# Hive table creation (types of tables, partitions, bucketing)

**Tables**

There are 2 types of tables in HIVE

1.Managed Table

2.External Table

**Managed Table**

**It is the type of table that are owned and managed by Hive whenever we create a Internal table and load data from Hdfs path,the entire data get transferred from hdfs location to hive warehouse location while if we load data from local location a copy of that file will be made in the table directory created.Thus if we use Managed Table the data will be moved from hdfs location to hive warehouse location.**

**Another thing is that if we use a Managed table and if we delete the table the entire data will be deleted.**

**Third thing is that if we use a managed Table only the file will be moved but the directory will remain which will be awkward if we use output of a mapreduce**

**Creating managed table employee data**

```
hive> create table employee_table(Name String,Skill String,id int,company String) row format delimited  fields terminated by
    > ',' ;
OK
Time taken: 0.107 seconds
```

**Inserting data into that table by using load command**

```
hive> Load data inpath '/emp_details.txt' into table employee_table;
Loading data to table default.employee_table
OK
Time taken: 0.527 seconds
hive>
```

**BEFORE LOADING DATA**

**We can see that the emp_details.txt present**

```
[acadgild@localhost ~]$ hadoop fs  -ls /
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which m
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
17/04/19 15:46:57 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 5 items
-rw-r--r--   1 acadgild supergroup        437 2017-04-13 10:42 /TemperatureDataset.txt
-rw-r--r--   1 acadgild supergroup        159 2017-04-19 15:45 /emp_details.txt
drwxr-xr-x   - acadgild supergroup          0 2017-04-13 10:08 /hive
drwxrwx---   - acadgild supergroup          0 2016-08-16 19:16 /tmp
drwxr-xr-x   - acadgild supergroup          0 2017-04-04 09:01 /user
```

**After loading data**

**The data is moved from hdfs location to directory created in table name as shown**

```
asses where applicable
Found 5 items
-rw-r--r--   1 acadgild supergroup        437 2017-04-13 10:42 /TemperatureDataset.txt
-rw-r--r--   1 acadgild supergroup        159 2017-04-19 15:45 /emp_details.txt
drwxr-xr-x   - acadgild supergroup          0 2017-04-13 10:08 /hive
drwxrwx---   - acadgild supergroup          0 2016-08-16 19:16 /tmp
drwxr-xr-x   - acadgild supergroup          0 2017-04-04 09:01 /user
[acadgild@localhost ~]$ hadoop fs  -ls /
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/na
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it wi
17/04/19 16:06:52 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your p
asses where applicable
Found 4 items
-rw-r--r--   1 acadgild supergroup        437 2017-04-13 10:42 /TemperatureDataset.txt
drwxr-xr-x   - acadgild supergroup          0 2017-04-13 10:08 /hive
drwxrwx---   - acadgild supergroup          0 2016-08-16 19:16 /tmp
drwxr-xr-x   - acadgild supergroup          0 2017-04-04 09:01 /user
```

```
[acadgild@localhost ~]$ hadoop fs -ls /user/hive/warehouse
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
17/04/19 16:21:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
asses where applicable
Found 5 items
drwxrwxr-x   - acadgild supergroup          0 2016-08-18 00:56 /user/hive/warehouse/college
drwxrwxr-x   - acadgild supergroup          0 2017-04-13 09:45 /user/hive/warehouse/custom.db
drwxrwxr-x   - acadgild supergroup          0 2017-04-19 16:00 /user/hive/warehouse/employee_data
drwxrwxr-x   - acadgild supergroup          0 2017-04-19 16:06 /user/hive/warehouse/employee_table
drwxrwxr-x   - acadgild supergroup          0 2016-08-12 15:05 /user/hive/warehouse/use
[acadgild@localhost ~]$ hadoop fs -ls /user/hive/warehouse/employee_table
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
17/04/19 16:21:39 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
asses where applicable
Found 1 items
-rwxrwxr-x   1 acadgild supergroup        159 2017-04-19 15:45 /user/hive/warehouse/employee_table/emp_details.txt
[acadgild@localhost ~]$ hadoop fs -cat  /user/hive/warehouse/employee_table/emp_details.txt
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
17/04/19 16:22:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java
asses where applicable
Amit,Big Data,1,BBSR
Venkat,Web Technology,2,BBSR
Aditya,DBA,1,BNG
Ravinder,Java,2,BBSR
Sunil,C#,1,BBSR
Anil,ASP,2,BNG
Mihir,Big Data,3,BBSR
Mohit,Java,1,BBSR
```

