

ROI를 고려한 데이터품질 확보방안

—
비투엔 컨설팅
조성덕 수석

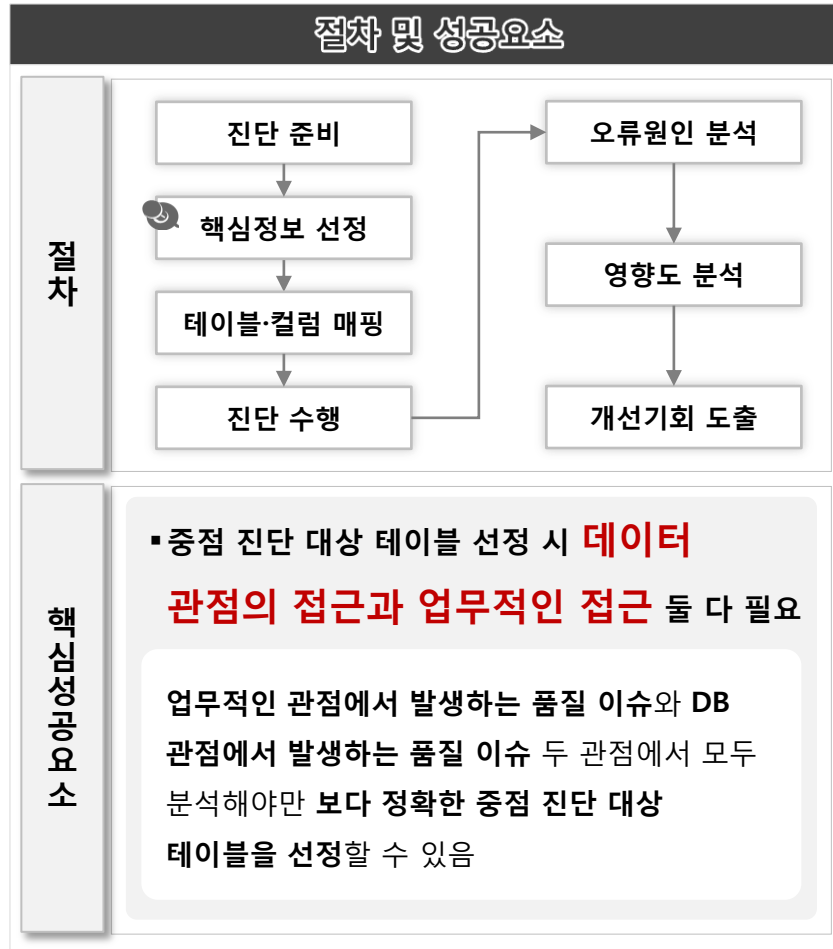
Contents

- I. 데이터 품질활동
- II. 데이터 값 관점의 개선방안
- III. ROI를 고려한 품질개선
- IV. 맺음말

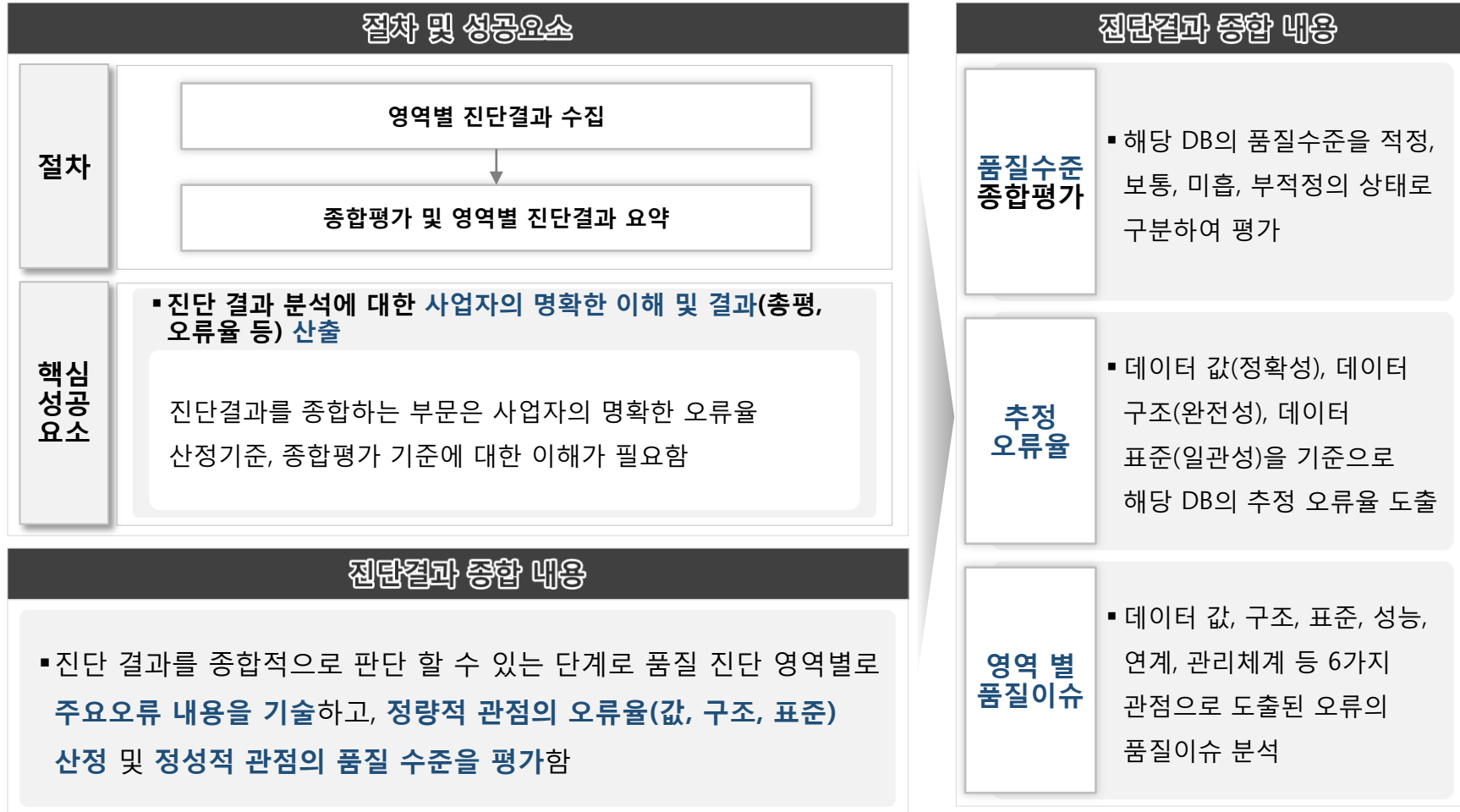
I. 데이터 품질활동

1. 품질진단
2. 품질진단 결과
3. 품질 개선

1. 품질진단



2. 품질진단 결과 (1/6) - 종합



2. 품질진단 결과(2/6) – 영역별 총평

□ 총평

구분	진단결과	비고
값	보통	- 농가정보 필수 값 누락, 인증일자 및 품목명칭 정합성 위배
구조	미흡	- 식별자 미정의, 정규화 위배, 미사용 테이블 존재
성능	미흡	- 농가정보 및 친환경 통계정보 등 주요화면에 대한 조회 성능 저하
연계	진단제외	
표준화	미흡	- 공통코드 중복, 동일한 용어가 서로 상이한 컬럼명 및 저장방식으로 정의되어 사용
관리체계	미흡	- 친환경 인증DB의 데이터 관리를 위한 조직·프로세스·기준 부재

※ [붙임 1] 총평 품질진단 기준

총평 품질진단 기준

구분	기준	점수
적정	해당 정보서비스를 제공하기 위한 중대한 데이터 품질의 문제점이 발견되지 않았으며, 관련 업무에 데이터를 활용하기에 충분한 상태	4
보통	해당 정보서비스를 제공하기 위한 중대한 데이터 품질의 문제점이 발견되었으나, 기관 내 자원활용을 통해 중대한 문제점이 개선되어 관련 업무에 데이터를 활용할 수 있는 상태	3
미흡	해당 정보서비스를 제공하기 위한 중대한 데이터 품질의 문제점이 발견되었으며, 사업 계획이 수립되어 있거나 계획된 자원의 정비가 선행되어야만 관련 업무에 데이터를 활용 할 수 있는 상태	2
부적정	현행 정보서비스를 제공하기 위한 중대한 데이터 품질의 문제점이 발견되었고, 사업 추진 전략이나 계획된 자원내에서 개선 할 수 없어 관련 업무에 데이터를 활용 하기 어려운 상태	1
진단 제외	진단 범위에 포함되지 않음	0

2. 품질진단 결과(3/6) – 영역별 추정 오류율

[추정 오류율]

○ 표준, 구조, 값 관점의 추정 오류율은 7.24% 이고, 구조 관점의 오류율이 전체 오류율의 큰 비중을 차지함.

< 표준·구조·값 오류율 진단 결과 >

구분	완전성(구조)	일관성(표준)	정확성(값)	전체
오류율	27.44%	19.00%	0.8%	7.24%
참고	- 전체 오류율 산정방법 $[완전성(구조) \times 0.1(가중치)] + [일관성(표준) \times 0.2(가중치)] + [정확성(값) \times 0.7(가중치)]$ ※ [붙임 2] 진단결과종합 - Profile_오류율_종합현황			

품질진단 지표별 분석 기준 정의			
분석영역	분석대상 단위	분석대상 상세	품질측정 지표
1. 데이터 표준 (Standard)	컬럼	1.1 컬럼 정의 불일치	일관성-속성
		1.2 행정표준용어 사용	일관성-표준
		1.3 행정표준코드 사용	일관성-표준
		1.4 표준 도메인 준수	일관성-표준
		1.5 표준 용어 준수	일관성-표준
2. 데이터 구조 (Structure)	테이블	2.1 데이터모델 vs 실제DB 불일치(테이블)	완전성-물리구조
	컬럼	2.2 데이터모델 vs 실제DB 불일치(컬럼)	완전성-물리구조
	테이블	2.3 미사용 테이블	완전성-물리구조
		2.4 테이블 중복	완전성-물리구조
		2.5 기본키 미정의 테이블	완전성-식별자
	컬럼	2.6 미사용 컬럼	완전성-물리구조
3. 데이터 값 (Value)	컬럼	3.1. 여부 도메인	정확성-범위 형식
	컬럼	3.2 수량 도메인	정확성-범위 형식
	컬럼	3.3 금액 도메인	정확성-범위 형식
	컬럼	3.4 율 도메인	정확성-범위 형식
	컬럼	3.5 날짜 도메인	정확성-범위 형식
	컬럼	3.6 번호 도메인	정확성-범위 형식
	컬럼	3.7 명칭 도메인	정확성-입력값
	컬럼	3.8 코드 도메인	정확성-참조관계
	컬럼	3.9 참조값	정확성-참조관계
4. 데이터 규칙 (Data Rule)	컬럼	4.1 번호 정확성	정확성-업무규칙
		4.2 시간순서 일관성	정확성-업무규칙
		4.3 컬럼간 논리 관계 일관성	정확성-업무규칙
		4.4 계산 및 집계 간 일관성	정확성-계산식
		4.5 필수값 완전성	정확성-입력값
		4.6 업무규칙(법령, 내규, 업무률)	정확성-업무규칙

