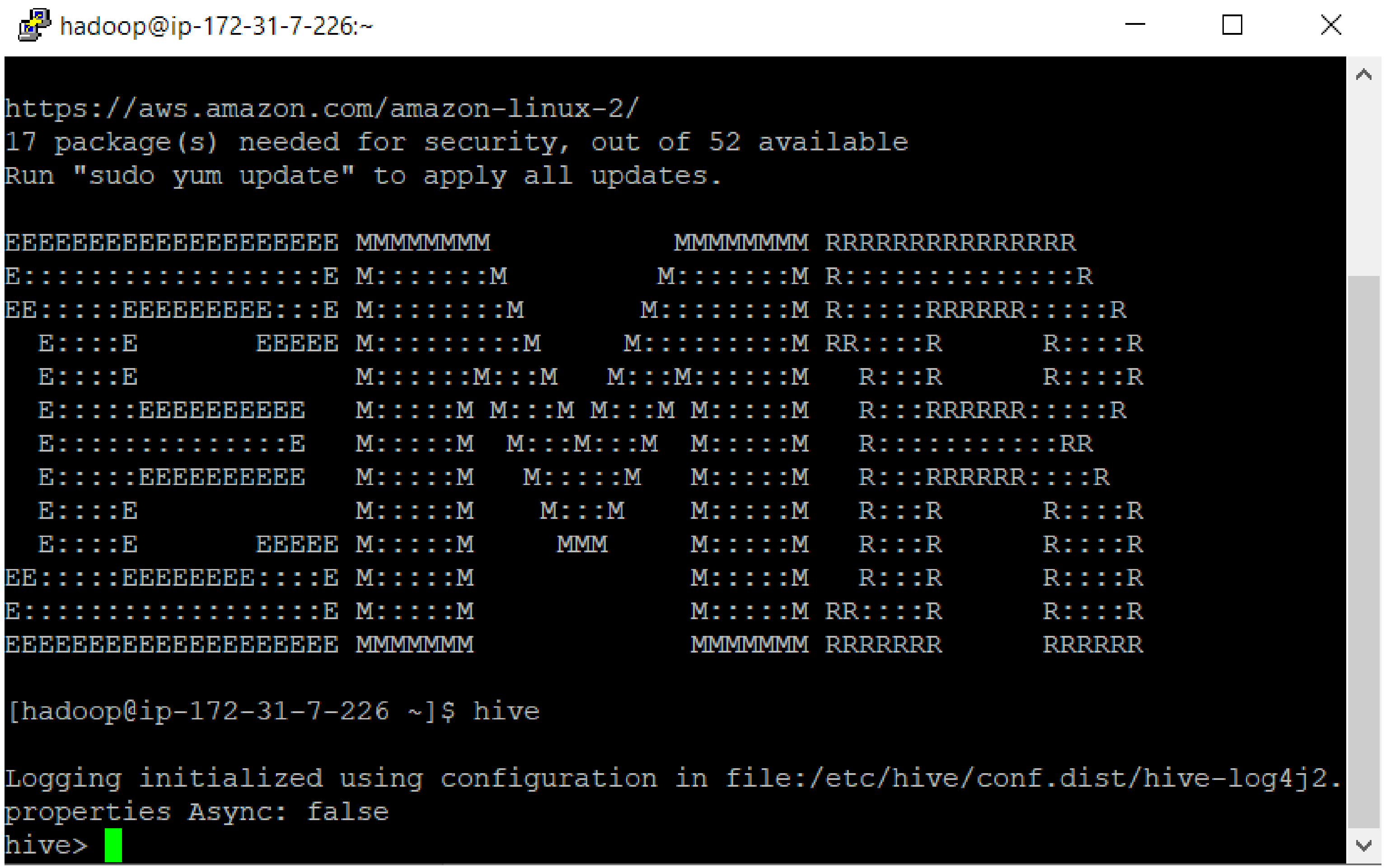
# AWS Hive Hands-On

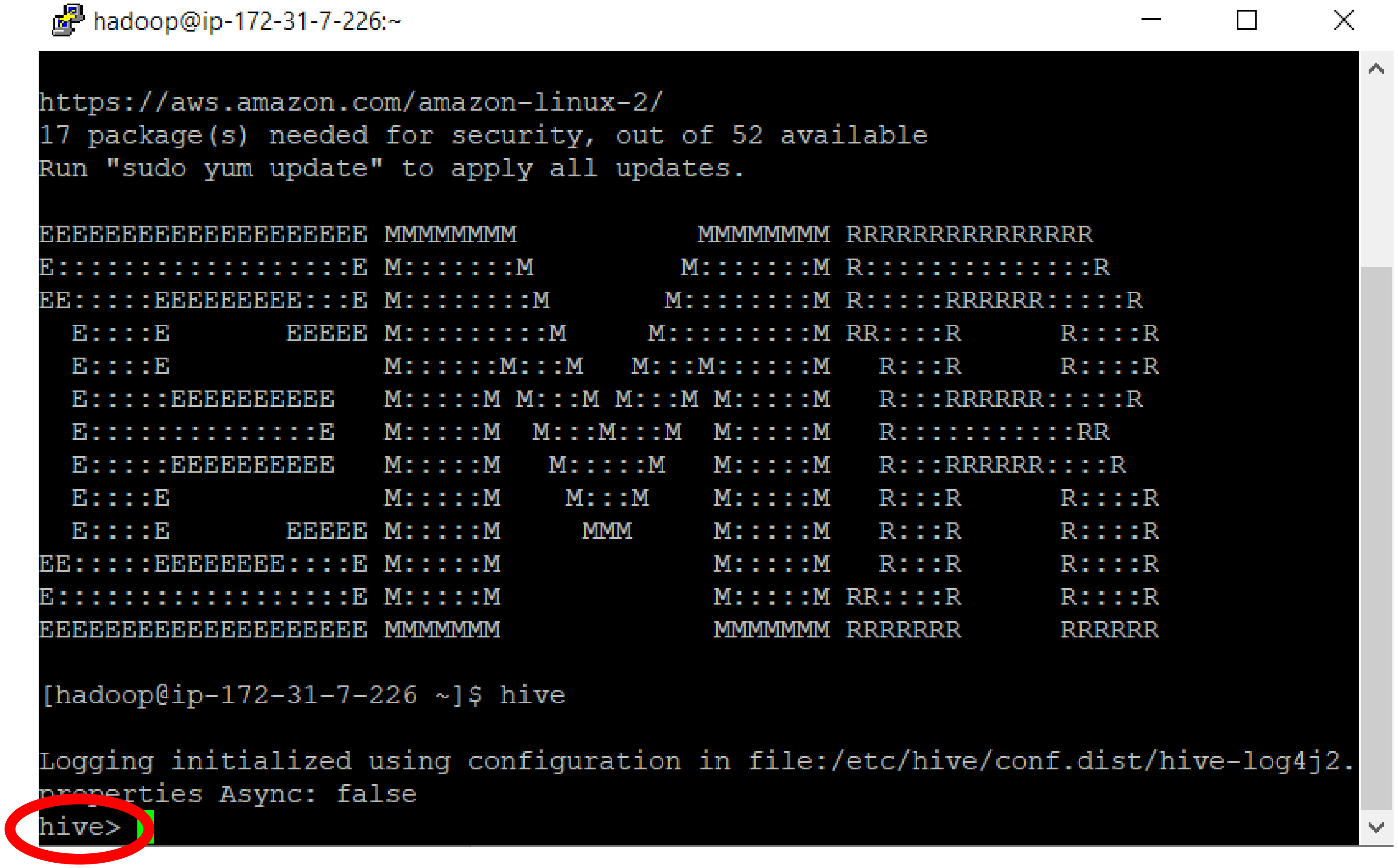
Dr. Villanes

# Once your cluster is running, type **hive**...



```
https://aws.amazon.com/amazon-linux-2/
17 package(s) needed for security, out of 52 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-7-226 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.
properties Async: false
hive>
```

# If you don’t see hive> you didn’t do the last step



```
https://aws.amazon.com/amazon-linux-2/
17 package(s) needed for security, out of 52 available
Run "sudo yum update" to apply all updates.

EEEEEEEEEEEEEEEEEEEE MMMMMMMM           MMMMMMMM RRRRRRRRRRRRRRR
E::::::::::::::::::E M:::::::M         M:::::::M R::::::::::::::R
EE:::::EEEEEEEEE:::E M::::::::M       M::::::::M R:::::RRRRRR:::::R
  E::::E       EEEEE M:::::::::M     M:::::::::M RR::::R      R::::R
  E::::E             M::::::M:::M   M:::M::::::M   R:::R      R::::R
  E:::::EEEEEEEEEE   M:::::M M:::M M:::M M:::::M   R:::RRRRRR:::::R
  E::::::::::::::E   M:::::M  M:::M:::M  M:::::M   R:::::::::::RR
  E:::::EEEEEEEEEE   M:::::M   M:::::M   M:::::M   R:::RRRRRR::::R
  E::::E             M:::::M    M:::M    M:::::M   R:::R      R::::R
  E::::E       EEEEE M:::::M     MMM     M:::::M   R:::R      R::::R
EE:::::EEEEEEEE::::E M:::::M             M:::::M   R:::R      R::::R
E::::::::::::::::::E M:::::M             M:::::M RR::::R      R::::R
EEEEEEEEEEEEEEEEEEEE MMMMMMM             MMMMMMM RRRRRRR      RRRRRR

[hadoop@ip-172-31-7-226 ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j2.
properties Async: false
hive>
```

# Hive

# WARNING!

- When trying to copy-paste from the PDF to the terminal, you might want to use a notepad in between to verify that all the commands were copied correctly. Your queries might not run because you made a mistake while copying-pasting.

