Call Center Data Analysis: Using Postgres SQL

- Data Set: Call center data set of Telecom domain
- Tool Used : Postures SQL (Pgadmin)
- **Objective:** to analyse the call center dataset and provide the insights on the different KPI parameters to the Telecom manger 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 (abonded 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 PSSQL.

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).

```
49
      import psycopa2
50
      try:
51
          conn = psycopg2.connect(
          database = "logisticksDB",
53
          user = "postgres",
          host = 'localhost'
          password = "12345678",
          port = 5432
57
58
          # print the message if connection was successfull
          print("the connection was successful")
59
60
      except(Exception_psycopg2.Error) as error:
61
          print("error while connecting to PSSQL",error)
62
63
      finally:
64
          if conn:
65
              conn.close()
              print("connection was closed successfully")
68
```

2. Reading the Excel File from Python using Pandas

```
# Now we will read the Excel File using python pandas
file_path = "D:\\PWC- Job Simulation\\Call Center Data analysisi using PSSQL\\telecom.xlsx"
data = pd.read_excel(file_path)
print(data.head(10))

print(data.describe()) # to check the basic statasticks of our numerical columns

print(data.columns) # to see the total column names in the dataset
```

3. Inserting that data into PSSQL Database.

```
import sqlalchemy
from sqlalchemy import create_engine
conn_string = 'postgresql://postgres:12345678@localhost/logisticksDB'
db = create_engine(conn_string)
conn1 = db.connect()

data.to_sql('telecom',con=conn1,if_exists='replace',index=False)

data.to_sql('telecom',con=conn1,if_exists='replace',index=False)
```

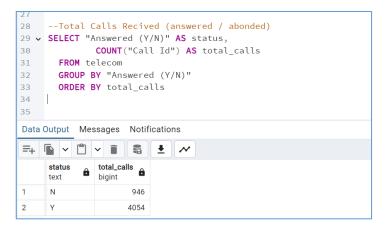
Note: Pandas only supports sqlalchemy, so again we have created the connection and then inserted that data into our database.

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

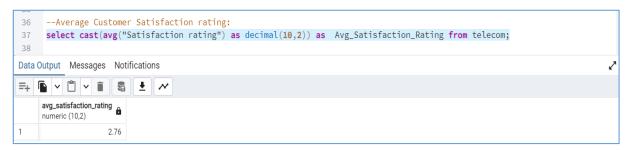
1. Handling the null values:

- 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.

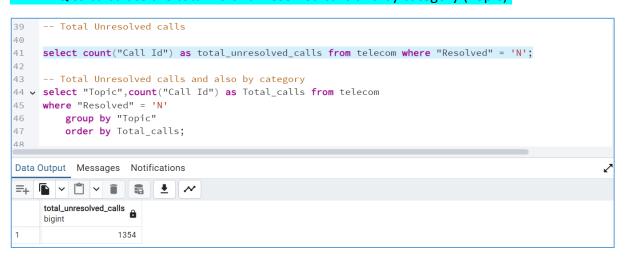
2. Total calls answered / Abonded

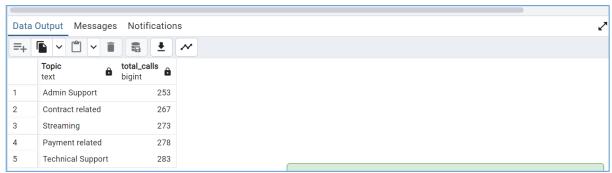


3. WAQ to calculate the Average Customer Satisfaction rating

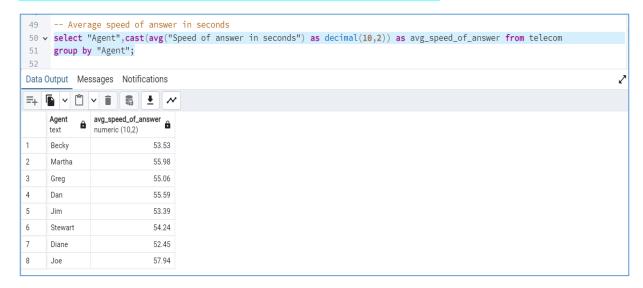


4. WAQ to calculate the total no of unresolved calls and by category (Topic).





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

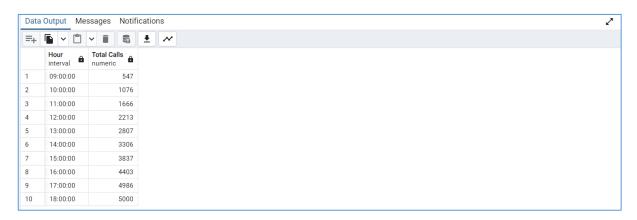


6. WAQ to get the Topic (category) wise complaints received.



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





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

