

Project 2

Gym Membership



Anjala Saikia

**Introduction to Gym Membership**

The health and fitness industry is booming as people prioritize their well-being, making gyms a significant part of their routines. As gym operations grow in complexity with increasing memberships, manual data handling becomes inefficient. To address this challenge, I developed a Gym Membership Management System as a proof-of-concept application that automates membership registration, calculates BMI and fitness levels, and tracks revenue distribution by membership type. This project serves as a practical demonstration of using SQL, Excel front-end, and VBA middleware to create an efficient, user-friendly system tailored for gym administrators. Through this project, I learned to bridge gaps between databases and user-friendly interfaces, design efficient data models, and implement real-world logic in an interactive system.

The application provides insights into member fitness trends, membership statuses, and financial performance, ensuring scalability for broader business applications.This project demonstrates automation for membership registration, fitness tracking, revenue management, and reporting. Through this initiative, I not only honed my technical skills in SQL and VBA but also learned to analyze business workflows and optimize user interactions.

**Understanding the Business**

The gym industry operates on a subscription model where each member has unique fitness needs and timelines. Analyzing gym operations revealed the following pain points:

* **Membership Tracking:** Manual systems led to inconsistencies in monitoring membership expiry, fitness levels, and renewals.
* **Health Analysis:** Categorizing members based on BMI and assigning fitness levels was cumbersome.
* **Revenue Management:** Calculating and visualizing revenue distribution across membership tiers was time-intensive.
* **Data Visibility:** Retrieving member details required significant manual effort.

**Project Objectives**

To address these challenges, I set out to:

1. **Automate Membership Management:** Ensure seamless member data storage, retrieval, and updates.
2. **Enhance Health Tracking:** Automate BMI calculations and fitness level categorization.
3. **Optimize Revenue Analysis:** Provide insights into revenue distribution across membership types.
4. **Streamline Reporting:** Create dashboards for easy data visualization and access.

The project focused on designing a system that integrates seamlessly into the gym's daily operations while being user-friendly and scalable for future needs.

**Structure Overview**

The Access database serves as the backbone of the system, with normalized tables to ensure efficient data handling and eliminate redundancy. Below is a detailed breakdown of the tables:

1. **MembersList**:
   * **Fields**: MemberID, FullName, MembershipJoinDate, MembershipEndDate, ContactNumber, Email, ClassID, MembershipType, Gender, Height and many more such fields.
   * **Purpose**: This table holds essential membership details, acting as the core repository for member records.
2. **MembersData**:
   * **Fields**: MemberID, Gender, Weight(kg), Height(cm), MembershipStatus.
   * **Purpose**: Stores health metrics like weight and height, which are used to calculate BMI and assign fitness levels.
3. **MembershipType**:
   * **Fields**: MembershipType, Price, and Validity.
   * **Purpose**: Stores membership plans (Gold, Silver, Bronze), their pricing, and duration.
4. **BMI\_RANGE**:
   * **Fields**: BMI and FitnessLevel.
   * **Purpose**: Maps BMI values to corresponding fitness categories like "Underweight" or "Healthy Weight."

**Key Queries**

1. **BMI and Fitness Level Calculation**: Used SQL query to calculate BMI for each member in the MembersData table using the formula above, rounded to two decimal places. Based on BMI, it categorizes members into fitness levels—Underweight, Healthy weight, Overweight, Obese, and Severely Obese—using nested IIF statements.This query is crucial for health analysis, providing insights into members' fitness levels and enabling the gym to offer tailored plans. It automates BMI classification efficiently, improving data readability and decision-making.
2. **Analysis of the Queries:**

**Gender-Based Count Query**:

**Purpose**: Counts the total number of members based on their gender (Male/Female).

**Key Functions**:

COUNT: Used to calculate the total members for each gender.

GROUP BY: Groups the members by their gender to aggregate the data effectively.

**Significance**: Provides insights into the gender distribution of gym members, helping to analyze target demographics.

1. **Membership Revenue Query**:

**Purpose**: Calculates the total revenue generated by each membership type (Bronze, Silver, Gold).

**Key Functions**:

SUM: Sums up the membership prices for each type.

GROUP BY: Aggregates the total revenue by membership type.

**Significance**: Useful for understanding which membership tier is the most profitable and making pricing decisions.

1. **Membership Type Count Query**:

**Purpose**: Counts the total number of members in each membership category.

**Key Functions**:

COUNT: Counts the occurrences of each membership type.

GROUP BY: Groups the data by membership type.

**Significance**: Helps identify the popularity of different membership tiers, aiding in marketing and strategy development.

These queries collectively provide valuable insights into the membership data, enabling data-driven decision-making for better business outcomes.

**VBA Middleware**

The VBA middleware serves as the backbone of the project, seamlessly connecting the Excel front-end with the Access database. It automates key workflows, validates inputs, and ensures smooth data transfer between the two systems.

**Key Subroutines**

**1. ImportRecords\_Simplified**

* **Purpose**: Fetches and displays member records dynamically from the Access database to the Excel dashboard.
* **Highlights**:
  + **Why CopyFromRecordset**: This method is optimized for transferring large datasets from a database to Excel, ensuring quick and efficient updates.
  + **Data Formatting**: Columns are auto-fitted, and conditional formatting is preserved for a professional look and to differentiate between expired and valid memberships.
  + **Error Handling**: User-friendly error messages ensure smooth execution even if the database or query encounters issues.

