ASSIGNMENT 11.3

Explain in brief partitioning and bucketing with an example

Partition is horizontally dividing the data into number of slice in a equal and manageable manner.Every partition is stored as directory within data warehouse table.

Hive partition it supported for Multiple columns in a table .In Hive we can apply Hive Partition concept on Managed tables and External tables.If we not crated dynamic partition for hive, Hive also creates an automatic partition scheme when the table is created

Example Hive Partition

```
create table cityreport(cityid string,creport string, ctover string)
partitioned by (city string)
row format delimited
fields terminated by '|'
stored as textfile;
```

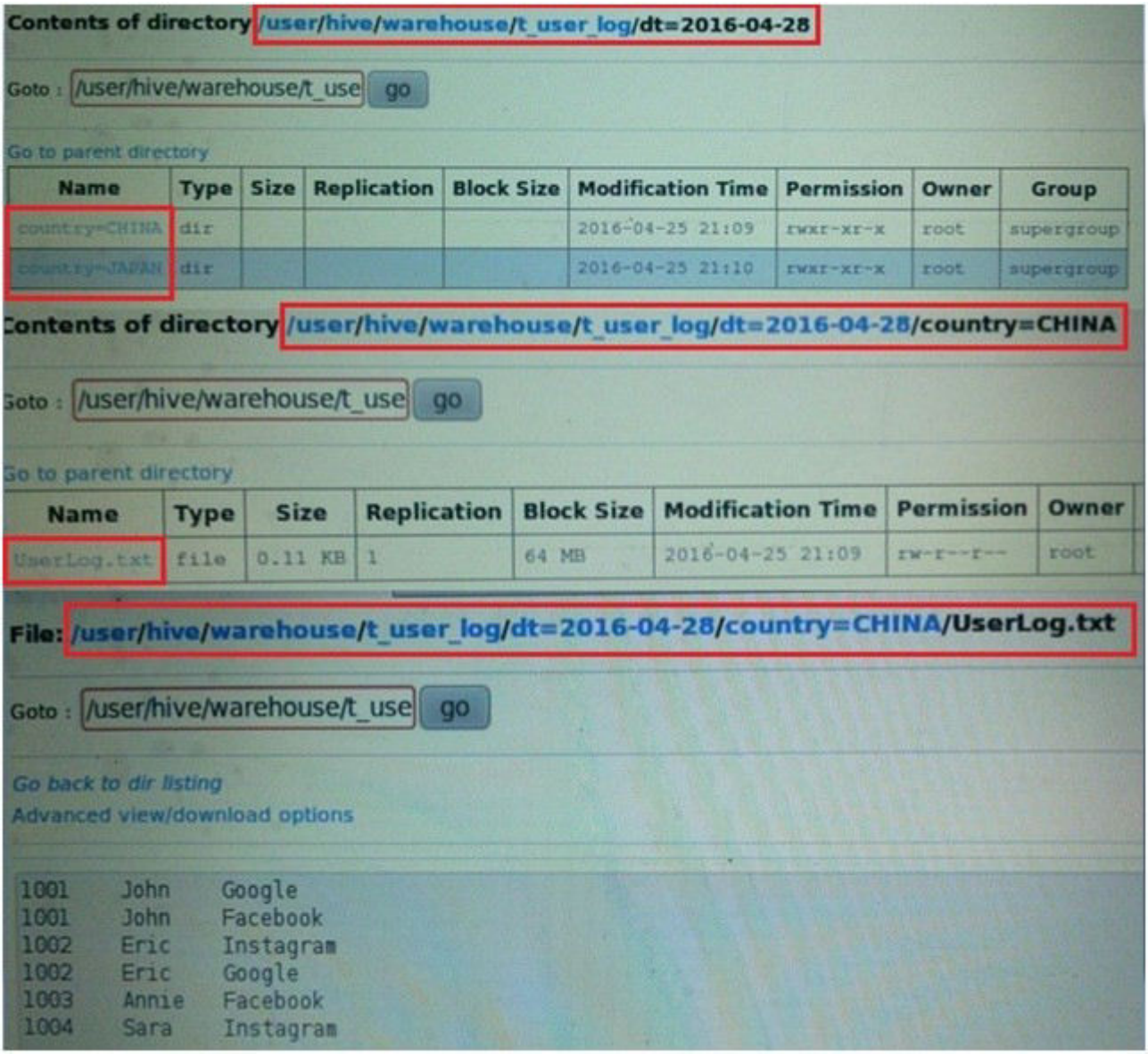
Static Partitioning:In Static Partition, we know the partition column before itself. Now when we load data there it makes the difference.

