

SK네트웍스 Family AI과정 3기

## 데이터 수집 및 저장 데이터 베이스 설계서

### □ 개요

- 산출물 단계 : 데이터 수집 및 저장
- 평가 산출물 : 데이터베이스 설계서
- 제출 일자 : 2026. 01. 26
- 깃허브 경로 : <https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN20-FINAL-3TEAM>
- 작성 팀원 : 나호성

개요	<ul style="list-style-type: none"><li>• 소개</li><li>• 시스템 개요</li></ul>
구조	<ul style="list-style-type: none"><li>• 시스템 아키텍처</li><li>• 요구사항 매트릭스</li></ul>
설계	<ul style="list-style-type: none"><li>• 데이터 설계</li><li>• 설계 근거</li><li>• ERD</li></ul>

소개	<ul style="list-style-type: none"> <li>- 목적           <ul style="list-style-type: none"> <li>- 이 데이터베이스는 사용자가 웹 서비스를 통해 채팅 기록 관리, 도면 정보를 효율적으로 저장, 관리할 수 있도록 설계됨</li> </ul> </li> <li>- 범위           <ul style="list-style-type: none"> <li>- 사용자 인증 및 정보 관리, 채팅방 및 채팅 기록(QnA) 저장, 도면 업로드 및 공간/객체 정보 관리, 각 기능별 세션 및 이력 관리 등을 포함함.</li> </ul> </li> </ul>						
시스템 개요	<ul style="list-style-type: none"> <li>- 시스템 역할           <ul style="list-style-type: none"> <li>- 인증된 사용자만 정상적인 경로(로그인 &gt; 웹 서비스 접속 &gt; 채팅방 진입)로 채팅 및 도면 관리 기능을 사용할 수 있도록 제한함.</li> <li>- 사용자가 채팅방에서 질문을 입력하면, LLM 기반 챗봇이 답변을 제공하고, 질문·답변(QnA) 내역을 데이터베이스에 저장함.</li> <li>- 데이터베이스는 사용자 정보, 업로드된 도면(플로어플랜) 정보, 채팅 방 및 QnA 기록, 공간/객체 정보를 저장·관리함.</li> </ul> </li> <li>- 주요 기능           <ul style="list-style-type: none"> <li>- 사용자 인증 및 로그인 이력 관리</li> <li>- 도면(플로어플랜) 업로드 및 관리</li> <li>- 채팅방 및 세션별 QnA 기록 저장</li> <li>- QnA 이력 및 도면/공간/객체 데이터 추적 및 관리</li> </ul> </li> </ul>						
시스템 아키텍처	<ul style="list-style-type: none"> <li>- 데이터베이스 구조</li> <li>- 관계형 데이터베이스(RDBMS)로 설계, 주요 테이블은 다음과 같음:           <ul style="list-style-type: none"> <li>- users: 사용자 정보를 저장</li> <li>- floor_plans: 업로드된 도면(플로어플랜) 정보를 저장</li> <li>- chatroom: 채팅방 정보를 저장</li> <li>- chat_history: QnA(질문/답변) 기록을 저장</li> <li>- room, objs, strs: 도면 내 공간 및 객체 정보를 저장</li> </ul> </li> <li>- 적용한 데이터베이스           <ul style="list-style-type: none"> <li>- AWS RDS 기반의 MySQL DB를 사용</li> <li>- 도면 이미지 등 대용량 파일은 별도의 스토리지(예: S3)에 저장하고, DB에는 파일 경로 및 메타데이터만 저장</li> <li>- MySQL DB는 사용자, 도면, 채팅방, QnA 등 주요 서비스 데이터를 관리</li> </ul> </li> </ul>						
요구사항 매트릭스	<p>요구사항 및 관련 테이블</p> <table border="1" data-bbox="434 1681 1426 1896"> <thead> <tr> <th>요구사항</th> <th>관련 테이블</th> </tr> </thead> <tbody> <tr> <td>사용자 로그인 정보 저장</td> <td>usertable</td> </tr> <tr> <td>채팅방 생성/ 수정/ 삭제</td> <td>Chatroom, usertable</td> </tr> </tbody> </table>	요구사항	관련 테이블	사용자 로그인 정보 저장	usertable	채팅방 생성/ 수정/ 삭제	Chatroom, usertable
요구사항	관련 테이블						
사용자 로그인 정보 저장	usertable						
채팅방 생성/ 수정/ 삭제	Chatroom, usertable						

