

## 데이터 웨어하우스의 메타데이터

서강대학교 박 석·김희원·김말희

### 1. 서 론

메타데이터(metadata)는 용어상 '데이터에 관한 데이터'이다. 메타데이터는 기존 시스템'에서도 사용되었으나, 최근 네트워크 기술과 함께 컴퓨터 시스템의 복잡도가 급증하면서 그 중요성이 새롭게 대두되고 있다. 특히 데이터 웨어하우스는 기존 시스템에 나누어 산재되어 있던 정보를 통합하여 저장하는 성격 때문에, 다양하고 서로 이질적인 많은 데이터들을 포함하며, 그 관리적 측면에서도 복잡하고 많은 작업들이 수반된다. 따라서, 적절한 메타데이터는 성공적인 데이터 웨어하우스 구축의 열쇠라고 해도 과언이 아니다.

본 논문에서는, 데이터 웨어하우스에서의 메타데이터의 개념 및 역할을 살펴보고, 종류별 메타데이터를 알아보기로 한다.

### 2. 데이터 웨어하우스에서 메타데이터의 역할

메타데이터는 '데이터 생존 주기(data life cycle)' 전체에 관련된다[10]. 따라서, 데이터 웨어하우스의 메타데이터의 개념은 원천(source) 데이터 수집, 데이터 웨어하우스 내에 저장, 가공, 검색, 보급, 교환, 폐기 등에 관련된 데이터를 일컫는다.

운영 환경에서, 카탈로그와 같은 메타데이터는 소프트웨어 개발자나 데이터베이스 관리자

에게 매우 귀중한 자료이다. 그러나, 은행 직원, 여행사 직원 등과 같은 최종 사용자는 데이터베이스에 정보가 어떻게 들어있는지 알 필요가 없다. 다만, 매일 매일 사용하고 있는 응용 프로그램이 제공하는 화면과 인터페이스 형태에 종속된 단순 작업을 반복하면 된다.

그러나, 정책 결정을 지원하기 위한 환경은 매우 다르다. 최종 사용자인 데이터 분석가들은 때때로, 전에는 알지 못했던 데이터 사이에 상관관계를 찾고, 새로운 유용한 정보들을 발견하기를 원한다. 따라서 틀에 박힌 기존 응용 프로그램은 무용지물이다. 왜냐하면, 이런 일하기 위해서는 사용자가 데이터들 사이에 직접 뛰어들어서, 그 구조와 의미를 이해하면서, 정보를 찾아가는 일이 필요하기 때문이다. 그런데, 적절한 메타데이터가 없는 데이터 웨어하우스는 어떤 목록이나 레이블도 없는 서류 더미와 같다. 분석가들은 길을 잃은 사람처럼, 웨어하우스 내를 배회하게 될 것이다. 사용자가 데이터 웨어하우스를 효과적으로 사용하기 위해서, 메타데이터는 반드시 정확하고, 최신의 것이어야 한다[3].

그런데, 메타데이터에는 표준이 없고 있을 수도 없다. 왜냐하면, 응용 프로그램 공급자들이 표준안이나 중앙 관리를 고려하지 않고 필요한 메타데이터를 생성하기 때문이다[2]. 따라서 최선의 길은 어떤 데이터들이 어떤 응용 프로그램에 소속되었는지 알 수 있도록, 모든 형태의 메타데이터들에 대해 주의 깊게 기록하는 것이다.

### 3. 데이터 웨어하우스의 메타데이터

\*중심화원

1) RDB의 카탈로그 정보는 메타데이터의 좋은 예이다.

메타데이터는 서로간에 긴밀히 연관되어 있으므로 분류하는 일이 쉽지 않다. 그러나 가장 대표적이고 호소력 있는 분류 방법을 찾다면, 아마도 그 메타데이터를 이용하는 시스템 구성요소의 분류를 따르는 방법일 것이다. 그림 1은 데이터 웨어하우스를 구성하는 가장 대표적인 구성요소들과, 메타데이터의 종류를 보여주고 있다.