LOAD DATA LOCAL INPATH [path_name] OVERWRITE INTO TABLE [table_name] PARTITION(partition_column='value'....). Here we have to give the partition column value explicitly whenever we want to create new partition as shown below:

CODE.

```
File  Edit  View  Search  Terminal  Help
e:null, serializationLib:org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe, parameters:{serialization.format=  , field.delim=
Time taken: 18.097 seconds
root@ubuntu:/home/GAURAV/HIVE# nano UserLog.txt
root@ubuntu:/home/GAURAV/HIVE# hive -e "LOAD DATA LOCAL INPATH 'UserLog.txt' OVERWRITE INTO TABLE T_USER_LOG PARTITION(dt='2016-04-29',country='UK')";
Hive history file=/tmp/root/hive_job_log_root_201604252107_817941186.txt
Copying data from file:/home/GAURAV/HIVE/UserLog.txt
Copying file: file:/home/GAURAV/HIVE/UserLog.txt
Loading data to table default.t_user_log partition (dt=2016-04-29, country=UK)
OK
Time taken: 25.392 seconds
root@ubuntu:/home/GAURAV/HIVE# hive -e "LOAD DATA LOCAL INPATH 'UserLog.txt' OVERWRITE INTO TABLE T_USER_LOG PARTITION(dt='2016-04-29',country='US')";
Hive history file=/tmp/root/hive_job_log_root_201604252108_1907926266.txt
Copying data from file:/home/GAURAV/HIVE/UserLog.txt
Copying file: file:/home/GAURAV/HIVE/UserLog.txt
Loading data to table default.t_user_log partition (dt=2016-04-29, country=US)
OK
Time taken: 19.501 seconds
root@ubuntu:/home/GAURAV/HIVE# hive -e "LOAD DATA LOCAL INPATH 'UserLog.txt' OVERWRITE INTO TABLE T_USER_LOG PARTITION(dt='2016-04-28',country='CHINA')";
Hive history file=/tmp/root/hive_job_log_root_201604252109_1905707855.txt
Copying data from file:/home/GAURAV/HIVE/UserLog.txt
Copying file: file:/home/GAURAV/HIVE/UserLog.txt
Loading data to table default.t_user_log partition (dt=2016-04-28, country=CHINA)
OK
Time taken: 18.981 seconds
root@ubuntu:/home/GAURAV/HIVE# hive -e "LOAD DATA LOCAL INPATH 'UserLog.txt' OVERWRITE INTO TABLE T_USER_LOG PARTITION(dt='2016-04-28',country='JAPAN')";
Hive history file=/tmp/root/hive_job_log_root_201604252109_1683935071.txt
Copying data from file:/home/GAURAV/HIVE/UserLog.txt
Copying file: file:/home/GAURAV/HIVE/UserLog.txt
Loading data to table default.t_user_log partition (dt=2016-04-28, country=JAPAN)
OK
Time taken: 18.635 seconds
root@ubuntu:/home/GAURAV/HIVE# hive -e "LOAD DATA LOCAL INPATH 'UserLog.txt' OVERWRITE INTO TABLE T_USER_LOG PARTITION(dt='2016-04-27',country='INDIA')";
Hive history file=/tmp/root/hive_job_log_root_201604252110_1808246724.txt
Copying data from file:/home/GAURAV/HIVE/UserLog.txt
Copying file: file:/home/GAURAV/HIVE/UserLog.txt
Loading data to table default.t_user_log partition (dt=2016-04-27, country=INDIA)
OK
Time taken: 18.902 seconds
root@ubuntu:/home/GAURAV/HIVE#
```

