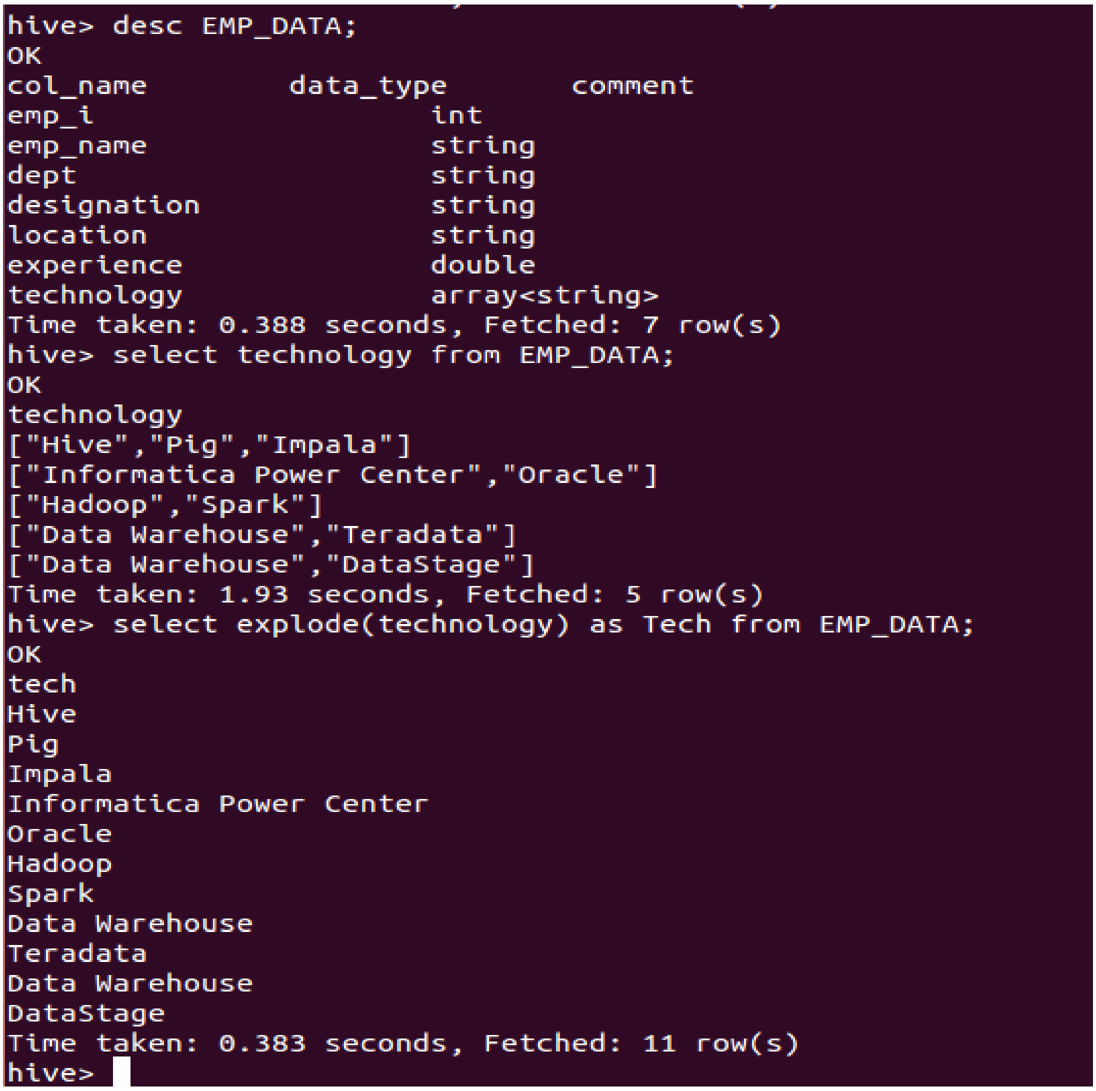
Explode and Lateral View in Hive

- **Explode:**

Explode() function takes an array as an input and returns elements of that array as separate rows.

