In Summary;

I analyzed Walmart sales data by leveraging Python and MySQL to derive meaningful insights and answer key business questions. Using Python, I began by importing the necessary libraries for data manipulation and analysis. I explored the dataset to understand its structure, reviewed data types, and generated descriptive statistics to gain an overview of the data. Through Python's mathematical and analytical capabilities, I calculated various metrics and summarized trends that provided a foundation for deeper analysis.

Next, I utilized MySQL to further analyze the data and address specific queries. By writing SQL queries, I extracted and aggregated data to uncover patterns and relationships within the sales data. This combined approach of Python and MySQL allowed me to efficiently process, analyze, and present results that can support data-driven decision-making for Walmart's sales strategy. For each code, you can find the explanation above.

PYTHON;

```
import pandas as pd
df = pd.read_csv('C:\\Users\\Asus\\OneDrive\\Masaüstü\\walmart.csv')
df.shape
df.head()
df.describe()
df.info()
df.duplicated().sum()
df.isnull().sum()
df.dtypes
df.columns
df['unit_price'] = df['unit_price'].replace('[\$,]', '', regex=True).astype(float)

df['total'] = df['unit_price'] * df['quantity']
df.head()
import pymysql
from sqlalchemy import create_engine
import psycopg2

help(df.to_sql)
help(create_engine)

#host = localhost
#port = 3306
#user = root
#password = Mine12345.
```

```
engine_mysql =
create_engine("mysql+pymysql://root:Mine12345.@localhost:3306/walmart_db")

try:

   engine_mysql

   print("successfull")

except:

   print("unsuccessfull")

df.to_sql(name='walmart', con = engine_mysql, if_exists='append', index = False)

df.columns

df.columns = df.columns.str.lower()

df.columns
```

MYSQL;

```
SELECT
*
FROM
walmart;

SELECT
COUNT(*)
FROM
walmart;

SELECT
payment_method, COUNT(*)
FROM
walmart
group by payment_method;

SELECT
COUNT(DISTINCT branch)
FROM
walmart;

SELECT
MIN(quantity)
FROM
walmart;
```

```
-- different payment method, number of transaction, number of quantity sold
        SELECT
        payment_method, COUNT(*) as no_payments, SUM(quantity) as no_qty_sold
        FROM
        walmart
        GROUP BY payment_method;

-- highest rated category in each brand, displaying the branch, category AVG rated
        SELECT
        *
        FROM
        (SELECT
        Branch, category, AVG(rating) as avg_rating,
        RANK() OVER(PARTITION BY Branch ORDER BY AVG(rating) DESC) as rank_rank
        FROM
        walmart
        group by 1,2
        ) AS t
        WHERE rank_rank=1;
```

```
-- identify the busiest day for each branch based on transcation
        SELECT
        *
        FROM
        (SELECT
        branch,
        date_format(STR_TO_DATE(date, '%d/%m/%y'), '%W') AS day_name,
        COUNT(*) as no_trans,
        RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) desc) as rankk
        FROM
        walmart
        group by 1,2) AS t
        WHERE rankk = 1;

-- total wuantitiy of items sold per payment method
        SELECT
        payment_method, SUM(quantity) as no_qty_sold
        FROM
        walmart
        GROUP BY payment_method;

-- determine the average, min and max rating of prodcts for each city
        SELECT
        city, category, MIN(rating), MAX(rating), AVG(rating)
        FROM
        walmart
        GROUP BY 1,2;
```

-- CALCULATE THE Total profit for each category by considering total profit as unit price* quantity* proft margin.

```
SELECT
category, SUM(total) as revenue, SUM(total*profit_margin) as profit
FROM
walmart
GROUP BY 1;
```

-- most commont payment method for eacg branch

```
WITH cte
AS
(SELECT
Branch, payment_method, COUNT(*) total_trans,
RANK() OVER(PARTITION BY Branch ORDER BY COUNT(*) DESC) as rankk
FROM
walmart
GROUP BY 1,2
)
SELECT *
FROM cte
WHERE rankk=1;
```