Retailer App Case Study

- Our Plantix Partner app allows Retailers to order supplies online. We are continuously working on optimizing the app.
- We have 3 tables attached.
- 1st table is login_logs contains information about users login in our site.
- 2nd table is sales_orders contains information about orders.
- 3rd table is sales_orders_items contains information about each order.
- We have data of July 2021 and July 2022.

Objectives:

To optimise the uses of Retailer app.

To provide suggestions for improving the business of our organization on the basis of findings

Work on data:

Following KPI to measure the performance of our app.

- monthly active users(MAU)
- daily active users
- daily sessions per active users
- cutomer life time value
- retained customer
- Total users by year
- churning rate

A. Monthly active users(MAU):

Query 1:

```
select month(login_time) as month,count(id) as Monthly_active_user, year(login_time) as year
from login_logs
group by month,year;
```

Output:

	month	Monthly_active_user	year
▶	7	271240	2021
	7	395117	2022
	0	13630	0

B . Daily active users(DAU):

Query 2:

```
select date(login_time) as date,count(distinct id) as daily_active_users
from login_logs group by date ;
```

	date	daily_active_users
▶	0000-00-00	4832
	2021-07-01	1563
	2021-07-02	1519
	2021-07-03	1485
	2021-07-04	1128

C .Daily sessions per active users:

Query 3:

```
select id,date(login_time) as date,count(id) as count_of_sessions
from login_logs group by date,id  order by count_of_sessions desc;
```

Output:

	id	date	count_of_sessions
▶	63652	2021-07-02	247
	114036	2021-07-23	212
	63652	2021-07-15	211
	63652	2021-07-07	210
	209276	2022-07-23	209

D. Customer life time value:

Query 4:

```
select fk_buyer_id,order_quantity_accepted*rate as customer_lifetime_purchase
from sales_orders as a
join sales_orders_items as b on
b.fk_order_id = a.order_id
group by fk_buyer_id
order by FK_buyer_id asc;
```

Output:

	fk_buyer_id	customer_lifetime_purchase
▶	996	12288
	1008	31260
	1016	3530.43994140625
	1032	0
	1088	17415

E. Retained customer:

Query 5:

```
select year(login_time) as year,count(distinct(id)) as retained_customer
from login_logs
where year(login_time) = '2022' and
id in
( select distinct(id) from login_logs where year(login_time) = '2021');
```

Output:

	year	retained_customer
▶	2022	3607

F. Total users by year:

Query 6:

```
select year(login_time) as year,count(distinct(id)) as users_in_year
from login_logs
group by year;
```

Output:

	year	users_in_year
▶	0	4832
	2021	10867
	2022	13022

G. Churn user :

Query 7:

```
select(
(select count(distinct id) as total_user  from login_logs where year(login_time)= '2021')-
( select count(distinct id) as retain_users from login_logs where year(login_time) = '2022' and id in
(select distinct id from login_logs where year(login_time)='2021'))) as churn_users
```

Output:

	churn_users
▶	7260

- Our 7260 user are churned in july2022.

2. Prepare a report regarding our growth between the 2 years.

A. Did our business grow?

Query:

```
select year(creation_time) as year , round(sum(order_quantity_accepted*rate)) as total_revenue
from sales_orders as a
join sales_orders_items as b on
b.fk_order_id = a.order_id
group by year(creation_time);
```

Output:

year	total_revenue
2021	60315971
2022	116858178

B. Does our app perform better now?

- **sales order performance**

Query:

```
select distinct count(sales_order_status) as Total_orders,year(creation_time),sales_order_status
from sales_orders
group by year(creation_time), sales_order_status;
```

Output:

Total_orders	year(creation_time)	sales_order_status
3764	2021	Rejected
2434	2021	Shipped
18	2021	Pending
3899	2022	Rejected
3489	2022	Shipped
25	2022	Pending
1	2022	Review

- Rejection rate of order is increase by 3.58% in 2022.
- Shipping rate of order is increase by 43.34% in 2022.
- Pending rate order is increased by 38.88% in 2022.

