

13장. 데이터 제어어(DCL)

DCL은 Data Control Language 의 약자로 데이터 제어어를 말합니다. 데이터 제어어는 데이터베이스에 접근하고 객체들을 사용하도록 권한을 주고 회수하는 등의 역할을 맡고있습니다. DCL에는 COMMIT, ROLLBACK, GRANT, REVOKE명령어가 있습니다.

13.1 COMMIT

COMMIT이란 모든 작업을 정상적으로 처리하겠다고 확정하는 명령어로 트랜잭션 과정을 종료하는 것 입니다. 트랜잭션에 의해 만들어진 모든 변경 사항은 다른 사람에게 표시되며 충돌이 발생하더라도 내구성이 보장됩니다. 또한 처리 과정을 DB에 영구 저장하며 이전 데이터는 완전히 UPDATE됩니다.

```
COMMIT [ WORK | TRANSACTION ] [ AND [ NO ] CHAIN ]
```

13.2 ROLLBACK

ROLLBACK은 작업에서 문제가 발생 했을때 트랜잭션의 처리 과정에서 발생한 변경 사항을 취소하고, 트랜잭션 과정을 종료 시킵니다. 트랜잭션을 ROLLBACK하고 트랜잭션에서 발생한 모든 업데이트를 버립니다. 변경 사항을 취소 할때 마지막 commit을 완료한 시점으로 돌아갑니다. 즉, commit하여 저장한것만 복구합니다.

```
ROLLBACK [ WORK | TRANSACTION ] [ AND [ NO ] CHAIN ]
```

13.3 GRANT

오브젝트가 만들어지면 소유자가 할당됩니다. 일반적으로 소유자는 작성할 명령문을 실행하는 역할입니다. 대부분의 유형의 오브젝트의 경우, 초기 상태에서 소유자(또는 슈퍼유저)만이 해당 오브젝트를 사용할 수 있습니다. 다른 사용자가 이 오브젝트를 사용하려면 권한이 부여되어야 합니다.

다음과 같이 오브젝트에 대한 적절한 유형의 ALTER 명령을 사용하여 오브젝트에 새 소유자를 지정할 수 있습니다.

```
ALTER TABLE table_name OWNER TO new_owner;
```

슈퍼 유저는 언제든지 소유자를 변경할 수 있습니다.

일반 역할은 대상 오브젝트의 현재 소유자(또는 소유자 역할의 구성원)이고 새 소유자 역할의 구성원인 경우에만 소유자를 변경할 수 있습니다.

권한을 할당하려면 **GRANT** 명령을 사용하십시오.

예를 들어, **joe**라는 기존 역할과 **accounts**라는 기존 테이블이 있는 경우, 이 테이블을 갱신할 수 있는 권한을 부여하는 구문은 다음과 같습니다.

```
GRANT UPDATE ON accounts TO joe;
```

시스템의 모든 역할에 권한을 부여하려면 특수한 역할 이름인 **PUBLIC**을 사용할 수 있습니다.

또한 그룹 역할을 사용하면 데이터베이스에 많은 사용자가 있는 경우 권한을 관리 할 수 있습니다.

가능한 권한은 다음과 같습니다.

SELECT

테이블, 뷰, 구체화된 뷰 또는 다른 테이블처럼 보이는 객체 에 대해 **SELECT**가 있는 열 또는 지정된 열(여러 개)에 허용합니다.

또한 **COPY**의 사용을 허용합니다. 이 권한은 **UPDATE** 또는 **DELETE**에서 기존 열을 참조하는 경우에도 필요합니다. 시퀀스에서 이 권한은 **currval** 함수를 사용할 수 있습니다. 대형 오브젝트에서 이 권한은 오브젝트를 조회할 수 있습니다.

INSERT

테이블, 뷰 등에 새 행을 **INSERT**할 수 있습니다. 특정 열만 **INSERT** 명령으로 지정하려는 경우 해당 열에 허용 할 수 있습니다. (따라서 다른 열에는 기본값이 설정됩니다). **COPY FROM**을 이용할 수도 있습니다.

