I. Introduction:

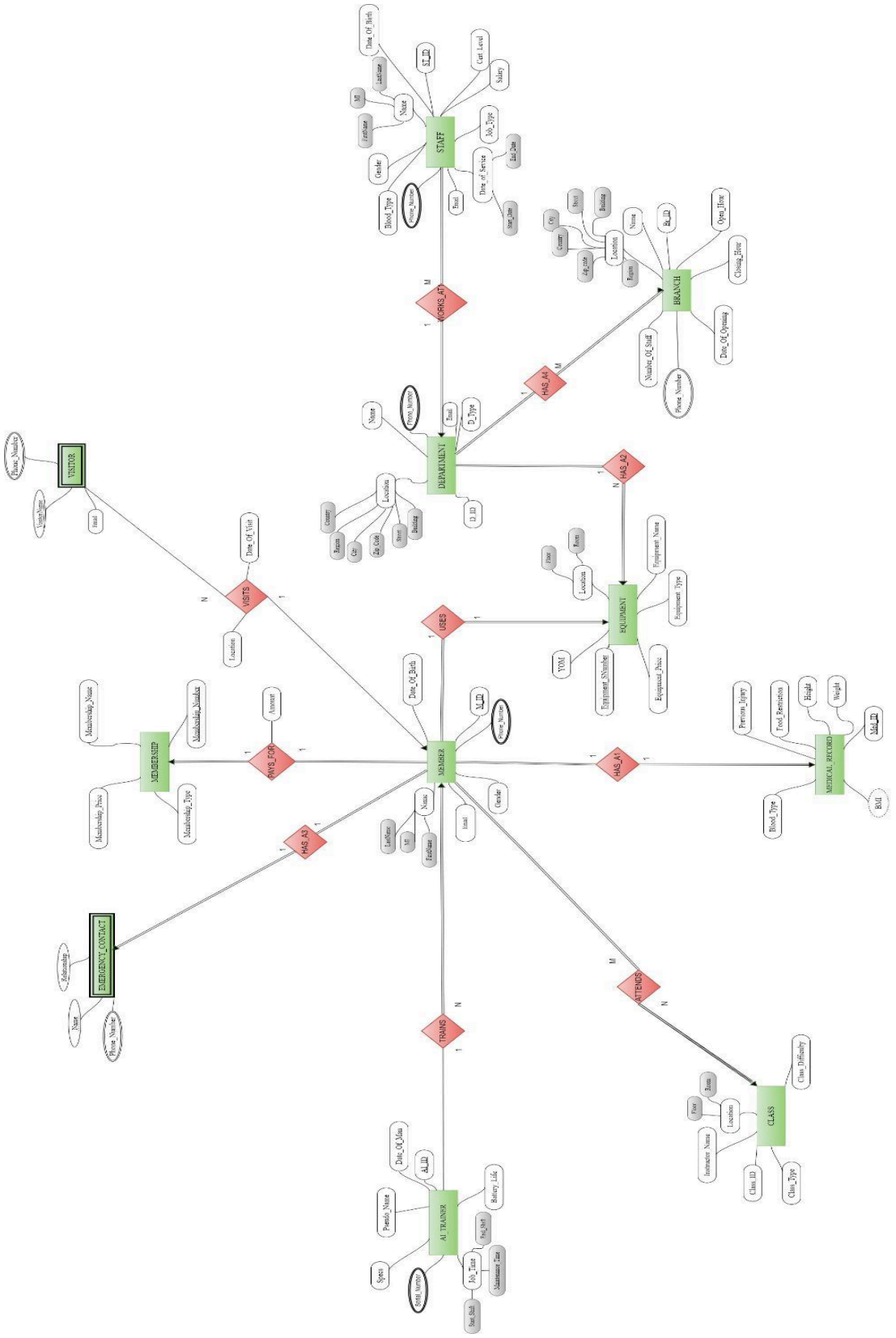
Founded in 2010, FitneX (now GymnAIsium) was an ordinary fitness club initially located in downtown Beirut. However, with the development of technology six years later, the founder of FitneX, Mrs. Elle Darzi decided to shift the gym's perspective to AI. GymnAIsium is an AI-based fitness center equipped with modern technologies to help better assist members within the club. Technologies include AI trainers, latest TV models, Virtual-Reality equipment, and so forth. GymnAIsium's AI trainers are programmed and embedded with motion sensors to allow peer-to-peer assistantship and better guidance for members. It also includes VR machines that are also programmed for the cause. The gym’s shift required a huge alter in the database system. As a result, our group "The Databasers" was contacted to create an entirely new database and incorporate the new changes made to the fitness center.

Our Purpose:

GymnAIsium is not your average fitness center. Our main goal is to provide the most comfortable and most efficient way to keep you healthy. This is where our specialty comes in, after the pandemic a lot of people weren’t comfortable to train with a normal trainer, so we provided AI trainers that are fully equipped to assist your fitness journey. And for the people who still want human trainers that is still available. Our facilities are well spread with different branches for wherever you go. We also have different classes that are for specific sport you desire, one of the famous classes is Virtual reality simulator, this class takes the member to places unreachable in other ways. If your have friends who wants to try our gym we have just the offer for you. All that and more in GymnAIsium.

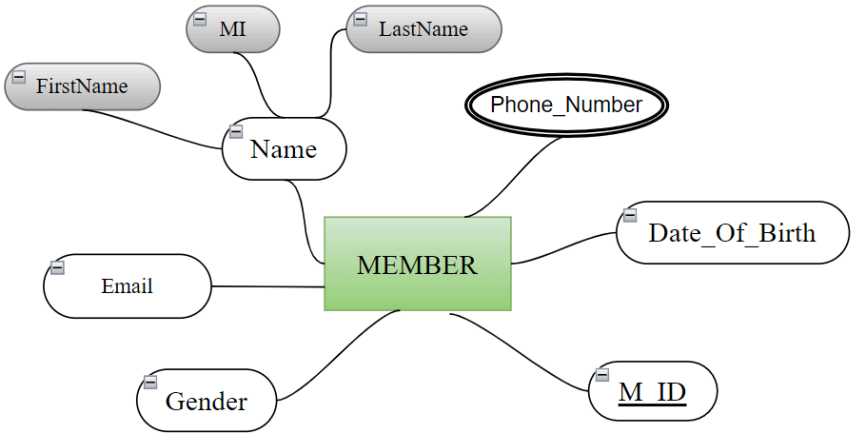
This database was created to allow members to choose from multiple MEMBERSHIPS offered by the fitness club. The member is offered with a variety of CLASSES and is allowed to register in multiple classes at once. These CLASSES are equipped with modern AI and NON-AI EQUIPMENTS to keep up with modern technologies. The member is also allowed to have multiple VISITORS in the club.Also, An EMERGENCY_CONTACT of the member is saved in case of emergencies.

II. ER Diagram:



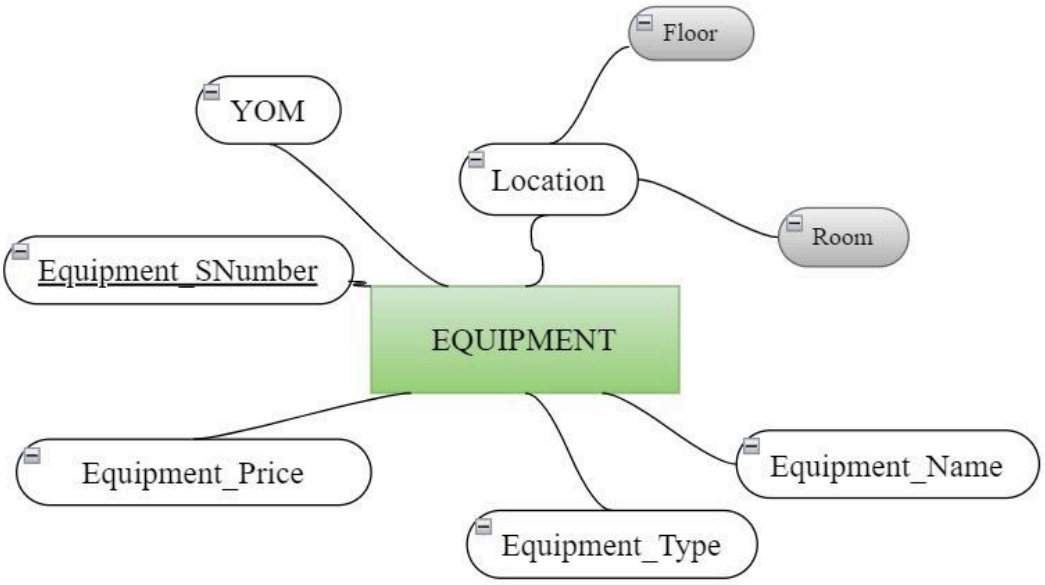
II. Entity Types:

1. MEMBER:



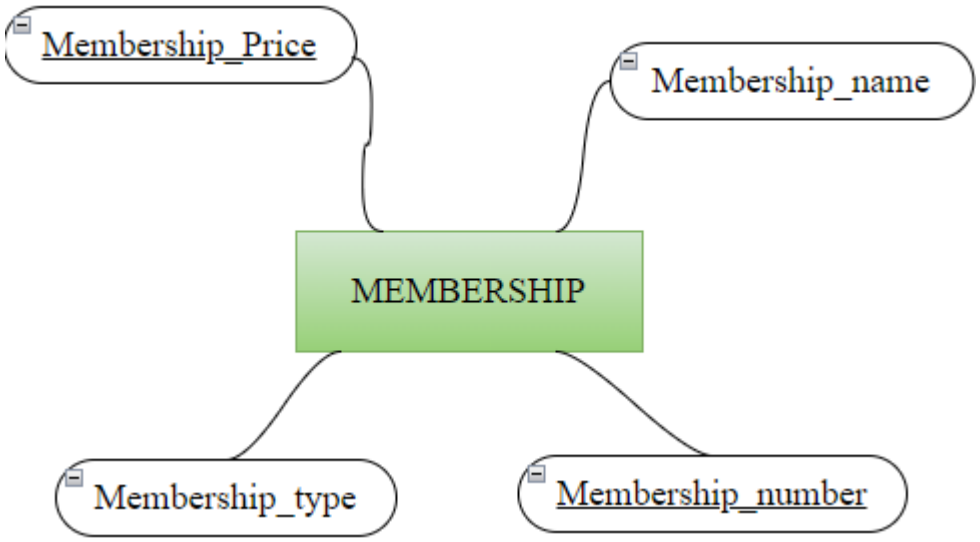
The MEMBER is the only entity type that trains at our gym. At GymnAIsium, the MEMBER is our top priority from serving with professionalism to helping them achieve their personal goals. The entity type includes a **Name (composite attribute)**, an **Email (attribute)**, **a Phone_Number (multi-valued attribute)**, a **Gender (attribute)** and a **Date_Of_Birth (attribute)**. In addition to all of that, every MEMBER has a unique M_ID ,which is a key attribute that makes each MEMBER unique.

2. EQUIPMENT:



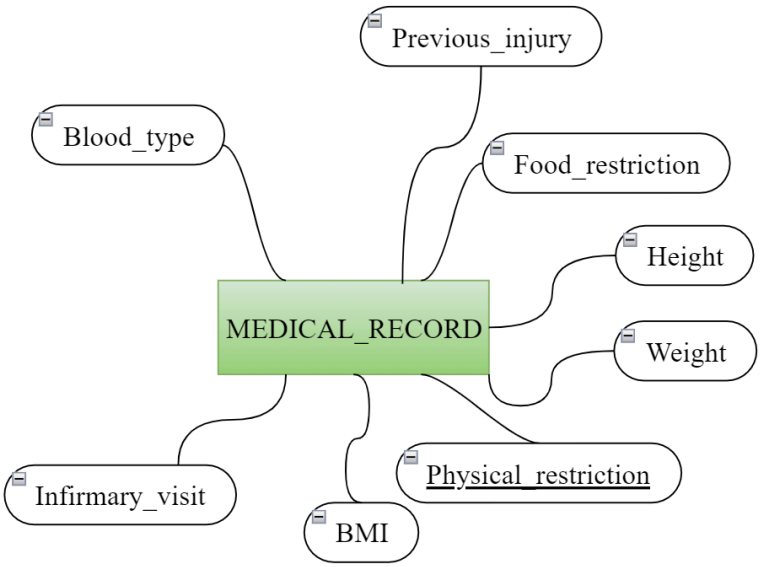
The EQUIPMENT entity type is used to keep track of all the EQUIPMENTS used in GymnAIsium. The EQUIPMENT entity type has different non unique attributes such as **Equipment_Name (attribute)**, **Equipment_Type (attribute)**, **Equipment_Price (attribute)**, **YOM (attribute)** and **Location (composite attribute)** divided into **Floor (attribute)** and **Room (attribute)**. Equipment_SNumber (key attribute) is unique to each EQUIPMENT in the GymAIsium.

3. MEMBERSHIP:



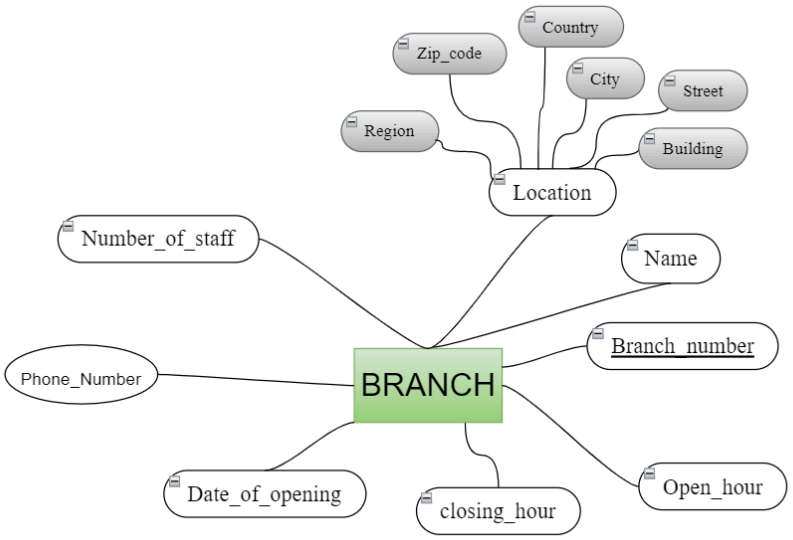
The MEMBERSHIP entity type is important to define the period of subscription of each person as a member. MEMBERSHIP contains non-unique attributes: **Membership_name (attribute)** and M**embership_type (attribute)**. However there are two key attribute for MEMBERSHIP such as Membership_number (key attribute) and Memebership_Price (key attribute) both combined form the uniqueness of each MEMBERSHIP which is very different from other entities since there are two key attributes defining uniqueness of MEMBERSHIP.

4. MEDICAL_RECORD:



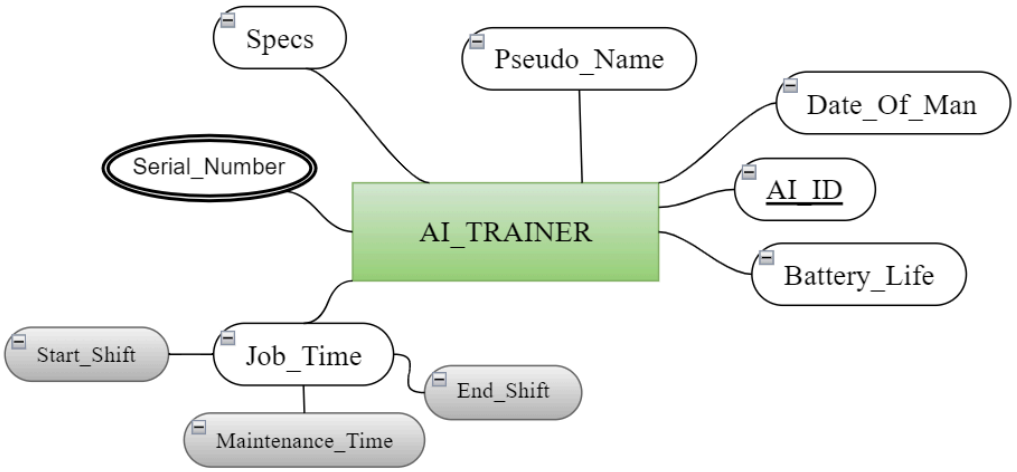
The 4th entity MEDICAL _RECORD is crucial to keep track of a MEMBER's medical record as they assess medical tests and examinations upon receiving their MEMBERSHIPS. MEDICAL_RECORD entity has non unique attributes such as **Blood_Type (attribute)**, **Infirmary_visit (attribute)**, **BMI (attribute)**, **Height (attribute)**, **Weight (attribute)**, **Food_restriction (attribute)** or even **Previous_injury (composite attribute)**. However, Med_ID (key attribute) differs from one person to another and can't be duplicated.

