**Gym Management Database**

**Table of Contents**

1. Introduction

2. Conceptual Design

3. Description of Entities

4. Logical Design

5. Physical Design

6. CRUD Operations

7. Scope of the Project

8. Problems Faced and Lessons Learned

9. Future Scope

10.Requirement Table

11.References

**Introduction**

The Gym Management System is a comprehensive system for gyms and recreation centers that manages all of your customers, memberships, and activities across multiple locations. It is made for gyms to keep track of their customers' needs and includes the staff members who work there, making it easier for them to schedule classes without any of them clashing.

**Conceptual Design**

The database design process begins with this stage. The process includes determining the entities, properties, and what links different things together. Also expressing the cardinality that exists between each of the related entities. The database is carefully designed, containing the type of relationship that ties up multiple entities i.e identifying and non-identifying relationship respectively.

The database for the Gym management system is designed in MySQL and consists of 9 entities.

|  |  |
| --- | --- |
| **Entity** | **Attributes** |
| Customer | Customer\_id,First\_Name,Last\_Name,Age,Email,  Phone\_no,Gender |
| membership | Gym\_id,Customer\_id,Expiry\_date,Rate,Date\_Start,  Membership\_id |
| membership\_has\_classes | Membership\_id,Classes\_id |
| gym | Gym\_id,Gym\_name,City,State |
| trainers | Trainers\_id,Trainer\_name,Experience,Age,Gender,Gym\_id,Manager\_id |
| classes | Trainers\_id,Activity\_Type,End\_time,Start\_time,  Classes\_id |
| manager | Manager\_id,Manager\_Name,Manager\_age,Gym\_id |
| equipments\_has\_classes | Equipments\_id,Classes\_id |
| equipments | Equipments\_id,Equipment\_name,Cost,  Total\_quantity |

**Description of Entities**

1. **Customer**- This entity contains customer information like their names, age and their contact details. Each customer has a unique customer\_id which can be used to check details about customers.
2. **Membership –** This entity contains the membership information like the start date and end date of the membership, it also contains a unique membership\_id for each of the customers.
3. **classes –** This entity contains information of the classes hosted in a specific gym such as Start and End time of activities,Trainer\_id. It also contains a unique classes\_id to track each of the classes.
4. **membership\_has\_classes –** This entity is created due to the fact that a many to many relationship was established between the classes and membership entity. This table acts as a join table. It consists of classes\_id and membership\_id.
5. **Gym –** This entity consists of the location of the various gyms such as the Gym\_name,City and the State, it also consists of a unique gym\_id as a representation of each gym.
6. **Trainers –** This entity consists of the details of the trainers such as their name,experience,age,gender, gym\_id,manager\_id along with the unique trainer\_id**.**
7. **Manager –** This entity consists of the details of the manager working at a particular gym like their name age and the gym\_id. It also consists of a unique manager\_id.
8. **equipments\_has\_classes –** Due to the many to many relationship present between the equipments and classes, this entity was formed which acts as a join entity.This consists of the equipment\_id and classes\_id.
9. **Equipments –** This entity consists of the list of equipments used by the various customers for the different types of activities. It consists of the equipment name,the cost of the equipment and total quantity for each of the equipment. It also consists of a unique equipments\_id.

**Logical Design**

The database design process is now in its second stage. Translation of entities and attributes was required and conversion of linkages from conceptual design. Additionally, it involved including data types and

describing the connection between the tables i.e primary and foreign keys.

