

무인 스터디카페 관리 시스템

데이터베이스 설계 및 활용

8조

20182536 정호진

20200581 김하원

20201338 신승훈

20222833 안정인



목차

- 1 주제 선정 및 목표
- 2 데이터 모델링
- 3 데이터베이스 설계
- 4 데이터베이스 애플리케이션 개발

1 주제 선정 및 목표

주제 선정: 무인 스터디 카페

- 공유경제 산업이 무인 시스템과 결합하여 급격한 성장을 이룸
- 스터디카페는 코로나 19 발생 이후 모든 무인 스터디카페 가맹점의 매출이 280% 증가(2021.04 기준)
- 공간대여사업은 정교한 데이터베이스를 기반으로 한 CRM 시스템을 통해 인건비 절약을 꾀함

시스템 목표

- 실제 무인 스터디카페에서 사용되는 키오스크의 예약 기능 구현
- 스터디카페 회원관리 시스템 제어 기능 구현
- 좌석 뿐만 아니라 락커 및 스터디룸 예약 기능 구현

2 데이터 모델링

E-R 다이어그램 (ENTITY)

- 시스템 구현에 필요한 ENTITY (총 15개)

USER,

MEMBER, MEMBER_PRODUCT, MEMBER_PAYMENT,

NON_MEMBER, NON_MEMBER_PRODUCT, NON_MEMBER_PAYMENT,

SPECIAL_PRODUCT, SP_MEMBER_PAYMENT,

SEAT, SEAT_RES,

LOCKER, LOCKER_RES,

STUDYROOM, STUDYROOM_RES

E-R 다이어그램 (ATTRIBUTES)

ENTITY	ATTRIBUTES
USER	UserID , UserType
MEMBER	<u>UserID</u> , Name, PhoneNumber, UsageTime, Duration, Permission, <u>InviteID</u>
MEMBER _PRODUCT	ProductID , ProductTime, ProductDuration, ProductPrice
MEMBER _PAYMENT	PaymentID , <u>ProductID</u> , <u>UserID</u> , PaymentStartTime, PaymentCompleteTime, PaymentAmount

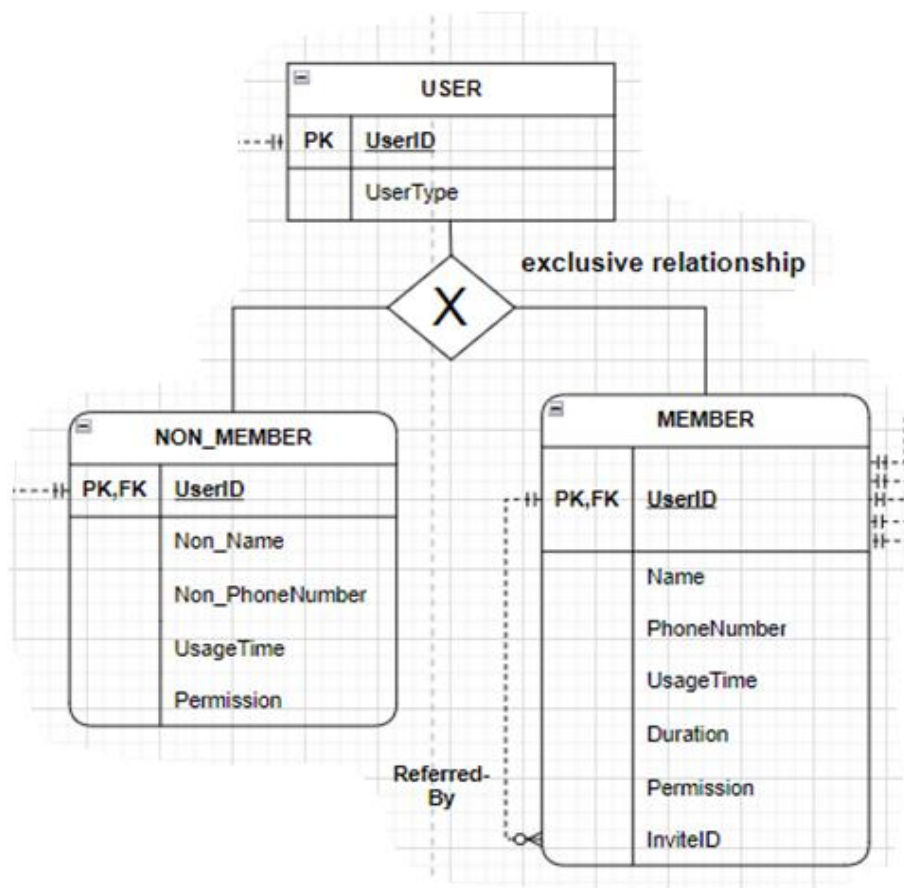
E-R 다이어그램 (ATTRIBUTES)

ENTITY	ATTRIBUTES
NON_MEMBER	<u>UserID</u> , Non_Name, Non_PhoneNumber, UsageTime, Permission
NON_MEMBER_PRODUCT	NonMemberProductID , NonMemberProductTime, NonMemberProductPrice
NON_MEMBER_PAYMENT	NonMemberPaymentID , <u>NonMemberProductID</u> , <u>UserID</u> , NonMemberPaymentStartTime, NonMemberPaymentCompleteTime, NonMemberPaymentAmount
SPECIAL_PRODUCT	SP_ProductID , SP_ProductTime, SP_ProductPrice
SP_MEMBER_PAYMENT	SP_PaymentID , <u>SP_ProductID</u> , <u>UserID</u> , SP_PaymentStartTime, SP_PaymentCompleteTime, SP_PaymentAmount, <u>InviteID</u>

E-R 다이어그램 (ATTRIBUTES)

ENTITY	ATTRIBUTES
SEAT	SeatID , FloorID, IsAvailable
SEAT_RES	ReservationID , <u>SeatID</u> , <u>UserID</u> , ReservationStartTime, CheckoutTime
LOCKER	LockerID , FloorID, IsAvailable
LOCKER_RES	ReservationID , <u>LockerID</u> , <u>UserID</u> , StartTime, EndTime
STUDYROOM	RoomID , FloorID, MaxCapacity, IsAvailable
STUDYROOM_RES	ReservationID , <u>RoomID</u> , <u>UserID</u> , StartTime, EndTime

