데이터 모델링 방법론

KANDLE Data Standardization Methodology

데이터에 질서를, 비즈니스에 통찰을



차 례

Preface 7

I KDSM 개요

1 KDSM 개요 11
1.1 Methodology Structure 11
1.2 Methodology Layer 12
1.3 Methodology Component 15

II Foundation Layer

2 데이터 모델링 프로세스 23
2.1 아키텍처 23
2.2 프로세스 23
2.3 산출물 흐름도 26

III Guidance Layer

3 표준 및 절차 매뉴얼 29
3.1 개요 29
3.2 Ⓟ 요구사항분석 31
3.3 Ⓟ 표준단어 정의 36
3.4 Ⓟ 표준용어 정의 39
3.5 Ⓟ 표준도메인 정의 42

3.6 Ⓟ 표준코드 정의 . . . 45
3.7 Ⓟ 표준화 확정 . . . 48
3.8 Ⓟ 표준화 적용 . . . 51
3.9 부록 . . . 56

4 산출물 템플릿 . . . 59
4.1 개요 . . . 59
4.2 템플릿 체계 . . . 59
4.3 요구사항분석 산출물 템플릿 . . . 61
4.4 데이터 표준 정의 산출물 템플릿 . . . 62
4.5 표준화 확정·공표 산출물 템플릿 . . . 68
4.6 표준화 적용 산출물 템플릿 . . . 68
4.7 표준화 관리 산출물 템플릿 . . . 75
4.8 템플릿 운영 지침 . . . 76

IV Support Layer

5 가이드 . . . 81
5.1 개요 . . . 81
5.2 표준단어 작성 가이드 . . . 83
5.3 표준용어 작성 가이드 . . . 87
5.4 표준도메인 작성 가이드 . . . 91
5.5 표준코드 작성 가이드 . . . 95
5.6 명명규칙 . . . 98
5.7 논리 데이터 모델링 가이드 . . . 102
5.8 물리 데이터 모델링 가이드 . . . 105
5.9 코딩 가이드 . . . 107

6 테크니컬 노트 . . . 119
6.1 개요 . . . 119
6.2 테크니컬 노트 체계 . . . 120
6.3 데이터 표준 정의 테크니컬 노트 . . . 121
6.4 데이터 모델 정의 테크니컬 노트 . . . 134

7 라이브러리 . . . 149
7.1 개요 . . . 149
7.2 라이브러리 체계 . . . 151
7.3 설치와 환경 준비 . . . 153

7.4 데이터 표준 정의 지원 . . . 156
7.5 데이터 모델 정의 지원 . . . 161
7.6 운영 단계의 진단 . . . 165
7.7 작업 흐름과 자동화 . . . 168

8 툴킷 . . . 173
8.1 개요 . . . 173
8.2 툴킷 체계 . . . 174
8.3 데이터 표준 정의 지원 . . . 177
8.4 데이터 모델 정의 지원 . . . 183
8.5 산출물의 생성과 적재 . . . 191
8.6 점검 . . . 192
8.7 관리 . . . 196
8.8 구축과 운영 . . . 197

V Experience Layer

9 경험 자산 . . . 203
9.1 개요 . . . 203
9.2 경험 자산의 체계 . . . 204
9.3 방법론 포털 . . . 205
9.4 데이터 표준화 저서 . . . 206
9.5 지식 베이스 . . . 207
9.6 경험 자산의 운영 . . . 209

용어 사전 . . . 211

실무 체크리스트 . . . 225

참고문헌 . . . 231

Preface

데이터 모델링 방법론에 대하여

데이터는 현대 조직의 핵심 자산입니다. 그러나 데이터의 가치는 그것이 **얼마나 잘 표준화되어 있는가**에 크게 좌우됩니다. 서로 다른 시스템, 부서, 조직 간에 데이터가 일관성 없이 관리될 때 발생하는 문제들 — 중복, 오류, 해석의 불일치 — 은 의사결정의 질을 떨어뜨리고 디지털 혁신을 가로막는 주요 장애물이 됩니다.

데이터 모델링 방법론은 데이터 표준화의 **이론적 기반**부터 **실무 적용**까지를 체계적으로 다룹니다.

방법론의 구성

데이터 모델링 방법론은 4 Layer로 구성되어 있습니다.

Foundation Layer는 데이터 표준화 방법을 사전에 준수하도록 통제하는 핵심 체계로, "표준화 목표를 달성하기 위해 수행해야 하는 **일련의 작업 절차와 흐름"을 정의**합니다.

Guidance Layer는 "표준화 목적 및 방향성 통제(Alignment), 품질과 다양성의 균형 유지(Trade-off Control), 실무적 지침 및 의사결정 지원(Decision Support), 일관성 및 신뢰성 보장(Standardization)을 가이드" 하는 **기준과 절차를 정의**합니다.

Support Layer는 "단순한 규칙 제정을 넘어, 실무에서의 품질 보장과 작업 생산성을 극대화하는 핵심 기반"으로 "**작업을 위한 규칙, 자동화 도구, 기술문서 등으로 구성**"됩니다.

Experience Layer는 "데이터 표준화에 대한 실무자의 노하우와 조직의 과거 데이터 관리 이력"으로 "**데이터 표준화 활동의 성공과 시행착오에 대한 명시적 지식 체계**"입니다.

독자 대상

> ℹ 이 데이터 모델링 방법론은 다음 담당자를 위해 만들어졌습니다
>
> - 데이터 아키텍처 및 모델링 담당자
> - IT 관리자 및 CIO
> - 데이터 모델링을 처음 접하는 실무자

표기 규칙

데이터 모델링 방법론에서 사용하는 표기 규칙은 다음과 같습니다.

- `고정폭 글꼴`: 코드, 파일명, 명령어
- **굵은 글꼴**: 핵심 용어의 첫 등장
- *기울임꼴*: 외래어, 강조

> **팁**
>
> 실무에서 바로 활용할 수 있는 유용한 정보를 제공합니다.

> **주의**
>
> 주의해야 할 사항이나 흔히 발생하는 실수를 알려줍니다.

> **중요**
>
> 반드시 숙지해야 할 핵심 내용입니다.

> **참고**
>
> 주요한 참고 내용을 함축해서 알려줍니다.

I

KDSM 개요

1 KDSM 개요 11
1.1 Methodology Structure 11
1.2 Methodology Layer 12
1.3 Methodology Component 15

CHAPTER 1

KDSM 개요

디플래닉스가 개발한 KDSM(KANDLE Data Standardization Methodology)은 국내 조직의 데이터 표준화 실무에 최적화된 **데이터 표준화 방법론**이다. 이 방법론은 **데이터 모델링 방법론**을 다룬다.

이 장에서는 데이터 모델링 방법론을 중심으로 디플래닉스 데이터 표준화 방법론인 `KDSM`의 철학과 어프로치를 다룬다.

1.1 Methodology Structure

1.1.1 “4 Layered Architecture”

- Methodology Structure는 Foundation 〉 Guidance 〉 Support 〉 Experience의 `4 Layered Architecture`로 구성된다

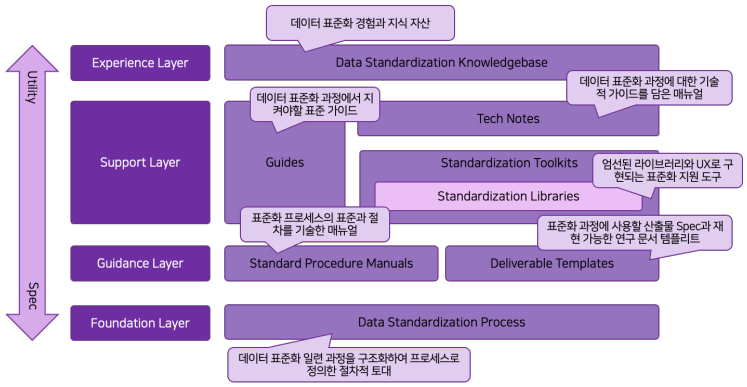

그림 1.1. 방법론 구조: 4 Layered Architecture

1.1.2 Layer별 Components 기능 소개

그림 1.2 에서 Layer별 기능 소개 테이블을 통해 데이터 모델링 방법론의 기능을 파악할 수 있다.

Layers	Components	Components(KO)	Functions	형상	예시
Foundation	Data Standardization Process	데이터 표준화 프로세스	데이터 표준화 프로세스의 정의	Process Diagram	데이터 표준화 프로세스, 데이터 표준화 관리 프로세스
Guidance	Standard Procedure Manuals	표준 및 절차 매뉴얼	데이터 표준화의 표준과 절차 제시	Documents	데이터 표준화 표준 및 절차 매뉴얼
Guidance	Deliverable Templates	산출물 템플릿	표준 산출물 템플릿 제공	Documents	테이블정의서, 컬럼정의서, 도메인정의서
Support	Guides	가이드	데이터 표준화 과정의 표준 가이드 제시	Documents	명명규칙, 단어(약어) 사전, 코딩가이드
Support	Standardization Library	표준화 라이브러리	데이터 표준화를 위한 라이브러리 제공	Python Library, SQL Function	데이터 품질 진단, 데이터 클렌징, 데이터 표준화 진단, 데이터 정합성 진단
Support	Standardization Toolkits	표준화 툴 박스	데이터 표준화를 위한 분석 툴 박스 제공	Toolkit / APP	단어(약어) 추천 앱 데이터 표준화 진단 앱
Support	Tech Notes	테크니컬 노트	데이터 표준화 테크니컬 노트 제공		표준사전(단어/용어,도메인, 코드) 정의 방법서
Experience	Standardization Knowledgebase	데이터 표준화 지식 베이스	데이터 표준화 명시적 경험/지식 제공	Knowledge Base	공공데이터 메타설계 소개서

그림 1.2. Layer별 구성 요소의 기능

1.2 Methodology Layer

1.2.1 Layer의 역할

- 하층부는 방법론의 기본 골격이 되는 Spec Layer이며,
  - Spec은 작업 절차, 기준, 요건과 같은 필요한 특성 정의를 의미한다
- 상층으로 갈수록 Utility 성향을 띤다.
  - Utility는 작업에 필요한 구체적인 형상이 있는 문서나 프로그램을 의미한다

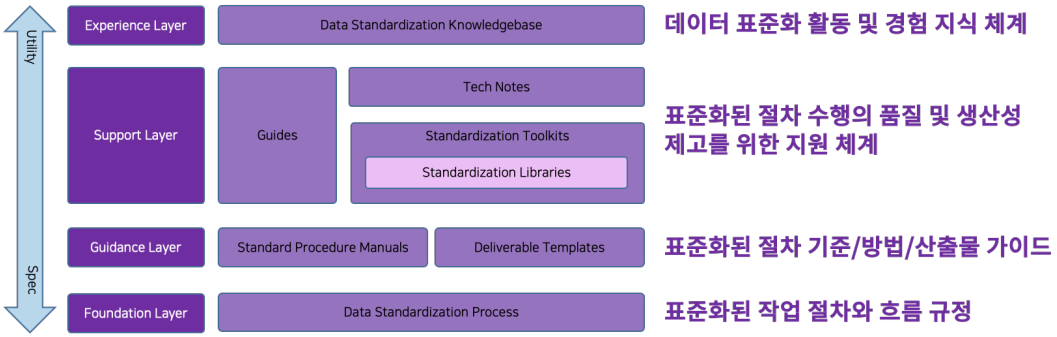

그림 1.3. Layer 계층별 역할

1.2.2 Foundation Layer

Foundation Layer는 데이터 표준화 방법을 사전에 준수하도록 통제하는 핵심 체계로, “**표준화 목표를 달성하기 위해 수행해야 하는 일련의 작업 절차와 흐름**”을 정의한다.

1.2.2.1 구조

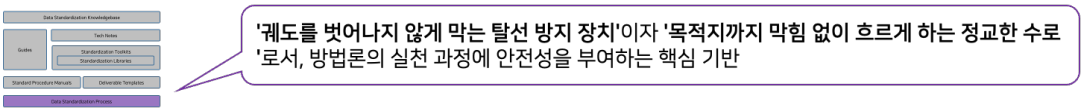

그림 1.4. 구조: Foundation Layer

1.2.2.2 구성 요소

컴포넌트	하위 컴포넌트	기능	비고
Data Standardization Process	Data Modeling Process	데이터 표준화 수행 프로세스의 정의	데이터 모델 관리
Data Standardization Process	Data Quality Management Process	데이터 품질 관리 프로세스 정의	데이터 품질 관리
Data Standardization Process	Meta Data Management Process	메타데이터 관리 프로세스 정의	메타데이터 관리

그림 1.5. 구성 요소: Foundation Layer

1.2.2.3 구성 요소 관계

데이터 모델링 방법론을 구성하는 요소의 관계를 도식으로 표현한다.

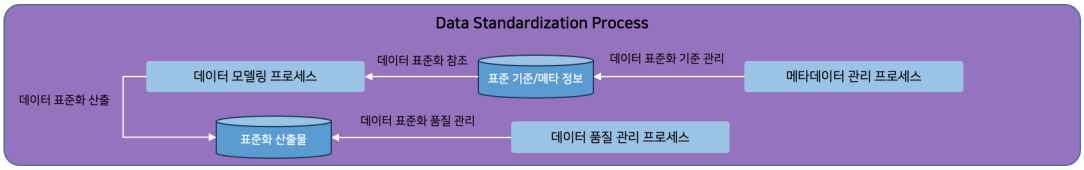

그림 1.6. 구성 요소 관계: Foundation Layer

> **i** 노트
>
> Foundation Layer의 구성 요소는 향후 확장할 수 있는 데이터 품질관리 프로세스, 메타데이터 관리 프로세스를 포함하여 설명한다.

1.2.3 Guidance Layer

Guidance Layer는 **표준화 목적 및 방향성 통제(Alignment), 품질과 다양성의 균형 유지(Trade-off Control), 실무적 지침 및 의사결정 지원(Decision Support), 일관성 및 신뢰성 보장(Standardization)을 가이드**하는 기준과 절차를 정의한다.

1.2.3.1 구조

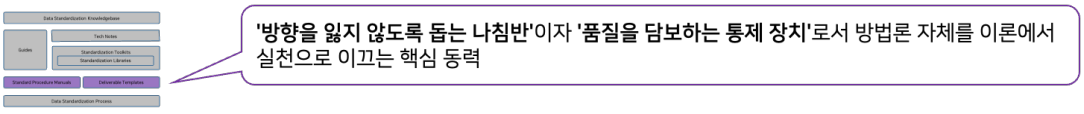

그림 1.7. 구조: Guidance Layer

1.2.3.2 구성 요소

Components	하위 컴포넌트	기능	비고
Standard Procedure Manuals	-	데이터 표준화의 표준과 절차 기준 정의	표준화 기준/절차
Deliverable Templates	-	데이터 표준화 과정에서의 표준 산출물 템플릿 제공	산출물의 표준화

그림 1.8. 구성 요소: Guidance Layer

1.2.3.3 구성 요소 관계

`Standard Procedure Manuals`는 `Data Standardization Process`에서 정의한 프로세스에 대해서 구체화되고 표준화된 절차를 제시하고, 이 과정에서 필요한 산출물은 `Deliverable Templates`에서 표준 템플릿으로 제공한다.

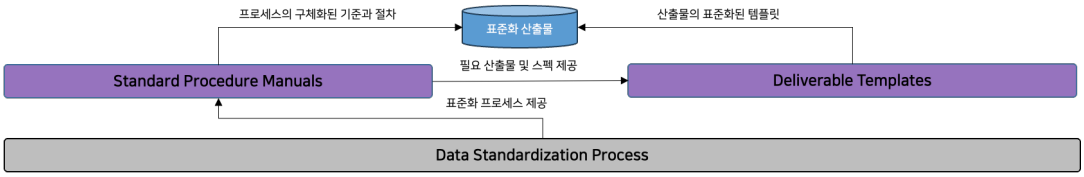

그림 1.9. 구성 요소 관계: Guidance Layer

1.2.4 Support Layer

Support Layer는 "단순한 규칙 제정을 넘어, **실무에서의 품질 보장과 작업 생산성을 극대화하는 핵심 기반**"으로 **작업을 위한 규칙, 자동화 도구, 기술문서 등으로 구성**한다.

1.2.4.1 구조

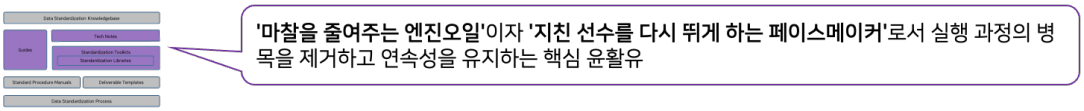

그림 1.10. 구조: Support Layer

1.2.4.2 구성 요소

Components	하위 컴포넌트	기능	비고
Guides	-	데이터 표준화 과정의 표준 가이드 제시	표준화 가이드
Tech Notes	-	데이터 표준화 테크니컬 노트 제공	표준화 작업 길잡이
Standardization Library	Standardization Toolkits	데이터 표준화를 위한 자동화 라이브러리/툴 박스 제공	표준화 작업 자동화

그림 1.11. 구성 요소: Support Layer

1.2.4.3 구성 요소 관계

`Guides`는 `Standard Procedure Manuals`에서 정의한 절차에 대해서 **작업 기준과 가이드를 제시**하고, `Standardization Library / Toolkits`은 작업 과정에서의 산출물 스팩과 기준에 부합하는 산출물 생성을 돕거나 자동화 생산한다. `Tech Notes`는 이러한 과정을 수행할 수 있는 자동화 도구의 운용방법을 제시한다.

개별 Component의 목적은 다음과 같다.

- Guides: 표준화된 산출물의 기준 제시
- Standardization Library / Toolkits: 산출물 생산 생산성 제고
- Tech Notes: 산출물 생산의 생산성 제고 및 품질 균질화

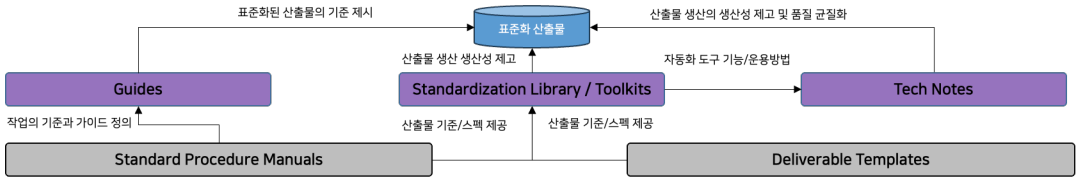

그림 1.12. 구성 요소 관계: Support Layer

1.2.5 Experience Layer

Experience Layer는 "**데이터 표준화에 대한 실무자의 노하우와 조직의 과거 데이터 관리 이력**"으로 "**데이터 표준화 활동의 성공과 시행착오에 대한 명시적 지식 체계**"다.

1.2.5.1 구조

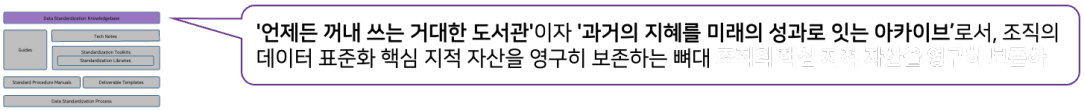

그림 1.13. 구조: Experience Layer

1.2.5.2 구성 요소

Components	하위 컴포넌트	기능	비고
Knowledgebase	-	데이터 표준화 명시적 경험/지식 제공	표준화 경험 자산

그림 1.14. 구성 요소: Experience Layer

1.2.5.3 구성 요소 관계

`Data Standardization Knowledgebase`는 `Foundation Layer`, `Guidance Layer`, `Support Layer`의 Components를 운용하면서 체득한 **경험 자산**과 **지식 자산**의 총체다.

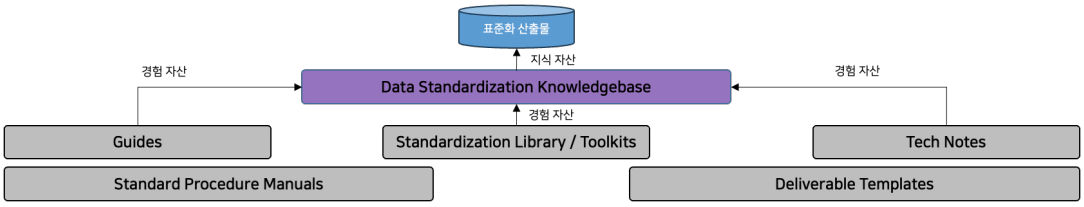

그림 1.15. 구성 요소 관계: Experience Layer

1.3 Methodology Component

1.3.1 Data Standardization Process

프로세스(Process)는 **목적 달성을 위한 작업의 흐름과 절차를 정의**한다. 전체적인 방법론의 뼈대를 이루며, **'누가, 언제, 무엇을(Who, When, What)' 수행해야 하는지 명확한 단계와 기준을 제공**하여 표준화 활동의 성공을 이끄는 핵심적인 역할을 한다.

1.3.1.1 구성

`Data Standardization Process`는 프로세스의 구조와 과정들의 관계를 설명하는 `Architecture`, 개별 프로세스를 정의한 `Process`, 프로세스 과정별로 유기적으로 생성/참조되는 산출물 흐름과 R&R을 정의한 `Deliverable Flow Diagram`로 구성된다.

구성 내역	역할	기능	비고
Architecture	Data Standardization Process 구조 소개	프로세스의 구조와 관계 정의	프로세스 구조
Process	표준화 작업 프로세스 제시	Phase > Activity > Task 레벨의 프로세스 정의	프로세스 정의
Deliverable Flow Diagram	역할별 협업 구조 및 산출물 제시	제시 표준화 과정에서의 역할별 표준화 산출물 생성 흐름 정의	협업과 산출물

그림 1.16. 구성: Data Standardization Process

1.3.1.2 목적물 예시

- **데이터 표준화 프로세스 정의서**: 요구사항분석 → 데이터 표준화 정의 → 데이터 표준화 확정 → 데이터 표준화 적용의 전체 라이프사이클을 Phase → Activity → Task의 3계층(3 Depth Breakdown)으로 분해하여 정의한 문서. 본서의 「데이터 모델링 프로세스」 장이 이 목적물에 해당한다.
- Deliverable Flow Diagram: Task 간 입력물–산출물의 연쇄와 R&R을 한눈에 표현한 산출물 흐름도. "앞 Task의 산출물이 다음 Task의 입력물이 된다"는 프로세스의 연결 구조를 검증하는 도구가 된다.

1.3.2 Standard Procedure Manuals

표준 및 절차 매뉴얼(Standard Procedure Manuals)은 **복잡한 업무와 프로젝트를 표준화하여 일관성, 효율성, 그리고 품질 안정성을 보장하는 핵심 지침**이다.

1.3.2.1 구성

`Standard Procedure Manuals`은 개별 프로세스별로 기술된 매뉴얼로 구성된다.

구성 내역	역할	기능	비고
표준과 절차 매뉴얼	프로세스를 수행하는 표준과 절차 제시	프로세스와 매칭된 표준화 작업의 지침서	표준화 지침

그림 1.17. 구성: Standard Procedure Manuals

1.3.2.2 목적물 예시

- **데이터 모델링 표준 및 절차 매뉴얼**: 프로세스의 각 Task를 개요 → 표준 기준 → 단계별 절차 → 입력물·산출물 → 역할과 책임(R&R) → 체크리스트 → 예시의 정형 구조로 기술하여, 수행자가 누구든 동일한 품질로 작업을 반복할 수 있게 한 매뉴얼. 본서의 「표준 및 절차 매뉴얼」 장이 이 목적물에 해당한다.

1.3.3 Deliverable Templates

산출물 템플릿(Deliverable Templates)은 **프로세스 각 단계별로 생성해야 하는 결과물의 표준화된 양식**이다. **작업의 품질을 상향 평준화하고, 정보 누락을 방지하며, 데이터 표준화 관계자 간의 원활한 의사소통을 돕는 핵심적인 역할**을 수행한다.

1.3.3.1 구성

템플릿은 파일로 제공되는 템플릿과 데이터베이스의 테이블로 제공되는 산출물로 구성된다.

구성 내역	역할	기능	비고
산출물 템플릿 파일	단계별로 생성해야 할 표준 산출물 제시	표준 산출물의 템플릿 및 작성 가이드	표준화 산출물
산출물 테이블	단계별로 시스템에서 생성되는 산출물 제시	라이브러리/앱으로 생성되는 테이블 산출물 구성 및 가이드	표준화 데이터베이스

그림 1.18. 구성: Deliverable Templates

1.3.3.2 목적물 예시

데이터 모델링 과정에서 도출해야 할 산출물은 목록화되어 템플릿으로 제공된다.

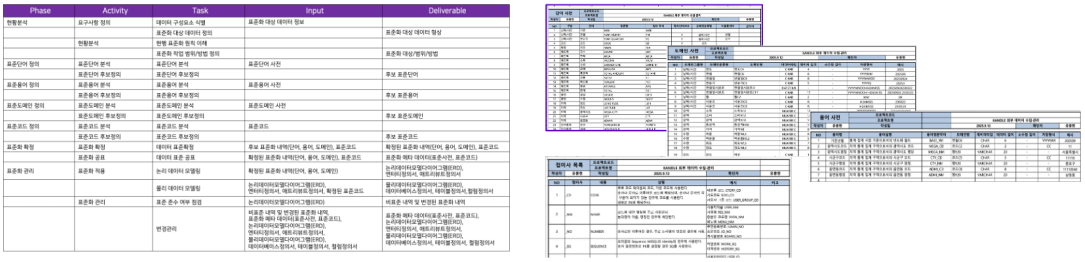

그림 1.19. 목적물 예시: Deliverable Templates

1.3.4 Guides

지침/규칙(Guides)은 **목적을 이루기 위해 따라야 할 방향성과 권장 사항**이며 **반드시 준수해야 하는 엄격한 제약이자 기준**이다. **지침은 비효율적인 시행착오를 줄여주고, 규칙은 치명적인 오류를 막아준다.**

1.3.4.1 구성

Guides는 지침과 규칙으로 구성된다.

구성 내역	역할	기능	비고
지침	표준화 작업의 방향성과 권장사항	표준화 작업의 방향성과 권장사항 가이드	표준화 가이드
규칙	표준화 과정에서 준수해야 하는 제약 기준	표준화 과정에서 준수해야 하는 지침	표준화 지침

그림 1.20. 구성: Guides

1.3.4.2 목적물 예시

명명규칙, 코딩 가이드, 표준사전(단어, 용어, 도메인) 등이 지침/규칙의 예다.

구분	명칭	기능	비고
지침	명명규칙	단어/용어, 테이블, 컬럼, 함수 등의 명명규칙	표준화된 명칭
지침	코딩 가이드	SQL, 프로그램 등의 코딩 가이드	코딩 가이드
규칙	표준사전(단어, 용어, 도메인)	신규 데이터 모델링에서 참조/추가해야 할 표준사전(단어, 용어, 도메인)	표준화 진단

그림 1.21. 목적물 예시: Guides

1.3.5 Tech Notes

기술 노트(Tech Notes)는 **현장에 적용하기 위한 실무 지침, 트러블슈팅, 세부 구현 방법을 담은 핵심 문서다. 이론적 프레임워크와 실무 실행 사이의 간극을 메우는 '실행 가이드' 역할을 수행**한다.

1.3.5.1 구성

구성 내역	역할	기능	비고
Technical Guide	주요 데이터 표준화를 위한 기술적인 가이드	데이터 표준화 수행을 위한 기술 가이드	표준화 가이드
User Guide/Manual	표준화를 위한 라이브러리/툴 박스 사용 가이드	Standardization Library / Toolkits 매뉴얼	라이브러리 매뉴얼

그림 1.22. 구성: Tech Notes

1.3.5.2 목적물 예시

- **도구 운용 가이드**: 표준사전 관리 도구와 모델링 도구의 설치·설정, 표준 템플릿 적용 절차를 다루는 문서
- **자동화 점검 노트**: 표준 준수 자동 점검 규칙의 사용법과 점검 결과의 해석, 자주 발생하는 오류의 트러블슈팅
- **산출물 생성 파이프라인 노트**: 정의서·매뉴얼 등 산출물을 조판 품질의 문서로 자동 생성하는 파이프라인의 운용 방법

1.3.6 Standardization Library / Toolkits

자동화 라이브러리/툴 박스(Library / Toolkits)는 **데이터 표준화 과정에서의 주요 작업을 지원하는 소프트웨어**다. **작업의 생산성을 높이고 표준화된 산출물을 만드는 품질관리 역할**을 수행한다.

1.3.6.1 구성

산출물을 직접 생산하거나 도움을 주는 **함수의 모음인 Library**와 사용자에게 **UI/UX를 제공하는 Toolkits**으로 구성된다.

구성 내역	역할	기능	비고
Standardization Library	CLI 기반 데이터 표준화 자동화 및 산출물 생산	데이터 표준화 단위 과정을 지원하는 라이브러리	표준화 라이브러리
Standardization Toolkits	UI/UX 기반 데이터 표준화 자동화 및 산출물 생산	데이터 표준화 단위 과정을 지원하는 툴 박스	표준화 앱

그림 1.23. 구성: Standardization Library / Toolkits

1.3.6.2 목적물 예시

데이터 모델링 과정에서 산출물을 생산/관리하는 패키지와 앱이 Standardization Library / Toolkits 목적물의 하나다.

구분	명칭	기능	비고
Library	데이터 표준화 라이브러리	표준사전 관리, 데이터 표준화 관련 패키지	표준사전 생성/관리
Toolkits / App	데이터 표준사전 생성 앱	표준사전(단어, 용어, 도메인) 추천 및 검색/생성/관리 기능	표준사전 생성/관리
Toolkits / App	데이터 표준화 진단 앱	데이터 표준화(데이터베이스의 테이블/컬럼 카탈로그 정보 기반) 진단	표준화 진단

그림 1.24. 목적물 예시: Standardization Library / Toolkits

1.3.7 Data Standardization Knowledgebase

지식 베이스(Knowledgebase)는 **데이터 표준화에 대한 실무자의 경험과 지식의 아카이브**다. 또한, **조직의 데이터 표준화 핵심 지적 자산으로 표준화 품질 향상과 생산성 제고의 선순환 체계**다.

1.3.7.1 구성

웹 사이트 기반의 지식 베이스와 파일매체 기록물(도서 형태)로 구성된다.

구성 내역	역할	기능	비고
Documents	데이터 표준화 활동의 성공과 시행착오에 대한 기록	데이터 표준화 활동의 성공과 시행착오에 대한 기록물	표준화 기록물
Website	Data Standardization Knowledgebase 시스템	Web Application 기반 Knowledgebase System	표준화 지식 베이스

그림 1.25. 구성: Data Standardization Knowledgebase

1.3.7.2 목적물 예시

지식 베이스는 다양한 채널 및 파일 포맷으로 제공된다. 산출물, 경험 자산이 디지털화되어 보관, 공유, 공개되는 형태다.

구분	형식	내용	비고
Documents	Word, PowerPoint, PDF 파일	데이터 표준화 산출물, 데이터 표준화에 대한 실무자의 경험과 지식의 콘텐트	경험/자산 아카이브
Website	Web Pages (Documents 포함)	데이터 표준화 방법론 리소스, 데이터 표준화 산출물, 데이터 표준화에 대한 실무자의 경험과 지식의 콘텐트	Knowledgebase

그림 1.26. 목적물 예시: Data Standardization Knowledgebase

II Foundation Layer

2 데이터 모델링 프로세스 23
2.1 아키텍처 23
2.2 프로세스 23
2.3 산출물 흐름도 26

CHAPTER 2

데이터 모델링 프로세스

2.1 아키텍처

2.1.1 3-Depth Breakdown Architecture

데이터 표준화 프로세스(Data Standardization Process)는 `Phase` 〉 `Activity` 〉 `Task`의 3개 레벨로 정의되며, 각 Task가 생성한 산출물(Deliverable)은 다음 Task의 입력물(Input)이 된다.

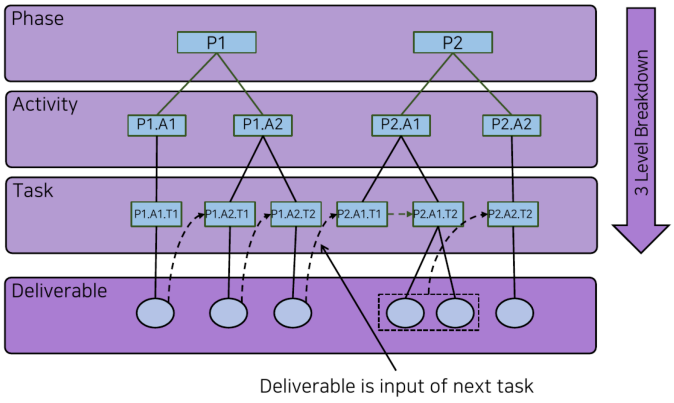

그림 2.1. 프로세스 3계층 구조와 산출물 연결

2.2 프로세스

데이터 모델링 프로세스는 7 Phase, 14 Activity, 18 Task로 구성된다.

2.2.1 프로세스 생명주기 다이어그램

데이터 모델링 표준화 프로세스는 산재한 데이터의 명칭, 정의, 형식, 규칙을 통일하여 일관성을 확보하는 과정으로, **현황분석 → 데이터 표준화 정의 → 데이터 표준화 확정 → 데이터 표준화 적용** 과정으로 진행된다.

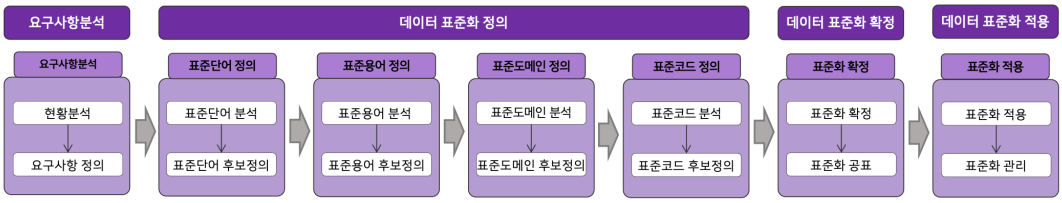

그림 2.2. 데이터 표준화 프로세스 생명주기

> **i 노트**
>
> `표준단어 정의`, `표준용어 정의`, `표준도메인 정의`, `표준코드 정의` 4개 단계는 상위 개념인 **데이터 표준화 정의**로 묶인다.

2.2.2 프로세스 단계 개요

프로세스는 크게 네 흐름으로 진행된다. 먼저 요구사항분석(Phase 1)에서 표준화의 원칙·범위·방법을 정하고 표준화 대상 데이터를 식별한다. 이어 데이터 표준화 정의 단계에서 표준단어(Phase 2)·표준용어(Phase 3)·표준도메인(Phase 4)·표준코드(Phase 5)를 차례로 분석하고 정의한다. 표준단어는 데이터를 구성하는 최소 의미 단위이고, 표준용어는 이를 조합한 이름이며, 표준도메인은 용어가 가질 데이터 형식을, 표준코드는 코드성 데이터의 값 체계를 규정한다. 정의된 후보 표준은 표준화 확정(Phase 6) 단계에서 이해관계자 합의를 거쳐 확정·공표되고, 마지막으로 표준화 적용(Phase 7) 단계에서 논리·물리 데이터 모델에 반영된 뒤 표준 준수 여부 점검과 변경관리로 이어진다.

각 Phase는 하나 이상의 Activity와 Task로 나뉘며, Task별 표준 기준·절차·체크리스트·예시는 3장 표준 및 절차 매뉴얼에서 상세히 다룬다.

2.2.3 프로세스 표

Phase	Activity	Task	Input	Deliverable
현황분석	현황분석	표준화 원칙 이해		
		표준화 범위/방법 정의		표준화 대상/범위/방법
	요구사항 정의	데이터 구성요소 식별	표준화 대상 데이터 정보	
		표준화 대상 데이터 정의		표준화 대상 데이터 형상
표준단어 정의	표준단어 분석	표준단어 분석	표준단어 사전	
	표준단어 후보정의	표준단어 후보정의		후보 표준단어
표준용어 정의	표준용어 분석	표준용어 분석	표준용어 사전	
	표준용어 후보정의	표준용어 후보정의		후보 표준용어
표준도메인 정의	표준도메인 분석	표준도메인 분석	표준도메인 사전	
	표준도메인 후보정의	표준도메인 후보정의		후보 표준도메인
표준코드 정의	표준코드 분석	표준코드 분석	표준코드	
	표준코드 후보정의	표준코드 후보정의		후보 표준코드
표준화 확정	표준화 확정	데이터 표준확정	후보 표준화 내역(단어, 용어, 도메인), 표준코드	확정된 표준화 내역(단어, 용어, 도메인), 표준코드
	표준화 공표	데이터 표준 공표	확정된 표준화 내역(단어, 용어, 도메인), 표준코드	표준화 메타 데이터(표준사전, 표준코드)
표준화 관리	표준화 적용	논리 데이터 모델링	확정된 표준화 내역(단어, 용어, 도메인)	논리데이터모델다이어그램(ERD), 엔터티정의서, 애트리뷰트정의서
		물리 데이터 모델링	논리데이터모델다이어그램(ERD), 엔터티정의서, 애트리뷰트정의서, 확정된 표준코드	물리데이터모델다이어그램(ERD), 데이터베이스정의서, 테이블정의서,컬럼정의서
	표준화 관리	표준 준수 여부 점검	논리데이터모델다이어그램(ERD)	비표준 내역 및 변경된 표준화 내역
		변경관리	비표준 내역 및 변경된 표준화 내역, 표준화 메타 데이터(표준사전, 표준코드), 논리데이터모델다이어그램(ERD), 엔터티정의서, 애트리뷰트정의서, 물리데이터모델다이어그램(ERD), 데이터베이스정의서, 테이블정의서, 컬럼정의서	표준화 메타 데이터(표준사전, 표준코드), 논리데이터모델다이어그램(ERD), 엔터티정의서, 애트리뷰트정의서, 물리데이터모델다이어그램(ERD), 데이터베이스정의서, 테이블정의서, 컬럼정의서

그림 2.3. 7 Phase · 14 Activity · 18 Task 구성표

2.3 산출물 흐름도

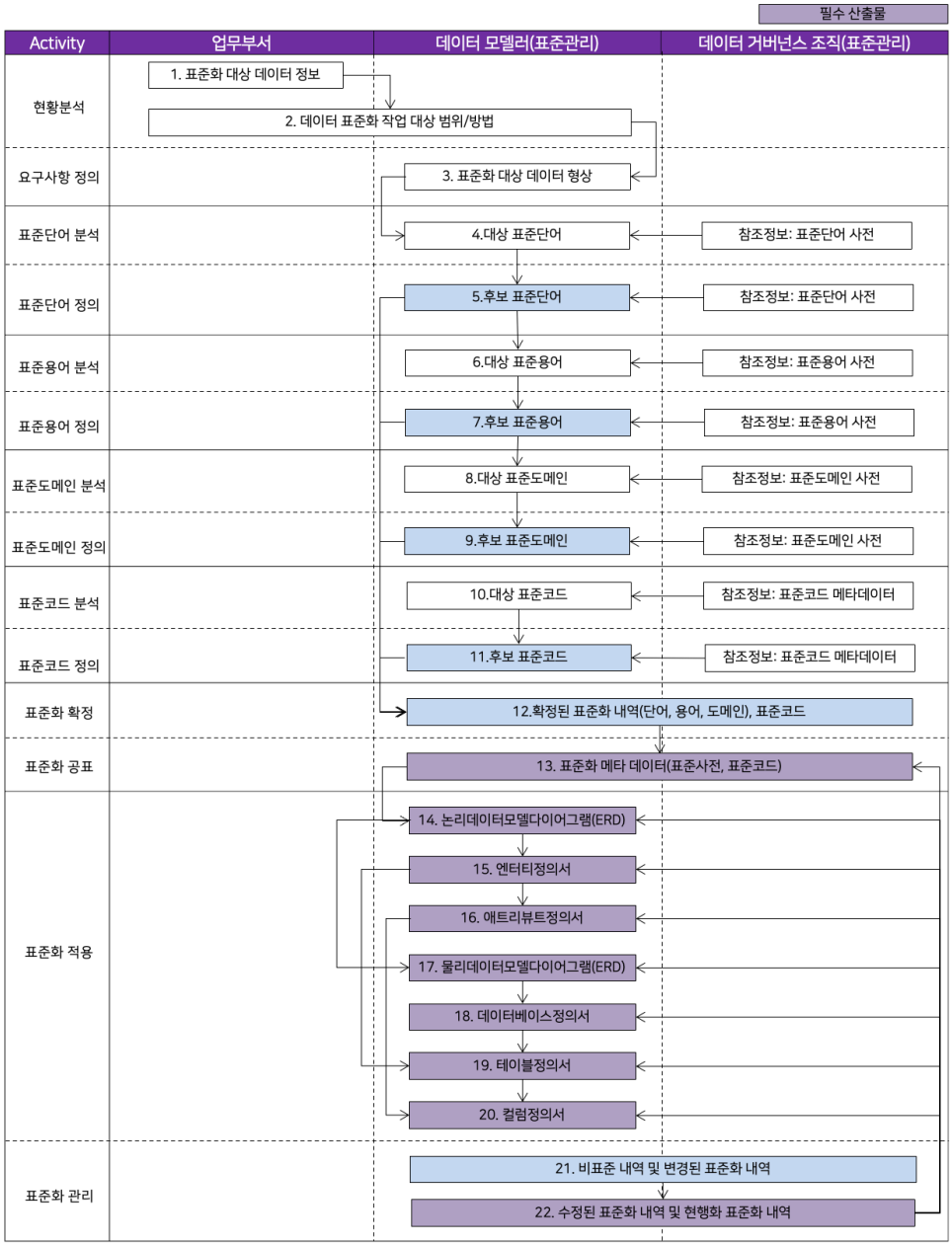

그림 2.4. 산출물 흐름도

III Guidance Layer

3 표준 및 절차 매뉴얼 29
3.1 개요 29
3.2 Ⓟ 요구사항분석 31
3.3 Ⓟ 표준단어 정의 36
3.4 Ⓟ 표준용어 정의 39
3.5 Ⓟ 표준도메인 정의 42
3.6 Ⓟ 표준코드 정의 45
3.7 Ⓟ 표준화 확정 48
3.8 Ⓟ 표준화 적용 51
3.9 부록 56

4 산출물 템플릿 59
4.1 개요 59
4.2 템플릿 체계 59
4.3 요구사항분석 산출물 템플릿 61
4.4 데이터 표준 정의 산출물 템플릿 62
4.5 표준화 확정·공표 산출물 템플릿 68
4.6 표준화 적용 산출물 템플릿 68
4.7 표준화 관리 산출물 템플릿 75
4.8 템플릿 운영 지침 76

CHAPTER 3

표준 및 절차 매뉴얼

3.1 개요

표준 및 절차 매뉴얼(Standards and Procedures Manual, **이하 SPM**)은 데이터 표준화 작업 수행 시 산출 결과의 균일한 품질 수준을 유지하고, 각종 공정 활동과 산출물의 품질 목표를 달성하기 위하여 적용하여야 할 **각종 절차, 표준을 정의**한 것이다. 이를 토대로 데이터 모델링 수행 시 대내적으로는 분석 표준을 정립하여 데이터 모델링 결과의 품질을 높이고 프로젝트 팀원의 기술력을 향상하며, 대외적으로는 분석 유관부서가 만족할 수 있는 고품질의 데이터 모델을 도출할 수 있도록 한다.

3.1.1 목적

본 매뉴얼은 데이터 표준화 프로세스(Data Standardization Process)를 기반으로, 조직이 프로젝트나 업무를 수행할 때 각 단계에서 준수해야 할 **표준 기준**과 **단계별 절차**를 상세히 정의하는 것을 목적으로 한다. 프로세스 참조 모델에 기반한 표준화 관리는 데이터 품질관리의 국제 표준 접근이기도 하다(ISO 2016년).

데이터 표준화 프로세스는 산재한 데이터의 명칭·정의·형식·규칙을 통일하여 일관성을 확보하고, 이를 논리·물리 데이터 모델에 반영하기까지의 전 과정을 다룬다. 본 매뉴얼은 이 과정을 실무자가 반복 가능한 방식으로 수행할 수 있도록 **각 Task별 기준·절차·체크리스트·예시를 제공**한다.

3.1.2 적용 범위

- 데이터 표준화 요구사항 분석 및 대상 정의
- 표준단어·표준용어·표준도메인·표준코드의 분석 및 정의
- 데이터 표준화 확정 및 공표
- 확정된 표준의 논리·물리 데이터 모델 적용 및 사후 관리

3.1.3 프로세스 아키텍처

데이터 표준화 프로세스는 Phase → Activity → Task의 3계층(3 Depth Breakdown)으로 정의한다. 각 Task는 하나 이상의 산출물(Deliverable)을 생성하며, 앞 Task의 산출물은 다음 Task의 입력물이

된다. 이는 프로젝트 관리 표준의 작업 분해(WBS)·산출물 중심 프로세스 정의 방식과 궤를 같이한다 (Project Management Institute 2021년).

표 3.1. 프로세스 3계층 구조

계층	설명	표기
Phase	프로세스의 최상위 단계	1, 2, 3 …
Activity	Phase를 구성하는 활동 단위	1.1, 1.2 …
Task	Activity를 구성하는 최소 수행 단위	1.1.1, 1.1.2 …

3.1.4 전체 프로세스 구성

본 매뉴얼이 다루는 전체 Phase 구성은 다음 그림 3.1 과 같다. 큰 흐름은 요구사항분석 → 데이터 표준화 정의 → 데이터 표준화 확정 → 데이터 표준화 적용의 4개 단계로 묶인다.

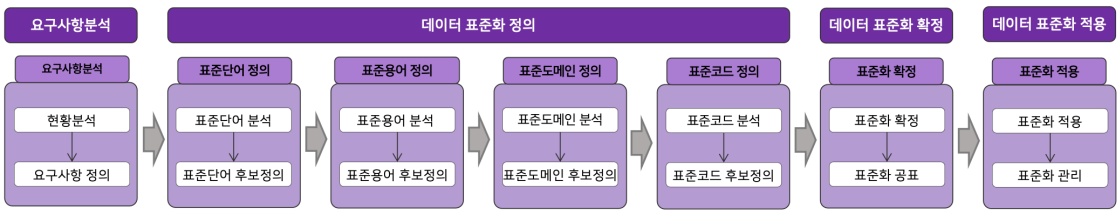

그림 3.1. 전체 Phase 구성과 4단계 흐름

표 3.2. 전체 Phase 구성

Phase	대분류	주요 산출물
1. 요구사항분석	요구사항분석	표준화 대상·범위·방법, 표준화 대상 데이터 형상
2. 표준단어 정의	데이터 표준화 정의	후보 표준단어
3. 표준용어 정의	데이터 표준화 정의	후보 표준용어
4. 표준도메인 정의	데이터 표준화 정의	후보 표준도메인
5. 표준코드 정의	데이터 표준화 정의	후보 표준코드
6. 표준화 확정	데이터 표준화 확정	확정 표준화 내역, 표준화 메타 데이터
7. 표준화 적용	데이터 표준화 적용	논리·물리 데이터 모델, 정의서

3.1.5 본 매뉴얼의 기술 방식

각 Task는 아래 항목 순서로 기술한다. 조직 표준으로 확정된 내용은 그대로 따르되, 원 프로세스 정의서에 명시되지 않아 일반적인 데이터 모델링 모범사례로 보강한 내용은 `[모범사례]`로 표시하여 구분한다.

- **개요**: Task의 목적과 위치
- **표준 기준**: 수행 시 반드시 준수해야 할 기준

- **단계별 절차**: 세부 수행 절차(필수/선택 구분)
- **입력물·산출물**: Task의 Input과 Deliverable
- **역할과 책임(R&R)**: 수행·검토·승인 주체
- **체크리스트**: 완료 판정 기준
- **예시**: 이해를 돕는 작성 예시

Phase → Activity → Task 레벨을 이해하기 위해서 이들 **헤더 이름 앞에 접두어(prefix)로서의 원문자를 붙여 표시**한다.

- Ⓟ : Phase
- Ⓐ : Activity
- Ⓣ : Task

3.2 Ⓟ 요구사항분석

요구사항분석 Phase는 데이터 표준화를 착수하기 전, **표준화의 원칙과 범위·방법을 정하고 표준화 대상 데이터를 식별하는 단계**이다. 이 단계의 산출물은 이후 모든 표준화 정의 활동의 기준선이 되므로, **대상과 범위를 명확히 확정하는 것이 핵심**이다.

요구사항분석은 그림 3.2 처럼 현황분석과 요구사항 정의의 순으로 진행한다.

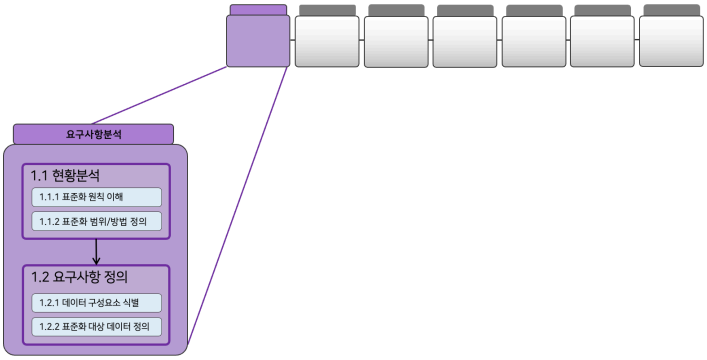

그림 3.2. 요구사항분석 프로세스 개요 (1.1 현황분석 → 1.2 요구사항 정의)

3.2.1 Ⓐ 현황분석

현황분석 Activity는 표준화 작업의 기반을 마련하는 활동으로, 표준화 원칙을 이해하고 표준화의 범위와 방법을 정의한다.

현황분석은 그림 3.3 처럼 표준화 원칙 이해를 기반으로 범위와 방법을 정의하는 순으로 진행한다.

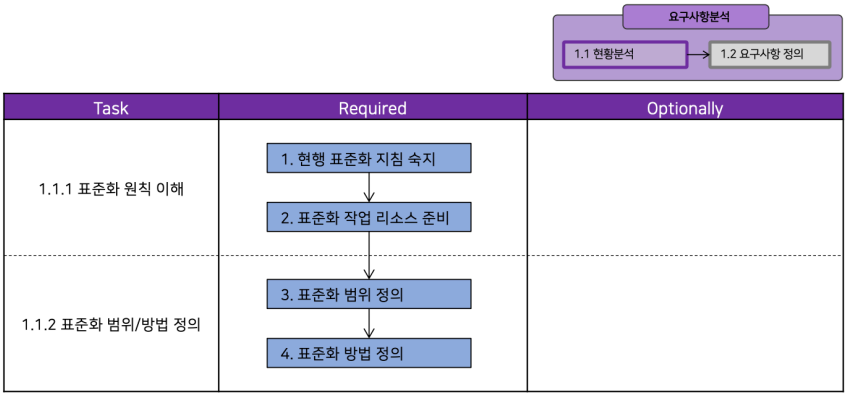

그림 3.3. 요구사항분석 : Detail Process — 현황분석

3.2.1.1 Ⓣ 표준화 원칙 이해

개요

표준화 작업에 착수하기 전, 조직의 현행 표준화 지침과 원칙을 숙지하고 작업에 필요한 리소스를 준비하는 Task이다. 표준화의 방향성을 결정하는 출발점이다.

표준 기준

- 표준화는 조직 전체에 일관되게 적용되어야 하며, 특정 시스템·부서에 국한되지 않는다.
- 표준화 원칙은 명문화된 지침(데이터 표준화 방법론, 거버넌스 규정 등)에 근거해야 한다.
- [모범사례] 표준화 3대 원칙을 작업 전반의 판단 기준으로 삼는다 (행정안전부 와/과 한국지능정보사회진흥원 2021년).
  - 유일성 (하나의 대상은 하나의 표준으로),
  - 일관성 (동일 개념은 동일 표기로),
  - 재사용성 (정의된 표준은 재사용 우선)

단계별 절차

1. **현행 표준화 지침 숙지** (필수)
   - 조직의 데이터 표준화 방법론, 거버넌스 규정, 기존 표준사전을 확인하고 적용 범위·우선순위를 파악한다.
2. **표준화 작업 리소스 준비** (필수)
   - 표준화 도구(모델링 툴, 표준사전 관리 도구), 참조 표준(공공표준용어 (행정안전부 2025년 b), 산업 표준), 작업 인력·일정을 확보한다.

입력물·산출물

표 3.3. 1.1.1 입력물·산출물

구분	항목
입력물	현행 표준화 지침, 거버넌스 규정
산출물	(다음 Task의 범위·방법 정의에 활용)

역할과 책임(R&R)

- 데이터 표준 담당자(수행)

- 데이터 아키텍트(검토)

체크리스트

- ☐ 현행 표준화 지침을 확인하고 적용 우선순위를 파악하였는가?
- ☐ 참조할 외부·내부 표준을 식별하였는가?
- ☐ 표준화 작업에 필요한 도구·인력·일정을 확보하였는가?

예시

- [모범사례] 착수 체크 항목
  - "기존 표준사전 버전 확인 → 참조 표준(행정표준용어 등) 선정 → 모델링 도구 표준 템플릿 준비" 순으로 리소스를 점검한다.

3.2.1.2 Ⓣ 표준화 범위·방법 정의

개요

표준화를 적용할 대상 범위와 수행 방법을 확정하는 Task이다. 범위가 모호하면 이후 모든 정의 활동의 산출물 품질이 저하되므로, 경계를 명확히 긋는 것이 중요하다.

표준 기준

- 표준화 범위는 대상 시스템·데이터 영역·제외 대상을 명시적으로 정의한다.
- 표준화 방법은 신규 정의와 기존 표준 재사용의 판단 기준을 포함한다.
- [모범사례] 범위는 "포함(In-scope) / 제외(Out-of-scope)"를 함께 기술하여 해석의 여지를 없앤다.

단계별 절차

3. **표준화 범위 정의** (필수)
   - 표준화 대상 시스템·업무 영역·데이터 집합을 확정하고, 제외 대상과 그 사유를 명시한다.
4. **표준화 방법 정의** (필수)
   - 표준 도출 방식(상향식/하향식), 기존 표준 재사용 원칙, 검토·승인 절차, 도구 사용 방법을 정의한다.

입력물·산출물

표 3.4. 1.1.2 입력물·산출물

구분	항목
입력물	현행 표준화 지침, 표준화 원칙
산출물	**표준화 대상/범위/방법**

역할과 책임(R&R)

- 데이터 아키텍트(수행)
- 데이터 거버넌스 책임자(승인)

체크리스트

- ☐ 표준화 대상 시스템·데이터 영역이 명시되었는가?
- ☐ 제외 대상과 사유가 기술되었는가?
- ☐ 신규 정의/재사용 판단 기준이 정의되었는가?
- ☐ 검토·승인 절차가 합의되었는가?

예시

- 범위 정의서 요약
  - 포함: 고객·주문·상품 도메인
  - 제외: 로그·임시 테이블
  - 방법: 기존 표준사전 우선 재사용, 부재 시 신규 정의

3.2.2 Ⓐ 요구사항 정의

요구사항 정의 Activity는 표준화 대상 데이터를 구체적으로 식별하고, 테이블·컬럼 수준의 표준화 대상 요소를 도출하는 활동이다.

요구사항 정의는 그림 3.4 처럼 데이터의 구성요소를 식별하고 표준화 대상 데이터를 정의하는 순으로 진행한다.

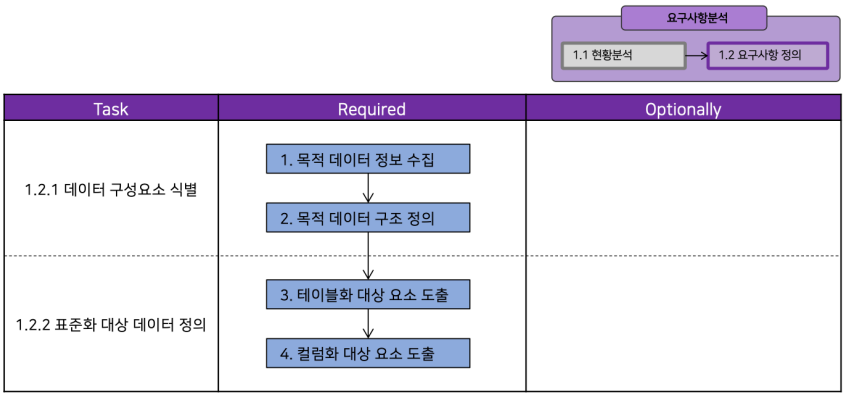

그림 3.4. 요구사항분석 : Detail Process — 요구사항 정의

3.2.2.1 Ⓣ 데이터 구성요소 식별

개요

표준화 대상이 되는 데이터의 정보를 수집하고 구조를 파악하는 Task이다. 목적 데이터가 어떤 구성요소로 이루어지는지 식별한다.

표준 기준

- 대상 데이터의 출처·정의·현행 구조를 근거 자료와 함께 수집한다.
- 데이터 구조는 개념적 수준(엔터티·주요 속성)에서 우선 파악한다.

단계별 절차

1. **목적 데이터 정보 수집** (필수)
   - 대상 데이터의 업무 정의, 현행 테이블·화면·문서 등 원천 정보를 수집한다.
2. **목적 데이터 구조 정의** (필수)
   - 수집된 정보를 바탕으로 데이터의 개념 구조(주요 개체와 관계)를 정의한다.

입력물·산출물

표 3.5. 1.2.1 입력물·산출물

구분	항목
입력물	**표준화 대상 데이터 정보**
산출물	(다음 Task의 대상 데이터 정의에 활용)

역할과 책임(R&R)
- 데이터 모델러(수행)
- 현업 담당자(정보 제공)

체크리스트
- ☐ 대상 데이터의 원천 정보를 수집하였는가?
- ☐ 주요 개체와 관계가 개념 수준에서 정의되었는가?

예시
- [모범사례] 원천 정보 수집표
  - “업무명 / 현행 테이블 / 화면·보고서 / 담당 현업” 항목으로 정리하면 누락을 줄일 수 있다.

3.2.2.2 Ⓣ 표준화 대상 데이터 정의

개요
식별된 데이터 구성요소로부터 표준화 대상이 되는 테이블화·컬럼화 요소를 도출하는 Task이다. 이후 단어·용어·도메인·코드 정의의 입력이 되는 대상 목록을 확정한다.

표준 기준
- 테이블화 대상은 업무적으로 독립적인 개체를 기준으로 도출한다.
- 컬럼화 대상은 원자적(더 이상 분해되지 않는) 속성 단위로 도출한다.
- [모범사례] 도출된 요소는 이후 표준단어·표준용어·표준도메인·표준코드 정의의 원천이 되므로, 각 요소의 한글명을 함께 기록한다.

단계별 절차
3. **테이블화 대상 요소 도출** (필수)
   - 개념 구조에서 테이블로 관리할 개체를 도출한다.
4. **컬럼화 대상 요소 도출** (필수)
   - 각 개체의 속성을 컬럼 단위로 도출한다.

입력물·산출물

표 3.6. 1.2.2 입력물·산출물

구분	항목
입력물	목적 데이터 구조
산출물	**표준화 대상 데이터 형상**

역할과 책임(R&R)
- 데이터 모델러(수행)

- 데이터 아키텍트(검토)

체크리스트

- ☐ 테이블화 대상 개체가 도출되었는가?
- ☐ 컬럼화 대상 속성이 원자 단위로 도출되었는가?
- ☐ 각 요소의 한글명이 기록되었는가?

예시

- 대상 데이터 형상 요약
  - 개체: 고객, 주문
  - 고객 속성: 고객번호, 고객명, 생년월일
  - 주문 속성: 주문번호, 주문일자, 주문금액

3.3 Ⓟ 표준단어 정의

표준단어 정의 Phase는 데이터 명칭을 구성하는 최소 의미 단위인 '단어'를 표준화하는 단계이다. 표준단어는 이후 표준용어(단어의 조합)의 기반이 되므로, 단어 수준의 일관성이 전체 표준화 품질을 좌우한다.

표준단어 정의는 그림 3.5 처럼 표준단어 분석과 표준단어 정의의 순으로 진행한다.

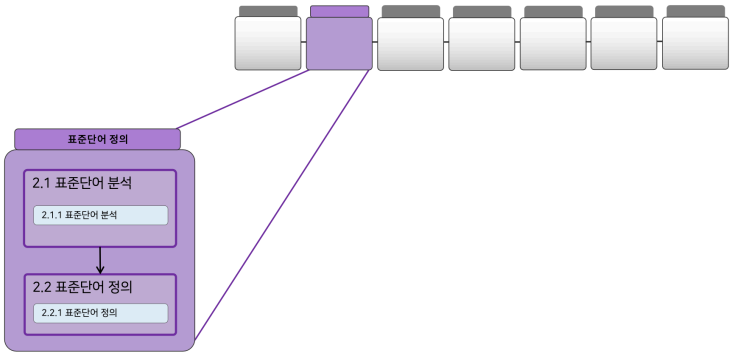

그림 3.5. 표준단어 정의 프로세스 개요 (2.1 표준단어 분석 → 2.2 표준단어 정의)

3.3.1 Ⓐ 표준단어 분석

표준단어 분석 Activity는 표준화 대상 데이터에서 단어를 추출하고 기존 표준단어사전과 대조하여 매칭 또는 신규 정의 후보를 도출하는 활동이다.

표준단어 분석은 그림 3.6 처럼 분석 대상 단어 정의 및 표준단어사전 검색을 통해서 단어 후보를 도출하는 순으로 진행한다.

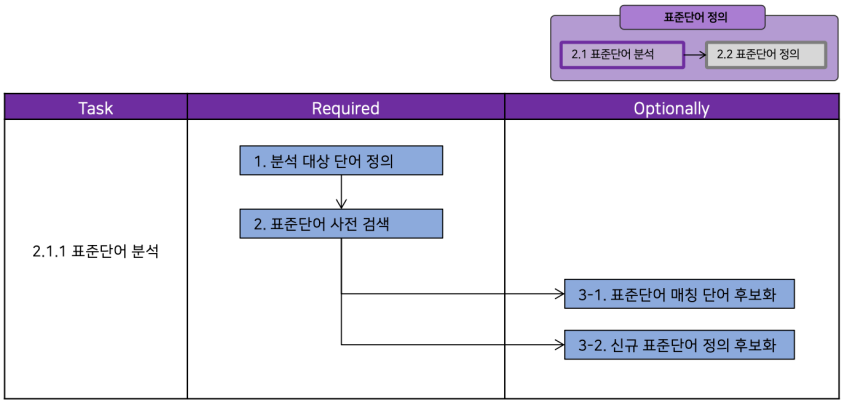

그림 3.6. 표준단어 정의 : Detail Process — 표준단어 분석

3.3.1.1 ⓣ 표준단어 분석

개요

분석 대상 단어를 정의하고 표준단어사전을 검색하여, 기존 표준과 매칭하거나 신규 표준단어로 정의할 후보를 도출하는 Task이다.

표준 기준

- 단어는 더 이상 분해되지 않는 최소 의미 단위로 추출한다 (한국데이터산업진흥원 2021년).
- 기존 표준단어사전에 존재하는 단어는 신규 정의하지 않고 재사용한다.
- [모범사례] 동일 의미의 서로 다른 표기(예: “고객”/“거래처”)는 하나의 대표 단어로 수렴하고 나머지는 이음동의어로 처리한다 (데이터가치연구소 2024년).

단계별 절차

1. **분석 대상 단어 정의** (필수)
   - 표준화 대상 데이터의 한글명을 최소 의미 단위로 분해하여 분석 대상 단어 목록을 작성한다.
2. **표준단어사전 검색** (필수)
   - 기존 표준단어사전에서 각 단어의 존재 여부를 검색한다.
3. **후보화** (선택)
   - 검색 결과에 따라 다음 중 하나로 후보화한다.
     - **3-1. 표준단어 매칭 단어 후보화** — 사전에 존재하면 해당 표준단어로 매칭한다.
     - **3-2. 신규 표준단어 정의 후보화** — 사전에 없으면 신규 표준단어 정의 후보로 등록한다.

입력물·산출물

표 3.7. 2.1.1 입력물·산출물

구분	항목
입력물	**표준단어사전**
산출물	매칭·신규 표준단어 정의 후보

역할과 책임(R&R)

- 데이터 표준 담당자(수행)
- 데이터 아키텍트(검토)

체크리스트

- ☐ 대상 데이터 한글명을 최소 단위로 분해하였는가?
- ☐ 각 단어의 표준단어사전 존재 여부를 검색하였는가?
- ☐ 매칭/신규 후보로 분류하였는가?

예시

- “주문상세내역”
  - 분해: 주문 / 상세 / 내역
  - 사전 검색: ‘주문’·‘내역’ 매칭, ‘상세’ 신규 후보

> ! 표준단어사전
>
> 표준단어사전 저장 매체가 파일이든 DBMS의 테이블이든, 웹사이트의 페이지이든 사전에 반드시 표준단어사전이 준비되어 있어야 한다. 만약 보유하고 있지 않다면 표준단어사전을 구축하라.

> 표준단어사전 대안
>
> 만약 표준단어사전을 보유하고 있지 않다면 **행정안전부의 공공데이터 공통표준단어**를 활용하여 구축하라 (행정안전부 2025년c).

3.3.2 Ⓐ 표준단어 정의

표준단어 정의 Activity는 후보화된 단어에 대해 단어명·이음동의어·금칙어를 확정하는 활동이다. (원 프로세스 정의서에서는 이 Activity를 ‘표준단어 후보정의’로 표기하기도 한다.)

표준단어 분석은 그림 3.7 처럼 단어명을 정의하고 이음동의어 및 금칙어를 정의하는 순으로 진행한다.

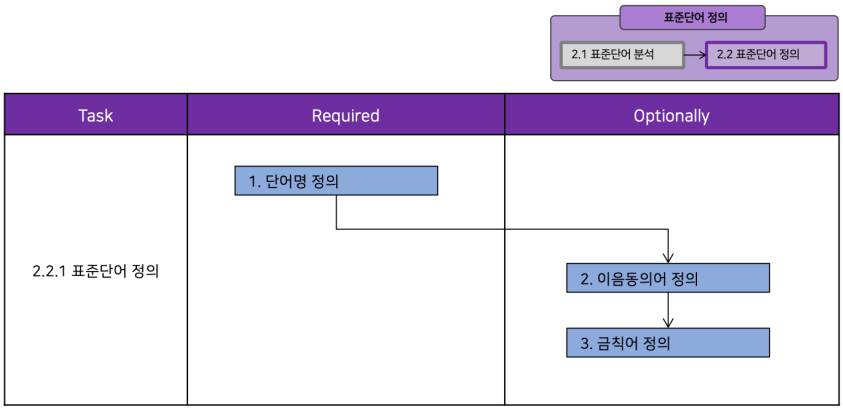

그림 3.7. 표준단어 정의 : Detail Process — 표준단어 정의

3.3.2.1 Ⓣ 표준단어 정의

개요

후보 단어의 표준 단어명을 확정하고, 이음동의어와 금칙어를 정의하여 표준단어를 완성하는 Task이다.

표준 기준

- 표준 단어명은 대표성·간결성·명확성을 갖추어야 한다.

- 이음동의어는 표준 단어명으로 대체되어야 하는 비표준 표기를 모두 포함한다.
- 금칙어는 사용을 금지할 단어(오해 소지·중복·비표준 약어 등)로 명시한다.
- [모범사례] 표준 단어에는 영문 약어(Abbreviation)를 함께 정의하여 물리명 생성 시 일관되게 사용한다.

단계별 절차

1. **단어명 정의** (필수)
   - 후보 단어의 표준 한글 단어명을 확정한다.
2. **이음동의어 정의** (선택)
   - 표준 단어명으로 수렴되어야 할 동의어·유사어를 등록한다.
3. **금칙어 정의** (선택)
   - 사용을 금지할 단어를 등록한다.

입력물·산출물

표 3.8. 2.2.1 입력물·산출물

구분	항목
입력물	매칭·신규 표준단어 정의 후보
산출물	**후보 표준단어**

역할과 책임(R&R)

- 데이터 표준 담당자(수행)
- 데이터 거버넌스 책임자(검토)

체크리스트

- ☐ 표준 단어명이 확정되었는가?
- ☐ 이음동의어가 표준 단어명으로 수렴되도록 정의되었는가?
- ☐ 금칙어가 식별되었는가?
- ☐ [모범사례] 영문 약어가 정의되었는가?

예시

- 표준단어 “고객”(약어 CUST)
- 이음동의어 “거래처·손님”
- 금칙어 “customer(영문 혼용 금지)”

3.4 Ⓟ 표준용어 정의

표준용어 정의 Phase는 표준단어를 조합하여 업무에서 실제 사용하는 ‘용어’(엔터티명·속성명 등)를 표준화하는 단계이다. 표준용어는 표준단어의 조합 규칙을 따른다.

표준용어 정의는 그림 3.8 처럼 표준용어 분석과 표준용어 정의의 순으로 진행한다.

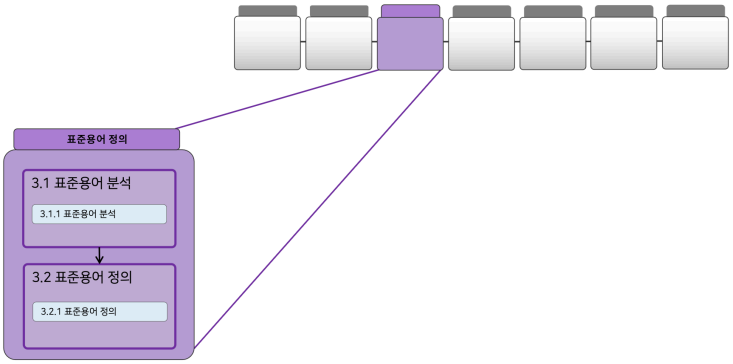

그림 3.8. 표준용어 정의 프로세스 개요 (3.1 표준용어 분석 → 3.2 표준용어 정의)

3.4.1 Ⓐ 표준용어 분석

표준용어 분석 Activity는 대상 용어를 정의하고 표준용어 사전과 대조하여 매칭 또는 신규 정의 후보를 도출하는 활동이다.

표준용어 분석은 그림 3.9 처럼 분석 대상 용어 정의 및 표준용어 사전 검색을 통해서 용어 후보를 도출하는 순으로 진행한다.

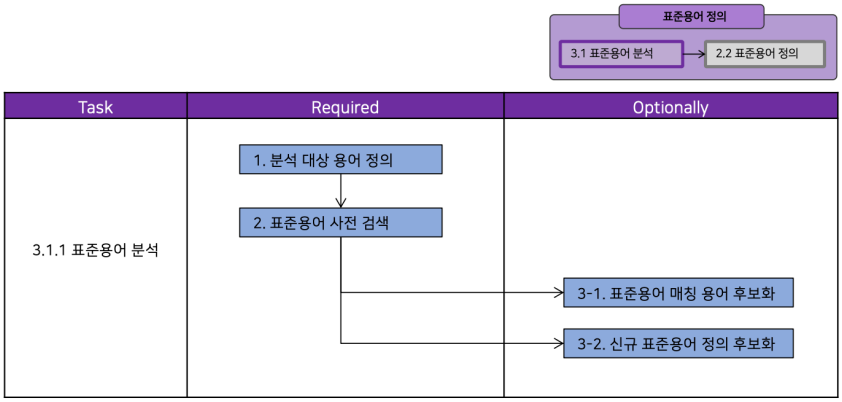

그림 3.9. 표준용어 정의 : Detail Process — 표준용어 분석

3.4.1.1 Ⓣ 표준용어 분석

개요

분석 대상 용어를 정의하고 표준용어 사전을 검색하여, 기존 표준과 매칭하거나 신규 표준용어로 정의할 후보를 도출하는 Task이다.

표준 기준

- 용어는 표준단어의 조합으로 구성하며, 비표준 단어를 포함하지 않는다.
- 기존 표준용어 사전에 존재하는 용어는 재사용한다.
- [모범사례] 용어의 구성 단어 순서는 “수식어 + 핵심명사” 규칙을 따른다(예: 고객 + 번호 → 고객번호).

단계별 절차

1. **분석 대상 용어 정의** (필수)
    - 표준화 대상의 엔터티명·속성명 등 용어 목록을 작성한다.

2. **표준용어 사전 검색** (필수)
    - 기존 표준용어 사전에서 각 용어의 존재 여부를 검색한다.
3. **후보화** (선택)
    - **3-1. 표준용어 매칭 용어 후보화**
        - 사전에 존재하면 해당 표준용어로 매칭한다.
    - **3-2. 신규 표준용어 정의 후보화**
        - 사전에 없으면 신규 표준용어 정의 후보로 등록한다.

입력물·산출물

표 3.9. 3.1.1 입력물·산출물

구분	항목
입력물	**표준용어 사전**
산출물	매칭·신규 표준용어 정의 후보

역할과 책임(R&R)
- 데이터 표준 담당자(수행)
- 데이터 아키텍트(검토)

체크리스트
- ☐ 대상 용어 목록을 작성하였는가?
- ☐ 용어가 표준단어 조합으로 구성되는지 확인하였는가?
- ☐ 매칭/신규 후보로 분류하였는가?

예시
- "고객생년월일" → 구성 단어: 고객·생년·월일 (모두 표준단어) → 사전 검색 후 신규 표준용어 후보로 등록.

3.4.2 Ⓐ 표준용어 정의

표준용어 정의 Activity는 후보 용어의 용어명과 용어 설명을 확정하는 활동이다. (원 프로세스 정의서에서는 '표준용어 후보정의'로 표기하기도 한다.)

표준용어 정의는 그림 3.10 처럼 용어명을 정의하고 용어 설명을 정의하는 순으로 진행한다.

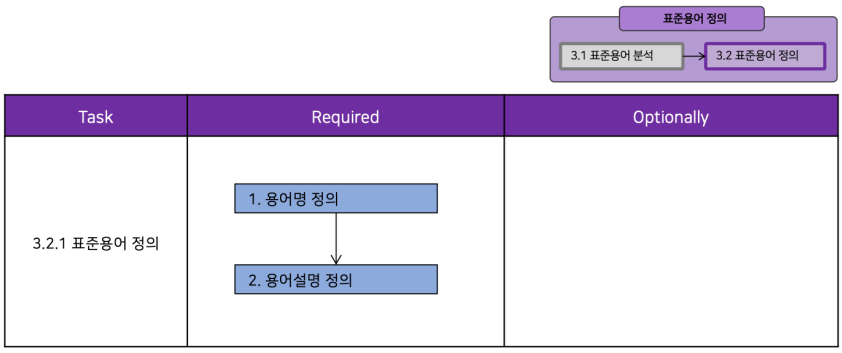

그림 3.10. 표준용어 정의 : Detail Process — 표준용어 정의

3.4.2.1 Ⓣ 표준용어 정의

개요
후보 용어의 표준 용어명을 확정하고 용어 설명(정의)을 작성하여 표준용어를 완성하는 Task이다.

표준 기준
- 표준 용어명은 표준단어의 조합 규칙과 순서를 준수한다.
- 용어 설명은 해당 용어의 업무적 의미를 명확하게 기술한다.
- [모범사례] 용어에는 영문명(표준단어 약어의 조합)과 대표 도메인을 함께 연결한다.

단계별 절차
1. **용어명 정의** (필수)
    - 표준단어를 조합하여 표준 용어명을 확정한다.
2. **용어설명 정의** (필수)
    - 용어의 업무적 의미를 설명으로 기술한다.

