# ***HADOOP***

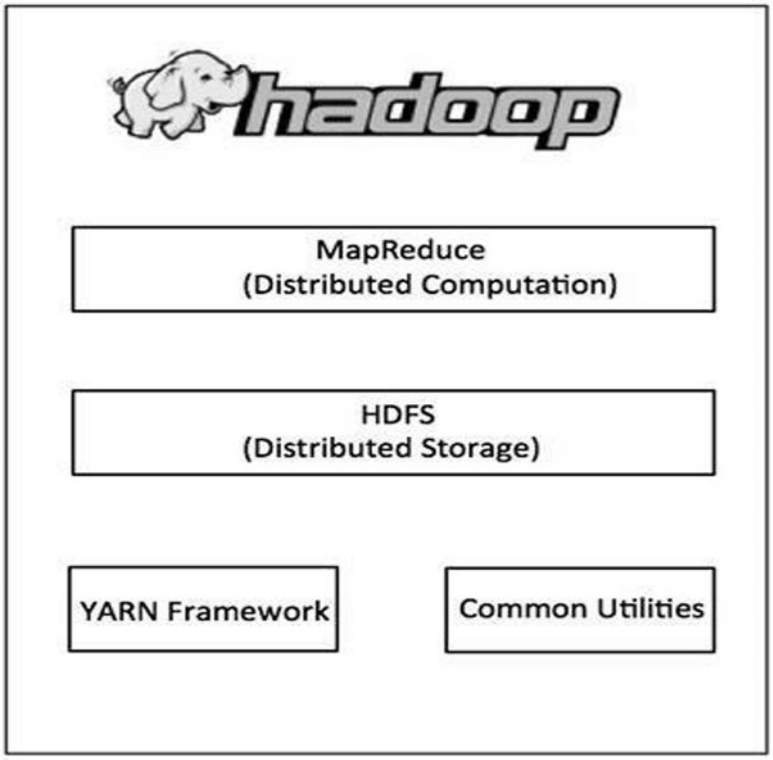
*Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.*

## ***HADOOP ARCHITECTURE***

*At its core, Hadoop has two major layers namely:*

*(a) Processing/Computation layer (MapReduce).*

*(b) Storage layer (Hadoop Distributed File System).*



## *Hadoop Distributed File System*

*The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on commodity hardware.*

*It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. It is highly fault-tolerant and is designed to be deployed on low-cost hardware.*

*It provides high throughput access to application data and is suitable for applications having large datasets. Apart from the above-mentioned two core components,*

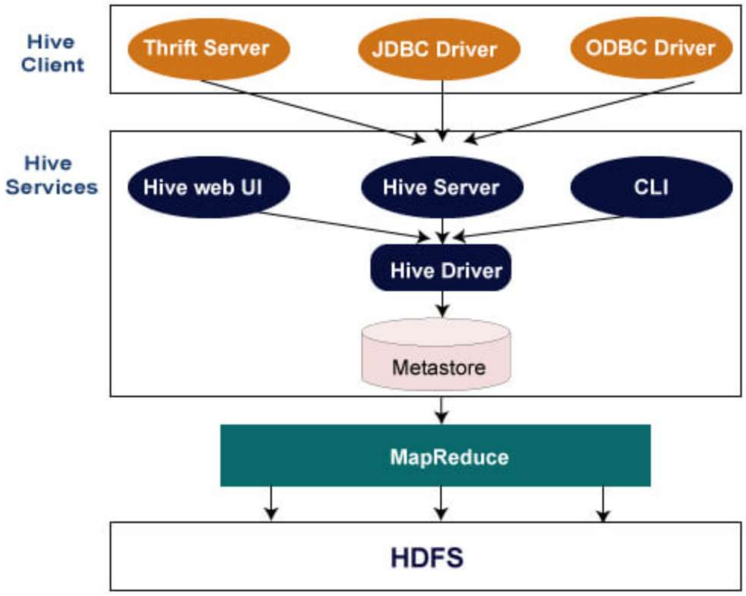
*Hadoop framework also includes the following two modules:*

- ***Hadoop Common**: These are Java libraries and utilities required by other Hadoop modules.*
- ***Hadoop YARN**: This is a framework for job scheduling and cluster resource management.*

