최종 보고서

(데이터베이스 설계 2023 년 1 학기)

Give Your Money

제출일: 2023 년 6 월 17 일

이름: 윤성배, 최현영

학번: 2019102203, 2019102236

◆ 설계 배경

■ 프로젝트 필요성

헬스장은 사람들의 건강을 유지하기 위해 방문한다. 신체적인 활동을 통해 체력을 증진시키고, 근력을 강화하며 유연성과 몸의 균형을 향상시킬 수 있다. 또한, 일상 생활에서의 스트레스를 해소하며 긍정적인 에너지를 얻을 수 있기에 남녀노소를 불문하고 많이들 방문한다. 뿐만 아니라, 바디 프로필 유행에 따른 몸의 가시적인 변화를 몸소 느끼고 싶어하는 욕구, 그리고 다양한 사람들과 소통하고 교류하며 친분을 쌓는 장소로도 사용되기에 더욱이 사람들의 헬스장 등록이 많아지고 있다. 증가하는 회원을 관리하고 헬스장 운영에 필요한 작업을 서류가 아닌, 어플리케이션과 DBMS 를 통한 자동화 덕분에 업무 프로세스 개선과 효율성 향상에 많은 영향을 미칠 것으로 사료된다.

■ 문제 정의

헬스 트레이너를 운영하는 입장에서 갈수록 많아지는 회원들의 정보를 보다 효율적으로 관리하고 이 정보에 기반한 다양한 서비스를 제공하며, 회원을 충족하기 위한 운동기구 및 편의시설과 같은 사내 물품을 효율적으로 운영해야 한다.

■ 프로젝트 완성 후의 기대 효과

헬스장의 데이터를 DBMS 가 관리함으로써 효율적으로 관리할 수 있다. 회원정보 및 운동기록, 신체검사 이력, 결제 정보, 회계정보 등을 체계적으로 저장하고 관리할 수 있기에 데이터 일관성을 유지할 수 있다. DBMS 를 통한 업무 효율이 증대될 수 있다. 대면/비대면 회원 등록 및 전산 결제, 스케쥴 및

당번 관리 등 헬스장의 전반적인 업무를 효율적으로 처리할 수 있어 시간과 비용의 절약을 기대할 수 있다. 이 덕분에 사업의 등록자 및 직원들은 데이터 관리에 부담을 가질 부담 없이 회원에게 질 좋은 서비스를 제공하는데 초점을 둘 수 있다. RDBMS 의 설계 덕분에 엔티티 및 관계에 기반한, 필요에 따른 서비스 확장이 용이하다.

이뿐만이 아니라, 회원정보를 기반으로 맞춤형 서비스를 제공할 수 있으며, 부수적인 기타 정보들을 관리하여 연령대 혹은 직업별로 특화된 서비스를 제공함으로써 수익 창출 효과를 기대할 수 있다. 개인 트레이닝 서비스 역시, 회원의 운동기록 및 성과, 목표량에 따른 데이터 추적이 가능하기에 트레이너가 이 정보에 기반한 맞춤형 서비스를 제공할 수 있다.

■ 기타, 설계 계획서의 내용 요약

회원의 정보, 직원의 정보 및 헬스장 당번, 운동기구 및 물품, 매출 및 지출과 관련된 장부(거래)을 통합적으로 관리하기 위한 스키마를 구현하고자 한다.

등록할 회원의 정보, 직원의 정보, 물품 및 운동기구, 매출과 지출과 같은 거래내역, PT 등록 및 할인율과 같은 서비스라는 엔티티를 두고, 이를 바탕으로 필수 및 부수적인 애트리뷰트들이 무엇이 있을 지 정의한 뒤, 각 엔티티 간의 관계를 정의해 준다.

◆ 설계 목표

■ 요구 사항 등을 분석 정리한 내용 요약

1. 헬스장을 사용하려는 자는 등록을 할 수 있다.
2. 회원은 주민번호, 이메일, 종목, 직업, 나이, 주소, 이름을 필수적으로 기입해야 한다.
3. 헬스장의 직원은 이름, 주민번호, 월급, 직책, 전화번호를 필수적으로 기입해야 한다.
4. 장소는 섹터별로 구별하여 당번으로 지정된 직원이 그 섹터를 담당하는 책임자가 된다.
5. 가격표는 서비스 이름마다 가격 및 할인이 적용된다.
6. PT 는 서비스 개념으로 횟수 및 담당 트레이너가 지정된다.
7. 운동기구마다 상태 및 고유번호, 종류 및 비치된 장소가 지정된다.
8. 물품마다 고유 번호 및 수량, 품목이 존재하며, 장소가 지정된다.
9. 매출 및 지출과 관련된 거래내역은 투명해야 하며 상세내용이 존재한다.
10. 회원의 신체 측정을 통해 만들어진 데이터가 회원에게만 투명하게 공개되어 살펴볼 수 있다.
11. 직원은 PT 시, 회원의 성취도 평가를 통해 진척률을 점검하고 회원에게 피드백을 제공할 수 있다.

■ 설계 과정에서 주요하게 고려하여야 할 사항 들 정리

- 데이터의 구조화 및 정규화: 데이터의 구조화와 정규화를 고려하여 데이터 항목을 적절한 엔티티와 관계로 구성하고, 중복을 최소화하여 일관성을 유지해야 한다.
- 데이터 보안과 개인 정보 보호: 회원 정보와 거래 데이터 등의 보안과 같은 민감한 정보들은 접근 제어, 사용자 인증 및 권한 관리 등을 구현하여 데이터의 기밀성을 유지해야 한다.
- 확장성: 실시간 서비스를 구현하는 것은 아니기에 성능이 매우 중요하지 않지만, 사업에서 가장 중요한 것은 다양한 서비스에 따른 수익 창출이기에 확장 가능한 아키텍처로 설계해야 한다.
- 백업과 복구: 금전거래가 이루어지는 사업이기에 전자 회계 장부 기록은 매우 중요하며, 회원의 소중한 데이터 손실을 방지하기 위해 백업과 복구를 매우 중요시해야 하며 구체적으로 대안을 정립해야 한다.
- 손쉬운 접근성: 사내 직원들이 데이터를 손 쉽게 접근 및 관리하여 작업 효율성을 향상시킬 수 있다.
- 사용자 요구사항: 사용자들의 요구에 부합하고 필요한 기능을 제공하는 데이터베이스를 설계하여 구축하여야 한다.

■ 사용할 시스템 환경에 대한 소개

1. 서버가 필요하다. 서버는 클라우드 플랫폼을 사용하여 실제 물리적인 서버를 두지 않고, 공급자자가 제공해주는 인프라를 사용하여 청구된 비용만큼 돈을 지불하여 서비스를 운영할 수 있다.
2. 또한, 올바른 회원의 출입을 위한 인증 시스템이 필요하다. 회원만의 고유한 생체인증 또는 회원번호 등을 통해 헬스장을 등록한 회원만이 사용할 수 있다.

3. 대면을 통한 등록이 아닌 애플리케이션을 통한 비대면 등록도 가능해야 한다. 안드로이드 및 iOS 용 애플리케이션을 구축하여 보다 많은 사람들에게 홍보를 할 수 있을 것이며, 회원의 이용권 및 신체 변화에 대한 정보를 한 눈에 확인할 수 있을 것이다.

4. 직원 전용(Admin) 홈페이지를 통해 회원 및 직원, 통합 자산을 효율적으로 관리할 수 있다.

5. 인프라는 클라우드 데이터베이스 환경을 통해 물리적인 데이터베이스 장치를 직접 설치하지 않고, 제공자로부터 지불한 만큼 서비스를 사용하여 인프라 구축에 대한 비용 및 시간을 절감한다.

