Hive Mini Project 1

Download Dataset 1:
https://drive.google.com/file/d/1WrG-9qv6atP-W3P_-gYln1hHyFKRKMHP/view

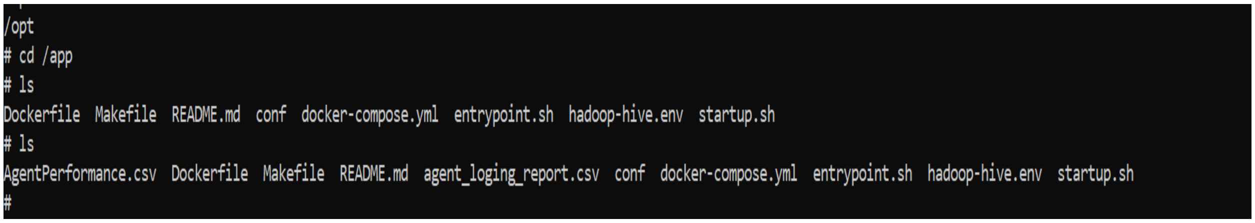
Download Dataset 2:
https://drive.google.com/file/d/1-JIPCZ34dyN6k9CqJa-Y8yxIGq6vTVXU/view

Note: both files are csv files.

Start a hive Shell using command hive inside the hive container

Our both csv files are store inside the local file system in container at
/app directory

/app directory is mounted on a volume in docker compose file



```
/opt
# cd /app
# ls
Dockerfile  Makefile  README.md  conf  docker-compose.yml  entrypoint.sh  hadoop-hive.env  startup.sh
# ls
AgentPerformance.csv  Dockerfile  Makefile  README.md  agent_loging_report.csv  conf  docker-compose.yml  entrypoint.sh  hadoop-hive.env  startup.sh
#
```

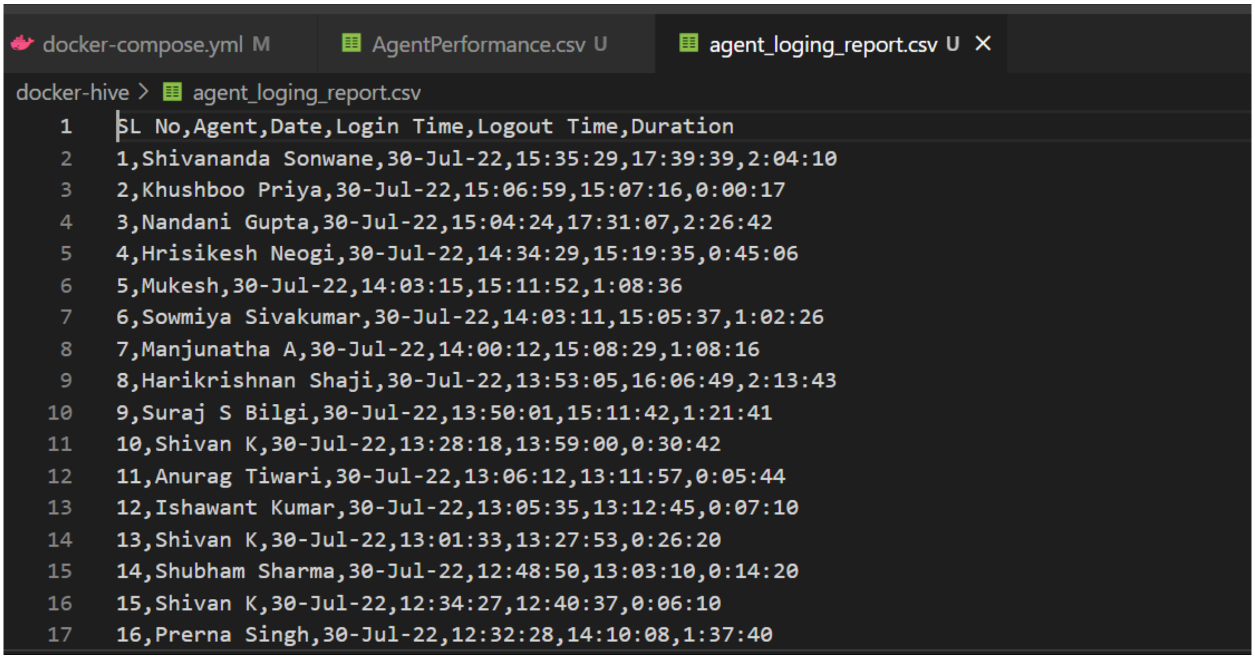
These files are

AgentPerformance.csv and agent_loging_report.csv

1. Create a schema based on the given dataset

Steps to create **agent_loging_report** table

This is how our **agent_login_report.csv** file looks like


docker-compose.yml M
AgentPerformance.csv U
agent_loging_report.csv U ×
docker-hive > agent_loging_report.csv

```
SL No,Agent,Date,Login Time,Logout Time,Duration
1,Shivananda Sonwane,30-Jul-22,15:35:29,17:39:39,2:04:10
2,Khushboo Priya,30-Jul-22,15:06:59,15:07:16,0:00:17
3,Nandani Gupta,30-Jul-22,15:04:24,17:31:07,2:26:42
4,Hrisikesh Neogi,30-Jul-22,14:34:29,15:19:35,0:45:06
5,Mukesh,30-Jul-22,14:03:15,15:11:52,1:08:36
6,Sowmiya Sivakumar,30-Jul-22,14:03:11,15:05:37,1:02:26
7,Manjunatha A,30-Jul-22,14:00:12,15:08:29,1:08:16
8,Harikrishnan Shaji,30-Jul-22,13:53:05,16:06:49,2:13:43
9,Suraj S Bilgi,30-Jul-22,13:50:01,15:11:42,1:21:41
10,Shivan K,30-Jul-22,13:28:18,13:59:00,0:30:42
11,Anurag Tiwari,30-Jul-22,13:06:12,13:11:57,0:05:44
12,Ishawant Kumar,30-Jul-22,13:05:35,13:12:45,0:07:10
13,Shivan K,30-Jul-22,13:01:33,13:27:53,0:26:20
14,Shubham Sharma,30-Jul-22,12:48:50,13:03:10,0:14:20
15,Shivan K,30-Jul-22,12:34:27,12:40:37,0:06:10
16,Prerna Singh,30-Jul-22,12:32:28,14:10:08,1:37:40
```

