# **Coffee Shop Sales Analysis with SQL and Power BI**

**Step By Step information for SQL :**

1. Download the Project.CSV File
2. Create a database for Coffee shop
3. Import Project.csv File and Give Table Name as “Project“
4. After import select New Query
5. Write a Query :- Select \* From Project;
6. Result Of above query and it will return :- 149116 rows are affected

**Step By Step information for Power BI desktop :-**

1. Open a Power Bi Desktop and Click on get data as SQL
2. Configure the Connections of SQL
3. Select the database and load the data in power BI

# **Problem statement for BI reports creation and Calculations**

1. **Total Sales Over Time:**
   * **Plot the total bill over transaction dates to visualize the trend of sales over time.**
2. **Sales Distribution by Product Category:**
   * **Create a bar plot showing the total sales for each product category to understand which categories contribute the most to overall sales.**
3. **Hourly Sales Trend:**
   * **Plot the number of transactions or total sales amount for each hour of the day to identify peak hours of sales activity.**
4. **Monthly Sales Trend:**
   * **Visualize the total sales amount or number of transactions for each month to observe any seasonal patterns.**
5. **Product Type Contribution to Total Sales:**
   * **Create a pie chart or bar plot showing the contribution of each product type to total sales.**
6. **Store-wise Sales Comparison:**
   * **Compare the total sales amount or number of transactions across different store locations to identify top-performing stores.**
7. **Day-wise Sales Variation:**
   * **Plot the total sales amount or number of transactions for each day of the week to understand variations in sales activity.**

**Step By Step information for Report Creation in Jupyter :-**

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame containing the transaction data

df = pd.read\_csv("Project.csv")

print(df.describe())

#checking the null values present in data set or not and in which column has null values.

print(df.isnull().sum())

# Total Sales Over Time

plt.figure(figsize=(10, 6))

sns.lineplot(x='transaction\_date', y='Total\_Bill', data=df)

plt.title('Total Sales Over Time')

plt.xlabel('Transaction Date')

plt.ylabel('Total Bill')

# Sales Distribution by Product Category

plt.figure(figsize=(10, 6))

sns.barplot(x='product\_category', y='Total\_Bill', data=df)

plt.title('Sales Distribution by Product Category')

plt.xlabel('Product Category')

plt.ylabel('Total Bill')

# Hourly Sales Trend

plt.figure(figsize=(10, 6))

sns.lineplot(x='Hour', y='Total\_Bill', data=df)

plt.title('Hourly Sales Trend')

plt.xlabel('Hour')

plt.ylabel('Total Bill')

# Monthly Sales Trend

plt.figure(figsize=(10, 6))

sns.lineplot(x='Month', y='Total\_Bill', data=df)

plt.title('Monthly Sales Trend')

plt.xlabel('Month')

plt.ylabel('Total Bill')

# Product Type Contribution to Total Sales

plt.figure(figsize=(10, 6))

df.groupby('product\_type')['Total\_Bill'].sum().plot(kind='bar')

plt.title('Product Type Contribution to Total Sales')

plt.xlabel('Product Type')

plt.ylabel('Total Bill')

# Store-wise Sales Comparison

plt.figure(figsize=(10, 6))

sns.barplot(x='store\_location', y='Total\_Bill', data=df)

plt.title('Store-wise Sales Comparison')

plt.xlabel('Store Location')

plt.ylabel('Total Bill')

# Day-wise Sales Variation

plt.figure(figsize=(10, 6))

sns.barplot(x='Day Name', y='Total\_Bill', data=df)

plt.title('Day-wise Sales Variation')

plt.xlabel('Day of the Week')

plt.ylabel('Total Bill')