

수평분할과 뷰를 이용한
다단계 데이터베이스에서의 질의 처리

나 민영 최 병갑^o
육군사관학교 전산학과

Query Processing for Multi-level Databases
Using Horizontal Partitioning and Views

Minyoung Ra Byungkab Choi^o
Dept. of Computer Science, Korea Military Academy

ABSTRACT

Most works done so far have concentrated on developing data modeling techniques such as multi-level relation for data protection. These techniques, however, cannot be applied to practical area. This is because they require new queries or new architectures. In this paper, we propose a query processing technique for multi-level databases using horizontal partitioning and views, which does not need any change in database architecture and query language.

1. 서론

최근 정보의 중요성이 증가됨에 따라 컴퓨터 보안이 중요시 되고있다. 데이터베이스 분야에 있어서도 안전한 데이터베이스 관리 시스템에 관한 연구가 수행되고 있다. 안전한 DBMS란 데이터가 저장됨에 있어 안전한 방법으로 저장 보호되며, 안전하게 저장된 데이터를 사용자가 원할때 적절한 방법으로 평범한 데이터처럼 액세스할 수 있도록 지원하는 DBMS를 말한다.

데이터를 안전하게 보호하고 관리하는 것은 매우 중요한 과제로 부각되어 여러 관련 연구가 이루어지고 있다. 지금까지의 연구의 대부분은 주로 데이터 모델링에 있어서 데이터의 보호에 관점을 맞춘 것으로 다단계 릴레이션 기법을 이용한 연구[Jajo 90a, 90b, Lunt 92], 시맨틱 데이터 모델을 이용한 연구 [Smit 90], 또는 지문을 이용한 연구[Wagn 90] 등이 수행되었다.

그러나 이러한 기법은 이미 데이터베이스가 구축되어 운영 중인 곳에서는 적용하기 곤란하다. 왜냐하면 이러한 기법들은 기존 데이터베이스의 구조변경이나 새로운 구조를 요구하기 때문이다. 우리는 이미 수평분할과 뷰를 이용하여 기 구축된 테이블 구조에 영향을 미치지 않고 다단계 보안을 이룰 수 있도록 해주는 다단계 기법을 제시하였다[나 민영 94]. 본 논문에서는 수평분할과 뷰를 이용한 다단계 데이터베이스에서의 질의처리 기법을 제시하고자 한다.

2. 수평분할과 뷰를 이용한 다단계 데이터베이스 보안기법

보안의 핵심은 사용자로 하여금 오직 권한이 있는 시스템 자원에 대해서만 액세스하도록 보장해주는 것이다. 이를 구현하기 위해 각 시스템 객체는 그 수준을 나타내는 등급이 있어야 하고, 각 사용자는 오직 해당하는 등급의 자료만 액세스할 수 있도록 해 주어야 한다. 이를 다단계 보안이라 한다. 이 장에서는 기 구축된 데이터베이스의 구조에 영향을 주지 않고 다단계 보안을 이루는 기법인 수평분할과 뷰를 이용한 기법을 설명한다.

2.1 기본개념

테이블에 있는 튜플들이 다음 그림 1과 같이 등급별로 구분될 수 있다고 생각해보자.

Top Secret
Secret
Confidential
Unclassified

< 그림 1 > 튜플의 그루핑

이것은 즉 적절한 프레디키트에 의한 튜플의 클러스터링과 같은 개념이고 이는 테이블의 수평 분할이 가능함을 의미한다. 그러나 실제로 partitioning을 실시하여 별도의 테이블로 데이터를 저장하면 데이터베이스의 원래 구조가 깨지므로 수평분할된 결과를 뷰로 만들어 일반 사용자는 오직 기저 테이블만 액세스하도록 하고 그 효과를 각 등급별 해당 뷰에 전파하여 데이터를 보안등급에 맞게 자동 관리한다. 여기서 중요한 점은, 모든 등급의 사용자는 항상 기저 테이블을 자기 보안등급에 맞는 해당 데이터들로만 구성된 테이블로 인식하여 자료처리를 실시하나 실제 자료는 뷰 테이블에 의하여 등급별로 자동 분류 관리된다는 것이다.

이 기법의 장점은 기 구축된 데이터베이스를 활용 가능하므로 물리적 데이터베이스의 아무런 변경없이 다단계 데이터베이스 보안이 가능해진다는 점이다.

