

## 사용자, 권한, 룰 관리

지금까지 SCOTT 계정의 여러 객체를 활용하여 다양한 SQL문을 사용해 보았습니다. 이 장에서는 SCOTT 계정 같은 오라클 사용자 그리고 각 사용자 권한을 관리하는 기본 명령어를 간단히 알아보겠습니다.

15-1 사용자 관리

15-2 권한 관리

15-3 룰 관리

### 이 장에서 꼭 익혀야 할 것

- 사용자와 권한 관리의 필요성
- 객체 권한의 부여와 취소
- 룰 개념

# 15-1 사용자 관리

## 사용자란?

오라클 데이터베이스를 활용하여 새로운 서비스를 구축한다면 테이블을 비롯한 여러 객체가 필요할 것입니다. 지금까지 여러 SQL문을 사용한 SCOTT 계정으로 접속하여 필요한 테이블과 객체를 생성하여 활용할 수도 있습니다. 하지만 SCOTT 계정은 오라클 데이터베이스를 공부해 본 사람이라면 대부분 비밀번호까지 알고 있는 계정이기 때문에 주요 데이터를 보관하고 관리하기에는 보안 위험이 있습니다. 따라서 SCOTT 계정 외에 오라클 데이터베이스에 접속할 수 있는 새로운 계정이 필요합니다. 이렇게 오라클 데이터베이스에서는 데이터베이스에 접속하여 데이터를 관리하는 계정을 사용자(USER)로 표현합니다.

### 사용자 관리가 필요한 이유

데이터를 활용한 서비스 규모가 크거나 작은 규모의 여러 서비스를 통합한 방식 등 실무에서 사용하는 여러 종류의 서비스는 한 사용자가 관리하기에는 데이터 분량이 너무 방대하거나 구조가 복잡해지는 경우가 많습니다. 따라서 업무 분할과 효율, 보안을 고려하여 업무에 따라 여러 사용자들을 나눕니다.

오라클 데이터베이스는 테이블·인덱스·뷰 등 여러 객체가 사용자별로 생성되므로 업무별 사용자를 생성한 후에 각 사용자 업무에 맞는 데이터 구조를 만들어 관리하는 방식을 사용할 수 있습니다. 반대로 대표 사용자를 통해 업무에 맞는 데이터 구조를 먼저 정의한 뒤에 사용할 수 있는 데이터 영역을 각 사용자에게 지정해 줄 수도 있습니다.

## 데이터베이스 스키마란?

데이터베이스에서 데이터 간 관계, 데이터 구조, 제약 조건 등 데이터를 저장 및 관리하기 위해 정의한 데이터베이스 구조의 범위를 스키마(schema)를 통해 그룹 단위로 분류합니다.

오라클 데이터베이스에서는 스키마와 사용자를 구별하지 않고 사용하기도 합니다. 사용자는 데이터를 사용 및 관리하기 위해 오라클 데이터베이스에 접속하는 객체를 뜻하고, 스키마는 오라클 데이터베이스에 접속한 사용자와 연결된 객체를 의미합니다. 지금까지 사용한 SCOTT 계정을 예로 들면 SCOTT은 사용자이고 SCOTT이 생성한 테이블·뷰·제약 조건·인

덱스·시퀀스·동의어 등 데이터베이스에서 SCOTT 계정으로 만든 모든 객체는 SCOTT의 스키마가 됩니다.

## 사용자 생성

오라클 사용자를 생성할 때는 CREATE USER문을 사용합니다. 다음과 같이 CREATE USER 명령어에는 사용할 수 있는 옵션이 여러 가지 있습니다. 기본적으로 사용자 이름과 패스워드만 지정해 주면 사용자를 생성할 수 있습니다.

<b>CREATE USER</b> 사용자 이름(필수)	기본 형식
<b>IDENTIFIED BY</b> 패스워드(필수)	
<b>DEFAULT TABLESPACE</b> 테이블 스페이스 이름(선택)	
<b>TEMPORARY TABLESPACE</b> 테이블 스페이스(그룹) 이름(선택)	
<b>QUOTA</b> 테이블 스페이스크기 ON 테이블 스페이스 이름(선택)	
<b>PROFILE</b> 프로파일 이름(선택)	
<b>PASSWORD EXPIRE(선택)</b>	
<b>ACCOUNT [LOCK/UNLOCK](선택);</b>	

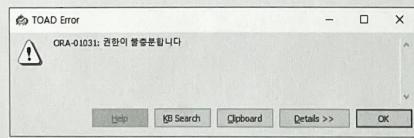
☞ 이 책에서는 사용자를 생성할 때 필요한 기본 옵션만 사용할 것입니다. 좀 더 자세한 내용을 알고 싶다면 오라클 공식 문서(docs.oracle.com/cd/B28359\_01/server.111/b28286/statements\_8003.htm#SQLRF01503)를 참고하세요.

하지만 다음 명령어는 SCOTT 계정으로 접속한 상태에서는 실행되지 않습니다. 사용자를 생성할 권한이 없기 때문이죠.

### 실습 15-1 SCOTT 계정으로 사용자 생성하기

```
01  CREATE USER ORCLSTUDY  
02  IDENTIFIED BY ORACLE;
```

:: 결과 화면



SCOTT 계정으로 접속해 있을 때는 CREATE USER 명령어가 실행되지 않습니다.

사용자 생성은 일반적으로 데이터베이스 관리 권한을 가진 사용자가 권한을 가지고 있습니다. 오라클 데이터베이스를 설치할 때 자동으로 생성된 SYS, SYSTEM이 데이터베이스 관리 권한을 가진 사용자입니다.

여기에서는 SQL\*PLUS를 통해 SYSTEM 사용자로 접속해 보겠습니다. 오라클 데이터베이스를 설치할 때 SYSTEM 사용자의 패스워드를 oracle로 지정했습니다. SYSTEM 사용자로 접속한 후 다음 CREATE USER문을 다시 실행해 보죠.