2. 품질진단 결과(4/6) – 영역 별 품질이슈

□ 데이터 값 관점의 주요 발견사항

분류	주요 발견사항	주요 오류수준	품질지표
필수값분석	- 농가등록화면에서 농가/단체명, 농가주소 등은 필수항목으로 정의되어 있으나, 실 데이터가 NULL이나 공백 등으로 누락된 오류 발생	농가/단체명 4.41% 농가주소 4.44%	정확성-입력값
유효값분석	- 시료채취여부 등은 유효값(Y/N, 0/1)이 혼재되어 사용하거나 유효 값을 벗어난 데이터가 존재	시료채취여부 1.05% 행정처분요청여부 0.05%	정확성-범위·형식
패턴분석	- 분번, 표준인증번호 등은 정해진 패턴을 벗어나거나 한글 및 특수문자가 포함되는 오류데이터 존재	표준인증번호 0.56% 분번 0.02%	정확성-범위·형식
날짜분석	- 인증 시작일에 유효하지 않은 날짜 및 공백이 포함된 날짜데이터가 존재함.	인증시작일 0.03% 신청일자 0.02%	정확성-범위·형식
명칭분석	- 품목 명칭, 농가단체명 등에 비정상형 문자나 숫자 발견	품목명칭 0.02% 농가/단체명 0.003%	정확성-입력값
코드분석	- 필지주소코드, 품목코드 값이 기준정보에 미 정의된 코드 값을 사용함.	필지주소(법정동) 0.06% 품목코드 0.01%	정확성-참조관계
관계분석	- 인증기관인증정보승계의 인증기관 일련번호가 인증마스터에 존재하지 않음.	인증신청일련번호 0.02%	정확성-참조관계
업무규칙분석	- 인증번호의 경우 일정한 규칙에 따라 구성되어 되지만 규칙에 위배되는 데이터가 존재함. - 농가정보 및 조건에 따라 필지정보는 업무적으로 중복이 허용되지 않으나 중복 등록됨. - 인증 신청일자와 인증 교부일자 간의 날짜 선후관계를 위배한 데이터가 존재함. - 인증구분이 재포장 일 경우 업태의 값이 입력되어야 하나 누락된 경우가 발생	표준인증번호 유효성 0.78% 필지정보 중복 10.84% 신청일자, 교부일자간 시간순서 0.85% 인증구분 완전성 0.85%	정확성-업무규칙

- > 업무입력화면 및 업무 기준에 따라 반드시 들어와야 하는 필수 항목 분석
- > 유효범위에 벗어나는 데이터 분석
- > 데이터가 발생하는 패턴에 어긋나는 항목 분석
- > 날짜형식에 어긋나는 데이터 분석
- > 명칭의 형식에 위배되는 데이터 분석
- > 코드에 정의되지 않은 데이터 발생 분석
- > 컬럼간 논리적인 관계를 위배하는 데이터 분석
- > 업무적인 규칙을 위배하는 데이터 분석

2. 품질진단 결과 (5/6) – 영역 별 품질이슈

□ 데이터 구조 관점의 주요 발견사항

분류	주요 발견사항
중복	임시 작업 및 마감 시점의 데이터 보관을 위하여 전체 테이블 285개 중 9.12%인 26개의 테이블이 중복 사용됨.
미사용	업무 변경 및 기능 개선으로 인한 업무적으로 미 사용 중인 테이블이 64개 존재함.
산출물현황	데이터 모델(ERD)이 존재하나 통제 프로세스 부재로 테이블 기준 70.18%, 컬럼 기준 66.02%가 현행화 되지 않음.
식별자	식별자가 미 정의된 테이블이 전체 테이블의 약 51%임.
정규화	인증기관코드, 원산지국가코드를 개별코드 테이블로 관리하고 있음에도 인증기관명, 원산지국가코드를 인증마스터 테이블에서 중복 관리함으로써 정규화에 위배함.

- > 동일한 데이터를 관리하는 중복 테이블 분석
- > 업무적으로 사용하지 않는 미사용 테이블 분석
- > 산출물(ERD, 주제영역정의서, 테이블정의서 등)의 현행화 수준 분석
- > 관리 모델(ERD)의 현행화 수준 분석
- > 데이터 모델 정규화 분석

□ 데이터 표준 관점의 주요 발견사항

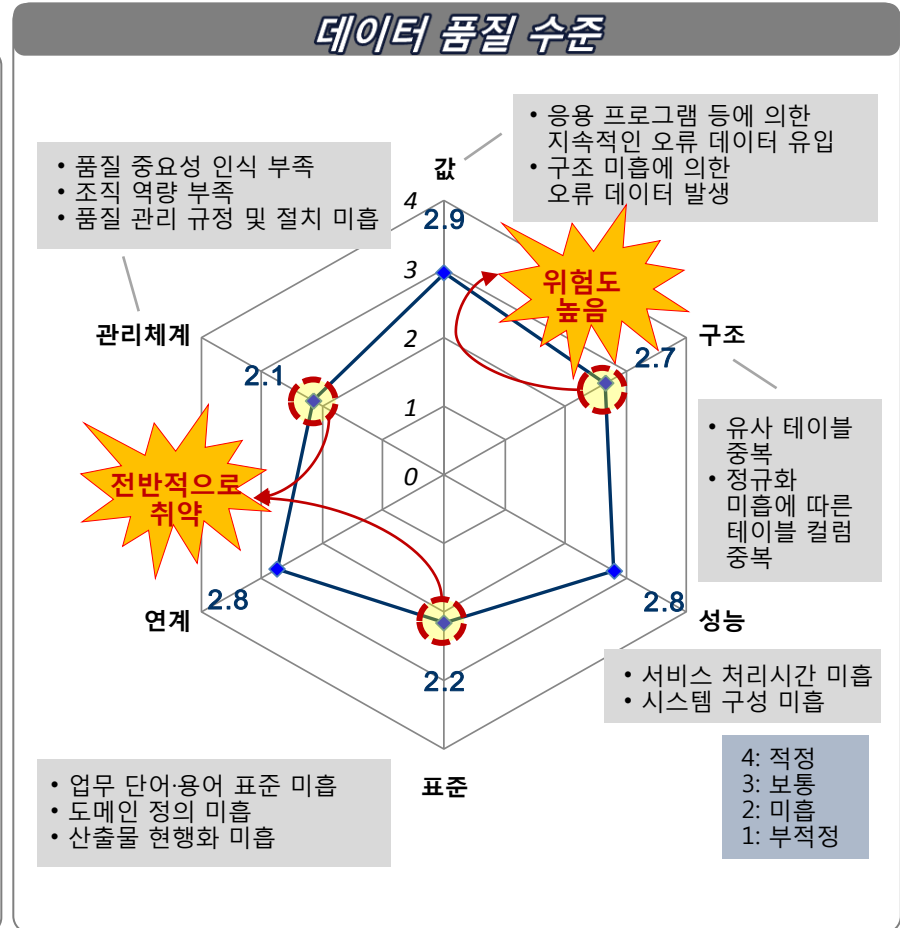
분류	주요 발견사항
지침	- 표준 용어 단어도메인 코드가 없으며, 표준화 변경통제 및 표준준수를 위한 표준관리체계와 지침이 부재함
용어	- 행정표준용어 사용현황이 3.86%이며, 행정표준용어를 사용하는 경우 영문 약어명을 미준수하는 경우가 존재함 - 데이터 표준에 대한 관리체계가 미흡하여 동일한 용어에 대하여 컬럼의 명칭이 불일치하거나 컬럼의 명칭은 동일하나 데이터 타입 및 길이가 다른 경우가 존재함
코드	- 기관 내부에서 정의한 공통코드들 중 동질적으로 유사한 코드가 두 개 이상 존재함.

- > 지침가이드의 제정 여부 등 현황 파악
- >
 - 데이터 표준 단어, 용어, 도메인 준수율 분석
 - 행정표준용어 준수율 분석
- >
 - 유사 코드 분석, 행정표준코드 준수율 등을 분석

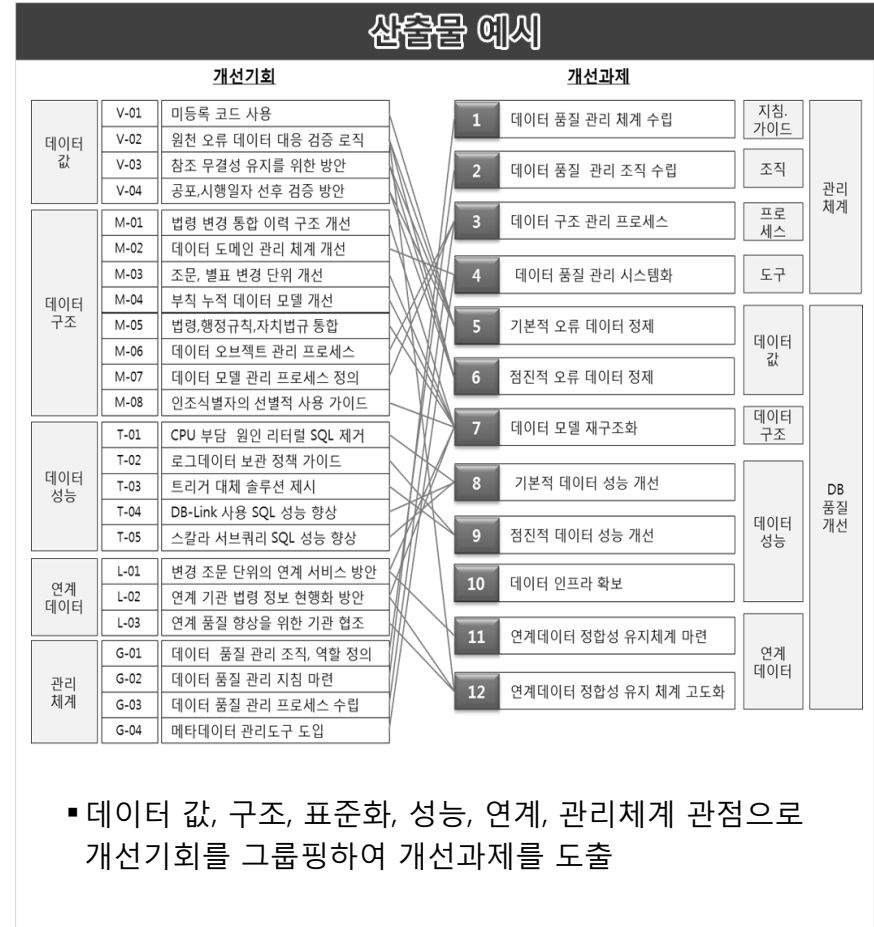
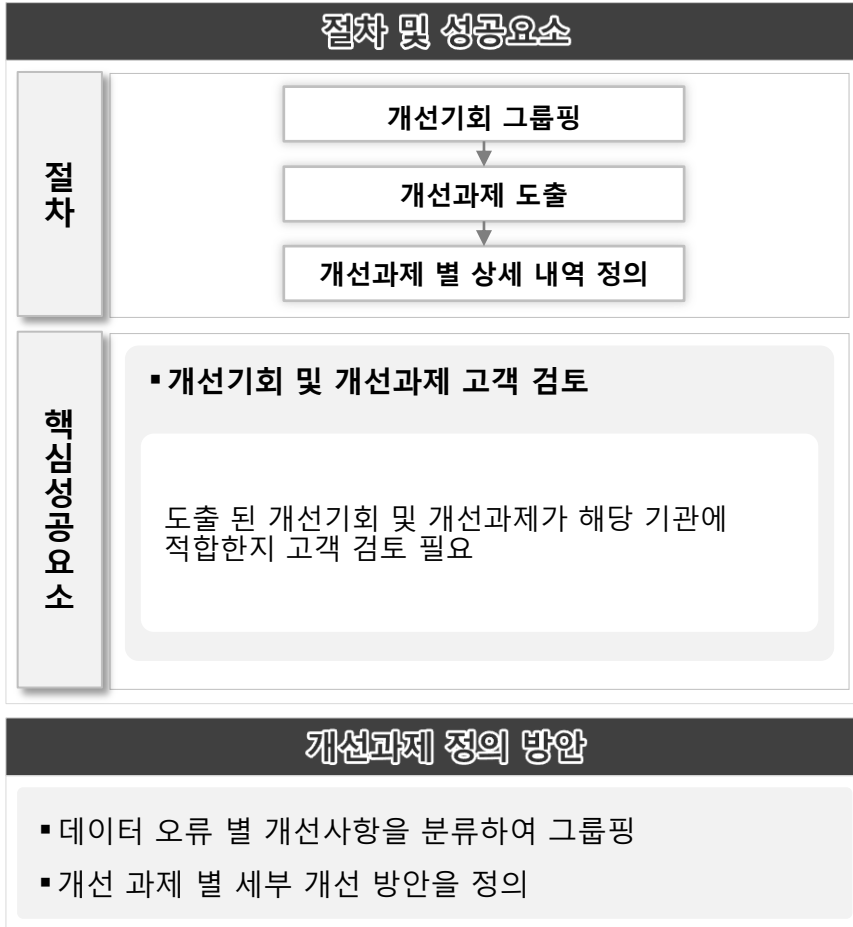
2. 품질진단 결과 (6/6) – 종합 품질진단 결과

