

데이터 수집 및 저장

데이터베이스 설계서

프로젝트명	Engineering Gym (엔지니어링 체육관)
작성일	2026-01-25
버전	v1.0
작성자	Engineering Gym 개발팀

1. 프로젝트 개요

Engineering Gym은 개발자들의 엔지니어링 역량을 종합적으로 훈련할 수 있는 온라인 학습 플랫폼입니다. 사용자는 시스템 디자인, 논리적 사고, 코딩 버그 수정 등 다양한 트랙의 연습 문제를 통해 실무 역량을 체계적으로 향상시킬 수 있습니다.

주요 기능

- 사용자 회원가입 및 프로필 관리
- 다중 트랙 훈련 시스템 (시스템 디자인, Logic Mirror, Bug Detective)
- 단계별 문제 풀이 (Progressive Problems)
- 학습 진행 현황 추적 및 대시보드
- 배지 및 리더보드 게임화 요소
- 공지사항 및 커뮤니티 기능
- 학습 경로 추천 시스템

2. ERD (Entity Relationship Diagram)

Engineering Gym 의 데이터베이스는 확장 가능한 구조로 설계되었으며, BaseModel 패턴을 통해 모든 테이블에서 일관된 감사 추적 기능을 제공합니다.

핵심 엔티티 관계

- users ↔ user_details (1:1 식별 관계)
- tracks → practices (1:N)
- practices → practice_details (1:N) - 타입별 상세 내용
- practices → practice_steps (1:N) - Progressive Problems 용
- users → user_submissions (1:N)
- users → user_practice_completion (1:N)

3. 테이블 정의서

3.1 공통 필드 (BaseModel)

모든 테이블이 상속받는 공통 감사 필드입니다. Django 의 Abstract Base Model 개념을 적용했습니다.

컬럼명	데이터 타입	설명
create_id	VARCHAR(100)	생성자 ID
update_id	VARCHAR(100)	최종 수정자 ID
create_date	TIMESTAMP	생성 시각 (자동)
update_date	TIMESTAMP	최종 수정 시각 (자동)
use_yn	VARCHAR(1)	사용 여부 (Y/N) - Soft Delete

3.2 users (사용자 기본 정보)

사용자의 기본 계정 정보를 저장하는 핵심 테이블입니다.

컬럼명	데이터 타입	NULL	키	설명
user_id	UUID	NOT NULL	PK	사용자 고유 ID
username	VARCHAR(100)	NOT NULL	UNIQUE	로그인 아이디
email	VARCHAR(255)	NOT NULL	UNIQUE	이메일 주소
password_hash	VARCHAR(255)	NOT NULL		암호화된 비밀번호
full_name	VARCHAR(200)	NULL		사용자 실명
user_nickname	VARCHAR(100)	NULL		닉네임
birth_date	DATE	NULL		생년월일
profile_image_url	TEXT	NULL		프로필 이미지 URL
experience_level	VARCHAR(50)	NULL		경험 수준
total_points	INTEGER	NULL		누적 포인트
streak_days	INTEGER	NULL		연속 출석일
last_active_date	DATE	NULL		마지막 활동일
is_active	BOOLEAN	NULL		활성 상태

인덱스: email, username, is_active

3.3 user_details (사용자 상세 정보)

사용자의 추가 프로필 정보를 저장합니다. users 와 1:1 식별 관계입니다.

컬럼명	데이터 타입	NULL	키	설명
user_id	UUID	NOT NULL	PK, FK	users.user_id 참조
is_developer	BOOLEAN	NULL		개발자 여부
job_role	TEXT	NULL		직군
interests	TEXT[]	NULL		관심 분야 배열
language_preference	VARCHAR(10)	NULL		언어 설정
theme_preference	VARCHAR(20)	NULL		테마 설정
notification_enabled	BOOLEAN	NULL		알림 사용 여부
daily_goal_minutes	INTEGER	NULL		일일 목표 시간
preferred_tracks	TEXT[]	NULL		선호 트랙 배열

외래키: user_id → users(user_id) ON DELETE CASCADE

3.4 tracks (훈련 트랙)

시스템 디자인, Logic Mirror, Bug Detective 등 훈련 트랙의 메타데이터를 저장합니다.

