**1. 사용자 접근 제어**

다중 사용자 환경에서는 데이터베이스 액세스와 사용의 보안 유지가 요구 됩니다.

1. 데이터베이스 액세스 제어
2. 데이터베이스에서 특정 객체에 대한 액세스 제공
3. 오라클 데이터 사전으로 주어지고 받는 Privilege 확인
4. 데이터베이스 객체에 대한 동의어 생성

**1.1 데이터베이스 보안의 두 범주**

**1.1.1 시스템 보안**

사용자에 의해 허용된 시스템 작업 같은 시스템 수준에서의 데이터베이스의 액세스와 사용을 규정합니다.

1. 사용자 명
2. 사용자의 비밀 번호
3. 사용자에게 할당된 디스크 공간

**1.1.2 데이터 보안**

객체에 대해 사용자가 할 수 있는 작업을 규정합니다.

**1.2 사용자 생성**

DBA는 CREATE USER문장을 사용하여 사용자를 생성 합니다. 사용자는 생성성 후 어떠한 권한도 가지지 않습니다. DBA는 이때 그 사용자에게 여러 권한을 부여 합니다. 이 권한은 데이터베이스 수준에서 사용자가 할 수 있는 것이 무엇인가를 결정 합니다.

**1.2.1 Syntax**

|  |
| --- |
| **CREATE USER** user\_name  **IDENTIFIED BY** password; |

문제1) 사용자 명은 YJB, 패스워드는 YOON인 사용자를 생성하고 CONNECT,RESOURCE권한을 부여하여라.

|  |
| --- |
| SQL> **conn system/manager**  Connected.  SQL> **CREATE USER yjb**  2 **IDENTIFIED BY yoon;**  User created.  SQL> **GRANT connect,resource TO yjb;**  Grant succeeded. |

**1.3 권한**

권한은 특정 SQL문장을 실행하기 위한 권한입니다. 데이터베이스 관리자는 데이터베이스와 그 객체에 대한 액세스를 사용자에게 부여하는 능력을 가진 상급 사용자 입니다. 사용자는 데이터베이스에 액세스하기 위해 system privilege가 필요하고 데이터베이스에서 객체의 내용을 조작하기 위해 object privilege가 필요합니다. 사용자는 관련 권한들의 이름있는 그룹인 role이나 다른 사용자에게 추가적으로 권한을 부여하기 위해 권한을 가질 수 있습니다.

♣ 참고

스키마는 테이블, 뷰, 시퀀스 같은 객체의 모음입니다. 스키마는 데이터베이스 사용자에 의해 소유되고 사용자와 동일 이름을 가집니다.

**1.3.1 시스템 권한**

1. 사용자와 ROLE에 대해 부여할 수 있는 시스템 권한의 종류는 80개 이상의 있다.
2. 시스템 권한은 주로 DBA가 부여한다.
3. DBA는 상급의 시스템 권한을 가집니다.
4. 새로운 사용자 생성(CREATE USER)
5. 사용자 제거(DROP USER)
6. 테이블 제거(DROP ANY TABLE)
7. 테이블 백업(BACKUP ANY TABLE)

