Course Project – ETL & Batch Processing

I : DATA INGESTION:

Firstly the Dataset which was hosted on the Amazon RDS was imported to the MySQL Workbench by connecting with the RDS through Cloud.

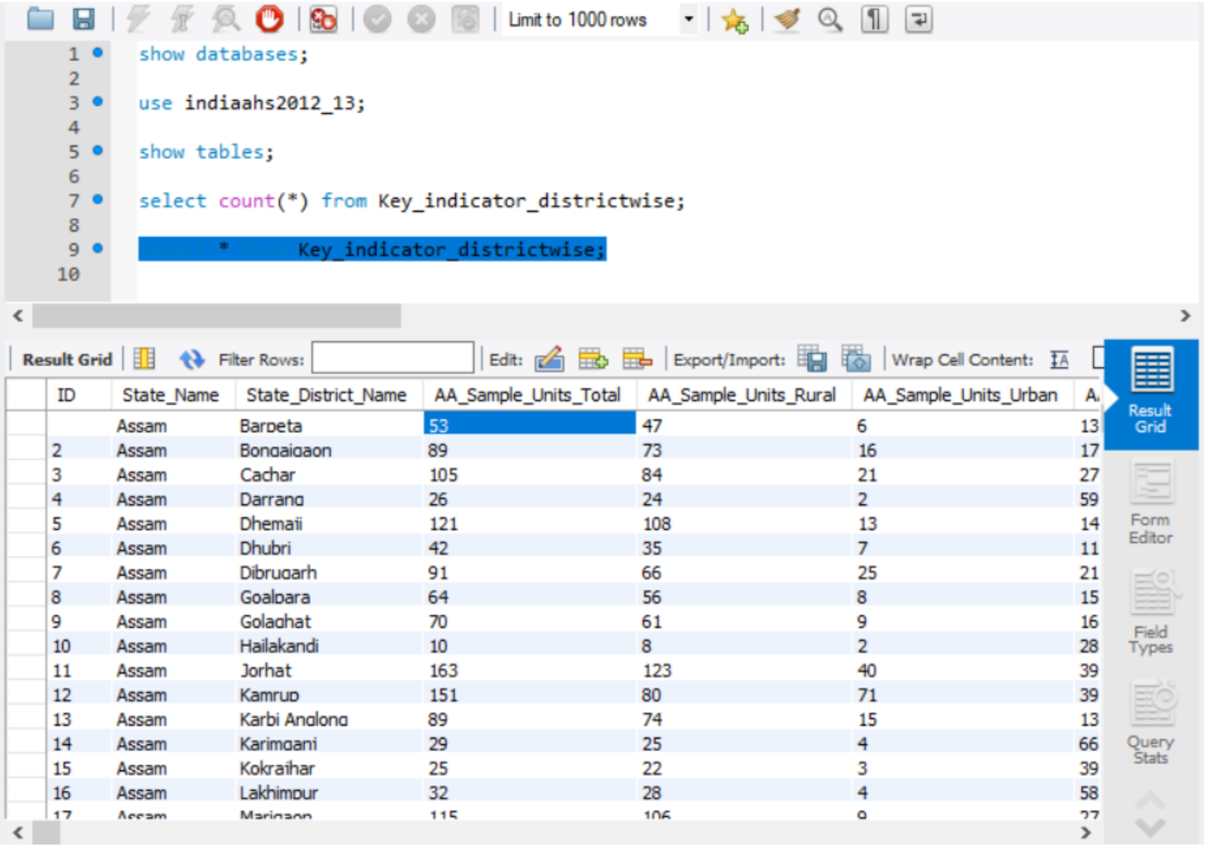
The Database which is required for Ingestion from Relational database (MySQL to HDFS) is indiaahs2012_13.

Database has the following table - Key_indicator_districtwise

SQOOP Import:

The Data present in the Relational database has to be transferred to the HDFS with the help of SQOOP. SQOOP transfers data Bi-Directionally. It is effective in transferring Bulk data between Relational Database to the Hadoop.

Data in MySQL-



```
show databases;

use indiaahs2012_13;

show tables;

select count(*) from Key_indicator_districtwise;

select * from Key_indicator_districtwise;

```

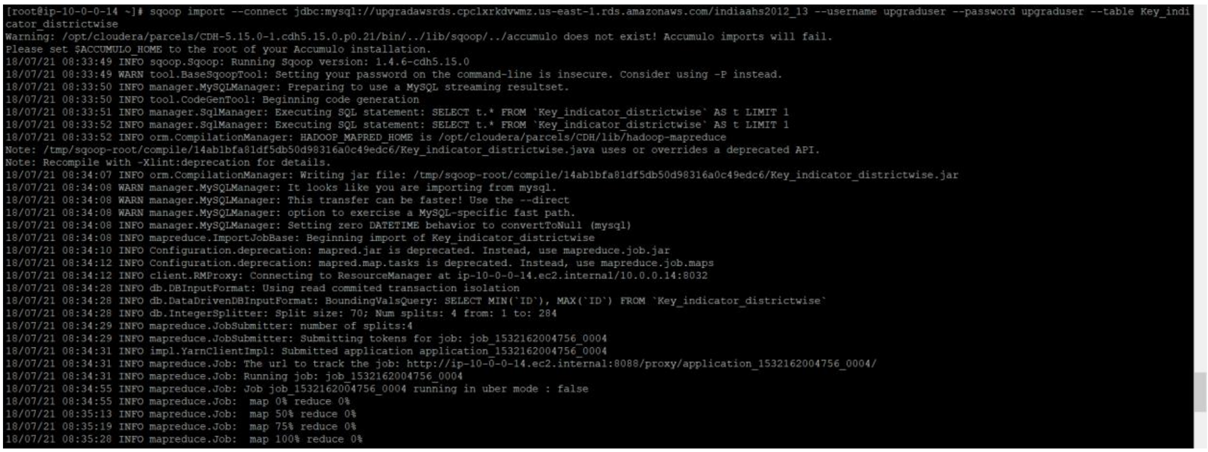
ID	State_Name	State_District_Name	AA_Sample_Units_Total	AA_Sample_Units_Rural	AA_Sample_Units_Urban	A
	Assam	Barpeta	53	47	6	13
2	Assam	Bongaigaon	89	73	16	17
3	Assam	Cachar	105	84	21	27
4	Assam	Darrang	26	24	2	59
5	Assam	Dhemaji	121	108	13	14
6	Assam	Dhubri	42	35	7	11
7	Assam	Dibrugarh	91	66	25	21
8	Assam	Goalpara	64	56	8	15
9	Assam	Golaghat	70	61	9	16
10	Assam	Hailakandi	10	8	2	28
11	Assam	Jorhat	163	123	40	39
12	Assam	Kamrup	151	80	71	39
13	Assam	Karbi Anglong	89	74	15	13
14	Assam	Karimganj	29	25	4	66
15	Assam	Kokrajhar	25	22	3	39
16	Assam	Lakhimpur	32	28	4	58
17	Assam	Marigaon	115	106	9	27

This data which is present in Amazon RDS has to be Ingested/Imported to HDFS by the SQOOP command shown below,

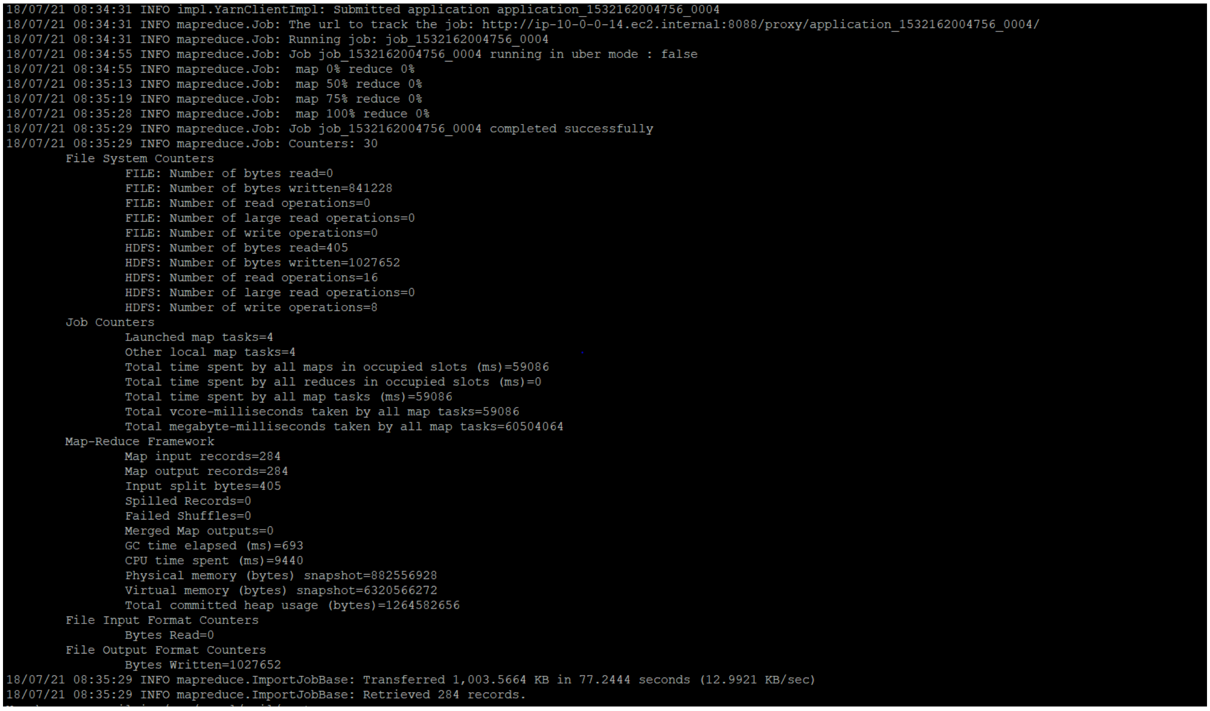
<u>**SQOOP Import Command**</u> :

sqoop import --connect jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com/indiaahs2012_13 --username upgraduser --password upgraduser --table Key_indicator_districtwise

After firing this command we can see the Ingestion has been completed and the data from the 'Key_indicator_districtwise' table from the database in MySQL is transferred to the HDFS.



```
[root@ip-10-0-0-14 ~]# sqoop import --connect jdbc:mysql://upgradawsrds.cpclxrkdvwmz.us-east-1.rds.amazonaws.com/indiaahs2012_13 --username upgraduser --password upgraduser --table Key_indi
cator_districtwise
Warning: /opt/cloudera/parcels/CDH-5.15.0-1.cdh5.15.0.p0.21/bin/../lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
18/07/21 08:33:49 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.15.0
18/07/21 08:33:49 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
18/07/21 08:33:50 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
18/07/21 08:33:50 INFO tool.CodeGenTool: Beginning code generation
18/07/21 08:33:51 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `Key_indicator_districtwise` AS t LIMIT 1
18/07/21 08:33:52 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `Key_indicator_districtwise` AS t LIMIT 1
18/07/21 08:33:52 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /opt/cloudera/parcels/CDH/lib/hadoop-mapreduce
Note: /tmp/sqoop-root/compile/14ab1bfa81df5db50d98316a0c49edc6/Key_indicator_districtwise.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
18/07/21 08:34:07 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-root/compile/14ab1bfa81df5db50d98316a0c49edc6/Key_indicator_districtwise.jar
18/07/21 08:34:08 WARN manager.MySQLManager: It looks like you are importing from mysql.
18/07/21 08:34:08 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
18/07/21 08:34:08 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
18/07/21 08:34:08 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
18/07/21 08:34:08 INFO mapreduce.ImportJobBase: Beginning import of Key_indicator_districtwise
18/07/21 08:34:10 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
18/07/21 08:34:12 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
18/07/21 08:34:12 INFO client.RMProxy: Connecting to ResourceManager at ip-10-0-0-14.ec2.internal/10.0.0.14:8032
18/07/21 08:34:28 INFO db.DBInputFormat: Using read commited transaction isolation
18/07/21 08:34:28 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`ID`), MAX(`ID`) FROM `Key_indicator_districtwise`
18/07/21 08:34:28 INFO db.IntegerSplitter: Split size: 70; Num splits: 4 from: 1 to: 284
18/07/21 08:34:29 INFO mapreduce.JobSubmitter: number of splits:4
18/07/21 08:34:29 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1532162004756_0004
18/07/21 08:34:31 INFO impl.YarnClientImpl: Submitted application application_1532162004756_0004
18/07/21 08:34:31 INFO mapreduce.Job: The url to track the job: http://ip-10-0-0-14.ec2.internal:8088/proxy/application_1532162004756_0004/
18/07/21 08:34:31 INFO mapreduce.Job: Running job: job_1532162004756_0004
18/07/21 08:34:55 INFO mapreduce.Job: Job job_1532162004756_0004 running in uber mode : false
18/07/21 08:34:55 INFO mapreduce.Job:  map 0% reduce 0%
18/07/21 08:35:13 INFO mapreduce.Job:  map 50% reduce 0%
18/07/21 08:35:19 INFO mapreduce.Job:  map 75% reduce 0%
18/07/21 08:35:28 INFO mapreduce.Job:  map 100% reduce 0%
```

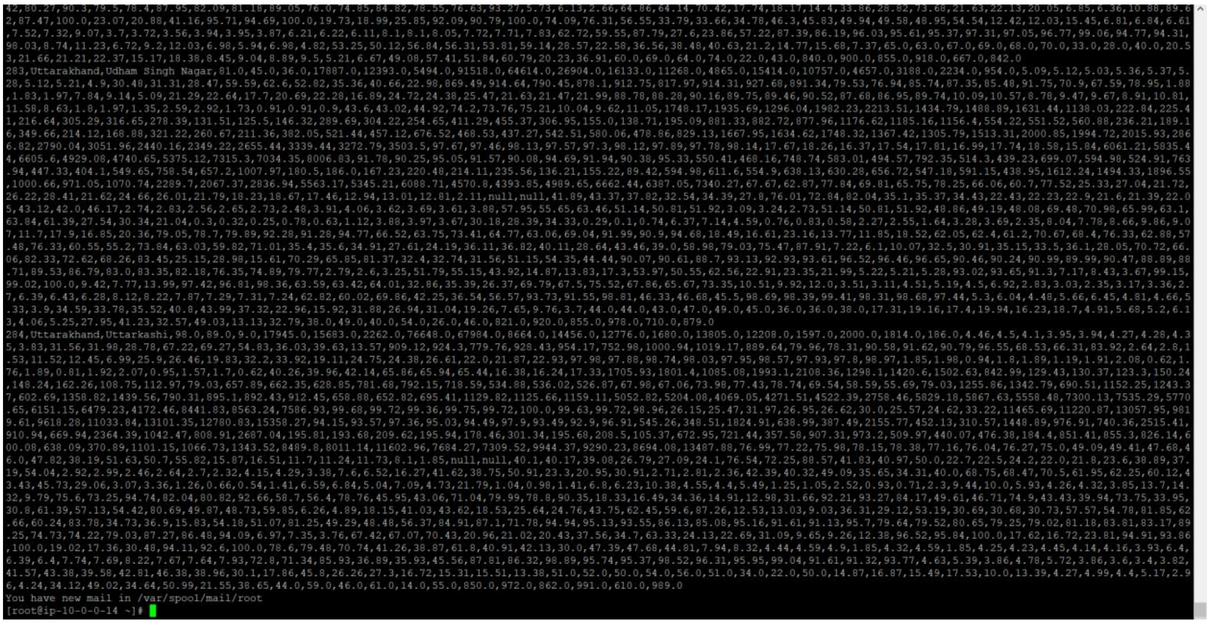


```
18/07/21 08:34:31 INFO impl.YarnClientImpl: Submitted application application_1532162004756_0004
18/07/21 08:34:31 INFO mapreduce.Job: The url to track the job: http://ip-10-0-0-14.ec2.internal:8088/proxy/application_1532162004756_0004/
18/07/21 08:34:31 INFO mapreduce.Job: Running job: job_1532162004756_0004
18/07/21 08:34:55 INFO mapreduce.Job: Job job_1532162004756_0004 running in uber mode : false
18/07/21 08:34:55 INFO mapreduce.Job:  map 0% reduce 0%
18/07/21 08:35:13 INFO mapreduce.Job:  map 50% reduce 0%
18/07/21 08:35:19 INFO mapreduce.Job:  map 75% reduce 0%
18/07/21 08:35:28 INFO mapreduce.Job:  map 100% reduce 0%
18/07/21 08:35:29 INFO mapreduce.Job: Job job_1532162004756_0004 completed successfully
18/07/21 08:35:29 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=841228
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=405
                HDFS: Number of bytes written=1027652
                HDFS: Number of read operations=16
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=8
        Job Counters
                Launched map tasks=4
                Other local map tasks=4
                Total time spent by all maps in occupied slots (ms)=59086
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=59086
                Total vcore-milliseconds taken by all map tasks=59086
                Total megabyte-milliseconds taken by all map tasks=60504064
        Map-Reduce Framework
                Map input records=284
                Map output records=284
                Input split bytes=405
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=693
                CPU time spent (ms)=9440
                Physical memory (bytes) snapshot=882556928
                Virtual memory (bytes) snapshot=6320566272
                Total committed heap usage (bytes)=1264582656
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=1027652
18/07/21 08:35:29 INFO mapreduce.ImportJobBase: Transferred 1,003.5664 KB in 77.2444 seconds (12.9921 KB/sec)
18/07/21 08:35:29 INFO mapreduce.ImportJobBase: Retrieved 284 records.
```

