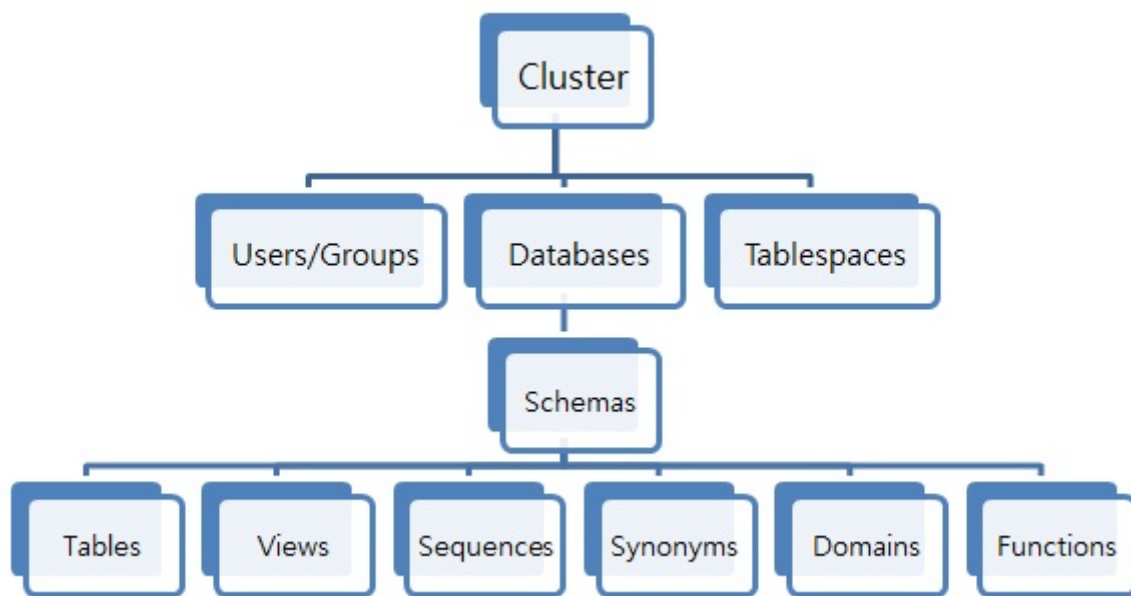


## 9장. Database 관리

하나의 Database Cluster에는 복수 개의 Database를 생성할 수 있으며, 하나의 Database 구조는 여러 개의 Schema와 오브젝트를 소유하고 있습니다.



### 9.1 Database 조회

PSQL에서 아래의 명령어로 Database 조회가 가능합니다.

```
postgres=# \l
```

List of databases

Name	Owner	Encoding	Collate	Ctype	Access privileges
mydb	postgres	UTF8	ko_KR.UTF-8	ko_KR.UTF-8	
postgres	postgres	UTF8	ko_KR.UTF-8	ko_KR.UTF-8	
template0	postgres	UTF8	ko_KR.UTF-8	ko_KR.UTF-8	=c/postgres +
				postgres=CTc/postgres	
template1	postgres	UTF8	ko_KR.UTF-8	ko_KR.UTF-8	=c/postgres +
				postgres=CTc/postgres	

아래의 Catalog View를 통해 Role 조회가 가능합니다.

```
select * from pg_database ;
```

## 9.2 Database 생성

Database를 생성하기 위해서는 SUPERUSER이거나 CREATEDB 권한이 있는 USER이어야 합니다. SUPERUSER는 다른 USER의 DATABASE를 소유할 수 있으며, 일반 USER들은 해당 USER가 생성한 DATABASE만 소유할 수 있습니다.

