JUNAID GIRKAR

60004190057

TE COMPS A4

**EXPERIMENT - 4**

**AIM**: Execute HIVE commands to load, insert, retrieve, update, or delete data in the tables.

**THEORY**:

Apache Hive is a data warehouse and an ETL tool which provides an SQL-like interface between the user and the Hadoop distributed file system (HDFS) which integrates Hadoop. It is built on top of Hadoop. It is a software project that provides data query and analysis. It facilitates reading, writing and handling wide datasets stored in distributed storage and queried by Structure Query Language (SQL) syntax. It is not built for Online Transactional Processing (OLTP) workloads. It is frequently used for data warehousing tasks like data encapsulation, Ad-hoc Queries, and analysis of huge datasets. It is designed to enhance scalability, extensibility, performance, fault-tolerance and loose-coupling with its input formats.

Initially Hive is developed by Facebook and Amazon, Netflix and It delivers standard SQL functionality for analytics. Traditional SQL queries are written in the MapReduce Java API to execute SQL Application and SQL queries over distributed data. Hive provides portability as most data warehousing applications functions with SQL-based query languages like NoSQL.

**Components of Hive**:

1. **HCatalog** –   
   It is a Hive component and is a table as well as a store management layer for Hadoop. It enables user along with various data processing tools like Pig and MapReduce which enables to read and write on the grid easily.
2. **WebHCat** –   
   It provides a service which can be utilized by the user to run Hadoop MapReduce (or YARN), Pig, Hive tasks or function Hive metadata operations with an HTTP interface.

**Modes of Hive:**

1. **Local Mode** –   
   It is used, when the Hadoop is built under pseudo mode which has only one data node, when the data size is smaller in term of restricted to single local machine, and when processing will be faster on smaller datasets existing in the local machine.
2. **Map Reduce Mode** –   
   It is used, when Hadoop is built with multiple data nodes and data is divided across various nodes, it will function on huge datasets and query is executed parallelly, and to achieve enhanced performance in processing large datasets.

**Characteristics of Hive**:

1. Databases and tables are built before loading the data.
2. Hive as data warehouse is built to manage and query only structured data which is residing under tables.
3. At the time of handling structured data, MapReduce lacks optimization and usability function such as UDFs whereas Hive framework have optimization and usability.
4. Programming in Hadoop deals directly with the files. So, Hive can partition the data with directory structures to improve performance on certain queries.
5. Hive is compatible for the various file formats which are TEXTFILE, SEQUENCEFILE, ORC, RCFILE, etc.
6. Hive uses derby database in single user metadata storage and it uses MYSQL for multiple user Metadata or shared Metadata.

