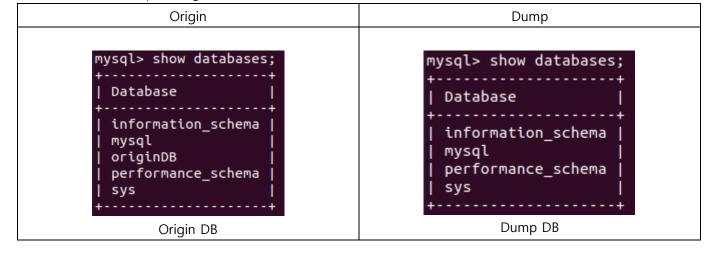
## MySQL - 백업(dump방식)

## MySQL 덤프 명령어

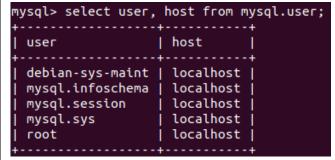
- mysqldump
  - 해당 명령어를 통해 DB, Table등의 정보를 dump하여 backup 시킬 수 있다.
- mysqldump 덤프 방법
  - \$ mysqldump --all-databases -u [user\_ID] -p --default-character-set=[charset] > [Dump All].sql >> 모든 DB를 dump
  - \$ mysqldump -u [user\_ID] -p [Target DB] > [Dump DB].sql >> 특정 DB만을 dump
  - \$ mysqldump -u [user\_ID] -p [Target DB] [Target Table] > [Dump Table].sql >> 특정 DB내의 특정 Table만을 dump
- mysqldump 복원 방법
  - \$ mysql -u [user\_ID] -p < [Dump All].sql
    - >> 모든 DB를 복원
  - \$ mysql -u [user\_ID] -p [Target DB] < [Dump DB].sql >> 특정 DB를 복원
  - \$ mysql -u [user\_ID] -p [Target DB] [Target Table] < [Dump Table].sql >> 특정 Table을 복원

## MySQL 모든 DB 백업 테스트

- 기존 DB 정보
  - 왼쪽 Origin은 덤프할 타겟이고, 오른쪽 Dump는 Origin의 정보를 덤프를 통해 덮어씌울 타겟이다.



```
mysql> select user, host from mysql.user;
                    | host
 user
                     localhost
  debian-sys-maint |
  mysql.infoschema
                      localhost
  mysql.session
                      localhost
                      localhost
  mysql.sys
  originUser
                      localhost
                      localhost
  root
              Origin DB의 User
mysql> select * from originDB.originTable;
 field |
      1
     3
     4
      5
              Origin DB의 Data
```



Dump DB의 User

MySQL을 설치하고 난 뒤 아무런 작업을 하지 않았음으로 Dump DB에는 Data가 존재하지 않는다.

- Origin에는 기존에 저장된 User와 DB, Table, Data가 존재한다.
- Dump는 MySQL DB의 가장 초기 단계이다. 즉, User, DB, Table, Data등이 존재하지 않는다.
- ✓ 이제부터 Origin의 모든 정보를 Dump로 덤프 시켜보도록 한다.
- 1. Origin에서 덤프 명령어 실행

\$ mysqldump --all-databases -u root -p --default-character-set=utf8 > all.sql

```
origin@origin:~/dumpsql$ mysqldump --all-databases -u root -p --default-char
er-set=utf8 > all.sql
Enter password:
origin@origin:~/dumpsql$ ls
all.sql
```

- >> 해당 명령어를 실행하여 덤프된 파일인 all.sql이 생성되었다.
- 2. 덤프된 all.sql 파일을 Dump로 이동
  - >> 여기서는 FileZilla 프로그램을 사용하였으며 FileZilla를 통해 이동하는 자세한 방법은 다루지 않는다.
  - >> FileZilla가 아닌 SSH, FTP 등의 유틸리티를 사용해도 무방하다. (당연히..)

```
dump@computer:~/dumpsql$ ls
all.sql
dump@computer:~/dumpsql$
```

- >> Origin에서 Dump로 옮겨진 all.sql 파일
- 3. all.sql 파일을 Dump의 local DB로 덮어씌우기 \$ mysql -u root -p < all.sql

```
dump@computer:~/dumpsql$ mysql -u root -p < all.sql
Enter password:
dump@computer:~/dumpsql$</pre>
```

- >> 해당 명령어를 입력하면 mysql 비밀번호를 입력 후 아무런 반응이 일어나지 않는다.
- >> 확인을 위해서는 직접 mysql에 접근하여 정상적으로 덮어씌워졌는지 확인해보도록 하자.
- 4. Dump에서 정상적으로 all.sql이 덮어씌워졌는지 확인

```
dump@computer:~/dumpsql$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with; or \g.
Your MySQL connection id is 23
Server version: 8.0.23-0ubuntu0.20.04.1 (Ubuntu)
Copyright (c) 2000, 2021, Oracle and/or its affiliates.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or 'h' for help. Type 'c' to clear the current input statement.
mysql> show databases;
| Database
| information schema |
| mysql
| originDB
| performance_schema |
sys
5 rows in set (0.00 sec)
mysql> select user, host from mysql.user;
                    host
user
| debian-sys-maint | localhost |
| mysql.infoschema | localhost |
| mysql.session | localhost |
| mysql.sys | localhost |
| originUser | localhost |
| root | localhost |
6 rows in set (0.00 sec)
mysql> select * from originDB.originTable;
| field |
      1 |
      2
      3 I
      4 1
      5
5 rows in set (0.00 sec)
```

>> 정상적으로 Origin의 모든 DB 정보(User, DB, Table 등)가 Dump로 덮어씌워진 것을 확인할 수 있다.

## 참고문헌

- https://server-talk.tistory.com/30
- http://chongmoa.com/sql/630