컬럼명	데이터 타입	NULL	키	설명
track_id	UUID	NOT NULL	PK	트랙 고유 ID
track_code	VARCHAR(50)	NOT NULL	UNIQUE	트랙 코드
track_name	VARCHAR(100)	NOT NULL		트랙명
description	TEXT	NULL		트랙 설명
icon_emoji	VARCHAR(10)	NULL		아이콘 이모지
color_code	VARCHAR(7)	NULL		HEX 컬러
is_active	BOOLEAN	NULL		활성 상태
display_order	INTEGER	NULL		표시 순서

3.5 practices (연습 문제 마스터)

모든 트랙의 연습 문제 메타데이터를 통합 관리하는 마스터 테이블입니다. Practice-PracticeDetail 패턴을 적용했습니다.

컬럼명	데이터 타입	NULL	키	설명
practice_id	UUID	NOT NULL	PK	문제 고유 ID
track_id	UUID	NOT NULL	FK	tracks.track_id 참조
category_id	UUID	NULL	FK	카테고리 ID
practice_code	VARCHAR(100)	NOT NULL	UNIQUE	문제 코드
unit_number	INTEGER	NULL		유닛 번호
level	INTEGER	NULL		난이도 (1-5)
title	VARCHAR(255)	NOT NULL		문제 제목
subtitle	VARCHAR(255)	NULL		부제목
emoji	VARCHAR(10)	NULL		이모지
color_code	VARCHAR(7)	NULL		HEX 컬러
icon_name	VARCHAR(100)	NULL		아이콘명
participant_count	INTEGER	NULL		참여자 수
estimated_time_minutes	INTEGER	NULL		예상 소요 시간
display_order	INTEGER	NULL		표시 순서
is_active	BOOLEAN	NULL		활성 상태
is_published	BOOLEAN	NULL		공개 여부
tags	TEXT[]	NULL		태그 배열

인덱스: track_id, level, is_published, tags (GIN)

3.6 practice_details (문제 상세 내용)

한 Practice 에 여러 타입의 상세 내용을 첨부할 수 있는 1:N 관계 테이블입니다. 타입별로 유연한 JSONB 구조를 사용합니다.

컬럼명	데이터 타입	NULL	키	설명
detail_id	UUID	NOT NULL	PK	상세 정보 ID
practice_id	UUID	NOT NULL	FK	practices.practice_id 참조
detail_type	VARCHAR(50)	NOT NULL		타입 (PROBLEM/CHATTING/REFERENCE/SOLUTION)
detail_title	VARCHAR(255)	NULL		상세 제목
content_data	JSONB	NOT NULL		타입별 상세 데이터
display_order	INTEGER	NULL		표시 순서
is_active	BOOLEAN	NULL		활성 상태

detail_type 종류:

- PROBLEM: 문제 설명 및 요구사항
- CHATTING: 대화형 콘텐츠 (AI 대화 등)
- REFERENCE: 참고 자료 (문서, 링크 등)
- SOLUTION: 해설 및 모범 답안

4. 데이터 수집 방법

4.1 사용자 데이터

항목	내용
수집 시점	회원가입 시 및 프로필 설정 시
수집 방법	프론트엔드 폼을 통한 사용자 직접 입력
수집 항목	이메일, 사용자명, 닉네임, 생년월일, 비밀번호
추가 수집	개발자 여부, 직군, 관심분야, 선호 트랙

4.2 연습 문제 데이터 (AI 기반 자동 생성)

연습 문제 콘텐츠는 대규모 언어 모델(LLM, Large Language Model)을 활용한 프롬프트 엔지니어링을 통해 구조화된 JSON 형식으로 자동 생성되었습니다. 생성된 데이터는 품질 검증 및 전문가 리뷰를 거쳐 정적 JSON 파일로 저장되며, 데이터베이스에는 메타데이터만 저장됩니다.