5. BRANCH:



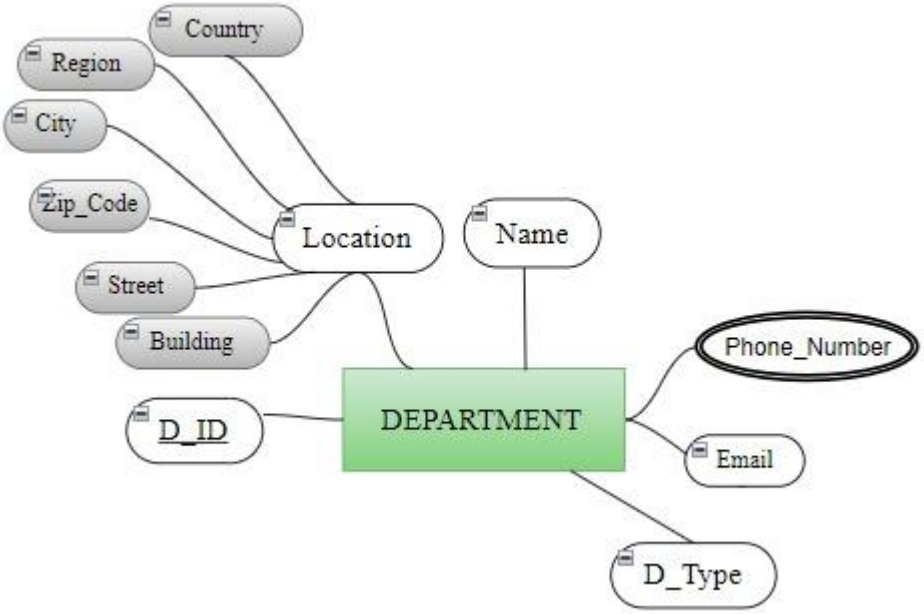
The BRANCH entity is the link and connection between GymnAIsium branches.The BRANCH entity has many non unique attributes such as **Name (attribute)**, **Open_hour (attribute), closing_hour (attribute), Number_of_staff (attribute), Phone_Number (attribute)** and **Date_of_opening (attribute)**. In addition there is **Location (composite attribute)** which is composed of **Region (attribute of Location), Zip_code (attribute), Country (attribute), City (attribute), Street (attribute)** and **Building (attribute).** Finally BRANCH can have a unique Branch_number (key attribute) to differentiate between different GymnAIsium branches.

6. AI_TRAINER:



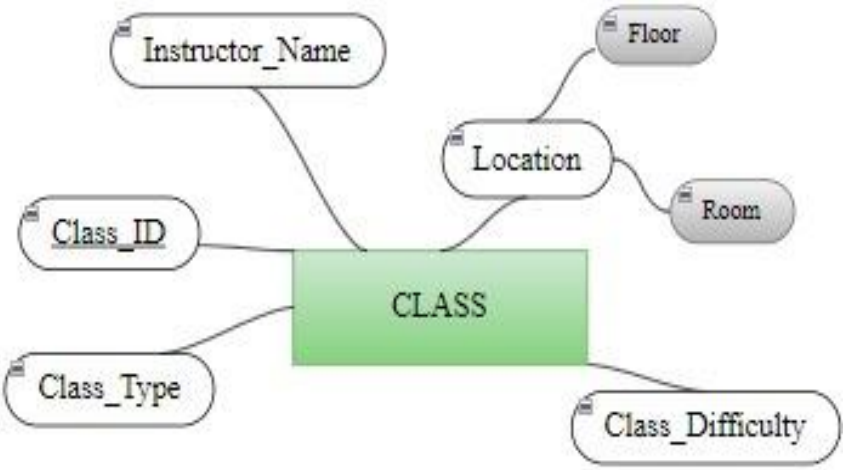
AI_TRAINER entity has the same role as PERSONAL_TRAINER except that it is AI. AI_TRAINER is composed of **four non-unique attributes** like **Pseudo_Name (attribute), Date_Of_Man (attribute), Battery_Life (attribute)** and **Specs (attribute)**. Additionally there is **(1 composite attribute) Job_Time** composed of **three simple attributes: Start_Shift (attribute), Maintenance_Time (attribute)** and **End_Shift (attribute)** then there is (**multi-valued attribute) Serial_Number**. Finally the unique part that identifies one AI_TRAINER from another is AI_ID (key attribute).

7. DEPARTMENT:



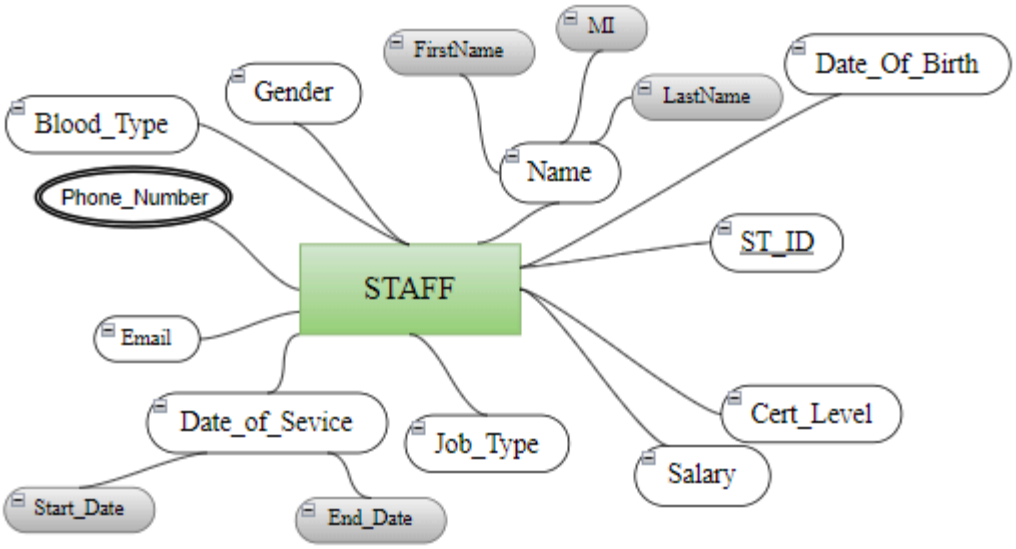
The DEPARTMENT entity represents the department of GymnAIsium. DEPARTMENT has attributes such as **Name (attribute), Email (attribute), D_Type (attribute)** which exists to indicate whether the type is a normal DEPARTMENT or a VR DEPARTMENT. In addition there is **Phone_Number (multi-valued attribute)** and **Location (composite attribute)** composed of **Country (attribute), Region (attribute), City (attribute), Zip_Code (attribute), Street (attribute)** and **Building (attribute)**. Finally the DEPARTMENT has a unique D_ID (key attribute) to differentiate between a DEPARTMENT and another one.

8. CLASS



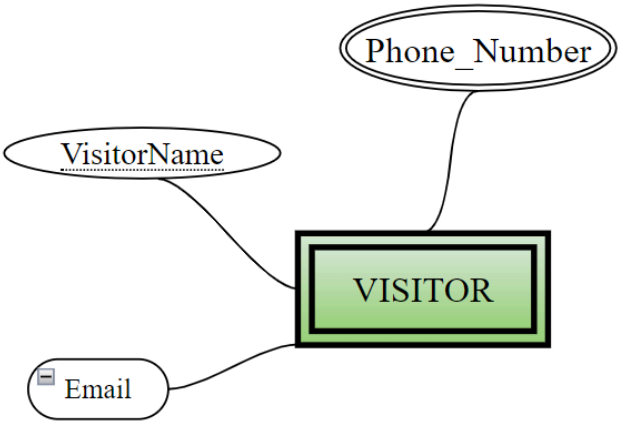
The CLASS entity represents classes of different difficulty levels within the fitness club. CLASS is composed of **three non-unique attributes: Instructor_Name (attribute), Class_Type (attribute), Class_Difficulty (attribute)** in addition to the **(composite attribute) Location** composed of: **Floor (attribute)** and **Room (attribute)**. Finally the unique attribute differentiating a CLASS from another is Class_ID (key attribute).

9. STAFF



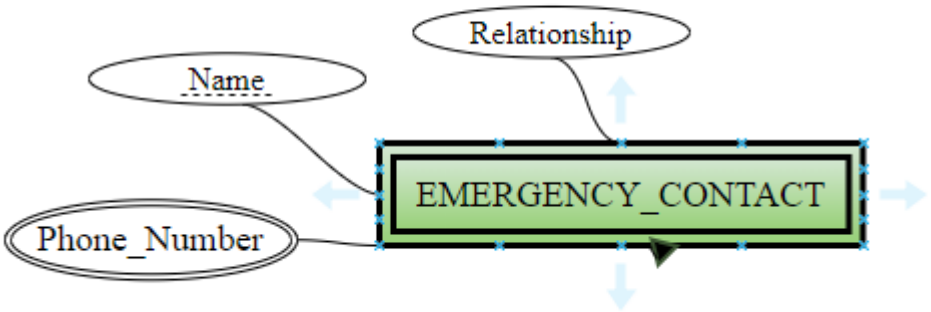
The STAFF entity type represents the staff of GymnAIsium. It includes simple, multivalued and composite attributes. The simple attributes are ST_ID (key attribute), **Gender, Blood_Type, Cert_Level, Salary, Job_Type, Email and Date_Of_Birth**. Composite attribute **Name** includes **FirstName, MI and LastName**, and **Date_of_Service** includes **Start_Date** and **End_Date**. The multivalued attribute in this entity type is **Phone_Number.**

10. VISITOR



VISITOR (weak entity) represents one time visitors of GymnAIsium. VISITOR is composed of one unique (partial key attribute) VisitorName and not key attribute because VISITOR is a weak entity. In addition it is also composed of **one non-unique (attribute) Email**.

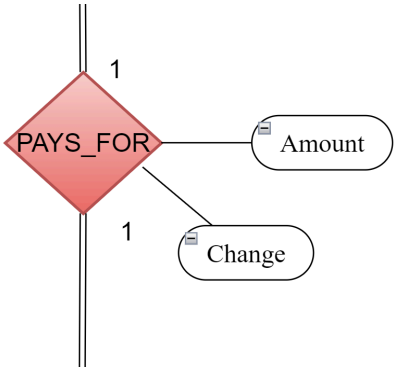
11. EMERGENCY_CONTACT



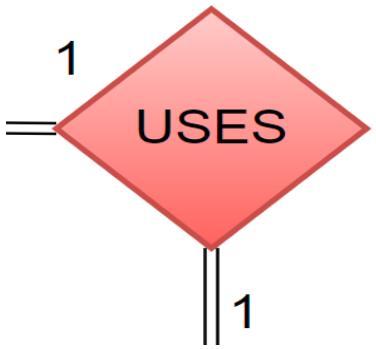
EMERGENCY_CONTACT (weak entity) represents the emergency contact information of members of GymnAIsium. EMERGENCY_CONTACT. The attribute Name acts as a partial key in this entity type.

IV-Relationships:

I. 1 TO 1 RELATIONSHIPS:

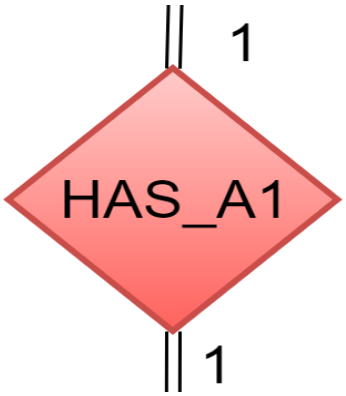


Every MEMBER Pays for a MEMBERSHIP. That's why we have to create a "PAYS_FOR" relation between MEMBER entity and MEMBERSHIP entity. The participation is total on both sides since each MEMBER has to pay for a MEMBERSHIP, while each membership is for a member. Their **relationship is a total 1:1.** Which means each MEMBER PAYS_FOR a MEMBERSHIP without any exception and each MEMBERSHIP is for one MEMBER only. This relation has two attributes: Amount and Change. The Amount attribute shows the amount paid by the member, and the Change attribute shows the amount of change given back to the member and if the member paid the exact amount the change will be 0.

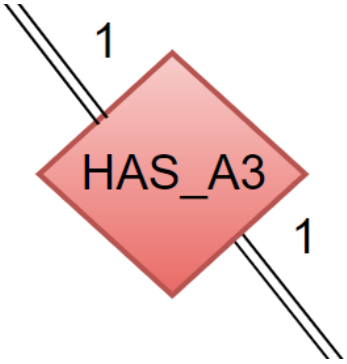


Every MEMBER USES an EQUIPMENT. That's why we have to create a "USES" relation between the MEMBER entity and the EQUIPMENT entity. The participation is total on both sides since each MEMBER USES an EQUIPMENT, while each EQUIPMENT is used by one MEMBER. Their

relationship is a total 1:1. Which means each MEMBER Uses an EQUIPMENT without any exception and each EQUIPMENT is used by one MEMBER at a time. This relation does not have any attributes.

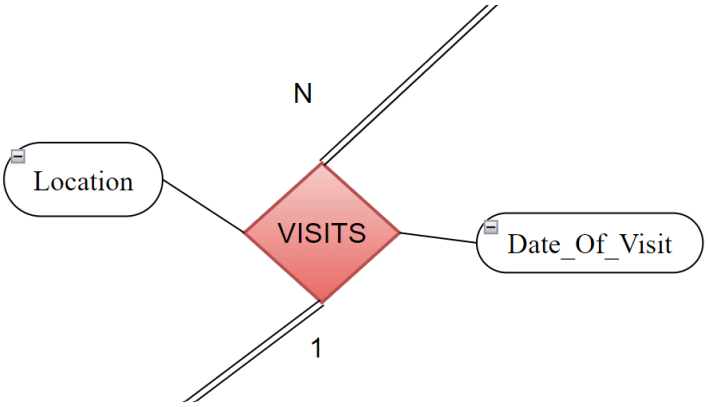


Every MEMBER Has a MEDICAL_RECORD. That's why we have to create a "HAS_A" relation between a MEMBER entity and MEDICAL_RECORD entity. The participation is total on both sides since each MEMBER has a MEDICAL_RECORD, while each MEDICAL_RECORD is for a MEMBER. Their **relationship is a total 1:1.** Which means each MEMBER HAS_A MEDICAL_RECORD without any exception and each MEDICAL_RECORD is for one MEMBER only. This relation does not have any attributes.

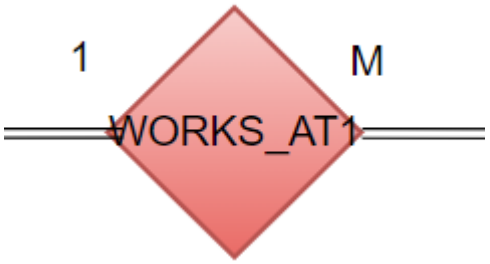


Every MEMBER Has an EMERGENCY_CONTACT. That's why we have to create a "HAS_A" relation between a MEMBER entity and EMERGENCY_CONTACT entity. The participation is total on both sides since each MEMBER HAS_A EMERGENCY_CONTACT, while each EMERGENCY_CONTACT is for a MEMBER. Their **relationship is a total 1:1.** Which means each MEMBER HAS_A EMERGENCY_CONTACT without any exception and each EMERGENCY_CONTACT is for one MEMBER only. This relation does not have any attributes.

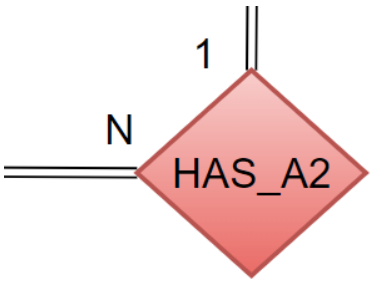
II.1 TO M RELATIONS :



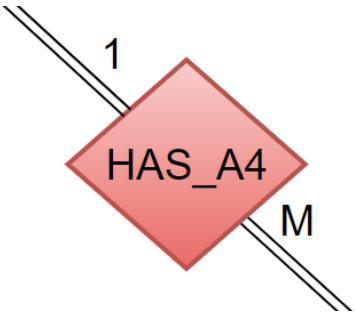
Many VISITOR (s) can visit with a MEMBER at a time. That's why we have to create a "VISITS" relation between VISITOR entity and MEMBER entity. The participation is total on both sides since each VISITOR can visit a MEMBER, while each MEMBER can have **many** VISITOR(s). In order to visit **multiple** times with a MEMBER, a VISITOR must schedule **many** visits. This relation has two attributes: Location and Date_Of_Visit. Date_Of_Visit **(Attribute)** to know when each visit is happening and Location(**Attribute**) to indicate the location of the visit. Therefore, the **relationship is a total participation 1:M.**



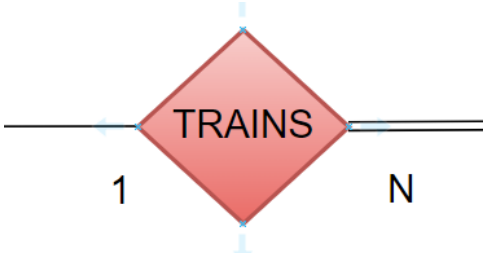
Many STAFF WORKS_AT1 DEPARTMENT. WORKS_AT1 links STAFF to DEPARTMENT. That's why we have to create a "WORKS_AT1" relation between STAFF entity and DEPARTMENT entity. The participation is total on both sides since each STAFF can WORKS_AT1 a DEPARTMENT, and each DEPARTMENT can have **many** STAFF. All staff members work in all departments , therefore it is a **total participation on both sides 1:M.** This relation does not have any attributes.



