EXPERIMENT NO: 2

AIM: Use of Sqoop tool to transfer data between Hadoop and relational database servers.

Theory:

Sqoop is a versatile and powerful tool used to facilitate data transfer between Hadoop-based systems, such as the Hadoop Distributed File System (HDFS), and relational database management systems (RDBMS). Here's a theoretical overview of the use of Sqoop for transferring data between Hadoop and RDBMS:

1. **Data Integration Needs**: Many organizations need to combine and analyze data from both their traditional RDBMS and Hadoop-based systems to gain valuable insights. This necessitates the need for a tool that can efficiently transfer data between these different storage and processing environments.
2. **Sqoop Overview**: Sqoop, short for "SQL to Hadoop," is an open-source data transfer tool specifically designed for this purpose. It acts as a bridge between Hadoop and RDBMS, enabling bi-directional data transfers.
3. **Data Import**: When importing data from an RDBMS to Hadoop, Sqoop reads data from the RDBMS, converts it to Hadoop-friendly formats (e.g., Avro or Parquet), and stores it in HDFS. Users can specify various options, such as the source database connection, target HDFS directory, and data transformation methods.
4. **Data Export**: When exporting data from Hadoop to an RDBMS, Sqoop reads data from HDFS, transforms it if necessary, and loads it into the RDBMS. Users can specify the target database connection, source HDFS directory, and other export options.
5. **Use Cases**: Sqoop is valuable for various use cases, including:
    1. Loading operational data into Hadoop for analytics.
    2. Moving data from Hadoop to an RDBMS for reporting.
    3. Data migration and replication between different RDBMS systems.
    4. Continuous data integration and synchronization.

Step 1: open clouders and connect to mysql and view databases.

```
[training@localhost ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database bank;
ERROR 1007 (HY000): Can't create database 'bank'; database exists
mysql> create database bank1;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| bank               |
| bank1              |
| db1                |
| db2                |
| example            |
| hivemetastore      |
| movielens          |
| mysql              |
ining@localhost:~      |
```

Step 2: create database:

```
mysql> use bank1;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table registercopy(accno varchar(20), name varchar(20), number va
Query OK, 0 rows affected (0.04 sec)

mysql> describe registercopy;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| accno    | varchar(20) | YES  |     | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| number   | varchar(20) | YES  |     | NULL    |       |
| email    | varchar(20) | YES  |     | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
| age      | varchar(20) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Step 3: insert values

```
+----------+-------------+------+-----+---------+-------+
| accmo    | varchar(20) | YES  |     | NULL    |       |
| name     | varchar(20) | YES  |     | NULL    |       |
| number   | varchar(20) | YES  |     | NULL    |       |
| email    | varchar(20) | YES  |     | NULL    |       |
| password | varchar(20) | YES  |     | NULL    |       |
| age      | varchar(3)  | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> insert into register values("1","harsh","21","harsh@email","harsh"
,"21");
Query OK, 1 row affected (0.00 sec)

mysql> insert into register values("2","disha","22","dishah@email","disha","20");
Query OK, 1 row affected (0.00 sec)

mysql> insert into register values("3","sahil","23","sahil@email","sahil","20");
Query OK, 1 row affected (0.00 sec)

mysql> insert into register values("4","ishwar","24","ishwar@email","ishwar","20");
Query OK, 1 row affected (0.00 sec)

mysql> insert into register values("5","pankaj","25","pankaj@email","pankaj","20");
Query OK, 1 row affected (0.00 sec)
```

Step 4: view data

```
elet * from registercopy' at line 1
mysql> select * from registercopy;
+-------+--------+--------+--------------+----------+------+
| accno | name   | number | email        | password | age  |
+-------+--------+--------+--------------+----------+------+
| 1     | harsh  | 21     | harsh@email  | harsh    | 21   |
| 2     | disha  | 22     | dishah@email | disha    | 20   |
| 3     | sahil  | 23     | sahil@email  | sahil    | 20   |
| 4     | ishwar | 24     | ishwar@email | ishwar   | 20   |
| 5     | pankaj | 25     | pankaj@email | pankaj   | 20   |
+-------+--------+--------+--------------+----------+------+
5 rows in set (0.00 sec)
```

Step 5: expoerting the data from hdfs to mysql

Syntax:

Sqoop export –connect jdbc:mysql://localhost/db—username root –table<table_name> --export-dir<directory>

