CPCS241-Database I-Project

1st Semester, 2023

[Pump House Gym]

Student name	Student ID
Logain Sendi	
Layla	
Roaa	
Rawan	

Course Instructor: Dr.Manar Ali

Lab Instructor: Ms. Lulwa Al-Hariqi

Group Number: 2

Table Of Contents

PART I: ANALYSIS

1. Problem Definition and Data Requirements

1.1 Problem Definition

We deal with enormous volumes of data every day, especially in the business industry. Therefore, we must use a database to keep our data alive and retrieve it when needed. As a part of our project, our team is working on creating a database for a gym to store its data and manipulate it. In addition to the members' data, which includes their memberships and relatives, as well as the employees' data and the departments to which they belong, the data also includes gym branches, events, facility, classes, devices, and schedules.

The Pump House Gym project aims to create an efficient and easy-to-use database that will allow the gym to access whatever data it needs. For example, our database will facilitate access to class types, duration, and personal information about gym members, such as their phone numbers, weights, and heights.

1.2 Data Requirements

Branch

Each branch has:

- BranchNo.: a unique number for each branch.
- Branch_phoneNo.: the phone number for each branch.
- Branch_address: (Neighborhood, street).
- Working hours: (Opening hour, Closing hour).

Department

Each department has:

- DepatmentID: a unique id for each department.
- Department_name: a name for each department. For example, the department of administration, maintenance, medical clinic, etc.

Facility

Each facility has:

- FacilityNo.: a unique number for each facility.
- Facility_type: represents the facility type, it either be a pool or a room, for example exercise room, rest room, sauna room, changing room, etc.

  Note: pool and room are sub entities of the facility.
- FloorNo.: the floor on which the facility is located.

Class

Each class has:

- Class_type: the type of the class. For example, Zumba, cardio, etc.
- Class_duration: duration of the class.

Schedule

Each schedule has:

- ScheduleID: a unique number for each schedule.
- Date: The schedule includes a date on which the class will be given.
- Time: Represents the start time of the class.

Membership

Any membership plan will guarantee the following:

- MembershipID: membership number.
- Membership_duration: 1 month, 3 months, 6 months, or 12 months.
- Membership_price: member subscription cost- the price might be different from branch to another even if the duration is the same.

Employee

Each employee has:

- EmployeeID: a unique number for each employee.
- Employee_name: (First name, Last name).
- Employee_DOB: the date of birth as (Day, month, year).
- Employee_jobTitle: the employee career. For example, managers, cleaners, trainers, maintenance workers, etc.

  Note: Trainers, cleaners, maintenance worker, and managers are sub entities of the employee with additional attributes for trainer which is Qualification: this attribute represents what the trainer is qualified to do.
- Employee_phoneNo.: the phone number of the employee.
- Employee_email: the email address of the employee.
- Employee_salary: the salary of the employee.

Member

Each member has:

- MemberID: a unique ID for each member.
- Member_name: a name for each member.
- Member_DOB: birth date for each Member.
- Member_PhoneNo.: a phone number for each member.
- HealthProfile: (Weight, Height, Water_percentage, Muscle_mass_percentage, Fat_percentage, and Date) The system stores each member's health profile along with the date of the measurements so that the health profile is updated monthly.

Device

Each device has:

- DeviceID: a unique number for each device.
- Device_name: name of the device.
- Device_cost: the cost of the device.

Event

Each event has:

- Event_name: a unique name for each event.
- Event_date: the date of the event as (Day, Month, Year).
- Event_duration: the number of hours the event will take.

Relative

The system will keep the information of members' relatives so that they can attend some events. Each relative has:

- Relative_name: the name of each relative.
- Relative_DOB: the date of birth for each relative as (Day, Month, Year).
- Relationship: describes the type of relationship between the member and the relative.

Device Maintenance Schedule

Each device maintenance schedule has:

- Date: indicates the times during which the device is maintained.

Pool Maintenance Schedule

Each pool maintenance schedule has:

- Date: indicates the times during which the pool is maintained.

1.3 Business Rules

- Each branch **has** several departments, and each department belongs to a branch.
- Each branch **offers** different memberships (either 3,6, or 12 months) and each membership is available at different branches.
- Each branch can **host** multiple events, and each event can be hosted by multiple branches
- Each department **has** several employees, and each employee works for a single department.
- Each member should **sign up** for one membership plan (either 1,3,6, or 12 months) and each membership plan has multiple members.
- Each member can **have** many relatives, and each relative must have a member.
- Each member can **register** in multiple schedules, and each schedule can include many registered members.
- Each schedule **includes** one class, and each class can be in multiple schedules.
- Each schedule **includes** one trainer, and each trainer can be in multiple schedules.
- Each schedule **includes** one facility, and each facility can be in multiple schedules.
- A room **has** multiple devices, and each device can exist in only one room.
- Each maintenance worker **has** multiple device maintenance schedules, and each schedule belongs to a single maintenance worker.
- Each device **has** multiple maintenance schedules, and each maintenance schedule contains the device to be maintained.
- Each maintenance worker **has** multiple pool maintenance schedules, and each schedule belongs to a single maintenance worker.
- Each pool **has** multiple maintenance schedules, and each maintenance schedule contains the pool to be maintained.
- A manager may **manage** several events, and an event can be managed by only one manager.
- Relatives of a member can **attend** many events, and an event can be attended by many relatives

1.4 Intended Output of the system

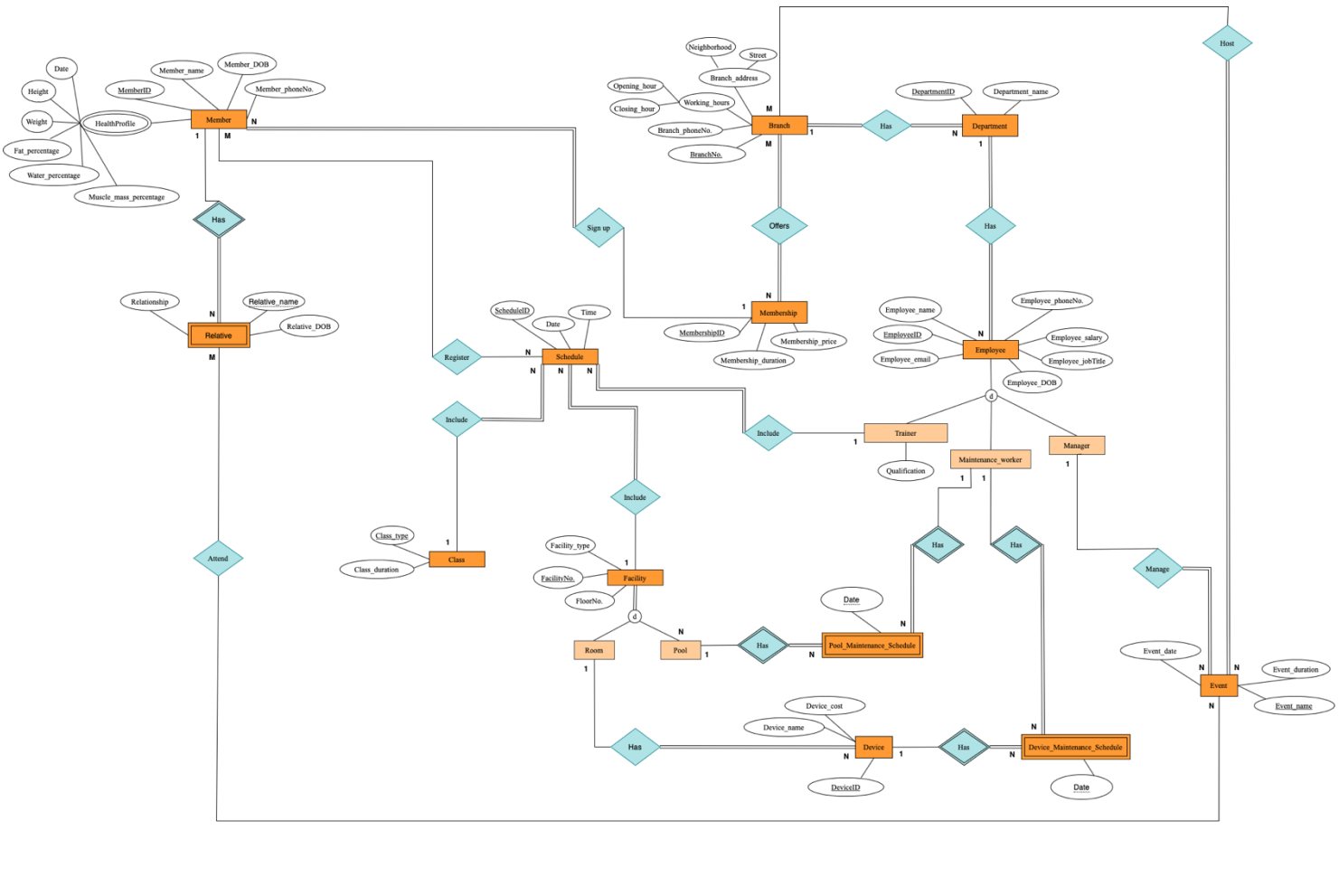
- Display the details of the device that has the highest cost.
- Find the top 3 registered classes.
- Display all employees with their job type.
- Display all trainers with their classes.
- Display all classes with their time.
- Display all devices with location.
- Display all members with their health profile.
- Display all department details.
- Display membership details.
- Display all lockers details.
- Search for a department by name.
- Search for room type by its room number.
- Search for an employee by ID.
- Search for a member by ID.
- Retrieve all members with the same membership plan, etc.

PART II: DB DESIGN

2. ER Diagram Design

2.1 ER Diagram

To see it more clearly follow the link: https://up6.cc/2022/11/166815078588261.png



2.2 Design of Business Rules

Business Rule	Design Decisions	Justification (if any)
Each branch **has** several departments, and each department belongs to a branch.	Binary 1:N	Both the branch and department entity have total participation with the relationship "has".
Each branch **offers** different memberships (either 1,3,6, or 12 months) and each membership is available at different branches.	Binary M:N	Both the branch and memberships entity have total participation with the relationship "offers".
Each branch can **host** multiple events, and each event can be hosted by multiple branches.	Binary M:N	The branch entity has partial participation with the relationship "host" while the event entity has total participation.
Each department **has** several employees, and each employee works for a single department.	Binary 1:N	Both the department and employee entity have total participation with the relationship "has".
Each member should **sign up** for one membership plan (either 3,6, or 12 months) and each membership plan has multiple members.	Binary N:1	The member entity has total participation with the relationship "sign up" while the membership entity has partial participation.
Each member can **have** many relatives, and each relative must have a member.	Binary 1:N	The relative entity is a weak entity since it is not needed if there is no member.
Each member can **register** in multiple schedules, and each schedule can include many registered members.	Binary M:N	Both the member and schedule entity have partial participation with the relationship "register". A member does not have to register for a schedule and can use the devices instead.
Each schedule **includes** one class, and each class can be in multiple schedules.	Binary N:1	The schedule entity has total participation with the relationship "include" while the class entity has partial participation.

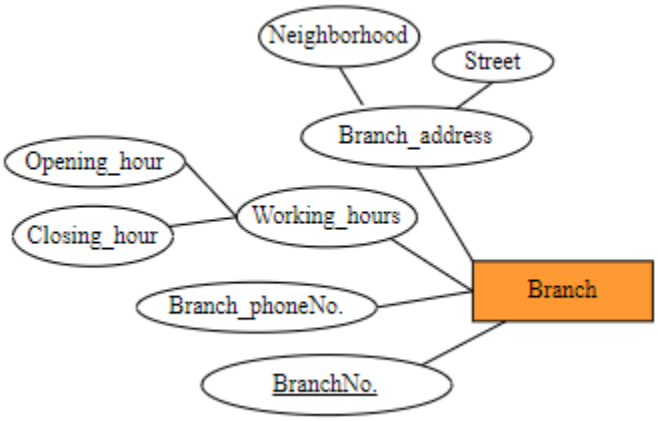
Each schedule **includes** one trainer, and each trainer can be in multiple schedules.	Binary N:1	The schedule entity has total participation with the relationship “include” while the trainer sub-entity has partial participation.
Each schedule **includes** one facility, and each facility can be in multiple schedules.	Binary N:1	The schedule entity has total participation with the relationship “include” while the facility entity has partial participation.
A room **has** multiple devices, and each device can exist in only one room.	Binary 1:N	The room sub-entity has partial participation with the relationship “has” while the device entity has total participation.
Each maintenance worker **has** multiple device maintenance schedules, and each schedule belongs to a single maintenance worker.	Binary 1:N	The maintenance worker sub-entity has partial participation with the relationship “has” while the device maintenance schedule entity has total participation.
Each device **has** multiple maintenance schedules, and each maintenance schedule contains the device to be maintained.	Binary 1:N	The device entity has partial participation with the relationship “has” while the device maintenance schedule entity has total participation.
Each maintenance worker **has** multiple pool maintenance schedules, and each schedule belongs to a single maintenance worker.	Binary 1:N	The maintenance worker sub-entity has partial participation with the relationship “has” while the pool maintenance schedule entity has total participation.
Each pool **has** multiple maintenance schedules, and each maintenance schedule contains the pool to be maintained.	Binary 1:N	The pool sub-entity has partial participation with the relationship “has” while the pool maintenance schedule entity has total participation.
A manager may **manage** several events, and an event can be managed by only one manager.	Binary 1:N	The manager sub-entity has partial participation with the relationship “manage” while the event entity has total participation.

Relatives of a member can **attend** many events, and an event can be attended by many relatives.	Binary M:N	Both the relative and event entity have partial participation with the relationship “attend”.

3. ER-to-logical schema mapping

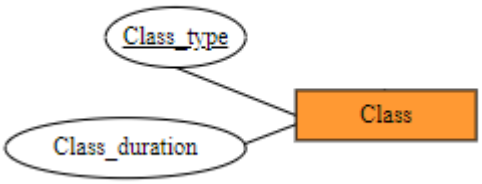
3.1 Mapping of Regular Entity Types

We will create a new relation for each regular entity and include all its simple attributes, then choose one of the key attributes, if there is more than one, to be the primary key.



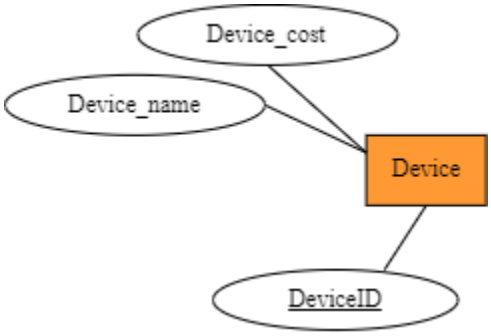
Branch

BranchNo.	Branch_phoneNo.	Neighborhood	Street	Opening_hour	Closing_hour



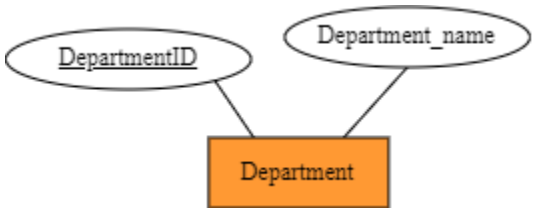
Class

Class_type	Class_duration



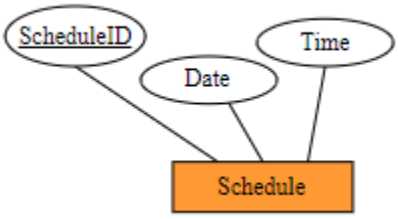
Device

DeviceID	Device_name	Device_cost



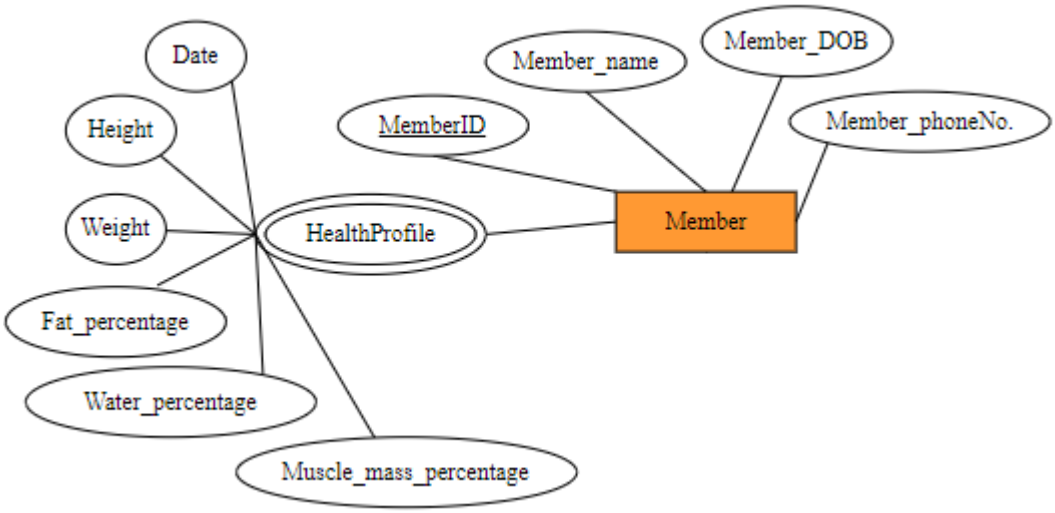
Department

DepartmentID	Department_name



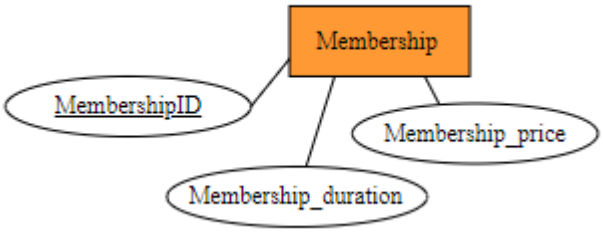
Schedule

ScheduleID	Date	Time



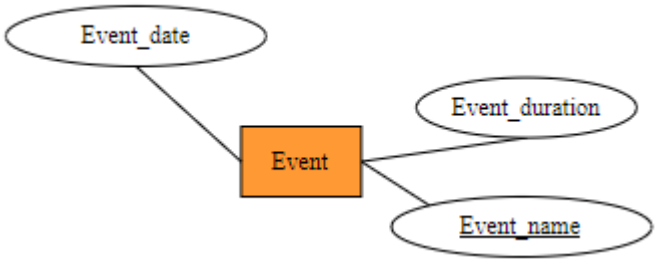
Member

MemberID	Member_name	Member_DOB	Member_phoneNo.