OUTPUT:



All partitions in hive is there as directories.Loading in hive is instantaneous process and it won't trigger a Map/Reduce job.That's why our file is stored asUserLog.txt instead of 00000_o file. Please follow the article as I will show in dynamic partition where we will LOAD table using another table where Map/reduce job is triggered.

DYNAMIC PARTITIONING:

Let us see now the CODE of Dynamic Partitioning. We will create new table T_USER_LOG_DYN for dynamic partition and also as we told earlier that we will load this table using a new table, let's create another table T_USER_LOG_SRC.

Data for the source table

1001	John	Google	2016-04-27	US	
1001	John	Facebook	2016-04-27	US	
1002	Eric	Instagram	2016-04-28	US	
1002	Eric	Google	2016-04-28	UK	
1003	Annie	Facebook	2016-04-28	UK	
1004	Sara	Instagram	2016-04-28	US	
1005	Wei	Google	2016-04-29	CHINA	
1006	Ming	Facebook	2016-04-29	CHINA	
1007	Li	Instagram	2016-04-29	CHINA	
1008	Sota	Google	2016-04-29	JAPAN	
1009	Yuto	Facebook	2016-04-27	JAPAN	
1010	Ryota	Instagram	2016-04-28	JAPAN	
1011	Gaurav	Google	2016-04-29	INDIA	
1012	Anu	Facebook	2016-04-27	INDIA	
1013	Maya	Instagram	2016-04-28	INDIA	
1014	Rohit	Google	2016-04-28	INDIA	
1015	Michel	Facebook	2016-04-28	AUSTRALIA	
1016	Ricky	Instagram	2016-04-28	AUSTRALIA	

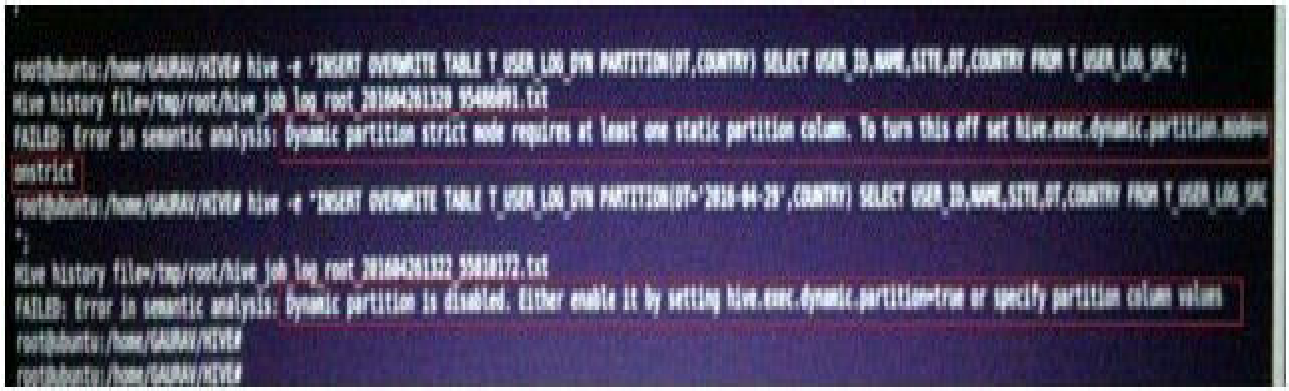
hive script for table DDL

```
DROP TABLE IF EXISTS T_USER_LOG_DYN;

CREATE TABLE T_USER_LOG_DYN (USER_ID INT
                            ,NAME STRING
                            ,SITE STRING
                            )
PARTITIONED BY (DT STRING,COUNTRY STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
;

DROP TABLE IF EXISTS T_USER_LOG_SRC;

CREATE TABLE T_USER_LOG_SRC (USER_ID INT
                            ,NAME STRING
                            ,SITE STRING
                            ,DT STRING
                            ,COUNTRY STRING
                            )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t'
STORED AS TEXTFILE
;

LOAD DATA LOCAL INPATH 'UserLogSrc.txt' OVERWRITE INTO TABLE T_USER_LOG_SRC;
```

