

## Lab10 Report-Arkadeep Ganguly

### Objective:

In this lab, I learned how to generate reports using Python and store them as CSV files.

### Report Generation using Python:

In this lab, I connected to a MySQL database, fetched data from the student table, and saved it as a CSV file using Python. I followed the steps of installing necessary libraries, connecting to the database, executing queries, and generating the report.

### Steps Followed in the Lab:

#### 1. Installed Required Libraries:

I installed the necessary libraries using the command:

```
C:\Users\Arko>python --version
Python 3.13.2

C:\Users\Arko>python -m pip install mysql-connector-python pandas
C:\Users\Arko\AppData\Local\Programs\Python\Python313\python.exe: No module named pip

C:\Users\Arko>cd downloads

C:\Users\Arko\Downloads>python get-pip.py
Collecting pip
  Downloading pip-25.0.1-py3-none-any.whl.metadata (3.7 kB)
  Downloading pip-25.0.1-py3-none-any.whl (1.8 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.8/1.8 MB 1.2 MB/s eta 0:00:00
Installing collected packages: pip
Successfully installed pip-25.0.1

C:\Users\Arko\Downloads>|
```

#### 2. Connected to MySQL Database:

I connected to my MySQL database using the following Python code:

```
import mysql.connector
import pandas as pd

# Step 1: Connect to MySQL Database
conn = mysql.connector.connect(
    host='localhost',
    user='root',
    password='root',
    database='arko5',
)
```

**Note:** I replaced `your_host_name`, `your_username`, `your_password`, and `your_database_name` with my actual credentials.

### 3. Fetched Data from Student Table:

I executed a SQL query to fetch data from the student table using the following Python code:

```
query = "SELECT * FROM cust_details"

# Use pandas to read the SQL query results into a DataFrame
df = pd.read_sql(query, conn)
```

### 4. Stored Data in CSV File:

I saved the fetched data into a CSV file using this code:

```
5.
6. csv_filename = "CustomerDetails.csv"
7. df.to_csv(csv_filename, index=False)
```

The Entire code:

```
import mysql.connector
import pandas as pd

# Step 1: Connect to MySQL Database
conn = mysql.connector.connect(
    host='localhost',
    user='root',
    password='root',
    database='arko5',
)

print("Connected to MySQL database successfully!")

# Write your SQL query to select all data from the student table
query = "SELECT * FROM cust_details"

# Use pandas to read the SQL query results into a DataFrame
df = pd.read_sql(query, conn)

# Let's verify we got some data
print("Successfully fetched data!")
print(f"Number of records: {len(df)}")
```

```

print("First few rows:")
print(df.head())

# (We'll close the connection in the next step)
# STEP 4: Save DataFrame to CSV
csv_filename = "CustomerDetails.csv"
df.to_csv(csv_filename, index=False) # index=False avoids extra column
numbers

print(f"\nSUCCESS: Saved {len(df)} records to '{csv_filename}'!")
# Close the connection (IMPORTANT to avoid resource leaks)
conn.close()
print("Database connection closed.")

```

Output:

```

SUCCESS: Saved 5 records to 'CustomerDetails.csv'!
PS C:\Users\Arko> & C:\Users\Arko\AppData\Local\Programs\Python\Python313\python.exe c:\Users\Arko\Desktop\Lab10.py
Connected to MySQL database successfully!
c:\Users\Arko\Desktop\Lab10.py:16: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested.
Please consider using SQLAlchemy.
  df = pd.read_sql(query, conn)
Successfully fetched data!
Number of records: 5
First few rows:
   CST_ID  cstdet_type  cstdet_dob  cstdet_status  cstdet_contact  ...  CSTDET_LOCAL_TS  CSTDET_ACPT_TS  CSTDET_ACPT_TS_UTC_0FST  cstdet_uuid  cstdet_acpt_ts_utc_of
0    1001  Individual  1980-05-15 00:00:00      ACTIVE      John Doe  ...  2025-03-23 16:02:06  2025-03-23 16:02:06      2025-03-23 16:02:06  uuid1001-DET      None
1    1002      None      NaT      None      None  ...      NaT      NaT      NaT      None      None      None
2    1003      None      NaT      None      None  ...      NaT      NaT      NaT      None      None      None
3    1004      None      NaT      None      None  ...      NaT      NaT      NaT      None      None      None
4    1005  INDIVIDUAL  1990-05-14 18:30:00      ACTIVE      John Doe  ...      NaT      NaT      NaT      None      None      None

[5 rows x 18 columns]

SUCCESS: Saved 5 records to 'CustomerDetails.csv'!
Database connection closed.
PS C:\Users\Arko> 