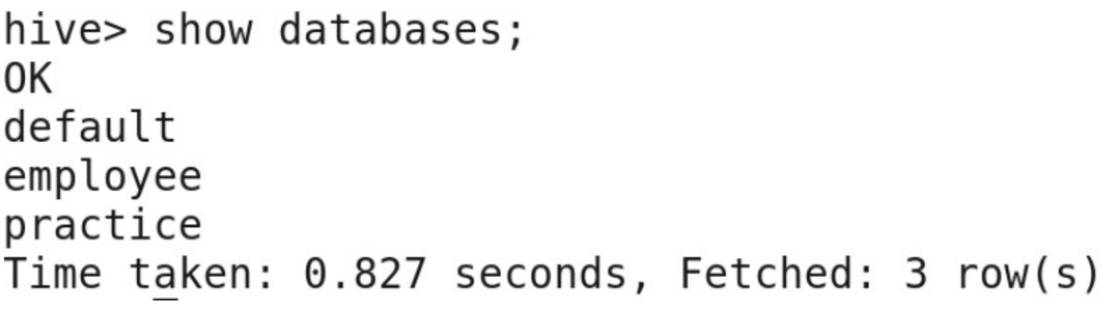
## *HIVE :*

*The Apache Hive  is a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale and facilitates reading, writing, and managing petabytes of data residing in distributed storage using SQL.*

## *Hive Architecture*

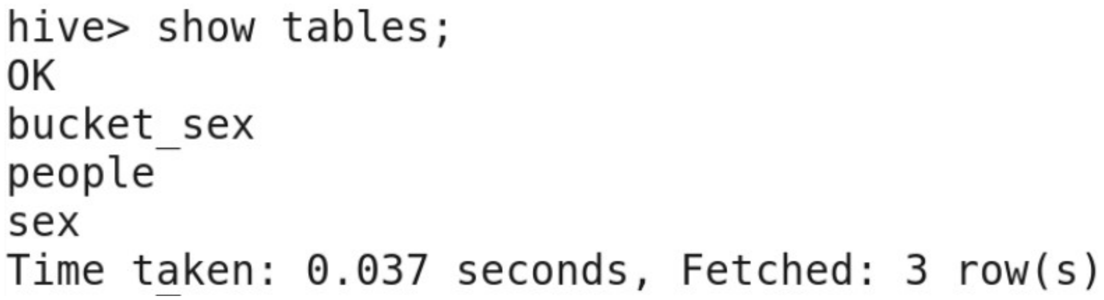


**To Check the Database inside the HDFS :**



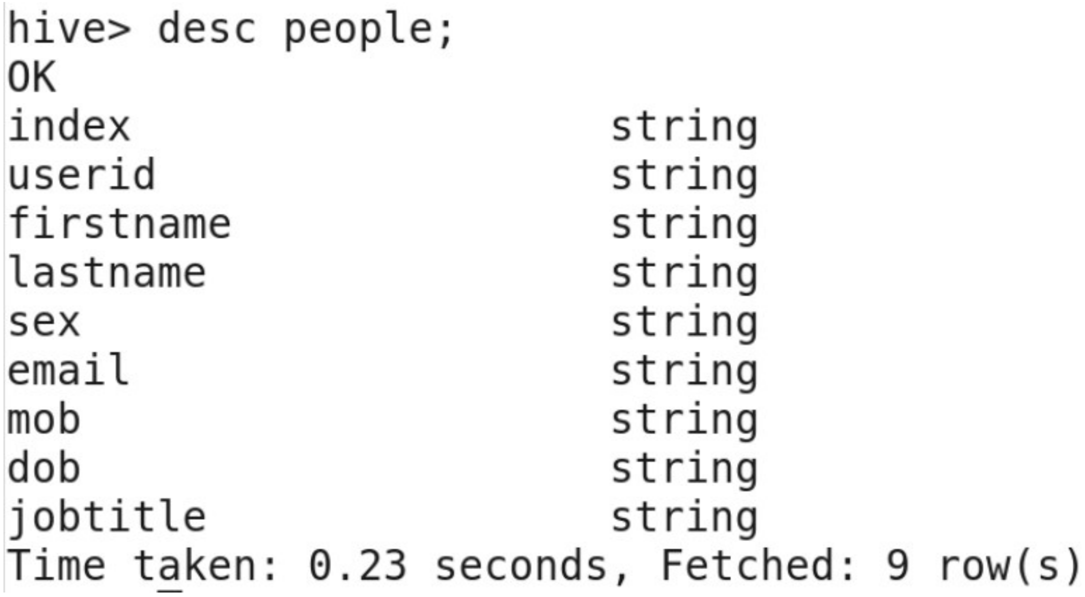
```
hive> show databases;
OK
default
employee
practice
Time taken: 0.827 seconds, Fetched: 3 row(s)
```