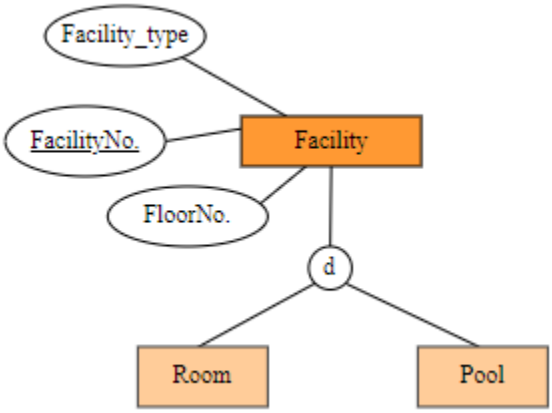
Membership

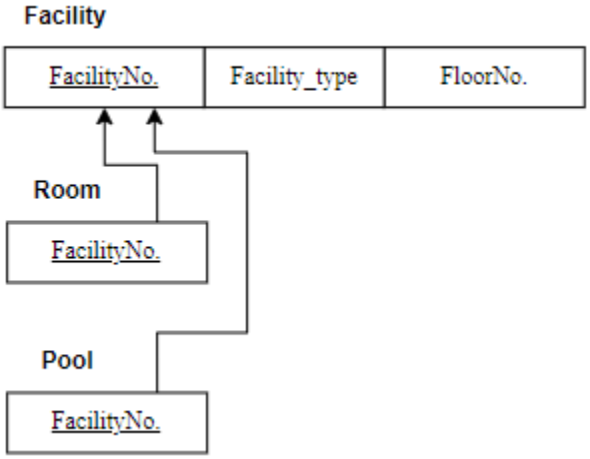
MembershipID	Membership_duration	Membership_price

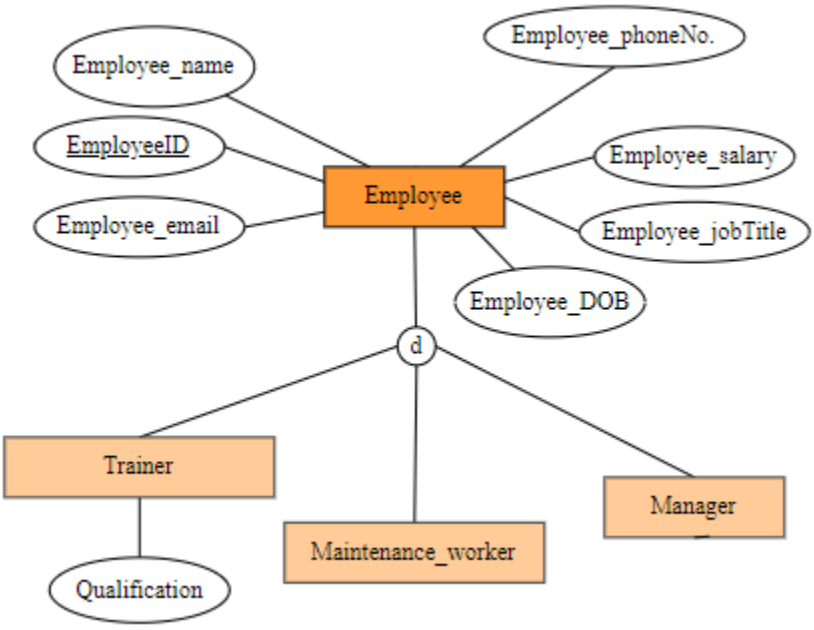


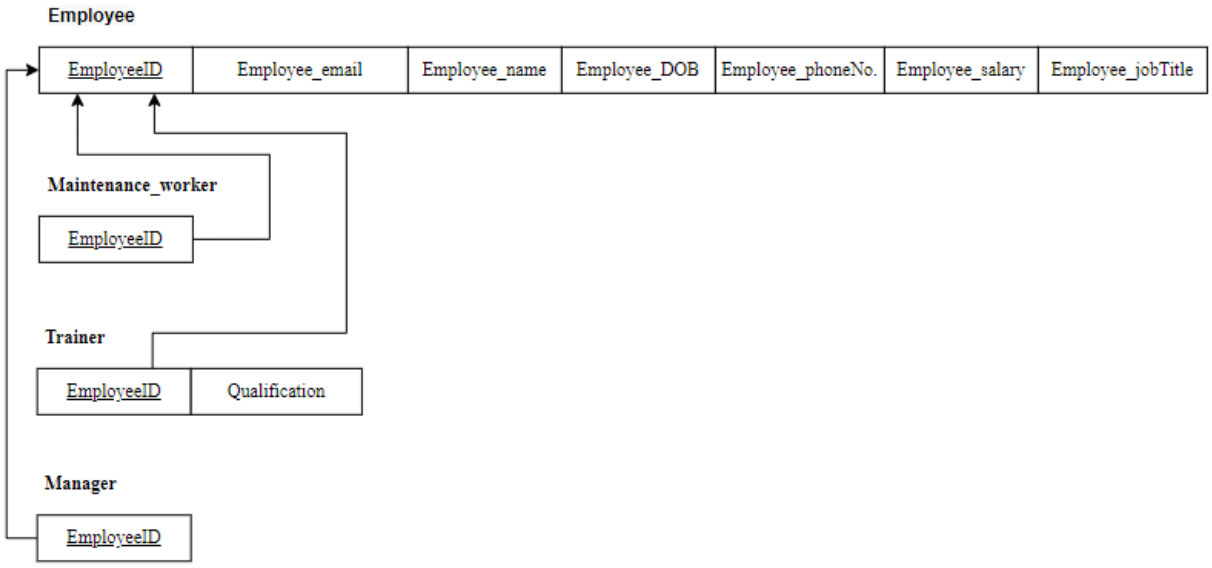
Event

Event_name	Event_date	Event_duration


Facility_type
FacilityNo.
Facility
FloorNo.
d
Room
Pool

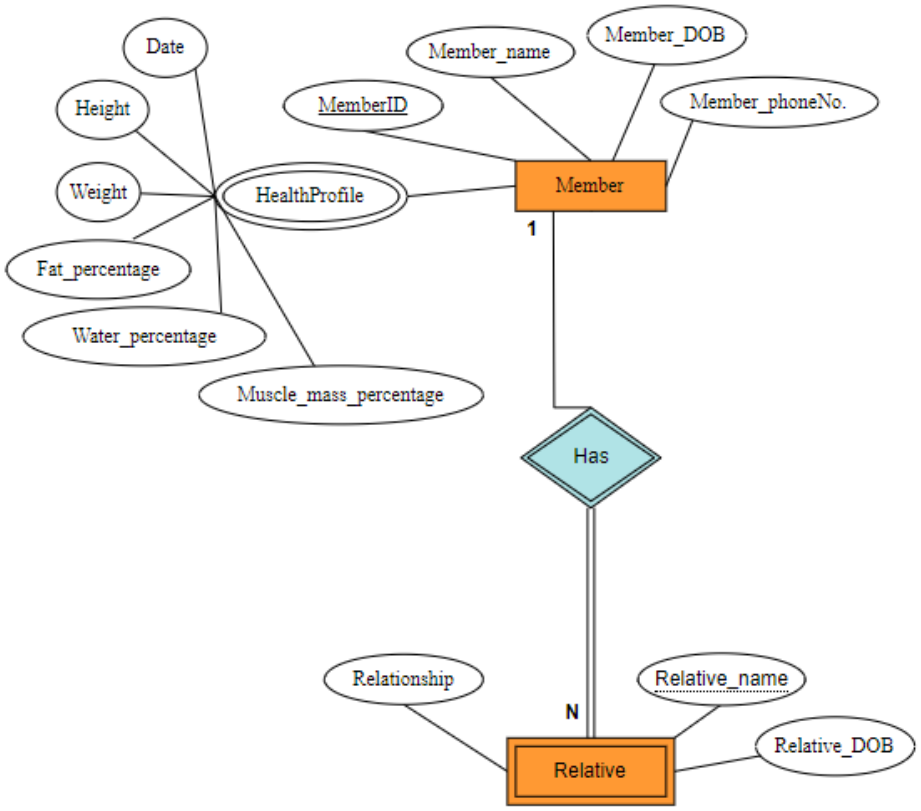

Facility
FacilityNo.
Facility_type
FloorNo.
Room
FacilityNo.
Pool
FacilityNo.

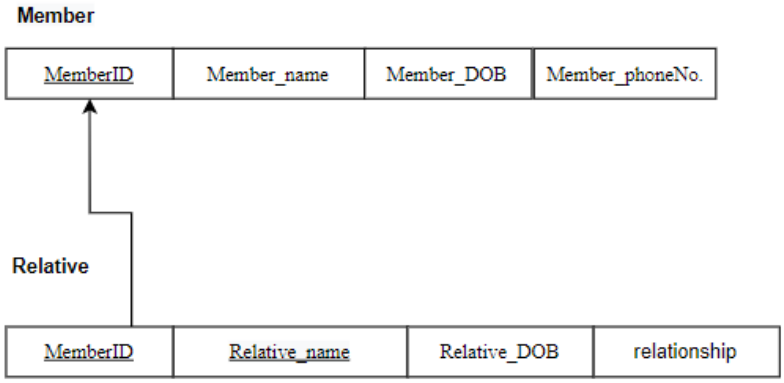

Employee_name
Employee_phoneNo.
EmployeeID
Employee_salary
Employee
Employee_email
Employee_jobTitle
Employee_DOB
d
Trainer
Manager
Maintenance_worker
Qualification

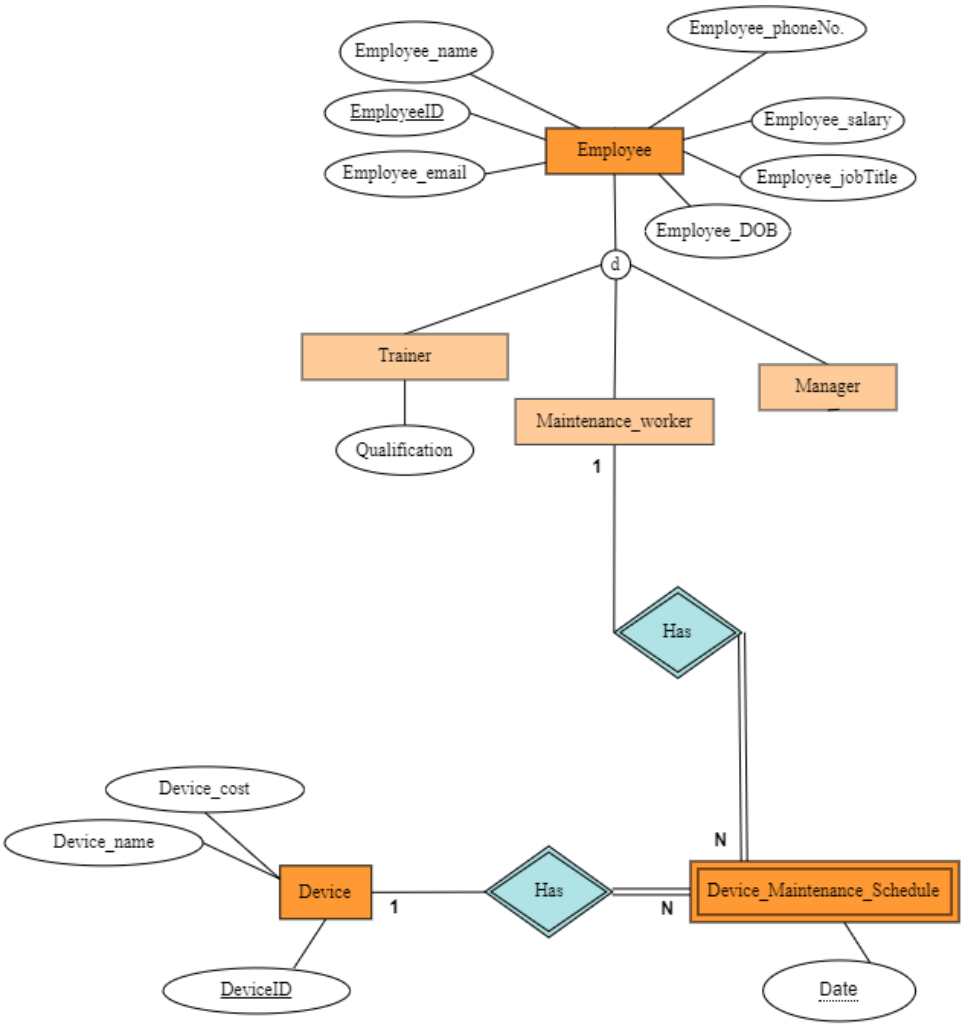

Employee
EmployeeID
Employee_email
Employee_name
Employee_DOB
Employee_phoneNo.
Employee_salary
Employee_jobTitle
Maintenance_worker
EmployeeID
Trainer
EmployeeID
Qualification
Manager
EmployeeID

3.2 Mapping of Weak Entity Types

We will create a new relation for the weak entity and include its simple attributes along with a foreign key that references the primary key of the owner entity. The primary key of the weak entity will be represented by both the partial key and the foreign key to the owning entity.







Employee

EmployeeID	Employee_email	Employee_name	Employee_DOB	Employee_phoneNo.	Employee_salary	Employee_jobTitle

Maintenance _worker

EmployeeID

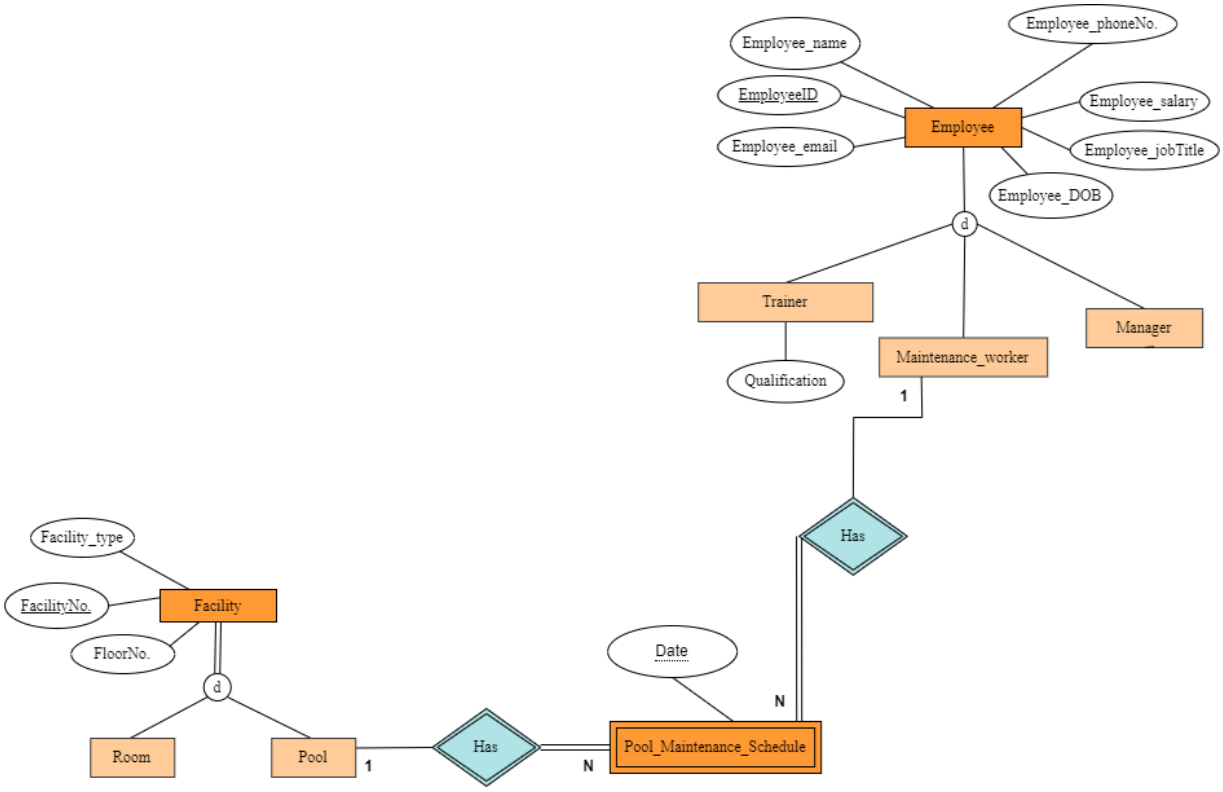
Device

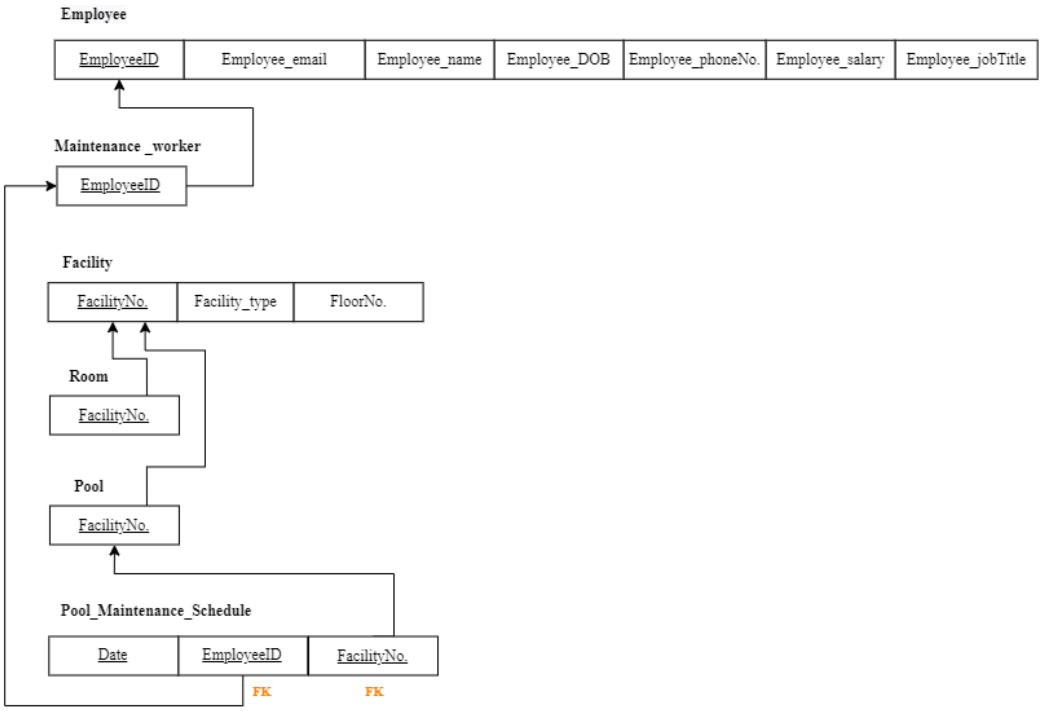
DeviceID	Device_name	Device_cost

Device_Maintenance_Schedule

Date	EmployeeID	DeviceID

FK FK


Employee_name
Employee_phoneNo.
EmployeeID
Employee_salary
Employee
Employee_email
Employee_jobTitle
Employee_DOB
d
Trainer
Manager
Maintenance_worker
Qualification
1
Has
Facility_type
FacilityNo.
Facility
FloorNo.
Date
d
N
Room
Pool
1
Has
N
Pool_Maintenance_Schedule

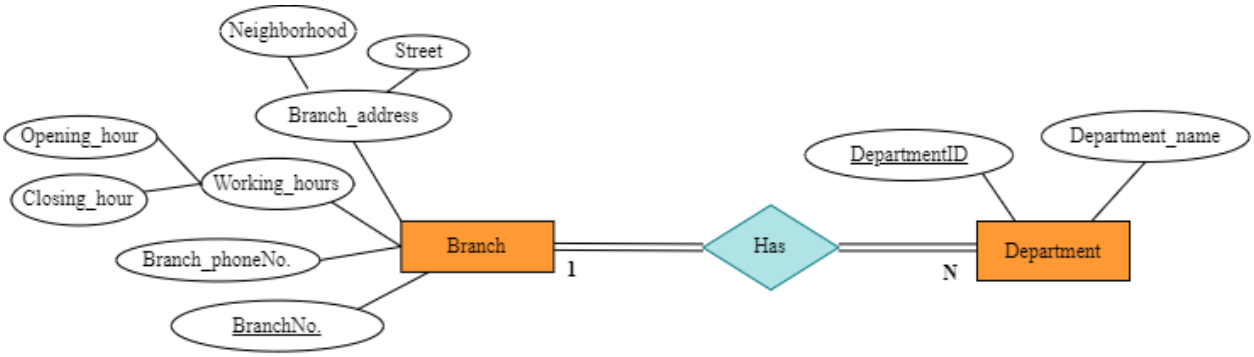

Employee
EmployeeID
Employee_email
Employee_name
Employee_DOB
Employee_phoneNo.
Employee_salary
Employee_jobTitle
Maintenance _worker
EmployeeID
Facility
FacilityNo.
Facility_type
FloorNo.
Room
FacilityNo.
Pool
FacilityNo.
Pool_Maintenance_Schedule
Date
EmployeeID
FacilityNo.
FK
FK

3.3 Mapping of binary 1-1 relationship types

The system does not have any 1-1 relationships.

3.4 Mapping of binary 1-N relationship types

To map binary 1:N relationship, we will include a foreign key on the N side.



