```
REM 사용자와 권한설정
 --1. 사용자 접근제어
   --1) 데이타베이스 액세스 제어
   --2) 데이타베이스의 특정 객체에 대한 액세스 제어
   --3) 오라클에서 주어진 Privilege 에 대한 제어
   --4) 동의어(SYNONYM)에 대한 생성

 --2. 데이타베이스 보안
   --1) 시스템 보안
     --a. 사용자 이름
     --b. 사용자 비밀번호
     --c. 사용자에게 할당된 디스크 공간
     --d. 데이타 보안 -- 객체에 대한 사용자가 할 수 있는 작업에 대한 규정
   --2) 사용자 제어
     --a. DBA는 CREATE USER 를 통해 사용자 생성.
     --b. 사용자에게 권한을 부여
     --c. 사용자가 어떤 레벨에서 객체를 사용할 수 있는 지 정의

REM 사용자 계정
--1. DATABASE USER
  --1) SYS
    -- 오라클 Super 사용자 ID
    -- 데이터베이스에서 발생하는 모든 문제를 처리할 수 있는 권한
  --2) SYSTEM
    -- 오라클 데이터베이스 유지보수 관리할 때 사용하는 사용자 ID
    -- SYS 와의 차이점은 데이터베이스를 생성할 수 있는 권한이 없다.
  --3) SCOTT
    -- 처음 오라클 데이터베이스를 사용자를 위해 만들어 놓은 SAMPLE 사용자 ID.
    -- ORACLE 엔진 개발자
  --4) HR
    -- SAMPLE 사용자 ID.

--2. USER CREATION
  --1) 사용자 계정 생성
    -- CREATE USER 구문 사용
    -- 생성했지만, 로그온 하기 위해 CREATE SESSION 권한 필요
    -- Syntax
        --CREATE USER user name IDENTIFIED BY password;

  --2) 사용자 암호 변경
    -- ALTER USER user_name IDENTIFIED BY new_password;

  --3) 사용자 계정 삭제
    -- DROP USER user_name   [CASCADE];

/*
    -- 사용자이름은 jmhan 이고, 패스워드는 123456  인 계정을 생성하시오.
    CREATE USER jmhan
    IDENTIFIED BY 123456;

    CONN jmhan/123456

    ERROR:
    ORA-01045: user JMHAN lacks CREATE SESSION privilege; logon denied
*/

  --4) 사용자 확인
    --현재 사용자에 대한 정보 조회
    --SELECT USERNAME FROM USER_USERS;

    --데이터베이스의 모든 사용자 조회
    --SELECT USERNAME FROM ALL_USERS;

    --데이터베이스의 모든 사용자 조회
    --SELECT USERNAME FROM DBA_USERS;
    --DBA 권한 사용자만 조회 가능


REM 권한(PRIVILEGES)
--1. 권한(Privilege)
  --1) 특정 SQL 문장을 실행하기 위한 권한
  --2) DBA 는 오라클 객체에 대해 사용자에게 액세스에 권한을 관리하는 역할
  --3) 일반 사용자는 특정 객체에 액세스하기 위해서는 SYSTEM PRIVILEGES가 필요하다.
  --4) 일반 사용자는 특정 객체를 조작하기 위해서는 OBJECT PRIVILEGES가 필요하다.
  --5) 일반 사용자는 관련 권한들의 이름있는 그룹(ROLE)이나, 다은 사용자에게 추가적인 권한을 부여하기 위해
  권한을 가질 수 있다.

--※ SCHEMA
  -- 테이블, 뷰, 시퀀스, 동의어, 인덱스 같은 객체의 모음이다.
  -- 데이타베이스 사용자에 의해 소유되고, 사용자와 같이 권한을 가진다.
  -- 사용자와 동일한 이름을 갖는다.

--2. 종류
  --시스템 권한과 스키마 객체 권한

--3. SYSTEM PRIVILEGES
  --1) 사용자와 Role 에  부여할 수 있는 시스템 권한은 208개 이상이다.
  --2) 시스템 권한은 주로 DBA가 부여한다.
  --3) DBA는 아래와 같은 시스템 권한을 갖는다.
    --a. CREATE USER :사용자 생성
    --b. DROP USER : 사용자 삭제
    --c. DROP ANY TABLE : 모든 SCHEMA 에서 TABLE 삭제
    --d. BACKUP ANY TABLE : 모든 SCHEMA 에서 EXP 프로그램을 이용하여 백업
    --e. SELECT ANY TABLE : 모든 SCHEMA 에서 테이블, 뷰, 스냅샷 검색 가능.
    --f. CREATE ANY TABLE : 모든 SCHEMA 에서 테이블 생성 가능.
