# CIND 719

# Big Data Analytics Tools

# Midterm Questions

3 Questions, 20 Points, 2 Hours.

* Answer Question 1 & 2 on this document.
* Attempt Question 3 on the VM and put the answers on this document.
  + Take the snapshots of your code and insert in this document in appropriate place.
  + Copy the code and paste in this document.
  + Explain your answers in your own words.
* If any question seems unclear to you, then make an assumption, explain your assumption clearly in the answer, and solve the question accordingly.

**Question 1 (6 Point).**

Here is a setup of Data Centers, Racks, and Nodes of a HDFS processing facility:

|  |  |
| --- | --- |
| Data Center 1 (in US) | Data Center 2 (in Europe) |
| - Rack 1 (contains 10 notes numbered as N1, N2, .., N10)  - Rack 2 (contains 10 notes numbered as N1, N2, .., N10)  - Rack 3 (contains 10 notes numbered as N1, N2, .., N10) | - Rack 1 (contains 10 notes numbered as N1, N2, .., N10)  - Rack 2 (contains 10 notes numbered as N1, N2, .., N10)  - Rack 3 (contains 10 notes numbered as N1, N2, .., N10)  - Rack 4 (contains 10 notes numbered as N1, N2, .., N10) |

The nodes are universally addressed with their data center number, rack, and node number. E.g. D1/R2/N5 is the node at Data Center 1 (in US), Rack 2, and Node 5. A Data File is composed of 4 blocks: B1, B2, B3, B4 (each of them 64GB). The HDFS system has a replication factor of 3. Here are the data file block locations with replication factor of 3.

|  |  |  |  |
| --- | --- | --- | --- |
| B1 | B2 | B3 | B4 |
| - D1/R1/N3  - D1/R1/N8  - D2/R2/N7 | - D1/R2/N4  - D1/R2/N9  - D2/R2/N8 | - D1/R3/N2  - D1/R3/N5  - D2/R3/N9 | - D2/R1/N3  - D2/R1/N5  - D2/R3/N5 |

A MapReduce task using B1, B2, B3, B4 needs to be run. The available processors for the Map operation are the following nodes:

|  |  |  |  |
| --- | --- | --- | --- |
| P1 | P2 | P3 | P4 |
| D1/R1/N3 | D1/R1/N7 | D2/R4/N2 | D1/R3/N8 |

HDFS tries to associate the lowest distance between the processors to the data nodes. Therefore, give a potential minimum distance association list of processors to the data blocks. **Hint**: you are expected to fill in the table below with the data block numbers processed by the processor with their locations and their distances between the processor and data block location.

|  |  |  |  |
| --- | --- | --- | --- |
| Processors | Data Block Number | Data Block Location | Distance/Cost |
| P1 (D1/R1/N3) | B1 | D1/R1/N3  D1/R1/N8  D2/R2/N7 | (0 + 2 + 6) = 8 |
| P2 (D1/R1/N7) | B2 | D1/R2/N4  D1/R2/N9  D2/R2/N8 | (4 + 4 + 6) = 14 |
| P3 (D2/R4/N2) | B4 | D2/R1/N3  D2/R1/N5  D2/R3/N5 | (4 + 4 + 4) = 12 |
| P4 (D1/R3/N8) | B3 | D1/R3/N2  D1/R3/N5  D2/R3/N9 | (2 + 2 + 6) = 10 |

**Question 2 (4 Points).**

The following text (from Roberto Bolano, 2666) is sent to a MapReduce to count the words. You can ignore the punctuation marks in the text (e.g., double quotes “”, comma, and point):

"*I felt happy because I saw the others were happy and because I knew I should feel happy, but I wasn’t really happy."*

The text will be processed by a **single Mapper** and the mapper runs **a Combiner** that executes the same task as the Reducer.

**Q2.1** (2 pts) Provide the output of the mapper after processing the text below. Be careful about the order of the output.

I 1 1 1 1 1

Felt 1

Happy 1 1 1 1

Because 1 1

Saw 1

The 1

Others 1

Were 1

And 1

Knew 1

Should 1

Feel 1

But 1

Wasn’t 1

Really 1

**Q2.2** (2 pts) Provide the output of the combiner after completing its task below.

I 5

Felt 1

Happy 4

Because 2

Saw 1

The 1

Others 1

Were 1

And 1

Knew 1

Should 1

Feel 1

But 1

Wasn’t 1

Really 1

**Question 3 (10 Points).**

An analyst is supposed to import the following information into the "diamonds" Hive table. The import of the data will be done from a csv (comma separated value) file.