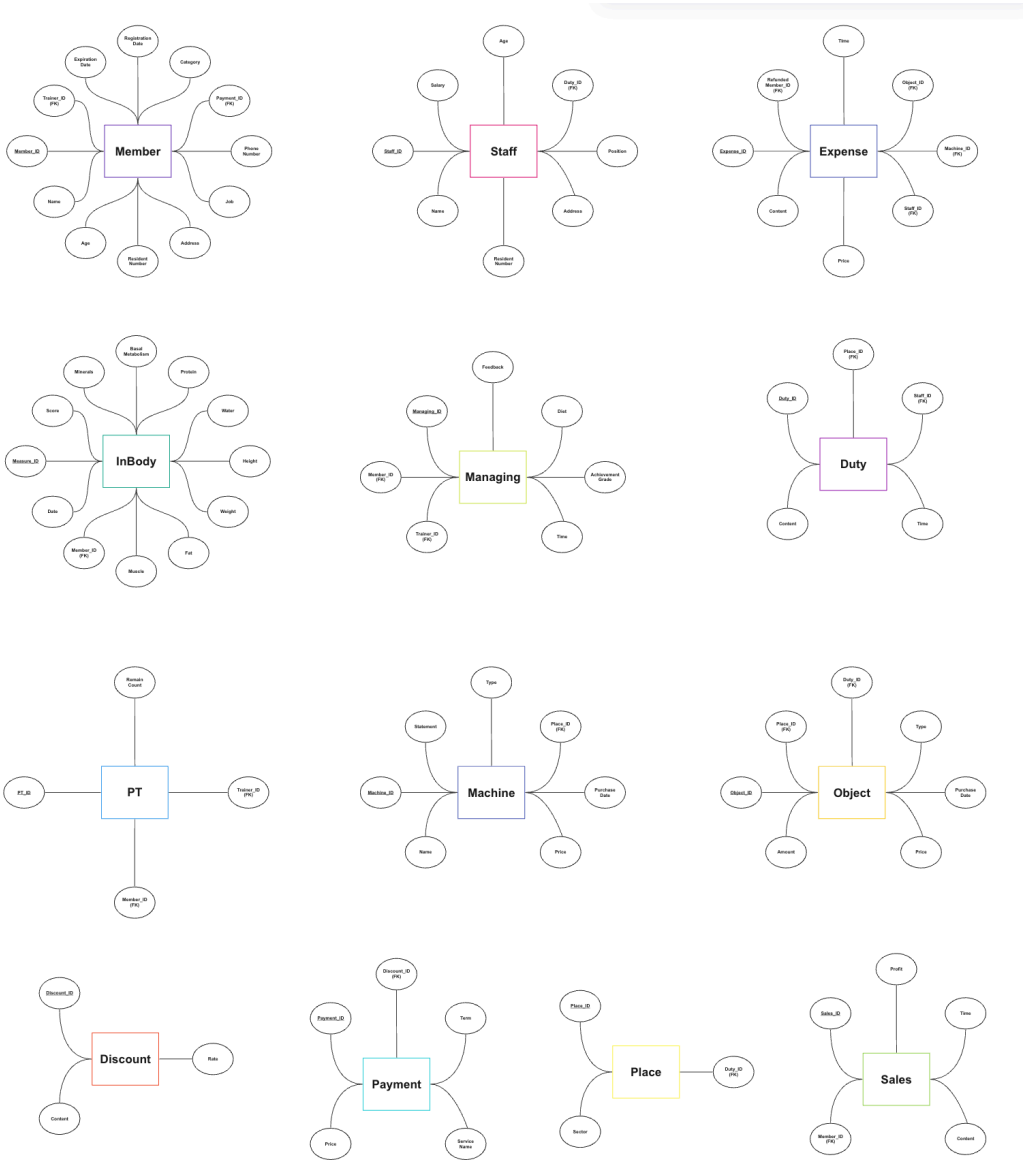
6. 오픈소스 RDBMS 인 MySQL 을 사용할 것인데, 이는 사업을 운용하는 입장에서 성능과 신뢰성을 보장받고, 널리 사용되고 있는 대중성을 고려해야 한다. MySQL 은 이를 모두 충족하며 다중 사용자를 지원하며, 다양한 프로그래밍 언어를 위한 API 를 지원하고 있어 서버 구축 시 언어에 대한 호환성을 체크 여부를 하지 않아도 되는 장점이 있다.

■ 설계 구현된 시스템을 평가할 수 있는 항목의 제시

- 정규화를 통한 데이터 중복의 최소화
- 데이터 접근의 편의성
- 적절한 제약 조건을 통한 데이터 무결성 유지
- 데이터 접근의 편의성
- ERD 에서 릴레이션으로 사상 시, 7 단계 알고리즘의 활용성
- 트랜잭션을 통한 자동화
- 유의미한 데이터인지 확인

◆ 설계 내용

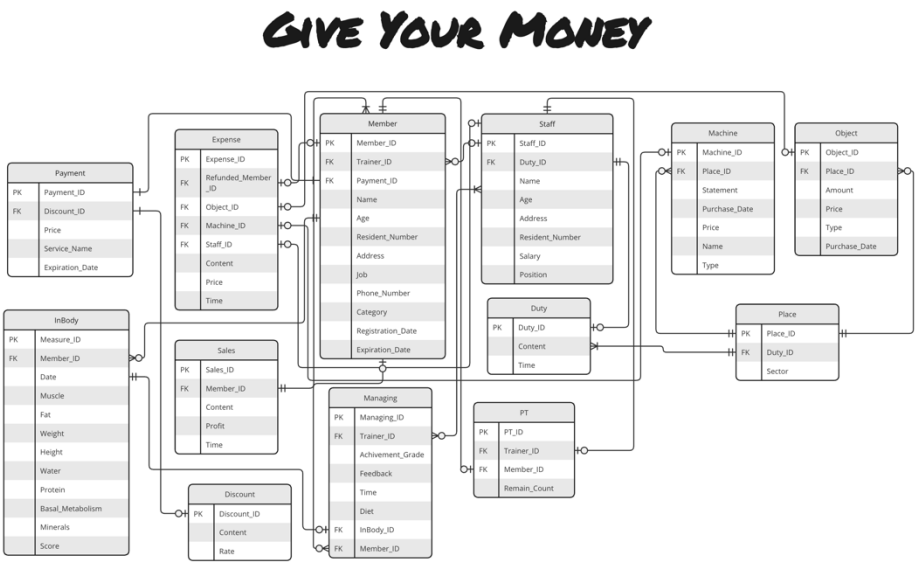
■ 앞에서 기술한 내용을 반영하는 개념적인 데이터베이스 설계



요구사항을 분석하여 마인드맵으로 우선 도식화하였다.

총 13 가지의 모델들을 구축할 수 있었다.

■ (ER Schema) ER 다이어그램을 통한 DB 구조 도시



■ ER 다이어그램을 중심으로 하여 데이터베이스 모습을 정리 요약

i. 회원이 PT를 등록할 시 PT 테이블 사용

ii. 회원이 신체 측정할 시 인바디 테이블 사용

iii. 트레이너가 회원 PT 를 통해 계획을 짤 때 PT 테이블, Managing 테이블 사용

iv. 직원들이 헬스장 내 당번을 할 때 Duty 테이블 사용

v. 맡은 당번이 어느 장소에서 이뤄지는 지 Place 테이블 사용

vi. 운동 기구나 물품들이 어디에 위치해 있는지 Place 테이블 사용

vii. 매출 확인 Sales 테이블 사용

viii. 지출 확인 Expense 테이블 사용

ix. 회원이 헬스장에 등록할 때 Payment 테이블 사용

x. Machine 테이블과 Object 테이블로 기구, 물품 상태 관리

xi. 결제할 때 Discount 테이블 사용

■ ER 다이어그램을 기반으로 생성한 릴레이션 스키마

- Payment(Payment_ID : Integer, Discount_ID : Integer, Price : Integer, Service_Name : Varchar(500), Expiration_Date : Timestamp)
- Expense(Expense_ID : Integer, Refunded_Member_ID : Integer, Object_ID : Integer, Machine_Id : Integer, Staff_ID : Integer, Content : Varchar(500), Price : Integer, Time : Timestamp)
- InBody(Measure_ID : Integer, Member_ID : Integer, Date : Timestamp, Muscle : Float, Fat : Float, Weight : Float, Height : Float, Water : Float, Protein : Float, Basal_Metabolism : Float, Minerals : Float, Score : Integer)
- Sales(Sales_ID : Integer, Member_ID : Integer, Content : Varchar(50), Profit : Integer, Time : Timestamp)
- Discount(Discount_ID : Integer, Content : Varchar(500), Rate : Float)
- Member(Member_ID : Integer, Trainer_ID : Integer, Payment_ID : Integer, Name : Varchar(5), Age: Integer, Resident_Number: Varchar(15), Address : Varchar(50), Job: Varchar(20), Phone_Number : Varchar(12), Category : char(10), Registration_Date : TIMESTAMP, Expiration_Date : Timestamp)
- Managing(Managing_ID : Integer, Trainer_ID : Integer, Achivement_Grade : char(10), Feedback : Varchar(500), Time : Timestamp, Diet : Varchar(200), InBody_ID : Integer, Member_ID : Integer)