Each DEPARTMENT HAS_A2 EQUIPMENT. HAS_A2 links DEPARTMENT to EQUIPMENT. That's why we have to create a "HAS_A2" relation between DEPARTMENT entity and EQUIPMENT entity. The participation is total on both sides since each DEPARTMENT can have **many** EQUIPMENT, and each EQUIPMENT is in a DEPARTMENT. All DEPARTMENT(s) have EQUIPMENT, therefore it is a **total participation on both sides 1:M.** This relation does not have any attributes.

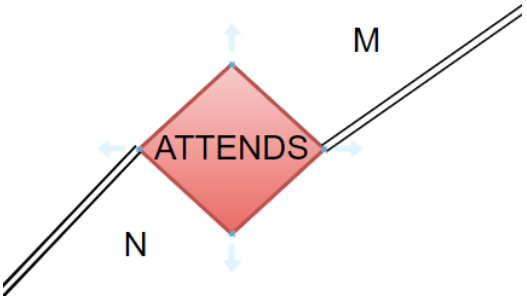


Each DEPARTMENT HAS_A4 BRANCH. HAS_A4 links DEPARTMENT to BRANCH. That's why we have to create a "HAS_A4" relation between DEPARTMENT entity and BRANCH entity. The participation is total on both sides since each DEPARTMENT can have a BRANCH, and each BRANCH has **several** DEPARTMENT(s), therefore it is a **total participation on both sides 1:M.** This relation does not have any attributes.



Each AI_TRAINER TRAINS a MEMBER. TRAINS links AI_TRAINER to MEMBER. That's why we have to create a "TRAINS" relation between AI_TRAINER entity and MEMBER entity. The participation is total on the MEMBER's side since all members have to have an AI_TRAINER training them, however it is partial participaction on the AI_TRAINER's side since not all AI_TRAINER(s) have to be in use all the time. Since one AI_TRAINER can train many members, their **relationship is a partial 1:M.**

III. M TO N RELATIONS :



Each MEMBER can ATTENDS CLASS(s). ATTENDS links MEMBER to CLASS. That's why we have to create an "ATTENDS" relation between MEMBER entity and CLASS entity. The participation is total on both sides since each MEMBER can ATTEND **several** CLASS, and each CLASS can have **several** MEMBER(s) enrolled in it, therefore it is a **total participation on both sides N:M.** This relation does not have any attributes.

IV. ER to Relational Mapping Algorithms

Upon designing the relational model (ER diagram) for GymnAIsium, the ER diagram is to be translated into a higher-level relational data model following the seven-step algorithm. Following is a detailed description of the mapping process.

STEP 1: Mapping of Regular Entity Types

The first step includes mapping the regular entity types into relations. Every regular entity will include simple attributes and only one unique primary key (denoted by underlining the attribute). The strong entities for GymnAIsium are **MEMBER, MEMBERSHIP, MEDICAL_RECORD, EQUIPMENT, BRANCH, DEPARTMENT, STAFF, PERSONAL_TRAINER, AI_TRAINER** and **CLASS**

1. MEMBER

M_ID	**FirstName**	**MI**	**LastName**	**Email**	**Gender**	**Date_Of_Birth**

The MEMBER entity contains simple, composite and multivalued attributes. The multivalued attribute *Phone_Number* is not represented in this relation. The primary key attribute ***M_ID*** is underlined to be able to distinguish it from the simple attributes. The MEMBER entity has ***Name*** as a composite attribute which is composed of ***FirstName***, ***MI*** and ***LastName*** included in the relation.

2. MEMBERSHIP

Membership_Number	**Membership_Name**	**Membership_Type**	**Membership_Price**

The MEMBERSHIP entity only contains simple attributes. The primary key attribute ***Membership_Number*** is underlined to distinguish it from other simple attributes.

3. MEDICAL_RECORD

Med_ ID	Heig ht	Weig ht	Blood_Ty pe	Food_Restrictio n	Previous_Inju ry

The MEDICAL_RECORD entity contains simple and derived attributes. The primary key attribute ***Med_ID*** is underlined to distinguish it from other simple attributes. The derived attribute ***BMI*** is not represented in the relation.

4. EQUIPMENT

Equipment_SN umber	Equipment_ Name	Equipment _Type	Equipment_ Price	YOM	Floor	Room

The EQUIPMENT entity contains simple and composite attributes. The primary key ***Equipment_SNumber*** is underlined to distinguish it from other simple attributes. The composite attribute ***Location*** consists of ***Floor*** and ***Room*** represented in the relation.

5. BRANCH

Br_ID	Name	Number_Of_St aff	Date_Of_Ope ning	Open_Hour	Close_Hour
Region	Zip_code	Country	City	Street	Building

The BRANCH entity contains simple, composite and multivalued attributes. The primary key ***Br_ID*** is underlined to distinguish it from other attributes. In addition, the composite attribute ***Location*** consists of attributes ***Region, Zip_code, Country, City, Street*** and ***Building***. However, the multivalued attribute ***Phone_Number*** is not included in this relation.

6. DEPARTMENT

D_ID	Name	D_Type	Email	Region	Zip_code
Country	City	Street	Building		

The DEPARTMENT entity contains simple, composite and multivalued attributes. The primary key ***D_ID*** is underlined to distinguish it from other attributes. In addition, the composite ***Location*** attribute consists of attributes ***Region, Zip_code, Country, City, Street*** and ***Building***. However, the multivalued attribute ***Phone_Number*** is not included in this relation.

7. STAFF

ST_ID	FirstName	MI	LastName	Gender	Blood_Type	Date_Of_Birth
Email	Job_Type	Start_Date	End_Date	Cert_Level	Salary	

The DEPARTMENT entity contains simple, composite and multivalued attributes. The primary key ***ST_ID*** is underlined to distinguish it from other attributes. In addition, the composite attribute ***Name*** consists of attributes ***FirstName, MI,*** and ***LastName*** and the other composite attribute ***Date_Of_Service*** consists of ***Start_Date*** and ***End_Date***. However, the multivalued attribute ***Phone_Number*** is not included in this relation.

8. AI_TRAINER

AI_ID	Psuodo_Name	Date_Of_Man	Battery_Life	Specs	Start_Shift	End_Shift
Maintenance_Time						

The AI_TRAINER entity contains simple, composite and multivalued attributes. The primary key ***AI_ID*** is underlined to distinguish it from other attributes. In addition, the composite attribute ***Job_Time*** consists of ***Start_Shift***, ***Maintenance_Time*** and ***End_Shift.*** However, the multivalued attribute ***Serial_Number*** is not included in this relation.

9. CLASS

Class_ID	Class_Type	Instructor_Name	Class_Difficulty	Floor	Room

The CLASS entity contains simple and composite attributes. The primary key ***Class_ID*** is underlined to distinguish it from other attributes. The composite attribute ***Location*** consists of attributes ***Floor*** and ***Room***.

STEP 2: Mapping of Weak Entity Types

The second step includes mapping the weak entity types into relations. Every weak entity will include foreign key attribute which is the primary key of the owner entity type. The partial key and the foreign key combined compose the primary key of the weak entity. The weak entities for GymnAIsium are **VISITOR** and **EMERGENCY_CONTACT.**

1. **VISITOR**

M_ID	**Visitor_Name**	**Email**	**Date_Of_Visit**	**Location**

The weak entity VISITOR includes simple and multivalued. The multivalued attribute ***Phone_Number*** is not included in this relation. The partial key of this entity is ***Visitor_Name***. ***Visitor_Name*** and ***M_ID*** attributes together form the primary key of this entity.

2. **EMERGENCY_CONTACT**

M_ID	**Name**	**Relationship**

The weak entity EMERGENCY_CONTACT includes only simple attributes. The partial key of this entity is ***Name***. ***Name*** and ***M_ID*** combined form the primary key of this entity.

STEP 3: Mapping 1:1 Relationship types

Step three includes mapping of the binary one-to-one relationships. In this mapping process, it is preferable to use the foreign key approach out of the other three approaches as it is more applicable to the design. The ***HAS_A1***, ***HAS_A3***, ***USES*** and ***PAYS_FOR*** are the three one to one relationships to be mapped in this model.

1. **MEDICAL_RECORD (HAS_A1)**

Med_ID	**M_ID**	**Height**	**Weight**	**Blood_Type**	**Food_Restriction**	**Previous_Injury**

Every member has a medical record. The "HAS_A1" relation links **MEMBER** entity and ***MEDICAL_RECORD*** entity. It is a total participation on both sides of the relation. ***M_ID*** primary key of ***MEMBER*** entity was added as foreign key to ***MEDICAL_RECORD*** relation.

2. **EMERGENCY_CONTACT (HAS_A3)**

M_ID	**Phone_Number**	**Relationship**	**Name**

Every member has an emergency contact. The "HAS_A3" relation links **MEMBER** entity and **EMERGENCY_CONTACT** entity. It is a total participation on both sides of the relationship. ***M_ID*** a primary key of the MEMBER entity, was added as foreign key to the EMERGENCY_CONTACT relation.

3. **EQUIPMENT (USES)**

Equipment_SNumber	**M_ID**	**Equipment_Price**	**Equipment_Type**
Equipment_Name	**Location**	**YOM**	

Every member uses equipment. The "USES" relation links **MEMBER** entity and **EQUIPMENT** entity. It is a total participation on both sides of the relationship. ***M_ID*** a primary key of the **MEMBER** entity, was added as foreign key to the EQUIPMENT relation.

4. MEMBERSHIP (PAYS_FOR)

Membership_Number	**M_ID**	**Membership_Type**	**Membership_Price**	**Membership_Name**	**Amount**

Every member has a membership. The "PAYS_FOR" relation links **MEMBER** entity and **MEMBERSHIP** entity. It is a total participation on both sides of the relation. ***M_ID*** primary key of ***MEMBER*** entity was added as foreign key to PAYS_FOR relation.

STEP 4: Mapping of Binary 1: N Relationship types:

In step four, we will map the binary one to many relationships. A foreign key is added on the "many" sides of the relationship. The foreign key acts as the primary key for the other entity type in this relationship. We also must include any simple attribute of the one-to-many relationship. The relationships that will be mapped are VISITS, HAS_A2, HAS_A4, TRAINS, WORKS_AT.

1. VISITOR(VISITS):

M_ID	**VisitorName**	**Phone_Number**	**Email**	**Location**	**Date_Of_VisitV**

Every ***VISITOR*** visits the gym with a ***MEMBER***. The "VISITS" relationship link ***VISITOR*** and ***MEMBER*** entity. It is a partial participation from both sides of the relation. Since ***VISITOR*** is on the many sides of the relationship, ***M_ID*** is added as foreign key to the relation. But Since ***VISITOR*** already has ***M_ID*** , nothing changes in this relation.

2. EQUIPMENT (HAS_A2):

Equipment_ SNumber	D_ID	Equipment _Name	Equipmen t_Type	Equipment _Price	YO M	Floor	Room

Every ***DEPARTMENT*** has ***EQUIPMENT***. The "HAS_A2" relationship links ***EQUIPMENT*** and ***DEPARTMENT*** entity. It is a total participation from both sides of the relation. Since ***EQUIPMENT*** is on the many side of the relationship, ***Equipment_SNumber*** is the primary key of the relation and ***D_ID*** primary key of ***DEPARTMENT*** was added as the foreign key.

3. BRANCH (HAS_A4):

Br_ID	D_ID	Name	Number_Of_ Staff	Date_Of_Op ening	Open_H our	Close_Hour
Region	Buildi ng	Zip_co de	Country	City	Street	

Every ***DEPARTMENT*** has a ***BRANCH***. The "HAS_A4" relationship links ***BRANCH*** and ***DEPARTMENT*** entity. It is a total participation from both sides of the relation. Since ***BRANCH*** is on the many sides of the relationship, ***Br_ID*** is the primary key of the relation and ***D_ID*** primary key of ***DEPARTMENT*** was added as the foreign key.

4. MEMBER (TRAINS):

M_ID	AI_ID	FirstName	MI	LastName	Email	Gender	Date_Of_Birth

Every ***AI_TRAINER*** trains ***MEMBER***. The "TRAINS" relationship links ***AI_TRAINER*** and ***MEMBER*** entity. It is total from the ***MEMBER*** side and partial from the ***AI_TRAINER*** side of the relation. Since ***MEMBER*** is on the many sides of the relationship, ***M_ID*** is the primary key of the relation and ***AI_ID*** primary key of ***AI_TRAINER*** was added as the foreign key.

5. STAFF (WORKS_AT1):

ST_ID	**D_ID**	**FirstName**	**MI**	**LastName**	**Gender**	**Blood_Type**
Email	**Date_Of_Birth**	**Job_Type**	**Start_Date**	**End_Date**	**Cert_Level**	**Salary**

Every ***STAFF*** works at a ***DEPARTMENT***. The "WORKS_AT1" relationship links ***STAFF*** and ***DEPARTMENT*** entity. It is a total participation from both sides of the relation. Since ***STAFF*** is on the many sides of the relationship, ***ST_ID*** is the primary key of the relation and ***D_ID*** primary key of ***DEPARTMENT*** was added as the foreign key.

STEP 5: Mapping of M: N Relationship types:

In step five, we will map the binary many-to-many relationships. For each many-to-many relationship we are going to create a new relation which includes, as foreign keys, the primary keys of all participating relations. Their combination will form the primary key of this newly created relation. We will also include any normal attribute of the many-to-many relationship. The relationship to be mapped is ATTENDS.

1. ATTENDS:

C_ID	**MEM_ID**

Many ***MEMBER(s)*** attend many ***CLASS(s)***. The "ATTENDS" relationship links ***MEMBER*** and the ***CLASS*** entity. We create a new relation called "ATTENDS" that include the primary key of both the ***CLASS*** and ***MEMBER*** entities. We renamed **CLASS_*ID*** which is the primary key of ***CLASS*** to ***C_ID*** and we renamed **M_ID** which is the primary key of ***MEMBER*** to ***MEM_ID***. The combination of both these added keys represent the primary key of the "ATTENDS" relation.

STEP 6: Mapping of Multivalued Attributes:

In this step, we are mapping the multivalued attributes which we weren't concerned with before. For each multivalued attribute we create a new relation containing the related attribute and the primary key of the entity to which it belongs. Their combination will represent the primary key of the newly created relation. We have three multivalued attributes which are: the member's phone number, the staff phone number, the visitor's phone number, the branch phone number, department phone number.

1. MEMBER_Phone_Number:

M_ID	**Phone_Number**

The multivalued attribute Phone_Number belongs to the **MEMBER** entity. To represent it, we create a relation called "**MEMBER_Phone_Number**". Its primary key is composed of **M_ID**, the primary key of the **MEMBER** entity, and the attribute Phone_Number which represents the multiple phones numbers a member can have.

2. STAFF_Phone_Number:

ST_ID	**Phone_Number**

The multivalued attribute Phone_Number belongs to the **STAFF** entity. To represent it, we create a relation called "**STAFF_Phone_Number**". Its primary key is composed of **ST_ID**, the primary key of the **STAFF** entity, and the attribute Phone_Number which represents the multiple phones numbers a staff can have.

3. VISITOR_Phone_Number:

VisitorName	**M_ID**	**Phone_Number**

The multivalued attribute Phone_Number belongs to the **VISITOR** entity. To represent it, we create a relation called "**VISITOR_Phone_Number**". Its primary key is composed of **VisitorName**,

M_ID the primary key of the **VISITOR** entity, and the attribute Phone_Number which represents the multiple phones numbers a visitor can have.

4. BRANCH_Phone_Number:

BR_ID	Phone_Number

The multivalued attribute Phone_Number belongs to the **BRANCH** entity. To represent it, we create a relation called "**BRANCH_Phone_Number**". Its primary key is composed of **BR_ID**, the primary key of the **BRANCH** entity, and the attribute Phone_Number which represents the multiple phones numbers a branch can have.

5. DEPARTMENT_Phone_Number:

D_ID	Phone_Number

The multivalued attribute Phone_Number belongs to the **DEPARTMENT** entity. To represent it, we create a relation called "**DEPARTMENT_Phone_Number**". Its primary key is composed of **D_ID**, the primary key of the **DEPARTMENT** entity, and the attribute Phone_Number which represents the multiple phones numbers a department can have.

STEP 7: Mapping of N-ary Relationship Types:

In this step, we are going to map the N-ary Relationship types. We should create a new relation containing the primary keys of all participating entities and any simple attributes of the relationship type. In our design we have no N-ary relationship types, so this step is not applicable here.