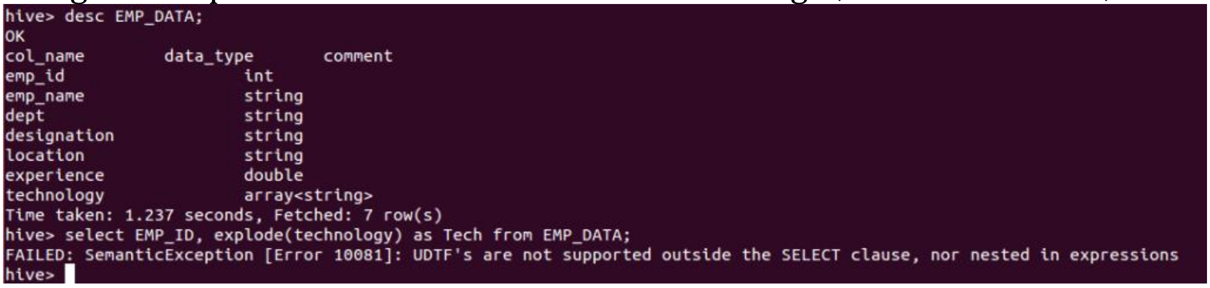
```
Select explode(column_name) from table_name;
```

In below example, we have column technology as array of string. And if we use explode function on technology column, each value of array is separated into rows.Which means, for every element of array a new row has been created in the output.



```
hive> desc EMP_DATA;
OK
col_name        data_type       comment
emp_i                   int
emp_name                string
dept                    string
designation             string
location                string
experience              double
technology              array<string>
Time taken: 0.388 seconds, Fetched: 7 row(s)
hive> select technology from EMP_DATA;
OK
technology
["Hive","Pig","Impala"]
["Informatica Power Center","Oracle"]
["Hadoop","Spark"]
["Data Warehouse","Teradata"]
["Data Warehouse","DataStage"]
Time taken: 1.93 seconds, Fetched: 5 row(s)
hive> select explode(technology) as Tech from EMP_DATA;
OK
tech
Hive
Pig
Impala
Informatica Power Center
Oracle
Hadoop
Spark
Data Warehouse
Teradata
Data Warehouse
DataStage
Time taken: 0.383 seconds, Fetched: 11 row(s)
hive>
```

Limitation of Explode() function – We can select only the column to be exploded in our select statement, we can not select other columns of table along with exploded column. Below is the error we get, when we do this,



```
hive> desc EMP_DATA;
OK
col_name        data_type       comment
emp_id                  int
emp_name                string
dept                    string
designation             string
location                string
experience              double
technology              array<string>
Time taken: 1.237 seconds, Fetched: 7 row(s)
hive> select EMP_ID, explode(technology) as Tech from EMP_DATA;
FAILED: SemanticException [Error 10081]: UDTF's are not supported outside the SELECT clause, nor nested in expressions
hive> 
```

