Introduction:

The data set belongs to a leading online E-Commerce company. An online retail (E commerce) company wants to know the customers who are going to churn, so accordingly they can approach customer to offer some promos.

In the fast-paced world of online retail, e-commerce companies face the challenge of retaining customers. To address this, identifying at-risk customers and implementing targeted retention strategies are crucial.

Problem Statement:

The goal of this project is to detailed analysis of a dataset from an online retail company, revealing valuable insights on customer churn. These insights offer essential guidance for decision-making, enabling proactive measures to reduce attrition and foster long-term loyalty.

Project Approach:

Dataset contains information such as customers' personal details, satisfaction scores, preferred payment mode, days since the last order, and cashback amount. I used SQL (Postgres SQL) to clean and analyze this dataset, and performed visualizations using Microsoft Power BI. This analysis is divided into several stages: data cleaning, data exploration, an insight section.

Data Dictionary:

Variable	Discription
CustomerID	Unique customer ID
Churn	Churn Flag
Tenure	Tenure of customer in organization
PreferredLoginDevice	Preferred login device of customer
CityTier	City tier
WarehouseToHome	Distance in between warehouse to home of customer
PreferredPaymentMode	Preferred payment method of customer
Gender	Gender of customer
HourSpendOnApp	Number of hours spend on mobile application or website
NumberOfDeviceRegistered	Total number of deceives is registered on particular customer
PreferedOrderCat	Preferred order category of customer in last month
SatisfactionScore	Satisfactory score of customers on service
MaritalStatus	Marital status of customer
NumberOfAddress	Total number of added on particular customer
Complain	Any complaint has been raised in last month
OrderAmountHikeFromlastYear	Percentage increases in order from last year
CouponUsed	Total number of coupons has been used in last month
OrderCount	Total number of orders has been places in last month
DaySinceLastOrder	Day Since last order by customer
CashbackAmount	Average cashback in last month

Data Cleaning:

Before going to analysis, it is essential to ensure the dataset is clean and reliable. The data cleaning process involves handling missing values, correcting inconsistencies, and formatting the data for analysis. In this project, we carefully cleaned the dataset to ensure the accuracy and integrity of our findings.

(1) Creation of Customer_churn Table

```
Query Query History

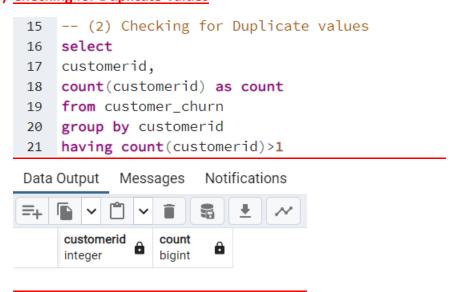
1 -- Creation customer_churn Table
2 CREATE TABLE customer_churn(
3 CustomerID integer, Churn integer, Tenure integer, PreferredLoginDevice text, CityTier integer,
4 WarehouseToHome integer, PreferredPaymentMode text, Gender text, HourSpendOnApp integer,
5 NumberOfDeviceRegistered integer, PreferedOrderCat text, SatisfactionScore text,MaritalStatus text,
6 NumberOfAddress integer, Complain integer, OrderAmountHikeFromlastYear integer, CouponUsed integer,
7 OrderCount integer, DaySinceLastOrder integer, CashbackAmount integer
8 )
```

(2) <u>Total No. of Customers</u>



Findings: There are 5630 customers in given dataset

(3) Checking for Duplicate Values



Findings: The result showing empty table that means there is no duplicate values in the dataset

