Advanced Hive- Assignment 2

Write a hive UDF that implements functionality of string concat_ws(string SEP, array<string>).
This UDF will accept two arguments, one string and one array of string.
It will return a single string where all the elements of the array are separated by the SEP.

For this assignment we are going to use eclipse to write User Defined function and use the Hive shell to execute the UDF function and execute the output.

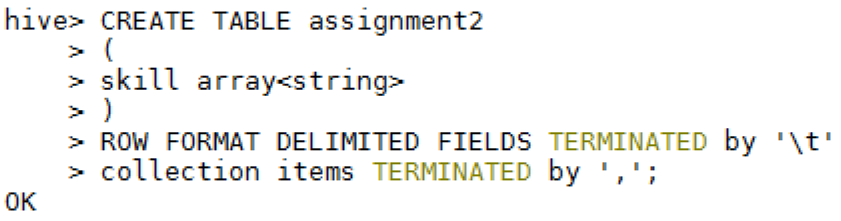
Below are the steps followed for this assignment.

Creating the table and loading the data

First, we have to create a table named assignment2, with the one field which is of type, array of String.
we are going use the contents of this table as input to provide solution for this assignment.
The command used to create table is as below.

```
CREATE TABLE assignment2
(
skill array<string>
)
ROW FORMAT DELIMITED FIELDS TERMINATED by '\t'
collection items TERMINATED by ',';
```

Screenshot of Mobaxterm for creating the table assignment2:

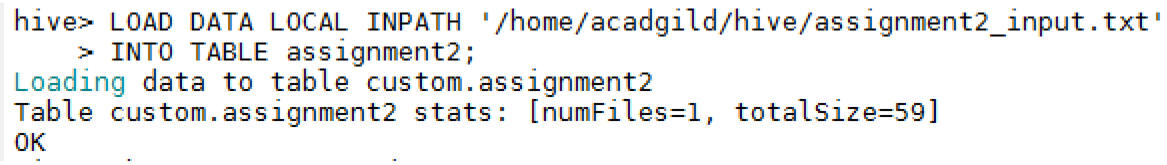


```
hive> CREATE TABLE assignment2
    > (
    > skill array<string>
    > )
    > ROW FORMAT DELIMITED FIELDS TERMINATED by '\t'
    > collection items TERMINATED by ',';
OK
```

1) Next we have to load the contents of assignment2.txt using the below commands.

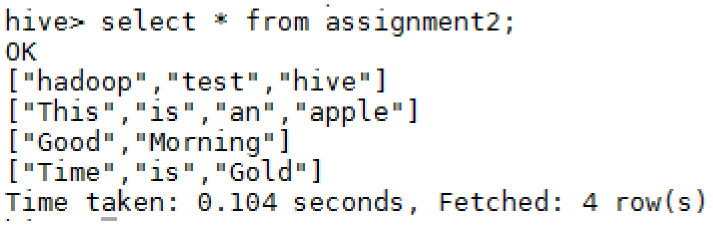
   ```
   LOAD DATA LOCAL INPATH '/home/acadgild/hive/assignment2_input.txt'
   INTO TABLE assignment2;
   ```

Screenshot of Mobaxterm for loading assignment2.txt into assignment2:



```
hive> LOAD DATA LOCAL INPATH '/home/acadgild/hive/assignment2_input.txt'
    > INTO TABLE assignment2;
Loading data to table custom.assignment2
Table custom.assignment2 stats: [numFiles=1, totalSize=59]
OK
```

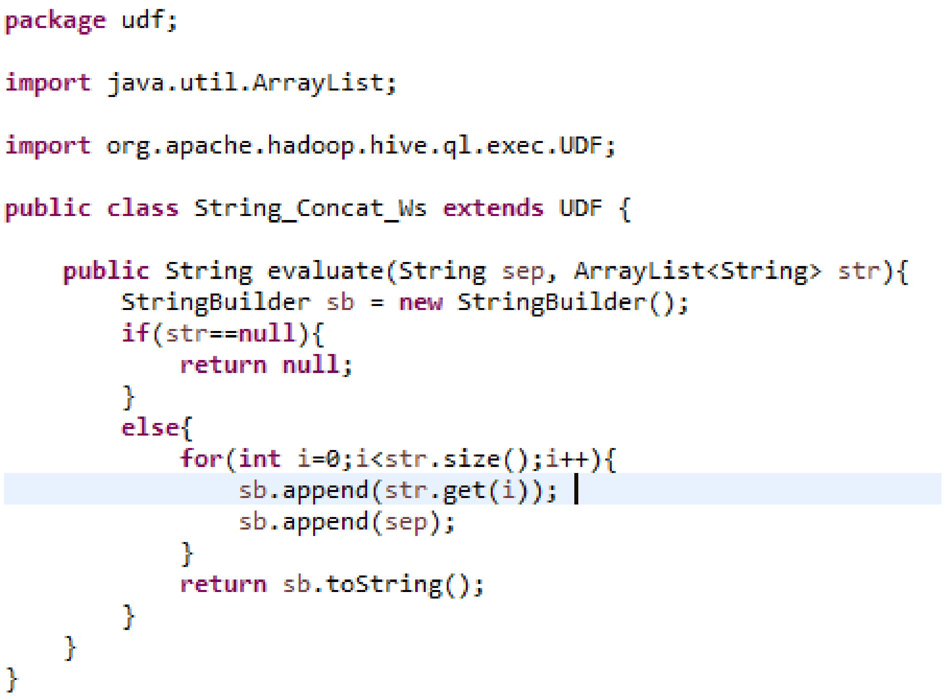
Screenshot of Mobaxterm for viewing the contents of assignment2:



```
hive> select * from assignment2;
OK
["hadoop","test","hive"]
["This","is","an","apple"]
["Good","Morning"]
["Time","is","Gold"]
Time taken: 0.104 seconds, Fetched: 4 row(s)
```

