Group Project

Final Report

## Group 1

Wanqin Liang 991694155

Xiaoyi Zhang 991711956

SDNE program, Sheridan College, Oakville, ON DBAS 27198: Database Design and Implementation

**Dr. Peter Riis**

December 07, 2023

# Table of content

|  |  |
| --- | --- |
| **Section I: EER Diagram** | **2** |
| **Section II: Data Structure Diagram** | **4** |
| **Section III: SQL Scripts for Table Creation** | **5** |
| **Section IV: Insert Date for Tables** | **8** |
| **Section V: Wireframe Information** | **11** |
| **Section VI: Convert Entities to MongoDB Collections** | **16** |
|  |  |
|  |  |
|  |  |
|  |  |

**Section I: EER Diagram**

**Assumptions for the Business Model:**

**User(Client):** All users can be uniquely identified by a User ID. Although every user has a single home location, users can register for as many classes and coaches as they want at any location. A user can review many coaches or choose not to leave review for any coach at all. Also, a user can leave reviews for the same coach multiple times. Users provide references to coaches. Due to the reserved keyword in SQL, the entity name "Client" will be utilized instead of "User" in the SQL queries later in this report.

**Coach:** A coach has a unique ID, name, self-introduction, and a level that represents their coaching seniority level. They may have multiple certifications as well as references from users. A coach can serve at many locations and assume multiple classes there. A coach may receive reviews from many users or not have any reviews at all.

**Location:** There are many branch locations scattered around at various addresses. Each location has specific information of location ID, location name, Tel, email address, physical address, and location icon. A location can have many coaches, classes, and amenities.

**Class:** There are various types of fitness courses offered by the gym. Every fitness course encompasses specific details such as class name, level, and price. A fitness course can be taught by one or more coaches and the same series of classes can happen at multiple locations.

**Amenity:** All locations are equipped with multiple amenities. Every amenity has specific information of amenity type and image icon. However, not all locations are fully equipped with the same amenities, for instance some locations may not have a swimming pool.

**Image 1: Enhanced Entity Relationships Diagram**

A diagram of a company

Description automatically generated

**Section II: Data Structure Diagram 3rd Normal Form**

**Image 2: Data Structure Diagram**

A diagram of a computer

Description automatically generated with medium confidence

**3rd Normal Form Analysis:**

We convert the conceptual model to a relational logical model and make a data structure diagram for the tables making sure that all the tables are in the 3rd normal form.

**Gym Location:**

**Primary Key:** LocationID

**Attributes:** OfficeTel, OfficeEmail, StreetAddress, UnitNo, CityName, StateName, Country, ZipCode

**3rd Normal Analysis:** This table is in 3NF as all attributes directly depend on LocationID without multi values and transitive dependency.

**User:**

**Primary Key:** UserID

**Attributes:** UserName, Gender, UserEmail, UserPhoneNo., AccessLevel, Home Location ID.

**3rd Normal Analysis:** This table is in 3NF as all attributes directly depend on the primary key of UserID without multi values and transitive dependency.

**Coach:**

**Primary Key:** CoachID

**Attributes:** CoachName, CoachLevel, SelfIntroduction

**3rd Normal Analysis:** The Coach table is in 3NF as all information is about the coach and depends on the CoachID without multi values and transitive dependency.

**Certificate:**

**Primary Key:** CertificateID

**Attributes:** CoachID (FK), CertificateName

**3rd Normal Analysis:** The Certificate table is the 3rd normal form. Both Coach ID and Certificate Name depend on and the CertificateID without multi values and transitive dependency, and CoachID is a foreign key to the Coach table.

**References:**

**Primary Key:** References ID

**Attributes:** User ID(FK), CoachID(FK)

**3rd Normal Analysis:** The References table is the 3rd normal form. As all attributes depend on Reference ID without multi values and transitive dependency.

**Class:**

**Primary Key:** ClassID

**Attributes:** ClassName, ClassLevel, ClassPrice

**3rd Normal Analysis:** The Classes table is in 3NF as all attributes are dependent on ClassID without multi values and transitive dependency.

