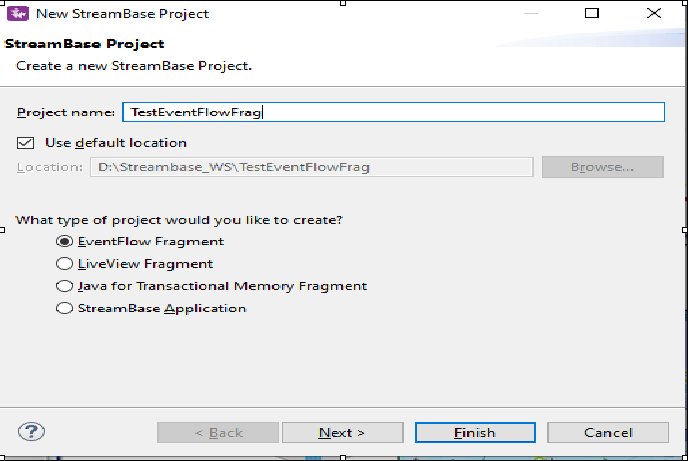
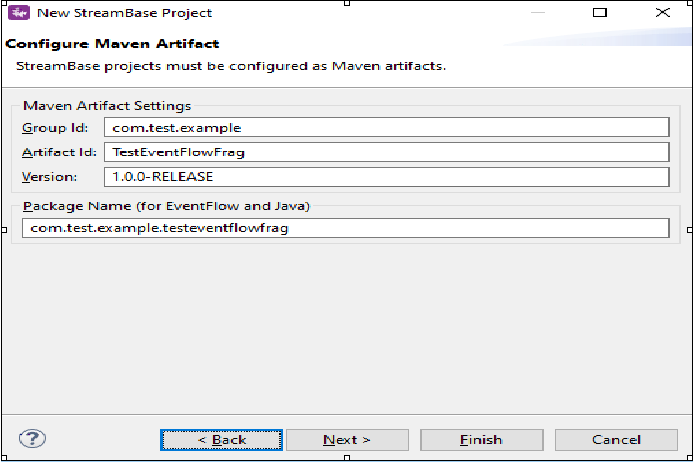
**Create Streambase Studio Project, Create Application Module and Configurations:**

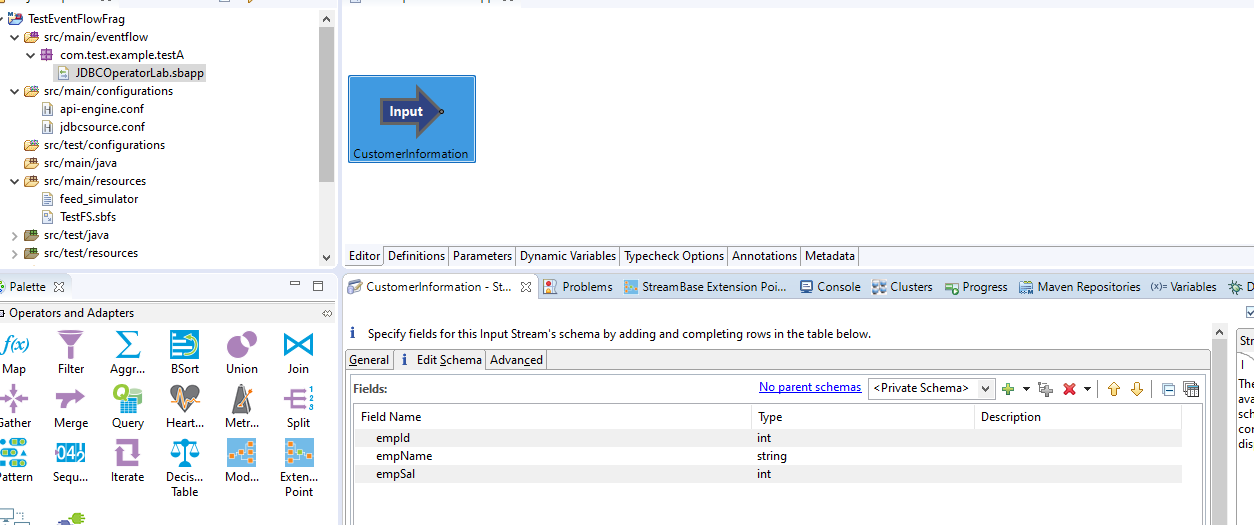
⦁ Create the EventFlow fragment and place all your logic inside this fragment. Select the Radio button as **EventFlow Fragment** as shown in below Screenshot.



⦁ Click Next and provide the value for **Group Id**, **Artifact Id** and **Version**. Provide the packagename as <<Group Id>>.<<any name>> as shown in below screenshot. Please note down these three parameters and click finish.



* Create a new event flow fragment and name it as JDBCOperatorLab.
* Now go to the folder src/maineventflow and click on JDBCOperatorLab.sbapp
* From the palette view click on Input Stream and Drag it into the canvas.
* Name the Input Stream as Customer Information.
* In the Edit Schema tab Give the Field Name as empId,empName,empSal and Select the type as int,string,int.



* Now Drag the JDBC table to the canvas.
* In the General Tab Give the name as CustomerDetailedInformation.
* In Data Source tab, we need to provide JDBC Data Source and the same need to be configured.For this we need to create a .conf file under src/main/configurations as shown below:

name = "testDB"

type = "com.tibco.ep.streambase.configuration.jdbcdatasource"

version = "1.0.0"

configuration = {

"JDBCDataSource" = {

driverClassName = "org.postgresql.Driver"

serverURL = "jdbc:postgresql://localhost:5432/postgres"

userName = "postgres"

password = "Oracle123"

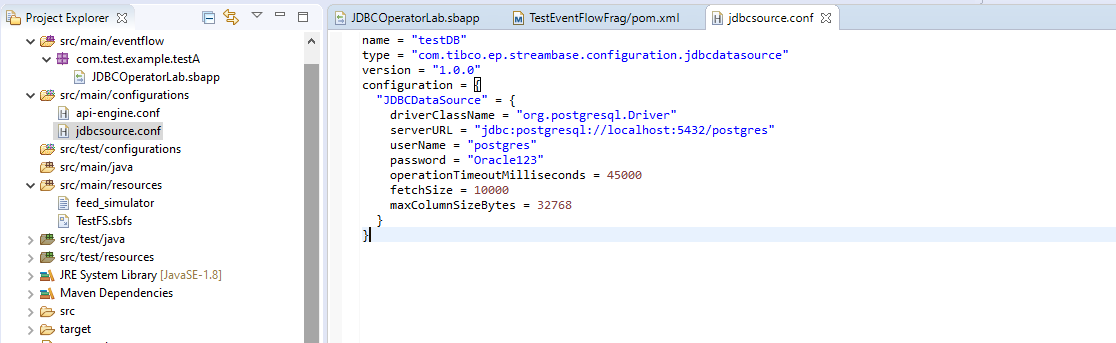
operationTimeoutMilliseconds = 45000

fetchSize = 10000

maxColumnSizeBytes = 32768

}

}



* Here **driverClassName** is the class name for the driver used which is different for different data source(Postgresql, Oracle, Mysql, etc.).In my case it is Postgresql.
* Here **serverURL** isurl used to connect the data source.
* Now in order to connect to any JDBC Data source you need a .jar file which is nothing but a driver or client used to connect to the data source. In case of Postgresql it is postgresql-42.2.14.jar , In case of MySQL it is ojdbc8.jar and so on. Please note that depending on the version of Postgersql or Mysql downloaded the versions for this .jar files may differ and the same can be downloaded from google.
* Once you have the .jar file, open the streambase command prompt and run the below command:

**mvn install:install-file -Dfile="D:\Downloads\jars\postgresql-42.2.14.jar" -DgroupId=org.postgresql -DartifactId=postgresql -Dversion=42.2.14 -Dpackaging=jar**



* Once the above step is done, the dependency need to be added in the pom.xml of our project and than we need to do a mvn clean install as we did in the below steps later.