	<table border="1"> <tr> <td>채팅 기록 저장/조회/삭제</td><td>ChatHistory, Chatroom</td></tr> <tr> <td>도면 관리</td><td>Floor_plans, usertable</td></tr> <tr> <td>공간 및 객체 관리</td><td>Room, objs, strs</td></tr> </table>	채팅 기록 저장/조회/삭제	ChatHistory, Chatroom	도면 관리	Floor_plans, usertable	공간 및 객체 관리	Room, objs, strs
채팅 기록 저장/조회/삭제	ChatHistory, Chatroom						
도면 관리	Floor_plans, usertable						
공간 및 객체 관리	Room, objs, strs						
데이터 설계	<p>-- 사용자 정보를 저장하는 테이블  CREATE TABLE users (     id BIGINT AUTO_INCREMENT PRIMARY KEY,     email VARCHAR(100) NOT NULL UNIQUE, -- 사용자 이메일(로그인 ID)     pw VARCHAR(255) NOT NULL, -- 비밀번호     name VARCHAR(50), -- 이름     phonenum INT, -- 전화번호     role VARCHAR(255) NOT NULL DEFAULT 'USER', -- 권한(기본값 USER)     create_at DATE, -- 가입일     update_at DATE -- 정보 수정일 );</p> <p>-- 도면(플로어플랜) 정보를 저장하는 테이블  CREATE TABLE floor_plans (     id BIGINT AUTO_INCREMENT PRIMARY KEY,     name VARCHAR(100), -- 도면 이름     imageUrl VARCHAR(255), -- 도면 이미지 경로     user_id BIGINT NOT NULL, -- 도면 소유자(사용자)     created_at DATE NOT NULL, -- 도면 생성일     FOREIGN KEY (user_id) REFERENCES users(id) );</p> <p>-- 채팅방 정보를 저장하는 테이블  CREATE TABLE chatroom (     id BIGINT AUTO_INCREMENT PRIMARY KEY,     user_id BIGINT NOT NULL, -- 채팅방 소유자(사용자)     name VARCHAR(255), -- 채팅방 이름     created_at DATE NOT NULL, -- 채팅방 생성일     FOREIGN KEY (user_id) REFERENCES users(id) );</p> <p>-- 채팅(QnA) 기록을 저장하는 테이블  CREATE TABLE chat_history (     id BIGINT AUTO_INCREMENT PRIMARY KEY,     chatroom_id BIGINT NOT NULL, -- 소속 채팅방     question VARCHAR(255), -- 질문 내용     answer VARCHAR(255) -- 답변 내용 );</p>						

```

answer VARCHAR(255),          -- 답변 내용
created_at DATE NOT NULL,    -- 기록 생성일
FOREIGN KEY (chatroom_id) REFERENCES chatroom(id)
);

-- 도면 내 공간(Room) 정보를 저장하는 테이블
CREATE TABLE room (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    spcname VARCHAR(100),           -- 공간 이름
    ocrname VARCHAR(255),          -- OCR로 추출한 이름
    bbox VARCHAR(255),             -- 바운딩 박스 정보
    centroid VARCHAR(255),         -- 중심 좌표
    area FLOAT,                   -- 면적
    areapercent FLOAT,            -- 전체 대비 면적 비율
    floorplan_id BIGINT,          -- 소속 도면
    FOREIGN KEY (floorplan_id) REFERENCES floor_plans(id)
);

-- 공간 내 객체(OBJ) 정보를 저장하는 테이블
CREATE TABLE objs (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),             -- 객체 이름
    bbox VARCHAR(255),             -- 바운딩 박스 정보
    centroid VARCHAR(255),          -- 중심 좌표
    room_id BIGINT,                -- 소속 공간(Room)
    FOREIGN KEY (room_id) REFERENCES room(id)
);

-- 공간 내 구조(STR) 정보를 저장하는 테이블
CREATE TABLE strss (
    id BIGINT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),             -- 구조 이름
    bbox VARCHAR(255),             -- 바운딩 박스 정보
    centroid VARCHAR(255),          -- 중심 좌표
    area VARCHAR(255),             -- 면적 정보
    room_id BIGINT,                -- 소속 공간(Room)
    FOREIGN KEY (room_id) REFERENCES room(id)
);

테이블 간 주요 관계
users 1:N chatroom, floor_plans
chatroom 1:N chat_history
floor_plans 1:N room

