Assignment Solution

Week3: Apache Sqoop - Moving Data into Hadoop

Assignment-Solutions -Week 3

Total Marks 100

Qu1) Solution:

Create a database test_db

```
mysql> create database test_db;
Query OK, 1 row affected (0.00 sec)

mysql> use test_db;
Database changed
```

Create Customers table with mentioned columns

```
mysql> CREATE TABLE Customers
    -> (
    -> Cust_Id INT NOT NULL,
    -> Customer_Name VARCHAR(255),
    -> Purchase_Date DATE NOT NULL,
    -> Item VARCHAR(255),
    -> City VARCHAR(255),
    -> Price INT PRIMARY KEY,
    -> Cust_Type VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> describe Customers;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Cust_Id       | int(11)     | NO   |     | NULL    |       |
| Customer_Name | varchar(255) | YES  |     | NULL    |       |
| Purchase_Date | date        | NO   |     | NULL    |       |
| Item          | varchar(255) | YES  |     | NULL    |       |
| City          | varchar(255) | YES  |     | NULL    |       |
| Price         | int(11)     | NO   | PRI | NULL    |       |
| Cust_Type     | varchar(255) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
7 rows in set (0.01 sec)
```

Insert records into the Customers table:

```
mysql> INSERT INTO Customers values
    -> (100,'Rishi','2020-08-16','Mobile','Kanpur',10000 ,'Regular'),
    -> (200,'Venu','2019-05-04','Laptop','Bangalore',61000,'Premium'),
    -> (300,'Priya','2018-06-25','Mobile','Jaipur',20000,'Premium'),
    -> (400,'Rini','2019-01-30','Handbag','Pune',1000,'Regular'),
    -> (700,'Deepu','2019-12-12','Appliances','Mumbai',25000,'Premium');
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

```
mysql> SELECT * FROM Customers;
+---------+---------------+---------------+------------+-----------+-------+-----------+
| Cust_Id | Customer_Name | Purchase_Date | Item       | City      | Price | Cust_Type |
+---------+---------------+---------------+------------+-----------+-------+-----------+
|     400 | Rini          | 2019-01-30    | Handbag    | Pune      |  1000 | Regular   |
|     100 | Rishi         | 2020-08-16    | Mobile     | Kanpur    | 10000 | Regular   |
|     300 | Priya         | 2018-06-25    | Mobile     | Jaipur    | 20000 | Premium   |
|     700 | Deepu         | 2019-12-12    | Appliances | Mumbai    | 25000 | Premium   |
|     200 | Venu          | 2019-05-04    | Laptop     | Bangalore | 61000 | Premium   |
+---------+---------------+---------------+------------+-----------+-------+-----------+
5 rows in set (0.00 sec)
```

1)

Command:

```
(base) [cloudera@quickstart ~]$ sqoop-eval \
> --connect jdbc:mysql://quickstart.cloudera:3306/test_db \
> --username root \
> --password cloudera \
> --query "SELECT * FROM Customers " 1>query.output
```

Output:

```
| Cust_Id     | Customer_Name        | Purchase_Date | Item                 | City                 | Price       | Cust_Type            |
| 400         | Rini                 | 2019-01-30    | Handbag              | Pune                 | 1000        | Regular              |
| 100         | Rishi                | 2020-08-16    | Mobile               | Kanpur               | 10000       | Regular              |
| 300         | Priya                | 2018-06-25    | Mobile               | Jaipur               | 20000       | Premium              |
| 700         | Deepu                | 2019-12-12    | Appliances           | Mumbai               | 25000       | Premium              |
| 200         | Venu                 | 2019-05-04    | Laptop               | Bangalore            | 61000       | Premium              |
```

Sqoop Import Command:

Using Normal Sqoop Import, with the default Primary key ‘Price’

```
(base) [cloudera@quickstart ~]$ sqoop-import \
> --connect jdbc:mysql://quickstart.cloudera:3306/test_db \
> --username root \
> --password cloudera \
> --table Customers \
> --columns Cust_Id,Customer_Name,Purchase_Date,Item,City,Price \
> --where "Purchase_Date > '2019-01-01' " \
> --fields-terminated-by '|' \
> --lines-terminated-by ';' \
> --target-dir /user/cloudera/sqoop_importdir 1>log.output 2>log.error
```

Note : To take care of nulls if any in the data we would have used

--null-string “NA”

--null-non-string “NA”

Content of log.error file:

```
(base) [cloudera@quickstart ~]$ cat log.error
                Bytes Written=167
20/05/01 13:37:16 INFO mapreduce.ImportJobBase: Transferred 167 bytes in 34.5661
 seconds (4.8313 bytes/sec)
20/05/01 13:37:16 INFO mapreduce.ImportJobBase: Retrieved 4 records.
```

Content of sqoop_importdir directory

```
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/sqoop_importdir/
Found 5 items
-rw-r--r--   1 cloudera cloudera          0 2020-05-01 13:37 /user/cloudera/sqoo
p_importdir/_SUCCESS
-rw-r--r--   1 cloudera cloudera         79 2020-05-01 13:37 /user/cloudera/sqoo
p_importdir/part-m-00000
-rw-r--r--   1 cloudera cloudera         45 2020-05-01 13:37 /user/cloudera/sqoo
p_importdir/part-m-00001
-rw-r--r--   1 cloudera cloudera          0 2020-05-01 13:37 /user/cloudera/sqoo
p_importdir/part-m-00002
-rw-r--r--   1 cloudera cloudera         43 2020-05-01 13:37 /user/cloudera/sqoo
p_importdir/part-m-00003
```

So for better work distribution among mappers we should use the following sqoop command ,using a split-by clause:

Second Sqoop Import Command:

```
(base) [cloudera@quickstart ~]$ sqoop-import \
> --connect jdbc:mysql://quickstart.cloudera:3306/test_db \
> --username root \
> --password cloudera \
> --table Customers \
> --columns Cust_Id,Customer_Name,Purchase_Date,Item,City,Price \
> --where "Purchase_Date > '2019-01-01' " \
> --split-by Cust_Id \
> --fields-terminated-by '|' \
> --lines-terminated-by ';' \
> --target-dir /user/cloudera/sqoop_importdir 1>log.output 2>log.error \
> --delete-target-dir
```

Output:

```
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/sqoop_importdir/
Found 5 items
-rw-r--r--   1 cloudera cloudera          0 2020-05-01 13:43 /user/cloudera/sqoo
p_importdir/_SUCCESS
-rw-r--r--   1 cloudera cloudera         84 2020-05-01 13:43 /user/cloudera/sqoo
p_importdir/part-m-00000
-rw-r--r--   1 cloudera cloudera          0 2020-05-01 13:43 /user/cloudera/sqoo
p_importdir/part-m-00001
-rw-r--r--   1 cloudera cloudera         38 2020-05-01 13:43 /user/cloudera/sqoo
p_importdir/part-m-00002
-rw-r--r--   1 cloudera cloudera         45 2020-05-01 13:43 /user/cloudera/sqoo
p_importdir/part-m-00003
```

```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/sqoop_importdir/*
| head
100|Rishi|2020-08-16|Mobile|Kanpur|10000;200|Venu|2019-05-04|Laptop|Bangalore|61
000;400|Rini|2019-01-30|Handbag|Pune|1000;700|Deepu|2019-12-12|Appliances|Mumbai
```

Custom Boundary Query for dealing with Outliers.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/test_db \
--username root \
--password cloudera \
--table Customers \
--columns Cust_Id,Customer_Name,Purchase_Date,Item,City,Price \
--where "Purchase_Date > '2019-01-01' " \
--split-by Cust_Id \
--boundary-query "SELECT 100, 700" \
--fields-terminated-by '|' \
--lines-terminated-by ';' \
--target-dir /user/cloudera/sqoop_importdir 1>log.output 2>log.error \
--delete-target-dir
```

Note: We will hardcode the min and max values of the split-by column while mentioning custom bound val query.In this case we cannot get even work distribution, one mapper will end up holding no records as per the scenario.

