

커뮤니티 구축 시 모델링이 주요 요소

데이터모델링 설계는 현실을 잘 반영하게

이현호 교수 / 안양과학대학 컴퓨터정보학부

연재 순서

- 1 웹환경에서 더욱 위력적인 Oracle8i의 새로운 기능 (이번호)
- 2 웹 커뮤니티 시스템 모델링(1)
- 3 웹 커뮤니티 시스템 모델링(2)
- 4 Oracle8i의 새로운 기능을 이용한 웹 커뮤니티 시스템 구현(1)
- 5 Oracle8i의 새로운 기능을 이용한 웹 커뮤니티 시스템 구현(2)

지난 호에서는 상업적인 대규모 웹 커뮤니티 시스템에서의 안정적인 DBMS 확보의 당위성을 설명하였다. 그리고, 그러한 DBMS 중 하나인 오라클, 그 중에서도 효율적인 커뮤니티 시스템의 구현을 위해서 반드시 필요한 오라클 8i의 새로운 기능인 analytic function/function-based index/materialized view에 대해서 간략하게 설명하였다.

이번 호와 다음 호에서는 웹 커뮤니티 시스템 구축에 있어서 성공의 열쇠라고 해도 과언이 아닐 시스템의 설계, 그 중에서도 커뮤니티 시스템 DB의 모델링의 측면을 설명하고, 실제 몇 가지 주요한 요구사항을 가정하여 어떻게 시스템이 설계되어야 바람직할지를 직접 시스템의 설계하며 논의해 보겠다.

인터넷의 사용형태는 사이트 운영자측의 일방적인 정보제공(일방향)에서 사용자측과 운영자측의 상호작용(양방향)으로, 이것은 다시 사용자들 간의 정보교환(커뮤니티) 형태로 발전해 왔다. 그러므로, 운영자측의 역할은 단순한 정보제공자로서의 역할에서 커뮤니티의 장을 제공하는 역할까지 포함한 포괄적이고 다양한 형태로 발전하였다.

즉, 어떻게 하면 사이트 방문자들끼리의 공동체를 활성화시킴으로써 사이트의 사용자 장악력을 키울 것인가 하는 것이 인터넷 온라인 서비스의 주요 관심사가 되었다. 그 대표적인 예가 동호회 혹은 카페로 불리우는 커뮤니티(community)이다.

커뮤니티의 전형적인 사례가 있을지라도 그 형태는 사이트마다 조금씩 차이가 있다. 불특정 다수로 이루어진 사용자들의 특성상, 커뮤니티가 올바른 방향으로 자리매김하기 위해서 여러 가지 규칙들이 존재하고 이것은 커뮤니티의 목적과 성격에 따라 약간의 차이를 보인다.

그러므로, 커뮤니티를 일반 게시판의 강화된 형태로만 파악하는 것은 큰 오산이다. 커뮤니티에서 운영자측의 역할은 축소되어야 하는 것처럼 보이지만 사이트의 안정적 운영, 건정성 보장, 커뮤니티의 원래 취지와 목적에로의 유도 등 여러 가지 눈이 보이지 않는 역할들을 훌륭하게 수행하지 못한다면 커뮤니티의 생명력은 단축될 수밖에 없다.

이러한 점을 비추어볼 때, 운영자측의 커뮤니티 구축에 있어서 가장 중요한 부분은 바로 모델링이다. 즉, 구축하려는 커뮤니티의 목적과 성격에 맞게 룰(rule)을 세우고 이것을 효과적이고도 융통성 있게 운영하기 위해서는 올바른 데이터모델링이 필수적이다. 결국, 온라인 커뮤니티의 물리적인 존재형태는 데이터이기 때문이다.

이것은 오히려 개발도구나 환경을 어떻게 가지고 갈 것이냐 보다 훨씬 더 중요할 수 있다. 이러한 점을 간과하여 엉성하게 데이터 설계를 했다가 운영이나 유지보수에 낭패를 보는 일은 허다하다.