The objective of this first portion is to calculate the **sum of hours and miles logged for each driver** using drivers.csv and timesheet.csv

# Hands on
# Define a Hive Table

# Step 1: Creating Table drivers

- First, create a Hive table:

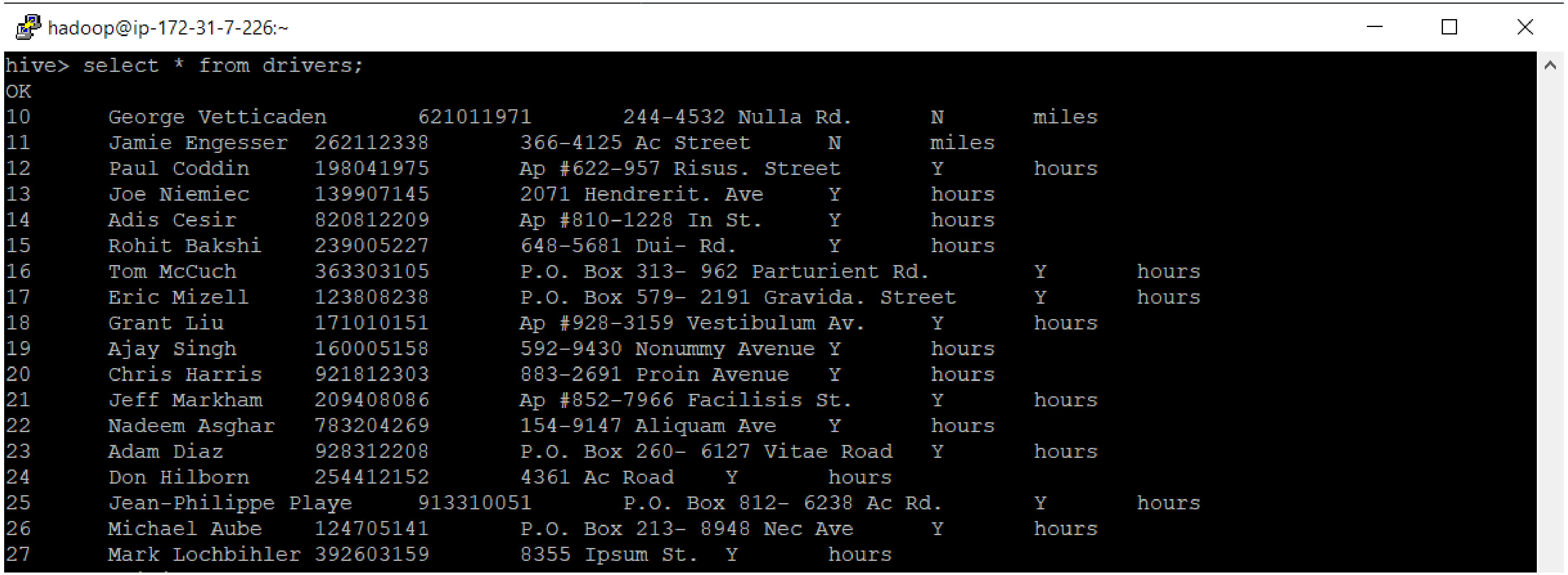
```
>> CREATE TABLE drivers (driverId INT, name STRING, ssn BIGINT,
location STRING, certified STRING, wageplan STRING) row format
delimited fields terminated by ',';
```

- Next, load the data file (drivers.csv) into the table drivers using the following query:

```
>> LOAD DATA INPATH 's3a://bucket_name/drivers.csv' OVERWRITE
INTO TABLE drivers;
```

Make sure to write your bucket_name in the query

Once the query executes, you can query the drivers table:



```
hive> select * from drivers;
OK
10      George Vetticaden       621011971       244-4532 Nulla Rd.      N       miles
11      Jamie Engesser  262112338       366-4125 Ac Street      N       miles
12      Paul Coddin     198041975       Ap #622-957 Risus. Street       Y       hours
13      Joe Niemiec     139907145       2071 Hendrerit. Ave     Y       hours
14      Adis Cesir      820812209       Ap #810-1228 In St.     Y       hours
15      Rohit Bakshi    239005227       648-5681 Dui- Rd.       Y       hours
16      Tom McCuch      363303105       P.O. Box 313- 962 Parturient Rd.        Y       hours
17      Eric Mizell     123808238       P.O. Box 579- 2191 Gravida. Street      Y       hours
18      Grant Liu       171010151       Ap #928-3159 Vestibulum Av.     Y       hours
19      Ajay Singh      160005158       592-9430 Nonummy Avenue Y       hours
20      Chris Harris    921812303       883-2691 Proin Avenue   Y       hours
21      Jeff Markham    209408086       Ap #852-7966 Facilisis St.      Y       hours
22      Nadeem Asghar   783204269       154-9147 Aliquam Ave    Y       hours
23      Adam Diaz       928312208       P.O. Box 260- 6127 Vitae Road   Y       hours
24      Don Hilborn     254412152       4361 Ac Road    Y       hours
25      Jean-Philippe Playe     913310051       P.O. Box 812- 6238 Ac Rd.       Y       hours
26      Michael Aube    124705141       P.O. Box 213- 8948 Nec Ave      Y       hours
27      Mark Lochbihler 392603159       8355 Ipsum St.  Y       hours
```

