SK네트웍스 **Family AI**과정 **20**기

데이터 조회 및 관리 프로그램 명세서

산출물 단계	데이터 수집 및 저장
평가 산출물	데이터 조회 및 관리 프로그램 명세서
제출 일자	2026-01-23
깃허브 경로	https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN20-FINAL-6TEAM/tree/main
작성 팀원	정소영

1. 프로젝트 개요

1.1 프로젝트 주제

CEO를 위한 비즈니스 AI 에이전트 플랫폼

1.2 목표 시스템

사용자(CEO)의 자연어 질의에 대해 사내 규정, 국가 법령, 정부 지원사업 데이터를 검색(Retrieval)하여 사용자 맞춤형 답변을 제공하는 서비스를 구축합니다. 또한 전처리된 법령, 해석례, 창업가이드, 세무일정 데이터를 효율적으로 조회하고 관리하기 위한 방법론 및 구조를 정의합니다.

---

2. 데이터 처리 프로세스

2.1 전체 데이터 파이프라인

[원본 데이터] → [전처리 파이프라인] → [통일 스키마 출력] → [Vector DB 적재] → [RAG System]

처리흐름:

1. 원본 데이터 수집 (JSON/CSV/PDF)
2. 도메인별 전처리 프로세서 실행
3. 통일된 JSONL 스키마로 변환
4. ChromaDB에 임베딩 벡터 저장
5. Multi-Agent가 벡터 검색 수행

데이터 유형	출처	형식	도메인
법령데이터	국가법령정보센터 Open API	JSON	legal, labor, tax
법령해석례	국가법령정보센터 Open API	JSON	legal, labor, tax, startup
창업가이드	기업마당/K-Startup	JSON	startup
세무일정	국세청	CSV	tax
규정/가이드	정부 공공문서	PDF	tax, labor
총 데이터 규모 : 약 8,516건 (벡터 임베딩 대상)			

---

3. 데이터베이스 구성

3.1 DB 사용 용도

DB 종류	기술 스택	주요 용도
Vector DB	ChromaDB	• RAG(검색 증강 생성) 구현 • 법령 및 공고문의 임베딩 벡터 저장 • 질문과 유사도(Similarity)가 높은 텍스트 청크 검색
RDBMS	MySQL	• 서비스 운영 및 관리 • 사용자 정보, 기업 정보, 채팅 히스토리 저장 • 정형 데이터(메타데이터) 관리

