

데이터베이스

- 순서 : Ch10(데이터베이스 보안과 권한 관리)
- 학기 : 2018학년도 2학기
- 학과 : 가천대학교 컴퓨터공학과 2학년
- 교수 : 박양재

데이터베이스

- 목차

10.1 데이터베이스 보안

10.2 권한관리

10.3 오라클의 보안과 권한관리

10장. 데이터베이스 보안과 권한 관리

□ 데이터베이스 보안과 권한 관리

- ✓ 데이터베이스가 손실되면 데이터베이스를 소유한 조직체의 운영에 중대한 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 허가(GRANT)하고, 허가한 권한을 취소(REVOKE)하는 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

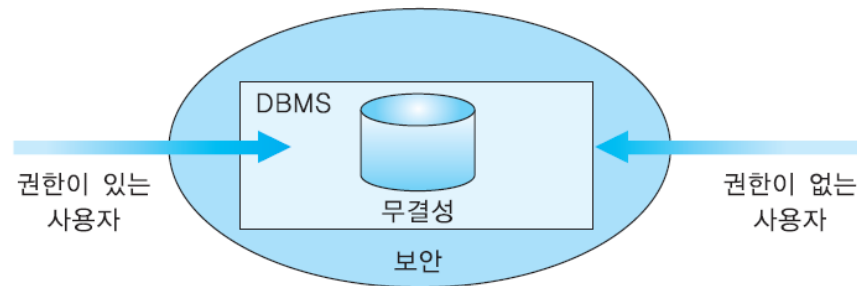
✓ 권한 보호

- 권한을 가진 사용자만 특정한 접근 모드로 데이터베이스를 접근할 수 있도록 보호
 - 예: 사용자마다 계정과 비밀번호 부여하고, 특정 부분은 특정 사용자에게만 허용

✓ 운영 보호

- 데이터베이스의 무결성에 대한 사용자 실수의 영향을 최소화하거나 제거하는 조치
 - 예 : 무결성 제약조건은 데이터베이스 관리자 또는 릴레이션을 생성한 사용자가 릴레이션 스키마에 명시

10.1 데이터베이스 보안(계속)



[그림 10.1] 무결성과 보안

- 무결성 : 권한이 있는 사용자로부터 데이터베이스를 보호하는 것
- 보안 : 권한이 없는 사용자로부터 데이터베이스를 보호하는 것
- 보안계획 :
 - ① 어떤 사용자가 어떤 데이터를 볼 수 있는가?
 - ② 데이터베이스에서 어떤 작업을 수행할 수 있는가를 정의
 - ③ 보안을 통해 제어해야 할 데이터베이스 내의 모든 항목을 열거하고, 조직체 내의 개인과 그룹을 열거하고, 두 리스트를 참조하여 데이터베이스 내의 어떤 사용자가 어떤 데이터 집합에 대하여 어떤 작업을 수행할 수 있는가를 지정한다.

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, 튜플, 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 취소 당했는가를 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 실행한 모든 연산들을 기록할 수 있음 (세션:사용자와 컴퓨터간 대화를 위한 논리적 연결)
- ✓ 권한이 없는 사용자가 데이터베이스를 갱신했다는 의심이 들면 **데이터베이스 감사**를 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 ] ;
```

- public : 모든 사용자에게 권한 허가
- WITH GRANT OPTION : 다른 사용자에게 권한허가 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)

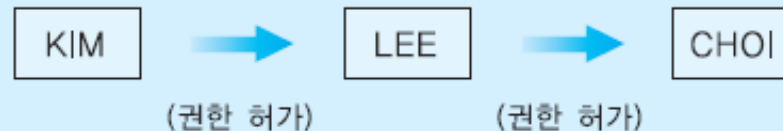
예4 : WITH GRANT OPTION과 함께 권한 허가

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 사용자에게 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;
```

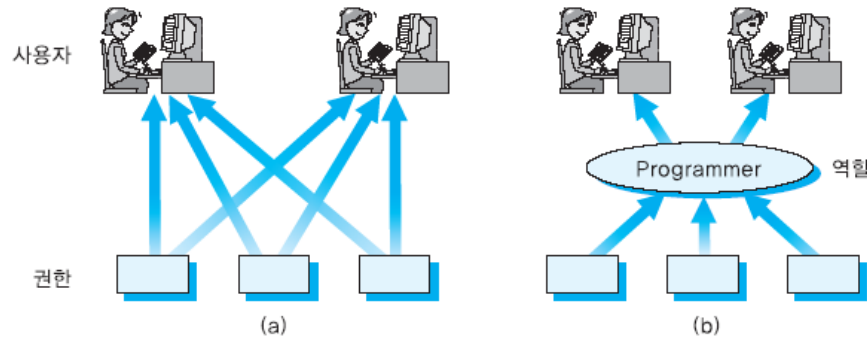


10.2 권한 관리(계속)

