Assignment 8.2

TASK :

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.

To complete this task below steps are followed:

Step 1: Started all hadoop daemons using start-all.sh command inside /$HADOOP_HOME/sbin. Started mysqld service using sudo service mysqld start.

Step 2: Launched hive using hive command.

Step 3: Created emp_array.txt file inside /home/acadgild/Documents/mydata/Hive/Practice with following contents:

Joe	Analyst,Data Engineer,Data Consultant
Dan	Analyst,Software Engineer,Software Consultant
Alex	Director,Project Manager,Project Consultant
John	Analyst,Test Engineer,Software Consultant

Step 4: Created emp_array table inside emp database as follows:

```
hive> show databases like 'e.*';
OK
emp
Time taken: 11.849 seconds, Fetched: 1 row(s)
hive> use emp;
OK
Time taken: 0.324 seconds
hive> show tables;                          emp_array table does
OK                                          not already exist
emp_with_salary
employee
employee_partitioned
Time taken: 0.396 seconds, Fetched: 3 row(s)
```

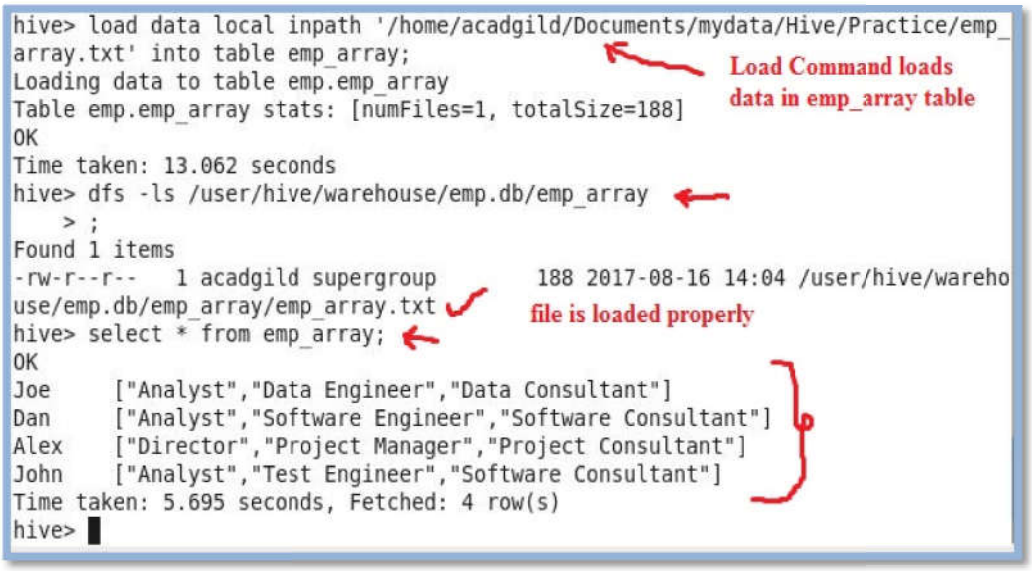
```
hive> create table emp_array(
    > name string,
    > desig array<string>)
    > row format delimited
    > fields terminated by '\t'
    > collection items terminated by ',';
OK
Time taken: 4.042 seconds
hive> describe emp_array;
OK
name                    string
desig                   array<string>
Time taken: 1.491 seconds, Fetched: 2 row(s)
hive> dfs -ls /user/hive/warehouse/emp.db;
Found 4 items
drwxr-xr-x   - acadgild supergroup          0 2017-08-16 13:55 /user/hive/wareho
use/emp.db/emp_array
drwxr-xr-x   - acadgild supergroup          0 2017-08-06 21:30 /user/hive/wareho
use/emp.db/emp_with_salary
drwxr-xr-x   - acadgild supergroup          0 2017-07-06 17:12 /user/hive/wareho
use/emp.db/employee
drwxr-xr-x   - acadgild supergroup          0 2017-07-09 20:41 /user/hive/wareho
use/emp.db/employee_partitioned
hive>
```