데이터 품질 추정 오류율	
전체 오류율	
데이터 값 (정확성)	2.11% 잠재적 위험요인
데이터 구조 (완전성)	13.81%
데이터 표준 (일관성)	10.70%
전체	5.00%

DB 특성별 오류율		업무 분류별 오류율	
제공형 (6개)	6.66%	일반공공행정(9개)	4.32%
자체생성형 (11개)	4.20%	과학기술(1개)	9.32%
복합형 (14개)	3.40%	통일·외교(1개)	2.27%
수집형 (6개)	6.01%	공공질서 및 안전(3개)	1.99%
		농림해양수산(8개)	6.21%
		산업·중소기업(1개)	6.32%
		수송 및 교통(1개)	1.45%
		환경보호(3개)	5.58%
		지역개발(1개)	4.87%
		보건(1개)	5.58%
		문화·체육·관광(2개)	3.12%
		사회복지(4개)	6.15%



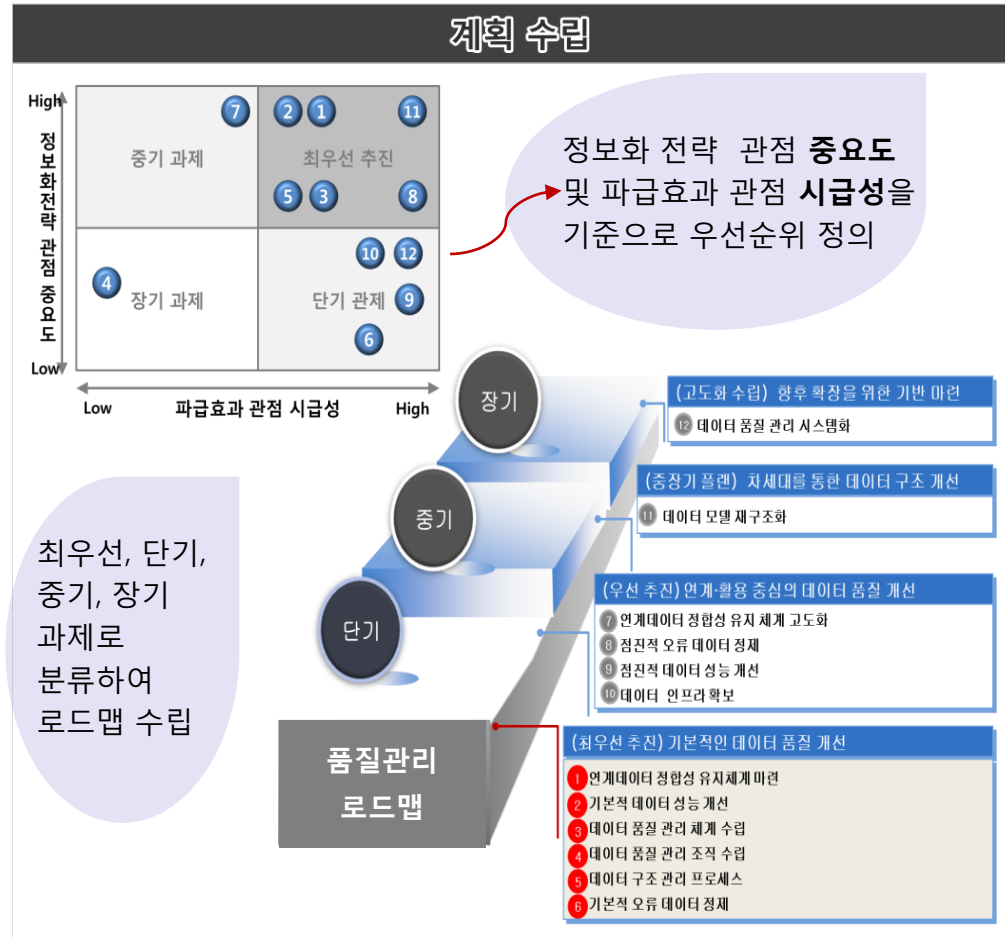
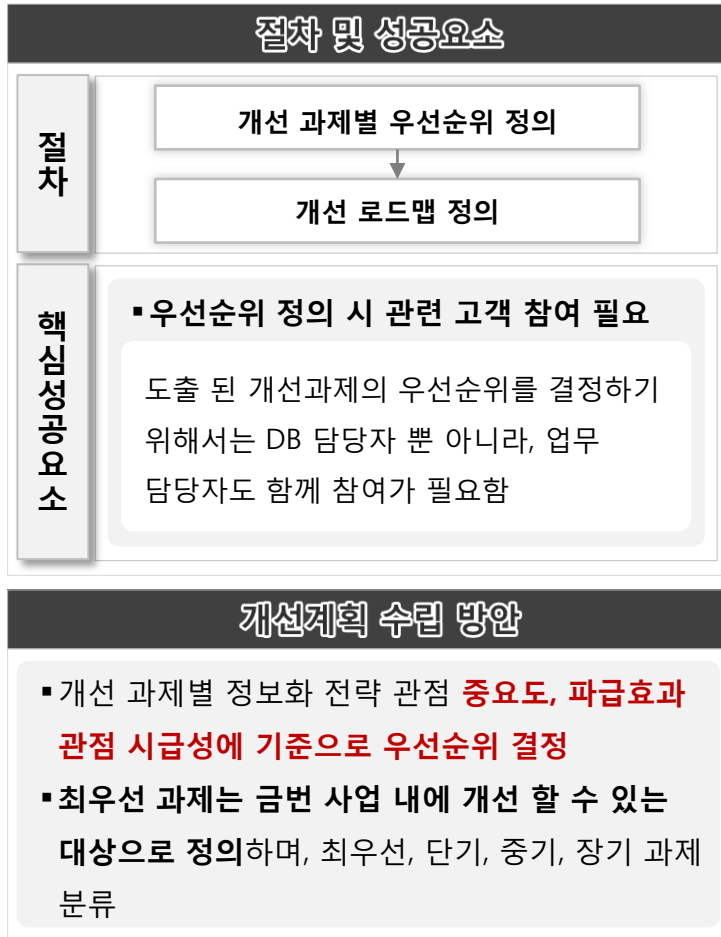
3. 품질개선 - 개선과제 도출



3. 품질 개선(2/2) – 개선과제 정의

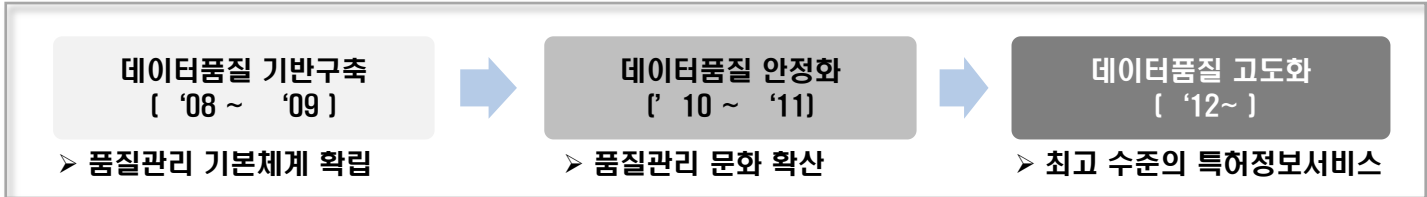
주요 개선과제	
DB품질개선	관리체계개선
오류데이터 정제 사용자 실수, 시스템 오류, 코드 불일치, 입력 서식 미비 등으로 인해 발생하는 데이터 값의 모순 해결	데이터 품질관리 지침 및 기준 수립 기관/DB 내 데이터의 품질관리를 위한 원칙 및 가이드 수립
응용 프로그램 개선 데이터 오류를 발생시키는 관련 기능, 화면 등을 수정하거나 필요 기능 추가 구현	데이터 품질관리 조직 및 역할 정립 데이터 품질관리를 위한 필요 기능 도출 및 R&R 정립
데이터 구조 재설계 데이터의 완전성, 일관성 등을 확보하고 업무요건을 충분히 반영할 수 있도록 데이터 모델 개선	데이터 품질관리 절차 수립 품질관리를 위해 도출된 필요 기능 수행을 위한 조직 별 절차 수립
데이터 표준화 단어, 용어, 도메인, 코드의 표준화 및 준수 뿐 아니라 행정표준의 적용	데이터 품질관리 도구 도입 및 운영 데이터 품질관리의 효과적 수행을 위한 자동화된 시스템 도입 및 운영
쿼리 튜닝 및 인덱스 재설계 성능에 부하를 발생시키는 SQL 튜닝	

3. 품질개선 - 개선계획 수립



3. 품질 개선 - 개선활동 (1/2)

2007년
 정보화 전략 계획(ISP)



2008년
 데이터품질관리 자동화 구축
 -예산: 15억, - 기간 6개월

- ▶ 데이터 관리체계 수립
- ▶ 데이터 모델 현행화
- ▶ 품질 진단, 품질 지표, 업무규칙 관리 기준
- ❖ 데이터 품질관리 포탈
- ❖ 메타데이터관리 시스템
- ❖ 데이터품질관리 시스템
- ❖ 영향도 분석 시스템

2009년
 데이터품질관리 고도화
 -예산: 10억, - 기간 7개월

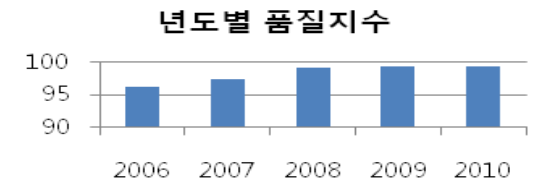
- ▶ 정보 오너쉽 체계 정립 및 관리방법
- DQM시스템 고도화(시스템 기능 확장)
- ❖ 품질평가관리시스템
- ❖ BR근거규정관리시스템
- ❖ 데이터이력관리 시스템