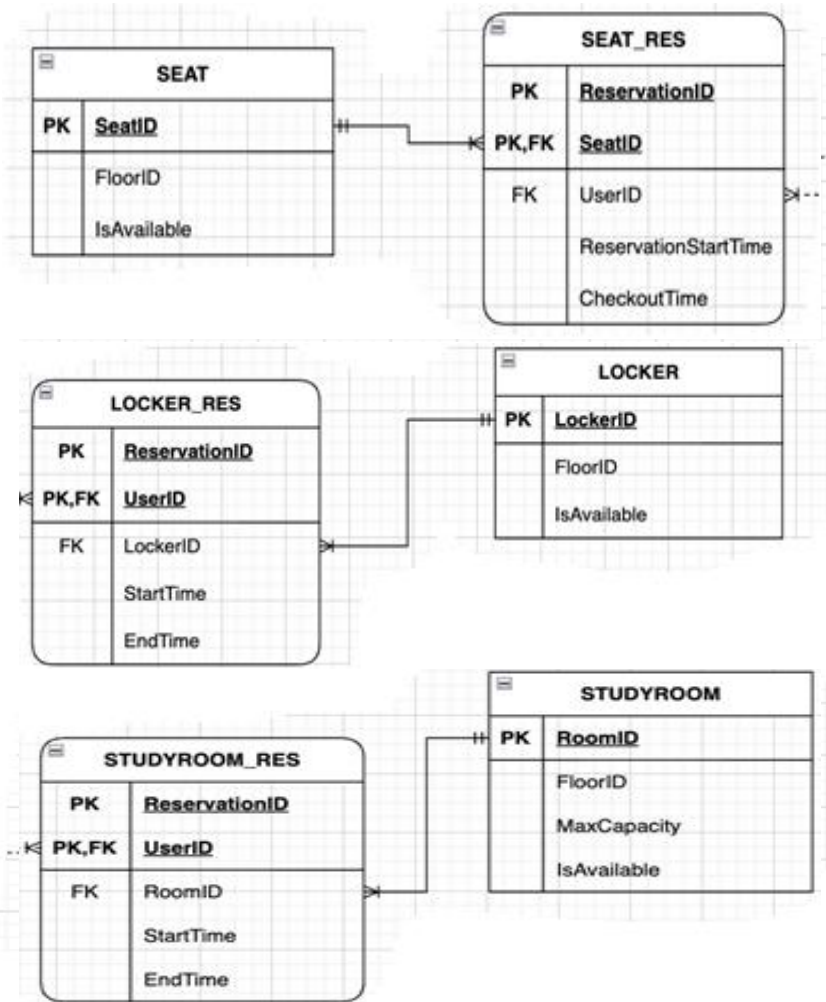
E-R 다이어그램



USER&NON_MEMBER, MEMBER

- NON_MEMBER와 MEMBER는 USER의 SUBTYPE ENTITY, 각 항목들은 UserID 중 회원 혹은 비회원 중 한 TYPE에만 속함.
- NON_MEMBER와 MEMBER의 UserID는 USER의 PK인 UserID의 FK.
- MEMBER의 InviteID는 추천인으로, UserID와 InviteID는 1:多 Recursive Relationship을 지님.

E-R 다이어그램

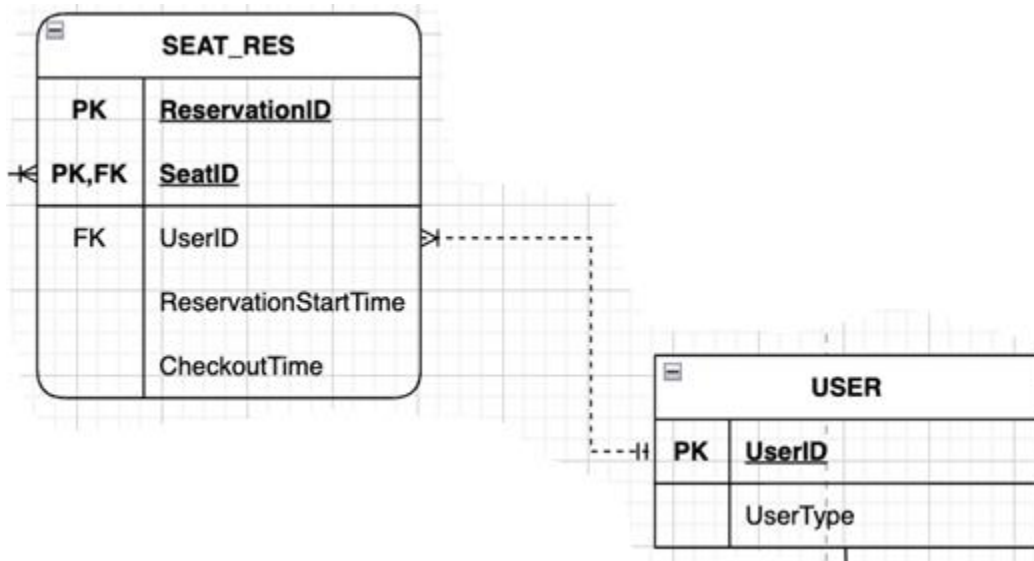


SEAT&SEAT_RES, LOCKER&LOCKER_RES, STUDYROOM&STUDYROOM_RES

- SEAT_RES는 ReservationID과 SeatID를 Composite PK로 가지며, 좌석의 ID를 알기 위한 SeatID는 SEAT의 PK인 SeatID의 FK.
- SEAT의 한 좌석이 SEAT_RES의 여러 좌석과 연결이 가능하므로 1:多 관계를 지니며, SEAT_RES는 SEAT의 WEAK ENTITY.
- LOCKER와 LOCKER_RES도 동일
- STUDYROOM과 STUDYROOM_RES도 동일.

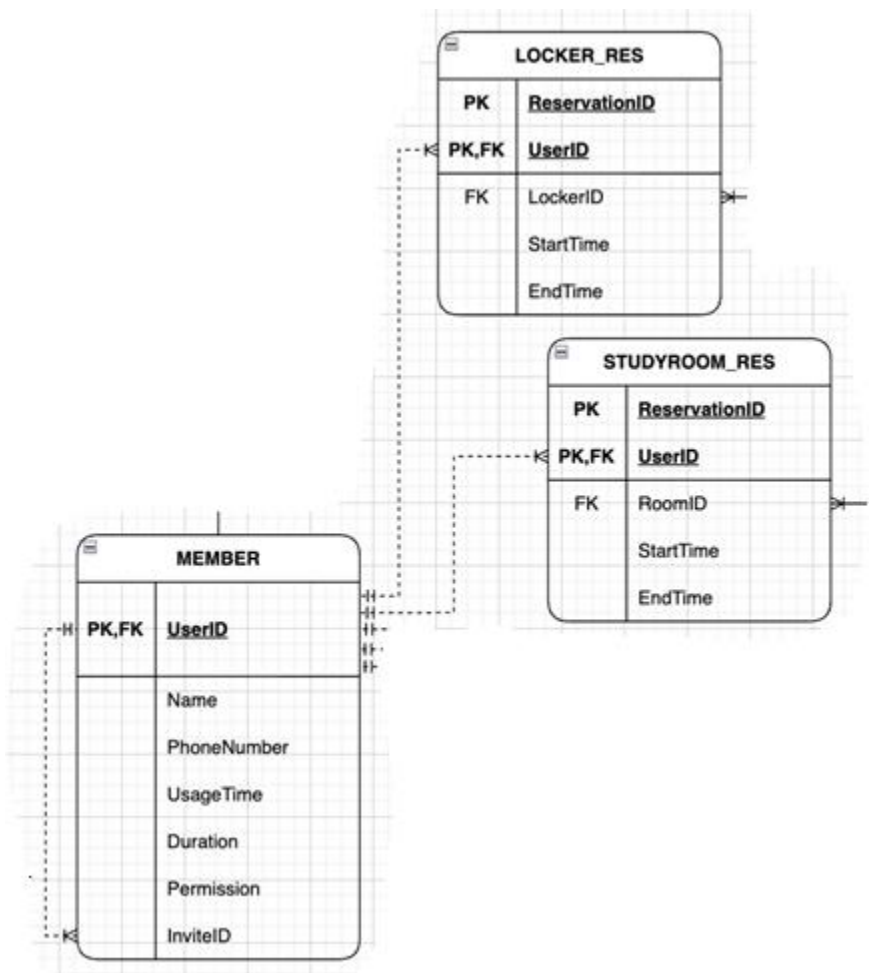
E-R 다이어그램

SEAT_RES&USER



- 좌석 배정은 회원과 비회원 모두에게 가능하도록 하여 USER 테이블과 연결.
- SEAT_RES에서 사용자의 ID를 알기 위한 UserID는 USER의 PK인 UserID의 FK.
- USER의 사용자들은 SEAT_RES의 여러 UserID와 연결이 가능하므로 1:多 관계를 지님.

