MasterDB 온톨로지 기반 구현 계획서

작성일: 2024-12-24 **버전**: 2.0 **Phase**: Phase 1 - SQLite 기반 프로토타입 **[완료]**

> **설계 참조 문서** (패턴/명세 참조용):
>
> - DATABASE_DESIGN.md - 테이블 구조, ID 체계, 인덱스 전략
> - DB_테이블명세서.xlsx - 테이블별 컬럼 상세 명세
> - database_schema.md - 신규 스키마 상세

---

1. Phase 1 구현 완료 요약

1.1 구현 결과

```
┌─────────────────────────────────────────────────────────┐
│                  Phase 1 구현 완료                      │
├─────────────────────────────────────────────────────────┤
│                                                         │
│  [Step 1] SQLite DB 생성 및 데이터 마이그레이션    ✓ 완료   │
│  [Step 2] 레거시 분류체계 기반 초기 Taxonomy 구축  ✓ 완료   │
│  [Step 3] 자동 태깅 시스템 구현                    ✓ 완료   │
│  [Step 4] 쿼리 인터페이스 및 검증                  ✓ 완료   │
│                                                         │
└─────────────────────────────────────────────────────────┘
```

1.2 데이터베이스 현황

테이블	건수	설명
companies	131	고객사
surveys	349	설문 프로젝트
questions	7,343	전체 문항
master_questions	3,274	대표 문항 (클러스터 대표)
embeddings	7,343	KoSBERT 임베딩 (768차원)
taxonomy	38	분류 용어
taxonomy_relations	43	용어 간 관계
question_tags	21,402	문항-태그 매핑

데이터베이스 파일: `db/masterdb.sqlite` (31.74 MB)

1.3 Taxonomy 구조

Taxonomy는 문항들을 분류하는 **계층적 용어 체계**입니다.

```
Taxonomy 계층 구조
==================

THEME (4개) - 최상위: 진단 유형
├─ 조직진단 (OD)     → 5,179 문항
├─ 리더십진단 (LD)   → 1,590 문항
├─ 다면평가 (MA)     → 364 문항
└─ 사외이사평가 (DD) → 210 문항

CONCEPT (8개) - 중간: 중분류 개념
├─ 리더십          → 2,421 문항
├─ 조직프로세스    → 2,456 문항
├─ 비전전략        → 483 문항
├─ 인사제도        → 898 문항
├─ 조직문화        → 420 문항
├─ 경영전략        → 47 문항
├─ 구성원몰입      → 240 문항
└─ 기타            → 378 문항

ASPECT (26개) - 하위: 소분류 관점
├─ 리더십일반 (535), 코칭육성 (252), 소통경청 (384)
├─ 동기부여 (136), 권한위임 (107), 배려관계 (96)
├─ 공정성 (71), 신뢰 (64), 역할책임 (164)
├─ 조직구조 (942), 업무프로세스 (806), 의사결정 (320)
├─ 업무효율성 (144), 협업팀워크 (113), 고객서비스 (76)
├─ 목표방향 (423), 목표KPI (152), 이해실천 (107)
├─ 변화혁신 (309), 평가제도 (223), 보상제도 (284)
├─ 승진제도 (173), 교육연수 (108), 조직문화유형 (212)
├─ 조직몰입 (231), 경영일반 (265)
```

Taxonomy 관계 예시:

```
조직진단 (THEME)
├── HAS_COMPONENT → 조직프로세스 (CONCEPT)
│   ├── HAS_COMPONENT → 조직구조 (ASPECT)
│   ├── HAS_COMPONENT → 업무프로세스 (ASPECT)
│   └── HAS_COMPONENT → 의사결정 (ASPECT)
├── HAS_COMPONENT → 리더십 (CONCEPT)
│   ├── HAS_COMPONENT → 리더십일반 (ASPECT)
│   ├── HAS_COMPONENT → 코칭육성 (ASPECT)
│   └── HAS_COMPONENT → 소통경청 (ASPECT)
└── HAS_COMPONENT → 인사제도 (CONCEPT)
    ├── HAS_COMPONENT → 평가제도 (ASPECT)
    ├── HAS_COMPONENT → 보상제도 (ASPECT)
    └── HAS_COMPONENT → 승진제도 (ASPECT)
```

