

Assignment: Decision making

Task:

- Clients complain that the searches for a destinations sometimes fail. Head of product decided to address the issue, and ask development team to work on fix.
- The team committed to work out the solution during August. It was agreed that the team's bonus payout would depend on effectiveness of the solution.
- The Head of Product ask you to analyze the data and help him to decide whether the team deserve the bonus?

UNDERSTANDING DATA

We calculate some diferent functions

August:

```
SELECT
    sum(users) as Total_Users,
    sum(Queries_correct) as Total_correct,
    sum(Queries_error) as Total_Error,
    max(Queries_error) as Max_error,
    min(Queries_error) as Min_error,
    avg(Queries_error) as Avg_error,
    max(Queries_correct) as Max_correct,
```

```
min(Queries_correct) as Min_correct,  
avg(Queries_correct) as Avg_correct  
  
From  
  
Agosto_2018__Assignment_2
```

Result

Total_Users= 841119

Total_correct= 834060

Total_Error = 12828

Max_error = 142

Min_error = 0

Avg_error = 4.5409

Max_correct = 896

Min_correct = 7

Avg_correct = 295.2425

September:

```
SELECT  
  
sum(users) as Total_Users,  
sum(Queries_correct) as Total_correct,  
sum(Queries_error) as Total_Error,  
max(Queries_error) as Max_error,  
min(Queries_error) as Min_error,  
avg(Queries_error) as Avg_error,  
max(Queries_correct) as Max_correct,  
min(Queries_correct) as Min_correct,
```

```
avg(Queries_correct) as Avg_correct
From
Setiembre_2018__Assigment_2
```

Total_Users= 866333

Total_correct= 858740

Total_Error = 8950

Max_error = 38

Min_error = 0

Avg_error = 3.7684

Max_correct = 1160

Min_correct = 10

Avg_correct = 361.5747

We can see that *Queries_error* in August month there are some extremely high values. We will **assume** that in that day or hours they had a problem with the server and we will exclude them of statistical studies below.

EXPLORING DATA

We would like to know the difference between how many users and *query_errors* occurred in August and how many users and *query_errors* occurred in September

```
SELECT
    ROUND((sum(A.queries_error)/sum(A.users))*100,2) AS '%
Errors_August',
```

```
ROUND((sum(A.queries_correct)/sum(A.users))*100,2) AS
'%Correct_August',
ROUND((sum(S.queries_error)/sum(S.users))*100,2) AS '%
Errors_September',
ROUND((sum(S.queries_correct)/sum(S.users))*100,2) AS
'%Correct_September'
FROM
Agosto_2018__Assigment_2 AS A inner join Setiembre_201
8__Assigment_2 as S
WHERE
NOT A.timestamp BETWEEN '18/08/2018 11:30' and '18/08/
2018 15:30'
```

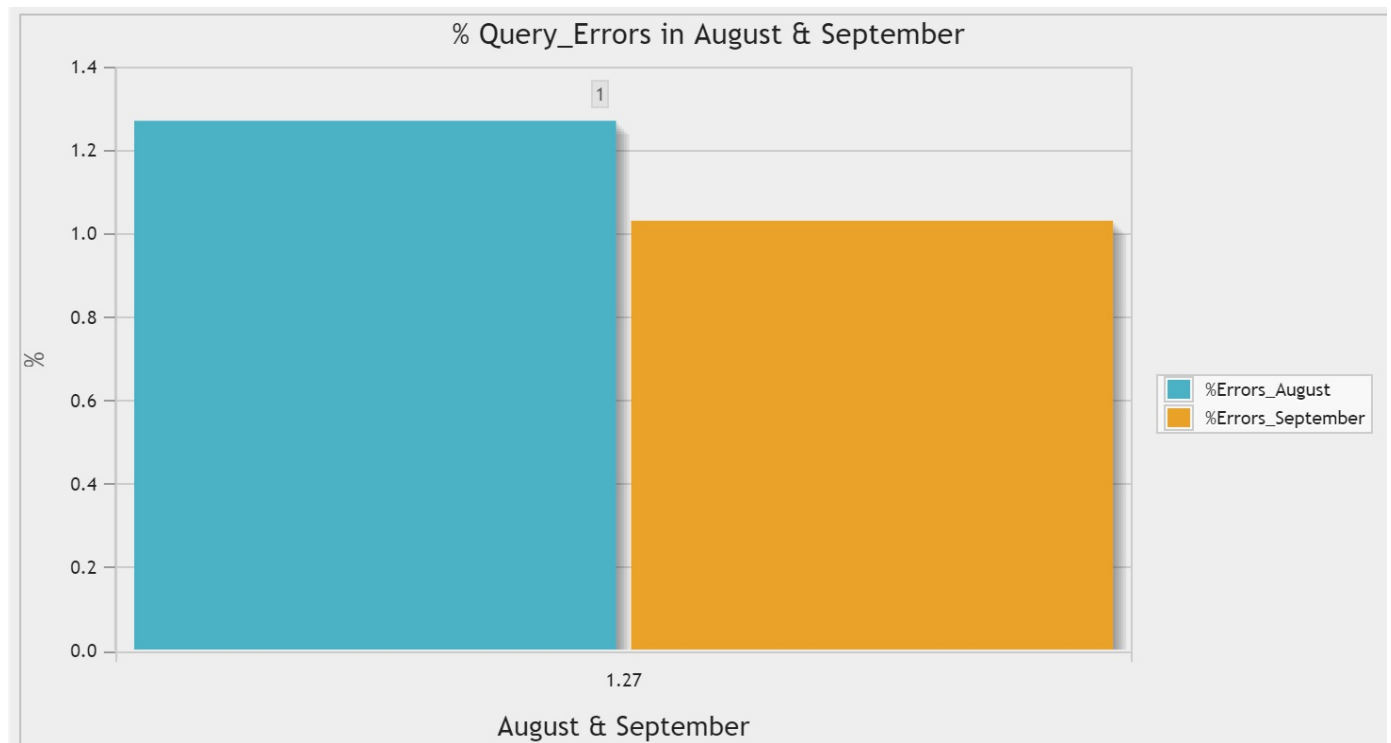
Result

%Errors_August: 1.27%

%Correct_August: 98.9%

%Errors_September: 1.03%

%Correct_September: 99.12%



As we can see above, in August where more Query_errors than in September. It seems that the developers could have fixed some issues.

