

Retail analysis project

Submitted by Sumit Mathur and Anurag Sharma

-- ----- 1ST ANSWER -----

```
select user_id, count(*) as Numbers_of_login, login_time from login_logs
group by user_id, login_time
order by Numbers_of_login desc;
```

-- ----- 2nd ANSWER -----

-- Daily trend of logins and trend of conversion rate (Number of orders placed per login)--

```
select USER_ID, A.login_time, sum(order_id)/sum(user_id) as conversion_rate from login_logs as A
inner join sales_orders as b
on A.user_id = b.fk_buyer_id
group by user_id, A.login_time
order by user_id;
```

-- ----- 3RD ANSWER -----

-- KPIs of the app will be active user,retention rate, UPT avg of unit per transction ,revenue through logins---

-- ---Active users daily -----

```
SELECT user_id,COUNT(*) AS num_logins, day(login_time) as days
FROM login_logs
where year(login_time) = 2021
GROUP BY day(login_time),user_id
order by num_logins desc;
```

-- final answer -- retention rate -- MEANS CUSTOMERS INITIAL VISIT AFTER THEIR FIRST VISIT

```
SELECT user_id,(COUNT(user_id) * 100.0) / (SELECT COUNT(DISTINCT user_id) FROM login_logs) as Retention_rate FROM login_logs
```

```
GROUP BY user_id
```

```
HAVING COUNT(user_id) > 1;
```

-- - Revenue generated through app ---

```
select A.user_id, cast(sum(C.rate) as int) as Total_Revenue from login_logs as A JOIN sales_orders as B
```

```
on A.user_id = b.fk_buyer_id
```

```
join sales_orders_items as C
```

```
on B.order_id = C.fk_order_id
```

```
where c.order_quantity_accepted > 0
```

```
group by a.user_id
```

```
order by a.user_id;
```

-----3RD ANSWER-----

-----Did our business grow?-----

```
select cast(sum(C.rate)as bigint) as Total_Revenue , 'Revenue in 2021' as "yearly"
```

```
from login_logs as A
```

```
JOIN sales_orders as B
```

```
on A.user_id = b.fk_buyer_id
```

```
join sales_orders_items as C
```

```
on B.order_id = C.fk_order_id
```

```
where c.order_quantity_accepted > 0 and year(login_time) = 2021
```

```
union
```

```
select cast(sum(C.rate)as bigint) as Total_Revenue , 'Revenue in 2022' as "yearly"
```

```
from login_logs as A
```

```

JOIN sales_orders as B
on A.user_id = b.fk_buyer_id
join sales_orders_items as C
on B.order_id = C.fk_order_id
where c.order_quantity_accepted > 0 and year(login_time) = 2022

```

```

-- ---- Does our app perform better now?-----
-----

```

```

select * from login_logs;

```

```

select * from sales_orders;

```

```

select * from sales_orders_items;

```

```

select USER_ID, count(abs(datediff(WEEK,creation_time,login_time))) as hours_taken_to_orders
from login_logs as A join
sales_orders as B on a.user_id = b.fk_buyer_id
join sales_orders_items as c on
c.fk_order_id = b.order_id
where order_quantity_accepted >0 and year(login_time) = 2022
group by user_id;

```

```

-- ----Did our user base grow?-----
-----

```

```

select count(*) As number_of_users,'number of customers in 2021' as "yearly data" from login_logs
where year(login_time) = 2021
-- total users 2,71,240 in 2021 ---
union
select count(*) As numbers_of_users,'number of customers in 2022' as "yearly data" from login_logs

```

```
where year(login_time) = 2022;  
-- total users 3,95,117 in 2022 ----
```

-- ---4. What are our top-selling products in each of the two years? Can you draw some insight from this?-----

```
select count(order_quantity_accepted) Total_number_of_orders, c.fk_product_id, cast(c.rate as int) as  
amount, year(b.creation_time) as yrs  
from sales_orders_items c join sales_orders b  
on c.fk_order_id = b.order_id  
group by c.fk_product_id,c.rate, year(b.creation_time)  
order by Total_number_of_orders desc , yrs ;
```

-- ---5. Looking at July 2021 data, what do you think is our biggest problem and how would you recommend fixing it?---

```
select count(*) as orders_count,'rejected' as "2021" from sales_orders  
where sales_order_status = 'rejected' and year(creation_time) = 2021
```

union

```
select count(*) as shipped,'shipped' as "2021" from sales_orders  
where sales_order_status = 'shipped' and year(creation_time) = 2021;
```

-- As per our analysis In july 2021 we have many more rejected sales_order_status as compared to shipped orders,

-- for fixing this we can increase the stock for the specific products and make shipped more than rejected.--

---6. Does the login frequency affect the number of orders made?-----

```
select count(user_id) as total_users_in_2021,sum(c.order_quantity_accepted) as total_orders,'number of user in 2021'as '2021' from login_logs as A JOIN sales_orders as B
```

```
on A.user_id = b.fk_buyer_id
```

```
join sales_orders_items as C
```

```
on B.order_id = C.fk_order_id
```

```
where year(login_time) = 2021 and c.order_quantity_accepted > 0
```

```
union
```

```
select count(user_id) as total_users_in_2022,sum(c.order_quantity_accepted) as total_orders , 'number of user in 2022'as '2022' from login_logs as A JOIN sales_orders as B
```

```
on A.user_id = b.fk_buyer_id
```

```
join sales_orders_items as C
```

```
on B.order_id = C.fk_order_id
```

```
where year(login_time) = 2022 and c.order_quantity_accepted > 0
```

-----Give at least 2 insights that are not mentioned above and are not clearly visible from the data.--

--- 2021 login was low and revenue was also low and

--- 2022 login was high and revenue was high just because login was got increased by login time