

Questions and guidelines for the task :-

Responded_Pro_Booking - is an ID and is also referred to as Pro/ Professional who are delivering services. Customer_Request_ID - is an ID and is also referred to as one Job done by the professional/Pro.

```
In [1]: ▶ import pandas as pd
import numpy as np
import seaborn as sns
```

```
In [2]: ▶ # reading the dataset
df = pd.read_excel(r'C:\Users\HP\Downloads\dert.xlsx').dropna()
df.reset_index(drop=True, inplace=True)
```

```
In [3]: ▶ df
```

Out[3]:

	CUSTOMER_REQUEST_ID		USER_ID	RESPONDED_PRO_BOOKING	E
0	5f13286a7e307b220006dc4f	58f319a5d8bcf00c5597fc54	5e22a250ec1f9d2400e3924d		
1	5f134d78f39b082600461bf5	5ccd8c9f2f3964b16a4bb2e8	5e54bba97e76f42500a467f0		
2	5f13dbe5c91943280003b8c1	5f12f2a4b7183f38dfe2a352	5ef5ce6752f5e7260061c287		
3	5e5a4badbaa6c72d00937170	5e48fbd8b7183f38dfd9ffa3	5d8dc921ad34f72600fa6a9d		
4	5e5a5e37eeddf2a00123453	5df912a97c707c34b0684720	5d2db75cb7d2082300659f2a		
...		
668120	5f9b942727791c290006078d	5c96511d3658cc250017f488	5ed9dd5c0798bd25000757f8		
668121	5f9b94279fc49324003d843c	59441c13d8bcf00c55ac0d6f	5f81b4ec0018a52600ac9fe5		
668122	5f9b9430948ce325009f14d4	5ec7915fb7183f38dff9829	5beebdd42baa35220094924b		
668123	5f9b944d6393fc230060188b	5f9b927110170503776b2263	5f469123fb27b72400f8a996		
668124	5f9b9450cdb6222500c4558d	5f81f26a10170503776aefd1	5df0c371224f98240089c789		

668125 rows × 4 columns

Question 1 :

Count month wise deliveries

```
In [5]: # based on the CUSTOMER_REQUEST
df.drop(columns=['RESPONDED_PRO_BOOKING', 'USER_ID']).groupby('BDATE_FINAL').
```

Out[5]:

CUSTOMER_REQUEST_ID	
BDATE_FINAL	
2020-01-01	546
2020-01-02	535
2020-01-03	581
2020-01-04	897
2020-01-05	1397
...	...
2020-10-27	2713
2020-10-28	5718
2020-10-29	3559
2020-10-30	5303
2020-10-31	6081

267 rows × 1 columns

```
In [7]: suyash = df.drop(columns=['RESPONDED_PRO_BOOKING', 'USER_ID']).dropna().groupby('BDATE_FINAL')
suyash
```

Out[7]:

BDATE_FINAL	
2020-01-01	546
2020-01-02	535
2020-01-03	581
2020-01-04	897
2020-01-05	1397
...	...
2020-10-27	2713
2020-10-28	5718
2020-10-29	3559
2020-10-30	5303
2020-10-31	6081

Length: 267, dtype: int64

Question 2 :

Count month wise active pros (Providers with atleast one delivery)

```
In [8]: # based on the provider ID with duplicate
df.drop(columns=['CUSTOMER_REQUEST_ID', 'USER_ID']).dropna().groupby('BDATE_F
```

Out[8]:

RESPONDED_PRO_BOOKING	
BDATE_FINAL	
2020-01-01	546
2020-01-02	535
2020-01-03	581
2020-01-04	897
2020-01-05	1397
...	...
2020-10-27	2713
2020-10-28	5718
2020-10-29	3559
2020-10-30	5303

```
In [9]: kp = list(df.drop(columns=['CUSTOMER_REQUEST_ID', 'USER_ID']).dropna().groupb
```

```
In [10]: # based on the provider ID without duplicate PROVIDER_ID
df.drop(columns=['CUSTOMER_REQUEST_ID', 'USER_ID']).dropna().drop_duplicates(
```

Out[10]:

RESPONDED_PRO_BOOKING	
BDATE_FINAL	
2020-01-01	220
2020-01-02	233
2020-01-03	265
2020-01-04	319
2020-01-05	351
...	...
2020-10-27	1293
2020-10-28	1984
2020-10-29	1568
2020-10-30	1942

```
In [11]: tejas = list(df.drop(columns=['CUSTOMER_REQUEST_ID', 'USER_ID']).dropna().dro
```

Question 3 :

Get Month wise New & Old Pro(Responded_Pro_Booking) count

New Pro = First Job(Customer_Request_Id) in the month (only consider 2020)

Old Pro = First Job(Customer_Request_Id) in past months (only 2020)

```
In [15]: vineet = df[['BDATE_FINAL', 'RESPONDED_PRO_BOOKING']].sort_values(['BDATE_FINAL'])
```

```
In [17]: siddarth = [list(vineet[vineet['BDATE_FINAL'] == i]['RESPONDED_PRO_BOOKING'])
```

```
In [19]: te, kp1, kp2 = [], [], []
for i in siddarth:
    suy_addr = []
    suy1_addr = []
    for j in i:
        if j not in te:
            suy_addr.append(j)
        else:
            suy1_addr.append(j)
    kp1.append(suy_addr)
    kp2.append(suy1_addr)
    te.extend(kp1[0])
```

```
In [22]: newdata = pd.DataFrame()
newdata['MTH'] = suyash.index
newdata_PRO_OLD = pd.DataFrame()
newdata_PRO_OLD['MTH'] = newdata['MTH']
newdata_PRO_OLD['kp1'] = [len(i) for i in kp1]
newdata_PRO_OLD['kp2'] = [len(i) for i in kp2]
```

```
In [23]: newdata_PRO_OLD
```