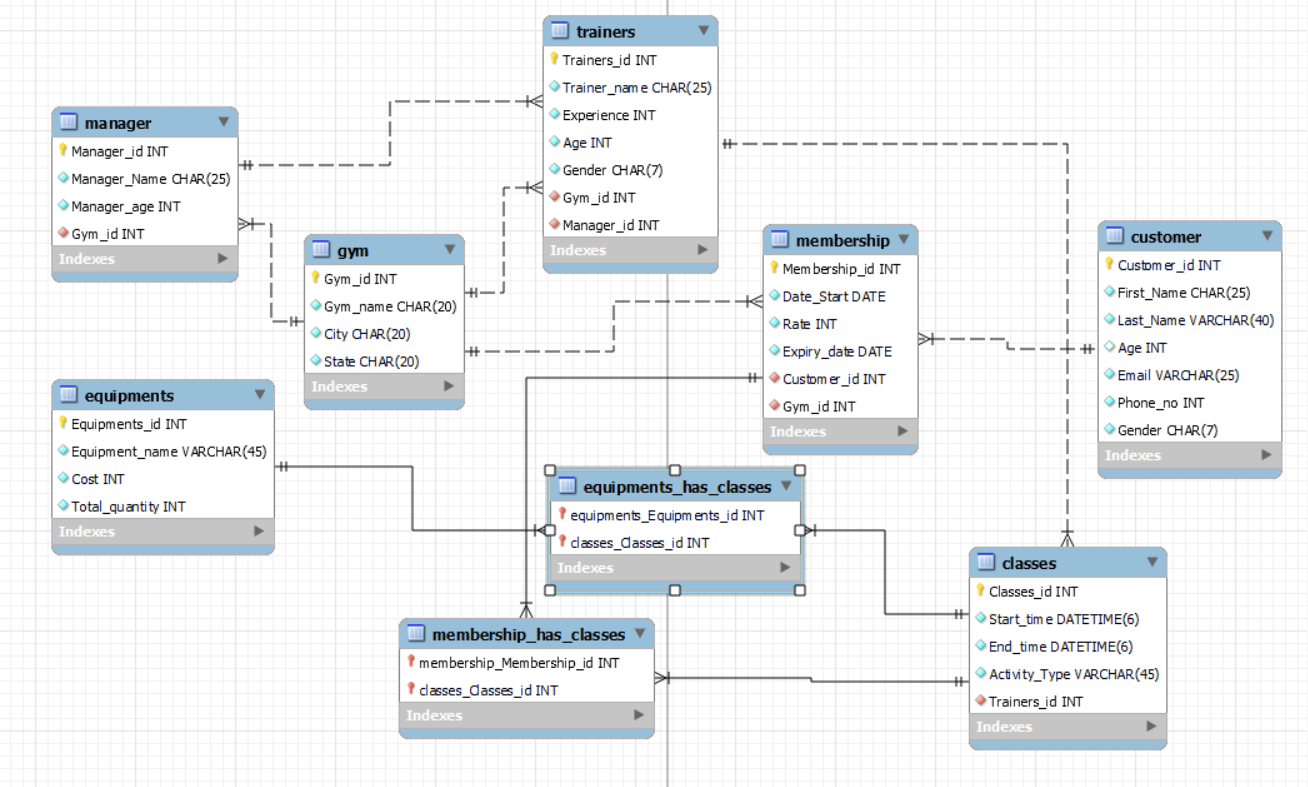
This phase consists of two steps-

1. Normalization

2. Drawing an ER diagram

Normalization is the process of eliminating partial dependency, transitive dependency and other potential problems involved in relations. We removed partial, transitive dependency from our tables. After establishing our relationship in 3 NF, we began building tables in MySQL.

The ERD in its 3NF form is as below-



Without any more adjustments, the aforementioned figure appears to be in its 3NF form. The two entities that make up this table, membership has classes and equipment has classes, serve as a replacement for the many-to-many relationship that existed between the linked tables.

The reason behind using a one to many relationship instead of many to many is as below-

Since this only saves one relationship, while we require several, we are unable to add the primary key of one table into the other or both.

In this case, maintaining and querying would be quite challenging because we would only have one column for storing various variables.

Additionally, we couldn't have a lot of columns with classes ID values because this would make things complicated and restrict the amount of relationships.