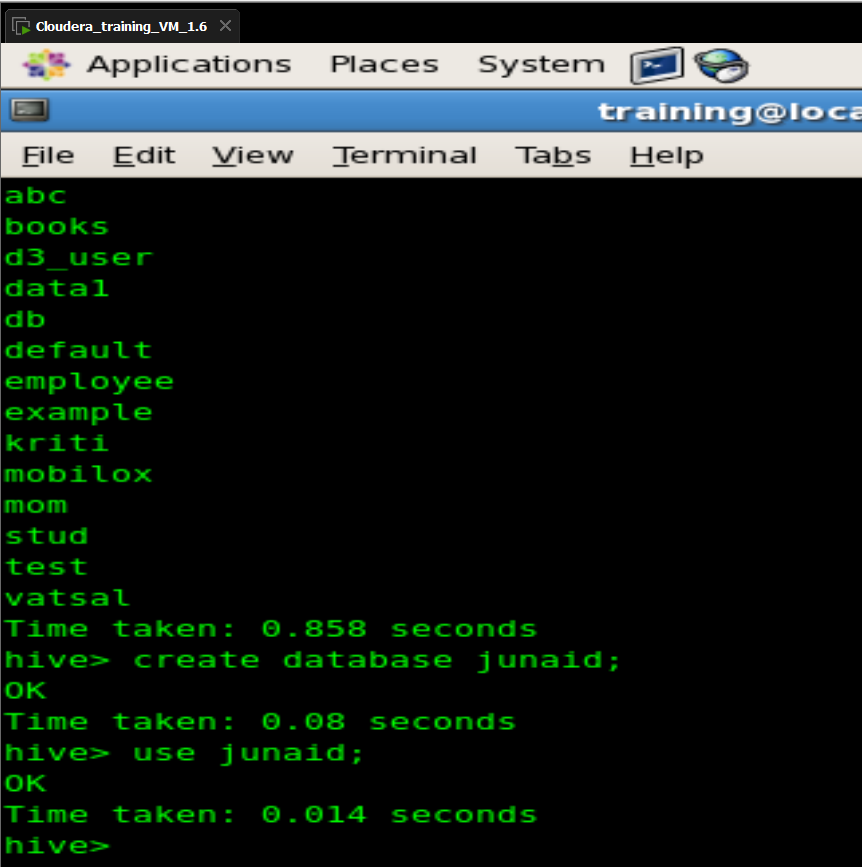
**Features of Hive:**

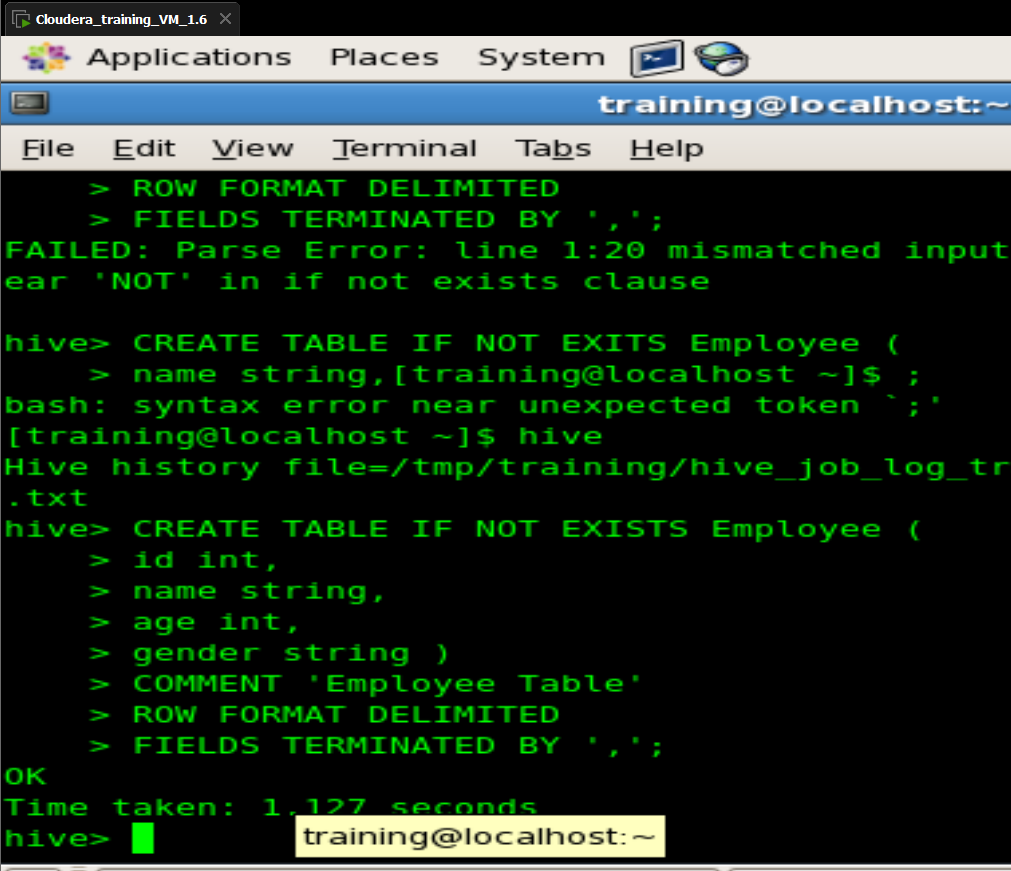
1. It provides indexes, including bitmap indexes to accelerate the queries. Index type containing compaction and bitmap index as of 0.10.
2. Metadata storage in a RDBMS, reduces the time to function semantic checks during query execution.
3. Built in user-defined functions (UDFs) to manipulation of strings, dates, and other data-mining tools. Hive is reinforced to extend the UDF set to deal with the use-cases not reinforced by predefined functions.
4. DEFLATE, BWT, snappy, etc are the algorithms to operation on compressed data which is stored in Hadoop Ecosystem.
5. It stores schemas in a database and processes the data into the Hadoop File Distributed File System (HDFS).
6. It is built for Online Analytical Processing (OLAP).
7. It delivers various types of querying language which are frequently known as Hive Query Language (HVL or HiveQL).

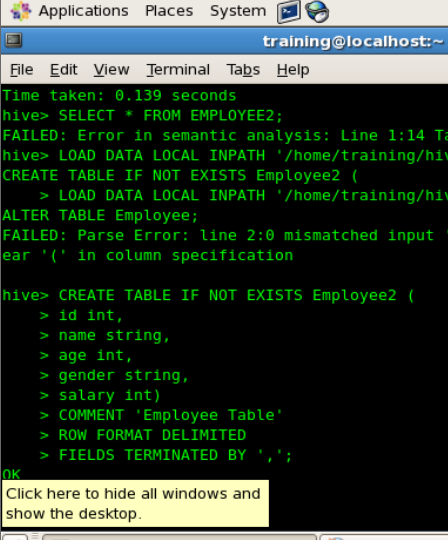
### **Working** –

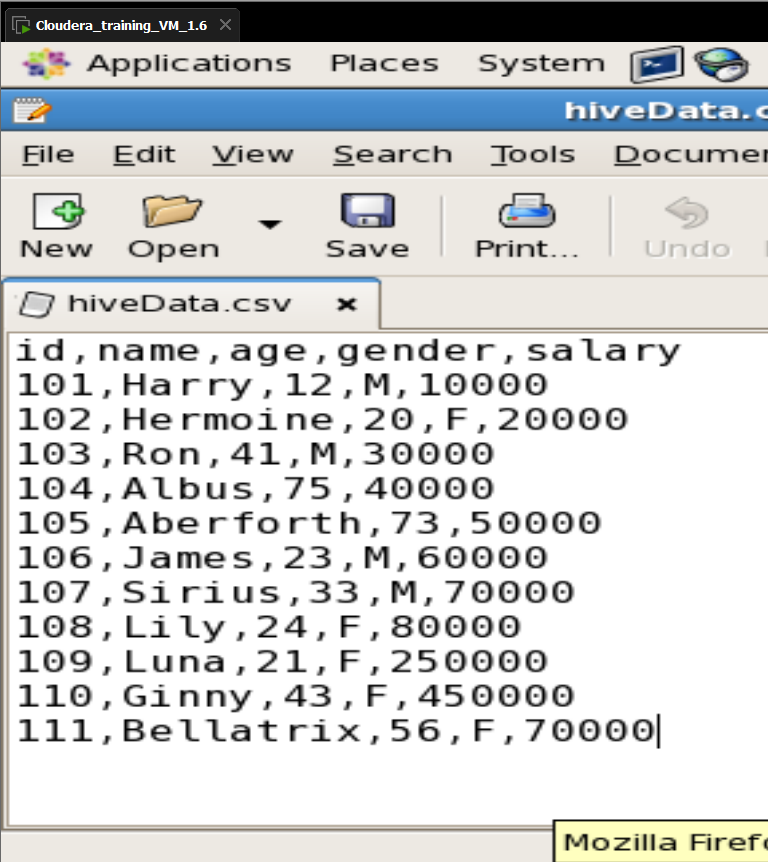
1. First of all, the user submits their query and CLI sends that query to the Driver.
2. Then the driver takes the help of the query compiler to check syntax.
3. Then compiler requests for Metadata by sending a metadata request to Metastore.
4. In response to that request, metastore sends metadata to the compiler.
5. Then the compiler resends the plan to the driver after checking requirements.
6. The Driver sends the plan to the execution engine.
7. Execution engine sends the job to Job tracker and assigns the job to Task Tracker. Here, the query executes the MapReduce job. And in the meantime the execution engine executes metadata operations with Metastore.
8. Then the execution engine fetches the results from the Data Node and sends those results to the driver.
9. At last, the driver sends the results to the hive interface.

