FACULTY OF COMPUTING
& INFORMATION TECHNOLOGY
KING ABDULAZIZ UNIVERSITY

FCIT
KAU

كلية الحاسبات
وتقنية المعلومات
جامعة الملك عبدالعزيز

CPCS241-Database I

B-FIT Gym Company Database

Streamlining Operations

Group No: [1]

Group leader: Nora Mohammad AlGhamdi

Student Name	Student Number
Nora Mohammad AlGhamdi	2207423
Shahad Ahmed Alsamrani	2205802
Wessal Ahmed Alrabighi	2205591
Jood Fahad Alsulami	2206110
Yara Ali Abdullah Alalawi	2209191

Table of Contents

PART I: Analysis

1. Problem Definition and Data Requirements

1.1 Problem Definition

In this project, our aim is to design and develop a comprehensive and efficient database system for the B-FIT Gym Company. The goal is to revolutionize their operations and enhance the members' experience. As a leading organization in the fitness industry, the company manages multiple fitness facilities and wellness programs, serving a diverse range of members with specific fitness goals. To maintain a competitive edge and surpass member expectations, our objective is to create a cutting-edge database system that empowers the gym management team and significantly improves the overall efficiency and effectiveness of the facility.

The purpose of this database system is to provide a centralized platform for storing, organizing, and retrieving data related to B-FIT Gym's members, fitness records, reservations, employees, memberships, gym classes, branches, equipment, departments, and events. The database system will serve as the backbone of the company's operations, enabling efficient data management, accurate reporting, and improved decision-making.

Key Features and Functionality:

Member Management: The database system will store comprehensive member profiles, including personal information, contact details, payment methods, and membership details. It will allow easy access to member information, facilitating personalized services and tracking progress.

Fitness Records: The system will capture and track fitness records, including measurements such as weight, height, body fat, and blood type. These records will be associated with member IDs for easy retrieval and analysis.

Reservation System: The database system will facilitate seamless class reservations for members. It will maintain class schedules, availability, and member reservations. This feature will help manage class capacities, avoid overbooking, and ensure a smooth experience for members.

Employee Management: The system will store employee profiles, including personal information, roles, and contact details. It will facilitate employee scheduling, task assignments, and coordination.

Membership Management: The database system will handle membership details, including membership types, start dates, expiration dates, and associated member information. It will enable efficient management and tracking of memberships.

Gym Class Management: The system will store information about gym classes, including class schedules, capacities, descriptions, and assigned coaches. It will facilitate class registration, attendance tracking, and management.

Branch Management: The database system will store branch information, including branch numbers, work hours, and addresses. It will enable efficient management of multiple gym locations and their respective operations.

Equipment Management: The system will track the inventory of gym equipment, including equipment numbers and names. It will facilitate equipment maintenance, usage tracking, and scheduling.

Department Management: The database system will handle department details, including department numbers, names, locations, and the number of employees in each department. It will support efficient department management and organization.

Event Management: The system will support the organization and scheduling of events, such as workshops, seminars, and competitions. It will store event details, dates, times, and locations. This feature will enhance member engagement and community building.

The designed database system will bring numerous benefits, including improved data management, enhanced member experiences, streamlined operations, data-driven decision making, and scalability for future growth.

1.1 Data Requirements

1. **Member**
   - Name (Fname, Mname, Lname)
   - Unique Member ID
   - Birth Date
   - Gender
   - Contact information (phone, E-mail)
   - Payment method

2. **Records**
   - Weight
   - Height
   - Date of record
   - Blood type
   - Body fat

3. **Branches**
   - Branch number
   - Work hours
   - Branch address (city, street)

4. **Employee**
   - Name (Fname, Mname, Lname)
   - Employee ID
   - Salary
   - Contact information (phone number, E-mail)
   - Job name

5. **Membership**
   - Type
   - Membership ID
   - State
   - Start date
   - Expired date

6. **Gym Classes**
   - Class Number
   - Class schedule (class capacity, class description, duration)
   - Class Name

7. **Reservations**
   - Reservation ID
   - Reservation Date (Time, Day, Month, Year)

8. **Department**
   - Name
   - Number
   - Location
   - Number of employees

9. **Equipment**
   - Equipment name
   - Equipment Number

10. **Event**
    - Event Name
    - Time
    - Date
    - Location

1.2 Business Rules

11. Member:

The system will store and manage member information, including name (first name, middle name, last name), unique member ID, birth date, gender, contact information (phone, email), and payment method.

12. Records:

The system will track member records, including weight, height, date of record, blood type, and body fat measurements.

13. Branches:

Each branch will have a branch number, work hours, and branch address (city, street).

14. Employee:

The system will manage employee details, including name (first name, middle name, last name), employee ID, role, and contact information (phone number, email), salary.

15. Membership:

Memberships will have a type, membership ID, state, start date, and expiration date.

16. Gym Classes:

Each gym class will have a class number, class schedule (including class capacity, class description, and duration), and class name.

17. Reservations:

The system will handle reservations for events or classes.

Each reservation will have a reservation ID and reservation date (time, day, month, year).

18. **Department:**

    The system will store department information, including name, number, location, and the number of employees in the department.

19. **Equipment:**

    The system will manage gym equipment details, including equipment name and equipment number.

20. **Event:**

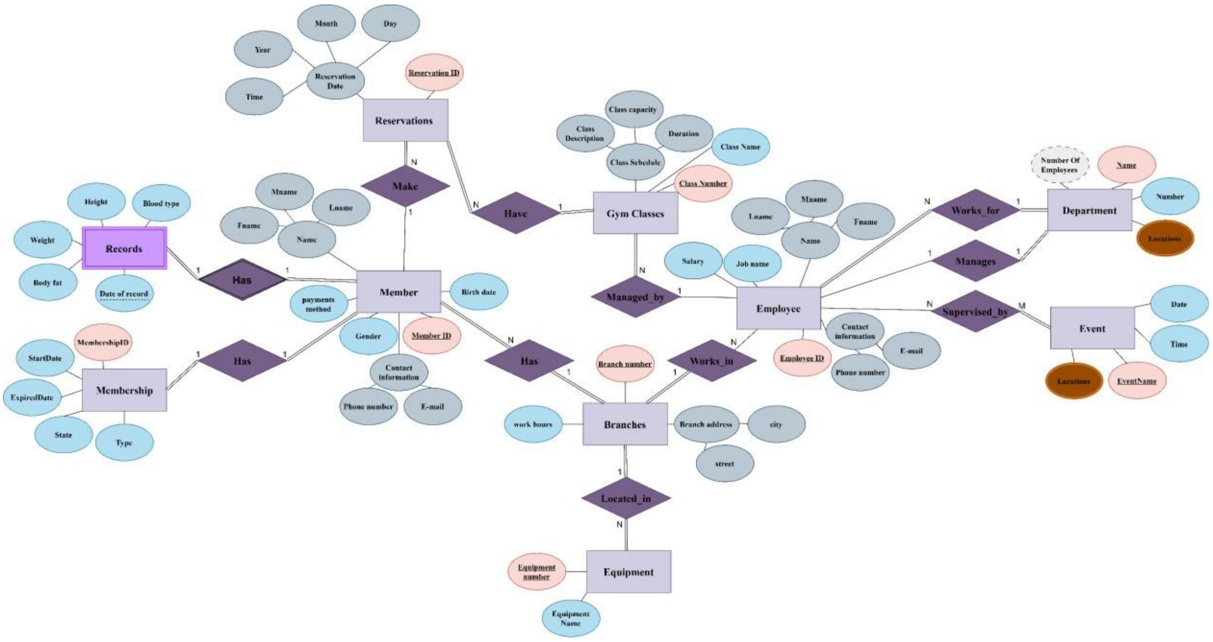
    Events will have an event name, time, date, and location.

These business rules provide an overview of the main entities and their associated attributes in the system. They serve as guidelines for how data will be stored and managed within the system.

PART II: DB DEISGN

2 ER Diagram Design

2.1 ER diagram



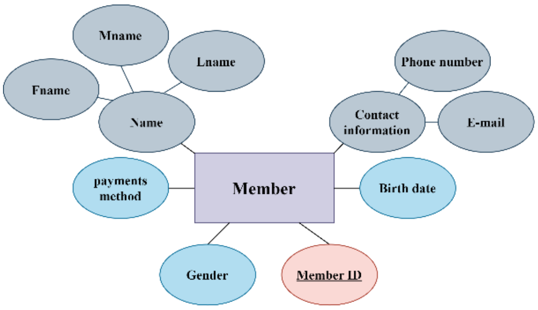
For clearer version of the ER diagram click this link (https)

2.2 Design of Business Rules

Business Rule	Design Decisions	Justification (if any)
Each member **has** records	1= binary relationship between member & records	Members & records have total participation on both sides
Each member **has** membership	1:1 binary relationship between member & membership	Members & membership have total participation on both sides
Each member can **make** multiple reservations	1:N binary relationship between member & reservations	Reservations side have total participation with member, and partial in the member side
Branches **has** multiple members	1:N binary relationship between branches & members	Branches & members have total participations on both sides.
Each employee **works in** a branch	1:N binary relationship between branch & employee	Branch side is total participation, and employee side is partial participation
Each equipment **located in** branch	1:N binary relationship between branch & equipment	Branch & equipment has total participation on both sides
Each gym class in **managed by** an employee	1:N binary relationship between employee & gym classes	The employee has partial participation on its side, and gym classes has total participation on its side
Each event is **supervised by** employees	N:M binary relationship between employee & event	Partial participation for employees, and total participation for event side
Each employee **works for** department	1:N binary relationship between department and employee	Both employee and department have total participation on both sides
Each employee **manages** department	1:1 binary relationship between department and employee	The employee side has partial participation, and department side has total participation
Each gym class **have** reservations	1:N Binary relationship between the gym class & reservations	Both sides have total participations, reservations & gym classes

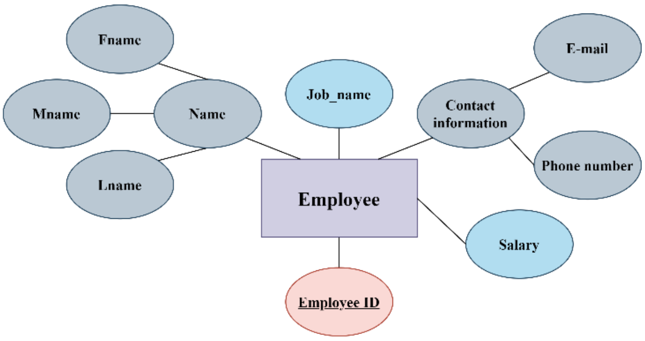
3 ER-to-logical schema mapping

3.1 Mapping of Regular Entity Types



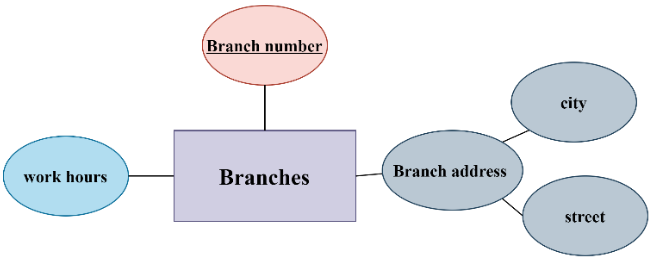
Member

Member ID	Fname	Mname	Lname	Gender	Phone Number	E-mail	payment method	Birth date



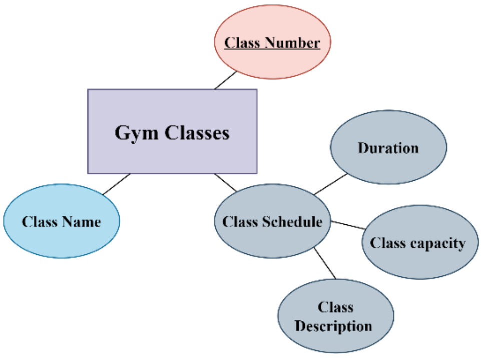
Employee

Employee ID	Fname	Mname	Lname	Job_name	E-mail	Phone Number	Salary



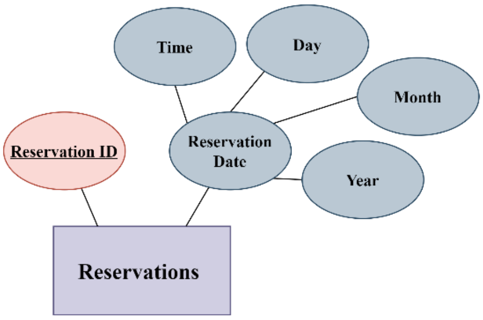
Branches

Branch number	work hours	city	Street



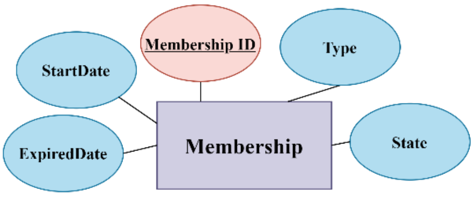
Gym Classes

<u>Class Number</u>	Class Name	Duration	Class Capacity	Class Description



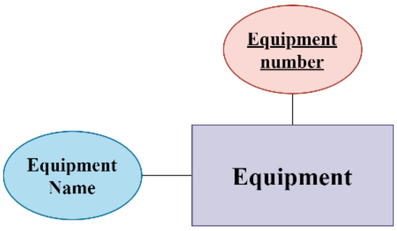
Reservations

<u>Reservation ID</u>	Time	Day	Month	Year



Membership

MembershipID	StartDate	ExpiredDate	Type	State

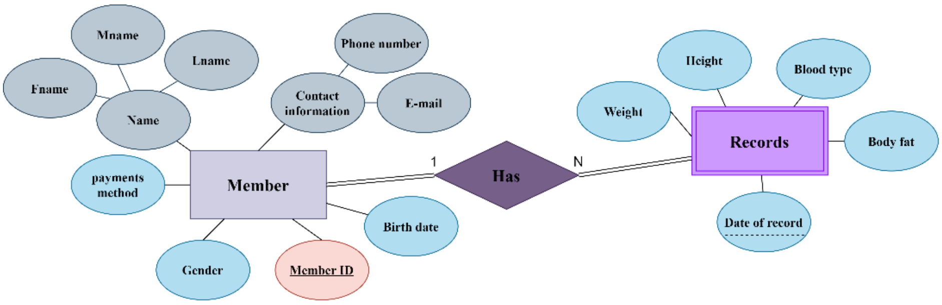


Equipment

Equipment Number	Equipment Name

For clearer version of the Regular Entity Mapping in this link: (https)

3.2 Mapping of Weak Entity Types



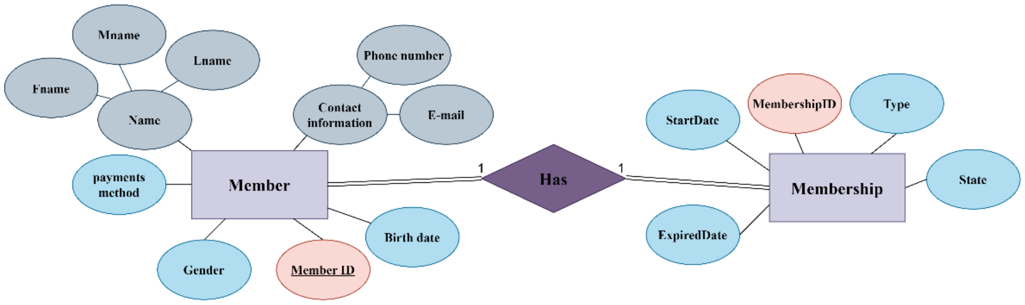
Member

Member ID	Fname	Mname	Lname	Gender	Phone Number	E-mail	payment method	Birth date

Records

Date of record	Mbr_ID	Weight	Hight	Blood type	Body fat

3.3 Mapping of binary 1-1 relationship types



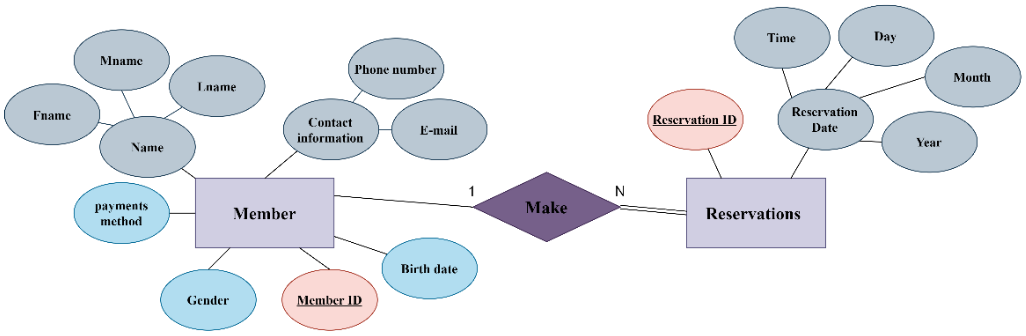
Member

<u>Member ID</u>	Fname	Mname	Lname	Gender	Phone Number	E-mail	payment method	Birth date

Membership

<u>MembershipID</u>	Mbr_ID	StartDate	ExpiredDate	Type	State

3.4 Mapping of binary 1-N relationship types

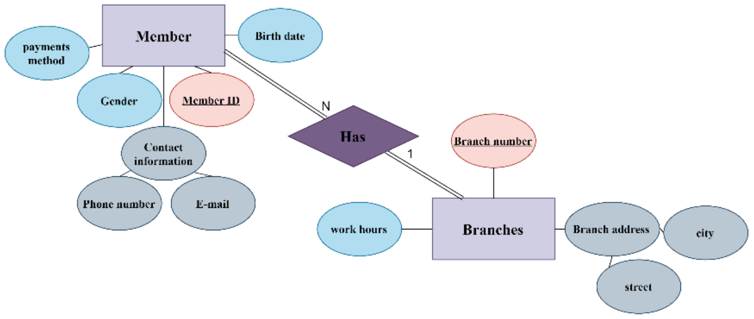


Member

Member ID	Fname	Mname	Lname	Gender	Phone Number	E-mail	payment method	Birth date

Reservations

Reservation ID	Mbr_ID	Time	Day	Month	Year

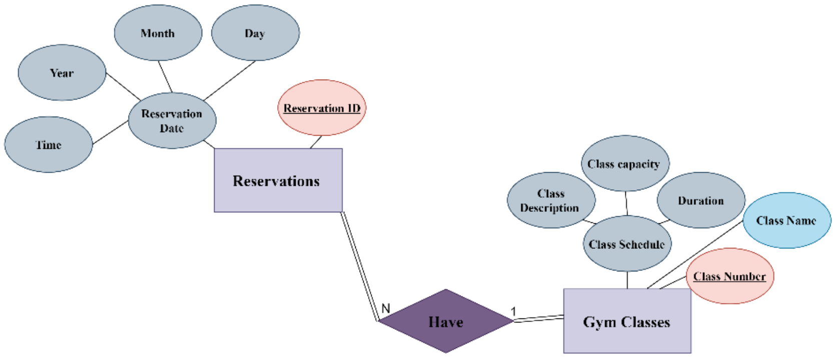


Branches

Branch number	work hours	city	Street

Member

Member ID	Brn_No	Fname	Mname	Lname	Gender	Phone Number	E-mail	payment method	Birth date