2010년
 데이터품질관리 제도 및 조직 확충

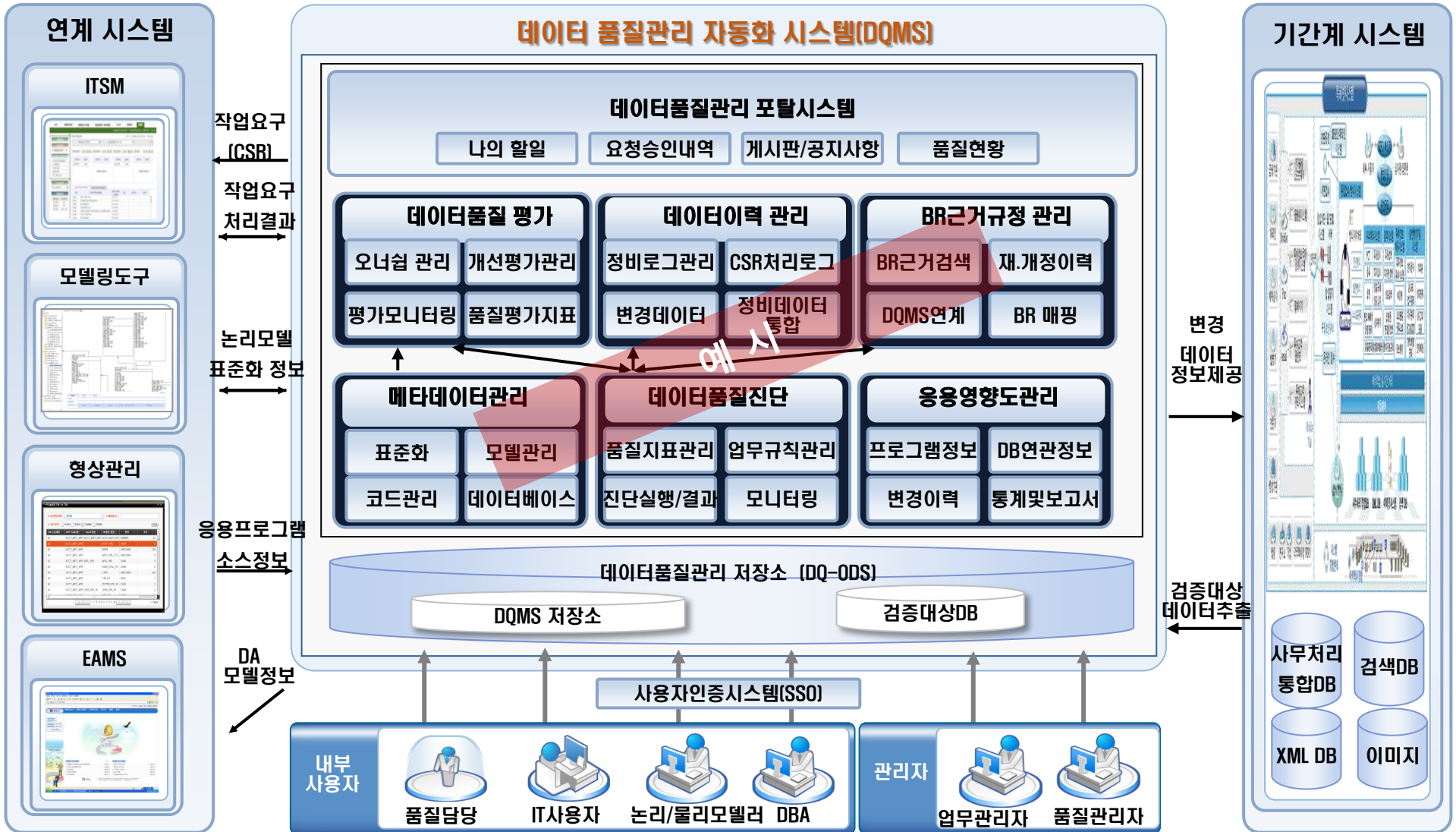
- ▶ 데이터품질관리 협의회 운영
- ▶ 데이터 오너쉽 부여(부서)
- ▶ 데이터관리센터 조직 확충 및 팀급 격상
 - DA 파트(4명),DQ 파트(4명),CSR 및 데이터 정비 파트(18명)
- ❖ 3세대 시스템 구축에 따른 DQMS 고도화

2011년~
 데이터품질 관리
 내재화 활동

- ▶ 정기적인 평가 및 매년 품질지수 관리
- ▶ 구축 및 운영시 BR 도출 의무화(제안요청서 적용)
- ▶ 현재 3세대 시스템 구축시 DA,DQ 체계 적용



3. 품질 개선 - 개선활동 (2/2)



II. 데이터 값 관점의 개선방안

1. 단기적 개선방안
2. 데이터 값 품질 vs 데이터 무결성
3. 데이터모델 vs 데이터 모델
4. 데이터 무결성
5. 데이터 베이스 제약

1. 단기적 개선방안(1/3)



1. 단기적 개선방안(2/3)



데이터 오류가 발생하지 않으면,
진단/정제/개선이 필요 없지 않을까?



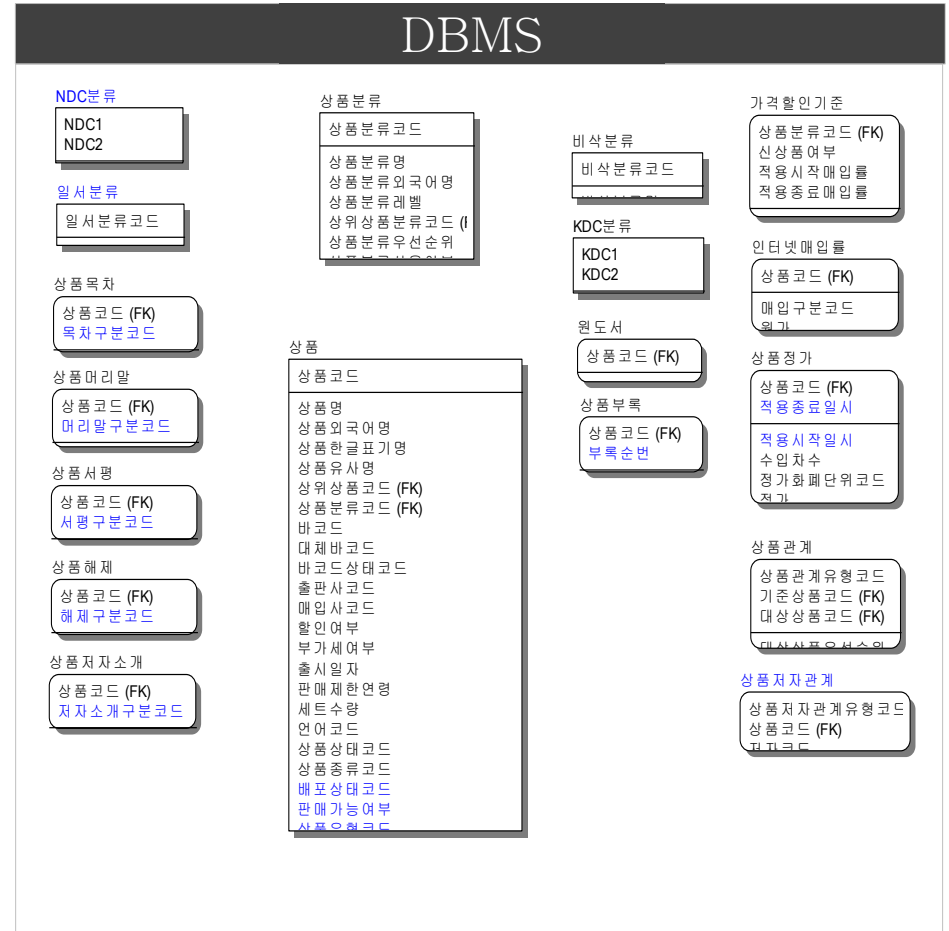
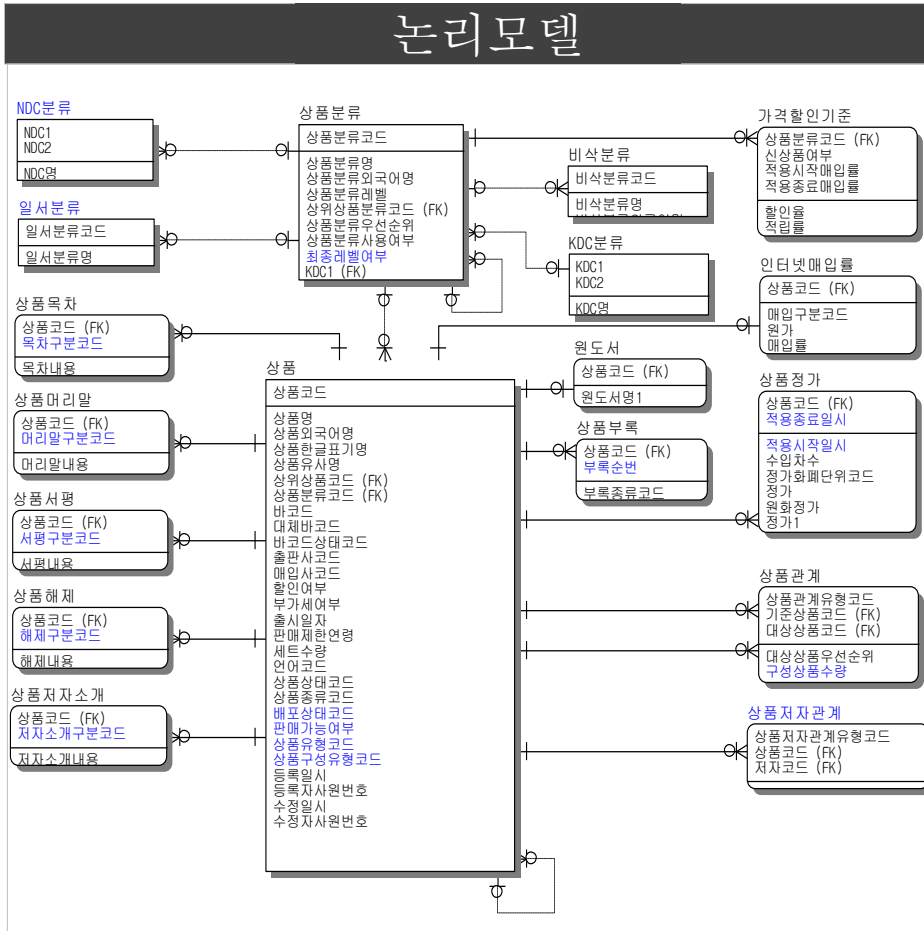
2. 데이터 값 품질 VS 데이터 무결성

□ 데이터 값 관점의 주요 발견사항

분류	주요 발견사항	주요 오류수준	품질지표
필수값분석	- 농가등록화면에서 농가/단체명, 농가주소 등은 필수항목으로 정의되어 있으나, 실 데이터가 NULL이나 공백 등으로 누락된 오류 발생	농가/단체명 4.41% 농가주소 4.44%	정확성-입력값
유효값분석	- 시료채취여부 등은 유효값(Y/N, 0/1)이 존재되어 사용하거나 유효 값을 벗어난 데이터가 존재	시료채취여부 1.05% 행정처분요청여부 0.05%	정확성-범위·형식
패턴분석	- 분번, 표준인증번호 등은 정해진 패턴을 벗어나거나 한글 및 특수문자가 포함되는 오류데이터 존재	표준인증번호 0.58% 분번 0.02%	정확성-범위·형식
날짜분석	- 인증 시작일에 유효하지 않은 날짜 및 공백이 포함된 날짜데이터가 존재함.	인증시작일 0.03% 신청일자 0.02%	정확성-범위·형식
명칭분석	- 품목 명칭, 농가단체명 등에 비완성형 문자나 숫자 발견	품목명칭 0.02% 농가_단체명 0.003%	정확성-입력값
코드분석	- 필지주소코드, 품목코드 값이 기준정보에 미 정의된 코드 값을 사용함.	필지주소(법정동) 0.06% 품목코드 0.01%	정확성-참조관계
관계분석	- 인증기관인증정보승계의 인증기관 일련번호가 인증마스터에 존재하지 않음.	인증신청일련번호 0.02%	정확성-참조관계
업무규칙분석	- 인증번호의 경우 일정한 규칙에 따라 구성되어 되지만 규칙에 위배되는 데이터가 존재함. - 농가정보 및 조건에 따라 필지정보는 업무적으로 중복이 허용되지 않으나 중복 등록됨. - 인증 신청일자와 인증 교부일자 간의 날짜 선후관계를 위배한 데이터가 존재함. - 인증구분이 재포장 일 경우 업태의 값이 입력되어야 하나 누락된 경우가 발생	표준인증번호 유효성 0.78% 필지정보 중복 10.84% 신청일자, 교부일자간 시간순서 0.85% 인증구분 완전성 0.85%	정확성-업무규칙