Steps to write UDF in Java Eclipse :

1. I've Created a , Java project within the project created a Java class for the User Defined Function which extends ora.apache.hadoop.hive.sq.exec.UDF and implement evaluate() methods.
2. Added the hive-exex jar to the project library.
3. Extendeds the UDF class
4. Then added the needed logic in the evaluate() method .
5. In the input we are taking 2 arguments, string and arraylist of String. Hive UDF uses arraylist to handle array datatype of hive. We are inserting the string seprater after each arraylist item. So we are iterating thru the arraylist and appending the string item in between the arraylist item.Its given in the below codes.
6. Once it is done, we are packaging as a Jar file.



```
package udf;

import java.util.ArrayList;

import org.apache.hadoop.hive.ql.exec.UDF;

public class String_Concat_Ws extends UDF {

    public String evaluate(String sep, ArrayList<String> str){
        StringBuilder sb = new StringBuilder();
        if(str==null){
            return null;
        }
        else{
            for(int i=0;i<str.size();i++){
                sb.append(str.get(i));
                sb.append(sep);
            }
            return sb.toString();
        }
    }
}
```

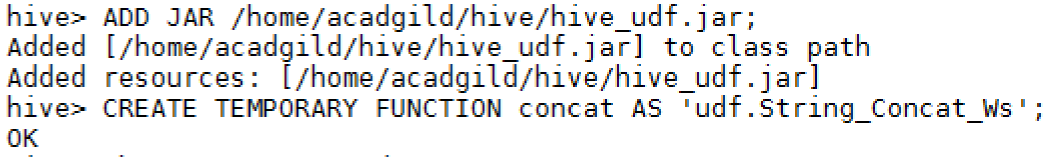
Steps to add the jar to Hive shell and use the jar :

1. First we to import the jar projct in mobaxterm.
2. Then we have to add the jar using the below command

   ADD JAR /home/acadgild/hive/hive_udf.jar;
3. CREATE TEMPORARY FUNCTION in Hive which points to your Java class

   CREATE TEMPORARY FUNCTION concat AS 'udf.String_Concat_Ws';

Screenshot of Mobaxterm for registering the hive_udf.jar:



```
hive> ADD JAR /home/acadgild/hive/hive_udf.jar;
Added [/home/acadgild/hive/hive_udf.jar] to class path
Added resources: [/home/acadgild/hive/hive_udf.jar]
hive> CREATE TEMPORARY FUNCTION concat AS 'udf.String_Concat_Ws';
OK
```

Steps to call the concat function by passing the arguments:

We are using the below query to call the concat funtion, we are passing the string and String[] arguments) which is from the assigment table. We can see the concatenated “*’ with each array item.

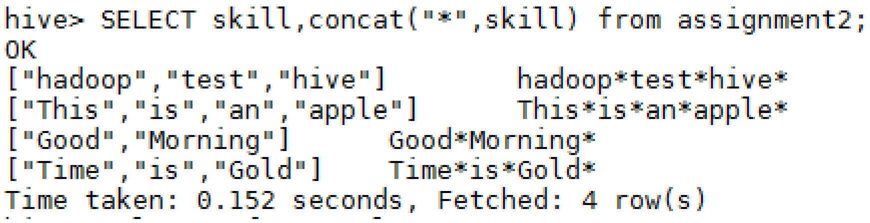
Query: :

SELECT skill,concat("*",skill) from assignment2;

Output : :

```
["hadoop","test","hive"]      hadoop*test*hive*
["This","is","an","apple"]     This*is*an*apple*
["Good","Morning"]     Good*Morning*
["Time","is","Gold"]    Time*is*Gold*
```

Screenshot of Mobaxterm for the query and output



```
hive> SELECT skill,concat("*",skill) from assignment2;
OK
["hadoop","test","hive"]        hadoop*test*hive*
["This","is","an","apple"]      This*is*an*apple*
["Good","Morning"]      Good*Morning*
["Time","is","Gold"]    Time*is*Gold*
Time taken: 0.152 seconds, Fetched: 4 row(s)
```