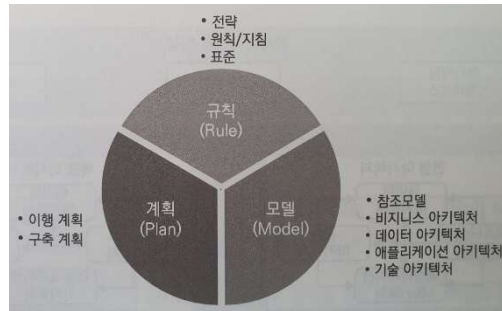


I. 전사 아키텍처 이해

1 장. 전사 아키텍처 개요

1 절. 전사 아키텍처 정의

- 전사 아키텍처는 조직의 방대한 시스템을 이해 및 파악 하기위한 설계도이다.
- 즉, 기업의 전체 정보화 설계도를 **전사 아키텍처(Enterprise Architecture)**라고 한다.
- 전사 아키텍처는 기업의 목표와 요구를 효율적 지원하기 위해 각 IT 인프라의 각 부분들이 어떻게 구성되고 작동되어야 하는가를 체계적으로 기술한 것이다.
- 비즈니스(활동), 데이터, 애플리케이션(S/W), 기술(H/W) 등의 측면에서 분석하고 표현하여 구축
- 따라서 EAI 를 수립하기 위해서는 ‘전사’의 범위를 명확히 정해야 한다.



가. 모델 관점 : 비즈니스와 시스템 아키텍처로 크게 나뉨

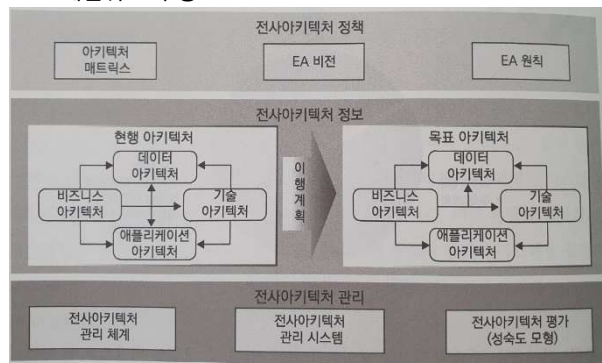
나. 규칙 관점 : 일관성의 유지를 위하여 준수해야하는 원칙을 수립하고 적용하여 아키텍처의 연속적인 모습이 관리됨

다. 계획 관점 : 현재와 목표 아키텍처를 정의한 후 수립한 목표 아키텍처를 달성할 수 있는 이행 전략과 이행 계획을 의미

2 절 전사 아키텍처 프레임워크

: 전사 아키텍처 활동에서 얻어지는 산출물을 분류하고 조직화하고 이를 유지 관리하기 위한 전체적인 틀을 정의하는 것

1. 프레임워크 구성 :



가. 전사 아키텍처 정책

: 전사 아키텍처의 정보를 어떻게 구성하고 수립을 통해 기업이 달성하고자 하는 궁극적인 모습은 무엇이며, 효과적으로 관리 및 활용하기 위한 원칙은 어떤 것인지 등을 정의하는 것이다.

나. 전사아키텍처 정보

: 아키텍처 정보 구축을 위해 먼저 정보의 영역(비즈니스, 데이터, 애플리케이션, 기술)을 구분하며, 이를 **아키텍처 도메인**이라고 함. 아키텍처 도메인이란 아키텍처 매트릭스 상에서 뷰의 관점으로 구분한 것. 현재와 목표 아키텍처를 이런 도메인 별로 정보를 구축.

다. 전사아키텍처 관리

: 정의된 전사 아키텍처 정보를 지속해서 유지 관리하고 효과적으로 활용하기 위해서는 EA 관리 체계의 정립과 EA 관리 시스템의 구축이 필요하며, EA 관리 수준을 제고하기 위해서는 지속적으로 평가하고 개선할 필요가 있음. (성숙도 모형)

※ EA : Enterprise Architecture

전사 아키텍처 프레임워크의 상세 구성

전사 아키텍처 정책	아키텍처 매트릭스	전사 아키텍처의 정보를 체계적으로 분류한 틀. 기업이 관리하려고 하는 EA 정보의 수준과 활용 계층을 결정하는 수단
	EA 비전	전사아키텍처 수립을 통해 기업이 궁극적으로 달성하고자 하는 모습
	EA 원칙	전사아키텍처 정보를 효율적으로 구축하고, 기업의 목적에 맞게 전사 아키텍처 정보를 효과적으로 활용하기 위해서 조직 구성원이 공유해야 할 규범
전사 아키텍처 정보	현행 아키텍처	아키텍처 도메인별로 정의된 산출물에 대해, 기업의 현재상태를 아키텍처 정보로 정의한 것
	이행 계획	아키텍처 도메인별로 현재 모습에서 바람직한 목표 모습으로 이행하기 위한 이행 전략과 해야 할 일을 정의한 것
	목표 아키텍처	아키텍처 도메인별로 정의된 산출물에 대하여 기업이 궁극적으로 달성하고자 하는 아키텍처의 상태를 아키텍처 정보로 정의한 것
전사 아키텍처 관리	전사아키텍처 관리 체계	전사아키텍처 거버넌스라고도 함. 구축된 전사 아키텍처를 유지하고 개선하기 위한 제도적 기반을 수립하는 것
	전사아키텍처 관리 시스템	전사 아키텍처의 정보 관리 효율성을 제고하고 전사아키텍처 정보의 공유를 활성화하기 위해 구축하는 정보시스템. 일반적으로 모델링 도구, 리포지터리, 포털 등으로 구성
	전사아키텍처 평가 (성숙도 모형)	EA 관리와 활용 수준의 제고를 위해서는 EA에 대해 주기적으로 평가하고 개선점을 도출하여 반영해야함.

2. 아키텍처 도메인 구성

- 기업이 아키텍처 매트릭스를 어떻게 정의하느냐에 따라 다름. 즉 각각의 아키텍처를 사용자 관점으로 매트릭스를 구성하여 필요한 것을 정의함. p.27 <표 4> 참고

- BA(Business Architecture), AA(Application Architecture), DA(Data Architecture), TA(Technical Architecture) 등이 있음. 각 아키텍처는 사용자에게 따라 활용 내용과 수행과업이 달라짐

가. 비즈니스 아키텍처

계획자 관점	전사사업모델, 조직모델
책임자 관점	업무기능 모델
설계자 관점	프로세스 모델
개발자 관점	업무 매뉴얼

: 기업의 경영 목표를 달성하기 위한 업무 구조를 정의함. 타 아키텍처의 방향을 정의하고 검증하는 시발점이 됨

나. 어플리케이션 아키텍처

계획자 관점	전사 어플리케이션 영역모델
책임자 관점	애플리케이션 모델
설계자 관점	컴포넌트 모델, 클래스 모델
개발자 관점	프로그램 목록

: 기업의 업무를 지원하는 전체 애플리케이션을 식별하고 연관성을 정의하고, 전체 애플리케이션 구조를 체계화.

다. 데이터 아키텍처

계획자 관점	전사 데이터 영역 모델
책임자 관점	개념 데이터 모델
설계자 관점	논리 데이터 모델
개발자 관점	물리 데이터 모델
데이터영역 분류	업무(운영계+정보계) 데이터, 메타데이터

: 기업의 업무 수행에 필요한 데이터의 구조를 체계적으로 정의하는 것.

라. 기술 아키텍처

계획자 관점	전사기술영역모델, 기술참조모델
책임자 관점	개념 데이터 모델
설계자 관점	기술 아키텍처 모델
개발자 관점	기술자원목록, 제품 목록

TRM 과 SP 구축을 통해 애플리케이션의 이식성과 확장성 강화, 벤더독립성 확보, 시스템상호운용성 강화.

개별기업에서도 기술참조모델 정의, 아키텍처 매트릭스에서 TRM 을 계획자 수준의 산출물로 포함

3 절. 전사 아키텍처 참조 모델

1. 참조 모델 정의

: 아키텍처 구성 요소를 식별하여 표준화한 것으로 기관이나 기업의 전사아키텍처를 수립할 때 참조하는 추상화한 모델이다. 특히 정부에서는 전사적 아키텍처 구성에 필요한 정보화 구성요소를 표준화된 분류체계와 형식으로 정의해 참조 모델로 보급하고 있음. 이를 통해 전사 아키텍처의 표준을 삼고 다른 시스템에서도 재사용이 가능하도록 한다.

2. 참조 모델 현황

가. 성과 참조 모델(PRM : Performance Reference Model)

나. 업무 참조 모델(BRM : Business Reference Model)

다. 서비스 참조 모델(SRM : Service Reference Model)

라. 데이터 참조 모델(DRM : Data Reference Model)

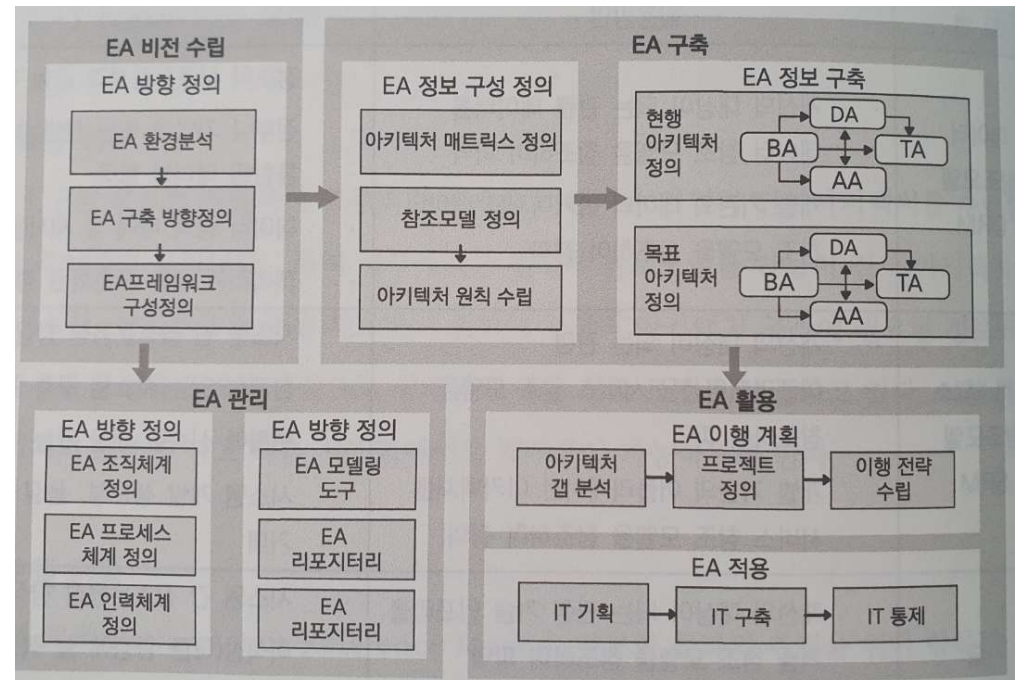
마. 기술 참조 모델(TRM : Technical Reference Model)

4 절. 전사 아키텍처(EA, Enterprise Architecture) 프로세스

: 전사 아키텍처 프로세스는 전사 아키텍처를 구축하고 관리하는 전체 절차에 관한 것으로, 작업의 단계와 공정, 작업 내용 등을 정의하는 것이다.

1. 전사 아키텍처 프로세스 구성

: EA 비전 수립 -> EA 구축 -> EA 관리 -> EA 활용 단계



2 장. 전사 아키텍처 구축

1 절. 전사 아키텍처 방향 수립

1. 개요

： 전사 아키텍처가 나아가야할 방향성을 정해야 함. 그렇지 않으면 다수의 이해관계자에게 정보시스템에 대한 공동의 목표와 목적을 제공할 수 없어 의견 충돌의 원인이 된다. 또한, EA 방향 수립은 경영 환경과 경영 전략에 능동적으로 대응하며 발전하기 위하여 전사 아키텍처 변화 요인에 대한 분석 작업을 수행하는 과정이다. EA 방향 수립은 전사 아키텍처 환경 분석, 전사 아키텍처 구축 방향 정의, 전사 아키텍처 프레임워크 정의 등으로 구성된다

2. 전사 아키텍처 환경 분석

： 기업 내·외부 환경을 분석하고, 이해 관계자로부터 EA 수립을 위한 요건을 도출한다. 즉, 전사 아키텍처 환경 분석의 수행 과정은 비즈니스 내·외부 환경을 분석하고, IT 내·외부의 법제도, 사회 및 기술 환경등을 분석하여 해당 프로젝트가 무엇으로부터 영향을 받는지를 분석하여 전사의 범위를 정하는 것이다.

3. 전사 아키텍처 구축 방향 정의

： 전사 아키텍처의 환경 분석 결과를 토대로, 구축 목적을 정의하고 구축 범위를 정의하는 것이다. 또한, 전사 아키텍처의 비전과 핵심목적, 핵심가치도 수립한다. **기업이 지향해야 하는 가치, 도입 목적의 근거, 목적 달성 시 추구하는 효과(기대효과)**등의 정의를 포함한다.

※ 전사아키텍처의 수립 목적은 상호운용성 증대, 정보화 투자의사 결정 체계 구축, 비즈니스 변화에 대한 신속한 대응 체계 구축, 전사적 정보화 표준 정립, 고객 지향의 정보화 체계 구축 등이다.

4. 전사 아키텍처 프레임워크 정의

： 전사 아키텍처 수립을 위해서는 EA 프레임워크가 정립되어야 함. 이는 전사 아키텍처 활동에서 얻어지는 산출물을 분류하고 조직화 및 유지관리 하기 위한 전체적인 틀이다. 이를 통해 EA 를 기술하는데 필요한 정보 유형을 식별하고, 이들 정보 유형을 논리적으로 구조화하고 정보 유형 간의 관계를 구조화한 것이다.

2 절. 전사 아키텍처 정보 구성 정의

： 전사 아키텍처 정보는 기업을 잘 이해하기 위해 필요한 업무와 정보기술에 대한 정보로써 활용 할 만한 가치가 있고 관리가 용이한 정보라고 정의할 수 있다. 또한 전사 아키텍처 정보는 업무와 정보기술의 구성 요소와 구성 요소 간의 관계를 포함한다. EA 정보는 가능한 한 변하지 않는 구성 요소를 도출해서 정의해야하고, 관리비용 대비 효과를 고려해야함.

가. 아키텍처 매트릭스 개념

： 아키텍처 매트릭스는 EA 프레임워크 핵심 구성요소로, 전사를 설명하는 모델과 원칙 정보를 통일된 시각으로 볼 수 있는 논리적 틀이다. 아키텍처 매트릭스는 의사결정 유형(관점)과 정보 유형(뷰)의 두 축을 기준으로 2 차원의 매트릭스 형태를 띠고 있다.

나. 아키텍처 매트릭스 구성

： 선진 사례를 무조건 도입하기보다는 기업의 현황을 고려하여 기업의 EA 목표 달성에 필요한 구조로 아키텍처 매트릭스를 정의하는 것이 바람직하다.

뷰/관점	업무	데이터	응용 (Application)	기술
계획자	○조직 구성도/정의서	○데이터구성도/정의서	○응용 시스템 구성도/정의서	○표준프로파일 ○기반 구조 구성도/정의서 ○기술 자원 목록
책임자	○업무 관계도/기술서 ○업무 기능 분할도/기술서	○개념 데이터 관계도/기술서 ○데이터 교환 기술서	○응용 시스템 관계도/기술서 ○응용 기능 분할도/기술서	○기반 구조 관계도/기술서
설계자	○업무 절차 설계서	○논리 데이터 설계서 ○데이터 교환 설계서	○응용 기능 설계서 ○응용 분산 시스템 설계서	○기반 구조 설계서 ○시스템 성능 설계서
개발자	○업무 매뉴얼	○물리 데이터 모델	○응용 프로그램 목록	○제품 목록

다. 아키텍처 매트릭스 정의 시 고려 사항

- 매트릭스를 정의할 때는 일반적인 아키텍처 개념을 포함하면서 매트릭스에 포함된 산출물이 범위와 목적에 적합하게 정의되었음을 조직 내 모든 계층의 사람이 확인할 수 있어야 함.
- 조직적, 정치적, 지리적 특성, 조직의 편견 등 다양한 조직 문화와 의사결정 구조가 반영되어야 함.
- 실제 시스템과 아키텍처 개발 표준에 대한 준수성을 높이고 조직별로 통일된 접근이 가능하도록 정의되어야하고, 통일성과 일관성을 유지할 수 있어야한다.
- 아키텍처 도메인은 상호 간에 연계성을 가져야 하며, 전사 차원에서 통합적인 아키텍처 관리가 이루어지도록 해야 한다.

3 절. 전사 아키텍처 정보 구축

1. 전사 아키텍처 정보 구축 준비

：아키텍처 정보를 구축하기 위해서는 우선 기존에 작성된 자료를 **수집**해야 한다. 수집해야 할 자료는 정의된 아키텍처 매트릭스에 따라 다르다. 전사 아키텍처 정보를 구축하는 방법에는 상향식과 하향식 또는 혼합식이 있으며, 기업의 상황에 맞는 방식(주로 의사결정 방식)에 따라 선택하는 것이 바람직하다.

2. 현행 아키텍처 정보 구축

：현행 아키텍처 정보의 구축은 현재의 업무나 정보시스템에 대하여 기존의 자료를 분석하여 전사 아키텍처 정보를 구축하는 것을 말한다. 일반적으로 상위 수준의 업무 기능과 시스템에 대한 분류 체계를 정의한 후 나머지 하위의 정보 구축은 병렬적으로 수행한다.

3. 목표 아키텍처 정보 구축

：목표 아키텍처 구축은 현행 아키텍처에 대한 문제점과 개선 사항을 도출하고, 이를 목표 아키텍처에 반영하는 방식으로 진행한다. 일반적으로 비즈니스 아키텍처(BA)를 먼저 정의하고, 이를 효율적으로 지원하는 정보기술 아키텍처를 정의한다. 목표 아키텍처의 구축 범위는 전사 아키텍처 구축의 목적에 따라 다르지만, 초기 전사 아키텍처 수립 시 개념적 수준까지 정의하고, 개념적 수준 이하의 산출물 작업은 실제 시스템 구축 단계에서 수행하는 것이 일반적이다.

3 장. 전사 아키텍처 관리 및 활용

1. 전사 아키텍처 관리 체계

- 전사 아키텍처 도입도 중요하지만, 더 중요한 것은 도입 후, 얼마나 잘 관리하고 활용하는지.
- 전사 아키텍처 관리 체계는 전사 아키텍처 관리 조직, 관리 프로세스, 관리 인력을 포함한다.
- 전사 아키텍처 관리 체계는 전사 아키텍처를 수립하고 관리하기 위한 활동을 정의하는 것

2. 전사 아키텍처 관리 시스템

- 전사 아키텍처 관리 시스템은 구축, 관리, 활용하는 모든 업무 프로세스를 효과적으로 지원하기 위한 시스템을 말한다
- 전사 아키텍처 정보를 생산하는 모델링 도구 영역, 정보를 저장 관리하는 리포지터리, 정보를 활용하는 전사 아키텍처 포털, 전사 아키텍처 정보 활용 영역으로 구분할 수 있다.

3. 전사 아키텍처 활용

- 전사 아키텍처의 활용 영역은 목표 아키텍처를 달성하기 위한 이행 계획을 수립하고 그에 따라서 프로젝트를 추진하는 기준으로 삼는 것과 전사 아키텍처 정보를 활용하여 일상의 IT 업무를 지원하는 것이다.

II. 데이터 요건 분석

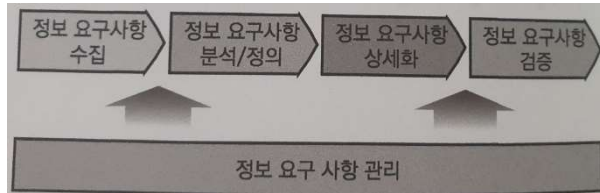
1장. 정보 요구사항 개요

1절. 정보 요구사항

1. 정의

: 업무의 개선 사항이나 신규 개발 사항으로 시스템을 통해 기능상의 목적을 달성하기 위해 요청하는 내용을 말한다. 일반적으로 사용자가 해당 시스템에 대해 바라는 것이 무엇이냐를 기술한 것이다.

2. 정보 요구사항 생명주기 모형



가. 정보 요구사항 수집

: 사용자 인터뷰, 설문지, 워크숍, 현행 시스템 분석 등을 통해 수집하며 이때 정보 요구사항은 아래 표 5를 참조하여 유형별로 정리하고, 이 때 요구사항의 중요도와 긴급성을 같이 조사하는 것이 좋습니다.