2.2 수평분할

릴레이션에 있는 레코드들이 특정 애트리뷰트(들)의 값(들)에 따라 비밀등급이 분류 가능하다고 하면 이는 곧 수평분할이 적용됨을 의미한다.

수평분할이란 전역 릴레이션을 수평파편이라 불리는 튜플들의 셀으로 조개는 작업으로써 수평 분할을 위한 요구사항으로써는 데이터베이스 스키마 정보, 응용 프로그램의 액세스 빈도, 그리고 응용 프로그램으로부터 결정된 단순 프레디키트 등을 들수 있다. 여기서 단순 프레디키트 (simple predicates)라 함은

'애트리뷰트 θ 값' 단, $\theta \in \{=, \neq, <, \leq, \geq, >\}$

의 형태를 가진 프레디키트으로써 이때 단순 프레디키트의 셀으로부터 생성된 임의의 파편에 속한 임의의 두 튜플이 응용 프로그램에 의해 액세스되는 확률이 같다고 가정한다. 단순 프레디키트의 예를 들면 다음과 같은 형태의 것들이다.

p1: TITLE="Elect. Eng."
 p2: TITLE="Syst. Anal."
 p3: SAL<40000
 p4: SAL>30000

예를 들어 다음 그림 2(a)와 같은 릴레이션 J가 있을 때

J₁ = $\sigma_{LOC="Montreal"}(J)$
 J₂ = $\sigma_{LOC="New York"}(J)$

$$J_3 = \sigma_{LOC='Paris'}(J)$$

과 같은 selection 연산자에 의한 릴레이션 J의 수평분할은 그림 2(b)와 같게 된다. 이러한 수평분할을 위한 기법에는 민텀 프레디카트 기법, adaptive 클러스터링 기법, 그래프를 이용한 기법 등이 있다.

J

JNO	JNAME	BUDGET	LOC
J1	Instrumentation	150000	Montreal
J2	Database Develop.	135000	New York
J3	CAD/CAM	250000	New York
J4	Maintenance	310000	Paris

(a) 예제 릴레이션

<p style="text-align: center;">J1</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>JNO</th> <th>JNAME</th> <th>BUDGET</th> <th>LOC</th> </tr> </thead> <tbody> <tr> <td>J1</td> <td>Instrumentation</td> <td>150000</td> <td>Montreal</td> </tr> </tbody> </table>	JNO	JNAME	BUDGET	LOC	J1	Instrumentation	150000	Montreal	<p style="text-align: center;">J2</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>JNO</th> <th>JNAME</th> <th>BUDGET</th> <th>LOC</th> </tr> </thead> <tbody> <tr> <td>J2</td> <td>Database Develop.</td> <td>135000</td> <td>New York</td> </tr> <tr> <td>J3</td> <td>CAD/CAM</td> <td>250000</td> <td>New York</td> </tr> </tbody> </table>	JNO	JNAME	BUDGET	LOC	J2	Database Develop.	135000	New York	J3	CAD/CAM	250000	New York
JNO	JNAME	BUDGET	LOC																		
J1	Instrumentation	150000	Montreal																		
JNO	JNAME	BUDGET	LOC																		
J2	Database Develop.	135000	New York																		
J3	CAD/CAM	250000	New York																		
<p style="text-align: center;">J3</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>JNO</th> <th>JNAME</th> <th>BUDGET</th> <th>LOC</th> </tr> </thead> <tbody> <tr> <td>J4</td> <td>Maintenance</td> <td>310000</td> <td>Paris</td> </tr> </tbody> </table>		JNO	JNAME	BUDGET	LOC	J4	Maintenance	310000	Paris												
JNO	JNAME	BUDGET	LOC																		
J4	Maintenance	310000	Paris																		

(b) 수평분할

< 그림 2 > 수평분할

2.3 뷰와 접근권한

뷰란 중요하거나 민감한 데이터를 그 데이터를 액세스할 권한이 없는 사용자로부터 감추기 위한 기법이다. 뷰에 대한 특성을 살펴보면 다음과 같다.

<뷰의 작성과 삭제>

뷰를 작성하려면 CREATE VIEW문을 사용한다. 뷰는 뷰 정의 SELECT문에 의해 결정되며 뷰를 정의하는 SELECT문은 시스템 카탈로그 sysviews에 수록된다. 그 후 다른 문에서 뷰가 지정되면 뷰 정의 SELECT문이 실행된다.