- Staff(Staff_ID : Integer, Duty_ID : Integer ,Name : Varchar(5) ,Age : Integer ,Address : Varchar(50) ,Resident_Number : Varchar(15) ,Salary : Integer ,Position : char(10))
- Duty(Duty_ID : Integer, Content : Varchar(500), Time : Timestamp)
- PT(PT_ID : Integer, Trainer_ID : Integer, Member_ID : Integer, Remain_Count : Integer)
- Machine(Machine_ID : Integer, Place_ID : Integer, Statement : Integer, Purchase_Date : Timestamp, Price : Integer, Name : char(20), Type : char(20))
- Object(Object_ID : Integer, Place_ID : Integer, Amount : Integer , Price : Integer, Type : char(20), Purchase_Date : TImestamp)
- Place(Place_ID : Integer, Duty_ID : Integer, Sector : Integer)

■ 논리적 DB 구조 (테이블 구조와 테이블 사이의 관련성 도시)

- Payment

```
CREATE TABLE payment (
  payment_id INT PRIMARY KEY AUTO_INCREMENT,
  price INT NOT NULL,
  duration INT NOT NULL
);

-- 제약조건
ALTER TABLE payment
ADD CONSTRAINT CHECK (price > 0 AND duration > 0);
```

```
| payment | CREATE TABLE `payment` (
  `payment_id` int NOT NULL AUTO_INCREMENT,
  `Price` int NOT NULL DEFAULT '0',
  `Duration` int NOT NULL DEFAULT '0',
  PRIMARY KEY (`payment_id`),
  CONSTRAINT `payment_chk_1` CHECK ((((`price` > 0) and (`duration` > 0)))
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

- Expense

```sql
CREATE TABLE Expense (
  Expense_ID INT PRIMARY KEY AUTO_INCREMENT,
  Object_ID INT,
  Machine_Id INT,
  Staff_ID INT,
  Content VARCHAR(500) NOT NULL,
  Price INT NOT NULL,
  Time TIMESTAMP NOT NULL,
  CONSTRAINT FK_Object_ID FOREIGN KEY (Object_ID) REFERENCES Object(Object_ID),
  CONSTRAINT FK_Machine_Id FOREIGN KEY (Machine_Id) REFERENCES Machine(Machine_Id),
  CONSTRAINT FK_Staff_ID FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
);

-- 제약조건
ALTER TABLE expense
ADD CONSTRAINT CHECK (price > 0);
```

```
---------------------+
| expense | CREATE TABLE `expense` (
  `Expense_ID` int NOT NULL AUTO_INCREMENT,
  `Object_ID` int DEFAULT NULL,
  `Machine_Id` int DEFAULT NULL,
  `Staff_ID` int DEFAULT NULL,
  `Content` varchar(500) NOT NULL,
  `price` int DEFAULT NULL,
  `Time` timestamp NOT NULL,
  PRIMARY KEY (`Expense_ID`),
  KEY `FK_Object_ID` (`Object_ID`),
  KEY `FK_Machine_Id` (`Machine_Id`),
  KEY `FK_Staff_ID` (`Staff_ID`),
  CONSTRAINT `FK_Machine_Id` FOREIGN KEY (`Machine_Id`) REFERENCES `Machine` (`machine_id`),
  CONSTRAINT `FK_Object_ID` FOREIGN KEY (`Object_ID`) REFERENCES `Object` (`object_id`),
  CONSTRAINT `FK_Staff_ID` FOREIGN KEY (`Staff_ID`) REFERENCES `Staff` (`Staff_ID`),
  CONSTRAINT `expense_chk_1` CHECK ((`price` > 0))
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+---------+---------------------------------------------------------------------------------
```

- Inbody

```sql
CREATE TABLE inbody (
  Inbody_id INT PRIMARY KEY AUTO_INCREMENT,
  member_id INT NOT NULL,
  date TIMESTAMP NOT NULL,
  muscle FLOAT(3,1) NOT NULL,
  fat FLOAT(3,1) NOT NULL,
  weight FLOAT(4,1) NOT NULL,
  height FLOAT(4,1) NOT NULL,
  water FLOAT(3,1) NOT NULL,
  protein FLOAT(3,1) NOT NULL,
  basal_metabolism INT NOT NULL,
  minerals FLOAT(3,2) NOT NULL,
  FOREIGN KEY (member_id) REFERENCES member(member_id)
);

-- 제약조건
ALTER TABLE inbody
ADD CONSTRAINT inbody_cascade FOREIGN KEY (member_id)
    REFERENCES member(member_id)
    ON DELETE CASCADE;
```

```
| inbody | CREATE TABLE `inbody` (
  `inbody_id` int NOT NULL AUTO_INCREMENT,
  `member_id` int NOT NULL,
  `date` timestamp NOT NULL,
  `muscle` float(3,1) NOT NULL,
  `fat` float(3,1) NOT NULL,
  `weight` float(4,1) NOT NULL,
  `height` float(4,1) NOT NULL,
  `water` float(3,1) NOT NULL,
  `protein` float(3,1) NOT NULL,
  `basal_metabolism` int NOT NULL,
  `minerals` float(3,2) NOT NULL,
  PRIMARY KEY (`inbody_id`),
  KEY `idx_private_inbody` (`member_id`,`date` DESC),
  CONSTRAINT `inbody_cascade` FOREIGN KEY (`member_id`) REFERENCES `member` (`Member_ID`) ON DELETE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

- Scale

```sql
CREATE TABLE Sales (
  Sales_ID INT PRIMARY KEY,
  Member_ID INT,
  Discount_ID INT,
  Content VARCHAR(50) NOT NULL,
  Profit INT not null,
  Time TIMESTAMP NOT NULL,
  CONSTRAINT FK_Member_ID FOREIGN KEY (Member_ID)
    REFERENCES Member(Member_ID),
  CONSTRAINT FK_Discount_ID FOREIGN KEY (Discount_ID)
    REFERENCES Discount(Discount_ID)
);
```

```
| sales | CREATE TABLE `sales` (
  `Sales_ID` int NOT NULL AUTO_INCREMENT,
  `Member_ID` int DEFAULT NULL,
  `Content` varchar(50) NOT NULL,
  `Profit` int NOT NULL,
  `Time` timestamp NOT NULL,
  PRIMARY KEY (`Sales_ID`),
  KEY `Member_ID` (`Member_ID`),
  CONSTRAINT `sales_ibfk_1` FOREIGN KEY (`Member_ID`) REFERENCES `member` (`Member_ID`) ON DELETE SET NULL
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

- Discount

```sql
CREATE TABLE discount (
  discount_id INT PRIMARY KEY AUTO_INCREMENT,
  rate FLOAT(3,2) NOT NULL
);

-- 제약조건
ALTER TABLE discount
ADD CONSTRAINT CHECK (rate > 0.0);
```

```
mysql> show create table discount;
+----------+------------------------------------------------------------
--------------------------------------------------------------------------+
| Table    | Create Table
                                                                          |
+----------+------------------------------------------------------------
--------------------------------------------------------------------------+
| discount | CREATE TABLE `discount` (
  `Discount_Id` int NOT NULL AUTO_INCREMENT,
  `rate` float(3,2) NOT NULL DEFAULT '0.00',
  PRIMARY KEY (`Discount_Id`),
  CONSTRAINT `discount_chk_1` CHECK ((`rate` > 0.0))
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
+----------+------------------------------------------------------------
--------------------------------------------------------------------------+
1 row in set (0.00 sec)
```

➢ Managing

```
CREATE TABLE managing (
  managing_id INT PRIMARY KEY AUTO_INCREMENT,
  member_id INT NOT NULL,
  trainer_id INT NOT NULL,
  grade VARCHAR(100) NOT NULL,
  feedback VARCHAR(100) NOT NULL,
  date TIMESTAMP NOT NULL,
  FOREIGN KEY (member_id) REFERENCES member(member_id),
  FOREIGN KEY (trainer_id) REFERENCES staff(staff_id),
  FOREIGN KEY (managing_id) REFERENCES inbody(inbody_id) ON DELETE CASCADE;
);
```

