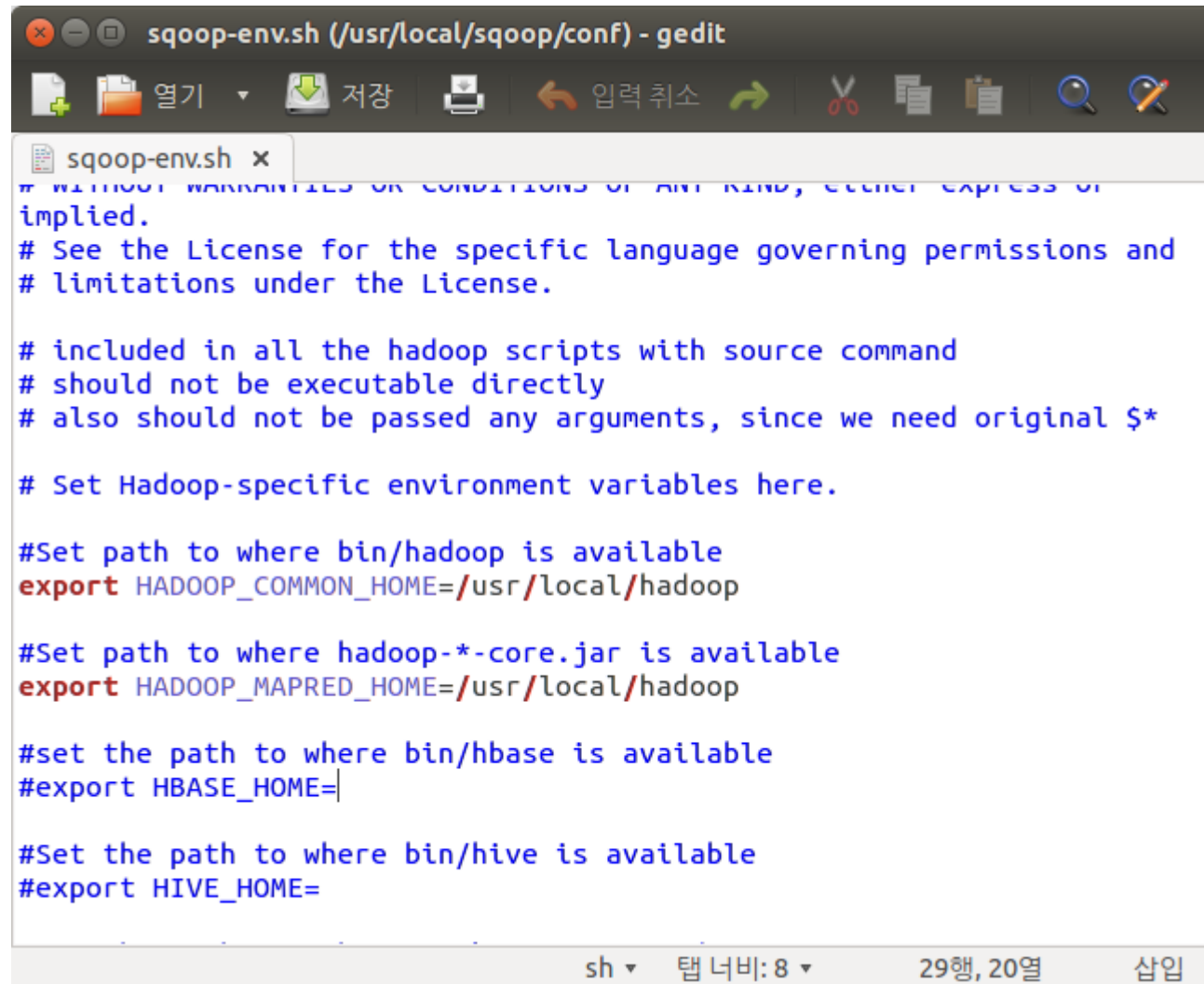


1.sqoop설치

<http://apache.mirror.cdnetworks.com/sqoop/1.4.6/>

2.sqoop설정

•Sqoop-env.sh



```
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# included in all the hadoop scripts with source command
# should not be executable directly
# also should not be passed any arguments, since we need original $*

# Set Hadoop-specific environment variables here.

#Set path to where bin/hadoop is available
export HADOOP_COMMON_HOME=/usr/local/hadoop

#Set path to where hadoop-*-core.jar is available
export HADOOP_MAPRED_HOME=/usr/local/hadoop

#set the path to where bin/hbase is available
#export HBASE_HOME=

#Set the path to where bin/hive is available
#export HIVE_HOME=
```