**Physical Design**

This is the third step of the database design. This requires setting up the database on the MySQL server. There were no modifications made to the ERD because we normalized the relations before adopting. We carried out the CRUD procedures after inserting the sample data into the tables.

We have made sure that each of the tables have at least 30 sample values.

**CRUD Operations**

**Query 1-**

Equipments handled by each Trainer-

CREATE VIEW v\_equip\_trainer as

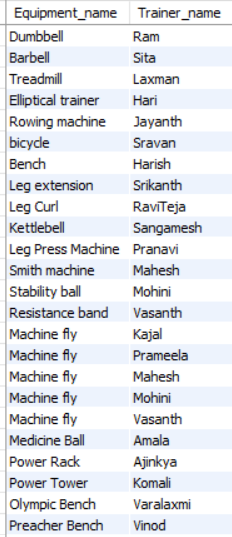
SELECT equipments.Equipment\_name,trainers.Trainer\_name

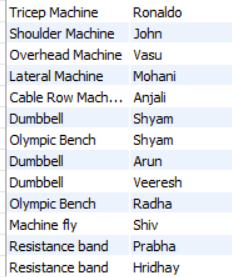
FROM equipments

JOIN equipments\_has\_classes ON (equipments.Equipments\_id=equipments\_has\_classes.Equipments\_id)

JOIN classes ON(classes.Classes\_id=equipments\_has\_classes.Classes\_id)

JOIN trainers ON(trainers.Trainers\_id=classes.Trainers\_id);





**Query 2-**

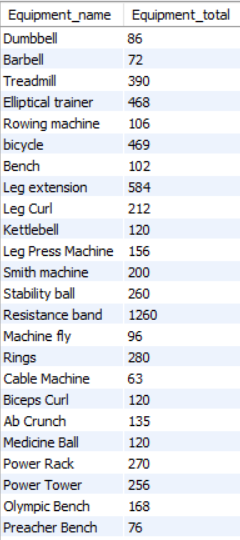
Total price of each equipments-

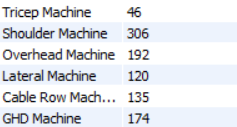
CREATE VIEW v\_total\_equipments as

SELECT Equipment\_name,

SUM(Total\_quantity\*cost) AS Equipment\_total

FROM equipments GROUP BY Equipment\_name;





**Query 3-**

Return the 2 customers that paid least membership-

CREATE VIEW v\_least\_cost as

SELECT concat(y.First\_Name,' ',y.Last\_name) as cust\_name, x.Date\_Start, x.Rate

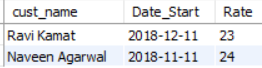
from membership x JOIN customer y ON (x.Customer\_id=y.Customer\_id)

GROUP BY 1=(SELECT

COUNT(distinct Rate)

from membership it

WHERE x.Rate >= it.Rate);



**Query 4-**

Return the COUNT of female candidates trained by each Trainer-

CREATE VIEW v\_COUNT\_candidates as

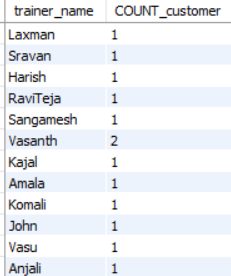
SELECT trainers.trainer\_name,COUNT(customer.customer\_id) as COUNT\_customer from ((trainers

JOIN membership ON((trainers.Gym\_id=membership.Gym\_id)))

JOIN customer ON((customer.Customer\_id=membership.Customer\_id)))

WHERE (customer.Gender="Female")

GROUP BY trainers.trainer\_name;



**Query 5-**

Finding the Employees who work at Fitness Center(Manager and trainers)-

CREATE VIEW v\_Employees as

SELECT Manager\_name,gym\_name,trainer\_name from manager

JOIN gym ON(manager.Gym\_id=gym.Gym\_id)

AND gym\_name like "Fitness Center"

JOIN trainers ON(trainers.gym\_id=gym.Gym\_id);



