

# Dastgyr Technologies

## Data Analysis Case Study

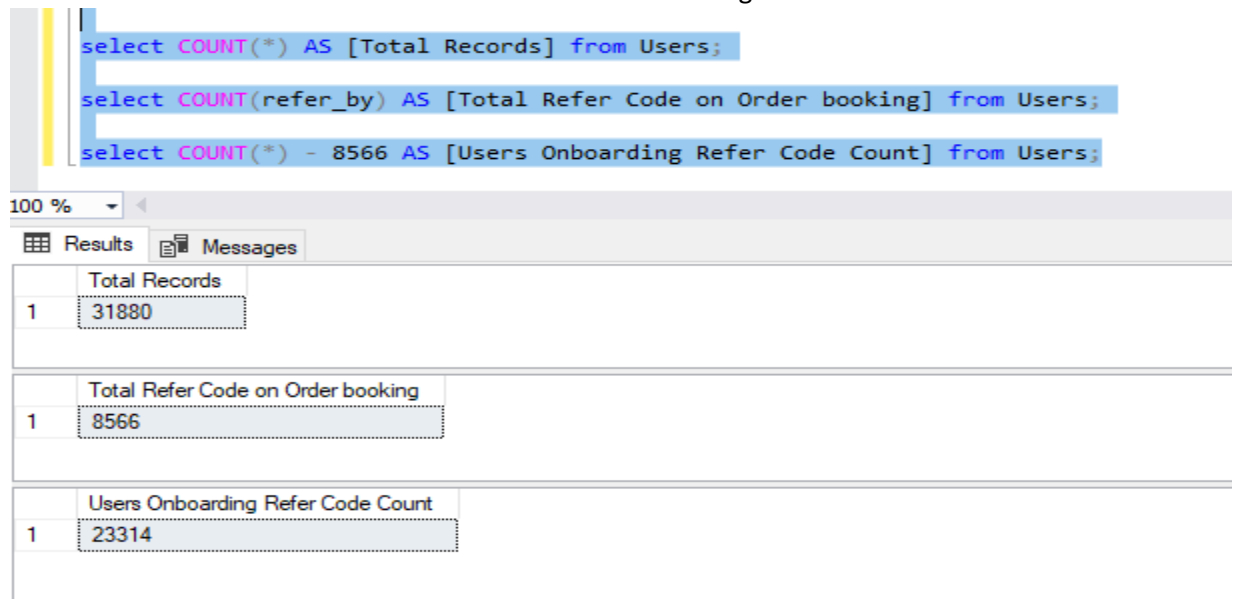
Q1.

Ans: **Approach:**

Order booking and onboarding process are two different things because order booking generates the revenue and also defines our agent performance while onboarding process is how much the retailers onboard through sales agent. For leading the team in both I have used performance metrics. We make 5 groups for onboarding and order booking. We select the leader of team based on how much people used his referral code.

First we will check how many number of refer code used in order booking and onboarding process.

1. Check count of users uses refer code on order and onboarding refer code count.



```
select COUNT(*) AS [Total Records] from Users;
select COUNT(refer_by) AS [Total Refer Code on Order booking] from Users;
select COUNT(*) - 8566 AS [Users Onboarding Refer Code Count] from Users;
```

Total Records	
1	31880

Total Refer Code on Order booking	
1	8566

Users Onboarding Refer Code Count	
1	23314

There are total **31,880** records out of which **8,566 user** uses refer code on order booking and **23,324 user** uses refer code on onboarding process.

**Note: Count function not include null values when counting total records.**

## 2. Make 5 teams By joining two tables users and orders

```
CREATE VIEW Orderbooking  
AS  
SELECT Users.users_id, Users.name, Users.created_at, Users.refer_by, Users.refer_at, Users.ref_code  
FROM Orders  
INNER JOIN Users  
ON Users.users_id = Orders.users_id;  
GO  
select * from Orderbooking;
```

Make 5 groups and assigned group number.

```
GO  
select *, ntile(5) over(order by users_id) AS [GroupNo] from Orderbooking;
```

100 %							
Results Messages							
	users_id	name	created_at	refer_by	refer_at	ref_code	GroupNo
1	17	Shazad	2020-02-29	NULL	NULL	NULL	1
2	17	Shazad	2020-02-29	NULL	NULL	NULL	1
3	17	Shazad	2020-02-29	NULL	NULL	NULL	1
4	17	Shazad	2020-02-29	NULL	NULL	NULL	1
5	18	saif	2020-02-29	NULL	NULL	NULL	1
6	18	saif	2020-02-29	NULL	NULL	NULL	1
7	18	saif	2020-02-29	NULL	NULL	NULL	1
8	19	U A	2020-02-29	NULL	NULL	NULL	1

## 3. Get the Leader of each Group:

```
SELECT refer_by, count(refer_by) AS [COUNT], avg(GroupNo) as GroupNo  
FROM OrderGroup  
GROUP BY refer_by  
ORDER BY GroupNo DESC;
```

100 %			
Results Messages			
	refer_by	COUNT	GroupNo
14	51496	167	5
34	46908	1115	4
47	16781	1296	3
66	18871	1133	2
85	11312	279	1
86	29465	51	1
87	11922	11	1
88	21082	83	1

Refer – By sales agent Id  
Count – Total count where sales agent id appears.  
Group No – Each Group Number

These are each group's top sales representatives. Most clients use these agent referral codes to place their orders.