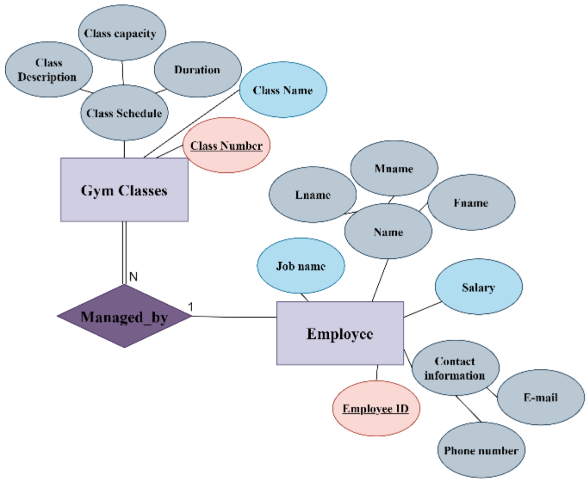


Gym Classes

Class Number	Mbr_ID	Class_Name	Duration	Class_Capacity	Class_Description

Reservations

Reservation_ID	Class_No	Mbr_ID	Time	Day	Month	Year

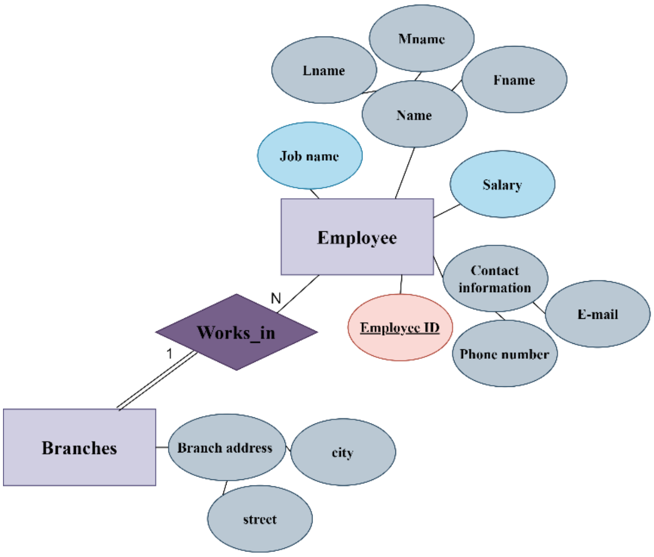


Employee

Employee ID	Dep_name	Fname	Mname	Lname	Job name	E-mail	Phone Number	Salary

Gym Classes

Class Number	Emp_ID	Mbr_ID	Class_Name	Duration	Class_Capacity	Class_Description

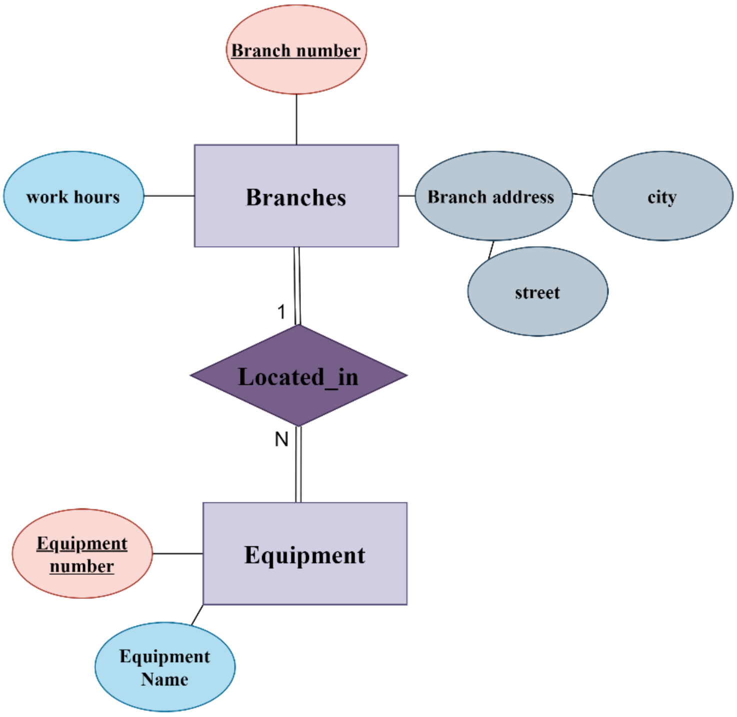


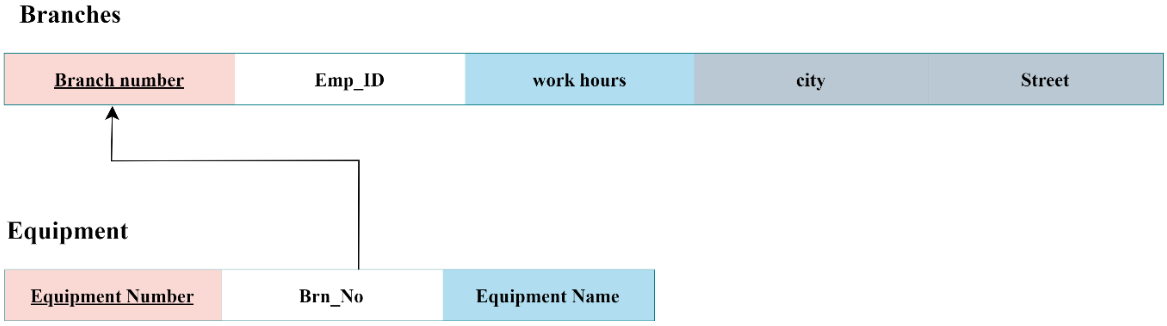
Employee

<u>Employee ID</u>	Fname	Mname	Lname	Job_name	E-mail	Phone Number	Salary

Branches

<u>Branch number</u>	Emp_ID	work hours	city	Street





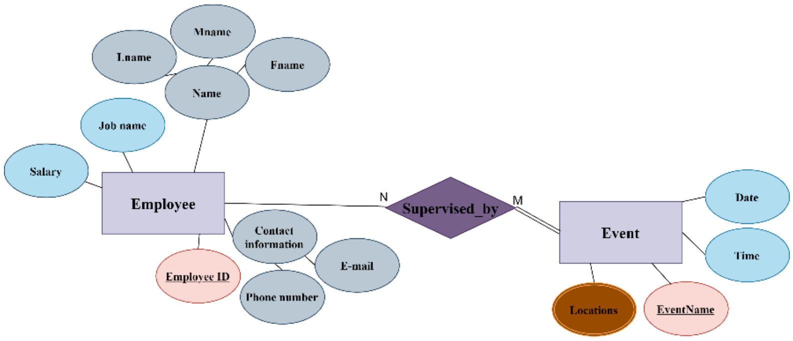
Branches

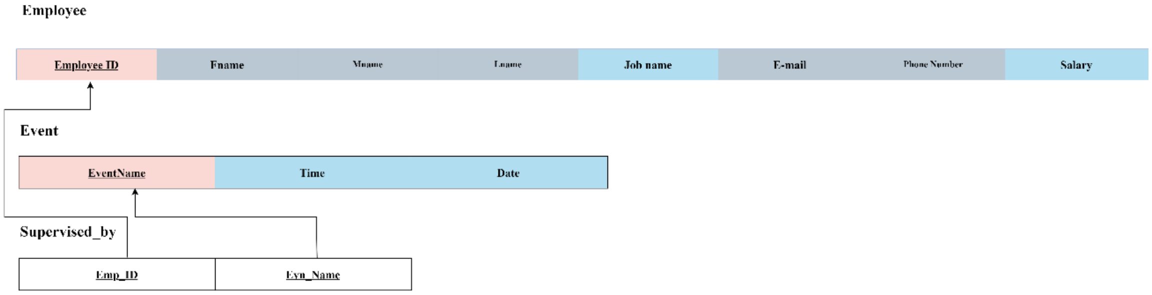
Branch number	Emp_ID	work hours	city	Street

Equipment

Equipment Number	Brn_No	Equipment Name

3.5 Mapping of binary M-N relationship types





Employee

Employee ID	Fname	Mname	Lname	Job name	E-mail	Phone Number	Salary

Event

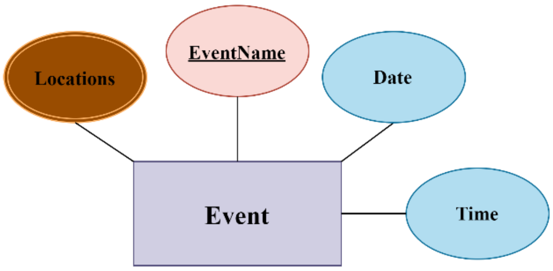
EventName	Time	Date

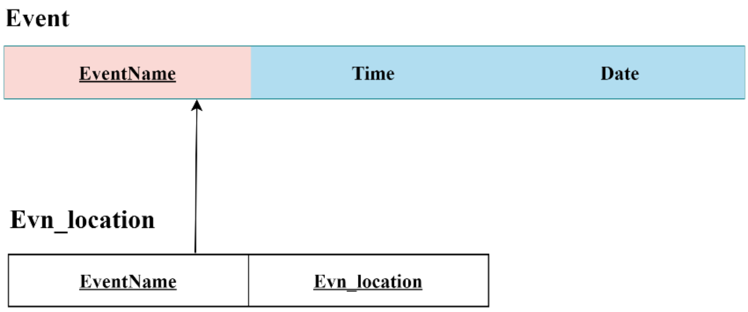
Supervised_by

Emp_ID	Evn_Name

For clearer version of the Mapping of 1:1 & 1:N & N:M in this link: (https)

3.6 Mapping of multivalued attributes

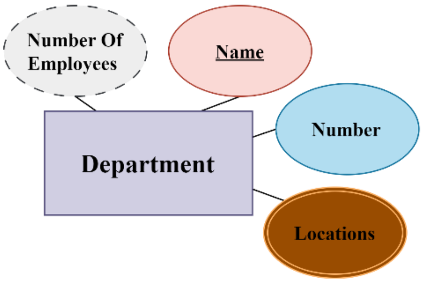


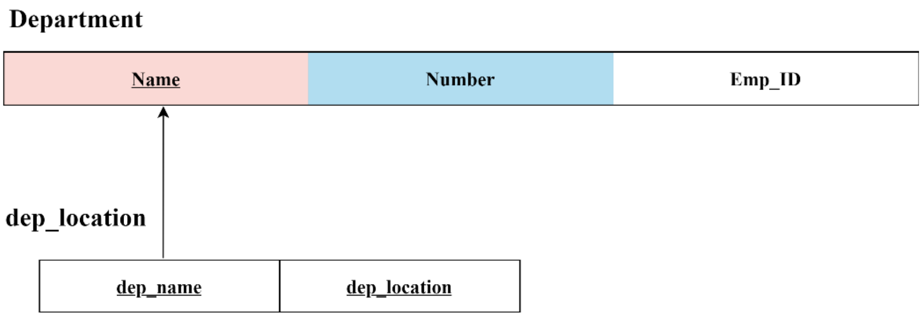

Event

EventName	Time	Date

Evn_location

EventName	Evn_location




Department

Name	Number	Emp_ID

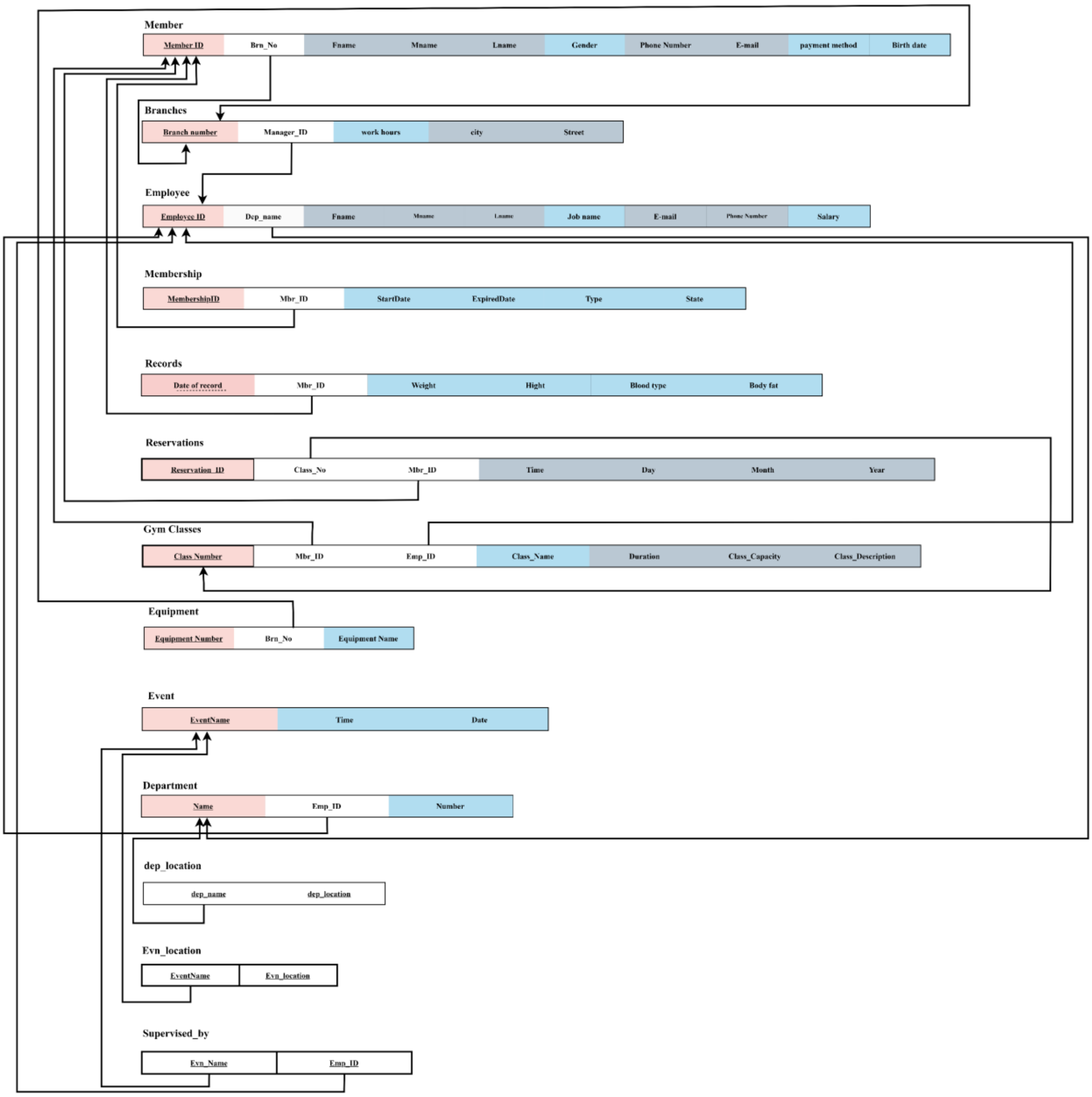
dep_location

dep_name	dep_location

3.1 Mapping of n-ary relationship types

There is no mapping of n-ary relationship types.

3.7 Schema Diagram

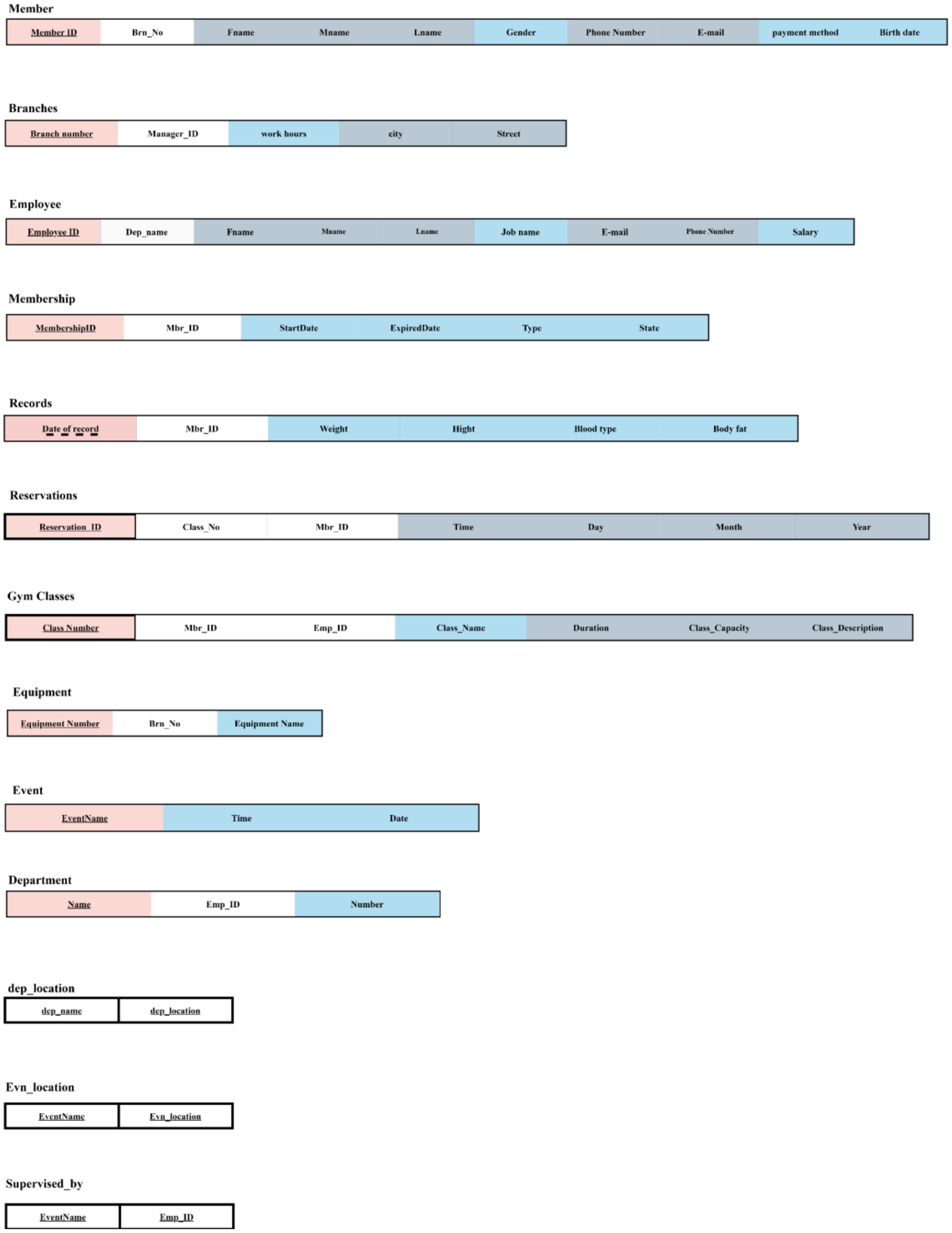


For clearer version of the Schema in this link: (https)

4 Normalization

4.1 First Normal Form

The First Normal Form (1NF) ensures that relations (tables) in a relational schema do not have multivalued attributes, grouping attributes, or nested relations. In our case, all the relations in our schema are already in 1NF because all attributes follow the rules of 1NF, being simple and dependent on the key.



Member

Member ID	Brn_No	Fname	Mname	Lname	Gender	Phone Number	E-mail	payment method	Birth date

Branches

Branch number	Manager_ID	work hours	city	Street

Employee

Employee ID	Dep_name	Fname	Mname	Lname	Job name	E-mail	Phone Number	Salary

Membership

MembershipID	Mbr_ID	StartDate	ExpiredDate	Type	State

Records

Date of record	Mbr_ID	Weight	Hight	Blood type	Body fat

Reservations

Reservation ID	Class_No	Mbr_ID	Time	Day	Month	Year

Gym Classes

Class Number	Mbr_ID	Emp_ID	Class_Name	Duration	Class_Capacity	Class_Description

Equipment

Equipment Number	Brn_No	Equipment Name

Event

EventName	Time	Date

Department

Name	Emp_ID	Number

dep_location

dep_name	dep_location

Evn_location

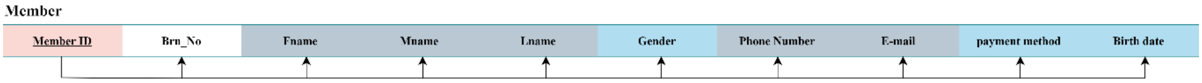
EventName	Evn_location

Supervised_by

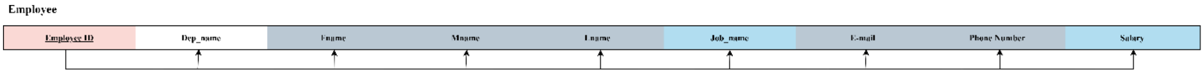
EventName	Emp_ID

4.2 Second Normal Form

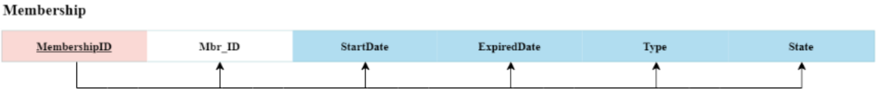
The Second Normal Form (2NF) is achieved when every non-prime attribute in a relation is fully functionally dependent on the primary key. In our case, all tables in our database are in 2NF because they meet this condition of full functional dependency between non-prime attributes and the primary key.



The **Member** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

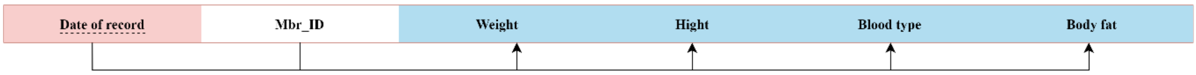


The **Employee** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.



The **Membership** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

Records



Date of record	Mbr_ID	Weight	Hight	Blood type	Body fat

The **Records** relation satisfies the conditions for the second normal form (2NF). It has two prime attributes, Mbr_ID and Date_of_record, and all the non-prime attributes in the relation are fully functionally dependent on both prime attributes. As a result, the Records relation is in 2NF.