3. 데이터모델 vs 데이터 모델



4. 데이터 무결성

데이터 무결성	데이터베이스 제약	무결성 강화 방법
개체 무결성	<ul style="list-style-type: none">• PK 제약• Unique 제약	<ul style="list-style-type: none">• 트리거• 사용자 정의 함수• 애플리케이션
참조 무결성	<ul style="list-style-type: none">• FK 제약	
도메인 무결성	<ul style="list-style-type: none">• Not Null 제약• Check 제약• Data Type & Length	
사용자 정의 무결성		

5. 데이터 베이스 제약(1/2)



5. 데이터 베이스 제약(2/2)



III. ROI를 고려한 품질개선

1. Why 데이터베이스 제약 조건을 설정하지 않을까요?
2. 데이터베이스 제약조건 vs 성능
3. 데이터베이스 제약조건 vs Look
4. 데이터베이스 제약조건 vs 개발생산성
5. 데이터베이스 제약조건 vs 기타
6. 도메인 무결성

ROI (Return On Investment)

: 비용대비 수익 비율

$$\text{ROI} = \frac{(\text{수익} - \text{매출원가})}{\text{매출원가}}$$

1. Why 데이터베이스 제약 조건을 설정하지 않을까요?

구분	내용
성능	<ul style="list-style-type: none">• 제약조건을 설정하면, 인덱스가 많아지고 성능이 느려지지 않을까?
Lock	<ul style="list-style-type: none">• 제약조건을 설정함으로써, 트랜잭션의 동시성 저하가 발생하지 않을까?
개발이슈	<ul style="list-style-type: none">• 개발난이도 증가할지 않을까?• 개발 생산성이 떨어지지 않을까?
기타	<ul style="list-style-type: none">• 과거 프로젝트에서 제약조건을 설정하였다가 고생한 경험이 있습니다.

2. 데이터베이스 제약조건 - 성능

■ DB암호화 에 따른 성능테스트

업무	테스트 상황	암호화 방식	응답속도	CPU부하	비고
배치	1000만건 암복호화	API방식	약 2~50분	약 3~18% 증가	암호화 알고리즘, 암호화 모듈 구성 등에 따라 차이
		Plug-In 방식	약 30 ~ 40분	약 30% 증가	테스트 조건에 따라 매우 큰 차이
OLTP	100User, 1분 동안 연속적으로 주민번호 1건 조건검색	API방식	0.3 초	-	암호화 전 0.2초
		Plug-In 방식	0.5 초	-	

- 암호화에 따른 저장공간의 증가, 메모리 및 CPU 사용량의 증가로 인한 시스템의 성능저하
- 해결을 위한 시스템증설, 설계변경 및 프로그램(SQL)변경 검토 필요

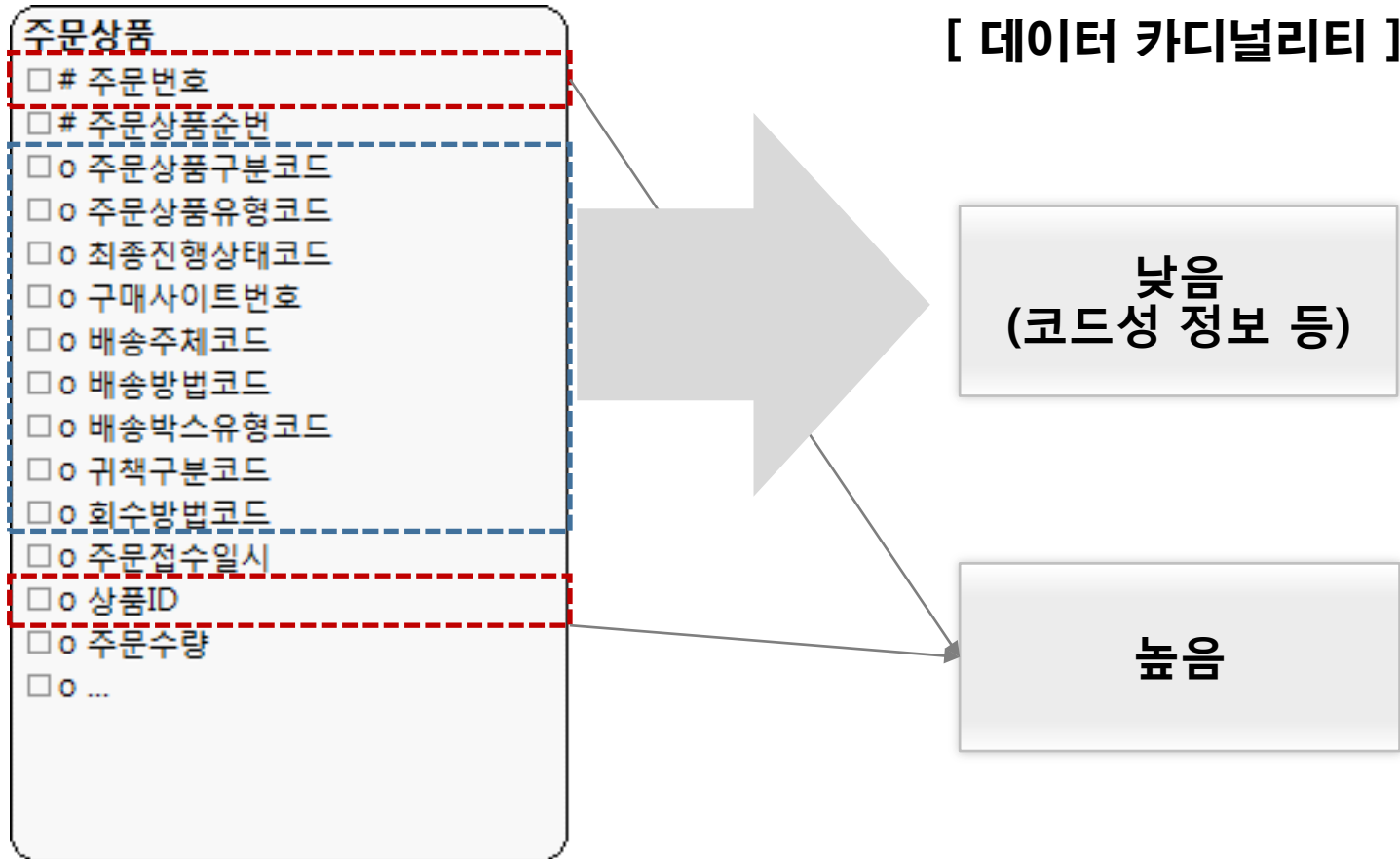
2. 데이터베이스 제약조건 - 성능

[100만 건 INSERT 테스트]

테스트 상황	측정지표	INSERT ... SELECT ...			Conventional Insert		
		FK 개수			FK 개수		
		0	1	10	0	1	10
낮은 카디널리티 (상품분류코드 등) 카디널리티=10 ※ PK 인덱스 H=1	소요시간 1	10.28초	11.32초	15.48초 3	64.77초	69.58초	83.30초
	QUERY	9,721	9,575	9,289	5,292	5,376	5,335
	CURRENT	44,407	43,420	43,470	1,054,726	2,053,830	11,054,036
높은 카디널리티 (계좌번호 등) 카디널리티=1,000,000 ※ PK 인덱스 H=3	소요시간 2	10.99초	17.95초	74.72초 4	68.69초	79.11초	112.64초
	QUERY	14,978	14,993	14,402	8,233	8,381	8,333
	CURRENT	69,238	3,067,750	30,085,449	1,073,585	4,074,523	31,092,797

- ① FK 없을 때 4.4만, FK 증가하더라도 CURRENT 블록 액세스 변함 없음 (1건당 0.0000032 초 증가)
- ② FK 없을 때 6.9만, FK 하나 증가할 때마다 300만개 CURRENT 블록 액세스 증가 (1건당 3 블록증가)
- ③ FK 없을 때 100만, FK 하나 증가할 때마다 100만개 CURRENT 블록 액세스 증가 (1건당 1 블록증가)
- ④ FK 없을 때 100만, FK 하나 증가할 때마다 300만개 CURRENT 블록 액세스 증가 (1건당 3 블록증가)

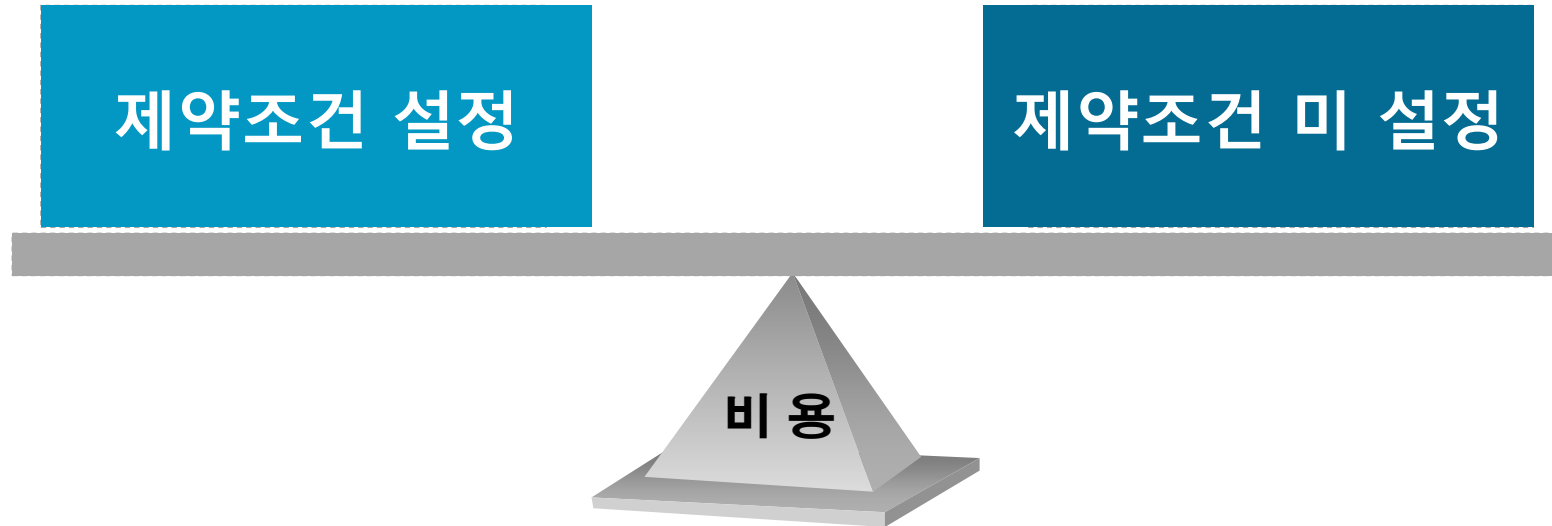
2. 데이터베이스 제약조건 - 성능



테이블을 구성하는 대부분의 FK제약 조건 대상은 코드 성 정보로 카디널리티 가 낮으며 따라서 FK 증가에 따른 성능저하는 미미 함

2. 데이터베이스 제약조건 vs 성능 - ROI

시스템 성능상의 차이는 있으나, 고객 서비스 성능상의 차이는 미비함



- ✓ 시스템 성능 : CPU, I/O, 메모리 등의 시스템 사용률로 **절대적인 수치**
- ✓ 서비스 성능 : 서비스(응용프로그램) 에서 요구하는 응답시간으로, **상대적인 값**

