- 목 차 -

1. 서론

1-1. 프로젝트 개요

제가 고안한 프로젝트는 대학생 맞춤형 자취방 정보를 열람하고 검색할 수 있는 시스템입니다. 저는 실제로 자취방을 구하는 과정에서 자취방을 구하는데 실제 후기나 방 구조를 보기 힘들어 어려움을 겪었고 이 문제를 해결하기 위해 대학생을 위한 자취방 정보 관리 시스템을 생각하게 되었습니다.

이 서비스를 통하여 대학생들이 자취방을 구할 때 보다 쉽게 지역, 가격, 평수, 샐활 옵션, 후기 등 다양한 조건에 따라 자취방을 검색하고, 신뢰도 높은 실제 사용자 후기를 통해 보다 합리적이고 편리한 거주지를 쉽게 찾아볼 수 있는 서비스를 제공하고자 합니다.

1-2. 프로젝트 목적

최근 대학가 주변의 자취방 수요는 지속적으로 증가하고 있으며, 특히 신입생이나 타지역에서 학교를 다니는 대학생들에게는 거주 공간을 선택하는 과정이 매우 중요한 문제로 여기지고 있습니다.

기존의 커뮤니티나 부동산 앱은 상업적 목적이 강하거나 후기를 볼 수 없거나 신뢰도가 낮은 경우가 많아, 실제 거주자가 작성한 솔직한 정보가 부족한 경우가 다반수입니다. 이 서비스를 통해 지역, 가격, 평수, 생활 옵션 및 실제 사용자 후기를 포함한 자취방 정보를 한눈에 열람할 수 있는 서비스를 제공하여 대학생들이 자취방을 찾을 때 겪는 정보 부족, 신뢰성 문제, 검색의 불편함 등을 해소 할 수 있는 맞춤형 자취방 정보 통합 시스템을 구축하는 것입니다.

2. 시스템 요구사항 분석

1. 사용자는 시스템에 가입하기 위해 이름, 이메일, 비밀번호를 입력해야 한다.
2. 이메일은 사용자 식별을 위한 고유 값이며, 중복 가입이 불가능해야 한다.
3. 로그인한 사용자는 전체 자취방 목록을 열람할 수 있으며, 지역·가격·평수·옵션 등의 조건으로 자취방을 검색할 수 있어야 한다.
4. 사용자는 자취방 목록 중 원하는 방의 상세 정보를 열람할 수 있어야 하며, 상세 정보에는 주소, 가격, 평수, 설명, 옵션, 이미지, 등록일, 후기 등이 포함되어야 한다.
5. 사용자는 자신이 관심 있는 자취방을 북마크 할 수 있어야 하며, 북마크한 자취방은 개인 마이페이지를 통해 확인할 수 있어야 한다.
6. 사용자는 자신이 열람한 자취방에 대해 별점(1~5점)과 텍스트로 후기를 작성할 수 있어야 하며, 하나의 자취방에 대해 한 번만 작성할 수 있어야 한다.
7. 자취방에 작성된 모든 후기는 누구나 열람할 수 있어야 하며, 최신순 정렬 기능을 지원해야 한다.
8. 사용자는 자취방에 대해 좋아요를 누를 수 있어야 하며, 좋아요는 동일한 사용자가 동일 방에 한 번만 누를 수 있어야 한다.
9. 좋아요 수는 자취방 정보에 표시되며, 인기순 정렬이나 추천 기능에 활용될 수 있어야 한다.
10. 사용자는 마이페이지를 통해 자신이 쓴 후기, 북마크 목록, 로그아웃 기능에 접근할 수 있어야 한다.
11. 관리자는 전체 자취방에 대해 등록, 수정, 삭제할 수 있어야 하며, 자취방 등록 시 옵션 항목도 함께 지정할 수 있어야 한다.
12. 관리자는 자취방에서 제공되는 옵션 목록을 관리할 수 있어야 하며, 옵션 항목을 추가하거나 삭제할 수 있어야 한다.
13. 관리자는 사용자들이 작성한 후기 중 부적절한 내용을 삭제할 수 있어야 하며,

악성 사용자를 제한하거나 계정을 삭제할 수 있어야 한다.

14. 사용자는 자신의 계정을 탈퇴할 수 있으며, 탈퇴 시 관련된 후기, 북마크, 좋아요 등의 정보도 함께 삭제되어야 한다.

15. 모든 데이터의 무결성을 위해 이메일은 고유해야 하며, 자취방-옵션은 다대다 관계로 구성되어야 하고, 후기와 좋아요는 중복되지 않도록 복합키로 관리되어야 한다.

3. 데이터 베이스 설계

3-1. 데이터 모델링

3-1-1. 개념적 설계

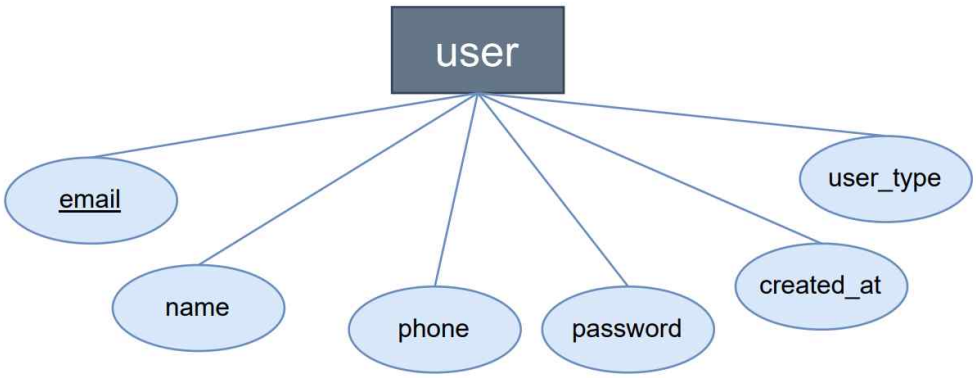
- 개체, 속성 추출

개체는 저장할 만한 가치가 있는 중요 데이터를 가진 데이터입니다. 요구사항 문장에서 업무와 관련이 깊은 의미 있는 명사를 찾아 개체와 속성으로 분리했습니다.