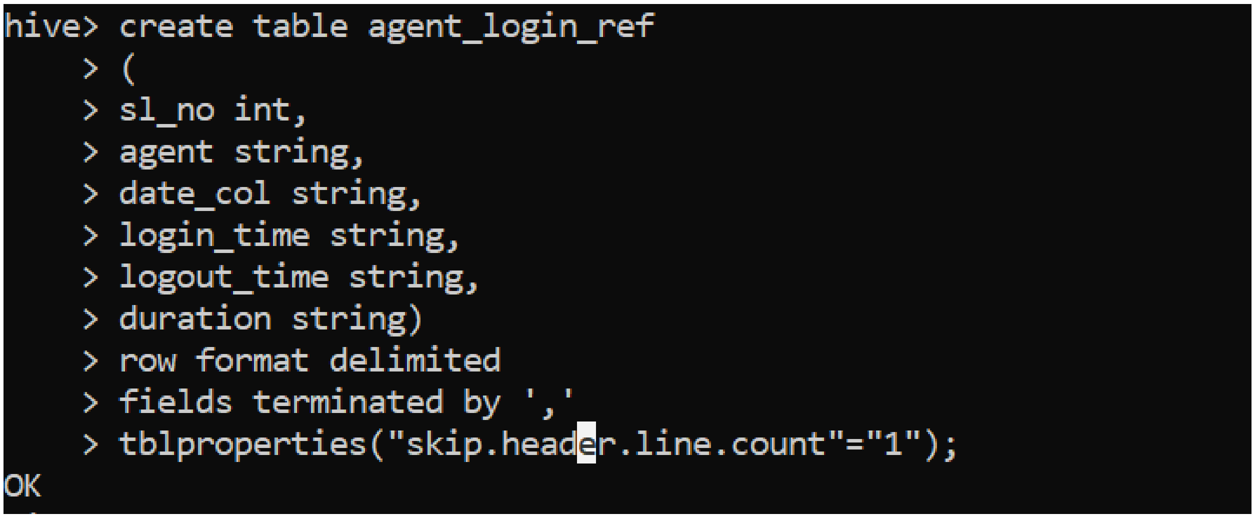
Date format is in **dd-mm-yy** format.

By **Default hive stores** the date in **yyyy-MM-dd** format

Approach

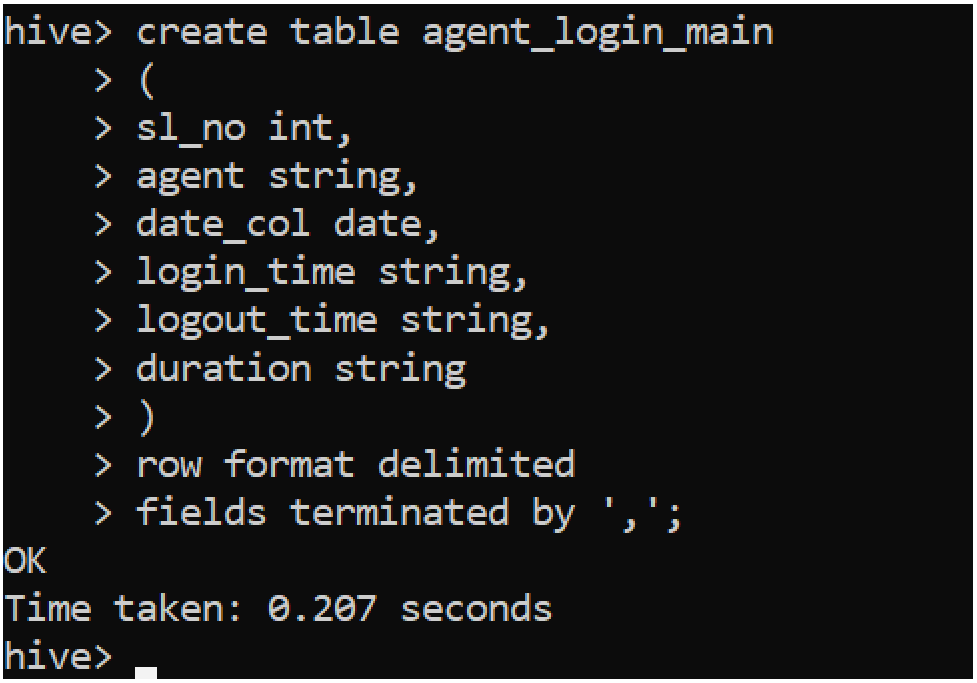
So we need to create a **reference table** first defining the **date column** in **string** format and then we will use this **reference table** to create the **main table.**

