

## **Query specific instructions on how to run our python program**

**1. How many trainers are associated with each type of diet plan? And how many members are following each type of diet plan? And (keto, diabetic etc.)?**

This query retrieves the number of trainers and members associated with each diet plan, such as keto and diabetic. Because it is a summary query, it was designed to not take any specific prompts from the user. By default, it shows all the trainers and the respective diet plan made by each trainer

**2.What times of day do most members come to the gym?**

In this query, we programmed to allow the number of results displayed to be customized by the user. Prompting the user to enter the number of results they want to view might be useful because it allows the user to customize the output to their specific needs. For example, if the user only wants to see the top 3 peak hours instead of the default number of results, they can enter "3" at the prompt and only the top 3 results will be displayed. This can save the user time and help them focus on the most important information. Additionally, it can make the code more flexible and reusable for different situations, as the user can input different limits based on their needs.

**3. What are the most commonly used payment methods from both expenses and payments?**

In this query, we are displaying the payment methods and total number of transactions performed using that method. we programmed to allow the number of results displayed to be customised by the user. Prompting the user to enter the number of results they want to view might be useful because it allows the user to customise the output to their specific needs. For example, if the user only wants to see the top 3 most commonly used payment methods instead of the default number of results, they can enter "3" at the prompt and only 3 results will be displayed. This can save the user time and help them focus on the most important information. Additionally, it can make the code more flexible and reusable for different situations, as the user can input different limits based on their needs.

**4. How much of each equipment type does the gym currently have?**

In this query returns a count for each type of equipment the gym currently has on hand. By tracking the quantity of each equipment type, owners can ensure they have enough equipment to meet demand, anticipate replacements, and plan for any expansions.

In addition to that here, we programmed to allow the number of results displayed to be customized by the user. Prompting the user to enter the name of the equipment the gym currently has as input in it might be useful because it allows the user to customize the output

to their specific needs. For example, if we give the input as "Treadmill" it will give the number of treadmills we have present in the gym.

### **5. How many members is each trainer currently training?**

In this query, we programmed to display the trainer's first name and number of members he is currently training. In addition to that, the user can access the member count of each trainer just by giving the trainer's first name as input. If a user inputs a wrong trainer name, the program will continue to prompt the user to enter the correct trainer name until he enters the correct one. For example, a user entered the name Cary. But as Cary is not present in the database as the trainer's name, it will prompt the user to enter a correct name. If the user enters a correct name, let's say Karry. The program will display the trainer's first name and number of members he is currently training.

### **6. How many members are signed up for each type of membership plan?**

This query returns every membership plan (membership\_id) and the number of members who are signed up for that plan. This information can be used to maximize revenue by adjusting the membership offerings based on the popularity of membership plans.

In addition to that here, we programmed to allow the results displayed to be customized by the user. Prompting the user to enter the membership id as input and it will give a description of the membership plan and the number of members following that particular plan, it might be useful because it allows the user to customize the output to their specific needs.

### **7. How does the amount of time each member spend in the gym compare to their scheduled hours for a week?**

This query displays all the member ids, their scheduled hours, and time spent in the gym for a week. In addition to that, the user can enter the member id to access the scheduled hours and time spent by that member in the gym. As there are only 20 members in the gym, it takes input only between 1 to 20. If the user enters any other number, it will prompt the user to enter the correct member id until he enters a correct one. Then, the program displays the member id, their scheduled hours and time spent by that member.

### **8. Are there any rejected transactions and which members have not paid the gym?**

This query shows all the rejected payment transactions and the associated member. This information will allow owners and administrators to reach out to members who have not paid their membership fees to collect updated payment information and any money owed to the gym.

In addition to that here, we programmed to allow the results displayed to be customized by the user. Prompting the user to enter the member id as input and it will give member's first

name payment amount & payment date of their rejected transaction, It might be useful because it allows the user to customize the output to their specific needs.

## **9. What is the ratio of trainers to members?**

This query shows the number of members divided by the number of trainers. It provides a quick overview of whether or not the gym needs to consider hiring more trainers to keep up with demand.