입력물·산출물

표 3.10. 3.2.1 입력물·산출물

구분	항목
입력물	매칭·신규 표준용어 정의 후보
산출물	**후보 표준용어**

역할과 책임(R&R)
- 데이터 표준 담당자(수행)
- 데이터 거버넌스 책임자(검토)

체크리스트
- ☐ 표준 용어명이 단어 조합 규칙을 따르는가?
- ☐ 용어 설명이 업무 의미를 명확히 기술하는가?
- ☐ [모범사례] 영문명·대표 도메인이 연결되었는가?

예시
- 표준용어 “고객번호”(영문명 CUSTOMER NUMBER, 물리명 CUST_NO)
- 설명 “고객을 유일하게 식별하는 번호”
- 도메인 “번호형”

3.5 Ⓟ 표준도메인 정의

표준도메인 정의 Phase는 데이터의 형식·유형·길이 등 물리적 표현 규칙인 ‘도메인’을 표준화하는 단계이다. 표준도메인은 속성(컬럼)의 데이터 타입 일관성을 보장한다.

표준도메인 정의는 그림 3.11 처럼 표준도메인 분석과 표준도메인 정의의 순으로 진행한다.

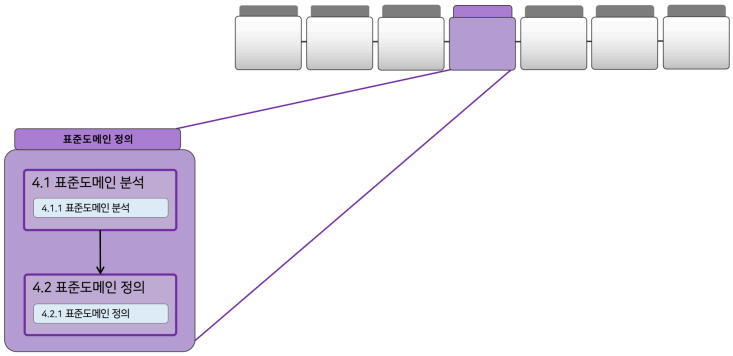

그림 3.11. 표준도메인 정의 프로세스 개요 (4.1 표준도메인 분석 → 4.2 표준도메인 정의)

3.5.1 Ⓐ 표준도메인 분석

표준도메인 분석 Activity는 대상 도메인을 정의하고 표준도메인 사전과 대조하여 매칭 또는 신규 정의 후보를 도출하는 활동이다.

표준도메인 분석은 그림 3.12 처럼 분석 대상 표준도메인 정의 및 표준도메인 사전 검색을 통해서 도메인 후보를 도출하는 순으로 진행한다.

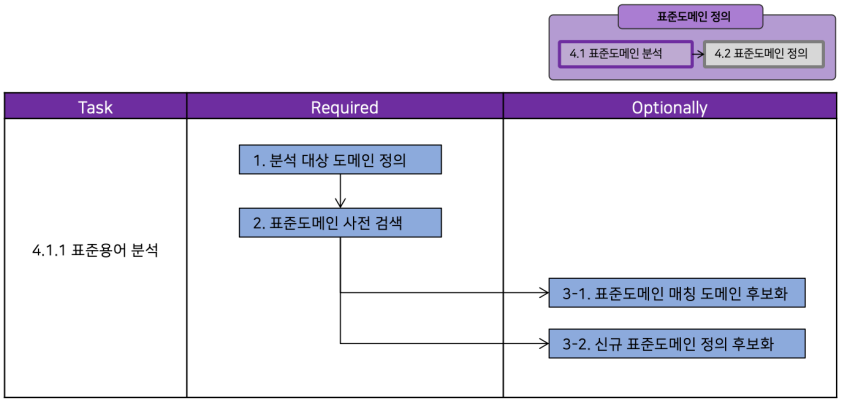

그림 3.12. 표준도메인 정의 : Detail Process — 표준도메인 분석

3.5.1.1 Ⓣ 표준도메인 분석

개요

분석 대상 도메인을 정의하고 표준도메인 사전을 검색하여, 기존 표준과 매칭하거나 신규 표준도메인으로 정의할 후보를 도출하는 Task이다.

표준 기준

- 도메인은 데이터 유형·길이·소수 자릿수·형식(마스크) 등을 포함하여 정의한다.
- 유사한 형식은 공통 도메인으로 통합하여 도메인 수를 최소화한다.
- [모범사례] 도메인은 “유형 그룹(문자·숫자·일자·여부·금액 등)” 단위로 분류 체계를 갖춘다 (한국데이터산업진흥원 2006년).

단계별 절차

1. **분석 대상 도메인 정의** (필수)
    - 속성에 필요한 도메인(형식) 목록을 작성한다.

2. **표준도메인 사전 검색** (필수)
    - 기존 표준도메인 사전에서 각 도메인의 존재 여부를 검색한다.
3. **후보화** (선택)
    - **3-1. 표준도메인 매칭 도메인 후보화**
        - 사전에 존재하면 해당 표준도메인으로 매칭한다.
    - **3-2. 신규 표준도메인 정의 후보화**
        - 사전에 없으면 신규 표준도메인 정의 후보로 등록한다.

입력물·산출물

표 3.11. 4.1.1 입력물·산출물

구분	항목
입력물	**표준도메인 사전**
산출물	매칭·신규 표준도메인 정의 후보

역할과 책임(R&R) 데이터 표준 담당자(수행), 데이터 아키텍트(검토).

체크리스트

- ☐ 대상 도메인(형식) 목록을 작성하였는가?
- ☐ 각 도메인의 표준도메인 사전 존재 여부를 검색하였는가?
- ☐ 유사 형식을 공통 도메인으로 통합 검토하였는가?

예시

- "생년월일" 속성 → 도메인 후보: 일자형(YYYYMMDD, 문자 8자리) → 사전 검색 후 기존 "일자형"에 매칭.

3.5.2 Ⓐ 표준도메인 정의

표준도메인 정의 Activity는 후보 도메인의 분류·데이터 구조·형식 정보를 확정하는 활동이다. (원 프로세스 정의서에서는 '표준도메인 후보정의'로 표기하기도 한다.)

표준도메인 정의는 그림 3.13 처럼 분석 대상 표준도메인 분류정보 정의, 도메인 데이터 구조 정보 정의, 도메인 형식 정보 정의 순으로 진행한다.

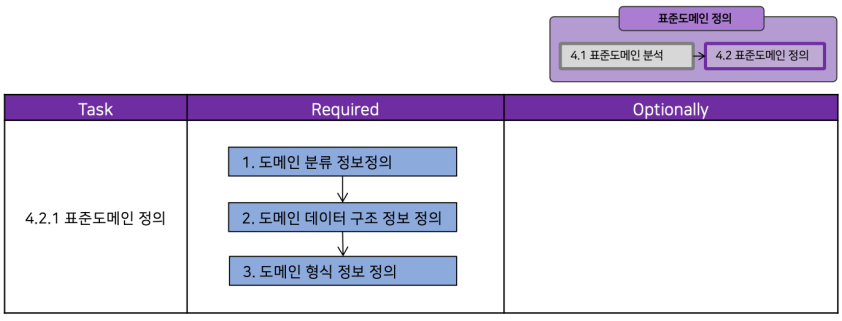

그림 3.13. 표준도메인 정의 : Detail Process — 표준도메인 정의

3.5.2.1 Ⓣ 표준도메인 정의

개요
후보 도메인의 분류 정보, 데이터 구조 정보, 형식 정보를 정의하여 표준도메인을 완성하는 Task이다.

표준 기준
- 도메인 분류는 조직의 도메인 분류 체계를 따른다.
- 데이터 구조 정보(유형·길이·소수 자릿수)와 형식 정보(표현 형식·허용 값 범위)를 모두 정의한다.
- [모범사례] 도메인에는 표준 데이터 타입(예: VARCHAR·NUMBER·DATE)과 물리 길이를 명시한다.

단계별 절차
1. **도메인 분류 정보정의** (필수)
   - 도메인의 분류(그룹·상위 도메인)를 정의한다.
2. **도메인 데이터 구조 정보 정의** (필수)
   - 데이터 유형·길이·소수 자릿수를 정의한다.
3. **도메인 형식 정보 정의** (필수)
   - 표현 형식(마스크)·허용 값 범위를 정의한다.

입력물·산출물

표 3.12. 4.2.1 입력물·산출물

구분	항목
입력물	매칭·신규 표준도메인 정의 후보
산출물	**후보 표준도메인**

역할과 책임(R&R)
- 데이터 표준 담당자(수행)
- 데이터 거버넌스 책임자(검토)

체크리스트
- ☐ 도메인 분류가 정의되었는가?
- ☐ 데이터 유형·길이·소수 자릿수가 정의되었는가?
- ☐ 형식·허용 값 범위가 정의되었는가?

예시
- 표준도메인 “금액형”
  - 분류: 숫자
  - 유형: NUMBER(18,2)
  - 형식: 소수 2자리
  - 허용 범위: 0 이상

3.6 Ⓟ 표준코드 정의

표준코드 정의 Phase는 한정된 값의 집합으로 관리해야 하는 데이터(코드)를 표준화하는 단계이다. 표준코드는 코드값과 그 의미를 일관되게 관리한다.

표준코드 정의는 그림 3.14 처럼 표준코드 분석과 표준코드 정의의 순으로 진행한다.

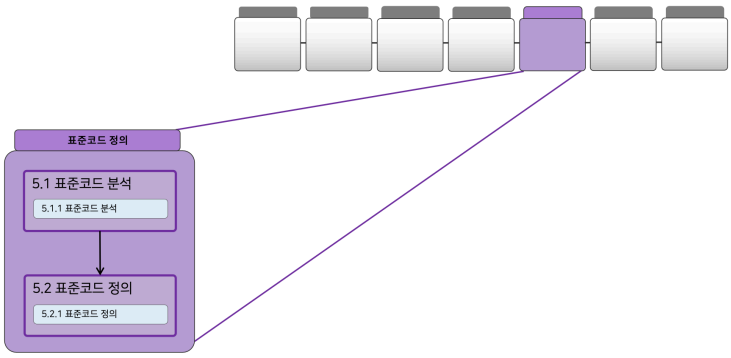

그림 3.14. 표준코드 정의 프로세스 개요 (5.1 표준코드 분석 → 5.2 표준코드 정의)

3.6.1 Ⓐ 표준코드 분석

표준코드 분석 Activity는 코드 관리 대상을 판단하고 표준코드를 검색하여 매칭 또는 신규 정의 후보를 도출하는 활동이다.

표준코드 분석은 그림 3.15 처럼 분석 대상 용어를 선정한 후 코드 관리 대상 여부를 판단하며, 이어서 표준코드 검색을 통해 후보 코드를 도출하는 순으로 진행한다.

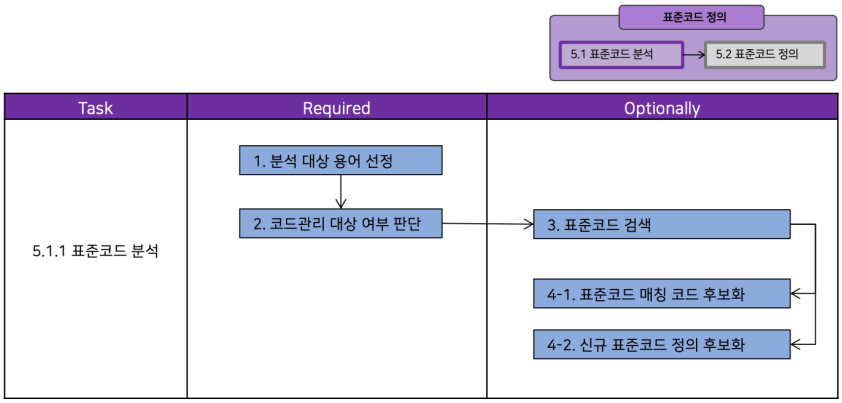

그림 3.15. 표준코드 정의 : Detail Process — 표준코드 분석

3.6.1.1 Ⓣ 표준코드 분석

개요

분석 대상 용어 중 코드 관리가 필요한 대상을 판단하고, 표준코드를 검색하여 매칭 또는 신규 정의 후보를 도출하는 Task이다.

표준 기준

- 값의 종류가 한정적이고 반복 사용되는 속성은 코드 관리 대상으로 판단한다.
- 기존 표준코드에 존재하는 코드는 재사용한다.
- [모범사례] 코드 관리 대상 판단 기준(값의 유한성·재사용성·통제 필요성)을 사전에 명문화한다.

단계별 절차

1. **분석 대상 용어 선정** (필수)
   - 코드화 가능성이 있는 용어(속성)를 선정한다.
2. **코드관리 대상 여부 판단** (필수)

    - 값의 유한성·통제 필요성을 근거로 코드 관리 대상 여부를 판단한다.
3. **표준코드 검색** (선택)
    - 코드 관리 대상에 대해 기존 표준코드를 검색한다.
4. **후보화** (선택)
    - **4-1. 표준코드 매칭 코드 후보화**
        - 표준코드에 존재하면 해당 코드로 매칭한다.
    - **4-2. 신규 표준코드 정의 후보화**
        - 존재하지 않으면 신규 표준코드 정의 후보로 등록한다.

입력물·산출물

표 3.13. 5.1.1 입력물·산출물

구분	항목
입력물	**표준코드**
산출물	매칭·신규 표준코드 정의 후보

역할과 책임(R&R)

- 데이터 표준 담당자(수행)
- 현업 담당자(값 의미 확인)

체크리스트

- ☐ 코드화 후보 용어를 선정하였는가?
- ☐ 코드 관리 대상 여부를 기준에 따라 판단하였는가?
- ☐ 기존 표준코드 존재 여부를 검색하였는가?

예시

- "주문상태" 속성 → 값이 (접수/배송중/완료 등)으로 유한 → 코드 관리 대상 → 표준코드 검색 후 신규 후보로 등록.

3.6.2 Ⓐ 표준코드 정의

표준코드 정의 Activity는 후보 코드의 속성 정보와 코드값을 확정하는 활동이다. (원 프로세스 정의서에서는 '표준코드 후보정의'로 표기하기도 한다.)

표준코드 정의는 그림 3.16 처럼 코드 속성 정보를 정의한 후, 코드값 의미 목록 도출 후 코드값 매핑 순으로 진행한다.

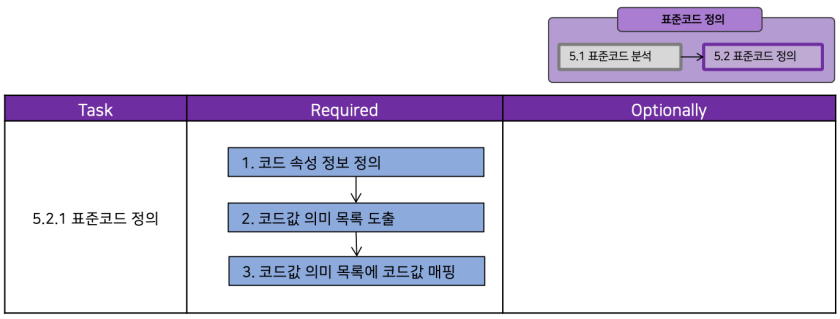

그림 3.16. 표준코드 정의 : Detail Process — 표준코드 정의

3.6.2.1 Ⓣ 표준코드 정의

개요
후보 코드의 속성 정보를 정의하고, 코드값 의미 목록을 도출하여 각 코드값을 매핑함으로써 표준코드를 완성하는 Task이다.

표준 기준
- 코드 속성 정보(코드명·코드 유형·자릿수 등)를 정의한다.
- 코드값은 의미(설명)와 1:1로 매핑되며, 중복·누락이 없어야 한다.
- [모범사례] 코드값은 확장 가능성을 고려해 여유 있는 체계(예: 2자리 이상)로 설계한다.

단계별 절차
1. **코드 속성 정보 정의** (필수)
   - 코드명·코드 유형·자릿수·기본값 등 코드의 속성을 정의한다.
2. **코드값 의미 목록 도출** (필수)
   - 코드가 표현할 의미(값의 종류) 목록을 도출한다.
3. **코드값 의미 목록에 코드값 매핑** (필수)
   - 각 의미에 실제 코드값을 부여한다.

입력물·산출물

표 3.14. 5.2.1 입력물·산출물

구분	항목
입력물	매칭·신규 표준코드 정의 후보
산출물	**후보 표준코드**

역할과 책임(R&R)
- 데이터 표준 담당자(수행)
- 현업 담당자(값 의미 확인)

체크리스트
- ☐ 코드 속성 정보가 정의되었는가?
- ☐ 코드값 의미 목록이 도출되었는가?
- ☐ 각 의미에 코드값이 중복·누락 없이 매핑되었는가?

예시
- 표준코드 “주문상태코드”(2자리)
  - 01: 접수,
  - 02: 배송중,
  - 03: 완료,
  - 09: 취소

3.7 Ⓟ 표준화 확정

표준화 확정 Phase는 정의된 표준화 후보(단어·용어·도메인·코드)를 검토·합의하여 조직의 공식 표준으로 확정하고, 이를 공표하여 실무에 적용 가능한 상태로 만드는 단계이다.

표준화 확정은 그림 3.17 처럼 표준화 확정과 표준화 공표의 순으로 진행한다.

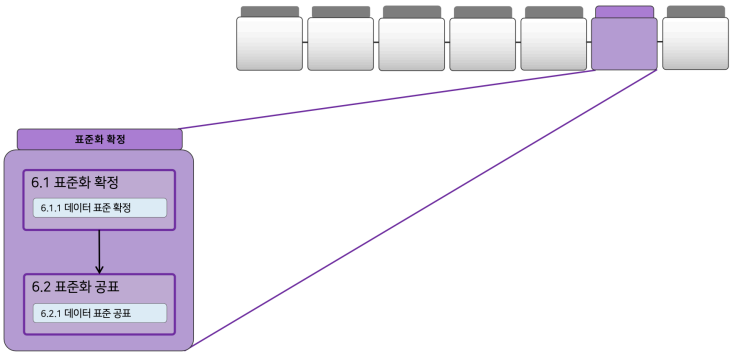

그림 3.17. 표준화 확정 프로세스 개요 (6.1 표준화 확정 → 6.2 표준화 공표)

3.7.1 Ⓐ 표준화 확정

표준화 확정 Activity는 후보 표준화 내역을 검토하고 이해관계자 합의를 거쳐 표준을 확정하는 활동이다.

표준코드 확정은 그림 3.18 처럼 정의된 후보 표준화 검토 후, 이해 관계자 합의를 통해서 표준화를 보완하는 순으로 진행한다.

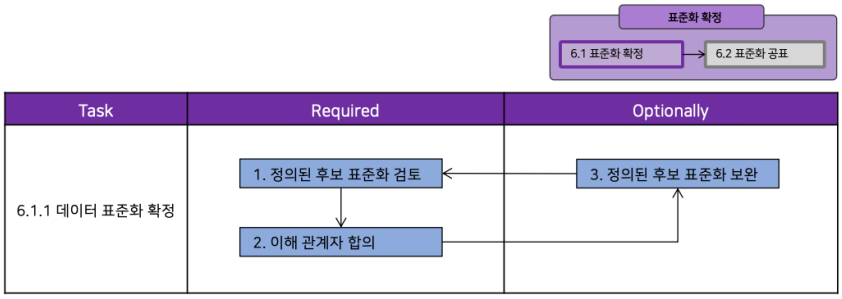

그림 3.18. 표준화 확정 : Detail Process — 표준화 확정

3.7.1.1 Ⓣ 데이터 표준화 확정

개요

정의된 후보 표준화 내역을 검토하고, 필요 시 보완하며, 이해관계자 합의를 통해 조직의 공식 표준으로 확정하는 Task이다.

표준 기준

- 확정 대상은 표준단어·표준용어·표준도메인·표준코드 전체를 포함한다.
- 확정은 관련 이해관계자의 합의를 전제로 한다.
- 검토 과정에서 발견된 미비점은 보완 후 재검토한다.
- [모범사례] 확정 이력(검토자·합의일·주요 변경)을 기록하여 추적성을 확보한다.

단계별 절차

1. **정의된 후보 표준화 검토** (필수)
   - 후보 표준화 내역의 완전성·일관성을 검토한다.
2. **이해 관계자 합의** (필수)
   - 관련 부서·담당자와 표준 내용을 합의한다.

3. **정의된 후보 표준화 보완** (선택, 반복)
    - 검토·합의 과정에서 미비점이 있으면 보완하고 1단계로 되돌아가 재검토한다.

입력물·산출물

표 3.15. 6.1.1 입력물·산출물

구분	항목
입력물	**후보 표준화 내역(단어·용어·도메인), 표준코드**
산출물	**확정된 표준화 내역(단어·용어·도메인), 표준코드**

역할과 책임(R&R)
- 데이터 거버넌스 책임자(승인)
- 데이터 표준 담당자(수행), 관련 현업·개발 담당자(합의)

체크리스트
- ☐ 후보 표준화 내역의 완전성·일관성을 검토하였는가?
- ☐ 이해관계자 합의가 이루어졌는가?
- ☐ 미비점 보완 후 재검토하였는가?
- ☐ [모범사례] 확정 이력을 기록하였는가?

예시
- 검토 회의에서 "고객유형코드" 값 누락 발견 → 보완(09: 기타 추가) → 재검토 후 합의·확정.

3.7.2 Ⓐ 표준화 공표

표준화 공표 Activity는 확정된 표준을 표준사전·표준코드 저장소에 등록하여 조직 전체가 참조·사용할 수 있도록 공표하는 활동이다.

표준코드 공표는 그림 3.19 처럼 표준 사전을 등록하고 표준코드 등록 순으로 진행한다.

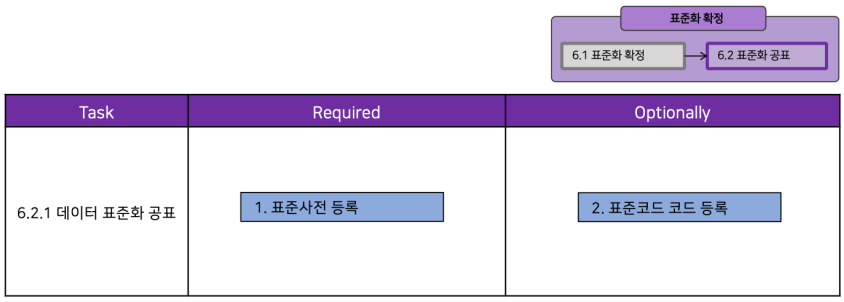

그림 3.19. 표준화 확정 : Detail Process — 표준화 공표

3.7.2.1 Ⓣ 데이터 표준화 공표

개요

확정된 표준을 표준사전에 등록하고, 코드는 표준코드 저장소에 등록하여 공식적으로 공표하는 Task 이다.

표준 기준

- 공표된 표준은 조직 전체의 단일 참조원(Single Source of Truth)이 된다 (DAMA International 2017년).
- 등록 후에는 정해진 변경관리 절차를 통해서만 수정한다.
- [모범사례] 공표 시 적용 시점·적용 대상 시스템을 함께 공지한다.

단계별 절차

1. **표준사전 등록** (필수)
    - 확정된 표준단어·표준용어·표준도메인을 표준사전에 등록한다.
2. **표준코드 코드 등록** (선택)
    - 확정된 표준코드를 표준코드 저장소에 등록한다.

입력물·산출물

표 3.16. 6.2.1 입력물·산출물

구분	항목
입력물	**확정된 표준화 내역(단어·용어·도메인), 표준코드**
산출물	**표준화 메타 데이터(표준사전, 표준코드)**

역할과 책임(R&R)

- 데이터 표준 담당자(수행)
- 데이터 거버넌스 책임자(공표 승인)

체크리스트

- ☐ 표준사전에 단어·용어·도메인을 등록하였는가?
- ☐ 표준코드를 저장소에 등록하였는가?
- ☐ [모범사례] 적용 시점·대상을 공지하였는가?

예시

- 표준사전 v2.0 공표
    - 적용 시점 2026-08-01
    - 대상: 신규 개발 전 시스템

3.8 Ⓟ 표준화 적용

표준화 적용 Phase는 공표된 표준을 논리·물리 데이터 모델에 실제로 반영하고, 적용 이후 표준 준수 여부를 점검하며 변경을 관리하는 단계이다. 표준화의 최종 산출물인 데이터 모델이 이 단계에서 완성된다.

표준화 적용은 그림 3.20 처럼 표준화 적용과 표준화 관리의 순으로 진행한다.

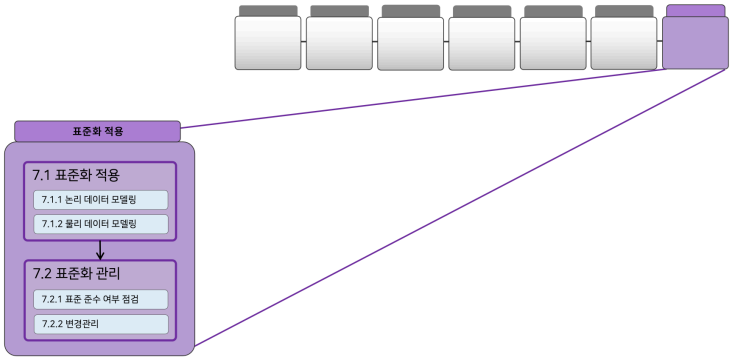

그림 3.20. 표준화 적용 프로세스 개요 (7.1 표준화 적용 → 7.2 표준화 관리)

3.8.1 Ⓐ 표준화 적용

표준화 적용 Activity는 확정된 표준을 사용하여 논리 데이터 모델과 물리 데이터 모델을 작성하는 활동이다.

표준코드 적용은 그림 3.21 처럼 논리 데이터 모델링 과정을 수행하고, 이를 기반으로 한 물리 모델링 과정 순으로 진행한다.

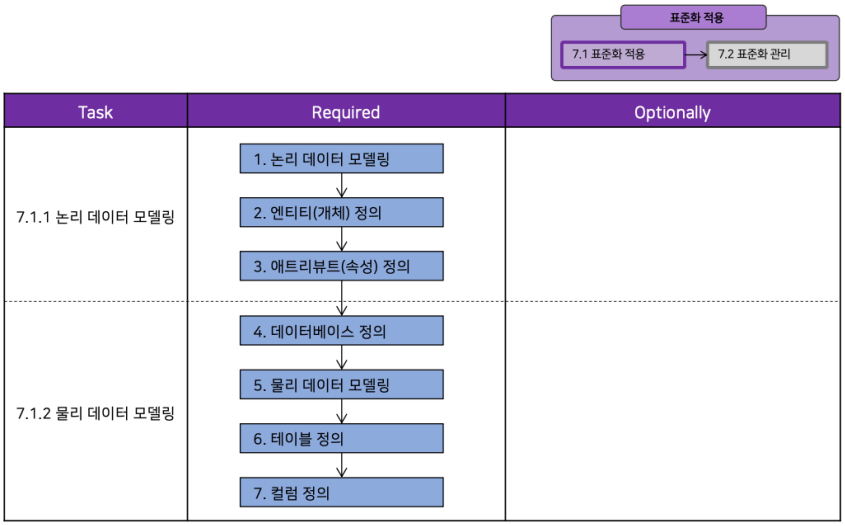

그림 3.21. 표준화 적용 : Detail Process — 표준화 적용

3.8.1.1 Ⓣ 논리 데이터 모델링

개요

확정된 표준(단어·용어·도메인)을 사용하여 업무 관점의 논리 데이터 모델을 작성하는 Task이다. 모델의 관리 구획인 주제영역을 정하고, 엔터티와 속성을 표준 용어·도메인으로 정의한다.

표준 기준

- 주제영역명·엔터티명·속성명은 반드시 표준용어를 사용한다.
- 속성의 형식은 표준도메인을 적용한다.
- [모범사례] 모델은 주제영역으로 구획하고, 하나의 엔터티는 하나의 주제영역에만 배정한다.

- [모범사례] 정규화(최소 3차 정규형 (Codd 1970년))를 원칙으로 하며, 식별자·관계를 명확히 정의한다.

단계별 절차

1. **논리 데이터 모델링** (필수)
   - 표준을 적용하여 논리 모델(엔터티-관계) 초안을 작성한다.
2. **주제영역 정의** (필수)
   - 모델의 관리 구획인 주제영역을 표준용어로 정의하고 순번을 부여한다.
3. **엔터티(개체) 정의** (필수)
   - 표준용어로 엔터티를 정의하고 식별자를 지정한다.
4. **속성(애트리뷰트) 정의** (필수)
   - 표준용어·표준도메인으로 각 속성을 정의한다.
5. **주제영역구성 정의** (필수)
   - 정의한 엔터티를 주제영역에 배정하고 구성 순서를 부여한다.

입력물·산출물

표 3.17. 7.1.1 입력물·산출물

구분	항목
입력물	**확정된 표준화 내역(단어·용어·도메인)**
산출물	**논리데이터모델 다이어그램(ERD), 주제영역 정의서, 주제영역구성 정의서, 엔터티 정의서, 속성 정의서**

역할과 책임(R&R)

- 데이터 모델러(수행)
- 데이터 아키텍트(검토)

체크리스트

- ☐ 주제영역명·엔터티명·속성명에 표준용어를 사용하였는가?
- ☐ 속성에 표준도메인을 적용하였는가?
- ☐ 식별자·관계가 정의되었는가?
- ☐ 모든 엔터티가 하나의 주제영역에 빠짐없이 배정되었는가?
- ☐ [모범사례] 정규화 원칙을 준수하였는가?

예시

- 주제영역 “영업”(순번: 1) — 구성: 고객, 주문
- 엔터티 “고객”(식별자: 고객번호)
  - 속성: 고객명(문자형), 생년월일(일자형), 고객유형코드(주문상태코드 참조)

3.8.1.2 Ⓣ 물리 데이터 모델링

개요

논리 데이터 모델을 대상 DBMS에 맞게 변환하여 물리 데이터 모델을 작성하는 Task이다. 표준단어 약어를 사용해 물리명을 생성하고 테이블·컬럼을 정의한다.

표준 기준

- 테이블명·컬럼명은 표준단어의 영문 약어 조합으로 생성한다.
- 컬럼의 데이터 타입·길이는 표준도메인의 물리 정의를 따른다.

- `[모범사례]` 물리명은 명명규칙(대문자 스네이크 케이스, 접두어 등)을 준수한다.

단계별 절차

1. **데이터베이스 정의** (필수)
   - 대상 DBMS·스키마·저장 특성을 정의한다.
2. **물리 데이터 모델링** (필수)
   - 논리 모델을 물리 모델로 변환한다.
3. **테이블 정의** (필수)
   - 엔터티를 테이블로 변환하고 물리명을 부여한다.
4. **컬럼 정의** (필수)
   - 속성을 컬럼으로 변환하고 표준도메인의 물리 타입을 적용한다.

입력물·산출물

표 3.18. 7.1.2 입력물·산출물

구분	항목
입력물	**논리데이터모델 다이어그램(ERD), 주제영역 정의서, 주제영역구성 정의서, 엔터티 정의서, 속성 정의서, 확정된 표준코드**
산출물	**물리데이터모델 다이어그램(ERD), 데이터베이스 정의서, 테이블 정의서, 컬럼 정의서**

역할과 책임(R&R)

- 데이터 모델러(수행)
- DBA(검토)

체크리스트

- ☐ 대상 DBMS·스키마가 정의되었는가?
- ☐ 테이블명·컬럼명이 표준 약어 조합으로 생성되었는가?
- ☐ 컬럼 타입·길이가 표준도메인 물리 정의를 따르는가?
- ☐ `[모범사례]` 물리명이 명명규칙을 준수하는가?

예시

- 엔터티 “고객” → 테이블 `TB_CUST_BASE_INFO`, 속성 “고객번호” → 컬럼 `CUST_NO VARCHAR(20)`.

3.8.2 Ⓐ 표준화 관리

표준화 관리 Activity는 적용된 표준의 준수 여부를 점검하고, 표준 및 모델의 변경을 관리하는 활동이다.

표준코드 관리는 그림 3.22 처럼 표준 준수 여부 점검 및 변경관리 과정 순으로 진행한다.

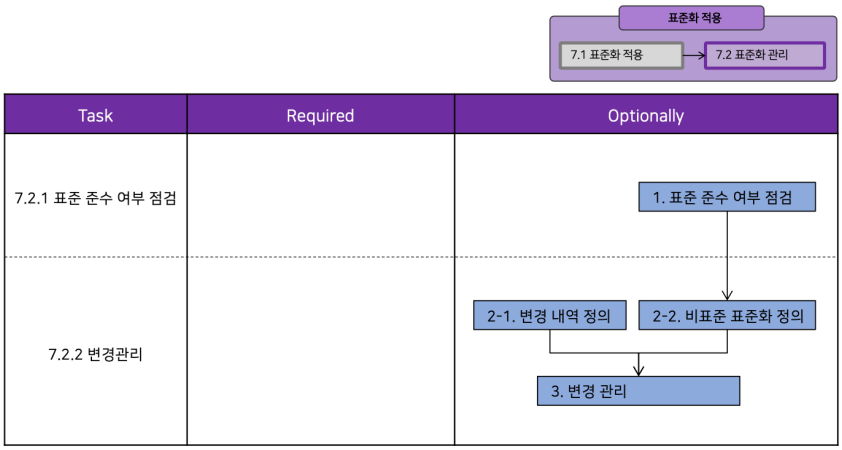

그림 3.22. 표준화 적용 : Detail Process — 표준화 관리

3.8.2.1 Ⓣ 표준 준수 여부 점검

개요

작성된 데이터 모델이 공표된 표준을 준수하는지 점검하여 비표준 내역을 식별하는 Task이다.

표준 기준

- 점검은 논리·물리 모델의 명칭·도메인·코드 사용 전반을 대상으로 한다.
- 비표준 내역은 원인과 함께 기록한다.
- `[모범사례]` 점검은 자동화 도구(표준 준수 검증 규칙)를 활용해 주기적으로 수행한다.

단계별 절차

1. **표준 준수 여부 점검** (선택)
    - 모델의 명칭·도메인·코드가 표준을 준수하는지 점검하고 비표준 내역을 도출한다.

입력물·산출물

표 3.19. 7.2.1 입력물·산출물

구분	항목
입력물	**논리데이터모델 다이어그램(ERD)**
산출물	**비표준 내역 및 변경된 표준화 내역**

역할과 책임(R&R)

- 데이터 아키텍트(수행)
- 데이터 거버넌스 책임자(검토)

체크리스트

- ☐ 명칭·도메인·코드 준수 여부를 점검하였는가?
- ☐ 비표준 내역을 원인과 함께 기록하였는가?

예시

- 점검 결과 컬럼 `CUSTOMER_NUMBER`(비표준) 발견 → 표준 약어 미적용으로 기록 → `CUST_NO`로 변경 대상 등록.

3.8.2.2 Ⓣ 변경관리

개요
점검 결과 도출된 변경 내역과 비표준 항목의 표준화를 정의하고, 표준·모델·산출물 전반에 변경을 반영·관리하는 Task이다.

표준 기준
- 모든 변경은 변경 내역으로 기록하고 영향 범위를 분석한다.
- 비표준 항목은 표준화하거나 예외로 승인·관리한다.
- 변경은 표준사전·표준코드·모델·정의서에 일관되게 반영한다.
- [모범사례] 변경관리 대장을 운영하여 변경의 요청-검토-승인-반영 이력을 관리한다.

단계별 절차
1. **변경 내역 정의** (선택)
   - 점검에서 도출된 변경 내역을 정의한다.
2. **비표준 표준화 정의** (선택)
   - 비표준 항목을 표준화(또는 예외 승인)한다.
3. **변경 관리** (선택)
   - 변경을 표준사전·표준코드·논리/물리 모델·각 정의서에 반영하고 이력을 관리한다.

입력물·산출물

표 3.20. 7.2.2 입력물·산출물

구분	항목
입력물	비표준 내역 및 변경된 표준화 내역, 표준화 메타 데이터, ERD, 엔터티·속성·데이터베이스·테이블·컬럼 정의서
산출물	갱신된 표준화 메타 데이터, ERD, 엔터티·속성·데이터베이스·테이블·컬럼 정의서

역할과 책임(R&R)
- 데이터 표준 담당자(수행)
- 데이터 거버넌스 책임자(변경 승인)

체크리스트
- ☐ 변경 내역을 정의하고 영향 범위를 분석하였는가?
- ☐ 비표준 항목을 표준화 또는 예외 승인하였는가?
- ☐ 변경을 표준사전·코드·모델·정의서에 일관되게 반영하였는가?
- ☐ [모범사례] 변경관리 대장에 이력을 기록하였는가?

예시
- 변경 요청 “고객 등급 코드 신설” → 영향 분석(관련 테이블 3개) → 표준코드 등록 → 모델·정의서 반영 → 대장 기록.

3.9 부록

3.9.1 부록 A: 전체 프로세스 요약표

아래 표는 전체 Phase → Activity → Task의 입력물과 산출물을 요약한 것이다. 원 프로세스 정의서의 Process Table을 기준으로 정리하였다.

표 3.21. 전체 프로세스 요약표

Phase	Activity	Task	입력물(Input)	산출물(Deliverable)
요구사항 분석	현황분석	표준화 원칙 이해	현행 표준화 지침	
요구사항 분석	현황분석	표준화 범위/방법 정의		표준화 대상/범위/방법
요구사항 분석	요구사항 정의	데이터 구성 요소 식별	표준화 대상 데이터 정보	
요구사항 분석	요구사항 정의	표준화 대상 데이터 정의		표준화 대상 데이터 형상
표준단어 정의	표준단어 분석	표준단어 분석	표준단어사전	
표준단어 정의	표준단어 정의	표준단어 후보정의		후보 표준단어
표준용어 정의	표준용어 분석	표준용어 분석	표준용어 사전	
표준용어 정의	표준용어 정의	표준용어 후보정의		후보 표준용어
표준도메인 정의	표준도메인 분석	표준도메인 분석	표준도메인 사전	
표준도메인 정의	표준도메인 정의	표준도메인 후보정의		후보 표준도메인
표준코드 정의	표준코드 분석	표준코드 분석	표준코드	
표준코드 정의	표준코드 정의	표준코드 후보정의		후보 표준코드
표준화 확정	표준화 확정	데이터 표준 확정	후보 표준화 내역, 표준코드	확정 표준화 내역, 표준코드
표준화 확정	표준화 공표	데이터 표준 공표	확정 표준화 내역, 표준코드	표준화 메타 데이터
표준화 적용	표준화 적용	논리 데이터 모델링	확정 표준화 내역	논리 ERD, 엔터티·속성 정의서
표준화 적용	표준화 적용	물리 데이터 모델링	논리 ERD, 정의서, 표준코드	물리 ERD, DB·테이블·컬럼 정의서
표준화 적용	표준화 관리	표준 준수 여부 점검	논리 ERD	비표준·변경 내역
표준화 적용	표준화 관리	변경관리	비표준·변경 내역, 메타 데이터, 모델·정의서	갱신된 메타 데이터, 모델·정의서

CHAPTER 4

산출물 템플릿

4.1 개요

산출물 템플릿(Deliverable Templates)는 데이터 표준화 프로세스의 각 단계에서 생성해야 하는 결과물의 **표준화된 양식**이다. 작업의 품질을 상향 평준화하고, 정보 누락을 방지하며, 표준화 관계자 간의 원활한 의사소통을 돕는 역할을 수행한다.

산출물 템플릿은 KDSM `4 Layered Architecture`의 Guidance Layer에 속하며, 같은 계층의 **표준 및 절차 매뉴얼(SPM, 3장)**과 짝을 이룬다. 표준 및 절차 매뉴얼이 "각 Task를 어떻게 수행하는가"(기준·절차)를 규정한다면, 산출물 템플릿은 그 수행 결과를 "어떤 양식에 담는가"(형상)를 규정한다.

> **i 노트**
>
> 본 장은 산출물의 **양식(항목 구성과 작성 예시)**에 집중한다. 각 산출물을 생성하는 Task의 목적·표준 기준·단계별 절차·체크리스트는 「표준 및 절차 매뉴얼」(3장)에서 다루므로, 본 장에서는 해당 Task를 상호참조로 연결하고 중복 기술하지 않는다.

4.2 템플릿 체계

4.2.1 산출물과 템플릿의 관계

데이터 표준화 프로세스는 Phase → Activity → Task의 3계층으로 정의되며, 각 Task는 하나 이상의 산출물(Deliverable)을 생성한다. 앞 Task의 산출물은 다음 Task의 입력물이 되므로, 산출물의 양식이 표준화되어 있어야 프로세스 전 구간에서 정보가 손실 없이 연결된다. 본 장의 각 템플릿은 이 산출물 흐름의 각 결과물에 대응한다.

4.2.2 템플릿 제공 형태

산출물 템플릿은 제공 매체에 따라 두 가지로 구분된다.

- **파일형 템플릿**: 정의서·계획서·대장과 같이 문서(스프레드시트·표) 형태로 작성·관리되는 산출물. 작성자가 항목을 직접 채운다.
- **DB형 산출물**: 표준사전·표준코드 저장소와 같이 데이터베이스의 테이블로 관리되는 산출물. 테이블의 컬럼 구성이 곧 템플릿이 된다.

두 형태는 항목 구성이 동일하므로, 본 장의 템플릿은 파일과 DB 어느 매체로 구현하든 그대로 적용할 수 있다.

4.2.3 템플릿 기술 방식

본 장의 각 산출물 템플릿은 다음 방식으로 기술한다.

- **템플릿 항목**: 산출물을 구성하는 항목을 `항목 ¦ 설명 ¦ 필수` 표로 정의한다. `필수`는 반드시 작성해야 하는 항목(필수)과 상황에 따라 작성하는 항목(선택)을 구분한다.
- **작성 예시**: 실제 작성 모습을 이해할 수 있도록 예시를 제시한다. 예시는 본 방법론 전반에서 사용하는 고객·주문 업무 데이터를 공통 소재로 삼는다.

원 프로세스 정의서에 명시되지 않아 일반적인 데이터 모델링 모범사례로 보강한 항목은 `[모범사례]`로 표시하여 구분한다.

4.2.4 프로세스 산출물–템플릿 매핑

전체 프로세스가 생성하는 산출물과 본 장에서 제공하는 템플릿의 대응 관계는 다음과 같다. 산출물의 상세 입력물·산출물 정의는 「표준 및 절차 매뉴얼」의 전체 프로세스 요약표(표 3.21)를 참조한다.

표 4.1. 프로세스 산출물–템플릿 매핑

Phase	산출물(Deliverable)	산출물 템플릿	제공 형태
1. 요구사항분석	표준화 대상/범위/방법	표준화 계획서	파일
1. 요구사항분석	표준화 대상 데이터 형상	표준화 대상 데이터 정의서	파일
2. 표준단어 정의	후보 표준단어	표준단어 정의서	파일·DB
3. 표준용어 정의	후보 표준용어	표준용어 정의서	파일·DB
4. 표준도메인 정의	후보 표준도메인	표준도메인 정의서	파일·DB
5. 표준코드 정의	후보 표준코드	표준코드 정의서	파일·DB
6. 표준화 확정·공표	표준화 메타 데이터	표준 확정·공표 대장	DB
7. 표준화 적용	논리 데이터 모델	ERD, 주제영역 정의서, 주제영역 구성 정의서, 엔터티 정의서, 속성 정의서	파일
7. 표준화 적용	물리 데이터 모델	데이터베이스·테이블·컬럼 정의서	파일
7. 표준화 적용	비표준·변경 내역	표준 준수 점검 결과서, 변경관리 대장	파일

4.2.4.1 일러두기

> ℹ 산출물 작성 예시는 일부만 제시한다
>
> 본 가이드에서 산출물의 사용 방법에 대한 이해를 높이고자 기술한 작성 예시는 모든 항목을 기술한 것이 아님을 밝힌다. **전체 항목이 포함된 작성 예시는 첨부 엑셀 파일의 템플릿을 참고**하기 바란다.

> 산출물 작성 방법을 제시한다
>
> 개별 산출물에 대한 작성 방법은 첨부 엑셀 파일의 템플릿 시트별 하단에 기술되어 있으므로, 작성 방법을 참고하여 산출물을 작성하면 된다.

4.3 요구사항분석 산출물 템플릿

4.3.1 표준화 계획서

표준화 계획서는 **표준화 범위·방법 정의** Task(3.2.1.2절)의 산출물로, 표준화의 목적·대상·범위·수행 방법을 확정하여 이후 모든 정의 활동의 기준선을 제시하는 문서이다.

표 4.2. 표준화 계획서 템플릿 항목

항목	설명	필수
표준화 목적	본 표준화 활동이 달성하려는 목표	필수
표준화 대상	대상 시스템·업무 영역	필수
포함 범위(In-scope)	표준화를 적용할 데이터 영역	필수
제외 범위(Out-of-scope)	표준화에서 제외하는 대상과 그 사유	필수
표준 도출 방식	상향식/하향식 등 표준 도출 접근	필수
기존 표준 재사용 원칙	신규 정의와 기존 표준 재사용의 판단 기준	필수
검토·승인 절차	표준 검토 및 승인 주체와 절차	필수
사용 도구	모델링 도구, 표준사전 관리 도구 등	선택
일정·담당	작업 일정과 수행·검토·승인 담당	선택

작성 예시

- 표준화 목적: 고객·주문 도메인의 명칭·형식 일관성 확보
- 포함 범위: 고객·주문·상품 도메인
- 제외 범위: 로그·임시 테이블(운영 편의 목적, 표준화 실익 낮음)
- 표준 도출 방식: 기존 표준사전 우선 재사용, 부재 시 신규 정의(상향식 보완)
- 검토·승인: 데이터 아키텍트 검토, 데이터 거버넌스 책임자 승인

4.3.2 표준화 대상 데이터 정의서

표준화 대상 데이터 정의서는 **표준화 대상 데이터 정의** Task(3.2.2.2절)의 산출물로, 표준화 대상이 되는 테이블화·컬럼화 요소를 목록화한 문서이다. 이후 단어·용어·도메인·코드 정의의 원천이 된다.

표 4.3. 표준화 대상 데이터 정의서 템플릿 항목

항목	설명	필수
대상 구분	테이블화 대상 / 컬럼화 대상	필수
개체(엔터티) 후보명	테이블화 대상 개체의 한글명	필수
속성 후보명	컬럼화 대상 속성의 한글명	필수
업무 정의	해당 개체·속성의 업무적 의미	필수
원천	현행 테이블·화면·보고서 등 출처	선택
담당 현업	정보를 제공한 현업 담당	선택
비고	코드 관리 가능성 등 특이사항	선택

작성 예시

표 4.4. 표준화 대상 데이터 정의서 작성 예시

대상 구분	개체 후보명	속성 후보명	업무 정의	원천
테이블화	고객	—	상품을 구매하는 거래 주체	현행 CUSTOMER 테이블
컬럼화	고객	고객번호	고객을 유일하게 식별하는 번호	CUSTOMER.CUST_ID
컬럼화	고객	생년월일	고객의 출생 연월일	고객가입 화면
컬럼화	주문	주문상태	주문의 진행 상태	주문관리 화면

4.4 데이터 표준 정의 산출물 템플릿

4.4.1 표준단어 정의서

표준단어 정의서는 **표준단어 정의** Task(3.3.2.1절)의 산출물로, 데이터 명칭을 구성하는 최소 의미 단위인 단어의 표준을 기록한다. 물리명 생성의 기준이 되는 영문 약어를 포함한다.

표 4.5. 표준단어 정의서 템플릿 항목

항목	설명	필수
표준단어명	표준으로 확정한 한글 단어명	필수
영문명	단어의 영문 표기	필수
영문 약어	물리명 생성에 사용할 표준 약어	필수
단어 설명	단어의 의미	필수
이음동의어	표준단어명으로 수렴되어야 할 동의어·유사어	선택
금칙어	사용을 금지할 단어	선택
단어 구분	일반어/도메인어/분류어 등 유형	선택
상태	후보/확정/폐기	필수

작성 예시

표 4.6. 표준단어 정의서 작성 예시

표준단어명	영문명	영문 약어	이음동의어	금칙어	상태
고객	Customer	CUST	거래처, 손님	customer(영문 혼용)	확정
번호	Number	NO	넘버	No., #	확정
금액	Amount	AMT	—	—	확정

4.4.2 표준용어 정의서

표준용어 정의서는 **표준용어 정의** Task(3.4.2.1절)의 산출물로, 표준단어를 조합하여 업무에서 실제로 사용하는 용어(엔터티명·속성명)의 표준을 기록한다.

표 4.7. 표준용어 정의서 템플릿 항목

항목	설명	필수
표준용어명	표준단어 조합으로 확정한 한글 용어명	필수
영문명	표준단어 영문명의 조합(대문자 Full Name)	필수
물리명	DBMS에 적용할 물리명(약어 조합)	선택
용어 설명	용어의 업무적 의미	필수
구성 표준단어	용어를 구성하는 표준단어 목록	필수
대표 도메인	용어에 연결되는 표준도메인	필수
표준코드명	해당 용어가 표준코드에 해당할 경우 연결되는 표준코드명	선택
용어 구분	엔터티명/속성명 등 유형	선택
상태	후보/확정/폐기	필수

작성 예시

표 4.8. 표준용어 정의서 작성 예시

표준용어명	영문명	물리명	구성 표준단어	대표 도메인	용어 설명
고객번호	CUSTOMER NUMBER	CUST_NO	고객 + 번호	번호형	고객을 유일하게 식별하는 번호
주문금액	ORDER AMOUNT	ORD_AMT	주문 + 금액	금액형	주문 건의 총 결제 금액

4.4.3 표준도메인 정의서

표준도메인 정의서는 **표준도메인 정의** Task(3.5.2.1절)의 산출물로, 데이터의 형식·유형·길이 등 물리적 표현 규칙을 기록한다. 속성(컬럼)의 데이터 타입 일관성을 보장한다. 표준도메인 정의서는 도메인 본체 정의와 함께 도메인 그룹·도메인 분류를 각각 별도 시트로 정의하며, 그룹·분류 체계는 행정안전부 공통표준도메인 체계(공통표준도메인그룹명·공통표준도메인분류명)를 준용한다.

표 4.9. 표준도메인 정의서 템플릿 항목

항목	설명	필수
표준도메인명	표준으로 확정한 도메인명	필수
도메인 그룹	표준도메인의 상위 구분(금액·날짜/시간·번호 등)	필수
도메인 분류	표준도메인의 세부 구분(가격·연월 등)	필수
데이터 유형	논리 데이터 유형	필수
길이	데이터 길이(전체 자릿수)	필수
소수 자릿수	소수점 이하 자릿수	선택
저장 형식	DBMS에 저장되는 값의 형식(예: YYYYMM)	선택
표현 형식	마스크·표시 형식	선택
단위	데이터 값을 측정하는 단위(m·kg·도 등)	선택
허용 값 범위	유효한 값의 범위·제약	선택
물리 데이터 타입	[모범사례] DBMS 물리 타입 (VARCHAR·NUMBER·DATE 등)	필수
상태	후보/확정/폐기	필수

작성 예시

표 4.10. 표준도메인 정의서 작성 예시

표준도메인명	도메인 그룹	도메인 분류	데이터 유형	길이	물리 데이터 타입	허용 값 범위
우편번호C5	번호	우편번호	문자	5	CHAR(5)	—
가격N10	금액	가격	숫자	10	NUMBER(10)	0 이상
연월일C8	날짜/시간	연월일	문자	8	CHAR(8)	유효 일자
여부C1	코드	여부	문자	1	CHAR(1)	Y/N

도메인 그룹 정의서 항목

표 4.11. 표준도메인 정의서 — 도메인 그룹 항목

항목	설명	필수
도메인 그룹명	도메인을 그룹핑하는 상위 구분명(금액·날짜/시간·번호 등)	필수
도메인 그룹 설명	도메인 그룹의 의미	선택
상태	후보/확정/폐기	필수

작성 예시

표 4.12. 도메인 그룹 작성 예시

도메인 그룹명	도메인 그룹 설명	상태
금액	화폐의 가치를 수로 표현하기 위한 도메인	확정
날짜/시간	특정 사건이 일어난 시점 또는 기간을 표현하기 위한 도메인	확정

도메인 분류 정의서 항목

표 4.13. 표준도메인 정의서 — 도메인 분류 항목

항목	설명	필수
도메인 분류명	도메인 그룹 하위의 세부 구분명(가격·비용·연월 등)	필수
도메인 분류 설명	도메인 분류의 의미	필수
도메인 그룹 이름	분류가 소속되는 상위 도메인 그룹명	필수
상태	후보/확정/폐기	필수

작성 예시

표 4.14. 도메인 분류 작성 예시

도메인 분류명	도메인 분류 설명	도메인 그룹 이름	상태
가격	물건이 지니고 있는 가치를 돈으로 나타낸 것	금액	확정
비용	어떤 일을 하는 데 드는 돈	금액	확정

4.4.4 표준코드 정의서

표준코드 정의서는 **표준코드 정의** Task(3.6.2.1절)의 산출물로, 한정된 값의 집합으로 관리하는 코드를 기록한다. 코드의 속성을 정의하는 **코드 마스터**와 개별 코드값을 정의하는 **코드값 상세**의 두 부분으로 구성된다.

표 4.15. 표준코드 정의서 — 코드 마스터 항목

항목	설명	필수
표준코드명	코드의 한글명	필수
코드 ID	코드를 식별하는 영문 코드명	필수
코드 유형	문자·숫자 등 코드값 유형	필수
자릿수	코드값의 자릿수	필수
기본값	기본 코드값	선택
코드 설명	코드의 의미·용도	필수
상태	후보/확정/폐기	필수

표 4.16. 표준코드 정의서 — 코드값 상세 항목

항목	설명	필수
코드 ID	코드값이 속한 표준코드의 코드 ID(표준용어 물리명)	필수
코드값	실제 저장되는 코드값	필수
코드값 명	코드값의 한글 의미	필수
코드값 설명	코드값의 상세 설명	선택
정렬 순서	화면·목록 표시 순서	선택
사용 여부	사용/미사용	필수

작성 예시 — 표준코드 “주문상태코드”(코드 ID: `ORD_STAT_CD`, 2자리)

표 4.17. 표준코드 정의서 작성 예시

코드값	코드값 명	정렬 순서	사용 여부
01	접수	1	사용
02	배송중	2	사용
03	완료	3	사용
09	취소	4	사용

4.4.5 산출물 템플릿 파일 (Excel)

데이터 표준 정의 산출물(표준단어·표준용어·표준도메인·표준코드 정의서)은 하나의 Excel 템플릿으로 제공한다. 파일은 표지, 정의서 양식 시트(표준도메인은 도메인·도메인 그룹·도메인 분류로, 표준코드는 코드 마스터·코드값 상세로 분리), 별첨 작성 예시로 구성된다. 출력 형식에 따라 전달 방식이 다르다.

인쇄본(PDF)이나 전자책에서는 파일을 지면에 직접 첨부할 수 없으므로, 배포 사이트에서 내려받도록 안내한다. 아래 주소로 접속하면 동일한 Excel 템플릿을 받을 수 있다.

다운로드(표준 정의 DM-DST-001): 데이터 표준 정의서 템플릿 내려받기 (.xlsx)

4.5 표준화 확정·공표 산출물 템플릿

4.5.1 표준 확정·공표 대장

표준 확정·공표 대장은 **데이터 표준화 확정**(3.7.1.1절)과 **데이터 표준화 공표**(3.7.2.1절) Task의 산출물로, 확정된 표준(단어·용어·도메인·코드)을 표준사전에 등록하고 그 확정·공표 이력을 관리하는 산출물이다. 등록 이후에는 변경관리 절차를 통해서만 수정하며, 조직 전체의 단일 참조원(Single Source of Truth)이 된다.

표 4.18. 표준 확정·공표 대장 템플릿 항목

항목	설명	필수
표준 유형	표준단어/표준용어/표준도메인/표준코드	필수
표준명	확정된 표준의 명칭	필수
표준 사전 버전	공표되는 표준사전의 버전	필수
확정일	이해관계자 합의·확정 일자	필수
적용 시점	표준이 적용되는 시점	필수
적용 대상	표준이 적용되는 시스템·범위	필수
검토자·승인자	검토 및 공표 승인 주체	필수
변경 이력	[모범사례] 확정 과정의 주요 변경 내역	선택
상태	확정/공표/폐기	필수

작성 예시

- 표준 사전 버전: v2.0
- 확정일: 2026-07-15 / 적용 시점: 2026-08-01
- 적용 대상: 신규 개발 전 시스템
- 검토자: 데이터 아키텍트 / 승인자: 데이터 거버넌스 책임자

4.6 표준화 적용 산출물 템플릿

4.6.1 논리 데이터 모델 다이어그램(ERD)

ERD는 **논리 데이터 모델링** Task(3.8.1.1절, 물리 모델은 3.8.1.2절)의 산출물로, 엔터티와 관계를 도식으로 표현한 다이어그램이다. 다이어그램 자체는 모델링 도구로 작성하되, 산출물로 관리하기 위한 식별 정보(모델 헤더)를 다음 항목으로 함께 기록한다.

표 4.19. ERD 모델 헤더 템플릿 항목

항목	설명	필수
모델명	데이터 모델의 명칭	필수
모델 구분	논리 모델 / 물리 모델	필수
버전	모델 버전	필수
대상 업무	모델이 다루는 업무 영역	필수
표기법	IE/Barker 등 ERD 표기법	선택
대상 DBMS	[모범사례] 물리 모델의 대상 DBMS	선택
작성자·작성일	모델 작성 주체와 일자	필수

4.6.2 주제영역 정의서

주제영역 정의서는 **논리 데이터 모델링** Task(3.8.1.1절)의 산출물로, 논리 데이터 모델을 업무 단위로 나눈 주제영역(Subject Area)의 명세를 기록한다. 엔터티가 수백 개에 이르면 모델 전체를 한 장의 ERD로 읽을 수 없으므로, 주제영역은 모델을 읽고 관리하는 단위이자 담당·검토 구획의 기준이 된다. 분할 원칙은 논리 데이터 모델링 가이드(5.7.2절)를, 명칭 규칙은 명명규칙(5.6.1절)을 따른다.

표 4.20. 주제영역 정의서 템플릿 항목

항목	설명	필수
주제영역명	표준용어로 정의한 주제영역 한글명. 하나의 논리 데이터 모델 안에서 유일해야 한다	필수
주제영역 순번	산출물에 싣는 차례. 1부터 빈틈없이 부여하며 업무 흐름의 순서를 따르는 것이 일반적이다	필수
주제영역 설명	주제영역의 업무적 범위. 어떤 엔터티가 이 영역에 속하고 어떤 것이 속하지 않는지 가릴 수 있도록 기술한다	필수

작성 예시

표 4.21. 주제영역 정의서 작성 예시

주제영역명	주제영역 순번	주제영역 설명
영업	1	영업업무 관련된 활동 및 평가 주제영역
상품	2	상품 및 상품 속성 관련 주제영역

4.6.3 주제영역구성 정의서

주제영역구성 정의서는 **논리 데이터 모델링** Task(3.8.1.1절)의 산출물로, 각 엔터티를 어느 주제영역에 배정했는지를 기록한다. 주제영역 정의서가 구획을 선언한다면, 주제영역구성 정의서는 그 구획에 무엇이 들어가는지를 채운다. 구성대상을 `테이블`로 지정하면 물리 데이터 모델(4.6.7절)의 주제영역 구획에도 같은 양식을 그대로 적용할 수 있다.

표 4.22. 주제영역구성 정의서 템플릿 항목

항목	설명	필수
소속 주제영역	배정 대상이 속한 주제영역명. 주제영역 정의서에 등재한 이름을 그대로 사용한다	필수
구성 순서	한 주제영역 안에서 배정 대상을 늘어놓는 차례. 1부터 부여한다	필수
구성대상	배정 대상의 구분. 논리 모델의 것이면 `엔터티`, 물리 모델의 것이면 `테이블`	필수
구성대상 이름	엔터티명 또는 테이블 물리명. 같은 산출물에 정의된 대상이어야 한다	필수
사용 부서	해당 대상을 주로 사용하는 부서명. 부서 개념이 없으면 비워 둔다	선택

작성 예시

표 4.23. 주제영역구성 정의서 작성 예시

소속 주제영역	구성 순서	구성대상	구성대상 이름	사용 부서
영업	1	엔터티	고객	영업기획팀
영업	2	엔터티	주문	영업기획팀
상품	1	엔터티	상품마스터	상품개발팀
상품	2	엔터티	상품상세	상품개발팀

4.6.4 엔터티 정의서

엔터티 정의서는 **논리 데이터 모델링** Task(3.8.1.1절)의 산출물로, 표준용어로 정의한 엔터티의 명세를 기록한다.

표 4.24. 엔터티 정의서 템플릿 항목

항목	설명	필수
엔터티명	표준용어로 정의한 엔터티 한글명	필수
영문명	엔터티 영문명	필수
엔터티 설명	엔터티의 업무적 의미	필수
식별자(PK)	엔터티를 식별하는 주 식별자	필수
보조 식별자(UK)	주 식별자 외에 엔터티에서 유일해야 하는 속성 조합. `UKn:속성명` 형식으로 표기	선택
엔터티 구분	기본/중심/행위 엔터티 등 유형	선택
상위 엔터티	상속·부모 관계 엔터티	선택
개인정보 포함여부	엔터티 속성에 개인정보가 포함되는지 여부(Y/N). 보안 등급 분류·법적 규제 준수	필수
주요 관계	다른 엔터티와의 관계	선택

작성 예시

표 4.25. 엔터티 정의서 작성 예시

엔터티명	영문명	식별자	엔터티 구분	개인정보 포함여부	설명
고객	CUSTOMER	고객번호	기본	Y	상품을 구매하는 거래 주체
주문	ORDER	주문번호	행위	N	고객이 상품을 구매한 거래

4.6.5 속성 정의서

속성 정의서는 **논리 데이터 모델링** Task(3.8.1.1절)의 산출물로, 각 엔터티에 속한 속성을 표준용어·표준도메인으로 정의한 명세이다.

표 4.26. 속성 정의서 템플릿 항목

항목	설명	필수
소속 엔터티	속성이 속한 엔터티명	필수
속성명	표준용어로 정의한 속성 한글명	필수
영문명	속성 영문명	필수
속성 유형	속성 값의 성격 유형(기본형/설계형/추출형)	선택
도메인	적용한 표준도메인	필수
기본키 순번	복합키 순서(1,2,3…), 기본키가 아니면 공란	선택
소속 유일키	보조 식별자에 참여하는 속성의 소속과 순번. `UKn-n` 형식으로 표기	선택
Null 허용	Null 허용 여부	필수
개인정보 여부	해당 속성의 개인정보 여부(Y/N). 보안 등급 분류·법적 규제 준수	필수
참조 엔터티	참조하는 관계가 있는 엔터티명(외래키인 경우)	선택
참조 속성명	참조하는 원천 속성명(외래키인 경우)	선택
기본값	속성의 기본값	선택
속성 설명	속성의 업무적 의미	필수

작성 예시

표 4.27. 속성 정의서 작성 예시

소속 엔터티	속성명	도메인	기본키 순번	Null	설명
고객	고객번호	번호형	1	불가	고객 식별 번호
고객	생년월일	일자형		허용	고객 출생 연월일
주문	주문상태코드	코드형		불가	주문상태코드 참조

4.6.6 데이터베이스 정의서

데이터베이스 정의서는 **물리 데이터 모델링** Task(3.8.1.2절)의 산출물로, 물리 모델이 적재될 데이터베이스의 명세를 기록한다.

표 4.28. 데이터베이스 정의서 템플릿 항목

항목	설명	필수
데이터베이스명	대상 데이터베이스명	필수
DBMS 종류·버전	Oracle·PostgreSQL 등 종류와 버전	필수
스키마명	대상 스키마	필수
문자셋	인코딩(UTF-8 등)	선택
저장 특성	테이블스페이스 등 저장 구성	선택
용도	운영/개발/분석 등 용도	선택

4.6.7 테이블 정의서

테이블 정의서는 **물리 데이터 모델링** Task(3.8.1.2절)의 산출물로, 엔터티를 변환한 테이블의 물리 명세를 기록한다.

표 4.29. 테이블 정의서 템플릿 항목

항목	설명	필수
테이블 물리명	표준 약어 조합으로 생성한 물리명	필수
테이블 한글명	대응하는 엔터티 한글명	필수
테이블 소유자	테이블 스키마의 소유권을 갖는 데이터베이스 사용자. 책임 소재·유지보수 담당 파악	선택
관련 엔터티	해당 테이블로 구현된 엔터티명(한글). 데이터 흐름·영향도 분석	선택
데이터베이스	테이블이 속한 데이터베이스명(생성 대상 정보 시스템)	필수
스키마	소속 스키마	필수
테이블 유형	기본/이력/코드 등 유형	선택
기본키(PK)	테이블의 기본키 컬럼	필수
유일키(UK)	기본키 외에 테이블에서 유일해야 하는 컬럼 조합. `UKn:컬럼명` 형식으로 표기	선택
예상 건수	예상 데이터 건수	선택
테이블 설명	테이블의 용도·의미	필수
보관 기간(년)	테이블 데이터의 보관 기간(년). 데이터 수명주기·유지보수 관리 목적	선택
발생주기	데이터의 생성·수정 주기(실시간/일/월 등). 용량 산정·성능 최적화	선택
백업 여부	테이블 백업 대상 여부(Y/N). 유지보수 관리 목적	선택

작성 예시

표 4.30. 테이블 정의서 작성 예시

테이블 물리명	한글명	기본키	설명
`TB_CUST_BASE_INFO`	고객	`CUST_NO`	고객 기본 정보
`TB_ORD_MST`	주문	`ORD_NO`	주문 마스터

4.6.8 컬럼 정의서

컬럼 정의서는 **물리 데이터 모델링** Task(3.8.1.2절)의 산출물로, 속성을 변환한 컬럼의 물리 명세를 기록한다. 컬럼의 데이터 타입은 표준도메인의 물리 정의를 따른다.

표 4.31. 컬럼 정의서 템플릿 항목

항목	설명	필수
소속 테이블	컬럼이 속한 테이블 물리명	필수
컬럼 순번	테이블 내 컬럼의 순번. 특정 컬럼 기준으로 정렬해도 원래 순서를 복원할 수 있게 한다	필수
컬럼 물리명	표준 약어 조합으로 생성한 물리명	필수
컬럼 한글명	대응하는 표준용어 한글명	필수
데이터 타입	표준도메인의 물리 데이터 타입	필수
길이	데이터 길이	필수
Null 허용	Null 허용 여부	필수
기본키 순번	복합키 순서(1,2,3…), 기본키가 아니면 공란	선택
소속 유일키	유일키에 참여하는 컬럼의 소속과 순번. `UKn-n` 형식으로 표기	선택
기본값	컬럼 기본값	선택
참조 테이블	외래키인 경우 참조 대상 테이블 물리명	선택
참조 컬럼	외래키인 경우 참조 대상 컬럼 물리명	선택
참조 코드	코드를 참조하는 경우 코드 명칭/아이디	선택
개인정보 여부	컬럼 값의 개인정보 포함 여부(Y/N). 보안 등급 분류·법적 규제 준수	필수
암호화 여부	개인정보 보호 등으로 컬럼을 암호화하는지 여부(Y/N). 보안 등급 분류·법적 규제 준수	필수
컬럼 설명	컬럼의 의미	필수

작성 예시

표 4.32. 컬럼 정의서 작성 예시

소속 테이블	컬럼 순번	컬럼 물리명	한글명	데이터 타입	Null	기본키 순번
`TB_CUST_BASE_INFO`	1	`CUST_NO`	고객번호	`VARCHAR(20)`	불가	1
`TB_CUST_BASE_INFO`	2	`CUST_NM`	고객명	`VARCHAR(100)`	불가	
`TB_CUST_BASE_INFO`	3	`BRDT`	생년월일	`CHAR(8)`	허용	

4.6.9 산출물 템플릿 파일 (Excel)

표준화 적용 단계의 정의서 산출물은 논리 데이터 모델(주제영역·주제영역구성·엔터티·속성 정의서)과 물리 데이터 모델(데이터베이스·테이블·컬럼 정의서) 두 개의 Excel 템플릿으로 제공한다. 각 파일은 표지, 양식 시트, 별첨 작성 예시로 구성된다. 출력 형식에 따라 전달 방식이 다르다.

인쇄본(PDF)이나 전자책에서는 파일을 지면에 직접 첨부할 수 없으므로, 배포 사이트에서 내려받도록 안내한다. 아래 주소로 접속하면 동일한 Excel 템플릿을 받을 수 있다.

다운로드(논리 DM-DST-002): 논리 데이터 모델 정의서 템플릿 내려받기 (.xlsx)

다운로드(물리 DM-DST-003): 물리 데이터 모델 정의서 템플릿 내려받기 (.xlsx)

4.7 표준화 관리 산출물 템플릿

4.7.1 표준 준수 점검 결과서

표준 준수 점검 결과서는 **표준 준수 여부 점검** Task(3.8.2.1절)의 산출물로, 작성된 데이터 모델이 공표된 표준을 준수하는지 점검한 결과와 비표준 내역을 기록한다.

표 4.33. 표준 준수 점검 결과서 템플릿 항목

항목	설명	필수
점검 대상	점검한 테이블·컬럼 등 대상	필수
점검 항목	명칭/도메인/코드 준수 등 점검 관점	필수
준수 여부	준수/위반	필수
비표준 내역	표준을 위반한 구체적 내용	필수
원인	비표준이 발생한 원인	필수
조치 방안	표준화·예외 승인 등 조치 방향	필수
점검 일자·점검자	점검 수행 일자와 담당	선택

작성 예시

- 점검 대상: `TB_CUST_BASE_INFO.CUSTOMER_NUMBER`
- 점검 항목: 명칭(물리명) 표준 약어 준수
- 준수 여부: 위반 / 원인: 표준 약어(`CUST_NO`) 미적용
- 조치 방안: `CUST_NO`로 변경 대상 등록

4.7.2 변경관리 대장

변경관리 대장은 **변경관리** Task(3.8.2.2절)의 산출물로, 표준·모델·산출물 전반의 변경을 요청-검토-승인-반영의 이력으로 관리하는 산출물이다.

표 4.34. 변경관리 대장 템플릿 항목

항목	설명	필수
변경 요청 ID	변경 건을 식별하는 번호	필수
요청일·요청자	변경 요청 일자와 요청 주체	필수
변경 대상	표준/모델/산출물 등 변경 대상	필수
변경 유형	신설/수정/폐기	필수
변경 내용	변경의 구체적 내용	필수
영향 범위	변경이 미치는 테이블·산출물 범위	필수
검토자·승인자	변경 검토 및 승인 주체	필수
반영일	변경이 반영된 일자	필수
상태	요청/검토/승인/반영/반려	필수

작성 예시

표 4.35. 변경관리 대장 작성 예시

요청 ID	변경 대상	변경 유형	변경 내용	영향 범위	상태
CR-2026-018	표준코드	신설	고객등급코드 신설	관련 테이블 3개	반영

4.8 템플릿 운영 지침

! 표준사전·표준코드는 DB형으로 운영한다

표준단어·표준용어·표준도메인·표준코드는 조직 전체가 참조하는 표준사전으로, 파일이 아닌 **데이터베이스 테이블(또는 표준사전 관리 도구)**로 관리하는 것을 원칙으로 한다. 파일형 정의서는 초기 정의·검토 단계의 작업본으로 활용하고, 확정 후에는 표준사전에 등록하여 단일 참조원을 유지한다.

템플릿은 버전과 함께 관리한다

모든 산출물은 버전과 상태(후보/확정/폐기)를 함께 기록한다. 특히 확정 이후의 변경은 반드시 변경관리 대장(4.7.2절)을 거쳐 표준사전·모델·정의서에 일관되게 반영한다.

i 자동 생성으로 산출물 품질을 균질화한다

정의서·대장과 같은 정형 산출물은 표준사전으로부터 자동 생성하면 작성 오류와 항목 누락을 줄일 수 있다. 산출물 자동 생성의 도구와 운용 방법은 Support Layer의 라이브러리/툴킷 및 기술 노트에서 다룬다.

IV Support Layer

5 가이드 81
5.1 개요 81
5.2 표준단어 작성 가이드 83
5.3 표준용어 작성 가이드 87
5.4 표준도메인 작성 가이드 91
5.5 표준코드 작성 가이드 95
5.6 명명규칙 98
5.7 논리 데이터 모델링 가이드 102
5.8 물리 데이터 모델링 가이드 105
5.9 코딩 가이드 107

6 테크니컬 노트 119
6.1 개요 119
6.2 테크니컬 노트 체계 120
6.3 데이터 표준 정의 테크니컬 노트 121
6.4 데이터 모델 정의 테크니컬 노트 134

7 라이브러리 149
7.1 개요 149
7.2 라이브러리 체계 151
7.3 설치와 환경 준비 153
7.4 데이터 표준 정의 지원 156
7.5 데이터 모델 정의 지원 161
7.6 운영 단계의 진단 165
7.7 작업 흐름과 자동화 168

8 툴킷 173
8.1 개요 173
8.2 툴킷 체계 174
8.3 데이터 표준 정의 지원 177
8.4 데이터 모델 정의 지원 183
8.5 산출물의 생성과 적재 191
8.6 점검 192
8.7 관리 196
8.8 구축과 운영 197

CHAPTER 5

가이드

5.1 개요

지침/규칙(가이드; Guides)는 **"목적을 이루기 위해 따라야 할 방향성과 권장 사항"**이며 **"반드시 준수해야 하는 엄격한 제약이자 기준"**이다. 지침은 비효율적인 시행착오를 줄여주고, 규칙은 치명적인 오류를 막아 준다.

본 방법론에서 가이드는 프로세스(제2장)와 표준 및 절차 매뉴얼(제3장)이 정의한 기준을 이어받아, 산출물 템플릿(4장)을 올바르게 작성하기 위한 표준화 원칙과 판단 기준을 제시한다. 여기서 제시한 원칙을 개별 항목에 적용하는 구체적인 작성·기술 방법은 테크니컬 노트(제6장)에서 한층 구체화하여 다룬다.

데이터 표준이 없는 조직의 데이터베이스는 시간이 지날수록 필연적으로 무질서해진다. 같은 고객 번호가 어떤 테이블에서는 `CUST_NO VARCHAR(10)`, 다른 테이블에서는 `CUSTOMER_ID NUMBER(10)`로 존재하고, '해지'와 '해약'과 '탈퇴'가 서로 다른 컬럼명으로 흩어지며, 날짜가 `DATE`, `CHAR(8)`, `VARCHAR(10)('2026-07-15')`의 세 가지 형식으로 공존하게 된다. 이러한 무질서는 단순한 미관의 문제가 아니라 다음과 같은 실질적 비용으로 전이된다.

- **조인 실패와 성능 저하**: 타입이 다른 컬럼 간 조인은 묵시적 형변환을 유발하여 인덱스를 무력화한다.
- **의미 해석 비용**: 개발자마다 컬럼의 의미를 원 개발자에게 물어보거나 데이터를 직접 열어 추측해야 한다.
- **중복과 불일치**: 같은 사실이 여러 곳에 다른 형식으로 저장되어 어느 값이 진실인지 판별할 수 없게 된다.
- **변경의 공포**: 명칭과 의미의 대응 관계가 불투명하므로 영향도 분석이 불가능해지고, 아무도 기존 구조를 건드리지 못하게 된다.

