뷰의 개념을 이용한 다단계 보안 관계 데이타베이스 시스템 구현

조완수* 배해영
국방정보체계연구소 인하대학교 전자계산공학과

An Implementation of a Multilevel Secure Relational Database System Using View Concepts

Wan-Soo Cho Hae-Young Bae
Institute for Defense Information Systems Inha University

요 약

다단계 보안을 지원하는 데이타베이스 시스템은 상이한 보안 등급을 갖는 데이타와 상이한 접근 권한을 갖는 사용자를 동시에 지원하는 데이타베이스 시스템이다. 본 논문에서는 다단계 보안 관계 데이타베이스 관리 시스템을 구현하기 위하여 주요 시스템 설계 전략을 분석하고, 이를 기초로 정보보안베이스 분할 및 계층화, 균형 보증 방식에 의한 보안 커널 구성, 다단계 보안 기반으로써의 뷰의 사용 등을 설계 전략으로 채택한다. 계층 구조에 의한 다단계 보안 데이타베이스 시스템의 설계는 기존 보안 기술의 사용을 가능하게 하여 최소한의 개발 노력으로 시스템을 구현할 수 있도록 한다. 최상위 계층의 다단계 질의 처리기는 사용자에 의한 다단계 릴레이션의 정의 및 다단계 질의어의 처리를 위한 전처리기로써 표준 관계 데이타베이스 관리 시스템과의 인터페이스를 담당한다.

1. 서론

데이타베이스는 데이타의 단순한 수동적인 집합체라기보다는 다양한 요구를 지닌 많은 사용자에 의해 공유될 수 있는 실세계 시스템을 동적으로 표현하는 모델로 간주할 수 있다. 사용자는 데이타베이스가 지니고 있는 강력하고 편리한 인터페이스를 통하여 상호 연관되고 통합되어 있는 대량의 데이타를 손쉽게 검색할 수 있기 때문에 데이타베이스 환경에서의 데이타 보안에 대한 요구사항은 명확히 정의되어야 한다.

데이타베이스 보안은 데이타베이스에 저장되어 있는 데이타에 대한 인가되지 않은 접근, 의도적인 데이타의 변경이나 파괴 및 데이타의 일관성을 저해하는 우발적인 사고 등으로부터 데이타 혹은 데이타베이스를 보호하는 것이다. 대부분의 상용 데이타베이스 시스템은 보안 대책으로 데이타에 대한 사용자의 사용 권한을 제어하는 접근 제어(access control)를 채택하고 있으나 이들은 운영체제(operating system)를 위한 보안 요구사항을 반영한 것으로 데이타베이스에 대해서는 적용이 부적합하거나 많은 제약사항을 유발하는 문제점을 지니고 있다[1]. 접근 제어를 위한 보안 정책은 크게 임의적 접근 제어(discretionary access control)와 강제적 접근 제어(mandatory access control)로 구분된다[2]. 임의적 접근 제어는 주체나 주체가 속해 있는 그룹의 식별자를 근거로 엔티티에 대한 접근을 제한하는 방식이며, 강제적 접근 제어는 엔티티에 포함된 정보의 비밀 등급(sensitivity 혹은 security markings)과 주체에 부여된 비밀 취급 인가(authorizations 혹은 clearances)를 기반으로 엔티티에 대한 접근을 제어하는 방식이다. 강제적 접근 제어는 다단계(multilevel) 보안의 구현을 위한 방법론의 핵심이 된다.

본 논문에서는 다단계 보안을 지원하는 관계 데이타베이스 시스템을 설계하기 위하여 주요한 시스템 설계 전략을 분석하고, 계층 구조 방식으로 다단계 보안 관계 데이타베이스 시스템을 설계한다. 또한 다단계 보안 요구사항을 준수하기 위한 최상위 계층의 다단계 질의 처리기를 설계하고, 다단계 질의 처리기의 주요 구성요소인 다단계 메타 데이타베이스 및 다단계 데이타 정의어 처리기 등을 기술한다.

2. 다단계 보안 데이타베이스 시스템 구현 전략

다단계 데이타베이스를 설계하고 구현하는 주요 접근 방법은 접근 제어를 수행하는 시스템 구성 요소와 보안 커널(security kernel)의 위치에 따라 크게 신뢰 필터(trusted filter, TF) 방식, 균형 보증(balanced assurance, BA) 방식 및 획일 보증(uniform assurance, UA) 방식의 세 가지로 구분된다[3,4,5].

신뢰 필터 방식 혹은 무결성 로크(integrity lock) 방식은 신뢰할 수 없는 전위 사용자 인터페이스와 후위 데이타베이스 사이에 신뢰 필터를 사용하여 데이타에 대한 접근 제어 및 보안 서비스를 제공한다. 다단계 보안을 지원하기 위해 신뢰 필터는 저장 객체에 보안 분류를 부착하고, 무결성 로크를 사용하여 데이타에 대한 모든 접근을 조정한다. 이를 위하여 신뢰 필터는 하부의 보안 운영 체제(secure operating system)에 의해 제공되는 보안 서비스 및 메카니즘에 의존한다. 신뢰 필터 방식은 균형 보증 방식이나 획일 보증 방식에 비해 간단하며 충분히 작기 때문에 보안 기능의 검증 및 평가가 매우 용이하다. 그러나 데이타의 보안을 침해하는 일부의 위협에 대해서는 취약점을 지니고 있다.