뷰의 컬럼에는 기저 테이블과 동일한 컬럼명을 사용할 수도 있고 새로운 컬럼명을 부여할 수도 있다. 단 뷰의 컬럼이 식의 값이 되는 경우 혹은 (예를 들면 조인을 정의하고 있는 컬럼을 모두 포함하고 조인된 테이블의 모든 컬럼을 뷰에 표시하기 위해서) 컬럼명이 하나 이상인 경우에는 새로운 이름을 할당할 필요가 있다. 이들 컬럼명은 기저 테이블의 컬럼명과 함께 시스템 카탈로그 syscolumns에 수록된다.

뷰를 삭제하는 경우는 DROP VIEW 문을 실행한다. 뷰를 삭제하면 그 뷰를 기초로 하여 정의된 뷰도 모두 삭제된다.

<뷰의 접근권한>

뷰를 작성한 경우 사용자는 기저 테이블에 대해 갖고 있는 접근 권한을 뷰에 대해서도 갖게 된다. 따라서 GRANT OPTION에 대한 특권을 갖고 있다면 뷰에 대해 다른 사용자에게 특권을 부여할 수 있다. 복수 테이블 뷰 전체에 대해 유효한 SELECT 접근권한을 갖기 위해서는 뷰에 포함되어 있는 모든 컬럼에 대해 SELECT 접근권한을 가져야 한다.

<사용자의 지위와 접근권한>

데이터베이스를 만든 사람(DBA)이 반드시 다른 사람도 사용할 수 있도록 권한 즉, 접근권한을 부여해야만 한다. 이제 이러한 접근권한과 사용자의 지위에 대해 알아보자. 여기서 다루는 개념 및 명령어는 INFORMIX에서 사용되는 것이나 다른 시스템에서도 비슷할 것이라 생각된다.

사용자의 접근권한에는 데이터베이스 레벨 권한과 테이블 레벨 권한의 두가지가 있다. 먼저 데이터베이스 레벨 접근권한에는 다음과 같은 것들이 있다.

- DBA 접근권한: 이 권한은 사용자에게 시스템 테이블 변경, 데이터베이스 삭제, CONNECT, RESOURCE, DBA접근권한 부여 및 취소등 전체 관리 권한을 부여한다.
- RESOURCE 접근권한: 이 권한은 사용자에게 모든 DDL과 DML 명령을 사용하도록 해준다.
- CONNECT 접근권한: 이 권한은 사용자에게 모든 DML 명령을 사용하도록 해준다.

데이터베이스가 작성된 시점에서 작성된 데이터베이스를 접근할 수 있는 사용자는 작성자뿐이다. 다른 사용자는 작성자가 CONNECT 명령을 이용하여 접근권한을 설정해야만 접근할 수 있다. 그러나 RESOURCE 명령에 의한 접근 권한을 부여하지 않으면 테이블과 색인을 작성하거나 삭제할 수 없다. DBA만이 이런 접근권한이 있다. DBA는 다른 사용자에게 DBA 접근권한을 줄 수도 있다. 테이블 레벨 접근권한에 대해서 뷰 테이블은 일반 테이블과 같이 취급되므로 테이블 레벨 접근권한의 대상이 됨에 유의해야 한다. 테이블 레벨의 접근권한을 사용하기 위해서는 먼저 데이터베이스 레벨의 접근권한을 부여받아야 한다. 테이블 수준의 접근권한이 있는 사용자는 SELECT문과 UPDATE문을 실행할때 특정 컬럼에 접근할 수 있다. 또 행의 첨가와 삭제, 색인의 작성 및 테이블 구조의 변경도 허용된다.

3. 다단계 데이터베이스 질의처리

데이터베이스 사용자는 뷰의 존재를 모르고 또한 뷰로부터 데이터를 액세스할 수가 없으므로 모든 액세스를 기저 테이블로부터 할 수 밖에 없다. 테이블 조작명령중 주요 명령에 대해 알아보면 다음과 같다.

