

**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** | <https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN11-FINAL-4Team/tree/feature/SKN-8_database_documents> |
| **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. 공통 엔터티**

* **설명:** 외래키로 참조되는 기초 데이터로, 전체 시스템에서 기반 정보로 사용되는 테이블

**2.1.1.1. 엔터티명: USER**

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

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| user\_id | VARCHAR(255) | ✓ | 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 |

**2.1.1.2. 엔터티명: GROUP**

* **설명:** 기업 내 팀 또는 조직 단위로 구성된 그룹 정보를 저장하는 테이블
* **식별자: PK:** group\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| group\_id | INTEGER | ✓ | auto\_increment | 그룹 고유 식별자,  기업 내부 팀 단위(Company ≠ Group) |
| group\_name | VARCHAR(100) | ✓ |  | 그룹명 |
| group\_description | TEXT |  |  | 그룹 설명 |
| created\_at | TIMESTAMP | ✓ | current\_  timestamp( ) | 그룹 정보 생성 시각 |
| updated\_at | TIMESTAMP | ✓ | current\_  timestamp( ) | 그룹 정보 마지막 수정 시각 |

**2.1.1.3. 엔터티명: USER-GROUP**

* **설명:** 사용자와 그룹 간의 소속 관계를 나타내는 매핑 테이블
* **식별자: PK:** (user\_id, group\_id), **FK:** user\_id, group\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| user\_id | VARCHAR(255) | ✓ | UUID | 내부 사용자 고유 식별자 |
| group\_id | INTEGER | ✓ |  | 그룹 고유 식별자 |

**2.1.1.4. 엔터티명: INFLUENCER\_MBTI**

* **설명:** AI 인플루언서의 성격 유형(MBTI)에 따른 특성을 정의한 기준 테이블
* **식별자: PK:** mbti\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| mbti\_id | INTEGER | ✓ | auto\_increment | MBTI 성격 고유 식별자 |
| mbti\_name | VARCHAR(100) | ✓ |  | MBTI 명칭 |
| mbti\_characteristic | VARCHAR(255) | ✓ |  | MBTI 별 성격 및 특성 |
| mbti\_speech | TEXT | ✓ |  | MBTI 말투 |

**2.1.2. 주요 객체 엔터티**

* **설명:** AI 인플루언서 생성과 관련된 핵심 도메인 테이블로, 시스템의 중심 기능을 구성하는 테이블

**2.1.2.1. 엔터티명: STYLE\_PRESET**

* **설명:** 스타일 기반 인플루언서 생성을 위한 프리셋 정보를 저장하는 테이블
* **식별자:** **PK:** style\_preset\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| style\_preset\_id | VARCHAR(255) | ✓ | UUID | 스타일 프리셋 고유 식별자 |
| style\_preset\_name | VARCHAR(100) | ✓ |  | 스타일 프리셋 이름 |
| influencer\_type | TINYINT | ✓ |  | 인플루언서 유형,  0:’인간’, 1:’동물’, 2:’사물’ |
| influencer\_gender | TINYINT | ✓ |  | 인플루언서 성별,  0:’남성’, 1:’여성’, 2:’없음’ |
| influencer\_age\_group | VARCHAR(20) | ✓ |  | 인플루언서 연령대(20대부터) |
| influencer\_hairstyle | VARCHAR(100) | ✓ |  | 인플루언서 헤어스타일 |
| influencer\_style | VARCHAR(255) | ✓ |  | 인플루언서 전체스타일(힙함,청순 등) |
| influencer\_personality | TEXT | ✓ |  | 인플루언서 성격 |
| influencer\_speech | TEXT | ✓ |  | 인플루언서 말투 |
| created\_at | TIMESTAMP | ✓ | current\_  timestamp() | 인플루언서 특징 생성 시각 |
| updated\_at | TIMESTAMP | ✓ | current\_  timestamp() | 인플루언서 특징 수정 시각 |

**2.1.2.2. 엔터티명: AI\_INFLUENCER**

* **설명:** 실제 생성된 AI 인플루언서 모델에 대한 상세 정보를 저장하는 테이블
* **식별자:** **PK:** influencer\_id, **FK**: user\_id, group\_id, style\_preset\_id, mbti\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| influencer\_id | VARCHAR(255) | ✓ | UUID | 인플루언서 고유 식별자 |
| user\_id | VARCHAR(255) | ✓ | UUID | 내부 사용자 고유 식별자 |
| group\_id | INTEGER | ✓ |  | 그룹 고유 식별자 |
| style\_preset\_id | VARCHAR(255) | ✓ | UUID | 스타일 프리셋 고유 식별자 |
| mbti\_id | INTEGER |  |  | MBTI 성격 고유 식별자 |
| influencer\_name | VARCHAR(100) | ✓ |  | AI 인플루언서 이름 |
| image\_url | TEXT |  |  | 기업이 제공해준 인플루언서 이미지 파일 URL 경로 |
| influencer\_data\_url | VARCHAR(255) |  |  | 인플루언서 학습에 사용된 데이터셋 URL 경로 (QA쌍) |
| learning\_status | TINYINT | ✓ |  | 0:’학습 중’, 1:’사용가능’,  인플루언서 학습 상태 |
| influencer\_model\_repo | VARCHAR(255) | ✓ |  | 학습 완료된 인플루언서 모델이 저장된 허깅페이스 repo URL 경로 |
| chatbot\_option | BOOLEAN | ✓ |  | 챗봇 생성 옵션 여부  (True: 챗봇 생성 버튼 노출) |
| created\_at | TIMESTAMP | ✓ | current\_ timestamp() | 인플루언서 생성 시각 |
| updated\_at | TIMESTAMP | ✓ | current\_ timestamp() | 인플루언서 정보 수정 시각 |

**2.1.3. 부가 데이터 엔터티**

* **설명:** AI\_INFLUENCER에 종속되거나 연관 기능을 수행하는 부속 데이터 테이블

**2.1.3.1. 엔터티명: BATCH\_KEY**

* **설명:** AI 인플루언서별로 자동 배치 실행을 위한 식별 키를 관리하는 테이블
* **식별자:** **PK:** batch\_ket\_id, **FK:** influencer\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| batch\_key\_id | VARCHAR(255) | ✓ | UUID | 배치키 고유 식별자 |
| influencer\_id | VARCHAR(255) | ✓ | UUID | 인플루언서 고유 식별자 |
| batch\_key | VARCHAR(255) | ✓ |  | 배치키 값 |

**2.1.3.2. 엔터티명: INFLUENCER\_API**

* **설명:** 생성된 AI 인플루언서의 모델에 대한 API Key 발급 정보를 저장하는 테이블
* **식별자:** **PK:** api\_id, **FK:** influencer\_id, user\_id, group\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| api\_id | VARCHAR(255) | ✓ | UUID | API 고유 식별자 |
| influencer\_id | VARCHAR(255) | ✓ | UUID | 인플루언서 고유 식별자 |
| api\_key | VARCHAR(255) | ✓ |  | 발급된 API key |
| created\_date | TIMESTAMP | ✓ | current\_ timestamp() | API 발급 일시 |
| updated\_date | TIMESTAMP | ✓ | current\_ timestamp() | API 발급정보 마지막 수정 일시 |

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

* **설명:** API 호출 이력의 일별 집계 데이터를 저장하는 테이블
* **식별자:** **PK:** api\_call\_id, **FK:** api\_id, influencer\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| api\_call\_id | VARCHAR(255) | ✓ | UUID | 집계 데이터 고유 식별자 |
| api\_id | VARCHAR(255) | ✓ | UUID | API 고유 식별자 |
| influencer\_id | VARCHAR(255) | ✓ | UUID | 인플루언서 고유 식별자 |
| daily\_call\_count | INTEGER | ✓ | 0 | 하루동안 호출된 총 횟수 |
| created\_date | TIMESTAMP | ✓ | current\_ timestamp() | 호출된 집계 일시 |
| updated\_date | TIMESTAMP | ✓ | current\_ timestamp() | 호출된 집계 변경 일시 |