Command to View the Ingested Records:

hadoop fs -cat Key_indicator_districtwise/part-m-*

The 284 Records has been Imported and the Data in HDFS is as shown below,



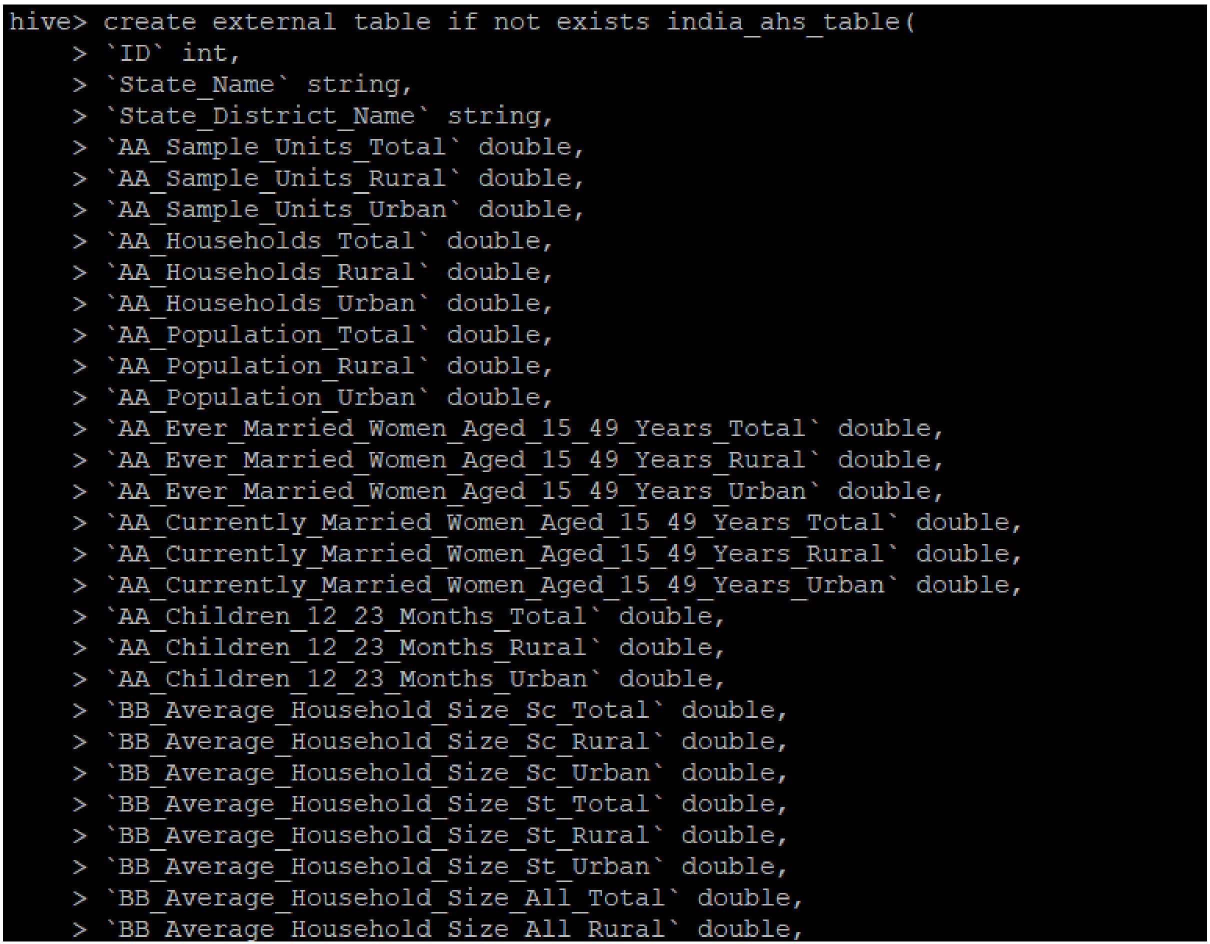
```
42,80.27,90.3,79.5,78.4,87.95,82.09,81.18,89.05,76.0,74.85,84.82,78.55,76.63,93.27,5.73,6.13,2.66,64.86,64.14,70.42,17.74,18.17,14.4,33.86,28.82,73.68,21.63,22.13,20.05,6.85,6.36,10.88,89.6
2,87.47,100.0,23.07,20.88,41.16,95.71,94.69,100.0,19.73,18.99,25.85,92.09,90.79,100.0,74.09,76.31,56.55,33.79,33.66,34.78,46.3,45.83,49.94,49.58,48.95,54.54,12.42,12.03,15.45,6.81,6.84,6.61
,7.52,7.32,9.07,3.7,3.72,3.56,3.94,3.95,3.87,6.21,6.22,6.11,8.1,8.1,8.05,7.72,7.71,7.83,62.72,59.55,87.79,27.6,23.86,57.22,87.39,86.19,96.03,95.61,95.37,97.31,97.05,96.77,99.06,94.77,94.31,
98.03,8.74,11.23,6.72,9.2,12.03,6.98,5.94,6.98,4.82,53.25,50.12,56.84,56.31,53.81,59.14,28.57,22.58,36.56,38.48,40.63,21.2,14.77,15.68,7.37,65.0,63.0,67.0,69.0,68.0,70.0,33.0,28.0,40.0,20.5
3,21.66,21.21,22.37,15.17,18.38,8.45,9.04,8.89,9.5,5.21,6.67,49.08,57.41,51.84,60.79,20.23,36.91,60.0,69.0,64.0,74.0,22.0,43.0,840.0,900.0,855.0,918.0,667.0,842.0
283,Uttarakhand,Udham Singh Nagar,81.0,45.0,36.0,17887.0,12393.0,5494.0,91518.0,64614.0,26904.0,16133.0,11268.0,4865.0,15414.0,10757.0,4657.0,3188.0,2234.0,954.0,5.09,5.12,5.03,5.36,5.37,5.
28,5.12,5.21,4.9,30.48,31.31,28.47,59.59,62.6,52.82,35.36,40.66,22.98,869.49,914.64,790.45,878.1,912.75,817.97,914.31,927.68,891.34,79.53,76.94,85.74,87.35,85.48,91.75,70.9,67.59,78.95,1.88
,1.83,1.97,7.84,9.14,5.09,21.29,22.64,17.7,20.69,22.28,16.89,24.72,24.38,25.47,21.63,21.47,21.99,88.78,88.28,90.16,89.75,89.46,90.52,87.68,86.95,89.74,10.09,10.57,8.78,9.47,9.67,8.91,10.81,
11.58,8.63,1.8,1.97,1.35,2.59,2.92,1.73,0.91,0.91,0.9,43.6,43.02,44.92,74.2,73.76,75.21,10.04,9.62,11.05,1748.17,1935.69,1296.04,1982.23,2213.51,1434.79,1488.89,1631.44,1138.03,222.84,225.4
1,216.64,305.29,316.65,278.39,131.51,125.5,146.32,289.69,304.22,254.65,411.29,455.37,306.95,155.0,138.71,195.09,881.33,882.72,877.96,1176.62,1185.16,1156.4,554.22,551.52,560.88,236.21,189.1
6,349.66,214.12,168.88,321.22,260.67,211.36,382.05,521.44,457.12,676.52,468.53,437.27,542.51,580.06,478.86,829.13,1667.95,1634.62,1748.32,1367.42,1305.79,1513.31,2000.85,1994.72,2015.93,286
6.82,2790.04,3051.96,2440.16,2349.22,2655.44,3339.44,3272.79,3503.5,97.67,97.46,98.13,97.57,97.3,98.12,97.89,97.78,98.14,17.67,18.26,16.37,17.54,17.81,16.99,17.74,18.58,15.84,6061.21,5835.4
4,6605.6,4929.08,4740.65,5375.12,7315.3,7034.35,8006.83,91.78,90.25,95.05,91.57,90.08,94.69,91.94,90.38,95.33,550.41,468.16,748.74,583.01,494.57,792.35,514.3,439.23,699.07,594.98,524.91,763
.94,447.33,404.1,549.65,758.54,657.2,1007.97,180.5,186.0,167.23,220.48,214.11,235.56,136.21,155.22,89.42,594.98,611.6,554.9,638.13,630.28,656.72,547.18,591.15,438.95,1612.24,1494.33,1896.55
,1000.66,971.05,1070.74,2289.7,2067.37,2836.94,5563.17,5345.21,6088.71,4570.8,4393.85,4989.65,6662.44,6387.05,7340.27,67.67,62.87,77.84,69.81,65.75,78.25,66.06,60.7,77.52,25.33,27.04,21.72,
26.22,28.41,21.62,24.66,26.01,21.79,18.23,18.67,17.46,12.94,13.01,12.81,2.11,null,null,41.89,43.37,37.82,32.54,34.39,27.8,76.01,72.84,82.04,35.1,35.37,34.43,22.43,22.23,22.9,21.6,21.39,22.0
5,43.12,42.0,46.17,2.74,2.83,2.56,2.65,2.73,2.48,3.91,4.06,3.62,3.69,3.61,3.88,57.95,55.65,63.46,51.14,50.81,51.92,3.09,3.24,2.73,51.14,50.81,51.92,48.86,49.19,48.08,69.48,70.98,65.99,63.1,
63.84,61.39,27.54,30.34,21.04,0.3,0.32,0.25,0.78,0.63,1.12,3.88,3.97,3.67,30.18,28.39,34.33,0.29,0.1,0.74,6.37,7.14,4.59,0.76,0.83,0.58,2.27,2.55,1.64,3.28,3.69,2.35,8.04,7.78,8.66,9.86,9.0
7,11.7,17.9,16.85,20.36,79.05,78.7,79.89,92.28,91.28,94.77,66.52,63.75,73.41,64.77,63.06,69.04,91.99,90.9,94.68,18.49,16.61,23.16,13.77,11.85,18.52,62.05,62.4,61.2,70.67,68.4,76.33,62.88,57
.48,76.33,60.55,55.2,73.84,63.03,59.82,71.01,35.4,35.6,34.91,27.61,24.19,36.11,36.82,40.11,28.64,43.46,39.0,58.98,79.03,75.47,87.91,7.22,6.1,10.07,32.5,30.91,35.15,33.5,36.1,28.05,70.72,66.
06,82.33,72.62,68.26,83.45,25.15,28.98,15.61,70.29,65.85,81.37,32.4,32.74,31.56,51.15,54.35,44.44,90.07,90.61,88.7,93.13,92.93,93.61,96.52,96.46,96.65,90.46,90.24,90.99,89.99,90.47,88.89,88
.71,89.53,86.79,83.0,83.35,82.18,76.35,74.89,79.77,2.79,2.6,3.25,51.79,55.15,43.92,14.87,13.83,17.3,53.97,50.55,62.56,22.91,23.35,21.99,5.22,5.21,5.28,93.02,93.65,91.3,7.17,8.43,3.67,99.15,
99.02,100.0,9.42,7.77,13.99,97.42,96.81,98.36,63.59,63.42,64.01,32.86,35.39,26.37,69.79,67.5,75.52,67.86,65.67,73.35,10.51,9.92,12.0,3.51,3.11,4.51,5.19,4.5,6.92,2.83,3.03,2.35,3.17,3.36,2.
7,6.39,6.43,6.28,8.12,8.22,7.87,7.29,7.31,7.24,62.82,60.02,69.86,42.25,36.54,56.57,93.73,91.55,98.81,46.33,46.68,45.5,98.69,98.39,99.41,98.31,98.68,97.44,5.3,6.04,4.48,5.66,6.45,4.81,4.66,5
.33,3.9,34.59,33.78,35.52,40.8,43.99,37.32,22.96,15.92,31.88,26.94,31.04,19.26,7.65,9.76,3.7,44.0,44.0,43.0,47.0,49.0,45.0,36.0,36.0,38.0,17.31,19.16,17.4,19.94,16.23,18.7,4.91,5.68,5.2,6.1
3,4.06,5.25,27.95,41.23,32.57,49.03,13.13,32.79,38.0,49.0,40.0,54.0,26.0,46.0,821.0,920.0,855.0,978.0,710.0,879.0
284,Uttarakhand,Uttarkashi,98.0,89.0,9.0,17945.0,15683.0,2262.0,76648.0,67984.0,8664.0,14456.0,12776.0,1680.0,13805.0,12208.0,1597.0,2000.0,1814.0,186.0,4.46,4.5,4.1,3.95,3.94,4.27,4.28,4.3
5,3.83,31.56,31.98,28.78,67.22,69.27,54.83,36.03,39.63,13.57,909.12,924.3,779.76,928.43,954.17,752.98,1000.94,1019.17,889.64,79.96,78.31,90.58,91.62,90.79,96.55,68.53,66.31,83.92,2.64,2.8,1
.53,11.52,12.45,6.99,25.9,26.46,19.83,32.2,33.92,19.11,24.75,24.38,26.61,22.0,21.87,22.93,97.98,97.88,98.74,98.03,97.95,98.57,97.93,97.8,98.97,1.85,1.98,0.94,1.8,1.89,1.19,1.91,2.08,0.62,1.
76,1.89,0.81,1.92,2.07,0.95,1.57,1.7,0.62,40.26,39.96,42.14,65.86,65.94,65.44,16.38,16.24,17.33,1705.93,1801.4,1085.08,1993.1,2108.36,1298.1,1420.6,1502.63,842.99,129.43,130.37,123.3,150.24
,148.24,162.26,108.75,112.97,79.03,657.89,662.35,628.85,781.68,792.15,718.59,534.88,536.02,526.87,67.98,67.06,73.98,77.43,78.74,69.54,58.59,55.69,79.03,1255.86,1342.79,690.51,1152.25,1243.3
7,602.69,1358.82,1439.56,790.31,895.1,892.43,912.45,658.88,652.82,695.41,1129.82,1125.66,1159.11,5052.82,5204.08,4069.05,4271.51,4522.39,2758.46,5829.18,5867.63,5558.48,7300.13,7535.29,5770
.65,6151.15,6479.23,4172.46,8441.83,8563.24,7586.93,99.68,99.72,99.36,99.75,99.72,100.0,99.63,99.72,98.96,26.15,25.47,31.97,26.95,26.62,30.0,25.57,24.62,33.22,11465.69,11220.87,13057.95,981
9.61,9618.28,11033.84,13101.35,12780.83,15358.27,94.15,93.57,97.36,95.03,94.49,97.9,93.49,92.9,96.91,545.26,348.51,1824.91,638.99,387.49,2155.77,452.13,310.57,1448.89,976.91,740.36,2515.41,
910.94,669.94,2364.39,1042.47,808.91,2687.04,195.81,193.68,209.62,195.94,178.46,301.34,195.68,208.5,105.37,672.95,721.44,357.58,907.31,973.2,509.97,440.07,476.38,184.4,851.41,855.3,826.14,6
00.08,638.09,370.89,1101.15,1066.73,1343.52,8489.8,8011.14,11602.96,7684.27,7309.52,9944.37,9290.23,8694.08,13487.88,76.99,77.22,75.98,78.15,78.38,77.16,76.04,76.27,75.0,49.09,49.41,47.68,4
6.0,47.82,38.19,51.63,50.7,55.82,15.87,16.51,11.7,11.24,11.73,8.1,1.85,null,null,40.1,40.17,39.08,26.79,27.09,24.1,76.54,72.25,88.57,41.83,40.97,50.0,22.7,22.5,24.2,22.0,21.8,23.6,38.89,37.
19,54.04,2.92,2.99,2.46,2.64,2.7,2.32,4.15,4.29,3.38,7.6,6.52,16.27,41.62,38.75,50.91,23.3,20.95,30.91,2.71,2.81,2.36,42.39,40.32,49.09,35.65,34.31,40.0,68.75,68.47,70.5,61.95,62.25,60.12,4
3.43,45.73,29.06,3.07,3.36,1.26,0.66,0.54,1.41,6.59,6.84,5.04,7.09,4.73,21.79,1.04,0.98,1.41,6.8,6.23,10.38,4.55,4.4,5.49,1.25,1.05,2.52,0.93,0.71,2.3,9.44,10.0,5.93,4.26,4.32,3.85,13.7,14.
32,9.79,75.6,73.25,94.74,82.04,80.82,92.66,58.7,56.4,78.76,45.95,43.06,71.04,79.99,78.8,90.35,18.33,16.49,34.36,14.91,12.98,31.66,92.21,93.27,84.17,49.61,46.71,74.9,43.43,39.94,73.75,33.95,
30.8,61.39,57.13,54.42,80.69,49.87,48.73,59.85,6.26,4.89,18.15,41.03,43.62,18.53,25.64,24.76,43.75,62.45,59.6,87.26,12.53,13.03,9.03,36.31,29.12,53.19,30.69,30.68,30.73,57.57,54.78,81.85,62
.66,60.24,83.78,34.73,36.9,15.83,54.18,51.07,81.25,49.29,48.48,56.37,84.91,87.1,71.78,94.94,95.13,93.55,86.13,85.08,95.16,91.61,91.13,95.7,79.64,79.52,80.65,79.25,79.02,81.18,83.81,83.17,89
.25,74.73,74.22,79.03,87.27,86.48,94.09,6.97,7.35,3.76,67.42,67.07,70.43,20.96,21.02,20.43,37.56,34.7,63.33,24.13,22.69,31.09,9.65,9.26,12.38,96.52,95.84,100.0,17.62,16.72,23.81,94.91,93.86
,100.0,19.02,17.36,30.48,94.11,92.6,100.0,78.6,79.48,70.74,41.26,38.87,61.8,40.91,42.13,30.0,47.39,47.68,44.81,7.94,8.32,4.44,4.59,4.9,1.85,4.32,4.59,1.85,4.25,4.23,4.45,4.14,4.16,3.93,6.4,
6.39,6.4,7.74,7.69,8.22,7.67,7.64,7.93,72.8,71.34,85.93,36.89,35.93,45.56,87.81,86.32,98.89,95.74,95.37,98.52,96.31,95.95,99.04,91.61,91.32,93.77,4.63,5.39,3.86,4.78,5.72,3.86,3.6,3.4,3.82,
41.57,43.38,39.58,42.81,46.38,38.96,30.1,17.86,45.8,26.26,27.3,16.72,15.31,15.51,13.38,51.0,52.0,50.0,54.0,56.0,51.0,34.0,22.0,50.0,14.87,16.87,15.49,17.53,10.0,13.39,4.27,4.99,4.4,5.17,2.9
6,4.24,34.12,49.02,34.64,50.99,21.55,38.65,44.0,59.0,46.0,61.0,14.0,55.0,850.0,972.0,862.0,991.0,610.0,989.0
You have new mail in /var/spool/mail/root
[root@ip-10-0-0-14 ~]# 
```

II : EXTERNAL TABLE CREATION IN HIVE and LOADING THE INGESTED DATA INTO IT:

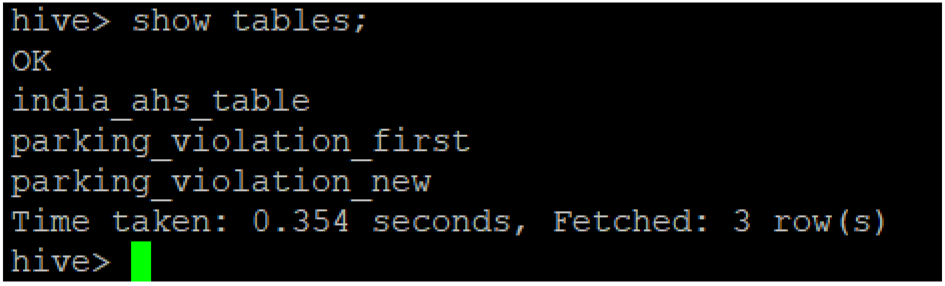
For Using the Data outside HIVE and for performing analysis on the Data external tables are Created.

Log in to HIVE using the HIVE command.

