제 10 장 데이터베이스 보안과 권한 관리

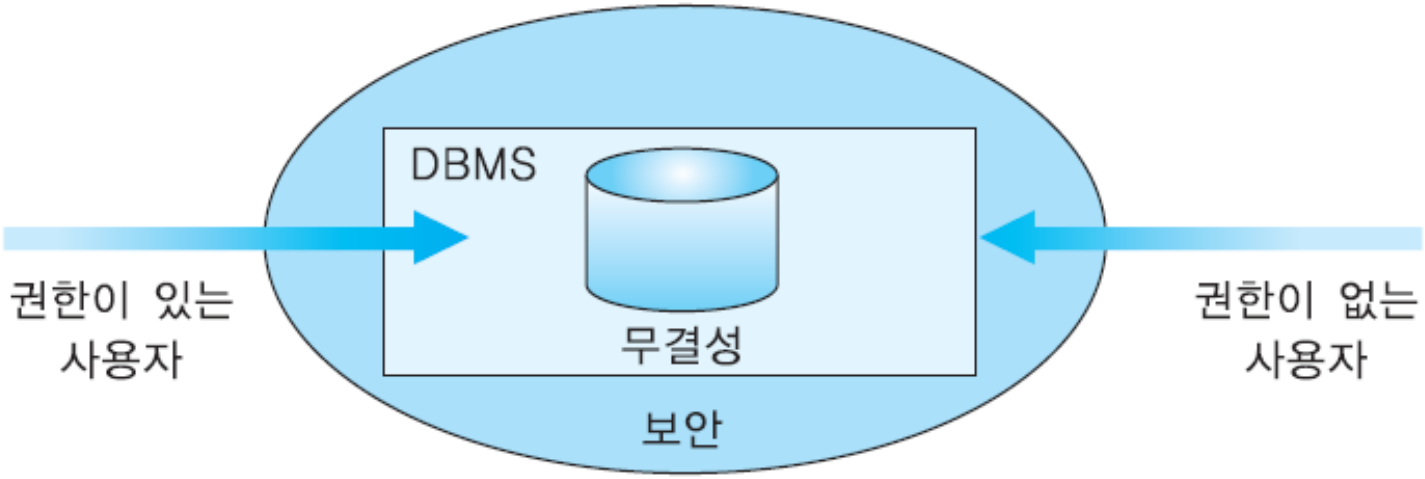
10장. 데이터베이스 보안과 권한 관리

- 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

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];
```

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.

(권한 허가) (권한 허가 불가능)

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

REVOKE문의 형식

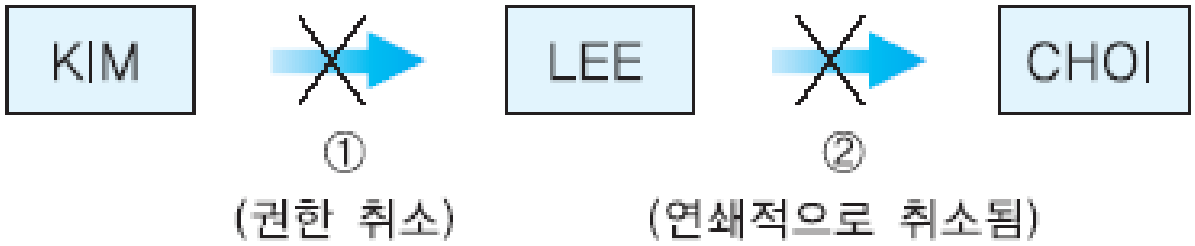
```
REVOKE  {권한들의 리스트 | ALL}
ON      객체
FROM    {사용자 | 역할 | PUBLIC};
```

10.2 권한 관리(계속)

예6 : 객체 권한을 취소

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

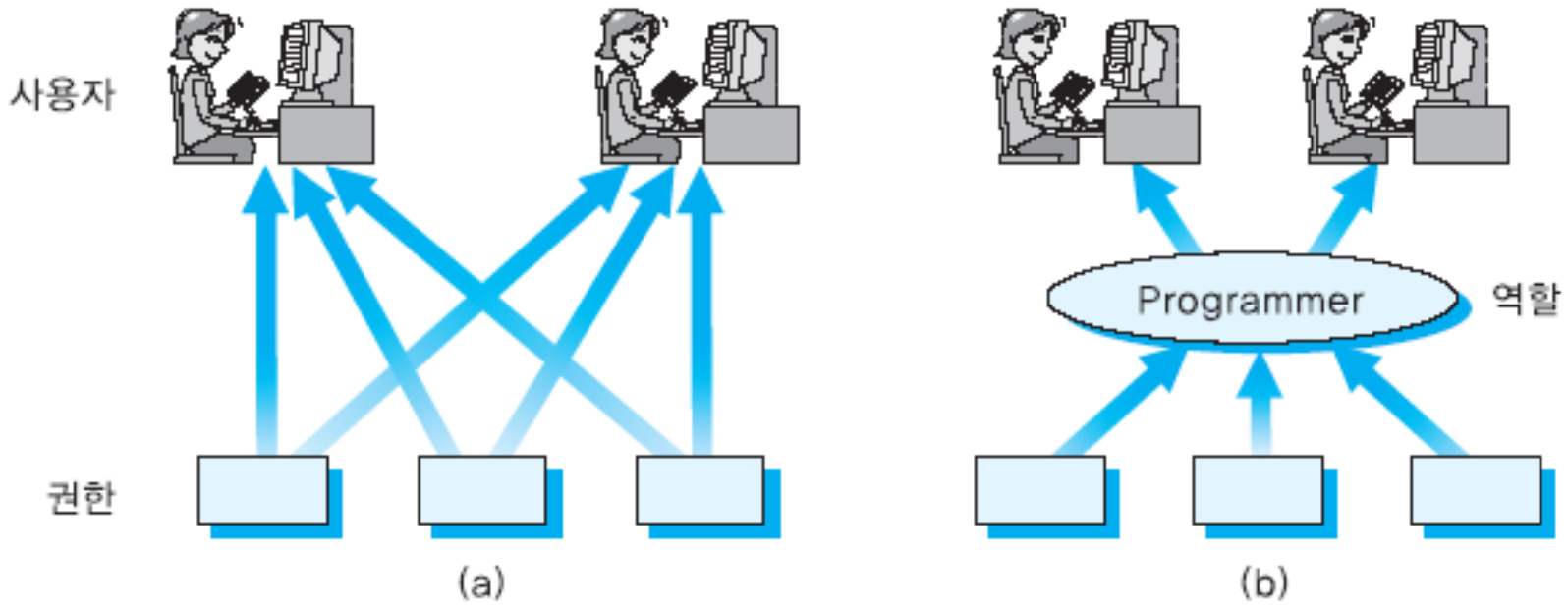
```
REVOKE    SELECT, INSERT
ON        DEPARTMENT
FROM      LEE;
```



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 오라클의 보안 및 권한 관리

❑ 오라클의 보안 및 권한 관리의 개요

- ✓ 오라클 사용자는 접속하려는 데이터베이스에 계정과 패스워드를 가져야 함