Department

Name	Emp_ID	Number

The **Department** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

dep_location

dep_name	dep_location

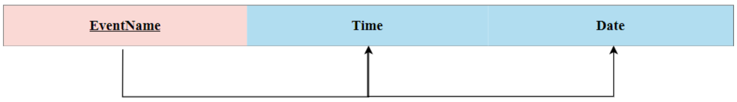
The **dep_location** relation is in the second normal form (2NF). It has two prime attributes, dep_location and dep_name, and does not contain any non-prime attributes. Therefore, since there are no non-prime attributes present, there is no need to prove functional dependency or perform further normalization. Thus, the **dep_location** relation already satisfies the requirements of 2NF.

Supervised_by

Evn_Name	Emp_ID

The Supervised_by relation is in the second normal form (2NF). It consists of two prime attributes, Event_Name and Emp_ID, and does not have any non-prime attributes. Therefore, since there are no non-prime attributes present, there is no need to prove functional dependency or further normalize the relation. As a result, the Supervised_by relation already satisfies the requirements of 2NF.

Event



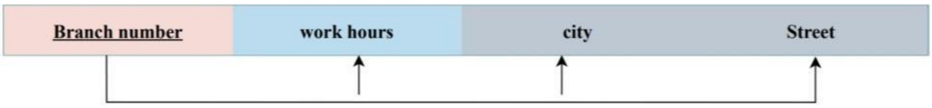
The **Event** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

Evn_location

EventName	Evn_location

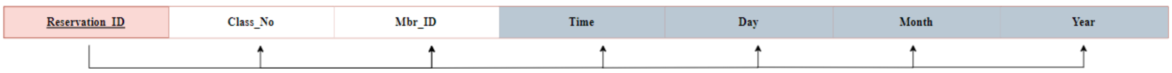
The **Evn_location** relation is in the second normal form (2NF). It has two prime attributes, EventName and Evn_location, and does not contain any non-prime attributes. Therefore, since there are no non-prime attributes present, there is no need to prove functional dependency or perform further normalization. Thus, the **Evn_location** relation already satisfies the requirements of 2NF.

Branches



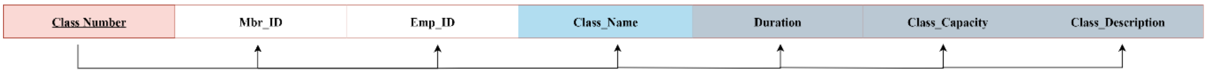
The **Branches** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

Reservations



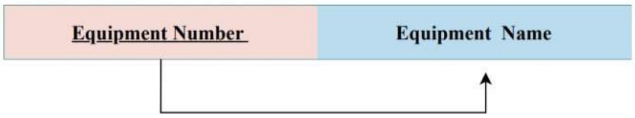
The **Reservations** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

Gym Classes



The **Gym classes** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

Equipment



The **Equipment** relation in the second normal form (2NF) indicates that all non-primary attributes are fully dependent on the entire key, ensuring there are no partial dependencies. In this case, the relation contains a single-attribute primary key, and all non-primary attributes are fully functionally dependent on it. Therefore, the relation satisfies the requirements of 2NF.

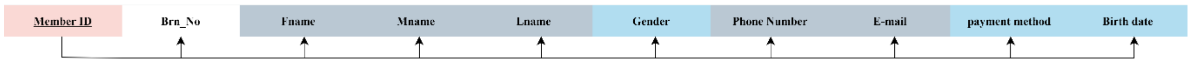
4.3 Third Normal Form

In section 4.2, all the relational schemas have been normalized into the second normal form (2NF). Now, we will identify and list the relational schemas that are already in the third normal form (3NF) as they do not contain any transitive dependencies on the primary key.

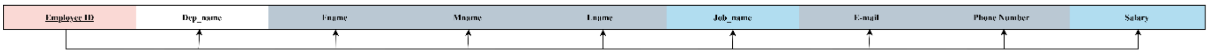
To determine if a relation is in 3NF, we need to check if there are any non-prime attributes that are functionally dependent on other non-prime attributes. If such dependencies exist, we need to ensure that the non-prime attribute being depended upon is a candidate key (unique) for the relation to satisfy 3NF.

All the following relations are in the 3NF:

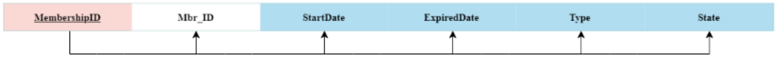
Member



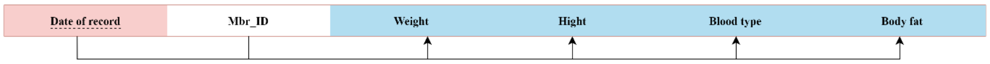
Employee



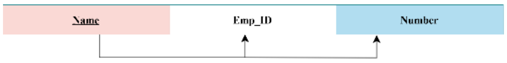
Membership



Records



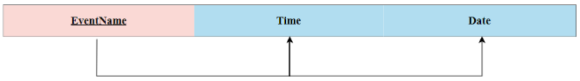
Department



dep_location

dep_name	dep_location

Event



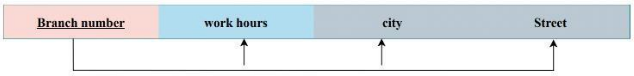
Evn_location

EventName	Evn_location

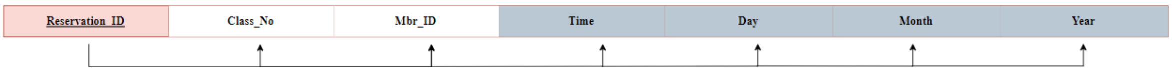
Supervised_by

Evn_Name	Emp_ID

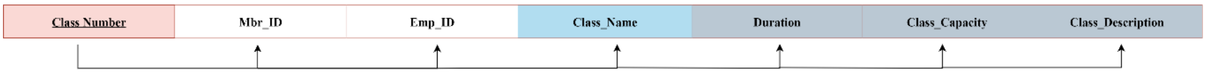
Branches



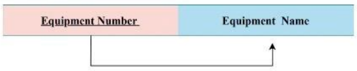
Reservations



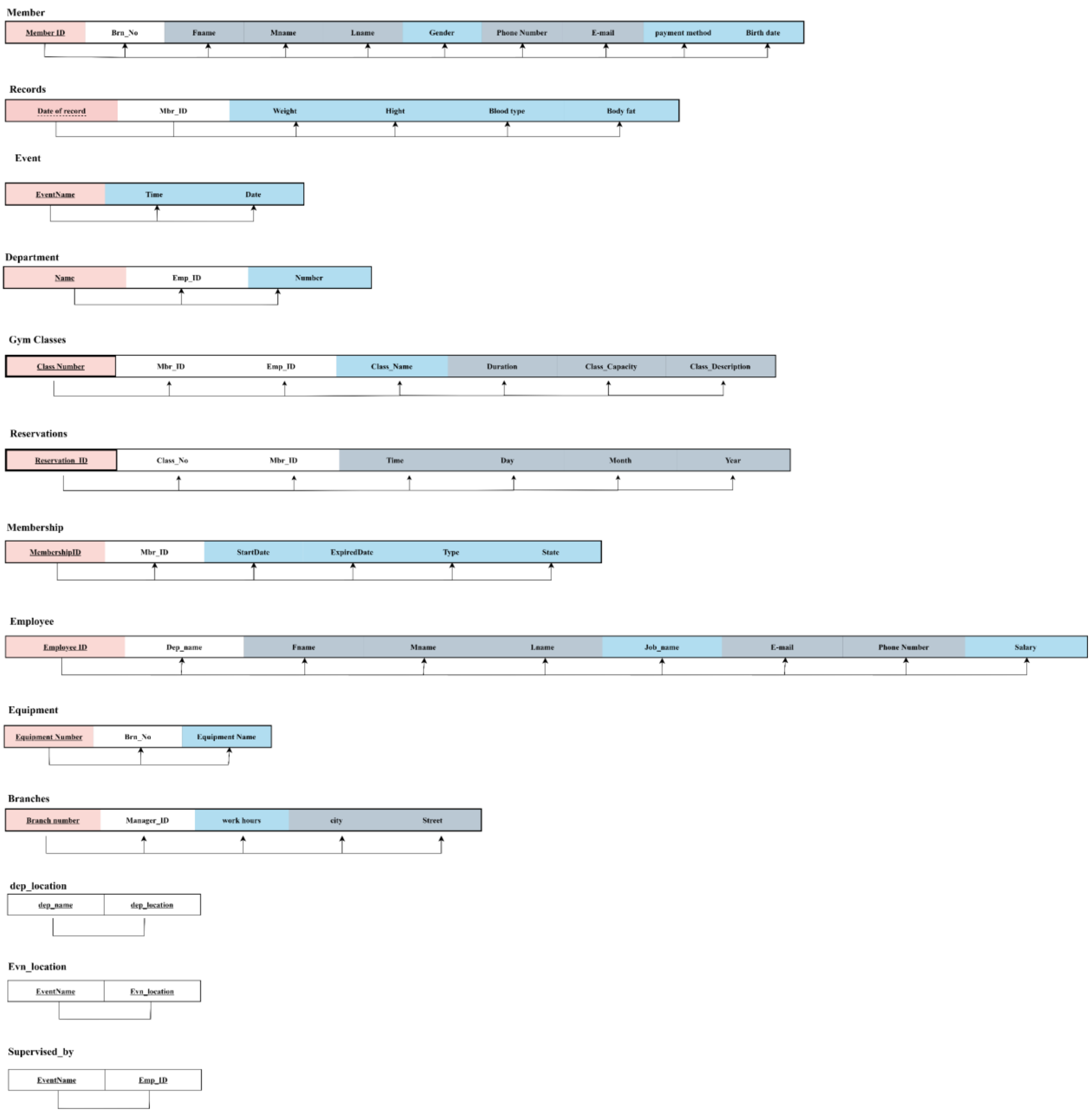
Gym Classes



Equipment



There is no transitive dependency in our relations, so all relation schemas are in 3NF.



Member

Member ID	Brn_No	Fname	Mname	Lname	Gender	Phone Number	E-mail	payment method	Birth date

Records

Date of record	Mbr_ID	Weight	Hight	Blood type	Body fat

Event

EventName	Time	Date

Department

Name	Emp_ID	Number

Gym Classes

Class Number	Mbr_ID	Emp_ID	Class_Name	Duration	Class_Capacity	Class_Description

Reservations

Reservation_ID	Class_No	Mbr_ID	Time	Day	Month	Year

Membership

MembershipID	Mbr_ID	StartDate	ExpiredDate	Type	State

Employee

Employee ID	Dep_name	Fname	Mname	Lname	Job_name	E-mail	Phone Number	Salary

Equipment

Equipment Number	Brn_No	Equipment Name

Branches

Branch number	Manager_ID	work hours	city	Street

dep_location

dep_name	dep_location

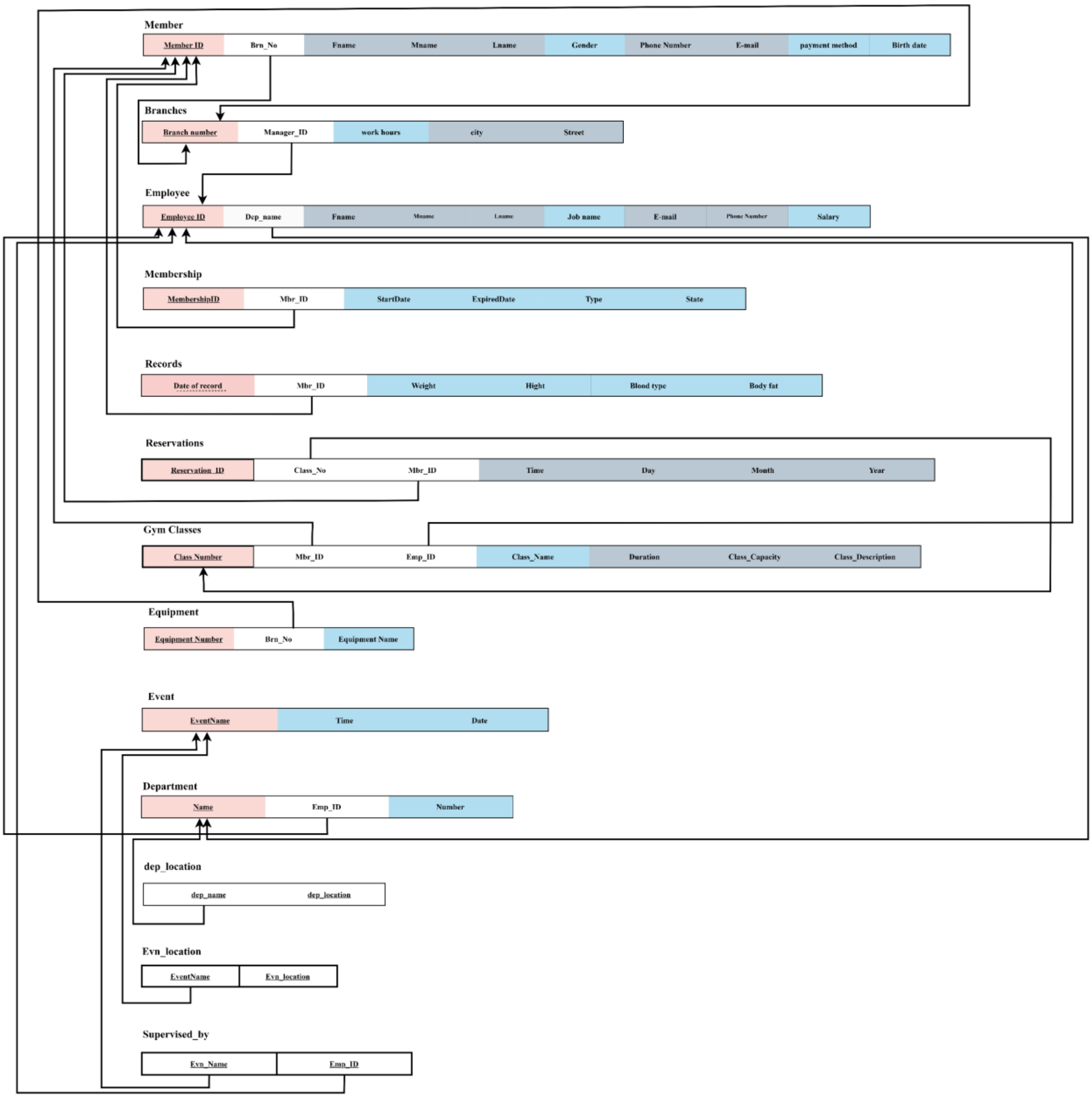
Evn_location

EventName	Evn_location

Supervised_by

EventName	Emp_ID

5 Final DB Schema Diagram



6 Queries

Query 1:

< Retrieve the top 3 blood types among members with gym class reservations >

Query 2:

< Find the highest salary in a specified department >

Query 3:

< Retrieve the name and ID of all employees who work for the Marketing department and pick the lowest salary in the marketing department and add a raise to it by 10% >

Query 4:

< Calculate the branch with the largest number of members >

Query 5:

< Given a coach name, display the reservation ID, class number of the Coach >

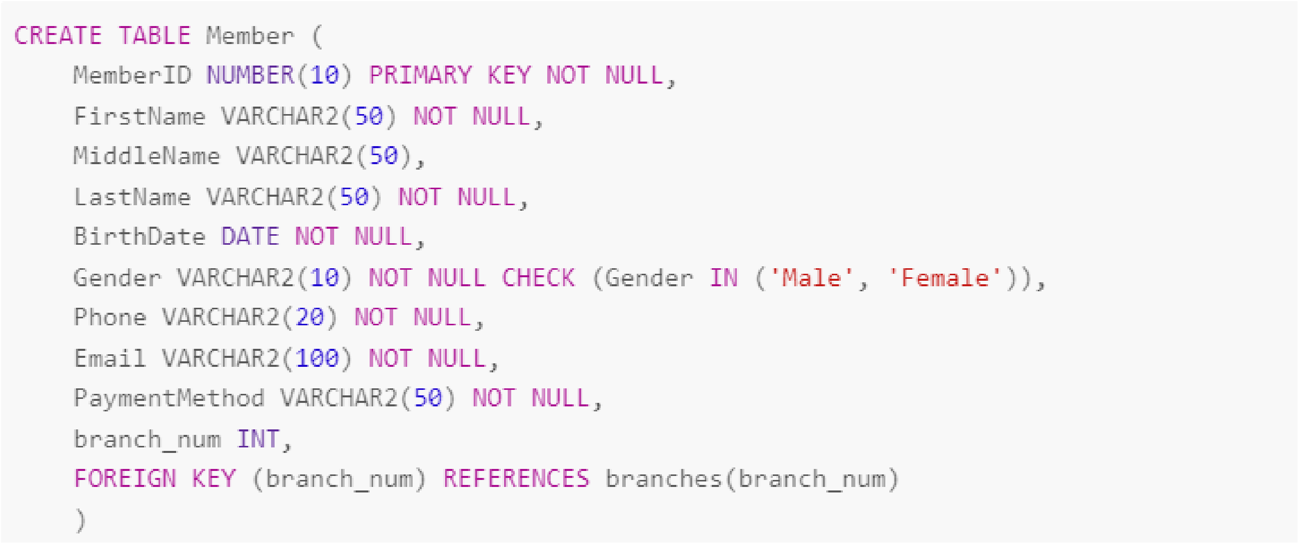
PART III: IMPLEMENTATION

6 Table Creation Script

In this section, we present the table creation scripts for the B-Fit system, ensuring that all tables are designed in accordance with the principles of the Third Normal Form (3NF). Each table is organized in a separate subsection, listing the table name and its corresponding attributes. Constraints, including referential integrity and attribute constraints, are highlighted for clarity and adherence to data integrity principles.

