

Terna Engineering College
Computer Engineering Department

Program: Sem VII

Course: Big Data Analytics & Computational Lab -I (BDA&CL-I)

Experiment No. 03

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per the following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case there is no Blackboard access available)

Roll No. 50	Name: AMEY THAKUR
Class: BE-COMPS-50	Batch: B3
Date of Experiment: 09-08-2021	Date of Submission: 09-08-2021
Grade :	

Aim: To install Sqoop and execute basic commands of Hadoop ecosystem component Sqoop.

B.1. Explain how to install Sqoop.

(Paste your Search material completed during the 2 hours of practical in the lab here)

Sqoop Installation – Steps to Install Sqoop 1.4.5 on Linux

1. Java Installation

Before Sqoop installation – version 1.4.5 on Linux, at very first we needed to have Java installed on our system.

Step 1

By visiting the following link, [download Java](#) (JDK <latest version> – X64.tar.gz). Hence, jdk-7u71-linux-x64.tar.gz will be downloaded on our system.

Step 2

Basically, we find the downloaded Java file in the downloads folder. So, by using the following commands, verify it and extract the jdk-7u71-linux-x64.gz file.

```
$ cd Downloads/
```

```
$ ls
```

```
jdk-7u71-linux-x64.gz
```

```
$ tar xzf jdk-7u71-linux-x64.gz
```

```
$ ls
```

```
jdk1.7.0_71 jdk-7u71-linux-x64.gz
```

Step 3

Then, we have to move it to the location `"/usr/local/"`, to make Java available to all the users. So, open root, and type the following commands.

```
$ su
password:
# mv jdk1.7.0_71 /usr/local/java
# exit
```

Step 4

Afterward, add the following commands to `~/.bashrc` file in order to set up `PATH` and `JAVA_HOME` variables.

```
export JAVA_HOME=/usr/local/java
export PATH=$PATH:$JAVA_HOME/bin
Now apply all the changes to the current running system.
$ source ~/.bashrc
```

Step 5

Now to configure Java alternatives use the following commands

```
# alternatives --install /usr/bin/java java usr/local/java/bin/java 2
# alternatives --install /usr/bin/javac javac usr/local/java/bin/javac 2
# alternatives --install /usr/bin/jar jar usr/local/java/bin/jar 2
# alternatives --set java usr/local/java/bin/java
# alternatives --set javac usr/local/java/bin/javac
# alternatives --set jar usr/local/java/bin/jar
```

So, by using the following command, let's verify Java installation.

```
$ java -version
```

However, we get to see the following response, if Java is already installed on your system

```
java version "1.7.0_71"
```

```
Java(TM) SE Runtime Environment (build 1.7.0_71-b13)
```

```
Java HotSpot(TM) Client VM (build 25.0-b02, mixed mode)
```

2. Hadoop Installation

It is very important that before Sqoop installation, Hadoop is installed on our system. Follow these steps if Hadoop is not installed on your system:

Download Hadoop

By using the following commands, download and extract Hadoop 2.4.1 from Apache Software Foundation.

```
$ su
password:
# cd /usr/local
# wget http://apache.claz.org/hadoop/common/hadoop-2.4.1/
hadoop-2.4.1.tar.gz
# tar xzf hadoop-2.4.1.tar.gz
# mv hadoop-2.4.1/* to hadoop/
# exit
```

So, let's verify the Hadoop installation by using following commands

```
$ hadoop version
```

We will get the following response if Hadoop is already installed on your system
Hadoop 2.4.1

—

Subversion <https://svn.apache.org/repos/asf/hadoop/common> -r 1529768

Compiled by hortonmu on 2013-10-07T06:28Z

Compiled with protoc 2.5.0

From source with checksum 79e53ce7994d1628b240f09af91e1af4

Hence, in this way, we can download Hadoop.

3. Sqoop Download – Sqoop Installation

Basically, we can download the latest version of Sqoop from here: [Download sqoop](#)

For this Sqoop Installation tutorial, we are using version 1.4.5, that is, `sqoop-1.4.5.bin__hadoop-2.0.4-alpha.tar.gz`.