<검색> 임의의 사용자가 조건 P를 만족하는 튜플들중 애트리뷰트 a1, a2, a3를 검색하기 원한다고 하자. 이 사용자는 당연히 기저 테이블인 BT에 관한 정보밖에는 가지고 있지 않으므로 BT로부터 데이터를 얻고자 다음과 같이 질의어를 작성한다.

```
SELECT a1, a2, a3
FROM BT
WHERE P
```

이와 같은 검색 구문은 사용자 보안등급에 따라 해당 뷰로부터의 검색으로 변환되어진다. 예를 들어 현 사용자의 보안등급이 U등급이라면 이 사용자는 U등급의 데이터만을 액세스할 수 있으므로

사용자 질의어는 다음과 같이 변환되어 뷰 테이블 V_u 로부터 액세스되도록 처리된다.

```
SELECT a1, a2, a3
FROM  $V_u$ 
WHERE P
```

그러나 사용자의 보안등급이 C등급이라면 이 사용자는 C등급의 데이터와 U등급의 데이터를 액세스할 수 있으므로 사용자 질의어는 다음과 같이 변환되어 처리된다.

```
(SELECT a1, a2, a3
FROM  $V_c$ 
WHERE P)
UNION
(SELECT a1, a2, a3
FROM  $V_u$ 
WHERE P)
```

또한 사용자의 등급이 S등급이라면 이 사용자는 S등급, C등급, U등급 등 모든 데이터를 액세스할 수 있으므로 이 경우에는 뷰 테이블로부터 액세스하도록 질의어를 변환할 필요 없이 바로 기저 테이블에서 액세스하면 된다.

<삽입>

데이터베이스 사용자가 새로운 튜플을 삽입하고자 할 때에는 자기가 사용중인 그 상태에서 삽입 명령을 사용하여 삽입하면 된다. 그러면 이 튜플은 기저 테이블에 그대로 삽입됨은 물론 해당 보안등급에 맞는 뷰에도 자동적으로 삽입되어 보안등급별로 자료관리가 이루어지게 된다.

<삭제>

데이터베이스 사용자가 어떤 튜플을 삭제하고자 할 때에는 자기가 사용중인 그 상태에서 삭제 명령을 사용하여 삭제할 수 있다. 그러면 이 튜플은 기저 테이블로부터 삭제됨은 물론 해당 보안등급에 맞는 뷰로부터도 자동적으로 삭제되어 보안등급별로 자료관리가 이루어지게 된다.

<갱신>

데이터베이스 사용자가 애틀리뷰트의 값을 변경하고자 할 때에는 자기가 사용중인 그 상태에서 갱신 명령을 사용하여 갱신하면 된다. 그러면 기저 테이블에 있는 해당 애틀리뷰트의 해당 값들이 갱신되면서 그 결과에 따라 갱신된 값의 보안등급에 맞는 뷰들로 자동적으로 수평분할이 재실시되게 되므로 보안등급별로 자료관리가 이루어지게 된다.

4. 시험결과 및 분석

본 논문에서 제안된 기법의 테스트는 본교의 SUN 4/490 상에서 운용되고 있는 INFORMIX DBMS [다우 90]를 이용하여 실시되었다. 먼저 다음 그림 3과 같은 기저 테이블 '자동차'를 DDL과 DML을 이용하여 생성한다.

다음 테이블에서 수급현황에 따라 보안등급이 결정되어 일시불로 완납한 사람은 S등급, 24개월 할부로 구입한 사람은 C등급, 그리고 36개월 할부로 구입한 사람은 U등급으로 분류된다고 하자. 그러면 수평분할의 개념을 적용하여 각 등급별로 뷰를 생성할 수 있다. 예를 들어 S등급의 뷰는 다음과 같은 명령으로 생성된다.

```
CREATE VIEW VS
AS SELECT *
FROM 자동차
WHERE 판매방법="일시불"
```

생성된 결과는 그림 4와 같이 VS, VC, VU의 3개의 뷰 테이블이다.

고객명	모델명	가 격	제조회사	판매방법
박 훈	액셀	5200000	현대	할부24
이정아	프라이드	4200000	기아	할부24
김수정	뉴그랜저	19000000	현대	일시불
최성한	세피아	7600000	기아	할부24
장태영	프린스	11500000	대우	일시불
송익화	티코	3500000	대우	할부36
한나영	액셀	5200000	현대	할부24
이강운	에스페로	8600000	대우	할부36
이영민	스포티지	13000000	기아	일시불
신도개발	그레이스	12500000	현대	할부36
박동환	겔로퍼	16400000	현대	할부36

< 그림 3 > 자동차 테이블

고객명	모델명	가 격	제조회사	판매방법
김수정	뉴그랜저	19000000	현대	일시불
장태영	프린스	11500000	대우	일시불
이영민	스포티지	13000000	기아	일시불