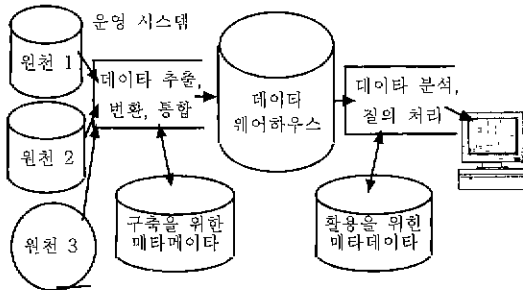


그림 1 데이터 웨어하우스의 메타데이터

그림 1과 같이, 데이터 웨어하우스는 일반적으로 크게 두개의 작업으로 되어 있다.<sup>2)</sup>

- 데이터 추출, 변환, 통합 작업-운영 시스템에서 데이터를 추출하고, 적당한 변환작업을 거쳐 통합하는 작업이다. 이 작업에서 사용되는 메타데이터를 구축을 위한 메타데이터라고 한다.
- 데이터 분석과 질의 처리-구축된 데이터 웨어하우스에서 사용자의 질의에 따라 데이터를 분석하고 처리하는 작업이다. 이 작업에서 사용되는 메타데이터를 활용을 위한 메타데이터라 한다.

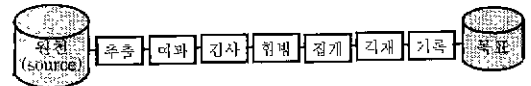
### 3.1 구축을 위한 메타데이터

구축을 위한 메타데이터는 운영 시스템에서 데이터 웨어하우스로 데이터가 유입되는 동안 거치는 작업들에 관한 정보이다. 그런데 이 작업은 매우 복잡하다. 심지어 데이터 웨어하우스

2) 예를 들어, 스탠포드 대학의 WHIPS 프로젝트[4]에서 제시한 시스템 구조는 '데이터 통합 구성요소(Data Integration Component)'와 '데이터 질의와 분석 구성요소(Data Querying & Analysis Component)'로 나누고 있으며, Vidette Poe[5]는 각각을 '데이터 민환(Data Transformation) 작업'과 'DSS 도구와 응용프로그램(DSS Tool & Application) 작업'으로 나누고 있다.

스를 설계하고 구현하는 노력의 80%를 바로 이 작업에서 소비한다[6]. 작업의 어려움은 크게 두 가지 이유를 갖고 있다.

그 첫 번째 이유는 데이터 추출 작업의 복잡성에 기인한다. 하나의 추출 작업은 몇단계의 처리 단계로 이루어져 있으며, 각 단계는 각기 다른 책임소재를 갖는다.<sup>3)</sup> 대략적인 단계들은 그림 2와 같다.



- 추출(extract) - 내부 혹은 외부의 원천(source)에서 데이터를 추출한다.
- 여과(filter) - 필요치 않은 부분을 삭제한다.
- 검사(validate) - 사용에 적합한지 데이터의 질을 측정한다.
- 합병(merge) - 다른 추출 데이터와 합병한다.
- 집계(aggregate) - 경우에 따라 요약 정보를 생성한다.
- 적재(load) - 새로 생성된 데이터를 목표(데이터 웨어하우스) 환경에 적재한다.
- 기록(archive) - 새로운 데이터에 밀려난 오래된 데이터를 값싼 저장공간으로 옮긴다.

그림 2 데이터 웨어하우스의 유입작업을 위한 작업 단계들의 예

두 번째 이유는 데이터의 원천과 데이터 웨어하우스 내의 테이블과의 대응이 다-대-다(many-to-many) 관계를 갖는 점에 있다. 즉, 그림 3과 같이 운영 환경의 데이터가 주제중심(subject-oriented)의 데이터 웨어하우스의 테이블로 옮겨지는 과정은, 다수의 임시화일과 변형 프로그램들을 통하여 다-대-다 대응을 이룬다.