- ✓ 별도로 권한을 허가 받지 않으면 데이터베이스에서 어떤 작업도 수행할 수 없음
- ✓ 시스템 권한과 객체 권한 등 두 가지 유형의 권한이 있음
- ✓ 시스템 권한은 사용자가 데이터베이스에서 특정 작업을 수행할 수 있도록 함(예, 테이블을 생성하기 위해서는 CREATE TABLE 시스템 권한이 필요)
- ✓ 객체 권한은 사용자가 특정 객체(테이블, 뷰, 프로시저 등)에 대해 특정 연산을 수행할 수 있도록 함

10.3 오라클의 보안 및 권한 관리(계속)

〈표 10.1〉 시스템 권한의 예

유형	예
TABLE	CREATE TABLE CREATE ANY TABLE ALTER ANY TABLE DROP ANY TABLE SELECT ANY TABLE
INDEX	CREATE ANY INDEX ALTER ANY INDEX DROP ANY INDEX
TABLESPACE	CREATE TABLESPACE ALTER TABLESPACE DROP TABLESPACE
SESSION	CREATE SESSION ALTER SESSION

10.3 오라클의 보안 및 권한 관리(계속)

- 시스템 권한의 허가
  - ✓ 데이터베이스 관리자는 GRANT문을 사용하여 사용자에게 특정 시스템 권한들을 허가

```
GRANT CREATE SESSION TO KIM WITH ADMIN OPTION;
```

  - ✓ WITH ADMIN OPTION을 사용하여 시스템 권한을 허가하면 권한을 받은 사용자가 다시 이 권한을 다른 사용자에게 허가할 수 있음
  - ✓ 시스템 권한을 취소할 때는 연쇄적인 취소가 일어나지 않음

10.3 오라클의 보안 및 권한 관리(계속)