균형 보증 방식(BA 방식)은 Hinke/Schaefer 방식 또는 커널(kernelized) 방식이라고도 하며, 정보보안 베이스(Trusted Computing Base, TCB) 분할(subsetting)이라는 개념에 의하여 구현된다[6,7,8,9]. TCB는 보안 정책의 시행을 책임지는 하드웨어, 펌웨어, 소프트웨어 및 이들의 조합을 포함하는 컴퓨터 시스템내의 모든 보호 메카니즘을 의미한다. TCB 분할은 TCB를 계층화 구조로 구현하는 것으로써, 각 계층은 고유의 보안 정책을 정의하기 때문에 기존의 TCB를 재 사용할 수 있고, TCB의 확장이 용이하게 된다. 이는 강제적 보안을 관리하기 위한 보안 커널(혹은 참조 모니터)은 검증 및 분석이 용이하도록 가급적이면 작아야 한다는 요구사항에 따른 개념이다. TCB 분할을 택하는 BA방식에 의하면 데이타베이스 시스템은 보안 커널 외부에 존재하면서 임의적 보안만을 관리한다. 즉, 데이타베이스 객체에 대한 임의적 접근 제어는 데이타베이스 관리 시스템에 의해 수행되지만, 데이타베이스 화일에 대한 임의적 접근 제어 및 모든 강제적 접근 제어는 하부의 보안 운영 체제에 의하여 제공된다. 따라서 BA 방식의 다단계 데이타베이스는 데이타베이스 객체를 위한 다단계 보안을 지원하기 위해 보안 운영 체제가 제공하는 보안 서비스에 의존한다.

획일 보증 방식(UA 방식)은 이중 커널(dual kernel) 방식이라고도 하며, 다단계 보안을 위한 모든 책임을 데이타베이스 시스템이 갖기 때문에 데이타베이스 자체에 보안 커널을 보유한다. 따라서 강제적 보안 기능을 갖는 데이타베이스 시스템을 구현하고, 이를 보안 운영 체제와 함께 시스템의 TCB로 간주하여 보안 시스템을 평가한다. 이는 보안 커널의 최소화를 요구하는 보안 요구사항에는 부적합한 방식으로 보안 기능의 검증 및 평가가 어려워진다.

다단계 데이타베이스를 구현하기 위한 세 가지 방식의 분석 결과를 [그림 1]에 제시하였다.

[그림 1] 다단계 보안을 위한 데이타베이스 구현 전략의 분석

	TF 방식	BA 방식	UA 방식
보안 운영 체체 의존도	상	상	중
보안 구조의 견고성	하	중	상
데이타 보호	하	상	상
보안 검증의 용이성	상	중	하
구현의 용이성	상	중	하
데이타베이스 상호 운영성	하	상	상

다단계 보안을 위한 관계 데이타베이스 시스템을 구현하기 위하여 표준 관계 데이타 모델을 확장하는 여러 가지 접근 방법이 제안되었다[10,11]. 이들 방법의 주요 관점은 릴레이션에 저장된 데이타에 접근 등급을 어떻게 할당하느냐 하는 것이다. I. P. Sharp 모델은 릴레이션 수준에 접근 등급을 할당하고, TRW 모델은 투플 수준에 접근 등급을 할당하며, Hinke/Schaefer 모델은 릴레이션의 개별적인 애트리뷰

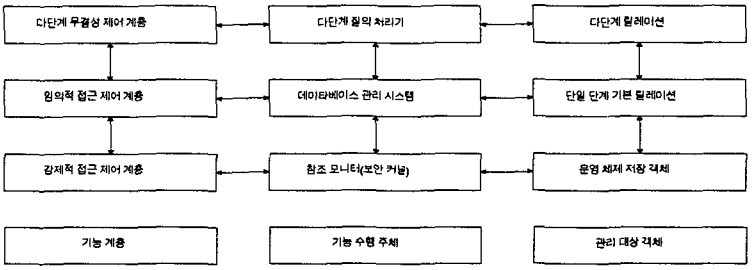
트에 접근 등급을 할당한다. 반면 SRI의 SeaView(Secure Data View)와 SCTC의 LDV(LOCK Data Views) 모델은 투플 내의 개별적인 데이타 요소에 접근 등급을 할당한다.

관계 데이타베이스 시스템의 뷰는 저장 데이타(stored data)와 유도 데이타(derived data)를 수학적으로 정의하기 때문에 다단계 관계 데이타베이스 시스템의 문맥 종속(context-dependent) 및 내용 종속(content-dependent) 보안 분류(classification), 동적 분류(dynamic classification), 추론(inference) 및 집단화(aggregation) 등을 처리하는 수단으로 제안되고 있다[12]. 보안의 목적으로 뷰를 사용하는 개념은 CODASYL과 IBM의 System R로부터 유래된다. System R에서, 뷰는 구조적 질의어 SQL로 표현되는 유도 릴레이션이다. System R의 접근 제어 메카니즘은 뷰를 권한 부여의 대상으로 간주한다. 즉, 뷰는 물리적 데이타보다 높은 수준의 추상화를 제공하며, 문맥 종속 및 내용 종속 제약조건의 명세를 허용한다. IBM의 연구와는 별도로 SRI의 Neumann은 뷰를 이용하여 SRI의 보안 운영체계인 PSOS(Provably Secure Operating System) 상에 보안 관계 데이타 관리 시스템을 구현할 수 있음을 밝혔다. PSOS 방식에서, 뷰는 단일 릴레이션의 부분 집합으로 제한되고 릴레이션의 선택적인 접근을 위한 자격(capability)의 역할을 수행한다. 그러나 IBM과 SRI의 PSOS 프로젝트는 강제적 보안과 데이타 분류를 위한 뷰의 사용 개념은 제안하지 않았다.