□ 역할(role)

- ✓ 여러 사용자에게 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 허가 받을 수 있음
- ✓ 동일한 권한들의 집합을 여러 사용자에게 허가하는 대신에 이 권한들을 역할에게 허가하고, 역할을 각 사용자에게 허가함
- ✓ 어떤 역할과 연관된 권한들에 변화가 생기면 그 역할을 허가 받은 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

□ 역할(role)

- ✓ 데이터베이스 내의 권한들의 관리를 쉽게하기 위해 사용
- ✓ [그림 10.2](a)는 두 명의 사용자가 동일한 역할(예: 프로그래머)을 수행하는 사원으로서 세가지 권한을 각 사용자에게 별도로 허가
- ✓ [그림 10.2](b)는 프로그래머 역할에게 권한을 추가하기만하면 두 사용자에게 자동적으로 그 권한이 허가
- 역할에 부여하거나 취소한 권한은 역할의 모든 구성원에게 적용된다.
예: 사원들이 업무에 참여시 역할을 부여하고 종료시 역할 제거가 자동으로 적용

10.2 권한 관리(계속)

❑ 역할을 생성하는 방법

- ✓ DBMS마다 차이가 있다.
- ✓ 데이터베이스 관리자가 역할을 생성
- ✓ 데이터베이스 관리자가 GRANT문을 사용하여 역할에게 권한들을 할당하고 사용자에게 역할을 할당한다.

❑ 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 권한(system privilege)은 사용자가 데이터베이스에서 특정 작업을 수행할 수 있도록 함(데이터베이스 객체를 생성, 수정, 삭제 권한)
예, 테이블을 생성하기 위해서는 CREATE TABLE 시스템 권한이 필요)
- ✓ 객체 권한(object privilege)은 사용자가 특정 객체(테이블, 뷰, 프로시저 등)에 대해 특정 연산을 수행할 수 있도록 함(객체 내용을 조작(추가, 변경, 삭제, 검색)할 수 있는 권한)

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

- create table : 자신의 스키마에 테이블 생성 권한
- create any table : 임의의 스키마에 테이블을 생성하는 권한
- any 키워드 : 사용자가 임의의 스키마에서 수행할 수 있는 권한을 가졌음을 의미
- create table 권한은 create index와 analyze 권한과 drop table 권한도 포함

10.3 오라클의 보안 및 권한 관리(계속)

- 시스템 특권

데이터베이스를 관리하는데 필요한 시스템 명령어를 사용하기 위해서는 시스템 특권을 부여 받아야 한다. 시스템 특권은 기본적으로 SYS 사용자가 소유하고 있으며, 다른 사용자에게 부여 할 수 있다.

- CREATE TABLE과 CREATE ANY TABLE 차이점

1) CREATE TABLE : 자신의 스키마에 테이블을 생성하는 권한

2) CREATE ANY TABLE : 임의의 스키마에 테이블을 생성하는 권한

10.3 오라클의 보안 및 권한 관리(계속)

시스템 특권	설명
ALTER SYSTEM	ALTER SYSTEM 문을 실행할 수 있는 권한이다.
CREATE SESSION	데이터베이스에 세션을 생성할 수 있는 권한이다. 즉, 로그인이 가능하다는 것을 의미한다.
CREATE USER	사용자를 생성하는 권한이다.
ALTER USER	사용자의 정보를 변경하는 권한이다.
DROP USER	사용자를 제거하는 권한이다.
CREATE TABLESPACE	테이블 스페이스를 생성하는 권한이다.
ALTER TABLESPACE	테이블 스페이스를 변경하는 권한이다.
DROP TABLESPACE	테이블 스페이스를 제거하는 권한이다.