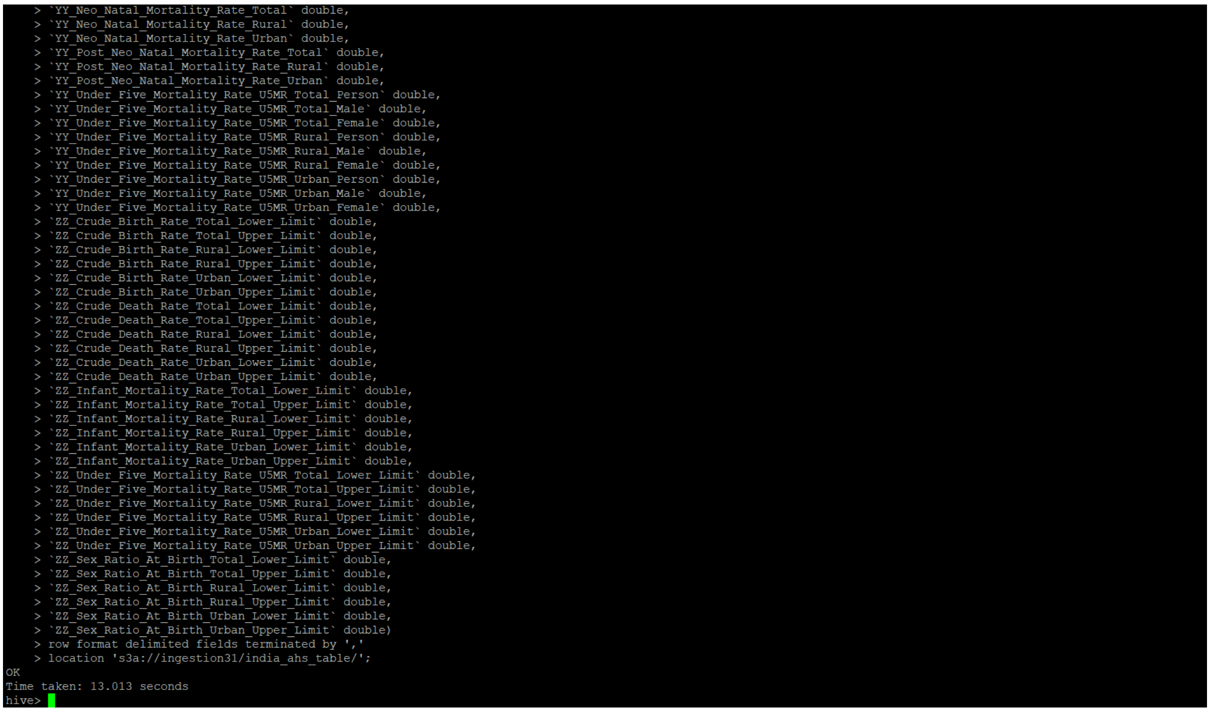
a) Creation of External table using Query :
create external table if not exists india_ahs_table with all fiels as taken as reference from Key_indicator_districtwise in Relational Database. (Not mentioning the fields due to space constrain,PFB Screenshots of create table statement with all fields)



```
hive> create external table if not exists india_ahs_table(
    > `ID` int,
    > `State_Name` string,
    > `State_District_Name` string,
    > `AA_Sample_Units_Total` double,
    > `AA_Sample_Units_Rural` double,
    > `AA_Sample_Units_Urban` double,
    > `AA_Households_Total` double,
    > `AA_Households_Rural` double,
    > `AA_Households_Urban` double,
    > `AA_Population_Total` double,
    > `AA_Population_Rural` double,
    > `AA_Population_Urban` double,
    > `AA_Ever_Married_Women_Aged_15_49_Years_Total` double,
    > `AA_Ever_Married_Women_Aged_15_49_Years_Rural` double,
    > `AA_Ever_Married_Women_Aged_15_49_Years_Urban` double,
    > `AA_Currently_Married_Women_Aged_15_49_Years_Total` double,
    > `AA_Currently_Married_Women_Aged_15_49_Years_Rural` double,
    > `AA_Currently_Married_Women_Aged_15_49_Years_Urban` double,
    > `AA_Children_12_23_Months_Total` double,
    > `AA_Children_12_23_Months_Rural` double,
    > `AA_Children_12_23_Months_Urban` double,
    > `BB_Average_Household_Size_Sc_Total` double,
    > `BB_Average_Household_Size_Sc_Rural` double,
    > `BB_Average_Household_Size_Sc_Urban` double,
    > `BB_Average_Household_Size_St_Total` double,
    > `BB_Average_Household_Size_St_Rural` double,
    > `BB_Average_Household_Size_St_Urban` double,
    > `BB_Average_Household_Size_All_Total` double,
    > `BB_Average_Household_Size_All_Rural` double,
```



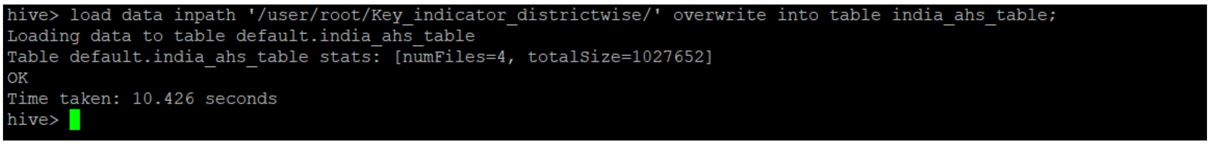
```
hive> show tables;
OK
india_ahs_table
parking_violation_first
parking_violation_new
Time taken: 0.354 seconds, Fetched: 3 row(s)
hive>
```



```
    > `YY_Neo_Natal_Mortality_Rate_Total` double,
    > `YY_Neo_Natal_Mortality_Rate_Rural` double,
    > `YY_Neo_Natal_Mortality_Rate_Urban` double,
    > `YY_Post_Neo_Natal_Mortality_Rate_Total` double,
    > `YY_Post_Neo_Natal_Mortality_Rate_Rural` double,
    > `YY_Post_Neo_Natal_Mortality_Rate_Urban` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Total_Person` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Total_Male` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Total_Female` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Rural_Person` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Rural_Male` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Rural_Female` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Urban_Person` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Urban_Male` double,
    > `YY_Under_Five_Mortality_Rate_U5MR_Urban_Female` double,
    > `ZZ_Crude_Birth_Rate_Total_Lower_Limit` double,
    > `ZZ_Crude_Birth_Rate_Total_Upper_Limit` double,
    > `ZZ_Crude_Birth_Rate_Rural_Lower_Limit` double,
    > `ZZ_Crude_Birth_Rate_Rural_Upper_Limit` double,
    > `ZZ_Crude_Birth_Rate_Urban_Lower_Limit` double,
    > `ZZ_Crude_Birth_Rate_Urban_Upper_Limit` double,
    > `ZZ_Crude_Death_Rate_Total_Lower_Limit` double,
    > `ZZ_Crude_Death_Rate_Total_Upper_Limit` double,
    > `ZZ_Crude_Death_Rate_Rural_Lower_Limit` double,
    > `ZZ_Crude_Death_Rate_Rural_Upper_Limit` double,
    > `ZZ_Crude_Death_Rate_Urban_Lower_Limit` double,
    > `ZZ_Crude_Death_Rate_Urban_Upper_Limit` double,
    > `ZZ_Infant_Mortality_Rate_Total_Lower_Limit` double,
    > `ZZ_Infant_Mortality_Rate_Total_Upper_Limit` double,
    > `ZZ_Infant_Mortality_Rate_Rural_Lower_Limit` double,
    > `ZZ_Infant_Mortality_Rate_Rural_Upper_Limit` double,
    > `ZZ_Infant_Mortality_Rate_Urban_Lower_Limit` double,
    > `ZZ_Infant_Mortality_Rate_Urban_Upper_Limit` double,
    > `ZZ_Under_Five_Mortality_Rate_U5MR_Total_Lower_Limit` double,
    > `ZZ_Under_Five_Mortality_Rate_U5MR_Total_Upper_Limit` double,
    > `ZZ_Under_Five_Mortality_Rate_U5MR_Rural_Lower_Limit` double,
    > `ZZ_Under_Five_Mortality_Rate_U5MR_Rural_Upper_Limit` double,
    > `ZZ_Under_Five_Mortality_Rate_U5MR_Urban_Lower_Limit` double,
    > `ZZ_Under_Five_Mortality_Rate_U5MR_Urban_Upper_Limit` double,
    > `ZZ_Sex_Ratio_At_Birth_Total_Lower_Limit` double,
    > `ZZ_Sex_Ratio_At_Birth_Total_Upper_Limit` double,
    > `ZZ_Sex_Ratio_At_Birth_Rural_Lower_Limit` double,
    > `ZZ_Sex_Ratio_At_Birth_Rural_Upper_Limit` double,
    > `ZZ_Sex_Ratio_At_Birth_Urban_Lower_Limit` double,
    > `ZZ_Sex_Ratio_At_Birth_Urban_Upper_Limit` double)
    > row format delimited fields terminated by ','
    > location 's3a://ingestion31/india_ahs_table/';
OK
Time taken: 13.013 seconds
hive>
```

The External Table has been created successfully.

b) Loading of Data from HDFS to HIVE:



```
hive> load data inpath '/user/root/Key_indicator_districtwise/' overwrite into table india_ahs_table;
Loading data to table default.india_ahs_table
Table default.india_ahs_table stats: [numFiles=4, totalSize=1027652]
OK
Time taken: 10.426 seconds
hive>
```

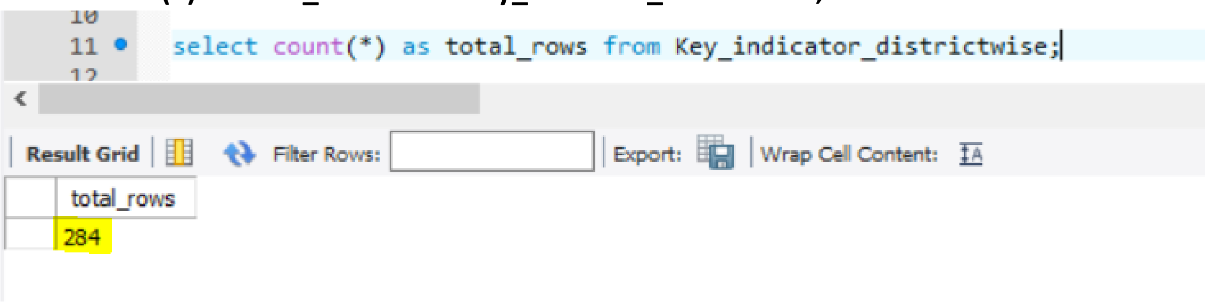
load data inpath '/user/root/Key_indicator_districtwise/' overwrite into table india_ahs_table;

c) Verfying Ingested Data and Query Validation in MySQL and HUE:

1. From the above query we can see that the Data stored in HDFS is now loaded into HIVE below are the comparisons from MySQL and HUE.

MySQL: Query to Count total number of Rows

select count(*) as total_rows from Key_indicator_districtwise;

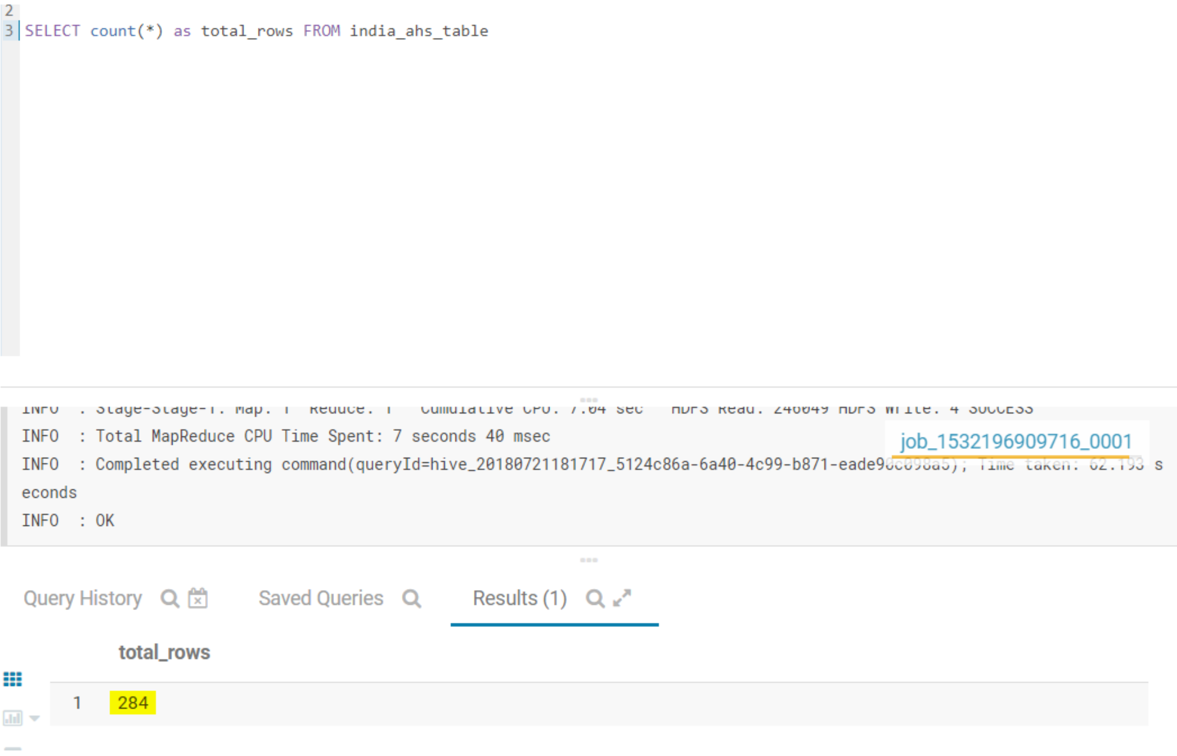


```
select count(*) as total_rows from Key_indicator_districtwise;
```

total_rows
284

Hue:

SELECT count(*) as total_rows FROM india_ahs_table



```
SELECT count(*) as total_rows FROM india_ahs_table
```

```
INFO  : Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 7.04 sec   HDFS Read: 246049 HDFS Write: 4 SUCCESS
INFO  : Total MapReduce CPU Time Spent: 7 seconds 40 msec
INFO  : Completed executing command(queryId=hive_20180721181717_5124c86a-6a40-4c99-b871-eade9dc098a5); Time taken: 62.193 s
econds
INFO  : OK
```

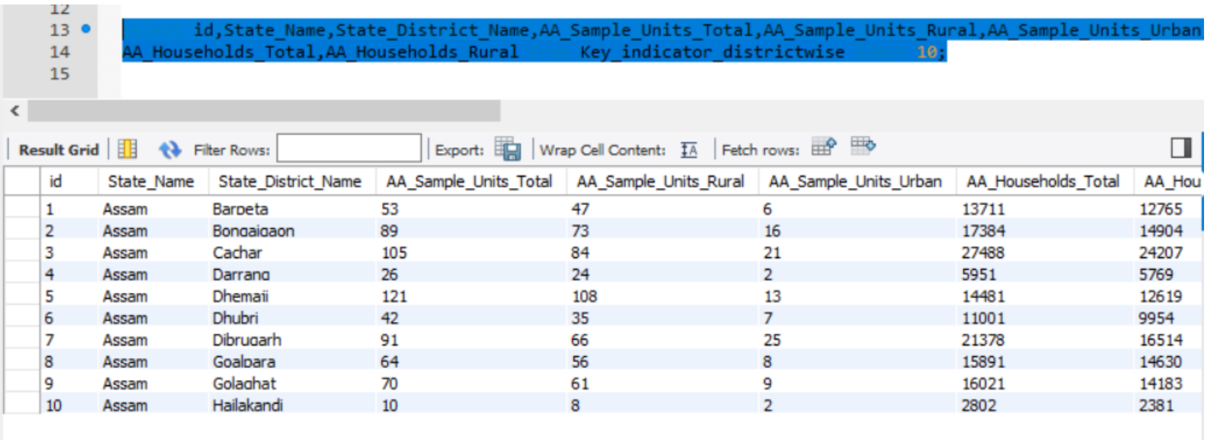
job_1532196909716_0001

	total_rows
1	284

2. Selecting Top 10 Rows and 8 Coloums from the Tables in MySQL workbench and HUE

MySQL:

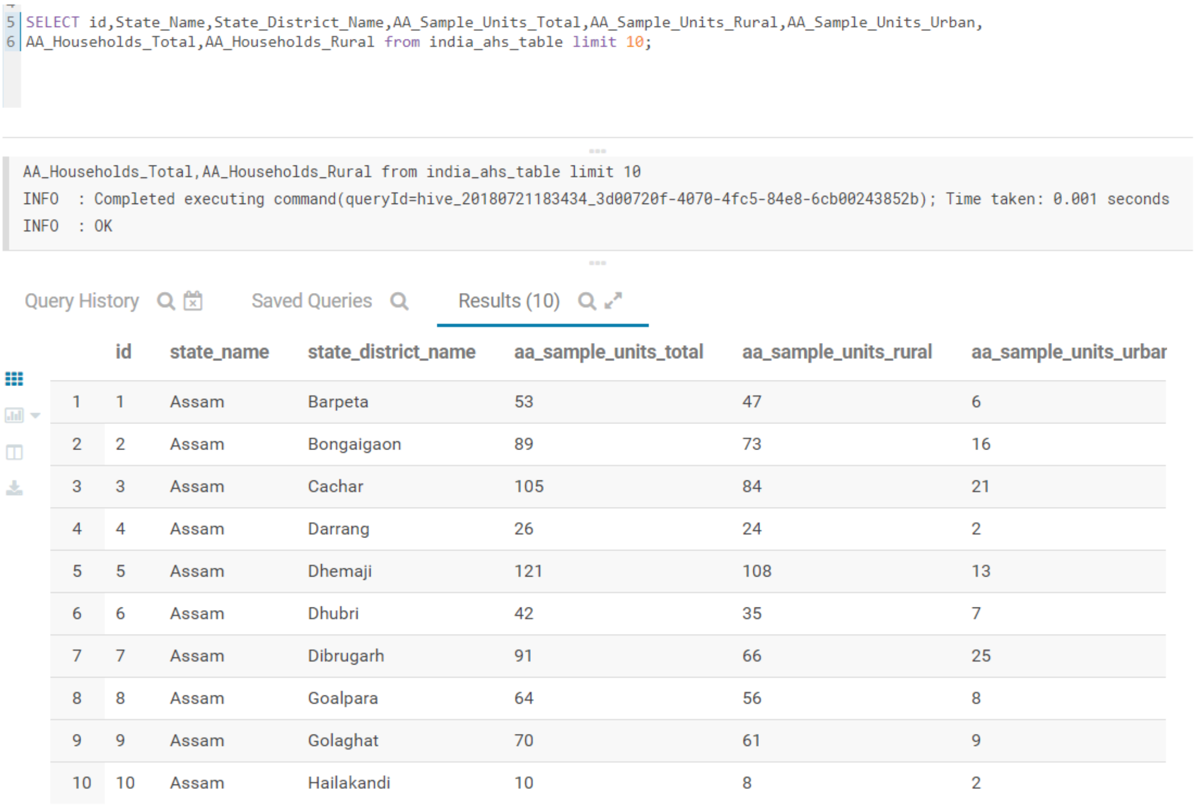
select id,State_Name,State_District_Name,AA_Sample_Units_Total,AA_Sample_Units_Rural,AA_Sample_Units_Urban, AA_Households_Total,AA_Households_Rural from Key_indicator_districtwise limit 10;



```
       id,State_Name,State_District_Name,AA_Sample_Units_Total,AA_Sample_Units_Rural,AA_Sample_Units_Urban
AA_Households_Total,AA_Households_Rural       Key_indicator_districtwise       10;
```

id	State_Name	State_District_Name	AA_Sample_Units_Total	AA_Sample_Units_Rural	AA_Sample_Units_Urban	AA_Households_Total	AA_Hou
1	Assam	Barpeta	53	47	6	13711	12765
2	Assam	Bongaigaon	89	73	16	17384	14904
3	Assam	Cachar	105	84	21	27488	24207
4	Assam	Darrang	26	24	2	5951	5769
5	Assam	Dhemaji	121	108	13	14481	12619
6	Assam	Dhubri	42	35	7	11001	9954
7	Assam	Dibrugarh	91	66	25	21378	16514
8	Assam	Goalpara	64	56	8	15891	14630
9	Assam	Golaghat	70	61	9	16021	14183
10	Assam	Hailakandi	10	8	2	2802	2381

Hue:



```
SELECT id,State_Name,State_District_Name,AA_Sample_Units_Total,AA_Sample_Units_Rural,AA_Sample_Units_Urban,
AA_Households_Total,AA_Households_Rural from india_ahs_table limit 10;
```

```
AA_Households_Total,AA_Households_Rural from india_ahs_table limit 10
INFO  : Completed executing command(queryId=hive_20180721183434_3d00720f-4070-4fc5-84e8-6cb00243852b); Time taken: 0.001 seconds
INFO  : OK
```

	id	state_name	state_district_name	aa_sample_units_total	aa_sample_units_rural	aa_sample_units_urbar
1	1	Assam	Barpeta	53	47	6
2	2	Assam	Bongaigaon	89	73	16
3	3	Assam	Cachar	105	84	21
4	4	Assam	Darrang	26	24	2
5	5	Assam	Dhemaji	121	108	13
6	6	Assam	Dhubri	42	35	7
7	7	Assam	Dibrugarh	91	66	25
8	8	Assam	Goalpara	64	56	8
9	9	Assam	Golaghat	70	61	9
10	10	Assam	Hailakandi	10	8	2

III . SUBSET SCHEMA CREATION IN HIVE TO SUPPORT THE ANALYSES

The Columns which are used in Subset Schema is for the analysis and below are the one's which have been used to create a Table in default format and one with the ORC format for Efficiency. Since ORC format provides faster retrieval since data is compressed it is compared with the default format here.

