

**SQOOP**

—----------------------------------------------------------------------------------------------------------------------------

As RDBMS, we will be using MySQL database.

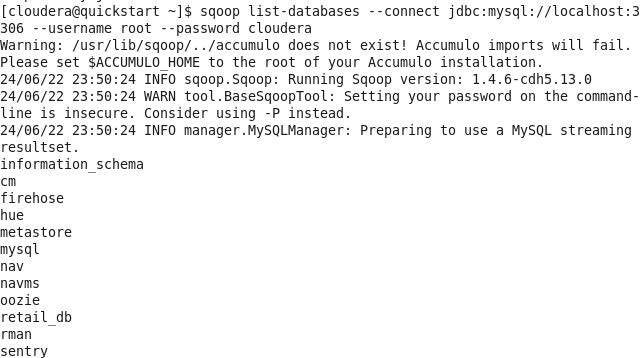
**You can connect with MySQL database through the commands:**

mysql -uroot -pcloudera

—----------------------------------------------------------------------------------------------------------------------------

**To list the databases in SQOOP:**

sqoop list-databases --connect jdbc:mysql://localhost:3306 --username root --password cloudera



—----------------------------------------------------------------------------------------------------------------------------

**Points to Remember:**

When we import the SQOOP:

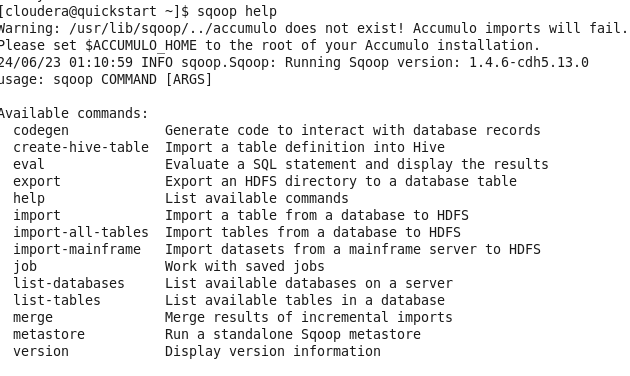
1. Data is selected with Select command
2. Min and Max query is applied
3. Default no of splits : 4
4. Mapper and Reduce tasks gets executed

By Default : It will import the data in the user's home directory.

Eg: /user/cloudera

—----------------------------------------------------------------------------------------------------------------------------

All SQOOP commands:



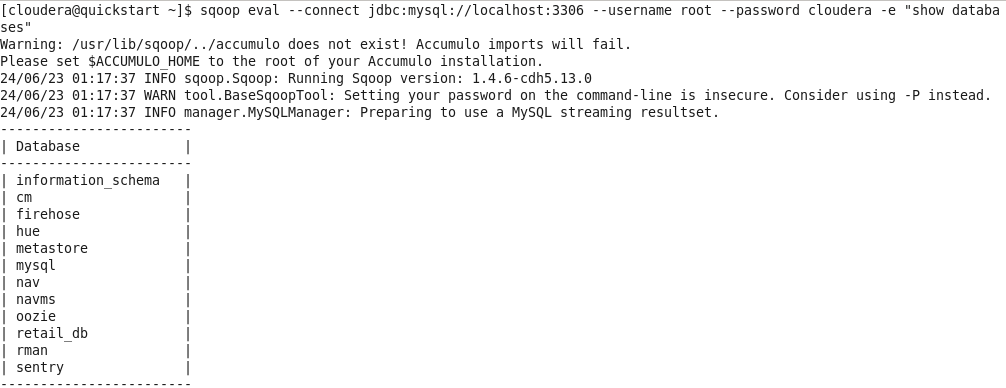
—----------------------------------------------------------------------------------------------------------------------------

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

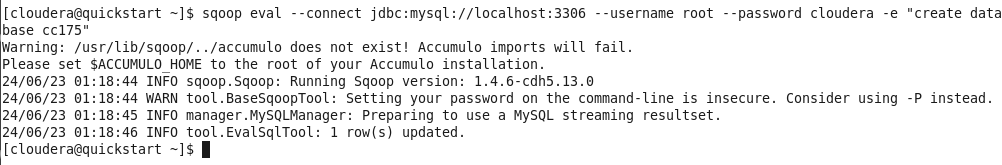
**SQOOP EVAL**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

sqoop eval --connect jdbc:mysql://localhost:3306 --username root --password cloudera -e "show databases"



sqoop eval --connect jdbc:mysql://localhost:3306 --username root --password cloudera -e "create database cc175"

