1 자료설계 작업의 함정과 탈출구

문송천
한국과학기술원
정보 및 통신공학과 교수
KBS 정보화 자문위원

자료설계 작업은 전산화에 있어서 소위 데이터베이스 설계 혹은 Data Modelling(DM)이라고 부르는 것을 가리킨다.

자료설계라고 하면 자료를 설계 한다는 것인데, 우리가 '프로그램을 짠다'고 하면서 자료(Data)는 짠다고 하지 않는 관행은 어디에서 비롯되었으며, 그 근본맹점이 무엇이었던가를 짚어보지 않으면 아니된다.

이는 1980년대 이후에 불어닥친 자료중시 사상의 관점에서 조명해 볼 수 있다. 프로그램이 더 중요하냐 아니면 데이터가 더 중요하냐는 쟁점으로 봐도 무방하다. 지금이 관계형 데이터베이스(Relational Database: RDB)시대라서 많은 이들이 관계형 데이터베이스(Relational Database: RDB) 설계에 관심을 갖고 있지만 사실은 자료중시 사상의 극치는 객체지향 데이터베이스에서 발현된다.

그런데 관계형 데이터베이스(Relational Database : RDB)자료 설계시에 실무자들이 부딪히는 어려움이랄까 장벽은 실로 간과하기에는 너무나 벅찬감이 있는 바, 여기에서는 이러한 쟁점에 관련된 핵심 현안들을 짚어보고 대책을 제시해 보고자 한다.

Data Modelling의 과정은 초반부의 개략설계와 후반부의 상세설계로 구성된다. 개략설계를 전문용어로는 개념적 설계(Conceptual Design: CD)라고 부르며, 상세설계는 논리가 해법(Algorithm)형태로 주입되어 실현된다고 해서 논리적

설계(Logical Design: LD)혹은 구현설계(Implementation Design: ID)라고 부른다. 데이터베이스화의 관점에서 Data Modelling 이전 단계를 요구형성/분석단계라고 칭하는데, 이는 전산화 하고자 하는 업무에 관한 처리규정의 명확한 도출작업 과정이다. 업무처리 규정은 1, 2, 3,...등으로 조목조목 명문화된 형식으로 정리될 수 있겠다.

업무처리 규정이 자료모형 형태로 표현된 모양이 소위 개체관계모형(Entity Relationship Model: ER Model)이다. 이러한 의미살리기 형태의 모형을 전문용어로는 의미모형(Semantic Model)이라고 칭하는데 개체관계모형(Entity Relationship Model)은 1976년 피터 첸의 전산학 박사 학위 논문에서 처음 제안된 이래 지금까지 수없이 확장되어 온 자료모형으로 확고 부동한 자리를 차지하고 있으며, 지금도 개체관계모형(Entity Relationship Model)을 주제로 매년 국제학술 행사가 치러지고 있을 정도로 사용자 단체가 두텁게 형성된 기반도 가지고 있다.

따라서 개념적 설계(Conceptual Design: CD)단계의 관심은 업무처리 규정을 개체관계모형(Entity Relationship Model)으로 변환하는 작업이다. 개체관계모형(Entity Relationship Model)을 문장형태로 나타내는 것보다는 그림형태로 나타내는 것이 데이터베이스화 대상 데이터들을 일목요연하게 볼 수 있겠으므로 개체관계도(Entity Relationship Diagram: ERD)가 실제로 활용되는 근거가 여기에 있다. 따라서 개념적 설계(Conceptual design)를 제대로 완수 하였느냐의 여부는 업무처리 규정에 도입된 각 조목이 개체관계도(Entity Relationship Diagram: ERD)의 그림 어느 한 부분에 상응되느냐 여부로 판가름될 수 있다.

조목마다 제대로 개체관계도(Entity Relationship Diagram: ERD)에 사상되었느냐를 점검하는 것이 설계자가 반드시 중간 검증해야 할 사안인 것이다. 만약 업무처리 규정이 개체관계도(Entity Relationship Diagram: ERD)로 대응되는 과정에서 사상의 불충분성, 즉 사상 자체에 빈 구멍이 생긴것을 발견했을때에는 즉각 업무처리 규정과 개체관계도(Entity Relationship Diagram: ERD) 양자간 양방향 대응관계를 명확히 짚고 넘어가는 작업을 반복적으로 시행될 수도 있다.

개념적 설계(Conceptual Design: CD) 단계에서 개체관계도(Entity Relationship Diagram: ERD)의 유용성을 잘모르겠다거나 개체관계도(Entity Relationship Diagram: ERD)를 아직도 사용해 보지 않았다거나 개체관계도(Entity Relationship Diagram: ERD)의 존재는 들어서 이미 알고 있으나 현업에서 지금까지도 활용은 안해보았다는 독자가 있다면 이 기회에 개체관계도(Entity Relationship Diagram: ERD)가 왜 필요한가를 반드시 파악하고 넘어가기 바란다. 개체관계도(Entity Relationship Diagram: ERD)의 유용성은 다음 두가지 관점에서 조명할 수 있다.

