

# Call Center Data Analysis : Using Postgres SQL

- **Data Set:** Call center data set of Telecom domain
- **Tool Used :** Postgres SQL (Pgadmin)
- **Objective:** to analyse the call center dataset and provide the insights on the different KPI parameters to the Telecom manager to explain the whole story.

**To achieve the required result we need to focus on important KPI Parameters as mentioned below,**

1. Total calls received ( abandoned and answered)
2. Average customer satisfaction rating
3. Total unresolved calls and by Topic.
4. Average speed of answer in seconds
5. Category wise complaints received.
6. Calls by time – Hourly wise count and cumulative count.
7. Agent's Performance Quadrant – Average handle time (talk duration) VS calls answered.

## **Steps:**

### **1. Connecting the Python to PostgreSQL.**

To connect PostgreSQL with python – we need to use the psycopg2 package, the most popular PostgreSQL database adapter for Python.

**Install psycopg2 package ---pip install psycopg2**

Before connecting to any database, we need to create a connection first and create an instance by using

**Psycopg2.connection**

The basic connection parameters required are as mentioned below:

- **Database:** The database name.
- **User:** User name required to authenticate. Default it will be "postgres"
- **Password:** Password used to authenticate. Starting it will be given
- **Host: Database** server address (in our case, the database is hosted locally, but it could be an IP address).
- **Port:** Connection port number (defaults to 5432 if not provided).

```

48 # -----
49 import psycopg2
50 try:
51     conn = psycopg2.connect(
52         database = "logisticsDB",
53         user = "postgres",
54         host = 'localhost',
55         password = "12345678",
56         port = 5432
57     )
58     # print the message if connection was successful
59     print("the connection was successful")
60
61 except(Exception,psycopg2.Error) as error:
62     print("error while connecting to PSQL",error)
63
64 finally:
65     if conn:
66         conn.close()
67         print("connection was closed successfully")
68

```

## 2. Reading the Excel File from Python using Pandas

```

72 # Now we will read the Excel File using python pandas
73 file_path = "D:\\PWC- Job Simulation\\Call Center Data analysis using PSQL\\telecom.xlsx"
74 data = pd.read_excel(file_path)
75 print(data.head(10))
76
77 print(data.describe()) # to check the basic statistics of our numerical columns
78
79 print(data.columns) # to see the total column names in the dataset
80

```

## 3. Inserting that data into PSQL Database.

```

138 import sqlalchemy
139 from sqlalchemy import create_engine
140 conn_string = 'postgresql://postgres:12345678@localhost/logisticsDB'
141 db = create_engine(conn_string)
142 conn1 = db.connect()
143
144 data.to_sql('telecom',con=conn1,if_exists='replace',index=False)
145

```

**Note:** Pandas only supports sqlalchemy, so again we have created the connection and then inserted that data into our database.

----Now we will start the data analysis with focus on the KPI parameters -----

### 1. Handling the null values:

- Before that as we have observed there are “null” values are present in 3 columns that need to be addressed i.e in “AvgTalkDuration”, “Speed of answer in seconds” and “Customer satisfaction rating”.
- As we can see that when the call is not answered then these should become zero.
- So we will replace all the “null” values present in 3 columns with “0” for "Speed of answer in seconds" and “Satisfaction rating” and with “00:00:00” for “AvgTalkDuration”.
- For this we will use the CASE statement.

## 2. Total calls answered / Abonded

```
27
28 --Total Calls Recived (answered / abonded)
29 ✓ SELECT "Answered (Y/N)" AS status,
30        COUNT("Call Id") AS total_calls
31        FROM telecom
32        GROUP BY "Answered (Y/N)"
33        ORDER BY total_calls
34
35
```

Data Output Messages Notifications



	status text	total_calls bigint
1	N	946
2	Y	4054

## 3. WAQ to calculate the Average Customer Satisfaction rating

```
36 --Average Customer Satisfaction rating:
37 select cast(avg("Satisfaction rating") as decimal(10,2)) as Avg_Satisfaction_Rating from telecom;
38
```

