**Lab Assignment #1**

**-Big Data Manipulation-**

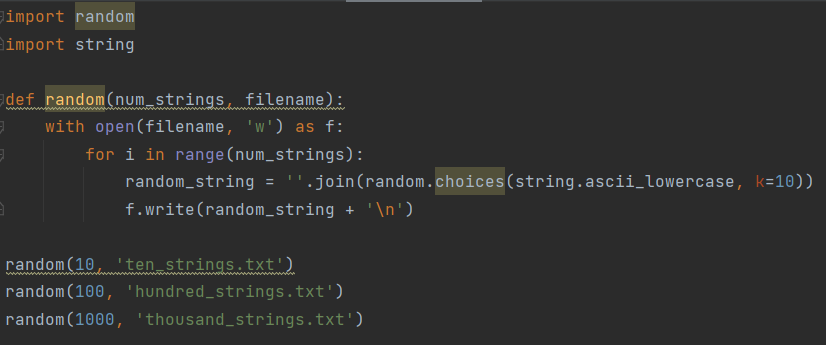
**19102127 ITM**

**Suho Lee**

1. **Make a simple python script to generate a given numbers of strings**

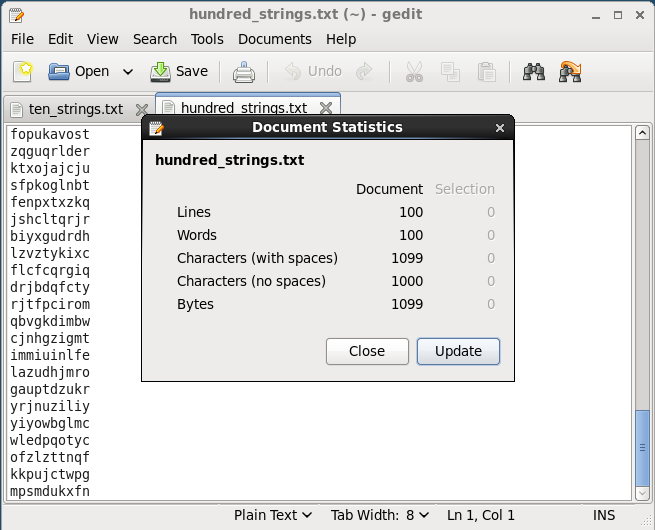
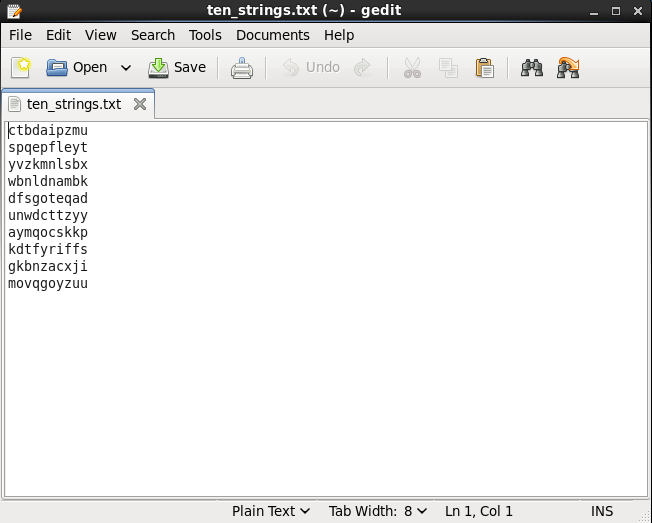
**(one per line)**

