*CREATE TABLE users*

*(*

*id INT,*

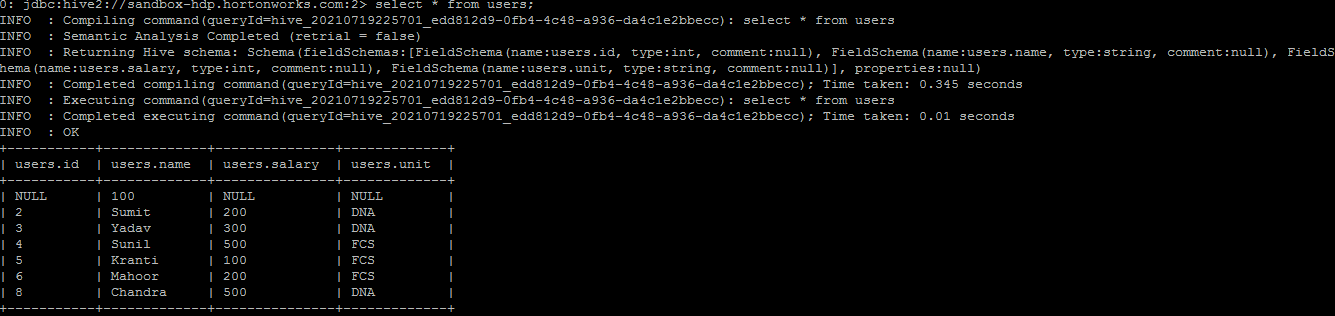
*name STRING,*

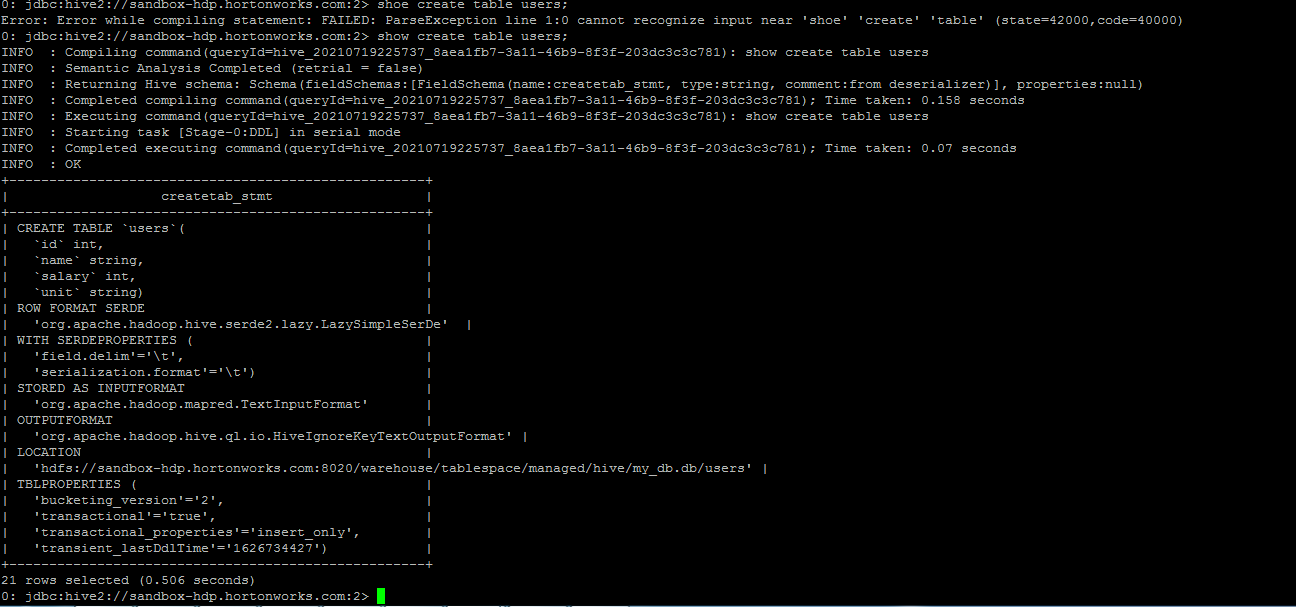
*salary INT,*

*unit STRING*

*) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as textfile ;*

*LOAD DATA LOCAL INPATH '/tmp/hive\_data/users.txt' INTO TABLE users;*





*CREATE TABLE locations*

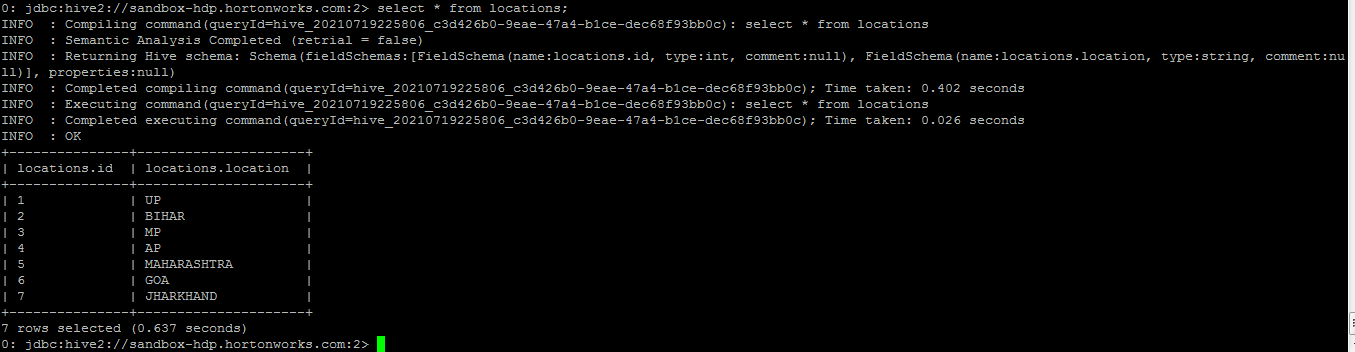
*(*

*id INT,*

*location STRING*

*) ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' stored as textfile ;*

*LOAD DATA LOCAL INPATH '/tmp/hive\_data/locations.txt' INTO TABLE locations;*



*CREATE TABLE buck\_users*

*(*

*id INT,*

*name STRING,*

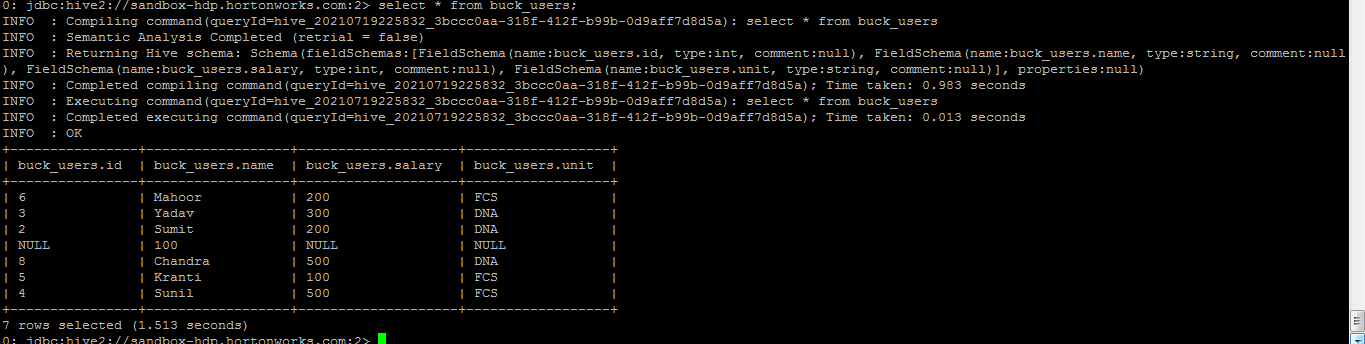
*salary INT,*

*unit STRING*

*) CLUSTERED BY (id) INTO 2 BUCKETS;*

*INSERT OVERWRITE TABLE buck\_users*

*SELECT \* FROM buck\_users;*



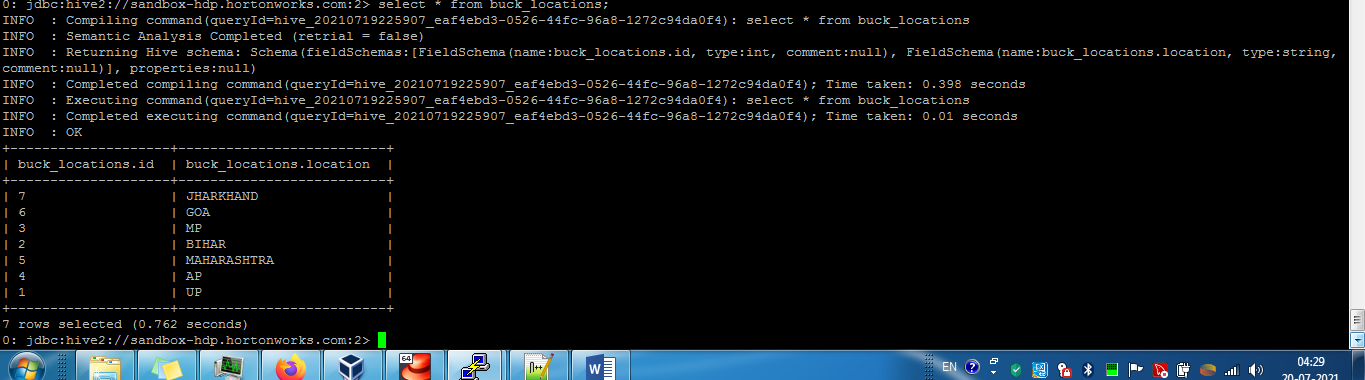
*CREATE TABLE buck\_locations*

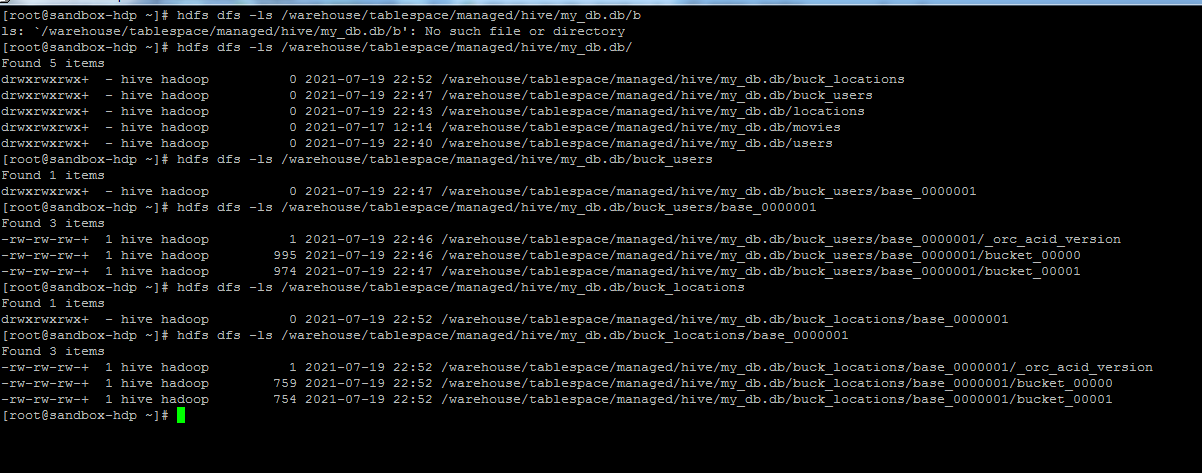
*(*

*id INT,*

*location STRING*

*) CLUSTERED BY (id) INTO 2 BUCKETS;*





-------------------------------------------------------------------------------------------------------

Csvserde.txt

*CREATE TABLE csv\_table(name string, location string)*

*ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'*

*WITH SERDEPROPERTIES (*

*"separatorChar" = ",",*

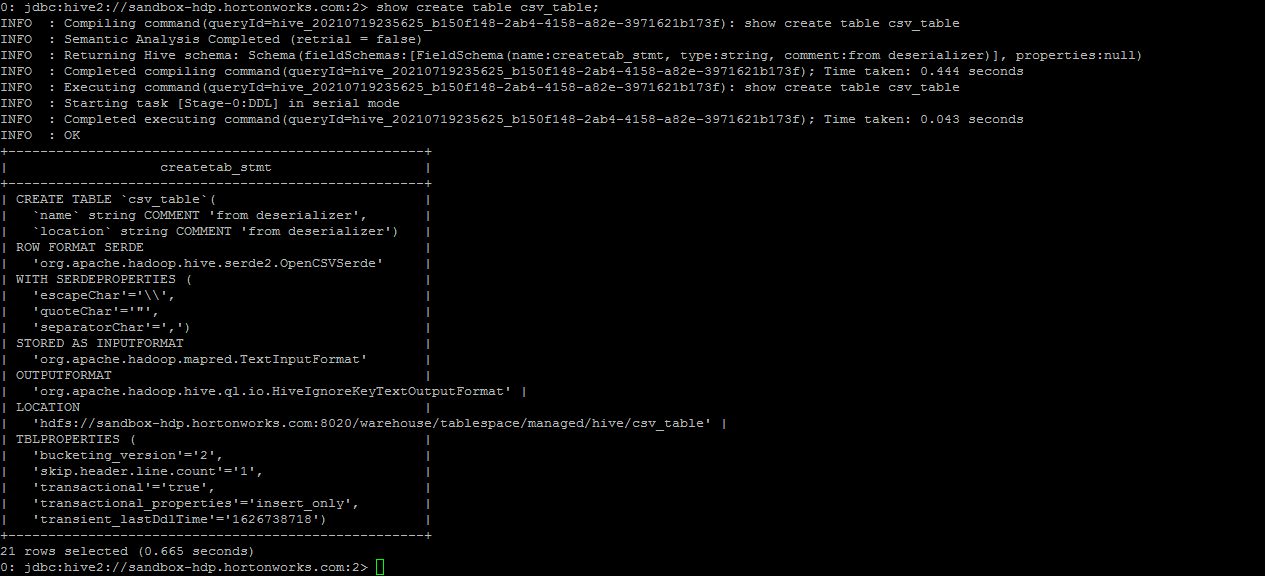
*"quoteChar" = "\"",*

*"escapeChar" = "\\"*

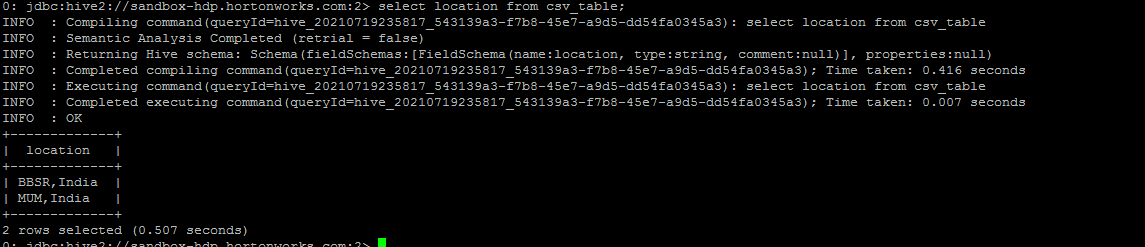
*)*

*STORED AS TEXTFILE*

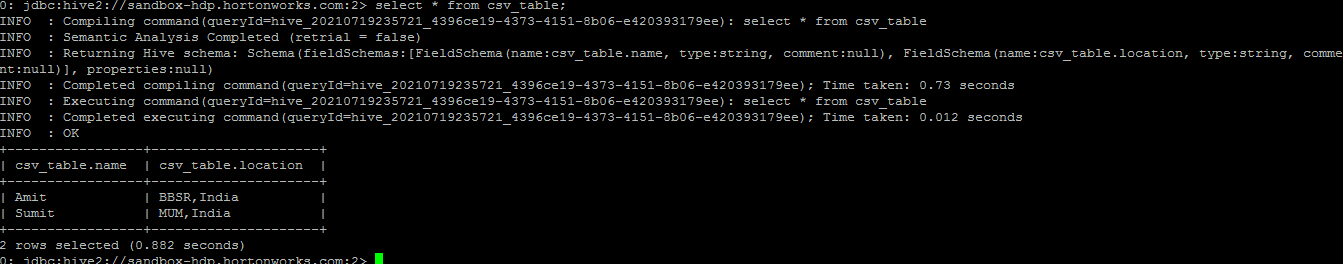
*TBLPROPERTIES ("skip.header.line.count"="1");*

**

*select location from csv\_table;*

**

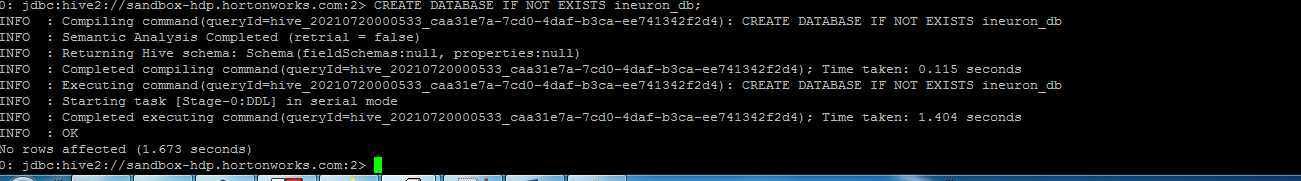
*select \* from csv\_table;*

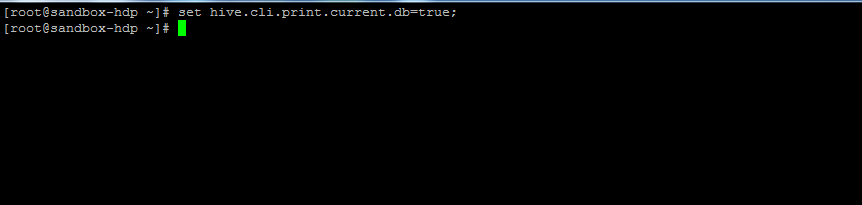


databaseddl

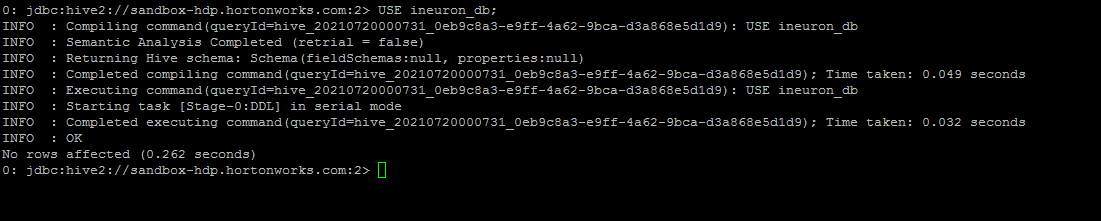
*CREATE DATABASE IF NOT EXISTS ineuron\_db;*