(4) Checking for Null Values

```
24 -- (3) Checking for Null values
25
   select 'churn' as column_name, count(*) as NullCount
26
   from customer_churn
27 where churn is null
   select 'tenure' as columns_name, count(*) as NullCount
29
   from customer_churn
31
   where tenure is null
32 union
33 select 'PreferredLoginDevice' as column_name, count(*) as NullCount
34 from customer_churn
   where PreferredLoginDevice is null
   union
36
37 select 'CityTier' as column_name, count(*) as NullCount
38 from customer_churn
   where CityTier is null
39
   union
40
   select 'WarehouseToHome' as column_name, count(*) as NullCount
42 from customer_churn
43
   where WarehouseToHome is null
44
   select 'PreferredPaymentMode' as column_name, count(*) as NullCount
   from customer_churn
47
   where PreferredPaymentMode is null
48
   select 'Gender' as column_name, count(*) as NullCount
49
   from customer_churn
51 where Gender is null
52 union
52 union
53 select 'HourSpendOnApp' as column_name, count(*) as NullCount
54 from customer_churn
55 where HourSpendOnApp is null
57 select 'NumberOfDeviceRegistered' as column_name, count(*) as NullCount
58
   from customer_churn
59 where NumberOfDeviceRegistered is null
60 union
   select 'PreferedOrderCat' as column_name, count(*) as NullCount
62 from customer_churn
63 where PreferedOrderCat is null
64 union
65 select 'SatisfactionScore' as column_name, count(*) as NullCount
66 from customer_churn
67 where SatisfactionScore is null
68 union
69 select 'MaritalStatus' as column_name, count(*) as NullCount
70 from customer_churn
```

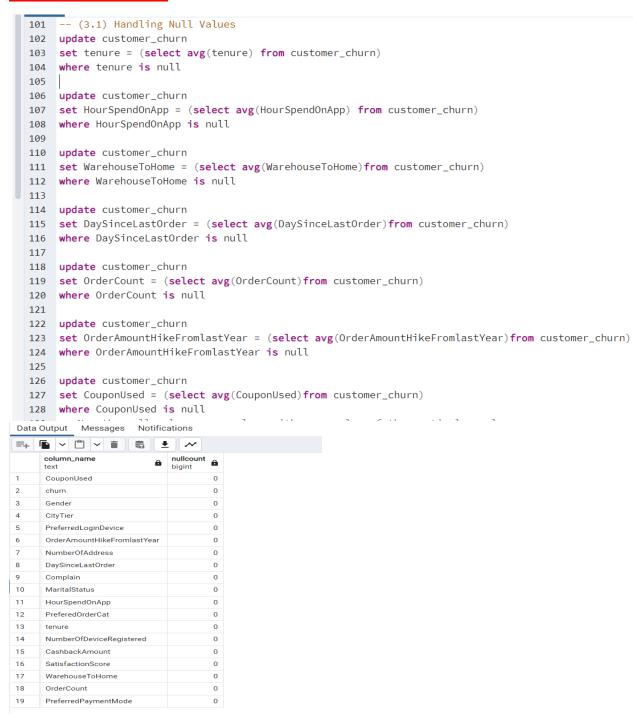
71 where MaritalStatus is null

72 union

```
72
 73
     select 'NumberOfAddress' as column_name, count(*) as NullCount
 74
     from customer_churn
     where NumberOfAddress is null
 75
 76 union
 77
     select 'Complain' as columns_name, count(*) as NullCount
 78 from customer_churn
 79 where Complain is null
 80 union
 81 select 'OrderAmountHikeFromlastYear' as columns_name, count(*) as NullCount
 82 from customer churn
 83 where OrderAmountHikeFromlastYear is null
 84
     select 'CouponUsed' as column_name, count(*) as NullCount
 85
     from customer_churn
 87
     where CouponUsed is null
 88 union
 89 select 'OrderCount' as column_name, count(*) as NullCount
 90 from customer churn
 91 where OrderCount is null
 93 select 'DaySinceLastOrder' as column_name, count(*) as NullCount
 94 from customer_churn
 95 where DaySinceLastOrder is null
 96 union
 97
    select 'CashbackAmount' as column_name, count(*) as NullCount
 98
     from customer_churn
 99
     where CashbackAmount is null
100
Data Output Messages
                         Graph Visualiser ×
                                             Notifications
     •
     column_name
                                nullcount
                                        0
                                bigint
      PreferredPaymentMode
2
                                      264
3
      HourSpendOnApp
4
      WarehouseToHome
                                      251
5
                                        0
6
                                        0
      Gender
7
      CityTier
                                        0
      PreferredLoginDevice
                                        0
8
9
      DaySinceLastOrder
                                      307
10
                                        0
      NumberOfAddress
11
      OrderCount
                                      258
12
      OrderAmountHikeFromlastYear
                                      265
13
      Complain
                                        0
14
      MaritalStatus
                                        0
15
      CouponUsed
                                      256
16
      PreferedOrderCat
                                        0
17
      NumberOfDeviceRegistered
                                        0
                                        0
18
      CashbackAmount
19
      SatisfactionScore
                                        0
```

<u>Findings:</u> CouponUsed, DaysSinceLastOrder, HourSpendOnApp, OrderAmountHikeFromLastYear, OrderCount, Tenure, and WarehouseToHome all have null values present, and the number of null values present for each column can be seen in the above table.