```

## Lab Exercises:

- **Exercise 1:**  
I fetched the customer details based on their language and age group.

Code:

```

import mysql.connector

import pandas as pd

# Connect to database

```

```
conn = mysql.connector.connect(  
  
    host='localhost',  
  
    user='root',  
  
    password='root',  
  
    database='arko5'  
  
)  
  
# Define parameters  
  
target_language = 'English'  
  
min_age = 20  
  
max_age = 30  
  
# SQL query with parameters  
  
query = f"""  
  
SELECT  
  
    CST_ID,  
  
    age,  
  
    language_spoken  
  
FROM  
  
    cust_details  
  
WHERE  
  
    language_spoken = '{target_language}'  
  
    AND age BETWEEN {min_age} AND {max_age}  
  
"""
```

```
# Execute query and load into DataFrame

df_customers = pd.read_sql(query, conn)

# Save to CSV

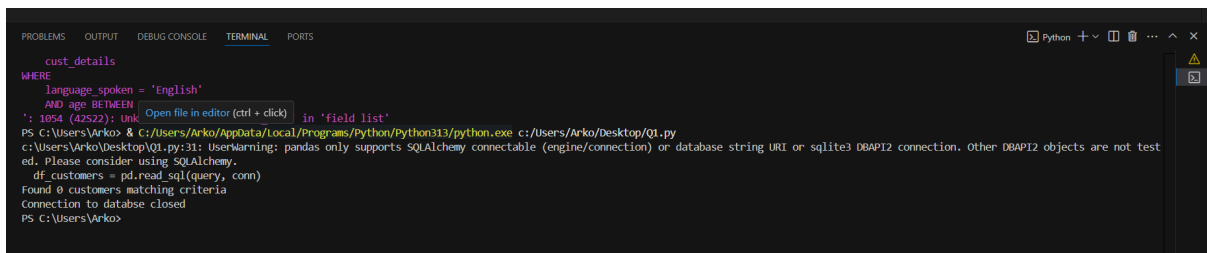
df_customers.to_csv('filtered_customers.csv', index=False)

print(f"Found {len(df_customers)} customers matching criteria")

conn.close()

print("Connection to databse closed")
```

Output:



The screenshot shows a Jupyter Notebook interface with a terminal window open. The terminal displays the output of a SQL query executed in a previous cell. The output includes a warning about pandas only supporting SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection, followed by the results of the query: 'Found 0 customers matching criteria' and 'Connection to databse closed'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - - - - - x

cust_details
WHERE
  language_spoken = 'English'
AND age BETWEEN
  1854 (42522) unk Open file in editor (ctrl + click) in 'field list'
PS C:\Users\Arko> & c:\Users\Arko\AppData\Local\Programs\Python\Python313\python.exe c:\Users\Arko\Desktop\Q1.py
c:\Users\Arko\Desktop\Q1.py:31: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not test
ed. Please consider using SQLAlchemy.
  df_customers = pd.read_sql(query, conn)
Found 0 customers matching criteria
Connection to databse closed
PS C:\Users\Arko>
```

- **Exercise 2:**

I calculated the ratio of male to female customers and saved the data in a CSV file.

Code:

```
import mysql.connector

import pandas as pd

# Establish connection

conn = mysql.connector.connect (

    host='localhost',
```

```

        user='root',

        password='root',

        database='arko5'