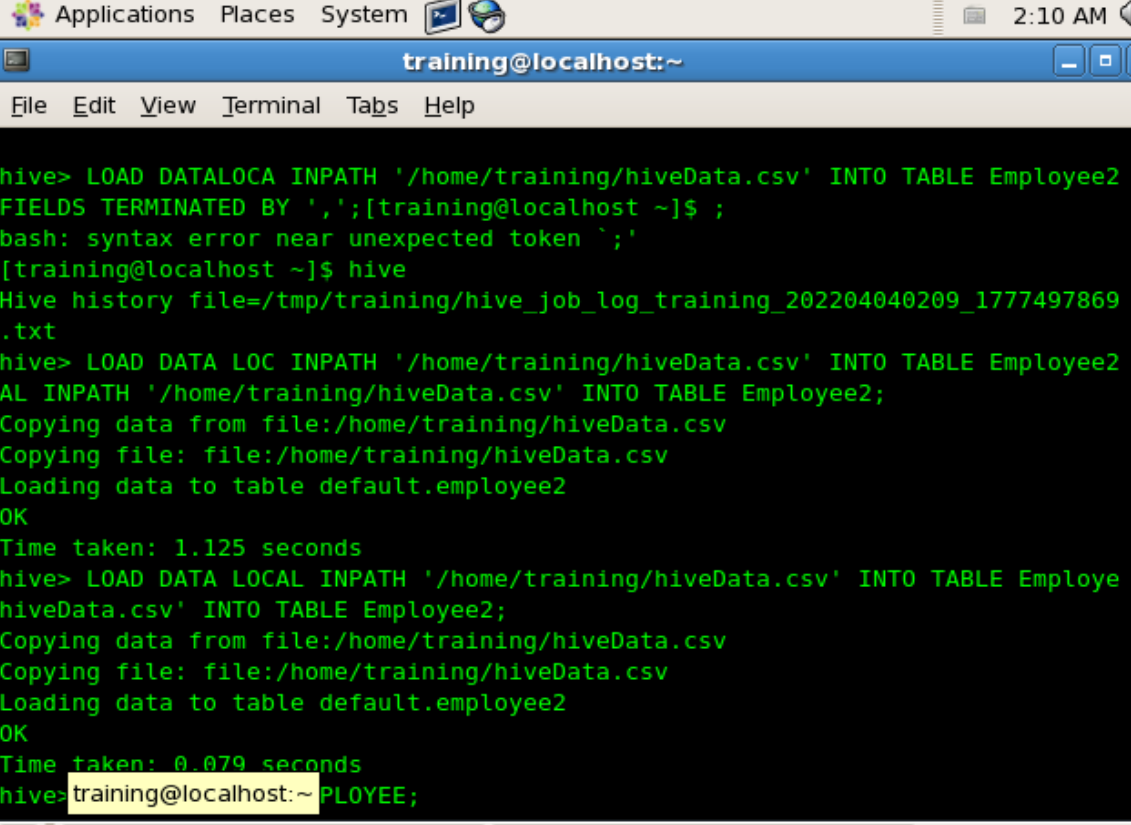
**IMPLEMENTATION:**

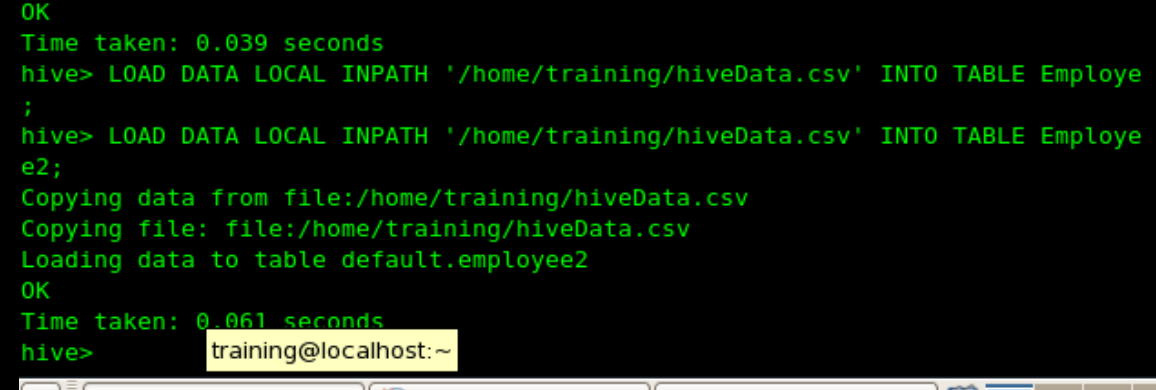


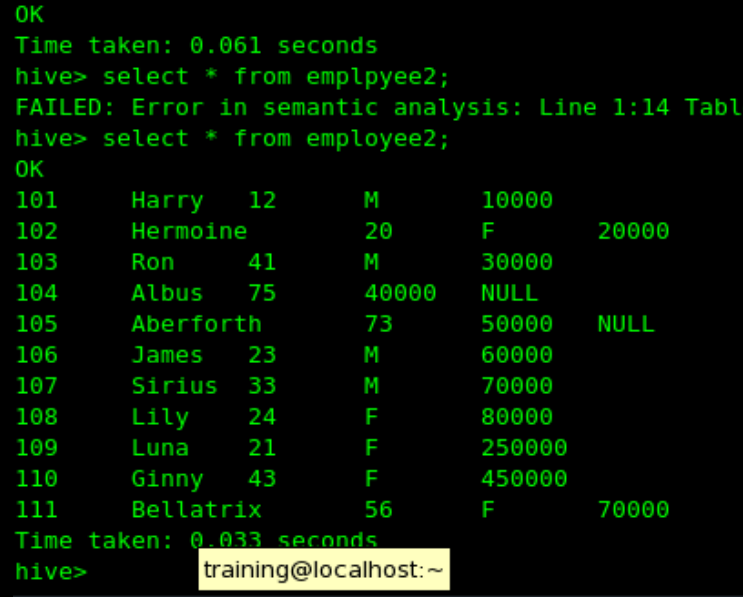


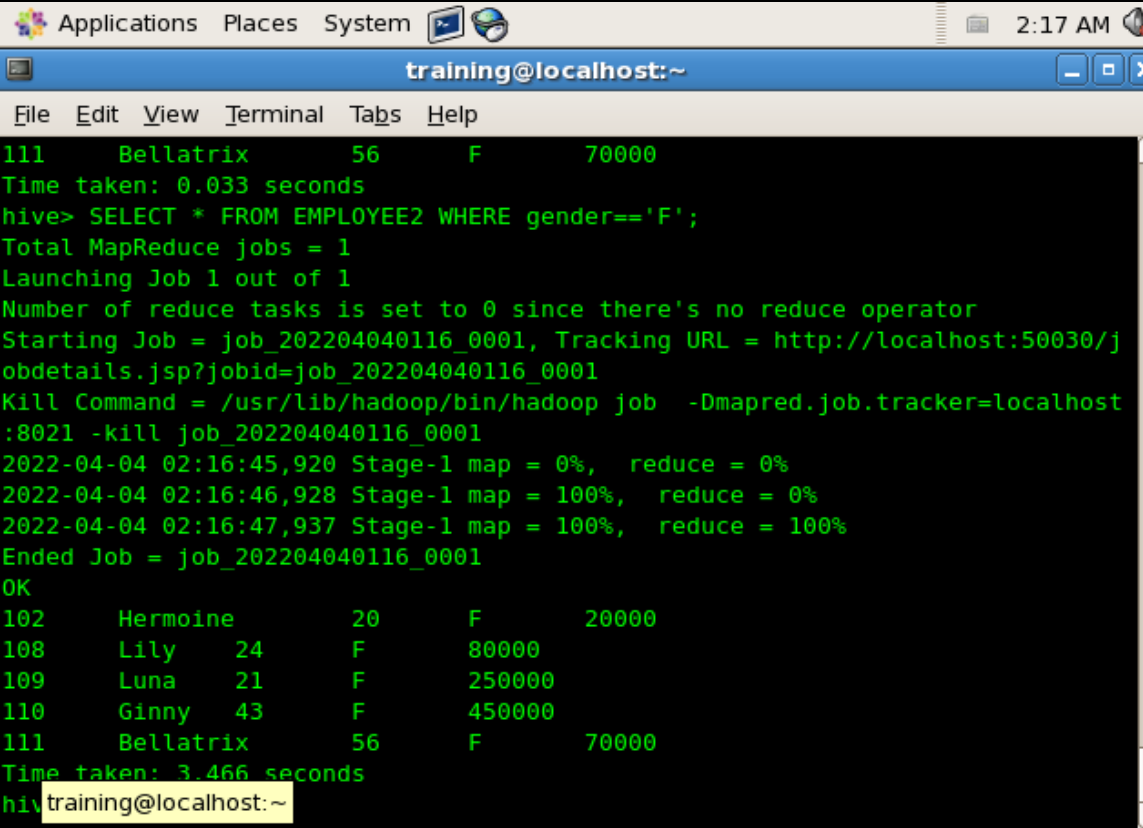