10.3 오라클의 보안 및 권한 관리(계속)

SELECT ANY DICTIONARY	DICTIONARY를 조회할 수 있는 권한이다. 이 권한을 할당 받으면 SYS, SYSCAT, SYSGIS 소유의 객체들을 조회할 수 있다.
CREATE TABLE	자신의 스키마에 테이블을 생성하는 권한이다.
CREATE ANY TABLE	임의의 스키마에 테이블을 생성하는 권한이다.
ALTER ANY TABLE	임의의 스키마에 속한 테이블을 변경하는 권한이다.
DROP ANY TABLE	임의의 스키마에 속한 테이블을 제거하는 권한이다.
COMMENT ANY TABLE	임의의 스키마에 속한 테이블에 주석을 추가하는 권한이다.
SELECT ANY TABLE	임의의 스키마에 속한 테이블을 조회하는 권한이다.
INSERT ANY TABLE	임의의 스키마에 속한 테이블에 로우를 삽입하는 권한이다.
UPDATE ANY TABLE	임의의 스키마에 속한 테이블에 로우를 갱신하는 권한이다.
DELETE ANY TABLE	임의의 스키마에 속한 테이블에 로우를 제거하는 권한이다.
TRUNCATE ANY TABLE	임의의 스키마에 속한 테이블에 TRUNCATE를 수행 할 수 있다. 이 권한을 사용하기 위해서는 USE_TRUNCATE_PRIVILEGE
CREATE ANY INDEX	임의의 스키마에 속한 테이블에 인덱스를 생성하는 권한이다.

10.3 오라클의 보안 및 권한 관리(계속)

ALTER ANY INDEX	임의의 스키마에 속한 테이블에 인덱스를 수정하는 권한이다.
DROP ANY INDEX	임의의 스키마에 속한 테이블에 인덱스를 제거하는 권한이다.
CREATE SYNONYM	자신의 스키마에 동의어를 생성하는 권한이다.
CREATE ANY SYNONYM	임의의 스키마에 동의어를 생성하는 권한이다.
DROP ANY SYNONYM	임의의 스키마에 속한 동의어를 제거하는 권한이다.
SYSDBA	SHUTDOWN, ALTER DATABASE, CREATE DATABASE, ARCHIVELOG, RECOVERY 문을 실행할 수 있는 권한이다.
CREATE PUBLIC SYNONYM	PUBLIC 스키마에 동의어를 생성하는 권한이다.
DROP PUBLIC SYNONYM	PUBLIC 스키마에 속한 동의어를 제거하는 권한이다.
CREATE VIEW	자신의 스키마에 뷰를 생성하는 권한이다.
CREATE ANY VIEW	임의의 스키마에 뷰를 생성하는 권한이다.
DROP ANY VIEW	임의의 스키마에 속한 뷰를 제거하는 권한이다.
CREATE SEQUENCE	자신의 스키마에 시퀀스를 생성하는 권한이다.
CREATE ANY SEQUENCE	임의의 스키마에 시퀀스를 생성하는 권한이다.
ALTER ANY SEQUENCE	임의의 스키마에 속한 시퀀스를 변경하는 권한이다.
DROP ANY SEQUENCE	임의의 스키마에 속한 시퀀스를 제거하는 권한이다.
SELECT ANY SEQUENCE	임의의 스키마에 속한 시퀀스를 조회하는 권한이다.
CREATE ROLE	역할을 생성하는 권한이다.
DROP ANY ROLE	역할을 제거하는 권한이다.
GRANT ANY ROLE	임의의 역할에 부여하는 권한이다.
ALTER ANY ROLE	역할을 수정하는 권한이다.
ALTER DATABASE	데이터베이스를 변경하는 권한이다.

10.3 오라클의 보안 및 권한 관리(계속)

CREATE PROCEDURE	자신의 스키마에 프로시저를 생성하는 권한이다.
CREATE ANY PROCEDURE	임의의 스키마에 프로시저를 생성하는 권한이다.
ALTER ANY PROCEDURE	임의의 스키마에 속한 프로시저를 변경하는 권한이다.
DROP ANY PROCEDURE	임의의 스키마에 속한 프로시저를 제거하는 권한이다.
EXECUTE ANY PROCEDURE	임의의 스키마에 속한 프로시저를 실행하는 권한이다.
CREATE TRIGGER	자신의 스키마에 속한 트리거를 생성하는 권한이다.
CREATE ANY TRIGGER	임의의 스키마에 속한 트리거를 생성하는 권한이다.
ALTER ANY TRIGGER	임의의 스키마에 속한 트리거를 변경하는 권한이다.
DROP ANY TRIGGER	임의의 스키마에 속한 트리거를 제거하는 권한이다.
GRANT ANY OBJECT PRIVILEGE	모든 스키마 객체에 대한 특권을 가지는 권한이다.
GRANT ANY PRIVILEGE	모든 특권을 전부 부여할 수 있는 권한이다.