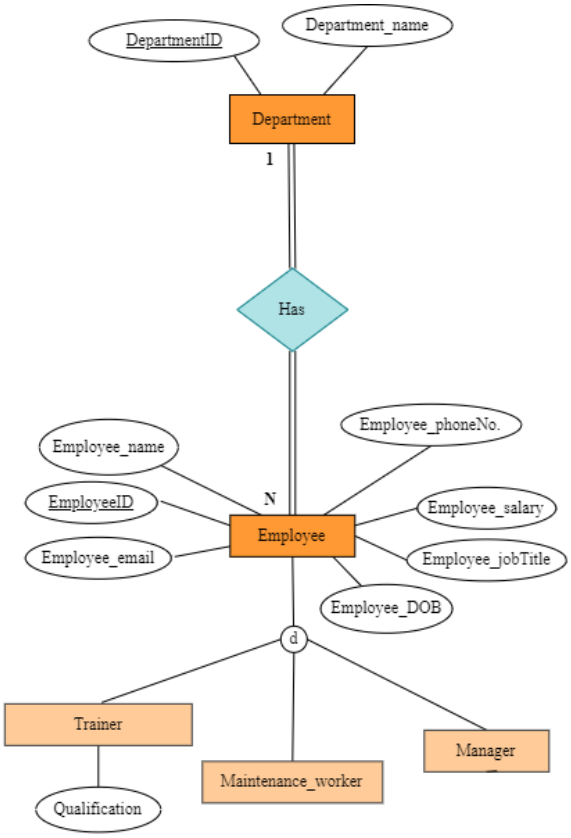
Branch

BranchNo.	Branch_phoneNo.	Neighborhood	Street	Opening_hour	Closing_hour

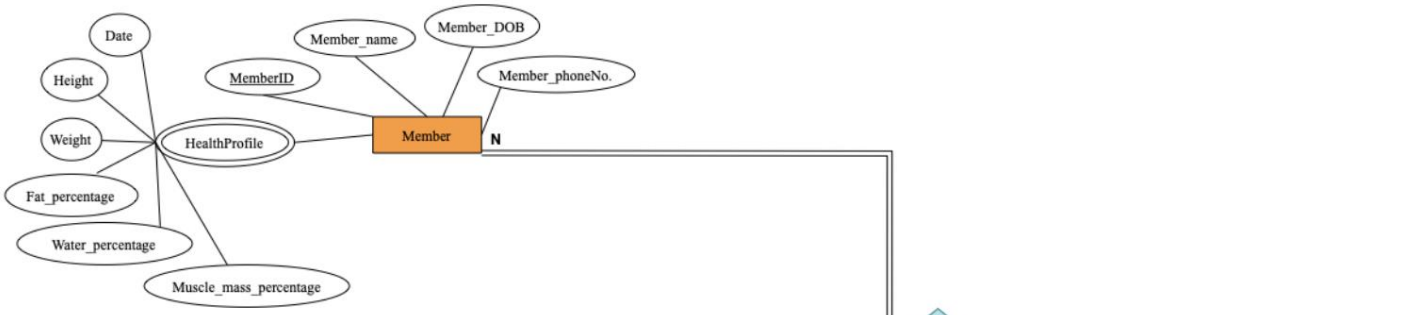
Department

DepartmentID	Department_name	BranchNo.

FK


DepartmentID
Department_name
Department
1
Has
N
Employee_name
EmployeeID
Employee_email
Employee_phoneNo.
Employee_salary
Employee_jobTitle
Employee_DOB
Employee
d
Trainer
Qualification
Maintenance_worker
Manager


Department
DepartmentID
Department_name
BranchNo.
FK
Employee
EmployeeID
Employee_email
Employee_name
Employee_DOB
Employee_phoneNo.
Employee_salary
Employee_jobTitle
DepartmentID
FK
Maintenance_worker
EmployeeID
Trainer
EmployeeID
Qualification
Manager
EmployeeID



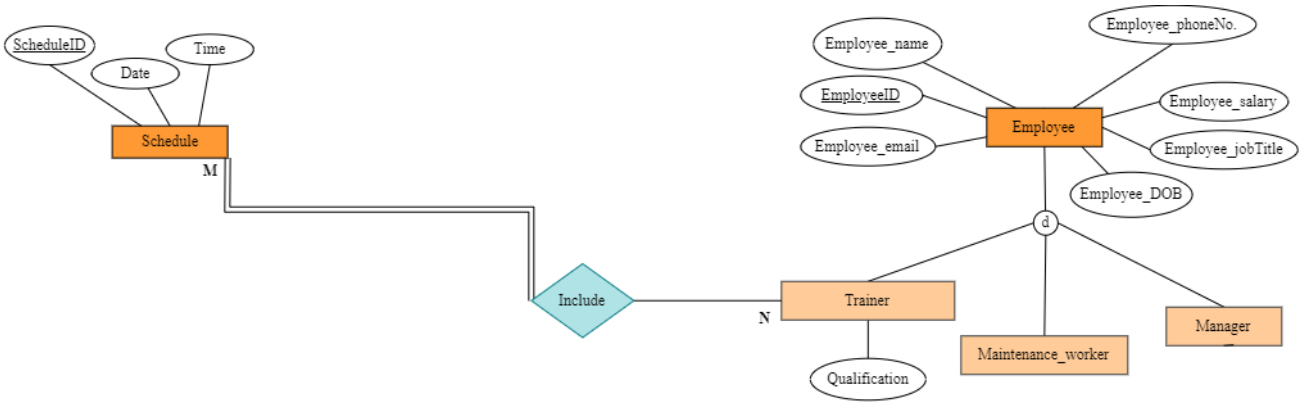
Membership

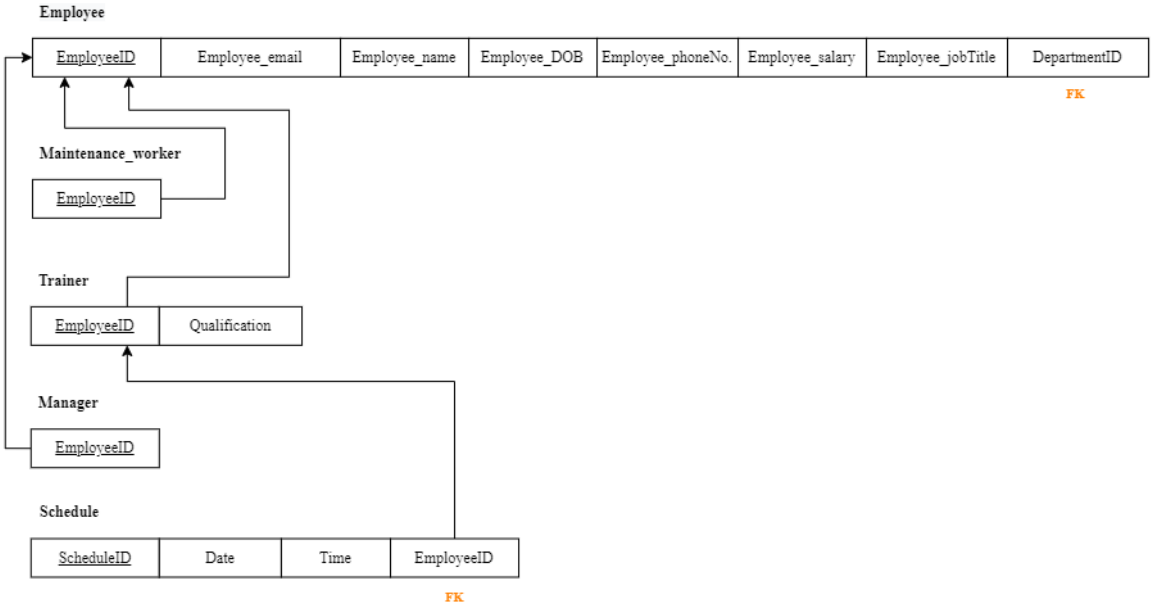
MembershipID	Membership_duration	Membership_price

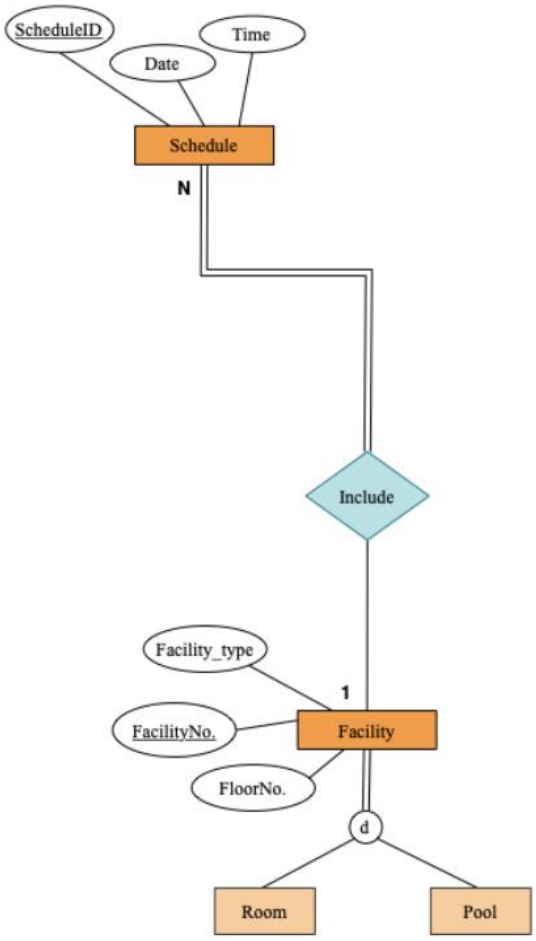
Member

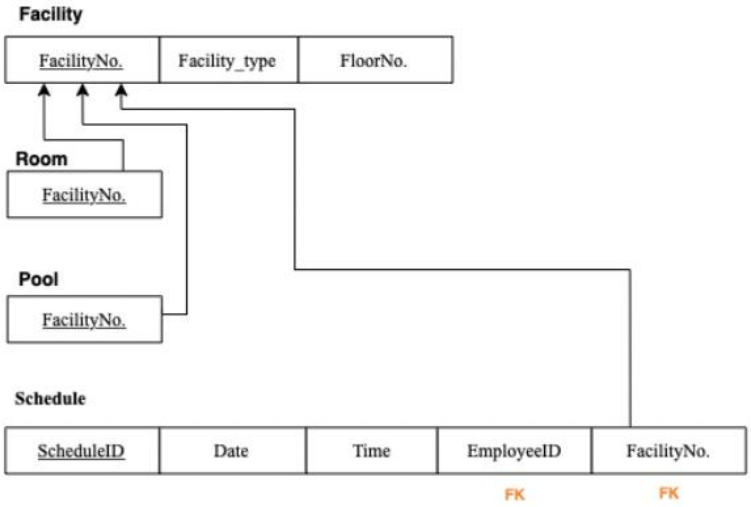
MemberID	Member_name	Member_DOB	Member_phoneNo.	MembershipID

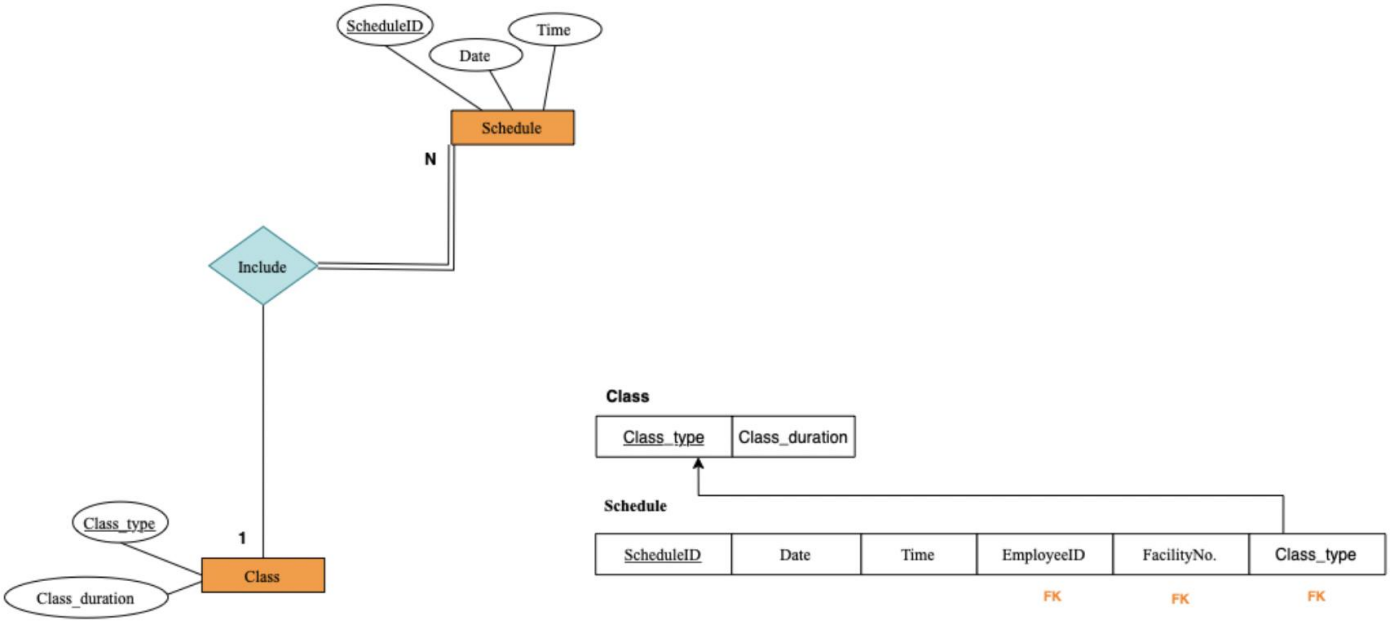
FK

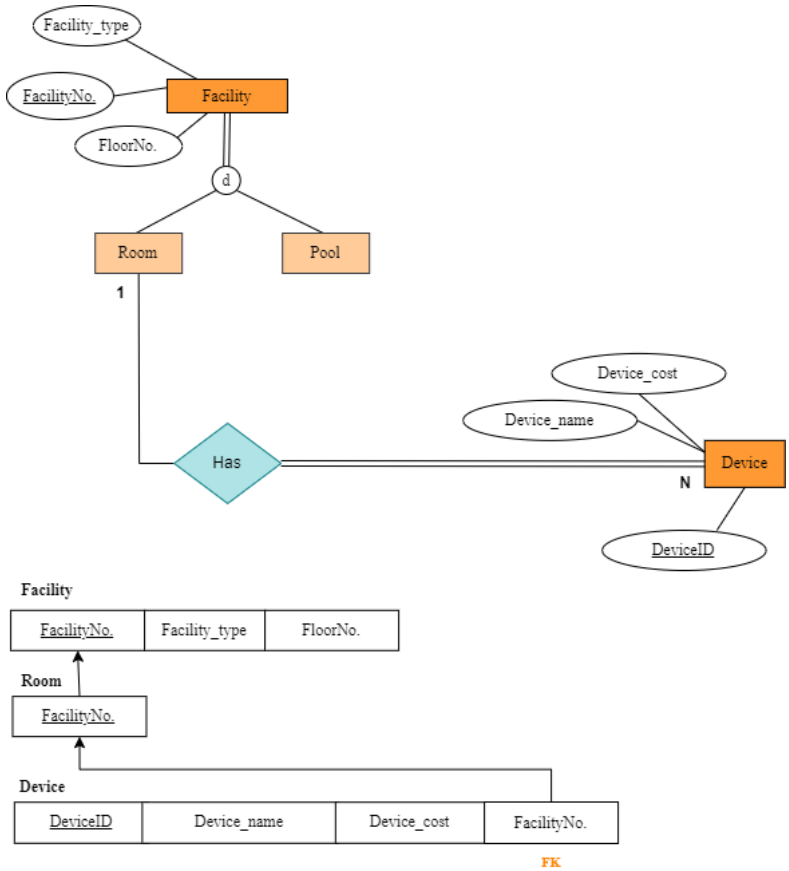

ScheduleID
Time
Date
Schedule
M
Include
N
Employee_name
Employee_phoneNo.
EmployeeID
Employee_salary
Employee
Employee_email
Employee_jobTitle
Employee_DOB
d
Trainer
Manager
Maintenance_worker
Qualification

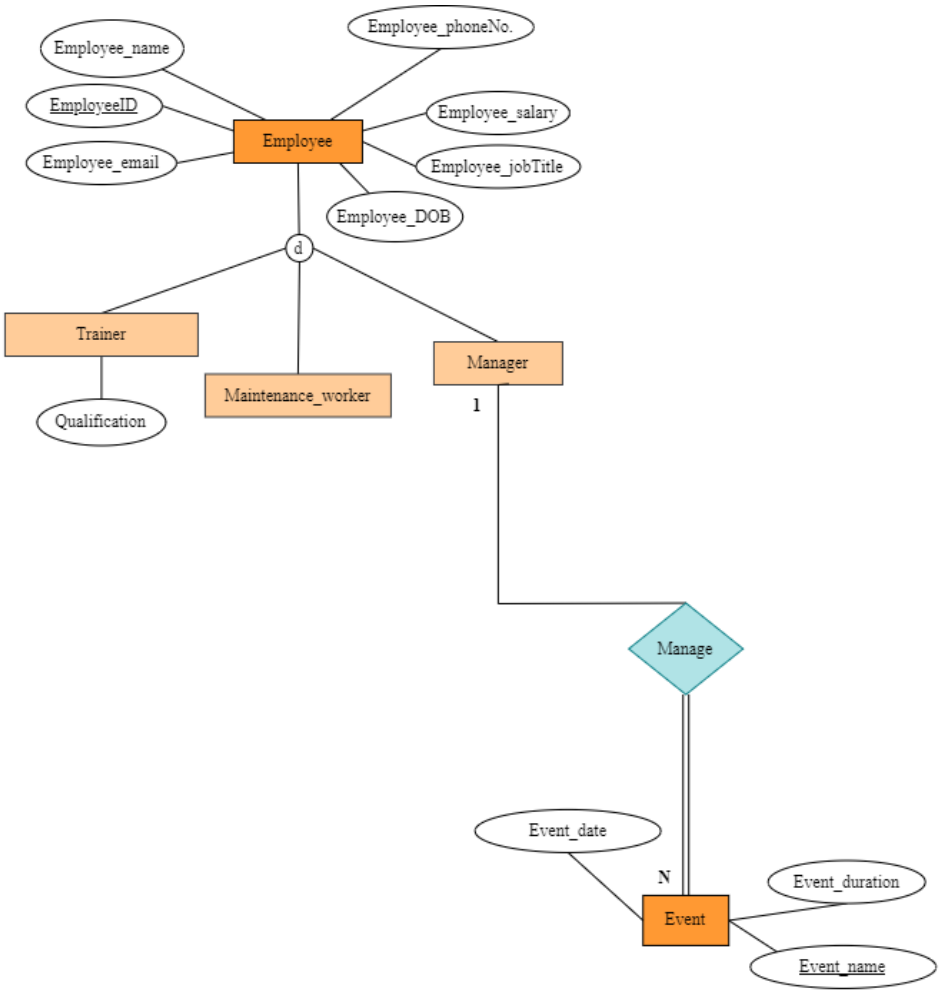

Employee
EmployeeID
Employee_email
Employee_name
Employee_DOB
Employee_phoneNo.
Employee_salary
Employee_jobTitle
DepartmentID
FK
Maintenance_worker
EmployeeID
Trainer
EmployeeID
Qualification
Manager
EmployeeID
Schedule
ScheduleID
Date
Time
EmployeeID
FK


ScheduleID
Time
Date
Schedule
N
Include
Facility_type
1
FacilityNo.
Facility
FloorNo.
d
Room
Pool


Facility
FacilityNo.
Facility_type
FloorNo.
Room
FacilityNo.
Pool
FacilityNo.
Schedule
ScheduleID
Date
Time
EmployeeID
FacilityNo.
FK
FK


ScheduleID
Time
Date
Schedule
N
Include
Class_type
1
Class
Class_duration
Class
Class_type
Class_duration
Schedule
ScheduleID
Date
Time
EmployeeID
FacilityNo.
Class_type
FK
FK
FK