UPDATE

테이블, 뷰 등의 열을 **UPDATE**할 수 있습니다. (실용적으로 쉽지 않은 **UPDATE** 명령에는 **SELECT** 권한이 필요합니다. 어떤 행을 업데이트할지 결정하거나 열에 대한 새 값을 계산하기 위해 테이블 열을 참조해야 합니다.)

대형 오브젝트에서 이 권한은 오브젝트에 쓰거나 자르는 것을 허용합니다.

DELETE

테이블, 뷰 등의 열을 **DELETE**할 수 있습니다. (**DELETE** 명령에는 **SELECT** 권한도 필요합니다. 삭제할 행을 결정하기 위해 테이블 열을 참조해야 하기 때문입니다.)

TRUNCATE

테이블 또는 뷰의 **TRUNCATE**를 허용합니다.

REFERENCES

테이블 또는 테이블의 특정 열을 참조하는 외래 키 제약 조건을 만들 수 있습니다.

TRIGGER

테이블 또는 뷰에 트리거를 만들 수 있습니다.

CREATE

데이터베이스에 대해 데이터베이스에 새 **Schema**와 게시를 만들고 데이터베이스에 신뢰할 수 있는 확장을 만들 수 있습니다.

Schema의 경우 **Schema**에 새 오브젝트를 만들 수 있습니다. 기존 오브젝트의 이름을 바꾸려면 오브젝트를 소유하고 오브젝트를 포함하는 **Schema**에 대해 이 권한을 가져야 합니다.

테이블 스페이스에 대해서는, 그 테이블스페이스 내에 테이블, 인덱스, 임시파일을 작성하는 것을 허용해, 그 테이블 스페이스를 디폴트의 테이블 스페이스로서 가지는 데이터베이스를 작성하는 것을 허가합니다.

이 권한을 박탈해도 기존의 객체의 존재, 또는 그 배치를 변경하지 않습니다.

CONNECT

권한이 부여된 사람이 데이터베이스에 연결할 수 있습니다. (`pg_hba.conf`가 부과하는 제한 점검외에도)이 권한은 연결이 시작될 때 점검됩니다.

TEMPORARY

데이터베이스 사용 중 임시 테이블을 작성할 수 있습니다.

형식과 도메인의 경우 테이블, 함수 및 기타 **Schema** 오브젝트를 생성할 때 형식과 도메인을 사용할 수 있습니다. (이 권한의 주요 목적은 어떤 사용자가 특정 유형에 대한 종속성을 만들 수 있는지 제어하고 나중에 소유자가 이 유형을 변경하지 못하게 하는 것입니다.)

외부 데이터 래퍼의 경우 외부 데이터 래퍼를 사용하여 새 서버를 만들 수 있습니다.

외부 서버의 경우 해당 서버를 사용하여 외부 테이블을 만들 수 있습니다. 권한을 부여받은 사람은 서버에 연결된 사용자 매핑을 생성, 수정 및 삭제할 수 있습니다.

13.4 REVOKE

이전에 주어진 권한을 취소하려면 해당 이름의 **REVOKE** 명령을 사용하십시오.

```
REVOKE ALL ON accounts FROM PUBLIC;
```

일반적으로 오브젝트의 소유자(또는 슈퍼유저)만 오브젝트에 대한 권한을 부여하거나 박탈할 수 있습니다. 그러나 "**with grant option**"을 붙이면 권한이 부여된 사용자가 소유자와 마찬가지로

다른 사용자에게 권한을 부여 할 수 있습니다. 만약 나중에 **grant** 옵션이 박탈되면, 박탈된 유저로부터 (직접 혹은 권한 부여의 체인에 의해) 권한을 부여받은 유저는 모두 그 권한이 박탈됩니다.

오브젝트의 소유자는 소유한 일반 권한을 삭제하도록 선택할 수 있습니다. 예를 들어, 테이블을 읽기 전용으로 만들 수 있습니다. 그러나 소유자는 항상 모든 부여 옵션을 가진 것으로 취급됩니다. 따라서 언제든지 자신의 권한을 다시 부여할 수 있습니다.