File schema of the csv file:

* Id: row id for the data. The id contains only integer numbers
* Carat: weight of the TBL
* Cut: quality of the cut (Fair, Good, Very Good, Premium, Ideal)
* Color: diamond color, from J (worst) to D (best)
* Clarity: a measurement of how clear the diamond is (I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF)
* Depth: total depth percentage that is already calculated using the formula -> z / mean(x, y) = 2 \* z / (x + y) (43 - 79)
* Table Width: of top of diamond relative to widest point (43 - 95)
* Price: The sales price of the diamond with the properties given in US dollars ($326-$18,823)
* X: length in mm (0 - 10.74)
* Y: width in mm (0 - 58.9)
* Z: depth in mm (0 - 31.8)

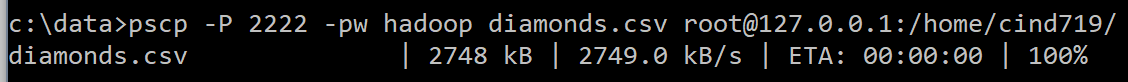
**Important Notes**:

1. The information between the last parenthesis represents the range of values or possible values of these database columns.
2. The data columns given in the csv file may not be in the order given above
3. The csv data file may contain a header row that should not be imported as data.

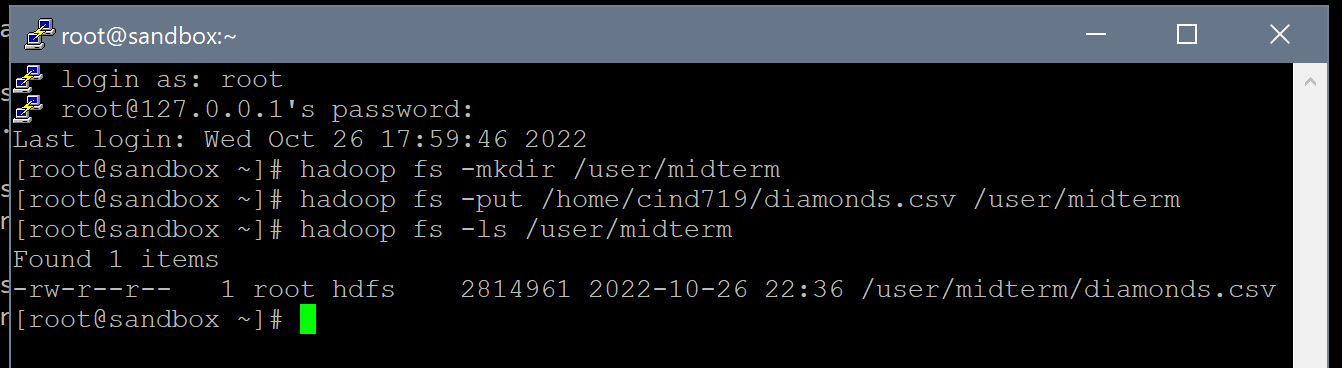
Download the diamonds.csv file from course shell (Lab Resources & Datasets).

**Q 3.1** (2 pts) Transfer the file to HDFS. Provide screenshot of the console showing the file is in HDFS.

Transferring the diamonds.csv file to the HDP access node using PSCP utility:

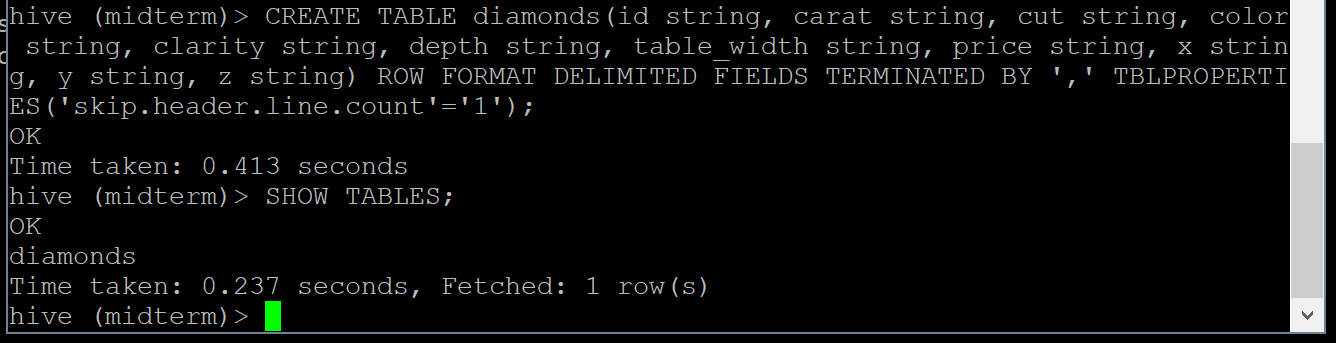


Moving the file to the HDFS in a new directory called ‘midterm’

