사용자 접근 제어

1. 사용자 접근 제어

- 다중 사용자 환경에서는 데이터베이스 액세스와 사용의 보안 유지가 요구됨
  - ✓ 데이터베이스 액세스 제어
  - ✓ 데이터베이스에서 특정 객체에 대한 액세스 제공
  - ✓ 오라클 데이터 사전으로 주어지고 받는 Privilege 확인
  - ✓ 데이터베이스 객체에 대한 동의어 생성

- 데이터베이스 보안의 범주
  - ✓ 시스템 보안: 사용자에 의해 허용된 시스템 작업 같은 시스템 수준에서의 데이터베이스의 액세스와 사용을 규정
    - 사용자 명
    - 사용자의 비밀 번호
    - 사용자에게 할당된 디스크 공간
  - ✓ 데이터 보안: 객체에 대해 사용자가 할 수 있는 작업을 규정

2. 사용자 생성

- DBA는 CREATE USER문장을 사용하여 사용자를 생성하고 사용자는 생성 후 어떠한 권한도 가지지 않기 때문에 DBA는 이때 그 사용자에게 여러 권한을 부여
- 권한은 데이터베이스 수준에서 사용자가 할 수 있는 것이 무엇인가를 결정

```
CREATE  USER  user_name
IDENTIFIED  BY  password;
```

- 사용자 명은 ggangpae1, 패스워드는 tjoeun인 사용자를 생성하고 CONNECT, RESOURCE권한을 부여

```
CREATE USER ggangpae1
IDENTIFIED BY tjoeun;
```

3. 권한

- 권한은 특정 SQL문장을 실행하기 위한 권한으로 데이터베이스 관리자는 데이터베이스와 그 객체에 대한 액세스를 사용자에게 부여하는 능력을 가진 상급 사용자
- 사용자는 데이터베이스에 액세스하기 위해 system privilege가 필요하고 데이터베이스에서 객체의 내용을 조작하기 위해 object privilege가 필요
- 사용자는 관련 권한들의 이름있는 그룹인 role이나 다른 사용자에게 추가적으로 권한을 부여하기 위해 권한을 가질 수 있음
- 시스템 권한
  - 사용자와 ROLE에 대해 부여할 수 있는 시스템 권한의 종류는 80개 이상
  - 시스템 권한은 주로 DBA가 부여
  - DBA는 상급의 시스템 권한을 가짐
    - ✓ 새로운 사용자 생성(CREATE USER)
    - ✓ 사용자 제거(DROP USER)
    - ✓ 테이블 제거(DROP ANY TABLE)
    - ✓ 테이블 백업(BACKUP ANY TABLE)

3. 권한

❖ 권한 부여 Syntax

```
GRANT                  system_privilege1[,system_privilege2, . . . . .]
TO                     user_name1[,user_name2, . . . . . .]
[WITH ADMIN OPTION];
```

system_privilege	시스템 권한
user_name	사용자 명
WITH ADMIN OPTION	받은 시스템 권한을 다른 사용자에게 부여할 수 있는 권한

3. 권한

❖ 시스템 권한의 종류

```
SELECT *
FROM system_privilege_map;
```

system_privilege	시스템 권한
user_name	사용자 명
WITH ADMIN OPTION	받은 시스템 권한을 다른 사용자에게 부여할 수 있는 권한

3. 권한

시스템 권한	허가된 내용(Grantee:권한을 받은 사용자)
ALTER ANY TABLE	Grantee가 Schema에 있는 Index를 Alter
ALTER ANY PROCEDURE	Grantee가 Schema에 내장 프로시저,함수,또는 패키지 바꾸기
ALTER ANY ROLE	Grantee가 데이터베이스에서 역할 바꾸기
ALTER ANY TABLE	Grantee가 Schema에서 TABLE이나 VIEW를 변경
ALTER ANY TRIGGER	Grantee가 Schema에서 데이터베이스 TRIGGER를 활성화,비활성화 또는 Compile 가능
ALTER DATABASE	Grantee가 데이터베이스 바꾸기를 허용
ALTER USER	Grantee가 사용자 바꾸기를 할 수 있으며 이 권한은 Grantee가 다른 사용자의 Password나 확인 방법을 바꾸도록 권한을 주고 DEFAULT TABLESPACE, TEMPORARY TABLESPACE, PROFILE, QUOTA의 양을 바꿀 수 있도록 함
CREATE ANY INDEX	Grantee가 어떤 Schema에서나 테이블에 인덱스 만들기를 허용
CREATE ANY PROCEDURE	Grantee가 어떤 Schema에서 내장 프로시저,함수,패키지를 만들 수 있도록 허용
CREATE ANY TABLE	Grantee가 어떤 Schema에서나 테이블을 만들 수 있도록 허용
CREATE ANY TRIGGER	Grantee가 어떤 Schema에서나 테이블과 연관된 Schema에서 데이터베이스 트리거를 만들 수 있도록 허용
CREATE ANY VIEW	Grantee가 어떤 Schema에서나 VIEW를 만들 수 있도록 허용
CREATE PROCEDURE	Grantee가 자체 Schema에서 내장 프로시저,함수,패키지를 만들 수 있도록 허용