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;

;데이터베이스에 세션을 생성할 수 있는 권한=로그인이 가능

- ✓ 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

- ✓ 객체 권한은 특정 객체(테이블, 뷰, 프로시저 등)에 대해 특정연산을 수행 할 수 있도록 한다.
- ✓ 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	○	○
DELETE	○	○
EXECUTE		
INDEX	○	○
INSERT	○	○
REFERENCES	○	
SELECT	○	○
UPDATE	○	○

10.3 오라클의 보안 및 권한 관리(계속)

□ 미리 정의된 역할

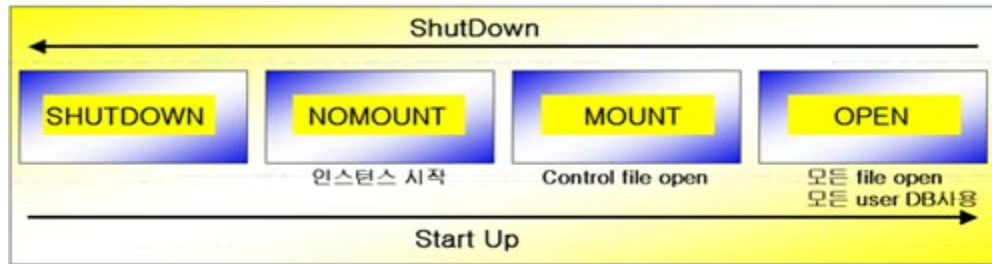
- ✓ 사용 패턴을 여러 관점에서 분석하여 각 사용 패턴에 맞게 미리 정해놓은 역할이 약 20여개 있음-이 역할을 사용하여 권한관리를 편리하게 할 수 있다.
- ✓ **connect** 역할만 있으면 자신의 테이블과 인덱스 등을 생성하지 못한다. 오라클 데이터베이스에 로그인하고, 만일 다른 사용자의 데이터를 검색할 수 있도록 권한을 허가 받았으면 이를 검색하고, 만일 다른 사용자의 데이터를 갱신할 수 있도록 권한을 허가 받았으면 이를 갱신할 수 있음
- ✓ **connect** 역할과 함께 **resource** 역할이 있으면 테이블과 인덱스를 생성하고, 자신의 객체에 대해 다른 사용자에게 권한을 허가하거나 취소할 수 있음

〈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)

❑ 데이터베이스의 시작단계



- NOMOUNT : 오라클 인스턴스만 시작된 단계로 보통 오라클 DB 생성에 사용

SQL> **STARTUP NOMOUNT** pfile=C:\Oracle\database2\initora.ora

- MOUNT : SGA를 메모리에 올리는 단계, 오라클 복구 수행은 MOUNT단계에서 수행

SQL> **STARTUP MOUNT;**

- OPEN : 모든 데이터베이스 파일이 열려 오라클 데이터베이스서버를 사용 가능

SQL> **ALTER DATABASE OPEN;**

- ALTER DATABASE : STARTUP MOUNT 단계에서 MOUNT 단계로, 또는 MOUNT 단계에서 OPEN 단계로 데이터베이스를 열려면 ALTER DATABASE 명령을 사용

SQL> **ALTER DATABASE** db_name **OPEN** [READ WRITE|READ ONLY]

10.3 오라클의 보안 및 권한 관리(계속)

❑ 데이터베이스 관리자 권한

- ✓ 데이터베이스 관리자만 관리자 권한을 가진 채 데이터베이스에 접속할 수 있어야 함
- ✓ SYSDBA로서 데이터베이스에 연결하면 데이터베이스와 데이터베이스 내의 모든 객체들에 대해 임의의 연산을 수행할 수 있다.