setting hive properties

SET hive.exec.dynamic.partition= true;

SET hive.exec.dynamic.partition.mode= nonstrict



While loading the table, the partition will be created dynamically on all

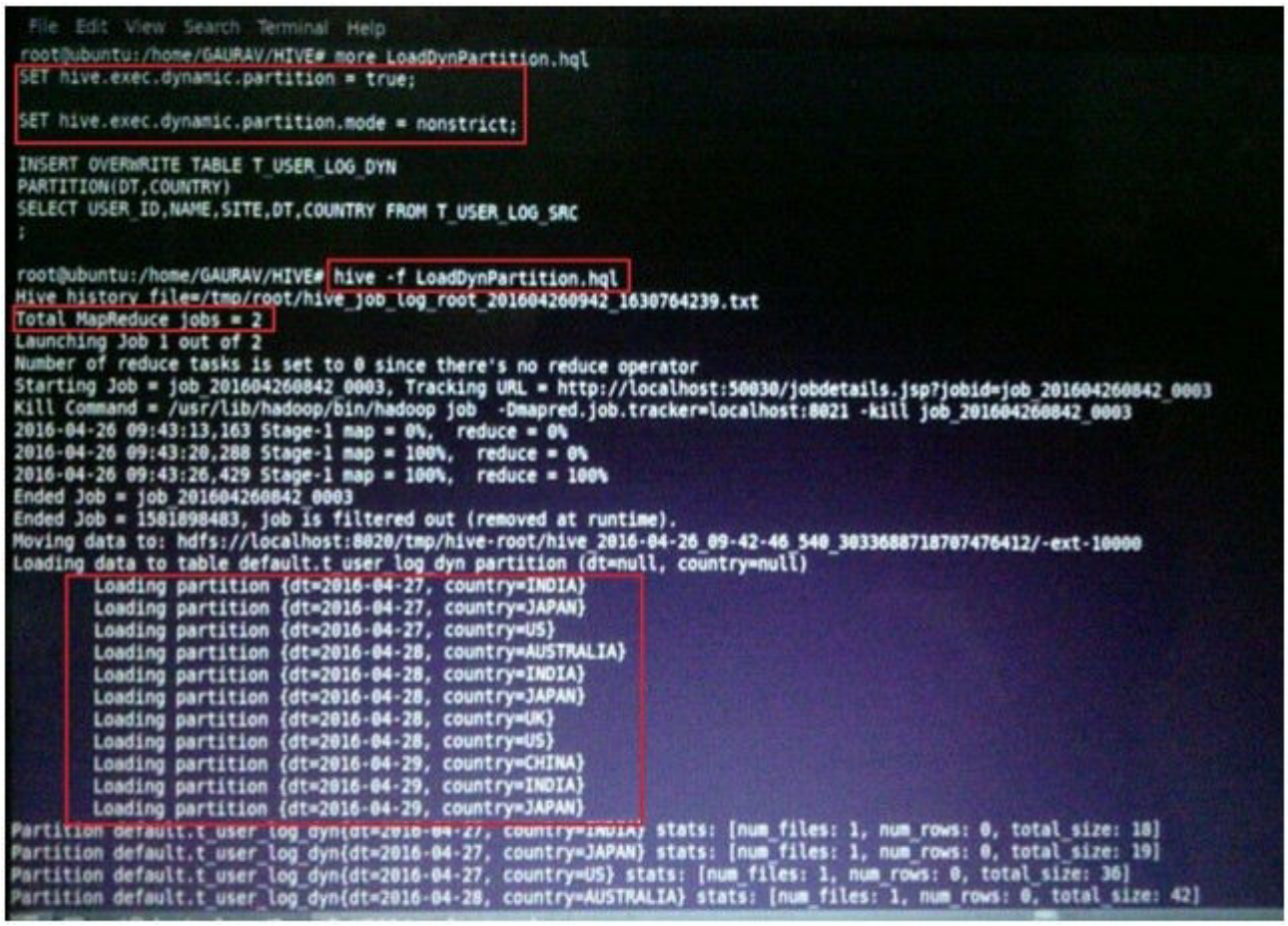
partition columns if hive.exec.dynamic.partition.mode= nonstrict is set.