❑ 객체 권한

- ✓ 객체의 소유자는 객체에 대한 모든 권한을 보유
- ✓ 객체의 소유자는 자신의 객체에 대한 특정 권한을 다른 사용자나 역할에게 허가할 수 있음
- ✓ PUBLIC 키워드를 사용하여 권한을 허가하면 모든 사용자에게 권한을 부여하게 됨
- ✓ 각 객체마다 허가할 수 있는 권한들에 차이가 있음

10.3 오라클의 보안 및 권한 관리(계속)

〈표 10.2〉 객체에 대해 허용 가능한 권한

권한	테이블	뷰
ALTER	O	O
DELETE	O	O
EXECUTE		
INDEX	O	O
INSERT	O	O
REFERENCES	O	
SELECT	O	O
UPDATE	O	O

10.3 오라클의 보안 및 권한 관리(계속)

❑ 미리 정의된 역할

- ✓ 사용 패턴을 여러 관점에서 분석하여 각 사용 패턴에 맞게 미리 정해놓은 역할이 약 20여개 있음
- ✓ connect 역할만 있으면 오라클 데이터베이스에 로그인하고, 만일 다른 사용자의 데이터를 검색할 수 있도록 권한을 허가 받았으면 이를 검색하고, 만일 다른 사용자의 데이터를 갱신할 수 있도록 권한을 허가 받았으면 이를 갱신할 수 있음
- ✓ connect 역할과 함께 resource 역할이 있으면 테이블과 인덱스를 생성하고, 자신의 객체에 대해 다른 사용자에게 권한을 허가하거나 취소할 수 있음

10.3 오라클의 보안 및 권한 관리(계속)

〈표 10.3〉 미리 정의된 몇 개의 역할

역할	기능
connect, resource	이 역할들은 역 호환성을 위해 제공됨
dba	WITH ADMIN OPTION과 함께 모든 시스템 권한을 보유

10.3 오라클의 보안 및 권한 관리(계속)