(a) VS 테이블

고객명	모델명	가 격	제조회사	판매방법
박 훈	액셀	5200000	현대	할부24
이정아	프라이드	4200000	기아	할부24
최성한	세피아	7600000	기아	할부24
한나영	액셀	5200000	현대	할부24

(b) VC 테이블

고객명	모델명	가 격	제조회사	판매방법
송익화	티코	3500000	대우	할부36
이강운	에스페로	8600000	대우	할부36
신도개발	그레이스	12500000	현대	할부36
박동환	겔로퍼	16400000	현대	할부36

(c) VU 테이블

< 그림 4 > 뷰 테이블

생성된 기저 테이블과 뷰 테이블을 다른 사용자들도 액세스할 수 있도록 하기 위해 접근권한을 부여한다. 먼저 데이터베이스 레벨에서 접근권한은 다른 사람들이 DML 명령만 사용하면 되므로 시스템상의 질의어 화면에서 다음과 같은 명령으로 이루어진다

GRANT CONNECT TO PUBLIC

이제 테이블 레벨의 접근권한을 부여한다. 기저 테이블 BT에 대한 주요 액세스를 다른 사용자에게도 허용하고자 하므로 다음과 같은 GRANT 명령을 이용하여 권한을 허용한다.

GRANT SELECT, UPDATE, INSERT, DELETE ON 자동차 TO PUBLIC

그러나 뷰 테이블에 대해서는 일반 사용자에게 어떠한 액세스도 허용되어서는 안되므로 다음과 같은 REVOKE 명령을 이용하여 권한을 제한해야 한다.

REVOKE ALL ON VS FROM PUBLIC

REVOKE ALL ON VC FROM PUBLIC

REVOKE ALL ON VU FROM PUBLIC

만들어진 기저 테이블과 뷰 테이블상에서 질의어는 다음과 같이 수행된다. 사용자의 보안등급은 이미 규정되어져 있고 이를 인식하여 질의어를 변환해 주는 모듈은 개발되어 있다고 하자. 예를 들면 ID가 1000단위이면 S등급, 2000단위이면 C등급, 그리고 3000단위이면 U등급이다.

(1) SELECT 명령

ID가 2020번인 사용자가 다음 그림 5(a)와 같은 질의를 수행시켰다. 이 질의어는 C등급의 질의어이므로 이는 내부적으로 그림 5(b)와 같은 뷰 테이블에 대한 질의로 변환되어 그림 5(c)와 같은 결과를 얻게 된다.

```
SELECT 고객명, 모델명, 제조회사, 가격
FROM 자동차
WHERE 제조회사 LIKE "현대"
(a) 사용자 질의어
```

```
(SELECT 고객명, 모델명, 제조회사, 가격
FROM VC
WHERE 제조회사 LIKE "현대")
UNION
(SELECT 고객명, 모델명, 제조회사, 가격
FROM VU
WHERE 제조회사 LIKE "현대")
(b) 변환된 질의어
```

고객명	모델명	가 격	제조회사	판매방법
박 훈	액셀	5200000	현대	할부24
한나영	액셀	5200000	현대	할부24
신도개발	그레이스	12500000	현대	할부36
박동환	갤로퍼	16400000	현대	할부36

(c) 수행된 결과

< 그림 5 > 검색질의 수행

(2) INSERT 명령

다음의 레코드 <김선달, 서울, 02-555-6666, 소나타, 11000000, 현대, 92.6.1, 할부36>를 삽입해 보자. 이 레코드를 기저 테이블에 삽입하기 위해서는 다음과 같은 삽입 명령을 이용하면 된다.

```
INSERT INTO 자동차
VALUES ("김선달", "소나타", 11000000, "현대", "할부36")
```

이 레코드는 U등급의 정보를 나타내고 있지만 이 등급에 대한 정보는 삽입시에는 없어도 된다. 왜냐하면 사용자는 자신의 등급에 맞는 데이터만을 다루고 또한 기저 테이블은 자신의 등급에 해당하는 데이터들로만 구성되어 있다고 생각하기 때문이다. 삽입된 결과를 검사해보기 위한 질의어 및 그 결과는 다음 그림 6과 같다.

```
SELECT *
FROM 자동차
(a) 확인용 질의어

SELECT *
FROM VU
(b) 변환된 질의어
```