본고에서는 관계형(relational) 데이터모델에 기반하여, 커뮤니티 설계에 있어서 올바른 데이터모델링기법에 대해서 간략하게 소개하고자 한다. 우선, 커뮤니티 구축의 요구사항을 가정하고 이러한 요구사항을 효과적으로 구현하기 위한 데이터모델링은 어떠한 식으로 되어야 할 것인지에 대해서 서술하겠다.

참고적으로, 본고는 관계형 데이터베이스 모델에 대한 기초적인 이해를 가진 독자를 대상으로 작성하였고 본 글에서 논의되는 데이터모델링기법은 (주)엔코어정보컨설팅이 보유한 기술을 바탕으로 한 것이다. 모델링기법은 매우 다양한 요소를 포함하고 있다. 그러므로, 커뮤니티 시스템을 모델링하면서 모델링기법의 전부를 다룰 수는 없다.

또한, 지면의 한계상 모델링기법 하나하나에 충실한 설명이 안 될 수도 있음을 미리 밝힌다. 본 글의 바탕이 되는 모델링기법의 전반적인 내용을 알아보고 싶거나 상세 설명이 필요한 독자는 엔코어 홈페이지(www.en-core.com)를 참조하거나 엔코어 주최 모델링교육을 받아보기를 권한다.

위와 같은 동호회에 기반을 둔 커뮤니티 요구사항에 대한 가정 하에 올바른 데이터모델링에 대해 논의해 보자. 논의에 들어가기에 앞서, 앞으로 빈번히 사용하게 될 몇 가지 모델링 용어에 대해서 먼저 살펴보자.

● 엔터티의 UID(Unique Identifier)

엔터티를 이루는 개체(tuple)들 간을 identify하는 대표적인 key attributes 즉, primary key를 말한다.

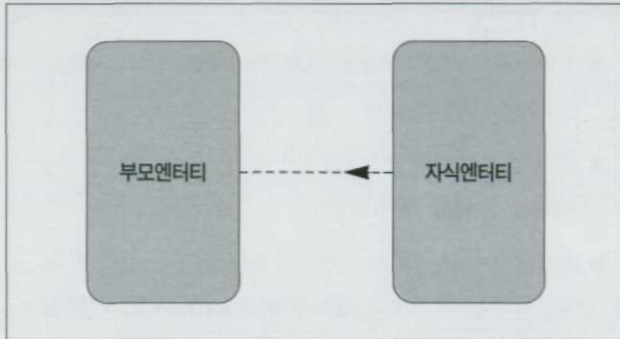
● 부모 엔터티(parent entity)와 자식 엔터티(entity)

Entity-Relationship Diagram(ERD) 상의 엔터티와 엔터티 사이의 대응수(cardinality)는 1:1, 1:M, M:M 관계로 분류할 수 있다. 1:1 관계는 엔터티의 수직분할에 의해서 탄생할 수는 있지만 잘 나타나지 않으며, M:M 관계도 교차엔터티를 탄생시켜 해소시키기 때문에 최종 ERD 상에는 나타나지 않는다. 그러므로, 대부분의 엔터티는 1:M 관계로 정의되며 이 때, 1쪽 엔터티를 부모 엔터티라고 하고 M쪽 엔터티를 자식 엔터티라고 한다.