다단계 보안 데이타베이스 시스템의 기반으로써의 뷰의 사용은 Claybrook과 Denning에 의해서 처음 제안되었다[13]. SRI의 SeaView는 다단계 보안을 위한 뷰의 사용 개념을 지원하는 최초의 관계 데이타베이스 관리 시스템을 구현한 프로젝트로 다단계 보안을 지원하는 관계 데이타베이스 분야의 연구를 상당히 진전시켰다. SeaView에서 제안된 TCB 분할 개념에 의하면 다단계 릴레이션은 단일 단계 기본 릴레이션(single level base relation) 상의 뷰로써 구현된다. 각각의 단일 단계 릴레이션은 다시 참조 모니터(reference monitor)에 의해서 보호되는 하나 이상의 단일 단계 세그먼트(segment) 혹은 화일에 사상된다[14]. 따라서 각 주체는 자신의 접근 등급이 데이타가 저장되는 객체의 등급을 지배하지 않는 한 다단계 릴레이션을 유도하기 위해 기본 릴레이션 내의 어떠한 데이타에도 접근할 수 없기 때문에 데이타베이스 보안을 위한 강제적 접근 제어의 요구사항을 충족시킬 수 있게 된다. 다단계 릴레이션을 뷰로써 구현하는 것은 다단계 릴레이션에 대한 삽입, 삭제 및 갱신 연산이 단일 단계의 저장 릴레이션(stored relations)에 대한 대응되는 연산으로 전환될 수 있도록 한다.

3. 다단계 보안 관계 데이타베이스 시스템의 계층 구조

뷰의 개념을 이용한 다단계 릴레이션의 구현은 계층 구조에 의한 다단계 보안 관계 데이타베이스 관리 시스템의 설계를 가능하게 한다. 다단계 데이타베이스를 구현하기 위한 기능 계층 및 각 계층에서의 기능 수행 주체와 관리 대상 객체에 대한 계층 구조는 [그림 2]와 같이 표현할 수 있다. 본 장에서는 다단계 데이타베이스 관리 시스템의 구현을 위하여 사용자에 의한 다단계 릴레이션의 정의와 다단계 질의어의 처리를 위한 전처리기 개념의 다단계 질의 처리기의 구조 및 주요 구성요소에 관하여 논한다.

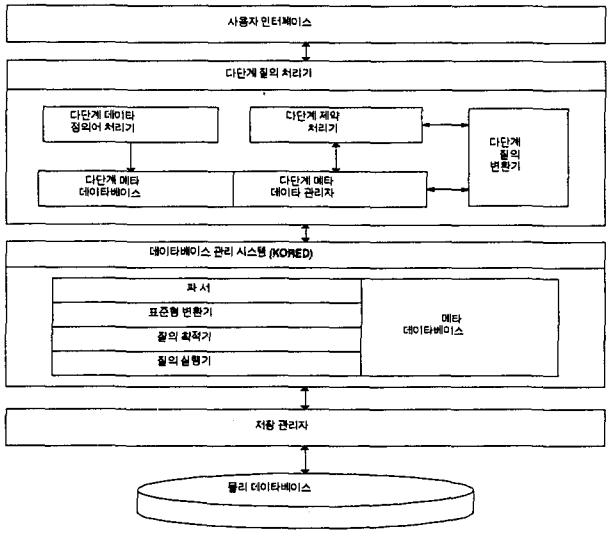


[그림 2] 다단계 데이타베이스 구현을 위한 계층 구조

3.1 다단계 질의 처리기 구조

다단계 릴레이션을 단일 단계 기본 릴레이션 상의 뷰로 구현하게 되면 다단계 릴레이션에 대한 연산을 단일 단계 기본 릴레이션에 대한 대응되는 연산으로 전환할 수 있게 된다. 따라서 단일 단계 기본 릴레이션의 관리를 위해 표준적인 관계 데이타베이스 관리 시스템을 사용하고, 뷰의 형태로 구현되는 다단계 릴레이션을 관리하기 위한 별도의 소프트웨어 계층을 그 위에 추가함으로써 계층 구조 방식으로 다단계 보안 관계 데이타베이스 관리 시스템을 설계할 수 있다. 이러한 계층 구조 방식의 설계는 이미 충분히 테스트되고 그 기능이 증명된 기존 보안 기술의 사용을 가능하게 하여 최소한의 개발 노력으로 다단계 시스템을 구현할 수 있도록 한다[15,16,17,18].

다단계 보안 관계 데이타베이스 시스템을 구현하는 시스템 구조는 [그림 3]과 같다. 다단계 질의 처리기는 하부 데이타베이스 관리 시스템에 대하여 사용자에게 제공되는 유일한 인터페이스로써 다단계 릴레이션을 정의 및 관리하며, 다단계 관계 무결성 제약조건을 유지한다. 또한 다단계 릴레이션에 대한 사용자의 연산을 대응되는 기본 릴레이션의 연산으로 변환하며, 하부 데이타베이스 관리 시스템에 의해 제공되는 모든 데이타베이스 관리 기능을 지원한다. 다단계 질의 처리기의 구성요소는 다단계 데이타 정의어 처리기, 다단계 메타 데이타 관리자, 다단계 질의 변환기 및 다단계 제약 처리기 등으로 이루어진다.