〈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

-- KIM, LEE 사용자 생성하기

```
CREATE USER KIM IDENTIFIED BY bluesky  
  DEFAULT TABLESPACE users  
  TEMPORARY TABLESPACE temp;
```

(1)

```
CREATE USER LEE IDENTIFIED BY redsun  
  DEFAULT TABLESPACE users  
  TEMPORARY TABLESPACE temp;
```

-- Grant 명령으로 접속, 사용 권한 주기

```
grant connect, resource, create session, create view to KIM;  
grant connect, resource, create session, create view to LEE;
```

(2)

[예제 3.1] 사용자를 생성하고 권한을 허가

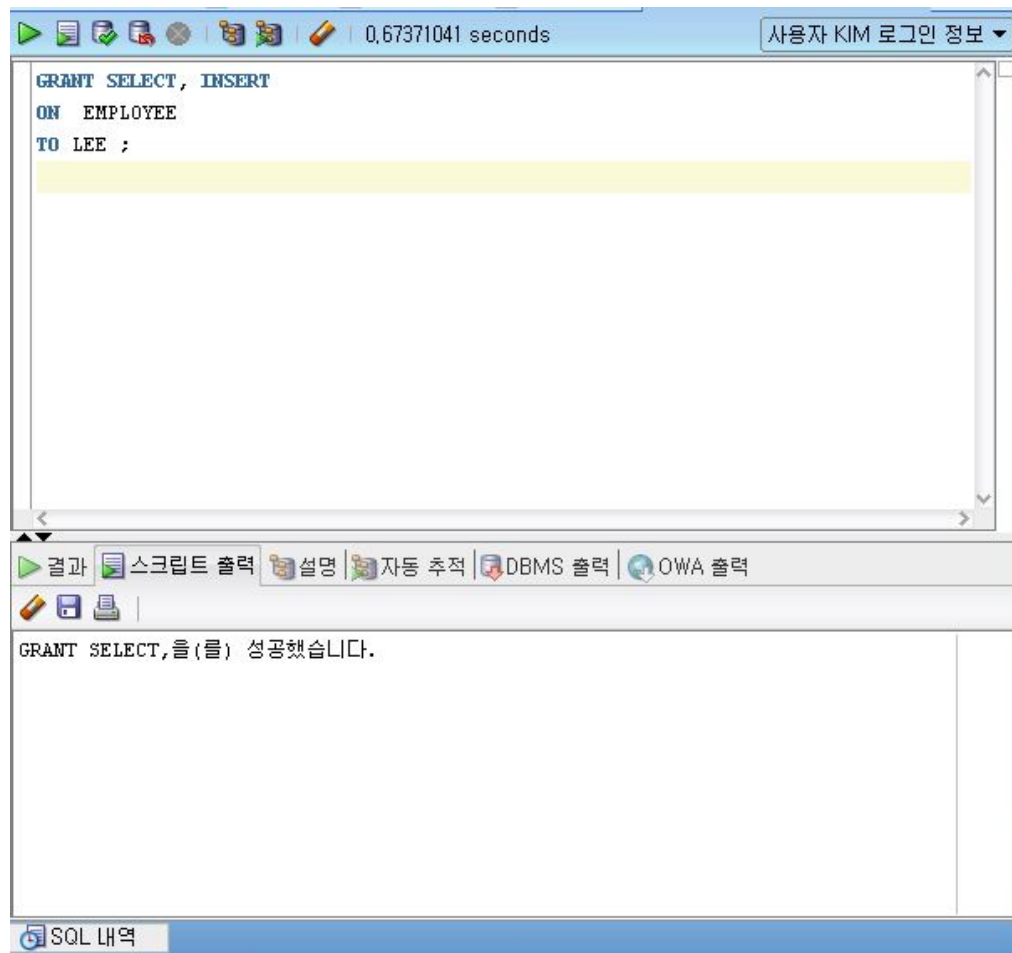
10.3 오라클의 보안 및 권한 관리(계속)

❑ 사용자 LEE에게 SELECT와 INSERT 권한 허가

- ✓ 예제 3.2에서 KIM이 DEPARTMENT, EMPLOYEE 테이블을 생성하고, 이 두 테이블에 튜플들을 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)



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;
```

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

```
REVOKE 을 (를) 성공했습니다.
```

- LEE에게 허가 한 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;
```

❑ 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

- ✓ 사용자 KIM으로 로그인 한 상태에서 LEE에게 EMPLOYEE 릴레이션에서 SELECT와 INSERT 권한을 허가

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

- ✓ USER_TAB_PRIVS 데이터 사전 뷰 또는 DBA_TAB_PRIVS 데이터 사전 뷰를 통해서 검색할 수 있음 (사용자가 보유한 권한에 관련된 정보)

```
SELECT *
FROM   USER_TAB_PRIVS;
```

