In [7]:

```python
import pandas as pd
import sqlite3
df=pd.read_csv("Customer_support_data.csv")
conn=sqlite3.connect("Customer_support_data.db")
df.to_sql("Customer_support_table",conn,if_exists="replace",index=False)
```

Out[7]:

```
85907
```

In [33]:

```python
#displaying all the records
pd.read_sql_query("SELECT * FROM Customer_support_table",conn)
```

Out[33]:

	Unique id	channel_name	category	Sub-category	Customer Remarks	Order_id	order_date_time	Issue_reported at	i
0	7e9ae164-6a8b-4521-a2d4-58f7c9fff13f	Outcall	Product Queries	Life Insurance	None	c27c9bb4-fa36-4140-9f1f-21009254ffdb	None	01/08/2023 11:13	
1	b07ec1b0-f376-43b6-86df-ec03da3b2e16	Outcall	Product Queries	Product Specific Information	None	d406b0c7-ce17-4654-b9de-f08d421254bd	None	01/08/2023 12:52	
2	200814dd-27c7-4149-ba2b-bd3af3092880	Inbound	Order Related	Installation/demo	None	c273368d-b961-44cb-beaf-62d6fd6c00d5	None	01/08/2023 20:16	
3	eb0d3e53-c1ca-42d3-8486-e42c8d622135	Inbound	Returns	Reverse Pickup Enquiry	None	5aed0059-55a4-4ec6-bb54-97942092020a	None	01/08/2023 20:56	
4	ba903143-1e54-406c-b969-46c52f92e5df	Inbound	Cancellation	Not Needed	None	e8bed5a9-6933-4aff-9dc6-ccefd7dcde59	None	01/08/2023 10:30	
...	...	...	...	...	...	...	...	...	
85902	505ea5e7-c475-4fac-ac36-1d19a4cb610f	Inbound	Refund Related	Refund Enquiry	None	1b5a2b9c-a95f-405f-a42e-5b1b693f3dc9	None	30/08/2023 23:20	
85903	44b38d3f-1523-4182-aba2-72917586647c	Inbound	Order Related	Seller Cancelled Order	Supported team customer executive good	d0e8a817-96d5-4ace-bb82-adec50398e22	None	31/08/2023 08:15	
85904	723bce2c-496c-4aa8-	Inbound	Order Related	Order status enquiry	need to improve with	bdefe788-ccec-4eda-	None	31/08/2023 18:57	

	Unique id	channel_name	category	Sub-category	Customer Remarks	Order_id	order_date_time	Issue_reported at	i
	a64b-ca17004528f0				proper details.	8ca4-51045e68db8a			
85905	707528ee-6873-4192-bfa9-a491f1c08ab5	Inbound	Feedback	UnProfessional Behaviour	None	a031ec28-0c5e-450e-95b2-592342c40bc4	None	31/08/2023 19:59	
85906	07c7a878-0d5a-42e0-97ef-de59abec0238	Inbound	Returns	Reverse Pickup Enquiry	None	3230db30-f8da-4c44-8636-ec76d1d3d4f3	None	31/08/2023 23:36	

85907 rows × 20 columns

In [16]:

```
#Total number of support tickets
pd.read_sql_query("SELECT COUNT(*) AS total_tickets FROM Customer_support_table",conn)
```

Out[16]:

	total_tickets
0	85907

In [18]:

```
#Tickets by Channel
pd.read_sql_query(
    """SELECT channel_name, COUNT(*) AS ticket_count
    FROM Customer_support_table GROUP BY channel_name
    ORDER BY ticket_count DESC""",
    conn)
```

Out[18]:

	channel_name	ticket_count
0	Inbound	68142
1	Outcall	14742
2	Email	3023

In [19]:

```
#Averae CSAT Score by Agent
pd.read_sql_query(
    """SELECT Agent_name, AVG("CSAT Score") AS avg_csat
    FROM Customer_support_table
    GROUP BY Agent_name
    ORDER BY avg_csat DESC
    LIMIT 10""",
    conn)
```

