

# 운동 후 리워드를 지급하는 프로그램 데이터베이스

20224251 : 김성현

# 목차 Table of Contents

◆ 계획 수립

◆ 목적

◆ 시나리오

◆ 엔티티 타입 선정

◆ 개념설계

◆ 관계 설계

◆ ER 다이어그램

◆ 논리 설계

◆ 데이터베이스 구축

◆ 질의

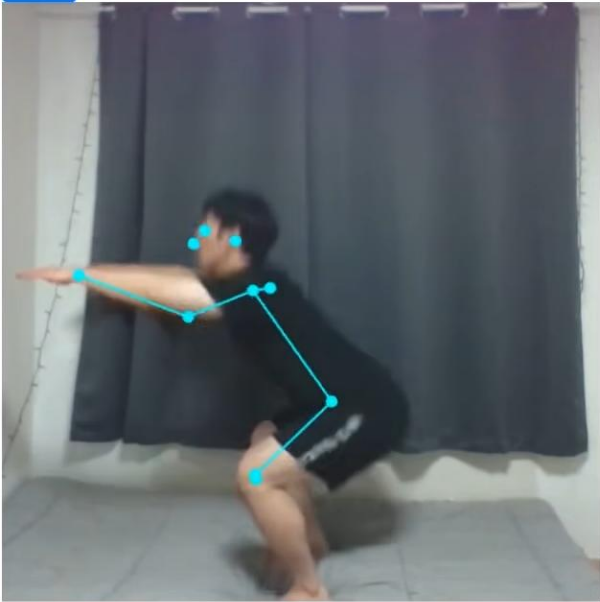
# 🖥 시작하기에 앞서..