2.sqoop설정

•Sqoop/lib에 jdbc드라이버 복사

```
sist@sist-Samsung-DeskTop-System: /usr/local/sqoop/lib
LICENSE.txt      conf      metastore_db    sqoop-test-1.4.6.jar
NOTICE.txt       derby.log  pom-old.xml     src
README.txt       docs      sale.sh         testdata
SALE_TB.java     ivy       sex_sal_tb.java
sist@sist-Samsung-DeskTop-System: /usr/local/sqoop$ cd lib
sist@sist-Samsung-DeskTop-System: /usr/local/sqoop/lib$ ls
ant-contrib-1.0b3.jar      kite-data-mapreduce-1.0.0.jar
ant-eclipse-1.0-jvm1.2.jar kite-hadoop-compatibility-1.0.0.jar
avro-1.7.5.jar            ojdbc6.jar
avro-mapred-1.7.5-hadoop2.jar opencsv-2.3.jar
commons-codec-1.4.jar     paranamer-2.3.jar
commons-compress-1.4.1.jar parquet-avro-1.4.1.jar
commons-io-1.4.jar        parquet-column-1.4.1.jar
commons-jexl-2.1.1.jar    parquet-common-1.4.1.jar
commons-logging-1.1.1.jar parquet-encoding-1.4.1.jar
hsqldb-1.8.0.10.jar       parquet-format-2.0.0.jar
jackson-annotations-2.3.0.jar parquet-generator-1.4.1.jar
jackson-core-2.3.1.jar    parquet-hadoop-1.4.1.jar
jackson-core-asl-1.9.13.jar parquet-jackson-1.4.1.jar
jackson-databind-2.3.1.jar slf4j-api-1.6.1.jar
jackson-mapper-asl-1.9.13.jar snappy-java-1.0.5.jar
kite-data-core-1.0.0.jar  xz-1.0.jar
kite-data-hive-1.0.0.jar
sist@sist-Samsung-DeskTop-System: /usr/local/sqoop/lib$
```