Creation of emp_array table with two fields

Schema of emp_array table

command shows, emp_array is created inside emp database

Step 5: Loaded emp_array table with data inside emp_array.txt file as follows:



Step 6: Created JoinArray class extending UDF class inside JoinArray.java file using eclipse, and code for UDF is as follows:

```
import java.util.ArrayList;    //Since ArrayList class is used, so this import is required
import org.apache.hadoop.hive.ql.exec.UDF;    //this import is required to create custom UDF that would work with hive query

public class JoinArray extends UDF {          //extending UDF class
    public String evaluate(String sep,ArrayList<String> arr) {    //overriding evaluate method
                                                    //ArrayList class uses a dynamic array for storing the
                                                    elements. It inherits AbstractList class and implements List
                                                    interface.
                                                    //Here, arr object is created of type ArrayList<String>
        StringBuffer strBuffer;        //strBuffer object of StringBuffer class is created as it is mutable
        if(arr == null){
            return null;               //null is returned if user provides second element as null in query
        }
        strBuffer = new StringBuffer();    //object is instantiated
        strBuffer.append(arr.get(0));       //strBuffer object contains first element of arr using append method
        for(int i=1;i<arr.size();i++){      //arr.size returns total size of array
            strBuffer.append(sep);          //loop runs from 1st index of array to its last index and all elements of
            strBuffer.append(arr.get(i));   //arr get appended to strBuffer one by one separated by sep provided by user as
        }                                   //first element in query
        return strBuffer.toString();        //finally strBuffer is converted to String type i.e. immutable and returned

    }

}
```

Step 7: Created JoinEmpDesigArray.jar for file JoinArray.java.

Step 8: At hive prompt, added this jar i.e. JoinEmpDesigArray.jar, after successful addition of jar, created temporary function arraySep(param1,param2).

```
hive> add jar /home/acadgild/Documents/mydata/Hive/Practice/JoinEmpDesigArray.ja
r;
Added [/home/acadgild/Documents/mydata/Hive/Practice/JoinEmpDesigArray.jar] to c
lass path
Added resources: [/home/acadgild/Documents/mydata/Hive/Practice/JoinEmpDesigArra
y.jar]
hive> create temporary function arraySep as 'JoinArray';
OK
Time taken: 3.15 seconds
hive>
```

Added jar

created temporary function i.e. "arraySep" from "JoinArray.class" [without using .class with class name]

Step 9: At hive prompt, executed query with UDF i.e. arraySep(), as follows:

```
hive> select arraySep('*',desig) from emp_array;
OK
Analyst*Data Engineer*Data Consultant
Analyst*Software Engineer*Software Consultant
Director*Project Manager*Project Consultant
Analyst*Test Engineer*Software Consultant
Time taken: 2.261 seconds, Fetched: 4 row(s)
hive> select arraySep('| ',desig) from emp_array;
OK
Analyst| Data Engineer| Data Consultant
Analyst| Software Engineer| Software Consultant
Director| Project Manager| Project Consultant
Analyst| Test Engineer| Software Consultant
Time taken: 0.497 seconds, Fetched: 4 row(s)
hive>
```

Here, first argument provided to "arraySep" UDF is '*', and second argument is "desig" which is array of strings, therefore, output is all strings separated by '*'

Here, first argument provided to "arraySep" UDF is '| ', and second argument is "desig" which is array of strings, therefore, output is all strings separated by '| '

```
hive> select arraySep(name,desig) from emp_array;
OK
AnalystJoeData EngineerJoeData Consultant
AnalystDanSoftware EngineerDanSoftware Consultant
DirectorAlexProject ManagerAlexProject Consultant
AnalystJohnTest EngineerJohnSoftware Consultant
Time taken: 6.977 seconds, Fetched: 4 row(s)
hive>
```

Here, first argument provided to "arraySep" UDF is 'name', and second argument is "desig" which is array of strings, therefore, output is all strings separated by value of name

```
hive> select arraySep('| ',null) from emp_array;
OK
NULL
NULL
NULL
NULL
Time taken: 0.337 seconds, Fetched: 4 row(s)
hive>
```

when user supplies null as second argument, then null is returned