Step 1- Sqoop Installation

However, to extract the Sqoop tarball and move it to “/usr/lib/sqoop” directory we use the following command.

```
$tar -xvf sqoop-1.4.4.bin__hadoop-2.0.4-alpha.tar.gz
```

```
$ su
```

```
password:
```

```
# mv sqoop-1.4.4.bin__hadoop-2.0.4-alpha /usr/lib/sqoop
```

```
#exit
```

Step 2- Configuring bashrc

Also, by appending the following lines to ~/.bashrc file we have to set up the Sqoop environment

```
#Sqoop
```

```
export SQOOP_HOME=/usr/lib/sqoop export PATH=$PATH:$SQOOP_HOME/bin
```

Now, to execute ~/.bashrc file we use the following command.

```
$ source ~/.bashrc
```

Step 3 - Configuring Sqoop

While, we need to edit the `sqoop-env.sh` file, that is placed in the `$SQOOP_HOME/conf` directory, in order to configure Sqoop with Hadoop. Now, using the following command redirect to Sqoop config directory and copy the template file

```
$ cd $SQOOP_HOME/conf
```

```
$ mv sqoop-env-template.sh sqoop-env.sh
```

Also, open `sqoop-env.sh` and edit the following lines

```
export HADOOP_COMMON_HOME=/usr/local/hadoop
```

```
export HADOOP_MAPRED_HOME=/usr/local/hadoop
```

Step 4 - Download and Configure MySQL-connector-java

From the following link, we can [download the MySQL-connector-java-5.1.30.tar.gz](#) file.

In addition, to extract MySQL-connector-java tarball and move MySQL-connector-java-5.1.30-bin.jar to the /usr/lib/sqoop/lib directory we use the following command.

```
$ tar -zxf mysql-connector-java-5.1.30.tar.gz
$ su
password:
# cd mysql-connector-java-5.1.30
# mv mysql-connector-java-5.1.30-bin.jar /usr/lib/sqoop/lib
```

Step 5 - Verifying Sqoop

Finally, to verify the Sqoop version we use the following command.

```
$ cd $SQOOP_HOME/bin
$ sqoop-version
```

Expected output

```
14/12/17 14:52:32 INFO sqoop.Sqoop: Running Sqoop version: 1.4.5
Sqoop 1.4.5 git commit id 5b34accaca7de251fc91161733f906af2eddb83
Compiled by abe on Fri Aug 1 11:19:26 PDT 2014
Hence, in this way Sqoop installation is complete.
```

B.2 Input and Output: (Sqoop Commands)

1. Sqoop – IMPORT Command

Importing a table from a relational database to HDFS is done with the Import command. We'll be importing tables from MySQL databases to HDFS in our example.

We have an employees table in the employees database, which we will import into HDFS, as you can see in the figure below.

```
mysql> use employees;
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> show tables
-> ;
+-----+
| Tables_in_employees |
+-----+
| current_dept_emp     |
| departments          |
| dept_emp             |
| dept_emp_latest_date |
| dept_manager         |
| employees            |
| expected_values      |
| found_values         |
| salaries             |
| tchecksum            |
| titles               |
+-----+
```

The command for importing table is:

```
sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees
```

```
[edureka@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees
```

As you can see in the below image, after executing this command Map tasks will be executed at the back end.

```
17/11/20 17:04:48 INFO mapreduce.Job: Counters: 31
  File System Counters
    FILE: Number of bytes read=0
    FILE: Number of bytes written=620704
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
    HDFS: Number of bytes read=464
    HDFS: Number of bytes written=13821993
    HDFS: Number of read operations=16
    HDFS: Number of large read operations=0
    HDFS: Number of write operations=8
  Job Counters
    Killed map tasks=1
    Launched map tasks=5
    Other local map tasks=5
    Total time spent by all maps in occupied slots (ms)=217032
    Total time spent by all reduces in occupied slots (ms)=0
    Total time spent by all map tasks (ms)=217032
    Total vcore-milliseconds taken by all map tasks=217032
    Total megabyte-milliseconds taken by all map tasks=222240768
  Map-Reduce Framework
    Map input records=300024
```