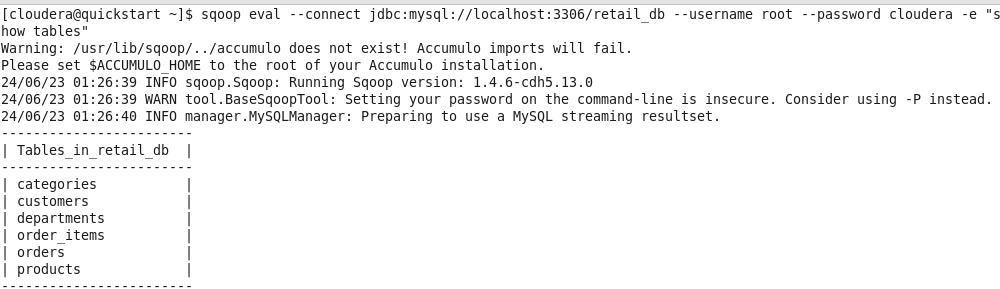


Note 👏

It is not mandatory to start the Hadoop services, except for importing the data.

—------------------------------------------------------------------------------------------------------------------------

sqoop eval --connect jdbc:mysql://localhost:3306/retail\_db --username root --password cloudera -e "show tables"



—---------------------------------------------------------------------------------------------------------------------------

**To store the connection username and password:**

Approach 1:

We can create the files in which have the information such as below:

eval

--connect

jdbc:mysql://localhost:3306/retail\_db

--username

root

--password

cloudera

And then can run the below commands:

sqoop --options-file /home/cloudera/sqoop-script/eval-connection.txt -e "show databases"



Approach 2: (Manually asks to type the password.)

sqoop import \

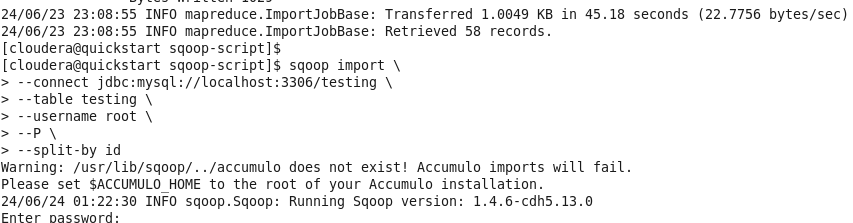
--connect jdbc:mysql://localhost:3306/testing \

--table testing \

--username root \

--P \

--split-by id



—----------------------------------------------------------------------------------------------------------------------------

Approach 3: Creating a password file

echo -n “cloudera” > sqoop.pwd

sqoop import \

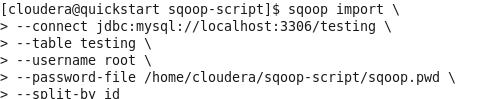
--connect jdbc:mysql://localhost:3306/testing \

--table testing \

--username root \

--password-file file:///home/cloudera/sqoop-script/sqoop.pwd \

--split-by id



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

SQOOP IMPORT

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Boundary Query:

Boundary query is used to increase the performance.

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

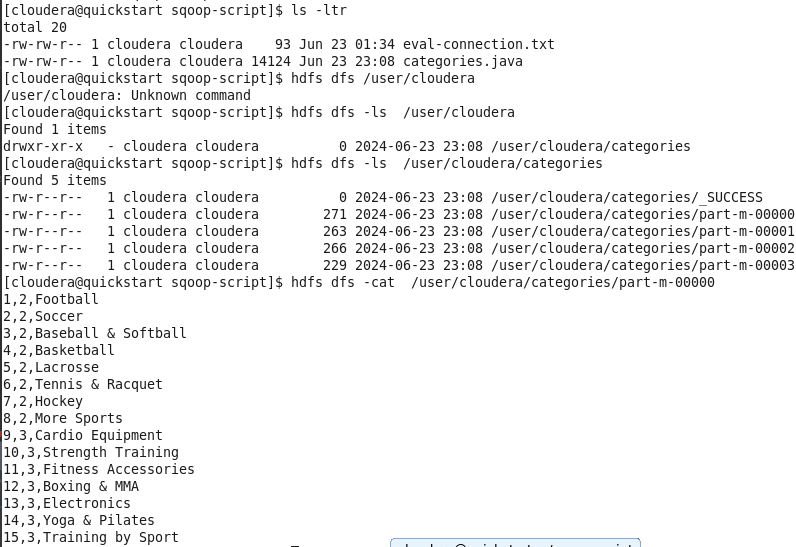
--password cloudera \

--table categories \

--split-by category\_id \

--boundary-query "select min(category\_id), max(category\_id) from categories"





—----------------------------------------------------------------------------------------------------------------------------

**Speed up Transfers: Direct Import:**

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table categories \

--direct

Note:

1. Sqoop can only perform --direct mode imports from PostgreSQL,Oracle and Netezza.
2. Binary formats like sequence file or Avro wont work with direct mode import.
3. In case of MySQL, when using --direct parameters, sqoop will takes advantages of MySQL native utility like mysqldump and mysqlimport, rather than using JDBC interface for transferring data.

