



## 18장 데이터베이스 롤 권한 제어

오라클에서는 간편하게 권한을 부여할 수 있는 방법으로 롤을 제공하는데 이번 장에서는 롤에 대해서 학습하겠습니다.

# 이 장에서 다룰 내용



## 1 물이란

---

## 2 물의 종류

---

## 3 사용자 물 정의

---

## 4 물 회수하기

---

## 5 물의 장점

---

## 01. 롤이란



- ❖ 롤은 사용자에게 보다 효율적으로 권한을 부여할 수 있도록 여러 개의 권한을 묶어 놓은 것이라고 생각하면 됩니다.
- ❖ 사용자를 생성했으면 그 사용자에게 각종 권한을 부여해야만 생성된 사용자가 데이터베이스를 사용할 수 있습니다.
- ❖ 데이터베이스의 접속 권한(CREATE SESSION), 테이블 생성 권한(CREATE TABLE), 테이블 수정(UPDATE), 삭제(DELETE), 조회(SELECT) 등과 같은 권한은 사용자에게 기본적으로 필요한 권한들인데 사용자를 생성할 때마다 일일이 이러한 권한을 부여하는 것은 번거롭습니다.
- ❖ 이 때문에 다수의 사용자에게 공통적으로 필요한 권한들을 롤에 하나의 그룹으로 묶어두고 사용자에게는 특정 롤에 대한 권한 부여를 함으로서 간단하게 권한 부여를 할 수 있도록 합니다.

## 01. 롤이란

- ❖ 또한 여러 사용자에게 부여된 권한을 수정하고 싶을 때에도 일일이 사용자마다 권한을 수정하지 않고 롤만 수정하면 그 롤에 대한 권한 부여를 한 사용자들의 권한이 자동 수정됩니다. 이 밖에 롤을 활성화 비활성화 함으로서 일시적으로 권한을 부여했다 철회할 수 있으므로 사용자 관리를 간편하고 효율적으로 할 수 있습니다.



## 02. 롤의 종류

- ❖ 롤은 오라클 데이터베이스를 설치하면 기본적으로 제공되는 사전 정의된 롤과 사용자가 정의한 롤로 구분됩니다.
- ❖ 사용자가 직접 롤을 정의하는 방법은 복잡하므로 사전에 정의된 롤부터 살펴보도록 합시다.



## 2.1 사전 정의된 롤의 종류



❖ 다음과 사전 정의된 시스템에서 제공해주는 롤입니다.

### ❖ CONNECT 롤

- 사용자가 데이터베이스에 접속 가능하도록 하기 위해서 다음과 같이 가장 기본적인 시스템 권한 8가지를 묶어 놓았습니다.

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

### ❖ RESOURCE 롤

- 사용자가 객체(테이블, 뷰, 인덱스)를 생성할 수 있도록 하기 위해서 시스템 권한을 묶어 놓았습니다.

CREATE CLUSTER, CREATE PROCEDURE, CREATE SEQUENCE,  
CREATE TABLE, CREATE TRIGGER

### ❖ DBA 롤

- 사용자들이 소유한 데이터베이스객체를 관리하고 사용자들을 작성하고 변경하고 제거할 수 있도록 하는 모든 권한을 가집니다. 즉, 시스템 자원을 무제한적으로 사용하며 시스템 관리에 필요한 모든 권한을 부여할 수 있는 강력한 권한을 보유한 롤입니다.

## <실습하기> 롤 부여하기

일반적으로 데이터베이스 관리자는 새로운 사용자를 생성할 때 CONNECT 롤과 RESOURCE 롤을 부여합니다. USER04 사용자를 생성하여 CONNECT 롤과 RESOURCE 롤을 부여합니다.

1. 우선 데이터베이스 관리자로 접속합니다.

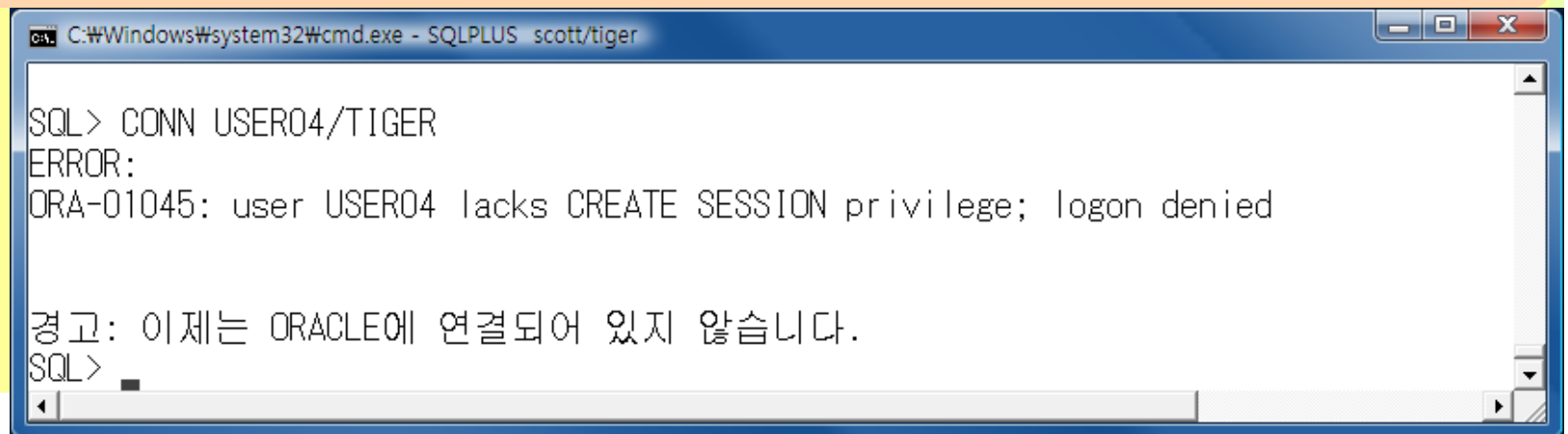
**CONN system/manager**

2. 새롭게 사용자를 생성합니다.