Overview | elegant-fermi-dff276 x AI Fitness Trainer x +


elegant-fermi-dff276.netlify.com

## 인공지능(AI) Fitness Trainer

Start

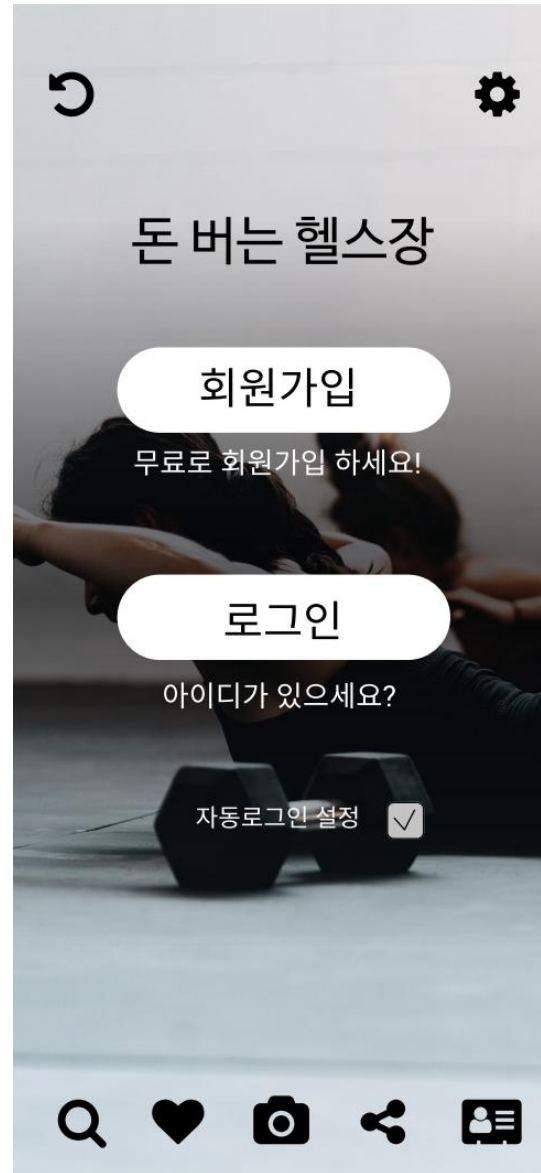
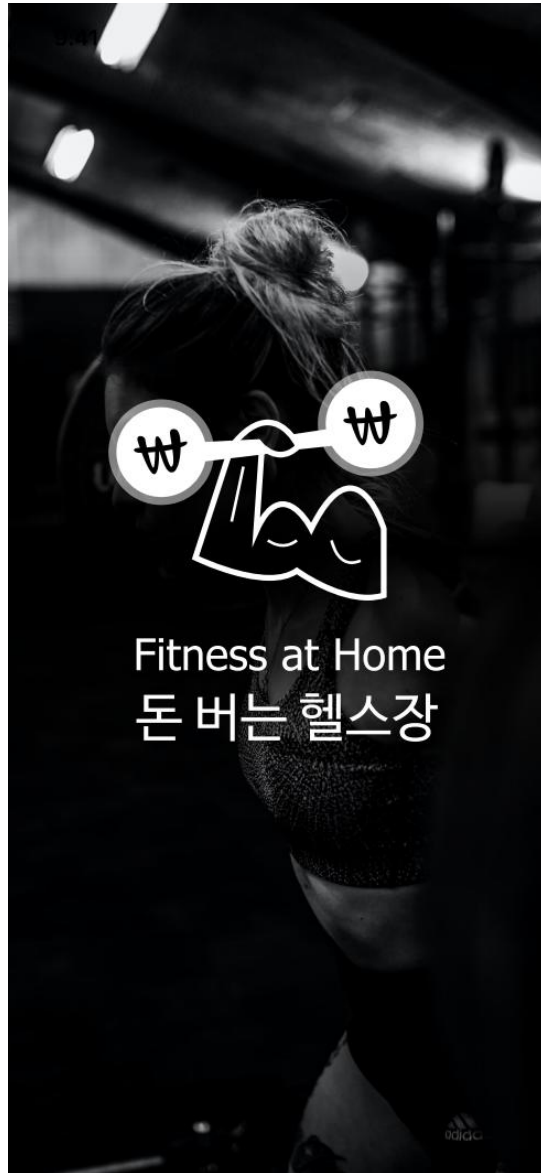


stand: 0.00  
squat: 1.00  
bent: 0.00



컴퓨터 (웹) 버전 계획

# 🖥 시작하기에 앞서..



모바일 (앱) 버전 계획

## ◆ 계획 수립 및 목적

- 개인 토이 프로젝트로 개발중인 서비스에다가 수업시간에 배운 데이터베이스 지식을 적용해 보고 싶어서 시작.
- 기존에 만보를 걸었을 경우 약간의 리워드를 지급하고, 그 리워드를 가지고 기프티콘을 구매할 수 있게 해주는 앱에서 만보 걷기 뿐만 아니라 다른 운동을 했을 시에도 리워드를 받을 수 있게 만들면 사용자의 건강에 조금 더 보탬이 될 것이다.

## ◆ 업무 시나리오

- 고객들은 이 프로그램을 이용할 때 아이디와 비밀번호를 입력하고 로그인을 한다.  
(회원이 아닐 경우, 회원 가입을 통해 회원 등록을 하게 된다.) 이후 운동 목록들 중에서 자신이 하고자 하는 운동을 선택한 후 스마트폰 또는 컴퓨터 웹캠에 자신이 운동을 하고 있는 모습을 인식시켜 카운트가 올라가게 된다. 운동이 끝나면, 개수에 따라 리워드 포인트를 지급한다. 포인트가 일정 수준 쌓이면 고객은 문화상품권, 또는 편의점과 음식점에서 사용할 수 있는 기프티콘을 포인트와 교환할 수 있다.