10.3 오라클의 보안 및 권한 관리(계속)

```
SELECT *  
FROM USER_TAB_PRIVS ;
```

GRANTEE	OWNER	TABLE_NAME
LEE	KIM	EMPLOYEE
LEE	KIM	EMPLOYEE

2 rows selected

GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY
KIM	INSERT	NO	NO
KIM	SELECT	NO	NO

- 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
- HIERARCHY 계층적 관계 여부

10.3 오라클의 보안 및 권한 관리(계속)

❑ 사용자 LEE가 허가 받은 시스템 권한 확인

- ✓ 사용자 LEE로 로그인한 후에 아래의 질의를 수행

```
SELECT *  
FROM   USER_SYS_PRIVS;
```

- ✓ **USER_SYS_PRIVS**는 사용자에게 허가된 시스템 권한을 보여주는 데이터 사전 뷰

10.3 오라클의 보안 및 권한 관리(계속)

```
SELECT *  
FROM USER_SYS_PRIVS ;
```

USERNAME	PRIVILEGE	ADMIN_OPTION
KIM	CREATE VIEW	NO
KIM	UNLIMITED TABLESPACE	NO
KIM	CREATE SESSION	NO

3 rows selected

✓KIM은 CREATE VIEW 권한, UNLIMITED TABLESPACE 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)

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

```
SELECT *  
FROM USER_TAB_PRIVS;
```

```
SELECT *  
FROM USER_TAB_PRIVS ;
```

GRANTEE	OWNER	TABLE_NAME
LEE	KIM	EMPLOYEE
LEE	KIM	EMPLOYEE

2 rows selected

GRANTOR	PRIVILEGE	GRANTABLE	HIERARCHY
KIM	INSERT	NO	NO
KIM	SELECT	NO	NO

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 튜플을 삽입

```
INSERT INTO KIM.EMPLOYEE  
VALUES (8888, '김팔팔', '사원', 2106, 1500000, 2);
```

```
INSERT INTO KIM.EMPLOYEE  
VALUES (8888, '김팔팔', '사원', 2106, 1500000, 2) ;
```

1 행 삽입됨

10.3 오라클의 보안 및 권한 관리(계속)

❑ 동의어(synonym) 정의

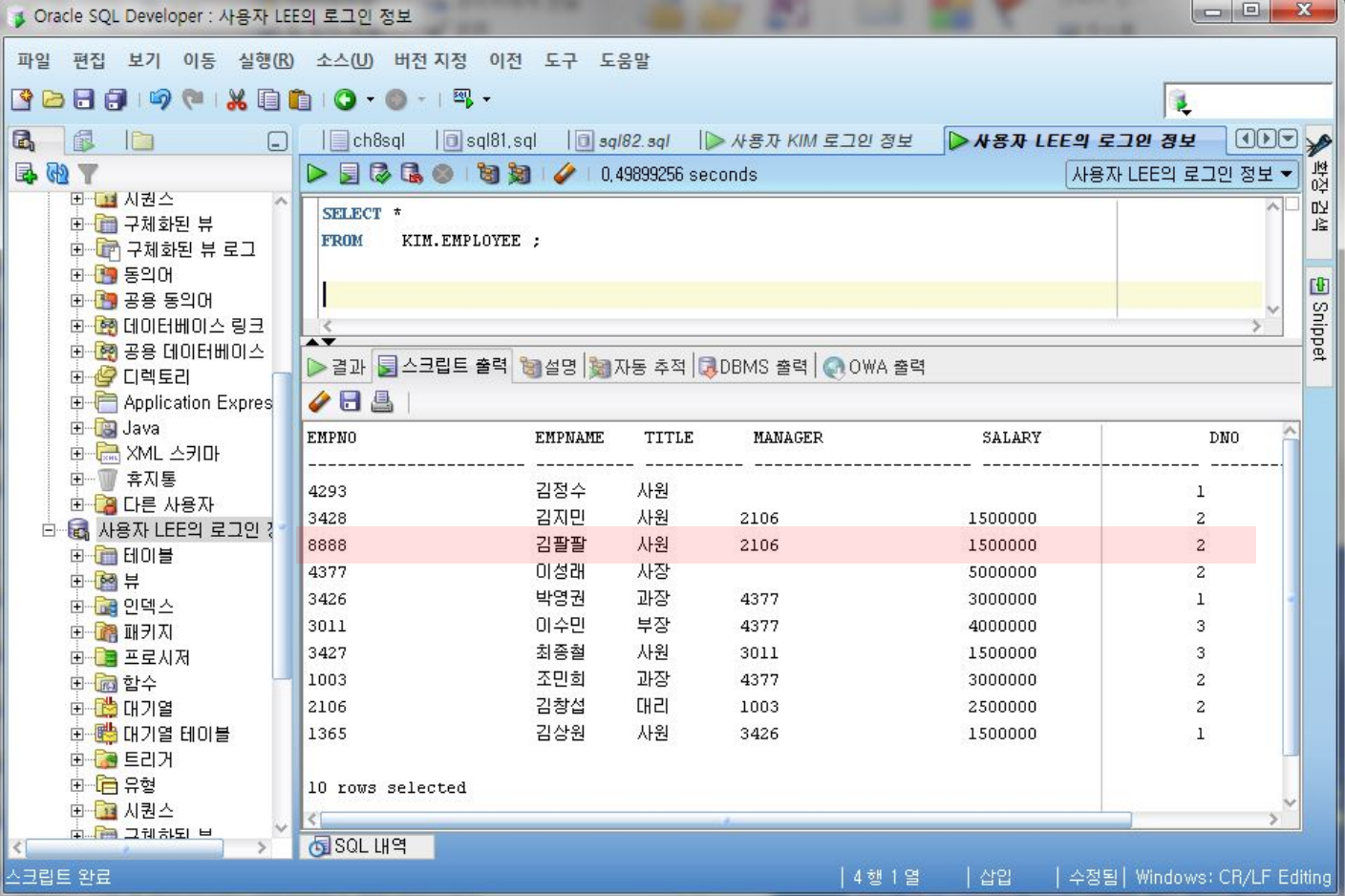
- ✓ 사용자 LEE가 사용자 KIM의 EMPLOYEE 테이블을 자주 접근한다면 매번 EMPLOYEE 테이블 앞에 KIM을 붙이는 것이 번거로움
- ✓ 동의어를 만드는 구문