본 장의 가이드는 이러한 엔트로피 증가를 사전에 차단하는 통제 장치이다. 각 조항은 다음 두 수준으로 구분하여 기술한다.

표 5.1. 가이드 조항의 수준 구분

수준	표기	의미	위반 시 처리
규칙(Rule)	[필수]	예외 없이 준수해야 하는 강제 조항	표준 심의 반려. 반영 불가
지침(Guideline)	[권장]	특별한 사유가 없는 한 따라야 하는 권고 조항	예외 사유서 작성 및 데이터 관리 조직(DA)의 승인 필요

[권장] 조항의 예외를 승인한 경우에도 예외는 누적 관리한다. 동일한 예외가 반복 승인되면 이는 표준 자체가 현실과 괴리되었다는 신호이므로, 예외를 계속 승인할 것이 아니라 표준 개정 절차를 발의해야 한다. 예외 관리 대장이 없는 표준은 시간이 지나면 "지켜지지 않는 문서"로 전락한다. 표준의 제정 못지않게 표준의 준수·예외·개정을 다루는 거버넌스가 데이터 관리 성숙도를 결정한다 (DAMA International 2017년).

5.1.1 목적

본 가이드의 목적은 데이터 모델링과 데이터 표준화 과정에서 산출되는 모든 명칭·형식·구문을 **하나의 일관된 체계**로 통제하여, 다음을 달성하는 데 있다.

- **의사소통의 명확화**: 업무 담당자, 모델러, 개발자, 운영자가 동일한 데이터를 동일한 이름과 의미로 부르게 한다. "One Fact, One Name" — 하나의 사실에는 하나의 이름만 존재해야 하며, 역으로 하나의 이름은 하나의 사실만 가리켜야 한다. 이 양방향 유일성이 무너지면 그 지점에서 반드시 커뮤니케이션 오류가 발생한다.
- **데이터 품질의 구조적 보장**: 데이터 품질을 사후 검사(profiling)로 확보하는 것은 한계가 있다. 표준은 잘못된 데이터가 만들어질 수 있는 구조 자체를 봉쇄하는 **사전 예방적 품질 장치**다 (ISO 2016년; ISO/IEC 2015년b). 예컨대 여부 도메인을 `CHAR(1)`, 허용값 `'Y'/'N'`으로 강제하면 `'y'`, `'1'`, `'예'` 같은 오염 값의 발생 가능성이 스키마 수준에서 차단된다.
- **개발·운영 생산성 향상**: 명명과 형식에 관한 의사결정은 프로젝트마다, 개발자마다 반복되는 소모적 논쟁의 원천이다. 표준은 이 결정을 한 번만 내리고 재사용하게 함으로써 설계 리드타임을 단축한다.
- **시스템 간 상호운용성 확보**: 시스템·기관 간 데이터 연계 및 통합 시 명칭·형식 변환 매핑의 규모가 통합 비용을 결정한다. 표준을 공유하는 시스템 간의 연계는 매핑이 거의 필요 없다.
- **자동화의 기반 마련**: 표준 사전이 기계가독형(machine-readable)으로 관리되면 물리명 자동 생성, 모델–DB 정합성 검증, 비표준 컬럼 자동 탐지, DDL 자동 생성 등 데이터 관리 업무 전반을 자동화할 수 있다. 표준화는 문서 작업이 아니라 **자동화 가능한 메타데이터의 구축**이다.

5.1.2 적용 범위

본 가이드는 데이터 표준화 및 데이터 모델링 활동의 전 과정에 적용하며, 구체적인 적용 대상은 다음과 같다.

- **표준 사전 산출물**: 표준단어 사전, 표준용어 사전, 표준도메인 사전, 표준코드 사전의 작성·변경·폐기
- **논리 데이터 모델**: 주제영역, 주제영역구성, 엔터티, 속성, 관계, 식별자의 명명 및 정의
- **물리 데이터 모델**: 테이블, 컬럼, 인덱스, 제약조건, 뷰, 시퀀스 등 데이터베이스 오브젝트의 명명 및 정의
- **SQL 및 데이터베이스 접근 코드**: 신규 개발·변경되는 모든 SQL 구문, 저장 프로시저, 배치 스크립트

- **적용 시점**: 신규 시스템 구축 시 전면 적용을 원칙으로 한다. 운영 중인 시스템은 재구축·고도화 등 데이터베이스 개선 시점에 적용하되, 적용 전까지는 비표준 항목과 표준 항목 간의 **매핑 정보**를 관리하여 전환 근거를 보존한다 (행정안전부 2025년a).

다만 다음은 본 가이드의 적용 대상에서 제외한다.

- 패키지 솔루션·상용 소프트웨어가 자체적으로 생성·관리하는 내부 오브젝트. 단, 패키지의 데이터를 추출·적재하는 인터페이스 영역(스테이징)부터는 표준을 적용한다.
- 외부 기관이 제공하는 인터페이스 규격에 종속되어 임의 변경이 불가능한 항목. 이 경우 수신 원본은 규격대로 저장하되, 내부 활용 계층에서는 표준 명칭·형식으로 변환하여 관리하고 원본–표준 매핑을 문서화한다.
- 분석 실험용 샌드박스 등 수명이 명시적으로 한정된 일회성 영역. 단, 샌드박스 산출물이 운영으로 승격되는 시점에는 표준 적용이 선행되어야 한다.

5.2 표준단어 작성 가이드

표준단어(Standard Word)는 용어를 구성하는, 더 이상 나눌 수 없는 **의미의 최소 단위**이다. 표준화 체계 전체를 화학에 비유하면 단어는 **원자**, 용어는 원자가 결합한 **분자**, 도메인은 분자의 물성을 결정하는 **결합 규칙**에 해당한다. 원자가 오염되면 그것으로 만들어지는 모든 분자가 오염되듯, 단어 사전의 품질 결함은 용어 사전 전체로 증폭되어 전파된다. 표준단어 사전이 데이터 표준화의 출발점이자 가장 엄격하게 통제되어야 하는 산출물인 이유가 여기에 있다 (행정안전부 와/과 한국지능정보사회진흥원 2021년).

5.2.1 표준단어의 수집과 선정

단어 사전은 책상 위에서 창작하는 것이 아니라 **현행 데이터로부터 채굴**하는 것이다 (한국데이터산업진흥원 2021년). 수집·선정은 다음 절차를 따른다.

1. **전수 추출**: 현행 데이터베이스의 테이블·컬럼 주석(comment), 기존 논리 모델의 엔터티·속성명, 화면·보고서의 항목 레이블, 업무 매뉴얼·법령·규정의 용어 정의를 대상으로 명칭을 전수 추출한다. 추출 원천별로 출처를 기록해 두면 이후 대표어 선정 시 판단 근거가 된다.
2. **단어 분해(파싱)**: 추출된 명칭을 의미의 최소 단위로 분해한다. 예: “고객주민등록번호” → 고객 + 주민등록 + 번호. 이 단계에서 분해 결과가 흔들리면 이후 모든 용어 파싱이 흔들리므로, 분해 기준(5.2.3절 참조)을 먼저 확정하고 진행한다.
3. **빈도·중요도 분석**: 분해된 단어의 출현 빈도를 집계하고, 업무 핵심 개념(고객, 계약, 상품 등)인지, 법령·감독규정에 정의된 단어인지 표시한다.
4. **정제와 판정**: 표기 오류·비속한 축약(“주민번호”, “데이타”)을 정리하고, 각 단어를 표준어·동의어·금칙어로 판정한다.
5. **영문명·영문약어 부여**: 판정이 끝난 표준어에 대해 영문 명명 규칙(5.2.7절 참조)에 따라 영문명과 약어를 부여하고 전체 사전에 대해 중복 검사를 수행한다.

- **[필수]** 공공 부문이거나 공공기관과의 데이터 연계가 있는 경우, 행정안전부가 보급하는 **공통표준단어를 우선 준용**하고, 기관 고유 개념에 한해 기관표준단어를 추가 제정한다. 공통표준과 기관표준이 충돌하면 공통표준이 우선한다. 공공데이터 공통표준의 제정·준용 의무는 「공공데이터의 제공 및 이용 활성화에 관한 법률」 제23조와 행정안전부 고시에 근거한다 (대한민국 국회 2013년; 행정안전부 2025년a, 2025년b).
- **[권장]** 단어 사전의 초기 구축 시 목표 규모를 의식적으로 관리한다. 일반적인 전사 사전은 표준어 기준 1,000~3,000개 수준에서 안정화되며, 이를 크게 초과한다면 분해가 덜 되었거나(복합어 과다) 이음동의어 정리가 덜 된 것이다.

5.2.2 단어 분해의 판정 기준

"어디까지 쪼갤 것인가"는 단어 표준화에서 가장 자주 논쟁이 되는 지점이다. 다음 세 가지 판정 기준을 순서대로 적용한다.

1. **의미 보존 기준**: 분해했을 때 각 조각이 원래의 의미를 유지하는가. '주민등록'을 '주민'+'등록'으로 나누면 "주민등록제도"라는 고유 개념이 사라지고 일반적인 '등록' 행위로 오독될 수 있다. 이 경우 '주민등록'을 하나의 복합 단어로 등록한다.
2. **재사용성 기준**: 분해된 조각이 다른 용어의 구성에도 독립적으로 쓰이는가. '전화'와 '번호'는 각각 '전화국번', '계좌번호' 등 다른 조합에 재사용되므로 분해가 유리하다. 반면 '스마트폰'을 '스마트'+'폰'으로 나누어 얻는 재사용 이익은 없다.
3. **관용성 기준**: 업무 현장에서 항상 붙여서만 사용되는가. 법령·규정에 고유명사처럼 정의된 단어(예: '사업자등록', '지식재산')는 관용 단위를 존중한다.

세 기준이 충돌할 때는 의미 보존을 최우선으로 한다. 분해의 목적은 재사용이지만, 의미가 훼손된 재사용은 표준화의 목적 자체를 부정하기 때문이다. 구체적인 분해 판정 사례는 표 5.2 와 같다.

표 5.2. 단어 분해 판정 사례

원래 명칭	잘못된 분해	올바른 분해	판정 근거
주민등록번호	주민+등록+번호	주민등록+번호	의미 보존('주민등록'은 제도 고유명)
계좌개설일자	계좌개설+일자	계좌+개설+일자	재사용성('개설'은 카드·펀드 등에 재사용)
신용카드	신용+카드	신용카드 (또는 신용+카드)	업권 관용에 따라 결정하되 전사 통일

- **[필수]** 복합어의 등록 여부와 분해 방식은 개별 담당자가 임의 판단하지 않고 표준 심의에서 결정하며, 결정 결과는 판정 근거와 함께 사전에 기록한다. 동일한 유형의 재논쟁을 방지하기 위함이다.

5.2.3 한글명 작성 원칙

- **[필수]** 하나의 단어는 하나의 의미만 가진다(단어 의미의 원자성). 단어 설명에는 "~등", "또는"과 같이 의미를 열어 두는 표현을 사용하지 않는다.
- **[필수]** 한글 맞춤법과 표준어 규정을 준수하고, 외래어는 국립국어원 외래어 표기법을 따른다 (국립국어원 2017년). (데이타→데이터, 화일→파일, 어플리케이션→애플리케이션)
- **[필수]** 단어는 명사 또는 명사형으로 정의한다. 동사·형용사가 필요한 개념은 명사형으로 전성하여 등록한다. (승인하다→승인, 유효한→유효)
- **[권장]** 의미가 지나치게 포괄적인 단어(정보, 자료, 사항, 관련)는 표준어 등록을 지양한다. 이런 단어는 용어에 들어가도 정보량을 더하지 못하면서 명칭만 길게 만든다. '고객정보번호'와 '고객번호'가 가리키는 것이 같다면 '정보'는 군더더기다.
- **[권장]** 한 글자 단어는 파싱 모호성의 주된 원인이므로 최소화한다. 특히 '수', '율', '명'처럼 분류어로 쓰이는 한 글자 단어와 일반 단어의 결합부에서 오파싱이 발생하기 쉬우므로(예: "수수료율" = 수수료+율 인가 수수+요율 인가), 자동 파싱 도구를 쓰는 조직은 한 글자 단어의 결합 규칙을 명시적으로 정의한다.

5.2.4 동음이의어의 처리

동음이의어(같은 표기, 다른 의미)는 "하나의 이름은 하나의 사실"이라는 대원칙을 정면으로 위협한다 (데이터가치연구소 2024년).

- **[필수]** 동음이의어는 원칙적으로 **해당 조직의 업무에서 지배적으로 사용되는 하나의 의미만** 표준으로 등록한다. 유통업이라면 가구(家具)를, 주택·복지 업무라면 가구(家口)를 등록하는 식이다.
- **[필수]** 두 의미 모두 핵심 업무 개념이어서 불가피하게 복수 등록해야 하는 경우, 표기 자체를 다르게 하여 충돌을 해소한다. 해소 방법은 다음 우선순위를 따른다.
  1. 한쪽을 더 구체적인 복합어로 등록: 이자(利子)와 이자(異字)가 충돌하면 후자를 '이체문자' 등 실제 업무 명칭으로 대체
  2. 수식어를 포함한 형태로 등록: '분리과세이자' 처럼 항상 결합형으로만 사용
  3. 최후 수단으로 의미 구분자를 부여: 가구(가정), 가구(집기) — 단, 구분자가 붙은 단어는 물리명 생성 시 구분자를 제외한다
- **[필수]** 동음이의어를 복수 등록한 경우 각각의 영문약어는 반드시 달라야 한다. 한글명의 충돌은 사람이 문맥으로 구별할 수 있지만, 물리명의 충돌은 시스템이 구별할 수 없다.

5.2.5 이음동의어의 처리

이음동의어(다른 표기, 같은 의미)는 동음이의어보다 훨씬 빈번하며, 방치하면 사전이 소리 없이 부풀어 오른다.

- **[필수]** 같은 의미의 단어 군(群)에서는 **하나의 대표어만** 표준으로 등록하고, 나머지는 동의어로 등록하여 대표어로의 매핑을 유지한다. (해지·해약·해제 → 업무 정의를 검토하여 대표어 확정)
- **[권장]** 대표어 선정은 다음 기준을 순서대로 적용한다.
  1. **법령·감독규정의 용어**: 법정 용어가 있으면 그것을 대표어로 한다(예: 보험업법상 '해지').
  2. **공통표준단어**: 공공 공통표준에 등재된 표기를 따른다.
  3. **사용 빈도**: 현행 시스템·문서에서의 출현 빈도가 높은 쪽을 택한다.
  4. **의미의 정확성**: 국어사전의 정의가 업무 의미와 더 정확히 일치하는 쪽을 택한다.
- **[권장]** 진짜 동의어와 "유사하지만 다른 개념"을 혼동하지 않도록 주의한다. '검사'와 '검토', '변경'과 '수정'은 업무에 따라 구별되는 개념일 수 있다. 통합 여부가 애매하면 두 단어의 업무 정의를 각각 작성해 보고, 정의가 사실상 같으면 통합하고 다르면 각각 등록하되 설명에 상호 구별 기준을 명시한다.

5.2.6 금칙어 관리

- **[필수]** 사용을 금지할 단어는 사전에서 삭제하는 것이 아니라 **금칙어로 등재**한다. 사전에 없는 단어는 검증 도구가 "미등록"으로만 판정하지만, 금칙어로 등재된 단어는 "금지 — 대체어: ○○"라고 능동적으로 안내할 수 있다 (데이터가치연구소 2024년). 금칙어 등재 시 반드시 대체 표준어를 함께 지정한다.
- **[권장]** 금칙어의 전형적인 유형은 다음과 같다.
  - 표기 오류·구식 표기: 데이타, 화일, 리포트(→보고서)
  - 정리된 동의어의 잔재: 해약(→해지)
  - 과도한 축약: 주민번호(→주민등록번호)
  - 의미 없는 범용어: 정보, 데이터(단독 사용 시)
  - 은어·내부 속어: 특정 프로젝트에서만 통용되던 코드네임

5.2.7 영문명·영문약어 작성 원칙

- **[필수]** 모든 표준단어는 **영문명(Full Name)**과 **영문약어(Abbreviation)**를 함께 정의한다 (ISO/IEC 2015년a). 영문약어는 물리 모델의 테이블·컬럼명 구성 재료가 된다.

- **[필수]** 한글 단어와 영문약어는 **양방향 1:1**로 대응한다. 하나의 한글 단어에 두 개의 약어가 있으면 물리명이 갈라지고, 하나의 약어가 두 한글 단어에 대응하면 물리명에서 의미를 복원할 수 없다. 신규 약어 등록 시 사전 전체에 대한 중복 검사를 통과해야 한다.
- **[필수]** 관용적으로 확립된 국제 약어(ID, IP, URL, GPS, ISBN 등)는 재약어화하지 않고 그대로 사용한다.
- **[권장]** 영문약어의 생성은 다음 규칙을 순서대로 적용하며, 상위 규칙으로 생성한 약어가 이미 사용 중이면 다음 규칙으로 내려간다.
  1. **관용 약어 확인**: 업계 표준 약어가 있으면 사용 (Number→NO, Amount→AMT, Date→DT, Name→NM, Code→CD, Sequence→SEQ, Management→MGMT)
  2. **모음 제거 축약**: 영문명에서 첫 글자를 제외한 모음을 제거 (Customer→CSTMR→길면 CUST로 절단)
  3. **음절 두문 축약**: 각 음절·구성어의 앞 글자들을 조합 (Standard Word→STD_WD의 STD)
  4. **앞부분 절단**: 위 방법이 모두 충돌하면 영문명의 앞 4~6자를 절단하고 충돌 회피 문자를 부가
- **[권장]** 약어 길이는 2~6자를 표준 범위로 하고 8자를 상한으로 한다. 물리명은 여러 약어를 `_`로 연결해 만들어지므로, 약어 하나하나의 길이가 곧 물리명의 "길이 예산"을 소모한다. DBMS 식별자 제한(Oracle 12.2 미만 30바이트, PostgreSQL 63바이트, SQL Server 128자)을 고려하면 단어당 평균 4자 내외가 실용적이다.
- **[권장]** 로마자 표기(음차)는 고유명사(회사명, 브랜드명, 지명)에 한해 허용하고, 일반 개념어에는 사용하지 않는다. (예: '마일리지'는 외래어이므로 MILG 가능, '고객'을 GOGAEK으로 음차하는 것은 금지)

5.2.8 형식단어(분류어)의 지정

- **[필수]** 단어 중 용어의 **마지막에 위치하여 데이터의 형식적 성격을 결정하는 단어**를 형식단어(분류어)로 지정하고, 사전에 형식단어 여부를 별도 플래그로 관리한다. 분류어는 도메인그룹과 1:1 또는 1:N으로 연결된다. (명→명 도메인그룹, 일자→일자 도메인그룹, 금액→금액 도메인그룹)
- **[필수]** 분류어는 용어의 끝에서만 분류어로 기능한다. '번호'가 용어 중간에 오면(예: 번호이동가입자수) 일반 수식어이며, 이 용어의 도메인은 마지막 분류어 '수'가 결정한다.
- **[권장]** 분류어의 집합은 도메인그룹의 집합과 함께 설계한다. 분류어를 먼저 확정하지 않으면 도메인 체계가 표류한다.

5.2.9 관리항목

표준단어 정의서는 최소한 다음 항목을 관리한다(표 5.3).

표 5.3. 표준단어 정의서 관리항목

관리항목	설명	필수 여부
단어 한글명	표준으로 정한 한글 단어명	필수
단어 영문명	한글명에 대응하는 영문 전체 명칭	필수
단어 영문약어명	물리명 구성에 사용하는 영문 약어	필수
단어 설명	업무 관점의 의미 정의(단일 의미로 완결)	필수
형식단어 여부	분류어 여부 및 연결 도메인그룹	필수
단어 유형	표준어/동의어/금칙어 구분	필수
대표어 참조	동의어·금칙어인 경우 매핑되는 표준어	조건부 필수
출처	법령·공통표준·업무규정 등 제정 근거	권장
이음동의어 목록	이 단어로 치환해야 하는 유의어	권장
제정·개정 이력	제정일, 변경일, 변경 사유, 승인자	필수

5.2.10 단어 사전 품질 체크리스트

단어 사전의 배포 전, 그리고 정기 점검 시 다음을 기계적으로 검증한다.

- 영문약어 중복이 없는가 (약어 → 한글명 역인덱스로 전수 검사)
- 한글명 중복(동음이의 미해소)이 없는가
- 동의어·금칙어가 대표어 참조를 모두 갖고 있는가, 참조 대상이 표준어인가
- 대표어 참조가 순환(A→B→A)하지 않는가
- 형식단어가 유효한 도메인그룹에 연결되어 있는가
- 설명이 비어 있거나 한글명을 단순 반복(“고객: 고객을 의미함”)하지 않는가

5.3 표준용어 작성 가이드

표준용어(Standard Term)는 표준단어의 조합으로 만들어지며 표준도메인이 적용되는, 업무에서 실제로 사용하는 데이터 항목의 명칭이다. 표준용어의 한글명은 논리 모델의 **속성명**으로, 영문약어명은 물리 모델의 **컬럼명**으로 사용된다 (한국정보통신기술협회 2023년). 즉 용어 사전은 논리 세계와 물리 세계를 잇는 **번역 사전**이며, 이 사전이 완비되면 속성명→컬럼명 변환은 사람의 판단이 개입하지 않는 기계적 치환이 된다.

5.3.1 용어의 구성 문법

- **[필수]** 용어는 표준단어 사전에 등록된 표준어로만 구성한다. 미등록 단어가 필요하면 단어 제정이 선행되어야 한다. 용어를 만들면서 시간이 급하다는 이유로 단어 제정을 건너뛰는 관행이 사전 붕괴의 시작이다.
- **[필수]** 용어의 조합은 다음 구조를 따르며, 각 구성 요소의 역할·규칙은 표 5.4 와 같다.

  용어 = 주제어(0개 이상) + 수식어(0개 이상) + 분류어(정확히 1개, 반드시 마지막)

표 5.4. 용어의 구성 요소

구성 요소	역할	출현 규칙	예시
주제어	데이터의 소속·발생처·대상을 나타냄	0개 이상, 용어의 앞부분	고객, 계약, 상품
수식어	의미를 한정·구체화함	0개 이상, 주제어와 분류어 사이	최초, 최종, 월별, 누적
분류어	데이터의 형식적 성격과 도메인을 결정	정확히 1개, 반드시 마지막	명, 번호, 일자, 금액, 코드, 여부, 수, 율

예를 들어 '고객최초가입일자'는 **고객**(주제어) + **최초**(수식어) + **가입**(수식어) + **일자**(분류어)로 파싱된다.

이 구조는 ISO/IEC 11179-5가 정의하는 객체어(object class term)–특성어(property term)–표현어(representation term)의 명명 체계와 대응한다 (ISO/IEC 2015년a). 주제어는 객체어에, 수식어는 한정어(qualifier)에, 분류어는 표현어에 해당하며, 본 방법론의 용어 문법은 이 국제 표준 구조의 한국어 실무 구현으로 볼 수 있다.

- **[필수]** 용어의 마지막 단어는 반드시 분류어여야 하며, 분류어가 도메인그룹을, 도메인그룹 내의 구체 도메인이 타입·길이를 결정한다. 분류어로 끝나지 않는 명칭(예: "고객최초")은 용어로 성립하지 않는다.
- **[권장]** 단어의 배열 순서는 "큰 개념에서 작은 개념으로, 소유자에서 피소유물로" 원칙을 따른다. '고객주소' (고객의 주소)이지 '주소고객'이 아니다. 시간 한정 수식어(최초, 최종, 직전)는 분류어 바로 앞에 배치한다. (계약최초체결일자 × → 계약체결최초일자 △ → **최초계약체결일자** 또는 **계약최초체결일자** 중 전사 단일 규칙 확정)
- **[권장]** 하나의 용어는 5단어 이내로 구성한다. 6단어 이상이 필요하다면 대개 엔터티 설계가 잘못되어 여러 사실이 하나의 속성에 눌려 담긴 경우이므로, 명명이 아니라 모델을 먼저 의심한다.

5.3.1.1 구성 요소의 개수

주제어와 수식어의 개수에는 물리적인 제한이 없다. 표 5.4 가 "0개 이상"으로 규정한 것도 그 때문이다. 다만 데이터의 의미를 명확하고 식별 가능하게 표현해야 한다는 기본 원칙에서 실무상의 제약이 파생된다. 표준 문서가 상한을 정하지 않았다는 사실이 단어를 마음껏 이어 붙여도 된다는 뜻은 아니다.

- **[필수]** 의미가 중복되는 단어를 겹쳐 쓰지 않는다. 수식어나 분류어에 주제어의 의미가 이미 담겨 있으면 그 단어를 제거한다. '직원+직원명+명칭'은 주제어 '직원'의 의미가 수식어 '직원명'에 이미 들어 있고 그 '명'이 분류어 '명칭'과도 겹치므로 '직원명'으로 축약한다. 같은 이유로 '주문+주문일자'는 '주문일자'가 된다.
- **[권장]** 통상적인 조합은 분류어를 포함하여 2~3단어로 수렴시킨다. 앞의 5단어는 넘어서는 안 되는 상한이고, 2~3단어는 대부분의 용어가 머물러야 할 범위이다. 단어가 늘어날수록 물리명이 길어져 DBMS의 식별자 길이 제한(Oracle 11g 이하의 30바이트 등)에 걸리고, 사람이 읽어 구조를 파악하는 속도도 떨어진다.
- **[권장]** 단어 수를 줄이려고 약어를 새로 만들지 않는다. 길이가 문제라면 줄일 것은 표기가 아니라 용어에 담긴 사실의 개수이다.

5.3.1.2 단어 순서와 예외적 수용

ISO/IEC (2015년a) 의 명명 원칙은 객체어가 특성어보다 앞에 오도록 규정하며, 본 방법론의 주제어–수식어 순서도 이를 따른다. 순서 자체가 뒤바뀌는 구조적 예외는 원칙적으로 인정하지 않는다. 의미의 모호성을 없애고 분류 체계를 일관되게 유지하기 위한 규칙이기 때문이다.

그러나 실무에서는 순서가 바뀐 것처럼 보이는 상황이 발생한다. 다음 세 경우는 조직의 표준화 지침에 따라 예외적으로 수용할 수 있으며, 어느 경우든 판단의 근거를 지침에 명문화하고 표준 관리 조직의 승인을 받는다.

표 5.5. 단어 순서의 예외적 수용 기준

상황	판별 기준	처리
한정어와 수식어의 구분이 모호한 경우	무엇이 핵심 대상(주제)이고 무엇이 성질(수식)인지 판별한다	한정어 등록 규칙을 조율하여 해소한다
관용적·사회적 표준 용어	법적 고시 명칭, 정부 행정표준용어, 전사적으로 대체 불가능한 관용어인가	하나의 복합 단어로 등록한다
언어 특성에 따른 도치	논리적 개념 구조와 화면 표시 명칭을 구분한다	개념 구조는 주제–수식을 유지하고 표시 명칭만 국어 어순을 따른다

한정어와 수식어의 구분 — 표준은 주제어나 수식어 앞에 이를 한정하는 한정어(qualifier)를 붙이는 것을 허용한다. 한국어 어순의 특성상 이 한정어 때문에 순서가 바뀐 것처럼 착각하기 쉽다. '개인고객번호'와 '고객개인번호'가 그 예이다. '개인'이 고객의 유형을 좁히는 조건이라면 [개인(한정어) + 고객(주제어)] + 번호(분류어) 구조이므로 순서를 지킨 것이다. 반대로 '고객'이 주제이고 '개인'이 그 특성을 뜻한다면 순서가 어긋난 오류 용어이다. 한정어의 등록 규칙을 엄격히 정해 두어야 이 판별이 사람마다 달라지지 않는다.

관용어의 수용 — 수십 년간 굳어져 순서를 강제로 바꾸면 현업에 혼란을 주는 용어는 예외로 인정한다. '주민등록번호'가 대표적이다. 정석대로라면 주민(주제어) + 등록(수식어) + 번호(분류어)이지만, 법적·행정적 공식 명칭이므로 '주민등록번호' 전체를 하나의 고유한 복합 단어로 등록한다. 이 예외는 법적 고시 명칭과 정부 행정표준용어, 그리고 전사적으로 대체가 불가능한 관용어에 한정하며, 편의를 이유로 범위를 넓히면 예외가 규칙을 잠식한다.

언어 특성에 따른 도치 — ISO/IEC (2015년a) 는 영미권을 중심으로 만들어진 표준이므로 번역·현지화 과정에서 어순이 충돌한다. `Employee Name`은 직원(주제어) + 이름(수식어)으로 양쪽 어순이 일치하지만, `Total of Invoice`를 직역한 '청구서의 총액'은 수식이 앞에 온 것처럼 보인다. 표준의 부속서는 아시아권 언어의 특성을 고려하여, 메타데이터의 논리적 개념 구조는 주제–수식 순서를 유지하되 화면에 출력되는 표시용 명칭(label)은 해당 국가의 어순을 따르도록 허용한다. 따라서 표준용어 사전에는 개념 구조대로 등록하고, 화면 표시 문구는 별도 항목으로 관리한다.

5.3.2 분류어의 선택 기준

분류어의 오용은 용어 표준화에서 가장 흔한 품질 결함이다. 혼동하기 쉬운 분류어 쌍의 구별 기준을 명확히 한다.

표 5.6. 분류어 선택 기준

분류어	사용 기준	예시	혼동 주의
코드	사전에 정의된 코드 체계의 값을 담을 때. 반드시 표준코드 사전의 코드와 연결	상품유형코드	코드값 자체가 아닌 자유 입력 문자열에 '코드'를 쓰지 않는다
구분/유형	단독 분류어로 쓰지 않는다. '구분코드', '유형코드' 형태로 코드와 결합하거나, 코드가 아니면 다른 분류어로 대체	회원구분코드	'고객구분'처럼 코드 없이 끝나는 용어 금지
번호	업무 규칙에 따라 체계적으로 부여되는 식별 번호. 사람이 읽고 사용하는 번호	계좌번호, 사업자등록번호	연산 대상이 아니므로 문자 타입
ID	시스템이 부여하는 기술적 식별자	사용자ID, 세션ID	업무 번호와 시스템 ID를 혼용하지 않는다
일련번호	동일 부모 내에서의 발생 순서를 나타내는 순번	주문상품일련번호	식별 체계가 아닌 단순 순번
수/수량/건수	세는 대상에 따라 구별: 수(일반), 수량(물량), 건수(사건·거래)	가입자수, 주문수량, 민원건수	셋을 임의 혼용하지 않도록 사용 기준을 사전에 명시
금액	화폐 단위의 값	주문금액	단가·수수료 등 하위 분류어를 둘지 전사 결정
율	비율 값. 저장 단위(% 단위인지 소수인지)를 도메인에 명시	이자율	'비율', '요율'은 동의어 정리 대상
일자/일시	날짜만 필요하면 일자, 시각까지 필요하면 일시. 발생 시점의 정밀도 요건으로 판단	가입일자, 최종접속일시	"나중에 시간이 필요할지도 몰라서 일시"는 금지 — 요건이 확정한다
여부	참/거짓 2치의 상태	승인여부	상태가 3개 이상이면 여부가 아니라 상태코드

- **[필수]** 참/거짓 외의 제3의 상태가 존재할 가능성이 있는 항목은 '여부'가 아닌 '상태코드'로 설계한다. 여부 컬럼에 'Y', 'N' 외의 값을 무리하게 담는 순간 표준은 붕괴한다. 업무 분석 단계에서 "승인/미승인 둘뿐인가, 아니면 승인대기가 있는가"를 반드시 확인한다.
- **[필수]** 부정형 명칭을 금지한다. '미승인여부'는 값 'Y'가 "승인 안 됨"을 의미하게 되어 이중 부정 해석을 강요한다. 항상 긍정형('승인여부')으로 정의한다.

5.3.3 용어 작성 원칙

- **[필수]** 동일한 의미의 데이터에는 시스템·부서를 불문하고 동일한 용어를 사용하고, 동일한 용어에는 항상 동일한 도메인을 적용한다.
- **[필수]** 용어의 영문약어명은 구성 단어의 영문약어를 등재 순서대로 언더스코어(`_`)로 연결하여 **기계적으로 생성**한다. 예: 고객+전화+번호 → `CUST_TEL_NO`. 수작업 조정을 허용하면 파싱 가역성(물리명에서 구성 단어를 복원할 수 있는 성질)이 깨진다.
- **[필수]** 신규 용어 등록 전에 다음 순서로 재사용 가능성을 검토한다: ① 동일 용어 존재 확인 → ② 동의어 관계 확인(구성 단어 중 동의어가 섞였는지) → ③ 유사 용어와의 의미 비교. ①~②에서 발견되면 기존 용어를 사용하고, ③에서 의미가 실제로 다르면 구별 수식어를 붙여 신규 등록한다.

- **[필수]** 용어 설명은 다음 네 요소가 드러나도록 작성한다: (a) 무엇을 담는가, (b) 값이 정해지는 업무 규칙·기준 시점, (c) 값의 형식·단위, (d) 유사 용어와의 구별점. "고객이 가입한 일자"처럼 명칭을 되풀이하는 설명은 반려한다. 정의는 명칭이 아닌 개념을 기술해야 하며, 단일 의미로 완결되어야 한다는 원칙은 데이터 정의 작성의 국제 표준 규칙이기도 하다 (ISO/IEC 2023년e). 좋은 설명과 나쁜 설명의 대비는 표 5.7 와 같다.

표 5.7. 용어 설명의 품질 대비

품질	용어 설명 예시
나쁨	계약체결일자: 계약이 체결된 일자
좋음	계약체결일자: 청약 승낙으로 계약의 효력이 발생한 일자(YYYYMMDD). 청약일자와 구별되며, 승낙 통지가 도달한 날을 기준으로 한다. 취소 후 재체결 시 최종 체결일로 갱신하지 않고 이력 엔터티에 보존한다.

5.3.4 관리항목

표준용어 정의서는 최소한 다음 항목을 관리한다(표 5.8).

표 5.8. 표준용어 정의서 관리항목

관리항목	설명	필수 여부
용어 한글명	표준단어 조합의 한글 용어명(논리 속성명)	필수
용어 영문약어명	단어 영문약어의 `_` 연결(물리 컬럼명)	필수
용어 설명	의미·결정 규칙·형식·구별점을 포함한 정의	필수
구성 단어 목록	구성 표준단어의 순서 목록(파싱 결과)	필수
적용 도메인명	분류어에 따라 결정된 표준도메인	필수
참조 표준코드	코드성 용어인 경우 참조 코드 체계	조건부 필수
대체 용어	폐기 시 대체 용어(용어 이관 경로)	조건부 필수
제정·개정 이력	제정일, 변경일, 변경 사유, 승인자	필수

5.4 표준도메인 작성 가이드

표준도메인(Standard Domain)은 데이터 값이 공통으로 갖는 **데이터 타입, 길이, 저장 형식, 값의 허용 범위**를 정의한 것이다. 도메인의 존재 이유는 단 하나의 문장으로 요약된다 — **"같은 성격의 데이터는 어디에 있든 같은 형식이어야 한다."** 고객번호가 A 테이블에서 `VARCHAR(10)`, B 테이블에서 `NUMBER(10)`이면 두 테이블의 조인마다 묵시적 형변환이 발생하고, 형변환은 인덱스를 무력화하며, 값 '0012345678'과 12345678의 불일치 같은 데이터 정합성 사고를 부른다. 도메인은 이 계열의 사고 전체를 원천 차단하는 장치다.

5.4.1 도메인 체계의 구조

- **[필수]** 도메인은 **도메인그룹(분류어 단위) → 도메인분류(그룹을 세분한 중간 구분) → 도메인(타입·길이 단위)**의 3계층으로 관리하며, 이렇게 확정된 도메인을 용어에 적용한다. 도메인분류 계층의 정의 기준과 예시는 5.4.4절 에서 다룬다.

```
도메인그룹: 금액                                  -- 화폐의 가치를 수로 표현
 ├ 도메인분류: 가격
 │   ├ 도메인: 가격N10      NUMERIC(10)           -- 원 단위 정수 가격
 │   └ 도메인: 가격N9,2     NUMERIC(9,2)          -- 소수 둘째 자리까지의 단가
 └ 도메인분류: 금액
     ├ 도메인: 금액N15      NUMERIC(15)           -- 원화 일반 금액
     └ 도메인: 금액N22,2    NUMERIC(22,2)         -- 대규모·소수 포함 금액

적용: 주문금액, 결제금액, 환불금액, ... (분류어가 '금액'인 용어가 공유)
```

- **[필수]** 표준도메인명은 **"도메인분류명 + 데이터 타입 약어 + 길이"**로 짓는다(`C`=CHAR, `V`=VARCHAR, `N`=NUMERIC, 소수가 있으면 `가격N9,2`처럼 소수 자릿수를 덧붙인다). 이름만으로 소속 분류와 물리 형식을 읽을 수 있어야 한다.
- **[필수]** 용어에 적용되는 타입·길이는 도메인을 통해서만 지정한다. 개별 컬럼에서의 임의 타입·길이 지정을 금지한다. 도메인을 우회한 순간 "같은 성격 = 같은 형식" 보장이 사라진다.
- **[권장]** 도메인의 개수는 의식적으로 억제한다. 길이가 조금 다르다는 이유로 `명_V50`, `명_V60`, `명_V80`을 남발하면 도메인은 사실상 컬럼별 정의로 퇴화한다. 명칭류는 50/100/300, 내용류는 1000/4000처럼 **대표 길이 몇 단계로 수렴**시키고, 새 길이 요구는 기존 단계로의 흡수를 먼저 검토한다. 저장 공간 절약을 이유로 길이를 세분화하는 것은 가변 길이 타입(VARCHAR)에서는 실익이 거의 없다.

5.4.2 타입 선택의 판정 기준

숫자 타입 vs 문자 타입 — 다음 질문 순서로 판정한다.

1. **산술 연산의 대상인가?** 합계·평균 등 연산을 하는 값(금액, 수량, 율)은 숫자 타입이다.
2. **선행 0(leading zero)이 유의미한가?** 계좌번호 '0012345', 우편번호 '06236'처럼 앞자리 0이 의미를 갖는 값을 숫자 타입에 넣으면 0이 소실된다. 문자 타입이어야 한다.
3. **자릿수·체계가 변할 수 있는가?** 전화번호, 사업자번호처럼 제도 변경으로 체계가 바뀔 수 있는 식별 값은 문자 타입이 안전하다.
4. **대소 비교가 산술적인가 사전식인가?** 문자 '9' 〉 '10'(사전식)의 오류를 피해야 하는 순번은 숫자 타입이 낫다.

요컨대 **"계산하면 숫자, 식별하면 문자"**가 제1원칙이며, 번호·코드류는 겉보기에 숫자여도 문자 타입이다.

일자·일시의 저장 방식 — 조직마다 갈리는 대표적 쟁점이므로 장단점을 명시하고 전사 단일 방식을 확정한다(표 5.9).

표 5.9. 일자 저장 방식의 비교

방식	장점	단점
`CHAR(8)` 'YYYYMMDD'	DBMS 독립적, 인터페이스·화면 친화적, 부분 검색(년월 LIKE) 용이	무결성 미보장('20260231' 저장 가능), 날짜 연산에 변환 필요, 시간대 개념 부재
`DATE` / `TIMESTAMP`	DBMS가 무결성 보장, 날짜 연산·함수 직접 사용, 시간대 처리 가능	이기종 연계 시 형식 변환 필요, 화면 표시 형식과 저장 형식 분리 필요

- **[필수]** 어느 방식을 채택하든 **전사 단일 방식**으로 통일하고, 혼용을 금지한다. 절반의 테이블이 CHAR(8), 절반이 DATE인 상태가 최악이다.

- **[권장]** 문자형(CHAR(8))을 채택한 경우 유효 일자 검증을 애플리케이션 공통 모듈 또는 CHECK 제약으로 보완하고, '99991231'(무한 종료일자) 등의 특수값 사용 규칙을 도메인 정의에 명시한다.
- **[권장]** 일시를 문자형(CHAR(14), 'YYYYMMDDHH24MISS')으로 저장하는 방식은 밀리초 요건·시간대 요건이 없는 경우에만 허용한다. 글로벌 서비스 등 시간대가 개입하면 TIMESTAMP(및 UTC 저장 원칙)를 채택한다.

금액·율의 정밀도

- **[필수]** 금액·율에 부동소수점 타입(FLOAT, REAL, DOUBLE)의 사용을 금지한다. 이진 부동소수점은 0.1조차 정확히 표현하지 못하므로 금융 계산에서 원 단위 오차를 만든다. 반드시 고정소수점(NUMBER, NUMERIC, DECIMAL)을 사용한다.
- **[필수]** 율 도메인은 저장 단위를 명시한다. 3.5%를 3.5로 저장하는지(백분율 단위) 0.035로 저장하는지(소수 단위)를 도메인 정의에 확정하지 않으면, 두 방식이 섞여 백 배 오차 사고가 발생한다. 권장은 백분율 단위 저장 + `NUMBER(18,6)`이며, 어느 쪽이든 전사 통일이 핵심이다.
- **[권장]** 원화 금액은 `NUMBER(15)`(천조 단위까지), 외화 병행 시 `NUMBER(18,4)` 수준을 기본으로 하되, 업권 규정(보험 책임준비금 등 대규모 집계)이 있으면 그 요건으로 상향한다.

5.4.3 관리항목

표준도메인 정의서는 최소한 다음 항목을 관리한다(표 5.10). 항목별 작성 요령은 테크니컬 노트(6.3.3절)에서 다룬다.

표 5.10. 표준도메인 정의서 관리항목

관리항목	설명	필수 여부
표준도메인명	"도메인분류명 + 데이터 타입 약어 + 길이" 형태의 식별 명칭 (예: `금액N15`)	필수
도메인그룹명	도메인이 소속되는 최상위 범주	필수
도메인분류명	그룹을 세분한 중간 구분. 도메인명의 어간이 된다	필수
데이터 유형	논리 데이터 타입(CHAR, VARCHAR, NUMERIC, DATETIME)	필수
길이	전체 자릿수(소수 자리 포함). 한글은 2바이트 기준	조건부 필수
소수 자릿수	소수점 이하 최대 자릿수(숫자형에 소수가 있을 때)	조건부 필수
저장 형식	DB에 저장되는 형식(`YYYYMM`, `9999999999999999` 등)	필수
표현 형식	화면·보고서로 제공할 때의 형식(`YYYY-MM` 등)	권장
단위	값의 측정 단위(원, 연월, m 등)	조건부 필수
허용 값 범위	유효값 목록 또는 최소~최대 범위	조건부 필수
물리 데이터 타입	DBMS가 인식하는 실제 타입·길이(`NUMERIC(15)` 등)	필수
상태	후보 / 확정 / 폐기	필수

- **[권장]** 조직의 필요에 따라 기본값, 보안 속성(암호화·마스킹 대상 여부와 방식), 제정·개정 이력을 확장 항목으로 추가한다. 표준 체계의 필수 항목은 아니지만 운영 단계에서 유용하다.

5.4.4 표준 도메인 분류 체계

도메인 체계의 구조(5.4.1절)에서 밝힌 대로, 표준도메인은 **도메인그룹 → 도메인분류 → 도메인**의 3계층 분류 체계로 관리된다. 이 절에서는 상위 두 계층인 도메인그룹과 도메인분류의 정의 기준을 구체화한다.

도메인그룹은 성격이 유사한 도메인을 묶은 최상위 범주로, 금액·날짜/시간·명칭·번호·코드처럼 소수의 큰 구분으로 유지한다. 도메인분류는 그룹을 한 단계 세분한 중간 계층으로, 금액 그룹의 가격·비용, 날짜/시간 그룹의 연월·연월일처럼 개별 도메인이 소속될 세부 구분을 정의한다. 표준도메인명이 "도메인분류명 + 데이터 타입 + 길이"(예: `가격N10`) 형태이므로, 도메인분류명은 곧 도메인명의 어간이 된다.

「표준도메인 정의서」는 이 두 계층을 각각 **도메인 그룹** 시트와 **도메인 분류** 시트로 분리해 관리하며(3.5.2.1절), 각 분류는 반드시 하나의 상위 그룹에 소속된다. 공공 부문은 행정안전부「공통표준도메인그룹명」「공통표준도메인분류명」 체계를 준용한다. 그룹·분류의 정의 방법과 자동 검증은 테크니컬 노트에서 상세히 다룬다(6.3.3.1절, 6.3.3.2절).

5.4.4.1 표준 도메인그룹

KDSM 표준도메인 정의서가 관리하는 도메인그룹은 표 5.11 의 여덟 가지다.

표 5.11. 표준 도메인그룹 체계

도메인그룹	설명	대표 도메인분류	대표 도메인
금액	화폐의 가치를 수로 표현	가격, 금액, 비용, 요금	`금액N15` NUMERIC(15)
날짜/시간	사건의 시점 또는 시점 간의 기간을 표현	연도, 연월, 연월일, 시분초	`연월일C8` CHAR(8)
내용	자유 형식의 서술 텍스트를 표현	내용	`내용V4000` VARCHAR(4000)
명칭	문자로 객체를 식별(명·주소·유형 포함)	명, 주소, 상세주소, 유형	`명V100` VARCHAR(100)
번호	체계나 자릿값의 의미를 갖는 식별 번호	번호, 일련번호, 우편번호, 계좌번호	`우편번호C5` CHAR(5)
수량	객체의 개수·양과 측량 값을 표현	수, 순서, 면적, 위도, 경도	`수N10` NUMERIC(10)
율	다른 수나 양에 대한 비를 백분율·소수로 표현	율	`율N7,2` NUMERIC(7,2)
코드	분류·구분값 또는 Y/N, 유/무를 표현	코드, 여부, 유무	`여부C1` CHAR(1)

대표 도메인의 타입·길이는 예시이며, 실제 값은 채택 DBMS와 업무 요건에 맞추어 확정한다. 중요한 것은 개별 값이 아니라 **"모든 컬럼이 여덟 그룹 중 하나에 소속된다"는 규율**이다. 공공 부문의 공통표준도메인도 도메인그룹명–도메인분류명–도메인명–타입·길이–저장형식의 동일한 계층 구조로 보급된다 (행정안전부 2025년b; 한국데이터산업진흥원 2006년).

- **[필수]** 그룹은 함부로 늘리지 않는다. 새 그룹을 만들기 전에 기존 여덟 그룹으로 수용 가능한지 먼저 검토한다. 그룹이 늘어나는 만큼 상위 구분으로서의 변별력은 떨어진다.

- **[필수]** 아이디는 명칭 그룹에, 여부는 코드 그룹에 소속시킨다. 겉보기 형태가 아니라 **표현하려는 의미**를 기준으로 그룹을 정한다.
- **[권장]** 개인정보·민감정보를 담는 도메인(주민등록번호, 카드번호 등)은 도메인 정의에 **보안 속성**(암호화 대상 여부, 마스킹 규칙)을 함께 관리한다. 암호화 적용 시 암호문 길이가 평문보다 길어지므로, 암호화 대상 도메인의 물리 길이는 암호화 방식의 출력 길이를 기준으로 산정한다.

5.4.4.2 표준 도메인분류 예시

도메인분류와 소속 도메인그룹의 예시는 표 5.12 와 같다.

표 5.12. 도메인분류 예시

도메인분류명	도메인분류 설명	소속 도메인그룹
가격	물건이 지니고 있는 가치를 돈으로 나타낸 것	금액
요금	사물의 사용·소비·관람의 대가로 치르는 돈	금액
연월	특정한 연(年)과 월(01~12)을 정하여 표현한 것	날짜/시간
연월일	특정한 날짜를 연·월·일로 정하여 표현한 것	날짜/시간
내용	자유 형식의 서술 텍스트로 표현한 것	내용
명	사물·단체·현상을 다른 것과 구별하기 위해 붙인 이름	명칭
주소	사람이나 기관이 자리 잡은 곳을 행정 구역으로 나타낸 이름	명칭
우편번호	우편물 분류를 위해 지역마다 매긴 번호	번호
일련번호	일률적으로 연속되어 있는 번호	번호
수	셀 수 있는 사물을 세어서 나타낸 값	수량
율	다른 수나 양에 대한 어떤 수나 양의 비	율
여부	그러함과 그러하지 아니함을 구분하는 코드	코드

- **[필수]** 모든 분류는 반드시 하나의 상위 도메인그룹에 소속시킨다. 소속 그룹이 없는 분류는 존재할 수 없다.
- **[권장]** 같은 분류에 속한 도메인은 데이터 유형이 일관되도록 설계한다. 가격·금액·비용·요금은 모두 금액 그룹이므로 고정소수점 숫자형으로 수렴하고, 연월·연월일은 날짜/시간 그룹의 저장 방식(표 5.9)을 따른다.
- **[권장]** 분류는 필요한 만큼만 세분한다. 번호 그룹처럼 법정 번호(주민등록번호·사업자등록번호 등)가 각각 자릿수와 검증 규칙을 갖는 경우에는 분류를 개별로 두고, 그 밖의 일반 번호는 '번호' 하나로 수용한다.

5.5 표준코드 작성 가이드

표준코드(Standard Code)는 성별·계약상태·처리결과처럼 **한정된 값의 집합으로 관리되는 데이터**의 허용값과 그 의미를 표준화한 것이다. 코드가 없으면 같은 "정상" 상태가 어떤 시스템에서는 `'1'`, 다른 곳에서는 `'Y'`, 또 다른 곳에서는 `'정상'`으로 저장되어 값의 해석과 시스템 간 연계가 매번 번역을 요구한다. 표준단어·용어가 데이터의 **이름**을 표준화한다면, 표준코드는 데이터의 **값**을 표준화한다. 표준코드는 이 번역 비용을 제거하고 허용값을 스키마·검증 수준에서 강제하여 오염값의 발생 자체를 차단하는 **값 영역(value domain)의 표준**이다 (DAMA International 2017년; 행정안전부 2025년b).

5.5.1 코드의 구조

- **[필수]** 표준코드는 코드 자체의 속성을 정의하는 **코드 마스터**와, 그 코드에 속한 개별 값을 정의하는 **코드값 상세**의 두 계층으로 관리한다. 둘은 「**코드 ID**」로 연결되며, 마스터 1건에 상세 N건이 딸리는 1:N 관계다. 두 계층의 코드 ID는 정확히 일치해야 하며, 양식 항목과 작성 요령은 테크니컬 노트(6.3.4절)를 참조한다.

```
코드 ID: CUST_GRD_CD   (코드 마스터: 고객등급코드)
 ├ 코드값 상세: '10' = 일반
 ├ 코드값 상세: '20' = 실버
 ├ 코드값 상세: '30' = 골드
 └ 코드값 상세: '40' = VIP
```

- **[필수]** 분류어가 '코드'인 용어(코드성 용어)는 반드시 하나의 표준코드에 연결한다. 자유 입력 문자열에 '코드' 분류어를 붙이지 않는다(5.3.2절). 코드 컬럼에는 코드 도메인그룹을 적용하고, 허용값은 연결된 표준코드의 코드값으로 제한한다(표 5.11).
- **[권장]** 코드값의 물리 저장은 문자 타입을 기본으로 한다. 코드값은 산술 연산의 대상이 아니며, 선행 0·자릿수 체계 변경 가능성이 있으므로 숫자 타입에 담지 않는다(5.4.2절).

5.5.2 코드값 부여 방식

코드값을 "의미 있게" 부여할 것인가 "순차적으로" 부여할 것인가는 코드 설계의 첫 갈림길이다. 장단점을 이해하고 대상별로 선택한다(표 5.13).

표 5.13. 코드값 부여 방식의 비교

방식	정의	장점	단점
무의미(순차) 코드	`'10'`, `'20'`, `'30'`처럼 의미 없이 부여	값 추가·재분류에 안정적, 의미 변경의 영향이 없음	값만으로 의미 파악 불가
유의미(연상) 코드	`'M'`=남성, `'F'`=여성처럼 의미를 연상	코드값 자체로 가독	의미 변경 시 코드 붕괴, 국제화·확장에 취약

- **[권장]** 원칙적으로 **무의미(순차) 코드를 기본**으로 한다. 유의미 코드는 값의 의미가 굳어져, 나중에 분류 체계가 바뀌면 코드값을 재해석하거나 이관해야 하는 부채가 된다. 성별처럼 불변에 가깝고 국제 관행이 확립된 극소수에 한해 유의미 코드를 허용한다.
- **[권장]** 코드값의 자릿수는 예상 최대 값 수에 한 단계 여유를 두어 설계한다(예: 2자리 → 최대 100개). 자릿수가 부족해 전 값을 재부여하는 것은 재앙이다.
- **[권장]** 표시 순서가 코드값의 크기와 무관하다면 별도의 **정렬순서** 항목으로 관리하고, 코드값 자체에 정렬 의도를 담지 않는다.

5.5.3 코드 설계 원칙

- **[필수]** 하나의 코드 체계 안에서 코드값은 **상호 배타적이고 전체를 포괄**해야 한다(MECE). 어떤 실제 값이 두 코드에 동시에 해당하거나 어느 코드에도 속하지 못하면 설계가 잘못된 것이다. 포괄을 위해 '기타(90)' 코드를 둘 수 있으나, '기타'가 과도하게 쌓이면 분류 체계를 재검토한다.
- **[필수]** 한 번 부여·운영된 코드값의 의미를 **재사용(변경)하지 않는다**. `'30' = 골드`였던 값을 나중에 `'30' = 해지`로 바꾸면 과거 데이터의 `'30'`이 소급하여 오염된다. 더 이상 쓰지 않는 코드는 삭제가 아니라 **사용여부 = 미사용**으로 폐기하고 값은 보존한다.

```
-- 나쁨: 운영 중인 코드값의 의미를 변경 → 과거 데이터가 소급 오염
'30' = 골드   →(변경)   '30' = 해지

-- 좋음: 기존 값은 사용여부로 폐기하고, 새 의미는 새 값으로 부여
'30' = 골드   (사용여부 = 미사용)
'99' = 해지   (신규)
```

- **[필수]** 참/거짓 2치로 충분한 상태는 코드가 아니라 여부 도메인으로 설계하고, 3치 이상으로 확장될 수 있는 상태만 상태코드로 설계한다(5.3.2절). 부정형 코드명('미승인')을 금지하고 긍정형으로 정의한다.
- **[권장]** 계층(대·중·소분류)이 필요한 코드는 코드값에 계층을 인코딩(`'01'`, `'0101'`)하기보다 **부모 코드 참조**로 계층을 표현하는 방식을 우선 검토한다. 값에 계층을 직접 담으면 재편성 시 전체 값 체계가 흔들린다.

5.5.4 공통코드와 개별코드

- **[필수]** 여러 업무·시스템이 공유하는 코드(성별·국가·통화·행정구역 등)는 **공통코드**로 단일하게 관리하고, 특정 업무에만 쓰이는 코드는 **개별코드**로 둔다. 공통코드를 업무별로 복제하면 값 불일치가 재발한다.
- **[필수]** 공공 부문이거나 공공기관과 연계하는 경우, 국가·행정구역·산업분류 등은 **법정·국가 표준 코드체계를 준용**하고 자체 코드를 신설하지 않는다. 공통표준의 준용 원칙은 공통표준단어 준용(5.2.1절)과 동일한 근거를 따른다 (행정안전부 2025년b, 2025년a).
- **[권장]** 외부 표준 코드체계를 준용하되 내부 코드값·형식이 다르면, 원본은 규격대로 저장하고 내부 표준값으로의 **매핑**을 관리한다(5.1.2절).

5.5.5 관리항목

표준코드 정의서는 코드 마스터와 코드값 상세로 나누어 최소한 다음 항목을 관리한다(표 5.14).

표 5.14. 표준코드 정의서 관리항목

구분	관리항목	설명	필수 여부
마스터	코드 ID	코드 체계의 식별자(코드성 용어의 영문약어명과 정합)	필수
마스터	코드명	코드 체계의 한글 명칭	필수
마스터	코드 설명	관리하는 값 집합의 의미·분류 기준	필수
마스터	공통/개별 구분	공통코드 여부	필수
상세	코드값	개별 허용값	필수
상세	코드값명	코드값의 한글 명칭	필수
상세	코드값 설명	값의 의미·적용 기준	권장
상세	정렬순서	표시 순서	권장
상세	사용여부	사용/미사용(폐기) 구분	필수
공통	제정·개정 이력	제정일, 변경일, 변경 사유, 승인자	필수

5.5.6 코드 품질 체크리스트

코드 정의서의 배포 전, 그리고 정기 점검 시 다음을 기계적으로 검증한다.

- 코드 마스터의 코드 ID와 코드값 상세의 코드 ID가 모두 일치하는가(고아 상세·빈 마스터 없음)
- 한 코드 체계 안에서 코드값이 중복되지 않는가
- 코드성 용어(분류어 '코드')가 모두 유효한 코드 ID에 연결되어 있는가
- 코드값이 상호 배타·전수 포괄(MECE)을 만족하는가
- 폐기된 코드값이 삭제가 아니라 사용여부=미사용으로 관리되는가
- 코드값명이 비어 있거나 코드값을 단순 반복하지 않는가

5.6 명명규칙

명명규칙은 논리 모델(엔터티·속성)과 물리 모델(테이블·컬럼)의 명칭을 일관되게 부여하기 위한 규칙이다. 좋은 이름의 판정 기준은 하나다 — **모델을 처음 보는 사람이 이름만으로 그 데이터의 의미와 형식을 유추할 수 있는가.** 명명은 자의가 아니라 국제 표준(ISO/IEC 11179-5)이 규정하는 객체어(주제어)·특성어(수식어)·표현어(분류어)의 조합 구조를 따르며 (ISO/IEC 2015년a), 본 절의 규칙도 이 골격 위에 있다.

공통 원칙은 다음과 같다.

- **[필수]** 논리명은 표준용어 사전의 한글명을, 물리명은 표준단어 사전의 영문약어 조합을 사용하며, 논리명과 물리명은 1:1로 대응한다. 논리명이 같은데 물리명이 다르거나 그 역인 경우는 존재할 수 없다.
- **[필수]** DBMS 예약어(ORDER, USER, DATE, LEVEL 등)를 물리명으로 사용하지 않는다. 예약어 충돌은 따옴표 인용으로 회피할 수 있으나, 인용 부호 의존 코드는 이식성과 가독성을 해치므로 명칭 자체를 바꾼다 (Celko 2005년). (ORDER → ORD)
- **[필수]** 물리명은 영문 대문자, 숫자, 언더스코어(`_`)만 사용하고 영문자로 시작한다. 공백·하이픈·한글 물리명을 금지한다.
- **[권장]** **길이 예산**을 의식한다. 물리명의 상한(채택 DBMS 기준)에서 접두어·접미어가 차지할 자리를 먼저 빼고, 남는 예산 안에서 단어 수를 설계한다. 예: 상한 30바이트, 이력 접미어 `_HST`(4자) 예약 시 본체는 26바이트 이내.
- **[필수]** 대소문자 폴딩(folding)이 DBMS마다 다름을 유의한다. 따옴표로 감싸지 않은 식별자를 Oracle은 대문자로, PostgreSQL은 소문자로 접는다. 물리명을 대문자로 통일하고 따옴표 인용을 쓰지 않으면 두 DBMS에서 동일하게 동작하므로, 인용 식별자("Cust")를 만들지 않는다(6.4.2.2절).

> **i 노트**
>
> 채택 DBMS의 식별자 최대 길이는 서로 다르다 — Oracle 30바이트(12c R2부터 128바이트), PostgreSQL 63바이트. 두 DBMS를 함께 지원한다면 더 짧은 상한을 공통 기준으로 삼아 설계한다. DBMS별 길이·인코딩(바이트 vs 문자) 규칙은 표 6.8 에 정리되어 있다.

5.6.1 주제영역

- **[필수]** 주제영역명은 표준용어 사전에 등록된 한글 용어를 사용하고, 하나의 논리 데이터 모델 안에서 유일해야 한다.
- **[필수]** '~영역', '~부문', '~업무', '~시스템'처럼 구획임을 되풀이하는 접미어를 붙이지 않는다. '영업영역'이 '영업'보다 알려 주는 것은 없다.

- **[권장]** 주제영역명은 관리 대상을 가리키는 짧은 명사로 부여한다(영업·상품·고객). 조직명이나 프로젝트명을 쓰지 않는 것은, 그런 이름이 조직 개편과 함께 낡기 때문이다.
- **[권장]** 물리 모델에서 주제영역을 스키마나 접두어로 구현한다면, 주제영역명을 구성하는 표준단어의 영문약어를 그대로 사용하여 논리명과의 대응을 유지한다(상품 → `PRD`).

5.6.2 엔터티

- **[필수]** 엔터티명은 업무에서 사용하는 **단수형 명사**로 부여한다. 엔터티는 인스턴스의 집합이지만 이름은 개별 인스턴스의 유형을 가리키므로 단수가 논리적이다. (‘고객들’ × → ‘고객’)
- **[필수]** 엔터티명은 전사 데이터 모델 내에서 유일해야 한다. 서로 다른 주제영역이라도 동일 명칭을 허용하면 통합 관점의 담론이 불가능해진다.
- **[필수]** ‘~정보’, ‘~데이터’, ‘~테이블’, ‘~파일’, ‘~관리’처럼 의미를 더하지 않는 접미어를 금지한다. ‘고객정보’가 ‘고객’보다 알려 주는 것은 없다.
- **[권장]** 엔터티의 성격에 따라 다음 유형어 체계(표 5.15)를 마지막 단어로 사용한다. 유형어는 엔터티의 데이터 발생 패턴을 이름에 각인시켜, 모델을 읽는 사람이 즉시 성격을 파악하게 한다.

표 5.15. 엔터티 유형어 체계

유형어	엔터티 성격	발생 패턴	예시
(없음)	키(기본) 엔터티: 업무의 주체·대상	업무 개시 시 등록, 저빈도 변경	고객, 상품, 조직
내역	행위(트랜잭션) 엔터티: 업무 행위의 발생 기록	행위마다 1건 발생, 불변 지향	주문내역, 입금내역
이력	변경 전 상태의 시점별 보존	원본 변경 시 스냅숏 적재	고객등급변경이력
상세	상위에 종속되는 반복 상세(1:N의 N)	상위 1건에 N건	주문상품상세
집계	원천을 요약한 파생 데이터	배치 주기로 재생성	일별매출집계
관계	M:N 관계 해소 교차 엔터티	양쪽 참여 시 발생	고객상품관계

- **[권장]** 이력 엔터티의 명명은 “무엇의 변경을 보존하는가”가 드러나게 한다 (Kimball 와/과 Ross 2013년a). 원본 엔터티명 + 변경 대상 + ‘이력’의 구조(고객주소변경이력)를 기본으로 하고, 엔터티 전체 스냅숏이면 원본명+‘이력’(고객이력)으로 한다.
- **[권장]** 서브타입(배타적 하위 유형)의 명칭은 슈퍼타입명을 포함한다. 슈퍼타입 ‘고객’의 서브타입은 ‘개인고객’, ‘법인고객’으로 하여 계보가 이름에 드러나게 한다.
- **[권장]** M:N 관계를 해소한 교차 엔터티는 양쪽 엔터티명의 결합(고객상품관계)을 기본으로 하되, 그 관계가 업무적으로 고유한 이름을 갖는다면(예: 고객과 상품의 M:N = ‘가입’) 업무 명칭을 우선한다. 업무 명칭이 있는 교차 엔터티는 대개 그 자체가 독립적 업무 개념이다.
- **[권장]** 논리명(엔터티명)은 약어 없이 완전한 한글 용어로 쓰고, 약어화는 물리명(테이블명) 생성 단계에서만 적용한다(5.6.4절). 논리 모델 단계의 성급한 약어는 커뮤니케이션 비용만 키운다.

엔터티 명명에서 자주 나타나는 오류와 교정은 표 5.16 와 같다.

표 5.16. 엔터티 명명 대비 예시

유형	나쁨(Bad)	좋음(Good)	근거
복수·군집 표현	고객들, 고객목록	고객	이름은 인스턴스 유형(단수)을 가리킴
무의미 접미어	고객정보, 고객테이블	고객	의미를 더하지 않는 접미어 금지
유형어 누락	주문(행위 데이터)	주문내역	발생 패턴을 유형어로 각인
서브타입 계보 단절	개인, 법인	개인고객, 법인고객	슈퍼타입명 포함

5.6.3 속성

- **[필수]** 속성명은 표준용어 사전에 등록된 **용어 한글명**을 그대로 사용한다. 모델링 도구에서 임의 명칭 입력을 허용하지 말고 사전 연동 입력을 강제하는 것이 이상적이다.
- **[필수]** 속성명은 분류어로 끝난다. 이는 모든 속성이 어떤 도메인에 소속됨을 뜻한다.
- **[필수]** 하나의 엔터티 내에서 속성명은 유일해야 한다.
- **[권장]** 자신이 속한 엔터티명을 속성명 앞에 반복하지 않는다. '고객' 엔터티의 속성이므로 '고객명'이 아니라 '명'으로 충분하다는 판단은 성급하다 — 여기에는 중요한 예외가 있다. **식별자와 명칭 속성은 엔터티명을 포함하는 것이 실무적으로 유리하다.** '고객' 엔터티의 식별자를 '번호'로만 정의하면, 이 속성이 다른 엔터티로 상속(FK)되는 순간 '번호'가 무엇의 번호인지 알 수 없게 된다. 식별자·대표 명칭('고객번호', '고객명')은 엔터티명을 포함하고, 그 외 일반 속성('가입일자', '등급코드')은 반복하지 않는 것을 규칙으로 한다.
- **[필수]** 관계로 상속된 외래 식별자는 원천 엔터티에서의 명칭을 유지한다. '주문' 엔터티에 상속된 고객의 식별자는 원천과 동일하게 '고객번호'다. 명칭이 유지되어야 조인 경로를 이름만으로 추적할 수 있다.
- **[필수]** 동일 원천이 서로 다른 **역할**로 두 번 이상 상속되면 역할 수식어를 붙여 구별한다. (주문 엔터티의 '주문고객번호'와 '수취고객번호', 배송의 '출발지우편번호'와 '도착지우편번호') 자기참조 관계(조직의 상위 조직)도 같은 원리로 '상위조직번호'와 같이 역할을 명시한다.
- **[권장]** 속성의 배치 순서는 ① 주식별자 → ② 외래 식별자 → ③ 업무 속성(업무 흐름 순) → ④ 시스템 관리 속성으로 통일한다. 순서의 일관성은 모델 가독성에 직접 기여한다.
- **[필수]** 모든 속성은 표준 도메인 하나에 소속되며, 분류어가 그 도메인을 결정한다(5.4절). '…금액'은 금액 도메인을,'…여부'는 여부 도메인을, '…율'은 비율 도메인을 상속한다. 분류어와 값 형식의 불일치('…율'인데 정수 도메인)는 표준 검증에서 오류로 잡는다.
- **[필수]** 참·거짓 두 값만 갖는 속성은 분류어 '여부'로 끝맺고, 임의의 상태 플래그(`flag`, `is_…`)를 만들지 않는다. 논리명 '삭제여부'는 물리명 `DEL_YN`(5.6.5절)으로 대응한다. 세 가지 이상의 상태가 필요하면 여부가 아니라 코드(분류어 '코드')로 설계한다 — 여부의 남용은 뒤늦게 상태가 늘어날 때 스키마 변경을 부른다.

5.6.4 테이블

- **[필수]** 테이블명은 대응 엔터티명을 구성하는 표준단어의 영문약어를 언더스코어로 연결하여 생성한다. (고객주소변경이력 → `CUST_ADDR_CHG_HST`) 엔터티-테이블이 1:1이 아닌 경우(수직 분할 등)에도 파생 테이블명은 원 엔터티의 물리명을 어간으로 유지한다.
- **[필수]** 테이블명은 스키마 내에서 유일해야 하며 채택 DBMS의 식별자 길이 제한을 초과할 수 없다.

- **[권장]** 시스템·용도 구분이 필요한 경우 접두어/접미어 체계를 표준화하여 사용하고, 본체 명칭과 수식 요소를 혼동하지 않게 한다.

표 5.17. 테이블 접두·접미어 체계

구분	표기	예시	비고
임시 테이블	`TMP_` 접두	`TMP_CUST_MIG`	수명·삭제 규칙을 함께 관리
인터페이스	`IF_` 접두	`IF_ORD_RCV`	수신/송신 구분 포함 가능
집계 테이블	`_AGG` 접미	`DLY_SALE_AGG`	재생성 주기를 주석에 명시
이력 테이블	`_HST` 접미	`CUST_GRD_CHG_HST`	원본 테이블명 어간 유지
뷰	`V_` 접두	`V_CUST_SUMM`	테이블로 오인 방지
머티리얼라이즈드 뷰	`MV_` 접두	`MV_MTHLY_SALE`	갱신 정책 명시
메타테이블	`MT_` 접두	`MT_ENTTY`	메타데이터임을 인식

- **[권장]** 백업·작업용 사본 테이블(`CUST_BAK_20260715` 류)의 운영 스키마 방치를 금지한다. 작업 완료 시점에 삭제하는 것을 원칙으로 하고, 보존이 필요하면 별도 보관 스키마로 이동한다. 방치된 사본은 표준 검증 도구의 소음이자 개인정보 관리의 사각지대다.
- **[권장]** 테이블명을 구성하는 약어의 순서는 대응 엔터티명의 단어 순서를 그대로 유지한다. 임의로 재배열하면 논리명↔물리명의 기계적 상호 복원이 깨진다. (고객주소변경이력 → `CUST_ADDR_CHG_HST`, `ADDR_CUST_CHG_HST` ×)

5.6.5 컬럼

- **[필수]** 컬럼명은 대응 속성(표준용어)의 영문약어명을 그대로 사용하고, 타입·길이는 적용 도메인의 정의를 그대로 따른다. 어떠한 개별 조정도 금지한다.

컬럼의 물리 접미어는 분류어에서 파생하며, 접미어만으로 값의 성격과 도메인이 드러나야 한다. 자주 쓰이는 접미어를 다음과 같이 표준화한다.

표 5.18. 컬럼 분류어별 물리 접미어 표준

성격	물리 접미어	예시	도메인 · 저장 형식
여부(불리언)	`_YN`	`USE_YN`, `DEL_YN`	여부 도메인, `CHAR(1)` `'Y'`/`'N'`
코드	`_CD`	`ORD_STS_CD`	코드 도메인, 코드 마스터 참조
식별자(기술 부여)	`_ID`	`USR_ID`	ID 도메인
식별번호(업무 부여)	`_NO`	`CUST_NO`, `ORD_NO`	번호 도메인
금액	`_AMT`	`ORD_AMT`	금액 도메인, `NUMBER`/`NUMERIC`
수량	`_QTY`	`ORD_QTY`	수량 도메인, 정수형
일자·일시·년월	`_DT`·`_DTM`·`_YM`	`JOIN_DT`, `REG_DTM`	표 5.23 참조

- **[필수]** 동일 용어는 어느 테이블에 있든 동일 컬럼명·동일 도메인이다. 특히 **조인에 사용되는 식별자 컬럼의 명칭·타입 불일치는 절대 허용하지 않는다.** `CUST.CUST_NO VARCHAR(10)`과 `ORD.CUST_NO NUMBER`의 조인은 묵시적 형변환 → 인덱스 무력화 → 전 건 스캔이라는 성능 사고의 전형적 경로다.
- **[필수]** 시스템 관리 공통 컬럼은 전사 표준으로 정의하고 모든 테이블의 마지막에 동일 명칭·동일 순서로 배치한다.

표 5.19. 시스템 관리 공통 컬럼(예시)

논리명	물리명	도메인	용도
최초등록자ID	`FRST_REG_USR_ID`	ID	최초 INSERT 수행 주체
최초등록일시	`FRST_REG_DTM`	일시	최초 INSERT 시점
최종수정자ID	`LAST_UPD_USR_ID`	ID	최종 UPDATE 수행 주체
최종수정일시	`LAST_UPD_DTM`	일시	최종 UPDATE 시점

- **[권장]** 제약조건·인덱스 등 부속 오브젝트의 명명도 표준화한다. 오브젝트명만으로 소속 테이블과 성격이 드러나야 장애 메시지(제약 위반 오류 등)를 읽는 즉시 원인 테이블을 특정할 수 있다.

표 5.20. 부속 오브젝트 명명 규칙

오브젝트	규칙	예시
기본키	`PK_` + 테이블명	`PK_CUST`
외래키	`FK_` + 자식테이블 + `_` + 부모테이블(+순번)	`FK_ORD_CUST`
유니크 인덱스	`UX_` + 테이블명 + 순번	`UX_CUST_01`
일반 인덱스	`IX_` + 테이블명 + 순번	`IX_ORD_03`
체크 제약	`CK_` + 테이블명 + 컬럼명	`CK_CUST_GNDR_CD`

5.7 논리 데이터 모델링 가이드

논리 데이터 모델링은 확정된 데이터 표준(단어·용어·도메인·코드) 위에서 업무 요건을 **엔터티·속성·관계**로 구조화하는 활동이다. 명칭 부여는 명명규칙(5.6절)을, 각 정의서 항목의 작성 요령은 테크니컬 노트(6.4.1절)를 함께 참조한다. 본 절은 "무엇을 엔터티로 삼고, 어떤 속성을 어떻게 배치할 것인가"의 **설계 원칙**을 다루며, 물리 관심사(성능·저장)는 배제하고 업무 규칙의 정확한 표현에 집중한다.



그림 5.1. 논리 데이터 모델링 절차

논리 데이터 모델링은 그림 5.1 와 같이 여섯 단계로 진행된다. 먼저 개념 모델을 검토하여 대상 업무의 범위를 확정하고, 관리 대상이 되는 집합을 엔터티로 도출한다(5.7.3절). 이어 각 엔터티의 특성을 속성으로 정의하고(5.7.4절), 각 인스턴스를 유일하게 식별할 주식별자와 후보키를 결정한다. 그다음 엔터티 사이의 관계(차수·선택성)를 설정해 업무 규칙을 구조로 표현하고, 마지막으로 정규화 수준과 무결성 관점에서 논리 모델을 검증한다. 검증을 통과한 논리 모델은 물리 데이터 모델링(5.8절)의 입력이 된다. 개념 모델 검토에서 확정한 범위는 주제영역으로 구획하여(5.7.2절) 이후 엔터티 도출과 산출물 관리의 단위로 삼는다.

5.7.1 개념 모델 검토

- **[필수]** 모델링에 앞서 대상 업무의 범위를 확정한다. 요구사항분석의 표준화 대상 데이터 정의(3.2.2.2절)를 바탕으로 모델에 포함할 주제영역(subject area)과 경계를 정한다.
- **[필수]** 핵심 관리 대상(주요 엔터티 후보)과 그들 사이의 큰 관계를 개념 수준에서 식별하여 이후 상세 설계의 골격으로 삼는다.
- **[권장]** 전사 데이터 모델이나 기존 표준이 있으면 재사용 대상을 먼저 식별해 중복 설계를 피하고, 신규로 정의할 영역만 상세화 대상으로 좁힌다.

5.7.2 주제영역 정의

주제영역(Subject Area)은 논리 데이터 모델을 업무 단위로 나눈 구획이다. 두꺼운 책을 장(章)으로 나누어야 읽을 수 있듯, 엔터티가 수백 개에 이르는 모델도 구획이 있어야 읽고 나누어 맡을 수 있다. 주제영역은 ERD 분할의 단위이자 담당·검토·변경 영향 범위의 단위가 된다.