**UserCoachReview Relation:**

**Primary Key:** ReviewID

**Attributes:** UserID (FK), CoachID (FK), Communication, Enthusiasm, Punctuality, Comments

**3rd Normal Analysis:** This table is in 3NF. All non-key attributes are dependent on the primary key without multi values and transitive dependency.

**Amenity:**

**Primary Key:** AmenityID

**Attributes:** AmenityType, UserGuideline

**3rd Normal Analysis:** The Amenity table is in 3NF, as all attributes depends on the Amenity ID without multi values and transitive dependency.

**AmenityLocation Relation:**

**Composite Key:** LocationID, AmenityID

**Attributes:** Quantity

**3rd Normal Analysis:** The AmenityLocation is a junction table with a composite primary key. All other attributes are dependent on the composite key but not a partial of the composite key. Also the relation is without multi values and transitive dependency

**Visits Relation:**

**Composite Key:** UserID, LocationID

**Attributes:** VisitTimes

**3rd Normal Analysis:** VisitTimes depends on both UserID and LocationID but not a partial of the composite key without multi values and transitive dependency, so this table is in 3NF.

**CoachLocationClassAssignment Relation:**

**Primary Key:** AssignmentID

**Attributes:** CoachID (FK), LocationID (FK), ClassID (FK), RoomNo, ClassTime

**3rd Normal Analysis:** This table is in the 3rd normal form. As the primary key directly identifies each attribute without multi values and transitive dependency.

# Section III: SQL Scripts for Table Creation

-- Creating Amenity table

CREATE TABLE Amenity (

AmenityID INT IDENTITY(11111,1) PRIMARY KEY,

AmenityType VARCHAR(255) NOT NULL,

ImageIcon VARCHAR(255)

);

-- Creating GymLocation table

CREATE TABLE GymLocation (

LocationID INT IDENTITY(1111,1) PRIMARY KEY,

OfficeTel VARCHAR(50),

OfficeEmail VARCHAR(255) NOT NULL,

StreetAddress VARCHAR(255),

UnitNo VARCHAR(50),

CityName VARCHAR(100),

StateName VARCHAR(100),

Country VARCHAR(100) DEFAULT 'Canada',

ZipCode VARCHAR(20),

LocationName VARCHAR(255),

LocationIcon VARCHAR(255)

);

-- Creating AmenityLocation table

CREATE TABLE AmenityLocation (

LocationID INT,

AmenityID INT,

Quantity INT NOT NULL,

PRIMARY KEY (LocationID, AmenityID),

FOREIGN KEY (AmenityID) REFERENCES Amenity (AmenityID),

FOREIGN KEY (LocationID) REFERENCES GymLocation (LocationID)

);

-- Creating Client table

CREATE TABLE Client (

ClientID INT IDENTITY(11111111,1) PRIMARY KEY,

ClientName VARCHAR(255) NOT NULL,

ClientEmail VARCHAR(255) NOT NULL,

ClientPhoneNo VARCHAR(50),

HomeLocationID INT,

FOREIGN KEY (HomeLocationID) REFERENCES GymLocation(LocationID)

);

-- Creating Visit table

CREATE TABLE Visit (

ClientID INT,

LocationID INT,

VisitTimes INT NOT NULL,

PRIMARY KEY (ClientID, LocationID),

FOREIGN KEY (ClientID) REFERENCES Client(ClientID),

FOREIGN KEY (LocationID) REFERENCES GymLocation(LocationID)

);

-- Creating Coach table

CREATE TABLE Coach (

CoachID INT IDENTITY(111,1) PRIMARY KEY,

CoachName VARCHAR(255) NOT NULL,

CoachLevel INT,

TrainingPhilosophy TEXT,

CoachImage VARCHAR(255),

Rating DECIMAL(3, 2)

);

-- Creating ClientCoachReview table

CREATE TABLE ClientCoachReview (

ReviewID INT IDENTITY(1,1) PRIMARY KEY,

ClientID INT,

CoachID INT,

Communication INT NOT NULL,

Enthusiasm INT NOT NULL,

Punctuality INT NOT NULL,

Comments TEXT,

FOREIGN KEY (ClientID) REFERENCES Client(ClientID),

FOREIGN KEY (CoachID) REFERENCES Coach(CoachID)