- **Lateral View:**

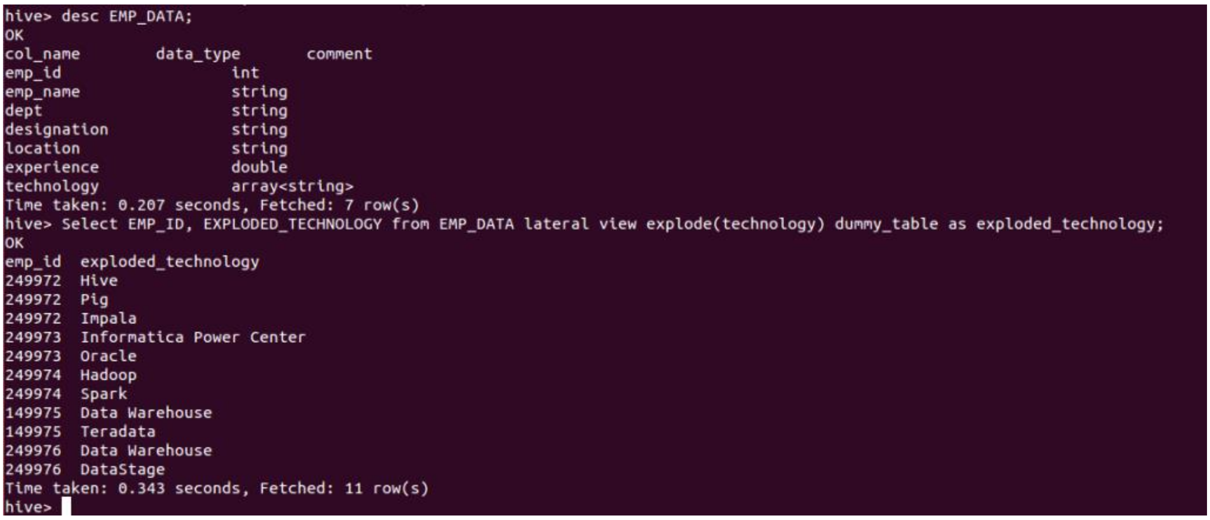
With Lateral View, we can select any number of columns along with Exploded column.

Lateral view creates virtual table and output of exploded column is stored temporarily in virtual table and then that virtual table is joined with the base table to get the desired output.

Syntax is –

```
Select Column_name1, New_Exploded_Column_name from table_name
lateral view explode(column_name_to_be_exploded)
virtual_table_name as New_Exploded_Column_name;
```

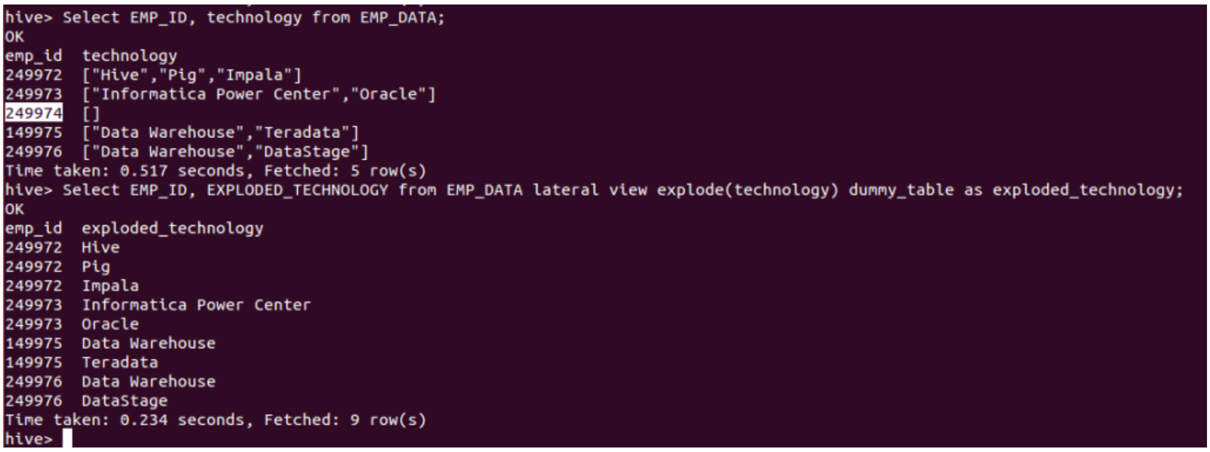
In below example, we have exploded column ‘technology’ stored it in virtual table ‘dummy_table’ and under column ‘exploded_technology’ and selected ’emp_id’ from base table along with ‘exploded_technology’ in select statement.



```
hive> desc EMP_DATA;
OK
col_name        data_type       comment
emp_id                  int
emp_name                string
dept                    string
designation             string
location                string
experience              double
technology              array<string>
Time taken: 0.207 seconds, Fetched: 7 row(s)
hive> Select EMP_ID, EXPLODED_TECHNOLOGY from EMP_DATA lateral view explode(technology) dummy_table as exploded_technology;
OK
emp_id  exploded_technology
249972  Hive
249972  Pig
249972  Impala
249973  Informatica Power Center
249973  Oracle
249974  Hadoop
249974  Spark
149975  Data Warehouse
149975  Teradata
249976  Data Warehouse
249976  DataStage
Time taken: 0.343 seconds, Fetched: 11 row(s)
hive> 
```