**가) 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

**나) 시스템 권한의 종류**

|  |  |
| --- | --- |
| SQL> **SELECT \***  2 **FROM system\_privilege\_map;**  PRIVILEGE NAME  --------- ----------------------------------------  -3 ALTER SYSTEM  -4 AUDIT SYSTEM  . . . . . . . . . . . .  86 rows selected. | |
| 시스템 권한 | 허가된 내용(Grantee:권한을 받은 사용자) | |
| ALTER ANY TABLE | Grantee가 Schema에 있는 Index를 Alter할 수 있다. | |
| ALTER ANY PROCEDURE | Grantee가 Schema에 내장 프로시저,함수,또는 패키지 바꾸기를 할 수 있다. | |
| ALTER ANY ROLE | Grantee가 데이터베이스에서 역할 바꾸기를 할 수 있다. | |
| ALTER ANY TABLE | Grantee가 Schema에서 TABLE이나 VIEW를 바꾸도록 할 수 있다. | |
| ALTER ANY TRIGGER | Grantee가 Schema에서 데이터베이스 TRIGGER를 활성화,비활성화 또는 Compile 하게할 수 있다. | |
| ALTER DATABASE | Grantee가 데이터베이스 바꾸기를 허용한다. | |
| ALTER USER | Grantee가 사용자 바꾸기를 할 수 있다. 이 권한은 Grantee가 다른 사용자의 Password나 확인 방법을 바꾸도록 권한을 주고 DEFAULT TABLESPACE, TEMPORARY TABLESPACE, PROFILE, QUOTA의 양을 바꿀 수 있도록 한다. | |
| CREATE ANY INDEX | Grantee가 어떤 Schema에서나 테이블에 인덱스 만들기를 허용한다. | |
| CREATE ANY PROCEDURE | Grantee가 어떤 Schema에서 내장 프로시저,함수,패키지를 만들 수 있도록 허용한다. | |
| CREATE ANY TABLE | Grantee가 어떤 Schema에서나 테이블을 만들 수 있도록 허용한다. | |
| CREATE ANY TRIGGER | Grantee가 어떤 Schema에서나 테이블과 연관된 Schema에서 데이터베이스 트리거를 만들 수 있도록 허용한다. | |
| CREATE ANY VIEW | Grantee가 어떤 Schema에서나 VIEW를 만들 수 있도록 허용한다. | |
| CREATE PROCEDURE | Grantee가 자체 Schema에서 내장 프로시저,함수,패키지를 만들 수 있도록 허용한다. | |
| CREATE PROFILE | Grantee가 PROFILE을 만들 수 있도록 허용한다. | |
| CREATE ROLE | Grantee가 ROLE을 만들 수 있도록 허용한다. | |
| CREATE SYNONYM | Grantee가 자체 Schema에서 시너임을 만들 수 있도록 허용한다. | |
| CREATE TABLE | Grantee가 자체 Schema에서 테이블을 만들 수 있도록 허용한다. | |
| CREATE TRIGGER | Grantee가 자체 Schema에서 트리거를 만들 수 있도록 허용한다. | |
| CREATE USER | Grantee가 사용자를 만들 수 있도록 허용한다. | |
| CREATE VIEW | Grantee가 자체 Schema에서 VIEW을 만들 수 있도록 허용한다. | |
| DELETE ANY TABLE | Grantee가 어떤 Schema에서 테이블의 자료를 삭제할 수 있도록 허용한다. | |
| DROP ANY INDEX | Grantee가 어떤 Schema에서나 인덱스를 삭제할 수 있다. | |
| DROP ANY PROCEDURE | Grantee가 어떤 Schema에서나 내장 프로시저,함수,패키지를 삭제할 수 있도록 허용한다. | |
| DROP ANY ROLE | Grantee가 ROLE을 삭제하도록 허용한다. | |
| DROP ANY SYNONYM | Grantee가 어떤 Schema에서나 시너임을 삭제할 수 있도록 허용한다. | |
| DROP ANY TABLE | Grantee가 어떤 Schema에서나 테이블을 식제할 수 있도록 허용한다. | |
| DROP ANY TRIGGER | Grantee가 어떤 Schema에서나 데이터베이스 트리거를 삭제할 수 있도록 허용한다. | |
| DROP ANY VIEW | Grantee가 어떤 Schema에서나 VIEW를 삭제할 수 있도록 허용한다. | |
| DROP USER | Grantee가 사용자를 삭제할 수 있도록 허용한다. | |
| EXECUTE ANY PROCEDURE | Grantee가 어떤 Schema에서나 프로시저,함수,패키지를 실행할 수 있도록 허용한다. | |
| TRANSACTION | Local Database에서 자체의 불안정한 분산 Transaction의 BACK을 허용한다. | |
| GRANT ANY PRIVILEGE | Grantee가 시스템 권한을 주는 것을 허용한다. | |
| GRANT ANY ROLE | Grantee가 데이터베이스에서 어떠한 ROLE이라도 GRANT할 수 있는 권한을 허용한다. | |
| INSERT ANY TABLE | Grantee가 어떠한 Schema에서나 테이블과 VIEW에 자룔를 삽입할 수 있도록 허용한다. | |
| LOCK ANY TABLE | Grantee가 어떤 Schema에서나 테이블과 VIEW에 LOCK을 걸도록 허용한다. | |
| SELECT ANY SEQUENCE | Grantee가 어떤 Schema에서나 시퀀스를 참조할 수 있도록 허용한다. | |
| SELECT ANY TABLE | Grantee가 어떤 Schema에서나 테이블,VIEW, Snapshot을 참조할 수 있도록 허용한다. | |
| UPDATE ANY | Grantee가 테이블에서 행을 수정하도록 허용한다. | |