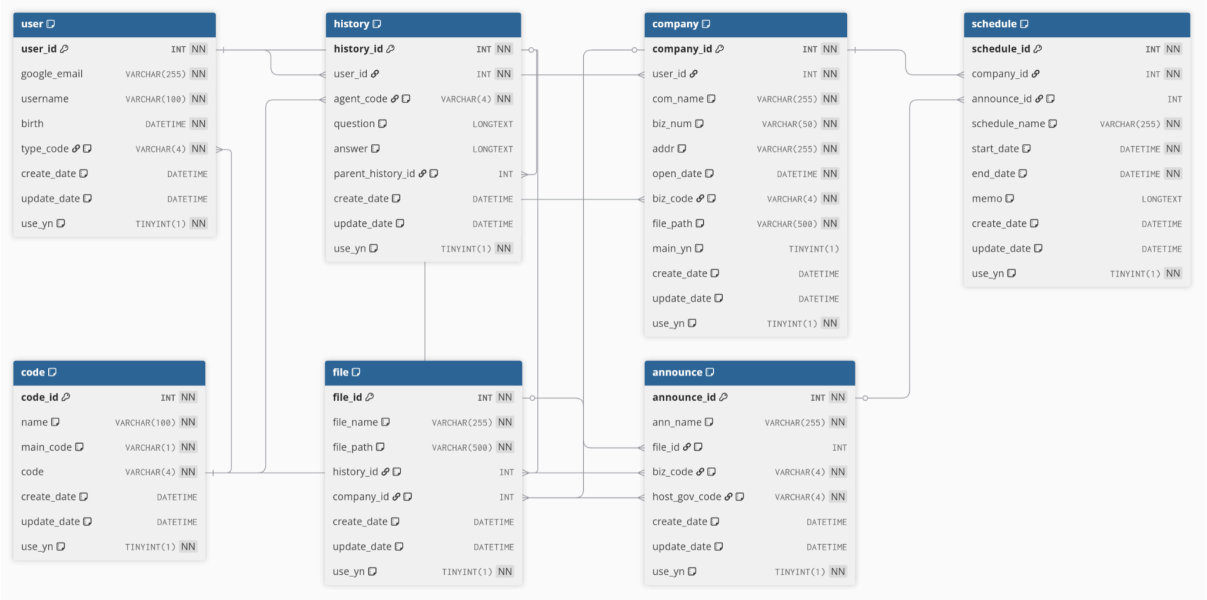
4. 상세 데이터 구조

4.1 전처리 완료 데이터 위치

```
data/processed/
├── laws/
│   ├── laws_full.jsonl          # 5,539건
│   └── law_lookup.json          # 법령명→ID 매핑
├── interpretations/
│   ├── smba_interp.jsonl        # 중소벤처기업부 500건
│   ├── labor_interp.jsonl       # 고용노동부 200건
│   ├── nts_interp.jsonl         # 국세청 200건
│   └── ftc_interp.jsonl         # 공정거래위원회 200건
├── guides/
│   ├── common_info.jsonl        # 공통 창업정보
│   ├── industries.jsonl         # 업종별 가이드 1,589건
│   └── pdf_guides.jsonl         # PDF 추출 섹션
└── schedules/
    └── tax_schedule.jsonl       # 세무일정 238건
```

4.2 MySQL 내부 구조 (ERD)

서비스 운영을 위한 관계형 데이터베이스 테이블을 설계합니다.



5. 주요 기능 설명 (예시)

5.1 조건별 조회

(1) 타입별 조회

기능 설명: 특정 데이터 타입(law, interpretation, guide, schedule)만 필터링하여 조회

```
import json

def query_by_type(file_path, target_type):
    """타입별 데이터 조회"""
    results = []

    try:
        with open(file_path, 'r', encoding='utf-8') as f:
            for line in f:
                data = json.loads(line)
                if data['type'] == target_type:
                    results.append(data)
    except FileNotFoundError:
        print(f"❌ 파일을 찾을 수 없습니다: {file_path}")
        return []
    except json.JSONDecodeError as e:
        print(f"❌ JSON 파싱 오류: {e}")
        return []

    return results

# 실행 예시
laws = query_by_type('data/processed/laws/laws_full.jsonl', 'law')
print(f"법령 데이터: {len(laws)}건 조회")
```

실행 결과:
```
법령 데이터: 5539건 조회
```

(2) 타입별 조회

기능 설명: 특정 비즈니스 도메인(legal, tax, labor, startup 등)의 데이터만 조회

```
def query_by_domain(file_path, target_domain):
    """도메인별 데이터 조회"""
    results = []

    try:
        with open(file_path, 'r', encoding='utf-8') as f:
```

```
            for line in f:
                data = json.loads(line)
                if data['domain'] == target_domain:
                    results.append(data)
    except Exception as e:
        print(f"❌ 조회 중 오류 발생: {e}")
        return []

    return results

# 실행 예시
labor_docs = query_by_domain('data/processed/laws/laws_full.jsonl', 'labor')
print(f"노동 관련 법령: {len(labor_docs)}건 조회")
```

실행 결과:
```
노동 관련 법령: 847건 조회
```

(3) 날짜 범위 조회

기능 설명**:** 특정 기간 내 시행/마감된 데이터를 조회

```
from datetime import datetime

def query_by_date_range(file_path, start_date, end_date):
    """날짜 범위 내 데이터 조회"""
    results = []
    start = datetime.strptime(start_date, '%Y-%m-%d')
    end = datetime.strptime(end_date, '%Y-%m-%d')

    try:
        with open(file_path, 'r', encoding='utf-8') as f:
            for line in f:
                data = json.loads(line)
                if data.get('effective_date'):
                    doc_date = datetime.strptime(data['effective_date'], '%Y-%m-%d')
                    if start <= doc_date <= end:
                        results.append(data)
    except Exception as e:
        print(f"❌ 날짜 조회 오류: {e}")
        return []

    return results

# 실행 예시
recent_laws = query_by_date_range(
    'data/processed/laws/laws_full.jsonl',
    '2024-01-01',
    '2024-12-31'
```

```
)
print(f"2024년 시행 법령: {len(recent_laws)}건")
```

실행 결과:
```
2024년 시행 법령: 342건
```

문서 버전: 1.0.2

최종 수정일: 2026년 1월 27일