3. 데이터베이스 제약조건 - TM Lock 호환성

TX Lock

- 트랜잭션 Lock → DML 로우 Lock 구현
- 트랜잭션당 하나

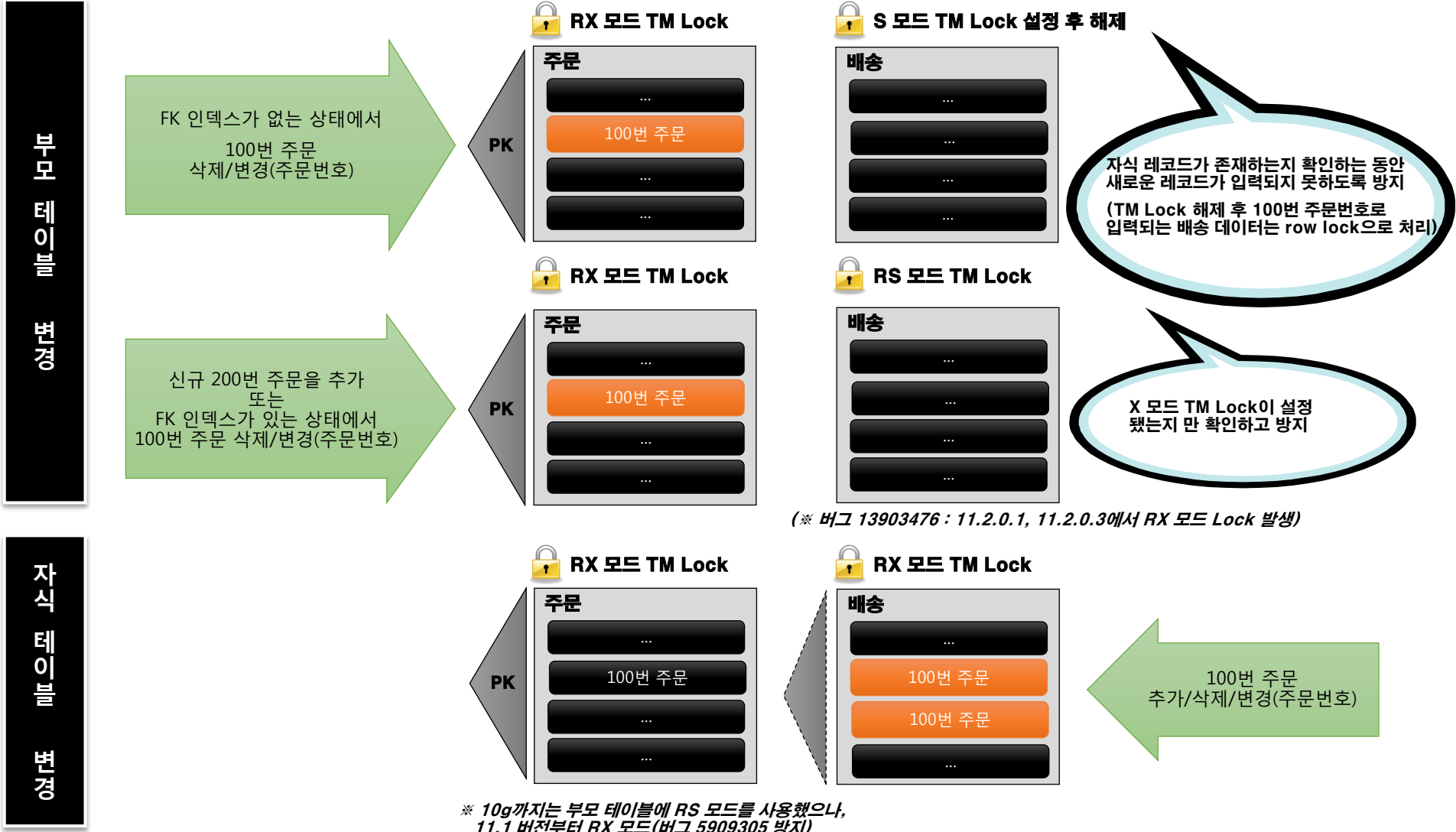
TM Lock

- DML 테이블 Lock
- 각 트랜잭션이 액세스하는 테이블마다 하나씩

	Null	RS	RX	S	SRX	X
Null	○	○	○	○	○	○
RS	○	○	○	○	○	
RX	○	○	○			
S	○	○		○		
SRX	○	○				
X	○					

- RS : row share
- RX : row exclusive
- S : share
- SRX : share row exclusive
- X : exclusive

3. 데이터베이스 제약조건 - FK 제약 설정 시 , TM Lock



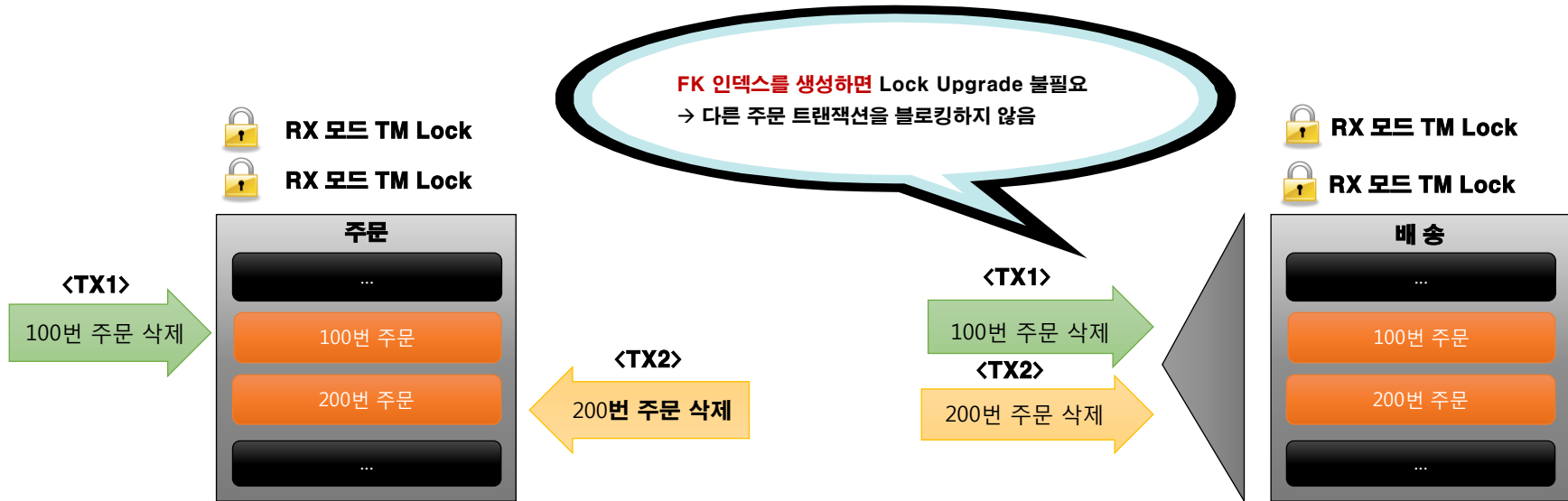
3. 데이터베이스 제약조건 - TM Lock에 의한 블로킹 (사례1)



	Null	RS	RX	S	SRX	X
Null	○	○	○	○	○	○
RS	○	○	○	○	○	
RX	○	○	○			
S	○	○		○		
SRX	○	○				
X	○					

- RS : row share
- RX : row exclusive
- S : share
- SRX : share row exclusive
- X : exclusive

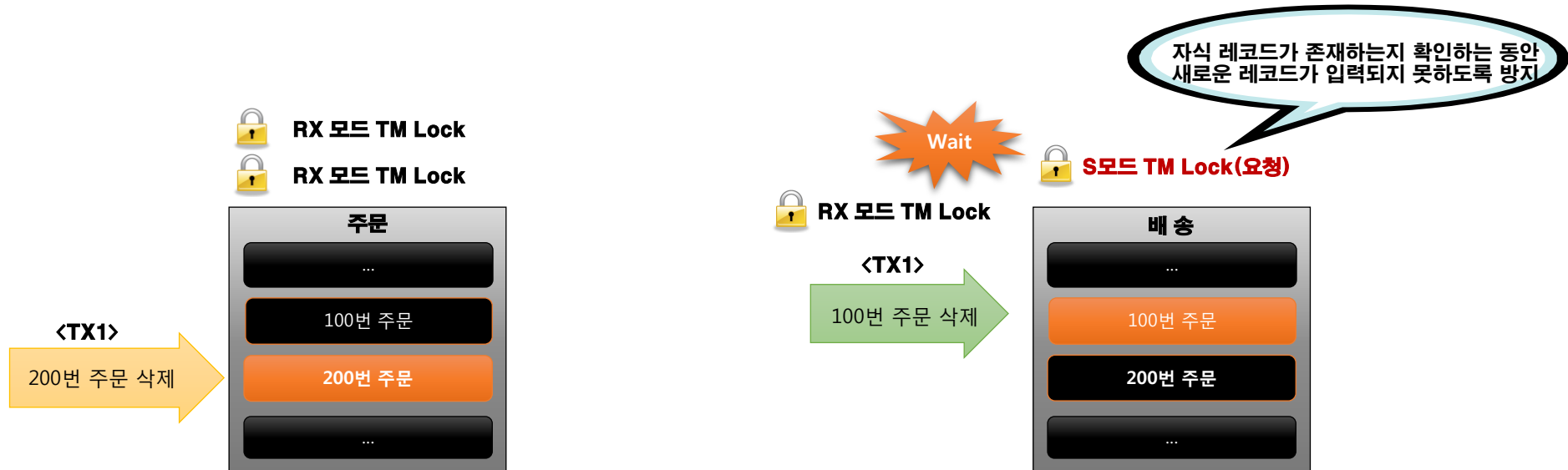
3. 데이터베이스 제약조건 - TM Lock에 의한 블로킹 (사례1)



	Null	RS	RX	S	SRX	X
Null	○	○	○	○	○	○
RS	○	○	○	○	○	
RX	○	○	○			
S	○	○		○		
SRX	○	○				
X	○					

- RS : row share
- RX : row exclusive
- S : share
- SRX : share row exclusive
- X : exclusive

3. 데이터베이스 제약조건 - TM Lock에 의한 블로킹 (사례2)



	Null	RS	RX	S	SRX	X
Null	○	○	○	○	○	○
RS	○	○	○	○	○	
RX	○	○	○			
S	○	○		○		
SRX	○	○				
X	○					

- RS : row share
- RX : row exclusive
- S : share
- SRX : share row exclusive
- X : exclusive

3. 데이터베이스 제약조건 - TM Lock에 의한 블로킹 (사례2)



	Null	RS	RX	S	SRX	X
Null	○	○	○	○	○	○
RS	○	○	○	○	○	
RX	○	○	○			
S	○	○		○		
SRX	○	○				
X	○					

- RS : row share
- RX : row exclusive
- S : share
- SRX : share row exclusive
- X : exclusive