문제2) SCOTT에게 CREATE ROLE권한을 부여하여라.

|  |
| --- |
| SQL> **conn system/manager**  Connected.  SQL> **GRANT create role TO scott;**  Grant succeeded. |

**1.3.2 시스템 권한 최소**

REVOKE명령으로 시스템 권한을 취소할 수 있다. WITH ADMIN OPTION을 통해 부여된 권한은 취소되지 않는다.

**가) Syntax**

|  |
| --- |
| **REVOKE** system\_privilege1[,system\_privilege2, . . . .] | role1[,role2, . . . .]  **FROM** {user1[,user2, . . . .] | role1[,role2, . . . .] | **PUBLIC**}; |

문제3) SCOTT에게 부여된 CREATE ROLE권한을 취소하여라.

|  |
| --- |
| SQL> **conn system/manager**  Connected.  SQL> **REVOKE create role FROM scott;**  Revoke succeeded. |

☞ 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 취소되지 않는다.

**1.3.3 객체 권한**

1. 객체 권한은 객체마다 다양하다.
2. 객체 소유자는 객체에 대한 모든 권한을 가지고 있다.
3. 소유자는 사용자 객체에 대한 특정 권한을 제공할 수 있습니다.

♣ 주의

DBA는 일반적으로 시스템 권한을 할당합니다. 객체를 소유한 모든 사용자는 객체 권한을 부여할 수 있습니다. WITH GRANT OPTION으로 부여 받은 권한은 부여자에 의해 다른 사용자와 ROLE에게 다시 부여될 수 있습니다. WITH GRANT OPTION으로 테이블을 질의할 수 있고 테이블에 행을 추가할 수 있도록 해 줍니다. 테이블의 소유자는 PUBLIC키워드를 사용하여 모든 사용자에게 액세스 권한을 부여할 수 있습니다.

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

**나) OBJECT 권한의 종류**

|  |
| --- |
| SQL> **SELECT \***  2 **FROM table\_privilege\_map;**  PRIVILEGE NAME  --------- ----------------------------------------  0 ALTER  1 AUDIT  2 COMMENT  3 DELETE  . . . . . . . . . . . . .  13 rows selected. |

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

문제4) 앞에서 생성한 YJB 사용자에게 SCOTT이 소유하고 있는 EMP 테이블을 조회하고 삽입할 수 있는 권한을 부여하여라.

|  |
| --- |
| SQL> **conn scott/tiger**  Connected.  SQL> **GRANT select,insert ON emp TO yjb;**  Grant succeeded.  SQL> **conn yjb/yoon**  Connected.  SQL> **SELECT empno,ename,job,hiredate,sal**  2 **FROM scott.emp**  3 **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 |
| SQL> **conn scott/tiger**  Connected.  SQL> **GRANT select,insert ON emp TO yjb;**  Grant succeeded.  SQL> **conn yjb/yoon**  Connected.  SQL> **CREATE SYNONYM emp FOR scott.emp;**  Synonym created.  SQL> **SELECT empno,ename,job,hiredate,sal**  2 **FROM emp**  3 **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 |

**1.3.4 객체 권한 철회**

다른 사용자에게 부여된 권한을 철회하기 위하여 REVOKE문장을 사용합니다. WITH GRANT OPTION을 통해 다른 사용자에게 부여된 권한도 같이 취소된다.