### 실습 15-2 SYSTEM 사용자로 접속 후 사용자 생성하기(SQL\*PLUS)

```
01  CREATE USER ORCLSTUDY  
02  IDENTIFIED BY ORACLE;
```

:: 결과 화면

```
C:\W>SQLPLUS SYSTEM/oracle  
SQL*Plus: Release 11.2.0.1.0 Production on 목 5월 18 03:52:34 2017  
Copyright (c) 1982, 2010, Oracle. All rights reserved.  
  
다음에 접속됨:  
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production  
With the Partitioning, OLAP, Data Mining and Real Application Testing options  
  
SQL> CREATE USER ORCLSTUDY  
2  IDENTIFIED BY ORACLE;  
 사용자가 생성되었습니다.  
SQL>
```

이 장은 여러 개정 사용의 편의를 위해 SQL\*PLUS에서 실행하고 있습니다. 하지만 토드를 사용해도 상관없습니다.

하지만 CONN 명령어를 사용해 새로 생성한 ORCLSTUDY 사용자로 접속을 시도하면 접속이 되지 않습니다. 이는 사용자가 생성되긴 했지만 데이터베이스 연결을 위한 권한, 즉 CREATE SESSION 권한을 부여받지 못했기 때문입니다.

```
SQL> CONN ORCLSTUDY/ORACLE  
ERROR:  
ORA-01045: user ORCLSTUDY lacks CREATE SESSION privilege; logon denied  
  
경고: 이제는 ORACLE에 연결되어 있지 않습니다.  
SQL>
```

다시 SYSTEM 사용자로 접속하여 다음 명령어를 실행해 봅시다. GRANT문은 권한을 부여하기 위해 사용하는 명령어로 조금 후에 자세히 살펴보겠습니다. 여기에 사용된 GRANT문은 CREATE SESSION 권한을 ORCLSTUDY 사용자에게 부여하고 있으며, 이는 데이터베이스 접속 권한을 주겠다는 의미입니다.

### 실습 15-3 SYSTEM 사용자로 접속 후 ORCLSTUDY 사용자에게 권한 부여하기

```
01 GRANT CREATE SESSION TO ORCLSTUDY;
```

:: 결과 화면

```
SQL> CONN SYSTEM/oracle  
연결되었습니다.  
SQL> GRANT CREATE SESSION TO ORCLSTUDY;
```

권한이 부여되었습니다.

```
SQL>
```

이제 ORCLSTUDY 사용자로 다음과 같이 데이터베이스에 접속할 수 있습니다. ORCLSTUDY 사용자가 SCOTT 계정처럼 테이블을 만들고 데이터를 사용하려면 몇몇 권한이 더 필요합니다. 먼저 사용자 관련 명령어를 살펴본 후 알아보겠습니다.

```
SQL> CONN ORCLSTUDY/oracle  
연결되었습니다.  
SQL>
```

## 사용자 정보 조회

사용자 또는 사용자 소유 객체 정보를 얻기 위해 다음과 같이 데이터 사전을 사용할 수 있습니다.

```
SELECT * FROM ALL_USERS  
WHERE USERNAME = 'ORCLSTUDY';
```

```
SELECT * FROM DBA_USERS  
WHERE USERNAME = 'ORCLSTUDY';
```

```
SELECT * FROM DBA_OBJECTS  
WHERE OWNER = 'ORCLSTUDY';
```

## 오라클 사용자의 변경과 삭제

### 오라클 사용자 변경

앞에서 사용자를 생성할 때는 CREATE USER문을 사용했는데요. 사용자 정보를 변경할 때에는 ALTER USER문을 사용합니다. 앞에서 생성한 ORCLSTUDY 사용자의 패스워드를 ORCL로 변경해 볼까요?

#### 실습 15-4 사용자 정보(패스워드) 변경하기

```
01 ALTER USER ORCLSTUDY  
02 IDENTIFIED BY ORCL;
```

:: 결과 화면

```
SQL> CONN SYSTEM/oracle  
연결되었습니다.  
SQL> ALTER USER ORCLSTUDY  
  2 IDENTIFIED BY ORCL;
```

사용자가 변경되었습니다.

```
SQL>
```

당연한 이야기이지만 ALTER USER문을 통해 패스워드를 변경하면 기존 패스워드로는 접속할 수 없고 새 패스워드를 써야만 합니다. 그리고 사용자 생성과 마찬가지로 사용자 정보 변경도 SYSTEM 사용자로 수행하고 있다는 점 잊지 마세요.

```
SQL> CONN ORCLSTUDY/ORACLE  
ERROR:  
ORA-01017: invalid username/password; logon denied
```

```
경고: 이제는 ORACLE에 연결되어 있지 않습니다.  
SQL> CONN ORCLSTUDY/ORCL  
연결되었습니다.  
SQL>
```

☞ ALTER USER문의 좀 더 자세한 내용을 알고 싶다면 오라클 공식 문서([docs.oracle.com/cd/B28359\\_01/server.111/b28286/statements\\_4003.htm#SQLRF01103](http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_4003.htm#SQLRF01103))와 다른 자료 및 서적을 참고하세요.

#### 오라클 사용자 삭제

DROP USER문을 사용하여 사용자를 삭제합니다. 만약 삭제하려는 사용자가 다른 곳에서 접속되어 있다면 삭제되지 않는다는 점도 주의하세요.

#### 실습 15-5 사용자 삭제하기

```
01 DROP USER ORCLSTUDY;
```

:: 결과 화면

```
SQL> CONN SYSTEM/oracle  
연결되었습니다.  
SQL> DROP USER ORCLSTUDY;
```

사용자가 삭제되었습니다.

```
SQL>
```

☞ DROP USER문으로 데이터베이스 관리 권한을 가진 SYS, SYSTEM 등의 사용자를 삭제하지 않도록 주의하세요.