**Query 6-**

Finding the trainer who uses Dumbell for training purpose-

SELECT trainer\_name from trainers

WHERE trainers\_id in

(SELECT trainers\_id from classes

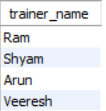
WHERE Classes\_id in

(SELECT Classes\_id from equipments\_has\_classes

WHERE equipments\_id in

(SELECT Equipments\_id from equipments

WHERE Equipment\_name like "Dumbbell")));



**Query 7-**

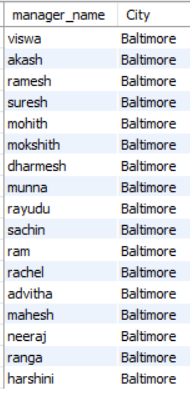
List of managers names working in the city of Baltimore

CREATE VIEW v\_manager\_city as

SELECT manager.manager\_name,gym.City from manager

JOIN gym ON(manager.gym\_id=gym.Gym\_id)

WHERE gym.city="Baltimore";



**Query 8-**

Find the list of Customers who’s membership expires by end of 2022-

CREATE VIEW v\_membership\_expiry as

SELECT concat(First\_Name,' ',Last\_name) as cust\_name from customer

WHERE Customer\_id in

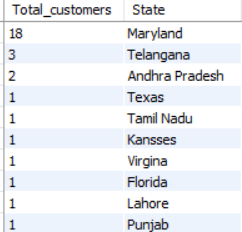
(SELECT Customer\_id from membership

WHERE Expiry\_date<"2022-12-30");



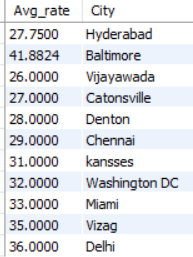
Query 9-

Count the number of customers state-wise



Query 10-

Average rate of Membership City-wise-



**Scope of the Project-**

The project is currently dealing with the storing the equipment's, customer details, membership details, schedule of the classes, the trainers and managers working at multiple gyms.

**Problems Faced-**

1. Searching for Dummy Data: We made an effort to hunt for a method of populating the data utilizing some external internet data sources. We were unable to get the data online, either because it was unavailable or because it wasn't in the format we needed.

2. Unable to work in collaboration: We were forced to wait for each other because MySQL Workbench is not a collaborative tool, preventing us from working in parallel. Since only one person could work at a time, this took up the majority of our time.

3. Leadership Challenges: Coordinating the efforts of multiple team members was challenging, as it is time consuming to coordinate schedules, arrange meetings, meet, correspond, make decisions collectively, integrate the contributions of group members, etc.

**Lessons Learned** -

1. The tools - MySQL workbench and MySQL server should be installed after validating their versions within the team.

2. The team should pick a collaborative tool other than MySQL workbench for parallel

work.

3. Leadership role may be too stressful, it would have been more ideal for each one of the peers to take up leadership role at different phases of the project.

**Future Scope-**

1. Adding the ratings of each Gym will enable customers to select the gym they want to be a member of.

2.Storing the different types of facilities present in a particular gym for customer visibility.

3.Having a sample video section of how the trainers train their customer, which will enable easier selection of trainer.

**Requirement Table-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Query Number | Req. A | Req. B | Req. C | Req. D | Req. E |
| Query 1 | ✓ |  |  | ✓ |  |
| Query 2 | ✓ |  | ✓ |  |  |
| Query 3 | ✓ | ✓ | ✓ | ✓ |  |
| Query 4 | ✓ | ✓ | ✓ | ✓ |  |
| Query 5 | ✓ | ✓ |  | ✓ |  |
| Query 6 |  | ✓ |  |  | ✓ |
| Query 7 | ✓ | ✓ |  | ✓ |  |
| Query 8 | ✓ | ✓ |  |  | ✓ |
| Query 9 | ✓ | ✓ | ✓ |  |  |
| Query 10 | ✓ |  | ✓ |  |  |

References -

<https://www.inettutor.com/source-code/gym-management-system-database-design/>