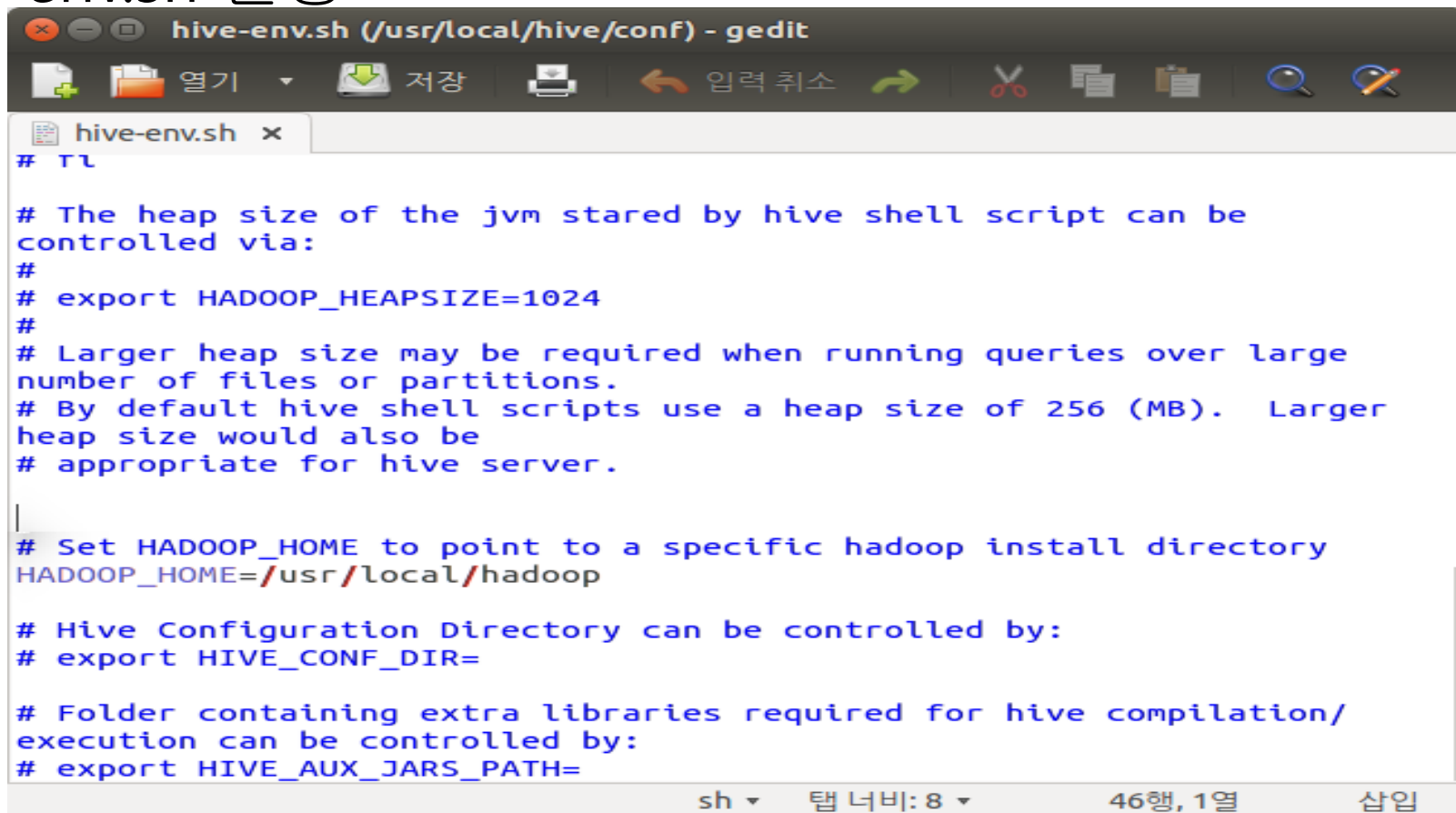
3.hive설치

- <http://apache.mirror.cdnetworks.com/hive/hive-2.3.2/>

4.Hive 설정

- mv conf/hive-env.sh.template conf/hive-env.sh로 hive-env.sh 생성

- Hive-env.sh 설정



The screenshot shows a gedit editor window titled "hive-env.sh (/usr/local/hive/conf) - gedit". The window contains the following text:

```
# TL

# The heap size of the jvm started by hive shell script can be
controlled via:
#
# export HADOOP_HEAPSIZE=1024
#
# Larger heap size may be required when running queries over large
number of files or partitions.
# By default hive shell scripts use a heap size of 256 (MB).  Larger
heap size would also be
# appropriate for hive server.

# Set HADOOP_HOME to point to a specific hadoop install directory
HADOOP_HOME=/usr/local/hadoop

# Hive Configuration Directory can be controlled by:
# export HIVE_CONF_DIR=

# Folder containing extra libraries required for hive compilation/
execution can be controlled by:
# export HIVE_AUX_JARS_PATH=
```

The status bar at the bottom indicates "sh", "탭 너비: 8", "46행, 1열", and "삽입".

4.Hive 설정

- mv conf/hive-env.sh.template conf/hive-env.sh로 hive-env.sh 생성

- Hive-site.xml 설정

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
```

```
<configuration>
```

```
  <property>
```

```
    <name>hive.metastore.local</name>
```

```
    <value>>false</value>
```

```
  </property>
```

```
  <property>
```

```
    <name>javax.jdo.option.ConnectionURL</name>
```

```
    <value>jdbc:oracle:thin:@211.238.142.109:1521:orcl</value>
```

4.Hive 설정

- hive/lib에 jdbc드라이버 복사

5.sqoop에서 hive로 import

```
./bin/sqoop import --connect  
jdbc:oracle:thin:@211.238.142.109:1521:orcl --  
username AC --password 4321 --table SALE_TB -  
-hive-import --create-hive-table --hive-table  
default.sale
```


6.hive 실행

hive폴더로 들어가서 ./bin/hive를 하면 실행됨.

7.hive 쿼리 실행

Show tables로 테이블 확인

