

**SK네트웍스 Family AI 과정 11기  
 데이터 수집 및 저장 데이터베이스 설계문서**



| **산출물 단계** | 데이터 수집 및 저장 |
| --- | --- |
| **평가 산출물** | 데이터베이스 설계문서 |
| **제출 일자** | 2025.06.24 (화) |
| **깃허브 경로** |  |
| **작성 팀원** | 김상익, 나지윤 |

**1. 채택한 모델링 방법론: Crow’s Foot**

본 프로젝트의 데이터베이스는 엔터티 간의 관계를 시각적으로 명확하게 표현하기 위해 Crow’s Foot 모델링 방법론을 채택하여 설계되었다. 이 표기법을 통해 각 테이블 간의 식별/비식별 관계 및 1:1, 1:N, 0:N과 같은 관계 구조를 직관적으로 나타낼 수 있도록 하였다.

**2. 논리 데이터 모델**

**2.1 엔터티 목록**

**2.1.1. 엔터티명: USER**

* **설명:** 소셜 로그인 기반의 사용자 정보를 저장하는 테이블
* **식별자: PK:** user\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| user\_uuid | UUID | ✓ | UUID | 내부 사용자 고유 식별자 |
| provider\_id | VARCHAR(255) | ✓ |  | 소셜 제공자의 고유 사용자 식별자 |
| provider | VARCHAR(20) | ✓ |  | 소셜 로그인 제공자 |
| user\_name | VARCHAR(20) | ✓ |  | 사용자 이름 |
| user\_email | VARCHAR(50) | ✓ |  | 사용자 이메일 |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 사용자가 처음 가입한 시각 |
| updated\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 사용자의 정보가 마지막으로 수정된 시각 |

**2.1.2. 엔터티명: USER-GROUP**

* **설명:** 사용자와 그룹 간의 소속 관계를 관리하는 매핑 테이블
* **식별자: PK:** (user\_uuid, group\_uuid)

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| user\_uuid | UUID | ✓ | UUID | 내부 사용자 고유 식별자 |
| group\_uuid | UUID | ✓ | UUID | 그룹 고유 식별자 |

**2.1.3. 엔터티명: GROUP**

* **설명:** AI 모델을 관리하는 팀 단위의 그룹 정보를 저장하는 테이블
* **식별자: PK:** group\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| group\_uuid | UUID | ✓ | UUID | 그룹 고유 식별자 |
| group\_name | VARCHAR(100) | ✓ |  | 그룹명 |
| group\_description | TEXT |  |  | 그룹 설명 |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 그룹 정보 마지막 수정 시각 |
| updated\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 그룹 정보 생성 시각 |

**2.1.4. 엔터티명: ML**

* **설명:** 사용자별로 생성한 AI 모델 정보를 저장하는 테이블
* **식별자:** **PK:** model\_uuid, **FK:** group\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| model\_uuid | UUID | ✓ | UUID | 모델 고유 식별자 |
| group\_uuid | UUID | ✓ | UUID | 그룹 고유 식별자 |
| mbti\_id | INTEGER | ✓ | Auto\_  increment( ) | MBTI 성격 고유 식별자 |
| model\_name | VARCHAR(100) | ✓ |  | 모델 이름 |
| model\_description | TEXT |  |  | 모델 설명 |
| model\_personality | TEXT | ✓ |  | 모델 성격 |
| model\_speaks | TEXT | ✓ |  | 모델 말투 |
| model\_repo | VARCHAR(255) | ✓ |  | 허깅페이스 repo URL |
| image\_url | JSON |  |  | 대표 이미지 파일 URL |
| model\_status | TINYINT | ✓ |  | 0:’학습 중’, 1:’사용가능’ |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 모델 생성시점 |
| updated\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 모델 마지막 수정일 |

**2.1.5. 엔터티명: BOARD**

* **설명:** 사용자가 작성한 게시물의 메타데이터 및 본문 정보를 저장하는 테이블
* **식별자:** **PK:** board\_uuid, **FK:** group\_uuid, model\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| board\_uuid | UUID | ✓ | UUID | 게시물 고유 식별자 |
| group\_uuid | UUID | ✓ | UUID | 그룹 고유 식별자 |
| model\_uuid | UUID | ✓ | UUID | 모델 고유 식별자 |
| board\_topic | VARCHAR(255) | ✓ |  | 게시물 주제 또는 카테고리명 |
| board\_description | TEXT |  |  | 게시물 상세 설명 |
| board\_platform | TINYINT | ✓ |  | 0:’인스타그램’, 1:’블로그’, 2:’페이스북’ |
| board\_hash\_tag | JSON |  |  | JSON 배열 형태로 저장된 해시태그 리스트 |
| board\_status | TIMYINT | ✓ | 0 | 0:’NEW’, 1:’임시저장’, 2:’예약’, 3:’발행됨’ |
| image\_url | JSON | ✓ |  | 게시물 이미지 URL 경로 |
| reservation\_at | TIMESTAMP |  |  | 게시물 예약 발행 일시 |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 게시물 생성 시각 |
| updated\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 게시물 수정 시각 |
| published\_at | TIMESTAMP |  |  | 게시물 발행 시각 |

**2.1.6. 엔터티명: ML\_MBTI**

* **설명:** 모델에 부여가능한 MBTI 유형 정보를 저장하는 테이블
* **식별자:** **PK:** mbti\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| mbti\_id | INTEGER | ✓ | Auto\_  increment( ) | MBTI 성격 고유 식별자 |
| mbtiI\_name | VARCHAR(100) | ✓ |  | MBTI 명칭 |
| mbti\_chara | VARCHAR(255) | ✓ |  | MBTI 성격 및 특성 |
| mbti\_speaks | TEXT | ✓ |  | MBTI 말투 |

**2.1.7. 엔터티명: SYSTEM\_LOG**

* **설명:** 사용자 활동 및 시스템 내 이벤트 기록을 저장하는 테이블
* **식별자:** **PK:** log\_uuid, **FK:** user\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| log\_uuid | UUID | ✓ | UUID | 시스템 로그 고유 식별자 |
| user\_id | UUID |  | UUID | 내부 사용자 고유 식별자 |
| log\_type | TINYINT | ✓ |  | 0:’API요청’, 1:’시스템오류’, 2:’인증관련’ |
| log\_content | JSON | ✓ |  | API 요청 내용, 오류 메시지 등 상세한 로그 내용 |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 로그 발생 시각 |