A. Non-Partition Table Creation and Insertion for Default and ORC Formats

Columns Used - **`ID` ,`State_Name` ,`State_District_Name` ,`AA_Population_Total` , AA_Households_Total,`BB_Population_Below_Age_15_Years_Total` ,`CC_Sex_Ratio_All_Ages_Total` ,,`LL_Total_Fertility_Rate_Total` ,`YY_Infant_Mortality_Rate_Imr_Total_Person` ,`YY_Under_Five_Mortality_Rate_U5MR_Total_Person`**

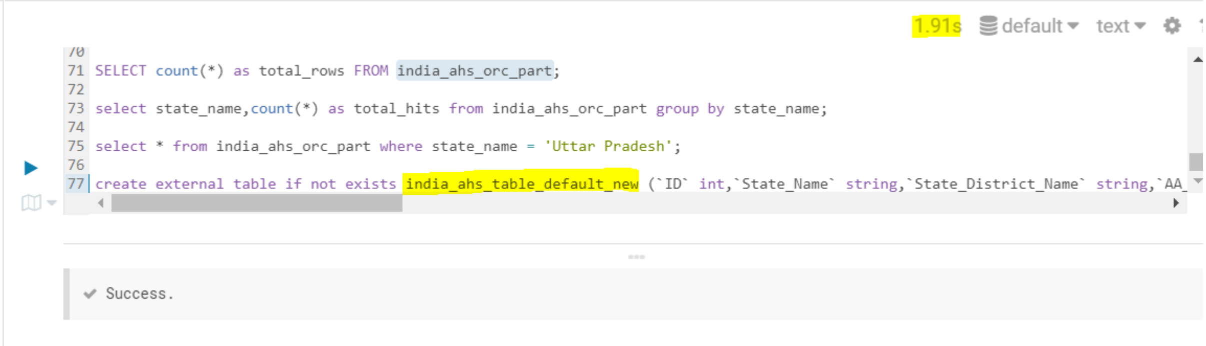
DEFAULT FORMAT – NON PARTITION TABLE:

1a) Creation of Default table -

Query:

create external table if not exists india_ahs_table_default_new(`ID` int,`State_Name` string,`State_District_Name` string,`AA_Population_Total` double,`AA_Households_Total` double,`BB_Population_Below_Age_15_Years_Total` double,`CC_Sex_Ratio_All_Ages_Total` double,`LL_Total_Fertility_Rate_Total` double,`YY_Infant_Mortality_Rate_Imr_Total_Person`

double,`YY_Under_Five_Mortality_Rate_U5MR_Total_Person` double) row format delimited fields terminated by ',';



```
SELECT count(*) as total_rows FROM india_ahs_orc_part;

select state_name,count(*) as total_hits from india_ahs_orc_part group by state_name;

select * from india_ahs_orc_part where state_name = 'Uttar Pradesh';

create external table if not exists india_ahs_table_default_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_
```

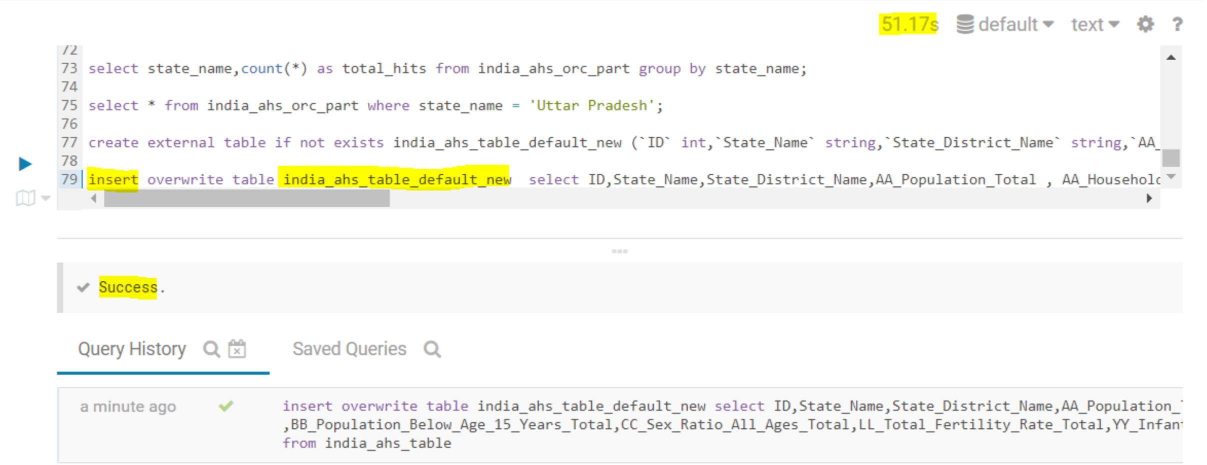
```
✓ Success.
```

Time Taken for Create Query in Default format – 1.91 Seconds

1b) Insertion into Default table-

Query:

insert overwrite table india_ahs_table_default_new select ID,State_Name,State_District_Name,AA_Population_Total , AA_Households_Total ,BB_Population_Below_Age_15_Years_Total,CC_Sex_Ratio_All_Ages_Total,LL_Total_Fertility_Rate_Total,YY_Infant_Mortality_Rate_Imr_Total_Person,YY_Under_Five_Mortality_Rate_U5MR_Total_Person from india_ahs_table;



```
select state_name,count(*) as total_hits from india_ahs_orc_part group by state_name;

select * from india_ahs_orc_part where state_name = 'Uttar Pradesh';

create external table if not exists india_ahs_table_default_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_

insert overwrite table india_ahs_table_default_new  select ID,State_Name,State_District_Name,AA_Population_Total , AA_Householc
```

```
✓ Success.
```

```
a minute ago  ✓  insert overwrite table india_ahs_table_default_new select ID,State_Name,State_District_Name,AA_Population_
                 ,BB_Population_Below_Age_15_Years_Total,CC_Sex_Ratio_All_Ages_Total,LL_Total_Fertility_Rate_Total,YY_Infan
                 from india_ahs_table
```

Time Taken for Insert Query in Default format – 51.17 secs

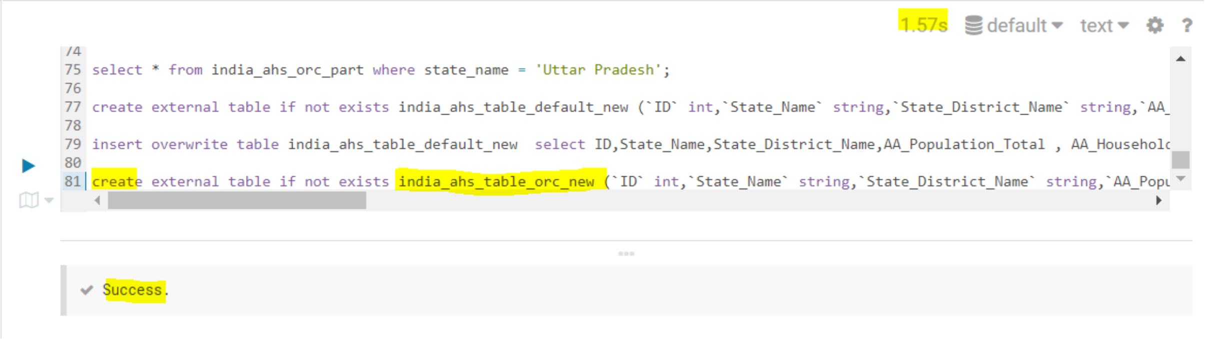
ORC FORMAT – NON PARTITION TABLE :

2a) Creation of ORC Format table-

Query:

create external table if not exists india_ahs_table_orc_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_Population_Total` double, `AA_Households_Total` double,`BB_Population_Below_Age_15_Years_Total` double,`CC_Sex_Ratio_All_Ages_Total` double,`LL_Total_Fertility_Rate_Total` double,`YY_Infant_Mortality_Rate_Imr_Total_Person`

double,`YY_Under_Five_Mortality_Rate_U5MR_Total_Person` double) row format delimited fields terminated by ',' stored as orc TBLPROPERTIES ('orc.compress'='SNAPPY') ;



```
select * from india_ahs_orc_part where state_name = 'Uttar Pradesh';

create external table if not exists india_ahs_table_default_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_

insert overwrite table india_ahs_table_default_new  select ID,State_Name,State_District_Name,AA_Population_Total , AA_Householc

create external table if not exists india_ahs_table_orc_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_Popu
```

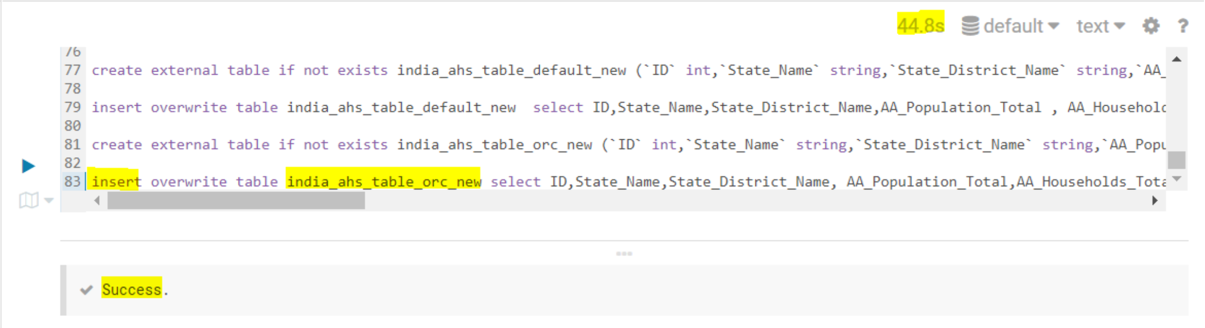
```
✓ Success.
```

Time Taken for Create Query in ORC format – 1.57 Seconds

2b) Insertion into ORC table-

Query:

insert overwrite table india_ahs_table_orc_new select ID,State_Name,State_District_Name, AA_Population_Total,AA_Households_Total,BB_Population_Below_Age_15_Years_Total,CC_Sex_Ratio_All_Ages_Total,LL_Total_Fertility_Rate_Total,YY_Infant_Mortality_Rate_Imr_Total_Person, YY_Under_Five_Mortality_Rate_U5MR_Total_Person from india_ahs_table;



```
create external table if not exists india_ahs_table_default_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_

insert overwrite table india_ahs_table_default_new  select ID,State_Name,State_District_Name,AA_Population_Total , AA_Householc

create external table if not exists india_ahs_table_orc_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_Popu

insert overwrite table india_ahs_table_orc_new select ID,State_Name,State_District_Name, AA_Population_Total,AA_Households_Tota
```

```
✓ Success.
```

Time Taken for Insert Query in ORC format – 44.8 secs

B. CREATION AND INSERTION in PARTITION TABLES for Default and ORC Formats:

Partitioned with State_Name

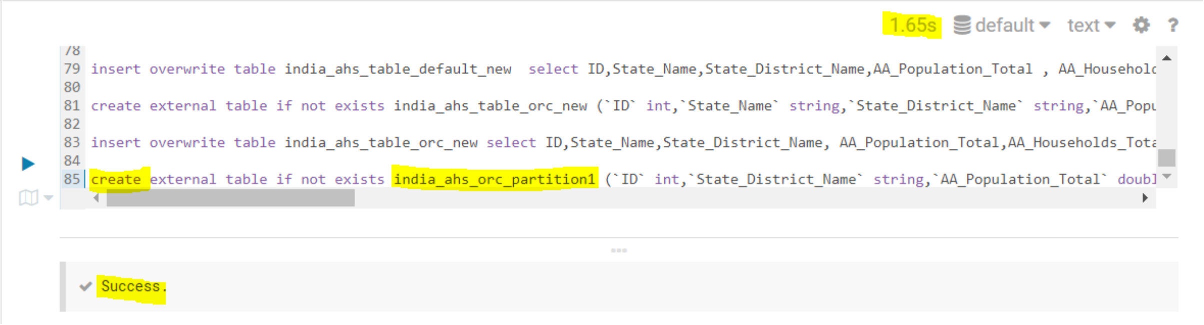
ORC FORMAT – PARTITION TABLE:

Creation of External Table with Partition in ORC Format

Query:

create external table if not exists india_ahs_orc_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Total` double, `AA_Households_Total` double ,`BB_Population_Below_Age_15_Years_Total` double,`CC_Sex_Ratio_All_Ages_Total` double,`LL_Total_Fertility_Rate_Total` double,`YY_Infant_Mortality_Rate_Imr_Total_Person` double,`YY_Under_Five_Mortality_Rate_U5MR_Total_Person` double) partitioned by

(State_Name string) row format delimited fields terminated by ',' stored as orc TBLPROPERTIES ('orc.compress'='SNAPPY');



```
insert overwrite table india_ahs_table_default_new  select ID,State_Name,State_District_Name,AA_Population_Total , AA_Householc

create external table if not exists india_ahs_table_orc_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_Popu

insert overwrite table india_ahs_table_orc_new select ID,State_Name,State_District_Name, AA_Population_Total,AA_Households_Tota

create external table if not exists india_ahs_orc_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Total` doubl
```

```
Success.
```

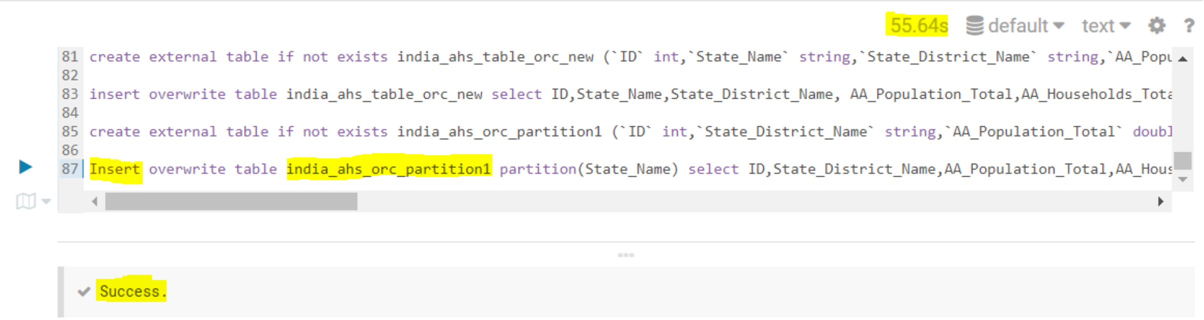
Time Taken for Create Query for Partition table in ORC format – 1.65 Seconds

1b) Insertion into External table in ORC Format

Query:

set hive.exec.dynamic.partition.mode=nonstrict; (Since transferring data from Non-partitioned table to a partitioned table in Dynamic partitioning is Strict as default)

Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID, State_District_Name,AA_Population_Total,AA_Households_Total,BB_Population_Below_Age_15_Years_Total,CC_Sex_Ratio_All_Ages_Total,LL_Total_Fertility_Rate_Total,YY_Infant_Mortality_Rate_Imr_Total_Person, YY_Under_Five_Mortality_Rate_U5MR_Total_Person,State_Name from india_ahs_table;



```
create external table if not exists india_ahs_table_orc_new (`ID` int,`State_Name` string,`State_District_Name` string,`AA_Popu

insert overwrite table india_ahs_table_orc_new select ID,State_Name,State_District_Name, AA_Population_Total,AA_Households_Tota

create external table if not exists india_ahs_orc_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Total` doubl

Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID,State_District_Name,AA_Population_Total,AA_Hous
```

```
Success.
```

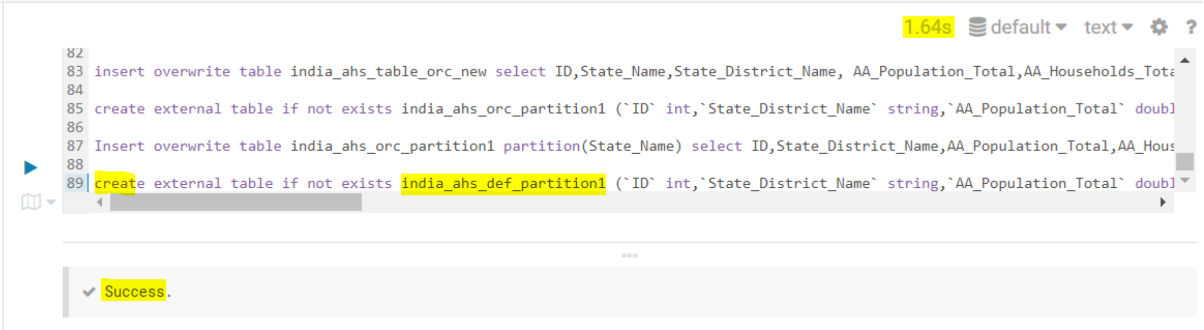
Time Taken for Insert Query for Partitioned Table in ORC format – 55.64 secs

DEFAULT FORMAT – PARTITION TABLE:

2a) Creation of External table with Partition in Default Format:

Query:

create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Total` double, `AA_Households_Total` double ,`BB_Population_Below_Age_15_Years_Total` double,`CC_Sex_Ratio_All_Ages_Total` double,`LL_Total_Fertility_Rate_Total` double,`YY_Infant_Mortality_Rate_Imr_Total_Person` double,`YY_Under_Five_Mortality_Rate_U5MR_Total_Person` double) partitioned by (State_Name string)row format delimited fields terminated by ',' ;



```
insert overwrite table india_ahs_table_orc_new select ID,State_Name,State_District_Name, AA_Population_Total,AA_Households_Tota

create external table if not exists india_ahs_orc_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Total` doubl

Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID,State_District_Name,AA_Population_Total,AA_Hous

create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Total` doubl
```