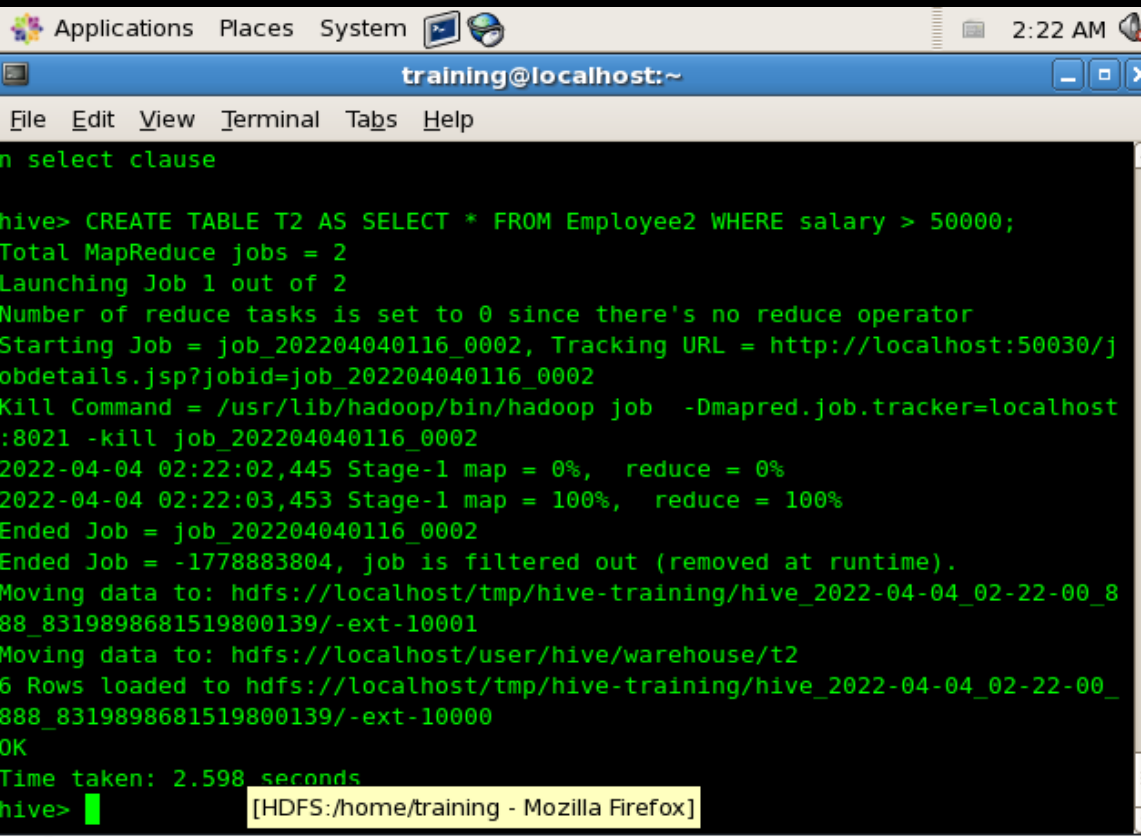


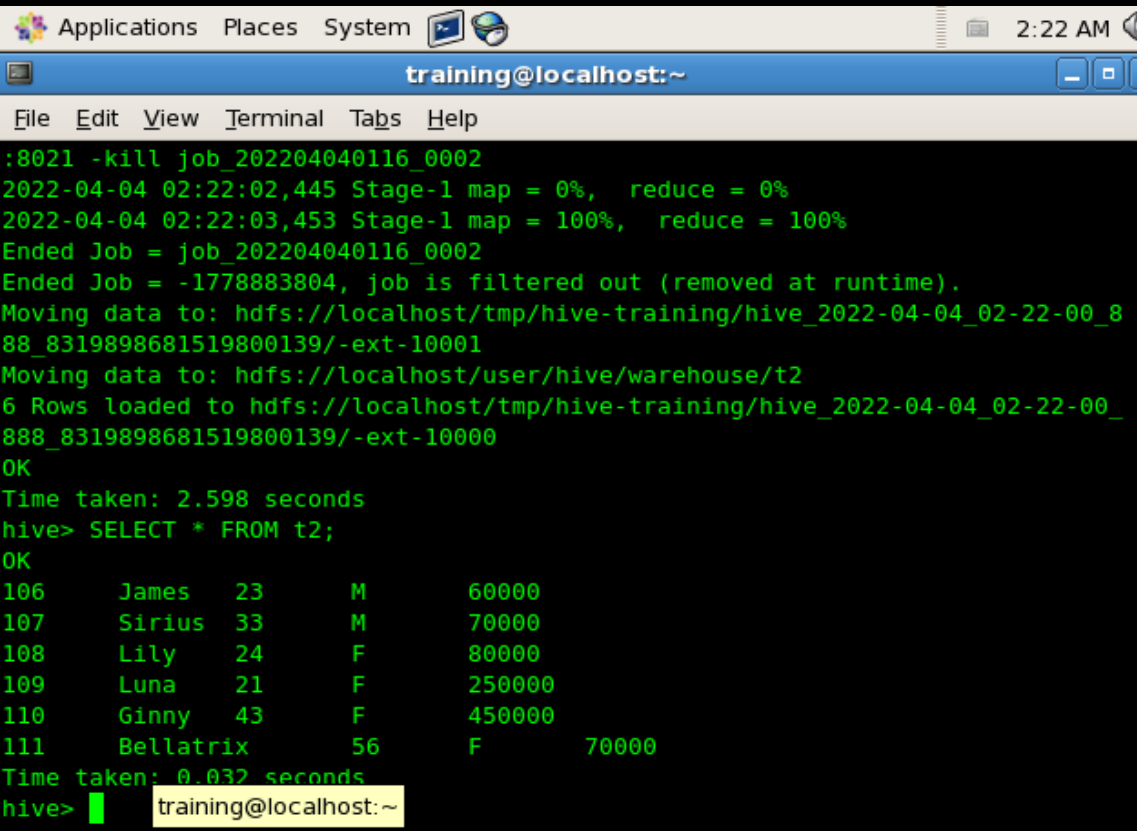


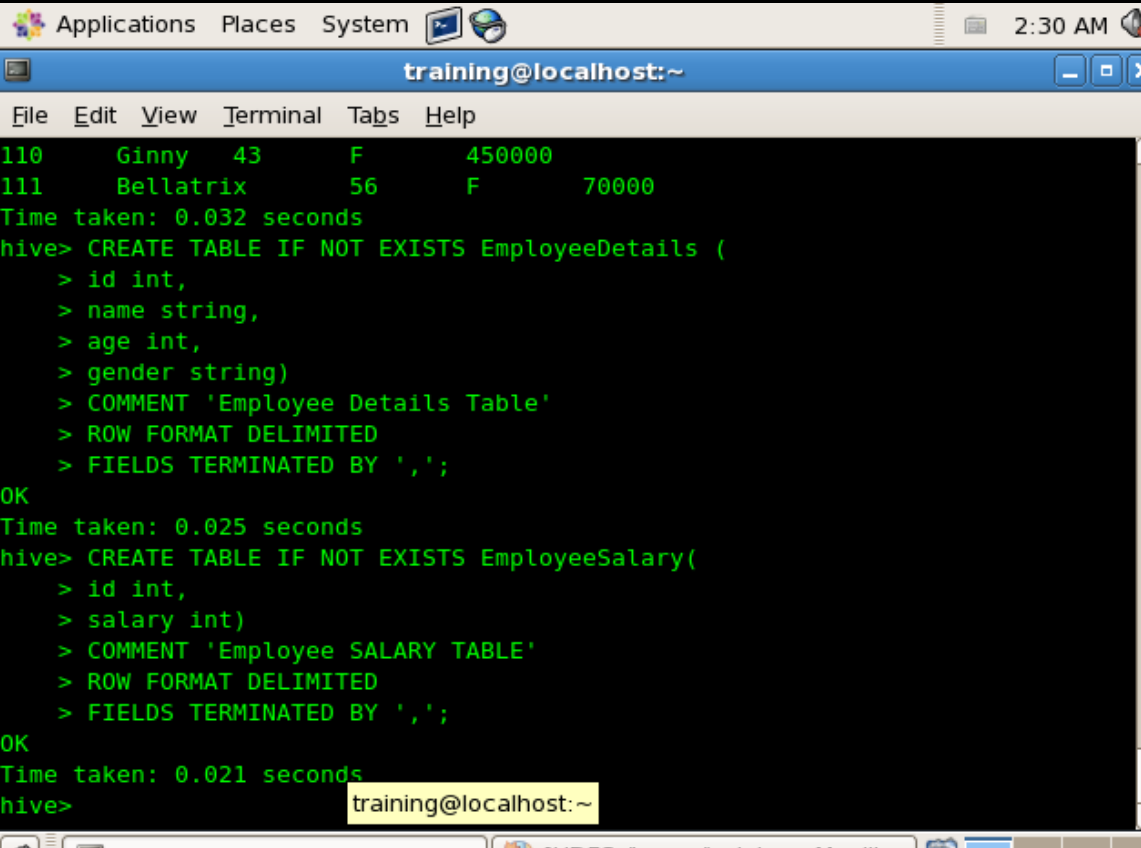


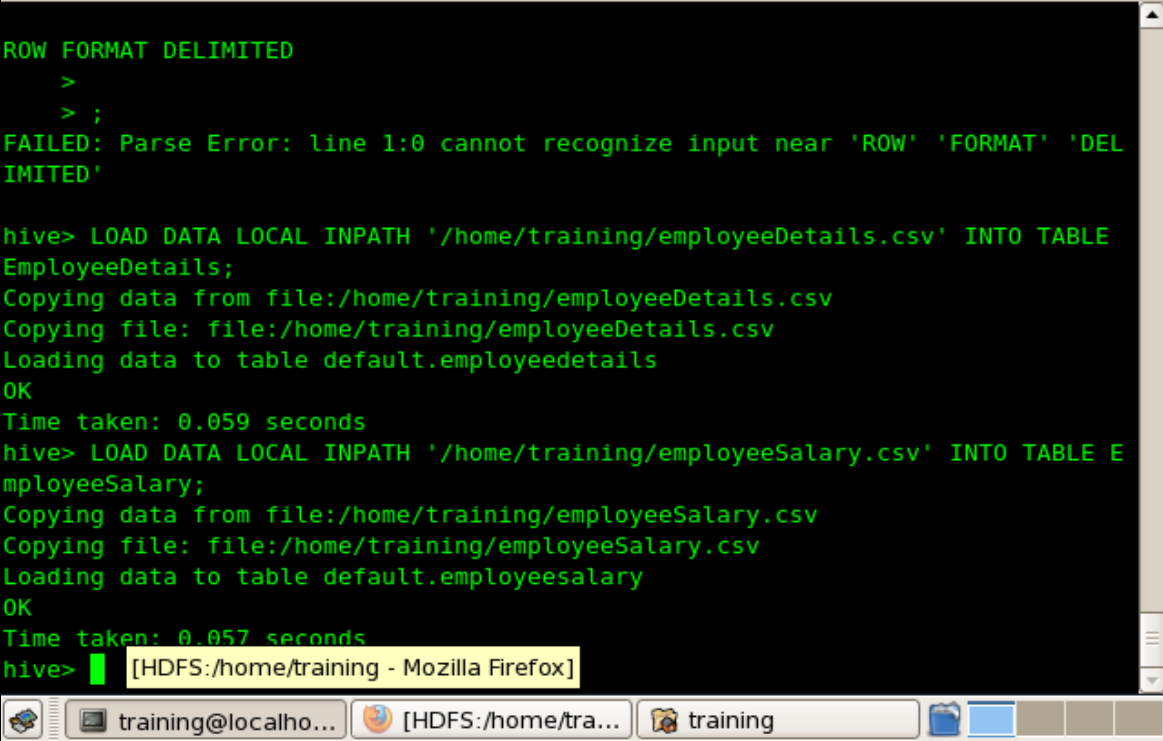