3. 데이터베이스 제약조건 - Guidelines for Indexing Foreign Keys

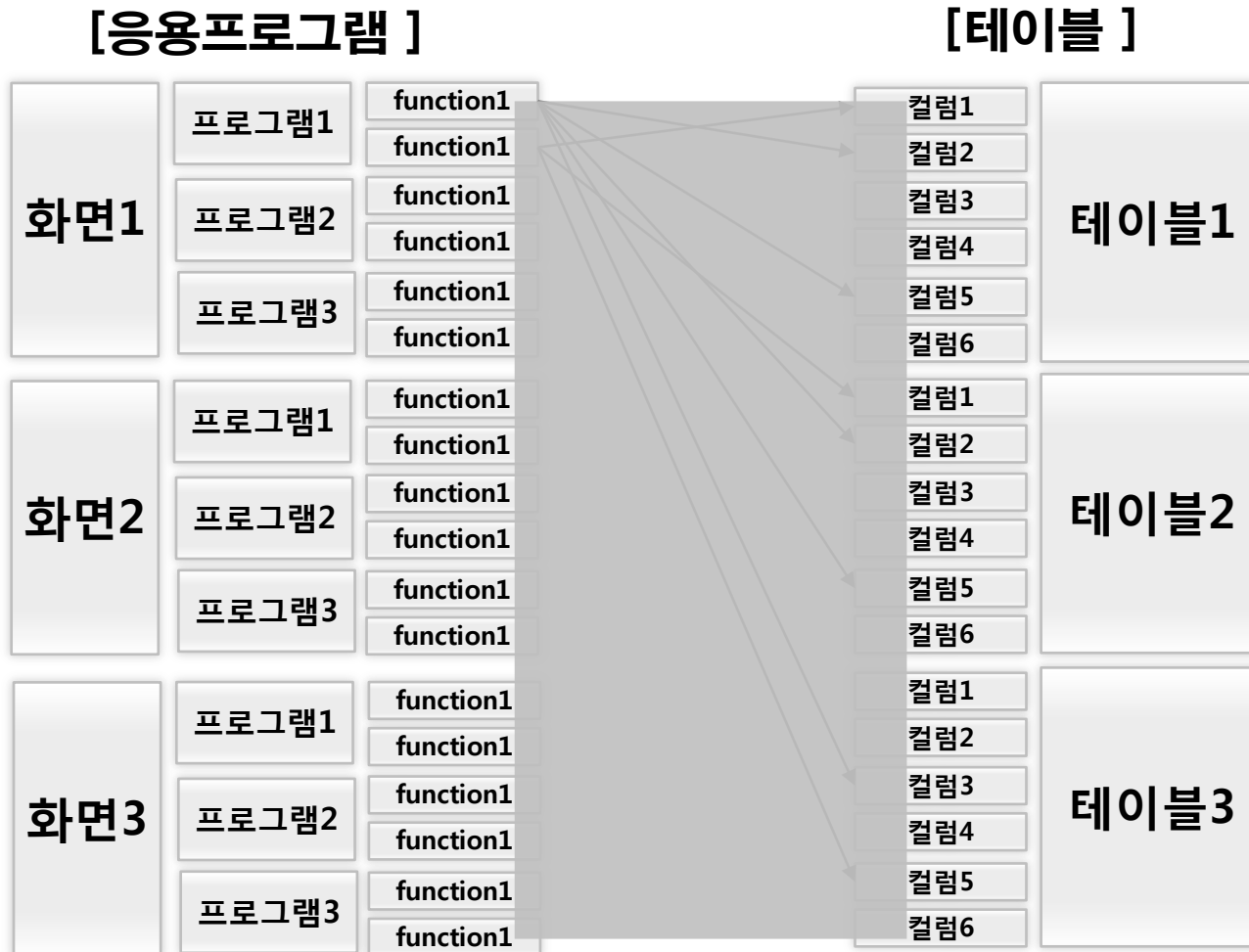
III. ROI를 고려한 품질개선

- 오라클 曰

“Index foreign keys **unless** the matching unique or primary key is **never** updated or deleted.”

→ 부모 키가 변경되거나 지워질 수 있는 FK 제약이라면,
반드시 **인덱스를 생성하라!!**

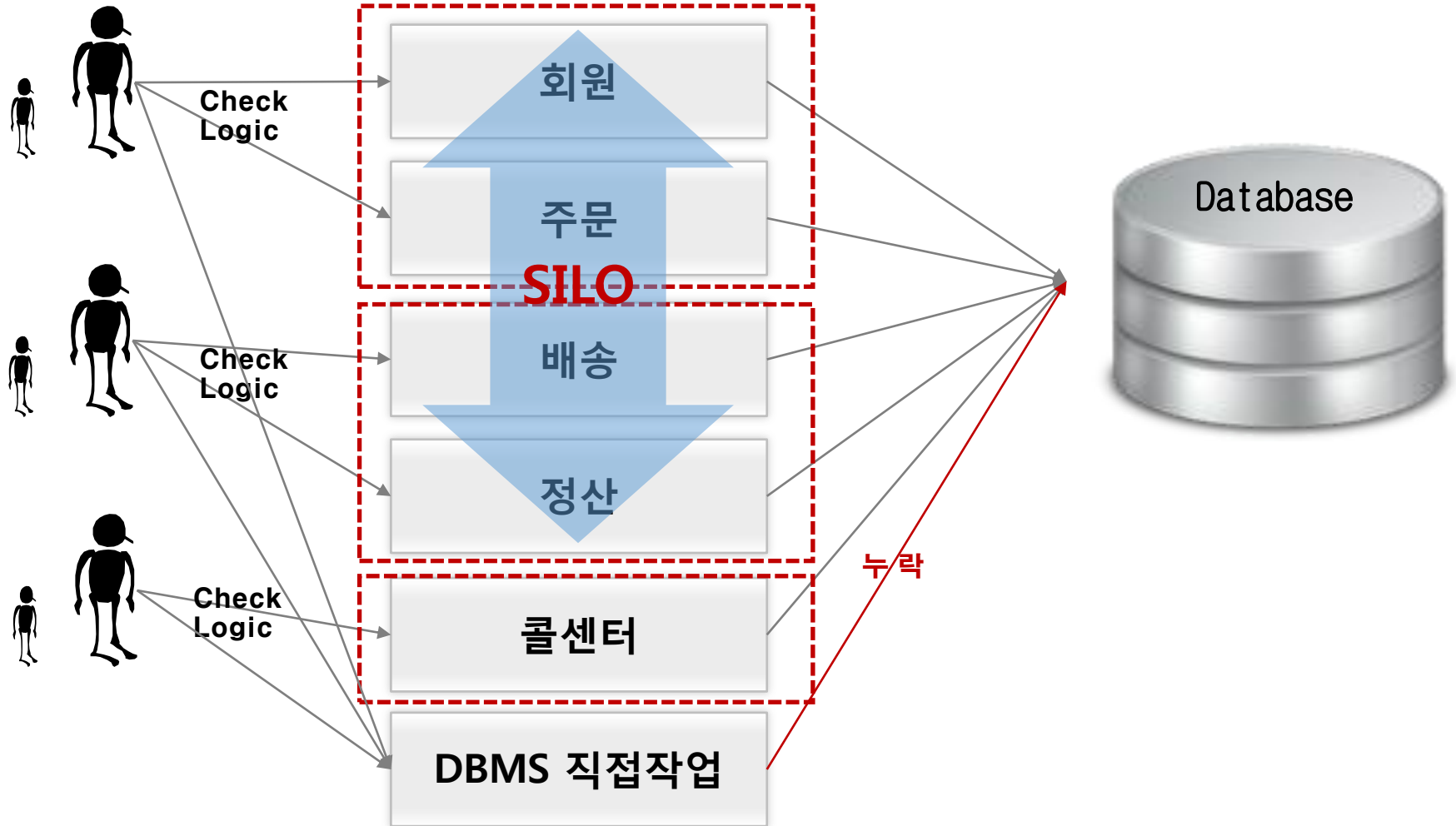
■ 응용프로그램 vs 테이블



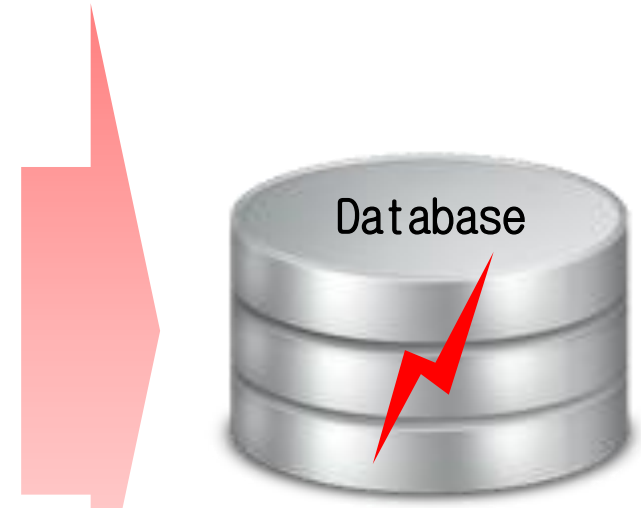
3. 데이터베이스 제약조건 - 응용프로그램을 통한 정합성 체크

[개발자 그룹]

[업무별 응용프로그램]



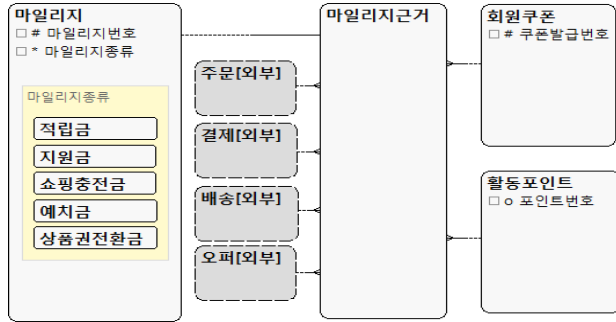
- ✓ 프로그램 **개발자의 능력**이 모두 동일한가요 ?
- ✓ 응용프로그램에서 **모든 제약조건에 대한 로직 구현 및 누락된 로직**을 확인할 수 있나?
- ✓ 프로그램을 통한 제약조건 처리가 **FK보다 성능, 신뢰성 면에서 우수한가요?**
- ✓ 응용프로그램을 통하지 않고, **DBMS에 직접 작업**하는 경우에 대한 통제 절차는?



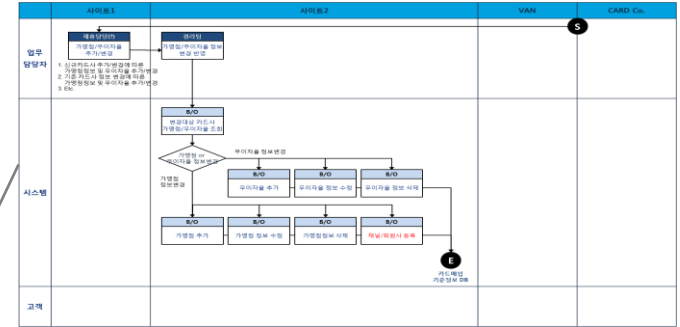
참조 무결성 오류 발생

4. 데이터베이스 제약조건 - 개발생산성

[데이터 모델]



[프로세스 흐름도]



[DBMS]



[화면 정의서]

■ 기타 결제내역조회

1. 화면설계: 2014.01.10 ~ 2014.01.10

- 검색수단: 전체
- 검색자: []
- 승인구분: 전체
- 카드번호: []
- 승인번호: []

2. 기타 검색목록

순번	승인일시	승인번호	결제수단	승인구분	카드번호	검색자	사용포인트
1	2013-01-10 0:10	20140110-159737	신세계포인트	승인	4579-7205-****-8023	홍길달	1,000
2	2013-01-10 0:12	20140110-159737	전지상용권-기프트카드	취소	4579-7205-****-8023	홍길삼	0
3	2013-01-10 0:13	20140110-159737	신세계모바일상품권	승인	4579-7205-****-8023	홍길삼	1,000
4	2013-01-10 0:13	20140110-159737		승인	4579-7205-****-8023	홍길삼	1,000
5	2013-01-10 0:13	20140110-159737		승인	4579-7205-****-8023	김철사	0
							3000

- ✓ 데이터의 발생 **선후관계**, 업무규칙, 데이터의 참조 무결성은 모두 개발자의 몫?
- ✓ 시스템 적인 **통제 수단**은?