If it is strict [which is by default], it will need at least one partition column to be defined in CODE.

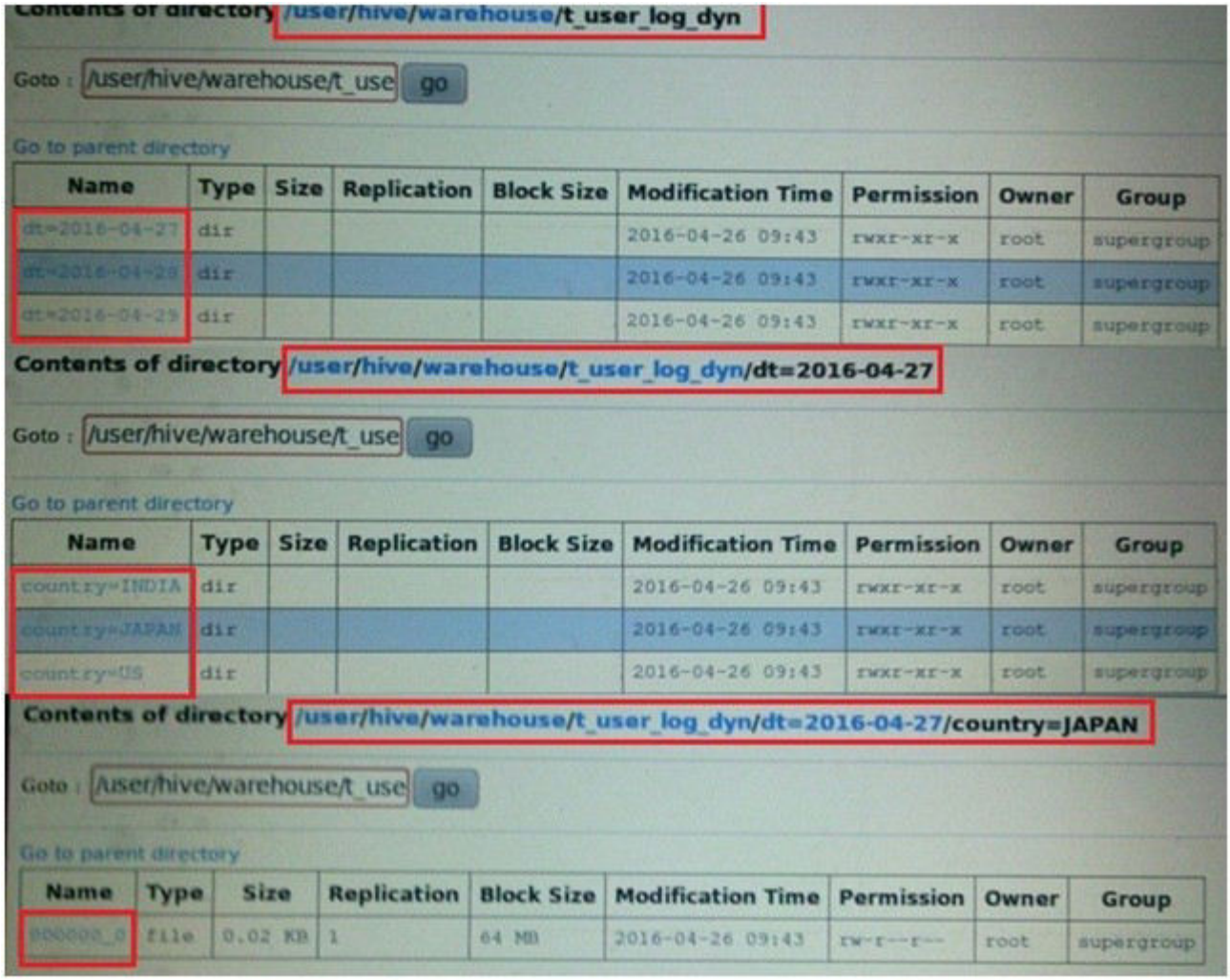
CODE:

loading the table T_USER_LOG_DYN using the data from T_USER_LOG_SRC and creating dynamic partitions.

LoadDynPartition.hql loads the table dynamically as shown below:



OUTPUT:



Advantages of Hive Partition

- Distribute execution load horizontally
- Faster execution of queries in case of partition with low volume of data. e.g. Get the population from “Vatican city” returns very fast instead of searching entire population of world.
- No need to search entire table columns for a single record.

Disadvantages with Hive Partition

- there is a possibility for creating too many folders in HDFS that is extra burden for Namenode metadata.

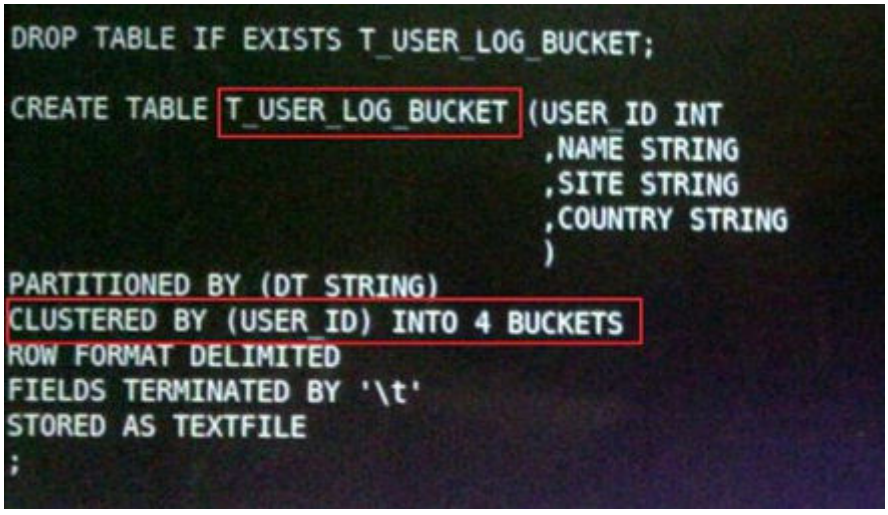
Effective for low volume data for a given partition. But some queries like group by on high volume of data still take long time to execute.

Example:

Grouping of population of China will take long time compared to grouping of population in Vatican city. Partition is not solving responsiveness problem in case of data skewing towards a particular partition value.So there is no guarantee for query optimization for all the times.

