SQL

✓ 원리를 알면 IT가 맛있다

SQL for Beginners

chapter 12.

사용자 관리

❑ 학습목표

- 사용자 생성
- 권한종류
- 시스템 권한
- 객체 권한
- 권한할당 및 취소
- WITH GRANT OPTION
- 롤(role) 역할

❑ 1) 사용자 관리

- 데이터베이스 보안
  : 다중 사용자 환경에서 개별 사용자들은 데이터베이스 접근 및 사용에 있어서 적절한 보안을 유지해야 한다.

- 데이터베이스 보안 종류

1. 시스템 보안
   : 사용자 계정 생성, 암호변경, 디스크 공간 할당, 시스템 작업등과 같이 시스템 수준에서의 데이터베이스 접근 및 사용을 관리하는 것.
   : **인증(Authentication)**관련

2. 데이터 보안
   : 데이터베이스 객체에 대한 사용자들의 접근 및 사용을 관리하는 것이다.
   : **권한(Authorization)**관련

- 데이터베이스 보안을 위한 서버 작업
  : **데이터베이스 접근 제어**
  : **데이터베이스 특정 객체에 대한 접근 권한 부여**
  : **데이터베이스 객체에 대한 동의어 작성**

❑ 1) 사용자 관리

- 사용자 생성
  - 데이터베이스 관리자(DBA)가 사용자를 생성한다.
  - 생성된 사용자는 아무런 권한도 부여되지 않았기 때문에 어떠한 작업도 불가능하다.

```
CREATE USER user
IDENTIFIED BY password;
```

```
SQL> CONN / as sysdba
연결되었습니다.
SQL> CREATE USER user01
  2  IDENTIFIED BY oracle;

사용자가 생성되었습니다.
```

```
SQL> ALTER USER user01
  2  IDENTIFIED BY user01;

사용자가 변경되었습니다.
```

- 권한 (Privilege)

  - 특별한 SQL문을 실행 할 수 있는 권리를 의미한다.
  - DBA가 일반 사용자에게 데이터베이스와 데이터베이스 객체에 접근 할 수 있는 권한을 부여 할 수 있다.
  - 일반사용자도 다른 사용자 또는 롤(Role)에게 권한을 부여 할 수 있는 권한을 부여 받을 수 있다. (WITH GRANT OPTION)

- 권한 종류

  1. 시스템 권한
     - 사용자의 데이터베이스 접근권한 (DBA 가 부여한다)
     - 사용자가 데이터베이스에 특별한 작업을 수행하는 것을 가능하게 해줌.

  2. 객체 권한
     - 데이터베이스내의 객체의 내용을 조작하기 위한 권한
     - 사용자가 특정 객체에 접근하고 조작하는 것을 가능하게 해줌.

- 스키마 (Schema)

  - 테이블, 뷰 , 시퀀스와 같은 객체들의 모음을 의미한다.
  - 데이터베이스 사용자가 소유하며, 사용자 이름과 동일한 이름을 갖는다.

- 시스템 권한 (System Privilege)

Typical DBA Privileges

System Privilege	Operations Authorized
CREATE USER	Grantee can create other Oracle users.
DROP USER	Grantee can drop another user.
DROP ANY TABLE	Grantee can drop a table in any schema.
BACKUP ANY TABLE	Grantee can back up any table in any schema with the export utility.
SELECT ANY TABLE	Grantee can query tables, views, or materialized views in any schema.
CREATE ANY TABLE	Grantee can create tables in any schema.

- 권한 부여

: 사용자가 생성되면 DBA는 특정한 시스템 권한을 사용자에게 부여해야 한다.

```
GRANT privilege [, privilege ...]
TO user [, user | role | PUBLIC ...];
```

❑ 1) 사용자 관리

- 사용자에게 부여 가능한 시스템 권한 종류

After the DBA creates a user, the DBA can assign privileges to that user.