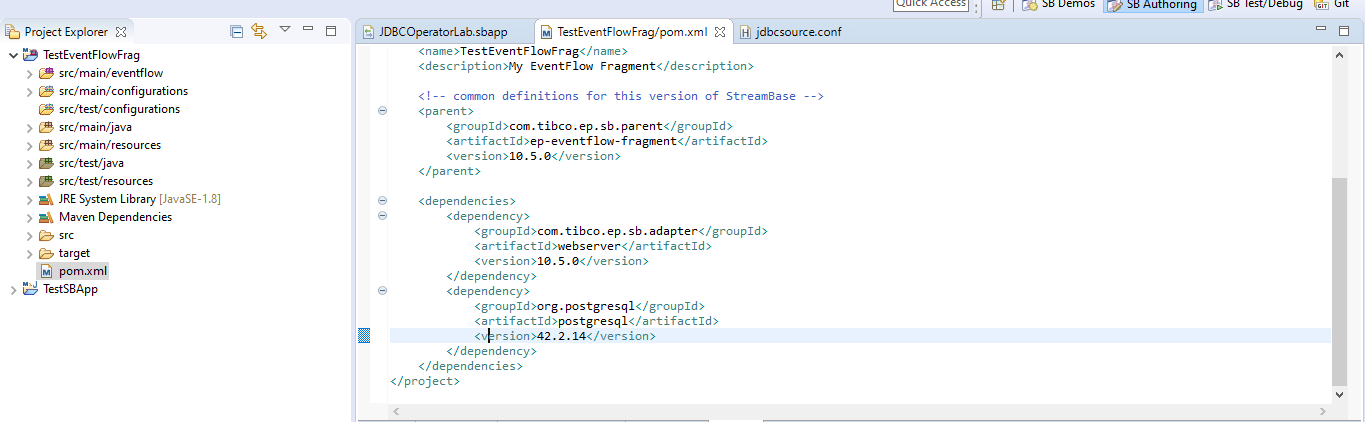
**<dependency>**

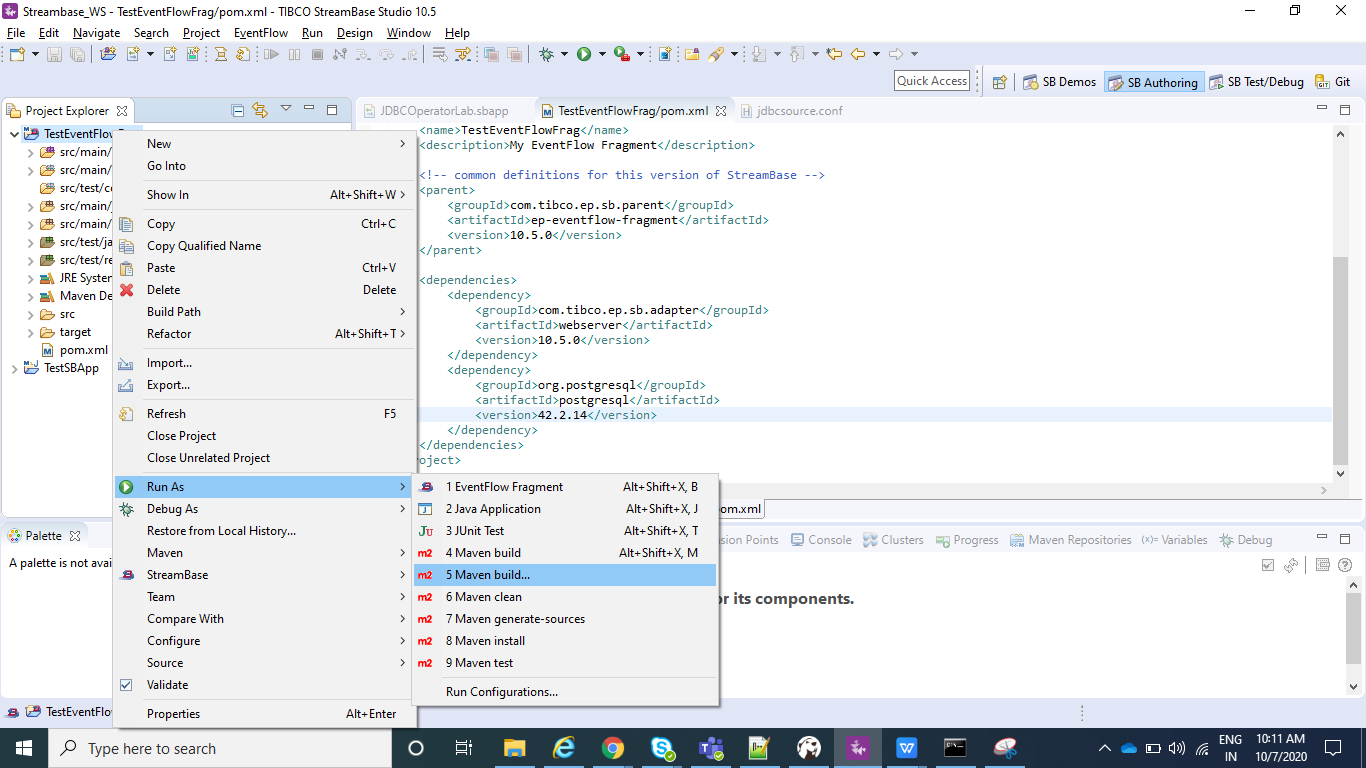
**<groupId>org.postgresql</groupId>**

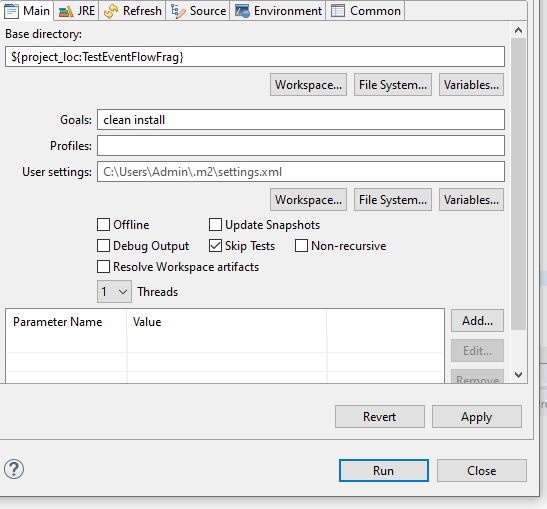
**<artifactId>postgresql</artifactId>**

**<version>42.2.14</version>**

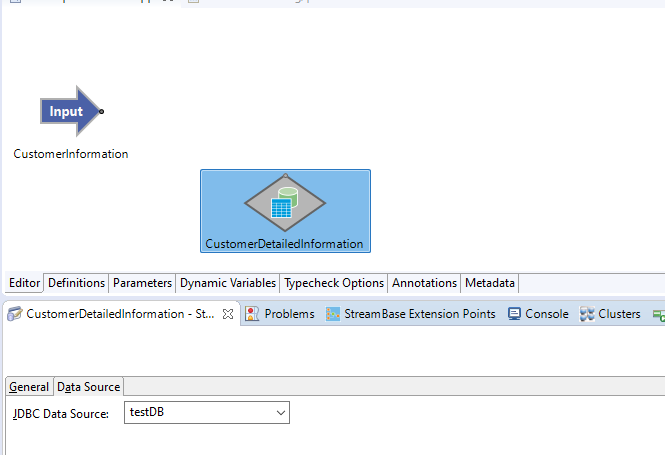
**</dependency>**



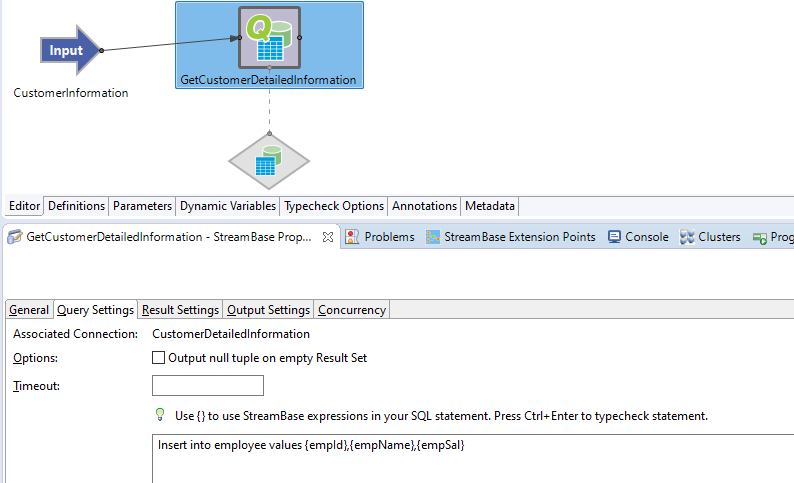




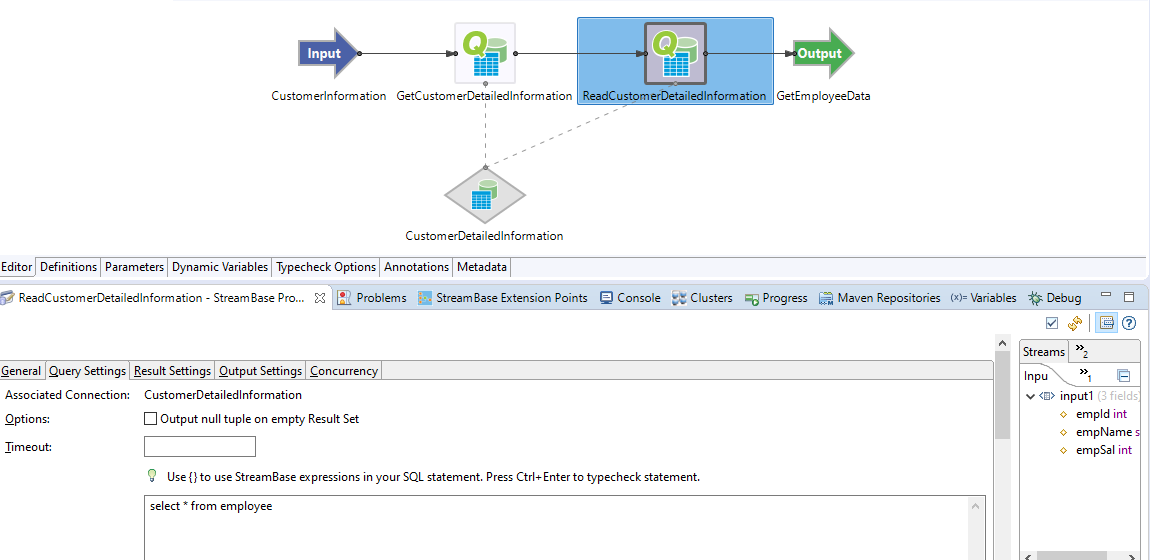
* Once all the above things are done and if everything is proper than you can see your data source in drop down and we can select the same.

