제 10 장 데이터베이스 보안과 권한 관리

10장. 데이터베이스 보안과 권한 관리

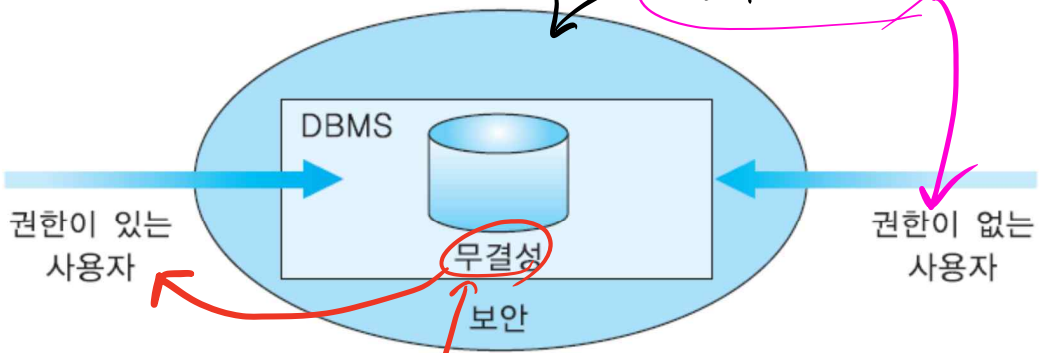
❑ 데이터베이스 보안과 권한 관리

- ✓ 데이터베이스가 손실되면 데이터베이스를 소유한 조직체의 운영에 심대한 지장을 초래할 수 있으므로 권한이 없는 사용자로부터 데이터베이스를 보호하는 것이 중요함
- ✓ 데이터베이스에서 릴레이션을 생성하면 생성자를 제외한 다른 사용자들은 그 릴레이션을 접근할 수 없음
- ✓ 공유 데이터베이스에 생성된 릴레이션들은 일반적으로 여러 사용자들이 접근할 수 있도록 권한을 허가함
- ✓ DBMS는 릴레이션의 생성자가 다른 사용자들에게 적절한 수준의 권한을 허가하고, 허가한 권한을 취소하는 권한 관리 기법을 제공함

10.1 데이터베이스 보안

- 세 가지 유형의 보안
  - ✓ 물리적 보호
    - 화재, 홍수, 지진 등과 같은 자연 재해, 도둑, 컴퓨터 시스템에 대한 우연한 손상, 데이터에 손상을 주는 기타 유형의 위험으로부터 데이터베이스를 보호하는 것
  - ✓ 권한 보호
    - 권한을 가진 사용자만 특정한 접근 모드로 데이터베이스를 접근할 수 있도록 보호
  - ✓ 운영 보호
    - 데이터베이스의 무결성에 대한 사용자 실수의 영향을 최소화하거나 제거하는 조치

10.1 데이터베이스 보안(계속)



[그림 10.1] 무결성과 보안

※ 데이터 베이스 보안
① 무결성
② 일의 보안 기법.

일의 보안 기법

권한이 있는 사용자가
갱신을 잘못하는 경우를 방지한다.

10.1 데이터베이스 보안(계속)

- DBMS가 데이터베이스 보안과 관련하여 제공해야 하는 두 가지 기능
  - ✓ 접근 제어(access control)
    - 데이터베이스 시스템에 대한 접근을 통제할 수 있는 기능
    - DBMS는 로그인 과정을 통제하기 위하여 사용자 계정과 암호를 관리함
  - ✓ 보안 및 권한 관리
    - DBMS는 특정 사용자 또는 사용자들의 그룹이 지정된 데이터베이스 영역만 접근할 수 있고 그 외의 영역은 접근할 수 없도록 통제하는 기능을 제공함

10.1 데이터베이스 보안(계속)

- 두 가지 보안 기법
  - 임의 보안 기법(discretionary security mechanism)
    - 사용자들에게 특정 릴레이션, 투플, 또는 애트리뷰트를 지정된 모드(예를 들어, 읽기, 삽입, 삭제, 또는 수정)로 접근할 수 있는 권한을 허가하고(grant) 취소하는(revoke) 기법
    - 대부분의 상용 관계 DBMS에서 사용되는 기법
    - DBMS는 시스템 카탈로그에 누가 권한을 허가받았고 권한을 취소당했는가를 유지함

10.1 데이터베이스 보안(계속)

- 두 가지 보안 기법(계속)
  - ✓ 강제 보안 기법(mandatory security mechanism)
    - 데이터와 사용자들을 다양한 보안 등급(1급 비밀, 2급 비밀, 3급 비밀, 일반 정보 등)으로 분류하고 해당 조직에 적합한 보안 정책을 적용하여 다단계 보안을 시행하기 위해 사용됨
    - 아직 대부분의 상용 관계 DBMS는 이런 보안 기법을 제공하지 않음