**Drop Table**

**On dropping the table the entire data get deleted**

**Deleting table**

```
hive> drop table employee_table;
OK
Time taken: 1.769 seconds
```

**Before and After deleting table**

**After deleting table the data gets missed**

```
drwxrwxr-x   - acadgild supergroup          0 2016-08-18 00:56 /user/hive/warehouse/college
drwxrwxr-x   - acadgild supergroup          0 2017-04-13 09:45 /user/hive/warehouse/custom.db
drwxrwxr-x   - acadgild supergroup          0 2017-04-19 16:00 /user/hive/warehouse/employee_data
drwxrwxr-x   - acadgild supergroup          0 2017-04-19 16:25 /user/hive/warehouse/employee_table
drwxrwxr-x   - acadgild supergroup          0 2016-08-12 15:05 /user/hive/warehouse/use
[acadgild@localhost ~]$ hadoop fs -ls  /user/hive/warehouse
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecst
17/04/19 16:27:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... usin
asses where applicable
Found 4 items
drwxrwxr-x   - acadgild supergroup          0 2016-08-18 00:56 /user/hive/warehouse/college
drwxrwxr-x   - acadgild supergroup          0 2017-04-13 09:45 /user/hive/warehouse/custom.db
drwxrwxr-x   - acadgild supergroup          0 2017-04-19 16:00 /user/hive/warehouse/employee_data
drwxrwxr-x   - acadgild supergroup          0 2016-08-12 15:05 /user/hive/warehouse/use
```

**External Table**

**It is the type of table in which table only contains the metadata but not the actual data but the problem with external table is that since it has only the metadata the data should be available in HDfs and not in local location**