**To Check the table inside the Database inside the HDFS :**



```
hive> show tables;
OK
bucket_sex
people
sex
Time taken: 0.037 seconds, Fetched: 3 row(s)
```

**To Check the schema of the table in HDFS:**



```
hive> desc people;
OK
index                   string
userid                  string
firstname               string
lastname                string
sex                     string
email                   string
mob                     string
dob                     string
jobtitle                string
Time taken: 0.23 seconds, Fetched: 9 row(s)
```

**To Create a table inside the HDFS :**

```
hive> create table sample(index string,userid string,firstname string,lastname string,sex string,email string,mob string,dob string,jobtitle string)
> row format delimited fields terminated by ',';
OK
```

```
Time taken: 1.65 seconds
```

**To Load data inside the created table using a csv or tsv file :**

```
hive> load data local inpath '/home/cloudera/Training/19.07.23/people-100.csv' into table sample;
Loading data to table practice.sample
Table practice.sample stats: [numFiles=1, totalSize=11099]
OK
Time taken: 0.884 seconds
```

**To Drop a table from the HDFS :**

```
hive> drop table sample;
OK
Time taken: 0.692 seconds
```

## Partitioning of the table

**To Create a table inside the HDFS using Partitions :**

```
hive> create table sex(index string,userid string,firstname string,lastname string,email string,mob string,dob string,jobtitle string)
    > partitioned by (sex string)
    > row format delimited fields terminated by ',';
OK
Time taken: 0.084 seconds
```

**Enable Permissions to accept the partitions :**

```
hive> set hive.exec.dynamic.partition = true;
hive> set hive.exec.dynamic.partition.mode = nonstrict;
```

**Insert data inside the partitions table :**

```
hive> insert overwrite table sex
    > partition(sex)
    > select index,userid,firstname,lastname,email,mob,dob,jobtitle,sex from people;
Query ID = cloudera_20230725000202_0797619b-5691-4f2c-b5a8-f258f6554c64
```

```
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_1690266933628_0001, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1690266933628_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1690266933628_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-07-25 00:03:11,130 Stage-1 map = 0%,  reduce = 0%
2023-07-25 00:03:19,858 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.12 sec
MapReduce Total cumulative CPU time: 3 seconds 120 msec
Moving data to: hdfs://quickstart.cloudera:8020/user/hive/warehouse/practice.db/sex/.hive-
staging_hive_2023-07-25_00-02-57_382_4486635514177853212-1/-ext-10000
Loading data to table practice.sex partition (sex=null)
         Time taken for load dynamic partitions : 362
        Loading partition {sex=Male}
        Loading partition {sex=Female}
         Time taken for adding to write entity : 1
Partition practice.sex{sex=Female} stats: [numFiles=1, numRows=53, totalSize=5318, rawDataSize=5265]
Partition practice.sex{sex=Male} stats: [numFiles=1, numRows=47, totalSize=4741, rawDataSize=4694]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.12 sec   HDFS Read: 16078 HDFS Write: 10182 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 120 msec
OK
Time taken: 25.514 seconds
```

## To Check the Partitions created:

```
hive> dfs -ls /user/hive/warehouse/practice.db/sex;
Found 2 items
drwxrwxrwx   - cloudera supergroup          0 2023-07-25 00:03
/user/hive/warehouse/practice.db/sex/sex=Female
drwxrwxrwx   - cloudera supergroup          0 2023-07-25 00:03
/user/hive/warehouse/practice.db/sex/sex=Male
```

## To Check the Contents inside the partitions created:

```
hive> dfs -ls /user/hive/warehouse/practice.db/sex/sex=Female ;

Found 1 items
-rwxrwxrwx   1 cloudera supergroup       5318 2023-07-25 00:03
/user/hive/warehouse/practice.db/sex/sex=Female/000000_0

hive> dfs -cat /user/hive/warehouse/practice.db/sex/sex=Female/000000_0 ;
```