## 오라클 사용자와 객체 모두 삭제

사용자 스키마에 객체가 있을 경우에 CASCADE 옵션을 사용하여 사용자와 객체를 모두 삭제할 수 있습니다.

### 실습 15-6 사용자와 객체 모두 삭제하기

```
01  DROP USER ORCLSTUDY CASCADE;
```

☞ DROP USER문은 오라클 공식 문서(docs.oracle.com/cd/B28359\_01/server.111/b28286/statements\_9008.htm#SQLRF01811)에서 좀 더 자세한 정보를 확인할 수 있습니다.

1분  
복습

ORCLSTUDY 사용자의 패스워드를 ORASTDY로 변경하는 다음 SQL문의 코드를 채워 보세요.

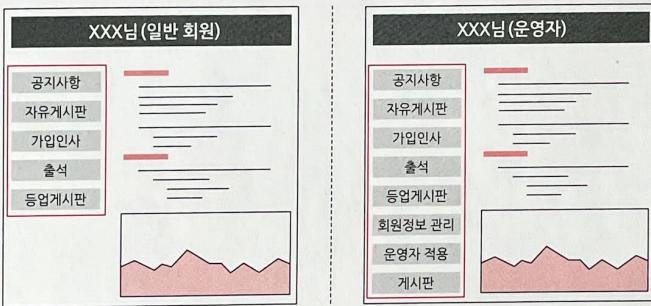
```
1          ORCLSTUDY  
IDENTIFIED BY 2      ;
```

답지 1. ALTER USER 2. ORASTDY

## 15-2 권한 관리

데이터베이스에 보관 및 관리되는 데이터는 대부분 데이터를 소유한 특정 단체 또는 기업에게 재산 이상 가치를 가지는 경우가 많습니다. 따라서 데이터를 안전하게 보관하고 특정 데이터에 대해서 관련된 사용자만 데이터를 사용 및 관리할 수 있는 보안 장치가 필요합니다. 사용자 이름과 패스워드를 통해 데이터베이스 접속을 허가하는 것이 그 첫 번째가 됩니다.

하지만 특정 사용자 정보를 통해 데이터베이스에 접속하는 것만으로 데이터베이스의 모든 데이터를 사용할 수 있다면 여전히 데이터 안전을 보장하기는 어려울 것입니다. 따라서 데이터베이스는 접속 사용자에 따라 접근할 수 있는 데이터 영역과 권한을 지정해 줄 수 있는데요. 오라클에서는 권한을 시스템 권한(system privilege)과 객체 권한(object privilege)으로 분류하고 있습니다. 이제 이 두 가지 권한의 특성과 더불어 권한을 부여하고 회수하는 방법을 알아보겠습니다.



인터넷 카페의 경우 접속 사용자 등급에 따라 사용 가능한 메뉴가 다른데요. 데이터베이스도 이처럼 접속 사용자에 따라 사용 가능한 데이터가 달라지도록 설정할 수 있습니다. 바로 '권한'을 이용해서 말이죠.

### 시스템 권한이란?

오라클 데이터베이스의 시스템 권한(system privilege)은 사용자 생성과 정보 수정 및 삭제, 데이터베이스 접근, 오라클 데이터베이스의 여러 자원과 객체 생성 및 관리 등의 권한을 포함합니다. 이러한 내용은 데이터베이스 관리 권한이 있는 사용자가 부여할 수 있는 권한입니다. 다음은 시스템 권한의 일부이며 ANY 키워드가 들어 있는 권한은 소유자에 상관없이 사용 가능핚 권한을 의미합니다.

시스템 권한 분류	시스템 권한	설명
USER(사용자)	CREATE USER	사용자 생성 권한
	ALTER USER	생성된 사용자의 정보 수정 권한
	DROP USER	생성된 사용자의 삭제 권한
SESSION(접속)	CREATE SESSION	데이터베이스 접속 권한
	ALTER SESSION	데이터베이스 접속 상태에서 환경 값 변경 권한
TABLE(테이블)	CREATE TABLE	자신의 테이블 생성 권한
	CREATE ANY TABLE	임의의 스키마 소유 테이블 생성 권한
	ALTER ANY TABLE	임의의 스키마 소유 테이블 수정 권한
	DROP ANY TABLE	임의의 스키마 소유 테이블 삭제 권한
	INSERT ANY TABLE	임의의 스키마 소유 테이블 데이터 삽입 권한
	UPDATE ANY TABLE	임의의 스키마 소유 테이블 데이터 수정 권한
	DELETE ANY TABLE	임의의 스키마 소유 테이블 데이터 삭제 권한
	SELECT ANY TABLE	임의의 스키마 소유 테이블 데이터 조회 권한
	CREATE ANY INDEX	임의의 스키마 소유 테이블의 인덱스 생성 권한
INDEX(인덱스)	ALTER ANY INDEX	임의의 스키마 소유 테이블의 인덱스 수정 권한
	DROP ANY INDEX	임의의 스키마 소유 테이블의 인덱스 삭제 권한
VIEW(뷰)	(생략)	뷰와 관련된 여러 권한
SEQUENCE(시퀀스)	(생략)	시퀀스와 관련된 여러 권한
SYNONYM(동의어)	(생략)	동의어와 관련된 여러 권한
PROFILE(프로파일)	(생략)	사용자 접속 조건 지정과 관련된 여러 권한
ROLE(룰)	(생략)	권한을 묶은 그룹과 관련된 여러 권한

#### 이하 생략

☞ 오라클 데이터베이스에서 정의하는 권한에 대한 더욱 자세한 정보는 오라클 공식 문서(docs.oracle.com/cd/B28359\_01/server.111/b28286/statements\_9013.htm#BGBCIIEG)를 참고하세요.