*USE ineuron\_db;*

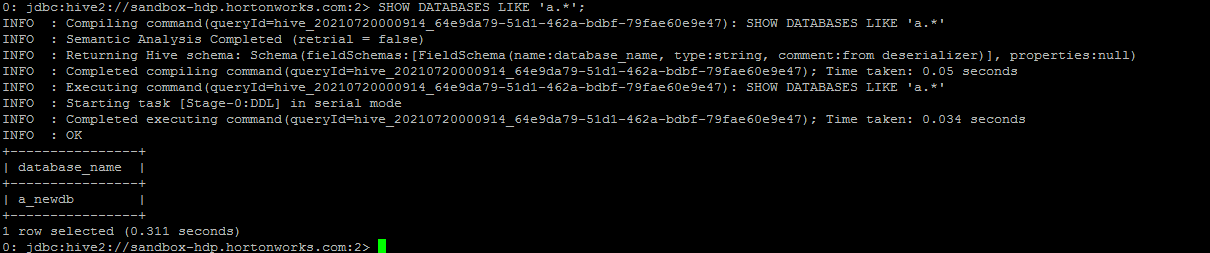
**

**

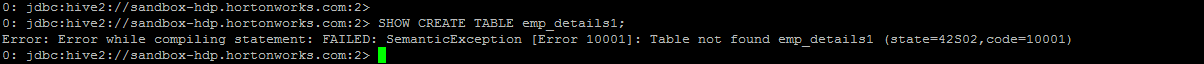
*create database if not exists a\_newdb;*

**

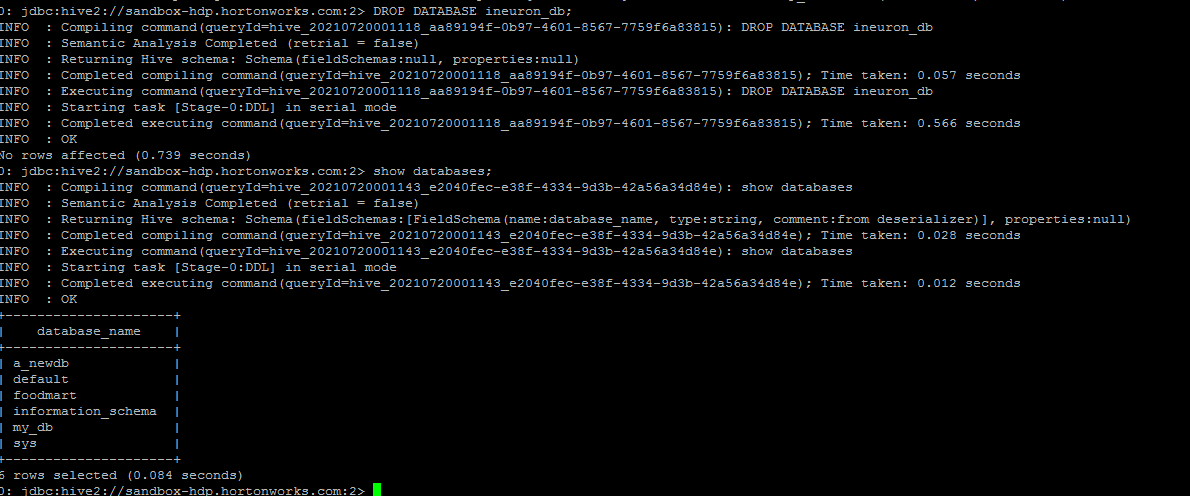
*SHOW DATABASES LIKE 'a.\*';*

**

*SHOW CREATE TABLE emp\_details1;*

**

*DROP DATABASE ineuron\_db;*

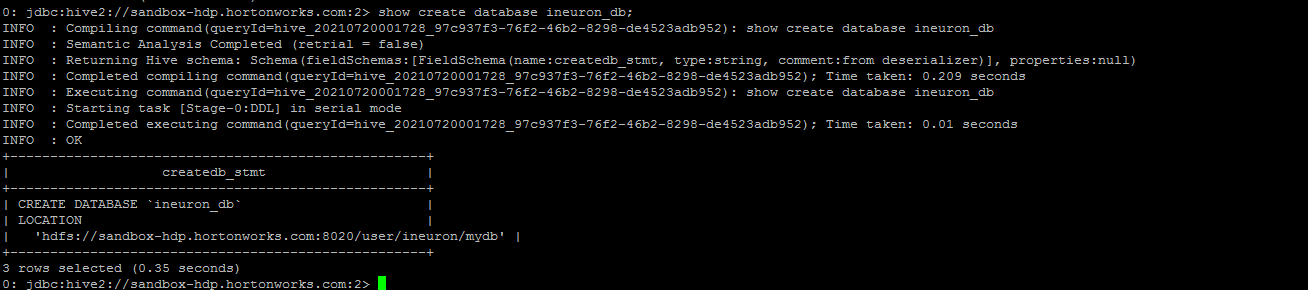
**

*su hdfs*

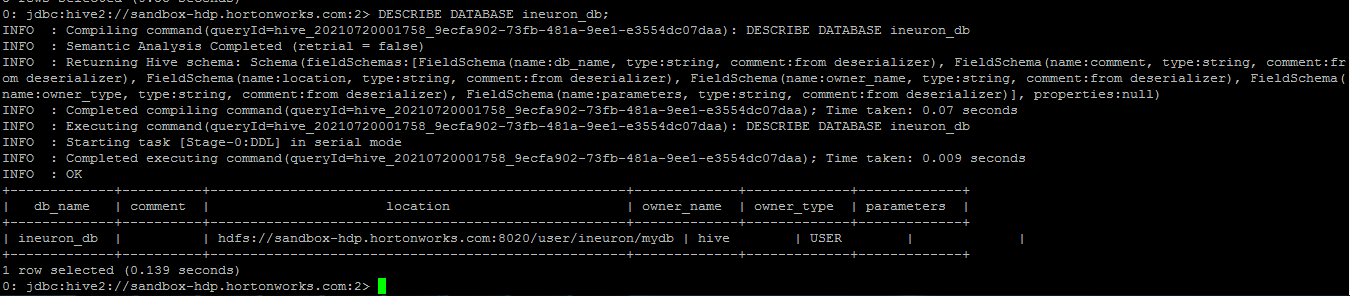
*hdfs dfs -mkdir /user/ineuron*

*hdfs dfs -chmod 777 /user/ineuron*

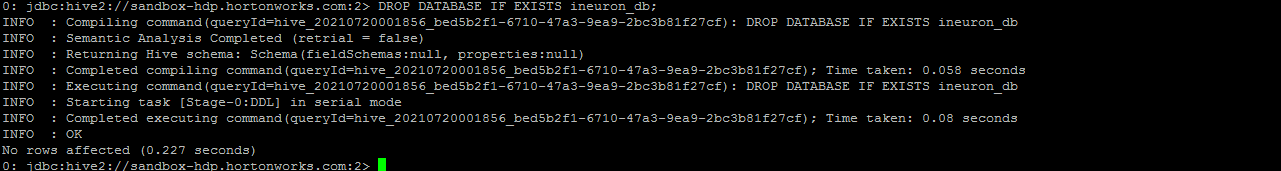
*CREATE DATABASE ineuron\_db LOCATION '/user/ineuron/mydb';*

**

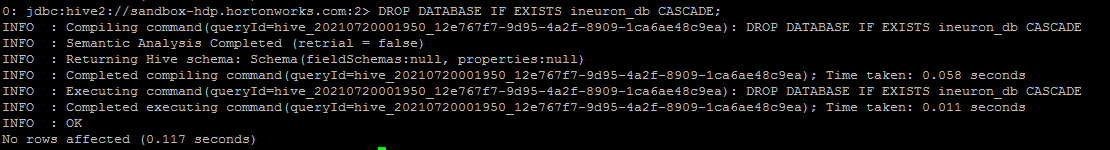
*DESCRIBE DATABASE ineuron\_db;*

**

*DROP DATABASE IF EXISTS ineuron\_db;*

**

*DROP DATABASE IF EXISTS ineuron\_db CASCADE;*

**

*create table emp\_details1*

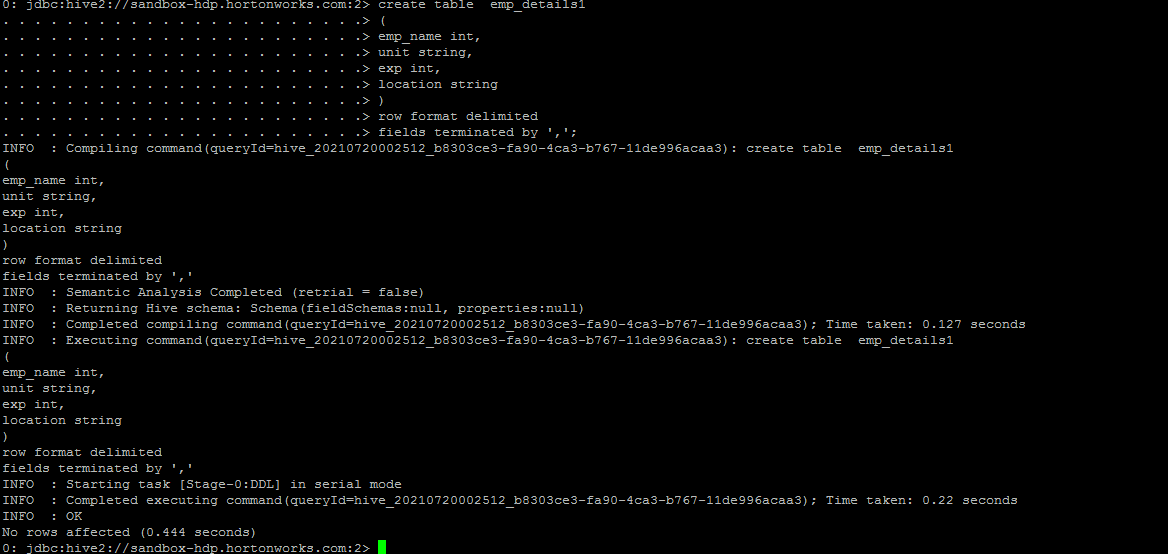
*(*

*emp\_name int,unit string,exp int,location string*

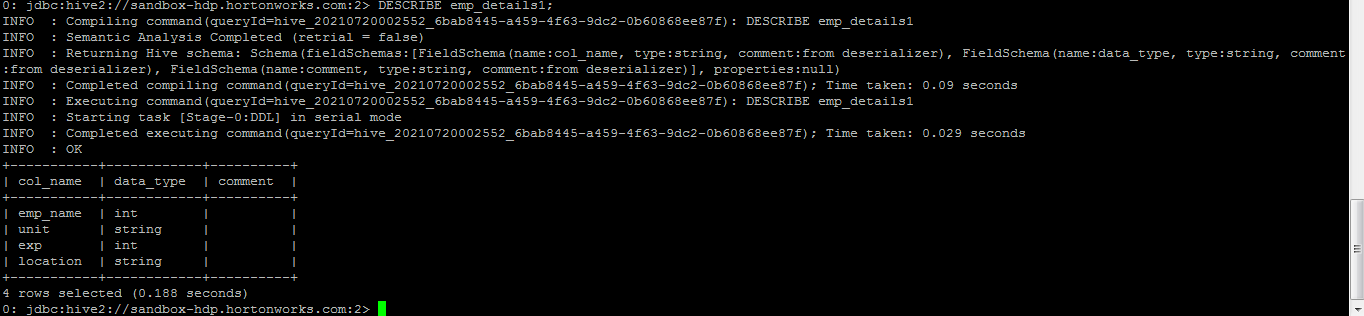
*)*

*row format delimited*

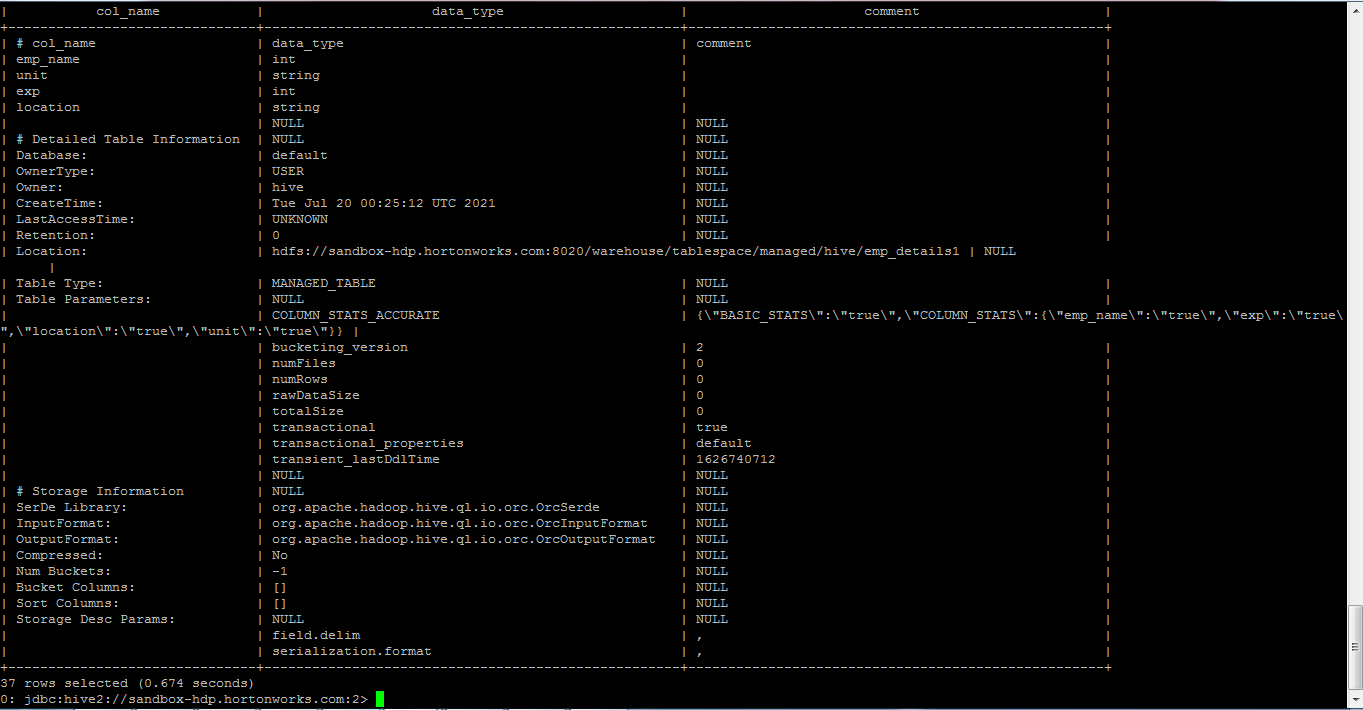
*fields terminated by ',';*



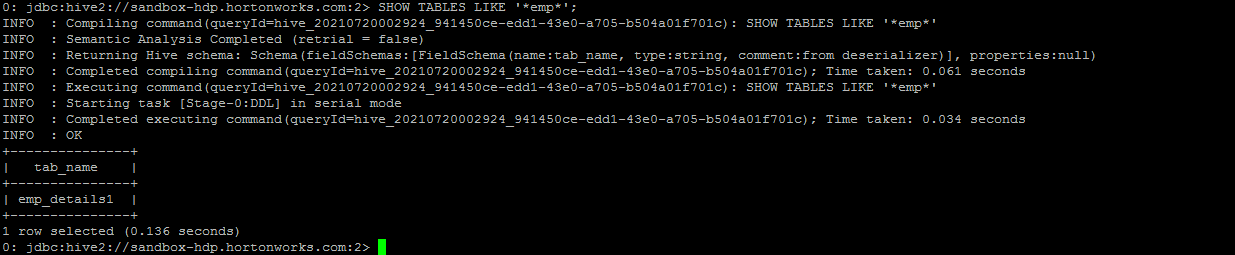
*DESCRIBE emp\_details1;*



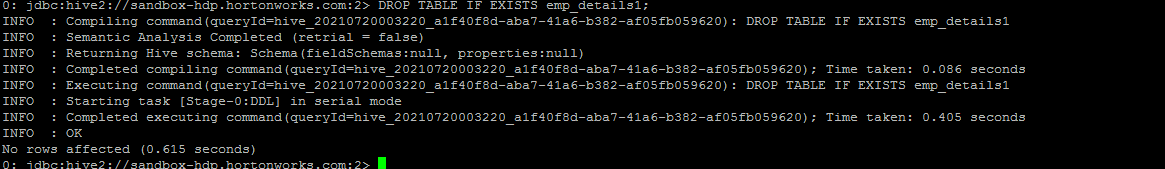
*describe formatted emp\_details1;*



*SHOW TABLES LIKE '\*emp\*';*



*DROP TABLE IF EXISTS emp\_details1;*

**

*create table emp\_details1*

*(*

*emp\_name int,*

*unit string,*

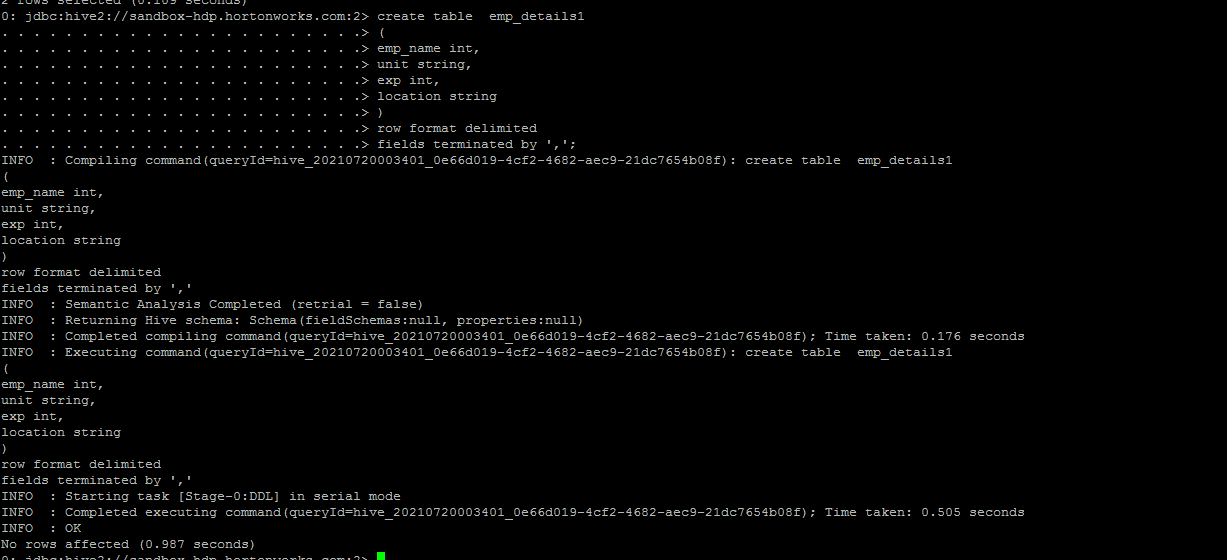
*exp int,*

*location string*

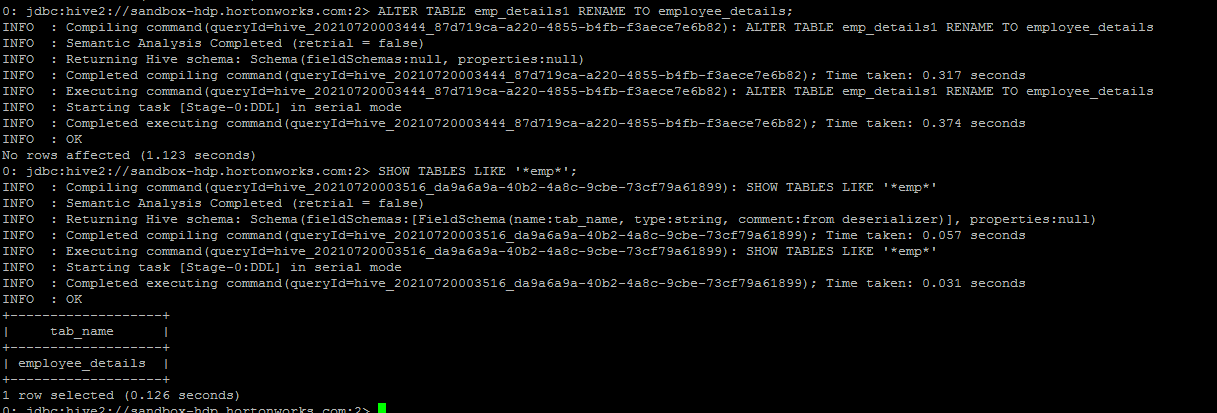
*)*

*row format delimited*

*fields terminated by ',';*

**

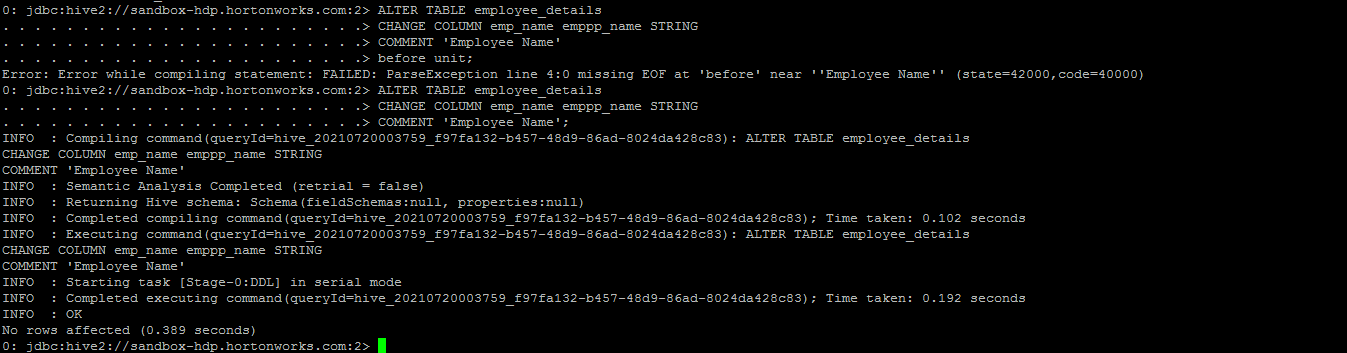
*ALTER TABLE emp\_details1 RENAME TO employee\_details;*