Out[19]:

	Agent_name	avg_csat
0	Pamela Robinson	4.956522
1	Virginia Lane	4.909910
2	Sean Gay	4.909091
3	Taylor Nelson	4.888889
4	Nancy Singh	4.868421
5	Anthony Sims	4.868421
6	Morgan Smith	4.866667
7	John Hoffman	4.861111
8	Kelly Thomas	4.857143
9	Nicole Simpson DVM	4.850000

In [20]:

```
# Top 5 Cities with Most Issues Reported
pd.read_sql_query(
    """SELECT Customer_City, COUNT(*) AS issue_count
    FROM Customer_support_table
    WHERE Customer_City IS NOT NULL
    GROUP BY Customer_City
    ORDER BY issue_count DESC
    LIMIT 5""",
    conn)
```

Out[20]:

	Customer_City	issue_count
0	HYDERABAD	722
1	NEW DELHI	688
2	PUNE	435
3	MUMBAI	406
4	BANGALORE	352

In [21]:

```
#Most Common Product Issues
pd.read_sql_query(
    """SELECT Product_category, COUNT(*) AS issue_count
    FROM Customer_support_table
    WHERE Product_category IS NOT NULL
    GROUP BY Product_category
    ORDER BY issue_count DESC;""",
    conn)
```

Out[21]:

	Product_category	issue_count
0	Electronics	4706
1	LifeStyle	4118
2	Books & General merchandise	3323
3	Mobile	1758
4	Home	1328
5	Home Appliences	1300
6	Furniture	471
7	Affiliates	166
8	GiftCard	26

In [23]:

```
#Average Handling Time by Agent
pd.read_sql_query(
    """SELECT Agent_name, AVG(connected_handling_time) AS avg_handling_time
    FROM Customer_support_table
    WHERE connected_handling_time IS NOT NULL
    GROUP BY Agent_name
    ORDER BY avg_handling_time ASC
    LIMIT 10""",
    conn)
```

Out[23]:

	Agent_name	avg_handling_time
0	Robert Lara	0.0
1	Morgan Floyd	63.0
2	Adam Hammond	72.0
3	Eric Miller MD	125.0
4	Daryl Martin	142.0
5	Cole Moore	156.0
6	Bruce Smith	171.0
7	Omar Odonnell	176.0
8	John Barrett	196.0
9	Corey Gibson	206.0

In [27]:

```
#CSAT Score Distribution
pd.read_sql_query(
    """SELECT "CSAT Score", COUNT(*) AS count
    FROM Customer_support_table
    GROUP BY 'CSAT Score'
    ORDER BY 'CSAT Score'""",
    conn)
```

Out[27]:

	CSAT Score	count
0	5	85907

In [28]:

```
# Issues by Category and Sub-category
pd.read_sql_query(
    """SELECT category, "Sub-category", COUNT(*) AS count
    FROM Customer_support_table
    GROUP BY category, "Sub-category"
    ORDER BY count DESC
    LIMIT 10;""",
    conn)
```

Out[28]:

	category	Sub-category	count
0	Returns	Reverse Pickup Enquiry	22389
1	Returns	Return request	8523
2	Order Related	Delayed	7388
3	Order Related	Order status enquiry	6922
4	Order Related	Installation/demo	4116
5	Returns	Fraudulent User	4108
6	Product Queries	Product Specific Information	3589
7	Refund Related	Refund Enquiry	2665
8	Returns	Wrong	2597
9	Returns	Missing	2556

In [30]:

```
# Tickets Handled by Agent Shift
pd.read_sql_query(
    """SELECT "Agent Shift", COUNT(*) AS tickets_handled
    FROM Customer_support_table
    GROUP BY "Agent Shift"
    ORDER BY tickets_handled DESC""",
    conn)
```

Out[30]:

	Agent Shift	tickets_handled
0	Morning	41426
1	Evening	33677
2	Afternoon	5840
3	Split	3648
4	Night	1316

In [ ]: