🥍

1주차 실습 - Sqoop

Gitub

0. 사전 환경설정

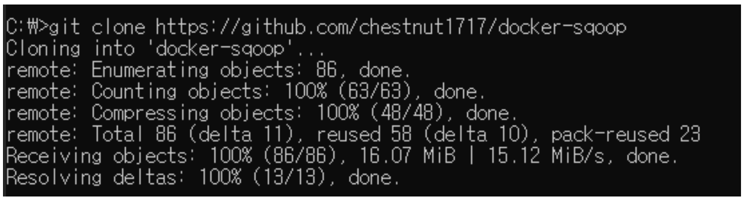
- docker기반으로 실습하기 때문에 방학 base session 6주차 **하둡 wordcount** 실습과 동일

> 💡 **필요한 것!**
>
> 1. **Docker Desktop**
> 2. **git**
> 3. **jdk(환경변수 설정까지)**
> 4. ***(Optional)*** 1번(하둡 및 Sqoop 설치)까지 진행

1. 하둡 및 Sqoop 설치

```
# 사전에 구축해놓은 Dockerfile clone
git clone https://github.com/chestnut1717/docker-sqoop
```

▼ **진행 예시**



```
C:₩>git clone https://github.com/chestnut1717/docker-sqoop
Cloning into 'docker-sqoop'...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (63/63), done.
remote: Compressing objects: 100% (48/48), done.
remote: Total 86 (delta 11), reused 58 (delta 10), pack-reused 23
Receiving objects: 100% (86/86), 16.07 MiB | 15.12 MiB/s, done.
Resolving deltas: 100% (13/13), done.
```

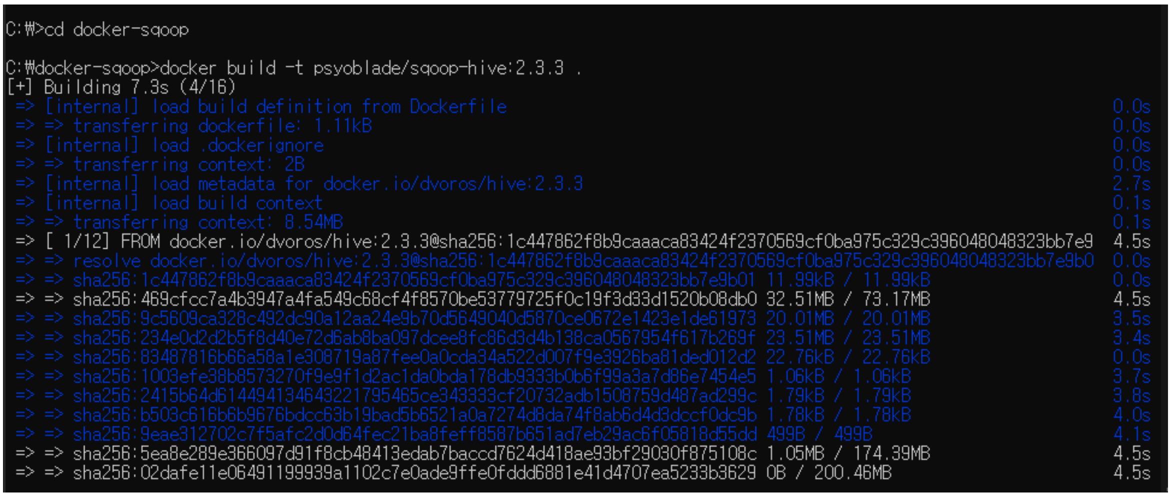
하둡, sqoop이 한번에 설치되기 위해 이미지 bulid

> ✅ **image를 다운받고 build하는데 많은 용량(약 2G)과 시간을 소모하니 미리 해오시면 더욱 좋습니다**

```
cd docker-sqoop
docker build -t psyoblade/sqoop-hive:2.3.3 .

# image가 제대로 올라왔는지 확인
docker images
```