After the code is executed, you can check the Web UI of HDFS i.e. localhost:50070 where the data is imported.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	edureka	supergroup	0 B	Nov 20 17:04	1	128 MB	_SUCCESS
-rw-r--r--	edureka	supergroup	4.34 MB	Nov 20 17:04	1	128 MB	part-m-00000
-rw-r--r--	edureka	supergroup	2.43 MB	Nov 20 17:04	1	128 MB	part-m-00001
-rw-r--r--	edureka	supergroup	1.99 MB	Nov 20 17:04	1	128 MB	part-m-00002
-rw-r--r--	edureka	supergroup	4.42 MB	Nov 20 17:04	1	128 MB	part-m-00003

Showing 1 to 5 of 5 entries

```

200000,1960-01-11,Selwyn,Koshiba,M,1987-06-05
200001,1957-09-10,Bedrich,Markovitch,M,1985-11-22
200002,1961-02-07,Pascal,Benzmuller,F,1986-03-12
200003,1958-10-14,Arvind,Dechter,M,1987-09-29
200004,1960-09-03,Masaru,Rusmann,M,1985-02-18
200005,1961-12-24,Geoffry,Sherertz,M,1986-07-22
200006,1960-11-25,Jacopo,Marchesini,M,1985-05-15
200007,1957-09-11,Danae,Rouquie,F,1998-04-21
200008,1953-08-05,Zeydy,Hutter,F,1986-10-10
200009,1954-03-17,Gererd,Capobianchi,F,1989-07-12
200010,1954-06-20,Petter,Pelz,F,1994-11-14
200011,1960-11-25,Shahar,Baranowski,F,1992-09-10
200012,1953-11-09,Kwun,Poujol,F,1989-05-17
200013,1953-03-24,Kannan,Czap,M,1993-09-03
200014,1961-05-23,Rildo,Georg,M,1988-10-01
200015,1958-07-09,Laurentiu,Jervis,F,1988-09-30
200016,1961-12-24,Baocai,Hofman,M,1996-03-20
200017,1958-03-09,Alselm,Arlazarov,F,1992-12-05
200018,1963-01-09,Adit,Butner,M,1989-06-07
200019,1955-02-25,Sham,Danley,F,1990-10-15
200020,1956-12-12,Filipe,Mungall,F,1989-01-01
200021,1961-01-30,Shuzo,Bratten,F,1995-07-11
200022,1961-09-11,Ortrud,Menhardt,M,1995-02-05
200023,1958-05-05,Shen,Colorni,F,1985-03-20
200024,1962-12-15,Subhada,Mansanne,M,1995-07-23
200025,1958-07-13,Alper,Luit,M,1989-01-11
200026,1963-08-21,Denis,Varman,F,1988-06-27

```

2. Sqoop – IMPORT Command with target directory

You can also import the table in a specific directory in HDFS using the below command:

```
sqoop import --connect jdbc:mysql://localhost/employees --username edureka
--table employees --m 1 --target-dir /employees
```

Sqoop imports data in parallel from most database sources. -m property is used to specify the number of mappers to be executed.

Sqoop imports data in parallel from most database sources. You can specify the number of map tasks (parallel processes) to use to perform the import by using the -m or -num-mappers argument. Each of these arguments takes an integer value which corresponds to the degree of parallelism to employ.

You can control the number of mappers independently from the number of files present in the directory. Export performance depends on the degree of parallelism. By default, Sqoop will use four tasks in parallel for the export process. This may not be optimal, you will need to experiment with your own particular setup. Additional tasks may offer better concurrency, but if the database is already bottlenecked on updating indices, invoking triggers, and so on, then additional load may decrease performance.

```
[edureka@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees --m 1 --target-dir /employees
```

You can see in the below image, that the number of mapper tasks is 1.

```
Job Counters
  Launched map tasks=1
  Other local map tasks=1
  Total time spent by all maps in occupied slots (ms)=17172
  Total time spent by all reduces in occupied slots (ms)=0
  Total time spent by all map tasks (ms)=17172
  Total vcore-milliseconds taken by all map tasks=17172
  Total megabyte-milliseconds taken by all map tasks=17584128
```

The number of files that are created while importing MySQL tables is equal to the number of mappers created.