System Privilege	Operations Authorized
CREATE SESSION	Connect to the database.
CREATE TABLE	Create tables in the user's schema.
CREATE SEQUENCE	Create a sequence in the user's schema.
CREATE VIEW	Create a view in the user's schema.
CREATE PROCEDURE	Create a stored procedure, function, or package in the user's schema.

```
SQL> GRANT CREATE SESSION, CREATE TABLE
  2  TO user01;

권한이 부여되었습니다.
```

```
SQL> CONN USER01/USER01
연결되었습니다.
SQL>
```

```
SQL> SELECT * FROM SESSION_PRIVS;

PRIVILEGE
----------------------------------------
CREATE SESSION
CREATE TABLE
```

- 시스템 권한 회수

```
SQL> REVOKE CREATE SESSION
  2  FROM USER01;

권한이 취소되었습니다.
```

```
SQL> CONN USER01/USER01
ERROR:
ORA-01045: user USER01 lacks CREATE SESSION privilege; logon denied

경고: 이제는 ORACLE에 연결되어 있지 않습니다.
```

❑ 1) 사용자 관리

- 객체 권한 (Object Privilege)

  : 특정 테이블, 뷰, 시퀀스, 프로시저 등에 특별한 작업을 수행 할 수 있는 권리이다.
  : 객체에 따라서 부여 할 수 있는 권한이 모두 다르다.
  : **사용자는 자신의 스키마에 저장된 모든 객체에 대하여 권한을 갖는다.**
  따라서 다른 사용자 또는 롤(role)에게 자신이 소유한 권한을 부여할 수 있다.

Object Privileges

Object Privilege	Table	View	Sequence	Procedure
ALTER	√		√	
DELETE	√	√		
EXECUTE				√
INDEX	√			
INSERT	√	√		
REFERENCES	√			
SELECT	√	√	√	
UPDATE	√	√		

❑ 1) 사용자 관리

■ 객체 권한 부여

```
GRANT object_priv [(columns)]
ON object
TO {user | role | PUBLIC}
[WITH GRANT OPTION];
```

* WITH GRANT OPTION
: 권한을 부여 받은 사람이 부여 받은
권한을 다른 사용자에게 다시 부여할 수 있는 방법

```
SQL> CONN scott/tiger
연결되었습니다.
SQL> GRANT SELECT
  2  ON DEPT
  3  TO user01;

권한이 부여되었습니다.
```

```
SQL> CONN user01/user01;
연결되었습니다.
SQL> SELECT *
  2  FROM DEPT;
FROM DEPT
     *
2행에 오류:
ORA-00942: 테이블 또는 뷰가 존재하지 않습니다
```

```
SQL> SELECT *
  2  FROM SCOTT.DEPT;

    DEPTNO DNAME                        LOC
---------- ---------------------------- ---------
        10 ACCOUNTING                   NEW YORK
        20 RESEARCH                     DALLAS
        30 SALES                        CHICAGO
        40 OPERATIONS                   BOSTON
```

❑ 1) 사용자 관리

```
SQL> CONN SCOTT/TIGER
연결되었습니다.
SQL> GRANT UPDATE(DNAME)
  2  ON DEPT
  3  TO USER01;

권한이 부여되었습니다.


SQL> CONN USER01/USER01
연결되었습니다.
SQL> UPDATE SCOTT.DEPT
  2  SET LOC ='AAA';
UPDATE SCOTT.DEPT
       *
1행에 오류:
ORA-01031: 권한이 불충분합니다


SQL> UPDATE SCOTT.DEPT
  2  SET DNAME='인사'
  3  WHERE DEPTNO = 40;

1 행이 갱신되었습니다.
```

```
SQL> CONN SCOTT/TIGER
연결되었습니다.
SQL> GRANT SELECT, INSERT
  2  ON DEPT
  3  TO USER01
  4  WITH GRANT OPTION;

권한이 부여되었습니다.

SQL> CONN USER01/USER01
연결되었습니다.
SQL> GRANT SELECT
  2  ON SCOTT.DEPT
  3  TO PUBLIC;

권한이 부여되었습니다.

SQL> CONN HR/HR
연결되었습니다.
SQL> SELECT * FROM SCOTT.DEPT;

    DEPTNO DNAME                        LOC
---------- ---------------------------- ---------
        10 ACCOUNTING                   NEW YORK
        20 RESEARCH                     DALLAS
        30 SALES                        CHICAGO
        40 OPERATIONS                   BOSTON
```