- **[필수]** 주제영역은 조직 구조가 아니라 **데이터의 응집도**를 기준으로 나눈다. 같은 주제영역 안의 엔터티끼리는 관계가 조밀하고 주제영역을 넘는 관계는 성긴 상태가 좋은 분할이다. 조직도를 그대로 옮기면 조직 개편이 있을 때마다 모델이 흔들린다.
- **[필수]** 하나의 엔터티는 **하나의 주제영역에만** 배정한다. 여러 영역에서 참조된다는 이유로 중복 배정하면 변경 시 어느 쪽이 기준인지 알 수 없게 된다. 여러 영역이 공통으로 참조하는 엔터티(코드·조직 등)는 공통 성격의 주제영역을 따로 두어 한곳에 모은다.
- **[필수]** 주제영역명은 표준용어로 정의하고 하나의 논리 데이터 모델 안에서 유일해야 한다(5.6.1절).
- **[권장]** 주제영역 하나가 담는 엔터티 수는 한 장의 ERD로 읽을 수 있는 규모(대략 10~20개)를 넘지 않게 한다. 넘어서면 하위 주제영역으로 분할하거나 경계를 다시 본다.

- **[권장]** 주제영역 순번은 업무 흐름의 순서를 따라 1부터 빈틈없이 부여하고, 정의서의 게재 순서와 ERD 분할 순서를 이 순번으로 통일한다.
- **[권장]** 주제영역 설명은 "무엇이 들어오고 무엇이 들어오지 않는가"를 판정할 수 있게 쓴다. 새로 도출한 엔터티의 배정처를 설명만 읽고 정할 수 있어야 구획이 유지된다.
- **[권장]** 주제영역 구획은 물리 모델에서 스키마·테이블 접두어·ERD 분할 등으로 이어받되, 논리 단계의 구획을 임의로 재편하지 않는다(5.8.2절).
- 주제영역과 그 배정 결과는 주제영역 정의서(4.6.2절)와 주제영역구성 정의서(4.6.3절)에 기록하며, 항목별 작성 요령은 테크니컬 노트(6.4.1.1절)를 따른다.

5.7.3 엔터티 정의

- **[필수]** 엔터티는 관리 대상이 되는 명사 후보 중 다음을 **모두** 만족하는 것으로 도출한다: ① 인스턴스가 둘 이상 존재하고, ② 각 인스턴스를 유일하게 식별할 수 있으며, ③ 관리할 속성을 가진다. 한 건만 존재하거나 식별자가 없으면 엔터티가 아니라 다른 엔터티의 속성이다.
- **[필수]** 하나의 엔터티는 **하나의 주제**만 표현한다. 성격이 다른 데이터를 한 엔터티에 섞지 않는다(고객과 주문을 한 집합에 담지 않는다).
- **[필수]** 하나의 속성이 **다중 값**을 갖거나 반복되면 별도 엔터티로 분리한다(고객의 여러 연락처 → '고객연락처' 엔터티). 다중값을 콤마로 이어 한 속성에 담지 않는다.
- **[권장]** 배타적 하위 유형은 슈퍼타입/서브타입으로 표현한다(고객 → 개인고객·법인고객). 물리 전환 방식(단일 테이블·개별 테이블·통합)은 물리 단계에서 결정한다.
- 엔터티 성격(기본·중심·행위) 구분은 테크니컬 노트(6.4.1.2절)를, 엔터티명 규칙은 명명규칙(5.6.2절)을 따른다.

5.7.4 속성 정의

- **[필수]** 모든 속성은 **원자값(1NF)**을 갖는다. 더 쪼갤 수 있으면 쪼갠다(주소 → 우편번호·기본주소·상세주소). 단, 업무상 항상 하나로만 다루는 값은 통합을 허용한다.
- **[필수]** 모든 속성은 **표준용어명**을 명칭으로, **표준도메인**을 형식으로 갖는다(5.4절). 개별 속성에서 타입·길이를 임의 지정하지 않으며, 분류어가 도메인을 결정한다(5.3.2절).
- **[필수]** 코드성 속성(분류어 '코드')은 반드시 **표준코드**에 연결하여 허용값을 제한한다(5.5절). 자유 입력 문자열에 상태·구분을 담지 않는다.
- **[권장]** **추출(파생) 속성**은 원칙적으로 최소화한다. 저장이 불가피하면(집계 성능 등) 산출식과 재계산 시점을 정의서에 명시하고 원천과의 정합성 검증 주기를 정한다.
- **[권장]** Null 정책을 명시한다. 업무상 반드시 존재하는 값은 필수(Not Null)로, 발생 시점이 다른 값은 선택으로 둔다. "조건부 필수"는 조건을 설명에 남기며, `NULL`의 3치 논리가 유발하는 오류는 코딩 가이드(5.9.1.4절)를 참조한다.

```
-- 나쁨: 다중값을 한 속성에 콤마로 저장 → 검색·정합성·집계 불가
고객.연락처 = "010-1111-2222, 02-333-4444"

-- 좋음: 반복은 별도 엔터티로 (고객 1 : N 고객연락처)
고객연락처( 고객번호(FK), 연락처구분코드, 연락처값 )
```

5.7.5 식별자 정의

- **[필수]** 모든 엔터티는 각 인스턴스를 유일하게 식별하는 **주식별자**를 갖는다. 후보키가 여럿이면 최소성·불변성·간결성을 기준으로 주식별자를 선정하고, 나머지 후보키는 유니크 제약으로 보존한다.
- **[필수]** 주식별자 속성에는 Null과 사후 값 변경을 허용하지 않는다. 식별자 값이 바뀌면 이를 참조하는 관계 전체가 흔들린다.

- **[권장]** 식별자는 업무 식별자(본질키)를 우선 검토하되, 본질키가 길거나·변경될 수 있거나·복합도가 높으면 인조키(surrogate)를 도입하고 본질키는 유니크 제약으로 보존한다.
- 주식별자·외래식별자 표기와 식별/비식별 관계 규칙은 테크니컬 노트(6.4.1.2절)를 따른다.

5.7.6 관계 정의

- **[필수]** 엔터티 사이의 업무 규칙을 **관계**로 표현하고, 각 관계에 차수(1:1, 1:N, M:N)와 선택성(필수/선택)을 명시한다.
- **[필수] M:N 관계는 논리 단계에서 교차(연관) 엔터티로 해소**한다. 교차 엔터티는 양쪽 엔터티의 식별자를 상속받으며 관계 고유의 속성을 가질 수 있다.
- **[필수]** 관계로 상속되는 **외래 식별자**는 원천의 명칭을 유지하고, 동일 원천이 서로 다른 역할로 상속되면 역할 수식어로 구별한다(주문고객·수취고객)(5.6.3절).
- **[권장]** 부모의 식별자가 자식의 주식별자 일부가 되면 **식별 관계**, 일반 속성으로 상속되면 **비식별 관계**로 구분한다. 관계 차수 표기법은 테크니컬 노트(6.4.1.2절)를 따른다.

5.7.7 논리 모델 검증

- **[필수]** 논리 모델은 **정규화(3NF 지향)**하여 이상현상(갱신·삽입·삭제 anomaly)을 제거한다. 반복 그룹은 1NF로, 부분 함수 종속은 2NF로, 이행 함수 종속은 3NF로 분해한다. 성능을 위한 반정규화는 논리 단계가 아니라 물리 단계(5.8.6절)에서 근거와 함께 판단한다.
- **[필수]** 모든 엔터티에 주식별자가 있는지, 모든 관계의 차수·선택성이 정의됐는지, 소속 엔터티가 불명확한 고아 속성이 없는지 점검한다.
- **[필수]** 모든 속성이 표준용어·표준도메인·표준코드에 연결됐는지(표준 준수) 확인한다. 비표준 명칭이나 임의 타입이 남아 있으면 표준화 단계로 되돌린다.
- **[권장]** 완성된 논리 모델은 업무 담당자와 함께 주요 시나리오(등록·조회·변경)를 대입해 업무 규칙이 정확히 표현됐는지 검토한다.

5.8 물리 데이터 모델링 가이드

물리 데이터 모델링은 논리 모델을 **채택 DBMS의 실제 객체**(데이터베이스·테이블·컬럼·제약·인덱스)로 전환하는 활동이다. 표준(명칭·도메인·키·참조)은 유지하고 타입 표기·저장 구조·성능·무결성을 DBMS에 맞춰 확정한다. DBMS별 타입·구현 요령은 테크니컬 노트(6.4.2절)를, 명칭은 명명규칙을 참조한다.



그림 5.2. 물리 데이터 모델링 절차

물리 데이터 모델링은 그림 5.2 와 같이 여섯 단계로 진행된다. 먼저 워크로드 성격과 성능·저장 요건을 바탕으로 대상 DBMS를 선택하고(5.8.1절), 논리 엔터티를 물리 테이블로 변환한다(5.8.2절). 이어 속성을 컬럼으로 옮기며 도메인에 따라 타입·길이를 확정하고(5.8.3절), 기본키와 외래키를 설정한다. 그다음 조회 성능과 무결성을 고려해 인덱스와 제약을 정의하고, 마지막으로 필요한 경우 반정규화를 적용한 뒤 성능 최적화 관점에서 물리 모델을 검증한다. 표준(명칭·도메인·키·참조)은 그대로 유지하되, 저장 구조와 성능은 채택 DBMS에 맞추어 구체화한다.

5.8.1 대상 DBMS 선택

- **[필수]** DBMS·데이터베이스는 **워크로드 성격(OLTP/OLAP)**, 라이선스·비용, 요구 기능(파티셔닝·복제·확장), 운영 조직의 역량을 종합해 선정한다. 본 방법론은 Oracle과 PostgreSQL을 예시 기준으로 삼는다(6.4.2.1절).
- **[필수]** 문자셋은 **유니코드**를 기본으로 한다(Oracle `AL32UTF8`, PostgreSQL `UTF8`). 문자셋이 한글 바이트 길이와 정렬 순서(collation)를 좌우하므로 도메인의 길이 정의와 일관되게 맞춘다.
- **[권장]** 운영·개발·분석 등 용도별로 데이터베이스·스키마를 분리하고, 스키마(소유자) 단위로 권한을 관리한다.
- **[권장]** 대용량·이력 테이블은 별도 테이블스페이스로 분리하여 I/O·백업을 독립적으로 운영한다.

5.8.2 테이블 정의

- **[필수]** 엔터티–테이블은 **1:1을 기본**으로 전환한다. 조회 성능·보안을 위해 수직 분할하거나 여러 엔터티를 통합하면, 관련 엔터티를 정의서에 모두 남긴다(6.4.2.2절).
- **[필수]** 논리 단계에서 도출한 슈퍼타입/서브타입의 물리 전환 방식(단일·개별·통합 테이블)은 조회·저장 패턴을 근거로 이 단계에서 확정한다.
- **[권장]** 대량·고빈도 테이블은 **파티셔닝**을 검토하고 분할 키(예: 발생일자 월 단위)를 명시한다. 이력 보존은 이력 테이블(`_HST`)로 설계하고, 삭제는 물리 삭제 대신 상태·이력 관리로 대체할지 개인정보 파기 정책과 함께 판단한다.

5.8.3 컬럼 정의

- **[필수]** 컬럼의 타입·길이는 **적용 도메인의 정의를 그대로** 따르고 개별 조정하지 않는다(5.4절). 같은 용어는 어느 테이블에서든 동일 컬럼명·동일 타입이다.
- **[필수]** **조인에 쓰이는 키 컬럼의 타입·길이는 반드시 일치**시킨다. 불일치는 묵시적 형변환 → 인덱스 무력화로 이어지는 대표적 성능 사고다(5.9.1.5절).
- **[필수]** 시스템 관리 공통 컬럼(최초등록자·일시, 최종수정자·일시)을 전사 표준으로 정의하고 모든 테이블 말미에 동일 순서로 배치한다(표 5.19).
- **[권장]** 개인정보 컬럼은 개인정보 여부·암호화 여부를 표기하고, 암호화 대상은 **암호문 출력 길이**로 물리 길이를 산정한다.

5.8.4 키 정의

- **[필수]** 모든 테이블에 **기본키(PK)**를 정의한다. 논리 주식별자를 PK로 전환하며, 제약명은 명명규칙의 `PK_`+테이블명 규칙을 따른다(표 5.20).
- **[필수]** 논리 관계는 **외래키(FK)**로 구현한다. FK 컬럼은 부모 PK와 **동일 타입·길이**로 정의하고(5.8.3절), 참조 무결성 동작(자식의 갱신·삭제 규칙)을 명시한다. FK 명은 `FK_`+자식테이블+`_`+부모테이블 규칙을 따른다(표 5.20).
- **[권장]** 인조키(surrogate)를 PK로 채택한 경우, 업무 식별자(본질키)는 **유니크 제약**으로 함께 보존하여 중복 입력을 막는다.

5.8.5 인덱스·제약 정의

- **[필수]** 무결성은 제약으로 강제한다: 필수값은 `NOT NULL`, 유일성은 `UNIQUE`, 참조는 `FK`, 허용값은 `CHECK`(또는 코드 테이블 FK), 기본값은 `DEFAULT`.
- **[필수] FK 컬럼과 자주 조회되는 조건·정렬 컬럼에 인덱스**를 생성한다. FK에 인덱스가 없으면 부모 삭제·조인 시 전 건 스캔이 발생한다(6.4.2.3.2절). 인덱스명은 유니크는 `UX_`, 일반은 `IX_` 규칙을 따른다(표 5.20).
- **[권장]** 인덱스는 조회를 빠르게 하지만 입력·갱신 비용과 저장 공간을 늘리므로 **과도한 인덱스를 지양**한다. 실제 쿼리 패턴에 근거해 선정하고, 복합 인덱스는 선행 컬럼의 순서를 신중히 정한다.

5.8.6 반정규화·물리 검증

- **[권장] 반정규화는 물리 단계의 선택적 최적화**다. 조인·집계 성능이 실측으로 문제될 때만 적용하고, ① 반정규화 근거와 ② 원본과의 정합성 유지 방법(트리거·배치·애플리케이션)을 반드시 함께 정의한다. 근거 없는 선반정규화는 갱신 이상을 부른다.
- **[필수]** 물리 모델이 표준(명칭·도메인·키·참조)을 유지하는지, 모든 테이블에 PK가 있는지, FK·인덱스·제약이 누락되지 않았는지 점검한다.
- **[권장]** 예상 데이터량과 주요 쿼리로 성능을 점검하고, 필요 시 파티셔닝·인덱스·반정규화를 조정한다. 반정규화 적용 내역은 근거와 함께 정의서에 남긴다.

```
-- 나쁨: 근거 없이 주문에 고객명을 복제 → 고객명 변경 시 갱신 이상
ORD( ORD_NO, CUST_NO, CUST_NM, ... )

-- 좋음: 정규화 유지, 조회는 조인/뷰로 (성능 문제가 실측될 때만 반정규화)
ORD( ORD_NO, CUST_NO, ... )  JOIN  CUST( CUST_NO, CUST_NM, ... )
```

5.9 코딩 가이드

코딩 가이드는 데이터베이스에 접근하는 코드가 표준 모델의 의도를 훼손하지 않고, 읽기 쉽고, 성능 저하 없이 동작하도록 하기 위한 규약이다. 데이터 표준이 "무엇을 어떻게 저장할 것인가"의 약속이라면, 코딩 가이드는 "그것을 어떻게 다룰 것인가"의 약속이다. 아무리 정교한 표준 모델도 `SELECT *`와 리터럴 남발, 컬럼 가공 조건으로 접근하는 순간 그 가치가 잠식된다.

5.9.1 SQL 코딩 가이드

SQL은 표준 데이터 모델과 실제 데이터가 만나는 접점이다. 아무리 정교한 표준 모델도 `SELECT *`와 리터럴 남발, 컬럼을 가공하는 조건으로 접근하는 순간 그 가치가 잠식된다. 본 절은 SQL 작성의 규약을 **① 개요 및 목적 → ② 명명 규칙 → ③ 코드 포맷 및 스타일 → ④ 구문 작성 표준 → ⑤ 성능 및 최적화 → ⑥ 보안 및 안전 관리 → ⑦ 툴 및 자동화**의 7개 영역으로 나누어 규정한다. 각 조항은 앞 절과 마찬가지로 **필수·권장**의 두 수준으로 구분한다(표 5.1 참조). 규약의 취지는 **잘못된 예시(Bad)와 올바른 예시(Good)를 직접 대비**할 때 가장 선명하므로, 본 절은 모든 핵심 조항에 대비 예시를 함께 제시한다.

5.9.1.1 1. 개요 및 목적

1.1 도입 배경

명명·도메인 표준이 데이터의 **구조**를 통제한다면, SQL 코딩 표준은 그 구조에 **접근하는 행위**를 통제한다. 표준이 없는 SQL은 작성자마다 대소문자·들여쓰기·조인 문법·조건 작성 방식이 제각각이 되어, 같은 질의를 읽는 데에도 해석 비용이 들고, 코드 리뷰의 초점이 로직이 아니라 스타일 교정에 소모된다.

더 심각한 것은 스타일의 불일치가 곧 **성능·정합성 사고로 전이**된다는 점이다. 대소문자 혼용은 실행 계획 캐시를 분산시키고, 컬럼을 가공하는 조건은 인덱스를 무력화하며, `WHERE` 절이 빠진 `DELETE`는 테이블 전 건을 지운다. 즉 SQL 코딩 규약은 "미관"의 문제가 아니라 가독성·유지보수성·성능·안전을 동

시에 좌우하는 **품질 장치**다. 본 가이드는 이러한 개인차에서 비롯되는 위험을 조직 차원의 단일 규약으로 제거하기 위해 제정한다.

1.2 기대 효과

- **코드 가독성 향상**: 누가 작성했든 동일한 형태로 읽히므로 로직 파악·인수인계·장애 대응의 시간이 단축된다. 표준화된 코드는 "작성자의 서명"이 아니라 "조직의 자산"이 된다.
- **유지보수 비용 절감**: 일관된 형식은 형상관리 diff를 깨끗하게 하여 변경 이력 추적을 쉽게 하고, 리뷰를 스타일이 아닌 로직에 집중시키며, 표준 검증·정적 분석 도구의 자동 점검을 가능하게 한다.
- **쿼리 성능 최적화**: sargable 조건·바인드 변수·명시적 조인처럼 성능에 직결되는 관행을 규약으로 강제함으로써, 성능 문제를 사후 튜닝이 아니라 **작성 시점에 예방**한다.

1.3 적용 범위

- **적용 대상 코드**: 본 가이드는 5.1.2절 의 적용 범위를 SQL 코드 관점에서 구체화한 것으로, 신규 개발·변경되는 모든 애플리케이션 내장 SQL, 저장 프로시저·함수·트리거, 배치 및 데이터 이행(migration) 스크립트, 수작업 운영 쿼리에 적용한다.
- **적용 대상 인원**: SQL을 작성하는 개발자, 리뷰하는 리더·DA(Data Architect), 운영·튜닝을 담당하는 DBA 전원이 대상이다. 규약의 준수 여부는 코드 리뷰와 자동화 도구(5.9.1.7절)로 점검한다.
- **ORM·자동 생성 SQL**: ORM(객체–관계 매핑) 프레임워크가 생성하는 SQL도 최종적으로 본 규약의 정확성·성능 기준(SELECT * 회피, sargable 조건, 바인드 변수)을 충족해야 하며, 프레임워크가 이를 위배하면 네이티브 쿼리로 대체한다.
- **적용 제외**: 패키지 솔루션이 자체 생성·관리하는 내부 오브젝트 등 5.1.2절 에서 정한 제외 대상은 본 가이드에서도 제외한다. 다만 그 데이터를 추출·적재하는 인터페이스 영역부터는 본 규약을 적용한다.

5.9.1.2 2. 명명 규칙

SQL 코드가 다루는 이름은 두 부류다. 하나는 이미 표준으로 확정된 **데이터베이스 오브젝트명**(테이블·컬럼·제약)이고, 다른 하나는 작성자가 그 자리에서 만들어 내는 **코드 지역 명칭**(별칭, CTE명, 바인드 변수명)이다. 전자는 5.6절 의 규칙을 그대로 따르며, 본 절은 그 핵심 결정을 SQL 관점에서 재확인하고 후자의 규칙을 추가한다.

기본 원칙 — 표기법과 언어.

- **[필수]** 물리 식별자는 **영문 대문자 + 스네이크 케이스**(`UPPER_SNAKE_CASE`)로 작성한다. 카멜 케이스(`CustOrdHst`)는 사용하지 않는다. 스네이크 케이스는 단어 경계가 `_`로 명시되어 표준단어 약어의 조합(5.2.7절)을 육안·기계 양쪽에서 복원할 수 있고, 대문자 통일은 식별자 대소문자 구분이 DBMS·OS·따옴표 인용 여부에 따라 달라지는 데서 오는 이식성 문제를 원천 차단한다 (Holywell 2016년).
- **[필수]** 식별자에는 영문·숫자·`_`만 사용하고 영문자로 시작한다. 한글·공백·하이픈 식별자, 그리고 로마자 음차(`GOGAEK`)를 금지한다(5.2.7절). 식별자 표기의 좋고 나쁜 예는 표 5.21 와 같다.

표 5.21. 식별자 표기 규칙의 대비

구분	나쁨(Bad)	좋음(Good)
예약어·함수 표기	`select custno, count(*) from custorder`	`SELECT CUST_NO, COUNT(*) FROM CUST_ORD`
케이스 규칙	`CustOrderHist` (카멜)	`CUST_ORD_HST` (대문자 스네이크)
언어·음차	`GOGAEK_JUSO` (음차)	`CUST_ADDR` (표준약어)

오브젝트 명명.

- **[필수]** 테이블·컬럼명은 5.6.4절, 5.6.5절 을 따른다. 테이블은 대응 엔터티명(**단수형**)의 표준약어 조합으로 하고, 복수형(`CUSTOMERS`)을 쓰지 않는다. 뷰·집계·이력·임시·인터페이스 테이블의 접두·접미어는 표 5.17 를, 제약·인덱스 등 부속 오브젝트는 표 5.20 를 준용한다.
- **[필수]** 프로그래머블 오브젝트(프로시저·함수 등)는 종류가 이름에서 즉시 드러나도록 다음 접두어 체계(표 5.22)를 사용한다. 접두어는 물리명 규칙에 맞추어 **대문자**로 쓴다(`sp_`·`fn_`의 소문자 표기는 채택하지 않는다).

표 5.22. 프로그래머블 오브젝트 접두어

오브젝트	접두어	예시	비고
저장 프로시저	`SP_`	`SP_CUST_GRD_UPD`	동사(처리 내용)를 약어에 포함
함수	`FN_`	`FN_MASK_RRNO`	반환값의 의미를 약어에 포함
트리거	`TRG_`	`TRG_ORD_AINS`	대상 테이블 + 시점(A/B, INS/UPD/DEL)
패키지·모듈	`PKG_`	`PKG_SETTLE`	관련 프로시저·함수 묶음
시퀀스	`SEQ_`	`SEQ_ORD_NO`	부여 대상 컬럼 명시

컬럼 명명 — 식별자와 날짜.

- **[필수]** 식별자 컬럼은 `ID`·`NO`만 단독으로 두지 않고 **엔터티명을 포함**한다(`ID` × → `CUST_NO`, `CUST_ID`). 식별자는 FK로 상속되는 순간 소속 엔터티 정보가 이름에서 사라지면 조인 경로를 이름만으로 추적할 수 없기 때문이다(5.6.3절). 즉 범용 `id`가 아니라 `테이블_id` 방식을 표준으로 한다. 시스템이 부여하는 기술 식별자는 분류어 `ID`(`_ID`), 업무 규칙으로 부여되는 식별번호는 분류어 `번호`(`_NO`)로 구분한다(표 5.6).
- **[필수]** 동일 원천 식별자가 서로 다른 **역할**로 두 번 이상 상속되면 역할 수식어를 붙인다(`ORD_CUST_NO`, `RCV_CUST_NO`).
- **[필수]** 날짜·시각 컬럼의 접미어는 임의 지정(`_at`, `_date`)이 아니라 **분류어의 영문약어**에서 파생한다. 정밀도 요건이 접미어와 도메인을 함께 결정한다(5.3.2절, 5.4.2절).

표 5.23. 날짜·시각 접미어 표준

성격	일반적 관행(참고)	본 표준 접미어	분류어 / 도메인
날짜(일자)	`_date`	`_DT`	일자 / `CHAR(8)` `YYYYMMDD`
일시(타임스탬프)	`_at`, `_datetime`	`_DTM`	일시 / `TIMESTAMP`
년월	`_ym`, `_month`	`_YM`	년월 / `CHAR(6)` `YYYYMM`

예약어를 식별자로 사용 금지.

- **[필수]** DBMS 예약어(`ORDER`, `USER`, `DATE`, `LEVEL`, `COMMENT` 등)를 테이블·컬럼·별칭명으로 사용하지 않는다. 따옴표 인용(`"ORDER"`)으로 회피할 수 있으나, 인용 부호에 의존하는 코드는 이식성·가독성을 해치므로 명칭 자체를 바꾼다 Celko (2005년). 예약어 casing 규칙(키워드는 대문자)은 5.9.1.3절 를 따른다.

코드 지역 명칭 — 별칭·CTE·바인드 변수.

- **[필수]** 테이블 별칭은 무의미한 순번(`a`, `b`, `c`)이 아니라 **테이블 물리명의 의미 있는 축약**으로 짓고, 한 문장 안에서 일관되게 사용한다(`CUST` → `c`, `ORD` → `o`, `ORD_DTL` → `od`). 계산식·집계 결과의 별칭에는 반드시 의미 있는 이름을 부여하며, 그 이름도 표준단어 약어로 짓는다(`SUM(o.ORD_AMT)` → `TOT_ORD_AMT`).
- **[권장]** CTE(공통 테이블 식) 이름은 그 단계가 만들어 내는 결과 집합의 의미를 표준약어로 표현한다(`MTH_NEW_CUST`, `SETTLE_TGT`). `T1`, `TMP` 같은 익명 이름은 다단 쿼리의 가독성 이점을 스스로 무력화한다.
- **[권장]** 바인드 변수명은 대응 컬럼·파라미터의 의미가 드러나게 짓는다(`:1`, `:p` × → `:ST_DT`, `:CUST_NO`).

5.9.1.3 3. 코드 포맷 및 스타일

포맷은 취향의 문제로 보이지만, 아래 규칙들은 가독성뿐 아니라 형상관리·성능·오류 예방과 직접 연결된다.

대소문자.

- **[필수]** SQL 예약어(`SELECT`, `FROM`, `WHERE`, `JOIN`, `GROUP BY` 등)와 내장 함수는 **대문자**로, 테이블·컬럼 식별자는 표준 물리명 그대로 작성한다 (Holywell 2016년). 표기 일관성은 가독성 문제인 동시에 **성능 문제**다. 다수의 DBMS는 SQL 문장의 문자열 자체를 해시하여 실행계획 캐시를 공유하므로, `SELECT`와 `select`가 섞이면 같은 질의가 서로 다른 문장으로 취급되어 하드 파싱이 반복된다.

줄 바꿈과 들여쓰기.

- **[필수]** 주요 절(`SELECT`, `FROM`, `WHERE`, `GROUP BY`, `HAVING`, `ORDER BY`)은 각각 새 줄에서 시작하고, 조인·서브쿼리·논리 연산의 계층은 들여쓰기로 표현한다.
- **[필수]** 들여쓰기는 **공백(Space)**을 사용하고 탭(Tab)을 쓰지 않는다. 탭은 편집기·뷰어 설정에 따라 폭이 달라져 정렬이 깨지고, 형상관리에서 공백과 뒤섞이면 무의미한 diff를 만든다. 들여쓰기 폭은 전사 단일 값(예: 공백 4칸)으로 고정한다.
- **[권장]** 절 키워드의 오른쪽 끝을 세로로 맞추어 값이 정렬되게 하는 배치를 표준으로 삼는다(본절 5.9.1.8절 의 예시 참조). 어떤 정렬 방식을 택하든 **전사 단일 방식**으로 통일하고 자동 포맷터(5.9.1.7절)로 강제한다.

콤마 위치.

- **[권장]** `SELECT` 목록의 콤마는 **줄 앞(Leading Comma)**에 배치한다. 컬럼 추가·삭제 시 변경되는 줄이 해당 컬럼 한 줄로 국한되어 형상관리 diff가 깨끗해지고, 마지막 컬럼 뒤 콤마 누락 오류를 예방한다.

```
-- 나쁨(trailing comma): 컬럼 추가 시 직전 줄도 함께 변경되고, 마지막 콤마 실수가 잦다
SELECT c.CUST_NO,
       c.CUST_NM,
       c.JOIN_DT
  FROM CUST c;

-- 좋음(leading comma): 추가·삭제가 한 줄로 국한되고 콤마 누락이 눈에 띈다
SELECT c.CUST_NO
     , c.CUST_NM
     , c.JOIN_DT
  FROM CUST c;
```

연산자와 괄호.

- **[필수]** 이항 연산자(`=`, `<`, `>`, `+`, `||` 등)와 논리 연산자(`AND`, `OR`) 앞뒤에는 공백을 한 칸씩 둔다(`A=B` × → `A = B`).

- **[권장]** `AND`와 `OR`가 함께 쓰이는 복합 조건은 괄호로 우선순위를 명시한다. 괄호 없는 `A AND B OR C`는 연산자 우선순위(`AND` 우선)에 대한 오해로 논리 결함을 부른다.

```
-- 나쁨: AND/OR 우선순위가 눈에 보이지 않아 의도와 다르게 해석될 수 있다
WHERE CUST_GRD_CD = 'VIP' OR CUST_GRD_CD = 'GOLD' AND USE_YN = 'Y'

-- 좋음: 괄호로 의도를 못박는다 (VIP·GOLD 중 하나 이면서 사용중)
WHERE ( CUST_GRD_CD = 'VIP' OR CUST_GRD_CD = 'GOLD' )
  AND USE_YN = 'Y'
```

주석(Comment).

- **[필수]** 모든 쿼리·프로그램 단위의 상단에 블록 주석(`/* */`)으로 **메타데이터**(목적, 작성자, 작성일, 변경 이력)를 기재한다. 표준 헤더 양식을 정해 두고 도구의 스니펫으로 삽입하게 한다.
- **[권장]** 업무 규칙이 담긴 조건(상태코드 값 등)에는 한 줄 주석(`--`)으로 의미를 병기한다. `ORD_STS_CD = '10'`이라는 조건은 주석 없이는 아무 정보도 주지 못한다.
- **[권장]** 주석 처리로 비활성화한 "죽은 코드"를 소스에 남기지 않는다. 과거 코드의 보존은 주석이 아니라 형상관리(Git)의 책임이다.

```
/*
 * 목적   : 당월 신규 가입 고객의 정상 주문 합계 조회
 * 작성   : 홍길동 (2026-07-15)
 * 변경   : 2026-07-20 김철수 - 주문상태 '부분취소(15)' 제외
 */
```

5.9.1.4 4. SQL 구문 작성 표준

데이터 조회(DML).

- **[필수]** `SELECT *`를 금지하고 필요한 컬럼만 명시한다. 근거는 세 가지다. ① 테이블 컬럼 추가·순서 변경 시 이를 위치 기반으로 소비하는 코드가 소리 없이 오동작한다. ② 불필요한 컬럼 전송으로 I/O·네트워크가 낭비되고, 커버링 인덱스(인덱스만으로 응답) 최적화가 봉쇄된다. ③ 뷰 정의에 쓰인 `*`는 뷰 생성 시점의 컬럼으로 고정되어 기반 테이블 변경 시 뷰가 무효화되거나 불일치한다.
- **[필수]** `INSERT` 문에는 반드시 컬럼 목록을 명시한다. 컬럼 목록 없는 `INSERT`는 테이블 구조 변경 시 오동작을 일으키는 잠재적 위험 요소다.
- **[필수]** 컬럼·테이블 별칭에는 `AS`를 명시하고(테이블 별칭의 `AS` 생략은 일부 DBMS 문법 예외로 허용), 계산식·집계 결과에는 반드시 의미 있는 별칭을 부여한다(5.9.1.2절).

조인(JOIN).

- **[필수]** 콤마로 나열하는 **암시적 조인(Implicit Join)을 금지**하고, `INNER JOIN`·`LEFT OUTER JOIN` 등 **명시적 조인(Explicit Join)과** `ON` **조건**을 사용한다 (ISO/IEC 2023년b). 암시적 조인은 조인 조건과 필터 조건이 `WHERE`에 뒤섞여 판독이 어렵고, 조인 조건을 빠뜨리면 경고 없이 카티전 곱(전 행 × 전 행)을 만든다.

```
-- 나쁨: 암시적 조인 – 조인/필터 구분 불명, 조인 조건 누락 시 카티전 곱
SELECT c.CUST_NO, o.ORD_NO
  FROM CUST c, ORD o
 WHERE o.CUST_NO = c.CUST_NO
   AND o.ORD_STS_CD = '10';

-- 좋음: 명시적 조인 – 조인 조건은 ON, 필터 조건은 WHERE
SELECT c.CUST_NO, o.ORD_NO
  FROM CUST c
 INNER JOIN ORD o
    ON o.CUST_NO = c.CUST_NO
 WHERE o.ORD_STS_CD = '10';
```

- **[필수]** 3개 이상의 테이블을 조인하는 SQL에서는 모든 컬럼에 테이블 별칭을 접두한다. 별칭 없는 컬럼은 소속 판독 비용을 발생시키고, 나중에 다른 테이블에 동명 컬럼이 추가되면 모호성 오류로 갑자기 실패한다.
- **[필수]** 외부 조인에서 **보존 측 테이블의 필터 조건은 `ON`에** 둔다. `WHERE`에 두면 `NULL` 행이 걸러져 사실상 `INNER JOIN`으로 변질된다 — 이는 문체가 아니라 결과를 바꾸는 의미의 문제다.

```
-- 의도: 전체 고객과, 있다면 2026년 주문을 함께
-- 나쁨: WHERE의 o.* 조건이 주문 없는 고객을 제거 (INNER JOIN화)
SELECT c.CUST_NO, o.ORD_NO
  FROM CUST c
  LEFT OUTER JOIN ORD o ON o.CUST_NO = c.CUST_NO
 WHERE o.ORD_DT >= '20260101';

-- 좋음: 보존 측을 유지하려면 조건을 ON으로
SELECT c.CUST_NO, o.ORD_NO
  FROM CUST c
  LEFT OUTER JOIN ORD o ON o.CUST_NO = c.CUST_NO
                       AND o.ORD_DT >= '20260101';
```

서브쿼리와 CTE.

- **[권장]** 여러 겹 중첩된 인라인 서브쿼리를 지양하고, 복잡한 다단 로직은 **CTE(`WITH` 절)**로 분리하여 각 단계에 이름을 부여한다(5.9.1.2절). 중첩 서브쿼리는 안쪽부터 거꾸로 읽어야 하지만, CTE는 위에서 아래로 읽힌다.
- **[권장]** CTE가 여러 번 참조될 때의 실체화(materialization) 여부는 DBMS별로 다르므로, 대량 데이터 CTE의 반복 참조는 실행계획을 확인한다.

NULL 처리.

NULL의 3치 논리(Three-Valued Logic, 3VL)는 데이터베이스(SQL)에서 참(`TRUE`)·거짓(`FALSE`) 외에 '알 수 없음(`UNKNOWN`)'이라는 제3의 진리값을 다루는 논리 체계다 (Winand 2023년). 값이 비어 있거나 모르는 상태인 `NULL`과 연산하면 결과의 참·거짓이 확정되지 않아 `UNKNOWN`이 반환되는 데서 비롯된다. `WHERE`·`ON`·`CHECK` 절은 조건이 `TRUE`인 행만 통과시키고 `FALSE`와 `UNKNOWN`을 똑같이 탈락시키므로, `UNKNOWN`의 취급을 오해하면 문법 오류 없이 조용히 틀린 결과가 나온다.

- **[필수]** `NULL`의 3치 논리를 정확히 다룬다 (ISO/IEC 2023년b). `NULL`과의 모든 비교(`= NULL`, `<> NULL`)는 `UNKNOWN`이며, 특히 `NOT IN` 서브쿼리의 결과에 `NULL`이 하나라도 포함되면 **전체 결과가 공집합**이 된다. 이는 문법 오류가 아니라 조용히 틀린 답을 내는 결함이므로, 부정 존재 검사는 `NOT EXISTS`를 사용한다.

```
-- 나쁨: 서브쿼리 결과에 NULL이 있으면 0건 반환 (논리 결함)
SELECT c.CUST_NO
  FROM CUST c
 WHERE c.CUST_NO NOT IN ( SELECT o.CUST_NO FROM ORD o );

-- 좋음: NULL에 안전하고 일반적으로 실행계획도 우수
SELECT c.CUST_NO
  FROM CUST c
 WHERE NOT EXISTS ( SELECT 1
                      FROM ORD o
                     WHERE o.CUST_NO = c.CUST_NO );
```

- **[필수]** `NULL` 대체 함수는 **ANSI 표준인 `COALESCE`를 기본**으로 채택하여 이식성을 확보한다. DBMS 방언 함수는 표준 대체가 불가능한 경우에만 사용하고 사유를 주석으로 남긴다(표 5.24).

표 5.24. NULL 처리 함수의 표준안

표준 함수	DBMS 방언(동등)	비고
`COALESCE(a, b, ...)`	`NVL(a, b)` (Oracle), `IFNULL(a, b)` (MySQL), `ISNULL(a, b)` (SQL Server)	`COALESCE`는 인자 2개 이상 · ANSI 표준 → **기본 채택**
`NULLIF(a, b)`	(공통)	두 값이 같으면 `NULL` 반환. 0으로 나누기 회피 등에 사용

집합 연산.

- **[필수]** 중복이 발생할 수 없는 집합 결합에는 `UNION`이 아닌 `UNION ALL`을 사용한다. `UNION`은 항상 전체 결과의 중복 제거(정렬 또는 해시)를 수행하므로 불필요한 사용은 성능 낭비이며, 반대로 중복 제거가 필요한 자리에 `UNION ALL`을 쓰면 정합성 결함이다. 선택은 취향이 아니라 데이터 분석의 결과여야 한다.
- **[권장]** `DISTINCT`는 결과 중복의 **원인을 이해한 뒤에만** 사용한다. 1:N 조인으로 생긴 중복을 `DISTINCT`로 덮는 것은 원인(조인 설계)을 두고 증상만 가리는 것이다. `EXISTS`로 조인 자체를 준조인화하는 것이 정도(正道)다.

DBMS 이식성 — 방언 차이의 관리.

본 방법론은 Oracle과 PostgreSQL을 기준 DBMS로 삼는다(6.3.3절). 두 DBMS는 ANSI SQL을 공유하지만 함수·구문·`NULL` 취급에 방언 차이가 있어, 무심코 한쪽 방언으로 작성한 코드가 다른 쪽에서 조용히 오작동한다.

- **[권장]** 표준(ANSI) 구문을 우선하고, 방언 고유 구문은 한곳에 격리하여 대체 지점을 명확히 한다. 이식이 예정된 코드일수록 방언 의존을 줄인다.
- **[필수]** 두 DBMS에서 동작이 갈리는 지점은 방언에 기대지 말고 명시적으로 작성한다. 대표적 차이는 다음과 같다(표 5.25).

표 5.25. Oracle · PostgreSQL 주요 방언 차이

구분	Oracle	PostgreSQL	표준·권장
현재 일시	`SYSDATE`, `SYSTIMESTAMP`	`now()`, `CURRENT_TIMESTAMP`	ANSI `CURRENT_TIMESTAMP` (양쪽 지원)
빈 문자열	`''` = `NULL`	`''` ≠ `NULL` (빈 문자열 보존)	값 없음은 `IS NULL`로 판정
결합 시 `NULL`	`'A' || NULL` → `'A'`	`'A' || NULL` → `NULL`	`CONCAT()` · `COALESCE`로 통일
형변환	`TO_NUMBER`, `TO_DATE`	`::` 캐스트, `TO_DATE`	ANSI `CAST(... AS ...)`
더미 조회	`SELECT 1 FROM DUAL`	`SELECT 1` (FROM 생략)	표현식 조회는 방언 격리
불리언 저장	네이티브 타입 없음 → `CHAR(1)`	네이티브 `BOOLEAN`	물리 저장은 `_YN` `CHAR(1)` 표준(표 5.18)

결과 행 제한 구문의 DBMS별 차이는 표 5.27 를 참조한다.

> ⚠ 빈 문자열과 NULL — DBMS가 다르게 해석한다
>
> Oracle은 빈 문자열 `''`을 `NULL`과 동일하게 취급하지만 PostgreSQL은 길이 0의 문자열로 구분한다. 그 결과 `WHERE COL = ''`은 Oracle에서는 결코 참이 되지 않고(=`NULL` 비교 → `UNKNOWN`), PostgreSQL에서는 빈 문자열 행에 대해 참이 된다. 두 DBMS에서 동일하게 동작시키려면 값 없음의 판정을 `IS NULL`로 명시하고, 애초에 빈 문자열이 저장되지 않도록 애플리케이션과 제약(`CHECK (COL <> '')`)에서 차단한다.

5.9.1.5 5. 성능 및 최적화 가이드

인덱스 활용 — 컬럼을 가공하지 않는다.

- **[필수]** 인덱스 대상 컬럼을 `WHERE`·`JOIN` 절에서 **가공하지 않는다**(sargable 조건 유지). 컬럼에 함수·연산·형변환이 적용되는 순간 그 컬럼의 인덱스는 사용 불가가 된다. 가공은 컬럼이 아니라 **상수 쪽**에 한다 (Winand 2012년). 컬럼 가공 조건의 개선 예는 표 5.26 와 같다.

표 5.26. 컬럼 가공 조건의 개선

나쁨(인덱스 무력화)	좋음(sargable)
`WHERE SUBSTR(REG_DT,1,6) = '202607'`	`WHERE REG_DT BETWEEN '20260701' AND '20260731'`
`WHERE YEAR(ORD_DT) = 2026`	`WHERE ORD_DT >= '20260101' AND ORD_DT < '20270101'`
`WHERE SAL_AMT * 1.1 > 50000`	`WHERE SAL_AMT > 50000 / 1.1`
`WHERE CUST_NO = 12345` (문자 컬럼)	`WHERE CUST_NO = '12345'`

- **[필수]** 조인·비교의 양변은 데이터 타입이 일치해야 하며 묵시적 형변환에 의존하지 않는다. 위 표의 마지막 행이 전형이다 — 문자 컬럼과 숫자 리터럴을 비교하면 DBMS는 대개 **컬럼 쪽을 숫자로 변환**하여 인덱스를 무력화한다. 도메인 표준(5.4절)이 지켜지면 이 문제의 대부분은 스키마 수준에서 예방된다 — 코딩 가이드와 도메인 가이드는 이렇게 맞물린다.

비효율 구문의 개선.

- **[필수]** 선행 와일드카드 검색(`LIKE '%키워드%'`)을 지양한다. 앞이 열린 패턴은 인덱스 범위 탐색이 불가능하여 전 건 스캔을 유발한다. 전방 일치(`LIKE '키워드%'`)로 대체하거나, 부분·전문 검색이 업무 요건이라면 전문 검색 인덱스(Full-Text) 또는 전용 검색엔진을 도입한다.

```
-- 나쁨: 선행 와일드카드 → 인덱스 사용 불가, 전 건 스캔
WHERE CUST_NM LIKE '%김%';

-- 좋음: 전방 일치는 인덱스 범위 탐색 가능
WHERE CUST_NM LIKE '김%';
-- (부분 검색이 진짜 요건이면 전문 검색 인덱스/검색엔진을 사용)
```

- **[권장]** `IN`과 `EXISTS`를 상황에 맞게 선택한다. 소수의 상수 목록 필터에는 `IN`이 명료하고, 존재 여부만 필요한 대량 상관 조건에는 `EXISTS`가 유리하다(첫 일치에서 조기 종료). 부정 조건에서 `NOT IN`은 5.9.1.4절 의 `NULL` 처리 오류가 있으므로 `NOT EXISTS`를 사용한다.

데이터 정렬과 결과 제한.

- **[권장]** 불필요한 `ORDER BY`를 제거한다. 정렬은 비용이 큰 연산이며, 존재 검사(`EXISTS`)나 다른 연산의 입력으로만 쓰이는 서브쿼리 내부의 정렬, 최종 표시에 필요 없는 정렬은 낭비다. 정렬은 결과의 표시 순서가 실제로 의미를 가질 때만 둔다.
- **[필수]** 개발·검증 단계에서 대상 건수가 큰 테이블을 조회할 때는 **행 제한 구문을 필수로** 적용하여 운영 자원을 보호한다. 무심코 실행한 전 건 조회는 운영 장애의 흔한 원인이다.

표 5.27. DBMS별 결과 행 제한 구문

DBMS	행 제한 구문
ANSI / Oracle 12c+ / PostgreSQL / SQL Server 2012+	`FETCH FIRST n ROWS ONLY`
MySQL / MariaDB / PostgreSQL	`LIMIT n`
SQL Server	`SELECT TOP n ...`
Oracle(구버전)	`WHERE ROWNUM <= n`

옵티마이저와 페이징.

- **[권장]** 옵티마이저 힌트는 최후 수단이다. 힌트는 통계 갱신·데이터 증가·버전 업그레이드에도 계획을 고정시켜 "그때는 맞고 지금은 틀린" 계획을 강제하는 부채가 된다. 사용 시 반드시 주석으로 사유와 검증 일자를 남기고 정기 점검 대상으로 등록한다.
- **[권장]** 대량 페이징에서 `OFFSET` 방식(뒤 페이지로 갈수록 앞 건을 모두 읽고 버림)의 성능 한계를 인지하고, 연속 조회형 화면·배치에는 마지막 키 기준의 seek 방식(`WHERE (ORD_DT, ORD_NO) > (:LAST_DT, :LAST_NO)`)을 우선 검토한다 (Winand 2012년).
- **[권장]** 성능이 중요한 쿼리는 추측이 아니라 **실행계획을 근거로** 판단한다. PostgreSQL은 `EXPLAIN (ANALYZE, BUFFERS)`로, Oracle은 `EXPLAIN PLAN FOR ...` 실행 후 `DBMS_XPLAN.DISPLAY`(실측 계획은 `DBMS_XPLAN.DISPLAY_CURSOR`)로 계획을 확인한다. 인덱스 접근 형태(Index Range Scan vs Full Scan), 조인 방식(Nested Loop·Hash·Merge), 예상 행수와 실제 행수의 괴리를 점검한다.

5.9.1.6 6. 보안 및 안전 관리

방어적 데이터 변경.

- **[필수]** `WHERE` 절이 없는 `UPDATE`·`DELETE`를 금지한다. 전 건 갱신·삭제가 실제 의도라면 주석으로 전 건 대상임을 명시하고 승인 절차를 거친다. 배포·리뷰 도구에 `WHERE` 부재 검출 규칙을 설정한다 (5.9.1.7절).
- **[권장]** 대량·비가역 변경은 실행 전에 **동일 조건의 `SELECT COUNT(*)`로 영향 범위를 먼저 확인**하는 절차를 표준화한다. "몇 건이 바뀔 것인가"를 모른 채 실행하는 DML이 사고의 대부분이다.

```
-- 나쁨: WHERE 없는 DELETE — 전 건 삭제
DELETE FROM CUST;

-- 좋음: 실행 전 영향 범위 확인 → 조건부 삭제
SELECT COUNT(*) FROM CUST WHERE WTHDRW_YN = 'Y' AND WTHDRW_DT < '20200101';
DELETE FROM CUST     WHERE WTHDRW_YN = 'Y' AND WTHDRW_DT < '20200101';
```

- **[필수]** 트랜잭션의 범위는 업무적 원자성의 단위와 일치시키고, 커밋(`COMMIT`)·롤백(`ROLLBACK`)의 경계를 명확히 관리한다. 자동 커밋(auto-commit) 설정에 의존하여 여러 DML의 원자성이 깨지지 않도록 주의하고, 사용자 응답 대기 중에 트랜잭션(락)을 열어 두지 않는다.

- **[권장]** 대량 배치의 DML은 커밋 단위를 설계한다. 무한정 단일 트랜잭션은 언두(undo) 고갈과 락 경합을 부르고, 지나치게 잦은 커밋은 재시작 지점 관리를 복잡하게 한다. 처리량과 재처리 전략(어디서부터 다시 시작하는가)을 함께 정의한 커밋 간격을 표준으로 정한다.
- **[권장]** 삭제가 업무적으로 "말소"를 의미하면 물리 `DELETE` 대신 상태코드 갱신 + 이력 적재를 우선 검토한다. 단, 개인정보 파기 의무가 걸린 데이터는 반대로 물리 삭제(또는 비식별화)가 법적 요건이므로, 삭제 방식은 데이터 보존·파기 정책과 반드시 연동하여 결정한다.

SQL 인젝션 방지.

- **[필수]** 사용자 입력이 개입하는 조건값은 **바인드 변수(Bind Variable)**로 전달한다. 문자열을 결합하여 SQL을 조립하는 방식은 SQL 인젝션의 통로일 뿐 아니라, 값마다 다른 문장으로 취급되어 하드 파싱을 반복시키고 라이브러리 캐시를 오염시킨다 — 보안과 성능이 같은 규칙에서 함께 해결된다.

```
-- 나쁨: 문자열 결합 → SQL 인젝션 노출, 하드 파싱 유발
"SELECT ... FROM CUST WHERE CUST_NO = '" + inputCustNo + "'"

-- 좋음: 바인드 변수 → 주입 차단 + 실행계획 재사용
"SELECT ... FROM CUST WHERE CUST_NO = :CUST_NO"
```

- **[권장]** 정렬 컬럼·테이블명처럼 바인드가 불가능한 동적 요소가 필요하면, 입력을 그대로 결합하지 말고 **허용 목록(whitelist)**으로 검증한 뒤 매핑된 안전한 식별자만 사용한다.

개인정보·민감정보 관리.

- **[필수]** 민감 정보(주민등록번호·카드번호 등)를 조회·출력할 때는 **마스킹 함수 또는 마스킹 뷰**를 적용하고, 평문 원문의 직접 조회는 권한과 감사(audit) 통제 하에서만 허용한다. 대상 컬럼의 암호화·마스킹 여부는 도메인의 보안 속성(5.4.3절)에 정의된 규칙을 따른다.

```
-- 나쁨: 주민등록번호 평문 노출
SELECT CUST_NM, RRNO FROM CUST;

-- 좋음: 마스킹 함수 적용 (뒷자리 마스킹)
SELECT CUST_NM, FN_MASK_RRNO(RRNO) AS RRNO_MSK FROM CUST;
```

5.9.1.7 7. 툴 및 자동화 설정

규약은 사람의 선의가 아니라 **도구로 강제될 때** 지켜진다. 본 절의 조항 상당수는 자동 검사·자동 교정이 가능하므로, 이를 개발 파이프라인에 내장한다.

린터·포맷터.

- **[필수]** SQLFluff 등 SQL 린터/포맷터를 도입하여 스타일 규칙(대문자 키워드, leading comma, 들여쓰기 폭, `SELECT *` 금지 등)을 코드로 고정하고, 커밋 훅(pre-commit) 또는 CI 파이프라인에서 자동 검사·차단한다. 규칙 설정 파일(예: `.sqlfluff`)은 전사 공용으로 형상관리하여 프로젝트 간 편차를 없앤다.
- **[권장]** 린터 규칙은 본 가이드의 조항과 1:1로 매핑하여 관리한다. 문서에만 있고 도구에 없는 규칙은 시간이 지나면 지켜지지 않는다.

IDE·클라이언트 설정.

- **[권장]** DBeaver·DataGrip 등 주요 SQL 클라이언트의 자동 정렬·포맷 설정(키워드 대소문자, 콤마 위치, 들여쓰기)을 본 가이드에 맞추어 통일하고, 설정 파일(export)을 배포하여 개인별 포맷 차이를 제거한다.
- **[권장]** 도구 설정과 본 문서의 조항이 어긋나면 **본 가이드를 단일 진실 원본(SSOT)**으로 삼아 도구를 문서에 맞춘다. 그 역이 아니다.

5.9.1.8 종합 예시

```sql
/*
 * 목적   : 당월 신규 가입 고객의 정상 주문 합계 상위 조회
 * 작성   : 홍길동 (2026-07-15)
 * 변경   : 2026-07-20 김철수 - 주문상태 '부분취소(15)' 제외 조건 추가
 */
SELECT c.CUST_NO
     , c.CUST_NM
     , COUNT(o.ORD_NO)   AS ORD_CNT
     , SUM(o.ORD_AMT)    AS TOT_ORD_AMT
  FROM CUST c
 INNER JOIN ORD o
    ON o.CUST_NO = c.CUST_NO
 WHERE c.JOIN_DT BETWEEN :ST_DT AND :ED_DT     -- 바인드 변수, 컬럼 무가공
   AND o.ORD_STS_CD = '10'                     -- 주문상태: 정상
   AND o.DEL_YN = 'N'                          -- 논리삭제 제외
 GROUP BY c.CUST_NO
        , c.CUST_NM
HAVING SUM(o.ORD_AMT) > 0
 ORDER BY TOT_ORD_AMT DESC;
```

이 예시는 본 절의 규칙 전부를 응축한다 — 상단 메타데이터 주석(③), 예약어 대문자·절 단위 줄바꿈·leading comma(③), 전 컬럼 별칭 접두와 의미 있는 집계 별칭(②·④), ANSI 명시적 조인과 `ON`/`WHERE` 구분(④), 바인드 변수(⑤·⑥), sargable 조건(⑤), 업무 규칙을 밝히는 라인 주석(③). 코드 리뷰 시 이 예시를 기준 잣대로 사용한다.

CHAPTER 6

테크니컬 노트

6.1 개요

테크니컬 노트(Technical Notes)는 산출물 작성 시 실무에 즉시 적용할 수 있도록 항목별로 요약한, 구체적이고 전문적인 실무 지침서이다. 산출물 템플릿(4장)이 담아야 할 항목의 양식을 규정하고 가이드(5장)가 준수해야 할 원칙과 기준을 규정한다면, 테크니컬 노트는 각 항목을 어떻게 채우고 검증하는지에 관한 구체적인 작성·기술 방법을 항목 단위로 제시한다.

본 방법론의 서술은 프로세스(제2장), 표준 및 절차 매뉴얼(제3장), 가이드(제5장), 테크니컬 노트(제6장)의 순서로 점차 구체화되고 심화된다. 앞 장이 방향과 기준을 정의하면, 뒷 장은 이를 상충 없이 더 구체적인 형태로 보완한다. 이 흐름에서 테크니컬 노트는 가장 구체적인 최종 단계로서, 산출물 템플릿(4장)을 완전하게 작성하고 활용하도록 지원하는 것을 목적으로 한다.

세 문서는 다음과 같이 역할을 분담한다. 산출물 템플릿은 값을 담을 양식을, 가이드는 값이 준수해야 할 원칙과 기준을, 테크니컬 노트는 각 항목을 채우는 구체적인 방법과 기술적 절차를 제공한다. 작성자는 템플릿의 각 항목을 채우면서 본 장의 작성 요령을 참조하고, 판단이 갈리는 지점에서는 가이드의 기준으로 되돌아가 근거를 확인한다.

본 장은 각 항목의 정의를 설명하는 데 그치지 않고, 값을 채울 때 발생하는 실제 결정과 오류의 예방에 초점을 둔다. 각 절은 항목별 작성 요령을 표로 제시한 뒤, 판단이 갈리는 항목에는 작성 예시와 오류 대비(Bad → Good)를 제시하고, 끝으로 자동화와 품질 검증을 위한 구체적 기술 방법(생성 규칙, 검증 질의, 도구 활용)을 덧붙인다. 표기 규약으로 물리명과 데이터 타입은 코드체로 적고, 잘못된 예는 Bad, 권장하는 예는 Good으로 대비한다.

각 절의 「자동화와 품질 검증」이 제시하는 원리를 실제로 수행하는 도구는 KDSM 라이브러리(7장)에서 다룬다. 본 장이 무엇을 어떻게 검증해야 하는지를 규정한다면, 라이브러리는 그 검증을 실행하는 명령을 제공한다.

> **i 노트**
>
> 본 장의 작성 지침은 산출물 템플릿 파일의 각 시트 하단에 수록된 「**항목 정의 및 작성 지침**」과 동일한 기준을 따른다. 템플릿 파일(4.4.5절)과 본 장을 함께 사용하면, 파일에서 항목을 채우다가 막히는 지점을 본문에서 예시와 함께 확인할 수 있다.

6.2 테크니컬 노트 체계

6.2.1 산출물과 테크니컬 노트의 관계

데이터 표준 정의 단계는 **표준단어 → 표준용어 → 표준도메인 → 표준코드**의 순서로 산출물을 생성하며, 앞 산출물의 항목이 뒤 산출물의 입력이 된다. 단어가 모여 용어가 되고, 용어에 도메인이 적용되며, 코드성 용어는 표준코드로 상세화된다. 따라서 각 산출물의 항목 작성 요령도 이 의존 관계를 따라 연결된다. 본 장의 각 절은 하나의 산출물에 대응하며, 그 산출물의 **모든 항목을 빠짐없이** 항목별 작성 지침 표로 제시하고, 판단이 어려운 항목은 예시로 보강한다.

6.2.2 프로세스 산출물–테크니컬 노트 매핑

본 장은 데이터 표준 정의(Phase 2~5)의 4종 산출물과, 표준화 적용 단계의 데이터 모델 정의(논리·물리) 산출물을 함께 다룬다. 표준코드 정의서는 코드의 속성을 정의하는 **코드 마스터**와 개별 코드값을 정의하는 **코드값 상세**로 나뉘므로, 실제 작성 시트는 5종이다.

표 6.1. 데이터 표준 정의 산출물–테크니컬 노트 매핑

산출물(정의서)	대응 템플릿	작성 가이드(원칙)	테크니컬 노트(작성 요령)
표준단어 정의서	표 4.5	5.2절	6.3.1절
표준용어 정의서	표 4.7	5.3절	6.3.2절
표준도메인 정의서	표 4.9	5.4절	6.3.3절
표준코드 정의서(마스터·상세)	표 4.15, 표 4.16	5.5절	6.3.4절

데이터 모델 정의(표준화 적용) 단계는 논리 데이터 모델(주제영역·주제영역구성·엔터티·속성 정의서)과 물리 데이터 모델(데이터베이스·테이블·컬럼 정의서)을 산출한다. 각 산출물의 항목 작성 요령은 아래와 같이 연결된다.

표 6.2. 데이터 모델 정의 산출물–테크니컬 노트 매핑

산출물(정의서)	대응 템플릿	작성 가이드(원칙)	테크니컬 노트(작성 요령)
주제영역 정의서	표 4.20	5.7.2절	6.4.1.1절
주제영역구성 정의서	표 4.22	5.7.2절	6.4.1.1절
엔터티 정의서	표 4.24	5.7.3절	6.4.1.2절
속성 정의서	표 4.26	5.7.4절	6.4.1.3절
데이터베이스 정의서	표 4.28	5.8.1절	6.4.2.1절
테이블 정의서	표 4.29	5.8.2절	6.4.2.2절
컬럼 정의서	표 4.31	5.8.3절	6.4.2.3절

6.3 데이터 표준 정의 테크니컬 노트

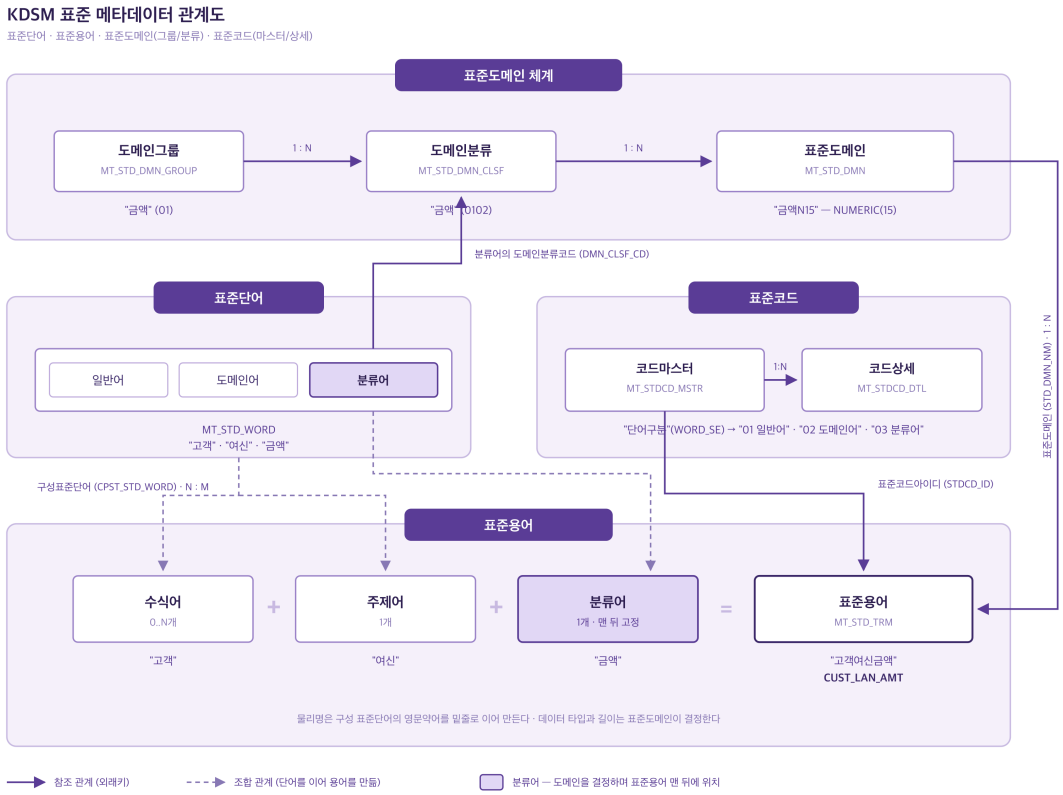

그림 6.1. KDSM 표준 메타데이터 관계도

KDSM은 데이터 표준과 관련된 메타데이터 테이블과 초기 데이터를 제공하는데, 메타데이터 관계도인 그림 6.1 이 데이터 표준은 표준단어·표준용어·표준도메인·표준코드의 관계를 잘 설명해 준다.

데이터 표준은 표준단어·표준용어·표준도메인·표준코드 네 가지로 이루어지며, 각각 별도의 메타데이터 테이블에 저장되어 외래키(참조 관계)와 조합 관계로 서로 맞물린다. 그림 6.1 는 이 네 표준의 메

타데이터 관계를 나타낸다. 개별 작성 요령은 이어지는 절(6.3.1절, 6.3.2절, 6.3.3절, 6.3.4절)에서 다루며, 여기서는 이들이 어떻게 연결되는지를 개괄한다.

표준단어(`MT_STD_WORD`)는 명칭의 최소 의미 단위로, 단어구분(`WORD_SE`)에 따라 일반어·도메인어·분류어로 나뉜다. 표준용어(`MT_STD_TRM`)는 이 단어들을 이어 만든 이름으로, 수식어(0~N개) + 주제어(1개) + 분류어(1개, 맨 뒤 고정)의 구조를 가진다. 용어와 단어는 구성표준단어(`CPST_STD_WORD`) 테이블을 매개로 N:M 조합 관계를 이루며, 용어의 물리명은 구성 단어들의 영문약어를 밑줄로 이어 만든다. 예를 들어 '고객'(수식어) + '여신'(주제어) + '금액'(분류어)은 표준용어 '고객여신금액'(`CUST_LAN_AMT`)이 된다.

표준도메인은 도메인그룹(`MT_STD_DMN_GROUP`) → 도메인분류(`MT_STD_DMN_CLSF`) → 표준도메인(`MT_STD_DMN`)의 1:N 계층으로 관리된다(6.3.3.1절, 6.3.3.2절). 용어의 데이터 형식은 맨 뒤 분류어가 결정한다. 분류어 단어가 도메인분류코드(`DMN_CLSF_CD`)로 도메인분류에 연결되고, 그 도메인분류에 속한 표준도메인이 데이터 타입·길이를 확정하기 때문이다. 따라서 표준용어는 표준도메인을 참조(`STD_DMN_NM`, 1:N)하여 자신의 형식을 갖는다. 앞의 예에서 분류어 '금액'은 도메인분류 '금액'(0102, 도메인그룹 '금액' 01의 하위)에 대응하고, 그 아래 표준도메인 '금액N15'(`NUMERIC(15)`)가 적용되어 '고객여신금액'은 `NUMERIC(15)`이 된다.

표준코드는 코드마스터(`MT_STDCD_MSTR`)와 코드상세(`MT_STDCD_DTL`)가 1:N으로 구성된다. 코드성 표준용어(분류어가 '코드'인 용어)는 표준코드아이디(`STDCD_ID`)로 코드마스터를 참조하여 허용값을 표준코드의 코드값으로 제한한다(5.5절). 그림에서 실선 화살표는 외래키 참조 관계를, 점선 화살표는 단어를 이어 용어를 만드는 조합 관계를 나타낸다.

6.3.1 표준단어 정의

표준단어 정의서는 명칭을 구성하는 최소 의미 단위인 단어의 표준을 기록하는 산출물이다. 작성 원칙과 규칙은 표준단어 작성 가이드(5.2절)를, 양식 항목은 표 4.5 를 함께 참조한다. 아래 지침은 각 항목을 실제로 채울 때의 요령이다.

표 6.3. 표준단어 정의서 항목별 작성 요령

항목	작성 요령
표준단어명	표준으로 확정한 한글 단어명을 기재한다. 명사 또는 명사형으로 적고, 하나의 단어는 하나의 의미만 갖도록 한다.
영문명	단어의 영문 전체명(Full Name)을 기재한다. 약어가 아니라 온전한 단어다(고객 → `CUSTOMER`, 번호 → `NUMBER`).
영문 약어	물리명 생성에 쓰이는 영문 약어명을 기재한다. 사전 전체에서 약어가 중복되지 않아야 하며, 2~6자를 표준 범위로 한다.
단어 설명	단어의 의미·정의를 기재한다. "~등", "또는"처럼 의미를 열어 두는 표현을 쓰지 않는다.
도메인 분류명	형식단어(분류어)인 경우 연결되는 표준도메인의 분류명을 기재한다. 일반어·도메인어는 비워 둔다.
이음동의어	소리는 다르나 의미가 같은 단어(연도/년도, 비율/율)를 표준단어명으로 수렴시켜 목록으로 기재한다. 이음동의어는 한글명만 관리한다.
금칙어	사용을 금지할 단어(화일→파일)를 기재한다. 금칙어는 정의 대상이 아니며 한글명만 관리한다.
단어 구분	단어의 역할을 **일반어 / 도메인어 / 분류어** 중에서 선택한다(아래 구분 기준 참조).
상태	후보 / 확정 / 폐기 중 현재 상태를 기재한다.

단어 구분의 판정 — 「단어 구분」은 단어가 용어 안에서 맡는 역할을 정하는 항목으로, 세 유형의 성격이 다르다.

표 6.4. 단어 구분 유형

구분	개념	용어 내 역할	예시
일반어	업무·일상에서 의미를 전달하는 일반 단어	주제어·수식어로 쓰여 데이터의 성격을 나타냄	고객, 사원, 주문, 상품, 주소
도메인어	특정 업무 영역에서 쓰이는 전문 용어	해당 도메인 안에서 명확한 의미를 갖는 주제어	거래처, 계약, 청구, 마일리지
분류어(형식단어)	데이터의 형식·타입을 결정하는 단어	용어의 **맨 마지막**에 위치해 값의 형식을 예고	일자, 금액, 여부, 코드, 명, 번호, 수, 율

분류어는 「도메인 분류명」을 반드시 채워 도메인그룹과 연결하고, 일반어·도메인어는 「도메인 분류명」을 비워 둔다. 분류어 여부가 이후 용어의 도메인 결정을 좌우하므로, 단어 등록 시점에 정확히 판정한다.

영문 약어 생성의 실무 절차 — 약어는 물리명의 "길이 예산"을 소모하므로 짧고 유일해야 한다. 다음 우선순위로 만들고, 상위 규칙의 결과가 이미 사전에 있으면 다음 규칙으로 내려간다.

표 6.5. 영문 약어 생성 규칙

순위	규칙	예시
1	업계 관용 약어를 그대로 사용	Number→ `NO`, Amount→ `AMT`, Date→ `DT`, Name→ `NM`, Code→ `CD`, Sequence→ `SEQ`, Management→ `MGMT`
2	첫 글자를 제외한 모음 제거 축약	Customer→ `CSTMR` →(길면) `CUST`
3	음절·구성어의 두문 조합	Standard Word→ `STD`, `WD`
4	앞부분 절단 + 충돌 회피 문자	위 방법이 모두 충돌할 때 앞 4~6자 절단

- **관용 국제 약어**(ID, IP, URL, GPS, ISBN)는 재약어화하지 않고 그대로 쓴다.
- **로마자 음차는 고유명사에만** 허용한다. 일반 개념어를 음차하면(고객을 `GOGAEK`으로) 물리명에서 의미를 복원할 수 없다.

> ⚠ 자주 하는 실수
>
> - **약어 중복**: `CUST`(고객)와 `CUST`(관리)가 충돌 → 하나는 `CUST`, 다른 하나는 `MGMT`로 분리한다. 신규 약어는 **약어→한글명 역인덱스**로 전수 중복 검사를 통과해야 한다.
> - **한 글자 단어의 파싱 모호**: '수수료율'이 수수료+율인지 수수+요율인지 갈린다. '수', '율', '명' 같은 한 글자 분류어는 결합 규칙을 사전에 명시한다.
> - **과분해/미분해**: '주민등록번호'를 주민+등록+번호로 쪼개면 제도 고유명이 깨진다(주민등록+번호가 옳다). 반대로 '고객명'을 한 단어로 두면 재사용이 막힌다(고객+명).

6.3.1.1 자동화와 품질 검증

표준단어 사전은 파일이 아니라 DB 테이블(`WORD_DICT`)로 운영할 때 생성과 검증을 자동화할 수 있다(4장 의 "표준사전은 DB형으로 운영"). 단어가 늘어도 품질이 흔들리지 않게 하는 핵심은 **약어를 수작업으로 만들지 않는 것**과 **검증을 상시 쿼리로 수행하는 것**이다.

- 생성 영문약어는 규칙 → 사전 조회 → 충돌 회피 순으로 반자동 생성한다. ① 동일 영문명이 사전에 있으면 그 약어를 재사용하고, ② 없으면 모음 제거·앞부분 절단 규칙(5.2.7절)으로 후보를 만들며, ③ 기존 약어와 충돌하면 순번·대체문자로 회피한다. 이 로직을 스프레드시트 함수나 스크립트로 구현하면 등록 시 약어를 자동 제안할 수 있다.
- 검증 확정 사전에 대해 약어 충돌(1약어 다단어)과 필수 항목 누락을 쿼리로 상시 탐지한다.

```sql
-- 약어 충돌: 하나의 영문약어에 서로 다른 표준단어가 매핑됨
SELECT ABBR, COUNT(DISTINCT WORD_NM) AS WORD_CNT
  FROM WORD_DICT
 WHERE STAT = '확정'
 GROUP BY ABBR
HAVING COUNT(DISTINCT WORD_NM) > 1;

-- 필수 누락: 확정 단어인데 영문명 또는 약어가 비어 있음
SELECT WORD_NM
  FROM WORD_DICT
 WHERE STAT = '확정'
   AND (ENG_NM IS NULL OR ABBR IS NULL);
```

- **[도구]** 사전 테이블을 형상관리·CI에 연동하면 변경(커밋)마다 위 검증을 자동 실행해 사전 품질을 조직 차원에서 균질화한다. 실패 건이 0이 될 때까지 병합을 막는다.

- **[도구]** KDSM 라이브러리의 `kdsm validate`가 위 검증을 규칙으로 구현하며, 미등록 낱말의 신규 표준단어 초안은 `kdsm suggest --coin`으로 얻는다(7.4.2절, 7.4.1절).

6.3.2 표준용어 정의

표준용어 정의서는 표준단어를 조합하여 업무에서 실제 사용하는 데이터 항목의 명칭을 기록한다. 작성 원칙은 표준용어 작성 가이드(5.3절), 양식 항목은 표 4.7 을 참조한다.

표 6.6. 표준용어 정의서 항목별 작성 요령

항목	작성 요령
표준용어명	엔터티·속성 등의 명명에 쓰일 용어를 한글로 기재한다. 특수문자·공백은 쓰지 않으며, 숫자와 관용 영문약어(URL, IP 등)는 최소한으로만 허용한다.
영문명	대문자 Full Name으로 기재한다. 예: "가공식품" → `PROCESSED FOOD`.
물리명	구성 단어의 영문약어를 등재 순서대로 `_`로 연결한 물리명을 기재한다. 대문자로 적고 특수문자를 피하며, 동의어이면 표준어와 동일한 약어를 쓴다.
용어 설명	용어의 업무적 의미를 상세히 서술한다. 필요 시 금칙어 목록을 함께 관리한다(휴대폰 표준어 ↔ 핸드폰·모바일폰 금칙어).
구성 표준단어	용어를 표준단어로 분해하여 `_`로 연결해 기재한다. **[수식어]+[주제어]+[분류어]** 순서를 지킨다(아래 참조).
대표 도메인	도메인 정의서에 정의된 도메인명을 기재한다(분류어가 결정).
표준코드명	해당 용어가 표준코드에 해당하면 연결되는 표준코드명을 기재한다. 코드성이 아니면 비워 둔다.
용어 구분	엔터티명 / 속성명 / 엔터티명·속성명 중 사용 영역을 기재한다.
상태	후보 / 확정 / 폐기 중 현재 상태를 기재한다.

「구성 표준단어」의 분해와 기재 — 용어 표준화의 핵심 항목이다. 용어를 최소 단위의 표준단어로 쪼개고 `_`로 연결하되, 다음 세 원칙을 지킨다.

1. **분해의 최소화**: 표준단어 사전에 존재하는 가장 작은 단위까지 쪼갠다. '고객명'은 하나가 아니라 고객 + 명으로 분리한다.
2. **논리적 순서 유지**: 수식하는 단어가 앞에, 피수식어가 뒤에 오도록 배열한다.
3. **분류어의 단일화**: 하나의 용어에는 분류어를 정확히 하나만, 맨 뒤에 결합한다. '주문금액코드'처럼 금액과 코드가 겹치면 안 된다.

표 6.7. 구성 표준단어 분해 예시

표준용어	구성 단어 분해	기재 형태
최초등록일자	최초(수식어) + 등록(주제어) + 일자(분류어)	`최초_등록_일자`
고객계좌번호	고객(주제어) + 계좌(도메인어) + 번호(분류어)	`고객_계좌_번호`
카드해지여부	카드(도메인어) + 해지(주제어) + 여부(분류어)	`카드_해지_여부`

물리명은 이 분해 결과의 각 단어 영문약어를 순서대로 `_`로 연결해 기계적으로 생성한다. 예: 고객_계좌_번호 → `CUST_ACCT_NO`. 수작업 조정을 허용하면 물리명에서 구성 단어를 복원하는 가역성이 깨진다.

물리명 길이와 DBMS 식별자 제한 — 물리명은 여러 약어를 `_`로 연결하므로 금세 길어진다. 채택 DBMS의 식별자 길이 상한을 초과하면 객체 생성 자체가 실패하므로, 용어 설계 시점에 예산을 관리한다.

