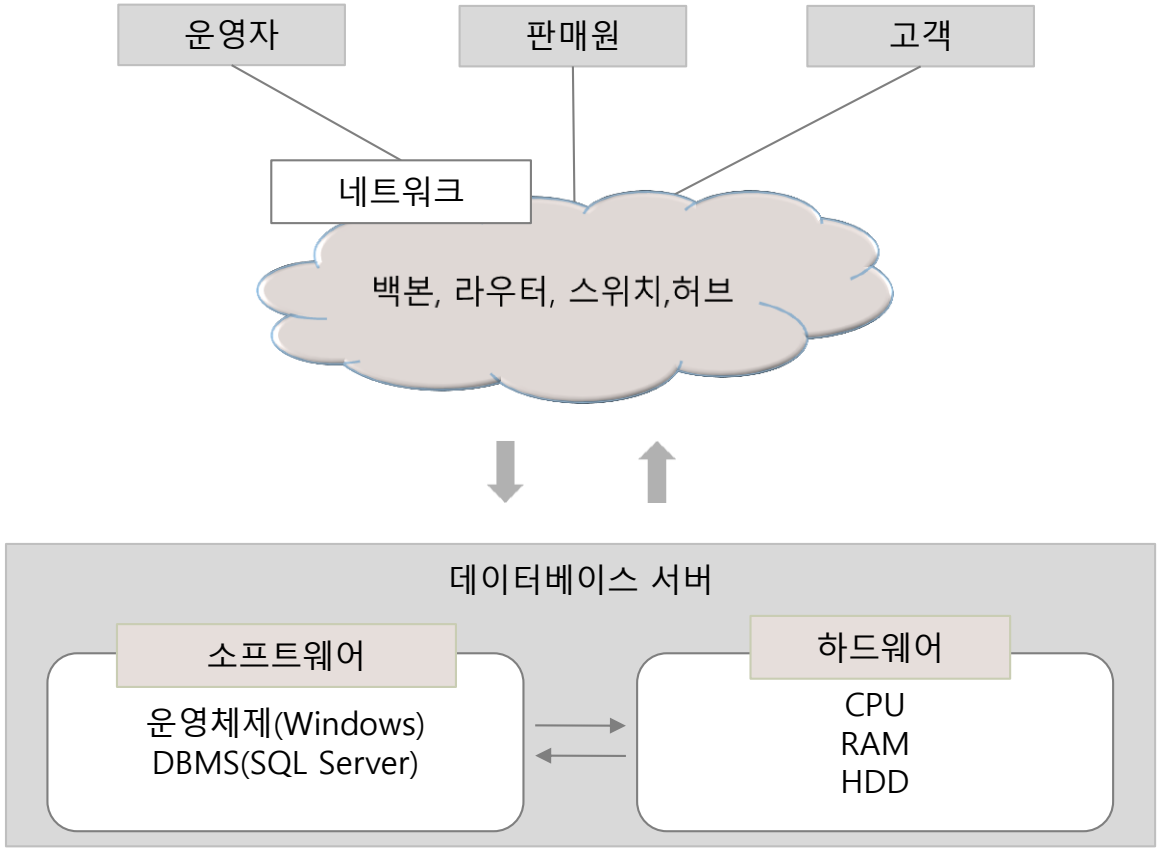
5-3 사용자관리

1. **데이터베이스 관리**
2. **보안과 권한**
   - User와 Schema
   - Profile 관리
   - Privilege(권한) 관리
   - Role관리

데이터베이스 관리

- 데이터베이스 관리의 중요성
- 데이터베이스 관리 업무

1. 데이터베이스 관리의 중요성



데이터베이스 운영 환경

2. 데이터베이스 관리 업무

- 서비스 관리
- 점검 및 모니터링
- 장애 대처
- 백업과 복원
- 사용자 관리 및 권한 관리
- 시스템 데이터베이스 관리
- 사용자 데이터베이스 관리
- 데이터베이스 저장 공간 관리
- 인덱스 관리