```
machine | CREATE TABLE `machine` (
  `machine_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL,
  `price` int NOT NULL,
  `purchase_date` timestamp NOT NULL,
  `type` varchar(100) NOT NULL,
  `place` varchar(100) NOT NULL,
  `state` varchar(100) NOT NULL,
  PRIMARY KEY (`machine_id`),
  KEY `idx_machine` (`name`),
  CONSTRAINT `machine_chk_1` CHECK ((`price` > 0)),
  CONSTRAINT `machine_chk_2` CHECK ((`place` in (_utf8mb4'pilates',_utf8mb4'machine',_utf8mb4'free weight',_utf8mb4'dressing room',_utf8mb4'stretching'))),
  CONSTRAINT `machine_chk_3` CHECK ((`type` in (_utf8mb4'leg',_utf8mb4'arm',_utf8mb4'shoulder',_utf8mb4'chest',_utf8mb4'aerobic',_utf8mb4'back'))),
  CONSTRAINT `machine_chk_4` CHECK ((`state` in (_latin1'good',_latin1'soso',_latin1'bad')))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

➢ Member

```
CREATE TABLE Member (
  Member_ID INT AUTO_INCREMENT PRIMARY KEY,
  Trainer_ID INT,
  Payment_ID INT NOT NULL,
  Name VARCHAR(5) NOT NULL,
  Age INT NOT NULL,
  Sex VARCHAR(2) NOT NULL,
  Address VARCHAR(50) NOT NULL,
  Job VARCHAR(20) NOT NULL,
  Phone_Number VARCHAR(12) NOT NULL UNIQUE,
  Category CHAR(10),
  Registration_Date TIMESTAMP NOT NULL,
  Expiration_Date TIMESTAMP NOT NULL,
  FOREIGN KEY (Trainer_ID) REFERENCES Staff(Staff_ID),
  FOREIGN KEY (Payment_ID) REFERENCES Payment(Payment_ID)
);

-- 제약조건
ALTER TABLE member
ADD CONSTRAINT
FOREIGN KEY (trainer_id)
REFERENCES staff(staff_id)
ON DELETE SET NULL;

ALTER TABLE member
ADD CONSTRAINT CHECK (category in ('PT', 'Pilates', 'Free')),
ADD CONSTRAINT CHECK (sex in ('M', 'F'));
```

```
| member | CREATE TABLE `member` (
  `Member_ID` int NOT NULL AUTO_INCREMENT,
  `Trainer_Id` int DEFAULT NULL,
  `payment_id` int DEFAULT NULL,
  `Name` varchar(5) NOT NULL,
  `Age` int NOT NULL,
  `Address` varchar(50) NOT NULL,
  `Job` varchar(20) NOT NULL,
  `Phone_Number` varchar(12) NOT NULL,
  `Category` char(10) DEFAULT NULL,
  `registration_date` timestamp NULL DEFAULT NULL,
  `expiration_date` timestamp NULL DEFAULT NULL,
  `Sex` char(2) NOT NULL,
  `discount_id` int DEFAULT NULL,
  PRIMARY KEY (`Member_ID`),
  UNIQUE KEY `Phone_Number` (`Phone_Number`),
  KEY `Payment_ID` (`payment_id`),
  KEY `idx_member_name` (`Name`),
  KEY `Trainer_Id` (`Trainer_Id`),
  KEY `discount_id` (`discount_id`),
  CONSTRAINT `member_ibfk_2` FOREIGN KEY (`payment_id`) REFERENCES `Payment` (`payment_id`),
  CONSTRAINT `member_ibfk_3` FOREIGN KEY (`Trainer_Id`) REFERENCES `staff` (`Staff_ID`) ON DELETE SET NULL,
  CONSTRAINT `member_ibfk_4` FOREIGN KEY (`discount_id`) REFERENCES `discount` (`Discount_Id`),
  CONSTRAINT `member_chk_1` CHECK ((`Category` in (_utf8mb4'PT',_utf8mb4'Pilates',_utf8mb4'Free'))),
  CONSTRAINT `member_chk_2` CHECK ((`sex` in (_latin1'M',_latin1'F')))
) ENGINE=InnoDB AUTO_INCREMENT=39 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

➢ Staff

```
CREATE TABLE Staff (
  Staff_ID INT AUTO_INCREMENT PRIMARY KEY,
  Duty_ID INT,
  Name VARCHAR(5) NOT NULL,
  Age INT NOT NULL,
  Address VARCHAR(100) NOT NULL,
  Sex CHAR(2) NOT NULL,
  Salary INT NOT NULL,
  Position CHAR(10) NOT NULL,
  FOREIGN KEY (Duty_ID) REFERENCES Duty(Duty_ID)
);

-- 제약조건
ALTER TABLE staff
ADD CONSTRAINT CHECK (sex in ('M', 'F')),
ADD CONSTRAINT CHECK (position in ('Trainer', 'Boss', 'Manager'));
```

```
| staff | CREATE TABLE `staff` (
  `Staff_ID` int NOT NULL AUTO_INCREMENT,
  `Duty_ID` int DEFAULT NULL,
  `Name` varchar(5) NOT NULL,
  `Age` int NOT NULL,
  `Address` varchar(100) NOT NULL,
  `Sex` char(2) NOT NULL,
  `Salary` int NOT NULL,
  `Position` char(10) NOT NULL,
  PRIMARY KEY (`Staff_ID`),
  KEY `Duty_ID` (`Duty_ID`),
  CONSTRAINT `staff_ibfk_1` FOREIGN KEY (`Duty_ID`) REFERENCES `Duty` (`Duty_id`),
  CONSTRAINT `staff_chk_1` CHECK ((`Sex` in (_utf8mb4'M',_utf8mb4'F'))),
  CONSTRAINT `staff_chk_2` CHECK ((`position` in (_latin1'Trainer',_latin1'Boss',_latin1'Manager')))
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

➢ Duty

```
CREATE TABLE duty (
  duty_id INT PRIMARY KEY AUTO_INCREMENT,
  place VARCHAR(100) NOT NULL,
  time INT NOT NULL
);

-- 제약조건
ALTER TABLE duty
ADD CONSTRAINT CHECK (place in
  ('free weight', 'pilates', 'stretching', 'machine', 'dressing room'));
```

```
| duty  | CREATE TABLE `duty` (
  `Duty_id` int NOT NULL AUTO_INCREMENT,
  `place` varchar(100) NOT NULL,
  `time` int NOT NULL,
  PRIMARY KEY (`Duty_id`),
  CONSTRAINT `duty_chk_1` CHECK ((`place` in (_utf8mb4'free weight',_utf8mb4'pilates',_utf8mb4'stretching',_utf8mb4'machine',_utf8mb4'dressing room'))),
  CONSTRAINT `duty_chk_2` CHECK (((`time` >= 0) and (`time` < 24)))
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

➢ PT

```sql
CREATE TABLE pt (
  pt_id INT PRIMARY KEY AUTO_INCREMENT,
  member_id INT NOT NULL,
  remain_count INT NOT NULL,
  FOREIGN KEY (member_id) REFERENCES member(member_id) ON DELETE CASCADE
);

-- 제약조건
ALTER TABLE pt
ADD CONSTRAINT CHECK (remain_count >= 0),
```

```
| pt    | CREATE TABLE `pt` (
  `pt_id` int NOT NULL AUTO_INCREMENT,
  `member_id` int NOT NULL,
  `remain_count` int NOT NULL,
  PRIMARY KEY (`pt_id`),
  KEY `member_id` (`member_id`),
  CONSTRAINT `pt_ibfk_2` FOREIGN KEY (`member_id`) REFERENCES `member` (`Member_ID`) ON DELETE CASCADE,
  CONSTRAINT `pt_chk_1` CHECK ((`remain_count` >= 0))
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

