# MySQL + Python Data Analysis Complete Process

***MySQL + Python:*** Connector Guide

Here’s a concise guide on analyzing datasets in MySQL with Python, specifically designed for data analysts:

***Libraries & Tool Uses:***

Make sure to install these libraries for database management, visualization, and data manipulation:

* MySQL Python Connector
* matplotlib
* pandas
* seaborn
* Os
* Jupyter Notebook

***Key Libraries & Tool:***

* **MySQL Connector** for database interaction
* **Pandas** for data handling
* **Matplotlib & Seaborn** for visualization
* **Os** for handling file paths to locate CSV files in a specific folder
* **Jupyter Notebook** as the interactive environment

***Step 1: Set Up a Connection and Create a Database***

1. **Create Database in MySQL:**

First, open your MySQL management tool and connect to your MySQL server. Once connected, use the following command to create a new database:

CREATE DATABASE ecommerce;

1. **Import Libraries:**

import mysql.connector

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import os

1. **Define CSV Files and Table Names:**

csv\_files = [

('customers.csv', 'customers'),

('orders.csv', 'orders'),

('sellers.csv', 'sellers'),

('products.csv', 'products'),

('geolocation.csv', 'geolocation'),

('payments.csv', 'payments'),

('order\_items.csv', 'order\_items')

]

* This list of tuples specifies each CSV file and the corresponding table name in MySQL, allowing easy mapping between the file and its destination table.

1. **Connect to MySQL Server:**

Connect to the MySQL database.

db = mysql.connector.connect(

host="your\_host",

user="your\_user",

password="your\_password"

)

cur = db.cursor()

* Establishes a connection to the MySQL database and creates a cursor cur to execute SQL commands.

1. **Define Folder Path for CSV Files:**

folder\_path = 'add\_own\_folder\_path'

* Specifies the folder where the CSV files are stored.

1. **Define Helper Function for SQL Data Type Mapping:**

def get\_sql\_type(dtype):

if pd.api.types.is\_integer\_dtype(dtype):

return 'INT'

elif pd.api.types.is\_float\_dtype(dtype):

return 'FLOAT'

elif pd.api.types.is\_bool\_dtype(dtype):

return 'BOOLEAN'

elif pd.api.types.is\_datetime64\_any\_dtype(dtype):

return 'DATETIME'

else:

return 'TEXT'

* This function converts pandas data types to SQL data types, allowing the script to assign correct data types when creating MySQL tables.

1. **Iterate Through Each CSV File:**

for csv\_file, table\_name in csv\_files:

file\_path = os.path.join(folder\_path, csv\_file)

* The os.path.join function ensures that the correct file path is generated, regardless of the operating system.

***Step 2: Data Cleaning:***

1. **Load the CSV data using Pandas:**

df = pd.read\_csv("file\_path")

1. **Handle Missing Values:**

df = df.where(pd.notnull(df), None)

* Replace NaN values with `None` so MySQL can recognize them as `NULL`.

1. **Check NaN values:**

print(f"Processing {csv\_file}")

print(f"NaN values before replacement:\n{df.isnull().sum()}\n")

* Use `df.info()` and `df.describe()` to get a snapshot.

**4. Standardize Column Names:**

Ensure no spaces or special characters in column names for SQL compatibility.

df.columns = [col.replace(' ', '\_').replace('-', '\_').replace('.', '\_') for col in df.columns]

***Step 3: Creating Tables in MySQL :***

**1. Define Table Structure:**

Write a SQL `CREATE TABLE` statement based on your data columns and types.

columns = ', '.join([f'`{col}` {get\_sql\_type(df[col].dtype)}' for col in df.columns])

create\_table\_query = f'CREATE TABLE IF NOT EXISTS `{table\_name}` ({columns})'

cur.execute(create\_table\_query)

**2. Insert Data into MySQL:**

Insert cleaned data from the DataFrame into the MySQL table:

for i, row in df.iterrows():

Convert row to tuple and handle NaN/None explicitly

values = tuple(None if pd.isna(x) else x for x in row)

sql = f"INSERT INTO `{table\_name}` ({', '.join(['`' + col + '`' for col in df.columns])}) VALUES ({', '.join(['%s'] \* len(row))})"

cur.execute(sql, values)

**3. Commit Transaction:**

Always commit after batch inserts to save changes.

db.commit()

***Step 5: Clean Up:***

After finishing analysis, close the cursor and connection:

db.close()

With these steps, you'll have a fully complete process to clean, store, and analyze datasets using MySQL and Python!