Qu 2)

Solution:

```
mysql> create database test_new_db;
Query OK, 1 row affected (0.00 sec)

mysql> use test_new_db;
Database changed
```

Create the three tables in mysql and insert records in them :

```
mysql> CREATE TABLE City_Tbl
    -> (
    -> City_Name VARCHAR(255),
    -> City_ID INT PRIMARY KEY
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO City_Tbl values
    -> ('Bangalore',1000),
    -> ('Mumbai',1001),
    -> ('Chennai', 1002),
    -> ('Kolkata', 1003),
    -> ('Delhi',1004),
    -> ('Pune', 1005),
    -> ('Nagpur', 1006),
    -> ('Surat', 1007),
    -> ('Kochi',1008);
Query OK, 9 rows affected (0.01 sec)
Records: 9  Duplicates: 0  Warnings: 0

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CREATE TABLE State_Tbl
    -> (
    -> State_Name VARCHAR(255),
    -> Districts INT
    -> );
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> INSERT INTO State_Tbl values
    -> ('Karnataka', 30),
    -> ('TamilNadu', 32),
    -> ('Goa', 2),
    -> ('Kerala',14),
    -> ('Assam', 33);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE TABLE Country_Tbl
    -> (
    -> Name VARCHAR(255),
    -> Country_Code INT
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO Country_Tbl values
    -> ('Belgium',32),
    -> ('Brazil' ,55),
    -> ('France', 33),
    -> ('Iran', 98),
    -> ('India',91);
Query OK, 5 rows affected (0.01 sec)
Records: 5  Duplicates: 0  Warnings: 0

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

Verify the table data:

```
mysql> select * from City_Tbl;
+-----------+---------+
| City_Name | City_ID |
+-----------+---------+
| Bangalore |    1000 |
| Mumbai    |    1001 |
| Chennai   |    1002 |
| Kolkata   |    1003 |
| Delhi     |    1004 |
| Pune      |    1005 |
| Nagpur    |    1006 |
| Surat     |    1007 |
| Kochi     |    1008 |
+-----------+---------+
9 rows in set (0.00 sec)

mysql> select * from State_Tbl;
+------------+-----------+
| State_Name | Districts |
+------------+-----------+
| Karnataka  |        30 |
| TamilNadu  |        32 |
| Goa        |         2 |
| Kerala     |        14 |
| Assam      |        33 |
+------------+-----------+
5 rows in set (0.00 sec)
```

```
mysql> select * from Country_Tbl;
+---------+--------------+
| Name    | Country_Code |
+---------+--------------+
| Belgium |           32 |
| Brazil  |           55 |
| France  |           33 |
| Iran    |           98 |
| India   |           91 |
+---------+--------------+
5 rows in set (0.00 sec)
```

Sqoop Import Command:

```
(base) [cloudera@quickstart ~]$ sqoop import-all-tables \
> --connect jdbc:mysql://quickstart.cloudera:3306/test_new_db \
> --username root \
> --password cloudera \
> --warehouse-dir /user/cloudera/sqoop_all_tbl \
> --exclude-tables Country_Tbl \
> --num-mappers 3 \
> --autoreset-to-one-mapper
```

For City_Tbl :

```
20/05/01 04:23:16 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`City_ID`), MAX(`City_ID`) FROM `City_Tbl`
20/05/01 04:23:16 INFO db.IntegerSplitter: Split size: 2; Num splits: 3 from: 1000 to: 1008
20/05/01 04:23:16 INFO mapreduce.JobSubmitter: number of splits:3
```

For State_Tbl:

```
Split by column not provided or can't be inferred.  Resetting to one mapper
```

Output :

```
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/sqoop_all_tbl
Found 2 items
drwxr-xr-x   - cloudera cloudera          0 2020-05-01 04:23 /user/cloudera/sqoop_all_tbl/City_Tbl
drwxr-xr-x   - cloudera cloudera          0 2020-05-01 04:24 /user/cloudera/sqoop_all_tbl/State_Tbl
```

```
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/sqoop_all_tbl/City_Tbl/
Found 4 items
-rw-r--r--   1 cloudera cloudera          0 2020-05-01 04:23 /user/cloudera/sqoop_all_tbl/City_Tbl/_SUCCESS
-rw-r--r--   1 cloudera cloudera         40 2020-05-01 04:23 /user/cloudera/sqoop_all_tbl/City_Tbl/part-m-00000
-rw-r--r--   1 cloudera cloudera         34 2020-05-01 04:23 /user/cloudera/sqoop_all_tbl/City_Tbl/part-m-00001
-rw-r--r--   1 cloudera cloudera         34 2020-05-01 04:23 /user/cloudera/sqoop_all_tbl/City_Tbl/part-m-00002
```