**CREATE USER USER04 IDENTIFIED BY TIGER;**

3. 생성된 생성자로 로그인을 시도합니다.

**CONN USER04/TIGER**



```
C:\Windows\system32\cmd.exe - SQLPLUS scott/tiger

SQL> CONN USER04/TIGER
ERROR:
ORA-01045: user USER04 lacks CREATE SESSION privilege; logon denied

경고: 이제는 ORACLE에 연결되어 있지 않습니다.
SQL>
```

## 〈실습하기〉 롤 부여하기

새롭게 생성된 사용자에는 데이터베이스의 접속 권한인 CREATE SESSION 권한이 부여 되지 않았으므로 로그인에 실패합니다.

4. 새로운 사용자에게 권한 부여를 하기 위해서는 다시 데이터베이스 관리자로 접속해야 합니다.

### **CONN system/manager**

5. 데이터베이스 관리자로 접속했으면 이제 CONNECT롤과 RESOURCE롤을 부여합니다.

### **GRANT CONNECT, RESOURCE TO USER04;**

6. 권한이 부여가 되었으면 다시 USER04 사용자로 로그인 시도해봅시다.

### **CONN USER04/TIGER**

CONNECT롤에 데이터베이스의 접속 권한인 CREATE SESSION 권한이 포함되어 있으므로 로그인에 성공합니다.



## 2.2 롤 관련 데이터 디렉터리



- ❖ 다음은 사용자에게 부여된 롤을 확인해 보겠습니다. 롤을 확인하기 위한 데이터 디렉터리는 무수히 많습니다.

```
SELECT * FROM DICT WHERE TABLE_NAME LIKE '%ROLE%';
```

- ❖ 위 조회 결과로 얻어진 데이터 디렉터리를 통해서 부여된 권한에 대한 정보를 확인할 수 있습니다. 다음은 롤 관련 데이터 디렉터리를 정리한 표입니다.

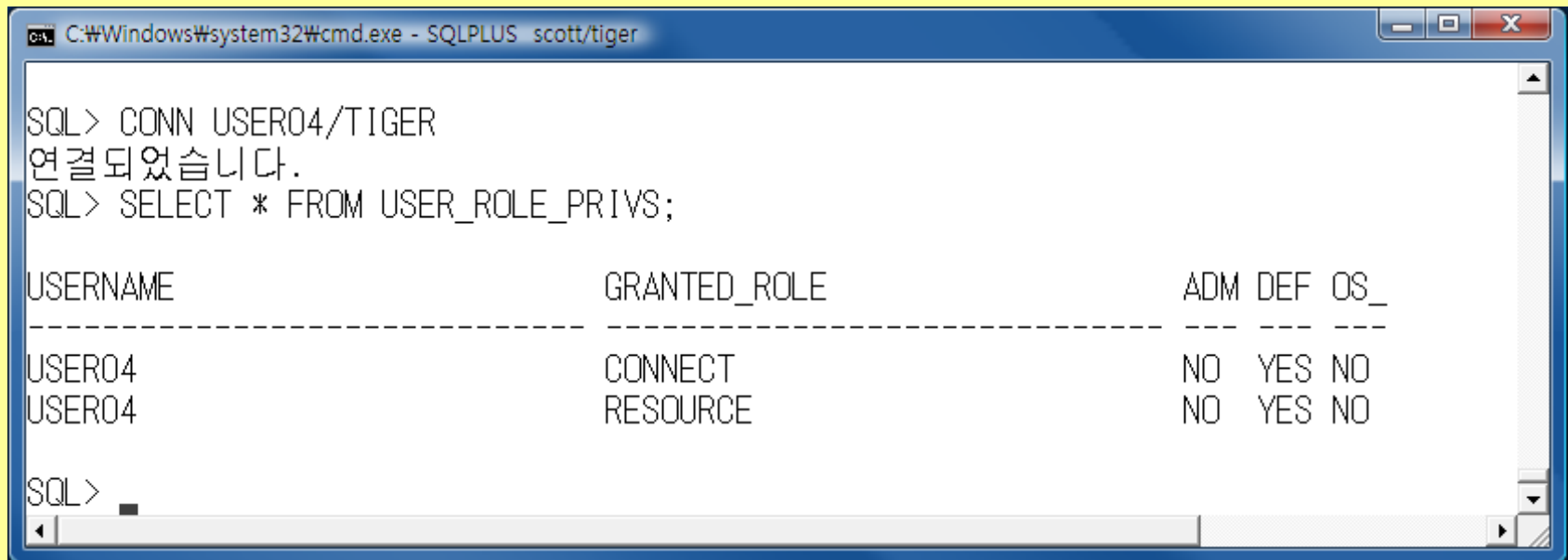
디렉터리 명	설 명
ORLE_SYS_PRIVS	롤에 부여된 시스템 권한 정보
ROLE_TAB_PRIVS	롤에 부여된 테이블 관련 권한 정보
USER_ROLE_PRIVS	접근 가능한 롤 정보
USER_TAB_PRIVS_MADE	해당 사용자 소유의 오브젝트에 대한 오브젝트 권한 정보
USER_TAB_PRIVS_RECD	사용자에게 부여된 오브젝트 권한 정보
USER_COL_PRIVS_MADE	사용자 소유의 오브젝트 중 칼럼에 부여된 오브젝트 권한 정보
USER_COL_PRIVS_REDC	사용자에게 부여된 특정 칼럼에 대한 오브젝트 권한 정보

## 〈실습하기〉 롤을 확인하기

롤 관련 데이터 덱서너리 중에서 현재 사용자에게 부여된 롤을 확인하기 위한 데이터 덱서너리는 USER\_ROLE\_PRIVS 입니다. USER04로 로그인 하였으므로 다음과 같이 입력하면 사용자 USER04에 부여된 롤에 대한 정보를 확인할 수 있습니다.