6.1 <*Member*> TABLE

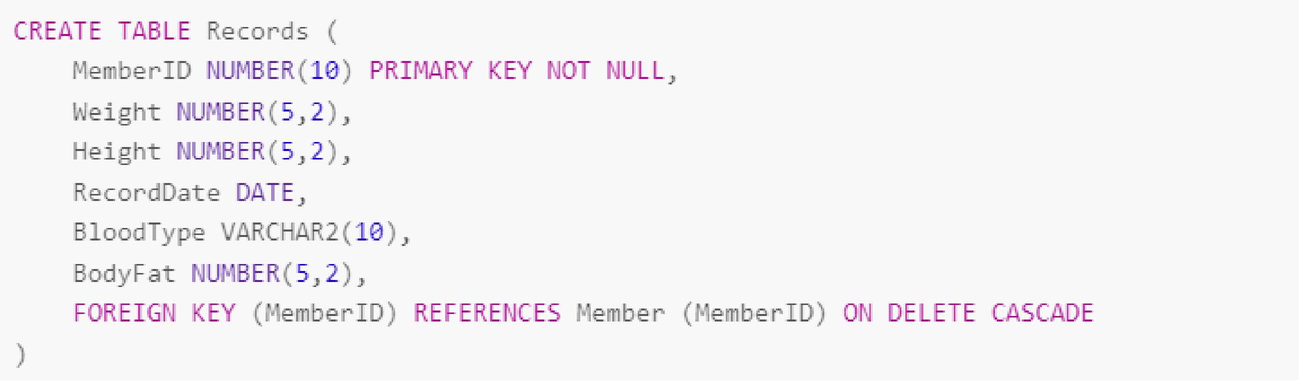
```
CREATE TABLE Member (
    MemberID NUMBER(10) PRIMARY KEY NOT NULL,
    FirstName VARCHAR2(50) NOT NULL,
    MiddleName VARCHAR2(50),
    LastName VARCHAR2(50) NOT NULL,
    BirthDate DATE NOT NULL,
    Gender VARCHAR2(10) NOT NULL CHECK (Gender IN ('Male', 'Female')),
    Phone VARCHAR2(20) NOT NULL,
    Email VARCHAR2(100) NOT NULL,
    PaymentMethod VARCHAR2(50) NOT NULL,
    branch_num INT,
    FOREIGN KEY (branch_num) REFERENCES branches(branch_num)
    )
```



```
CREATE TABLE Member (
    MemberID NUMBER(10) PRIMARY KEY NOT NULL,
    FirstName VARCHAR2(50) NOT NULL,
    MiddleName VARCHAR2(50),
    LastName VARCHAR2(50) NOT NULL,
    BirthDate DATE NOT NULL,
    Gender VARCHAR2(10) NOT NULL CHECK (Gender IN ('Male', 'Female')),
    Phone VARCHAR2(20) NOT NULL,
    Email VARCHAR2(100) NOT NULL,
    PaymentMethod VARCHAR2(50) NOT NULL,
    branch_num INT,
    FOREIGN KEY (branch_num) REFERENCES branches(branch_num)
    )
```

6.2 <*Records*> TABLE

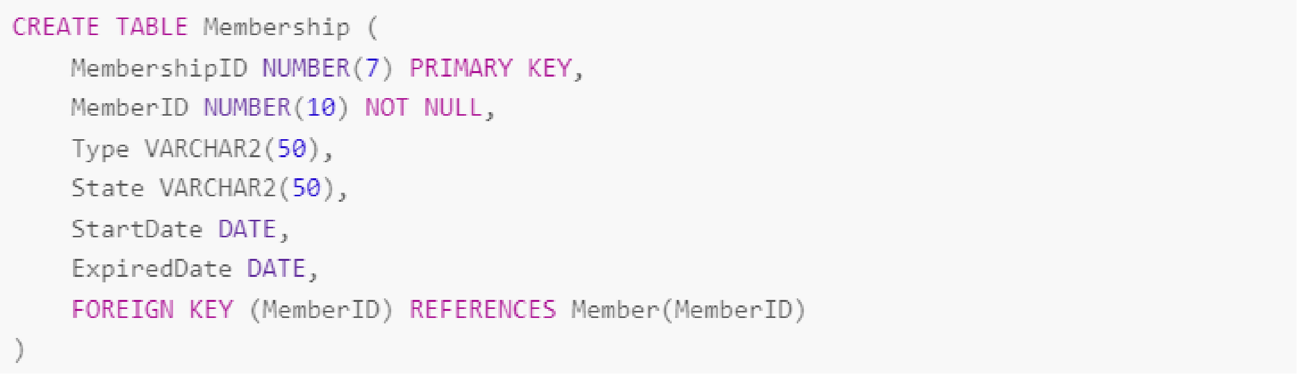
```
CREATE TABLE Records (
    MemberID NUMBER(10) PRIMARY KEY NOT NULL,
    Weight NUMBER(5,2),
    Height NUMBER(5,2),
    RecordDate DATE,
    BloodType VARCHAR2(10),
    BodyFat NUMBER(5,2),
    FOREIGN KEY (MemberID) REFERENCES Member (MemberID) ON DELETE CASCADE
)
```



```
CREATE TABLE Records (
    MemberID NUMBER(10) PRIMARY KEY NOT NULL,
    Weight NUMBER(5,2),
    Height NUMBER(5,2),
    RecordDate DATE,
    BloodType VARCHAR2(10),
    BodyFat NUMBER(5,2),
    FOREIGN KEY (MemberID) REFERENCES Member (MemberID) ON DELETE CASCADE
)
```

6.3 <*Membership*> TABLE

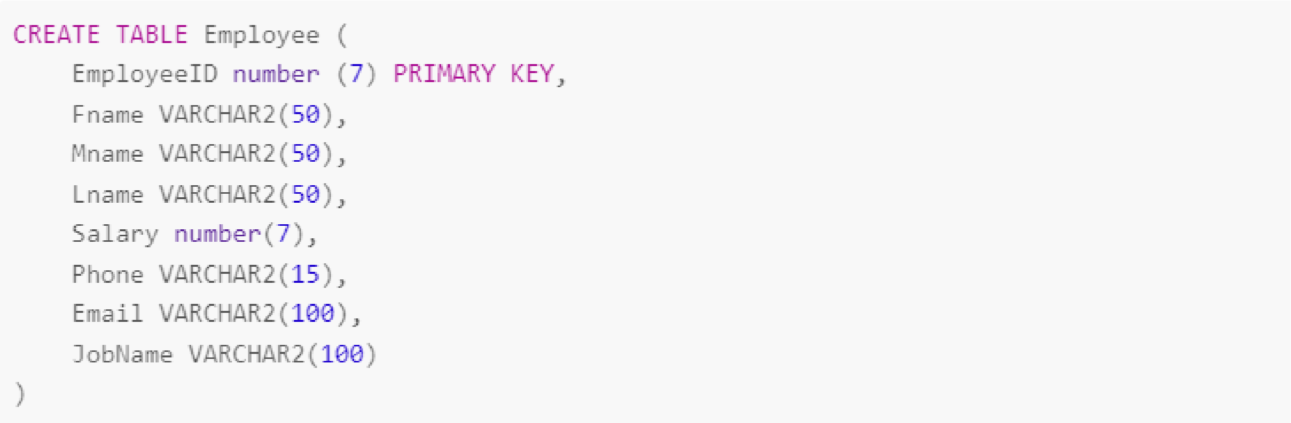
```
CREATE TABLE Membership (
    MembershipID NUMBER(7) PRIMARY KEY,
    MemberID NUMBER(10) NOT NULL,
    Type VARCHAR2(50),
    State VARCHAR2(50),
    StartDate DATE,
    ExpiredDate DATE,
    FOREIGN KEY (MemberID) REFERENCES Member(MemberID)
)
```



```
CREATE TABLE Membership (
    MembershipID NUMBER(7) PRIMARY KEY,
    MemberID NUMBER(10) NOT NULL,
    Type VARCHAR2(50),
    State VARCHAR2(50),
    StartDate DATE,
    ExpiredDate DATE,
    FOREIGN KEY (MemberID) REFERENCES Member(MemberID)
)
```

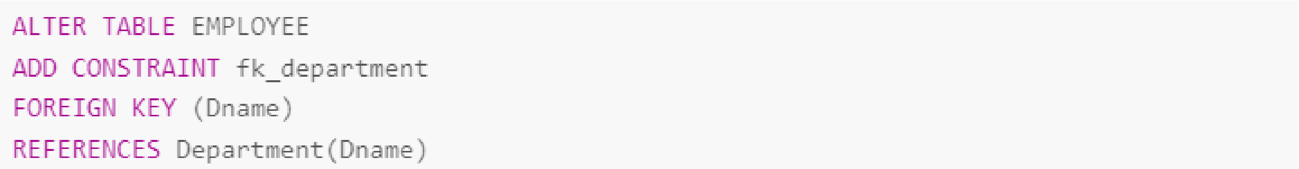
6.4 <*Employee*> TABLE

```
CREATE TABLE Employee (
    EmployeeID number (7) PRIMARY KEY,
    Fname VARCHAR2(50),
    Mname VARCHAR2(50),
    Lname VARCHAR2(50),
    Salary number(7),
    Phone VARCHAR2(15),
    Email VARCHAR2(100),
    JobName VARCHAR2(100)
)

ALTER TABLE EMPLOYEE
ADD Dname VARCHAR2(30)

ALTER TABLE EMPLOYEE
ADD CONSTRAINT fk_department
FOREIGN KEY (Dname)
REFERENCES Department(Dname)
```



```
CREATE TABLE Employee (
    EmployeeID number (7) PRIMARY KEY,
    Fname VARCHAR2(50),
    Mname VARCHAR2(50),
    Lname VARCHAR2(50),
    Salary number(7),
    Phone VARCHAR2(15),
    Email VARCHAR2(100),
    JobName VARCHAR2(100)
)
```

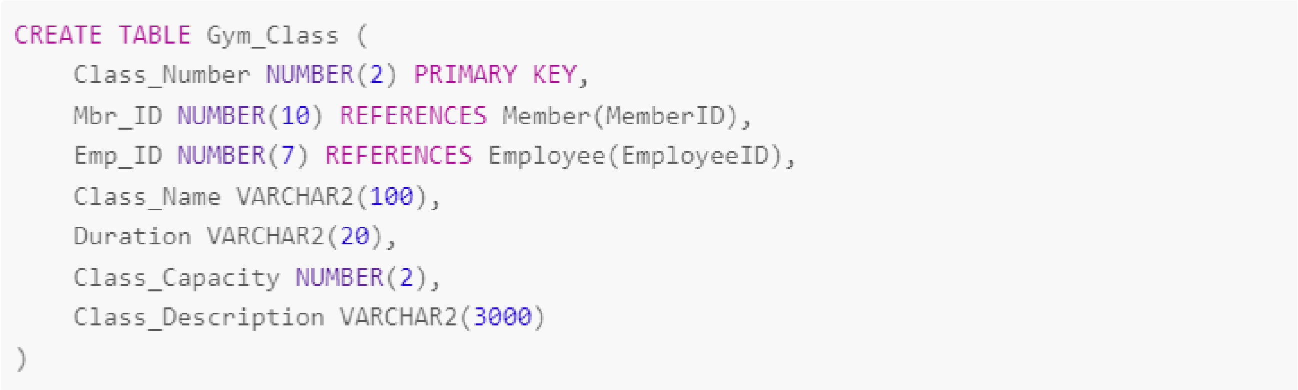
```
ALTER TABLE EMPLOYEE
ADD Dname VARCHAR2(30)
```



```
ALTER TABLE EMPLOYEE
ADD CONSTRAINT fk_department
FOREIGN KEY (Dname)
REFERENCES Department(Dname)
```

6.5 <*Gym Classes*> TABLE

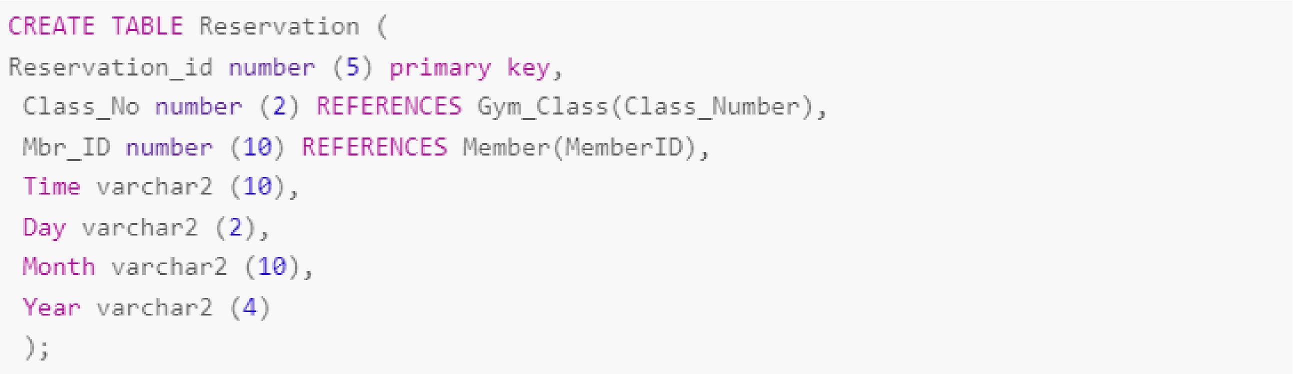
```
CREATE TABLE Gym_Class (
    Class_Number number (2) primary key,
    Mbr_ID number (10) REFERENCES Member(MemberID) ,
    Emp_ID number (7) REFERENCES Employee(EmployeeID),
    Class_Name varchar2 (100),
    Duration varchar2 (20),
    Class_Capacity number (2),
    Class_Descreption varchar2 (3000)
    )
```



```
CREATE TABLE Gym_Class (
    Class_Number NUMBER(2) PRIMARY KEY,
    Mbr_ID NUMBER(10) REFERENCES Member(MemberID),
    Emp_ID NUMBER(7) REFERENCES Employee(EmployeeID),
    Class_Name VARCHAR2(100),
    Duration VARCHAR2(20),
    Class_Capacity NUMBER(2),
    Class_Description VARCHAR2(3000)
)
```

6.6 <*Reservations*> TABLE

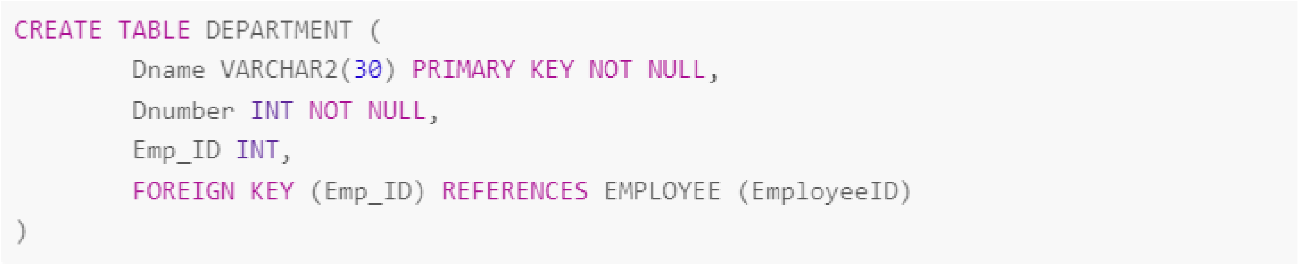
```
CREATE TABLE Reservation (
    Reservation_id number (5) primary key,
    Class_No number (2) REFERENCES Gym_Class(Class_Number),
    Mbr_ID number (10) REFERENCES Member(MemberID),
    Time varchar2 (10),
    Day varchar2 (2),
    Month varchar2 (10),
    Year varchar2 (4)
    )
```



```
CREATE TABLE Reservation (
Reservation_id number (5) primary key,
 Class_No number (2) REFERENCES Gym_Class(Class_Number),
 Mbr_ID number (10) REFERENCES Member(MemberID),
 Time varchar2 (10),
 Day varchar2 (2),
 Month varchar2 (10),
 Year varchar2 (4)
 );
```

6.7 <*Department*> TABLE

```
CREATE TABLE DEPARTMENT (
       Dname VARCHAR2(30) PRIMARY KEY NOT NULL,
       Dnumber INT NOT NULL,
       Emp_ID INT,
       FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE (EmployeeID)
)
```



```
CREATE TABLE DEPARTMENT (
        Dname VARCHAR2(30) PRIMARY KEY NOT NULL,
        Dnumber INT NOT NULL,
        Emp_ID INT,
        FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE (EmployeeID)
)
```

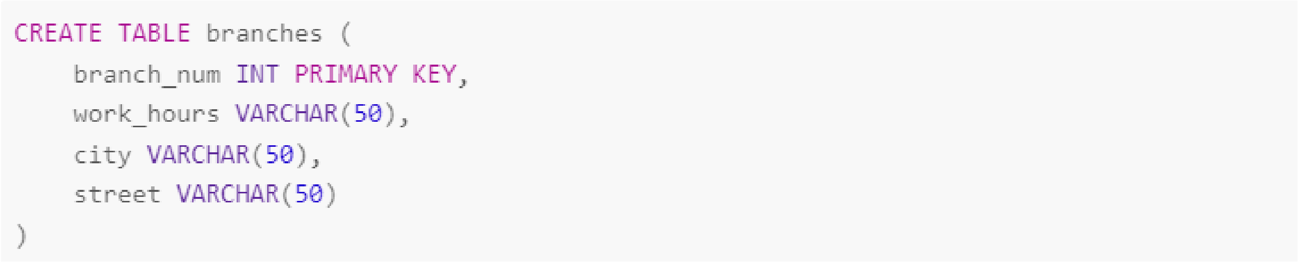
6.8 <Event> TABLE

```
CREATE TABLE EVENT (
    EventName VARCHAR2(30) PRIMARY KEY NOT NULL,
    EventTime VARCHAR2(10),
    EventDate DATE
)
```

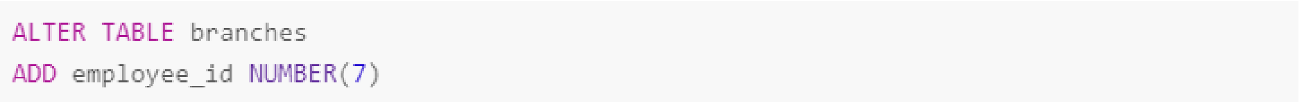
```
CREATE TABLE EVENT (
    EventName VARCHAR2(30) PRIMARY KEY NOT NULL,
    EventTime VARCHAR2(10),
    EventDate DATE
)
```

6.9 <*Branches*> TABLE

```
CREATE TABLE branches (
    branch_num INT PRIMARY KEY,
    work_hours VARCHAR(50),
    city VARCHAR(50),
    street VARCHAR(50)
)

ALTER TABLE branches
ADD employee_id NUMBER(7);

ALTER TABLE branches
ADD CONSTRAINT fk_employee
FOREIGN KEY (employee_id)
REFERENCES Employee(EmployeeID);
```



```
CREATE TABLE branches (
    branch_num INT PRIMARY KEY,
    work_hours VARCHAR(50),
    city VARCHAR(50),
    street VARCHAR(50)
)
```



```
ALTER TABLE branches
ADD employee_id NUMBER(7)
```

```
ALTER TABLE branches
ADD CONSTRAINT fk_employee
FOREIGN KEY (employee_id)
REFERENCES Employee(EmployeeID)
```

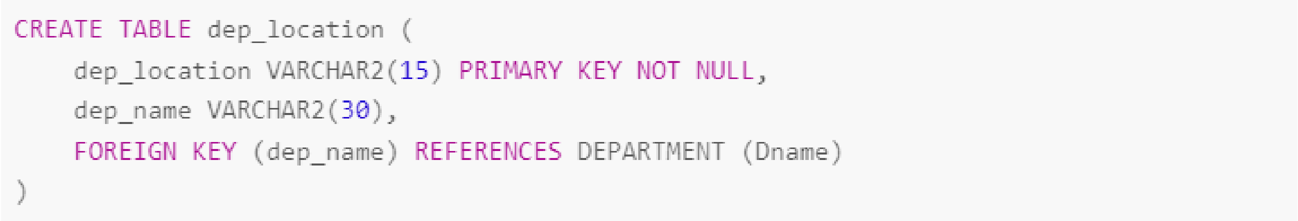
6.10 <*Equipment*> TABLE

```
CREATE TABLE equipment (
    equipment_name VARCHAR(255),
    equipment_id INT PRIMARY KEY
    branch_num INT,
    CONSTRAINT fk_branch_num FOREIGN KEY (branch_num) REFERENCES branches(branch_num)
)
```

```
CREATE TABLE equipment (
    equipment_name VARCHAR(255),
    equipment_id INT PRIMARY KEY,
    branch_num INT,
    CONSTRAINT fk_branch_num FOREIGN KEY (branch_num) REFERENCES branches(branch_num)
)
```

6.11 <*DEP_LOCATION*> TABLE

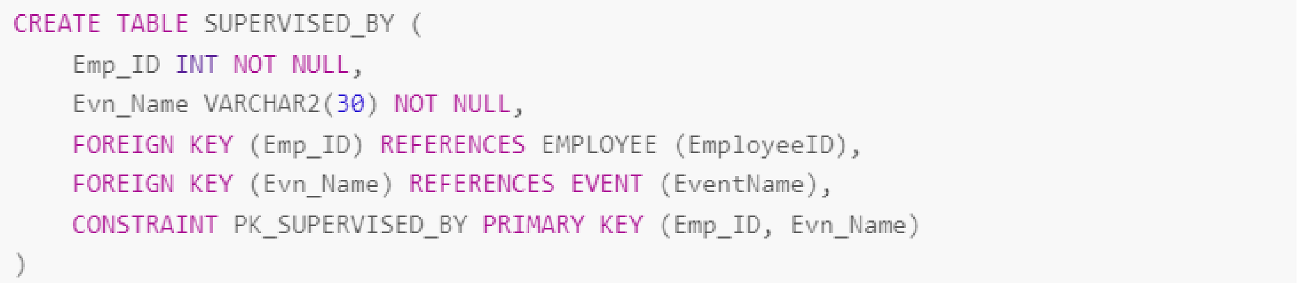
```
CREATE TABLE dep_location (
    dep_location VARCHAR2(15) PRIMARY KEY NOT NULL,
    dep_name VARCHAR2(30),
    FOREIGN KEY (dep_name) REFERENCES DEPARTMENT (Dname)
)
```



```
CREATE TABLE dep_location (
    dep_location VARCHAR2(15) PRIMARY KEY NOT NULL,
    dep_name VARCHAR2(30),
    FOREIGN KEY (dep_name) REFERENCES DEPARTMENT (Dname)
)
```

6.12 <*SUPERVISED_BY*> TABLE

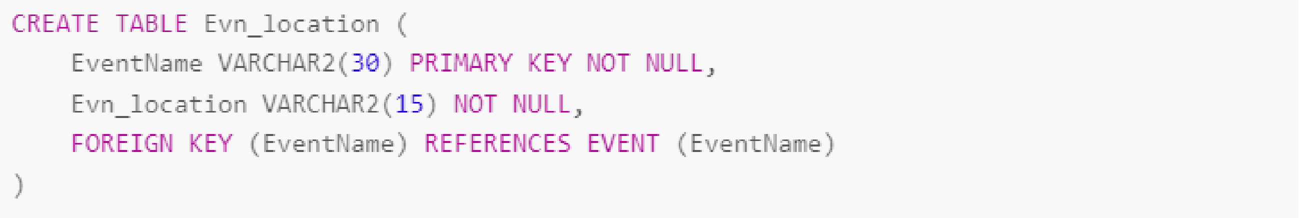
```
CREATE TABLE SUPERVISED_BY (
    Emp_ID INT NOT NULL,
    Evn_Name VARCHAR2(30) NOT NULL,
    FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE (EmployeeID),
    FOREIGN KEY (Evn_Name) REFERENCES EVENT (EventName),
    CONSTRAINT PK_SUPERVISED_BY PRIMARY KEY (Emp_ID, Evn_Name)
)
```



```
CREATE TABLE SUPERVISED_BY (
    Emp_ID INT NOT NULL,
    Evn_Name VARCHAR2(30) NOT NULL,
    FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE (EmployeeID),
    FOREIGN KEY (Evn_Name) REFERENCES EVENT (EventName),
    CONSTRAINT PK_SUPERVISED_BY PRIMARY KEY (Emp_ID, Evn_Name)
)
```

6.13 <*EVN_LOCATION*> TABLE

```
CREATE TABLE Evn_location (
    EventName VARCHAR2(30) PRIMARY KEY NOT NULL,
    Evn_location VARCHAR2(15) NOT NULL,
    FOREIGN KEY (EventName) REFERENCES EVENT (EventName)
)
```



```
CREATE TABLE Evn_location (
    EventName VARCHAR2(30) PRIMARY KEY NOT NULL,
    Evn_location VARCHAR2(15) NOT NULL,
    FOREIGN KEY (EventName) REFERENCES EVENT (EventName)
)
```

7 Constraints Script

In this subsection, show how the business rules have been translated into SQL script. Refer to section 2.2.