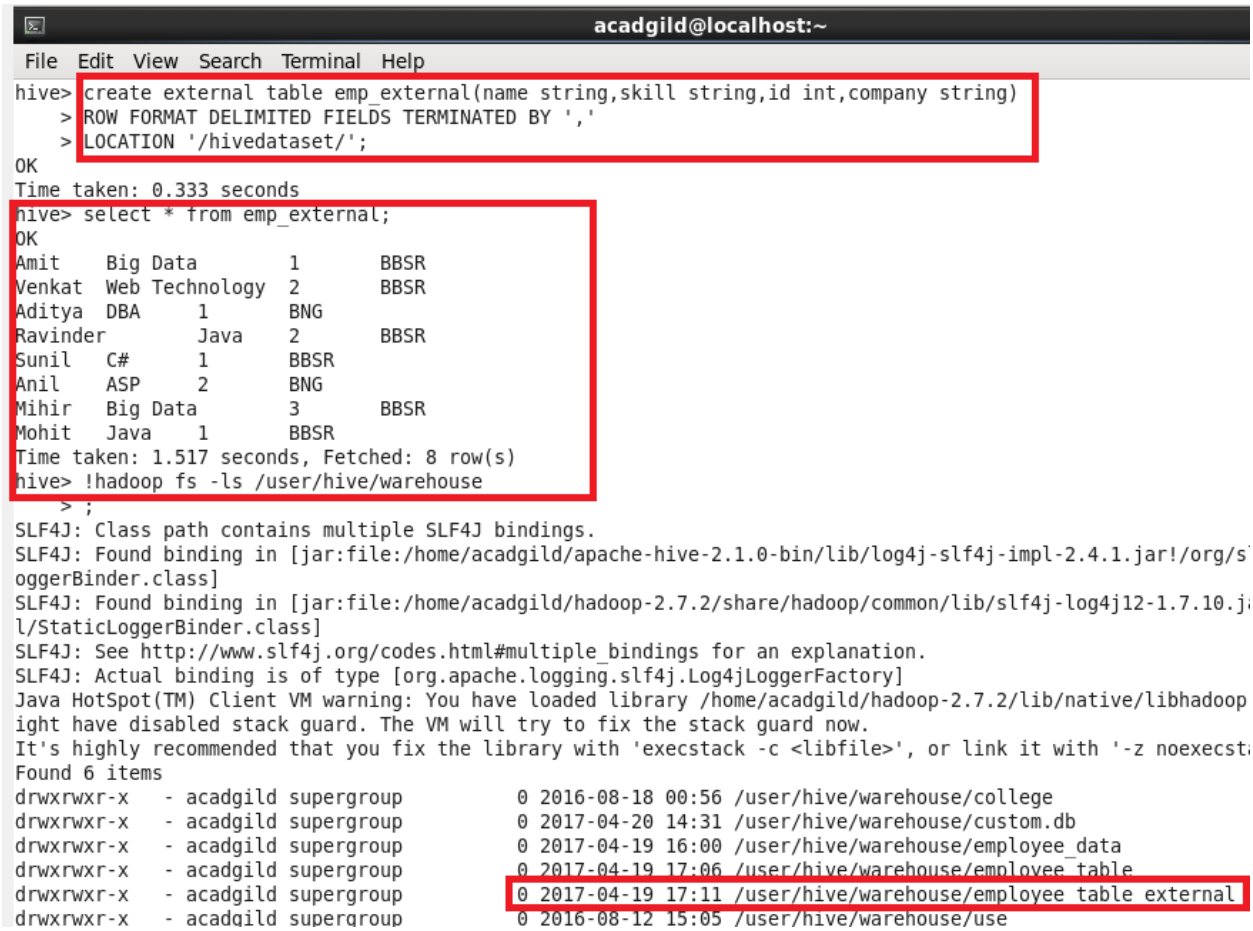
**I will use the same data for external table.So I will put the file into hdfs**

```
[acadgild@localhost Desktop]$ hadoop fs -put /home/acadgild/Desktop/emp_details.txt /
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which m
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
17/04/19 16:41:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
[acadgild@localhost Desktop]$
```

acadgild@localhost:~

**Creating external Table**

**External table can be created by using external word before Table as shown and loading data by directly giving the location**




**Unlike managed table where file will be moved from hdfs to hive/warehouse here data remains with the parent location**

```
hive> describe formatted emp_external;
OK
# col_name              data_type               comment

name                    string
skill                   string
id                      int
company                 string

# Detailed Table Information
Database:               default
Owner:                  acadgild
CreateTime:             Thu Apr 20 16:59:30 IST 2017
LastAccessTime:         UNKNOWN
Retention:              0
Location:               hdfs://localhost:9000/hivedataset
Table Type:             EXTERNAL_TABLE
Table Parameters:
        EXTERNAL                TRUE
        numFiles                1
        totalSize               159
        transient_lastDdlTime   1492687770

# Storage Information
SerDe Library:          org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:            org.apache.hadoop.mapred.TextInputFormat
OutputFormat:           org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:             No
Num Buckets:            -1
Bucket Columns:         []
Sort Columns:           []
Storage Desc Params:
        field.delim             ,
        serialization.format    ,
Time taken: 0.281 seconds, Fetched: 32 row(s)
```

```
college
emp_external
employee_data
use
Time taken: 0.037 seconds, Fetched: 4 row(s)
hive> !hadoop fs -ls /
    > ;
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.4.1.ja
oggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-
l/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/l
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z
Found 6 items
-rw-r--r--   1 acadgild supergroup        437 2017-04-13 10:42 /TemperatureDataset.txt
-rw-r--r--   1 acadgild supergroup        159 2017-04-19 17:29 /emp_details.txt
drwxr-xr-x   - acadgild supergroup          0 2017-04-13 10:08 /hive
drwxr-xr-x   - acadgild supergroup          0 2017-04-20 16:55 /hivedataset
drwxrwx---   - acadgild supergroup          0 2016-08-16 19:16 /tmp
drwxr-xr-x   - acadgild supergroup          0 2017-04-04 09:01 /user
hive> !hadoop fs -ls /hivedataset;
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.4.1.ja
oggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-
l/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/l
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z
Found 1 items
-rw-r--r--   1 acadgild supergroup        159 2017-04-20 16:55 /hivedataset/emp_details.txt
```