(4) Handling Null Values



Findings: so, there is no null values in the dataset

(4) Creating new column for an already existing column "Churn"

we can observe the "churn" column contained 0 and 1 values.o means that customer did not churn, while 1 means that customer churned.it is difficult to remember this, i we will create a new column called "customer status" that shows "stayed" or "churned



<u>Findings:</u> The new column "customer_status" has affected in table.it has two distinct values "churned", "stayed".

(5) Creating new column for an already existing column "complain"

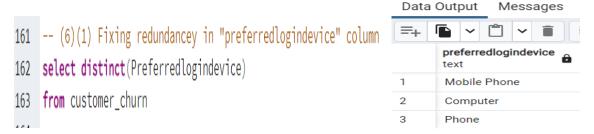
we noticed that the complaint column also contained 0 and 1. '0' means that the customer did not record any complaints, while '1' means the customer recorded a complaint. For clarity purposes, we will create a new column called 'complain_recieved' that shows "No" when a customer did not complain and "Yes" when a customer complained.



<u>Findings:</u> the new columns complaint_received affected in the database and it has two distinct values "yes", "no".

(6) Checking Redundancies in each column

(6.1) Checking redundancies in "PreferredLoginDevice" column



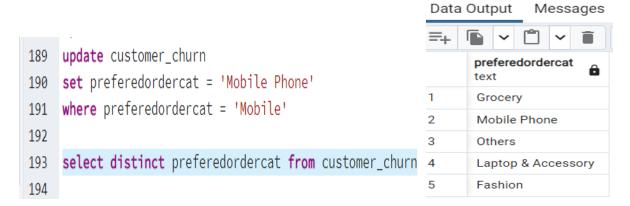
<u>Findings:</u> I have observed phone and mobile phone in the same column, but they mean same thing. so, we will replace the mobile phone with phone.



(6.2) Checking redundancies in "preferedordercat" column

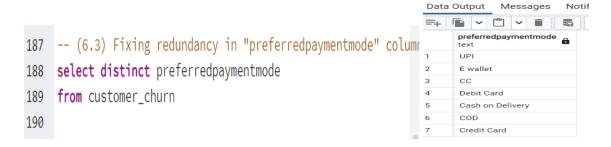


<u>Findings:</u> I have observed mobile phone and phone appear in the column, but their meaning is same.so we will replace the phone with mobile phone.



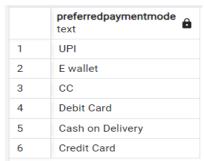
Findings: I have observed Finally redundancy in "preferedordecat" column has been fixed

(6.3) Checking redundancies in "preferredpaymentmode" column



<u>Findings:</u> I have observed "Cash on Delivery" and "COD" both appear in same column, but their meaning is same. So we will replace COD with Cash on Delivery.

```
199    update customer_churn
200    set preferredpaymentmode = 'Cash on Delivery'
201    where preferredpaymentmode = 'COD'
202
203    select distinct preferredpaymentmode from customer_churn
```



Findings: I have observed the redundancy in "preferredpaymentmode" has been fixed.

