**Vandita Online Game Data Analysis Report**

**Personal Note :**

First of all thank you very much for this amazing assignment. I find it quite challenging and very interesting and I have learned lot of new things while doing this assignment. I am very glad that I got this assignment because it upscaled my skills in SQL and Data Analysis. As I am a recent Post graduate I don’t have any professional experience still I have done some projects you can find it [here](https://github.com/Rohit-Bhavikatte1?tab=repositories) , but I didn’t encounter challenges that I faced here during this assignment.

**Introduction :**

This report provides an analysis of the loyalty points system for an ABC online gaming platform.  
The analysis is based on three datasets: user data, deposit data, and withdrawal data.  
Key objectives include calculating loyalty points, identifying top players, and providing recommendations for improving player engagement.

**Dataset Overview :**

The dataset consists of three tables: user\_data, deposit\_data, and withdrawal\_data.  
The total records in each dataset are calculated, and NULL value checks confirm data integrity.

**Loyalty Points Calculation :**

Q 1. Find Playerwise Loyalty points earned by Players in the following slots:-

a. 2nd October Slot S1

b. 16th October Slot S2

b. 18th October Slot S1

b. 26th October Slot S2

SOLUTION :

For this i first need to gather the data like creating slots and computing

aggregates like, total deposit, total withdrawal , total no. of wthdrawal etc....

i wll create independent cte's for easier reading and debugging

slots cte i'm using user\_data table for this

WITH slots\_cte

AS(

SELECT

user\_id,

SUM(games\_played) AS total\_played\_games,

CASE

WHEN CONVERT(TIME, datetime) BETWEEN '00:00:00' AND '11:59:00' THEN 'slot 1' getting only time in hours and minutes from datetime column

ELSE 'slot 2'

END AS slots, --- Creating slots

(CAST(SUM(games\_played) AS FLOAT) \* 0.2) AS total\_played\_games\_points

FROM user\_data

GROUP BY

user\_id,

CASE

WHEN CONVERT(TIME, datetime) BETWEEN '00:00:00' AND '11:59:00' THEN 'slot 1'

ELSE 'slot 2'

END

)

creating the temporary table to avoid executing cte everytime.and inserting all the data into

temporary table

SELECT \*

INTO #slots

FROM slots\_cte

depo-cte its used deposit data for this

WITH depo\_cte AS

(SELECT

user\_id,

SUM(dip\_amount) AS total\_diposit\_per\_user,

(CAST(SUM(dip\_amount) AS FLOAT) \* 0.01) AS deposit\_points,

COUNT(\*) AS no\_of\_deposits

FROM deposit\_data

GROUP BY

user\_id

)

--- Creating and inserting data at same time

SELECT \*

INTO #deposit\_points

FROM depo\_cte

withdhrawal\_cte used withdrawal data for this

WITH withdrawal\_cte AS (

SELECT

user\_id,

CASE

WHEN CONVERT(TIME, datetime) BETWEEN '00:00:00' AND '11:59:59' THEN 'S1'

ELSE 'S2'

END AS slots,

SUM(wid\_amount) AS total\_withdrawal,

(CAST(SUM(wid\_amount) AS FLOAT) \* 0.01) AS withdrawal\_points,

COUNT(\*) AS num\_withdrawals

FROM withdrawal\_data

GROUP BY user\_id,

CASE

WHEN CONVERT(TIME, datetime) BETWEEN '00:00:00' AND '11:59:59' THEN 'S1'

ELSE 'S2'

END

)

Creating and inserting data at same time

SELECT \*

INTO #withdrawal\_points

FROM withdrawal\_cte

Checking all the tables.

SELECT \* FROM #slots

SELECT \* FROM #deposit\_points

SELECT \* FROM #withdrawal\_points

Calculating all the points and required columns

In this query i have joined all the temporary tables

WITH loyalty\_points AS

(SELECT

s.user\_id,

s.slots,

dep.datetime,

COALESCE(d.total\_diposit\_per\_user, 0) AS total\_diposit,

COALESCE(w.total\_withdrawal, 0) AS total\_withdrawal,

COALESCE(d.no\_of\_deposits, 0) AS no\_diposits,

COALESCE(w.num\_withdrawals, 0) AS num\_withdrawals,

COALESCE(s.total\_played\_games, 0) AS total\_games,

(0.01 \* COALESCE(d.total\_diposit\_per\_user, 0)) +

(0.005 \* COALESCE(w.total\_withdrawal, 0)) +

(0.001 \* GREATEST(COALESCE(d.no\_of\_deposits, 0) - COALESCE(w.total\_withdrawal, 0), 0)) +

(0.2 \* COALESCE(s.total\_played\_games, 0)) AS total\_loyalty\_points

FROM #slots AS s

FULL OUTER JOIN #deposit\_points AS d

ON s.user\_id = d.user\_id

FULL OUTER JOIN #withdrawal\_points AS w

ON d.user\_id = w.user\_id

FULL OUTER JOIN deposit\_data AS dep

ON w.user\_id = dep.user\_id

)

Result of this cte is stored in another temporary table

SELECT \*

INTO #loyalty\_points

FROM loyalty\_points

Getting the final result

SELECT

user\_id,

total\_loyalty\_points,

datetime,

slots

FROM #loyalty\_points

WHERE slots = 'slot 1' AND CONVERT(DATE, datetime) IN('2022-10-02','2022-10-10')

OR slots = 'slot 2' AND CONVERT(DATE, datetime) IN('2022-10-16','2022-10-26')

ORDER BY total\_loyalty\_points DESC

Q2. Calculate overall loyalty points earned and rank players on the basis of loyalty points in the month of October. In case of tie, number of games played should be taken as the next criteria for ranking.