**2.1.3.4. 엔터티명: BOARD**

* **설명:** 인플루언서 게시물 콘텐츠 등록 및 플랫폼별 예약 게시를 위한 테이블
* **식별자:** **PK:** board\_id, **FK:** influencer\_id, user\_id, group\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| board\_id | VARCHAR(255) | ✓ | UUID | 게시물 고유 식별자 |
| influencer\_id | VARCHAR(255) | ✓ | UUID | 인플루언서 고유 식별자 |
| user\_id | VARCHAR(255) | ✓ | UUID | 내부 사용자 고유 식별자 |
| group\_id | INTEGER | ✓ |  | 그룹 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\_hashtag | TEXT |  |  | 해시태그 리스트, JSON 형식으로 저장 |
| board\_status | TINYINT | ✓ | 0 | 0:’최초생성’,1:’임시저장’,1:’예약’,2:’발행됨’ |
| image\_url | TEXT | ✓ |  | 게시물 이미지,영상 URL 경로,  JSON 형식으로 저장 |
| created\_at | TIMESTAMP | ✓ | current\_ timestamp() | 게시물 생성 시각 |
| updated\_at | TIMESTAMP | ✓ | current\_ timestamp() | 게시물 수정 시각 |
| reservation\_at | TIMESTAMP |  | current\_ timestamp() | 게시물 예약 발행 시각 |
| published\_at | TIMESTAMP |  | current\_ timestamp() | 게시물 발행 시각 |

**2.1.3.5. 엔터티명: CHAT\_MESSAGE**

* **설명:** 테스트 챗봇 상에서 사용자와 인플루언서 간의 대화 내역을 저장하는 테이블
* **식별자:** **PK:** session\_id, **FK:** influencer\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| session\_id | INTEGER | ✓ | auto\_increment | 대화 세션 고유 식별자 |
| influencer\_id | VARCHAR(255) | ✓ | UUID | 인플루언서 고유 식별자 |
| created\_at | TIMESTAMP | ✓ |  | 대화 시작 시각 |
| end\_at | TIMESTAMP |  |  | 대화 종료 시각 |
| message\_content | TEXT | ✓ |  | 총 대화 내용,  JSON 형식으로 저장 |

**2.1.4. 관리용 엔터티**

* **설명:** 시스템 운영, 외부 연동 및 모니터링을 위한 관리 및 로그 테이블

**2.1.4.1. 엔터티명: HF\_TOKEN\_MANAGE**

* **설명:** 허깅페이스 모델 저장소 연결을 위한 그룹별 토큰 정보를 저장하는 테이블
* **식별자:** **PK:** hf\_manage\_id, **FK:** group\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| hf\_manage\_id | VARCHAR(255) | ✓ | UUID | 허깅페이스 토큰 관리 고유 식별자 |
| group\_id | INTEGER | ✓ |  | 그룹 고유 식별자 |
| hf\_token\_value | TEXT | ✓ |  | 허깅페이스 실제 토큰 값 (암호화) |
| 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 |

**2.1.4.2. 엔터티명: SYSTEM\_LOG**

* **설명:** 시스템 전반의 사용자 활동, 인증, 에러 등 로그 정보를 기록하는 테이블
* **식별자:** **PK:** log\_id, **FK:** user\_id

| **속성명** | **데이터 타입** | **Not Null** | **Default value** | **설명** |
| --- | --- | --- | --- | --- |
| log\_id | VARCHAR(255) | ✓ | UUID | 시스템 로그 고유 식별자 |
| user\_id | VARCHAR(255) | ✓ | UUID | 내부 사용자 고유 식별자 |
| log\_type | TINYINT | ✓ |  | 0:’API요청’,1:’시스템오류’,2:’인증관련’ |
| log\_content | TEXT | ✓ |  | API 요청 내용,  오류 메시지 등 상세한 로그 내용,  JSON 형식으로 저장 |
| created\_at | TIMESTAMP | ✓ | current\_ timestamp() | 로그 발생 시각 |

**2.2. 엔터티 간 관계   
2.2.1. 사용자 및 그룹 관련 관계  
2.2.1.1. 관계명: ‘USER’ - ‘USER-GROUP’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER | USER-GROUP | 1:N | 한 사용자는 여러 개의 그룹에 소속될 수 있음. USER-GROUP 테이블은 사용자와 그룹 간 다대다 관계를 중간에서 매핑함. |

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

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | USER-GROUP | 1:N | 하나의 그룹은 여러 명의 사용자를 가질 수 있음.  USER-GROUP 테이블을 통해 사용자-그룹 간의 소속 관계를 정의함. |

**2.2.2. 사용자 및 그룹 관련 관계  
2.2.2.1. 관계명: ‘USER-GROUP’ - ‘AI\_INFLUENCER’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER-  GROUP | AI\_INFLUENCER | 1:N | 하나의 그룹은 여러 명의 사용자를 가질 수 있으며, 각 사용자는 소속된 그룹의 권한 내에서 인플루언서를 생성할 수 있음.  이때 인플루언서 데이터는 특정 `user\_id`와 `group\_id`를 함께 참조하여, 누가 어느 그룹 소속으로 인플루언서를 생성했는지를 명확히 기록함. |

**2.2.2.2. 관계명: ‘USER’ - ‘AI\_INFLUENCER’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER | AI\_INFLUENCER | 1:N | 한 사용자는 여러 AI 인플루언서를 생성할 수 있음. 인플루언서를 생성한 사람이 누구인지를 명확히 하기 위해 AI\_INFLUENCER 테이블에 `user\_id`를 외래키로 포함시킴. 이는 동일한 그룹 내 사용자 간 생성 이력 구분을 가능하게 해줌. |

**2.2.3. 스타일, MBTI 등 인플루언서의 구성 요소 관계  
2.2.3.1. 관계명: ‘STYLE\_PRESET’ - ‘AI\_INFLUENCER’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| STYLE\_  PRESET | AI\_INFLUENCER | 1:N | 하나의 스타일 프리셋은 여러 인플루언서에 적용될 수 있음. 인플루언서 생성 시 선택된 프리셋 정보는 `style\_preset\_id`를 통해 연결되며, 해당 프리셋을 기반으로 AI 인플루언서의 말투, 성격, 스타일 등을 학습함. 이를 통해 이전에 정의된 스타일을 재활용하여 빠르게 신규 인플루언서를 생성할 수 있도록 지원함. |

**2.2.3.2. 관계명: ‘MBTI’ - ‘AI\_INFLUENCER’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| MBTI | AI\_INFLUENCER | 1:N | 인플루언서는 선택적으로 MBTI 성격 유형을 참조할 수 있으며, 유형별 말투/성격 생성에 활용됨. `mbti\_id`를 참조함. |

**2.2.4. 콘텐츠 및 커뮤니케이션 관련 관계  
2.2.4.1. 관계명: ‘AI\_INFLUENCER’ - ‘CHAT\_MESSAGE’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| AI\_  INFLUENCER | CHAT\_  MESSAGE | 1:N | 하나의 인플루언서와 여러 대화 세션이 연결될 수 있으며, 해당 인플루언서와 사용자 간의 대화 내용은 CHAT\_MESSAGE 테이블에 저장됨. |

**2.2.4.2. 관계명: ‘AI\_INFLUENCER’ - ‘BOARD’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| AI\_  INFLUENCER | BOARD | 1:N | 인플루언서가 생성한 콘텐츠 게시물이 게시판에 저장됨. 각 인플루언서의 게시물은 BOARD 테이블을 통해 관리됨. |

**2.2.5. API 및 호출 집계 관련  
2.2.5.1. 관계명: ‘AI\_INFLUENCER’ - ‘INFLUENCER\_API’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| AI\_  INFLUENCER | INFLUENCER\_  API | 1:N | 하나의 인플루언서는 하나의 API 키를 발급받음. 인플루언서별로 고유 API 호출이 가능하도록 INFLUENCER\_API에 정보를 저장함. |

**2.2.5.2. 관계명: ‘INFLUENCER\_API’ - ‘API\_CALL\_AGGREGATION’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| INFLUENCER\_  API | API\_CALL\_  AGGREGATION | 1:N | 발급된 하나의 API에 대해 일자별 호출 통계를 기록함. 하루 단위 호출 횟수 집계를 저장하기 위한 테이블임. |