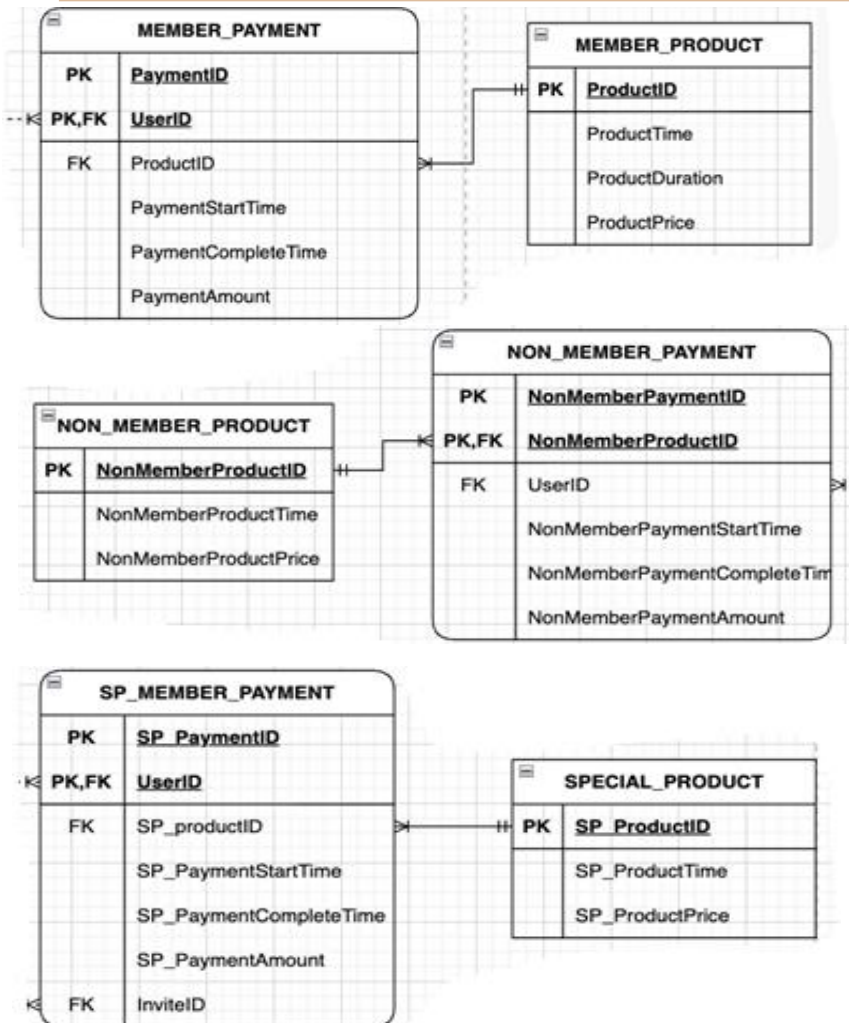
E-R 다이어그램



LOCKER_RES, STUDYROOM_RES & MEMBER

- 락커 및 스터디룸 배정은 회원에게만 가능하도록 하여 MEMBER 테이블과 연결.
- LOCKER_RES과 STUDYROOM_RES에서 사용자의 ID를 알기 위한 UserID는 MEMBER의 PK인 UserID의 FK.
- USER의 사용자들은 LOCKER_RES과 STUDYROOM_RES의 여러 UserID와 연결이 가능하므로 1:多 관계를 지님.

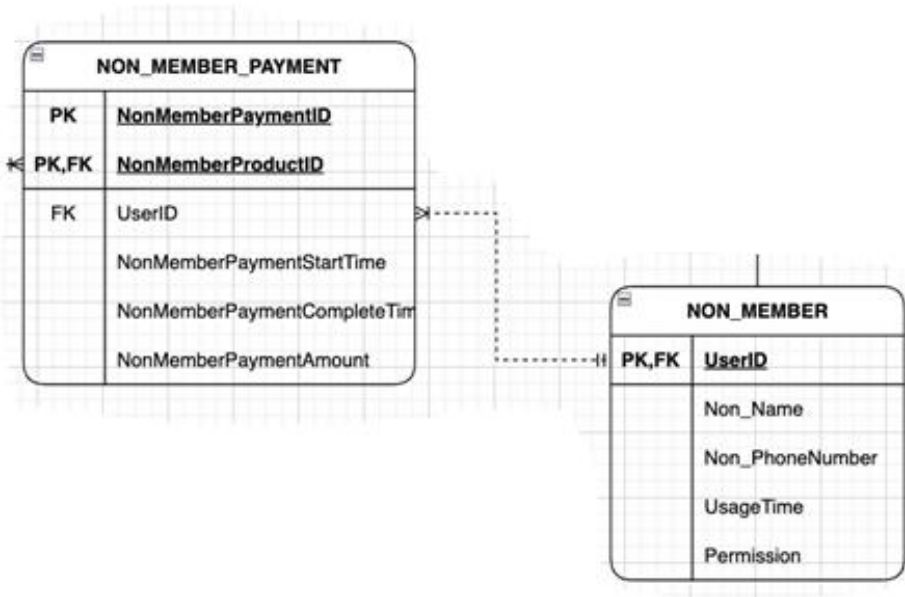
E-R 다이어그램



MEMBER_PRODUCT&MEMBER_PAYMENT, NON_MEMBER_PRODUCT&NON_MEMBER_PAYMENT, SPECIAL_PRODUCT&SP_MEMBER_PAYMENT

- M_PAYMENT는 PaymentID와 UserID를 PK로 가지며, 상품의 ID를 알기 위한 ProductID는 MEMBER_PRODUCT의 PK인 ProductID의 FK.
- MEMBER_PRODUCT의 한 상품ID가 MEMBER_PAYMENT의 여러 결제와 연결이 가능하므로 1:多 관계를 지님.
- NON_MEMBER_PAYMENT와 NON_MEMBER_PRODUCT도 동일
- SP_MEMBER_PAYMENT와 SPECIAL_PRODUCT도 동일.

