Assignment 20.2

2. 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.

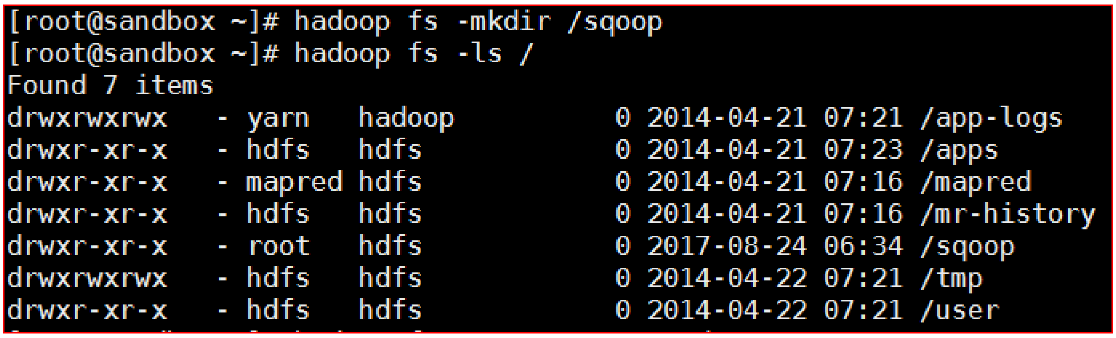
Step 1: use ‘customers.dat’ file as an input to the sqoop.

Data is ‘,’ separated and file contains field as id,name,location,age.

Step 2: move ‘customers.dat’ file to HDFS

$ hadoop fs -mkdir /sqoop

$ hadoop fs -ls /



```
[root@sandbox ~]# hadoop fs -mkdir /sqoop
[root@sandbox ~]# hadoop fs -ls /
Found 7 items
drwxrwxrwx   - yarn   hadoop          0 2014-04-21 07:21 /app-logs
drwxr-xr-x   - hdfs   hdfs            0 2014-04-21 07:23 /apps
drwxr-xr-x   - mapred hdfs            0 2014-04-21 07:16 /mapred
drwxr-xr-x   - hdfs   hdfs            0 2014-04-21 07:16 /mr-history
drwxr-xr-x   - root   hdfs            0 2017-08-24 06:34 /sqoop
drwxrwxrwx   - hdfs   hdfs            0 2014-04-22 07:21 /tmp
drwxr-xr-x   - hdfs   hdfs            0 2014-04-22 07:21 /user
```

Step 3: start mysql.