## 시스템 권한 부여

앞에서 CREATE USER문을 통해 사용자를 처음 생성한 후 데이터베이스 접속을 허가하기 위해 다음 명령어를 실행했습니다. 이 명령어는 ORCLSTUDY 사용자에게 CREATE SESSION 권한을 부여하겠다는 뜻입니다.

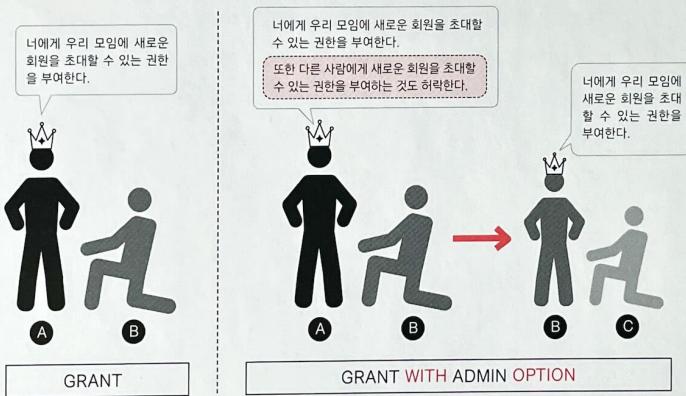
```
GRANT CREATE SESSION TO ORCLSTUDY;
```

이처럼 시스템 권한을 부여할 때 다음과 같이 GRANT문을 사용합니다.

GRANT [시스템 권한] TO [사용자 이름/롤(Role)이름/PUBLIC]  
[WITH ADMIN OPTION];

기본 형식

번호	설명
①	오라클 데이터베이스에서 제공하는 시스템 권한을 지정합니다. 한 번에 여러 종류의 권한을 부여하려면 쉼표(,)로 구분하여 권한 이름을 여러 개 명시해 주면 됩니다(필수).
②	권한을 부여하려는 대상을 지정합니다. 사용자 이름을 지정해 줄 수도 있고, 이후 소개할 룰(role)을 지정할 수도 있습니다. 여러 사용자 또는 룰에 적용할 경우 쉼표(,)로 구분합니다. PUBLIC은 현재 오라클 데이터베이스의 모든 사용자에게 권한을 부여하겠다는 의미입니다(필수).
③	WITH ADMIN OPTION은 현재 GRANT문을 통해 부여받은 권한을 다른 사용자에게 부여할 수 있는 권한도 함께 부여받습니다. 현재 사용자가 권한이 사라져도, 권한을 재부여한 다른 사용자의 권한은 유지됩니다(선택).



WITH ADMIN OPTION을 사용하면 부여받은 권한을 다른 사용자에게 부여할 수 있게 됩니다.  
데이터베이스 관리 권한이 없는 사용자라도 말이죠.

DROP 명령어로 ORCLSTUDY를 지웠다면 다시 CREATE USER 명령어로 생성해 볼까요?  
그리고 GRANT문으로 권한을 부여해 보겠습니다. SQL\*PLUS에 SYSTEM으로 접속하여  
ORCLSTUDY 사용자를 생성합니다.

실습 15-7 SYSTEM 계정으로 접속하여 사용자(ORCLSTUDY) 생성하기(SQL\*PLUS)

```
01 CREATE USER ORCLSTUDY  
02 IDENTIFIED BY ORACLE;
```

다음과 같이 GRANT문을 통해 ORCLSTUDY 사용자에게 데이터베이스 접속 권한과 테이블 생성 권한을 부여합니다.

실습 15-8 사용자 권한 부여하기(SQL\*PLUS)

```
01 GRANT RESOURCE, CREATE SESSION, CREATE TABLE TO ORCLSTUDY;
```

:: 결과 화면

```
SQL> CREATE USER ORCLSTUDY  
2 IDENTIFIED BY ORACLE;  
사용자가 생성되었습니다.  
  
SQL> GRANT CREATE SESSION, CREATE TABLE TO ORCLSTUDY;  
권한이 부여되었습니다.  
  
SQL>
```

ORCLSTUDY 사용자로 데이터베이스 접속과 테이블 생성이 가능해졌음을 알 수 있습니다. ORCLSTUDY 소유 테이블을 생성했으므로 INSERT, SELECT문을 사용할 수 있다는 점도 눈여겨보세요.

```
SQL> CONN ORCLSTUDY/ORACLE  
연결되었습니다.  
SQL> CREATE TABLE TEMP1 <  
2   COL1  VARCHAR2(20),  
3   COL2  VARCHAR2(20)  
4 >;  
테이블이 생성되었습니다.  
  
SQL> INSERT INTO TEMP1 VALUES ('USER', 'GRANT_TEST');  
1 개의 행이 만들어졌습니다.  
  
SQL> SELECT * FROM TEMP1;  
COL1          COL2  
-----  
USER          GRANT_TEST  
SQL>
```

## 한 발 더 나가기!! GRANT에 사용된 RESOURCE 키워드

RESOURCE는 오라클 데이터베이스에서 제공하는 롤(role) 중 하나입니다. 롤은 여러 권한을 하나의 이름으로 묶어 권한 부여 관련 작업을 간편하게 하려고 사용합니다. 앞에서 예제를 통해 생성한 ORCLSTUDY 사용자는 CREATE USER문에서 비밀번호만을 지정하여 생성했습니다.

만약 GRANT문에 RESOURCE를 지정하지 않는다면, ORCLSTUDY 사용자에게 테이블 생성 권한을 부여해도 CREATE문으로 테이블을 생성할 수 없거나 테이블이 생성되더라도 INSERT문에서 다음과 같은 오류 메시지를 출력하며 동작하지 않는 경우가 발생합니다.

ORA-01950: 테이블 스페이스 USERS 권한이 없습니다.