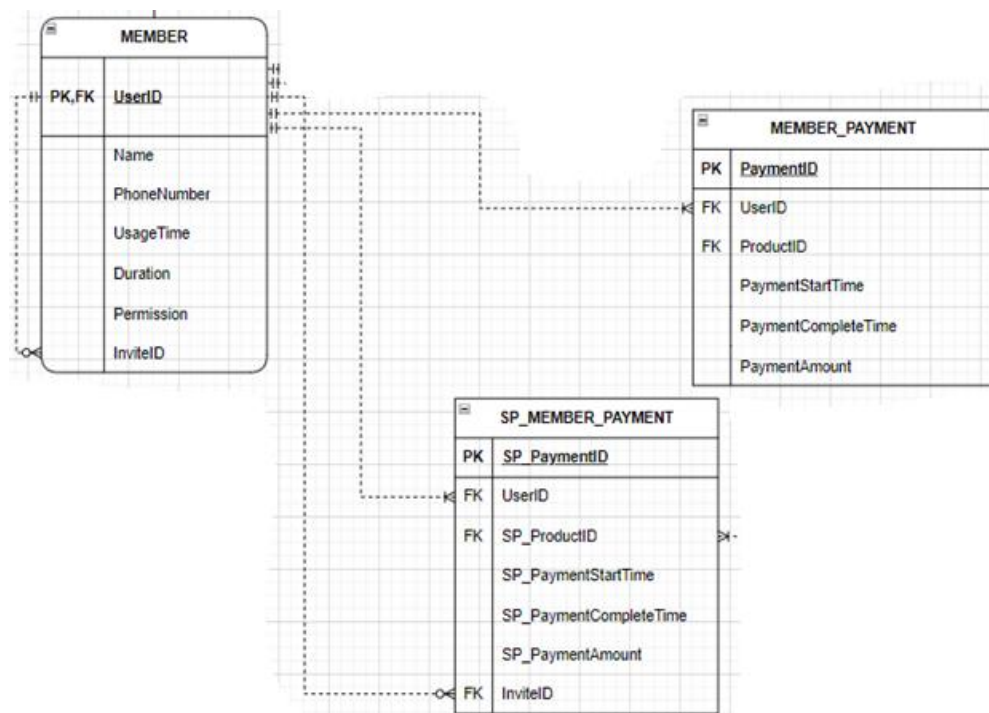
E-R 다이어그램



NON_MEMBER_PAYMENT & NON_MEMBER

- 비회원 결제는 비회원에게만 가능하도록 하여 NON_MEMBER 테이블과 연결.
- NON_M_PAYMENT에서 사용자의 ID를 알기 위한 UserID는 NON_MEMBER의 PK인 UserID의 FK.
- NON_MEMBER의 사용자들은 NON_M_PAYMENT의 여러 UserID와 연결이 가능하므로 1:多 관계를 지님.

E-R 다이어그램



MEMBER_PAYMENT, SP_MEMBER_PAYMENT&NON_MEMBER

- 회원 결제와 추천인 결제는 회원에게만 가능하도록 하여 MEMBER 테이블과 연결.
- MEMBER_PAYMENT 및 SP_MEMBER_PAYMENT에서 사용자의 ID를 알기 위한 UserID는 MEMBER의 PK인 UserID의 FK.
- 이때 SP_MEMBER_PAYMENT는 누가 추천을 해줬는지 알기 위해 InviteID가 존재하고, 이는 MEMBER의 InviteID의 FK.
- MEMBER의 사용자들은 MEMBER_PRODUCT와 SP_MEMBER_PAYMENT의 여러 UserID와 연결이 가능하므로 1:多 관계를 지님.

3 데이터베이스 설계

CREATE TABLE : USER

```
• CREATE TABLE USER (  
    UserID INT PRIMARY KEY,  
    UserType ENUM('Member', 'NonMember') NOT NULL  
);
```

USER

- UserID는 PK.
- UserType의 ENUM 데이터 타입은 'Member'와 'NonMember'의 입력만을 허용.

CREATE TABLE : NON_MEMBER

```
• CREATE TABLE NON_MEMBER (  
    UserID INT PRIMARY KEY,  
    Non_Name_ VARCHAR(20) NOT NULL,  
    Non_PhoneNumber VARCHAR(15) NOT NULL,  
    UsageTIME TIME,  
    Permission BOOLEAN DEFAULT FALSE,  
    FOREIGN KEY (UserID) REFERENCES USER(UserID)  
);
```

NON_MEMBER

- NON_MEMBER 테이블의 UserID는 USER 테이블의 UserID를 참조.
- UsageTIME은 사용자가 스터디 카페 사용 이전일 수 있으므로 NULL값을 허용.
- Permission은 기본값이 FALSE로 설정되어 있다. 이 열은 회원의 권한 여부를 나타내며, TRUE랑 FALSE의 값을 가질 수 있고, TRUE는 회원 권한이 있는 것을 의미.

CREATE TABLE : MEMBER

```
• CREATE TABLE MEMBER (  
    UserID INT PRIMARY KEY,  
    Name_ VARCHAR(20) NOT NULL,  
    PhoneNumber VARCHAR(20) NOT NULL,  
    UsageTIME TIME,  
    Duration DATETIME,  
    Permission BOOLEAN DEFAULT FALSE,  
    InviteID INT,  
    FOREIGN KEY (UserID) REFERENCES USER(UserID),  
    FOREIGN KEY (InviteID) REFERENCES MEMBER(UserID)  
);
```

MEMBER

- MEMBER 테이블의 UserID는 USER 테이블의 UserID를 참조
- InviteID는 추천인 코드로 Recursive Relationship을 의미하고, UserID를 참조.
- UsageTIME과 Duration은 사용자가 스터디 카페 사용 이전 일 수 있으므로 NULL값을 허용.
- Permission은 기본값이 FALSE로 설정되어 있다. 이 열은 회원의 권한 여부를 나타내며, TRUE랑 FALSE의 값을 가질 수 있고, TRUE는 회원 권한이 있는 것을 의미.

CREATE TABLE : SEAT

```
• CREATE TABLE SEAT (  
    SeatID INT PRIMARY KEY,  
    FloorID INT NOT NULL,  
    IsAvailable BOOLEAN DEFAULT TRUE  
);
```

SEAT

- SeatID는 PK.
- IsAvailable은 기본값이 TRUE로 설정되어 있다. 이 열은 회원의 이용 가능 여부를 나타내며, TRUE 또는 FALSE 값을 가질 수 있고, TRUE는 회원이 이용이 가능한 것을 의미.

CREATE TABLE : LOCKER

```
• CREATE TABLE LOCKER (  
    LockerID INT PRIMARY KEY,  
    FloorID INT NOT NULL,  
    IsAvailable BOOLEAN DEFAULT TRUE  
);
```

LOCKER

- LockerID는 PK.
- IsAvailable은 기본값이 TRUE로 설정되어 있다. 이 열은 회원의 이용 가능 여부를 나타내며, TRUE 또는 FALSE 값을 가질 수 있고, TRUE는 회원이 이용이 가능한 것을 의미.

CREATE TABLE : STUDYROOM

```
• CREATE TABLE STUDYROOM (  
    RoomID INT PRIMARY KEY,  
    FloorID INT NOT NULL,  
    MaxCapacity INT NOT NULL,  
    IsAvailable BOOLEAN DEFAULT TRUE  
);
```

STUDYROOM

- RoomID는 PK.
- IsAvailable은 기본값이 TRUE로 설정되어 있다. 이 열은 회원의 이용 가능 여부를 나타내며, TRUE 또는 FALSE 값을 가질 수 있고, TRUE는 회원이 이용이 가능한 것을 의미.

