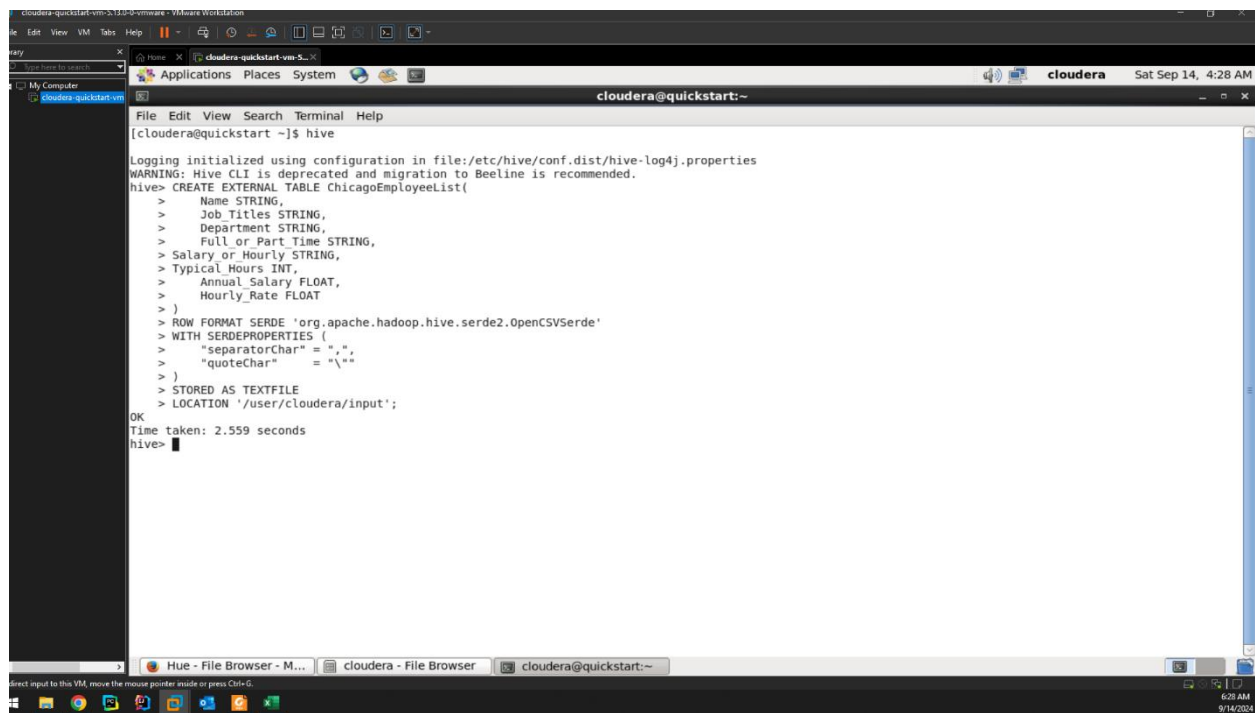
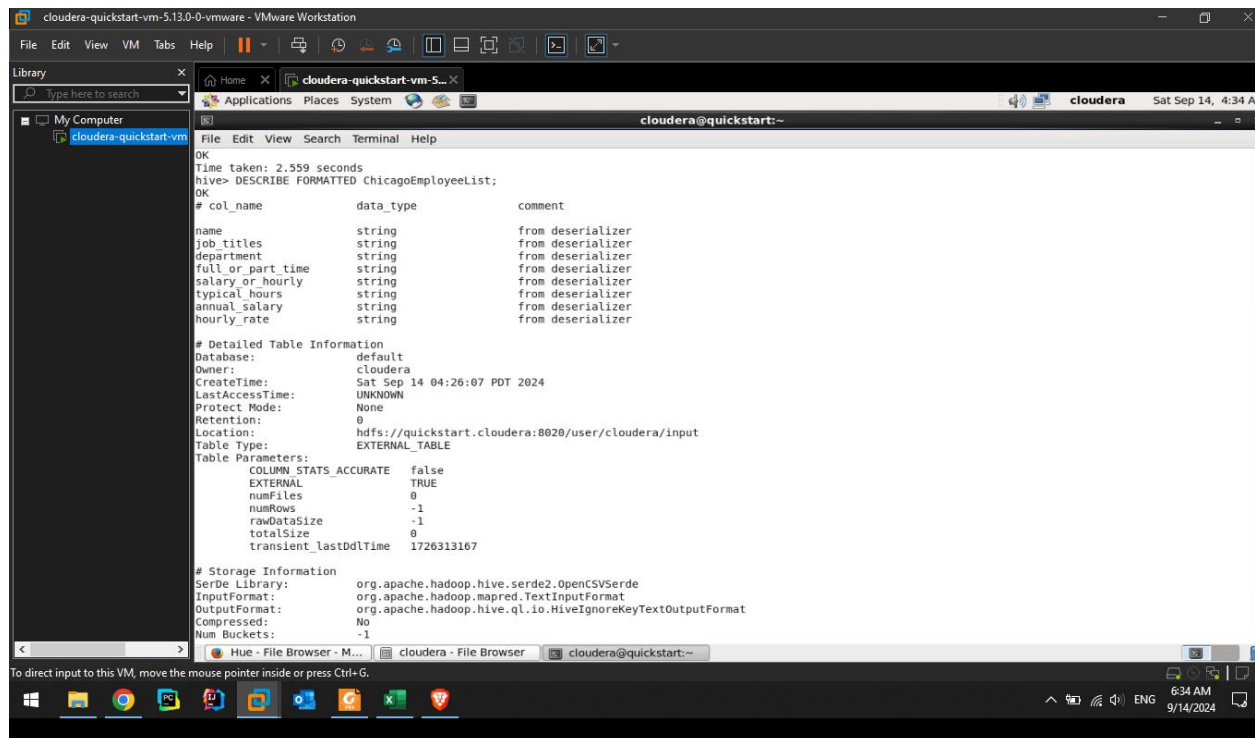