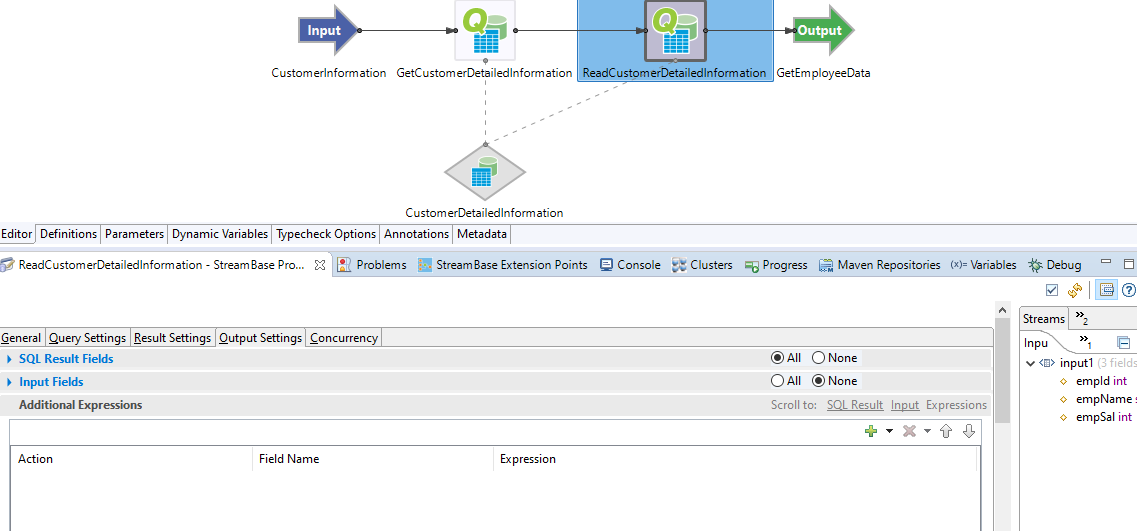


* Now Drag the Query Operator onto the canvas and name it as GetCustomerDetailedInformation.Now connect the input port to the Query Operator port and the Query operator port to the JDBC table. In the Query Operator settings go to Query Setting and and write down your query to read, write or delete as we do normally in database.

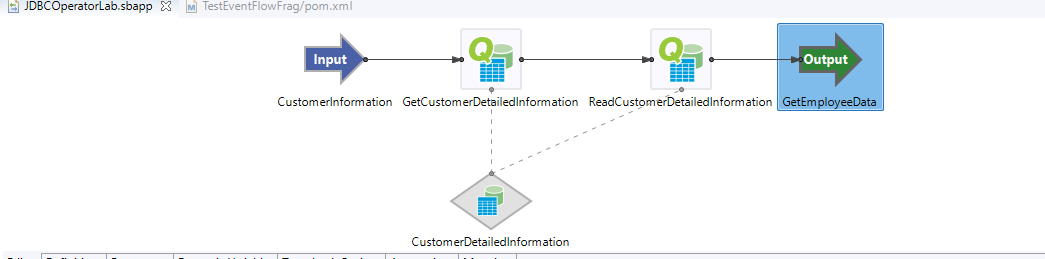


* Now as you can see in above screenshot, I am doing a insert and getting the values of empid, empname and empsal from Input Stream(CustomerInformation). so whenever we get the value from some input we need to use that as the variable name enclosed in {}.Please note the values need to be in () which is missed in above screenshot. So the query would be : Insert into employee values ({empId},{empName},{empSal})
* Now after inserting, I will also do a read operation to read all the values from database.





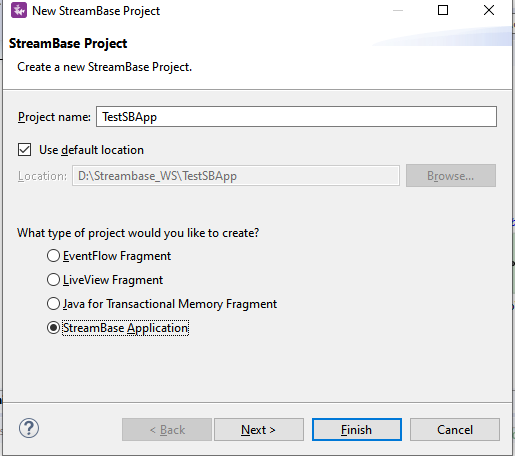
⦁ Now Drag the Output to the canvas and name it as GetEmployeeData.



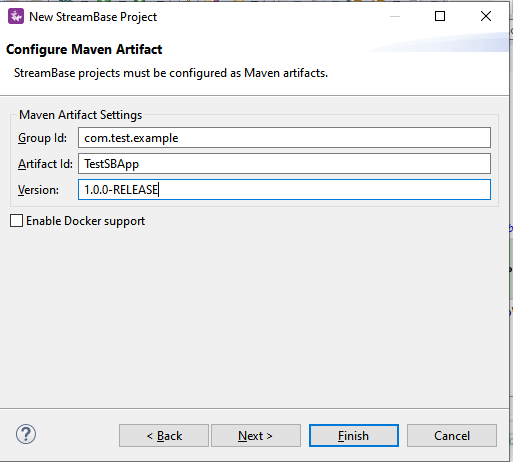
⦁ Now In the Project Explorer view, select and right-click the Streambase Application Project **(TestEventFlowFrag in my case)--> Run As>Maven Build...**

⦁ After that In the **Goals** field, enter **clean install** and Select the **Skip Tests** check box. Click Apply and Run.

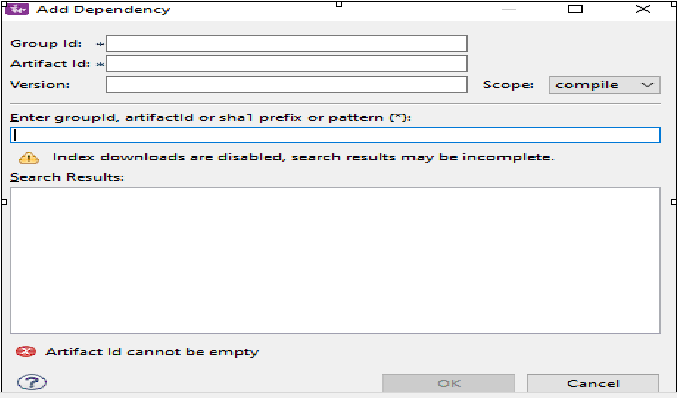
⦁ Now create a new Streambase Project. Select Radio button as **StreamBase Application**.

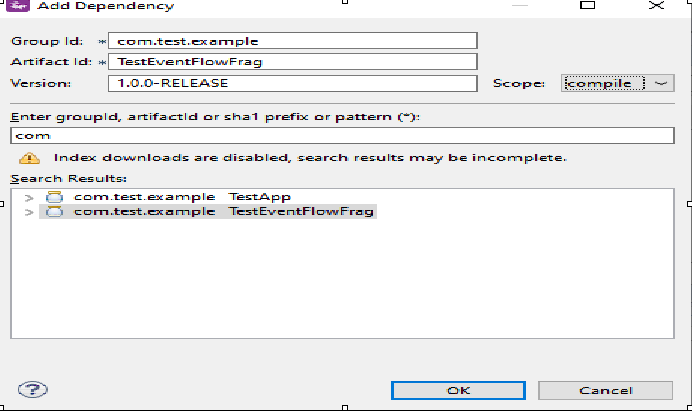


⦁ Now click next and make sure you provide the same **Group Id** and **Version** that you specified for the EventFlow Fragment created Earlier and click **Finish**.



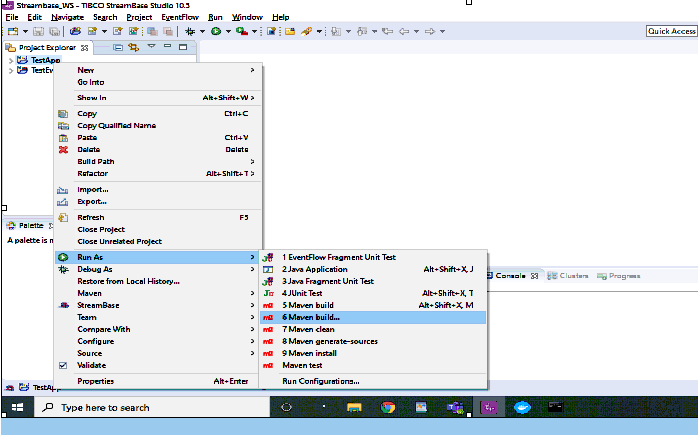
⦁ Add the EventFlow Fragment Project depedency to our main Streambase Application. Right click on Streambase Application Project **(TestSBApp in my case) --> StreamBase>Manage Project Dependencies**. Add the depedency for the EventFlow Fragment and click ok.