**Thus if a External Table is dropped only metadata and the schema gets dropped and the data does not get dropped**

```
hive> drop table emp_external;
OK
Time taken: 1.627 seconds
hive> show tables;
OK
college
employee_data
use
Time taken: 0.044 seconds, Fetched: 3 row(s)
hive> !hadoop fs -ls /hivedataset;
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf4j/impl/StaticL
oggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/imp
l/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.so.1.0.0 which m
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
Found 1 items
-rw-r--r--   1 acadgild supergroup        159 2017-04-20 16:55 /hivedataset/emp_details.txt
hive>
```

**Partioning:**

**Working of Partitioning**

Since during querrying in hive for a large dataset,for a simplequerry say we want HOUSES with beds=2 it want to search the entire database and find the result which will take a large time .So in order to optimize querrying time Partition is introduced

In Partitioning,the data is divided into directories based on column specified under Partition so that while querring the hive searches form that directory (eg ) if we use people from beds=2, It will go directly to that directory and the operation will be performed

**Creating Partitioned Table**:It is similar to ordinary table but just adding Partitioned By() where column based on which partitioning needs to be done

Here we will create table with partition for which we will add data from a large realestate data based on bedrooms and flat_type


**Static Partitioning**

**Scenario:**

**WE have a real estate database for which we want a separate list of only data with bedroom=2**

**So we will create a static partitioning of bed=2**

**Thus in static partitioning,we know the type of data for example if we know the data is of bedroom with 2 we will create a partitioning into the partitioned table by giving the following command by giving type as residential and beds as2**

```
hive> insert into table sep_list partition(type='Residential',Beds=2) select City,Baths,Sq__ft,Price from realEstate where Beds=2 and type='Res
idential';
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e.
spark, tez) or using Hive 1.X releases.
Query ID = somanath_20170419071058_c59a7d10-4e37-42f0-826b-58afb768b1a8
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1492565022638_0001, Tracking URL = http://somanath-HP-Notebook:8088/proxy/application_1492565022638_0001/
Kill Command = /usr/local/hadoop/hadoop-2.7.0/bin/hadoop job  -kill job_1492565022638_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-04-19 07:11:09,140 Stage-1 map = 0%,  reduce = 0%
2017-04-19 07:11:17,039 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.66 sec
MapReduce Total cumulative CPU time: 3 seconds 660 msec
Ended Job = job_1492565022638_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/sep_list/type=Residential/beds=2/.hive-staging_hive_2017-04-19_07-10-58_714_
2184028071921798518-1/-ext-10000
Loading data to table default.sep_list partition (type=Residential, beds=2)
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.66 sec   HDFS Read: 118260 HDFS Write: 2559 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 660 msec
OK
```

**Output:we can see that a separate directory for residential as type and bed=2 is created which contains all the data of beds=2**

```
somanath@somanath-HP-Notebook:~$ hadoop fs -ls /user/hive/warehouse/sep_list
Found 1 items
drwxrwxr-x   - somanath supergroup          0 2017-04-19 07:10 /user/hive/warehouse/sep_list/type=Residential
somanath@somanath-HP-Notebook:~$ hadoop fs -ls /user/hive/warehouse/sep_list/type=Residential
Found 5 items
drwxrwxr-x   - somanath supergroup          0 2017-04-19 07:11 /user/hive/warehouse/sep_list/type=Residential/beds=2
drwxrwxr-x   - somanath supergroup          0 2017-04-18 20:26 /user/hive/warehouse/sep_list/type=Residential/beds=3
drwxrwxr-x   - somanath supergroup          0 2017-04-18 20:26 /user/hive/warehouse/sep_list/type=Residential/beds=4
drwxrwxr-x   - somanath supergroup          0 2017-04-18 20:26 /user/hive/warehouse/sep_list/type=Residential/beds=5
drwxrwxr-x   - somanath supergroup          0 2017-04-18 20:26 /user/hive/warehouse/sep_list/type=Residential/beds=6
somanath@somanath-HP-Notebook:~$ hadoop fs -cat /user/hive/warehouse/sep_list/type=Residential/beds=2
cat: `/user/hive/warehouse/sep_list/type=Residential/beds=2': Is a directory
somanath@somanath-HP-Notebook:~$ hadoop fs -ls /user/hive/warehouse/sep_list/type=Residential/beds=2
Found 1 items
-rwxrwxr-x   4 somanath supergroup       2460 2017-04-19 07:11 /user/hive/warehouse/sep_list/type=Residential/beds=2/000000_0
somanath@somanath-HP-Notebook:~$ hadoop fs -ls /user/hive/warehouse/sep_list/type=Residential/beds=2/000000_0
-rwxrwxr-x   4 somanath supergroup       2460 2017-04-19 07:11 /user/hive/warehouse/sep_list/type=Residential/beds=2/000000_0
somanath@somanath-HP-Notebook:~$ hadoop fs -cat /user/hive/warehouse/sep_list/type=Residential/beds=2/000000_0
SACRAMENTO,1,796,68880
SACRAMENTO,1,852,69307
SACRAMENTO,1,797,81900
SACRAMENTO,2,1022,108750
RIO LINDA,1,844,113263
SACRAMENTO,1,588,120000
ANTELOPE,2,1043,161250
SACRAMENTO,2,1341,221000
POLLOCK PINES,2,1284,280908
SACRAMENTO,1,1126,292024
GOLD RIVER,2,1520,299000
ORANGEVALE,1,1690,334150
SACRAMENTO,1,800,78000
SACRAMENTO,1,746,78400
SACRAMENTO,1,868,90000
SACRAMENTO,1,610,93675
SACRAMENTO,1,1220,98000
SACRAMENTO,2,967,114800
SACRAMENTO,1,952,134000
SACRAMENTO,1,722,145000
ELK GROVE,2,1006,152000
SACRAMENTO,1,810,156000
ELK GROVE,2,1123,156000
```

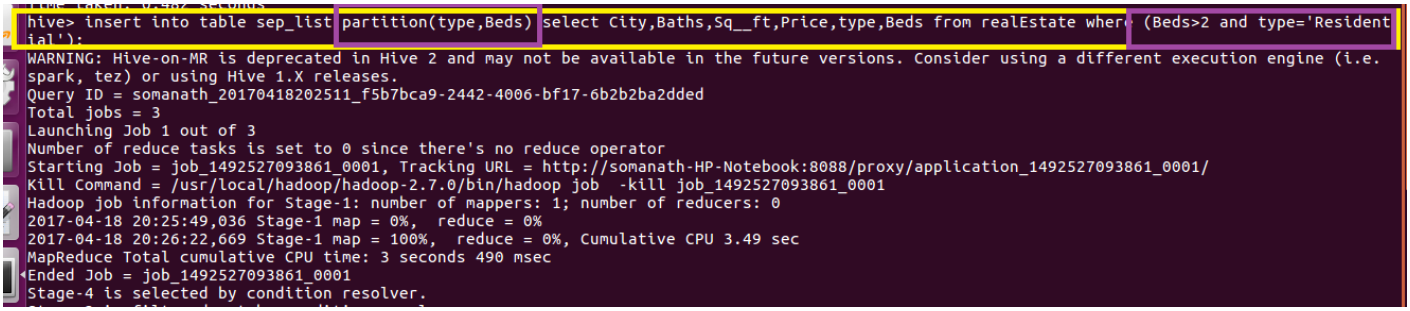
**Drawback: the major problem with static partitioning is that suppose the dataset is large and contains house with bedroom from 2 to 6 .**

