

2018년 DA 설계공모전 문제	
분 야	학생부
수행과제 및 전제조건	<p>제시된 지문에 대한 요구사항분석서, 데이터표준정의서를 작성하고, 최적의 논리 데이터 모델을 작성하시오.</p> <p>단, ERD 작성 시 반드시 다음 조건을 충족해야 함.</p> <ol style="list-style-type: none"> <li>1. 제출된 논리 데이터 모델에는 다대다 관계가 남아 있지 않아야 함.</li> <li>2. 식별자는 지문 상에서 특별한 지시가 없는 한 본질식별자를 기준으로 정의해야 함.</li> <li>3. 서브타입 정의가 필요한 경우에 구분(유형)코드 엔터티로 대체하는 것은 불가피한 경우를 제외하고는 가급적 지양할 것</li> </ol>

우리는 현재 사용하고 있는 정보시스템의 데이터를 새로 개발되거나 도입한 정보시스템으로 옮기는(이행하는) 사업을 주로 하고 있다. 데이터를 이행하게 되는 정보시스템은 차세대 기간제 시스템이거나 데이터웨어하우스와 같은 정보계 시스템이 주 대상이지만 간혹 산재한 여러 시스템의 데이터들을 하나의 데이터베이스로 옮기는 경우도 있다. 현행 정보시스템의 데이터는 관계형 데이터베이스에 저장되어 있는 정형 데이터를 비롯하여 사진 파일 같은 일부 비정형 데이터와 현업 사용자들이 엑셀 파일로 별도 관리하고 있는 수기 데이터까지 포함될 수 있다. 이와 같은 데이터 이행은 목표 시스템으로의 전환을 위해 목표 일정을 정해 놓고 수많은 테스트와 검증을 거쳐 목표한 시점에 맞춰 수행하게 되는 경우도 있고, 주기를 정하여 지속적으로 수행하게 되는 경우도 있는데, 어떠한 경우라도 한정된 시간 내에 대용량의 데이터를 오류 없이 정확하게 처리해야 하면서도 적은 인원으로 완수해야하기 때문에 보다 체계적이고 효과적으로 수행할 수 있는 관리시스템을 구축하여 사업에 활용하려 한다.

우리가 이행 대상으로 삼는 데이터는 주로 관계형 데이터베이스 시스템의 테이블에 저장된 데이터이기 때문에 현행 시스템(또는 AS-IS 시스템)과 목표 시스템(또는 TO-BE 시스템)의 테이블과 컬럼을 관리해야 한다. 테이블명은 하나의 DB 또는 유저 스키마(이후 “스키마”로 통칭) 내에서 고유하게 식별되지만 대상 시스템이나 서버 전체로 보면 여러 개의 스키마가 있을 수 있고, 이러한 경우에 스키마를 달리 하여 동일한 테이블명이 존재할 수 있다. 테이블명이 동일하더라도 컬럼 구성은 동일하지 않을 수 있어 같은 테이블이라고 단정할 수는 없다. 대부분의 고객사는 하나의 서버 환경에서 현행 시스템이 운영되고 있지만, 어떤 고객사는 복수의 서버로 분산 환경으로 구성하여 이행 대상이 되는 현행이나 목표 시스템 범위 내에 다수의 스키마 구성이 존재할 수 있고, 따라서 동일한 이름의 테이블이 여러 개 존재할 수 있다. 그래서 이행 대상이 되는 테이블을 식별하려면 테이블명 외에도 서버가 위치한 지역명과 서버(서버명, IP주소 등), 해당 테이블이 생성되어 있는 스키마 이름을 함께 알아야 한다.

테이블명은 employee, department 등과 같이 보통 영문으로 되어 있으며, 테이블명의 길

이가 짧을 때는 Full Name을 사용하기도 하지만 보통은 테이블의 내용을 적절하게 나타낼 수 있는 단어들을 조합하여 복합 명사 형태로 테이블을 명명하기 때문에 길이 제약이나 간편한 코딩을 위해 영문 약어 조합을 많이 사용한다. 이행 대상으로 관리할 테이블에 대해 영문 테이블명 외에 테이블 한글명, 테이블 정보를 수집한 기준일자를 추가로 관리해야 하며, 가급적이면 현행 테이블에 대해서는 수집한 기준일자 시점에서의 저장 건수를, 목표 테이블에 대해서는 예상되는 최대 저장 건수를 추가로 관리하는 것이 좋다. 이행 대상이 되는 현행, 목표 테이블은 업무 사정에 따라 언제라도 추가, 삭제, 변경 등이 발생할 수 있기 때문에 주기적으로 테이블 정보를 다시 수집하여 변경된 내용을 반영하되, 언제라도 원하는 시점에서의 테이블 목록을 재구성할 수 있도록 해야 한다.