➢ Machine

```sql
CREATE TABLE machine (
  machine_id INT PRIMARY KEY AUTO_INCREMENT,
  name VARCHAR(100) NOT NULL,
  price INT NOT NULL,
  purchase_date TIMESTAMP NOT NULL,
  state VARCHAR(100) NOT NULL,
  type VARCHAR(100) NOT NULL,
  place VARCHAR(100) NOT NULL
);

-- 제약조건
ALTER TABLE machine
ADD CONSTRAINT CHECK (price > 0),
ADD CONSTRAINT CHECK (place in
  ('free weight', 'pilates', 'stretching', 'machine', 'dressing room')),
ADD CONSTRAINT CHECK (type in
  ('aerobic', 'chest', 'back', 'arm', 'leg', 'shoulder')),
ADD CONSTRAINT CHECK (state in ('good', 'soso', 'bad'));
```

```
| machine | CREATE TABLE `machine` (
  `machine_id` int NOT NULL AUTO_INCREMENT,
  `name` varchar(100) NOT NULL,
  `price` int NOT NULL,
  `purchase_date` timestamp NOT NULL,
  `type` varchar(100) NOT NULL,
  `place` varchar(100) NOT NULL,
  `state` varchar(100) NOT NULL,
  PRIMARY KEY (`machine_id`),
  KEY `idx_machine` (`name`),
  CONSTRAINT `machine_chk_1` CHECK ((`price` > 0)),
  CONSTRAINT `machine_chk_2` CHECK ((`place` in (_utf8mb4'pilates',_utf8mb4'machine',_utf8mb4'free weight',_utf8mb4'dressing room',_utf8mb4'stretching'))),
  CONSTRAINT `machine_chk_3` CHECK ((`type` in (_utf8mb4'leg',_utf8mb4'arm',_utf8mb4'shoulder',_utf8mb4'chest',_utf8mb4'aerobic',_utf8mb4'back'))),
  CONSTRAINT `machine_chk_4` CHECK ((`state` in (_latin1'good',_latin1'soso',_latin1'bad')))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

➢ Object

```sql
CREATE TABLE object (
  object_id INT PRIMARY KEY AUTO_INCREMENT,
  place VARCHAR(100) NOT NULL,
  amount INT NOT NULL,
  price INT NOT NULL,
  purchase_date TIMESTAMP not null
)

-- 제약조건
ALTER TABLE object
ADD CONSTRAINT CHECK (amount >= 0 and price > 0),
ADD CONSTRAINT CHECK (place in
('free weight', 'pilates', 'stretching', 'machine', 'dressing room'));
```

```
| object | CREATE TABLE `object` (
  `object_id` int NOT NULL AUTO_INCREMENT,
  `place` varchar(100) NOT NULL,
  `amount` int NOT NULL,
  `price` int NOT NULL,
  `purchase_date` timestamp NOT NULL,
  PRIMARY KEY (`object_id`),
  CONSTRAINT `object_chk_1` CHECK (((`amount` >= 0) and (`price` > 0))),
  CONSTRAINT `object_chk_2` CHECK ((`place` in (_latin1'free weight',_latin1'pilates',_latin1'stretching',_latin1'machine',_latin1'dressing room')))
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci |
```

■ 릴레이션 사상 과정

가) 마인드맵을 사용해서 엔티티와 애트리뷰트들을 구성해봤지만 ER 다이어그램을 그리면서 부족한 부분을 파악하고 새로운 외래 키 애트리뷰트를 만들었다.

나) 엔티티들의 관계를 따져보니 순환 참조가 발생했고, 그 문제를 해결하기 위해 외래키를 없애는 과정에서 1:N 매칭을 따지며 1 인 테이블에서 외래키를 제거하였다.

다) 이 데이터베이스를 전 세계적으로 사용할 수 있도록 테이블과 애트리뷰트들의 이름을 영어로 하였고, 그 애트리뷰트의 특성을 한눈에 알 수 있는 단어로 이름을 선정하였다.

라) 유도된 애트리뷰트는 관계 데이터베이스에서 릴레이션의 애트리뷰트로 사용하지 않는 것이 좋다 하여 인바디 테이블에서 유도된 애트리뷰트인 인바디 점수를 제거하였다.

마) 다만, 불가피하게 만료일자와 같은 애트리뷰트는 회원의 이용가능 여부를 판단하기 위해 확인해야 함으로 프로시저를 통해 자동으로 update 되도록 구현하였다.

바) 회원과 직원의 관계는 PT 밖에 없는데 PT 테이블이 존재하기 때문에 Member 테이블에서 Trainer_ID 애트리뷰트를 제거하였다.

사) Managing 테이블에서 Feedback 애트리뷰트가 저장하는 내용이 너무 긴 문자열이 될 것 같아서 제거하였다.

■ 정규화 과정

✓ 1차 정규화

다치 애트리뷰트를 처음부터 만들지 않았다.

✓ 2차 정규화

PT 테이블에서 기본 키를 Member ID 와 Trainer ID 의 조합으로 이루어진 복합 키에서 새로운 PT ID 라는 애트리뷰트 추가하여 기본 키로 선정하였다.

✓ 3차 정규화

- Member, Staff 테이블에서 Resident_number 애트리뷰트를 제거하였다.
- Managing 테이블에서 Managing_ID 제거 후, Inbody_ID를 기본 키로 설정하였다.
- Payment 테이블에서 Content 애트리뷰트를 제거하였다.
- Duty 테이블에서 Content 애트리뷰트를 제거하였다.
- Place 테이블에서 Sector 애트리뷰트를 제거하였다.
- PT 테이블에서 Trainer_ID 애트리뷰트를 제거하였다.
- Machine, Object 테이블에서 Place_ID 애트리뷰트를 제거하였다.
- Place 테이블에 Machine_ID, Object_ID 애트리뷰트를 외래 키로 추가하였다. Place 테이블은 릴레이션 테이블로 두기로 했다가 이행적 종속성이 계속 발생해서 Place 테이블에서 제거하고 Duty, Machine, Object 테이블에 Place 애트리뷰트를 추가하였다

■ 물리적 설계(인덱스 등)

i. 인덱스 설계

설계한 테이블에서 가장 중요하고, 가장 많이 쓰이는 테이블과 질의가 무엇일지 고민해 보았고, 헬스장에서 회원들이 자주 이용하는 인바디 측정과 직원의 회원 조회, 운동기구 검색일 것이라 판단하였다.

따라서, 총 3개의 인덱스를 형성하였고 다음과 같다.

가) 회원의 Inbody 조회 : 회원마다 자신이 최근에 측정한 인바디를 많이 조회할 것으로 판단하여 Member_Id와 Date의 내림차순의 조합으로 인덱스를 정의하였다.

```
CREATE INDEX idx_private_inbody ON inbody (member_id, date DESC);
```

나) 회원 정보 조회

```
CREATE INDEX idx_member_name ON member (name);
```

다) 운동기구 조회

```
CREATE INDEX idx_machine ON machine (name);
```

ii. 트리거

A) 할인 적용

INSERT된 Member(회원) 테이블에서 Discount_Id 애트리뷰트의 값이 NULL이 아니라면, 할인을 받은 회원이다. 따라서, IF구문을 통해 Discount_Id에 해당하는 Rate(할인율) 애트리뷰트 값을 통하여 최종 할인된 값을 계산하고, Sales(매출) 테이블에 INSERT 한다.

```
DELIMITER //

CREATE TRIGGER apply_discount
AFTER INSERT ON member
FOR EACH ROW
BEGIN
DECLARE result INT;
SET result = (SELECT price FROM payment WHERE payment_id = NEW.payment_id);
IF NEW.discount_id IS NOT NULL THEN
  SET result = result * (1 - (SELECT rate FROM discount
                              WHERE discount_id = NEW.discount_id));
END IF;

INSERT INTO sales(member_id, content, profit, time)
VALUES (NEW.member_id, 'registration', result, CURRENT_TIMESTAMP());
END //

DELIMITER ;
```

B) 회원 신규 등록

INSERT된 Member(회원) 테이블의 Registration_Date(등록일) 애트리뷰트의 날짜와 Payment(이용권) 테이블의 Duration(이용가능 기간) 애트리뷰트의 날짜를 계산하여 Member 테이블의 Expiration_Date(만료일)에 자동으로 기입한다.