10.1 데이터베이스 보안(계속)

- 데이터베이스 보안을 위해 데이터베이스 관리자가 수행하는 작업
  - ✓ 사용자 또는 사용자들의 그룹에 대한 새로운 계정과 암호의 생성, 권한 부여와 취소, 특정 계정에 대한 특정 권한의 부여와 취소 등
  - ✓ 각 로그인 세션 동안 사용자가 데이터베이스에 가한 모든 연산들을 기록할 수 있음
  - ✓ 권한이 없는 사용자가 데이터베이스를 갱신했다는 의심이 들면 데이터베이스 감사를 실시함
    - 데이터베이스 감사는 특정 기간 동안 데이터베이스에서 수행된 모든 연산들을 검사하기 위해서 시스템 로그를 조사하는 것

10.2 권한 관리

❑ 권한 허가

- ✓ 서로 다른 객체들에 대해서 다양한 권한들이 존재함
- ✓ 객체의 생성자(소유자)는 객체에 대한 모든 권한을 가짐
- ✓ 생성자는 자신이 소유한 임의의 객체에 대한 특정 권한을 GRANT문을 사용하여 다른 사용자나 역할에게 허가할 수 있음

GRANT문의 형식

```
GRANT    권한 [(애트리뷰트들의 리스트)]
ON       객체
TO       {사용자 | 역할 | PUBLIC}
[WITH GRANT OPTION];
```

(테이블 명, 뷰들, 애트리뷰트)

↳ 권한을 받은 사용자가 자신의 권한을 다른 사용자에게 양도 가능.

10.2 권한 관리(계속)

❑ 권한 허가(계속)

- ✓ GRANT절에 SELECT, INSERT, DELETE, UPDATE, REFERENCES 중 한 개 이상의 권한을 포함할 수 있음
- ✓ UPDATE문을 사용하여 애트리뷰트를 수정하려면 그 애트리뷰트에 대한 UPDATE 권한이 필요
- ✓ 릴레이션을 참조하는 외래 키 제약 조건을 만들려면 해당 릴레이션에 대해 REFERENCES 권한이 필요
- ✓ 만일 어떤 사용자가 WITH GRANT OPTION절과 함께 권한을 허가받았으면 그 사용자도 WITH GRANT OPTION과 함께 또는 WITH GRANT OPTION 없이 그 권한을 다른 사용자에게 허가할 수 있음
- ✓ 기본 릴레이션의 소유자가 다른 사용자들이 릴레이션에 직접 접근하지 못하게 하려는 경우에는 릴레이션 자체에 대한 권한은 허가하지 않고, 릴레이션을 참조하는 뷰를 정의한 후 이 뷰에 대해 권한을 부여할 수 있음

10.2 권한 관리(계속)

예1 : WITH GRANT OPTION 없이 SELECT 권한 허가

사용자 KIM이 자신이 소유한 EMPLOYEE 릴레이션에 대한 SELECT 권한을 사용자 LEE에게 허가한다.

```
GRANT   SELECT
ON      EMPLOYEE
TO      LEE;
```

LEE는 WITH CHECK OPTION 없이 SELECT 권한을 허가받았기 때문에 다른 사용자(예, CHOI)에게 권한을 다시 허가할 수 없다.

10.2 권한 관리(계속)

예2 : WITH GRANT OPTION 없이 특정 애트리뷰트들을 수정할 수 있는 권한을 허가

사용자 KIM이 자신이 소유한 EMPLOYEE 릴레이션의 TITLE과 MANAGER 애트리뷰트에 대한 UPDATE 권한을 사용자 LEE에게 허가한다.

```
GRANT  UPDATE (TITLE, MANAGER)
ON     EMPLOYEE
TO     LEE;
```

예3 : REFERENCES 권한 허가

사용자 KIM이 자신이 소유한 EMPLOYEE 릴레이션의 기본 키 애트리뷰트인 EMPNO에 대한 REFERENCES 권한을 사용자 CHOI에게 허가한다.

```
GRANT  REFERENCES (EMPNO)
ON     EMPLOYEE
TO     CHOI;
```

10.2 권한 관리(계속)

예4 : WITH GRANT OPTION과 함께 권한 허가

사용자 KIM이 자신이 소유한 DEPARTMENT 릴레이션에 대한 SELECT와 INSERT 권한을 WITH GRANT OPTION과 함께 사용자 LEE에게 허가한다.

```
GRANT  SELECT, INSERT
ON     DEPARTMENT
TO     LEE
WITH GRANT OPTION;
```

LEE는 다시 이 권한들을 다른 사용자들에게 WITH GRANT OPTION과 함께 또는 WITH GRANT OPTION 없이 허가할 수 있다. 따라서 이렇게 권한을 허가받은 사용자들의 긴 체인이 형성될 수 있다.