<u>Change user to root</u>

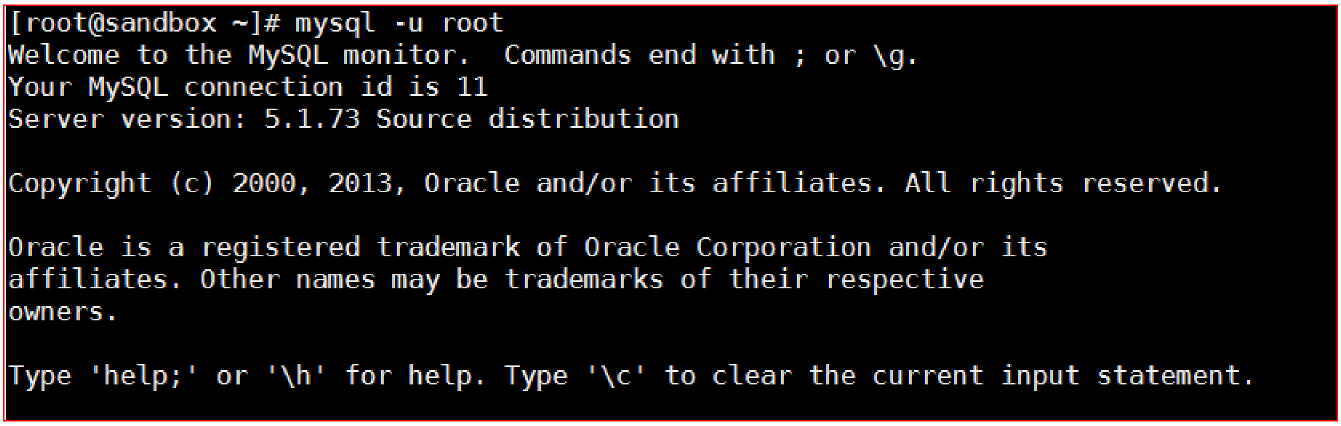
$ sudo su

<u>Start mysqld service</u>

# service mysqld start

<u>Start mysql as user root</u>

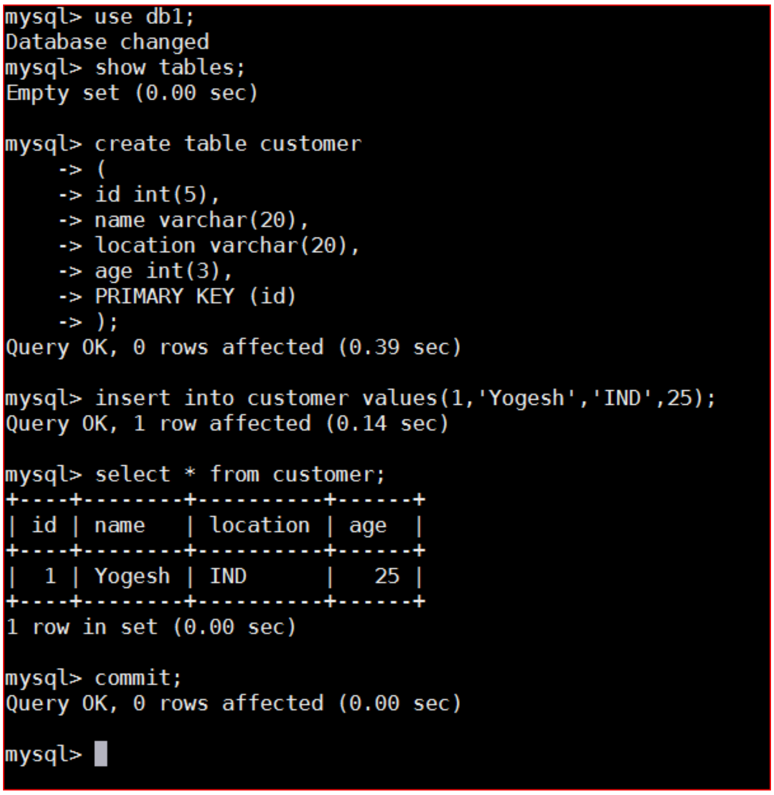
# mysql -u root



```
[root@sandbox ~]# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

Step 4: create table with the above mentioned fields (columns)

```
use db1;
show tables;
create table customer
(
id int(5),
name varchar(20),
location varchar(20),
age int(3),
PRIMARY KEY (id)
);
insert into customer values(1,'Yogesh','IND',25);
select * from customer;
commit;
```



```
mysql> use db1;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> create table customer
    -> (
    -> id int(5),
    -> name varchar(20),
    -> location varchar(20),
    -> age int(3),
    -> PRIMARY KEY (id)
    -> );
Query OK, 0 rows affected (0.39 sec)

mysql> insert into customer values(1,'Yogesh','IND',25);
Query OK, 1 row affected (0.14 sec)

mysql> select * from customer;
+----+--------+----------+------+
| id | name   | location | age  |
+----+--------+----------+------+
|  1 | Yogesh | IND      |   25 |
+----+--------+----------+------+
1 row in set (0.00 sec)

