CN5000 Group Coursework Report

Hard Joshi Jayrup Nakawala Shyam Jagani Yogi Patel

Table of contents

1 Entities and ERD (10 Marks)

1.1 Identification of Potential Entities

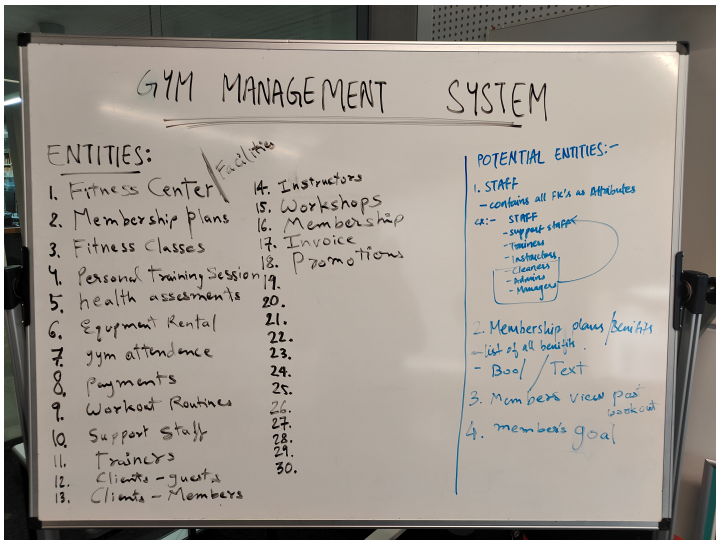

Figure 1: Potential Entities

During the initial phase of the design, we identified **17 entities** as shown in the brainstorming process. These entities were:

1. Fitness Center
2. Membership Plans
3. Fitness Classes
4. Personal Training Sessions
5. Health Assessments
6. Equipment Rentals
7. Gym Attendance
8. Payments
9. Workout Routines
10. Support Staff
11. Trainers
12. Clients - Guests
13. Clients - Members
14. Instructors
15. Workshops
16. Membership
17. Invoice

The process of refinement involved carefully analyzing the identified entities to eliminate redundancies, merge similar entities, and ensure that only relevant entities were retained to represent the system requirements effectively. The following steps were taken during this refinement process:

1. **Combining Overlapping Entities**:
    - We combined *Clients - Guests* and *Clients - Members* into a single entity **Clients** to avoid duplication and simplify the management of client data.
    - We merged *Support Staff*, *Trainers*, and *Instructors* into a single entity **Staff**. This unified representation simplified the model while allowing roles to be distinguished through attributes.
2. **Eliminating Redundant Entities**:
    - *Fitness Center* was removed as it represented a general concept rather than a specific data entity.
    - *Workshops* were treated as a subset of **Fitness_Classes** or **Personal_Training_Sessions**, depending on the context, and therefore were eliminated.
    - *Payments* and *Invoice* were consolidated into **Billing**, which effectively captured all payment-related activities and attributes.
3. **Creating New Relationships**:
    - To ensure the system captured bookings for classes, we introduced a new entity **Class_Bookings**. This entity connected **Clients** and **Fitness_Classes**, representing many-to-many relationships.
    - We introduced **Discounts** to manage promotional offers and incentives, ensuring flexibility within the billing process.
4. **Streamlining Entity Scope**:
    - We refined entities to focus on core data management processes within the Gym Management System. Entities such as *Workout Routines* were excluded as they could be treated as attributes or separate functionalities within **Personal_Training_Sessions**.

The final streamlined list of **12 entities** was as follows:

1. **Clients**
2. **Membership**
3. **Fitness_Classes**
4. **Class_Bookings**
5. **Personal_Training_Sessions**
6. **Health_Assessments**

7. **Gym_Attendance**
8. **Billing**
9. **Discounts**
10. **Staff**
11. **Facilities**
12. **Equipment_Rentals**

This refined set of entities ensured clarity, reduced redundancy, and effectively supported the required system functionalities.

1.3 Appropriate Naming for Entities

We ensured appropriate and consistent naming conventions for all entities to maintain clarity, ease of understanding, and alignment with database design principles. Entities were represented using singular nouns and underscores to separate words where necessary (e.g., **Fitness_Classes**, **Class_Bookings**). These naming conventions aligned with standard database design practices.