```
sist@sist-Samsung-DeskTop-System: /usr/local/hive
sist@sist-Samsung-DeskTop-System: /usr/local/hive$ ls
${system:java.io.tmpdir}  bin                examples  metastore_db
LICENSE                  binary-package-licenses  hcatalog  scripts
NOTICE                   conf               jdbc
RELEASE_NOTES.txt        derby.log          lib
sist@sist-Samsung-DeskTop-System: /usr/local/hive$ ./bin/hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-2.3.2-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.8.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]

Logging initialized using configuration in jar:file:/usr/local/apache-hive-2.3.2-bin/lib/hive-common-2.3.2.jar!/hive-log4j2.properties Async: true
Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
hive> show tables;
OK
sale
Time taken: 2.696 seconds, Fetched: 1 row(s)
hive> 
```

7.hive 쿼리 실행

•Show create table로 테이블 생성

```
sist@sist-Samsung-DeskTop-System: /usr/local/hive
Time taken: 2.696 seconds, Fetched: 1 row(s)
hive> show create table sale;
OK
CREATE TABLE `sale` (
  `sto_id` string,
  `sal_dt` string,
  `pdt_no` string,
  `sal_cnt` double,
  `sex_cd` string,
  `age_cd` string)
COMMENT 'Imported by sqoop on 2018/01/02 15:38:32'
ROW FORMAT SERDE
  'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe'
WITH SERDEPROPERTIES (
  'field.delim'='09',
  'line.delim'='\n',
  'serialization.format'='09')
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  'hdfs://localhost:9000/user/hive/warehouse/sale'
TBLPROPERTIES (
```

7.hive 쿼리 실행

• 쿼리 문 실행

```
sist@sist-Samsung-DeskTop-System: /usr/local/hive
hive> select * from sale;
OK
11111 2017-12-12 00:00:00.0 00001 10.0 1 20
11111 2017-12-12 00:00:00.0 00002 7.0 1 20
11111 2017-12-12 00:00:00.0 00003 9.0 1 20
11111 2017-12-12 00:00:00.0 00004 11.0 1 20
11111 2017-12-12 00:00:00.0 00005 15.0 1 20
11111 2017-12-12 00:00:00.0 00006 19.0 1 20
11111 2017-12-13 00:00:00.0 00003 9.0 1 20
11111 2017-12-20 12:32:52.0 00001 1.0 A1_01 A2_02
11111 2017-12-20 12:32:57.0 00001 1.0 A1_01 A2_02
11111 2017-12-20 12:33:41.0 00003 2.0 A1_01 A2_01
11111 2017-12-20 12:33:45.0 00003 2.0 A1_01 A2_01
11111 2017-12-20 12:45:32.0 00001 2.0 A1_01 A2_02
11111 2017-12-21 11:55:07.0 00001 1.0 A1_01 A2_02
11111 2017-12-28 09:01:47.0 00003 5.0 A1_02 A2_01
11111 2017-12-19 18:35:07.0 00001 3.0 A2_03 A3_01
11111 2017-12-19 18:35:37.0 00001 3.0 A2_03 A3_01
11111 2017-12-12 00:00:00.0 00030 10.0 1 20
11111 2017-12-12 00:00:00.0 00031 10.0 1 20
11111 2017-12-12 00:00:00.0 00032 7.0 1 20
11111 2017-12-13 00:00:00.0 00033 9.0 1 20
11111 2017-12-14 00:00:00.0 00030 10.0 1 20
11111 2017-12-20 12:27:51.0 00001 3.0 A1_01 A2_01
```

8.hive export

.1. select 문을 가지고 테이블을 만들 수 있음.

.2. export 명령어

```
../bin/sqoop export --connect  
jdbc:oracle:thin:@211.238.142.109:1521:orcl --username AC --  
password 4321 --table sex_sal_tb --direct --export-dir  
/user/hive/warehouse/sex_sal_tb --input-fields-terminated-by  
'₩001'
```

.Hive table에 기본 구분자

.-FIELDS TERMINATED BY '₩001'

.-COLLECTION ITEMS TERMINATED BY '₩002'

.-MAP KEYS TERMINATED BY '₩003'

.-LINES TERMINATED BY '₩n'