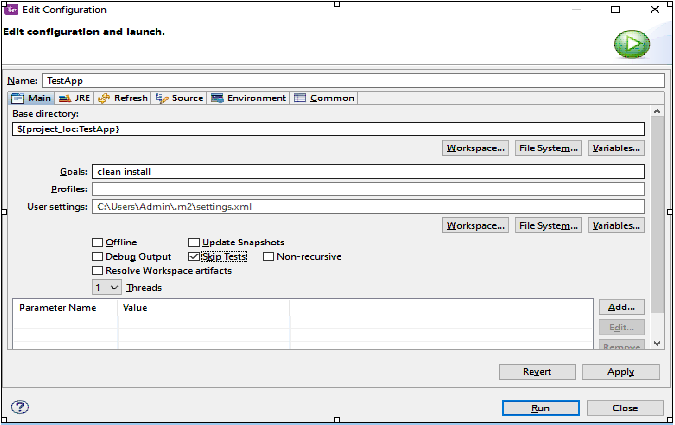


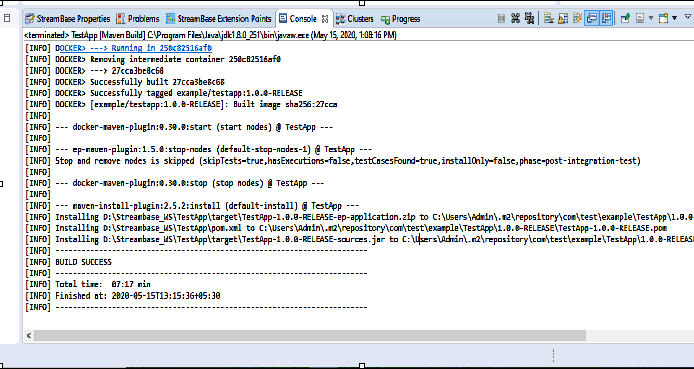


⦁ Now In the Project Explorer view, select and right-click the Streambase Application Project **(TestSBApp in my case)--> Run As>Maven Build...**

⦁ After that In the **Goals** field, enter **clean install** and Select the **Skip Tests** check box. Click Apply and Run.

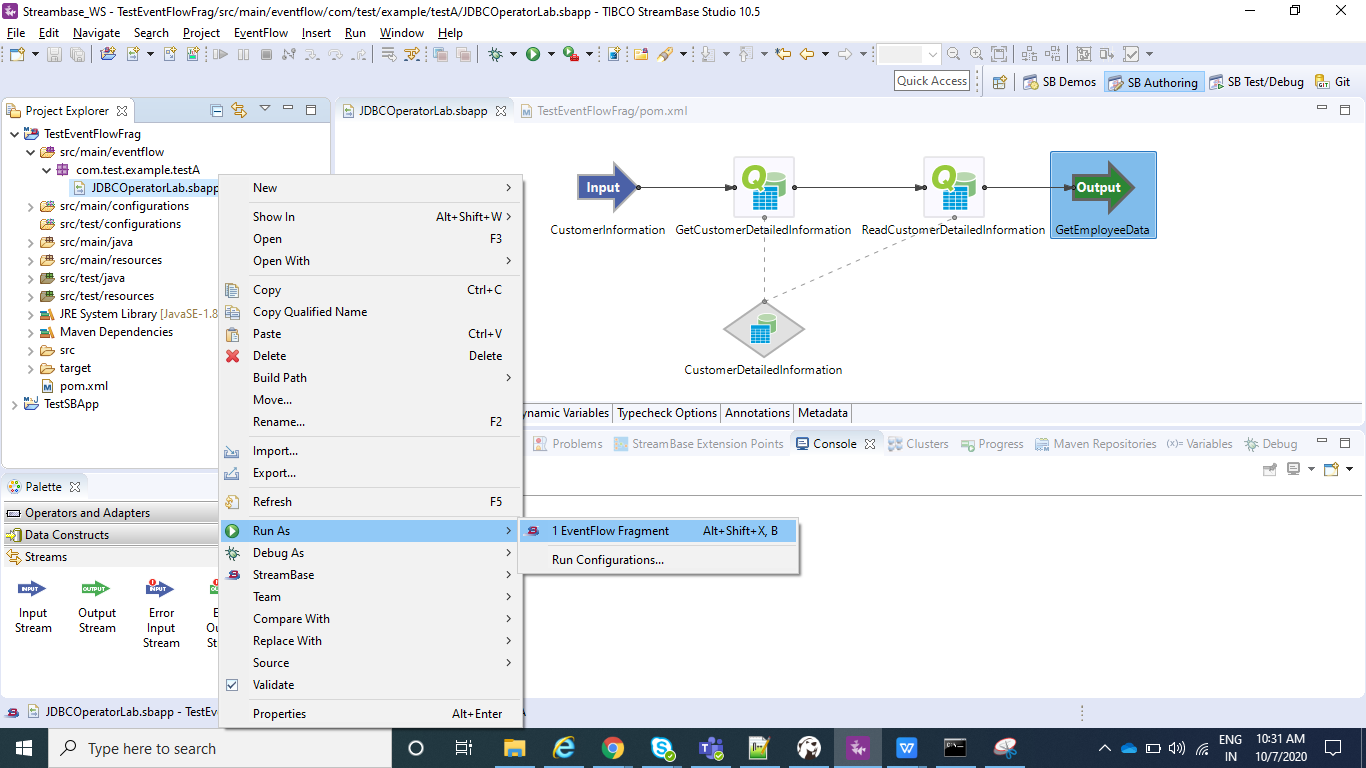




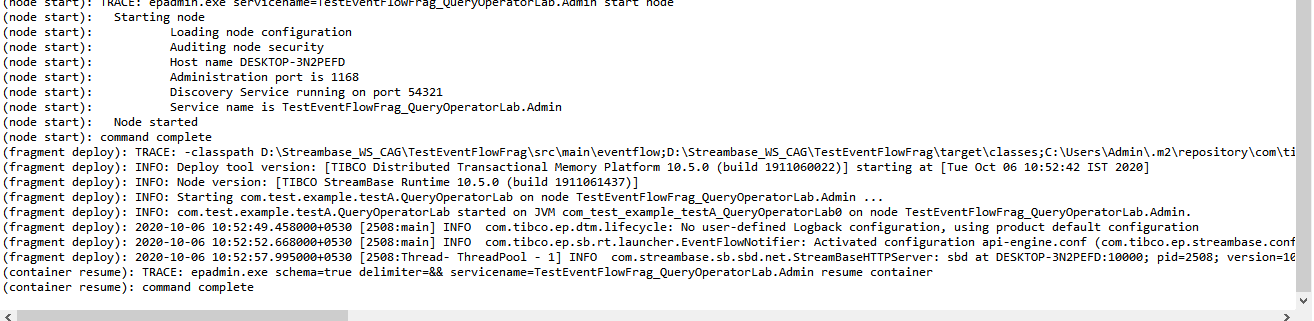
⦁ After the above step check the console and it should be showing BUILD SUCCESS message.

**Run the application in Studio:**

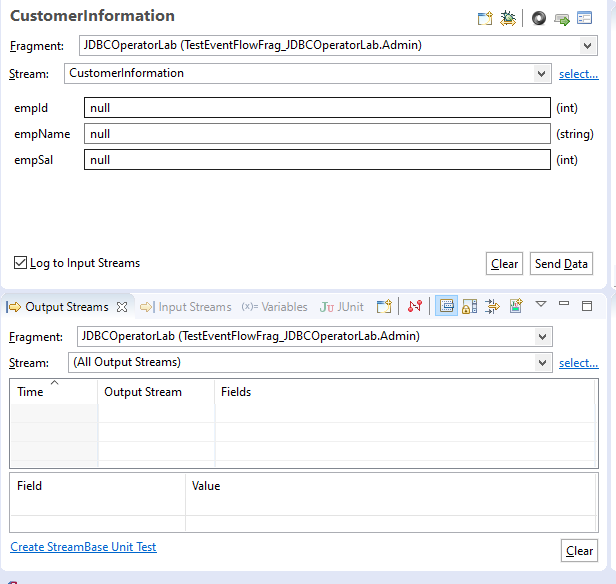
⦁ Right click on your .sbapp and select Run as EventFlowFragment as shown in below screenshot:

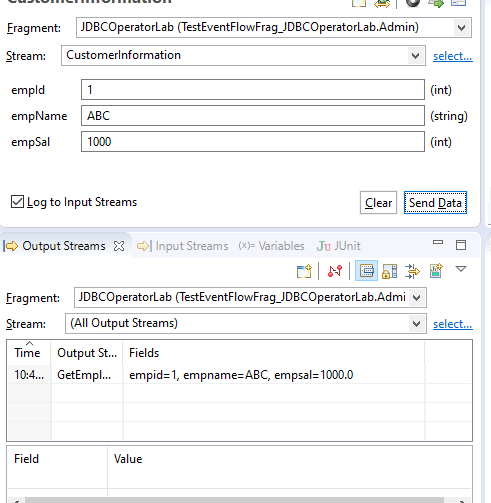


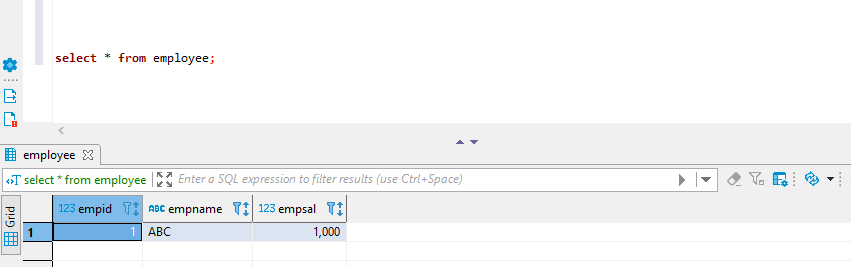
⦁ After you application is running you can see to the console logs as shown below:

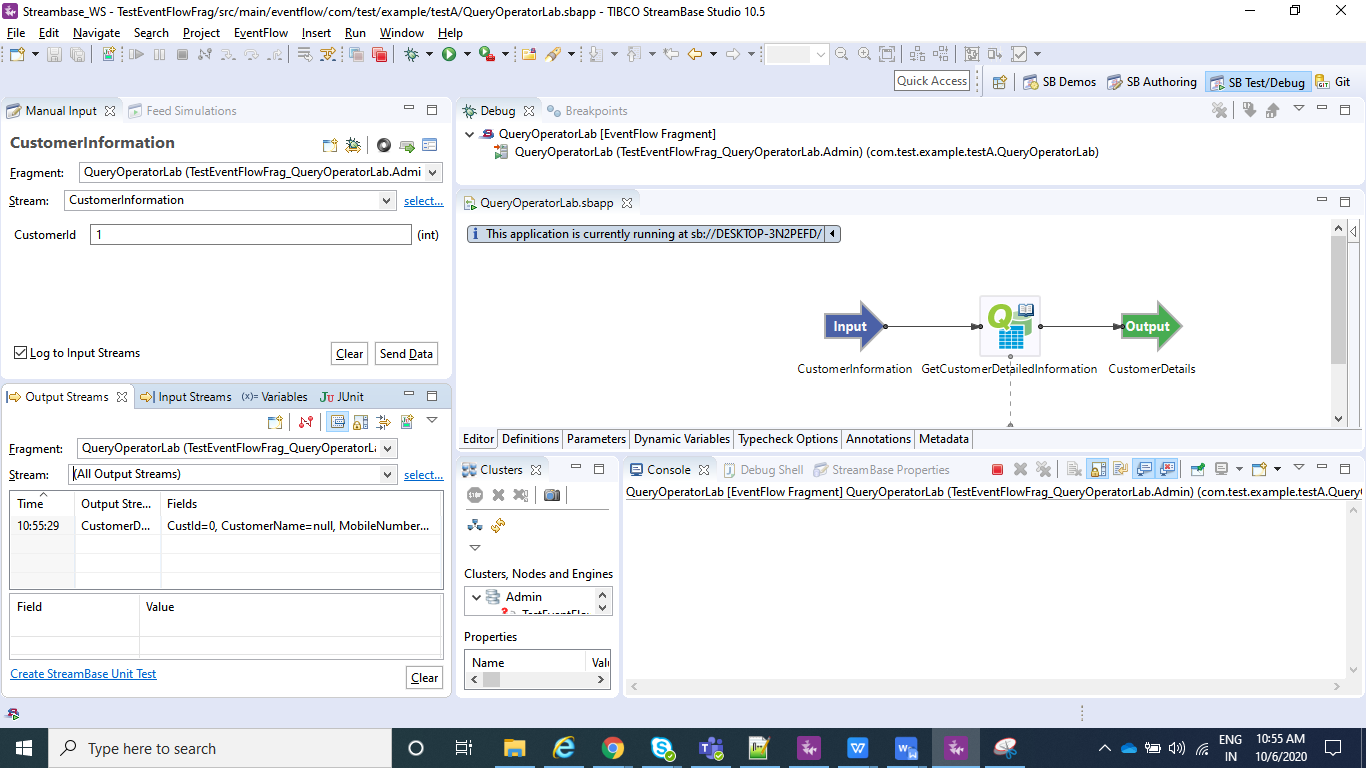


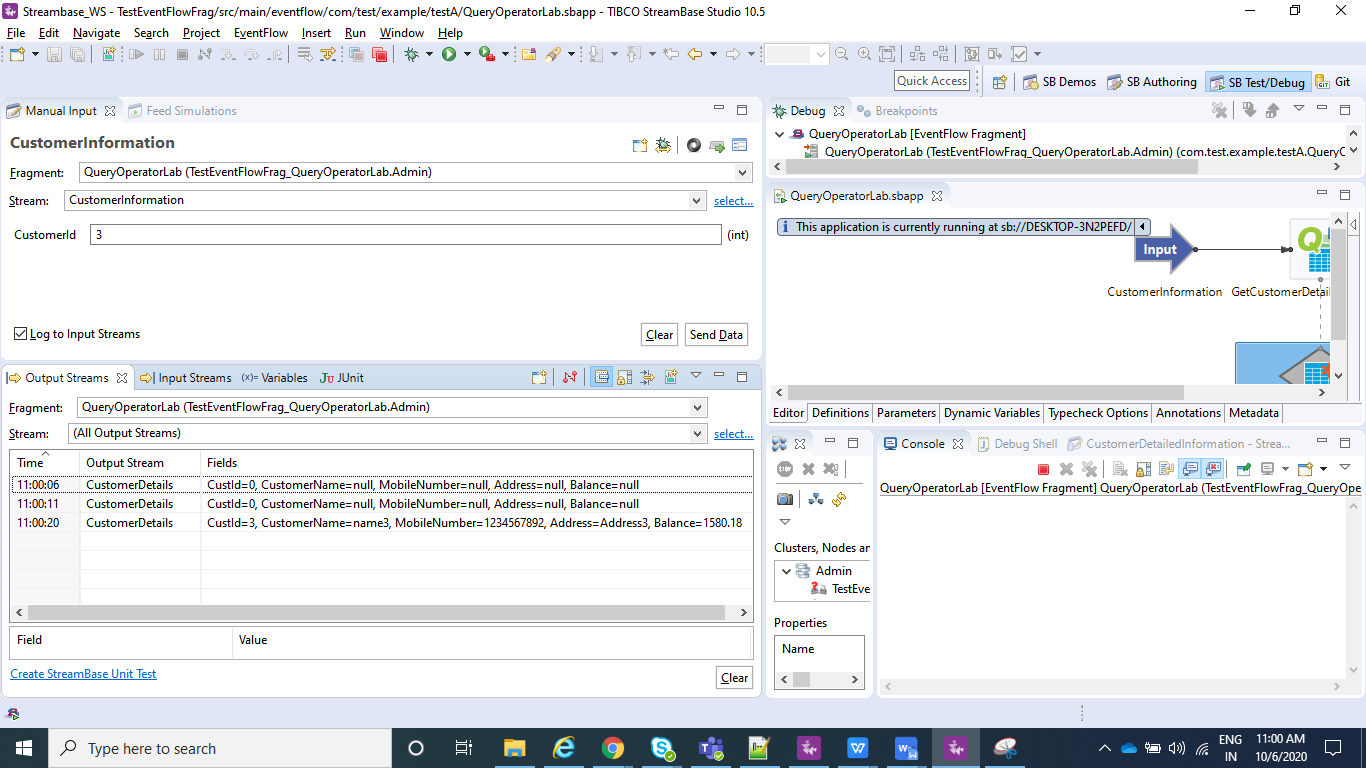
⦁ Now your application is running and same can be tested using Manual Input as shown in below screenshot.Provide the value for empId, empName, empSal and the same than can be seen in output stream and also in your database.