**So we want to type the same SQL  querry 5 times with beds=3, beds=4, beds=5, beds=6 if we use static partitioning**

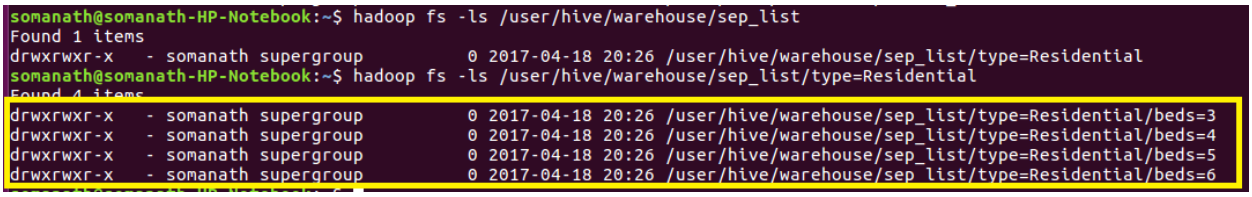
**So for this we use dynamic partitioning in which we will not specify the beds as 2 but we will just mention the columns on which partition to be done and the columns will be added as last two column in select statement so that hive will automatically do partitioning**

**Dynamic Partitioning**

**So if I want the partitioning to be done on houses on bedrooms greater than 2 I will just specify the columns based on which partitioning need to be done and the hive will automatically do partitioning with beds=3, beds=4, beds=5, beds=6 as shown**




**Output**




**Bucketting**

In order to Increase the performance of queries Partitions are introduced in hive.So if there is a huge dataset regarding “world population” and suppose if we want to filter data by each country using where hive has to scan the entire dataset.**To ensure faster querying Partioning is made on country name** and

**Now for each country a directory will be created on hive/warehouse and the querying can be faster**

**Limitation with hive partitions:**

**No1:**

**If the dataset is so large and now for each country a directory will be created which will cause a increased overload on namenode**

**No2:**

**Now think of the above example where partitions are nade on country.Since population varies the Data say 100 GB for say 100 countries will not be equal.So again the processing on these partition will increase time if we use a group by like operation So to encounter these issues hive provides BUCKETTING**

**First problem is encountered as bucketing creates this much number of buckets so whatever may be the size the entire data will be divided among these buckets**

**2 Problem is encountered by since hashcode is code in the range of 1 to 10 say 10000 records all these data will be divided within these equally as 1000**

**Similarly If we want to further classify partitioned data,bucketing can be made over partitioned data and the bucketed record will be stored as files within the directory**

**So from above scenario we can define bucketing as**

**Hive partition divides table into number of partitions and these partitions can be further subdivided into more manageable parts known as Buckets or Clusters. The Bucketing concept is based on Hash function, which depends on the type of the bucketing column. Records which are bucketed by the same column will always be saved in the same bucket.**

**In this example as you can see after inserting into table emp_table 2 separate directories will be created**

```
hive> insert into table emp_table partition(location) select * from emp_table1;
WARNING: Hive on MR is deprecated in Hive 2 and may not be available in the future versions. Co
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20170421164327_e41d97d4-ca32-4a11-b0d9-0f271406a3ac
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1492761713744_0001, Tracking URL = http://localhost:8088/proxy/application_1
Kill Command = /home/acadgild/hadoop-2.7.2/bin/hadoop job  -kill job_1492761713744_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2017-04-21 16:43:41,658 Stage-1 map = 0%,  reduce = 0%
2017-04-21 16:43:47,296 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.86 sec
MapReduce Total cumulative CPU time: 860 msec
Ended Job = job_1492761713744_0001
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to directory hdfs://localhost:9000/user/hive/warehouse/emp_table/.hive-staging_hive
53346240277368-1/-ext-10000
Loading data to table default.emp_table partition (location=null)

Loaded : 2/2 partitions.
        Time taken to load dynamic partitions: 1.645 seconds
        Time taken for adding to write entity : 0.001 seconds
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 0.86 sec   HDFS Read: 4439 HDFS Write: 257 SUCCESS
Total MapReduce CPU Time Spent: 860 msec
OK
Time taken: 23.171 seconds
hive>
```

**When we go inside the directory the files will be of varying size as shown**

/user/hive/warehouse/emp_table Go!