3. 권한

시스템 권한	허가된 내용(Grantee:권한을 받은 사용자)
CREATE PROFILE	Grantee가 PROFILE을 만들 수 있도록 허용
CREATE ROLE	Grantee가 ROLE을 만들 수 있도록 허용
CREATE SYNONYM	Grantee가 자체 Schema에서 시너임을 만들 수 있도록 허용
CREATE TABLE	Grantee가 자체 Schema에서 테이블을 만들 수 있도록 허용
CREATE TRIGGER	Grantee가 자체 Schema에서 트리거를 만들 수 있도록 허용
CREATE USER	Grantee가 사용자를 만들 수 있도록 허용
CREATE VIEW	Grantee가 자체 Schema에서 VIEW을 만들 수 있도록 허용
DELETE ANY TABLE	Grantee가 어떤 Schema에서 테이블의 자료를 삭제 가능
DROP ANY INDEX	Grantee가 어떤 Schema에서나 인덱스를 삭제 가능
DROP ANY PROCEDURE	Grantee가 어떤 Schema에서나 내장 프로시저,함수,패키지를 삭제
DROP ANY ROLE	Grantee가 ROLE을 삭제하도록 허용
DROP ANY SYNONYM	Grantee가 어떤 Schema에서나 시너임을 삭제할 수 있도록 허용
DROP ANY TABLE	Grantee가 어떤 Schema에서나 테이블을 식제할 수 있도록 허용

3. 권한

시스템 권한	허가된 내용(Grantee:권한을 받은 사용자)
DROP ANY TRIGGER	Grantee가 어떤 Schema에서나 데이터베이스 트리거를 삭제할 수 있도록 허용
DROP ANY VIEW	Grantee가 어떤 Schema에서나 VIEW를 삭제할 수 있도록 허용
DROP USER	Grantee가 사용자를 삭제할 수 있도록 허용
EXECUTE ANY PROCEDURE	Grantee가 어떤 Schema에서나 프로시저,함수,패키지를 실행
TRANSACTION	Local Database에서 자체의 불안정한 분산 Transaction의 BACK을 허용
GRANT ANY PRIVILEGE	Grantee가 시스템 권한을 주는 것을 허용
GRANT ANY ROLE	Grantee가 데이터베이스에서 어떠한 ROLE이라도 GRANT할 수 있는 권한을 허용
INSERT ANY TABLE	Grantee가 어떠한 Schema에서나 테이블과 VIEW에 자룔를 삽입할 수 있도록 허용
LOCK ANY TABLE	Grantee가 어떤 Schema에서나 테이블과 VIEW에 LOCK을 걸도록 허용
SELECT ANY SEQUENCE	Grantee가 어떤 Schema에서나 시퀀스를 참조할 수 있도록 허용
SELECT ANY TABLE	Grantee가 어떤 Schema에서나 테이블,VIEW, Snapshot을 참조할 수 있도록 허용
UPDATE ANY	Grantee가 테이블에서 행을 수정하도록 허용

3. 권한

❖ ggangpae1에게 CREATE ROLE 권한을 부여

```
GRANT CREATE ROLE TO ggangpae1;

Grant succeeded.
```

3. 권한

❖ 권한 취소 Syntax

```
REVOKE  system_privilege1[,system_privilege2, . . . .] | role1[,role2, . . . .]
FROM {user1[,user2, . . . .] | role1[,role2, . . . .] | PUBLIC};
```

system_privilege	시스템 권한
user_name	사용자 명
WITH ADMIN OPTION	받은 시스템 권한을 다른 사용자에게 부여할 수 있는 권한

❖ ggangpae1에게 부여된 CREATE ROLE 권한을 취소

```
REVOKE create role FROM ggangpae1;
```

❖ Guidelines

사용자가 WITH ADMIN OPTION으로 권한을 부여 받았다면 그 사용자는 WITH ADMIN OPTION으로 권한을 부여해줄 수 있어 수여자 간의 긴 체인이 가능하지만 소유자가 다른 사용자에게 부여한 권한을 취소하면 모든 권한을 연이어 취소하지는 않음