```

	room 1:N objs, strs
설계 근거	<ul style="list-style-type: none"> <li>-데이터 무결성 및 트랜잭션 보장을 위해 RDBMS(MySQL) 선택</li> <li>-JPA 기반 ORM 사용으로 객체-관계 매핑 및 유지보수 용이</li> <li>-사용자, 채팅방, 채팅기록, 도면, 공간, 객체 등 도메인별로 테이블 분리</li> <li>-외래키(FK)로 엔티티 간 관계 명확히 정의</li> <li>-확장성과 성능을 고려해 인덱스, 제약조건, 데이터 타입 선정</li> </ul>
ERD	<pre>     erDiagram         users   --o{ floorPlans : "room"         users   --o{ room : "room"         floorPlans   --o{ room : "room"         chatroom   --o{ chathistory         chathistory   --o{ objs         chathistory   --o{ strs         objs   --o{ room         strs   --o{ room     }   </pre> <p>The ER diagram illustrates the database schema with the following entities and their relationships:</p> <ul style="list-style-type: none"> <li><b>users</b>: Contains fields like id (BIGINT), email (VARCHAR), pw (VARCHAR), name (VARCHAR), phonenum (INT), role (VARCHAR), create_at (DATE), update_at (DATE). It has many-to-many relationships with <b>floor_plans</b> and <b>room</b>.</li> <li><b>floor_plans</b>: Contains fields like id (BIGINT), name (VARCHAR), imageUrl (VARCHAR), user_id (BIGINT), created_at (DATE). It has many-to-many relationships with <b>room</b> and <b>chatroom</b>.</li> <li><b>room</b>: Contains fields like id (BIGINT), spcname (VARCHAR), ocrname (VARCHAR), bbox (VARCHAR), centroid (VARCHAR), area (FLOAT), areapercent (FLOAT), floorplan_id (BIGINT). It has many-to-many relationships with <b>chathistory</b>, <b>objs</b>, and <b>strs</b>.</li> <li><b>chatroom</b>: Contains fields like id (BIGINT), user_id (BIGINT), name (VARCHAR), created_at (DATE). It has many-to-many relationships with <b>chathistory</b> and <b>strs</b>.</li> <li><b>chathistory</b>: Contains fields like id (BIGINT), chatroom_id (BIGINT), question (VARCHAR), answer (VARCHAR), created_at (DATE). It has many-to-many relationships with <b>chatroom</b>, <b>objs</b>, and <b>strs</b>.</li> <li><b>objs</b> and <b>strs</b>: Both contain fields like id (BIGINT), name (VARCHAR), bbox (VARCHAR), centroid (VARCHAR), area (VARCHAR). They have many-to-many relationships with <b>room</b>, <b>chatroom</b>, and <b>chathistory</b>.</li> </ul>