FINAL STEP: Final Displays

MEMBER

M_ID	FirstName	MI	LastName	Email	Gender	Date_Of_Birth

MEMBERSHIP

Membership_Number	Membership_Name	Membership_Type	Membership_Price

MEDICAL_RECORD

Med_ID	Height	Weight	Blood_Type	Food_Restriction	Previous_Injury

EQUIPMENT

Equipment _SNumber	Equipment _Name	Equipment _Type	Equipment_ Price	YOM	Floor	Room

BRANCH

Br_ID	Name	Number_Of_ Staff	Date_Of_Opening	Open_Hour	Close_Hour
Region	Zip_code	Country	City	Street	Building

DEPARTMENT

D_ID	Name	D_Type	Email	Region	Zip_code
Country	City	Street	Building		

STAFF

ST_ID	FirstName	MI	LastName	Gender	Blood_Type	Date_Of_Birth
Email	Job_Type	Start_Date	End_Date	Cert_Level	Salary	

AI_TRAINER

AI_ID	Pseudo_Name	Date_Of_Man	Battery_Life	Specs	Start_Shift	End_Shift
Maintenance_Time						

CLASS

Class_ID	Class_Type	Instructor_Name	Class_Difficulty	Floor	Room

VISITOR

M_ID	Visitor_Name	Email	Date_Of_Visit	Location

EMERGENCY_CONTACT

M_ID	Name	Relationship

MEDICAL_RECORD (HAS_A1)

Med_ID	M_ID	Height	Weight	Blood_Type	Food_Restriction	Previous_Injury

EMERGENCY_CONTACT (HAS_A3)

M_ID	Relationship	Name

EQUIPMENT (USES)

Equipment_SNumber	M_ID	Equipment_Price	Equipment_Type
Equipment_Name	Location	YOM	

MEMBERSHIP (PAYS_FOR)

Membership_Number	M_ID	Membership_Type	Membership_Price	Membership_Name

VISITOR(VISITS)

M_ID	VisitorName	Phone_Number	Email	Location	Date_Of_Visit

EQUIPMENT (HAS_A2)

Equipment_SNumber	D_ID	Equipment_Name	Equipment_Type	Equipment_Price	YOM	Floor	Room

BRANCH(HAS_A4)

D_ID	Br_ID	Name	Number_Of_Staff	Date_Of_Opening	Open_Hour	Close_Hour
Region	Building	Zip_code	Country	City	Street	

MEMBER (TRAINS)

M_ID	AI_ID	FirstName	MI	LastName	Email	Gender	Date_Of_Birth

STAFF (WORKS_AT1)

ST_ID	D_ID	FirstName	MI	LastName	Gender	Blood_Type
Email	Date_Of_Birth	Job_Type	Start_Date	End_Date	Cert_Level	Salary

ATTENDS

C_ID	MEM_ID

MEMBER_Phone_Number

M_ID	Phone_Number

STAFF_Phone_Number

ST_ID	Phone_Number

VISITOR_Phone_Number

VisitorName	M_ID	Phone_Number

BRANCH_Phone_Number

BR_ID	Phone_Number

DEPARTMENT_Phone_Number

D_ID	Phone_Number

VI. SQL CODE:

MEMBER

```
Create Table MEMBER
(
M_ID VARCHAR(9) PRIMARY KEY,
AI_ID VARCHAR(9) NOT NULL,
FOREIGN KEY (AI_ID) REFERENCES AI_TRAINER(AI_ID),
FirstName VARCHAR(20) NOT NULL,
MI CHAR(1),
LastName VARCHAR(20) NOT NULL,
Gender CHAR NOT NULL CHECK(GENDER IN ('F', 'M')),
Date_Of_Birth DATE NOT NULL,
Email VARCHAR(20)

);
```

MEMBERSHIP

```
CREATE TABLE MEMBERSHIP
(

 Membership_number NUMERIC PRIMARY KEY CHECK((Membership_number>0)),
 Membership_Name VARCHAR(10) NOT NULL,
 Membership_Type VARCHAR(10) NOT NULL,
 Membership_Price NUMBER CHECK((Membership_Price>0)),
 M_ID VARCHAR(8),
 FOREIGN KEY(M_ID) REFERENCES MEMBER(M_ID)

);
```

DEPARTMENT

```
CREATE TABLE DEPARTMENT
(

 D_ID CHAR(9) PRIMARY KEY NOT NULL,
 Name VARCHAR(10) NOT NULL,
 D_Type VARCHAR(3) NOT NULL CHECK(D_Type IN('AI','NAI')),
 Email VARCHAR(320),
 Region VARCHAR(20) NOT NULL,
 Zip_code VARCHAR(12) NOT NULL,
 Country VARCHAR(13) NOT NULL,
 City VARCHAR(163) NOT NULL,
 Street VARCHAR(39) NOT NULL,
 Building VARCHAR(30) NOT NULL,
);
```

BRANCH

```
CREATE TABLE BRANCH
(
Br_ID VARCHAR(9) PRIMARY KEY ,
Name VARCHAR(15) NOT NULL,
Number_Of_Staff NUMBER(6,0),
Date_Of_Opening DATE,
Open_Hour TIMESTAMP(6),
Close_Hour TIMESTAMP(6),
Region VARCHAR(15),
Zip_code NUMBER(8,0),
Country VARCHAR(20),
City VARCHAR(10),
Street VARCHAR(15),
Building VARCHAR(15)
);
```

EQUIPMENT

```
CREATE TABLE EQUIPMENT
(
Equipment_SNumber NUMBER(9,0) PRIMARY KEY ,
Equipment_Name VARCHAR(15) NOT NULL,
Equipment_Type VARCHAR(25) ,
Equipment_Price NUMBER(10,3) CHECK(Equipment_Price>0),
YOM NUMBER(4,0),
Floor NUMBER(3,0),
Room VARCHAR(20),
M_ID VARCHAR(8),
FOREIGN KEY (M_ID) REFERENCES MEMBER(M_ID)
);
```

AI_TRAINER

```
Create Table AI_TRAINER
(
AI_ID VARCHAR(9) PRIMARY KEY,
Pseudo_Name VARCHAR(20) NOT NULL,
Battery_Life INT CHECK(Battery_Life<=100 AND Battery_Life>=0),
Date_Of_Man DATE NOT NULL,
Start_Shift DATE NOT NULL,
Maintenance_Time DATE,
End_Shift DATE,
Specs NUMBER(1) CHECK(Specs IN (1,0))

);
```

VISITOR

```
CREATE TABLE VISITOR
(
 M_ID VARCHAR(9),
 VisitorName VARCHAR(20) NOT NULL,
 Email VARCHAR(20),
 Date_Of_Visit VARCHAR(20) NOT NULL,
 Location VARCHAR(25) NOT NULL,
 PRIMARY KEY(M_ID,VisitorName),
 FOREIGN KEY(M_ID) REFERENCES MEMBER(M_ID)
);
```

STAFF

```
CREATE TABLE STAFF
(
ST_ID VARCHAR(9) PRIMARY KEY,
Cert_Level  VARCHAR(10),
Date_Of_Birth Date NOT NULL,
Salary NUMERIC,
Job_Type VARCHAR(20) NOT NULL,
Start_Date DATE,
End_Date DATE,
Email VARCHAR(30),
Blood_Type VARCHAR(3) NOT NULL,
Gender CHAR NOT NULL CHECK(GENDER IN('F','M')),
FirstName VARCHAR(15) NOT NULL,
MI CHAR,
LastName VARCHAR(15) NOT NULL
);
```

MEDCIAL_RECORD

```
CREATE TABLE MEDICAL_RECORD
(
Med_ID VARCHAR(9) PRIMARY KEY,
BMI NUMERIC(2) CHECK (BMI>=0 AND BMI<=90),
Blood_type VARCHAR(3) NOT NULL,
Previous_injury VARCHAR(20),
Food_restriction VARCHAR(20),
Height  DEC(9,2) CHECK(Height>0),
Weight DEC(9,2) CHECK(Weight>0),
M_ID VARCHAR(9),
FOREIGN KEY(M_ID) REFERENCES MEMBER(M_ID)


);
```

EMERGENCY_CONTACT

```
CREATE TABLE EMERGENCY_CONTACT
(
M_ID VARCHAR(9),
Relationship VARCHAR(20) NOT NULL,
Phone_Number NUMERIC(8),
Name VARCHAR(20),
PRIMARY KEY (M_ID,Name),
FOREIGN KEY(M_ID) REFERENCES MEMBER(M_ID)

);
```

CLASS

```
CREATE TABLE CLASS
(
Class_ID NUMBER(9,0) PRIMARY KEY ,
Class_Type VARCHAR(15) ,
Instructor_Name VARCHAR(15) NOT NULL,
Class_Difficulty VARCHAR(30) CHECK(CLASS_DIFFICULTY IN
('Beginner','Intermediate','Professional')),
Floor NUMBER(3,0),
Room VARCHAR(20)
);
```

ATTENDS

```
CREATE TABLE ATTENDS
(
 C_ID NUMBER(9,0),
 MEM_ID VARCHAR(9) ,
 PRIMARY KEY (MEM_ID, C_ID),
 FOREIGN KEY(MEM_ID) REFERENCES MEMBER(M_ID) ,
 FOREIGN KEY(C_ID) REFERENCES CLASS (Class_ID)
);
```

MEMBER_Phone_Number

```
CREATE TABLE MEMBER_Phone_Number
(
M_ID CHAR(9),
Phone_Number NUMERIC(8),
PRIMARY KEY(M_ID,Phone_Number)
);
```

STAFF_Phone_Number

```
CREATE TABLE STAFF_Phone_Number
(
ST_ID VARCHAR(9),
Phone_Number NUMERIC(8),
PRIMARY KEY(ST_ID,Phone_Number),
FOREIGN KEY(ST_ID) REFERENCES STAFF(ST_ID)
);
```

DEPARTMENT_Phone_Number

```
CREATE TABLE DEPARTMENT_PHONE_NUMBER
(
 DEPT_ID VARCHAR(9) ,
 Phone_Number VARCHAR(15),
 PRIMARY KEY (DEPT_ID, Phone_Number),
 FOREIGN KEY(DEPT_ID) REFERENCES DEPARTMENT(D_ID)
);
```

VISITOR_PHONE_NUMBER

```
CREATE TABLE VISITOR_PHONE_NUMBER
(

 VisitorName VARCHAR(20),
 Phone_Number VARCHAR(8),
 M_ID VARCHAR(9),
 PRIMARY KEY(VisitorName,Phone_Number,M_ID)

);
```

BRANCH_PHONE_NUMBER

```
CREATE TABLE BRANCH_PHONE_NUMBER
(
 Branch_ID VARCHAR(9) ,
 Phone_Number VARCHAR(15),
 PRIMARY KEY (Branch_ID, Phone_Number),
 FOREIGN KEY(Branch_ID) REFERENCES BRANCH(Br_ID)
);
```

ALTER STATEMENTS:

```
ALTER TABLE EQUIPMENT ADD FOREIGN KEY(D_ID) REFERENCES DEPARTMENT(D_ID);

ALTER TABLE BRANCH ADD FOREIGN KEY(D_ID) REFERENCES DEPARTMENT(D_ID);

ALTER TABLE STAFF ADD FOREIGN KEY(D_ID) REFERENCES DEPARTMENT(D_ID);

ALTER TABLE DEPARTMENT_PHONE_NUMBER ADD FOREIGN KEY(D_ID) REFERENCES
DEPARTMENT(D_ID);

ALTER TABLE MEMBER_PHONE_NUMBER ADD FOREIGN KEY(M_ID) REFERENCES
MEMBER(M_ID);

ALTER TABLE VISITOR_PHONE_NUMBER ADD FOREIGN KEY(VISITORNAME,M_ID) REFERENCES
VISITOR(VISITORNAME,M_ID);
```

VII. TABLE DESCRIPTIONS

MEMBER

Column Name	Data Type	Nullable	Default	Primary Key
M_ID	VARCHAR2(9)	No	-	1
AI_ID	VARCHAR2(9)	No	-	-
FIRSTNAME	VARCHAR2(20)	No	-	-
MI	CHAR(1)	Yes	-	-
LASTNAME	VARCHAR2(20)	No	-	-
GENDER	CHAR(1)	No	-	-
DATE_OF_BIRTH	DATE	No	-	-
EMAIL	VARCHAR2(20)	Yes	-	-

1 - 8

MEMBERSHIP

Column Name	Data Type	Nullable	Default	Primary Key
MEMBERSHIP_NUMBER	NUMBER	No	-	1
MEMBERSHIP_NAME	VARCHAR2(10)	No	-	-
MEMBERSHIP_TYPE	VARCHAR2(10)	No	-	-
MEMBERSHIP_PRICE	NUMBER	Yes	-	-
M_ID	VARCHAR2(8)	Yes	-	-

1 - 5

DEPARTMENT

Column Name	Data Type	Nullable	Default	Primary Key
D_ID	CHAR(9)	No	-	1
NAME	VARCHAR2(10)	No	-	-
D_TYPE	VARCHAR2(3)	No	-	-
EMAIL	VARCHAR2(320)	Yes	-	-
REGION	VARCHAR2(20)	No	-	-
ZIP_CODE	VARCHAR2(12)	No	-	-
COUNTRY	VARCHAR2(13)	No	-	-
CITY	VARCHAR2(163)	No	-	-
STREET	VARCHAR2(39)	No	-	-
BUILDING	VARCHAR2(30)	No	-	-
BR_ID	VARCHAR2(9)	Yes	-	-
ST_ID	VARCHAR2(9)	Yes	-	-

1 - 12

BRANCH

Column Name	Data Type	Nullable	Default	Primary Key
BR_ID	VARCHAR2(9)	No	-	1
NAME	VARCHAR2(15)	No	-	-
NUMBER_OF_STAFF	NUMBER(6,0)	Yes	-	-
DATE_OF_OPENING	DATE	Yes	-	-
OPEN_HOUR	TIMESTAMP(6)	Yes	-	-
CLOSE_HOUR	TIMESTAMP(6)	Yes	-	-
REGION	VARCHAR2(15)	Yes	-	-
ZIP_CODE	NUMBER(8,0)	Yes	-	-
COUNTRY	VARCHAR2(20)	Yes	-	-
CITY	VARCHAR2(10)	Yes	-	-
STREET	VARCHAR2(15)	Yes	-	-
BUILDING	VARCHAR2(15)	Yes	-	-
D_ID	CHAR(9)	No	-	-

1 - 13

EQUIPMENT

Column Name	Data Type	Nullable	Default	Primary Key
EQUIPMENT_SNUMBER	NUMBER(9,0)	No	-	1
EQUIPMENT_NAME	VARCHAR2(15)	No	-	-
EQUIPMENT_TYPE	VARCHAR2(25)	Yes	-	-
EQUIPMENT_PRICE	NUMBER(10,3)	Yes	-	-
YOM	NUMBER(4,0)	Yes	-	-
FLOOR	NUMBER(3,0)	Yes	-	-
ROOM	VARCHAR2(20)	Yes	-	-
M_ID	VARCHAR2(8)	Yes	-	-
D_ID	CHAR(9)	Yes	-	-

1 - 9

AI_TRAINER

Column Name	Data Type	Nullable	Default	Primary Key
AI_ID	VARCHAR2(9)	No	-	1
PSEUDO_NAME	VARCHAR2(20)	No	-	-
BATTERY_LIFE	NUMBER	Yes	-	-
DATE_OF_MAN	DATE	No	-	-
START_SHIFT	DATE	No	-	-
MAINTENANCE_TIME	DATE	Yes	-	-
END_SHIFT	DATE	Yes	-	-
SPECS	NUMBER(1,0)	Yes	-	-

1 - 8

VISITOR

Column Name	Data Type	Nullable	Default	Primary Key
M_ID	VARCHAR2(9)	No	-	1
VISITORNAME	VARCHAR2(20)	No	-	2
EMAIL	VARCHAR2(20)	Yes	-	-
DATE_OF_VISIT	VARCHAR2(20)	No	-	-
LOCATION	VARCHAR2(25)	No	-	-

1 - 5

STAFF

Column Name	Data Type	Nullable	Default	Primary Key
ST_ID	VARCHAR2(9)	No	-	1
CERT_LEVEL	VARCHAR2(10)	Yes	-	-
DATE_OF_BIRTH	DATE	No	-	-
SALARY	NUMBER(6,2)	Yes	-	-
JOB_TYPE	VARCHAR2(12)	No	-	-
START_DATE	DATE	Yes	-	-
END_DATE	DATE	Yes	-	-
EMAIL	VARCHAR2(30)	Yes	-	-
PHONE_NUMBER	NUMBER(8,0)	No	-	-
BLOOD_TYPE	VARCHAR2(3)	No	-	-
GENDER	CHAR(1)	No	-	-
FIRSTNAME	VARCHAR2(15)	No	-	-
MI	CHAR(1)	Yes	-	-
LASTNAME	VARCHAR2(15)	No	-	-
D_ID	CHAR(9)	No	-	-

1 - 15

MEDICAL_RECORD

Column Name	Data Type	Nullable	Default	Primary Key
MED_ID	VARCHAR2(9)	No	-	1
BMI	NUMBER(2,0)	Yes	-	-
BLOOD_TYPE	VARCHAR2(3)	No	-	-
PREVIOUS_INJURY	VARCHAR2(20)	Yes	-	-
FOOD_RESTRICTION	VARCHAR2(20)	Yes	-	-
HEIGHT	NUMBER(9,2)	Yes	-	-
WEIGHT	NUMBER(9,2)	Yes	-	-

1 - 7

EMERGENCY_CONTACT

Column Name	Data Type	Nullable	Default	Primary Key
M_ID	VARCHAR2(9)	No	-	1
NAME	VARCHAR2(20)	No	-	2
RELATIONSHIP	VARCHAR2(20)	No	-	-
PHONE_NUMBER	NUMBER(8,0)	Yes	-	-

1 - 4

CLASS

Column Name	Data Type	Nullable	Default	Primary Key
CLASS_ID	NUMBER(9,0)	No	-	1
CLASS_TYPE	VARCHAR2(15)	Yes	-	-
INSTRUCTOR_NAME	VARCHAR2(15)	No	-	-
CLASS_DIFFICULTY	VARCHAR2(30)	Yes	-	-
FLOOR	NUMBER(3,0)	Yes	-	-
ROOM	VARCHAR2(20)	Yes	-	-