❑ 데이터베이스 관리자 권한

✓ 데이터베이스 관리자만 관리자 권한을 가진 채 데이터베이스에 접속할 수 있어야 함

〈표 10.4〉 SYSOPER과 SYSDBA의 권한

유형	권한
SYSOPER	STARTUP SHUTDOWN ALTER DATABASE OPEN ALTER DATABASE BACKUP
SYSDBA	WITH ADMIN OPTION과 함께 SYSOPER의 권한 CREATE DATABASE

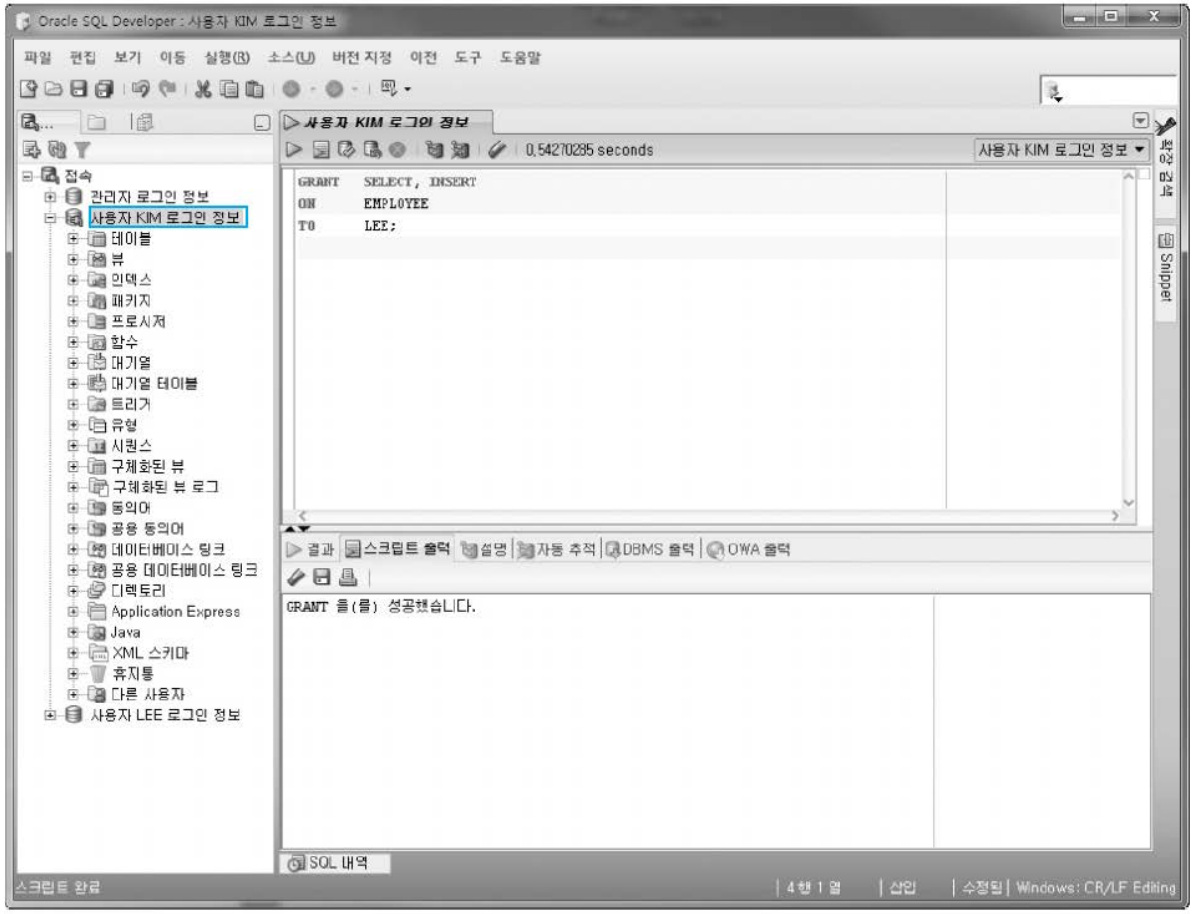
10.3 오라클의 보안 및 권한 관리(계속)

❑ 사용자 LEE에게 SELECT와 INSERT 권한 허가

- ✓ 3.3.1절에서 사용자 KIM과 LEE를 DBSERVER 데이터베이스에 등록하고 권한을 부여하였음
- ✓ 예제 3.2에서 KIM이 DEPARTMENT, EMPLOYEE 테이블을 생성하고, 이 두 테이블에 투플들을 삽입하고, EMP_PLANNING이라는 뷰를 정의하였음
- ✓ Oracle SQL Developer에 사용자 KIM으로 로그인을 하고 사용자 LEE에게 EMPLOYEE 테이블에 대한 SELECT와 INSERT 권한을 허가하기 위해 아래와 같은 GRANT문을 수행

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

10.3 오라클의 보안 및 권한 관리(계속)



[그림 10.3] LEE에게 권한 허가

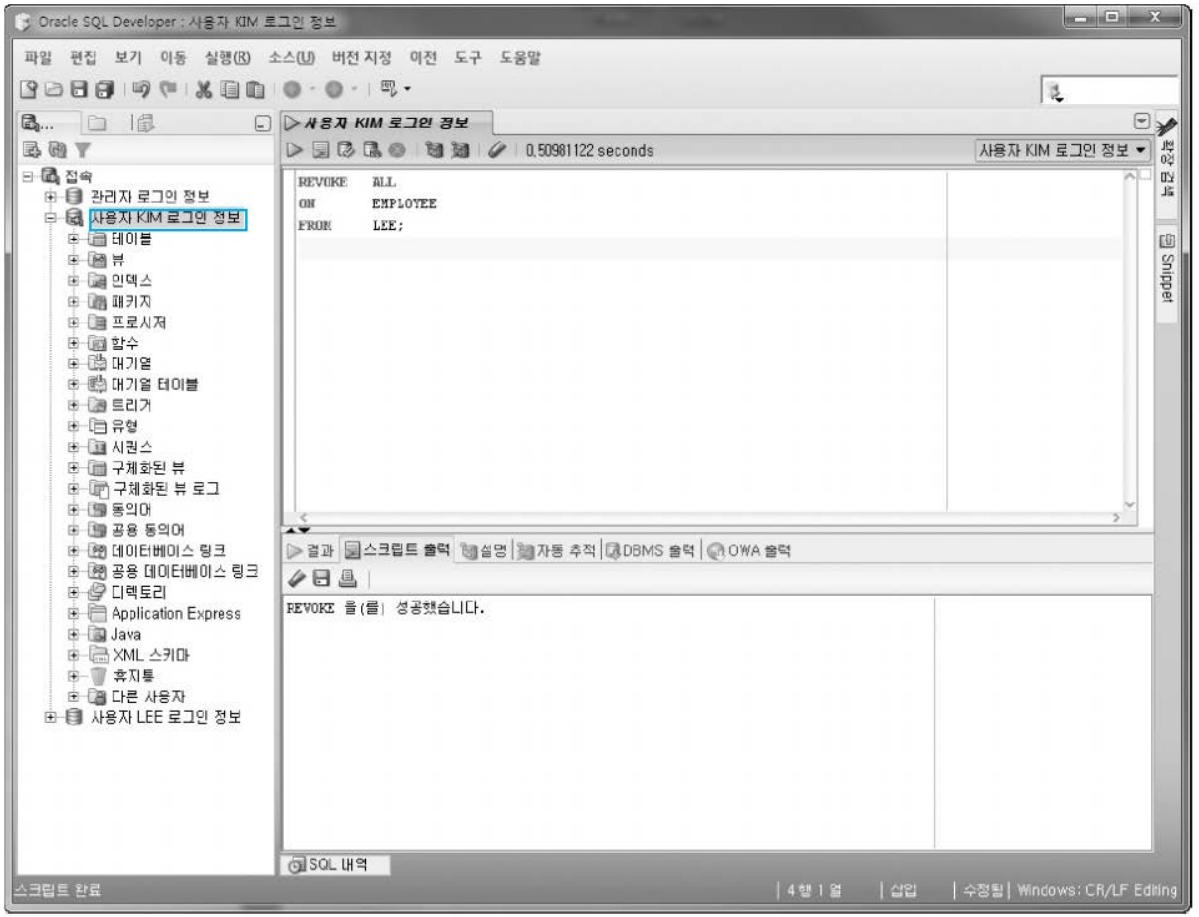
10.3 오라클의 보안 및 권한 관리(계속)

❑ 모든 권한 취소

