**ASSIGNMENT 26.2**

**Explain in breif with an example**

**● Bucketing**

**● Bucketing V/S Partitioning**

**● Sampling**

**BUCKETING:**

Partition helps in increasing the efficiency when performing a query on a table. Instead of scanning the whole table, it will only scan for the partitioned set and does not scan or operate on the unpartitioned sets, which helps us to provide results in lesser time and the details will be displayed very quickly because of Hive Partition.

there is a huge [dataset](https://acadgild.com/big-data/big-data-development-training-certification). At times, even after partitioning on a particular field or fields, the partitioned file size doesn’t match with the actual expectation and remains huge and we want to manage the partition results into different parts. To overcome this problem of partitioning, Hive provides Bucketing concept, which allows user to divide table data sets into more manageable parts.

Thus, Bucketing helps user to maintain parts that are more manageable and user can set the size of the manageable parts or Buckets too.

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.

Here, CLUSTERED BY clause is used to divide the table into buckets.

In [Hive Partition,](http://www.hadooptpoint.com/introduction-hive-partition-big-data/)each partition will be created as directory. But in Hive Buckets, each bucket will be created as file.

**Advantages of Bucketing:**

Bucketed tables allows much more efficient [sampling](https://cwiki.apache.org/confluence/display/Hive/LanguageManual+Sampling) than the non-bucketed tables. With sampling, we can try out queries on a section of data for testing and debugging purpose when the original data sets are very huge. Here, the user can fix the size of buckets according to the need.

Bucketing concept also provides the flexibility to keep the records in each bucket to be sorted by one or more columns. Since the data files are equal sized parts, map-side joins will be faster on the bucketed tables.

Bucketing is helpful for 2 reasons

•Enables more efficient queries

•Makes sampling more efficient

•Hash(column) MOD(number of buckets) –evenly distributed

**Bucketing V/S Partitioning:**

Partitioning data is often used for distributing load horizontally, this has performance benefit, and helps in organizing data in a logical fashion. Example: if we are dealing with a large employee table and often run queries with WHERE clauses that restrict the results to a particular country or department . For a faster query response Hive table can be PARTITIONED BY (country STRING, DEPT STRING).

If query limits for employee from country=ABC, it will only scan the contents of one directory country=ABC. This can dramatically improve query performance, but only if the partitioning scheme reflects common filtering. Partitioning feature is very useful in Hive, however, a design that creates too many partitions may optimize some queries, but be detrimental for other important queries. Other drawback is having too many partitions is the large number of Hadoop files and directories that are created unnecessarily and overhead to NameNode since it must keep all metadata for the file system in memory.

you have to be careful when partitioning, because if you for instance partition by employee\_id and you have millions of employees, you'll end up having millions of directories in your file system. The term '**cardinality**' refers to the number of possible value a field can have. For instance, if you have a 'country' field, the countries in the world are about 300, so cardinality would be ~300. For a field like 'timestamp\_ms', which changes every millisecond, cardinality can be billions. In general, when choosing a field for partitioning, it should not have a high cardinality, because you'll end up with way too many directories in your file system.

**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.Clustering aka bucketing on the other hand, will result with a fixed number of files, since you do specify the number of buckets. What hive will do is to take the field, calculate a hash and assign a record to that bucket. But what happens if you use let's say 256 buckets and the field you're bucketing on has a low cardinality (for instance, it's a US state, so can be only 50 different values) ? You'll have 50 buckets with data, and 206 buckets with no data.

So, bucketing works well when the field has high cardinality and data is evenly distributed among buckets. Partitioning works best when the cardinality of the partitioning field is not too high.

**Sampling:**

Sampling is concerned with the selection of a subset of data from a large dataset to run queries and verify results. The dataset may be too large to run queries on the whole data. Therefore in development and testing phases it is a good idea to run queries on a sample of dataset.

**There are two types of sampling. They are**

* **Sampling by bucketing**
* **Block sampling**

**Sampling by Bucketing**

We can use TABLESAMPLE clause to bucket the table on the given column and get data from only some of the buckets.

    TABLESAMPLE (BUCKET x OUT OF y [ON colname])

Column name indicates the column to be used to bucket the data into y buckets[1-y]. All the rows which are in the bucket x are returned.

If the table is not bucketed on the column(s) used in sampling, TABLESAMPLE will scan the entire table and fetch the sample.

If the hive table is bucketed on some column(s), then we can directly use that column(s) to get a sample. In this case Hive need not read all the data to generate sample as the data is already organized into different buckets using the column(s) used in the sampling query. Hive will read data only from some buckets as per the size specified in the sampling query.

**Block Sampling**

Block sampling allows Hive to select at least n% data from the whole dataset. Sampling granularity is at the HDFS block size level. If HDFS block size is 64MB and n% of input size is only 10MB, then 64MB of data is fetched.

TABLESAMPLE (n PERCENT)