—----------------------------------------------------------------------------------------------------------------------------

**Target dir import:**

sqoop import \

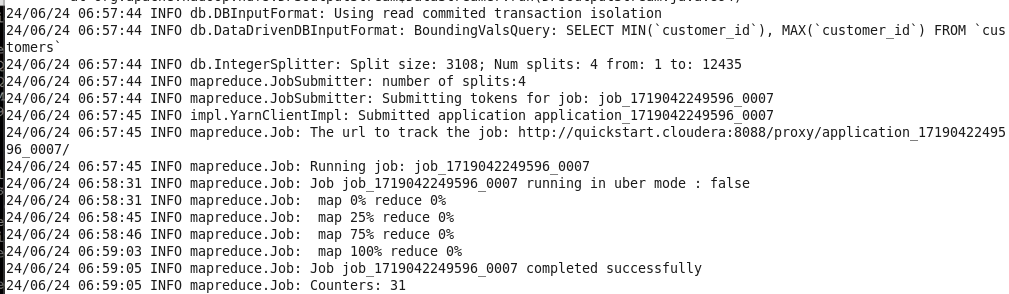
--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table customers \

--target-dir /user/cloudera/customers



—--------------------------------------------------------------------------------------------------------------------------

**Delete the target dir and overwrite it:**

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

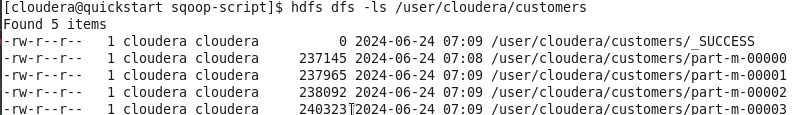
--username root \

--P \

--table customers \

--delete-target-dir \

--target-dir /user/cloudera/customers



—----------------------------------------------------------------------------------------------------------------------------

Append data to an existing directory:

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

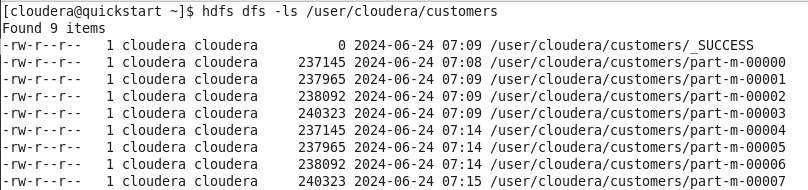
--username root \

--P \

--table customers \

--target-dir /user/cloudera/customers \

--append



—----------------------------------------------------------------------------------------------------------------------------

**Importing the data inside parent directory:**

sqoop import \

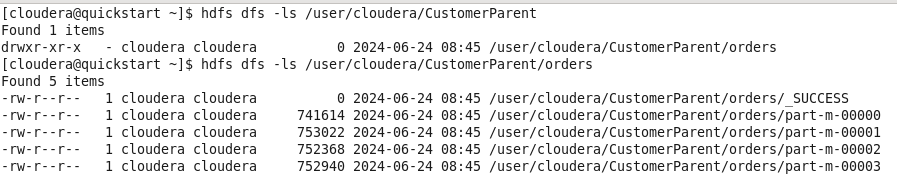
--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table orders \

--warehouse-dir /user/cloudera/CustomerParent



—----------------------------------------------------------------------------------------------------------------------------

**Import all rows of a table from MySQL, but specific columns of table:**

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

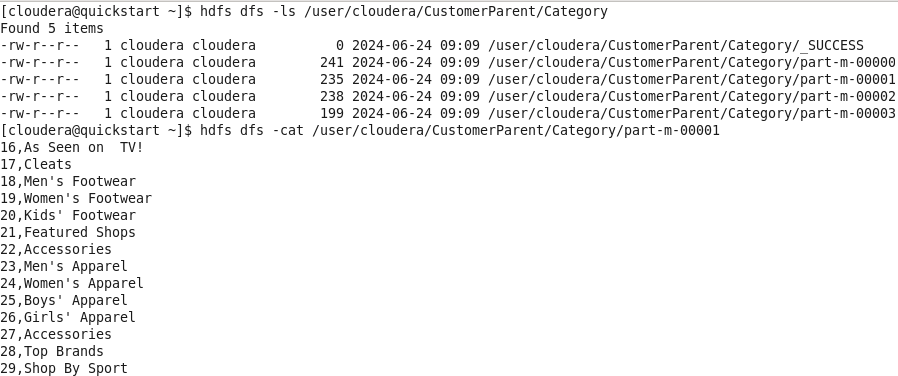
--table categories \

--username root \

--P \

--target-dir /user/cloudera/CustomerParent \

--columns "category\_id,category\_name"