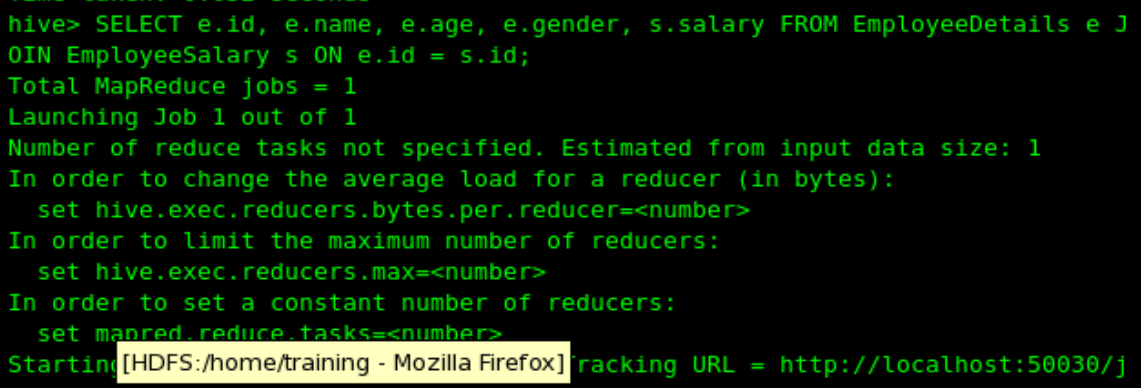


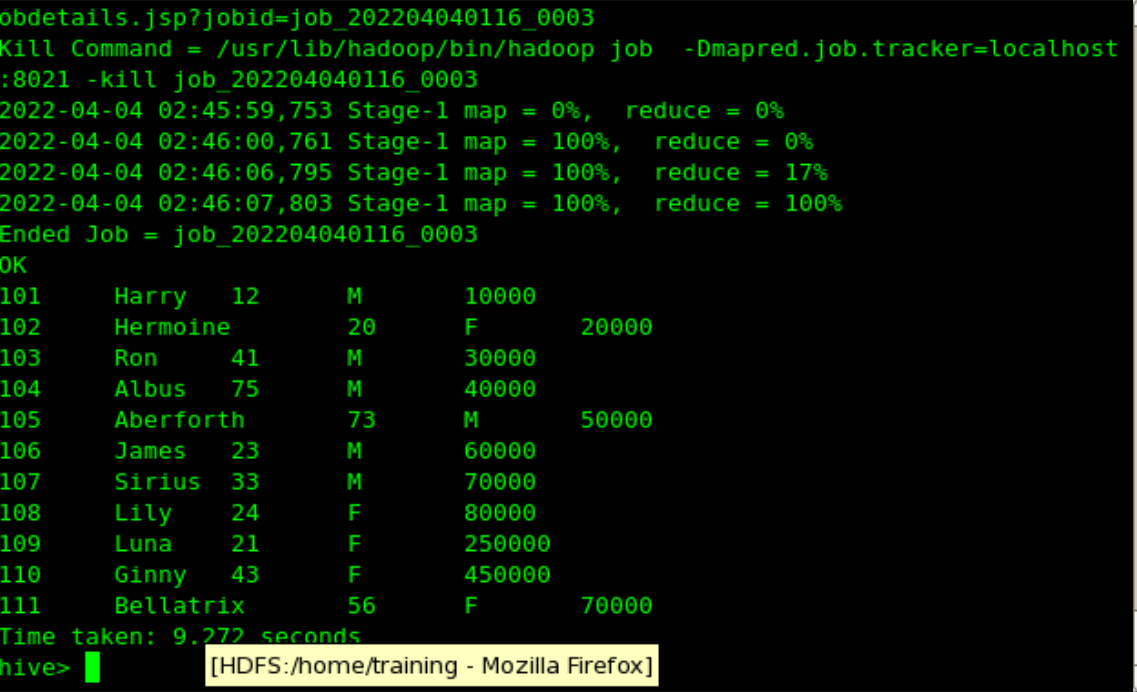




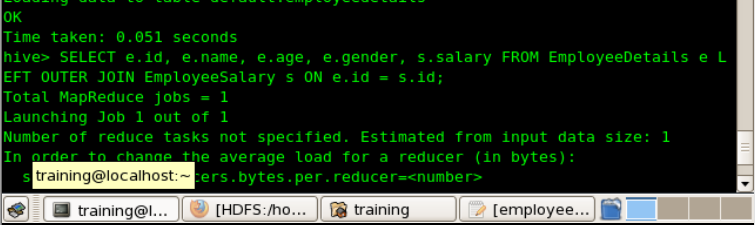


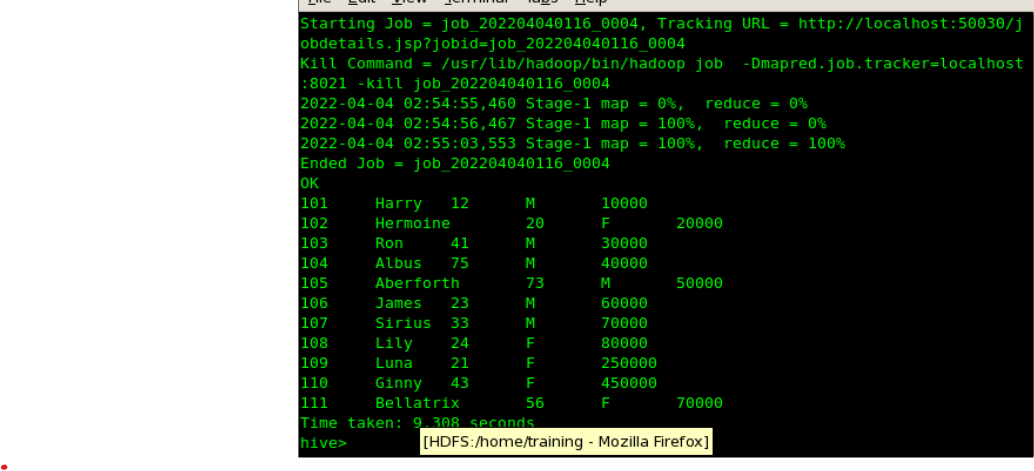
