

# Project Phase I

## OLYMPIA

### Team: Shershaah

*Abhishek Sharma* 2020101050

*Vaibhav Agarwal* 2020101041

*Sarthak Aggarwal* 2020101008

## Introduction

The following is the mini-world representation of gym branches in an Indian state (Maharashtra). The following consists of all the basic functionalities present in the gym database of a renowned gym chain.

It consists of the details for all the members of the gym and staff (trainers and non-trainers) with branch specialization. Also finance and monetary details of the gym will be handled in the DBMS (Database Management System). The gym is present at 4 places namely Pune, Mumbai, Nashik, Nagpur representing a real-world scenario.

## Purpose

The purpose of this database is to represent a mini world which has a focus on the creativity of an individual. This solves the purpose where one would have done all the managing stuff manually. To save this we used DBMS, to make it more organized while having a good control and systematic form of data storage. It helps to maintain an organized system of data comprising all the members, venues, and other important stuff which the end user can directly utilize.

## Users

- Whenever a new member joins, a query will be made to the database wherein we will add the member to the list of that branch.
- All the data of trainers will be stored and will assign them an age group based on their experience.

- The Finance department is another user of the database. To generate monthly/annual reports of the profit/sales/income of the resorts they use the various parameters.

## Applications

In these modern days when all people are becoming concerned about their health and diet, it has become obvious that they continually seek out for gym centers. This management system which we implemented is an effortless way to use gym and health membership system.

It can help to keep records of all the members and their memberships along with their staff, along with all the branches and locations with details of equipment and various trainers present at each branch.

## Database Requirements

### Entities:

#### 1. Branch

- City code\*
- Branch place
- Branch code\*
- Branch head (Composite)

#### 2. Member

- Name (Composite)
- Member Id\*
- Branch place
- Contact Number
- Email id
- D.O.B
- Age group (Derived) (Constraint: {<23,23-35,35-50,>50})
- Body Measurements
- Plan Duration
- Personal Trainer

#### 3. Trainer

- Name (Composite)
- Contact Details\* (Multivalued)--> (Email id, Phone no.)
- Branch Code
- Experience
- Address (Composite)
- Trainer Id\*

- Age Group Constraint: {<23,23-35,35-50,>50}

#### 4. Non-Trainee Staff

- Name (Composite)
- Staff id\*
- Contact Number (Multivalued) --> (Email id, Phone no.)
- Purpose (Multivalued) --> (Hygiene, Electrician, Mechanic etc.)
- Payment
- Working Timing

#### 5. Equipment

- Model
- Branch Code\*
- Tag Id\*
- Current condition (update column)
- Number of parts
- Body part

### Weak Entity:

#### 1. Alumni (member id (foreign key))

- Name (Composite)
- Competition Name
- Competition Level
- Competition Rank
- Year

#### 2. Finance (branch id (foreign key) + month)

- Expenditure
- Month
- Income
- Profit (Derived)
- Income
- Average Income (Derived)

### Derived Attribute

1. Average Income: Total income in N months/(N)
2. Profit: Income: Expenditure
3. Age group: Present Year – Year (D.O.B)

## Composite Attribute

1. Name: Full name including First and last name

## Relationships

### Binary

- a. Branch ---> Trainer Relationship Name->(Training)

**Degree:** 2 **Cardinality ratio:** 1: N **Cardinality Constraints:** (1,1) ---> (1, N)

- b. Non-Trainee Staff ----> Branch Relationship Name->(Payment)

**Degree:** 2 **Cardinality ratio:** M:1 **Cardinality Constraints:** (0, M) ---> (1,1)

- c. Branch ---> Finance Relationship Name--> Earning

**Degree:**2 **Cardinality ratio:** 1:1 **Cardinality Constraints:** (1,1) ---> (1,1)

## N>3 Relationships

A *Member* from a *Branch* will get a *Trainer* according to his/her *age group*.

Branch->Member -> Age group -> Trainer [ 1:M: N: 1]

## SUBCLASS

1. Location ->superclass

Branch, Trainer, Equipment ---->subclass

2. Person -> superclass

Trainer, Member, Non-Trainee Staff ---->subclass

## ENTITIES WITH OTHER KEYS

Entity: Member

1. **Super Key:** {Member id}, {Contact No.}, {Email id},

{Member id, Name}, {Contact No, Name.}, {Email id, Name},

{Member id, Email id}, {Contact No., Email id},  
{Contact No., Member id}, {Contact No, Email id, Member id} etc.

**2.Candidate Key:** {Member id}, {Contact No.}, {Email id}

**3.Alternate Key:** {Contact No.}, {Email id} as {Member id} is a primary key.

## Functional Requirements

### Modifications

**Insert:** Members, Trainers, Equipment, Branches, Age group, Payments, Competition winning by any of the member/trainers

**Delete:** Members who left, trainers who left, broken/old equipment

**Update:** Financial details every month, Equipment conditions, salary for all staff.

### Retrievals

**Selection:** Example:" All the members under the age category (23-35)", "All trainers from all the branches having experience of more than 4 years".

**Projection:** Query to filter out members based on their age-group and branch.

**Aggregate:** "Total Payment given to Non-Trainee Staff using (sum)," Total broken equipment available with same model(count)."

**Search:**" Search for the Competition by text "Mr." which gives results as Mr. India prize, Mr. Maharashtra etc."

### Analysis

- i. Members who could reduce their weight. (Using measurements attribute from Member entity.
- ii. Under which Trainer, members are feeling beneficial, so branch changes the order of trainers.
- iii. People have less BMI or over weighted members so they will be guided for their diet and workouts.