**2.2.5.3. 관계명: ‘GROUP’ - ‘INFLUENCER\_API’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | INFLUENCER\_  API | 1:N | 하나의 그룹은 여러 인플루언서 API를 생성할 수 있음. 각 API는 특정 그룹에 속한 인플루언서에 대해 발급되며, 이를 구분하기 위해 INFLUENCER\_API 테이블에 `group\_id`를 외래키로 저장함. 이를 통해 어떤 그룹이 어떤 API키를 보유하고 있는지를 추적 가능하며, 기업별 API 사용 현황을 분리 관리할 수 있음. |

**2.2.5.4. 관계명: ‘GROUP’ - ‘API\_CALL\_AGGREGATION’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | API\_CALL\_  AGGREGATION | 1:N | 하나의 그룹은 여러 인플루언서의 API 호출 데이터를 가질 수 있음. API 호출 이력은 그룹 단위로 집계되어 기록되며, `group\_id`를 통해 소속 그룹 기준으로 호출 통계를 관리할 수 있음. |

**2.2.5.5. 관계명: ‘USER’ - ‘INFLUENCER\_API’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER | INFLUENCER\_  API | 1:N | 하나의 사용자는 여러 인플루언서 모델에 대한 API 키를 발급받을 수 있음. API 키 발급자는 `user\_id`를 통해 식별되며, 사용자가 어떤 인플루언서에 대한 키를 요청했는지 추적 가능함. |

**2.2.5.6. 관계명: ‘USER’ - ‘API\_CALL\_AGGREGATION’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER | API\_CALL\_  AGGREGATION | 1:N | 하나의 사용자는 여러 API 호출을 수행할 수 있으며, 호출 집계 기록은 `user\_id` 기준으로 저장됨. 이를 통해 각 사용자의 API 이용 빈도 및 사용 이력을 모니터링할 수 있음. |

**2.2.6. 배치/허깅페이스/시스템 관리  
2.2.6.1. 관계명: ‘GROUP’ - ‘HF\_TOKEN\_MANAGE’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | HF\_TOKEN\_  MANAGE | 1:N | 하나의 그룹은 여러 HuggingFace 토큰 정보를 가질 수 있음. 그룹 단위로 모델 배포 권한을 관리하며, `group\_id`를 통해 토큰과 그룹 간 소속 관계를 명확히 함. |

**2.2.6.2. 관계명: ‘AI\_INFLUENCER’ - ‘BATCH\_KEY’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| AI\_  INFLUENCER | BATCH\_KEY | 1:N | 하나의 인플루언서는 여러 번의 배치 처리 이력을 가질 수 있음. 각 배치 작업은 어떤 인플루언서에 대해 수행되었는지를 `influencer\_id`로 연결하여 관리함. |

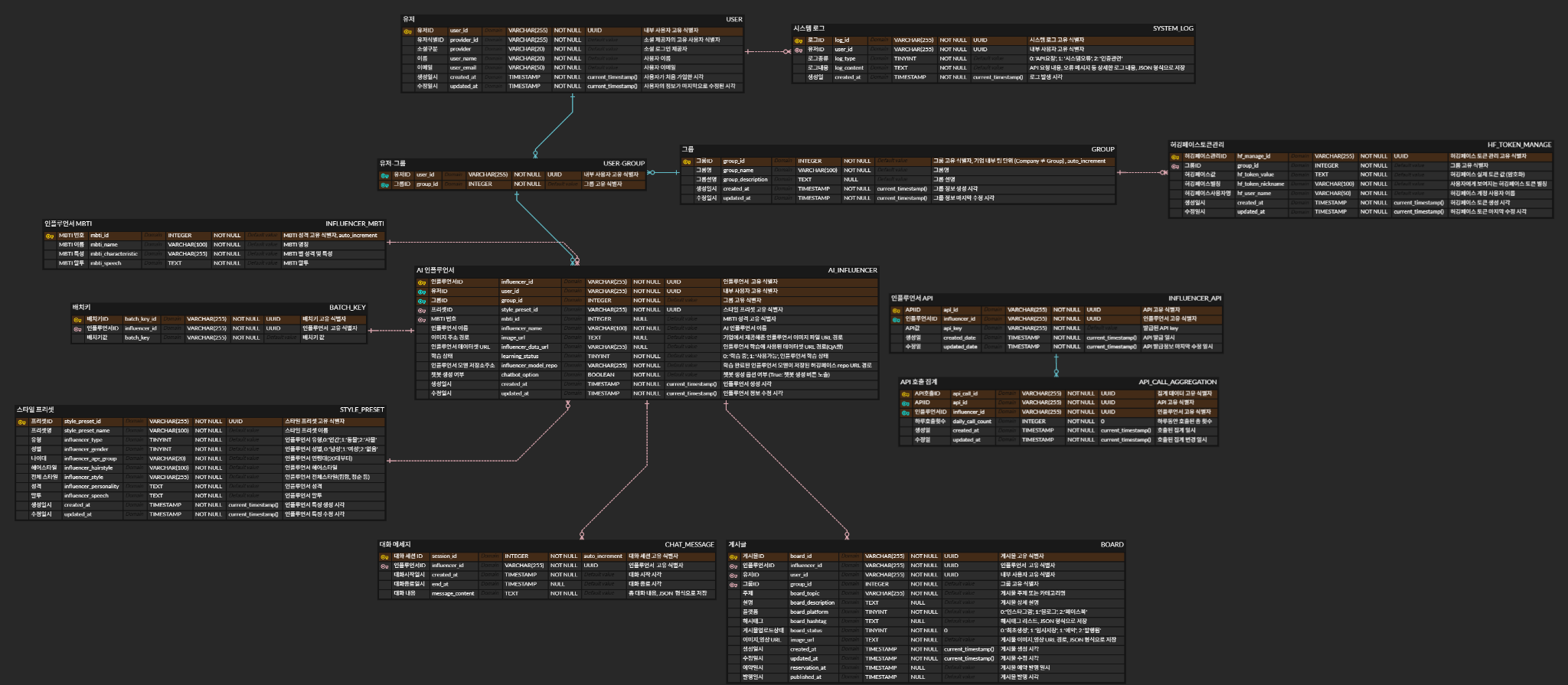
**2.2.6.3. 관계명: ‘GROUP’ - ‘BATCH\_KEY’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| GROUP | BATCH\_KEY | 1:N | 하나의 그룹은 소속 인플루언서에 대해 여러 배치 작업을 수행할 수 있음. 배치키는 `group\_id`를 함께 저장하여 어떤 그룹이 실행한 작업인지를 기록함. |