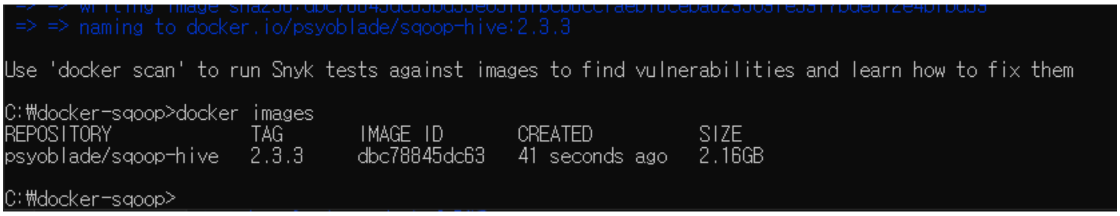
▼ **진행 예시**



```
C:₩>cd docker-sqoop

C:₩docker-sqoop>docker build -t psyoblade/sqoop-hive:2.3.3 .
[+] Building 7.3s (4/16)
 => [internal] load build definition from Dockerfile                                                      0.0s
 => => transferring dockerfile: 1.11kB                                                                    0.0s
 => [internal] load .dockerignore                                                                         0.0s
 => => transferring context: 2B                                                                           0.0s
 => [internal] load metadata for docker.io/dvoros/hive:2.3.3                                              2.7s
 => [internal] load build context                                                                         0.1s
 => => transferring context: 8.54MB                                                                       0.1s
 => [ 1/12] FROM docker.io/dvoros/hive:2.3.3@sha256:1c447862f8b9caaaca83424f2370569cf0ba975c329c396048048323bb7e9  4.5s
 => => resolve docker.io/dvoros/hive:2.3.3@sha256:1c447862f8b9caaaca83424f2370569cf0ba975c329c396048048323bb7e9b0  0.0s
 => => sha256:1c447862f8b9caaaca83424f2370569cf0ba975c329c396048048323bb7e9b01 11.99kB / 11.99kB          0.0s
 => => sha256:469cfcc7a4b3947a4fa549c68cf4f8570be53779725f0c19f3d33d1520b08db0 32.51MB / 73.17MB          4.5s
 => => sha256:9c5609ca328c492dc90a12aa24e9b70d5649040d5870ce0672e1423e1de61973 20.01MB / 20.01MB          3.5s
 => => sha256:234e0d2d2b5f8d40e72d6ab8ba097dcee8fc86d3d4b138ca0567954f617b269f 23.51MB / 23.51MB          3.4s
 => => sha256:83487816b66a58a1e308719a87fee0a0cda34a522d007f9e3926ba81ded012d2 22.76kB / 22.76kB          0.0s
 => => sha256:1003efe38b8573270f9e9f1d2ac1da0bda178db9333b0b6f99a3a7d86e7454e5 1.06kB / 1.06kB            3.7s
 => => sha256:2415b64d6144941346432217954 65ce343333cf20732adb1508759d487ad299c 1.79kB / 1.79kB           3.8s
 => => sha256:b503c616b6b9676bdcc63b19bad5b6521a0a7274d8da74f8ab6d4d3dccf0dc9b 1.78kB / 1.78kB            4.0s
 => => sha256:9eae312702c7f5afc2d0d64fec21ba8feff8587b651ad7eb29ac6f05818d55dd 499B / 499B                4.1s
 => => sha256:5ea8e289e366097d91f8cb48413edab7baccd7624d418ae93bf29030f875108c 1.05MB / 174.39MB          4.5s
 => => sha256:02dafe11e06491199939a1102c7e0ade9ffe0fddd6881e41d4707ea5233b3629 0B / 200.46MB              4.5s
```

진행중인 창

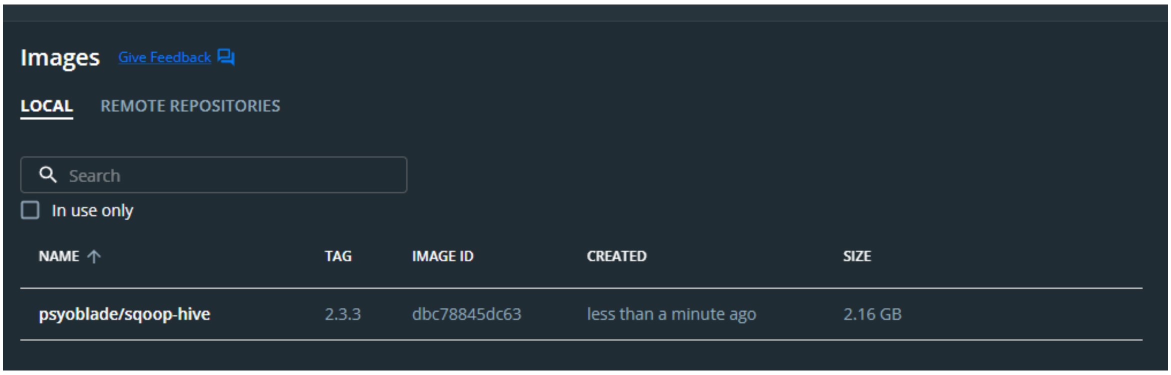


```
 => => writing image sha256:[illegible]
 => => naming to docker.io/psyoblade/sqoop-hive:2.3.3

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them

C:₩docker-sqoop>docker images
REPOSITORY             TAG       IMAGE ID       CREATED          SIZE
psyoblade/sqoop-hive   2.3.3     dbc78845dc63   41 seconds ago   2.16GB

C:₩docker-sqoop>
```



NAME ↑	TAG	IMAGE ID	CREATED	SIZE
psyoblade/sqoop-hive	2.3.3	dbc78845dc63	less than a minute ago	2.16 GB

성공하면 다음과 같이 이미지가 올라온 것을 확인 가능

2. MySQL 설치

```
docker run -d --rm --name mysql -e "MYSQL_ALLOW_EMPTY_PASSWORD=yes" -v `pwd`/data/mysql:/var/lib/mysql -it mysql
```

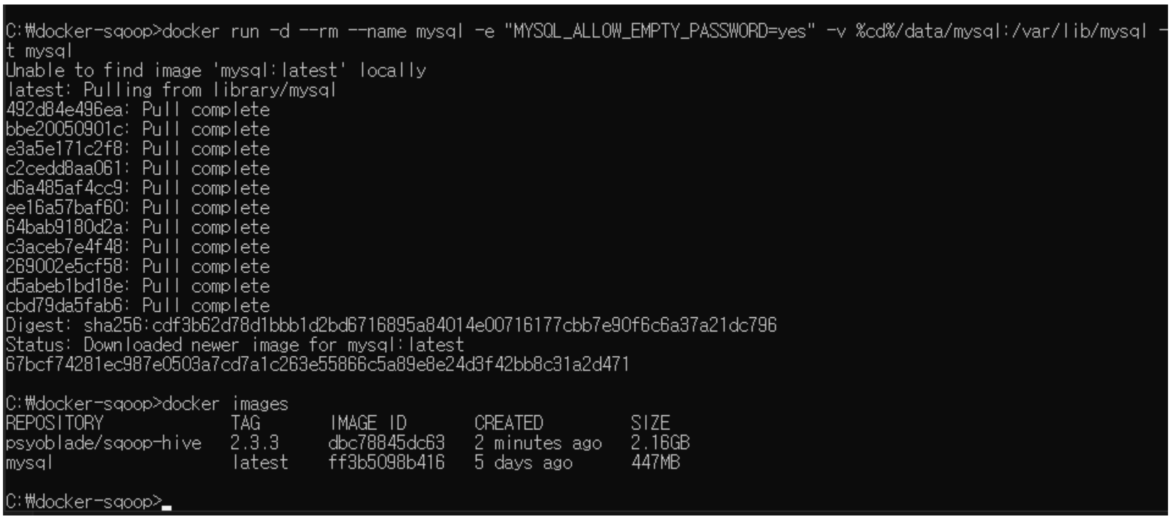
✅ **Windows cmd**창에서 할 경우 `‘pwd’` ⇒ `%cd%`로 하기

```
# for windows cmd
docker run -d --rm --name mysql -e "MYSQL_ALLOW_EMPTY_PASSWORD=yes" -v %cd%/data/mysql:/var/lib/mysql -it mysql
```

✅ **Windows powershell**에서 할 경우 `‘pwd’` ⇒ `${pwd}`로 하기

```
# for windows powershell
docker run -d --rm --name mysql -e "MYSQL_ALLOW_EMPTY_PASSWORD=yes" -v ${pwd}/data/mysql:/var/lib/mysql -it mysql
```

▼ **진행 예시**



```
C:₩docker-sqoop>docker run -d --rm --name mysql -e "MYSQL_ALLOW_EMPTY_PASSWORD=yes" -v %cd%/data/mysql:/var/lib/mysql -
t mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
492d84e496ea: Pull complete
bbe20050901c: Pull complete
e3a5e171c2f8: Pull complete
c2cedd8aa061: Pull complete
d6a485af4cc9: Pull complete
ee16a57baf60: Pull complete
64bab9180d2a: Pull complete
c3aceb7e4f48: Pull complete
269002e5cf58: Pull complete
d5abeb1bd18e: Pull complete
cbd79da5fab6: Pull complete
Digest: sha256:cdf3b62d78d1bbb1d2bd6716895a84014e00716177cbb7e90f6c6a37a21dc796
Status: Downloaded newer image for mysql:latest
67bcf74281ec987e0503a7cd7a1c263e55866c5a89e8e24d3f42bb8c31a2d471

C:₩docker-sqoop>docker images
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
psyoblade/sqoop-hive    2.3.3     dbc78845dc63   2 minutes ago    2.16GB
mysql                   latest    ff3b5098b416   5 days ago       447MB

C:₩docker-sqoop>_
```

성공하면 docker images 명령어를 입력하면 다음과 같이 뜸

MySQL 컨테이너 실행

```
docker exec -it mysql -uroot
mysql
```

✅ **Windows 환경에서는 `-uroot` ⇒ `bash`로 하기**

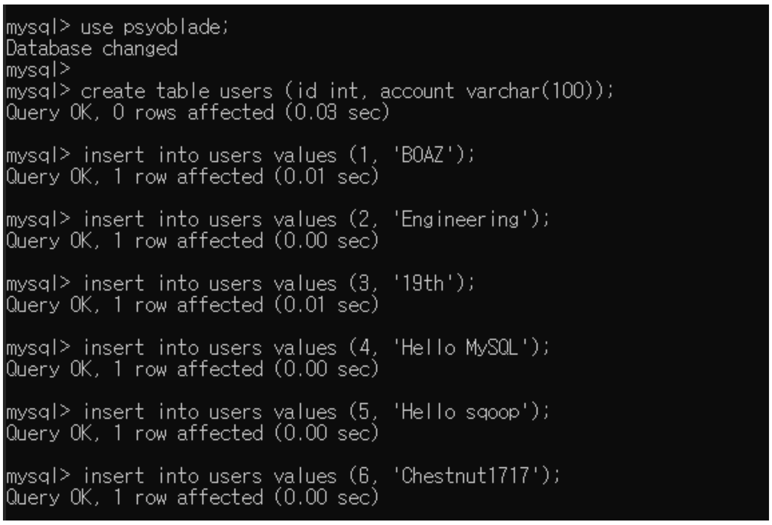
```
docker exec -it mysql bash
mysql
```

MySQL에 테스트 테이블 만들기

```
create database psyoblade;
use psyoblade;

create table users (id int, account varchar(100));
insert into users values (1, 'BOAZ');
insert into users values (2, 'Engineering');
insert into users values (3, '19th');
insert into users values (4, 'Hello Hadoop');
insert into users values (5, 'Hello Sqoop');
insert into users values (6, 'Chestnut1717');
```

▼ **진행 예시**



```
mysql> use psyoblade;
Database changed
mysql>
mysql> create table users (id int, account varchar(100));
Query OK, 0 rows affected (0.03 sec)

mysql> insert into users values (1, 'BOAZ');
Query OK, 1 row affected (0.01 sec)

mysql> insert into users values (2, 'Engineering');
Query OK, 1 row affected (0.00 sec)

mysql> insert into users values (3, '19th');
Query OK, 1 row affected (0.01 sec)

mysql> insert into users values (4, 'Hello MySQL');
Query OK, 1 row affected (0.00 sec)

mysql> insert into users values (5, 'Hello sqoop');
Query OK, 1 row affected (0.00 sec)

mysql> insert into users values (6, 'Chestnut1717');
Query OK, 1 row affected (0.00 sec)
```

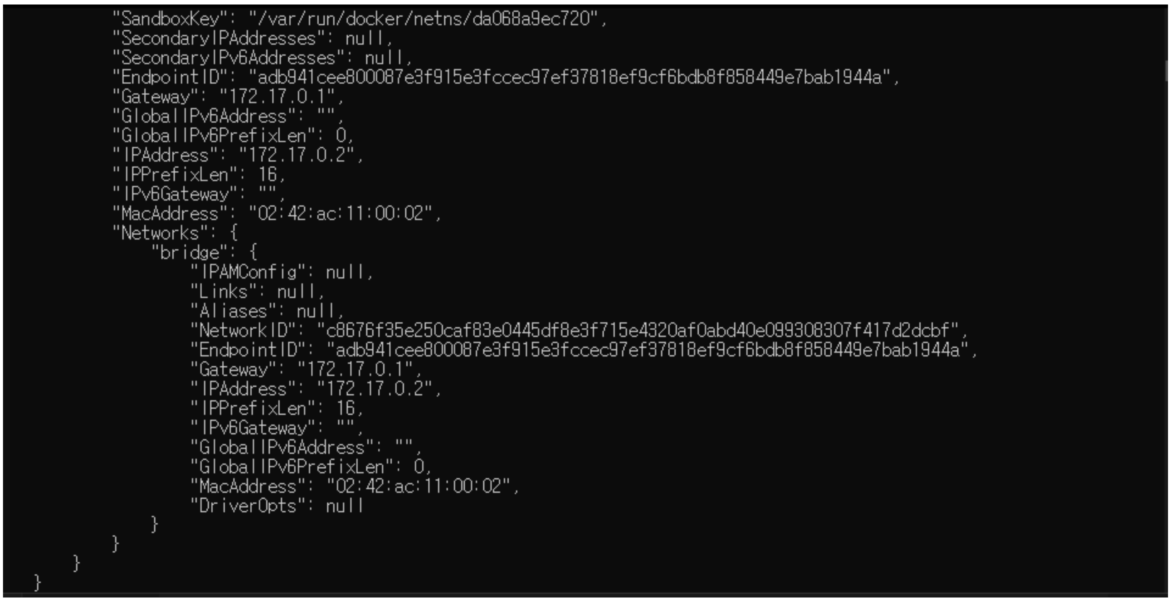
테이블을 생성했으면, docker 명령어 입력하기 위해 command창으로 돌아오기(ctrl (cmd)+ C or D)

3. internal IP 확인

```
docker inspect mysql # 명령을 통해 network 섹션을 확인합니다

# 결과창
        [생략]
                    "GlobalIPv6PrefixLen": 0,
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "MacAddress": "02:42:ac:11:00:02",
                    "Networks": {
                        "bridge": {
                            "IPAMConfig": null,
                            "Links": null,
                            "Aliases": null,
                            "NetworkID": "262cd6269a27c837f570cb6a3cc9ed665527e459b43acd0442d6ddd9e60f08e0",
                            "EndpointID": "2f05d5a0aacbd522120ae1941a6453bd960d81f6e8bf3fd9735a15f3c66e1f81",
                            "Gateway": "172.17.0.1",
                            "IPAddress": "172.17.0.2",
                            "IPPrefixLen": 16,
                            "IPv6Gateway": "",
                            "GlobalIPv6Address": "",
                            "GlobalIPv6PrefixLen": 0,
                            "MacAddress": "02:42:ac:11:00:02",
                            "DriverOpts": null
                        }
                    }
                }
            }
        ]
```

▼ 진행 예시



```
            "SandboxKey": "/var/run/docker/netns/da068a9ec720",
            "SecondaryIPAddresses": null,
            "SecondaryIPv6Addresses": null,
            "EndpointID": "adb941cee800087e3f915e3fccec97ef37818ef9cf6bdb8f858449e7bab1944a",
            "Gateway": "172.17.0.1",
            "GlobalIPv6Address": "",
            "GlobalIPv6PrefixLen": 0,
            "IPAddress": "172.17.0.2",
            "IPPrefixLen": 16,
            "IPv6Gateway": "",
            "MacAddress": "02:42:ac:11:00:02",
            "Networks": {
                "bridge": {
                    "IPAMConfig": null,
                    "Links": null,
                    "Aliases": null,
                    "NetworkID": "c8676f35e250caf83e0445df8e3f715e4320af0abd40e099308307f417d2dcbf",
                    "EndpointID": "adb941cee800087e3f915e3fccec97ef37818ef9cf6bdb8f858449e7bab1944a",
                    "Gateway": "172.17.0.1",
                    "IPAddress": "172.17.0.2",
                    "IPPrefixLen": 16,
                    "IPv6Gateway": "",
                    "GlobalIPv6Address": "",
                    "GlobalIPv6PrefixLen": 0,
                    "MacAddress": "02:42:ac:11:00:02",
                    "DriverOpts": null
                }
            }
        }
    }
```

4. Sqoop 실행 - import 과정

간단하게 네트워크 생성 후 연결

```
# container간(sqoop - mysql) 네트워크 통신 가능하게 하기 위함
docker network create sqoop-mysql
docker network connect sqoop-mysql mysql

# sqoop image run 후 container 실행
docker run --name sqoop --network sqoop-mysql -v %cd%/jars:/jdbc -dit psyoblade/sqoop-hive:2.3.3
docker exec -u root -it sqoop bash
```

sqoop image run

```
docker run --name sqoop --network sqoop-mysql -v `pwd`/jars:/jdbc -dit psyoblade/sqoop-hive:2.3.3
```

✅ **Windows cmd**창에서 할 경우 `‘pwd’` ⇒ `%cd%`로 하기

```
# for windows cmd
docker run --name sqoop --network sqoop-mysql -v %cd%/jars:/jdbc -dit psyoblade/sqoop-hive:2.3.3
```

✅ **Windows powershell**에서 할 경우 `‘pwd’` ⇒ `${pwd}`로 하기

```
# for windows powershell
docker run --name sqoop --network sqoop-mysql -v ${pwd}/jars:/jdbc -dit psyoblade/sqoop-hive:2.3.3
```

container 실행

```
# root 이름으로 sqoop 실행
docker exec -u root -it sqoop -uroot
```

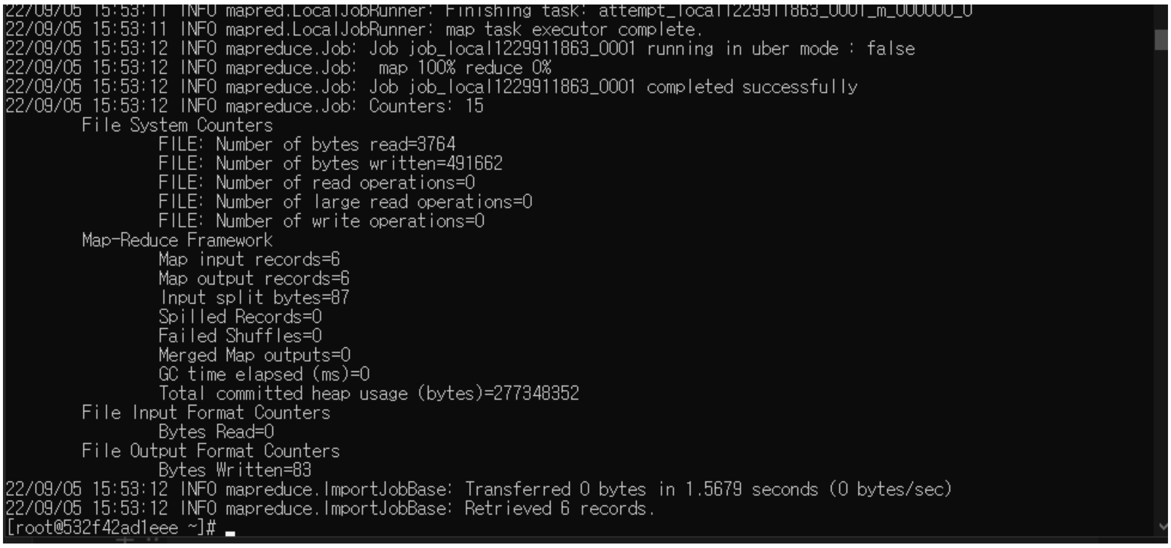
✅ **Windows** 환경에서는 `-uroot` ⇒ `bash`로 하기

```
# for windows
docker exec -u root -it sqoop bash
```

MySQL에 저장된 테이블의 데이터를 import

```
sqoop import \
          -jt local \
          -fs local \
          -m 1 \
          --connect jdbc:mysql://mysql:3306/psyoblade \
          --username root \
          --table users \
          --target-dir /tmp/sqoop/t2
```