**

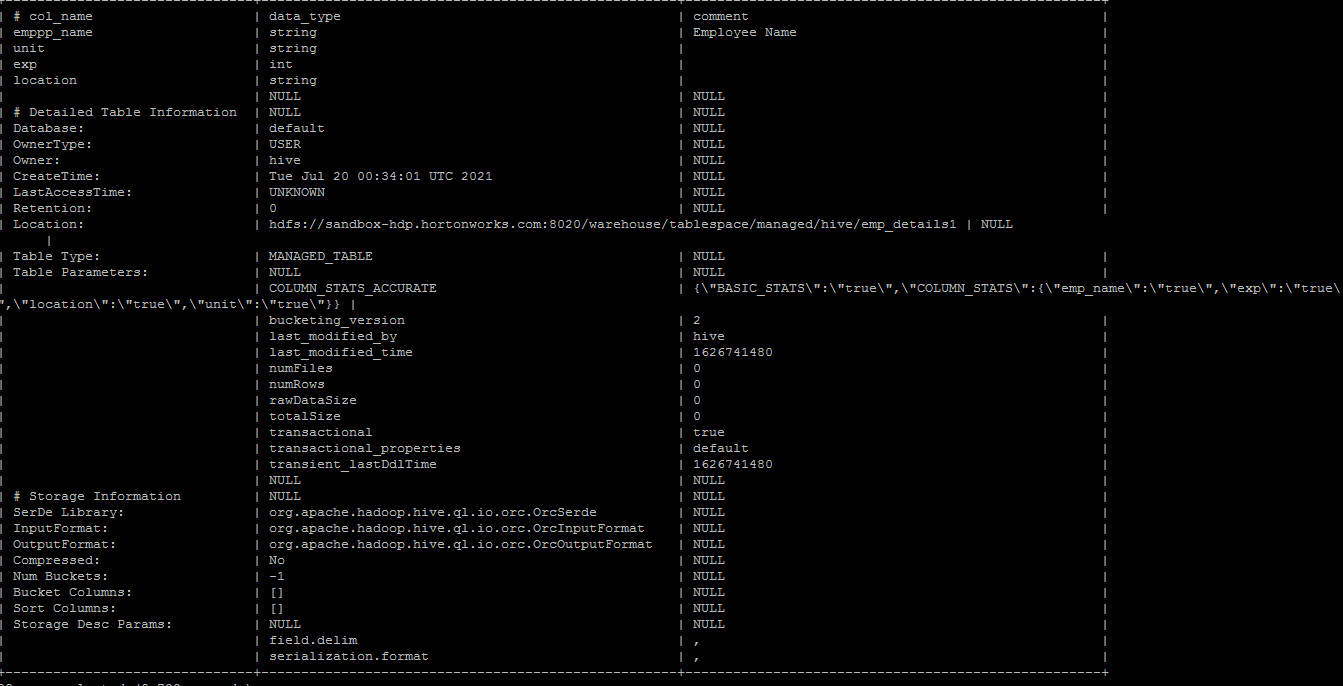
*ALTER TABLE employee\_details*

*CHANGE COLUMN emp\_name emppp\_name STRING*

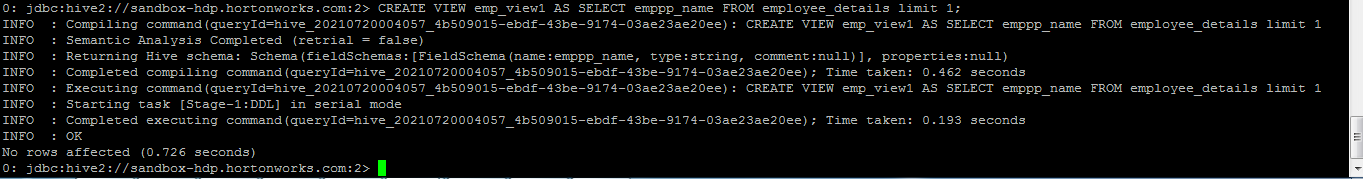
*COMMENT 'Employee Name';*

**

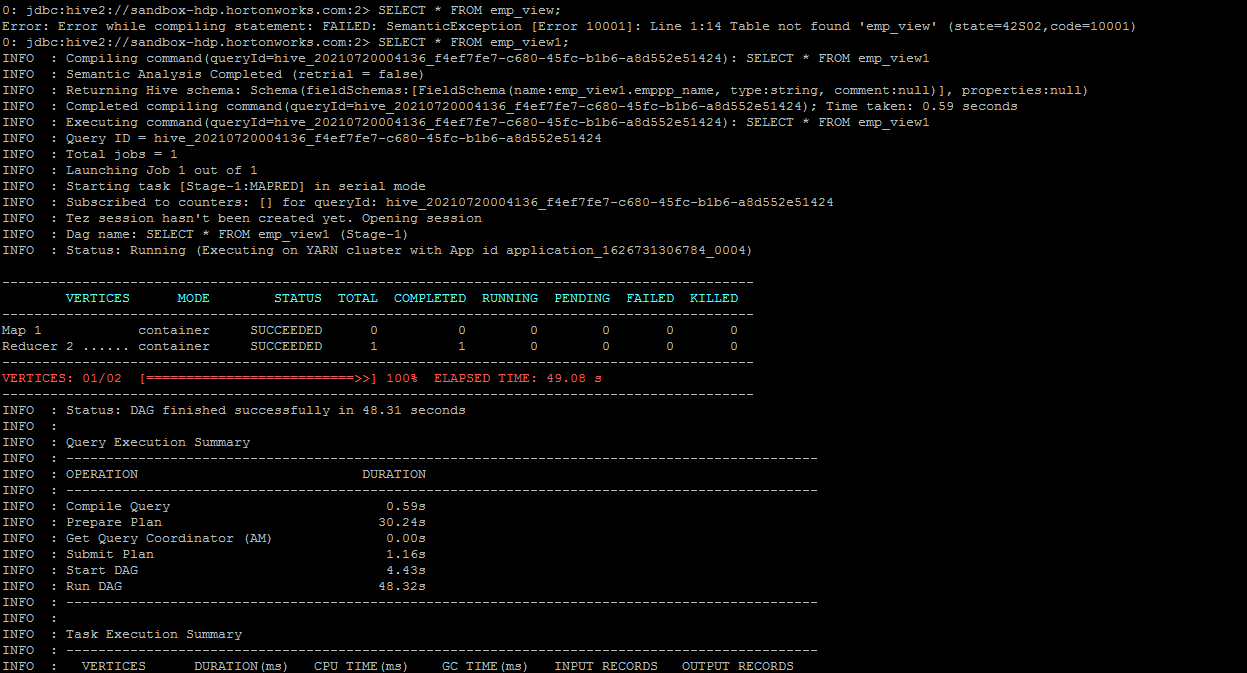
*DESCRIBE FORMATTED emp\_details;*

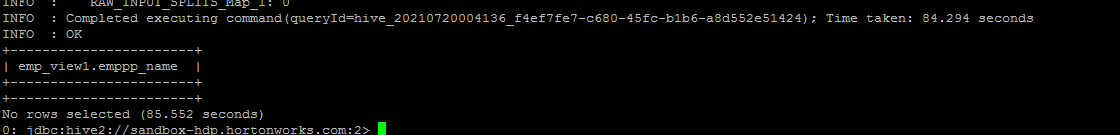
**

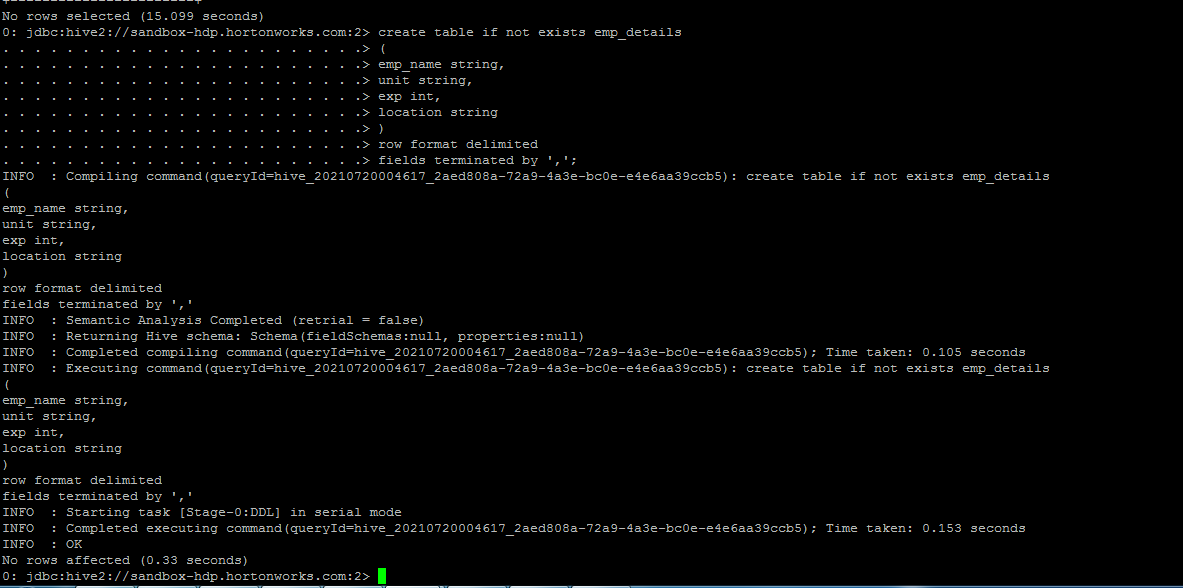
*CREATE VIEW emp\_view1 AS SELECT emppp\_name FROM employee\_details limit 1;*

**

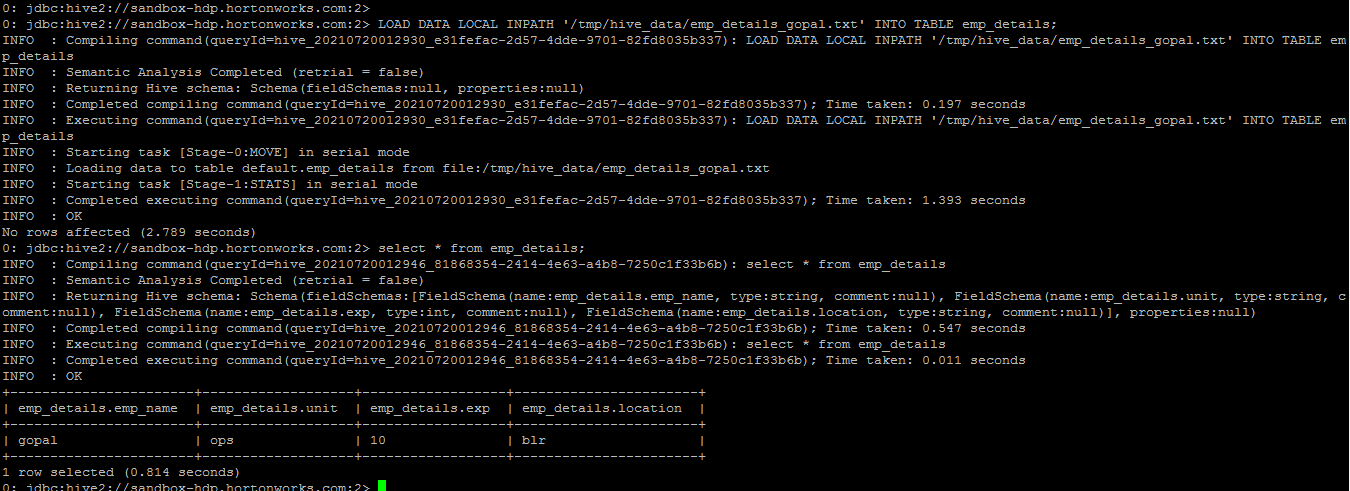
*SELECT \* FROM emp\_view1;*



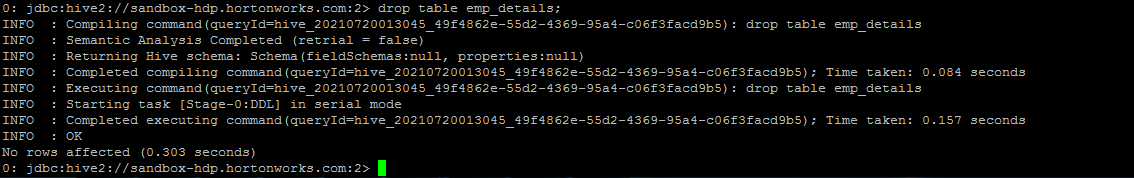




*LOAD DATA LOCAL INPATH '/tmp/hive\_data/emp\_details\_gopal.txt' INTO TABLE emp\_details;*



*drop table emp\_details;*

**

*create table emp\_details2*

*(*

*emp\_name string,*

*unit string,*

*exp int,*

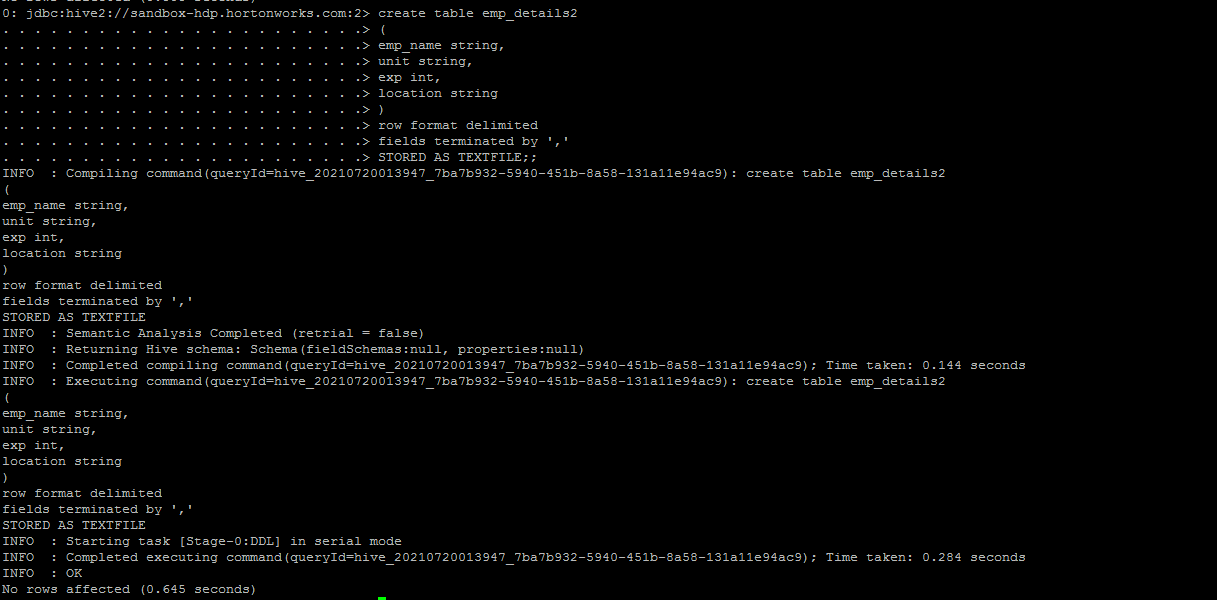
*location string*

*)*

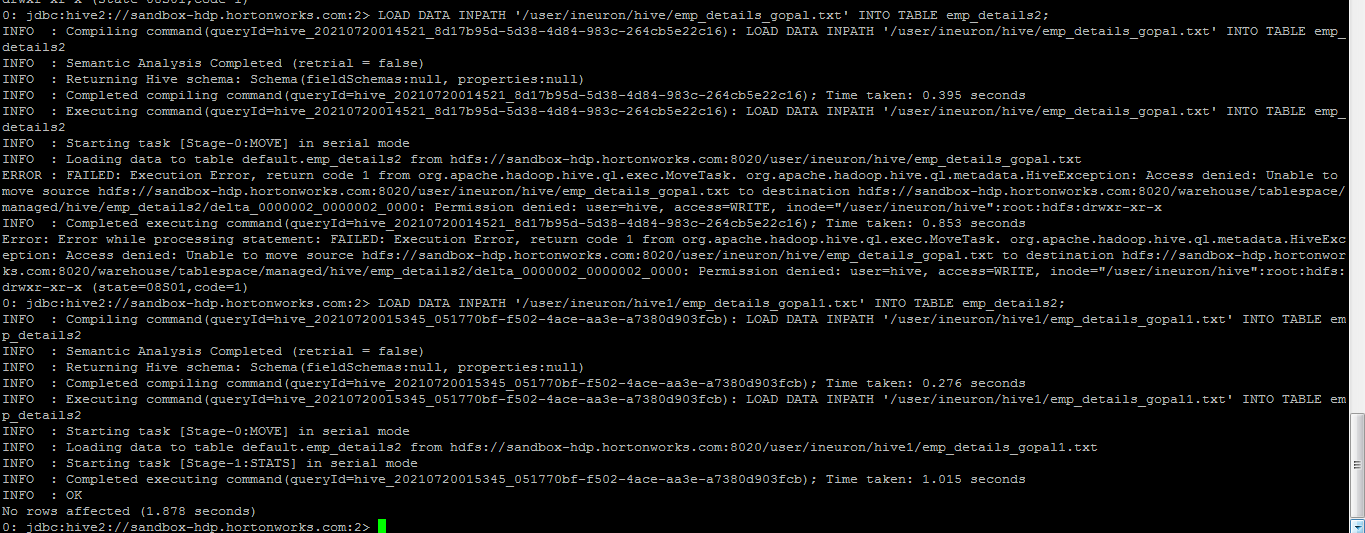
*row format delimited*

*fields terminated by ','*

*STORED AS TEXTFILE;*

**

*LOAD DATA INPATH '/user/ineuron/hive1/emp\_details\_gopal1.txt' INTO TABLE emp\_details2;*