02. 보안과 권한

- 로그인 사용자 관리
- 권한 관리

02. 보안과 권한

- DBMS는 ① 로그인 단계에서 DBMS 접근을 제한하는 로그인 사용자 관리 ② 로그인한 사용자별로 특정 데이터로의 접근을 제한하는 권한 관리의 기능 제공

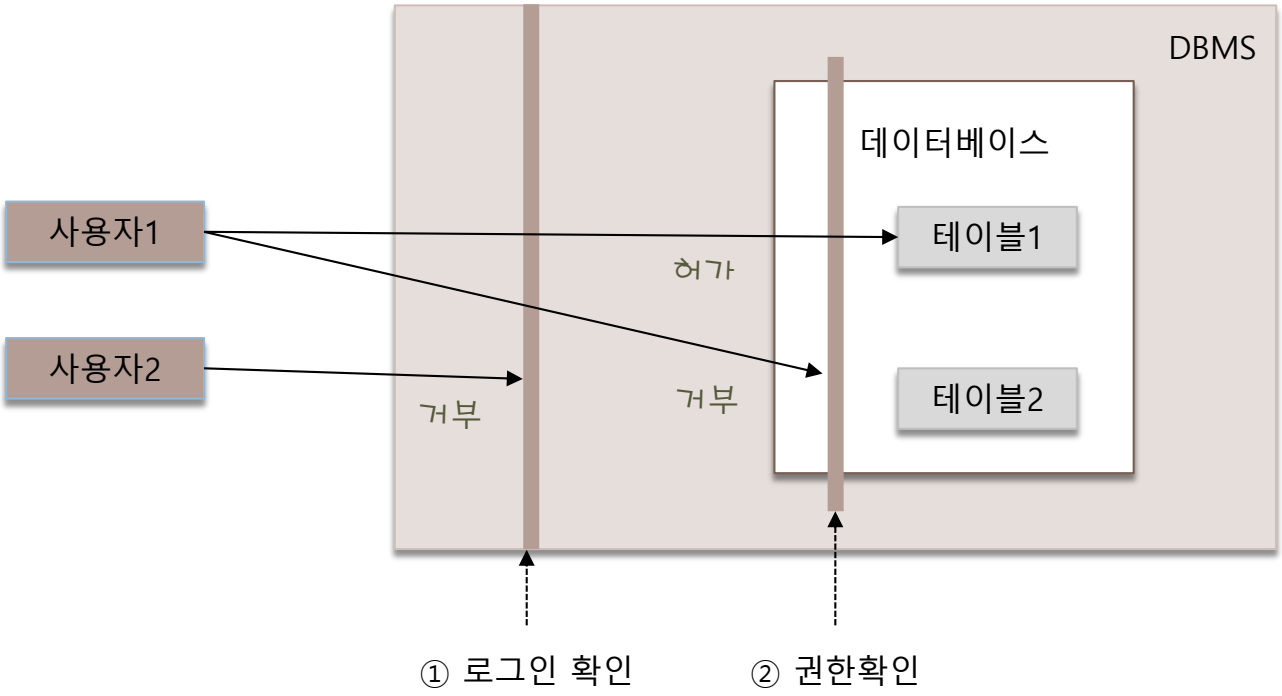


그림 데이터베이스 접근 권한

1. 테이블스페이스와 로그인 사용자 관리

테이블스페이스 생성하기

- 테이블스페이스 : 오라클에서 데이터를 저장할 때 사용하는 논리적 저장 공간(하드디스크에서는 실제 여러 개의 물리적인 데이터 파일로 구성될 수 있음). 오라클 시스템 운영에 필요한 필수 정보를 담고 있음
- CREATE TABLESPACE 테이블스페이스명
  DATAFILE '저장될 경로 및 사용할 파일명' SIZE 저장공간