Browse Directory

/employees

Show entries Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
-rw-r--r--	edureka	supergroup	0 B	Nov 20 19:12	1	128 MB	_SUCCESS	
-rw-r--r--	edureka	supergroup	13.18 MB	Nov 20 19:12	1	128 MB	part-m-00000	

Showing 1 to 2 of 2 entries

Hadoop, 2017.

3. Sqoop – IMPORT Command with Where Clause

You can import a subset of a table using the 'where' clause in Sqoop import tool. It executes the corresponding SQL query in the respective database server and stores the result in a target directory in HDFS. You can use the following command to import data with 'where' clause:

```
sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees --m 3 --where "emp_no > 49000" --target-dir /Latest_Employees
```

```
[edureka@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/employees --username edureka --table employees --m 3 --where "emp_no > 49000" --target-dir /Latest_Employees
```

Browse Directory											
<input type="text" value="/Latest_Employees"/>										<input type="button" value="Go!"/>	
Show	25	entries		Search: <input type="text"/>							
Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name				
-rw-r--r--	edureka	supergroup	0 B	Nov 20 21:05	1	128 MB	_SUCCESS				
-rw-r--r--	edureka	supergroup	2.65 MB	Nov 20 21:05	1	128 MB	part-m-00000				
-rw-r--r--	edureka	supergroup	4.42 MB	Nov 20 21:05	1	128 MB	part-m-00001				
-rw-r--r--	edureka	supergroup	4.42 MB	Nov 20 21:05	1	128 MB	part-m-00002				
Showing 1 to 4 of 4 entries										<input type="button" value="Previous"/> <input checked="" type="button" value="1"/> <input type="button" value="Next"/>	

4. Sqoop – Incremental Import

Sqoop provides an incremental import mode which can be used to retrieve only rows newer than some previously-imported set of rows. Sqoop supports two types of incremental imports: append and lastmodified. You can use the `--incremental` argument to specify the type of incremental import to perform.

You should specify append mode when importing a table where new rows are continually being added with increasing row id values. You specify the column containing the row's id with `--check-column`. Sqoop imports rows where the check column has a value greater than the one specified with `--last-value`.

An alternate table update strategy supported by Sqoop is called lastmodified mode. You should use this when rows of the source table may be updated, and each such update will set the value of a last-modified column to the current timestamp.

When running a subsequent import, you should specify `--last-value` in this way to ensure you import only the new or updated data. This is handled automatically by creating an incremental import as a saved job, which is the preferred mechanism for performing a recurring incremental import.

First, we are inserting a new row which will be updated in our HDFS.

```
mysql> INSERT INTO employees VALUES (500000, '1994-04-22', 'Avi', 'Anish', 'M', '2013-08-10');
Query OK, 1 row affected (0.07 sec)
```

The command for incremental import is:

```
sqoop import --connect jdbc:mysql://localhost/employees --username edureka
--table employees --target-dir /Latest_Employees --incremental append
--check-column emp_no --last-value 499999
```



```
[edureka@localhost ~]$ sqoop import --connect jdbc:mysql://localhost/employees -  
-username edureka --table employees --target-dir /Latest_Employees --incremental  
append --check-column emp_no --last-value 499999
```

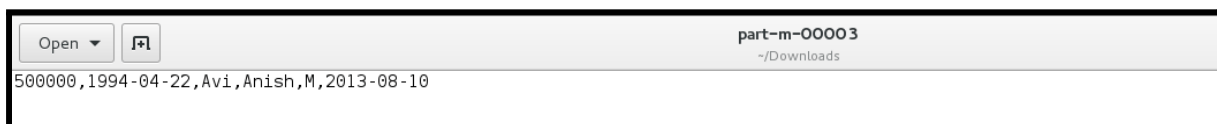
Job Counters

```
Launched map tasks=1  
Other local map tasks=1  
Total time spent by all maps in occupied slots (ms)=9793  
Total time spent by all reduces in occupied slots (ms)=0  
Total time spent by all map tasks (ms)=9793  
Total vcore-milliseconds taken by all map tasks=9793  
Total megabyte-milliseconds taken by all map tasks=10028032
```