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

mysql> 
```

Step 5: Use Hive import to get data from Sqoop from SQL to Hive.

*$ sqoop job --create mysqoopjob *

-- \

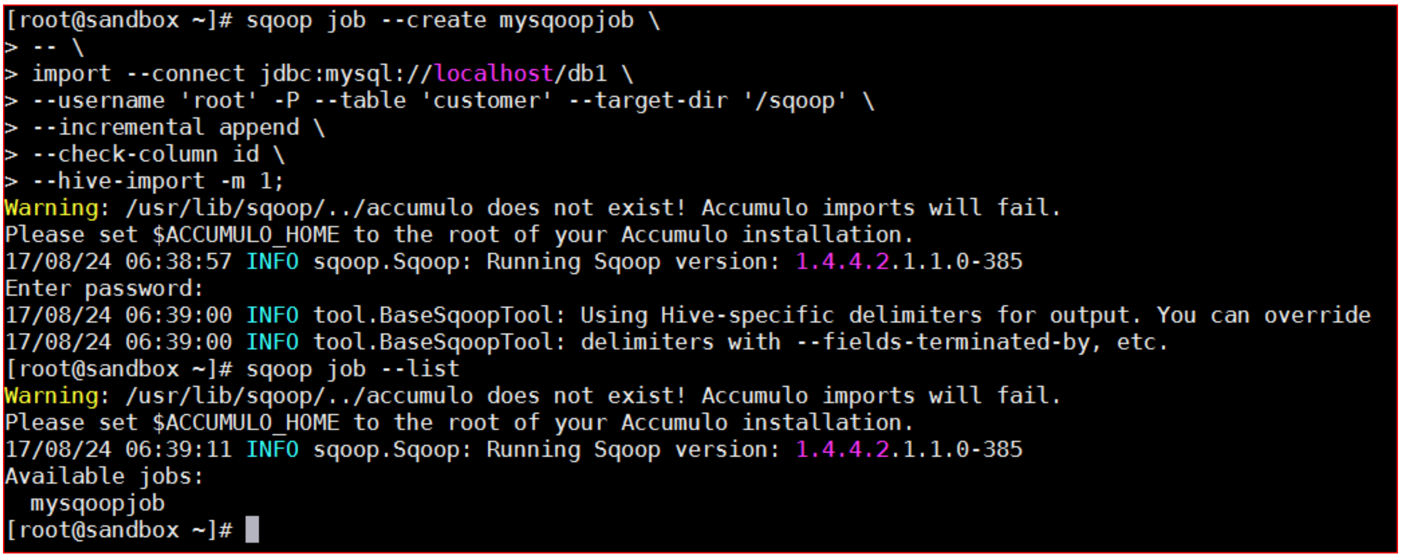
*import --connect jdbc:mysql://localhost/db1 *