```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/sqoop_all_tbl/City_Tbl/*
Bangalore,1000
Mumbai,1001
Chennai,1002
Kolkata,1003
Delhi,1004
Pune,1005
Nagpur,1006
Surat,1007
Kochi,1008
```

```
(base) [cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/sqoop_all_tbl/State_Tbl/
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2020-05-01 04:24 /user/cloudera/sqoop_all_tbl/State_Tbl/_SUCCESS
-rw-r--r--   1 cloudera cloudera         51 2020-05-01 04:24 /user/cloudera/sqoop_all_tbl/State_Tbl/part-m-00000
```

```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/sqoop_all_tbl/State_Tbl/*
Karnataka,30
TamilNadu,32
Goa,2
Kerala,14
Assam,33
```

Qu 3)

Solution:

```
mysql> select * from Categories;
+-------------+------------------------+--------------------+---------------------+
| category_id | category_department_id | category_name      | inclusion_date      |
+-------------+------------------------+--------------------+---------------------+
|           1 |                      2 | Football           | 2020-04-30 00:00:00 |
|           2 |                      2 | Handball           | 2020-05-01 00:00:00 |
|           3 |                      2 | Baseball & Softball | 2020-05-01 00:00:00 |
|           4 |                      2 | Basketball         | 2020-04-30 00:00:00 |
|           5 |                      3 | Tennis             | 2020-04-30 00:00:00 |
|           6 |                      3 | Hockey             | 2020-05-01 00:00:00 |
|           7 |                      3 | Swimming           | 2020-05-01 00:00:00 |
|           8 |                      3 | Cardio Equipment   | 2020-05-01 00:00:00 |
|           9 |                      4 | Strength Training  | 2020-05-01 00:00:00 |
|          10 |                      4 | Athletics          | 2020-05-02 00:00:00 |
|          11 |                   NULL | Cycling            | 2020-02-02 00:00:00 |
|          12 |                      5 | NULL               | 2020-01-15 00:00:00 |
+-------------+------------------------+--------------------+---------------------+
```

A. Sqoop Incremental Import Command:

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/test_db \
--username root \
--password cloudera \
--table Categories \
```

--null-non-string “-1” \

--null-string ‘\\N’ \

--incremental lastmodified \

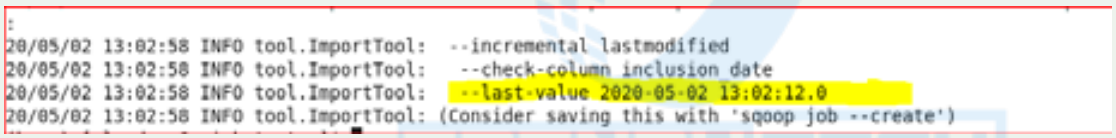
--check-column inclusion_date \

--last-value 0 \

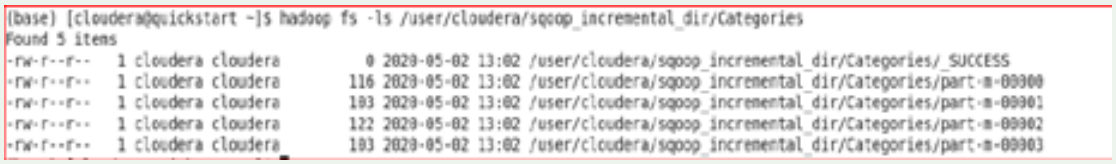
--verbose \

--warehouse-dir /user/cloudera/sqoop_incremental_dir

Boundary Query and the query run by each mapper on the splits internally



Output:



All 12 records are pulled in.

B. Inserting New Records and Updating existing Records in Mysql Categories Table:

```
mysql> INSERT INTO Categories values
    -> (13,6,'Surfing',CURRENT_TIMESTAMP),
    -> (14,2,'Mountaineering',CURRENT_TIMESTAMP);
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