AI 기반 문제 생성 프로세스

- 프롬프트 설계: 교육 목표, 난이도, 트랙 특성을 반영한 구조화된 프롬프트 작성
- LLM 생성: Claude 3.5 Sonnet 모델을 활용한 문제 콘텐츠 자동 생성
- 품질 검증: 생성된 문제의 논리적 정합성, 난이도 적절성, 교육적 가치 검토
- 데이터 정제: JSON 스키마 검증 및 표준화된 형식으로 변환
- 전문가 리뷰: 도메인 전문가의 최종 검수 및 피드백 반영

데이터 파일	트랙	설명	수량
test.json	시스템 디자인	Instagram, YouTube 등 대규모 시스템 설계	10 개
progressive-problems.json	Bug Detective	ML 파이프라인, 마케팅 분석 버그	3 개 프로젝트
stages.js	Logic Mirror	순차/선택/반복 논리 훈련 퀘스트	11 개 퀘스트

5. 데이터 저장 전략

5.1 데이터베이스 선택

항목	선택
DBMS	PostgreSQL (운영) / SQLite (개발)
ORM	Django ORM
마이그레이션	Django Migrations
버전 관리	Git (스키마 변경 이력 추적)

5.2 Soft Delete 전략

모든 테이블에 `use_yn` 컬럼을 두어 논리적 삭제를 구현합니다. 데이터 복구가 가능하며 감사 추적에 유리합니다.

- 삭제 시: `use_yn = 'N'`으로 업데이트
- 조회 시: `WHERE use_yn = 'Y'` 조건 추가
- 복구 시: `use_yn`을 다시 'Y'로 변경

6. 보안 고려사항

6.1 비밀번호 암호화

항목	내용
암호화 알고리즘	PBKDF2 (Django 기본) 또는 bcrypt/argon2
함수	django.contrib.auth.hashers.make_password()
정책	평문 비밀번호 저장 절대 금지, 최소 8자 이상
해싱 라운드	PBKDF2: 최소 100,000 iterations

6.2 개인정보 보호

- 이메일 중복 확인 시 정보 노출 최소화
- 민감 정보 로깅 금지 (비밀번호, 세션 토큰 등)
- API 응답에 password_hash 필드 제외
- GDPR 준수: 사용자 데이터 삭제 요청 처리

6.3 접근 제어

항목	방법
인증	JWT 토큰 기반 인증
권한 검증	API 엔드포인트별 데코레이터/미들웨어 적용
세션 관리	Access Token (15 분) + Refresh Token (7 일)
CSRF 방어	Django CSRF 토큰 활용

7. 부록: 공통 코드 목록

7.1 직군 코드 (JOB_ROLE)

top_code	code	code_name	display_order
JOB_ROLE	FRONTEND	프론트엔드 개발자	10
JOB_ROLE	BACKEND	백엔드 개발자	20
JOB_ROLE	FULLSTACK	풀스택 개발자	30
JOB_ROLE	DATA	데이터 엔지니어	40
JOB_ROLE	DEVOPS	DevOps 엔지니어	50
JOB_ROLE	MOBILE	모바일 개발자	60

7.2 논리 유형 코드 (LOGIC_TYPE)

top_code	code	code_name	display_order
LOGIC_TYPE	SEQUENCE	순차	10
LOGIC_TYPE	SELECTION	선택	20
LOGIC_TYPE	ITERATION	반복	30
LOGIC_TYPE	LIST	리스트 순회	40
LOGIC_TYPE	NESTED	중첩 조건	50

8. 변경 이력

버전	날짜	변경 내용	작성자
v1.0	2026-01-25	최초 작성 - Engineering Gym ERD 설계	개발팀

본 문서는 Engineering Gym 프로젝트의 데이터베이스 설계를 기술한 문서입니다