10.2 권한 관리(계속)

예5 : 모든 사용자들에게 권한 허가

사용자 KIM이 자신이 생성한 EMPLOYEE 릴레이션에 대한 SELECT 권한을 모든 사용자들에게 허가한다. PUBLIC이라고 부르는 특별한 사용자는 모든 사용자를 의미한다.

```
GRANT  SELECT
ON     EMPLOYEE
TO     PUBLIC;
```

10.2 권한 관리(계속)

❑ 권한 취소

- ✓ 다른 사용자에게 허가한 권한을 취소하기 위해서 REVOKE문을 사용함
- ✓ 만일 어떤 사용자가 다른 사용자에게 허가했던 권한을 취소하면, 권한을 취소 당한 사용자가 WITH GRANT OPTION을 통해서 다른 사용자에게 허가했던 권한들도 연쇄적으로 취소됨
- ✓ 취소하려는 권한을 허가했던 사람만 그 권한을 취소할 수 있음
- ✓ 권한을 허가했던 사람은 자신이 권한을 허가했던 사용자로부터만 권한을 취소할 수 있음

객체에 대한 권한.

REVOKE문의 형식

```
REVOKE  {권한들의 리스트 | ALL}
ON      객체
FROM    {사용자 | 역할 | PUBLIC};
```

10.2 권한 관리(계속)

예6 : 객체 권한을 취소

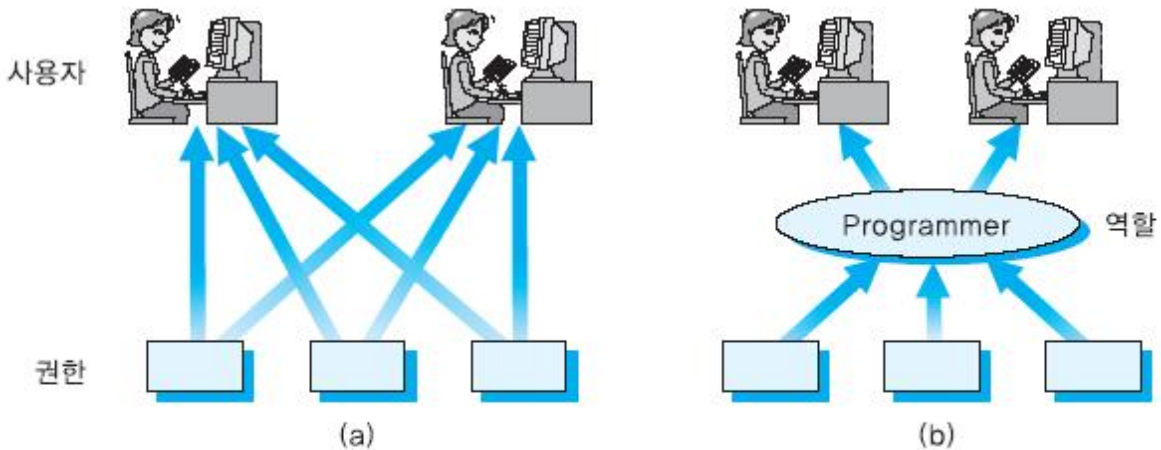
사용자 KIM이 DEPARTMENT 릴레이션에 대해 LEE에게 허가한 SELECT, INSERT 권한을 취소한다.

```
REVOKE    SELECT, INSERT
ON        DEPARTMENT
FROM      LEE;
```

10.2 권한 관리(계속)

- 역할(role)
  - ✓ 여러 사용자들에 대한 권한 관리를 단순화하기 위해 역할을 사용함
  - ✓ 역할은 사용자에게 허가할 수 있는 연관된 권한들의 그룹으로서 이름을 가짐
  - ✓ 각 사용자는 여러 역할들에 속할 수 있으며 여러 사용자들이 동일한 역할을 허가받을 수 있음
  - ✓ 동일한 권한들의 집합을 여러 사용자들에게 허가하는 대신에 이 권한들을 역할에게 허가하고, 역할을 각 사용자에게 허가함
  - ✓ 어떤 역할과 연관된 권한들에 변화가 생기면 그 역할을 허가받은 모든 사용자들은 자동적으로 즉시 변경된 권한들을 가지게 됨
  - ✓ 역할을 생성하는 방법은 DBMS마다 차이가 있음
  - ✓ 오라클에서는 CREATE ROLE문을 사용하여 역할을 생성함