JOINS



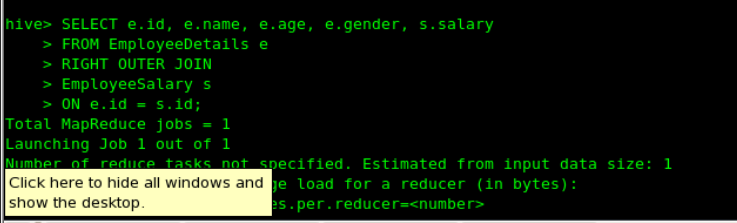


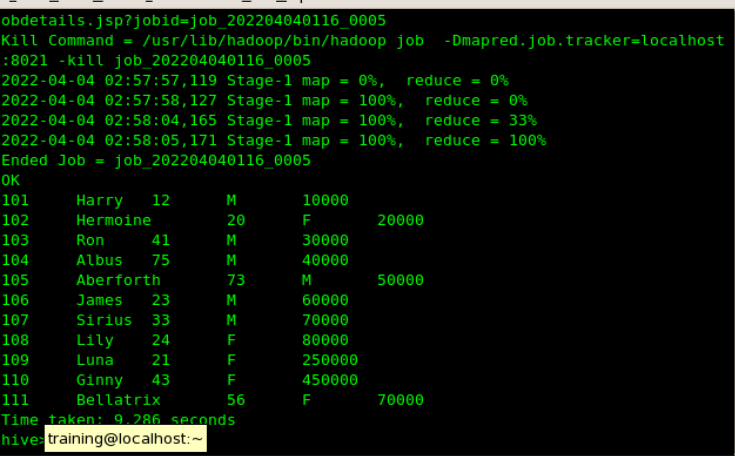
LEFT OUTER JOIN



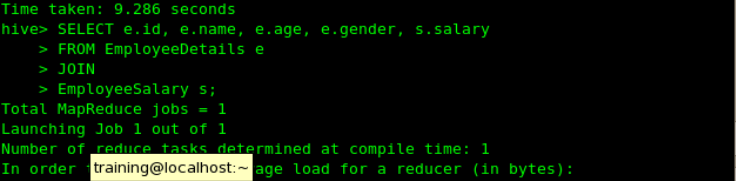


