<u>Problem Statement -</u> Perform incremental load in Hive. Read from MySQL Table and load it in Hive table. Create hive table if it does not exist. If it exists, perform the incremental load.

Solution-

In order to do above task we will create a table first named emp in MySQL inside database ‘db1’ with four columns in it as shown below-

```
mysql> CREATE TABLE emp
    -> (
    -> emp_id int,
    -> emp_name varchar(20),
    -> emp_sal int,
    -> emp_rating int
    -> );
Query OK, 0 rows affected (0.05 sec)
```

Then we will insert some values (rows) in it as shown below-

```
mysql> insert into emp values(101, 'Amitabh' ,20000,1);
Query OK, 1 row affected (0.08 sec)

mysql> insert into emp values(102, 'Shahrukh' ,10000,2);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(103, 'Akshay' ,11000,3);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(104, 'Anubhav' ,5000,4);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(105, 'Pawan' ,2500,5);
Query OK, 1 row affected (0.00 sec)
```

As of now we have inserted only 5 rows in it as shown below till emp_id-105-

```
mysql> select * from emp;
+--------+----------+---------+------------+
| emp_id | emp_name | emp_sal | emp_rating |
+--------+----------+---------+------------+
|    101 | Amitabh  |   20000 |          1 |
|    102 | Shahrukh |   10000 |          2 |
|    103 | Akshay   |   11000 |          3 |
|    104 | Anubhav  |    5000 |          4 |
|    105 | Pawan    |    2500 |          5 |
+--------+----------+---------+------------+
5 rows in set (0.00 sec)
```

Now we will use below Sqoop script to import the above inserted data in HIVE-

- **sqoop import --connect jdbc:mysql://localhost/db1 **
- **--username 'root' -P --table 'emp' **
- **--target-dir '/myhive2' --fields-terminated-by ',' **
- **--hive-import --create-hive-table --hive-table 'default.emp3' **
- **--incremental append **
- **--check-column emp_id **
- **-m 1**

  Here in above script the first line is used to make a JDBC connection with MySQL database db1. Username has been specified as 'root' and it has been directed to ask for password using –P. The table which we are importing is emp. The target directory has been specified as myhive2 in HDFS and the fields will be delimited by ','.

  The 4th line instructs to import the data and create the hive table 'emp3' inside 'default' database.

  The 5th line shows to do an incremental append which means whenever some new rows are added it will be appended to the hive table. Now the 6th line performs the appending check on the column- 'emp_id' i.e. hive will check on emp_id column to append to the table.

  -m 1 specifies that only 1 mapper has been used for this task-

```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/db1 \
> --username 'root' -P --table 'emp'  \
> --target-dir '/myhive2' --fields-terminated-by ',' \
> --hive-import --create-hive-table --hive-table 'default.emp3' \
> --incremental append \
> --check-column emp_id \
> -m 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

After running above command we can see that a new table emp3 has been created inside default database in HIVE-

```
hive (default)> show tables;
OK
tab_name
emp
emp3
sample_07
sample_08
Time taken: 1.892 seconds, Fetched: 4 row(s)
hive (default)>
```

Also the table emp3 in HIVE contains only those 5 rows which were inserted in the MySQL table emp-

```
hive (default)> select * from emp3;
OK
emp3.emp_id     emp3.emp_name   emp3.emp_sal    emp3.emp_rating
101     Amitabh 20000   1
102     Shahrukh        10000   2
103     Akshay  11000   3
104     Anubhav 5000    4
105     Pawan   2500    5
Time taken: 1.975 seconds, Fetched: 5 row(s)
hive (default)>
```

Now again we are inserting some more values inside MySQL table emp after emp_id 105-

```
mysql> insert into emp values(106, 'Aamir' ,25000,1);
Query OK, 1 row affected (0.04 sec)

mysql> insert into emp values(107, 'Salman' ,17500,2);
Query OK, 1 row affected (0.02 sec)

mysql> insert into emp values(108, 'Ranbir' ,14000,3);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(109, 'Katrina' ,1000,4);
Query OK, 1 row affected (0.00 sec)

mysql> insert into emp values(110, 'Priyanka' ,2000,5);
Query OK, 1 row affected (0.00 sec)
```

Same can be seen below if we do a SELECT from emp table in MySQL table-

```
mysql> select * from emp;
+--------+----------+---------+------------+
| emp_id | emp_name | emp_sal | emp_rating |
+--------+----------+---------+------------+
|    101 | Amitabh  |   20000 |          1 |
|    102 | Shahrukh |   10000 |          2 |
|    103 | Akshay   |   11000 |          3 |
|    104 | Anubhav  |    5000 |          4 |
|    105 | Pawan    |    2500 |          5 |
|    106 | Aamir    |   25000 |          1 |
|    107 | Salman   |   17500 |          2 |
|    108 | Ranbir   |   14000 |          3 |
|    109 | Katrina  |    1000 |          4 |
|    110 | Priyanka |    2000 |          5 |
+--------+----------+---------+------------+
```

Now we will run below script to do an incremental append inside hive table-

- **sqoop import --connect jdbc:mysql://localhost/db1 **
- **--username 'root' -P --table 'emp' **
- **--target-dir '/myhive2' --fields-terminated-by ',' **
- **--hive-import --hive-table 'default.emp3' **
- **--incremental append **
- **--check-column emp_id **
- **--last-value 110 **
- **-m 1**

Here in above script we have kept everything same except we have specified the last value of the column emp_id as 105 which we added previously. So that HIVE will append only those rows which have value of emp_id greater than 105.

```
[root@sandbox ~]# sqoop import --connect jdbc:mysql://localhost/db1 \
> --username 'root' -P --table 'emp'  \
> --target-dir '/myhive2' --fields-terminated-by ',' \
> --hive-import --hive-table 'default.emp3' \
> --incremental append \
> --check-column emp_id \
> --last-value 105 \
> -m 1
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

After running above script we can see that rows with emp_id value greater than 105 have been appended to the HIVE table emp3 as shown below-

```
hive (default)> select * from emp3;
OK
emp3.emp_id     emp3.emp_name   emp3.emp_sal    emp3.emp_rating
101     Amitabh 20000   1
102     Shahrukh        10000   2
103     Akshay  11000   3
104     Anubhav 5000    4
105     Pawan   2500    5
106     Aamir   25000   1
107     Salman  17500   2
108     Ranbir  14000   3
109     Katrina 1000    4
110     Priyanka        2000    5
Time taken: 1.835 seconds, Fetched: 10 row(s)
hive (default)>
```