Facility_type
FacilityNo.
Facility
FloorNo.
d
Room
Pool
1
Device_cost
Device_name
Has
Device
N
DeviceID
Facility
FacilityNo.
Facility_type
FloorNo.
Room
FacilityNo.
Device
DeviceID
Device_name
Device_cost
FacilityNo.
FK



Employee

EmployeeID	Employee_email	Employee_name	Employee_DOB	Employee_phoneNo.	Employee_salary	Employee_jobTitle	DepartmentID

FK

Manager

EmployeeID

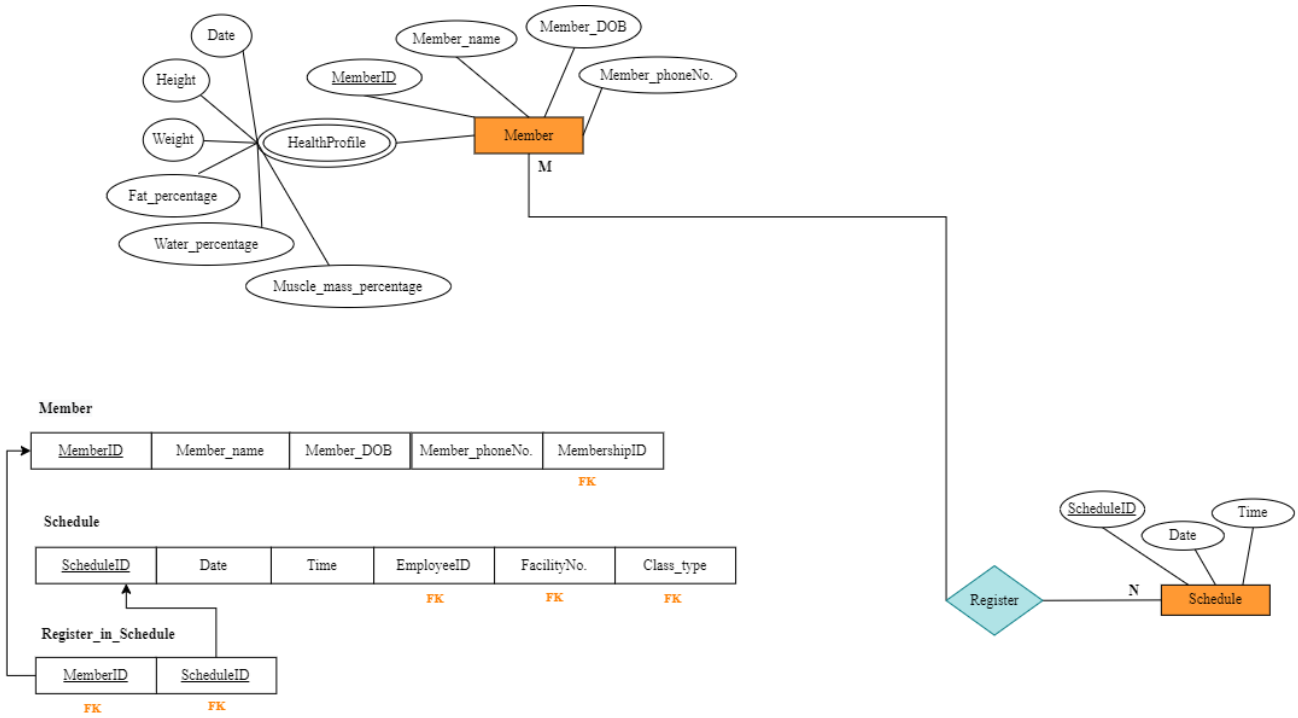
Event

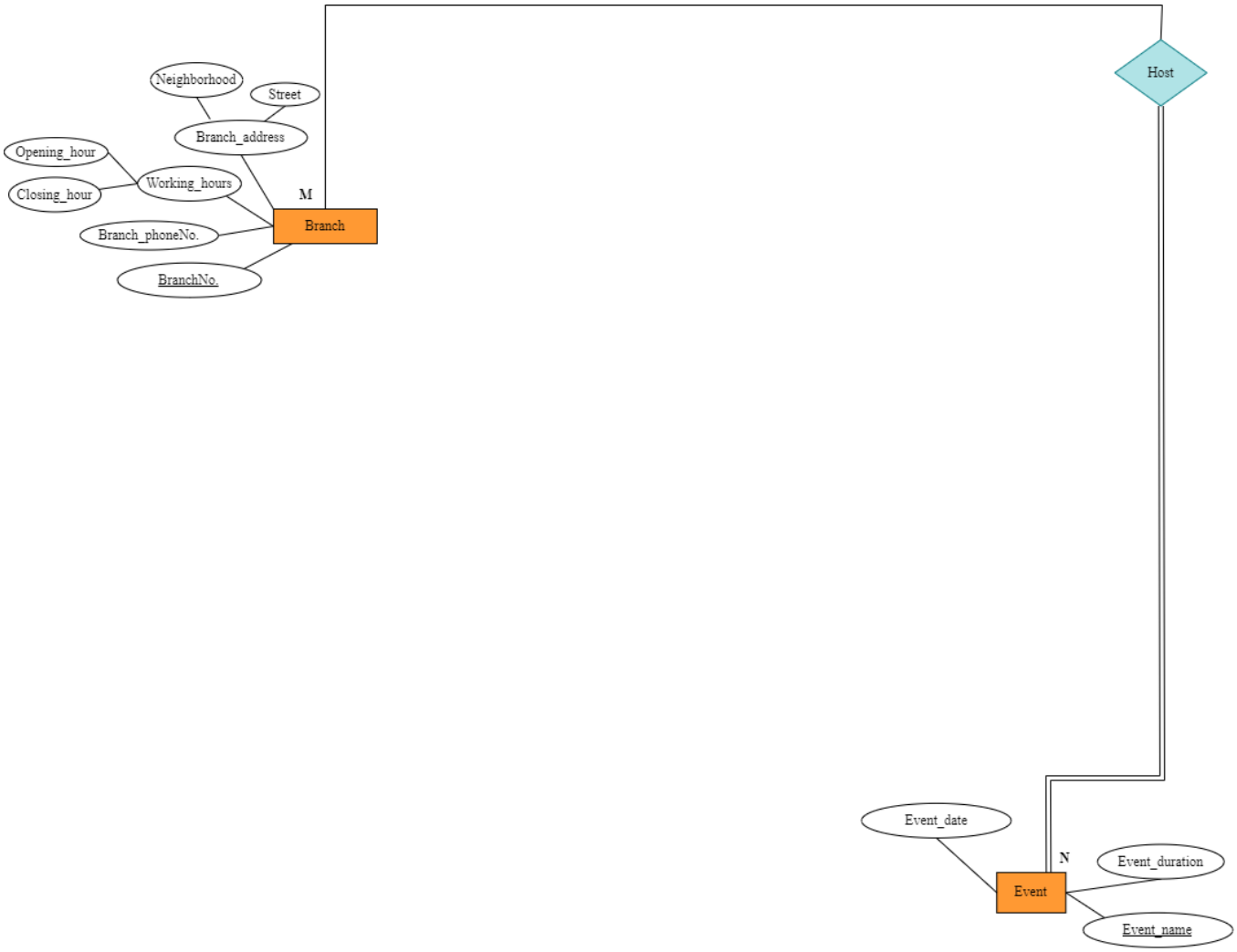
Event_name	Event_date	Event_duration	EmployeeID

FK

3.5 Mapping of binary M-N relationship types

Create a new relation for the relationship then the combination of primary keys of both entities will be the primary key of the new relation.





Branch

BranchNo.	Branch_phoneNo.	Neighborhood	Street	Opening_hour	Closing_hour

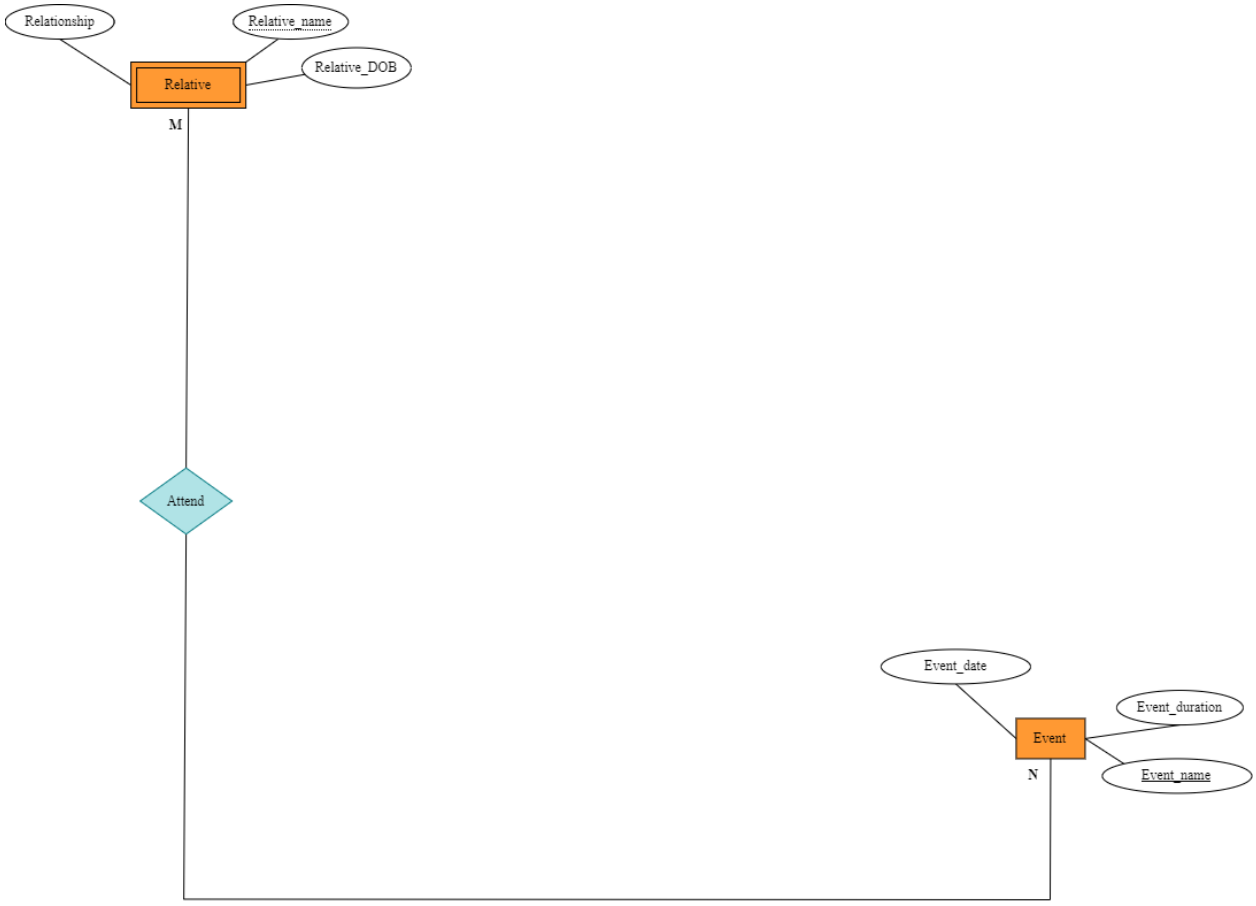
Event

Event_name	Event_date	Event_duration	EmployeeID

FK

Event_Hosting

Event_name	BranchNo.



Relative

MemberID	Relative_name	Relative_DOB	Relationship

Event

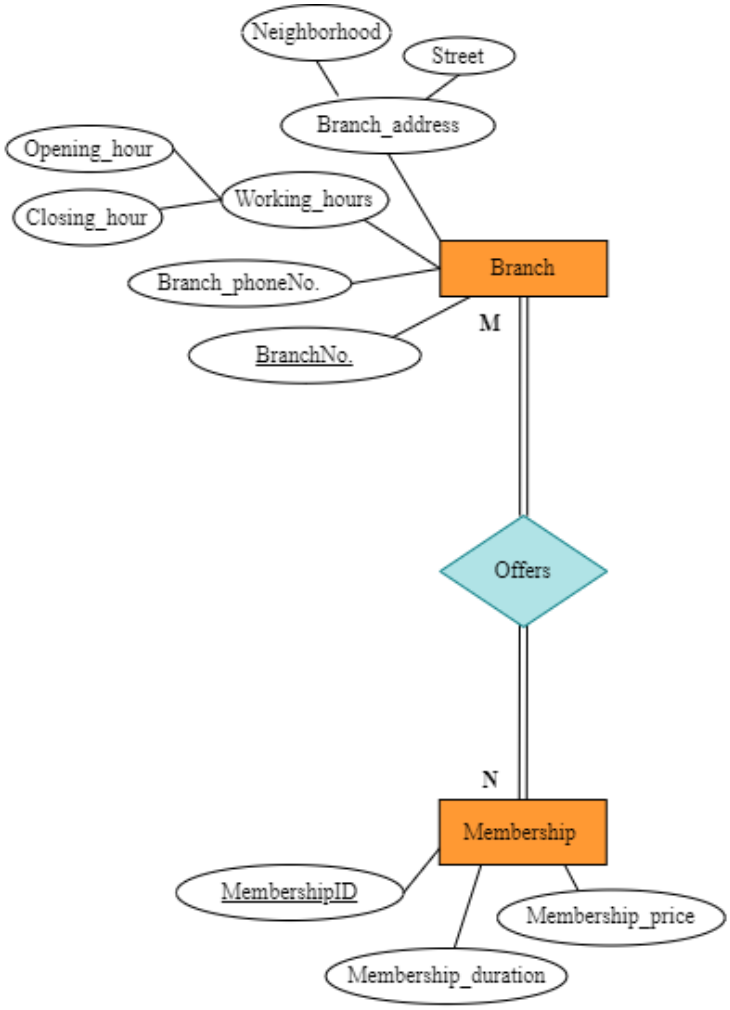
Event_name	Event_date	Event_duration	EmployeeID

FK

Attend_Event

MemberID	Relative_name	Event_name

FK FK



Branch

BranchNo.	Branch_phoneNo.	Neighborhood	Street	Opening_hour	Closing_hour

Membership

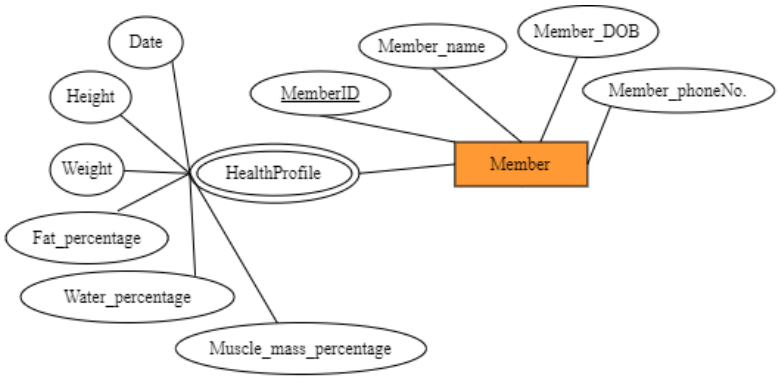
MembershipID	Membership_duration	Membership_price

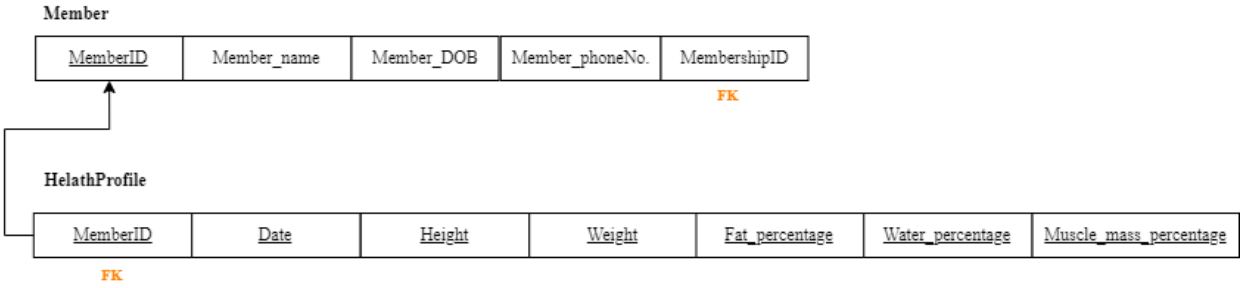
Branch_Membership

BranchNo.	MembershipID

3.6 Mapping of multivalued attributes

To map the multivalued attribute, we need to create a separate relation for multivalued attribute along with a foreign key that references the primary key of the relation that has this attribute.



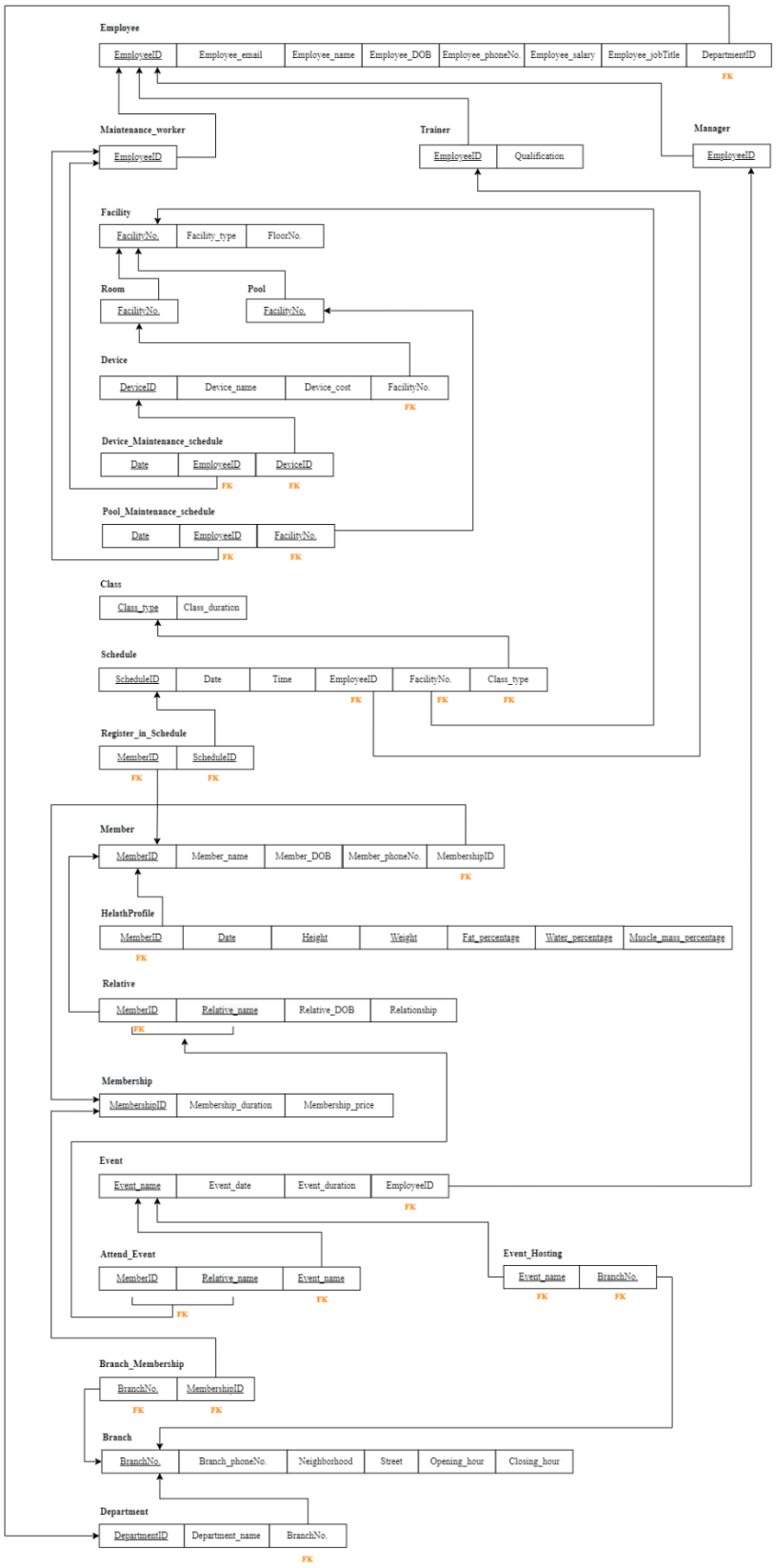


3.7 Mapping of n-ary relationship types

The system does not have any n-ary relationships

3.8 Schema Diagram

To see it more clearly follow the link: https://up6.cc/2022/11/166815100806131.png