SOLUTION :

SELECT

user\_id,

datetime,

total\_games,

total\_loyalty\_points,

rankings

FROM

-- created subquery to gather data

(SELECT \*,

DENSE\_RANK() OVER(ORDER BY total\_loyalty\_points DESC, total\_games DESC ) AS rankings

FROM #loyalty\_points

WHERE CONVERT(DATE,datetime) BETWEEN '2022-10-01' AND '2022-10-31'

) AS t1

WHERE user\_id IS NOT NULL

AND slots IS NOT NULL

ORDER BY rankings ASC

Q3. What is the average deposit amount?

SOLUTION :

SELECT

user\_id,

SUM(dip\_amount) / COUNT(DISTINCT user\_id) avg\_diposit\_amount

FROM deposit\_data

GROUP BY user\_id

ORDER BY avg\_diposit\_amount DESC

Q4. What is the average deposit amount per user in a month?

SOLUTION :

SELECT

user\_id,

SUM(dip\_amount) / COUNT(DISTINCT user\_id) avg\_diposit\_amount,

DATEPART(MONTH,datetime) AS months

FROM deposit\_data

GROUP BY

user\_id,

DATEPART(MONTH,datetime)

ORDER BY avg\_diposit\_amount DESC

Q5. What is the average number of games played per user?

SOLUTION:

SELECT

user\_id,

SUM(games\_played) / COUNT(DISTINCT user\_id) AS avg\_played\_games

FROM user\_data

GROUP BY

user\_id

ORDER BY avg\_played\_games DESC

Part B - How much bonus should be allocated to leaderboard players?

SOLUTION

My Suggestion is that we should allocate this Bonus amount on the basis of Loyalty points

As we calculate loyalty points on the basis of deposits, withdrawals and number of games played, As it covers all the aspects.

First we can calculate total loyalty points then we calculate loyalty points per user.

After this we calculate loyalty point percentage of the users

On the basis of loyalty point percentage we can give bonus to the users.

I allready calulated loyalty points of the user and created separate temporary table

**#loyalty\_points**

i will select top 50 users from #loyalty\_points according yo highest loyalty points.

Beacuse of the slots and datetime the loyalty points data has many dulicates so

first i will remove all the duplicates and remaninig top 50 playres i will rank them

SELECT \*

INTO #top\_players

FROM

(

SELECT

user\_id,

total\_loyalty\_points,

ROW\_NUMBER() OVER(PARTITION BY user\_id ORDER BY total\_loyalty\_points DESC) AS rankings

FROM #loyalty\_points

WHERE user\_id IS NOT NULL

AND user\_id > 0

) AS t1

WHERE rankings = 1

this is our leaderboard

I will save this into another temporary table #leaderboard.

SELECT TOP 50

user\_id,

total\_loyalty\_points,

player\_ranking

INTO #leaderboard

FROM

(SELECT \*,

DENSE\_RANK() OVER(ORDER BY total\_loyalty\_points DESC) AS player\_ranking

FROM #top\_players

) AS t2

**final Results**

SELECT

user\_id,

total\_loyalty\_points,

ROUND(CAST(total\_loyalty\_points AS FLOAT) / (SELECT SUM(total\_loyalty\_points) FROM #leaderboard ) \* 50000,2) AS bonus

FROM #leaderboard

Part C

Q.Would you say the loyalty point formula is fair or unfair ?

SOLUTION :

1. First of all The fairness of the loyalty point formula depends on the company's goals and how well it incentivizes desired player behaviors.

but from the analysis i have observed that the users who has deposit lot of times are getting

more weightage while calculating loyalty points.

Players who spend more are rewarded more, but this might disproportionately benefit high spenders over more consistent, active players.

1. From the top 50 players the users with highest loyalty points are the users who made lot of deposit transaction and they played games very little.

MEANS Players with high deposits will likely dominate the leaderboard, even if their engagement (games played) is low.

Q. Can you suggest any way to make the loyalty point formula more robust?

SOLUTION :

* I will suggest to do some changes in loyalty point formula .
* There is some changes needs to done in weightage to give all the players same advantage.
* Like Decrease the weight of deposit points and Increase the weight of games played in the loyalty point calculation to better reward consistent engagement.

PERSONAL NOTE :

Rohit Bhavikatte

rohitbhavikatte2@gmail.com

8329102082