Map-Reduce Framework

```
Map input records=1  
Map output records=1  
Input split bytes=117  
Spilled Records=0  
Failed Shuffles=0  
Merged Map outputs=0  
GC time elapsed (ms)=107  
CPU time spent (ms)=2200  
Physical memory (bytes) snapshot=144601088  
Virtual memory (bytes) snapshot=2084610048  
Total committed heap usage (bytes)=125763584
```

You can see in the below image, a new file is created with the updated data.



5. Sqoop - Import All Tables

You can import all the tables from the RDBMS database server to the HDFS. Each table data is stored in a separate directory and the directory name is the same as the table name. It is mandatory that every table in that database must have a primary key field. The command for importing all the table from a database is:
sqoop import-all-tables --connect jdbc:mysql://localhost/employees --username edureka

```
[edureka@localhost ~]$ sqoop import-all-tables --connect jdbc:mysql://localhost/  
employees --username edureka
```

Browse Directory

Show 25 entries
 Search:

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 21:39	0	0 B	_sqoop	🗑️
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 22:36	0	0 B	departments	🗑️
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 22:37	0	0 B	dept_emp	🗑️
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 22:38	0	0 B	dept_manager	🗑️
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 22:40	0	0 B	employees	🗑️
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 22:40	0	0 B	expected_values	🗑️
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 22:41	0	0 B	found_values	🗑️
drwxr-xr-x	edureka	supergroup	0 B	Nov 20 22:43	0	0 B	salaries	🗑️

Showing 1 to 8 of 8 entries

1

6. Sqoop – List Databases

You can list out the databases present in relation databases using Sqoop. Sqoop list-databases tool parses and executes the 'SHOW DATABASES' query against the database server. The command for listing databases is:

```
sqoop list-databases --connect jdbc:mysql://localhost/ --username edureka
```

```
[edureka@localhost ~]$ sqoop list-databases --connect jdbc:mysql://localhost/ --username edureka
```

```
information_schema
employees
```

7. Sqoop – List Tables

You can also list out the tables of a particular database in MySQL database server using Sqoop. Sqoop list-tables tool parses and executes the 'SHOW TABLES' query.

The command for listing tables is a database is:

```
sqoop list-tables --connect jdbc:mysql://localhost/employees --username edureka
```

```
[edureka@localhost ~]$ sqoop list-tables --connect jdbc:mysql://localhost/employees --username edureka
```

```
departments
dept_emp
dept_manager
employees
expected_values
found_values
salaries
```

8. Sqoop – Export

As we discussed above, you can also export data from the HDFS to the RDBMS database. The target table must exist in the target database. The data is stored as records in HDFS. These records are read and parsed and delimited with user-specified delimiter. The default operation is to insert all the records from the input files to the database table using the INSERT statement. In update mode, Sqoop generates the UPDATE statement that replaces the existing record into the database.

So, first we are creating an empty table, where we will export our data.

```
mysql> CREATE TABLE emp(id INT NOT NULL PRIMARY KEY, birth_date date, f_name varchar(14), l_name varchar(16), gender varchar(3), hire_date date);
Query OK, 0 rows affected (0.10 sec)
```

The command to export data from HDFS to the relational database is:

```
sqoop export --connect jdbc:mysql://localhost/employees --username edureka
--table emp --export-dir /user/edureka/employees
```

```
[edureka@localhost ~]$ sqoop export --connect jdbc:mysql://localhost/employees -
-username edureka --table emp --export-dir /user/edureka/employees
```

```
mysql> select count(*) from emp;
+-----+
| count(*) |
+-----+
|    300025 |
+-----+
1 row in set (0.13 sec)
```

9. Sqoop – Codegen

In object-oriented applications, every database table has one Data Access Object class that contains 'getter' and 'setter' methods to initialize objects. Codegen generates the DAO class automatically. It generates a DAO class in Java, based on the Table Schema structure.