웹 커뮤니티 요구사항(가정)

- 커뮤니티는 사이트 회원들의 이해와 요구에 의해서 생성된 동호회를 기반으로 한다.
- 사이트에 가입한 회원이 동호회 만들기 신청을 하면 일정심의 거쳐 허가를 내어준다.
- 동호회는 게시판 형태로 운영되며 동호회 소개정보와 각 게시판 소개정보가 동호회 관리자에 의해 html문서로 작성가능하다.
- 게시판은 글번호, 제목, 글쓴이, 작성일시, 조회수, 글내용으로 구성되어 있다. 글쓴이는 회원ID로 올린다. 작성일시는 당일 올린 글은 시각을 표시하고 하루가 지난 글을 날짜를 표시한다.
- 동호회 관리자는 동호회 내의 여러가지 게시판을 생성/삭제할 수 있는 권한이 있고 부적절한 내용에 대한 글삭제 권한과 회원강제추방 권한도 가지고 있다. 또한, 게시판을 외부에 공개할 것인지 아닌지에 대한 결정 권한 관리도 한다. 동호회 관리자는 여러 명 일 수 있다.
- 동호회 회원이 되려면 우선 사이트 회원으로 가입한 후 동호회 회원가입을 다시 해야 한다. 동호회 회원가입시 아이디는 새로 부여받지 않고 사이트 아이디를 그대로 사용한다. 단, 동호회 활동에 필요한 추가정보를 더 입력받을 수는 있다.
- 동호회 회원은 해당 동호회의 모든 게시판에 글읽기와 자기 글에 대한 수정, 삭제 권한을 가진다. 또한, 게시판의 권한종류를 제목읽기권한, 내용읽기권한, 읽기권한, 쓰기권한 등으로 세분화하여 가입하지 않은 일반회원들에게도 둘러보기 차원의 정보를 제공할 수 있다.
- 사이트에 가입한 회원 모두에게 마일리지가 부여되고 관리된다. 동호회에서의 활동도 마일리지 가산에 적용되는 사항이며, 개인 마일리지 또는 동호회 회원 마일리지 총합에 따라 개인 또는 동호회에 혜택이 부여된다.
- 사이트 운영자는 동호회를 분류별로 관리하며 불건전한 동호회에 관한 폐쇄 권한을 가지고 있다. 또한, 동호회 회원수와 월별 글등록수 그리고, 글의 총조회수 등의 통계정보를 생성관리하여 일정한 규칙에 따라 우수 동호회를 선정하여 발표하고 이벤트행사 시 참고자료로 활용한다. 또한, 동호회 회원에게 분류에 따른 관련 동호회를 소개를 한다.

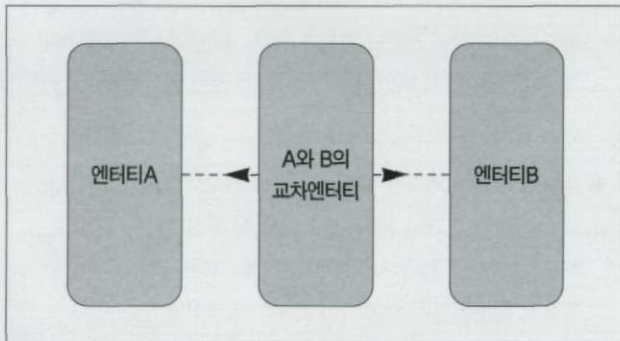
<그림 1>



● 교차엔터티(interaction entity)

위에서도 설명했듯이, 엔터티들 간의 M:M 관계를 해소하기 위하여 양쪽 엔터티의 UID를 받아서 탄생한 엔터티를 교차엔터티라고 한다.

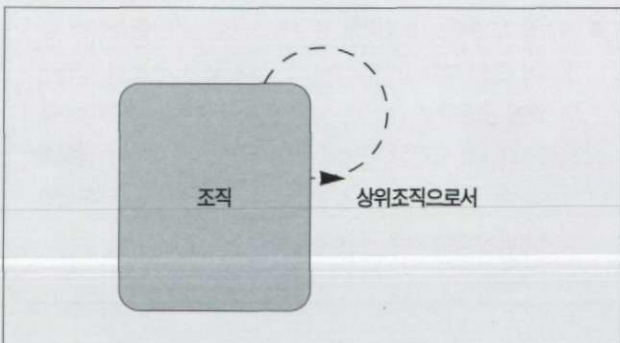
<그림 2>



● 순환관계(recursive relationship)

한 엔터티 내에서 이루어지는 속성(attribute) 들 간의 관계를 말한다. 예를 들어, 계층적 구조를 가지는 조직 엔터티를 정의할 때, 조직 엔터티 내의 '조직ID' 속성과 '상위조직ID' 속성은 M:1 관계를 가진다. 이 때, 조직ID와 상위조직ID는 순환관계에 있다고 하며, ERD 상에 다음과 같이 표현된다.