따라서, 구축을 위한 메타데이터는 데이터의 원천과 목표 테이블간의 대응 정보와 각 작업

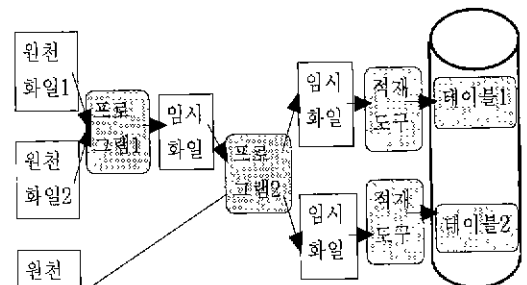


그림 3 데이터의 원천과 웨어하우스 내의 테이블

3) 각기 다른 시스템 구성요소가 수행을 해야 한다.

단계별 변환 알고리즘 등의 정보를 가지고 있어야 한다. 그림 4는 데이터 유입 과정을 표현하는 메타데이터 모델을 단순화하여 세 개의 개체만을 갖는 ER 다이어그램<sup>4)</sup>으로 표현한 것이다.

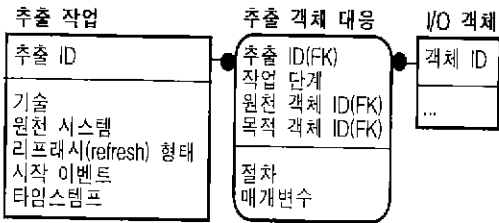


그림 4 데이터 유입과정을 묘사하는 메타데이터

- I/O 객체 개체-운영 환경의 테이블이나 화일들과 데이터 웨어하우스의 테이블로서, 각각 추출 작업에서 입력 장치와 출력 장치에 해당한다.
- 추출 작업 개체
  - 추출 ID-각 추출 작업에 대해 ID를 갖는다.
  - 원천 시스템-원천 데이터가 있는 시스템의 위치에 대한 데이터이다
  - 리프레시(refresh) 형태-새로운 데이터의 범위<sup>5)</sup>와 적재하는 방법<sup>6)</sup>을 결정한다.
  - 시작 이벤트-추출 작업을 일으키는 이벤트에 관한 데이터이다.
- 추출 객체 대응 개체-각각의 추출 작업 단계에 사용될 절차와 그 매개변수를 결정해준다.
  - 작업 단계-추출, 여과, 검사, 합병, 집계, 적재, 기록 단계들을 말한다.
  - 절차-주로 SQL문이나 응용 프로그램(특히, 데이터 변환 프로그램), 혹은 프로그램의 소스<sup>7)</sup>일 수 있다.

- 매개변수-응용 프로그램이나 절차를 작업에 맞게 구성하기 위한 데이터이다.
- 타임스탬프-추출이 일어난 시간 등의 시간 정보를 저장한다.

그림 4에 표현되지 않았으나 필수적인 메타데이터는 대략 다음과 같다.

- 이벤트 스케줄링-시간 주기적 이벤트, 미리 정의된 비주기 이벤트, 관리자에 의한 임의적 이벤트 등의 시작 조건들을 관리해야 한다.
- 여과 작업의 기준-추출 데이터의 유용성 여부를 판단할 기준이 필요하다.

이러한 이벤트에 의한 작업처리를 위해서는 트리거를 지원하는 액티브 데이터베이스의 기능을 가지고 있거나, 이에 상응하는 응용 프로그램 모듈<sup>8)</sup>이 필요하다.

### 3.2 활용을 위한 메타데이터

활용을 위한 메타데이터는 데이터 웨어하우스의 자체 관리와 사용자를 지원하기 위한 정보를 포함하고 있다. 활용을 위한 메타데이터를 다음의 세 개의 계층으로 나눌 수 있다.

- 데이터 웨어하우스의 관리를 위한 메타데이터
- 질의처리 지원을 위한 메타데이터
- 사용자 지원을 위한 메타데이터

#### 3.2.1 데이터 웨어하우스의 관리를 위한 메타데이터

데이터 웨어하우스 자체를 관리하기 위한 메타데이터이다. 즉, RDB에서의 카탈로그 시스템을 확장한 것으로 볼 수 있다. 그림 5는 데이터 웨어하우스를 위한 카탈로그의 대략적 형태를 ER 모델로 표현한 것이다.

그림 5의 메타데이터는 데이터 웨어하우스의 데이터 모델을 데이터로 저장한다. 즉, 데이터 모델을 이루는 개체(entity). 개체간 관계(relationship), 개체의 속성(attribute)을 포함한다. 다음은 그림 5의 주요 개체와 데이터 모델의 3 요소를 대응시킨 것이다.