**2.1.8. 엔터티명: HF\_TOKEN\_MANAGE**

* **설명:** 각 그룹에 허깅페이스 토큰 정보를 저장하고 관리하는 테이블
* **식별자:** **PK:** hf\_manage\_uuid, **FK:** group\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| hf\_manage\_uuid | UUID | ✓ | UUID | 권한 고유 식별자 |
| group\_uuid | UUID | ✓ | UUID | 그룹 고유 식별자 |
| hf\_token\_value | TEXT | ✓ |  | 허깅페이스 실제 토큰 값 (비공개) |
| hf\_token\_nickname | VARCHAR(100) | ✓ |  | 사용자에게 보여지는 허깅페이스 토큰 별칭 |
| hf\_user\_name | VARCHAR(50) | ✓ |  | 허깅페이스 계정 사용자 이름 |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 허깅페이스 토큰 생성 시각 |
| updated\_at | TIMESTAMP |  | Current\_  timestamp( ) | 허깅페이스 토큰 마지막 수정 시각 |

**2.1.9. 엔터티명: ML\_API**

* **설명:** 그룹이 생성한 AI 모델에 대해 발급된 API Key와 관련 정보를 저장하는 테이블
* **식별자:** **PK:** api\_uuid, group\_uuid, **FK:** group\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| api\_uuid | UUID | ✓ | UUID | API 고유 식별자 |
| model\_uuid | UUID | ✓ | UUID | 모델 고유 식별자 |
| group\_uuid | UUID | ✓ | UUID | 그룹 고유 식별자 |
| api\_key | VARCHAR(255) | ✓ |  | 발급된 API key |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | API 발급 일시 |
| updated\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | API 발급정보 마지막 수정 일시 |

**2.1.10. 엔터티명: API\_CALL\_AGGREGATION**

* **설명:** API키별 일별 호출 횟수를 집계하여 저장하는 테이블
* **식별자:** **PK:** api\_call\_uuid, **FK:** api\_uuid, model\_uuid, user\_uuid

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| api\_call\_uuid | UUID | ✓ | UUID | 집계 데이터 고유 식별자 |
| group\_uuid | UUID | ✓ | UUID | 그룹 고유 식별자 |
| model\_uuid | UUID | ✓ | UUID | 모델 고유 식별자 |
| api\_uuid | UUID | ✓ | UUID | API 고유 식별자 |
| daily\_call\_count | INT | ✓ | 0 | 하루동안 호출된 총 횟수 |
| created\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 호출된 집계 일시 |
| updated\_at | TIMESTAMP | ✓ | Current\_  timestamp( ) | 호출된 집계 변경 일시 |

**2.2 엔터티 간 관계**

**2.2.1. 관계명: ‘USER’ - ’USER-GROUP’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER | USER-GROUP | 1:N | 한 사용자는 여러 개의 그룹에 소속될 수 있음 |

**2.2.2. 관계명: ‘GROUP’ - ‘USER-GROUP’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | USER-GROUP | 1:N | 한 그룹에는 여러 사용자가 소속될 수 있음 |

**2.2.3. 관계명: ‘GROUP’ - ‘HF\_TOKEN\_MANAGE’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | HF\_TOKEN\_  MANAGE | 1:N | 하나의 그룹은 여러 개의 허깅페이스 토큰을 부여받을 수 있음 |

**2.2.4. 관계명: ‘GROUP’ - ‘ML’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | ML | 1:N | 하나의 그룹은 여러 개의 AI 모델을 가질 수 있음 |

**2.2.5. 관계명: ‘USER’ - ‘SYSTEM\_LOG’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER | SYSTEM\_LOG | 1:N | 사용자 없이도 로그가 기록될 수 있음 |

**2.2.6. 관계명: ‘ML’ - ‘BOARD’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| ML | BOARD | 1:N | 하나의 모델은 여러 개의 게시물과 연결될 수 있음 |

**2.2.7. 관계명: ‘ML’ - ‘ML\_API’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| ML | ML\_API | 1:1 | 하나의 모델은 하나의 API만 가질 수 있음 |