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|
| drwxr-xr-x | acadgild | supergroup | 0 B | 4/21/2017, 4:43:46 PM | 0 | 0 B | location=BBSR |
| drwxr-xr-x | acadgild | supergroup | 0 B | 4/21/2017, 4:43:46 PM | 0 | 0 B | location=BNG |

/user/hive/warehouse/emp_table/location=BBSR Go!

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|
| -rwxr-xr-x | acadgild | supergroup | 97 B | 4/21/2017, 4:43:46 PM | 1 | 128 MB | 000000_0 |

emp_details.txt | 000000_0

```
Amit,Big Data,1
Venkat,Web Technology,2
Ravinder,Java,2
Sunil,C#,1
Mihir,Big Data,3
Mohit,Java,1
```

/user/hive/warehouse/emp_table/location=BNG Go!

| Permission | Owner | Group | Size | Last Modified | Replication | Block Size | Name |
|---|---|---|---|---|---|---|---|
| -rwxr-xr-x | acadgild | supergroup | 24 B | 4/21/2017, 4:43:46 PM | 1 | 128 MB | 000000_0 |

**But if we use bucketing and specify the bucketing as 2**

```
hive> create table emp_table_bucket(name String,skill string,bus_no int,location string ) clustered by (location)
    > into 2 buckets row format delimited fields
    > terminated by ',';
OK
Time taken: 0.077 seconds
```

```
hive> insert into table emp_table_bucket select * from emp_table1;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a
tion engine (i.e. spark, tez) or using Hive 1.X releases.
Query ID = acadgild_20170421173403_435aac84-29ff-432f-95cf-53491d0f6faa
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 2
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 mapreduce.job.reduces=<number>
Starting Job = job_1492761713744_0002, Tracking URL = http://localhost:8088/proxy/application_1492761713744_00
Kill Command = /home/acadgild/hadoop-2.7.2/bin/hadoop job  -kill job_1492761713744_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 2
2017-04-21 17:34:11,857 Stage-1 map = 0%,  reduce = 0%
2017-04-21 17:34:17,411 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 0.92 sec
2017-04-21 17:34:28,934 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 2.02 sec
MapReduce Total cumulative CPU time: 2 seconds 20 msec
Ended Job = job_1492761713744_0002
Loading data to table default.emp_table_bucket
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 2   Cumulative CPU: 3.15 sec   HDFS Read: 12645 HDFS Write: 290 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 150 msec
OK
Time taken: 27.928 seconds
```

```
hive> !hadoop fs -ls /user/hive/warehouse/emp table bucket;
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/acadgild/apache-hive-2.1.0-bin/lib/log4j-slf4j-impl-2.4.1.jar!/org/slf
oggerBinder.class]
SLF4J: Found binding in [jar:file:/home/acadgild/hadoop-2.7.2/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar
l/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Java HotSpot(TM) Client VM warning: You have loaded library /home/acadgild/hadoop-2.7.2/lib/native/libhadoop.s
ight have disabled stack guard. The VM will try to fix the stack guard now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstac
Found 2 items
-rwxr-xr-x   1 acadgild supergroup          0 2017-04-21 17:34 /user/hive/warehouse/emp_table_bucket/000000_0
-rwxr-xr-x   1 acadgild supergroup        159 2017-04-21 17:34 /user/hive/warehouse/emp_table_bucket/000001_0
```

**we can see that files will be created and no additional directories will not be created**

**Bucketting Vs Partitioning**

| **BUCKETTING** | **PARTITIONING** |
|---|---|
| bucketing helps in organizing data in each partition into multiple files, so that the same set of data is always written in same bucket. | Partitioning helps in elimination of data, if used in WHERE clause |
| **Divides the records into file based on hashcode** | **Divides the records among directories based on key specified** |
| **It results in creation of files** | **It results in creation of Directories** |
| **It does not result in memory overhead of namenode as however large the record may be only specified files will be created**<br>**Say if we define 4 buckets only 4 files will be created** | **It will result in memory overhead as it depends on the type of Data** |
| **Equal distribution of data** | **Unequal distribution of Data** |
| **Since the data files are equal sized parts, map-side joins will be faster on the bucketed tables and sampling will also be faster.** | |