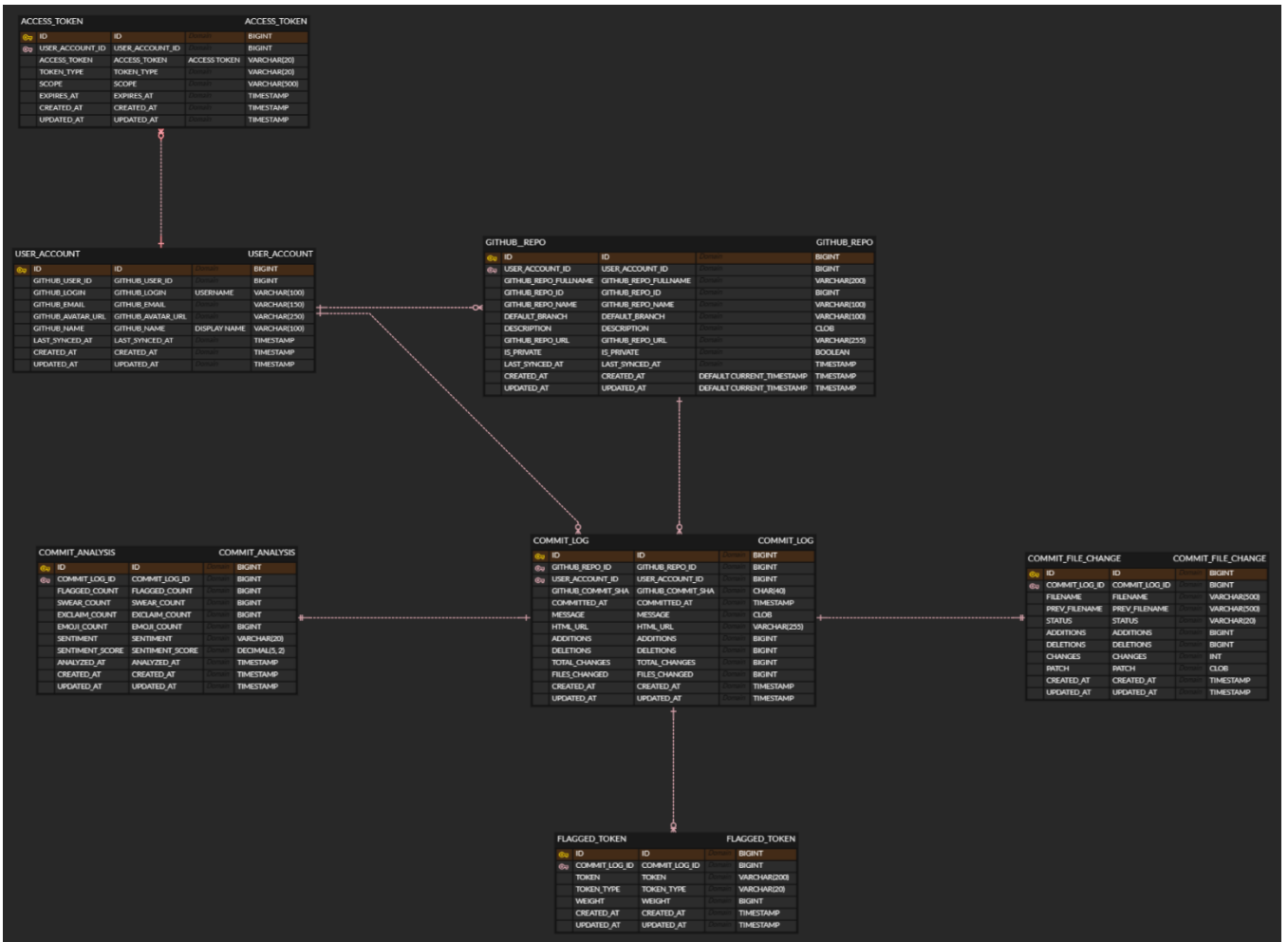




ER 다이어그램

#02_workspace/03_ssafy/commitmood







목차

메모

본 문서는 CommitMood 서비스의 데이터베이스 구조를 ER(Entity-Relationship) 다이어그램 관점에서 정의합니다.

실제 MySQL 스키마(DDL)를 기준으로 각 엔티티와 관계를 명확히 표현하는 것을 목적으로 합니다.

- 📦 ER 다이어그램 개요
- 👤 사용자 및 인증 엔티티
- 📁 GitHub 리포지토리 엔티티

-  커밋 데이터 엔티티
-  분석 및 키워드 엔티티
-  엔티티 관계 요약
-  ER 다이어그램 해설

ER 다이어그램 개요

메모

CommitMood의 ER 다이어그램은 구현된 데이터베이스 스키마와 1:1로 대응되며, 서비스의 핵심 데이터 흐름과 저장 구조를 설명하기 위한 기준 문서로 활용됩니다.

