Data and Applications – Project Phase 1

Team Name: db.roll() db.join()

Team Members:

- Naman Ahuja (2019101042)
- Nikhil Chandak (2019111040)
- Aman Atman (2020121006)

A CENTRAL DATABASE MANAGEMENT SYSTEM FOR GYM COMPANIES

Introduction:

The last few months have seen a significant rise in the total weight Earth has been carrying. With COVID enforcing house arrest of people throughout the world, there is a general sense of frustration among the people due to their enforced choice of an unhealthy lifestyle primarily due to lack of physical activity. Peeking into the future, it is not difficult to estimate that gymnasiums are going to face a steep upsurge once they reopen. To enhance user experience and provide better functionality system for the gyms to manage their staff and members, we have come up with a central database management system for a gym company with branches worldwide.

The users of this database system are the members who are enrolled in the gym and the administrative staff of the gym. The members can keep track of all their recent activity using the system, for instance, their gym schedule, their target goals, progress reports, diet plans, and more. The administrative staff can rely on this system to keep a track of their trainers and support staff, closely monitor their performance to ensure efficient gym management.

Data Requirements:

1. Entities:
   - Members
   - Employee – It is divided three *disjoint* subclasses below:
     - Manager
     - Support Staff
     - Trainer
   - Gym Equipment
   - Exercises
   - Body-Part (Weak Entity Type)
   - Diet Plan
   - Dependant (Weak Entity Type)
   - Branch

2. Relationships:
    - Package (A member *may enrol* in a package *to follow* a specific diet plan and certain sets of exercises *under* a specialized trainer)
      Constraints: One member takes a package that can have only one diet plan with any number of packages along with only 1 trainer.

    - Trainer and Members (*trains*)
      Constraints: A member can be trained by only one trainer while a trainer can train at max 20 members.

    - Member, Body-Part and Exercise (Member *works* on a body part *through* an exercise)
      Constraints: One member can work on any number of body parts through any number of exercises.

    - Exercises and Equipment (Each exercise *uses* some equipment)
      Constraints: Each exercise can use any number of equipment while each equipment may be used in any number of exercises.

    - Employee and Branch (Each employee *works* in some branch)
      Constraints: Each employee can work in only one branch, but 1 branch can have any number of employees.

    - Member and Branch (Each member *works out* in some branch)
      Constraints: Each member works out in only 1 branch, but 1 branch can have any number of members.

    - Manager, Trainer and Support Staff (Manager *supervises* Trainer and Support Staff)
      Constraints: Each manager can supervise any number of trainers and support staff while each trainer or support staff person can have only manager.

3. Attribute Details:
    - MEMBER
        - Member ID (Primary Key) (varchar)
        - Name (Composite – f_name, m_init, l_name) (varchar)
        - Address (Composite – street, building, city, state, locality) (varchar)
        - Height (float)
        - Weight (float)
        - BMI (Derived Attribute) (float)
        - Joining-Date (date)
        - Monthly Fee (integer)
        - Efficiency Quotient (Derived) (float) - This signifies the index of change in physique of a member which depends upon loss/gain in weight and change in BMI and time taken for this change.

    - DEPENDENT (weak entity)
        - Name (varchar)
        - Relationship (Partial Key) (varchar)

- EMPLOYEE
    - Employee ID (Primary Key) (varchar)
    - Salary (integer)
    - Age (integer)
    - Address (varchar)
    - Email ID (varchar)
    - Phone Number (integer)

- GYM-EQUIPMENT
    - Name (varchar)
    - Barcode (Primary Key) (long integer)

- DIET-PLAN
    - Supper (Multivalued) (varchar)
    - Breakfast (Multivalued) (varchar)
    - Lunch (Multivalued) (varchar)
    - Dinner (Multivalued) (varchar)
    - Calories consumed per day (integer) - It is the sum of calories consumed from the first 4 attributes.

- EXERCISE
    - Name (Primary Key) (varchar)
    - Description (varchar)
    - Reps (integer)

- BRANCH
    - Address (Composite key - street, building, city, state, country, locality) (varchar)
    - Branch Number (Primary Key) (integer)

- PACKAGE (relationship)
    - Package ID (Primary Key) (varchar)
    - Type (Gold/Silver/Bronze) (varchar)
    - Duration (integer)
    - Weekly workout hours (integer)
    - Cost (integer)

- TRAINS (relationship)
    - Start date (date)
    - End date(date)

- TRAINER (subclass)
    - Shift (varchar)
    - Specialization (varchar)

- SUPPORT STAFF (subclass)
    - Designation (Example: janitor, accountant, technical staff) (varchar)
    - Shift (varchar)

- Body Part (weak entity)
    - Name (Partial Key) (varchar)

Functional Requirements:

1. Retrieval:
   - Selection:
     - a. Get a complete list of members.
     - b. Get a complete list of all equipment in a branch.

     `
   - Projection:
     - c. Show list of members PACKAGE wise sorted by their efficiency quotient.
     - d. List diet plans sorted by their calory count
     - e. Classify members based on their BMI (underweight/overweight/fit).
   - Aggregate:
     - f. Get a complete list of enrolled members.
     - g. Retrieve trainers with maximum/minimum number of clients.
   - Search:
     - h. Partially search name of exercises and equipment.
     - i. A function to partially search for food items in a given diet plan.
   - Analysis:
     - j. *Member Analysis*: This reports the correspondence between the package chosen (if any), the number of working hours put in with the efficiency quotient. Their height, mass, and BMI changes over time will be shown to help them identify their strong and weak periods, thus, helping them to track progress on goals efficiently.

     - k. *Trainer Analysis*: This report analyses the trainers on the metrics of the number of trainees they have, average efficiency quotient of their members, sorted by exercises and grouped by packages opted.
2. Modifications:
   - a. Creation/Insertion:
     - o Add an employee. The employee name (*f_name, m_init, l_name*), address(es), age, salary, email id, phone number(s) should be input. Each employee is assigned a unique id (*Employee ID*).
     - o Add a new member. The member's name (*f_name, m_init, l_name*), address(es), age, height, weight, date of joining (DD/MM/YY) should be input. From these details, the *B.M.I* should be calculated. Each member is assigned a unique id (*Member ID*).
     - o Add a new exercise. The name, description, and reps are stored for each exercise.

- Insert gym equipment details. Equipment's name and barcode are input.
- Create a new branch. The branch's address and name are saved.
- Record the body part a member will train. The member id is also stored to uniquely identify the body part.
- Add a dependent. The name, a relationship type will be stored with employee id.
- Insert a new diet plan. For every plan, we track the supper, breakfast, lunch dinner, and the calories consumed per day.

b. Update:
   - Modify member's package. (For example, upgrading the package from *silver* to *gold*)
   - Constantly update member's height, weight, and calories gained by a member.
   - Change trainer of a member.
   - Update items in a diet plan
   - Change the reps of an exercise.
   - Change package details depending on feedback from its members.
   - Update contact details (address, phone number, email id) of a member.

c. Delete:
   - Delete members whose membership is over or have discontinued gym services.
   - Delete instances of gym equipment which have been damaged or worn out.
   - The resignation/retirement of an employee deletes the details of the employee as well as his/her dependents.
   - Delete a package if it is not relevant anymore.