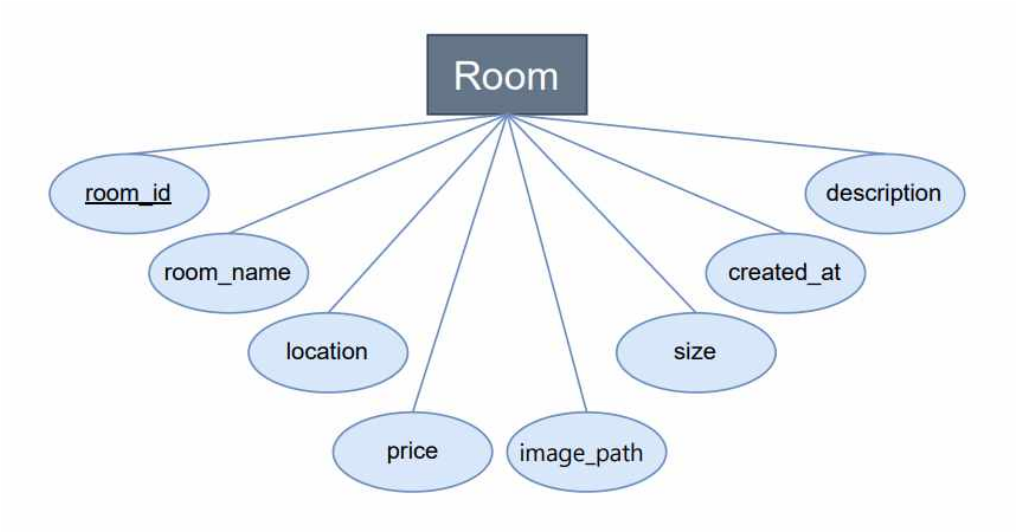
- 사용자는 시스템에 가입하기 위해 이름, 이메일, 비밀번호를 입력해야 한다.
- 이메일은 사용자 식별을 위한 고유 값이며, 중복 가입이 불가능해야 한다.
- 로그인한 사용자는 전체 자취방 목록을 열람할 수 있으며, 지역·가격·평수 등의 조건으로 자취방을 검색할 수 있어야 한다.
- 사용자는 자취방 목록 중 원하는 방의 상세 정보를 열람할 수 있어야 하며, 상세 정보에는 주소, 가격, 평수, 설명, 이미지, 등록일, 후기 등이 포함되어야 한다.
- 사용자는 자신이 열람한 자취방에 대해 별점(1~5점)과 텍스트로 후기를 작성할 수 있어야 하며, 하나의 자취방에 대해 한 번만 작성할 수 있어야 한다.
- 좋아요 수는 자취방 정보에 표시되며, 인기순 정렬이나 추천 기능에 활용될 수 있어야 한다.
- 관리자는 전체 자취방에 대해 등록, 수정, 삭제할 수 있어야 하며, 자취방 등록 시 옵션 항목도 함께 지정할 수 있어야 한다.
- 관리자는 자취방에서 제공되는 옵션 목록을 관리할 수 있어야 하며, 옵션 항목을 추가하거나 삭제할 수 있어야 한다.

요구사항을 분석한 후 추출한 요구사항 기반 개체와 속성들을 정리하면 아래 표처럼 나타납니다.

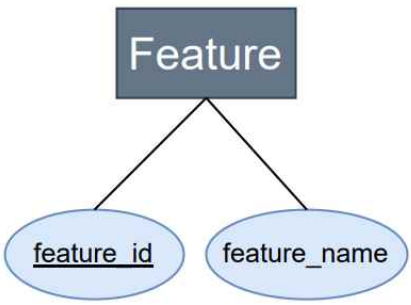
개체	속성
User(사용자)	user_id, name, email, phone, passweord, user_type, create_at
Room(자취방)	room_id, room_name, description, location, price, size, created_at, image_path, created_at
Feature(옵션)	feature_id, feature_name



[user 개체 E-R 다이어그램]



[Room 개체 E-R 다이어그램]



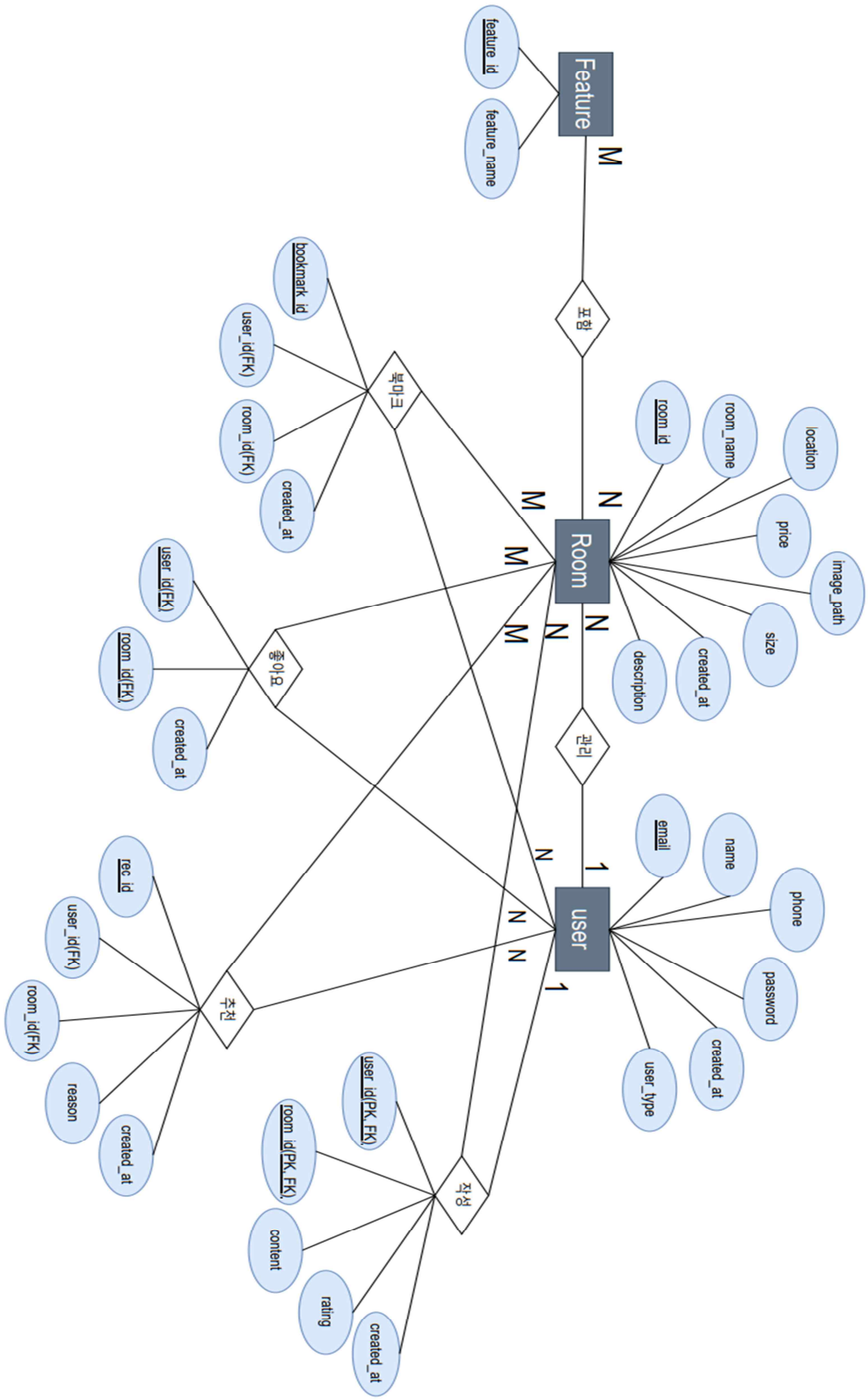
[Feature 개체 E-R 다이어그램]

- 관계 추출

관계는 개체 간의 의미 있는 연관성입니다. 일반적으로 요구사항 문장에서 개체간의 연광성을 의미 있게 표현한 동사로 표현됩니다. 의미가 같은 동사가 여러개일 경우에는 대표 동사를 하나만 선택하였습니다.