질의 1 (system 계정) 10M의 용량의 테이블스페이스 md_tbs, mb_test를 C:₩madang₩oradata 폴더에 생성하시오. 이때 데이터 파일 이름은 각각 md_tbs_data01.dbf, md_test_data01.dbf로 한다(폴더가 없으면 생성 후 진행).

```
CREATE TABLESPACE md_tbs
DATAFILE 'C:₩madang₩oradata₩md_tbs_data01.dbf' SIZE 10M;
```

```
tablespace MD_TBS이(가) 생성되었습니다.
```

```
CREATE TABLESPACE md_test
DATAFILE 'C:₩madang₩oradata₩md_test_data01.dbf' SIZE 10M;
```

```
tablespace MD_TEST이(가) 생성되었습니다.
```

1. 테이블스페이스와 로그인 사용자 관리

□ 테이블스페이스 삭제하기

DROP TABLESPACE 테이블스페이스이름
[INCLUDING CONTENTS [AND DATAFILES] [CASCADE CONSTRAINTS]];

질의 2 (system 계정) md_test 테이블스페이스를 데이터 파일까지 포함하여 모두 삭제하라.

DROP TABLESPACE md_test INCLUDING CONTENTS AND DATAFILES;

```
tablespace MD_TEST이(가) 삭제되었습니다.
```

1.2 신규 로그인 사용자 계정 생성하기

사용자 계정 생성하기

```
CREATE USER [사용자이름]
        IDENTIFIED BY [비밀번호]
        DEFAULT TABLESPACE [테이블스페이스];
```

사용자 계정 설정 변경하기

```
ALTER USER [사용자이름]
      IDENTIFIED BY [비밀번호];
```

사용자 계정 삭제하기

```
DROP USER [사용자이름] CASCADE;
```

1.2 신규 로그인 사용자 계정 생성하기

질의 3 (system 계정) 새로운 사용자 mdguest를 생성하시오. 비밀번호는 mdguest, 테이블스페이스는 users로 설정한다.

CREATE USER mdguest IDENTIFIED BY mdguest;

```
user MDGUEST이(가) 생성되었습니다.
```

질의 4 (system 계정) 새로운 사용자 mdguest2를 생성하시오. 비밀번호는 mdguest2, 테이블스페이스는 앞에서 생성한 md_tbs로 설정한다.

CREATE USER mdguest2 IDENTIFIED BY mdguest2 DEFAULT TABLESPACE md_tbs;

```
user MDGUEST2이(가) 생성되었습니다.
```

2. 권한 관리

- 소유한 개체에 대한 사용 권한을 관리하기 위한 명령을 DCL(Data Control Language)이라고 함
- 대표적 DCL 문 : 권한 허가 GRANT 문, 권한 취소 REVOKE 문

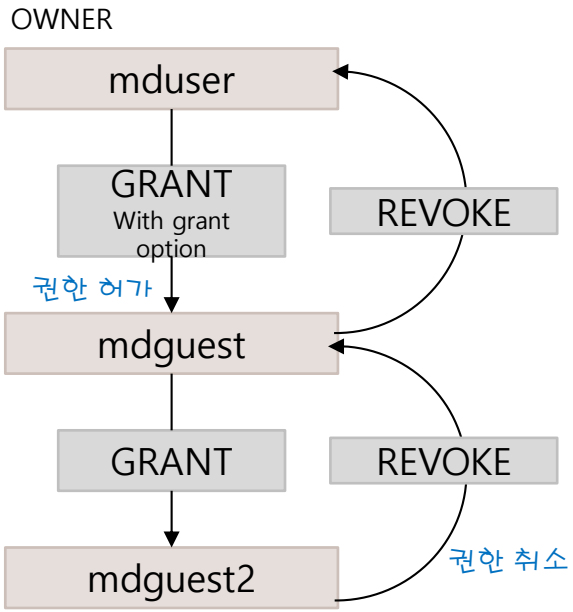


그림 6 GRANT 문과 REVOKE 문의 관계

2.1 권한 허가 - GRANT

- **객체를 생성한 소유자가 대상 객체에 대한 권한을 다른 사용자에게 허가하는 명령**

```
GRANT 권한 [(컬럼[ ,...n ])] [ ,...n ]
        [ON 객체] TO {사용자I롤IPUBLIC [ ,...n ]}
        [WITH GRANT OPTION]
```