[그림 3] 다단계 보안 관계 데이타베이스 시스템 구조

다단계 데이타 정의어 처리기는 사용자가 정의하는 다단계 릴레이션 스키마를 다단계 메타 데이타베이스 내의 여러 관리 테이블이 필요로 하는 형태의 정보로 변환하여 저장한다. 다단계 메타 데이타 관리자는 다단계 릴레이션 및 질의 변환에 필요한 정보를 저장하는 메타 데이타베이스를 관리하며, 필요한 정보를 다단계 질의 변환기에게 제공한다. 다단계 제약 처리기는 응용 종속적인 속성을 지니는 다양한 보안 규칙을 저장하며, 사용자의 다단계 질의 처리시 고려하여야 하는 보안 제약조건의 관리를 담당한다. 다단계 질의 변환기는 사용자 인터페이스를 통하여 입력된 질의어를 다단계 메타 데이타와 다단계 보안 제약조건을 이용하여 구문적인 관점 및 의미적인 관점에서 검사하고, 이들 질의어를 대응되는 단일 단계 기본 릴레이션 상의 질의어로 변환한다. 또한 기본 릴레이션에 대한 표준적인 질의의 결과로 제시된 질의 응답을 원래의 다단계 질의에 대한 적절한 형태의 다단계 응답으로 재구성한다.

하부의 데이타베이스 관리 시스템은 다단계 릴레이션의 분해 결과로 생성되는 단일 단계 기본 릴레이션을 관리하며, 다단계 질의 처리기에 의해 변환된 사용자 질의를 처리하여 그 결과를 다단계 질의 처리기에게 제공한다. 다단계 질의 처리기로부터 전달되는 변형된 표준 형태의 질의어는 KORED[22]와 같은 데이타베이스 관리 시스템이 제공하는 질의 처리기, 질의 최적기, 질의 실행기 등에 의해 처리된다. 즉, 변형된 질의가 파서(parser)에 의해 표준형으로 변환되는 이해 단계와 이를 최적의 실행 비용을 갖는 최적화된 정규형으로 변환하는 개념 단계, 그리고 위의 두 단계를 거쳐 생성된 표준형(canonical logic form)을 사용하여 직접 데이타베이스를 접근하는 내부 단계를 거치며 처리된다.

3.2 다단계 메타 데이타베이스

메타 데이타베이스는 데이타베이스 관리 시스템의 기능 수행을 위하여 필요한 정보를 수록한 데이타 사전으로 질의 처리를 위한 메타 데이타를 정의한다. 데이타 사전은 테이블, 애트리뷰트, 뷰, 인덱스 등의 데이타베이스 구성요소에 관한 정보, 테이블간의 관계에 관한 정보, 사용자의 접근 권한에 관한 정보 등을 포함하며, 이들은 자동적으로 시스템에 의해 최신의 상태로 유지 관리된다. 데이타 사전에 저장된 정보는 테이블 형태로써 데이타베이스 관리 시스템의 기능 수행에 도움을 줄 뿐만 아니라 사용자가 정의한 데이타베이스 구조를 이해하는데 도움을 주어야 한다[19,20,21]. 데이타 사전은 권한을 가진 사용자에 한해서 선택문의 일반적인 질의 형태로만 검색이 허용된다. 따라서 데이타 조작어(Data Manipulation Language)를 사용한 데이타 사전 내용의 직접적인 삽입, 삭제 등은 허용되지 않으며 오로지 데이타 정의어(Data Definition Language)에 의한 스키마의 정의, 변경, 삭제 연산에 의해서만 영향을 받는다.

다단계 메타 데이타베이스는 다단계 질의 처리를 위한 메타 데이타를 제공하므로 데이타 사전의 내용에 따라 사용자의 데이타베이스 사용이나 질의 처리가 영향을 받게 된다. 따라서 사용자 정의 릴레이션의 접근을 제어하는 것과 동일한 접근 제어 규칙을 데이타 사전에도 적용하여야 하며, 데이타 사전은 다단계 보안을 지원할 수 있도록 확장되어야 한다. 본 논문에서 제안하는 다단계 메타 데이타베이스는 다단계 릴레이션의 형태로 구성되고 일반적인 사용자 정의 릴레이션과 동일한 접근 방식을 사용한다. 즉, 다단계 릴레이션의 질의에 사용하는 동일한 질의어를 동일한 소프트웨어 모듈을 사용하여 데이타 사전에 대해서도 사용할 수 있다. 본 절에서는 다단계 릴레이션의 관리 및 다단계 릴레이션과 단일 단계 기본 릴레이션간의 매핑을 관리하는데 필요한 데이타 사전의 자료 구조를 기술한다.

다단계 메타 데이타베이스는 사용자에 대한 접근 허용의 관점에서 정의 부분과 매핑 부분의 두 부분으로 구성된다. 정의 부분은 사용자가 정의하는 다단계 릴레이션에 대한 정보를 포함하는 부분으로 사용자에 의한 접근이 가능하다. 매핑 부분은 다단계 릴레이션과 이에 대응되는 단일 단계 기본 릴레이션간의 매핑에 대한 정보를 포함하는 부분으로 이 부분은 사용자에 의한 접근이 허용되지 않는다. 다단계 메타 데이타베이스에 대한 엔트리는 다단계 데이타 정의어 CREATE TABLE 문이 실행될 때 자동적으로 이루어진다.