나. 정보 요구사항 분석 및 정의

: 수집된 요구 사항을 분야별 프로세스별, 단계별로 종합하고 이들 중에서 중요하고 긴급한 것 등을 분석하여 정리하는 단계로 이 단계에서 각 요구사항의 우선순위를 정함

다. 정보 요구사항 상세화

: 우선순위가 높은 것들을 중심으로 개별 사항에 대해 세밀하게 분석

라. 정보 요구사항 검증

: 비즈니스 관점, 조직관점, 어플리케이션 관점과 상관분석을 통해 검증

정보 요구사항 유형

유형	구분	설명
외부 인터페이스 요건	정의	모든 입출력에 관한 요건으로서 대외기관 간의 송수신하는 입출력 방식이 추가, 변경 되었을 경우와 각종제도 및 기준변경 시 발생하는 요건
	관리 기준	○중복성 : 기존에 동일한 인터페이스가 존재하는지 체크한다 ○표준 준수도 : 관련 국제, 국가 표준 존재 시 그에 적합한 형태로 제공
	관리 방법	항목 이름, 목적 설명, 입력의 원천 및 출력의 방향, 유효 범위, 시간, 다른 입출력과의 관계, 데이터 포맷, 최종 메시지 등이 관리되어야 함
기능 개선 요건	정의	시스템에서 입력을 받아들이고 처리하고 출력을 만들어 내는 주요 활동 및 프로세스에 대한 요건
	관리 기준	○불가변성 : 향후에 재변경되지 않도록 근본적인 개선방안을 요청해야 함 ○범용성 : 많은 사용자가 편리하게 사용할 수 있는 내용 우선 요청해야 함
	관리 방법	입력에 대한 유효 체크, 정확한 처리 순서, 비정상 상태에 대한 반응(오버플로우, 통신 장비, 에러처리), 매개변수의 기능, 출력과 입력의 관계, 입출력 순서, 입력을 출력으로 변환하는 공식 등이 포함되어 관리되어야 함
성능 개선 요건	정의	동시 사용자 수, 처리하는 정보의 양과 종류, 트랜잭션 소요시간 등이 있다
	관리 기준	○실현 가능성 : 현행 기술 수준과 서비스 특성을 고려해 구현 가능한지 확인 후 제시해야 한다 ○측정 가능성 : 측정이 불가능한 모호한 형태로 제시되면 안 된다.
	관리 방법	각 서비스 특성을 고려하여 정적·동적 기준을 만들고 해당 기준에 맞는지 모니터링을 통해 항상 관리해야 한다.
보안 개선 요건	정의	물리적 접근통제(제한구역, 통제구역 등) 및 사용통제(인증, 암호화, 방화벽 등)에 대한 요건을 말한다.
	관리 기준	○불가변성 : 향후에 변경되지 않도록 근본적인 개선방안을 요청해야 한다. ○실현 가능성 : 현행 기술 수준과 서비스 특성을 고려해 구현 가능한지 확인 후 제시해야 한다.
	관리 방법	보안 관리가 필요한 장비에 대한 등급 관리가 필요하고 해당 등급별 이용자 등급 관리가 필요하며, 접근 방식에 대한 기준이 있어야 한다. 해당 기준에 따라 모니터링 작업을 통해 안정적인 서비스가 제공되도록 관리해야 한다.

2 절. 정보 요구사항 관리

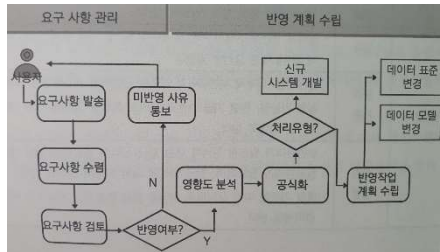
1. 정의 및 관리 목적

：사용자의 요구를 수집하고 분류하여 반영하는 작업 절차를 말하며, 데이터·어플리케이션·비즈니스 등의 요구사항을 전부 포함하는 통합관리 프로세스를 정립해야 한다.

2. 정보 요구사항 관리 프로세스

가. 업무 흐름 프로세스

- 요구사항 **발송** : 요구사항 정의서 양식에 기록하여 정보시스템 담당자에게 발송한다.
- 요구사항 **수렴** : 정보 요구사항 정의서를 수집 후 규칙에 맞는지 확인하고 해당 건을 처리할 담당자를 지정하여 이송한다.
- 요구사항 **검토** : 반영 여부를 판단하고 불가능할 경우 미반영 사유와 함께 발송한 담당자에게 전달
- **영향도 분석** : 영향을 받는 설계서, 기존 어플리케이션, 데이터베이스 등을 파악한다.
- **공식화** : 담당자들과의 협의를 통해 규모 및 시간, 시급성에 따라 반영 유형 결정
- 반영 작업 계획 수립 : 영향 분석 결과를 근거로 관련 담당자들과의 미팅 후 반영 계획 수립. 작업 일정 계획에는 표준과 설계서 변경, 데이터베이스 및 어플리케이션 수정, 테스트, 이관 작업이 명시 되어야 함.



나. 역할별 담당 임무

역할	담당 업무
사용자	<ul style="list-style-type: none"> ○ 변경 요청 ○ 변경 여부 확인 ○ 미결 사항에 대한 의사결정 실시
담당자	<ul style="list-style-type: none"> ○ 사용자 정보 요구사항 접수 ○ 사용자 정보 요구사항에 대한 기본적인 검토 ○ 반영 여부 결정을 위한 사용자와 1차 미팅 ○ 접수요건에 대한 처리방식 및 처리 기한 결정 ○ 관련 부서별 담당자 수집 및 요건협의 주도 ○ 사용자 정보 요구사항 반영 ○ 테스트 및 검증 ○ 사용자 반영결과 통보
데이터 아키텍처 전문가	<ul style="list-style-type: none"> ○ 사용자 정보요구 사항에 대한 표준/데이터베이스/어플리케이션 차원에 대한 영향도 분석 및 보고 ○ 접수된 요구사항에 대한 표준 준수 여부 체크 ○ 영향도 분석을 통한 수정 및 변경 계획 수립 ○ 표준 제시 및 준수 여부 검토

2 장. 정보 요구사항 조사

1 절. 정보 요구사항 수집

1. 정보 요구사항 수집 형태

：관련 문서, 사용자 면담, 워크숍, 현행 업무 처리 매뉴얼(업무 관련 규정이나 지침 등), 현행 정보 시스템 관련 산출물을 통해 수집한다.

※ 요구: 시스템이 가져야 할 기능이나 시스템이 만족하여야 할 조건

예) 현금 인출기

· 기능 요구 : 데이터의 등록, 조회, 수정, 삭제 등의 업무적으로 필요한 사항

· 비기능 요구 : 성능, 효율, 반응 시간, 제약 조건 등의 시스템(H/W)적으로 필요한 사항

※ 요구 사항 명세 : 요구사항 추출과 분석을 통해 얻어지는 “고객이 시스템에 요구하는 기능 또는 서비스와 제약사항에 대한 설명”

· 프로젝트 수주/발주를 위한 제안의 기초가 됨

· 프로젝트 계약의 기초가 됨

2. 관련 문서 수집

가. 문서 수집 목적

：구현 시스템의 대상과 범위를 명확히 정의하기 위해

나. 문서 수집 자료

：경영 계획에 대한 자료, 정보시스템에 대한 자료, 과거 수행한 컨설팅 보고서, 전산 처리 업무 매뉴얼, 현업 부서 업무 자료 등이 있다.

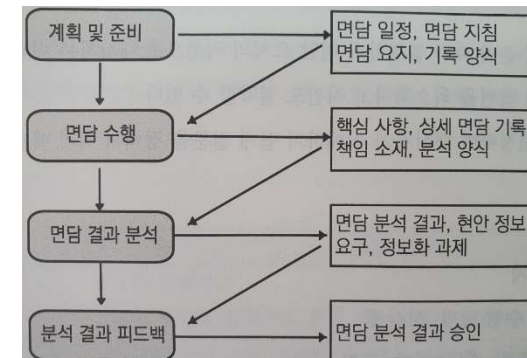
다. 문서 수집 원칙

- 기존의 문서를 변형하지 않고 수집, 정보시스템 자료는 별도의 정리 양식을 사용해서 수집한다.
- 수집문서를 바탕으로 경영 및 정보시스템 현황에 대한 요약표를 작성하여 숙지한다.
- 수집된 문서들은 계획 수립 기간, 문서 관리자를 지정하여 운영한다.
- 유형별 문서 분류 방식을 결정한 후에 일정한 장소에 보관한다.
- 통상 대외비 형식이 많으므로 문서 보안 관리에 주의한다.

3. 사용자 면담

- 실무자와 대면하여 질의응답을 통해 정보를 수집한다.
- 개별 면담은 워크숍보다 융통성이 좋고 유연한 진행이 가능하다.
- 적은 시간으로 면담이 가능하므로 일정 수립이 용이하고 추가 계획을 세우기가 쉽다.

가. 사용자 면담 진행



<p>※ 사용자 요구 사항 명세</p> <ul style="list-style-type: none"> · 고객이 이해할 수 있도록 작성· 시스템이 제공할 서비스와 수행될 때의 제약 조건을 자연어, 그림 등을 이용하여 표현한다.
<p>※ 시스템 요구사항 명세· 개발자를 위해서 작성 시스템이 제공해야 할 서비스, 제약조건이 무엇인지 체계적으로 자세히 적는다.</p>
<p>※ 문제점의 이해</p> <ul style="list-style-type: none"> · 요구분석에 있어서 사용자가 현재 안고 있는 문제점을 이해하는 것이 중요 · 사용자의 입장에서 현황을 파악하고 사용자가 원하는 문제 해결이 어떤 것인지를 이해한다. <p>※ 요구 사항의 추출· 요구분석가의 일은 신문, 방송의 기사를 취재하는 기자의 업무와 비슷하다.</p> <ul style="list-style-type: none"> - 면담, 실제 업무 수행 등의 기법 사용

- 1) 계획 및 준비
- 가) 면담 주제 선정
- 작업과 면담대상자 수준에 따라 결정하고 면담 요지의 작성으로 대상자는 미리 답변을 준비할 수 있다. 상상에 의한 답변을 최소화하고 시간도 절약할 수 있다.
 - 질문항목은 명확히 선정하여 대상자가 이해하기 쉽게 질문을 정해서 미리 배포하여 답변을 준비
 - 현업 부서용 면담 요지
 - 면담의 취지, 목적, 수행 방법, 시간 등
 - 프로젝트의 개요 : 목표, 범위, 기간, 조직
 - 업무의 향후 수행 방향에 대한 의견
 - 면담 대상자가 소속된 부서의 업무 현황 및 개선 요구 사항
 - 현재 사용하는 정보시스템에 관한 의견
 - 프로젝트에 관한 의견 : 요구 사항, 프로젝트 참여 방안 등
 - 전산 부서용 면담 요지
 - 면담의 취지, 목적, 수행 방법, 시간 등
 - 프로젝트의 개요 : 목표, 범위, 기간, 조직
 - 기획 분야 현황 및 계획 : 전산 부서 조직 및 인력, 연혁, 계획, 문제점, 과제 등
 - 시스템 분야 현황 및 계획 : 조직 및 인력, 시스템/네트워크 구성, 시스템 운영 절차, 향후 계획
 - 어플리케이션 분야 현황 및 계획 : 조직 및 인력, 어플리케이션 구성, DB 구성, 진행 중 개발 업무, 개발 및 유지 보수 계획, 문제점 및 계획 등
 - ‘6 하 원칙’에 따른 질문
 - 분석 대상 업무에 누가 관련되는가? (Who, 관계자들의 작업, 사용자 수준)
 - 현재의 상태는 무엇인가? (What, 문제를 일으킨 상태, 제안된 시스템의 기능)
 - 새로운 시스템은 언제 완성되어야 하나? (When)
 - 새로운 시스템은 어떤 환경에 놓일 것인가? (Where, 새 시스템에서의 조직, 환경)
 - 왜 새로운 시스템을 고려하게 되었나? (Why)
 - 새 시스템을 어떻게 작동할 것인가? (제약, 하드웨어 요구, 비용, 사용 언어)

나) 면담 진행 팀 구성

모델	활용 방안
면담자	<ul style="list-style-type: none"> · 면담을 진행하자 · 면담의 취지를 설명하고 면담 대상자에게 질문한다.
기록자	<ul style="list-style-type: none"> · 면담 대상자의 답변 내용을 기록한다. (내용을 요약하지 말고 표현 그대로를 기억해야 함) · 면담 대상자의 답변 내용을 충분히 이해하고 기록하기 위하여 면담 대상 업무에 대한 사전 지식이 있어야 함 · 면담 종료 시 기록 내용 중 주요 사항(수치, 업무 분장 및 책임소재조직 등에 대한 내용) 확인
관찰자	<ul style="list-style-type: none"> · 면담이 수행 의도되로 진행되고 있는지 관찰 · 면담이 주제의 범위를 벗어나는 경우, 주의를 환기 · 면담자가 놓치는 부분에 대해 보충 질문 · 최종적으로 면담의 종료에 대해 판단

- 다) 면담 대상자 선정
- 작업에 따라 면담 대상을 선정하고 대상자는 업무에 대해 명확한 이해를 하는 사람이어야 함.
 - 대상 선택을 위해 전체 조직 구성도와 프로젝트 범위를 검토하고 해당하는 대상자를 프로젝트 후원자나 사용자측으로부터 추천받는 것이 좋다.
 - 유사 업무를 지원하는 조직이 여러 명일 경우 차이점 파악을 위해 적어도 2명 이상 면담
- 라) 면담 일정 수립
- 후원자의 지원을 얻은 후 대상자들에게 프로젝트의 목적과 범위를 통보하고 관련 문서 자료 요청
 - 초기 단계에서 전체일정이 정해져야 하며 최소 1주일 전에 세부 일정을 확정하고 가능하면 하향식으로 일정 수립
 - 면담 시간은 1.5~3 시간으로 필요 시 집단 면담을 수행한다. 하루 3 회 이하로, 세부 일정은 담당 면접 진행팀도 함께 참여하도록 함.
- 마) 면담 준비
- 이용 가능한 자료를 활용하여 대상자의 업무 활동 역할, 신상 명세, 경력, 개인적 성향 파악
 - 대상자에게 배포하지 않는 면담 시나리오 준비
 - 도표를 이용하여 효율성 증대
 - 면담 수행 전 주제 목록 등을 중간/실무 관리자에게 배포하여 면담 진행자들을 미리 소개하고 사용 가능 자료도 요청
 - 면담 수행 30 분 전 최종 준비 상황 점검
- 2) 면담
- 가) 면담 시작
- 시작 30 분 전 다른 면담 진행 팀과 함께 필요한 정보 요구와 진행 순서를 점검하고 각자 역할 점검
 - 프로젝트 목적, 범위, 일정 등을 먼저 설명하며, 필요할 경우 주제나 질문을 수정
 - 면담은 복수의 팀에서 수행될 수 있으므로 수행 방식을 통일하기 위해 모든 절차가 면담 지침에 세세한 문구까지 모두 반영되어 있어야 한다.
- 나) 면담 주제 토의
- 준비된 면담 요지에 따라 내용은 모두 기록지에 기록
 - 개방적 질문을 사용하여 주제나 질문지의 순서와 범위를 벗어나지 않도록 함

- 효율적인 진행을 위해 엔터티 관계도나 업무 기능 계층도(메뉴 구성도)를 이용할 수 있으며, 답변을 가정하지 않으며 진행 도중 간간히 수집된 정보나 순서 등을 확인
- 면담 결과의 후속 분석 작업을 위해 공통의 기준으로 사용할 수 있는 표준 기록 양식을 정함
- 진행 중 대상자의 주요 책임 업무를 정의하고 각 업무가 시간과 같은 논리적인 순서에 따르는 지 확인

3) 면담 결과 분석

- 면담 진행 팀은 기록된 내용과 면담 중의 응답에 대한 개인적 의견을 고려하여 면담 결과를 정리하고 면담 종료 후 팀 전원이 참석하여 주요 이슈 정리
- 현재 업무와 발생하는 차이점에 주의하고 가능하면 면담 대상자의 업무 용어를 사용하며 별도의 양식은 없음
- 의문 사항, 추가 사항은 즉시 면담 대상자에게 확인하고 필요한 경우 추가 면담

4) 분석 결과 피드백

- 정리 내용이 없거나 필요 경우 기록지 전체를 확인 받는다 결과에 대해 대상자 본인에게 승인을 받는 것이 중요. 또한 이 과정에서 대상자는 수정사항을 제시할 수 있으며 기록지에 대상자의 의견을 기입 하는 것이 좋다.
- 개인별 피드백이 곤란한 경우 실무자 전원을 대상으로 워크숍을 진행할 수 있으며, 민감한 사항은 사전 조정 필요
- 면담 수행 시 고려 사항
- 면담 시간 준수, 비밀 보장, 기대 수준 설정, 면담 범위 준수
- 적절한 대상자 선정, 적절한 응답 유도, 면담 내용 문서화, 잘못된 선입견의 배제, 애매한 질문 배제

4. 워크숍

가. 워크숍 개요 및 목적

- 전문 진행자의 진행 하에 프로젝트의 현업부서, 전산부서 주요 구성원들이 함께 참여하는 회의로 정치·개인적 요소를 피하고 다양한 정보를 빠르게 추출하여 공유할 수 있다.
- 경영층 또는 현업 부서장의 공통된 의견 도출
- 유사한 업무 또는 관련된 업무 등을 수행하는 부서를 따로 면담하는 데 드는 노력을 절감한다.
- 전문가들의 판단력을 이용하여 최적의 결론을 도출해냄

나. 워크숍 준비

· 워크숍 과제 선정과 계획 수립, 참가 대상자 선정, 대상자에 대한 사전 브리핑 및 교육, 키오프 모임 수행, 워크숍 자료 준비, 설비와 물품 준비, 워크숍 장소 선정, 워크숍 기간 선정 프로그램 준비

다. 워크숍의 수행

· 프로젝트 관리자와 현업 책임자는 이해관계를 가지고 있기 때문에 워크숍 진행자로 부적절하며 진행자는 전문 용어를 자제하고 사용자 입장에서 언어를 사용해야 함

1) 워크숍 개시

· 시작을 알리고 간략한 인사와 일정 확인

2) 워크숍 수행 준비 : 목적과 접근 방법의 개요 설명, 작업을 수행하기 위한 기법 습득

3) 워크숍 수행

· 구체적인 수행 방식은 형태나 목적에 따라 다르게 수행, 목적에 맞게 진행

4) 워크숍 종료

· 진행 사항 요약, 도출된 사항 공유, 전체에게 1 차적으로 검토 받아야함.

5. 현행 업무 조사서

- 전체 부서에 동일한 기준으로 조사하는 것을 원칙
- 동일한 업무를 수행하는 부서가 여러 개인 경우 표본 추출 및 발체 조사도 가능
- 조사서의 양식은 간단하고 이해가 쉬워야 하며 작성 방법, 표본을 같이 배포
- 잘못 작성이나 추가 내용을 넣어야 하는 경우를 고려하여 일정 분배
- 사용자가 처리하고 있는 업무 기능을 정리된 양식으로 기록하여 향후 작업에 도움되도록 함

6. 현행 프로그램/데이터 관련 문서

- 요구 사항을 세부적으로 진행하기 위한 사전 단계로서 현행 시스템을 파악하기 위한 작업
- 현행 시스템 프로세스 구조는 계층적 구조로 표현하고, 향후 완전성 검증을 위한 비교자료로 활용하며 현행 프로세스 전체적인 부분을 이해

2 절. 정보 요구사항 정리

1. 정보 요구사항 정리

가. 사용자 면담 정리

: 기록에 오류가 있는지 사용자에게 확인받아야함

나. 업무 조사서 정리

: 프로세스 업무 흐름, 문서, 사용중 시스템 정리

다. 워크숍 정리

: 워크숍 목적, 내용, 해결과제에 대한 상태, 특이사항 정리

라. 기타기법 정리

: 설문조사법, 심층면접법, 초점집단면접법, 투사법, 인적 관찰과 기계적 관찰, 인위적관찰과 자연적 관찰 등

2. 정보 요구 우선순위 분석

: 본 분석 방법은 실제 사용시 시간/노력 소모가 심하므로. 현재는 우선순위, 중요도, 소요 비용 등을 고려한 비교적 판단이 용이한 방법으로 판단됨

가. 화폐가치 산출 방법

: 최종적으로 구해진 가치가 높을수록 우선순위 높음

1) 정보 요구사항 나열 후 기업/시스템 차원 중요성 별로 1~3 점 부여

2) 상호 관련성에 1~5 점 부여 및 세 점수를 곱함

3) 백분율 환산 후 금액으로 환산 -> 우선순위 결정

나. 상대적 중요도 산정 방법

1) 정보 요구사항 나열 후, 업무에 기여하는 수준에 따라 1~5 점 부여

2) 현행 시스템이 요구사항을 얼마나 충족하나에 따라 1~3 점 부여

3) 요구사항 매트릭스 작성 후 관련이 크면 9 점을 부여하고 나머지는 상대 점수 부여

4) 부여한 3 가지 점수에 대해 가중치 결정

3 절. 정보 요구사항 통합

1. 정보 요구사항 목록 검토

: 전사 관점에서 동일한 정보 요구사항이 있었는지 검토한다.

