

## DCL = 권한 관리 SQL

핵심 명령어: GRANT, REVOKE, SHOW GRANTS

주로 DBA(관리자)가 사용자별 접근 권한을 관리할 때 사용

# 20문제에 대한 DCL SQL 예시

---

### reader1

1. reader1 계정을 만들고 naver\_db 전체 테이블에 SELECT 권한만 부여하세요.

```
CREATE USER 'reader1'@'%' IDENTIFIED BY 'pw1234';  
GRANT SELECT ON naver_db.* TO 'reader1'@'%';
```

2. reader1 계정에 부여했던 naver\_db 권한을 모두 회수하세요.

```
REVOKE ALL PRIVILEGES ON naver_db.* FROM 'reader1'@'%';
```

---

### writer1

3. writer1 계정을 만들고 naver\_db.member 테이블에 INSERT, UPDATE 권한을 부여하세요.

```
CREATE USER 'writer1'@'%' IDENTIFIED BY 'pw1234';  
GRANT INSERT, UPDATE ON naver_db.member TO 'writer1'@'%';
```

4. writer1 계정에서 UPDATE 권한만 회수하고, INSERT만 유지되도록 하세요.

```
REVOKE UPDATE ON naver_db.member FROM 'writer1'@'%';
```

---

### buy\_cleaner

5. `buy_cleaner` 계정을 만들고 `naver_db.buy` 테이블에 DELETE 권한만 부여하세요.

```
CREATE USER 'buy_cleaner'@'%' IDENTIFIED BY 'pw1234';  
GRANT DELETE ON naver_db.buy TO 'buy_cleaner'@'%';
```

6. `buy_cleaner` 계정에 부여한 DELETE 권한을 회수하세요.

```
REVOKE DELETE ON naver_db.buy FROM 'buy_cleaner'@'%';
```

---

## **view\_only**

7. `view_only` 계정을 만들고 `naver_db`의 뷰(View)에 대해서만 SELECT 권한을 부여하세요.

```
CREATE USER 'view_only'@'%' IDENTIFIED BY 'pw1234';  
GRANT SELECT ON naver_db.view_name TO 'view_only'@'%'; --  
view_name은 실제 뷰 이름
```

8. `view_only` 계정에 부여한 뷰 접근 권한을 회수하세요.

```
REVOKE SELECT ON naver_db.view_name FROM 'view_only'@'%';
```

---

## **tmp\_user**

9. `tmp_user` 계정을 만들고 `naver_db`에서 CREATE TEMPORARY TABLE 권한을 부여하세요.

```
CREATE USER 'tmp_user'@'%' IDENTIFIED BY 'pw1234';  
GRANT CREATE TEMPORARY TABLES ON naver_db.* TO 'tmp_user'@'%';
```

10. `tmp_user` 계정에 부여한 임시 테이블 생성 권한을 회수하세요.

```
REVOKE CREATE TEMPORARY TABLES ON naver_db.* FROM 'tmp_user'@'%';
```

---

## reader\_univ

11. `reader_univ` 계정을 만들고 `univ_db` 전체 테이블에 `SELECT` 권한만 부여하세요.

```
CREATE USER 'reader_univ'@'%' IDENTIFIED BY 'pw1234';  
GRANT SELECT ON univ_db.* TO 'reader_univ'@'%';
```

12. `reader_univ` 계정의 `univ_db` 접근 권한을 모두 회수하세요.

```
REVOKE ALL PRIVILEGES ON univ_db.* FROM 'reader_univ'@'%';
```

---

## student\_writer

13. `student_writer` 계정을 만들고 `univ_db.student` 테이블에 `INSERT`, `UPDATE` 권한을 부여하세요.

```
CREATE USER 'student_writer'@'%' IDENTIFIED BY 'pw1234';  
GRANT INSERT, UPDATE ON univ_db.student TO 'student_writer'@'%';
```

14. `student_writer` 계정에서 `UPDATE` 권한만 회수하세요.

```
REVOKE UPDATE ON univ_db.student FROM 'student_writer'@'%';
```

---

## student\_course\_admin

15. `student_course_admin` 계정을 만들고 `univ_db.student_course` 테이블에 모든 권한(`ALL`)을 부여하세요.

```
CREATE USER 'student_course_admin'@'%' IDENTIFIED BY 'pw1234';  
GRANT ALL PRIVILEGES ON univ_db.student_course TO  
'student_course_admin'@'%';
```

16. `student_course_admin` 계정에 부여했던 모든 권한을 회수하세요.

```
REVOKE ALL PRIVILEGES ON univ_db.student_course FROM  
'student_course_admin'@'%';
```

---

## **department\_admin**

17. `department_admin` 계정을 만들고 `univ_db.department` 테이블에 모든 권한과 GRANT OPTION을 부여하세요.

```
CREATE USER 'department_admin'@'%' IDENTIFIED BY 'pw1234';  
GRANT ALL PRIVILEGES ON univ_db.department TO 'department_admin'@'%'  
WITH GRANT OPTION;
```

18. `department_admin` 계정에서 GRANT OPTION 권한만 회수하세요.

```
REVOKE GRANT OPTION ON univ_db.department FROM  
'department_admin'@'%';
```

---

## **mixed1**

19. `mixed1` 계정을 만들고 `naver_db`에서는 DML 전체(SELECT, INSERT, UPDATE, DELETE), `univ_db`에서는 SELECT만 부여하세요.

```
CREATE USER 'mixed1'@'%' IDENTIFIED BY 'pw1234';  
GRANT SELECT, INSERT, UPDATE, DELETE ON naver_db.* TO 'mixed1'@'%';  
GRANT SELECT ON univ_db.* TO 'mixed1'@'%';
```

20. `mixed1` 계정에서 `univ_db`의 SELECT 권한을 회수하세요.

```
REVOKE SELECT ON univ_db.* FROM 'mixed1'@'%';
```