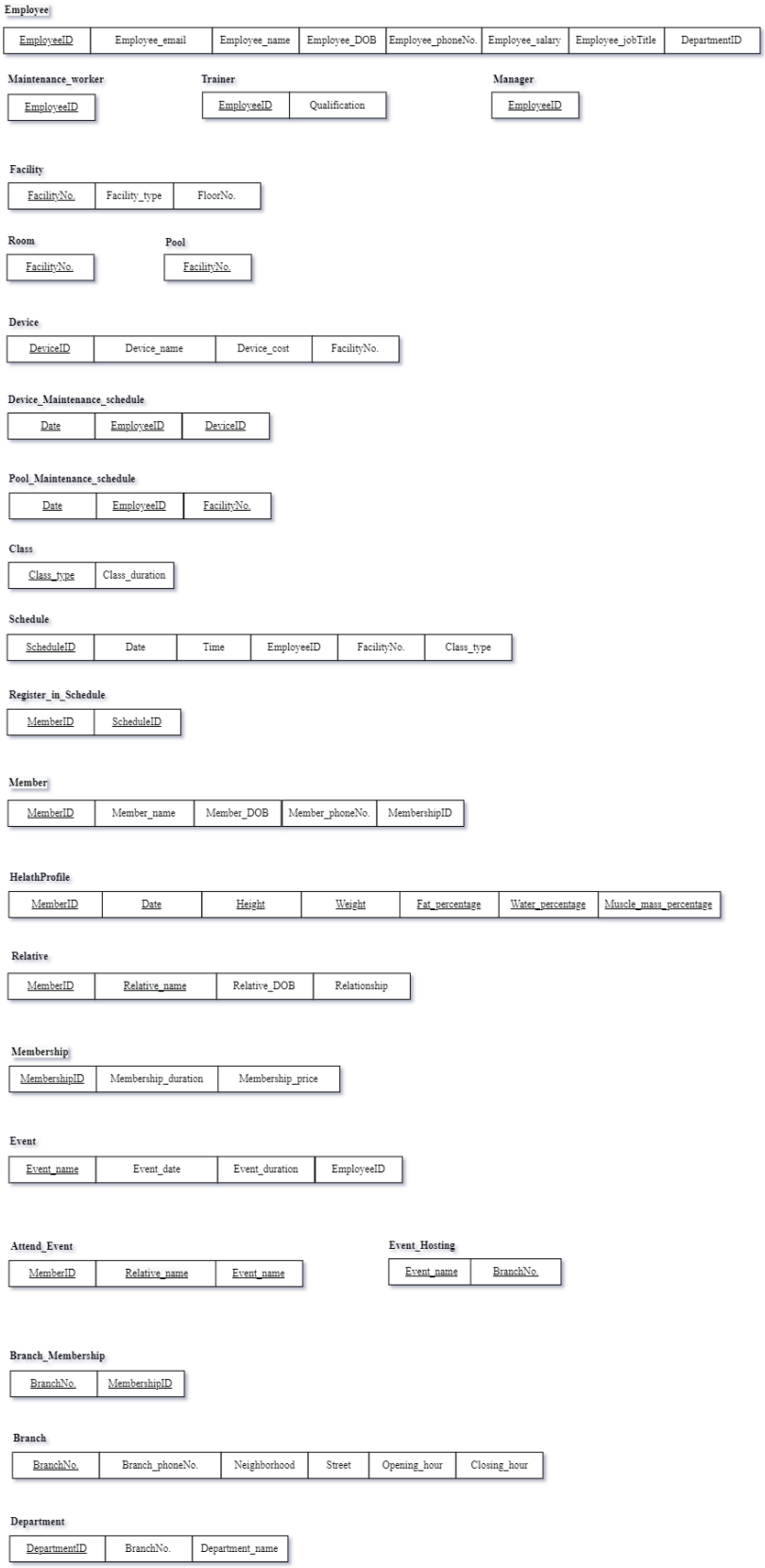


4. Normalization

4.1 First Normal Form

The first normal form disallows composite or multivalued attributes and nested relations. All tables are in the first normal form because they satisfy the above condition.

To see it more clearly follow the link: https://up6.cc/2022/11/166815149627371.png

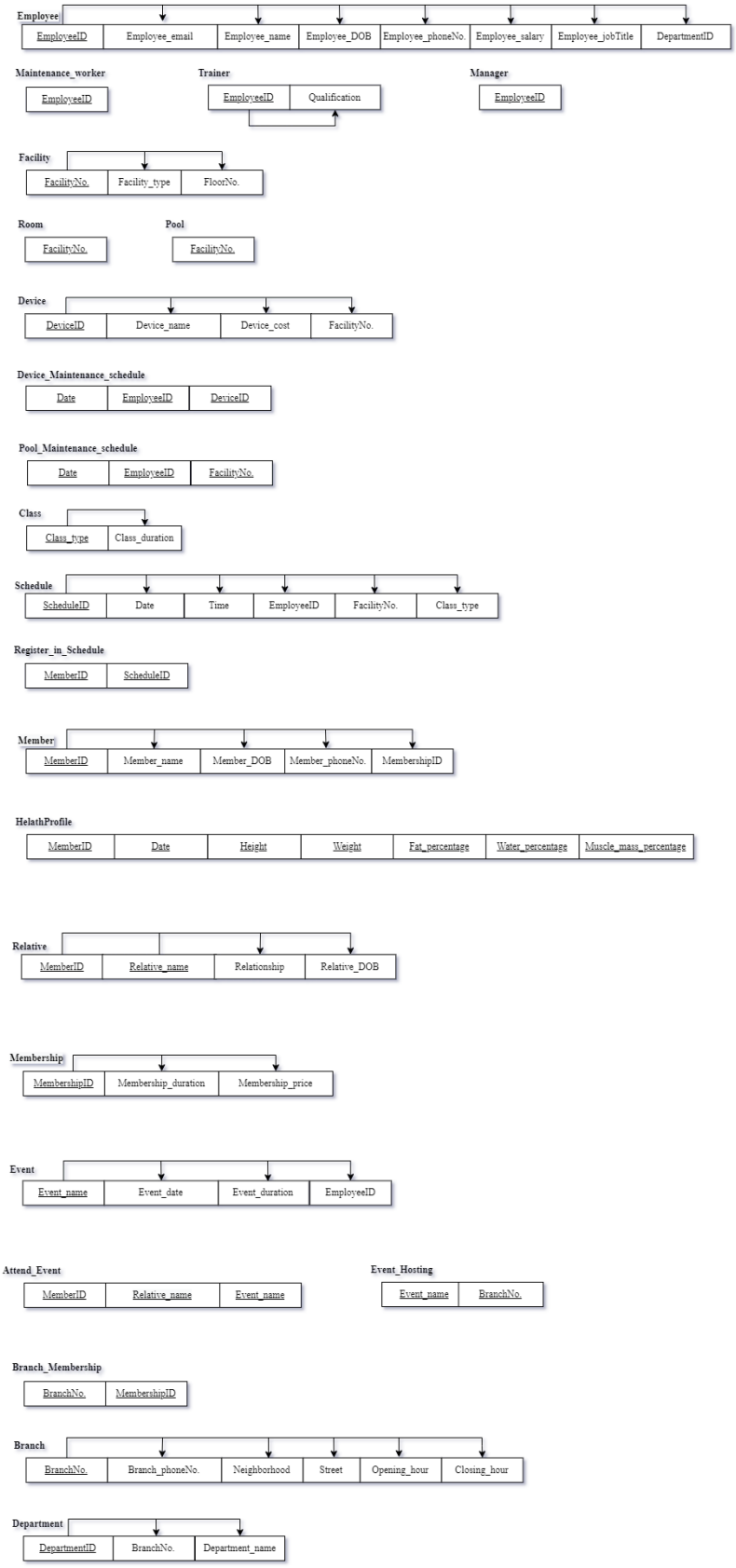


4.2 Second Normal Form

The second normal form disallows partial dependency, all attributes must be fully dependent on the primary key. All tables are on the second normal form because they satisfy the above condition.

The schema below represents all the attributes and their full functional dependence on the primary key.

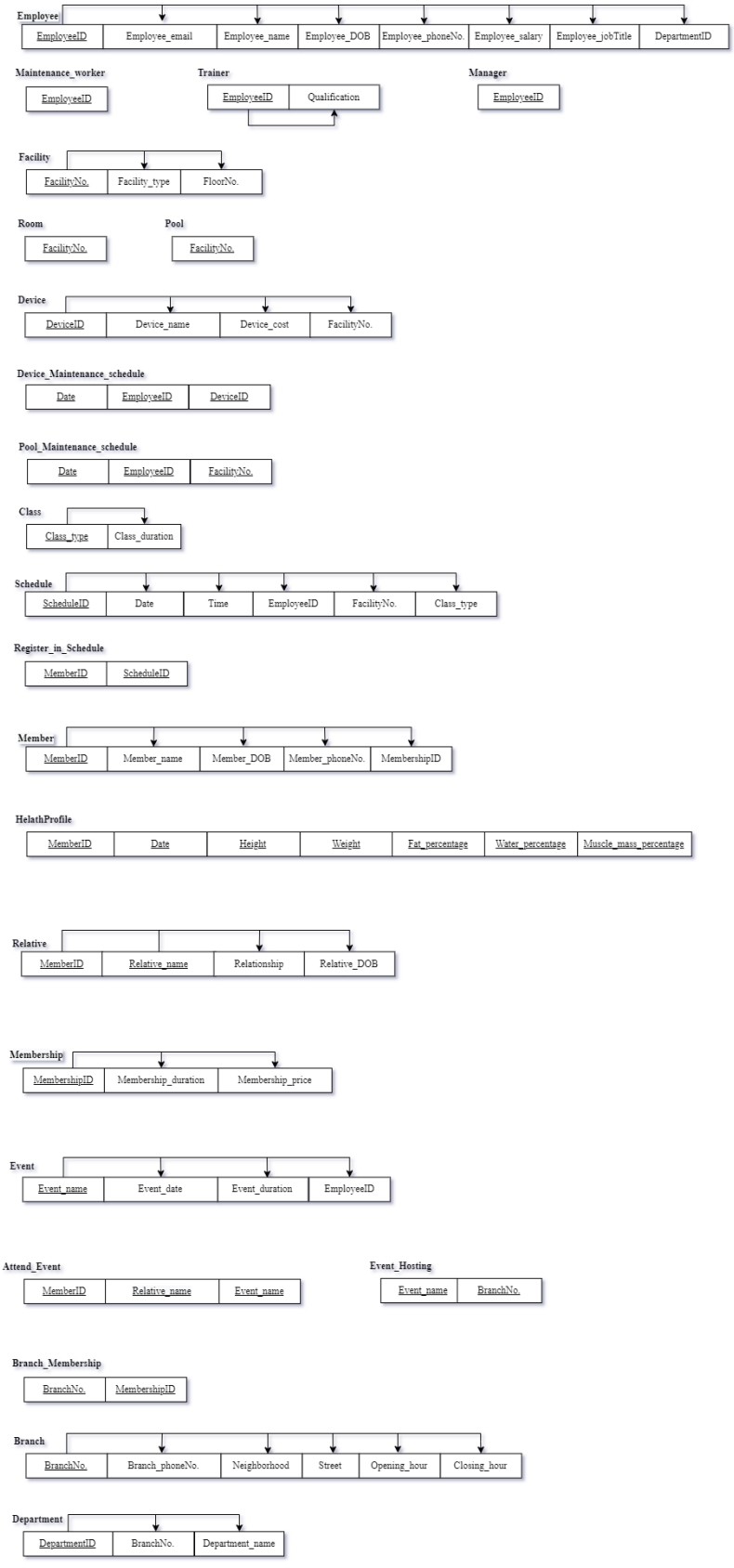
To see it more clearly follow the link: https://up6.cc/2022/11/166815168023811.png



4.3 Third Normal Form

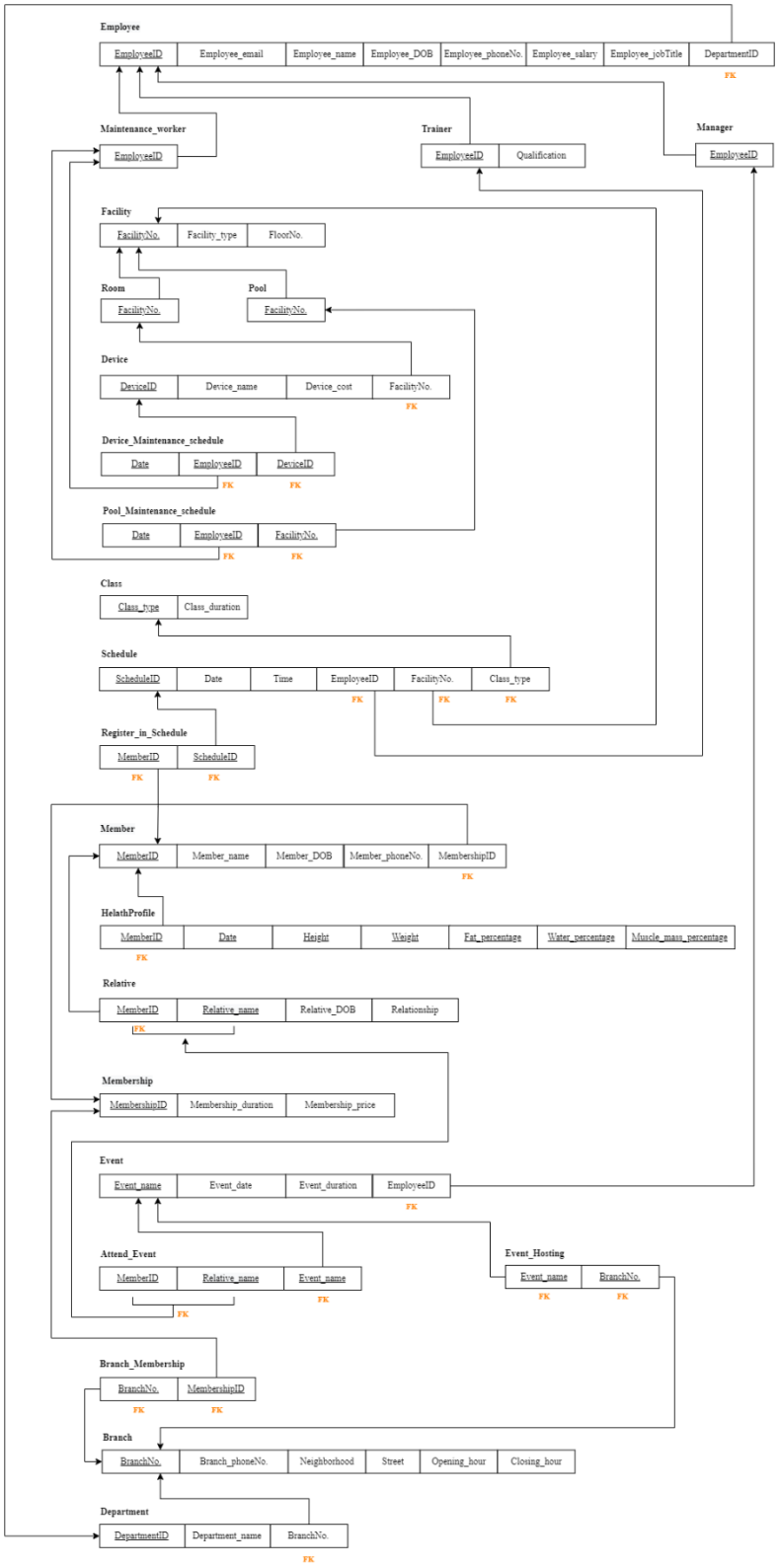
The third normal form disallows transitive functional dependency. That happens when a non-prime attribute in a relation is transitively dependent on another non-prime attribute. So, all non-prime attributes should be functionally dependent only on the primary key to apply the third normal form. All tables are in the third normal form because they satisfy the above condition.

To see it more clearly follow the link: https://up6.cc/2022/11/166815168023811.png



5. Final DB Schema Diagram

To see it more clearly follow the link: https://up6.cc/2022/11/16681510080613l.png



PART III: IMPLEMENTATION

6. Table Creation Script

6.1 <Branch> TABLE

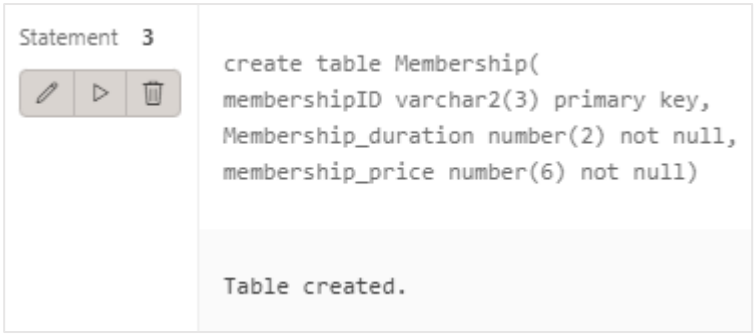
```
create table Branch (
branchNo number(3) primary key ,
branch_phoneNo number(10) ,
Neighborhood varchar2(15),
street varchar2(15),
opening_hour number(2),
closing_hour number(2));
```

6.2 <Department> TABLE

```
create table Department(
DepartmentID number(3) primary key,
Department_name varchar2(25),
BranchNo number(3),
constraint BrNo foreign key(BranchNo)
references Branch(BranchNo) on delete cascade );
```

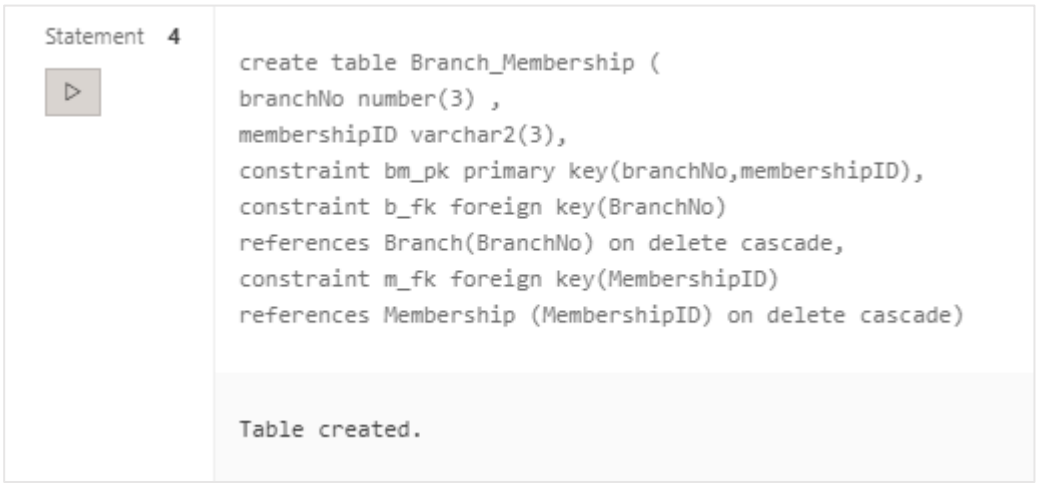
6.3 <Membership> TABLE

```
create table Membership(
membershipID varchar2(3) primary key,
Membership_duration number(2) not null,
membership_price number(6) not null);
```