- GitHub OAuth 기반 사용자 인증 구조 표현
- 사용자-리포지토리-커밋 중심의 데이터 계층 구조 정의
- 커밋 분석 및 감정 키워드 데이터의 종속 관계 명시
- 배치 및 분석 처리 확장을 고려한 설계 구조

사용자 및 인증 엔티티

USER_ACCOUNT

- **설명**
 - CommitMood 서비스를 이용하는 사용자를 표현하는 핵심 엔티티입니다.
 - GitHub 사용자 식별자와 서비스 내부 사용자 정보를 함께 관리합니다.
- **주요 컬럼**
 - ID (PK)
 - GITHUB_USER_ID (GitHub 사용자 고유 ID, Unique)
 - GITHUB_LOGIN
 - GITHUB_EMAIL
 - GITHUB_AVATAR_URL
 - GITHUB_NAME
 - LAST_SYNCED_AT
 - CREATED_AT

- UPDATED_AT



ACCESS_TOKEN

- **설명**
 - GitHub OAuth 인증을 통해 발급된 Access Token 정보를 저장하는 엔티티입니다.
 - GitHub API 호출 시 인증 수단으로 사용됩니다.
 - **주요 컬럼**
 - ID (PK)
 - USER_ACCOUNT_ID (FK)
 - ACCESS_TOKEN
 - TOKEN_TYPE
 - SCOPE
 - EXPIRES_AT
 - CREATED_AT
 - UPDATED_AT
-



GitHub 리포지토리 엔티티



GITHUB_REPO

- **설명**
 - 사용자가 소유하거나 관리하는 GitHub 리포지토리를 표현하는 엔티티입니다.
 - 커밋 데이터 수집의 기준 단위로 사용됩니다.
- **주요 컬럼**
 - ID (PK)
 - USER_ACCOUNT_ID (FK)
 - GITHUB_REPO_ID (GitHub 리포지토리 고유 ID, Unique)
 - GITHUB_REPO_NAME
 - GITHUB_REPO_FULL_NAME
 - DEFAULT_BRANCH
 - DESCRIPTION
 - GITHUB_REPO_URL
 - IS_PRIVATE
 - LAST_SYNCED_AT
 - CREATED_AT

- UPDATED_AT
-



커밋 데이터 엔티티



COMMIT_LOG

- **설명**
 - GitHub 리포지토리에서 수집된 개별 커밋 정보를 저장하는 핵심 엔티티입니다.
 - 분석, 통계, 랭킹 산출의 최소 단위 데이터입니다.
- **주요 컬럼**
 - ID (PK)
 - GITHUB_REPO_ID (FK)
 - USER_ACCOUNT_ID (FK, 커밋 작성자)
 - GITHUB_COMMIT_SHA (리포지토리 내 Unique)
 - COMMITTED_AT
 - MESSAGE
 - HTML_URL
 - ADDITIONS
 - DELETIONS
 - TOTAL_CHANGES
 - FILES_CHANGED
 - CREATED_AT
 - UPDATED_AT



COMMIT_FILE_CHANGE

- **설명**
 - 하나의 커밋에 포함된 파일 변경 내역을 저장하는 엔티티입니다.
 - 파일 단위 변경 분석 및 상세 조회를 위해 사용됩니다.
- **주요 컬럼**
 - ID (PK)
 - COMMIT_LOG_ID (FK)
 - FILENAME
 - PREV_FILENAME
 - STATUS
 - ADDITIONS

- DELETIONS
 - CHANGES
 - PATCH
 - CREATED_AT
 - UPDATED_AT
-

분석 및 키워드 엔티티

COMMIT_ANALYSIS

- **설명**
 - 커밋 메시지를 기반으로 수행된 감정 및 키워드 분석 결과를 저장하는 엔티티입니다.
 - 각 커밋당 하나의 분석 결과만 존재합니다.
- **주요 컬럼**
 - ID (PK)
 - COMMIT_LOG_ID (FK, Unique)
 - FLAGGED_COUNT
 - SWEAR_COUNT
 - EXCLAIM_COUNT
 - EMOJI_COUNT
 - SENTIMENT
 - SENTIMENT_SCORE
 - ANALYZED_AT
 - CREATED_AT
 - UPDATED_AT

FLAGGED_TOKEN

- **설명**
 - 커밋 메시지에서 추출된 감정 키워드 및 특수 토큰 정보를 저장하는 엔티티입니다.
 - 분석 결과의 세부 구성 요소로 활용됩니다.
- **주요 컬럼**
 - ID (PK)
 - COMMIT_LOG_ID (FK)
 - TOKEN
 - TOKEN_TYPE

- WEIGHT
- CREATED_AT
- UPDATED_AT



엔티티 관계 요약

- USER_ACCOUNT는 여러 개의 ACCESS_TOKEN을 가질 수 있습니다. (1:N)
- USER_ACCOUNT는 여러 개의 GITHUB_REPO를 소유할 수 있습니다. (1:N)
- GITHUB_REPO는 여러 개의 COMMIT_LOG를 포함합니다. (1:N)
- USER_ACCOUNT는 여러 개의 COMMIT_LOG의 작성자가 될 수 있습니다. (1:N)
- COMMIT_LOG는 여러 개의 COMMIT_FILE_CHANGE를 가집니다. (1:N)
- COMMIT_LOG는 하나의 COMMIT_ANALYSIS를 가집니다. (1:1)
- COMMIT_LOG는 여러 개의 FLAGGED_TOKEN을 가질 수 있습니다. (1:N)



ER 다이어그램 해설

- 본 ER 다이어그램은 CommitLog를 중심으로 데이터가 확장되는 구조를 가집니다.
- 분석 및 파일 변경 정보는 모두 CommitLog에 종속된 부가 엔티티로 설계되었습니다.
- GitHub에서 제공하는 외부 식별자(GITHUB_USER_ID, GITHUB_REPO_ID, GITHUB_COMMIT_SHA)는 내부 PK와 분리하여 Unique 제약 조건으로 관리합니다.
- OAuth 인증 정보는 ACCESS_TOKEN 엔티티로 분리하여 보안 및 확장성을 고려했습니다.

메모

본 ER 다이어그램 문서는 클래스 다이어그램, 유스케이스 다이어그램, API 명세의 기준 문서로 활용됩니다.