- 사용자는 자취방 목록 중 원하는 방의 상세 정보를 열람할 수 있어야 하며, 상세 정보에는 주소, 가격, 평수, 설명, 이미지, 등록일, 후기 등이 포함되어야 한다.
- 사용자는 자신이 관심 있는 자취방을 북마크 할 수 있어야 하며, 북마크한 자취방은 개인 마이페이지를 통해 확인할 수 있어야 한다.
- 사용자는 자신이 열람한 자취방에 대해 별점(1~5점)과 텍스트로 후기를 작성할 수 있어야 하며, 하나의 자취방에 대해 한 번만 작성할 수 있어야 한다.
- 사용자는 자취방에 대해 좋아요를 누를 수 있어야 하며, 좋아요는 동일한 사용자가 동일 방에 한 번만 누를 수 있어야 한다.
- 좋아요 수는 자취방 정보에 표시되며, 인기순 정렬이나 추천 기능에 활용될 수 있어야 한다.
- 관리자는 전체 자취방에 대해 등록, 수정, 삭제할 수 있어야 하며, 자취방 등록 시 옵션 항목도 함께 지정할 수 있어야 한다.

관계	관계에 참여하는 개체	관계 유형	속성
관리	User-Room	1:N	X
작성	User-Room	M:N	rating, content, created_at
좋아요	User-Room	M:N	X
북마크	User-Room	M:N	X
추천	User-Room	M:N	reason, created_at
포함	Room-Feature	M:N	X


Feature
feature_id
feature_name
포함
M
N
Room
room_id
room_name
location
price
image_path
size
created_at
description
관리
1
N
user
email
name
phone
password
created_at
user_type
북마크
bookmark_id
user_id(FK)
room_id(FK)
created_at
좋아요
user_id(FK)
room_id(FK)
created_at
추천
rec_id
user_id(FK)
room_id(FK)
reason
created_at
작성
user_id(PK, FK)
room_id(PK, FK)
content
rating
created_at

[전체 E-R 다이어그램]

3-1-2. 논리적 설계

[User 릴레이션]

user_id	name	email(UNIQUE)	phone	password	created_at	user_type

[Room 릴레이션]

room_id	user_id(FK)	room_name	location	price	size	image_path	created_at	description

[Feature 릴레이션]

feature_id	eature_name

[포함 릴레이션]

room_id(PK, FK)	feature_id(PK, FK)

[작성 릴레이션]

user_id(PK, FK)	room_id(PK, FK)	content	rating	created_at

[북마크 릴레이션]

room_id(PK, FK)	user_id(PK, FK)	created_at

[추천 릴레이션]

rec_id	user_id(FK)	room_id(FK)	reason	created_at

[좋아요 릴레이션]

user_id(PK, FK)	room_id(PK, FK)	created_at

3-1-3. 물리적 설계

```sql
-- 데이터베이스 생성 및 사용
CREATE DATABASE living_alone;
USE living_alone;
```

● 사용자 테이블 생성 쿼리문

```sql
CREATE TABLE User (
    user_id INT AUTO_INCREMENT PRIMARY KEY,                    -- 사용자 고유 ID
    name VARCHAR(100) NOT NULL,                                -- 사용자 이름
    email VARCHAR(100) UNIQUE NOT NULL,                        -- 이메일 (로그인 ID로 사용, 중복 불가)
    phone VARCHAR(20),                                         -- 연락처
    password VARCHAR(255) NOT NULL,                            -- 비밀번호
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,             -- 가입 일시
    user_type ENUM('user', 'admin') DEFAULT 'user'             -- 사용자 유형 (일반 or 관리자)
);
```

● 자취방 테이블 생성 쿼리문

```sql
CREATE TABLE Room (
    room_id INT AUTO_INCREMENT PRIMARY KEY,                    -- 자취방 고유 ID
    user_id INT NOT NULL,                                      -- 등록한 사용자 ID (방 등록자)
    room_name VARCHAR(100) NOT NULL,                           -- 자취방 이름
    location VARCHAR(255),                                     -- 위치 정보
    price DECIMAL(10, 2),                                      -- 월세
    size FLOAT,                                                -- 평수
    image_path TEXT,                                           -- 이미지 경로
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,             -- 등록 일시
    description TEXT,                                          -- 상세 설명
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE
);
```

● 옵션(Feature) 테이블 생성 쿼리문

```sql
CREATE TABLE Feature (
    feature_id INT AUTO_INCREMENT PRIMARY KEY,                 -- 옵션 고유 ID
    feature_name VARCHAR(100) UNIQUE NOT NULL                  -- 옵션 이름 (중복 불가)
);
```

● 방-옵션 중간 테이블 생성 쿼리문

```sql
CREATE TABLE RoomFeature (
    room_id INT NOT NULL,                                      -- 자취방 ID
    feature_id INT NOT NULL,                                   -- 옵션 ID
    PRIMARY KEY (room_id, feature_id),                         -- 복합 기본키
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE,
    FOREIGN KEY (feature_id) REFERENCES Feature(feature_id) ON DELETE CASCADE
);
```

● 리뷰 테이블 생성 쿼리문

```
CREATE TABLE Review (
    user_id INT NOT NULL,                                   -- 작성자 ID
    room_id INT NOT NULL,                                   -- 대상 자취방 ID
    content TEXT,                                           -- 후기 텍스트
    rating INT CHECK (rating BETWEEN 1 AND 5),              -- 별점 (1~5점 사이만 허용)
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,          -- 후기 작성일시
    PRIMARY KEY (user_id, room_id),                         -- 하나의 유저는 하나의 방에만 한 번 후기 작성 가능
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

● 북마크 테이블 생성 쿼리문

```
CREATE TABLE Bookmark (
    user_id INT NOT NULL,                                   -- 북마크한 사용자
    room_id INT NOT NULL,                                   -- 북마크한 자취방
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,          -- 북마크한 시각
    PRIMARY KEY (user_id, room_id),                         -- 같은 유저가 같은 방 북마크 1회만 가능
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

● 추천 테이블 생성 쿼리문

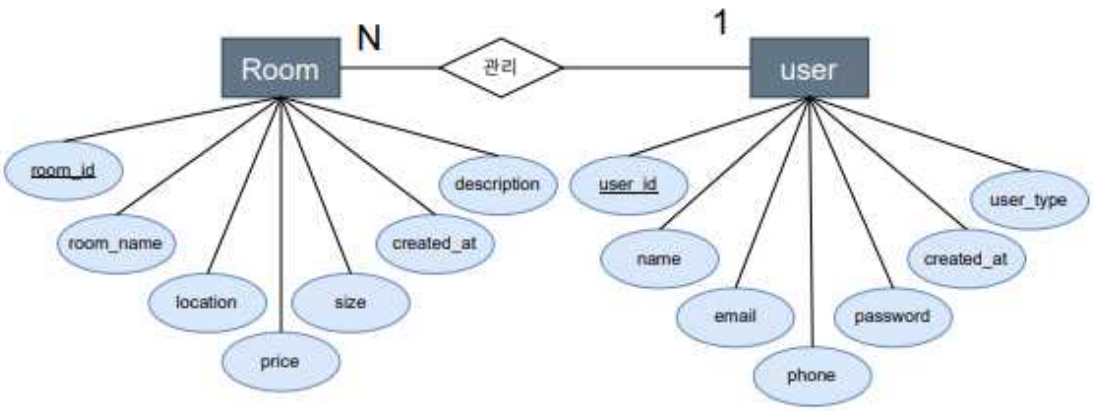
```
CREATE TABLE Recommendation (
    rec_id INT AUTO_INCREMENT PRIMARY KEY,                  -- 추천 고유 ID
    user_id INT NOT NULL,                                   -- 추천한 사용자
    room_id INT NOT NULL,                                   -- 추천 대상 자취방
    reason TEXT,                                            -- 추천 사유
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,          -- 추천 등록 시각
    UNIQUE (user_id, room_id),                              -- 같은 유저가 같은 방 중복 추천 불가
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

● 좋아요 테이블 생성 쿼리문