2. 정보 요구사항 목록 통합/분할

- 수집/정리한 요구사항을 별도 양식으로 취합 조정 후, 중복 도출 여부를 검토

- 동일 부서 내, 그리고 서로 다른 부서 간 중복 요구사항이 없었는지 검토

3 장. 정보 요구사항 분석

1 절. 분석 대상 정의

1. 현행 업무 분석 대상 정의

가. 분석 대상 자료

: 분석 대상 자료에는 현행 업무 흐름도, 현행 업무 설명서, 현행 업무 분장 기술서가 있다.

나. 분석 대상 업무 영역 선정

: 분류기준에 따라 현행 업무목록을 작성한다. 분류기준이란 통상적으로 현행 업무 기능 분해도의 단위 업무 또는 업무 분장상의 구분 등을 의미한다.

2. 현행 시스템 분석 대상 정의

가. 분석 대상 현행 시스템 선정

: 업무 분석 프로젝트의 수행범위를 정확히 파악하는 것이 선행되어야만 업무 영역별 대상 현행 시스템 선정 작업이 가능하다.

나. 분석 대상 현행 시스템 관련 자료

○현행 시스템 구성도 ○현행시스템의 분석, 설계 및 개발 보고서

○화면, 장표 및 보고서 레이아웃

○현행 시스템 테이블 목록 및 테이블 정의서

○프로그램 목록

○사용자 및 운영자 지침서

○시스템 지원 및 유지보수 이력

○시스템 개선 요구사항 등

- 수집된 문서의 평가는 유용성, 완전성, 정확성, 유효성을 기준으로 보완하면 좋음

다. 추가적인 분석 대상

- 보다 상세히 분석하기 위해 사용자 뷰도 분석 대상에 포함

- 데이터 뷰는 전체적인 정보 중 일부만을 바라보는 관점을 가짐

- 이러한 사용자 뷰가 종합되어 나타나는 것이 화면, 수작업 파일, 수작업/전산 양식, 보고서 등

라. 분석 단계 : 현재의 상태를 파악하고 요구를 정의. 문제 해결 및 구현될 시스템의 목표를 명확히 도출

라. 명세서 작성 과정

- 어떤 기능을 가지게 될지 정확히 작성

- 구현·운영에 제약 조건 명시

- 개발자와 사용자가 합의한 성능에 관한 사항 명시

- 명세서는 사용자와 개발자의 계약을 나열한 문서이다.

2 절. 정보 요구사항 상세화

: 현행 업무 영역 관련 자료 및 시스템 관련 자료에 대해 분석하고, 분석 산출물을 토대로 사용자의 정보 요구사항을 보완하고 비기능적 정보 요구사항을 포함하여 정보 요구사항 정의서를 보완

※ 기능적 요구사항

- 외부 요소가 직접 상호작용을 통해 시스템에게 요구하는 기능 또는 서비스.

- 시스템이 사용자를 위해 무엇을 해야하는 지 거시적으로 기술

※ 비기능적 요구사항

- 시스템이 수행하는 기능 이외. 시스템이 만족해야하는 제약조건(기술적, 성능적)

- 신뢰성, 확장성, 이식성, 보안이 중요

1. 프로세스 관점의 정보 요구사항 상세화

: 말그대로 시작/종료가 명확하고 입출력을 갖는 프로세스 관점에서 확인

가. 수행 절차

- 프로세스 중심의 프로세스 목록, 업무 흐름도를 포함한 업무조서서를 바탕으로 프로세스 계층도와 프로세스 정의서 작성

- 프로세스 기준으로 필요한 정보 항목과 산출되는 정보 항목 정리

- 표준화 과정을 통해 통합성/분리성 여부 검토 후 사용자 정보 요구사항을 충족하는 목록 작성

나. 수행 작업 지침

1) 프로세스 분해 상세화

- 하위에 업무가 포함되지 않을때까지. 단위 업무 수준까지.

- 분해 수준은 3 차 수준까지. 각 프로세스가 균형있도록 분해

- 높은 응집도, 낮은 결합도를 유지한 모듈성 확보

2) 정보 항목 도출 및 표준화

- 프로세스별 CRUD 로 구분

3) 정보 항목별 통합성 검증

- 세세히 분리된 정보를 전사 관점에서 통합 및 분리 여부 검토

2. 객체지향 관점의 정보 요구사항 상세화

: 객체지향 관점에서는 유스케이스 다이어그램으로 기능적 정보 요구사항을 정리함

1) 유스케이스

- 사용자 관점에서 작성

- 액터 : 시스템과 상호작용하는 외부 엔터티

- 유스케이스 : 액터에게 보이는 시스템의 기능과 외부 동작

- 확장<<Extends>> 점선화살표 - 포함<<include>> 점선화살표

2) 클래스 다이어그램

: 시스템을 구성하는 클래스의 구조를 나타내고 객체들의 공통 구조와 동작들을 추상화.

- 객체, 클래스, 속성, 오퍼레이션, 연관 관계

- 연관관계: 선으로 표현

- 전체/부분관계(Aggregation) 다이아몬드로 표시

3) 순서 다이어그램

: 시스템의 동작을 정형화하고 객체들의 메시지 교환을 시각화

- 참여객체를 추가적으로 나타내고 객체 사이에 일어나는 상호작용을 파악하기 위해 사용

- 구성 요소 : 액터, 객체, 메시지, 수직선

- 이벤트 흐름을 나타냄

4) 상태 다이어그램

- 객체가 갖는 여러 상태와 상태 사이의 전환을 표현.

- 상태란 객체가 만족하는 조건을 의미

- 단일 객체의 동작을 나타내고 상태를 점검해 빠진 오퍼레이션 점검

- 원은 객체의 상태를 나타내며, 화살표는 전환을 나타냄

5) 액티비티 다이어그램

- 시스템에서 수행되는 작업/ 클래스의 메소드 (액티비티)를 표현

- 액션 상태인 상태 다이어그램

- 액티비티와 전환 사이의 제어 흐름

- 동근 사각형 : 액티비티 / 화살표 : 전환 / 동기 막대: 제어 흐름 동기화 / 다이아몬드: 분기

3 절. 정보 요구 사항 확인

1. 수행 절차

- 분석 결과 도출된 산출물에 대해 재검토 기준 정의 및 계획 수립

- 완전성, 정확성, 일관성, 안정성 등 다양한 측면에서 재검토

2. 수행 작업 내용

○재검토 계획 수립 : - 대상이 되는 분석 결과 및 정보 요구 사항 정의서 산출물 확인

- 대상 산출물 별로 재검토 기준(체크리스트 정의)

○재검토 실시 : 재검토 계획서 작성&승인. 업무 영역별 재검토.

○보완 결과 확인: 재검토 결과를 토대로 산출물 보완, 정보 요구 사항 정의서의 안전성 분석

3. 수행시 고려사항

- 일관성 있는 기준 및 일정을 수립하여 모든 인력의 공감대 형성이 중요하고 이를 바탕으로 작업 수행

- 재검토는 두 번 이상 진행

- 세션별로 해당 기준에 초점을 맞추어 수행하며, 효율성을 고려하여 참여대상 수 선정

4 절. 정보 요구 분석 방법

1. 구조적 분석 방법

가. 구조적 분석의 관점

- 시스템을 기능적 관점에서 다룸

- 소프트웨어 시스템을 거시적 관점에서 데이터와 데이터를 처리하는 프로세스로 봄

- 데이터보다는 처리, 기능 위주로 분석.

- 프로세스는 추상적인 개념부터 점차적으로 세분화

나. 구조적 분석 방법

- 전통적인 데이터 처리 시스템을 개발하는데 적절.

- 개발될 시스템의 모형을 만드는 것이 중요

- 프로세스간 데이터의 흐름 파악이 중요

- 자료 흐름과 가공 절차를 그림 중심으로 표현

- 시스템을 구성하는 요소들의 상호 작용과 기능들을 나타냄

다. 구조적 분석 방법의 가정

- 시스템이 무엇을 하는지에 집중

- 상위에서 하위 순서로 문제를 분할

라. 구조적 분석 방법 작업의 순서

- 배경도 작성->상위 자료 흐름도 작성->하위 자료 흐름도 작성->자료 사전 작성->소단위 명세서 작성

마. 구성요소

- 프로세스 : 원이나 둥근 사각형. 이름을 해당 도형 안에 적음
- 자료 흐름 : 화살표. 화살표 위에는 자료의 이름을 씀
- 파일 또는 저장소 : 한쪽이 열린 직사각형. 이름도 안에
- 자료 출처와 도착지 : 직사각형 안에 이름을 기재

바. 최상위 자료 흐름도(배경도)

- 작업은 시스템 경계의 입출력 식별로부터 시작.
- 각 프로세스를 구체화해서 분할.

사. 프로세스

- : 동사구분을 사용하여 표시 및 단일 직접목적어 사용
- 최하위 프로세스는 차후 소단위 명세의 대상이 됨

아. 자료 흐름도 작성 원칙

- 1) 자료 보존의 원칙(Conservation Rule)
- 2) 최소 자료 입력의 원칙(Parsimony Rule)
- 3) 독립성의 원칙(Independence Rule)
- 4) 지속성의 원칙(Persistence Rule)
- 5) 순차 처리의 원칙(Ordering Rule)
- 6)영구성의 원칙(Permanence Rule)
- 7)자료 변환의 원칙(Nature of Change)
- 8)과도하지 않게 프로세스 세분화

2. 객체 지향 분석 방법

- 데이터와 데이터에 적용될 기능을 추상화하는 방법
- 시스템에 존재하는 객체를 먼저 찾고 객체 안에 어떤 자료와 오퍼레이션이 필요한지 알아야됨
- 객체 사이에 존재하는 여러 가지 관계 파악 중요

3. 요구 분석 명세서

가. 요구 분석서가 갖추어야할 사항

- 사용자 개발자 모두 쉽게 이해되어야함
- 기술된 조건은 합의된 것이어야함
- 제안된 시스템에 수행될 모든 기능을 정확히 기술하고 모든 제약 조건 명시
- 테스트 기준 필요.
- 품질 측정 방법 필요

나. 요구 분석 문제점

- 사용자의 부정확한 요구 표명
- 잦은 요구 변경

- 커뮤니케이션 장애

- 시스템 복잡도

다. 명세서의 평가

- 1)평가 기준 : 무결성과 완벽성/일관성/명확성/기능적/검증 가능성/추적 가능성 및 변경 용이성
- 2)평가 방법 : 검토 회의, 테스트 사례 작성, 프로토타입 도구 / CASE 도구

4 장. 정보 요구 검증

1 절. 정보 요구 상관분석 기법

: 도출된 정보 요구사항을 다른 영역(기능·프로세스·조직 등)과 비교·분석함으로써 정보 요구사항의 도출이 효과적으로 이루어졌는지 파악할 수 있다.

1. 주체별 분류

가. 요구사항 분석가 수행

- : 요구를 수집/분석한 주 담당자를 기준으로 검토 기준 항목을 마련후 분석 수행
- 정보 요구사항을 도출한 분석가에 의해 수행되므로 객관성 저하
- 도출 절차 및 업무팀과 의사소도 원활로 상관분석이 원활히 진행 가능
- 요구사항 분석가는 업무 이해도가 높아 정확한 분석 가능성이 높음

나. 품질보증 팀 수행

: 업무 이해도가 낮으나 상관분석 작업 특화로 전체적 인터페이스 검증에 용이

다. 외부 감리 수행

- 업무 파악 한계, 제 3자 시각 검토 가능. 객관성 강화. 잘못된 분석 결과 초래

2. 정보 요구/어플리케이션 상관분석

- 정보 요구사항에서 도출된 항목과 어플리케이션 영역에서 도출된 항목으로 매트릭스를 작성하여
- 각 crud 중 어떤 것이 사용되었는지 정의 (공백은 해당 없음으로)
- 정보 항목과 기본 프로세스가 모두 누락인 경우 분석이 불가능

3. 정보 요구/업무 기능 상관 분석

- : 정보 요구사항과 비즈니스 아키텍처에서 도출된 업무 기능을 비교하여 분석. 비즈니스에서 요구하는 항목은 모델링의 근간이 되므로 업무 기능별 필요 정보 항목의 누락 여부를 확인하는 건 매우 중요
- 가치 사슬 분석 등의 기법을 통해 도출된 최하위 수준의 전사 업무 기능을 도출하고 매트릭스 열에 배치
- 정보 요구사항에서 도출한 업무 기능은 행에 배치
- 생성, 수정, 삭제는 C(Create Or Change)로 하고 검색만 하는 경우 U(use)로 표기

2 절. 추가 및 삭제 정보 요구사항 도출

1. 정보 요구/어플리케이션 상관 분석

가. 어플리케이션 충족도 분석 매트릭스

- 정보 항목을 생성하는 기본 프로세스가 반드시 존재해야 함
- 상태를 종료하는 기본 프로세스도 반드시 존재해야 함

- 생성된 항목은 조회, 수정, 삭제 액션 중 하나 발생해야함
- 하나의 정보 항목의 프로세스는 7 개를 보통 넘지 않아야함

나. 매트릭스 분석

- 기본 프로세스가 CRUD 하는 정보 항목이 없는 경우, 정보 항목을 도출하고, 이것도 없으면 해당 프로세스 삭제
- 정보 항목에서 7 개 이상의 기본 프로세스가 사용될 경우, 정보 세분화 필요
- 정보 항목을 삭제하는 기본 프로세스가 없으면 도출해야하고, 삭제가 필요한지 확인
- 정보 항목이 생성만 되고 사용되는 곳이 없으면, 기본 프로세스 도출
- 조회만 하는 기본 프로세스는 기본 프로세스가 아니므로 해당 모듈 검토
- 기본 프로세스가 여러 액션을 수행하면 프로세스를 추가 분해

2. 정보 요구/업무 기능 상관분석

- 매트릭스 분석
- 각 정보 항목은 한 번 이상의 Create 를 가지는 지
- 모든 업무기능은 정보 항목과 연관이 있는지
- 생성된 정보 항목은 다른 업무에 의해 Updatte 가 되는 건지
- 연관성이 없으면 검토 필요

3. 정보 요구/조직 기능 상관분석

- 매트릭스 분석
- 모든 조직은 정보 항목과 연관이 있는가
- 위의 Create 랑 Update 동일
- 정보 항목의 활용도 파악으로 모델링 기법 적용
- 데이터 관리 주체선정에 주의를 기울여야함

3 절. 정보 요구사항 보완 및 확정

1. 정보 요구 보완

: 분석을 통해 파악된 추가 및 삭제 요구사항에 대해 담당자와 구체적인 미팅을 통해 정보 요구 목록 보완

2. 정보 요구 확정

: 보완된 사항에 따라 재차 재검토를 하며 추가 반영사항에 대해 최종 정보요구 목록에 대해 확정

Ⅲ. 데이터 표준화

1 장. 데이터 표준화 개요

1 절. 데이터 표준화 정의

1. 데이터 표준화가 필요한 이유

- 비표준화된 데이터로 인한 사용자간의 '의사소통 혼란, 부정확한 정보생산, 관리비용 증가 -> 신뢰할 수 있는 데이터 품질확보에 대한 인식 확산 -> 데이터 표준화 필요성 대두' 로 이어짐
 - 기업에서 업무 시스템(계정계)에서 분석시스템(정보계)을 구축하거나 여러 공공기관에서 개방되는 데이터를 모아서 활용하고자 할 경우 생성기준, 정보 항목, 데이터 유형 등이 제 각각이라 어려움을 겪는 상황이 발생
 - 한 예로 주차장 앱을 서비스하려는 경우, 해당기관의 관리목적과 관리수준에 따라 관리하는 정보항목, 명칭, 형식, 유효 값이 다를 수 있다.

2. 데이터 표준화 정의

: 동시다발적인 정보시스템 개발과 전사 데이터 표준 관리 도구의 부재, 그리고 전사 데이터 관리 마인드 및 관리 인력의 부재는 데이터 품질 저하를 야기시켰고 이는 곧 데이터의 활용 상의 문제점을 드러냄. **데이터 품질 향상을 위해 일관된 데이터 형식 및 규칙을 적용**

가. 데이터 활용 상의 문제점

- 데이터의 중복 및 조직, 업무, 시스템별 데이터 불일치 발생
- 데이터에 대한 의미 파악 지연으로 정보 제공의 적시성 결여
- 데이터 통합의 어려움
- 정보시스템 변경 및 유지 보수 곤란

나. 데이터 문제점의 원인

- 동시다발적인 정보시스템 개발
- 전사 데이터 관리 마인드
- 전사 데이터 관리 인력 부재
- 전사 데이터 표준 관리 도구 부재

다. 데이터 관리 개선 방안

- 데이터 표준화, 규격화를 위한 기본 방침 설정
- 전사적인 정보 공유를 위해 유지되어야 할 공통 데이터 요소 도출
- 전사적인 데이터 요소 등록 및 관리 체계 구축
- 정보시스템 개발 및 유지보수 시 승인된 데이터 요소를 활용해 효율성 및 공유성 향상

3. 데이터 표준화 기대 효과

- 명칭 통일로 인한 의사소통 증대
- 필요한 데이터의 소재파악에 소요되는 시간 및 노력 감소
- 일관된 데이터 형식 및 규칙의 적용으로 인한 데이터 품질 향상
- 정보 시스템 간 데이터 인터페이스 시 데이터 변환, 정제 비용 감소

2 절. 데이터 표준화 개념

1. 데이터 표준화 정의

- : 현실 세계의 정보를 컴퓨터로 관리하기 위해 데이터베이스에 저장하는 정보 항목의 종류, 명칭, 형식, 유효 값, 관리 절차 등을 특정 기준에 따라 표준을 만드는 일
- 데이터 정보 요소에 대한 명칭, 정의, 형식, 규칙에 대한 원칙을 수립하여 전사적으로 적용

가. 데이터 명칭(표준용어 · 표준단어)

1) 유일성

- 하나의 개념에 대해 사용자 모두가 통일된 용어를 사용할 수 있도록 하나의 명칭만 허용

2) 업무적 관점의 보편성

- 업무적 관점에서 보편적으로 인지 되어야 한다.

3) 의미 전달의 충분성

- 데이터 명칭 이름만으로 데이터의 의미 및 범위가 파악될 수 있어야 한다.

나. 데이터 정의(용어 정의서)

: 데이터가 의미하는 범위 및 자격 요건을 규정.

- 사용자 관점에서 잘 이해되게 명시. 데이터 소유자를 결정하는 기준이 됨
- 제 3 자 입장에서 기술함으로써 데이터 의미를 잘 이해하도록 하게 함
- 데이터 명칭을 그대로 쓰거나 약어 또는 전문 용어는 가급적 사용하지 말아야 함

다. 데이터 형식(표준 도메인)

- 데이터 타입 : Numeric, Text, Date, Char, Timestamp 등

- 데이터 길이 및 소수점 자리

※ 데이터 형식 정의 고려 사항

- 도메인을 정의하여 성격이 유사한 데이터 간의 형식을 통일화 해야됨
- 데이터의 최대값 또는 최대 길이가 고정되지 않은 경우 여유 있게 정의
- 특수 타입(CLOB, Long Raw 등)은 데이터 조회, 백업, 이행 등에 제약이 많으므로 비권장.