4) 데이터 웨어하우스의 메타데이터 저장소로서 RDB를 사용하는 현재적 추세에 편승하여, 메타데이터 모델링 방법으로 ER 다이어그램을 사용하기로 한다.  
 5) 일반적으로 전체 데이터를 추출하거나, 혹은 갱신 부분만을 추출할 수 있다.  
 6) 새로운 데이터가 기존 데이터를 대체하거나, 혹은 단순히 추가될 수 있다.  
 7) 각 작업단계에 따라 해당 절차를 호출하는 소스 프로그램 화일을 완성하여 컴파일 과정을 거쳐 수행하는 방법이다.

8) 특정 데이터나 시간 정보를 계속적으로 확인하여 어떤 이벤트의 시작 조건을 만족하면, 해당 추출 작업의 각 단계를 위한 절차를 호출하는 프로그램이 필요하다.

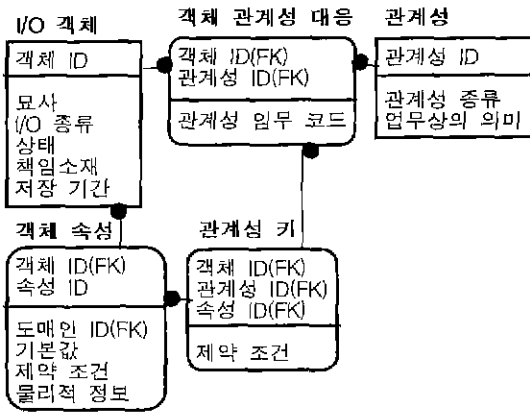


그림 5 데이터 웨어하우스 관리를 위한 메타데이터 모델

- I/O 객체 개체-데이터 모델의 개체(entity)에 대응
- 관계성 개체-데이터 모델의 관계(relationship)에 대응
- 객체 속성 개체-데이터 모델에서 개체가 갖는 속성(attribute)에 대응

다음은 각 개체와 속성들에 대한 설명이다.

- I/O 객체 개체-데이터 모델 개체(entity)에 관한 데이터를 튜플로 갖는 테이블이다.
  - I/O 종류-객체의 논리적/물리적 종류에 관한 데이터이다. 즉, 테이블, 문서화일, 코드화된 화일 등등이다.
  - 상태-때때로 동일한 데이터 웨어하우스의 각 부분들이 상이한 개발단계에 있을 수 있다. 예를 들어, 테이블은 '설계중', '시험중', '활성', '비활성' 등으로 분류될 수 있다.
  - 책임소재(stewardship)-데이터 웨어하우스는 통합적 속성 때문에 특정 부서나 업무 팀에만 소속되지 않는다. 따라서, 각 데이터 집합의 제공자를 식별하여, 조사와 오류수정 등이 적합한 팀에 의해 이루어지도록 하는 일이 필요하다.
  - 저장기간-데이터가 웨어하우스 내에 유지되는 시간에 관한 데이터이다. 또 오래된 데이터를 삭제(purging)하는 기준과 시간적 척도 등을 알려주는 것도 바람직하다.
- 관계성 개체-데이터 모델의 각 관계(re-

lationship)에 관한 데이터를 튜플로 갖는 테이블이다.

- 관계성 종류-두 개체간의 관계(relationship)가 갖는 성격을 분류한다. 예를 들어, 다-대-다 혹은 1-대-1 관계 등이 있다.
  - 업무상의 의미-관계가 갖는 의미를 업무상의 용어로 설명해주는 문서 데이터이다.
  - 객체 속성 개체-데이터 모델에서 개체가 갖는 각 속성들에 관한 데이터를 튜플로 갖는 테이블이다.
    - 도메인 ID-속성이 갖는 도메인을 외부 테이블의 ID<sup>9)</sup>형태로 지원한다.
    - 기본값-속성의 기본값 등
    - 제약조건-예를 들어 널(NULL) 값의 허용 여부 등
    - 물리적 정보-디스크에서 시작 위치, 길이, 사용된 인덱스 구조 등에 대한 정보이다.
  - 객체 관계 대응 개체-개체와 개체가 가진 관계(relationship) 사이를 연결해준다. 관계성 임무 코드 속성은 '부모(parent)'와 '자식(child)' 값을 가질 수 있다.<sup>10)</sup>
  - 관계성 키 개체-관계(relationship)에 사용된 키 값들을 알려준다.
- 이 외에도 다음의 관리 정보를 메타데이터로 정의하는 것이 필요하다.