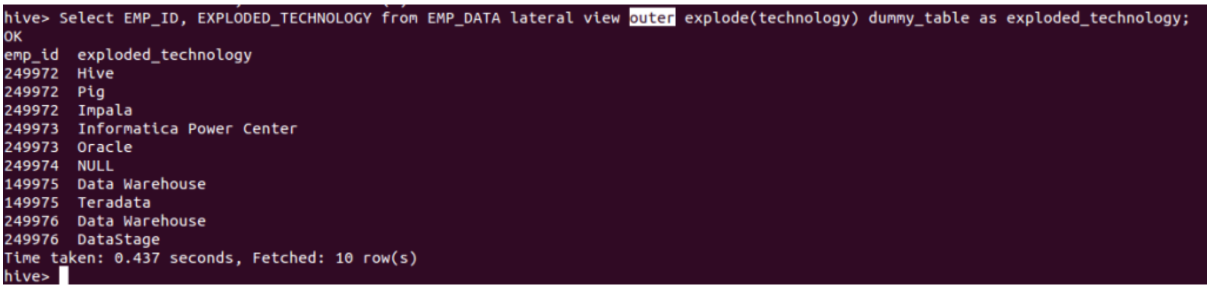
- **Outer Lateral View:**

Suppose we have some rows having null array value. In such case Lateral view along wit explode() function will skip that row. Observe the same in below example:



```
hive> Select EMP_ID, technology from EMP_DATA;
OK
emp_id  technology
249972  ["Hive","Pig","Impala"]
249973  ["Informatica Power Center","Oracle"]
249974  []
149975  ["Data Warehouse","Teradata"]
249976  ["Data Warehouse","DataStage"]
Time taken: 0.517 seconds, Fetched: 5 row(s)
hive> Select EMP_ID, EXPLODED_TECHNOLOGY from EMP_DATA lateral view explode(technology) dummy_table as exploded_technology;
OK
emp_id  exploded_technology
249972  Hive
249972  Pig
249972  Impala
249973  Informatica Power Center
249973  Oracle
149975  Data Warehouse
149975  Teradata
249976  Data Warehouse
249976  DataStage
Time taken: 0.234 seconds, Fetched: 9 row(s)
hive>
```

To overcome this, use keyword outer between Lateral view and Explode() function. As given below, now we can see the array with null value.