By default, Hive doesn’t show column names. If you want to show column names, submit:

*hive> set hive.cli.print.header=true;*

Try it now:

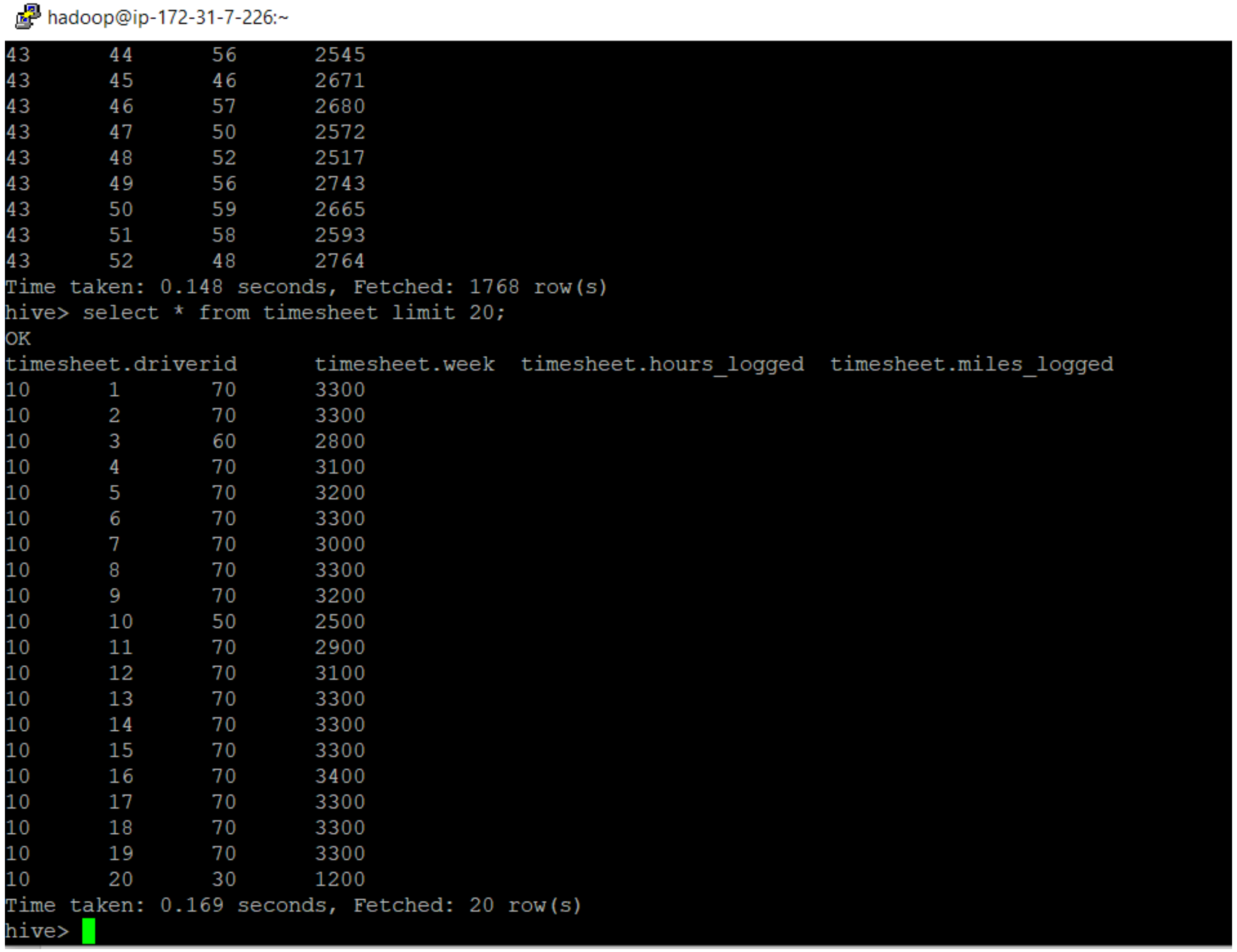
*hive> select * from drivers;*

# Repeat the process but with timesheet.csv

- Create a table called **timesheet**, then load the sample **timesheet.csv** file. Type the following queries one by one:

>> CREATE TABLE timesheet (driverId INT, week INT, hours_logged INT , miles_logged INT) row format delimited fields terminated by ',';

>>LOAD DATA INPATH 's3a://*bucket_name*/timesheet.csv' OVERWRITE INTO TABLE timesheet;