);

-- Creating Certificate table

CREATE TABLE Certificate (

CertificateID INT IDENTITY(1,1) PRIMARY KEY,

CoachID INT,

CertificateName VARCHAR(255) NOT NULL,

FOREIGN KEY (CoachID) REFERENCES Coach(CoachID)

);

-- Creating CoachLocation table

CREATE TABLE CoachLocation (

CoachID INT,

LocationID INT,

PRIMARY KEY (CoachID, LocationID),

FOREIGN KEY (CoachID) REFERENCES Coach(CoachID),

FOREIGN KEY (LocationID) REFERENCES GymLocation(LocationID)

);

-- Creating References table

CREATE TABLE Reference (

ReferenceID INT IDENTITY(1,1) PRIMARY KEY,

CoachID INT,

ClientID INT,

FOREIGN KEY (CoachID) REFERENCES Coach(CoachID),

FOREIGN KEY (ClientID) REFERENCES Client(ClientID)

);

-- Creating Class table

CREATE TABLE Class (

ClassID INT IDENTITY(111111,1) PRIMARY KEY,

ClassName VARCHAR(255) NOT NULL,

ClassPrice DECIMAL(10, 2) CHECK (ClassPrice >= 0)

);

-- Creating CoachLocationClassAssignment table

CREATE TABLE CoachLocationClassAssignment (

AssignmentID INT IDENTITY(1,1) PRIMARY KEY,

CoachID INT,

LocationID INT,

ClassID INT,

RoomNo VARCHAR(50),

ClassTime TIME,

ClassDay DATE,

FOREIGN KEY (CoachID) REFERENCES Coach(CoachID),

FOREIGN KEY (LocationID) REFERENCES GymLocation(LocationID),

FOREIGN KEY (ClassID) REFERENCES Class(ClassID)

);

-- Creating ClassEnrollment table

CREATE TABLE ClassEnrollment (

ClassID INT,

ClientID INT,

EnrollmentDate DATE,

PRIMARY KEY (ClassID, ClientID),

FOREIGN KEY (ClassID) REFERENCES Class(ClassID),

FOREIGN KEY (ClientID) REFERENCES Client(ClientID)

);

# Section IV: Insert Data for Tables

-- Insert data into Amenity

INSERT INTO Amenity (AmenityType, ImageIcon) VALUES

('Sauna', 'sauna\_icon.png'),

('Green Zone', 'green\_zone.png'),

('Hot Yoga', 'hot\_yoga.png'),

('Spinning', 'soinning\_icon.png'),

('Pool', 'pool\_icon.png');

-- Insert data into GymLocation

INSERT INTO GymLocation (OfficeTel, OfficeEmail, StreetAddress, UnitNo, CityName, StateName, Country, ZipCode, LocationName, LocationIcon) VALUES

('555-1000', 'midtown@gym.com', '123 Midtown Ave', 'Suite 100', 'Metropolis', 'Metro', 'Canada', 'M1T 2A3', 'Midtown', 'midtown\_icon.png'),

('555-2000', 'cabbagetown@gym.com', '456 Cabbagetown St', 'Suite 200', 'Metropolis', 'Metro', 'Canada', 'D2T 3B4', 'Cabbagetown', 'cabbagetown\_icon.png'),

('555-3000', 'eastview@gym.com', '789 Eastview Blvd', 'Suite 300', 'Metropolis', 'Metro', 'Canada', 'U3T 4C5', 'EasetView', 'eastview\_icon.png'),

('555-4000', 'westside@gym.com', '101 Westside Rd', 'Suite 400', 'Metropolis', 'Metro', 'Canada', 'E4T 5D6', 'WestSide', 'westside\_icon.png');

-- Insert data into AmenityLocation

INSERT INTO AmenityLocation (LocationID, AmenityID, Quantity) VALUES

(1111, 11111, 2),

(1111, 11112, 1),

(1111, 11113, 3),

(1111, 11114, 5),

(1111, 11115, 1),

(1112, 11113, 1),

(1113, 11114, 2);