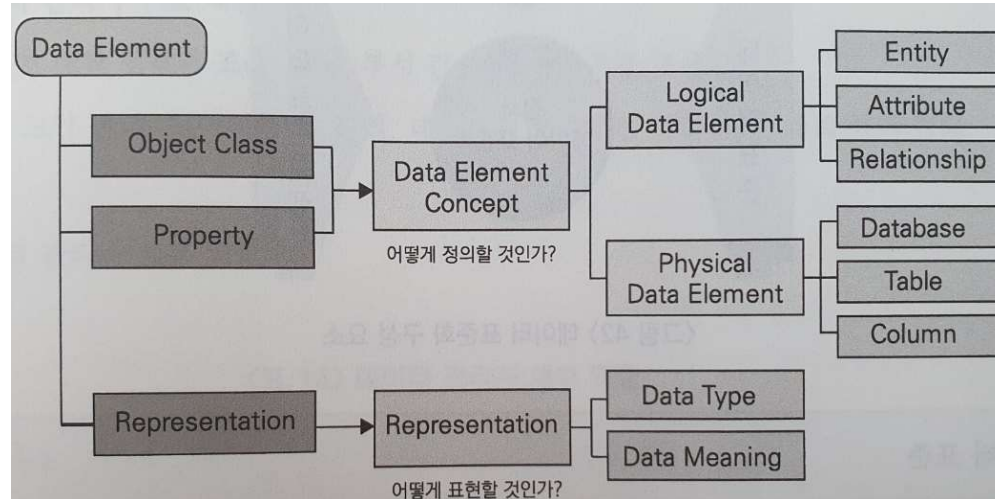
라. 데이터 규칙(업무 규칙)

: 발생 가능 데이터를 사전 정의해 입력 오류와 통제 위험 최소화(정합성/완전성 향상)

- 기본 값 : 입력 생략 시 자동으로 입력되는 값
- 허용 값 : 입력 가능한 데이터 값을 제한
- 허용 범위 : 입력 가능한 값을 범위로 제한

2. 데이터 표준화 대상

ISO/IEC 11179 : 데이터 구성요소



데이터 표준 사전의 구성 요소

- 1) 단어
- 2) 도메인
- 3) 용어

3. 데이터 표준화 구성 요소

: 데이터 표준, 데이터 표준 관리 조직, 데이터 표준화 절차

가. 데이터 표준

: 정의 할 수 있는 모든 객체를 대상으로 하는 것이 이상적이지만 그 대상이 너무 많을 때는 핵심 데이터(CTQ : Critical To Quality Data)를 선정해 주로 관리해야 할 것만 표준화 하는 것도 효율적

- 표준 용어 : 업무적 용어(보고서나 매뉴얼, 색인 등)와 기술적 용어(테이블명, 칼럼명 등)이 있음
- 표준 단어 : 동일한 개념을 의미하는 용어 생성 예방. 영문명이 작성 기준이됨.
- 표준 도메인 : 칼럼에 대한 성질을 그룹화한 개념. 데이터 타입 및 길이를 일관적으로 정의 가능
- 표준 코드 : 도메인의 한 유형으로서 특정 도메인이 가질 수 있는 값을 이미 정의해 해 둔 것
- 기타 데이터 표준 관련 요소 : 주제 영역, DB 스키마, Tablespace, INDEX 등의 표준 관리

나. 데이터 표준 관리 조직

: 데이터 관리자는 하나의 기업 또는 조직 내에서 데이터에 대한 정의, 체계화, 감독, 보안 업무 담당

1) 데이터 관리자 주요 역할

- 데이터에 대한 정책과 표준 정의, 부서 간 데이터 구조 조율
- 데이터 보안 관리, 데이터 모델 관리, 데이터의 효율적인 활용 방안 계획 등

2) 데이터 관리자 세부 역할

구분	주요 활동
전사 데이터 관리자	- 데이터 표준화에 대한 정책 결정 - 검토된 데이터 표준 제안에 대한 승인
업무 데이터 관리자 (Ownership)	- 담당 업무 기능의 데이터 요구 사항 반영을 위해 필요한 데이터 표준 정의 - 업무 관련 데이터 표준 변경 제안에 대한 합동 검토
업무 시스템 관리자 (Stewardship)	- 시스템 관리 목적의 데이터 요구 사항을 위해 필요한 데이터 표준 정의 - 업무 관련 데이터 표준 변경 제안에 대한 합동 검토 - 데이터 모델에 대한 데이터 표준 적용 및 준수 여부 체크

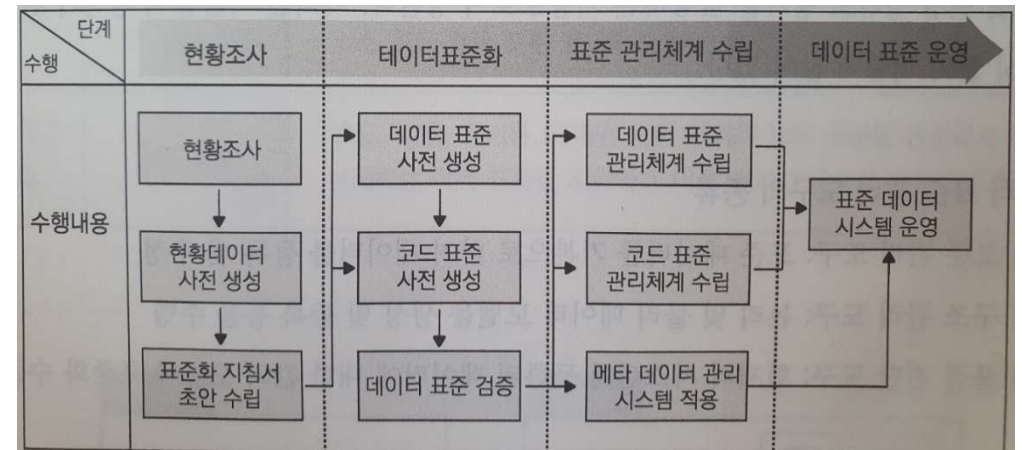
- Ownership : 데이터의 생성, 변경, 삭제 권한의 관리주체로 주로 해당 시스템을 사용하여 시스템 사용 목적으로 달성하고자 하는 부서나 담당자를 말한다.

- Stewardship : 데이터 오너십을 갖는 주체로부터 데이터에 대한 입력, 수정, 삭제 권한을 위임받은 정보화 담당자 또는 담당부서를 말한다.

3) 데이터베이스 관리자와 비교

구분	데이터 관리자(DA)	데이터 베이스 관리자(DBA)
관리 대상	데이터 요구 사항을 반영한 데이터 모델 및 각종 표준	데이터 모델을 특정 데이터베이스 제품 특성에 맞게 구축한 데이터베이스
주 업무	업무에 필요한 데이터의 메타 데이터를 정의하고 신규 또는 변경된 요구 사항을 신속하게 데이터 모델에 반영	요구되는 성능 수준을 발휘하면서 안정적으로 운영되도록 데이터베이스를 관리
품질 수준 확보	데이터 표준의 관리 및 적용을 통해 품질 수준을 확보	데이터의 정확성 관리를 통해 데이터 품질 수준을 확보
전문 기술	담당 업무 분야에 대한 업무 지식과 데이터 모델링에 대한 전문성 필요	데이터 모델에 대한 해독 능력 및 특정 데이터베이스 제품에 대한 전문 지식 필요

4. 데이터 표준화 절차



절차별 활동

구분	주요 활동
데이터 표준화 요구사항 수집	<ul style="list-style-type: none"> - 개별 시스템 데이터 표준 수집 - 데이터 표준화 요구 사항 수집 - 표준화 현황 진단
데이터 표준 정의	<ul style="list-style-type: none"> - 표준화 원칙 - 데이터 표준 정의 : 표준 용어, 표준 단어, 표준 도메인, 표준코드, 기타 표준
데이터 표준 확정	<ul style="list-style-type: none"> - 데이터 표준 검토 및 확정 - 데이터 표준 공표
데이터 표준 관리	<ul style="list-style-type: none"> - 데이터 표준 이행 - 데이터 표준 관리 절차 수립: 데이터 표준 적용, 변경, 준수 검사 절차.

3 절. 데이터 표준 관리 도구

: 수립된 전사 데이터 표준 정보의 관리, 데이터 표준에 의한 개발 및 유지 보수 지원, 데이터 표준 준수 및 변경 영향도 평가를 담당하는 기능으로 구성.

1. 데이터 표준 관리 도구의 종류

- 데이터 표준 관리 도구 : 표준 데이터를 기본으로 하여 데이터를 등록 및 수정
- 데이터 구조 관리 도구 : 논리 및 물리 데이터 모델을 생성 및 등록 등을 수행
- 데이터 품질 진단 도구 : 현재의 시스템에 등록된 데이터에 대한 값의 오류나 표준화 수준을 진단
- 데이터 흐름관리 도구 : 데이터페이스와 애플리케이션의 상호 영향도 분석을 수행

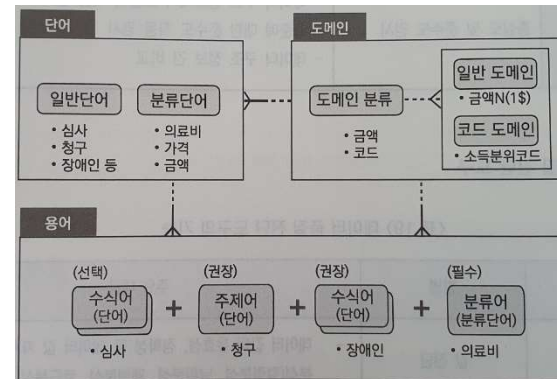
2. 데이터 표준 관리 시스템

가. 데이터 표준 관리 기능

- 보통 메타데이터 관리시스템이라고 하며, 이를 통해 단어, 용어, 도메인 코드를 등록 및 조회할 때 사용

기능	세부설명	주요 내용
단어 관리	전사 단어 사전 및 금칙어	<ul style="list-style-type: none"> - 전사 관점에서의 단어 사전 관리 - 금칙어의 사전 정의 및 관리
용어 관리	용어 사전	<ul style="list-style-type: none"> - 업무적으로 정의된 표준 용어에 대한 관리 - 기본 단어의 조합으로 업무 용어를 생성함
도메인 관리	도메인 사전	<ul style="list-style-type: none"> - 대표 및 그룹 속성에 대한 데이터 타입, 길이, 소수점 이하 길이 등을 사전에 정의한 도메인 관리
표준 코드 관리	전사 표준 코드	- 수집된 코드로부터 코드 통합 과정을 거쳐 전사 표준 코드를 도출한 후 관리
	코드 변환 매핑	- 소스 코드 값과 표준 코드와의 변환 매핑 관리
멀티 표준 관리	멀티 표준	- 코드, 칼럼, 테이블, 도메인 등에 대하여 멀티 표준을 관리해서 전사에 존재하는 여러 표준을 지원하고 이후 전사 표준으로 통합 되도록 함.

단어, 용어, 도메인, 코드 관계도



나. 데이터 구조 관리 기능

기능	세부설명	주요 내용
ER 모델 구조 관리	모델, 테이블의 구조 정보	<ul style="list-style-type: none"> - ER 모델 관리 - 리포지토리로부터 데이터 구조 정보를 추출 및 로드
DB 스키 마 관리	다양한 DBMS 지원 및 DB 스키마 로드	- 다양한 DBMS로부터 DB 카탈로그를 추출 및 로드
가변 속성 관리	모델의 사용자 속성을 자동 생성	- 모델 기본 속성 외에 설계 속성을 쉽게 추가
이력 관리	이력 관리	<ul style="list-style-type: none"> - 데이터 모델 변경 이력 - 형상 관리 지원
모델 비교 관리	충실도 및 준수도 검사	<ul style="list-style-type: none"> - 데이터 구조 정보에서 표준화 자동 검사 - 표준에 대한 준수도 자동 검사 - 데이터 구조 정보 간 비교

다. 데이터 품질 진단 도구

기능	방법	주요 내용
프로파일 링	값 진단	- 데이터 값의 유효성, 정확성 등 데이터 값 자체의 오류 분석 (컬럼 분석, 날짜분석, 패턴 분석, 코드 분석 등)
	구조 진단	- 논리적 데이터 구조의 오류로 인한 인리관성, 정합성 등을 확보하지 못하는 결함을 분석하고 진단 (표준화 수준, 테이블 구조, 정규화 수준, 컬럼 및 관계 정의 등)
업무 규칙(BR: Business Rule) 진단		- 법, 규정에 정의된 업무기준에 근거하여 데이터가 관리되고 있는지 진단하는 방법

3. 데이터 표준 관리 도구 도입 시 고려 사항

- 확장성 : DBMS의 정보 수집과 OLAP 툴 등의 다양한 데이터 구조 정보를 추출할 수 있는지 검토
- 유연성 : 표준을 전사적으로 한번에 적용하기 어려우므로 복수 표준 관리가 가능한지 검토
- 편의성 : 한글->영문명 자동 변환, 표준 검증 주기적인 작업 수행 기능, 메타 정보 수집 시 Import 수작업 최소화 등 사용자 편의성을 검토

2 장. 데이터 표준 수립

1 절. 데이터 표준화 원칙 정의

1. 데이터 표준화 요구사항 수집

: 현업 및 개발자로부터 표준 관련 요구 사항을 인터뷰 및 설문 조사로 자료 마련하되, 반드시 전체적인 관점 및 제 3자 관점에서 기술할 수 있어야함.

- 자신이 맡은 영역과 다른 영역 간의 인터페이스 부분에서 발생하는 불편사항 및 개선사항 파악
- 특정 영역에 대해 조사할 시 해당 영역에 익숙치 않은 관리자의 관점이 더 유용

2. 현행 데이터 표준 원칙 분석

가. 현행 데이터 표준 원칙 수집

- 현 정보시스템 개발 지침 문서 및 데이터 표준 확보
- 현행 정보시스템 모델 분석: 유추해 낼 수 있는 항목은 제한적임

나. 데이터 표준 원칙 사용 현황 분석

- 현행 시스템에서 적용하고 있는 항목을 도출

3. 데이터 표준 개선 방안 정의

- 문제점 및 개선 방안 도출 시 반드시 전사적 관점에서 접근하고 전사적 필요성을 검토하여 신규로 정의하거나 정의 대상에서 제외. 또한 기존 관리정책에 부합해야함

4. 데이터 표준 원칙 수립

가. 데이터 표준 기본 원칙 정의 : 전체적으로 정의할 기본 원칙 수립

나. 데이터 표준 지침 작성

1) 데이터 표준 지침 구성

- 개요: 목적 기술
- 데이터 표준화 관련자의 역할과 책임 : 일반적인 관련자로는 전사데이터 관리자, 데이터베이스 관리자, 모델러 등
- 데이터 표준 관리 절차 : 작업 프로세스를 규정하고 프로세스별 표준화 관련자들의 역할을 기술하고, 일반적인 관리 절차로는 데이터 표준 정의, 데이터 표준 변경, 데이터 표준 준수 프로세스가 있음
- 데이터 표준 기본 원칙 : 대상 모두에 대한 기본 원칙 기술
- 데이터 표준 대상별 명명 규칙
 - 사용 문자 : 알파벳, 한글, 특수문자, 숫자 등 허용 여부 및 사용 조건 규정
 - 영문 대소문자 : 알파벳 사용시 대소문자 규칙 규정
 - 한글명 영문명 동시 정의 여부 : DB 의 경우 대부분 영어 사용하므로 정의 필요
 - 명칭의 구조 : 테이블명 혹은 컬럼명을 어떻게 조합해 표현할 것인지
 - 명칭에 대한 허용 길이 : DB에 따른 길이 제약
 - 명칭 표준화에 대한 기준 : 유사한 개념의 단어/용어가 복수 개 존재하면 어떤 단어를 택할 것인지
 - 명칭에 대한 예 : 위의 명칭에 따른 3 가지 기준에 따른 샘플을 제시해 명시함

2) 데이터 표준 개발 지침 작성 시 유의 사항

- 다른 정보 시스템에 적용할 수 있도록 범용성을 고려해 데이터 표준 지침 정의
- 대상 DBMS 에 모두 적용하도록 고려, 영문 약어를 추가로 정의하거나 뷰 사용
- 표준 용어/도메인을 서로 다른 DBMS 에 어떻게 적용할 것인지 방안 고려 (매핑)

2 절. 데이터 표준 정의

1. 표준 단어 사전 정의

- 데이터 표준화의 출발은 단어를 정의하는 것으로 해당 기관이 사용하는 모든 단어를 추출한 후 종류와 유형을 분류하고 업무정의 및 용도를 고려하여 표준단어 정의
- 이음동이어, 동음이의어 처리에 주의
- 정의된 표준 단어는 원칙을 참고하여 영문명과 영문 약어명 정의

가. 표준 단어 사전

- 업무상 사용하며 일정한 의미를 갖는 최소 단위의 단어를 정의한 사전
- 정의된 단어는 궁극적으로 속성에 사용
- 이음동이어 사용시 논리적으로 사용하고, 약어도 정의하여 구분할 수 있도록 하면 좋음
- 표준화의 핵심은 동일한 의미의 속성명을 통일시키는 것

1) 표준 단어 관리 기준

- 표준성 : 일반적으로 업무에 사용하는 단어에서 추출, 너무 전문적인 용어는 최소화
- 일반성 : 일반적인 사전적 의미와 크게 다르지 않아 일반인도 이해 가능 해야함
- 대표성 : 비슷한 의미의 동의어들을 대표할 수 있어야함

2) 표준 단어 작성 형식

- 전사적으로 관리는 엔티티와 속성을 개별 단위로 추출해 동음이의어, 이음동이어를 정비한 후 논리명(한글명)을 기준으로 물리명(영문/영문 약어), 유사 용어까지 정리
- 개별 단어 외에도 동의어, 유의어, 반의어 등과 같은 단어 간의 구조도 함께 정의

나. 표준 단어 정의

현행 용어 수집 -> 단어 분할 -> 단어 정련 -> 표준 단어 사전 도출(정의)

1) 현행 용어 수집

- 기업내 모든 데이터 모델 또는 정의서에서 현행 용어에 대한 한글/영문명을 수집하며, 현행 용어를 통해 영문 약어명을 도출하려는 것이 목적이므로 한글명이 존재하지 않는 모델은 수집 대상에서 제외

2) 단어 분할

- 최소 단위의 단어로 분할하며, 분할시 한글명을 비롯해 영문명도 같이 분할하는가를 고려하고 단독으로 분할되는 숫자는 표준화 대상 단어에서 제외

3) 단어 정련

- 분할하여 취합된 단어 중 의미가 동일한 단어에 대해 대표 단어를 표준으로 선정하고 영문 약어명을 선택하며 최종적으로 한글명과 영문명이 모두 유일해야함
- 한글명이 동일한 단어와 의미가 동일한 단어 취합후 활용 빈도가 가장 높은 한글명을 표준으로 채택
- 약어는 영어 약어를 선택

- 4) 표준 단어 사전 정의 : 표준으로 선택한 단어의 한글명, 영문명 등록. 의미가 유사한 것은 유사어로 나머지 이음동의어들은 금칙어로 등록하고, 향후 적절한 표준 단어의 검색을 위해 관련 표준 용어를 같이 기술

다. 표준 단어 정의 시 고려사항

- 표준 단어의 단위는 최소 단위 기준 및 사용 빈도가 높은 단어 조합들을 표준 단어로 정의 -> 유리
- 접두어, 접미어같이 한 글자로 된 것은 가급적 표준에서 배제하는 대신 앞뒤에 나오는 단어와 조합하여 표준 단어로 정의하는 것이 바람직

2. 표준 도메인 사전 정의

가. 표준 도메인 사전 : 표준 도메인 사전은 논리/물리적으로 유사한 데이터를 그룹화하여 유형과 길이를 정의한 것. 도메인은 여러 개의 하위 도메인으로 구성되거나 하나의 도메인이 여러 개의 도메인에 중복적으로 사용될 수 있음

- 표준화 수행 시 중요한 요소는 데이터 타입과 길이, 포맷 등이 같은 값의 집합
- 하나의 속성에는 허용된 유효한 값의 형태가 같아야 하므로 도메인은 하나만 사용

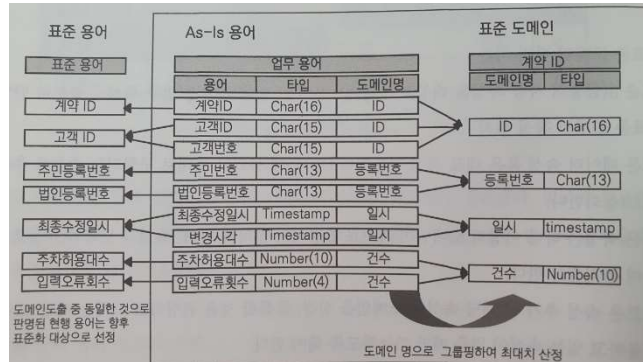
1) 표준 도메인 관리 기준

- 표준성(공통적 사용 속성을 대상으로 정의), 유일성, 업무 지향성(업무 특성을 충분히 반영)

2) 표준 도메인 작성 형식

- 모든 데이터 속성 혹은 대표 속성 가운데에 DBMS에 동일한 형태로 구현되는 속성을 추출하여 그룹화
- 모든 속성은 하나 이상의 도메인이 복수로 할당되면 안되고, 속성과 도메인은 상호 매핑하여 관리
- 새로운 속성 추가 시 해당 속성의 도메인 선정, 등록할 것을 권장하고 삭제는 해당 도메인을 사용하고 있는 속성이 없을 때만 가능하도록 해야 함.