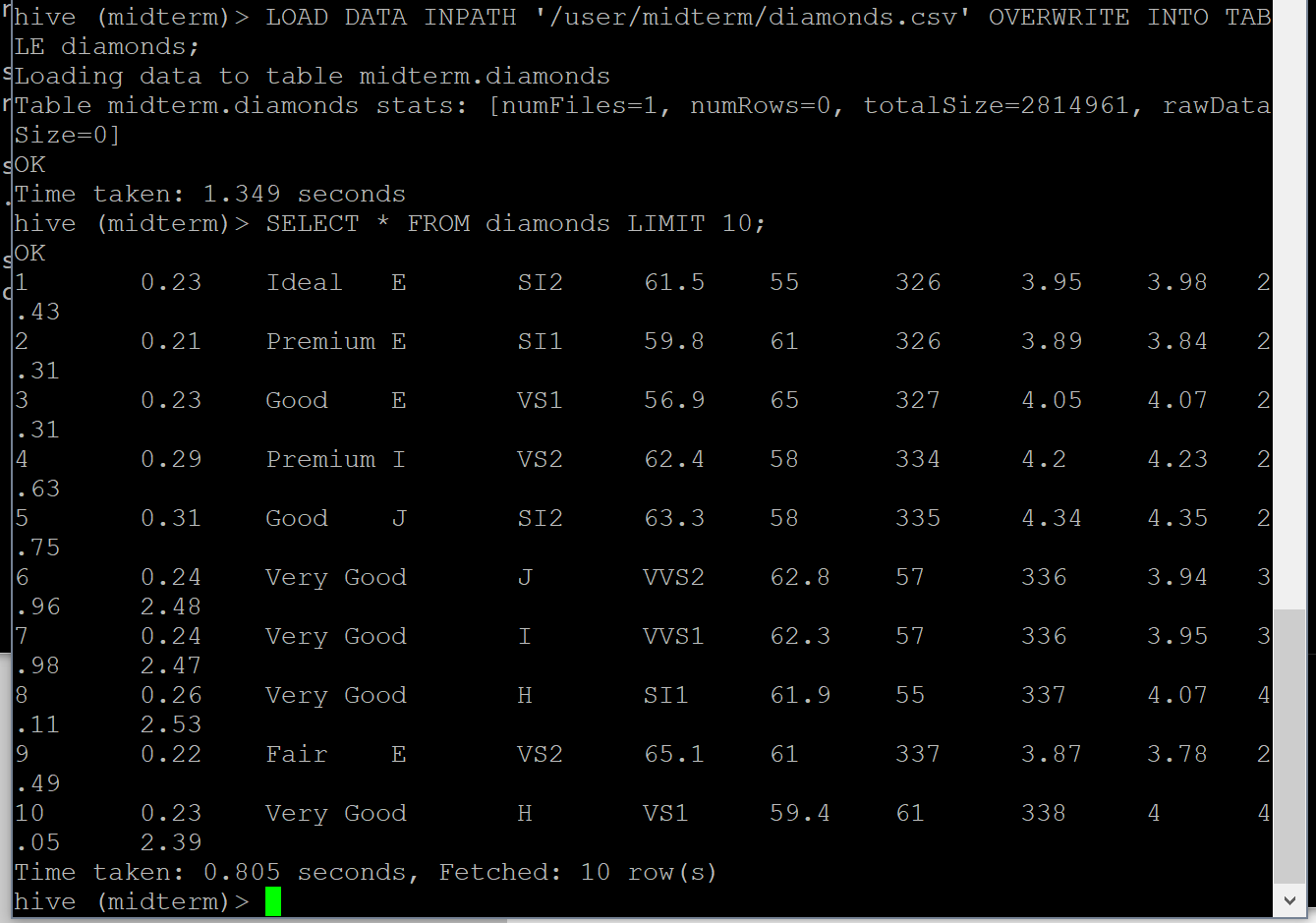


**Q 3.2** (2 pts) Write the script to create the **diamonds** table in Hive. Provide screenshot of the console showing the table creation command and first ten rows of the hive table.