```
CREATE TABLE Likes (
    user_id INT NOT NULL,                                   -- 좋아요 누른 사용자
    room_id INT NOT NULL,                                   -- 좋아요 누른 자취방
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,          -- 좋아요 누른 시각
    PRIMARY KEY (user_id, room_id),                         -- 같은 유저가 같은 방 좋아요 한 번만 가능
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

3-2. E-R 다이어그램

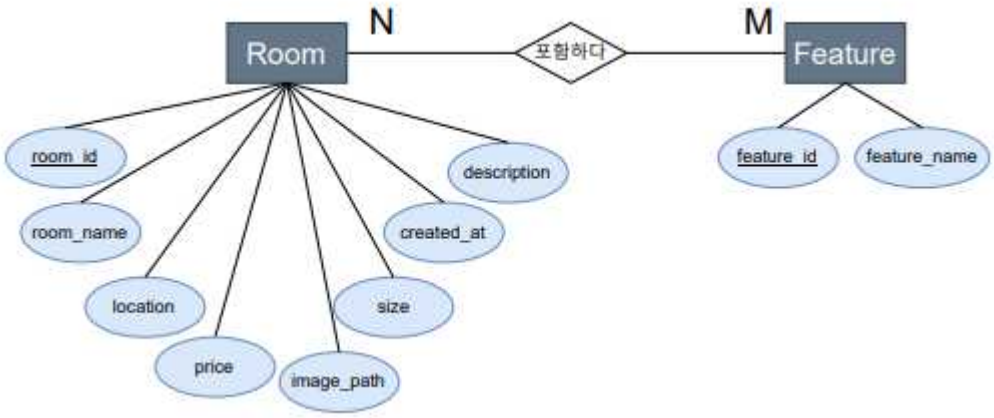
- Room 테이블과 user 테이블의 관계



[Room 테이블과 user 테이블의 E-R 다이어그램]

- User 테이블과 Room 테이블은 1:N(일대다) 관계로 정의됩니다. 한명의 사용자가 여러 자취방을 등록, 수정, 삭제 할 수 있습니다. 사용자 정보를 저장하고 있는 User 테이블의 기본키인 user_id는 Room 테이블의 외래키로 연결됩니다. 이 다이어그램을 통해 각 방이 어떤 사용자에 의해 등록되었는지 구분 할 수 있습니다.
- 사용자가 방을 등록하게 되면, 해당 방에 대한 정보가 Room 테이블에 저장되고 동시에 등록자의 user_id도 함께 기록됩니다.

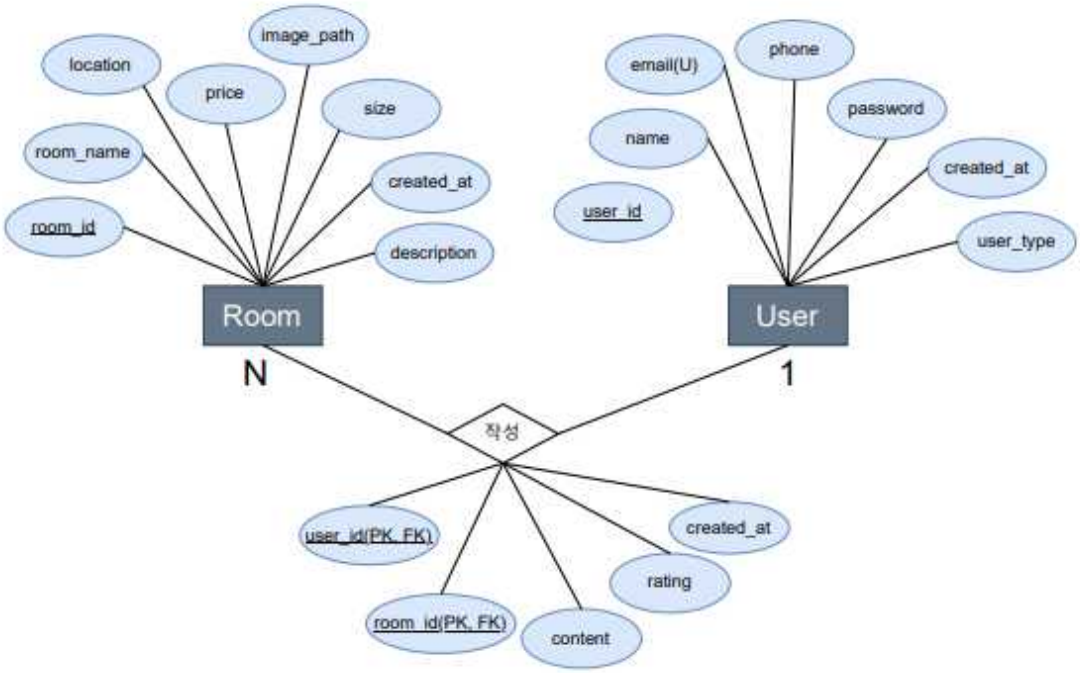
- **Room 테이블과 Feature 테이블의 관계**



[Room 테이블과 Feature 테이블의 E-R 다이어그램]

- Room 테이블과 Feature 테이블은 N:M(다대다) 관계로 정의됩니다. 이 두 테이블은 '포함하다'라는 관계로 연결되어 있습니다. 하나의 자취방(Room)은 여러 개의 옵션(Feature)을 포함할 수 있고, 하나의 옵션 역시 여러 자취방에 중복으로 저장될 수 있습니다.
- 실제 논리 또는 물리 설계 단계에서는 이 관계를 구현하기 위해 RoomFeature 중간 테이블을 생성하게 됩니다. 중간 테이블인 RoomFeature에는 room_id와 feature_id 두 개의 외래키가 저장되며, 이를 통해 어떤 방에 어떤 옵션이 포함되어 있는지를 쉽게 나타낼 수 있습니다.