```
DELIMITER //
CREATE TRIGGER trg_expiration_date
Before INSERT ON member
FOR EACH ROW
BEGIN
declare duration_date int;
SELECT duration INTO duration_date FROM Payment
                                   WHERE payment_id = new.Payment_ID;
set new.expiration_date = timestamp(DATE_ADD(new.registration_date,
                          interval duration_date MONTH));
end //
DELIMITER ;
```

C) 환불

Member(회원) 테이블의 투플이 삭제되었다고 한다면,탈퇴한 회원에게 남은 기간만큼 비례해서 환불한다. 환불된 금액은 Expense(지출) 테이블에 자동으로 환불 내역이 생성된다. 이때, 1개월은 30일로 가정한다. 공식은 다음과 같다.

$$originMoney * (remainDay/(originDay * 30))$$

```
DELIMITER //
CREATE TRIGGER before_delete_member
BEFORE DELETE ON member
FOR EACH ROW
BEGIN
DECLARE origin_money INT;
DECLARE tmp1 TIMESTAMP;
DECLARE remain_day INT;
DECLARE origin_day INT;
DECLARE result_money INT;

SELECT profit INTO origin_money FROM sales WHERE member_id = OLD.member_id;
SELECT expiration_date INTO tmp1 FROM member WHERE member_id = OLD.member_id;
SET remain_day = DATEDIFF(tmp1, CURRENT_TIMESTAMP());

IF (remain_day > 0) THEN
SELECT duration INTO origin_day
FROM payment WHERE payment_id = old.payment_id;

SET result_money = CAST(origin_money * (CAST(remain_day AS DECIMAL(4, 1)) /
(origin_day * 30)) AS UNSIGNED);
INSERT INTO expense (content, price, time)
VALUES ('refund', result_money, CURRENT_TIMESTAMP());
END IF;
END //
DELIMITER ;
```

D) 입장

PT를 받는 회원은 입장 시, 그날 PT를 한다고 가정하고 PT의 남은 횟수를 차감하는 방식으로 트리거를 작성하였지만, AFTER SELECT문을 트리거 조건으로 사용할 수 없다고 하여 실패하였다.

```
DELIMITER //

CREATE TRIGGER decrease_pt_count
AFTER SELECT ON member
FOR EACH ROW
BEGIN
    IF NEW.expiration_date IS NOT NULL AND (SELECT category FROM member
        WHERE member_id = NEW.member_id) == 'PT'
    THEN
        UPDATE pt SET remain_count = remain_count - 1
        WHERE member_id = NEW.member_id;
    END IF;
END //

DELIMITER ;
```

iii. 프로시저

A) 만료 여부 확인

매일 정각마다 회원들의 만료여부를 판단한다. 만료가 되었다면 Payment_Id(요금제), Registration_Date(등록 일자), Expiration_Date(만료 일자)를 NULL로 UPDATE해주어 해당 회원을 더이상 유효하지 않는 상태로 만든다.
매일 정각마다 해당 프로시저를 호출하는 이벤트 스케줄러를 생성하여 매일 정각마다 만료여부를 확인해주도록 하였다.

```
DELIMITER //

CREATE PROCEDURE check_member_expiration()
BEGIN
DECLARE currentDate TIMESTAMP;
SET currentDate = CURRENT_TIMESTAMP;

UPDATE member
SET expiration_date = NULL, payment_id = NULL, registration_date = NULL
WHERE expiration_date < currentDate;

END //

DELIMITER ;

CREATE EVENT daily_member_expiration_check
ON SCHEDULE
   EVERY 1 DAY
   STARTS CURDATE() + INTERVAL 1 DAY
DO
CALL check_member_expiration();
```

iv. 이벤트

A) 당번 자동 갱신

직원들이 헬스장의 관리를 위해, 하루에 각자가 맡을 일을 부여받는 당번의 기능이 존재한다. 할일의 목록은 다음과 같으며 이는 정각마다 자동으로 갱신되기에 직원들은 매일 다른 일을 부여받는다.

이 기능을 위해 직원의 수와 할 일의 수는 같다고 가정하였고, 직원의 추가 및 삭제가 일어나지 않는다고 가정하였다. 할 일의 목록은 다음과 같다.

1. Pilates Zone: 물품 정리, 바닥+거울 청소
2. Stretching Zone: 물품 정리, 바닥+거울 청소
3. Free weight Zone: 원판 정리, 거울 청소
4. Machine Zone: 기구 정리, 바닥 청소
5. Dressing room Zone: 빨래, 샤워실 물청소

```
DELIMITER //

CREATE EVENT renew_duty_event
ON SCHEDULE
    EVERY 1 DAY
    STARTS CURDATE() + INTERVAL 1 DAY
DO
    BEGIN
        DECLARE max INT;
        SET max = (SELECT COUNT(*) FROM duty);
        UPDATE staff
        SET duty_id = CASE
            WHEN (duty_id + 1) > max THEN 1
            ELSE (duty_id + 1)
        END;
    END //

DELIMITER ;
```

■ DML을 활용한 기능 시연

➢ 회원 등록 시

```
insert into member(Trainer_id, payment_id, discount_id, name, age, address, job,
phone_number, category, registration_date, sex)
values (5, 2, null, 'Han', 26, '010 Street', 'Developer', '01051893255', 'FREE',
current_timestamp, 'M');
```

```
mysql> select * from member;
+-----------+------------+--------------+------+-----+-------------+-----------+--------------+----------+---------------------+---------------------+-----+-------------+
| Member_ID | Trainer_Id | payment_id   | Name | Age | Address     | Job       | Phone_Number | Category | registration_date   | expiration_date     | Sex | discount_id |
+-----------+------------+--------------+------+-----+-------------+-----------+--------------+----------+---------------------+---------------------+-----+-------------+
|         1 |          3 |            2 | Woo  |  26 | 159 Street  | Student   | 01012345678  | FREE     | 2023-06-08 20:20:04 | 2023-12-08 20:20:04 | M   |           1 |
|        27 |          4 |            1 | Yu   |  24 | 401 Street  | Designer  | 01024151259  | PT       | 2023-06-08 20:24:54 | 2023-09-08 20:24:54 | F   |           1 |
|        28 |       NULL |            1 | Kyung|  25 | 123 Street  | Engineer  | 01000000001  | PT       | 2023-05-23 10:00:22 | 2023-08-23 10:00:22 | M   |           1 |
|        29 |       NULL |         NULL | Lee  |  27 | 111 Street  | Salaryman | 01083745739  | FREE     | NULL                | NULL                | F   |        NULL |
|        33 |       NULL |            1 | Park |  29 | 456 Street  | Teacher   | 01011110000  | FREE     | 2023-06-08 21:08:15 | 2023-09-08 21:08:15 | F   |           1 |
|        34 |          4 |            1 | Lee  |  27 | 133 Street  | Student   | 01022221111  | Pilates  | 2023-06-13 10:00:22 | 2023-09-13 10:00:22 | F   |           1 |
|        36 |       NULL |         NULL | Lim  |  27 | 122 Street  | Salaryman | 01083745600  | FREE     | NULL                | NULL                | F   |        NULL |
|        37 |          5 |            2 | Han  |  26 | 010 Street  | Developer | 01051893255  | FREE     | 2023-06-08 21:24:37 | 2023-12-08 21:24:37 | M   |        NULL |
+-----------+------------+--------------+------+-----+-------------+-----------+--------------+----------+---------------------+---------------------+-----+-------------+
8 rows in set (0.00 sec)

mysql> select * from sales;
+----------+-----------+--------------+--------+---------------------+
| Sales_ID | Member_ID | Content      | Profit | Time                |
+----------+-----------+--------------+--------+---------------------+
|        1 |        36 | registration | 240000 | 2023-06-08 21:20:33 |
|        2 |        37 | registration | 360000 | 2023-06-08 21:24:37 |
+----------+-----------+--------------+--------+---------------------+
2 rows in set (0.01 sec)
```

회원을 등록할 때, 삽입과 동시에 회원이 요금을 낸 기록이 매출 테이블에 기록되었다.

```
mysql> select * from discount;
+-------------+------+
| Discount_Id | rate |
+-------------+------+
|           1 | 0.10 |
|           2 | 0.20 |
|           3 | 0.25 |
|           4 | 0.15 |
+-------------+------+
4 rows in set (0.00 sec)
```

위의 테이블은 할인 정보를 보여준다.