Steps to create **reference table** named **agent_login_report_ref**



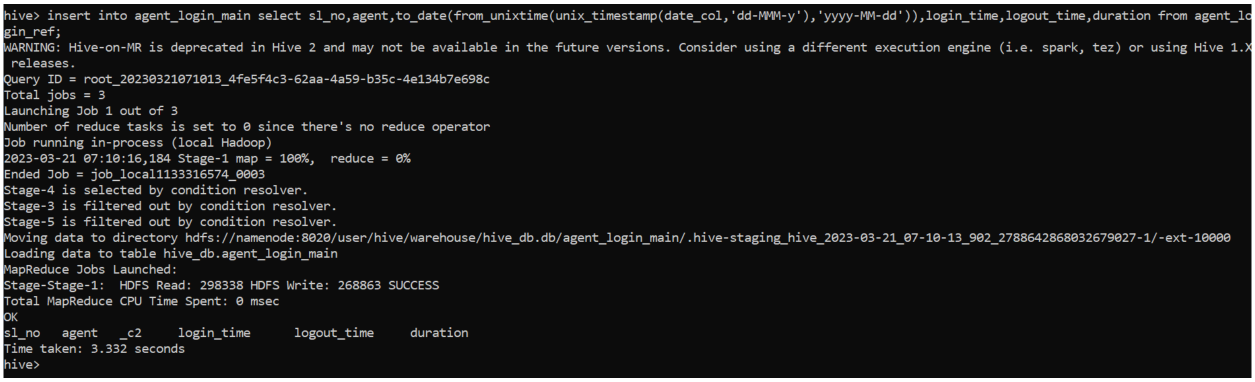
```
hive> create table agent_login_ref
    > (
    > sl_no int,
    > agent string,
    > date_col string,
    > login_time string,
    > logout_time string,
    > duration string)
    > row format delimited
    > fields terminated by ','
    > tblproperties("skip.header.line.count"="1");
OK
```

Now creating **main table** using this **ref table**



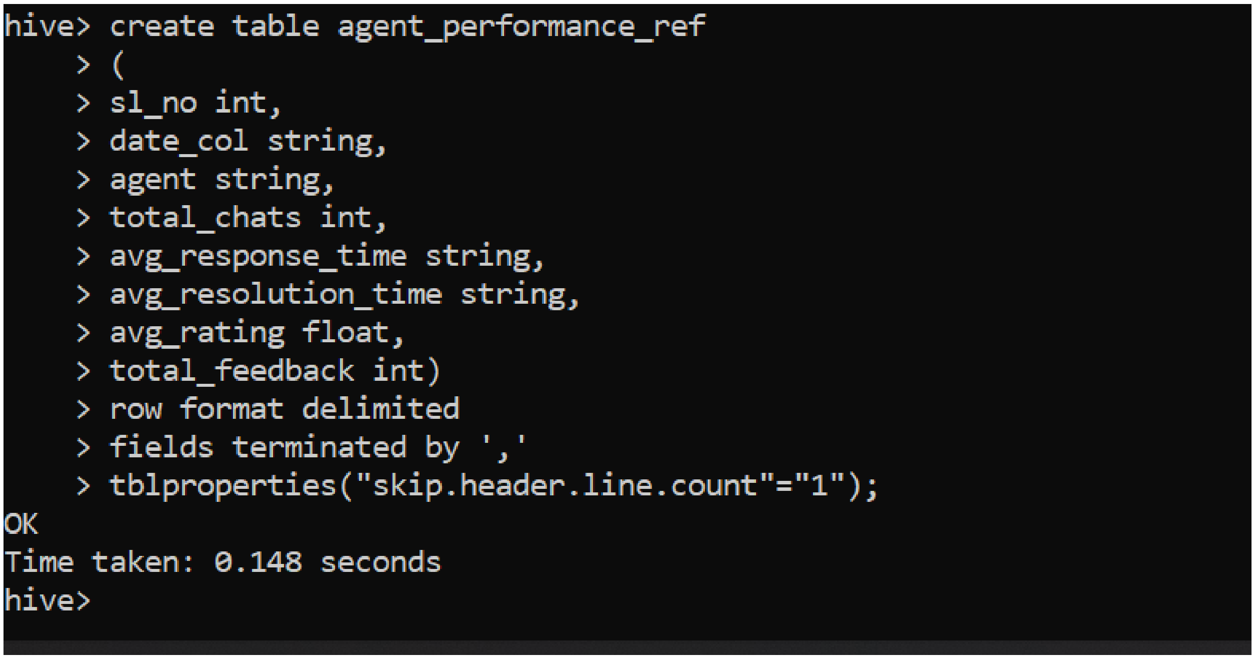
```
hive> create table agent_login_main
    > (
    > sl_no int,
    > agent string,
    > date_col date,
    > login_time string,
    > logout_time string,
    > duration string
    > )
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.207 seconds
hive>
```

Loading Data in this main table



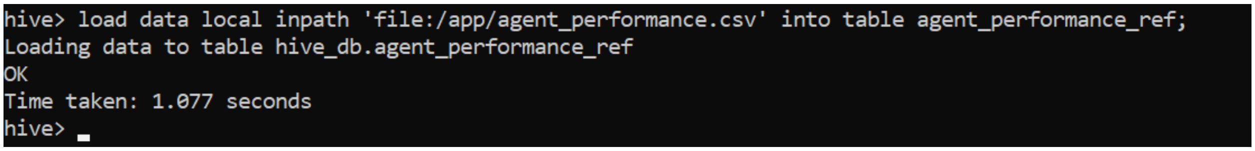
```
hive> insert into agent_login_main select sl_no,agent,to_date(from_unixtime(unix_timestamp(date_col,'dd-MMM-y'),'yyyy-MM-dd')),login_time,logout_time,duration from agent_lo
gin_ref;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X
 releases.
Query ID = root_20230321071013_4fe5f4c3-62aa-4a59-b35c-4e134b7e698c
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2023-03-21 07:10:16,184 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local1133316574_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 directory hdfs://namenode:8020/user/hive/warehouse/hive_db.db/agent_login_main/.hive-staging_hive_2023-03-21_07-10-13_902_2788642868032679027-1/-ext-10000
Loading data to table hive_db.agent_login_main
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 298338 HDFS Write: 268863 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
sl_no   agent   _c2     login_time      logout_time     duration
Time taken: 3.332 seconds
hive>
```