✓ EMPLOYEE 테이블을 정의한 사용자 KIM으로 로그인을 하고, EMPLOYEE 테이블에 대한 모든 권한을 취소하려면 아래와 같은 REVOKE문을 수행

```
REVOKE  ALL
ON      EMPLOYEE
FROM    LEE;
```

10.3 오라클의 보안 및 권한 관리(계속)



[그림 10.4] LEE에게 허가한 모든 권한 취소

10.3 오라클의 보안 및 권한 관리(계속)

❑ 일부 권한 취소

✓ EMPLOYEE 테이블에 대한 INSERT 권한만 취소하려면 아래와 같은 REVOKE문을 수행

```
REVOKE  INSERT
ON      EMPLOYEE
FROM    LEE;
```

10.3 오라클의 보안 및 권한 관리(계속)

❑ 모든 권한 취소

✓ 데이터베이스 관리자로 로그인을 해서 아래의 REVOKE문을 수행하면 LEE에게 허가된 전체 권한이 취소됨

```
REVOKE  ALL PRIVILEGES
FROM    LEE;
```

10.3 오라클의 보안 및 권한 관리(계속)

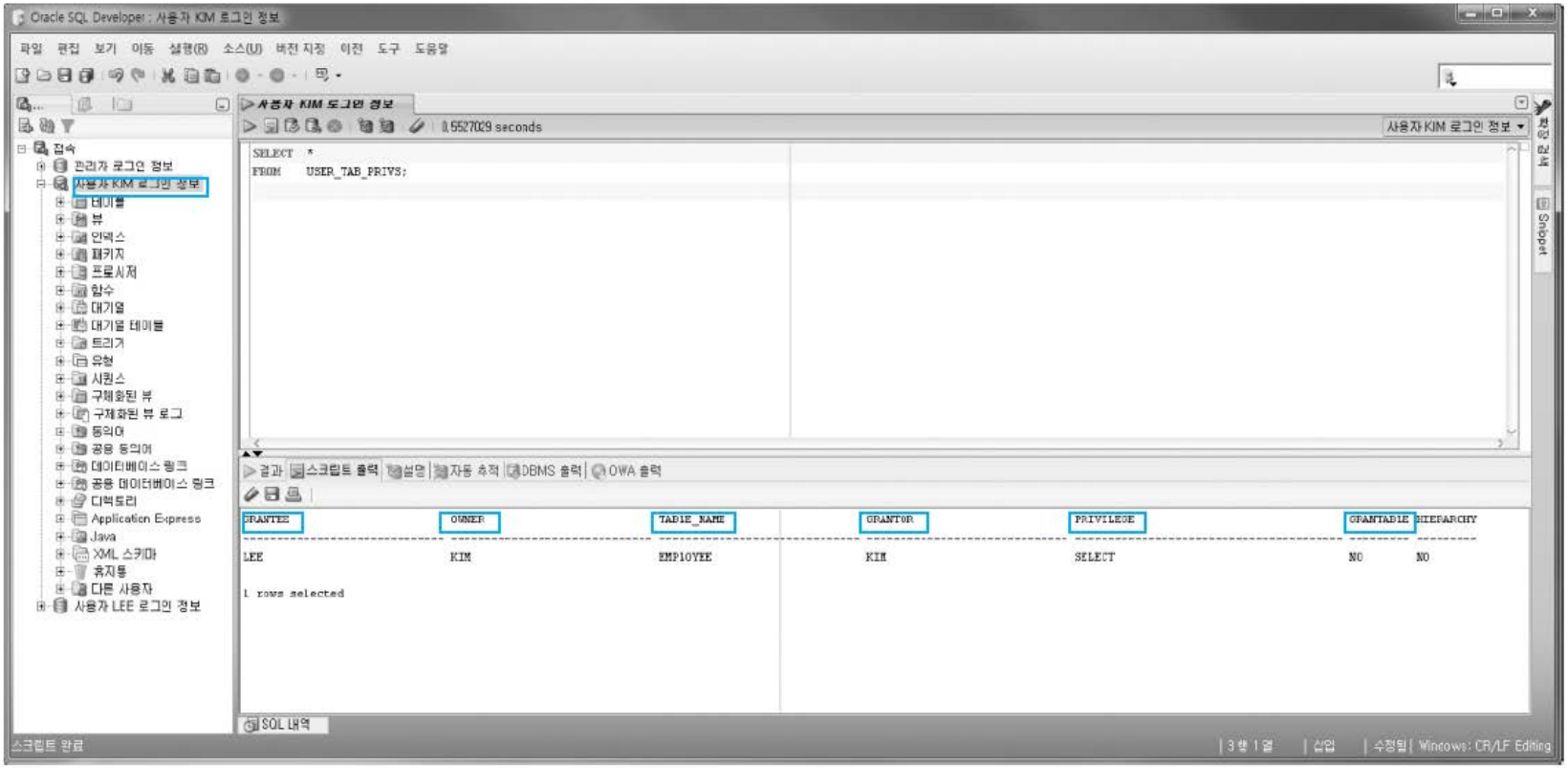
❑ 데이터 사전 뷰를 사용하여 권한에 관련된 정보를 검색

✓ USER_TAB_PRIVS 데이터 사전 뷰 또는 DBA_TAB_PRIVS 데이터 사전 뷰를 통해서 검색할 수 있음

```
SELECT *
FROM   USER_TAB_PRIVS;
```

✓ GRANTEE는 권한을 허가 받은 사용자, OWNER는 객체의 소유자, TABLE_NAME은 테이블의 이름, GRANTOR는 권한을 허가한 사용자, PRIVILEGE는 객체에 대한 권한, GRANTABLE은 권한을 허가받은 사용자가 다시 다른 사용자에게 이 권한을 허가할 수 있는지의 여부