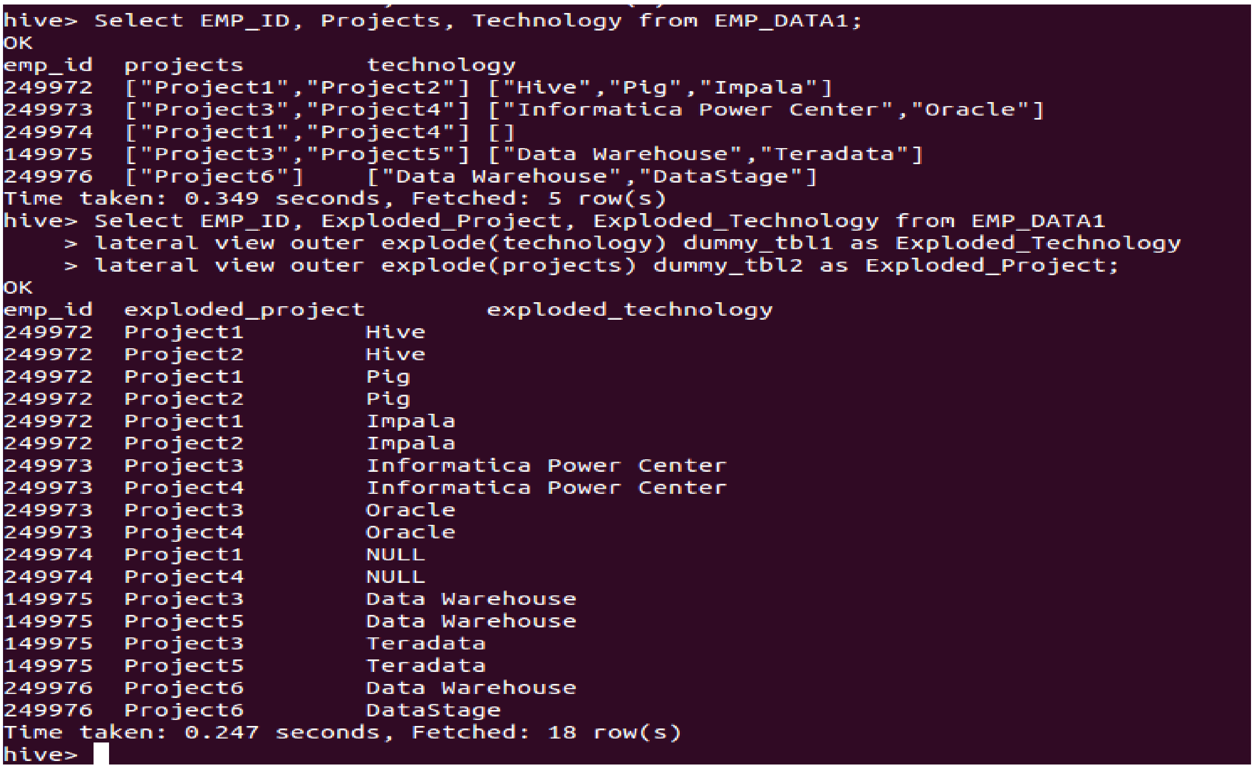


```
hive> Select EMP_ID, EXPLODED_TECHNOLOGY from EMP_DATA lateral view outer explode(technology) dummy_table as exploded_technology;
OK
emp_id  exploded_technology
249972  Hive
249972  Pig
249972  Impala
249973  Informatica Power Center
249973  Oracle
249974  NULL
149975  Data Warehouse
149975  Teradata
249976  Data Warehouse
249976  DataStage
Time taken: 0.437 seconds, Fetched: 10 row(s)
hive>
```

- **Multiple Lateral Views:**

Suppose a table have more than 1 array column and we want both of them to be transposed at once, we can use two lateral view statements in single query like given in below example.