```
insert into member(Trainer_id, payment_id, discount_id, name, age, address, job,
phone_number, category, registration_date, sex)
values (4, 2, 2, 'Ahn', 22, '101 Street', 'Building owner', '01077777777', 'Pilates',
current_timestamp, 'F');
```

```
mysql> select * from sales;
+----------+-----------+--------------+--------+---------------------+
| Sales_ID | Member_ID | Content      | Profit | Time                |
+----------+-----------+--------------+--------+---------------------+
|        1 |        36 | registration | 240000 | 2023-06-08 21:20:33 |
|        2 |        37 | registration | 360000 | 2023-06-08 21:24:37 |
|        3 |        38 | registration | 288000 | 2023-06-08 21:26:20 |
+----------+-----------+--------------+--------+---------------------+
3 rows in set (0.00 sec)
```

위에서 등록한 회원과 똑같은 요금제로 할인률 20%를 적용한 회원을 등록하면 매출 테이블에 할인된 가격으로 기록된 것을 확인할 수 있다.

- 회원 탈퇴 시

```
mysql> select * from inbody;
+------------+-----------+---------------------+--------+------+--------+--------+-------+---------+---------------+----------+
| inbody_id | member_id | date                | muscle | fat  | weight | height | water | protein | basal_metabolism | minerals |
+------------+-----------+---------------------+--------+------+--------+--------+-------+---------+---------------+----------+
|          1 |        37 | 2023-06-08 21:31:42 |   32.1 | 17.9 |   70.1 |  175.2 |  45.3 |    11.5 |          1651 |     4.03 |
|          2 |        38 | 2023-06-08 21:33:16 |   33.3 | 18.0 |   75.4 |  178.1 |  44.3 |    12.5 |          1750 |     4.37 |
|          3 |        37 | 2023-06-08 21:33:53 |   32.7 | 17.8 |   70.3 |  175.3 |  44.9 |    12.1 |          1700 |     4.12 |
+------------+-----------+---------------------+--------+------+--------+--------+-------+---------+---------------+----------+
3 rows in set (0.00 sec)
```

```
mysql> select * from pt;
+-------+-----------+--------------+
| pt_id | member_id | remain_count |
+-------+-----------+--------------+
|     6 |        36 |           10 |
|     7 |        38 |            9 |
|     8 |        37 |            8 |
+-------+-----------+--------------+
3 rows in set (0.01 sec)
```

```
mysql> select * from managing;
+--------------+-----------+------------+-------+-------------------------------------+---------------------+
| managing_id | member_id | trainer_id | grade | feedback                            | date                |
+--------------+-----------+------------+-------+-------------------------------------+---------------------+
|            1 |        37 |          2 | bad   | 숨, 아식 자제 필요                  | 2023-06-13 11:00:22 |
|            2 |        38 |          4 | good  | 몸의 밸런스가 매우 좋음             | 2023-08-14 11:00:22 |
|            3 |        37 |          5 | good  | 몸의 밸런스가 매우 훌륭함           | 2023-07-14 11:00:22 |
+--------------+-----------+------------+-------+-------------------------------------+---------------------+
3 rows in set (0.00 sec)
```

기존에 있던 Member_id 가 37 번인 회원은 Inbody, PT, Managing 테이블에 정보가 기록되어 있었다.

```
mysql> delete from member where member_id = 37;
Query OK, 1 row affected (0.01 sec)
```

하지만, 환불이나 양도의 이유로 회원 테이블에 삭제될 시 아래와 같은 변화들이 발생한다..

```
mysql> select * from expense;
+------------+-----------+------------+----------+---------+--------+---------------------+
| Expense_ID | Object_ID | Machine_Id | Staff_ID | Content | price  | Time                |
+------------+-----------+------------+----------+---------+--------+---------------------+
|         18 |      NULL |       NULL |     NULL | refund  |   NULL | 2023-06-08 21:24:21 |
|         19 |      NULL |       NULL |     NULL | refund  |   NULL | 2023-06-08 21:24:25 |
|         20 |      NULL |       NULL |     NULL | refund  | 366000 | 2023-06-08 21:36:26 |
+------------+-----------+------------+----------+---------+--------+---------------------+
3 rows in set (0.00 sec)
```

```
mysql> select * from inbody;
+-----------+-----------+---------------------+---------+------+--------+--------+---------+---------+-----------------+----------+
| inbody_id | member_id | date                | muscle  | fat  | weight | height | water   | protein | basal_metabolism | minerals |
+-----------+-----------+---------------------+---------+------+--------+--------+---------+---------+-----------------+----------+
|         2 |        38 | 2023-06-08 21:33:16 |    33.3 | 18.0 |   75.4 |  178.1 |    44.3 |    12.5 |            1750 |     4.37 |
+-----------+-----------+---------------------+---------+------+--------+--------+---------+---------+-----------------+----------+
1 row in set (0.00 sec)
```

```
mysql> select * from pt;
+-------+-----------+--------------+
| pt_id | member_id | remain_count |
+-------+-----------+--------------+
|     6 |        36 |           10 |
|     7 |        38 |            9 |
+-------+-----------+--------------+
2 rows in set (0.00 sec)
```

```
mysql> select * from managing;
+-------------+-----------+------------+-------+-------------------------------+---------------------+
| managing_id | member_id | trainer_id | grade | feedback                      | date                |
+-------------+-----------+------------+-------+-------------------------------+---------------------+
|           2 |        38 |          4 | good  | 몸의 밸런스가 매우 좋음       | 2023-08-14 11:00:22 |
+-------------+-----------+------------+-------+-------------------------------+---------------------+
1 row in set (0.00 sec)
```

Member_Id 가 37 번이었던 회원의 모든 기록이 사라진 것을 확인할 수 있다.

- 회원 만료 시

```
mysql> select * from member;
+-----------+------------+--------------+------+-----+-------------+---------------+--------------+----------+---------------------+---------------------+-----+-------------+
| Member_ID | Trainer_Id | payment_id   | Name | Age | Address     | Job           | Phone_Number | Category | registration_date   | expiration_date     | Sex | discount_id |
+-----------+------------+--------------+------+-----+-------------+---------------+--------------+----------+---------------------+---------------------+-----+-------------+
|         1 |          3 |            2 | Woo  |  26 | 159 Street  | Student       | 01012345678  | FREE     | 2023-06-08 20:20:04 | 2023-12-08 20:20:04 | M   |           1 |
|        27 |          4 |            1 | Yu   |  24 | 401 Street  | Designer      | 01024151269  | PT       | 2023-06-08 20:24:54 | 2023-09-08 20:24:54 | F   |           1 |
|        28 |       NULL |            1 | Kyung|  25 | 123 Street  | Engineer      | 01000000001  | PT       | 2023-05-23 10:00:22 | 2023-08-23 10:00:22 | M   |           1 |
|        29 |       NULL |         NULL | Lee  |  27 | 111 Street  | Salaryman     | 01083745739  | FREE     | NULL                | NULL                | F   |        NULL |
|        30 |          5 |            2 | Han  |  26 | 010 Street  | Developer     | 01051893255  | FREE     | 2023-06-08 20:51:26 | 2023-12-08 20:51:26 | M   |        NULL |
|        31 |          4 |            2 | Ahn  |  22 | 101 Street  | Building owner| 01077777777  | Pilates  | 2023-06-08 20:51:53 | 2023-12-08 20:51:53 | F   |           2 |
|        33 |       NULL |            1 | Park |  29 | 456 Street  | Teacher       | 01011110000  | FREE     | 2023-06-08 21:08:15 | 2023-09-08 21:08:15 | F   |           1 |
|        34 |          4 |            1 | Lee  |  27 | 133 Street  | Student       | 01022221111  | Pilates  | 2023-06-13 10:00:22 | 2023-09-13 10:00:22 | F   |           1 |
|        36 |       NULL |            1 | Lin  |  27 | 122 Street  | Salaryman     | 01083745500  | FREE     | 2023-03-21 22:00:11 | 2023-06-21 22:00:11 | F   |        NULL |
+-----------+------------+--------------+------+-----+-------------+---------------+--------------+----------+---------------------+---------------------+-----+-------------+
9 rows in set (0.00 sec)