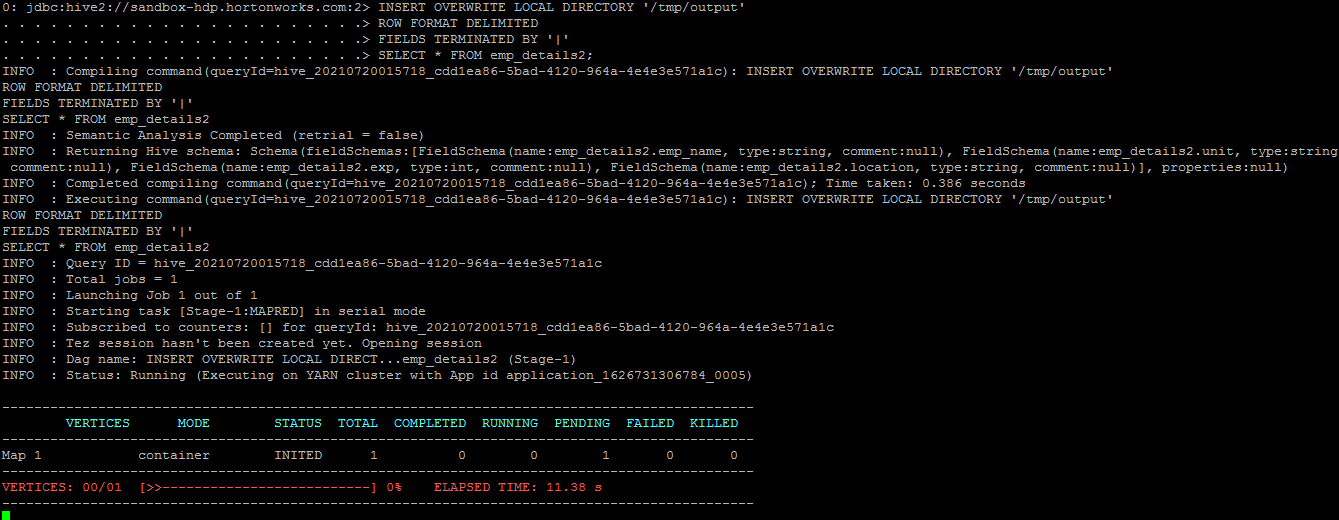
**

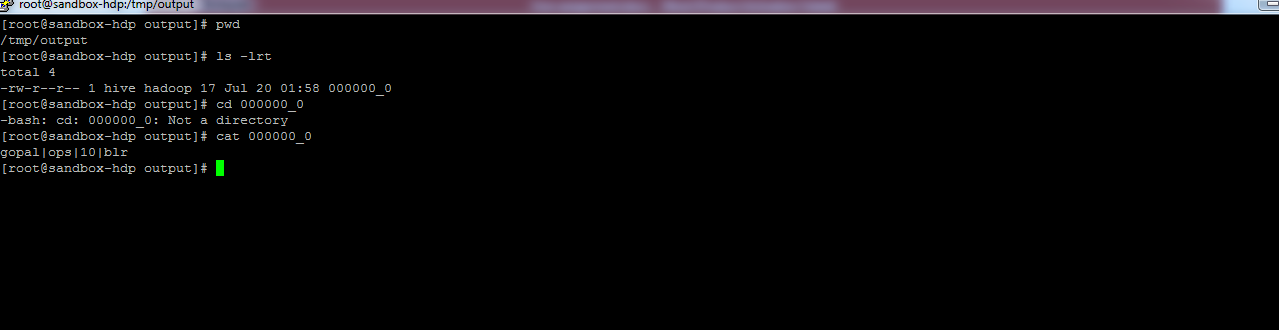
*INSERT OVERWRITE LOCAL DIRECTORY '/tmp/output'*

*ROW FORMAT DELIMITED*

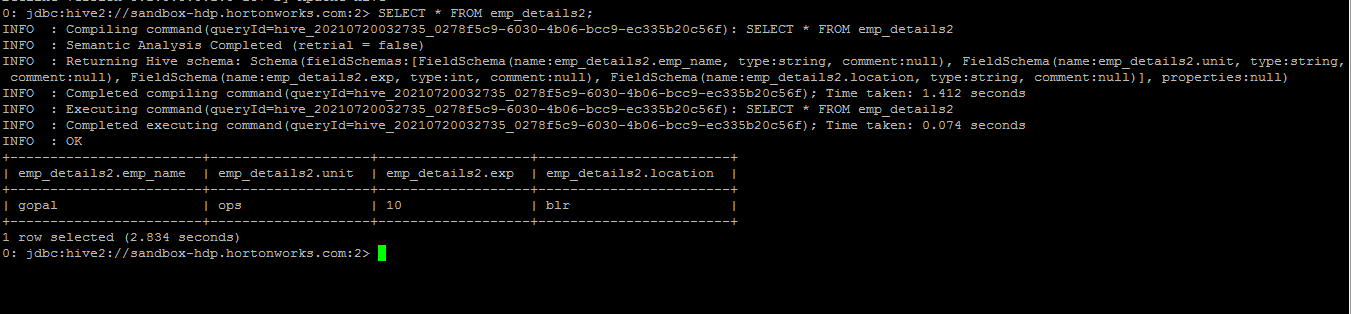
*FIELDS TERMINATED BY '|'*

*SELECT \* FROM emp\_details2;*

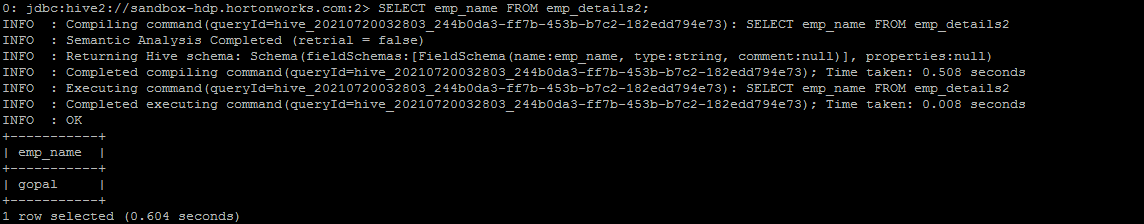
**

**

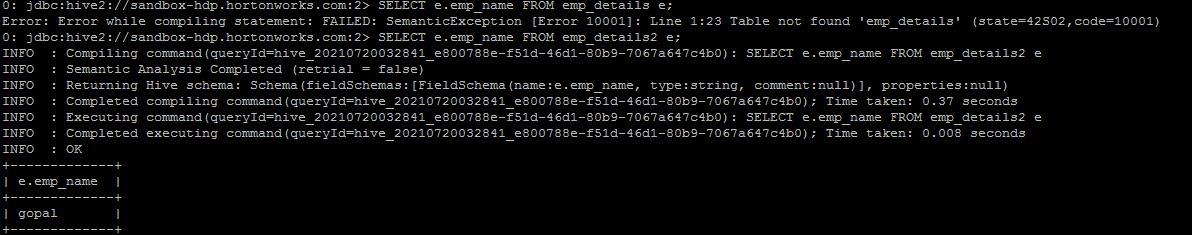
*SELECT \* FROM emp\_details2;*

**

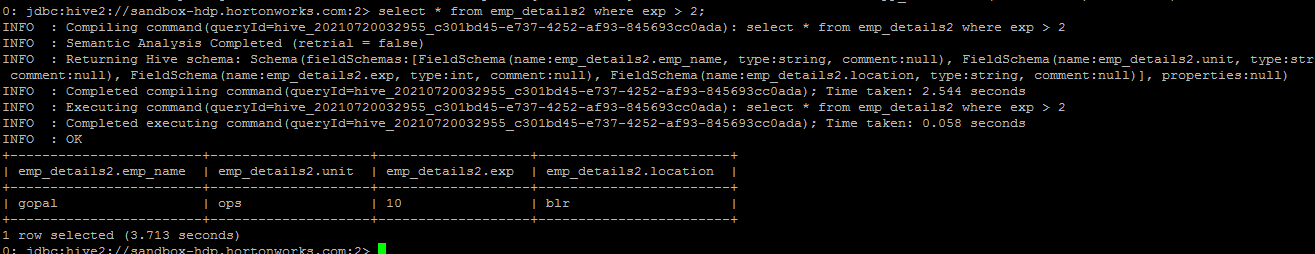
*SELECT emp\_name FROM emp\_details2;*

**

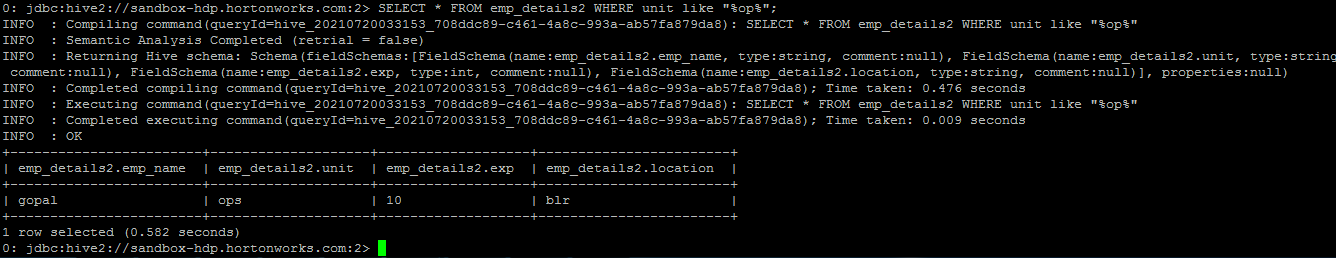
*SELECT e.emp\_name FROM emp\_details2 e;*

**

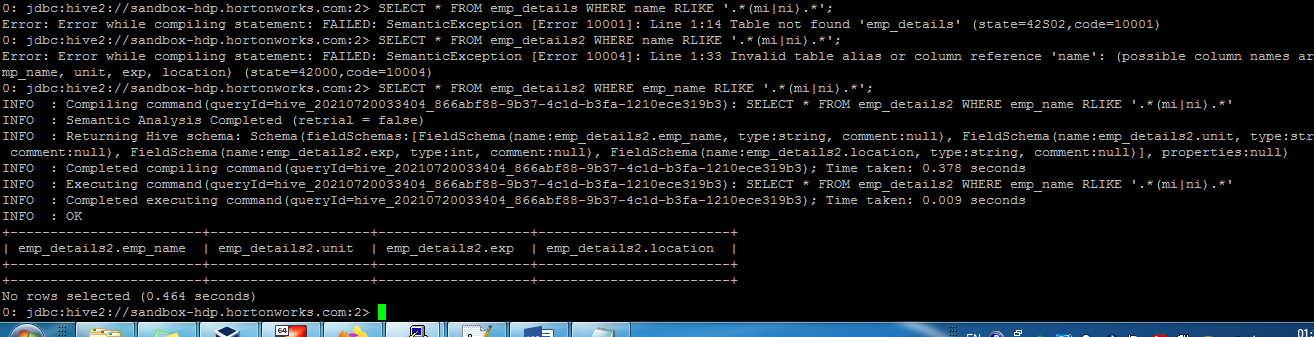
*select \* from empp\_details2 where exp > 2;*

**

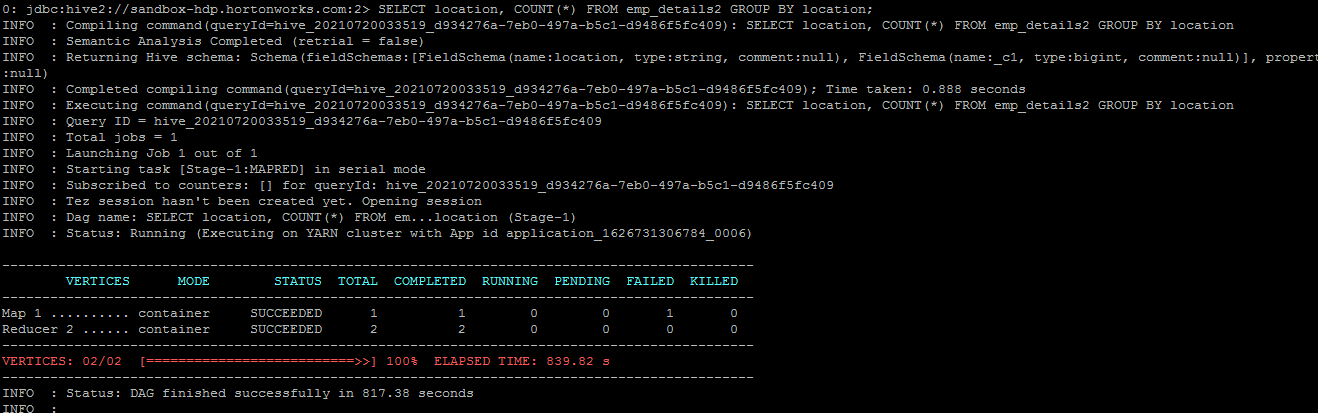
*SELECT \* FROM emp\_details2 WHERE unit like "%op%";*

**

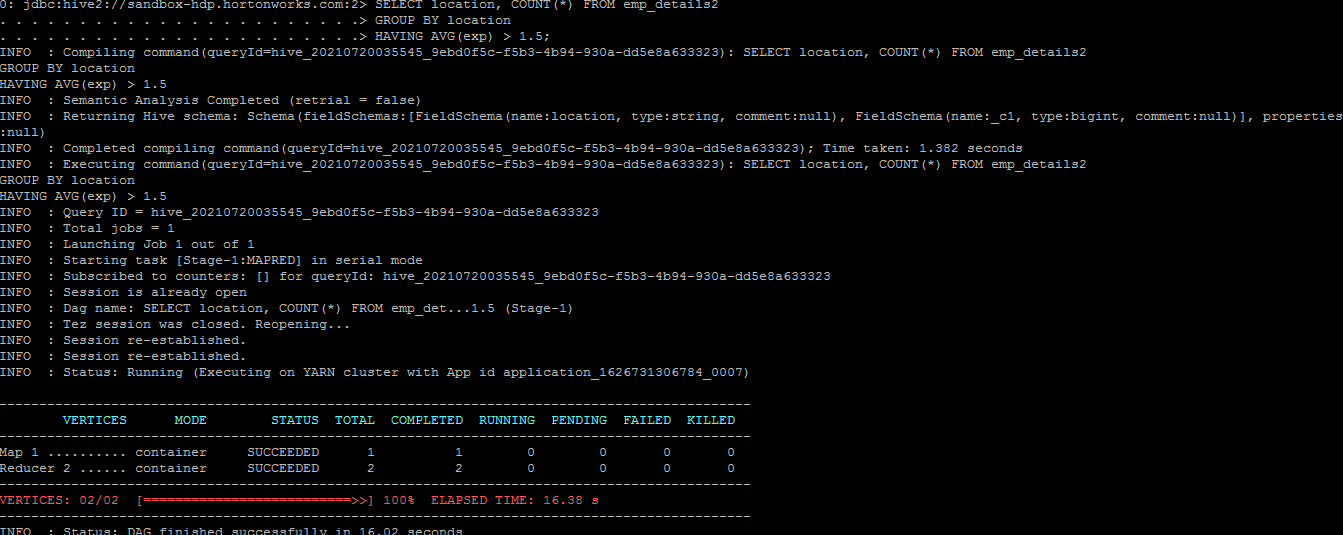
*SELECT \* FROM emp\_details2 WHERE emp\_name RLIKE '.\*(mi|ni).\*';*

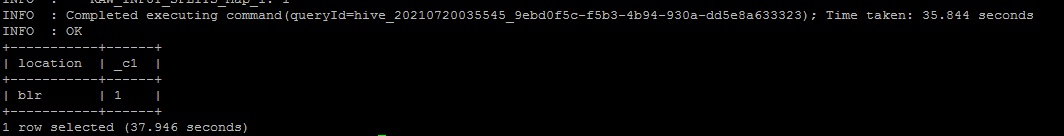
**

*SELECT location, COUNT(\*) FROM emp\_details2 GROUP BY location;*

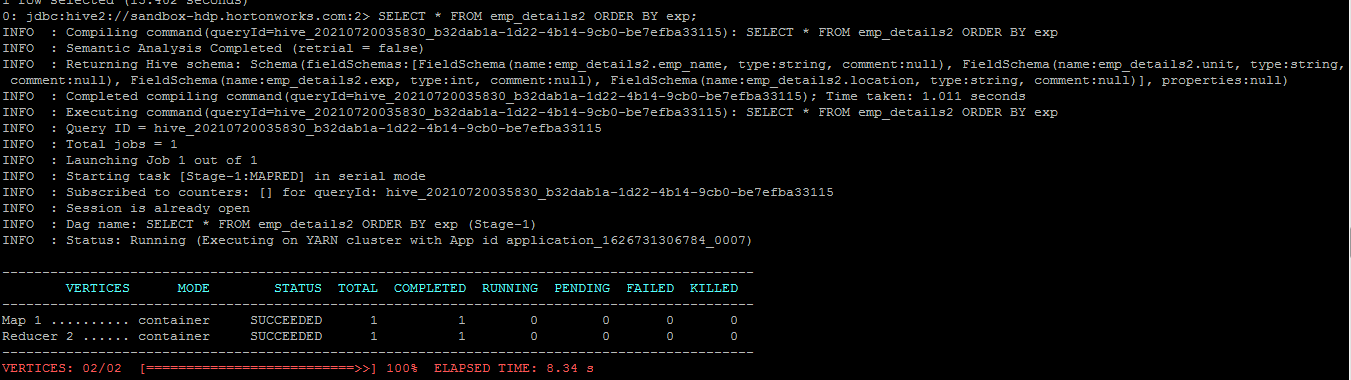
**

*SELECT location, COUNT(\*) FROM emp\_details2 GROUP BY location HAVING AVG(exp) > 1.5;*