(6.4) Fixing wrongly entered values in "warehousetohome" column

			warehousetohome integer
		24	33
		25	13
		26	5
		27	18
		28	127
187	(6.4) Fixing wrongly entered values in "warehousetohome" column	29	16
188	select distinct warehousetohome	30	27
189	<pre>from customer_churn</pre>	31	23
190		32	126

<u>Findings:</u> I have observed 126 and 127 values. They are definitely outliers and most likely wrongly entered to fix this we will replace those values with 26 and 27 to fall within range of the values.

```
187 -- (6.4) Fixing wrongly entered values in "warehousetohome" column
                                                                                  warehousetohome
                                                                                   integer
188 select distinct warehousetohome
                                                                            23
                                                                                                22
189 from customer_churn
                                                                            24
                                                                                                33
                                                                            25
                                                                                                13
191 update customer_churn
                                                                            26
                                                                                                 5
192 set warehousetohome = '27'
                                                                            27
                                                                                                18
193 where warehousetohome = '127'
                                                                            28
                                                                                                16
194
                                                                                                27
195 update customer_churn
196 set warehousetohome = '26'
                                                                            30
                                                                                                23
                                                                            31
197 where warehousetohome = '126'
                                                                            32
                                                                                                11
198
```

Findings: I have observed 126 and 127 values are replaced with 26 and 27.

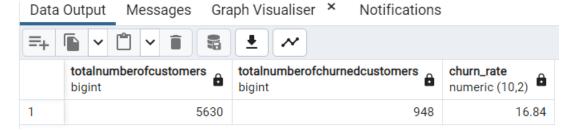
Now data cleaning is completed, so we will look into Data Exploration.

Data Exploration:

(1) The Overall customer churn rate.

```
/* Data Exploration
(1) The overall customer churn rate.

206 */
207 select Totalnumberofcustomers,
208 Totalnumberofchurnedcustomers,
209 cast((Totalnumberofchurnedcustomers*1.0/Totalnumberofcustomers*1.0)*100 as decimal(10,2)) as churn_rate
210 from
211 (select count(*) as Totalnumberofcustomers
212 from customer_churn) as total,
213 (select count(*) as Totalnumberofchurnedcustomers
214 from customer_churn
215 where customer_status = 'churned') as churned
```



Findings: The overall customer churn_rate is approx 17%.

(2) Churn Rate based on preferred login device.

```
-- (2) Churn_rate based on preferred login device

select preferredlogindevice,

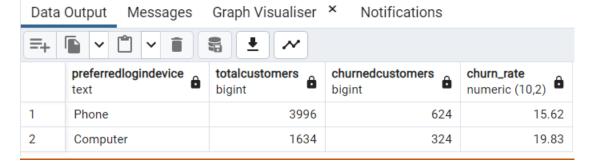
count(*) as Totalcustomers,

sum(churn) as churnedcustomers,

cast((sum(churn)*1.0/count(*))*100 as decimal(10,2)) as churn_rate

from customer_churn

group by preferredlogindevice
```



Findings:

(1) Customers who prefer login using computer have a slightly higher churn_rate compared to customers who prefer login using phone.

- (2) Appox 16% customers churned who are prefer to login using phone.
- (3) Approx 20% of the customers churned who are prefer to login using computer.
- (4) Most of the customers churned who are using computer.
- (3) Churn Rate based on different city tiers.

110000000

```
-- (3) Churn rate based on city tiers
225
226
    select
227
    citytier,
    count(*) as totalcustomer,
228
    sum(churn) as churnedcustomers,
229
    cast((sum(churn)*1.0/count(*))*100 as decimal(10,2))as churn_rate
230
    from customer_churn
231
    group by citytier
232
233 order by churn_rate
```

Data Output Messages Graph Visualiser ^ Notifications					
=+					
	citytier integer	totalcustomer bigint	churnedcustomers bigint	churn_rate numeric (10,2)	
1	1	3666	532	14.51	
2	2	242	48	19.83	
3	3	1722	368	21.37	

Cranh Vigualiaar X

Findings:

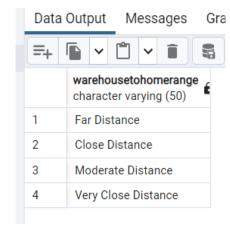
Data Output

- (1) Customers who prefer login using computer have a slightly higher churn_rate compared to customers who prefer login using phone.
- (2) Most of the cutomers churned from city tier1
- (3) City tier 1 has lower churn_rate compare to the tier 2 and tier 3 cities and city tier 3 is highest churn_rate.

(4) Churn_Rate based on warehouse to home distance.