나. 표준 도메인 정의 : 존재되어 사용되는 컬럼명, 데이터 타입, 길이 등을 정리해 표준 도메인 정립



1) 현행 용어 정보 분석

- 기업에 존재하는 시스템의 모델, 정의서를 이용해 정보 수집 후 물리적으로 유사한 유형의 용어를 그룹화
- 동일한 정보시스템에 대한 데이터 모델에서 추출된 현행 용어들을 유일하게 추출
- 한글명, 영문명, 데이터 타입 및 길이 모두 일치하는지 확인
- 데이터 타입 및 길이가 동일한 용어들을 검색해 유사한 속성의 용어들을 그룹핑

2) 표준 도메인 정의

- 가급적이면 업무적으로 의미가 있는 도메인명을 부여
- 기존 데이터와 호환성 및 범용성을 위해 그룹화된 용어들 중 가장 큰 데이터 길이를 표준으로 택함
- 다. 표준 도메인 정의 시 고려사항
- 현실적으로 어느 도메인도 속하지 않는 칼럼이 있을 수 있어 모든 용어를 포괄하는 도메인은 필요 없다.
- 표준 도메인에 정의할 형식을 각기 다른 DBMS에 어떻게 적용할지 고려
- 동일한 용어로 판명된 현행 용어들을 별도로 기록, 향후 동일한 데이터 표준 용어로 택할 때 참고

3. 표준 코드 사전 정의

- 현 코드를 바탕으로 통합 필요성에 따라 통합 대상을 파악하고 표준 코드를 정의하고 현 코드와 매핑 설계함. 정의된 표준 코드 별로 오너십을 정의하여 향후 해당 코드에 대한 수정, 삭제에 대한 권한 관리

가. 표준 코드 사전

1) 표준 코드 관리 기준

- 재사용성 : 표준화 기구, 정부 등에서 정의한 코드 재사용이 더 효과적
- 일관성 : 업무 범위 내에서 가능한 한 유일하게 정의
- 정보 분석성 : 가능한 범위의 데이터는 모두 코드화해 관리

2) 표준 코드 작성 형식

- 전사적으로 사용하는 코드를 추출해 부여된 코드와 동일인지 확인하고 동일한 값을 가지는 것은 통합작업을 통해 단일화 작업을 수행. 코드는 표준화 팀의 엄격한 관리를 받아야함. 도메인 값이 명확한 경우(예: Y/N)은 코드화하지 않아도 된다.

나. 표준 코드 정의

1) 현행 코드 수집

- 단독(개별) 코드 테이블 : 고객번호, 사원번호와 같이 하나의 코드를 한 테이블에서 관리하는 형태. 이런 형태로 관리하는 코드는 대부분 필수적으로 가지고 있어야 할 코드 번호, 코드 값 외에 부가 정보들을 관리할 경우가 많으므로, 이런 형태로 관리되는 데이터가 코드인지, 정보성 데이터인지를 명확히 구분한 뒤 추출 여부를 판단
- 통합 코드 테이블 : 직원 유형, 고객 유형 등 복수 개의 코드를 하나의 테이블에서 관리하는 형태로 공통으로 관리하는 코드는 모두 수집
- 어플리케이션 관리 코드: DB에 수집하지 않고 애플리케이션에서 관리하는 형태는 사용자 인터페이스를 조회하거나 해당 코드 정보를 SQL 등으로 컬럼 데이터를 추출하는 간접적인 방법 사용

※ 코드 정부 누락을 방지하기 위한 방법

- 코드 데이터값 수집 : 테이블, 통합 코드 테이블, 어플리케이션 사용자 인터페이스를 통해 수집
- 코드성 칼럼 파악 : 해당 컬럼이 코드를 저장하는 컬럼인지 참조 데이터를 저장하는 컬럼인지 구분
- 수집된 코드에 대한 사용처 파악 : 어떤 코드를 저장하는지 파악으로 누락된 코드 확인

2) 현행 코드 상세 분석

- 코드 값이 일치하는 동일한 코드 인스턴스를 가지는 코드를 찾은 뒤 해당 코드의 모든 코드 인스턴스를 확인하고 비교함으로써 통합 가능한 코드 식별
- 분석할 코드가 너무 많은 경우 사용하는 업무 기능별로 코드를 분류하고 분류된 단위로 코드 분석

3) 표준 코드 정의

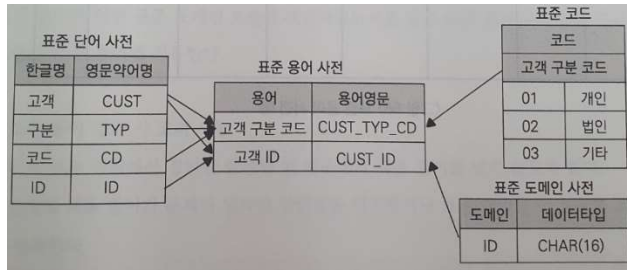
- 통합 대상이 없는 코드는 현행 코드 인스턴스를 그대로 유지
- 통합 대상이 존재하고 통합 대상 코드의 코드 번호가 서로 상이할 경우 새로운 코드 부여
- 다. 표준 코드 활용
- 향후 모든 정보 시스템은 표준 코드를 사용해야함
- 모든 코드를 사용하지 않고 일부 코드 값만을 사용할 경우 표준 코드부터 파생된 코드를 사용
- 이 경우 표준 코드로부터 파생된 코드도 반드시 표준 코드에 정의되어 있어야 하며, 파생 코드에 코드 인스턴스를 추가해야할 경우 표준 코드에 먼저 정의해야 한다.

라. 표준 코드 정의 시 고려 사항

- 향후 확장성을 고려해 저으이해야 하며 여러 업무에서 사용하도록 통합된 코드로서의 일관성 유지
- 시스템 운영 중 변경되는 경우 해당 코드 사용한 기존 데이터를 위해 삭제하지 말고 중지 상태로 관리하고 새로운 코드 값을 신규로 정의
- 표준 코드를 도출하여 파악한 표준 코드 : 현행 코드 간의 변환 매칭 정보를 별도로 기록하여 향후 신규 정보시스템으로의 이행 시 참고함.

4. 표준 용어 사전 정의

- 단어, 도메인, 코드 표준이 정의되면 이를 바탕으로 표준 용어를 구성하고, 단어의 조합, 도메인 분류, 데이터 타입 길이, 코드 값 등을 기준으로 표준 적용이 무리 없는지 검토한다.



가. 표준 용어 사전

- 용어는 업무에서 자주 사용되는 단어의 조합을 의미하며 전사적으로 사용하는 엔터티와 속성을 대상으로 하여 표준 단어 사전에 정의된 단어를 조합하여 정의함.

1) 표준 용어 관리 기준

표준성 : 동일한 의미를 서로 다른 용어로 사용하는 경우가 많음. 표준화를 통해 전사 차원의 혼란을 막을 수 있음

일반성 : 지나치게 업무 관점에서 정의하면 안 됨

업무 지향성 : 기업의 업무 범위 내에서 약어나 별도의 정의해서 사용 가능 (남용 주의)

2) 표준 용어 작성 형식

- 용어 사전은 엔터티 용어사전과 속성 용어사전을 구분해 정의 및 관리. 논리명(한글), 물리명(영어)를 각각 가지며, 용어 범위 및 자격 형식 등이 설명되어야 함

나. 표준 용어 정의

1) 현행 용어에 대한 표준 단어 및 표준 용어 정의

- 현행 용어로부터 도출은 단어 수준에서의 표준화를 통해 이루어진다. 현행 용어를 분할하여 구성 단어 도출 후 해당 단어와 유사 단어를 검색한다. 유사 단어 중 표준 단어를 찾아낸 뒤 각각의 현행 구성 단어에 대한 표준 단어를 조합하면 표준 용어 도출

2) 표준 단어에 대한 도메인/코드 정의

- 현행 용어가 어떤 표준 도메인 도출에 관련되었는지를 찾고 표준 도메인을 위의 과정에서 도출된 표준 용어에 적용한다.

다. 표준 용어 정의 시 고려사항

- 데이터 표준 원칙에서 정의된 허용 길이를 넘지 않게 함
- 영문명 허용 길이가 문제가 되면 한글명을 변경하거나 표준 단어들 중 일부를 조합해 새로 등록
- 생성된 표준 용어가 너무 길면 두 개의 표준 용어를 복합하여 생성하는 방법도 고려

5. 표준화 원칙의 예

- 특정 한 날짜를 의미할 때는 '일자'를 사용 (예) 입금일자
- '시분초'까지 의미할 때는 '일시'를 사용 (예) 방문일시
- 년, 월, 일 중 일부분을 의미할 때는 '년', '년월', '월', '월일', '일' 등으로 사용
- 가격, 좌수, 단가, 잔액 등 관행적으로 사용하는 단어를 제외하고 금전을 의미할 때는 '금액' 사용
- 비율을 의미할 때는 '율' 사용
- 표준화 시스템에서 지원하는 기능 중에 핵심은 속성을 등록해 관리하는 것이며, 엔터티에 어떤 속성이 존재하며 특정 속성이 어떤 엔터티에서 사용됐는지를 보여주는 기능이 표준화 시스템의 핵심

3 절. 데이터 표준 확정

1. 데이터 표준 확정

가. 데이터 표준 검토 계획 수립

- 데이터 표준에 대한 검증 기준 : 유일성, 완전성, 정확성, 범용성

나. 데이터 표준 검토

- 검토 기준 및 대상 산출물을 준비하고 검토 참여 대상자에게 배포
- 담당자별 역할을 주지
- 진행자는 이슈에 대해 결론 도출을 위한 토론이 발생하지 않게 이슈 목록을 정리하고 일정 내 마칠 수 있게 함
- 검토 세션이 종료되면 세션별로 그 결과를 정리
- 검토 결과가 정리되면 데이터 표준 대상별로 보완 사항을 작성

다. 데이터 표준 보완 및 승인

- 보완 결과에 대한 확인 준비를 함. 검토 결과, 보완 목록, 보완 사항이 반영된 데이터 표준을 준비하고 배포
- 보완 목록에 준하여 데이터 표준 반영 여부를 확인한다. 미반영 사유가 있을 시 타당성 검토
- 보완 사항이 모델에 모두 반영된 것을 확인하고 전사 데이터 관리자의 승인을 얻는다.

2. 데이터 표준 공표

: 확정된 데이터 표준을 배포하여 전사 시스템에 적용할 수 있도록 하며 관련 내역에 대한 교육 수행

가. 데이터 표준 배포

- 검토가 종료되고 전사 데이터 관리자의 승인을 득한 데이터 표준은 데이터 표준 관리 도구에 등록하여 전사의 모든 사용자가 데이터 표준을 조회할 수 있도록 조치하고, 정보시스템 개발 관련자들이 데이터 표준을 준수하여 개발할 것을 공지

나. 데이터 표준 교육

- 데이터 표준에 대한 이해 및 효과적인 적용을 위해 사용자 및 운영자에 대한 교육 훈련 계획을 수립하고, 데이터 표준 지침 및 기타 데이터 표준 관련 교육 고재를 작성하고 교육을 수행한다.

3 장. 데이터 표준 관리

1 절. 데이터 표준 관리

1. 데이터 표준 관리 개요

: 표준화 작업절차 이후 수립된 표준에 근거하여 관리 프로세스를 정립하여 데이터 표준이 관리되게 함

2. 데이터 표준 관리 프로세스 유형

- 개발/운영 과정에서 데이터 표준의 신규 요건이 발생하는 경우 이를 처리하기 위한 프로세스
- 표준이 변경/삭제될 경우 영향도를 분석할 수 있는 절차와 이를 처리하기 위한 프로세스
- 표준을 준수하는지 체크하는 프로세스

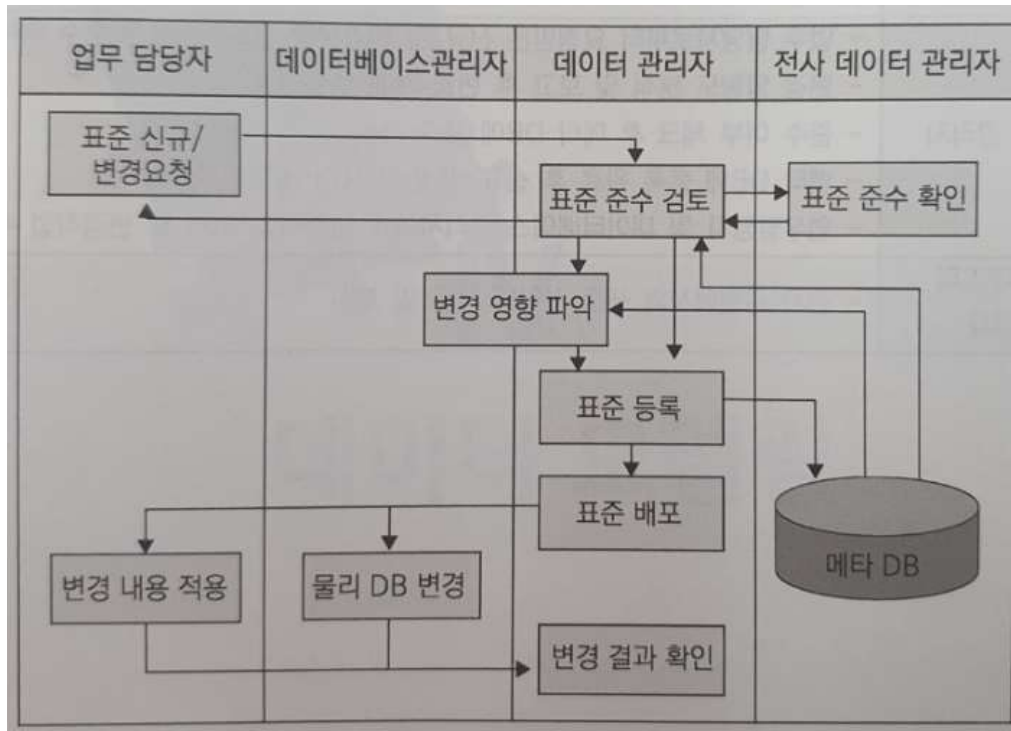
2 절. 데이터 표준 관리 프로세스

1. 데이터 표준 관리 프로세스 구성 요소

: 프로세스, 태스크, 역할과 담당 업무가 명확하게 정의되어야 함.

2. 구성 요소별 설명

가. 프로세스



나. 태스크

- 표준 신규/변경 요청 : 업무 담당자는 데이터 관리자에서 신규/변경을 요청
- 표준 준수 검토 : 요청된 사항에 대해 표준 준수 여부를 검토하고 업무 담당자에게 피드백을 제공

다. 역할과 담당 업무

역할	담당업무
업무 담당자	<ul style="list-style-type: none"> - 표준 신규 및 변경 요청 - 데이터 관리자로부터 지시받은 변경 내용 적용
데이터베이스 관리자	<ul style="list-style-type: none"> - 데이터 관리자로부터 변경 표준 사항에 대한 변경 영향 파악 협조 및 평가서 작성 - 데이터 관리자로부터 지시받은 변경 내용 적용 - 테스트 및 검증 - 사용자 반영 결과 통보
데이터 관리자	<ul style="list-style-type: none"> - 업무 담당자로부터 요청받은 신규 및 변경사항 검토 및 표준 준수 여부 체크 - 변경 영향도 분석 및 보고 후 변경 계획 수립 - 준수 여부 체크 후 메타 DB 에 표준 등록 - 메타 DB 에 등록 완료 후 신규 변경 표준 배포 - 업무담당자 및 데이터베이스 관리자에게 변경작업 지시 후 변경작업 수행결과 확인
전사 데이터 관리자	<ul style="list-style-type: none"> - 전사 관점에서의 표준 가이드 지문 및 제시

IV. 데이터 모델링

1 장. 데이터 모델링 이해

1 절. 데이터 모델링 개요

1. 데이터 모델링 탄생 배경

: 초창기 정보 시스템은 데이터 저장 매체가 없었으며, 배치 프로그램 위주의 정보 시스템.

- 정보 기술 발전로 배치 위주의 정보 시스템은 한계가 있었고 데이터 중심의 관리 기법이 발전
- 따라서 데이터를 어떻게 하면 중복 없이 정확하게 유지/관리에 대한 고민으로 모델링 기법 등장
- 데이터 모델링이란 인간이 결정할 대부분의 사항을 정의하는 시스템 설계의 전 과정

2. 모델 정의

: 어떤 대상을 의미하는 포괄적 의미가 있으며, 특히 데이터 모델은 현실 세계에 대한 우리가 관심 있는 대상을 데이터베이스화하기 위한 개념적 도구

3. 모델링이 필요한 이유

: 프로세스 모델링과 데이터 모델링 이 2 가지는 매우 중요

- 설계를 너무 프로세스 위주로 하면 업무 변화에 대응이 어려워 데이터 모델링에 중점적으로 함.
- 데이터 모델링의 효과
 - : 시스템을 구축하는 많은 이해관계자 간 훌륭한 의사소통의 도구가 될 수 있음
 - : 데이터 품질의 향상에 도움

4. 데이터 모델링의 필요성

- 여러 개의 많은 애플들을 하나로 묶어 포털격 앱을 구축때 반드시 데이터 통합이 이루어져야함.
 - 개발자들의 시스템 이해에 도움

5. 데이터 모델링 시 주의할 점

- 중복(Duplication) : 같은 데이터를 이용한다는 사람, 시간, 장소를 파악하는 데 도움을 줌으로써 DB의 여러 장소에 같은 정보를 저장하는 잘못을 막아줌
- 비유연성(Inflexibility) : 데이터 정의를 사용 프로세스와 분리함으로써 데이터 혹은 프로세스의 변화에 유연하게 대처할 수 있음
- 비일관성(Inconsistency) : 데이터 간 상호연관 관계에 대한 명확한 정의는 이러한 위험을 방지

6. 데이터 모델링 단계

가. 개념 데이터 모델링

- 사용자의 요구 사항을 찾고 핵심 엔터티와 그들 간의 관계 발견 및 ERD 작성
- 개념 모델은 추상적이므로 상위의 문제를 쉽게 구조화하며 사용자와 개발자가 논의할 기반 마련
- 추상화 수준이 높고, 업무 중심적이고 포괄적인 수준의 모델링 진행, 전사적 데이터 모델링, EA 수립 시 많이 이용

나. 논리 데이터 모델링

- 비즈니스 정보의 논리적인 구조와 규칙을 명확히 표현하는 기법 또는 과정
- 데이터 모델링의 최종 완료 단계 모델로 모델링 과정 중 제일 핵심 단계
- 가장 중요한 활동으로 정규화, 식별자 지정, M:M 관계 해소, 참조 무결성 작업 실시
- 재사용성이 높음

다. 물리 데이터 모델링

- 목표하는 DBMS의 특성 등을 고려한 스키마를 만들고 칼럼, 데이터 타입, 크기 정의
- 실제로 DB에 이식할 수 있도록 성능, 저장 등 물리적인 성격을 고려해 설계

7. 모델링 기본 원칙

- 1) 커뮤니케이션 원칙 : 최종 사용자 및 이해 당사자들에게 시스템의 지향점을 분명하게 설명하려고
- 2) 모델링 상세화 원칙 : 데이터의 상세화 정도를 제시하고 조직이 사용하는 정보 구조의 '최소 공통 분모'를 제시. 데이터는 데이터의 본질과 잠재적 사용을 이해할 수 있는 만큼 상세화되어야 함
- 3) 논리적 표현 원칙 : 데이터에 대한 논리적 측면을 최대한 표현

8. 좋은 데이터 모델의 요소

- 완전성 : 업무의 필요한 모든 데이터가 모델에 정의됨
- 중복 배제 : 동일한 사실이 데이터에 한번만 있어야 함. 데이터 중복 시 일관성에 문제
- 비즈니스 룰 : 업무 규칙을 애플리케이션 레이어에서도 구현이 가능하지만, 먼저 데이터 모델에서 구현할 수 있다면 구현하는 게 맞음
- 데이터 재사용 : 데이터가 애플리케이션에 독립적이어야 좋음 -> 재사용성 향상
- 안전성 및 확장성 : 안정성뿐만 아닌 추후 데이터가 쉽게 변경되도록 설계
- 간결성 : 테이블 수가 많다고 절대 좋은 건 아님. 통합하여 모델의 간결해야 좋은 거
- 의사소통 : 요구사항에서 도출되는 많은 업무규칙을 데이터 모델의 엔터티, 서비타입, 속성, 관계 등의 형태로 자세히 표현되어 의사소통의 도구로 역할됨
- 통합성 : 전체가 한 몸처럼 연결되어 움직여야 한다.

