

# 설계

## DB 설계 문서

### Users 테이블

컬럼명	타입	제약조건 / 기본값	설명
id	BIGINT	PK, AUTO_INCREMENT	사용자 고유 ID
email	VARCHAR(255)	UNIQUE, NOT NULL	로그인 이메일 ID
password_hash	VARCHAR(255)	NOT NULL	bcrypt 해시 비밀번호
name	VARCHAR(100)	NOT NULL	이름
nickname	VARCHAR(50)	NULL	닉네임
phone	VARCHAR(20)	NULL	전화번호
role	VARCHAR(20)	NOT NULL, DEFAULT USER	권한(USER/ADMIN 등)
group_id	BIGINT	FK → user_groups.id, NULL	사용자 그룹
is_active	BOOLEAN	DEFAULT TRUE	계정 활성 여부
social_provider	VARCHAR(20)	NULL	LOCAL / GOOGLE / KAKAO / NAVER
social_id	VARCHAR(255)	NULL	소셜 로그인 ID
created_at	TIMESTAMP	DEFAULT now	가입일
updated_at	TIMESTAMP	on update now	수정일
last_login_at	TIMESTAMP	NULL	최근 로그인 시각

### Login History 테이블

컬럼명	타입	제약조건 / 기본값	설명
id	BIGINT	PK, AUTO_INCREMENT	로그인 이력 ID
user_id	BIGINT	FK → users.id	사용자 ID
login_at	TIMESTAMP	DEFAULT now	로그인 시도 시각
is_success	BOOLEAN	TRUE/FALSE	로그인 성공 여부
ip_address	VARCHAR(45)	NULL	접속 IP(IPv6 포함)
user_agent	TEXT	NULL	브라우저/클라이언트 정보

컬럼명	타입	제약조건 / 기본값	설명
device_info	VARCHAR(255)	NULL	디바이스 정보

## User Groups 테이블

컬럼명	타입	제약조건 / 기본값	설명
id	BIGINT	PK, AUTO_INCREMENT	그룹 ID
key	VARCHAR(50)	UNIQUE	그룹 키(USER/ADMIN 등)
name	VARCHAR(50)	NULL	그룹 이름
description	VARCHAR(255)	NULL	그룹 설명
is_admin	BOOLEAN	DEFAULT FALSE	관리자 그룹 여부
is_default	BOOLEAN	DEFAULT FALSE	기본 배정 그룹 여부
created_at	TIMESTAMP	DEFAULT now	생성 시각

## 주요 CRUD 함수 (backend/crud/user.py)

함수명	설명
get_user_by_email	이메일로 사용자 조회(로그인 시 사용자 로드)
authenticate_user	이메일 + 비밀번호 검증(bcrypt 해시 비교)
create_user	회원가입 처리, 비밀번호 bcrypt 해싱, 닉네임/전화/소셜/그룹 저장이메일 중복 시 <code>ValueError("EMAIL_ALREADY_EXISTS")</code> 발생 → 409 매핑 권장
update_user	사용자 정보 수정(name, nickname, phone, password_hash, is_active, group_id 등)
update_user_login_timestamp	로그인 성공 시 last_login_at 업데이트
create_login_history	로그인 시도 결과 저장(IP, UA, device, 성공여부 포함)
login_and_issue_tokens	로그인 전체 흐름: 인증 → 토큰 발급(access/refresh) → last_login_at 업데이트 → 로그인 이력 저장

## 스키마(요청/응답)

UserCreate 요청: email, name, password(평문 입력; 서버에서 해싱), nickname?, phone?, social\_provider?, social\_id?, group\_id?

UserUpdate 요청: name?, nickname?, phone?, password?(해싱), is\_active?, group\_id?

UserResponse 응답: id, email, name, nickname, phone, role, group\_id, is\_active, social\_provider/social\_id, created\_at/updated\_at/last\_login\_at

LoginHistoryCreate 요청: user\_id, ip\_address?, user\_agent?, device\_info?,

is\_success

LoginHistoryResponse 응답: id, user\_id, login\_at, ip\_address, user\_agent, device\_info, is\_success

## 권한/관리자 관련

그룹 테이블: user\_groups(backend/models/group.py)에 USER/ADMIN 등 구분 가능. users.group\_id로 FK 연결.

일반/관리자 구분: user\_groups.key(예: USER/ADMIN) + users.role(기본 USER) 두 군데에서 구분 가능, 관리자 플래그는 is\_admin에도 있음.

최초 관리자: backend/seed\_admin.py 실행 시 USER/ADMIN 그룹을 만들고, admin 계정을 생성합니다(기본 [admin@example.com](mailto:admin@example.com)/admin1234, 인자 변경 가능).