## ◆ '엔티티 타입 도출' 을 이용한 엔티티 타입 선정

- 1. 시나리오에서 명사를 구분한다.
- 고객, 프로그램, 아이디, 비밀번호, 로그인, 회원 가입, 회원 등록, 운동, 스마트폰, 컴퓨터, 인식, 카운트, 개수, 리워드, 포인트, 문화상품권, 기프티콘, 교환

## ◆ '엔티티 타입 도출' 을 이용한 엔티티 타입 선정

- 2. 개념이 불분명한 것, 광범위한 것을 제거한다.
- 고객, 프로그램, 아이디, 비밀번호, 로그인, 회원 가입, 회원 등록, 운동, 스마트폰, 컴퓨터, 인식, 카운트, 개수, 리뷰드, 포인트, 문화상품권, 기프티콘, 교환



## ◆ '엔티티 타입 도출' 을 이용한 엔티티 타입 선정

- 3. 엔티티 타입의 특성이거나 속성의 값을 제거한다.
- 고객, **아이디**, **비밀번호**, 로그인, 회원 가입, 회원 등록, 운동, 스마트폰, 컴퓨터, 카운트, 개수, 리워드, **포인트**, 문화상품권, 기프티콘, 교환

## ◆ '엔티티 타입 도출' 을 이용한 엔티티 타입 선정

- 4. 포괄적인 업무 프로세스에 해당하는 명사를 제거한다.
- 고객, 로그인, 회원 가입, 회원 등록, 운동, 스마트폰, 컴퓨터, 카운트, 개수, 리워드, 문화 상품권, 기프티콘, 교환

## ◆ '엔티티 타입 도출' 을 이용한 엔티티 타입 선정

- 5. 중복되는 명사를 제거한다.
- 문화상품권과 기프티콘을 '교환 가능 상품'으로 변경.
- '교환' 은 관계 이므로 역시 제거.
- 고객, 운동, 카운트, 개수, 리워드, 문화상품권, 기프티콘, 교환

## ◆ '엔티티 타입 도출' 을 이용한 엔티티 타입 선정

- 6. 최종 선정.
- 고객, 운동, 리워드(포인트), 교환 가능 상품

## ◆ 고객 테이블

- 고객 ID (Customer\_ID) (Primary Key)
- 비밀번호 (Customer\_Password)
- 이름 (Customer\_Name)
- 이메일 주소 (E-Mail\_Address)
- 휴대폰 번호 (Customer\_Phone\_number)

## ◆ 운동 테이블

- 운동 ID (Exercise\_ID) (Primary Key)
- 운동명 (Exercise\_Name)
- 개수 (Exercise\_Count)

## ◆ 리워드 테이블

- 리워드 ID (Reward\_ID) (Primary Key)
- 리워드명 (Reward\_Name)
- 보유 포인트 (Reward\_Amount)