1.4 Question Tags 구조

각 문항은 3가지 유형의 태그를 가집니다:

태그 유형	개수	설명
themes	7,343	진단 유형 (OD, LD, MA, DD)
concepts	7,343	중분류 개념 (리더십, 조직프로세스 등)
aspects	6,716	소분류 관점 (리더십일반, 코칭육성 등)

태깅 예시:

```
문항: "우리 회사의 비전이 무엇인지 이해하고 있다"

태그:
  - themes: 조직진단 (confidence: 1.0)
  - concepts: 비전전략 (confidence: 1.0)
  - aspects: 목표방향 (confidence: 1.0)
```

---

2. 구현된 모듈

2.1 프로젝트 구조

```
c:\Project\MasterDB\
├── db/
│   └── masterdb.sqlite           # SQLite 데이터베이스 (31.74 MB)
│
├── src/
│   ├── db/                       # DB 모듈
│   │   ├── __init__.py
│   │   ├── connection.py         # DB 연결 관리
│   │   └── schema.py             # 스키마 정의 (13 tables, 50 indexes)
│   │
│   ├── migration/                # 마이그레이션 모듈
│   │   ├── __init__.py
│   │   ├── run_migration.py      # 전체 마이그레이션 실행
│   │   ├── migrate_questions.py  # 문항 마이그레이션
│   │   ├── migrate_surveys.py    # 설문 마이그레이션
│   │   └── init_taxonomy.py      # Taxonomy 초기화
│   │
│   ├── tagging/                  # 태깅 모듈
│   │   ├── __init__.py
│   │   ├── embedding_search.py   # 임베딩 기반 유사도 검색
│   │   └── auto_tagger.py        # 자동 태깅 시스템
│   │
│   └── query/                    # 검색 모듈
│       ├── __init__.py
│       └── question_search.py    # 검색 API
│
```

```
├── data/
│   └── Survey Meta Data_251224.xlsx  # 원본 설문 메타데이터
│
├── reference/                    # 참조 파일 (Phase 0 산출물)
│   ├── Master_Questions.xlsx     # 3,274개 대표 문항
│   └── 전체_문항_클러스터링_Hybrid.xlsx
│
├── all_df_hybrid.pkl             # 문항 DataFrame 캐시
├── all_embeddings_hybrid.npy     # 임베딩 벡터 캐시
│
├── CLAUDE.md                     # 프로젝트 컨텍스트
├── VERIFICATION_REPORT.md        # Phase 1 검증 보고서
└── MasterDB_Implementation_Plan.md  # 본 문서
```

2.2 주요 모듈 설명

2.2.1 DB 모듈 (`src/db/`)

connection.py - DB 연결 관리

```
from db.connection import get_connection, get_db_info

conn = get_connection()  # SQLite 연결
info = get_db_info(conn)  # DB 정보 (크기, 인덱스 수 등)
```

schema.py - 스키마 정의

- 13개 테이블 정의 (companies, surveys, questions, master_questions, embeddings, taxonomy, taxonomy_relations, question_tags, scales, scale_questions, org_units, org_unit_surveys, survey_questions)
- 50개 인덱스 정의
- `create_all_tables(conn)` - 테이블 생성
- `get_table_counts(conn)` - 테이블별 행 수 조회

2.2.2 마이그레이션 모듈 (`src/migration/`)

run_migration.py - 전체 마이그레이션

```
python src/migration/run_migration.py
```

- pkl/npy 파일에서 questions, master_questions, embeddings 마이그레이션
- Excel에서 companies, surveys 마이그레이션

init_taxonomy.py - Taxonomy 초기화

```
python src/migration/init_taxonomy.py
```

- 레거시 중분류/소분류에서 Taxonomy 용어 추출
- Theme → Concept → Aspect 계층 구조 생성
- question_tags 초기 매핑 생성

2.2.3 태깅 모듈 (`src/tagging/`)

embedding_search.py - 임베딩 기반 검색

```
from tagging.embedding_search import EmbeddingSearch

searcher = EmbeddingSearch()

# 유사 문항 검색
results = searcher.search_by_question("Q_00001", top_k=5)

# 클러스터 멤버 조회
members = searcher.find_cluster_members("OD_0001")
```

auto_tagger.py - 자동 태깅

```
from tagging.auto_tagger import AutoTagger

tagger = AutoTagger()

# 기존 태그 조회
tags = tagger.get_question_tags("Q_00001")

# 태그 제안 (유사 문항 + 클러스터 기반)
suggestions = tagger.suggest_tags("Q_00001")

# 태그 적용
tagger.apply_tag("Q_00001", term_id=5, tag_type="concepts")
```