## Bucketing of the table

### To Create a table inside the HDFS using Buckets :

```
hive> create table bucket_sex(index string,userid string,firstname string,lastname string,sex
string,email string,mob string,dob string,jobtitle string)
    > clustered by (sex) into 2 buckets
    > row format delimited fields terminated by ',';
OK
Time taken: 0.089 seconds
```

### Enable Permissions to accept the Bucketing :

```
hive> set hive.enforce.bucketing = true;
```

### Insert data inside the Buckets table :

```
hive> insert overwrite table bucket_sex
    > select * from people;
Query ID = cloudera_20230725001212_a0ebabce-afdb-4138-9a6a-1e9bdde18400
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_1690266933628_0002, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1690266933628_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1690266933628_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 2
2023-07-25 00:12:44,960 Stage-1 map = 0%,  reduce = 0%
2023-07-25 00:12:53,519 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.44 sec
2023-07-25 00:13:03,486 Stage-1 map = 100%,  reduce = 50%, Cumulative CPU 7.7 sec
2023-07-25 00:13:04,580 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 11.39 sec
MapReduce Total cumulative CPU time: 11 seconds 390 msec
Ended Job = job_1690266933628_0002
Loading data to table practice.bucket_sex
```

```
Table practice.bucket_sex stats: [numFiles=2, numRows=100, totalSize=10665,
rawDataSize=10565]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 2   Cumulative CPU: 11.39 sec   HDFS Read: 24646 HDFS Write:
10819 SUCCESS
Total MapReduce CPU Time Spent: 11 seconds 390 msec
OK
Time taken: 28.894 seconds
```

**To Check the Buckets created:**

```
hive> dfs -ls /user/hive/warehouse/practice.db/bucket_sex;
Found 2 items
-rwxrwxrwx   1 cloudera supergroup       5689 2023-07-25 00:13
/user/hive/warehouse/practice.db/bucket_sex/000000_0
-rwxrwxrwx   1 cloudera supergroup       4976 2023-07-25 00:13
/user/hive/warehouse/practice.db/bucket_sex/000001_0
```

**To Check the Contents inside the Buckets created:**

```
hive> dfs -cat /user/hive/warehouse/practice.db/bucket_sex/000000_0;
```

***Store the tables in different file formats :***

***Text file :***

```
hive> create table sample_textfile(index string,userid string,firstname string,lastname
string,sex string,email string,mob string,dob string,jobtitle string)
    > row format delimited fields terminated by ','
    > stored as textfile;
OK
Time taken: 0.086 seconds
```

***Inserting the data inside the Textfile Format :***

```
hive> insert overwrite table sample_textfile
    > select * from people;
Query ID = cloudera_20230725004141_341770d5-c803-4211-97d8-60f54347e004
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_1690266933628_0004, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1690266933628_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1690266933628_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-07-25 00:41:30,743 Stage-1 map = 0%,  reduce = 0%
2023-07-25 00:41:38,156 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.39 sec
MapReduce Total cumulative CPU time: 3 seconds 390 msec
Ended Job = job_1690266933628_0004
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:
hdfs://quickstart.cloudera:8020/user/hive/warehouse/practice.db/sample_textfile/.hive-
staging_hive_2023-07-25_00-41-24_810_7553595839077842798-1/-ext-10000
Loading data to table practice.sample_textfile
Table practice.sample_textfile stats: [numFiles=1, numRows=100, totalSize=10665,
rawDataSize=10565]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.39 sec   HDFS Read: 15744 HDFS Write: 10749
SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 390 msec
OK
Time taken: 14.832 seconds
```

***Sequence file:***

```
hive> create table sample_sequencefile(index string,userid string,firstname string,lastname
string,sex string,email string,mob string,dob string,jobtitle string)
    > row format delimited fields terminated by ','
    > stored as sequencefile;
OK
Time taken: 0.09 seconds
```