표 6.8. 식별자 길이 제한 (Oracle · PostgreSQL)

DBMS	식별자 길이 상한	비고
Oracle	30바이트(12.1 이하) / 128바이트(12.2 이상)	레거시 연계가 있으면 보수적으로 30바이트 기준 설계
PostgreSQL	63바이트	초과분은 잘림(truncate)되어 이름 충돌 위험

두 DBMS 모두 물리명은 상한을 넘지 않도록 설계하되, Oracle 30바이트(구버전) 환경과 연계될 수 있으면 30바이트를 공통 기준으로 잡는 편이 안전하다. PostgreSQL은 미부여 식별자를 소문자로 접으므로(`ORD` → `ord`), 대문자 표준 물리명을 그대로 쓰되 큰따옴표 인용은 지양한다.

- **길이 예산**: 상한에서 이력·백업 등 접미어(`_HST` 4자)가 차지할 자리를 먼저 빼고, 남는 예산 안에서 단어 수와 약어 길이를 설계한다.
- **한글 바이트 주의**: 한글 물리명은 쓰지 않지만, 주석(comment)이나 일부 DBMS 설정에서 바이트 단위 제한이 걸리므로 인코딩(UTF-8은 한글 3바이트)을 감안한다.

> ⚠ 자주 하는 실수
>
> - **분류어 중복**: `주문_금액_코드`(Bad) — 금액과 코드가 함께 붙어 도메인이 모호하다. 금액이면 `주문_금액`, 상태 코드면 `주문_상태_코드`로 분리한다(Good).
> - **동의어 약어 불일치**: '해지'(표준어)와 '해약'(동의어)의 약어가 다르면 물리명이 갈라진다. 동의어는 표준어와 **동일 약어**를 쓴다.
> - **수작업 물리명 조정**: 보기 좋다고 `CUST_ACCT_NO`를 `CUSTOMER_ACCOUNT_NO`로 임의 확장하면 파싱 가역성이 깨지고 다른 용어와 규칙이 어긋난다.

6.3.2.1 자동화와 품질 검증

용어의 물리명은 구성 표준단어 약어를 순서대로 잇는 **기계적 산물**이므로(6.3.2절), 수작업으로 적지 말고 사전과의 조인으로 생성·검증한다. 용어의 구성 단어를 순서 있는 매핑(`TERM_WORD(용어, 순번, 단어)`)으로 관리하는 것이 전제다.

- 생성 구성 단어의 약어를 순번대로 `_`로 결합해 물리명을 자동 생성한다.

```sql
-- 구성 단어 약어를 순번대로 이어 물리명 생성
--   PostgreSQL: STRING_AGG(...)  ·  Oracle: LISTAGG(w.ABBR, '_') WITHIN GROUP (ORDER BY tw.SEQ)
SELECT tw.TERM_NM,
       STRING_AGG(w.ABBR, '_' ORDER BY tw.SEQ) AS GEN_PHYS_NM
  FROM TERM_WORD tw
  JOIN WORD_DICT w ON w.WORD_NM = tw.WORD_NM
 GROUP BY tw.TERM_NM;
```

- 검증 등록된 「물리명」이 생성 규칙과 다르면(수작업 조정) 오류다. 위 생성 결과와 등록 물리명을 대조해 불일치를 탐지하고, 마지막 구성 단어가 분류어가 아닌 용어(분류어 누락·중복)와 물리명 길이 초과도 함께 점검한다.
- 검증 같은 의미의 용어가 다른 이름으로 중복 등록되지 않았는지(동의어 미정리), 구성 단어가 모두 확정 사전에 존재하는지 조인으로 확인한다.
- **[도구]** 모델링 도구의 **용어 사전 연동**을 켜면 속성명 입력을 표준용어 선택으로 강제하고 물리명을 자동 채워, 사람이 물리명을 직접 입력하는 경로 자체를 없앤다.
- **[도구]** 한글 용어명을 표준단어로 분해하여 물리명·도메인까지 후보로 제시하는 작업은 `kdsm suggest` 가 수행한다(7.4.1절).

6.3.3 표준도메인 정의

표준도메인 정의서는 데이터 값의 타입·길이·형식·허용 범위를 정의한다. 작성 원칙은 표준도메인 작성 가이드(5.4절), 양식 항목은 표 4.9 을 참조한다.

표 6.9. 표준도메인 정의서 항목별 작성 요령

항목	작성 요령
표준도메인명	“도메인분류명 + 데이터 타입 + 길이(선택)” 형태로 정의한다. 예: 연도C4.
도메인 그룹	도메인의 상위 구분을 기재한다. 예: 날짜/시간. 「별첨_공통표준도메인그룹명목록」 참조.
도메인 분류	도메인의 세부 구분을 기재한다. 예: 연월. 「별첨_공통표준도메인분류명 목록」 참조.
데이터 유형	논리 데이터 타입을 기재한다. 예: 문자형(CHAR·VARCHAR), 숫자형(NUMERIC), 날짜형(DATETIME).
길이	전체 자릿수를 기재하며, 소수점이 있으면 소수 자리까지 포함한다. 바이트 단위이며 한글은 2바이트 기준이다.
소수 자릿수	소수점 이하 최대 자릿수를 기재한다(숫자형이고 소수 값이 있을 때만).
저장 형식	DBMS에 **저장**되는 형식을 기재한다. 예: 1999년 9월 → `YYYYMM` (199909).
표현 형식	데이터를 **제공·표시**할 때의 형식을 기재한다. 예: 1999년 9월 → `YYYY-MM` (1999-09).
단위	값을 측정하는 단위를 기재한다. 예: m, kg, 도.
허용 값 범위	최대·최소값이나 유효값을 기재한다. 예: 월 01~12, 여부 ‘Y’·‘N’.
물리 데이터 타입	DBMS가 실제 인식하는 타입·길이를 `타입(길이)` 또는 `타입(전체, 소수)`로 기재한다. 예: `VARCHAR(100)`, `NUMERIC(15,2)`, `CHAR(1)`.
상태	후보 / 확정 / 폐기 중 현재 상태를 기재한다.

「저장 형식」과 「표현 형식」의 구별 — 두 항목은 자주 혼동되지만 목적이 다르다. 저장 형식은 DB에 물리적으로 담기는 형태(`YYYYMM`)이고, 표현 형식은 화면·보고서에 보여 줄 때의 형태(`YYYY-MM`)이다. 하나의 도메인이 저장은 `YYYYMMDD`로, 표현은 `YYYY-MM-DD`로 갈릴 수 있으므로 두 항목을 분리해 관리한다. 저장 형식을 전사에서 단일화하지 않으면 같은 성격의 값이 테이블마다 다른 형태로 쌓여 정합성이 무너진다.

논리 데이터 유형 → DBMS 물리 타입 매핑 — 「데이터 유형」은 논리 타입으로 적고, 「물리 데이터 타입」은 채택 DBMS의 실제 타입으로 확정한다. DBMS별 대표 매핑은 다음과 같다.

표 6.10. 논리 유형–물리 타입 매핑 (Oracle · PostgreSQL)

논리 유형	Oracle	PostgreSQL
문자(가변)	`VARCHAR2(n)`	`VARCHAR(n)`
문자(고정)	`CHAR(n)`	`CHAR(n)`
정수	`NUMBER(p)`	`INTEGER` / `BIGINT`
소수(정밀)	`NUMBER(p,s)`	`NUMERIC(p,s)`
날짜	`DATE`	`DATE`
일시	`TIMESTAMP`	`TIMESTAMP`

Oracle의 `DATE`는 시·분·초까지 포함하지만, PostgreSQL은 `DATE`(일자)와 `TIMESTAMP`(일시)를 분리한다. 시각이 필요하면 Oracle은 `DATE`/`TIMESTAMP`, PostgreSQL은 `TIMESTAMP`를 쓴다. 정수는 Oracle이 `NUMBER(p)` 하나로 다루는 반면 PostgreSQL은 크기에 따라 `INTEGER`(약 21억)와 `BIGINT`를 구분한다.

- **가변 vs 고정**: 길이 편차가 큰 명칭·설명류는 `VARCHAR` 계열을, 자릿수가 고정된 코드·여부는 `CHAR` 계열을 쓴다. 고정 길이 문자열을 `VARCHAR`에 담아도 무방하나, `CHAR`에 가변 문자열을 담으면 뒤가 공백으로 채워져 비교 오류를 부른다.
- **금액·율의 정밀도**: `FLOAT`·`REAL`·`DOUBLE`(이진 부동소수점)은 0.1도 정확히 표현하지 못하므로 **금액·율에 금지**한다. 반드시 고정소수점(`NUMBER`/`NUMERIC`/`DECIMAL`)을 쓴다. 율은 저장 단위(3.5%를 `3.5`로 vs `0.035`로)를 도메인에 명시하고 전사 통일한다.
- **길이 단위(byte vs char)**: Oracle은 `VARCHAR2(100)`이 기본 바이트 기준(`NLS_LENGTH_SEMANTICS`)이라 UTF-8 한글이 컬럼당 3바이트를 차지한다 — 한글 100자를 담으려면 `VARCHAR2(100 CHAR)`로 지정한다. 반면 PostgreSQL의 `VARCHAR(100)`은 문자 기준이므로 한글 100자를 그대로 담는다.
- **암호화 컬럼 길이**: 주민등록번호·카드번호처럼 암호화 대상 도메인은 **암호문 출력 길이**로 물리 길이를 산정한다(평문 13자라도 암호문은 수십~수백 바이트).

표준 도메인 예시 — 실제로 자주 쓰이는 도메인의 확정 형태다.

표 6.11. 표준 도메인 확정 예시

도메인분류	데이터 유형	물리 타입	저장 형식/허용값	기본값
여부	문자(고정)	`CHAR(1)`	`'Y'` / `'N'`	`'N'`
코드	문자(가변)	`VARCHAR(n)`	표준코드 사전의 값만 허용	—
금액(원화)	숫자	`NUMBER(15)`	정수 원화	`0`
금액(정밀)	숫자(소수)	`NUMBER(18,4)`	외화·단가·이자	`0`
율	숫자(소수)	`NUMBER(18,6)`	백분율 단위(3.5% → 3.5)	`0`
일자	문자(고정)	`CHAR(8)`	`YYYYMMDD`, 무한일 `'99991231'`	—
일시	일시	`TIMESTAMP`	시스템 기록용(UTC 저장 권장)	—
명	문자(가변)	`VARCHAR(100)`	명칭 문자열	—

위 「물리 타입」은 Oracle 표기를 대표로 적었다. PostgreSQL은 `NUMBER`를 `NUMERIC`(정수는 `INTEGER`/`BIGINT`)으로 치환하며(표 6.10), 문자·날짜 타입과 나머지 표준(길이·형식·허용값·기본값)은 동일하다.

> ⚠ 자주 하는 실수
>
> - **날짜를 문자·날짜형 혼용**: 절반은 `CHAR(8)`, 절반은 `DATE`로 저장하면 조인·비교 때마다 형변환이 발생한다. 전사 단일 방식으로 통일한다. `CHAR(8)`을 택하면 `'20260231'` 같은 불가능 날짜를 막을 CHECK 제약이나 공통 검증을 함께 둔다.
> - **번호를 숫자형에 저장**: 계좌번호 `'0012345'`를 숫자형에 넣으면 선행 0이 사라진다. 번호·코드류는 겉보기에 숫자여도 **문자 타입**이다.
> - **길이 세분화 남발**: `명_V50`, `명_V60`, `명_V80`을 남발하면 도메인이 컬럼별 정의로 퇴화한다. 명칭류는 50/100/300처럼 **대표 길이 몇 단계**로 수렴시킨다.

6.3.3.1 도메인 그룹 정의

도메인 그룹은 성격이 유사한 도메인을 묶은 **최상위 범주**로, 도메인 체계의 **도메인그룹 → 도메인분류 → 도메인** 3계층 중 맨 위에 위치한다(5.4.1절). 용어의 분류어가 도메인그룹과 연결되므로(5.3.2절), 그룹 체계는 분류어 설계와 함께 확정한다. 「표준도메인정의서」의 **도메인 그룹** 시트에 정의하며, 공공부문은 행정안전부 「공통표준도메인그룹명」 체계를 준용한다.

표 6.12. 도메인 그룹 정의서 항목별 작성 요령

항목	작성 요령
도메인 그룹명	도메인을 그룹핑하는 상위 구분명(금액·날짜/시간·번호·여부·코드 등)을 기재한다. 공공 부문은 「공통표준도메인그룹명」을 준용한다.
도메인 그룹 설명	그룹의 의미와 포함 범위를 기재한다.
상태	후보 / 확정 / 폐기 중 현재 상태를 기재한다.

표 6.13. 도메인 그룹 정의 예시

도메인 그룹명	도메인 그룹 설명	상태
금액	화폐의 가치를 수로 표현하기 위한 도메인	확정
날짜/시간	특정 사건이 일어난 시점, 또는 시점과 시점 간의 정해진 기간을 표현하기 위한 도메인	확정
내용	서술 형식으로 상세 내용을 표현하는 자유 형식 텍스트 도메인. 부가 설명·의견·사유 등을 저장할 때 사용한다	확정
명칭	문자 형식으로 객체를 식별하기 위한 도메인(명·성명·명칭·상호 포함). 아이디와 주소도 이 그룹에 소속시킨다	확정
번호	일정한 체계를 갖거나 특정 자리에 의미가 있는 식별 번호. 무의미하게 연속 부여되는 번호도 포함한다	확정
수량	객체의 개수나 양을 수로 표현하기 위한 도메인. 일반적인 측량 단위도 포함한다	확정
율	다른 수나 양에 대한 어떤 수나 양의 비를 백분율이나 소수 형태로 표현하기 위한 도메인	확정
코드	데이터를 분류·구분하거나 긍정(Y)/부정(N), 유(有)/무(無)로 구분하기 위한 도메인	확정

- **[필수]** 그룹명은 전사에서 유일해야 하며, 하나의 분류어는 하나의 도메인그룹에 대응하도록 맞춘다('명' → 명칭 그룹, '금액' → 금액 그룹).
- **[권장]** 신규 그룹을 만들기 전에 기존 그룹으로 수용 가능한지 먼저 검토한다. 그룹이 과도하게 늘면 상위 구분으로서의 기능이 약해진다.

6.3.3.2 도메인 분류 정의

도메인 분류는 도메인그룹을 더 세분한 **중간 계층**으로, 각 분류는 반드시 하나의 상위 그룹에 소속된다. 표준도메인명이 "**도메인분류명 + 데이터 타입 + 길이**"(예: `가격N10`) 형태이므로, 분류명은 곧 도메인명의 어간이 된다. 「표준도메인정의서」의 **도메인 분류** 시트에 정의하며, 공공 부문은 행정안전부 「공통표준도메인분류명」 체계를 준용한다.

표 6.14. 도메인 분류 정의서 항목별 작성 요령

항목	작성 요령
도메인 분류명	그룹 하위의 세부 구분명(가격·비용·연월·우편번호 등)을 기재한다. 공공 부문은 「공통표준도메인분류명」을 준용한다.
도메인 분류 설명	분류의 의미를 기재한다.
도메인 그룹 이름	분류가 소속되는 상위 도메인 그룹명을 기재한다(필수).
상태	후보 / 확정 / 폐기 중 현재 상태를 기재한다.

표 6.15. 도메인 분류 정의 예시

도메인 분류명	도메인 분류 설명	도메인 그룹 이름	상태
가격	물건이 지니고 있는 가치를 돈으로 나타낸 것	금액	확정
비용	어떤 일을 하는 데 드는 돈	금액	확정
연월	특정한 연(年), 월(01~12)을 정하여 표현한 것	날짜/시간	확정
연월일	특정한 날짜를 연·월·일로 정하여 표현한 것	날짜/시간	확정
명	사물·단체·현상을 다른 것과 구별하기 위해 붙인 이름	명칭	확정
우편번호	우편물 분류를 위해 지역마다 매긴 번호	번호	확정
일련번호	일률적으로 연속되어 있는 번호	번호	확정
수	셀 수 있는 사물을 세어서 나타낸 값	수량	확정
여부	그러함과 그러하지 아니함을 구분하는 코드	코드	확정

- **[필수]** 모든 분류는 「도메인 그룹 이름」으로 상위 그룹에 연결한다. 소속 그룹이 없는 분류는 존재할 수 없다.
- **[권장]** 분류는 도메인의 타입·길이를 결정하는 연결고리다. 같은 분류에 속한 도메인은 데이터 유형이 일관되도록 설계한다(가격·비용 → 금액 그룹 → 숫자형).

도메인 그룹·도메인 분류 시트에서 확정한 값은 「표준도메인 정의서」의 「도메인 그룹」「도메인 분류」 항목(표 6.9)에 입력되어 개별 도메인과 연결된다.

6.3.3.3 자동화와 품질 검증

도메인의 가치는 "모든 컬럼이 도메인을 벗어나지 않음"을 **실제 DB에서 확인**할 때 실현된다. DB 카탈로그(PostgreSQL `information_schema`, Oracle `ALL_TAB_COLUMNS`)를 도메인 정의와 대조해 위반을 자동 탐지한다.

- 검증 도메인 정의(`DOMAIN_DEF`)와 컬럼–도메인 매핑(`COL_DOMAIN`)을 실제 컬럼과 조인해 타입·길이 불일치를 찾는다.

```
-- PostgreSQL: 매핑된 도메인과 실제 컬럼의 타입·길이 불일치 탐지
SELECT c.table_name, c.column_name,
       c.data_type, c.character_maximum_length,
       d.PHYS_TYPE, d.LEN
  FROM information_schema.columns c
  JOIN COL_DOMAIN m ON m.TBL = c.table_name AND m.COL = c.column_name
  JOIN DOMAIN_DEF d ON d.DOMAIN_NM = m.DOMAIN_NM
 WHERE c.data_type <> d.PHYS_TYPE
    OR COALESCE(c.character_maximum_length, 0) <> COALESCE(d.LEN, 0);
-- Oracle은 ALL_TAB_COLUMNS 의 DATA_TYPE·DATA_LENGTH·DATA_PRECISION 로 동일 점검
```

- 검증 **도메인 미지정 컬럼**(카탈로그엔 있으나 `COL_DOMAIN`에 없음)을 좌외부조인으로 탐지해 "표준을 우회한 컬럼"을 0으로 수렴시킨다. 이 미지정 비율은 표준 준수율의 핵심 지표(KPI)다.
- 생성 도메인 정의 하나에서 DBMS별 DDL 타입 조각을 함께 생성한다(표 6.10). `NUMBER` ↔ `NUMERIC` 같은 치환을 매핑 테이블로 두면 Oracle·PostgreSQL DDL을 한 정의에서 뽑는다.

- **[도구]** 도메인그룹·분류·도메인의 정합성과 컬럼의 도메인 준수 여부는 `kdsm validate`가 검사하고, 운영 데이터베이스의 도메인 준수 수준은 `kdsm diagnose`가 측정한다(7.4.2절, 7.6.2절).

6.3.4 표준코드 정의

표준코드 정의서는 한정된 값의 집합으로 관리하는 코드를 기록하며, 코드의 속성을 정의하는 **코드 마스터**와 개별 코드값을 정의하는 **코드값 상세**로 구성된다. 작성 원칙은 표준코드 작성 가이드(5.5절), 양식 항목은 표 4.15·표 4.16 를 참조한다.

마스터와 상세는 「**코드 ID**」로 연결된다. 마스터 한 건에 상세 여러 건이 딸리는 1:N 관계이므로, 두 시트의 코드 ID가 정확히 일치해야 한다.

코드 마스터 항목별 작성 요령

표 6.16. 표준코드 정의서(코드 마스터) 항목별 작성 요령

항목	작성 요령
표준코드명	표준용어를 준수하여 부여한 코드의 한글명을 기재한다.
코드 ID	표준용어의 물리명을 준수하여 코드를 식별하는 영문 코드명을 기재한다. 예: `ORD_STAT_CD`.
코드 유형	코드값을 표현하는 데이터 타입을 기재한다. 예: 문자형 `CHAR`, 숫자형 `NUMERIC`·`INTEGER`.
자릿수	코드값을 표현하는 최대 길이를 숫자로 기재한다.
기본값	코드값의 기본값을 기재한다(없으면 비움).
코드 설명	코드의 적용 범위·의미·예외사항을 기재한다.
상태	후보 / 확정 / 폐기 중 현재 상태를 기재한다.

코드값 상세 항목별 작성 요령

표 6.17. 표준코드 정의서(코드값 상세) 항목별 작성 요령

항목	작성 요령
코드 ID	이 코드값이 속한 마스터의 코드 ID를 기재한다(마스터와 동일 값).
코드값	실제 저장되는 허용 코드값을 기재한다.
코드값 명	코드값의 한글 의미를 기재한다. 예: 코드명 가족관계, 코드값 001, 코드값 명 본인.
코드값 설명	코드값 명에 대한 구체적 설명을 기재한다.
정렬 순서	사용·저장 시의 코드값 정렬 순서를 기재한다.
사용 여부	개별 코드값의 사용 / 미사용을 기재한다.

작성 예시 — 표준코드 “주문상태코드”(코드 ID: `ORD_STAT_CD`)의 마스터 한 건에 아래 상세가 연결된다.

표 6.18. 코드값 상세 작성 예시

코드 ID	코드값	코드값 명	정렬 순서	사용 여부
`ORD_STAT_CD`	01	접수	1	사용
`ORD_STAT_CD`	02	배송중	2	사용
`ORD_STAT_CD`	03	완료	3	사용
`ORD_STAT_CD`	09	취소	4	사용

코드값 부여의 실무 규칙

- **무의미(순차) 코드를 기본**으로 한다. `01`, `02`, `03`처럼 의미 없이 부여하면 분류 체계가 바뀌어도 값을 재해석할 필요가 없다. `M`=남성처럼 의미를 연상시키는 유의미 코드는 성별처럼 불변에 가깝고 국제 관행이 확립된 극소수에만 쓴다.
- **자릿수는 한 단계 여유**를 둔다. 2자리면 최대 100개까지 담긴다. 자릿수가 부족해 전체 값을 재부여하는 상황은 심각한 문제를 초래한다.
- **정렬 의도는 코드값이 아니라 「정렬 순서」**로 표현한다. 표시 순서가 값 크기와 무관하면 `09`(취소)를 맨 뒤에 두더라도 정렬 순서로 제어한다.
- **포괄을 위해 '기타(90)'**를 둘 수 있으나, '기타'가 과도하게 쌓이면 분류 체계를 재검토한다(MECE 위반 신호).

> ⚠ 자주 하는 실수
>
> - **운영 중 코드값 의미 변경(Bad)**: `30 = 골드`를 나중에 `30 = 해지`로 바꾸면 과거 데이터의 `30`이 소급 오염된다. Good: 기존 값은 「사용 여부 = 미사용」으로 폐기(값은 보존)하고, 새 의미는 새 코드값(`99 = 해지`)으로 부여한다.
> - **여부를 코드로 남발**: 참/거짓 2치면 상태코드가 아니라 여부 도메인(`CHAR(1)` `'Y'`/`'N'`)이다. 3치 이상으로 확장될 가능성이 있을 때만 상태코드로 설계한다.
> - **공통코드 자체 신설**: 국가·행정구역·산업분류처럼 법정·국가 표준 코드체계가 있으면 준용한다. 자체 코드를 새로 만들면 대외 연계 시 매핑 비용이 발생한다.

6.3.4.1 자동화와 품질 검증

코드 마스터·코드값 상세는 그대로 **코드 테이블의 DDL·데이터**로 전개되고, 실데이터가 코드값 집합을 벗어나지 않는지 상시 검증할 수 있다.

- 생성 「코드 마스터」에서 코드 테이블 DDL을, 「코드값 상세」에서 `INSERT`를 자동 생성한다.

```
-- 코드값 상세로부터 전개한 코드 테이블(예: 주문상태코드)
CREATE TABLE CD_ORD_STAT (
    CD        VARCHAR2(2)    NOT NULL,
    CD_NM     VARCHAR2(100)  NOT NULL,
    SORT_ORD  NUMBER(3),
    USE_YN    CHAR(1)        DEFAULT 'Y' NOT NULL,
    CONSTRAINT PK_CD_ORD_STAT PRIMARY KEY (CD)
);
```

- 검증 업무 테이블의 코드 컬럼 값 중 코드 집합에 없는 **고아 코드**를 탐지한다. 이 쿼리가 0건이어야 코드 표준이 데이터에서 지켜진 것이다.

```sql
-- 고아 코드: 주문의 상태코드가 코드 마스터에 없음
SELECT DISTINCT o.ORD_STAT_CD
  FROM ORD o
  LEFT JOIN CD_ORD_STAT c ON c.CD = o.ORD_STAT_CD
 WHERE c.CD IS NULL;
```

- 검증 반대로 데이터에 한 번도 나타나지 않은 **미사용 코드**도 주기적으로 뽑아 폐기·정리 후보로 삼는다.
- **[도구]** 코드성 컬럼에는 코드 테이블 FK 또는 `CHECK` 제약을 자동 부여해(6.4.1.3.2절) 위반을 입력 시점에 원천 차단한다. 사후 검증보다 사전 차단이 품질 균질화에 우선한다.

6.4 데이터 모델 정의 테크니컬 노트

데이터 표준 정의가 끝나면 표준을 실제 모델에 적용하여 **논리 데이터 모델**(엔터티·속성)과 **물리 데이터 모델**(데이터베이스·테이블·컬럼)을 산출하고, 이를 ERD로 시각화한 뒤 DDL로 구현한다. 본 절은 이 흐름을 **논리 데이터 모델링 → 물리 데이터 모델링 → ERD 작성 → DDL 생성**의 순서로 나누어, 각 산출물의 항목을 실제로 채우고 구현으로 옮길 때의 요령을 다룬다. 두 모델링 절 사이에서 갈라지는 지점은 「논리와 물리 모델의 차이」(6.4.3절)에 따로 모았다. 각 정의서의 작성 원칙은 명명규칙(5.6절)을, 양식 항목은 4장 의 각 정의서 항목표를 함께 참조한다.

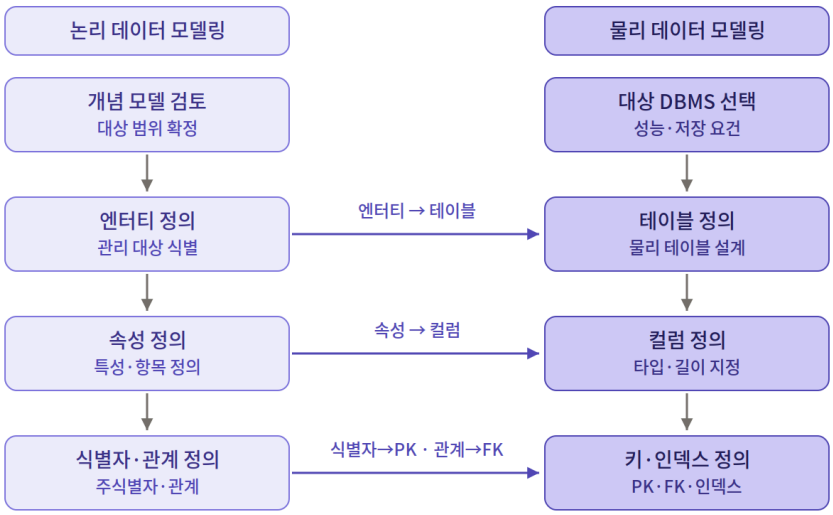

그림 6.2. 논리–물리 데이터 모델 변환(매핑)

물리 데이터 모델은 논리 데이터 모델을 채택 DBMS의 객체로 **변환(매핑)**한 결과다. 그림 6.2 은 논리 모델링과 물리 모델링의 절차를 좌우로 나란히 놓고, 논리 산출물이 어떤 물리 산출물로 이어지는지를 매핑 화살표로 보인다. 물리 모델링의 첫 단계인 대상 DBMS 선택은 논리 측에 대응물이 없는 물리 고유의 선행 작업으로, 이후 변환의 타입 표기·저장 구조 기준을 정한다.

변환의 핵심은 세 갈래다. 첫째, **엔터티는 테이블로** 옮겨진다. 논리 엔터티가 물리 테이블의 골격이 되며, 슈퍼타입/서브타입의 물리 전환 방식도 이때 확정한다. 둘째, **속성은 컬럼으로** 옮겨지고, 각 속성에 부여된 표준도메인이 컬럼의 데이터 타입·길이를 결정한다. 셋째, **식별자는 기본키(PK)로, 관계는 외래키(FK)로** 구현되며, 조인·참조 성능에 필요한 인덱스를 함께 정의한다.

이 변환은 반드시 **논리 모델을 확정한 뒤** 수행한다. 명칭·도메인·키·참조 같은 표준은 그대로 승계하고, 타입 표기·저장 구조·성능·무결성만 DBMS에 맞추어 구체화한다. 각 단계의 상세 요령은 논리 데이터 모델링(6.4.1절)과 물리 데이터 모델링(6.4.2절)에서 다룬다. 다만 이 1:1 대응은 성능과 저장 효율을 위해 의도적으로 깨지기도 하며, 그 경우는 논리와 물리 모델의 차이(6.4.3절)에서 다룬다.

6.4.1 논리 데이터 모델링

논리 데이터 모델링은 업무 요건을 **DBMS와 무관하게** 주제영역·엔터티·속성·관계로 구조화하는 단계다. 무엇을(엔터티) 어떤 성질로(속성) 어떻게 연결하여(관계) 관리할지를 표준용어·표준도메인 위에서 정의한다. 성능·저장 같은 물리 관심사는 이 단계에서 배제하고 업무 규칙의 정확한 표현에 집중하며, 정규화(주로 3NF)를 통해 이상현상(갱신·삽입·삭제 anomaly)을 제거한다.

6.4.1.1 주제영역 정의

주제영역은 모델을 읽고 나누어 맡는 구획이므로, 정의서는 **구획의 선언**(주제영역 정의서)과 **구획의 내용**(주제영역구성 정의서) 둘로 나뉜다. 분할 원칙은 가이드(5.7.2절)를 따르고, 여기서는 두 정의서의 항목을 채우는 요령을 다룬다.

표 6.19. 주제영역 정의서 항목별 작성 요령

항목	작성 요령
주제영역명	「표준용어정의서」에 등록된 용어로 한글 기재한다. 하나의 논리 데이터 모델 안에서 유일해야 한다.
주제영역 순번	정의서에 싣는 차례를 1부터 빈틈없이 부여한다. 업무 흐름을 따르는 것이 일반적이다.
주제영역 설명	주제영역의 업무적 범위를 기재한다. 어떤 엔터티가 이 영역에 들고 어떤 것이 들지 않는지 가릴 수 있도록 쓴다.

표 6.20. 주제영역구성 정의서 항목별 작성 요령

항목	작성 요령
소속 주제영역	「주제영역 정의서」에 등재한 이름을 그대로 기재한다. 표기가 한 글자라도 다르면 배정이 연결되지 않는다.
구성 순서	한 주제영역 안에서 배정 대상을 늘어놓는 차례를 1부터 부여한다.
구성대상	논리 모델의 것이면 `엔터티`, 물리 모델의 것이면 `테이블`로 기재한다.
구성대상 이름	엔터티는 엔터티명을, 테이블은 테이블 물리명을 기재한다. 같은 산출물에 정의된 대상이어야 한다.
사용 부서	해당 대상을 주로 사용하는 부서명을 기재한다. 선택 항목이므로 부서 개념이 없으면 비워 둔다.

구성대상 구분의 쓰임 — 하나의 양식으로 논리와 물리를 함께 담기 위해 `구성대상` 항목을 둔다. 논리 모델 산출물에서는 `엔터티`만 나타나고, 물리 모델로 전환한 뒤에는 같은 구획을 `테이블`로 다시 기재하여 논리 구획이 물리로 그대로 이어졌음을 보인다. 두 값을 한 시트에 섞어 적으면 어느 모델의 구획인지 판별할 수 없으므로, 모델 단계별로 나누어 관리한다.

자동화와 품질 검증

- 검증 주제영역구성 정의서의 `구성대상 이름`을 엔터티 정의서와 대조하여 **미배정 엔터티**(어느 주제영역에도 없음)와 **유령 배정**(정의되지 않은 엔터티를 배정)을 탐지한다. 둘 다 0건이어야 구획이 완결된 것이다.
- 검증 같은 엔터티가 둘 이상의 주제영역에 배정된 **중복 배정**을 탐지한다. 공통 참조 엔터티라면 공통 주제영역 하나로 옮긴다.
- 검증 `소속 주제영역` 값이 주제영역 정의서에 없는 이름인지, 주제영역 순번·구성 순서에 빠진 번호나 중복이 없는지 점검한다.
- **[도구]** ERD를 주제영역 단위로 분할해 출력하면, 주제영역을 넘는 관계선이 한눈에 드러난다. 그 선이 지나치게 많으면 분할 기준을 다시 볼 신호다(6.4.4절).

6.4.1.2 엔터티 정의

표 6.21. 엔터티 정의서 항목별 작성 요령

항목	작성 요령
엔터티명	「표준용어정의서」에 등록된 용어로 **단수형** 한글 기재한다. 서브타입 엔터티는 서브타입 이름만 기재한다.
영문명	「표준용어정의서」에 등록된 영문 용어로 기재한다.
엔터티 설명	엔터티의 목적, 포함/불포함 대상 예시, 관련 업무 규칙을 기재한다.
식별자(PK)	집합의 유일성을 나타내는 속성(또는 속성 그룹)을 기재한다. 복수 구성이면 `+`로 연결한다.
보조 식별자(UK)	주 식별자 외에 엔터티를 고유하게 구별하는 속성 조합을 기재한다. 복수 구성이면 `+`로 연결하고, 접두어를 붙여 `UKn:속성명` 형식으로 표기한다. 보조 식별자가 둘 이상이면 쉼표(`,`)로 구분한다.
엔터티 구분	기본(Fundamental)·중심(Main)·행위(Action) 등 데이터 모델링 관점의 분류를 기재한다.
상위 엔터티	서브타입 엔터티인 경우에 한하여 상위 엔터티명을 기재한다.
개인정보 포함여부	엔터티 속성에 개인정보가 포함되면 `Y`, 아니면 `N`으로 표기한다.
주요 관계	연관 엔터티와의 부모–자식 관계 유형·차수를 「상대 엔터티명(기준 차수:상대 차수)」 형태로 기재한다.

엔터티 구분의 판정 — 데이터 발생 패턴으로 구분하면 이후 인덱싱·이력 전략이 자연스럽게 결정된다.

표 6.22. 엔터티 구분 유형

구분	성격	발생 패턴	예시
기본(Fundamental)	업무의 주체·대상	개시 시 등록, 저빈도 변경	고객, 상품, 조직
중심(Main)	기본 엔터티 간의 핵심 업무 개념	업무 발생 시 생성	계약, 주문
행위(Action)	업무 행위의 발생 기록	행위마다 1건, 불변 지향	주문상세내역, 입금내역

- **식별자(PK) 기재**: 단일이면 속성명 하나, 복합이면 `고객번호 + 상품번호`처럼 `+`로 연결한다. 본질키(업무 식별자)가 길거나 불안정하면 인조키(surrogate, `주문일련번호`)를 PK로 두고 본질키는 유니크 제약으로 보존하는 방식도 검토한다.
- **주요 관계 표기 예시**: 고객 1명이 여러 주문을 가지면, 주문 엔터티의 주요 관계는 `고객(1:N)`으로 적는다. M:N은 교차 엔터티로 해소한 뒤 각각 1:N으로 표기한다.
- **서브타입·정규화**: 배타적 하위 유형(개인고객·법인고객)은 슈퍼타입/서브타입으로 표현하고, 물리 전환 시 단일 테이블(Type)·개별 테이블(Rolled-down)·통합 중 하나로 선택한다. 논리 단계에서는 3NF를 지향하되, 이후 물리 단계에서 성능을 위한 반정규화는 근거와 함께 별도로 판단한다.

자동화와 품질 검증

- 검증 논리 모델을 리버스 엔지니어링하거나 정의서 리포지토리를 질의해, 식별자(PK)가 없는 엔터티·주요 관계가 정의되지 않은 엔터티를 탐지한다. 식별자 없는 엔터티는 물리 전환이 불가능하므로 완결의 선결 조건이다.
- 검증 **다중값 혐의**를 명명 패턴으로 자동 탐지한다. 같은 엔터티에 `연락처1`·`연락처2`·`연락처3`처럼 순번만 다른 반복 속성이 있으면 별도 엔터티 분리 대상으로 넘긴다.
- **[도구]** 슈퍼타입/서브타입은 모델링 도구의 상속 기능으로 정의하고, 물리 전환 방식(단일·개별·통합)을 도구 옵션으로 일괄 적용해 사람마다 다른 전환을 막는다.

6.4.1.3 속성 정의

표 6.23. 속성 정의서 항목별 작성 요령

항목	작성 요령
소속 엔터티	속성이 속한 엔터티명을 기재한다.
속성명	「표준용어정의서」에 등록된 용어(한글)를 사용한다.
영문명	「표준용어정의서」에 등록된 영문 용어를 사용한다.
속성 유형	값의 성격에 따라 기본형·설계형·추출형 등으로 구분한다(아래 참조).
도메인	「표준도메인정의서」에 등록된 도메인명을 사용한다. 개별 컬럼에서 타입·길이를 임의 지정하지 않는다.
기본키 순번	복합키 순서(1,2,3…)를 기재하고, 기본키가 아니면 공란으로 둔다.
소속 유일키	보조 식별자에 참여하는 속성이면 `UKn-n` 형식(앞은 보조 식별자 순번, 뒤는 그 안에서의 속성 순번)으로 기재하고, 아니면 공란으로 둔다.
Null 허용여부	Null 허용은 `Y`, 불가는 `N`으로 표시한다.
개인정보 여부	개인정보이면 `Y`, 아니면 `N`으로 표시한다.
참조 엔터티	외래키인 경우 논리적으로 참조하는 엔터티명을 기재한다.
참조 속성명	외래키인 경우 참조하는 엔터티의 속성명을 기재한다.
기본값	속성의 기본값을 기재한다.
속성 설명	추가 설명·값 영역 정보를 기재한다. Null 허용이 `N`이면서 필수입력이 "조건부필수"이면 그 조건을 여기에 명시한다.

속성 유형의 판정 — 값이 어떻게 정해지는가로 구분한다.

표 6.24. 속성 유형

유형	정의	예시
기본형(Basic)	업무에서 직접 도출되어 그대로 저장	고객명, 가입일자
설계형(Designed)	설계 편의를 위해 부여(식별자·코드 등)	고객번호, 주문상태코드
추출형(Derived)	다른 값에서 계산·집계되어 파생	주문총금액, 연령

식별 관계의 표현 — 기본키는 **기본키 순번**(복합키 순서)으로 나타내고, 외래키는 **참조 엔터티·참조 속성명**에 참조 대상을 명시하여 표현한다. 단일 PK는 기본키 순번 `1`, 복합 PK는 결합 순서대로 `1`, `2`, `3`을 부여한다. 주 식별자가 아니면서 유일해야 하는 속성(이메일주소·주민등록번호 등)은 엔터티 정의서의 **보조 식별자(UK)**에 `UK1: 이메일주소` 형태로 선언하고, 속성 정의서의 **소속 유일키**에 `UK1-1`처럼 참여 순번을 부여하여 양쪽을 연결한다. 동일 원천이 서로 다른 역할로 두 번 상속되면(주문의 주문고객·수취고객) 역할 수식어를 붙여 구별하고, 참조 엔터티/속성에는 원천 엔터티를 그대로 적는다. 추출형(Derived) 속성은 정합성 위험이 있으므로 「속성 설명」에 산출식과 재계산 시점을 반드시 남긴다.

표준도메인 매핑

속성의 값 형식은 개별적으로 정하지 않고, 「표준도메인정의서」에 등록된 도메인 하나에 **매핑**하여 결정한다. 매핑의 열쇠는 속성명의 마지막 단어인 **분류어**다 — 분류어가 도메인그룹을, 도메인그룹 내 구체 도메인이 타입·길이·표현형식을 확정한다(5.3.2절, 6.3.3절).

- **[필수]** 모든 속성은 정확히 하나의 표준도메인에 연결한다. 도메인을 우회한 개별 타입·길이 지정을 금지한다. 같은 도메인을 쓰는 속성은 어느 엔터티에 있든 동일 타입·길이가 된다.
- **[필수]** 분류어와 도메인이 충돌하면('…율'인데 정수 도메인, '…여부'인데 문자 10자리) 표준 검증에서 오류로 처리한다.
- **[권장]** 도메인에 없는 형식이 필요하면 개별 컬럼을 예외 처리하지 말고, 「표준도메인정의서」에 신규 도메인을 등재한 뒤 매핑한다. 예외 컬럼은 표준의 허점이 된다.

표 6.25. 속성–도메인 매핑 예시

속성명	분류어	매핑 도메인(예)	물리 타입(Oracle · PostgreSQL)
고객명	명	명칭50	`VARCHAR2(50)` · `VARCHAR(50)`
가입일자	일자	일자8	`CHAR(8)` · `CHAR(8)`
주문금액	금액	금액15	`NUMBER(15)` · `NUMERIC(15)`
사용여부	여부	여부1	`CHAR(1)` · `CHAR(1)`

표준코드 매핑

분류어가 '코드'인 속성(코드성 속성)은 자유 입력이 아니라 「표준코드정의서」의 특정 코드에 **매핑**하여, 허용값을 그 코드의 코드값 집합으로 제한한다(6.3.4절). 매핑이 없으면 같은 의미의 상태값이 테이블마다 다른 문자열로 저장되어 정합성이 무너진다.

- **[필수]** 코드성 속성은 반드시 하나의 표준코드(코드 ID)에 연결하고, 컬럼에는 코드 도메인을 적용한다. 「참조 코드」에 코드 ID를 명시한다(6.4.2.3절).

- **[필수]** 허용값은 연결된 표준코드의 코드값으로만 제한한다. 물리적으로는 코드값 테이블을 참조하는 FK 또는 `CHECK` **제약**으로 강제한다.
- **[권장]** 값이 적고 거의 바뀌지 않는 코드는 `CHECK`로, 값이 많거나 운영 중 증감하는 코드는 코드 테이블 FK로 구현한다.

```
-- 나쁨: 상태를 자유 텍스트로 저장 → '취소','CANCEL','9' 혼재
ORD_STAT     VARCHAR2(20)
-- 좋음: 표준코드 ORD_STAT_CD 에 매핑, 허용값 제한
ORD_STAT_CD  VARCHAR2(2)  NOT NULL          -- 참조 코드: ORD_STAT_CD (01,02,03,09)
  CONSTRAINT CK_ORD_STAT CHECK (ORD_STAT_CD IN ('01','02','03','09'))
```

자동화와 품질 검증

논리 모델의 품질은 **모든 속성이 표준용어·표준도메인에 연결되었는가**로 판정한다. 이 연결을 도구로 강제하고, 빠진 것을 쿼리로 탐지한다.

- **[도구]** 모델링 도구에서 속성명을 자유 입력하지 못하게 하고 표준용어 사전에서만 선택하도록 설정한다. 이때 물리명·도메인이 자동으로 딸려 와 논리–물리 일관성이 입력 시점에 보장된다.
- **검증** 표준용어에 매핑되지 않은 속성, 도메인이 비어 있는 속성을 모델 리포지토리(또는 정의서 테이블)에서 좌외부조인으로 탐지한다. 미매핑 0건이 논리 모델 완결의 조건이다.
- **검증** 코드성 속성(분류어 '코드')인데 참조 코드가 비어 있는 항목을 뽑아 표준코드 매핑(6.4.1.3.2절) 누락을 메운다.

6.4.2 물리 데이터 모델링

물리 데이터 모델링은 논리 모델을 **채택 DBMS의 실제 객체**(데이터베이스·테이블·컬럼·제약·인덱스)로 전환하는 단계다. 명칭·도메인·키·참조라는 표준은 그대로 두고, 타입 표기·저장 구조·인덱스·성능 설계를 DBMS에 맞춰 확정한다.

6.4.2.1 데이터베이스 선정

물리 설계의 출발점은 데이터를 담을 DBMS와 데이터베이스를 정하는 일이다. 워크로드 성격(OLTP/OLAP), 라이선스·비용, 요구 기능(파티셔닝·복제·확장), 운영 조직의 숙련도를 종합해 선정하고, 아래 정의서에 물리 구성을 기록한다.

표 6.26. 데이터베이스 정의서 항목별 작성 요령

항목	작성 요령
데이터베이스명	정보 시스템에서 DB를 식별하는 물리 정보명(영문명)을 기재한다.
DBMS 종류·버전	DBMS의 이름과 버전을 기재한다. 식별자 길이·타입 제약이 버전에 종속되므로 정확히 적는다.
스키마명	DBMS 데이터베이스 내 스키마(소유자) 이름을 기재한다.
문자셋	텍스트 데이터를 저장할 때의 문자 인코딩 방식(Character Set)을 기재한다.
저장 특성	테이블스페이스 등 저장 구성을 기재한다.
용도	운영·개발·분석 등 용도를 기재한다.

- **선정 고려사항**: OLTP 중심의 트랜잭션 시스템은 동시성·잠금·복구 성숙도를, 분석계는 컬럼 저장·병렬 처리를 우선 고려한다. 상용(Oracle)은 고가용성·파티셔닝·튜닝 기능이 강하고, 오픈소스

(PostgreSQL)는 라이선스 부담이 없고 표준 SQL 준수·확장(extension)이 강점이다. 본 방법론은 이 두 DBMS를 예시 기준으로 삼는다.

- **문자셋**: 신규 구축은 유니코드를 기본으로 한다. Oracle은 `AL32UTF8` 캐릭터셋을, PostgreSQL은 `UTF8` 인코딩(+ 업무에 맞는 `LC_COLLATE`)을 지정한다. 문자셋이 물리 길이 산정(한글 바이트)과 정렬 순서(collation)에 직접 영향을 주므로, 도메인의 길이 정의(6.3.3절)와 일관되게 맞춘다.
- **저장 특성**: 대용량·이력 테이블은 별도 테이블스페이스로 분리해 I/O와 백업을 독립적으로 관리하는 방침을 여기서 명시한다.

자동화와 품질 검증

- 생성 데이터베이스·스키마·테이블스페이스 생성과 권한 부여(`GRANT`)를 환경(운영·개발·분석)별 파라미터만 바꿔 재사용하는 스크립트로 표준화한다. 수작업으로 생성한 DB는 문자셋·권한이 환경마다 달라지는 대표적 편차 지점이다.
- 검증 카탈로그에서 문자셋(`server_encoding` / `NLS_CHARACTERSET`)이 표준(유니코드)과 다른 데이터베이스, 소유자·권한이 표준과 어긋난 객체를 점검한다.

6.4.2.2 테이블 정의

표 6.27. 테이블 정의서 항목별 작성 요령

항목	작성 요령
테이블 물리명	정보시스템에서 테이블을 식별하는 물리 정보명(영문명)을 기재한다.
테이블 한글명	「표준용어정의서」와 명명규칙에 따라 정의한 테이블 한글명을 기재한다. 1개 엔터티가 여러 테이블로 분리되면 일련번호를 부여한다.
테이블 소유자	테이블 스키마의 소유권(오너십)을 갖는 DB 사용자를 기재한다.
관련 엔터티	이 테이블이 구현된 원천 엔터티명(한글)을 기재한다. 여러 엔터티가 하나로 통합되면 모두 기재한다.
데이터베이스	테이블을 생성할 데이터베이스명을 기재한다.
스키마	테이블을 생성할 DB 스키마명을 기재한다.
테이블 유형	일반·파티션·클러스터·뷰 테이블 등 물리적/논리적 특성에 따른 유형을 기재한다.
기본키(PK)	테이블의 유일키에 해당하는 PK를 기재한다.
유일키(UK)	기본키 외에 테이블을 고유하게 구별하는 컬럼 조합을 기재한다. 복수 구성이면 `+`로 연결하고, 접두어를 붙여 `UKn:컬럼명` 형식으로 표기한다. 유일키가 둘 이상이면 쉼표(`,`)로 구분한다.
예상 건수	보존기간 동안 테이블이 보유할 저장 건수를 기재한다.
테이블 설명	물리적 특징을 기재한다. 파티션 테이블은 분할 조건을, 뷰 테이블은 구성 집합 결정 조건을 기재한다.
보관 기간(년)	발생 시점 기준 데이터 보존 기간을 연 단위로 기재한다.
발생주기	데이터 발생·수정 주기(실시간/일/주/월/분기/반기 등)를 기재한다.
백업 여부	시스템 안정성을 위한 백업 수행 여부를 기재한다.

- **엔터티-테이블 대응**: 1:1이 기본이나, 조회 성능을 위해 하나의 엔터티를 수직 분할하면 「관련 엔터티」에 원천을 적고 한글명에 일련번호를 붙인다. 반대로 여러 엔터티를 통합하면 관련 엔터티를 모두 기재한다.
- **예상 건수·발생주기는 설계 근거**다: 대량·고빈도 테이블은 「테이블 유형」을 파티션으로 잡고 분할 키(예: 발생일자 월 단위)를 「테이블 설명」에 명시한다. 이 두 항목이 이후 파티셔닝·인덱스·아카이빙 정책의 입력이 된다.
- **기본키(PK) 제약명**은 명명규칙(5.6.4절)의 `PK_`+테이블명 규약을 따른다. 뷰 테이블은 「테이블 설명」에 구성 집합의 결정 조건(조인·필터)을 남긴다.

ER 관계 정의

관계는 엔터티(테이블) 간 업무 규칙을 선(line)으로 표현한 것으로, 물리 모델에서는 **외래키(FK)**로 구현된다. 관계 정의는 관계의 **종류·차수(cardinality)·선택성(optionality)·식별성(identifying)**을 확정하는 일이다.

표 6.28. 관계 유형과 물리 구현

관계 유형	의미	물리 구현
1:1	한 건이 상대 한 건과 대응	한쪽 PK를 다른 쪽에 FK+UNIQUE로 부여, 또는 테이블 통합
1:N	부모 한 건이 자식 여러 건과 대응(가장 흔함)	부모 PK를 자식에 FK로 전파
M:N	양쪽 여러 건이 서로 대응	**교차(연관) 엔터티**로 해소 후 1:N 두 개로 분해

- **식별관계 vs 비식별관계**: 부모 PK가 자식의 **PK 일부**가 되면 식별관계(강한 결합, ERD 실선)이고, 부모 PK가 자식의 **일반 FK 속성**이면 비식별관계(ERD 점선)다. 자식이 부모 없이 독립 식별될 수 있으면 비식별을 택한다.
- **선택성(optionality)**: 자식이 부모를 반드시 가지면 필수(FK `NOT NULL`), 없을 수도 있으면 선택(FK `NULL` 허용)이다. 차수와 함께 `1:N(필수)` 형태로 명시한다.
- **M:N 해소**: 고객과 상품의 M:N '가입'은 `고객상품가입`(FK 고객번호 + FK 상품번호, 두 FK의 결합이 PK) 교차 엔터티로 분해한다. 교차 엔터티가 자체 업무 속성(가입일자·상태)을 가지면 독립 엔터티로 승격한다.
- **자기참조**: 조직의 상위 조직처럼 같은 엔터티를 참조하면 `상위조직번호` FK로 표현하고, 최상위 행의 FK는 `NULL`을 허용한다.

자동화와 품질 검증

- 생성 테이블 물리명은 대응 엔터티의 표준약어 조합으로, PK·접두접미어(표 5.17)는 유형에 따라 규칙적으로 결정되므로, 테이블 골격은 정의서에서 생성한다.
- 검증 카탈로그를 조회하여 **PK가 없는 테이블**, 표준 접두·접미어 규칙을 벗어난 테이블명, 운영 스키마에 방치된 백업 사본(`_BAK_YYYYMMDD` 류)을 탐지한다. PK 없는 테이블은 중복과 정합성 오류를 유발하는 주요 원인이다.

```
-- PostgreSQL: 기본키가 정의되지 않은 테이블
SELECT t.table_name
  FROM information_schema.tables t
 WHERE t.table_schema = 'APP' AND t.table_type = 'BASE TABLE'
   AND NOT EXISTS (
       SELECT 1 FROM information_schema.table_constraints tc
```

```
        WHERE tc.table_schema = t.table_schema
          AND tc.table_name   = t.table_name
          AND tc.constraint_type = 'PRIMARY KEY');
```

6.4.2.3 컬럼 정의

표 6.29. 컬럼 정의서 항목별 작성 요령

항목	작성 요령
소속 테이블	컬럼이 소속된 테이블의 이름(영문명)을 기재한다.
컬럼 순번	테이블 내 컬럼의 위치 순번을 기재한다.
컬럼 물리명	「표준용어정의서」에 등록된 영문 용어(표준약어 조합)를 사용한다.
컬럼 한글명	연관 속성명과 동일하게, 표준용어를 준수하여 기재한다.
데이터 타입	DBMS 종속적인 데이터 타입명을 기재한다. 도메인정의서의 물리 타입을 그대로 따른다.
길이	「표준도메인정의서」의 길이를 기재한다. 없으면 DDL에 지정할 길이를 DBMS 종속 표현으로 기재한다.
Null 허용	Nullable은 `Y`, Not Null은 `N`으로 표시한다.
기본키 순번	복합키 순서(1,2,3…)를 기재하고, 기본키가 아니면 공란으로 둔다.
소속 유일키	유일키에 참여하는 컬럼이면 `UKn-n` 형식(앞은 유일키 순번, 뒤는 그 안에서의 컬럼 순번)으로 기재하고, 아니면 공란으로 둔다.
기본값	컬럼의 기본값을 기재한다.
참조 테이블	외래키인 경우 참조 대상 테이블의 물리명을 기재한다.
참조 컬럼	외래키인 경우 참조 대상 컬럼의 물리명을 기재한다.
참조 코드	컬럼이 코드값이면 참조하는 코드의 명칭이나 코드 ID를 기재한다.
개인정보 여부	「개인정보 비식별 조치 가이드라인」의 식별자 조치 기준에 따른 개인정보 포함 여부(`Y`/`N`)를 기재한다.
암호화 여부	개인정보 보호 등의 이유로 컬럼을 암호화하는지 여부(`Y`/`N`)를 기재한다.
컬럼 설명	컬럼 이해를 돕는 부가 설명·예외 사항을 기재한다.

물리 컬럼의 키 표현 — 기본키는 **기본키 순번**(복합키 순서)으로, 외래키는 **참조 테이블·참조 컬럼**에 참조 대상을 명시하여 표현한다. 컬럼이 코드값을 담으면 **참조 코드**에 참조 코드 체계를 기재한다. 논리 모델의 보조 식별자는 테이블 정의서의 **유일키(UK)**와 컬럼 정의서의 **소속 유일키**로 이어받아, DDL에서 `UNIQUE` 제약으로 구현한다.

- **데이터 타입·길이는 도메인을 우회하지 않는다**: 같은 용어는 어느 테이블에 있든 동일 컬럼명·동일 타입이어야 한다. 특히 조인 키의 타입 불일치(`CUST.CUST_NO VARCHAR(10)` ↔ `ORD.CUST_NO NUMBER`)는 묵시적 형변환으로 인덱스를 무력화하는 대표적 성능 사고다(5.9.1.5절).
- **외래키 정합성**: 「참조 테이블·참조 컬럼」이 채워진 컬럼은 참조 대상과 타입·길이가 정확히 일치해야 하며, 이는 DDL에서 FK 제약으로 강제된다.

- **개인정보·암호화**: 「개인정보 여부 = Y」이고 암호화 대상이면 「암호화 여부 = Y」로 표기하고, 물리 길이는 암호문 출력 길이로 산정한다(6.3.3절).

데이터 유형 정의
컬럼의 데이터 타입은 새로 정하는 것이 아니라 **매핑된 도메인의 물리 타입을 채택 DBMS 표기로 옮긴** 것이다. Oracle·PostgreSQL 타입 대응은 표 6.10 을 따른다.

- **[필수]** 문자형은 가변 길이를 기본으로 한다(`VARCHAR2(n)` ↔ `VARCHAR(n)`). 고정 자릿수 코드·일자(`YYYYMMDD`)만 `CHAR(n)`을 쓴다. 길이 semantics 차이(Oracle 기본 바이트 vs 문자, PostgreSQL 문자)를 인지하고 한글 저장 길이를 도메인 길이와 맞춘다(표 6.8 인근 설명 참조).
- **[필수]** 숫자형은 `NUMBER(p,s)` ↔ `NUMERIC(p,s)`로 정밀도·스케일을 명시한다. **금액·수량에 `FLOAT`/`REAL`(부동소수점)을 쓰지 않는다** — 반올림 오차로 합계가 틀어진다.
- **[필수]** 날짜/시각은 도메인 정의를 따른다. 표시·연산 편의로 `CHAR(8)`(`YYYYMMDD`)을 쓸지, 시점 연산을 위해 `DATE`/`TIMESTAMP`를 쓸지는 도메인에서 확정하고 컬럼이 임의로 바꾸지 않는다.
- **[권장]** 불리언은 Oracle에 네이티브 타입이 없으므로 `CHAR(1)` `'Y'`/`'N'` 표준을 쓰고, PostgreSQL도 저장 표준을 맞춘다(표 5.18).

인덱스 및 기본키/외래키 정의
키와 인덱스는 무결성과 성능을 물리적으로 보장하는 장치다. 정의서의 키 표현(기본키 순번·참조 테이블/컬럼)을 실제 제약과 인덱스로 확정한다.

- **[필수] 기본키(PK)**: 모든 테이블에 하나를 둔다. 제약명은 `PK_`+테이블명(표 5.20). 대부분의 DBMS는 PK에 유니크 인덱스를 자동 생성한다. 본질키가 길거나 불안정하면 인조키를 PK로, 본질키는 UNIQUE로 보존한다.
- **[필수] 외래키(FK)**: 「참조 테이블·참조 컬럼」을 FK 제약으로 구현한다. 제약명은 `FK_`+자식+`_`+부모. 참조 컬럼과 **타입·길이가 정확히 일치**해야 한다.
- **[필수] FK 컬럼에는 인덱스를 생성**한다. 부모–자식 조인과 부모 행 삭제 시 자식 스캔에 쓰이며, 특히 **Oracle은 FK에 인덱스를 자동 생성하지 않아** 인덱스가 없으면 부모 변경 시 자식 테이블 전체 잠금·풀스캔이 발생한다.
- **[권장] 인덱스 설계**: 자주 쓰는 조회 조건·조인·정렬 컬럼을 대상으로 하고, 필요한 컬럼만 담아 인덱스만으로 응답하는 커버링 인덱스를 활용한다. 인덱스는 조회를 빠르게 하지만 `INSERT`/`UPDATE`/`DELETE`마다 갱신 비용이 들므로 **과다 생성을 피한다**.

```
-- 나쁨: FK 제약만 걸고 인덱스 미생성 → 부모 삭제·조인 시 자식 전체 스캔
ALTER TABLE ORD ADD CONSTRAINT FK_ORD_CUST FOREIGN KEY (CUST_NO) REFERENCES CUST (CUST_NO);
-- 좋음: FK 컬럼에 인덱스 동반 생성
CREATE INDEX IX_ORD_CUST_NO ON ORD (CUST_NO);
```

자동화와 품질 검증

- 생성 컬럼의 물리명·타입·길이는 표준용어와 도메인에서 결정되므로, 도구의 사전·도메인 연동으로 자동 채운다. 사람은 값을 선택하기만 하고 직접 입력하지 않는다.
- 검증 **명명 린트** — 카탈로그의 컬럼·테이블명이 물리명 규칙(대문자·숫자·`_`, 영문자 시작)을 지키는지 정규식으로 점검한다.

```
-- PostgreSQL: 대문자·숫자·_ 외 문자가 섞였거나 영문자로 시작하지 않는 물리명
SELECT table_name, column_name
  FROM information_schema.columns
 WHERE table_schema = 'APP'
   AND column_name !~ '^[A-Z][A-Z0-9_]*$';
-- Oracle: REGEXP_LIKE(COLUMN_NAME, '^[A-Z][A-Z0-9_]*$') = FALSE 로 동일 점검
```

- 검증 **조인키 타입 정합성** — 같은 용어의 컬럼(예: 여러 테이블의 `CUST_NO`)이 테이블마다 타입·길이가 다른지 카탈로그로 대조해, 묵시적 형변환에 의한 성능 사고(5.9.1.5절)를 사전에 방지한다.

6.4.3 논리와 물리 모델의 차이



그림 6.3. 논리 모델·물리 모델 요소 매핑

그림 6.3 처럼 논리 모델의 각 요소는 대체로 물리 모델의 요소와 1:1로 대응한다. 엔터티는 테이블로, 속성은 컬럼으로, 주식별자는 기본키로, 관계는 외래키로, 도메인은 데이터 타입·길이로 옮겨진다. 그러나 물리 단계에서는 성능과 저장 효율을 위해 이 1:1 대응이 **의도적으로 깨지기도 한다**. 바로 이 지점이 논리 모델과 물리 모델의 차이를 드러내는 핵심이므로, 설계 시 그 근거를 분명히 남겨야 한다.

- **수직 분할** — 하나의 엔터티를 여러 테이블로 나눈다. 조회가 잦은 컬럼과 드물게 쓰는 대형 컬럼을 분리하거나, 민감 정보를 별도 테이블로 떼어 접근을 통제할 때 사용한다.
- **테이블 통합** — 반대로 여러 엔터티를 하나의 테이블로 합친다. 항상 함께 조회되는 1:1 관계의 엔터티들을 조인 비용 없이 다루기 위해서다.
- **서브타입 전환** — 논리의 슈퍼타입/서브타입을 하나의 테이블로 합칠지(single table), 서브타입별 개별 테이블로 나눌지(table-per-subtype)를 데이터 밀도와 조회 패턴에 따라 결정한다.
- **반정규화·파생 컬럼** — 논리 모델에는 없던 반정규화(De-normalization) 컬럼이나 이력·집계용 파생 컬럼을 물리 단계에서 추가한다. 조인·집계 성능을 얻는 대신, 갱신 시 원본과의 정합성을 유지할 책임이 함께 생긴다.

이러한 변형은 편의가 아니라 성능·저장·보안 요건에 대한 판단의 결과다. 따라서 어떤 대응을 왜 깨뜨렸는지, 원본과의 정합성을 어떻게 유지하는지를 정의서에 함께 기록한다. 개별 결정의 상세 기준은 물리 데이터 모델링(6.4.2절)과 물리 가이드의 테이블 정의(5.8.2절)·반정규화(5.8.6절)를 참조한다.

6.4.4 ERD 작성

ERD(Entity–Relationship Diagram)는 엔터티·속성·관계를 한 장의 그림으로 표현하여, 모델의 구조와 업무 규칙을 이해관계자가 공유하게 한다. 논리 ERD(업무 관점)와 물리 ERD(구현 관점)를 분리해 관리한다.

6.4.4.1 ERD 작성 방법

- **[필수]** 표기법을 전사 하나로 통일한다. 실무에서는 **IE(정보공학) 표기법**(까마귀발, crow's foot)이 널리 쓰이며, Barker 표기법도 대안이다. 혼용하면 같은 그림을 다르게 읽는다.
- **[필수]** 관계선은 **차수와 선택성**을 함께 표기한다. 까마귀발은 'N(다)' 쪽을, 원(○)은 선택(0 가능)을, 막대(|)는 필수(1 이상)를 나타낸다. **식별관계는 실선, 비식별관계는 점선**으로 구분한다(6.4.2.2.1절).

표 6.30. ERD 관계 표기 요소(IE)

요소	IE(까마귀발) 표기
엔터티	사각형(상단 엔터티명, 구분선 위 식별자·아래 일반 속성)
1:N 관계	부모 쪽 막대(|), 자식 쪽 까마귀발
선택성	선택 = 원(○), 필수 = 막대(|)
식별/비식별	식별 = 실선, 비식별 = 점선

- **[권장]** 도구(ERwin·DA#·dbdiagram 등)로 작성하고, 논리↔물리 모델을 연동해 관리한다. 물리 모델에서 DDL을 뽑는 포워드 엔지니어링, 기존 DB에서 모델을 복원하는 리버스 엔지니어링을 활용해 문서–DB 간 표류를 막는다.
- **[권장]** 레이아웃은 부모를 위/왼쪽, 자식을 아래/오른쪽에 두어 관계선 교차를 최소화하고, 규모가 크면 **주제영역(Subject Area)**별로 분할해 가독성을 확보한다.

6.4.4.2 자동화와 품질 검증

- 생성 ERD는 수작업으로 그리는 그림이 아니라 **모델의 표현**이다. 모델링 도구에서 물리 모델을 확정하면 ERD·DDL·정의서가 하나의 원본에서 함께 산출되도록 구성한다(포워드 엔지니어링).
- 검증 운영 DB를 **리버스 엔지니어링**하여 현행 ERD를 복원하고, 관리 중인 모델과 비교(테이블·컬럼·관계 차이)해 문서–DB 표류를 정기적으로 탐지한다. 표류가 발견되면 어느 쪽이 정본인지 판정하고 동기화한다.
- **[도구]** 논리·물리 모델과 표준 사전을 도구의 단일 리포지토리에 두고 접근 권한을 통제해, 표준을 우회한 개별 수정이 원본에 반영되지 않게 한다.

> KDSM ERD 작성 기능
>
> KDSM의 라이브러리와 툴킷은 다음처럼 ERD를 자동으로 생성하는 기능을 제공한다.
>
> - **리버스 엔지니어링**: DB에서 모델을 복원하여 ERD 작성, 모델 유지보수에 활용
> - **포워드 엔지니어링**: 물리 모델을 확정하면 산출물에서 ERD 작성, 신규 데이터 모델링에 활용
>
> 명령은 `kdsm erd`이며, 산출물·메타테이블·카탈로그의 세 원천을 받는다(7.5.3절).

6.4.5 DDL 생성

물리 모델이 확정되면 정의서 항목을 실제 DDL(Data Definition Language)로 옮긴다. 정의서의 명칭·타입·키·참조가 DDL 문장과 1:1로 대응하며, 표기(타입명·구문)만 DBMS를 따른다.

6.4.5.1 DDL 생성 및 DDL 수행 순서 정의

아래는 위 규약을 따른 고객·주문 테이블의 생성문으로, 명칭·키·참조는 동일하고 **타입 표기만 DBMS를 따른다**(Oracle → PostgreSQL).

```sql
-- Oracle
CREATE TABLE CUST (
    CUST_NO        VARCHAR2(10)   NOT NULL,   -- 기본키 순번 1
    CUST_NM        VARCHAR2(100)  NOT NULL,
    CUST_GRD_CD    VARCHAR2(2)    DEFAULT '10' NOT NULL,  -- 참조 코드: CUST_GRD_CD
    RRNO_ENC       VARCHAR2(64),                          -- 개인정보 Y, 암호화 Y
    CONSTRAINT PK_CUST PRIMARY KEY (CUST_NO)
);

CREATE TABLE ORD (
    ORD_NO         VARCHAR2(20)   NOT NULL,   -- 기본키 순번 1
    CUST_NO        VARCHAR2(10)   NOT NULL,   -- 참조 테이블 CUST · 참조 컬럼 CUST_NO
    ORD_DT         CHAR(8)        NOT NULL,   -- 일자 도메인
    ORD_AMT        NUMBER(15)     DEFAULT 0 NOT NULL,
    ORD_STAT_CD    VARCHAR2(2)    NOT NULL,   -- 참조 코드: ORD_STAT_CD
    CONSTRAINT PK_ORD PRIMARY KEY (ORD_NO),
    CONSTRAINT FK_ORD_CUST FOREIGN KEY (CUST_NO) REFERENCES CUST (CUST_NO)
);
```

```sql
-- PostgreSQL
CREATE TABLE CUST (
    CUST_NO        VARCHAR(10)   NOT NULL,           -- 기본키 순번 1
    CUST_NM        VARCHAR(100)  NOT NULL,
    CUST_GRD_CD    VARCHAR(2)    NOT NULL DEFAULT '10',  -- 참조 코드: CUST_GRD_CD
    RRNO_ENC       VARCHAR(64),                       -- 개인정보 Y, 암호화 Y
    CONSTRAINT PK_CUST PRIMARY KEY (CUST_NO)
);

CREATE TABLE ORD (
    ORD_NO         VARCHAR(20)   NOT NULL,           -- 기본키 순번 1
    CUST_NO        VARCHAR(10)   NOT NULL,           -- 참조 테이블 CUST · 참조 컬럼 CUST_NO
    ORD_DT         CHAR(8)       NOT NULL,           -- 일자 도메인
    ORD_AMT        NUMERIC(15)   NOT NULL DEFAULT 0,
    ORD_STAT_CD    VARCHAR(2)    NOT NULL,           -- 참조 코드: ORD_STAT_CD
    CONSTRAINT PK_ORD PRIMARY KEY (ORD_NO),
    CONSTRAINT FK_ORD_CUST FOREIGN KEY (CUST_NO) REFERENCES CUST (CUST_NO)
);
```

두 DDL은 `VARCHAR2(10)` ↔ `VARCHAR(10)`, `NUMBER(15)` ↔ `NUMERIC(15)`처럼 타입 표기만 다를 뿐, `CUST_NO`가 두 테이블에서 동일 타입인 점, FK가 참조 테이블·참조 컬럼과 일치하는 점, 코드성 컬럼이 참조 코드로 연결되는 점은 같다. 표준(명칭·도메인·키·참조)은 DBMS와 무관하게 정의서 항목과 1:1로 대응한다.

DDL 수행 순서 — 객체 간 의존성 때문에 생성·삭제 순서를 지켜야 한다. 특히 FK는 부모 테이블이 이미 있어야 하므로, **모든 테이블을 먼저 만들고 FK를 나중에 `ALTER TABLE`로 추가**하면 테이블 생성 순서 의존을 없앨 수 있다.

표 6.31. DDL 수행 순서

순서	생성(CREATE)	비고
1	스키마·테이블스페이스	저장 구조 선행
2	시퀀스(`SEQ_`)	기본값·식별자 채번에 사용
3	테이블 + PK/UNIQUE	FK는 아직 제외
4	외래키(FK)	모든 테이블 생성 후 `ALTER TABLE ADD CONSTRAINT`
5	인덱스(FK 컬럼 포함)	성능·무결성
6	코멘트·권한(GRANT)	문서화·접근 제어

- **삭제(DROP)는 역순**으로 수행한다(FK 먼저 제거 후 테이블). 참조가 걸린 테이블을 먼저 지우면 오류가 난다.
- **트랜잭션 처리 차이**: PostgreSQL은 DDL을 트랜잭션으로 묶어 실패 시 전체 롤백할 수 있으나, **Oracle은 DDL이 암묵적으로 커밋**되므로 중간 실패에 대비해 순서와 재실행(idempotent) 전략을 스크립트에 반영한다.

6.4.5.2 자동화와 품질 검증

DDL은 물리 데이터 모델의 최종 산출물이자, 표준이 실제 DB로 옮겨지는 관문이다. 손으로 쓰지 말고 **정의서에서 생성 → 순서대로 수행 → 리버스로 검증**하는 파이프라인으로 다룬다.

- **생성** 컬럼 물리명·타입은 표준용어·도메인에서, 제약은 참조·코드 매핑에서 나오므로, 물리 정의서가 채워지면 DDL은 결정된다. 모델링 도구의 포워드 엔지니어링 또는 정의서→DDL 템플릿으로 생성하고, 사람은 생성물을 검토만 한다.
- **생성** 스크립트는 **재실행 가능(idempotent)**하게 만든다(객체 존재 여부 확인 후 생성, FK는 `ALTER` 로 분리). 그래야 실패 지점부터 다시 돌릴 수 있다.
- **검증** 배포 후 운영 DB를 리버스 엔지니어링하여 정의서와 대조한다. 테이블·컬럼·타입·제약의 차이를 뽑아 "정의서에는 있으나 DB에 없음/그 반대"를 0으로 수렴시킨다.
- **검증** 표준 준수 자동 점검을 배포 파이프라인(CI)에 넣어, 명명 린트·도메인 적합성·PK/FK 누락·고아 코드 검사가 모두 통과해야 배포되게 한다. 이로써 개인차에 의존하지 않고 품질을 균질화한다.
- **[도구]** 이 파이프라인은 KDSM 라이브러리의 `kdsm ddl`(생성) · `kdsm drift`(배포 후 대조) · `kdsm validate`(CI 점검)로 수행한다(7.5.1절, 7.7.2절).

표 6.32. 배포 전 표준 준수 자동 점검 목록

점검 항목	방법	기준
명명 린트	카탈로그 정규식(6.4.2.3.3절)	위반 0
도메인 적합성	카탈로그↔도메인 대조(6.3.3.3절)	불일치·미지정 0
PK/FK 무결성	카탈로그 제약 점검(6.4.2.2.2절)	PK 누락 0
코드 정합성	고아 코드 조회(6.3.4.1절)	고아 0
문서–DB 일치	리버스 대조	차이 0

CHAPTER 7

라이브러리

7.1 개요

KDSM 라이브러리(`kdsm`)는 데이터 표준화와 데이터 모델링의 산출물을 코드로 다루는 파이썬 라이브러리이다. 앞 장들이 기재한 내용을 기계가 읽어 검사하고 다른 형태로 변환하는 수단을 다룬다.

본 방법론의 서술은 프로세스(제2장), 표준 및 절차 매뉴얼(제3장), 가이드(제5장), 테크니컬 노트(제6장)의 순서로 구체화되며, 본 장은 그 마지막 자리에서 방향을 바꾼다. 앞의 네 장이 **사람이 무엇을 해야 하는가**를 규정한다면 본 장은 **그 가운데 무엇을 기계에 맡길 수 있는가**를 규정한다. 테크니컬 노트의 각 절 말미에 있는 「자동화와 품질 검증」이 원리를 제시하였다면, 본 장은 그 원리를 실제로 수행하는 도구를 제시한다.

표준화 작업에는 사람의 판단이 필요한 부분과 그렇지 않은 부분이 섞여 있다. 어떤 낱말을 표준단어로 삼을 것인지, 어떤 엔터티가 필요한지는 사람이 정한다. 그러나 그 결정을 여러 문서에 옮겨 적고, 옮겨 적은 값이 서로 어긋나지 않는지 확인하고, 확인을 마친 내용을 DDL이나 ERD로 바꾸는 일은 규칙이 정해져 있다. 라이브러리는 규칙이 정해져 있는 부분을 담당하며 사람의 판단을 대신하지 않는다.