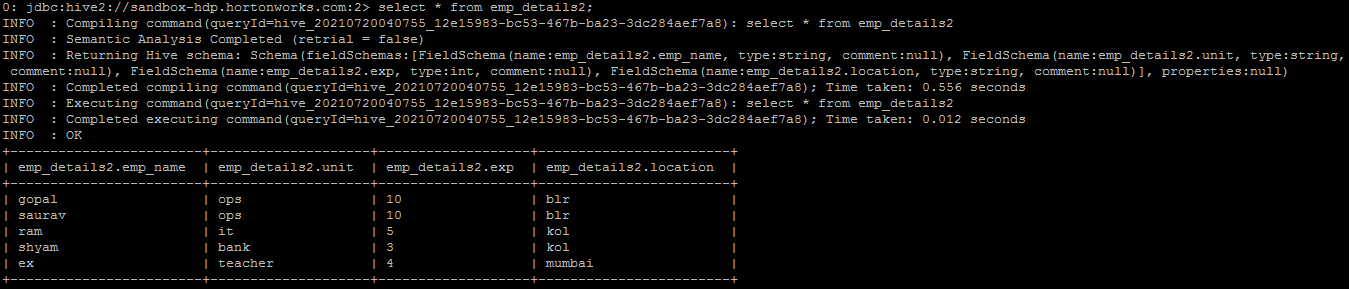
**

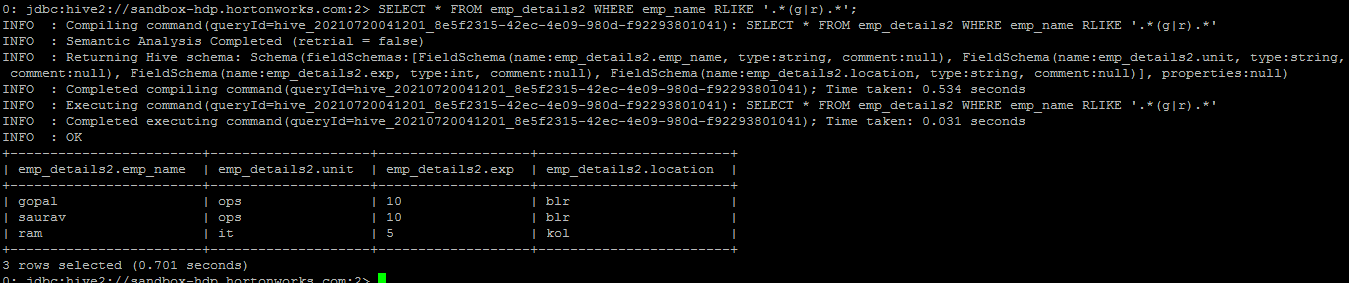
**

*SELECT \* FROM emp\_details2 ORDER BY exp;*

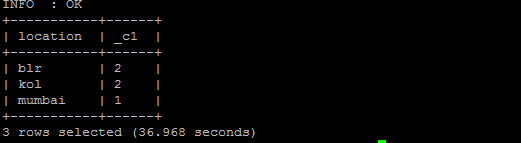
**

*Added few more rows in emp\_details2*

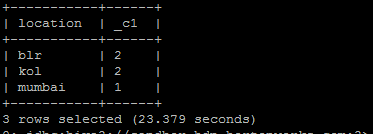
**

**

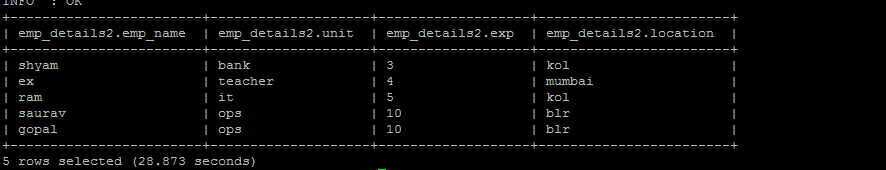
*SELECT location, COUNT(\*) FROM emp\_details2 GROUP BY location;*

**

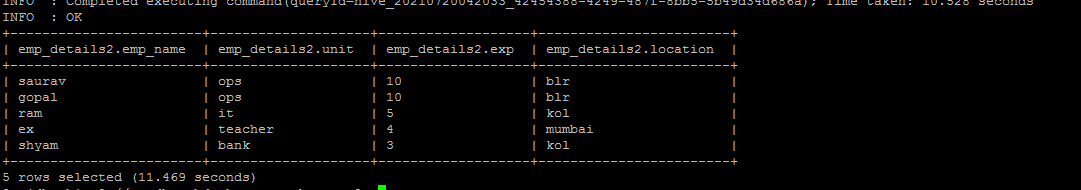
*SELECT location, COUNT(\*) FROM emp\_details2 GROUP BY location HAVING AVG(exp) > 2.5;*

**

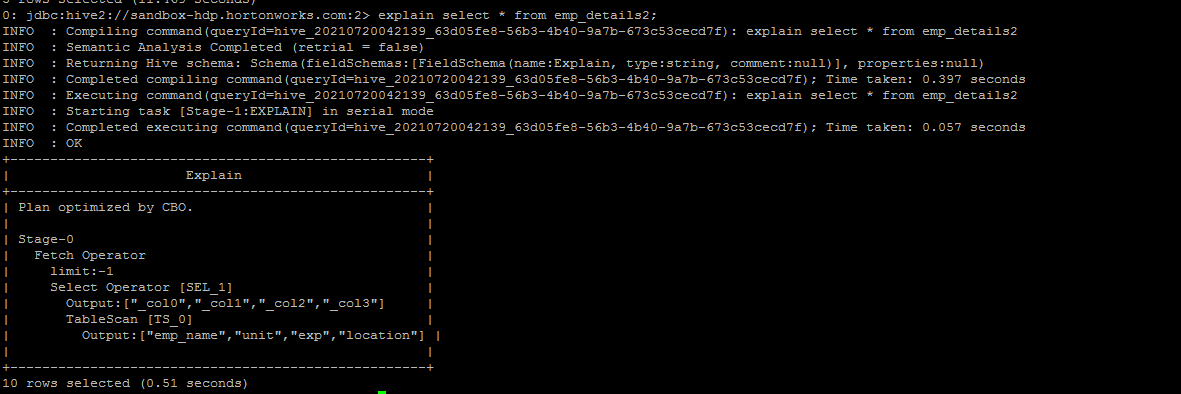
*SELECT \* FROM emp\_details2 ORDER BY exp;*

**

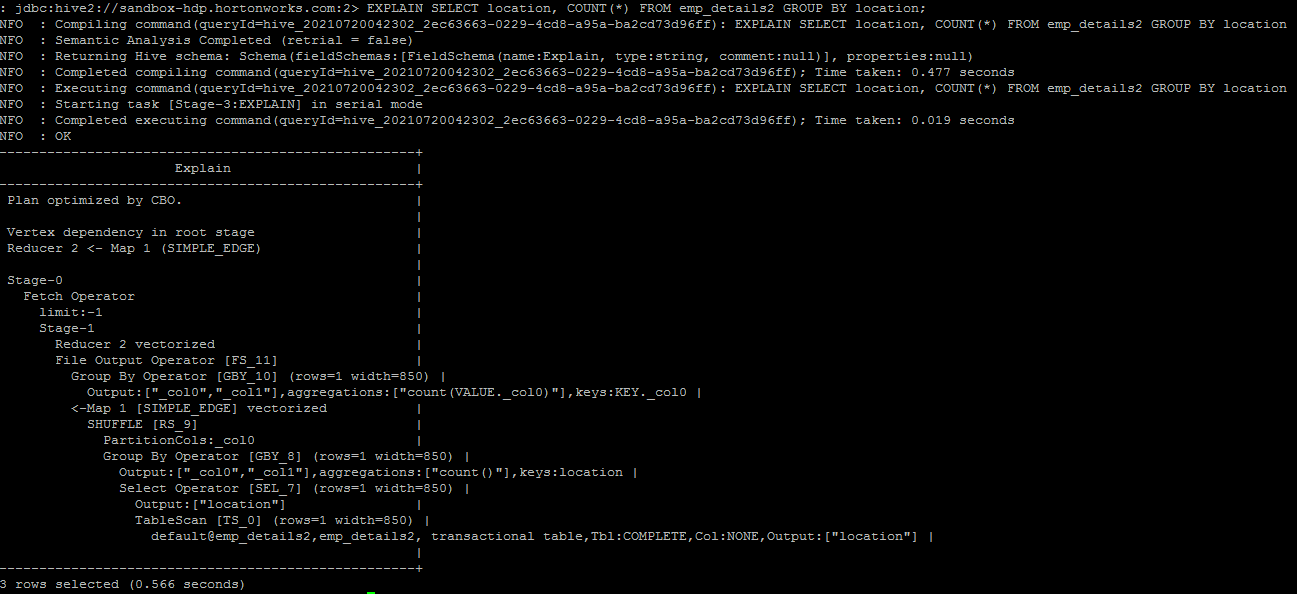
*SELECT \* FROM emp\_details2 ORDER BY exp desc;*

**

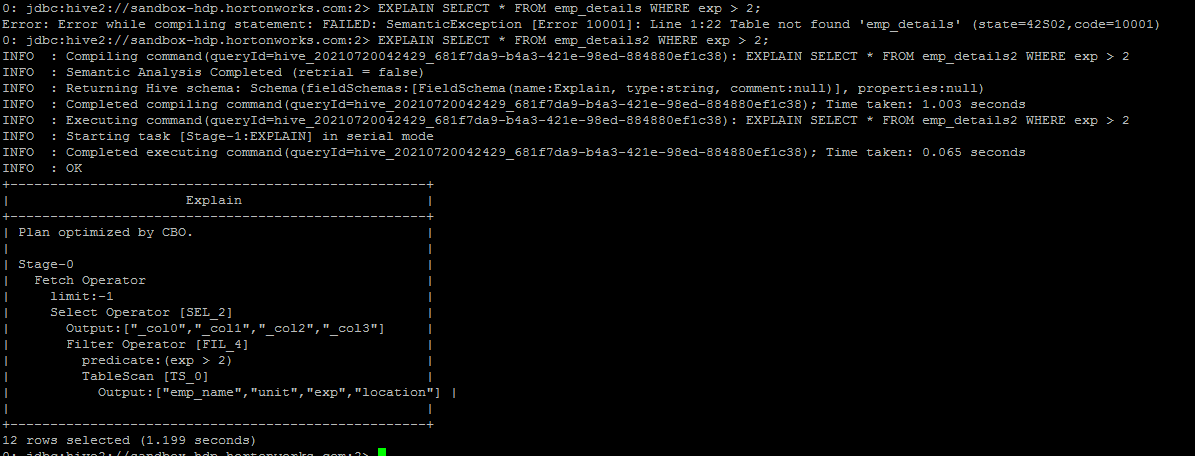
*explain select \* from emp\_details2;*

**

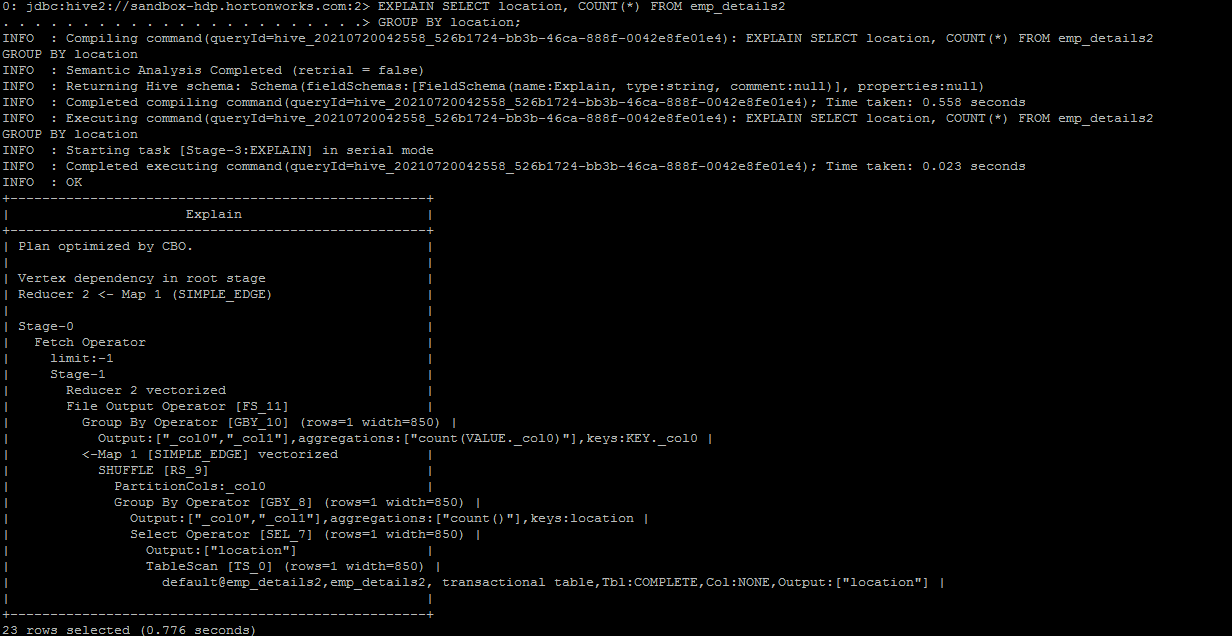
*EXPLAIN SELECT location, COUNT(\*) FROM emp\_details2 GROUP BY location;*