<그림 3>



이제, 커뮤니티 시스템의 데이터모델링에 대해 본격적으로 논의해 보자.

1. 엔터티(Entity)의 선정

데이터모델링은 실세계(real world)를 가장 잘 표현하는 방향으로 진행되어야 함은 두말할 나위가 없다. 그러므로, 처음부터 물리적인 개발환경을 고려할 필요는 없다. 이것은 자칫 실세계를 표현하는데 왜곡을 가져올 수 있기 때문이다.

데이터모델링은 크게 논리적 데이터모델링과 물리적 데이터모델링의 단계를 거치는데, 논리적 데이터모델링의 단계에서는 물리적인 개발환경의 고려라는 측면은 제외되며 이 단계의 가장 큰 목표는 요구사항에 맞게 현실세계를 가장 잘 표현하는데 있다.

논리적 데이터모델링의 첫 단계는 엔터티(entity)를 선정하는 것이다. 엔터티가 명확하게 정의되지 않으면 엔터티가 관리하는 데이터의 성격이 모호해지며 관계(relationship)의 설정도 모호해져서 결국은 이상한 잡동사니 집합으로 변질될 것이다. 이것이 데이터 유지보수를 어렵게 함은 두말할 나위가 없다. 엔터티는 우리가 관리하고자 하는 두 개 이상의 속성과 두 개 이상의 개체를 가지며 동질성을 지닌 독립적인 집합이다.

'우리가 관리하고자 하는' 은 엔터티가 모델링을 통해 표현하고자 하는 관심영역의 범주에 속해 있어야 함을 말하고, 동질성은 엔터티를 구성하는 개체들 간의 데이터 성격은 동일해야 함을 말하며, 독립적이라 함은 엔터티가 다른 엔터티와 명확하게 구분되는 독자적인 데이터 영역으로 존재해야 함을 말한다.

엔터티를 처음부터 명확화하기는 힘들다. 그러므로, 각종 관련 자료나 구시스템, 또는 실무자와의 인터뷰를 통해서 엔터티가 될 만한 후보들을 추리고 하나하나 검토해 가면서 엔터티를 선정하는 것이 바람직하다. 엔터티 후보들은 보통 명사형인 단어들이 대상이 된다. 그러면, 위의 커뮤니티 요구사항을 가지고 엔터티의 정의에 비추어 엔터티의 후보가 될 만한 것부터 나열해 보자.

- 커뮤니티 · 사이트 · 회원 · 동호회 · 신청 · 허가 · 게시판 · 소개 정보 · 글번호 · 제목 · 글쓴이 · 작성일시 · 조회수 · 글내용 · 관리자 · 권한 · 외부 · 동호회원 · 아이디 · 추가정보 · 둘러보기 · 마이리지 · 혜택 · 폐쇄 · 회원수 · 글등록수 · 총조회수 · 통계정보 · 우수동호회 · 동호회분류 · 관련동호회

우선, 커뮤니티나 사이트는 구축될 시스템의 도메인(domain)이지 엔터티로서 관리되어야 할 성질의 것은 아니다. 엔터티의 정의에 비추어 명확하게 엔터티로 분류할 수 있는 것은 회원, 동호회, 게시판, 통계정보 정도일 것이다.

이들은 두 개 이상의 속성과 개체가 존재하며 개체 간의 동질성과 다른 엔터티와 구별되는 명확한 데이터적 특성이 존재한다. 동호회나 게시판의 소개정보는 html문서로 관리되므로 동호회나 게시판 엔터티의 속성으로 보는 것이 타당하다.