10.3 오라클의 보안 및 권한 관리(계속)



[그림 10.5] 허가한 권한 확인

10.3 오라클의 보안 및 권한 관리(계속)

❑ 사용자 LEE가 허가 받은 시스템 권한 확인

✓ 사용자 LEE로 로그인을 한 후에 아래의 질의를 수행

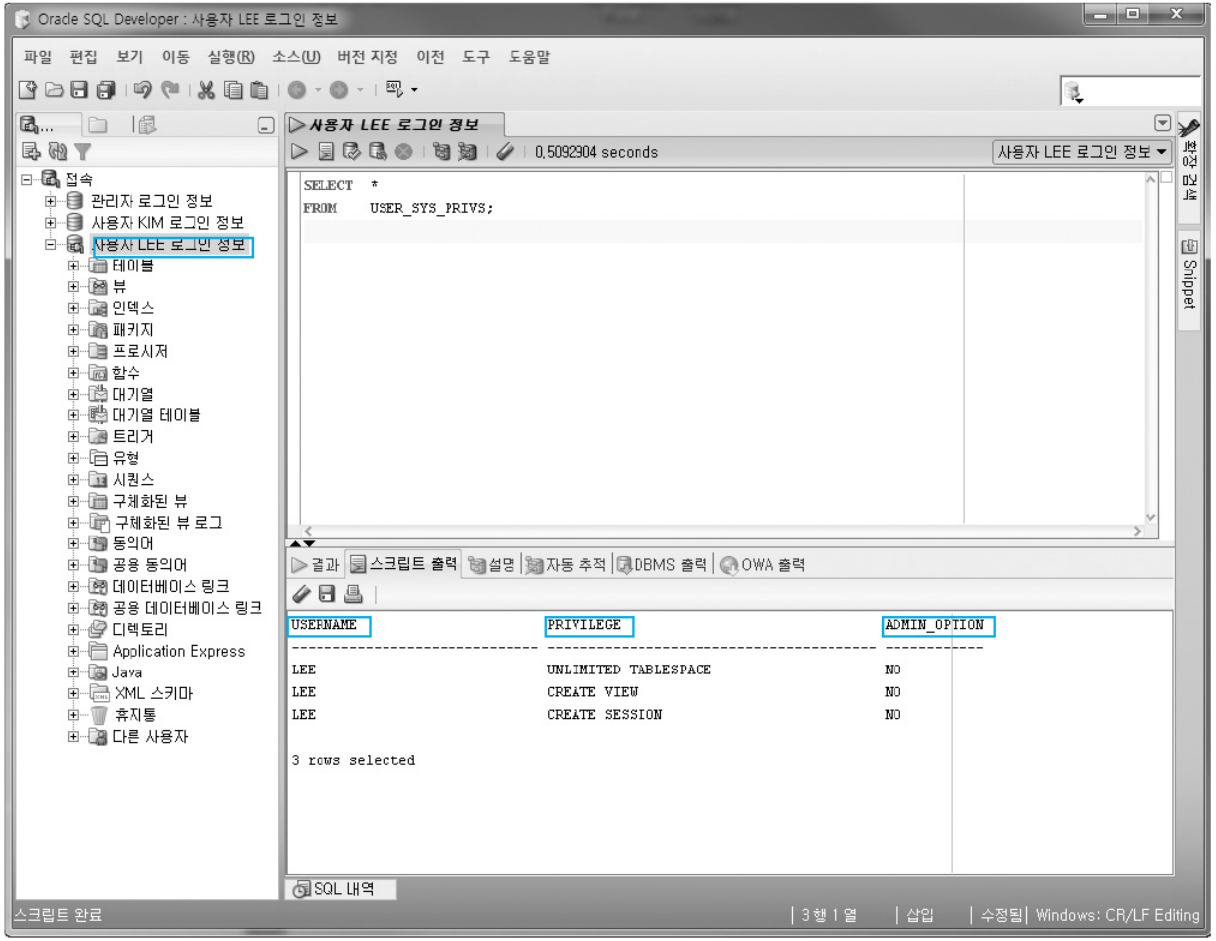
```
SELECT *
FROM   USER_SYS_PRIVS;
```

✓ USER_SYS_PRIVS는 사용자에게 허가된 시스템 권한을 보여주는 데이터 사전 뷰

✓ LEE는 UNLIMITED TABLESPACE 권한, CREATE VIEW 권한, CREATE SESSION 권한을 허가받았음을 알 수 있음

✓ USERNAME은 시스템 권한을 허가받은 사용자, PRIVILEGE는 허가받은 시스템 권한, ADMIN_OPTION은 시스템 권한을 허가받은 사용자가 다시 다른 사용자에게 이 권한을 허가할 수 있는지의 여부를 표시

10.3 오라클의 보안 및 권한 관리(계속)



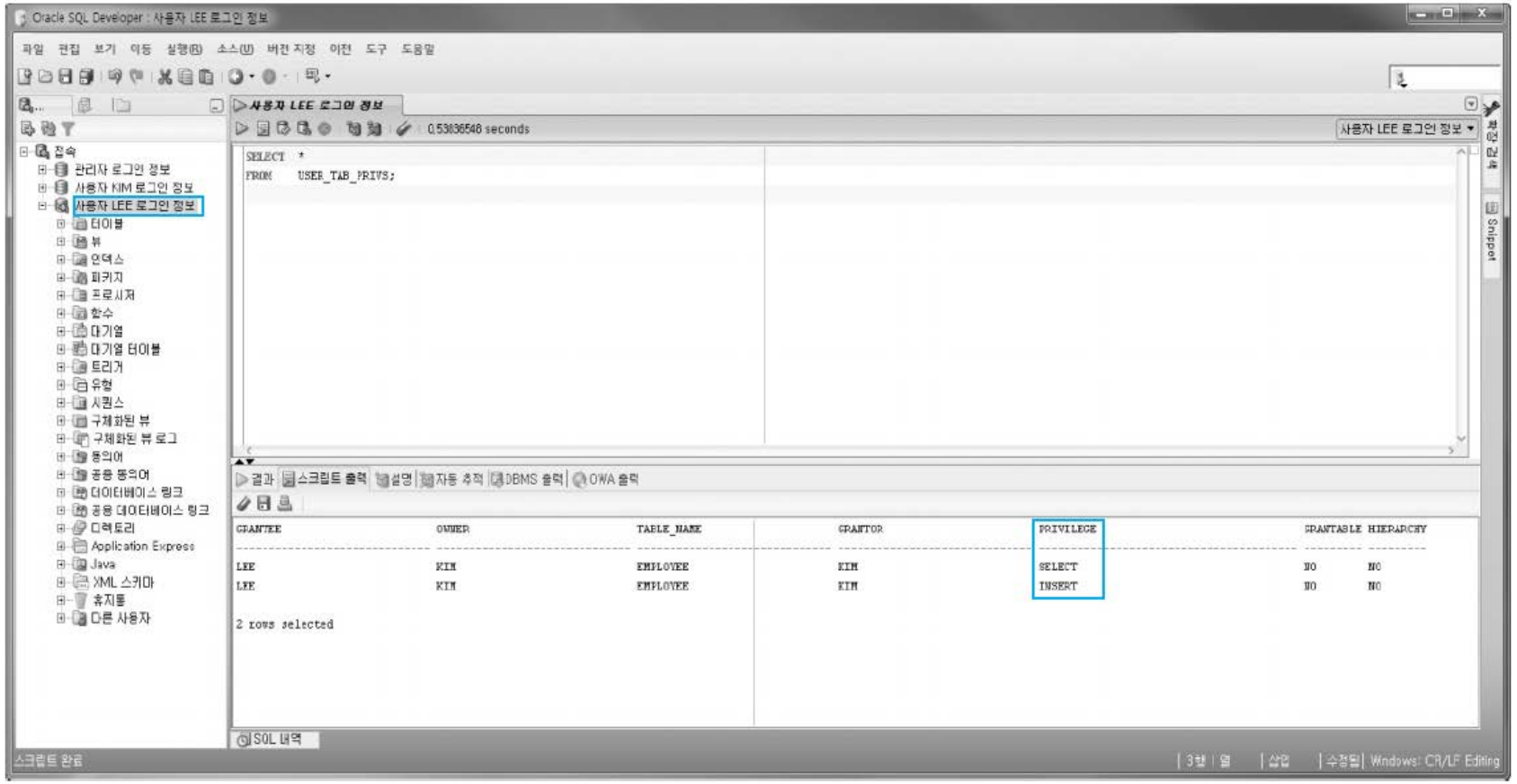
[그림 10.6] 시스템 권한 확인

10.3 오라클의 보안 및 권한 관리(계속)

- 사용자 KIM이 사용자 LEE에게 EMPLOYEE 테이블에 대한 SELECT와 INSERT 권한을 허가한 후에, 사용자 LEE가 허가 받은 객체 권한 확인
  - ✓ 사용자 LEE로 로그인을 한 후에 아래의 질의를 수행

```
SELECT *
FROM   USER_TAB_PRIVS;
```

10.3 오라클의 보안 및 권한 관리(계속)