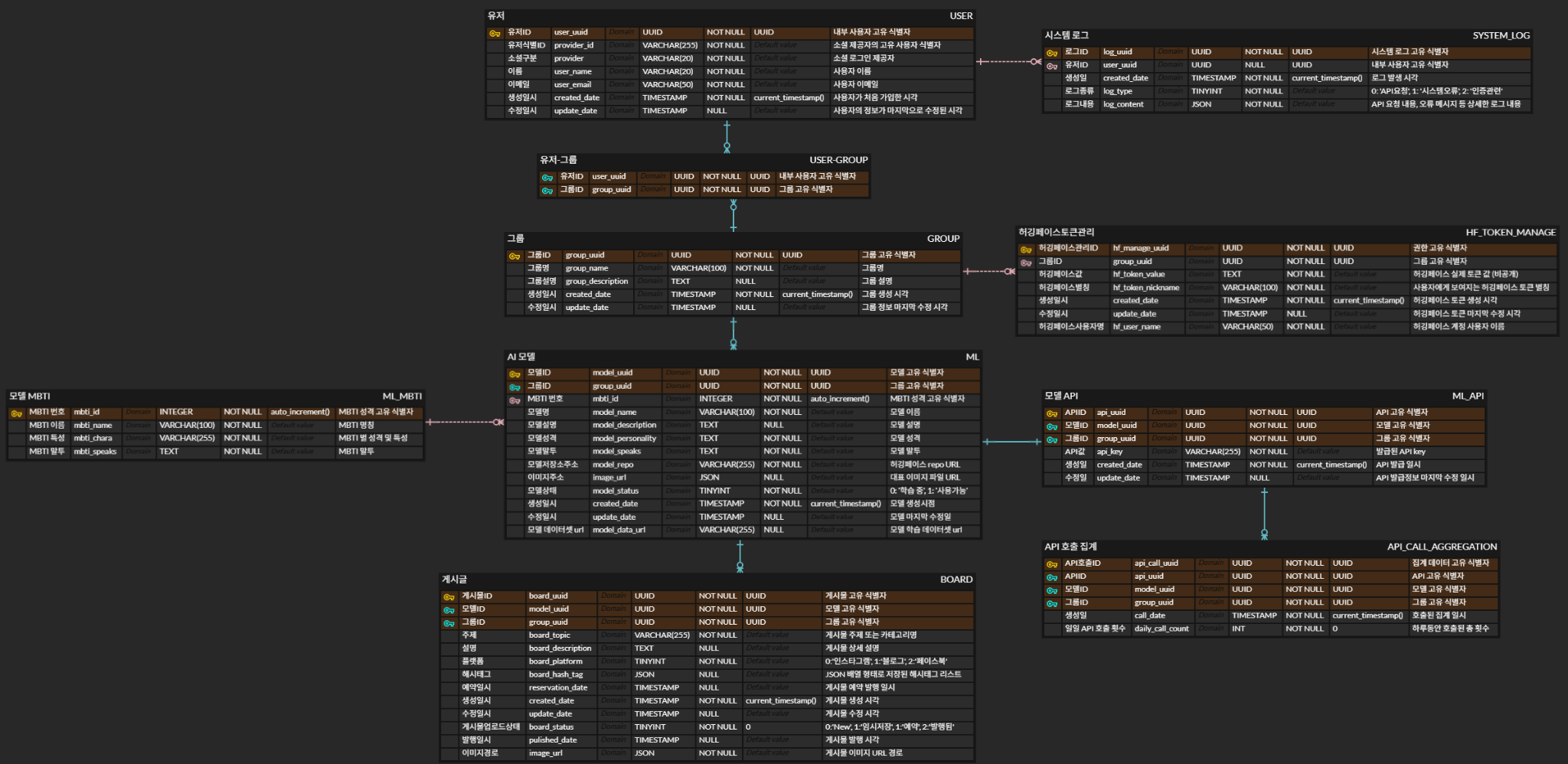
**2.2.8. 관계명: ‘ML’ - ‘ML\_MBTI’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| ML | ML\_MBTI | 1:1 | 하나의 모델은 하나의 MBTI만 가질 수 있음 |

**2.2.9. 관계명: ‘ML\_API’ - ‘API\_CALL\_AGGREGATION’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| ML\_API | API\_CALL\_  AGGREGATION | 1:N | 하나의 API는 하루 여러 번 호출되어 집계될 수 있음 |