1 - 6

ATTENDS

Column Name	Data Type	Nullable	Default	Primary Key
MEM_ID	VARCHAR2(9)	No	-	1
C_ID	NUMBER(9,0)	No	-	2

1 - 2

MEMBER_PHONE_NUMBER

Column Name	Data Type	Nullable	Default	Primary Key
M_ID	VARCHAR2(9)	No	-	1
PHONE_NUMBER	NUMBER(8,0)	No	-	2

1 - 2

STAFF_PHONE_NUMBER

Column Name	Data Type	Nullable	Default	Primary Key
ST_ID	VARCHAR2(9)	No	-	1
PHONE_NUMBER	NUMBER(8,0)	No	-	2

1 - 2

DEPARTMENT_PHONE_NUMBER

Column Name	Data Type	Nullable	Default	Primary Key
DEPT_ID	CHAR(9)	No	-	1
PHONE_NUMBER	VARCHAR2(15)	No	-	2

1 - 2

VISITOR_PHONE_NUMBER

Column Name	Data Type	Nullable	Default	Primary Key
VISITORNAME	VARCHAR2(20)	No	-	1
PHONE_NUMBER	VARCHAR2(8)	No	-	2
M_ID	VARCHAR2(9)	No	-	3

1 - 3

BRANCH_PHONE_NUMBER

Column Name	Data Type	Nullable	Default	Primary Key
BRANCH_ID	VARCHAR2(9)	No	-	1
PHONE_NUMBER	VARCHAR2(15)	No	-	2

1 - 2

VIII. INSERTION OF DATA:

MEMBER

```
INSERT INTO MEMBER
VALUES('MEM01','AI03','ROBERT',NULL,'NAAME','M','4-8-2','ROBNAAME02@GMAIL.C
OM');
INSERT INTO MEMBER
VALUES('MEM02','AI01','HUDA','M','ARNAOUT','F',DATE'2002-06-23','HUDAARR@GM
AIL.COM');
INSERT INTO MEMBER
VALUES('MEM03','AI04','JULIA','K','KHOURY','M',DATE'2001-09-19','JAYJAY123@
YAHOO.COM');
INSERT INTO MEMBER
VALUES('MEM04','AI04','ABDULLAH',NULL,'TOURBAH','M',DATE'2002-11-09','ABDUL
LAHT@YAHOO.COM');
INSERT INTO MEMBER
VALUES('MEM05','AI02','ALEX',NULL,'DARZI','M',DATE'1999-08-08','ALEXXDARZI@
YAHOO.COM');
INSERT INTO MEMBER
VALUES('MEM06','AI06','KATIA','M','ZAROUI','F',DATE'1988-09-01','SOMEONE123
@GMAIL.COM');
INSERT INTO MEMBER
VALUES('MEM07','AI04','NADINE',NULL,'SHBERO','F',DATE'1990-01-24','NADINE45
67@GMAIL.COM');
INSERT INTO MEMBER
VALUES('MEM08','AI07','FAISAL','E','ALAWI','M',DATE'1999-12-12','FAIS02@HOT
MAIL.COM');
INSERT INTO MEMBER
VALUES('MEM09','AI10','SARAH','A','ATTALAH','F',DATE'2004-09-25','ATALSARAH
@GMAIL.COM');
INSERT INTO MEMBER
VALUES('MEM10','AI09','IBRAHIM','A','HAZIME','M',DATE'2000-08-07','HAZIME_M
@GMAIL.COM');
INSERT INTO MEMBER
VALUES('MEM11','AI05','ROBIL',NULL,'MAALOUF','M',DATE'1987-09-07','ROBMAALO
UF@GMAIL.COM');
INSERT INTO MEMBER
VALUES('MEM12','AI08','GHADI',NULL,'AYOUB','M',DATE'1990-02-03','GHADIGHADI
@YAHOO.COM');
```

MEMBERSHIP

```
INSERT INTO MEMBERSHIP VALUES(1,'Premium','1 year',650.00,'MEM01');
INSERT INTO MEMBERSHIP VALUES(2,'Premium','1 month',60.00,'MEM06');
INSERT INTO MEMBERSHIP VALUES(3,'Standard','1 year',550.00,'MEM07');
INSERT INTO MEMBERSHIP VALUES(4,'Standard','1 month',50.00,'MEM08');
INSERT INTO MEMBERSHIP VALUES(5,'Standard','2 month',100.00,'MEM09');
INSERT INTO MEMBERSHIP VALUES(6,'Premium','2 month',120.00,'MEM02');
INSERT INTO MEMBERSHIP VALUES(7,'Premium','6 month',350.00,'MEM04');
INSERT INTO MEMBERSHIP VALUES(8,'Standard','6 month',300.00,'MEM03');
INSERT INTO MEMBERSHIP VALUES(9,'Premium','2 year',1400.00,'MEM05');
INSERT INTO MEMBERSHIP VALUES(10,'Standard','2 year',1200.00,'MEM10');
```

DEPARTMENT

```
INSERT INTO DEPARTMENT VALUES('D01','Bus
office','NAI','businessoff1@gymnAIsum.lb','coastal plain','2038
3054','Lebanon','Beirut','Hamra','Golden Tower');
INSERT INTO DEPARTMENT VALUES('D02','Bus
office','NAI','businessoff2@gymnAIsum.lb','coastal plain','2034
3054','Lebanon','Byblos','Blat','Red Tower');
INSERT INTO DEPARTMENT VALUES('D03','Bus
office','NAI','businessoff3@gymnAIsum.lb','mountain chain','2039
3054','Lebanon','Zahle','Aintoura','Blue Tower');
INSERT INTO DEPARTMENT VALUES('D04','Bus
office','NAI','businessoff4@gymnAIsum.lb','mountain chain','2037
3054','Lebanon','Tripoli','El Maarad','Emerald Tower');
INSERT INTO DEPARTMENT VALUES('D05','Bus
office','NAI','businessoff5@gymnAIsum.lb','coastal plain','2031
3054','Lebanon','Sidon','Takkeyeddine El Solh','Velvet Tower');
INSERT INTO DEPARTMENT VALUES('D06','Bus
office','NAI','businessoff6@gymnAIsum.lb','mountain chain','2033
3054','Lebanon','ehden','Ejbeaa','Green Tower');
INSERT INTO DEPARTMENT VALUES('D07','Bus
office','NAI','businessoff7@gymnAIsum.lb','coastal plain','2036
3054','USA','New york','Bedford-Stuyvesant','Q Tower');
INSERT INTO DEPARTMENT
VALUES('D08','Gymnaisum','AI','GymAIoff1@gymnAIsum.lb','coastal
plain','2038 3054','Lebanon','Beirut','Hamra','Golden Tower');
INSERT INTO DEPARTMENT
VALUES('D09','Gymnaisum','AI','GymAIoff2@gymnAIsum.lb','coastal
plain','2034 3054','Lebanon','Byblos','Blat','Red Tower');
INSERT INTO DEPARTMENT
VALUES('D10','Gymnaisum','AI','GymAIoff3@gymnAIsum.lb','mountain
chain','2039 3054','Lebanon','Zahle','Aintoura','Blue Tower');
INSERT INTO DEPARTMENT
VALUES('D11','Gymnaisum','AI','GymAIoff4@gymnAIsum.lb','mountain
chain','2037 3054','Lebanon','Tripoli','El Maarad','Emerald Tower');
INSERT INTO DEPARTMENT
VALUES('D12','Gymnaisum','AI','GymAIoff5@gymnAIsum.lb','coastal
plain','2031 3054','Lebanon','Sidon','Takkeyeddine El Solh','Velvet
Tower');
```

```
INSERT INTO DEPARTMENT
VALUES('D13','Gymnaisum','AI','GymAIoff6@gymnAIsum.lb','mountain
chain','2033 3054','Lebanon','ehden','Ejbeaa','Green Tower');
INSERT INTO DEPARTMENT
VALUES('D14','Gymnaisum','AI','GymAIoff7@gymnAIsum.lb','coastal
plain','2036 3054','USA','New york','Bedford-Stuyvesant','Q Tower');
```

BRANCH

```
INSERT INTO BRANCH VALUES('BR101','GymnAIsium',210,
DATE'2010-06-12',TIMESTAMP'2010-06-12 09:00:00',TIMESTAMP'2010-06-23
23:00:00','costal plain','20383054','Lebanon','Beirut','Hamra','Golden
Hour','D01');
INSERT INTO BRANCH VALUES('BR102','GymnAIsium2',306,
DATE'2011-12-26',TIMESTAMP'2011-12-26 09:00:00',TIMESTAMP'2011-12-26
23:00:00','costal plain','20343054','Lebanon','Byblos','Blat','Red
Tower','D02');
INSERT INTO BRANCH VALUES('BR103','GymnAIsium3',110,
DATE'2015-09-08',TIMESTAMP'2015-09-08 09:00:00',TIMESTAMP'2015-09-08
23:00:00','mountain chain','20393054','Lebanon','Zahle','Aintoura','Blue
Tower','D03');
INSERT INTO BRANCH VALUES('BR104','GymnAIsium4',218,
DATE'2017-11-23',TIMESTAMP'2017-11-23 09:00:00',TIMESTAMP'2017-11-23
23:00:00','mountain chain','20373054','Lebanon','Tripoli','El
Maarad','Emerald Tower','D04');
INSERT INTO BRANCH VALUES('BR105','GymnAIsium5',321,
DATE'2017-12-29',TIMESTAMP'2017-12-29 09:00:00',TIMESTAMP'2017-12-29
23:00:00','costal plain','20313054','Lebanon','Sidon','Takkeyedin
Solh','Velvet Tower','D05');
INSERT INTO BRANCH VALUES('BR106','GymnAIsium6',270,
DATE'2018-09-13',TIMESTAMP'2018-09-13 09:00:00',TIMESTAMP'2018-09-13
23:00:00','mountain chain','20333054','Lebanon','ehden','Ejbeaa','Green
Tower','D06');
INSERT INTO BRANCH VALUES('BR107','GymnAIsium7',155,
DATE'2019-12-22',TIMESTAMP'2019-12-22 09:00:00',TIMESTAMP'2019-12-22
23:00:00','costal plain','20363054','USA','New York','BedStuyvesant','Q
Tower','D07');
```

EQUIPMENT

```
INSERT INTO EQUIPMENT VALUES ('000001', 'Dumbbell set','NElectrical',
180,2016,'5', '101','MEM01','D13');
INSERT INTO EQUIPMENT VALUES ('000002',
'Treadmill','Electrical',500.5,2008,'1', '102','MEM07','D14');
INSERT INTO EQUIPMENT VALUES ('000003', 'Training bench','NElectrical'
,100,2010,'1', '103','MEM04','D05');
INSERT INTO EQUIPMENT VALUES ('000004',
'Bicycle','Electrical',250.5,2012,'2', '104','MEM09','D03');
```

```
INSERT INTO EQUIPMENT VALUES ('000005', 'Barbell
Set','NElectrical',600,2017,'4', '105','MEM07','D04');
INSERT INTO EQUIPMENT VALUES ('000006', 'Ellipticals','NElectrical',400,
2019,'3', '106','MEM02','D10');
INSERT INTO EQUIPMENT VALUES ('000007', 'Balance
Trainer','NElectrical',670.5,2016,'1', '107','MEM03','D12');
INSERT INTO EQUIPMENT VALUES ('000008', 'cables machine', 'Electrical', 500
,2016,'6', '108','MEM02','D06');
INSERT INTO EQUIPMENT VALUES ('000009', 'stairs','Electrical',770
,2016,'3', '109','MEM06','D08');
INSERT INTO EQUIPMENT VALUES ('000010', 'yoga ball','NElectrical',120.5
,2016,'3', '110','MEM05','D10');
INSERT INTO EQUIPMENT VALUES ('000011', 'Ketel
weight','NElectrical',440.6,2019,'3', '111','MEM08','D11');
INSERT INTO EQUIPMENT VALUES ('000012', 'Boxing Bag','NElectrical',200
,2018,'2', '112','MEM10','D03');
INSERT INTO EQUIPMENT VALUES ('000013', 'Smith Machine','R',400 ,2016,'3',
'113','MEM03','D07');
INSERT INTO EQUIPMENT VALUES ('000014', 'Leg Press','R',500.9 ,2012,'6',
'114','MEM04','D02');
INSERT INTO EQUIPMENT VALUES ('000015', 'VR TV','R',3999.9 ,2007,'5',
'115','MEM06','D09');
```

AI_TRAINER

```
INSERT INTO AI_TRAINER
VALUES('AI01','MUSK',67,DATE'2019-07-01',TIMESTAMP'2020-06-01
09:00:00',TIMESTAMP'2020-06-01 12:00:00',TIMESTAMP'2020-06-01 11:00:00',1);
INSERT INTO AI_TRAINER
VALUES('AI02','MARYLEIN',79,DATE'2019-09-01',TIMESTAMP'2020-06-01
09:00:00',TIMESTAMP'2020-06-01 12:00:00',TIMESTAMP'2020-06-01 11:00:00',0);
INSERT INTO AI_TRAINER
VALUES('AI03','GYMBOT',86,DATE'2020-01-23',TIMESTAMP'2020-06-01
09:00:00',TIMESTAMP'2020-06-01 12:00:00',TIMESTAMP'2020-06-01 11:00:00',0);
INSERT INTO AI_TRAINER
VALUES('AI04','ARTBOY',99,DATE'2020-05-05',TIMESTAMP'2020-06-01
09:00:00',TIMESTAMP'2020-06-01 12:00:00',TIMESTAMP'2020-06-01 11:00:00',1);
INSERT INTO AI_TRAINER
VALUES('AI05','MUSKETEER',10,DATE'2018-11-11',TIMESTAMP'2020-06-01
09:00:00',TIMESTAMP'2020-06-01 12:00:00',TIMESTAMP'2020-06-01 11:00:00',1);
INSERT INTO AI_TRAINER
VALUES('AI06','HOGWART',86,DATE'2019-07-05',TIMESTAMP'2020-06-01
09:00:00',TIMESTAMP'2020-06-01 12:00:00',TIMESTAMP'2020-06-01 11:00:00',1);
INSERT INTO AI_TRAINER
VALUES('AI07','ANGEL',98,DATE'2021-04-11',TIMESTAMP'2021-08-04
09:00:00',TIMESTAMP'2021-08-04 12:00:00',TIMESTAMP'2021-08-04 11:00:00',0);
INSERT INTO AI_TRAINER
VALUES('AI08','RAWR',12,DATE'2020-08-11',TIMESTAMP'2021-08-04
09:00:00',TIMESTAMP'2021-08-04 12:00:00',TIMESTAMP'2021-08-04 11:00:00',1);
```

```
INSERT INTO AI_TRAINER
VALUES('AI09','NOPELO',91,DATE'2019-04-19',TIMESTAMP'2021-08-04
09:00:00',TIMESTAMP'2021-08-04 12:00:00',TIMESTAMP'2021-08-04 11:00:00',1);
INSERT INTO AI_TRAINER
VALUES('AI10','PINNOCHIO',18,DATE'2020-04-26',TIMESTAMP'2021-08-04
09:00:00',TIMESTAMP'2021-08-04 12:00:00',TIMESTAMP'2021-08-04 11:00:00',0);
```

VISITOR

```
INSERT INTO VISITOR VALUES('MEM07' , 'Andrew Chemaly' , 'AC420@gmail.com',
'01-02-2020', 'vrDep');
INSERT INTO VISITOR VALUES('MEM02' , 'Omar El Baba' , 'OEB.HA@gmail.com',
'02-03-2020', 'Cardio bldg');
INSERT INTO VISITOR VALUES('MEM09' , 'Robert Naame', 'Fillory@gmail.com',
'05-05-2021', 'Medical Bldg');
INSERT INTO VISITOR VALUES('MEM01' , 'Chad Rover' , 'Chad18@hotmail.com',
'08-09-2019', 'Main Bldg');
INSERT INTO VISITOR VALUES('MEM01' , 'Bonny Bee' , 'BeeWa234@gmail.com',
'01-13-2022', 'Heavyweights Bldg');
INSERT INTO VISITOR VALUES('MEM03' , 'Caitlyn Yennefer' ,
'Caitt249@yahoo.com', '12-12-2012', 'Cardio Bldg');
INSERT INTO VISITOR VALUES('MEM09' , 'Fiora tawk' , 'FioraOP@hotmail.com',
'10-10-2010','Heavyweights Bldg');
INSERT INTO VISITOR VALUES('MEM02' , 'Jinx Saly' , 'Jinx34@gmail.com',
'09-15-2015', 'Medical Bldg'');
INSERT INTO VISITOR VALUES('MEM06' , 'Teemo Smith' , 'CaptainT@gmail.com',
'10-14-2014', 'Cardiol Bldg');
INSERT INTO VISITOR VALUES('MEM04' , 'Wukong Toppie' ,
'WukongNRS@gmail.com', '01-12-2022', 'Main Bldg');
```