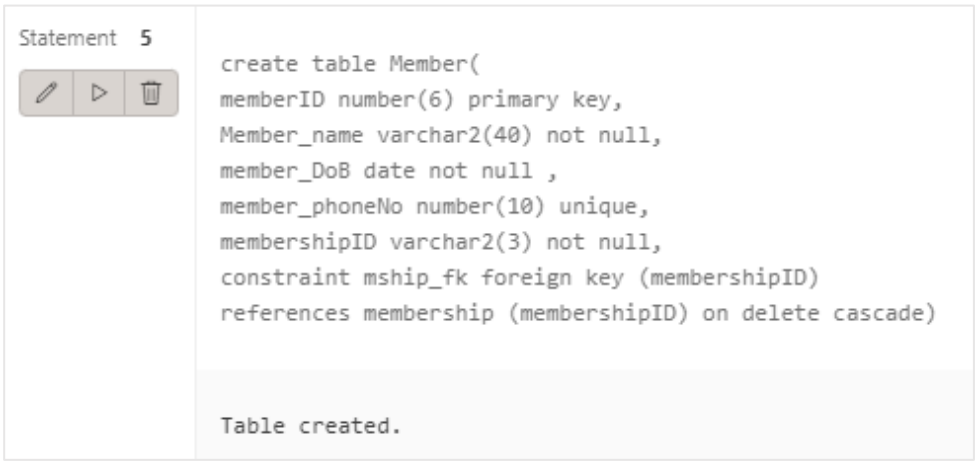
6.4 <Branch_Membership> TABLE

```
create table Branch_Membership (
branchNo number(3) ,
membershipID varchar2(3),
constraint bm_pk primary key(branchNo,membershipID),
constraint b_fk foreign key(BranchNo)
references Branch(BranchNo) on delete cascade,
constraint m_fk foreign key(MembershipID)
references Membership (MembershipID) on delete cascade);
```



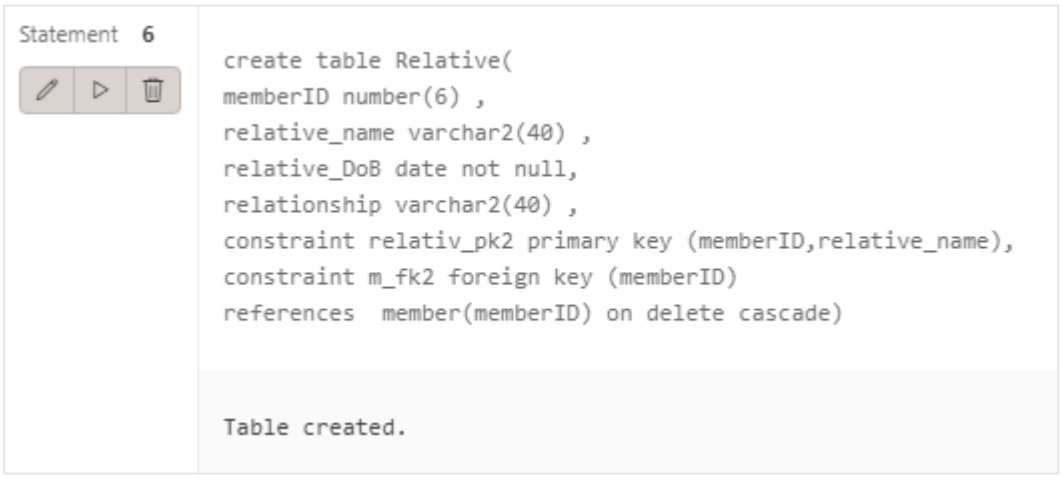
6.5 <Member> TABLE

```
create table Member(
memberID number(6) primary key,
Member_name varchar2(40) not null,
member_DoB date not null ,
member_phoneNo number(10) unique,
membershipID varchar2(3) not null,
constraint mship_fk foreign key (membershipID)
references membership (membershipID) on delete cascade);
```



6.6 <Relative> TABLE

```
create table Relative(
memberID number(6) ,
relative_name varchar2(40) ,
relative_DoB date not null,
relationship varchar2(40) ,
constraint relativ_pk2 primary key (memberID,relative_name),
constraint m_fk2 foreign key (memberID)
references  member(memberID) on delete cascade);
```

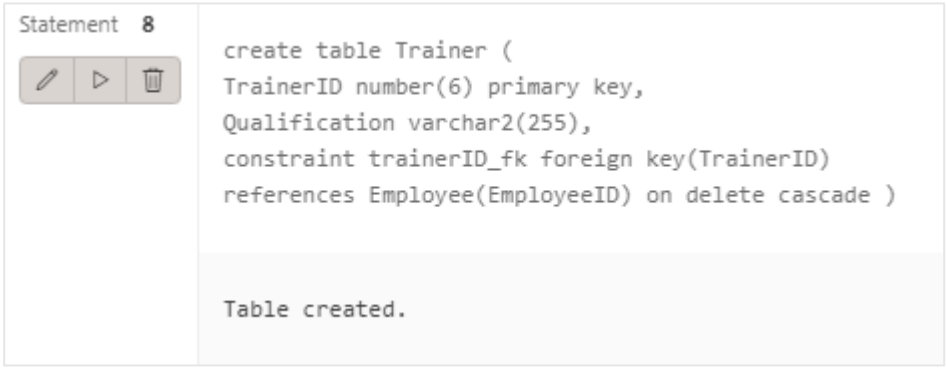


6.7 <Employee> TABLE

```
create table Employee (
EmployeeID number(6) primary key,
Employee_email varchar2(255),
Employee_name varchar2(50),
Employee_DOB Date,
Employee_phoneNo number(10),
Employee_salary decimal(9,2),
Employee_jobTitle varchar2(30) not null,
DepartmentID number(3),
constraint empDep_fk foreign key(DepartmentID)
references Department(DepartmentID) on delete cascade );
```

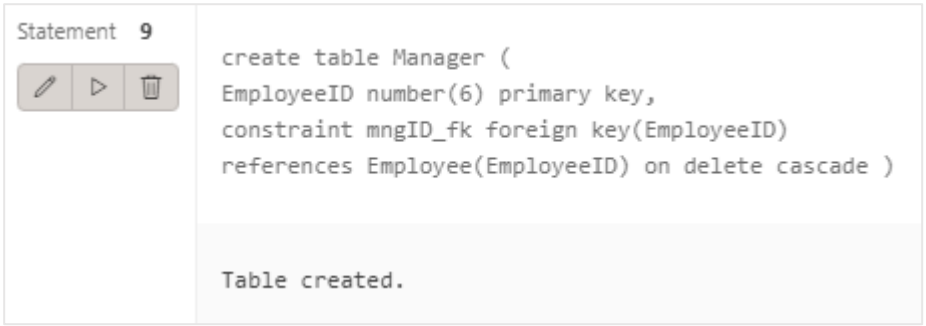
6.8 <Trainer> TABLE

```
create table Trainer (
TrainerID number(6) primary key,
Qualification varchar2(255),
constraint trainerID_fk foreign key(TrainerID)
references Employee(EmployeeID) on delete cascade );
```



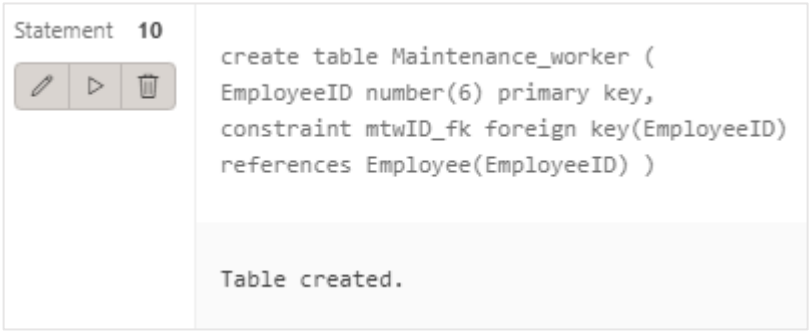
6.9 <Manager> TABLE

```
create table Manager (
EmployeeID number(6) primary key,
constraint mngID_fk foreign key(EmployeeID)
references Employee(EmployeeID) on delete cascade );
```



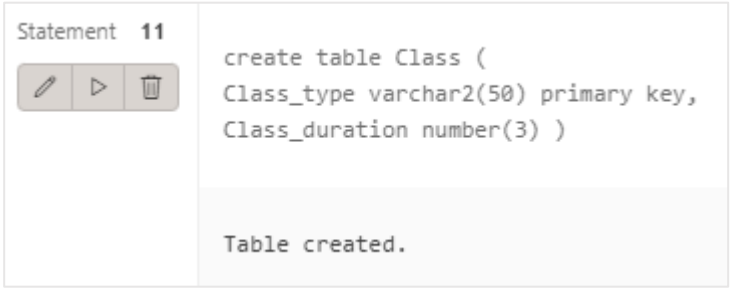
6.10 < Maintenance_worker > TABLE

```
create table Maintenance_worker (
EmployeeID number(6) primary key,
constraint mtwID_fk foreign key(EmployeeID)
references Employee(EmployeeID) );
```



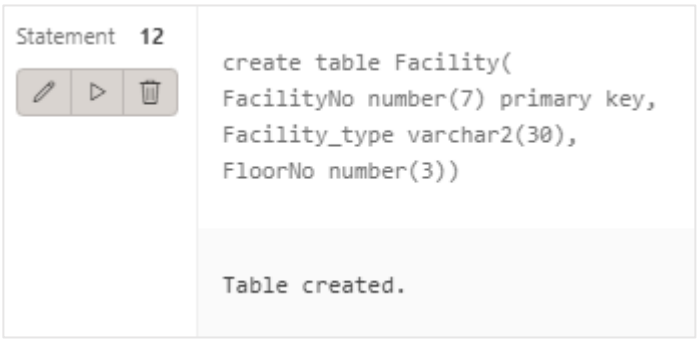
6.11 <Class> TABLE

```
create table Class (
Class_type varchar2(50) primary key,
Class_duration number(3) );
```



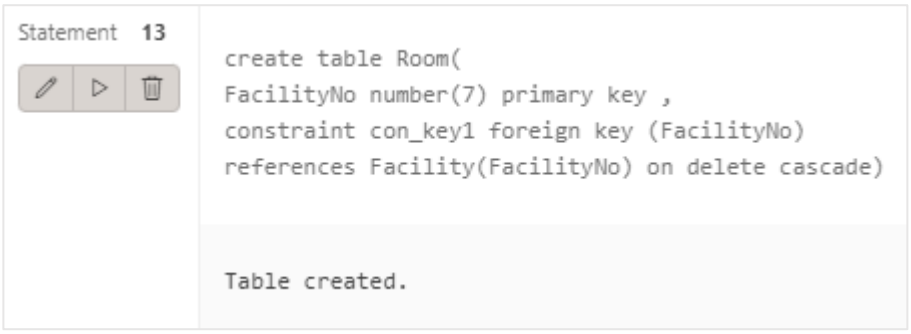
6.12 <Facility> TABLE

```
create table Facility(
FacilityNo number(7) primary key,
Facility_type varchar2(30),
FloorNo number(3));
```



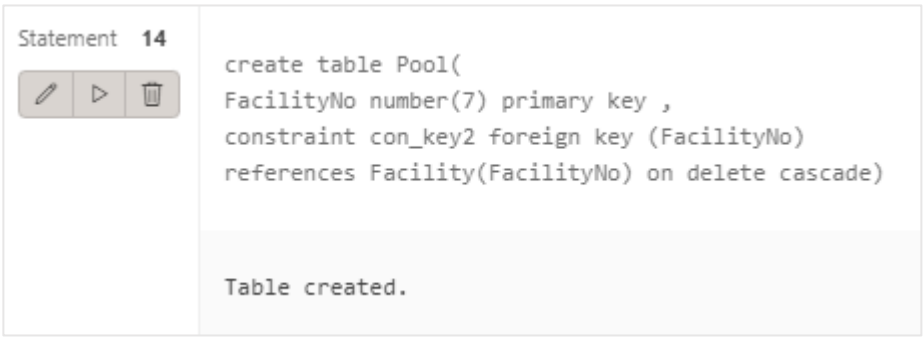
6.13 <Room> TABLE

```
create table Room(
FacilityNo number(7) primary key ,
constraint con_key1 foreign key (FacilityNo)
references Facility(FacilityNo) on delete cascade);
```



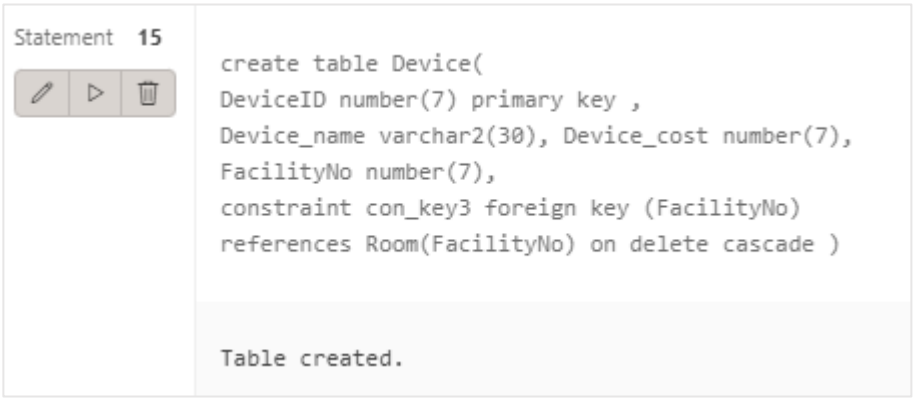
6.14 <Pool> TABLE

```
create table Pool(
FacilityNo number(7) primary key ,
constraint con_key2 foreign key (FacilityNo)
references Facility(FacilityNo) on delete cascade);
```



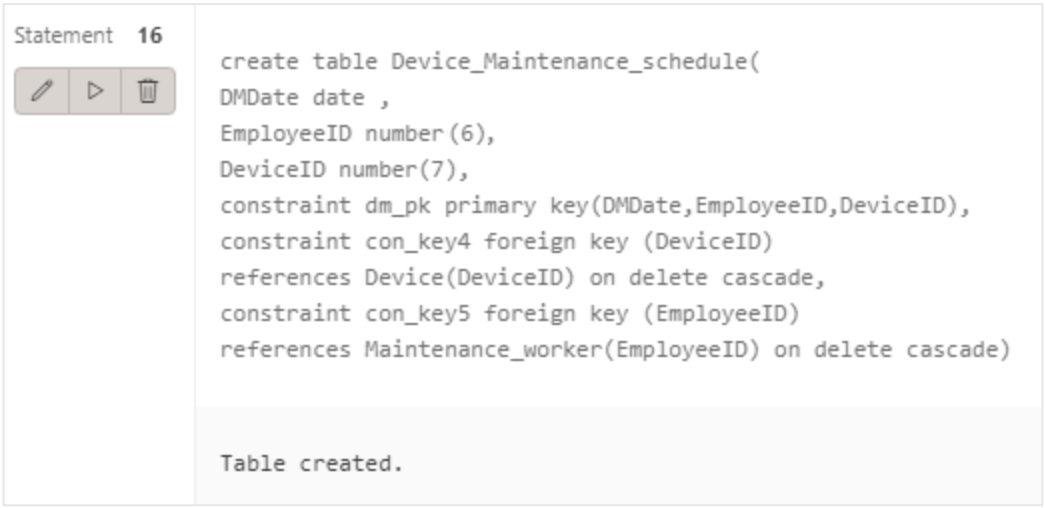
6.15 <Device> TABLE

```
create table Device(
DeviceID number(7) primary key ,
Device_name varchar2(30), Device_cost number(7),
FacilityNo number(7),
constraint con_key3 foreign key (FacilityNo)
references Room(FacilityNo) on delete cascade );
```



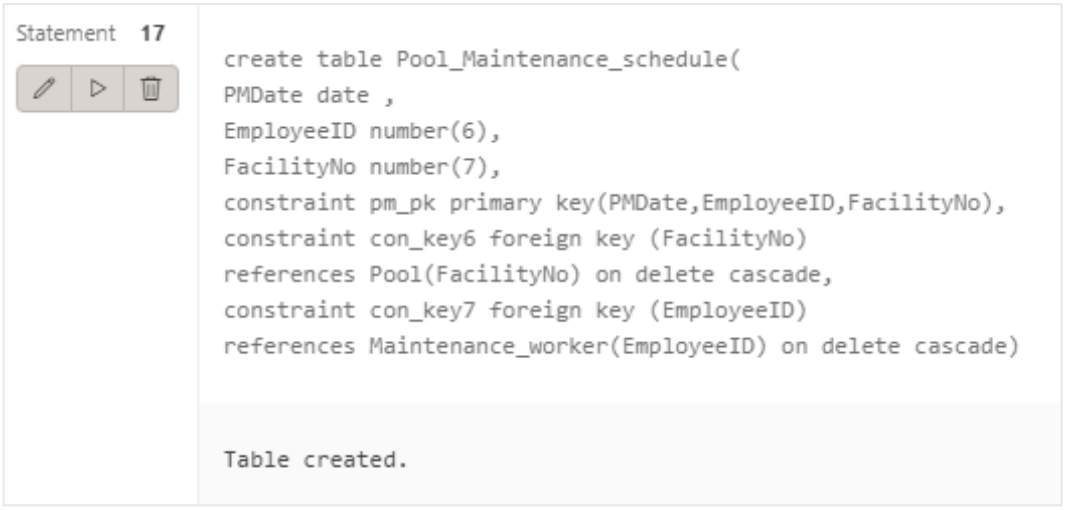
6.16 < Device_Maintenance_schedule > TABLE

```
create table Device_Maintenance_schedule(
DMDate date ,
EmployeeID number(6),
DeviceID number(7),
constraint dm_pk primary key(DMDate,EmployeeID,DeviceID),
constraint con_key4 foreign key (DeviceID)
references Device(DeviceID) on delete cascade,
constraint con_key5 foreign key (EmployeeID)
references Maintenance_worker(EmployeeID) on delete cascade);
```



6.17 < Pool_Maintenance_schedule > TABLE

```
create table Pool_Maintenance_schedule(

PMDate date ,

EmployeeID number(6),

FacilityNo number(7),

constraint pm_pk primary key(PMDate,EmployeeID,FacilityNo),

constraint con_key6 foreign key (FacilityNo)

references Pool(FacilityNo) on delete cascade,

constraint con_key7 foreign key (EmployeeID)

references Maintenance_worker(EmployeeID) on delete cascade);
```



6.18 < Schedule > TABLE

```
create table Schedule (
ScheduleID number(3) primary key,
Sdate Date,
Stime varchar2(8),
TrainerID number(6),
FacilityNo number(7),
Class_type varchar2(20),
constraint schTr_fk foreign key(TrainerID)
references Trainer(TrainerID),
constraint schFc_fk foreign key(FacilityNo)
references Facility(FacilityNo),
constraint schClass_fk foreign key(Class_type)
references Class(Class_type) );
```

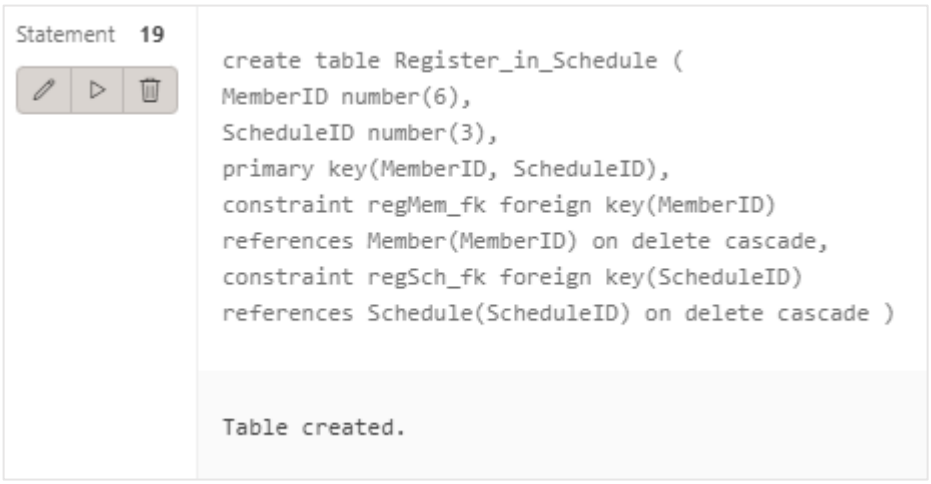
Statement 18

```
create table Schedule (
ScheduleID number(3) primary key,
Sdate Date,
Stime varchar2(8),
TrainerID number(6),
FacilityNo number(7),
Class_type varchar2(20),
constraint schTr_fk foreign key(TrainerID)
references Trainer(TrainerID),
constraint schFc_fk foreign key(FacilityNo)
references Facility(FacilityNo),
constraint schClass_fk foreign key(Class_type)
references Class(Class_type) )
```

```
Table created.
```