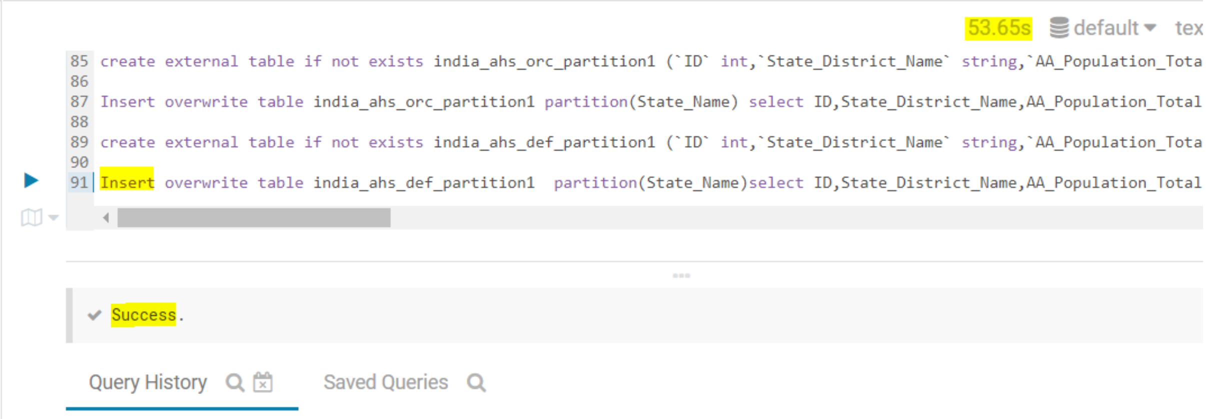
```
✓ Success.
```

Time Taken for Create Query in Default format – 1.64 Seconds

1b) Insertion into Default format Partition table-

Query:

Insert overwrite table india_ahs_def_partition1 partition(State_Name)select ID,State_District_Name,AA_Population_Total, AA_Households_Total ,BB_Population_Below_Age_15_Years_Total,CC_Sex_Ratio_All_Ages_Total,LL_Total_Fertility_Rate_Total,YY_Infant_Mortality_Rate_Imr_Total_Person,YY_Under_Five_Mortality_Rate_U5MR_Total_Person,state_name from india_ahs_table;



```
create external table if not exists india_ahs_orc_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Tota

Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID,State_District_Name,AA_Population_Total

create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_Population_Tota

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_Population_Total
```

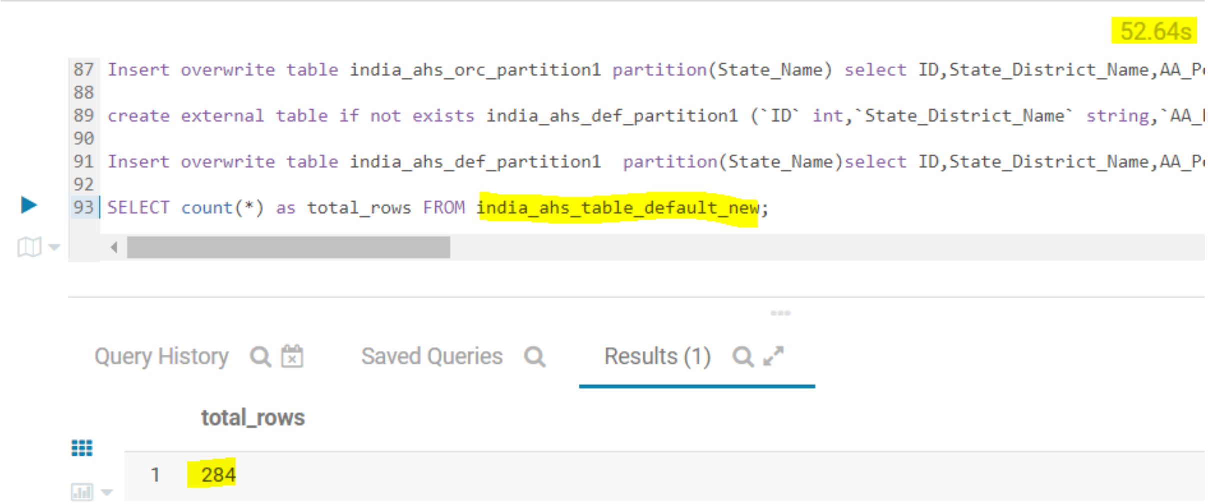
```
✓ Success.
```

Time Taken for Insert Query in Default format – 53.65 secs

C. CODE VALIDATION:

1a. Default Non Partitioned Table :

Query - SELECT count(*) as total_rows FROM india_ahs_table_default_new;
Time Taken – 52.64 secs



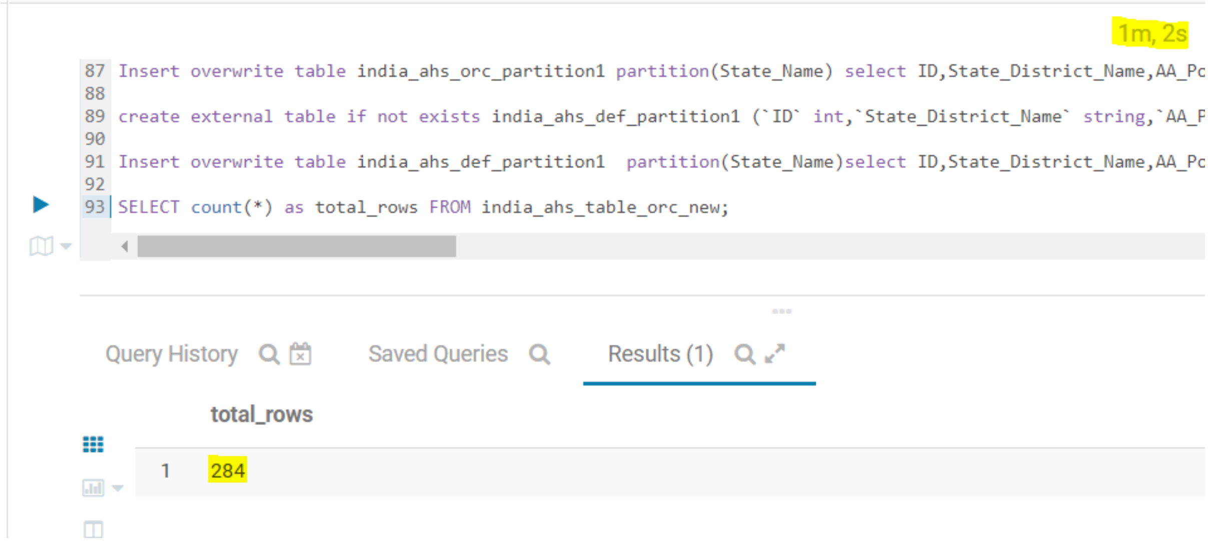
52.64s

```
Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID,State_District_Name,AA_P

create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_P

SELECT count(*) as total_rows FROM india_ahs_table_default_new;
```

	total_rows
1	284

1b. ORC Non Partitioned Table :

Query - SELECT count(*) as total_rows FROM india_ahs_table_orc_new;
Time Taken – 1 min, 2 sec



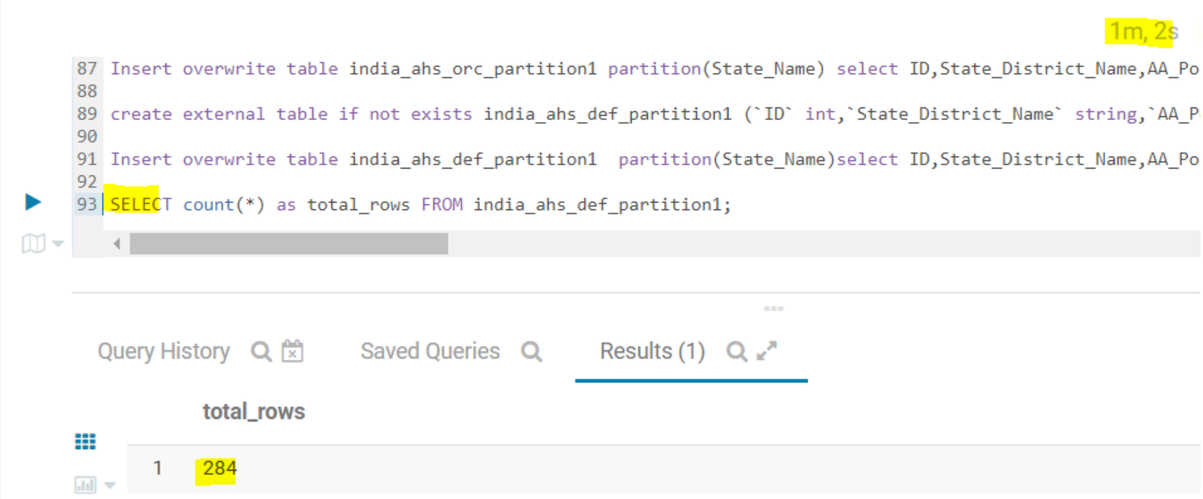
1m, 2s

```
Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID,State_District_Name,AA_Po

create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_F

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_Po

SELECT count(*) as total_rows FROM india_ahs_table_orc_new;
```

	total_rows
1	284

1c. Default Partitioned Table :

Query - SELECT count(*) as total_rows FROM india_ahs_def_partition1;
Time Taken – 1m 2 secs



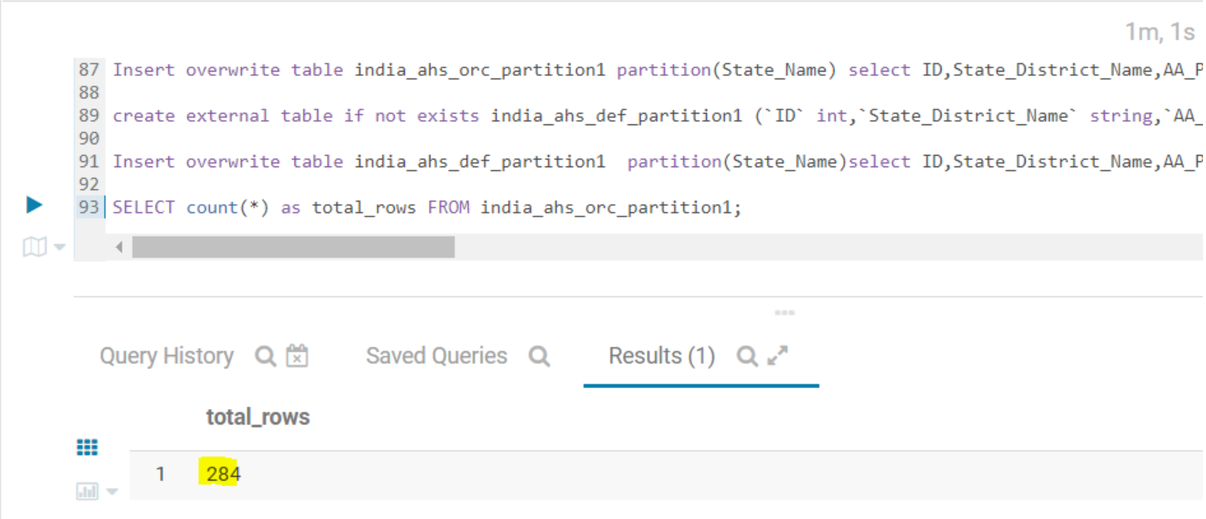
```
Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID,State_District_Name,AA_Po

create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_P

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_Po

SELECT count(*) as total_rows FROM india_ahs_def_partition1;
```

	total_rows
1	284

1d. ORC Partitioned Table :

Query - SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

Time Taken – 1min 1 Secs



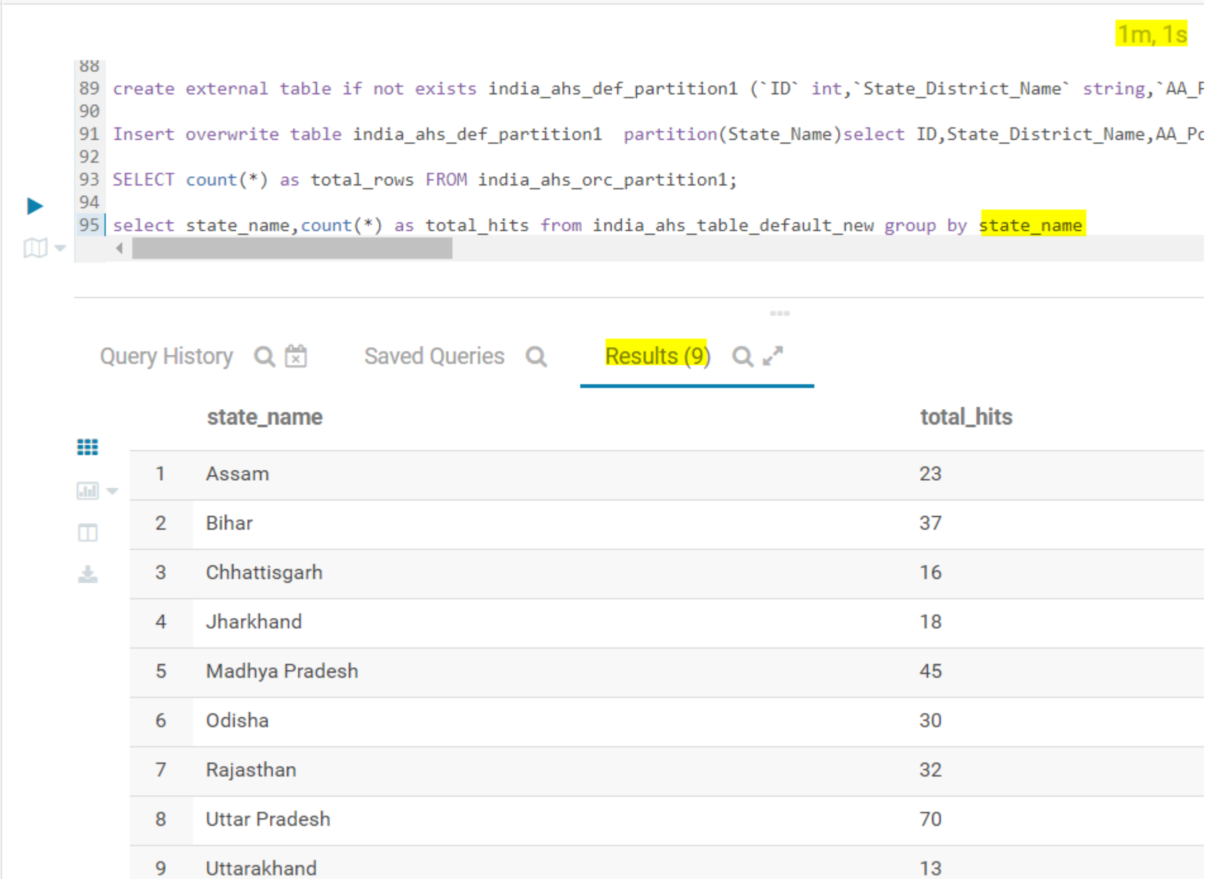
```
Insert overwrite table india_ahs_orc_partition1 partition(State_Name) select ID,State_District_Name,AA_P

create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_P

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;
```

	total_rows
1	284

2a. Default Non Partitioned Table :

Query - select state_name,count(*) as total_hits from india_ahs_table_default_new group by state_name

Time Taken – 1m 1sec



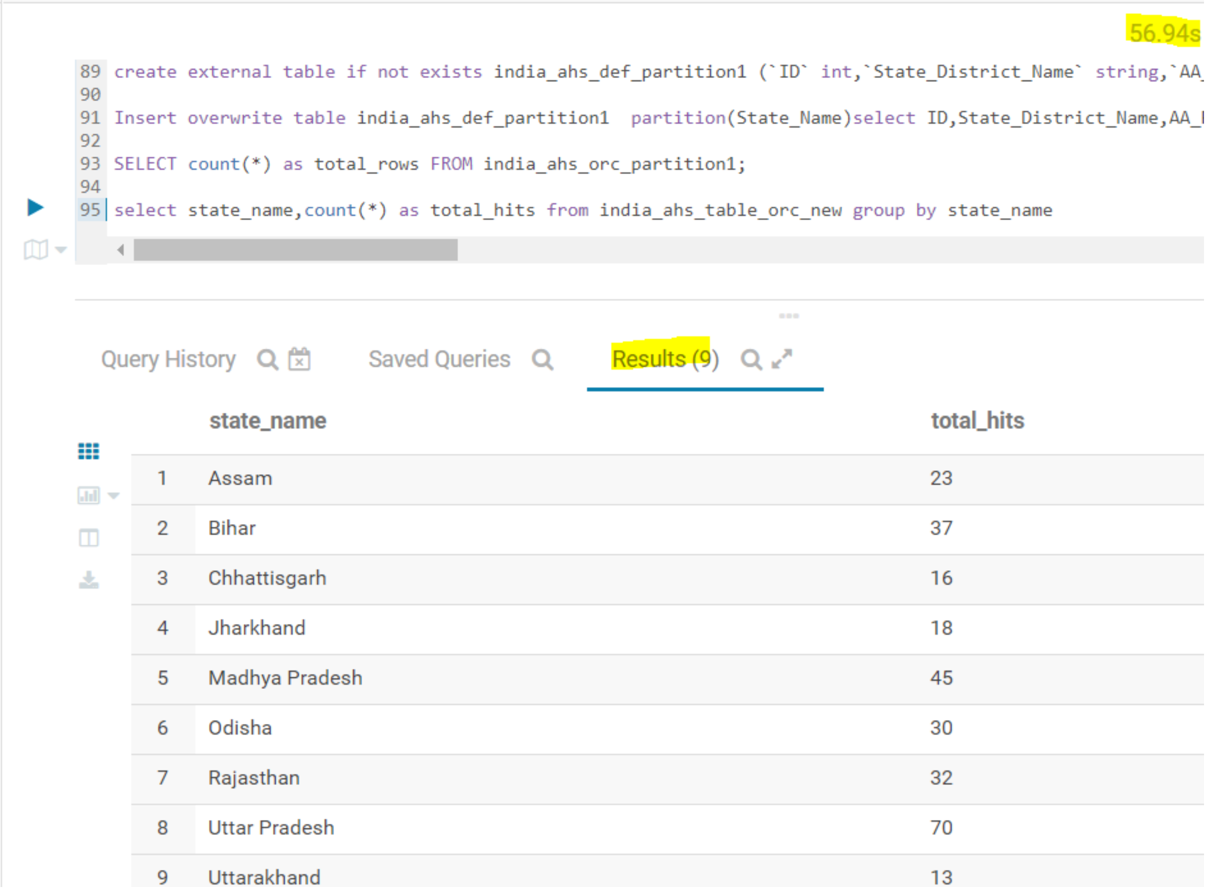
```
create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_F

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_Pc

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_table_default_new group by state_name
```

	state_name	total_hits
1	Assam	23
2	Bihar	37
3	Chhattisgarh	16
4	Jharkhand	18
5	Madhya Pradesh	45
6	Odisha	30
7	Rajasthan	32
8	Uttar Pradesh	70
9	Uttarakhand	13

2b. ORC Non Partitioned Table :

Query - select state_name,count(*) as total_hits from india_ahs_table_orc_new group by state_name