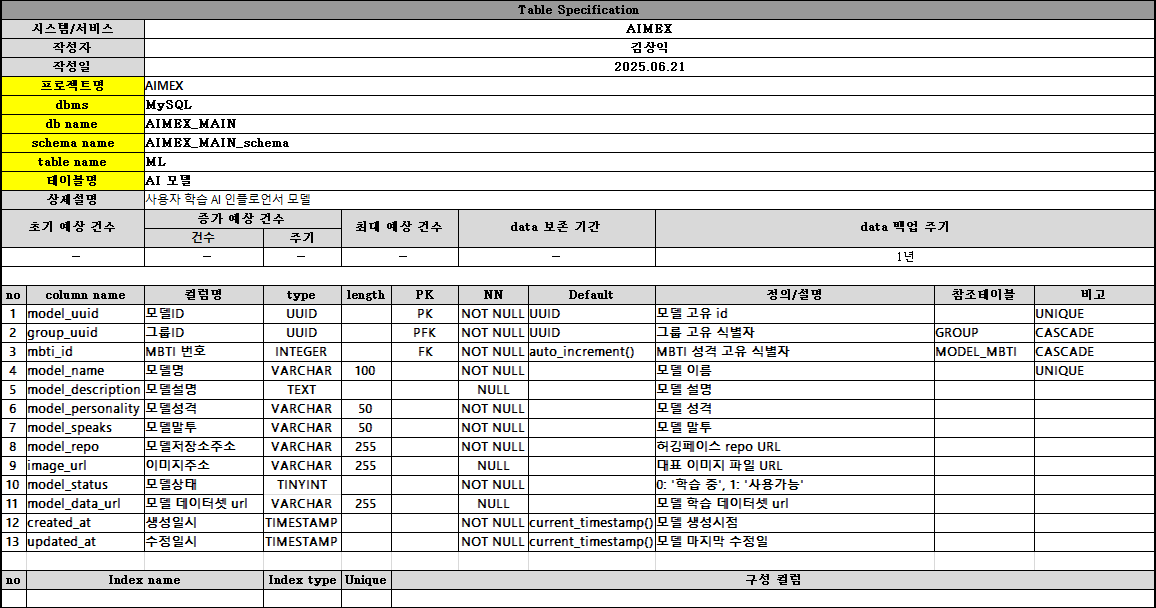
**2.3 ERD**



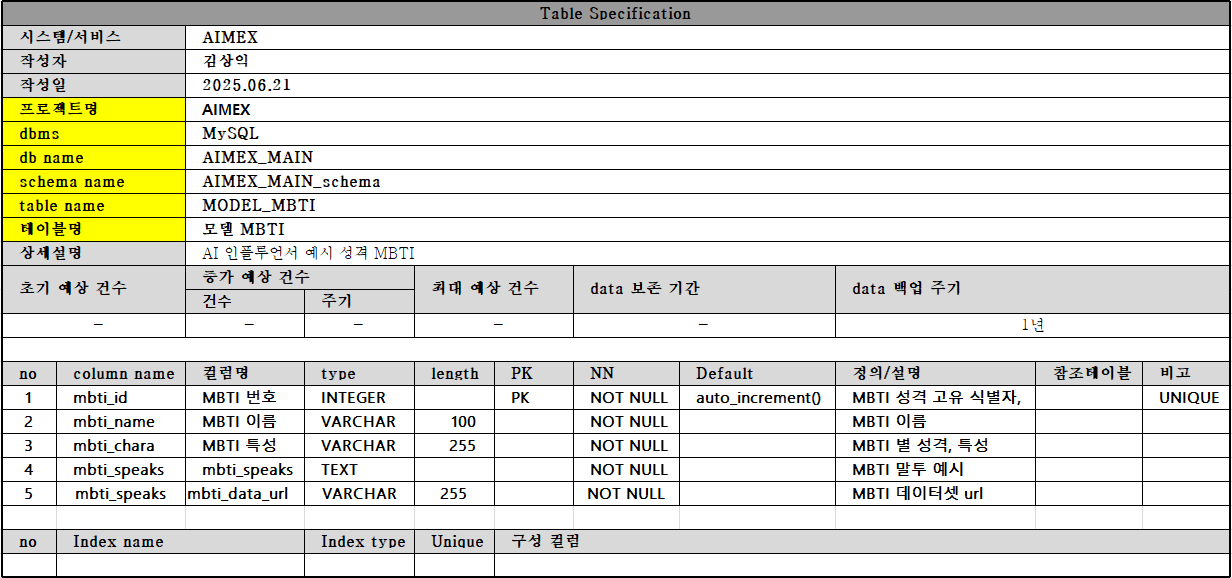
**3. 물리 데이터 모델**

**3.1 테이블 정의서**

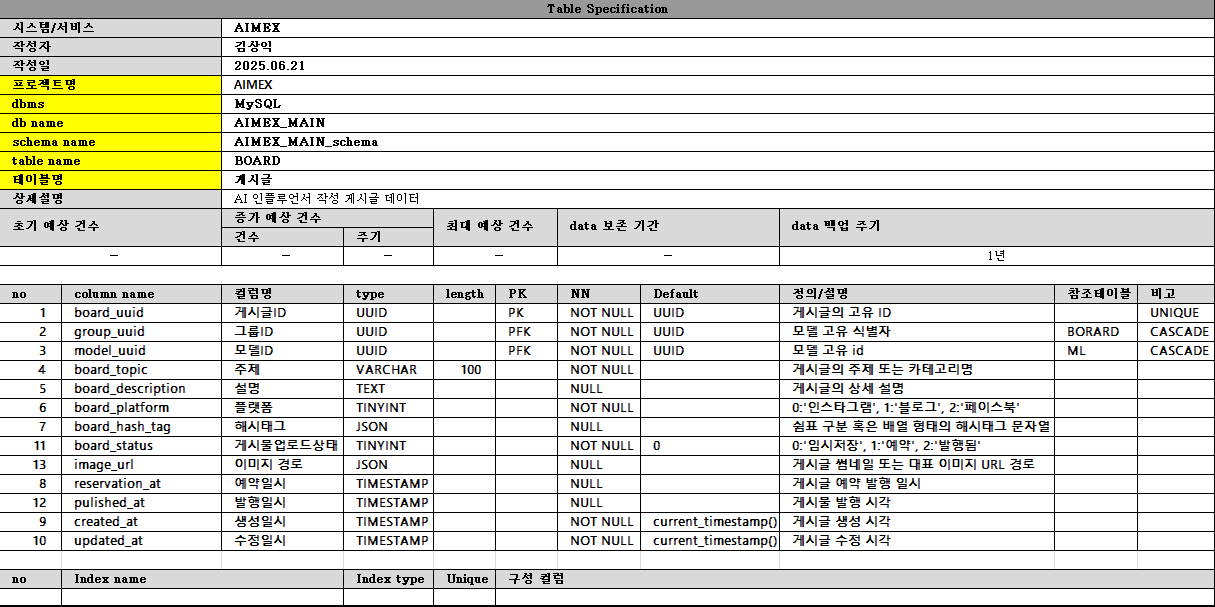
**3.1.1 AI 모델 테이블 명세서**

****

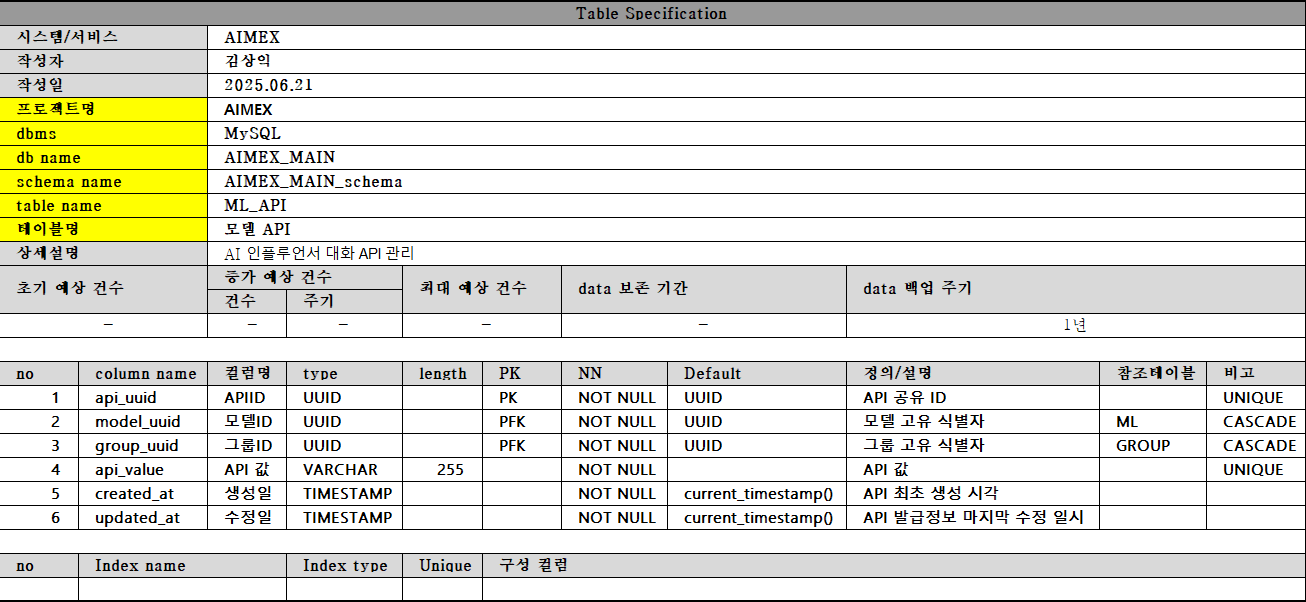
**3.1.2 모델 MBTI 테이블 명세서**



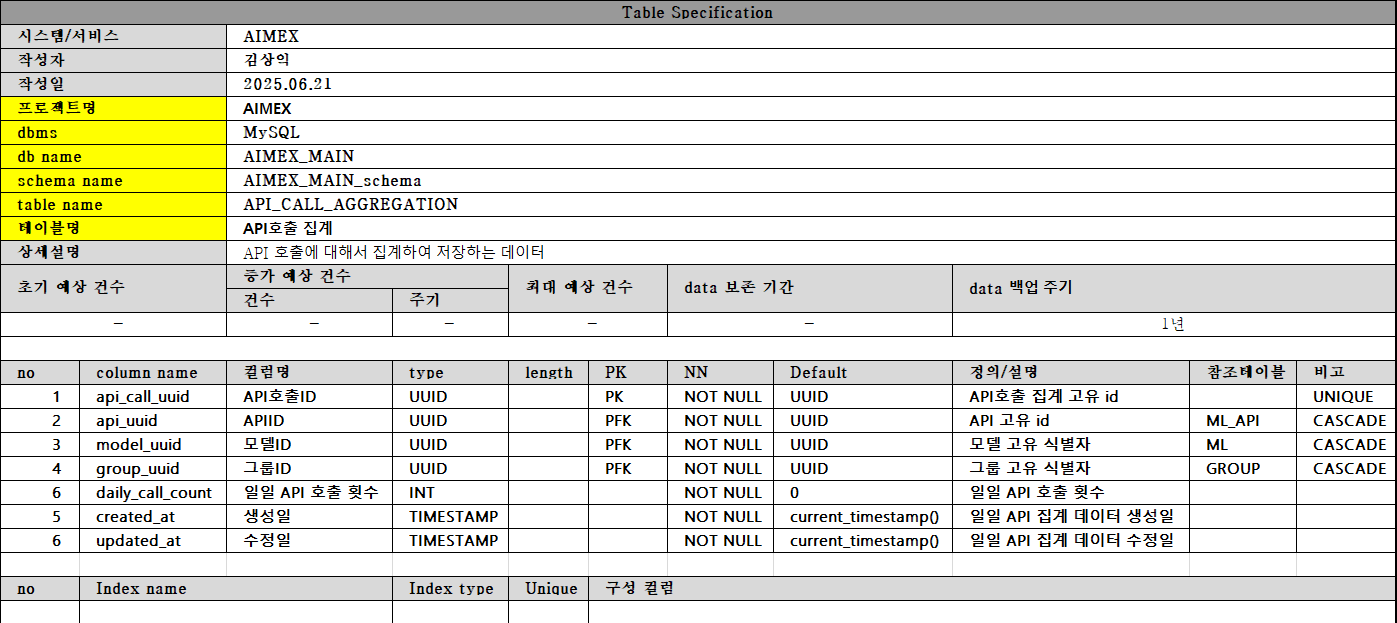
**3.1.3 게시글 테이블 명세서**



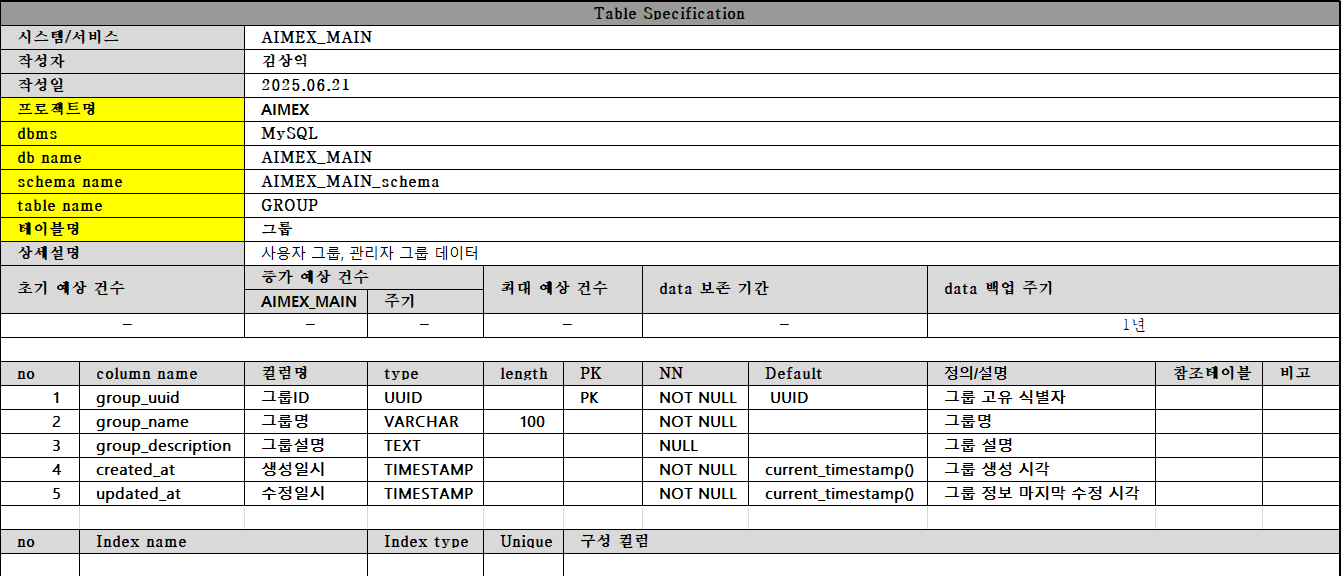
**3.1.4 모델 API 테이블 명세서**



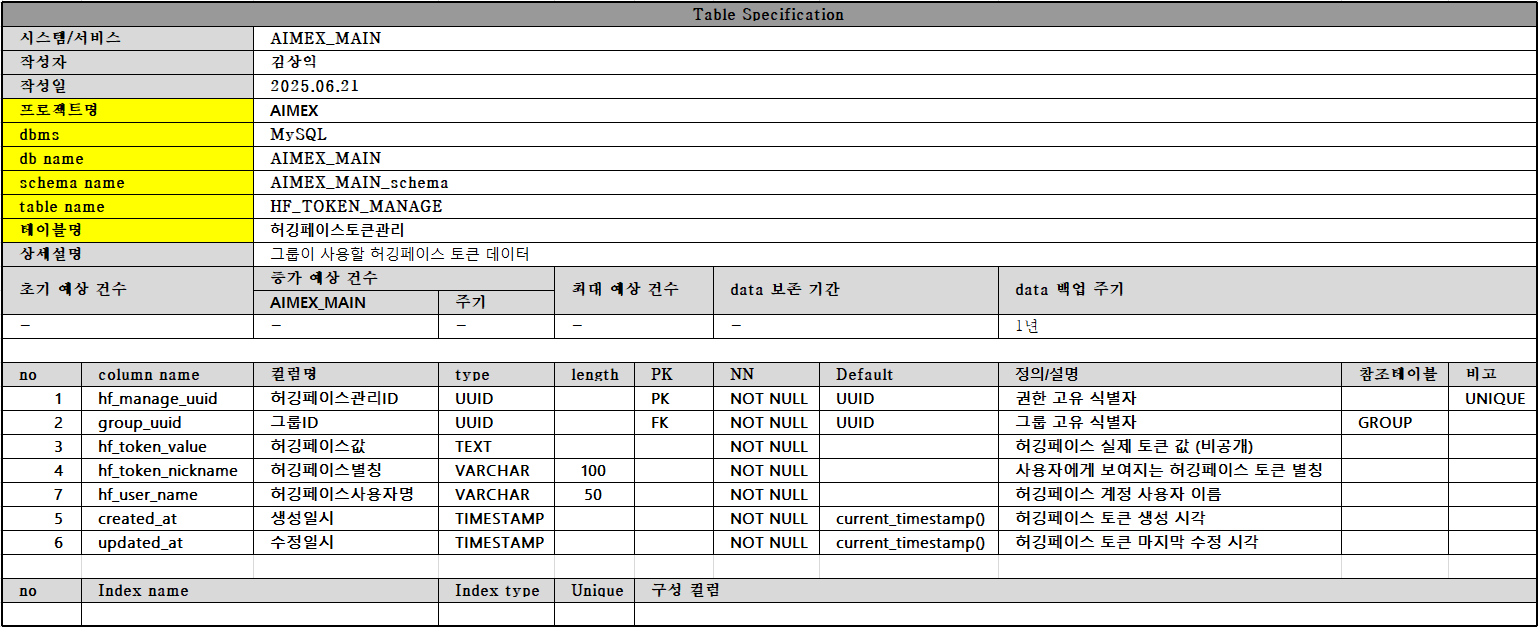
**3.1.5 API 호출 집계 테이블 명세서**



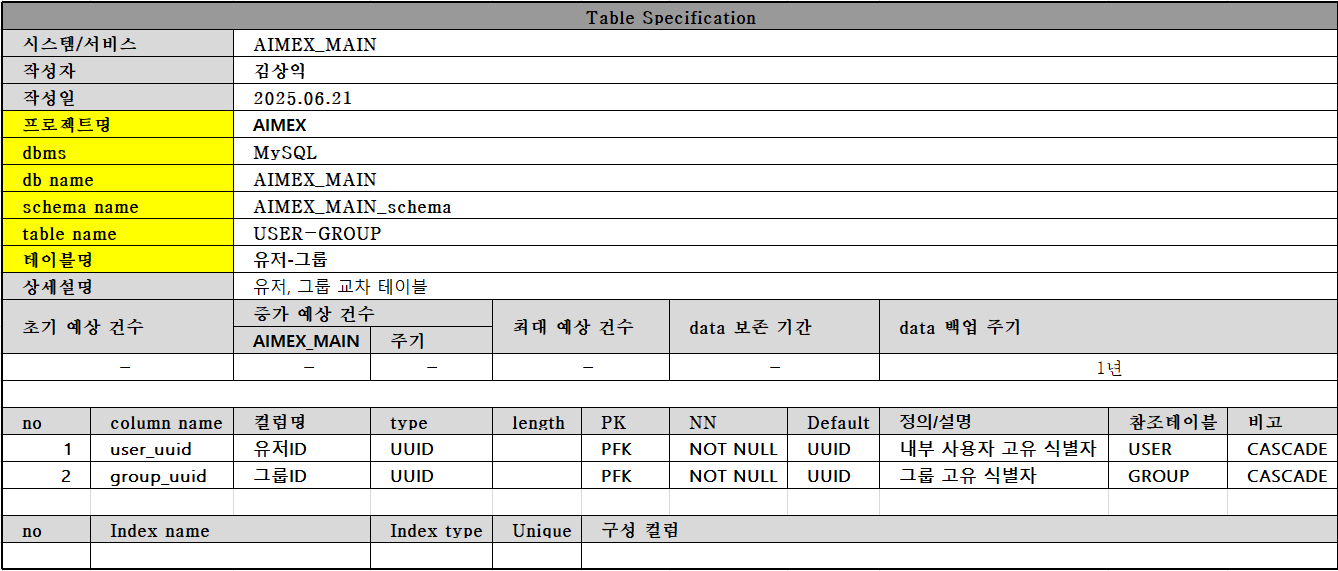
**3.1.6 그룹 테이블 명세서**



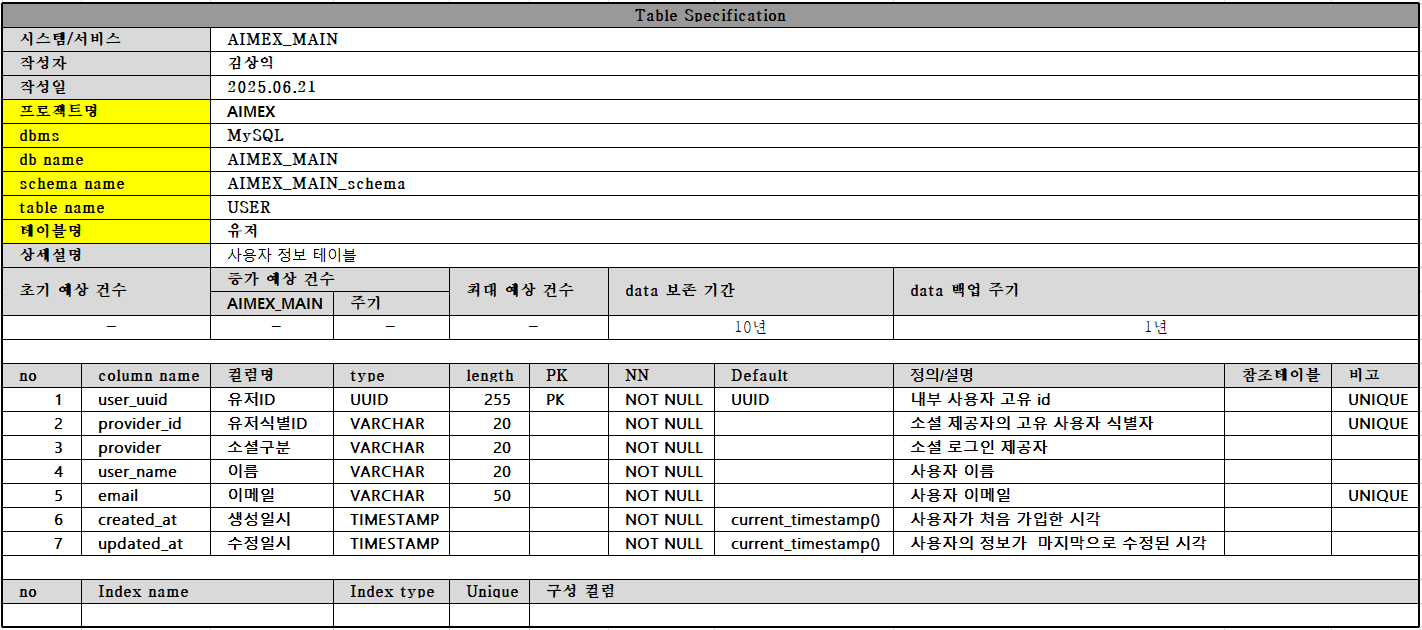
**3.1.7 허깅페이스토큰관리 테이블 명세서**

****

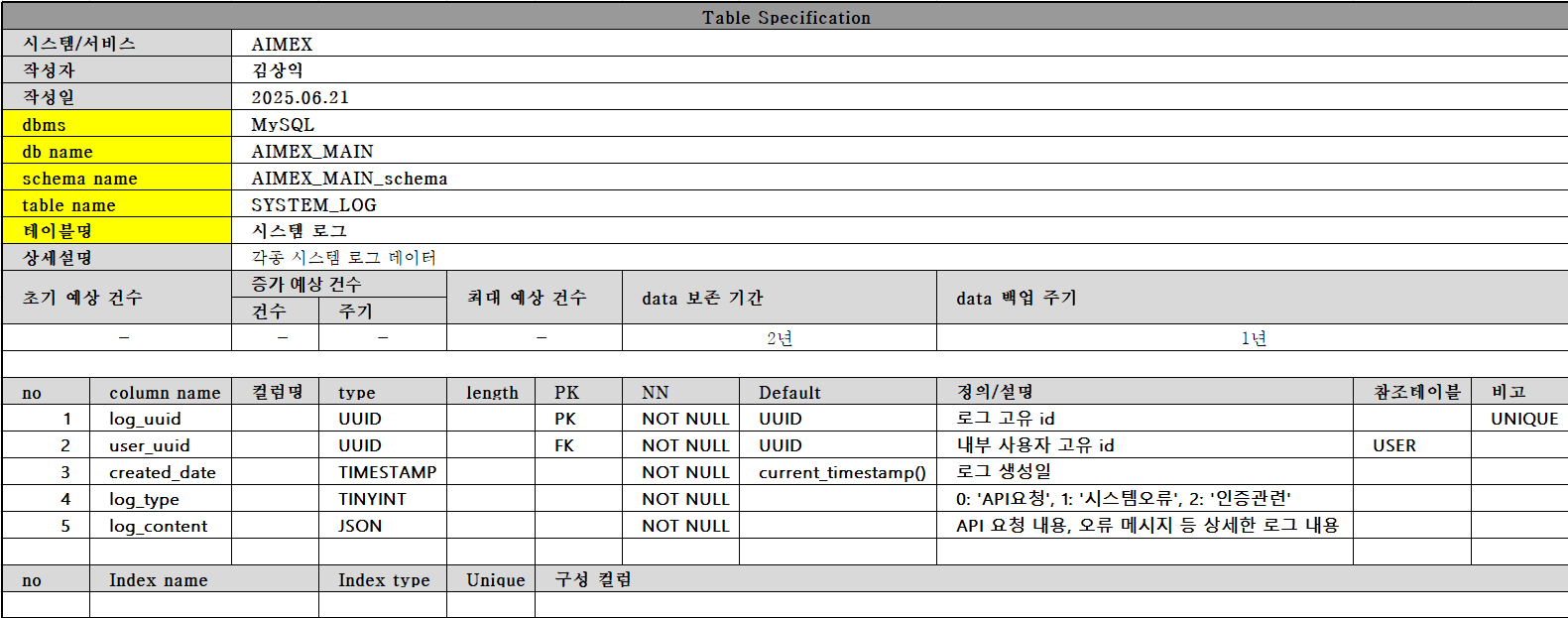
**3.1.8 유저-그룹 교차 테이블 명세서**

****

**3.1.9 유저 테이블 명세서**

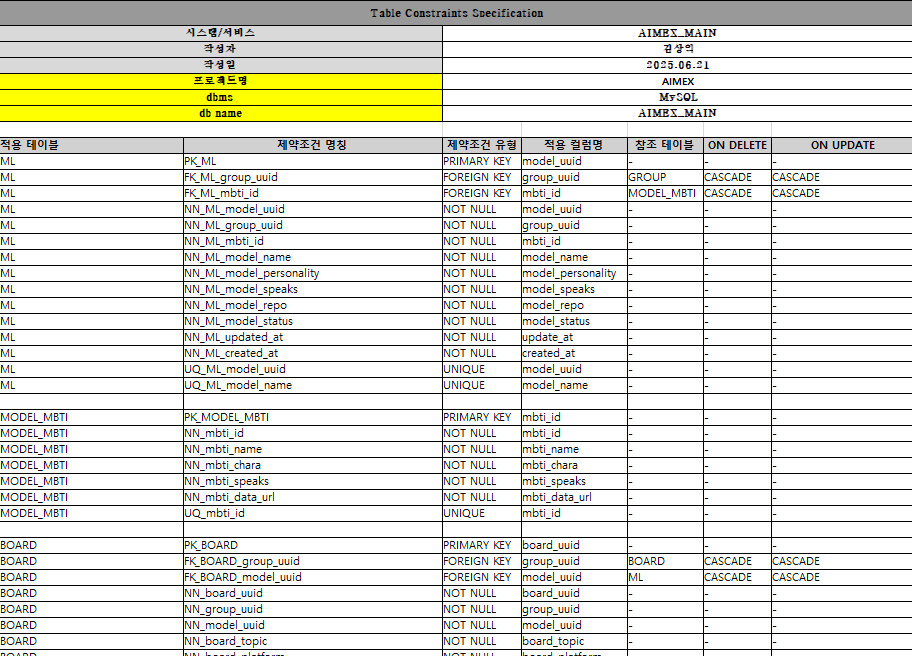