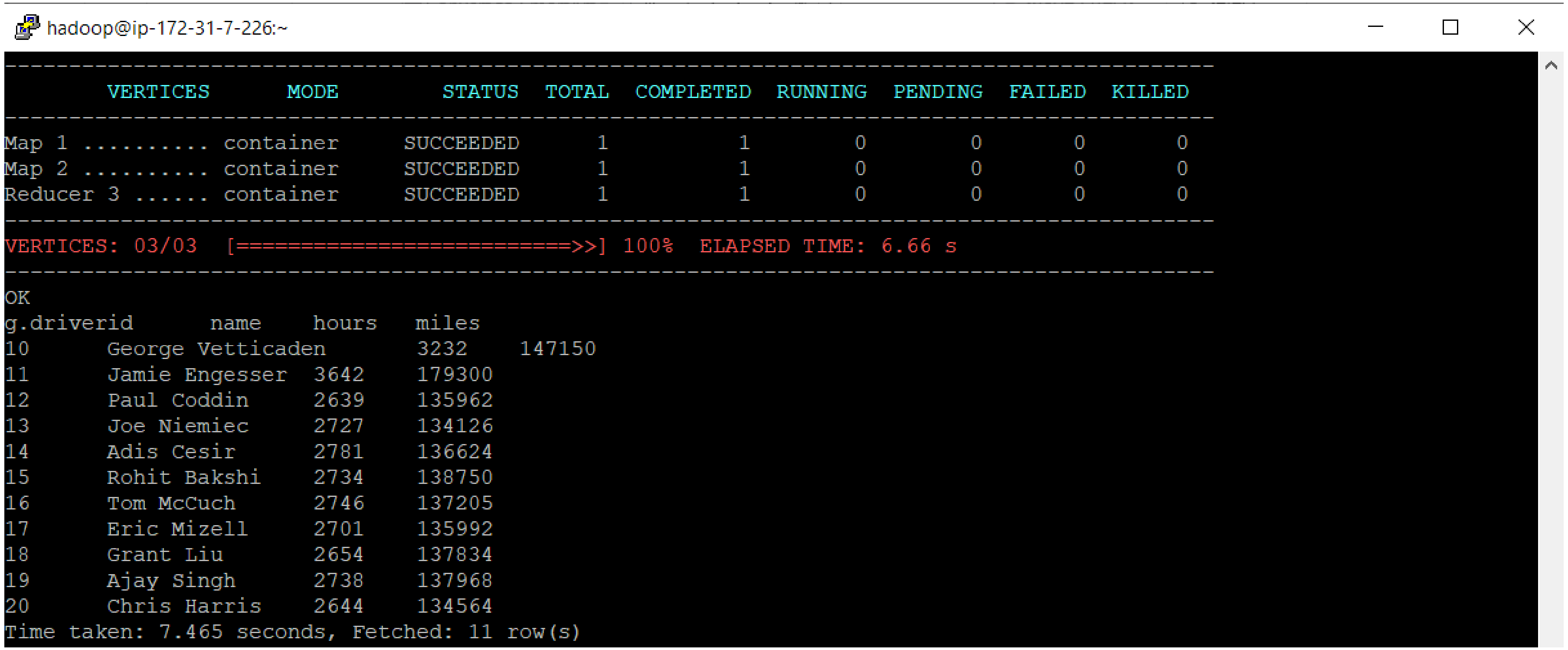
```
43      44      56      2545
43      45      46      2671
43      46      57      2680
43      47      50      2572
43      48      52      2517
43      49      56      2743
43      50      59      2665
43      51      58      2593
43      52      48      2764
Time taken: 0.148 seconds, Fetched: 1768 row(s)
hive> select * from timesheet limit 20;
OK
timesheet.driverid      timesheet.week  timesheet.hours_logged  timesheet.miles_logged
10      1       70      3300
10      2       70      3300
10      3       60      2800
10      4       70      3100
10      5       70      3200
10      6       70      3300
10      7       70      3000
10      8       70      3300
10      9       70      3200
10      10      50      2500
10      11      70      2900
10      12      70      3100
10      13      70      3300
10      14      70      3300
10      15      70      3300
10      16      70      3400
10      17      70      3300
10      18      70      3300
10      19      70      3300
10      20      30      1200
Time taken: 0.169 seconds, Fetched: 20 row(s)
hive>
```

Calculate the sum of hours logged and miles logged for each driver using the tables drivers and timesheet

Display the DriverID, Name, Sum of hours_logged and Sum of Miles_logged. Order your results by DriverID

# Viewing your results:

Your output should look like this. These are the first 11 observations in your output. Based on this, answer the first two questions on the Moodle Quiz.



```
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Map 2 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 3 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 03/03  [==========================>>] 100%  ELAPSED TIME: 6.66 s
----------------------------------------------------------------------------------------------
OK
g.driverid      name    hours   miles
10      George Vetticaden       3232    147150
11      Jamie Engesser  3642    179300
12      Paul Coddin     2639    135962
13      Joe Niemiec     2727    134126
14      Adis Cesir      2781    136624
15      Rohit Bakshi    2734    138750
16      Tom McCuch      2746    137205
17      Eric Mizell     2701    135992
18      Grant Liu       2654    137834
19      Ajay Singh      2738    137968
20      Chris Harris    2644    134564
Time taken: 7.465 seconds, Fetched: 11 row(s)
```