Time Taken – 56.94 secs



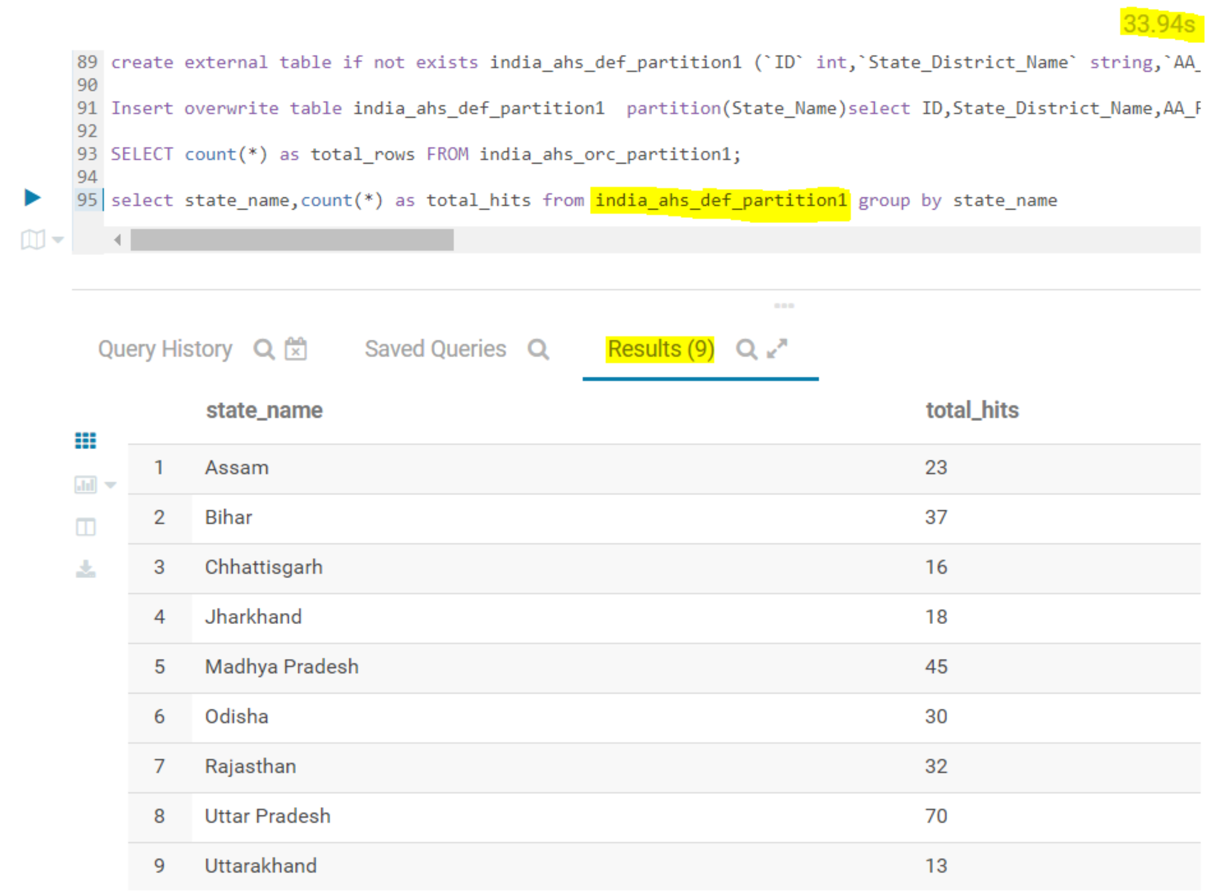
```
create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_table_orc_new group by state_name
```

	state_name	total_hits
1	Assam	23
2	Bihar	37
3	Chhattisgarh	16
4	Jharkhand	18
5	Madhya Pradesh	45
6	Odisha	30
7	Rajasthan	32
8	Uttar Pradesh	70
9	Uttarakhand	13

2c. Default Partitioned Table :

Query - select state_name,count(*) as total_hits from india_ahs_default_partition1 group by state_name

Time Taken – 33.94 secs



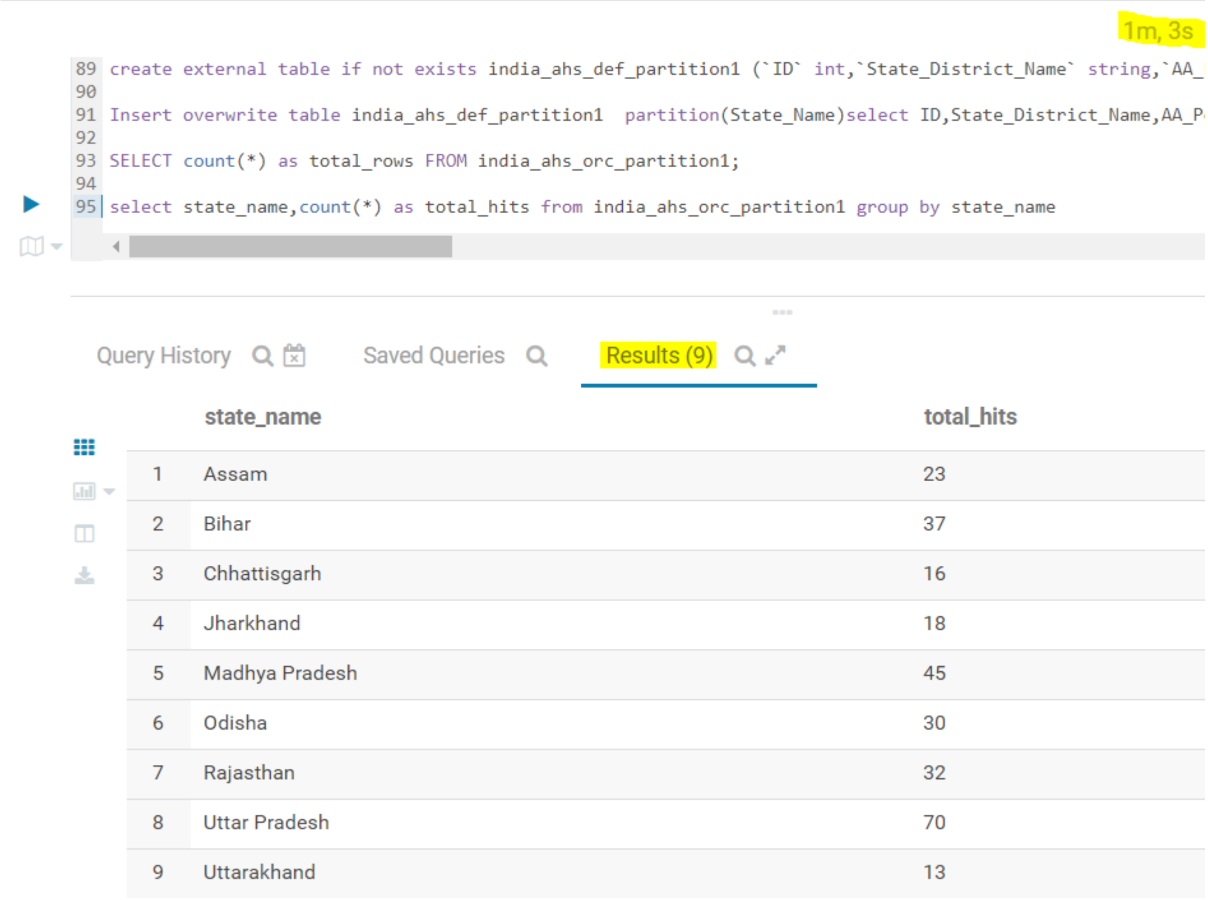
```
create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_F

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_def_partition1 group by state_name
```

	state_name	total_hits
1	Assam	23
2	Bihar	37
3	Chhattisgarh	16
4	Jharkhand	18
5	Madhya Pradesh	45
6	Odisha	30
7	Rajasthan	32
8	Uttar Pradesh	70
9	Uttarakhand	13

2d. ORC Partitioned Table :

Query -_select state_name,count(*) as total_hits from india_ahs_orc_partition1 group by state_name

Time Taken – 1m 3sec



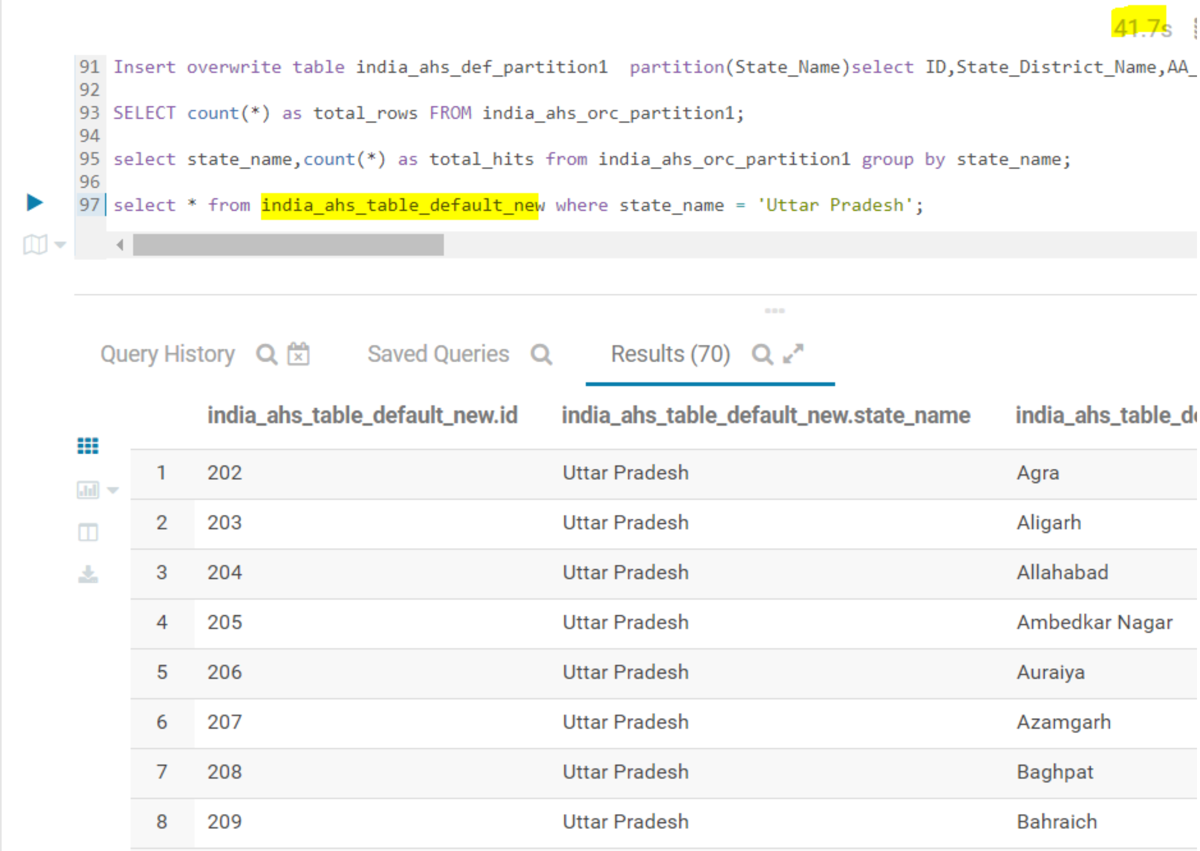
```
create external table if not exists india_ahs_def_partition1 (`ID` int,`State_District_Name` string,`AA_

Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_P

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_orc_partition1 group by state_name
```

	state_name	total_hits
1	Assam	23
2	Bihar	37
3	Chhattisgarh	16
4	Jharkhand	18
5	Madhya Pradesh	45
6	Odisha	30
7	Rajasthan	32
8	Uttar Pradesh	70
9	Uttarakhand	13

3a. Default Non Partition Table

Query - select * from india_ahs_table_default_new where state_name = 'Uttar Pradesh';

Time Taken – 41.7 secs



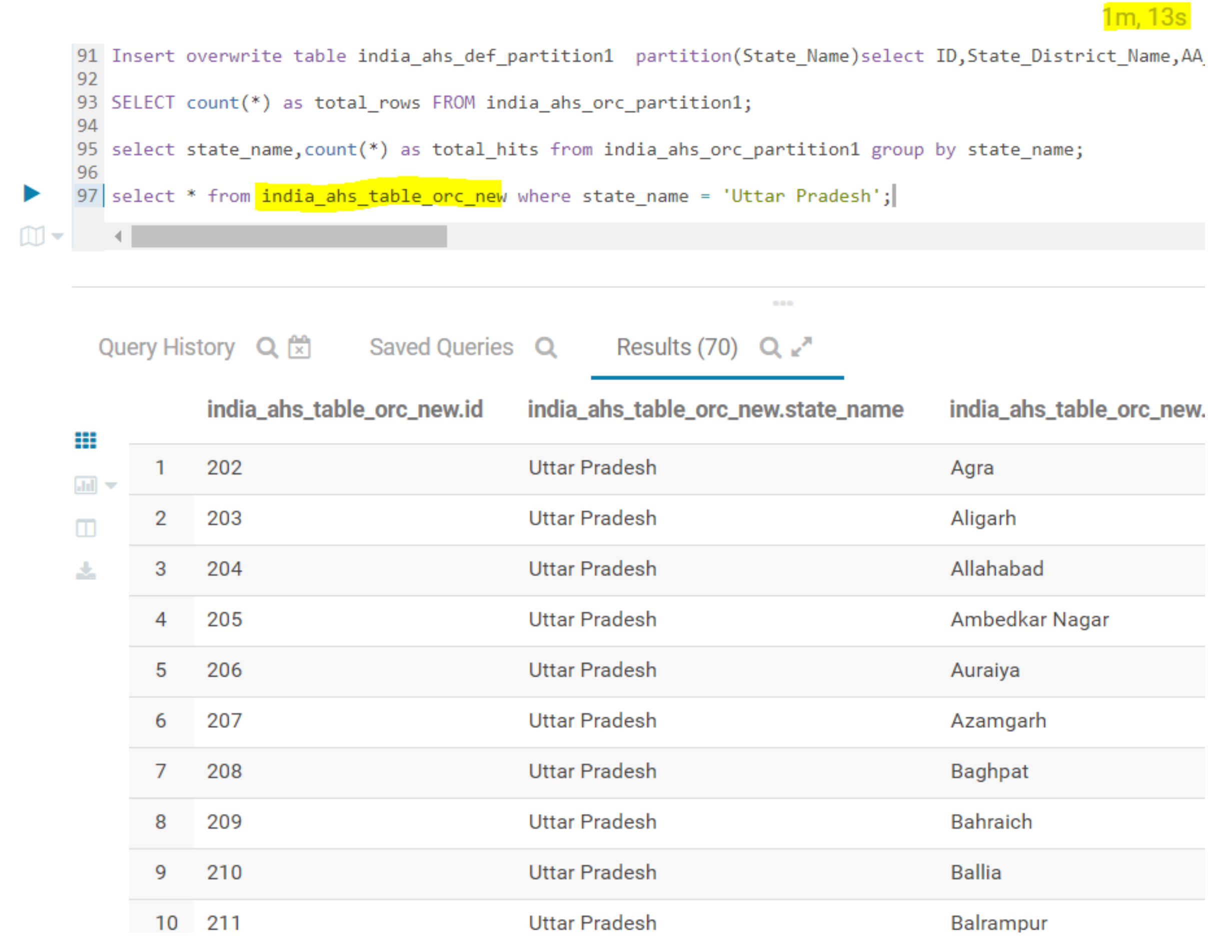
```
Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA_

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_orc_partition1 group by state_name;

select * from india_ahs_table_default_new where state_name = 'Uttar Pradesh';
```

	india_ahs_table_default_new.id	india_ahs_table_default_new.state_name	india_ahs_table_d
1	202	Uttar Pradesh	Agra
2	203	Uttar Pradesh	Aligarh
3	204	Uttar Pradesh	Allahabad
4	205	Uttar Pradesh	Ambedkar Nagar
5	206	Uttar Pradesh	Auraiya
6	207	Uttar Pradesh	Azamgarh
7	208	Uttar Pradesh	Baghpat
8	209	Uttar Pradesh	Bahraich

3b. ORC Non-Partitioned Table

Query - select * from india_ahs_table_orc_new where state_name = 'Uttar Pradesh';

Time Taken - 1m 13 sec

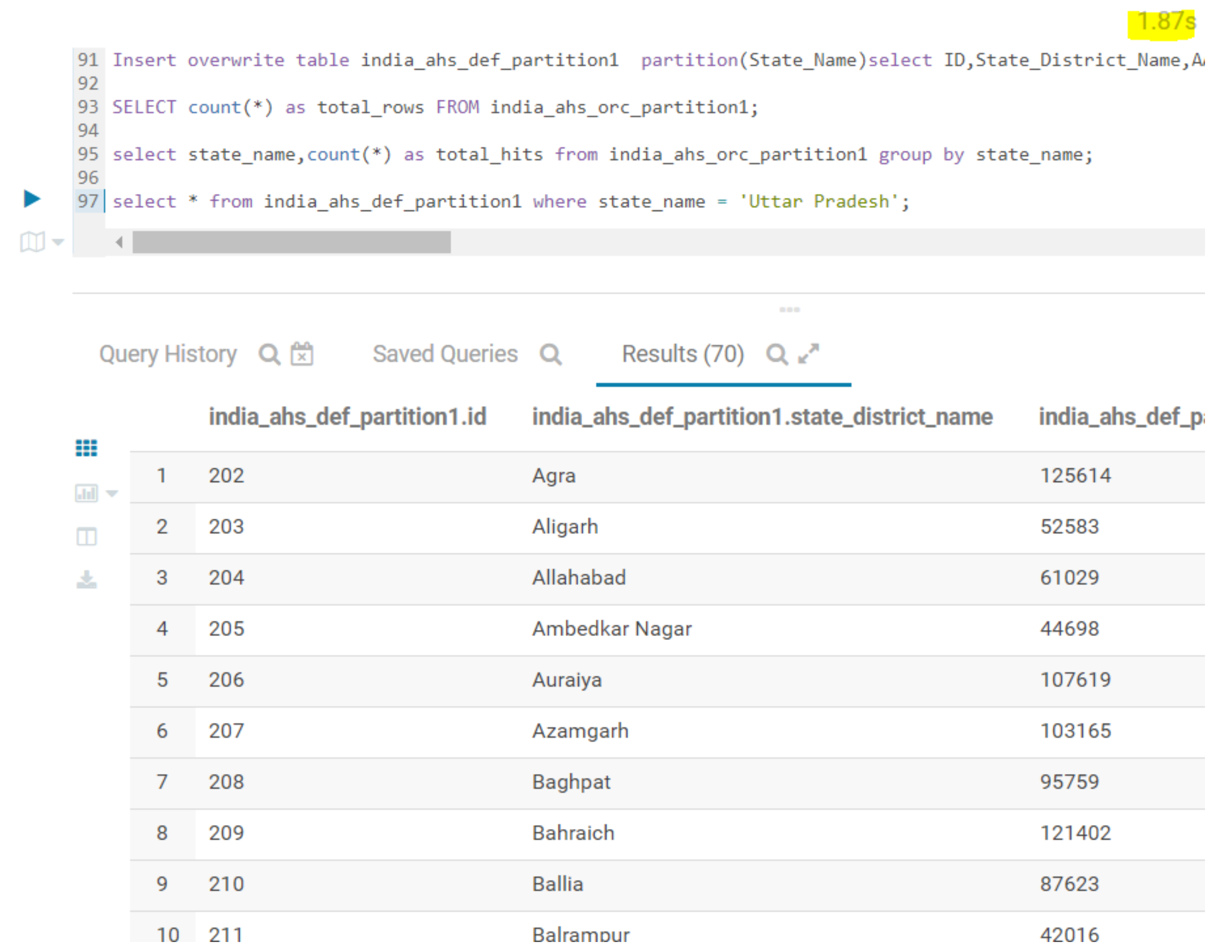


```
Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_orc_partition1 group by state_name;

select * from india_ahs_table_orc_new where state_name = 'Uttar Pradesh';
```

	india_ahs_table_orc_new.id	india_ahs_table_orc_new.state_name	india_ahs_table_orc_new.
1	202	Uttar Pradesh	Agra
2	203	Uttar Pradesh	Aligarh
3	204	Uttar Pradesh	Allahabad
4	205	Uttar Pradesh	Ambedkar Nagar
5	206	Uttar Pradesh	Auraiya
6	207	Uttar Pradesh	Azamgarh
7	208	Uttar Pradesh	Baghpat
8	209	Uttar Pradesh	Bahraich
9	210	Uttar Pradesh	Ballia
10	211	Uttar Pradesh	Balrampur