Steps to create **agent_performance** table



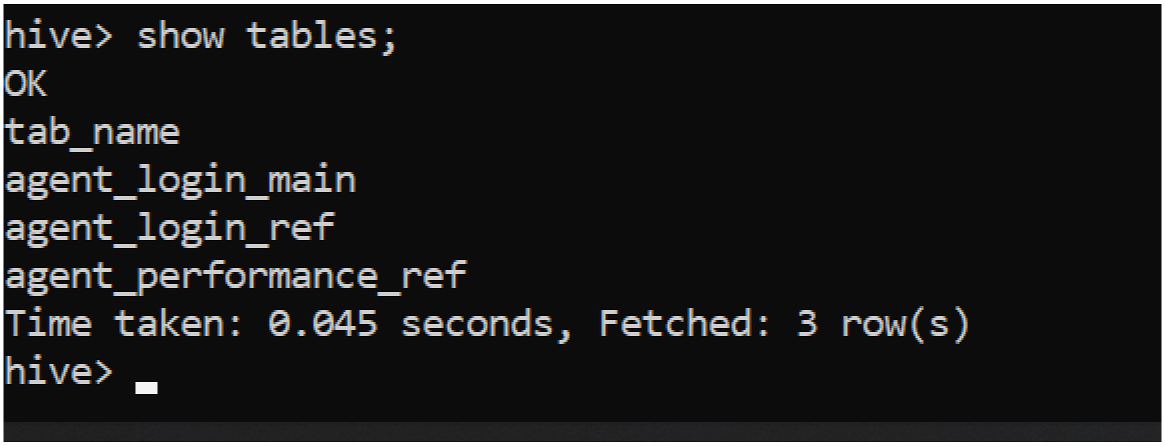
```
hive> create table agent_performance_ref
    > (
    > sl_no int,
    > date_col string,
    > agent string,
    > total_chats int,
    > avg_response_time string,
    > avg_resolution_time string,
    > avg_rating float,
    > total_feedback int)
    > row format delimited
    > fields terminated by ','
    > tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.148 seconds
hive>
```

Loading data from **local file system** in ref file



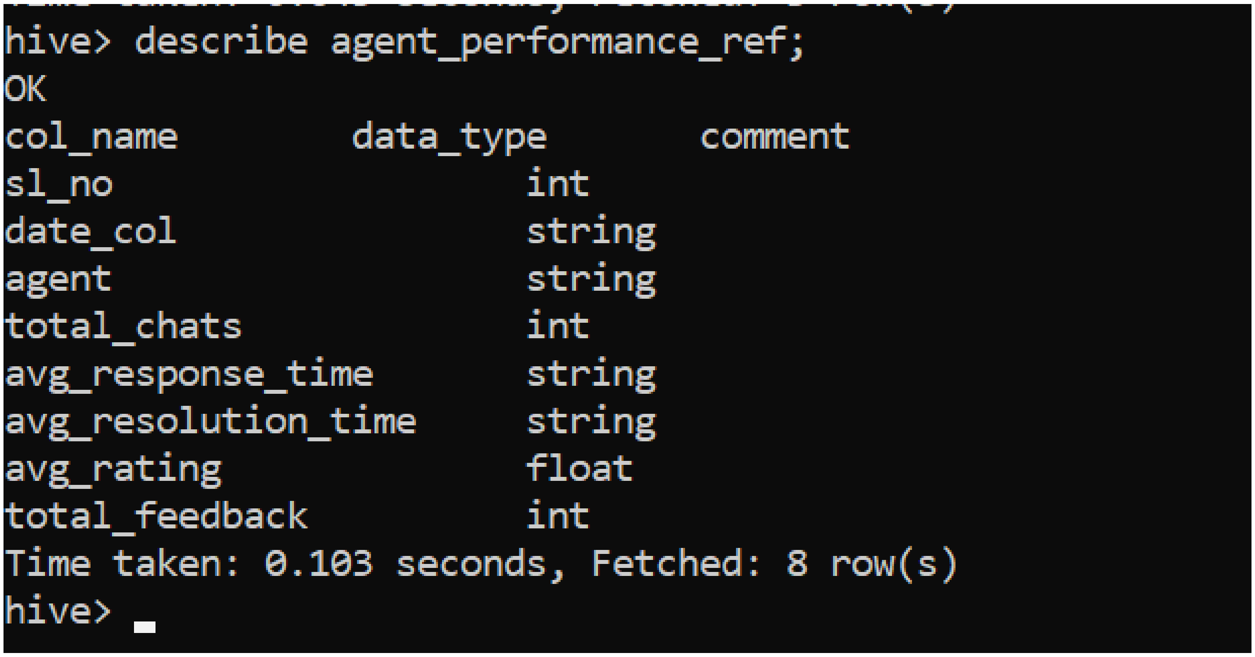
```
hive> load data local inpath 'file:/app/agent_performance.csv' into table agent_performance_ref;
Loading data to table hive_db.agent_performance_ref
OK
Time taken: 1.077 seconds
hive> _
```