```
[cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://localhost/bank --username root --password cloudera --tab
le registercopy --export-dir /home/cloudera/myfirstdata
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/09/13 05:25:52 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
23/09/13 05:25:52 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P i
nstead.
23/09/13 05:25:52 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/09/13 05:25:52 INFO tool.CodeGenTool: Beginning code generation
23/09/13 05:25:53 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `registercopy` AS t LIMIT 1
23/09/13 05:25:53 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `registercopy` AS t LIMIT 1
23/09/13 05:25:53 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/1651b0eae1f9ed0cb04ac1839851920a/registercopy.java uses or overrides a deprecated A
PI.
Note: Recompile with -Xlint:deprecation for details.
23/09/13 05:26:00 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/1651b0eae1f9ed0cb04ac183
9851920a/registercopy.jar
23/09/13 05:26:00 INFO mapreduce.ExportJobBase: Beginning export of registercopy
23/09/13 05:26:00 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker
.address
23/09/13 05:26:01 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
23/09/13 05:26:03 INFO Configuration.deprecation: mapred.reduce.tasks.speculative.execution is deprecated. Instead, u
se mapreduce.reduce.speculative
23/09/13 05:26:03 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use
mapreduce.map.speculative
23/09/13 05:26:03 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
23/09/13 05:26:03 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/09/13 05:26:08 INFO input.FileInputFormat: Total input paths to process : 1
23/09/13 05:26:08 INFO input.FileInputFormat: Total input paths to process : 1
23/09/13 05:26:08 INFO mapreduce.JobSubmitter: number of splits:4
23/09/13 05:26:08 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use
mapreduce.map.speculative
23/09/13 05:26:09 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1694606951119_0001
23/09/13 05:26:11 INFO impl.YarnClientImpl: Submitted application application_1694606951119_0001
23/09/13 05:26:11 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_169
4606951119_0001/
23/09/13 05:26:11 INFO mapreduce.Job: Running job: job_1694606951119_0001
23/09/13 05:26:42 INFO mapreduce.Job: Job job_1694606951119_0001 running in uber mode : false
23/09/13 05:26:42 INFO mapreduce.Job:  map 0% reduce 0%
23/09/13 05:27:08 INFO mapreduce.Job:  map 100% reduce 0%
23/09/13 05:27:09 INFO mapreduce.Job: Job job_1694606951119_0001 completed successfully
23/09/13 05:27:09 INFO mapreduce.Job: Counters: 30
        File System Counters
```

```
4606951119_0001/
23/09/13 05:26:11 INFO mapreduce.Job: Running job: job_1694606951119_0001
23/09/13 05:26:42 INFO mapreduce.Job: Job job_1694606951119_0001 running in uber mode : false
23/09/13 05:26:42 INFO mapreduce.Job:  map 0% reduce 0%
23/09/13 05:27:08 INFO mapreduce.Job:  map 100% reduce 0%
23/09/13 05:27:09 INFO mapreduce.Job: Job job_1694606951119_0001 completed successfully
23/09/13 05:27:09 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=683776
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=1209
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=19
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=4
                Data-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=85742
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=85742
                Total vcore-milliseconds taken by all map tasks=85742
                Total megabyte-milliseconds taken by all map tasks=87799808
        Map-Reduce Framework
                Map input records=5
                Map output records=5
                Input split bytes=691
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=504
                CPU time spent (ms)=4650
                Physical memory (bytes) snapshot=691548160
                Virtual memory (bytes) snapshot=6285340672
                Total committed heap usage (bytes)=695730176
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
23/09/13 05:27:09 INFO mapreduce.ExportJobBase: Transferred 1.1807 KB in 66.0434 seconds (18.3062 bytes/sec)
23/09/13 05:27:09 INFO mapreduce.ExportJobBase: Exported 5 records.
[cloudera@quickstart ~]$
```

Step 6: we have successfully exported the data

```
mysql> select * from register;
+-------+--------+--------+---------------+----------+------+
| accmo | name   | number | email         | password | age  |
+-------+--------+--------+---------------+----------+------+
| 1     | harsh  | 21     | harsh@email   | harsh    | 21   |
| 2     | disha  | 22     | dishah@email  | disha    | 20   |
| 3     | sahil  | 23     | sahil@email   | sahil    | 20   |
| 4     | ishwar | 24     | ishwar@email  | ishwar   | 20   |
| 5     | pankaj | 25     | pankaj@email  | pankaj   | 20   |
+-------+--------+--------+---------------+----------+------+
5 rows in set (0.00 sec)
```