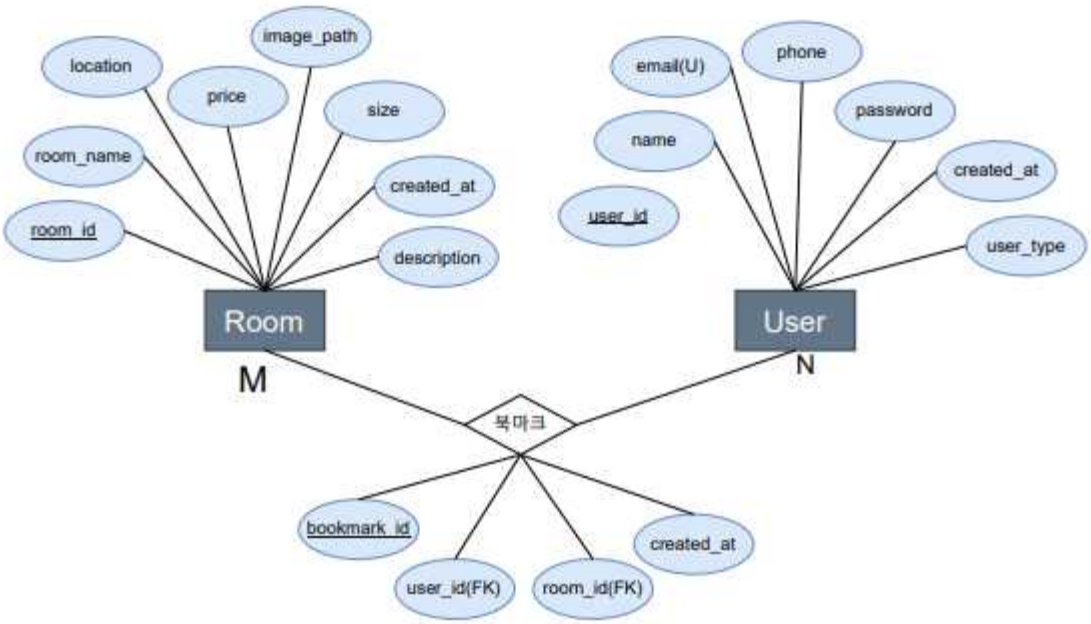
- **User – 작성 – Room 테이블의 관계**



[User – 작성 – Room 테이블의 E-R 다이어그램]

- User 테이블과 Room 테이블은 N:M(다대다) 관계로 정의됩니다. 두 테이블은 '작성'이라는 관계로 연결되어 있습니다. 한 명의 사용자는 여러 자취방에 대해 리뷰를 작성할 수 있고, 하나의 자취방은 여러 사용자로부터 리뷰를 받을 수 있습니다.
- '작성' 관계에는 리뷰 내용을 담는 content, 평점을 나타내는 rating, 작성 시점을 기록하는 created_at과 같은 관계 속성이 포함되며, 사용자 후기 정보(review)를 관리할 수 있도록 해 줍니다. 또한 동일 사용자가 동일 방에 중복 리뷰를 작성하지 못하도록 user와 room 간의 조합이 하나만 존재하게 해야합니다.

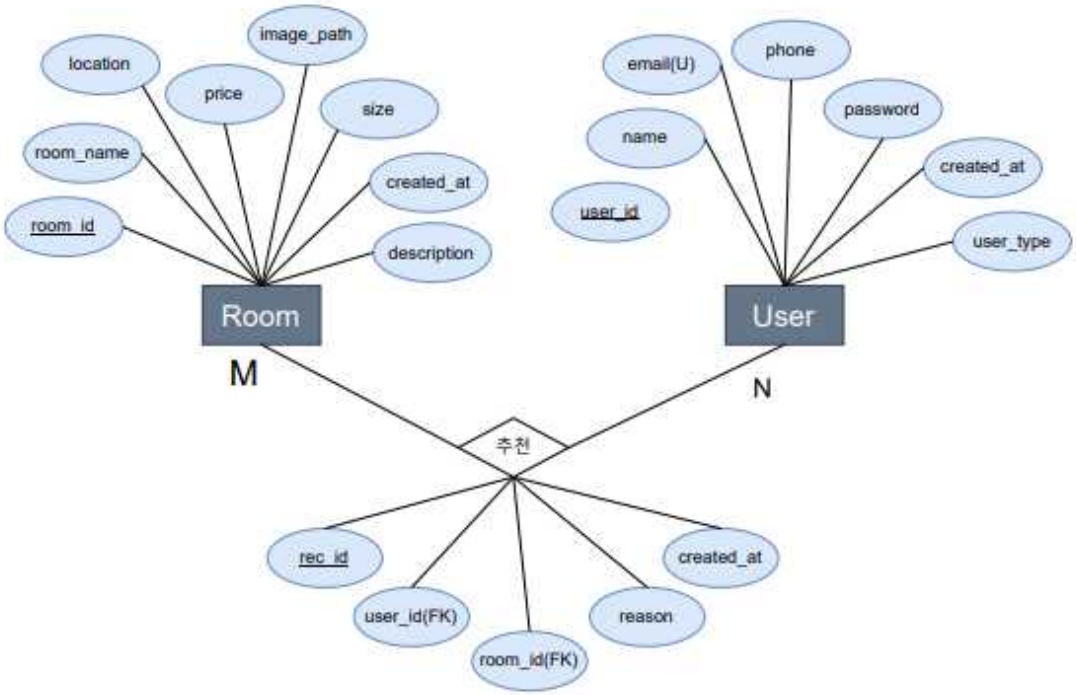
- **User – 북마크 – Room 테이블의 관계**



[User – 북마크 – Room 테이블의 E-R 다이어그램]

- User 테이블과 Room 테이블은 N:M(다대다) 관계로 정의됩니다. 이 두 테이블은 '북마크' 라는 관계를 통해 연결되어 있습니다. 한 명의 사용자는 여러 자취방을 북마크할 수 있고, 하나의 자취방도 여러 사용자에 의해 북마크 될 수 있습니다.
- '북마크' 관계에는 북마크한 시점을 나타내는 created_at과 북마크 식별을 위한 bookmark_id 등의 관계 속성이 포함되어 있습니다.

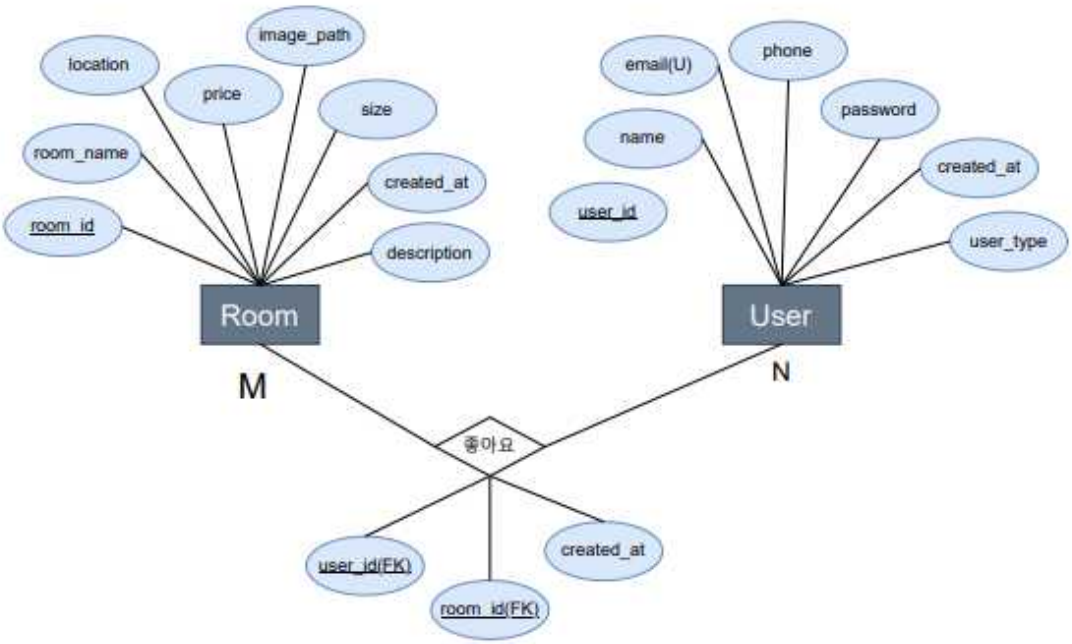
- **User – 추천 – Room 테이블의 관계**



[User – 추천 – Room의 E-R 다이어그램]

