**1. Customer Sales Analysis**

**Objective:** Analyze sales data to find total sales, average purchase, and top customers.

**Table Structure**

CREATE TABLE sales (

id INT PRIMARY KEY AUTO\_INCREMENT,

customer\_id INT,

amount DECIMAL(10,2),

date DATE

);

**Sample Data**

INSERT INTO sales (customer\_id, amount, date) VALUES

(1, 200.50, '2024-02-01'),

(2, 150.00, '2024-02-01'),

(1, 300.75, '2024-02-02'),

(3, 500.20, '2024-02-02');

**Python Code**

import numpy as np

import pymysql

conn = pymysql.connect(host='localhost', user='root', password='', database='analytics')

cursor = conn.cursor()

cursor.execute("SELECT amount FROM sales")

data = np.array(cursor.fetchall())

print("Total Sales:", np.sum(data))

print("Average Sale Amount:", np.mean(data))

print("Highest Sale:", np.max(data))

conn.close()

**2. Employee Performance Analysis**

**Objective:** Calculate the average performance rating per department.

**Table Structure**

CREATE TABLE employees (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

department VARCHAR(50),

performance\_rating INT

);

**Sample Data**

INSERT INTO employees (name, department, performance\_rating) VALUES

('Alice', 'IT', 4),

('Bob', 'HR', 3),

('Charlie', 'IT', 5),

('David', 'Finance', 4);

**Python Code**

cursor.execute("SELECT department, performance\_rating FROM employees")

data = cursor.fetchall()

departments = {}

for dept, rating in data:

if dept not in departments:

departments[dept] = []

departments[dept].append(rating)

for dept, ratings in departments.items():

print(f"{dept} Average Rating: {np.mean(ratings)}")

**3. Website Traffic Analysis**

**Objective:** Analyze visitor trends.

**Table Structure**

CREATE TABLE traffic (

id INT PRIMARY KEY AUTO\_INCREMENT,

visitor\_id INT,

page\_views INT,

visit\_date DATE

);

**Python Code**

cursor.execute("SELECT page\_views FROM traffic")

views = np.array(cursor.fetchall())

print("Total Page Views:", np.sum(views))

print("Average Page Views per Visit:", np.mean(views))

**4. Product Demand Forecasting**

**Objective:** Predict future demand.

**Table Structure**

CREATE TABLE demand (

id INT PRIMARY KEY AUTO\_INCREMENT,

product\_id INT,

sales INT,

month VARCHAR(20)

);

**Python Code**

cursor.execute("SELECT sales FROM demand")

sales = np.array(cursor.fetchall())

future\_demand = np.mean(sales) + np.std(sales)

print("Predicted Demand:", future\_demand)

**5. Financial Risk Analysis**

**Objective:** Identify customers with overdue payments.

**Table Structure**

CREATE TABLE payments (

id INT PRIMARY KEY AUTO\_INCREMENT,

customer\_id INT,

amount\_due DECIMAL(10,2),

due\_date DATE

);

**Python Code**

cursor.execute("SELECT amount\_due FROM payments WHERE due\_date < CURDATE()")

overdue = np.array(cursor.fetchall())

print("Total Overdue:", np.sum(overdue))

**6. Customer Retention Analysis**

**Objective:** Calculate churn rate.

**Table Structure**

CREATE TABLE customers (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(50),

last\_purchase DATE

);

**Python Code**

cursor.execute("SELECT last\_purchase FROM customers")

dates = np.array(cursor.fetchall())

churn\_rate = len(dates[dates < np.datetime64('2024-01-01')]) / len(dates)

print("Churn Rate:", churn\_rate)

**7. Revenue Forecasting**

**Objective:** Predict revenue based on historical data.

**Table Structure**

CREATE TABLE revenue (

id INT PRIMARY KEY AUTO\_INCREMENT,

month VARCHAR(20),

earnings DECIMAL(10,2)

);

**Python Code**

cursor.execute("SELECT earnings FROM revenue")

earnings = np.array(cursor.fetchall())

forecast = np.mean(earnings) + np.std(earnings)

print("Projected Revenue:", forecast)

**8. Inventory Management**

**Objective:** Identify low-stock products.

**Table Structure**

CREATE TABLE inventory (

id INT PRIMARY KEY AUTO\_INCREMENT,

product VARCHAR(50),

stock INT

);

**Python Code**

cursor.execute("SELECT stock FROM inventory")

stock\_levels = np.array(cursor.fetchall())

low\_stock = stock\_levels[stock\_levels < np.mean(stock\_levels) - np.std(stock\_levels)]

print("Low Stock Items:", len(low\_stock))

**9. Credit Score Analysis**

**Objective:** Identify risky borrowers.

**Table Structure**

CREATE TABLE credit\_scores (

id INT PRIMARY KEY AUTO\_INCREMENT,

customer\_id INT,

score INT

);

**Python Code**

cursor.execute("SELECT score FROM credit\_scores")

scores = np.array(cursor.fetchall())

risky\_customers = scores[scores < np.mean(scores) - np.std(scores)]

print("High-Risk Customers:", len(risky\_customers))

**10. Social Media Engagement Analysis**

**Objective:** Find the average engagement rate.

**Table Structure**

CREATE TABLE engagement (

id INT PRIMARY KEY AUTO\_INCREMENT,

post\_id INT,

likes INT,

comments INT

);

**Python Code**

cursor.execute("SELECT likes, comments FROM engagement")

engagement = np.array(cursor.fetchall())

engagement\_rate = np.mean(engagement[:, 0] + engagement[:, 1])

print("Average Engagement Rate:", engagement\_rate)