* [, ... n] : 반복가능을 의미

질의 6　(pgm 계정) mdguest에게 Book 테이블의 SELECT 권한을 부여하시오.

```
GRANT SELECT ON student TO mdguest;
```

```
GRANT을(를) 성공했습니다.
```

질의 7　(pgm 계정) mdguest에게 Customer 테이블의 SELECT, UPDATE 권한을 WITH GRANT OPTION과 함께 부여하시오.

```
GRANT SELECT, UPDATE ON Customer TO mdguest WITH GRANT OPTION;
```

```
GRANT을(를) 성공했습니다.
```

2.1 권한 허가 - GRANT

질의 8　(mdguest 계정) pgm.Student 테이블과 pgm.professor 테이블의 SELECT 권한을 mdguest2에 부여하시오.

```
GRANT SELECT ON student TO mdguest2;
```

```
GRANT SELECT ON professor TO mdguest2;
```

```
GRANT을(를) 성공했습니다.
```

질의 9　(madang 계정) emp 테이블을 모든 사용자가 SELECT할 수 있도록 권한을 부여하시오.

```
GRANT SELECT ON emp TO PUBLIC;
```

```
GRANT을(를) 성공했습니다.
```

2.2 권한 취소 - REVOKE

- **GRANT 문으로 허가한 권한을 취소, 회수하는 명령**

```
REVOKE 권한 [(컬럼[ ,...n ])] [ ,...n ]
    [ON 객체] FROM { 사용자I롤IPUBLIC [ ,...n ]}
    [CASCADE CONSTRAINTS]
```

- **GRANT 문이 권한 부여를 위해 'TO 사용자'를 표기하였다면, REVOKE 문은 권한 취소를 위해 'FROM 사용자'를 표기함**

- **권한을 재부여하는 WITH GRANT OPTION의 회수를 위해 'CASCADE' 옵션을 사용함**

- **CASCADE는 사용자가 다른 사용자에게 부여한 권한까지 연쇄적으로 취소하라는 의미로, 사전에 주의 깊게 확인하고 사용해야 함**

2.2 권한 취소 - REVOKE

질의 10　(madang 계정) mdguest에게 부여된 student 테이블의 SELECT 권한을 취소하시오.

REVOKE SELECT ON Book FROM mdguest;

```
REVOKE을(를) 성공했습니다.
```

질의 11　(madang 계정) mdguest에게 부여된 Customer 테이블의 SELECT 권한을 취소하시오.