오류 메시지에서 테이블 스페이스는 테이블이 저장되는 공간을 의미하며 따로 지정하지 않으면 기본 테이블 스페이스 USERS가 할당됩니다. 위 오류는 이 테이블 스페이스의 사용 영역을 정하지 않아 발생하는 오류입니다. RESOURCE 롤에는 사용자를 생성할 때 사용 테이블 스페이스의 영역을 무제한 사용 가능(UNLIMITED TABLESPACE)하게 해 주는 권한이 포함되어 있기 때문에, RESOURCE 롤을 GRANT문에 추가하면 별 문제없이 사용자가 테이블을 생성하고 신규 데이터를 저장할 수 있습니다. 하지만 테이블 스페이스의 영역 사용에 한계를 두지 않는 UNLIMITED TABLESPACE 권한은 엄밀한 관리가 필요한 경우에 적절하지 않으므로 사용자를 생성 및 수정할 때 QUOTA 절로 사용 영역에 제한을 두기도 합니다.

```
ALTER USER ORCLSTUDY  
QUOTA 2M ON USERS;
```

이러한 이유 때문에 오라클 데이터베이스 12C 버전에서는 RESOURCE 롤에 UNLIMITED TABLESPACE 권한을 부여하지 않습니다.

## 시스템 권한 취소

GRANT 명령어로 부여한 권한의 취소는 REVOKE 명령어를 사용합니다.

REVOKE [시스템 권한] FROM [사용자 이름/롤(Role)이름/PUBLIC];

기본 형식

REVOKE문을 사용하여 ORCLSTUDY 사용자의 RESOURCE, CREATE TABLE 권한을 취소해 보죠.

```

SQL> CONN SYSTEM/oracle
연결되었습니다.
SQL> REVOKE RESOURCE, CREATE TABLE FROM ORCLSTUDY;
권한이 취소되었습니다.

SQL>

```

권한이 취소된 ORCLSTUDY 사용자는 더 이상 테이블을 생성할 수 없습니다.

```

SQL> CONN ORCLSTUDY/ORACLE
연결되었습니다.
SQL> CREATE TABLE TEMP2 (
  2   COL1  VARCHAR2(20),
  3   COL2  VARCHAR2(20)
  4 );
CREATE TABLE TEMP2 *
*
1행에 오류:
ORA-01031: 권한이 불충분합니다

SQL>

```

### 객체 권한이란?

객체 권한(object privilege)은 특정 사용자가 생성한 테이블·인덱스·뷰·시퀀스 등과 관련된 권한입니다. 예를 들어 SCOTT 소유 테이블에 ORCLSTUDY 사용자가 SELECT나 INSERT 등의 작업이 가능하도록 허용할 수 있습니다. 다음은 주로 사용하는 객체 권한 중 일부입니다.

객체 권한 분류	객체 권한	설명
TABLE(테이블)	ALTER	테이블 변경 권한
	DELETE	테이블 데이터 삭제 권한
	INDEX	테이블 인덱스 생성 권한
	INSERT	테이블 데이터 삽입 권한
	REFERENCES	참조 데이터 생성 권한
	SELECT	테이블 조회 권한
	UPDATE	테이블 데이터 수정 권한
VIEW(뷰)	DELETE	뷰 데이터 삭제 권한
	INSERT	뷰 데이터 삽입 권한
	REFERENCES	참조 데이터 생성 권한
	SELECT	뷰 조회 권한
	UPDATE	뷰 데이터 수정 권한

SEQUENCE(시퀀스)	ALTER SELECT	시퀀스 수정 권한 시퀀스의 CURRVAL과 NEXTVAL 사용 권한
PROCEDURE(프로시저)	(생략)	프로시저 관련 권한
FUNCTION(함수)	(생략)	함수 관련 권한
PACKAGE(패키지)	(생략)	패키지 관련 권한

이하 생략

② 오라클 데이터베이스에서 정의한 객체 권한의 상세 정보가 필요하다면 오라클 공식 문서(docs.oracle.com/cd/B28359\_01/server.111/b28286/statements\_9013.htm#BGBCIIEG)를 참고하세요.

## 객체 권한 부여

객체 권한 부여 역시 GRANT문을 사용합니다.

GRANT [객체 권한/ALL PRIVILEGES] -①  
 ON [스키마.객체 이름] -②  
 TO [사용자 이름/롤(Role)이름/PUBLIC] -③  
 [WITH GRANT OPTION]; -④

기본 형식

번호	설명
①	오라클 데이터베이스에서 제공하는 객체 권한을 지정합니다. 한 번에 여러 종류의 권한을 부여하려면 쉼표(,)로 구분하여 권한을 여러 개 명시해 주면 됩니다. ALL PRIVILEGES는 객체의 모든 권한을 부여함을 의미합니다(필수).
②	권한을 부여할 대상 객체를 명시합니다(필수).
③	권한을 부여하려는 대상을 지정합니다. 사용자 이름을 지정해 줄 수도 있고 이후 소개할 룰(role)을 지정할 수도 있습니다. 여러 사용자 또는 룰에 적용할 경우 쉼표(,)로 구분합니다. PUBLIC은 현재 오라클 데이터베이스의 모든 사용자에게 권한을 부여하겠다는 의미입니다(필수).
④	WITH GRANT OPTION은 현재 GRANT문을 통해 부여받은 권한을 다른 사용자에게 부여할 수 있는 권한도 함께 부여받습니다. 현재 권한을 부여받은 사용자의 권한이 사라지면, 다른 사용자에게 재부여된 권한도 함께 사라집니다(선택).