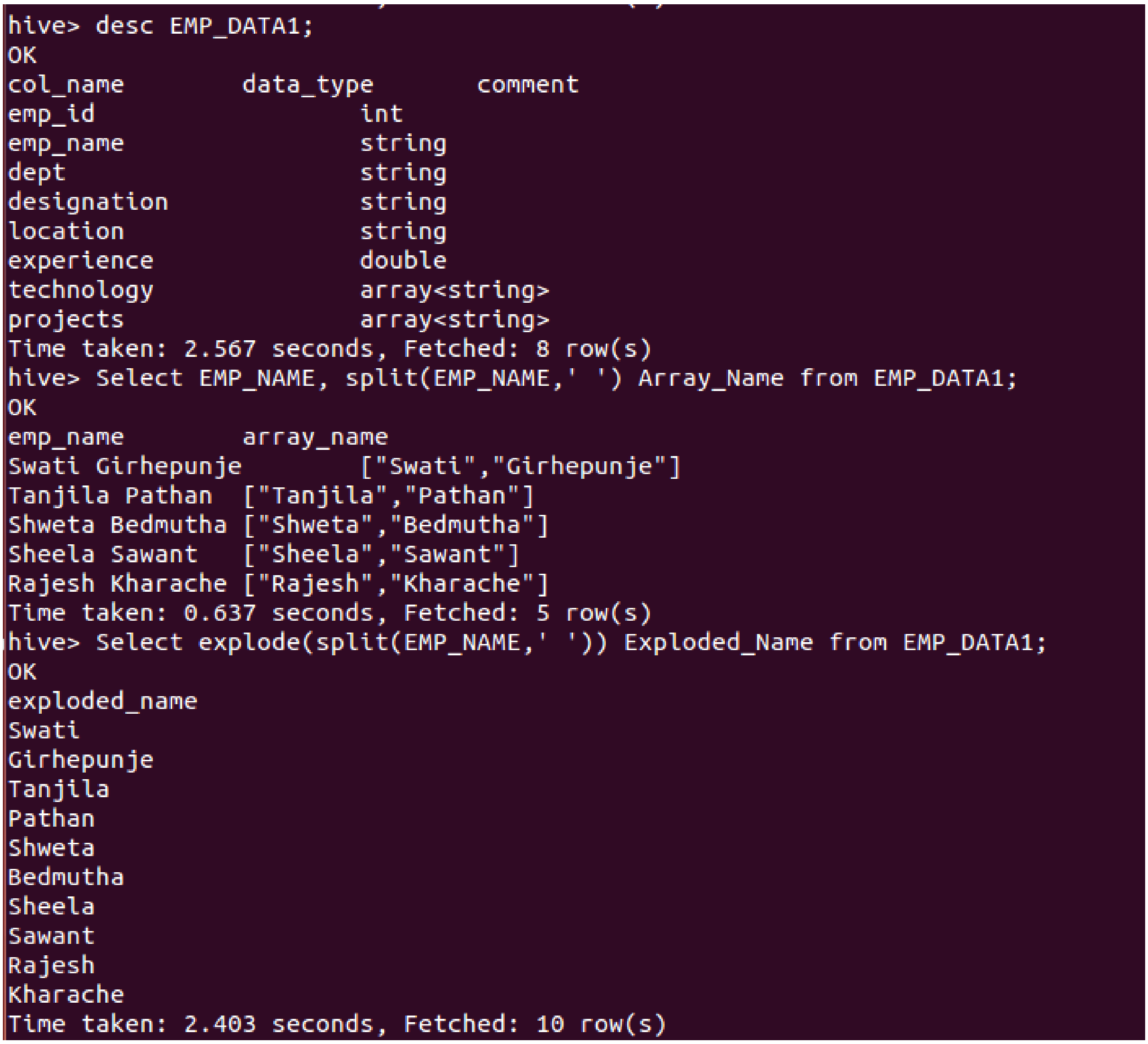
(Pto)