3. 권한

- 객체 권한
  - ✓ 객체 권한은 객체마다 다양
  - ✓ 객체 소유자는 객체에 대한 모든 권한을 가지고 있음
  - ✓ 소유자는 사용자 객체에 대한 특정 권한을 제공할 수 있음

- 주의
  - ✓ DBA는 일반적으로 시스템 권한을 할당
  - ✓ 객체를 소유한 모든 사용자는 객체 권한을 부여할 수 있음
  - ✓ WITH GRANT OPTION으로 부여 받은 권한은 부여자에 의해 다른 사용자와 ROLE에게 다시 부여될 수 있으며 WITH GRANT OPTION으로 테이블을 질의할 수 있고 테이블에 행을 추가할 수 있도록 해 줌
  - ✓ 테이블의 소유자는 PUBLIC 키워드를 사용하여 모든 사용자에게 액세스 권한을 부여할 수 있음

3. 권한

객체 권한	TABLE	VIEW	SEQUENCE	PROCEDURE	SNAPSHOT
ALTER	♣		♣		
DELETE	♣	♣			
EXECUTE				♣	
INDEX	♣				
INSERT	♣	♣			
REFERENCES	♣				
SELECT	♣	♣	♣		♣
UPDATE	♣	♣			

3. 권한

❖ OBJECT 권한의 종류

```
SELECT *
FROM table_privilege_map;

PRIVILEGE NAME
--------- ----------------------------------------
        0 ALTER
        1 AUDIT
        2 COMMENT
        3 DELETE
. . . . . .
26 rows selected.
```

3. 권한

OBJECT 권한	허가된 내용(Grantee:권한을 받은 사용자)
ALTER	Grantee가 OBJECT에 대해 ALTER할 수 있도록 허용
AUDIT	Grantee가 OBJECT에 대해 감사할 수 있도록 허용
COMMENT	Grantee가 OBJECT에 대해 COMMENT할 수 있도록 허용
DELETE	Grantee가 OBJECT에 대해 자료를 삭제할 수 있도록 허용
GRANT	Grantee가 OBJECT에 대해 GRANT할 수 있도록 허용
INDEX	Grantee가 OBJECT에 대해 인덱스를 생성할 수 있도록 허용
INSERT	Grantee가 OBJECT에 대해 자료를 삽입할 수 있도록 허용
LOCK	Grantee가 OBJECT에 대해 Locking할 수 있도록 허용
RENAME	Grantee가 OBJECT에 대해 이름을 변경할 수 있도록 허용
SELECT	Grantee가 OBJECT에 대해 자료를 조회할 수 있도록 허용
UPDATE	Grantee가 OBJECT에 대해 자료를 갱신할 수 있도록 허용
REFERENCES	Grantee가 OBJECT에 대해 자료를 참조할 수 있도록 허용
EXECUTE	Grantee가 프로시저,함수,패키지에 대해 실행할 수 있도록 허용

3. 권한

❖ ggangpae1 사용자에게 SCOTT이 소유하고 있는 EMP 테이블을 조회하고 삽입할 수 있는 권한을 부여

```
GRANT SELECT, INSERT ON EMP TO ggangpae1;

conn ggangpae1/tjoeun

SELECT EMPNO, ENAME, JOB, HIREDATE, SAL
FROM scott.EMP
WHERE DEPTNO = 10;
```

```
    EMPNO ENAME      JOB       HIREDATE                SAL
--------- ---------- --------- ------------------ ---------
     7839 KING       PRESIDENT 17-NOV-81                5000
     7782 CLARK      MANAGER   09-JUN-81                2450
     7934 MILLER     CLERK     23-JAN-82                1300
```

3. 권한

- 객체 권한 철회: 다른 사용자에게 부여된 권한을 철회하기 위하여 REVOKE문장을 사용하며 WITH GRANT OPTION을 통해 다른 사용자에게 부여된 권한을 같이 취소
- syntax

```
REVOKE          {object_privilege1[,object_privilege2, . . . . .] | ALL}
ON    object_name
FROM{user1[,user2, . . . .] | role1[,role2, . . . . .] | PUBLIC}
[CASCADE  CONSTRAINTS];
```

CASCADE CONSTRAINTS REFERENCES권한을 사용하여 만들어진 객체에 대한 참조 무결성 제약 조건을 제거하기 위해 사용

3. 권한

❖ EMP 테이블에 부여한 SELECT권한을 ggangpae1에게서 취소

```
REVOKE SELECT ON EMP
FROM ggangpae1;

conn ggangpae1/tjoeun;

SELECT *
FROM scott.EMP;

FROM scott.EMP
          *
ERROR at line 2:
ORA-00942: table or view does not exist
```