표 7.1. 사람의 판단과 라이브러리의 역할

사람이 하는 일	라이브러리가 하는 일
표준단어·표준용어를 무엇으로 정할지 결정한다	이름을 낱말로 분해하여 후보를 제시한다
어떤 엔터티와 속성이 필요한지 결정한다	기재한 이름이 사전에 등재되어 있는지 검사한다
산출물에 값을 기재한다	여러 문서에 기재한 값이 서로 일치하는지 대조한다
개선 대상과 순서를 결정한다	표준 준수 수준을 측정하여 제시한다

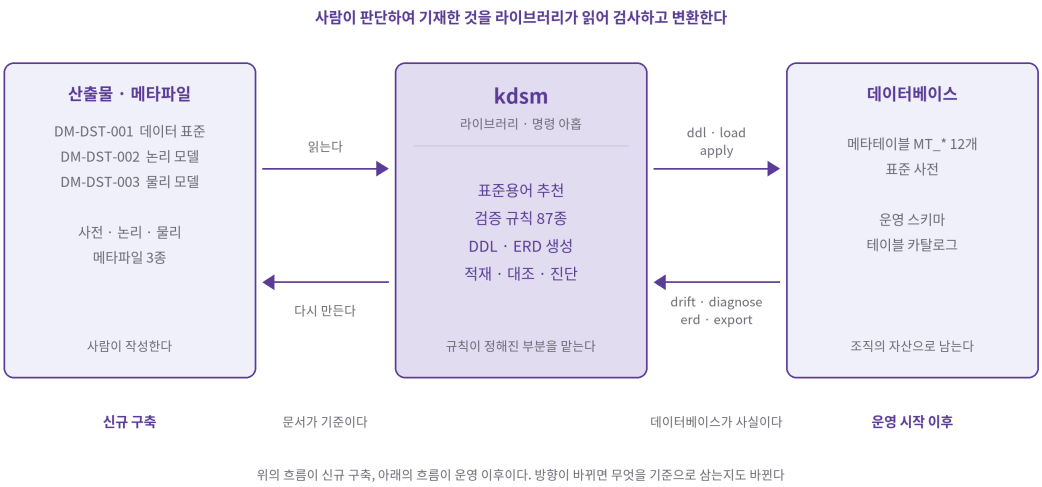

그림 7.1. 라이브러리가 놓인 자리와 두 방향

라이브러리가 다루는 대상은 산출물 파일과 데이터베이스 카탈로그이며, 그림 7.1 와 같이 두 방향으로 사용한다. **문서에서 데이터베이스로** 향하는 방향은 산출물을 읽어 검증하고 DDL을 생성하여 스키마를 만들며 메타데이터를 적재하는 것으로, 신규 구축 시점의 방향이다. **데이터베이스에서 문서로** 향하는 방향은 운영 중인 데이터베이스의 카탈로그를 읽어 정의서와 대조하고 표준 준수 수준을 진단하며 산출물과 ERD를 다시 생성하는 것으로, 운영이 시작된 뒤의 방향이다.

이 라이브러리는 데이터를 처리하지 않는다. 다루는 것은 데이터의 구조와 이름에 관한 정보, 즉 **메타데이터**이다. 업무 데이터를 조회하거나 변환하는 기능은 제공하지 않는다.

표 7.2. 육안 검사와 라이브러리 검사의 대비

항목	사람이 수행할 때	라이브러리를 사용할 때
표준 준수 확인	산출물을 육안으로 대조한다	104종 규칙으로 전수 검사한다
문서 사이의 불일치	적재 시점에 드러난다	적재 이전에 검출한다
DDL 작성	정의서를 보고 직접 작성한다	정의서에서 생성한다
산출물 갱신	데이터베이스를 보고 문서를 고친다	데이터베이스에서 다시 생성한다
레거시 진단	표본을 추출하여 확인한다	카탈로그 전체를 측정하여 비율로 제시한다

가장 큰 차이는 **검사 범위**이다. 육안 대조는 표본을 볼 수밖에 없으므로 통과하였다는 사실이 나머지를 보증하지 못한다. 전수 검사는 검사 대상과 결과가 함께 기록되므로 무엇을 확인하였고 무엇을 확인하지 않았는지가 남는다. 두 번째 차이는 **검출 시점**이다. 문서 사이의 불일치는 파일을 읽는 데에는 지장이 없으므로 적재 시점까지 드러나지 않으며, 그 시점에는 이미 스키마를 만든 뒤이므로 되돌리는 비용이 크다.

> **노트**
>
> 본 장은 라이브러리의 기능을 방법론의 수행 절차에 대응시켜 요약한 것이다. 명령별 전체 옵션과 규칙별 상세 목록은 **데이터 표준화 라이브러리 사용자 매뉴얼**(유충현 2026년a)에 수록되어 있으며, 본 장과 함께 사용하면 각 명령을 실제로 실행할 때의 인수와 출력을 확인할 수 있다.

7.2 라이브러리 체계

7.2.1 두 계층 — 라이브러리와 커맨드라인



그림 7.2. 라이브러리와 커맨드라인의 관계

`kdsm`은 그림 7.2 와 같이 두 계층으로 제공된다. 아래 계층은 파이썬 API이고 위 계층은 커맨드라인이다.

표 7.3. 라이브러리의 두 계층

계층	대상	제공하는 것
라이브러리 (`kdsm.*`)	애플리케이션을 개발하는 사람	모델·리더·규칙·생성기를 파이썬 API로 제공한다
커맨드라인 (`kdsm.cli`)	표준화와 모델링을 수행하는 사람	아홉 명령으로 전형적인 작업을 한 줄에 수행한다

커맨드라인은 별도의 구현이 아니다. 인자를 해석하고 라이브러리의 함수를 호출한 뒤 결과를 화면에 출력하는 얇은 층이다. 따라서 두 계층의 판단 기준이 어긋나는 일은 발생하지 않으며, 커맨드라인으로 확인한 결과를 애플리케이션에서 재현할 수 있고 그 반대도 성립한다. 아래 둘은 같은 작업이다.

```
kdsm ddl 고객주문_V0.1.0_20260812.xlsx -o dbms/ddl/고객주문.sql --drop
```

```python
from kdsm.ddl import generate_ddl_from_spec

generate_ddl_from_spec(
    "고객주문_V0.1.0_20260812.xlsx",
    output="dbms/ddl/고객주문.sql",
    include_drop=True,
)
```

명령을 사용하면 인자 검사와 오류 안내를 함께 얻고, 함수를 직접 호출하면 결과를 파이썬 객체로 받아 이어서 처리할 수 있다. 표준화와 모델링을 수행하는 실무자에게는 커맨드라인이 사실상 유일한 진입점이며, 파이썬을 사용하지 않아도 아홉 명령으로 추천부터 진단까지 수행할 수 있다.

7.2.2 프로세스 산출물–명령 매핑

데이터 표준 정의(Phase 2~5) 단계의 산출물은 아래 명령으로 지원한다. 가이드가 규정한 원칙과 테크니컬 노트가 제시한 작성 요령을 기계가 검사하는 자리가 `validate`이며, 작성 자체를 돕는 자리가 `suggest`이다.

표 7.4. 데이터 표준 정의 산출물–명령 매핑

단계	산출물(정의서)	작성 요령	지원 명령
표준단어 정의	표준단어 정의서	6.3.1절	`suggest --coin` · `validate`
표준용어 정의	표준용어 정의서	6.3.2절	`suggest` · `validate`
표준도메인 정의	표준도메인 정의서	6.3.3절	`validate`
표준코드 정의	표준코드 정의서	6.3.4절	`validate`

데이터 모델 정의(표준화 적용) 단계의 산출물은 아래 명령으로 지원한다. 테크니컬 노트가 ERD 작성(6.4.4절)과 DDL 생성(6.4.5절)에서 제시한 「정의서에서 생성 → 순서대로 수행 → 리버스로 검증」의 파이프라인이 그대로 명령에 대응한다.

표 7.5. 데이터 모델 정의 산출물–명령 매핑

단계	산출물(정의서)	작성 요령	지원 명령
엔터티·속성 정의	엔터티 정의서 · 속성 정의서	6.4.1.2절 · 6.4.1.3절	`validate` · `erd`
데이터베이스·테이블·컬럼 정의	데이터베이스 · 테이블 · 컬럼 정의서	6.4.2.1절 · 6.4.2.2절 · 6.4.2.3절	`validate` · `erd`
ERD 작성	ERD	6.4.4절	`erd`
DDL 생성	DDL 스크립트	6.4.5절	`ddl`
배포 후 대조	—	6.4.5.2절	`drift` · `diagnose`
메타 적재 · 문서 재생성	—	—	`load` · `apply` · `export`

7.2.3 아홉 명령

표 7.6. `kdsm` 명령 아홉

명령	하는 일	방향
`suggest`	한글 용어명을 분해하여 표준용어 후보를 제시한다	작성
`validate`	산출물의 표준 준수 여부를 104종 규칙으로 검사한다	검사
`ddl`	물리 데이터 모델 산출물에서 DDL을 생성한다	문서 → DB
`load`	메타데이터 파일을 메타테이블에 적재한다	문서 → DB
`apply`	프로젝트가 작성한 DM-DST 산출물을 적재한다	문서 → DB
`erd`	산출물·메타테이블·카탈로그에서 ERD를 작성한다	양방향
`drift`	정의서와 운영 데이터베이스의 차이를 대조한다	DB → 문서
`export`	메타테이블이나 카탈로그를 읽어 산출물을 생성한다	DB → 문서
`diagnose`	표준 준수 수준을 측정하여 비율과 등급으로 제시한다	DB → 문서

명령은 **종료 코드**로 결과를 알린다. 0은 정상, 1은 처리 오류 또는 검증 실패, 2는 사용법 오류이다. 다만 `diagnose`와 `drift`는 측정하고 대조하는 작업이지 합격 여부를 판정하는 작업이 아니므로 기본값이 항상 0이며, 각각 `--min-score`와 `--strict`를 지정한 경우에만 1이 될 수 있다. 이 구분은 CI 연계(7.7.2절)에서 그대로 사용된다.

7.3 설치와 환경 준비

7.3.1 설치

Git 저장소에서 바로 설치한다. 별도의 패키지 서버를 두지 않으며 저장소의 태그가 곧 판이다. 파이썬은 3.11 이상을 요구한다.

```
uv tool install 'kdsm[db] @ git+https://github.com/<조직>/<저장소>.git@v0.9.0#subdirectory=library'
```

표 7.7. 설치 주소의 구성

조각	뜻
`kdsm[db]`	패키지 이름과 선택 의존성
`git+https://...`	저장소 주소
`@v0.9.0`	설치할 판. 태그·브랜치·커밋을 지정한다
`#subdirectory=library`	패키지가 저장소 뿌리가 아니라 `library/`에 있다

주소 전체를 따옴표로 감싼다. bash와 zsh가 대괄호를 파일명 패턴으로 해석하고 `#` 뒤를 주석으로 보기 때문이다. 코드에서 `import kdsm` 하는 경우에는 `uv tool install` 대신 `uv pip install`로 프로젝트 환경에 넣거나 `pyproject.toml`의 의존성에 같은 주소를 적는다.

> ⚠ 자격 증명을 주소에 적지 않는다
>
> 비공개 저장소를 설치할 때 `https://<토큰>@github.com/...` 형태로 토큰을 주소에 넣지 않는다. 그 주소는 셸 기록과 `pyproject.toml`에 그대로 남는다. 자격 증명은 Git의 credential helper에 맡긴다.

7.3.2 선택 의존성

데이터베이스 접속과 외부 서비스 연동은 선택 의존성(extra)으로 분리되어 있다. 필요한 것만 설치하면 나머지 기능은 사용하지 않는 대신 설치 부담이 없다.

표 7.8. 선택 의존성 여섯

extra	활성화되는 기능
`db`	PostgreSQL 적재·조회·진단
`mysql`	MySQL 적재·조회·진단 (`--dbms mysql`)
`databricks`	Databricks 적재·조회·진단 (`--dbms databricks`)
`llm`	Claude API를 이용한 신규 표준단어 초안 제안 (`--coin`)
`bedrock`	AWS Bedrock을 이용한 `--coin` 제안
`pdf`	ERD의 PDF 합본 (`--pdf-merge`)

전부를 뜻하는 이름은 따로 두지 않았으므로 필요한 것을 쉼표로 나열한다(`kdsm[db,mysql,databricks]`). 설치하지 않은 기능을 사용하면 무엇을 설치해야 하는지 안내하고 중단하며, 다른 기능은 그대로 동작한다.

> ⚠ 선택 의존성을 나누어 설치하지 않는다
>
> `uv sync`는 기본이 **정확 동기화**이다. 요청한 것만 남기고 나머지를 제거하므로, `uv sync --extra db` 다음에 `uv sync --extra llm`을 실행하면 앞에서 설치한 것이 제거된다. 한 줄에 모두 지정하거나 `--all-extras`를 사용한다.

7.3.3 접속 정보와 비밀 취급

데이터베이스에 접속하는 명령은 환경변수에서 접속 정보를 읽는다. 접두어는 대상 DBMS가 정하며 `--env-prefix`로 변경할 수 있다.

표 7.9. 대상별 접속 환경변수

대상	환경변수
PostgreSQL	`KDSM_DB_HOST` · `PORT` · `NAME` · `SCHEMA` · `USER` · `PASSWORD`
MySQL	`MYSQL_*`
Databricks	`DATABRICKS_SERVER_HOSTNAME` · `DATABRICKS_HTTP_PATH` · `DATABRICKS_TOKEN`

파일만 읽는 작업에는 환경변수가 필요하지 않다. 산출물 검증(`validate --meta`), DDL 생성(`ddl`), 산출물에서 그리는 ERD(`erd -s`)가 이에 해당한다.

> ⚠ 비밀은 인수로 받지 않는다
>
> 비밀번호와 토큰은 어떤 명령도 커맨드라인 인수로 받지 않는다. 명령줄에 적은 값은 `ps` 출력과 셸 기록에 남기 때문이다. 값을 주입하는 방법은 셋이다. ① `read -rs`로 그 자리에서 입력받는다. ② 권한을 `600`으로 제한한 파일에 두고 읽어 들인다. ③ 비밀 저장소에서 꺼낸다. 접속 문자열을 만드는 함수도 비밀번호를 제외하며, 이는 로그와 예외 메시지에 실려 나가는 것을 막기 위한 의도적 설계이다.

7.3.4 KDSM 시스템 환경 준비

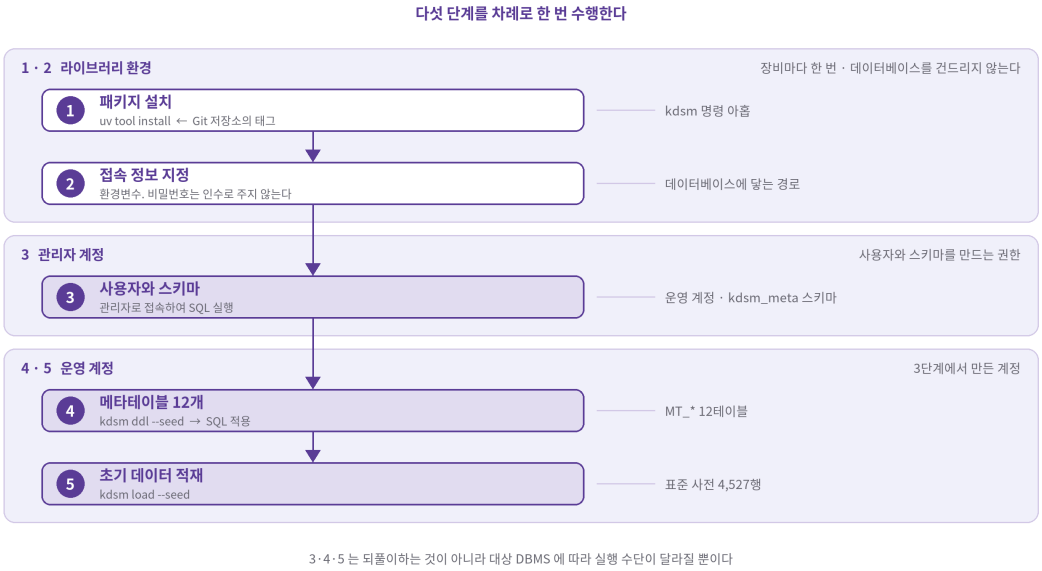

그림 7.3. 환경 준비의 다섯 단계

패키지를 설치하는 것과 데이터베이스에 메타테이블을 세우는 것은 다른 일이다. 앞의 것은 장비마다 한 번, 뒤의 것은 데이터베이스마다 한 번 수행한다. 그림 7.3 의 다섯 단계를 차례로 한 번 수행하며, 3·4·5단계는 대상 DBMS에 따라 실행 수단이 달라질 뿐 되풀이하는 것이 아니다.

표 7.10. KDSM 시스템 환경 준비의 다섯 단계

단계	하는 일	실행할 것	마치면 생기는 것
1	패키지를 설치한다	`uv tool install`	`kdsm` 명령 아홉
2	접속 정보를 지정한다	환경변수 지정	데이터베이스에 닿는 경로
3	사용자와 스키마를 만든다	관리자 계정으로 SQL 실행	운영 계정과 `kdsm_meta` 스키마
4	메타테이블 12개를 만든다	`kdsm ddl --seed`로 SQL 생성 후 적용	`MT_*` 12테이블
5	초기 데이터를 적재한다	`kdsm load --seed`	표준 사전 4,527행

1·2단계는 데이터베이스를 건드리지 않는다. 이 둘까지 마치면 파일만 읽는 작업은 모두 수행할 수 있다. **3단계만 관리자 계정으로 수행한다.** 사용자와 스키마를 만드는 권한과 테이블을 만드는 권한은 층위가 다르며, 한 계정으로 묶으면 운영 계정에 필요 이상의 권한을 주게 된다. 4단계부터는 3단계에서 만든 운영 계정으로 수행한다.

4단계에서 `kdsm ddl`은 SQL 파일을 만들 뿐 실행하지 않는다. 만드는 일과 적용하는 일은 필요한 권한도 되돌리는 방법도 다르기 때문이다. 적용 순서는 표준 사전 → 논리 모델 → 물리 모델이며, 뒤의 것이 앞의 것을 가리키므로 순서를 바꾸면 외래키에 걸린다.

중간에서 멈추었을 때에는 멈춘 단계부터 다시 시작한다. 3단계는 이미 있는 사용자의 비밀번호를 덮어쓰지 않고, 5단계는 기본키가 있으면 갱신하므로(UPSERT) 같은 파일을 여러 번 적재하여도 겹쳐 쌓이지 않는다.

7.4 데이터 표준 정의 지원

7.4.1 표준용어 추천

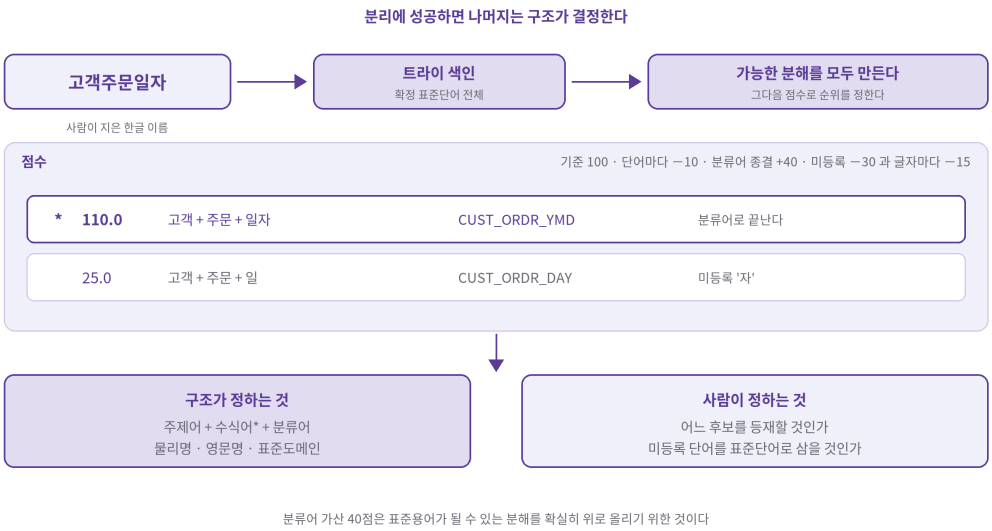

그림 7.4. 표준용어 추천의 원리

표준용어 작성 가이드(5.3절)는 표준용어가 **주제어(0개 이상) + 수식어(0개 이상) + 분류어(정확히 1개, 반드시 마지막)**의 구조를 갖는다고 규정한다. 이 구조가 확정되어 있으므로 한글 용어명을 표준단어 단위로 분리하는 데 성공하면 물리명·영문명·도메인은 기계적으로 결정된다. 어려운 부분은 분리 작업뿐이며, `kdsm suggest`가 그 부분을 담당한다.

```
kdsm suggest 고객여신금액 상품가격 --meta <메타파일>   # 사전을 파일에서
kdsm suggest 고객여신금액 --from-db                  # 적재된 사전에서
```

사전의 출처는 `--meta`와 `--from-db` 중 하나를 반드시 지정한다. 기본값을 두지 않은 것은 어느 사전으로 추천하였는지가 결과의 뜻을 바꾸기 때문이다. 표준을 수정하는 중이면 파일이 최신이므로 `--meta`를, 확정된 사전으로 작업하면 `--from-db`를 사용한다.

7.4.1.1 분해와 점수

그림 7.4 와 같이 확정 표준단어를 **트라이(trie)**로 색인한 뒤 가능한 분해를 모두 생성하고 점수를 부여하여 순위를 정한다. 4,143개 규모의 사전에서 색인 구축에 3밀리초, 분해 하나에 1밀리초 미만이 소요된다.

표 7.11. 분해 후보의 점수 기준

항목	점수
기준 점수	100
단어 하나마다	−10
마지막이 분류어	+40
미등록 단어 하나마다	−30
미등록 단어의 글자 하나마다	−15

```
"고객주문일자"
  110.0  고객 + 주문 + 일자    CUST_ORDR_YMD    분류어 종결
   25.0  고객 + 주문 + 일      CUST_ORDR_DAY    미등록 ['자']
```

점수 체계의 두 값이 중요하다. **분류어 가산 40점**은 단어 수만으로는 3단어와 4단어의 차이가 10점에 그치므로, 표준용어가 될 수 있는 분해를 확실히 상위에 올리기 위한 것이다. **미등록 단어의 글자 수 감점**은 다섯 글자를 인식하지 못한 분해와 세 글자를 인식하지 못한 분해를 구분하기 위한 것이다. 한 글자 미등록 단어도 45점이 감점되도록 정하였는데, 「수료」를 분류어 「수」와 미등록 「료」로 나누는 억지 분해를 막기 위해서이다.

일부만 인식하지 못한 이름도 나머지를 활용하여 후보를 만들며, 미등록 단어는 신규 표준단어 등재 후보가 된다. **폐기 상태의 표준단어는 색인에서 제외**하며 이음동의어와 금칙어도 색인에 포함하지 않는다. 표준단어가 아니기 때문이다.

7.4.1.2 금지어의 처리

표준단어 작성 가이드(5.2절)가 규정한 이음동의어와 금칙어는 추천 단계에서 대체어로 치환된다.

표 7.12. 금지어의 세 가지 구분

구분	기록 위치	대체어 제시
이음동의어	확정 단어의 이음동의어 항목	가능
금칙어	확정 단어의 금칙어 항목	가능
폐기된 표준단어	상태가 폐기인 표준단어	불가

```
# 계약날짜
  *  120.0  CTRT_YMD  [연월일C8]  Contract Date
     이름: 계약날짜 → 계약일자
     구성: 계약 + 일자
     · '날짜'는 이음동의어. 일자(YMD)로 쓴다
```

치환에는 두 가지 제약을 둔다. 첫째, **인식하지 못한 구간만 치환한다.** 이름 전체에 치환을 적용하면 `설명 → 설이름`, `매출액 → 매출금액`, `비교 → 비용교` 같은 오변형이 생긴다. 둘째, **대체어가 여럿이면 고르지 않는다.** 시스템이 임의로 선택하면 잘못된 이름이 인지되지 않은 채 등재된다.

7.4.1.3 신규 표준단어 초안

추천기는 미등록 단어를 찾는 데까지 관여하고, 영문명·영문약어·정의·단어구분은 사람이 작성한다. `--coin`을 지정하면 그 초안을 LLM으로부터 제공받는다.

```
– 신규 표준단어 제안 (LLM · 확정 아님)
  * 데이터마트 (조각을 합침)
    Data Mart  DMART  02 도메인어  상태 01 후보
    정의: (data mart) 특정 주제 영역의 자료를 이용 목적에 맞게 모아 둔 저장소
    근거: '마트'는 단독으로 쓰이지 않는다. 사전에 '메타데이터'가 한 단어로 등재되어 있다.
```

이 기능의 설계에는 세 가지 판단이 반영되어 있다. 첫째, **영문약어는 LLM이 고르지 않는다.** LLM은 확정 표준단어 수천 개의 약어를 알지 못하므로 후보를 셋 제공받고 사용되지 않은 것을 코드가 선택한다. 사전 전체를 프롬프트에 넣는 방식보다 비용이 적고 결과가 확실하다. 둘째, **제공받은 초안을 기존 규칙으로 다시 검사**하고 통과한 것만 제시한다. 셋째, **결과는 화면에만 출력한다.** 시트에 기록하려면 사람의 판단이 선행되어야 하기 때문이다.

> ⚠ 추천과 검증을 섞지 않는다
>
> `--coin`은 `validate`·`load`·`ddl`에 연결되어 있지 않다. 검증은 사실 확인이고 제안은 새로운 생성이므로, 둘을 섞으면 CI 실행마다 결과가 달라진다. 자동화의 전제는 같은 입력이 같은 결과를 낳는 것이다.

7.4.2 표준 준수 검증



그림 7.5. 검증 규칙이 놓인 자리

`kdsm validate`는 그림 7.5 와 같이 산출물 3종을 읽어 **적재 이전에 적재 가능 여부를 판정**한다. 가이드와 테크니컬 노트가 규정한 기준을 규칙으로 옮긴 것이며, 현재 104종이 정의되어 있다.

표 7.13. 검증 대상과 목적

산출물	검사 대상	무엇을 묻는가
DM-DST-001	표준단어·표준용어·표준도메인·표준코드	사전에 적재할 대상이 무결한가. 기존 등재분과 중복되지 않는가
DM-DST-002	엔터티·속성	논리 모델 테이블에 적재할 대상이 무결한가
DM-DST-003	데이터베이스·테이블·컬럼	물리 모델 테이블에 적재할 대상이 무결한가. 논리 모델과 상충하지 않는가

7.4.2.1 규칙 104종의 구성

표 7.14. 검증 규칙의 구성

대상	규칙 수
표준단어	8
표준용어	8
표준도메인	4
도메인그룹·분류	9
표준코드	9
메타파일 헤더	4
논리 모델 엔터티	7
논리 모델 속성	10
주제영역	3
주제영역 배정	8
물리 모델 컬럼	9
물리 모델 테이블	5
물리 모델 데이터베이스	2
엔터티 ↔ 테이블 대조	6
속성 ↔ 컬럼 대조	10
신규 등재분 ↔ 메타테이블 대조	2

규칙 이름은 `<대상>.<검사내용>` 형태이다(`word.abbr-unique`, `column.length-satisfied`, `table.primary-key`). 규칙마다 이름이 붙어 있으므로 리포트에서 위반을 규칙별로 묶어 제시할 수 있고, 조직의 사정에 따라 특정 규칙을 제외하거나 심각도를 바꾸어 다시 넣을 수 있다.

7.4.2.2 심각도

표 7.15. 검증 결과의 심각도

등급	뜻	종료 코드 반영
ERROR	반드시 고쳐야 할 위반	반영
WARNING	검토가 필요한 것	`--strict` 일 때만
INFO	판정하지 않았다는 기록	미반영

INFO는 검사하지 않았다는 사실을 남기기 위한 등급이다. 규칙이 특정 항목을 검사할 수 없어 생략하였을 때 그 사실을 기록한다. 검사하지 않은 것을 통과한 것으로 오인하지 않도록 하기 위한 것이며, 같

은 이유로 참조 사전을 지정하지 않으면 그 사전을 요구하는 규칙은 실행되지 않고 그 사실이 리포트에 남는다.

```
검사 699건 · 오류 1건 · 경고 0건

[ERROR] column.length-satisfied — 1건
  MT_STD_DMN.PRM_VL_SCP_CN: VARCHAR(256) 컬럼인데 길이를 넘는 값이 1건 있습니다. 최대 300자
```

7.4.2.3 자동화와 품질 검증

- 검증 산출물 3종을 한 번에 지정한다. 종류는 파일명이 아니라 **시트 구성**으로 판별하므로 이름과 순서에 제약이 없다. 셋을 함께 주면 서로를 참조하여 문서 사이에 걸친 대조까지 수행한다. 데이터 표준 정의서를 먼저 읽어 사전에 더한 뒤, 그 사전을 기준으로 논리·물리 산출물을 검사한다.
- 검증 두 계층을 구분한다. **표준 준수**는 KDSM 규칙의 준수 여부이며 메타파일로도 판정할 수 있다. **적재 가능성**은 기본키가 중복되는지와 같이 적재 대상 테이블만 판정할 수 있는 것이다. 사전을 파일로 지정하면 뒤의 계층은 수행하지 않으며, 그 사실을 INFO로 남긴다.
- 검증 기존 사전과의 중복은 세 형태로 구분하여 보고한다. **이름과 내용이 모두 같으면** 경고이다. 폐기 항목을 다시 등재하려는 의도일 수 있어 담당자의 판단 여지를 남긴다. **이름이 같고 내용이 다르면** 오류이며 어긋난 항목의 두 값을 함께 제시한다. **이름은 새것인데 약어가 겹치면** 오류이다.
- 검증 검증을 통과하지 못하면 적재하지 않는다. `load`와 `apply`가 적재 직전에 다시 검증하며, 위반이 있으면 어떤 항목도 적재하지 않는다. 검증 시점과 적재 시점 사이에 사전이 변경되었을 수 있기 때문이다.
- **[도구]** CI에 등록하면 산출물을 수정할 때마다 자동으로 실행된다(7.7.2절).

7.5 데이터 모델 정의 지원

7.5.1 DDL 생성

DDL은 직접 작성하지 않고 물리 데이터 모델 산출물에서 생성한다. 테크니컬 노트가 「모델링 도구의 포워드 엔지니어링 또는 정의서→DDL 템플릿으로 생성하고 사람은 생성물을 검토만 한다」고 제시한 방식(6.4.5.2절)을 그대로 수행한다.

```
kdsm ddl 고객주문_V0.1.0_20260812.xlsx -o dbms/ddl/고객주문.sql --drop
kdsm ddl 고객_....xlsx 주문_....xlsx -o dbms/ddl      # 주제영역별로 나눈 경우
```

산출물을 여럿 지정하면 주제영역별로 나눈 것으로 보고 **파일 수만큼 SQL 파일**을 만들며 이름은 주제영역 이름을 따른다. 여러 벌을 하나로 합치지 않는 것은 적용 순서를 사람이 통제할 수 있어야 하기 때문이다. 주제영역 사이를 넘는 외래키는 참조 대상이 그 파일에 없더라도 제약문을 그대로 생성하며, 적용 순서는 참조 관계가 정한다.

7.5.1.1 DBMS별 표기

정의서는 그대로 두고 표기만 대상 DBMS의 것으로 바꾼다. 하나의 물리 데이터 모델로 세 DBMS의 DDL을 만들 수 있다.

표 7.16. DBMS별 DDL 표기

항목	PostgreSQL	MySQL	Databricks
이름	`스키마.테이블`	`스키마.테이블`	`카탈로그.스키마.테이블`
인용 부호	큰따옴표	backtick	backtick
문자열	`VARCHAR(40)`	`VARCHAR(40)`. 길이가 없으면 `TEXT`	`VARCHAR(40)`. 길이가 없으면 `STRING`
수치	`NUMERIC(9,2)`	`DECIMAL(9,2)`	`DECIMAL(9,2)`
일시	`TIMESTAMP`	`DATETIME`	`TIMESTAMP`
주석	`COMMENT ON` 문	정의 안에 기재	`COMMENT ON` 문
삭제	`DROP TABLE ... CASCADE`	`DROP TABLE ...`	`DROP TABLE ...`

세 DBMS에서 확인할 것이 각각 있다. PostgreSQL은 인용하지 않은 식별자를 소문자로 접어 저장한다. 표준 물리명이 모두 대문자·숫자·밑줄이므로 생성기는 식별자를 인용하지 않으며, 카탈로그를 직접 조회할 때에는 소문자로 조회하거나 인용하지 않는다. MySQL은 `VARCHAR`를 길이 없이 사용할 수 없고 `TEXT`는 기본키에 그대로 쓸 수 없으므로, 키가 되는 컬럼의 도메인에는 길이를 지정해 둔다. Databricks는 기본키와 외래키가 선언될 뿐 강제되지 않으며, 문자열은 길이를 가진 도메인만 `VARCHAR(n)`·`CHAR(n)`이 되고 길이가 없는 도메인은 `STRING`이 된다.

> ⚠ `--drop`은 개발 환경에서만
>
> `--drop`은 `CREATE` 앞에 `DROP TABLE IF EXISTS`를 넣는다. PostgreSQL에는 `CASCADE`가 함께 붙으므로 참조하는 객체까지 사라진다. 운영 데이터베이스에 이미 데이터가 있다면 지정하지 않고 필요한 `ALTER` 문을 별도로 작성한다.

7.5.1.2 자동화와 품질 검증

- 생성 DDL을 만들기 전에 검증을 통과시킨다. 생성기는 정의서 내용을 그대로 옮길 뿐이므로 잘못된 정의는 잘못된 DDL로 이어진다. 특히 외래키가 참조 대상 복합 기본키의 일부만 가리키는 경우는 DDL이 생성되더라도 실행 단계에서 거부되며, 검증 규칙이 이를 미리 검출한다.
- 생성 복합 외래키는 컬럼 순번에 따라 하나로 묶는다. 검증 규칙도 같은 방식으로 묶으므로 검사를 통과한 정의가 생성 단계에서 어긋나지 않는다.
- 검증 배포 후에는 `kdsm drift`로 정의서와 카탈로그를 대조한다. **테이블·컬럼·타입·제약** 네 가지를 비교하며, 차이를 **정의서에만 있음 · 데이터베이스에만 있음 · 서로 다름** 셋으로 구분하여 제시한다. **어느 쪽이 옳은지는 판정하지 않는다.** 적재 이전이면 정의서가 계획이므로 정의서가 기준이고 운영 중이면 데이터베이스가 사실이므로 데이터베이스가 기준인데, 프로그램은 그 시점을 알 수 없기 때문이다.
- 검증 제약은 이름이 아니라 구성 컬럼으로 대조한다. 정의서에는 제약 이름을 기재하는 항목이 없으므로 이름을 대조하면 사람이 직접 생성한 제약이 모두 다르다고 판정된다. **기본키는 컬럼의 순서까지 본다.** 복합 기본키의 컬럼 순서가 인덱스의 쓰임을 바꾸기 때문이다.

7.5.2 메타데이터 적재

확정한 표준과 모델은 메타테이블에 적재하여 조직의 자산으로 관리한다. 적재는 두 갈래이다.

표 7.17. 적재의 두 가지 목적

목적	명령	적재 대상	수행 시점
메타데이터 파일의 적재	`load`	사전·논리·물리 메타파일 3종	KDSM 구축 초기 단계
산출물 파일의 적재	`apply`	프로젝트가 작성한 DM-DST 3종	해당 산출물의 검증 완료 후

전자는 사전 자체를 구축하는 작업이고 후자는 구축된 사전에 프로젝트의 표준과 모델을 더하는 작업이다. 후자는 기존 등재분과의 충돌 여부를 먼저 확인해야 한다는 점도 다르다.

적재 순서는 참조 관계가 정한다. 열두 테이블이 하나의 순서를 이루며 파일이 셋으로 나뉘어 있어도 순서는 하나이다. 일부 시트만 선택하여도 이 순서는 유지된다. 지정한 순서를 그대로 따르면 외래키 제약을 위반하게 되기 때문이다.

세 DBMS 모두 결과는 같다. 기본키가 이미 있으면 갱신하고 없으면 넣는다. 다만 문장과 트랜잭션의 성질이 다르다.

표 7.18. DBMS별 적재 방식

DBMS	적재 문장	트랜잭션
PostgreSQL	임시 테이블에 `COPY` 후 `INSERT ... ON CONFLICT DO UPDATE`	지원한다
MySQL	`INSERT ... ON DUPLICATE KEY UPDATE`	지원한다
Databricks	`MERGE INTO ... USING (VALUES ...)`	없다

PostgreSQL은 임시 테이블에 일괄 적재한 뒤 한 번의 문장으로 본 테이블에 반영하므로, 네트워크 왕복 횟수가 행 수가 아니라 테이블 수에 비례한다. 4,193행 규모의 표준단어도 몇 번의 일괄 요청으로 적재를 마친다. **Databricks에는 트랜잭션이 없으므로 중간에 실패하면 앞의 것이 남는다.** 수정한 뒤 처음부터 다시 적재한다.

적재는 감사 컬럼을 함께 기록한다. 갱신할 때 생성일시와 생성자를 그대로 두는 것이 핵심이다. 누가 언제 최초로 적재하였는지가 남아 있어야 이력이 의미를 갖는다.

> ⚠ UPSERT는 삭제하지 않는다
>
> 적재는 삽입과 갱신만 수행한다. 시트에서 삭제한 행은 데이터베이스에 그대로 남는다. 항목을 없애려면 상태를 폐기로 변경하여 다시 적재하거나 데이터베이스에서 직접 삭제한다.

7.5.3 ERD 자동 작성

테크니컬 노트는 ERD를 정의서에서 생성할 것을 권고한다(6.4.4.2절). `kdsm erd`는 **논리 ERD를 엔터티와 속성에서, 물리 ERD를 테이블과 컬럼에서** 도출하며 세 가지 원천을 받는다.

표 7.19. ERD 작성의 세 가지 원천

방향	원천	사용 시점
포워드 엔지니어링	작성한 산출물	신규 데이터 모델 개발
리버스 엔지니어링	적재된 메타테이블	데이터 모델 유지보수
리버스 엔지니어링	DBMS 카탈로그	기존 데이터베이스 구조 파악

출력은 Mermaid 소스(`.mmd` · `.md`)와 렌더링 이미지(`.png` · `.svg` · `.pdf`)이다. 기본값에 이미지가 없는 것은 이미지 생성에 외부 도구가 필요하기 때문이다. 기본 동작이 외부 도구를 요구하면 그 도구가 없는 환경에서 아무 결과도 얻지 못한다.

관계선의 출처는 둘이다. **주 출처**는 속성 정의서의 참조 엔터티·참조 속성이며, **보조 출처**는 엔터티 정의서의 주요 관계이다. 두 출처를 모두 읽어야 한다. 어느 한쪽만 읽어 그린 ERD는 실제 구조보다 관계선이 적다. 식별관계 여부는 참조하는 속성에 기본키 순번이 있는지로 판정하며, 상위 엔터티 항목은 판정에 사용하지 않는다. 이 항목은 실무에서 식별관계가 아니라 소속을 나타내는 뜻으로 기재되기 때문이다.

> 관계선의 수가 검사 기준이 된다
>
> 논리 ERD와 물리 ERD의 관계선 수를 대조하면 누락을 확인할 수 있다. 물리 ERD의 관계선은 생성된 DDL의 외래키 개수와 일치해야 하며, 논리 ERD가 그보다 많다면 그 차이는 외래키로 표현되지 않는 관계이다. 두 ERD를 나란히 놓고 대조하는 것이 논리 모델에만 있는 관계를 발견하는 가장 빠른 방법이다.

7.5.4 산출물 내보내기

`kdsm export`는 적재된 메타테이블이나 DBMS 카탈로그를 읽어 산출물 엑셀을 생성한다. 사람이 작성하는 일을 대신하는 것이 아니라 두 경우를 위한 기능이다. **산출물 없이 구축한 데이터베이스에서 산출물을 얻을 때**와 **이미 있는 산출물의 판을 데이터베이스에 맞출 때**이다.

생성 대상은 DM-DST 3종이다. 산출물은 표지·헤더·설명 행·데이터·작성 지침이 한 시트에 함께 있는 서식 문서이므로, 빈 파일에서 생성하면 서식과 지침을 모두 잃는다. 따라서 템플릿을 복사하고 **데이터 영역만 교체**한다. 행 수가 달라지면 지침 블록을 이동하고 새 행에는 첫 데이터 행의 서식을 적용한다.

메타테이블이 담고 있지 않은 것이 둘 있으며, 그 처리 방식이 이 기능의 성격을 정한다.

표 7.20. 내보낸 산출물의 표지 기재

항목	기본값	그렇게 정한 이유
버전	`0.0.0`	사람이 검토하여 올린 판과 자동 생성된 판이 혼용되지 않도록 한다
작성자	공란	아직 사람이 검토하지 않았음을 나타낸다

사람이 작성한 버전의 번호와 이름을 **승계하지 않는다.** 파일명의 버전과 날짜도 함께 바뀌며 이름에서 「템플릿」을 제외한다. 기본 출력 경로도 원본과 다른 곳으로 둔다. 내보낸 문서로 원본을 덮어쓰면 복구할 수 없기 때문이다.

> **노트**
>
> 데이터베이스를 읽어 산출물 엑셀을 생성하는 것은 DM-DST 3종만 지원한다. KDSM 메타테이블의 초기 데이터를 정의한 DM-LIB 산출물은 시스템 관리자가 작성하는 문서이므로 생성 대상에 넣지 않는다.

7.6 운영 단계의 진단

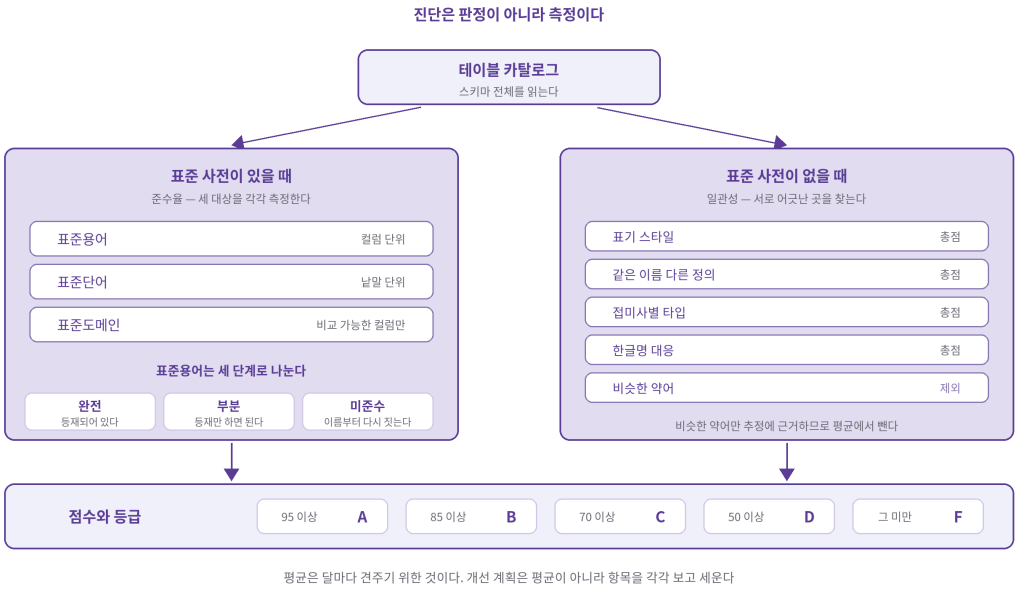

그림 7.6. 진단의 구조

7.6.1 검증과 진단의 구분

`kdsm validate`는 사전과 산출물이 어긋나지 않는지 확인하며 결과가 통과 또는 실패이다. `kdsm diagnose`는 이미 구축된 데이터베이스가 표준을 어느 정도 준수하는지 측정하며 **결과가 비율**이다. 전자는 「이대로 구축해도 되는가」를, 후자는 「이미 구축한 것을 어디부터 개선할 것인가」를 묻는다. 판정이 아니라 측정이므로 진단은 기본적으로 정상 종료한다.

그림 7.6 는 진단의 구조를 나타낸다. 표준 사전이 있으면 준수율을, 없으면 일관성을 산출하며 어느 쪽이든 점수와 등급으로 마무리된다.

7.6.2 세 가지 진단 대상

표 7.21. 표준 사전 대조의 세 가지 진단 대상

진단 대상	측정 항목	집계 단위
표준용어	컬럼 물리명이 사전에 등재된 용어인가	컬럼
표준단어	물리명을 밑줄로 분리한 낱말이 등재된 영문약어인가	낱말
표준도메인	용어가 정의한 타입·길이가 실제 컬럼과 일치하는가	용어가 등재된 컬럼

표준용어는 세 단계로 구분한다. 이 구분이 진단의 실용성을 결정한다.

```
완전   CUST_NM        표준용어로 등재되어 있다
부분   CUST_ORDR_NM   등재는 없지만 고객·주문·이름 약어 조합이다
미준수 CUST_FOO_NM    FOO 가 표준단어가 아니다
```

부분 준수는 사전에 등재만 하면 되는 상태이고 **미준수는 이름부터 다시 부여해야 하며 애플리케이션 코드까지 변경된다.** 조치의 성격과 비용이 전혀 다르므로 둘을 묶어 집계하면 「미준수 40%」라는 수치만 남아 어디부터 조치할지 판단할 수 없다.

도메인은 비교할 수 있을 때만 집계한다. 용어가 등재되지 않은 컬럼은 준수해야 할 도메인도 없으므로, 0으로 집계하면 표준용어 미준수가 두 번 반영된다. 그러므로 분모에서 제외한다.

함께 제시하는 두 항목이 개선의 출발점이 된다. **미등록 약어**는 표준단어가 아닌 약어를 사용 빈도가 높은 순서로 집계한 것이다. 62개 위치에서 쓰인 약어 하나를 등재하면 62개 컬럼이 한 번에 부분 준수가 된다. **같은 이름 다른 정의**는 같은 컬럼명이 테이블마다 다른 타입으로 정의된 경우이며, 표준화가 필요하다는 가장 강한 신호이다.

7.6.3 점수와 등급

세 가지 진단 대상의 평균을 종합 점수로 산출한다.

표 7.22. 진단 점수의 등급

점수	등급
95 이상	A
85 이상	B
70 이상	C
50 이상	D
그 미만	F

진단 대상마다 집계 단위가 다르므로 평균 자체에 큰 의미는 없다. 다만 지난달과 이번 달을 비교하려면 하나의 수치가 필요하다. **개선 계획을 수립할 때에는 평균이 아니라 세 가지를 각각 확인한다.**

7.6.4 사전이 없는 스키마의 일관성 진단

표준 사전을 아직 만들지 않은 조직이나 인수한 레거시 스키마도 진단할 수 있다. 전제는 하나이다. **여러 위치에서 같은 대상을 다르게 지칭하고 있다면 그중 하나는 잘못된 것이다.** 무엇이 옳은지는 알 수 없더라도 서로 어긋나 있다는 사실은 확인할 수 있다.

표 7.23. 사전 없이 확인하는 다섯 가지

구분	확인 내용	예	총점 반영
표기 스타일	한 스키마에 표기법이 혼용되어 있는가	`CUST_NM`과 `custNm`	O
같은 이름 다른 정의	같은 컬럼명의 타입이 서로 다른가	`CUST_NM`이 `VARCHAR(40)`과 `VARCHAR(100)`	O
접미사별 타입	같은 접미사의 타입이 서로 다른가	`_YMD`가 `DATE`와 `CHAR(8)`	O
한글명 대응	주석의 한글명과 컬럼명이 일대일인가	같은 `고객번호`에 컬럼명이 셋	O
비슷한 약어	같은 낱말을 다르게 축약하였는가	`CUSTOMER` · `CUST` · `CUS`	—

항목마다 「해당 문제에 해당하지 않는 컬럼의 비율」을 산출하고 네 값을 평균하여 일관성 점수를 얻는다. 가중치를 두지 않은 것은 무엇이 더 중요한지 외부에서 정할 근거가 없기 때문이다. **비슷한 약어만 총점에서 제외하였다.** 이 항목만 추정에 근거하므로 점수는 산출하되 평균에서는 뺀다.

사전이 있으면 두 항목의 판정이 달라진다. **접미사별 타입**은 표준도메인이 있으면 접미사별 타입이 다른 것이 정상일 수 있으므로, 도메인을 준수한 컬럼을 판정에서 제외하고 남은 컬럼끼리만 비교한다. 제외하지 않으면 표준화할수록 점수가 내려가는 모순이 생긴다. **비슷한 약어**는 사전에 물어본다. 서로 다른 낱말로 등재되어 있으면 앞부분이 겹쳐도 묶지 않는다.

7.6.5 이름 통계

준수율이 판정이라면 이름 통계는 집계이다. 설명 채움율, 낱말 길이 순위, 전체 낱말 길이 순위, 낱말 개수 순위, 접미사 순위, 최댓값 셋을 제시한다. **통계에는 등급이 없다.** 몇 자가 적당한지는 조직이 정할 문제이며, 자료는 그 판단의 근거를 제공할 뿐이다.

설명 채움율은 값이 높을수록 좋다고 판단할 수 있는 유일한 지표이지만 일관성 점수에는 포함하지 않는다. 일관성 점수는 불일치를 측정하는 것이고, 주석이 없는 것은 불일치가 아니라 미작성이기 때문이다.

7.6.6 진단 보고서

동일한 결과를 세 가지 형태로 제공한다. 계산은 이미 수행되었으며 출력 형식만 다르다.

표 7.24. 진단 보고서의 세 형태

형태	지정 방법	용도
화면	기본	즉시 확인. `-q`로 한 줄 요약
엑셀	`-o`	열네 시트. 담당자가 조치 대상을 추려 나가는 작업지
Quarto · HTML · PDF	`--qmd` · `--render`	배포용 보고서. 표지·목차·머리말을 갖춘 조판 문서

엑셀 리포트의 시트 구성은 **사전의 유무와 관계없이 항상 같다.** 사전이 없을 때에는 준수율 시트에 `측정 불가 (표준 사전 없음)`이라고 기재한다. 시트가 사라지면 그 항목을 검사하지 않았다는 사실도 함께 사라지기 때문이다.

마지막 시트인 「진단 제언」이 진단의 결론이다. 지적 사항을 **해당하는 컬럼이 많은 순서**로 배열하고, 각 항목에 조치 사항 한 문장, 판단 근거가 되는 수치, 영향 컬럼 수, 난이도, 근거 시트를 함께 기재한다.

표 7.25. 조치의 난이도

난이도	해당하는 작업
쉬움	사전에 등재하거나 주석을 작성한다. 기존 코드에 영향이 없다
보통	타입이나 길이를 조정한다. DDL을 수정해야 하지만 이름은 유지된다
어려움	이름을 다시 부여한다. 그 컬럼을 참조하는 모든 코드가 영향을 받는다

컬럼명을 변경하는 SQL은 생성하지 않는다. 개명은 애플리케이션 전체에 영향을 미치므로 사람이 범위를 확인한 뒤 수행할 일이다.

7.7 작업 흐름과 자동화

7.7.1 처음부터 끝까지



그림 7.7. 명령의 관점에서 본 작업 차례

방법론의 수행 절차를 명령의 관점에서 다시 배열하면 그림 7.7 와 같다. 앞의 넷은 사람이 산출물을 작성하는 단계이고, 라이브러리는 각 단계 사이에서 다음 단계로 문제가 이월되는 것을 막는다.

표 7.26. 명령의 관점에서 본 작업 차례

차례	하는 일	명령
1	표준단어·표준용어·표준도메인·표준코드를 작성한다	`suggest` · `validate`
2	논리 데이터 모델을 작성한다	`validate`
3	물리 데이터 모델을 작성한다	`validate`
4	논리와 물리를 대조한다	`validate` (3종 동시 지정)
5	ERD를 작성한다	`erd -s`
6	DDL을 생성하여 적용한다	`ddl`
7	메타데이터를 적재한다	`load` · `apply`
8	정의서와 데이터베이스를 대조한다	`drift`
9	표준화 수준을 진단한다	`diagnose`
10	산출물을 다시 생성한다	`export`

1부터 7까지가 신규 구축이고 8부터 10까지는 구축 이후의 작업이다. **기존 시스템을 인수한 경우에는 9부터 시작한다.** 진단으로 현재 상태를 측정하고, `export --from-catalog`로 현재 구조를 정의서로 옮긴 뒤, 진단이 알려 준 순서대로 개선한다.

운영이 시작되면 기준이 뒤집힌다. 문서가 기준이던 관계가 데이터베이스가 기준인 관계로 바뀌며, 그때는 문서를 수작업으로 맞추는 대신 `export`로 다시 생성한다.

> 실제 시작점은 논리 모델이다
>
> 표준단어부터 순서대로 완성한 뒤 모델링을 시작하는 것이 아니다. 사전을 먼저 완성하려 하면 사용되지 않을 단어까지 등재하게 된다. 논리 모델을 작성하다가 필요한 표준용어가 없으면 그 시점에 등재하고 모델링으로 복귀한다. 표준 정의는 모델링과 별개의 준비 작업이 아니라 모델링이 요구하는 작업이다.

7.7.2 정기 점검과 CI 연계

한 번 통과한 산출물이 계속 통과하는 것은 아니다. 사전을 수정하면 그 사전을 참조하는 모든 산출물의 판정이 달라진다. 그러므로 점검을 사람의 기억이 아니라 일정과 자동화에 맡긴다.

표 7.27. 정기 점검의 주기와 명령

주기	무엇을	명령	판정
산출물 수정마다	문서와 문서의 대조	`validate ... -q`	`오류 0건`이면 정상
적재 이후	문서와 데이터베이스의 대조	`drift -s ... -q`	`차이 없음`이면 일치
매월 한 번 정도	표준 준수 수준의 추이	`diagnose -o 진단.xlsx`	전월과 비교한다
정기적으로	폐기 항목의 뒷정리	폐기 항목을 포함한 검사	고아 행과 잔존 참조를 찾는다

폐기 항목의 점검을 따로 두는 이유가 있다. 모든 규칙이 폐기 상태의 항목을 검사 대상에서 제외하므로, 폐기한 코드에 상세가 없거나 참조가 남아 있어도 일반 검사에서는 위반으로 보고되지 않는다. 정기 점검에서만 폐기 항목까지 포함하여 실행한다.

CI에 연결할 때에는 판정하는 명령과 측정하는 명령을 구분한다.

```
set -euo pipefail
kdsm validate DM-DST-00{1,2,3}_*.xlsx --meta <메타파일> -q
kdsm diagnose --schema <스키마> --min-score 95 -q
```

> 기준선은 지금 점수에서 시작한다
>
> 처음부터 높은 점수를 기준으로 설정하면 매번 실패하여 결과를 확인하지 않게 된다. 현재 점수를 기준선으로 설정하고 개선할 때마다 단계적으로 상향한다. 목적은 기준 통과가 아니라 상태가 악화되지 않도록 하는 것이다.

7.7.3 자주 걸리는 문제

증상이 아니라 원인으로 묶는다. 같은 원인에서 나오는 문제는 증상이 달라 보여도 예방 방법이 하나이기 때문이다.

같은 사실을 두 곳에 기재하면 어긋난다. 사람이 옮겨 기재하는 지점마다 불일치가 발생할 여지가 생긴다. 옮겨 기재할 수밖에 없는 지점을 먼저 파악하고 각각에 검증 규칙을 두는 것이 예방 방법이다.

표 7.28. 정리가 절반만 되었을 때 남는 것

상황	남는 것
표준코드 폐기	코드상세가 고아 행으로 남는다
테이블 이름 변경	테이블 정의서만 변경하고 컬럼 정의서는 그대로 남는다
컬럼 물리명 개명	그 컬럼을 가리키는 참조 컬럼 항목은 그대로 남는다
테이블 물리명 개명	메타파일 시트명이 이전 이름으로 남는다
개명 후 DDL 재실행	이전 이름의 테이블이 데이터를 포함한 채 남는다

검증은 통과하지만 데이터베이스가 거부한다. 검증 규칙은 정의서 안의 정합성을 보고 데이터베이스 제약은 실제 값을 본다. Null 허용 여부와 도메인 길이가 대표적이며, 두 규칙 모두 **산출물을 함께 지정해야 동작한다.** 지정하지 않으면 적재 시점에 데이터베이스가 거부한 뒤에야 확인하게 된다. 설명·표현형식·값범위처럼 내용을 기술하는 컬럼은 작성 시점의 값에 맞추어 길이를 정하지 말고 여유를 둔다. 길이 확대는 이미 적재한 데이터가 있는 상태에서 해야 하므로 처음에 넉넉히 잡는 편이 비용이 적다.

> ⚠ 자주 하는 실수
>
> - **엑셀을 열어 둔 채 실행** — 저장하지 않은 변경 사항은 프로그램이 읽지 못하며 오류도 발생하지 않는다. 결과를 보고 나서야 무엇이 반영되지 않았는지 확인하게 된다. 작업 전에 엑셀을 닫는다.
> - **용어구분을 비워 둠** — 비어 있으면 분류어 종결 검사와 도메인 등재 검사를 건너뛴다. 검사하지 않은 것은 통과한 것이 아니다.
> - **컬럼명을 일괄 치환** — `sed`로 컬럼명을 바꾸면 코드의 문자열 리터럴과 주석까지 함께 바뀐다. 치환 후에는 반드시 차이를 확인한다.
> - **통과한 규칙 수를 안전의 근거로 삼음** — 참조 사전을 지정하지 않으면 그 사전을 요구하는 규칙은 실행되지 않는다. 리포트에 어떤 규칙이 포함되었는지를 먼저 확인한다.

이름 하나를 변경할 때 함께 변경할 곳의 목록을 미리 만들어 두는 것이 가장 효과적인 예방책이다. 표준단어의 영문약어를 변경하면 그 단어를 쓰는 모든 표준용어의 물리명, 그 용어를 쓰는 모든 컬럼, 이미 생성한 테이블의 컬럼명, 메타파일 시트의 헤더가 함께 바뀐다. 변경을 시작하기 전에 목록을 작성하고, 마친 뒤에는 검증을 실행하여 빠진 곳이 없는지 확인한다.

CHAPTER 8

툴킷

8.1 개요

KDSM 툴킷은 KDSM 라이브러리(`kdsm`) 위에 세운 웹 애플리케이션이다. 표준 사전을 화면에서 조회하고, 쓰려는 이름의 표준용어 후보를 추천받아 제안하며, 표준 관리자가 그 제안을 판정하여 사전에 반영한다. 데이터 모델의 편집, 산출물의 생성과 적재, 운영 데이터베이스의 점검도 같은 화면에서 수행한다.

KDSM 개요(제1장)는 Support Layer의 소프트웨어를 **함수의 모음인 라이브러리**와 **사용자에게 UI/UX를 제공하는 툴킷**으로 나누었다. 라이브러리(7장)가 표준화 규칙을 기계가 수행할 수 있게 만든 것이라면, 툴킷은 그 규칙을 **조직의 여러 사람이 함께 사용하는 업무 절차**로 만든 것이다.

라이브러리만으로도 산출물을 검증하고 DDL과 ERD를 생성할 수 있다. 툴킷은 그 위에 세 가지를 더한다.

표 8.1. 툴킷이 라이브러리에 더하는 것

더하는 것	내용
공유	표준 사전을 조직의 모든 사용자가 같은 화면에서 조회한다. 사전은 `kdsm_meta` 스키마에 적재된 한 벌이므로 사용자마다 어느 판의 사전을 쓰는지 확인할 필요가 없다
통제	사전은 화면에서 직접 수정하지 않는다. 제안 권한이 있는 사용자가 등록·수정·폐기를 제안하고 표준승인 권한이 있는 사용자가 판정한다
기록	승인은 사전 반영과 검토 이력, 변경이력을 하나의 트랜잭션으로 남긴다. 검증·진단·대조·산출물 생성의 실행 기록은 작업 목록에 남는다

8.1.1 화면은 판단을 다시 하지 않는다

후보를 만들고, 규칙을 실행하고, DDL과 ERD를 생성하고, 진단 점수를 산출하는 판단은 모두 `kdsm` 라이브러리가 수행한다. 화면은 그 결과를 전달할 뿐 같은 판단을 따로 구현하지 않는다. 그러므로 화면의 추천·검증·진단은 같은 조건의 커맨드라인과 결과가 같다.

반대로 사용자·권한·제안·변경이력처럼 툴킷에만 있는 개념은 라이브러리에 두지 않는다. 커맨드라인만 쓰는 조직에게는 존재하지 않는 개념이기 때문이다. 의존은 한 방향으로만 흐른다.

표 8.2. 라이브러리와 툴킷의 분담

구분	라이브러리 (`kdsm`)	툴킷 (웹 애플리케이션)
담당	표준화 규칙 · 검증 · DDL · ERD · 진단 · CLI	화면 · 인증과 권한 · 제안과 판정 · 변경 이력
사용자	산출물을 작성하는 실무자, 애플리케이션 개발자	조직의 모든 표준화·모델링 참여자
산출물의 출처	사람이 작성한 엑셀 파일	메타테이블에 적재된 사전과 모델
의존	툴킷을 알지 못한다	라이브러리를 호출한다

이 원칙은 읽는 방법에도 영향을 준다. 화면이 보고한 위반이나 점수의 근거는 라이브러리 장(7장)에서 찾고, 본 장은 그 결과가 어느 화면의 어느 자리에 표시되며 결과를 보고 어디에서 조치하는지를 다룬다.

> **i 노트**
>
> 본 장은 툴킷의 기능을 방법론의 수행 절차에 대응시켜 요약한 것이다. 화면별 전체 조작과 서버 설정의 상세는 **데이터 표준화 애플리케이션 사용자 매뉴얼**(유충현 2026년b)에 수록되어 있다. 표준의 기준과 `kdsm` 명령의 상세는 **데이터 표준화 라이브러리 사용자 매뉴얼**(유충현 2026년a)이 정본이다.

8.2 툴킷 체계

8.2.1 화면 구성

화면은 상단·좌측·본문의 세 영역으로 구성된다. 상단에는 전역 검색창과 사용자 메뉴가 있고, 좌측 메뉴는 일곱 묶음이다.

표 8.3. 좌측 메뉴의 묶음과 항목

묶음	항목
표준 사전	표준단어 · 표준용어 · 도메인그룹 · 도메인분류 · 표준도메인 · 표준코드
용어 추천	제안 검토 · 변경이력
데이터 모델	주제영역 · 논리 · 물리 · ERD
산출물	생성 · 적재 · 초기 적재
점검	검증 · 진단 · 대조
작업	하위 항목이 없다
관리	계정 · 접속 대상

좌측 메뉴의 차례가 곧 작업의 차례이다. 사전을 세우고, 그 위에 모델을 만들고, 산출물을 생성하고, 마지막에 점검한다. 점검에서 발견한 문제는 앞의 차례로 돌아가 조치한다. 사전의 위반은 제안으로, 모델의 위반은 모델 편집으로 고친다. 작업과 관리는 이 차례의 밖에 있으며, 각각 실행 기록을 확인하는 자리와 계정·접속 대상을 다루는 자리이다.

8.2.2 프로세스 산출물–화면 매핑

방법론의 단계와 산출물을 툴킷의 화면에 대응시키면 다음과 같다. 라이브러리 장의 산출물–명령 매핑과 같은 축으로 정리하였으므로 두 표를 나란히 놓으면 명령과 화면의 대응을 함께 확인할 수 있다.

표 8.4. 프로세스 산출물–툴킷 화면 매핑

단계	산출물(정의서)	툴킷 화면	대응 명령
표준단어·표준용어 정의	데이터 표준 정의서(DM-DST-001)	표준 사전 · 용어 추천 · 제안 검토	`suggest` · `validate`
표준도메인·표준코드 정의	데이터 표준 정의서(DM-DST-001)	표준 사전 · 제안 검토	`validate`
주제영역·엔터티·속성 정의	논리 데이터 모델 정의서(DM-DST-002)	주제영역 · 논리	`validate`
데이터베이스·테이블·컬럼 정의	물리 데이터 모델 정의서(DM-DST-003)	물리(논리에서 생성 포함)	`validate`
ERD 작성	ERD	ERD	`erd`
DDL 생성	DDL 스크립트	물리 → DDL 생성	`ddl`
메타 적재 · 문서 재생성	—	산출물(생성 · 적재 · 초기 적재)	`export` · `apply` · `load`
배포 후 점검	—	점검(검증 · 진단 · 대조)	`validate` · `diagnose` · `drift`

8.2.3 다섯 권한

권한은 역할이 아니라 항목 단위로 부여하며 한 사용자가 여럿을 가질 수 있다. 코드값의 정본은 표준 코드 `AP_AUTHRT`이다.

표 8.5. 다섯 권한

권한	하는 일	대표 화면
조회	사전과 데이터 모델을 읽는다. 모든 사용자에게 부여한다	표준 사전 · 용어 추천 · 모델 조회 · DDL 생성 · ERD · 산출물 생성 · 점검
제안	표준 항목의 등록·수정·폐기를 제안한다	제안 등록
모델편집	주제영역과 논리·물리 모델을 편집하여 저장한다	주제영역 · 논리 · 물리 편집, 논리에서 생성
표준승인	제안을 승인하거나 반려하고, 산출물을 적재한다	판정 · 산출물 적재
시스템관리	계정과 권한, 접속 대상을 관리하고 메타파일을 초기 적재한다	계정 · 접속 대상 · 초기 적재

권한 사이에 상하 관계가 없다. 시스템관리 권한이 있어도 조회 권한이 없으면 사전을 읽지 못한다. 권한 판정은 요청마다 데이터베이스에서 수행하므로 부여하거나 회수하면 다음 요청부터 즉시 적용되며 재로그인이 필요하지 않다.

표준승인 권한은 한 사람의 승인으로 사전이 바뀌는 권한이므로 좁게 부여한다. 산출물 적재가 표준승인 권한을 요구하는 것도 같은 이유이다. 적재는 제안과 판정을 거치지 않고 사전과 모델에 반영되기 때문이다.

8.2.4 화면과 커맨드라인의 대응

표 8.6. 화면과 커맨드라인의 대응

하는 일	화면	커맨드라인	차이
용어 추천	용어 추천	`kdsm suggest --from-db`	화면은 적재된 사전만 사용한다
검증	점검 → 검증	`kdsm validate --from-db`	커맨드라인은 산출물 파일도 검사한다
DDL 생성	물리 → DDL 생성	`kdsm ddl`	화면은 메타테이블의 모델에서, 커맨드라인은 산출물에서 생성한다
ERD	데이터 모델 → ERD	`kdsm erd`	원천 세 가지는 같다
산출물 생성	산출물 → 생성	`kdsm export`	같다
산출물 적재	산출물 → 적재	`kdsm apply`	같다
메타파일 적재	산출물 → 초기 적재	`kdsm load`	같다
진단	점검 → 진단	`kdsm diagnose`	화면은 등록한 접속 대상을, 커맨드라인은 환경변수를 사용한다
대조	점검 → 대조	`kdsm drift`	화면은 메타테이블을, 커맨드라인은 산출물 파일을 정의서로 사용한다

커맨드라인에만 있는 옵션이 셋이다. 진단의 `--top`과 `--min-score`, 대조의 `--strict`이다. 뒤의 둘은 CI에 연결하는 용도이므로(7.7.2절) 화면에 두지 않는다.

8.3 데이터 표준 정의 지원

8.3.1 표준 사전 조회

좌측 메뉴 「표준 사전」의 여섯 화면은 적재된 사전을 읽기만 한다. 여섯 화면의 목록은 같은 표를 사용하므로 조작이 동일하다.

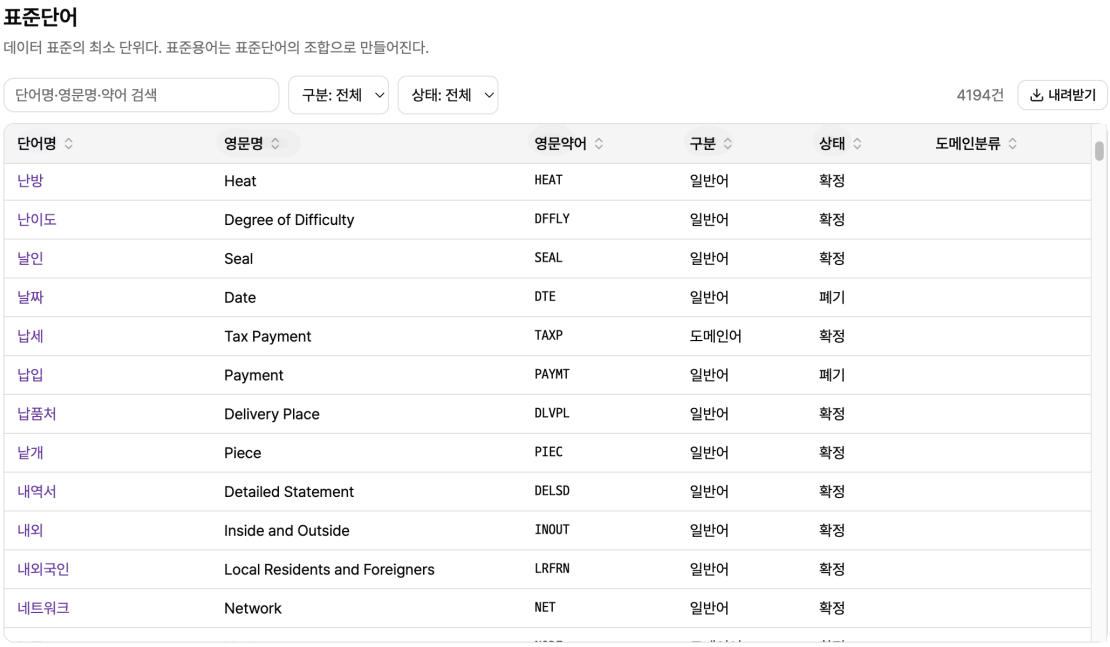

그림 8.1. 표준단어 목록. 검색창·필터·건수·내려받기가 표 위의 한 줄에 배치된다.

표 8.7. 목록 화면의 공통 조작

조작	동작
검색창	입력한 문자열이 포함된 행만 표시한다
필터	화면별로 구분·상태·그룹·분류를 제공한다
정렬	열 머리글을 선택할 때마다 정렬이 전환된다
건수	거른 결과가 전체와 다르면 「표시 건수 / 전체 건수」로 표시한다
내려받기	화면에 표시된 상태 그대로 CSV 파일을 생성한다

상세 화면은 서로 연결된다. 표준단어 상세에는 그 단어를 사용하는 표준용어 목록이 있어 단어의 수정이나 폐기가 무엇에 영향을 주는지 확인할 수 있다. 표준용어 상세는 구성 표준단어·표준도메인·표준코드의 상세로, 도메인그룹 상세는 소속 도메인분류로, 도메인분류 상세는 그 분류를 쓰는 표준도메인으로 이어진다. 이 경로가 모두 동작하면 사전이 온전히 적재된 것이다.

상단의 **전역 검색**은 Enter를 누르면 네 사전(표준단어·표준용어·표준도메인·표준코드)을 한 번에 조회한다. 이름·영문명·약어·물리명·코드값명을 대조하며 설명 본문은 대조하지 않는다. 설명까지 대조하면 흔한 낱말 하나로 수백 건이 표시되어 상세로 이동하는 경로로 쓸 수 없기 때문이다. 결과는 사전별로 20건까지 표시되므로 전량을 확인할 때에는 해당 사전의 목록 화면을 사용한다.

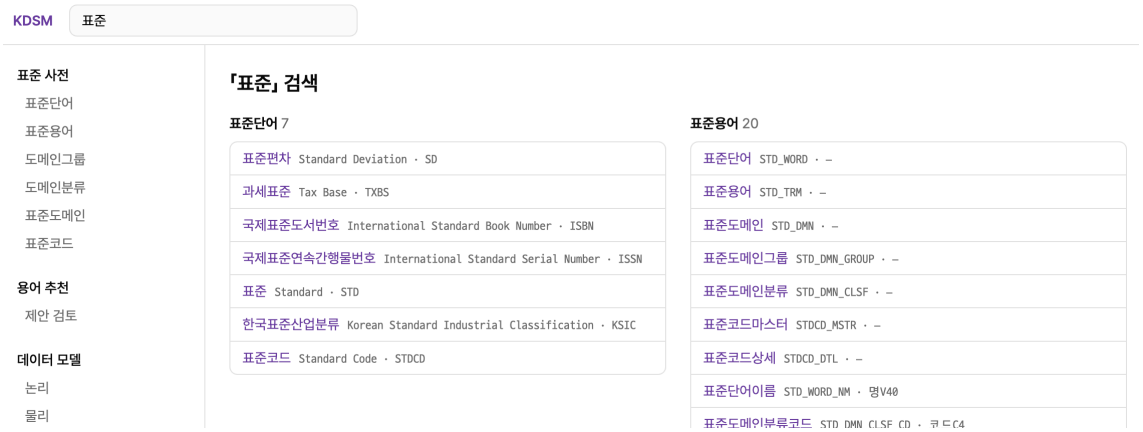

그림 8.2. 전역 검색 결과. 사전별로 묶여 표시되고 각 줄이 상세로 연결된다.

8.3.2 용어 추천

「용어 추천」은 새 표준용어와 새 표준단어가 사전에 들어가는 출발점이다. 후보를 만드는 판단은 라이브러리의 추천기(7.4.1절)가 수행하므로 점수 체계와 금지어 처리는 커맨드라인과 같다.

표 8.8. 용어 추천의 다섯 단계

단계	하는 일
1	쓰려는 이름을 입력하고 「추천」을 선택한다. 띄어쓰기는 제거한 뒤 대조한다
2	같은 이름의 표준용어가 이미 있는지 확인한다. 있으면 물리명과 표준도메인이 먼저 표시된다
3	점수가 높은 차례로 표시되는 최대 5건의 후보에서 하나를 선택한다
4	사전에 없는 조각이 있으면 표준단어를 먼저 제안한다
5	사유를 기재하고 표준용어 제안을 등록한다

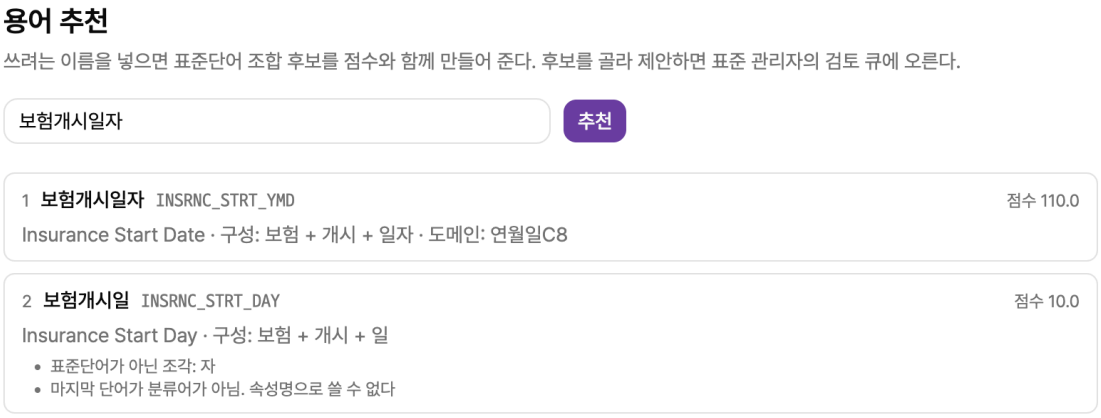

그림 8.3. 용어 추천 결과. 후보마다 점수와 구성 표준단어, 표준도메인이 함께 표시된다.

후보 한 줄은 용어명·물리명·점수·영문명·구성 표준단어·표준도메인으로 이루어지며, 점수를 만든 근거가 후보 아래에 함께 표시된다. 금지어가 섞인 이름은 대체어로 바꾼 후보임을, 사전에 없는 조각을 쓰는 후보는 표준단어 등재가 필요함을 알린다. 후보를 하나도 만들지 못하면 그 사유를 표시한다. 빈 목록만 표시하면 사전에 없는 말인지 사용할 수 없는 말인지 구별할 수 없기 때문이다.