- User 테이블과 Room 테이블은 N:M(다대다) 관계로 정의됩니다. 이 테이블은 '추천'이라는 관계명을 통해 연결되어 있습니다. 한명의 사용자는 여러 자취방을 추천할 수 있고, 하나의 자취방도 여러 사용자에게 추천될 수 있습니다.
- '추천' 관계에는 추천 이유를 기록하는 reason, 추천한 시점을 나타내는 created_at, 그리고 추천을 식별하기 위한 rec_id 등의 관계 속성이 포함되어 있어 해당 관계를 통해 사용자가 추천한 내역을 관리할 수 있습니다.

- **User – 좋아요 – Room 테이블의 관계**



[User – 좋아요 – Room 테이블의 E-R 다이어그램]

- User 테이블과 Room 테이블은 N:M(다대다) 관계로 정의됩니다. 이 테이블은 '좋아요'라는 관계명을 통해 연결되어 있습니다. 하나의 사용자는 여러 자취방에 '좋아요'를 누를 수 있고, 하나의 자취방도 여러 사용자로부터 '좋아요'를 받을 수 있습니다.
- '좋아요' 관계에는 좋아요를 누른 시점을 나타내는 created_at이라는 관계 속성이 포함되어 있으며, 자취방 인기 순위를 정하는 데 활용될 수 있습니다.

3-3. 테이블 정의 및 설명

USER 테이블

- 자취방 서비스를 사용하는 회원 정보를 저장하는 테이블입니다. 자취방 서비스를 이용하는 사용자의 고유한 정보들을 관리합니다. 사용자 고유 ID, 이름, 이메일, 연락처, 비밀번호, 가입일, 사용자 유형(일반/관리자) 등을 저장합니다.

속성	설명	데이터타입	제약조건
user_id	사용자 고유 ID	INT	PRIMARY KEY, AUTO_INCREMENT
name	사용자 이름	VARCHAR(50)	NOT NULL
email	이메일 주소	VARCHAR(100)	NOT NULL, UNIQUE
phone	연락처	VARCHAR(20)	NULL
password	비밀번호	VARCHAR(225)	NOT NULL
user_type	사용자 유형	ENUM('user', 'admin')	DEFAULT 'user'
created_at	가입 일시	DATETIME	DEFAULT CURRENT_TIMESTAMP

Room 테이블

- 등록된 자취방 정보를 저장하는 테이블입니다. 각 방의 고유 ID, 이름, 위치, 월세 가격, 면적, 설명, 등록일을 저장합니다.

속성	설명	데이터타입	제약조건
room_id	방 고유 ID	INT	PRIMARY KEY, AUTO_INCREMENT
room_name	방 이름	VARCHAR(100)	NOT NULL
location	위치 정보	VARCHAR(200)	NOT NULL
price	월세 가격	INT	NOT NULL
size	평수	FLOAT	NOT NULL
description	상세 설명	TEXT	NULL
image_path	이미지 파일 경로	VARCHAR(255)	NULL 허용
created_at	등록 일시	DATETIME	DEFAULT CURRENT_TIMESTAMP

Feature 테이블

- 자취방에 포함될 수 있는 옵션을 저장하는 테이블입니다. 각 옵션은 고유 ID와 이름을 가지며, 여러 자취방에서 공통적으로 사용될 수 있습니다

속성	설명	데이터타입	제약조건
feature_id	옵션 고유 ID	INT	PRIMARY KEY, AUTO_INCREMENT
feature_name	옵션 이름	VARCHAR(50)	NOT NULL, UNIQUE

포함 테이블

- Romm 테이블과 Feature 테이블의 N:M(다대다) 관계를 관리하기 위한 중간 테이블입니다.

속성	설명	데이터타입	제약조건
room_id	자취방 ID	INT	PRIMARY KEY, FOREIGN KEY → Room(room_id), ON DELETE CASCADE
feature_id	옵션 ID	INT	PRIMARY KEY, FOREIGN KEY → feature(feature_id), ON DELETE CASCADE

Review(작성) 테이블

- 사용자가 작성한 자취방 후기를 저장하는 테이블입니다. 리뷰는 특정 자취방과 사용자에 연결되며, 평점 (1~5), 텍스트 내용, 작성 일시 등의 정보를 포함합니다.

속성	설명	데이터타입	제약조건
user_id	작성자 ID	INT	PRIMARY KEY (user_id, room_id), FOREIGN KEY → User(user_id), ON DELETE CASCADE
room_id	대상 방 ID	INT	PRIMARY KEY (user_id, room_id), FOREIGN KEY → Room(room_id), ON DELETE CASCADE
content	후기내용	TEXT	NULL
rating	평점(1~5)	INT	CHECK (rating BETWEEN 1 AND 5), NOT NULL
created_at	등록 일시	DATETIME	DEFAULT CURRENT_TIMESTAMP

Bookmark(북마크) 테이블

- 사용자가 관심 있는 자취방을 북마크한 기록을 저장합니다.

속성	설명	데이터타입	제약조건
bookmark_id	북마크 ID	INT	PRIMARY KEY, AUTO_INCREMENT
user_id	사용자 ID	INT	FOREIGN KEY → User(user_id), ON DELETE CASCADE
room_id	북마크 대상 방 ID	INT	FOREIGN KEY → Room(room_id), ON DELETE CASCADE
created_at	북마크 일시	DATETIME	DEFAULT CURRENT_TIMESTAMP