**CONN USER04/TIGER**

**SELECT \* FROM USER\_ROLE\_PRIVS;**



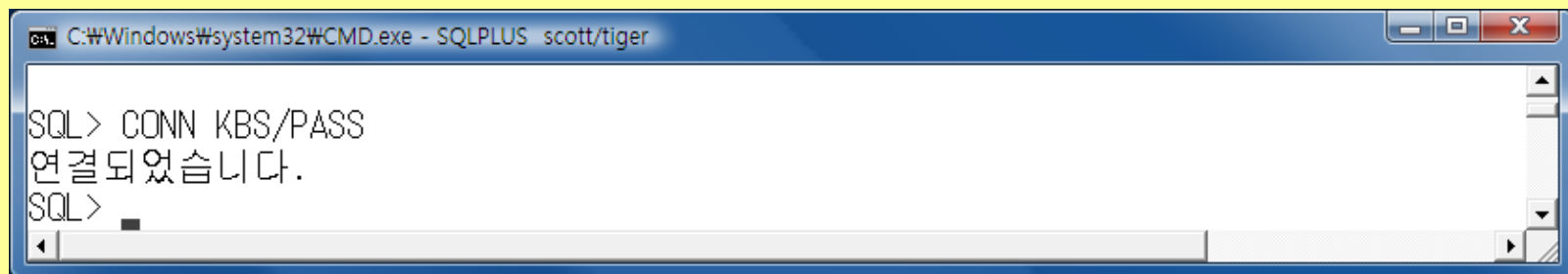
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - SQLPLUS scott/tiger". The user has entered the command "CONN USER04/TIGER" and received the message "연결되었습니다." (Connected). Then, the user entered "SELECT \* FROM USER\_ROLE\_PRIVS;" and the following table of results was displayed:

USERNAME	GRANTED_ROLE	ADM	DEF	OS_
USER04	CONNECT	NO	YES	NO
USER04	RESOURCE	NO	YES	NO

The prompt "SQL>" is visible at the bottom of the window.

## <탄탄히 다지기>

1. KBS 라는 사용자를 생성(암호는 PASS)하고 기본적인 권한 부여를 하지 않으면 데이터베이스에 로그인 불가능하므로 CONNECT 과 RESOURCE 권한을 KBS 사용자에게 부여하시오.



```
C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger

SQL> CONN KBS/PASS
연결되었습니다.
SQL>
```

### 03. 사용자 롤 정의



- ❖ CONNECT, RESOURCE 롤과 같은 기본적으로 제공되는 사전 정의된 롤을 사용자에게 부여해 보았습니다.
- ❖ 이번에는 사용자가 정의해서 사용하는 롤에 대해 살펴보겠습니다. 사용자는 CREATE ROLE 명령어로 다음 형식에 따라 롤을 생성해야 합니다.

```
CREATE ROLE role_name;  
GRANT privilege_name TO role_name;
```

### 03. 사용자 롤 정의



- ❖ 새로 생성된 USER04로 로그인해서 SCOTT 사용자의 EMP 테이블에 접근해 보도록 합시다.

```
CONN USER04/TIGER;  
SELECT * FROM SCOTT.EMP;
```

```
C:\Windows\system32\cmd.exe - SQLPLUS scott/tiger  
  
SQL> CONN USER04/TIGER;  
연결되었습니다.  
SQL> SELECT * FROM SCOTT.EMP;  
SELECT * FROM SCOTT.EMP  
*  
1행에 오류:  
ORA-00942: 테이블 또는 뷰가 존재하지 않습니다  
  
SQL>
```

- ❖ 사용자 USER04는 SCOTT 사용자 소속인 EMP 테이블 객체를 조회할 수 없습니다. USER04는 SCOTT 사용자 소속인 EMP 테이블 객체를 조회할 수 있도록 권한을 부여 해야만 합니다.

### 03. 사용자 롤 정의



- ❖ 이번에는 사용자 USER04에 객체 권한을 직접 부여하지 않고 롤을 이용해 보도록 하겠습니다.
- ❖ 다음과 같은 순서로 작업을 진행할 것입니다.

① 롤을 생성한다.(데이터베이스 관리자)

② 롤에 권한 부여한다.  
(데이터베이스 관리자 혹은 특정 사용자)

③ 사용자에게 롤을 부여한다.(데이터베이스 관리자)

## 03. 사용자 롤 정의



- ① 롤을 생성하기 위한 DBA에서 이루어집니다.
  - **CREATE ROLE ROLE\_NAME;**
- ② 롤에 부여할 권한의 종류에 따라서 DBA에서 부여할 수도 있고, 객체를 소유한 사용자로 접속한 후 부여해야 합니다.
- ❖ 다음과 같이 시스템 권한 일 경우에는 DBA에서 이루어집니다.
  - **GRANT CREATE SESSION, CREATE TABLE, CREATE VIEW TO ROLE\_NAME;**
- ❖ 다음과 같이 객체 권한일 경우에는 특정 객체로 접근해서 부여해야 합니다.
  - **GRANT OBJECT\_PRIV TO ROLE\_NAME;**
- ③ 사용자에게 롤을 부여하는 작업 역시 DBA에서 이루어집니다.
  - **GRANT ROLE\_NAME TO USER\_NAME;**

## 〈실습하기〉 롤 생성하여 시스템 권한 할당하기

대략적인 순서는 위와 같습니다. 롤을 생성하여 할당하는 본격적인 실습을 하도록 합시다.

1. 롤을 생성할 수 있는 사용자는 반드시 DBA 권한이 있는 사용자여야만 하기에 롤을 생성하기 앞서서 반드시 DBA 권한을 가진 사용자로 접속해야만 합니다.

```
CONN system/manager  
CREATE ROLE MROLE;
```

2. 생성된 롤에게 권한을 부여합니다.

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

3. 사용자를 생성하여 롤을 부여합니다.

```
CREATE USER USER05 IDENTIFIED BY TIGER;  
GRANT MROLE TO USER05;
```



## 〈실습하기〉 롤 생성하여 객체 권한 할당하기

롤을 생성하여 할당하는 객체 권한을 할당해 봅시다.