Show table to **confirm** if table is created or not



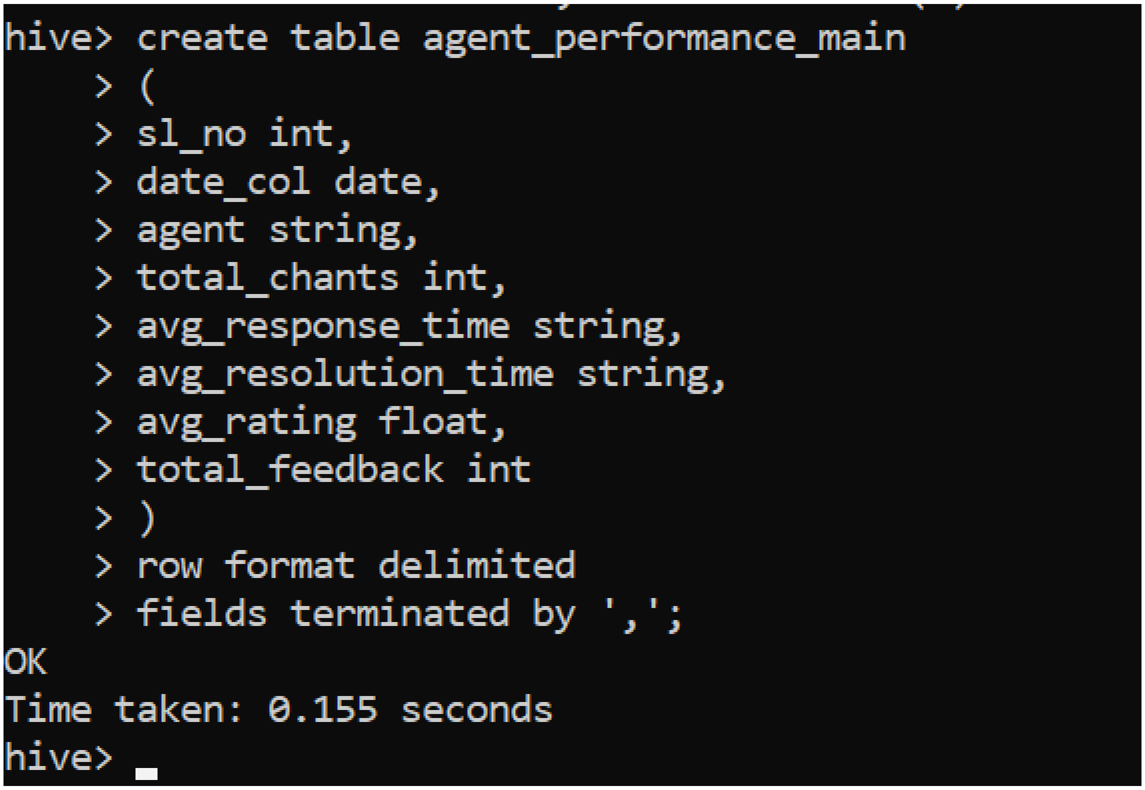
```
hive> show tables;
OK
tab_name
agent_login_main
agent_login_ref
agent_performance_ref
Time taken: 0.045 seconds, Fetched: 3 row(s)
hive> _
```

Use **describe** to check the **details** on columns



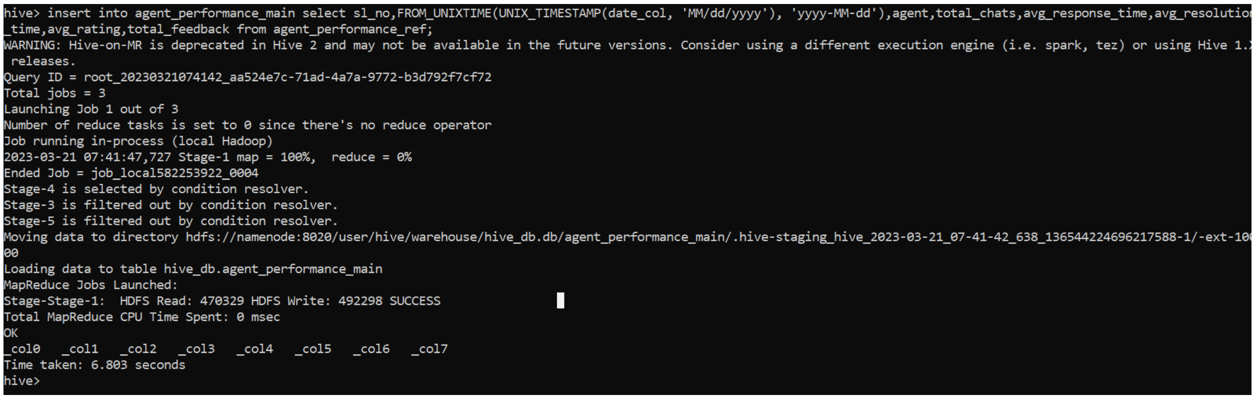
```
hive> describe agent_performance_ref;
OK
col_name        data_type       comment
sl_no                   int
date_col                string
agent                   string
total_chats             int
avg_response_time       string
avg_resolution_time     string
avg_rating              float
total_feedback          int
Time taken: 0.103 seconds, Fetched: 8 row(s)
hive> _
```

Creating **Main table** named **agent_performance_main**



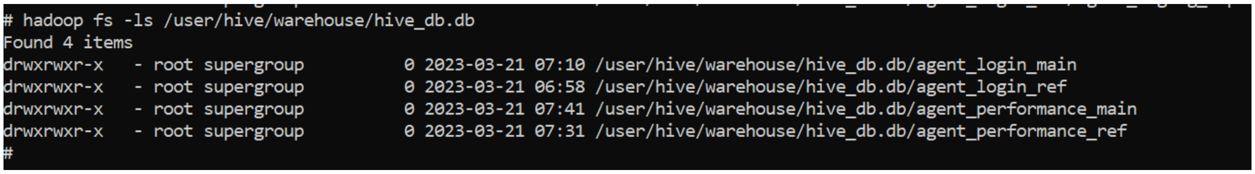
```
hive> create table agent_performance_main
    > (
    > sl_no int,
    > date_col date,
    > agent string,
    > total_chants int,
    > avg_response_time string,
    > avg_resolution_time string,
    > avg_rating float,
    > total_feedback int
    > )
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.155 seconds
hive> _
```