글번호, 제목, 글쓴이, 작성일시, 조회수, 글내용도 이들 각각이 원자값(atomic value)으로서 하나의 속성이자 이들 자체 내에 여러 가지(두개 이상의) 정보들이 존재하지는 않는다. 회원수, 글등록수, 총 조회수도 통계정보의 속성이다. 그럼, 좀 더 모호한 것들을 하나씩 살펴보자.

신청은 동호회 만들기 신청을 말하는 것으로 구축할 시스템이 신청내역을 데이터로서 관리를 하고자 한다면, 엔터티 자격이 충분하다. 신청내역과 실제로 승인되어 만들어진 동호회와는 관리될 데이터의 성격이 분명 구별된다.

허가도 신청과 비슷한 성격을 지닌다. 그러나, 허가가 되면 동호회가 만들어지고 만들어진 동호회에 관한 데이터는 동호회 엔터티에서 관리될 것이므로 독립적인 엔터티로 굳이 관리되어질 필요는 없을 것으로 판단된다. 단, 허가일시라든가 조건부 허가의 내용 등을 관리하고 싶다면 엔터티로 만들어야 할 것이다.

동호회 관리자는 분명 두 개 이상의 속성과 개체를 가질 수 있으나, 넓은 의미에서는 동호회원이며 오히려 권한에 따른 동호회원의 한 종류로 보는 것이 맞다. 그러므로, 동호회원의 한 서브타입(subtype)으로 보는 것이 타당하다.

권한은 여러 가지 권한내역이 존재한다면, 엔터티로서의 자격이 있다. 그러나, 이에 대해서 조금 체계적으로 접근할 필요가 있다. 회원 하나하나에게 각기 다른 권한을 준다는 것은 무의미하고 현실성이 없으므로 회원을 여러 등급으로 나누고 등급에 따른 권한 내역을 엔터티로 관리하는 형식이 타당할 것이다.

외부는 회원이 아닌 사용자 전체를 가리키는 것으로 그 의미가 너무 추상적이다. 또한, 커뮤니티 시스템이 관리하고자 하는 대상은 아니다. 그러므로, 엔터티로는 부적절하다.

동호회원은 어떤가? 이것을 회원의 한 서브타입(subtype)으로 볼 수 있을까? 조그만 더 곰곰히 생각하면, 사이트 회원과 동호회원은 분명 다르다는 것을 알 것이다. 한 사이트 회원이 여러 동호회의 동호회원이 될 수 있다.

이것은 곧 사이트 회원은 사이트 가입과 동시에 독립적으로 발생하는 개체집합으로서의 엔터티이지만 동호회원은 회원과 동호회가 만나서야 의미가 부여되는 행위집합으로서의 엔터티임을 의미한다. 그러므로, 동호회원은 사이트 회원과 구별되는 독립적인 엔터티로서의 자격이 충분하다.

추가정보는 동호회마다 독자적으로 관리하는 추가적인 동호회원정보를 의미하는데, 이는 개념적으로는 동호회원 엔터티의 속성이자 엔터티로 보기는 힘들다. 그러나, 데이터의 성격상 상세 데이터모델링 단계에서 엔터티로 분리되어 나올 수는 있을 것이다. 이는 추후 다시 설명한다.

둘러보기는 외부 사용자의 한 기능을 의미하는 단어이지 엔터티의 대상이 될 수는 없다.

마일리지는 사용자의 마일리지 포인트를 가리키는 속성이다. 혜택은 마일리지에 따른 혜택을 뜻하는 것으로 데이터로 관리할 것인가? 하는 것은 선택의 문제이다. 데이터로 관리하고자 한다면, 마일리지와 혜택을 묶어서 마일리지 혜택정보라는 집합을 상정해 볼 수 있다.