The command for generating java code is:

```
sqoop codegen --connect jdbc:mysql://localhost/employees --username edureka
--table employees
```

```
[edureka@localhost ~]$ sqoop codegen --connect jdbc:mysql://localhost/employees
--username edureka --table employees
```

```

17/11/20 23:39:55 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
17/11/20 23:39:55 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employees` AS t LIMIT 1
17/11/20 23:39:55 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-2.8.1
Note: /tmp/sqoop-edureka/compile/4ff615554b18eb98adc7d5e88a711f06/employees.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
17/11/20 23:39:59 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-edureka/compile/4ff615554b18eb98adc7d5e88a711f06/employees.jar

```

You can see the path in the above image where the code is generated. Let us go the path and check the files that are created.

```

[edureka@localhost ~]$ cd /tmp/sqoop-edureka/compile/4ff615554b18eb98adc7d5e88a711f06
[edureka@localhost 4ff615554b18eb98adc7d5e88a711f06]$ ls
employees.class  employees.jar  employees.java
[edureka@localhost 4ff615554b18eb98adc7d5e88a711f06]$

```

B.3 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)

We are able to acquire fundamental enabling techniques and scalable algorithms like Hadoop, Map Reduce and NO SQL in big data analytics.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

Hence, we can install Sqoop and execute basic commands of Hadoop ecosystem component Sqoop.

B.5 Question of Curiosity

(To be answered by student based on the practical performed and learning/observations)

Q1) What is the default file format to import data using Apache Sqoop?

Ans:

The default file type is a text file format. It is the same as specifying `-as-textfile` clause to sqoop import command.

Q2) How will you list all the columns of a table using Apache Sqoop?

Ans:

There is no straight way to list all the columns of a table in Apache Sqoop like `sqoop-list-columns`, so first we should retrieve the columns of the particular table and transform to a file containing the column names of the particular table.

Q3) Name a few import control commands. How can Sqoop handle large objects?

Ans:

To import RDBMS data, we use import control commands

Append: Append data to an existing dataset in HDFS.

-append

Columns: columns to import from the table.

-columns

<col,col.....>

Where: where clause to use during import. —

Where the common large objects are Blob and Clob. Suppose the object is less than 16 MB, it is stored inline with the rest of the data. If there are big objects, they are temporarily stored in a subdirectory with the name _lob. Those data are then materialized in memory for processing. If we set the lob limit as ZERO (0) then it is stored in external memory.

Q4) How can we import data from a particular row or column? What are the destination types allowed in the Sqoop import command?

Ans:

Basically, on the basis of where clause, Sqoop allows to Export and Import the data from the data table. So, the syntax is

-columns

<col1,col2.....> -where

-query

For Example:

```
sqoop import -connect jdbc:mysql://db.one.com/corp -table INTELLIPAAT_EMP  
-where "start_date>'2016-07-20'"
```

```
sqoopeval -connect jdbc:mysql://db.test.com/corp -query "SELECT * FROM  
intellipaat_emp LIMIT 20"
```

```
sqoop import -connect jdbc:mysql://localhost/database -username root  
-password aaaaa -columns "name,emp_id,jobtitle"
```

However, into following services Sqoop supports data imported:

1. HDFS
2. Hive
3. Hbase
4. Hcatalog
5. Accumulo

Q5) What is the process to perform an incremental data load in Sqoop?

Ans:

In Sqoop, the process to perform incremental data load is to synchronize the modified or updated data (often referred as delta data) from RDBMS to Hadoop. Moreover, in Sqoop the delta data can be facilitated through the incremental load command.

In addition, by using Sqoop import command we can perform incremental load. Also, by loading the data into the hive without overwriting it. However, in Sqoop the different attributes that need to be specified during incremental load are

1) Mode (incremental)

It shows how Sqoop will determine what the new rows are. Also, it has value as Append or Last Modified.

2) Col (Check-column)

Basically, it specifies the column that should be examined to find out the rows to be imported.

3) Value (last-value)

It denotes the maximum value of the check column from the previous import operation.