- 버전 관리-데이터 웨어하우스는 최근의 메타데이터만을 포함하지 않고, 장시간에 걸쳐 메타데이터의 변천사(metadata history)<sup>11)</sup>를 포함해야 한다. 데이터 웨어하우스가 성장하면서, 데이터 모델, 설명문과 소유권 기록, 제약 사항들, 카디널리티(cardinality)에 관한 메타데이터도 같이 변화하게 된다. 따라서 시간적인 질의를 처리하고자 할 때, 적당한 데이터에 정확

9) 외부 도메인 테이블은 다양한 조건을 표현할 수 있어야 한다. 예를 들어, 성별 속성은 '남', '여', '무', '모름'값만을 허용하는 조건이 있다.

10) 데이터 웨어하우스가 스타스키마를 기반으로 하고 있다면, 일반적으로 각 차원 테이블은 '부모'가 되고 사실 테이블은 '자식'이 된다.

11) 메타데이터의 변천사(history)의 필요성은 자연스럽게 메타-메타데이터(metametadata)[7]의 등장을 요구한다. 즉, 메타데이터를 데이터로서 저장할 때 필요한 더 상위 수준의 데이터의 필요성이 대두된 것이다.

한 메타데이터가 결합되어야 한다.

- 성장계측(growth metrics)-데이터 양의 갑작스런 증가는 적재 시간과 성능 측면에서 충격을 가한다. 또한 데이터마트(data-mart)에 있는 낡은 정보의 양을 증가시킨다. 따라서 성장계측도 없었던 안될 중요한 메타데이터이다. 이 정보를 가지고 적절한 시간에 적절한 시스템 자원의 획득을 보장하도록 해준다. 은행업무 환경에 적용된 어떤 데이터 웨어하우스 프로젝트는 하루의 데이터 적재시간이 24시간을 넘는 이유로 인해 실패했다[9]. 사전 모니터와 데이터 성장 패턴을 예측하는 것은 웨어하우스 프로젝트의 실패를 방지하는 중요한 작업이다.

### 3.2.2 질의처리 지원을 위한 메타데이터

데이터 웨어하우스는 질의 성능의 향상을 목적으로 다단계 요약 데이터를 갖는다. 그림 1의 단순화된 구조에서 데이터 웨어하우스만을 자세히 도식화시킨 것이 그림 6이다.

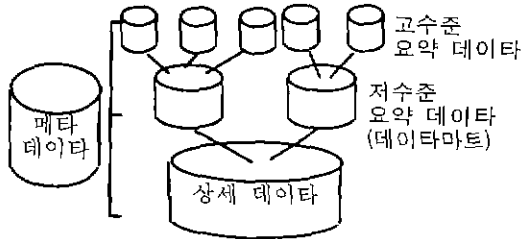


그림 6 다단계 요약 데이터를 위한 메타데이터

데이터 웨어하우스를 비롯한 응용분야에서 요약 데이터의 비중이 폭증함에 따라, 사용자 응용 프로그램과 DBMS사이에서 새로운 소프트웨어 계층이 탄생하게 되었다. 이러한 새로운 소프트웨어 계층을 요약 데이터 탐색기라고 한다.

그림 7에서와 같이 요약 데이터 탐색기는 사용자와 실제 데이터베이스 사이에서 교량의 역할을 한다. 요약 데이터 탐색기의 주기능은 사용자의 기본 SQL문을 적합한 수준의 요약 데이터들을 사용한 SQL문으로 번역하는 것이다. 데이터베이스에서 얻어진 결과는 다시 요약 데이터 탐색기를 통하여 사용자에게 보내진다.