8.3.2.1 등재 초안과 두 단계 제안

사전의 표준단어로 나눌 수 없는 조각이 있으면「등재 초안 만들기」단추가 표시된다. 초안은 LLM(AWS Bedrock 또는 Claude API)(Amazon Web Services 2026년)이 작성하며 단어명·영문명·영문약어·구분·설명·근거로 이루어지고, 규칙 검사를 통과하지 못한 항목이 있으면 그 사유가 함께 표시된다. LLM 호출에는 비용과 대기 시간이 있으므로 초안은 단추를 선택하였을 때에만 생성한다.



그림 8.4. 표준단어 등재 초안. 초안마다 규칙 검사의 결과가 함께 표시된다.

> ! 초안의 상태는 언제나 후보이다
>
> LLM이 무엇을 제안하든 등재 여부는 사람이 판정한다. 초안은 제안의 내용을 채우는 재료이며 그 자체로 사전에 들어가지 않는다.

사전에 없는 조각을 쓰는 후보를 선택하면 표준용어 제안 폼이 아니라 **표준단어 등재 제안 폼**이 열린다. 단어와 용어를 한 번에 제안하지 않는다. 단어가 반려되면 그 단어에 기댄 용어 제안이 성립하지 않으므로, 승인되지 않을 제안이 검토 목록에 쌓이지 않도록 두 단계로 나눈 것이다. 표준단어가 승인된 뒤 「추천 다시 받기」를 선택하면 그 단어를 포함한 정식 후보가 표시되고 표준용어를 제안할 수 있다.

표준용어 제안 폼에는 용어 구분(속성명·엔터티명), 표준도메인(후보가 둘 이상일 때만), 제안 사유(필수), 용어 설명(선택)을 기재한다. 용어명·물리명·영문명·구성 표준단어는 선택한 후보의 값을 그대로 사용하므로 폼에서 고치지 않는다.

8.3.3 제안과 판정

표준 사전의 항목은 화면에서 직접 수정되지 않고 **제안과 판정**을 거쳐 바뀐다. 대상은 표준 사전의 여섯 가지이고 유형은 등록·수정·폐기의 세 가지이다.

표 8.9. 제안의 세 가지 유형

유형	작성하는 자리	유의 사항
등록	각 목록 화면의 「등록 제안」, 표준용어는 용어 추천 화면	이미 있는 이름이면 알리지만 막지 않는다. 승인 시점에 옛 행을 덮는다
수정	각 상세 화면의 「수정 제안」	이름(기본키)은 고치지 않는다. 이름을 바꾸려면 등록과 폐기를 각각 제안한다
폐기	각 상세 화면의 「폐기 제안」	행을 삭제하지 않고 상태를 폐기로 바꾼다. 이전 내용은 변경이력에 남는다

> ⚠ 표준코드의 코드값 목록은 한 벌을 대신한다
>
> 표준코드의 수정 폼에 있는 코드값 목록은 승인 시점에 기존 목록을 대신한다. 목록에서 지운 코드값은 승인될 때 폐기되며, 몇 건이 폐기되는지 보내기 전에 화면에 표시된다. 그 건수가 의도한 것인지 확인한 뒤에 등록한다.

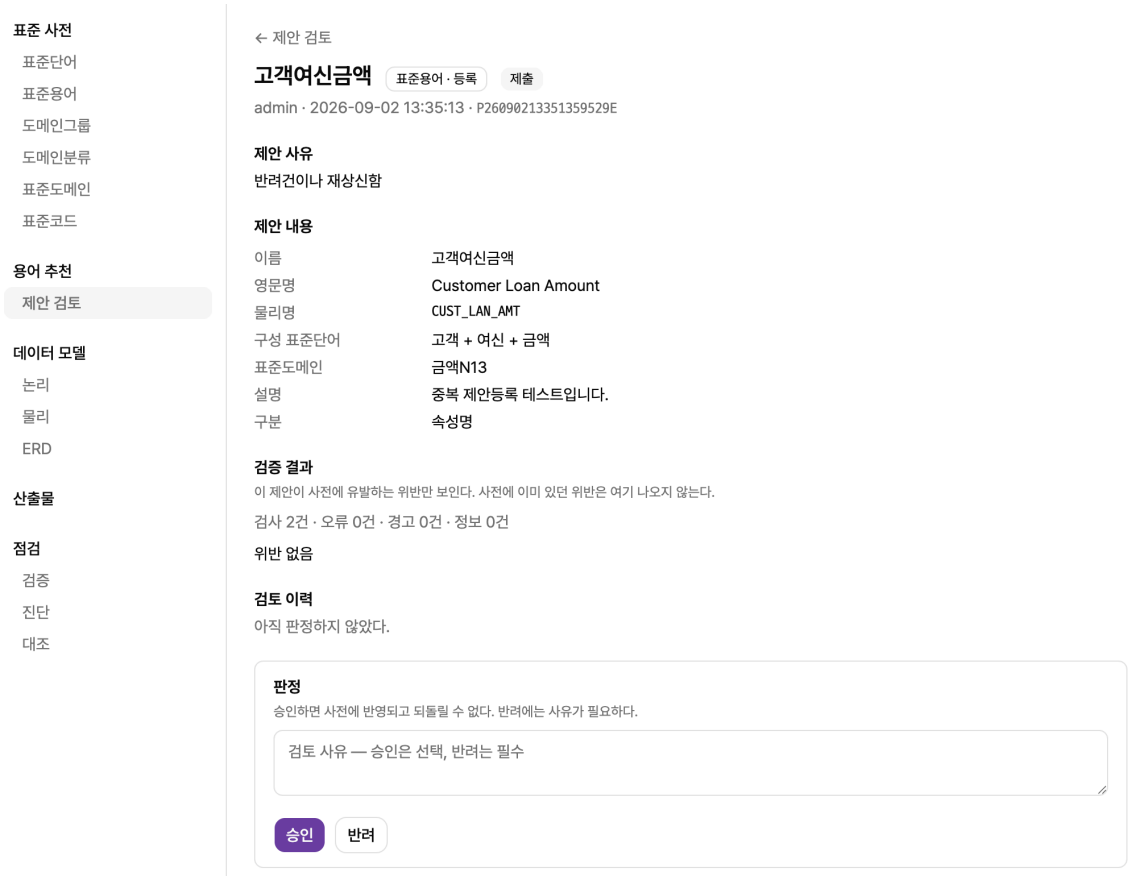

그림 8.5. 제안 상세. 제안 내용과 검증 결과, 검토 이력, 판정이 한 화면에 놓인다.

제안 상세는 머리·제안 사유·제안 내용·검증 결과·검토 이력의 다섯 부분으로 구성되며, 판정은 상세의 아래쪽에서 수행한다.

8.3.3.1 제안의 검증 결과

검증 결과에는 **이 제안이 새로 만드는 위반만** 표시된다. 사전에 이미 있던 위반은 제외하므로 판정하는 사람은 이 제안이 무엇을 어긋나게 하는지만 확인한다.

표 8.10. 제안의 검증 결과를 산출하는 방법

유형	판정하는 방법
등록 · 수정	제안 내용을 사전에 더한 상태로 규칙을 실행하고, 사전에 이미 있던 위반을 제외한다
폐기	지금의 사전과 그 항목만 폐기한 사전을 각각 검사하여 늘어난 위반만 남긴다

판정을 돕는 정보가 두 가지 함께 표시된다. 표준용어의 폐기에는 그 용어를 사용하는 엔터티·속성·컬럼의 수가, 표준코드의 수정에는 폐기되는 코드값의 건수가 표시된다. 해당하는 것이 없으면 없다고 표시한다. 세어 보지 않은 것과 구별하기 위해서이다.

8.3.3.2 판정

판정에는 표준승인 권한이 필요하다. 승인의 사유는 선택이고 반려의 사유는 필수이다. 반려된 제안은 제안자가 고쳐 다시 제출할 수 있다. 같은 대상의 같은 이름으로 제출 상태의 제안이 남아 있는 동안에는 새 제안이 거부되며 검토 중인 제안의 번호가 함께 표시된다.

> ! 승인은 한 트랜잭션이다
>
> 승인은 표준 사전(`kdsm_meta`) 반영, 검토 이력, 변경이력, 제안 상태 변경을 한 트랜잭션으로 수행한다. 어느 하나가 실패하면 전부 되돌아가 제안이 제출 상태로 남고 그 사유가 화면에 표시된다. 그러므로 승인 상태의 제안은 사전에 반영된 것이며 중간 상태가 없다. 서버를 재시작하지 않아도 다음 요청부터 반영된다.

8.3.3.3 변경이력

승인이 표준 사전을 바꿀 때마다 변경이력이 한 행 기록된다. 「용어 추천 → 변경이력」에서 사전이 지금의 내용이 된 경위를 확인하며, 각 행은 그 변경을 일으킨 제안으로 연결된다. 상세는 변경전과 변경후의 전체를 나란히 표시하지 않고 **값이 서로 다른 항목만** 항목·전·후의 세 열로 추려 표시한다.

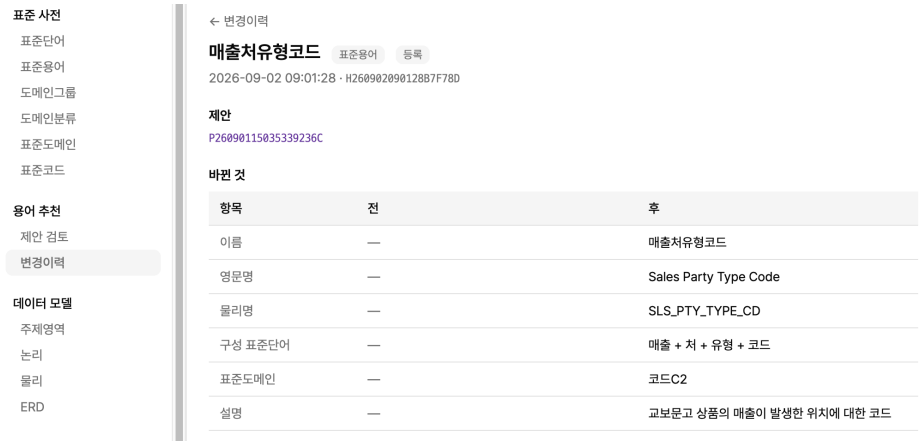

그림 8.6. 변경이력 상세. 변경된 항목만 전후로 대조하여 표시한다.

변경이력에는 유형을 기재하는 항목이 없다. 변경전이 없으면 등록, 변경후가 없으면 폐기, 양쪽이 모두 있으면 수정으로 판별한다. 변경이력은 조회만 할 수 있으며 수정하거나 삭제하는 기능은 화면과 서버 모두 제공하지 않는다.

8.4 데이터 모델 정의 지원

8.4.1 사전과 모델의 변경 경로

표준 사전은 제안과 판정을 거쳐 바뀌지만 데이터 모델은 **모델편집 권한으로 표를 편집하여 즉시 저장**한다. 사전은 조직 전체가 공유하는 기준이고 모델은 그 기준을 적용한 결과물이기 때문이다. 대신 모델의 저장은 표준 검증 규칙을 통과하여야 한다.

표 8.11. 사전과 모델의 변경 경로 비교

구분	표준 사전	데이터 모델
변경 방법	제안 → 판정	표 편집 → 저장
요구 권한	제안 · 표준승인	모델편집
검증 시점	판정 전 제안 상세에서 확인	저장 시점에 검사하고 위반이 있으면 거부
이력	변경이력에 기록	수정 시각과 수정자를 기록

8.4.2 표 편집기의 공통 규칙

주제영역·배정·엔터티·속성·테이블·컬럼의 여섯 가지 편집이 같은 편집기를 사용한다. 행 추가·삭제·이동·되돌리기와 함께, 스프레드시트에서 복사한 범위를 격자 단위로 붙여넣을 수 있다. 변경한 행은 순번 칸의 좌측 선과 굵은 글자로, 추가한 행은 기울임까지 함께 표시한다. 색만으로 구분하지 않으므로 색을 판별하기 어려운 환경에서도 확인할 수 있다.

저장에는 세 가지 규칙이 공통으로 적용된다.

1. **표 전체를 한 번에 저장한다.** 행 단위로 전달하지 않으므로 삭제한 행과 이동한 차례가 함께 반영된다.
2. **위반이 하나라도 있으면 표 전체가 저장되지 않는다.** 위반한 칸에 테두리가 표시되고 사유가 말풍선으로 표시된다. 위반이 없는 행만 저장하면 화면에 보이는 것과 적재된 것이 달라지기 때문이다.
3. **먼저 저장한 쪽이 반영된다.** 편집을 시작하는 시점에 수정 시각을 함께 읽어 두고, 그 사이에 다른 사람이 같은 표를 저장하였으면 거부한다. 「다시 불러오기」로 최신 상태를 받은 뒤 다시 편집한다.

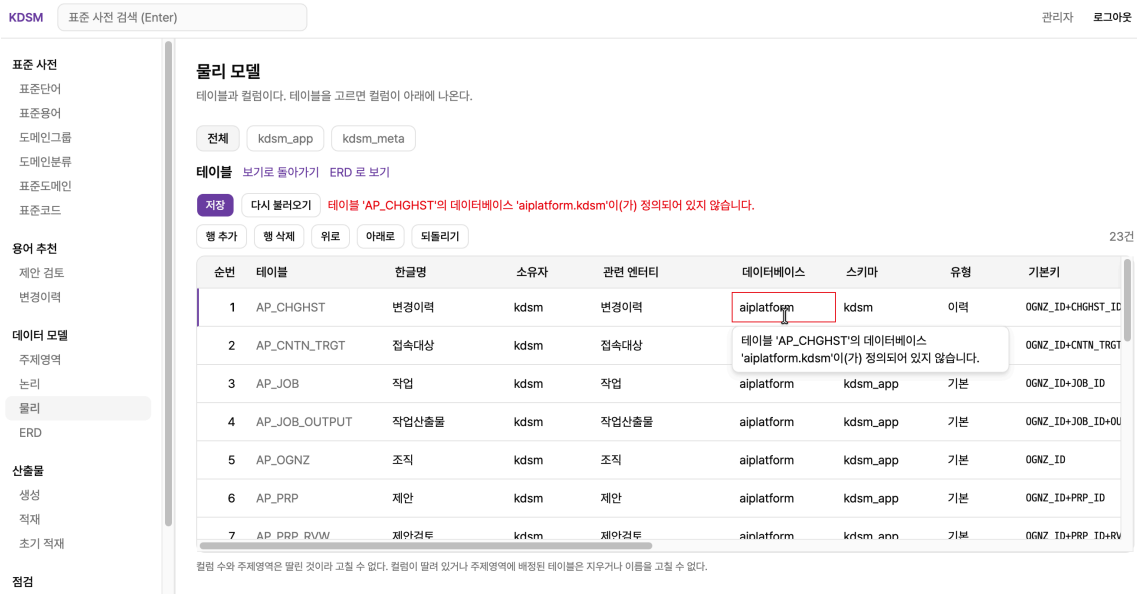

그림 8.7. 저장이 거부된 표. 위반한 칸에 테두리가 표시되고 사유가 말풍선으로 표시된다.

> ! 기존 항목의 이름은 변경할 수 없다
>
> 주제영역명·엔터티명·테이블 물리명은 새로 추가한 행에서만 입력할 수 있다. 배정·속성·컬럼이 이름으로 그 항목을 참조하므로 이름을 바꾸면 참조가 끊어진다. 이름을 바꾸려면 새 항목을 추가하고 하위 항목을 옮긴 다음 이전 항목을 제거한다.

8.4.3 주제영역

주제영역은 논리 데이터 모델을 업무 성격에 따라 구분한 최상위 묶음이다. 목록은 차례·주제영역·엔터티 수·테이블 수·설명을 표시하고, 상세는 그 영역에 배정된 엔터티와 테이블을 구분하여 표시한다.

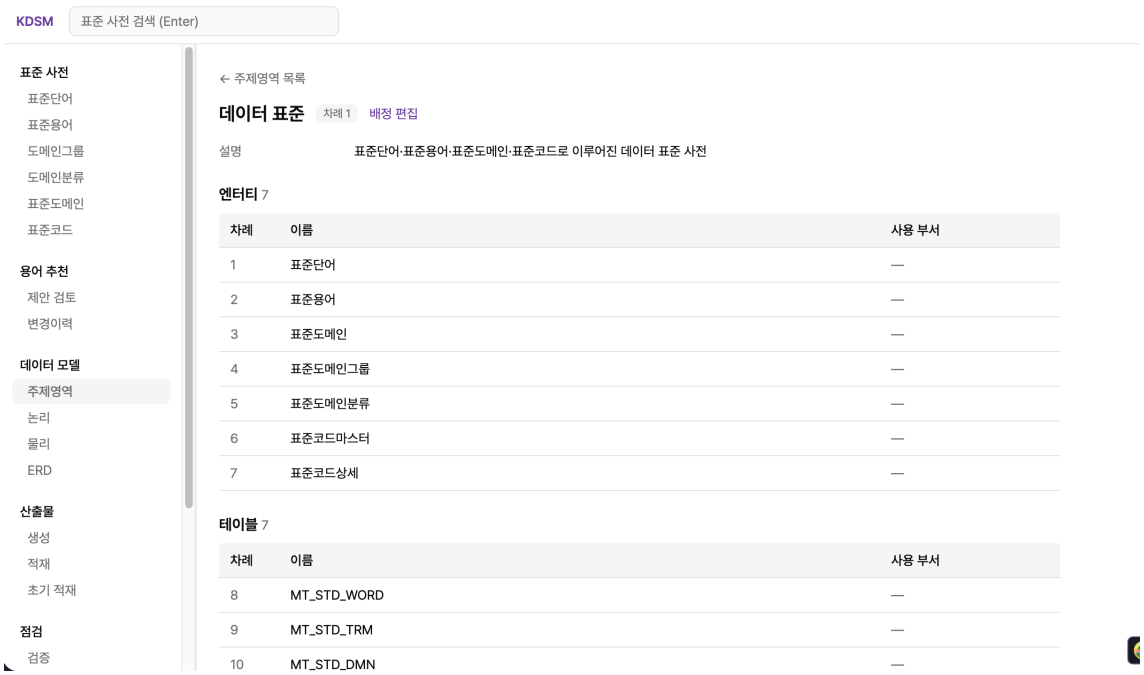

그림 8.8. 주제영역 상세. 설명 아래에 엔터티와 테이블의 배정 목록이 구분되어 표시된다.

영역 자체의 추가·제거와 차례·설명의 변경은 목록의 「편집」으로, 엔터티와 테이블의 배정은 상세의 「배정 편집」으로 수행한다. 목록 편집의 차례 열은 직접 입력하는 값이지만 배정 편집에는 차례 열이 없고 행의 자리가 그대로 차례가 된다. 배정이 있는 영역은 제거할 수 없다.

배정은 세 가지 경로로 메타테이블에 반영된다.

표 8.12. 주제영역 배정의 세 경로

경로	반영하는 것
산출물·메타파일 적재	논리 데이터 모델 정의서의 주제영역과 주제영역구성 시트를 함께 반영한다
논리 모델에서 생성	엔터티가 속한 영역을 생성된 테이블이 물려받는다
배정 편집	영역 하나의 배정을 전부 교체한다

> **i 배정의 누락은 적재 시점에 보고된다**
>
> 모든 엔터티와 테이블이 어느 영역엔가 배정되어야 한다는 규칙은 모든 영역의 배정을 보아야 판단할 수 있다. 한 영역을 편집하는 사람이 다른 영역의 누락으로 저장을 거부당하지 않도록, 배정 편집은 이 규칙을 검사하지 않는다. 누락은 산출물을 적재하는 시점에 보고되며, 엔터티의 누락은 적재를 중단시키고 테이블의 누락은 경고로 보고된다.

8.4.4 논리 모델

논리 모델 화면은 위에 엔터티 목록, 아래에 선택한 엔터티의 속성 목록을 표시한다. 조회용 표와 편집용 표를 따로 둔다. 정렬이 적용된 상태에서 행을 이동하면 「위로」가 가리키는 자리가 확정되지 않기 때문이다.

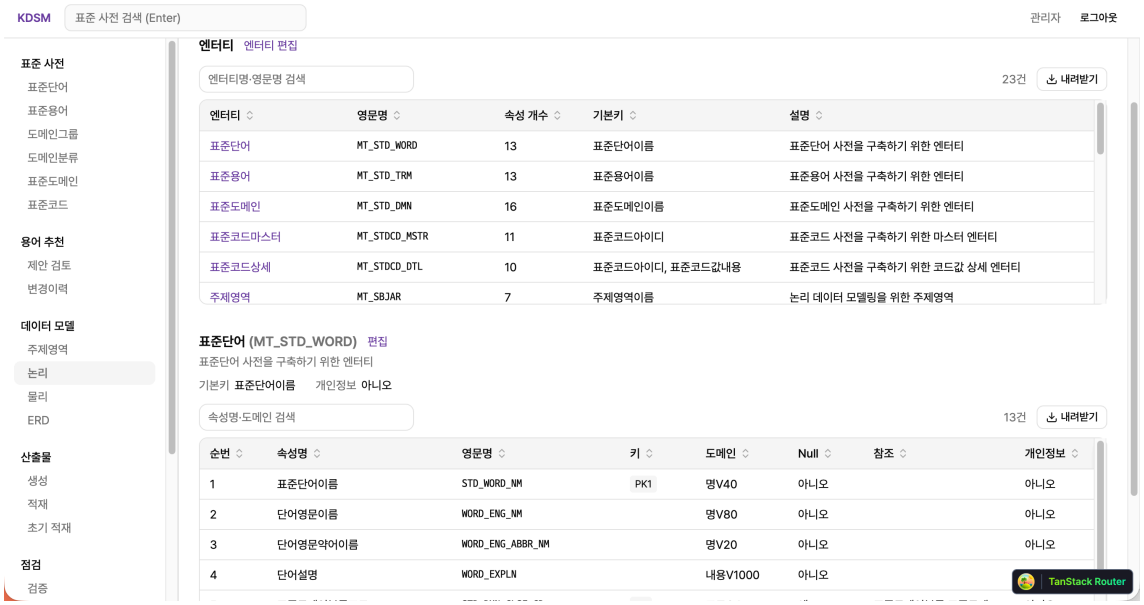

그림 8.9. 논리 모델 화면. 위가 엔터티 목록이고 아래가 선택한 엔터티의 속성 목록이다.

표 8.13. 논리 모델 편집의 열

편집	열	입력 방식의 특징
엔터티	엔터티·영문명·구분·기본키·유일키·상위 엔터티·개인정보·주요 관계·설명	구분은 표준코드 `ENTTY_CLSF`, 상위 엔터티는 같은 모델의 엔터티에서 선택한다
속성	속성명·영문명·유형·도메인·PK순번·유일키·Null·개인정보·참조 엔터티·참조 속성·기본값·설명	유형은 `ATRB_TYPE`, 도메인은 표준도메인에서 선택한다. 참조 속성은 입력하는 값이다

속성의 순번은 입력하는 값이 아니라 표에 놓인 차례이며 저장할 때 다시 매겨진다. 엔터티를 편집하는 동안에는 속성이 표시되지 않는다. 엔터티명이 바뀌는 중에 속성이 어느 엔터티를 가리키는지 구별되지 않기 때문이다.

8.4.5 물리 모델

물리 모델 화면은 위에 테이블 목록, 아래에 선택한 테이블의 컬럼 목록을 표시한다. 테이블 목록은 스키마로 거를 수 있으며, 선택지는 테이블이 실제로 속한 스키마에서 생성된다. 테이블 목록의 주제영역 열이 비어 있으면 배정이 없는 것이다.

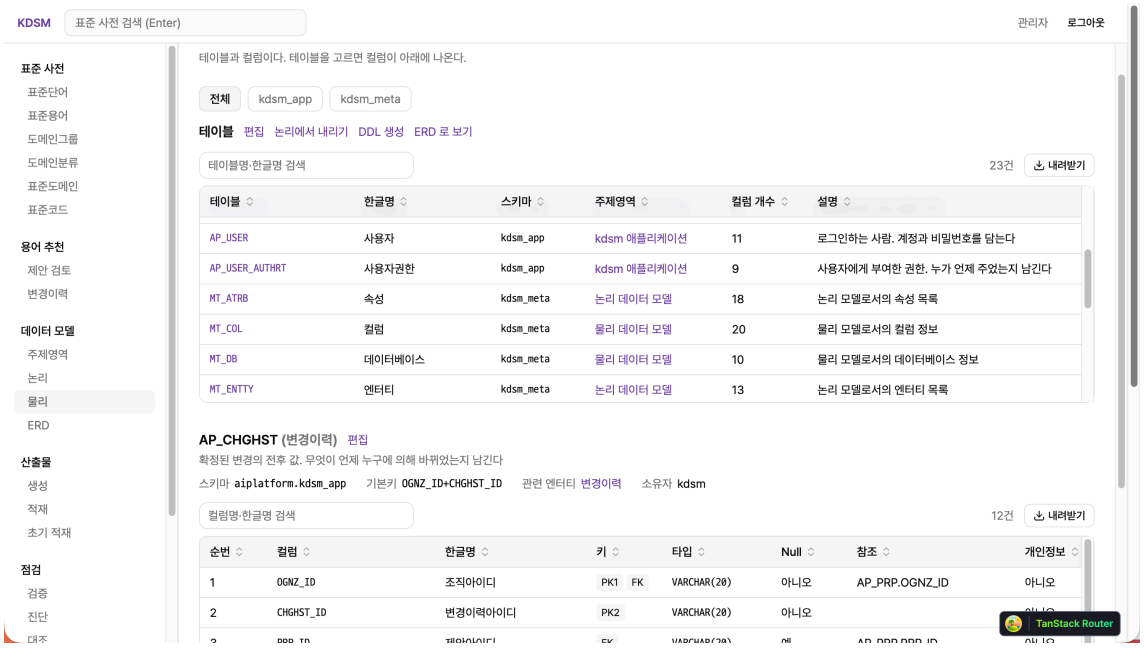

그림 8.10. 물리 모델 화면. 위가 테이블 목록이고 아래가 선택한 테이블의 컬럼 목록이다.

> ! 데이터 타입과 길이는 컬럼 물리명이 정한다
>
> 컬럼에는 표준도메인을 지정하는 항목이 없다. 컬럼 물리명이 표준용어를 정하고 그 용어의 표준도메인이 타입과 길이를 정한다. 물리명 칸을 수정하면 데이터 타입과 길이가 그 값으로 교체된다. 같은 사실을 두 곳에 기록하지 않는다는 원칙(7.7.3절)을 화면에서도 그대로 따른다.
>
> 물리명이 표준용어에 없으면 두 칸은 그대로 유지된다. 사전에 없다는 사실은 저장 시점의 검증이 보고하며, 그때까지 기재해 둔 값을 화면이 제거하지 않는다.

테이블 편집은 모델 전체에 하나인 표를 다룬다. 컬럼이 속해 있거나 주제영역에 배정된 테이블은 제거할 수 없으며, 그 위반은 표 위에 문장의 목록으로 표시된다. 표 편집으로 추가한 테이블에는 주제영역이 배정되지 않으므로 배정 편집으로 지정한다.

8.4.6 논리 모델에서 생성

「논리에서 내리기」는 엔터티를 테이블로, 속성을 컬럼으로 생성한다. 한 번에 여러 테이블을 변경하는 작업이므로 **미리 보기 → 선택 적용**의 두 걸음으로 수행한다.

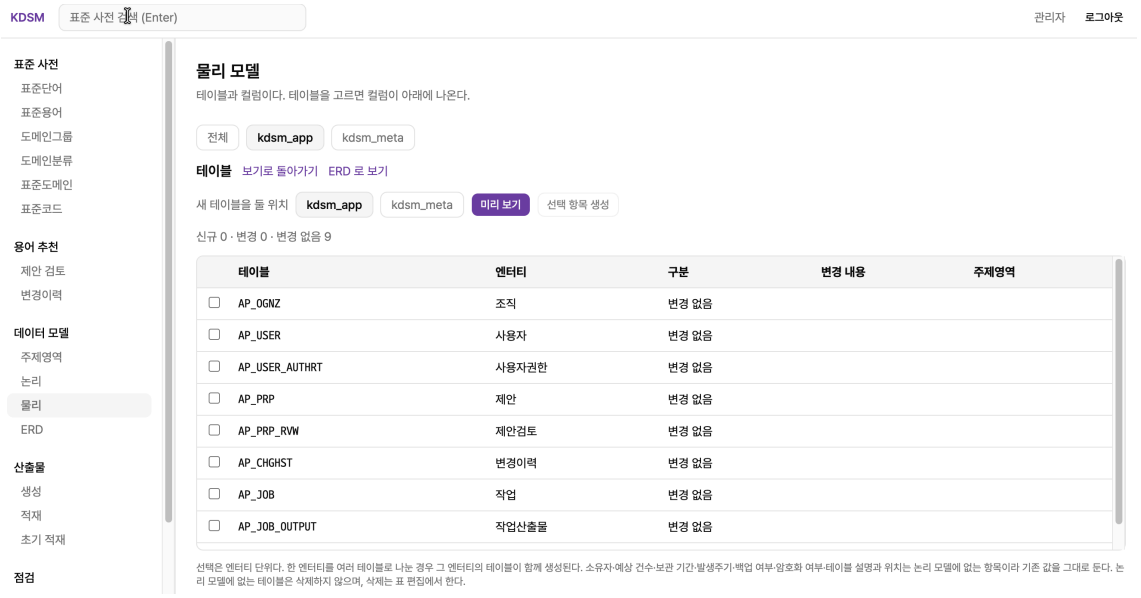

그림 8.11. 논리 모델에서 생성. 미리 보기가 대상과 변경 내용을 표시하고, 확인한 뒤에 선택한 것만 적용한다.

표 8.14. 논리 모델에서 생성하는 절차

단계	하는 일
1	새 테이블을 둘 데이터베이스와 스키마를 선택한다. 엔터티에는 위치를 기재하는 항목이 없기 때문이다
2	「미리 보기」로 테이블마다 신규·변경·변경 없음과 변경 항목의 수를 확인한다
3	적용할 엔터티를 선택한다. 변경이 있는 행만 처음에 선택되어 있다
4	「선택 항목 생성」으로 적용한다. 엔터티가 속한 주제영역을 테이블이 물려받는다

생성하지 못한 자리는 「확인 필요」로 표시되고 비운 상태로 생성된다. 이름을 임의로 지어 채우지 않는다. 소유자·예상 건수·보관 기간·발생주기·백업 여부·암호화 여부처럼 논리 모델에 대응하는 항목이 없는 것은 생성 대상이 아니며 이미 있는 값이 유지된다. 논리 모델에 없는 테이블은 제거하지 않는다.

> **i 미리 보기에서 본 것이 그대로 적용된다**
>
> 적용 요청에는 선택한 엔터티의 이름만 전달하고 표의 내용은 전달하지 않는다. 서버가 같은 것을 다시 생성하여 적용하므로 확인한 것과 적용된 것이 달라지지 않는다.

8.4.7 DDL 생성

「DDL 생성」은 물리 모델에서 스키마 하나의 DDL을 생성하여 확인하고 `.sql` 파일로 내려받는다. 생성은 라이브러리의 DDL 생성기(7.5.1절)가 수행하며, 메타테이블을 변경하지 않으므로 조회 권한으로 수행한다.

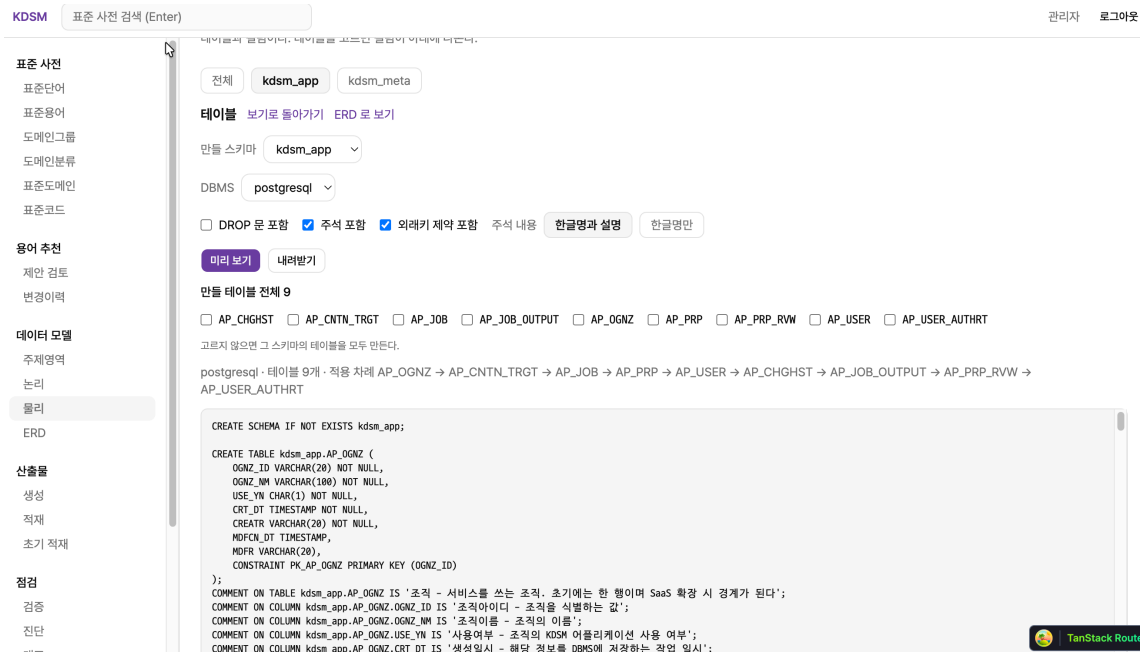

그림 8.12. DDL 생성. 범위와 옵션을 지정하여 확인한 뒤에 파일로 받는다.

표 8.15. DDL 생성의 지정 항목

지정하는 것	내용
만들 스키마	생성 범위. 여러 스키마를 한 번에 생성하지 않는다
만들 테이블	하나도 선택하지 않은 상태가 그 스키마의 전체이다
DBMS	데이터베이스 정의의 「DBMS 종류·버전」에서 판별한다. 판별하지 못하면 비워 두고 사용자가 선택한다
DROP 문 포함	기본값은 해제이다
주석 포함 · 주석 내용	기본값은 포함, 한글명과 설명이다
외래키 제약 포함	기본값은 포함이다. 포함하면 적용 차례를 외래키가 정한다
카탈로그 · `STRING`	Databricks를 선택하였을 때만 표시된다

선택하지 않은 상태를 전체로 해석하는 것은, 화면이 전체를 선택한 목록을 전달하면 화면을 연 뒤에 추가된 테이블이 안내 없이 누락되기 때문이다. DBMS를 판별하지 못하였을 때 임의의 DBMS로 생성하지 않는 것도 같은 취지이다.

> ! 생성한 DDL을 화면에서 실행하지 않는다
>
> 이 화면은 SQL을 작성하여 보여 주고 파일로 전달하는 데까지가 범위이다. 화면의 조작 한 번이 운영 스키마를 변경하는 경로는 복구하기 어렵기 때문이다. 내려받은 파일을 검토한 뒤에 적용한다. 테크니컬 노트가 제시한 「정의서에서 생성 → 사람은 검토만」의 원칙(6.4.5.2절)과 같다.

8.4.8 ERD

ERD 화면은 라이브러리가 조립한 Mermaid(Sveidqvist 와/과 Mermaid contributors 2026년) 본문을 받아 브라우저에서 그린다. 원천은 라이브러리(7.5.3절)와 같은 세 가지이다.

표 8.16. ERD의 세 가지 대상

대상	읽는 것	방향	도면
KDSM 메타	적재된 메타테이블	리버스 엔지니어링	스키마를 선택하여 주제영역마다 한 장과 전체 한 장
데이터베이스	등록한 접속 대상의 카탈로그	리버스 엔지니어링	물리 도면 한 장
산출물	업로드한 논리·물리 데이터 모델 정의서	포워드 엔지니어링	업로드한 모델의 도면

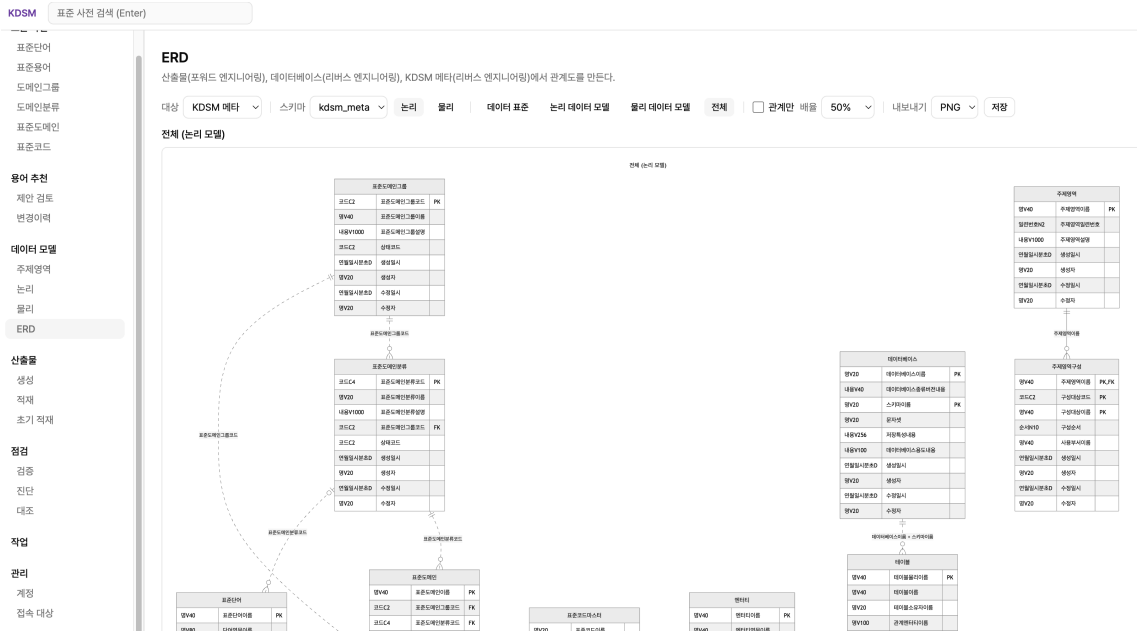

그림 8.13. ERD 화면. 대상이 「KDSM 메타」이고 스키마를 선택하여 도면을 표시한 상태이다.

KDSM 메타의 도면은 적재된 주제영역 배정으로 나뉘므로 배정을 바꾸면 도면의 구성도 바뀐다. 데이터베이스 대상은 외래키가 설정되지 않은 관계를 그리지 않는다. 정의서에 관계가 있어도 제약이 없으면 관계선이 생기지 않으며, 그 차이를 항목 단위로 확인하는 기능은 대조(8.6.3절)이다. 업로드한 산출물은 도면을 그리는 동안에만 사용하고 서버에 남기지 않는다.

「관계만」을 선택하면 속성을 접고 관계선만 표시하며, 배율은 50~200%의 다섯 단계이다. 내보내기(PNG·SVG)는 화면에 표시된 도면을 그대로 저장한다. 정의서는 식별관계로 기재하였으나 기본키 구조는 식별관계가 아닌 항목이 있으면 도면 아래에 목록으로 표시되며, 어느 쪽을 따를지는 사용자가 판단한다.

8.5 산출물의 생성과 적재

8.5.1 생성

「산출물 → 생성」은 적재된 메타테이블을 읽어 DM-DST 산출물 3종을 생성한다. 라이브러리의 `kdsm export`(7.5.4절)와 같은 결과이다.

표 8.17. 생성하는 산출물 3종

종류	문서번호	읽는 대상
데이터 표준 정의	DM-DST-001	표준단어 · 표준용어 · 표준도메인 · 도메인그룹 · 도메인분류 · 표준코드
논리 데이터 모델	DM-DST-002	주제영역 · 주제영역구성 · 엔터티 · 속성
물리 데이터 모델	DM-DST-003	데이터베이스 · 테이블 · 컬럼

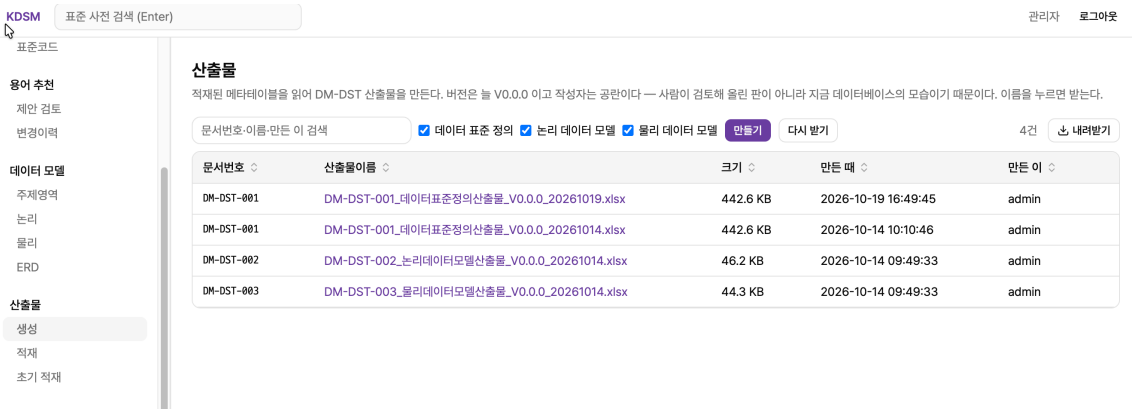

그림 8.14. 산출물 화면. 생성할 종류를 선택하고 생성한 산출물이 목록에 표시된 상태이다.

생성한 산출물의 표지 버전은 항상 `0.0.0`이고 작성자는 비어 있다. 검토를 거쳐 확정한 것이 아니라 그 시점의 메타테이블을 옮긴 것이기 때문이다. 배포할 산출물로 사용할 때에는 표지를 작성하여 버전을 부여한다.

생성한 파일은 서버의 `KDSM_OUTPUT_DIR`에 작업 단위로 보관되고 목록에서 내려받는다. 보존 기간은 `KDSM_OUTPUT_KEEP_DAYS`(기본값 7일)가 정하며, 기간이 지난 산출물은 다음 생성 요청 시점에 파일과 목록의 행이 함께 제거된다.

8.5.2 적재와 초기 적재

적재는 두 화면으로 나뉜다. 라이브러리의 적재 두 갈래(7.5.2절)와 같은 구분이다.

표 8.18. 적재와 초기 적재

화면	올리는 파일	대응 명령	검증 범위	요구 권한
적재	작성한 DM-DST 산출물(여러 개 가능)	`kdsm apply`	적재된 메타테이블과 대조한다	표준승인
초기 적재	엑셀 메타파일(한 번에 하나)	`kdsm load`	올린 파일 안에서만 검사한다	시스템관리

두 화면 모두 **미리보기 → 적재**의 두 단계로 수행한다. 처음에는 「미리보기」만 표시되고, 미리본 결과에 반영할 내용이 있을 때에만 「이대로 적재」가 나타난다. 미리보기를 거치지 않는 경로는 제공하지 않는다. 적재는 제안과 판정을 거치지 않고 사전과 모델에 반영되기 때문이다.

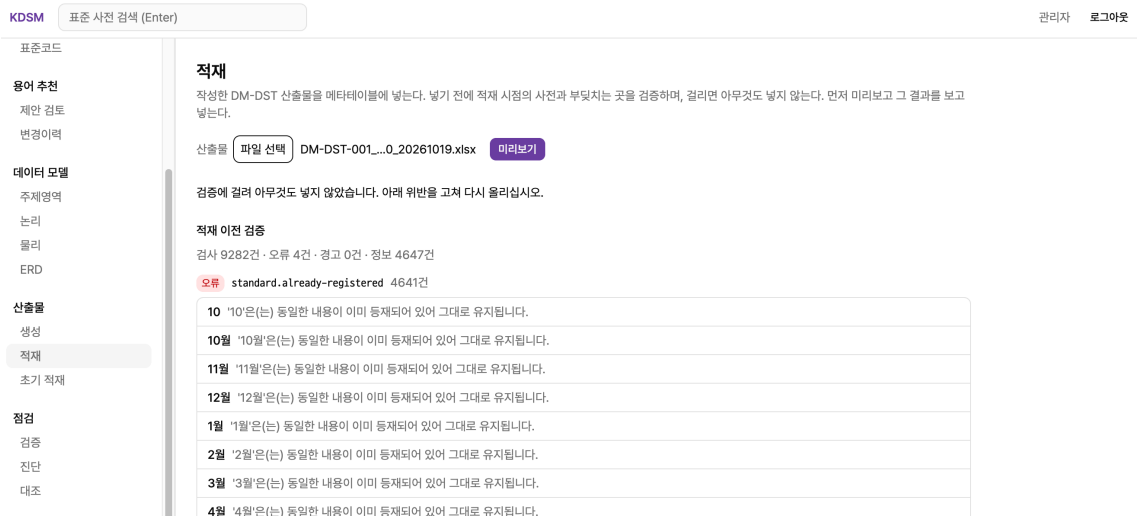

그림 8.15. 적재 화면. 산출물을 선택하여 미리보기를 실행한 상태이다.

적재는 반영하기 전에 검증을 수행하고 위반이 있으면 아무것도 반영하지 않는다. 한 트랜잭션에서 수행하므로 일부만 반영된 상태가 남지 않는다. 논리와 물리 산출물을 함께 올리면 엔터티와 테이블, 속성과 컬럼의 대조도 검증에 포함된다. 결과는 적재 완료·미리보기·검증 위반·대상 없음의 네 가지로 구분하며, 앞의 둘에만 시트 수와 행의 합계를 표시한다. 0으로 표시하면 적재할 대상이 없었던 것과 적재하지 못한 것이 구별되지 않기 때문이다.

> **i 파일을 다시 선택하면 앞의 결과가 제거된다**
>
> 선택한 파일을 바꾸면 앞서 미리본 결과가 화면에서 제거된다. 남겨 두면 다른 파일을 미리본 결과로 적재가 수행되어 확인한 것과 적재한 것이 달라진다.

초기 적재는 표준 사전 → 논리 모델 → 물리 모델의 차례로 올린다. 외래키가 참조하는 대상이 먼저 적재되어야 하기 때문이다. 서버를 기동하기 전에 사전을 세우는 일은 커맨드라인(7.3.4절)이 담당한다.

8.6 점검

좌측 메뉴 「점검」의 검증·진단·대조와 「작업」은 모두 **보고만 수행**한다. 무엇을 고칠지는 결과를 읽는 사람이 정하며, 고치는 자리는 제안과 모델 편집 화면이다. 네 화면 모두 조회 권한으로 사용한다.

표 8.19. 점검 화면의 구분

화면	대상	묻는 것	결과의 형태
검증	적재된 표준 사전 전체	사전이 규칙을 지키는가	규칙별·심각도별 위반
진단	등록한 접속 대상의 카탈로그	운영 데이터베이스가 표준을 얼마나 지키는가	준수율·일관성 점수·조치 사항
대조	적재된 모델 정의와 접속 대상의 카탈로그	정의서와 데이터베이스가 같은가	한쪽에만 있음·서로 다름
작업	실행 기록	무엇을 언제 누가 실행하였는가	작업 목록과 결과 전문

8.6.1 검증

화면을 열면 `kdsm_meta`에 적재된 표준 사전 전체를 규칙 104종으로 한 번 검사한다. 요약 한 줄에 검사 건수와 심각도(오류·경고·정보)별 위반 건수가 표시되고, 그 아래에 규칙별 묶음이 심각도 순서로 놓인다. 위반 대상이 표준단어·표준용어·표준도메인·표준코드이면 해당 상세로 연결된다. 규칙과 심각도의 뜻은 라이브러리 장(7.4.2절)과 같다.

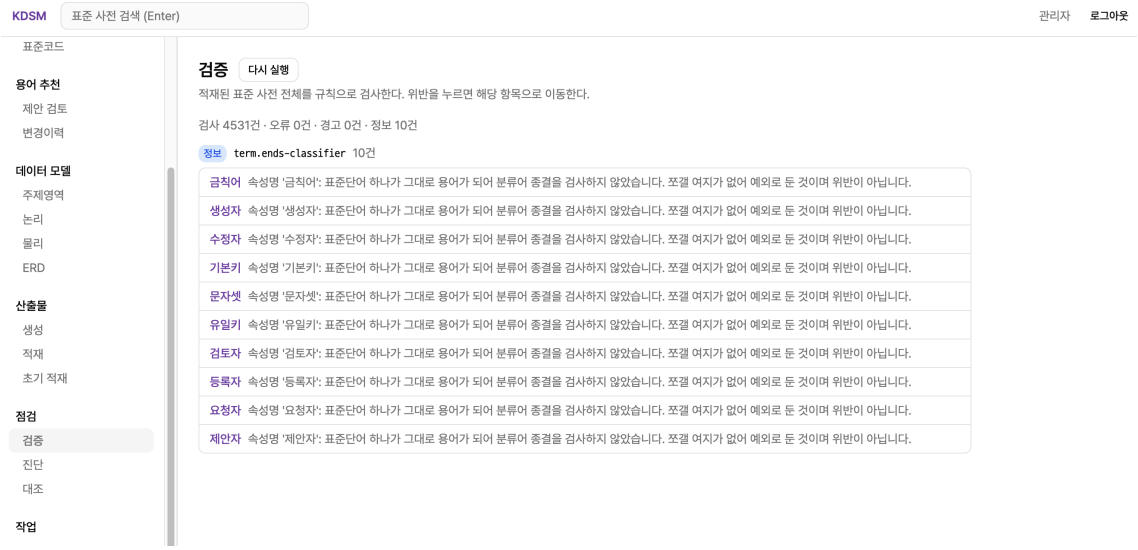

그림 8.16. 검증 화면. 요약 한 줄 아래에 규칙별 묶음이 배치된다.

제안 상세의 검증 결과(8.3.3절)는 그 제안이 새로 만드는 위반만 표시하므로, 사전 전체의 위반을 표시하는 이 화면과 건수가 다르다.

8.6.2 진단

등록한 접속 대상의 카탈로그를 읽어 표준화 수준을 측정한다. 대조에 쓰는 표준 사전은 툴킷에 적재된 것이다. 「사전 없이 진단」을 선택하면 스키마 내부의 일관성만 측정하며, 표준 사전을 갖추지 못한 데이터베이스의 첫 진단이 이에 해당한다. 측정의 기준은 라이브러리의 진단(7.6.1절)과 같다.

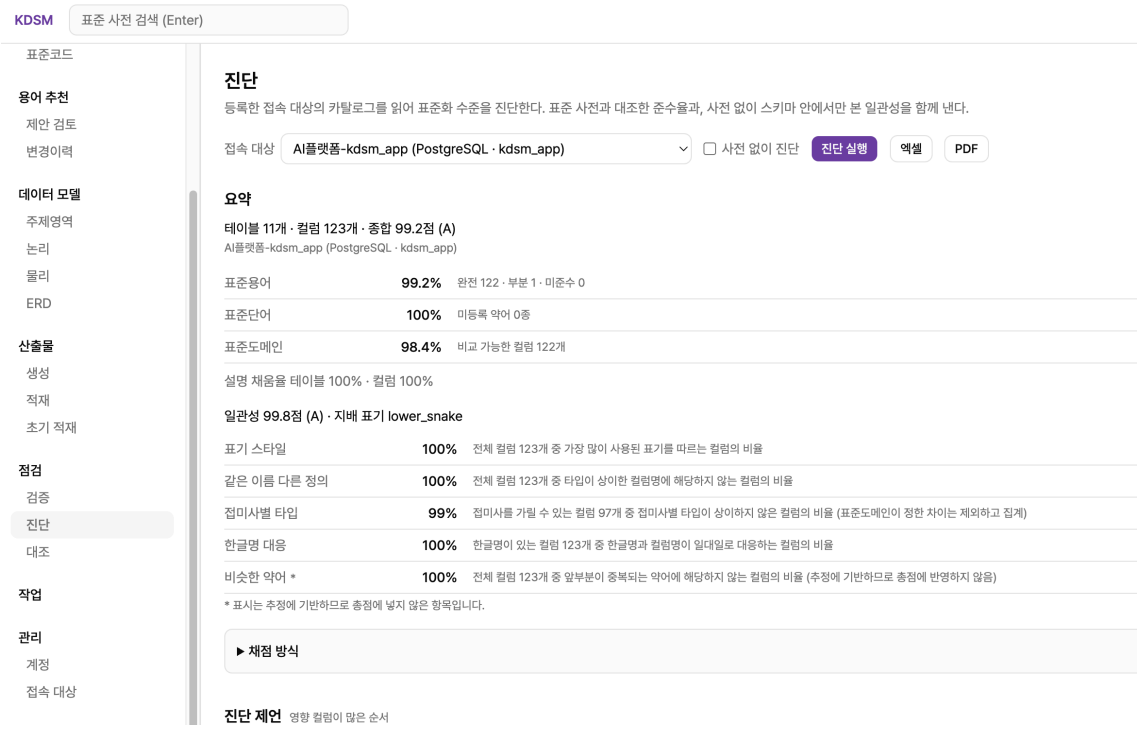

그림 8.17. 진단 화면. 요약 아래에 제언과 지적이 영역별로 배치된다.

표 8.20. 진단 결과의 구성

영역	표시하는 것
요약	표준용어·표준단어·표준도메인의 준수율, 설명 채움율, 일관성 점수와 항목별 점수
진단 제언	조치 사항과 판단 근거. 영향 컬럼이 많은 순서로 정렬한다
표준단어가 아닌 약어	사전에 등재되지 않은 약어와 사용 횟수
일관성 지적	구분별 건수와 지적 내용
조치가 필요한 컬럼	컬럼과 실제 타입과 문제
이름 통계	낱말 사용 현황과 접미사 순위

측정할 모집단이 없었던 항목은 0%가 아니라 「측정 불가」로 표시한다. 표준을 지키지 않은 것과 비교 기준이 없었던 것은 서로 다른 사실이기 때문이다. 「사전 없이 진단」의 점수는 종합 점수가 아니라 일관성 점수이므로 사전과 대조한 진단의 점수와 나란히 비교하지 않는다.

결과는 엑셀과 PDF로 내려받는다. 화면은 긴 목록을 상위 일부만 표시하고(조치가 필요한 컬럼 500건, 일관성 지적 50건, 약어 50종) 엑셀 리포트에는 전량을 싣는다. 설명이 누락된 컬럼 목록은 보완 작업의 대상이므로 화면에는 개수만, 엑셀에는 전량을 싣는다. PDF는 서버에 Quarto(Posit Software, PBC 2026년)가 설치되어 있어야 생성된다.

8.6.3 대조

적재된 모델 정의와 등록한 접속 대상의 카탈로그를 대조한다. 툴킷은 모델의 정본을 메타테이블로 정하므로 화면에서 작성한 테이블도 대조 대상이 된다. 접속 대상과 모델 스키마를 따로 지정하는 것은 운영 스키마의 이름이 모델에 기재된 이름과 같다는 보장이 없기 때문이다.

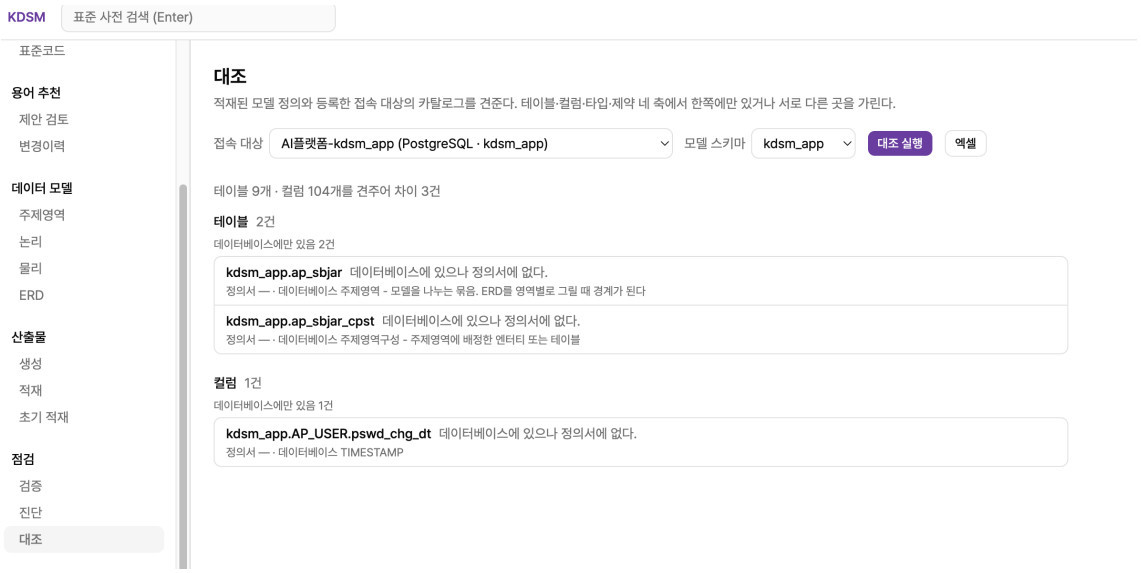

그림 8.18. 대조 화면. 대조 대상별 묶음 안에서 구분별로 다시 묶어 표시한다.

대조 대상은 테이블·컬럼·타입·제약의 네 가지이며, 불일치 한 줄에 위치·사유와 정의서·데이터베이스 각각의 값을 표시한다. 라이브러리의 `drift`와 마찬가지로 **어느 쪽이 옳은지는 표시하지 않는다**(7.5.1.2절). 카탈로그가 길이를 제공하지 않는 타입(예: Databricks의 `STRING`)은 「재지 못한 자리」에 따로 표시한다. 측정하지 못한 것을 일치한 것으로 표시하지 않기 위해서이다. 결과는 「요약」과 「차이」 두 시트의 엑셀로 내려받으며, 불일치가 없어도 생성된다.

8.6.4 작업

「작업」은 이 조직이 화면에서 실행한 작업의 기록을 표시한다. 종류·대상·상태·요청자·요청일시·소요시간·결과 요약을 목록으로 보이고, 상세에서 줄바꿈을 유지한 결과 전문을 표시한다. 커맨드라인으로 실행한 작업은 기록되지 않는다.

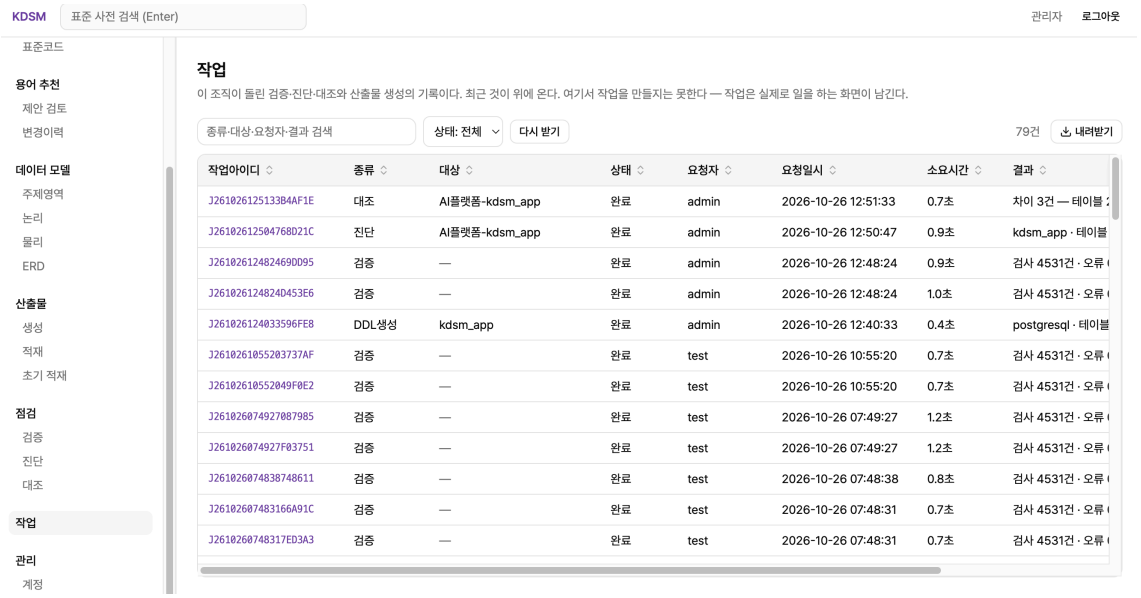

그림 8.19. 작업 목록. 최근 실행이 위에 배치되고 작업아이디가 상세로 연결된다.

점검의 결과 자체는 보관되지 않는다. 화면을 다시 열면 검증은 다시 실행되고 진단과 대조는 실행을 요청하여야 결과가 표시된다. 남는 것은 실행하였다는 기록과 요약 한 줄이다.

8.7 관리

8.7.1 계정

시스템관리 권한으로 이 조직의 계정을 등록하고, 상세에서 이름·이메일·사용여부를 고치고 권한을 교체하며 비밀번호를 초기화한다. **계정을 삭제하는 기능은 없다.** 사용하지 않게 하려면 사용여부를 중지로 변경한다. 권한 바꾸기는 더하는 조작이 아니라 보유할 권한 전체를 교체하는 조작이다. 비밀번호는 15자 이상이어야 한다.

조직에 관리자가 남지 않는 상태와 본인 확인 없는 비밀번호 변경을 막기 위해, 다음 네 가지는 시스템관리 권한으로도 수행할 수 없다.

표 8.21. 시스템관리 권한으로도 수행할 수 없는 조작

제약	조치
본인의 계정을 중지할 수 없다	다른 시스템관리자에게 요청한다
사용 중인 마지막 시스템관리자를 중지할 수 없다	다른 계정에 시스템관리 권한을 먼저 부여한다
본인의 시스템관리 권한을 회수할 수 없다	다른 시스템관리자에게 요청한다
계정 상세에서 본인의 비밀번호를 변경할 수 없다	내 계정에서 현재 비밀번호와 함께 변경한다

「내 계정」은 권한을 요구하지 않는다. 자신이 무엇을 할 수 있는지 확인하는 화면이므로, 권한을 요구하면 권한이 없는 사용자가 그 사실도 확인할 수 없기 때문이다.

8.7.2 접속 대상

ERD의 「데이터베이스」 원천과 진단·대조가 읽을 서버를 등록한다. 한 행이 서버·계정·스키마 하나를 가리키며, 지원 DBMS는 PostgreSQL·MySQL·Databricks이다. 여러 스키마를 대상으로 하려면 대상을 여럿 등록한다. 조회 시점에 스키마를 바꾸는 방법을 두지 않은 것은 등록으로 한정한 접근 범위를 조회로 확장하지 않기 위해서이다.

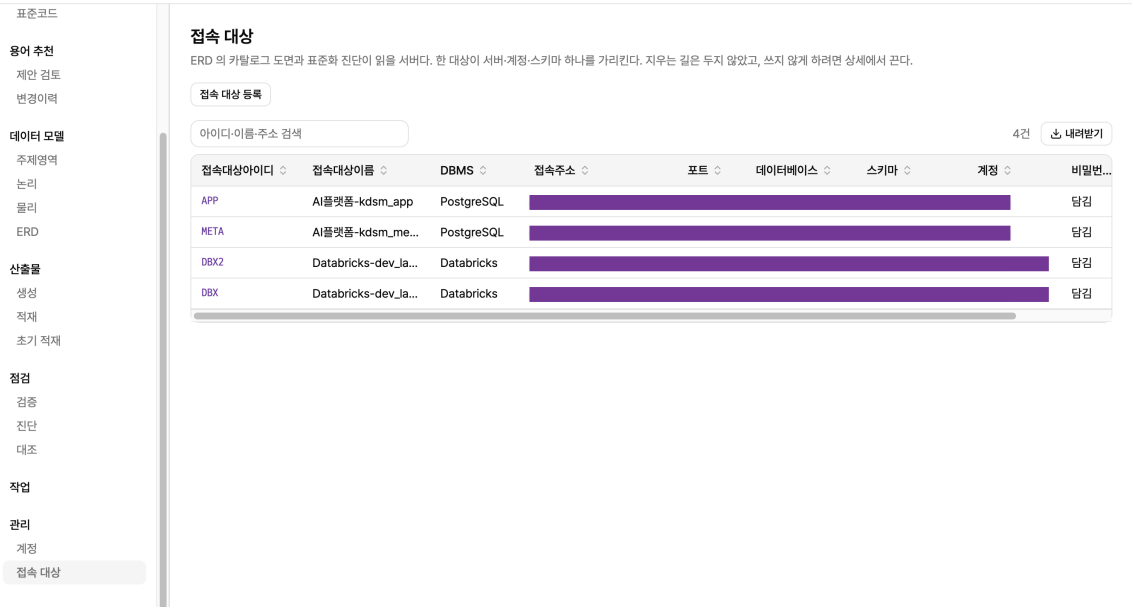

그림 8.20. 접속 대상 목록. 비밀번호 열은 적재 여부만 표시한다.

비밀번호는 `KDSM_ENCRYPTION_KEY`로 암호화하여 적재하며, 화면은 값이 아니라 「담김」·「없음」만 표시한다. 등록 시점에는 값의 형식만 확인하고, 실제 접속은 상세의 「접속 확인」으로 검사한다. 접속 실패는 「접속하지 못하였다」와 「카탈로그를 읽지 못하였다」로 구분되며, 사유에는 접속 주소와 계정을 싣지 않는다. 조회 권한 사용자에게 제공하는 선택 목록에도 접속 주소와 계정은 포함하지 않는다.

8.8 구축과 운영

8.8.1 구축의 일곱 단계

툴킷 구축은 라이브러리 장의 KDSM 시스템 환경 준비(7.3.4절)를 마쳐, `kdsm_meta` 스키마와 표준 사전이 적재된 PostgreSQL 데이터베이스가 있는 상태에서 시작한다. 툴킷은 그 사전을 읽는 화면이므로 사전이 먼저이다.

표 8.22. 툴킷 구축의 일곱 단계

단계	하는 일	실행할 것
1	도구를 설치한다	uv · Node.js 20 이상 · pnpm
2	저장소와 의존성을 준비한다	`uv sync` · `pnpm install`
3	애플리케이션 스키마(`kdsm_app`)를 만든다	`kdsm ddl`로 SQL을 생성하여 적용
4	애플리케이션 표준 사전과 모델을 적재한다	`kdsm validate` → `kdsm apply`
5	환경변수를 지정한다	`app/backend/.env`
6	관리자 계정과 권한을 만든다	`create_admin` · `grant_authority`
7	실행하고 기동을 확인한다	`uvicorn` · `pnpm dev`

툴킷 자신도 KDSM 방법론으로 설계되었다. 애플리케이션 테이블(`AP_` 접두어)과 권한·제안 코드의 정본은 `app/deliverable/`의 DM-DST 산출물 3종이다. 3·4단계는 그 산출물에서 DDL을 생성하고 검증하여 적재하는 것으로, 본 방법론이 프로젝트에 요구하는 절차와 같다.

```
cd library
uv run kdsm ddl ../app/deliverable/DM-DST-003_물리데이터모델-KDSM-APP_*.xlsx \
    -o ../dbms/ddl/kdsm_app.sql
uv run kdsm validate ../app/deliverable/DM-DST-00{1,2,3}_*.xlsx \
    --meta <메타파일> --name-prefix AP_
uv run kdsm apply ../app/deliverable/DM-DST-001_데이터표준정의-KDSM-APP_*.xlsx
```

> ! 참조되는 코드는 확정 상태여야 한다
>
> 서버는 기동 시점에 권한 코드와 제안 코드를 코드테이블에서 확정 상태인 것만 읽는다. 필요한 코드명이 없으면 즉시 중단하고 무엇이 없는지 출력한다. 오류 없이 기동하면 4단계가 완료된 것이다.

6단계가 커맨드라인 도구인 이유는 순환을 끊기 위해서이다. 로그인은 사용자가 있어야 동작하고 사용자를 만드는 화면은 로그인 뒤에 있다. 이 도구는 처음 들어가는 경로를 만들 때에만 사용하고, 이후의 계정 관리는 화면에서 수행한다.

8.8.2 환경변수와 비밀 취급

표 8.23. 백엔드 환경변수

변수	뜻	기본값
`KDSM_DB_HOST` · `PORT` · `NAME` · `USER` · `PASSWORD`	메타 데이터베이스 접속 정보	포트 `5432` 외 필수
`KDSM_DB_SCHEMA` · `KDSM_APP_SCHEMA`	표준 사전 스키마 · 애플리케이션 스키마	`kdsm_meta` · `kdsm_app`
`KDSM_JWT_SECRET`	토큰 서명 키. 32자 이상	필수
`KDSM_JWT_ACCESS_MINUTES` · `KDSM_JWT_REFRESH_DAYS`	액세스·리프레시 토큰 수명	`15`분 · `14`일
`KDSM_ENCRYPTION_KEY`	접속 대상 비밀번호의 암호화 키. 44자	선택
`KDSM_OUTPUT_DIR` · `KDSM_OUTPUT_KEEP_DAYS`	산출물 보관 절대경로 · 보존 기간	선택 · `7`일
`KDSM_ENTITY_PREFIXES`	엔터티 영문명에 허용하는 접두어	`MT_,AP_`
`KDSM_LLM_PROVIDER` · `KDSM_LLM_MODEL`	등재 초안의 LLM 제공자와 모델	선택

`.env`는 저장소에 추적하지 않으며 비밀번호와 토큰은 어느 명령도 인자로 받지 않는다(7.3.3절). 선택 변수가 비어 있으면 서버는 기동하고, 그 변수가 필요한 기능을 요청한 시점에 중단하여 변수의 이름을 안내한다. 암호화 키가 없으면 접속 대상만, 산출물 경로가 없으면 산출물 생성만 사용할 수 없다.

8.8.3 운영 시 확인할 것

표 8.24. 운영 중에 확인할 것

상황	확인할 것
정의서에서 코드명을 바꾸어 적재하였다	서버가 기동 시점에 그 이름을 찾지 못하여 중단한다. 코드값은 바꾸어도 되지만 코드명은 서버 코드의 기준점이다
새 접두어의 엔터티를 쓰려 한다	`KDSM_ENTITY_PREFIXES`에 더한다. 커맨드라인의 `--name-prefix`와 값이 어긋나면 한쪽에서 통과한 모델이 다른 쪽에서 위반으로 보고된다
`.env`나 라이브러리 소스를 수정하였다	서버를 재시작한다. 설정은 처음 사용할 때 한 번 읽고, `--reload`는 백엔드 디렉터리만 감시한다
등재 초안이 자격 증명 오류로 실패한다	AWS 자격 증명은 `.env`가 아니라 서버 프로세스의 환경이나 `~/.aws` 프로파일에 둔다
PDF 진단 리포트가 생성되지 않는다	uvicorn을 기동한 셸의 `PATH`에서 `quarto`가 실행되는지 확인한다
`KDSM_JWT_SECRET`을 교체하였다	모든 토큰이 무효가 되어 전 사용자가 다시 로그인한다. 유출이 의심되면 즉시 교체한다
`KDSM_ENCRYPTION_KEY`를 교체하였다	적재된 접속 비밀번호를 복호화할 수 없으므로 대상마다 다시 지정한다

8.8.4 라이브러리와 툴킷을 함께 쓰는 흐름

라이브러리 장의 작업 차례(7.7.1절)를 툴킷과 함께 수행하면 사람이 하는 일과 기계가 하는 일의 경계가 다음과 같이 나뉜다.

표 8.25. 툴킷과 라이브러리를 함께 쓰는 흐름

차례	하는 일	툴킷	라이브러리
1	표준 사전을 세운다	초기 적재	`load --seed`
2	필요한 표준용어를 찾고 없으면 제안한다	용어 추천 · 제안 · 판정	`suggest`
3	논리·물리 모델을 작성한다	주제영역 · 논리 · 물리 편집, 논리에서 생성	`validate`
4	ERD로 구조를 확인한다	ERD	`erd`
5	DDL을 생성하여 검토 후 적용한다	DDL 생성	`ddl`
6	산출물을 생성하여 배포한다	산출물 생성	`export`
7	운영 데이터베이스를 점검한다	진단 · 대조	`diagnose` · `drift`
8	정기 점검을 자동화한다	—	CI에서 `validate` · `diagnose --min-score`

화면은 여러 사람이 사전과 모델을 함께 다루는 일에, 커맨드라인은 반복과 자동화에 적합하다. 두 경로가 같은 라이브러리를 호출하므로 어느 쪽에서 확인한 결과든 다른 쪽에서 재현된다. 조직은 화면으로 합의하고 커맨드라인으로 그 합의가 유지되는지 점검한다.

V Experience Layer

9 경험 자산 203
9.1 개요 203
9.2 경험 자산의 체계 204
9.3 방법론 포털 205
9.4 데이터 표준화 저서 206
9.5 지식 베이스 207
9.6 경험 자산의 운영 209
9.6.3 용어 사전 211

CHAPTER 9

경험 자산

9.1 개요

KDSM 개요(제1장)는 Experience Layer를 **데이터 표준화에 대한 실무자의 노하우와 조직의 과거 데이터 관리 이력**이자 **데이터 표준화 활동의 성공과 시행착오에 대한 명시적 지식 체계**로 정의하였다. 그 구성 요소인 `Data Standardization Knowledgebase`는 Foundation·Guidance·Support Layer의 구성 요소를 운용하면서 체득한 경험 자산과 지식 자산의 총체이다(그림 1.15).

본 장은 이 정의를 실제로 제공되는 목적물로 구체화한다. KDSM은 경험 자산을 다음 세 가지로 정한다.

표 9.1. KDSM의 세 가지 경험 자산

경험 자산	형태	담는 것	제공 형식
방법론 포털	웹 사이트	방법론으로 들어가는 단일 진입점과 공통 리소스	HTML (방법론의 HTML·PDF·EPUB로 연결)
데이터 표준화 저서	웹 북	데이터 표준화의 일반 이론부터 KDSM까지의 체계적 서술	HTML · PDF · EPUB
지식 베이스	웹 사이트	방법론 요약, 실무 가이드, 참고자료, 사례, 블로그	HTML

앞의 장들이 **무엇을 어떻게 해야 하는가**를 규정하고 도구를 제공한다면, 경험 자산은 **그 규정과 도구를 사용한 결과를 어떻게 남기고 전달하는가**를 다룬다. 프로세스와 가이드는 개정되기 전까지 같은 내용을 유지하지만, 경험 자산은 프로젝트를 수행할 때마다 늘어난다. 늘어난 경험이 다시 가이드와 테크니컬 노트의 개정 근거가 되도록 연결하는 것이 Experience Layer의 역할이다.

> **노트**
>
> 제1장의 「Data Standardization Knowledgebase」는 웹 사이트 기반의 지식 베이스와 도서 형태의 기록물로 구성된다고 하였다. 본 장의 세 자산 가운데 지식 베이스가 앞의 것에, 데이터 표준화 저서가 뒤의 것에 해당하며, 방법론 포털은 두 자산과 방법론을 하나의 진입점으로 묶는다.

9.2 경험 자산의 체계

9.2.1 세 자산의 역할 분담

세 자산은 독자와 서술의 깊이가 다르다. 같은 주제가 세 곳에 모두 나타날 수 있으나, 각 자산이 맡는 질문이 다르므로 서로를 대체하지 않는다.