# Questions to be answered in the quiz based on this exercise

1. What is the Sum of Hours Logged for driver Jeff Markham?

2. What is the Sum of Miles logged for driver Jeff Markham?

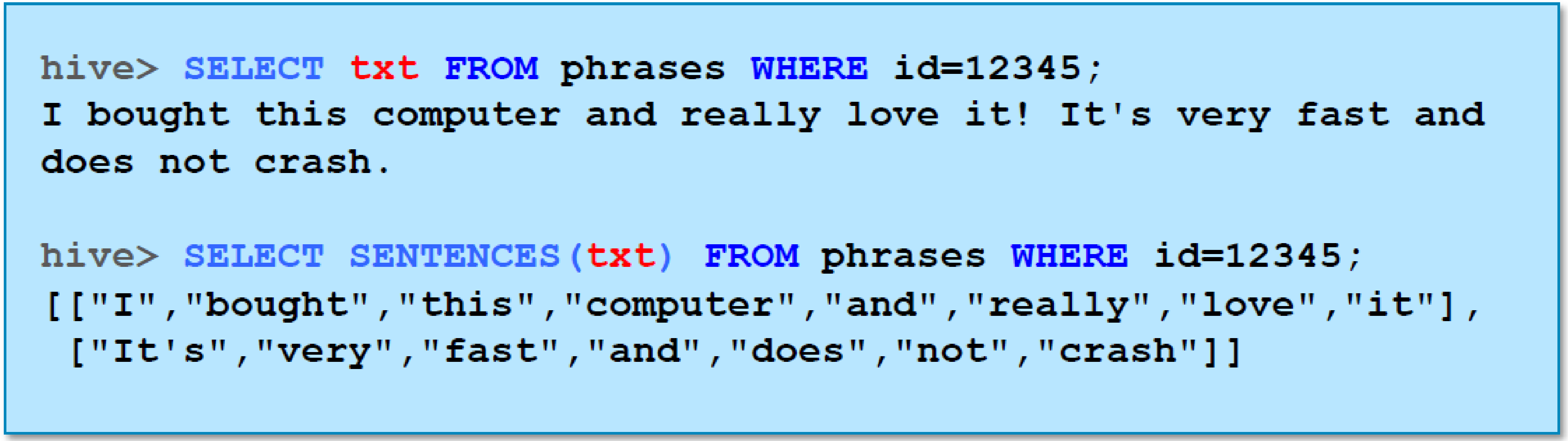
# Hands-On: Text Processing with Hive

# Phrases data setup

```
>> CREATE TABLE phrases (ID BIGINT, txt STRING) row format delimited fields terminated by ',';

>> LOAD DATA INPATH 's3a://bucket_name/Phrases.csv' OVERWRITE INTO TABLE phrases;
```