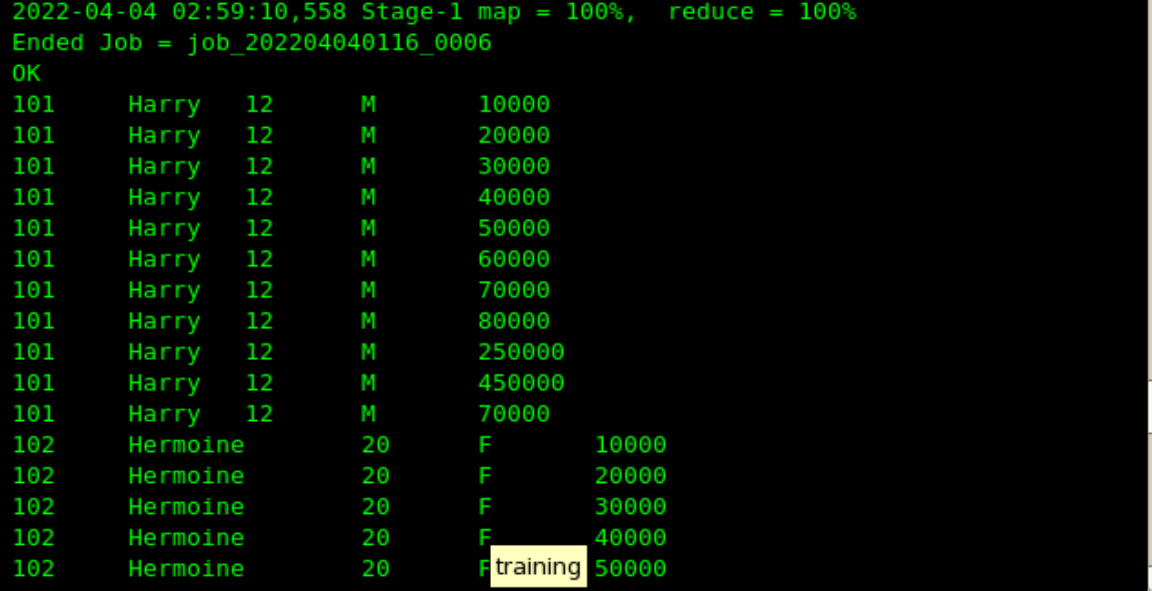
RIGHT OUTER

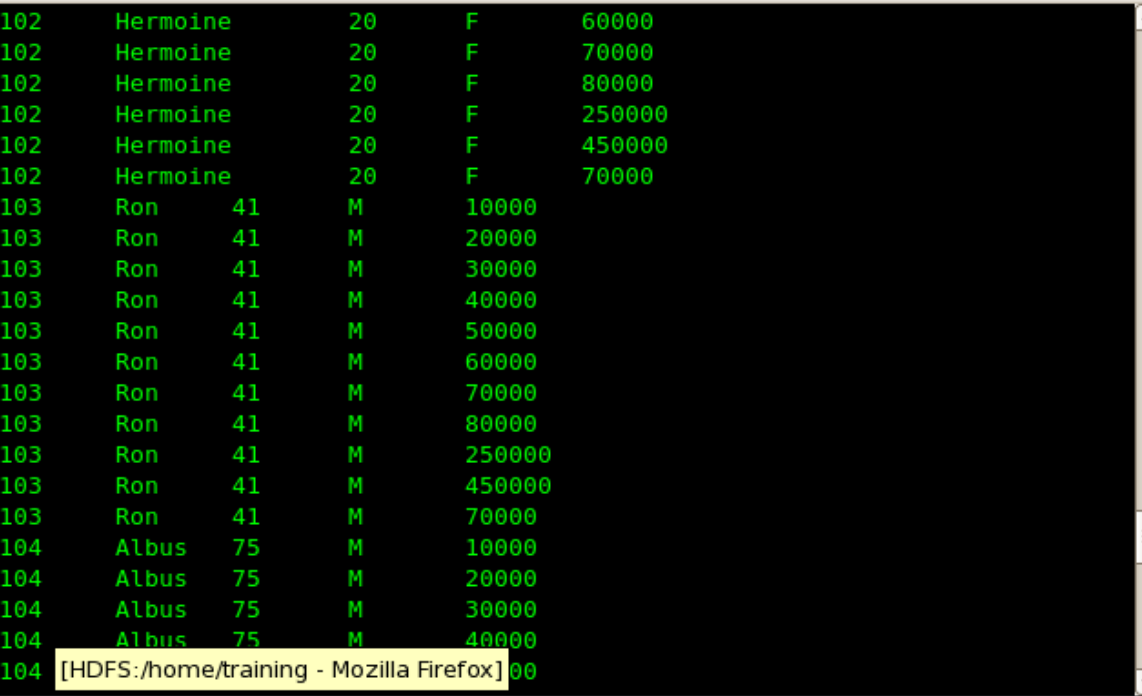


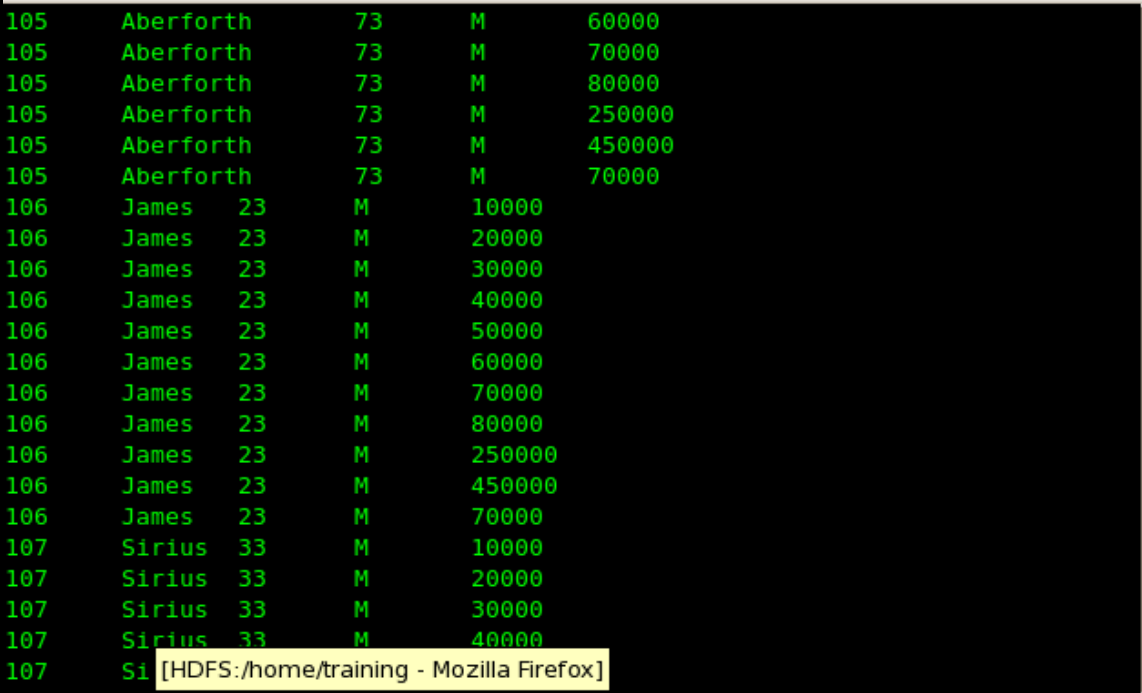


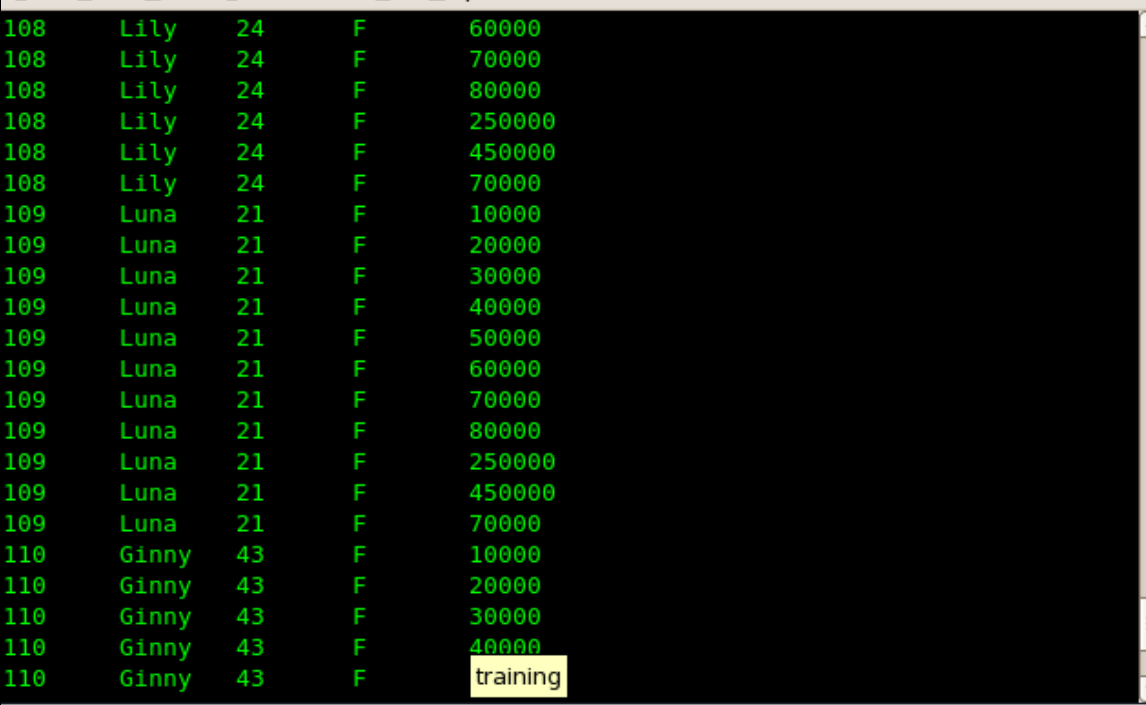
CROSS JOIN