CREATE TABLE : SEAT_RES

```
• CREATE TABLE SEAT_RES (  
    ReservationID INT AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    SeatID INT NOT NULL,  
    ReservationStartTime DATETIME,  
    CheckoutTime DATETIME,  
    PRIMARY KEY (ReservationID, SeatID),  
    FOREIGN KEY (UserID) REFERENCES USER(UserID),  
    FOREIGN KEY (SeatID) REFERENCES SEAT(SeatID)  
);
```

SEAT_RES

- ReservationID와 SeatID는 PK.
- ReservationID는 AUTO_INCREMENT로 일련번호가 부여.
- SEAT_RES 테이블의 UserID는 USER 테이블의 UserID를 참조,
- SEAT_RES 테이블의 SeatID는 SEAT 테이블의 SeatID를 참조.
- ReservationStartTime과 CheckoutTime은 사용자의 이용 조건에 따라 달라지므로 NULL값을 허용.

CREATE TABLE : STUDYROOM_RES

```
• CREATE TABLE STUDYROOM_RES (  
    ReservationID INT AUTO_INCREMENT,  
    RoomID INT NOT NULL,  
    UserID INT NOT NULL,  
    StartTime DATETIME,  
    EndTime DATETIME,  
    PRIMARY KEY (ReservationID, RoomID),  
    FOREIGN KEY (RoomID) REFERENCES STUDYROOM(RoomID),  
    FOREIGN KEY (UserID) REFERENCES MEMBER(UserID)  
);
```

STUDYROOM_RES

- ReservationID와 RoomID는 PK.
- ReservationID는 AUTO_INCREMENT로 일련번호가 부여.
- STUDYROOM_RES 테이블의 UserID는 USER 테이블의 UserID를 참조.
- STUDYROOM_RES 테이블의 RoomID는 SEAT 테이블의 RoomID를 참조.
- StartTime과 EndTime은 사용자의 이용 조건에 따라 달라지므로, NULL값을 허용.

CREATE TABLE : LOCKER_RES

```
• CREATE TABLE LOCKER_RES (  
    ReservationID INT AUTO_INCREMENT,  
    LockerID INT NOT NULL,  
    UserID INT NOT NULL,  
    StartTime DATETIME,  
    EndTime DATETIME,  
    PRIMARY KEY (ReservationID, LockerID),  
    FOREIGN KEY (LockerID) REFERENCES LOCKER(LockerID),  
    FOREIGN KEY (UserID) REFERENCES MEMBER(UserID)  
);
```

LOCKER_RES

- ReservationID와 LockerID는 PK.
- ReservationID는 AUTO_INCREMENT로 일련번호가 부여.
- LOCKER_RES 테이블의 UserID는 USER 테이블의 UserID를 참조.
- LOCKER_RES 테이블의 LockerID는 SEAT 테이블의 LockerID를 참조.
- StartTime과 EndTime은 사용자의 이용 조건에 따라 달라지므로, NULL값을 허용.

CREATE TABLE : SPECIAL_PRODUCT

```
• CREATE TABLE SPECIAL_PRODUCT (  
    SP_ProductID INT PRIMARY KEY,  
    SP_ProductTime TIME NOT NULL,  
    SP_ProductPrice DECIMAL(10, 2) NOT NULL  
);
```

SPECIAL_PRODUCT

- SP_ProductID는 PK.
- SP_ProductTime은 상품 별 지속 기간을 의미.
- SPECIAL_PRODUCT는 일일권만 취급하므로 Duration이 부재.
- SP_ProductPrice의 DECIMAL(10,2)는 소수점이 있는 10자리의 숫자를 나타내며, 소수점 이하 2자리까지 저장 가능.

CREATE TABLE : NON_MEMBER_PRODUCT

```
• CREATE TABLE NON_MEMBER_PRODUCT (  
    NonMemberProductID INT PRIMARY KEY,  
    NonMemberProductTime TIME NOT NULL,  
    NonMemberProductPrice DECIMAL(10, 2) NOT NULL  
);
```

NON_MEMBER_PRODUCT

- NonMemberProductID는 PK.
- NonMemberProductTime은 상품 별 지속 기간을 의미.
- NON_MEMBER_PRODUCT는 일일권만 취급하므로 Duration 이 부재.
- NonMemberProductPrice의 DECIMAL(10,2)는 소수점이 있는 10자리의 숫자를 나타내며, 소수점 이하 2자리까지 저장.

CREATE TABLE : MEMBER_PRODUCT

```
• CREATE TABLE MEMBER_PRODUCT (  
    ProductID INT PRIMARY KEY,  
    ProductTime TIME NOT NULL,  
    ProductDuration DATETIME,  
    ProductPrice DECIMAL(10, 2) NOT NULL  
);
```

MEMBER_PRODUCT

- ProductID는 PK.
- SP_ProductTime은 상품 별 지속 기간을 의미.
- MEMBER_PRODUCT는 기간권도 포함하므로 Duration 열이 존재.
- ProductPrice의 DECIMAL(10,2)는 소수점이 있는 10자리의 숫자를 나타내며, 소수점 이하 2자리까지 저장.

CREATE TABLE : MEMBER_PAYMENT

```
• CREATE TABLE MEMBER_PAYMENT (  
    PaymentID INT AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    ProductID INT NOT NULL,  
    PaymentStartTime DATETIME,  
    PaymentCompleteTime DATETIME,  
    PaymentAmount DECIMAL(10, 2),  
    PRIMARY KEY (PaymentID, ProductID),  
    FOREIGN KEY (UserID) REFERENCES MEMBER(UserID),  
    FOREIGN KEY (ProductID) REFERENCES MEMBER_PRODUCT(ProductID)  
);
```

MEMBER_PAYMENT

- PaymentID와 ProductID는 PK.
- PaymentID는 AUTO_INCREMENT로 일련번호가 부여.
- MEMBER_PAYMENT테이블의 UserID는 MEMBER 테이블의 UserID를 참조.
- MEMBER_PAYMENT 테이블의 ProductID는 MEMBER_PRODUCT 테이블의 ProductID를 참조.
- PaymentStartTime, PaymentEndTime, PaymentAmount는 사용자의 이용 조건에 따라 달라지므로, NULL값을 허용.
- PaymentAmount의 DECIMAL(10,2)는 소수점이 있는 10자리의 숫자를 나타내며, 소수점 이하 2자리까지 저장.

CREATE TABLE : NON_MEMBER_PAYMENT

```
• CREATE TABLE NON_MEMBER_PAYMENT (  
    NonMemberPaymentID INT AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    NonMemberProductID INT NOT NULL,  
    NonMemberPaymentStartTime DATETIME,  
    NonMemberPaymentCompleteTime DATETIME,  
    NonMemberPaymentAmount DECIMAL(10, 2),  
    PRIMARY KEY (NonMemberPaymentID, NonMemberProductID),  
    FOREIGN KEY (UserID) REFERENCES NON_MEMBER(UserID),  
    FOREIGN KEY (NonMemberProductID) REFERENCES NON_MEMBER_PRODUCT(NonMemberProductID)  
);
```

NON_MEMBER_PAYMENT

- NonMemberPaymentID와 NonMemberProductID는 PK.
- NonMemberPaymentID는 AUTO_INCREMENT로 일련번호가 부여.
- NON_MEMBER_PAYMENT테이블의 UserID는 NON_MEMBER 테이블의 UserID를 참조.
- NONMEMBER_PAYMENT 테이블의 NonMemberProductID는 NONMEMBER_PRODUCT 테이블의 NonMemberProductID를 참조.
- NonMemberPaymentStartTime, NonMemberPaymentEndTime, NonMemberPaymentAmount 은 사용자의 이용 조건에 따라 달라지므로, NULL값을 허용.
- NonMemberPaymentAmount의 DECIMAL(10,2)는 소수점이 있는 10자리의 숫자를 나타내며, 소수점 이하 2자리까지 저장.