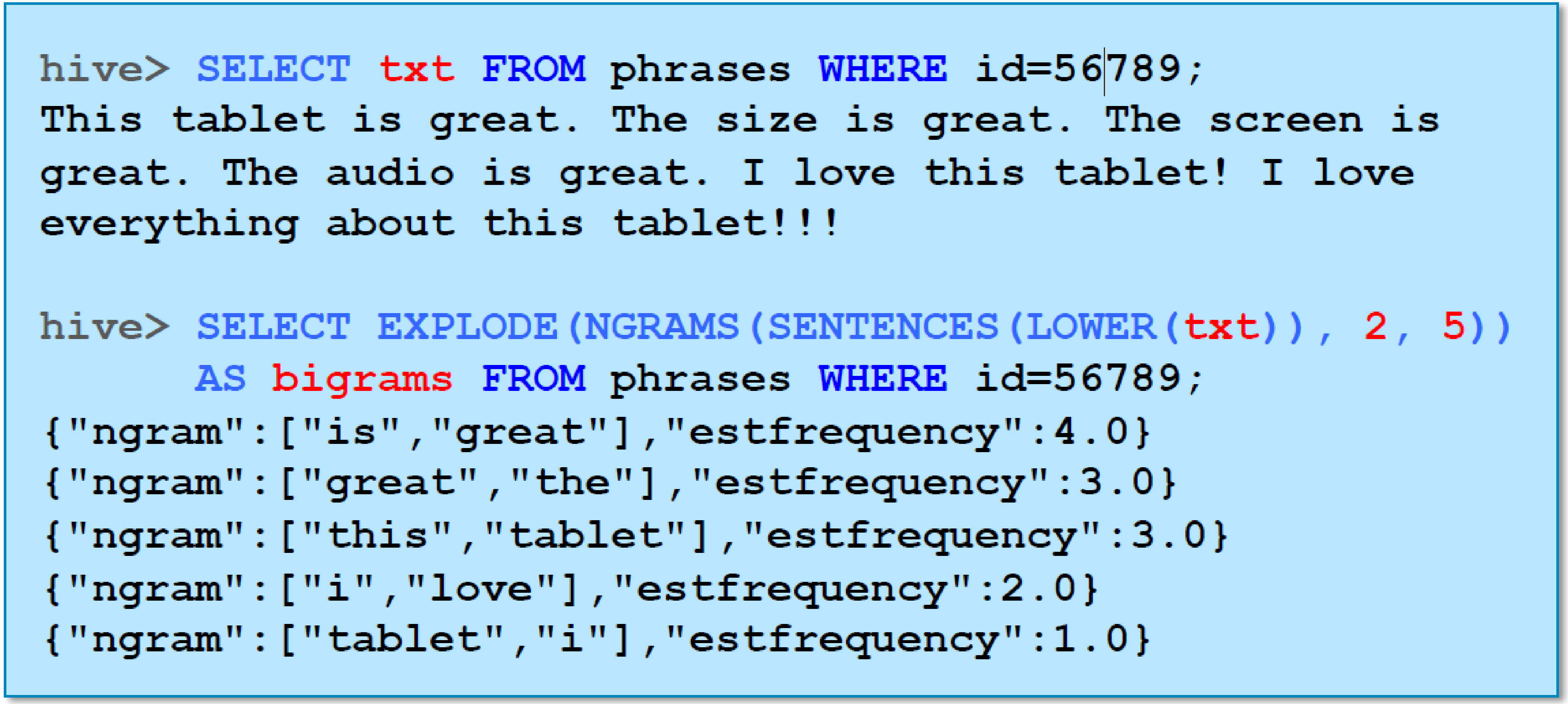
# Parsing Sentences into Words

- The SENTENCES function parses supplied text into words
- Output is a two-dimensional array of strings



```
hive> SELECT txt FROM phrases WHERE id=12345;
I bought this computer and really love it! It's very fast and
does not crash.

hive> SELECT SENTENCES(txt) FROM phrases WHERE id=12345;
[["I","bought","this","computer","and","really","love","it"],
 ["It's","very","fast","and","does","not","crash"]]
```

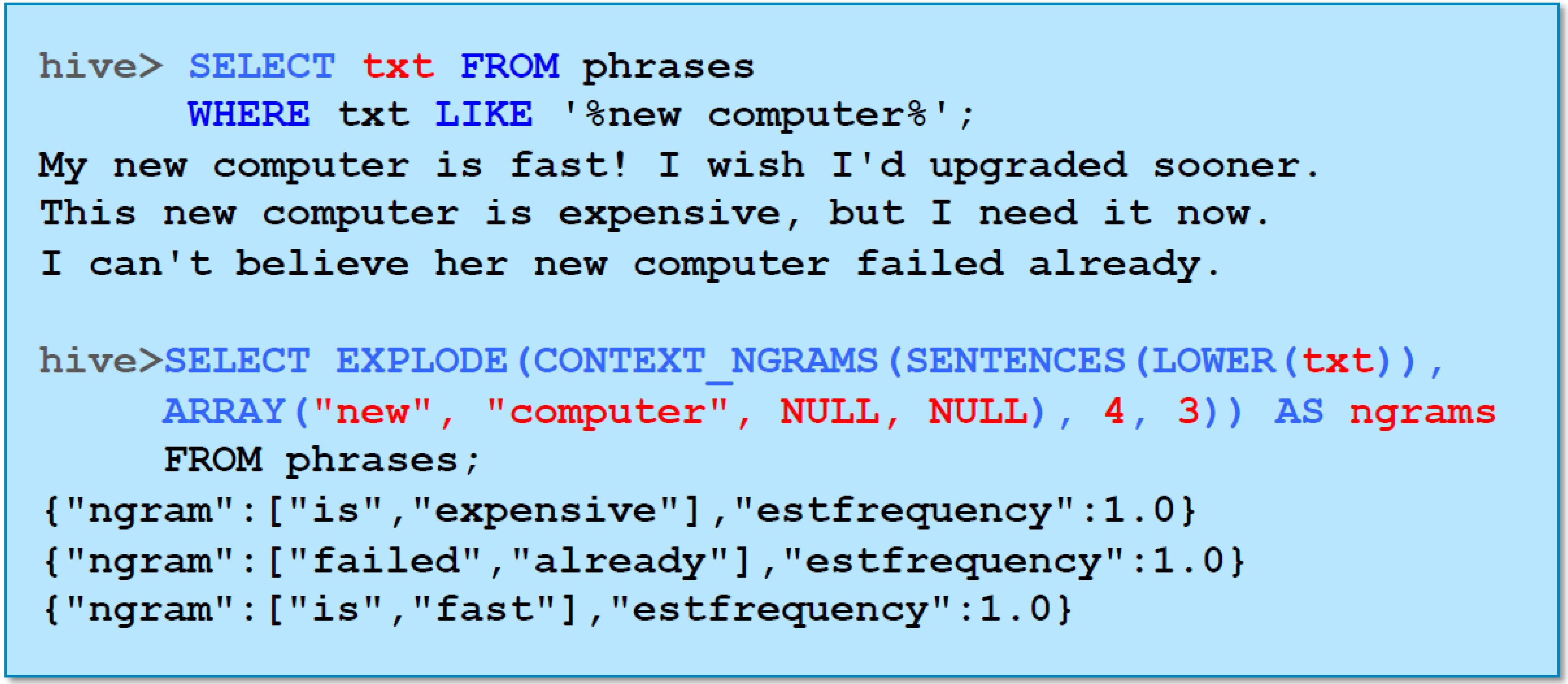
# Calculating n-grams in Hive



```
hive> SELECT txt FROM phrases WHERE id=56789;
This tablet is great. The size is great. The screen is
great. The audio is great. I love this tablet! I love
everything about this tablet!!!

hive> SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(txt)), 2, 5))
      AS bigrams FROM phrases WHERE id=56789;
{"ngram":["is","great"],"estfrequency":4.0}
{"ngram":["great","the"],"estfrequency":3.0}
{"ngram":["this","tablet"],"estfrequency":3.0}
{"ngram":["i","love"],"estfrequency":2.0}
{"ngram":["tablet","i"],"estfrequency":1.0}
```

