Table
* **Visual Type:** Table
* **Columns:** Day, Retention Day1, Retention Day7, Retention Day15
* **Purpose:** Complements the line chart by providing exact values for cohorts. Ideal for spotting drop-offs or successes in retention patterns.

### **Overall Purpose:**

This page is designed to monitor and improve user retention by clearly showing **when and how users drop off** or return. It supports lifecycle marketing, push timing, and feature optimization strategies.

## **Boosters & Sessions Dashboard**



### **1.KPI Cards (Top Section)**

* **Title:** Daily & Weekly KPIs
* **Visual Type:** Card
* **Fields:**
  + avg\_daily\_boosters\_per\_user
  + avg\_weekly\_boosters\_per\_user
  + avg\_daily\_sessions\_per\_user
  + avg\_weekly\_sessions\_per\_user
* **Tables Used:**
  + View or combined CTE using session and player\_action tables
* **Purpose:** These cards summarize average engagement per user. Perfect for quick monitoring of core activity metrics like playing and booster usage frequency.

### **2.Clustered Bar Chart (Bottom Left)**

* **Title:** Booster vs Session Usage
* **Visual Type:** Clustered Bar Chart
* **X-axis:** Metric Type
* **Y-axis:** Average Value
* **Fields:**
  + Daily\_boosters, Daily\_sessions, Weekly\_sessions, Weekly\_boosters
* **Purpose:** Visual comparison of user activity types. Highlights whether users are more engaged with core gameplay (sessions) or monetizable features (boosters).

### **3.Donut Chart (Bottom Right)**

* **Title:** Proportion of Engagement Types
* **Visual Type:** Donut Chart
* **Fields:** Same as bar chart
* **Purpose:** Emphasizes distribution between session and booster behavior. Helps understand feature usage balance and the weight of monetization-related actions.

**Overall Purpose:**

This dashboard tracks how frequently users interact with the app and its monetizable features. It supports feature design, in-game economy balance decisions, and even tutorial placement (if boosters are underused, maybe they need better onboarding!).

**4 Responsibilities in the Project**

Şamil- data generation, reports 1,2,3,4

Onur- reports 5,6