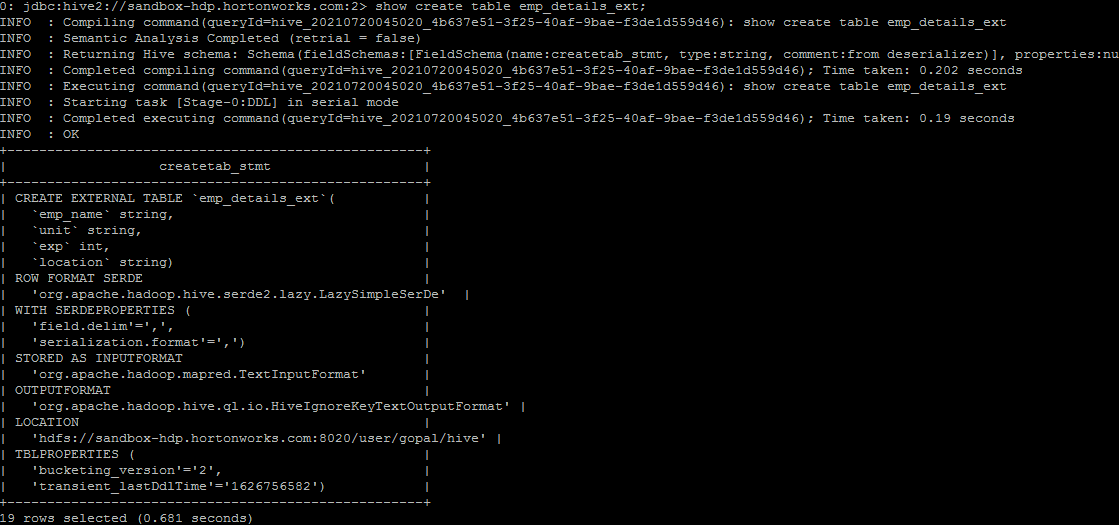
**

*EXPLAIN SELECT \* FROM emp\_details2 WHERE exp >= 2;*

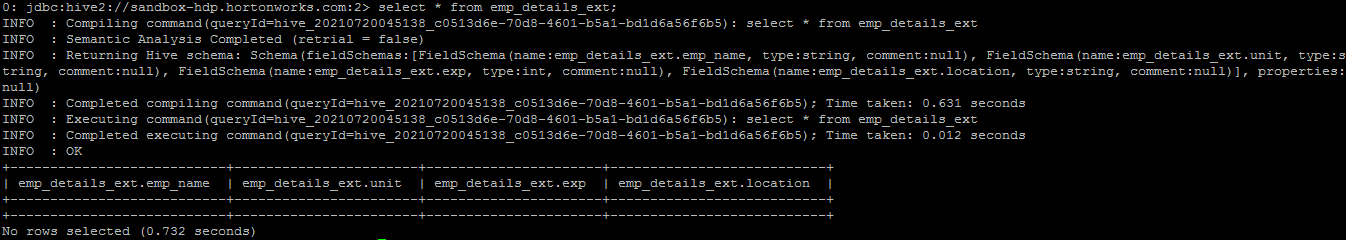
**

*EXPLAIN SELECT location, COUNT(\*) FROM emp\_details2 GROUP BY location;*

**

**

*select \* from emp\_details\_ext;*

**

*create external table emp\_details\_ext*

*(*

*emp\_name string,*

*unit string,*

*exp int,*

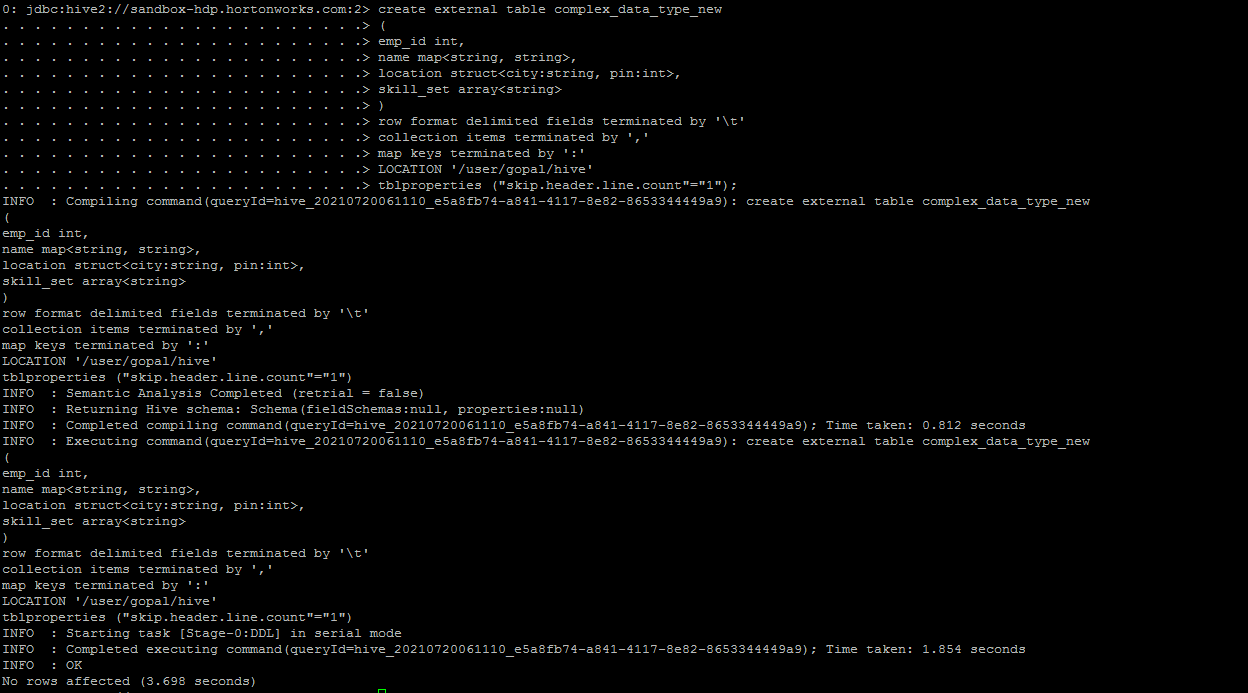
*location string*

*)*

*row format delimited*

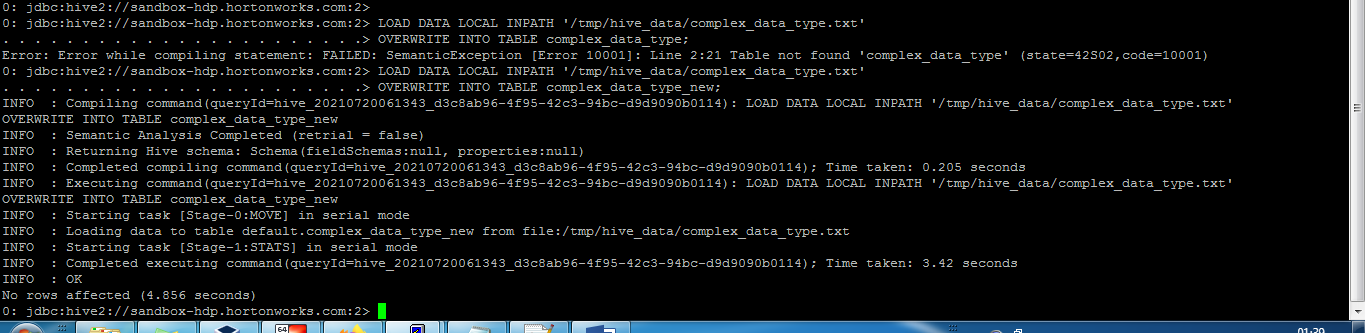
*fields terminated by ','*

*location '/user/gopal/hive/';*

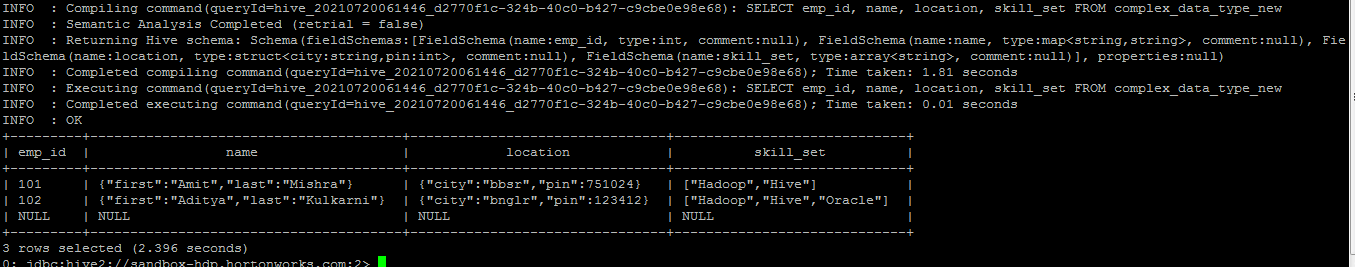
**

*LOAD DATA LOCAL INPATH '/tmp/hive\_data/complex\_data\_type.txt'*

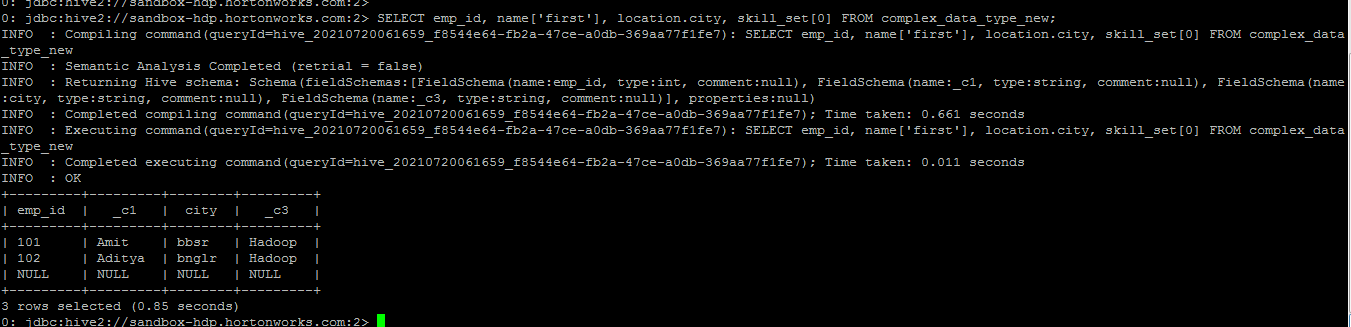
*OVERWRITE INTO TABLE complex\_data\_type\_new;*

**

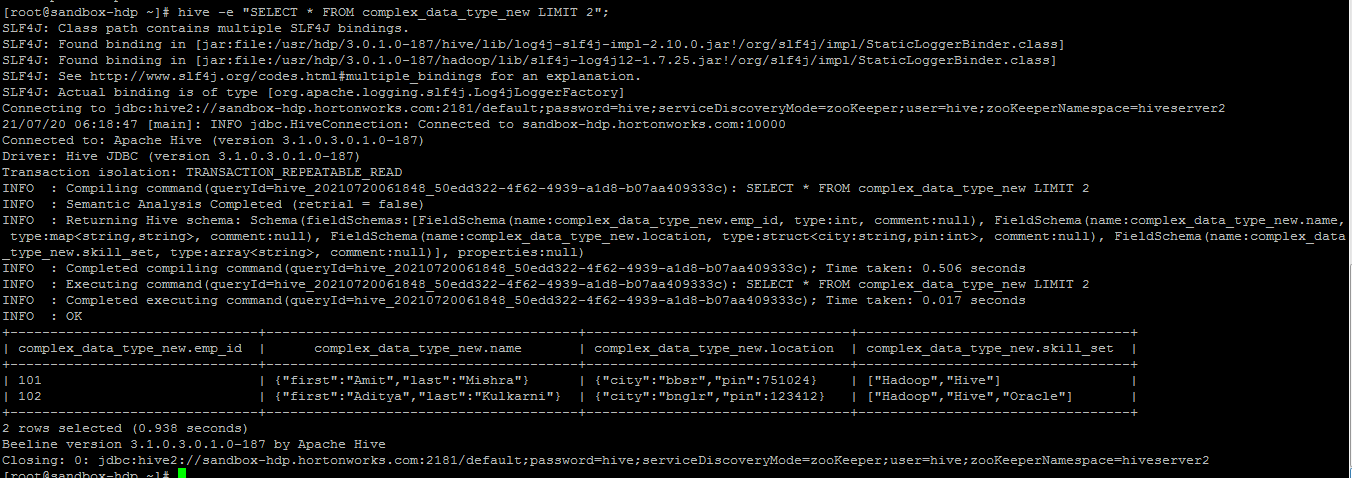
*SELECT emp\_id, name, location, skill\_set FROM complex\_data\_type\_new;*

**

*SELECT emp\_id, name['first'], location.city, skill\_set[0] FROM complex\_data\_type\_new;*

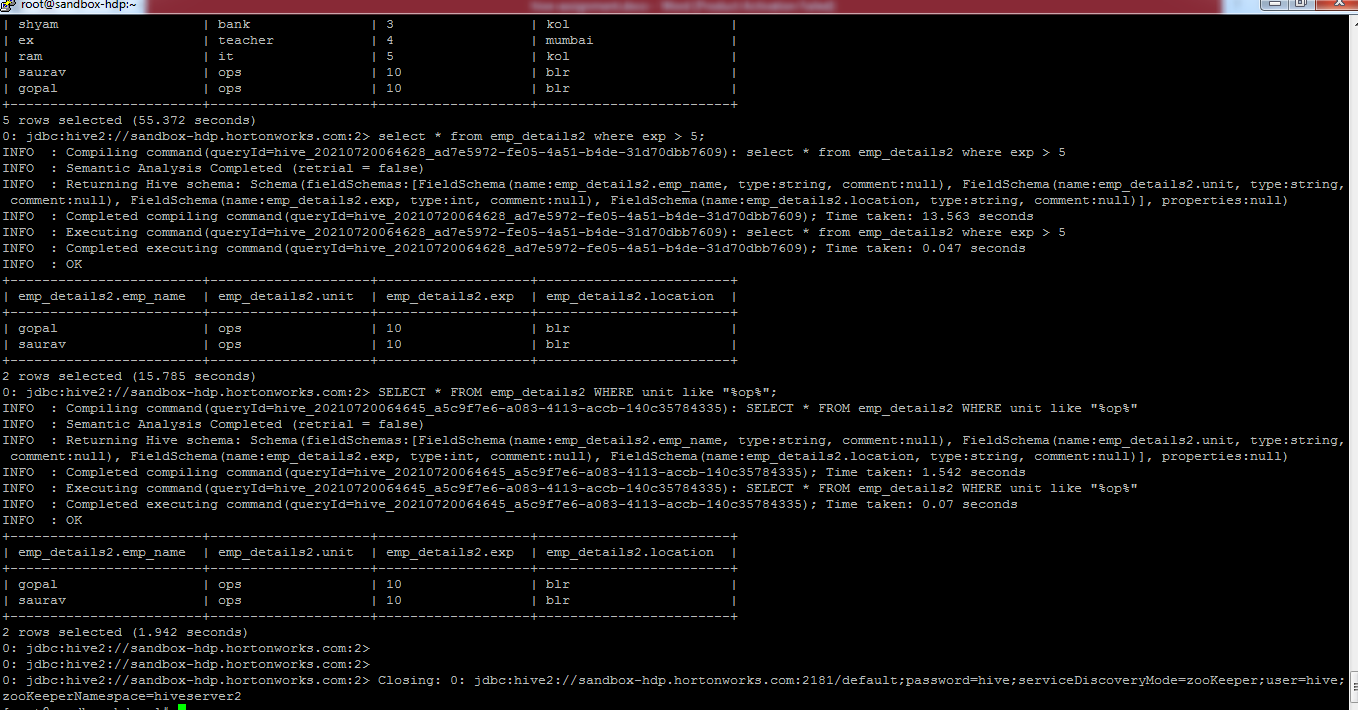
**

*hive -e "SELECT \* FROM complex\_data\_type\_new LIMIT 2";*

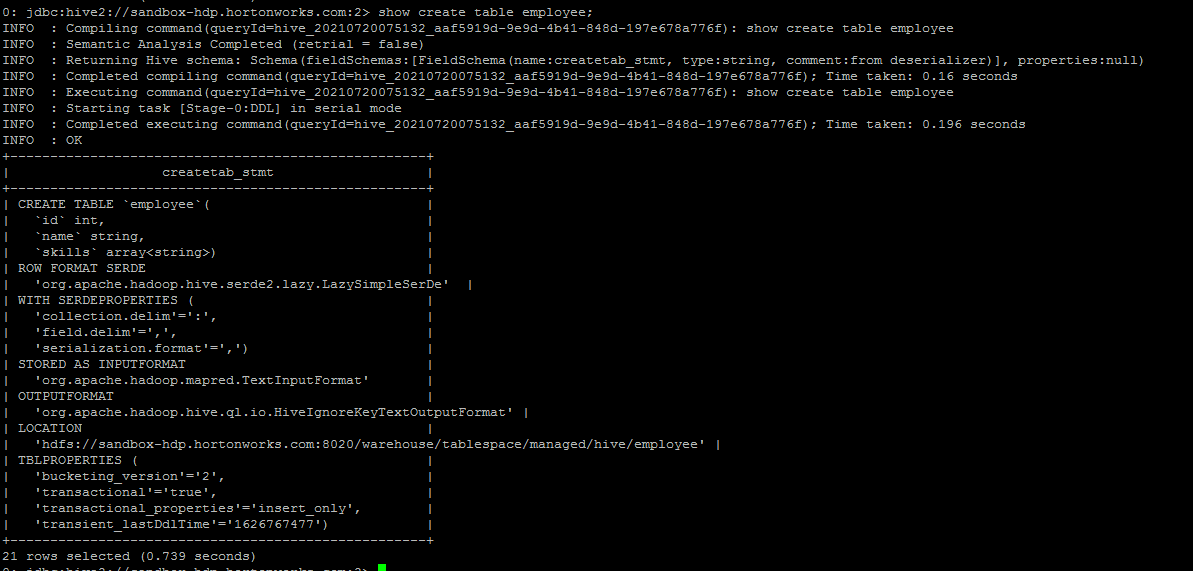
**

*hive --define ROWS=2 -S -e "select \* FROM complex\_data\_type\_new LIMIT ${ROWS}" > /tmp/hive\_data/myquery*

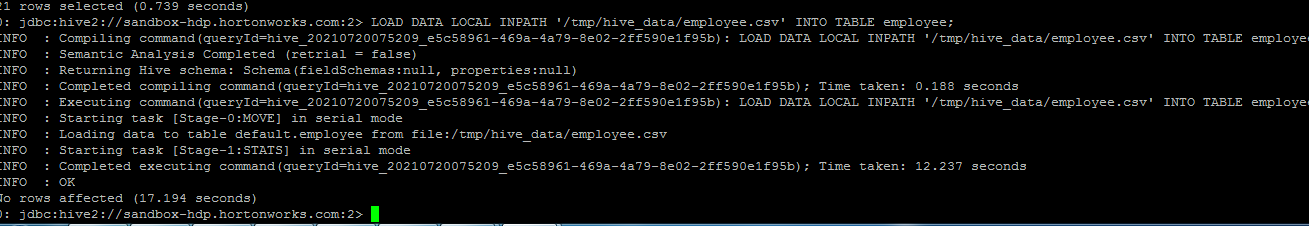
*hive -f /tmp/hive\_data/query.hql*

**

*Hive practical:*

**

*LOAD DATA LOCAL INPATH '/tmp/hive\_data/employee.csv' INTO TABLE employee;*

**

*SELECT*

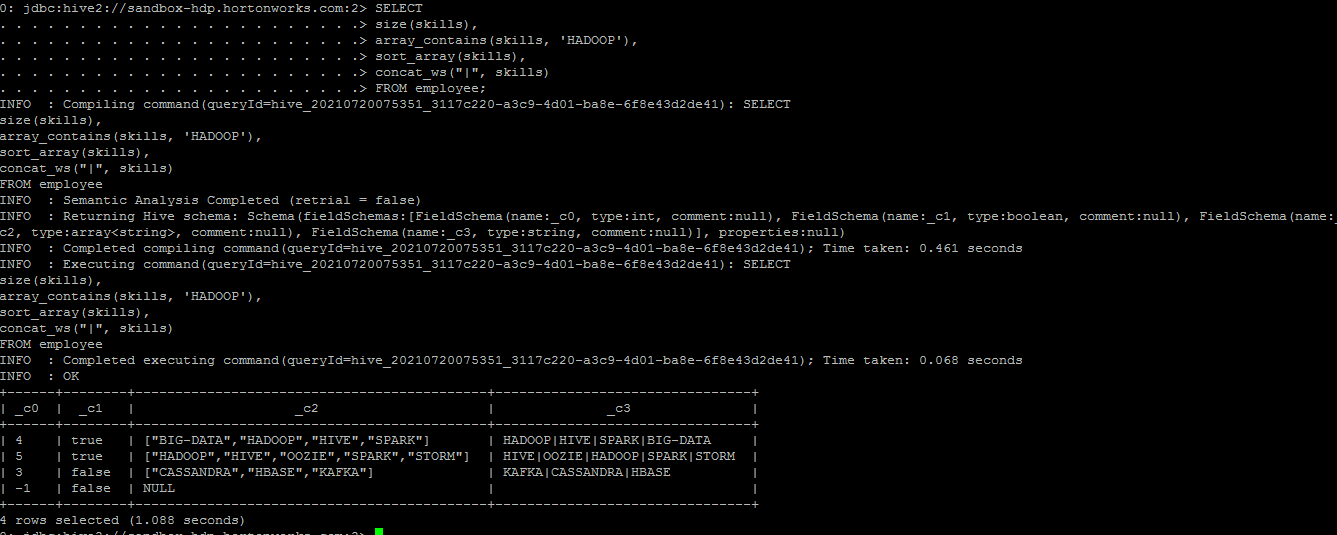
*size(skills),*

*array\_contains(skills, 'HADOOP'),*

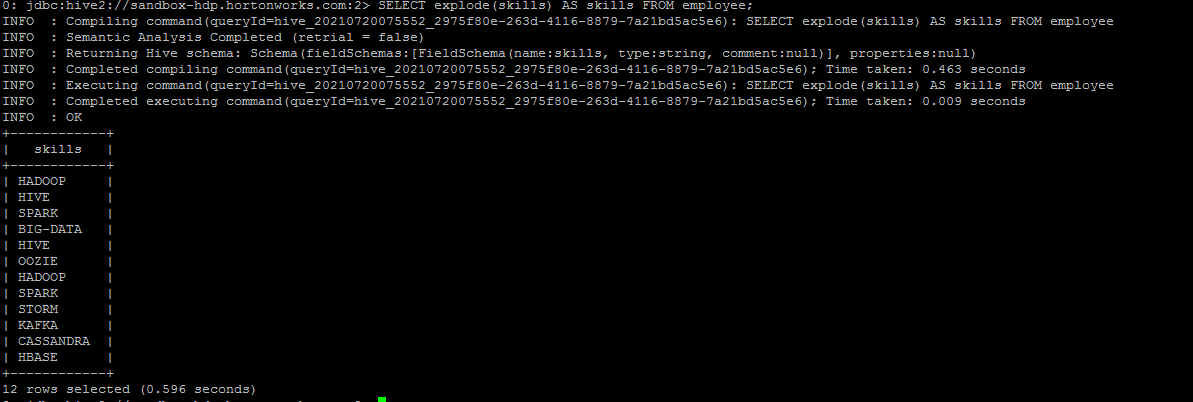
*sort\_array(skills),*

*concat\_ws("|", skills)*

*FROM employee;*

**

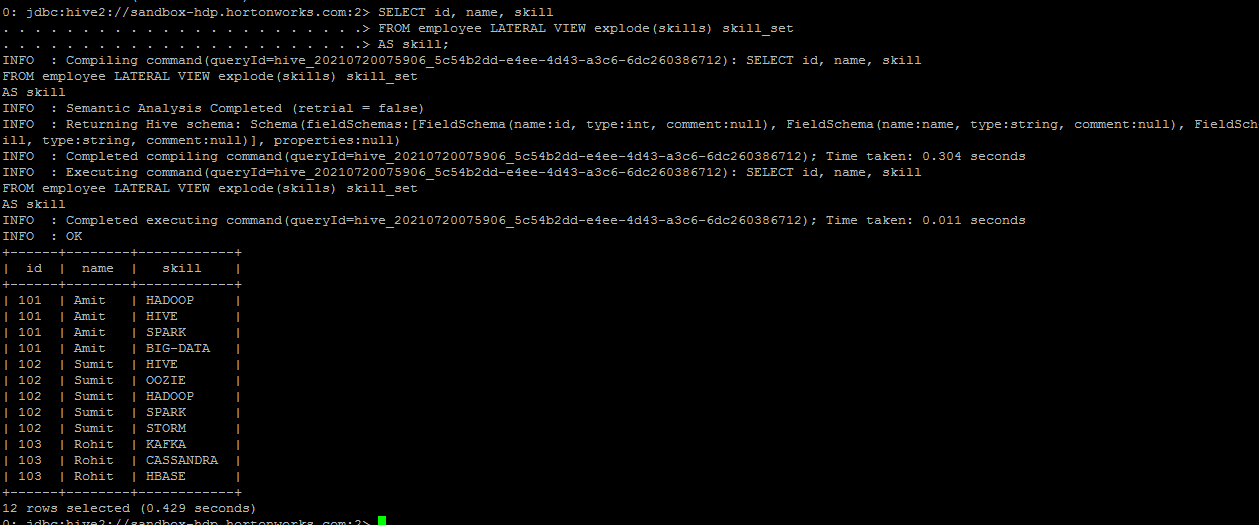
*SELECT explode(skills) AS skills FROM employee;*