표 9.2. 세 경험 자산의 역할 분담

구분	방법론 포털	데이터 표준화 저서	지식 베이스
답하는 질문	어디에서 무엇을 찾는가	데이터 표준화란 무엇이며 KDSM은 어디에 서 있는가	실무에서 무엇이 통하였고 무엇이 실패하였는가
주 독자	방법론을 처음 접하는 사람, 자료를 찾는 실무자	데이터 표준화를 체계적으로 학습하려는 사람	프로젝트를 수행 중인 실무자
서술 단위	안내 페이지	부·장·절	페이지 · 블로그 글
갱신 주기	방법론의 판이 바뀔 때	판 단위의 개정	수시
프로젝트	`data-standardization-portal`	`data-standardization-book`	`data-standardization-website`

9.2.2 공통 제작 기반

세 자산은 모두 Quarto 프로젝트로 작성하며 원고는 마크다운(`.qmd`)이다. 한 벌의 원고에서 HTML·PDF·EPUB를 함께 생성하므로 형식마다 원고를 따로 관리하지 않는다. 판 번호는 각 프로젝트의 `_quarto.yml`에 있는 `version` 항목으로 관리한다.

표 9.3. 경험 자산 프로젝트의 제작 기반

프로젝트	Quarto 프로젝트 유형	출력 위치	2026-10-30 현재 판
`data-standardization-portal`	website (하위 book 1개 포함)	`docs/`	0.9.0
`data-standardization-book`	book	`docs/`	0.9.2
`data-standardization-website`	website	`docs/`	0.9.0

세 프로젝트 모두 빌드 결과를 `docs/`에 두고, 배포 스크립트(`deploy.sh`)가 그 결과를 게시 위치로 복사한다. 원고와 빌드 결과를 같은 저장소에 두므로, 저장소의 특정 판을 받으면 그 판의 사이트를 그대로 다시 게시할 수 있다.

9.3 방법론 포털

9.3.1 목적

방법론 포털은 KDSM의 **단일 진입점**이다. KDSM은 데이터 모델링 방법론을 독립된 책으로 제공한다. 책과 리소스를 따로 배포하면 사용자는 어떤 자료가 있는지, 어느 판이 최신인지를 스스로 확인하여야 한다. 포털은 책과 공통 리소스, 저서와 지식 베이스로 가는 경로를 한 화면에 모아 그 확인을 대신한다.

9.3.2 메뉴 구성

포털의 상단 메뉴는 다섯 항목이다.

표 9.4. 방법론 포털의 메뉴

메뉴	담는 것
홈	데이터 표준화의 필요성, 방법론 카드, 4계층 체계와 제공 형식의 요약
소개	방법론을 만든 배경, 방법론이 지키는 원칙, 운영 조직
방법론	데이터 모델링 방법론 사이트로 이동
리소스	용어 사전, 실무 체크리스트, 산출물 템플릿, PDF·EPUB 내려받기
지식자산	데이터 표준화 저서와 지식 베이스로 이동

「방법론」 메뉴의 항목은 방법론 사이트를 새 창으로 연다. 방법론 사이트는 포털과 다른 탐색 구조(장 단위의 사이드바)를 가지므로, 같은 창에서 이동하면 포털로 돌아오는 경로를 잃기 쉽기 때문이다.

9.3.3 방법론 사이트

방법론 사이트의 장 구성은 KDSM의 4계층 체계를 장의 차례로 그대로 옮긴 것이다.

표 9.5. 방법론 사이트의 장 구성

장	계층	데이터 모델링
1	KDSM 개요	`01_kdsm-dm`
2	Foundation	`02_dm-dsp`
3·4	Guidance	`03_dm-spm` · `04_dm-sdt`
5~8	Support	`05_dm-gud` ~ `08_dm-tok`
9	Experience	`09_dm-kwb`

장 파일명은 `<차례>_<방법론 약어>-<구성 요소 약어>` 형식이다. 방법론 약어는 `dm`(데이터 모델링)이고, 구성 요소 약어는 `dsp`(프로세스)·`spm`(표준 및 절차 매뉴얼)·`sdt`(산출물 템플릿)·`gud`(가이드)·`tne`(테크니컬 노트)·`lib`(라이브러리)·`tok`(툴킷)·`kwb`(지식 베이스)이다. 파일명만으로 그 장이 어느 방법론의 어느 구성 요소인지 확인할 수 있다.

방법론 사이트는 HTML로 읽고, 같은 원고에서 생성한 PDF와 EPUB을 사이트에서 내려받는다. 포털의 「리소스 → PDF / EPUB 다운로드」에도 같은 파일로 가는 경로가 있다.

9.3.4 프로젝트 구성

포털 프로젝트는 최상위 웹 사이트 하나와 하위 책 한 권으로 이루어진다.

```
data-standardization-portal/
├── _quarto.yml          # 포털(website) 설정과 상단 메뉴
├── index.qmd            # 홈
├── about.qmd            # 소개
├── resources.qmd        # 리소스
├── experience.qmd       # 지식자산
├── sub_modeling/        # 데이터 모델링 방법론 (book)
├── build.sh             # 전체 빌드
├── deploy.sh            # 게시 위치로 복사
└── docs/                # 빌드 결과 (포털과 책)
```

`build.sh`는 세 단계로 빌드한다. 먼저 책의 편집 원본 이미지(`assets/images/`)와 본문 참조본(`chapters/images/`)이 같은지 점검하고, 책을 렌더링한 뒤, 마지막에 포털 페이지를 렌더링한다. 책을 먼저 빌드하는 것은 포털의 메뉴가 책의 빌드 결과(`docs/sub_*/index.html`)를 가리키기 때문이다.

```
./build.sh     # 이미지 점검 → 책 렌더링 → 포털 렌더링
./deploy.sh    # docs/ 를 게시 위치로 복사
```

9.4 데이터 표준화 저서

9.4.1 목적

「데이터 표준화 방법론: 이론과 실무를 아우르는 종합 가이드」는 데이터 표준화의 개념에서 시작하여 일반적인 방법론과 실무 적용, 고급 주제를 거쳐 KDSM에 이르는 체계적 서술이다. 방법론 사이트가 **KDSM으로 무엇을 어떻게 수행하는가**를 규정한다면, 저서는 **그 규정이 데이터 표준화 전반에서 어떤 위치에 있는가**를 설명한다. 방법론을 적용하기 전에 배경을 이해하려는 사람과, 조직에 방법론을 설명하여야 하는 사람이 주 독자이다.

9.4.2 5부 구성

저서는 18장과 부록 3종으로 구성되며, 장은 다섯 부로 묶인다.

표 9.6. 데이터 표준화 저서의 구성

부	주제	장
1부	데이터 표준화 개요	1장 데이터 표준화의 필요성 · 2장 핵심 개념과 용어 · 3장 주요 표준화 프레임워크
2부	데이터 방법론	4장 데이터 모델링 표준 · 5장 메타데이터 관리 · 6장 데이터 품질 관리 · 7장 마스터 데이터 관리(MDM)
3부	실무 적용	8장 표준화 구현 전략 · 9장 데이터 거버넌스 체계 · 10장 표준화 도구와 기술 · 11장 실제 적용 사례
4부	고급 주제	12장 AI·ML과 데이터 표준화 · 13장 클라우드 환경의 표준화 · 14장 국제 표준 심화
5부	디플래닉스 방법론	15장 디플래닉스 방법론 개요 · 16장 KDSM 데이터 모델링 · 17장 KDSM 메타데이터 관리 · 18장 KDSM 데이터 품질 관리
부록	—	A 용어 사전 · B 표준화 템플릿 모음 · C 실무 체크리스트

1부부터 4부까지는 특정 방법론에 기대지 않는 일반론이고, 5부가 KDSM을 다룬다. 이 순서는 의도된 것이다. 일반론을 먼저 서술하여야 KDSM이 어떤 문제를 어떤 방식으로 다루는지를 다른 방법론과 견주어 판단할 수 있다.

9.4.3 방법론 사이트와의 관계

저서의 5부와 방법론 사이트는 같은 대상을 다르게 다룬다.

표 9.7. 저서 5부와 방법론 사이트의 관계

구분	저서 5부	방법론 사이트
범위	방법론을 한 부에서 개관한다	방법론을 한 권으로 다룬다
깊이	구조와 원리를 설명한다	절차·기준·산출물·도구를 규정한다
쓰임	이해와 설득	수행과 점검
개정	저서의 판 단위	방법론의 판 단위

두 자산 사이에 내용이 어긋나면 방법론 사이트가 정본이다. 저서는 방법론의 판이 바뀐 뒤 다음 판에서 5부를 맞춘다.

9.4.4 프로젝트 구성

```
data-standardization-book/
├── _quarto.yml          # 책 설정과 5부 구성
├── index.qmd            # 서문
├── chapters/            # 1~18장
├── appendix/            # 부록 A·B·C
├── references.qmd       # 참고문헌
├── references.bib
├── assets/              # 테마·조판 설정·폰트·이미지
├── deploy.sh            # 게시 위치로 복사
└── docs/                # 빌드 결과 (HTML·PDF·EPUB)
```

`quarto render`가 HTML과 함께 PDF·EPUB을 생성하며, 사이트의 내려받기 단추가 그 파일을 제공한다.

9.5 지식 베이스

9.5.1 목적

지식 베이스는 **실무에서 축적된 경험을 가장 짧은 경로로 공유하는 자리**이다. 방법론과 저서는 판 단위로 개정되므로, 프로젝트에서 얻은 경험이 반영되기까지 시간이 걸린다. 지식 베이스는 그 사이를 맡는다. 검증이 끝나지 않은 경험도 사례와 블로그 글로 먼저 공유하고, 반복하여 확인된 것은 방법론의 개정 대상으로 넘긴다.

9.5.2 다섯 메뉴

표 9.8. 지식 베이스의 다섯 메뉴

메뉴	하위 항목	담는 것
방법론	개요 · 데이터 모델링 · 메타데이터 관리 · 데이터 품질 · 마스터 데이터(MDM)	방법론별 핵심을 한 페이지로 요약한다
가이드	시작하기 · 구현 전략 · 거버넌스	표준화 프로젝트를 시작하고 추진하는 실무 안내
참고자료	용어 사전 · 체크리스트 · 템플릿 · 국제 표준	작업 중에 바로 찾아 쓰는 자료
사례 연구	—	업종별 적용 사례와 공통 성공 패턴
블로그	—	모범 사례, 문제 해결 경험, 시장 동향

앞의 세 메뉴는 정리된 지식을, 뒤의 두 메뉴는 개별 경험을 담는다. 사례 연구는 한 프로젝트의 전 과정을 정리한 것이고, 블로그는 한 가지 주제를 짧게 다룬다.

9.5.3 블로그

블로그는 세 종류의 글을 싣는다. 글의 종류에 따라 읽는 사람이 기대하는 것이 다르므로, 글마다 종류를 범주로 표시하고 종류별 작성 기준을 둔다.

표 9.9. 블로그 글의 세 종류

종류	범주	담는 것	반드시 포함할 것
모범 사례	`best-practice`	효과가 확인된 작업 방식	적용한 상황, 수행 방법, 효과를 확인한 근거
문제 해결	`troubleshooting`	실무에서 겪은 문제와 해결 과정	증상, 원인, 조치, 재발 방지 방법
시장 동향	`trend`	기술·도구·규제·표준의 변화	출처, 기준일, 표준화 실무에 미치는 영향

문제 해결 글은 증상이 아니라 원인으로 정리한다. 같은 원인에서 나온 문제는 증상이 달라 보여도 예방 방법이 하나이기 때문이다(7.7.3절). 시장 동향 글은 시점이 지나면 내용이 맞지 않게 되므로 기준일을 반드시 적고, 확인되지 않은 사실을 단정하지 않는다.

2026-10-30 현재 실린 글은 일곱 편이며, 위의 세 종류로 나누면 다음과 같다. 기존 글의 범주에는 아직 종류 범주가 적혀 있지 않으므로 종류 범주를 더하여 맞춘다.

표 9.10. 게시된 블로그 글

게시일	제목	종류
2026-01-15	데이터 표준화 시작하기: 5가지 핵심 원칙	모범 사례
2026-02-03	메타데이터 레지스트리 구축 실전 가이드	모범 사례
2026-03-10	데이터 품질 KPI 설계 가이드	모범 사례
2026-04-05	마스터 데이터 관리(MDM) 도입 사례	문제 해결
2026-05-20	데이터 카탈로그 도구 비교: 2025년판	시장 동향
2026-06-15	데이터 거버넌스 조직 설계 방법	모범 사례
2026-07-01	AI 시대의 데이터 표준화: 무엇이 달라지는가	시장 동향

글은 `pages/blog/YYYY-MM-DD-slug.qmd` 파일로 작성하며, 머리에 제목·설명·게시일·범주·저자를 적는다. 목록 페이지는 게시일의 역순으로 정렬하고 범주로 거를 수 있다.

```
---
title: "포스트 제목"
description: "포스트 설명"
date: YYYY-MM-DD
categories: [troubleshooting, data-modeling]
author: "저자명"
---
```

범주에는 글의 종류 하나와 주제 범주를 함께 적는다. 종류 범주는 위의 세 가지 가운데 하나이고, 주제 범주는 `data-modeling`·`metadata`·`data-quality`·`governance`·`tools`처럼 방법론과 영역을 나타낸다.

9.5.4 프로젝트 구성

```
data-standardization-website/
├── _quarto.yml          # 사이트 설정과 다섯 메뉴
├── index.qmd            # 랜딩 페이지
├── pages/
│   ├── methodology/     # 방법론 (5페이지)
│   ├── guide/           # 가이드 (3페이지)
│   ├── reference/       # 참고자료 (4페이지)
│   ├── case-studies.qmd # 사례 연구
│   ├── blog/            # 블로그 (목록과 글)
│   ├── about.qmd
│   └── contributing.qmd # 기여 방법
├── assets/              # 테마·스타일·스크립트·이미지
├── deploy.sh            # 게시 위치로 복사
└── docs/                # 빌드 결과
```

9.6 경험 자산의 운영

9.6.1 경험이 방법론으로 돌아가는 경로

경험 자산은 쌓아 두는 것으로 끝나지 않는다. 지식 베이스에 기록된 경험 가운데 반복하여 확인된 것은 방법론의 규정이 되고, 개정된 방법론은 저서에 반영된다.

표 9.11. 경험이 방법론으로 돌아가는 경로

단계	하는 일	자리
1	프로젝트에서 얻은 경험을 글로 남긴다	지식 베이스 블로그 · 사례 연구
2	같은 경험이 여러 프로젝트에서 반복되는지 확인한다	지식 베이스 범주별 목록
3	반복된 모범 사례는 가이드의 원칙으로, 반복된 문제는 테크니컬 노트의 주의 사항이나 검증 규칙으로 옮긴다	방법론 사이트 제5장·제6장, 라이브러리
4	개정된 방법론의 판을 올리고 포털에 게시한다	방법론 포털
5	다음 판에서 저서의 해당 부를 맞춘다	데이터 표준화 저서

3단계에서 문제 해결 경험이 검증 규칙이 되면 사람이 기억하지 않아도 같은 문제가 다시 검출된다. 라이브러리의 검증 규칙(7.4.2절)과 진단(7.6.1절)은 그렇게 옮겨진 경험의 누적이다.

9.6.2 판 관리

표 9.12. 경험 자산의 판 관리

자산	판을 올리는 시점	판의 기록 위치
방법론 포털	메뉴·리소스가 바뀌거나 하위 방법론의 판이 바뀔 때	포털 `_quarto.yml`의 `version`
방법론 사이트	장의 규정·산출물·도구가 바뀔 때	각 책 `_quarto.yml`의 `version`
데이터 표준화 저서	내용이 바뀔 때. 방법론 개정을 반영할 때 포함	저서 `_quarto.yml`의 `version`
지식 베이스	메뉴·페이지 구성이 바뀔 때. 블로그 글의 추가는 판을 올리지 않는다	사이트 `_quarto.yml`의 `version`

블로그 글은 게시일이 곧 식별자이므로 판을 따로 매기지 않는다. 게시한 글의 내용을 고칠 때에는 본문에 수정한 날짜와 수정 내용을 함께 적는다. 읽는 사람이 어느 시점의 내용인지 판단할 수 있어야 하기 때문이다.

9.6.3 점검 항목

게시하기 전에 다음을 확인한다.

- **[필수]** 포털의 방법론 메뉴와 지식자산 메뉴가 각각의 최신 빌드 결과로 연결되는가.
- **[필수]** 방법론 사이트와 저서의 PDF·EPUB이 HTML과 같은 판의 원고에서 생성되었는가.
- **[필수]** 저서 5부와 방법론 사이트의 내용이 어긋나는 곳은 없는가. 어긋나면 방법론 사이트를 정본으로 맞춘다.
- **[필수]** 시장 동향 글에 출처와 기준일이 적혀 있는가.
- **[권장]** 문제 해결 글 가운데 두 번 이상 반복된 것이 방법론의 개정 대상으로 등록되었는가.
- **[권장]** 각 사이트의 내부 링크와 다른 자산으로 가는 외부 링크가 끊어지지 않았는가.

용어 사전

이 부록은 본서에서 사용된 데이터 표준화 관련 주요 용어를 가나다순으로 정리한 종합 사전이다. 각 용어는 실무에서 자주 혼동되거나 정확한 이해가 필요한 것들 중심으로 선정되었으며, 관련 챕터를 병기하여 심화 학습이 가능하도록 하였다.

용어는 다음 원칙에 따라 정리되었다:

- **한국어 표기가 널리 사용되는 경우**: 한국어를 표제어로, 영문을 병기
- **영문 약어가 더 널리 사용되는 경우**: 영문 약어를 표제어로 배치 (별도 섹션)
- **동의어·유사어**: 대표어에 정의하고 별칭에 참조 표시
- **국제 표준의 공식 정의가 있는 경우**: 해당 표준을 인용

가

개념 도메인 (Conceptual Domain)
ISO/IEC (2023년d) 메타모델에서 값의 의미적 범위를 정의하는 요소. 예: '성인 나이 범위'는 개념 도메인이며, '18~150 정수'는 이를 실현한 값 도메인(Value Domain). (→ 제 5장)

개념 모델 (Conceptual Data Model)
데이터의 비즈니스 의미를 추상적으로 표현하는 최상위 모델. 엔티티와 관계를 정의하되 기술적 세부사항은 배제한다. (→ 제 3장)

검증 (Validation)
산출물이나 사전이 표준 규칙을 지키는지 검사하여 통과 또는 실패를 판정하는 작업. KDSM 라이브러리는 104종의 규칙으로 적재 이전에 적재 가능 여부를 판정하며, 툴킷은 적재된 사전 전체와 제안 하나가 새로 유발하는 위반을 각각 검사한다. 비율을 측정하는 진단과 구별된다. (→ 제 7장, 제 8장)

권한 (Authority)
툴킷에서 사용자가 수행할 수 있는 기능의 단위. 조회·제안·모델편집·표준승인·시스템관리의 다섯 가지이며, 역할이 아니라 항목 단위로 부여하고 한 사용자가 여럿을 가질 수 있다. 권한 사이에 상하 관계가 없으므로 시스템관리 권한이 있어도 조회 권한이 없으면 사전을 읽지 못한다. (→ 제 8장)

규칙 (Rule)
예외 없이 준수해야 하는 강제 조항으로 본서에서 **[필수]**로 표기. 위반 시 표준 심의를 통과할 수 없다. 치명적인 오류를 막는 장치라는 점에서, 시행착오를 줄이는 지침(Guideline)과 구별된다. (→ 제 1장, 제 5장)

금칙어 (Banned Word)
의미가 모호하거나 오용 우려가 있어 사용을 금지한 단어(예: 데이타, 주민번호). 사전에서 삭제하지 않고 대체 표준어와 함께 등재하여, 검증 도구가 "금지 — 대체어" 형태로 능동적으로 안내할 수 있게 한다. (→ 제 5장)

기술 노트 (Tech Notes)
실무 지침, 트러블슈팅, 세부 구현 방법을 담아 이론적 프레임워크와 실무 실행 사이의 간극을 메우는 '실행 가이드' 문서. KDSM Support Layer의 구성 요소. (→ 제 1장)

길이 예산 (Length Budget)
물리명 길이 상한(DBMS 식별자 제약)에서 접두어·접미어가 차지할 몫을 먼저 빼고, 남은 가용 길이 안에서 단어 수와 약어 길이를 설계하는 개념. (→ 제 5장)

나

날짜 형식 표준 (Date Format Standard)
날짜·시간 데이터의 통일된 표현 형식. 국제 표준 ISO 8601 (YYYY-MM-DD) 이 대표적이다. (→ 제 5장)

논리 데이터 모델
논리 데이터 모델(Logical Data Model)은 개념적 데이터 모델을 구체화하여 비즈니스의 데이터 구조와 규칙을 명확하게 표현한 설계도. (→ 제 3장, 제 4장)

논리명 (Logical Name)
논리 모델에서 사용하는 한글 명칭(엔터티명·속성명). 표준용어 사전의 한글명을 사용하며 물리명과 1:1로 대응한다. (→ 제 5장)

다

대조 (→ 드리프트)
정의서와 운영 데이터베이스의 구조를 견주어 차이를 찾는 작업. 라이브러리의 `drift` 명령과 툴킷의 「점검 → 대조」 화면이 수행한다. (→ 제 7장, 제 8장)

대표어 (Representative Word)
이음동의어 군(群)에서 표준으로 채택된 단어. 법령·감독규정 용어 → 공통표준 → 사용 빈도 → 의미 정확성의 우선순위로 선정한다. (→ 제 5장)

데이터 (Data)
사실·측정·관찰의 원시적 표현. DIKW 계층의 최하위 요소. (→ 제 1장)

데이터 도메인 (Data Domain)
(1) 특정 데이터 요소가 가질 수 있는 허용 가능한 값의 집합. (2) 조직 내 데이터의 주요 관심 영역(고객·제품·재무 등). (→ 제 5장)

데이터 딕셔너리 (Data Dictionary)
데이터베이스·정보 시스템에서 사용되는 데이터 요소들의 정의·형식·관계를 문서화한 참조 자료. (→ 제 3장, 제 4장)

데이터 라이프사이클 (Data Lifecycle)
데이터의 생성(Create) → 저장(Store) → 사용(Use) → 공유(Share) → 아카이빙(Archive) → 폐기(Destroy)의 전체 수명 주기. (→ 제 1장)

데이터 마트 (Data Mart)
특정 부서·주제에 특화된 소규모 데이터 웨어하우스.

데이터 모델 (Data Model)
데이터의 구조·관계·제약 조건을 형식적으로 표현한 추상적 모델. 개념·논리·물리의 3계층으로 구분한다. (→ 제 3장, 제 4장)

데이터 모델링 (Data Modeling)
비즈니스 요구사항을 데이터 구조로 변환하는 활동. Chen (1976년) 의 ER 모델이 이론적 기반. (→ 제 1장)

데이터 사전 (→ 데이터 딕셔너리)
데이터 딕셔너리(Data Dictionary)는 데이터베이스의 구조, 스키마, 사용자 권한, 메모리 구조 등 시스템 운영에 필요한 모든 메타데이터를 저장하는 시스템 테이블 및 뷰의 집합.

데이터 웨어하우스 (Data Warehouse, DW)
통합·주제 지향·비휘발성·시간 종속적 데이터 저장소. Inmon (2005년) 와 Kimball 와/과 Ross (2013년b) 이 서로 다른 접근법을 제시한다.

데이터 카탈로그 (Data Catalog)
조직 내 데이터 자산의 메타데이터를 중앙에서 관리·검색·거버넌스하는 시스템. Alation, Atlan, Collibra, DataHub 등이 대표적이다.

데이터 통합 (Data Integration)
여러 원천의 데이터를 통합된 뷰로 결합하는 활동. ETL, ELT, 페더레이션 등 여러 접근이 있다.

데이터 표준화 (Data Standardization)
조직 내·외부에서 데이터를 공통의 정의·구조·형식·값으로 일관되게 관리하기 위한 정책·프로세스·기술의 총체. 데이터의 '공통 언어'를 확립하는 것이다. (→ 제 1장, 제 5장)

데이터 표준화 정의 (Data Standardization Definition)
프로세스 라이프사이클에서 표준단어·표준용어·표준도메인·표준코드 정의의 4개 Phase를 묶어 부르는 상위 단계. 현황분석(요구사항분석) → **데이터 표준화 정의** → 데이터 표준화 확정 → 데이터 표준화 적용의 흐름 가운데에 위치한다. (→ 제 2장)

데이터베이스 정의서 (Database Definition)
물리 데이터 모델링 단계에서 물리 모델이 적재될 데이터베이스의 명세(DBMS 종류·버전, 스키마, 문자셋, 저장 특성 등)를 기록하는 산출물. (→ 제 3장, 제 4장)

도메인그룹 (Domain Group)
성격이 유사한 표준도메인을 묶은 최상위 범주. KDSM은 금액·날짜/시간·내용·명칭·번호·수량·율·코드의 여덟 그룹으로 관리한다. 표준도메인은 도메인그룹 → 도메인분류 → 도메인의 3계층 체계를 따른다. (→ 제 5장, 제 6장)

도메인분류 (Domain Classification)
도메인그룹을 한 단계 세분한 중간 계층으로, 각 분류는 반드시 하나의 상위 그룹에 소속된다(가격·요금 → 금액 그룹). 표준도메인명이 "도메인분류명 + 데이터 타입 약어 + 길이"(예: `가격N10`) 형태이므로 분류명은 곧 도메인명의 어간이 된다. (→ 제 5장, 제 6장)

동음이의어 (Homonym)
표기가 같고 의미가 다른 단어(예: 가구 家具/家口). "하나의 이름은 하나의 사실" 원칙을 위협하므로, 원칙적으로 업무에서 지배적으로 사용되는 하나의 의미만 표준으로 등록한다. (→ 제 5장)

드리프트 (Drift)
정의서와 운영 데이터베이스의 구조가 서로 어긋난 상태. 테이블·컬럼·타입·제약을 대조하여 「정의서에만 있음 · 데이터베이스에만 있음 · 서로 다름」으로 구분해 검출하며, 어느 쪽이 옳은지는 시점에 따라 달라지므로 도구가 판정하지 않는다. (→ 제 7장, 제 8장)

등재 초안 (Registration Draft)
표준단어로 나눌 수 없는 조각에 대하여 LLM이 작성한 표준단어 등재안. 단어명·영문명·영문약어·구분·설명·근거로 이루어지며 기존 규칙으로 다시 검사한 결과가 함께 제시된다. 상태는 언제나 후보이며 등재 여부는 사람이 제안과 판정으로 정한다. (→ 제 7장, 제 8장)

라

라이프사이클 (→ 데이터 라이프사이클)
데이터 라이프사이클(DLM)은 데이터가 생성되는 시점부터 유용성을 다해 안전하게 폐기될 때까지의 전 과정을 관리하는 정책 기반 접근 방식.

로우 데이터 (→ 원시 데이터)
로우 데이터(Raw Data)는 센서, 설문, 로그 등에서 수집된 후 어떠한 가공이나 정제도 거치지 않은 '미가공 원시 자료'를 뜻한다.

리버스 엔지니어링 (Reverse Engineering)
운영 데이터베이스의 카탈로그를 읽어 데이터 모델·ERD·정의서를 복원하는 작업. 문서가 기준이 되는 포워드 엔지니어링과 반대 방향이며, 운영이 시작되어 데이터베이스가 사실이 된 뒤의 유지보수 수단이다. (→ 제 6장, 제 7장)

마

명명 규칙 (Naming Convention)
데이터 요소의 이름을 부여하는 표준 규칙. ISO/IEC (2023년c) Part 5가 4원칙(유일성·필요성·정확성·간결성)과 3부분 구조를 규정한다. (→ 제 5장)

모범사례 (Best Practice)
본서의 SPM에서 원 프로세스 정의서에 명시되지 않았으나 일반적인 데이터 모델링 우수 관행으로 보강한 내용에 붙이는 표기(**[모범사례]**). 조직 표준으로 확정된 조항과 구별하여 읽어야 한다. (→ 제 3장)

묵시적 형변환 (Implicit Type Conversion)
비교·조인 양변의 데이터 타입이 달라 DBMS가 자동으로 변환하는 것. 대개 컬럼 쪽이 변환되어 인덱스가 무력화되므로, 도메인 표준 준수와 명시적 변환으로 원천 차단해야 한다. (→ 제 5장)

물리 데이터 모델
물리 데이터 모델(Physical Data Model)은 논리적 데이터 모델을 특정 데이터베이스 관리 시스템(DBMS) 환경에 맞게 변환하여, 실제 데이터를 컴퓨터 저장 장치에 물리적으로 저장할 수 있도록 구체화한 설계 도면. (→ 제 3장, 제 4장)

물리명 (Physical Name)
테이블·컬럼 등 DB 오브젝트의 영문 명칭. 표준단어의 영문약어를 언더스코어(`_`)로 연결하여 기계적으로 생성하며, 수작업 조정은 파싱 가역성을 깨뜨리므로 금지한다. (→ 제 5장)

미리보기 (Preview)
툴킷에서 적재나 생성을 실제로 반영하기 전에 반영될 내용을 확인하는 단계. 산출물 적재와 논리 모델에서의 생성은 미리보기를 거쳐야만 적용할 수 있으며, 미리보기를 거치지 않는 경로는 제공하지 않는다. (→ 제 8장)

바

바인드 변수 (Bind Variable)
SQL 조건값을 리터럴 상수 대신 변수로 전달하는 기법. 하드 파싱 반복과 라이브러리 캐시 오염을 막고 SQL 인젝션을 차단한다. (→ 제 5장)

변경관리 (Change Management)
표준·모델·정의서의 변경을 요청–검토–승인–반영의 절차와 이력으로 통제하는 활동. 비표준 항목은 표준화하거나 예외로 승인하며, 변경관리 대장으로 추적성을 확보한다. (→ 제 3장, 제 4장)

변경이력 (Change History)
승인된 제안이 표준 사전을 바꿀 때마다 남는 기록. 변경 전후의 내용과 그 변경을 일으킨 제안을 함께 담으며, 승인과 같은 트랜잭션에서 기록된다. 조회만 가능하고 수정·삭제할 수 없다. (→ 제 8장)

보조 식별자 (Alternate Key, UK)
주 식별자(PK) 외에 엔터티 안에서 값이 유일해야 하는 속성 또는 속성 조합. 이메일주소·주민등록번호·ISBN처럼 업무적으로 중복될 수 없는 값이 해당한다. 엔터티 정의서에 `UK1: 이메일주소` 형식으로 선언하고, 복수 속성이면 `+`로 연결하며, 보조 식별자가 둘 이상이면 쉼표로 구분한다. 속성 정의서의 '소속 유일키'가 `UK1-1` 형식으로 참여 속성과 순번을 지정하고, 물리 모델에서는 유일키(→ 유일키)를 거쳐 `UNIQUE` 제약으로 구현된다. (→ 제 4장, 제 6장)

부분 준수 (Partial Compliance)
컬럼 물리명이 표준용어로 등재되어 있지는 않으나, 구성 낱말이 모두 등재된 영문약어의 조합인 상태. 사전 등재만으로 해소되므로, 이름부터 다시 부여해야 하는 미준수와 조치 비용이 크게 다르다. 진단이 둘을 구분해 집계하는 이유가 여기에 있다. (→ 제 7장)

분류어 (→ 형식단어)
용어의 마지막에 위치하여 데이터의 형식적 성격을 결정하는 단어. 상세 정의는 '형식단어' 참조.

사

산출물 (Deliverable)
Task 수행의 결과로 생성되는 문서·데이터. 앞 Task의 산출물은 다음 Task의 입력물이 되어 프로세스를 사슬처럼 연결한다. (→ 제 2장, 제 3장, 제 4장)

산출물 템플릿 (Deliverable Templates)
프로세스 각 단계별로 생성해야 하는 결과물의 표준화된 양식. 작업 품질의 상향 평준화, 정보 누락 방지, 관계자 간 의사소통을 지원하며, 파일 템플릿과 DB 테이블 형태로 제공된다. KDSM Guidance Layer의 구성 요소. (→ 제 1장, 제 4장)

상향식·하향식 표준 도출 (Bottom-up / Top-down)
현행 데이터에서 표준 후보를 추출해 올라가는 방식(상향식)과 참조 표준·지침에서 내려받아 적용하는 방식(하향식). 표준화 방법 정의 시 두 방식의 조합 원칙을 확정한다. (→ 제 3장)

선택 의존성 (Extra)
파이썬 패키지가 기본 설치에 포함하지 않고 필요할 때만 함께 설치하도록 분리해 둔 의존성. KDSM 라이브러리는 데이터베이스 접속과 외부 서비스 연동을 선택 의존성으로 두어, 사용하지 않는 기능의 설치 부담을 없앤다. (→ 제 7장)

속성 정의서 (Attribute Definition)
논리 데이터 모델링 단계에서 각 엔터티에 속한 속성(애트리뷰트)을 표준용어·표준도메인으로 정의한 명세. 속성명·속성유형·PK/FK관련 정보·Null 허용·기본값 등을 포함한다. (→ 제 3장, 제 4장)

스키마 (Schema)
데이터의 구조·형식·제약 조건을 정의하는 명세. 관계형 DB, JSON, Avro 등 여러 표기법이 있다. (→ 제 5장)

스키마 레지스트리 (Schema Registry)
스트리밍 데이터의 스키마 버전 관리·호환성 검증을 제공하는 시스템. Kafka Schema Registry 등이 대표적이다.

심각도 (Severity)
검증 결과의 등급. ERROR는 반드시 고쳐야 할 위반, WARNING은 검토가 필요한 사항, INFO는 규칙을 실행하지 못해 판정하지 않았다는 기록이다. INFO를 별도 등급으로 두는 것은 검사하지 않은 것을 통과한 것으로 오인하지 않게 하기 위함이다. (→ 제 7장)

아

엔터티 정의서 (Entity Definition)
논리 데이터 모델링 단계에서 표준용어로 정의한 엔터티의 명세(엔터티명·식별자·엔터티 구분·관계 등)를 기록하는 산출물. (→ 제 3장, 제 4장)

엔티티 (Entity)
데이터 모델에서 관심 대상이 되는 실체. Chen (1976년) ER 모델의 기본 요소. (→ 제 4장, 제 5장)

엔티티 관계 모델 (Entity-Relationship Model, ER Model)
Chen (1976년) 이 제안한 데이터 모델링의 이론적 기반. 개체(Entity)와 관계(Relationship)로 도메인을 표현한다. (→ 제 3장)

예약어 (Reserved Word)
DBMS가 문법 요소로 예약한 단어(ORDER, USER, DATE 등). 물리명으로 사용할 수 없으며, 인용 부호로 회피하는 대신 명칭 자체를 바꾼다(ORDER → ORD). (→ 제 5장)

원시 데이터 (Raw Data)
가공되지 않은 원본 데이터. 원시 형태 그대로 저장하는 것이 데이터 레이크의 특징.

유일키 (Unique Key, UK)
논리 모델의 보조 식별자(→ 보조 식별자)를 물리 모델에서 이어받은 것으로, 기본키 외에 테이블에서 값이 유일해야 하는 컬럼 조합. 테이블 정의서에 `UK1: 이메일주소` 형식으로 선언하고, 컬럼 정의서의 '소속 유일키'가 `UK1-1` 형식으로 참여 컬럼과 순번을 지정한다. DDL에서는 `UNIQUE` 제약(및 유니크 인덱스)으로 구현되어 중복 입력을 데이터베이스가 직접 차단한다. 기본키가 행을 대표하는 하나의 열쇠라면, 유일키는 같은 문을 여는 또 다른 열쇠에 해당한다. (→ 제 4장, 제 6장)

이력 엔터티 (History Entity)
원본 변경 시 변경 전 상태를 시점별로 보존하는 엔터티. '~이력'으로 명명하며, 행위의 발생 기록인 '~내역'(트랜잭션 엔터티)과 구별한다. (→ 제 5장)

이음동의어 (Synonym)
표기가 다르고 의미가 같은 단어들(예: 해지·해약). 하나의 대표어만 표준으로 등록하고 나머지는 동의어로 매핑하여, 표준사전이 불필요하게 비대해지는 것을 막는다. (→ 제 5장)

입력물 (Input)
Task 수행에 투입되는 선행 산출물·참조 자료. 입력물–산출물의 연쇄가 프로세스의 실행 순서를 규정한다. (→ 제 2장, 제 3장)

자

자동화 라이브러리·툴킷 (Standardization Library / Toolkits)
데이터 표준화 과정의 주요 작업을 지원하는 소프트웨어. 산출물을 직접 생산하거나 돕는 함수 모음(Library)과 UI/UX를 제공하는 도구(Toolkits)로 구성된다. KDSM Support Layer의 구성 요소이며, 그 구현이 KDSM 라이브러리(`kdsm`)와 KDSM 툴킷(웹 애플리케이션)이다. (→ 제 1장, 제 7장, 제 8장)

작업 (Job)
툴킷에서 산출물 생성 등 실행 단위를 기록한 행. 종류·대상·상태·요청자·소요시간·결과 요약을 남기며, 점검 결과 자체는 보관하지 않고 실행하였다는 기록만 남긴다. 커맨드라인으로 실행한 작업은 기록되지 않는다. (→ 제 8장)

전수 검사 (Exhaustive Check)
표본이 아니라 대상 전체를 검사하는 방식. 검사 대상과 결과가 함께 기록되므로 무엇을 확인하였고 무엇을 확인하지 않았는지가 남는다. 표본만 보는 육안 대조는 통과 사실이 나머지를 보증하지 못한다는 점에서 구별된다. (→ 제 7장)

접속 대상 (Connection Target)
툴킷이 ERD·진단·대조를 위해 카탈로그를 읽을 데이터베이스의 등록 정보. 한 행이 서버·계정·스키마 하나를 가리키며 비밀번호는 암호화하여 적재한다. 여러 스키마를 대상으로 하려면 대상을 여럿 등록한다. (→ 제 8장)

정규화 (Normalization)
(1) DB 이론: 데이터 중복을 최소화하기 위한 관계 스키마 분해. Codd (1970년) 의 1~5NF. (2) 표준화 문맥: 데이터 값을 표준 형식으로 변환한다. (→ 제 3장)

제안 (Proposal)
표준 사전의 항목을 등록·수정·폐기하려는 요청. 툴킷에서는 사전을 직접 수정하지 않고 제안 권한이 있는 사용자가 제안을 등록하며, 표준승인 권한이 있는 사용자가 판정한다. 같은 대상의 같은 이름으로 제출 상태의 제안이 있으면 새 제안은 거부된다. (→ 제 8장)

종료 코드 (Exit Code)
명령이 수행 결과를 셸에 알리는 정수 값. KDSM 라이브러리는 0을 정상, 1을 처리 오류 또는 검증 실패, 2를 사용법 오류로 정한다. 측정을 목적으로 하는 명령은 기본값이 항상 0이며, 임계값을 지정한 경우에만 실패를 반환한다. (→ 제 7장)

주제영역 (Subject Area)
논리 데이터 모델을 업무 단위로 나눈 구획. 두꺼운 책을 장(章)으로 나누어야 읽히듯, 엔터티가 수백 개인 모델도 구획이 있어야 읽고 나누어 맡을 수 있다. 조직 구조가 아니라 데이터의 응집도를 기준으로 나누며(영역 안의 관계는 조밀하게, 영역을 넘는 관계는 성기게), 하나의 엔터티는 하나의 주제영역에만 배정한다. ERD 분할·담당 배정·변경 영향 범위의 단위가 된다. (→ 제 4장, 제 5장, 제 6장)

주제영역 배정 (Subject Area Assignment)
엔터티와 테이블을 주제영역에 소속시키는 것. 주제영역구성 정의서에 기재되며, 툴킷에서는 산출물 적재·논리 모델에서의 생성·배정 편집의 세 경로로 반영된다. 적재된 배정은 ERD를 주제영역별 도면으로 나누는 기준이 된다. (→ 제 8장)

주제영역 정의서 (Subject Area Definition)
논리 데이터 모델링 단계에서 주제영역의 명세(주제영역명·순번·설명)를 기록하는 산출물. 구획을 '선언'하는 문서로, 그 구획에 무엇이 들어가는지는 주제영역구성 정의서(→ 주제영역구성 정의서)가 기록한다. (→ 제 3장, 제 4장)

주제영역구성 정의서 (Subject Area Composition)
각 엔터티(또는 물리 모델의 테이블)를 어느 주제영역에 배정했는지를 기록하는 산출물. 소속 주제영역·구성 순서·구성대상(엔터티/테이블)·구성대상 이름·사용 부서로 구성된다. 주제영역 정의서가 구획을 선언한다면, 이 산출물은 그 구획을 채운다. 미배정 엔터티와 중복 배정이 없어야 구획이 완결된다. (→ 제 3장, 제 4장, 제 6장)

지식 베이스 (Knowledgebase)
데이터 표준화에 대한 실무자의 경험과 지식의 아카이브이자 조직의 핵심 지적 자산. 표준화 품질 향상과 생산성 제고의 선순환 체계를 이루며, KDSM Experience Layer를 구현한다. (→ 제 1장)

지침 (Guideline)
특별한 사유가 없는 한 따라야 하는 권고 조항으로 본서에서 **[권장]**으로 표기. 예외 적용 시 사유서와 승인이 필요하며, 동일 예외가 반복되면 표준 개정의 신호로 본다. (→ 제 1장, 제 5장)

진단 (Diagnosis)
이미 구축된 데이터베이스가 표준을 어느 정도 준수하는지 측정하여 비율과 등급으로 제시하는 작업. 통과 또는 실패를 판정하는 검증과 달리 결과가 비율이며, 「이미 구축한 것을 어디부터 개선할 것인가」에 답한다. (→ 제 7장, 제 8장)

차

차원 모델링 (Dimensional Modeling)
Kimball 와/과 Ross (2013년b) 이 제안한 데이터 웨어하우스 모델링 접근. Fact와 Dimension 테이블로 구성된다.

카

카탈로그 (→ 데이터 카탈로그)
조직 내 산재된 데이터 자산의 위치, 출처, 품질 및 구조를 파악할 수 있도록 돕는 중앙 집중식 인벤토리이다.

컬럼 정의서 (Column Definition)
물리 데이터 모델링 단계에서 속성을 변환한 컬럼의 물리 명세(물리명·데이터 타입·길이·Null·PK/FK 관련 정보 등)를 기록하는 산출물. 데이터 타입은 표준도메인의 물리 정의를 따른다. (→ 제 3장, 제 4장)

커버링 인덱스 (Covering Index)
조회에 필요한 컬럼을 모두 포함하여, 테이블 접근 없이 인덱스만으로 질의에 응답할 수 있는 인덱스. `SELECT *`나 불필요한 컬럼 조회는 커버링 인덱스 최적화를 봉쇄한다. (→ 제 5장)

코드 관리 대상
값의 유한성·재사용성·통제 필요성을 기준으로 코드로 관리하기로 판단한 속성. 한정된 값의 집합으로 반복 사용되는 데이터가 대상이 된다. (→ 제 3장)

코드 마스터·코드값 상세 (Code Master / Code Detail)
표준코드를 관리하는 두 계층. 코드 마스터는 코드 체계 자체의 속성(코드 ID·코드명·설명 등)을, 코드값 상세는 그에 속한 개별 코드값과 의미를 정의한다. 둘은 「코드 ID」로 연결되는 1:N 관계이며, 코드성 용어(분류어 '코드')는 반드시 하나의 코드 마스터에 연결된다. (→ 제 4장, 제 5장)

타

테이블 정의서 (Table Definition)
물리 데이터 모델링 단계에서 엔터티를 변환한 테이블의 물리 명세(물리명·한글명·기본키·테이블 유형 등)를 기록하는 산출물. (→ 제 3장, 제 4장)

툴킷 (KDSM Toolkit)
KDSM 라이브러리 위에 세운 웹 애플리케이션. 표준 사전의 조회, 용어 추천과 제안·판정, 데이터 모델 편집, 산출물 생성·적재, 검증·진단·대조를 화면으로 제공한다. 판단은 모두 라이브러리가 수행하므로 화면의 결과는 같은 조건의 커맨드라인과 같으며, 사용자·권한·제안·변경이력처럼 툴킷에만 있는 개념은 라이브러리에 두지 않는다. (→ 제 1장, 제 8장)

트라이 (Trie)
문자열을 글자 단위 경로로 저장하여 접두어 탐색을 빠르게 수행하는 자료 구조. 표준용어 추천은 확정 표준단어를 트라이로 색인한 뒤 한글 용어명의 가능한 분해를 모두 생성하여 순위를 매긴다. (→ 제 7장)

파

파싱 가역성 (Round-trip Parsability)
물리명에서 구성 표준단어를 역으로 복원할 수 있는 성질. 한글 단어와 영문약어의 양방향 1:1 대응, 그리고 약어의 기계적 결합 규칙으로 보장된다. (→ 제 5장)

판정 (Decision)
제출된 제안을 승인하거나 반려하는 행위. 표준승인 권한이 필요하며 반려에는 사유가 필수이다. 승인은 사전 반영·검토 이력·변경이력·제안 상태 변경을 한 트랜잭션으로 수행하므로 승인된 제안은 곧 사전에 반영된 제안이다. (→ 제 8장)

포워드 엔지니어링 (Forward Engineering)
확정한 정의서와 모델에서 ERD·DDL 등 하위 산출물을 생성하여 데이터베이스를 구축하는 방향의 작업. 문서가 기준이 되는 신규 구축 시점의 방향이며, 데이터베이스를 기준으로 삼는 리버스 엔지니어링과 대비된다. (→ 제 6장, 제 7장)

표준 및 절차 매뉴얼(SPM)
표준 및 절차 매뉴얼(Standards and Procedures Manual; SPM)은 조직 내 특정 업무나 프로젝트를 수행할 때 구성원 누구나 동일한 품질과 기준으로 일관되게 처리할 수 있도록 작업 표준과 절차를 기술한 가이드 문서. (→ 제 1장, 제 3장)

표준단어
데이터를 통합 관리하고 데이터베이스(DB)의 일관성을 유지하기 위해 사전에 정의한 핵심 용어 및 단위. (→ 제 2장, 제 3장, 제 4장, 제 5장)

표준도메인
데이터베이스(DB)를 구축하거나 데이터를 관리할 때, 데이터의 형식(Type), 길이(Length), 표현 규칙(Format) 등을 전사적으로 통일한 데이터 타입 및 값의 영역 표준. (→ 제 2장, 제 3장, 제 4장, 제 5장)

표준사전 (Standard Dictionary)
확정·공표된 표준단어·표준용어·표준도메인(및 표준코드)을 등록하여 조직 전체가 참조하는 저장소. 공표된 표준사전은 데이터 표준의 단일 참조원(SSOT)이 되며, 이후에는 변경관리 절차를 통해서만 수정한다. (→ 제 3장, 제 4장)

표준용어
표준단어의 조합으로 구성된 업무 용어(엔티티·속성명 등). 표준용어는 전사적인 데이터 일관성을 유지하기 위해 도메인, 단어, 용어를 표준화하는 작업의 결과물. (→ 제 2장, 제 3장, 제 4장, 제 5장)

표준 준수 점검 (Standards Compliance Check)
작성된 논리·물리 모델의 명칭·도메인·코드 사용이 공표된 표준을 준수하는지 점검하여 비표준 내역을 원인과 함께 도출하는 활동. 자동화 도구를 활용한 주기적 점검을 권장한다. (→ 제 3장, 제 4장)

표준코드
표준코드란 정보의 효율적인 처리와 의사소통을 위해, 특정 항목(명칭, 분류 등)에 대해 **규칙을 정해 통일한 공통 부호 체계**. (→ 제 2장, 제 3장, 제 4장, 제 5장)

표준화 (Standardization)
표준을 개발·적용·유지하는 활동. (→ 제 2장, 제 3장)

표준화 3대 원칙
표준화 작업 전반의 판단 기준이 되는 세 원칙. **유일성**(하나의 대상은 하나의 표준으로), **일관성**(동일 개념은 동일 표기로), **재사용성**(정의된 표준은 재사용 우선). (→ 제 3장)

표준화 계획서 (Standardization Plan)
요구사항분석 단계에서 표준화의 목적·대상·범위(In/Out-of-scope)·수행 방법·검토 절차를 확정하여 이후 모든 정의 활동의 기준선을 제시하는 산출물. (→ 제 3장, 제 4장)

표준화 대상 데이터 정의서 (Target Data Definition)
요구사항 정의 단계에서 표준화 대상이 되는 테이블화·컬럼화 요소를 목록화한 산출물. 이후 표준단어·표준용어·표준도메인·표준코드 정의의 원천이 된다. (→ 제 3장, 제 4장)

표준화 프로세스 (Standardization Process)
표준화 프로세스는 이해관계자 간의 합의를 바탕으로 품질, 안전, 호환성을 극대화하기 위한 가이드라인이나 기술 표준을 개발하는 과정. (→ 제 1장, 제 2장, 제 3장)

프로세스 3계층 구조 (3 Depth Breakdown)
데이터 표준화 프로세스를 Phase → Activity → Task로 분해하는 체계. 각 Task는 하나 이상의 산출물을 생성하며, 계층 표기(1 → 1.1 → 1.1.1)로 위치를 식별한다. (→ 제 2장, 제 3장)

하

하드 파싱 (Hard Parsing)
DBMS가 SQL 문장을 처음 해석·최적화하는 고비용 과정. 리터럴 상수가 박힌 SQL은 값마다 다른 문장으로 취급되어 하드 파싱을 반복시킨다. 바인드 변수로 방지한다. (→ 제 5장)

형식단어 (Class Word; 분류어)
용어의 마지막에 위치하여 데이터의 형식적 성격과 적용 도메인그룹을 결정하는 표준단어(명·번호·일자·금액·코드·여부 등). 용어는 [주제어]+[수식어]+[분류어]의 문법으로 구성되며, 이 구조는 ISO/IEC 11179-5의 표현어(representation term) 체계에 대응한다. (→ 제 5장)

후보화 (Candidating)
분석 대상 항목을 표준사전과 대조하여, 기존 표준과의 **매칭 후보** 또는 **신규 정의 후보**로 분류하는 절차. 단어·용어·도메인·코드 정의 Phase에 공통으로 반복되는 패턴이다. (→ 제 3장)

영문 약어 사전

4 Layered Architecture
KDSM의 방법론 구조. Foundation 〉 Guidance 〉 Support 〉 Experience의 4계층으로 구성되며, 하층부는 작업 절차·기준·요건을 정의하는 Spec 성향, 상층으로 갈수록 구체적 형상이 있는 문서·프로그램인 Utility 성향을 띤다. (→ 제 1장)

Activity
Phase를 구성하는 활동 단위. (→ 제 2장, 제 3장)

CI (Continuous Integration)
소스나 산출물을 변경할 때마다 정해진 검사를 자동으로 실행하는 개발 관행. 표준화에서는 산출물 수정마다 검증을, 정기적으로 진단을 수행하도록 등록하여 점검을 사람의 기억이 아니라 자동화에 맡긴다. (→ 제 6장, 제 7장)

CTE (Common Table Expression)
`WITH` 절로 서브쿼리에 이름을 부여하여 다단 질의를 단계화하는 SQL 구문. CTE 이름도 표준단어 약어로 짓는다. (→ 제 5장)

Deliverable
Task 수행으로 생성되는 산출물. (→ 제 2장, 제 3장)

Deliverable Flow Diagram
Task 간 입력물–산출물의 연쇄와 R&R을 한눈에 표현한 산출물 흐름도. Data Standardization Process의 구성 요소로, 프로세스 연결 구조의 정합성을 검증하는 도구가 된다. (→ 제 1장, 제 2장)

ER Model (→ 엔티티 관계 모델)
현실 세계의 데이터를 추상화하여 데이터베이스의 구조와 관계를 시각적으로 설계하기 위한 개념적 데이터 모델링 기법이다.

ERD (Entity-Relationship Diagram)
엔터티와 관계를 도식으로 표현한 데이터 모델 다이어그램. 논리 ERD와 물리 ERD로 구분되며 논리·물리 데이터 모델링의 핵심 산출물이다. 이론적 기반인 ER Model(→ 엔티티 관계 모델)이 표현 기법을 가리킨다면, ERD는 실제 설계 산출물로서의 도면을 가리킨다. (→ 제 3장, 제 4장)

ETL (Extract, Transform, Load)
원천에서 추출·변환하여 대상에 적재하는 전통적 데이터 통합 접근.

Experience Layer
데이터 표준화에 대한 실무자의 노하우와 조직의 과거 데이터 관리 이력을 담는 KDSM 최상위 계층. 성공과 시행착오에 대한 명시적 지식 체계이며, 지식 베이스로 구현된다. (→ 제 1장)

Foundation Layer
표준화 목표를 달성하기 위해 수행해야 하는 일련의 작업 절차와 흐름을 정의하는 KDSM의 기본 골격 계층. Data Standardization Process가 핵심 구성 요소다. (→ 제 1장)

Guidance Layer
표준화의 방향성 통제(Alignment), 품질·다양성의 균형(Trade-off Control), 의사결정 지원, 일관성 보장을 가이드하는 KDSM 계층. 표준 및 절차 매뉴얼(SPM)과 산출물 템플릿으로 구성된다. (→ 제 1장)

JWT (JSON Web Token)
서명된 토큰으로 인증 상태를 전달하는 방식. 툴킷은 수명이 짧은 액세스 토큰과 세션의 상한을 정하는 리프레시 토큰을 사용하며, 서명 키를 교체하면 모든 토큰이 무효가 되어 전 사용자가 다시 로그인한다. (→ 제 8장)

KDSM (KANDLE Data Standardization Methodology)
: 디플래닉스가 개발한, 국내 조직의 데이터 표준화 실무에 최적화된 데이터 표준화 방법론. 데이터 모델링 방법론을 다루며, 4계층 아키텍처로 구조화된다. (→ 제 1장)

kdsm (KDSM 라이브러리)
: 데이터 표준화와 데이터 모델링의 산출물을 코드로 다루는 파이썬 라이브러리. 파이썬 API와 커맨드라인의 두 계층으로 제공되며, `suggest` · `validate` · `ddl` · `load` · `apply` · `erd` · `drift` · `export` · `diagnose`의 아홉 명령을 갖는다. 업무 데이터가 아니라 메타데이터만 다룬다. (→ 제 7장)

LLM (Large Language Model)
: 대규모 텍스트로 학습한 언어 모델. KDSM에서는 사전에 없는 낱말의 표준단어 등재 초안을 작성하는 데에만 사용하며(→ 등재 초안), 검증·적재·DDL 생성에는 연결하지 않는다. 같은 입력이 같은 결과를 낳아야 하는 자동화의 전제를 지키기 위해서이다. (→ 제 7장, 제 8장)

MECE (Mutually Exclusive, Collectively Exhaustive)
: 상호 배타적이고 전체를 포괄하는 분류 원칙. 표준코드의 코드값 집합은 어떤 실제 값도 두 코드에 겹치지 않고(배타), 빠짐없이 어느 하나에 속하도록(포괄) 설계해야 한다. (→ 제 5장)

One Fact, One Name
: 하나의 사실에는 하나의 이름만, 하나의 이름은 하나의 사실만 가리켜야 한다는 데이터 표준화의 대원칙. 이 양방향 유일성이 무너지는 지점에서 커뮤니케이션 오류가 발생한다. (→ 제 5장)

Phase
: 프로세스의 최상위 단계. (→ 제 2장, 제 3장)

Process Lifecycle Diagram
: 데이터 표준화 프로세스의 전체 흐름(요구사항분석 → 데이터 표준화 정의 → 확정 → 적용)을 표현한 라이프사이클 도식. (→ 제 2장)

R&R (Roles and Responsibilities)
: Task별 수행·검토·승인·합의 주체의 구분. 산출물의 품질 책임 소재를 명확히 한다. (→ 제 3장)

Sargable (Search ARGument ABLE)
: 조건식이 인덱스를 사용할 수 있는 형태라는 뜻. WHERE 절에서 컬럼을 가공(함수·연산·형변환)하지 않고 상수 쪽을 가공하는 것이 원칙이다. (→ 제 5장)

Seek 페이징 (Keyset Pagination)
: 마지막으로 읽은 키를 기준으로 다음 페이지를 조회하는 방식. 뒤 페이지로 갈수록 앞 건을 모두 읽고 버리는 OFFSET 방식의 성능 한계를 극복한다. (→ 제 5장)

SSOT (Single Source of Truth)
: 조직 전체가 참조하는 단일 참조원. 공표된 표준사전·표준코드 저장소가 데이터 표준의 SSOT가 된다. (→ 제 3장)

Support Layer
: 규칙, 자동화 도구, 기술문서로 실무의 품질 보장과 작업 생산성을 극대화하는 KDSM 계층. Guides(작업 기준), Standardization Library/Toolkits(산출물 생산), Tech Notes(도구 운용)로 구성된다. (→ 제 1장)

Task
: Activity를 구성하는 최소 수행 단위. (→ 제 2장, 제 3장)

Three-Valued Logic (3VL)
: TRUE/FALSE/UNKNOWN의 3치로 평가되는 SQL 논리 체계. NULL과의 모든 비교는 UNKNOWN이며, `NOT IN` 서브쿼리에 NULL이 섞이면 결과가 공집합이 되는 오류의 근원이다(Winand 2023년). (→ 제 5장)

UK (Unique Key, → 유일키)
: 기본키 외에 테이블에서 값이 유일해야 하는 컬럼 조합. 논리 모델에서는 보조 식별자로, 물리 모델에서는 `UNIQUE` 제약으로 나타난다. 산출물에서는 선언에 `UKn:컬럼명`, 참여 컬럼 지정에 `UKn-n` 표기를 사용한다. 상세 정의는 '유일키' 참조.

UPSERT (Update or Insert)
: 기본키가 이미 있으면 갱신하고 없으면 삽입하는 적재 방식. 같은 파일을 여러 번 적재하여도 행이 겹쳐 쌓이지 않으므로 중간에 멈춘 작업을 다시 시작할 수 있다. 삭제는 수행하지 않으므로 시트에서 지운 행은 데이터베이스에 그대로 남는다. (→ 제 7장)

참고 사항

이 용어사전은 본서에서 사용된 용어의 참조 사전으로, 각 용어의 최신 상세 정의는 다음 출처를 확인하기 바란다:

- **국제 표준 원문**: ISO/IEC (2023년c), ISO (2022년), ISO/IEC (2008년), ISO/IEC (2015년b), ISO/IEC (2023년a), ISO/IEC (2024년) 등 ISO 표준 원문
- **DAMA 용어집**: DAMA International (2020). The DAMA Dictionary of Data Management (2nd ed.)
- **한국 표준**: e나라표준인증 (https://www.standard.go.kr) — 국내 KS 표준 검색
- **국내 정책 자료**: 국가기술표준원(KATS), 한국지능정보사회진흥원(NIA), 한국데이터산업진흥원(K-DATA)의 발간 자료

용어사전 활용 가이드

1. **본서 학습 시**: 각 용어의 정의 뒤에 병기된 관련 챕터로 이동하여 심화 학습이 가능하다.
2. **인용 자료 확인 시**: `@` 로 시작하는 인용 키는 참고문헌에서 상세 출처를 확인할 수 있다.
3. **조직 내 용어집 구축 시**: 본 사전을 조직 표준 용어의 기초 자료로 활용할 수 있으며, 조직 고유의 정의를 추가하여 확장 가능하다.

실무 체크리스트

데이터 모델링 프로세스 Task별 체크리스트

데이터 모델링 프로세스의 각 Task에 대한 완료 판정 기준을 한곳에 모은 통합 체크리스트다. 항목 번호는 Phase → Activity → Task의 3계층 표기를 따르며, 각 항목의 수행 방법과 표준 기준은 본서 「표준 및 절차 매뉴얼」 장의 해당 Task 절을 참조한다. [모범사례] 표시 항목은 원 프로세스 정의서에 없으나 일반적인 데이터 모델링 우수 관행으로 보강된 판정 기준이다.

Ⓟ 요구사항분석

1.1.1 표준화 원칙 이해

- ☐ 현행 표준화 지침을 확인하고 적용 우선순위를 파악하였는가?
- ☐ 참조할 외부·내부 표준을 식별하였는가?
- ☐ 표준화 작업에 필요한 도구·인력·일정을 확보하였는가?

1.1.2 표준화 범위·방법 정의

- ☐ 표준화 대상 시스템·데이터 영역이 명시되었는가?
- ☐ 제외 대상과 사유가 기술되었는가?
- ☐ 신규 정의/재사용 판단 기준이 정의되었는가?
- ☐ 검토·승인 절차가 합의되었는가?

1.2.1 데이터 구성요소 식별

- ☐ 대상 데이터의 원천 정보를 수집하였는가?
- ☐ 주요 개체와 관계가 개념 수준에서 정의되었는가?

1.2.2 표준화 대상 데이터 정의

- ☐ 테이블화 대상 개체가 도출되었는가?
- ☐ 컬럼화 대상 속성이 원자 단위로 도출되었는가?
- ☐ 각 요소의 한글명이 기록되었는가?

Ⓟ 표준단어 정의

2.1.1 표준단어 분석

- ☐ 대상 데이터 한글명을 최소 단위로 분해하였는가?
- ☐ 각 단어의 표준단어사전 존재 여부를 검색하였는가?
- ☐ 매칭/신규 후보로 분류하였는가?

2.2.1 표준단어 정의

- ☐ 표준 단어명이 확정되었는가?
- ☐ 이음동의어가 표준 단어명으로 수렴되도록 정의되었는가?
- ☐ 금칙어가 식별되었는가?
- ☐ [모범사례] 영문 약어가 정의되었는가?

Ⓟ 표준용어 정의

3.1.1 표준용어 분석

- ☐ 대상 용어 목록을 작성하였는가?
- ☐ 용어가 표준단어 조합으로 구성되는지 확인하였는가?
- ☐ 매칭/신규 후보로 분류하였는가?

3.2.1 표준용어 정의

- ☐ 표준 용어명이 단어 조합 규칙을 따르는가?
- ☐ 용어 설명이 업무 의미를 명확히 기술하는가?
- ☐ [모범사례] 영문명·대표 도메인이 연결되었는가?

Ⓟ 표준도메인 정의

4.1.1 표준도메인 분석

- ☐ 대상 도메인(형식) 목록을 작성하였는가?
- ☐ 각 도메인의 표준도메인 사전 존재 여부를 검색하였는가?
- ☐ 유사 형식을 공통 도메인으로 통합 검토하였는가?

4.2.1 표준도메인 정의

- ☐ 도메인 분류가 정의되었는가?
- ☐ 데이터 유형·길이·소수 자릿수가 정의되었는가?
- ☐ 형식·허용 값 범위가 정의되었는가?

Ⓟ 표준코드 정의

5.1.1 표준코드 분석

- ☐ 코드화 후보 용어를 선정하였는가?
- ☐ 코드 관리 대상 여부를 기준에 따라 판단하였는가?
- ☐ 기존 표준코드 존재 여부를 검색하였는가?

5.2.1 표준코드 정의

- ☐ 코드 속성 정보가 정의되었는가?
- ☐ 코드값 의미 목록이 도출되었는가?
- ☐ 각 의미에 코드값이 중복·누락 없이 매핑되었는가?

Ⓟ 표준화 확정

6.1.1 데이터 표준화 확정

- ☐ 후보 표준화 내역의 완전성·일관성을 검토하였는가?

- ☐ 이해관계자 합의가 이루어졌는가?
- ☐ 미비점 보완 후 재검토하였는가?
- ☐ [모범사례] 확정 이력을 기록하였는가?

6.2.1 데이터 표준화 공표

- ☐ 표준사전에 단어·용어·도메인을 등록하였는가?
- ☐ 표준코드를 저장소에 등록하였는가?
- ☐ [모범사례] 적용 시점·대상을 공지하였는가?

Ⓟ 표준화 적용

7.1.1 논리 데이터 모델링

- ☐ 주제영역명·엔터티명·속성명에 표준용어를 사용하였는가?
- ☐ 속성에 표준도메인을 적용하였는가?
- ☐ 식별자·관계가 정의되었는가?
- ☐ 모든 엔터티가 하나의 주제영역에 빠짐없이 배정되었는가?
- ☐ [모범사례] 정규화 원칙을 준수하였는가?

7.1.2 물리 데이터 모델링

- ☐ 대상 DBMS·스키마가 정의되었는가?
- ☐ 테이블명·컬럼명이 표준 약어 조합으로 생성되었는가?
- ☐ 컬럼 타입·길이가 표준도메인 물리 정의를 따르는가?
- ☐ [모범사례] 물리명이 명명규칙을 준수하는가?

7.2.1 표준 준수 여부 점검

- ☐ 명칭·도메인·코드 준수 여부를 점검하였는가?
- ☐ 비표준 내역을 원인과 함께 기록하였는가?

7.2.2 변경관리

- ☐ 변경 내역을 정의하고 영향 범위를 분석하였는가?
- ☐ 비표준 항목을 표준화 또는 예외 승인하였는가?
- ☐ 변경을 표준사전·코드·모델·정의서에 일관되게 반영하였는가?
- ☐ [모범사례] 변경관리 대장에 이력을 기록하였는가?

데이터 표준화 준비도 진단 체크리스트

번호	진단 항목	1(미흡)	2	3(보통)	4	5(우수)
1	데이터 표준화 정책 및 지침이 문서화되어 있다	☐	☐	☐	☐	☐
2	데이터 표준 관리 조직(거버넌스 책임자·표준 담당자·아키텍트)의 R&R이 지정되어 있다	☐	☐	☐	☐	☐
3	표준단어 사전이 구축되어 있고 이음동의어·금칙어가 관리된다	☐	☐	☐	☐	☐
4	표준단어에 영문명·영문약어가 1:1로 정의되어 있다	☐	☐	☐	☐	☐
5	표준용어 사전이 구축되어 있고 용어는 표준단어의 조합으로만 구성된다	☐	☐	☐	☐	☐

번호	진단 항목	1(미흡)	2	3(보통)	4	5(우수)
6	표준도메인이 도메인그룹 체계(금액·일자·여부 등)로 정의되어 있다	☐	☐	☐	☐	☐
7	표준코드가 정의되고 중앙 저장소에서 관리된다	☐	☐	☐	☐	☐
8	일자·금액·여부 등 값의 형식 표준이 전사 단일 방식으로 통일되어 있다	☐	☐	☐	☐	☐
9	신규 표준 제정 시 기존 표준의 재사용을 우선하는 절차가 있다	☐	☐	☐	☐	☐
10	엔터티·속성명에 표준용어, 컬럼 타입·길이에 표준도메인이 적용된다	☐	☐	☐	☐	☐
11	테이블·컬럼의 물리명이 표준단어 영문약어의 조합으로 생성된다	☐	☐	☐	☐	☐
12	표준의 확정·공표 절차(이해관계자 합의, 표준사전 등록)가 운영된다	☐	☐	☐	☐	☐
13	표준 준수 여부를 주기적으로 점검하고 비표준 내역을 관리한다	☐	☐	☐	☐	☐
14	표준 변경관리 절차와 변경관리 대장이 운영된다	☐	☐	☐	☐	☐
15	예외 승인 이력이 관리되고 반복되는 예외는 표준 개정으로 이어진다	☐	☐	☐	☐	☐

총점 계산: ____ / 75점

표 0.1. 준비도 점수 해석

점수 범위	성숙도 수준	권고 사항
60~75	최적화 단계	지속적 개선 및 혁신
45~59	관리 단계	프로세스 강화
30~44	정의 단계	표준화 체계 수립
16~29	초기 단계	기반 구축 필요
15	부재 단계	즉각적인 조치 필요

데이터 표준화 구현 체크리스트

데이터 모델링 프로세스의 라이프사이클(요구사항분석 → 데이터 표준화 정의 → 확정 → 적용 → 관리)을 기준으로 구현 진행 상황을 점검한다.

1단계: 요구사항분석

- ☐ 현행 표준화 지침·거버넌스 규정 숙지 및 리소스(도구·참조 표준·인력) 확보
- ☐ 표준화 범위(포함/제외)와 방법(신규 정의/재사용 판단 기준) 확정
- ☐ 표준화 대상 데이터의 원천 정보 수집 및 개념 구조 정의
- ☐ 테이블화·컬럼화 대상 요소 도출(표준화 대상 데이터 형상)

2단계: 데이터 표준화 정의

- ☐ 표준단어 분석·후보화(매칭/신규) 및 영문약어 정의
- ☐ 표준용어 분석·후보화 및 대표 도메인 연결
- ☐ 표준도메인 분석·후보화(타입·길이·형식·허용 값 범위)
- ☐ 표준코드 분석·후보화(코드 관리 대상 판단, 코드값 정의)

3단계: 데이터 표준화 확정

- ☐ 후보 표준화 내역의 완전성·일관성 검토
- ☐ 이해관계자 합의 및 미비점 보완·재검토
- ☐ 표준사전·표준코드 저장소 등록(공표)
- ☐ 적용 시점·대상 시스템 공지

4단계: 데이터 표준화 적용

- ☐ 표준용어·표준도메인을 적용한 논리 데이터 모델 작성
- ☐ 표준단어 영문약어 조합으로 물리명 생성 및 물리 데이터 모델 작성
- ☐ 명명규칙·코딩 가이드 준수 확인
- ☐ 정의서(주제영역·주제영역구성·엔터티·속성·데이터베이스·테이블·컬럼) 산출

5단계: 표준화 관리

- ☐ 표준 준수 여부 주기 점검 및 비표준 내역 기록
- ☐ 비표준 항목의 표준화 또는 예외 승인
- ☐ 변경의 영향 범위 분석과 표준사전·모델·정의서 일관 반영
- ☐ 변경관리 대장 이력 관리

참고문헌

Amazon Web Services. 2026년. Amazon Bedrock User Guide. Amazon Web Services. https://docs.aws.amazon.com/bedrock/latest/userguide/what-is-bedrock.html.

Celko, Joe. 2005년. Joe Celko's SQL Programming Style. Morgan Kaufmann.

Chen, Peter Pin-Shan. 1976년. "The Entity-Relationship Model – Toward a Unified View of Data". ACM Transactions on Database Systems 1 (1): 9–36. https://doi.org/10.1145/320434.320440.

Codd, Edgar F. 1970년. "A Relational Model of Data for Large Shared Data Banks". Communications of the ACM 13 (6): 377–87. https://doi.org/10.1145/362384.362685.

DAMA International. 2017년. DAMA-DMBOK: Data Management Body of Knowledge. 2nd ed. Technics Publications.

Holywell, Simon. 2016년. SQL Style Guide (한국어판). https://www.sqlstyle.guide/ko/.

Inmon, W. H. 2005년. Building the Data Warehouse. 4th ed. Wiley.

ISO. 2016년. Data quality – Part 61: Data quality management: Process reference model. ISO 8000-61:2016. International Organization for Standardization. https://www.iso.org/standard/63086.html.

ISO. 2022년. Data quality (Multiple parts). ISO 8000. International Organization for Standardization.

ISO/IEC. 2008년. Software engineering – SQuaRE – Data quality model. ISO/IEC 25012:2008. International Organization for Standardization.

ISO/IEC. 2015년a. Information technology – Metadata registries (MDR) – Part 5: Naming principles. ISO/IEC 11179-5:2015. International Organization for Standardization. https://www.iso.org/standard/60341.html.

ISO/IEC. 2015년b. Systems and software engineering – SQuaRE – Measurement of data quality. ISO/IEC 25024:2015. International Organization for Standardization. https://www.iso.org/standard/35749.html.

ISO/IEC. 2023년a. Information technology – Artificial intelligence – Management system. ISO/IEC 42001:2023. International Organization for Standardization.

ISO/IEC. 2023년b. Information technology – Database languages SQL – Part 2: Foundation (SQL/Foundation). ISO/IEC 9075-2:2023. International Organization for Standardization. https://www.iso.org/standard/76584.html.

ISO/IEC. 2023년c. Information technology – Metadata registries (MDR) – Part 1: Framework. ISO/IEC 11179-1:2023. International Organization for Standardization. https://www.iso.org/standard/78914.html.

ISO/IEC. 2023년d. Information technology – Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes. ISO/IEC 11179-3:2023. International Organization for Standardization. https://www.iso.org/standard/78916.html.

ISO/IEC. 2023년e. Information technology – Metadata registries (MDR) – Part 4: Rules and guidelines for the formulation of data definitions. ISO/IEC 11179-4:2023. International Organization for Standardization. https://www.iso.org/standard/78914.html.

ISO/IEC. 2024년. Annex SL: High-level structure for management system standards. https://www.iso.org/sites/directives/current/consolidated/index.xhtml.

Kimball, Ralph, 와/과 Margy Ross. 2013년b. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. Wiley.

Kimball, Ralph, 와/과 Margy Ross. 2013년a. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd ed. Wiley.

Posit Software, PBC. 2026년. Quarto: An Open-Source Scientific and Technical Publishing System. https://quarto.org.

Project Management Institute. 2021년. A Guide to the Project Management Body of Knowledge (PMBOK Guide). 7th ed. Project Management Institute.

Sveidqvist, Knut, 와/과 Mermaid contributors. 2026년. Mermaid: Diagramming and Charting Tool. https://mermaid.js.org.

Winand, Markus. 2012년. SQL Performance Explained: Everything Developers Need to Know about SQL Performance. Markus Winand. https://use-the-index-luke.com.

Winand, Markus. 2023년. Three-Valued Logic (3VL): Purpose, Benefits and Special Cases. Modern SQL. https://modern-sql.com/concept/three-valued-logic.

국립국어원. 2017년. 외래어 표기법. 문화체육관광부. https://korean.go.kr/kornorms/regltn/regltnView.do?regltn_code=0003.

대한민국 국회. 2013년. 공공데이터의 제공 및 이용 활성화에 관한 법률. https://www.law.go.kr/법령/공공데이터의제공및이용활성화에관한법률.

데이터가치연구소. 2024년. 동음이의어, 이음동의어, 한 글자 단어, 금칙어, 동의어/유사어 관리. https://www.s-data.co.kr/?p=1241.

유충현. 2026년a. 데이터 표준화 라이브러리 사용자 매뉴얼: 기준과 방법, 그리고 자동화를 통한 효율성, 일관성, 품질 향상 제시. 0.11.0. 디플래닉스.

유충현. 2026년b. 데이터 표준화 애플리케이션 사용자 매뉴얼: KDSM 웹 애플리케이션의 사용과 구축·운영. 0.1.0. 디플래닉스.

한국데이터산업진흥원. 2006년. 데이터 품질관리 지침 (Ver 2.1). 한국데이터산업진흥원(K-DATA). https://www.kdata.or.kr.

한국데이터산업진흥원. 2021년. DA 가이드: 데이터 표준화 – 데이터 표준 수립. DBGuide.net. http://www.dbguide.net/db.db?cmd=view&boardUid=12867&boardConfigUid=9&categoryUid=216.

한국정보통신기술협회. 2023년. 모든 데이터를 연결하는 데이터 표준화. 공공기술플랫폼협회 웹진. https://ppta.or.kr/webzine/2023_04/a1.html.

행정안전부. 2025년a. 공공기관의 데이터베이스 표준화 지침. 행정안전부고시 제2025-19호. 대한민국 행정안전부. https://law.go.kr/행정규칙/공공기관의데이터베이스표준화지침.

행정안전부. 2025년b. 공공데이터 공통표준용어·공통표준단어·공통표준도메인. 공공데이터포털. https://www.data.go.kr/data/15156442/fileData.do.

행정안전부. 2025년c. 데이터 상세 행정안전부_공공데이터 공통표준단어. https://www.data.go.kr/data/15156439/fileData.do#tab-layer-openapi.

행정안전부, 와/과 한국지능정보사회진흥원. 2021년. 공공데이터베이스 표준화 관리 매뉴얼. 행정안전부. https://www.cisp.or.kr/wp-content/uploads/2021/12/공공데이터베이스_표준화_관리_매뉴얼_202106.pdf.