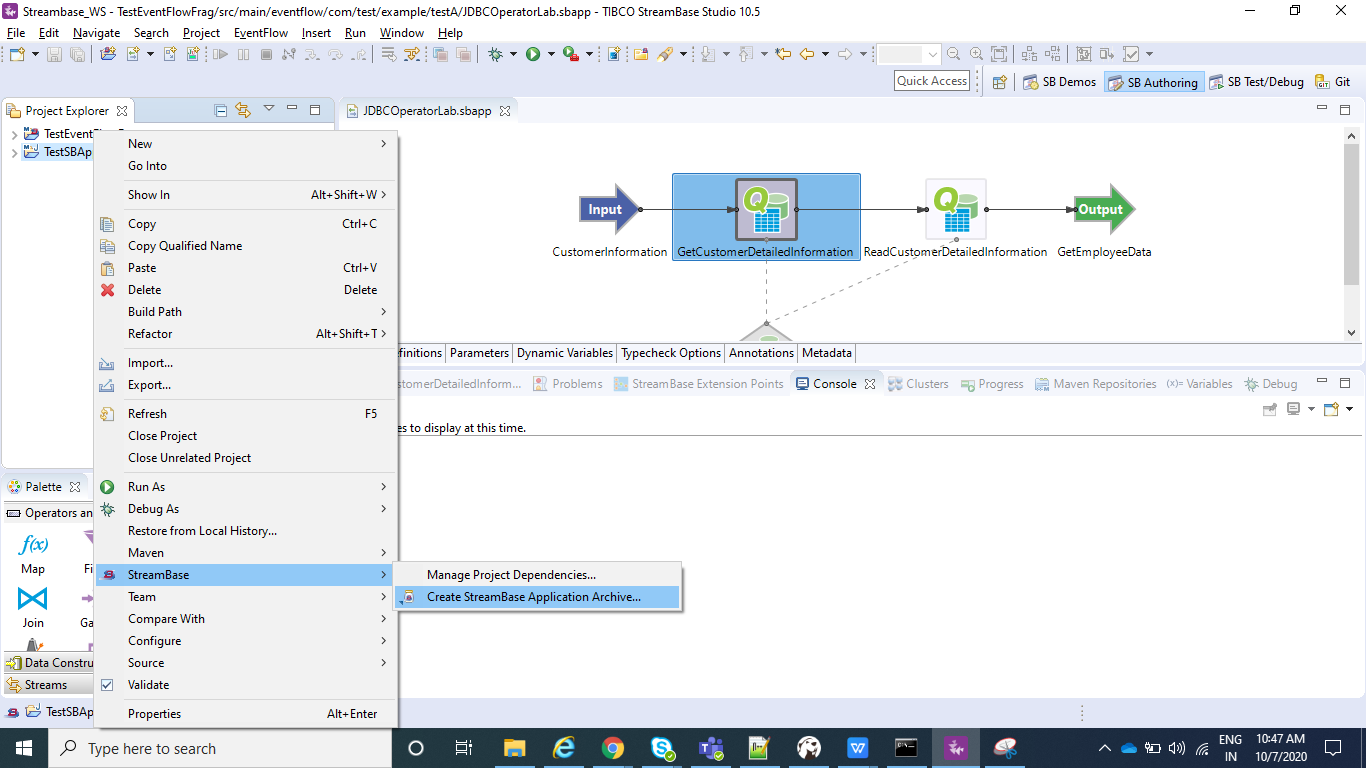


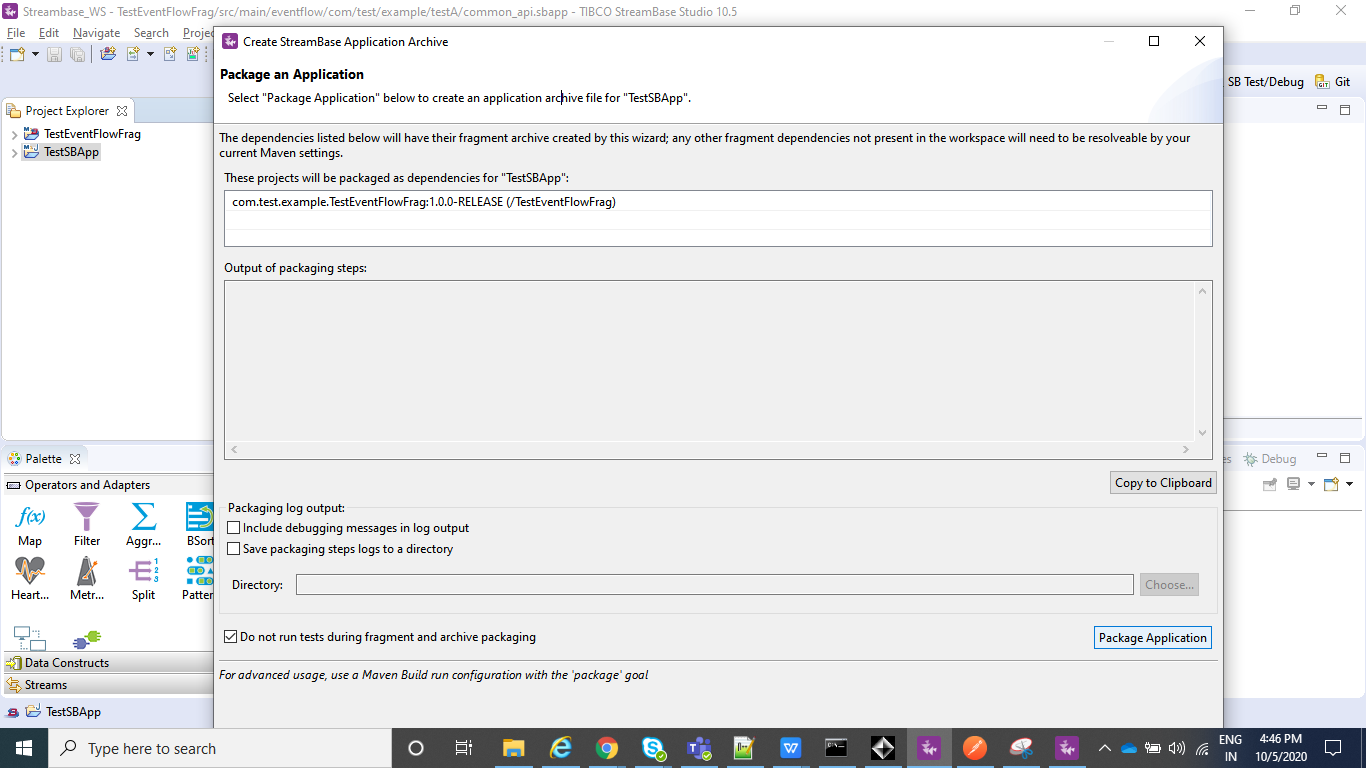




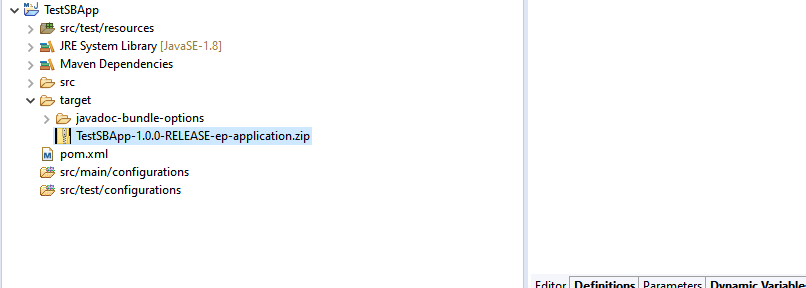
**Create the Application Archive:**

⦁ Right click on the Streambase Application(**TestSBApp** in my case) and than select Create Streambase Application Archive and than select Package Application**.**





⦁ After the above step is completed we can see a .zip file under our Streambase Application under target folder.



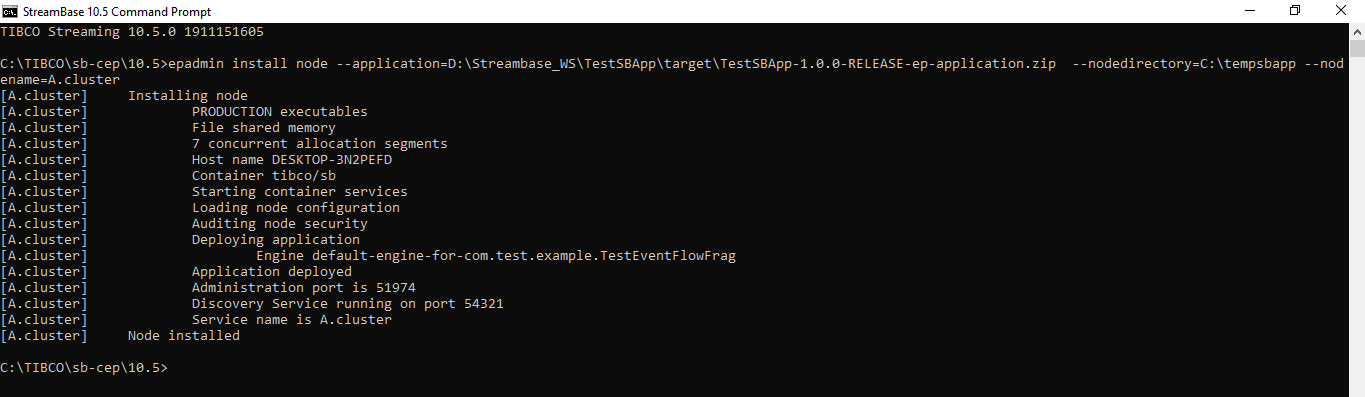
**Deploy and Run the Application:**

⦁ Open the Streambase CommandPrompt to Deploy and Run the application locally as shown in below screenshot.



⦁ Open the Streambase CommandPrompt to Deploy and Run the application locally as shown in below screenshot and run the below command where --**application=D:\Streambase\_WS\TestSBApp\target\TestSBApp-1.0.0-RELEASE-ep-application.zip** is the path of the zip file we just created above,  **--nodedirectory=C:\tempsbapp** is any temp directory created in C drive and **nodename=A.cluster** is the any random name assigned to the node which will be running.

**epadmin install node --application=D:\Streambase\_WS\TestSBApp\target\TestSBApp-1.0.0-RELEASE-ep-application.zip --nodedirectory=C:\tempsbapp --nodename=A.cluster**

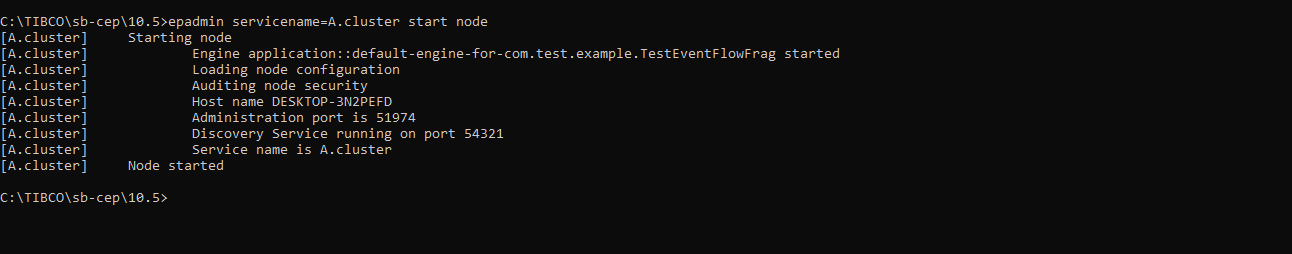


⦁ Now after your Node is installed, we can **start**, **stop** or **remove** node using below commands and once you have started the node you can test your application using postman in the same way we did before:

**epadmin servicename=A.cluster start node**

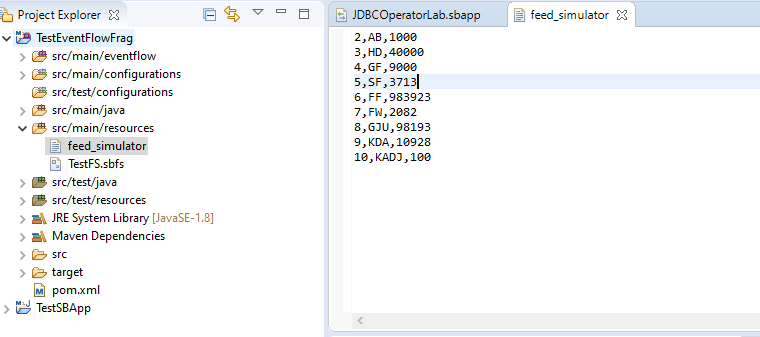
**epadmin servicename=A.cluster stop node**