다단계 메타 데이타베이스의 엔트리는 사용자 정의 릴레이션과 마찬가지로 다중 사례 요소 및 다중 사례 엔티티의 형태로 다중 사례가 발생할 수 있다. 데이타 사전에서의 다중 사례 요소는 데이타 사전 내의 테이블에 동적으로 변하는 값을 갖는 애트리뷰트가 존재할 때 발생한다. 예로서, 다단계 릴레이션의 이름이나 애트리뷰트의 이름 등은 값이 변하지 않는 정적인 애트리뷰트이지만, 투플의 개수 등은 계속적으로 값이 변화할 수 있는 동적인 애트리뷰트이다. 다중 사례 요소 관계에 있는 모든 투플은 동일한 릴레이션의 정의를 표현한다. 반면에 다중 사례 엔티티 관계에 있는 모든 투플은 상이한 릴레이션의 정의를 의미한다. 엔트리가 생성되어 다단계 메타 데이타베이스 내의 테이블에 저장될 때 삽입되는 각 투플의 요소 분류 값은 모두 릴레이션을 생성한 사용자의 접근 등급으로 설정되지만 동적 애트리뷰트는 생성 이후에 해당 애트리뷰트 값의 변경을 초래하는 다단계 질의어를 발행한 사용자의 접근 등급으로

설정된다. 또한 각 투플의 투플 등급은 사용자 정의 다단계 릴레이션의 투플 등급과 동일한 방식으로 계산된다.

사용자 질의를 변환 및 처리할 때, 다단계 질의 처리기는 다단계 릴레이션 및 릴레이션의 애트리뷰트 등에 관한 정보를 얻기 위하여 다단계 메타 데이타베이스에 적용되는 일련의 질의를 자체적으로 생성하며, 이 질의의 결과를 이용하여 사용자 질의를 처리한다. 따라서 다단계 질의 처리기는 반드시 질의를 발행한 사용자와 동일한 접근 등급에서 수행되어야 한다. 또한 다단계 질의 처리기에 의해 생성되는 모든 내부적 질의문은 다단계 릴레이션의 연산 의미를 준수하여야 한다.

다단계 릴레이션의 생성에 따른 메타 데이타의 관리와 다단계 릴레이션의 분해에 따른 매핑 정보의 관리에는 다음과 같은 메타 데이타베이스 관리 테이블이 요구된다.

(1) 데이타베이스 관리 테이블

데이타베이스에 관한 정보를 관리하며 데이타베이스 이름, 테이블 수, 생성자, 생성 일자, 접근 경로 등의 애트리뷰트로 구성되어 데이타베이스 전반에 걸친 정보를 포함한다.

(2) 다단계 릴레이션 관리 테이블

사용자가 정의한 다단계 릴레이션, 뷰, 그리고 질의어 실행으로 생성되는 임시 결과 테이블 등에 관한 정보가 저장된다. 따라서 다단계 릴레이션 이름, 투플 길이, 투플 개수, 다단계 릴레이션에 속한 인덱스 개수, 다단계 릴레이션의 애트리뷰트 개수 등을 표현하는 애트리뷰트로 구성된다.

(3) 애트리뷰트 관리 테이블

다단계 릴레이션을 구성하는 애트리뷰트에 대한 정보를 관리하며 애트리뷰트가 속한 다단계 릴레이션 이름, 애트리뷰트 이름, 도메인, 분류 등급의 범위, 키에 관한 정보, 널 값의 허용에 관한 정보 등이 저장된다. 분류 등급의 범위는 하한 값과 상한 값으로 분리되어 각각 별도의 애트리뷰트로 저장된다. 분류 범위의 하한 값과 상한 값은 정적 애트리뷰트로 그 값이 변하지 않는다. 그러나 상한 값은 상한 값에 의해 지배되는 접근 등급의 주체가 다단계 메타 데이타베이스를 접근할 때에는 항상 주체의 접근 등급으로 대체되어 표시된다.

(4) 매핑 관리 테이블

다단계 릴레이션의 분해와 이에 대응되는 접근 등급별 단일 단계 기본 릴레이션간의 매핑에 대한 정보를 관리하며, 사용자에 의한 접근은 허용되지 않고 다단계 질의어 처리기에 의해서만 접근이 허용되는 메타 데이타베이스 부분이다. 따라서 다단계 릴레이션 이름, 릴레이션의 존재 가능한 접근 등급 내의 모든 범위 값, 접근 등급별 기본 릴레이션의 존재 여부, 기본 릴레이션의 이름 등을 표현하는 애트리뷰트로 구성된다. 기본 릴레이션의 이름은 매핑 관리 테이블의 애트리뷰트로 저장할 수도 있으나 기본 릴레이션에 이름을 부여하는 명칭 부여 규칙을 정의하여 활용할 수도 있다. 즉, 사용자가 정의하는 다단계 릴레이션의 이름에 기본(base)을 의미하는 접두어 "B"와 기본 릴레이션이 존재하는 접근 등급을 의미하는 접근 등급의 값 "C"를 접미어로 부착하는 등의 방식으로 기본 릴레이션의 이름을 정의할 수 있다. 사용자가 다단계 릴레이션을 정의할 때, 다단계 릴레이션의 분해에 대응되는 기본 릴레이션은 기본 릴레이션이 존재할 수 있는 모든 가능한 접근 등급별로 사전에 생성되지는 않는다. 그대신 사용자가 정의한 다단계 릴레이션에 사용자의 투플이 삽입될 때 데이타의 저장 필요에 따라 단일 단계 기본 릴레이션

이 생성된다. 즉, 다단계 릴레이션이 생성되는 시점에서 정적으로 모든 가능한 접근 등급에 대해 기본 릴레이션을 생성하는 것이 아니고 데이타의 저장을 위해 필요한 경우에만 동적으로 기본 릴레이션을 생성한다.