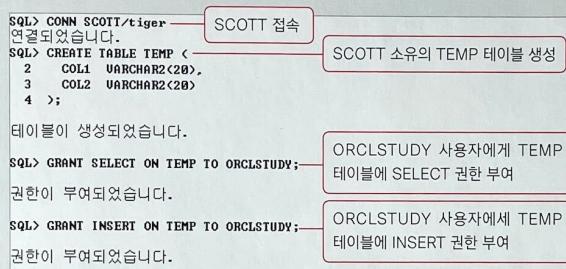
그리면 SCOTT 계정으로 접속하여 새로운 테이블을 하나 만든 후 ORCLSTUDY 사용자에게 해당 테이블의 SELECT, INSERT 권한을 부여해 볼까요?

### 실습 15-9 ORCLSTUDY 사용자에게 TEMP 테이블 권한 부여하기

```
01  CONN SCOTT/tiger  
  
02  CREATE TABLE TEMP(  
03      COL1 VARCHAR(20),  
04      COL2 VARCHAR(20)  
05  );  
  
06  GRANT SELECT ON TEMP TO ORCLSTUDY;  
  
07  GRANT INSERT ON TEMP TO ORCLSTUDY;
```

:: 결과 화면

```
SQL> CONN SCOTT/tiger  
연결되었습니다.  
SQL> CREATE TABLE TEMP (  
 2   COL1 VARCHAR2(20),  
 3   COL2 VARCHAR2(20)  
 4 >;  
테이블이 생성되었습니다.  
SQL> GRANT SELECT ON TEMP TO ORCLSTUDY;  
권한이 부여되었습니다.  
SQL> GRANT INSERT ON TEMP TO ORCLSTUDY;  
권한이 부여되었습니다.
```



위 결과 화면에서는 SELECT와 INSERT 권한을 두 개의 GRANT문으로 나누어 객체 권한을 부여했지만 다음과 같이 쉼표(,)로 구분하여 한 번에 지정할 수도 있습니다.

### 실습 15-10 ORCL에게 TEMP 테이블의 여러 권한을 한 번에 부여하기

```
01  GRANT SELECT, INSERT ON TEMP  
02    TO ORCLSTUDY;
```

이제 SCOTT 계정의 TEMP 테이블 사용을 허가받은 ORCLSTUDY 사용자로 접속해 보죠. SELECT문과 INSERT문도 실행해 보겠습니다.

## 실습 15-11 ORCLSTUDY로 사용 권한을 부여받은 TEMP 테이블 사용하기

```
01  CONN ORCLSTUDY/ORACLE  
  
02  SELECT * FROM SCOTT.TEMP;  
  
03  INSERT INTO SCOTT.TEMP VALUES('TEXT', 'FROM ORCLSTUDY');  
  
04  SELECT * FROM SCOTT.TEMP;
```

:: 결과 화면

```
SQL> CONN ORCLSTUDY/ORACLE  
연결되었습니다.  
SQL> SELECT * FROM SCOTT.TEMP;  
선택된 레코드가 없습니다.  
  
SQL> INSERT INTO SCOTT.TEMP VALUES('TEXT', 'FROM ORCLSTUDY');  
1 개의 행이 만들어졌습니다.  
  
SQL> SELECT * FROM SCOTT.TEMP;  
  
COL1          COL2  
-----  
TEXT          FROM ORCLSTUDY  
  
SQL>
```

ORCLSTUDY 사용자의 소유는 아니지만 SCOTT 계정의 TEMP 테이블을 조회하고 INSERT도 가능해졌습니다.

☞ ORCLSTUDY 사용자로 접속한 상태에서 COMMIT을 한 후 SCOTT 계정으로 접속하여 TEMP 테이블을 조회해 보면 INSERT된 데이터를 확인할 수 있습니다.

## 객체 권한 취소

객체 권한의 취소도 시스템 권한과 마찬가지로 REVOKE문을 사용합니다.

```
REVOKE [객체 권한/ALL PRIVILEGES](필수)  
      ON [スキ마.객체 이름](필수)  
      FROM [사용자 이름/롤(Role) 이름/PUBLIC](필수)  
      [CASCADE CONSTRAINTS/FORCE](선택);
```

기본 형식

☞ REVOKE문의 CASCADE CONSTRAINTS와 FORCE 옵션의 자세한 내용은 오라클 공식 문서([docs.oracle.com/cd/B28359\\_01/server.111/b28286/statements\\_9020.htm#SQLRF01609](http://docs.oracle.com/cd/B28359_01/server.111/b28286/statements_9020.htm#SQLRF01609))를 참고하세요.

그러면 다시 SCOTT 계정으로 접속하여 ORCLSTUDY 사용자에게 부여한 TEMP 테이블 사용 권한을 취소해 볼까요?

### 실습 15-12 ORCLSTUDY에 부여된 TEMP 테이블 사용 권한 취소하기

```
01  CONN SCOTT/tiger  
  
02  REVOKE SELECT, INSERT ON TEMP FROM ORCLSTUDY;
```

:: 결과 화면

```
SQL> CONN SCOTT/tiger  
연결되었습니다.  
SQL> REVOKE SELECT, INSERT ON TEMP FROM ORCLSTUDY;  
권한이 취소되었습니다.  
SQL>
```

REVOKE로 권한을 취소하면 ORCLSTUDY 사용자는 더 이상 SCOTT 계정의 TEMP 테이블을 사용할 수 없게 됩니다.

### 실습 15-13 ORCLSTUDY로 권한 철회된 TEMP 테이블 조회하기(실패)

```
01  CONN ORCLSTUDY/ORACLE  
  
02  SELECT * FROM SCOTT.TEMP;
```

:: 결과 화면

```
SQL> CONN ORCLSTUDY/ORACLE  
연결되었습니다.  
SQL> SELECT * FROM SCOTT.TEMP;  
SELECT * FROM SCOTT.TEMP  
          *  
1행에 오류:  
ORA-00942: 테이블 또는 뷰가 존재하지 않습니다  
SQL>
```



다음 빈칸을 채우며 복습해 보세요.