첫째, 문장형식의 업무처리 규정을 그림형태의 무엇인가 흘러가고 흘러들어오는 모양으로 총정리 한다는 의미에서 대단한 수준의 문서로서 충분한 가치가 있다.

둘째, 데이터베이스 구현으로 들어가는 논리적 설계(Logical Design: LD) 혹은 구현설계(Implementation Design: ID) 단계에서 반드시 필요로 하는 정규화(Normalization)된 테이블(혹은 표)를 만들기 위한 시발 근거자료로 개체관계도(Entity Relationship Diagram: ERD)가 요구된다는 것이다.

논리적 설계(Logical De-

sign: LD)단계의 관심사는 개체 관계도(Entity Relationship Diagram: ERD)의 각 부분인 개체(Entity)나 관계(Relationship)를 테이블 모양으로 변환할때 개체 하나가 테이블 하나로 바뀌느냐 아니면 두개로 바뀌느냐 하는 사안이다. 이는 관계 하나 하나에 대해서도 마찬가지이다. 개체하나를 표 하나로 1:1로 사상하더라도 표준화, 즉 정규화된 형태의 표가 나오면 좋은데 만약 1:1로 했더니 정규화 기준과는 거리가 있는 모양으로 귀결된다면 1:2 혹은 더 나아가서 1:3이 될 것은 필연적이다.

도대체 이와 같은 쪼개기 에 관한 근본기준이 무엇이느냐가 거론되게 마련이다. 이 기준설정을 위하여 개체 하나(혹은 관계 하나)에 소속되어 있는 속성들이 여러개 있을경우, 이 여러개 속성들간의 강약관계를 설정하는 것이 유용하다. 강약의 의미는 속성 A가 속성 B를 향하여 얼마만한 영향력을 행사하느냐는 차원에서 해석한다. 즉 B의 입장에서 볼때에는 B가 A에 얼마나 종속되느냐 하는 정도 차원에서 해석한다. 이러한 영향력 행사 관계 혹은 결정력 수용관계를 전문용어로는 종속성(Dependency)라고 부른다. 예를들어 "A가 B를 결정한다."고 부를때 즉 "B가 A에 종속한다."고 일컫을 때 이 호칭의 깊은 뜻은 A의 값 하나가 B의 값 하나만을 사상한다는 사실을 나타내는 것이다.

이 경우 'A → B'로 표기하는 것이 통상적 약속인데 A값 하나에 B값이 하나뿐이라면 역시 1:1관계로서 함수에서 종속변수 값 하나에 대응하는 독립변수 값도 역시 하나라는 관찰에서 'A→B' 형태의 표기에 대해 A는 B를 함수적으로 결정한다.(또는 B는 A에 함수적으로 종속한다.)고 칭할 수 있겠다. 따라서 이러한 종속성을 함수적 종속성(Functional Dependency:FD)이라 일컫는다.

개체하나에 대하여 파악하는 일은 중요하다. 더우기 중요한 일은 데이터베이스 설계자가 반드시 도입하고자 하는 함수적 종속성(Functional Dependency: FD)의 설정이다.

즉 포기 불가능한 함수적 종속성(Functional Dependency: FD)들만 남기고 다른 FD들은 과감하게 버릴 수도 있다는 말이다. 관계하나에 대해서 성립하는 함수적 종속성(Functional Dependency: FD)들을 파악하는 일도 역시 중요하다. 이 함수적 종속성(Functional Dependency: FD)들을 기준으로 해서만이 개체 대표 사상이 1:1이 될지 1:2가 될지가 결판난다는 점에서 함수적 종속성(Functional Dependency: FD)의 결정에 대해서 대단히 신중해야만 한다. 사실 함수적 종속성(Functional De-pendency: FD)라는 거울에 비추어 1:1 또는 1:2로 사상되는 결과가 어떤 모양의 표들의 최종적으로 나오느냐에 관한 전부를 결정짓게 되는데, 이러한 관점을 소위 정규화(표준화 혹은 분해화)라고 호칭하는 것이다.

정규화라고 부르는 배경에는 가장 세련된 정형화된 모습이라는 뜻도 담겨있고, 표준화라고 부르는 배경에는 1차 정규형 혹은 3차 정규형(Third Normal Form: 3NF)등의 표준화된 모습이 여러개 있을 수 있다는 뜻도 담겨 있으며, 분해화라고 부르는 배경에는 1:1로만 항상 사상되는 경우 뿐만 아니라 1:2로 되듯이 두개의 표로 쪼개지는 모양으로 결과지워 진다는 뜻도 담겨 있다.

마지막으로 설계상의 함정에 대해 한가지만 더 짚고 넘어가자. 3차 정규형(Third Normal Form: 3NF)가 가장 바람직한 정규형으로 알려져 있는 바, 1차 → 2차 → 3차 식으로 거쳐가는 어리석고 시간소모적인 변환이 아니라 직통으로 구해내는 이른바 Non-Stop Third Normal Form(3NF) 를 터득하지 않고는 애로사항이 많다는 점이다. DC