Recommendation(추천) 테이블

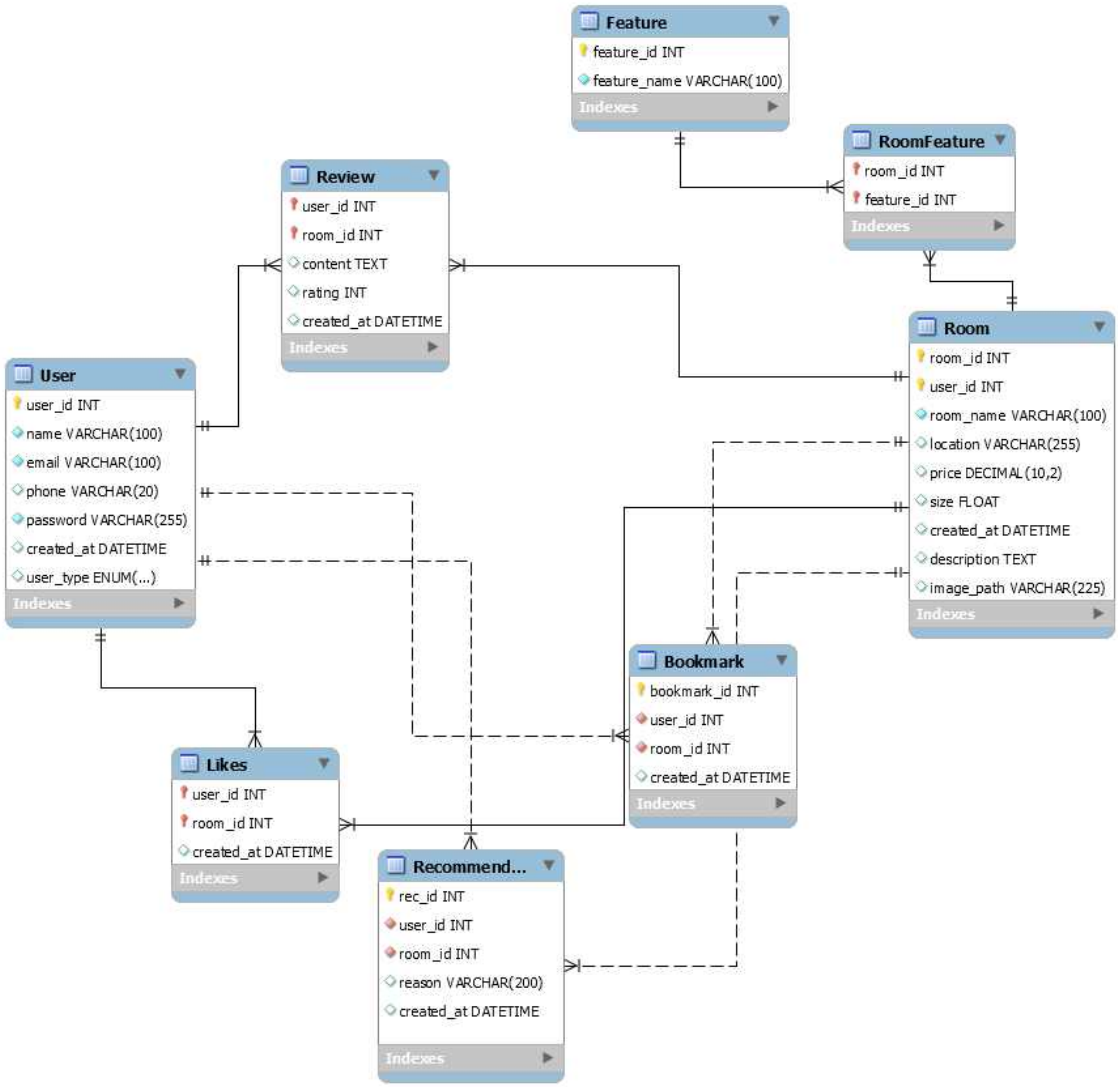
- 사용자가 자취방을 다른 사용자에게 추천한 기록을 저장합니다. 추천사유, 추천 일시 등의 정보가 포함됩니다.

속성	설명	데이터타입	제약조건
rec_id	추천 고유 ID	INT	PRIMARY KEY, AUTO_INCREMENT
user_id	추천 사용자 ID	INT	FOREIGN KEY → User(user_id), ON DELETE CASCADE
room_id	추천한 방 ID	INT	FOREIGN KEY → Room(room_id), ON DELETE CASCADE
reason	추천 사유	VARCHAR(200)	NULL
created_at	추천 일시	DATETIME	DEFAULT CURRENT_TIMESTAMP

Likes(좋아요) 테이블

- 사용자가 자취방에 '좋아요'를 누른 기록을 저장합니다. 사용자와 자취방 간 중복을 허용하지 않습니다.

속성	설명	데이터타입	제약조건
user_id	좋아요한 사용자 ID	INT	PRIMARY KEY, FOREIGN KEY → User(user_id), ON DELETE CASCADE
room_id	좋아요한 방 ID	INT	PRIMARY KEY, FOREIGN KEY → Room(room_id), ON DELETE CASCADE
created_at	좋아요 일시	DATETIME	DEFAULT CURRENT_TIMESTAMP



[MySQL Wo rkbench에서 각 쿼리문으로 직접 설계한 테이블]

4. 데이터 베이스 정규화

데이터 베이스 설계에서 데이터의 중복을 최소화 하고 데이터의 무결성을 유지하기 위해 사용되는 프로세스입니다.

정규화를 통해 데이터를 여러 릴레이션으로 나누어 중복 저장되는 데이터를 줄일 수 있고, 중복된 데이터를 수정할 때 발생하는 데이터 불일치 문제를 방지할 수 있습니다. 또한 정규화 된 데이터베이스는 구조가 논리적으로 잘 구성되어 있어 데이터 수정, 삭제, 삽입 등 유지보수가 용이하고, 데이터 검색과 수정이 간단하고 일관되게 이루어집니다. 마지막으로 데이터가 잘 구조화되어 있으면 릴레이션 간의 관계가 명확하기 때문에 새로운 릴레이션이나 속성을 추가할 때다른 데이터에 영향을 주지 않고 쉽게 확장할 수 있습니다.

```sql
-- User 테이블
CREATE TABLE User (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    email VARCHAR(100),
    department_name VARCHAR(50)
);

-- Room 테이블
CREATE TABLE Room (
    room_id INT PRIMARY KEY AUTO_INCREMENT,
    room_name VARCHAR(100) NOT NULL,
    location VARCHAR(200) NOT NULL,
    price INT NOT NULL,
    size FLOAT NOT NULL,
    description TEXT,
    image_path VARCHAR(255),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

```sql
-- Feature 테이블
CREATE TABLE Feature (
    feature_id INT PRIMARY KEY AUTO_INCREMENT,
    feature_name VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE RoomFeature (
    room_id INT,
    feature_id INT,
    feature_name VARCHAR(50),
    PRIMARY KEY (room_id, feature_id)
);
```

```sql
-- Review 테이블
CREATE TABLE Review (
    user_id INT,
    room_id INT,
    content TEXT,
    rating INT NOT NULL CHECK (rating BETWEEN 1 AND 5),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (user_id, room_id),
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);

-- Bookmark 테이블
CREATE TABLE Bookmark (
    bookmark_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    room_id INT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

```sql
-- Recommendation 테이블
CREATE TABLE Recommendation (
    rec_id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT,
    room_id INT,
    reason VARCHAR(200),
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);

-- Likes 테이블
CREATE TABLE Likes (
    user_id INT,
    room_id INT,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    PRIMARY KEY (user_id, room_id),
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

[초기의 스키마]

4-1. 제 1 정규형 (1NF)

제 1정규형은 릴레이션에 속한 모든 속성의 도메인이 원자값(Atomic Value)을 가져야 한다는 특징이 있습니다.

초기 스키마는 이 요구사항을 만족합니다

4-2. 제 2 정규형 (2NF)

제 2 정규형에서는 모든 속성이 기본 키 전체에 완전 함수 종속되어야 합니다.

기본 키가 복합키일 경우, 기본 키의 일부에만 종속된 속성이 존재하면 2NF를 위반한 것입니다. 현재 스키마 중 RoomFeature 테이블은 (room_id, feature_id)를 복합 기본 키로 가지고 있습니다. 이 테이블 안에 feature_name 속성이 존재한다면, 이 값은 feature_id에만 종속되어 있습니다.

기본 키 전체가 아닌 feature_id 하나에만 종속된 속성이므로 부분 함수 종속에 해당하며, 2NF를 위반합니다. 이 문제를 해결하기 위해 feature_name 속성을 별도의 테이블인 Feature로 분리합니다. Feature 테이블은 feature_id를 기본 키로 하여, 각 옵션 이름을 고유하게 저장합니다.

RoomFeature 테이블은 room_id와 feature_id의 관계만 저장하며, feature_name에 직접 접근하지 않도록 설계하여 2정규형을 만족하도록 정규화합니다.

