Populating Tables

Now that we have created a table with two column families. The table name is going to be employees. And we have also check that the column family names are going to be a basic info and personal info. Note that we haven't created any columns. We just created two column families.

Now, how do we create the columns and how do we insert the data. There is no separate step to create the columns and then insert the data. Here in HBase, we have to insert the data automatically by specifying the column names to whichever we want to insert into. Now that let me quickly describe the schema of the table called employees.

I just have to load the HBase shell and then I have to use a describe command-describe employees. So, here we have two column families, second column family is going to be personal info and first column family is going to be basic info. If we do a scan operation on employees, it's not going to have any data.

Now, how do we insert data into our employees table? Basically, the operation for that is going to be a put. I want to put data into my employees table and I have to compulsorily specify a RowKey. So I specify

the RowKey for my first row to be 1. So in this RowKey, after that I have to specify, in which column family I am going to insert the data. So I have two column families over here. I need to insert data into my basic info column family followed by a semi colon. I have to mention the name of the column. Let me name the column right now while I am trying to insert the data. I will name this column as employee id then using the comma separator, I have to specify the value what I want to insert into this column. The value I want to insert is the employee id is going to be 101 and I press the enter key.

It says zero rows in 0.250 secs. Let us quickly scan employees. Here what happened is, we get a RowKey 1. Everything is going to be in terms of key value pairs.

For a RowKeys 1, the value is going to be 101. This value called as 101 belongs to a column family called basic underscore info and the column name is going to be eid. So this entire thing is going be the pointer for your column. The column name is going to be eid and that belongs to the column family called as basic underscore info and the data which is going to be stored in that particular column called as eid whose value is going to be 101.

Note that, there is a system generated value which is going to be called as a timestamp field associated with every field that goes into the HBase table.

Now, this value along with the timestamp field and the actual column itself together is actually called as a cell and the cell value is 101 and the RowKey to access that cell is going to be 1.

Note that, we have just added one column into the column family called basic info. Let's quickly add another column and also insert data. Notice that the step involved in creating the column as well as adding the data is a single step operation out here.

Now, we added a numerical value. Let's try to add a string value to another column. So, put data into my table called employees where I have to specify the same RowKey because in the same row, under the same basic underscore info column, family name, I am going to create another column by the name employee names. So, what is going to be the name of the employee? So, since it's going to be a string, I have to specify within single codes. So I would say the employee name is going to be Alice.

Now, quickly verify if the insertion has happened correctly by scanning the contents of employees. So here, for RowKey 1, now, there are two values, one value is going to be 101, the other value is going to be Alice. If you notice that, there is different timestamp for this cell.

As I mentioned earlier, the column name along with the timestamp and the data which goes inside the column which is called as a value together makes up a cell. So we have inserted two cells meaning each cell can be visualized as a column here. So we have two columns. One column is actually called as Eid and the other column called as name and both of these columns actually belong to the column family called basic underscore info and the data corresponding to each of those column is going to be 101 and Alice corresponding to the RowKey 1.

Now, let's try to add some data into the second column family which is called as person underscore info. The command is going to look almost the same except instead of basic info column family name, we just have to put it as personal info. So instead of name, I have to create a new column inside my personal underscore info column family. Let me create this as age. So let us say Alice is going to be 37 years old. This is one of the

fields which we are creating dynamically during run time under the personal underscore info column family. Let's press the enter key.

And now scan the employees table. Here, we have. For RowKey1, we have three columns. Two of those columns actually belong to the basic info column family and the third column actually belongs to the personal underscore info column family where the age of the employee actually is going to be stored.

Now, let's create another column under the personal underscore info column family. This time, we will have to give a different column name. Instead of age, I will just say salary. And mention here as say for example 4000 dollars per month, something like that. Press enter key. And now scan employees.

So, here we have. We have two columns under the personal underscore info column family and we have two columns under the basic underscore info column family and the corresponding data which actually goes inside those columns are going to be 101 for employee id column and Alice under the name column and then we have 37 in the age column and then we have 4000 dollars as a monthly salary in the salary column under the personal underscore info column family.

Let's try to insert another row but this time with a different RowKey for another employee. So, this time, inside my same employee table, I want to create another row with RowKey 2. But this time, I am going to create a column under the basic info and employee id is going to be say 102.

Fine, Let's try to put another piece of data here under basic info, names, the employee name is going to be say Bob. All right. Now, let's quickly scan the contents of the employee table. Here we have. We now have two rows into it.

First row, basically, has four cells into it. remember each cell can be visualized as one column. So we have four columns inside. Row 1 which is identified by RowKey 1 and for the second row identified by RowKey 2, we just have the basic info. We don't have the personal info data loaded yet. Now, let's quickly load up the personal information data for the RowKey 2. So, for RowKey 2, instead of basic info, let me give it as personal info and let's say the age of Bob is going to be say 41.

And let's also quickly load up the salary. Salary is going to be say 5600 dollars per month. Now, the interesting thing to note over here is that, I am going to create a third column into the personal info column family

specifically for the RowKey 2. So I am going to name it as his status, whether is he married or single or whatever it is. So, I would put here as married. And this column is not actually existing for the RowKey 1. That's the beauty of HBase.

Your table actually can be completely flexible in terms of schema specification. All the rows need not have the same number of columns in fact. They would be having the same column families or same number of column families. But within each written, there is no restriction that row with RowKey 1 should have so many number of columns. If you notice here, the rows with RowKey 1 is just having two columns under basic info column family and just two columns under personal info column family.

However, if you notice the row with RowKey 2 is going to be two columns under the basic info column family that is over here, the employee id and the name and I have already inserted two more column data under the personal underscore info column family with age and salary. I am now actually trying to insert data into this marital status column under the personal underscore info column family where I have the third entry over here. Press the enter key.

Now quickly perform a scan operation on the employees table. You would notice here that for the row with RowKey as 2, we have three columns under the personal underscore info column family. We have age, the marital status and then we have salary whereas for the rows with RowKey 1, we just have two columns. We can go ahead and create another column with a completely different name and different data for the row with RowKey 1.

However, this is just to demonstrate that not all the columns might be the same for all the rows or another way not all the rows might have the same number of columns but the column families has to be fixed throughout the table when you create the table itself.