Hand-on: HBase basic data manipulation

1. Install HBase. For standalone installation, we may not need to install Hadoop.

2. After complete installation, open a terminal and start HBase service.

```
./script_start_service.sh
```

3. Use an HBase shell to communicate with HBase.
   Note that HBase is written in Java and has a Java Native API but in this lab, we would like to access Hbase via a Shell.

```
hbase shell
```

4. *List is a command used to get the list of all the tables in HBase. Verifty the installation and the configuration of HBase in your system using this command as shown below.*

```
list
```

5. *Use ‘status’ command to return the status of the system including the details of the servers running on the system*

```
status
```

6. Use *create `<table name>`,`<column family>`* to create a table named *student* with two column families.

```
create `student`, {NAME => `personal`}, {NAME => `skill`}

or

create `student`, `personal`, `skill`
```

column family

Row key	*personal*		*skill*	
	name	*age*	*programming*	*speaking*
A1	*Phond*	*25*	*Java*	*Thai*
A2	…	…	…	…

Show your capture.

7. List existing tables by using the command *list*. You must see all the existing tables. Show a capture.

8. Insert data of “Phond” shown in the table in 6 by using the command

   put `table name`, `row key`, `column family:column name`, `value`.

   Example to add her name
   put `student`, `A1`, `personal:name`, `Phond`

   Show the rest of your commands and capture.

9. Insert data of yourself. Show your commands and the captured result.

10. Type ***scan `student`*** *on the shell. Show your capture. What is the command used for?*

11. Type ***scan `student`, {COLUMNS => [`personal:age`]}***. *Show the result. What is the difference between the commands in 8 and 9.*

12. *Type* ***scan `student`, {COLUMNS => [`personal:age`] , LIMIT => 1****}. Show the result.*

13. Use the *get* operation which allows you to get one row of data at a time. Type the following. What do you see?

   get `student`, `A1`

14. Use *get `table name`, `row id`, {COLUMN ⇒ `column family:column name`} to read Phond`s programming skill. Show the result.*

15. What is the key difference between scan and get operations?

16. Use the put operation to update Phond`s programming skill from Java to Python. Show the command and capture.

17. Use ***alter `table name`, {NAME=> `column family2`}*** to add a new column family named “friends”. Show your command and answer what happens after the command is used.

18. Next use ***alter ` table name `, `delete` ⇒ ` column family`*** *to delete the column family in 17.* Show what happens after the command is used.

19. Use ***count `table name`*** to count the number of rows in the target table.

20. Type ***get `student`, `A1`.*** After that type ***delete `student`, `A1`, `personal:age`***. What happens after the delete command is used?

21. Type ***disable `student`*** *and then type* **scan `student`**. *What is the result?*

22. To check whether the table `student` is disable, show the command and result.

23. Exit the shell. You exit the shell by typing the *'exit'* command.