STAFF

```
INSERT INTO STAFF VALUES ('ST01','BS in cs',DATE'1985-12-15'
,120000.50,'account creator',DATE'2010-09-02',DATE'2021-12-11',
'angel.pierre@lau.edu.lb','O-', 'M', 'Angel','E','Pierre','D01');
INSERT INTO STAFF VALUES ('ST02','BE in cce',DATE'1978-03-17'
,250000.00,'personal manager',DATE'2014-10-08',DATE '2014-11-08',
'alice.wonderland@lau.edu.lb','B+', 'F', 'Alice','E','Wonderland','D02');
INSERT INTO STAFF VALUES ('ST03','BA in Arch',DATE'1990-07-02'
,785870.50,'gym designer',DATE'2009-03-12',DATE'2022-11-11',
'john.estay@lau.edu.lb', 'O+','M', 'John','N','Estay','D03');
INSERT INTO STAFF VALUES ('ST04','BS in Eco',DATE'1968-01-01'
,890000.00,'account fixer',DATE'2011-05-08',DATE'2021-03-19',
'emily.lapelle@lau.edu.lb', 'A+', 'F', 'Emily','J','Lapelle','D04');
INSERT INTO STAFF VALUES ('ST05','BS in Bus',DATE'1992-01-10'
,520000.00,'personal trainer',DATE'2016-09-28',DATE'2019-12-01',
'xavier.cesstoy@lau.edu.lb', 'O-', 'M', 'Xavier','L','Cesstoy','D11');
INSERT INTO STAFF VALUES ('ST06','BS in Info',DATE'1989-01-13'
,327000.50,'security guard',DATE'1999-06-01',DATE'2020-02-12',
'jad.sinc@lau.edu.lb', 'A-', 'M', 'Jad','W','Sinc','D12');
```

```
INSERT INTO STAFF VALUES ('ST07','BS in Bio',DATE'1988-05-15'
,700000.00,'medical doctor',DATE'2010-01-27',DATE '2010-10-27',
'stacy.watson@lau.edu.lb', 'B+', 'F', 'Stacy','K','Watson','D13');
```

MEDICAL_RECORD

```
INSERT INTO MEDICAL_RECORD VALUES ('202000325',42,'AB-','arm
injury','meat',172.50,61.23,'MEM07');
INSERT INTO MEDICAL_RECORD VALUES ('202204102',55,'AB+',NULL, NULL,
166.00,59.84,'MEM04');
INSERT INTO MEDICAL_RECORD VALUES ('202102689',23,'B-','finger
injury','fish',179.50,67.00,'MEM02');
INSERT INTO MEDICAL_RECORD VALUES ('201900210',29,'A+','bleeding nose',
'soft drinks',152.53,53.21,'MEM06');
INSERT INTO MEDICAL_RECORD VALUES
('202003149',11,'A-','Headache',NULL,170.70,72.84,'MEM08');
INSERT INTO MEDICAL_RECORD VALUES ('202101480',68,'B+',NULL, 'paracetamol',
158.30,87.31,'MEM01');
INSERT INTO MEDICAL_RECORD VALUES ('201903420',54,'O+','asthma',
'peanut',192.30,75.40,'MEM10');
INSERT INTO MEDICAL_RECORD VALUES ('202101292',24,'O-','leg
injury','lactose' ,171.30,64.27,'MEM05');
INSERT INTO MEDICAL_RECORD VALUES ('202201783',33,'AB+',NULL,NULL,
182.50,68.38,'MEM03');
INSERT INTO MEDICAL_RECORD VALUES ('202000025',38,'AB-',NULL ,'eggs',
185.00,71.82,'MEM09');
```

EMERGENCY_CONTACT

```
INSERT INTO EMERGENCY_CONTACT VALUES('MEM01','MALIK
KANAAN','BROTHER','78654238');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM02','NANCY
NEHME','SISTER','71235536');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM03','HAIFA
ABDELWAHAB','SPOUSE','89725422');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM04','ELON
MUSKTEER','FIANCE','78645343');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM05','SABAH
SOUBRA','MOTHER','81563423');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM06','WARDA
MAGHRIBIYA','BROTHER','77226625');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM07','RIWA
SOUBRA','BROTHER','84522273');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM08','JAD
ABOBRAHIM','COUSIN','78873191');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM09','ANTOUN
ATALLAH','FATHER','72426778');
INSERT INTO EMERGENCY_CONTACT VALUES('MEM10','MUSTAFA
HAZIME','FATHER','87255522');
```

CLASS

```
INSERT INTO CLASS VALUES(1,'kids yoga','Yakov
Smirnoff','Beginner',02,'0203');
INSERT INTO CLASS VALUES(2,'yoga','Yakov
Smirnoff','Intermediate',05,'0503');
```

```
INSERT INTO CLASS VALUES(3,'pro-yoga','Yakov
Smirnoff','Professional',05,'0506');
INSERT INTO CLASS VALUES(4,'Boxing','Jerry
Ferrara','Intermediate',03,'0303');
INSERT INTO CLASS VALUES(5,'pro-Boxing','Jerry
Ferrara','Professional',05,'0508');
INSERT INTO CLASS VALUES(6,'VR-parkour','Vera
Wang','Professional',010,'01001');
INSERT INTO CLASS VALUES(7,'VR-skiing','Jodie
Sweetin','Professional',010,'01002');
INSERT INTO CLASS VALUES(8,'cross-fit','Bart
Starr','Intermediate',05,'0502');
INSERT INTO CLASS VALUES(9,'pro-cross-fit','Bart
Starr','Professional',010,'01003');
INSERT INTO CLASS VALUES(10,'pilates','Judy
Garland','Professional',010,'01010');
```

ATTENDS

```
INSERT INTO ATTENDS VALUES(2,'MEM01');
INSERT INTO ATTENDS VALUES(6,'MEM01');
INSERT INTO ATTENDS VALUES(7,'MEM02');
INSERT INTO ATTENDS VALUES(8,'MEM03');
INSERT INTO ATTENDS VALUES(1,'MEM02');
INSERT INTO ATTENDS VALUES(1,'MEM04');
INSERT INTO ATTENDS VALUES(4,'MEM05');
INSERT INTO ATTENDS VALUES(5,'MEM06');
INSERT INTO ATTENDS VALUES(3,'MEM07');
INSERT INTO ATTENDS VALUES(9,'MEM08');
INSERT INTO ATTENDS VALUES(10,'MEM09');
INSERT INTO ATTENDS VALUES(10,'MEM10');
INSERT INTO ATTENDS VALUES(10,'MEM06');
INSERT INTO ATTENDS VALUES(2,'MEM08');
```

MEMBER_Phone_Number

```
INSERT INTO MEMBER_Phone_Number VALUES ('MEM01','70130256');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM02','70110531');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM03','70130256');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM04','72105497');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM05','03133442');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM06','77136596');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM07','70300201');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM08','76551140');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM09','79130678');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM10','78128401');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM11','79140767');
INSERT INTO MEMBER_Phone_Number VALUES ('MEM12','70225501');
```

STAFF_Phone_Number

```
INSERT INTO STAFF_Phone_Number VALUES ('ST01','70131256');
INSERT INTO STAFF_Phone_Number VALUES ('ST02','76234257');
INSERT INTO STAFF_Phone_Number VALUES ('ST03','78167256');
INSERT INTO STAFF_Phone_Number VALUES ('ST04','71125257');
INSERT INTO STAFF_Phone_Number VALUES ('ST05','70211211');
```

```
INSERT INTO STAFF_Phone_Number VALUES ('ST06','78141414');
INSERT INTO STAFF_Phone_Number VALUES ('ST07','70771776');
```

DEPARTMENT_Phone_Number

```
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D01','70131256');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D02','76234257');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D03','78167256');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D04','71125257');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D05','70211211');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D06','78141414');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D07','70771776');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D08','70778878');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D09','70971999');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D10','70111776');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D11','70777777');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D12','70661666');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D13','70234576');
INSERT INTO DEPARTMENT_Phone_Number VALUES ('D14','70991796');
```

VISITORS_PHONE_NUMBER

```
INSERT INTO VISITOR_PHONE_NUMBER VALUES( 'Andrew
Chemaly','70003994','MEM07');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Omar El Baba','80214794','MEM02');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Robert Naame','70207867','MEM09');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Chad Rover','70204368','MEM01');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Bonny Bee','70190359','MEM01');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Caitlyn
Yennefer','82009987','MEM03');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Fiora tawk','80200454','MEM09');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Jinx Saly','80200569','MEM02');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Teemo Smith','80200378','MEM06');
INSERT INTO VISITOR_PHONE_NUMBER VALUES('Wukong
Toppie','70220457','MEM04');
```

BRANCH_Phone_Number

```
INSERT INTO BRANCH_Phone_Number VALUES ('BR101','70131876');
INSERT INTO BRANCH_Phone_Number VALUES ('BR101','01675422');
INSERT INTO BRANCH_Phone_Number VALUES ('BR102','79876542');
INSERT INTO BRANCH_Phone_Number VALUES ('BR102','01234678');
INSERT INTO BRANCH_Phone_Number VALUES ('BR103','78976512');
INSERT INTO BRANCH_Phone_Number VALUES ('BR103','01765489');
INSERT INTO BRANCH_Phone_Number VALUES ('BR104','75239861');
INSERT INTO BRANCH_Phone_Number VALUES ('BR104','01456866');
INSERT INTO BRANCH_Phone_Number VALUES ('BR105','78873191');
INSERT INTO BRANCH_Phone_Number VALUES ('BR105','01864122');
INSERT INTO BRANCH_Phone_Number VALUES ('BR106','79965414');
INSERT INTO BRANCH_Phone_Number VALUES ('BR106','01765423');
```

IX.DATA INSERTED TABLE VIEW:

MEMBER

M_ID	AI_ID	FIRSTNAME	MI	LASTNAME	GENDER	DATE_OF_BIRTH	EMAIL
MEM01	AI03	ROBERT	-	NAAME	M	04/08/0002	ROBNAAME02@GMAIL.COM
MEM02	AI01	HUDA	M	ARNAOUT	F	06/23/2002	HUDAARR@GMAIL.COM
MEM03	AI04	JULIA	K	KHOURY	M	09/19/2001	JAYJAY123@YAHOO.COM
MEM04	AI04	ABDULLAH	-	TOURBAH	M	11/09/2002	ABDULLAHT@YAHOO.COM
MEM05	AI02	ALEX	-	DARZI	M	08/08/1999	ALEXXDARZI@YAHOO.COM
MEM06	AI06	KATIA	M	ZAROUI	F	09/01/1988	SOMEONE123@GMAIL.COM
MEM07	AI04	NADINE	-	SHBERO	F	01/24/1990	NADINE4567@GMAIL.COM
MEM08	AI07	FAISAL	E	ALAWI	M	12/12/1999	FAIS02@HOTMAIL.COM
MEM09	AI10	SARAH	A	ATTALAH	F	09/25/2004	ATALSARAH@GMAIL.COM
MEM10	AI09	IBRAHIM	A	HAZIME	M	08/07/2000	HAZIME_M@GMAIL.COM
MEM11	AI05	ROBIL	-	MAALOUF	M	09/07/1987	ROBMAALOUF@GMAIL.COM
MEM12	AI08	GHADI	-	AYOUB	M	02/03/1990	GHADIGHADI@YAHOO.COM

row(s) 1 - 12 of 12

MEMBERSHIP

MEMBERSHIP_NUMBER	MEMBERSHIP_NAME	MEMBERSHIP_TYPE	MEMBERSHIP_PRICE	M_ID
1	Premium	1 year	650	MEM01
2	Premium	1 month	60	MEM06
3	Standard	1 year	550	MEM07
4	Standard	1 month	50	MEM08
5	Standard	2 month	100	MEM09
6	Premium	2 month	120	MEM02
7	Premium	6 month	350	MEM04
8	Standard	6 month	300	MEM03
9	Premium	2 year	1400	MEM05
10	Standard	2 year	1200	MEM10

DEPARTMENT

D_ID	NAME	D_TYPE	EMAIL	REGION	ZIP_CODE	COUNTRY	CITY	STREET	BUILDING
D08	Gymnaisum	AI	GymAIoff1@gymnAIsum.lb	coastal plain	2038 3054	Lebanon	Beirut	Hamra	Golden Tower
D09	Gymnaisum	AI	GymAIoff2@gymnAIsum.lb	coastal plain	2034 3054	Lebanon	Byblos	Blat	Red Tower
D10	Gymnaisum	AI	GymAIoff3@gymnAIsum.lb	mountain chain	2039 3054	Lebanon	Zahle	Aintoura	Blue Tower
D11	Gymnaisum	AI	GymAIoff4@gymnAIsum.lb	mountain chain	2037 3054	Lebanon	Tripoli	El Maarad	Emerald Tower
D12	Gymnaisum	AI	GymAIoff5@gymnAIsum.lb	coastal plain	2031 3054	Lebanon	Sidon	Takkeyeddine El Solh	Velvet Tower
D13	Gymnaisum	AI	GymAIoff6@gymnAIsum.lb	mountain chain	2033 3054	Lebanon	ehden	Ejbeaa	Green Tower
D14	Gymnaisum	AI	GymAIoff7@gymnAIsum.lb	coastal plain	2036 3054	USA	New york	Bedford-Stuyvesant	Q Tower
D01	Bus office	NAI	businessoff1@gymnAIsum.lb	coastal plain	2038 3054	Lebanon	Beirut	Hamra	Golden Tower
D02	Bus office	NAI	businessoff2@gymnAIsum.lb	coastal plain	2034 3054	Lebanon	Byblos	Blat	Red Tower
D03	Bus office	NAI	businessoff3@gymnAIsum.lb	mountain chain	2039 3054	Lebanon	Zahle	Aintoura	Blue Tower
D04	Bus office	NAI	businessoff4@gymnAIsum.lb	mountain chain	2037 3054	Lebanon	Tripoli	El Maarad	Emerald Tower
D05	Bus office	NAI	businessoff5@gymnAIsum.lb	coastal plain	2031 3054	Lebanon	Sidon	Takkeyeddine El Solh	Velvet Tower
D06	Bus office	NAI	businessoff6@gymnAIsum.lb	mountain chain	2033 3054	Lebanon	ehden	Ejbeaa	Green Tower
D07	Bus office	NAI	businessoff7@gymnAIsum.lb	coastal plain	2036 3054	USA	New york	Bedford-Stuyvesant	Q Tower

BRANCH

BR_ID	NAME	NUMBER_OF_STAFF	DATE_OF_OPENING	OPEN_HOUR	CLOSE_HOUR	REGION	ZIP_CODE	COUNTRY	CITY	STREET	BUILDING	D_ID
BR101	GymnAIsium	210	06/12/2010	12-JUN-10 09.00.00.000000 AM	23-JUN-10 11.00.00.000000 PM	costal plain	20383054	Lebanon	Beirut	Hamra	Golden Hour	D01
BR102	GymnAIsium2	306	12/26/2011	26-DEC-11 09.00.00.000000 AM	26-DEC-11 11.00.00.000000 PM	costal plain	20343054	Lebanon	Byblos	Blat	Red Tower	D02
BR103	GymnAIsium3	110	09/08/2015	08-SEP-15 09.00.00.000000 AM	08-SEP-15 11.00.00.000000 PM	mountain chain	20393054	Lebanon	Zahle	Aintoura	Blue Tower	D03
BR104	GymnAIsium4	218	11/23/2017	23-NOV-17 09.00.00.000000 AM	23-NOV-17 11.00.00.000000 PM	mountain chain	20373054	Lebanon	Tripoli	El Maarad	Emerald Tower	D04
BR106	GymnAIsium6	270	09/13/2018	13-SEP-18 09.00.00.000000 AM	13-SEP-18 11.00.00.000000 PM	mountain chain	20333054	Lebanon	ehden	Ejbeaa	Green Tower	D06
BR107	GymnAIsium7	155	12/22/2019	22-DEC-19 09.00.00.000000 AM	22-DEC-19 11.00.00.000000 PM	costal plain	20363054	USA	New York	BedStuyvesant	Q Tower	D07
BR105	GymnAIsium5	321	12/29/2017	29-DEC-17 09.00.00.000000 AM	29-DEC-17 11.00.00.000000 PM	costal plain	20313054	Lebanon	Sidon	Takkeyedin Solh	Velvet Tower	D05