*--username 'root' -P --table 'customer' --target-dir '/sqoop' *

*--incremental append *

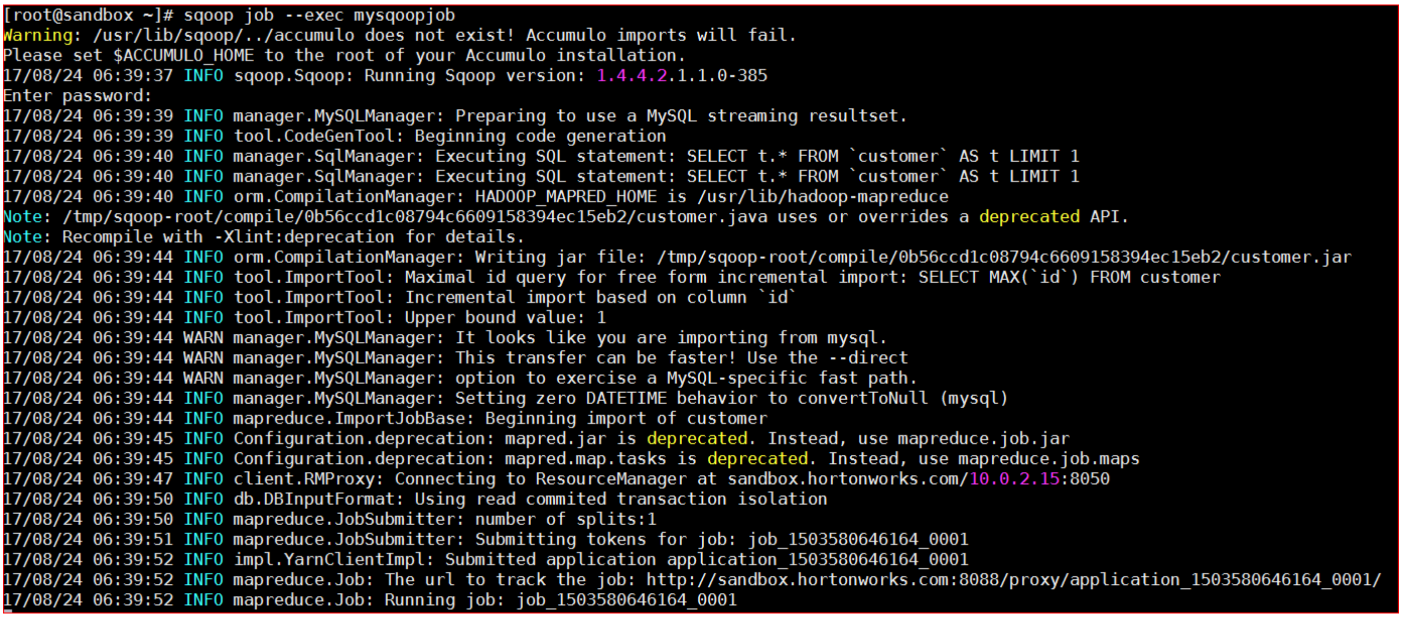
*--check-column id *

--hive-import -m 1;

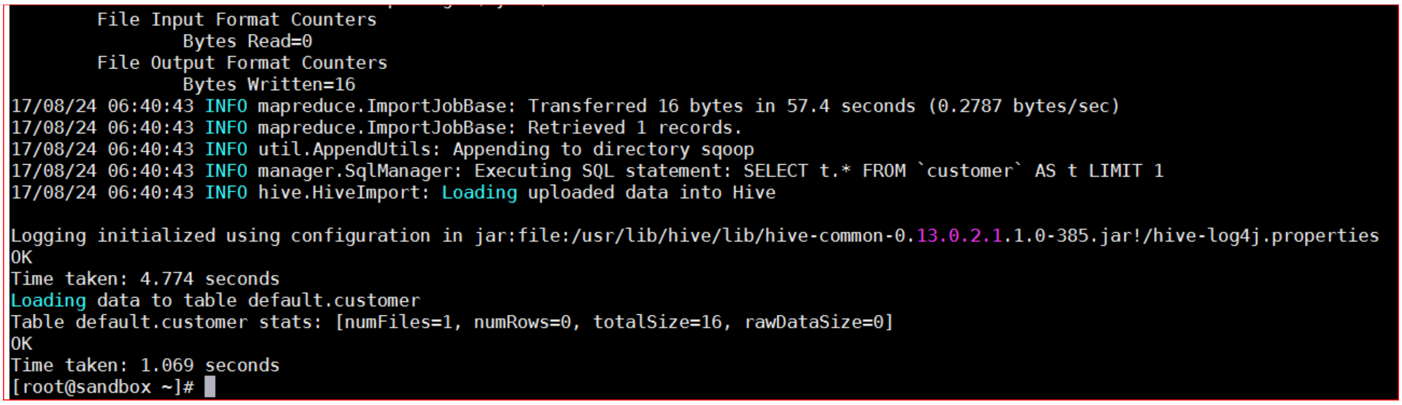