—------------------------------------------------------------------------------------------------------------------------

**Use WHERE clause:**

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--table categories \

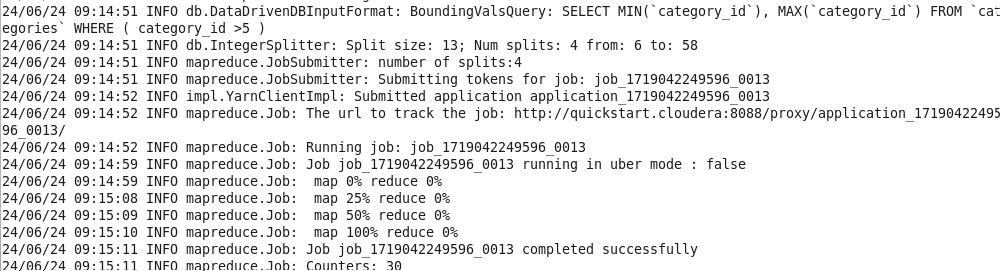
--username root \

--P \

--target-dir /user/cloudera/CustomerParent1 \

--columns "category\_id,category\_name" \

--where "category\_id >5"



—----------------------------------------------------------------------------------------------------------------------------

**Import all tables of MySQL DB into HDFS 🙂**

--target-dir parameter is not allowed

--warehouse-dir parameter is allowed

Note:

1. Each table must have a single column primary key or --autoreset-to-one-mapper option must be used.
2. We must intend to import all columns of each table

(it means we cannot use --columns )

sqoop import-all-tables \

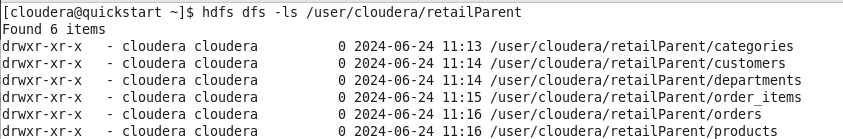
--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--warehouse-dir /user/cloudera/retailParent \

--autoreset-to-one-mapper



sqoop import-all-tables \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--warehouse-dir /user/cloudera/retailParent\_exclude \

--exclude-tables "departments"

—----------------------------------------------------------------------------------------------------------------------

**Compressing the imported data:**

Syntax 1: gzip (By default)

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

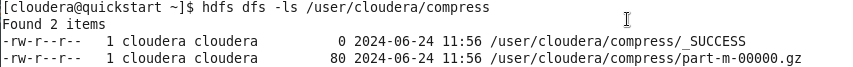
--P \

--table "departments" \

--m 1 \

--compress \

--target-dir /user/cloudera/compress



Syntax 2: Bzip2

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

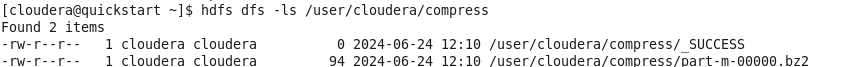
--P \

--table departments \

--m 1 \

--compress --compression-codec org.apache.hadoop.io.compress.BZip2Codec \

--target-dir /user/cloudera/compress



—---------------------------------------------------------------------------------------------------------------------------

**Import MySQL data into HDFS in various file format:**

Syntax 1: (SEQUENCE file format)

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table departments \

--m 1 \

--target-dir /user/cloudera/compress \

--as-sequencefile

Syntax 2: (AVRO DATA FILE)

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

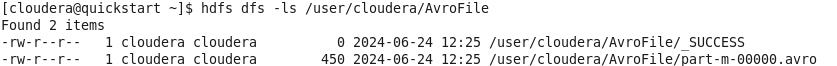
--P \

--table departments \

--m 1 \

--target-dir /user/cloudera/AvroFile \

--as-avrodatafile



—----------------------------------------------------------------------------------------------------------------------------

**Delimiter 👍**

Note:

1. The file format is by default textfile
2. By default delimiter is taken as ‘,’

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

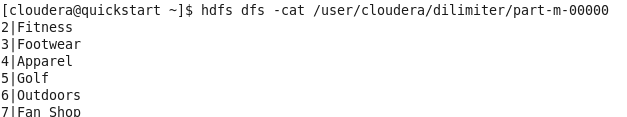
--P \

--table departments \

--m 1 \

--target-dir /user/cloudera/dilimiter \

--fields-terminated-by '|'



—--------------------------------------------------------------------------------------------------------------------------

**Incremental Import:**

create table inc\_imp (id int, name varchar(15, city varchar(15 ));

—------------------------------------------------------------------------------------------