EQUIPMENT

EQUIPMENT_SNUMBER	EQUIPMENT_NAME	EQUIPMENT_TYPE	EQUIPMENT_PRICE	YOM	FLOOR	ROOM	M_ID	D_ID
1	Dumbbell set	NElectrical	180	2016	5	101	MEM01	D13
2	Treadmill	Electrical	500.5	2008	1	102	MEM07	D14
3	Training bench	NElectrical	100	2010	1	103	MEM04	D05
4	Bicycle	Electrical	250.5	2012	2	104	MEM09	D03
5	Barbell Set	NElectrical	600	2017	4	105	MEM07	D04
6	Ellipticals	NElectrical	400	2019	3	106	MEM02	D10
8	cables machine	Electrical	500	2016	6	108	MEM02	D06
9	stairs	Electrical	770	2016	3	109	MEM06	D08
10	yoga ball	NElectrical	120.5	2016	3	110	MEM05	D10
11	Ketel weight	NElectrical	440.6	2019	3	111	MEM08	D11
12	Boxing Bag	NElectrical	200	2018	2	112	MEM10	D03
13	Smith Machine	R	400	2016	3	113	MEM03	D07
14	Leg Press	R	500.9	2012	6	114	MEM04	D02
15	VR TV	R	3999.9	2007	5	115	MEM06	D09
7	Balance Trainer	NElectrical	670.5	2016	1	107	MEM03	D12

row(s) 1 - 15 of 15

AI_TRAINER

AI_ID	PSEUDO_NAME	BATTERY_LIFE	DATE_OF_MAN	START_SHIFT	MAINTENANCE_TIME	END_SHIFT	SPECS
AI01	MUSK	67	07/01/2019	06/01/2020	06/01/2020	06/01/2020	1
AI02	MARYLEIN	79	09/01/2019	06/01/2020	06/01/2020	06/01/2020	0
AI03	GYMBOT	86	01/23/2020	06/01/2020	06/01/2020	06/01/2020	0
AI04	ARTBOY	99	05/05/2020	06/01/2020	06/01/2020	06/01/2020	1
AI05	MUSKETEER	10	11/11/2018	06/01/2020	06/01/2020	06/01/2020	1
AI06	HOGWART	86	07/05/2019	06/01/2020	06/01/2020	06/01/2020	1
AI07	ANGEL	98	04/11/2021	08/04/2021	08/04/2021	08/04/2021	0
AI08	RAWR	12	08/11/2020	08/04/2021	08/04/2021	08/04/2021	1
AI09	NOPELO	91	04/19/2019	08/04/2021	08/04/2021	08/04/2021	1
AI10	PINNOCHIO	18	04/26/2020	08/04/2021	08/04/2021	08/04/2021	0

row(s) 1 - 10 of 10

VISITOR

M_ID	VISITORNAME	EMAIL	DATE_OF_VISIT	LOCATION
MEM07	Andrew Chemaly	AC420@gmail.com	01-02-2020	vrDep
MEM02	Omar El Baba	OEB.HA@gmail.com	02-03-2020	Cardio bldg
MEM09	Robert Naame	Fillory@gmail.com	05-05-2021	Medical Bldg
MEM01	Chad Rover	Chad18@hotmail.com	08-09-2019	Main Bldg
MEM01	Bonny Bee	BeeWa234@gmail.com	01-13-2022	Heavyweights Bldg
MEM03	Caitlyn Yennefer	Caitt249@yahoo.com	12-12-2012	Cardio Bldg
MEM09	Fiora tawk	FioraOP@hotmail.com	10-10-2010	Heavyweights Bldg
MEM02	Jinx Saly	Jinx34@gmail.com	09-15-2015	Medical Bldg'
MEM06	Teemo Smith	CaptainT@gmail.com	10-14-2014	Cardiol Bldg
MEM04	Wukong Toppie	WukongNRS@gmail.com	01-12-2022	Main Bldg

row(s) 1 - 10 of 10

STAFF

ST_ID	CERT_LEVEL	DATE_OF_BIRTH	SALARY	JOB_TYPE	START_DATE	END_DATE	EMAIL	BLOOD_TYPE	GENDER	FIRSTNAME	MI	LASTNAME	D_ID
ST03	BA in Arch	07/02/1990	785870.5	gym designer	03/12/2009	11/11/2022	john.estay@lau.edu.lb	O+	M	John	N	Estay	D03
ST01	BS in cs	12/15/1985	120000.5	account creator	09/02/2010	12/11/2021	angel.pierre@lau.edu.lb	O-	M	Angel	E	Pierre	D01
ST02	BE in cce	03/17/1978	250000	personal manager	10/08/2014	11/08/2014	alice.wonderland@lau.edu.lb	B+	F	Alice	E	Wonderland	D02
ST04	BS in Eco	01/01/1968	890000	account fixer	05/08/2011	03/19/2021	emily.lapelle@lau.edu.lb	A+	F	Emily	J	Lapelle	D04
ST05	BS in Bus	01/10/1992	520000	personal trainer	09/28/2016	12/01/2019	xavier.cesstoy@lau.edu.lb	O-	M	Xavier	L	Cesstoy	D11
ST06	BS in Info	01/13/1989	327000.5	security guard	06/01/1999	02/12/2020	jad.sinc@lau.edu.lb	A-	M	Jad	W	Sinc	D12
ST07	BS in Bio	05/15/1988	700000	medical doctor	01/27/2010	10/27/2010	stacy.watson@lau.edu.lb	B+	F	Stacy	K	Watson	D13

MEDICAL_RECORD

MED_ID	BMI	BLOOD_TYPE	PREVIOUS_INJURY	FOOD_RESTRICTION	HEIGHT	WEIGHT	M_ID
202204102	55	AB+	-	-	166	59.84	MEM04
202102689	23	B-	finger injury	fish	179.5	67	MEM02
201900210	29	A+	bleeding nose	soft drinks	152.53	53.21	MEM06
202003149	11	A-	Headache	-	170.7	72.84	MEM08
202101480	68	B+	-	paracetamol	158.3	87.31	MEM01
201903420	54	O+	asthma	peanut	192.3	75.4	MEM10
202101292	24	O-	leg injury	lactose	171.3	64.27	MEM05
202201783	33	AB+	-	-	182.5	68.38	MEM03
202000025	38	AB-	-	eggs	185	71.82	MEM09
202000325	42	AB-	arm injury	meat	172.5	61.23	MEM07

EMERGENCY_CONTACT

M_ID	RELATIONSHIP	PHONE_NUMBER	NAME
MEM01	MALIK KANAAN	BROTHER	78654238
MEM02	NANCY NEHME	SISTER	71235536
MEM03	HAIFA ABDELWAHAB	SPOUSE	89725422
MEM04	ELON MUSKTEER	FIANCE	78645343
MEM05	SABAH SOUBRA	MOTHER	81563423
MEM06	WARDA MAGHRIBIYA	BROTHER	77226625
MEM07	RIWA SOUBRA	BROTHER	84522273
MEM08	JAD ABOBRAHIM	COUSIN	78873191
MEM09	ANTOUN ATALLAH	FATHER	72426778
MEM10	MUSTAFA HAZIME	FATHER	87255522

CLASS

CLASS_ID	CLASS_TYPE	INSTRUCTOR_NAME	CLASS_DIFFICULTY	FLOOR	ROOM
1	kids yoga	Yakov Smirnoff	Beginner	2	0203
2	yoga	Yakov Smirnoff	Intermediate	5	0503
3	pro-yoga	Yakov Smirnoff	Professional	5	0506
4	Boxing	Jerry Ferrara	Intermediate	3	0303
5	pro-Boxing	Jerry Ferrara	Professional	5	0508
6	VR-parkour	Vera Wang	Professional	10	01001
7	VR-skiing	Jodie Sweetin	Professional	10	01002
8	cross-fit	Bart Starr	Intermediate	5	0502
9	pro-cross-fit	Bart Starr	Professional	10	01003
10	pilates	Judy Garland	Professional	10	01010

ATTENDS

C_ID	MEM_ID
2	MEM01
6	MEM01
1	MEM02
7	MEM02
8	MEM03
1	MEM04
4	MEM05
5	MEM06
10	MEM06
3	MEM07
2	MEM08
9	MEM08
10	MEM09
10	MEM10

MEMBER_PHONE_NUMBER

M_ID	PHONE_NUMBER
MEM01	70130256
MEM02	70110531
MEM03	70130256
MEM04	72105497
MEM05	3133442
MEM06	77136596
MEM07	70300201
MEM08	76551140
MEM09	79130678
MEM10	78128401
MEM11	79140767
MEM12	70225501

STAFF_PHONE_NUMBER

ST_ID	PHONE_NUMBER
ST01	70131256
ST02	76234257
ST03	78167256
ST04	71125257
ST05	70211211
ST06	78141414
ST07	70771776

DEPARTMENT_PHONE_NUMBER

DEPT_ID	PHONE_NUMBER
D01	70131256
D02	76234257
D03	78167256
D04	71125257
D05	70211211
D06	78141414
D07	70771776
D08	70778878
D09	70971999
D10	70111776
D11	70777777
D12	70661666
D13	70234576
D14	70991796

VISITOR_PHONE_NUMBER

VISITORNAME	PHONE_NUMBER	M_ID
Andrew Chemaly	70003994	MEM07
Bonny Bee	70190359	MEM01
Caitlyn Yennefer	82009987	MEM03
Chad Rover	70204368	MEM01
Fiora tawk	80200454	MEM09
Jinx Saly	80200569	MEM02
Omar El Baba	80214794	MEM02
Robert Naame	70207867	MEM09
Teemo Smith	80200378	MEM06
Wukong Toppie	70220457	MEM04

BRANCH_PHONE_NUMBER

BRANCH_ID	PHONE_NUMBER
BR101	01675422
BR101	70131876
BR102	01234678
BR102	79876542
BR103	01765489
BR103	78976512
BR104	01456866
BR104	75239861
BR105	01864122
BR105	78873191
BR106	01765423
BR106	79965414

X.QUERIES:

In this section, we present simple and complex real life scenarios that can be applied to our database.

1)A staff member wants to email all the members with premium membership to send a christmas gift

```
SELECT M.Membership_Name, EMAIL

FROM MEMBERSHIP M, MEMBER MEM

WHERE M.M_ID=MEM.M_ID AND Membership_Name ='Premium' ;
```

OUTPUT

MEMBERSHIP_NAME	EMAIL
Premium	ROBNAAME02@GMAIL.COM
Premium	SOMEONE123@GMAIL.COM
Premium	HUDAARR@GMAIL.COM
Premium	ABDULLAHT@YAHOO.COM
Premium	ALEXXDARZI@YAHOO.COM

2)A vaccination campaign for the staff asked to view most common blood type among staff

```
SELECT Blood_Type, COUNT(*)

FROM STAFF

GROUP BY Blood_Type

HAVING COUNT(*) >= ALL (SELECT COUNT(*)

                        FROM STAFF

                       GROUP BY Blood_Type);
```

OUTPUT

BLOOD_TYPE	COUNT(*)
B+	2
O-	2

3)A robbery occurred on 02-03-2020 at GymnAIsium, the manager wants to know who visited the club on that day and who that person is associated with

```
SELECT V.M_ID, VISITORNAME,MEMBER.FIRSTNAME,MEMBER.MI,MEMBER.LASTNAME

FROM VISITOR V,MEMBER

WHERE V.M_ID=MEMBER.M_ID AND DATE_OF_VISIT IN DATE'2020-02-03';
```

OUTPUT

M_ID	VISITORNAME	FIRSTNAME	MI	LASTNAME
MEM02	Omar El Baba	HUDA	M	ARNAOUT

4)The dietitian at GymnAIsium asked for a new diet program for males with with weight<70

```
SELECT mr.MED_ID, m.FIRSTName,m.MI,m.LASTNAME,m.gender

FROM MEMBER m, MEDICAL_RECORD mr

WHERE m.M_ID=mr.M_ID and mr.weight<70 and m.Gender='M';
```

OUTPUT

MED_ID	FIRSTNAME	MI	LASTNAME	GENDER
202204102	ABDULLAH	-	TOURBAH	M
202101292	ALEX	-	DARZI	M

5)The manager of thinking of buying new equipment, he wants to check The Number of rooms that the equipment are in along with the average price of the equipment

```
SELECT COUNT(DISTINCT(ROOM)) AS "TOTAL NO. OF ROOMS", AVG(EQUIPMENT_PRICE)

FROM EQUIPMENT;
```

OUTPUT

TOTAL NO. OF ROOMS	AVG(EQUIPMENT_PRICE)
15	642.22666666666666666666666666666666667

6)We need to know the id of all patients who are in good shape and fit (a fit person is a person whose BMI is between 18.5 and 25). Retrieve the IDS,FirstName, MI, and LastName of all patients whose BMI is between 18.5 and 25.

```
SELECT M.M_ID,M.FirstName,M.MI,M.LASTNAME

FROM MEMBER M, MEDICAL_RECORD MR

WHERE M.M_ID=MR.M_ID AND

(MR.Weight/((MR.Height/100)*(MR.Height/100)))>=18.5 AND

(MR.Weight/((MR.Height/100)*(MR.Height/100)))<=25;
```

OUTPUT

M_ID	FIRSTNAME	MI	LASTNAME
MEM07	NADINE	-	SHBERO
MEM04	ABDULLAH	-	TOURBAH
MEM02	HUDA	M	ARNAOUT
MEM06	KATIA	M	ZAROUI
MEM08	FAISAL	E	ALAWI
MEM10	IBRAHIM	A	HAZIME
MEM05	ALEX	-	DARZI
MEM03	JULIA	K	KHOURY
MEM09	SARAH	A	ATTALAH

7)We want to retrieve First names of members that attend most classes to Feature them on our website

```
SELECT M.FirstName

FROM MEMBER M, ATTENDS A

WHERE M.M_ID = A.MEM_ID

GROUP BY M.FirstName

HAVING COUNT(A.C_ID) =

    (SELECT MAX(COUNT(A.C_ID))

     FROM ATTENDS

     GROUP BY MEM_ID);
```

OUTPUT

FIRSTNAME
ROBERT
HUDA
KATIA
FAISAL

8)The Trainer could not unfortunately make it to class 1 today, she wants to retrieve member emails to let them know about it.

```
SELECT M.FIRSTName,M.email
```

```
FROM MEMBER M,ATTENDS A

WHERE M.M_ID = A.MEM_ID and A.C_ID = 1;
```

OUTPUT

FIRSTNAME	EMAIL
HUDA	HUDAARR@GMAIL.COM
ABDULLAH	ABDULLAHT@YAHOO.COM

9) A member left his phone number on a paper for the receptionist, she was drinking her coffee and it spilled all over the paper, half of the number is only seen. She does not know the member's name so tries to search his phone number in the system's database

```
SELECT FirstName, LastName, Phone_Number

FROM MEMBER M

INNER JOIN MEMBER_Phone_Number P ON M.M_ID = P.M_ID

WHERE P.Phone_Number LIKE '3133%';
```

OUTPUT

FIRSTNAME	LASTNAME	PHONE_NUMBER
ALEX	DARZI	3133442

10)The Supervisor is planning on surprising every birthday in september. Retrieve all members whose birthdays are in september

```
SELECT M_ID,DATE_OF_BIRTH

FROM MEMBER

WHERE to_char(MEMBER.DATE_OF_BIRTH , 'MM')= '09';
```

OUTPUT

M_ID	DATE_OF_BIRTH
MEM03	09/19/2001
MEM06	09/01/1988
MEM11	09/07/1987
MEM09	09/25/2004

11)Query to get all members and their AI Trainers

```
SELECT s.FirstName, s.LastName, s.Email, a.AI_ID

FROM MEMBER s

INNER JOIN AI_TRAINER a

ON s.AI_ID = a.AI_ID
```

FIRSTNAME	LASTNAME	EMAIL	AI_ID
ROBERT	NAAME	ROBNAAME02@GMAIL.COM	AI03
JULIA	KHOURY	JAYJAY123@YAHOO.COM	AI04
ALEX	DARZI	ALEXXDARZI@YAHOO.COM	AI02
KATIA	ZAROUI	SOMEONE123@GMAIL.COM	AI06
IBRAHIM	HAZIME	HAZIME_M@GMAIL.COM	AI09
ROBIL	MAALOUF	ROBMAALOUF@GMAIL.COM	AI05
HUDA	ARNAOUT	HUDAARR@GMAIL.COM	AI01
NADINE	SHBERO	NADINE4567@GMAIL.COM	AI04
FAISAL	ALAWI	FAIS02@HOTMAIL.COM	AI07
GHADI	AYOUB	GHADIGHADI@YAHOO.COM	AI08

12) Selects Member First Name , Last Name and Phone number of members trained by AI trainer 'MUSK'

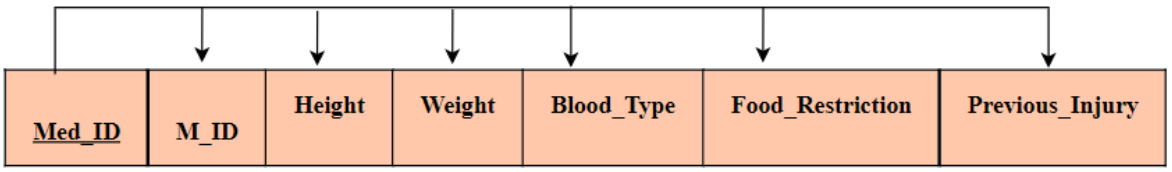
```
SELECT Member.FirstName, Member.LastName, Member_Phone_Number.Phone_Number

FROM Member

JOIN Member_Phone_Number ON Member.M_ID = Member_Phone_Number.M_ID
```

```
WHERE Member.AI_ID IN

(SELECT AI_Trainer.AI_ID

FROM AI_Trainer

WHERE AI_Trainer.Pseudo_Name = 'MUSK');
```

FIRSTNAME	LASTNAME	PHONE_NUMBER
HUDA	ARNAOUT	70110531

NORMALIZATION:

1. MEDICAL_RECORD



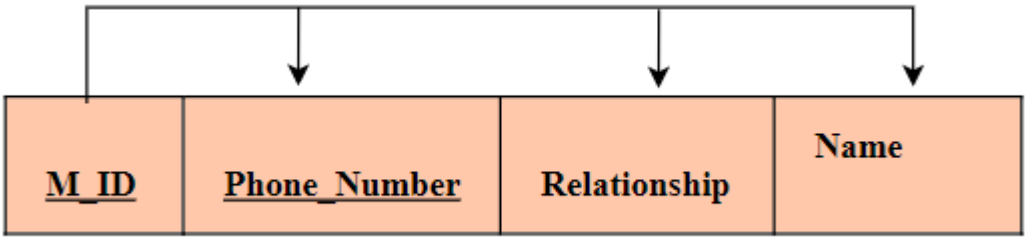
A. All the conditions of the first normal form (1NF) are satisfied within the MEDICAL_RECORD relational schema, since it does not have multivalued attributes or composite attributes. All attributes within the schema are single and atomic.