CREATE TABLE : SP_MEMBER_PAYMENT

```
• CREATE TABLE SP_MEMBER_PAYMENT (  
    SP_PaymentID INT AUTO_INCREMENT,  
    UserID INT NOT NULL,  
    SP_ProductID INT NOT NULL,  
    SP_PaymentStartTime DATETIME,  
    SP_PaymentCompleteTime DATETIME,  
    SP_PaymentAmount DECIMAL(10, 2),  
    InviteID INT NOT NULL,  
    PRIMARY KEY (SP_PaymentID, SP_ProductID),  
    FOREIGN KEY (UserID) REFERENCES MEMBER(UserID),  
    FOREIGN KEY (InviteID) REFERENCES MEMBER(UserID),  
    FOREIGN KEY (SP_ProductID) REFERENCES SPECIAL_PRODUCT(SP_ProductID)  
);
```

SP_MEMBER_PAYMENT

- SP_PaymentID와 SP_ProductID는 PK.
- SP_PaymentID는 AUTO_INCREMENT로 일련번호가 부여.
- SP_MEMBER_PAYMENT 테이블의 UserID는 MEMBER 테이블의 UserID를 참조.
- SP_MEMBER_PAYMENT 테이블의 InviteID는 MEMBER 테이블의 UserID를 참조.
- SP_MEMBER_PAYMENT 테이블의 SP_ProductID는 SPECIAL_PRODUCT 테이블의 SP_ProductID를 참조.
- SP_PaymentStartTime, SP_PaymentCompleteTime, SP_PaymentAmount 은 사용자의 이용 여부에 따라 NULL값을 허용한다.
- SP_PaymentAmount의 DECIMAL(10,2)는 소수점이 있는 10 자리의 숫자를 나타내며, 소수점 이하 2자리까지 저장할 수 있다.

INSERT Data : MEMBER 데이터 예시

```
1 • INSERT INTO MEMBER VALUES (1, '박서준', '01048094818', '02:00:00', '2023-11-23', TRUE, NULL);
2 • INSERT INTO MEMBER VALUES (2, '박원빈', '01034384448', '02:00:00', '2023-11-23', TRUE, NULL);
3 • INSERT INTO MEMBER VALUES (3, '신민아', '01048324845', '06:00:00', '2023-11-23', TRUE, NULL);
4 • INSERT INTO MEMBER VALUES (4, '김지연', '01040904858', '06:00:00', '2023-11-23', TRUE, NULL);
5 • INSERT INTO MEMBER VALUES (5, '이서진', '01034884858', '08:00:00', '2023-11-23', TRUE, NULL);
6 • INSERT INTO MEMBER VALUES (6, '정유미', '01048334558', '08:00:00', '2023-11-24', TRUE, NULL);
7 • INSERT INTO MEMBER VALUES (7, '최우식', '01068382758', '08:00:00', '2023-11-24', TRUE, NULL);
8 • INSERT INTO MEMBER VALUES (8, '김혁수', '01038384086', '10:00:00', '2023-11-24', TRUE, NULL);
9 • INSERT INTO MEMBER VALUES (9, '주현영', '01088351458', '10:00:00', '2023-11-24', TRUE, NULL);
10 • INSERT INTO MEMBER VALUES (10, '이승민', '01018384458', '10:00:00', '2023-11-24', TRUE, NULL);
11
12 • UPDATE MEMBER
13     SET InviteID = 2
14     WHERE UserID = 1; #1번이 2번을 추천
15 • UPDATE MEMBER
16     SET InviteID = 3
17     WHERE UserID = 2; #2번이 3번을 추천
18 • UPDATE MEMBER
19     SET InviteID = 6
20     WHERE UserID = 3; #3번이 6번을 추천
```

- Create Table 구문에서 작성한 조건문을 기반으로 데이터를 삽입.

ALTER TABLE

```
ALTER TABLE MEMBER  
MODIFY COLUMN InviteID INT NULL;
```

```
ALTER TABLE SP_MEMBER_PAYMENT  
MODIFY COLUMN InviteID INT NULL;
```

ALTER MEMBER, SP_MEMBER_PAYMENT

- InviteID를 컬럼으로 갖고 있는테이블 MEMBER, SP_MEMBER_PAYMENT는 InviteID가 항상 존재할 필요가 없으므로 NULL RKQT을 허용하는 ALTER 구문을 사용해서 변경해줌.

회원이 상품 결제 시 데이터가 UPDATE 될 과정

MEMBER DURATION 첫 갱신 시나리오

```
INSERT INTO MEMBER (UserID, Name_, PhoneNumber, UsageTIME, Duration, Permission, InviteID)
VALUES (1, '호진', '010-5446-2693', NULL, NULL, FALSE, 22);
```

PAYMENT LOG 발생

```
INSERT INTO SP_MEMBER_PAYMENT (UserID, SP_ProductID, SP_PaymentStartTime, SP_PaymentCompleteTime, SP_PaymentAmount, InviteID)
VALUES (1, 1, '2023-11-24 12:00:00', '2023-11-24 12:00:03', 7000, 22);
```

MEMBER 테이블 갱신

```
UPDATE MEMBER
SET UsageTIME = '06:00:00',
    Permission = TRUE,
    Duration = (SELECT SP_PaymentCompleteTime + INTERVAL 30 DAY
                FROM SP_MEMBER_PAYMENT
                WHERE SP_PaymentID = 1)
WHERE UserID = 1;
```

- 1) 회원이 회원가입 시, 자신의 정보를 기입하여, MEMBER의 일부 데이터가 INSERT되며 갱신
- 2) 해당 회원이 상품을 결제하여 ,PAYMENT의 데이터가 INSERT
- 3) 회원이 상품 결제를 하여, MEMBER의 데이터가 수정

회원이 상품 결제 시 데이터가 UPDATE 될 과정

기존 MEMBER 테이블 레코드 수정 시나리오

```
INSERT INTO MEMBER (UserID, Name_, PhoneNumber, UsageTIME, Duration, Permission, InviteID)
VALUES (1, '효진', '010-5446-2693', '06:00:00', '2023-12-24 12:00:03', TRUE, 22);
```