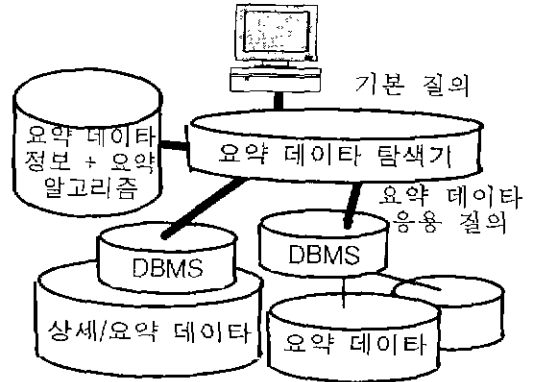


그림 7 요약 데이터 탐색기 구조

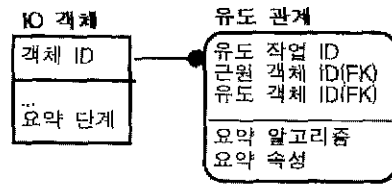


그림 8 다단계 요약 데이터의 유도 관계를 표현하는 골격 스키마

요약 데이터 탐색기는 다단계 요약 데이터를 갖는 데이터 웨어하우스 환경의 필수적인 구성 요소이다.<sup>12)</sup> 요약 데이터 탐색기를 위한 메타 데이터는 기존 카탈로그 정보 외에도, 원천 데이터와 요약 데이터 사이에 유도 관계를 표현하는 정보를 가지고 있어야 한다.<sup>13)</sup> 그림 8은 개체간의 유도 관계를 저장하기 위한 골격 스키마이다.

- 요약 단계-개체에 포함된 데이터의 요약된 정도를 식별하는 데이터이다.
- 요약 알고리즘-다수의 원천 데이터들에서 유도 데이터를 얻는 과정에 사용되는 데이터 변환 알고리즘을 가리킨다. 예를 SUM, MAX, MIN, COUNT, AVG와 같은 집계

12) 요약 데이터 탐색기는 요약 데이터들의 민화가 사용자 응용 프로그램에 미칠 수 있는 영향력을 제거해주어서, 관리자가 응용 프로그램에 대해 고려하지 않고도 임의로 요약 데이터들을 변경하는 것을 가능하게 해준다.  
 13) 통계 데이터베이스는 마이크로데이터(근원데이터)와 매크로데이터(유도/요약 데이터), 그리고 둘의 관계에 대한 메타데이터로 구성된다. 이러한 모델을 제공하는 메타데이터 모델링 기법이 연구되고 있다[10].

함수들이나 사용자가 정의한 통계함수일 수도 있다.

- 요약 속성-그룹핑(grouping)과 집계(aggregate) 작업에 사용된 속성들과 그 밖의 변수들을 정의한다.
- 사용 패턴-데이터 웨어하우스에 접근하는 패턴을 기록하는 것은, 성능을 최적화하고 튜닝(tuning)하는데 매우 유용하다. 즉, 사용 빈도가 낮은 데이터는 값싼 저장 매체에 옮기고, 요구가 많은 데이터에 대해서는 접근속도의 향상 방향을 강구한다.
- 용량 측정 정보-시간과 자원의 측면에서 질의에 드는 비용을 계산하는 정보이다. 용량 측정 정보에는 줄(row) 수, 증가율, 사용 성장적 특징, 인덱싱 등을 포함하는 것이 유용하다. 최적의 질의 처리 방법을 도출하기 위해 사용된다[8].

### 3.2.3 사용자 지원을 위한 메타데이터

Inmon[3]에 따르면, 새로 구축된 데이터 웨어하우스를 사용하고자 하는 분석자들이 다음과 같은 정보를 알기 원한다고 한다.

- 데이터 웨어하우스가 어떤 테이블들과 속성들, 키 값들을 포함하고 있는가?
- 이러한 데이터들의 출처는 어디인가?
- 그 데이터를 얻기 위해서 사용된 변환 방법(transformation logic)은 무엇인가?
- 이 데이터는 얼마나 자주 재적재(reload)되는가?
- 그 밖에도, 별명(alias), 용어 설명, 입력 질의의 수행 기능 여부 등에 관한 정보가 필요하다.