```

```
  --4) 일반 사용자에게 부여하는 시스템 권한
    --a. CREATE SESSION : 데이터베이스 연결
    --b. CREATE TABLE : 사용자 스키마에 테이블 생성
    --c. CREATE SEQUENCE : 사용자 스키마에 시퀀스 생성
    --d. CREATE VIEW : 사용자 스키마에 뷰 생성
    --e. CREATE PROCEDURE : 사용자 스키마에 저장 프로시저, 함수, 패키지 생성
  --5) 권한부여
    --GRANT {system_privilege | role}
    --TO {user | role | PUBLIC}
    --[WITH ADMIN OPTION]
    --PUBLIC : 모든 사용자에게 지정된 권한을 부여
    --WITH ADMIN OPTION : 부여받은 사용자는 데이터베이스 관리자가 아닌데도 자신이 부여받은 시스템 권한을
    다른 사용자에게 부여할 수 있는 권한도 함께 부여받는다.
/*
    SELECT * FROM system_privilege_map;
    CONN SYSTEM/javaoracle
    GRANT CREATE SESSION TO JMHAN;

    CONN JMHAN/123456
    SHOW USER

    CREATE TABLE EMP
    (
        EMPNO NUMBER(4),
        ENAME VARCHAR2(20),
        JOB VARCHAR2(9),
        DEPTNO NUMBER(2)
    );
    CREATE TABLE EMP
    *
    ERROR at line 1:
    ORA-01031: insufficient privileges

    CONN SYSTEM/javaoracle
    GRANT CREATE TABLE TO JMHAN;

    CONN JMHAN/123456
    CREATE TABLE EMP
    (
        EMPNO NUMBER(4),
        ENAME VARCHAR2(20),
        JOB VARCHAR2(9),
        DEPTNO NUMBER(2)
    );
    CREATE TABLE EMP
    *
    ERROR at line 1:
    ORA-01950: no privileges on tablespace 'SYSTEM'

    --TABLESPACE : 디스크 공간을 소비하는 테이블과 뷰 그리고 그 밖의 다른 데이터베이스 객체들이 저장되는 장소
    --JMHAN 의 테이블스페이스 확인하기
    CONN SYSTEM/javaoracle

    SELECT USERNAME, DEFAULT_TABLESPACE
    FROM DBA_USERS
    WHERE USERNAME IN ('JMHAN', 'SCOTT');

*/

   --6) TABLE에 대한 시스템 권한
     --a. ALTER ANY TABLE
     --b. CREATE ANY TABLE
     --c. CREATE TABLE
     --d. DELETE ANY TABLE
     --e. INSERT ANY TABLE
     --f. LOCK ANY TABLE
     --g. SELECT ANY TABLE
     --h. DROP ANY TABLE

   --7) 일반 시스템 권한
     --a. ALTER ANY ROLE
     --b. ALTER DATABASE
     --c. ALTER USER
     --d. CREATE USER
     --e. CREATE ROLE
     --f. DROP USER
     --g. GRANT ANY PRIVILEGE

   --8) 시스템 권한 제거
     --a. REVOKE 명령어로 권한을 제거(취소)할 수 있다.
     --b. WITH ADMIN OPTION 을 통해서 부여된 권한은 취소가 되지 않는다.
     --c. Syntax
       --REVOKE {system privilege | role}
       --FROM {user | role | PUBLIC};

/*
    REVOKE CREATE SESSION FROM JMHAN;
*/

   --9) 권한 확인
      --딕셔너리는 DBA_SYS_PRIVS
/*
    SELECT *
    FROM dba sys privs
    WHERE grantee = 'SCOTT';
```

```
*/

REM 객체 권한(OBJECT PRIVILEGES)
--1. 객체 권한은 객체마다 다르다
--2. 객체의 소유자는 객체에 대해 모든 권한을 갖는다.
--3. 객체의 소유자는 다른 사용자에게 권한을 부여할 수 있다.
--4. DBA는 일반적으로는 시스템 권한을 할당하고, 객체의 소유자가 객체 권한을 부여한다.
--5. Syntax
     --GRANT  {object privilege | ALL}
     --ON      schema.object_name
     --TO  {user | ROLE | PUBLIC}
     --[WITH GRANT OPTION]

     --REVOKE {object_privilege | ALL}
     --ON     schema.object_name
     --FROM {user | ROLE | PUBLIC};

    --※ WITH GRANT OPTION : 권한을 부여한 사람에 의해 다른 사용자와 ROLE에 다시 부여할 수 있다.
    --※ PUBLIC : 객체의 소유자는 모든 사용자에게 권한을 부여할 수 있다.

--6. 종류
--                          TABLE      VIEW      SEQUENCE      PROCEDURE
--------------------------------------------------------------------------------
--ALTER                 O                          O
--DELETE                   O           O
--EXECUTE                                                              O
--INDEX                    O
--INSERT                   O           O
--REFERENCES            O
--SELECT                   O           O             O
--UPDATE                  O           O

--AUDIT, COMMENT, GRANT, LOCK
   --테이블에 관련한 객체 권한 보기
      --SELECT * FROM table privilege map;

/*
    CONN JMHAN/123456;
    SELECT * FROM SCOTT.EMP;

        SELECT * FROM SCOTT.EMP
                              *
    ERROR at line 1:
    ORA-00942: table or view does not exist

    --한지민에게 emp 테이블을 조회할 수 있는 권한을 부여하시오.