❑ 1) 사용자 관리

■ 객체 권한 회수

: WITH GRANT OPTION 에 의해 다른 사람에게 부여된 권한도 **연쇄적으로 회수**된다.

```
REVOKE {privilege [, privilege ...] | ALL}
ON object
FROM {user [, user ...] | role | PUBLIC}
[CASCADE CONSTRAINTS];
```

```
SQL> REVOKE SELECT ,INSERT
  2  ON DEPT
  3  FROM USER01;

권한이 취소되었습니다.
```

■ CASCADE CONSTRAINTS 실습 예제

: references 권한에 의해 객체에 부여된 참조 무결성 제약조건도 삭제된다.

1. A가 B에게 자신의 테이블에 대해 references, update, select 권한을 할당한다.
2. B는 A 테이블에 대해 참조하는 테이블을 생성한다.
3. A는 B에게서 권한을 revoke하려고 하지만 , 바로 revoke할 수 없다.
4. cascade constraints 옵션을 이용하여 revoke 한다.

```
SQL> CONN SCOTT/TIGER
연결되었습니다.
SQL> GRANT REFERENCES , UPDATE, SELECT
  2  ON DEPT
  3  TO USER01;

권한이 부여되었습니다.

SQL> CONN USER01/USER01
연결되었습니다.
SQL> CREATE TABLE DEPT_REF
  2  ( D_NO NUMBER(2) REFERENCES SCOTT.DEPT(DEPTNO) );

테이블이 생성되었습니다.

SQL> INSERT INTO DEPT_REF
  2  VALUES ( 30 );

1 개의 행이 만들어졌습니다.
```

```
SQL> CONN SCOTT/TIGER
연결되었습니다.
SQL> REVOKE REFERENCES,UPDATE,SELECT
  2  ON DEPT
  3  FROM USER01;
REVOKE REFERENCES,UPDATE,SELECT
*
1행에 오류:
ORA-01981: 현 권한취소를 수행하려면 (

SQL> REVOKE REFERENCES,UPDATE,SELECT
  2  ON DEPT
  3  FROM USER01
  4  CASCADE CONSTRAINTS;

권한이 취소되었습니다.
```

- 롤 (Role)

: 권한들의 묶음을 의미한다.
: 일반 사용자에게 권한을 부여 및 회수하듯이 롤(Role)에 권한 부여 및 회수 가능하다.
: 롤은 시스템 권한과 객체 권한으로 구성될 수 있다.
: 각 사용자에게 부여된 롤은 활성화/비활성화가 가능하다.
: 롤은 특정 사용자가 소유하는 것이 아니기 때문에 어떤 스키마에도 저장되지 않는다.

- 롤 (Role) 장점

: 편리한 권한 관리
: **동적 권한 관리**
  - 롤과 관련된 권한이 변경되면 롤을 부여 받은 모든 사용자들은 자동적으로 변경된 권한을 즉시 부여 받게 된다.
: 권한의 선택적 가용성
  - 롤을 활성화/비활성화가 가능하다.