## ◆ 교환 가능 상품 테이블

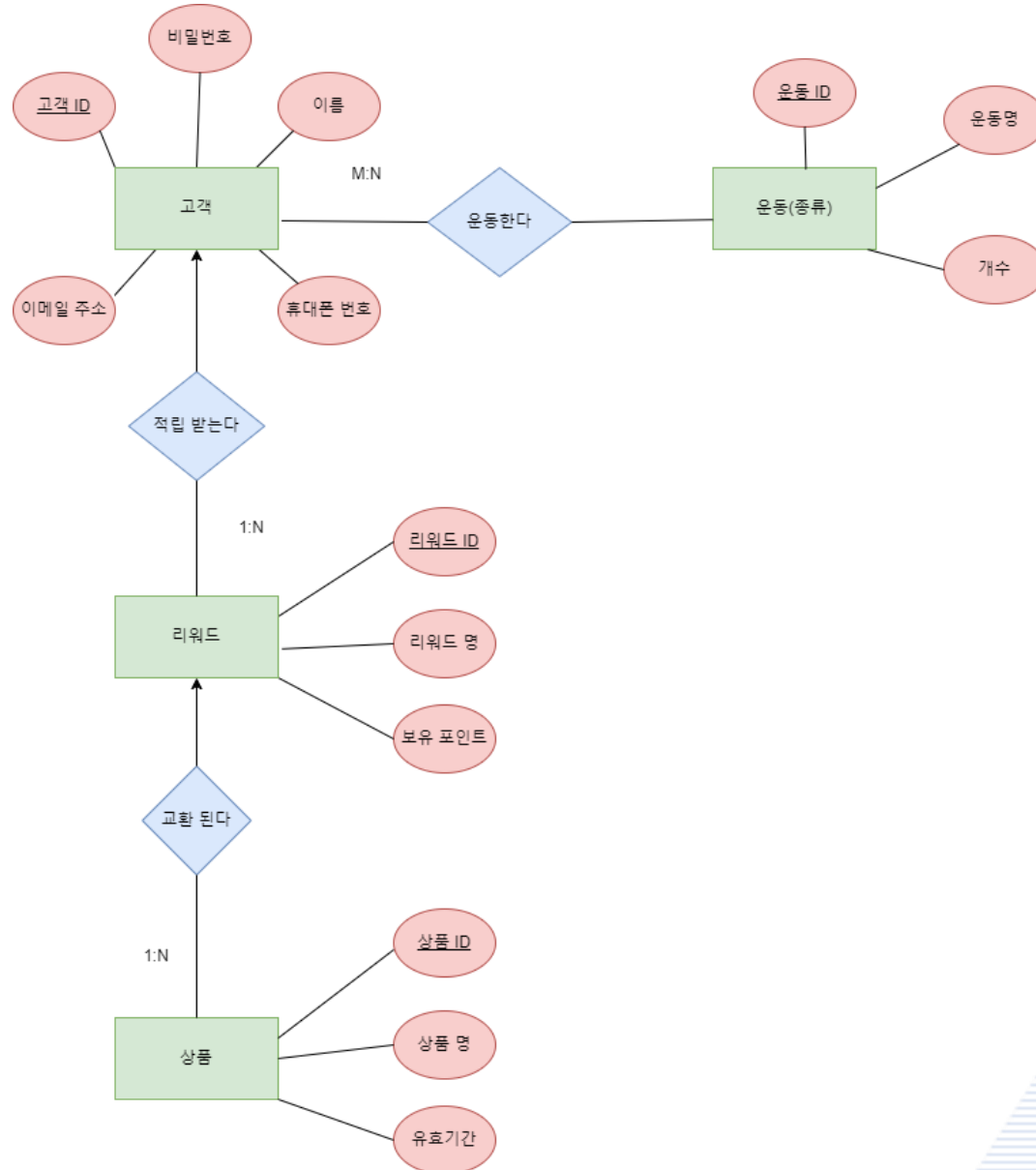
- 상품 ID (Product\_ID) (Primary Key)
- 상품명 (Product\_Name)
- 유효기간 (Expiry\_Date)



## ◆ 관계 설계

- 고객과 운동: 고객은 여러 운동을 할 수 있고, 각 운동은 여러 고객에 의해 수행될 수 있다. (다대다 관계)
- 고객과 리워드: 고객은 리워드를 적립 받을 수 있다. (일대다 관계)
- 교환은 하나의 고객에 의해 이루어진다.
- 리워드와 상품: 각 리워드는 하나 이상의 상품으로 교환될 수 있다. (일대다 관계)

## ◆ ERD



## ◆ 테이블 설계

- 고객:

고객 ID	비밀번호	고객 이름	이메일 주소	휴대폰 번호
Customer_ID	Customer_Password	Customer_Name	E-Mail_Address	Customer_Phone_number
varchar(20)	varchar(20)	varchar(20)	varchar(20)	varchar(20)

## ◆ 테이블 설계

- 운동:

운동 ID	운동 이름	개수
Exercise_ID	Exercise_Name	Exercise_Count
varchar(20)	varchar(20)	int(11)

## ◆ 테이블 설계

- 리워드:

리워드 ID	리워드 이름	보유 포인트
Reward_ID	Reward_Name	Reward_Amount
varchar(20)	varchar(20)	int(11)

## ◆ 테이블 설계

- 상품:

상품 ID	상품명	유효기간
Product_ID	Product_Name	Expiry_Date
varchar(20)	varchar(20)	date

## ◆ 테이블 생성

- 고객:

```
create table Customer(  
    Customer_ID    varchar(20) not null,  
    Customer_Password    varchar(20) not null,  
    Customer_Name    varchar(20) not null,  
    E-Mail_Address    varchar(20) not null,  
    Customer_Phone_number    varchar(20),  
    Primary Key(Customer_ID)  
);
```

## ◆ 테이블 생성

- 운동:

```
create table Exercise(  
    Exercise_ID      varchar(20) not null,  
    Exercise_Name     varchar(20) not null,  
    Points_Per_Unit   int not null,  
    Unit              varchar(20) not null,  
    Primary Key(Exercise_ID)  
);
```



## ◆ 테이블 생성

- 리워드:

```
create table Reward(  
    Reward_ID      varchar(20) not null,  
    Reward_Name     varchar(20) not null,  
    Reward_Amount   int(11),  
    Primary Key(Reward_ID)  
);
```

## ◆ 테이블 생성

- 상품:

```
create table Product(  
    Product_ID      varchar(20) not null,  
    Product_Name     varchar(20) not null,  
    Expiry_Date      date,  
    Primary Key(Product_ID)  
);
```

## ◆ 테이블 생성

- 고객 운동 기록:

```
create table Customer_Exercise(  
    Customer_Exercise_ID      int auto_increment,  
    Customer_ID               varchar(20) not null,  
    Exercise_ID               varchar(20) not null,  
    Exercise_Date              datetime not null,  
    Exercise_Count             int(11) not null,  
    Earned_Points              int not null,  
    Primary Key(Customer_Exercise_ID),  
    Foreign Key(Customer_ID) references Customer(Customer_ID),  
    Foreign Key(Exercise_ID) references Exercise(Exercise_ID)  
);
```

## ◆ 테이블 생성

- 고객 리워드 기록:

```
create table Customer_Reward(  
    Customer_Reward_ID          int auto_increment,  
    Customer_ID                 varchar(20) not null,  
    Reward_ID                   varchar(20) not null,  
    Redeem_Date                 datetime not null,  
    Primary Key(Customer_Reward_ID),  
    Foreign Key(Customer_ID) references Customer(Customer_ID),  
    Foreign Key(Reward_ID) references Reward(Reward_ID)  
);
```

## ◆ 데이터 삽입

- 고객:

```
insert into Customer values('ASZD12345', 'kasrantin234', '신동근', 'dgshin12@gmail.com', '010-4857-9344');  
insert into Customer values('FMAK12345', 'fhanz2946', '손유진', 'yjson23@gmail.com', '010-4864-4587');  
insert into Customer values('AA347563', 'ssgrn645', '김준호', 'jhkim@gmail.com', '010-6245-3360');
```

## ◆ 데이터 삽입

- 운동:

```
insert into Exercise values('ex01', '팔굽혀펴기', 1, '10개');  
insert into Exercise values('ex02', '윗몸일으키기', 1, '10개');  
insert into Exercise values('ex03', '스쿼트', 1, '20개');  
insert into Exercise values('ex04', '버피테스트', 1, '20개');  
insert into Exercise values('ex05', '플랭크자세', 1, '30초');
```

## ◆ 데이터 삽입

- 리워드:

```
insert into Reward values('rw01', '문화상품권', 1000);  
insert into Reward values('rw02', '편의점 기프티콘', 500);  
insert into Reward values('rw03', '음식점 기프티콘', 1500);
```

## ◆ 데이터 삽입

- 상품:

```
insert into Product values('pd01', '문화상품권', '2025-12-31');
```

```
insert into Product values('pd02', '편의점 기프티콘', '2024-12-31');
```

```
insert into Product values('pd03', '음식점 기프티콘', '2024-06-30');
```