***Inserting the data inside the Sequence File Format :***

```
hive> insert overwrite table sample_sequencefile
    > select * from people;
Query ID = cloudera_20230725004242_927084c1-5f32-45e6-9676-f405fff14e49
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_1690266933628_0005, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1690266933628_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1690266933628_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-07-25 00:42:08,717 Stage-1 map = 0%,  reduce = 0%
2023-07-25 00:42:14,140 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.66 sec
MapReduce Total cumulative CPU time: 1 seconds 660 msec
Ended Job = job_1690266933628_0005
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:
hdfs://quickstart.cloudera:8020/user/hive/warehouse/practice.db/sample_sequencefile/.hive-staging_hive_2023-07-25_00-42-00_714_6855151151982995237-1/-ext-10000
Loading data to table practice.sample_sequencefile
Table practice.sample_sequencefile stats: [numFiles=1, numRows=100, totalSize=12054,
rawDataSize=10565]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.66 sec   HDFS Read: 15882 HDFS Write: 12142
SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 660 msec
OK
Time taken: 14.921 seconds
```

***<u>Parquet file:</u>***

```
hive> create table sample_parquet(index string,userid string,firstname string,lastname
string,sex string,email string,mob string,dob string,jobtitle string)
    > row format delimited fields terminated by ','
    > stored as parquet;
OK
Time taken: 0.087 seconds
```

***<u>Inserting the data inside the Parquet Format :</u>***

```
hive> insert overwrite table sample_parquet
    > select * from people;
Query ID = cloudera_20230725004343_764900c1-8b34-4367-8139-ccb81842f189
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_1690266933628_0006, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1690266933628_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1690266933628_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-07-25 00:43:58,393 Stage-1 map = 0%,  reduce = 0%
2023-07-25 00:44:05,804 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 4.21 sec
MapReduce Total cumulative CPU time: 4 seconds 210 msec
Ended Job = job_1690266933628_0006
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:
hdfs://quickstart.cloudera:8020/user/hive/warehouse/practice.db/sample_parquet/.hive-
staging_hive_2023-07-25_00-43-52_408_2623996529817986814-1/-ext-10000
Loading data to table practice.sample_parquet
Table practice.sample_parquet stats: [numFiles=1, numRows=100, totalSize=13693,
rawDataSize=900]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 4.21 sec   HDFS Read: 16049 HDFS Write: 13774
SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 210 msec
OK
Time taken: 16.001 seconds
```

***Avro file:***

```
hive> create table sample_avro(index string,userid string,firstname string,lastname string,sex
string,email string,mob string,dob string,jobtitle string)
    > row format delimited fields terminated by ','
    > stored as avro;
OK
Time taken: 0.135 seconds
```

***Inserting the data inside the Avro Format :***

```
hive> insert overwrite table sample_avro
    > select * from people;
Query ID = cloudera_20230725015252_bffc841f-3fb5-40cf-891d-45248c591a60
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_1690266933628_0007, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1690266933628_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1690266933628_0007
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-07-25 01:52:49,057 Stage-1 map = 0%,  reduce = 0%
2023-07-25 01:52:56,600 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.65 sec
MapReduce Total cumulative CPU time: 2 seconds 650 msec
Ended Job = job_1690266933628_0007
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:
hdfs://quickstart.cloudera:8020/user/hive/warehouse/practice.db/sample_avro/.hive-
staging_hive_2023-07-25_01-52-41_291_7381657658923414155-1/-ext-10000
Loading data to table practice.sample_avro
Table practice.sample_avro stats: [numFiles=1, numRows=100, totalSize=12212,
rawDataSize=0]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 2.65 sec   HDFS Read: 16615 HDFS Write: 12272
SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 650 msec
OK
Time taken: 16.777 seconds
```

***Orc file :***

```
create table sample_orc(index string,userid string,firstname string,lastname string,sex
string,email string,mob string,dob string,jobtitle string)
   > row format delimited fields terminated by ','
   > stored as orc;
OK
Time taken: 0.104 seconds
```

***Inserting the data inside the Orc Format :***

```
hive> insert overwrite table sample_orc
   > select * from people;
Query ID = cloudera_20230725003737_a1af694d-0985-434c-840b-82eed82dcaf8
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
```

```
Starting Job = job_1690266933628_0003, Tracking URL =
http://quickstart.cloudera:8088/proxy/application_1690266933628_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1690266933628_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-07-25 00:37:45,803 Stage-1 map = 0%,  reduce = 0%
2023-07-25 00:37:53,245 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 1.98 sec
MapReduce Total cumulative CPU time: 1 seconds 980 msec
Ended Job = job_1690266933628_0003
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:
hdfs://quickstart.cloudera:8020/user/hive/warehouse/practice.db/sample_orc/.hive-
staging_hive_2023-07-25_00-37-36_191_5857662608120748751-1/-ext-10000
Loading data to table practice.sample_orc
Table practice.sample_orc stats: [numFiles=1, numRows=100, totalSize=6440,
rawDataSize=85000]
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 1.98 sec   HDFS Read: 16003 HDFS Write: 6519
SUCCESS
Total MapReduce CPU Time Spent: 1 seconds 980 msec
OK
Time taken: 18.552 seconds
```