****

**3.1.10 시스템로그 테이블 명세서**

****

**3.2 제약 조건 명세**

**데이터베이스 제약조건 명세서**

****

**4. 데이터 정합성 및 무결성 관리 방안**

**4.1 중복 방지 전략**

| **주 엔터티** | **속성명** | **데이터 타입** | **중복방지 여부** | **설명** |
| --- | --- | --- | --- | --- |
| ML | model\_uuid | UUID | UNIQUE | 모델 데이터 고유 식별자 중복 불가 |
| ML | model\_name | VARCHAR | UNIQUE | 모델 이름 중복 불가 |
| MODEL\_MBTI | mbti\_id | INTEGER | UNIQUE | 모델 MBTI 고유 식별자 중복 불가 |
| ML\_API | api\_uuid | UUID | UNIQUE | 모델 API 데이터 고유 식별자 중복 불가 |
| ML\_API | api\_value | VARCHAR | UNIQUE | 모델 API 실제 값 중복 불가 |
| API\_CALL\_AGGREGATION | api\_call\_uuid | UUID | UNIQUE | 모델 API 호출 집계 데이터 고유 식별자 중복 불가 |
| GROUP | group\_uuid | UUID | UNIQUE | 그룹 데이터 고유 식별자 중복 불가 |
| GROUP | group\_name | VARCHAR | UNIQUE | 그룹 이름 중복 불가 |
| HF\_TOKEN\_MANAGE | hf\_manage\_uuid | UUID | UNIQUE | 허깅페이스토큰관리 고유 식별자 중복불가 |