하나의 테이블 내에서 컬럼명은 유일하게 사용되며, 테이블명처럼 보통은 영문으로 되어 있고, 컬럼명의 길이가 짧을 때는 Full Name을 사용하기도 하지만 컬럼의 의미를 적절하게 나타낼 수 있는 단어들을 복합 명사 형태로 조합하여 컬럼명을 부여하기 때문에 길이 제약이나 간편한 코딩을 위해 영문 약어 조합을 컬럼명으로 많이 사용한다. 테이블을 구성하는 컬럼에 대해 컬럼명(영문컬럼명), 한글컬럼명, 컬럼순서, 데이터타입, 길이, NOT NULL 여부, PK여부 및 순서 등을 관리한다. 여기서 PK여부 및 순서란 하나 이상의 컬럼으로 PK를 구성하는 경우 1, 2, 3 등과 같이 PK컬럼의 구성 순서를 나타내는 것이며, PK가 단일 컬럼으로 구성된 경우는 1이라고 표시하면 PK컬럼이면서 첫 번째라는 순서까지 의미하게 되므로 이와 같은 integer 값으로 PK컬럼 여부와 그 순서를 동시에 표시함을 의미한다. 주기적인 테이블 정보 수집 시 테이블이 변경되었다면 그것은 주로 컬럼이 추가되거나, 삭제 또는 컬럼의 데이터 타입이나 길이 등이 변경된 경우이고, 또는 컬럼의 순서가 바뀌었을 수도 있기 때문에, 변경이 발생한 테이블에 대해 기존에 갖고 있던 컬럼 데이터들은 일괄적으로 수집일자 하루 전 시점으로 유효종료일자를 변경하고, 새로 수집한 컬럼들에 대해 수집일자를 유효시작일자로, 유효종료일자는 9999년 12월 31일로 부여하여 새로 저장한다. 이와 같은 데이터 처리를 통해 원하는 임의의 시점에서의 테이블별 컬럼 현황을 재구성할 수 있어야 한다. 테이블 정보 수집 시 삭제된 테이블이 발견되면 해당 테이블의 컬럼들에 대해 유효종료일자를 수집일자 하루 전 날짜로 갱신 처리한다. 어떤 분산 환경에서는 동일한 명칭의 테이블이 분산 서버에 따라 컬럼 구성이나 데이터 타입이 다르게 운영되는 곳도 존재할 수 있다. 어떤 컬럼은 여러 테이블에서 공통적으로 사용될 수 있고, 때로는 여러 컬럼을 그룹핑하여 임의의 테이블에 해당 그룹의 컬럼들이 한꺼번에 반영될 수도 있다. 이를 위해 컬럼을 하나의 개체처럼 생각하고 core에 해당하는 항목들로 별도의 unique한 컬럼 정보를 관리하면서 임의의 테이블에 컬럼이 할당되었을 때 필요한 항목들만 추가로 관리하도록 구성하고자 한다.

현행 시스템 데이터를 목표 시스템으로 이행하려면 기본적으로 현행 데이터 구조와 목표 데이터 구조 간의 매핑 정의가 선행되어야 하는데, 이를 위해 테이블과 컬럼 각각에 대한 매핑 정보를 관리해야 한다. 테이블 매핑 정보는 각각의 목표 테이블에 대해 어떤 현행 테이블의 데이터를 이행할 것인지를 정의한 것이다. 보통 하나의 목표 테이블은 하나의 현행 테이블과 매핑되거나 복수의 현행 테이블과 매핑될 수 있고, 하나의 현행 테이블은 복수의 목표 테이블에 매핑될 수도 있다. 하나의 테이블 매핑 정보에 대해 작성자와 작성일자를 관리해야 하며, 테이블 정보 수집에서 변경 사항을 반영해야 하는 테이블에 대해서는 기존

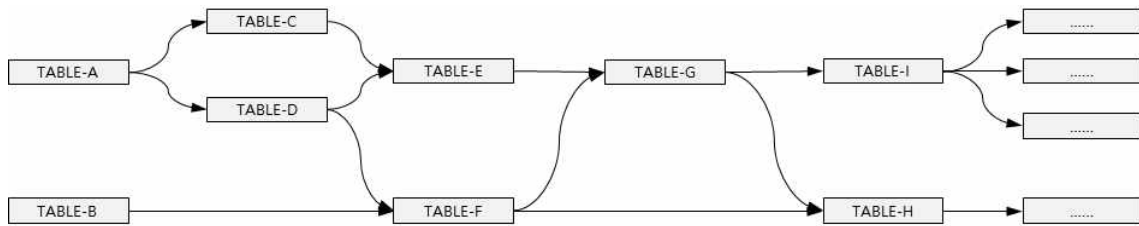
의 매핑정보에 대한 유효종료일자를 수집일자의 하루 전 날짜로 변경하고 새로운 매핑 정보를 구성하여 다시 저장해야 한다.

