**1. Customer Purchase Analysis**

**Tables**

**customers**

| **customer\_id** | **name** | **age** | **gender** | **city** |
| --- | --- | --- | --- | --- |
| 1 | John | 28 | Male | New York |
| 2 | Alice | 32 | Female | London |

**orders**

| **order\_id** | **customer\_id** | **product** | **price** | **purchase\_date** |
| --- | --- | --- | --- | --- |
| 101 | 1 | Laptop | 800 | 2024-02-01 10:30:00 |
| 102 | 2 | Headphone | 50 | 2024-02-03 12:00:00 |

**Python Code**

import mysql.connector

import pandas as pd

conn = mysql.connector.connect(host='localhost', user='root', password='', database='analytics\_db')

cursor = conn.cursor()

query = """

SELECT c.name, c.city, o.product, o.price, o.purchase\_date

FROM customers c

JOIN orders o ON c.customer\_id = o.customer\_id

"""

df = pd.read\_sql(query, conn)

print(df)

conn.close()

**2. Employee Performance Dashboard**

**Tables**

**employees**

| **emp\_id** | **name** | **department** | **salary** | **hire\_date** |
| --- | --- | --- | --- | --- |
| 1 | Mark | IT | 50000 | 2020-01-15 |
| 2 | Lisa | HR | 45000 | 2021-06-01 |

**performance**

| **emp\_id** | **year** | **rating** |
| --- | --- | --- |
| 1 | 2023 | 4.5 |
| 2 | 2023 | 4.0 |

**Python Code**

query = """

SELECT e.name, e.department, e.salary, p.year, p.rating

FROM employees e

JOIN performance p ON e.emp\_id = p.emp\_id

"""

df = pd.read\_sql(query, conn)

print(df)

**3. Sales Revenue Prediction**

**Tables**

**sales**

| **sale\_id** | **product** | **quantity** | **price** | **sale\_date** |
| --- | --- | --- | --- | --- |
| 1 | Phone | 5 | 300 | 2024-01-10 |
| 2 | Laptop | 2 | 1000 | 2024-01-12 |

**Python Code (Predict Sales Revenue)**

query = "SELECT product, SUM(quantity \* price) AS revenue FROM sales GROUP BY product"

df = pd.read\_sql(query, conn)

print(df)

**4. Online Course Enrollment Trend**

**Tables**

**courses**

| **course\_id** | **course\_name** | **category** |
| --- | --- | --- |
| 1 | Python Basics | Programming |
| 2 | Data Science | AI |

**enrollments**

| **enroll\_id** | **course\_id** | **student\_name** | **enroll\_date** |
| --- | --- | --- | --- |
| 1 | 1 | Tom | 2024-02-05 |
| 2 | 2 | Emily | 2024-02-07 |

**Python Code**

query = """

SELECT c.course\_name, COUNT(e.enroll\_id) AS enroll\_count

FROM courses c

JOIN enrollments e ON c.course\_id = e.course\_id

GROUP BY c.course\_name

"""

df = pd.read\_sql(query, conn)

print(df)

**5. Hotel Room Booking Analysis**

**Tables**

**hotels**

| **hotel\_id** | **hotel\_name** | **location** |
| --- | --- | --- |
| 1 | Grand Palace | Paris |
| 2 | Ocean View | Miami |

**bookings**

| **booking\_id** | **hotel\_id** | **customer** | **check\_in** | **check\_out** |
| --- | --- | --- | --- | --- |
| 1 | 1 | James | 2024-03-01 | 2024-03-05 |
| 2 | 2 | Sarah | 2024-03-02 | 2024-03-06 |

**Python Code**

query = """

SELECT h.hotel\_name, h.location, COUNT(b.booking\_id) AS total\_bookings

FROM hotels h

JOIN bookings b ON h.hotel\_id = b.hotel\_id

GROUP BY h.hotel\_name

"""

df = pd.read\_sql(query, conn)

print(df)

**6. Social Media Engagement Analysis**

**Tables**

**posts**

| **post\_id** | **user\_name** | **likes** | **comments** | **shares** | **post\_date** |
| --- | --- | --- | --- | --- | --- |
| 1 | Alex | 120 | 30 | 20 | 2024-02-08 |
| 2 | Linda | 200 | 50 | 40 | 2024-02-10 |

**Python Code**

query = "SELECT user\_name, SUM(likes + comments + shares) AS total\_engagement FROM posts GROUP BY user\_name"

df = pd.read\_sql(query, conn)

print(df)

**7. Hospital Patient Admission Analysis**

**Tables**

**patients**

| **patient\_id** | **name** | **age** | **disease** | **admitted\_date** |
| --- | --- | --- | --- | --- |
| 1 | Jack | 45 | Diabetes | 2024-02-11 |
| 2 | Lily | 50 | Heart Issue | 2024-02-12 |

**Python Code**

query = "SELECT disease, COUNT(patient\_id) AS patient\_count FROM patients GROUP BY disease"

df = pd.read\_sql(query, conn)

print(df)

**8. E-commerce Website Traffic Analysis**

**Tables**

**website\_visits**

| **visit\_id** | **page\_visited** | **user\_location** | **time\_spent** |
| --- | --- | --- | --- |
| 1 | Homepage | USA | 5 min |
| 2 | Product | UK | 10 min |

**Python Code**

query = "SELECT page\_visited, COUNT(visit\_id) AS visit\_count FROM website\_visits GROUP BY page\_visited"

df = pd.read\_sql(query, conn)

print(df)

**9. Library Book Borrowing Analysis**

**Tables**

**books**

| **book\_id** | **title** | **genre** |
| --- | --- | --- |
| 1 | Python Tricks | Programming |
| 2 | AI for Good | AI |

**borrowed\_books**

| **borrow\_id** | **book\_id** | **user\_name** | **borrow\_date** |
| --- | --- | --- | --- |
| 1 | 1 | Kevin | 2024-02-13 |
| 2 | 2 | Anna | 2024-02-14 |

**Python Code**

query = "SELECT b.title, COUNT(bb.borrow\_id) AS borrow\_count FROM books b JOIN borrowed\_books bb ON b.book\_id = bb.book\_id GROUP BY b.title"

df = pd.read\_sql(query, conn)

print(df)

**10. Movie Rating Analysis**

**Tables**

**movies**

| **movie\_id** | **title** | **genre** |
| --- | --- | --- |
| 1 | Inception | Sci-Fi |
| 2 | Titanic | Romance |

**ratings**

| **rating\_id** | **movie\_id** | **user\_name** | **rating** |
| --- | --- | --- | --- |
| 1 | 1 | Henry | 5 |
| 2 | 2 | Sophia | 4 |

**Python Code**

query = "SELECT m.title, AVG(r.rating) AS avg\_rating FROM movies m JOIN ratings r ON m.movie\_id = r.movie\_id GROUP BY m.title"

df = pd.read\_sql(query, conn)

print(df)