데이터 웨어하우스는 위의 요구사항을 충족하기 위한 메타데이터 탐색기를 제공한다. 메타데이터 탐색기는, 메타데이터에 대해 브라우징(browsing), 탐색(searching) 등의 기능을 제공한다. 예를 들어 사용자의 원하는 주제(subject)에 대한 스타스키마(star schema)를 그래프 형태로 보여주고, 데이터의 유도 관계 등을 표시한다. 뿐만 아니라, 사용자를 위한 설명(description)과 별명(alias), 코드화/참조(reference) 등을 갖는다.

- 설명(description)-메타데이터는 각 개체

와 릴레이션, 변환함수 등에 대해 업무상의 의미로 풀어서 설명해주는 데이터이다. 이를 통해서, 컴퓨터에 익숙하지 않은 사람도 친숙하게 메타데이터 탐색을 가능하게 해준다.

- 별명(alias)-별명은 데이터 웨어하우스를 보다 더 사용자에게 친숙하게 만들어주는 구실을 한다. 예를 들어, 'F-부품' 대신에 '각 공장(factory)에서 생산되는 부품들'로 질의하는 것이 가능해진다. 또, 서로 다른 부서들이 동일한 데이터에 그 부서 고유의 이름을 붙이고자 할 때 매우 유용하다.
- 코드화 데이터/참조 테이블-데이터를 코드화(encoding)시켰을 경우, 외부 테이블에 참조 데이터를 제공하여 일반적인 해석 내용을 사용자가 알 수 있도록 한다. 참조 테이블의 내용은 코드화된 정보에서 원래의 코드화되지 않는 정보로 복귀 가능함을 보장할 수 있어야 한다.
- 처리시간 추정-데이터 웨어하우스 프로젝트가 실패하는 가장 중요한 이유 중에 하나가 질의의 처리시간이 유효한 시간보다 훨씬 많이 걸렸을 때, 사용자들의 욕구불만에 기인한다. 사용자들이 특정 질의가 필요로 하는 시간을 미리 알 수 있다면, 그러한 시간 지연에 대해 인내심을 가질 수 있을 것이다. 데이터 웨어하우스의 성공을 위해서, 질의가 필요로 하는 자원과 시간의 양에 대한 통계적 정보를 갖는 것이 바람직하다.

## 4. 결론과 연구 방향

지금까지 데이터 웨어하우스의 구조적 특성을 분석하여, 필요한 메타데이터들의 종류와 간단한 골격 스키마를 살펴보았다. 데이터 모델링의 분야가 그러하듯이, 메타데이터 모델링도 일반성 있는 완벽한 표준을 제시하기란 매우 어렵다. 메타데이터 요구사항을 면밀히 검토하고 적합한 메타데이터 모델을 구현하는 일은 결국 개인적 소양에 달려있는 것이다. 다만 본 논문이 메타데이터 모델을 연구하는 길잡이

가 되고, 생각의 범위를 넓혀주는 계기가 되기를 바란다. 본문에서 언급하지 않은 몇 가지 메타데이터에 관한 최근 연구방향은 다음과 같다.

- 메타데이터 저장소-최근의 대부분의 메타데이터 도구나 저장소(repository)는 관계형 구조에 의존한다. 그런데 상이한 여러 원천 데이터와 관련된 메타데이터의 복잡성을 다루기에는 부족하다. 객체지향 기술은 데이터 웨어하우스 구축과 실제 통합된 메타데이터 모델을 위해 보다 더 탄력적인 환경을 제공할 수 있는 잠재적 가능성을 가지고 있다[9]. 그리고, 이벤트 처리방식을 지원하기 위해서 트리거 기능을 보완한 액티브(active) 데이터베이스의 기능의 필요성이 발생한다. 이러한 액티브 관계형 데이터베이스에 적합한 메타데이터 모델링 연구도 병행되어야 할 것이다.
- 데이터 통합(integration)과 메타데이터의 재사용-사내(혹은 조직 내)의 모든 데이터베이스와 데이터 웨어하우스의 통합적 이용을 위해서 메타데이터들의 공통 성향을 찾고 메타데이터의 재사용 기회를 높히려는 연구가 있다[12].
- 메타데이터 탄력성(scalability)-데이터 웨어하우스가 성장함에 따라 발생하는 여러 변화에 대해 탄력적으로 대처할 수 있는 메타데이터 모델이 필요하다. 메타데이터의 탄력성을 보장하는 요구사항을 정의하고, 적합한 모델링 방법을 제시하기 위한 연구가 이루어져야 한다.
- 메타데이터 교환(interchange)을 위한 표준(standard) 표기법-메타데이터는 특징으로 비표준성을 갖는다. 따라서, 메타데이터의 공유를 위해서는 교환을 위한 표준안의 필요성이 대두된다[13].