```
[root@sandbox ~]# sqoop job --create mysqoopjob \
> -- \
> import --connect jdbc:mysql://localhost/db1 \
> --username 'root' -P --table 'customer' --target-dir '/sqoop' \
> --incremental append \
> --check-column id \
> --hive-import -m 1;
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
17/08/24 06:38:57 INFO sqoop.Sqoop: Running Sqoop version: 1.4.4.2.1.1.0-385
Enter password:
17/08/24 06:39:00 INFO tool.BaseSqoopTool: Using Hive-specific delimiters for output. You can override
17/08/24 06:39:00 INFO tool.BaseSqoopTool: delimiters with --fields-terminated-by, etc.
[root@sandbox ~]# sqoop job --list
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
17/08/24 06:39:11 INFO sqoop.Sqoop: Running Sqoop version: 1.4.4.2.1.1.0-385
Available jobs:
  mysqoopjob
[root@sandbox ~]# 
```

$ sqoop job --exec mysqoopjob



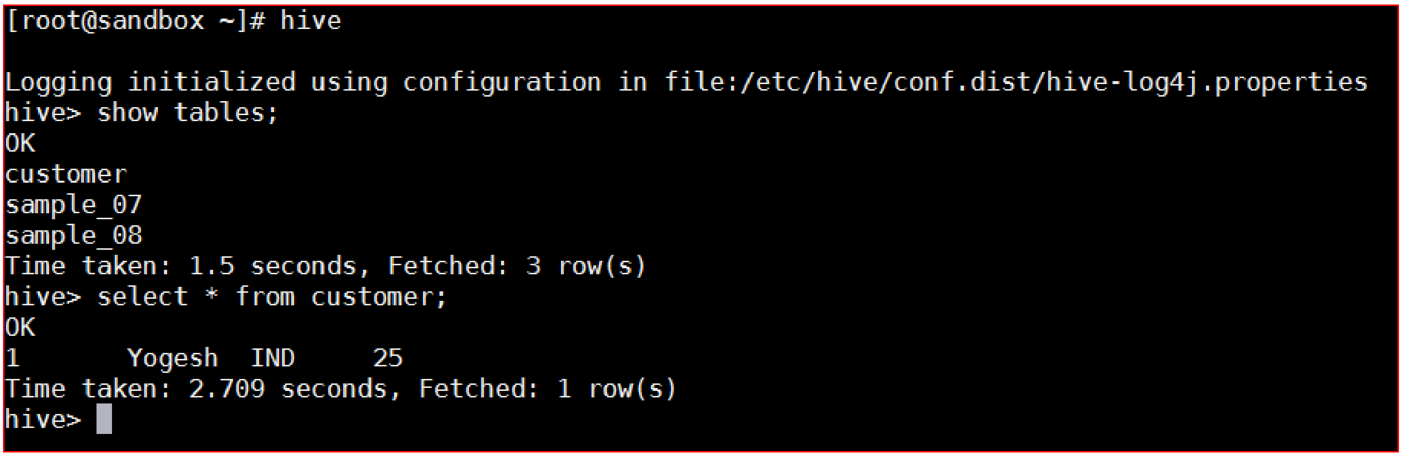
```
[root@sandbox ~]# sqoop job --exec mysqoopjob
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
17/08/24 06:39:37 INFO sqoop.Sqoop: Running Sqoop version: 1.4.4.2.1.1.0-385
Enter password:
17/08/24 06:39:39 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
17/08/24 06:39:39 INFO tool.CodeGenTool: Beginning code generation
17/08/24 06:39:40 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `customer` AS t LIMIT 1
17/08/24 06:39:40 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `customer` AS t LIMIT 1
17/08/24 06:39:40 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/0b56ccd1c08794c6609158394ec15eb2/customer.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
17/08/24 06:39:44 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/0b56ccd1c08794c6609158394ec15eb2/customer.jar
17/08/24 06:39:44 INFO tool.ImportTool: Maximal id query for free form incremental import: SELECT MAX(`id`) FROM customer
17/08/24 06:39:44 INFO tool.ImportTool: Incremental import based on column `id`
17/08/24 06:39:44 INFO tool.ImportTool: Upper bound value: 1
17/08/24 06:39:44 WARN manager.MySQLManager: It looks like you are importing from mysql.
17/08/24 06:39:44 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
17/08/24 06:39:44 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
17/08/24 06:39:44 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
17/08/24 06:39:44 INFO mapreduce.ImportJobBase: Beginning import of customer
17/08/24 06:39:45 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
17/08/24 06:39:45 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
17/08/24 06:39:47 INFO client.RMProxy: Connecting to ResourceManager at sandbox.hortonworks.com/10.0.2.15:8050
17/08/24 06:39:50 INFO db.DBInputFormat: Using read commited transaction isolation
17/08/24 06:39:50 INFO mapreduce.JobSubmitter: number of splits:1
17/08/24 06:39:51 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1503580646164_0001
17/08/24 06:39:52 INFO impl.YarnClientImpl: Submitted application application_1503580646164_0001
17/08/24 06:39:52 INFO mapreduce.Job: The url to track the job: http://sandbox.hortonworks.com:8088/proxy/application_1503580646164_0001/
17/08/24 06:39:52 INFO mapreduce.Job: Running job: job_1503580646164_0001
```