Inserting data in the **main table** using **ref table**

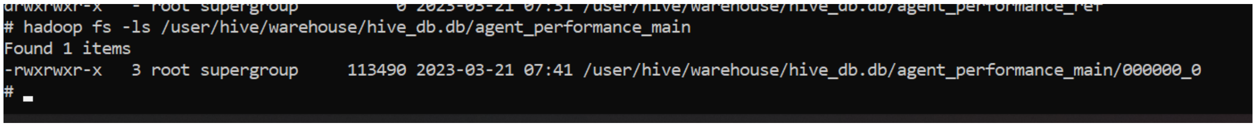


```
hive> insert into agent_performance_main select sl_no,FROM_UNIXTIME(UNIX_TIMESTAMP(date_col, 'MM/dd/yyyy'), 'yyyy-MM-dd'),agent,total_chats,avg_response_time,avg_resolutio
_time,avg_rating,total_feedback from agent_performance_ref;
WARNING: Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.
 releases.
Query ID = root_20230321074142_aa524e7c-71ad-4a7a-9772-b3d792f7cf72
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Job running in-process (local Hadoop)
2023-03-21 07:41:47,727 Stage-1 map = 100%,  reduce = 0%
Ended Job = job_local582253922_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 directory hdfs://namenode:8020/user/hive/warehouse/hive_db.db/agent_performance_main/.hive-staging_hive_2023-03-21_07-41-42_638_136544224696217588-1/-ext-10
00
Loading data to table hive_db.agent_performance_main
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 470329 HDFS Write: 492298 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
_col0   _col1   _col2   _col3   _col4   _col5   _col6   _col7
Time taken: 6.803 seconds
hive>
```

Let’s check in **HDFS file system** if **tables** are created or not



```
# hadoop fs -ls /user/hive/warehouse/hive_db.db
Found 4 items
drwxrwxr-x   - root supergroup          0 2023-03-21 07:10 /user/hive/warehouse/hive_db.db/agent_login_main
drwxrwxr-x   - root supergroup          0 2023-03-21 06:58 /user/hive/warehouse/hive_db.db/agent_login_ref
drwxrwxr-x   - root supergroup          0 2023-03-21 07:41 /user/hive/warehouse/hive_db.db/agent_performance_main
drwxrwxr-x   - root supergroup          0 2023-03-21 07:31 /user/hive/warehouse/hive_db.db/agent_performance_ref
#
```



```
drwxrwxr-x   - root supergroup          0 2023-03-21 07:31 /user/hive/warehouse/hive_db.db/agent_performance_ref
# hadoop fs -ls /user/hive/warehouse/hive_db.db/agent_performance_main
Found 1 items
-rwxrwxr-x   3 root supergroup     113490 2023-03-21 07:41 /user/hive/warehouse/hive_db.db/agent_performance_main/000000_0
# 
```

3. List of all agents' names.

```
hive> select distinct trim(agent) from agent_performance_main
union select distinct trim(agent) from agent_login_main
    > ;
```

4. Find out agent average rating.

```
hive> select agent,avg(avg_rating) as avg_rating
    > from agent_performance_main
    > group by agent
    > ;
```

5. Total working days for each agents

```
hive> select agent,count(*) as login_count from agent_login_main
    > group by agent
    > order by login_count desc;
```

6. Total query that each agent have taken

```
hive> select agent,sum(total_chants) as total_queries
    > from agent_performance_main
    > group by agent
    > order by total_queries;
```

7. Total Feedback that each agent have received

```
hive> select agent,sum(total_feedback) as total_feedback
    > from agent_performance_main
    > group by agent
    > order by total_feedback desc;
```

8.Agent name who have average rating between 3.5 to 4

```
select agent,avg(avg_rating) as avg_rating from
agent_performance_main group by agent having avg_rating between
3.5 and 4;
```

9. Agent name who have rating less than 3.5

```
hive> select agent,avg(avg_rating) as avg_rating from
agent_performance_main group by agent having avg_rating <3.5;
```

10. Agent name who have rating more than 4.5

```
hive> select agent,avg(avg_rating) as avg_rating from
agent_performance_main group by agent having avg_rating >4.5;
```

11. How many feedback agents have received more than 4.5 average

```
hive> select agent,count(case when total_feedback>4.5 then 1
end) as feedback_count from agent_performance_main group by
agent;
```

12. average weekly response time for each agent

```
hive> select agent,weekofyear(date_col) as
week_no,from_unixtime(cast(avg(unix_timestamp(avg_response_time,
'H:mm:ss'))as bigint),'H:mm:ss') as avg_response_week_time from
    > agent_performance_main
    > group by agent,weekofyear(date_col);

```

13. average weekly resolution time for each agents

```
hive> select agent,weekofyear(date_col) as
week_no,from_unixtime(cast(avg(unix_timestamp(avg_resolution_tim
e,'H:mm:ss'))as bigint),'H:mm:ss') as avg_resolution_week_time
from
> agent_performance_main
> group by agent,weekofyear(date_col);
```

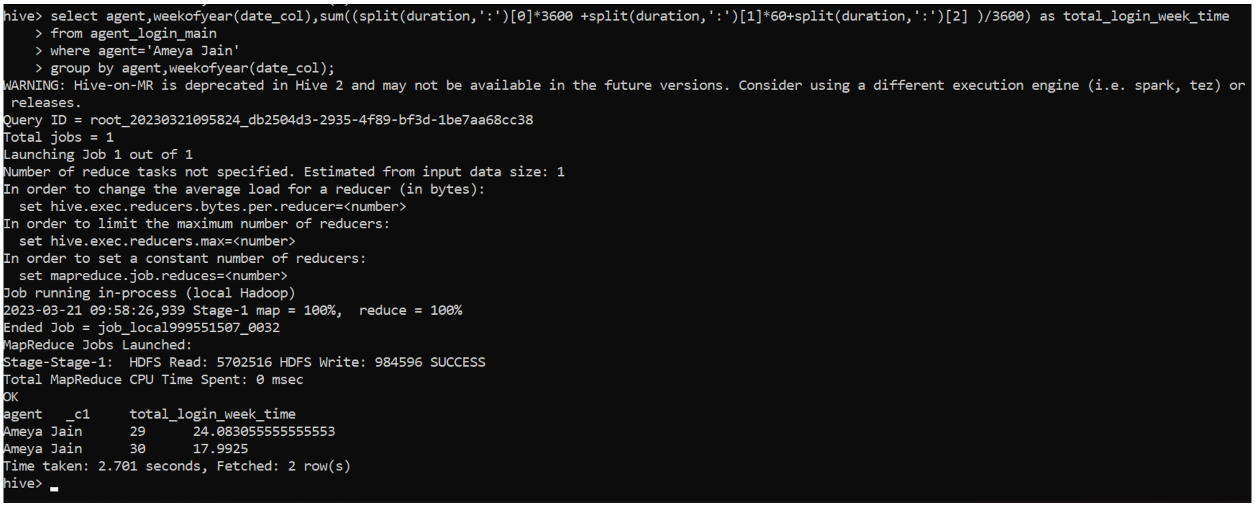
14. Find the number of chat on which they have received a feedback

```
hive> select agent,count(total_chants) as total_chants
    > from agent_performance_main
    > where (total_feedback<>0 and total_feedback is not null)
    > group by agent;
```