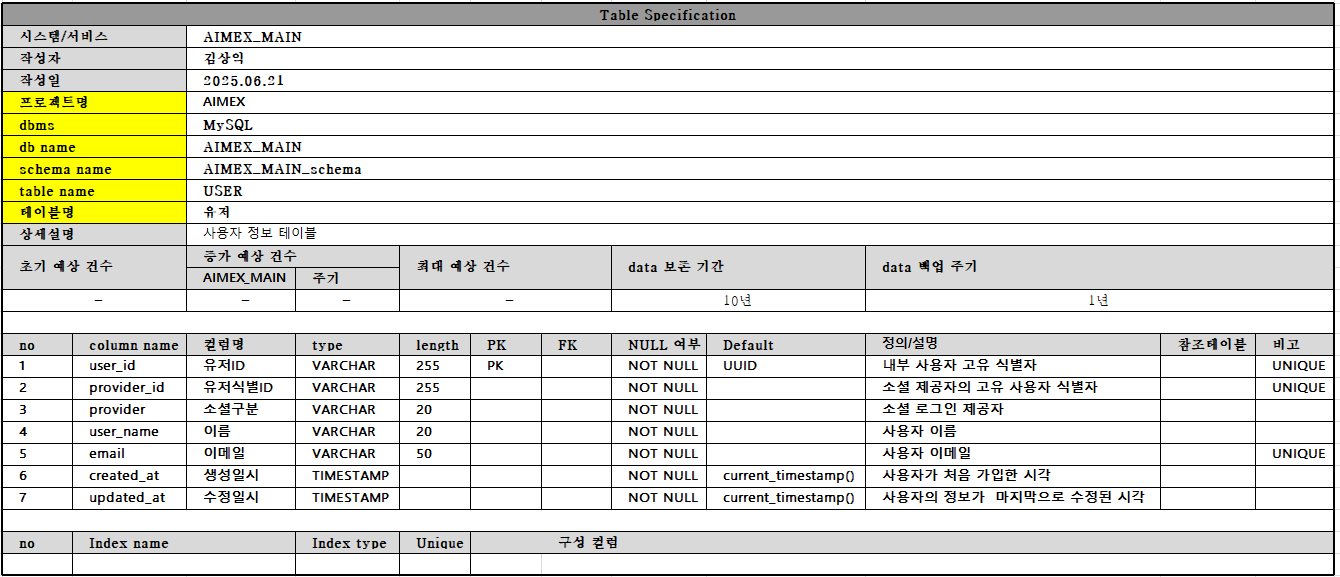
**2.2.6.4. 관계명: ‘USER’ - ‘BATCH\_KEY’**

| **주 엔터티** | **종 엔터티** | **관계 종류** | **설명** |
| --- | --- | --- | --- |
| USER | BATCH\_KEY | 1:N | 하나의 사용자는 여러 배치 작업을 실행할 수 있으며, 각 배치키에는 실행한 사용자 `user\_id`가 기록됨. 이를 통해 배치 작업의 수행 주체를 식별할 수 있음. |

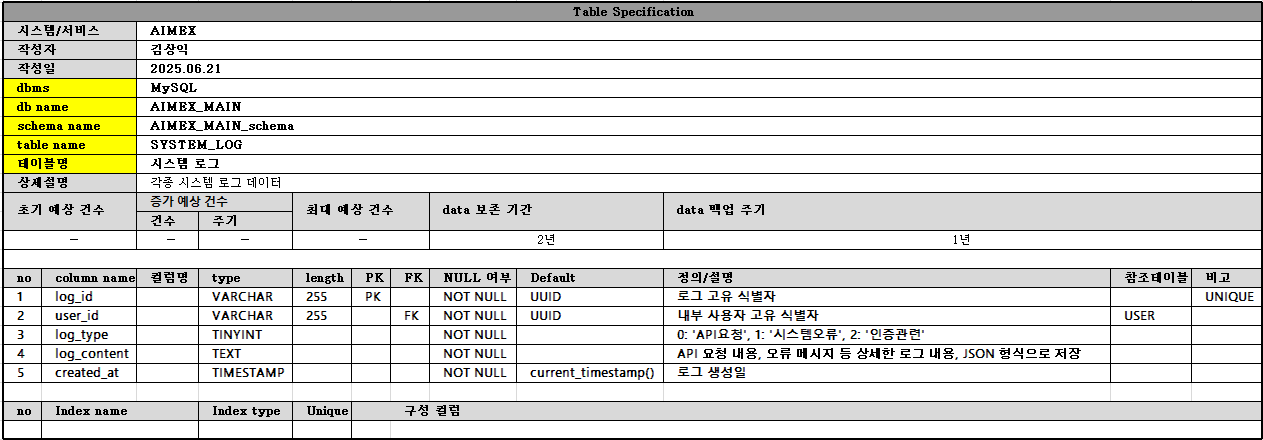
**2.3 ERD**

****

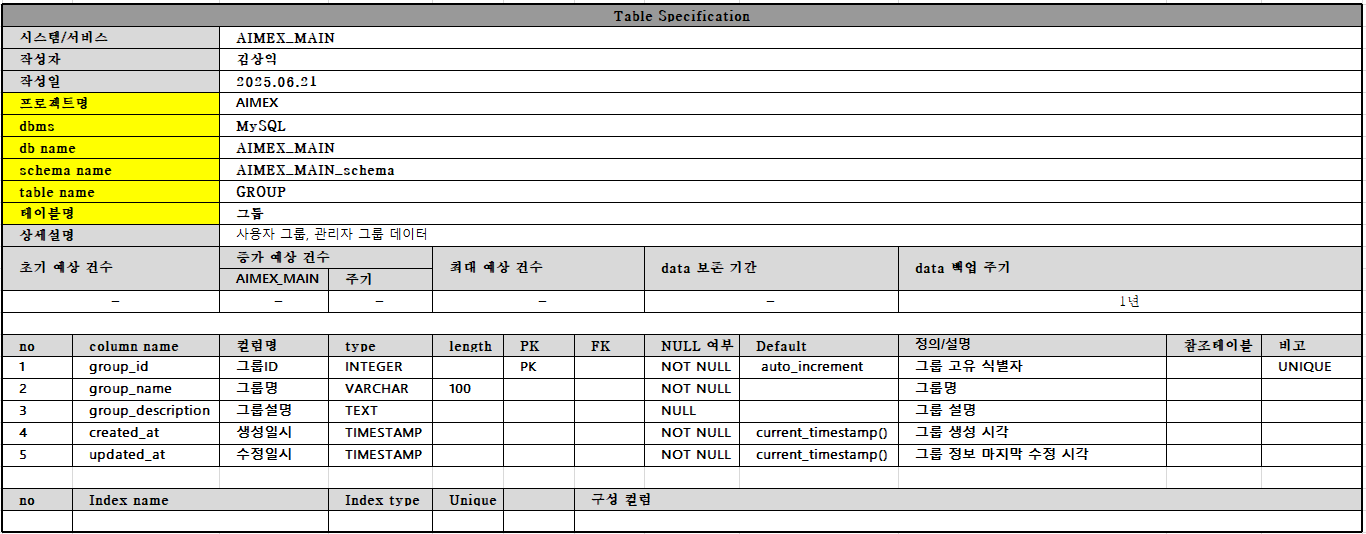
**3. 물리 데이터 모델  
3.1 테이블 정의서 (자세한 내용은 첨부파일 참고)  
3.1.1 유저 테이블 명세서**