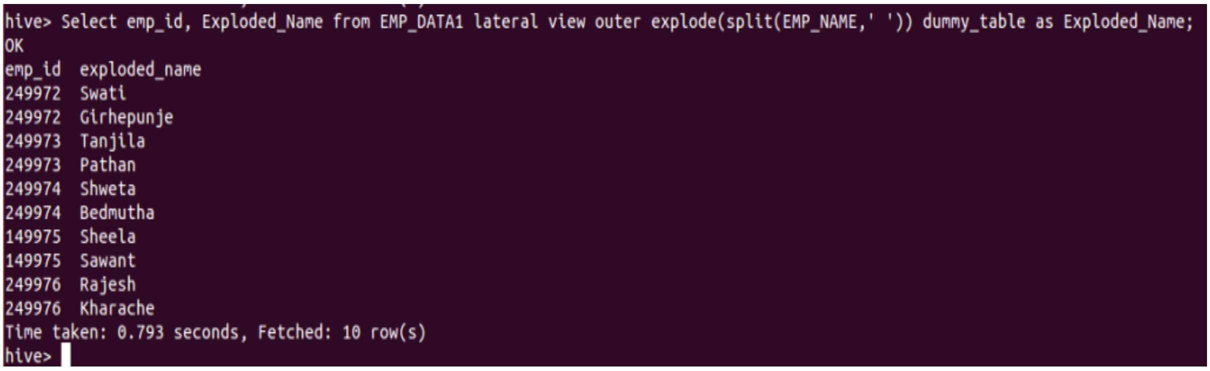
```
hive> Select EMP_ID, Projects, Technology from EMP_DATA1;
OK
emp_id  projects        technology
249972  ["Project1","Project2"] ["Hive","Pig","Impala"]
249973  ["Project3","Project4"] ["Informatica Power Center","Oracle"]
249974  ["Project1","Project4"] []
149975  ["Project3","Project5"] ["Data Warehouse","Teradata"]
249976  ["Project6"]    ["Data Warehouse","DataStage"]
Time taken: 0.349 seconds, Fetched: 5 row(s)
hive> Select EMP_ID, Exploded_Project, Exploded_Technology from EMP_DATA1
    > lateral view outer explode(technology) dummy_tbl1 as Exploded_Technology
    > lateral view outer explode(projects) dummy_tbl2 as Exploded_Project;
OK
emp_id  exploded_project        exploded_technology
249972  Project1        Hive
249972  Project2        Hive
249972  Project1        Pig
249972  Project2        Pig
249972  Project1        Impala
249972  Project2        Impala
249973  Project3        Informatica Power Center
249973  Project4        Informatica Power Center
249973  Project3        Oracle
249973  Project4        Oracle
249974  Project1        NULL
249974  Project4        NULL
149975  Project3        Data Warehouse
149975  Project5        Data Warehouse
149975  Project3        Teradata
149975  Project5        Teradata
249976  Project6        Data Warehouse
249976  Project6        DataStage
Time taken: 0.247 seconds, Fetched: 18 row(s)
hive>
```

- **Converting String Data to Array Data and then applying Explode:**

In below example, we have column EMP_NAME with String data type, but the data in it is separated by space. So we converted it to array by using function split and then applied the explode function on it.



```
hive> desc EMP_DATA1;
OK
col_name        data_type       comment
emp_id                  int
emp_name                string
dept                    string
designation             string
location                string
experience              double
technology              array<string>
projects                array<string>
Time taken: 2.567 seconds, Fetched: 8 row(s)
hive> Select EMP_NAME, split(EMP_NAME,' ') Array_Name from EMP_DATA1;
OK
emp_name        array_name
Swati Girhepunje        ["Swati","Girhepunje"]
Tanjila Pathan  ["Tanjila","Pathan"]
Shweta Bedmutha ["Shweta","Bedmutha"]
Sheela Sawant   ["Sheela","Sawant"]
Rajesh Kharache ["Rajesh","Kharache"]
Time taken: 0.637 seconds, Fetched: 5 row(s)
hive> Select explode(split(EMP_NAME,' ')) Exploded_Name from EMP_DATA1;
OK
exploded_name
Swati
Girhepunje
Tanjila
Pathan
Shweta
Bedmutha
Sheela
Sawant
Rajesh
Kharache
Time taken: 2.403 seconds, Fetched: 10 row(s)
```



```
hive> Select emp_id, Exploded_Name from EMP_DATA1 lateral view outer explode(split(EMP_NAME,' ')) dummy_table as Exploded_Name;
OK
emp_id  exploded_name
249972  Swati
249972  Girhepunje
249973  Tanjila
249973  Pathan
249974  Shweta
249974  Bedmutha
149975  Sheela
149975  Sawant
249976  Rajesh
249976  Kharache
Time taken: 0.793 seconds, Fetched: 10 row(s)
hive>
```