Alternatively

```
agent,
date_format(`date`,'W') week_no,
sum((split(avg_resolution_time,':')[0]*3600
+split(avg_resolution_time,':')[1]*60+split(avg_resolution_time,
':')[2] )/3600) total_weekly_contri_hrs,
avg((split(avg_response_time ,':')[0]*3600
+split(avg_response_time ,':')[1]*60+split(avg_response_time
,':')[2] )/3600) Avg_weekly_response_time_hrs
from agent_performance_main
group by
agent,date_format(`date`,'W')
```

15. Total contribution hour for each and every agents weekly basis



```
hive> select agent,weekofyear(date_col),sum((split(duration,':')[0]*3600 +split(duration,':')[1]*60+split(duration,':')[2] )/3600) as total_login_week_time
    > from agent_login_main
    > where agent='Ameya Jain'
    > group by agent,weekofyear(date_col);
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
 releases.
Query ID = root_20230321095824_db2504d3-2935-4f89-bf3d-1be7aa68cc38
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
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>
Job running in-process (local Hadoop)
2023-03-21 09:58:26,939 Stage-1 map = 100%,  reduce = 100%
Ended Job = job_local999551507_0032
MapReduce Jobs Launched:
Stage-Stage-1:  HDFS Read: 5702516 HDFS Write: 984596 SUCCESS
Total MapReduce CPU Time Spent: 0 msec
OK
agent   _c1     total_login_week_time
Ameya Jain      29      24.083055555555553
Ameya Jain      30      17.9925
Time taken: 2.701 seconds, Fetched: 2 row(s)
hive> _
```

16. Perform inner join, left join and right join based on the agent column and after joining the table export that data into your local system.

Ways to Export File in local

```
INSERT OVERWRITE LOCAL DIRECTORY '/test/' ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',' SELECT * FROM agent_login_main limit 2;
```

#Exports to HDFS directory

```
INSERT OVERWRITE DIRECTORY '/user/data/output/export' ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',' SELECT * FROM emp.employee;
```

16. Perform inner join, left join and right join based on the agent column and after joining the table export that data into your local system.

Performing Inner Join

```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/inner_join/' ROW FORMAT
DELIMITED FIELDS TERMINATED BY ',' SELECT * FROM
agent_performance_main join agent_login_main on
agent_login_main.agent=agent_performance_main.agent
```

Performing Left Join

```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/join/left_join/' ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT * FROM
agent_performance_main left join agent_login_main on
agent_login_main.agent=agent_performance_main.agent
```

```
    > ;
```

Performing Right Join

```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/join/right_join/' ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' SELECT * FROM
agent_performance_main right join agent_login_main on
agent_login_main.agent=agent_performance_main.agent
    > ;
```

Checking our local file system if files were migrated or not

Go to the local file system

Navigate to the root directory

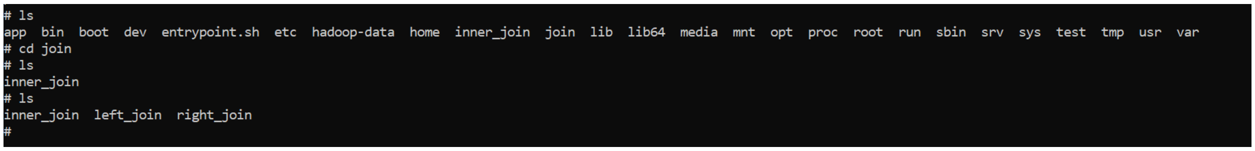
Cd /

You will get join folder

Cd to join folder

Cd join

You will get list of the folders inside the join directory



```
# ls
app  bin  boot  dev  entrypoint.sh  etc  hadoop-data  home  inner_join  join  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  test  tmp  usr  var
# cd join
# ls
inner_join
# ls
inner_join  left_join  right_join
#
```