BUCKETING in HIVE

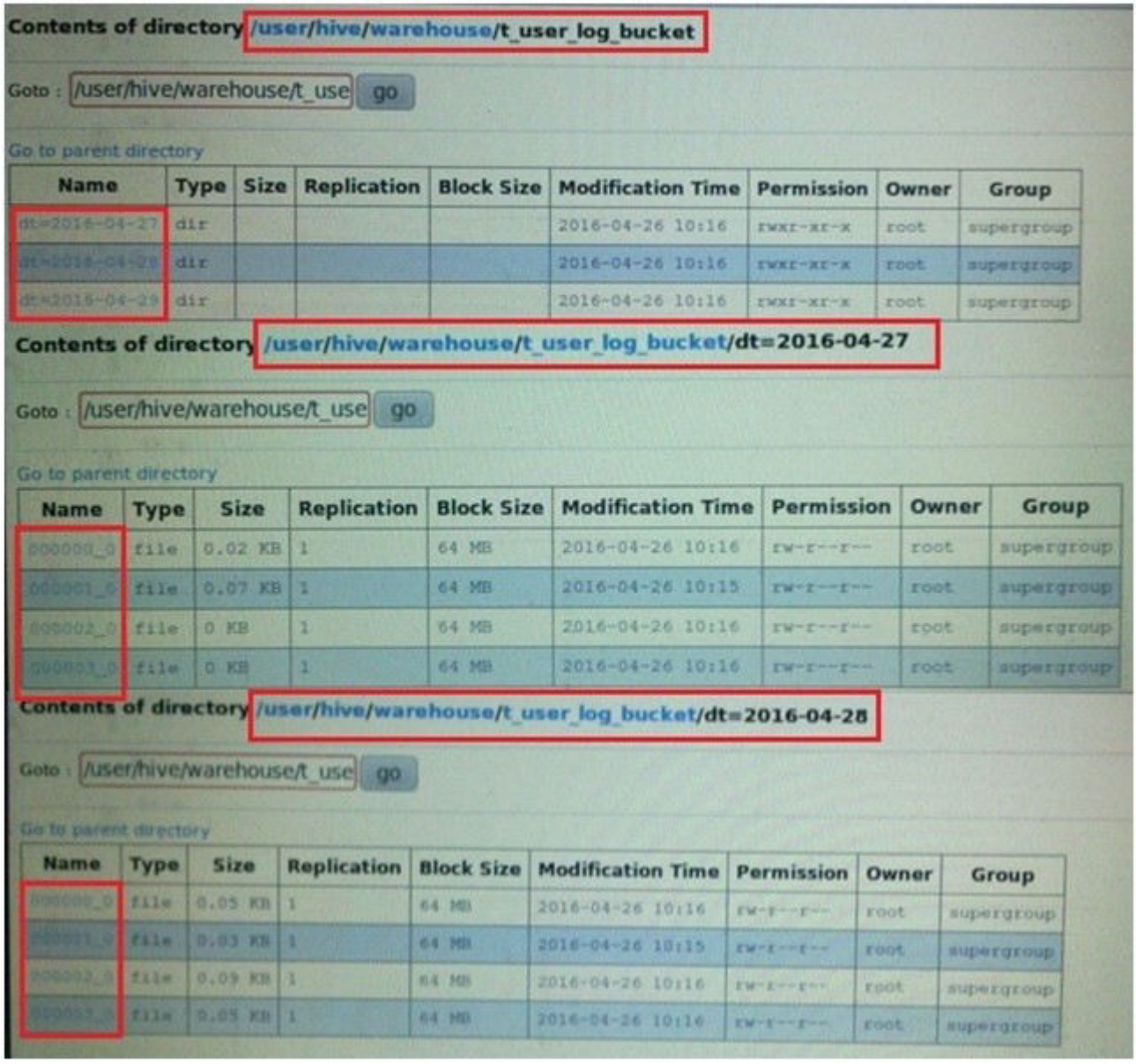
When we write data in bucketed table in hive, it places the data in distinct buckets as files. Hive uses some hashing algorithm to generate a number in range of 1 to N buckets [as mentioned in DDL] and based on the result of hashing, data is placed in a particular buckets as a file. Let's create a hive bucketed table T_USER_LOG_BUCKET with a partition column as DT and having 4 buckets. We specify bucketing column in CLUSTERED BY (column_name) clause in hive table DDL.



```
SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

SET hive.exec.max.dynamic.partitions.pernode=1000;

SET hive.enforce.bucketing=true;

INSERT OVERWRITE TABLE T_USER_LOG_BUCKET PARTITION(DT)

SELECT USER_ID,NAME,SITE,COUNTRY,DT FROM T_USER_LOG_SRC;
```

```
File  Edit  View  Search  Terminal  Help

SELECT USER_ID,NAME,SITE,COUNTRY,DT FROM T_USER_LOG_SRC;


root@ubuntu:/home/GAURAV/HIVE# hive -f BucketTable.hql
Hive history file=/tmp/root/hive_job_log_root_201604261014_710318457.txt
OK
Time taken: 20.867 seconds
OK
Time taken: 1.09 seconds
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 4
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201604260842_0004, Tracking URL = http://localhost:50030/jobdetails.jsp?jobid=job_201604260842_000
Kill Command = /usr/lib/hadoop/bin/hadoop job  -Dmapred.job.tracker=localhost:8021 -kill job_201604260842_0004
2016-04-26 10:15:21,884 Stage-1 map = 0%,  reduce = 0%
2016-04-26 10:15:38,290 Stage-1 map = 100%,  reduce = 0%
2016-04-26 10:16:01,533 Stage-1 map = 100%,  reduce = 25%
2016-04-26 10:16:02,558 Stage-1 map = 100%,  reduce = 50%
2016-04-26 10:16:23,062 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_201604260842_0004
Loading data to table default.t_user_log_bucket partition (dt=null)
        Loading partition {dt=2016-04-27}
        Loading partition {dt=2016-04-28}
        Loading partition {dt=2016-04-29}
Partition default.t_user_log_bucket{dt=2016-04-27} stats: [num_files: 4, num_rows: 0, total_size: 91]
Partition default.t_user_log_bucket{dt=2016-04-28} stats: [num_files: 4, num_rows: 0, total_size: 228]
Partition default.t_user_log_bucket{dt=2016-04-29} stats: [num_files: 4, num_rows: 0, total_size: 119]
Table default.t_user_log_bucket stats: [num_partitions: 3, num_files: 12, num_rows: 0, total_size: 438]
18 Rows loaded to t_user_log_bucket
OK
Time taken: 89.025 seconds
root@ubuntu:/home/GAURAV/HIVE#
```