****

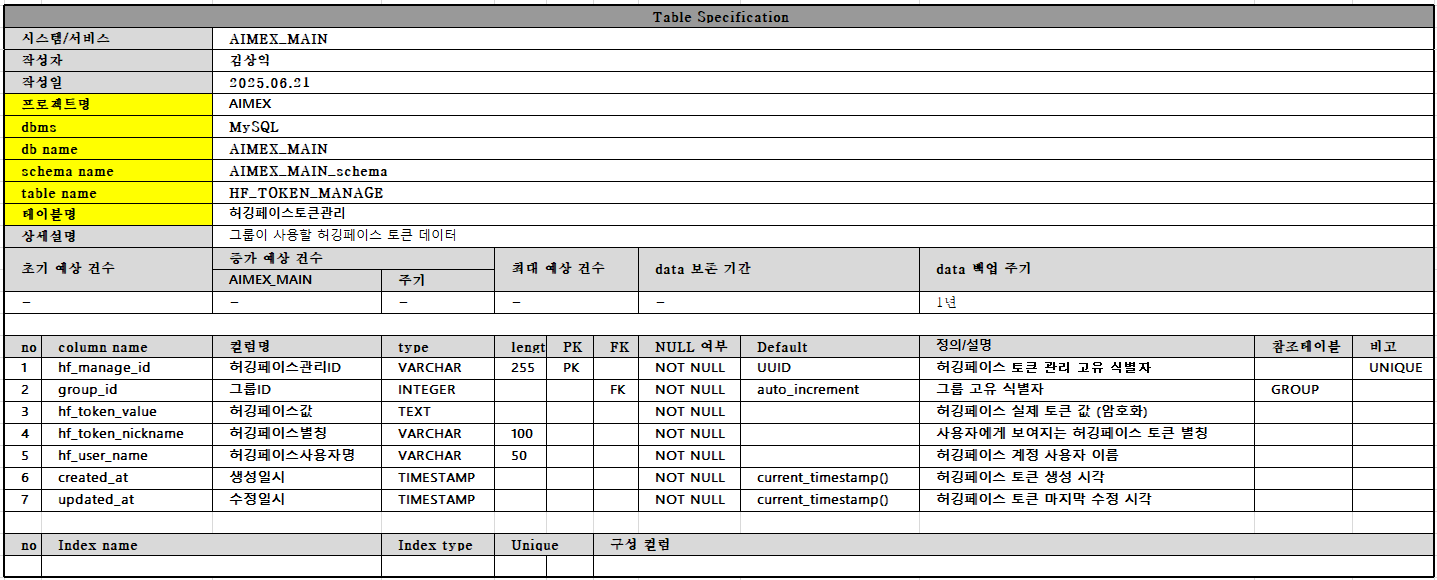
**3.1.2 시스템 로그 테이블 명세서**



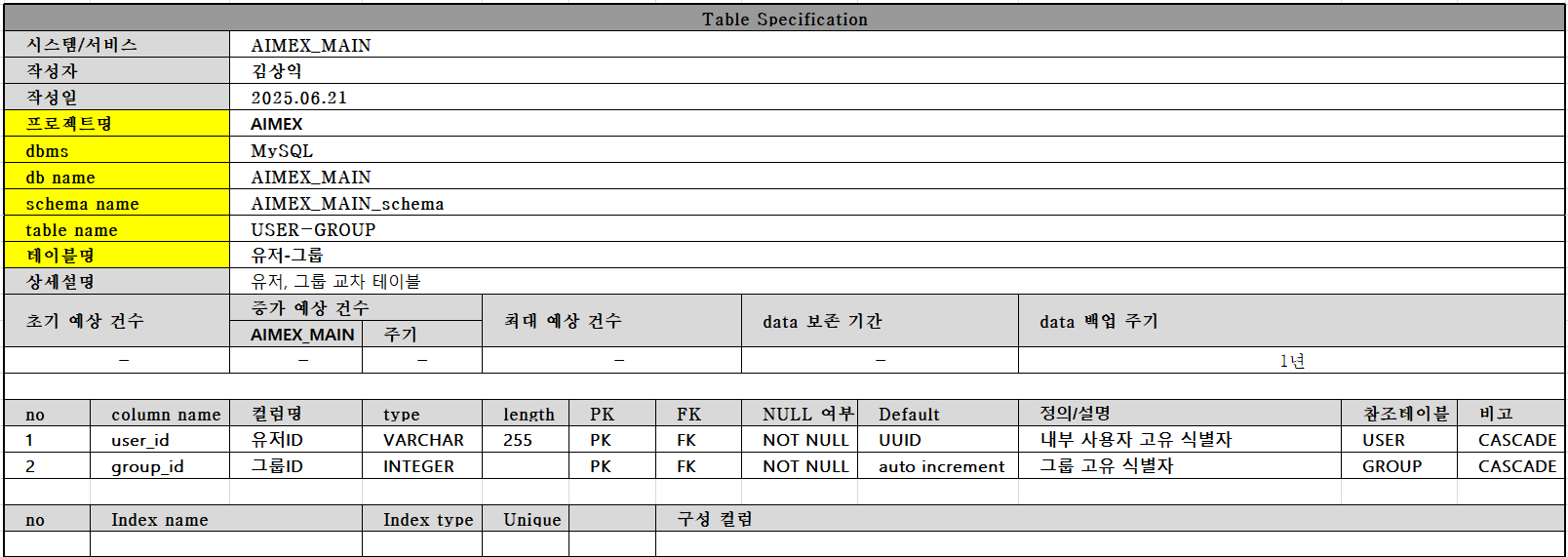
**3.1.3 그룹 테이블 명세서**



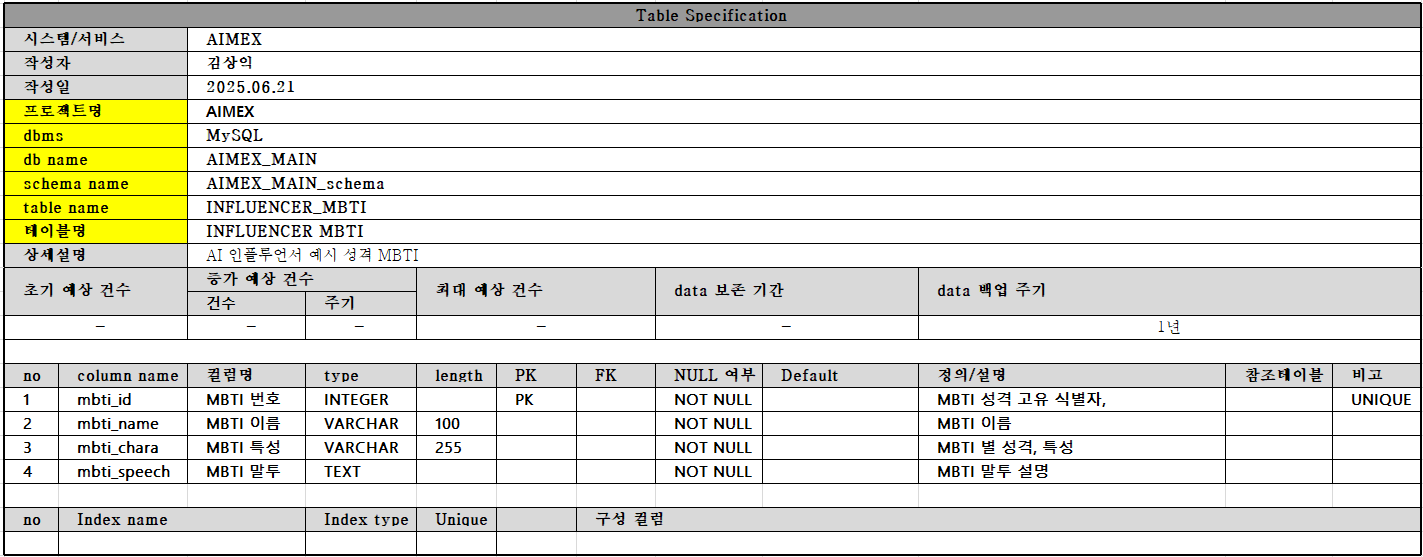
**3.1.4 허깅페이스 토큰관리 테이블 명세서**



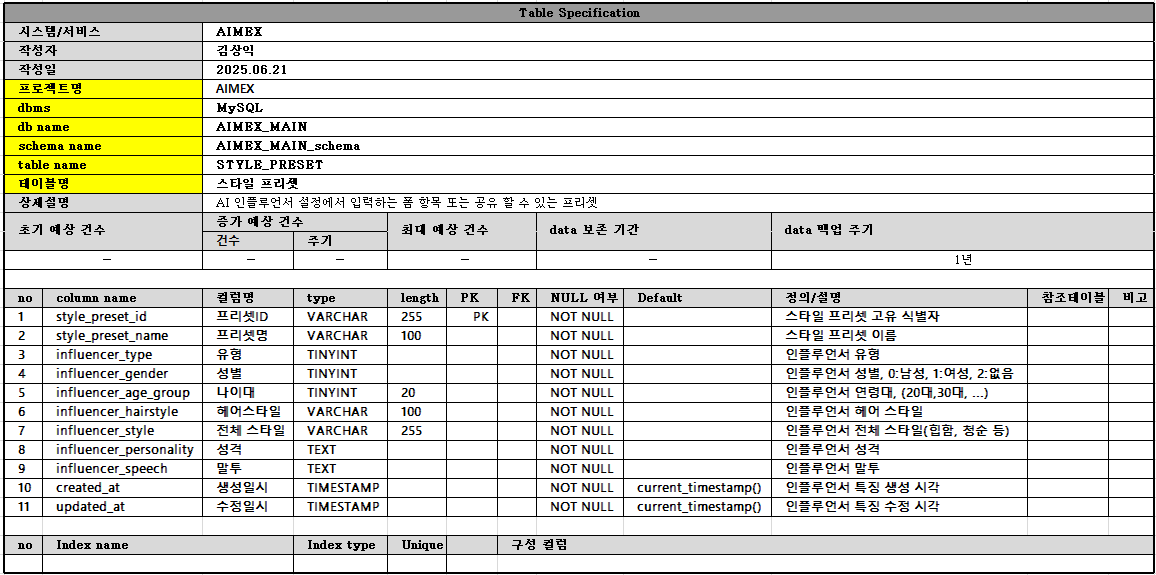
**3.1.5 유저 - 그룹 교차 테이블 명세서**



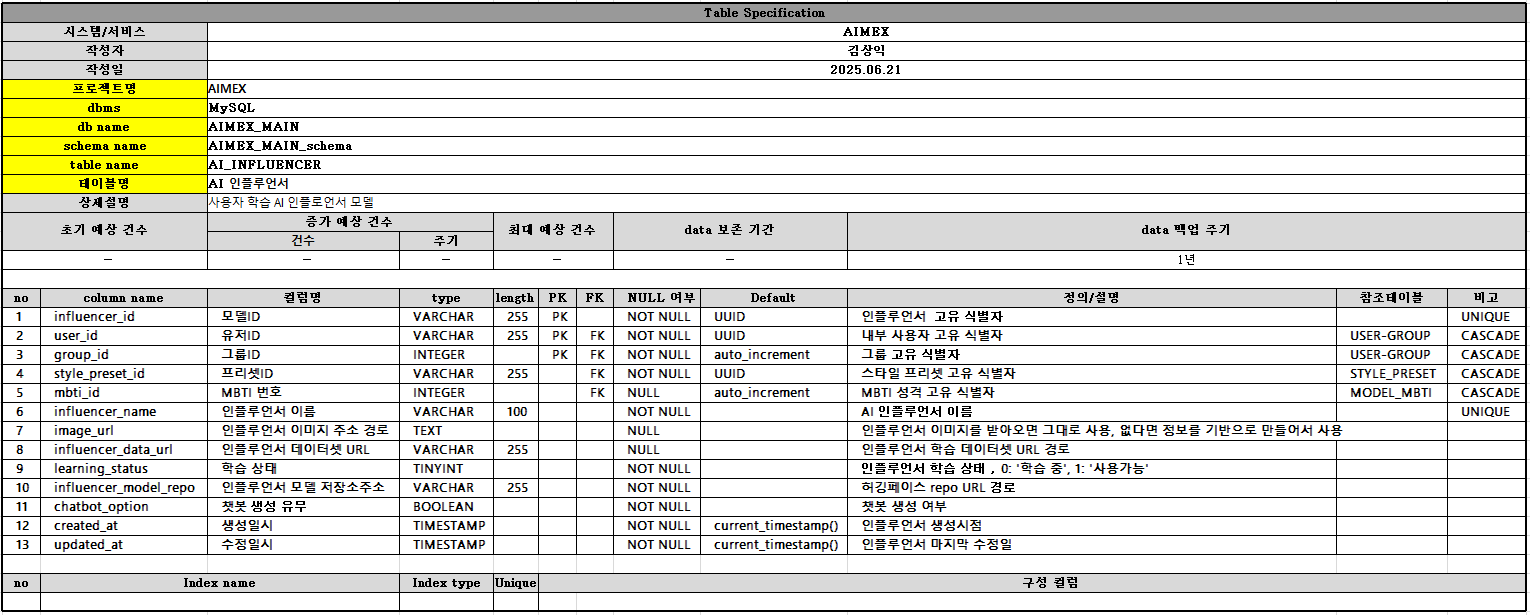
**3.1.6 인플루언서 MBTI 테이블 명세서**



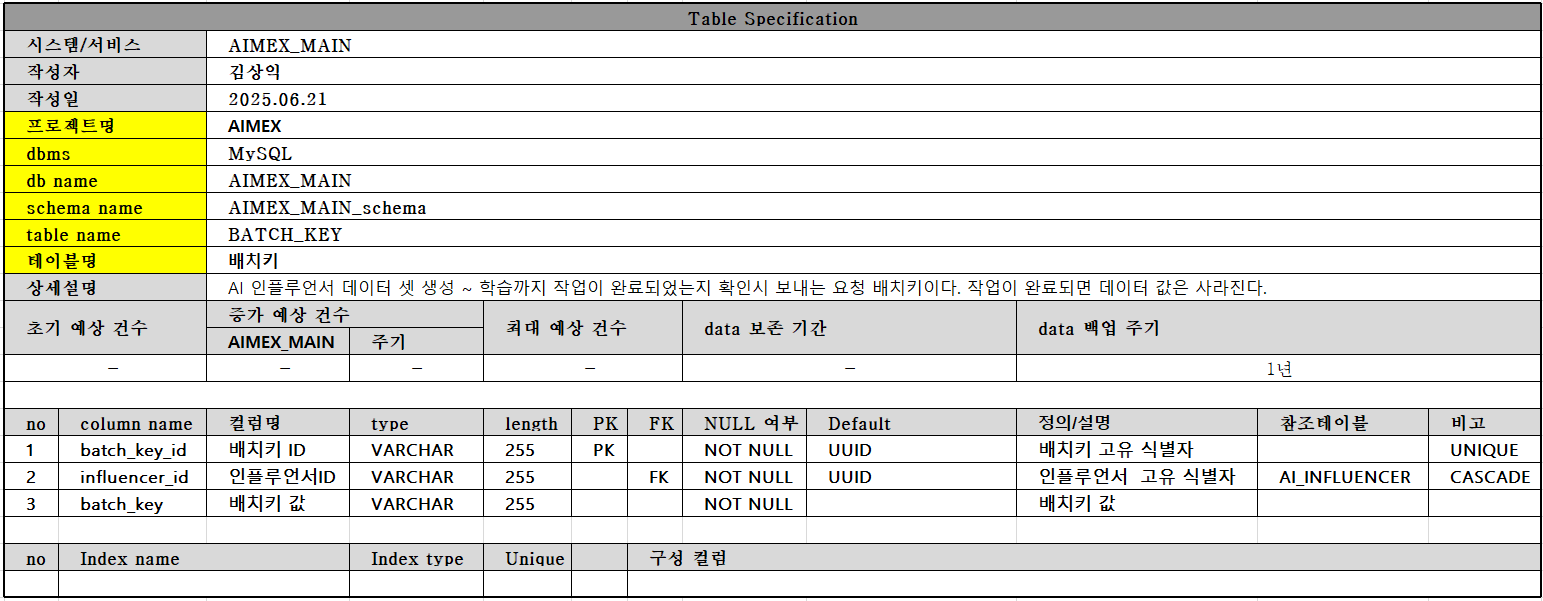
**3.1.7 스타일 프리셋 테이블 명세서**

****

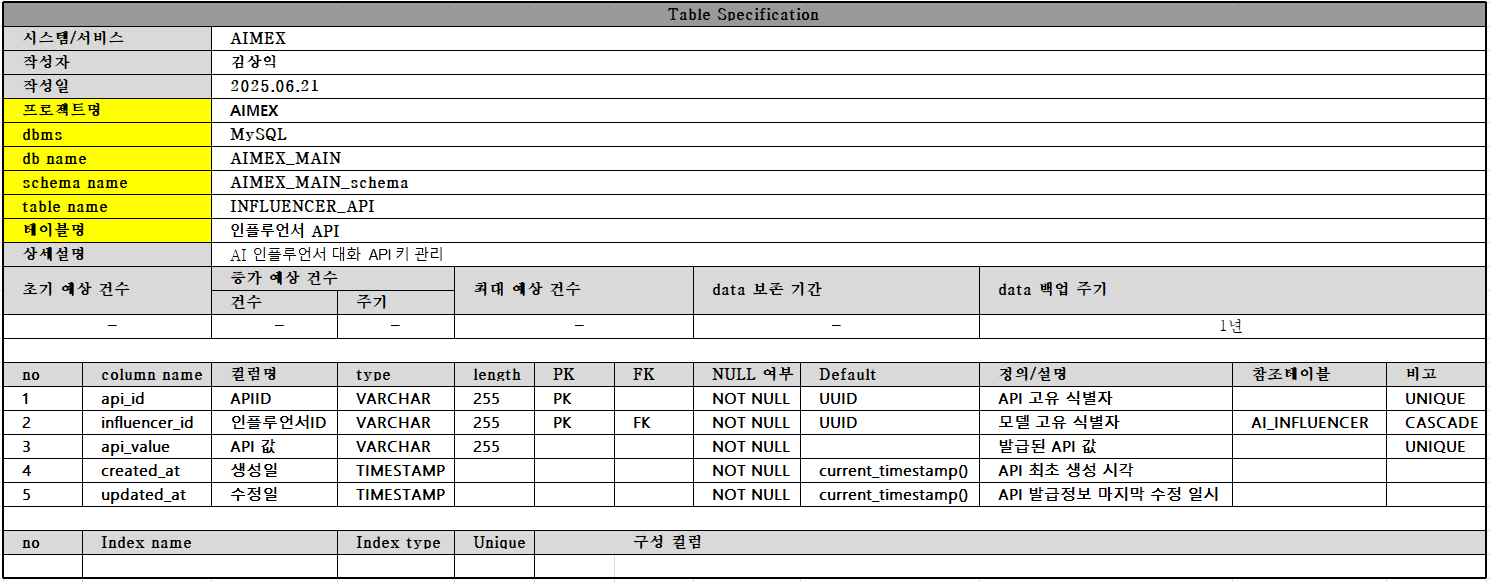
**3.1.8 AI 인플루언서 테이블 명세서**

****

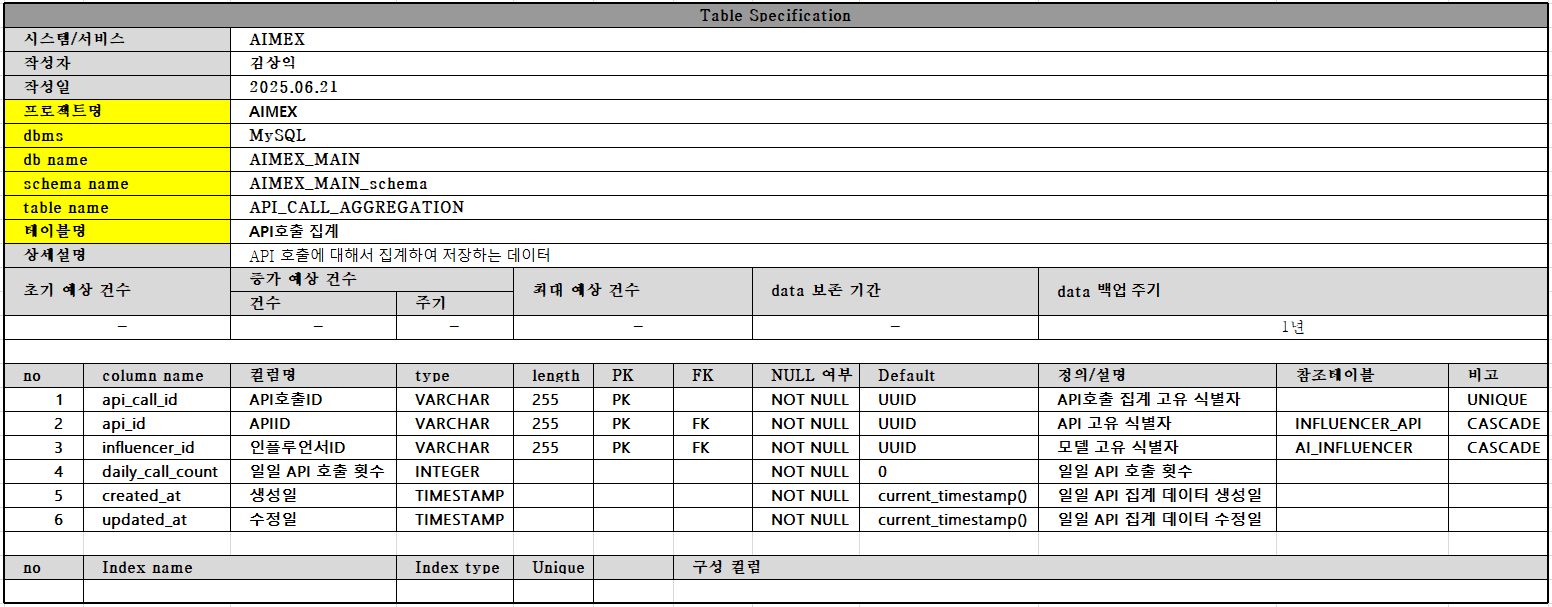
**3.1.9 배치키 테이블 명세서**

****

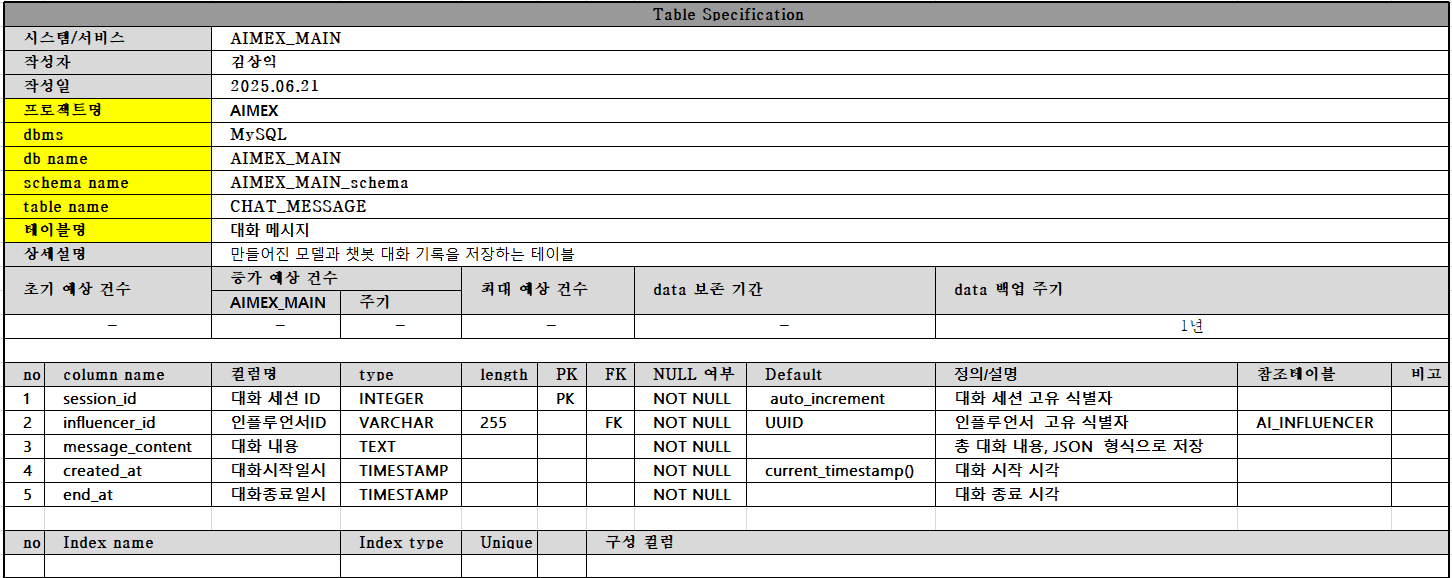
**3.1.10 인플루언서 API 명세서**

****

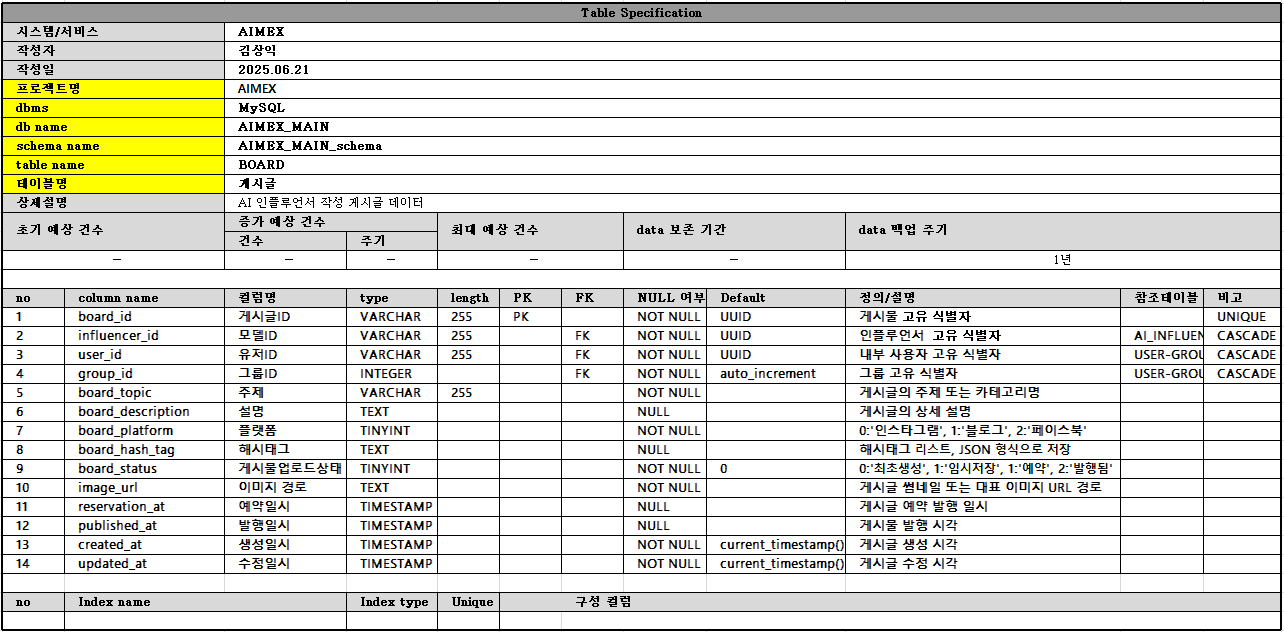
**3.1.11 API 호출 집계 테이블 명세서**

****

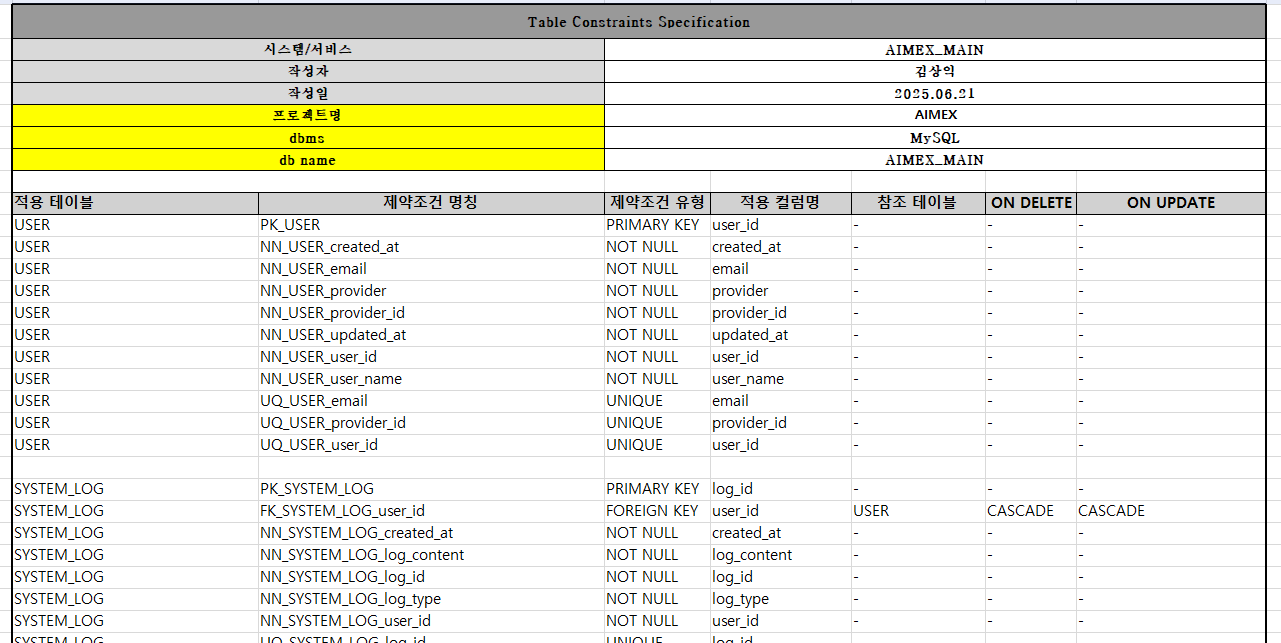
**3.1.12 대화 메세지 테이블 명세서**

****

**3.1.13 게시글 테이블 명세서**

****

**3.2 제약 조건 명세 (자세한 내용은 첨부파일 참고)  
3.2.1 데이터베이스 제약조건 명세서**