We will create a new column called "WareHouseToHomeRange" that groups the distance into very close, close, moderate and far using CASE Statement

```
235
     -- (4) Churn_rate by WareHouseToHomeDistance
236
     /* now we will create new colum "warehousetohomerange"
    */
237
238
    alter table customer_churn
239
    add warehousetohomerange varchar(50)
240
241
    update customer_churn
242
    set warehousetohomerange =
243
244
         when warehousetohome <= 10 then 'Very Close Distance'</pre>
245
         when warehousetohome > 10 and warehousetohome <= 20 then 'Close Distance'</pre>
         when warehousetohome > 20 and warehousetohome <= 30 then 'Moderate Distance'</pre>
246
247
         when warehousetohome > 30 then 'Far Distance'
248 end
```



249
250 **select distinct** warehousetohomerange **from** customer_churn

```
252  select warehousetohomerange,
253  count(*) as Total_customers,
254  sum(churn) as Churned_customers,
255  cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as Churn_rate
256  from customer_churn
257  Group by 1
258  order by 4 desc
```

	warehousetohomerange character varying (50)	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	Very Close Distance	1969	266	13.51
2	Close Distance	2318	408	17.60
3	Moderate Distance	874	176	20.14
4	Far Distance	469	98	20.90

<u>Findings:</u> The distance between warehouse and customer's home seems to have some influence on churn_rate.

- (1) Customers who residing very close to warehouse tend to have lower churn_rate.
- (2) Customers who residing very far to warehouse tend to have higher churn_rate, so we will conclude that warehousetohome distance influence the churn_rate.

(5) Churn Rate based on preferred payment mode.

```
-- (5) Churn_rate by preferredpaymentmode
select preferredpaymentmode,
count(*) as Total_customers,
sum(churn) as Churned_customers,
cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as Churn_rate
from customer_churn
group by 1
order by 4 desc
```

	preferredpaymentmode text	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	Cash on Delivery	514	128	24.90
2	E wallet	614	140	22.80
3	CC	273	59	21.61
4	UPI	414	72	17.39
5	Debit Card	2314	356	15.38
6	Credit Card	1501	193	12.86

Findings:

- (1) Which customers had payment mode as "Cash on Delivery", those customers having higher churn rate.
- (2) Payment mode like "Credit Card" and "Debit Card" have lower churn rate.
- (3) The preferred payment mode has some influence on churn rate.

(6) Churn Rate based on Tenure.

First of all, I have created new column called "tenure_range". That groups the customers tenure into 6months, 1 year, 2 years and more than 2 years using case statement

```
269 -- (6) Churn rate by Tenure
270 /* Create new column called "tenure_range"
                                                     Data Output
                                                                     Messages
                                                                                   Not
271 */
272 alter table customer_churn
273 add tenure_range varchar(20)
274
                                                           tenure_range
275 update customer_churn
                                                           character varying (20)
276 set tenure_range =
277 case
                                                    1
                                                            2 Years
278
       when tenure <=6 then '6 Months'</pre>
279
       when tenure >6 and tenure <=12 then '1 Year'
                                                            6 Months
       when tenure >12 and tenure <=24 then '2 Years'
280
                                                    3
                                                            More than 2 years
       when tenure >24 then 'More than 2 years'
281
282
                                                    4
                                                            1 Year
283 end
287 select tenure_range,
288 count(*) as Total_customers,
289 sum(churn) as Churned_customers,
290 cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as Churn_rate
291 from customer_churn
      group by 1
292
293 order by 4 desc
```

	tenure_range character varying (20)	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	6 Months	2150	697	32.42
2	1 Year	1584	156	9.85
3	2 Years	1467	95	6.48
4	More than 2 years	429	0	0.00

Findings:

- (1) Most of the customers had churned whose tenure _range is 6 Months.
- (2) The customers who had tenure range is more than 2 years, there were not interested to churn. i.e., the churn rate Is "0".

(7) Churn Rate based on Gender.

```
295 -- (7) Churn_rate by Gender
296 select gender,
297 count(*) as Total_customers,
298 sum(churn) as Churned_customers,
299 cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as Churn_rate
300 from customer_churn
301 group by 1
302 order by 4 desc
```

	gender text	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	Male	3384	600	17.73
2	Female	2246	348	15.49

Findings:

- (1) Churn rate of Male customers has higher when compare to Female customer.
- (2) 600 Male customers churned.