**epadmin servicename=A.cluster remove node**

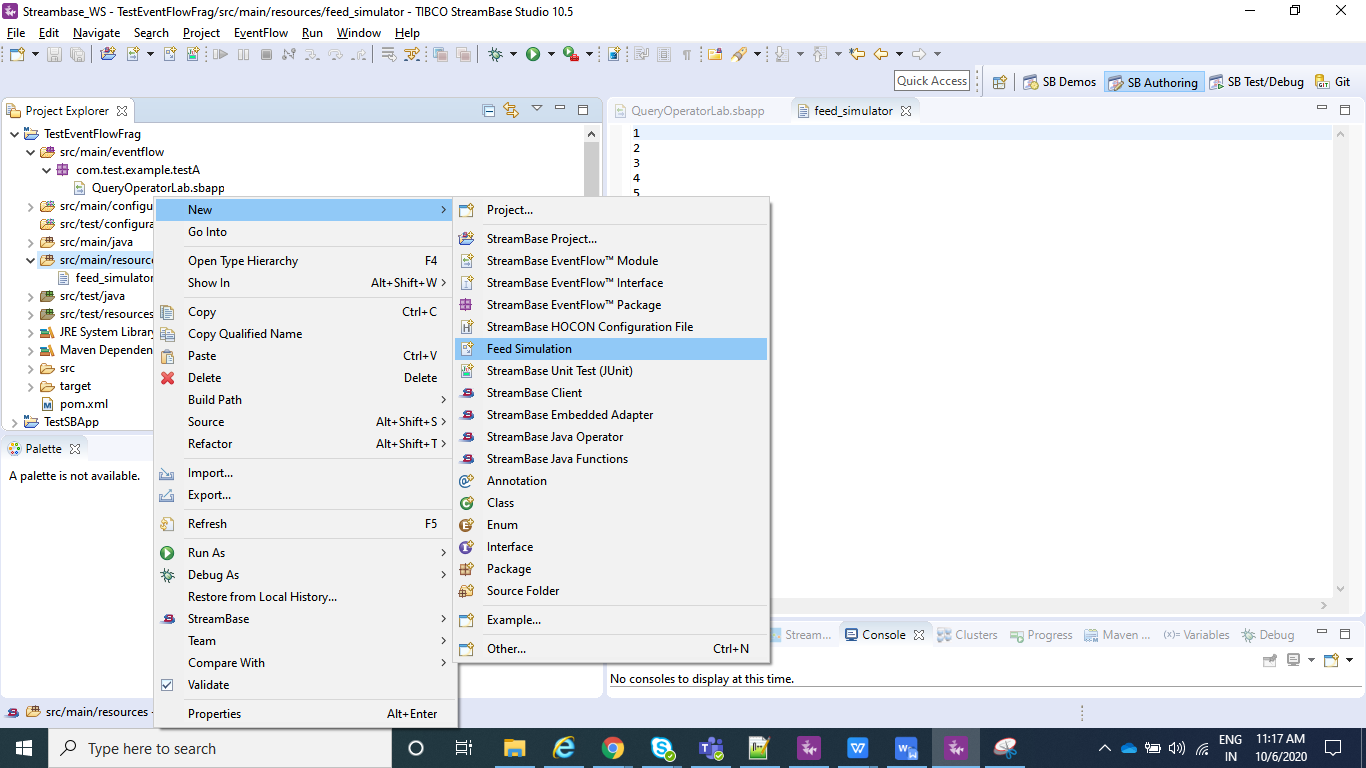


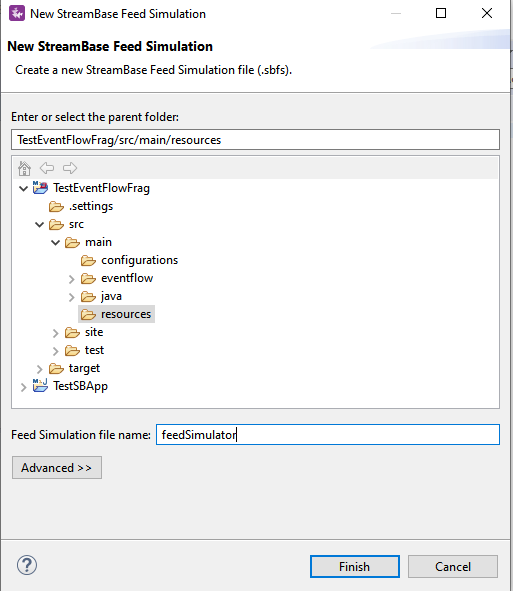
**Test the application (Using Feed Simulator file from command line):**

* Import a feed\_simulator.txt file with below entries under src/main/resources.

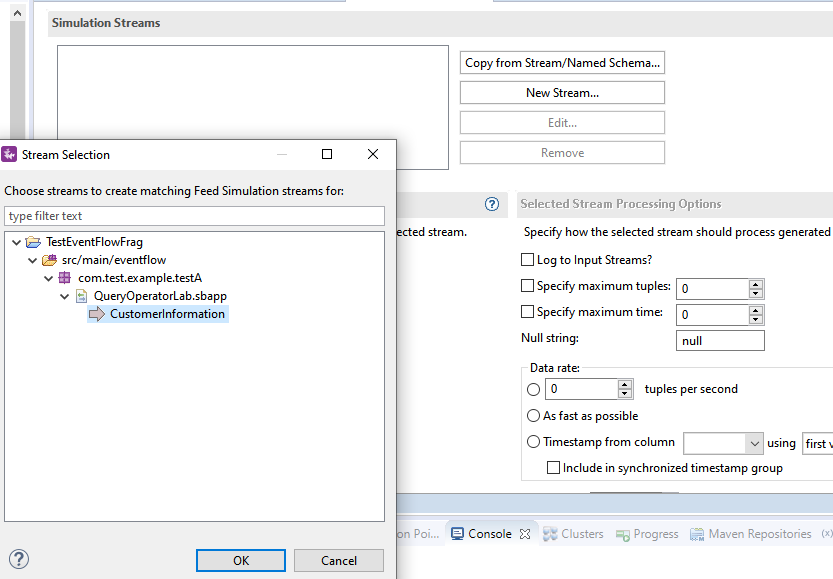


* To Configure feed simulator please follow below screenshots :

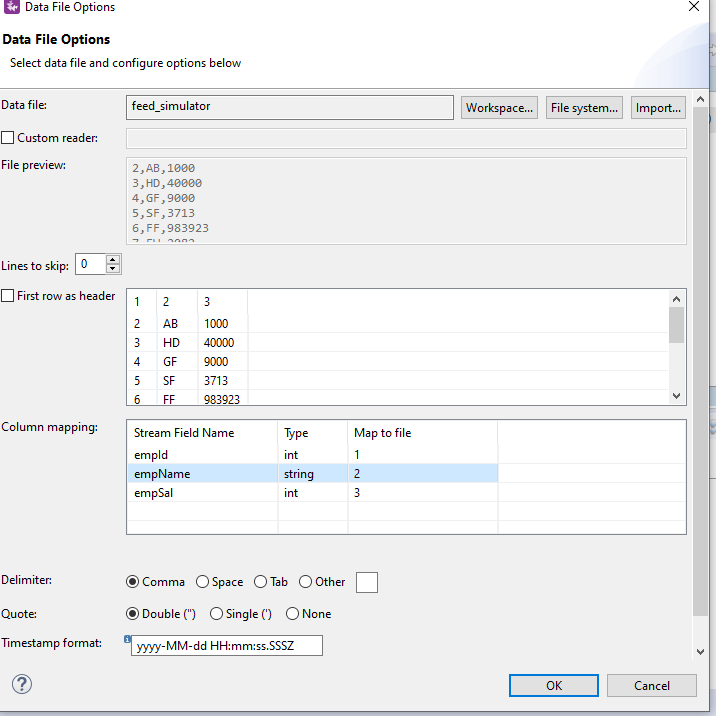




* Click on Copy from Stream/Named Schema and go to your Input Stream(CustomerInformation in my case) and click OK.



* Now select Data File option and select the file you just created , i.e, feed\_simulator.txt file where you have your data loaded.



* Create a configuration file which will contains information on which of your .sbapp need to run when an streambase application is deployed.Sample for the same is :

name = "sample-StreamBaseEngine-document"

type = "com.tibco.ep.streambase.configuration.sbengine"

version = "1.0.0"

configuration = {

StreamBaseEngine = {

jvmArgs = [

"-Xmx8g"

"-Xms512m"

"-XX:+UseG1GC"

"-XX:MaxGCPauseMillis=500"

"-XX:ConcGCThreads=1"

]

}

EventFlowDeployment = {

modules = [

{

moduleName="com.test.example.testA.JDBCOperatorLab"

containerName="JDBCOperatorLab"