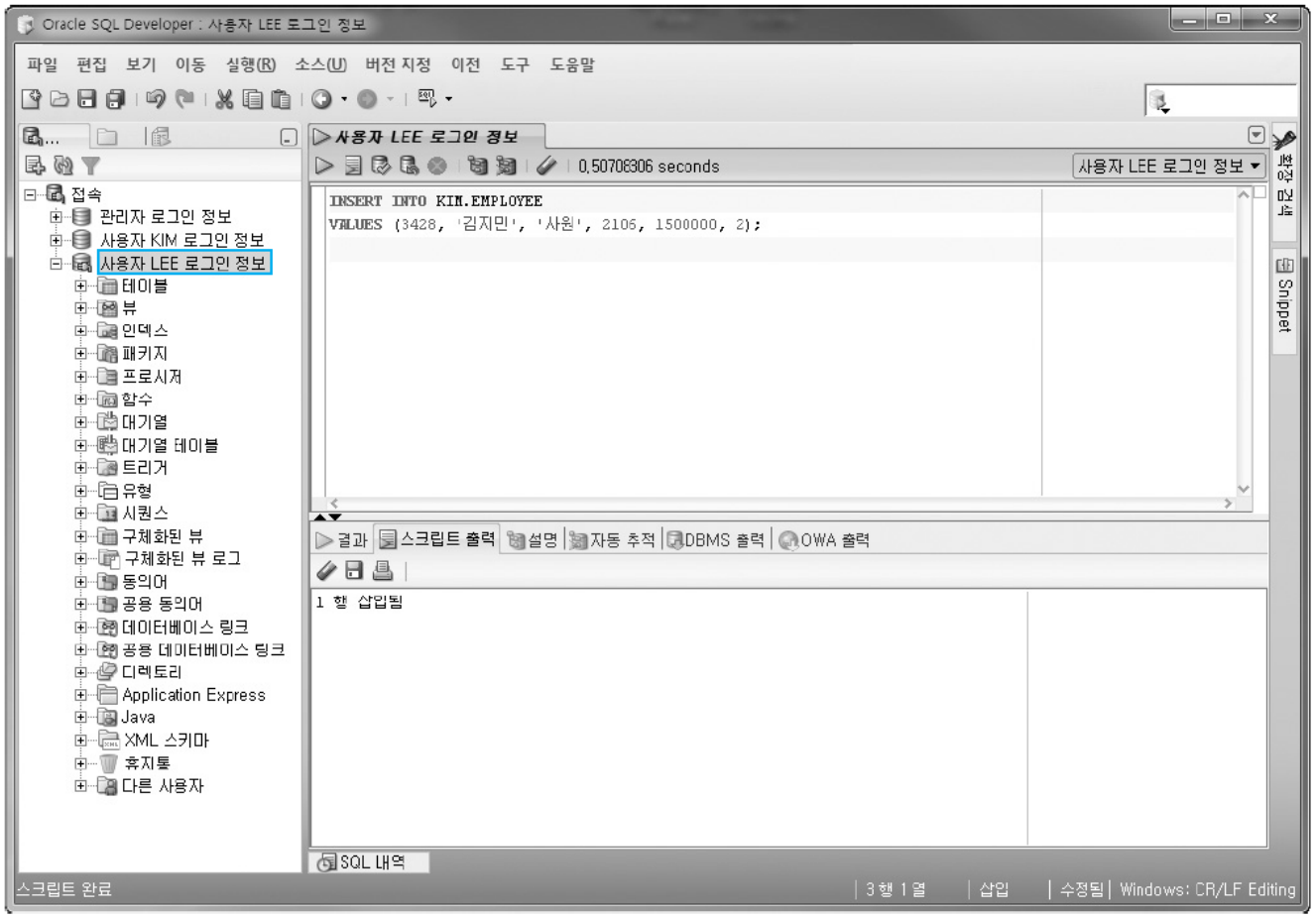
[그림 10.7] 허가받은 권한 확인

10.3 오라클의 보안 및 권한 관리(계속)

- 사용자 LEE로 로그인해서 EMPLOYEE 테이블 대한 질의들을 수행
  - ✓ 다른 소유자의 테이블을 접근하는 것이기 때문에 테이블 앞에 소유자의 계정을 붙여야 함
  - ✓ 먼저 EMPLOYEE 릴레이션에 투플을 삽입

```
INSERT INTO KIM.EMPLOYEE
VALUES(3428, '김지민', '사원', 2106, 1500000, 2);
```

10.3 오라클의 보안 및 권한 관리(계속)



[그림 10.8] KIM.EMPLOEE 테이블에 투플 삽입

10.3 오라클의 보안 및 권한 관리(계속)

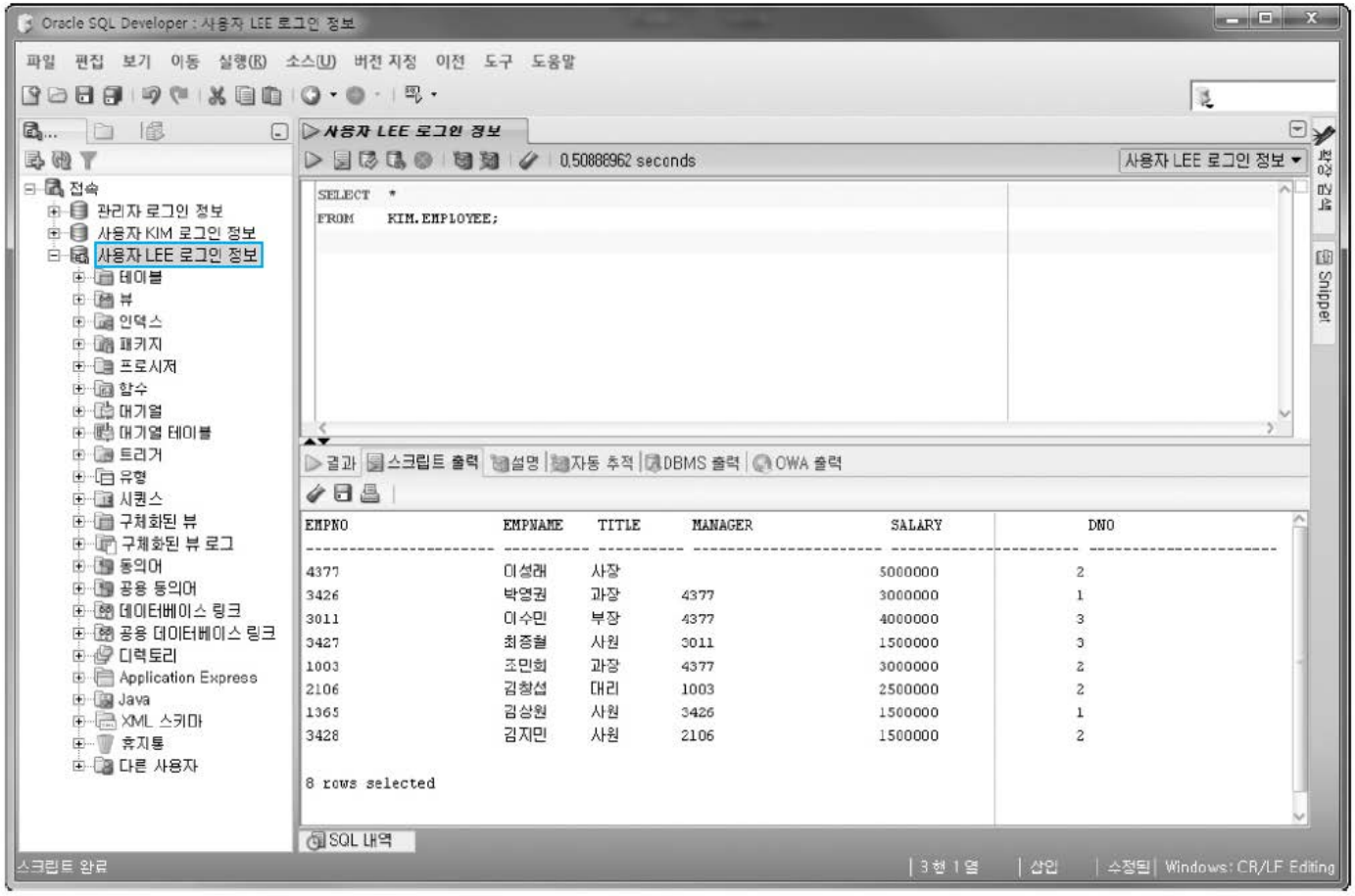
- 동의어(synonym) 정의
  - ✓ 사용자 LEE가 사용자 KIM의 EMPLOYEE 테이블을 자주 접근한다면 매번 EMPLOYEE 테이블 앞에 KIM을 붙이는 것이 번거로움
  - ✓ 동의어를 만드는 구문

```
CREATE SYNONYM 동의어 FOR 객체;
```

  - ✓ 예: 사용자 LEE가 KIM.EMPLOYEE를 EMP로 간단하게 지정

```
CREATE SYNONYM EMP FOR KIM.EMPLOYEE;
```

10.3 오라클의 보안 및 권한 관리(계속)



[그림 10.9] 삽입된 투플 확인

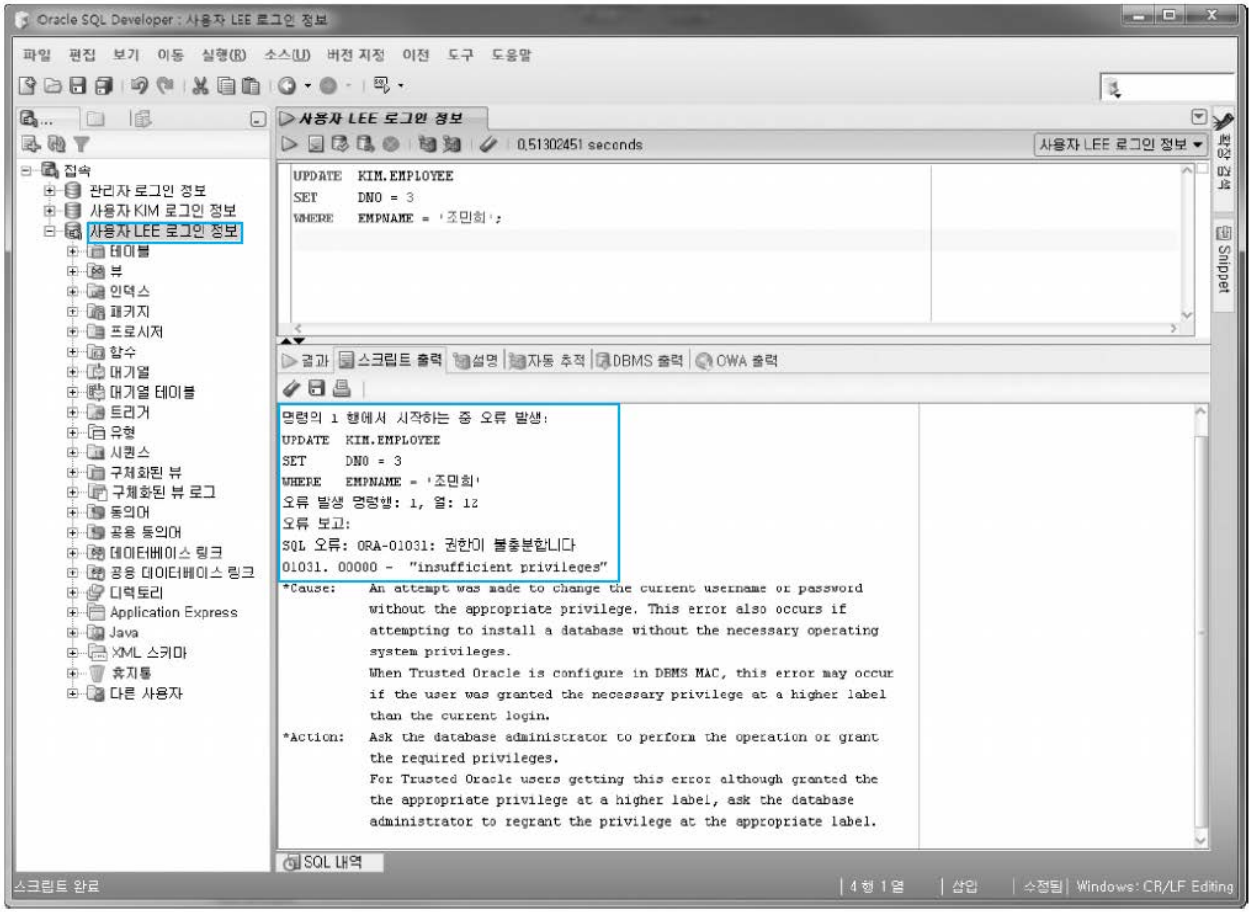
10.3 오라클의 보안 및 권한 관리(계속)