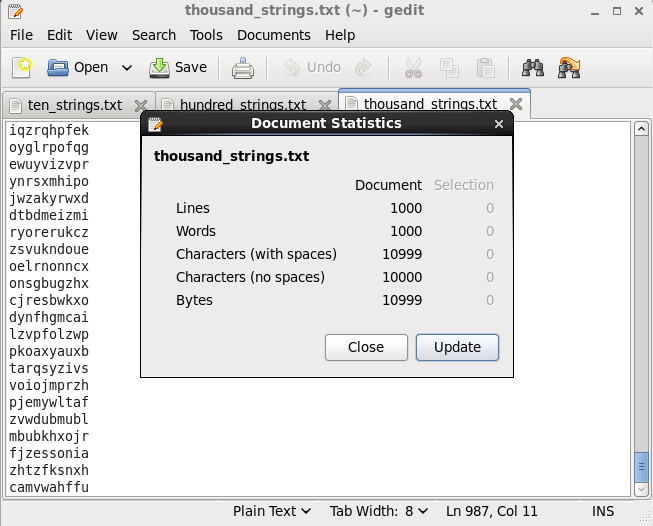
* **Random.py**



* For convenience, It is desirable to make a function that makes random string values.
* In this case, I make a function named ‘random’ and it received two parameters as input.
* Iterates as many times as the 'num\_strings' value, randomly generates 10 characters and writes them to the file one line at a time.

1. **Make three files using the python script to generate random strings with 10, 100, and 1000 number, respectively.**



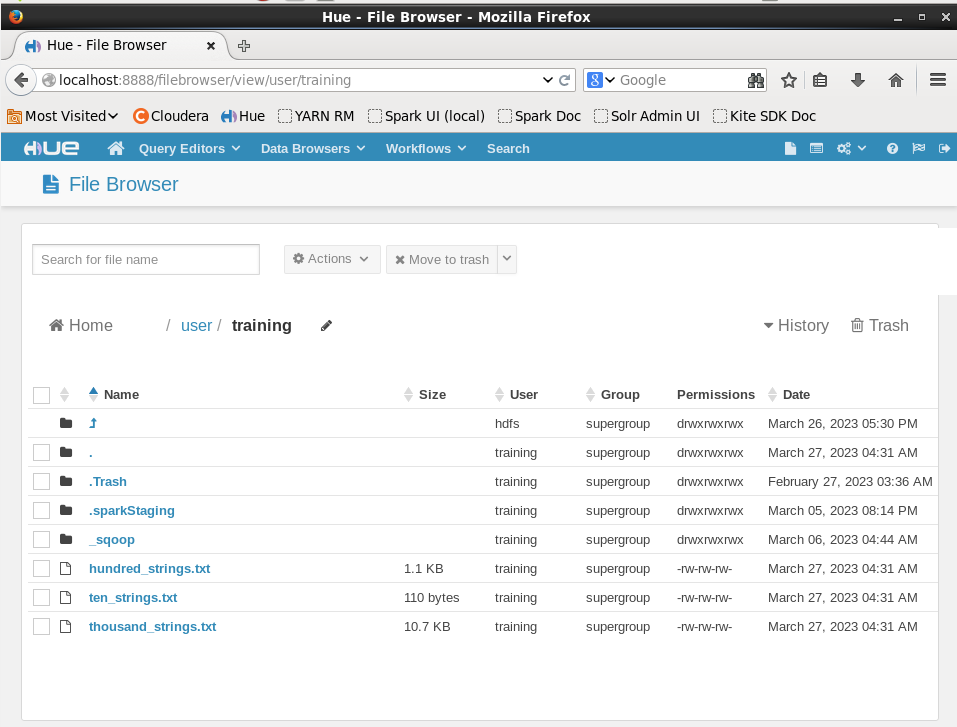


* The three pictures are individual files obtained from 'random.py'.
* You can see that 10, 100, and 1000 random string variables are stored, respectively.

1. **Upload the generated files into HDFS**

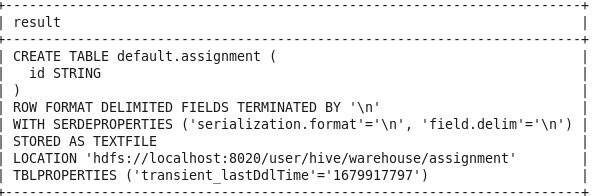
C:\Users\이수호\Desktop\ITM 3-1\DB & WEB\DB&WEB Assignment\3-1. hdfs dfs 명령어.png

* To upload a local text file to the HDFS system, the following command must be executed in the terminal.