**

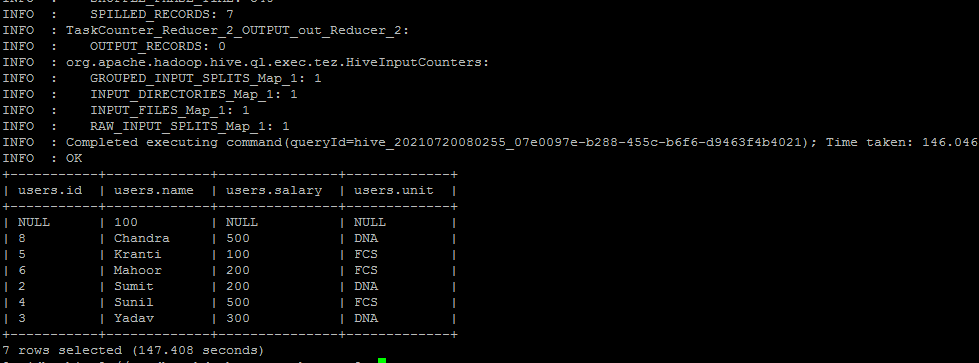
*SELECT id, name, skill*

*FROM employee LATERAL VIEW explode(skills) skill\_set*

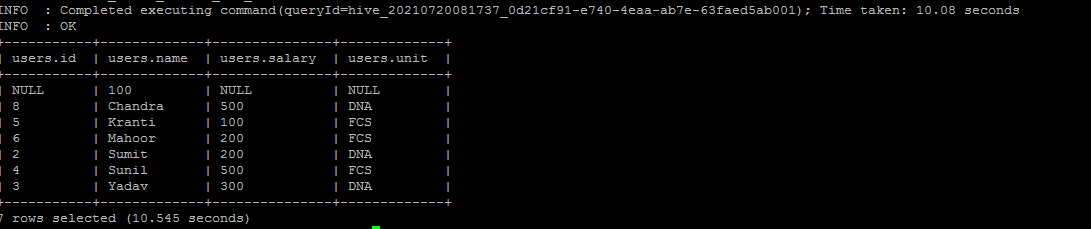
*AS skill;*

**

*SELECT \* FROM my\_db.users ORDER BY name ASC;*

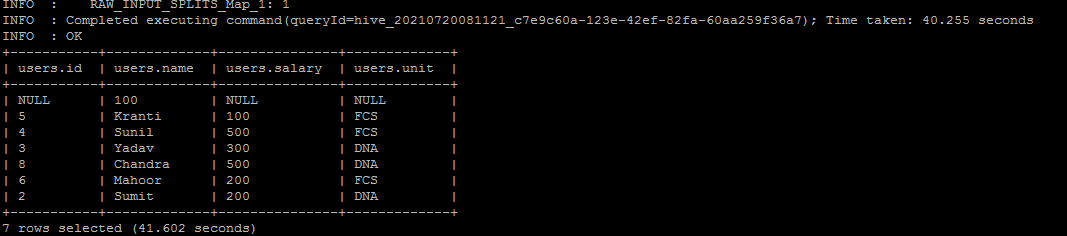
**

*SELECT \* FROM my\_db.users SORT BY name ASC;*

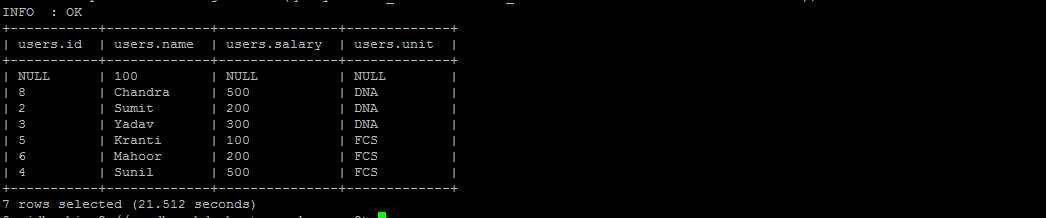
**

*set mapred.reduce.tasks=2;*

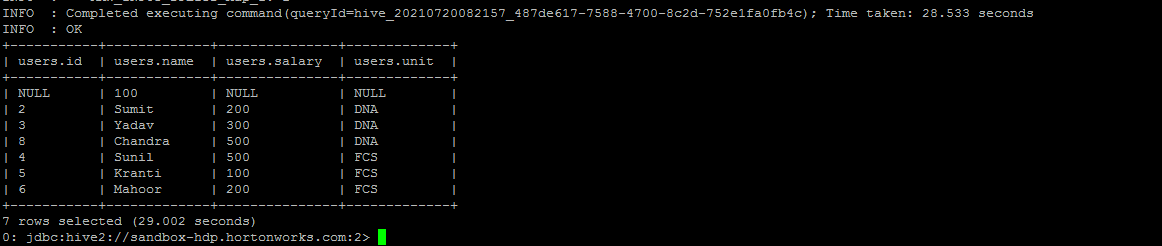
*SELECT \* FROM my\_db.users SORT BY name ASC;*

**

*SELECT \* FROM my\_db.users DISTRIBUTE BY unit SORT BY name ASC;*

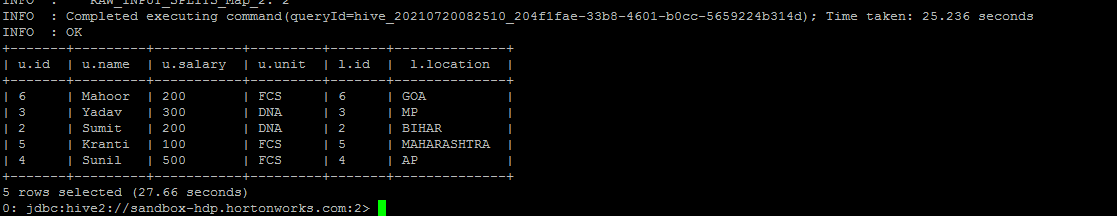
**

*SELECT \* FROM my\_db.users CLUSTER BY unit;*

**

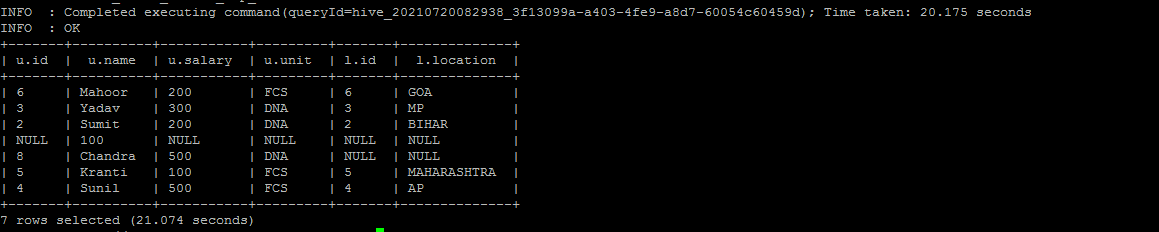
*SELECT \* FROM my\_db.buck\_users u INNER JOIN my\_db.buck\_locations l*

*ON u.id = l.id;*

**

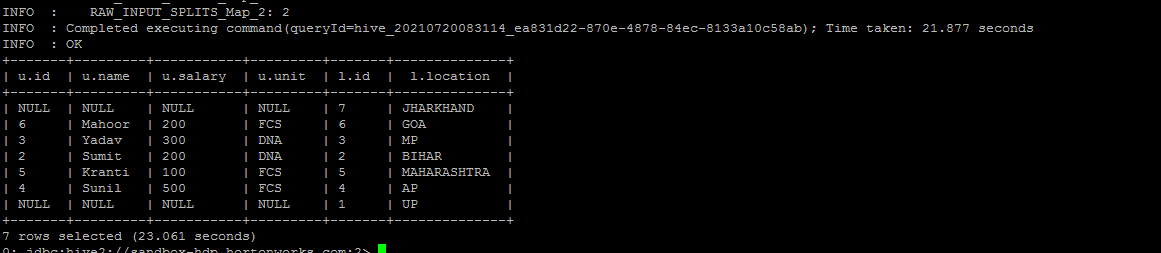
*SELECT \* FROM my\_db.buck\_users u LEFT OUTER JOIN my\_db.buck\_locations l*

*ON u.id = l.id;*

**

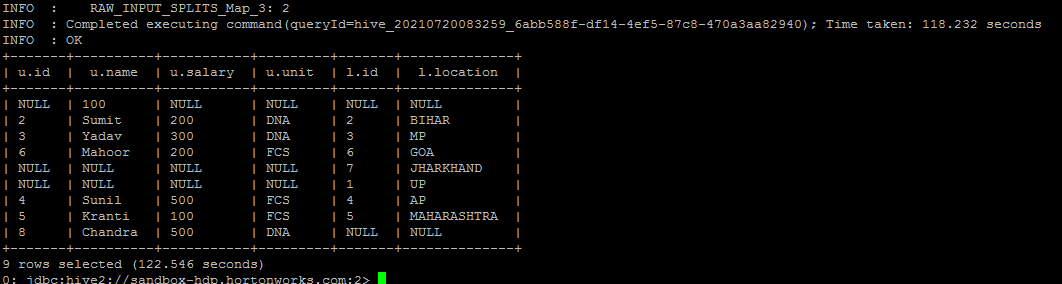
*SELECT \* FROM my\_db.buck\_users u RIGHT OUTER JOIN my\_db.buck\_locations l*

*ON u.id = l.id;*

**

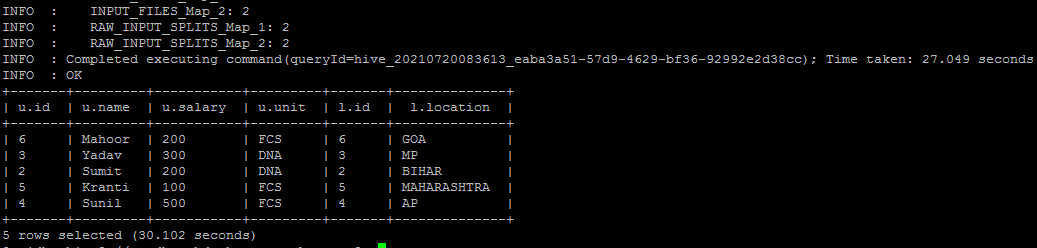
*SELECT \* FROM my\_db.buck\_users u FULL OUTER JOIN my\_db.buck\_locations l*

*ON u.id = l.id;*

**

*SELECT \* FROM my\_db.buck\_users u JOIN my\_db.buck\_locations l*

*ON u.id = l.id;*

**

*create table emp\_details*

*(*

*emp\_name string,*

*unit string,*

*exp int,*

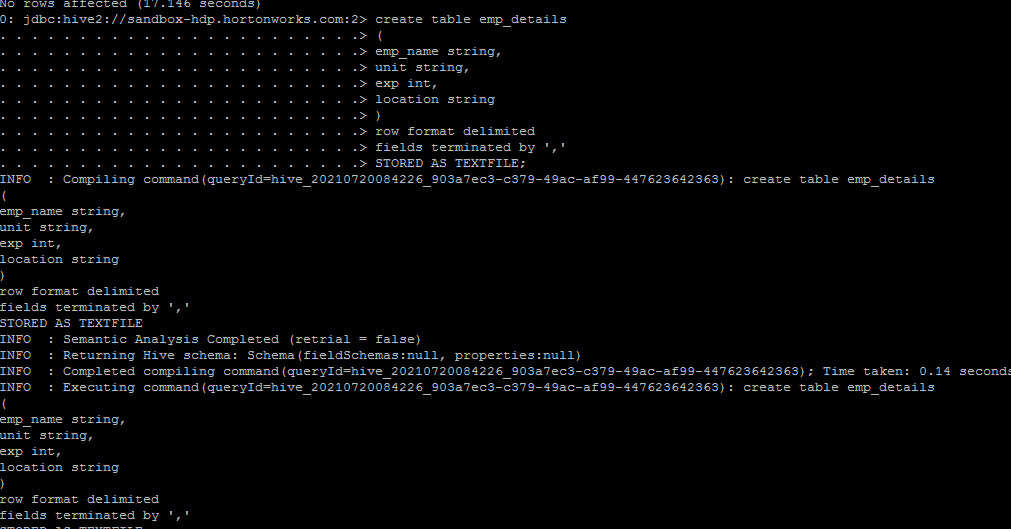
*location string*

*)*

*row format delimited*

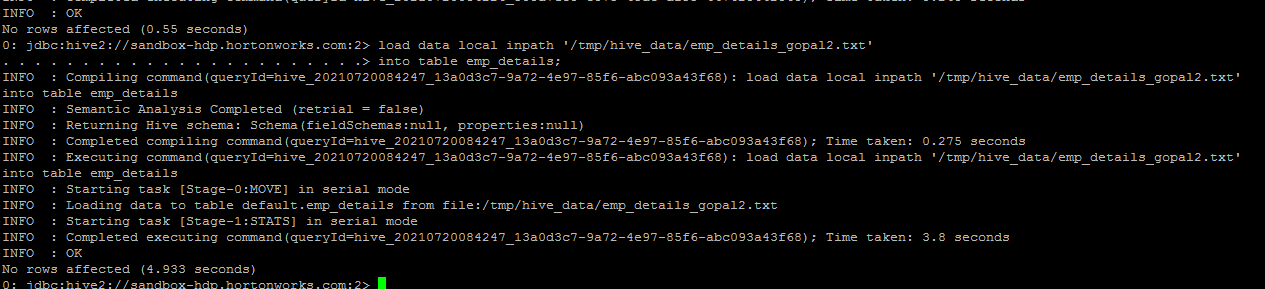
*fields terminated by ','*

*STORED AS TEXTFILE;*

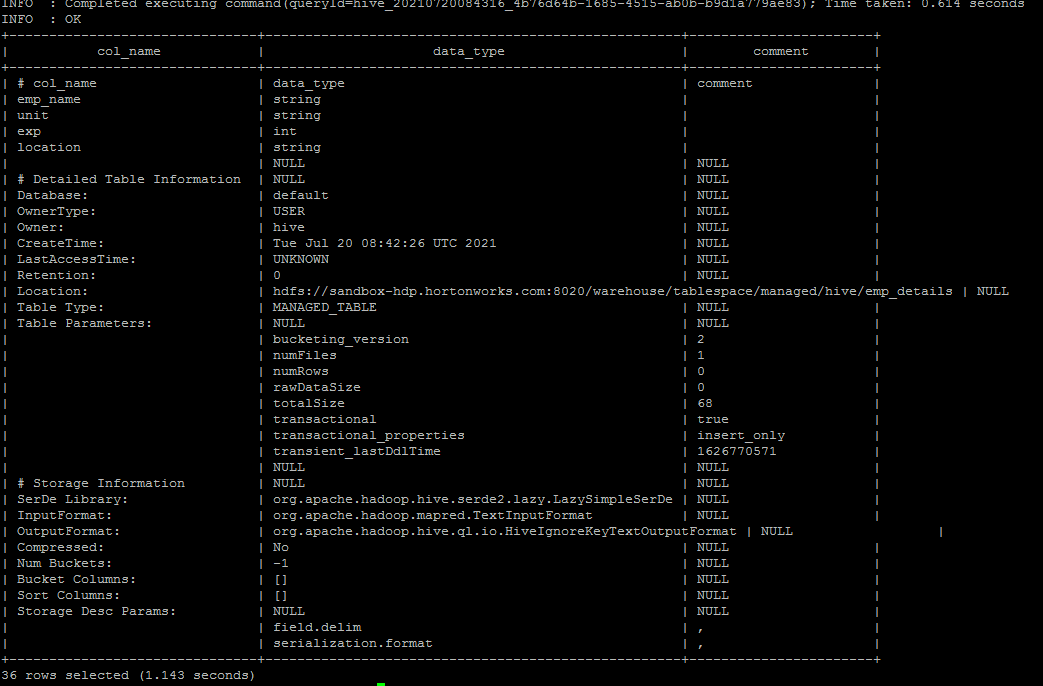
**

*load data local inpath '/tmp/hive\_data/emp\_details\_gopal2.txt'*

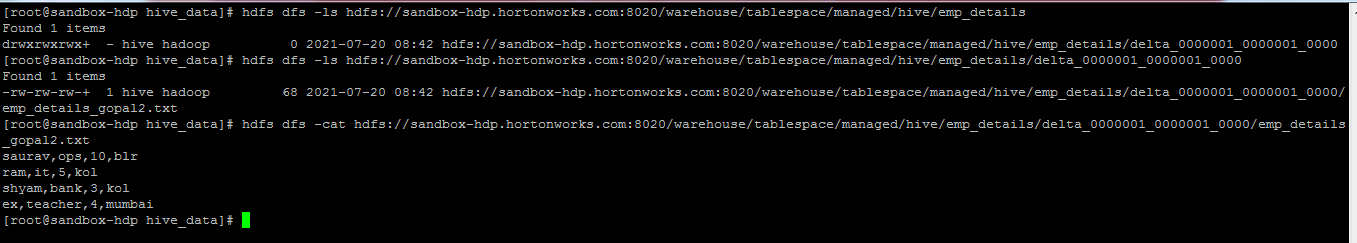
*into table emp\_details;*

**

*describe formatted emp\_details;*

**

*hdfs dfs -ls hdfs://sandbox-hdp.hortonworks.com:8020/warehouse/tablespace/managed/hive/emp\_details*