3c. Default Partitioned Table

Query -_select * from india_ahs_def_partition1 where state_name = 'Uttar Pradesh';
Time Taken - 1.87 secs



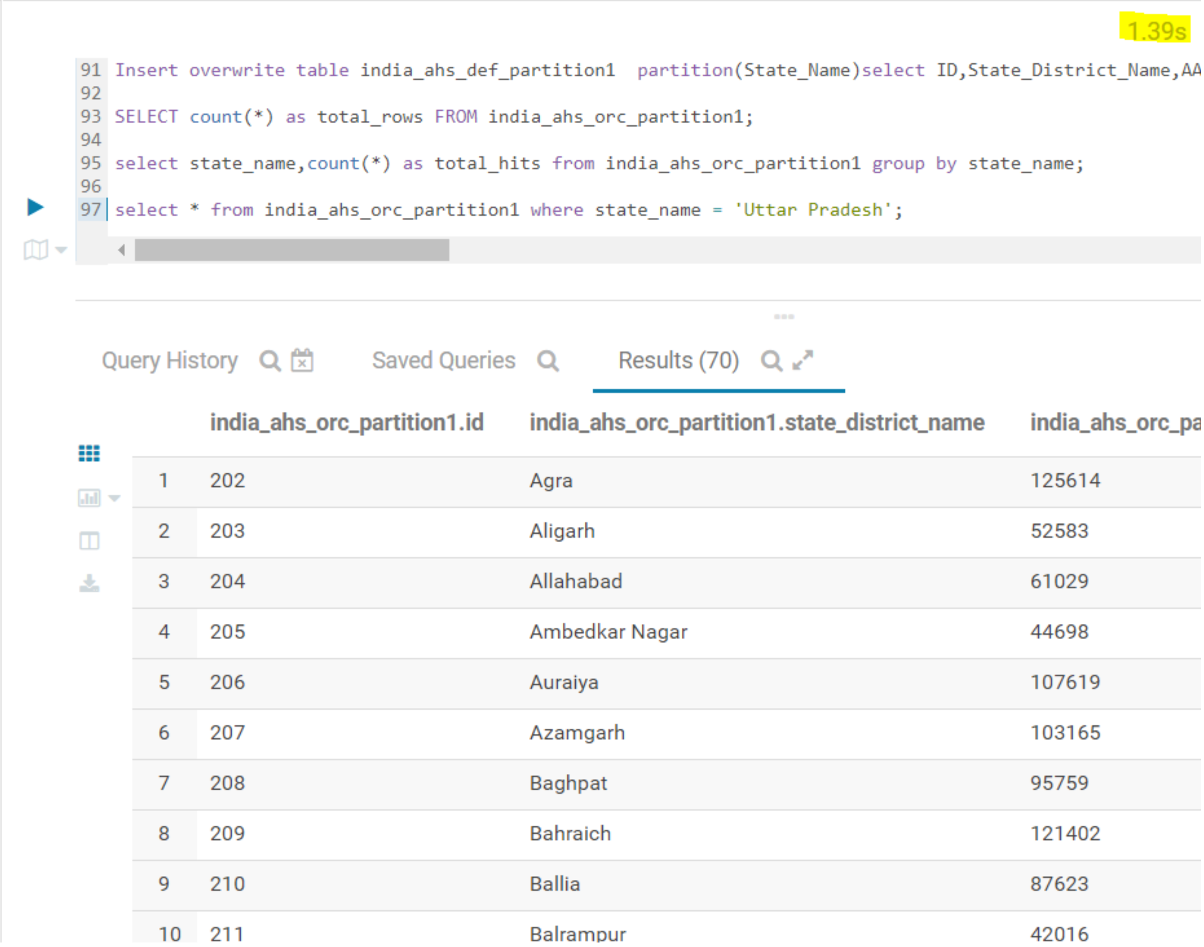
```
Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,A

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_orc_partition1 group by state_name;

select * from india_ahs_def_partition1 where state_name = 'Uttar Pradesh';
```

	india_ahs_def_partition1.id	india_ahs_def_partition1.state_district_name	india_ahs_def_p
1	202	Agra	125614
2	203	Aligarh	52583
3	204	Allahabad	61029
4	205	Ambedkar Nagar	44698
5	206	Auraiya	107619
6	207	Azamgarh	103165
7	208	Baghpat	95759
8	209	Bahraich	121402
9	210	Ballia	87623
10	211	Balrampur	42016

3d. ORC Partitioned table

Query - select * from india_ahs_orc_partition1 where state_name = 'Uttar Pradesh';

Time Taken – 1.39 secs



```
Insert overwrite table india_ahs_def_partition1  partition(State_Name)select ID,State_District_Name,AA

SELECT count(*) as total_rows FROM india_ahs_orc_partition1;

select state_name,count(*) as total_hits from india_ahs_orc_partition1 group by state_name;

select * from india_ahs_orc_partition1 where state_name = 'Uttar Pradesh';
```

	india_ahs_orc_partition1.id	india_ahs_orc_partition1.state_district_name	india_ahs_orc_pa
1	202	Agra	125614
2	203	Aligarh	52583
3	204	Allahabad	61029
4	205	Ambedkar Nagar	44698
5	206	Auraiya	107619
6	207	Azamgarh	103165
7	208	Baghpat	95759
8	209	Bahraich	121402
9	210	Ballia	87623
10	211	Balrampur	42016

Confirmation of Table storage format after comparisons made above,

“As we can see from the Results above clearly the format of ORC with Partition provides better efficiency on Analysis. Hence proceeding further with the ORC Partition table for the Analysis for Optimization Efficiency”.

IV . ANALYSES

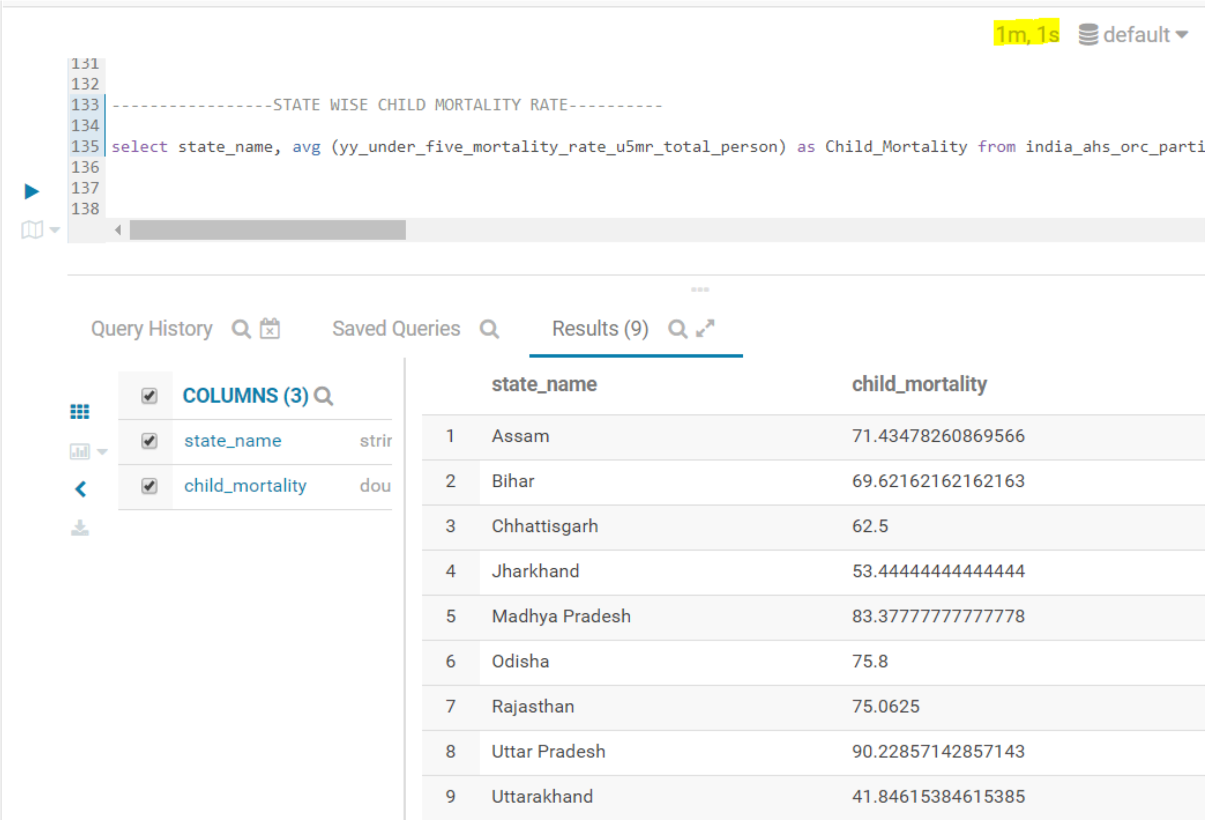
1. STATE WISE CHILD MORTALITY RATE

Query:

select state_name, avg (yy_under_five_mortality_rate_u5mr_total_person) as Child_Mortality from india_ahs_orc_partition1 group by state_name;

Screenshot:

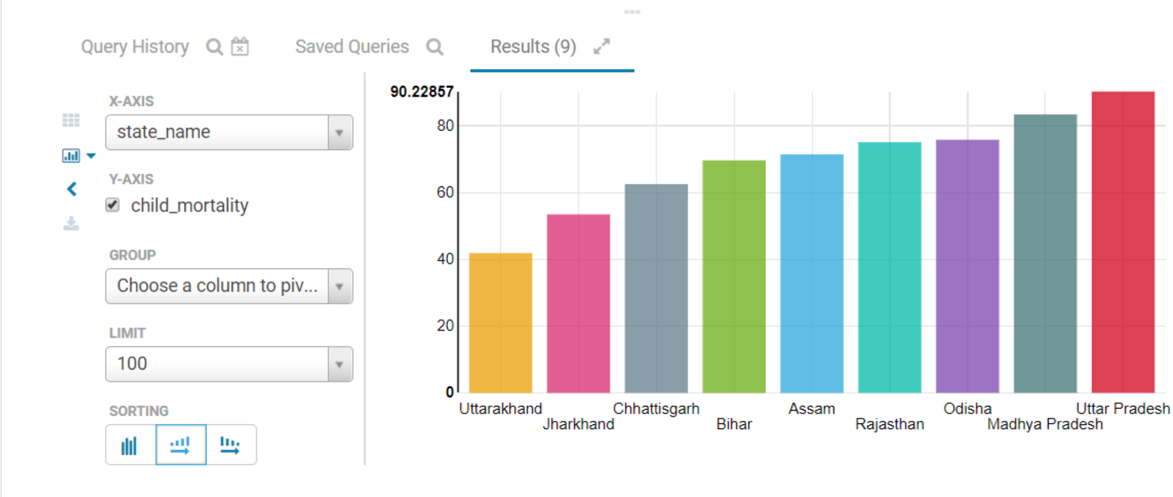
TIME TAKEN – 1min 1sec



```
-----------------STATE WISE CHILD MORTALITY RATE----------

select state_name, avg (yy_under_five_mortality_rate_u5mr_total_person) as Child_Mortality from india_ahs_orc_parti
```

	state_name	child_mortality
1	Assam	71.43478260869566
2	Bihar	69.62162162162163
3	Chhattisgarh	62.5
4	Jharkhand	53.44444444444444
5	Madhya Pradesh	83.37777777777778
6	Odisha	75.8
7	Rajasthan	75.0625
8	Uttar Pradesh	90.22857142857143
9	Uttarakhand	41.84615384615385

Chart:



X-AXIS
state_name

Y-AXIS
☑ child_mortality

GROUP
Choose a column to piv...

LIMIT
100

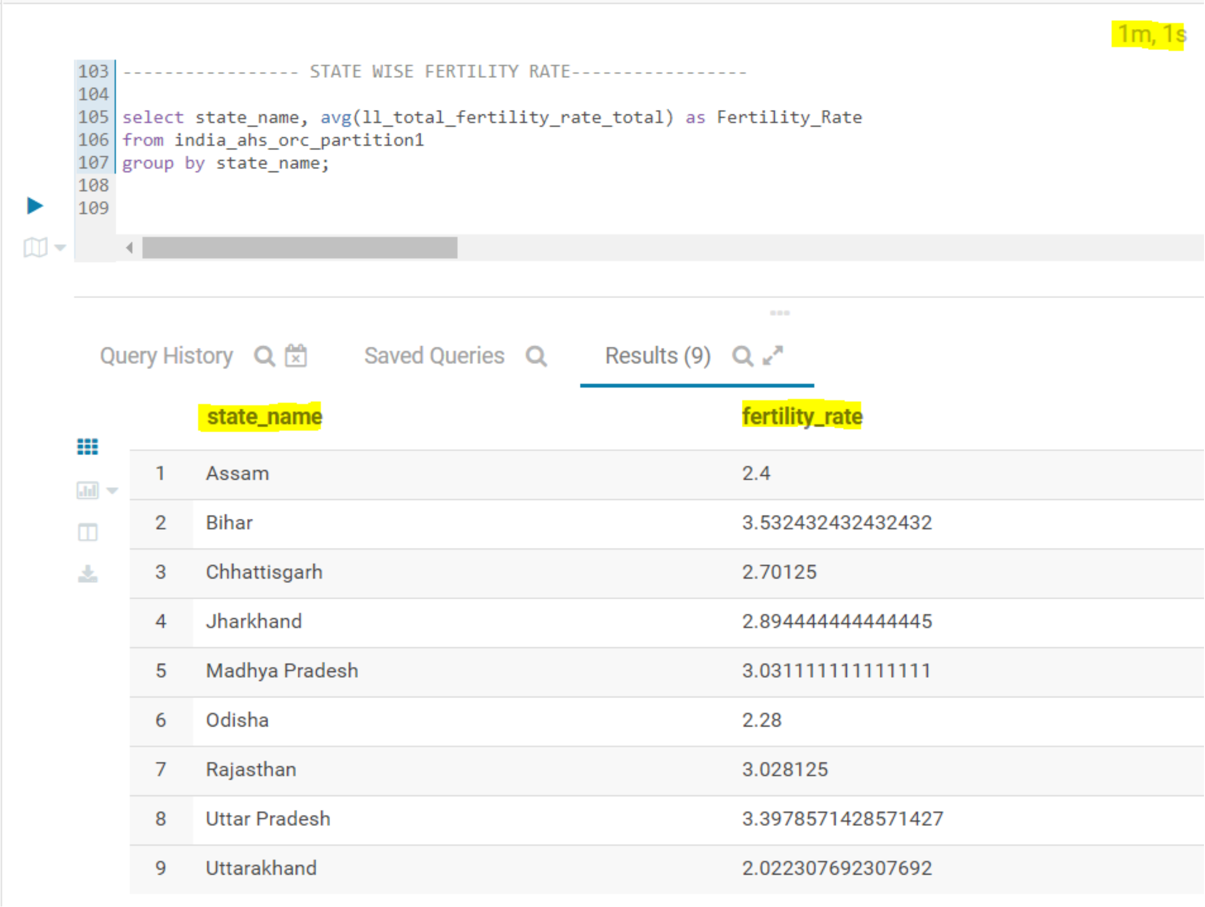
2. STATE WISE FERTILITY RATE

Query:

select state_name, avg(ll_total_fertility_rate_total) as Fertility_Rate from india_ahs_orc_partition1 group by state_name;

Screenshot:

TIME TAKEN – 1min 1sec

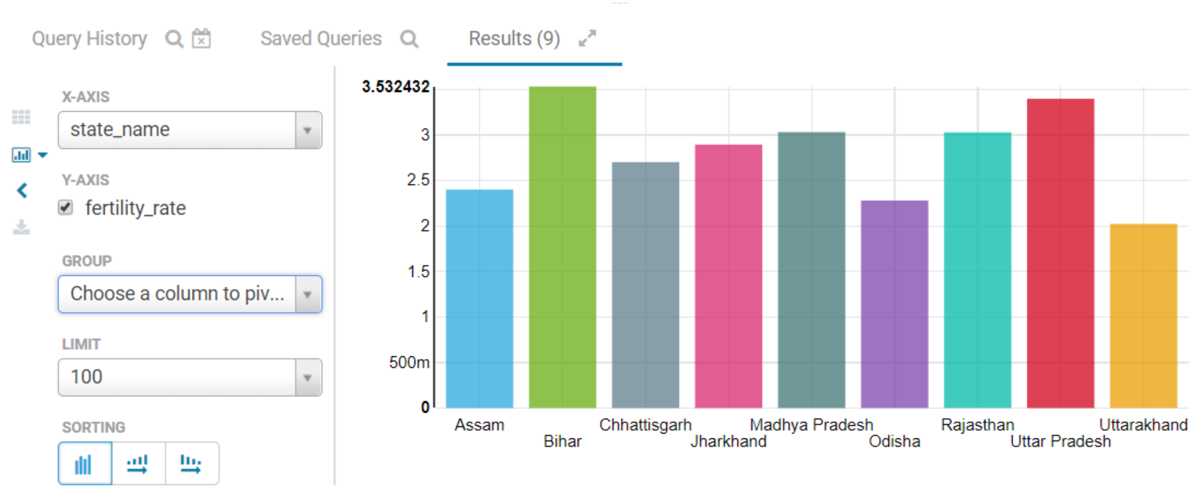


1m, 1s

```
---------------- STATE WISE FERTILITY RATE----------------

select state_name, avg(ll_total_fertility_rate_total) as Fertility_Rate
from india_ahs_orc_partition1
group by state_name;
```

	state_name	fertility_rate
1	Assam	2.4
2	Bihar	3.532432432432432
3	Chhattisgarh	2.70125
4	Jharkhand	2.894444444444445
5	Madhya Pradesh	3.031111111111111
6	Odisha	2.28
7	Rajasthan	3.028125
8	Uttar Pradesh	3.3978571428571427
9	Uttarakhand	2.022307692307692

Chart:



X-AXIS

state_name

Y-AXIS

☑ fertility_rate

GROUP

Choose a column to piv...