Create EXTERNAL table in Hive for this given sample dataset and find out some interesting facts from this data.



```
cloudera@quickstart:~$ hive
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> CREATE EXTERNAL TABLE ChicagoEmployeeList(
  >   Name STRING,
  >   Job_Titles STRING,
  >   Department STRING,
  >   Full or Part Time STRING,
  >   Salary or Hourly STRING,
  >   Typical_Hours INT,
  >   Annual_Salary FLOAT,
  >   Hourly_Rate FLOAT
  > )
  > ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
  > WITH SERDEPROPERTIES (
  >   "separatorChar" = ",",
  >   "quoteChar" = "\""
  > )
  > STORED AS TEXTFILE
  > LOCATION '/user/cloudera/input';
OK
Time taken: 2.559 seconds
hive>
```

Describe Command:

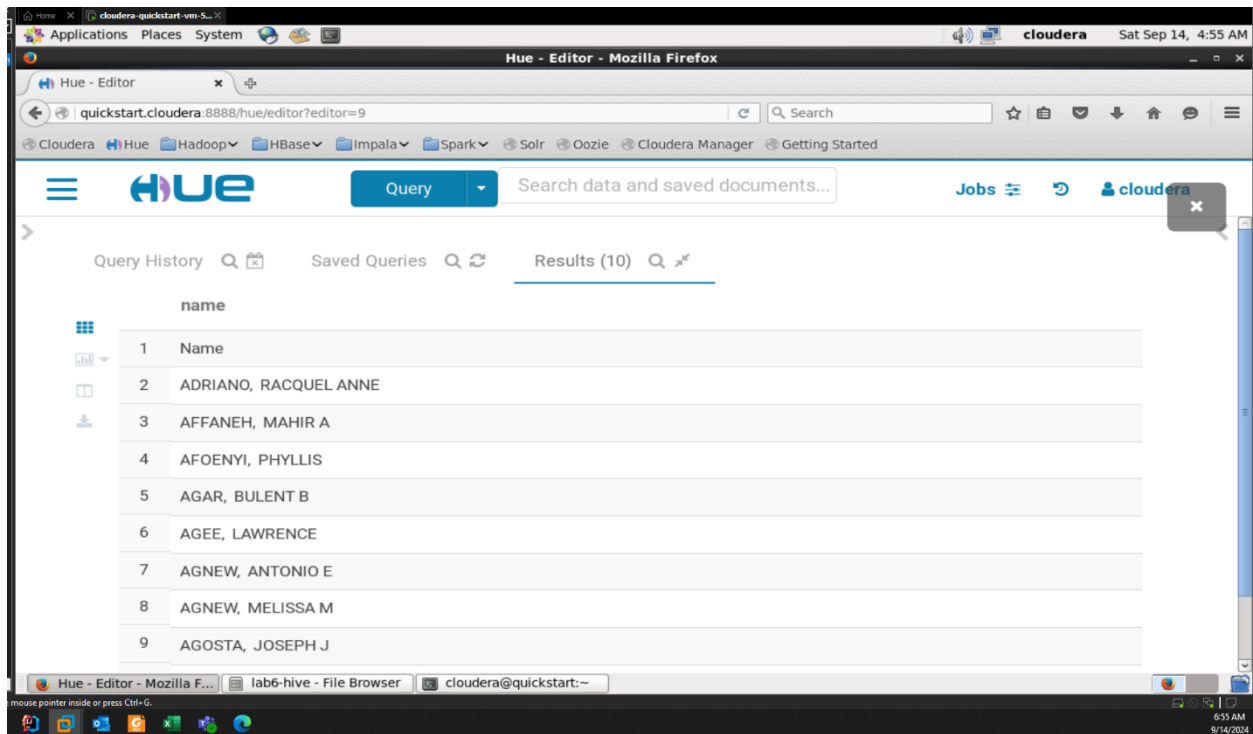
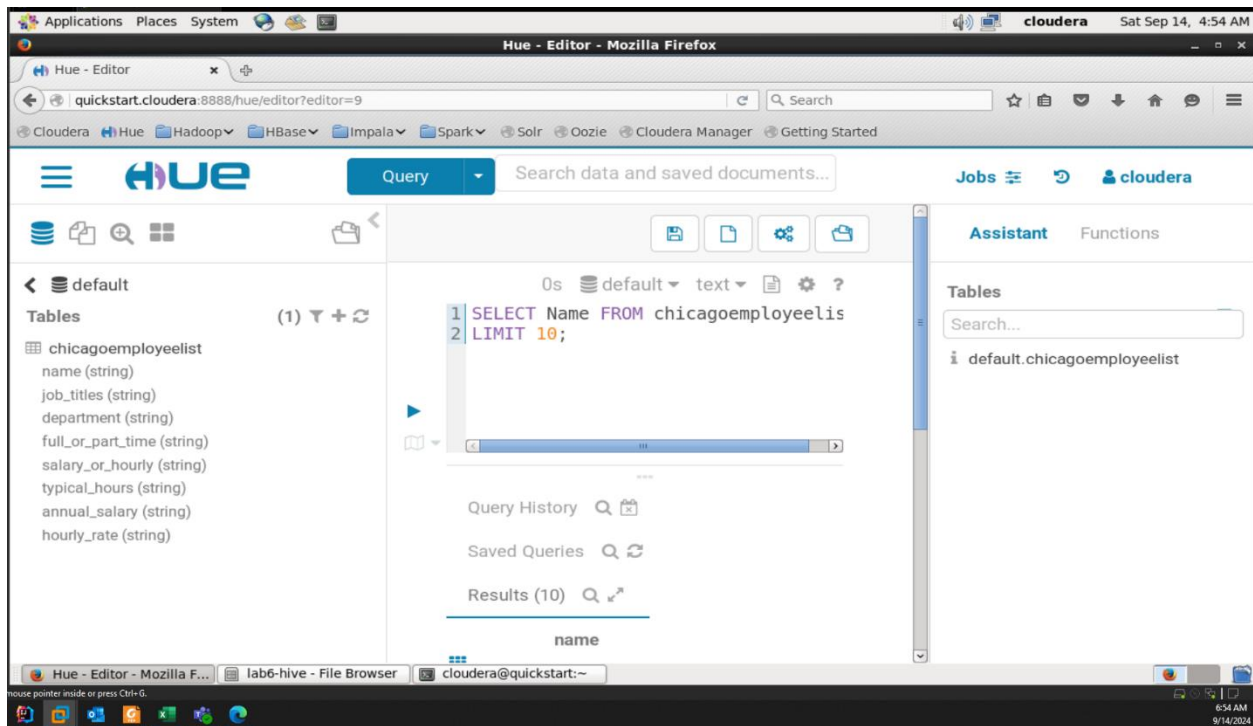


```
OK
Time taken: 2.559 seconds
hive> DESCRIBE FORMATTED ChicagoEmployeeList;
OK
# col_name           data_type           comment
name                 string              from deserializer
job_titles            string              from deserializer
department            string              from deserializer
full or part time    string              from deserializer
salary or hourly     string              from deserializer
typical_hours        string              from deserializer
annual_salary         string              from deserializer
hourly_rate           string              from deserializer

# Detailed Table Information
Database:             default
Owner:                cloudera
CreateTime:           Sat Sep 14 04:26:07 PDT 2024
LastAccessTime:       UNKNOWN
Protect Mode:         None
Retention:            0
Location:             hdfs://quickstart.cloudera:8020/user/cloudera/input
Table Type:           EXTERNAL_TABLE
Table Parameters:
  COLUMN_STATS_ACCURATE false
  EXTERNAL              TRUE
  numFiles              0
  numRows               -1
  rawDataSize           -1
  totalSize             0
  transient_lastDdlTime 1726313167

# Storage Information
Serde Library:         org.apache.hadoop.hive.serde2.OpenCSVSerde
InputFormat:           org.apache.hadoop.mapred.TextInputFormat
OutputFormat:          org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:            No
Num Buckets:           -1
```