```
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=16
17/08/24 06:40:43 INFO mapreduce.ImportJobBase: Transferred 16 bytes in 57.4 seconds (0.2787 bytes/sec)
17/08/24 06:40:43 INFO mapreduce.ImportJobBase: Retrieved 1 records.
17/08/24 06:40:43 INFO util.AppendUtils: Appending to directory sqoop
17/08/24 06:40:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `customer` AS t LIMIT 1
17/08/24 06:40:43 INFO hive.HiveImport: Loading uploaded data into Hive

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.13.0.2.1.1.0-385.jar!/hive-log4j.properties
OK
Time taken: 4.774 seconds
Loading data to table default.customer
Table default.customer stats: [numFiles=1, numRows=0, totalSize=16, rawDataSize=0]
OK
Time taken: 1.069 seconds
[root@sandbox ~]#
```

Step 6: goto Hive shell and check table customer.

*select * from customer;*



```
[root@sandbox ~]# hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
hive> show tables;
OK
customer
sample_07
sample_08
Time taken: 1.5 seconds, Fetched: 3 row(s)
hive> select * from customer;
OK
1       Yogesh  IND     25
Time taken: 2.709 seconds, Fetched: 1 row(s)
hive>
```

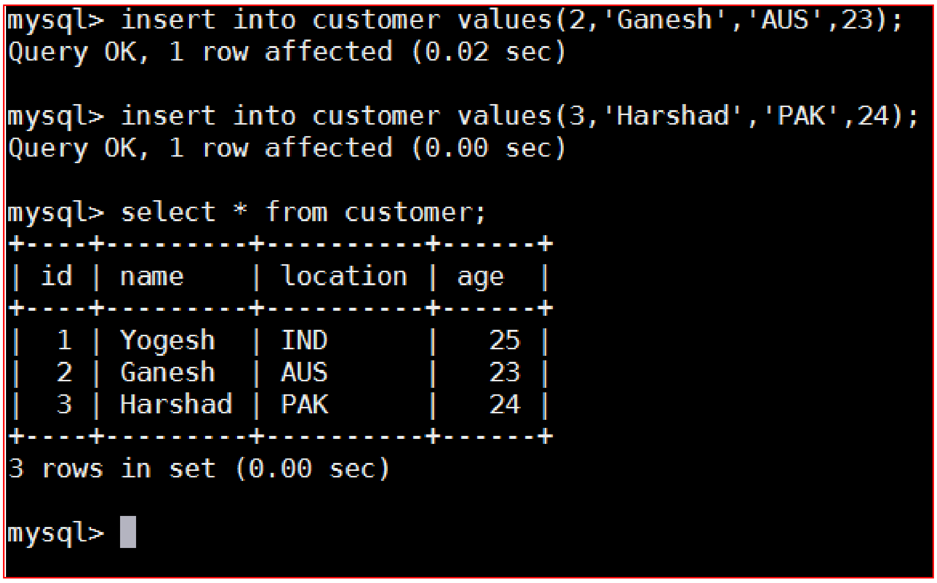
Table has been loaded with correct values.

Step 7: insert rows into existing table in SQL.

insert into customer values(2,'Ganesh','AUS',23);

insert into customer values(3,'Harshad','PAK',24);

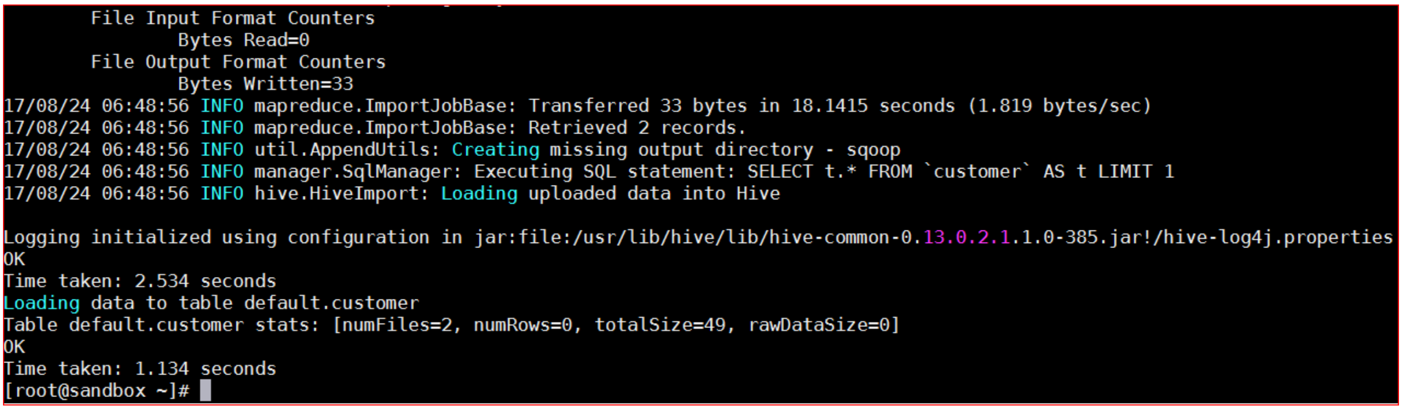
*select * from customer;*