    CONN SCOTT/tiger
    GRANT SELECT ON emp TO jmhan;

    CONN JMHAN/123456;
    SELECT * FROM SCOTT.EMP;

*/
   --8) 객체 권한 확인
     --USER_TAB_PRIVS_MADE
        --현재 자용자가 다른 사용자에게 부여한 권한의 정보 조회
     --USER_TAB_PRIVS_RECD
        --다른 유저가 자신에게 부여된 사용자 권한 조회
/*
    CONN JMHAN/123456
    SELECT * FROM USER_TAB_PRIVS_MADE;
    SELECT * FROM USER_TAB_PRIVS_RECD;

    CONN SCOTT/tiger
    SELECT * FROM USER_TAB_PRIVS_MADE;
    SELECT * FROM USER_TAB_PRIVS_RECD;
*/

   --7) 객체 권한 취소
     --a. REVOKE 를 사용하여 객체 권한을 취소할 수 있다.
     --b. WITH GRANT OPTION 을 통해서 다른 사람에게 부여된 권한도 같이 취소된다.
       --REVOKE object privilege
       --ON schema.object_name
       --FROM user
       --[CASCADE CONSTRAINT]

/*
    --한지민에게 부여한 객체 권한을 취소하시오.
    REVOKE select
    ON scott.emp
    FROM jmhan;

   --8) 각종 딕셔너리
     --a. ROLE_SYS_PRIVS
     --b. ROLE_TAB_PRIVS
     --c. USER_ROLE_PRIVS
     --d. USER_TAB_PRIVS_MADE
     --e. USER_TAB_PRIVS_RECO
     --f. USER_COL_PRIVS_MADE
     --g. USER_COL_PRIVS_RECO

/*
    CONN JMHAN/123456

    SELECT * FROM SCOTT.EMP;  --가능
```

```
    UPDATE SCOTT.EMP SET SAL = SAL * 1.1; --권한없음
    DELETE FROM SCOTT.EMP;  --권한없음

    --각각 SCOTT 가 GRANT UPDATE ON EMP TO JMHAN, GRANT DELETE ON EMP TO JMHAN
    --그렇지 않으면 GRANT ALL ON EMP TO JMHAN 해야 함.
*/

REM ROLE
--1. 사용자에게 허가할 수 있는 관련된 권한들의 집합이다.
--2. 권한들을 부여하거나 취소하기에 보다 쉽게 하기 위해 사용.
  --1) 사용자들에게 일일이 필요한 권한들을 부여하는 것은 여간 번거로운 일이 아님.
  --2) 따라서 다수의 사용자에게 공통적으로 필요한 권한들을 롤에 하나의 그룹으로 묶어 두고, 사용자에게 특정 롤에 대한 권한을 부여할 수 있도록 하는 것이 편함.
--3. 한 사용자가 여러개의 role 에 액세스할 수 있다.
--4. 여러 사용자가 같은 role 에 액세스할 수 있다.
--5. role 을 생성하기 위해서는 DBA, CREATE ROLE 권한이 있어야 한다.
--6. 작성 순서
  --1) DBA가 role 을 생성한다.
  --2) role 에 권한을 할당한다.
  --3) 사용자에게 role 을 부여한다.
--7. Syntax
  --CREATE ROLE role_name;

--8. 사전에 정의된 시스템에서 제공해 주는 role
  --1) CONNECT Role : 사용자가 데이터베이스에 접속할 수 있는 기본적인 시스템 권한 8개
    --ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION, CREATE SYNONYM, CREATE TABLE, CREATE VIEW
  --2) RESOURCE Role : 사용자가 객체를 생성할 수 있도록 시스템 권한으로 묶어 놓음.
    --CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENCE, CREATE TABLE, CREATE TRIGGER
  --3) DBA Role : 사용자들이 소유한 데이터베이스 객체를 관리하고 사용자들을 작성하고 변경하고 제거할 수 있는 모든 권한. 시스템 관리에 필요한 모든 권한을 부여할 수 있는 강력한 권한.

--9. 현재 사용자에게 부여된 롤 확인 딕셔너리
   --SELECT * FROM USER ROLE PRIVS;
/*
    --1. level1 이라는 role 을 생성하시오.
    CREATE ROLE level1;

    --2. level1 에 CREATE SESSION, CREATE TABLE, CREATE VIEW 라는 권한을 할당하시오.
    GRANT create session, create table, create view TO level1;

    --3. test1/tiger, test2/tiger라는 계정을 생성하시오.
    CREATE USER test1
    IDENTIFIED BY tiger;

    CREATE USER test2
    IDENTIFIED BY tiger;

    --4. 사용자 test1, test2에게 level1이라는 role 을 부여하시오.
    GRANT level1 TO test1, test2;

    --5. 사용자로 커넥트하여 롤을 확인하시오.
        CONN test1/tiger;
        SELECT * FROM USER_ROLE_PRIVS;

    --6. 사용자에게 부여된 롤을 회수하시오.
        CONN sys as sysdba;
    REVOKE level1 FROM test1;

    CONN test1/tiger;
        SELECT * FROM USER ROLE PRIVS;

    --7. 롤을 삭제하시오.
    DROP ROLE level1;
```