❑ EMPLOYEE 릴레이션을 수정

```
UPDATE KIM.EMPLOYEE
SET    DNO=3
WHERE  EMPNAME='조민희';
```

✓ 사용자 LEE는 EMPLOYEE 테이블의 소유자 KIM으로부터 EMPLOYEE 테이블에 대한 UPDATE 권한을 허가 받지 않았기 때문에 그림 10.10과 같이 오류 메시지가 나타남

10.3 오라클의 보안 및 권한 관리(계속)



[그림 10.10] UPDATE 권한이 없다는 오류 메시지

10.3 오라클의 보안 및 권한 관리(계속)

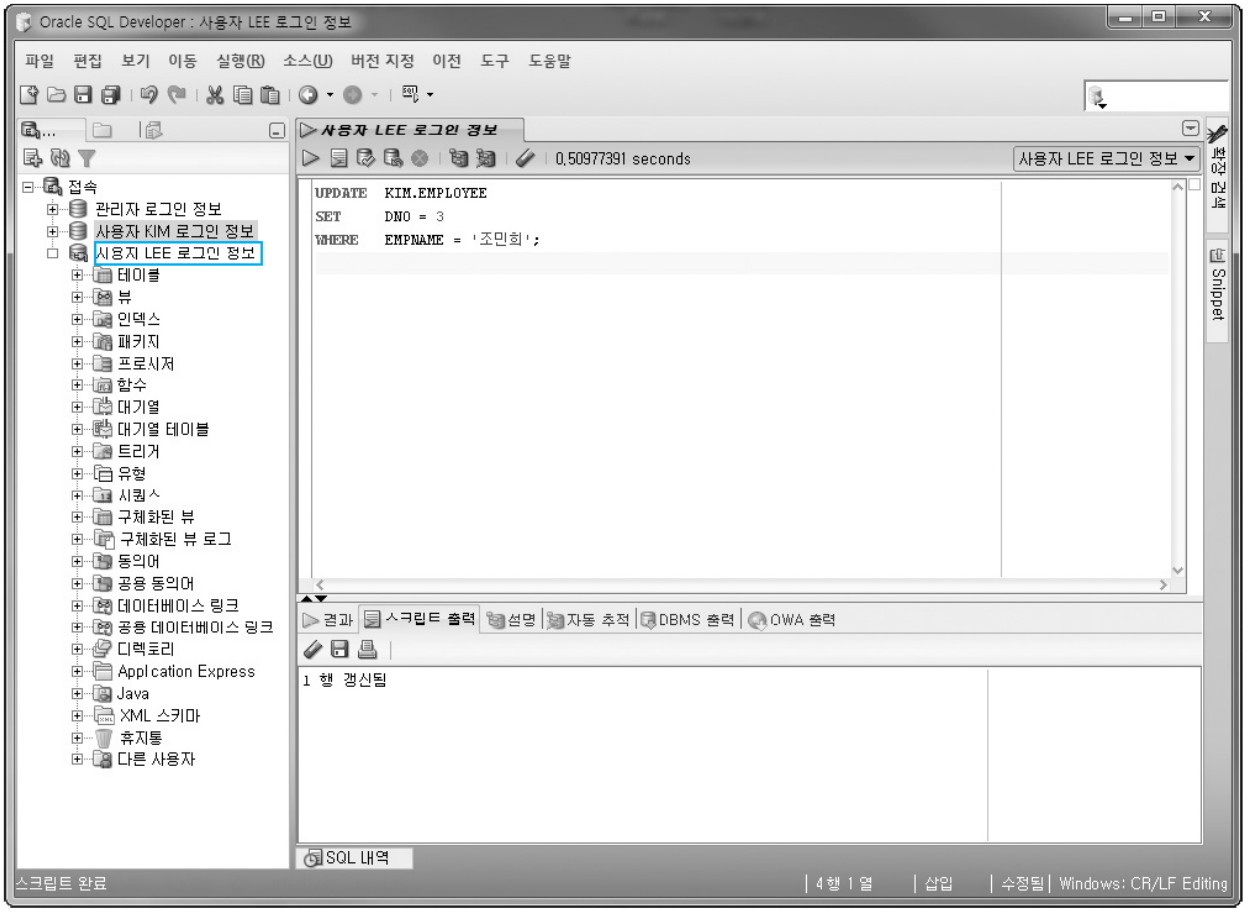
❑ UPDATE 권한을 추가로 허가

✓ 사용자 KIM으로 로그인을 한 후에 아래의 GRANT문을 수행

```
GRANT  UPDATE
ON     EMPLOYEE
TO     LEE;
```

✓ 사용자 LEE로 로그인을 한 후에 UPDATE문을 다시 수행하면 성공적으로 투플을 수정할 수 있음

10.3 오라클의 보안 및 권한 관리(계속)



[그림 10.11] UPDATE문 실행

10.3 오라클의 보안 및 권한 관리(계속)

- 사용자 LEE에 대해서 EMPLOYEE 테이블에서 SALARY 애트리뷰트를 제외한 애트리뷰트들만 SELECT할 수 있도록 하려면
  - ✓ 오라클에서는 애트리뷰트 단위로 SELECT 권한을 허가할 수 없음
  - ✓ SALARY 애트리뷰트를 제외한 나머지 애트리뷰트들을 포함하는 뷰를 정의한 후 이 뷰에 대한 SELECT 권한을 사용자 LEE에게 허가
  - ✓ INSERT, UPDATE, REFERENCES 권한은 애트리뷰트 단위로 허가할 수 있음