```sql
-- Feature 테이블
CREATE TABLE Feature (
    feature_id INT PRIMARY KEY AUTO_INCREMENT,
    feature_name VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE RoomFeature (
    room_id INT,
    feature_id INT,
    feature_name VARCHAR(50),
    PRIMARY KEY (room_id, feature_id)
);
```

[제 2 정규형을 만족하지 않는 스키마]

```sql
CREATE TABLE Feature (
    feature_id INT PRIMARY KEY AUTO_INCREMENT,
    feature_name VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE RoomFeature (
    room_id INT,
    feature_id INT,
    PRIMARY KEY (room_id, feature_id),
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE,
    FOREIGN KEY (feature_id) REFERENCES Feature(feature_id) ON DELETE CASCADE
);
```

[제 2 정규형을 만족하는 스키마]

4-3. 제 3 정규형 (3NF)

제 3 정규형에서는 모든 속성이 기본 키에 직접적으로 종속되어야 하며, 기본 키가 아닌 속성 간의 종속(이행적 종속)이 존재하면 안 됩니다. 스키마 중 User 테이블의 구조를 보면, user_id가 기본 키입니다. 그 외 속성으로는 email, name, phone, password, user_type 등이 존재합니다.

email에는 UNIQUE 제약조건이 설정되어 있어 중복이 허용되지 않으며, name, phone, password, user_type은 email을 통해서 유추되거나 결정되지 않습니다.

속성 간에 실질적인 종속 관계가 존재하지 않으며, 모든 속성이 user_id에 직접적으로 종속되도록 설계되어 있습니다. User 테이블 내의 모든 속성은 user_id를 통해서만 유일하게 식별되며, 다른 일반 속성에 종속되지 않습니다. 정규화된 설계를 통해 이행적 종속 관계를 제거하고, 결과적으로 User 테이블은 제 3 정규형(3NF)을 만족하는 구조가 됩니다.

```
-- User 테이블
CREATE TABLE User (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50),
    email VARCHAR(100),
    department_name VARCHAR(50)
);
```

[제 3 정규형 만족하지 않는 스키마]

```
CREATE TABLE User (
    user_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    phone VARCHAR(20),
    password VARCHAR(255) NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
    user_type ENUM('user', 'admin') DEFAULT 'user'
);
```

[제 3 정규형 만족하는 스키마]

4-4. 최종 정규화된 스키마

```sql
-- 사용자 테이블: 회원 계정 정보 저장
CREATE TABLE User (
    user_id INT AUTO_INCREMENT PRIMARY KEY,                    -- 사용자 고유 ID
    name VARCHAR(100) NOT NULL,                                -- 사용자 이름
    email VARCHAR(100) UNIQUE NOT NULL,                        -- 이메일 (중복 불가, 로그인 ID로 사용)
    phone VARCHAR(20),                                         -- 연락처
    password VARCHAR(255) NOT NULL,                            -- 로그인 비밀번호
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,             -- 가입 일시
    user_type ENUM('user', 'admin') DEFAULT 'user'             -- 사용자 유형 ('user': 일반, 'admin': 관리자)
);

-- 자취방 테이블: 방 상세 정보 저장
CREATE TABLE Room (
    room_id INT AUTO_INCREMENT PRIMARY KEY,                    -- 자취방 고유 ID
    room_name VARCHAR(100) NOT NULL,                           -- 자취방 이름
    location VARCHAR(255),                                     -- 위치 정보
    price DECIMAL(10, 2),                                      -- 월세 가격
    size FLOAT,                                                -- 평수
    image_path TEXT,                                           --
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,             -- 방 등록 일시
    description TEXT                                           -- 상세 설명
);
```

```sql
-- 옵션 테이블: 자취방에 제공되는 옵션 목록 저장
CREATE TABLE Feature (
    feature_id INT AUTO_INCREMENT PRIMARY KEY,                 -- 옵션 고유 ID
    feature_name VARCHAR(100) UNIQUE NOT NULL                  -- 옵션 이름
);

-- 자취방-옵션 관계 테이블 (N:M 매핑용)
CREATE TABLE RoomFeature (
    room_id INT NOT NULL,                                      -- 자취방 ID
    feature_id INT NOT NULL,                                   -- 옵션 ID
    PRIMARY KEY (room_id, feature_id),                         -- 복합 기본키
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE,
    FOREIGN KEY (feature_id) REFERENCES Feature(feature_id) ON DELETE CASCADE
);

-- 후기 테이블: 사용자들이 남긴 리뷰 정보 저장
CREATE TABLE Review (
    user_id INT NOT NULL,                                      -- 작성자 ID
    room_id INT NOT NULL,                                      -- 대상 자취방 ID
    content TEXT,                                              -- 후기 텍스트
    rating INT CHECK (rating BETWEEN 1 AND 5),                 -- 별점 (1~5점 사이만 허용)
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,             -- 후기 작성일시
    PRIMARY KEY (user_id, room_id),                            -- 하나의 유저는 하나의 방에만 한 번 후기 작성 가능
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

```sql
-- 북마크 테이블
CREATE TABLE Bookmark (
    user_id INT NOT NULL,                                      -- 북마크한 사용자
    room_id INT NOT NULL,                                      -- 북마크한 자취방
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,             -- 북마크한 시각
    PRIMARY KEY (user_id, room_id),                            -- 같은 유저가 같은 방 북마크 1회만 가능
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```

```sql
-- 추천 테이블
CREATE TABLE Recommendation (
    rec_id INT AUTO_INCREMENT PRIMARY KEY,                    -- 추천 고유 ID
    user_id INT NOT NULL,                                     -- 추천한 사용자
    room_id INT NOT NULL,                                     -- 추천 대상 자취방
    reason TEXT,                                              -- 추천 사유
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,            -- 추천 등록 시각
    UNIQUE (user_id, room_id),                                -- 같은 유저가 같은 방 중복 추천 불가
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);

-- 좋아요 테이블: 자취방 좋아요 기록
CREATE TABLE Likes (
    user_id INT NOT NULL,                                     -- 좋아요 누른 사용자
    room_id INT NOT NULL,                                     -- 좋아요 누른 자취방
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP,            -- 좋아요 누른 시각
    PRIMARY KEY (user_id, room_id),                           -- 같은 유저가 같은 방 좋아요 한 번만 가능
    FOREIGN KEY (user_id) REFERENCES User(user_id) ON DELETE CASCADE,
    FOREIGN KEY (room_id) REFERENCES Room(room_id) ON DELETE CASCADE
);
```