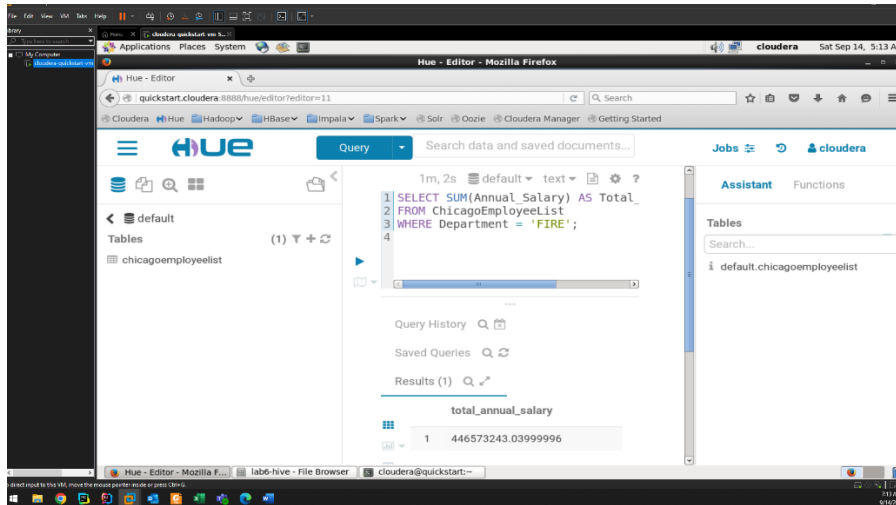
SerDe for this dataset because you need to take care of the header row and the "comma" in name:



Show at least 3 different analyses with “limited” rows.

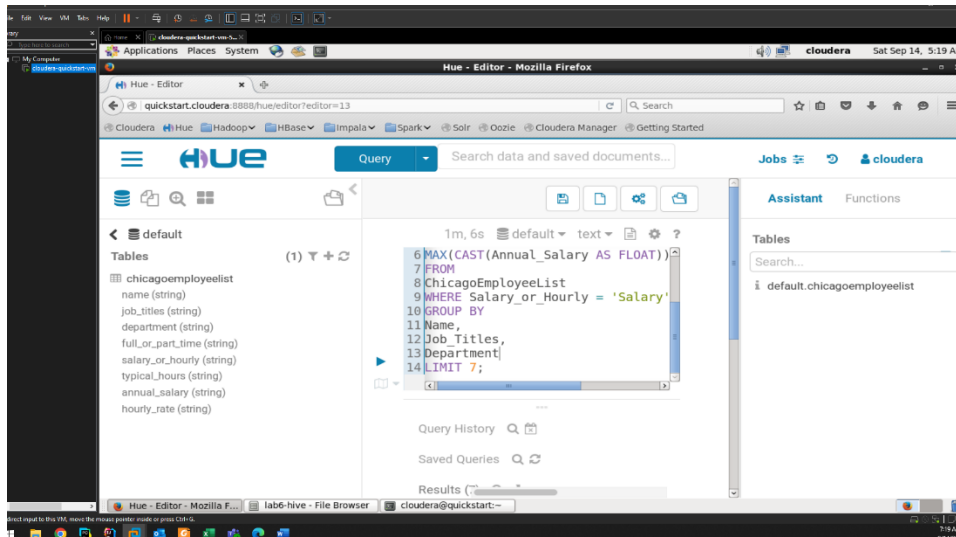
Q_1

Calculates the total sum of annual salaries for employees who work in the 'FIRE' department?