***<u>Comparing the file size of all the different file formats which has same data inserted  :</u>***

```
hive> dfs -ls /user/hive/warehouse/practice.db/sample_avro ;
Found 1 items
-rwxrwxrwx   1 cloudera supergroup      12212 2023-07-25 01:52
/user/hive/warehouse/practice.db/sample_avro/000000_0

hive> dfs -ls /user/hive/warehouse/practice.db/sample_orc;
Found 1 items
-rwxrwxrwx   1 cloudera supergroup       6440 2023-07-25 00:37
/user/hive/warehouse/practice.db/sample_orc/000000_0

hive> dfs -ls /user/hive/warehouse/practice.db/sample_parquet;
Found 1 items
```

```
-rwxrwxrwx   1 cloudera supergroup      13693 2023-07-25 00:44
/user/hive/warehouse/practice.db/sample_parquet/000000_0

hive> dfs -ls /user/hive/warehouse/practice.db/sample_textfile;
Found 1 items
-rwxrwxrwx   1 cloudera supergroup      10665 2023-07-25 00:41
/user/hive/warehouse/practice.db/sample_textfile/000000_0

hive> dfs -ls /user/hive/warehouse/practice.db/sample_sequencefile;
Found 1 items
-rwxrwxrwx   1 cloudera supergroup      12054 2023-07-25 00:42
/user/hive/warehouse/practice.db/sample_sequencefile/000000_0
```

***JOIN*** *is a clause that is used for combining specific fields from two tables by using values common to each one. It is used to combine records from two or more tables in the database. It is more or less similar to SQL JOIN.*

***Syntax***

```
join_table:
table_reference JOIN table_factor [join_condition]
| table_reference {LEFT|RIGHT|FULL} [OUTER] JOIN table_reference join_condition
| table_reference LEFT SEMI JOIN table_reference join_condition
| table_reference CROSS JOIN table_reference [join_condition]
```

***Example***

*We will use the following two tables in this chapter. Consider the following table named CUSTOMERS.*

```
+----+--------------+-----------+----------------+-------------+

| ID | NAME     | AGE      | ADDRESS     | SALARY   |

+----+-------------+----------+------------------+------------+

| 1 | Ramesh    | 32       | Ahmedabad | 2000.00 |

| 2 | Khilan      | 25       | Delhi            | 1500.00 |

| 3 | kaushik     | 23      | Kota             | 2000.00 |

| 4 | Chaitali     | 25      | Mumbai       | 6500.00 |
| 5 | Hardik     | 27      | Bhopal         | 8500.00 |
| 6 | Komal      | 22      | MP             | 4500.00 |
| 7 | Muffy      | 24      | Indore         | 10000.00 |
+----+----------+-----+-----------+----------+
```

**Consider another table ORDERS as follows:**

```
+-----+---------------------+------------+--------+
|OID | DATE       | CUSTOMER_ID | AMOUNT |
+-----+---------------------+------------+--------+
| 102 | 2009-10-08 00:00:00 |     3 | 3000    |
| 100 | 2009-10-08 00:00:00 |     3 | 1500    |
| 101 | 2009-11-20 00:00:00 |     2 | 1560    |
| 103 | 2008-05-20 00:00:00 |     4 | 2060    |
+-----+---------------------+------------+--------+
```

*There are different types of joins given as follows:*

- *JOIN*
- *LEFT OUTER JOIN*
- *RIGHT OUTER JOIN*
- *FULL OUTER JOIN*

***JOIN***

*JOIN clause is used to combine and retrieve the records from multiple tables. JOIN is same as OUTER JOIN in SQL. A JOIN condition is to be raised using the primary keys and foreign keys of the tables.*

*The following query executes JOIN on the CUSTOMER and ORDER tables, and retrieves the records:*

```
hive> SELECT c.ID, c.NAME, c.AGE, o.AMOUNT
FROM CUSTOMERS c JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);
```