REVOKE SELECT ON Customer FROM mdguest;

```
REVOKE을(를) 성공했습니다.
```

2.3 역할 - ROLE

- 롤(ROLE): 데이터베이스 객체에 대한 권한을 모아둔 집합

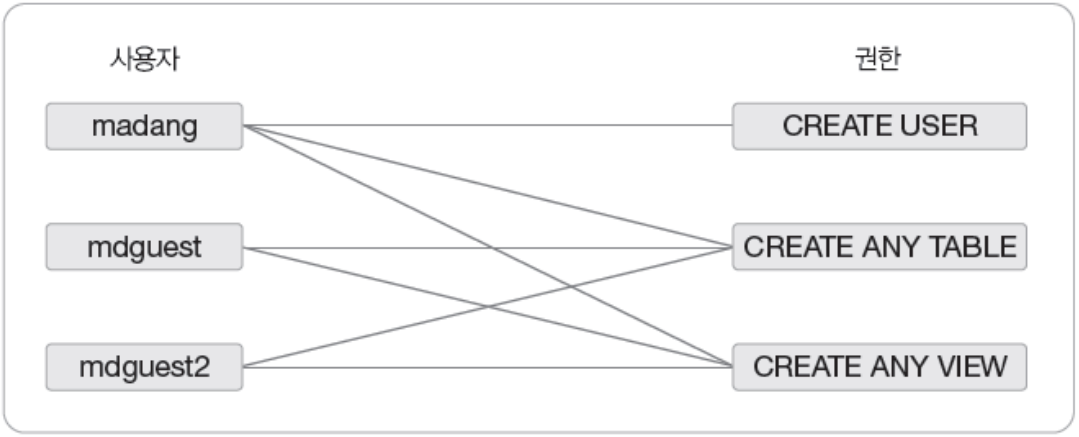


그림 9-12 마당서점 사용자별 시스템 권한

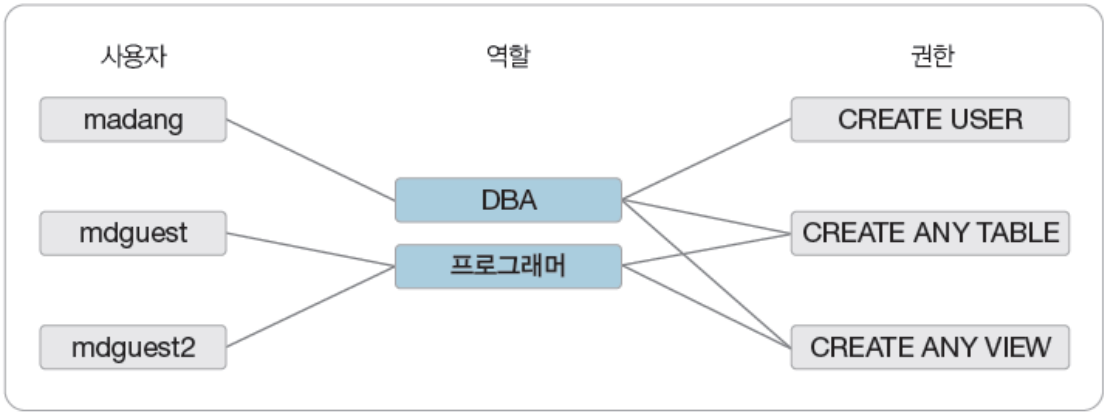


그림 9-12 마당서점 사용자별 역할과 권한

2.3 역할 - ROLE

- **역할 생성**

```
CREATE ROLE 역할 이름
```

- **역할 제거**

```
DROP ROLE  역할이름
```

- **역할에 권한 부여**

```
GRANT 권한 [ON 객체] TO 역할이름
```

- **역할에 권한 회수**

```
REVOKE 권한 [ON 객체] FROM 역할이름;
```

- **사용자에게 역할 부여**

```
GRANT 역할이름 TO 사용자
```

2.3 역할 - ROLE

역할 생성부터 사용자 추가까지의 단계

1. CREATE ROLE - 역할 생성
2. GRANT - 만들어진 역할에 권한 부여
3. GRANT - 사용자에게 역할 부여

역할을 제거하면 반대로 수행

1. DROP ROLE - 역할 삭제(사용자에게 부여된 역할에 대한 권한 역시 제거됨)

질의 12 (system 계정) 'programmer'라는 역할을 생성하시오.

CREATE ROLE programmer;

```
role PROGRAMMER이(가) 생성되었습니다.
```

2.3 역할 - ROLE

질의 13 (system 계정) programmer 역할에 CREATE ANY TABLE과 CREATE ANY VIEW 권한을 부여하시오.

GRANT CREATE ANY TABLE, CREATE ANY VIEW TO programmer;

```
GRANT을(를) 성공했습니다.
```

질의 14 (system 계정) mdguest에 programmer 역할의 권한을 부여하시오.