'2023-12-24 12:00:03'이 됐음

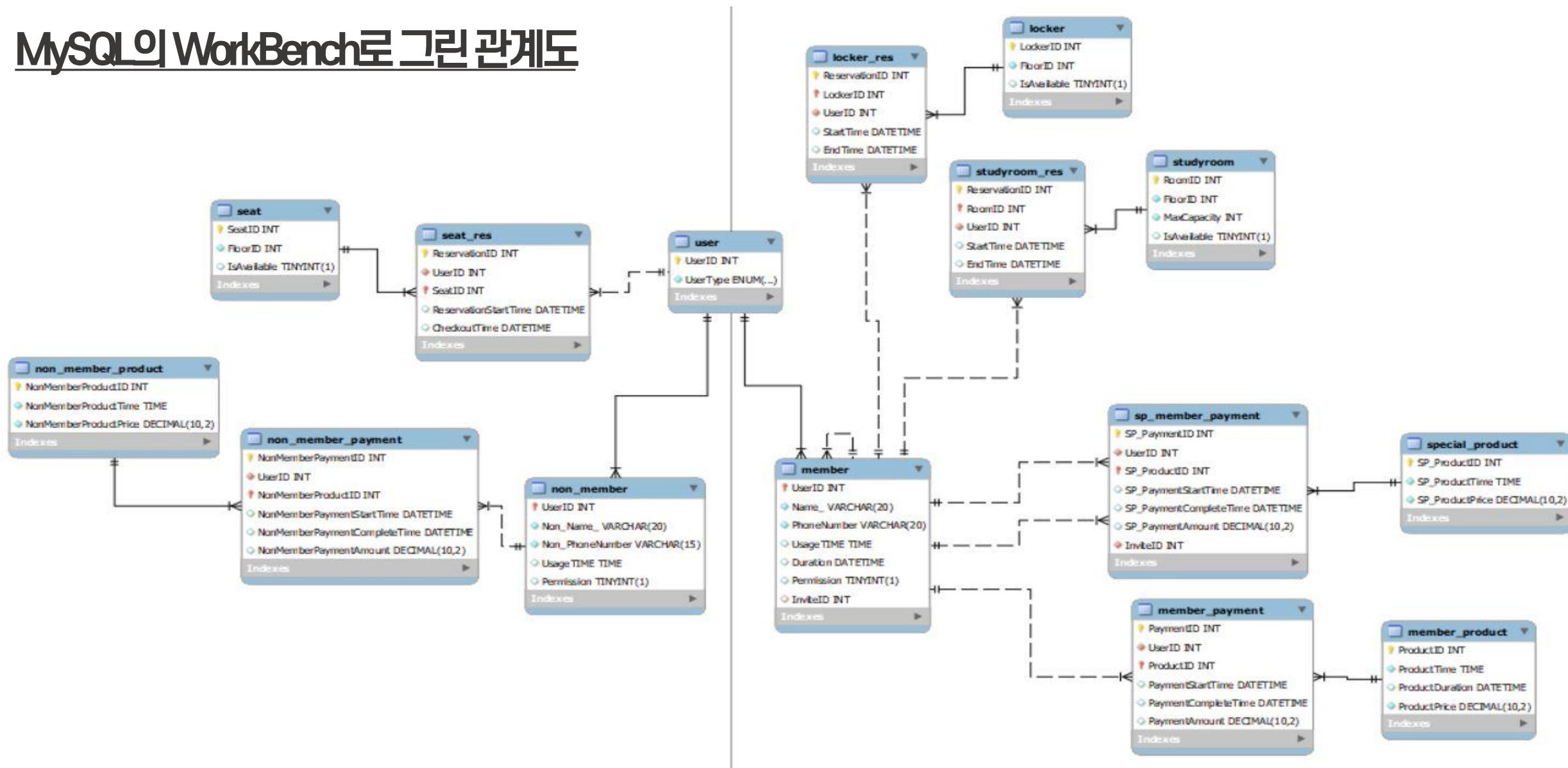
```
UPDATE MEMBER
```

```
SET UsageTIME = NULL,
    Duration = NULL,
    Permission = FALSE
```

```
WHERE UserID = 1 AND NOW() > '2023-12-24 12:00:03';
```

- 4) MEMBER의 데이터가 수정되어 INSERT
- 5) 회원이 결제한 상품의 기간이 다 되었을 때, 회원 정보가 수정

MySQL의 WorkBench로 그린 관계도



관리자 기능 설정

[관리자 기능 기본 설정]

- 1) 관리자용 계정 5개 : Master, Kim, Shin, Jeong, An
- 2) MySQL 특성 상, 관리자 기능 설정을 쿼리문으로 작성
- 3) 함께 관리할 DB를 'STUDYCAFE'로 설정
- 4) 각 관리자 별로 접근이 가능한 테이블을 다르게 설정

관리자 기능 설정

```
create user 'Master'@'localhost' identified by '0000';  
grant all on studycafe.* to Master@localhost;
```

```
create user 'Kim'@'localhost' identified by '0001';  
grant all on studycafe.member_product to Kim@localhost;  
grant all on studycafe.non_member_product to Kim@localhost;  
grant all on studycafe.sp_member_product to Kim@localhost;
```

```
create user 'Shin'@'localhost' identified by '0002';  
grant all on studycafe.seat_res to Shin@localhost;
```

```
create user 'Jeong'@'localhost' identified by '0003';  
grant all on studycafe.studyroom_res to Jeong@localhost;
```

```
create user 'An'@'localhost' identified by '0004';  
grant all on studycafe.locker_res to An@localhost;
```

Master: studycafe DB에 대한 모든 테이블의 접근과 모든 권한이 부여됨

Kim: studycafe DB에 대한 PRODUCT 테이블의 접근만 가능하고, 그 테이블에 대한 모든 권한이 부여됨

Shin: studycafe DB에 대한 SEAT_RES 테이블의 접근만 가능하고, 그 테이블에 대한 모든 권한이 부여됨

Jeong: studycafe DB에 대한 STUDY_RES 테이블의 접근만 가능하고, 그 테이블에 대한 모든 권한이 부여됨

An: studycafe DB에 대한 LOCKER_RES 테이블의 접근만 가능하고, 그 테이블에 대한 모든 권한이 부여됨

4 데이터베이스 어플리케이션 개발

메인화면 HTML 코드 실행 결과

스터디카페 조회 시스템

승실대학교 산업정보시스템공학과 데이터베이스 설계 및 활용 8조

정호진 김하원 신승훈 안정인

Welcome! SS Study Cafe Page

스터디카페 이용자 조회 ASP :

[Display the Member Table](#)

[Display the NonMember Table](#)

스터디카페 이용 중인 좌석 조회 ASP : [Display the Seat Table](#)

Member ASP 코드(회원 테이블 생성 코드)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>User List Display ASP Page</title>
    <style type="text/css">
      h1 {text-align: center; color: blue}
      h2 {font-family: Ariel, sans-serif; text-align: left; color: blue}
      p.footer {text-align: center}
    </style>
  </head>
  <body>
    <h1>The Member Table</h1>
    <hr />
    <h2>Member List</h2>

    <% Session.CodePage = 949
       Response.CharSet = "euc-kr"
       Response.AddHeader "Pragma", "no-cache"
       Response.AddHeader "cache-control", "no-staff"
       Response.Expires = -1
    %>

    <%
      Dim dbConCmd, SQLQuery, dbCon, rs, rsList
      dbConCmd= "DSN=root; Database=sr; UID=root; PWD=kxr8018@"
      SQLQuery = "SELECT Name_ , PhoneNumber FROM Member"

      Set dbCon = Server.CreateObject("ADODB.Connection")
      Set rs = Server.CreateObject("ADODB.RecordSet")

      dbCon.Open dbConCmd
      rs.Open SQLQuery, dbCon

    %>

    <table>
      <tr>
        <th>Name</th>
        <th>PhoneNumber</th>
      </tr>
    <%
      IF (rs.EOF AND rs.EOF) THEN
        Response.Write("유저정보가 존재하지 않습니다.")
      ELSE
        rsList= rs.GetRows()
        FOR x=0 TO UBound(rsList, 2)
          Response.Write("<tr>")
          FOR y=0 TO Ubound(rsList, 1)
            Response.Write("<td>")
            Response.Write(rsList(y, x))
            Response.Write("</td>")
          NEXT
          Response.Write("</tr>")
        NEXT
        Response.Write("</table>")
      END IF
    %>