```
mysql> UPDATE Categories SET category_department_id = 4,inclusion_date = CURRENT_TIMESTAMP WHERE category_id = 11;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE Categories SET category_name = 'Skating',inclusion_date = CURRENT_TIMESTAMP WHERE category_id = 12;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> commit;
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> select * from Categories;
+-------------+-----------------------+--------------------+---------------------+
| category_id | category_department_id | category_name      | inclusion_date      |
+-------------+-----------------------+--------------------+---------------------+
|           1 |                     2 | Football           | 2020-04-30 00:00:00 |
|           2 |                     2 | Handball           | 2020-05-01 00:00:00 |
|           3 |                     2 | Baseball & Softball | 2020-05-01 00:00:00 |
|           4 |                     2 | Basketball         | 2020-04-30 00:00:00 |
|           5 |                     3 | Tennis             | 2020-04-30 00:00:00 |
|           6 |                     3 | Hockey             | 2020-05-01 00:00:00 |
|           7 |                     3 | Swimming           | 2020-05-01 00:00:00 |
|           8 |                     3 | Cardio Equipment   | 2020-05-01 00:00:00 |
|           9 |                     4 | Strength Training  | 2020-05-01 00:00:00 |
|          10 |                     4 | Athletics          | 2020-05-02 00:00:00 |
|          11 |                     4 | Cycling            | 2020-05-02 14:22:03 |
|          12 |                     5 | Skating            | 2020-05-02 14:22:36 |
|          13 |                     6 | Surfing            | 2020-05-02 14:20:53 |
|          14 |                     2 | Mountaineering     | 2020-05-02 14:20:53 |
+-------------+-----------------------+--------------------+---------------------+
```

B.

Second Sqoop Import

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/test_db \
--username root \
--password cloudera \
--table Categories \
--null-non-string "-1" \
--null-string '\\N' \
--incremental lastmodified \
--check-column inclusion_date \
--last-value '2020-05-02 13:02:12.0' \
--verbose \
--warehouse-dir /user/cloudera/sqoop_incremental_dir \
--append
```

Bounding Val Query:

```
20/05/02 14:30:38 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`category_id`), MAX(`category_id`) FROM `Categories` WHERE ( `inclusion_date` >= '2020-
05-02 13:02:12.0' AND `inclusion_date` < '2020-05-02 14:30:31.0' )
20/05/02 14:30:38 INFO db.IntegerSplitter: Split size: 0; Num splits: 4 from: 11 to: 14
20/05/02 14:30:38 DEBUG db.IntegerSplitter: Splits: [                 11 to                 14] into 4 parts
20/05/02 14:30:38 DEBUG db.IntegerSplitter:                 11
20/05/02 14:30:38 DEBUG db.IntegerSplitter:                 12
20/05/02 14:30:38 DEBUG db.IntegerSplitter:                 13
20/05/02 14:30:38 DEBUG db.IntegerSplitter:                 14
20/05/02 14:30:38 DEBUG db.IntegerSplitter:                 14
20/05/02 14:30:38 DEBUG db.DataDrivenDBInputFormat: Creating input split with lower bound '`category_id` >= 11' and upper bound '`category_id` < 12'
20/05/02 14:30:38 DEBUG db.DataDrivenDBInputFormat: Creating input split with lower bound '`category_id` >= 12' and upper bound '`category_id` < 13'
20/05/02 14:30:38 DEBUG db.DataDrivenDBInputFormat: Creating input split with lower bound '`category_id` >= 13' and upper bound '`category_id` < 14'
20/05/02 14:30:38 DEBUG db.DataDrivenDBInputFormat: Creating input split with lower bound '`category_id` >= 14' and upper bound '`category_id` <= 14'
```

C .

Output:

Total 16 records in hdfs now: Yes , we found duplicate records as data is appended to existing records.Highlighted are the duplicate data

```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/sqoop_incremental_dir/Categories/*
1,2,Football,2020-04-30 00:00:00.0
2,2,Handball,2020-05-01 00:00:00.0
3,2,Baseball & Softball,2020-05-01 00:00:00.0
4,2,Basketball,2020-04-30 00:00:00.0
5,3,Tennis,2020-04-30 00:00:00.0
6,3,Hockey,2020-05-01 00:00:00.0
7,3,Swimming,2020-05-01 00:00:00.0
8,3,Cardio Equipment,2020-05-01 00:00:00.0
9,4,Strength Training,2020-05-01 00:00:00.0
10,4,Athletics,2020-05-02 00:00:00.0
11,-1,Cycling,2020-02-02 00:00:00.0
12,5,\N,2020-01-15 00:00:00.0
11,4,Cycling,2020-05-02 14:22:03.0
12,5,Skating,2020-05-02 14:22:36.0
13,6,Surfing,2020-05-02 14:20:53.0
14,2,Mountaineering,2020-05-02 14:20:53.0
```

```
20/05/02 14:31:11 INFO mapreduce.ImportJobBase: Transferred 147 bytes in 39.9229 seconds (3.6821 bytes/sec)
20/05/02 14:31:11 INFO mapreduce.ImportJobBase: Retrieved 4 records.
20/05/02 14:31:11 INFO util.AppendUtils: Appending to directory Categories
```

```
20/05/02 14:31:11 INFO tool.ImportTool:  --incremental lastmodified
20/05/02 14:31:11 INFO tool.ImportTool:   --check-column inclusion_date
20/05/02 14:31:11 INFO tool.ImportTool:   --last-value 2020-05-02 14:30:31.0
```