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)



The screenshot shows the Oracle SQL Developer interface. The title bar indicates the current session is for user 'LEE'. The main window displays a query: `SELECT * FROM KIM.EMPLOYEE ;`. The results are shown in a table with 10 rows. The table has columns: EMPNO, EMPNAME, TITLE, MANAGER, SALARY, and DNO. The row with EMPNO 8888 is highlighted in red. The status bar at the bottom shows '10 rows selected' and 'SQL 내역'.

EMPNO	EMPNAME	TITLE	MANAGER	SALARY	DNO
4293	김정수	사원			1
3428	김지민	사원	2106	1500000	2
8888	김팔팔	사원	2106	1500000	2
4377	이성래	사장		5000000	2
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
3427	최종철	사원	3011	1500000	3
1003	조민희	과장	4377	3000000	2
2106	김창섭	대리	1003	2500000	2
1365	김상원	사원	3426	1500000	1

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

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

명령의 1 행에서 시작하는 중 오류 발생:

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

오류 발생 명령행: 1, 열: 11

오류 보고:

SQL 오류: ORA-01031: 권한이 불충분합니다

10.3 오라클의 보안 및 권한 관리(계속)

❑ UPDATE 권한을 추가로 허가

- ✓ 사용자 KIM으로 로그인한 후에 아래의 GRANT문을 수행

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



- ✓ 사용자 LEE로 로그인한 후에 UPDATE문을 다시 수행하면 성공적으로 튜플을 수정할 수 있음

10.3 오라클의 보안 및 권한 관리(계속)

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

결과 | 스크립트 출력 | 설명 | 자동 추적 | DBMS 출력 | OWA 출력

1 행 갱신됨

SELECT *
FROM KIM.EMPLOYEE ;

4293	김정수	사원			1
3428	김지민	사원	2106	1500000	2
8888	김팔팔	사원	2106	1500000	2
4377	이성래	사장		5000000	2
3426	박영권	과장	4377	3000000	1
3011	이수민	부장	4377	4000000	3
3427	최종철	사원	3011	1500000	3
1003	조민희	과장	4377	3000000	3
2106	김창섭	대리	1003	2500000	2
1365	김상원	사원	3426	1500000	1

10 rows selected

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
-VIEW를 통해서 가능
 - ✓ 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
 - ✓ DELETE의 경우 튜플의 구성 컬럼 전체에 대한 작업이므로 컬럼단위의 권한부여 불가능