    <br/>
    <hr/>
    <p class="footer">
      <a href=" ../wwwroot/study1.html">Return to Study Cafe Home Page</a>
    </p>
  </body>
</html>
```

1) 스터디 카페 회원 정보를 보여주는 테이블 생성 ASP 코드.

2) ODBC로 MYSQL에 있는 데이터베이스와의 연동을 통해 데이터를 가져옴.

3) 회원의 이름과 전화번호를 나타내는 테이블로 구성.

(MYSQL 서버 데이터베이스에 있는 Member 테이블에 있는 Name_, PhoneNumber 2개의 컬럼에서 데이터를 가져옴)

Member ASP 코드 실행 결과 (회원 테이블 생성)

The Member Table

Member List

Name	PhoneNumber
박서준	01048094818
박원빈	01034384448
신민아	01048324845
김지연	01040904858
이서진	01034884858
정유미	01048334558
최우식	01068382758
김혁수	01038384086
주현영	01088351458
이승민	01018384458

[Return to Study Cafe Home Page](#)

NonMember ASP 코드 (비회원 테이블 생성 코드)

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Member List Display ASP Page</title>
    <style type="text/css">
      h1 {text-align: center; color: blue}
      h2 {font-family: Arial, sans-serif; text-align: left; color: blue}
      p.footer {text-align: center}
    </style>
  </head>
  <body>
    <h1>The NonMember Table</h1>
    <hr />
    <h2>NonMember List</h2>

    <%@ CODEPAGE="65001" LANGUAGE="VBSCRIPT" %>

    <%
      Session.CodePage = 949
      Response.CharSet = "euc-kr"
      Response.AddHeader "Pragma","no-cache"
      Response.AddHeader "cache-control", "no-staff"
      Response.Expires = -1

      Dim dbConCmd, SQLQuery, dbCon, rs, rsList
      dbConCmd= "DSN=root; Database=sr; UID=root; PWD=krx8018@";
      SQLQuery = "SELECT Non_Name_, Non_PhoneNumber FROM Non_Member"

      Set dbCon = Server.CreateObject("ADODB.Connection")
      Set rs = Server.CreateObject("ADODB.RecordSet")

      dbCon.Open dbConCmd
      rs.Open SQLQuery, dbCon

      <table>
        <tr>
          <th>Name</th>
          <th>PhoneNumber</th>
        </tr>
      <%
        IF (rs.BOF AND rs.EOF) THEN
          Response.Write("유저정보가 존재하지 않습니다.")
        ELSE
          rsList= rs.GetRows()
          FOR x=0 TO UBound(rsList, 2)
            Response.Write("<tr>")
            FOR y=0 TO Ubound(rsList, 1)
              Response.Write("<td>")
              Response.Write(rsList(y, x))
              Response.Write("</td>")
            NEXT
            Response.Write("</tr>")
          NEXT
          Response.Write("</table>")
        END IF
      <%~
    </br>
    </hr>
    <p class="footer">
      <a href="http://wwwroot/study1.html">Return to Heather Sweeney Designs Home Page</a>
    </p>
    </hr>
  </body>
</html>
```

- 1) 스터디 카페 비회원 정보를 보여주는 테이블 생성 ASP 코드.
- 2) ODBC로 MYSQL에 있는 데이터베이스와의 연동을 통해 데이터를 가져옴.
- 3) 비회원의 이름과 전화번호를 나타내는 테이블로 구성. (MYSQL 서버 데이터베이스에 있는 Member 테이블에 있는 Non_Name_, Non_PhoneNumber 2개의 컬럼에서 데이터를 가져옴)

NonMember ASP 코드 실행 결과(비회원 테이블 생성)

The NonMember Table

NonMember List

Name	PhoneNumber
강동원	01048384858
박민지	01049754858
김선호	01048254858
박지성	01046734848
신서영	01048384858
이수호	01023584858
최지연	01045684828
서지혜	01067384878
최아영	01028384818
이강인	01048384858

[Return to Heather Sweeney Designs Home Page](#)

Seat ASP 코드 (좌석 테이블 생성 코드)

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Seat List Display ASP Page</title>
    <style type="text/css">
      h1 {text-align: center; color: blue}
      h2 {font-family: Ariel, sans-serif; text-align: left; color: blue}
      p.footer {text-align: center}
    </style>
  </head>
  <body>
    <h1>The Seat Table</h1>
    <hr />
    <h2>Seat List</h2>

    <%@ CODEPAGE="65001" LANGUAGE="VBSCRIPT" %>

    <%
      Session.CodePage = 949
      Response.CharSet = "euc-kr"
      Response.AddHeader "Pragma","no-cache"
      Response.AddHeader "cache-control", "no-staff"
      Response.Expires = -1

      Dim dbConCmd, SQLQuery, dbCon, rs, rsList
      dbConCmd= "DSN=root; Database=2schema; UID=root; PWD=krx8018@";
      SQLQuery = "SELECT SeatID, FloorID, IsAvailable FROM Seat"

      Set dbCon = Server.CreateObject("ADODB.Connection")
      Set rs = Server.CreateObject("ADODB.RecordSet")

      dbCon.Open dbConCmd
      rs.Open SQLQuery, dbCon

      <table>
        <tr>
          <th>SeatID</th>
          <th>FloorID</th>
          <th>IsAvailable</th>
        </tr>
      <%
        IF (rs.BOF AND rs.EOF) THEN
          Response.Write("유저정보가 존재하지 않습니다.")
        ELSE
          rsList= rs.GetRows()
          FOR x=0 TO UBound(rsList, 2)
            Response.Write("<tr>")
            FOR y=0 TO Ubound(rsList, 1)
              Response.Write("<td>")
              Response.Write(rsList(y, x))
              Response.Write("</td>")
            NEXT y
          NEXT x
        END IF
      <%
        <br/>
        <hr/>
        <p class="footer">
          <a href="http://www.root/study1.html">Return to Heather Sweeney Designs Home Page</a>
        </p>
        <hr/>
      </body>
    </html>
  </body>
</html>

```

- 1) 스터디 카페 좌석 정보를 보여주는 테이블 생성 ASP 코드.
- 2) ODBC로 MYSQL에 있는 데이터베이스와의 연동을 통해 데이터를 가져옴.
- 3) 좌석의 ID, 층수(floorID), 현재 이용가능한 좌석인지를(IsAvailable) 나타내는 테이블로 구성. (MYSQL 서버 데이터베이스에있는 Seat 테이블에 있는 SeatID, FloorID, IsAvailable 3개의 컬럼에서 데이터를 가져옴)

4 데이터베이스 어플리케이션 개발

Seat ASP 코드 실행 결과 (좌석 테이블 생성)

The Seat Table

Seat List

SeatID	FloorID	IsAvailable
1	1	1
2	1	1
3	1	0
4	1	1
5	1	1
6	1	0
7	1	0
8	1	1
9	1	0
10	1	1
11	2	1
12	2	0
13	2	1
14	2	0
15	2	1
16	2	1
17	2	0
18	2	1
19	2	1
20	2	0
21	3	1
22	3	1
23	3	1
24	3	1
25	3	1
26	3	0
27	3	1
28	3	0
29	3	1
30	3	1