INSERT INTO inc\_imp VALUES (1, 'Komal', 'Mumbai');

INSERT INTO inc\_imp VALUES (2, 'Komi', 'Ireland');

INSERT INTO inc\_imp VALUES (3, 'Tadano', 'Ireland');

—------------------------------------------------------------------------------------------

sqoop import \

--connect jdbc:mysql://localhost:3306/testing \

--username root \

--P \

--table inc\_imp \

--m 1 \

--target-dir /user/cloudera/incremental\_import

—-------------------------------------------------------------------------------------------

INSERT INTO inc\_imp VALUES (4, 'Kom', 'Mumbai');

INSERT INTO inc\_imp VALUES (5, 'Komi-san', 'Ireland');

—-------------------------------------------------------------------------------------------

(Hard-coded values 👍)

sqoop import \

--connect jdbc:mysql://localhost:3306/testing \

--username root \

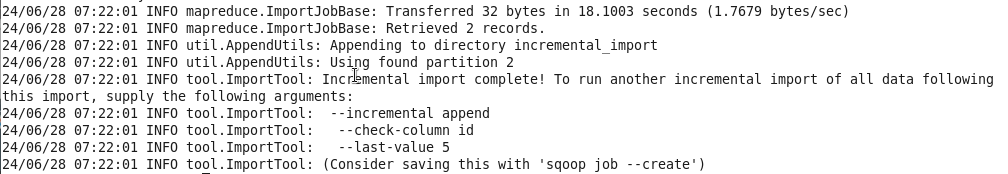
--P \

--table inc\_imp \

--m 1 \

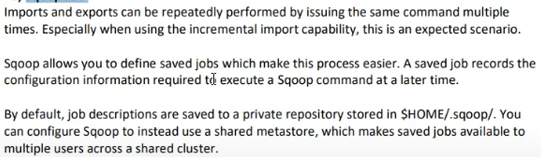
--target-dir /user/cloudera/incremental\_import \

--incremental append --check-column id --last-value 3



—----------------------------------------------------------------------------------------------------------------------------

**Sqoop Job:**



sqoop job --create inc\_imp\_id2 \

-- import --connect jdbc:mysql://localhost:3306/testing \

--username root \

--P \

--table inc\_imp \

--m 1 \

--target-dir /user/cloudera/incremental\_import\_id2 \

--incremental append --check-column id --last-value 0

—-----------------------------------------------------------------------------------------------------------

sqoop job --exec inc\_imp\_id2

—----------------------------------------------------------------------------------------------------------

sqoop job --show inc\_imp\_id | grep ‘incremental.last.value’

—---------------------------------------------------------------------------------------------------------

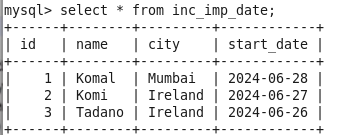
Incremental import by date:

create table inc\_imp\_date (id int, name varchar(15), city varchar(15 ),start\_date date);

INSERT INTO inc\_imp VALUES (1, 'Komal', 'Mumbai',now()-interval 1 day);

INSERT INTO inc\_imp VALUES (2, 'Komi', 'Ireland', now()- interval 2 day);

INSERT INTO inc\_imp VALUES (3, 'Tadano', 'Ireland', now()-interval 3 day);



sqoop job --create inc\_imp\_dt \

-- import --connect jdbc:mysql://localhost:3306/testing \

--username root \

--P \

--table inc\_imp\_date \

--m 1 \

--target-dir /user/cloudera/incremental\_import\_id\_date \

--incremental append --check-column start\_date --last-value 0000-00-00

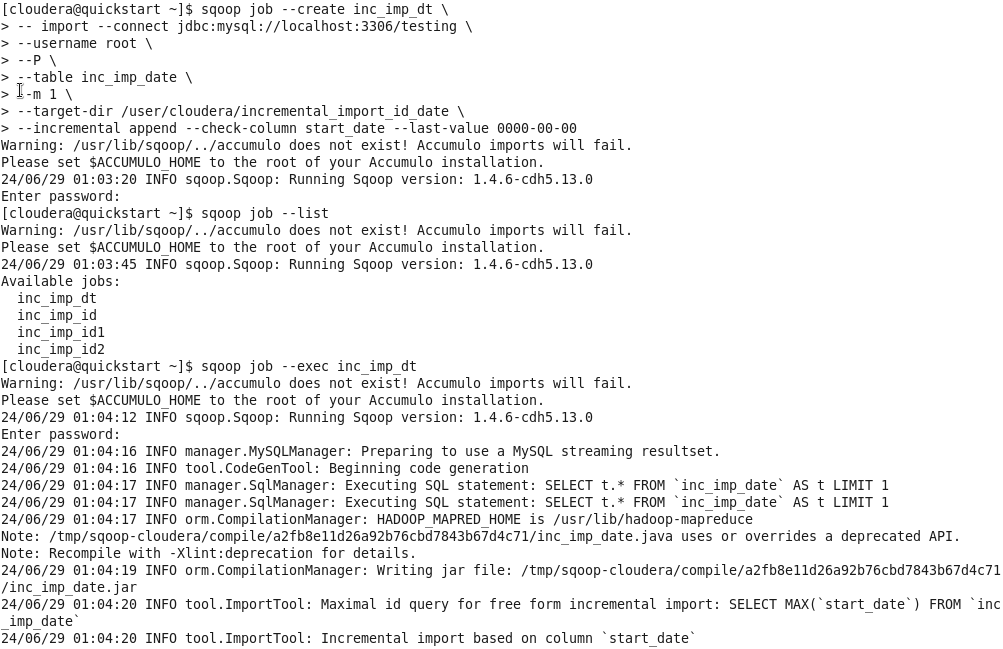
—---------------------------------------------------------------