3.3 다단계 데이타 정의어 처리기

다단계 데이타 정의어는 데이타베이스 자체의 스키마를 정의할 뿐만 아니라 다단계 보안을 지원하기 위하여 확장된 관계 모델의 모든 자료 구조를 정의할 수 있는 수단을 제공한다. 본 절에서는 다단계 릴레이션을 생성하고 삭제하기 위한 다단계 데이타 정의어와 이의 처리를 위한 다단계 데이타 정의어 처리기를 기술한다.

사용자가 정의하는 다단계 테이블 즉, 다단계 릴레이션은 단일 단계 기본 릴레이션 상의 뷰로 정의된다. 따라서 다단계 정의어 CREATE TABLE 문의 실행으로 정의되는 다단계 릴레이션은 그 어커런스가 실제의 저장 매체에 저장되는 것이 아니라 매번 다단계 릴레이션을 사용하는 다단계 질의문에 대해 해당 질의문을 다단계 질의 변환기로 수정 및 변환해서 실제 저장되어 있는 기본 릴레이션으로부터 해당 투플들을 가져 온다. 이와 같은 질의 변형 기법을 통한 다단계 릴레이션의 스키마 정의 방식은 저장 공간을 절약하고 기본 릴레이션과 그로부터 유도되는 다단계 릴레이션 상태 사이에 일치된 관점을 제공한다[22].

다단계 릴레이션을 생성하고 삭제하기 위한 확장된 CREATE TABLE 문과 DROP TABLE 문의 구문 구조는 [그림 4]와 같다.

```
CREATE TABLE 릴레이션_이름
        (애트리뷰트_정의 [, 애트리뷰트_정의]…
         [, 기본_키_정의]
         [, 외래_키_정의 [, 외래_키_정의]…])
         [IN 데이타베이스_공간]
         애트리뷰트_정의 = [GROUP() 애트리뷰트_이름 데이타_유형
                              [, 애트리뷰트_이름 데이타_유형…)] [분류_범위] [NOT NULL]
         기본_키_정의 = PRIMARY-KEY (애트리뷰트 [, 애트리뷰트]…)
         외래_키_정의 = FOREIGN-KEY 릴레이션_이름
                        (애트리뷰트 [, 애트리뷰트]…)
                        [ON DELETE [RESTRICT|CASCADE|SET NULL]]

DROP TABLE 릴레이션_이름
```

[그림 4] CREATE TABLE 및 DROP TABLE의 구문 구조

애트리뷰트_정의에서 둘 이상의 애트리뷰트가 GROUP으로 지정되면 이들은 동일한 접근 등급을 갖게 되는 애트리뷰트 그룹을 형성한다. 이러한 애트리뷰트 그룹의 사용은 기본 키의 지정시 다단계 엔티티 무결성의 검사를 쉽게 한다. 분류_범위는 애트리뷰트_이름으로 지정되는 데이타 애트리뷰트에 대한 대응되는 분류 애트리뷰트를 지정하는 것으로 접근 등급 속의 부속을 표현하는 접근 등급의 쌍[L_i, H_i]으로 지정된다. 따라서 분류 애트리뷰트의 이름은 사용자에 의해 명시적으로 부여되지 않고, 대응되는 데이타 애트리뷰트의 이름을 이용하여 별도로 정의되는 분류 애트리뷰트 명칭 부여 규칙에 따라 다단계 데이타 정의어 처리기가 내부적으로 부여한다. 분류_범위가 생략되는 경우에도 분류 애트리뷰트의 이름

은 항상 부여되며, 이때의 분류 애트리뷰트 값은 CREATE TABLE 문을 수행하여 다단계 릴레이션 스키마를 생성하는 주체의 접근 등급 즉, 다단계 릴레이션 이름의 접근 등급으로 결정된다. CREATE TABLE 문을 수행하는 c-주체에 의해 접근 등급의 쌍[L_i, H_i]으로 분류_범위가 지정되는 경우에는 모든 쌍[L_i, H_i]에서 L_i의 값은 반드시 다단계 릴레이션을 생성하는 주체의 접근 등급을 지배하여야 한다. 즉, 모든 L_i에 대하여 $L_i \geq c$가 항상 성립하여야 한다. 또한 모든 L_i는 모두 동일한 값으로 지정되어야 한다.

다단계 데이타 정의어 처리기는 다단계 데이타 정의어로 기술된 스키마를 다단계 메타 데이타베이스 내의 여러 관리 테이블이 필요로 하는 정보로 변환하여 저장한다. 다단계 릴레이션을 생성하기 위한 테이블 정의어는 다단계 데이타 정의어 처리기에 의해 파서를 통하여 다단계 릴레이션 이름, 애트리뷰트 이름, 데이타 유형, 분류 등급 범위 등에 관한 정보로 분석되며, 분석된 정보는 다음과 같은 단계를 거쳐 다단계 메타 데이타베이스에 저장된다. 우선 다단계 데이타 정의어 처리기는 데이타 정의어로 기술된 다단계 릴레이션에 관한 정보를 다단계 메타 데이타베이스에 생성시키기 위하여 CREATE TABLE 문을 작성한 사용자가 릴레이션을 생성할 수 있는 권한이 있는지를 검사한다. 이때 처리기는 다단계 릴레이션의 생성, 변경, 삭제 등의 권한이 기록되어 있는 메타 데이타베이스를 참조하여 검사를 수행한다. 만약 사용자가 권한을 갖고 있지 못한 경우라면 수행을 거부한다. 사용자의 권한 검사를 수행한 후에는 다단계 릴레이션의 이름을 파서로부터 받아 다단계 릴레이션 관리 테이블을 참조하여 다단계 릴레이션의 중복 여부를 검사한다. 같은 이름의 다단계 릴레이션이 동일한 접근 등급에서 중복되어 존재한다면 다단계 릴레이션을 생성할 수 없다. 다단계 릴레이션의 이름과 접근 등급이 중복되지 않는다면 애트리뷰트의 이름과 데이타 유형 등에 관한 정보를 파서로부터 받아 다단계 메타 데이타베이스 내의 다단계 릴레이션 관리 테이블, 애트리뷰트 관리 테이블 등에 기록한다. 또한 생성되는 다단계 릴레이션의 분해 및 기본 릴레이션과의 매핑에 관한 정보를 생성하여 다단계 메타 데이타베이스 내의 매핑 관리 테이블에 기록한다.