Q_2

displays the names of employees whose salary is Max Salary?



The screenshot shows the Hue web interface in a Mozilla Firefox browser. The URL is `quickstart.cloudera:8888/hue/editor?editor=13`. The interface displays a table with 4 columns: `name`, `job_titles`, `department`, and `max_salary`. The table contains 7 rows of data.

	name	job_titles	department	max_salary
1	AARON, JEFFERY M	SERGEANT	POLICE	111444
2	AARON, KARINA	POLICE OFFICER (ASSIGNED AS DETECTIVE)	POLICE	94122
3	AARON, KIMBERLEI R	CHIEF CONTRACT EXPEDITER	DAIS	118608
4	ABAD JR, VICENTE M	CIVIL ENGINEER IV	WATER MGMNT	117072
5	ABARCA, FRANCES J	POLICE OFFICER	POLICE	48078
6	ABBATE, TERRY M	POLICE OFFICER	POLICE	93354
7	ABBATEMARCO, JAMES J	FIRE ENGINEER-EMT	FIRE	103350

Find the job titles of employees who work part-time (Hourly) and have an Hourly Rate greater than \$40?

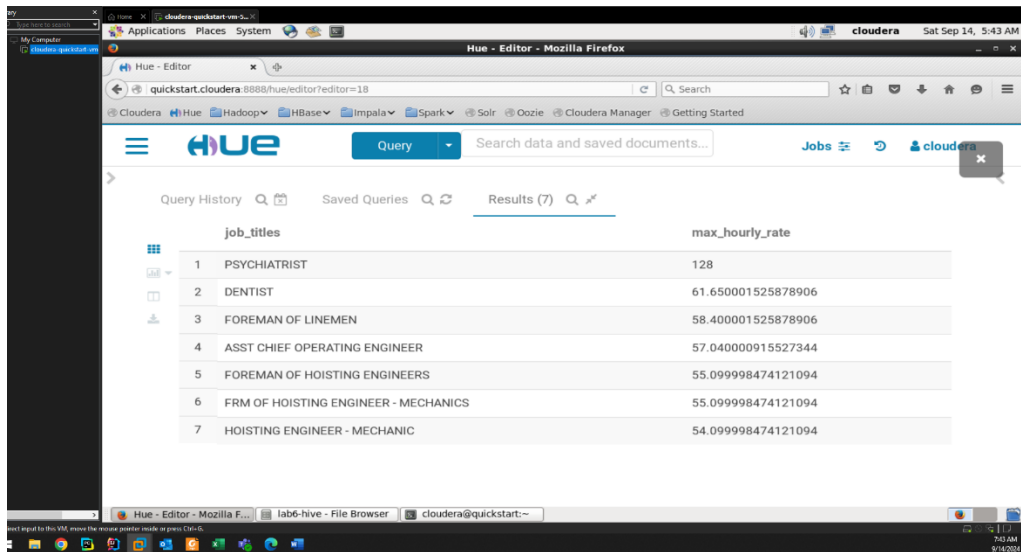
The screenshot shows the Hue web interface with a SQL query being written in the editor. The query is as follows:

```

1 SELECT Job Titles, MAX(CAST(Hourly
2 FROM ChicagoEmployeeList
3 WHERE Salary or Hourly = 'Hourly'
4 AND CAST(Hourly Rate AS FLOAT) > 40
5 GROUP BY Job Titles
6 ORDER BY Max_Hourly_Rate DESC
7 LIMIT 7;

```

The interface also shows a sidebar with a table named `chicagoemployeeelist` with columns: `name (string)`, `job_titles (string)`, `department (string)`, `full_or_part_time (string)`, `salary_or_hourly (string)`, `typical_hours (string)`, `annual_salary (string)`, and `hourly_rate (string)`.



Create dynamic partitions based on the “Salary or Hourly” column.