| USER | user\_uuid | UUID | UNIQUE | 유저 데이터 고유 식별자 중복 불가 |
| USER | provider\_id | VRACHAR | UNIQUE | 유저 프로바이더 아이디 중복 불가 |
| USER | email | VRACHAR | UNIQUE | 유저 이메일 중복불가 |
| BOARD | board\_uuid | UUID | UNIQUE | 게시물 데이터 식별자 중복 불가 |
| SYSTEM\_LOG | log\_uuid | UUID | UNIQUE | 시스템 로그 데이터 식별자 중복 불가 |

**4.2 정규화 수준**

| **주 엔터티** | **1NF** | **2NF** | **3NF** | **정규형** |
| --- | --- | --- | --- | --- |
| ML | 각 필드가 원자값 만을 저장 | PK(model\_uuid, group\_uuid) 복합키, 부분 함수 종속 없음 | API 관련 속성 ML\_API 테이블로 분리로 이행적 종속 제거 | 3NF |
| MODEL\_MBTI | 각 필드가 원자값 만을 저장 | PK(mbti\_id) 단일 기본키,부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| BOARD | 각 필드가 원자값 만을 저장 | PK(board\_uuid, group\_uuid, model\_uuid) 복합키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| ML\_API | 각 필드가 원자값 만을 저장 | PK(api\_uuid, model\_uuid, group\_uuid) 복합키, 부분 함수 종속 없음 | API 집계 속성 분리로 이행적 종속 제거 | 3NF |