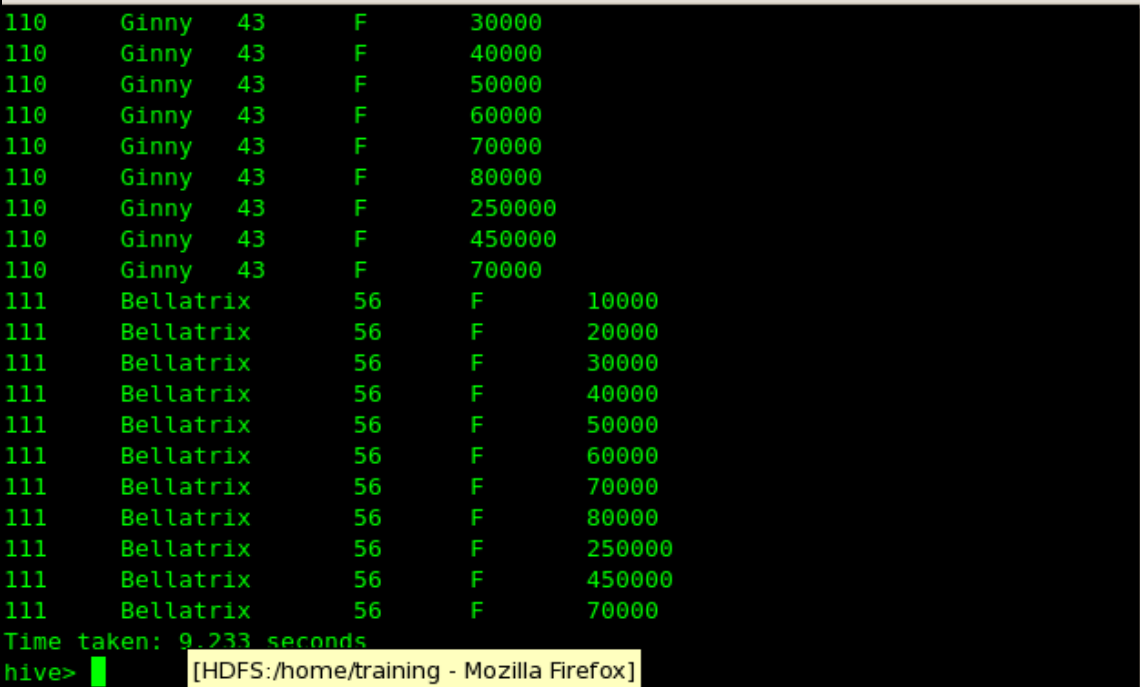












**CONCLUSION**: We have successfully executed HIVE commands to load, insert, retrieve, update, or delete data in the tables.