▼ **진행 예시**

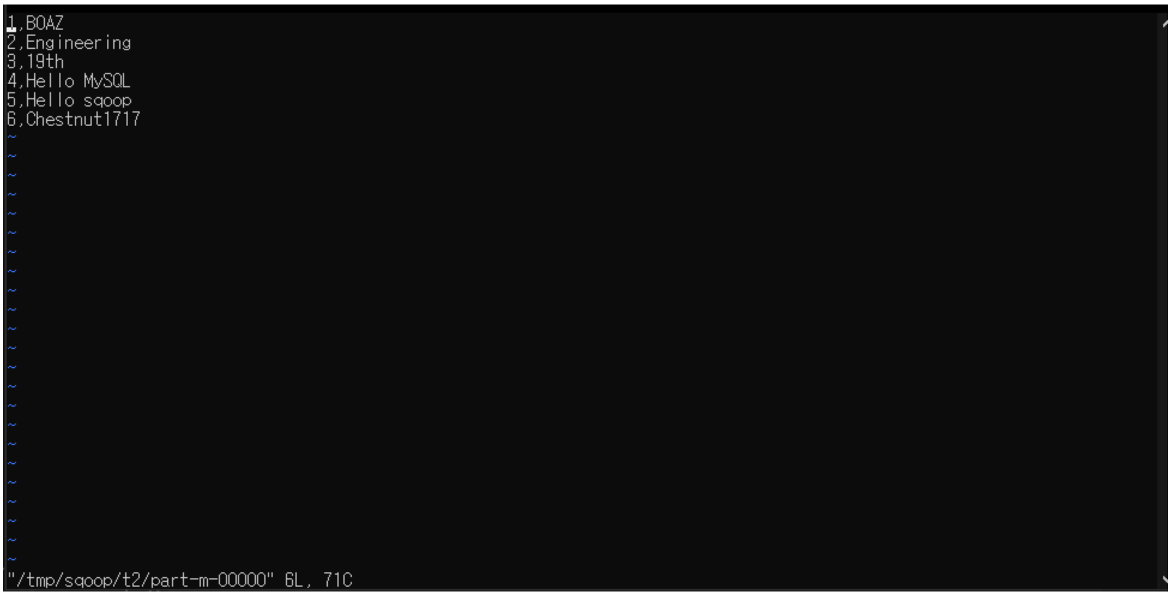


```
22/09/05 15:53:11 INFO mapred.LocalJobRunner: Finishing task: attempt_local1229911863_0001_m_000000_0
22/09/05 15:53:11 INFO mapred.LocalJobRunner: map task executor complete.
22/09/05 15:53:12 INFO mapreduce.Job: Job job_local1229911863_0001 running in uber mode : false
22/09/05 15:53:12 INFO mapreduce.Job:  map 100% reduce 0%
22/09/05 15:53:12 INFO mapreduce.Job: Job job_local1229911863_0001 completed successfully
22/09/05 15:53:12 INFO mapreduce.Job: Counters: 15
        File System Counters
                FILE: Number of bytes read=3764
                FILE: Number of bytes written=491662
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
        Map-Reduce Framework
                Map input records=6
                Map output records=6
                Input split bytes=87
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=0
                Total committed heap usage (bytes)=277348352
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=83
22/09/05 15:53:12 INFO mapreduce.ImportJobBase: Transferred 0 bytes in 1.5679 seconds (0 bytes/sec)
22/09/05 15:53:12 INFO mapreduce.ImportJobBase: Retrieved 6 records.
[root@532f42ad1eee ~]#
```

import한 데이터 살펴보기

```
vi /tmp/sqoop/t2/part-m-00000
```

▼ **진행 예시**



```
1,BOAZ
2,Engineering
3,19th
4,Hello MySQL
5,Hello sqoop
6,Chestnut1717
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"/tmp/sqoop/t2/part-m-00000" 6L, 71C
```

참고자료

- https://hub.docker.com/r/psyoblade/docker-sqoop
- https://velog.io/@yoounseules/도커-볼륨을-이용한-소스-변경시-볼륨실행-에러윈도우환경
- sqoop version : https://blog.naver.com/PostView.nhn?isHttpsRedirect=true&blogId=shyoo_1990&logNo=221274353927&parentCategoryNo=&categoryNo=23&viewDate=&isShow