테이블 매핑 정보와 함께 현행 및 목표 테이블의 컬럼들 간에 대해 상세한 매핑 정보를 추가로 관리해야 한다. 하나의 목표 테이블의 컬럼에 대해 현행 컬럼의 매핑은 해당하는 테이블 매핑 정보에 정의된 현행 테이블의 컬럼들로 한정되며, 하나의 목표 컬럼에 대해 하나의 현행 컬럼이 매핑되거나 서로 다른 현행 테이블의 컬럼이 복수로 매핑될 수도 있다. 목표 컬럼에 대한 현행 컬럼의 매핑은 현행 컬럼에 저장된 데이터를 그대로 단순 이행하거나 변환함수를 사용하여 변환 이행할 수 있고, 또는 현행 컬럼에 저장된 데이터를 목표 컬럼에 이행할 때 특정 값으로 치환할 수도 있다. 현행 컬럼에 저장된 코드값을 목표 시스템에서 새로운 코드 값으로 변환하고자 하는 경우는 사전에 신규 코드 매핑 테이블을 생성하여 현행 테이블과 조인 처리를 통해 이행하는 매핑 변환으로 처리할 수도 있다. 또 어떤 경우에는 현행에 존재하지 않았던 컬럼이거나 여타의 사유로 인해 목표 컬럼에 현행 컬럼을 매핑하지 않고 고정된 상수값을 매핑해야 할 수도 있다. 이와 같은 변환유형을 구분하는 것은 컬럼 매핑 정보에서 중요한 사항이며, 변환유형에 따라 필요한 매핑규칙을 추가로 정의하기 위한 매핑규칙 속성이 관리되어야 한다. 하나의 테이블 매핑 정보에 대해 해당하는 컬럼 매핑 정보들은 컬럼매핑순번 속성을 추가하여 식별한다.

컬럼 매핑 정보에서 변환유형이 매핑 변환인 경우에 사용할 신규코드매핑 정보는 현행 시스템에서 사용하고 있는 임의의 코드그룹, 코드가 목표 시스템의 어떤 코드그룹, 코드로 변환되는지를 정의한 것이다. 현행 및 목표 시스템의 코드 정보는 코드그룹, 코드로 유일하게 식별할 수 있는 통합 코드 집합으로 전제한다. 코드 매핑을 위해 사전에 현행 시스템의 코드 데이터를 수집하여 코드그룹, 코드, 코드명, 코드설명 등의 항목으로 통합해 둘 필요가 있으며, 목표 시스템에서 사용할 코드에 대해서도 미리 코드 데이터를 동일한 항목으로 구성하여 사용한다. 편의상 코드 매핑에서 하나의 현행 코드가 복수의 목표 코드로 분할되는 경우는 없고, 복수의 현행 코드가 목표 시스템의 특정 코드로 변환되는 반대의 경우는 존재할 수 있다고 전제한다. 대부분의 경우는 현행 코드가 목표 코드에 대해 1:1로 변환된다. 현행 및 목표 컬럼 중에는 동일한 코드그룹을 사용하는 컬럼들이 다수 존재하기 때문에 이들이 어떤 코드그룹의 코드셋을 사용하는지를 컬럼 정보에 추가로 정의한다. 하나의 컬럼이 복수의 코드그룹을 사용하는 경우는 없고, 어떤 컬럼은 할당되는 테이블에 따라 동일한 코드그룹 내에서 서로 다른 코드셋을 사용할 수도 있지만 편의상 이와 같은 부분은 고려하지 않기로 한다.

목표 테이블로 현행 테이블의 데이터를 이행할 때는 데이터의 선후 관계 때문에 데이터 생성 순서를 고려해야 한다. 이를 위해 목표 테이블에 데이터를 이행하는 순서에 대한 선후행 관계를 관리해야 한다. 예를 들어, 어떤 고객이 상품을 구매한 내역 데이터를 이행하려면 그 전에 고객정보를 먼저 이행하여 고객번호가 생성되어 있어야 한다. 임의의 목표 테이블에 대해 이보다 먼저 데이터를 이행해야 하는 목표 테이블은 하나 이상일 수 있고, 어떤 경우에는 복수의 목표 테이블이 하나의 목표 테이블을 선이행한 후에 이행되어야 하는 경우도 있다. 목표 테이블을 이행하는 순서에 따라 모두 나열해 보면 아래 예시와 같은 형태가 될 수 있다. 데이터 이행의 선후 관계는 참조무결성을 보장하기 위한 측면에서 매

우 중요하기 때문에 세밀하게 관리되어야 하며, 테이블 정보를 수집하여 변경된 부분이 있을 경우 해당 테이블이 관련된 이행 순서에도 영향이 있는지를 따져 변경 내용을 반영해야 한다. 데이터 이행 선후 관계를 변경할 때는 해당 테이블과 관련된 데이터만 변경 처리하며, 기존 데이터에 대한 유효종료일자를 수집일자의 하루 전 날짜로 변경하고 새로 저장할 이행 선후 관계 데이터에 대해 수집일자를 유효시작일자로, 유효종료일자는 9999년 12월 31일로 하여 저장한다. 이것은 필요 시 원하는 임의의 시점에서의 이행 순서를 재구성할 수 있도록 하기 위함이다.