4. 데이터베이스 제약조건 - 개발생산성

▪ 개발 생산성 관련 이점

• 애플리케이션 제약조건 로직 보완

- DBMS의 FK Constraint 체크 기능을 활용하면, 개발자의 능력 여하에 관계없이 **참조 무결성에 대해서는 동일한(높은) 수준의 데이터 품질을 보장받을 수 있음**
- 애플리케이션을 거치지 않은 **DB 작업에 대해서도 참조 무결성이 보장됨**

• 테이블 간 관계 명시 효과

- FK제약을 걸면, **참조 테이블과 종속 테이블 간의 관계를 명확하게 보여 줄 수 있음**
- 테이블 간 **관계를 명확히 모르는 상태에서 개발이 진행될 경우, 잘못된 조인으로 인한 성능 저하 등의 이슈가 발생할 수 있음**

4. 데이터베이스 제약조건 vs 응용프로그램- ROI

구분	DBMS FK제약	응용프로그램 체크
<p>비용 항목</p>	<ul style="list-style-type: none"> • FK 설정에 따른 인덱스 수 증가 • 인덱스 증가에 따른 Storage 증가 	<ul style="list-style-type: none"> • 개발자 교육 • 모든 응용프로그램에 대한 제약조건 구현 • 제약조건 누락에 대한 재 작업 • 제약조건 체크에 대한 성능개선활동 • 데이터 베이스에서 직접 작업하는 데이터에 대해서는 체크불가 • 오류 데이터에 대한 정제작업
<p>비용</p>	<ul style="list-style-type: none"> • DBA 인덱스 관리대상 증가 • Storage 증가비용 	<ul style="list-style-type: none"> • 교육비용 • 지속적인 프로그램 관리비용 증가 • 성능개선 비용 • 오류데이터 정제비용

5. 데이터베이스 제약조건 - 기타

✓ 부모 테이블 변경

※ delete cascade 옵션을 사용한 경우는 논외

버전	부모 테이블	자식 테이블		비고
		with indexes 또는 Insert	Update/Delete without indexes	
7.1.6 이하	RX	Null	S	
8.1.7 이상		RS ^{주)}	S	트랜잭션 종료할 때까지 S 모드 유지
9.0.1 이상			S → RX	update/delete 수행 중에만 S 모드, 수행을 마치면 RX 모드로 downgrade
9.2.0 이상			S → Null	update/delete 수행 중에만 S 모드 사용

주) 버그 13903476 : 11.2.0.1, 11.2.0.3에서 RX 모드 Lock 발생

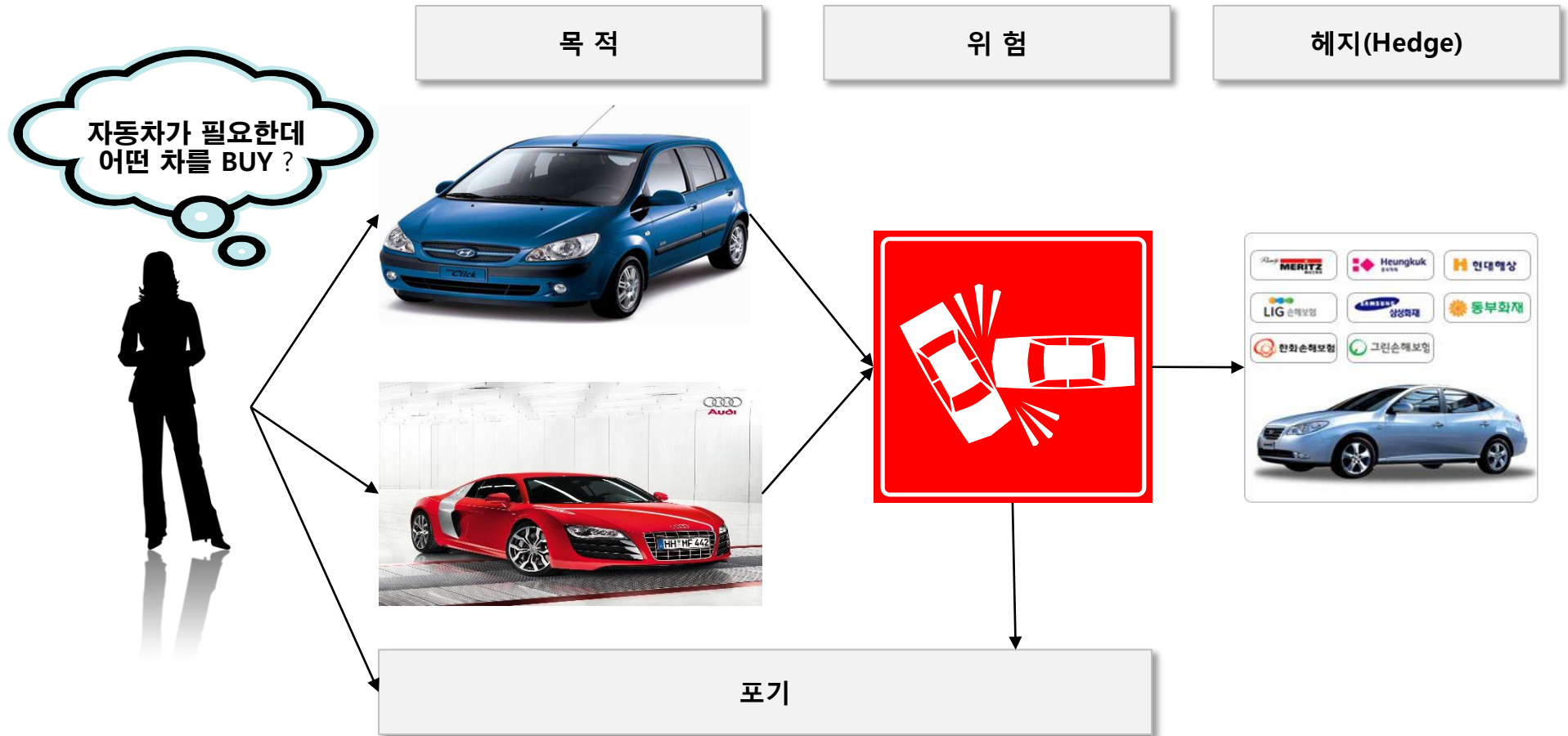
✓ 자식 테이블 변경

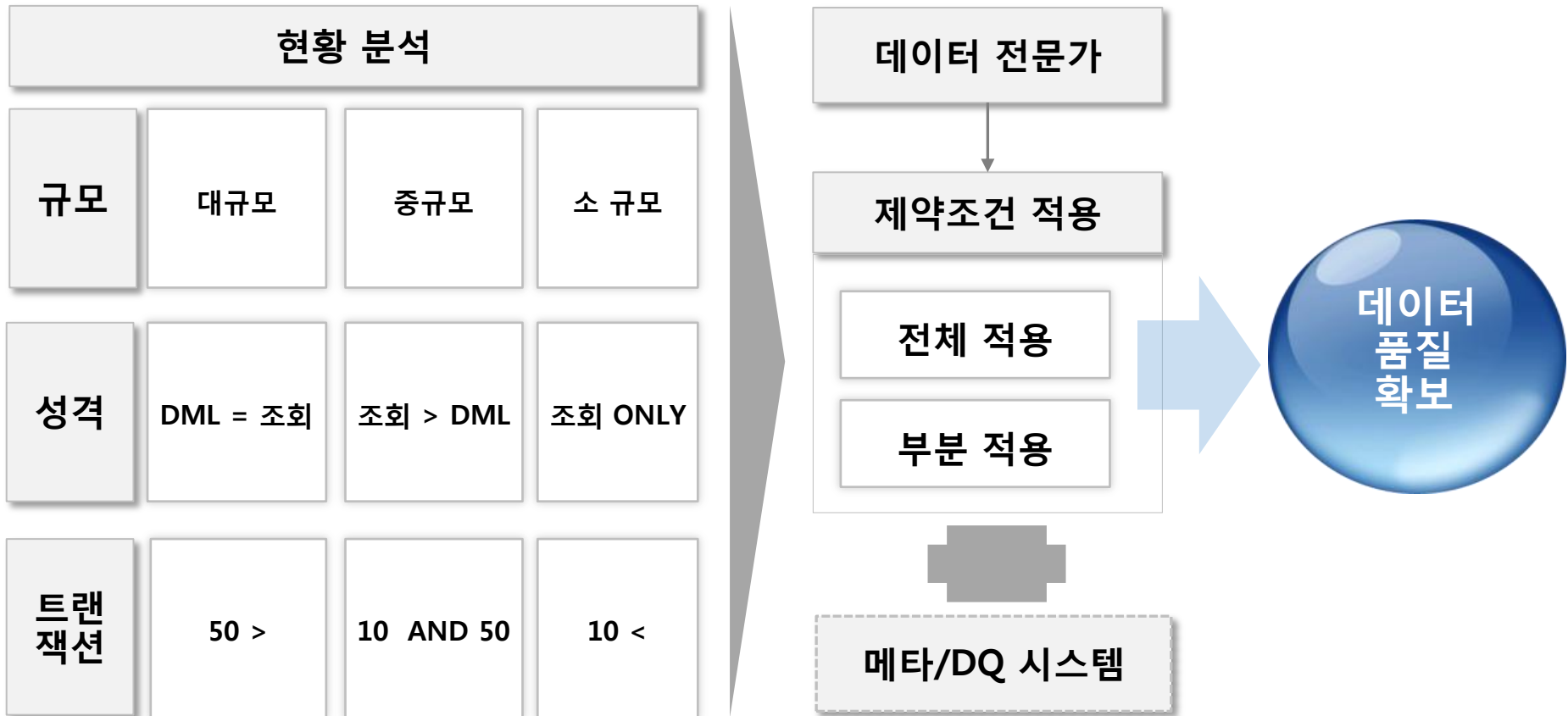
버전	자식 테이블	부모 테이블		비고
		with indexes	without indexes	
7.1.5 이하	RX	Null	S	
7.1.6 이상		Null		
9i 이상		RS		
11.1 이상		RX		버그 5909305(600 에러 발생) 방지하기 위한 변화

6. 도메인 무결성

구분	VARCHAR2	DATE
기본	<ul style="list-style-type: none"> 날짜 형식 : YYYY-MM-DD 	
장점	<ul style="list-style-type: none"> 범위 조건에 LIKE 패턴 매칭을 사용할 수 있다는 점에서 개발편리성 예제) WHERE 일자 LIKE '201401%' 	<ul style="list-style-type: none"> 잘못된 데이터가 입력될 수 없으므로, 데이터의 품질을 높일 수 있음
단점	<ul style="list-style-type: none"> DBMS 차원에서 데이터의 유효 값을 확인할 수 없음 프로그램에서 Validation Check 누락 시 오류 데이터 발생으로 인한 데이터 품질저하 	<ul style="list-style-type: none"> 프로그램 개발 시 소스코드가 길어질 수 있음 (예제) WHERE 일자 BETWEEN TO_DATE('20140101','YYYYMMDD') AND LAST_DAY(TO_DATE('20140101','YYYYMMDD'))

IV. 맺음말





Q&A

Thank You ○