### 9.2.1 Syntax

```
CREATE DATABASE name
[ WITH ] [ OWNER [=] user_name ]
[ TEMPLATE [=] template ]
[ ENCODING [=] encoding ]
[ STRATEGY [=] strategy ]
[ LOCALE [=] locale ]
[ LC_COLLATE [=] lc_collate ]
[ LC_CTYPE [=] lc_ctype ]
[ ICU_LOCALE [=] icu_locale ]
[ LOCALE_PROVIDER [=] locale_provider ]
[ COLLATION_VERSION = collation_version ]
[ TABLESPACE [=] tablespace_name ]
[ ALLOW_CONNECTIONS [=] allowconn ]
[ CONNECTION LIMIT [=] conlimit ]
[ IS_TEMPLATE [=] istemplate ]
[ OID [=] oid ]
```

- [ [ WITH ] [ OWNER [=] user\_name ] DATABASE를 소유하는 USER 지정

- [ TEMPLATE [=] template ] 지정한 TEMPLATE을 복제하여 미리 정의된 표준객체를 포함한 DATABASE 생성
- [ ENCODING [=] encoding ] DATABASE에서 사용할 ENCODING 설정 (ex : SQL\_ASCII)
- [ LC\_COLLATE [=] lc\_collate ] column, index, order by절에서 문자열에 적용되는 정렬 순서를 설정
- [ LC\_CTYPE [=] lc\_ctype ] 문자의 분류로 대/소문자, 숫자 분류를 설정
- [ TABLESPACE [=] tablespace ] 생성될 Database의 Default Tablespace를 지정
- [ CONNECTION LIMIT [=] conlimit ] Database에 동시 접속을 제한 (Default : -1(제한없음))

### 9.2.2 Example

```
CREATE DATABASE name;

CREATE DATABASE T_D2 OWNER BITNINE;
```

## 9.3 Database 삭제

### 9.3.1 Syntax

```
DROP DATABASE [ IF EXISTS ] name [ [ WITH ] ( option [, ...] ) ]
where option can be:
    FORCE
```

### 9.3.2 Example

```
DROP DATABASE T_D2 ;
```

## 9.4 Database 변경

### 9.4.1 Syntax

```
ALTER DATABASE name [ [ WITH ] option [ ... ] ]

where option can be:

    ALLOW_CONNECTIONS allowconn
    CONNECTION LIMIT conlimit
    IS_TEMPLATE istemplate
```

### 9.4.2 Example

```
# 테이블스페이스 지정(default tablespace는 pg_default)
ALTER DATABASE T_D SET TABLESPACE TBS01 ;

# DB 이름 변경
ALTER DATABASE T_D RENAME TO T_D3 ;

# DB Owner 변경
ALTER DATABASE T_D2 OWNER TO TEST4 ;

# DB Index Scan 사용 변경
ALTER DATABASE T_D3 SET ENABLE_INDEXSCAN TO OFF ;
ALTER DATABASE T_D SET ENABLE_INDEXSCAN FROM CURRENT ;

# DB Index Scan 사용 변경 RESET
ALTER DATABASE T_D RESET ENABLE_INDEXSCAN ;
ALTER DATABASE T_D RESET ALL ;
```

## 10장. Tablespace 관리

Tablespace를 사용하면 데이터베이스 오브젝트를 나타내는 파일을 저장할 수 있는 파일 시스템의 위치를 정의할 수 있습니다. 이를 통해 Database Cluster를 초기화한 파티션 또는 볼륨의 용량이 부족하여 확장이 불가능한 경우, 시스템을 재구성할 때까지 다른 파티션에 Tablespace를 작성하여 이를 사용할 수 있습니다. 또한 오브젝트의 사용 패턴에 따라 데이터 저장 위치를 조정하여 성능 향상을 꾀할 수 있습니다. 기본적으로 DB로 지정된 디렉토리 전체가 하나의 기본 테이블스페이스로 인식되며, 하위 디렉토리에 오브젝트 파일이 생성됩니다. 해당 경로가 pg\_tblspc 디렉토리 밑에 심볼릭 링크로 걸리게 되고, 이 링크를 통해 디렉토리 파일들을 조회가 됩니다.(단 'pg\_'로 시작하는 이름은 불가능)

주의할 점은 Tablespace도 Database Cluster의 필수적인 부분이기 때문에 파티션 위치가 다르더라도 백업이나 이관시 함께 수행해야 하며, HA 구성시에도 양쪽 서버에 동일한 디렉토리를 생성 후 작업해야 합니다.

### 10.1 Tablespace 조회

PSQL에서 아래의 명령어로 Tablespace 조회가 가능합니다.

```
postgres=# \db+
               List of tablespaces
  Name  | Owner  | Location | Access privileges | Options | Size  | Description
-----+-----+-----+-----+-----+-----+-----
pg_default | postgres |          |                   |         | 31 MB |
```