****

**4. 데이터 정합성 및 무결성 관리 방안  
4.1 중복 방지 전략**

| **주 엔터티** | **속성명** | **데이터 타입** | **중복방지 여부** | **설명** |
| --- | --- | --- | --- | --- |
| USER | user\_id | VARCHAR(255) | UNIQUE | 내부 사용자 고유 식별자 중복 불가 |
| USER | provider\_id | VARCHAR(255) | UNIQUE | 소셜 제공자의 고유 사용자 식별자 중복 불가 |
| USER | email | VARCHAR(50) | UNIQUE | 사용자 이메일 중복 불가 |
| SYSTEM\_ LOG | log\_id | VARCHAR(255) | UNIQUE | 로그 고유 식별자 중복 불가 |
| GROUP | group\_id | INTEGER | UNIQUE | 그룹 고유 식별자 중복 불가 |
| HF\_TOKEN\_ MANAGE | hf\_manage\_  id | VARCHAR(255) | UNIQUE | 허깅페이스 토큰 관리 고유 식별자 중복 불가 |
| INFLUENCER\_ MBTI | mbti\_id | INTEGER | UNIQUE | MBTI 성격 고유 식별자 중복 불가 |
| AI\_INFLUENCER | influencer\_id | VARCHAR(255) | UNIQUE | 인플루언서 고유 식별자 중복 불가 |
| AI\_INFLUENCER | Influencer\_  name | VARCHAR(100) | UNIQUE | AI 인플루언서 이름 중복 불가 |
| BATCH\_KEY | batch\_key\_id | VARCHAR(255) | UNIQUE | 배치키 고유 식별자 중복 불가 |
| INFLUENCER\_API | api\_id | VARCHAR(255) | UNIQUE | API 공유 식별자 중복 불가 |
