

```
In [21]: import ollama
import mysql.connector
import pandas as pd
import json
```

```
In [22]: response = ollama.generate(
    model='llama3',
    prompt='Write a haiku about the highschool.'
)

print(response['response'])
```

```
Halls of learning stress  
Lockers creak, anxiety  
Growing up begins
```

```
In [23]: import mysql.connector
import pandas as pd

def run_query_df(sql):
    conn = mysql.connector.connect(
        host="localhost",
        user="root",
        password="root",
        database="call_transcripts",
        port=3306
    )

    cursor = conn.cursor()
    cursor.execute(sql)
    rows = cursor.fetchall()
    col_names = [desc[0] for desc in cursor.description]

    conn.close()

    return rows, col_names
```

```
In [24]: rows, col_names = run_query_df("SELECT * FROM calls;")
df = pd.DataFrame(rows, columns=col_names)
df.info()
df.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
 ---  --          -----          Dtype  
 0   CallID      100 non-null    int64  
 1   Date        100 non-null    object  
 2   Duration    100 non-null    timedelta64[ns]
 3   Transcript  100 non-null    object  
 4   summary     0 non-null     object  
 5   category    0 non-null     object  
dtypes: int64(1), object(4), timedelta64[ns](1)
memory usage: 4.8+ KB
```

Out[24]:

	CallID	Date	Duration	Transcript	summary	category
0	1	2025-01-15	0 days 06:32:00	Caller: Hi there, I recently purchased a blend...	None	None
1	2	2025-02-20	0 days 07:45:00	Caller: Hello, I was wondering if your webshop...	None	None
2	3	2025-03-10	0 days 05:18:00	Caller: The pair of shoes I ordered arrived, b...	None	None
3	4	2025-04-05	0 days 08:10:00	Caller: Hi, I'm shopping for wireless earbuds ...	None	None
4	5	2025-05-22	0 days 04:25:00	Caller: I see a pending charge on my credit ca...	None	None

In [ ]:

```
GOLD_CATEGORIES = {
    "Shipping Issue": [
        "wrong address", "lost package", "delivery problem", "shipping refund"
    ],
    "Product Inquiry": [
        "eco friendly", "materials", "certifications", "sustainability"
    ],
    "Account / Data Issue": [
        "wish list", "account access", "data missing", "restore"
    ],
    "Return / Refund": [
        "wrong color", "defective", "return label", "replacement"
    ],
    "Service Request": [
        "installation", "schedule service", "haul away"
    ]
}
gold_string = json.dumps(GOLD_CATEGORIES, indent=2)
```

In [ ]:

```
def analyze_call(transcript):
    prompt = f"""
You are a call summarization and classification assistant.

Use ONLY the categories from this taxonomy:
{gold_string}

TASK:
1. Provide a 1-2 sentence summary of the call.
2. Choose EXACTLY ONE category from the taxonomy.

Return ONLY valid JSON in this structure:
{{{
    "summary": "...",
    "category": ...
}}}

Transcript:
\"\"\"{transcript}\"\"\"
"""

response = ollama.generate(
    model="llama3",
```

```
        prompt=prompt
    )

raw_output = response["response"].strip()

try:
    data = json.loads(raw_output)
except:
    import re
    match = re.search(r"\{.*\}", raw_output, re.DOTALL)
    if not match:
        raise ValueError("Could not parse LLM output: " + raw_output)
    data = json.loads(match.group(0))

return data["summary"], data["category"]

# -----
# Loop over the dataframe
# -----
for idx, row in df.iterrows():
    transcript = row["Transcript"]

    summary, category = analyze_call(transcript)

    df.at[idx, "summary"] = summary
    df.at[idx, "category"] = category
```

```
In [ ]: df.head(100)
```

Out[ ]:

	CallID	Date	Duration	Transcript	summary	category
0	1	2025-01-15	0 days 06:32:00	Caller: Hi there, I recently purchased a blend...	Customer received a defective blender after on...	Return / Refund
1	2	2025-02-20	0 days 07:45:00	Caller: Hello, I was wondering if your webshop...	The caller inquired about international shipp...	Shipping Issue
2	3	2025-03-10	0 days 05:18:00	Caller: The pair of shoes I ordered arrived, b...	The caller received a pair of shoes that is a ...	Return / Refund
3	4	2025-04-05	0 days 08:10:00	Caller: Hi, I'm shopping for wireless earbuds ...	The caller is looking for wireless earbuds und...	Product Inquiry
4	5	2025-05-22	0 days 04:25:00	Caller: I see a pending charge on my credit ca...	Caller has a pending charge on their credit ca...	Account / Data Issue
...	...	...	...	...	...	...
95	96	2024-12-22	0 days 07:50:00	Caller: Site isn't optimized for mobile, hard ...	The caller had difficulty navigating the websi...	Product Inquiry
96	97	2023-01-04	0 days 05:20:00	Caller: Hi, policy on reselling your products?...	The caller asked about the policy on reselling...	Product Inquiry
97	98	2023-02-19	0 days 08:35:00	Caller: Printer ink is leaking everywhere. Age...	The caller reported a defective printer cartri...	Return / Refund
98	99	2023-03-03	0 days 06:10:00	Caller: Hello, what's included in VIP membersh...	The caller inquired about the benefits of a VI...	Product Inquiry
99	100	2023-04-18	0 days 07:25:00	Caller: Package was rerouted to the wrong city...	The caller reported a shipping issue with thei...	Shipping Issue

100 rows × 6 columns

In [32]:

```
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="root",
    database="call_transcripts",
    port=3306
)

cursor = conn.cursor()

for idx, row in df.iterrows():
    sql = """
        UPDATE calls
        SET summary = %s,
            category = %s
    """
    cursor.execute(sql, (row['summary'], row['category']))
```

```
        WHERE CallID = %s
"""
val = (row["summary"], row["category"], row["CallID"])
cursor.execute(sql, val)

conn.commit()
cursor.close()
conn.close()

print("SQL table updated successfully.")
```

SQL table updated successfully.