    )

gender_query = """

SELECT

    gender,

    COUNT(*) as count

FROM

    cust_details

WHERE

    gender IN ('M', 'F')  -- Only count M/F if other values exist

GROUP BY

    gender

"""

df_gender = pd.read_sql(gender_query, conn)

# Initialize counts

male_count = 0

female_count = 0

# Extract values from query results

for _, row in df_gender.iterrows():

    if row['gender'] == 'M':

        male_count = row['count']

```

```

        elif row['gender'] == 'F':

            female_count = row['count']

# Calculate ratio (handle division by zero)
try:

    ratio = male_count / female_count

except ZeroDivisionError:

    ratio = float('inf') # If no female customers

    report_data = {

        'Metric': ['Total Customers', 'Male Customers', 'Female
Customers', 'Male:Female Ratio'],

        'Value': [male_count + female_count, male_count,
female_count, f"{ratio:.2f}:1"]

    }

df_report = pd.DataFrame(report_data)

df_report.to_csv('gender_ratio_report.csv', index=False)

print("Gender ratio report generated successfully!")

conn.close()

```

Output:

```

PS C:\Users\Arko> & C:/Users/Arko/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arko/Desktop/Q2.py
c:/Users/Arko/Desktop/Q2.py:23: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not test
ed. Please consider using SQLAlchemy.
  df_gender = pd.read_sql(gender_query, conn)
Gender ratio report generated successfully!
PS C:\Users\Arko>

```

---

- **Exercise 3:**

I analyzed the distribution of languages spoken by customers in different countries and saved the result in a CSV file

Code:

```
import mysql.connector

import pandas as pd

# Establish connection

conn = mysql.connector.connect(

    host='localhost',

    user='root',

    password='root',

    database='arko5'

)

query = """

SELECT

    country,

    language_spoken,

    COUNT(*) AS speaker_count

FROM

    cust_details

GROUP BY

    country, language_spoken
```



```

ORDER BY

    country, speaker_count DESC

"""

df_lang_dist = pd.read_sql(query, conn)

pivot_table = df_lang_dist.pivot(

    index='country',

    columns='language_spoken',

    values='speaker_count'

).fillna(0)  # Replace NaN with 0

pivot_table['Total'] = pivot_table.sum(axis=1)

# Save raw distribution data

df_lang_dist.to_csv('language_distribution_raw.csv', index=False)

# Save pivot table (more readable)

pivot_table.to_csv('language_distribution_pivot.csv')

print("Language distribution reports saved successfully!")

conn.close()

```

Output:

```

PS C:\Users\Arko> & c:/Users/Arko/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Arko/Desktop/Q3.py
c:\Users\Arko\Desktop\Q3.py:23: UserWarning: pandas only supports SQLAlchemy connectable (engine/connection) or database string URI or sqlite3 DBAPI2 connection. Other DBAPI2 objects
ed. Please consider using SQLAlchemy.
  df_lang_dist = pd.read_sql(query, conn)
c:\Users\Arko\Desktop\Q3.py:24: FutureWarning: The behavior of Index.insert with object-dtype is deprecated, in a future version this will return an object-dtype Index instead of in
object dtype. To retain the old behavior, do `idx.insert(loc, item).infer_objects(copy=False)`
  pivot_table = df_lang_dist.pivot(
  pivot_table = df_lang_dist.pivot(
c:\Users\Arko\Desktop\Q3.py:24: FutureWarning: The behavior of Index.insert with object-dtype is deprecated, in a future version this will return an object-dtype Index instead of in
object dtype. To retain the old behavior, do `idx.insert(loc, item).infer_objects(copy=False)`
  pivot_table = df_lang_dist.pivot(
  pivot_table = df_lang_dist.pivot(
Language distribution reports saved successfully!

```

**Conclusion:**

In this lab, I have successfully connected to a MySQL database, fetched data, and saved it in CSV format. I have also completed the exercises related to customer data analysis and report generation using Python.