Business Rule	SQL Script	Table
Each member **has** records	FOREIGN KEY (MemberID) REFERENCES Member (MemberID) ON DELETE CASCADE	Records
Each member **has** membership	FOREIGN KEY (MemberID) REFERENCES Member(MemberID)	Membership
Each member can **make** multiple reservations	Mbr_ID number (10) REFERENCES Member(MemberID)	Reservations
Branches **has** multiple members	FOREIGN KEY (branch_num) REFERENCES branches(branch_num)	Member

Each employee **works in** a branch	ALTER TABLE branches ADD employee_id NUMBER(7);
	ALTER TABLE branches ADD CONSTRAINT fk_employee FOREIGN KEY (employee_id) REFERENCES Employee(EmployeeID);
		Branches

Each equipment **located in** branch	branch_num INT, CONSTRAINT fk_branch_num FOREIGN KEY (branch_num) REFERENCES branches(branch_num)	Equipment
Each gym class in **managed by** an employee	Emp_ID number (7) REFERENCES Employee(EmployeeID)	Gym Class
Each event is **supervised by** employees	CREATE TABLE SUPERVISED_BY (Emp_ID INT NOT NULL, Evn_Name VARCHAR2(30) NOT NULL, FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE (EmployeeID), FOREIGN KEY (Evn_Name) REFERENCES EVENT (EventName), CONSTRAINT PK_SUPERVISED_BY PRIMARY KEY (Emp_ID, Evn_Name))	Supervised_by
Each employee **works for** department	FOREIGN KEY (Dname) REFERENCES Department(Dname)	Employee
Each employee **manages** department	Emp_ID INT, FOREIGN KEY (Emp_ID) REFERENCES EMPLOYEE (EmployeeID)	Department
Each gym class **have** reservations	Class_No number (2) REFERENCES Gym_Class(Class_Number)	Reservations

8 Queries

In this subsection, I will provide five different SQL queries that demonstrate potential outputs based on the system described in Section 1.4.

8.1 < Retrieve the top 3 blood types >

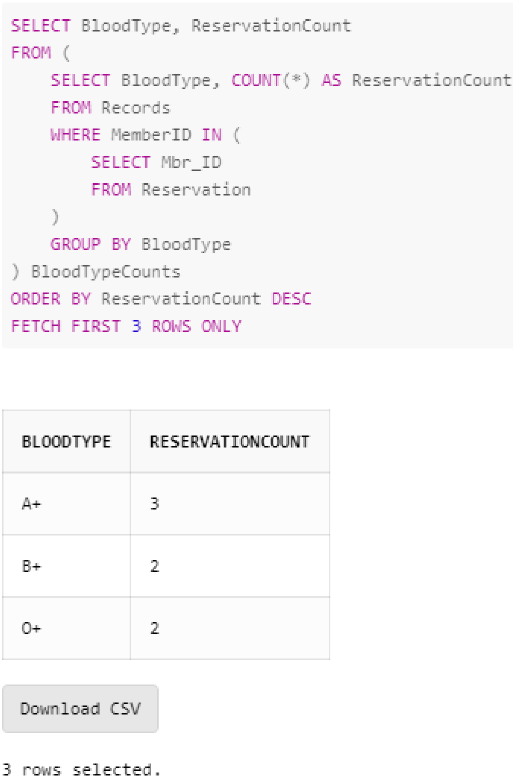
Query in natural language (ENGLISH)

This query retrieves the blood types, and their corresponding reservation counts from the database. It first selects the blood type and counts the number of reservations for each blood type. The inner subquery finds the member IDs from the "Reservation" table. The outer query then groups the results by blood type and orders them based on the reservation count in descending order. Finally, it fetches only the top 3 rows with the highest reservation counts. In simpler terms, the query is used to find the top 3 blood types with the highest number of reservations.

SQL script

```
SELECT BloodType, ReservationCount
FROM (
    SELECT BloodType, COUNT(*) AS ReservationCount
    FROM Records
    WHERE MemberID IN (
        SELECT Mbr_ID
        FROM Reservation
    )
    GROUP BY BloodType
) BloodTypeCounts
ORDER BY ReservationCount DESC
FETCH FIRST 3 ROWS ONLY
```

Caption of the first five rows of the output



```
SELECT BloodType, ReservationCount
FROM (
    SELECT BloodType, COUNT(*) AS ReservationCount
    FROM Records
    WHERE MemberID IN (
        SELECT Mbr_ID
        FROM Reservation
    )
    GROUP BY BloodType
) BloodTypeCounts
ORDER BY ReservationCount DESC
FETCH FIRST 3 ROWS ONLY
```

BLOODTYPE	RESERVATIONCOUNT
A+	3
B+	2
O+	2

3 rows selected.

8.2 < Highest Salary >

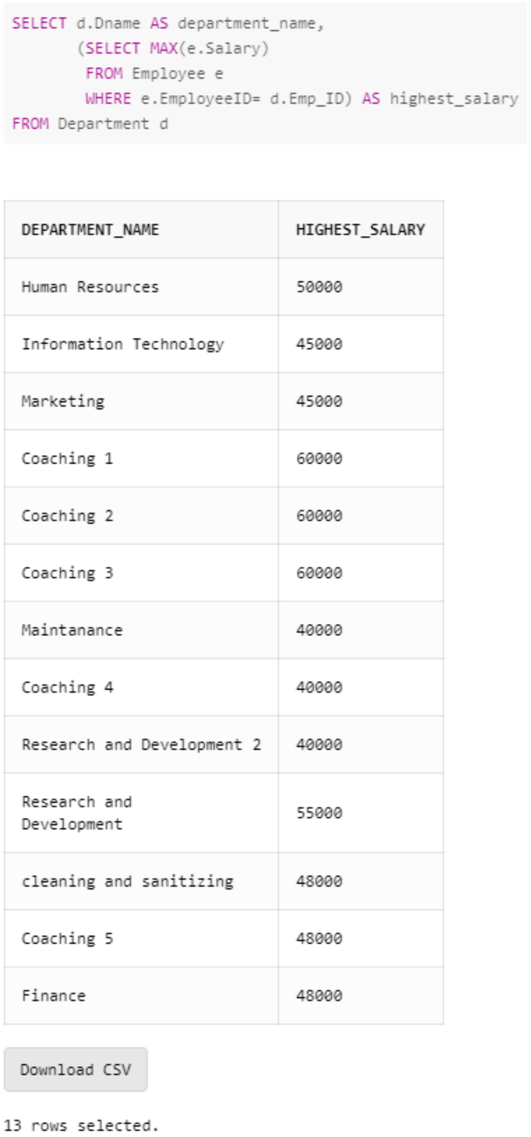
Query in natural language (ENGLISH)

The query selects the department name (Dname) from the Department table and aliases it as department_name. It also includes a subquery that calculates the maximum salary (MAX(e.Salary)) from the Employee table for each department's manager (d.Emp_ID). This subquery filters employees based on the condition e.EmployeeID = d.Emp_ID, where e.EmployeeID is the employee's ID in the Employee table and d.Emp_ID is the manager's ID in the Department table. The result of this subquery is aliased as highest_salary. Overall, the query retrieves the department name along with the highest salary observed among all employees managed by each department's manager.

SQL script

```
SELECT d.Dname AS department_name,
       (SELECT MAX(e.Salary)
        FROM Employee e
        WHERE e.EmployeeID= d.Emp_ID) AS highest_salary
FROM Department d
```

Caption of the first five rows of the output



```
SELECT d.Dname AS department_name,
       (SELECT MAX(e.Salary)
        FROM Employee e
        WHERE e.EmployeeID= d.Emp_ID) AS highest_salary
FROM Department d
```

DEPARTMENT_NAME	HIGHEST_SALARY
Human Resources	50000
Information Technology	45000
Marketing	45000
Coaching 1	60000
Coaching 2	60000
Coaching 3	60000
Maintanance	40000
Coaching 4	40000
Research and Development 2	40000
Research and Development	55000
cleaning and sanitizing	48000
Coaching 5	48000
Finance	48000

13 rows selected.

8.3 < Increasing salary >

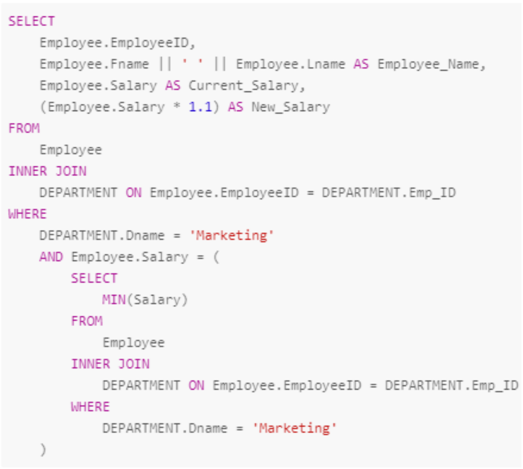
Query in natural language (ENGLISH)

This query retrieves the EmployeeID, name, current salary, and new salary (after a 10% increase) of employees working in the Marketing department whose salary is the lowest among all employees in that department. Here's how it works:
- It selects EmployeeID and concatenates the first and last names of employees from the Employee table.
- It also selects the current salary from the Employee table.
- It calculates the new salary by multiplying the current salary by 1.1 (which represents a 10% increase).
- It filters the results to only include employees who work in the Marketing department.
- Within this filter, it ensures that the employee's salary matches the minimum salary among all employees in the Marketing department.
This query provides a concise way to identify employees with the lowest salary in the Marketing department and calculate their new salary after a raise.

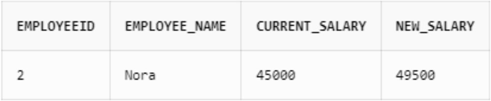
SQL script

```
SELECT
    Employee.EmployeeID,
    Employee.Fname || ' ' || Employee.Lname AS Employee_Name,
    Employee.Salary AS Current_Salary,
    (Employee.Salary * 1.1) AS New_Salary
FROM
    Employee
INNER JOIN
    DEPARTMENT ON Employee.EmployeeID = DEPARTMENT.Emp_ID
WHERE
    DEPARTMENT.Dname = 'Marketing'
    AND Employee.Salary = (
        SELECT
            MIN(Salary)
        FROM
            Employee
        INNER JOIN
            DEPARTMENT ON Employee.EmployeeID = DEPARTMENT.Emp_ID
        WHERE
            DEPARTMENT.Dname = 'Marketing'
    )
```

Caption of the first five rows of the output



```
SELECT
    Employee.EmployeeID,
    Employee.Fname || ' ' || Employee.Lname AS Employee_Name,
    Employee.Salary AS Current_Salary,
    (Employee.Salary * 1.1) AS New_Salary
FROM
    Employee
INNER JOIN
    DEPARTMENT ON Employee.EmployeeID = DEPARTMENT.Emp_ID
WHERE
    DEPARTMENT.Dname = 'Marketing'
    AND Employee.Salary = (
        SELECT
            MIN(Salary)
        FROM
            Employee
        INNER JOIN
            DEPARTMENT ON Employee.EmployeeID = DEPARTMENT.Emp_ID
        WHERE
            DEPARTMENT.Dname = 'Marketing'
    )
```



EMPLOYEEID	EMPLOYEE_NAME	CURRENT_SALARY	NEW_SALARY
2	Nora	45000	49500

8.4 < largest number of members >

<u>Query in natural language (ENGLISH)</u>

This query retrieves information about branches, including the branch number, city, street, and the total number of members associated with each branch. The query uses a join operation between the "branches" table and the "Member" table, linking them based on the branch number.

The result is grouped by the branch number, city, and street using the GROUP BY clause. This allows us to calculate the count of members for each branch using the COUNT() function, which is aliased as "total_members".

The HAVING clause is used to filter the result based on a condition. In this case, it checks if the count of members for a branch is equal to the maximum member count across all branches.

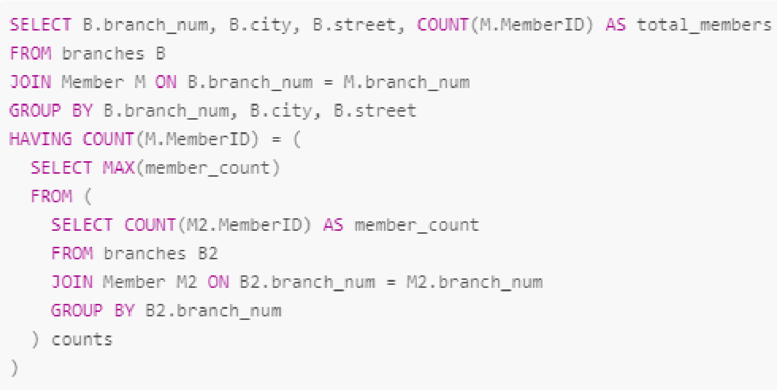
To find the maximum member count, an inner query is used. This inner query calculates the member count for each branch by joining the "branches" table with the "Member" table and grouping by the branch number. The result is then used in the outer query to determine the branch with the highest member count.

Overall, the query retrieves branch information and filters the result to only include branches with the highest member count.

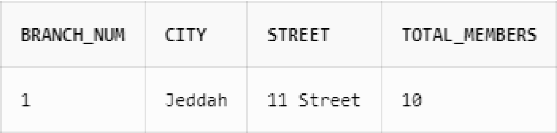
<u>SQL script</u>

```sql
SELECT B.branch_num, B.city, B.street, COUNT(M.MemberID) AS total_members
FROM branches B
JOIN Member M ON B.branch_num = M.branch_num
GROUP BY B.branch_num, B.city, B.street
HAVING COUNT(M.MemberID) = (
  SELECT MAX(member_count)
  FROM (
    SELECT COUNT(M2.MemberID) AS member_count
    FROM branches B2
    JOIN Member M2 ON B2.branch_num = M2.branch_num
    GROUP BY B2.branch_num
  ) counts
)
```

<u>Caption of the first five rows of the output</u>



```sql
SELECT B.branch_num, B.city, B.street, COUNT(M.MemberID) AS total_members
FROM branches B
JOIN Member M ON B.branch_num = M.branch_num
GROUP BY B.branch_num, B.city, B.street
HAVING COUNT(M.MemberID) = (
  SELECT MAX(member_count)
  FROM (
    SELECT COUNT(M2.MemberID) AS member_count
    FROM branches B2
    JOIN Member M2 ON B2.branch_num = M2.branch_num
    GROUP BY B2.branch_num
  ) counts
)
```



BRANCH_NUM	CITY	STREET	TOTAL_MEMBERS
1	Jeddah	11 Street	10

8.5 < Find the coach's name based on reservation >

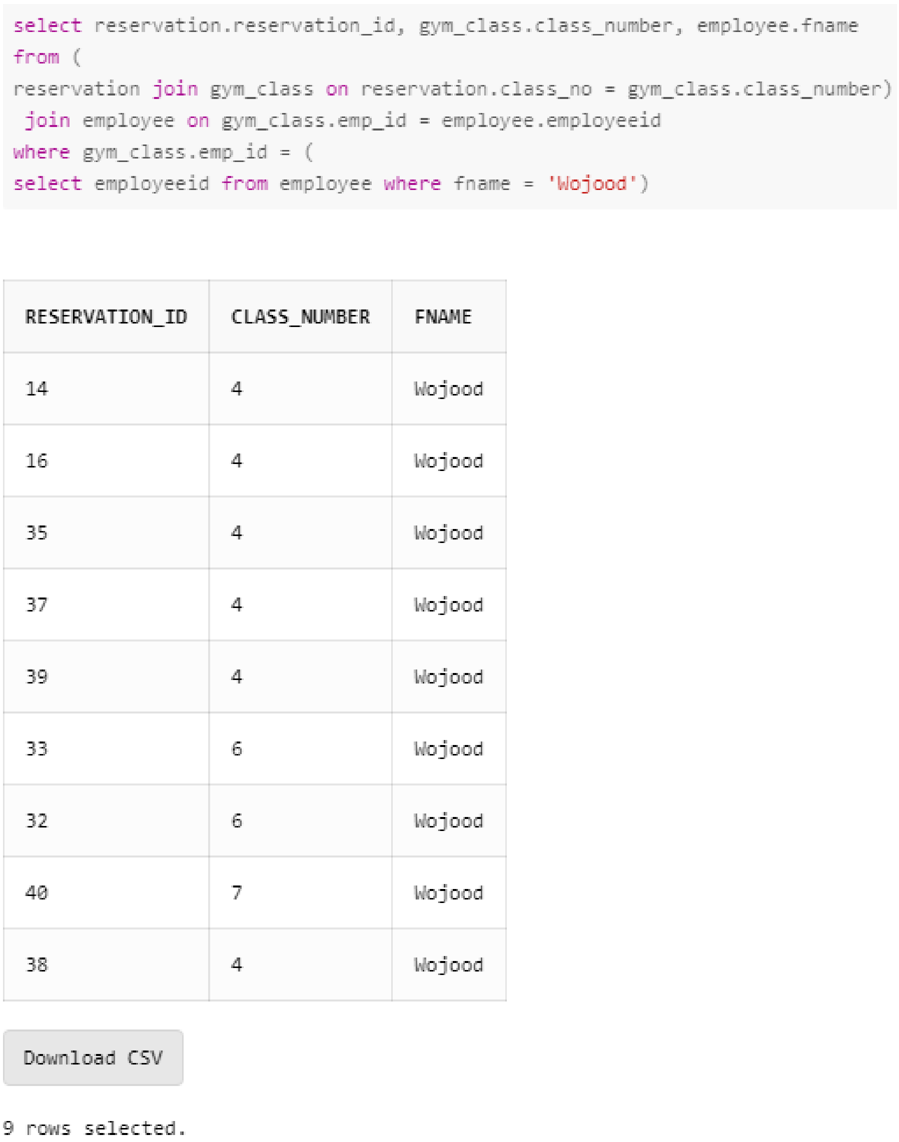
Query in natural language (ENGLISH)

This query retrieves 3 columns which are Reservation _ID from table Reservation, Class_number from Gym_class table, and F_name (First name of the coach) from Employee table. Therefore, there are 2 joined tables in the query. The first one is between Reservation and Gym_class. The second one is between Gym class and the Employee. In the "WHERE" command, there is an inner query that retrieves the employee ID for a specific coach. Finally, the inner query is compared with the employee ID in the Gym class table to display the outer query.