GRANT programmer TO mdguest;

```
GRANT을(를) 성공했습니다.
```

2.3 역할 - ROLE

질의 11　(mdguest 계정) mdguest2 사용자에게 다음의 테이블을 생성하고 데이터를 삽입(INSERT)하시오.

```
CREATE TABLE mdguest2.NEWTABLE (
    myname VARCHAR2(40),
    myphone VARCHAR2(20)
    );
```

```
table MDGUEST2.NEWTABLE이(가) 생성되었습니다.
```

```
INSERT INTO mdguest2.NEWTABLE (myname, myphone)
    VALUES ('홍길동', '000-000-0100');
```

```
SQL 오류: ORA-01031: 권한이 불충분합니다
01031. 00000 -  "insufficient privileges"
*Cause:    An attempt was made to change the current username or password
           without the appropriate privilege. This error also occurs if
           attempting to install a database without the necessary operating
           system privileges.
           When Trusted Oracle is configure in DBMS MAC, this error may occur
           if the user was granted the necessary privilege at a higher label
           than the current login.
*Action:   Ask the database administrator to perform the operation or grant
           the required privileges.
```

2.3 역할 - ROLE

질의 16 (system 계정) programmer 역할에 mdguest2.NEWTABLE 테이블에 대한 SELECT와 INSERT 권한을 부여하시오. 그리고 (mdguset 계정) INSERT 문을 수행한 후 조회(SELECT)해 보시오.

(system 계정)

```
GRANT SELECT, INSERT ON mdguest2.NEWTABLE TO programmer;
```

```
GRANT을(를) 성공했습니다.
```

(mdguest 계정)

```
INSERT INTO mdguest2.NEWTABLE (myname, myphone)
        VALUES ('홍길동', '000-000-0100');
COMMIT;
```

```
1개 행 이(가) 삽입되었습니다.
```

```
SELECT *
FROM mdguest2.NEWTABLE;
```

MYNAME	MYPHONE
홍길동	000-000-0100

2.3 역할 - ROLE

질의 17 (system 계정) mdguest2.NEWTABLE 계정의 SELECT 권한을 회수하시오. 그리고 (mdguest 계정) mdguest2.NEWTABLE 테이블을 조회(SELECT)해 보시오.

(system 계정)
REVOKE SELECT ON mdguest2.NEWTABLE FROM programmer;

```
REVOKE을(를) 성공했습니다.
```

(mdguest 계정)
SELECT *
FROM mdguest2.NEWTABLE;

```
ORA-01031: 권한이 불충분합니다
01031, 00000 - "insufficient privileges"
*Cause:    An attempt was made to change the current username or password
           without the appropriate privilege. This error also occurs if
           attempting to install a database without the necessary operating
           system privileges.
           When Trusted Oracle is configure in DBMS MAC, this error may occur
           if the user was granted the necessary privilege at a higher label
           than the current login.
```

2.3 역할 - ROLE

질의 18 (system 계정) programmer 역할을 제거하시오. mdguest2.NEWTABLE 역시 제거하시오

DROP ROLE programmer;

```
role PROGRAMMER이(가) 삭제되었습니다.
```

DROP TABLE mdguest2.NEWTABLE;

```
table MDGUEST2.NEWTABLE이(가) 삭제되었습니다.
```

User와 Schema(스키마)

- User : 오라클에 접속하기 위해 사용되는 사용자를 의미
- Schema : 특정 User 가 만든 Object 들의 모음을 의미
- 일반적으로 혼용해서 많이 사용함.

3. PROFILE 관리하기

PASSWORD PROFILE 관련 파라미터

1) FAILED_LOGIN_ATTEMPTS : 로그인 실패 할 경우 계정이 잠기는데 허용될 횟수를 지정

2) PASSWORD_LOCK_TIME : 위 1번의 상황에서 계정이 잠길 기간 설정

3) PASSWORD_LIFE_TIME : 암호를 변경 없이 사용할 수 있는 기간 설정

4) (PASSWORD_REUSE_TIME : 동일한 암호를 쓸 수 없는 기간 설정

5) SWORD_GRACE_TIME : 암호 변경 추가 시간 설정

6) PASSWORD_REUSE_MAX : 동일한 암호를 쓸 수 없는 횟수 설정

7) PASSWORD_VERIFY_FUNCTION : 암호를 복잡하게 만드는 함수 사용

- 암호는 최소한 4글자 이상 되어야 합니다.
- 암호는 사용자 계정과 달라야 합니다.
- 암호는 하나의 특수문자나, 알파벳 , 숫자가 포함되어야 합니다.
- 암호는 이전 암호와 3글자 이상 달라야 합니다