C. Did our user base grow?

Query:

```
select count(id) as Total_users,year(login_time) as year
from login_logs
group by year(login_time);
```

Output:

	Total_users	year
▶	271240	2021
	395117	2022
	13630	0

- Our user base is grow by 45.67% in 2022.

3. What are our top-selling products in each of the two years? Can you draw some insight from this?

Top selling products 2021:

Query:

```
select fk_product_id,count(*) as Order_Quentity,year(creation_time) as year
from sales_orders_items as a
join sales_orders as b on
a.fk_order_id = b.order_id
where year(creation_time) = 2021
group by fk_product_id,year(creation_time) order by year(creation_time),count(*) desc
limit 10
;
```

Output:

We showing only top 10 selling products in 2021.

	fk_product_id	Order_Quentity	year
▶	10235	550	2021
	8444	337	2021
	1548	300	2021
	1041	268	2021
	9925	249	2021
	8425	245	2021
	10975	212	2021
	8219	204	2021
	8210	187	2021

Result 13 ×

Top selling products 2022:

Query:

```
select fk_product_id,count(*) as Order_Quentity,year(creation_time) as year
from sales_orders_items as a
join sales_orders as b on
a.fk_order_id = b.order_id
where year(creation_time) = 2022
group by fk_product_id,year(creation_time) order by year(creation_time),count(*) desc
limit 10
;
```

Output:

We showing only top 10 selling products in 2022.

fk_product_id	Order_Quentity	year
8444	485	2022
10235	358	2022
1041	356	2022
8425	345	2022
3610	335	2022
10975	257	2022
10224	236	2022
1038	225	2022

4. Looking at July 2021 data, what do you think is our biggest problem and how would you recommend fixing it?

Query:

```
select distinct month(creation_time) as month,sales_order_status,count(sales_order_status) as Total_orders
from sales_orders where month(creation_time) = 07 and year(creation_time) = 2021
group by month(creation_time), sales_order_status;
```

Output:

month	sales_order_status	Total_orders
7	Rejected	3764
7	Shipped	2434
7	Pending	18

- Rejection of order is very biggest problem of our organization.
- We need to work on reduce rate of rejection of order .

5. Does the login frequency affect the number of orders made?

Query 1:

Login Frequency of users.

```
select year(login_time) as year,count(*) as login_frequency_of_users
from login_logs
group by year;
```

Output:

year	login_frequency_of_users
2021	271240
2022	395117
0	13630

Query 2:

Orders quantity per year.

```
select year(creation_time) as year,count(*) as Order_Quentity
from sales_orders_items as a
join sales_orders as b on
a.fk_order_id = b.order_id
group by year(creation_time) order by year(creation_time),count(*) desc
;
```

Output:

year	Order_Quentity
2021	9826
2022	10662

year	login_frequency_of_users
2021	271240
2022	395117

- Login frequency in 2021 is 271240 and order_quantity is 9826.

  So percentage of order is **3.622** in 2021.

- Login frequency in 2022 is 395117 and order_quantity is 10662.
  So percentage of order is **2.698** in 2021.

Conclusion and suggestions:

We have worked on optimisation of the usage of our partner Retailer App and providing suggestions for improving the business on basis of Findings:

- We have 13022 users in 2022 and 10867 users in 2021 with retentions rate of 33% and Churning rate of 67%. So, we need to increase retention rate.
- Our daily active users and daily users who orders changing frequently good thing is it increasing trend, We need steadilly increase it.
- Our business has grown by 94% and user base growth by 46% with better performance of our app, we were able decrease the no of rejected orders.
- 8444, 10235, 1041, 8425, 10975 are top selling products.
- Our biggest problem in 2021 was number rejected items and we are improving it, which we have achieved in 2022.
- Login frequency is not that much affecting the number of orders made.