**

*create table emp\_details\_partitioned*

*(*

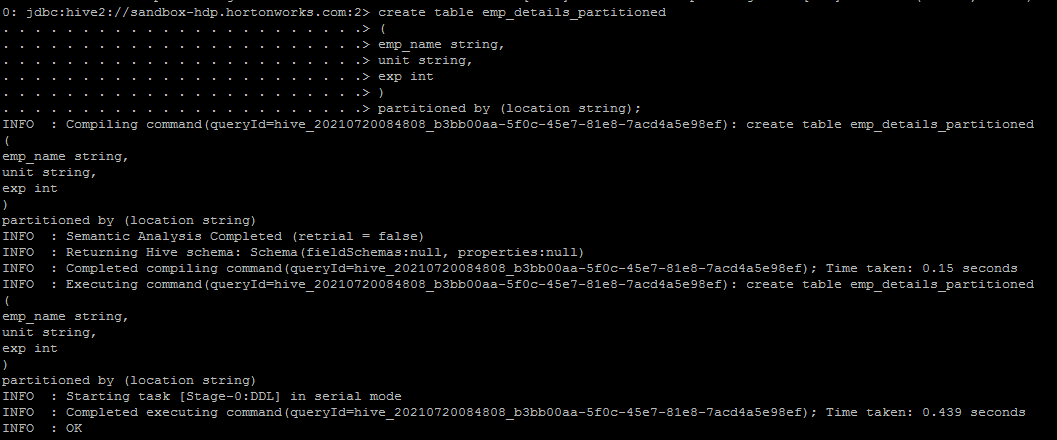
*emp\_name string,*

*unit string,*

*exp int*

*)*

*partitioned by (location string);*

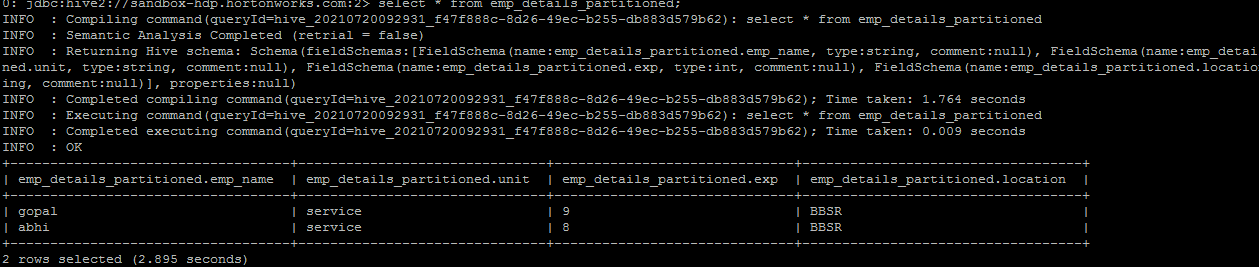
**

*insert overwrite table emp\_details\_partitioned*

*partition(location = 'BBSR')*

*select emp\_name, unit, exp from emp\_details*

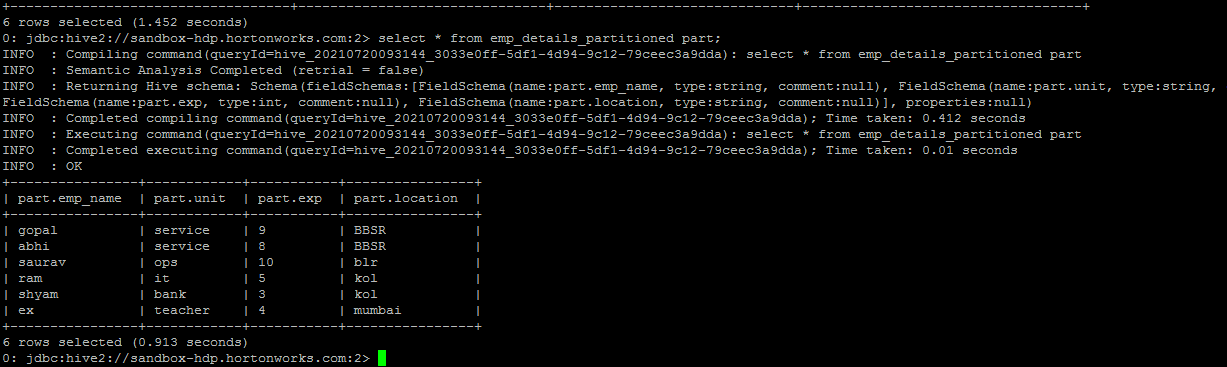
*where location = 'BBSR';*

**

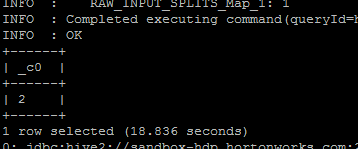
*insert overwrite table emp\_details\_partitioned*

*partition (location)*

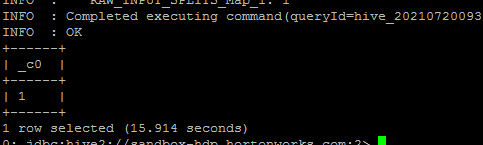
*select \* from emp\_details;*

**

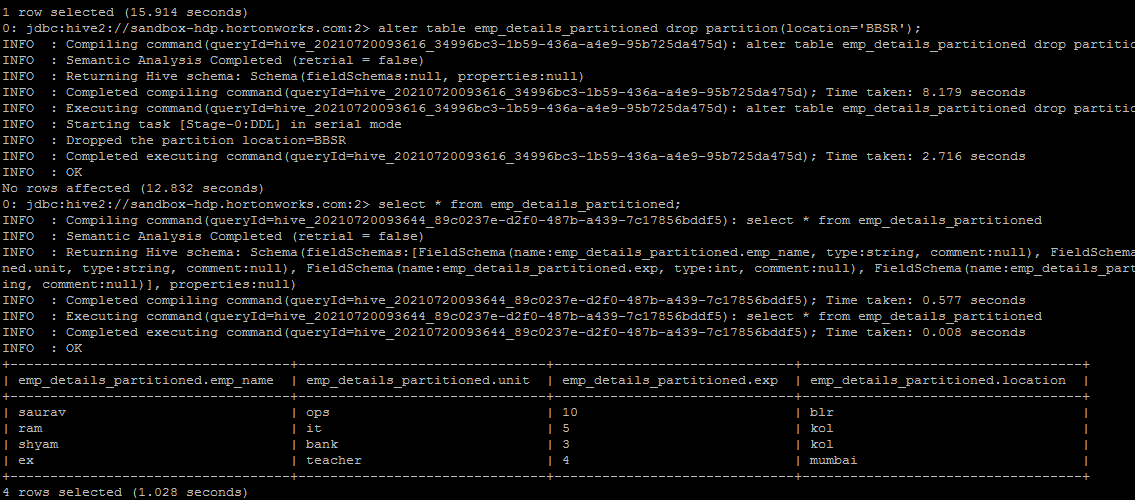
*select count(\*) from emp\_details where location='BBSR';*

**

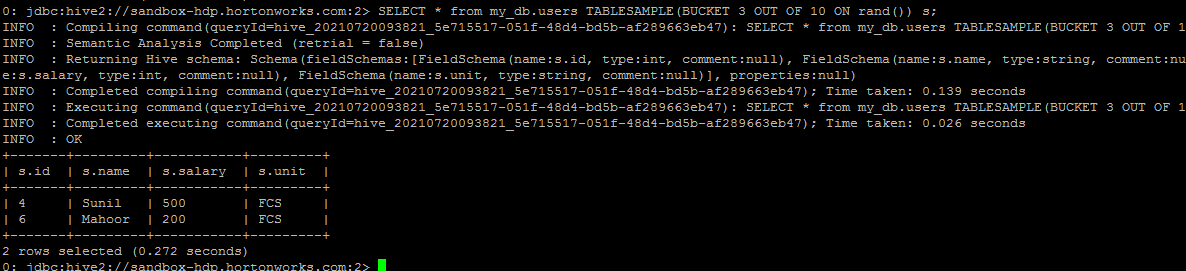
*select count(\*) from emp\_details where emp\_name='saurav';*

**

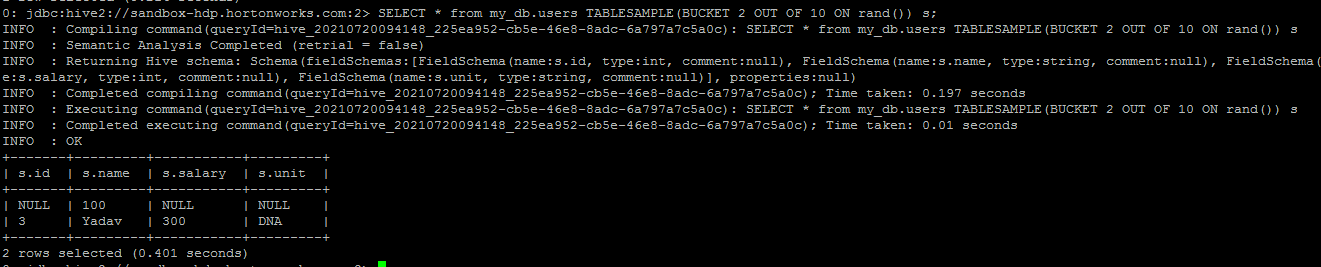
*alter table emp\_details\_partitioned drop partition(location='BBSR');*

**

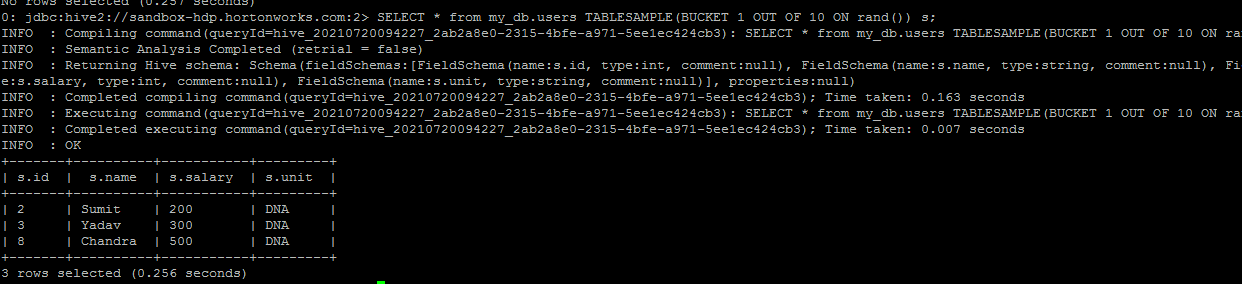
*SELECT \* from my\_db.users TABLESAMPLE(BUCKET 3 OUT OF 10 ON rand()) s;*

**

*SELECT \* from my\_db.users TABLESAMPLE(BUCKET 2 OUT OF 10 ON rand()) s;*

**

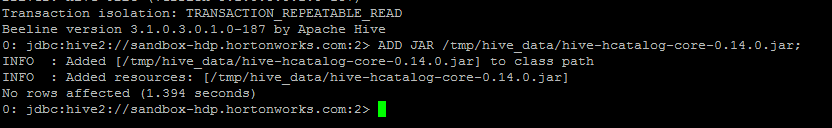
*SELECT \* from my\_db.users TABLESAMPLE(BUCKET 1 OUT OF 10 ON rand()) s;*

**

*load data local inpath '/root/hive/datasets\_for\_fileformats/ratings.dat'*

*into table text\_table;*

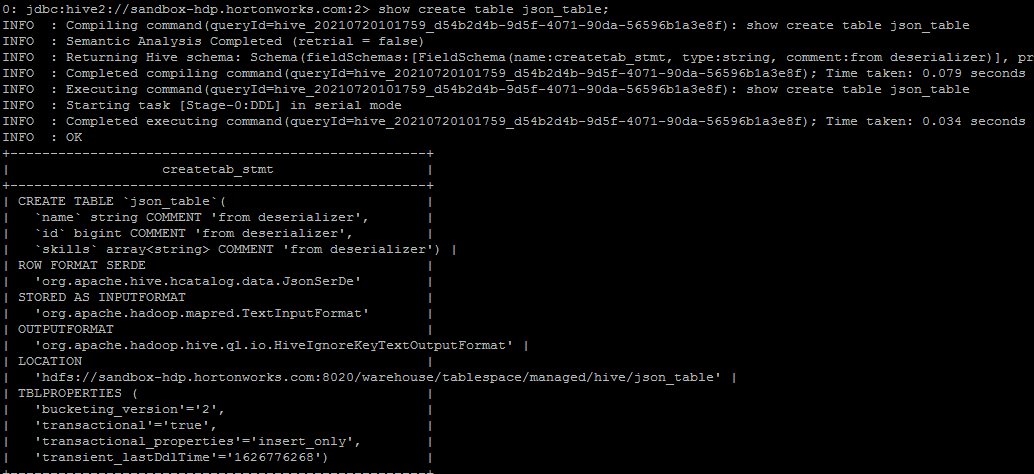
*ADD JAR /tmp/hive\_data/hive-hcatalog-core-0.14.0.jar;*

**

*CREATE TABLE json\_table(name string, id bigint, skills array<string>)*

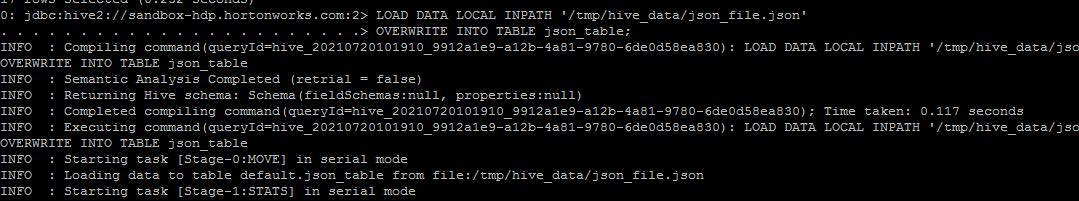
*ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'*

*STORED AS TEXTFILE;*

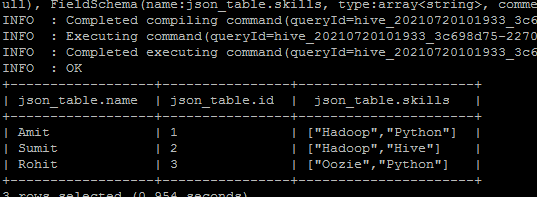
**

*LOAD DATA LOCAL INPATH '/tmp/hive\_data/json\_file.json'*

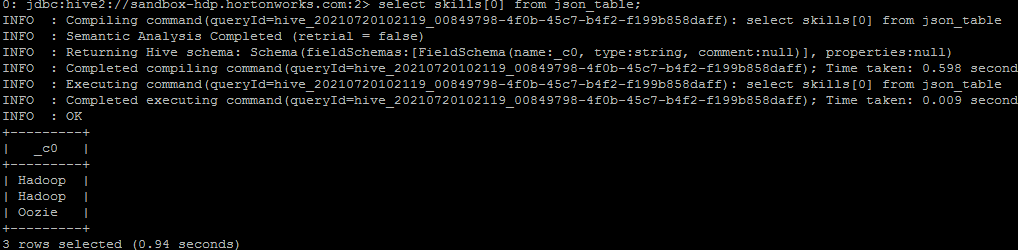
*OVERWRITE INTO TABLE json\_table;*

**

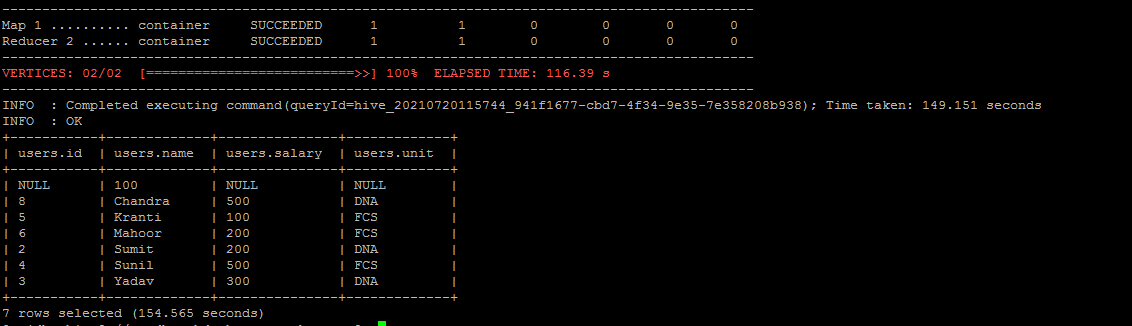
*select \* from json\_table;*

**

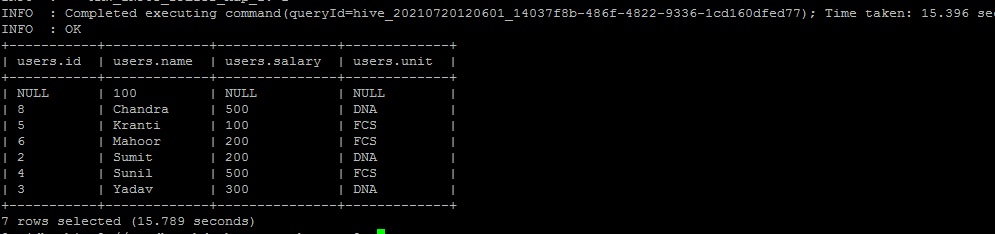
*select skills[0] from json\_table;*

**

*SELECT \* FROM my\_db.users ORDER BY name ASC;*

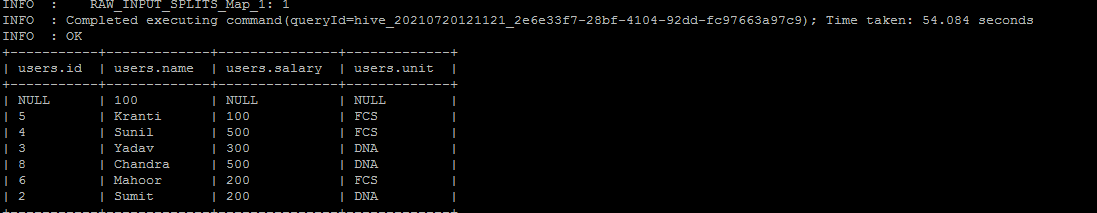
**

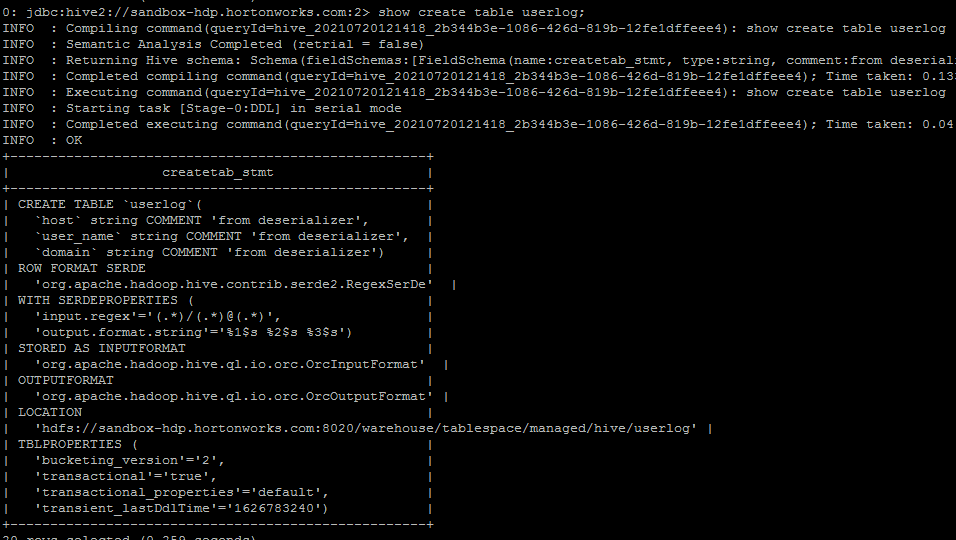
*SELECT \* FROM my\_db.users SORT BY name ASC;*

**

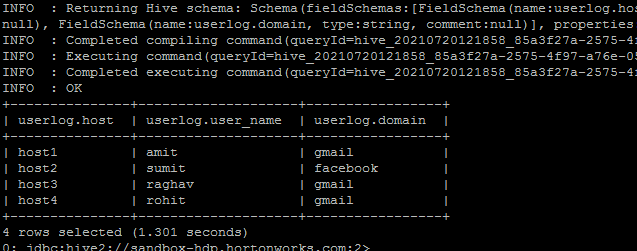
*set mapred.reduce.tasks=2;*

*SELECT \* FROM my\_db.users SORT BY name ASC;*

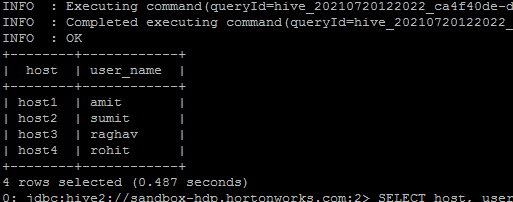
**

**

*select \* from userlog;*

**

*SELECT host, user\_name FROM userlog;*

**