SQL script

```
select reservation.reservation_id, gym_class.class_number, employee.fname
from (
reservation join gym_class on reservation.class_no = gym_class.class_number)
 join employee on gym_class.emp_id = employee.employeeid
where gym_class.emp_id = (
select employeeid from employee where fname = 'Wojood')
```

Caption of the first five rows of the output



```
select reservation.reservation_id, gym_class.class_number, employee.fname
from (
reservation join gym_class on reservation.class_no = gym_class.class_number)
 join employee on gym_class.emp_id = employee.employeeid
where gym_class.emp_id = (
select employeeid from employee where fname = 'Wojood')
```

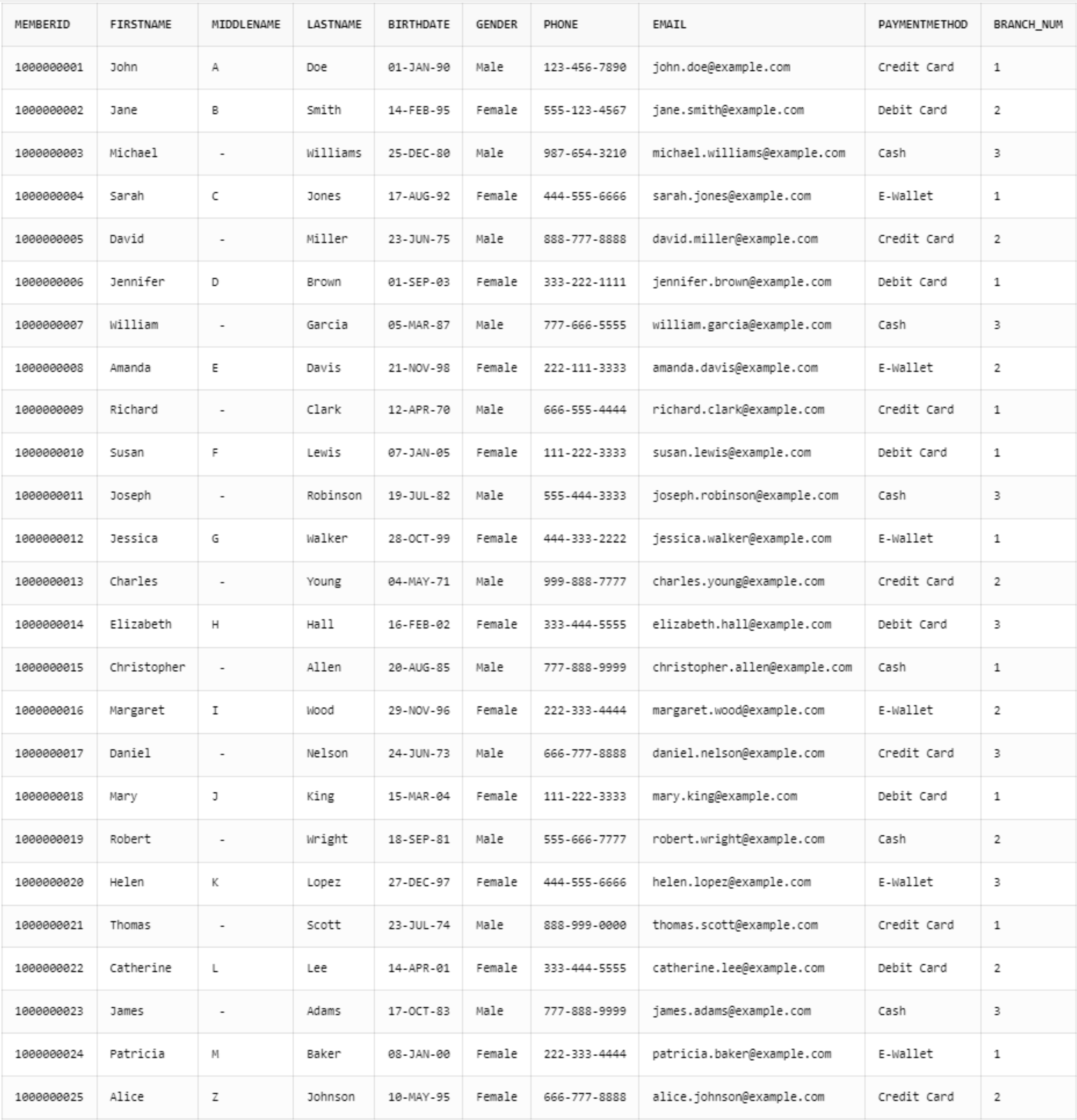
RESERVATION_ID	CLASS_NUMBER	FNAME
14	4	Wojood
16	4	Wojood
35	4	Wojood
37	4	Wojood
39	4	Wojood
33	6	Wojood
32	6	Wojood
40	7	Wojood
38	4	Wojood

```
9 rows selected.
```

APPENDIX

For each table, listing all the rows.

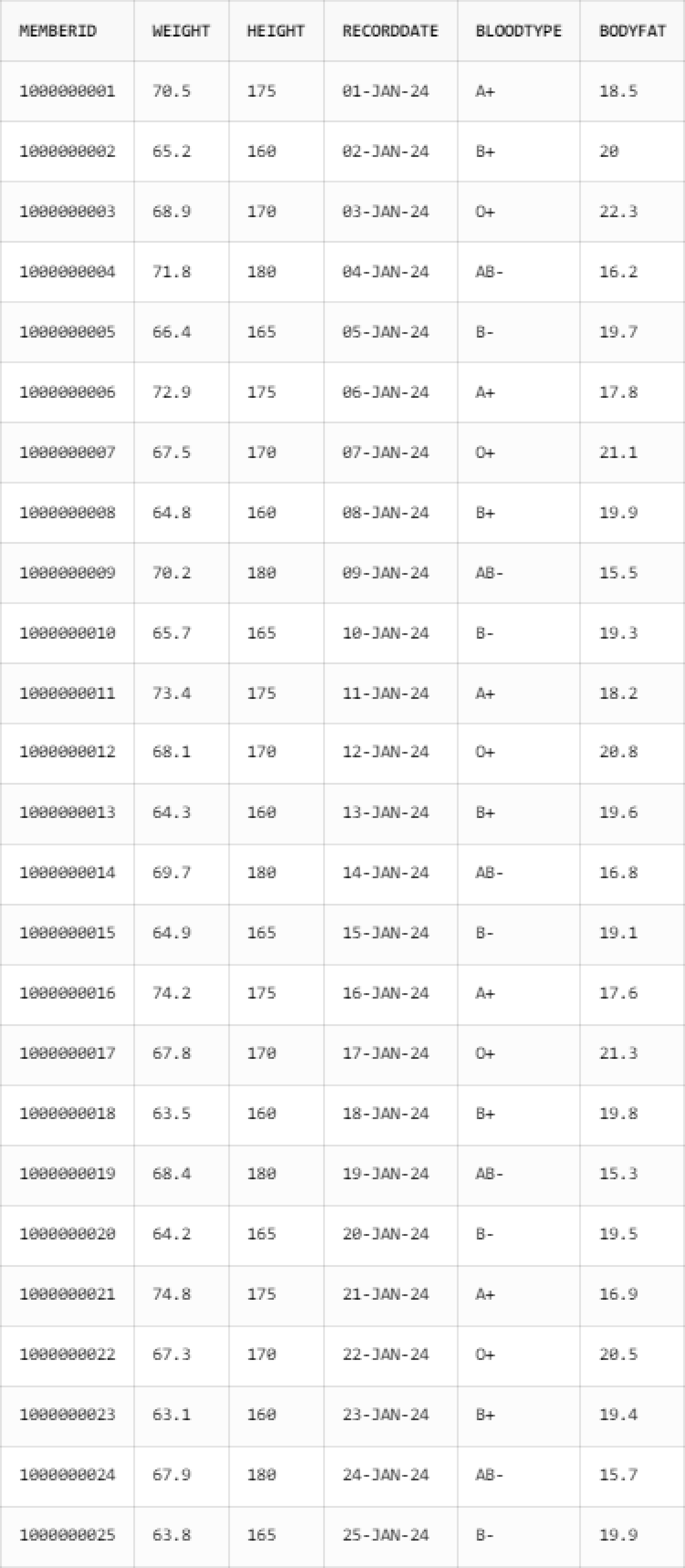
Member



MEMBERID	FIRSTNAME	MIDDLENAME	LASTNAME	BIRTHDATE	GENDER	PHONE	EMAIL	PAYMENTMETHOD	BRANCH_NUM
1000000001	John	A	Doe	01-JAN-90	Male	123-456-7890	john.doe@example.com	Credit Card	1
1000000002	Jane	B	Smith	14-FEB-95	Female	555-123-4567	jane.smith@example.com	Debit Card	2
1000000003	Michael	-	Williams	25-DEC-80	Male	987-654-3210	michael.williams@example.com	Cash	3
1000000004	Sarah	C	Jones	17-AUG-92	Female	444-555-6666	sarah.jones@example.com	E-Wallet	1
1000000005	David	-	Miller	23-JUN-75	Male	888-777-8888	david.miller@example.com	Credit Card	2
1000000006	Jennifer	D	Brown	01-SEP-03	Female	333-222-1111	jennifer.brown@example.com	Debit Card	1
1000000007	William	-	Garcia	05-MAR-87	Male	777-666-5555	william.garcia@example.com	Cash	3
1000000008	Amanda	E	Davis	21-NOV-98	Female	222-111-3333	amanda.davis@example.com	E-Wallet	2
1000000009	Richard	-	Clark	12-APR-70	Male	666-555-4444	richard.clark@example.com	Credit Card	1
1000000010	Susan	F	Lewis	07-JAN-05	Female	111-222-3333	susan.lewis@example.com	Debit Card	1
1000000011	Joseph	-	Robinson	19-JUL-82	Male	555-444-3333	joseph.robinson@example.com	Cash	3
1000000012	Jessica	G	Walker	28-OCT-99	Female	444-333-2222	jessica.walker@example.com	E-Wallet	1
1000000013	Charles	-	Young	04-MAY-71	Male	999-888-7777	charles.young@example.com	Credit Card	2
1000000014	Elizabeth	H	Hall	16-FEB-02	Female	333-444-5555	elizabeth.hall@example.com	Debit Card	3
1000000015	Christopher	-	Allen	20-AUG-85	Male	777-888-9999	christopher.allen@example.com	Cash	1
1000000016	Margaret	I	Wood	29-NOV-96	Female	222-333-4444	margaret.wood@example.com	E-Wallet	2
1000000017	Daniel	-	Nelson	24-JUN-73	Male	666-777-8888	daniel.nelson@example.com	Credit Card	3
1000000018	Mary	J	King	15-MAR-04	Female	111-222-3333	mary.king@example.com	Debit Card	1
1000000019	Robert	-	Wright	18-SEP-81	Male	555-666-7777	robert.wright@example.com	Cash	2
1000000020	Helen	K	Lopez	27-DEC-97	Female	444-555-6666	helen.lopez@example.com	E-Wallet	3
1000000021	Thomas	-	Scott	23-JUL-74	Male	888-999-0000	thomas.scott@example.com	Credit Card	1
1000000022	Catherine	L	Lee	14-APR-01	Female	333-444-5555	catherine.lee@example.com	Debit Card	2
1000000023	James	-	Adams	17-OCT-83	Male	777-888-9999	james.adams@example.com	Cash	3
1000000024	Patricia	M	Baker	08-JAN-00	Female	222-333-4444	patricia.baker@example.com	E-Wallet	1
1000000025	Alice	Z	Johnson	10-MAY-95	Female	666-777-8888	alice.johnson@example.com	Credit Card	2

MEMBERID	FIRSTNAME	MIDDLENAME	LASTNAME	BIRTHDATE	GENDER	PHONE	EMAIL	PAYMENTMETHOD	BRANCH_NUM
1000000001	John	A	Doe	01-JAN-90	Male	123-456-7890	john.doe@example.com	Credit Card	1
1000000002	Jane	B	Smith	14-FEB-95	Female	555-123-4567	jane.smith@example.com	Debit Card	2
1000000003	Michael	-	Williams	25-DEC-80	Male	987-654-3210	michael.williams@example.com	Cash	3
1000000004	Sarah	C	Jones	17-AUG-92	Female	444-555-6666	sarah.jones@example.com	E-Wallet	1
1000000005	David	-	Miller	23-JUN-75	Male	888-777-8888	david.miller@example.com	Credit Card	2
1000000006	Jennifer	D	Brown	01-SEP-03	Female	333-222-1111	jennifer.brown@example.com	Debit Card	1
1000000007	William	-	Garcia	05-MAR-87	Male	777-666-5555	william.garcia@example.com	Cash	3
1000000008	Amanda	E	Davis	21-NOV-98	Female	222-111-3333	amanda.davis@example.com	E-Wallet	2
1000000009	Richard	-	Clark	12-APR-70	Male	666-555-4444	richard.clark@example.com	Credit Card	1
1000000010	Susan	F	Lewis	07-JAN-05	Female	111-222-3333	susan.lewis@example.com	Debit Card	1
1000000011	Joseph	-	Robinson	19-JUL-82	Male	555-444-3333	joseph.robinson@example.com	Cash	3
1000000012	Jessica	G	Walker	28-OCT-99	Female	444-333-2222	jessica.walker@example.com	E-Wallet	1
1000000013	Charles	-	Young	04-MAY-71	Male	999-888-7777	charles.young@example.com	Credit Card	2
1000000014	Elizabeth	H	Hall	16-FEB-02	Female	333-444-5555	elizabeth.hall@example.com	Debit Card	3
1000000015	Christopher	-	Allen	20-AUG-85	Male	777-888-9999	christopher.allen@example.com	Cash	1
1000000016	Margaret	I	Wood	29-NOV-96	Female	222-333-4444	margaret.wood@example.com	E-Wallet	2
1000000017	Daniel	-	Nelson	24-JUN-73	Male	666-777-8888	daniel.nelson@example.com	Credit Card	3
1000000018	Mary	J	King	15-MAR-04	Female	111-222-3333	mary.king@example.com	Debit Card	1
1000000019	Robert	-	Wright	18-SEP-81	Male	555-666-7777	robert.wright@example.com	Cash	2
1000000020	Helen	K	Lopez	27-DEC-97	Female	444-555-6666	helen.lopez@example.com	E-Wallet	3
1000000021	Thomas	-	Scott	23-JUL-74	Male	888-999-0000	thomas.scott@example.com	Credit Card	1
1000000022	Catherine	L	Lee	14-APR-01	Female	333-444-5555	catherine.lee@example.com	Debit Card	2
1000000023	James	-	Adams	17-OCT-83	Male	777-888-9999	james.adams@example.com	Cash	3
1000000024	Patricia	M	Baker	08-JAN-00	Female	222-333-4444	patricia.baker@example.com	E-Wallet	1
1000000025	Alice	Z	Johnson	10-MAY-95	Female	666-777-8888	alice.johnson@example.com	Credit Card	2

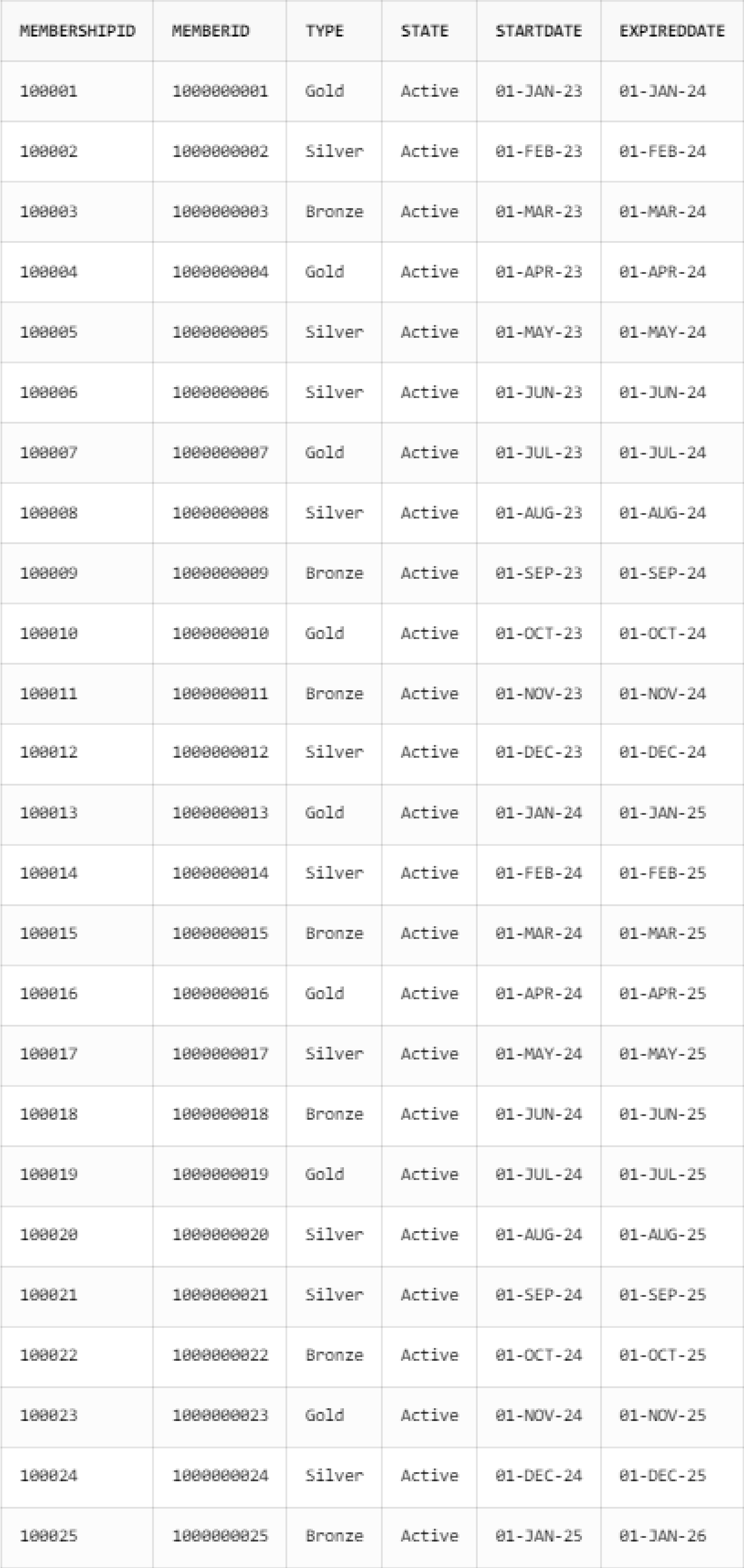
Records



MEMBERID	WEIGHT	HEIGHT	RECORDDATE	BLOODTYPE	BODYFAT
1000000001	70.5	175	01-JAN-24	A+	18.5
1000000002	65.2	160	02-JAN-24	B+	20
1000000003	68.9	170	03-JAN-24	O+	22.3
1000000004	71.8	180	04-JAN-24	AB-	16.2
1000000005	66.4	165	05-JAN-24	B-	19.7
1000000006	72.9	175	06-JAN-24	A+	17.8
1000000007	67.5	170	07-JAN-24	O+	21.1
1000000008	64.8	160	08-JAN-24	B+	19.9
1000000009	70.2	180	09-JAN-24	AB-	15.5
1000000010	65.7	165	10-JAN-24	B-	19.3
1000000011	73.4	175	11-JAN-24	A+	18.2
1000000012	68.1	170	12-JAN-24	O+	20.8
1000000013	64.3	160	13-JAN-24	B+	19.6
1000000014	69.7	180	14-JAN-24	AB-	16.8
1000000015	64.9	165	15-JAN-24	B-	19.1
1000000016	74.2	175	16-JAN-24	A+	17.6
1000000017	67.8	170	17-JAN-24	O+	21.3
1000000018	63.5	160	18-JAN-24	B+	19.8
1000000019	68.4	180	19-JAN-24	AB-	15.3
1000000020	64.2	165	20-JAN-24	B-	19.5
1000000021	74.8	175	21-JAN-24	A+	16.9
1000000022	67.3	170	22-JAN-24	O+	20.5
1000000023	63.1	160	23-JAN-24	B+	19.4
1000000024	67.9	180	24-JAN-24	AB-	15.7
1000000025	63.8	165	25-JAN-24	B-	19.9