*On successful execution of the query, we get to see the following response:*

```
+----+----------+-----+--------+
| ID | NAME     | AGE | AMOUNT |
+----+----------+-----+--------+
| 3 | kaushik | 23 | 1500   |
| 2 | Khilan     | 25 | 1560      |
| 4 | Chaitali | 25 | 2060   |
+----+----------+-----+--------+
```

***LEFT OUTER JOIN***

*The HiveQL LEFT OUTER JOIN returns all the rows from the left table, even if there are no matches in the right table. This means, if the ON clause matches 0 zero records in the right table, the JOIN stil returns a row in the result, but with NULL in each column from the right table.*

*A LEFT JOIN returns all the values from the left table, plus the matched values from the right table, or NULL in case of no matching JOIN predicate.*

*The following query demonstrates LEFT OUTER JOIN between CUSTOMER and ORDER tables:*

```
hive> SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c
LEFT OUTER JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);
```

*On successful execution of the query, we get to see the following response:*

```
+----+----------+--------+---------------------+
| ID | NAME     | AMOUNT | DATE     |
+----+----------+--------+---------------------+
| 1 | Ramesh    | NULL     | NULL     |
| 2 | Khilan     | 1560     | 2009-11-20 00:00:00 |
| 3 | kaushik | 3000   | 2009-10-08 00:00:00 |
| 3 | kaushik | 1500   | 2009-10-08 00:00:00 |
| 4 | Chaitali | 2060   | 2008-05-20 00:00:00 |
| 5 | Hardik     | NULL     | NULL     |
| 6 | Komal     | NULL     | NULL     |
| 7 | Muffy     | NULL     | NULL     |
+----+----------+--------+---------------------+
```

### *RIGHT OUTER JOIN*

*The HiveQL RIGHT OUTER JOIN returns all the rows from the right table, even if there are no matches in the left table. If the ON clause matches 0 zero records in the left table, the JOIN still returns a row in the result, but with NULL in each column from the left table.*

*A RIGHT JOIN returns all the values from the right table, plus the matched values from the left table, or NULL in case of no matching join predicate.*

*The following query demonstrates RIGHT OUTER JOIN between the CUSTOMER and ORDER tables.*

```
notranslate"> hive> SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c RIGHT
OUTER JOIN ORDERS o ON c. ID = o. CUSTOMERID;
```

*On successful execution of the query, we get to see the following response:*

```
+------+----------+-----------------+---------------------+

| ID   | NAME     | AMOUNT | DATE     |

+------+----------+----------------+---------------------------+

| 3    | kaushik | 1500        | 2009-10-08 00:00:00 |

| 2    | Khilan     | 1560        | 2009-11-20 00:00:00 |

| 4    | Chaitali | 2060        | 2008-05-20 00:00:00 |

+------+----------+--------+------------------------------------+
```

### *FULL OUTER JOIN*

*The HiveQL FULL OUTER JOIN combines the  records of both the  left and the  right outer tables that fulfil the JOIN condition. The joined table contains either all the records from both the tables, or fills in NULL values for missing matches on either side.*

*The following query demonstrates FULL OUTER JOIN between CUSTOMER and ORDER tables:*

```
hive> SELECT c.ID, c.NAME, o.AMOUNT, o.DATE FROM CUSTOMERS c
FULL OUTER JOIN ORDERS o ON (c.ID = o.CUSTOMER_ID);
```

```
+------+----------+--------+---------------------+

| ID   | NAME     | AMOUNT | DATE     |

+------+----------+--------+---------------------+

| 1    | Ramesh  | NULL      | NULL      |

| 2    | Khilan     | 1560        | 2009-11-20 00:00:00 |

| 3    | kaushik | 3000        | 2009-10-08 00:00:00 |
| 3    | kaushik | 1500      | 2009-10-08 00:00:00 |
| 4    | Chaitali | 2060     | 2008-05-20 00:00:00 |
| 5    | Hardik   | NULL     | NULL       |
| 6    | Komal    | NULL     | NULL       |
| 7    | Muffy    | NULL     | NULL       |
| 3    | kaushik | 3000      | 2009-10-08 00:00:00 |
| 3    | kaushik | 1500      | 2009-10-08 00:00:00 |
| 2    | Khilan   | 1560     | 2009-11-20 00:00:00 |
| 4    | Chaitali | 2060     | 2008-05-20 00:00:00 |
+------+----------+--------+---------------------+
```