(8) <u>Customer_status based on hourspentonapp.</u>

	(8) Churn_status by average hourspendonapp select customer_status,		customer_status character varying (50)	averagehourspentonapp numeric
306	<pre>avg(hourspendonapp) as averagehourspentonApp</pre>	1	stayed	2.9286629645450662
307	<pre>from customer_churn</pre>	_		0.04405004007044
308	<pre>group by customer_status</pre>	2	churned	2.9641350210970464

- (1) Both Stayed and Churned customers had the same average hours spent on app.
- (2) Hourspentonapp might not be influenced churned customers.

(9) Churn_Rate based on Numberofregistered devices.

```
310 -- (9) Churn_rate by NumberOfRegisteredDevices
311 select numberofdeviceregistered,
312 count(*) as Total_customers,
313 sum(churn) as Churned_customers,
314 cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2))as Churn_rate
315 from customer_churn
316 group by 1
317 order by 4
```

	numberofdeviceregistered integer	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	1	235	22	9.36
2	2	276	26	9.42
3	3	1699	254	14.95
4	4	2377	392	16.49
5	5	881	198	22.47
6	6	162	56	34.57

Findings:

- (1) We can observe there is positive correlation between "Numberofdeviceregistered" and "Churn rate"
- (2) Customers with less registered devices more likely to low churn rate.
- (3) Numberofdeviceregistered is influence the churn rate.

(10) Churn Rate based on preferordercat.

```
319 -- (10) Churn_rate by PreferorderCat
320 select preferedordercat,
321 count(*) as total_customers,
322 sum(churn) as Churned_customers,
323 cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as churn_rate
324 from customer_churn
325 group by 1
326 order by 4 desc
```

	preferedordercat text	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	Mobile Phone	2080	570	27.40
2	Fashion	826	128	15.50
3	Laptop & Accessory	2050	210	10.24
4	Others	264	20	7.58
5	Grocery	410	20	4.88

- (1) Mobile Phone category had Highest Churn_rate compare to all other categories.
- (2) Grocery category had lowest churn_rate.

(11) Churn_Rate based on SatisfactionScores.

```
328 -- (11) Churn_rate by SatisfactionScores.
329 select satisfactionscore,
330 count(*) as total_customers,
331 sum(churn) as Churned_customers,
332 cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as Churn_rate
333 from customer_churn
334 group by 1
335 order by 4 asc
```

	satisfactionscore text	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	1	1164	134	11.51
2	2	586	74	12.63
3	4	1074	184	17.13
4	3	1698	292	17.20
5	5	1108	264	23.83

Findings:

- (1) The satisfaction score ranges from 1 to 5
- (2) Most of customers had satisfaction score as 1 and 5
- (3) Those who had Higher satisfaction score as 5, had higher churn_rate. And satisfaction score positive correlated with churn_rate.

(12) Churn Rate based on Marital Status.

```
-- (12) Churn_rate by Marrital_status

select maritalstatus,

count(*) as total_customers,

sum(churn) as Churned_Customers,

cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as Churn_rate

from customer_churn

group by 1
```

	maritalstatus text	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	Married	2986	344	11.52
2	Divorced	848	124	14.62
3	Single	1796	480	26.73

- (1) Single Customers have the highest churn_rate compare to customers with other marital_status
- (2) Married customers have the lowest churn_rate followed by Divorced.

(13) Churn_Rate based on Number of Address.



Findings:

(1) Customers churned who were 4 addresses changed on an Average.

(14) Churn Rate based on Complaints received.

```
350 -- (14) Churn_rate by Complaint_received
351
    select complaint_received,
352 count(*) as Total_customers,
353 sum(churn) as Churned_customers,
    cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as Churn_rate
355
    from customer_churn
356
     group by 1
357 order by 4
      complaint_received
                            total_customers
                                              churned_customers
                                                                  churn_rate
       character varying (10)
                                                                  numeric (10,2)
                            bigint
                                              bigint
1
                                        4026
                                                                            10.93
       no
                                                             440
2
                                        1604
                                                             508
                                                                            31.67
       yes
```

Findings:

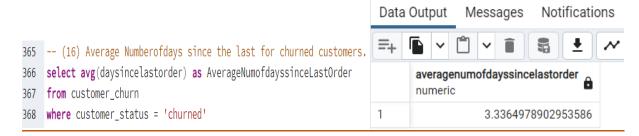
- (1) The customers who had raised the complaint last month they were more likely to churn.
- (2) Churn_rate of the complaint raised is more than complaints not raised customers.