2 절. 데이터 모델링 기법 이해

1. 데이터 모델 목적

- 설계에 대한 계획이자 청사진이 곧 데이터 모델의 목적이다. 이는 구축 시스템을 이해하고 원활한 의사소통 도모하게 해주며 초기에 오류 발견 시 많은 비용과 시간을 절약할 수 있게 됨

2. 개체 - 관계 모델 기법

: 데이터에 대해 관리자, 사용자, 개발자들이 서로 다르게 인식하고 있는 뷰들을 하나로 통일하고 단일화된 설계안을 만들 수 있고 서로 다른 뷰를 충족시킬 수 있는 데이터 처리와 제약 조건 등의 요구사항을 정의할 수 있음. ERD는 '객체, 관계, 속성'으로 표현

3. 개체 관계 모델 요소

- : 일반적으로 논리/물리 모델링으로 나누어짐. 논리 모델은 업무 수행 방식이 바뀌어도 설계 변경이 거의 발생하지 않음
- : 논리 데이터 모델에서의 하나의 엔터티는 반드시 물리적으로 하나의 테이블이나 세그먼트가 되지는 않는다.

가. 엔터티

- : 초기 모델 설계 시 엔터티를 먼저 도출
- : 키 엔터티를 먼저 도출 하고, 메인, 액션 엔터티 도출

나. 속성

: 엔터티에 저장되는 개체 집합의 특성을 설명하는 항목. 일반적으로 서로 다른 집합에 정의된 속성은 같은 도메인을 공유할 수 있음 (예. 주소- 전화번호)

다. 식별자

: 개체를 식별할 수 있는 속성. 하나 또는 그 이상의 속성으로 이루어짐.: (물리관점에서 키

- 본질 식별자
- 후보 식별자
- 대체(보조) 식별자
- 인조 식별자 : 기본 식별자를 쓸 수 없을 때 사용
- 실질 식별자 : 공식적으로 인스턴스를 식별하기 위한 의미상의 식별자

라. 관계

: 엔터티 간의 연관성을 표현하는 것으로 엔터티의 정의, 속성 정의, 관계 정의에 의해 다양하게 변함

- 관계 연결
- 일대일 ,
- 일대다 : X에 속하는 한 개체는 Y와 하나만 연결, Y에 속하는 개체는 X의 여러 개와 연결
- 다대다

마. 카디널리티

: 관계에 참여하는 하나의 개체에 대해 몇 개의 개체가 참여하는지를 나타냄.

(n,m)으로 나타내면 최소 n개 최대 M개의 범위를 가질 수 있음을 의미

바. 존재 종속

: 한 엔터티의 존재가 다른 엔터티의 존재에 영향을 받으면 이를 존재 종속이라 함.

사. 서브타입

: 전체 집합인 슈퍼타입의 부분 집합이며, 배타적(1개만 관련)/ 포괄적(1개 이상 관련)으로 구분
: 슈퍼 타입은 공통적인 모든 속성을 포함하고 있고 서브타입은 적절한 속성만 포함됨

3 절. 데이터 모델링 표기법 이해

1. 바커 표기법(Barker Notation) - (DB 를 모델링할때 사용)

가. 엔터티 : 반드시 두 개 이상의 속성을 가져야 함

나. 관계

1) 엔터티와 엔터티 간의 관계

1:1, 1:M, M:N 등

2) 엔터티와 엔터티 간 상관 관계의 조건

필수 조건 : 실선 표시, 상대 엔터티에 조건에 만족하는 엔터티가 반드시 존재할 경우 표시

선택 조건 : 점선 표시, 상대 엔터티에 존재할 수도 있고 없을 수도 있을 때 표시

다. 식별자

: 엔터티를 대표할 수 있는 속성. 하나의 엔터티에는 반드시 하나의 식별자가 존재한다.

1) 식별자의 유형

- 본질 식별자 : 집합의 본질을 명확하게 설명하는 의미상 주어 즉 고유한 성질을 가지면 됨
- 후보 식별자 : 기본키가 될 수 있는 식별자들. 인스턴스를 유일하게 식별할 수 있는 속성 하나 또는 속성들의 조합. 여기서 기본키가 선택됨.
- 대체(보조) 식별자 : 원래의 식별자를 대신할 수 있는 식별자. 즉, 후보 식별자에서 기본키로 선택되지 않은 남은 식별자들.
- 인조 식별자 : 기존의 본질 식별자를 사용하기 어려울때 임의의 값을 가진 속성을 만들어 사용하는 것. (예시로 주민번호가 너무 길다 판단되면 사원번호를 쓴다던가 00)
- 실질 식별자 : 인스턴스를 식별하기 위해 공식적으로 부여된 식별자 (본질이나 인조 식별자)

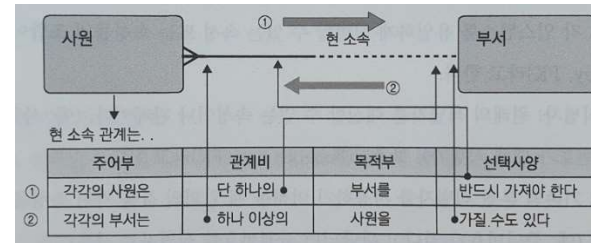
2) 작성 방법

- 엔터티 : 둥근 사각형 안에 엔터티 명과 속성명을 넣음
- 각 속성 식별자 종류에 따라 기호를 다르게 표현

라. 서브타입

: 슈퍼 타입 안에 상자로 표시하고, 중복을 허용하지 않으며 상호 배타적 관계

마. 관계의 표현 비교



2. I/E 표기법

가. 엔터티

: 큰 사각형으로 표현하고, 식별자를 위의 사각형에 배치, 나머지(속성)를 아래 사각형에 배치

나. 관계

ER win의 부호 사용	Identifying	Non-Identifying	의미
타원, 해쉬 마크 및 까마귀 발	⌞— — ⌞	⌞----- ⌞	0,1 또는 그 이상의 개체 허용
까마귀 발이 있는 해쉬 마크	⌞— — ⌞	⌞----- ⌞	1 또는 그 이상의 개체 허용
해쉬 마크가 있는 타원	⌞— — ⌞	⌞----- ⌞	0 또는 1 개체 허용
해쉬 마크만 있음	⌞— — ⌞	⌞----- ⌞	정확히 1 개체 허용

다. 서브타입

: 배타적/ 포괄적일 수 있고, 배타적일 경우 관계에 X 표시

4 절. 데이터 모델링 사례

1. 엔터티 후보 목록 도출

- 현업자료와 현황평가 및 요구사항 정리
- 엔티비티 산출물 검토

2. 엔터티 정의

- 다른 엔터티와 명확한 구분 가능하며, 기본키에 의해 식별 가능
- 속성 존재
- 데이터로서의 중요성과 효용성

3. 엔터티 간 관계 정의

- 업무상 연관관계를 갖는 의미 있는 관계 부여
- M:N 관계는 두 개의 1:N 관계로 대체

2 장. 개념 데이터 모델링

1 절. 개념 데이터 모델링 이해

1. 개념 데이터 모델 정의

: 골격을 세워 놓는 형태. 주제별로 분류 가능한 업무를 분석해서, 핵심 엔터티를 추출하고 그들 간의 관계를 정의하며 데이터의 골격을 생성하는 것.

2 절. 주제 영역 정의

1. 주제 영역 개념

: 주제 영역은 기업이 사용하는 데이터의 최상위 집합. 하나의 주제 영역 내의 데이터 관계는 밀접해야하고, 다른 주제 영역에 포함되는 데이터 간의 상호작용은 최소화할 수 있도록 정의해야함.

- 계획 수립 단계는 하향식 분석을 원칙으로 하고, 검증을 위해서 부분적으로 상향식 분석을 사용
- 데이터를 하향식으로 분석하기 위한 개념으로 유용한 것이 주제 영역.
- 주제영역을 분해하면 하위 수준의 주제 영역 또는 엔터티가 나타남

2. 주제 영역 분류 원칙 및 기준

가. 주제 영역 분류 원칙

- : 동일 기능을 하는 자원이 중복되지 않도록 데이터 중복 최소화
- : 또한 데이터의 확장성이 보장되어야 하며, 데이터 관련성 및 편의성이 확보되어야 함.

나. 주제 영역 명명

- 실 업무에서 보편적으로 사용하는 업무 용어 부여
- 유일한 단수형 명사 사용
- 데이터 그룹을 의미하는 이름 부여

다. 주제 영역 정의 절차

: 데이터 현황 분석, DA 원칙 및 방향성 분석, 선진 모델 분석 -> 데이터 분류 지침 정의 -> 주제 영역 분류 및 정의 -> 주제 영역 도식화

라. 주제 영역 분류 방법

- 1 차 분류 : 주요 데이터 집합의 유형 정의. 기존 시스템별 데이터의 성격 및 특성을 고려하여 영역 분류. 예) 데이터를 발생시키는 주제, 데이터 발생 주제 간의 상호작용으로 발생하는 대상 등

- 2 차 분류 : Biz 활동에 필요한 분석 주제와 현황 등의 영역으로 분류. (1 차 분류를 세분화)

(예시 : 관계자 기본, 관계자 상세 등)

- 3 차 분류 : 2 차 영역의 분류를 더욱 세분화

(예시 : 관계자 - 고객, 법인, 조직, 직원 등)

○ 업무 활동을 의미하는 이름을 배제하고 데이터 그룹을 의미하는 이름을 사용하도록 함.

3. 주제 영역 활용

: 주제 영역은 데이터의 계층 구조를 파악하는데 도움. 품질 확보에도 기여. 또한 효율적 데이터 관리를 위한 기준을 제공. 데이터 구성 및 통합에 대한 방향도 제시. 주제 영역은 기업의 전사 업무를 위한 전체 구성에 대한 청사진 제공

4. 주제 영역 정의 내용 도출

- 업무에 사용하는 데이터 명사형 도출
- 업무 기능 이름으로 도출
- 하향식 : 주제 영역에서 출발해 엔터타입으로 전개
- 상향식 : 엔터티 타입을 그룹화하여 주제 영역 도출
- 분석 단계에서의 도출 : 아키텍처 모델을 정련하는 과정에서 도출, 데이터 모델 상세하에서 도출

3 절. 후보 엔터티 선정

1. 후보 엔터티 선정

2. 엔터티 후보 수집

3. 엔터티 후보 식별

4. 엔터티 후보 선정 시 유의 사항

- 엔터티 가능성 있으면 일단 검토 대상
- 후보 자격 여부만

5. 엔터티 분류

- 키 엔터티
- 메인 엔터티 : 업무 중심에 해당되는 것들.
- 액션 엔터티

4 절. 핵심 엔터티 정의

1. 엔터티 정의의 요건

가. 엔터티 정의

- 관리하고자 하는 것인지 확인
- 대상 개체 간의 동질성이 있는지 확인
- 다른 개체와 확연히 구분되는 독립성을 가지는지
- 순수한 개체이거나 행위를 하는 행위 집합인지

다. 엔터티 파악 요령

- 현업 담당자와의 인터뷰 활용
- 기존 시스템 있으면 기존 시스템 산출물 검토
- DFD를 통해 업무 분석을 진행하였으면 DFD의 Data Store 활용

라. 엔터티 작성 예

- 1) 업무 시나리오 작성
- 2) 엔터티 정의서 작성
- 3) 엔터티 특징
- 4) 엔터티 분류

* 유무형에 따른 분류

- 유형 엔터티
- 개념 엔터티
- 사건 엔터티

* 발생 시점에 따른 분류

- 기본 엔터티
- 중심 엔터티
- 행위 엔터티

2. 의미상의 주어 정의(본질 식별자)

: 정확한 식별자를 부여해야 의미가 명확해짐

3. 코드성 키 엔터티 모델링

: 코드성 엔터티는 잘 도출해야됨 (어려움)

- 자식이 있는지

- 다양한 속성을 가질 건지

- 여러 엔터티에 관계를 맺을 건지

등에 따라 본질 식별자를 결정한다.

4. 집합 순수성

가. 집합 순수성의 의미

: 순수한 본질 집합이 되어야 한다는 것은 사람, 상품 등과 같이 단위 사물을 정의한 개체 집합이 되든지, 임금, 계약 등과 같은 행위 집합이 되든지 간에 둘 중의 어느 하나가 되어야 한다는 것임. 서로 결합된 형태면 관계에 해당된다.

나. 집합 순수성

예시로 납입자와 같은 단어는 순수 본질 집합인 고객과 납입이 결합된 릴레이션

다. 집합 순수성 적용 예외 사항

1) 관계의 엔터티화

: 릴레이션이 M:M 관계라면 더 이상 관계로만 존재할 수 없어 엔터티로 바꿈. 이를 릴레이션 엔터티, 제휴 엔터티, 교차 엔터티라 부름 (중간 엔터티라 보면 됨)

2) 일부 집합 정의

- 전체 지법 중 일부의 집합만 엔터티로 정의하고자 할 때

- 금융 기관은 기관으로써의 엔터티이고 수납기관은 관계일 뿐이다.

3) 배타적 관계 대체

- 여러 엔터티와 동일한 내용의 관계를 갖는 배타적 관계를 맺을 때 만약 배타적 관계의 변화 가능성이 높다면 별도의 엔터티 구성

5. 집합 동질성

: 집합이 들어갈 개체들의 동일한 성질을 어디까지 한정할 것인가

6. 엔터티 명칭

: 함축적인 의미로 설명하지 않아도 되어야하고 오해도 최소화 해야함

7. 서브타입

가. 서브타입 지정 의미

- 구체적인 부분 집합의 종류(서브타입) 명시

- 개체 관계 도표를 입체적이고 구체적으로 작성하기 위해서 집합의 부분 집합을 표현해야함

- 다른 추가적인 속성에 해당

- 서브타입에서 상호배타적인 다수의 부류(카테고리)가 형성될 수 있음

나. 고려사항

1) 교집합 허용 불가

2) 서브타입의 합이 전체 집합

3) 서브타입 표현 기준

- 개별 속성을 가질 때, 개별 관계를 가질 때, 가독성을 위해

다. 서브타입 도출

1) 분류 속성 : 엔터티의 정보가 차별되는 경우

2) 다수의 선택적 속성

3) 선택적 관계가 존재하는 경우 : 분할함으로써 관계가 필수적으로 변하는지 확인

라. 활용

1) 데이터 모델에 업무 규칙을 명확히 표현해 업무를 정확히 이해하고, 속성 및 관계의 선택성 제거

2) 서브타입 표현은 업무 규칙의 명확성과 표현의 복합성의 트레이드 오프 관계로 적절한 조화 필요

마. 서브타입의 이해

- 서브타입은 인스턴스들의 집합인 엔터티라는 전체 집합에서 일부의 인스턴스들만 모아놓은 부분집합

- 즉 하나의 엔터티의 여러 개의 서브타입이 있을 수 있음

바. 서브타입 명

: 그 서브타입의 집합이 구체적으로 또 어떤 집합인지 나타낼 수 있는 용어 사용

사. 서브타입 세트

: 또 2 개 이상의 서브타입을 그룹화하여 서브타입 세트 구성 가능

아. 서브타입 세트명

: 서브타입을 도출하기 위해 사용된 기준이 되는 속성

: 다만 서브타입을 표현하기 위한 속성이 반드시 물리적으로 있을 필요 없음 (조건에 따라 구분한다던지) - 즉 논리적으로 구분해도 됨

자. 서브타입 표현의 비교

: DataWare DA /ERwin 툴들의 표현이 조금씩 다름

DataWare DA 는 서브타입을 엔터티 안쪽에 표현하고 ERwin 은 외부에 엔터티 처럼 표현

ERwin 이 복잡해질 여지가 있음 주의.

8. 엔터티 통합과 분할

가. 엔터티의 독립성

: 새롭게 정의하고자 하는 엔터티가 앞서 정의해 둔 어떤 엔터티에도 포함되지 않는 독립적인 결합인지 확인해야함.

나. 엔터티 분할/통합

: 집합의 일부가 겹쳐 있을 때 모양 결정

- 어느 한 집합을 확장해서 나머지를 포함시키거나

- 교차된 부분을 어느 한쪽만 가지게 분리

- 교차되는 부분이 있더라도 별개로 간주하고 관계를 맺기

라. 유연성 향상을 위한 통합

: 동질성 확보해 중첩된 집합을 포함시키면 이들 간의 집합적 관계는 명확해짐. 유연성과 단순성을 향상할 수 있지만 지나친 확장은 집합의 의미를 희석시킴

- 키 엔터티는 최대한 통합되어 집합이 늘어나도 거의 영향이 없어 통합 시도에 적합

5 절. 관계 정의

1. 관계 정의

- 개체 간의 관계 또는 속성 간의 관계

- 일대일, 일대다, 다대다

2. 관계 이해

가. 관계도 집합에 해당됨

나. 직접 관계를 관계라 함 : 직접 종속인 것만 관계로 본다

다. 두 엔터티 간에는 하나 이상의 관계가 존재할 수 있음

라. 외래키로 정의

마. 관계의 관점

- 항상 두 엔터티 간에 존재, 항상 두 개의 관점을 가짐

- 데이터의 양방향 업무 규칙을 표현, 관계를 통해 정보로서의 활용 가치 상승

- 관계는 외래키로 구현되어 참조 무결성(RI, Referential Integrity)로 데이터 정합성 유지.

3. 관계 표현

가. 관계 형태(Degree/Cardinality)

1) 하나 이상 : 1:M 관계는 1:1 관계가 포함되어 있다는 것

2) 단 하나(Only One)

나. 선택 사양(Optionality) : 점선(선택)은 존재하지 않을 수 있지만, 직선(필수)은 반드시 존재

다. 두 엔터티 간 연관성 및 관계의 연관도

1) UNARY 관계 : 엔터티가 자기 자신과 관계를 맺는 것. (자기 참조 관계) (1:1, 1:M 가능)

2) BINARY 관계 : 일반적인 두 개의 엔터티가 서로 관계를 맺는 상황을 의미

라. 관계명의 부여

1) 두 개의 관계 멤버십에 각각 부여 : 각자 상대방 입장에서 기술

2) 현업에서 사용하는 간결한 동사형으로 표현

- 두 엔터티 타입 간의 업무적 연관성을 나타내는 이름 부여

- 현재 시제 사용, 다른 관계명과의 유일성은 확보되지 못함, 능동/수동형 배제

3) 업무적 의미가 없거나 애매모호한 용어는 배제

4) 관계명은 관계명 부여의 중심이 되는 엔터티에서 보았을 때 관계의 시계방향에 위치

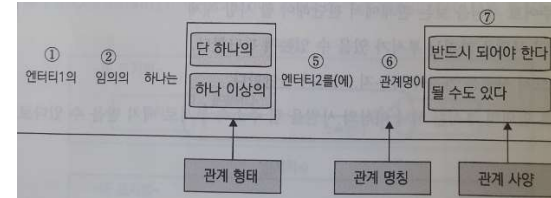
- 보통 관계하면 양방향으로 생각할 수 있으나 의미가 더욱 명확한 관계명을 선택하여 한쪽 방향의 관계명만을 나타냄

마. 관계 수

즉 카디널리티를 의미 -> (0, M)로, 0~M 으로 관계 될수 있다

4. 관계 정의 방법

가. 관계 구문 이해



나. 관계 정의 절차

- 각 양쪽 마다 중심으로 관계를 판단

5. 관계의 페어링(Pairing)

- 엔터티 안에 인스턴스가 개별적으로 관계를 가지는 것

- 페어링의 집합 -> 관계

6. 관계의 표기법

- 관계명, 관계차수(Cardinality), 관계 선택사양 표시.