1. 롤을 생성할 수 있는 사용자는 반드시 DBA 권한이 있는 사용자여야만 하기에 롤을 생성하기 앞서서 반드시 DBA 권한을 가진 사용자로 접속해야만 합니다. CREATE ROLE 명령문을 사용하여 MROLE02 를 생성합니다.

```
CONN system/manager  
CREATE ROLE MROLE02;
```

2. 생성한 롤에 권한 부여를 하기 위해서 EMP 테이블 객체를 소유하고 있는 SCOTT 사용자로 로그인 합니다. MRLOE02 에게 EMP 테이블 객체를 조회할 수 있도록 SELECT 권한을 줍시다.

```
CONN scott/tiger  
GRANT SELECT ON EMP TO MROLE02;
```

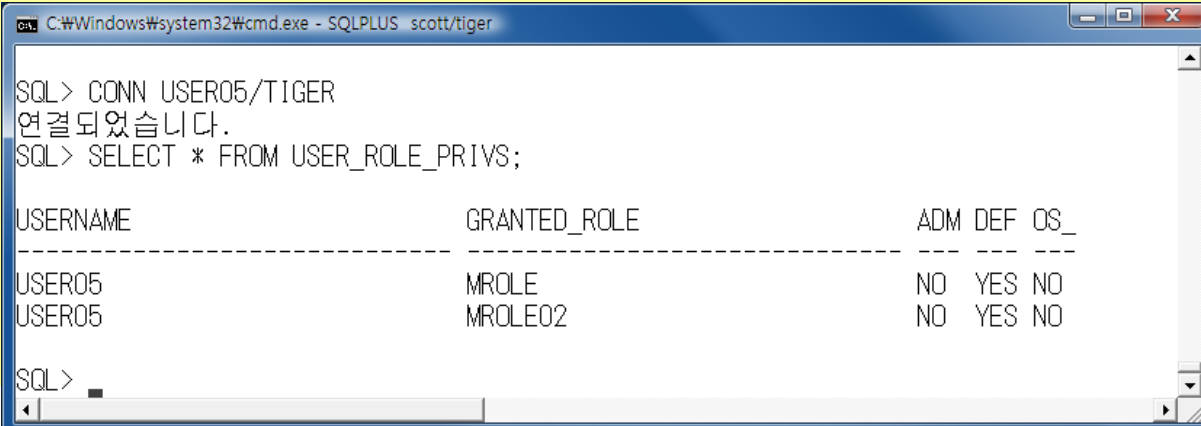
## 〈실습하기〉 롤 생성하여 객체 권한 할당하기

3. MRLOE02 에 EMP 테이블 객체를 조회할 수 있도록 SELECT 권한을 주었습니다. 생성된 롤을 사용자에게 부여하기 위해서 다시 데이터베이스 관리자로 로그인하여 사용자 USER05 에게 롤에 대한 권한 부여 합니다.

```
CONN system/manager
GRANT MROLE02 TO USER05;
```

4. 사용자 USER05 에게 롤에 대한 권한 부여를 마쳤으면 사용자 USER05 로 로그인하여 롤에 대한 권한이 부여되었는지 확인해봅시다.

```
CONN USER05/TIGER
SELECT * FROM USER_ROLE_PRIVS;
```



```
C:\Windows\system32\cmd.exe - SQLPLUS scott/tiger

SQL> CONN USER05/TIGER
연결되었습니다.
SQL> SELECT * FROM USER_ROLE_PRIVS;

USERNAME          GRANTED_ROLE          ADM DEF OS_
-----
USER05            MROLE                 NO  YES NO
USER05            MROLE02              NO  YES NO

SQL>
```

## <실습하기> WITH ADMIN OPTION을 지정하지 않고 권한 부여

WITH ADMIN OPTION을 지정하지 않으면 부여 받은 권한을 다른 사용자에게 부여할 수 없음을 확인해 봅시다.

1. 사용자명은 USER03 암호는 TIGER로 사용자를 생성해봅시다. 사용자를 생성하기 위해서는 CREATE USER 명령어를 사용합니다.
2. USER03에게 WITH ADMIN OPTION을 지정하지 않고 CREATE SESSION 권한을 부여합니다.

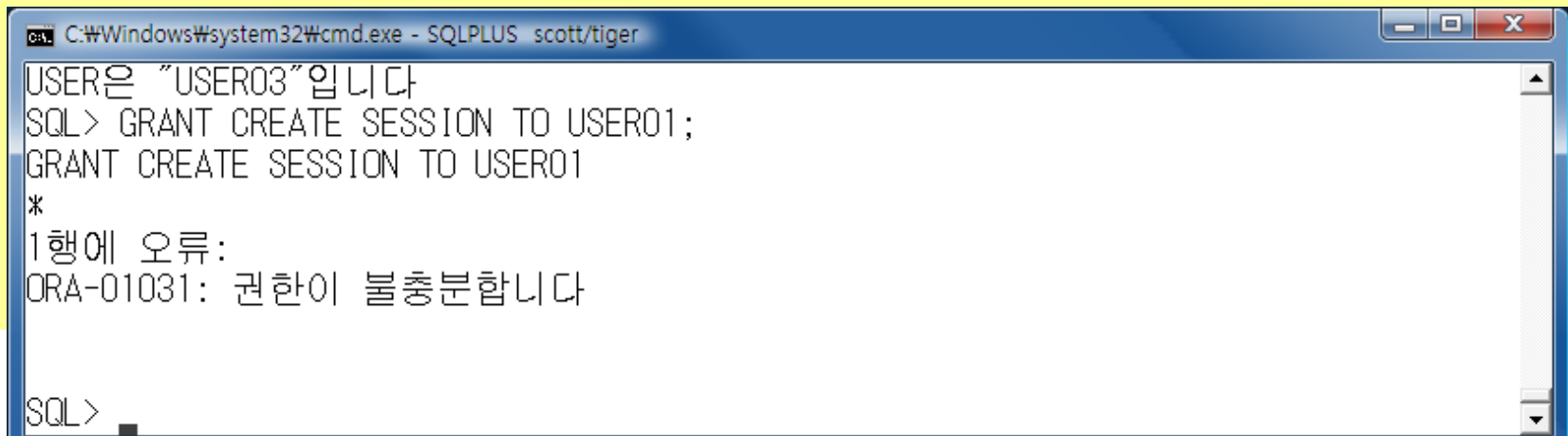
**GRANT CREATE SESSION TO USER03;**

3. USER03 사용자로 접속합니다.

**CONN USER03/TIGER;**

4. USER03 사용자는 자기가 받은 권한을 다른 사용자에게 부여할 수 없습니다

**GRANT CREATE SESSION TO USER01;**



```
C:\Windows\system32\cmd.exe - SQLPLUS scott/tiger
USER은 "USER03"입니다
SQL> GRANT CREATE SESSION TO USER01;
GRANT CREATE SESSION TO USER01
*
1행에 오류:
ORA-01031: 권한이 불충분합니다
SQL>
```

## 04. 롤 회수하기



- ❖ 롤 역시 권한처럼 사용하지 않게 되었을 경우 이를 회수할 수 있습니다. 다음은 롤을 회수하기 위한 DROP ROLE 명령어의 형식입니다.