1.4 Entity-Relationship Diagram (ERD)

We created the Entity-Relationship Diagram (ERD) to represent the relationships among the final 12 entities. Key design considerations included:

- **Identification of Major Entities**: We retained all major entities critical to the system in the final design.
- **Weak Entities**: We eliminated weak or redundant entities (e.g., "Support Staff" and "Trainers" merged into **Staff**) to simplify the design.
- **Relationships**: We clearly defined the relationships between entities to ensure a logical flow of data within the system.
- **Multiplicity and Optionality**: We annotated each relationship to indicate the multiplicity (one-to-one, one-to-many) and optionality (mandatory/optional) between entities. For example:
  - A **Client** had one or more **Memberships**.
  - **Fitness_Classes** were booked by multiple **Clients** through **Class_Bookings**.
  - **Billing** records were associated with **Clients** and reflected applicable **Discounts**.

The ERD visually illustrated these entities, their attributes, and the relationships that facilitated data integrity and functionality within the **Gym Management System**. We ensured precision in the design using CASE tools to comply with database modeling standards.

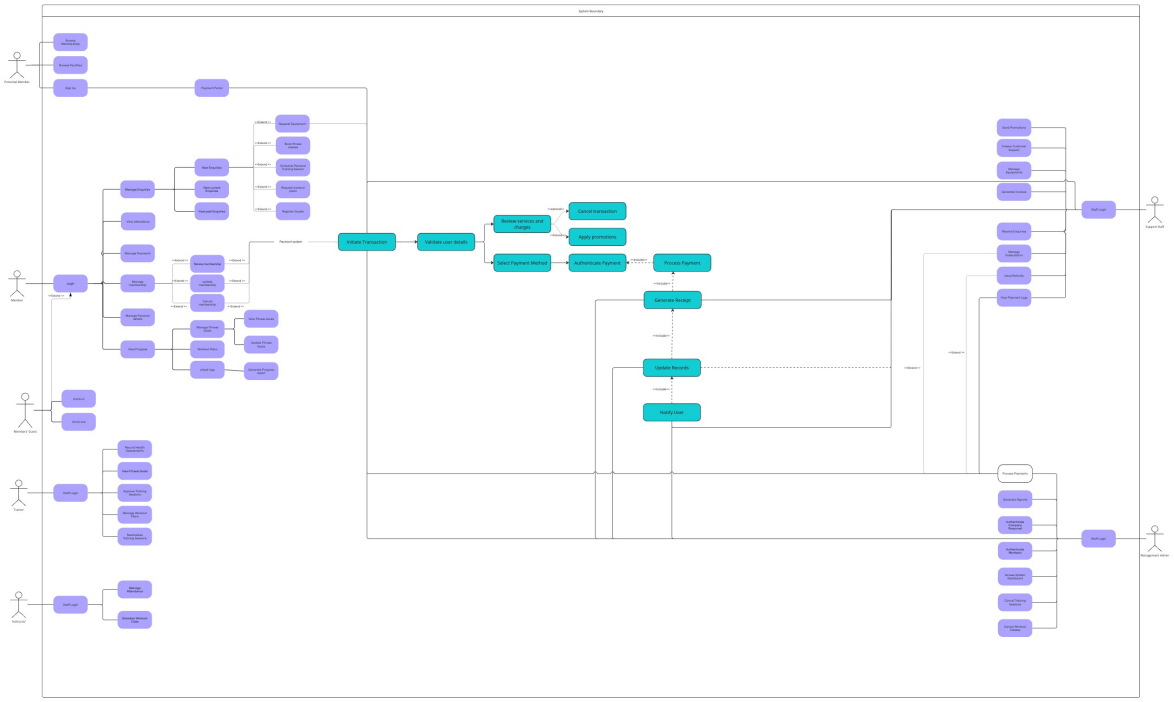

Figure 2: Use Case Diagram

To access the full Use Case Diagram, click here

2. Normalization Process

Normalization is a key process in relational database design that we followed to reduce data redundancy and improve data integrity. It involved organizing tables and attributes into well-defined structures that minimized duplication and eliminated undesirable anomalies (insert, update, and delete anomalies).

The normalization process followed three key stages:

1. **First Normal Form (1NF)**:
    - We ensured all attributes were atomic, i.e., each attribute contained only indivisible values.
    - We removed repeating groups or arrays within tables.

- Example: In a table with Client and Fitness_Class details, classes attended by a client were represented as separate rows rather than a single row with multiple class values.

2. **Second Normal Form (2NF)**:
    - We ensured that all non-key attributes were fully functionally dependent on the primary key.
    - We removed partial dependencies, where non-key attributes depended on only part of a composite primary key.
    - Example: In a table with Class_Bookings, attributes such as Client_Name were not depending only on Class_ID but rather on the composite key **(Client_ID, Class_ID)**.
3. **Third Normal Form (3NF)**:
    - We ensured that all attributes were only dependent on the primary key and not on other non-key attributes.
    - We removed transitive dependencies, where non-key attributes depended on other non-key attributes.
    - Example: In a Billing table, if Discount_Percentage was depending on Discount_ID, this dependency was moved to a separate **Discounts** table.