-- Insert data into Client

INSERT INTO Client (ClientName, ClientEmail, ClientPhoneNo, HomeLocationID) VALUES

('John Doe', 'john.doe@example.com', '123-456-7890', 1111),

('Jane Smith', 'jane.smith@example.com', '234-567-8901', 1112),

('Robert Jones', 'robert.jones@example.com', '345-678-9012', 1113),

('Emily Johnson', 'emily.johnson@example.com', '456-789-0123', 1114);

INSERT INTO Visit (ClientID, LocationID, VisitTimes) VALUES

(11111112, 1111, 15),

(11111112, 1112, 8),

(11111112, 1113, 3),

(11111113, 1111, 10),

(11111113, 1112, 5),

(11111113, 1113, 3);

-- Insert data into Coach

INSERT INTO Coach (CoachName, CoachLevel, TrainingPhilosophy, CoachImage, Rating) VALUES

('Sally', 3, 'Training Philosophy for Sally', 'sally\_image.png', 5),

('Adem', 2, 'Training Philosophy for Adem', 'adem\_image.png', 4),

('Sander', 3, 'Training Philosophy for Sander', 'sander\_image.png', 5),

('Sayed', 5, 'Training Philosophy for Sayed', 'sayed\_image.png', 4.6),

('Layla', 5, 'Training Philosophy for Layla', 'layla\_image.png', 4.6),

('Mark', 5, 'Training Philosophy for Mark', 'mark\_image.png', 4.6);

-- Insert data into ClientCoachReview

INSERT INTO ClientCoachReview (ClientID, CoachID, Communication, Enthusiasm, Punctuality, Comments) VALUES

(11111112, 111, 5, 5, 5, 'Great coach, very motivating.'),

(11111112, 112, 5, 5, 5, 'Great coach, very motivating.'),

(11111113, 112, 4, 4, 4, 'Good sessions but needs to be more punctual.'),

(11111114, 113, 3, 3, 3, 'Satisfactory but could communicate better.'),

(11111115, 114, 5, 5, 5, 'Excellent, keeps me on track with my goals.');

-- Insert data into Certificate

INSERT INTO Certificate (CoachID, CertificateName) VALUES

(111, 'Advanced Fit'),

(111, 'BikramYoga'),

(111, 'Massage III'),

(112, 'BikramYoga'),

(113, 'Massage III'),

(114, 'Advanced Fit');

-- Insert data into CoachLocation

INSERT INTO CoachLocation (CoachID, LocationID) VALUES

(111, 1111),

(112, 1112),

(113, 1113),

(114, 1114),

(114, 1111),

(115, 1111),

(116, 1111);

-- Insert data into Reference

INSERT INTO Reference (CoachID, ClientID) VALUES

(111, 11111112),

(112, 11111113),

(113, 11111114),

(114, 11111115);

-- Insert data into Class

INSERT INTO Class (ClassName, ClassPrice) VALUES

('Body Pump', 10.00),

('Hiit', 5.00),

('Yoga', 8.00);

-- Insert data into CoachLocationClassAssignment

INSERT INTO CoachLocationClassAssignment (CoachID, LocationID, ClassID, RoomNo, ClassTime, ClassDay) VALUES

(111, 1113, 111111, 'Big Studio', '09:50:00', '2023-09-01'),

(112, 1111, 111112, 'Studio 1', '09:50:00', '2023-09-02'),

(113, 1111, 111111, 'Studio 1', '10:20:00', '2023-09-03'),

(114, 1113, 111113, 'Small Studio', '09:50:00', '2023-09-04');

-- Insert data into ClassEnrollment

INSERT INTO ClassEnrollment (ClassID, ClientID, EnrollmentDate) VALUES

(111111, 11111112, '2023-09-01'),

(111112, 11111113, '2023-09-08'),

(111113, 11111114, '2023-09-15'),

(111111, 11111115, '2023-09-22');