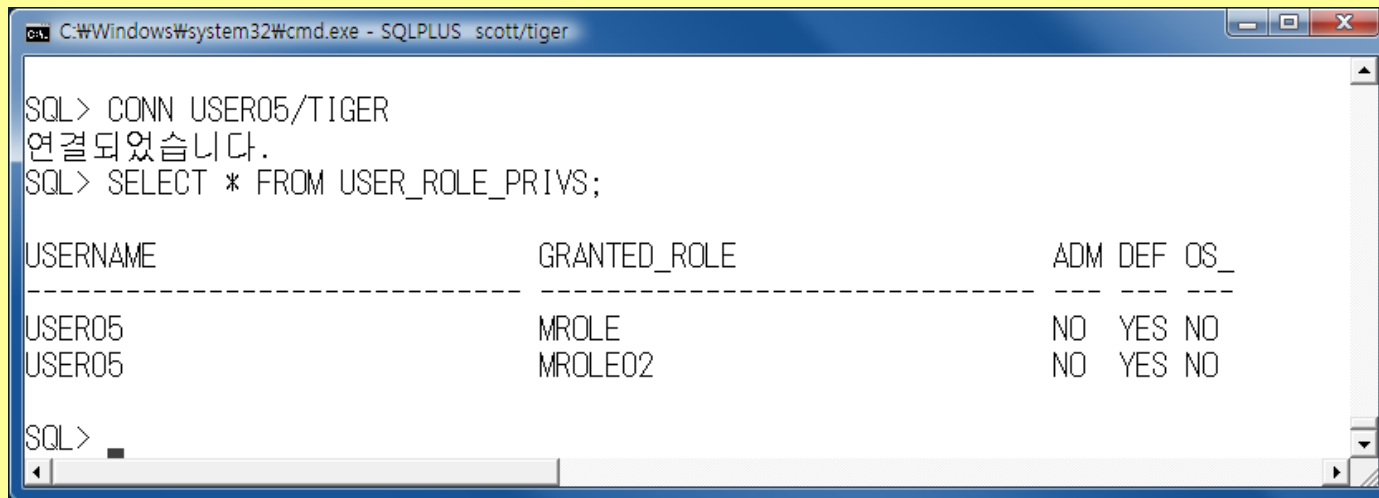
```
DROP ROLE role_name FROM user_name;
```

## <실습하기> 롤 회수하기

이번에는 REVOKE 문을 사용하여 롤을 회수해 보도록 합시다.

1. 현재 사용자에게 부여된 롤 권한을 확인하기 위해서 다음과 같이 명령문을 수행해 보도록 하겠습니다.

```
CONN USER05/TIGER
SELECT * FROM USER_ROLE_PRIVS;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - SQLPLUS scott/tiger". The user has entered the following commands:

```
SQL> CONN USER05/TIGER
연결되었습니다.
SQL> SELECT * FROM USER_ROLE_PRIVS;
```

The output of the query is displayed as follows:

USERNAME	GRANTED_ROLE	ADM	DEF	OS_
USER05	MROLE	NO	YES	NO
USER05	MROLE02	NO	YES	NO

The prompt "SQL>" is visible at the bottom of the window.


## 〈실습하기〉 롤 회수하기

## 2. 데이터베이스 관리자로 접속한 후에 롤을 회수합니다.

CONN system/manager  
REVOKE MROLE FROM USER05;

3. 다시 USER05로 접속하여 USER05에 부여된 롤을 확인해보면 MROLE 롤이 회수된 것을 확인할 수 있습니다.

```
CONN USER05/TIGER
SELECT * FROM USER_ROLE_PRIVS;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger". The prompt is "SQL>". The user has entered the command "CONN USER05/TIGER", and the response is "연결되었습니다." (Connected.). The user has then entered "SELECT \* FROM USER\_ROLE\_PRIVS;", and the output is displayed as follows:

USERNAME	GRANTED_ROLE	ADM	DEF	OS_
USER05	MROLE	NO	YES	NO

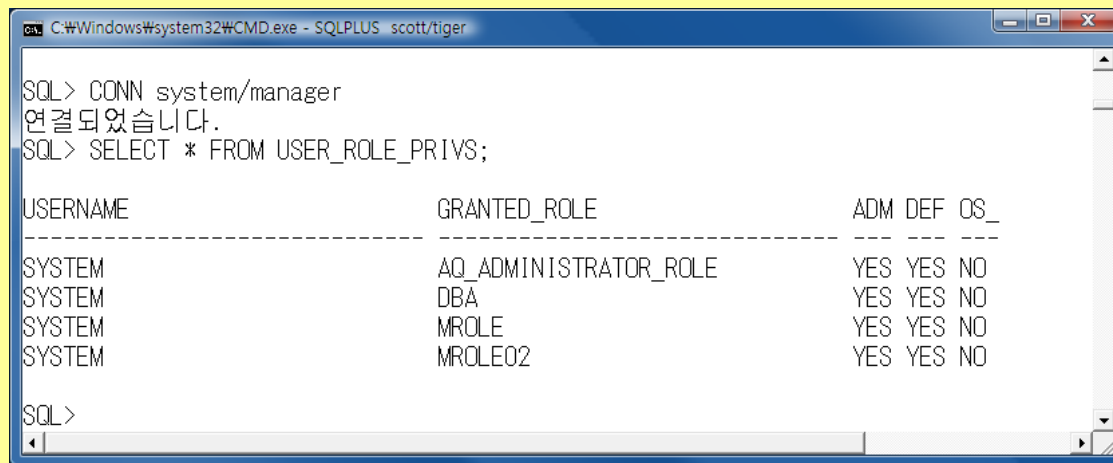
The prompt "SQL>" is visible at the bottom of the window.