6.19 < Register_in_Schedule > TABLE

```
create table Register_in_Schedule (
MemberID number(6),
ScheduleID number(3),
primary key(MemberID, ScheduleID),
constraint regMem_fk foreign key(MemberID)
references Member(MemberID) on delete cascade,
constraint regSch_fk foreign key(ScheduleID)
references Schedule(ScheduleID) on delete cascade );
```

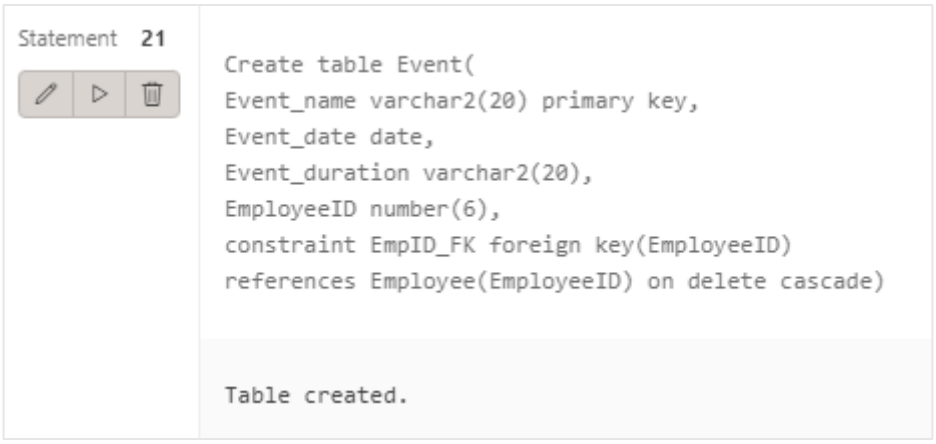


6.20 < HealthProfile > TABLE

```
Create table HealthProfile (

MemberID number(6),

HDate date ,

Height number(10) ,

Weight number(10) ,

Fat_precentage varchar2(15) ,

Water_precentage varchar2(15) ,

Muscle_mass_precentage varchar2(15)  ,

constraint MID foreign key (MemberID)

references Member(MemberID) on delete cascade,

constraint HPK primary key

(MemberID,HDate,Height,Weight,Fat_precentage,Water_precentage,Muscle_mass_
precentage) );
```

6.21 < Event > TABLE

```
Create table Event(
Event_name varchar2(20) primary key,
Event_date date,
Event_duration varchar2(20),
EmployeeID number(6),
constraint EmpID_FK foreign key(EmployeeID)
references Employee(EmployeeID) on delete cascade);
```



6.22 < Attend_Event> TABLE

Create table Attend_Event(MemberID number(6),

Relative_name varchar2(40),

Event_name varchar2(20) ,

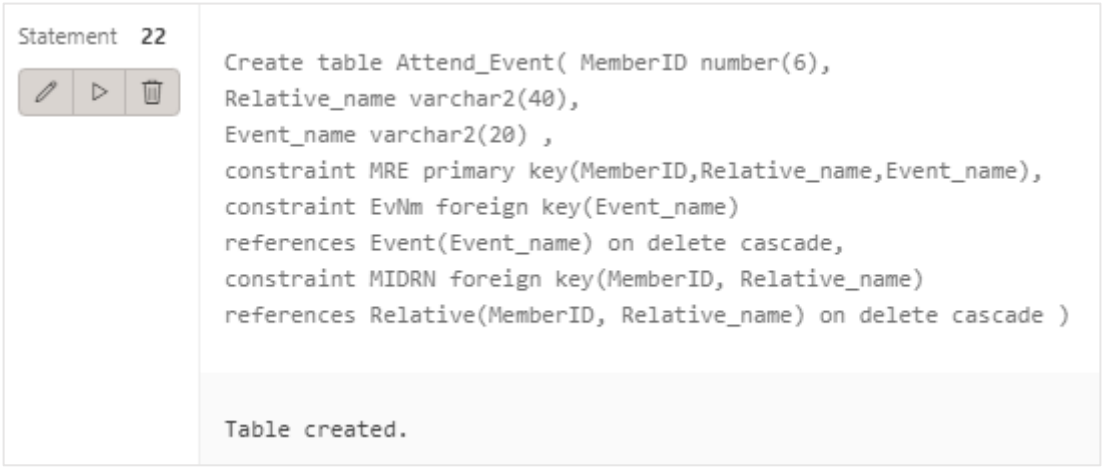
constraint MRE primary key(MemberID,Relative_name,Event_name),

constraint EvNm foreign key(Event_name)

references Event(Event_name) on delete cascade,

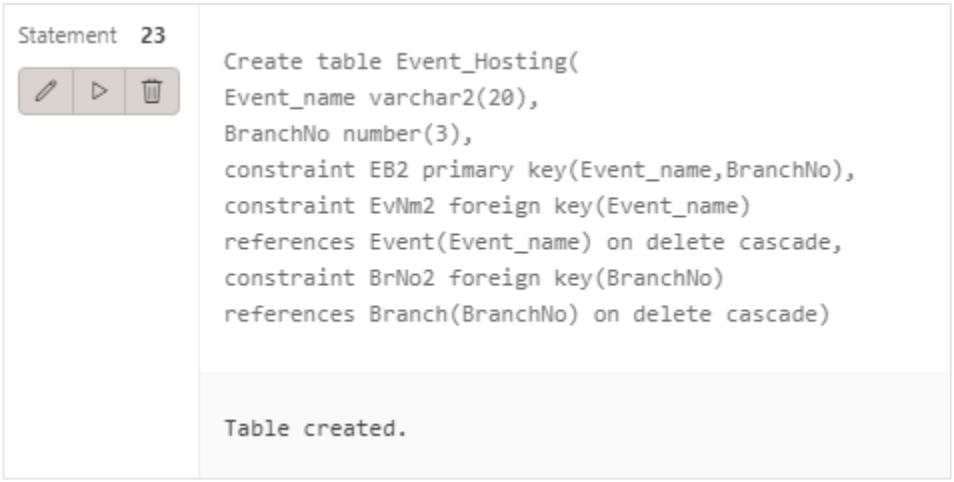
constraint MIDRN foreign key(MemberID, Relative_name)

references Relative(MemberID, Relative_name) on delete cascade);



6.23 < Event_Hosting > TABLE

```
Create table Event_Hosting(

Event_name varchar2(20),

BranchNo number(3),

constraint EB2 primary key(Event_name,BranchNo),

constraint EvNm2 foreign key(Event_name)

references Event(Event_name) on delete cascade,

constraint BrNo2 foreign key(BranchNo)

references Branch(BranchNo) on delete cascade);
```



7. Constraints Script

Business Rule	SQL Script	Table
No department can exist without a branch. Thus, if the branch is deleted, the department will be deleted as well.	constraint BrNo foreign key (BranchNo) references Branch (BranchNo) on delete cascade	Department
Each membership must have a duration.	Membership_duration number(2) not null	Membership
Each membership must have a price.	membership_price number(6) not null	Membership
Each member must have name.	Member_name varchar2(40) not null	Member
Each member must have a date of birth.	member_DoB date not null	Member
Each member must have a unique phone number.	member_phoneNo number(10) unique	Member
No one can become a member without a membership.	membershipID varchar2(3) not null, constraint mship_fk foreign key (membershipID) references membership (membershipID) on delete cascade	Member
Each relative must have a date of birth.	relative_DoB date not null	Relative
No need for relatives' information when the member is deleted.	constraint m_fk2 foreign key (memberID) references	Relative

	member(memberID) on delete cascade	
Each employee must have a specific job.	Employee_jobTitle varchar2(30) not null	Employee
Each employee must be assigned to a department.	constraint empDep_fk foreign key(DepartmentID) references Department(DepartmentID) on delete cascade	Employee
All devices must be placed in a room.	constraint con_key3 foreign key (FacilityNo) references Room(FacilityNo) on delete cascade	Device
Each device maintenance schedule contains devices to be maintained.	constraint con_key4 foreign key (DeviceID) references Device(DeviceID) on delete cascade	Device_Maintenance_schedule
Each device maintenance schedule must be assigned to a maintenance worker.	constraint con_key5 foreign key (EmployeeID) references Maintenance_worker(EmployeeID) on delete cascade	Device_Maintenance_schedule
Each pool maintenance schedule contains pools to be maintained.	constraint con_key6 foreign key (FacilityNo) references Pool(FacilityNo) on delete cascade	Pool_Maintenance_schedule
Each pool maintenance schedule must be assigned to a maintenance worker.	constraint con_key7 foreign key (EmployeeID) references Maintenance_worker(EmployeeID) on delete cascade)	Pool_Maintenance_schedule
Each schedule contains the ID of the trainer.	constraint schTr_fk foreign key(TrainerID) references Trainer(TrainerID)	Schedule

Each schedule contains the ID of the facility.	constraint schFc_fk foreign key(FacilityNo) references Facility(FacilityNo)	Schedule
Each schedule contains the type of the class.	constraint schClass_fk foreign key(Class_type) references Class(Class_type)	Schedule
Each member has health profiles that are updated over time.	constraint MID foreign key (MemberID) references Member(MemberID) on delete cascade	HealthProfile
Each event is managed by manager.	constraint EmpID_FK foreign key(EmployeeID) references Employee(EmployeeID) on delete cascade	Event

8. Queries

8.1 <Display trainers and number of schedules>

Query in natural language (ENGLISH):

List the trainers and the number of schedules they have provided

SQL Script:

```
SELECT EmployeeID AS TrainerID,
Employee_name AS Trainer_name,
COUNT(ScheduleID) AS Num_Of_Schedules
FROM Employee
INNER JOIN Trainer t ON EmployeeID = t.TrainerID
LEFT JOIN Schedule s
ON t.TrainerID = s.TrainerID
GROUP BY EmployeeID, Employee_name;
```

Caption of the output:

TRAINERID	TRAINER_NAME	NUM_OF_SCHEDULES
100038	Rina Sumait Al Mahyawy	0
100039	Dana Ayed Almaliki	0
100020	Rawady Maatouk Zafarani	7
100023	Ghayaa Sumait Almahyawi	5
100024	Rollina Saad Eddin Kattouh	1
100022	Fay Abdul Aziz Alhamimah	2
100037	Atheer Saad Aljahdali	0
100019	Rawan Yasser Aldossary	5
100021	Amy Jacob Russell	5
100040	Maha Abdul Aziz Alahiwi	0

Download CSV

10 rows selected.

8.2 <Display members and attended schedules>

Query in natural language (ENGLISH):

List the members who have attended more than 5 schedules

SQL Script:

```
SELECT m.MemberID, Member_name, COUNT(ScheduleID) Num_Of_Schedules
FROM Member m
INNER JOIN Register_in_Schedule r ON m.MemberID = r.MemberID
GROUP BY m.MemberID, Member_name having COUNT(ScheduleID) > 5;
```

Caption of the output:

MEMBERID	MEMBER_NAME	NUM_OF_SCHEDULES
110001	Suha Ali	6
110004	Foazia Hadi	6

Download CSV

2 rows selected.

8.3 <Display Device which maintained more than 2 >

Query in natural language (ENGLISH):

List the device which has been maintained more than 2 times

SQL Script:

```
SELECT d.DeviceID, d.Device_name, COUNT(dms.DeviceID)
FROM Device d
INNER JOIN Device_Maintenance_schedule dms
ON d.DeviceID = dms.DeviceID
GROUP BY d.DeviceID, d.Device_name having COUNT(dms.DeviceID) > 2;
```

Caption of the output:

DEVICEID	DEVICE_NAME	COUNT(DMS.DEVICEID)
200003	lat pull	3
200005	pull down	3
200004	back extention	3
200006	row	3
200002	biceps	3
100004	fly read delt	5
100003	overhead press	5
100002	triceps prees	5
100001	chest press	5
200001	arm curl	3
100005	arm extention	5

Download CSV

11 rows selected.

8.4 <Display the maintenance-worker with the highest number of maintenance schedules>

Query in natural language (ENGLISH):

Retrieve the maintenance-worker who have performed the highest number of maintenance schedules

SQL Script:

```
select EmployeeID as Maintenance_Worker_ID, COUNT(*) as Num_Of_Schedules
FROM Pool_Maintenance_schedule
GROUP BY EmployeeID having COUNT(*) = (
select MAX(num)
FROM (
select EmployeeID, COUNT(*) as num FROM Device_Maintenance_schedule GROUP BY EmployeeID
UNION
select EmployeeID, COUNT(*) as num FROM Pool_Maintenance_schedule GROUP BY EmployeeID
))
UNION
select EmployeeID, COUNT(*)
FROM Device_Maintenance_schedule
GROUP BY EmployeeID having COUNT(*) = (
select MAX(num)
FROM (
select EmployeeID, COUNT(*) as num FROM Device_Maintenance_schedule GROUP BY EmployeeID
UNION
select EmployeeID, COUNT(*) as num FROM Pool_Maintenance_schedule GROUP BY EmployeeID));
```

Caption of the output:

MAINTENANCE_WORKER_ID	NUM_OF_SCHEDULES
100032	21

Download CSV

8.5 <Display members and the attended classes with their trainer name>

Query in natural language (ENGLISH):

For each member, retrieve the name of the classes they attended and the name of the trainer

SQL Script:

```
SELECT ris.MemberID, m.Member_name, Class_type, Employee_name AS
Trainer_Name
FROM Member m
INNER JOIN Register_in_Schedule ris ON m.MemberID = ris.MemberID
INNER JOIN Schedule s ON ris.ScheduleID = s.ScheduleID
INNER JOIN Trainer t ON s.TrainerID = t.TrainerID
INNER JOIN Employee e ON e.EmployeeID = t.TrainerID;
```

Caption of the output:

MEMBERID	MEMBER_NAME	CLASS_TYPE	TRAINER_NAME
110001	Suha Ali	CARDIO	Rawan Yasser Aldossary
110001	Suha Ali	ZUMBA	Rawady Maatouk Zafarani
110001	Suha Ali	UPPER BODY	Ghayaa Sumait Almahyawi
110001	Suha Ali	FULL BODY	Rawan Yasser Aldossary
110001	Suha Ali	CONDITIONING	Rawady Maatouk Zafarani
110001	Suha Ali	PILATES	Rawady Maatouk Zafarani
110002	Sara Ahmed	BODYCOMBAT	Amy Jacob Russell
110002	Sara Ahmed	AQUA ZUMBA	Fay Abdul Aziz Alhamimah
110002	Sara Ahmed	UPPER BODY	Ghayaa Sumait Almahyawi
110002	Sara Ahmed	SPRINT	Amy Jacob Russell
110003	Huda Mohammed	SPRINT	Amy Jacob Russell
110003	Huda Mohammed	SPINNING	Rawady Maatouk Zafarani
110003	Huda Mohammed	ABS	Rawan Yasser Aldossary
110003	Huda Mohammed	STRONG ABS	Rollina Saad Eddin Kattouh
110004	Foazia Hadi	BODYPUMP	Rawan Yasser Aldossary
110004	Foazia Hadi	BODYCOMBAT	Amy Jacob Russell
110004	Foazia Hadi	BOOTCAMP	Rawady Maatouk Zafarani
110004	Foazia Hadi	CIRCUIT	Amy Jacob Russell
110004	Foazia Hadi	AQUA AEROBICS	Fay Abdul Aziz Alhamimah
110004	Foazia Hadi	PILATES	Rawady Maatouk Zafarani
110005	Lana Karem	AQUA ZUMBA	Fay Abdul Aziz Alhamimah
110005	Lana Karem	AQUA AEROBICS	Fay Abdul Aziz Alhamimah
110006	Judy Yaser	CARDIO	Rawan Yasser Aldossary
110006	Judy Yaser	BODYPUMP	Rawan Yasser Aldossary
110006	Judy Yaser	SPINNING	Rawady Maatouk Zafarani
110006	Judy Yaser	BODYATTACK	Ghayaa Sumait Almahyawi
110007	Eenas Fared	UPPER BODY	Ghayaa Sumait Almahyawi
110007	Eenas Fared	STRONG ABS	Rollina Saad Eddin Kattouh
110007	Eenas Fared	BODYBALANCE	Ghayaa Sumait Almahyawi
110008	Nada Nader	CARDIO	Rawan Yasser Aldossary
110008	Nada Nader	ZUMBA	Rawady Maatouk Zafarani
110008	Nada Nader	BODYCOMBAT	Rawan Yasser Aldossary
110009	Alaa Hamza	BODYPUMP	Rawan Yasser Aldossary
110009	Alaa Hamza	FULL BODY	Rawady Maatouk Zafarani
110009	Alaa Hamza	BOOTCAMP	Rawady Maatouk Zafarani
110009	Alaa Hamza	SPRINT	Ghayaa Sumait Almahyawi
110010	Minnie Mouse	ABS	Rawan Yasser Aldossary
110010	Minnie Mouse	CONDITIONING	Rawady Maatouk Zafarani
110011	Cinderella	FULL BODY	Rawan Yasser Aldossary
110011	Cinderella	CARDIO	Amy Jacob Russell
110011	Cinderella	STRONG ABS	Rollina Saad Eddin Kattouh
110011	Cinderella	AQUA AEROBICS	Fay Abdul Aziz Alhamimah
110012	Snow White	ZUMBA	Rawady Maatouk Zafarani
110012	Snow White	BODYCOMBAT	Rawan Yasser Aldossary
110012	Snow White	PILATES	Rawady Maatouk Zafarani
110013	Ariel Mermaid	CIRCUIT	Amy Jacob Russell
110013	Ariel Mermaid	BODYATTACK	Ghayaa Sumait Almahyawi
110014	Jasmine Ali	CARDIO	Rawan Yasser Aldossary
110014	Jasmine Ali	FULL BODY	Rawady Maatouk Zafarani
110014	Jasmine Ali	AQUA ZUMBA	Fay Abdul Aziz Alhamimah

Download CSV

Rows 1 – 50. More rows exist.

9. Appendix

<Branch> Table

BRANCHNO	BRANCH_PHONENO	NEIGHBORHOOD	STREET	OPENING_HOUR	CLOSING_HOUR
1	558877665	Al-Safa	Sun top Street	7	24
2	566778899	Al-Marwah	Sushi Street	6	24
3	505068686	Al-Rehab	Pizza Street	7	23

Download CSV

3 rows selected.

<Department> Table

DEPARTMENTID	DEPARTMENT_NAME	BRANCHNO
1	Training Department	1
2	Management Department	1
3	Maintenance Department	1
4	Cleaning Department	1
5	Reception Department	1
6	Training Department	2
7	Management Department	2
8	Maintenance Department	2
9	Cleaning Department	2
10	Reception Department	2
11	Training Department	3
12	Management Department	3
13	Maintenance Department	3
14	Cleaning Department	3
15	Reception Department	3

Download CSV

15 rows selected.