LIMIT

100

SORTING

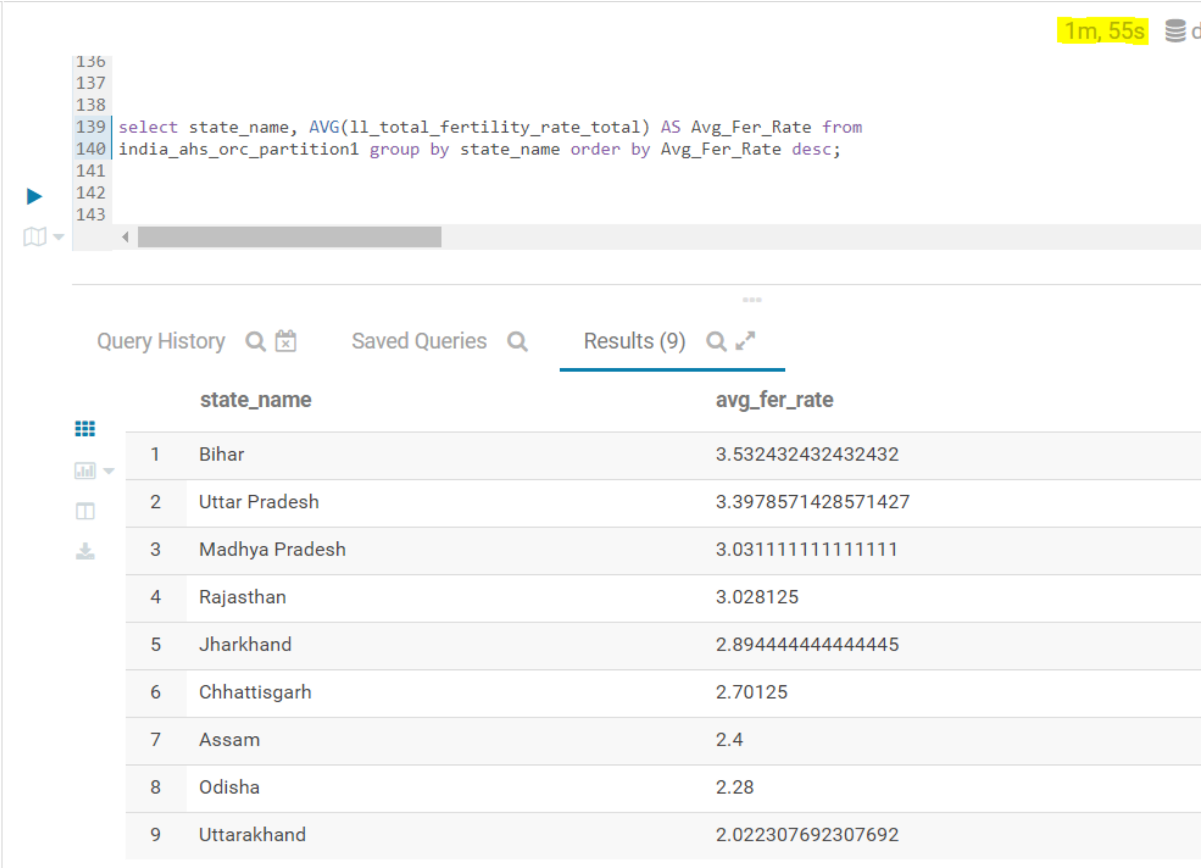
3. DOES HIGH FERTILITY CORRELATE WITH HIGH CHILD MORTALITY?

Query:

select state_name, AVG(ll_total_fertility_rate_total) AS Avg_Fer_Rate from india_ahs_orc_partition1 group by state_name order by Avg_Fer_Rate desc; **Screenshot:**

Screenshot:

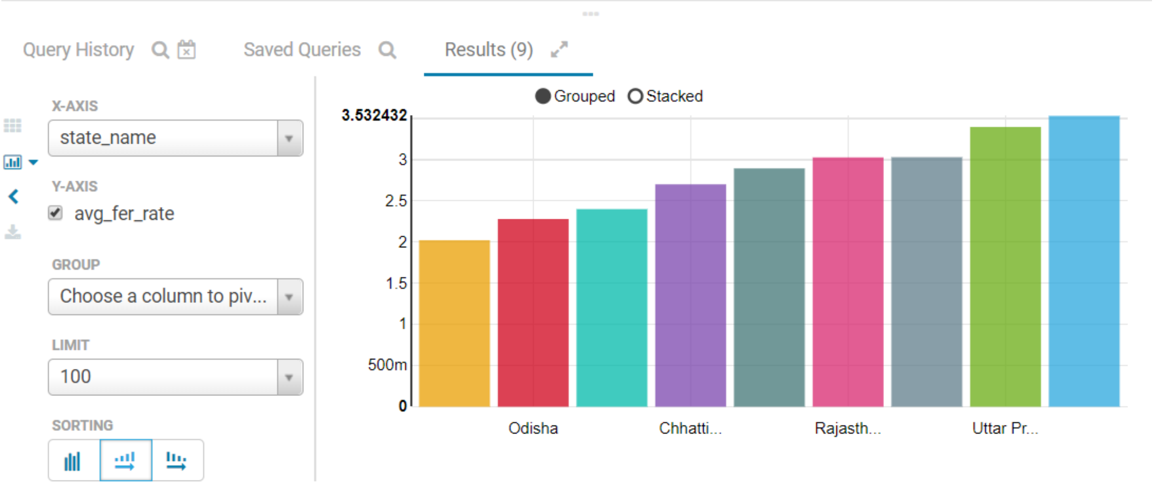
TIME TAKEN-> 1min 55sec



```
select state_name, AVG(ll_total_fertility_rate_total) AS Avg_Fer_Rate from
india_ahs_orc_partition1 group by state_name order by Avg_Fer_Rate desc;
```

	state_name	avg_fer_rate
1	Bihar	3.532432432432432
2	Uttar Pradesh	3.3978571428571427
3	Madhya Pradesh	3.031111111111111
4	Rajasthan	3.028125
5	Jharkhand	2.894444444444445
6	Chhattisgarh	2.70125
7	Assam	2.4
8	Odisha	2.28
9	Uttarakhand	2.022307692307692

Chart:



X-AXIS

state_name

Y-AXIS

☑ avg_fer_rate

GROUP

Choose a column to piv...

LIMIT

100

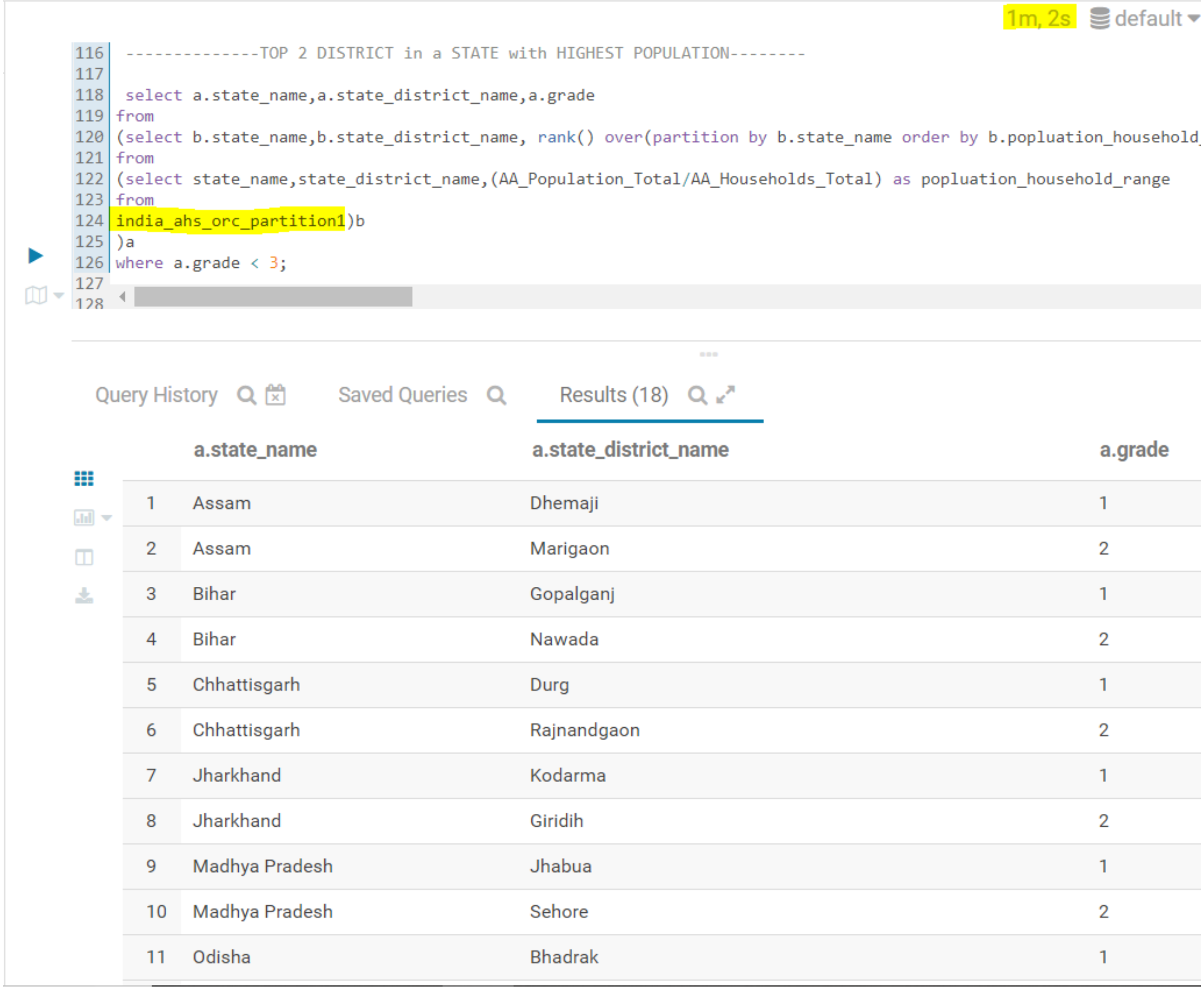
4. Find top 2 districts per state with the highest population per household

Query:

select a.state_name,a.state_district_name,a.grade from (select b.state_name,b.state_district_name, rank() over(partition by b.state_name order by b.popluation_household_range desc) as grade from (select state_name,state_district_name,(AA_Population_Total/AA_Households_Total) as popluation_household_range from india_ahs_orc_partition1)b)a where a.grade < 3;

Screenshot:

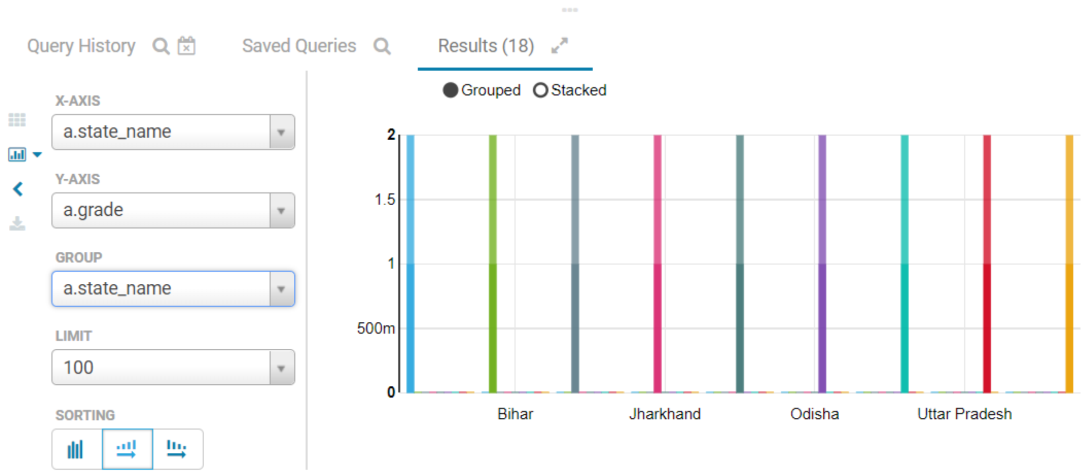
TIME TAKEN -> 1m 2sec



```
 --------------TOP 2 DISTRICT in a STATE with HIGHEST POPULATION--------

 select a.state_name,a.state_district_name,a.grade
from
(select b.state_name,b.state_district_name, rank() over(partition by b.state_name order by b.popluation_household_
from
(select state_name,state_district_name,(AA_Population_Total/AA_Households_Total) as popluation_household_range
from
india_ahs_orc_partition1)b
)a
where a.grade < 3;
```

	a.state_name	a.state_district_name	a.grade
1	Assam	Dhemaji	1
2	Assam	Marigaon	2
3	Bihar	Gopalganj	1
4	Bihar	Nawada	2
5	Chhattisgarh	Durg	1
6	Chhattisgarh	Rajnandgaon	2
7	Jharkhand	Kodarma	1
8	Jharkhand	Giridih	2
9	Madhya Pradesh	Jhabua	1
10	Madhya Pradesh	Sehore	2
11	Odisha	Bhadrak	1

Chart:



X-AXIS
a.state_name

Y-AXIS
a.grade

GROUP
a.state_name

LIMIT
100

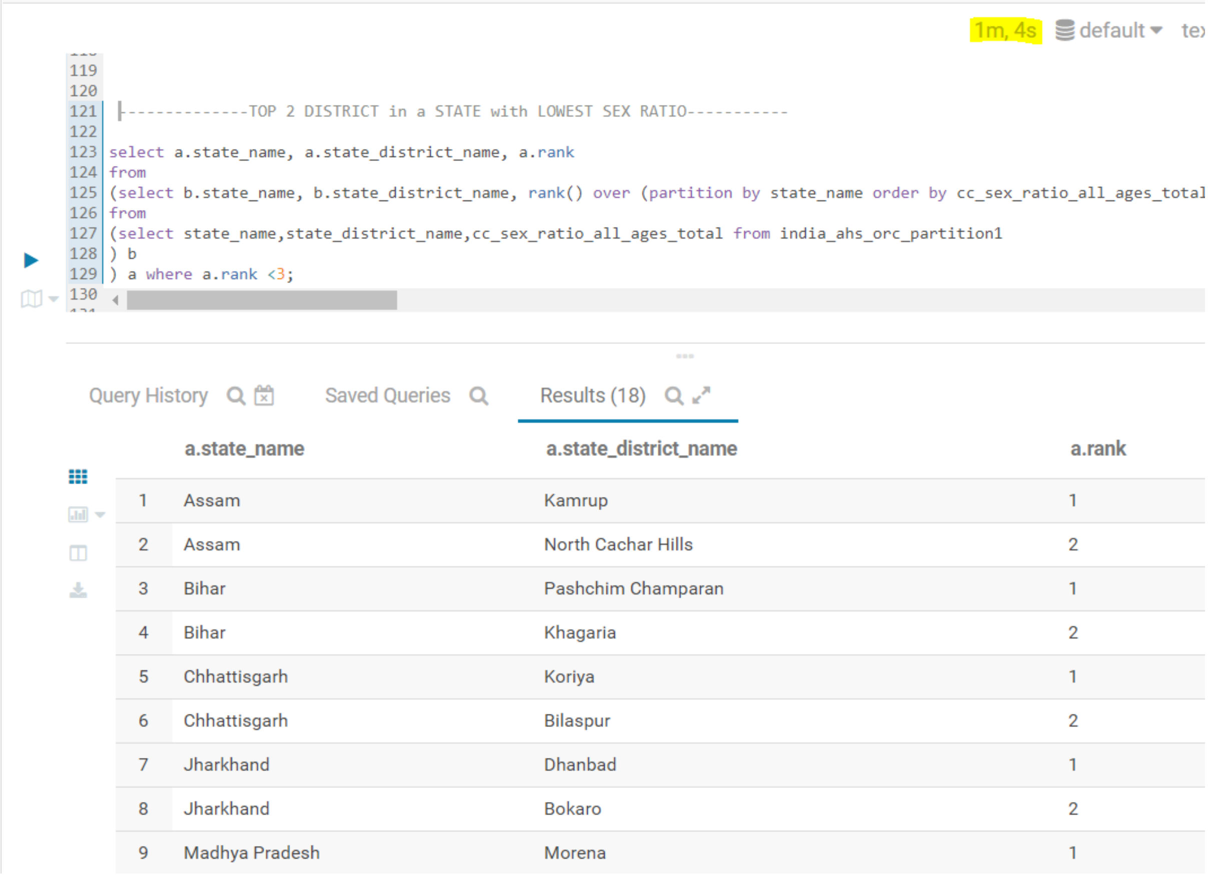
5. Find top 2 districts per state with the lowest sex ratios

<u>Query:</u>

select a.state_name, a.state_district_name, a.rank from (select b.state_name, b.state_district_name, rank() over (partition by state_name order by CC_SEX_RATIO_ALL_AGES_TOTAL() as rank from (select state_name,state_district_name,cc_sex_ratio_all_ages_total from india_ahs_orc_partition1)b)a where a.rank <3;

<u>Screenshot:</u>

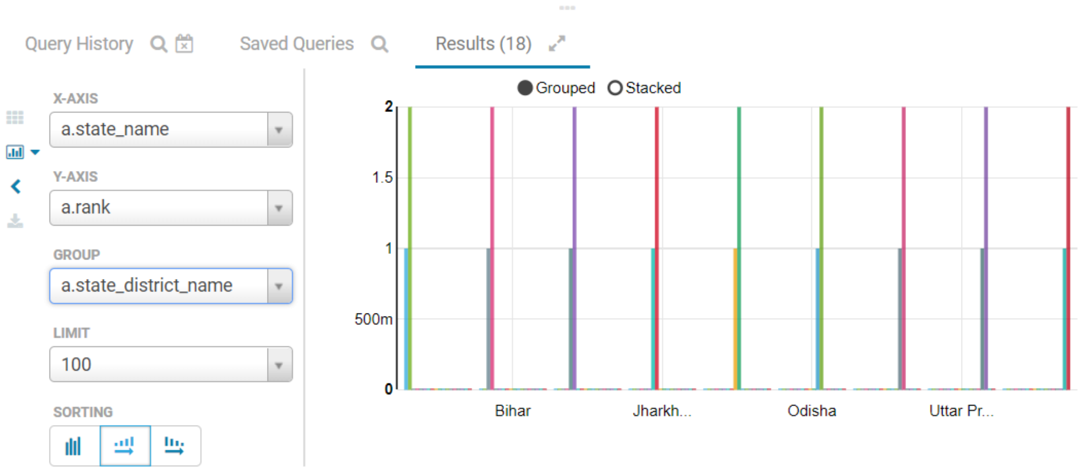
TIME TAKEN -> 1 min 4secs



```
-------------TOP 2 DISTRICT in a STATE with LOWEST SEX RATIO-----------

select a.state_name, a.state_district_name, a.rank
from
(select b.state_name, b.state_district_name, rank() over (partition by state_name order by cc_sex_ratio_all_ages_total
from
(select state_name,state_district_name,cc_sex_ratio_all_ages_total from india_ahs_orc_partition1
) b
) a where a.rank <3;
```

	a.state_name	a.state_district_name	a.rank
1	Assam	Kamrup	1
2	Assam	North Cachar Hills	2
3	Bihar	Paschim Champaran	1
4	Bihar	Khagaria	2
5	Chhattisgarh	Koriya	1
6	Chhattisgarh	Bilaspur	2
7	Jharkhand	Dhanbad	1
8	Jharkhand	Bokaro	2
9	Madhya Pradesh	Morena	1

Chart:



X-AXIS
a.state_name

Y-AXIS
a.rank

GROUP
a.state_district_name

LIMIT
100