❑ 2) 롤 [role]

- 빌트인 롤 (Built-in Role)

롤	시스템 권한
CONNECT	CREATE SESSION
RESOURCE	CREATE TABLE, CREATE PROCEDURE, CREATE SEQUENCE CREATE TRIGGER, CREATE TYPE, CREATE CLUSTER CREATE INDEXTYPE, CREATE OPERATOR
SCHEDULAR_ADMIN	CREATE ANY JOB, CREATE JOB EXECUTE ANY CLASS, EXECUTE ANY PROGRAM MANAGE SCHEDULER, CREATE EXTERNALJOB
DBA	대부분의 시스템 권한과 일부 롤을 포함. 일반 사용자에게 부여해서는 안된다.

```
SQL> desc dba_roles;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 ROLE                                      NOT NULL VARCHAR2(30)
 PASSWORD_REQUIRED                                  VARCHAR2(8)

SQL> select role from dba_roles;

ROLE
------------------------------
CONNECT
RESOURCE
DBA
SELECT_CATALOG_ROLE
EXECUTE_CATALOG_ROLE
```

❑ 2) 롤 (role)

- 롤 생성 및 부여 방법

```
SQL> CREATE ROLE clerk;

SQL> GRANT create session, create table
  2  TO clerk;

SQL> GRANT select, insert
  2  ON sh.sales
  3  TO clerk;

SQL> GRANT clerk
  2  TO kim;
```

```
SQL> conn kim/oracle
Connected.
SQL>
SQL> select * from sh.sales
  2  where rownum < 6;

   PROD_ID    CUST_ID TIME_ID   CHANNEL_ID
---------- ---------- --------- ----------
        13        987 10-JAN-98          3
        13       1660 10-JAN-98          3
```

```
SQL> SELECT * FROM dba_roles
  2  WHERE role = 'CLERK';

ROLE                           PASSWORD
------------------------------ --------
CLERK                          NO
```

```
SQL> SELECT * FROM dba_sys_privs
  2  WHERE grantee = 'CLERK';

GRANTEE                        PRIVILEGE
------------------------------ ---------------
CLERK                          CREATE TABLE
CLERK                          CREATE SESSION
```

```
SQL> SELECT * FROM dba_tab_privs
  2  WHERE grantee = 'CLERK';

GRANTEE    OWNER      TABLE_NAME GRANTOR    PRIVILEGE
---------- ---------- ---------- ---------- ----------
CLERK      SH         SALES      SH         SELECT
CLERK      SH         SALES      SH         INSERT
```

```
SQL> SELECT * FROM dba_role_privs
  2  WHERE grantee = 'KIM';

GRANTEE    GRANTED_ROLE
---------- ------------------------
KIM        CLERK
```

❑ 2) 롤 (role)

- 롤의 동적 권한 관리 실습

```
SQL> create user park
  2  identified by oracle;


SQL> create role park_s;

Role created.

SQL> create role park_o;


SQL> grant create session , create table
  2  to park_s;

Grant succeeded.

SQL> grant select, insert
  2  on sh.sales
  3  to park_o;


SQL> grant park_s, park_o
  2  to park;

Grant succeeded.
```

```
SQL> conn park/oracle
Connected.
SQL> select * from session_privs;

PRIVILEGE
----------------------------------
CREATE SESSION
CREATE TABLE

 SQL> conn / as sysdba
 Connected.
 SQL>
 SQL> grant create view
   2  to park_s;

 Grant succeeded.

 SQL> grant update , delete
   2  on sh.sales
   3  to park_o;

 Grant succeeded.

 SQL> conn park/oracle
 Connected.
 SQL> select * from session_privs;

 PRIVILEGE
 ----------------------------------
 CREATE SESSION
 CREATE TABLE
 CREATE VIEW
```

- 계정 잠금

```
SQL> ALTER USER SCOTT ACCOUNT LOCK;

사용자가 변경되었습니다.

SQL> CONN SCOTT/TIGER
ERROR:
ORA-28000: the account is locked


경고: 이제는 ORACLE에 연결되어 있지 않습니다.
SQL> CONN /AS SYSDBA
연결되었습니다.
SQL> ALTER USER SCOTT ACCOUNT UNLOCK;

사용자가 변경되었습니다.

SQL> CONN SCOTT/TIGER
연결되었습니다.
SQL>
```

Thank you