MEMBERID	WEIGHT	HEIGHT	RECORDDATE	BLOODTYPE	BODYFAT
1000000001	70.5	175	01-JAN-24	A+	18.5
1000000002	65.2	160	02-JAN-24	B+	20
1000000003	68.9	170	03-JAN-24	O+	22.3
1000000004	71.8	180	04-JAN-24	AB-	16.2
1000000005	66.4	165	05-JAN-24	B-	19.7
1000000006	72.9	175	06-JAN-24	A+	17.8
1000000007	67.5	170	07-JAN-24	O+	21.1
1000000008	64.8	160	08-JAN-24	B+	19.9
1000000009	70.2	180	09-JAN-24	AB-	15.5
1000000010	65.7	165	10-JAN-24	B-	19.3
1000000011	73.4	175	11-JAN-24	A+	18.2
1000000012	68.1	170	12-JAN-24	O+	20.8
1000000013	64.3	160	13-JAN-24	B+	19.6
1000000014	69.7	180	14-JAN-24	AB-	16.8
1000000015	64.9	165	15-JAN-24	B-	19.1
1000000016	74.2	175	16-JAN-24	A+	17.6
1000000017	67.8	170	17-JAN-24	O+	21.3
1000000018	63.5	160	18-JAN-24	B+	19.8
1000000019	68.4	180	19-JAN-24	AB-	15.3
1000000020	64.2	165	20-JAN-24	B-	19.5
1000000021	74.8	175	21-JAN-24	A+	16.9
1000000022	67.3	170	22-JAN-24	O+	20.5
1000000023	63.1	160	23-JAN-24	B+	19.4
1000000024	67.9	180	24-JAN-24	AB-	15.7
1000000025	63.8	165	25-JAN-24	B-	19.9

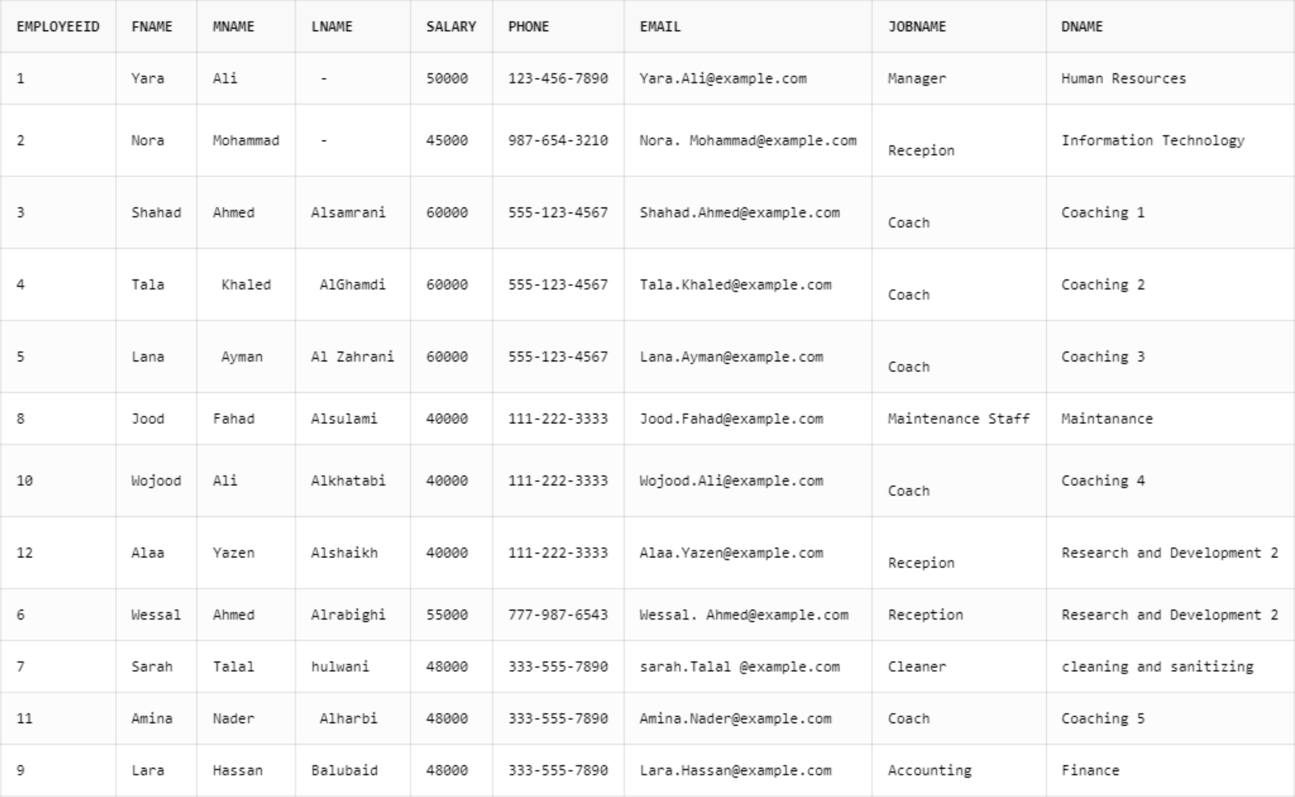
Membership



MEMBERSHIPID	MEMBERID	TYPE	STATE	STARTDATE	EXPIREDDATE
100001	1000000001	Gold	Active	01-JAN-23	01-JAN-24
100002	1000000002	Silver	Active	01-FEB-23	01-FEB-24
100003	1000000003	Bronze	Active	01-MAR-23	01-MAR-24
100004	1000000004	Gold	Active	01-APR-23	01-APR-24
100005	1000000005	Silver	Active	01-MAY-23	01-MAY-24
100006	1000000006	Silver	Active	01-JUN-23	01-JUN-24
100007	1000000007	Gold	Active	01-JUL-23	01-JUL-24
100008	1000000008	Silver	Active	01-AUG-23	01-AUG-24
100009	1000000009	Bronze	Active	01-SEP-23	01-SEP-24
100010	1000000010	Gold	Active	01-OCT-23	01-OCT-24
100011	1000000011	Bronze	Active	01-NOV-23	01-NOV-24
100012	1000000012	Silver	Active	01-DEC-23	01-DEC-24
100013	1000000013	Gold	Active	01-JAN-24	01-JAN-25
100014	1000000014	Silver	Active	01-FEB-24	01-FEB-25
100015	1000000015	Bronze	Active	01-MAR-24	01-MAR-25
100016	1000000016	Gold	Active	01-APR-24	01-APR-25
100017	1000000017	Silver	Active	01-MAY-24	01-MAY-25
100018	1000000018	Bronze	Active	01-JUN-24	01-JUN-25
100019	1000000019	Gold	Active	01-JUL-24	01-JUL-25
100020	1000000020	Silver	Active	01-AUG-24	01-AUG-25
100021	1000000021	Silver	Active	01-SEP-24	01-SEP-25
100022	1000000022	Bronze	Active	01-OCT-24	01-OCT-25
100023	1000000023	Gold	Active	01-NOV-24	01-NOV-25
100024	1000000024	Silver	Active	01-DEC-24	01-DEC-25
100025	1000000025	Bronze	Active	01-JAN-25	01-JAN-26

EMPLOYEE ID	FNAME	MNAME	LNAME	SALARY	PHONE	EMAIL	JOBNAME	DNAME
1	Yara	Ali	-	50000	123-456-7890	Yara.Ali@example.com	Manager	Human Resources
2	Nora	Mohammad	-	45000	987-654-3210	Nora.Mohammad@example.com	Reception	Information Technology
3	Shahad	Ahmed	Alsamrani	60000	555-123-4567	Shahad.Ahmed@example.com	Coach	Coaching 1
4	Tala	Khaled	AlGhamdi	60000	555-123-4567	Tala.Khaled@example.com	Coach	Coaching 2
5	Lana	Ayman	Al Zahrani	60000	555-123-4567	Lana.Ayman@example.com	Coach	Coaching 3
8	Jood	Fahad	Alsulami	40000	111-222-3333	Jood.Fahad@example.com	Maintenance Staff	Maintenance
10	Wojood	Ali	Alkhatabi	40000	111-222-3333	Wojood.Ali@example.com	Coach	Coaching 4
12	Alaa	Yazen	Alshaikh	40000	111-222-3333	Alaa.Yazen@example.com	Reception	Research and Development 2
6	Wessal	Ahmed	Alrabighi	55000	777-987-6543	Wessal.Ahmed@example.com	Reception	Research and Development 2
7	Sarah	Talal	hulwani	48000	333-555-7890	sarah.Talal @example.com	Cleaner	cleaning and sanitizing
11	Amina	Nader	Alharbi	48000	333-555-7890	Amina.Nader@example.com	Coach	Coaching 5
9	Lara	Hassan	Balubaid	48000	333-555-7890	Lara.Hassan@example.com	Accounting	Finance

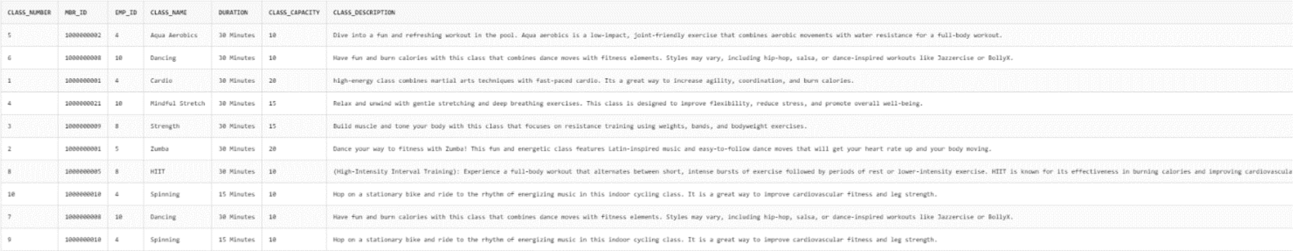
Employee



EMPLOYEEID	FNAME	MNAME	LNAME	SALARY	PHONE	EMAIL	JOBNAME	DNAME
1	Yara	Ali	-	50000	123-456-7890	Yara.Ali@example.com	Manager	Human Resources
2	Nora	Mohammad	-	45000	987-654-3210	Nora. Mohammad@example.com	Recepion	Information Technology
3	Shahad	Ahmed	Alsamrani	60000	555-123-4567	Shahad.Ahmed@example.com	Coach	Coaching 1
4	Tala	Khaled	AlGhamdi	60000	555-123-4567	Tala.Khaled@example.com	Coach	Coaching 2
5	Lana	Ayman	Al Zahrani	60000	555-123-4567	Lana.Ayman@example.com	Coach	Coaching 3
8	Jood	Fahad	Alsulami	40000	111-222-3333	Jood.Fahad@example.com	Maintenance Staff	Maintanance
10	Wojood	Ali	Alkhatabi	40000	111-222-3333	Wojood.Ali@example.com	Coach	Coaching 4
12	Alaa	Yazen	Alshaikh	40000	111-222-3333	Alaa.Yazen@example.com	Recepion	Research and Development 2
6	Wessal	Ahmed	Alrabighi	55000	777-987-6543	Wessal. Ahmed@example.com	Reception	Research and Development 2
7	Sarah	Talal	hulwani	48000	333-555-7890	sarah.Talal @example.com	Cleaner	cleaning and sanitizing
11	Amina	Nader	Alharbi	48000	333-555-7890	Amina.Nader@example.com	Coach	Coaching 5
9	Lara	Hassan	Balubaid	48000	333-555-7890	Lara.Hassan@example.com	Accounting	Finance

EMPLOYEEID	FNAME	MNAME	LNAME	SALARY	PHONE	EMAIL	JOBNAME	DNAME
1	Yara	Ali	-	50000	123-456-7890	Yara.Ali@example.com	Manager	Human Resources
2	Nora	Mohammad	-	45000	987-654-3210	Nora. Mohammad@example.com	Reception	Information Technology
3	Shahad	Ahmed	Alsamrani	60000	555-123-4567	Shahad.Ahmed@example.com	Coach	Coaching 1
4	Tala	Khaled	AlGhamdi	60000	555-123-4567	Tala.Khaled@example.com	Coach	Coaching 2
5	Lana	Ayman	Al Zahrani	60000	555-123-4567	Lana.Ayman@example.com	Coach	Coaching 3
8	Jood	Fahad	Alsulami	40000	111-222-3333	Jood.Fahad@example.com	Maintenance Staff	Maintenance
10	Wojood	Ali	Alkhatabi	40000	111-222-3333	Wojood.Ali@example.com	Coach	Coaching 4
12	Alaa	Yazen	Alshaikh	40000	111-222-3333	Alaa.Yazen@example.com	Reception	Research and Development 2
6	Wessal	Ahmed	Alrabighi	55000	777-987-6543	Wessal. Ahmed@example.com	Reception	Research and Development 2
7	Sarah	Talal	hulwani	48000	333-555-7890	sarah.Talal @example.com	Cleaner	cleaning and sanitizing
11	Amina	Nader	Alharbi	48000	333-555-7890	Amina.Nader@example.com	Coach	Coaching 5
9	Lara	Hassan	Balubaid	48000	333-555-7890	Lara.Hassan@example.com	Accounting	Finance

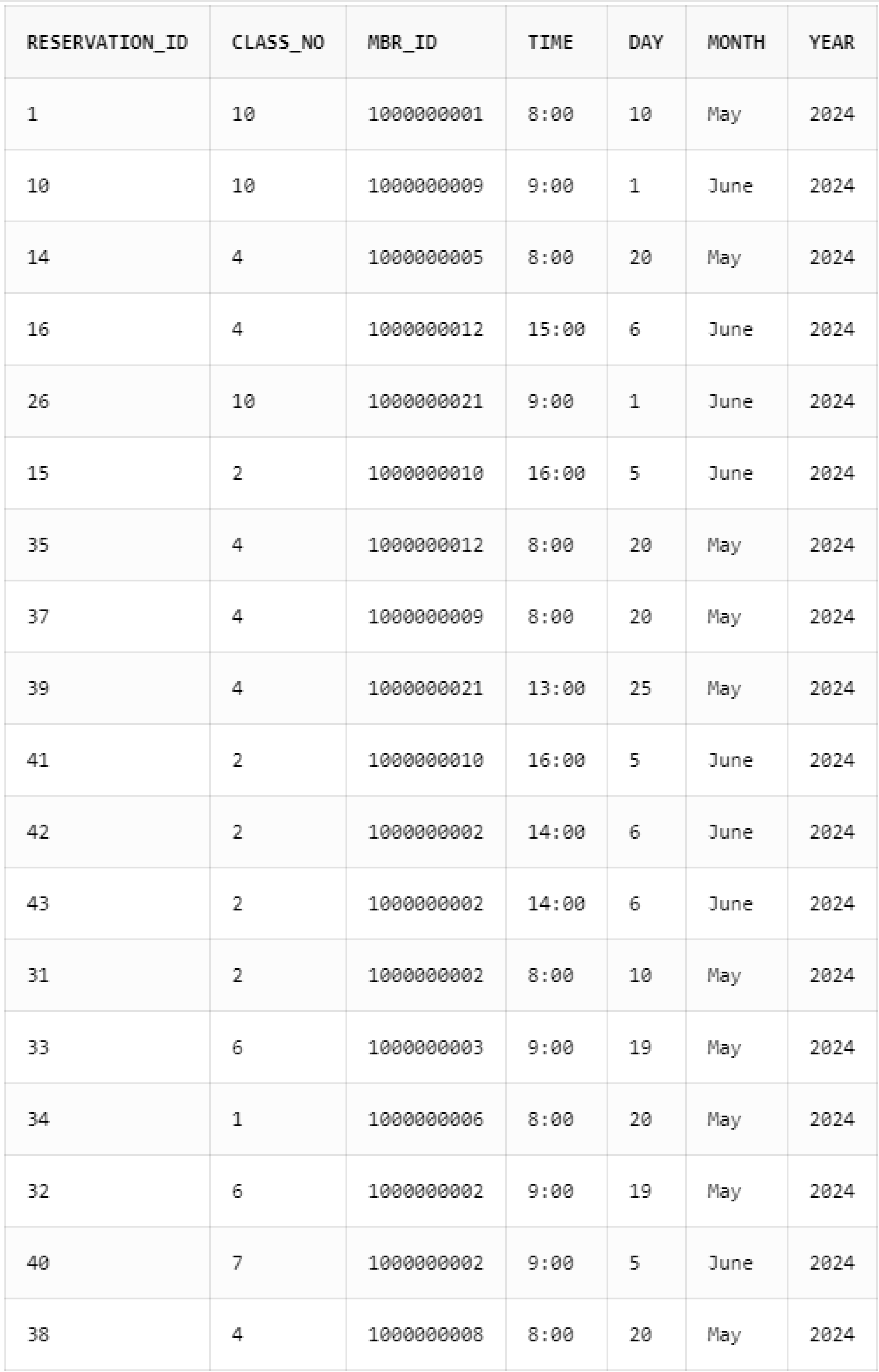
Gym Class



CLASS_NUMBER	MBR_ID	EMP_ID	CLASS_NAME	DURATION	CLASS_CAPACITY	CLASS_DESCRIPTION
5	1000000002	4	Aqua Aerobics	30 Minutes	10	Dive into a fun and refreshing workout in the pool. Aqua aerobics is a low-impact, joint-friendly exercise that combines aerobic movements with water resistance for a full-body workout.
6	1000000008	10	Dancing	30 Minutes	10	Have fun and burn calories with this class that combines dance moves with fitness elements. Styles may vary, including hip-hop, salsa, or dance-inspired workouts like Jazzercise or BollyX.
1	1000000001	4	Cardio	30 Minutes	20	high-energy class combines martial arts techniques with fast-paced cardio. Its a great way to increase agility, coordination, and burn calories.
4	1000000021	10	Mindful Stretch	30 Minutes	15	Relax and unwind with gentle stretching and deep breathing exercises. This class is designed to improve flexibility, reduce stress, and promote overall well-being.
3	1000000009	8	Strength	30 Minutes	15	Build muscle and tone your body with this class that focuses on resistance training using weights, bands, and bodyweight exercises.
2	1000000001	5	Zumba	30 Minutes	20	Dance your way to fitness with Zumba! This fun and energetic class features Latin-inspired music and easy-to-follow dance moves that will get your heart rate up and your body moving.
8	1000000005	8	HIIT	30 Minutes	10	(High-Intensity Interval Training): Experience a full-body workout that alternates between short, intense bursts of exercise followed by periods of rest or lower-intensity exercise. HIIT is known for its effectiveness in burning calories and improving cardiovascula
10	1000000010	4	Spinning	15 Minutes	10	Hop on a stationary bike and ride to the rhythm of energizing music in this indoor cycling class. It is a great way to improve cardiovascular fitness and leg strength.
7	1000000008	10	Dancing	30 Minutes	10	Have fun and burn calories with this class that combines dance moves with fitness elements. Styles may vary, including hip-hop, salsa, or dance-inspired workouts like Jazzercise or BollyX.
9	1000000010	4	Spinning	15 Minutes	10	Hop on a stationary bike and ride to the rhythm of energizing music in this indoor cycling class. It is a great way to improve cardiovascular fitness and leg strength.

CLASS_NUMBER	MBR_ID	EMP_ID	CLASS_NAME	DURATION	CLASS_CAPACITY	CLASS_DESCRIPTION
1	1000000001	4	Cardio	30 Minutes	20	high-energy class combines martial arts techniques with fast-paced cardio. It’s a great way to increase agility, coordination, and burn calories.
2	1000000001	5	Zumba	30 Minutes	20	Dance your way to fitness with Zumba! This fun and energetic class features Latin-inspired music and easy-to-follow dance moves that will get your heart rate up and your body moving.
3	1000000001	5	Zumba	30 Minutes	20	Dance your way to fitness with Zumba! This fun and energetic class features Latin-inspired music and easy-to-follow dance moves that will get your heart rate up and your body moving.
4	1000000001	5	Zumba	30 Minutes	20	Dance your way to fitness with Zumba! This fun and energetic class features Latin-inspired music and easy-to-follow dance moves that will get your heart rate up and your body moving.
5	1000000009	8	Strength	30 Minutes	15	Build muscle and tone your body with this class that focuses on resistance training using weights, bands, and bodyweight exercises.
6	1000000010	12	Spinning	15 Minutes	10	Hop on a stationary bike and ride to the rhythm of energizing music in this indoor cycling class. It is a great way to improve cardiovascular fitness and leg strength.
7	1000000002	4	Aqua Aerobics	30 Minutes	10	Dive into a fun and refreshing workout in the pool. Aqua aerobics is a low-impact, joint-friendly exercise that combines aerobic movements with water resistance for a full-body workout.
8	1000000002	4	Aqua Aerobics	30 Minutes	10	Dive into a fun and refreshing workout in the pool. Aqua aerobics is a low-impact, joint-friendly exercise that combines aerobic movements with water resistance for a full-body workout.
9	1000000008	10	Dancing	30 Minutes	10	Have fun and burn calories with this class that combines dance moves with fitness elements. Styles may vary, including hip-hop, salsa, or dance-inspired workouts like Jazzercise or Bully.
10	1000000008	10	Dancing	30 Minutes	10	Have fun and burn calories with this class that combines dance moves with fitness elements. Styles may vary, including hip-hop, salsa, or dance-inspired workouts like Jazzercise or Bully.
11	1000000005	8	HIIT	30 Minutes	10	(High-Intensity Interval Training): Experience a full-body workout that alternates between short, intense bursts of exercise followed by periods of rest or lower-intensity exercise. HIIT is known for its effectiveness in burning calories and improving cardiovascular fitness.
12	1000000005	8	HIIT	30 Minutes	10	(High-Intensity Interval Training): Experience a full-body workout that alternates between short, intense bursts of exercise followed by periods of rest or lower-intensity exercise. HIIT is known for its effectiveness in burning calories and improving cardiovascular fitness.
13	1000000021	10	Mindful Stretch	30 Minutes	15	Relax and unwind with gentle stretching and deep breathing exercises. This class is designed to improve flexibility, reduce stress, and promote overall well-being.