오라클에서는 새로운 사용자를 생성하기 위해 <sup>1</sup> 문을 사용합니다. 생성된 계정에는 여러 가지 권한을 부여할 수 있습니다. 권한을 부여하기 위해서 사용하는 명령어는 <sup>2</sup> 이며, 부여한 권한을 취소하기 위해서는 <sup>3</sup> 명령어를 사용합니다.

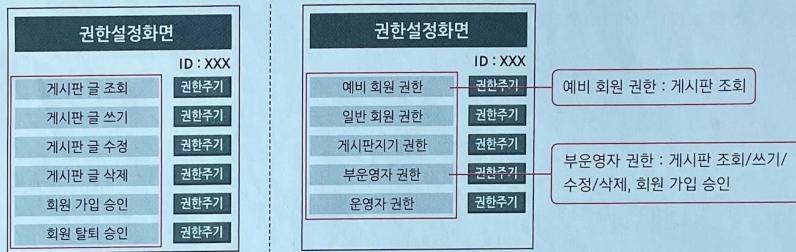
답변 1. CREATE USER 2. GRANT 3. REVOKE

## 15-3 롤 관리

### 롤이란?

앞에서 ORCLSTUDY 사용자를 생성하고 여러 가지 권한을 부여하고 취소해 보았습니다. 사용자는 데이터베이스에서 어떤 작업을 진행하기 위해 해당 작업과 관련된 권한을 반드시 부여받아야 합니다.

하지만 신규 생성 사용자는 아무런 권한이 없으므로 오라클 데이터베이스에서 제공하는 다양한 권한을 일일이 부여해 주어야 합니다. 이러한 불편한 점을 해결하기 위해 룰(role)을 사용합니다. 룰은 여러 종류의 권한을 묶어 놓은 그룹을 뜻합니다. 룰을 사용하면 여러 권한을 한 번에 부여하고 해제할 수 있으므로 권한 관리 효율을 높일 수 있습니다.



권한을 일일이 주는 것보다 룰을 만들어서 여러 종류의 권한을 한 번에 부여한다.

룰은 오라클 데이터베이스를 설치할 때 기본으로 제공되는 사전 정의된 룰(predefined roles)과 사용자 정의 룰(user roles)로 나뉩니다.

### 사전 정의된 룰

#### CONNECT 룰

사용자가 데이터베이스에 접속하는 데 필요한 CREATE SESSION 권한을 가지고 있습니다. 오라클 9i 버전까지는 다음 8가지 권한을 가지고 있었지만 10g 버전부터 CREATE SESSION 권한만 있습니다.

```
ALTER SESSION, CREATE CLUSTER, CREATE DATABASE LINK, CREATE SEQUENCE, CREATE SESSION,  
CREATE SYNONYM, CREATE TABLE, CREATE VIEW
```

### RESOURCE 룰

사용자가 테이블, 시퀀스를 비롯한 여러 객체를 생성할 수 있는 기본 시스템 권한을 묶어 놓은 룰입니다.

```
CREATE TRIGGER, CREATE SEQUENCE, CREATE TYPE, CREATE PROCEDURE, CREATE CLUSTER,  
CREATE OPERATOR, CREATE INDEXTYPE, CREATE TABLE
```

보통 새로운 사용자를 생성하면 CONNECT 룰과 RESOURCE 룰을 부여하는 경우가 많습니다. CONNECT 룰에서 뷰를 생성하는 CREATE VIEW 권한과 동의어를 생성하는 CREATE SYNONYM 권한이 제외되었기 때문에 뷰와 동의어 생성 권한을 사용자에게 부여하려면 이 두 권한을 따로 부여해 주어야 합니다.

☞ 12장에서 뷰와 동의어를 생성하기 위해 SCOTT 계정에 GRANT 명령어를 사용한 것을 떠올려 보세요.

### DBA 룰

데이터베이스를 관리하는 시스템 권한을 대부분 가지고 있습니다. 오라클 11g 버전 기준 202 개 권한을 가진 매우 강력한 룰입니다. 그 밖에도 사전 정의된 룰은 여러 종류가 있습니다.

☞ 좀 더 자세한 내용을 알고 싶다면 오라클 공식 문서([docs.oracle.com/cd/B28359\\_01/network.111/b28531/authorization.htm#DBSEG004](http://docs.oracle.com/cd/B28359_01/network.111/b28531/authorization.htm#DBSEG004))를 참고하세요.

### 사용자 정의 룰

사용자 정의 룰은 필요에 의해 직접 권한을 포함시킨 룰을 뜻합니다. 다음 절차를 따라 룰을 생성해서 사용할 수 있습니다.

- ① CREATE ROLE문으로 룰을 생성합니다.
- ② GRANT 명령어로 생성한 룰에 권한을 포함시킵니다.
- ③ GRANT 명령어로 권한이 포함된 룰을 특정 사용자에게 부여합니다.
- ④ REVOKE 명령어로 룰을 취소시킵니다.

## 롤 생성과 권한 포함

롤을 생성하려면 데이터 관리 권한이 있는 사용자가 필요하므로 SYSTEM 계정으로 접속하여 ROLESTUDY 를을 생성하겠습니다. 룰을 생성한 후 GRANT 명령어로 권한을 포함시킬 수 있습니다.

이미 존재하는 룰도 포함시킬 수 있습니다.

ROLESTUDY 룰에는 CONNECT 롤, RESOURCE 롤 그리고 뷰와 동의어 생성을 위한 CREATE VIEW, CREATE SYNONYM 권한이 포함되었습니다.