다단계 릴레이션을 삭제하기 위한 테이블 삭제문은 다단계 데이타 정의어 처리기에 의해 파서를 통하여 삭제할 다단계 릴레이션 이름이 분석되고, 다음의 단계를 거쳐 다단계 메타 데이타베이스에서 다단계 릴레이션에 관한 정보가 삭제된다. DROP TABLE 문을 사용한 사용자가 해당 릴레이션을 삭제할 수 있는 권한이 있는지를 메타 데이타베이스를 참조하여 검사한다. 권한 검사가 통과되면 삭제하려는 다단계 릴레이션의 이름이 존재하는지 다단계 릴레이션 관리 테이블을 참조하여 검사한다. 다단계 릴레이션이 존재하는 경우, 삭제 대상 릴레이션으로부터 유도된 뷰 또는 릴레이션에 연관된 모든 인덱스 화일 등을 제거하고 다단계 릴레이션이 생성될 때 다단계 메타 데이타베이스에 저장된 모든 정보와 응용 프로그램을 통해 생성된 메타 데이타베이스 내의 모든 관련 정보를 삭제한다. 이때 삭제되는 다단계 릴레이션의 정의와 다중 사례 요소 관계에 있는 모든 투플은 다단계 메타 데이타베이스로부터 삭제되어야 한다.

3.4 다단계 질의 변환기

다단계 질의 변환기는 사용자에 의한 다단계 질의어를 입력으로 받아 들여 이를 대응되는 기본 릴레이션에 대한 일련의 표준적인 질의어로 변환한다. 변환된 사용자 질의어는 하부의 데이타베이스 관리 시스템에게 전달되고, 하부 데이타베이스 관리 시스템은 이들 일련의 질의어를 수행하여 그 결과를 다단계 질의 변환기에게 전달한다. 다단계 질의 변환기는 이 질의 결과를 원래의 다단계 질의어에 적합한 형태의 질의 결과로 재구성하여 사용자에게 제공한다. 따라서 모든 데이타베이스 사용자는 다단계 질의 처리기를 통해서만 데이타베이스에 접근을 할 수 있다.

다단계 질의 변환기는 다단계 질의어를 어떻게 변경할 것인가 혹은 어떠한 질의를 거절할 것인가 등의 결정을 내릴 때 개별적인 질의어의 구문뿐만 아니라 다단계 질의를 발행한 사용자의 접근 등급도

고려하여야 한다. 다단계 질의 변환기가 다단계 릴레이션에 대한 다단계 질의어를 단일 단계 기본 릴레이션에 대한 대응되는 일련의 표준적인 질의어로 변환하는 과정은 크게 구문 분석 단계, 보안 분석 단계, 질의 보완 단계 및 질의 변환 단계로 구분된다.

다단계 질의 변환기에 의해 변환 처리되어 하부의 데이타베이스 관리 시스템으로 전달되는 일련의 표준 질의어는 데이타의 일관성 유지 및 병행 처리되는 다른 다단계 트랜잭션을 방해하지 않도록 하나의 원자(atomic) 트랜잭션으로 수행되어야 한다. 하부의 데이타베이스 관리 시스템에 의해 제공되는 질의 처리 결과는 하나 이상의 표준적인 투플로 구성되며, 이러한 투플 집합은 원래의 다단계 질의에 대한 응답이 될 수 있도록 투플 등급이 추가되는 등 다단계 투플 형태로 확장 변환되어야 한다. 이와 같은 질의 처리 결과의 변환 과정은 다단계 릴레이션의 복구 알고리즘으로부터 유도된다.

4. 결론

본 연구는 다단계 보안을 지원하는 관계 데이타베이스 관리 시스템을 설계하기 위한 연구로써 설계 방법으로는 정보보안베이스 분할 및 계층화, 균형 보증 방식에 의한 보안 커널 구성, 다단계 데이타베이스 보안 기반으로써의 뷰의 사용 등을 채택하였다. 다단계 질의 처리기는 다단계 무결성 제약조건 및 다단계 데이타베이스 운영 의미 등 다단계 보안 요구사항을 보장하는 전처리기로써 다단계 데이타 정의어 처리기, 다단계 메타 데이타 관리자, 다단계 질의 변환기 등이 포함된다. 또한 다단계 릴레이션에 대응되는 표준적인 기본 릴레이션을 관리하는 하부 데이타베이스 관리 시스템과의 인터페이스를 제공한다.