B. All the conditions of the second normal form (2NF) are satisfied within the MEDICAL_RECORD relational schema, since every non-prime attribute is fully functionally dependent on the primary key ‘’Med_ID”.

C. All the conditions of the third normal form (3NF) are satisfied within the MEDICAL_RECORD relational schema, since the second normal form (2NF) is satisfied and there are no non-prime attributes that are transitively dependent on the primary key ‘’Med_ID”.

D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

2-EMERGENCY_CONTACT



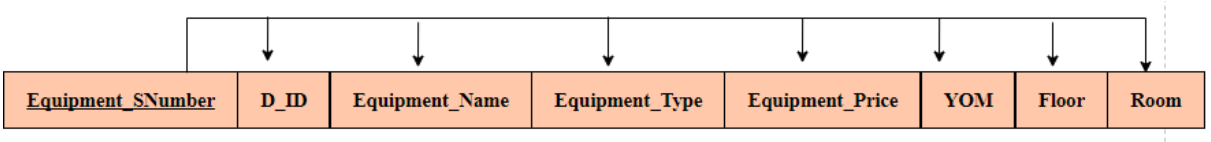
A. All the conditions of the first normal form (1NF) are satisfied within the EMERGENCY_CONTACTrelational schema, since it does not have multivalued attributes or composite attributes. All attributes within the schema are single and atomic.

B. All the conditions of the second normal form (2NF) are satisfied within the EMERGENCY_CONTACTrelational schema, since every non-prime attribute is fully functionally dependent on the primary key ‘’M_ID”.

C. All the conditions of the third normal form (3NF) are satisfied within the EMERGENCY_CONTACT relational schema, since the second normal form (2NF) is satisfied and there are no non-prime attributes that are transitively dependent on the primary key ‘’M_ID”.

D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

3. EQUIPMENT



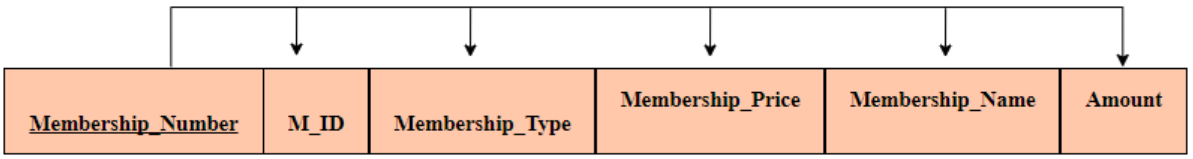
A. All the conditions of the first normal form (1NF) are satisfied within the EQUIPMENTrelational schema, since it does not have multivalued attributes or composite attributes. All attributes within the schema are single and atomic.

B. All the conditions of the second normal form (2NF) are satisfied within the EQUIPMENT relational schema, since every non-prime attribute is fully functionally dependent on the primary key ‘’Equipment_SNumber”.

C. All the conditions of the third normal form (3NF) are satisfied within the EQUIPMENT relational schema, since the second normal form (2NF) is satisfied and there are no non-prime attributes that are transitively dependent on the primary key ‘’Equipment_SNumber”.

D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

4. MEMBERSHIP



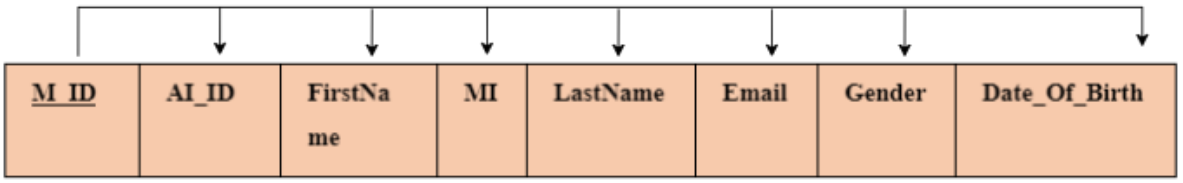
A. All the conditions of the first normal form (1NF) are satisfied within the MEMBERSHIPrelational schema, since it does not have multivalued attributes or composite attributes. All attributes within the schema are single and atomic.

B. All the conditions of the second normal form (2NF) are satisfied within the MEMBERSHIP relational schema, since every non-prime attribute is fully functionally dependent on the primary key ‘’Membership_Number”.

C. All the conditions of the third normal form (3NF) are satisfied within the MEMBERSHIP relational schema, since the second normal form (2NF) is satisfied and there are no non-prime attributes that are transitively dependent on the primary key ‘’Membership_Number”.

D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

5. **MEMBER**



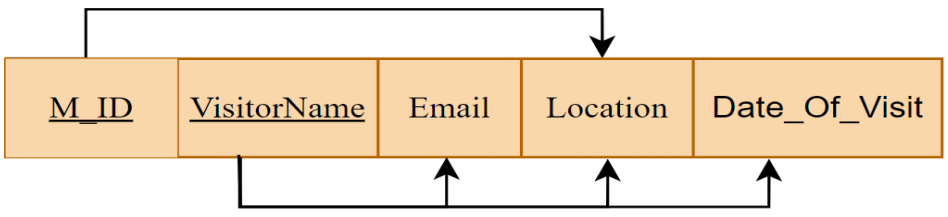
A. All the conditions of the first normal form (1NF) are satisfied within the MEMBERrelational schema, since it does not have multivalued attributes or composite attributes. All attributes within the schema are single and atomic.

B. All the conditions of the second normal form (2NF) are satisfied within the MEMBER relational schema, since every non-prime attribute is fully functionally dependent on the primary key ‘’Membership_Number”.

C. All the conditions of the third normal form (3NF) are satisfied within the MEMBER relational schema, since the second normal form (2NF) is satisfied and there are no non-prime attributes that are transitively dependent on the primary key ‘’Membership_Number”.

D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

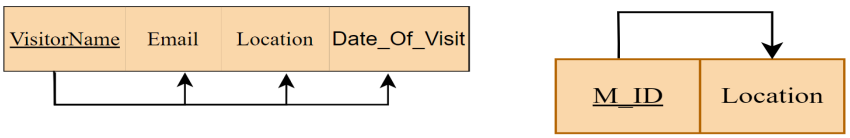
6. **VISITOR**



A. All the conditions of the first normal form (1NF) are satisfied within the VISITOR relational schema, since it does not have multivalued attributes or composite attributes. All attributes within the schema are single and atomic.

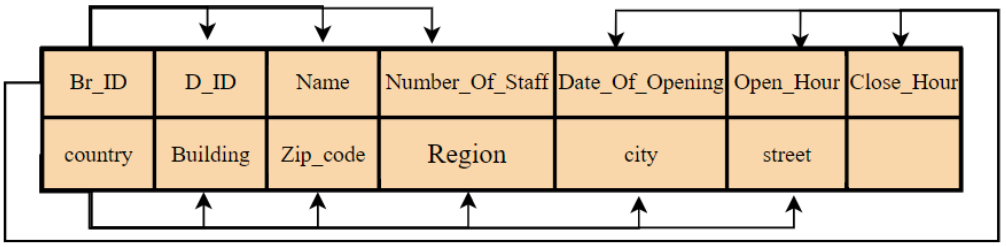
B. All the conditions of the second normal form (2NF) are satisfied within the VISITOR relational schema, since every non-prime attribute is fully functionally dependent on the primary key ‘’VisitorName and M_ID”.

C. Unfortunately, the (3NF) doesn't pass this test since {VisitorName, Patient_ID} is not a superkey because the attributes Email and Date_Of_Visit belong to VistorName exclusively. Therefore, to satisfy the client's requirements we must separate the relational schema into two tables like below:



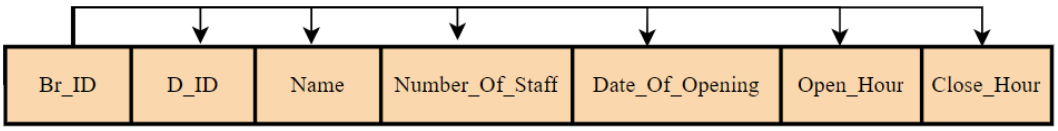
D. All the conditions of the Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X is not a super key.

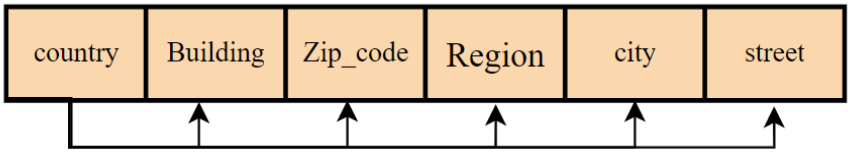
7.BRANCH



A. All the conditions of the first normal form (1NF) are satisfied within the BRANCH relational schema, since it does not have multivalued attributes and the original composite attribute got decomposed so now all attributes within the schema are single and atomic.

B. Unfortunately, the (2NF) doesn't pass this test since Building, Street and Country are non-prime attributes and are fully functionally dependent on City and Region. To satisfy the client's requirements we must construct two tables like below:

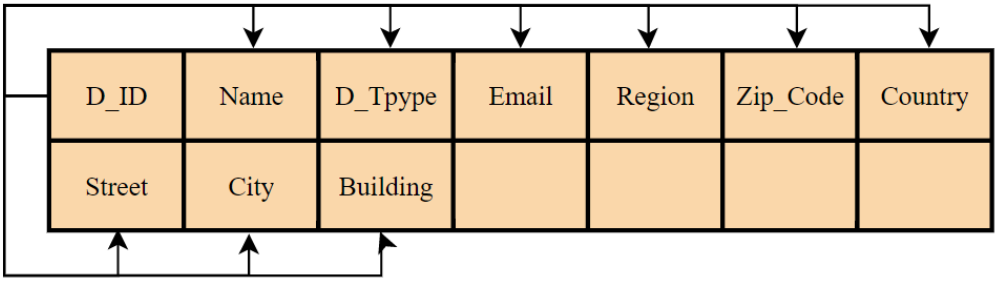




C. All the conditions of the third normal form (3NF) are satisfied within the BRANCH relational schema, since the second normal form is satisfied (2NF) and there are no non-prime attributes that are transitively dependent on the primary key "BR_ID".

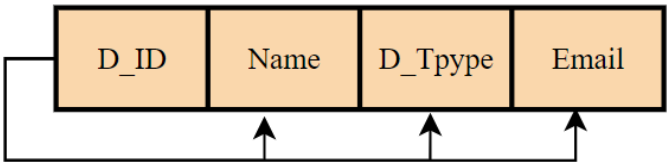
D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

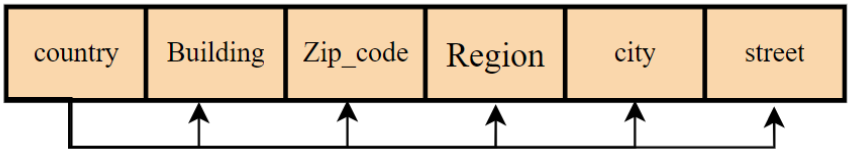
8.**DEPARTMENT**



A. All the conditions of the first normal form (1NF) are satisfied within the DEPARTMENT relational schema, since it does not have multivalued attributes and composite attributes. All attributes within the schema are single and atomic.

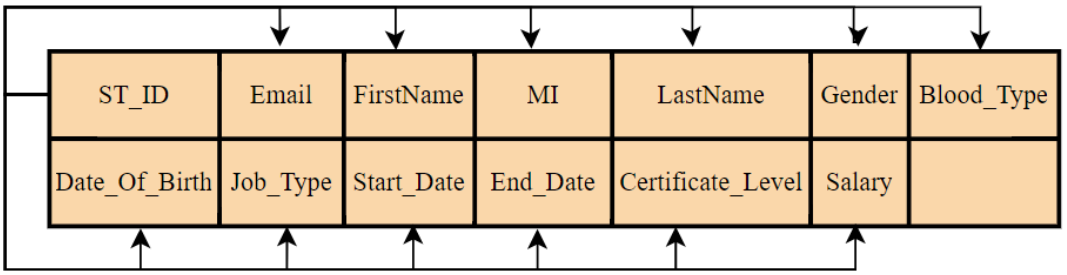
B. Unfortunately, the (2NF) doesn't pass this test since Building, Street and Country are non-prime attributes and are fully functionally dependent on City and Region. To satisfy the client's requirements we must construct two tables like below:





C. All the conditions of the third normal form (3NF) are satisfied within the DEPARTMENT relational schema, since the second normal form is satisfied (2NF) and there are no non-prime attributes that are transitively dependent on the primary key "D_ID".
D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

9. **STAFF**

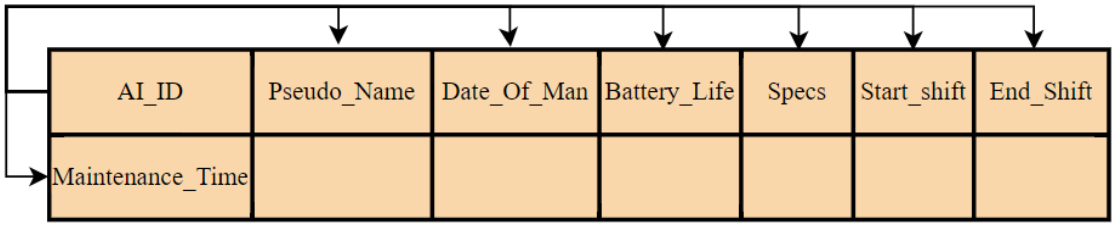


A. All the conditions of the first normal form (1NF) are satisfied within the STAFF relational schema, since it does not have multivalued attributes and composite attributes. All attributes within the schema are single and atomic.
B. All the conditions of the second normal form (2NF) are satisfied within the STAFF relational schema, since every non-prime attribute is fully functionally dependent on the primary key ''ST_ID".

C. All the conditions of the third normal form (3NF) are satisfied within the STAFF relational schema, since the second normal form is satisfied (2NF) and there are no non-prime attributes that are transitively dependent on the primary key "ST_ID".

D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

10. **AI_TRAINER**



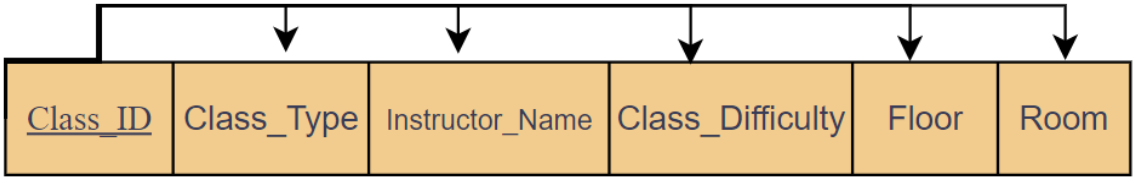
A. All the conditions of the first normal form (1NF) are satisfied within the AI_TRAINER relational schema, since it does not have multivalued attributes and composite attributes. All attributes within the schema are single and atomic.

B. All the conditions of the second normal form (2NF) are satisfied within the AI_TRAINER relational schema, since every non-prime attribute is fully functionally dependent on the primary key ''AI_ID".

C. All the conditions of the third normal form (3NF) are satisfied within the AI_TRAINER relational schema, since the second normal form is satisfied (2NF) and there are no non-prime attributes that are transitively dependent on the primary key "AI_ID".

D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

11. CLASS



A. All the conditions of the first normal form (1NF) are satisfied within the CLASS relational schema, since it does not have multivalued attributes and composite attributes. All attributes within the schema are single and atomic.
B. All the conditions of the second normal form (2NF) are satisfied within the CLASS relational schema, since every non-prime attribute is fully functionally dependent on the primary key ‘’Class_ID’’.
C. All the conditions of the third normal form (3NF) are satisfied within the CLASS relational schema, since the second normal form is satisfied (2NF) and there are no non-prime attributes that are transitively dependent on the primary key ‘’Class_ID’’.
D. All the conditions of Boyce-Codd Normal Form (BCNF) are satisfied since there exists no functional dependency X→ Y, where X is not a super key or Y is a prime attribute and X not a super key.

XV-Relation Schemas without non-prime attributes:

ATTENDS

C_ID	MEM_ID

MEMBER_Phone_Number

M_ID	Phone_Number

STAFF_Phone_Number

ST_ID	Phone_Number

VISITOR_Phone_Number

VisitorName	M_ID	Phone_Number

BRANCH_Phone_Number

BR_ID	Phone_Number

DEPARTMENT_Phone_Number

D_ID	Phone_Number

Conclusion

Databases are key components in our everyday lives, and society became dependent on them whether it's for a hospital, restaurant or a gymnasium like in our case.It is crucial to be able to track every customer, staff, trainers and how and where equipment is stored and to be capable of querying anything that might be needed at any time. It is important to store this data in a safe manner to protect everyone’s privacy.