## <실습하기> 롤 제거하기

1. 사용자 USER05 에게 부여되었던 롤에 대한 권한만을 회수할 뿐, 롤 MROLE02 은 아직 존재합니다.

SYSTEM 계정에서 롤을 생성하였으므로 SYSTEM 계정으로 접속하여 데이터 디렉터리 USER\_ROLE\_PRIVS의 내용을 출력하여 MROLE02이 존재함을 확인합니다.

```
CONN system/manager
SELECT * FROM USER_ROLE_PRIVS;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger". The user has entered the command "CONN system/manager" and received the message "연결되었습니다." (Connected). Then, the user entered "SELECT \* FROM USER\_ROLE\_PRIVS;" and the following table of results was displayed:

USERNAME	GRANTED_ROLE	ADM	DEF	OS_
SYSTEM	AQ_ADMINISTRATOR_ROLE	YES	YES	NO
SYSTEM	DBA	YES	YES	NO
SYSTEM	MROLE	YES	YES	NO
SYSTEM	MROLE02	YES	YES	NO

The prompt "SQL>" is visible at the bottom of the window.

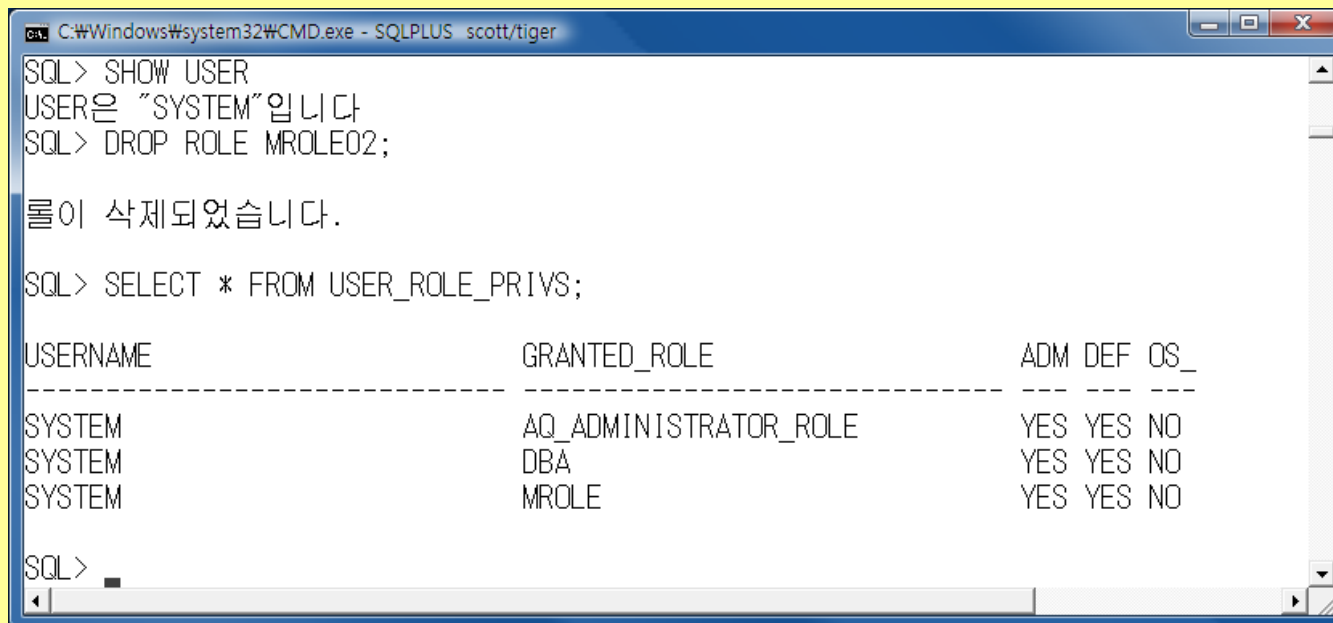
## <실습하기> 롤 제거하기

2. MROLE02을 제거해봅시다. MROLE을 제거한 후 USER\_ROLE\_PRIVS 데이터 디렉터리를 살펴보면 MROLE02이 나타나지 않습니다.

**SHOW USER**

**DROP ROLE MROLE02;**

**SELECT \* FROM USER\_ROLE\_PRIVS;**



```
C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger

SQL> SHOW USER
USER은 "SYSTEM"입니다
SQL> DROP ROLE MROLE02;

롤이 삭제되었습니다.