### 실습 15-14 SYSTEM 계정으로 ROLESTUDY 를 생성 및 권한 부여하기

01 CONN SYSTEM/oracle

02 CREATE ROLE ROLESTUDY;

03 GRANT CONNECT, RESOURCE, CREATE VIEW, CREATE SYNONYM

04 TO ROLESTUDY;

:: 결과 화면

```
SQL> CONN SYSTEM/oracle
연결되었습니다.
SQL> CREATE ROLE ROLESTUDY;
룰이 생성되었습니다.

SQL> GRANT CONNECT, RESOURCE, CREATE VIEW, CREATE SYNONYM
2   TO ROLESTUDY;

권한이 부여되었습니다.

SQL>
```

이렇게 완성된 룰을 GRANT 명령어로 사용자에게 부여할 수 있습니다. 다음과 같이 앞에서 생성한 ORCLSTUDY 사용자에게 ROLESTUDY 를을 적용할 수 있습니다.

### 실습 15-15 ORCLSTUDY 사용자에게 룰(ROLESTUDY) 부여하기

01 GRANT ROLESTUDY TO ORCLSTUDY;

:: 결과 화면

```
SQL> GRANT ROLESTUDY TO ORCLSTUDY;
권한이 부여되었습니다.

SQL>
```

## 부여된 룰과 권한 확인

ORCLSTUDY 사용자에 현재 부여된 권한과 룰을 확인하려면 USER\_SYS\_PRIVS, USER\_

ROLE\_PRIVS 데이터 사전을 사용하면 됩니다. 데이터 관리 권한을 가진 계정은 DBA\_SYS\_PRIVS, DBA\_ROLE\_PRIVS를 사용해도 됩니다.

#### 실습 15-16 ORCLSTUDY에 부여된 룰과 권한 확인하기

```
01  CONN ORCLSTUDY/ORACLE  
02  SELECT * FROM USER_SYS_PRIVS;  
03  SELECT * FROM USER_ROLE_PRIVS;
```

:: 결과 화면

```
SQL> CONN ORCLSTUDY/ORACLE
```

연결되었습니다.

```
SQL> SELECT * FROM USER_SYS_PRIVS;
```

USERNAME	PRIILEGE	ADM
ORCLSTUDY	CREATE SESSION	NO

```
SQL> SELECT * FROM USER_ROLE_PRIVS;
```

USERNAME	GRANTED_ROLE	ADM	DEF	OS_
ORCLSTUDY	ROLESTUDY	NO	YES	NO

```
SQL>
```

② DBA\_ROLE\_PRIVS와 DBA\_SYS\_PRIVS 데이터 사전을 조회하려면 [WHERE GRANTEE = 'ORCLSTUDY'] 조건을 사용하세요.

#### 부여된 룰 취소

GRANT 명령어로 부여한 ROLE을 취소할 때 REVOKE 문을 사용합니다.

```
SQL> CONN SYSTEM/oracle  
연결되었습니다.  
SQL> REVOKE ROLESTUDY FROM ORCLSTUDY;
```

권한이 취소되었습니다.

```
SQL>
```

#### 룰 삭제

룰 삭제는 DROP 명령어를 사용합니다. 룰을 삭제하면 해당 룰을 부여받은 모든 사용자의 룰이 취소(REVOKE)됩니다.

```
SQL> DROP RULE ROLESTUDY;  
룰이 삭제되었습니다.  
SQL>
```

이 장에서는 오라클의 사용자, 권한, 룰의 생성 및 기본 관리 방법을 알아보았습니다. 앞에서 살펴본 SQL문과 달리 생소한 내용이 많아 다소 난해한 느낌을 많이 받았을 텐데요. 데이터베이스 관리 업무를 담당하지 않는 이상 이 장의 내용은 업무에서 사용해야 하는 경우가 그리 많지 않습니다. 이 책에서 소개하고 있는 권한 관리의 기본 내용을 숙지하고 이후 좀 더 깊이 있는 내용이 필요할 때 공식 문서 등의 자료를 참고해 주세요.

**Q1** 다음 조건을 만족하는 SQL문을 작성해 보세요.

- ① SYSTEM 계정으로 접속하여 PREV\_HW 계정을 생성해 보세요.
- ② 비밀번호는 ORCL로 지정합니다. 접속 권한을 부여하고 PREV\_HW 계정으로 접속이 잘되는지 확인해 보세요.

```
C:\Users\easy_spublishing>SQLPLUS PREV_HW/ORCL
SQL*Plus: Release 11.2.0.1.0 Production on 월 7월 9 04:02:29 2018
Copyright (c) 1982, 2010, Oracle. All rights reserved.

다음에 접속됨:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL>
```

**Q2** SCOTT 계정으로 접속하여 위에서 생성한 PREV\_HW 계정에 SCOTT 소유의 EMP, DEPT, SALGRADE 테이블에 SELECT 권한을 부여하는 SQL문을 작성해 보세요. 권한을 부여했으면 PREV\_HW 계정으로 SCOTT의 EMP, DEPT, SALGRADE 테이블이 잘 조회되는지 확인해 보세요.

**Q3** SCOTT 계정으로 접속하여 PREV\_HW 계정에 SALGRADE 테이블의 SELECT 권한을 취소하는 SQL문을 작성해 보세요. 권한의 변경이 완료되면 다음과 같이 PREV\_HW 계정으로 SALGRADE 테이블의 조회 여부를 확인해 봅시다.

```
C:\Users\easy_spublishing>SQLPLUS PREV_HW/ORCL
SQL*Plus: Release 11.2.0.1.0 Production on 월 7월 9 04:02:29 2018
Copyright (c) 1982, 2010, Oracle. All rights reserved.

다음에 접속됨:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
SQL> SELECT * FROM SCOTT.SALGRADE;
SELECT * FROM SCOTT.SALGRADE
*
1행에 오류:
ORA-00942: 테이블 또는 뷰가 존재하지 않습니다
SQL>
```

정답 이지스퍼블리싱 홈페이지에서 확인하세요.