Let's see now how is the evolution during the month

We will show how many **Query_errors** per day have occurred during August month comparing with September Month and related to how many users has been conected.

```
SELECT
```

```
    sum(a.users) AS Users_A,
```

```
    sum(a.queries_error) AS Error_August,
```

```
    (sum(a.queries_error)/sum(a.users)*100) AS '% August',
```

```
    sum(s.users) AS Users_S,
```

```
    sum(s.queries_error) AS Error_September,
```

```
    (sum(s.queries_error)/sum(s.users)*100) AS '% Septembe
```

r',

LEFT(a.timestamp,2) AS DAY_August,

LEFT(s.timestamp,2) AS DAY_September

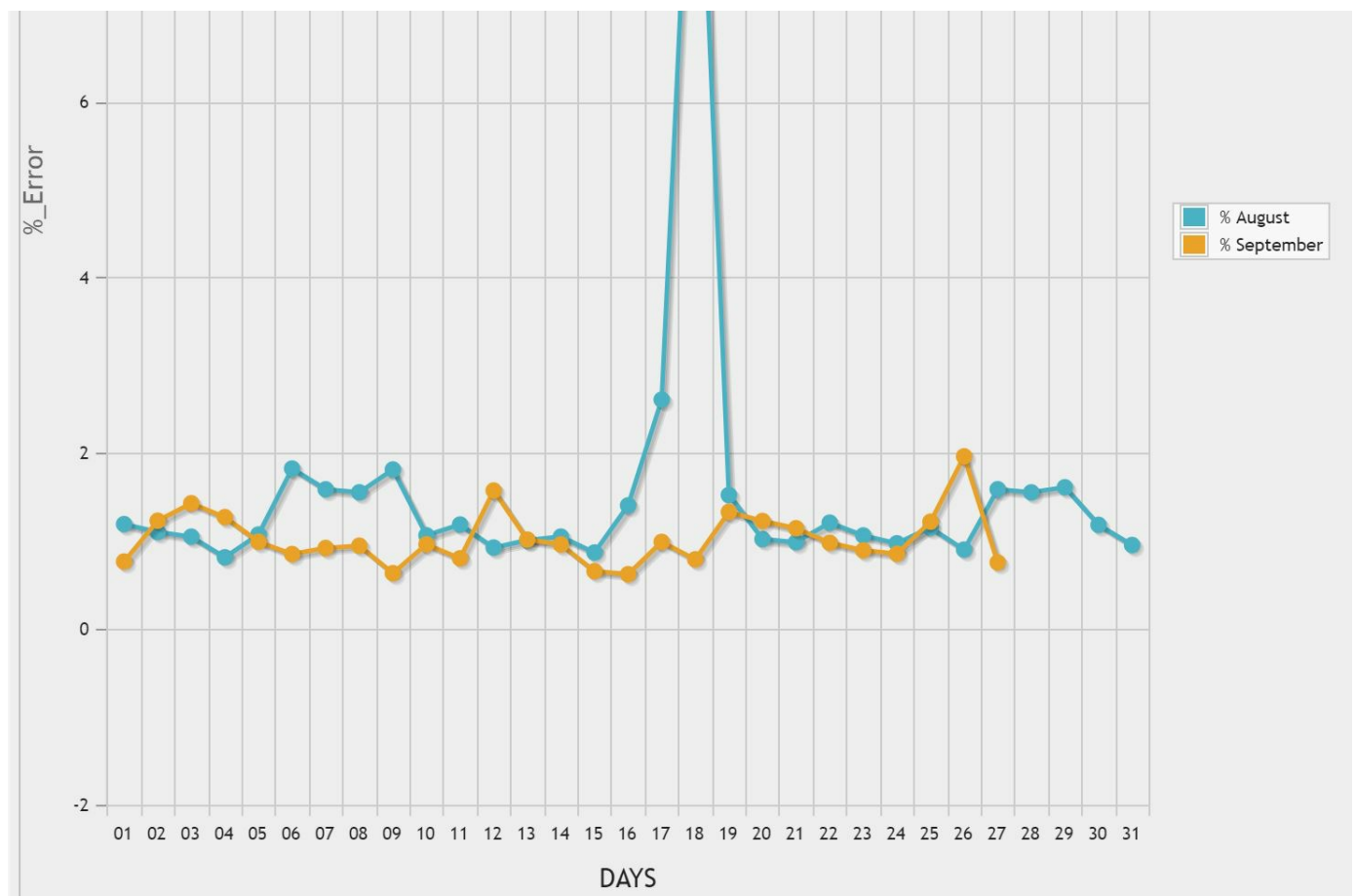
FROM

Agosto_2018__Assigment_2 AS a LEFT JOIN Setiembre_2018

__Assigment_2 AS s ON LEFT(a.timestamp,2)=LEFT(s.timestamp
,2)

GROUP BY

DAY_August



As we can see above in September month sometimes queries_error where higher than in August