```
mysql> insert into customer values(2,'Ganesh','AUS',23);
Query OK, 1 row affected (0.02 sec)

mysql> insert into customer values(3,'Harshad','PAK',24);
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer;
+----+---------+----------+------+
| id | name    | location | age  |
+----+---------+----------+------+
|  1 | Yogesh  | IND      |   25 |
|  2 | Ganesh  | AUS      |   23 |
|  3 | Harshad | PAK      |   24 |
+----+---------+----------+------+
3 rows in set (0.00 sec)

mysql>
```

Step 8: Run the sqoop job mysqoopjob to perform incremental load into hive table.

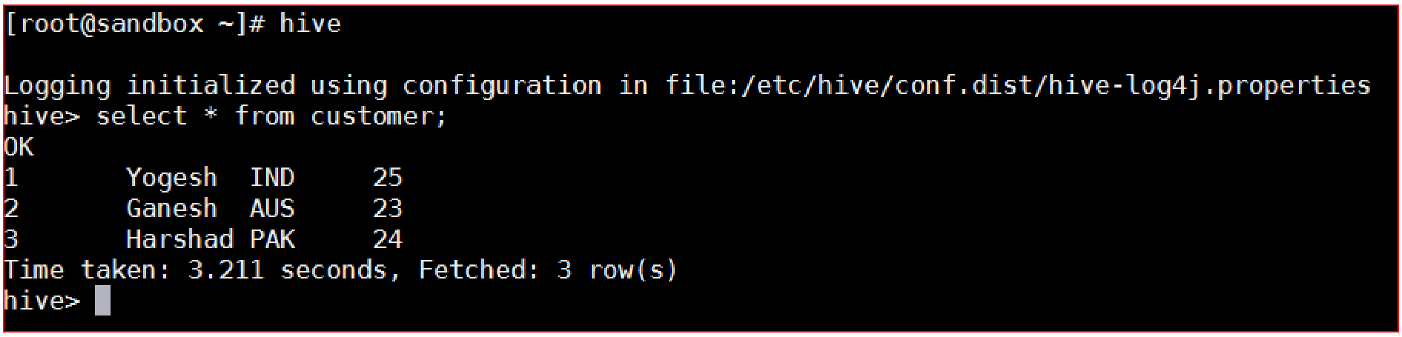
sqoop job --exec mysqoopjob



```
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=33
17/08/24 06:48:56 INFO mapreduce.ImportJobBase: Transferred 33 bytes in 18.1415 seconds (1.819 bytes/sec)
17/08/24 06:48:56 INFO mapreduce.ImportJobBase: Retrieved 2 records.
17/08/24 06:48:56 INFO util.AppendUtils: Creating missing output directory - sqoop
17/08/24 06:48:56 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `customer` AS t LIMIT 1
17/08/24 06:48:56 INFO hive.HiveImport: Loading uploaded data into Hive

Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-0.13.0.2.1.1.0-385.jar!/hive-log4j.properties
OK
Time taken: 2.534 seconds
Loading data to table default.customer
Table default.customer stats: [numFiles=2, numRows=0, totalSize=49, rawDataSize=0]
OK
Time taken: 1.134 seconds
[root@sandbox ~]#
```

Step 9: goto Hive shell and check table customer.

*select * from customer;*



```
[root@sandbox ~]# hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
hive> select * from customer;
OK
1       Yogesh  IND     25
2       Ganesh  AUS     23
3       Harshad PAK     24
Time taken: 3.211 seconds, Fetched: 3 row(s)
hive>
```

Table has been loaded with correct values.