7. 관계 형태

가. 1:1

나. 1:M

다. M:N

라. 다중 관계

+ 병렬식 관계

: 두 개의 엔터티 사이에 서로 다른 두 개 이상의 관계가 존재하는 경우

- 하나의 행에서 관리되므로 새로운 테이블 추가할 필요 없음

- 인덱스 수 증가 및 SQL 복잡

- 새로운 관계 추가. 관계 형태의 변경 등에 매우 취약

+ 직렬식 관계

: 두 엔터티 간의 관계가 일정한 순서로 연속적으로 연결되어 있는 관계. 의존성이 있는 관계

- 예를 들어 구매 프로세스 : 주문 - 결제 - 배송 - 수령

- 새로운 엔터티 추가 필요

- 관계가 행 형태로 나타냄

- 인덱스 수 감소 및 SQL 이 단순해짐

- 새로운 관계 및 추가 등에 유연함

- 관계 내용별 상세 정보 관리 가능(자식 엔티티를 거느릴 수 있음)

마. 특수한 관계형태

1) 순환 관계

- 하나의 순환 엔터티는 각 엔터티의 모든 속성을 포함해야함.
- 순환 모델은 직선(필수 관계)로 존재 할 수 없음(무한 루프), 반드시 선택 관계

2) BOM(Bill of Materials) 관계 : 네트워크 구조

3) Arc(Mutally Exclusive) 관계 : 어떤 엔터티가 두 개 이상의 다른 엔터티의 합집합과 릴레이션을 갖음

6 절. 관계 데이터 모델

1. 관계 데이터 모델 개념

- 개념적 구조를 논리적 구조로 표현하는 논리적 데이터 모델
- 하나의 개체에 대한 데이터를 하나의 릴레이션에 저장

2. 관계 데이터 모델 기본 용어

- 릴레이션 : 하나의 개체에 대한 데이터를 2 차원 테이블의 구조로 저장. 파일에 대응됨
- 속성(Attribute) : 릴레이션의 열, 속성. 필드에 대응
- 튜플(Tuple) : 릴레이션의; 행에 해당. 레코드에 대응
- 도메인 : 하나의 속성이 가질 수 있는 모든 값의 집합
- 널 : 속성 값을 아직 모르거나 해당되는 값이 없음
- 차수 : 하나의 릴레이션에서 속성의 전체 개수
- 카디널리티 : 하나의 릴레이션에서 튜플의 전체 개수에 해당

3. 릴레이션의 구성

- 릴레이션 스키마 (릴레이션 내포)
- 릴레이션 인스턴스 : 레코드의 집합 (릴레이션 외포)

4. 릴레이션의 특징

- 튜플의 유일성
- 튜플의 무순서
- 속성의 무순서
- 속성의 원자성

5. 데이터베이스의 구성

- DB 스키마
- DB 인스턴스

6. 키(Key)

: 릴레이션에서 튜플들을 유일하게 구별하는 속성 또는 속성들의 집합

가. 키의 특성

: 유일성, 최소성

나. 키의 종류

- 슈퍼키
- 후보키, 기본키, 외래키
- 대체키 : 후보키중 선택되지 못한 나머지 키

7. 관계 데이터 모델의 제약

가. 무결성 제약조건(integrity constraint)

: 데이터의 무결성을 보장하고 일관된 상태로 유지하기 위한 규칙

- 무결성 : 데이터의 결함이 없는 상태. 즉 정확하고 유효하게 유지하는 것

나. 개체 무결성 제약 조건

기본키를 구성하는 **모든 속성**은 널을 가질 수 없다.

다. 참조 무결성 제약조건

외래키는 참조할 수 없는 값을 가질 수 없다. (즉 기본키에 없는 값을 가질 수 없다)

하지만 외래키의 값은 널이 될 수 있다. (외래키가 기본키가 아닌 경우)

3 장. 논리 데이터 모델링

1 절. 논리 데이터 모델링 이해

1. 논리 데이터 모델링 정의

- 데이터베이스 설계 프로세스의 기초 설계 단계로써 **비즈니스 정보의 구조와 규칙**을 명확히 표현할 수 있는 기법
- ER 다이어그램으로 개념적 구조를 데이터베이스에 저장할 형태로 표현한 논리적 구조
- 물리적 스키마 설계 전 단계의 데이터 모델 상태
- 보통 데이터 타입과 제약조건을 여기서 정함

2. 논리 데이터 모델의 구성요소

- 엔터티
- 관계(연관도 및 관계수)
- 슈퍼타입 및 서브타입
- 속성 및 도메인
- 주키, 후보키, 대체키, 외래키
- 업무 규칙

3. 논리 데이터 모델링 이해

- 논리 데이터 모델링은 개념 모델링에서 정의한 핵심 엔터티와 관계를 바탕으로 속성을 정의하고 식별자를 확정하는 과정
- 논리에서는 정규화를 통해 새로운 엔터티가 생성되거나 새로운 관계 생성
- M:M 관계가 해소되어 새로운 행위 엔터티가 생성되는 과정
- 비즈니스 관점이 아닌 비즈니스 및 관리에 필요한 데이터 관점에서 모델링. 즉 시스템 관점으로도 속성에 반영

가. 정규화 과정

- DBMS 의 데이터 보존성을 높이고 중복성을 줄여 저장공간의 최소화를 꾀하고 조회속도 등을 향상 시키기 위한 일련의 과정
- 1~5 단계 까지 있지만 보통 3 정규화 혹은 BCNF 정규화 까지 수행
- 정규화 과정을 통해 도출된 엔터티의 속성들의 기본 데이터 타입 지정 (문자, 숫자, 날짜)
- 논리 데이터 모델링에서는 DBMS 유형 고려하지 않음

나. 정규화 설계의 이점

- 기억장소 크기의 최소화
- 데이터 불일치 위험의 최소화
- 수정 및 삭제로 인한 이상 가능성 최소화
- 데이터 구조의 안정성 최대화

다. 제 1 정규형

- 속성 원자 값 원칙 및 중복성 제거 원칙 및 PK 설정 원칙 준수
- 중복되는 속성은 분류해야함.

라. 제 2 정규형

- 부분적 함수 종속성 제거 원칙 준수
- 즉, 키의 일부 속성 하나가 다른 속성에만 관련있는 경우, 다른 키의 속성은 그 다른 속성과 관련이 없는 경우 문제.
- 키의 속성은 다른 속성들과 관련이 있어야한다는 것

마. 제 3 정규형

- 이행적 함수의 종속성 제거
- 기본적으로 엔터티내 모든 속성은 기본키에 의존성을 가져야 한다는 것.
- 만약 특정 속성이 기본키가 아닌 속성에 의존하는 경우 문제.

바. 제 3.5 정규형 (BCNF 정규형, Boyce-Codd)

- 모든 결정자가 후보키여야 한다는 제약 조건
- 즉 기본키의 구성 (특히 복합키)는 모두 후보키의 조합으로 이루어져야 있어야 한다는 것.

4. 논리 데이터 모델링 목적 및 효과

- 해당 비즈니스에 대한 데이터 관점에서의 명확한 이해를 도움
- 전사적인 통합 데이터 체계를 확립
- 데이터의 일관성 및 정확성 유지를 위한 규칙 도출
- 안정적인 데이터베이스 설계의 토대를 마침
- 사용자와의 명확한 의사소통을 위한 수단으로 활용 가능

5. 논리 데이터 모델링의 주요 성공 요인

- 업무의 능통한 현업 사용자와 함께 모델링 진행
- 절차보다는 데이터에 초점을 두고 모델링 진행
- 데이터의 구조와 무결성을 함께 고려
- 개념화, 정규화 기법 채택으로 데이터 무결성 확보
- 가능하면 다이어그램으로 업무를 표현
- 데이터 모델링을 지원하는 데이터 사전 구축

2 절. 속성 정의

1. 속성 개념

- 가. 속성 정의 : 더 이상 분리될 수 없는 최소의 데이터 보관 단위
- 릴레이션의 열, 어트리뷰트
- 속성도 엔터티와 같이 업무 내용, 다양한 문서들을 통하여 도출

나. 속성 특징

- 1) 속성의 어원적 의미 : 가공되지 않은 것, 원천적인 것, 독자적인 성질이 있는 것
- 2) 속성도 일종의 집합
- 3) 릴레이션도 속성이다.
- 4) 속성들 간은 서로 독립적이다. (속성들은 반드시 식별자에 직접 종속되어야 함, 2 정규형)
- 5) 업무에 필요한 정보
- 6) 한 개의 속성값만 가지며, 다중값일 경우 별도의 엔터티를 이용하여 분리

다. 속성의 분류

* 특성에 따른 분류

- 기본 속성 : 업무로부터 추출한 값
- 설계 속성 : 규칙화를 위해 변형/새로정의한 값
- 파생 속성 : 다른 속성에 영향을 받아 발생한 값

* 엔터티 구성방식에 따른 분류

- PK, -FK, - 일반 속성
- 단순형, 복합형

라. 엔터티, 인스턴스, 속성, 속성값의 관계

* IE 표기법은 1 관계에 직선을 그음

* Barker 표기법은 1 관계에 직선을 굵지 않음

마. 업무 규칙

: 속성이 가질 수 있는 값을 통제하여 논리적 데이터 모델의 무결성을 유지하기 위한 명세이다

- 키 업무 규칙 : 엔터티의 주키와 외래키의 값을 정확히 유지하기 위한 업무 규칙.
- 속성 업무 규칙 : 엔터티 속성이 갖는 값의 타입, 범위, 특성 등을 정의
- 연쇄 반응 : 입력, 수정, 삭제, 조회 작업의 정당성과 이러한 작업이 다른 엔터티 또는 동일 엔터티의 속성값에 미치는 영향 규정

2. 속성 후보 도출

가. 속성 후보 수집처

- 구 시스템 문서 자료, 보고서, 협의, DFD, DD 등 참조하여 수집

나. 속성 후보 선정 원칙

- 1) 원시 속서으로 보이는 후보는 버리지 않음
- 2) 소그룹별로 후보 군을 만들고 가장 근접한 엔터티에 할당

다. 속성의 기본 구성요소

- 1) 속성명 : 함축성 있는 명사 혹은 명사구 사용, 필요시 표준 약어 생성 후 적용. 단 하나만.
- 2) 도메인 : 업무적 제약 조건으로 파악된 일련의 특성
- 3) 선택성 : 모든 건의 해당 속성이 반드시 값을 가져야 하는지 여부

3. 속성 검증 및 확정

가. 최소 단위

1) 검증 방법

- 집합 개념 속성은 단순 개념으로 분할
- 가능한 최소 단위로 분할 후 관리 필요에 따라 통합
- 일자, 시간, 성명, 주변, 우편번호는 분할안하는게 좋음

2) 분할 속성의 대표적 유형

- 가) 일자 형태의 속성 : 매출 일자 -> 통합된 것이 속성으로 봐야함
- 나) 외부에서 공인된 속성 : 우편번호
- 다) 전화번호 유형 :전화, 팩스
- 라) 주소 유형 : 고객 주소 -> 지역 주소 + 상세주소

나. 하나의 값만을 가지는 지 검증

1) 검증 방법

- 여러 값을 가지거나 반복되는 속성은 잘못된 속성
- 반복되는 속성은 새로운 엔터티로 분할해야 1 차 정규화의 대상이 됨

2) 대표적 유형

- 계약일 -> 고객은 여러 계약일을 가질 수 있으므로 등.

다. 추출 속성(Derived Attribute)

라. 상세 관리

: 현재뿐만 아닌 미래까지도 고려

4. 가공 속성 규칙

- 초기엔 중복 값이므로 표현하지 말라고 했으나 현재는 기업의 관리자들의 관심 항목이므로 모델 내에 반드시 기술을 권장
- 도출 속성이란 하나 이상의 속성의 값을 누적함으로써 선택적으로 주제를 도출하는 속성에 대해 값을 창출기 위해 추가 계산 작업을 수행함으로써 창출되는 속성
- 계산 속성은 엔터티의 단값가레에 대한 어떤 특성을 기술하며, 일반적으로 관련 속성의 또 다른 단일 사례로부터 계산
- 도출 속성은 사용자들의 데이터 요구 사항을 나타냄
- 도출 속성은 향후 과정에서 기본키로서의 역할을 맡아서는 안 된다.

5. 속성 정의 시 유의 사항

가. 명확한 속성 명칭 사용

예를 들어 학점 -> A,B 를 말하는지 정의

나. 유일한 복합명사 사용

다. 단수형으로 속성명을 사용함

라. 표준 단어 제정

: 기준 준수 용어 및 일관성

마. 자의적인 전용 금지

6. 속성 업무 규칙 정의

3 절. 엔터티 상세화

1. 식별자 확정

가. 엔터티 구체화 4 단계

1) 식별자 확정

: 논리적 의미의 식별자(본질 식별자)를 기준으로 관계가 형성되었다면. 이 단계에서는 실질적 식별자 생성

2) 정규화 단계

3) M:M 관계 해소 단계

: 개념 데이터 모델에서 핵심 엔터티들 간의 M:M 관계가 해소되면서 교차 엔터티가 생성되는 단계

4) 참조 무결성 정의 단계

나. 본질 식별자

: 키 엔터티는 부모가 없이 창조된 집합이므로 식별자 또한 창조해야 하고 행위 엔터티는 항상 부모가 누구인지를 확인하는 방식으로 진행

1) 키 엔터티의 본질 식별자

: 사원 엔터티를 예로 들면 사원번호는 인조 식별자이고 주민등록번호가 본질 식별자임

2) 절대 종속/상대 종속 의미

: 나를 태어나게 하는 데 절대적인 영향을 주었는지 아닌지를 따지는 것

- 어떤 엔터티에 부모 엔터티가 반드시 필요해야 존재하면 절대 종속, 아니면 상대 종속

3) 직접 종속/간접 종속 의미

: 부모 엔터티와의 관계가 1 층이면 직접이고 아니면 간접

4) 행위 엔터티의 본질 식별자 : 절대 종속이면서 직접 종속인 것을 찾는 것이며, 결국 자신을 태어나게 한 근본을 찾는 것으로 육하원칙을 이용해 찾을

다. 후보 식별자 도출

: 하나의 엔터티에는 식별자로 사용할 수 있는 하나 이상의 식별자가 있으며, 후보 식별자들은 다음과 같은 조건을 만족해야 함

1) 각 인스턴스를 유일하게 식별할 수 있어야 함

2) 나머지 속성들을 직접 식별할 수 있어야 하며, Null 이 될 수 없음

3) 후보 식별자로 속성 WLQKGKQ 를 선택하는 경우에는 당연히 개념적으로 유일해야 함

4) 후보 식별자의 데이터는 자주 변경되지 않는 것이어야 한다.

라. 대체(보조 식별자)

: 식별자를 대신 할 수 있는 속성들이나 관계를 말함

마. 인조 식별자 지정

1) 최대한 범용적인 값을 사용

2) 유일한 값을 만들기 위한 인조 식별자 사용

3) 하나의 인조 식별자 속성으로 대체할 수 없는 형태를 주의

4) 편의성/단순성 확보를 위한 인조 식별자 사용 가능

5) 의미의 체계화를 위한 인조 식별자 사용 가능

6) 내부적으로만 사용하는 인조 식별자

바. 식별자 확정

: 자기 자신에 대한 존재 가치뿐 아니라 남들에 대한 배려를 어떻게 조화시키느냐가 중요하다

1) uid Bar 의 두가지 의미

가) 식별자로서의 역할 : 구별될 수 있도록 유일한 값을 만드는 데 일조

나) 정보로서의 역할 : 참조하는 엔터티 입장에서 상속받았기 때문에 자신의 정보도 증가했다 의미

2) UID 상속과 단절의 의미

: 실질적인 처리의 단순화를 가져다줄 수 있음

3) 식별자 확정 절차 : 하향식 방식이 좋음

2. 정규화

가. 정규화의 의미

: 변경 이상이 발생하면 데이터의 일관성과 무결성을 해칠 수 있음

→ 입력 이상, 삭제 이상, 갱신 이상

나. 정규화의 장점

1) 중복 제거

2) Null 값이 줄어듦

3) 복잡한 코드로 데이터 모델을 보완할 필요 없음

4) 새로운 요구 사항의 발견 과정을 도움

5) 업무 규칙의 정밀한 포착을 보증

6) 데이터 구조의 안정성 최대화

다. 정규화 단계 (생략)

3. M:M 관계 해소

가. M:M 관계의 의미

- 아직 모델이 덜 완성된 모습이며 최종적인 단계에서는 존재할 수 없는 형태

- 실제 업무 및, 엔터티들 도 실제로는 M:M 관계임

나. M:M 관계 해소의 의미

- 새로운 릴레이션 엔터티를 추가해 M:1 관계로 변경

- 연관 실체 엔터티는 M:M 관계 미결 시 관과해 버렸을 추가 업무 규칙, 논리를 내포할 수 있음

- M:M 관계는 데이터 종속성에 대한 결정을 어렵게 하고, 논리적 완성과 부분 집합 식별 능력을 제한

4. 참조 무결성 규칙 정의

- 관계 테이블의 모든 외부 식별자 값은 관련 있는 관계 테이블의 주 식별자 값이 존재해야 함.

- PK 와 마찬가지로 FK 도 데이터 무결성에 대한 업무 규칙을 내포하고 있음

- DB 설계 관점에서 선택하지 말고, 사용자의 업무 공칙에 따라 적절한 규칙 선택

가. 입력 규칙

1) Dependent: 대응되는 부모가 있는 경우에만 자식에 입력 허용

2) Automatic : 자식에 입력을 허용하고 대응 부모가 없으면 자동 생성

3) Nullify : 부모가 없으면 Null 처리

4) Default : 부모가 없으면 FK 를 지정된 기본값으로 처리

5) Customized : 특별한 검증 조건을 제시하여 입력을 적용

6) No Effect : 조건 없이 자유로인 자식 입력 허용

나. 삭제 규칙

1) Restirct : 대응되는 자식이 없는 경우에만 부모 삭제 가능

2) Cascade : 자식도 같이 삭제

3) Nullify : 부모 삭제 시 자식을 Null 로 바꿈

4) Default : 부모 삭제시 자식을 기본값으로 바꿈

5) Customized : 특별한 검증 조건이 만족하는 경우에만 부모 삭제 가능

6) No Effect : 조건 없이 자유롭게 부모 삭제 허용

3 절. 이력 관리 정의

1. 이력 관리 정의

- 이력 데이터의 관리는 모든 데이터를 대상으로 하지 않는다.
- 실제 이력이 발생하는 데이터 중에서도 꼭 필요한 데이터에 한정해서 이력 관리를 수행한다.
- 중요한 사항은 이력 관리의 형태로 시점이력과 선분이력 중에 적합한 형태를 결정하는 것이 중요

2. 이력 관리 대상 선정

가. 사용자 조사

- 변경 내역을 감시할 필요가 있는가
- 시간의 경과에 따라 데이터 또는 관계가 변할 수 있는가
- 과거의 데이터를 조회할 필요가 있는가
- 과거 버전을 보관할 필요가 있는가?

나. 이력 데이터의 종류

1) 발생 이력(Occurrence) 데이터

- 발생할 때마다 이력 정보를 남기는 것으로 이벤트 발생 시 남기는 것과 날마다 남기는 것이 있음

2) 변경 이력(Modification) 데이터

- 변경될 때마다 전/후 차이를 확인해야 한다면 변경 이력을 남긴다.

3) 진행 이력(Progress) 데이터

- 진행에 따라 남기는 것으로 대표적인 예로 주문과 같은 업무가 있다.