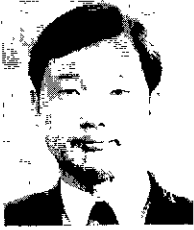
### 참고문헌

- [1] "Briefing Paper : What is Metadata", APT Data Group, 1996.
- [2] R. Kimball, The Data Warehouse Toolkit, John Wiley & Sons. Inc., 1996.
- [3] W.H. Inmon, Building the Data Warehouse 2nd Edition, John Wiley & Sons, Inc., 1996.
- [4] J. L. Wiener, H. Gupta, W. J. Labio, Y. Zhuge, H. Gracia-Molina, and J. Widom, "A System Prototype for Warehouse View Maintenance", Proceedings of the Workshop on Materialized Views : Techniques and Applications(In Cooperation with ACM SIGMOD), pp. 26-33, 1996.
- [5] V. Poe, "Building A Data Warehouse for Decision Support", Prentice Hall, Inc., 1996.
- [6] R. C. Barquin, and H. A. Edelstein, "Planning and Designing the Data Warehouse", Prentice Hall, inc., 1997.
- [7] M.H. Brackett, "The Data Warehouse Challenge Taming Data Chaos", John Wiley & Sons, Inc., 1996.
- [8] A Red Brick Systems White Paper, "Star Schemas and STARjoin Technology".
- [9] S. Sachdeva, "METADATA : Guiding Users Through Disparate Data Layers", Application Development Trends, 1995.
- [10] H.-J. Lenz, "The Conceptual Schema and External Schemata of Metadatabases", Proceedings of the 7th International Working Conference on Scientific and Statistical Database Management, pp. 160-165, 1994.
- [11] A. Gupta, V. Harinarayan, and D. Quass, "Aggregate-Query Processing in Data Warehousing Environments", Proceedings of the 21st VLDB Conference Zurich, Swizerland, pp. 358-369, 1995.
- [12] L. Seligman, A. Rosenthal, "A Metadata Resource to Promote Data Integration", Proceedings of IEEE Metadata Conference, Silver Spring, MD, April

1996.

[13] "Proposal for Version 1.0 Metadata Interchange Specification(MDIS)", the Metadata Interchange Institute, June 7, 1996.

**박 석**



1978 서울대학교 계산통계학과 학사  
 1980 한국과학기술원 전산학과 석사  
 1983 한국과학기술원 전산학과 박사  
 1983~현재 서강대 전자계산학과 교수  
 1989~1991 University of Virginia 방문교수  
 1996~현재 한국정보과학회 데이터베이스 연구회 운영위원장

관심분야: 실시간 데이터베이스, 보안 데이터베이스, 멀티미디어 데이터베이스, 트랜잭션 관리, 데이터 웨어하우스, 디지털 도서관

**김 회 원**



1996 서강대학교 전자계산학과 학사  
 1996~현재 서강대학교 대학원 전자계산학과 석사  
 관심분야: 병행수행 제어, 실시간 데이터베이스, 데이터 웨어하우스, Materialized View

**김 말 희**



1996 서강대학교 전자계산학과 학사  
 1996~현재 서강대학교 대학원 전자계산학과 석사  
 관심분야: 병행수행 제어, 실시간 데이터베이스, 데이터 웨어하우스

● '97 데이터베이스 춘계튜토리얼 ●

- 일 자 : 1997년 5월 22(목)~23일(금)
- 장 소 : 과학기술회관
- 주 최 : 데이터베이스연구회
- 문 의 처 : 서강대학교 전자계산학과 박 석 교수  
T. 02-705-8487