SQL> SELECT * FROM USER_ROLE_PRIVS;


```

USERNAME	GRANTED_ROLE	ADM	DEF	OS_
SYSTEM	AQ_ADMINISTRATOR_ROLE	YES	YES	NO
SYSTEM	DBA	YES	YES	NO
SYSTEM	MROLE	YES	YES	NO

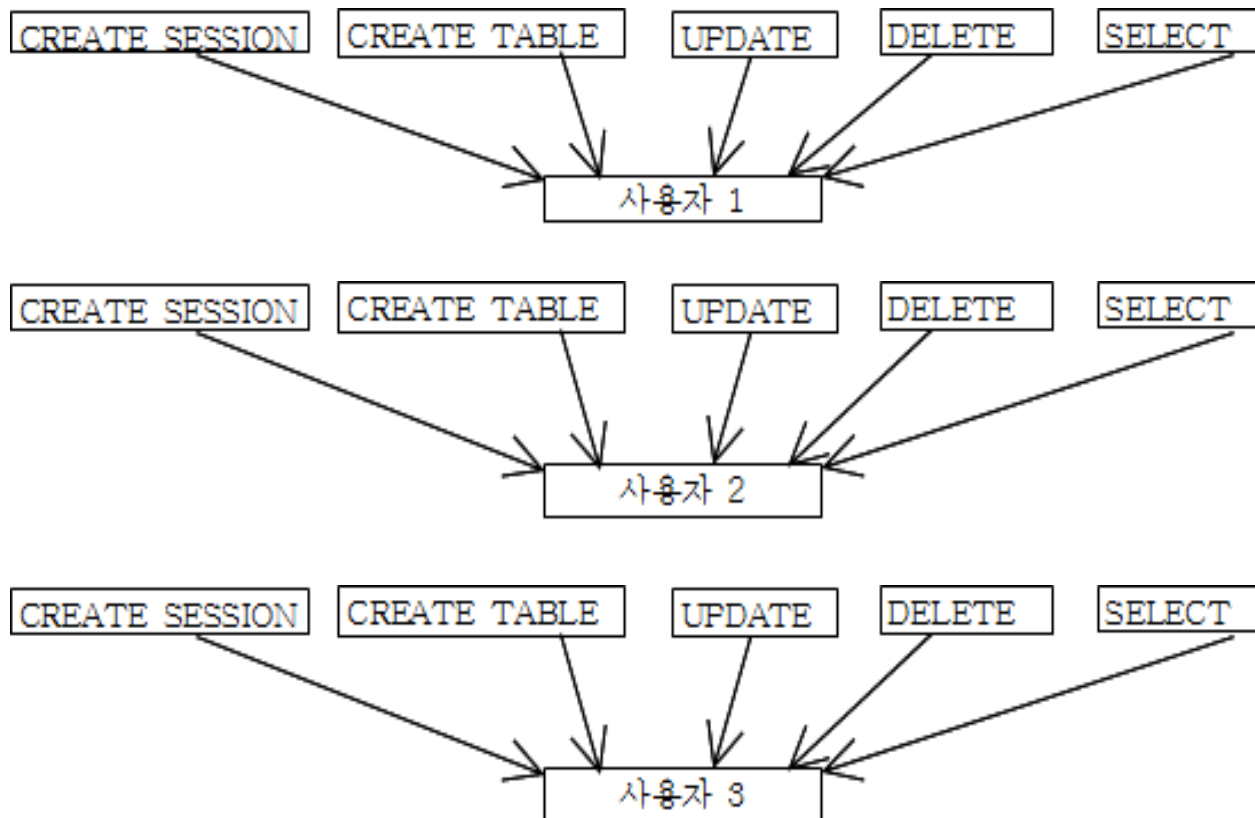
```
SQL>
```



## 05. 룰의 장점



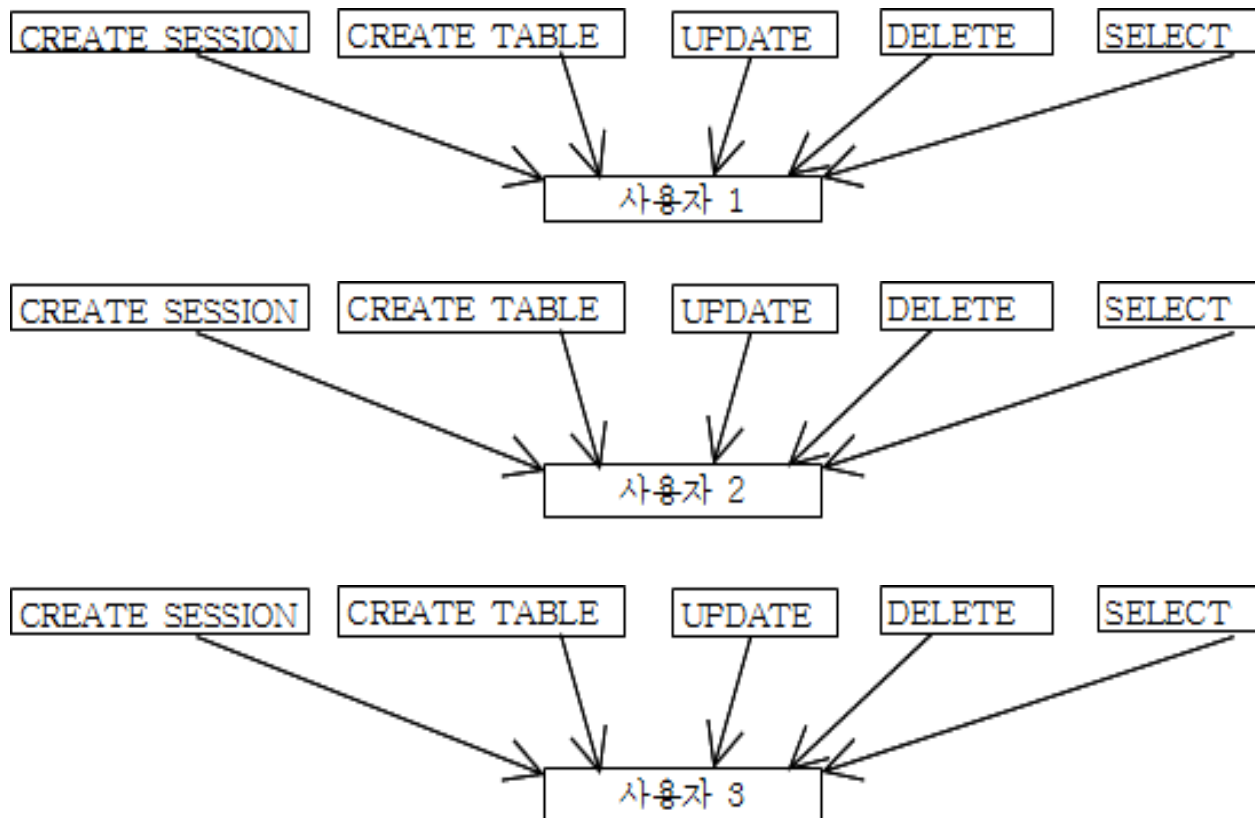
- ❖ 이번에는 룰을 사용하여 권한 부여함으로써 생기는 장점에 대해서 살펴보겠습니다. 시스템권한이나 객체 권한을 사용자마다 일일이 부여하게 되면 번거롭습니다.



## 05. 룰의 장점



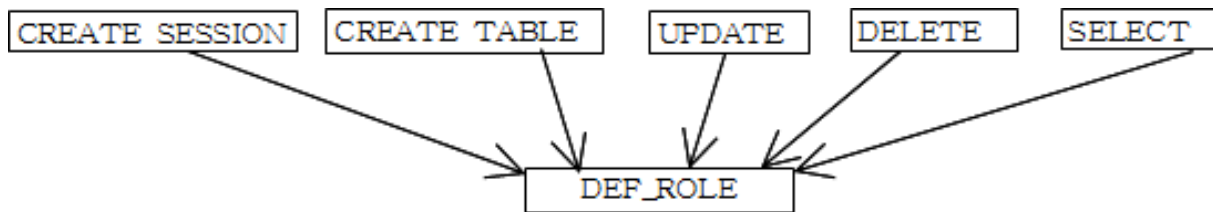
- ❖ 이번에는 룰을 사용하여 권한 부여함으로써 생기는 장점에 대해서 살펴보겠습니다. 시스템권한이나 객체 권한을 사용자마다 일일이 부여하게 되면 번거롭습니다.



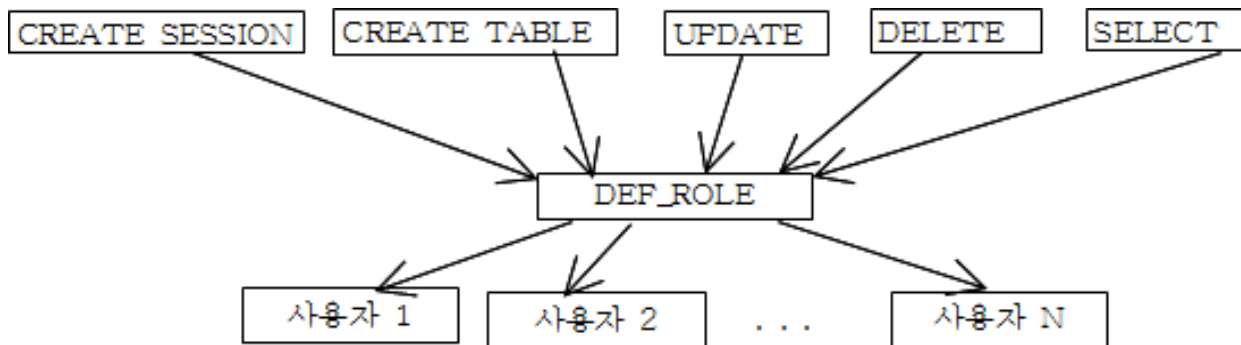
## 05. 롤의 장점



- ❖ 3명의 사용자에게 일일이 권한 부여하는 명령어를 부여하려면 굉장히 번거롭습니다.
- ❖ 이러한 단점을 롤로 보안할 수 있습니다, 우선 롤에 시스템 권한과 객체 권한을 부여합니다.



- ❖ 그런 후에 롤을 사용자에게 대해 권한 부여함으로써 작업을 간소화할 수 있습니다.



## 〈실습하기〉 디폴트 롤을 생성하여 여러 사용자에게 부여하기

롤에 시스템 권한과 객체 권한을 부여한 후에 롤을 사용자에게 대해 권한 부여함으로써 작업을 간소화해 봅시다.

1. 데이터베이스 관리자로 접속하여 롤(DEF\_ROLE)을 생성합니다.