폐쇄는 동호회 폐쇄를 뜻하는 것으로 폐쇄 유무만 관리하지 않고 폐쇄일시, 사유 등을 관리하려했다면 엔터티로서의 자격이 있다. 그러나, 그렇다 하더라도 폐쇄 동호회는 동호회의 한 서브타입(subtype)으로 보는 것이 더 타당하다.

우수동호회는 동호회의 한 속성으로 보는 것이 타당하며, 관련 동호회는 독자적인 엔터티라기 보다는 순환관계(recursive relationship)로 보는 것이 타당하다. (단, M:M 순환관계이기 때문에 상세 데이터모델링 단계에서 교차엔터티로 떨어져 나온다. 이를 BOM구조라고 한다.)

동호회분류의 경우 분류정보 자체는 동호회 자체정보하고는 성격이 다른 독립적 데이터 성격을 가진다. 또한, 분류체계가 다단계로 정의된다면, 두 개 이상의 속성이 필요하기 때문에 독립적인 엔터티로 구분할 수 있다.

이상으로 각 엔터티 후보들을 대상으로 엔터티 자격검증을 간단하게 해 보았다. 필자가 '엔터티로 보는 것이 타당하다'라는 표현을 자주 사용하였는데, 이는 모델링에는 정답이 없기 때문이다. 이는 그만큼 모델링이 쉽지 않다는 것을 반증하는 표현이다. 어쨌든, 검증된 내용으로 엔터티를 추출하여 나열해 본다면, 대체로 다음과 같다.

회원, 동호회, 신청, 게시판, 회원등급, 권한, 동호회원, 마일리지 혜택정보, 통계정보, 동호회분류

2. 엔터티(Entity)의 분류

엔터티는 성격에 따라 크게 키(key) 엔터티, 메인(main) 엔터티, 액션(action) 엔터티로 분류할 수 있다.

키 엔터티는 원래부터 존재하는 실체로서 모든 데이터를 발생시키는 주체로서의 역할을 한다. 즉, 모든 데이터는 키 엔터티로부터 파생된다. 그러므로, 데이터모델링에서 키 엔터티를 분류해 내는 것은 아주 중요한 의미를 지닌다.

키 엔터티에 따라서 전체적인 모델링 양상이 달라질 수 있기 때문이다. 키 엔터티도 대체로 두 가지 부류로 나누어서 생각해 볼 수 있다. 하나는 자식 엔터티를 많이 거느리면서 시스템의 핵심이 되는 엔터티 즉, 그 엔터티 없이는 시스템 자체가 정의될 수 없는 엔터티이다.

본 커뮤니티 시스템에서는 회원 엔터티와 동호회 엔터티가 여기에 속한다. 커뮤니티 시스템을 이루는 두 가지 없어서는 안 될 중요한 요소는 회원과 동호회이다. 커뮤니티 시스템에서 회원없는 동호회나 동호회 없는 회원은 있을 수 없기 때문이다.

키 엔터티의 또 한가지 부류는 코드성 엔터티이다. 이 엔터티는 키 엔터티의 성격을 가지고는 있으나 주요 엔터티들에 대해 간접 종속관계의 부모 엔터티의 역할을 한다. 그러므로, 종속관계가 아주 단편적이고 단순하다. 본 커뮤니티 시스템에서는 마일리지, 회원등급, 권한, 동호회분류 등이 그것이다.

이와 같은 코드성 엔터티들은 키 엔터티의 단순화 원칙에 의해서 보통 통합하여 하나의 엔터티로 관리하는 것이 여러 면에서 유리하다. 단순화 방안에 대해서는 뒤에서 다시 설명하겠다.