<데이터 이행 선후관계 구성 예시>

실제 데이터 이행은 테이블 매핑 정보와 컬럼 매핑 정보를 참조하여 각각의 목표 테이블로 현행 데이터를 이행하는 스크립트를 작성하여 수행하게 되며, 스크립트는 다양한 언어로 작성될 수 있으나 편의상 PL/SQL 이나 SQL 문장으로 한정한다. 스크립트는 테이블 매핑 정보를 기준으로 작성하는데, 어떤 경우에는 하나의 테이블 매핑 정보에 대해 하나 이상의 스크립트가 수행될 수도 있다. 스크립트는 매핑 정보로부터 자동 생성되거나 직접 작성하게 되고, 이렇게 작성된 스크립트는 ID를 부여하여 관리한다. 각각의 스크립트는 언제, 누가 작성했는지, 그리고 이상 유무를 언제 누가 검증했는지에 대한 정보와 함께 해당 스크립트의 소스 코드까지 관리한다. 데이터 이행 순서가 테이블 정보의 변경에 따라 변할 수 있고, 원하는 임의의 시점에서의 이행 순서를 재구성할 수 있어야 하는 것처럼, 이행 스크립트 또한 테이블 정보 변경 시 변경될 수 있으며, 원하는 임의의 시점에서 유효한 스크립트를 찾아낼 수 있도록 관리해야 한다. 스크립트는 이행 처리를 위한 것 외에 이행 결과를 검증하기 위한 검증 스크립트도 함께 작성된다. 보통 이행 스크립트가 실행되면 이 결과의 정확성을 검증하기 위한 검증 스크립트가 뒤이어 쌍으로 실행된다.

데이터 이행의 실행은 JOB으로 등록하여 관리한다. 한 번의 JOB은 여러 개의 이행/검증 스크립트를 실행 순서에 따라 묶어서 한 번의 실행 단위로 일괄 처리하는 것을 의미한다. 하나의 JOB으로 묶는 스크립트는 이행 대상 전체일수도 있고, 일부일 수도 있다. 실행시키고자 하는 스크립트들을 묶어서 JOB으로 등록한 후 즉시 실행할 수도 있고, 임의의 시점에 예약 실행을 하도록 할 수도 있다. 이를 위해 JOB 단위로 ID를 부여하여 식별하고, 누가 언제 등록했는지, 언제 실행하는지, 그리고, 그 JOB이 언제 끝났는지를 관리해야 한다. 또한 JOB이 실행된 결과가 성공인지 실패인지의 실행 상태도 관리해야 한다. JOB 단위로 실행 시점 및 실행 상태를 관리하지만 이 JOB에 속한 스크립트 단위로도 상세한 실행/종료시점과 실행 상태를 기록해야 한다. 임의의 JOB에 속한 스크립트들 중 하나라도 실패하게 되면 그 JOB은 실패로 기록된다. 검증 스크립트는 이상 없이 실행이 종료되면 검증 결과를 기록하는데, 이행 건수나 금액 컬럼의 합계 등 사전에 정의된 검증 유형에 따라 확인된 값

을 기록하여 JOB이 종료된 후 검증 결과를 확인함으로써 목표한 이행이 잘 처리되었는지 확인하는 수단으로 활용한다. 임의의 스크립트가 실패한 경우는 실패한 원인을 알 수 있도록 에러 코드와 에러 메시지, 에러가 발생한 시점 및 발생 위치 등의 정보를 관리한다. 경우에 따라서는 하나의 스크립트 실패 시 복수의 에러 코드와 에러 메시지가 발생할 수도 있다.

데이터 이행을 위해 관리해야 하는 수많은 정보에 대해 작성자 또는 검증자 등을 확인하기 위해 우리는 사용자 정보를 관리한다. 사용자는 우리 직원이거나 고객사의 직원일 수 있다. 사용자에 대해 소속사 및 부서, 이름, 연락처번호, 이메일, 직급, 담당업무 등의 정보와 함께 사용자ID를 부여하여 관리한다.

우리는 이상과 같은 내용으로 데이터 이행을 관리하는 시스템을 구축하고자 하며, 현행 및 목표 시스템의 데이터 구조를 포함한 관련 정보를 관리하는데 있어서 통합성 · 유연성 · 확장성을 확보할 수 있도록 최적화된 논리 데이터모델을 설계하고자 한다.