17. Perform partitioning on top of the agent column and then on top of that perform bucketing for each partitioning.

Approach

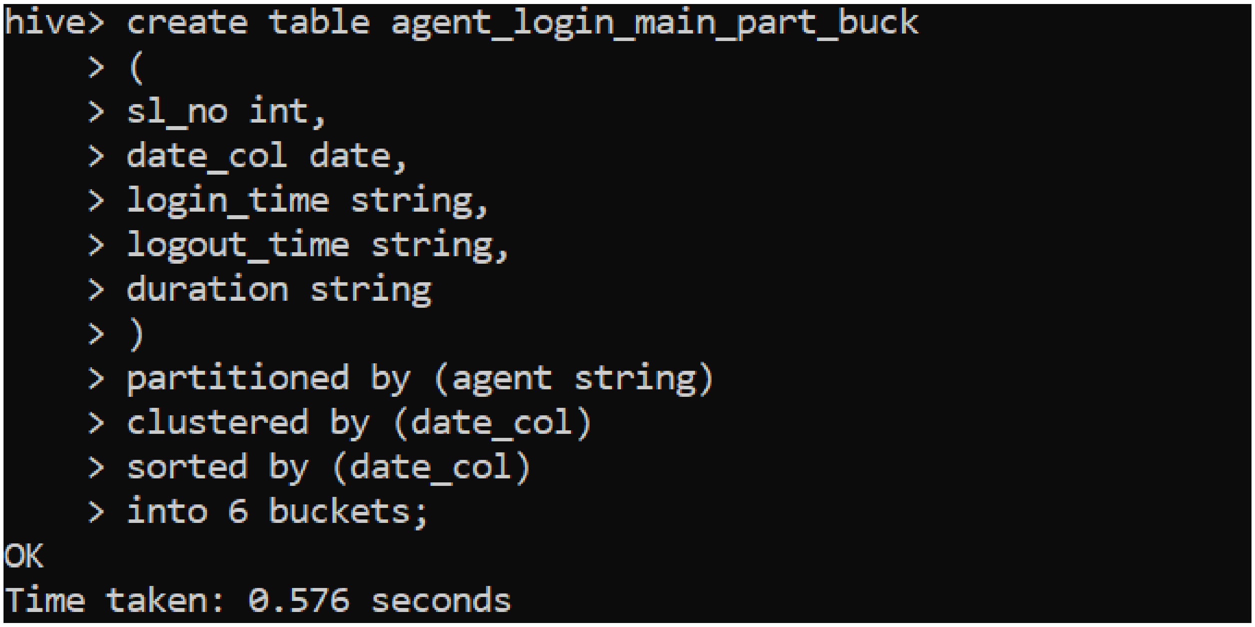
So here we will do these two thing

We will partition table on agent column

Table would be agent_login_main table
And on top of that we will create buckets on date_col column

Let's see

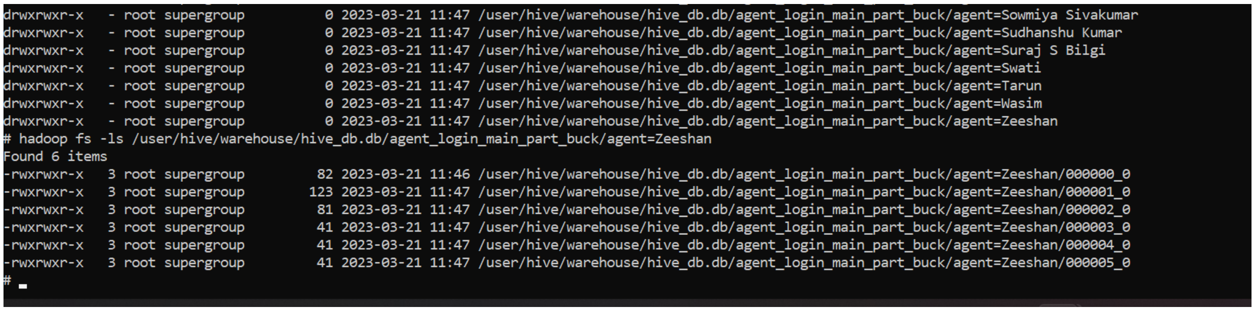
Let's create a partition table and bucketed table



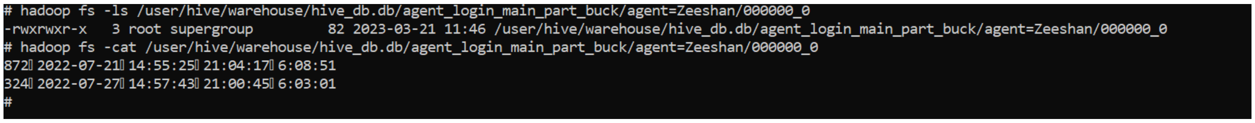
```
hive> create table agent_login_main_part_buck
    > (
    > sl_no int,
    > date_col date,
    > login_time string,
    > logout_time string,
    > duration string
    > )
    > partitioned by (agent string)
    > clustered by (date_col)
    > sorted by (date_col)
    > into 6 buckets;
OK
Time taken: 0.576 seconds
```

Loading data in partition and bucketing table

```
hive> set hive.exec.dynamic.partition.mode=nonstrict;
hive> insert overwrite table agent_login_main_part_buck
partition (agent)
    > select
sl_no,date_col,login_time,logout_time,duration,agent from
agent_login_main;
```



```
drwxrwxr-x   - root supergroup          0 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Sowmiya Sivakumar
drwxrwxr-x   - root supergroup          0 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Sudhanshu Kumar
drwxrwxr-x   - root supergroup          0 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Suraj S Bilgi
drwxrwxr-x   - root supergroup          0 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Swati
drwxrwxr-x   - root supergroup          0 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Tarun
drwxrwxr-x   - root supergroup          0 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Wasim
drwxrwxr-x   - root supergroup          0 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan
# hadoop fs -ls /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan
Found 6 items
-rwxrwxr-x   3 root supergroup         82 2023-03-21 11:46 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000000_0
-rwxrwxr-x   3 root supergroup        123 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000001_0
-rwxrwxr-x   3 root supergroup         81 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000002_0
-rwxrwxr-x   3 root supergroup         41 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000003_0
-rwxrwxr-x   3 root supergroup         41 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000004_0
-rwxrwxr-x   3 root supergroup         41 2023-03-21 11:47 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000005_0
# 
```



```
# hadoop fs -ls /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000000_0
-rwxrwxr-x   3 root supergroup         82 2023-03-21 11:46 /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000000_0
# hadoop fs -cat /user/hive/warehouse/hive_db.db/agent_login_main_part_buck/agent=Zeeshan/000000_0
872▯2022-07-21▯14:55:25▯21:04:17▯6:08:51
324▯2022-07-27▯14:57:43▯21:00:45▯6:03:01
#
```