2.1 Achieving 3NF in the Gym Management System

We ensured that all tables in the Gym Management System adhered to **Third Normal Form (3NF)**. The steps followed included: - Ensuring all attributes were atomic (1NF). - Eliminating partial dependencies (2NF) by clearly defining primary keys. - Removing transitive dependencies (3NF) to ensure attributes only depended on the primary key. For instance: - The **Billing** table referenced **Discount_ID** instead of directly storing discount details. - **Class_Bookings** linked **Clients** and **Fitness_Classes** without duplicating client or class data.

By adhering to these normalization principles, we ensured: - Minimal data redundancy. - Data integrity through consistent and well-structured relationships. - Efficient query performance and database maintenance.

2.2 ERD Diagram

The final ERD, which visually depicted the relationships and multiplicity, was provided as part of the **Appendices** section. It showed: - Entities with primary and foreign keys. - Relationships between tables. - Multiplicity and constraints clearly defined.

This comprehensive approach ensured that the database design was normalized, logically structured, and aligned with the requirements of the Gym Management System.

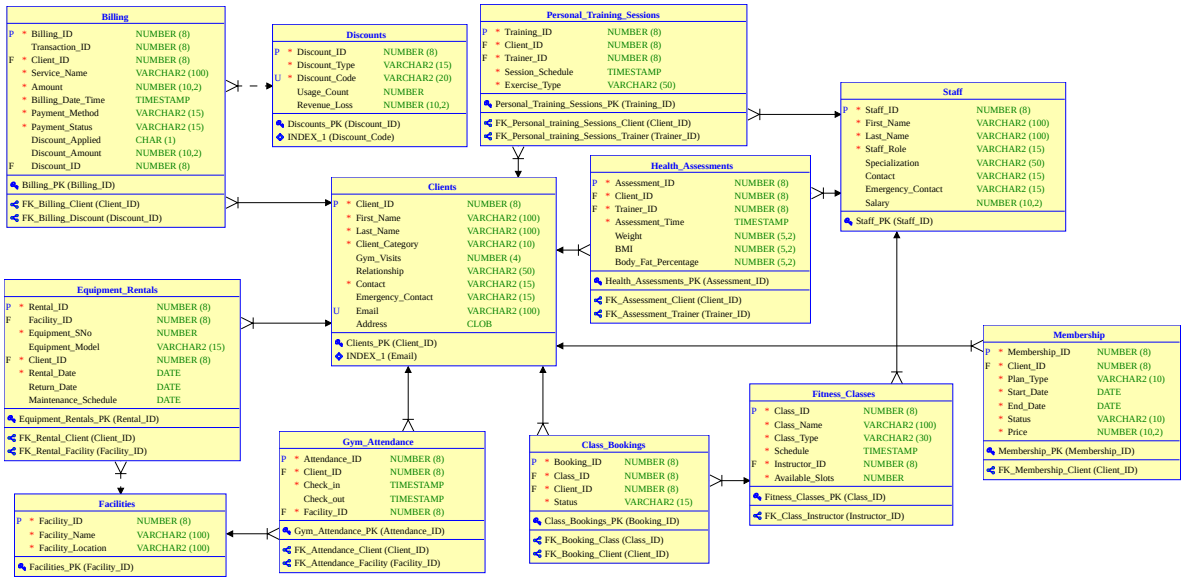

Figure 3: ERD Diagram

---

3. Creation of Tables in SQL and Population of Tables (10 Marks)

3.1 Creating Database in SQL

- **Correct SQL Syntax**:
  - Data types, primary key, foreign key, not null, unique, and check constraints where appropriate.
- **Evidence of Successful Execution**.
- **At Least 10 Records** in each table.

3.2 SQL Queries

- **Correct SQL Syntax**.
- **Evidence of Successful Execution**.

- **Correct SQL Syntax** to perform defined actions.
- **Evidence of Successful Execution**.
- Partial attempts can gain marks if effort is evident.

---

4. Report and Presentation (30 Marks)

4.1 Report Reflection (5 Marks)

- **Principles of Database Design**:
  - Application to the case study.
  - Recognition of alternative solutions.
  - Justification of design choices.
- **Discussion**:
  - Well-written and logically coherent.
  - Evaluates design decisions and alternatives.

4.2 Presentation (20 Marks)

- **Design and Implementation in Oracle**.
  - 10-minute presentation.
  - 5-minute Q&A session.

4.3 Gantt Chart (5 Marks)

- **Milestones and Group Involvement**.
- Documentation stating group member participation.

- **ERD and UML Diagrams**
- **SQL Scripts**
- **Gantt Chart**
- **Meeting Minutes and Group Contributions**.