To get the latest records in hdfs without duplicates:

```
mysql> describe Categories_new;
+------------------------+-------------+------+-----+---------+-------+
| Field                  | Type        | Null | Key | Default | Extra |
+------------------------+-------------+------+-----+---------+-------+
| category_id            | int(11)     | NO   |     | 0       |       |
| category_department_id | int(11)     | YES  |     | NULL    |       |
| category_name          | varchar(45) | YES  |     | NULL    |       |
| inclusion_date         | datetime    | NO   |     | NULL    |       |
+------------------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

The Categories_new table does not have a Primary Key.

D.

To Automate the import process we use a sqoop job

Sqoop Job Creation:

**sqoop job **

**--create job_Categories_new **

**-- import **

**--connect jdbc:mysql://quickstart.cloudera:3306/test_db **

**--username root **

**--password cloudera **

**--table Categories_new **

**--warehouse-dir /user/cloudera/sqoop_incremental_dir2 **

**--split-by category_id **

**--incremental lastmodified **

**--check-column inclusion_date **

**--last-value 0 **

**--verbose **

--merge-key category_id

Note: password-encryption has been shown at the end of the document.

Listing the sqoop jobs

sqoop job --list

Executing the sqoop job: The lastvalue will be taken care by the saved sqoop job

sqoop job --exec job_Categories_new

Output:

Bounding val query:

```
20/05/02 19:58:04 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`category_id`), MAX(`category_id`) FROM `Categories_new` WHERE ( `inclusion_date` >= '0
' AND `inclusion_date` < '2020-05-02 19:57:59.0' )
```

```
Retrieved 14 records
```

To see the last value in a Sqoop Job:

sqoop job --show job_Categories_new

```
incremental.last.value = 2020-05-02 19:57:59.0
db.connect.string = jdbc:mysql://quickstart.cloudera:3306/test_db
codegen.output.delimiters.escape = 0
codegen.output.delimiters.enclose.required = false
codegen.input.delimiters.field = 0
mainframe.input.dataset.type = p
split.limit = null
```

Hdfs Output:

```
(base) [cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/sqoop_incremental_
dir2/Categories_new/*
1,2,Football,2020-04-30 00:00:00.0
2,2,Handball,2020-05-01 00:00:00.0
3,2,Baseball & Softball,2020-05-01 00:00:00.0
4,2,Basketball,2020-04-30 00:00:00.0
5,3,Tennis,2020-04-30 00:00:00.0
6,3,Hockey,2020-05-01 00:00:00.0
7,3,Swimming,2020-05-01 00:00:00.0
8,3,Cardio Equipment,2020-05-01 00:00:00.0
9,4,Strength Training,2020-05-01 00:00:00.0
10,4,Athletics,2020-05-02 00:00:00.0
11,4,Cycling,2020-05-02 14:22:03.0
12,5,Skating,2020-05-02 14:22:36.0
13,6,Surfing,2020-05-02 14:20:53.0
14,2,Mountaineering,2020-05-02 14:20:53.0
```

Post Inserts and Updates in Categories_new table