2.2.4 검색 모듈 (`src/query/`)

question_search.py - 검색 API

```
from query.question_search import QuestionSearch

search = QuestionSearch()

# 텍스트 검색
results = search.search_by_text("리더", diagnosis_type="OD")

# Taxonomy 기반 검색
```

```
results = search.search_by_taxonomy(term="리더십", term_type="CONCEPT")

# 대표문항 + 변형문항 조회
master = search.get_master_question_with_variants("OD_0001")

# Taxonomy 계층 조회
tree = search.get_taxonomy_tree("조직진단")

# 통계 조회
stats = search.get_statistics()
```

---

3. 데이터 흐름

3.1 문항 데이터 파이프라인

```
원본 데이터                          마이그레이션                    DB 테이블
─────────────────────────────────────────────────────────────────────────────
all_df_hybrid.pkl          →    run_migration.py      →    questions (7,343)
(7,343 문항)                                                 master_questions (3,274)

all_embeddings_hybrid.npy →    run_migration.py      →    embeddings (7,343)
(7,343 × 768)                                                (BLOB 형태로 저장)

Survey Meta Data.xlsx      →    run_migration.py      →    companies (131)
                                                             surveys (349)
```

3.2 Taxonomy 구축 파이프라인

```
레거시 분류체계                      Taxonomy 생성                  DB 테이블
─────────────────────────────────────────────────────────────────────────────
questions.legacy_mid_category  →                      →    taxonomy (38)
questions.legacy_sub_category  →  init_taxonomy.py    →    taxonomy_relations (43)
questions.diagnosis_type       →                      →    question_tags (21,402)
```

3.3 검색 데이터 흐름

```
사용자 쿼리                          검색 엔진                      결과
─────────────────────────────────────────────────────────────────────────────
"리더십 문항 검색"          →    QuestionSearch          →    관련 문항 목록
                                  ├─ search_by_text()
                                  ├─ search_by_taxonomy()
                                  └─ get_master_question_with_variants()

"Q_00001과 유사한 문항"     →    EmbeddingSearch         →    유사도 순 문항 목록
```

```
                                    ├─ search_by_question()
                                    └─ cosine similarity 계산
```

---

4. ID 체계

테이블	ID 형식	예시	설명
companies	{CODE}	CJG, SKH	회사 코드
surveys	{CODE}-{YEAR}-{TYPE} 또는 IG{...}	IG200601UJULD	설문 코드
questions	Q_{5digits}	Q_00001	문항 일련번호
master_questions	{TYPE}_{4digits}	OD_0001	대표문항 ID
taxonomy	{auto_increment}	1, 2, 3	자동 증가

---

5. Phase 2 계획

5.1 다음 단계 작업

우선순위	작업	설명
1	Taxonomy 확장	수동 추가/편집 인터페이스
2	자동 태깅 개선	키워드 기반 태깅 추가, 정확도 향상
3	척도(Scale) 정의	scales, scale_questions 테이블 활용
4	survey_questions 매핑	설문-문항 매핑 데이터 구축
5	웹 기반 관리 UI	검색/편집 인터페이스

5.2 PostgreSQL 전환 (Phase 3)

```
Phase 3 목표:
├── SQLite → PostgreSQL 마이그레이션
├── JSONB 네이티브 지원 활용
├── 전문 검색 (Full-text Search) 구현
├── 벡터 검색 확장 (pgvector)
└── API 서버 구축
```

---

6. 참고 문서

문서	설명
CLAUDE.md	프로젝트 컨텍스트 (Claude Code용)
VERIFICATION_REPORT.md	Phase 1 검증 보고서

문서	설명
Survey_Question_Analysis.md	문항 분석 과정 기록
database_schema.md	상세 스키마 정의
DATABASE_DESIGN.md	설계 참조 문서

Last Updated: 2024-12-24 Phase 1 구현 완료