다. 이력 관리 형태

- 1) 시점 이력 : 변경이 발생한 시각만 관리, 특정한 시점의 데이터 추출할 경우 불필요한 작업 수행이 생김 (특정 시각)
- 2) 선분 이력 : 변경 시작부터 종료 시점까지 관리 (범위)

라. 선분 이력 관리 유형

1) 인스턴스 레벨 이력 관리

- : 하나의 인스턴스에 변경이 발생하면 전체 인스턴스를 새롭게 생성함
- 한번의 액세스로 해당 시점의 모든 데이터 참조 가능, 로그성 데이터가 목적일 때 좋음
- 다른 이력 관리 유형에 비해 저장에 쉬우나 저장 공간의 낭비가 발생
- 가장 큰 단점은 하나 이상의 컬럼에 변경이 있으면 이벤트가 모호해짐. 만약 이벤트가 자식 정보를 가지면 해당 이벤트를 찾기가 매우 어려움
- 실제 변경 데이터를 찾기 위해서는 과거 데이터와 비교해야함.
- 특정 순간의 스냅샷만 보는 게 아니면 처리가 복잡해짐
- 변화가 빈번하게 발생하는 상황에 고려해볼 수 있는 유형

2) 속성 레벨 이력 관리

- : 대상 속성에 변화가 생길 때만 이력을 생성하는 방식
- 실제 어떤 데이터가 변경되었는지가 명확함. 하나의 이력 관리 엔터티에서 다른 엔터티와 통합 이력 관리 가능
- 독립적 처리 가능, 변화는 자주 발생하지 않고 이력 관리 대상은 많은 경우 사용
- 특정 속성들에 변화가 집중되는 경우 유용. 여러 속성에 대한 이력 필요시 많은 Merge 발생
- 다른 유형에 비해 액세스 쿼리에서 조건 검색이 조금 어려움. 변화가 너무 많은 경우 곤란

3) 주제 레벨 이력 관리

: 유사하거나 연동될 확률이 높은 것별로 레벨 이력 관리

- 인스턴스, 속성 레벨의 장점을 합친 것
- 목적이 분명한 엔터티를 생성함으로써 확장성을 확보할 수 있는 용도로 사용할 수 있다.
- 변경 부분만 처리 가능(독립적), 다른 엔터티와 통합 이력 관리 가능, 속성 레벨의 단점 해소 가능
- 전체를 참조할 때 인스턴스 레벨에 비해 Merge가 발생하는 문제가 있음
- 부분에 따라 변경 정도의 차이가 심한 경우 유리

3. 선분 이력 관리용 식별자 확정

가. 선분 이력에서 식별자 결정 시 고려 사항

- 성능 문제 고려, 실제 데이터는 Unique 하지만 의미적으로 Unique 하지 않는 일이 발생
- Unique 여부를 검증할 수 있는 조치를 병행해야 함.

나. 선분 이력에서 종료점 처리 시 주의 사항

1) 종료점이 미정이므로 NULL

: 논리적으로 타당하지만 비교가 불가능하며, 인덱스를 사용하지 못하므로 수행 속도가 저하됨

2) 수렴하므로 최대치 부여

: 아직 종료되지 않았으므로 무한히 계속되는 것으로 간주. 최대치 부여 및 가능한 Table 속성 Default 제약 조건을 부여하며, 수행 속도에 유리

4 장. 물리 데이터 모델링

1 절. 물리 데이터 모델링 이해

1. 물리 데이터 모델 정의

: 논리 데이터 모델을 특정 데이터베이스로 설계함으로써 생성된 데이터를 저장할 수 있는 물리적인 스키마를 말함

- 물리 데이터 모델링은 데이터를 실제 영구적으로 저장하고 관리하는 ORACLE, MSSQL, MYSQL 등과 같은 RDBMS 환경에 맞게 명칭 및 데이터 타입을 노닐 데이터 모델링 산출물을 기반으로 지정
- RDBMS에 맞게 테이블명, 컬럼명, 데이터 타입을 정확히 지정

* 데이터 모델링 툴 : StarUML, Vision, Erwin

2. 물리 데이터 모델 의미

: 관계 데이터 모델링(RDM)이라고 하며 논리 데이터 모델링에서 도출된 내용 변환을 포함하여 데이터의 저장공간, 데이터의 분산, 데이터 저장 방법 등을 고려하는 단계. DB 운영 성능으로 나타나므로 소홀히 다루면 안됨

3. 논리 데이터 모델의 물리 데이터 모델로의 전환

엔터티 → 테이블, 어트리뷰트 → 컬럼, UID → PK, RelationShip → FK, Mandatory → Not null, Optional → null

2 절. 물리 요소 조사 및 분석

1. 시스템 구축 관련 명명 규칙

: 사내의 시스템 구축과 관련된 명명 규칙을 파악하여 적용

2. 하드웨어 자원 파악

: CPU, MEMORY, DISK, I/O Controller, Network (현재 처리 가능 속도, 집중부하 시간대 등)

3. 운영체제 및 DBMS 버전 파악

: 운영체제 관련 요소를 파악하고 적절히 관리되고 있는지 파악. (특히 인스턴스 관리 기법)

4. DBMS 파라미터 정보 파악

: 환경적응 단계에서 가장 중요하게 고려하는 단계. 저장공간/메모리 관리 기법 등에 관련된 파라미터에 주의. 쿼리에 사용하는 옵티마이저의 운영 방법 등도 중요한 고려 사항

5. 데이터베이스 운영과 관련된 관리 요소 파악

: 사용자 관리 기법 및 정책, 백업/복구 기법 및 정책, 보안 관리 정책

3 절. 논리-물리 모델 변환

1. 논리 데이터 모델- 물리 데이터 모델 변환 용어

추가 : Secondary UID → Unique Key , Business Constraints → Check Constraints

2. 엔터티 - 테이블 변환

가. 테이블 설명

나. 서브타입 변환

1) 하나의 테이블로 변환

- 서브타입을 슈퍼타입에 통합하여 하나의 테이블로 만들
- 이 통합된 테이블에는 모든 서브타입의 데이터를 포함해야 함
- 주로 서브타입에 적은 양의 속성이나 관계를 가진 경우 적용

가) 절차

- 슈퍼타입으로 명칭 부여
- 서브타입을 구분할 수 있도록 컬럼 추가
- 슈퍼타입의 속성을 컬럼명으로
- 서브타입의 속성을 컬럼명으로
- 슈퍼타입의 관계를 FK 로
- 서브타입의 관계를 FK 로

나) 테이블 사례

- EMPLOYEE 테이블 -> 시간제, 풀타임 직원 등등.. 컬럼으로 적용

다) 하나의 테이블로의 통합이 유리한 경우

- 데이터 액세스가 좀 더 간편하며, 뷰를 활용하여 각 서브타입만 액세스하거나 수정 가능
- 수행속도가 좋아지는 경우가 많음. 서브타입 구분 없는 임의 집합의 가공 용이
- 다수의 서브타입 통합 시 조인 감소 효과 크고, 복잡한 처리 하나의 SQL 로 통합이 용이

라) 하나의 테이블로의 통합이 불리한 경우

- 특정 서브타입의 Not Null 이 제한 불가하며, 컬럼 수와 블록수가 증가
- 처리시 마다 구분(TYPE)이 필요해지는 경우가 많음. 인덱스 크기가 증가
- 한 테이블에서 모으므로 널인 컬럼(서브타입)이 반드시 있게 됨

2) 여러 개의 테이블로 분할 (슈퍼타입을 서브타입에 분할) 즉 슈퍼타입 테이블은 없음

- 각각의 서브타입마다 하나의 테이블을 만들
- 이때 분할된 테이블에는 해당 서브타입의 데이터만 포함
- 주로 서브타입에 많은 양의 속성이나 관계를 가진 경우 적용

가) 절차

- 서브타입마다 테이블 명칭 부여
- 서브타입의 속성을 컬럼명으로
- 테이블마다 슈퍼타입의 속성을 컬럼으로
- 서브타입에 해당되는 관계들을 FK 로
- 테이블마다 슈퍼타입 관계들을 FK 로

나) 여러 개의 테이블로 분할한 경우가 유리한 경우

- 각 서브타입 속성들의 선택 사항이 명확한 경우에 유리하고, 처리 시마다 유형 구분이 불필요
- 전체 테이블 스캔 시 유리하고, 단위 테이블의 크기가 감소

다) 여러 개의 테이블로 분할한 경우가 불리한 경우

- 서브타입 구분 없이 처리할 경우 UNION 발생, 처리 속도가 감소
- 트랜잭션 처리 시 여러 테이블을 처리하는 경우 증가하고, 복잡한 처리의 SQL 통합이 어려움
- 부분 범위 처리가 불가능해질 수 있음. 여러 테이블을 합친 뷰는 조치가 가능하고, UID 유지 관리가 어려움

3) 아크(ARC) 형태로 적용 (슈퍼타입과 서브타입을 분리)

- 슈퍼타입과 서브타입을 각각 테이블로 변환한 경우
- 슈퍼타입과 서브타입 테이블 간에는 Arc 관계가 생성. 즉, 여러 서브타입중 하나만 관계를 갖게됨

가) 다음의 여러 가지 경우를 만족할 때 적용

- 전체 데이터 처리가 빈번히 발생할 때
- 서브타입의 처리가 주로 독립적으로 발생될 때 (서브타입에 릴레이션이 각각 있을 때)
- 테이블로 통합 시 컬럼 수가 너무 많아질 때
- 서브타입의 컬럼 수가 많은 경우
- 트랜잭션이 주로 공통 부분(슈퍼타입)에서 발생될 때
- 슈퍼타입 처리 범위가 넓고 빈번하여 단일 테이블 클러스터링을 해야 할 때

다. 서브타입 변환의 예

- 물리적 모델 단계에서 기존에 정의해 둔 서브타입이 아닌 다른 방법으로 하고 싶다면 논리적 데이터 모델에 새로운 서브타입 세트를 추가적으로 정의

3. 속성 - 컬럼 변환

가. 일반 속성 변환

- 엔터티에 있는 속성들을 사례표의 컬럼명란에 기록
- 컬럼의 명칭은 되도록 표준화된 약어를 사용
- SQL 의 예약어 사용을 피하고, 가능한 명칭은 짧은 것이 좋음(SQL 해독시간 단축)
- 필수 입력 속성은 Nulls/Unique 란에 NN 을 표시
- 실제 테이블 설계 검증을 위해 가능하면 표본 데이터를 넣음

나. Primary UID → 기본키(Primary Key) 변환

1) 변환 절차

- 키 형태 란에 엔터티의 Primary UID 에 속하는 모든 속성에 PK 를 표시
- PK 로 표시된 모든 칼럼은 반드시 NN,U 로 표시하고 여러 개의 칼럼으로 UID 로 구성된 경우 각각 NN,U1 을 표시
- 또 다른 Unique Key(Secondary UID) 있으면 U2 로 표시

다. Primary UID(관계의 UID Bar) → 기본키(Primary Key) 변환

- 논리 모델에서는 자체 생성된 것 외에 다른 집합(엔터티)와의 관계에서 생성되는 UID 속성도 존재함. 이런 관계 속성 UID 의 변환은 기본적인 속성 UID 변환과 약간 다르다.

1) 변환 절차

- 테이블에 외래키 칼럼 포함
- PK 의 일부분 표시
- 여러 UID Bar 가 있는 경우 Pk,Fk1,... 표시

라. Secondary(Alternate) UID → Unique Key 변환

- 논리 모델에서 정의한 Secondary/Alternate 들은 해당 집합과 상태 집합과의 선택적인 관계를 가질 수 있게 하는 데 중요한 역할. 변환 절차는 기본적으로 Primary UID 변환절차와 동일

마. 테이블 정의서

: 개발자들이 가장 많이 참조하는 산출물 중 하나

4. 관계 변환

가. 1:M 관계 변환

: 가장 많은 형태의 관계로, M 쪽 관계의 형태에 따라서 관계 칼럼의 선택 사양이 결정된다.

1) 변환 절차

- 1 에 있는 PK 를 M 의 FK 로 변환
- FK 명칭 결정
- 보통 NN 표시, 필수 관계가 아니면 NN 을 체크하지 않음

2) 1:M 관계에서 1 쪽에 Mandatory 관계일 때의 변환 시 주의사항

- 자식 쪽의 레코드(Row)가 반드시 하나 이상이어야 부모 쪽의 레코드를 생성할 수 있음
- 자식 쪽의 레코드를 삭제할 경우 전체를 다 삭제할 수는 없고 반드시 하나 이상의 자식레코드를 남겨두어야한다. 또는 자식, 부모 레코드를 동시 삭제 해야함

나. 1:1 관계 변환 : 자주 발생하지는 않음. 관계의 옵션에 따라 다른 방법으로 적용됨. 양쪽 모두 옵션인 경우 더욱 빈번하게 사용되는 테이블이 외래키를 가지는 것이 유리하다.

1) 변환 절차

Mandatory (필수 관계) 반대 쪽에 있는 테이블의 기본키를 Mandatory 쪽 테이블의 외래키로 변환하며, NN 표시를 한다.

2) 변환시 주의사항

- 1 대 1 관계에 의해 생긴 모든 웰키 부분은 Unique Key 가 필수적
- 한쪽이 Optional 이고 다른 쪽이 Mandatory 라면 Mandatory 쪽의 테이블에 외래키가 생성됨
- 양쪽다 Mandatory 라면 변환 시 어떤 테이블에 외래키를 생성할지 선택해야함

다. 1:M 순환 관계 변환

: 대부분의 경우는 계층 관계 표현에 사용, 최상위 관계 속성은 반드시 Optional 이어야 함

1) 변환 절차

- 해당 테이블 내에 외래키 칼럼을 추가하고, 외래키는 같은 테이블 내에 다른 로우의 기본키 컬럼을 참조함
- 외래키 칼럼 명칭은 가능한 관계 명칭을 반영하고 외래키는 결과 NN 이 될 수 없음

라. 배타적 관계 변환

(배타적 관계 - 오직 하나만의 개체와 연결될 수 있음을 의미)

1) 외래키 분리 방법

- 각각의 관계를 관계 컬럼으로 생성하는 방법으로, 실제 외래키 제약 조건을 생성할 수 있는 장점이 있으나, 각각의 키 컬럼들이 Optional 이어야 함. 또한 다음과 같은 제약 조건을 추가해야함.

2) 외래키 결합 방법

- 각각의 관계를 하나의 관계 컬럼으로 생성. 실제 외래키 제약조건을 생성할 수 없는 단점이 있고 각각의 관계를 선택적으로 구분할 수 있는 추가적인 컬럼 필요

5. 관리상 필요한 컬럼 추가

가. 개념 : 관리상의 이유나 프로그래밍이 좀 더 빨라질 수 있을 때 사용 (예. 등록일자, 시스템번호)

나. 시스템 컬럼 추가 예 : 생성 일시, 생성 프로세스 ID 등

6. 데이터 타입 선택

가. 개념 : 물리적인 DBMS 특성과 성능을 고려하여 최적의 데이터 타입 선택

나. 문자 타입

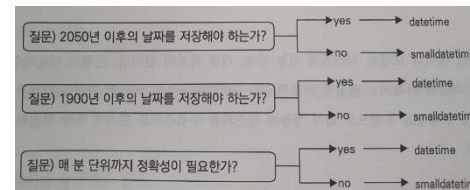


다. 숫자 타입

- Boolean : 0/1 참/거짓
- 정수(Integer) : 소수점 이하 처리 안할 때
- 소수(Decimal) : 소수점 이하 처리
- 화폐(Money) : 금액을 저장하기 위해 처리

라. 날짜 타입

: 날짜 타입으로 할지 문자 타입으로 할지 논리 단계에서 결정됨



7. 데이터 표준 적용

가. 개념: 컬럼명을 생성하고 변환하는 과정에서 미리 생성된 데이터 표준을 따르게 됨
(표준 용어, 도메인, 명명 규칙 등)

나. 데이터 표준 적용 대상

- 데이터베이스, 스토리지 그룹(물리적인 DISK를 묶어서 하나의 그룹으로 정의한 것), 테이블 스페이스(테이블이 생성되는 물리적인 영역), 테이블, 항목(Column), 인덱스, 뷰

다. 데이터 표준 적용 방법

1) 명명 규칙에 의한 표준화 적용

- 논리 → 물리 전환 시 엔터티 한글명과 동일한 용어를 영문명으로 전환
- 영문명은 영문 약어를 사용하며, 표준 용어 사전에 등록된 표준 영문 약어를 참조
- 테이블의 명명 순서 : 업무 영역 + 주제어 수식어 + 주제어 + 분류어 수식어 + 분류어 + 접미사

2) 표준 용어집에 의한 표준화 적용 : 사전에 모든 객체명과 데이터 타입, 길이 등을 정해놓고 사용

4 절. 반정규화

: 정규화된 데이터 모델은 시스템의 성능 향상, 개발 과정의 편의성, 운영의 단순화를 위해 정규화의 원칙들에 위배되는 행위를 의도적으로 수행하게 되는데 이러한 과정을 반정규화라고 한다. 일관성/무결성을 우선으로 할지 성능과 단순화를 우선순위로 할지에 따라 적절히 조화되어야 함.

1. 테이블 분할

가. 개념

: 수직/수평 분할하는 것을 테이블 분할 또는 파티셔닝이라고 함 (DB 디자인의 파티셔닝과 다름)

나. 수평 분할

- 레코드 기준으로 분할. 테이블 데이터가 너무 많고 특정 덩어리의 범위만을 액세스하는 경우 사용
- 분할 후 서로 다른 디스크에 위치 시켜 물리적인 디스크 효율성 극대
- 현재는 DBMS 차원에서 제공

다. 수직 분할

- 하나의 테이블에 컬럼이 너무 많아서 분할
- 조회 위주의 컬럼과 갱신 위주 컬럼이 나뉠 때
- 특별히 자주 조회되는 컬럼이 있을 때
- 특정 컬럼의 크기가 클 때
- 특정 컬럼에 보안을 적용해야 할 경우

2. 중복 테이블 생성

가. 개념

: 집계 함수를 이용하여 자주 조회할 때 특정 통계 테이블을 두거나 중복 테이블을 추가

나. 중복 테이블 생성 판단 근거

- 정규화에 충실하면 종속성, 활용성은 향상되지만 수행 속도 증가가 발생하는 경우
- 많은 범위 자주 처리할 때
- 처리 범위를 줄이지 않고는 수행속도 개선을 못할 때
- 요약 자료만 주로 요구되는 경우, 추가된 테이블의 처리를 위한 오버헤드 고려하여 결정
- 인덱스 조정이나 부분 범위 처리로 유도, 클러스터링을 이용해 해결할 수 있는지 철저히 검토&사용

다. 중복 테이블 유형

1) 집계(통계) 테이블 추가

가) 집계 테이블 유형

: 단일 테이블의 GROUP BY, 여러 테이블의 조인 GROUP BY

나) 집계 테이블 생성 시 유의 사항

- 로우 수와 활용도 분석을 통해 효율성을 검토해야 함
- 집계 테이블에 단일 테이블 클러스터링을 한다면 집계 레벨을 더 낮춰 활용도를 높일 방안 검토
- 클러스터링, 결합 인덱스, 고단위 SQL로 해결될 수 있는지 검토
- 지나친 집계 테이블은 만들지 않는게 좋음
- DB 트리거의 오버헤드에 주의하고 일관성 보장에 유의. 일관성 유지 중요

2) 진행 테이블 추가

가) 진행 테이블 추가 상황

- 여러 개의 조인이 빈번히 발생하며 처리 범위도 넓은 경우
- M:M 관계가 포함된 처리의 과정을 추적/관리하는 경우
- 검색 조건이 여러 테이블에 걸쳐 다양하게 사용되고 복잡한 처리량이 많을 때

나) wls godx p 이블 생성 시 유의사항

- 데이터 양이 적절하고 활용도가 높아지도록 기본키 선정
- 필요에 따라 추출 컬럼을 추가하여 집계 테이블 혁할도 하는 다목적 테이블 구성
- 진행 테이블을 만들어야 하는지 반드시 확인

3. 중복 컬럼 생성

가. 개념

- 정규화를 여기면서 중복을 수행

나. 중복 컬럼 생성 상황

- 빈번한 조인을 일으키는 컬럼,속도가 중요한 컬럼, 액세스 조건으로 자주 사용되는 컬럼에 대해서 고려해 볼 수 있다.
- 자주 사용되는 액세스 조건이 다른 테이블에 분산되어 액세스 범위를 줄이기 힘든 경우
- 복사된 컬럼은 원본과 동일 해야함
- 상위 레벨의 테이블에 집계된 컬럼을 추가 가능 (M:1)
- 하위 레벨 테이블로 중복 컬럼을 복사(M:1)
- 행으로 관리하던 데이터를 컬럼으로 관리하면, 기본키가 길거나 여러 개의 컬럼인 경우 인위적인 기본키 추가 가능

다. 유의사항

- SQL GROUP으로 처리 가능한지 검토
- 반복 컬럼은 가능한 사용하지 말고 직렬화 같은거 쓰지말기
- 가능한 적게 가져가면 좋고
- 클러스터링, 결합 인덱스, 적절한 SQL로 해결 가능한 것이 많음
- 경우에 따라 상대 테이블의 ROWID를 복사하는 경우가 효과적일 수 있음
- JOIN, SUB-QUERY 액세스 경로의 최적화 방안을 더욱 철저히 강구해야 함
- 지나치면 일관성 오류 및 오버헤드가 증가함