```
CONN system/manager  
CREATE ROLE DEF_ROLE;
```

2. 생성된 롤 DEF\_ROLE에 시스템 권한인 CREATE SESSION과 CREATE TABLE을 부여합니다.

```
GRANT CREATE SESSION TO DEF_ROLE;  
GRANT CREATE TABLE TO DEF_ROLE;
```

3. 생성된 롤 DEF\_ROLE에 SCOTT 사용자로 접속해서 EMP 테이블을 수정, 삭제, 조회할 수 있도록 객체 권한을 부여합니다.

```
CONN scott/tiger  
GRANT UPDATE ON EMP TO DEF_ROLE;  
GRANT DELETE ON EMP TO DEF_ROLE;  
GRANT SELECT ON EMP TO DEF_ROLE;
```

## 〈실습하기〉 디폴트 롤을 생성하여 여러 사용자에게 부여하기

4. 데이터베이스 관리자인 SYSTEM으로 접속해서 사용자 계정을 3개 만듭니다

```
CONN system/manager  
CREATE USER USERA1 IDENTIFIED BY A1234;  
CREATE USER USERA2 IDENTIFIED BY A1234;  
CREATE USER USERA3 IDENTIFIED BY A1234;
```

5. 생성된 사용자 계정에 각각 DEF\_ROLE에 대한 권한 설정을 합니다.

```
SHOW USER  
GRANT DEF_ROLE TO USERA1;  
GRANT DEF_ROLE TO USERA2;  
GRANT DEF_ROLE TO USERA3;
```

## 〈실습하기〉 디폴트 롤을 생성하여 여러 사용자에게 부여하기

6. ROLE\_SYS\_PRIVS은 시스템 권한에 대한 정보를 저장한 데이터 디렉터리이고, ROLE\_TAB\_PRIVS은 객체 권한에 대한 정보를 저장한 데이터 디렉터리입니다. 두 데이터 디렉터리의 내용을 출력함으로써 롤에 권한 설정이 제대로 되어 있는지 확인해 봅시다.

**SHOW USER**

```
SELECT * FROM ROLE_SYS_PRIVS WHERE  
ROLE='DEF_ROLE';
```

```
SELECT * FROM ROLE_TAB_PRIVS WHERE  
ROLE='DEF_ROLE';
```

이렇게 자주 사용되는 권한을 롤에 부여해 놓으면 언제 어느 때 새로운 사용자가 생기더라도 쉽게 권한 부여를 할 수 있게 됩니다.



# Thank You !

Dynamic\_오라클 11g + PL/SQL 입문 18장