Creating tables with standard datatypes for all columns of the table:



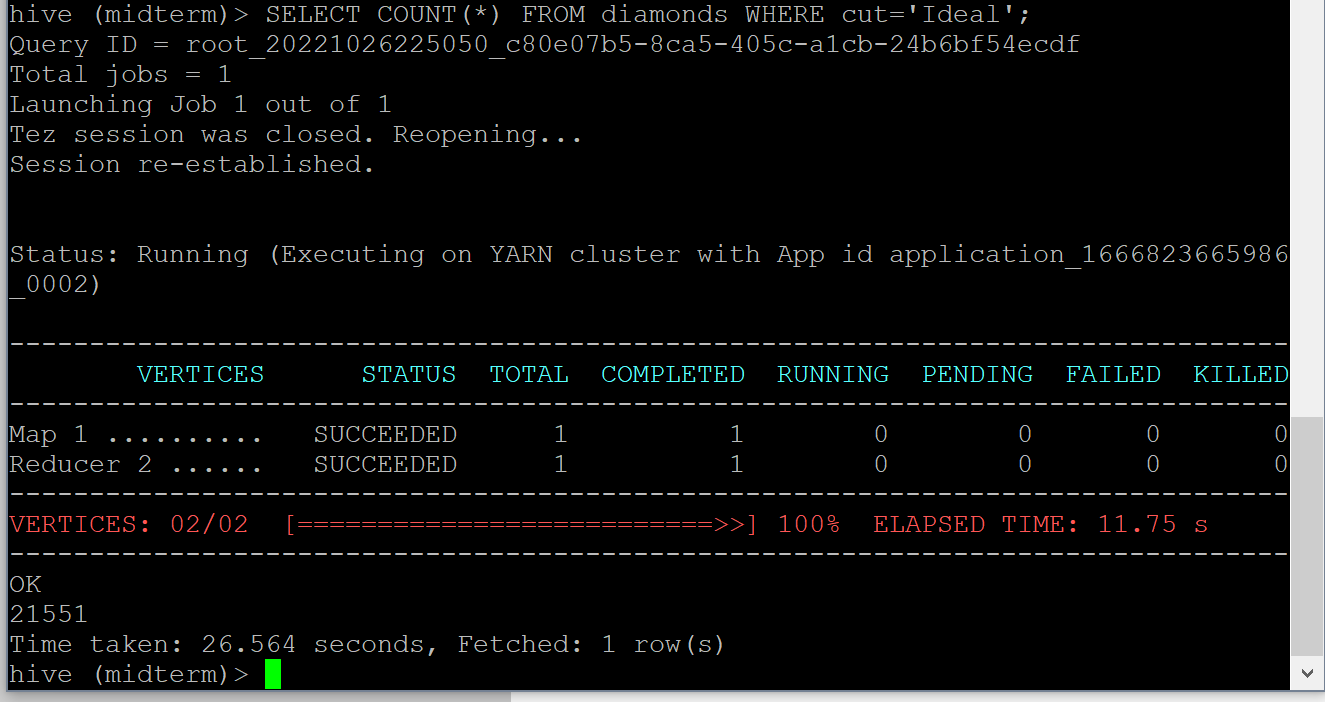
Loading the csv file into the table and displaying the first 10 rows of the table:



**Q 3.3** (3 pts) Write a query to get the total number of diamond rows where the Cut information is "Ideal". Provide screenshot of the console showing the query and the output.

SELECT COUNT(\*) FROM diamonds WHERE cut=’Ideal’;

Using count() will count the number of rows that satisfy the WHERE argument cut = ‘Ideal’. There are 21,551 diamonds in this file with ideal cuts.



**Q 3.4** (3 pts) Write a query to get the top 10 diamonds with the biggest weights (Carats). Provide screenshot of the console showing the query and the output.

SELECT id, carat FROM diamonds ORDER BY carat DESC LIMIT 10;

Displaying the diamond id and the carat weight but only displaying the biggest weights by ordering the carat weights and displaying them in descending order (greatest to smallest). Limiting the output by 10 returns the top 10 biggest diamonds (by weighted carats)