**Section V: Wireframe Information**

# Achieve the Wireframe 1

1. Return all the information displayed on the Locations tab (wireframe 1)

-- Home Location Times Visited This Month

SELECT Client.ClientID, ClientName, Client.HomeLocationID, Visit.VisitTimes

FROM Client

INNER JOIN Visit

on Visit.LocationID=Client.HomeLocationID and Visit.ClientID=Client.ClientID

where Client.ClientID=11111112

A screenshot of a computer

Description automatically generated

-- Other Location Times Visited This Month

SELECT Client.ClientID, ClientName, Client.HomeLocationID, Visit.VisitTimes

FROM Client

INNER JOIN Visit

on Visit.LocationID!=Client.HomeLocationID and Visit.ClientID=Client.ClientID

where Client.ClientID=11111112

A screenshot of a computer

Description automatically generated

# Achieve the wireframe 2

1. Return all the information displayed on the Midtown Location tab (wireframe 2)

SELECT GymLocation.\*, Amenity.AmenityType, AmenityLocation.AmenityID,

CoachLocation.CoachID, Coach.CoachName

FROM GymLocation

INNER JOIN AmenityLocation

on GymLocation.LocationID=AmenityLocation.LocationID

INNER JOIN Amenity

on Amenity.AmenityID=AmenityLocation.AmenityID

INNER JOIN CoachLocation

on GymLocation.LocationID=CoachLocation.LocationID

INNER JOIN Coach

on Coach.CoachID = CoachLocation.CoachID

where GymLocation.LocationName = 'Midtown'

A screenshot of a computer

Description automatically generated

# Achieve the wireframe 3

1. Return all the information displayed on the Coaches tab (wireframe 3)

-- My Coaches' Rating

SELECT Coach.CoachName, Coach.CoachLevel, Coach.Rating, Coach.CoachImage

FROM Coach

INNER JOIN ClientCoachReview

on Coach.CoachID=ClientCoachReview.CoachID

Where ClientCoachReview.ClientID=11111112

A screenshot of a computer

Description automatically generated

--Other Coaches Rating

SELECT Coach.CoachName, Coach.CoachLevel, Coach.Rating, Coach.CoachImage

FROM Coach

WHERE Coach.CoachID NOT IN (SELECT Coach.CoachID

FROM Coach

INNER JOIN ClientCoachReview

on Coach.CoachID=ClientCoachReview.CoachID

Where ClientCoachReview.ClientID=11111112)

A screenshot of a computer

Description automatically generated

# Achieve the wireframe 4

1. Return all the information displayed on the Coach Sally tab (wireframe 4)

SELECT Coach.\*, Certificate.CertificateName, GymLocation.LocationName, GymLocation.LocationIcon,

Client.ClientName, Client.ClientPhoneNo, Client.ClientEmail

FROM Coach

Inner JOIN Certificate

ON Coach.CoachID=Certificate.CoachID

INNER JOIN CoachLocation

ON Coach.CoachID=CoachLocation.CoachID

INNER JOIN GymLocation

ON GymLocation.LocationID=CoachLocation.LocationID

INNER JOIN REFERENCE

ON REFERENCE.CoachID=Coach.CoachID

INNER JOIN Client

ON REFERENCE.ClientID=Client.ClientID

WHERE Coach.CoachID=111

A screenshot of a computer

Description automatically generated

# Section VI: Convert Entities to MongoDB Collections

## 

## Collections Schema:

A screenshot of a computer

Description automatically generated

**Rationale for Client Collection:**

Choose referenced documents of Coach but embedding documents for Class in the Client collection.

Reason for embedding Class documents: if all clients disappear or cancel memberships, there should be no active class left in the system. Such relationship is atomic, hence the embedding documents of Class is designed for the Client collection.

Reason for referenced Coach documents: if there are no clients or users in the gym, the coaches can still be registered valid on the gym profile. As a result, coach class should be normalized from the Client collection using referenced documents.

**Rationale for Coach Collection:**

Choose embedding documents for User Review, Client Reference, and Certificate in the Coach collection. The reason is reviews, references, and certificates are specific to a coach. If a coach leaves the gym, all the records of its reviews, references, and certificates should be removed from the gym system. Hence, they constitute an atomic relationship, embedding documents of reviews, references, and certificates is the best choice.

## MongoDB Script for Client:

A screenshot of a computer code

Description automatically generated

**MongoDB Script for Coach:**

A screenshot of a computer program

Description automatically generated