}

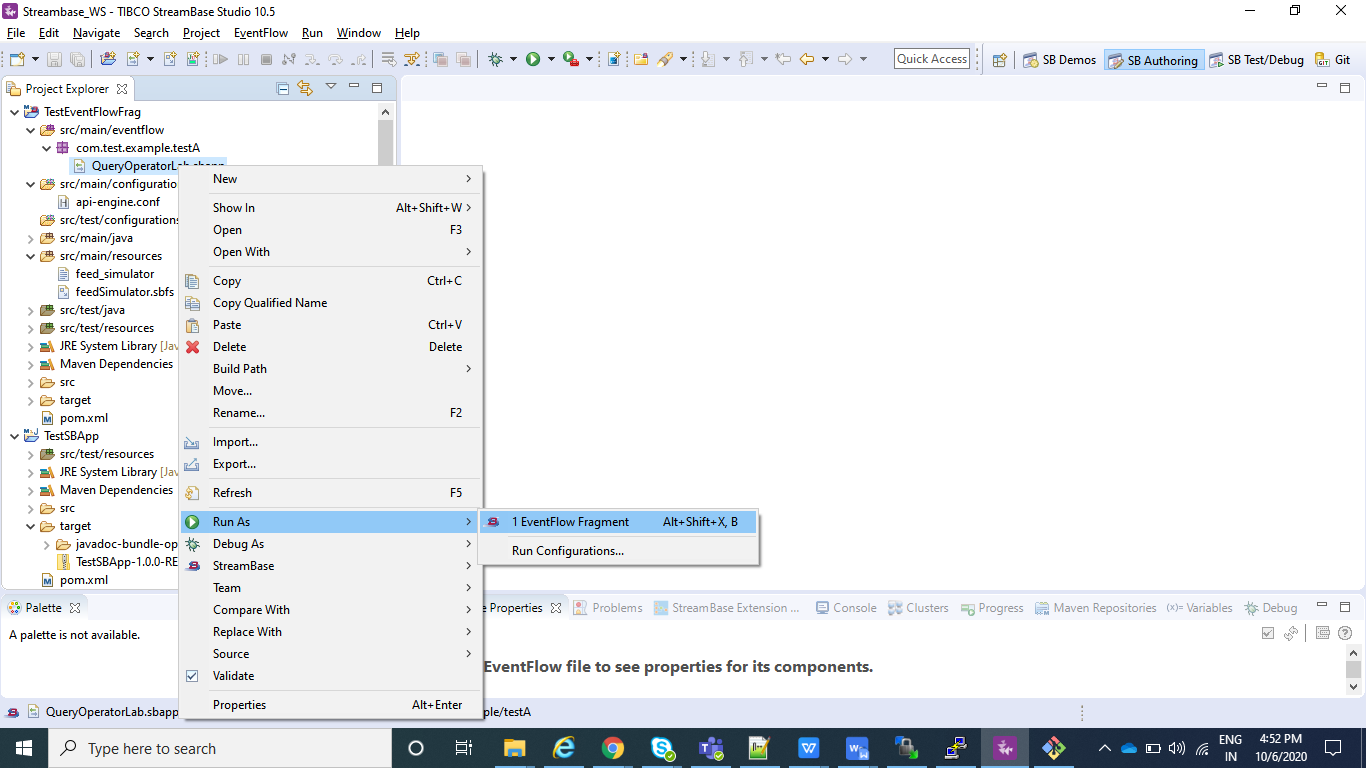
] }

}

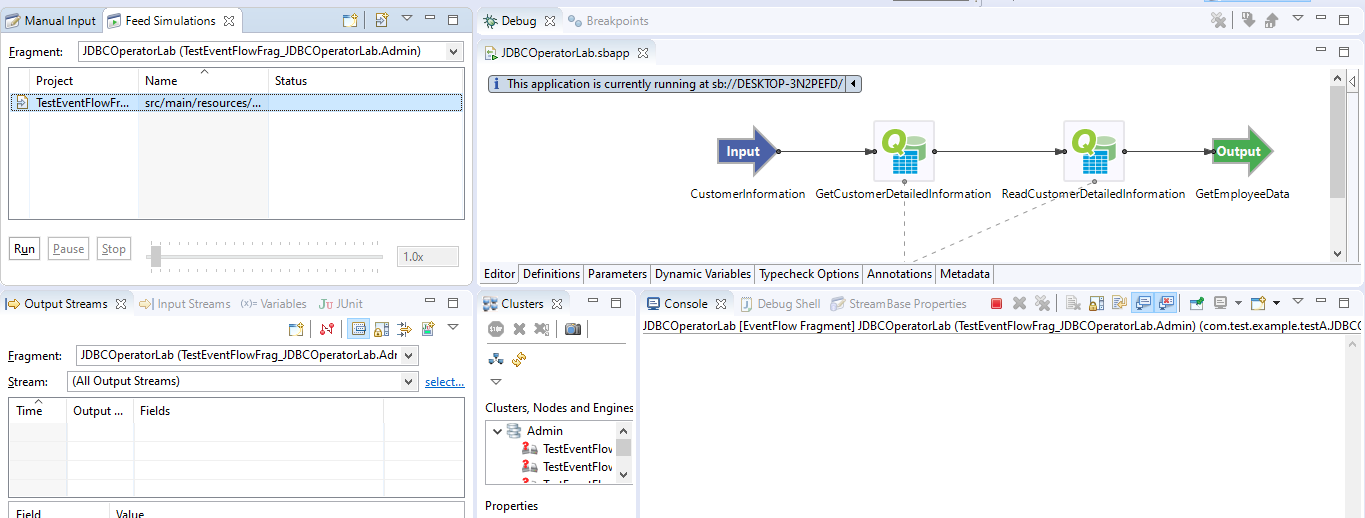
Note : moduleName="com.test.example.testA.JDBCOperatorLab" is nothing but my .sbapp which I wish to run when my application is deployed and containerName is just any name that you wish to give.

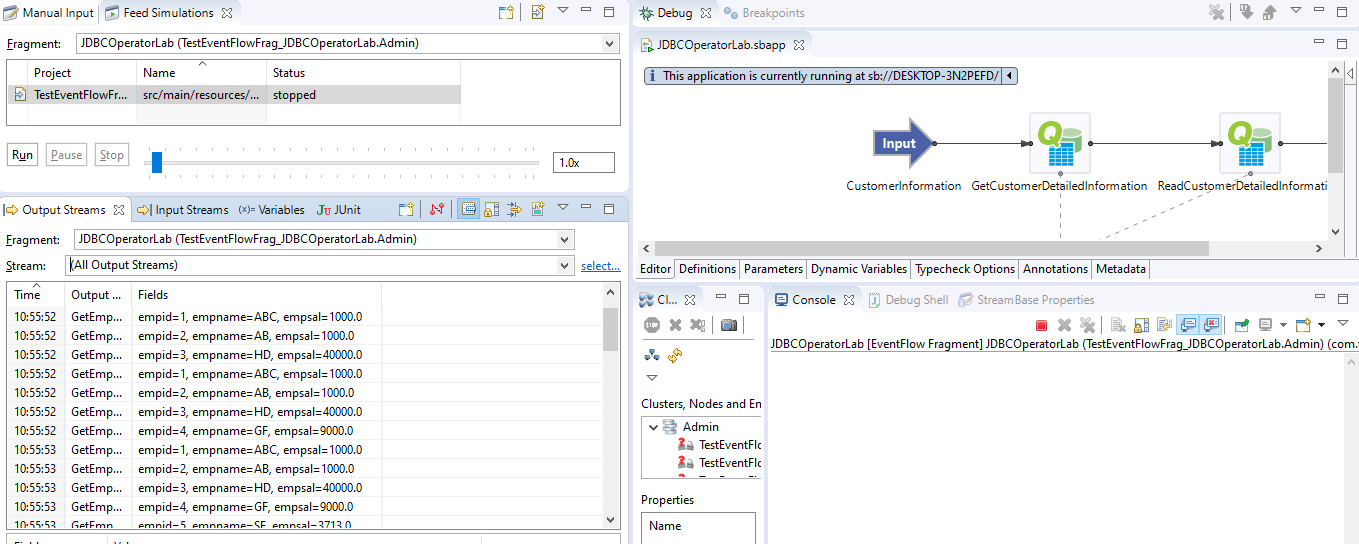


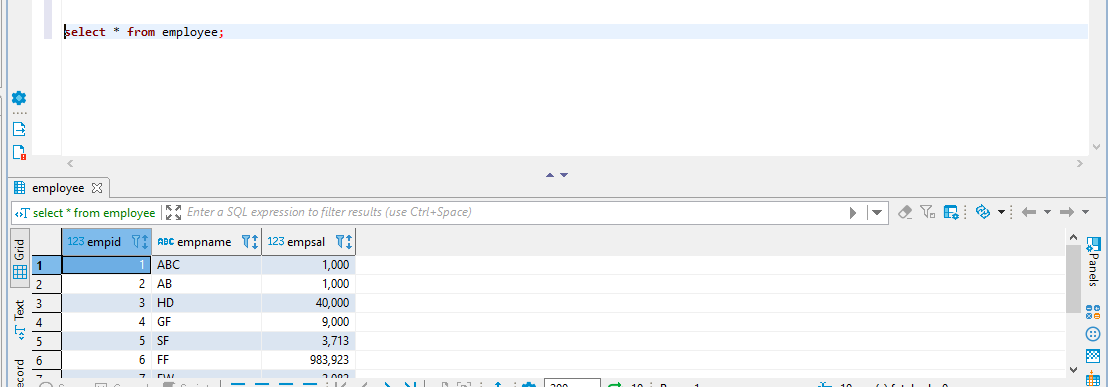
* Now make sure you do a clean install as explained above for the EventFlowFragment(TestEventFlowFrag in my case) and than for your Streambase Application(TestSBApp in my case) and than create Archive and Deploy the node as explained in above steps.
* In StreamBase command prompt navigate to TesteventFlowFrag project workspace location where your **feedSimulator.sbfs** is located and use **sbfeedsim** utility command and enter as below as **sbfeedsim feedSimulator.sbfs** and we can see Data from feed simulator.
* Also using the Streambase Studio we can run Feed Simulation.Just Right click on your .sbapp and run as eventFlowFragment.



* After your application is running select Feed Simulations instead of manual input as shown in below screenshot and create a new Feed Simulation and provide any name and after it is created configure in the same way we configured in above steps for feedSimulator.sbfs and save and than just do Run.







* As you can see data is inserted again and again and this is because we have not defined any primary key in our database table.