•예제 1. Password 관련 PROFILE 생성하기

- 조건 1: 로그인 시도 3회 실패 시 계정을 5일 동안 사용 못하게 할 것
- 조건 2: 계정의 암호는 15일에 한번씩 변경하게 할 것
- 조건 3: 동일한 암호는 15일 동안 사용 못하게 할 것

```
SYS>CREATE PROFILE sample_prof LIMIT
  2   FAILED_LOGIN_ATTEMPTS   3
  3   PASSWORD_LOCK_TIME   5
  4   PASSWORD_LIFE_TIME 15
  5   PASSWORD_REUSE_TIME 15 ;
```

2) RESOURCE PROFILE 관련 파라미터

- RESOURCE_LIMIT = true

- ALTER SYSTEM SET RESOURCE_LIMIT = true;

- **CPU_PER_SESSION :** 1 세션당 CPU 를 쓸 수 있는 시간 지정(1/100 초)
- **SESSIONS_PER_USER :** 1 유저당 동시 접속 가능한 세션 수 지정
- **CONNECT_TIME :** 접속 가능한 시간 지정 (분 단위)
- **IDLE_TIME :** 휴면 시간 지정 (분 단위)
- **LOGICAL_READS_PER_SESSION :** 1 세션에서 사용 가능한 Block 수 지정
- **PRIVATE_SGA :** MTS / Shared Server 의 경우 세션당 SGA 사용 가능량 지정
- **CPU_PER_CALL :** 1 세션당 사용 가능한 CPU 시간 지정
- **LOGICAL_READS_PER_CALL :** 1 call 당 사용 가능한 Block 수 지정

* 예제 2: RESOURCE 관련 PROFILE 만들기

```
SYS>ALTER  SYSTEM SET RESOURCE_LIMIT=true ;
```

- 조건 1: 1명당 연속적으로 CPU를 사용할 수 있는 시간을 10초로 제한할 것.
- 조건 2: 하루 중 8시간만 DB에 접속 가능하게 할 것.
- 조건 3: 10분 동안 사용하지 않으면 강제로 접속을 끊을 것

```
SYS>CREATE  PROFILE RE_SAMPLE_PROF LIMIT
2  CPU_PER_SESSIN 1000
3  CONNECT_TIME  480
4  IDLE_TIME 10 ;
```

3) 사용자에게 PROFILE 할당하기

(1) 현재 모든 사용자가 적용 받고 있는 PROFILE 확인하기

```
SYS>SELECT username "사용자명" , profile "적용 프로파일"
 2  FROM dba_users
 3  WHERE username='SCOTT' ;
```

(2) 해당 PROFILE 에 어떤 내용이 있는지 확인하기

```
SYS>SELECT * FROM dba_profiles
 2  WHERE PROFILE='SAMPLE_PROF' ;
```

```
SYS>  SELECT * FROM dba_profiles
 2  WHERE profile='RE_SAMPLE_PROF' ;
```

4) 사용자에게 PROFILE 적용시키고 확인하기

```
SYS>ALTER USER scott PROFILE sample_prof;

SYS>ALTER USER scott PROFILE re_sample_prof;
```

```
SYS>SELECT username, profile
  2  FROM dba_users
  3  WHERE username='SCOTT';

USERNAME                       PROFILE
------------------------------ ------------------------------
SCOTT                          RE_SAMPLE_PROF
```

(4) 사용 안 하는 PROFILE 삭제하기

```
SYS>DROP PROFILE re_sample_prof;
  drop PROFILE re_sample_prof
  *
ERROR at line 1:
ORA-02382: PROFILE RE_SAMPLE_PROF has users assigned, cannot drop
without CASCADE
```

```
SYS>DROP PROFILE re_sample_prof CASCADE;
```

```
SYS>SELECT username, PROFILE
  2  FROM dba_users
  3  WHERE username='SCOTT';
```