mysql> select * from member;
+-----------+------------+--------------+------+-----+-------------+---------------+--------------+----------+---------------------+---------------------+-----+-------------+
| Member_ID | Trainer_Id | payment_id   | Name | Age | Address     | Job           | Phone_Number | Category | registration_date   | expiration_date     | Sex | discount_id |
+-----------+------------+--------------+------+-----+-------------+---------------+--------------+----------+---------------------+---------------------+-----+-------------+
|         1 |          3 |            2 | Woo  |  26 | 159 Street  | Student       | 01012345678  | FREE     | 2023-06-08 20:20:04 | 2023-12-08 20:20:04 | M   |           1 |
|        27 |          4 |            1 | Yu   |  24 | 401 Street  | Designer      | 01024151269  | PT       | 2023-06-08 20:24:54 | 2023-09-08 20:24:54 | F   |           1 |
|        28 |       NULL |            1 | Kyung|  25 | 123 Street  | Engineer      | 01000000001  | PT       | 2023-05-23 10:00:22 | 2023-08-23 10:00:22 | M   |           1 |
|        29 |       NULL |         NULL | Lee  |  27 | 111 Street  | Salaryman     | 01083745739  | FREE     | NULL                | NULL                | F   |        NULL |
|        30 |          5 |            2 | Han  |  26 | 010 Street  | Developer     | 01051893255  | FREE     | 2023-06-08 20:51:26 | 2023-12-08 20:51:26 | M   |        NULL |
|        31 |          4 |            2 | Ahn  |  22 | 101 Street  | Building owner| 01077777777  | Pilates  | 2023-06-08 20:51:53 | 2023-12-08 20:51:53 | F   |           2 |
|        33 |       NULL |            1 | Park |  29 | 456 Street  | Teacher       | 01011110000  | FREE     | 2023-06-08 21:08:15 | 2023-09-08 21:08:15 | F   |           1 |
|        34 |          4 |            1 | Lee  |  27 | 133 Street  | Student       | 01022221111  | Pilates  | 2023-06-13 10:00:22 | 2023-09-13 10:00:22 | F   |           1 |
|        36 |       NULL |         NULL | Lin  |  27 | 122 Street  | Salaryman     | 01083745500  | FREE     | NULL                | NULL                | F   |        NULL |
+-----------+------------+--------------+------+-----+-------------+---------------+--------------+----------+---------------------+---------------------+-----+-------------+
9 rows in set (0.00 sec)
```

매일 정각마다 등록 기간이 끝나서 만료가 된 회원들의 요금제, 등록 일자, 만료 일자 애트리뷰트를 NULL 로 갱신하는 프로시저를 통하여 만료 상태로 만들어 준 모습이다.

➢ 당번 최신화

```
mysql> select * from duty;
+---------+---------------+------+
| Duty_id | place         | time |
+---------+---------------+------+
|       1 | Pilates       |   22 |
|       2 | Stretching    |   23 |
|       3 | Free weight   |    6 |
|       4 | Machine       |    6 |
|       5 | Dressing room |   23 |
+---------+---------------+------+
5 rows in set (0.00 sec)
```

위의 테이블은 할 일 목록을 나타낸 것이다.

```
mysql> select * from staff;
+----------+---------+------+-----+---------+-----+--------+----------+
| Staff_ID | Duty_ID | Name | Age | Address | Sex | Salary | Position |
+----------+---------+------+-----+---------+-----+--------+----------+
|        1 |       5 | Choi |  24 | Andong  | M   |      0 | Manager  |
|        2 |       1 | Yoon |  23 | Hanam   | M   |      0 | Boss     |
|        3 |       2 | Kwak |  24 | Jinhae  | M   |     10 | Trainer  |
|        4 |       3 | Park |  20 | Seoul   | F   |     20 | Trainer  |
|        5 |       4 | Oh   |  21 | Jeju    | M   |     10 | Trainer  |
+----------+---------+------+-----+---------+-----+--------+----------+
5 rows in set (0.00 sec)

mysql> select * from staff;
+----------+---------+------+-----+---------+-----+--------+----------+
| Staff_ID | Duty_ID | Name | Age | Address | Sex | Salary | Position |
+----------+---------+------+-----+---------+-----+--------+----------+
|        1 |       1 | Choi |  24 | Andong  | M   |      0 | Manager  |
|        2 |       2 | Yoon |  23 | Hanam   | M   |      0 | Boss     |
|        3 |       3 | Kwak |  24 | Jinhae  | M   |     10 | Trainer  |
|        4 |       4 | Park |  20 | Seoul   | F   |     20 | Trainer  |
|        5 |       5 | Oh   |  21 | Jeju    | M   |     10 | Trainer  |
+----------+---------+------+-----+---------+-----+--------+----------+
5 rows in set (0.00 sec)
```

할 일이 갱신되는 이벤트를 만들고 실행되었을 때 결과다. 똑같은 구문을 시행했는데 할 일이 달라진 것을 확인할 수 있다.

◆ 설계 결과 및 분석

■ 기능 보완이 필요한 부분 정리 및 개선 방안 제시

- 당번 최신화 알고리즘을 구현하여 직원의 업무 능률이 향상될 것으로 보인다. 하지만, 현재, 직원의 당번의 할 일의 개수가 동일해야만 정상적으로 작동하는 방식이다. 따라서 새로운 직원이 삽입되거나, 당번의 할 일이 새로 추가가 된다면 당번 최신화 알고리즘은 정상적으로 작동하지 않는다. 따라서, 데이터의 갱신을 할 수 있도록 알고리즘 개선이 최우선 목표이다.
- 또한, Object 테이블과 Machine 테이블에 대한 트리거 추가 여부를 검토하여 구현해 나아갈 계획이다. 주요 핵심 기능에만 집중하는 것이 아닌 자산관리에도 효율적으로 할 수 있도록 다양한 기능을 접목해 나아갈 계획이다.
- 마지막으로 회원들의 소중한 개인 정보 보안을 위해 접근 제어와 권한 관리에 집중하여 무분별한 데이터 접근을 차단할 예정이다. 현재 모든 권한을 지닌 root 계정만 두어 데이터베이스를 관리하고 있다. 따라서 직원 별, 회원 별 계정을 생성한 뒤 각자의 권한을 부여하여 데이터베이스 보안을 강화할 것이다.

■ 설계 목표의 달성 정도를 분석하여 기술

우선, 정규화를 통한 데이터 중복의 최소화하였다. 또한 각 테이블마다 적 절한 제약 조건을 정의해 줌으로써 데이터 무결성을 유지하였고, 직원 당번 최신화 알고리즘을 구현하여 업무의 효율을 증대할 수 있었다. 사용자의 요구사항을 충족시키고자 이윤을 최대화할 수 있는 정보가 무엇일지 생각해보았고, 회원의 직업, 나이를 바탕으로 다양한 이벤트를 열어 회원의 유입이 가능할 것이라 판단하였다. 이를 통해 이윤의 극대화를 누릴 수 있을 것으로 기대하고 있다. 무엇보다 프로시저 및 트리거, 이벤트 스케줄링과 같은 데이터베이스가 제공하는 기능을 최대한 활용하여 데이터 자동 갱신이 가능 해졌다.

◆결론 및 활용분야

■ 결론

요구사항을 정의하는 과정에서 우리 서비스만의 특별함이 뭐가 있을지 생각해 보았고, 헬스장을 운영하는 입장에서의 핵심 요구사항인 '이윤 극대화', 그리고 직원의 업무 개선을 위한 '운영 자동화'에 초점을 두고 설계를 시작하였다. 헬스장에서 운동을 하였던 경험을 바탕으로, 헬스장이라는 실세계의 엔티티들을 우리 나름대로 정의하여 이를 ERD로 표현해 보았고, 정규화 과정을 거치며 안정적인 데이터베이스 스키마를 구성해 볼 수 있었다. 자주 조회될 수 있는 테이블이 무엇인지 고민해보며 인덱스 설계를 해 볼 수 있었고, 자동화를 위해 프로시저 설계 및 트리거, 이벤트를 구현해보며 더욱 다채로운 기능들을 접해보고 공부해볼 수 있었다.

데이터베이스 설계과정에 의거하여 최종적으로 프로젝트를 완성해 본 데이터베이스의 모습을 보니 우리만의 요구사항이 충족된 '하나의 서비스'가 구축되었다는 것에 뿌듯함을 느낄 수 있었다. 전반적으로 수업시간에 배운 내용을 토대로 설계를 진행하며 응용해볼 수 있어 더욱 흥미를 가지며 프로젝트에 임할 수 있었다

요구사항 정의 후 바로 논리적 설계로 가는 것이 아닌 개념적 설계를 통한 실제 데이터베이스 스키마를 작성해보며 경험을 바탕으로 절차적으로 설계하는 것이 매우 중요하다는 것을 깨달을 수 있었다.

■ 목표

데이터베이스 설계가 마무리된 만큼 이제 그치지 않고, 실제 애플리케이션을 개발하여 배포까지 해보는 과정으로 확대할 계획이다. 사용자 측면에서 데이터베이스 설계에 이상이 없는지 확인해보며 서비스 기능을 확대해 나아갈 것이다.

◆ 참고 문헌

- Mysql
  - https://www.mysql.com
- MySQL 이벤트 스케줄러
  - https://jungeunpyun.tistory.com/64
- 프로시저
  - https://wakestand.tistory.com/518
  - https://velog.io/@donghoim/MySQL-저장-프로시저-Stored-Procedure
- 트리거
  - https://inpa.tistory.com/entry/MYSQL-📚-트리거
  - https://seung.tistory.com/entry/MySQL-트리거 Trigger-사용법