sqoop job --list

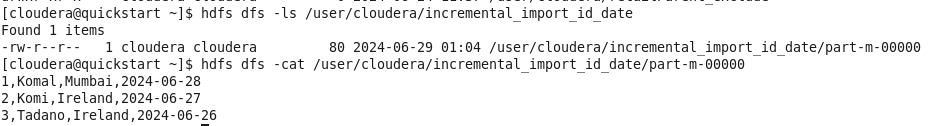
—--------------------------------------------------------------

sqoop job --exec inc\_imp\_dt

—--------------------------------------------------------------



—----------------------------------------------------------------------------------------------------------------------------



—----------------------------------------------------------------------------------------------------------------------------

Add 2 new row:

INSERT INTO inc\_imp\_date VALUES (4, 'Kom', 'Mumbai', now());

INSERT INTO inc\_imp\_date VALUES (5, 'Komi-san', 'Ireland',now());

—------------------------------------------------------------------------------

sqoop job --exec inc\_imp\_dt

—-------------------------------------------------------------------------------------------------------

24/06/29 01:15:32 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/8b9e98d080ae505666e98727f95a6531/inc\_imp\_date.jar

24/06/29 01:15:34 INFO tool.ImportTool: Maximal id query for free form incremental import: SELECT MAX(`start\_date`) FROM `inc\_imp\_date`

24/06/29 01:15:34 INFO tool.ImportTool: Incremental import based on column `start\_date`

24/06/29 01:15:34 INFO tool.ImportTool: Lower bound value: '2024-06-28'

24/06/29 01:15:34 INFO tool.ImportTool: Upper bound value: '2024-06-29'

24/06/29 01:15:34 WARN manager.MySQLManager: It looks like you are importing from mysql.

24/06/29 01:15:34 WARN manager.MySQLManager: This transfer can be faster! Use the --direct

24/06/29 01:15:34 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.

24/06/29 01:15:34 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)

24/06/29 01:15:34 INFO mapreduce.ImportJobBase: Beginning import of inc\_imp\_date

24/06/29 01:15:34 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address

24/06/29 01:15:34 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar

24/06/29 01:15:34 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps

24/06/29 01:15:34 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

24/06/29 01:16:55 INFO db.DBInputFormat: Using read commited transaction isolation

24/06/29 01:16:55 INFO mapreduce.JobSubmitter: number of splits:1

24/06/29 01:16:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1719042249596\_0041

24/06/29 01:16:56 INFO impl.YarnClientImpl: Submitted application application\_1719042249596\_0041

24/06/29 01:16:56 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application\_1719042249596\_0041/

24/06/29 01:16:56 INFO mapreduce.Job: Running job: job\_1719042249596\_0041

24/06/29 01:17:04 INFO mapreduce.Job: Job job\_1719042249596\_0041 running in uber mode : false

24/06/29 01:17:04 INFO mapreduce.Job: map 0% reduce 0%

24/06/29 01:17:10 INFO mapreduce.Job: map 100% reduce 0%

24/06/29 01:17:11 INFO mapreduce.Job: Job job\_1719042249596\_0041 completed successfully

24/06/29 01:17:11 INFO mapreduce.Job: Counters: 30

—----------------------------------------------------------------------------------------------------------------------------

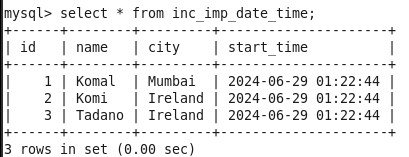
**Incremental import lastmodified 😣**

create table inc\_imp\_date\_time (id int, name varchar(15), city varchar(15 ),start\_time timestamp);