Data Output Messages Notifications



	avg_satisfaction_rating numeric (10,2)
1	2.76

## 4. WAQ to calculate the total no of unresolved calls and by category (Topic).

```
39 -- Total Unresolved calls
40
41 select count("Call Id") as total_unresolved_calls from telecom where "Resolved" = 'N';
42
43 -- Total Unresolved calls and also by category
44 ✓ select "Topic",count("Call Id") as Total_calls from telecom
45 where "Resolved" = 'N'
46     group by "Topic"
47     order by Total_calls;
48
```

Data Output Messages Notifications



	total_unresolved_calls bigint
1	1354

Data Output Messages Notifications



	Topic text	total_calls bigint
1	Admin Support	253
2	Contract related	267
3	Streaming	273
4	Payment related	278
5	Technical Support	283

## 5. WAQ to get the average speed of answer to a call in seconds.

```
49 -- Average speed of answer in seconds
50 v select "Agent",cast(avg("Speed of answer in seconds") as decimal(10,2)) as avg_speed_of_answer from telecom
51 group by "Agent";
52
```

Data Output Messages Notifications

	Agent text	avg_speed_of_answer numeric (10,2)
1	Becky	53.53
2	Martha	55.98
3	Greg	55.06
4	Dan	55.59
5	Jim	53.39
6	Stewart	54.24
7	Diane	52.45
8	Joe	57.94

## 6. WAQ to get the Topic (category) wise complaints received.

```
56 -- Category wise complaint recieved
57 v select "Topic" as "Category",count("Call Id") as "Total Calls" from telecom
58 group by "Topic"
59 order by "Total Calls";
60
```

Data Output Messages Notifications

	Category text	Total Calls bigint
1	Admin Support	976
2	Contract related	976
3	Payment related	1007
4	Technical Support	1019
5	Streaming	1022

## 7. WAQ to get the calls by hour and cumulative count also.

```
61 -- Total Calls by Hour
62 v select DATE_TRUNC('hour','Time') as "Hour",count("Call Id") as "Total Calls" from telecom
63 group by "Hour"
64 order by "Total Calls" desc;
65
66 -- Cumulative Calls by Hour
67 v select DATE_TRUNC('hour','Time') as "Hour",sum(count(*)) over (order by DATE_TRUNC('hour','Time')) as "Total Calls"
68 from telecom
69 group by "Hour";
70
```

Data Output Messages Notifications

	Hour interval	Total Calls bigint
1	13:00:00	594
2	11:00:00	590
3	17:00:00	583
4	16:00:00	566
5	09:00:00	547
6	12:00:00	547
7	15:00:00	531
8	10:00:00	529
9	14:00:00	499
10	18:00:00	14

Data Output Messages Notifications		
	Hour Interval	Total Calls numeric
1	09:00:00	547
2	10:00:00	1076
3	11:00:00	1666
4	12:00:00	2213
5	13:00:00	2807
6	14:00:00	3306
7	15:00:00	3837
8	16:00:00	4403
9	17:00:00	4986
10	18:00:00	5000

## 8. WAQ to get the Agent performance record – Avg talk duration and calls answered.

71	-- Agent Performance Quadrant - Avg talk duration and calls answered
72	Select "Agent",CAST(AVG(EXTRACT(EPOCH FROM "AvgTalkDuration"))) as decimal(10,2)) as "avg_duration" from telecom
73	group by "Agent";
74	
75	

Data Output Messages Notifications		
	Agent text	avg_duration numeric (10,2)
1	Becky	180.26
2	Martha	180.24
3	Greg	182.46
4	Dan	191.01
5	Jim	183.59
6	Stewart	185.40
7	Diane	173.30
8	Joe	182.91

75	
76	select "Agent",count("Call Id") as "calls_answered" from telecom
77	where "Answered (Y/N)" = 'Y'
78	group by "Agent"

Data Output Messages Notifications		
	Agent text	calls_answered bigint
1	Jim	536
2	Dan	523
3	Becky	517
4	Martha	514
5	Greg	502
6	Diane	501
7	Joe	484
8	Stewart	477