| INFLUENCER\_ API | api\_value | VARCHAR(255) | UNIQUE | 발급된 API 값 중복 불가 |
| API\_CALL\_ AGGREGATION | api\_call\_id | VARCHAR(255) | UNIQUE | API호출 집계 고유 식별자 중복 불가 |
| BOARD | board\_id | VARCHAR(255) | UNIQUE | 게시물 고유 식별자 중복 불가 |

**4.2 정규화 수준**

| **주 엔터티** | **1NF** | **2NF** | **3NF** | **정규형** |
| --- | --- | --- | --- | --- |
| USER | 각 필드가 원자값만을 저장 | PK(user\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| SYSTEM\_LOG | 각 필드가 원자값만을 저장 | PK(log\_id) 단일 기본키,  부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| GROUP | 각 필드가 원자값만을 저장 | PK(group\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| HF\_TOKEN\_ MANAGE | 각 필드가 원자값만을 저장 | PK(hf\_manage\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| USER-GROUP | 각 필드가 원자값만을 저장 | PK(user\_id, group\_id) 복합키, 부분 함수 종속 없음 | USER, GROUP 테이블 M:N 관계를 교차 테이블로 분리 | 3NF |
| INFLUENCER\_ MBTI | 각 필드가 원자값만을 저장 | PK(mbti\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| STYLE\_PRESET | 각 필드가 원자값만을 저장 | PK(style\_preset\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| AI\_INFLUENCER | 각 필드가 원자값만을 저장 | PK(influencer\_id) 단일 기본키, 부분 함수 종속 없음 | STYLE\_PRESET, MODEL\_MBTI 테이블로 분리하여 이행적 종속 제거 | 3NF |
| BATCH\_KEY | 각 필드가 원자값만을 저장 | PK(batch\_key\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| CHAT\_MESSAGE | 각 필드가 원자값만을 저장 | PK(session\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| INFLUENCER\_ API | 각 필드가 원자값만을 저장 | PK(api\_id, influencer\_id) 복합키, 부분 함수 종속 없음 | API 집계 속성을 API\_CALL\_ AGGREGATION 테이블로 분리하여 이행적 종속 제거 | 3NF |
| API\_CALL\_ AGGREGATION | 각 필드가 원자값만을 저장 | PK(api\_call\_id, api\_id, influencer\_id) 복합키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |
| BOARD | 각 필드가 원자값만을 저장 | PK(board\_id) 단일 기본키, 부분 함수 종속 없음 | 이행적 종속 없음 | 3NF |

**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 |