3. PRIVILEGE (권한) 관리하기

1) System 관련 주요 PRIVILEGE

대 분 류	PRIVILEGE	설 명
INDEX	CREATE ANY INDEX	소유자에 상관없이 모든 테이블에 인덱스를 생성할 수 있는 권한
	DROP ANY INDEX	소유자에 상관없이 모든 인덱스를 삭제할 수 있는 권한
	ALTER ANY INDEX	소유자에 상관없이 모든 인덱스를 수정할 수 있는 권한
TABLE	CREATE TABLE	자신 소유의 테이블을 생성할 수 있는 권한
	CREATE ANY TABLE	소유자에 상관없이 다른 user 이름으로 테이블을 생성할 수 있는 권한
	ALTER ANY TABLE	소유자에 상관없이 모든 테이블의 구조를 수정할 수 있는 권한
	DROP ANY TABLE	소유자에 상관없이 모든 사용자의 테이블을 삭제할 수 있는 권한
	UPDATE ANY TABLE	소유자에 상관없이 모든 사용자의 테이블을 업데이트 할 수 있는 권한
	DELETE ANY TABLE	소유자에 상관없이 모든 사용자의 테이블의 데이터를 삭제 할 수 있는 권한
	INSERT ANY TABLE	소유자에 상관없이 모든 사용자의 테이블에 데이터를 삽입 할 수 있는 권한

SESSION	CREATE SESSION	서버에 접속할 수 있는 권한
	ALTER SESSION	접속 상태에서 환경값을 변경할 수 있는 권한
	RESTRICTED SESSION	Restricted 모드로 open 된 DB에 접속할 수 있는 권한
TABLESPACE	CREATE TABLESPACE	Tablespace를 만들 수 있는 권한
	ALTER TABLESPACE	Tablespace를 수정 할 수 있는 권한
	DROP TABLESPACE	Tablespace를 삭제 할 수 있는 권한
	UNLIMITED TABLESPACE	Tablespace 사용용량을 무제한으로 허용 하는 권한. 즉 quota 옵션 적용을 받지 않게 됨

2) SYSOPER / SYSDBA PRIVILEGE

PRIVILEGE	할 수 있는 일
SYSOPER	Startup / shutdown
	Alter database mount / open
	Alter database backup control file to .....
	Recover database
	Alter database archivelog
	Restricted session
SYSDBA	SYSOPER PRIVILEGE with admin option
	Create database
	Alter tablespace ... begin backup / end backup
	Recover database until

3) SYSTEM 관련 권한 할당하기 / 해제하기

```
SYS> GRANT CREATE TABLE, CREATE SESSION TO SCOTT;
```

```
SYS> GRANT CREATE TABLE, CREATE SESSION TO SCOTT WITH ADMIN OPTION;
```

```
SYS> REVOKE CREATE TABLE FROM SCOTT ;
```

4) 사용자가 가지고 있는 권한 조회하기

```
SYS>SELECT * FROM DBA_SYS_PRIVS
  2  WHERE grantee='SCOTT';
```

- **Object 관련 PRIVILEGE – 주로 DML 과 관련된 권한들임**

6) Object 권한 할당하기 / 해제하기

-사용 예제 1:

SCOTT 사용자에게 HR 사용자가 만든 EMPLOYEES 테이블을 SELECT 할 수 있도록 권한을 할당하세요.

```
SYS> GRANT  SELECT  ON  HR.EMPLOYEES  TO  SCOTT ;
```

-사용 예제 2:

SCOTT 사용자에게 HR 가 만든 EMPLOYEES 테이블을 UPDATE 할 수 있도록 하세요.그리고 SCOTT 사용자가 이 권한을 다른 사람에게 줄 수 있는 권한도 주세요.

```
SYS> GRANT UPDATE ON HR.EMPLOYEES TO SCOTT WITH GRANT OPTION ;
```