(15) Customer Status by by Couponsused.

	(15) customer_status by Couponused select customer_status,		customer_status character varying (50)	sumofcouponsused bigint
361	<pre>sum(couponused) as SumofCouponsUsed</pre>	1	stayed	8292
362	<pre>from customer_churn</pre>		•	
363	<pre>group by customer_status</pre>	2	churned	1630

- (1) Most of the Stayed customers used coupons.
- (2) churned customers used 1630 coupons.

(16) Average Number of days since last order for churned customer.



Findings:

(1) the churned customers have, on average, a short period of time since their last order indicates that they recently stopped engaging with the company. By focusing on the overall customer experience, implementing targeted retention initiatives, and maintaining continuous engagement.

(17) Churn rate by CashbackAmount.

First of all I have created new column called "CashbackAmountRange" that groups the cashbackamount into low (<100), Moderate (between 100 and 200), High(between 200 and 300), and very High (more than 300) using the CASE Statement

```
370 -- (17) Churn_rate by CashbackAmount
371 -- Create new column called cashbackamountRange
372 alter table customer_churn
373 add cashbackamountRange varchar(50)
374
375 update customer_churn
376 set cashbackamountrange =
377 case
378
       when cashbackamount <= 100 then 'Low Cashback Amount'</pre>
379
       when cashbackamount > 100 and cashbackamount <= 200 then 'Moderate Cashback Amount'</pre>
380
       when cashbackamount > 200 and cashbackamount <= 300 then 'High Cashback Amount'
381
       when cashbackamount >300 then 'Very High Cashback Amount'
382 end
384 select cashbackamountRange,
385
      count(*) as total_customers,
     sum(churn) as Churned_Customers,
386
387
     cast((sum(churn)*1.0/count(*)*1.0)*100 as decimal(10,2)) as churn_rate
     from customer_churn
388
389
      group by 1
      order by 4 asc
390
```

	cashbackamountrange character varying (50)	total_customers bigint	churned_customers bigint	churn_rate numeric (10,2)
1	Low Cashback Amount	12	0	0.00
2	Very High Cashback Amount	156	10	6.41
3	High Cashback Amount	1157	124	10.72
4	Moderate Cashback Amount	4305	814	18.91

Findings:

- (1) Customers who had moderate cashback amount had a relatively higher churn_rate.
- (2) Those who received Higher and very High cashback amount having lower churn_rate.

Key Insights of Analysis

- The dataset has 5630 customers.
- The overall churn rate is 16.84%.
- Customers who prefer logging in with a computer have slightly higher churn rate compare to phone users.
- Tier 1 cities have lower churn rates than tier 2 and tier 3 cities.
- Customers who residing very close to warehouse tend to have lower churn rate and Customers who residing very far to warehouse tend to have higher churn rate.
- "Cash on Delivery" and "E wallet" payment modes have higher churn rates, while "Credit Card" and "Debit Card" have lower churn rates.
- Longer tenure is associated with lower churn rates.
- Male customers have slightly higher churn rates than Female Customers.
- Both Stayed and Churned customers had the same average hours spent on app.
- "Mobile Phone" order category has the highest churn rate, while "Grocery" has the lowest churn rate.
- More registered devices correlated with higher churn rates.
- Highly satisfied customers (rating 5) have a relatively higher churn rate.
- Single customers have the highest churn rate, while married customers have lower churn rate.
- Churned customers have an average of 4 associated addresses.
- Customer complaints are prevalent among churned customers, emphasizing the importance of addressing concerns to minimize churn.
- Coupon usage is higher among non-churned customers.
- Moderate cashback amounts correspond to higher churn rates, while higher amounts leads to lower churn rates.