```
mysql> select * from Categories_new;
+-------------+------------------------+--------------------+---------------------+
| category_id | category_department_id | category_name      | inclusion_date      |
+-------------+------------------------+--------------------+---------------------+
|           1 |                      2 | Football           | 2020-04-30 00:00:00 |
|           2 |                      2 | Handball           | 2020-05-01 00:00:00 |
|           3 |                      2 | Baseball & Softball | 2020-05-01 00:00:00 |
|           4 |                      2 | Basketball         | 2020-04-30 00:00:00 |
|           5 |                      3 | Tennis             | 2020-04-30 00:00:00 |
|           6 |                      3 | Hockey             | 2020-05-01 00:00:00 |
|           7 |                      3 | Swimming           | 2020-05-01 00:00:00 |
|           8 |                      3 | Cardio Equipment   | 2020-05-01 00:00:00 |
|           9 |                      4 | Strength Training  | 2020-05-01 00:00:00 |
|          10 |                      4 | Athletics          | 2020-05-02 00:00:00 |
|          11 |                      4 | Cycling            | 2020-05-02 14:22:03 |
|          12 |                      5 | Skating            | 2020-05-02 14:22:36 |
|          13 |                      4 | Surfing            | 2020-05-02 20:39:53 |
|          14 |                      4 | Mountaineering     | 2020-05-02 20:40:06 |
|          15 |                      6 | Boxing             | 2020-05-02 20:39:35 |
|          16 |                      6 | Cycling            | 2020-05-02 20:39:35 |
+-------------+------------------------+--------------------+---------------------+
16 rows in set (0.00 sec)
```

Running the sqoop job again post inserts and updates in source table:(avoiding duplicate records to be imported)

sqoop job --exec job_Categories_new

Internal Bound Val Query:

```
20/05/02 20:44:38 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`category_id`), MAX(`category_id`) FROM `Categories_new` WHERE ( `inclusion_date` >= '2
020-05-02 19:57:56.0' AND `inclusion_date` < '2020-05-02 20:44:25.0' )
20/05/02 20:44:38 INFO db.IntegerSplitter: Split size: 0; Num splits: 4 from: 13 to: 16
```

E)

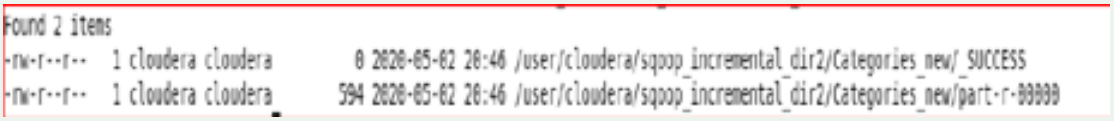
Output:

```
20/05/02 20:45:17 INFO mapreduce.ImportJobBase: Transferred 146 bytes in 52.1679 seconds (2.7987 bytes/sec)
20/05/02 20:45:17 INFO mapreduce.ImportJobBase: Retrieved 4 records.
20/05/02 20:45:17 INFO tool.ImportTool: Final destination exists, will run merge job.
```

Total 16 records will be in hdfs now

F)

Only one Reducer part file is generated:No mapper files are generated post the second import. As we are removing duplicates using --merge-key.



G)

Saved Last value of sqoop job for next run:

sqoop job --show job_Categories_new

```
verbose = true
hcatalog.drop.and.create.table = false
incremental.last.value = 2020-05-02 20:44:25.0
```

Note:

Encrypted Password Creation:

Alias -mysql.test_db.securepassword

Password file stored in hdfs at location: /user/cloudera/encryptpswd

Password filename : jceks_pswdfile

```
(base) [cloudera@quickstart ~]$ hadoop credential create mysql.test_db.securepas
sword -provider jceks://hdfs/user/cloudera/encryptpswd/jceks_pswdfile
WARNING: You have accepted the use of the default provider password
by not configuring a password in one of the two following locations:
    * In the environment variable HADOOP_CREDSTORE_PASSWORD
    * In a file referred to by the configuration entry
      hadoop.security.credstore.java-keystore-provider.password-file.
Please review the documentation regarding provider passwords in
the keystore passwords section of the Credential Provider API
Continuing with the default provider password.

Enter alias password:
Enter alias password again:
mysql.test_db.securepassword has been successfully created.
Provider jceks://hdfs/user/cloudera/encryptpswd/jceks_pswdfile has been updated.
```

Example of usage in say, sqoop eval command:

```
(base) [cloudera@quickstart ~]$ sqoop-eval \
> -Dhadoop.security.credential.provider.path=jceks://hdfs/user/cloudera/encryptpswd/jceks_pswdfile \
> --connect jdbc:mysql://quickstart.cloudera:3306/test_db \
> --username root \
> --password-alias mysql.test_db.securepassword \
> --query "select count(*) from Customers"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/05/13 21:08:27 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
20/05/13 21:08:29 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
----------------------
| count(*)           |
----------------------
| 6                  |
```

****