| API\_CALL\_AGGREGATION | 각 필드가 원자값 만을 저장 | PK(api\_call\_uuid, api\_uuid, model\_uuid, group\_uuid) 복합키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| GROUP | 각 필드가 원자값 만을 저장 | PK(group\_uuid) 단일 기본키,부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| HF\_TOKEN\_MANAGE | 각 필드가 원자값 만을 저장 | PK(hf\_manage\_uuid) 단일 기본키,부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| USER-GROUP | 각 필드가 원자값 만을 저장 | PK(user\_uuid, group\_uuid) 복합키, 부분 함수 종속 없음 | USER, GROUP 테이블 M:N 관계를 교차 테이블로 분리 | 3NF |
| USER | 각 필드가 원자값 만을 저장 | PK(user\_uuid) 단일 기본키,부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| SYSTEM\_LOG | 각 필드가 원자값 만을 저장 | PK(log\_uuid) 단일 기본키,  부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |

**4.3 어플리케이션/백엔드 레벨 검증 여부**

**4.4 예외 데이터 처리 전략**

유효성 검사 실패, DB 제약 조건 위반, 트랜잭션 실패 시 예외데이터 처리

1. API 요청시 애플리케이션 검증 - 사용자 입력 시 유효성 검사후 잘못된 요청 차단

2. DB 트랜잭션 롤백 - 제약 조건 위반시 트랜잭션 전으로 롤백

3. 시스템 로그 저장 - 제약 조건 위반시 시스템 로그 저장

4. 데이터, DB 스키마 정제 - 주기적으로 로그 확인 후 검토 수정/삭제

**5. 변경 이력 및 적용 전략**

| **버전** | **변경일** | **변경자** | **설명** | **영향 테이블.속성** | **비고** |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |

**적용 절차**

1. 기능 명세 수정 - 데이터 베이스 변경 이력으로 인해 영향받은 기능의 기능명세를 변경

2. 영향 받는 API 명시 - 영향 받은 API 명시 및 확인

3. 버전, 변경일, 변경자, 설명, 영향-테이블 속성, 비고 작성

4. ERD 수정 - 버전, 변경일, 변경자, 설명, 영향-테이블 속성, 비고를 기반으로 ERD 수정

5. SQL 쿼리 작성 후 DB에 반영

**6. 추가 (선택)**

* 개인정보 보호 항목 및 암호화 필드 정의

| 테이블 명 | 필드 | 암호화 방식 | 비고 |
| --- | --- | --- | --- |
| HF\_TOKEN\_MANAGE | hf\_token\_value | AES256 (대칭 암호화) | 비밀키를 사용하여 암호화 |