Out[23]:

	MTH	kp1	kp2
0	2020-01-01	220	0
1	2020-01-02	70	163
2	2020-01-03	92	173
3	2020-01-04	127	192
4	2020-01-05	143	208
...
262	2020-10-27	1201	92
263	2020-10-28	1868	116
264	2020-10-29	1474	94
265	2020-10-30	1820	122
266	2020-10-31	1844	140

267 rows × 3 columns

Question 4 :

Get month wise new. repeat pro & reactive pro count

Old pro count = repeat pro count + reactive pro count

Repeat pro has deliveries in previous month & current month

Reactive pro has deliveries some time in past & current month

In [25]: `newdata_PRO_OLD`

Out[25]:

	MTH	kp1	kp2
0	2020-01-01	220	0
1	2020-01-02	70	163
2	2020-01-03	92	173
3	2020-01-04	127	192
4	2020-01-05	143	208
...
262	2020-10-27	1201	92
263	2020-10-28	1868	116
264	2020-10-29	1474	94
265	2020-10-30	1820	122
266	2020-10-31	1844	140

267 rows × 3 columns

```
In [26]: NEW_PRO_ = newdata_PRO_OLD[['MTH','kp1']]
REPEAT_PRO_ = newdata_PRO_OLD[['MTH','kp2']]
NEW_PRO_['STATUS'] = ['NEW' for i in range(len(kp1))]
REPEAT_PRO_['STATUS'] = ['REPEAT' for i in range(len(kp2))]
NEW_PRO_.columns = ['MTH', 'PRO_COUNT', 'PRO_STATE']
REPEAT_PRO_.columns = ['MTH', 'PRO_COUNT', 'PRO_STATE']
```

```
In [28]: Question_4 = pd.concat([REPEAT_PRO_,NEW_PRO_]).sort_values('MTH')
Question_4.columns = ['MTH', 'PRO_COUNT', 'PRO_STATE']
Question_4 = Question_4[['MTH','PRO_STATE','PRO_COUNT']].sort_values(['MTH',
```

In [29]: `Question_4[Question_4['PRO_COUNT'] > 0]`

Out[29]:

	MTH	PRO_STATE	PRO_COUNT
0	2020-01-01	NEW	220
1	2020-01-02	NEW	70
1	2020-01-02	REPEAT	163
2	2020-01-03	NEW	92
2	2020-01-03	REPEAT	173
...
264	2020-10-29	REPEAT	94
265	2020-10-30	NEW	1820
265	2020-10-30	REPEAT	122
266	2020-10-31	NEW	1844
266	2020-10-31	REPEAT	140

530 rows × 3 columns

Question 5 :

Get month wise churn %

Churn % = Pro not delivered in next month out of the pros active in current month/pros active in current month

```
In [37]: Question_4[Question_4['PRO_STATE'] == 'NEW'].drop(columns='PRO_STATE')
```

Out[37]:

	MTH	PRO_COUNT
0	2020-01-01	220
1	2020-01-02	70
2	2020-01-03	92
3	2020-01-04	127
4	2020-01-05	143
...
262	2020-10-27	1201
263	2020-10-28	1868
264	2020-10-29	1474
265	2020-10-30	1820
266	2020-10-31	1844

267 rows × 2 columns

SQL QUERY:-

Question 1 : Count month wise deliveries

Answer :- Select count(deliveries), month from table group by month;

Question 2 : Count month wise active pros (Providers with at least one delivery)

Answer :- select count (activepros), month from the table groupby month;

Question 3 : Get Month wise New & Old Pro(Responded_Pro_Booking) count

New Pro = First Job(Customer_Request_Id) in the month (only consider 2020) Old Pro = First Job(Customer_Request_Id) in past months (only 2020)

Answer :- select month, count(deliveries in present and previous month) as newpro, count(deliveries in past and present month) as oldpro from table group by month;

Question 4: Get month-wise churn %

Churn % = Pro not delivered in next month out of the pros active in the current month/pros active in the current month.

Answer :- Select month, prostate state, and old pro count from the table, where prostate is new if it's the first job of the month, prostate is repeat if it had deliveries in the previous month and the current month, and prostate is reactive pro if it had deliveries at some point in the past and the current month; The percentage churn is computed by the pro not delivered to the active pros
Computed by using $75000/1000\% = 75\%$

Question 5: Get month wise pro state wise churn %

Pro State - Old (Reactive / Repeat) or New

New Pro = First Job in the month (only consider 2020)

Old Pro = First Job in some month in past (only 2020)

Old pro count = repeat pro count + reactive pro count

Repeat pro has deliveries in previous month & current month.

Reactive pro has delivered some time in the past

& current month.

Answer :- select month, pro_state, oldprocount from table where prostate is new if firstjob in the month, prostate is repeat if has deliveries in previous moth and current month, prostate is reactive_pro if has deliveries some time in past and current month
 $\text{old_pro_count} = \text{repeatprocount} + \text{reactiveprocount};$

THANK YOU