# Finding specific n-grams

- CONTEXT_NGRAMS is similar, but considers only specific combinations
    - Additional parameter: array of words used for filtering
    - Any NULL values in the array are treated as placeholders



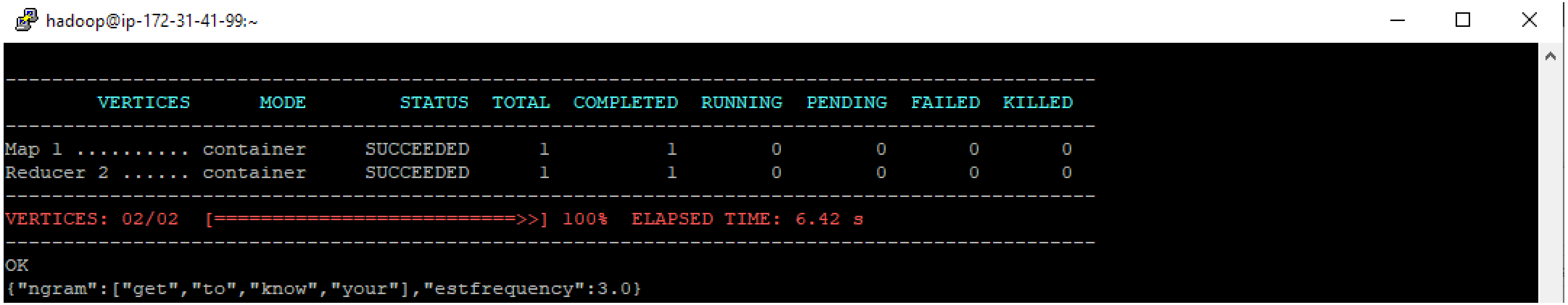
```
hive> SELECT txt FROM phrases
      WHERE txt LIKE '%new computer%';
My new computer is fast! I wish I'd upgraded sooner.
This new computer is expensive, but I need it now.
I can't believe her new computer failed already.

hive>SELECT EXPLODE(CONTEXT_NGRAMS(SENTENCES(LOWER(txt)),
     ARRAY("new", "computer", NULL, NULL), 4, 3)) AS ngrams
     FROM phrases;
{"ngram":["is","expensive"],"estfrequency":1.0}
{"ngram":["failed","already"],"estfrequency":1.0}
{"ngram":["is","fast"],"estfrequency":1.0}
```

# Exercise time – Tips dataset

# Using the TIPS.CSV file…

1. Create a table called tips that has one column: (Tip STRING)
2. Load the TIPS.CSV file into the table you created in step 1
3. Run the following query:

select explode(ngrams(sentences(lower(tip)),4,30)) as ngrams from tips;



```
----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED      1          1        0        0       0       0
Reducer 2 ...... container     SUCCEEDED      1          1        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 02/02  [==========================>>] 100%  ELAPSED TIME: 6.42 s
----------------------------------------------------------------------------------------------
OK
{"ngram":["get","to","know","your"],"estfrequency":3.0}
```

# Question to be answered in the quiz based on this exercise

3. What is the estimated frequency of ["don't","stress","too","much“]?

# Using the tips table…

Run the following query:

```
select
explode(context_ngrams(sentences(lower(tip)),array("learn",NULL,NULL,NULL),4,3)) from tips;
```

# Question to be answered in the quiz based on this exercise

4. What is top result you get?

# Once you are done with the exercises...

- Services → EMR
- Select your cluster
- Terminate