# ***Summary of Questions:***

Here's a summary of the analysis of all Questions: -

**1. Unique Customer Cities:**

* Lists all unique cities where customers are located.
* The highest customer is located in France.
* The output shows a sample of customer locations like "Franca," "Sao Bernardo do Campo," and "Sao Paulo."

**2. Order Count for 2017:**

* This part of the analysis calculates the total number of orders placed in 2017.
* In 2017, a total of 45,101 orders were placed.

**3. Total Sales per Category:**

* The types of categories 73 were based on the total sales for each category.
* Sales in the Perfumery category are 506738.66, in the Furniture Decoration category are 1430176.39, Telephony sales are 486882.05, and Fashion Bags and Accessories sales are 218158.28.
* Another category is like:

|  |  |  |
| --- | --- | --- |
| **0** | PERFUMERY | 506738.66 |
| **1** | FURNITURE DECORATION | 1430176.39 |
| **2** | TELEPHONY | 486882.05 |
| **3** | FASHION BAGS AND ACCESSORIES | 218158.28 |
| **4** | BED TABLE BATH | 1712553.67 |
| **...** | ... | ... |
| **69** | CDS MUSIC DVDS | 1199.43 |
| **70** | LA CUISINE | 2913.53 |
| **71** | FASHION CHILDREN'S CLOTHING | 785.67 |
| **72** | PC GAMER | 2174.43 |
| **73** | INSURANCE AND SERVICES | 324.51 |

1. **orders paid in instalments:**

* The percentage of orders that were paid in instalments is 99.9981.

**5**. **Customer Count by State:**

* Counts the number of customers from each state.
* The number of customers from each state is visualized in a bar chart, with the highest customer concentrations in the states SP, RJ, MG, and RS.
* The top 5 customers by state:

SP - 41746

RJ – 12852

MG - 11635

RS - 5466

PR - 5045

1. **Monthly Order Count in 2018:**

* The monthly order count for 2018 is displayed in a bar chart.
* The highest order counts were observed in the earlier months January, March, April, May and February etc.

months Total orders

January 7269

March 7211

April 6939

May 6873

February 6728

August 6512

July 6292

June 6167

September 16

October 4

1. **Average products per order by customer city:**

* The maximum number of products ordered, organized by customer city.
* The average scores for the cities are as follows: Padre Carvalho is 7, Celso Ramos is 6.50, Candido Godoi is 6, and Datas is 6.

1. **The total revenue contributed by product category:**

* The percentage of total revenue that each product category contributes.
* The highest distribution percentage is in the categories of bed, table, bath, health, beauty, and computer accessories.
* Top 5 category are: -

Category Percentage Distribution

Bed table bath 10.70

Health Beauty 10.35

Computer Accessories 9.90

Furniture decoration 8.93

Watches present 8.93

1. **The relationship between product price and the frequency of product purchases:**

* The correlation between product price and the frequency of purchases is

-0.1063151.

* The most frequently purchased product, bought 9,670 times, is Health Beauty, which is priced at $130.16.
* In the sports leisure category, the number of purchases is 8,641, and the price is $114.34. In the Cool Stuff category, the number of purchases is 3,796, and the price is $167.36.

1. **The cumulative sales per month for each year:**

* The cumulative sales period covers December 2016 to October 2018.
* In October 2018, the payment is $589.67, and the cumulative sales amount to 16008872.12.
* The payment for December 2017 is $878401.48, and the total cumulative sales amount to 7309109.07.
* The payment for December 2016 is $19.62, and the cumulative sales amount is 59362.34.

1. **The year-over-year growth rate of total sales:**

* In the year 2018, the growth rate percentage was 20.000924.
* In the year 2017, the growth rate percentage was 12112.703761.
* In 2016, the growth rate is recorded as not a number (NaN).

1. **Customer retention rate is the percentage of repeat purchases within six months:**

* None of the retention rate of customers as the percentage of customers who make another purchase within 6 months of their first purchase.