OUTPUT:



Contents of directory /user/hive/warehouse/t_user_log_bucket

Goto : /user/hive/warehouse/t_use go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
dt=2016-04-27	dir				2016-04-26 10:16	rwxr-xr-x	root	supergroup
dt=2016-04-28	dir				2016-04-26 10:16	rwxr-xr-x	root	supergroup
dt=2016-04-29	dir				2016-04-26 10:16	rwxr-xr-x	root	supergroup

Contents of directory /user/hive/warehouse/t_user_log_bucket/dt=2016-04-27

Goto : /user/hive/warehouse/t_use go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
000000_0	file	0.02 KB	1	64 MB	2016-04-26 10:16	rw-r--r--	root	supergroup
000001_0	file	0.07 KB	1	64 MB	2016-04-26 10:15	rw-r--r--	root	supergroup
000002_0	file	0 KB	1	64 MB	2016-04-26 10:16	rw-r--r--	root	supergroup
000003_0	file	0 KB	1	64 MB	2016-04-26 10:16	rw-r--r--	root	supergroup

Contents of directory /user/hive/warehouse/t_user_log_bucket/dt=2016-04-28

Goto : /user/hive/warehouse/t_use go

Go to parent directory

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
000000_0	file	0.05 KB	1	64 MB	2016-04-26 10:16	rw-r--r--	root	supergroup
000001_0	file	0.03 KB	1	64 MB	2016-04-26 10:15	rw-r--r--	root	supergroup
000002_0	file	0.09 KB	1	64 MB	2016-04-26 10:16	rw-r--r--	root	supergroup
000003_0	file	0.05 KB	1	64 MB	2016-04-26 10:16	rw-r--r--	root	supergroup