INSERT INTO inc\_imp\_date\_time VALUES (1, 'Komal', 'Mumbai',now());

INSERT INTO inc\_imp\_date\_time VALUES (2, 'Komi', 'Ireland', now());

INSERT INTO inc\_imp\_date\_time VALUES (3, 'Tadano', 'Ireland', now());



—---------------------------------------------------------------------------------------------------------------------------

sqoop job --create inc\_imp\_dt\_time \

-- import --connect jdbc:mysql://localhost:3306/testing \

--username root \

--P \

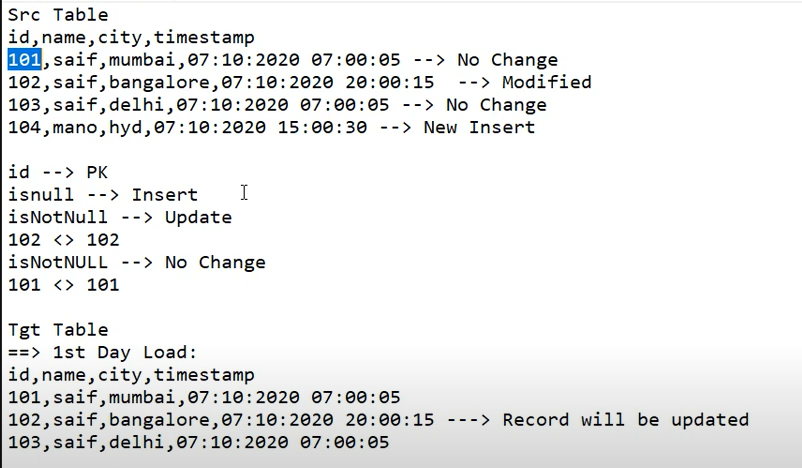
--table inc\_imp\_date\_time \

--m 1 \

--target-dir /user/cloudera/incremental\_import\_id\_date\_time \

--incremental lastmodified --check-column start\_time --last-value "0000-00-00 00:00:00" --merge-key id

Concept explanation:



Note:

24/06/29 01:40:49 INFO mapreduce.Job: map 0% reduce 0%

24/06/29 01:40:55 INFO mapreduce.Job: map 100% reduce 0%

24/06/29 01:40:56 INFO mapreduce.Job: Job job\_1719042249596\_0042 completed successfully

(only mapper is used)

—--------------------------------------------------------------------------------------

UPDATE inc\_imp\_date\_time SET name='Komu', city='Navi-Mumbai' WHERE id=2;

INSERT INTO inc\_imp\_date\_time VALUES (4, 'Komi-san', 'Ireland',now());

—--------------------------------------------------------------------------------------------------------------------

Again Execute the sqoop job:

sqoop job --exec inc\_imp\_dt\_time

24/06/29 01:47:18 INFO mapreduce.Job: map 0% reduce 0%

24/06/29 01:47:27 INFO mapreduce.Job: map 100% reduce 0%

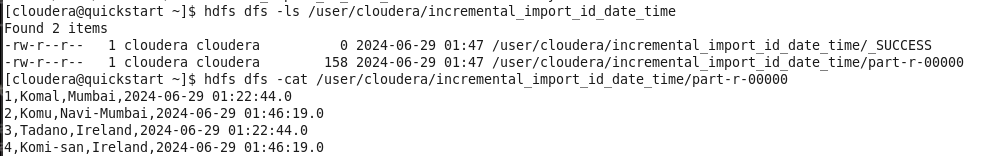
24/06/29 01:47:33 INFO mapreduce.Job: map 100% reduce 100%

24/06/29 01:47:34 INFO mapreduce.Job: Job job\_1719042249596\_0044 completed successfully

(Here we can see the reduce job also gets executed)

Note:

So, in Sqoop, it's the only scenario where the reducer is working.



—---------------------------------------------------------------------------------------------------------------------------

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**SQOOP EXPORT**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table departments \

--m 1 \

--target-dir /user/cloudera/departments

—----------------------------------------------------------------------------------------------------------------------

**Sqoop Export:**

sqoop export \

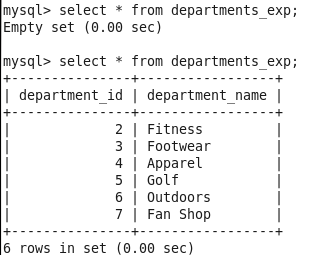
--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table departments\_exp \