**2. SearchRecords**

* **Purpose**: Allows users to search for specific members based on either their MemberID or FullName.
* **Highlights**:
  + **Why InStr**: Enables partial matching for a more user-friendly search experience. Users can search with partial names or IDs without needing to input exact values.
  + **Row Highlighting**: Matching records are highlighted in light yellow, making results visually distinguishable.
  + **Feedback**: Provides instant messages if no results are found, improving usability.

**3. CalculateMembershipEndDate**

* **Purpose**: Calculates and displays the membership expiration date based on the selected membership type and start date.

**Why DateAdd:**

* Simplifies date manipulation by automatically adding the specified number of months to the membership start date. This ensures accurate and consistent calculation of membership end dates based on the type of membership plan (e.g., Gold = 10 months, Silver = 8 months).

The DateAdd function is a built-in VBA function used for date calculations, such as adding or subtracting days, months, or years to a specific date. For example, if the start date is 01-Jan-2025 and the membership type is Gold (10 months), DateAdd("m", 10, "01-Jan-2025") will calculate the end date as 01-Nov-2025.

* **Dynamic Logic**: Uses a Select Case structure to assign different durations (e.g., Gold = 10 months, Silver = 8 months) based on the membership type.
* **Error Prevention**: Locked and hidden the Membership End Date field to ensure users don’t accidentally overwrite it, maintaining data integrity.

**4. InsertIntoMembersList**

* **Purpose**: Inserts new member details into the database from the Membership Form in Excel.
* **Highlights**:
  + **Validations**: Ensures all required fields (e.g., FullName, Membership Type, Weight, Height) are filled and valid before submission.
  + **Error Handling**: Implements robust error handling to provide clear feedback when data is missing or invalid, improving user experience.
  + **Integration**: Automatically calculates fields like BMI, Fitness Level, and Membership End Date and transfers them directly into the database, reducing manual effort.

**Working with Buttons**

I designed and implemented four buttons in the Excel front-end, each serving a unique purpose:

1. **Refresh Button**:
   * **Purpose**: Updates the Member Records Dashboard with the latest data from the database.
   * **Integration**: Calls the ImportRecords\_Simplified subroutine to dynamically populate the sheet while preserving formatting and conditional rules.
2. **Search Button**:
   * **Purpose**: Enables users to search for members by entering their MemberID or FullName.
   * **Functionality**: Highlights matching rows and provides instant feedback for unmatched queries.
   * **Benefit**: Simplifies member lookup, especially for a large database.
3. **Calculate End Date Button**:
   * **Purpose**: Automatically calculates and populates the membership end date based on the selected type.
   * **Functionality**: Uses the CalculateMembershipEndDate subroutine to ensure accurate calculations.
   * **Benefit**: Eliminates manual calculation errors and ensures consistency.
4. **Submit Membership Button**:
   * **Purpose**: Adds a new member to the database with validated inputs.
   * **Functionality**: Executes the InsertIntoMembersList subroutine to store all relevant details.
   * **Benefit**: Streamlines the membership registration process, ensuring data accuracy.

**Key Learnings**

1. **SQL Proficiency**:
   * Mastered JOIN operations to link multiple tables for queries.
   * Learned to use IIF, ROUND, and aggregate functions (COUNT, SUM) for conditional logic and summarizing data.
2. **VBA Automation**:
   * Automated workflows, reducing manual intervention and enhancing data accuracy.
   * Gained experience in error handling and input validations for robust functionality.
3. **UI/UX Enhancements**:
   * Balanced functionality and aesthetics in the Excel front-end.
   * Leveraged conditional formatting, pivot tables, and charts to make the system visually intuitive and easy to use.

**Conclusion**

This project demonstrates the seamless integration of SQL, VBA, and Excel to automate gym membership management. Each button and subroutine addresses a specific operational need, from managing memberships to tracking financial trends.

For future scalability, the following features could be added:

* **Online Integration**: Member self-service portals for renewals and updates.
* **Automated Notifications**: Email reminders for membership expiration or renewal.
* **Enhanced Analytics**: Fitness progress tracking for members based on their BMI and fitness levels.

This project showcases how technical expertise can be applied to solve real-world challenges, streamline processes, and provide actionable insights for business growth.

**References**

1. **Books**:
   * Walkenbach, J. (2019). *Excel VBA Programming For Dummies*. Wiley.
   * Alexander, M. & Kusleika, D. (2019). *Access 2019 Bible*. Wiley.
2. **Online Tutorials and Documentation**:
   * Microsoft Docs: [VBA Language Reference](https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/vba-language-reference)
   * Microsoft Support: [Access SQL Basics](https://support.microsoft.com/en-us/office/guide-to-sql-in-access-34c47f85-c657-48a7-89c1-56464fda0005)
3. **YouTube Channels**:
   * "Excel VBA Is Fun" - Tutorials on VBA basics and advanced concepts.
   * "Access Development" - Database creation and integration tips.