본 커뮤니티 시스템에서는 동호회원, 게시판HEADER 엔터티가 여기에 속한다. 동호회원은 키 엔터티인 회원과 동호회 엔터티를 통해 태어났지만, 이 자체가 시스템 요구사항에 따라 많은 자손을 거느릴 수 있다. 게시판HEADER 엔터티는 게시판 자체의 정보를 담는 엔터티로서, 동호회 엔터티로부터 태어났지만 게시판BODY 엔터티 등 요구사항에 따라 많은 자손을 거느릴 수 있다. 게시판 모델링은 뒤에서 좀 더 상세히 설명하겠다.

액션 엔터티는 자주 변경되고 지속적으로 데이터가 증가하는 성격을 띠는 트랜잭션(transaction) 데이터를 주로 저장한다. 이 엔터티는 반드시 부모를 가진다.(때에 따라서는 많은 부모를 가진다.) 이는 꼭 부모 없는 자식은 없다라는 논리와 비슷하다. 실제 시스템이 관리하는 데이터량의 절대적인 부분을 액션 엔터티가 차지한다. 본 커뮤니티 시스템에서는 동호회신청내역, 게시판

BODY 엔터티가 여기에 속한다.

동호회 신청은 빈도수는 낮더라도 회원을 통하여 끊임없이 이루어지는 성격을 가지고 있으며, 게시판BODY 엔터티는 실제로 게시판의 글리스트를 관리하는 엔터티로서 게시판에서 글쓰기/수정/삭제는 지속적으로도 방대하게 일어난다. 그러나, 동호회신청내역이나 게시판BODY 엔터티를 봐도 알 수 있듯이 액션 엔터티가 자식 엔터티를 가지는 경우는 드물다. 즉, 액션 엔터티는 엔터티 계보의 가장 말단에 있으며 실제 업무 처리의 내용을 볼 수 있는 엔터티인 것이다.

이와 같이, 엔터티를 분류할 수 있으려면 각 엔터티의 성격을 명확히 규정할 수 있어야 한다. 엔터티의 성격을 명확히 규정할 수 있는 가장 좋은 방법은 엔터티의 의미상 주어(명사)를 규정하는 것이다.

의미상 주어라 함은 엔터티를 탄생시키는데 직접적인 역할을 한 핵심 속성들의 집합을 말한다. 즉, 논리적으로 엔터티를 구성하는 각 개체들을 구별짓는(identify) 기준이 되는 키 속성(key attribute)들의 집합이다. 의미상 주어는 논리적으로 엔터티의 주어이며 이를 통하여 엔터티의 성격을 명확히 하는 결정적인 단서 역할을 한다.

예를 들어, 회원 엔터티에서의 주민번호, 동호회에서의 동호회명 등이 그것이다. 한가지 고려해야 할 것은 인조키(artificial UID)하고는 구별되어야 한다는 것이다. 또한, 게시판HEADER 엔터티에서 게시판ID 라는 속성을 정의하여 UID로 사용할 수 있지만, 이 엔터티의 의미상주어는 '동호회ID+게시판명' 이 된다는 것이다.

이렇게 인조키를 사용하는 이유는 자식 엔터티가 부모 엔터티의 UID를 상속받음으로써 발생하는 UID의 복잡성을 제거하기 위함이다. (보통 자식 엔터티의 UID는 '부모 엔터티의 UID+자신의 UID'로 정의됨) 그러나, 인조키의 값 자체는 해당 개체의 유일성을 보장한다는 것 이외에 아무런 의미가 없다.

다음 호에서는, 지금까지의 논의를 통해 어느 정도 분류된 엔터티들을 가지고 몇 가지 영역으로 나누어 부분적 모델링 이슈(issue)들에 대해 설명하고 전체적인 커뮤니티 시스템 ERD와 모델링 결과 테이블을 제시해 보겠다. ☞

참고문헌 및 참고사이트

- (1) DATA MODELING & DATABASE DESIGN, DB 모델링 교육용 교재, 엔코아정보컨설팅, 1997.
- (2) 엔코아정보컨설팅 홈페이지(www.en-core.com) 솔루션웨어하우스 (solution warehouse)