**가) 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 사용한다.

문제5) 앞에서 EMP 테이블에 부여한 SELECT권한을 YJB에게서 취소하여라.

|  |
| --- |
| SQL> **REVOKE select ON emp FROM yjb;**  Revoke succeeded.  SQL> **conn yjb/yoon;**  Connected.  SQL> **SELECT \***  2 **FROM scott.emp;**  FROM scott.emp  \*  ERROR at line 2:  ORA-00942: table or view does not exist |

☞ Guidelines

사용자가 WITH GRANT OPTION으로 권한을 부여 받았다면 그 사용자는 WITH GRANT OPTION으로 권한을 부여해줄 수 있어 수여자 간의 긴 체인이 가능하지만 원형 부여는 허용되지 않는다. 소유자가 다른 사용자에게 부여한 권한을 취소하면 모든 권한을 연이어 취소된다.

**1.3.5 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 |

문제6) SCOTT에게 할당되어 있는 SYSTEM ROLE를 확인 하여라.

|  |
| --- |
| SQL> **conn scott/tiger**  Connected.  SQL> **SELECT \***  2 **FROM role\_sys\_privs;**  ROLE PRIVILEGE ADM  ------------------------------ ---------------------------------------- ---  CONNECT ALTER SESSION NO  CONNECT CREATE CLUSTER NO  13 rows selected. |

**1.4 ROLE의 개념**

ROLE은 사용자에게 허가할 수 있는 관련된 권한들의 그룹이다. 이러한 ROLE을 이용하면 권한 부여와 회수를 쉽게 할 수 있다. 한 사용자가 여러 ROLE을 액세스할 수 있고 다른 여러 사용자에게 같은 ROLE을 지정할 수 있다. ROLE을 생성하기 위해서는 CREATE ROLE권한 또는 DBA 권한이 필요하다.

**1.4.1 ROLE의 작성과 지정 순서**

1. 먼저 DBA가 ROLE을 생성한다.
2. ROLE에 권한을 지정한다.
3. 사용자에게 ROLE을 부여한다.

**1.4.2 Syntax**

|  |
| --- |
| **CREATE ROLE** role\_name; |

role\_name 생성되는 ROLE의 이름

문제7) LEVEL1이라는 ROLE을 생성하여라.

|  |
| --- |
| SQL> **CONN SYSTEM/MANAGER**  Connected.  SQL> **CREATE ROLE level1;**  Role created. |

문제8) LEVEL1이라는 ROLE에 CREATE SESSION,CREATE TABLE,CREATE VIEW 의 권한을 부여하여라.

|  |
| --- |
| SQL> **GRANT CREATE SESSION,CREATE TABLE,CREATE VIEW,**  2 **TO level1;**  Grant succeeded. |

문제9) TEST1/TIGER1과 TEST2/TIGER2라는 사용자를 생성하여라

|  |
| --- |
| SQL> **CREATE USER test1**  2 **IDENTIFIED BY tiger1;**  User created.  SQL> **CREATE USER test2**  2 **IDENTIFIED BY tiger2;**  User created. |

문제10) TEST1,TEST2에 LEVEL1이라는 ROLE를 부여하여라.

|  |
| --- |
| SQL> **GRANT level1 TO test1,test2;**  Grant succeeded.  SQL> **conn test1/tiger1**  Connected. |

**◈ 연 습 문 제 ◈**

1. Oracle8에 로그 온하기 위해 필요한 권한은 무엇인가 ?

2. 테이블을 생성하기 위해 필요한 권한은 무엇인가 ?

3. SYSTEM PRIVILEGE와 OBJECT PRIVILEGE란 ?

4. 테이블을 생성한 OWNER는 어떠한 권한을 가지는가 ?

5. 사용자가 비밀 번호를 갱신하려면 어떤 문장을 기술하여야 하는가 ?

6. 사용자가 액세스할 수 있는 테이블을 조회하여라 ?

7. CONNECT ROLE와 RESOURCE ROLE에 대하여 설명하여라.

8. 현재 SESSION을 이루고 있는 사용자가 가지고 있는 OBJECT권한을 조회하여라.

9. 사용자는 KSH이고 패스워드는 KIM인 사용자를 생성하여라.

10. 9번에서 생성된 사용자에게 CONNECT와 RESOURCE권한을 부여하여라.

11. 10번에서 부여한 권한을 취소하고 KSH사용자를 삭제하여라.