## ◆ 데이터 삽입

- 고객 운동 기록:

```
insert into Customer_Exercise(Customer_ID, Exercise_ID, Exercise_Date, Exercise_Count, Earned_Points) values  
( 'ASZD12345', 'ex01', '2024-06-01 10:00:00', 10, 1),  
( 'FMAK12345', 'ex02', '2024-06-02 11:00:00', 10, 1),  
( 'AA347563', 'ex03', '2024-06-03 12:00:00', 20, 1);
```

## ◆ 데이터 삽입

- 고객 리워드 기록:

```
insert into Customer_Reward(Customer_ID, Reward_ID, Redeem_Date) values  
('ASZD12345', 'rw01', '2024-06-04 13:00:00'),  
('FMAK12345', 'rw02', '2024-06-05 14:00:00'),  
('AA347563', 'rw03', '2024-06-06 15:00:00');
```

## ◆ 데이터 삽입

- 일부 조회:

	Exercise_ID	Exercise_Name	Points_Per_Unit	Unit
▶	ex01	팔굽혀펴기	1	10개
	ex02	윗몸일으키기	1	10개
	ex03	스쿼트	1	20개
	ex04	버피테스트	1	20개
	ex05	플랭크자세	1	30초

## ◆ Q1. 모든 고객 목록 조회

- select \* from Customer;

	Customer_ID	Customer_Password	Customer_Name	E-Mail_Address	Customer_Phone_number
▶	AA347563	ssgrn645	김준호	jhkim@gmail.com	010-6245-3360
	ASZD12345	kasrantin234	신동근	dqshin12@gmail.com	010-4857-9344
	FMAK12345	fhanz2946	손유진	yjson23@gmail.com	010-4864-4587

## ◆ Q2. 특정 고객의 운동 기록 조회

- select \* from Customer\_Exercise where Customer\_ID = 'ASZD12345';

	Customer_Exercise_ID	Customer_ID	Exercise_ID	Exercise_Date	Exercise_Count	Earned_Points
▶	1	ASZD12345	ex01	2024-06-01 10:00:00	10	1

## ◆ Q3. 특정 고객의 리워드 조회

- select \* from Customer\_Reward where Customer\_ID = 'ASZD12345';

	Customer_Reward_ID	Customer_ID	Reward_ID	Redeem_Date
▶	1	ASZD12345	rw01	2024-06-04 13:00:00

## ◆ Q4. 포인트가 1000이상인 리워드 목록 조회

- select Reward\_ID, Reward\_Name, Reward\_Amount
- from Reward
- where Reward\_Amount >= 1000;

	Reward_ID	Reward_Name	Reward_Amount
▶	rw01	문화상품권	1000
	rw03	음식점 기프티콘	1500
✱	NULL	NULL	NULL

## ◆ Q5. 운동 종류별 수행 횟수 및 적립 포인트 조회

```
select e.Exercise_Name, sum(ce.Exercise_Count) as Total_Count, sum(ce.Earned_Points) as Total_Points
from Exercise e
join Customer_Exercise ce on e.Exercise_ID = ce.Exercise_ID
group by e.Exercise_Name;
```

	Exercise_Name	Total_Count	Total_Points
▶	팔굽혀펴기	10	1
	윗몸일으키기	10	1
	스쿼트	20	1



## ◆ Q6. 모든 고객의 총 적립 포인트 조회

```
select c.Customer_ID, c.Customer_Name, sum(ce.Earned_Points) as Total_Points
from Customer c
join Customer_Exercise ce on c.Customer_ID = ce.Customer_ID
group by c.Customer_ID, c.Customer_Name;
```

	Customer_ID	Customer_Name	Total_Points
▶	AA347563	김준호	1
	ASZD12345	신동근	1
	FMAK12345	손유진	1