10.2 권한 관리(계속)



[그림 10.2] 역할 (a) 역할 없이 권한을 허가 (b) 역할을 사용하여 권한을 허가

10.2 권한 관리(계속)

- 예: programmer 역할에게 CREATE TABLE 권한을 부여

```
GRANT CREATE TABLE
TO    programmer;
```

- 예: 사용자 CHOI에게 programmer 역할을 허가

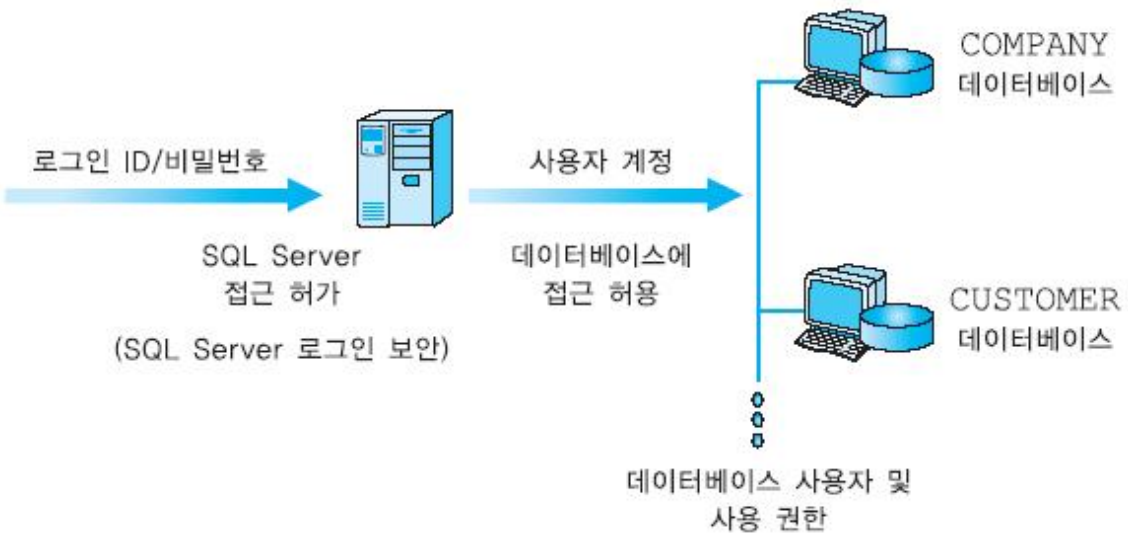
```
GRANT programmer
TO    CHOI;
```

10.3 SQL Server의 보안 및 권한 관리

❑ SQL Server의 보안 및 권한 관리의 개요

- ✓ SQL Server사용자는 사용 권한을 검사하는 두 단계를 거침
- ✓ 먼저 로그인 계정을 사용하여 사용자를 확인하고 SQL Server 인스턴스에 연결할 수 있는가를 확인함
- ✓ 인증된 사용자는 SQL Server 인스턴스에 연결됨
- ✓ 사용자가 접근해야 하는 각 데이터베이스마다 사용자 계정에 지정해야 함
- ✓ 사용자는 각 데이터베이스에서 사용자 로그인과 연결된 계정을 사용하여 서버의 데이터베이스에 접근할 수 있는 권한이 필요함

10.3 **SQL Server**의 보안 및 권한 관리(계속)



[그림 10.3] 사용 권한 검사의 두 단계

10.3 SQL Server의 보안 및 권한 관리(계속)

- SQL Server의 보안 및 권한 관리의 개요(계속)
  - ✓ 데이터베이스에 있는 개체(릴레이션, 뷰, 프로시저 등)의 모든 사용 권한과 소유권은 사용자 계정에 의해 제어됨
  - ✓ SQL Server의 보안 메카니즘에는 몇 개의 역할이 미리 정의되어 있음
  - ✓ 각 역할이 갖는 권한을 필요로 하는 사용자가 있으면 사용자의 계정을 미리 정의된 역할에 추가해야 함

10.3 SQL Server의 보안 및 권한 관리(계속)

- public과 guest
  - ✓ public
    - public 역할에 부여된 권한은 데이터베이스의 모든 사용자들에게 적용됨
    - 데이터베이스의 모든 사용자들은 일단 public 역할에 속함
    - 데이터베이스의 모든 사용자들이 특정 권한을 갖게 하려면 public 역할에 권한을 할당함
  - ✓ guest
    - 어떤 데이터베이스에 별도로 생성한 사용자 계정이 없을 때 사용하는 계정