고객명	모델명	가 격	제조회사	판매방법
송익화	티코	3500000	대우	할부36
이강운	에스페로	8600000	대우	할부36
신도개발	그레이스	12500000	현대	할부36
박동환	갤로퍼	16400000	현대	할부36
김선달	소나타	11000000	현대	할부36

(c) 수행된 결과

< 그림 6 > 레코드 삽입

(3) UPDATE 명령

고객명이 송익화인 사람이 수금 방법을 36할부에서 일시불로 완납하였다. 이에 따른 갱신이 수행되는 절차는 다음과 같다.

```
UPDATE 자동차
SET 판매방법="일시불"
WHERE 고객명="송익화"
```

수행된 결과는 다음 그림 7, 그림 8에서 확인된다. 이 레코드는 36개월 할부에서 일시불로 바뀌었으므로 U등급에서 S등급으로 바뀌어 관리됨을 알 수 있다.

```
SELECT *
FROM 자동차
(a) 확인용 질의어
```

```
SELECT *
FROM VU
(b) 변환된 질의어
```

고객명	모델명	가 격	제조회사	판매방법
이강운	에스페로	8600000	대우	할부36
신도개발	그레이스	12500000	현대	할부36
박동환	갤로퍼	16400000	현대	할부36
김선달	소나타	11000000	현대	할부36

(c) 수행 결과

< 그림 7 > U등급 테이블 확인

```
SELECT *
FROM 자동차
(a) 확인용 질의어
```

```
SELECT *
FROM VS
(b) 변환된 질의어
```

고객명	모델명	가 격	제조회사	판매방법
김수정	뉴그랜저	19000000	현대	일시불
장태영	프린스	11500000	대우	일시불
송익화	티코	3500000	대우	일시불
이영민	스포티지	13000000	기아	일시불

(c) 수행결과

< 그림 8 > S등급 테이블 확인

(4) DELETE 명령

삭제시에는 사용자의 등급에 따라 달라지는 것이 없고 모든 변화가 기저테이블에서 수행된다.

5. 결론

본 논문에서는 뷰의 개념과 수평분할을 이용한 다단계 데이터베이스에서의 질의 처리 기법을 제시하였다. 수평분할과 뷰를 이용한 다단계 데이터베이스에서는 먼저 데이터베이스를 보안등급별로 수평분할을 실시하여 뷰로 저장되므로 모든 등급의 사용자는 항상 기저 테이블을 자신의 보안 등급에 맞는 테이블로만 인식하고 이를 액세스하나 실제로는 뷰 테이블에 의해 등급별로 처리되어 진다. 본 논문에서는 이러한 데이터베이스에서의 질의 처리 기법을 검색, 삽입, 갱신, 삭제별로 구분하여 제시하고 INFORMIX상에서 각 경우별로 설명하였다.

앞으로 수행되어야 할 연구로써는 사용자의 보안등급을 인식하여 사용자 질의어를 적절한 형태로 변환해 주는 질의어 변환기에 관한 연구가 계속되어야 할 것이다. 특히 질의어 변환기의 구현은 [Thur93]에서처럼 추론 기능을 첨가하여 구현되어야 할 것으로 생각된다.

참고문헌

- [Jajo 90a] Jajodia, S., Sandhu, R., and Sibley, E., "Update Semantics for Multilevel Relations," The Sixth Computer Security Applications Conference, pp. 103-112, December 1990.
- [Jajo 90b] Jajodia, S., and Landwehr, C., Database Security IV, North Holland, 1990.
- [Lunt 92] Lunt, T. F., Research Directions in Database Security, Springer-Verlag, 1992.
- [Smit 90] Smith G. W., "The Semantic Data Model for Security : Representing the Security Semantics of an Application," International Conference on Data Engineering, pp.322-329, February 1990.
- [Thur 93] Thuraingham, B., et al., "Design and Implementation of a Database Inference Controller," Data & Knowledge Engineering (11), 1993.
- [Wagn 90] Wagner, N. R., Fountain, R. L., and Hazy R. J., "The Fingerprinted Database," International Conference on Data Engineering, pp.330-336, February 1990.
- [나 민영 94] 나 민영, 임 웅택, "수평분할과 뷰를 이용한 다단계 데이터베이스 보안기법", 육사 논문집 제47집, 1994.
- [다우 90] INFORMIX-SQL Training Manual, 다우기술, 1990.