<Member> Table

MEMBERID	MEMBER_NAME	MEMBER_DOB	MEMBER_PHONENO	MEMBERSHIPID
110001	Suha Ali	09-JAN-90	556678362	B06
110002	Sara Ahmed	02-JUN-85	504482552	B01
110003	Huda Mohammed	25-MAR-98	559834211	U06
110004	Foazia Hadi	07-NOV-95	507773549	B03
110005	Lana Karem	18-SEP-87	551287453	B12
110006	Judy Yaser	29-JUL-96	506657892	U03
110007	Eenas Fared	18-DEC-75	533683923	U12
110008	Nada Nader	14-JAN-88	550012340	U06
110009	Alaa Hamza	03-FEB-91	501235738	B03
110010	Minnie Mouse	19-JUN-99	556799139	U03
110011	Cinderella	17-APR-03	555588722	U06
110012	Snow White	26-AUG-79	551116537	U03
110013	Ariel Mermaid	15-JUN-99	506673362	U06
110014	Jasmine Ali	23-JAN-04	520164476	U12
110015	Pocahontas	20-OCT-89	551298765	U03
110016	Rapunzel	13-MAR-97	554466536	U06
110017	Bella Beast	22-AUG-00	551226544	U03
110018	Amani Rami	05-DEC-95	507553549	B06
110019	Mona Suhail	17-MAY-01	505584635	B12
110020	Sara Kalid	12-APR-85	503377936	B01

Download CSV

20 rows selected.

<Membership> Table

MEMBERSHIPID	MEMBERSHIP_DURATION	MEMBERSHIP_PRICE
B01	1	350
B03	3	950
B06	6	1800
B12	12	3500
U03	3	1050
U06	6	2000
U12	12	3850

Download CSV

7 rows selected.

<Event> Table

EVENT_NAME	EVENT_DATE	EVENT_DURATION	EMPLOYEEID
Zumba day	15-MAY-20	1 day	100000
Family Workout	20-SEP-21	2 days	100001
Cardio competition	07-NOV-22	3 days	100002
Dance the Night	09-OCT-21	1 day	100003
Strength is Key	18-JUL-20	1 day	100004

Download CSV

5 rows selected.

<Facility> Table

FACILITYNO	FACILITY_TYPE	FLOORNO
111	Exercise Room	1
112	Exercise Room	1
113	Exercise Room	2
114	Exercise Room	2
115	Rest Room	1
116	Rset Room	2
117	Sauna Room	3
118	Changing Room	3
119	Rest Room	3
1111	Pool	4
1112	Pool	4

Download CSV

11 rows selected.

<Room > Table

FACILITYNO
111
112
113
114
115
116
117
118
119

Download CSV

9 rows selected.

<Pool> Table

FACILITYNO
1111
1112

Download CSV

2 rows selected.

<Device> Table

DEVICEID	DEVICE_NAME	DEVICE_COST	FACILITYNO
100001	chest press	3500	111
100002	triceps prees	4600	111
100003	overhead press	2950	111
100004	fly read delt	5520	111
100005	arm extention	3110	111
100006	lateral raise	3500	111
100007	pectoral fly	7310	111
200001	arm curl	7560	111
200002	biceps	6300	111
200003	lat pull	6400	111
200004	back extention	8900	111
200005	pull down	9999	111
200006	row	5560	111
200007	torso rotation	7700	111
200008	abdominal	9200	111
300001	standing calf	6300	111
300002	seated calves	9360	111
300003	leg press	7900	111
300004	leg extention	4900	111
300005	seated leg curl	8999	111

Download CSV

20 rows selected.

<Employee> Table

EMPLOYEEID	EMPLOYEE_EMAIL	EMPLOYEE_NAME	EMPLOYEE_DOB	EMPLOYEE_PHONENO	EMPLOYEE_SALARY	EMPLOYEE_JOBTITLE	DEPARTMENTID
100000	albarakatiRahaf@pump.h.com	Rahaf Hashem Albarakati	13-MAY-80	547748983	15230	Manager	2
100001	aljuhaniSara@pump.h.com	Sarah Mahdi Aljuhani	03-SEP-77	594670203	15000	Manager	7
100002	alsolamiBashayer@pump.h.com	Bashayer Marzouk Alsolami	17-JUL-83	541783483	15000	Manager	12
100003	alsharifGhaida@pump.h.com	Ghaida Mohsen Alsharif	29-JAN-86	581720325	12550	Co Manager	2
100004	alsolamiGhada@pump.h.com	Ghada Fazaa Alsolami	16-FEB-88	514838011	12000	Co Manager	7
100005	khanAya@pump.h.com	Aya Issa Khan	09-AUG-90	596258695	12000	Co Manager	12
100006	moqremShahd@pump.h.com	Shahd Saeed Moqrem	17-JUN-78	509778603	5340	Sales Agent	2
100007	alsalamiLujain@pump.h.com	Lujain Saeed Alsalami	19-SEP-83	527671834	5340	Sales Agent	7
100008	almalikiReem@pump.h.com	Reem Said Almaliki	28-MAR-82	584547217	10400	Front Office Manager	2
100009	nahlawiSeham@pump.h.com	Seham Hassan Nahlawi	07-DEC-88	525554620	10400	Front Office Manager	7
100010	miyajanNaila@pump.h.com	Naila Asaad Miyajan	11-OCT-90	506967499	10450	Front Office Manager	12
100011	alkhomaniAbrar@pump.h.com	Abrar Rajeh Alkhomani	21-JAN-80	594410111	20500	Marketing Manager	2
100012	alghamdiRima@pump.h.com	Rima Ahmed Alghamdi	25-MAR-79	572593898	20500	Marketing Manager	12
100013	alghamdiDima@pump.h.com	Dima Ali Alghamdi	14-JUL-89	503914686	5300	Receptionist	5
100014	alsaqqafZainab@pump.h.com	Zainab Ali Alsaqqaf	01-FEB-91	548728329	5440	Receptionist	10
100015	alqarniRenad@pump.h.com	Renad Othman Alqarni	01-DEC-95	558202607	5300	Receptionist	15
100016	bafarhanAlia@pump.h.com	Alia Abdullah Bafarhan	04-APR-96	596379167	5800	Facility Supervisor	3
100017	baothmanSabah@pump.h.com	Sabah Abdullah Baothman	15-JUN-89	539926335	6000	Facility Supervisor	8
100018	koshakRahaf@pump.h.com	Rahaf Abdul Karim Koshak	22-MAR-92	529720848	6000	Facility Supervisor	13
100019	aldossaryRawan@pump.h.com	Rawan Yasser Aldossary	25-MAR-92	559515552	7500	Trainer	1
100020	zafaraniRawady@pump.h.com	Rawady Maatouk Zafarani	20-NOV-94	559515552	6850	Trainer	1
100021	russelAmy@pump.h.com	Amy Jacob Russell	13-AUG-96	588301987	7200	Trainer	1
100022	alhamimahFay@pump.h.com	Fay Abdul Aziz Alhamimah	20-FEB-94	501409836	6700	Trainer	1
100023	almahyawiGhayaa@pump.h.com	Ghayaa Sumait Almahyawi	11-OCT-95	559861601	7000	Trainer	1
100024	kattouhRollina@pump.h.com	Rollina Saad Eddin Kattouh	17-FEB-96	544227254	6850	Trainer	1
100025	hallSoraya@pump.h.com	Soraya Takeo Hall	11-OCT-88	508158738	2000	Cleaning Staff	4
100026	KhatanaMubeena@pump.h.com	Mubeena Bavand Khatana	04-SEP-85	532364534	2000	Cleaning Staff	4
100027	maherShaamila@pump.h.com	Shaamila Yahsi Mahar	15-OCT-87	537886713	1900	Cleaning Staff	9
100028	lodhraSameera@pump.h.com	Sameera Faatih Lodhra	20-MAR-80	543129733	1850	Cleaning Staff	9
100029	dilazakCamelia@pump.h.com	Camelia Niusha Dilazak	18-APR-95	504316392	1850	Cleaning Staff	14
100030	alonzoZoey@pump.h.com	Zoey Osana Alonzo	22-JUN-82	525230629	1900	Cleaning Staff	14
100031	fataniSamaher@pump.h.com	Samaher Abdel Rashid Fatani	09-JUN-79	562235730	3500	Maintenance Worker	3
100032	fataniSamaher@pump.h.com	Samaher Abdel Rashid Fatani	09-JUN-79	562235730	3500	Maintenance Worker	3
100033	namankaniHana@pump.h.com	Hana khalid Namankani	12-AUG-80	520651271	3000	Maintenance Worker	3
100034	brikAmani@pump.h.com	Amani Mohamed Brik	21-MAY-96	563724573	3000	Maintenance Worker	8
100035	salamaSeba@pump.h.com	Saba Mohamed Salama	06-MAY-90	541033860	3500	Maintenance Worker	8
100036	almalkiBayan@pump.h.com	Bayan Muhammad Almaliki	16-AUG-90	565640664	3500	Maintenance Worker	13
100037	aljahdalAtheer@pump.h.com	Atheer Saad Aljahdali	27-JUN-94	521323420	7500	Trainer	6
100038	almahyawyRina@pump.h.com	Rina Sumait Al Mahyawy	06-MAY-90	584962588	7200	Trainer	6
100039	almalkiDana@pump.h.com	Dana Ayed Almaliki	11-DEC-92	504386479	6700	Trainer	11
100040	alahiwiMaha@pump.h.com	Maha Abdul Aziz Alahiwi	07-OCT-89	588989903	7000	Trainer	11

Download CSV

41 rows selected.

<Branch_Membership> Table

BRANCHNO	MEMBERSHIPID
1	B01
1	B03
1	B06
1	B12
1	U03
1	U06
1	U12
2	B03
2	B06
2	B12
2	U03
2	U06
2	U12
3	B03
3	B06
3	B12
3	U03
3	U06

Download CSV

18 rows selected.

<Relative > Table

MEMBERID	RELATIVE_NAME	RELATIVE_DOB	RELATIONSHIP
110010	Susu Mouse	21-MAR-97	Cousin
110010	Daisy Duck	23-NOV-99	Friend
110003	Hanan Mohammed	25-FEB-95	Sister
110003	Hanaa Mahmoud	25-MAR-98	Cousin
110014	Jaen Ali	23-SEP-01	Sister
110014	Gigi Hani	23-JAN-04	Friend
110006	Jana Yaser	29-JUL-98	Sister
110006	Jain Yaser	29-JUL-94	Sister
110007	Amani Fared	18-DEC-67	Sister
110007	Roba Helo	04-DEC-74	Friend
110008	Nadeen Naser	09-SEP-88	Friend
110011	Nafesa	12-JUN-98	Sister

Download CSV

12 rows selected.

<Manager > Table

EMPLOYEEID
100000
100001
100002
100003
100004
100005
100008
100009
100010
100011
100012

Download CSV

11 rows selected.

<Trainer> Table

TRAINERID	QUALIFICATION
100019	Graduated from UK leading level 2 gym instructor course
100020	Three years experience as a sports trainer
100021	Bachelor’s degree in exercise science
100022	Diploma with 2 months experience
100023	Fitness Instructor Certification
100024	Assistant Fitness Instructor with 2 years experience
100037	Two years experience as a sports trainer
100038	NASM certificate
100039	Diploma degree in exercise science
100040	Fitness Instructor Certification

Download CSV

10 rows selected.

<Class> Table

CLASS_TYPE	CLASS_DURATION
CARDIO	45
FULL BODY	45
CIRCUIT	30
BOOTCAMP	45
AQUA AEROBICS	55
HIIT	30
BODYPUMP	60
AQUA ZUMBA	30
SPINNING	30
STRONG ABS	55
UPPER BODY	30
CORE	30
ZUMBA	45
SPRINT	30
BODYCOMBAT	55
BODYATTACK	55
CONDITIONING	30
PILATES	45
ABS	30
BODYBALANCE	60

Download CSV

20 rows selected.

<Maintenance_worker > Table

EMPLOYEEID
100031
100032
100033
100034
100035
100036

Download CSV

6 rows selected.

<HealthProfile > Table

MEMBERID	HDATE	HEIGHT	WEIGHT	FAT_PRECENTAGE	WATER_PRECENTAGE	MUSCLE_MASS_PRECENTAGE
110001	05-SEP-20	158	52	21%	45%	25%
110002	19-DEC-21	153	43	18%	45%	23%
110003	22-JAN-20	161	59	28%	55%	30%
110004	06-DEC-21	156	65	25%	50%	21%
110005	07-JUL-22	158	52	21%	45%	25%
110006	12-JUN-21	155	48	27%	53%	30%
110007	22-JUL-22	165	58	23%	50%	26%
110008	22-MAY-20	156	55	29%	57%	31%
110009	27-JAN-20	157	56	26%	52%	27%
110010	25-SEP-21	161	51	21%	50%	21%
110011	17-MAR-22	159	49	25%	47%	23%
110012	16-AUG-22	162	64	35%	60%	30%
110013	15-AUG-21	160	54	21%	50%	28%
110014	19-NOV-20	158	67	18%	45%	25%
110015	13-FEB-22	160	58	28%	53%	26%
110016	14-MAR-21	157	75	31%	54%	36%
110017	08-MAY-21	155	59	25%	51%	23%
110018	23-NOV-22	148	70	19%	60%	35%
110019	03-OCT-22	165	53	22%	55%	22%
110020	20-APR-20	153	66	24%	52%	32%

Download CSV

20 rows selected.

<Attend_Event> Table

MEMBERID	RELATIVE_NAME	EVENT_NAME
110003	Hanaa Mahmoud	Dance the Night
110003	Hanaa Mahmoud	Family Workout
110003	Hanaa Mahmoud	Strength is Key
110003	Hanan Mohammed	Dance the Night
110003	Hanan Mohammed	Family Workout
110003	Hanan Mohammed	Zumba day
110006	Jain Yaser	Cardio competition
110006	Jain Yaser	Family Workout
110006	Jain Yaser	Strength is Key
110006	Jain Yaser	Zumba day
110006	Jana Yaser	Dance the Night
110006	Jana Yaser	Family Workout
110006	Jana Yaser	Strength is Key
110007	Amani Fared	Cardio competition
110007	Amani Fared	Strength is Key
110007	Amani Fared	Zumba day
110007	Roba Helo	Cardio competition
110007	Roba Helo	Family Workout
110007	Roba Helo	Strength is Key
110008	Nadeen Naser	Cardio competition
110008	Nadeen Naser	Family Workout
110008	Nadeen Naser	Strength is Key
110008	Nadeen Naser	Zumba day
110010	Daisy Duck	Cardio competition
110010	Daisy Duck	Dance the Night
110010	Daisy Duck	Family Workout
110010	Daisy Duck	Strength is Key
110010	Daisy Duck	Zumba day
110010	Susu Mouse	Cardio competition
110010	Susu Mouse	Dance the Night
110010	Susu Mouse	Family Workout
110010	Susu Mouse	Zumba day
110011	Nafesa	Cardio competition
110011	Nafesa	Family Workout
110011	Nafesa	Strength is Key
110011	Nafesa	Zumba day
110014	Gigi Hani	Dance the Night
110014	Gigi Hani	Family Workout
110014	Gigi Hani	Strength is Key
110014	Gigi Hani	Zumba day
110014	Jaen Ali	Dance the Night
110014	Jaen Ali	Family Workout
110014	Jaen Ali	Zumba day

Download CSV

43 rows selected.

<Event_Hosting> Table

EVENT_NAME	BRANCHNO
Dance the Night	2
Family Workout	1
Family Workout	3
Zumba day	1
Zumba day	2

Download CSV

5 rows selected.

<Device_Maintenance_schedule> Table

DMDATE	EMPLOYEEID	DEVICEID
12-JUN-21	100031	200002
12-JUN-21	100031	200003
12-JUN-21	100031	200004
12-JUN-21	100031	200008
12-JUN-21	100031	300002
12-JUN-21	100031	300003
12-JUN-21	100032	100001
12-JUN-21	100032	100004
12-JUN-21	100032	100005
12-JUN-21	100032	200001
12-JUN-21	100032	200005
12-JUN-21	100032	200006
12-JUN-21	100032	200007
12-JUN-21	100032	300001
12-JUN-21	100032	300004
12-JUN-21	100032	300005
12-JUN-21	100033	100002
12-JUN-21	100033	100003
12-JUN-21	100033	100006
12-JUN-21	100033	100007
12-JUL-21	100031	200002
12-JUL-21	100031	200003
12-JUL-21	100031	200004
12-JUL-21	100031	200008
12-JUL-21	100032	200001
12-JUL-21	100032	200005
12-JUL-21	100032	200006
12-JUL-21	100032	200007
12-JUL-21	100034	100001
12-JUL-21	100034	100002
12-JUL-21	100034	100003
12-JUL-21	100034	100004
12-JUL-21	100034	300001
12-JUL-21	100034	300003
12-JUL-21	100035	100005
12-JUL-21	100035	100006
12-JUL-21	100035	100007
12-JUL-21	100035	300002
12-JUL-21	100035	300004
12-JUL-21	100035	300005
12-AUG-21	100032	100001
12-AUG-21	100032	100004
12-AUG-21	100032	100005
12-AUG-21	100033	100002
12-AUG-21	100033	100003
12-SEP-21	100031	100004
12-SEP-21	100031	100005
12-SEP-21	100032	100003
12-SEP-21	100033	100001
12-SEP-21	100033	100002

Download CSV
Rows 1 - 50. More rows exist.

<Pool_Maintenance_schedule> Table

PMDATE	EMPLOYEEID	FACILITYNO
12-JUN-21	100036	1111
12-JUN-21	100036	1112
12-JUL-21	100036	1111
12-JUL-21	100036	1112

Download CSV

4 rows selected.

<Schedule> Table

SCHEDULEID	SDATE	STIME	TRAINERID	FACILITYNO	CLASS_TYPE
1	01-JAN-22	8:00 AM	100019	112	CARDIO
2	01-JAN-22	10:30 AM	100019	113	BODYPUMP
3	01-JAN-22	5:15 PM	100020	114	ZUMBA
4	01-JAN-22	6:30 PM	100021	112	BODYCOMBAT
5	02-JAN-22	8:00 AM	100020	114	FULL BODY
6	02-JAN-22	10:30 AM	100022	1111	AQUA ZUMBA
7	02-JAN-22	4:00 PM	100023	112	UPPER BODY
8	02-JAN-22	5:15 PM	100021	114	SPRINT
9	02-JAN-22	6:30 PM	100019	113	FULL BODY
10	02-JAN-22	8:30 PM	100021	114	CARDIO
11	03-JAN-22	8:00 AM	100020	113	BOOTCAMP
12	03-JAN-22	10:30 AM	100020	112	SPINNING
13	03-JAN-22	5:15 PM	100021	114	CIRCUIT
14	03-JAN-22	6:30 PM	100023	112	BODYATTACK
15	03-JAN-22	6:30 PM	100019	113	BODYCOMBAT
16	03-JAN-22	8:30 PM	100019	114	ABS
17	04-JAN-22	8:00 AM	100021	112	CIRCUIT
18	04-JAN-22	10:30 AM	100024	113	STRONG ABS
19	04-JAN-22	5:15 PM	100023	112	BODYPUMP
20	04-JAN-22	6:30 PM	100020	114	CONDITIONING
21	04-JAN-22	8:30 PM	100023	112	BODYBALANCE
22	05-JAN-22	8:00 AM	100022	1112	AQUA AEROBICS
23	05-JAN-22	4:00 PM	100020	113	BOOTCAMP
24	05-JAN-22	5:15 PM	100023	114	SPRINT
25	05-JAN-22	6:30 PM	100020	113	PILATES

Download CSV

25 rows selected.

<Register_in_Schedule> Table

MEMBERID	SCHEDULEID
110001	1
110001	3
110001	7
110001	9
110001	20
110001	25
110002	4
110002	6
110002	7
110002	8
110003	8
110003	12
110003	16
110003	18
110004	2
110004	4
110004	11
110004	13
110004	22
110004	25
110005	6
110005	22
110006	1
110006	2
110006	12
110006	14
110007	7
110007	18
110007	21
110008	1
110008	3
110008	15
110009	2
110009	5
110009	23
110009	24
110010	16
110010	20
110011	9
110011	10
110011	18
110011	22
110012	3
110012	15
110012	25
110013	13
110013	14
110014	1
110014	5
110014	6

Download CSV
Rows 1 – 50. More rows exist.