향후의 연구는 본 논문에서 제안된 시스템이 보다 완전한 시스템으로 발전할 수 있도록 다음과 같은 분야에서 계속적으로 진행되어야 한다. 다단계 추론(inference)은 임의의 데이타가 자신보다 높은 접근 등급을 갖는 데이타에 대한 정보를 유도하기 위해 사용되어질 때 발생한다. 따라서 다단계 추론을 방지하기 위한 데이타의 설계와 통계적 추론(statistical inference)을 방지하기 위한 여러 연구 결과를 응용하여 이를 제어할 수 있는 기법 및 도구를 설계하여야 한다. 또한 다단계 보안 환경에서의 트랜잭션을 처리하기 위한 트랜잭션 관리 기법의 확장에 대한 연구가 수행되어야 한다. 특히 트랜잭션의 병행 수행 제어(concurrency control)와 복구(recovery), 다단계 스케줄러 설계와 직렬성(serializability) 조건의 확장과 응용 종속적인 다양한 보안 규칙의 저장 및 관리를 위한 보안 제약조건 관리 등을 고려하여 다단계 트랜잭션 관리자를 설계 및 구현하여야 한다.

참 고 문 헌

[1] R. Graubart, "Comparing DBMS and Operating System Security Requirements: The Need for a Separate DBMS Security Criteria," *Proc. IFIP WG 11.3 Workshop on DB Sec.*, pp. 109-114, Sep. 1989.

[2] R. Sandhu, "Mandatory Controls for Database Integrity," *Proc. IFIP WG 11.3 Workshop on DB Sec.*, pp. 143-150, Sep. 1989.

[3] C. Laferriere, "A Discussion of Implementation Strategies for Secure Database Management Systems," *Computer & Security*, 9, pp. 235 - 244, 1990.

[4] R. R. Henning and S. A. Walker, "Computer Architectures and Database Security," *Advances in Computer System Security*, Vol. III, Artech House, Inc., pp. 249 - 263, 1988.

[5] B. Thuraisingham, "Current Status of R&D in Trusted Database Management Systems," *SIGMOD Record*, Vol. 21, No. 3, pp. 44 - 50, Sep. 1992.

[6] J. Wilson, "A Security Policy for an A1 DBMS (a Trusted Subject)," *Proc. 1989 IEEE Computer Society Symp. on Security and Privacy*, pp. 116 - 125, May 1989.

[7] T. H. Hinke, "DBMS Trusted Computing Base Taxonomy," *Proc. IFIP WG 11.3 Workshop on DB Sec.*,

pp. 97 - 108, Sep. 1989.

[8] T. F. Lunt, "Multilevel Database Systems: Meeting Class A1," *Proc. IFIP WG 11.3 Workshop on DB Sec.*, pp. 177 - 186, Oct. 1988.

[9] C. Garvey, N. R. Jensen, and J. Wilson, "The Advanced Secure DBMS: Making Secure DBMSs Usable," *Proc. IFIP WG 11.3 Workshop on DB Sec.*, pp. 187 - 195, Oct. 1988.

[10] D. E. Denning, T. F. Lunt, R. R. Schell, M. Heckman and W. Shockley, "A Multilevel Relational Data Model", *Advances in Computer System Security*, Vol. III, Artech House, Inc., pp. 234 - 248, 1988.

[11] S. Jajodia and R. Sandhu, "Polyinstantiation Integrity in Multilevel Relations", *Proc. 1990 IEEE Computer Society Symp. on Research in Security and Privacy*, pp. 104 - 115, May 1990.

[12] S. G. Akl and D. E. Denning, "Views for Multilevel Database Security", *Advances in Computer System Security*, Vol. III, Artech House, Inc., pp. 223 - 233, 1988.

[13] B. G. Claybrook, "Using Views in a Multilevel Secure Database Management System", *Proc. 1983 IEEE Computer Society Symp. on Security and Privacy*, pp. 4 - 17, April 1983.

[14] T. F. Lunt, D. E. Denning, R. R. Schell, M. Heckman and W. R. Shockley, "The SeaView Security Model", *IEEE Trans. on SE*, Vol. 16, No. 6, pp. 593 - 607, June 1990.

[15] B. M. Thuraisingham and H. H. Rubinovitz, " Multilevel Security Issues in Distributed Database Management Systems - III," *Computer & Security*, Vol. 11, No. 7, pp. 661-674, Nov. 1992.

[16] T. F. Lunt, R. R. Schell, W. Shockley, and D. Warren, " Toward a Multilevel Relational Data Language," *Proc. 4th Aerospace Comp. Sec. Appl. Conf.*, pp. 72-79, Dec. 1988.

[17] M. H. Kang, J. N. Froscher, J. McDermott, O. Costich, and R. Peyton, " Achieving Database Security Through Data Replication: The SINTRA Prototype," *Proc. 17th Comp. Sec. Conf.*, pp. 77-87, Oct. 1994.

[18] T. F. Lunt and P. K. Boucher, " The SeaView Prototype: Project Summary," *Proc. 17th Comp. Sec. Conf.*, pp. 88-102, Oct. 1994.

[19] G. Vossen and J. Yacabucci, "An Extension of the Database Language SQL to Capture More Relational Concepts," *SIGMOD Record*, Vol. 17, No. 4, pp. 70 - 78, Dec. 1988.

[20] N. R. Jensen, "Implications of Multilevel Security on the Data Dictionary of a Secure Relational DBMS," *Proc. Fourth Aerospace Comp. Sec. Appl. Conf.*, pp. 58 - 65, Dec. 1988.

[21] B. Thuraisingham, "Issues in Multilevel Secure Object-Oriented Database Management Systems - A Position Paper," *Proc. 13th Natl. Comp. Sec. Conf.*, Vol. II, pp. 609 - 612, Oct. 1990.

[22] 배해영, " 한글 질의어를 내장한 확장된 관계 데이타베이스 시스템," 박사학위 논문, 숭실대학교, 1991.