**Q2.**

**Ans: Orders get cancelled after they are dispatched:**

After create the table, first we will analyze the total number of cancelled order. So the total number of cancelled order count is **8117**. After that we have to find out the issue of order cancellation.

```
create table CancellationReasonList(  
    order_id int not null,  
    cancellation_reason_id int,  
    cancel_reason_text VARCHAR(100),  
);  
  
BULK INSERT CancellationReasonList  
FROM 'E:\Dastgyr\Data\Cancellation_Reasons_List.csv'  
WITH (  
    FIRSTROW = 2,  
    FIELDTERMINATOR = ',',  
    ROWTERMINATOR = '\n'  
)  
GO  
  
select * from CancellationReasonList;
```

100 %

Results Messages

	order_id	cancellation_reason_id	cancel_reason_text
1	188424	60	NULL
2	188420	59	NULL
3	188401	51	NULL
4	188401	53	NULL
5	188396	53	NULL
6	188367	30	NULL
7	188366	30	NULL
8	188364	55	NULL
9	188354	49	NULL
10	188354	50	NULL

### Total Cancelled Orders:

```
select COUNT(*) AS [Total Cancelled Order] from Orders where status = 6;
```

Total Cancelled Order	
1	8117

### Reasons For Cancelling the Orders:

```
SELECT cancel_reason_text, count(cancel_reason_text) AS [COUNT]
FROM CancellationReasonList
Group BY cancel_reason_text
ORDER BY [COUNT] DESC;
```

cancel_reason_text	COUNT
1 Shop Close	475
2 no service area	435
3 cash issue	325
4 order cancel road block	207
5 Network issue	172
6 order cancel payment issue	158
7 No Go Area	133
8 shop closed	114
9 Road Block	97
10 Wrong location	94
11 incomplete order	83
12 order cancel shop close	81
13 order cancel location issue	70
14 location issue	62
15 Roads Blockage	54
16 Late delivery	49
17 office sy bula liya tha	48
18 Customer asking about hi...	43
19 road block issue	36

Cancel reason text – Reason why order is cancelled

Count – Number of customers faced this issue.

You can bring this down by defining each shop opening and closing timing, indicate that service is not available, estimated delivery time for road block issue and etc. For payment issue there are multiple strategies like online payment integration or advanced payment.

Q3.

Ans. Incomplete orders due to some SKUs not being in stock:

```
create table SKU(  
ID int not null,  
ProductID int,  
Price int,  
Discount float,  
Stock int,  
IsActive VARCHAR(40),  
IsStock VARCHAR(40),  
Weight VARCHAR(40),  
Unit VARCHAR(40),  
AllTypes VARCHAR(40),  
IsDeal VARCHAR(40)  
);  
  
BULK INSERT SKU  
FROM 'E:\Dastgyr\Data\SKU.csv'  
WITH (
```

100 %

Results Messages

	ID	ProductID	Price	Discount	Stock	IsActive	IsStock	Weight	Unit	AllTypes	IsDeal
1	87	84	720	0	0	TRUE	TRUE	NULL	NULL	NULL	FALSE
2	3188	388	1875	0	0	FALSE	FALSE	NULL	NULL	NULL	FALSE
3	2838	388	3750	0	0	FALSE	TRUE	NULL	NULL	NULL	FALSE
4	2479	84	383	5	0	TRUE	FALSE	NULL	NULL	NULL	TRUE
5	2478	84	383	5	0	TRUE	FALSE	NULL	NULL	NULL	TRUE
6	2477	84	383	0	0	TRUE	FALSE	NULL	NULL	NULL	TRUE
7	2459	142	4800	10	0	FALSE	FALSE	NULL	NULL	NULL	TRUE
8	3190	92	2115	0	0	FALSE	TRUE	NULL	NULL	NULL	FALSE
9	2452	133	7950	25	0	TRUE	TRUE	NULL	NULL	NULL	TRUE
10	3286	428	3750	0	0	FALSE	FALSE	NULL	NULL	NULL	FALSE

```
select ID, ProductID, CategoryID, stock from SKUProducts  
ORDER BY Stock;
```

100 %

Results Messages

	ID	ProductID	CategoryID	stock
1	63	776	114	-2814
2	63	776	167	-2814
3	1614	772	114	-2255
4	1614	772	57	-2255
5	96	774	114	-1496
6	96	774	115	-1496
7	1610	811	166	-1053
8	1610	811	179	-1053
9	1015	781	116	-963
10	1015	781	135	-963
11	687	811	166	-806

✓ Query executed successfully.

First we have joined the SKU and product categories table. After combining, the table is sorted in order of most frequent purchases. In the image above, we will see the category ID that we have to keep in our stock and address it to the relevant category manager.