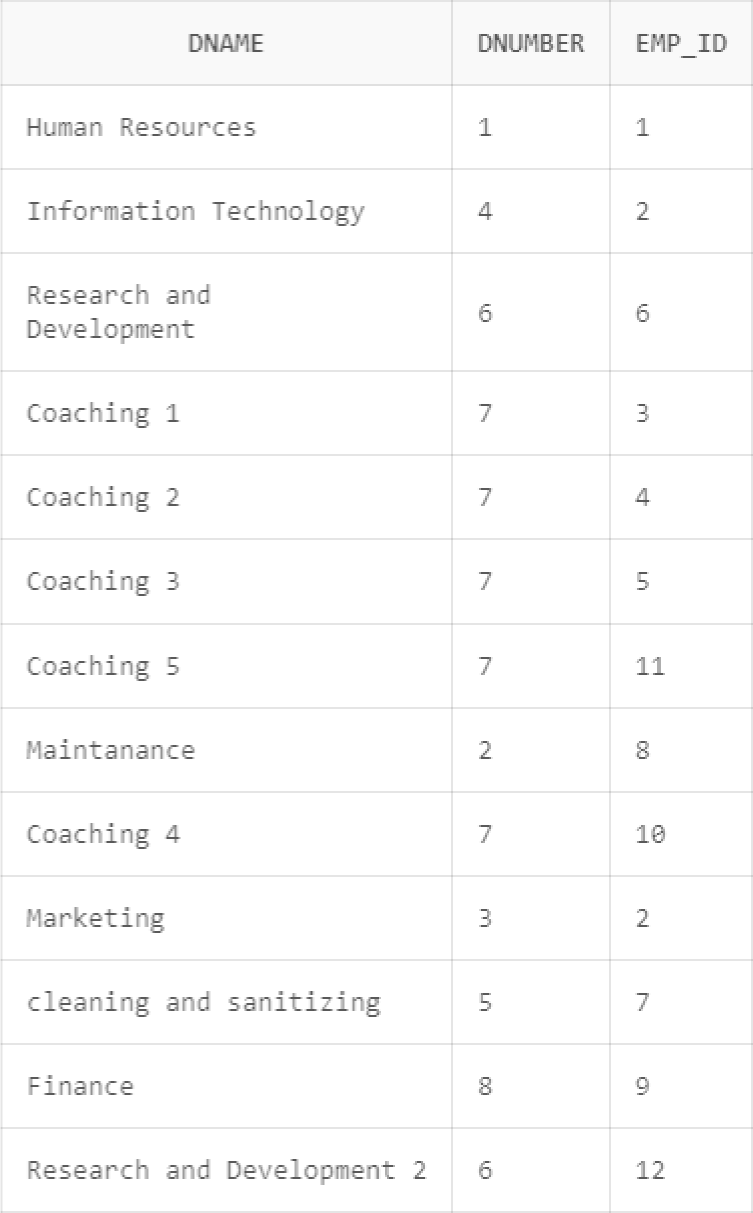
Reservation



RESERVATION_ID	CLASS_NO	MBR_ID	TIME	DAY	MONTH	YEAR
1	10	1000000001	8:00	10	May	2024
10	10	1000000009	9:00	1	June	2024
14	4	1000000005	8:00	20	May	2024
16	4	1000000012	15:00	6	June	2024
26	10	1000000021	9:00	1	June	2024
15	2	1000000010	16:00	5	June	2024
35	4	1000000012	8:00	20	May	2024
37	4	1000000009	8:00	20	May	2024
39	4	1000000021	13:00	25	May	2024
41	2	1000000010	16:00	5	June	2024
42	2	1000000002	14:00	6	June	2024
43	2	1000000002	14:00	6	June	2024
31	2	1000000002	8:00	10	May	2024
33	6	1000000003	9:00	19	May	2024
34	1	1000000006	8:00	20	May	2024
32	6	1000000002	9:00	19	May	2024
40	7	1000000002	9:00	5	June	2024
38	4	1000000008	8:00	20	May	2024

RESERVATION_ID	CLASS_NO	MBR_ID	TIME	DAY	MONTH	YEAR
1	10	1000000001	8:00	10	May	2024
10	10	1000000009	9:00	1	June	2024
14	4	1000000005	8:00	20	May	2024
16	4	1000000012	15:00	6	June	2024
26	10	1000000021	9:00	1	June	2024
15	2	1000000010	16:00	5	June	2024
35	4	1000000012	8:00	20	May	2024
37	4	1000000009	8:00	20	May	2024
39	4	1000000021	13:00	25	May	2024
41	2	1000000010	16:00	5	June	2024
42	2	1000000002	14:00	6	June	2024
43	2	1000000002	14:00	6	June	2024
31	2	1000000002	8:00	10	May	2024
33	6	1000000003	9:00	19	May	2024
34	1	1000000006	8:00	20	May	2024
32	6	1000000002	9:00	19	May	2024
40	7	1000000002	9:00	5	June	2024
38	4	1000000008	8:00	20	May	2024

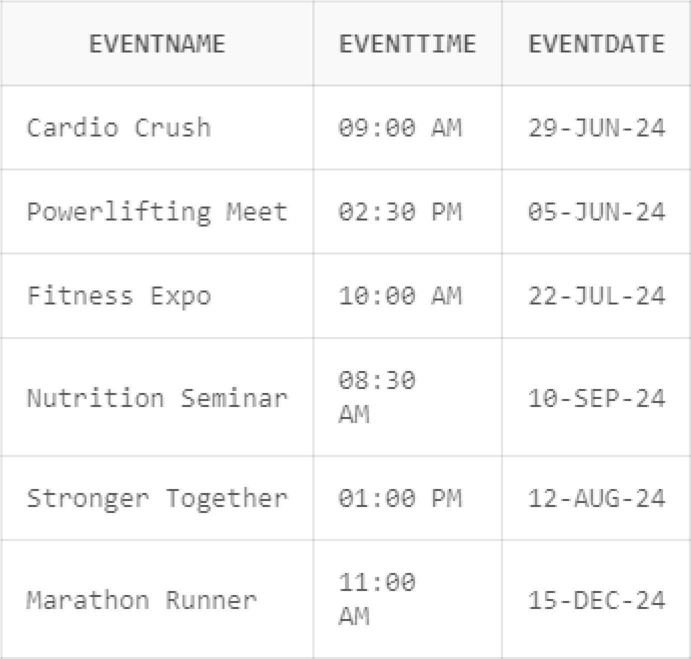
Department



DNAME	DNUMBER	EMP_ID
Human Resources	1	1
Information Technology	4	2
Research and Development	6	6
Coaching 1	7	3
Coaching 2	7	4
Coaching 3	7	5
Coaching 5	7	11
Maintanance	2	8
Coaching 4	7	10
Marketing	3	2
cleaning and sanitizing	5	7
Finance	8	9
Research and Development 2	6	12

DNAME	DNUMBER	EMP_ID
Human Resources	1	1
Information Technology	4	2
Research and Development	6	6
Coaching 1	7	3
Coaching 2	7	4
Coaching 3	7	5
Coaching 5	7	11
Maintenance	2	8
Coaching 4	7	10
Marketing	3	2
cleaning and sanitizing	5	7
Finance	8	9
Research and Development 2	6	12

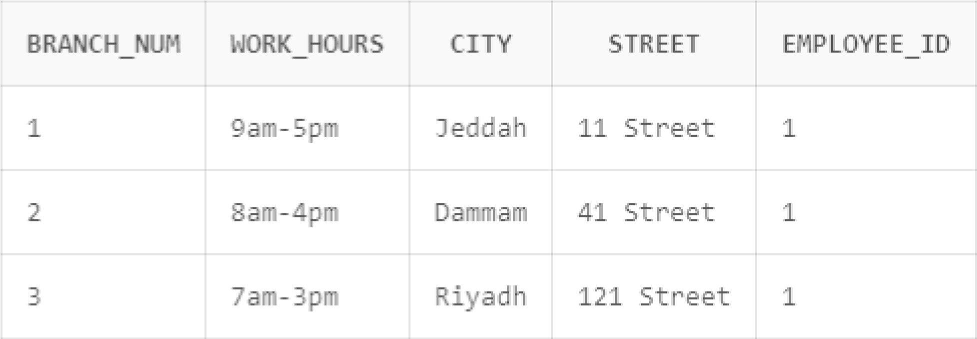
Event



EVENTNAME	EVENTTIME	EVENTDATE
Cardio Crush	09:00 AM	29-JUN-24
Powerlifting Meet	02:30 PM	05-JUN-24
Fitness Expo	10:00 AM	22-JUL-24
Nutrition Seminar	08:30 AM	10-SEP-24
Stronger Together	01:00 PM	12-AUG-24
Marathon Runner	11:00 AM	15-DEC-24

EVENTNAME	EVENTTIME	EVENTDATE
Cardio Crush	09:00 AM	29-JUN-24
Powerlifting Meet	02:30 PM	05-JUN-24
Fitness Expo	10:00 AM	22-JUL-24
Nutrition Seminar	08:30 AM	10-SEP-24
Stronger Together	01:00 PM	12-AUG-24
Marathon Runner	11:00 AM	15-DEC-24

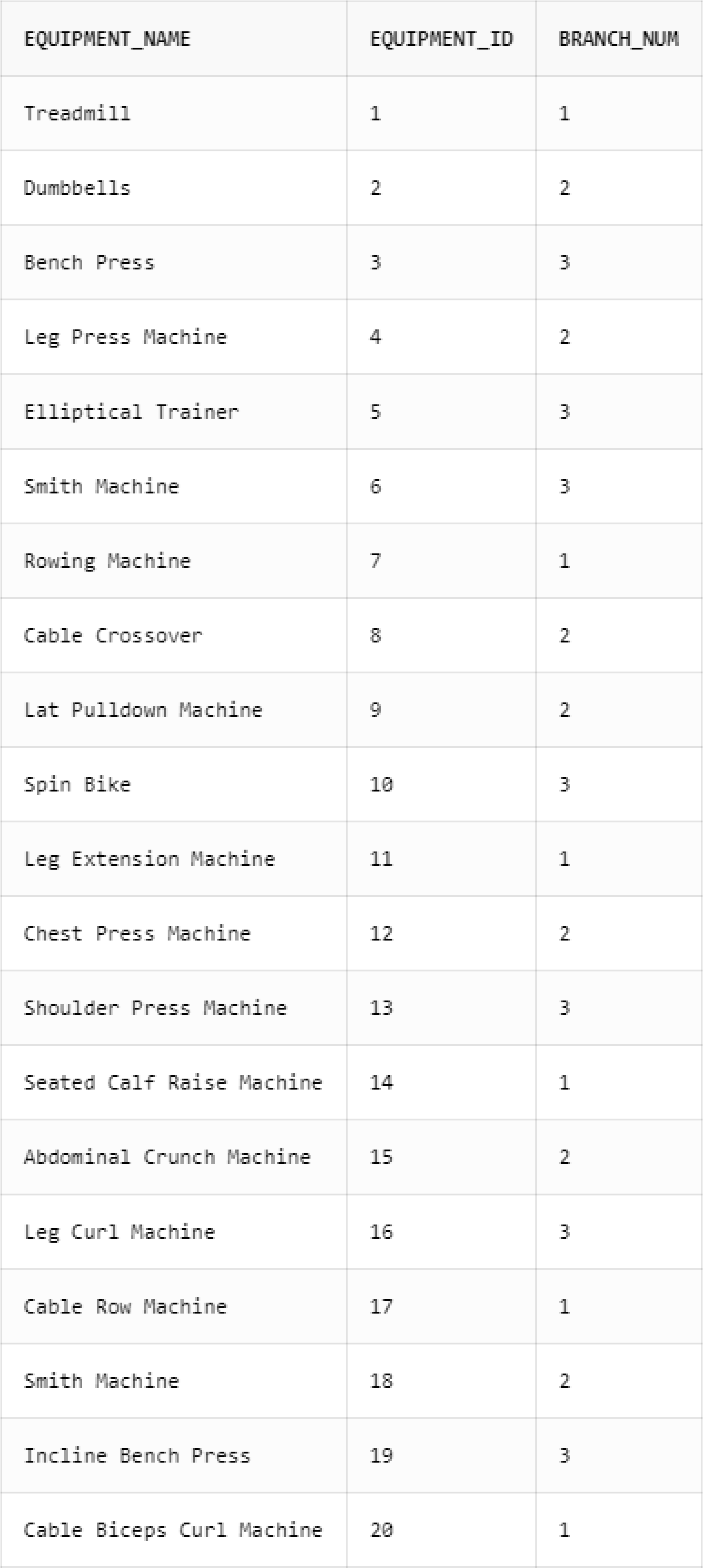
Branches



BRANCH_NUM	WORK_HOURS	CITY	STREET	EMPLOYEE_ID
1	9am-5pm	Jeddah	11 Street	1
2	8am-4pm	Dammam	41 Street	1
3	7am-3pm	Riyadh	121 Street	1

BRANCH_NUM	WORK_HOURS	CITY	STREET	EMPLOYEE_ID
1	9am-5pm	Jeddah	11 Street	1
2	8am-4pm	Dammam	41 Street	1
3	7am-3pm	Riyadh	121 Street	1

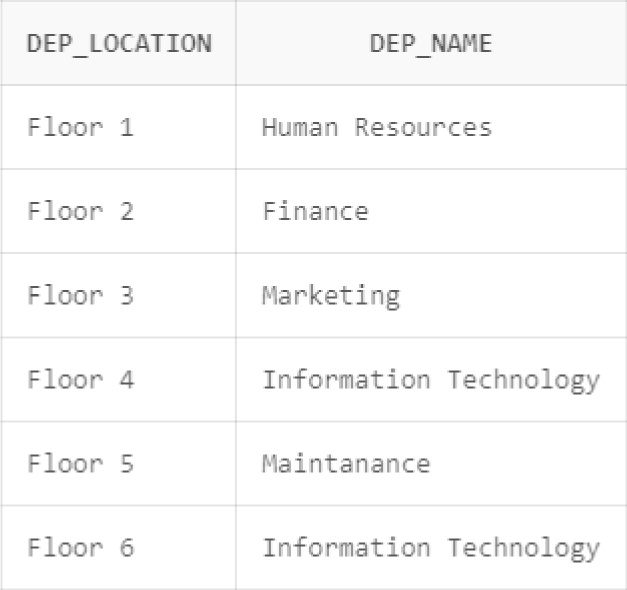
Equipment



EQUIPMENT_NAME	EQUIPMENT_ID	BRANCH_NUM
Treadmill	1	1
Dumbbells	2	2
Bench Press	3	3
Leg Press Machine	4	2
Elliptical Trainer	5	3
Smith Machine	6	3
Rowing Machine	7	1
Cable Crossover	8	2
Lat Pulldown Machine	9	2
Spin Bike	10	3
Leg Extension Machine	11	1
Chest Press Machine	12	2
Shoulder Press Machine	13	3
Seated Calf Raise Machine	14	1
Abdominal Crunch Machine	15	2
Leg Curl Machine	16	3
Cable Row Machine	17	1
Smith Machine	18	2
Incline Bench Press	19	3
Cable Biceps Curl Machine	20	1

EQUIPMENT_NAME	EQUIPMENT_ID	BRANCH_NUM
Treadmill	1	1
Dumbbells	2	2
Bench Press	3	3
Leg Press Machine	4	2
Elliptical Trainer	5	3
Smith Machine	6	3
Rowing Machine	7	1
Cable Crossover	8	2
Lat Pulldown Machine	9	2
Spin Bike	10	3
Leg Extension Machine	11	1
Chest Press Machine	12	2
Shoulder Press Machine	13	3
Seated Calf Raise Machine	14	1
Abdominal Crunch Machine	15	2
Leg Curl Machine	16	3
Cable Row Machine	17	1
Smith Machine	18	2
Incline Bench Press	19	3
Cable Biceps Curl Machine	20	1
Triceps Pushdown Machine	21	2
Hack Squat Machine	22	3
Glute Machine	23	1
Lat Pulldown Machine	24	2
Leg Press Machine	25	3

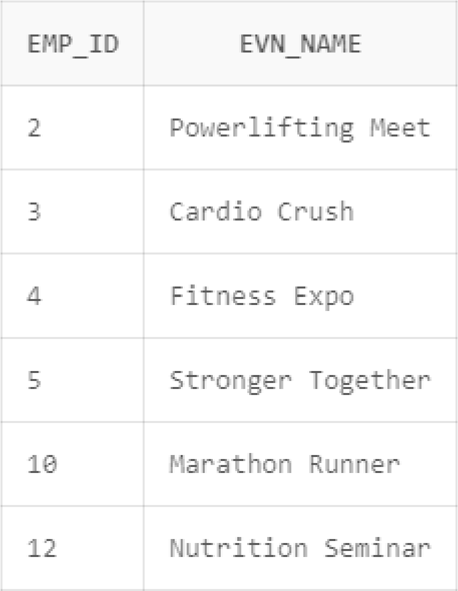
Dep_Location



DEP_LOCATION	DEP_NAME
Floor 1	Human Resources
Floor 2	Finance
Floor 3	Marketing
Floor 4	Information Technology
Floor 5	Maintanance
Floor 6	Information Technology

DEP_LOCATION	DEP_NAME
Floor 1	Human Resources
Floor 2	Finance
Floor 3	Marketing
Floor 4	Information Technology
Floor 5	Maintenance
Floor 6	Information Technology

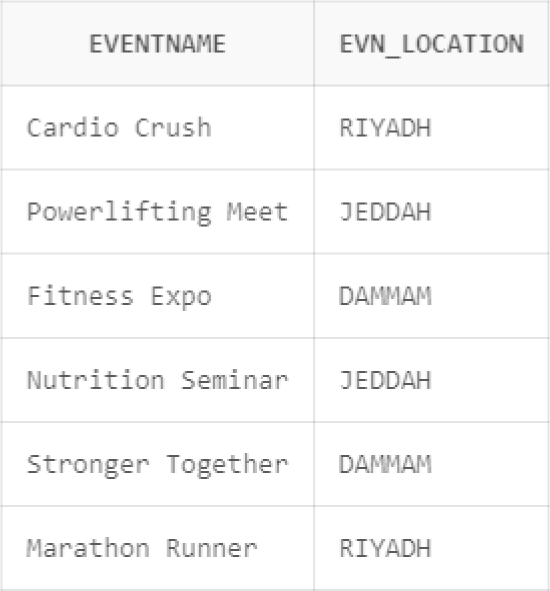
Supervised_by



EMP_ID	EVN_NAME
2	Powerlifting Meet
3	Cardio Crush
4	Fitness Expo
5	Stronger Together
10	Marathon Runner
12	Nutrition Seminar

EMP_ID	EVN_NAME
2	Powerlifting Meet
3	Cardio Crush
4	Fitness Expo
5	Stronger Together
10	Marathon Runner
12	Nutrition Seminar

Evn_Location



EVENTNAME	EVN_LOCATION
Cardio Crush	RIYADH
Powerlifting Meet	JEDDAH
Fitness Expo	DAMMAM
Nutrition Seminar	JEDDAH
Stronger Together	DAMMAM
Marathon Runner	RIYADH

EVENTNAME	EVN_LOCATION
Powerlifting Meet	JEDDAH
Fitness Expo	DAMMAM
Nutrition Seminar	JEDDAH
Stronger Together	DAMMAM
Marathon Runner	RIYADH
Cardio Crush	RIYADH