3. 권한

❖ 부여된 권한 확인

데이터 사전 테이블	설 명
ROLE_SYS_PRIVS	ROLE에게 부여된 시스템 권한
ROLE_TAB_PRIVS	ROLE에게 부여된 테이블 권한
USER_ROLE_PRIVS	사용자에 의해 액세스 가능한 ROLE.
USER_TAB_PRIVS_MADE	사용자가 부여된 객체 권한
USER_TAB_PRIVS_RECO	사용자에게 부여된 객체 권한
USER_COL_PRIVS_MADE	사용자가 객체의 열에 대해 부여한 객체 권한
USER_COL_PRIVS_RECO	특정 열에 대해 사용자에게 부여된 객체 권한

❖ SCOTT에게 할당되어 있는 SYSTEM ROLE을 확인

```
SELECT *
FROM ROLE_SYS_PRIVS;
```

4. ROLE

- 사용자에게 허가할 수 있는 관련된 권한들의 그룹
- ROLE을 이용하면 권한 부여와 회수를 쉽게 할 수 있는데 한 사용자가 여러 ROLE을 액세스할 수 있고 다른 여러 사용자에게 같은 ROLE을 지정할 수 있음
- ROLE을 생성하기 위해서는 CREATE ROLE권한 또는 DBA 권한이 필요

- ROLE의 작성과 지정 순서
  - ✓ 먼저 DBA가 ROLE을 생성
  - ✓ ROLE에 권한을 지정
  - ✓ 사용자에게 ROLE을 부여

- Syntax

  CREATE ROLE role_name;

  role_name 생성되는 ROLE의 이름

- LEVEL1이라는 ROLE을 생성

  CREATE ROLE LEVEL1;

4. ROLE

- LEVEL1이라는 ROLE에 CREATE SESSION,CREATE TABLE,CREATE VIEW 의 권한을 부여

```
GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW
TO LEVEL1;
```

- TEST1/TIGER1과 TEST2/TIGER2라는 사용자를 생성

```
CREATE USER TEST1
IDENTIFIED BY TIGER1;

CREATE USER TEST2
IDENTIFIED BY TIGER2;
```

- TEST1,TEST2에 LEVEL1이라는 ROLE를 부여하여라.

```
GRANT LEVEL1
TO TEST1, TEST2;
```

연습문제

1. 사용자는 IRIN 이고 패스워드는 SM 인 사용자를 생성

2. 이전에 생성된 사용자에게 CONNECT와 RESOURCE 권한을 부여

3. 이전에 부여한 권한을 취소하고 IRIN 사용자를 삭제

5. 백업과 복원

❖ 백업(Export) :데이터와 구조를 바이너리 파일로 저장 – 터미널에서 작업

exp userid=아이디/비밀번호@전역데이터베이스명 file=저장경로

system계정으로 전체 백업

exp userid=system/비밀번호@전역데이터베이스명 full=y file=c:₩dump.dmp

system 계정으로 scott 계정에 있는 DB백업

exp userid=system/비밀번호@전역데이터베이스명 owner=scott file=c:₩dump.dmp

scott계정으로 자신의 모든 데이터 백업

exp userid=scott/비밀번호@전역데이터베이스명 file=c:₩dump.dmp

scott계정으로 emp테이블만 백업

exp userid=scott/비밀번호@전역데이터베이스명 file=c:₩dump.dmp tables=emp

여러 개 테이블을 동시에 백업 받으려면 tables=(테이블1,테이블2,...)

백업 파일의 확장자는 보통 .dmp 혹은 .dat .bak

5. 백업과 복원

❖ 복원(Import)

imp 아이디/비밀번호@전역데이터베이스명 file=백업경로

system계정으로 전체 복원

imp system/비밀번호@전역데이터베이스명 file=c:₩dump.dmp

system 계정으로 scott 계정에 있는 DB복원

imp system/비밀번호@전역데이터베이스명 fromuser=scott touser=scott file=c:₩dump.dmp

scott계정으로 자신의 모든 데이터 복원

imp scott/비밀번호@전역데이터베이스명 file=file=c:₩dump.dmp

복원하고자 하는 DB에 같은 이름의 Object가 있을때,오류를 무시하고 건너 띄고 싶을때 ignore 옵션사용

imp 아이디/비밀번호@전역데이터베이스명 file=c:₩dump.dmp ignore=y

❖ system계정으로 들어가 scott에서 EXPORT한 데이터를 scott2에게 IMPORT

imp system/비밀번호@전역데이터베이스명 fromuser=scott touser=scott2 file=c:₩dump.dmp