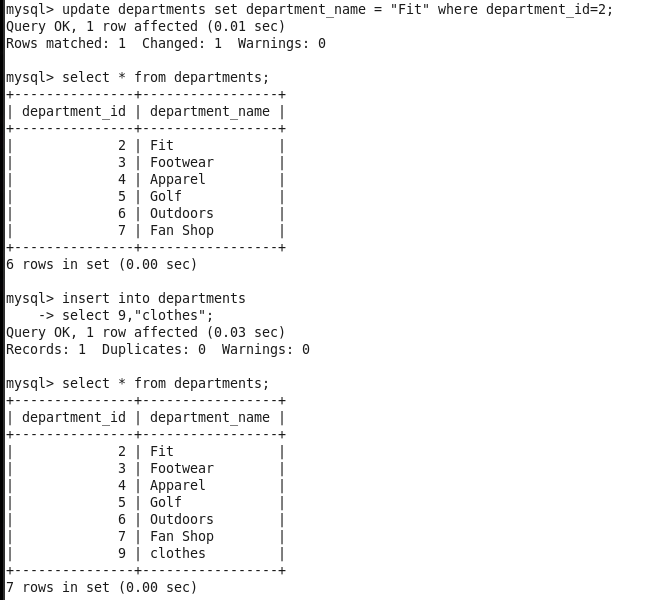
--export-dir /user/cloudera/departments



—--------------------------------------------------------------------------------------------------------------------------

**Sqoop export updateonly:**

It only exports the updated existing value, not the newly inserted value.



sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table departments \

--m 1 \

--delete-target-dir \

--target-dir /user/cloudera/departments

—------------------------------------------------------------------------------

sqoop export \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table departments\_exp \

--export-dir /user/cloudera/departments \

--update-mode updateonly --update-key departement\_id

—-------------------------------------------------------------------------------------------------------------------------

**Sqoop allowinsert 👍**

It boths insert and updates the data.

Note: But it will append the data, not override it.

sqoop export \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

--table departments\_exp \

--export-dir /user/cloudera/departments \

--update-mode allowinsert --update-key department\_id

So, to override it:

You can create the staging table where you can perform update and insert and then can export into main target table.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**ADDITIONAL COMMANDS**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Sqoop --null-string (Handling the null values):**

sqoop import \

--connect jdbc:mysql://localhost:3306/testing \

--username root \

--P \

--table inc\_imp\_date \

--m 1 \

--target-dir /user/cloudera/inc\_imp\_date\_new \

--null-string NA \

--null-non-string 9999

—------------------------------------------------------------------------------------------------------------------------

**Sqoop --map-column-java 👍👍**

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

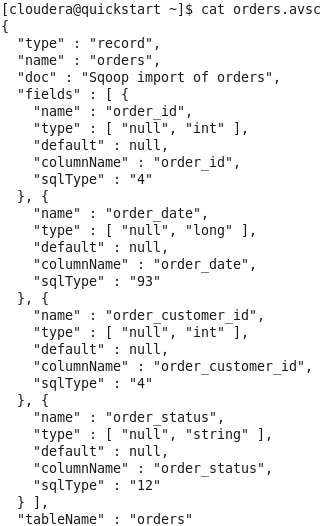
--P \

--table orders \

--m 1 \

--target-dir /user/cloudera/orders\_map \

--as-avrodatafile



—----------------------------------------------------------------------------------------------------------------------------

**Explicitly we need to change the data type of column:**

For above eg, lets change the order\_date column datatype from date to string.

sqoop import \

--connect jdbc:mysql://localhost:3306/retail\_db \

--username root \

--P \

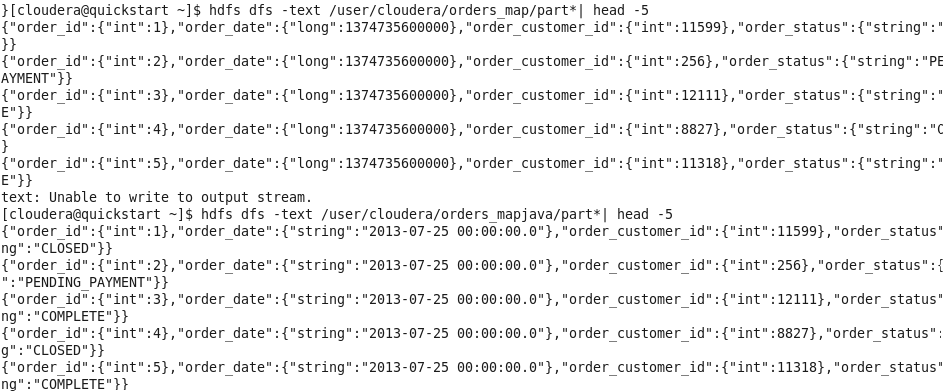
--table orders \

--m 1 \

--target-dir /user/cloudera/orders\_mapjava \

--as-avrodatafile \

--map-column-java order\_date=String



Note always delete that .avsc file or else it will not get overwrite.

—---------------------------------------------------------------------------------------------------------------------------