FITNESS ANALYSIS

The Gym Members dataset contains detailed information about individuals enrolled in a fitness center or gym. It records each member’s personal details, fitness goals, assigned trainers, membership plans, and health statistics such as height, weight, and BMI.

This dataset can be used for membership analysis, trainer performance evaluation, health trend analysis, and customer retention studies.

Dataset Summary

Feature	Description
Number of Records (Rows)	~32 (members)
Number of Columns	13
Data Type	Tabular (Structured)
File Type	CSV / Excel
Domain	Health & Fitness, Customer Analytics

SQL Queries (Basic → Moderate)

1️⃣ Basic – Show all members

```sql
SELECT * FROM gym_members;
```

2️⃣ Count total members

```sql
SELECT COUNT(*) AS TotalMembers FROM gym_members;
```

3️⃣ Find all members with Active membership

```sql
SELECT UserName, Membership, MembershipEnd
FROM gym_members
WHERE Status = 'Active';
```

4️⃣ Count members by Gender

```sql
SELECT Gender, COUNT(*) AS Total
FROM gym_members
GROUP BY Gender;
```

5️⃣ List members who joined after July 2024

```sql
SELECT UserName, JoinDate
FROM gym_members
WHERE JoinDate > '2024-07-01';
```

6️⃣ Average BMI by Goal

```sql
SELECT Goal, ROUND(AVG(BMI), 2) AS Avg_BMI
FROM gym_members
GROUP BY Goal
ORDER BY Avg_BMI DESC
```

7️⃣ Find top 5 members with highest BMI

```
SELECT UserName, BMI
FROM gym_members
ORDER BY BMI DESC
LIMIT 5;
```

8️⃣ Count members by Membership Type and Status

```
SELECT Membership, Status, COUNT(*) AS Total
FROM gym_members
GROUP BY Membership, Status
ORDER BY Membership, Status;
```

📊 Python Visualization (using Pandas + Matplotlib / Seaborn)

> Assume file name: `gym_data.csv`

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.read_csv("gym_data.csv")
```

1️⃣ Bar chart – Count of members by Gender

```
sns.countplot(x='Gender', data=df)
plt.title("Members by Gender")
plt.show()
```

2️⃣ Pie chart – Membership Status distribution

```
df['Status'].value_counts().plot.pie(autopct='%1.1f%%', startangle=90)
plt.title("Membership Status Distribution")
plt.ylabel('')
plt.show()
```

3️⃣ Histogram – BMI distribution

```
sns.histplot(df['BMI'], bins=10, kde=True)
plt.title("BMI Distribution")
plt.show()
```

4️⃣ Boxplot – BMI by Goal

```
sns.boxplot(x='Goal', y='BMI', data=df)
plt.title("BMI by Goal Type")
plt.xticks(rotation=45)
plt.show()
```

5️⃣ Bar chart – Average BMI by Gender

```
avg_bmi = df.groupby('Gender')['BMI'].mean().reset_index()
sns.barplot(x='Gender', y='BMI', data=avg_bmi)
plt.title("Average BMI by Gender")
plt.show()
```

6️⃣ Line chart – Membership duration (start to end)

```
df['MembershipStart'] =
pd.to_datetime(df['MembershipStart'])
df['MembershipEnd'] =
pd.to_datetime(df['MembershipEnd'])
df['Duration_Days'] = (df['MembershipEnd'] -
df['MembershipStart']).dt.days

sns.lineplot(x='UserName', y='Duration_Days',
data=df)
plt.xticks(rotation=90)
plt.title("Membership Duration per User")
plt.show()
```

7️⃣ Countplot – Members by Trainer Name

```
sns.countplot(y='Trainer Name', data=df,
order=df['Trainer Name'].value_counts().index)
plt.title("Members Trained by Each Trainer")
plt.show()
```

8️⃣ Scatterplot – Weight vs Height

```
sns.scatterplot(x='Height_cm', y='StartingWeight',
hue='Gender', data=df)
plt.title("Height vs Starting Weight")
plt.show()
```