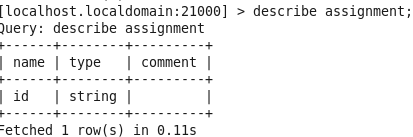


* You can check the result in HUE as well.

1. **Create a table in Impala to store the generated data**

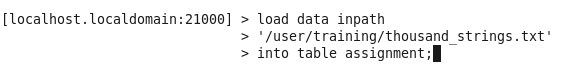


* First, the name of the table is designated as 'assignment'. And, since I handle a text file with 10, 100, and 1000 random string variables, respectively, the column is designated as a 'STRING' data type named id. The command to implement this is shown below.

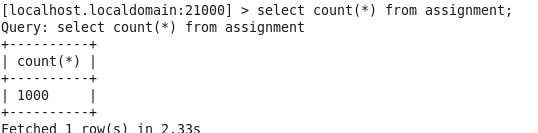


* ‘DESCRIBE table\_name’ can show the column information of that table.

1. **Load a file, called thousand\_strings.txt, into the table and check the total inserted number of data**

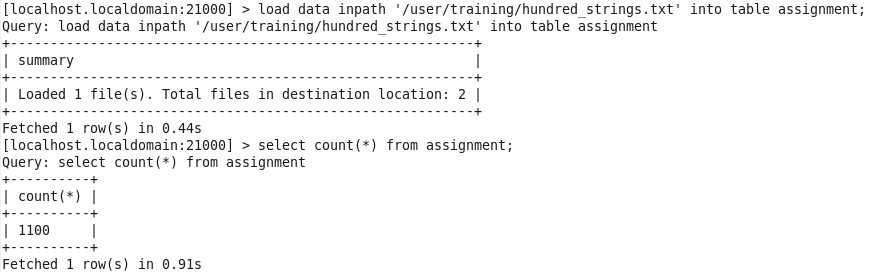


* To load 'thousand\_strings.txt' stored in HDFS into the created table, enter the following command.



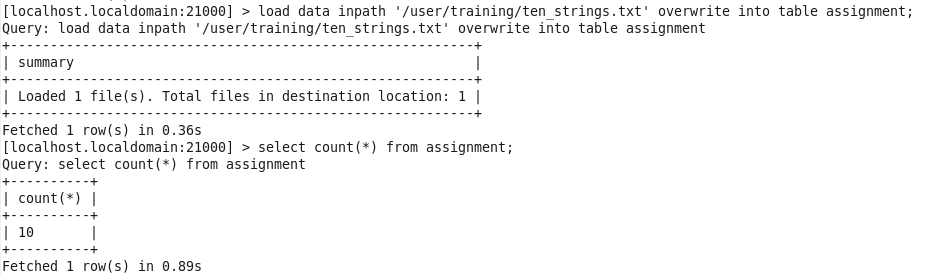
* Through this, 1000 random string variables were loaded in accordance with the table format, and this result can be checked with the following command. As you can see in the picture, you can see that 1000 data is loaded as expected.

1. **Append a file, called hundred\_strings.txt, into the table and check the total inserted number of data**



* Append is to add after the data entered in the existing table, so you just need to execute the same command as in problem (5).
* However, the file name must be changed to 'hundred\_strings.txt'. Likewise, to check the result, execute the same command.
* Since the existing 1000 pieces of data were stored, as a result of Append, 100 pieces of data were additionally stored, and finally, it can be seen that 1100 pieces of data are stored.

1. **Overwrite a file, called ten\_strings.txt, into the table and check the total inserted number of data**



* Finally, Overwrite works in a different way from the existing problems (5) and (6). Overwrite is a method of deleting all stored data in the table and saving it in a new one, so all 1100 previously stored data are lost. After that, 10 data of 'ten\_strings.txt' to be overwritten are added.
* The result is run with the same command. As explained, you can see that 10 pieces of data are stored in the table.