Practical 4

ETL Transformation with Pentaho.

```
SQL> Create table
Emp56( Emp_no
numeric(5), fname
varchar2(10),
lname varchar2(10),
salary numeric(5),
comm numeric(5)
);
Table created.
```

Now, enter 10 values:

```
SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (1,
'Ashutosh', 'Mazumdar', 50000, 2000);

1 row created

SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (2, 'Sankalp',
'Shukla', 60000, 2500);

1 row created.

SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (3, 'Aman',
'Mishra', 55000, 2200);

1 row created.

SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (4, 'Anish',
'Jadhav', 52000, 2100);
```

```
1 row created
SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (5, 'Rutuja',
'Khangal', 58000, 2400);

1 row created.


SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (6, 'Owais',
'Siddiqui', 53000, 2150);


1 row created.

SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (7, 'Aman',
'Prasad', 59000, 2450);

1 row created.


SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (8, 'Ganesh',
'Mahind', 54000, 2200);


1 row created.


SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (9, 'Sanjana',
'Pradhan', 56000, 2250);


1 row created.


SQL> INSERT INTO emp25 (emp_no, fname, lname, salary, comm) VALUES (10, 'Riya',
'Kholi', 57000, 2300);


1 row created.
```

Now show the values of the table-

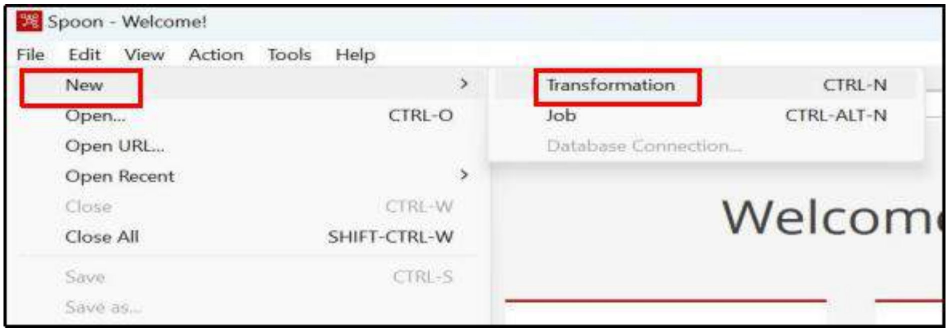
```
SQL> SELECT * FROM emp25;
```

```
    EMP_NO FNAME  LNAMESALARY         COMM




      1 Ashutosh  Mazumdar         50000  2000

         2 Sankalp        Shukla            60000          2500

         3 Aman           Mishra            55000          2200

         4 Anish         Jadhav            52000          2100

         5 Rutuja        Khangal            58000          2400

         6 Owais          Siddiqui          53000          2150

         7 Aman           Prasad           59000          2450

         8 Ganesh         Mahind             54000          2200

         9 Sanjana        Pradhan           56000          2250

        10 Riya          Kholi            57000          2300


10 rows selected.
```

TRANSFORMATION 1:

Step 1: In the data integration folder open *"Spoon (Windows Batch File)"*.

Step 2: Go to File-New-Transformation.



Step 3: Import SQL table to pentaho-

Design tab-Input folder-Drag and drop Table input

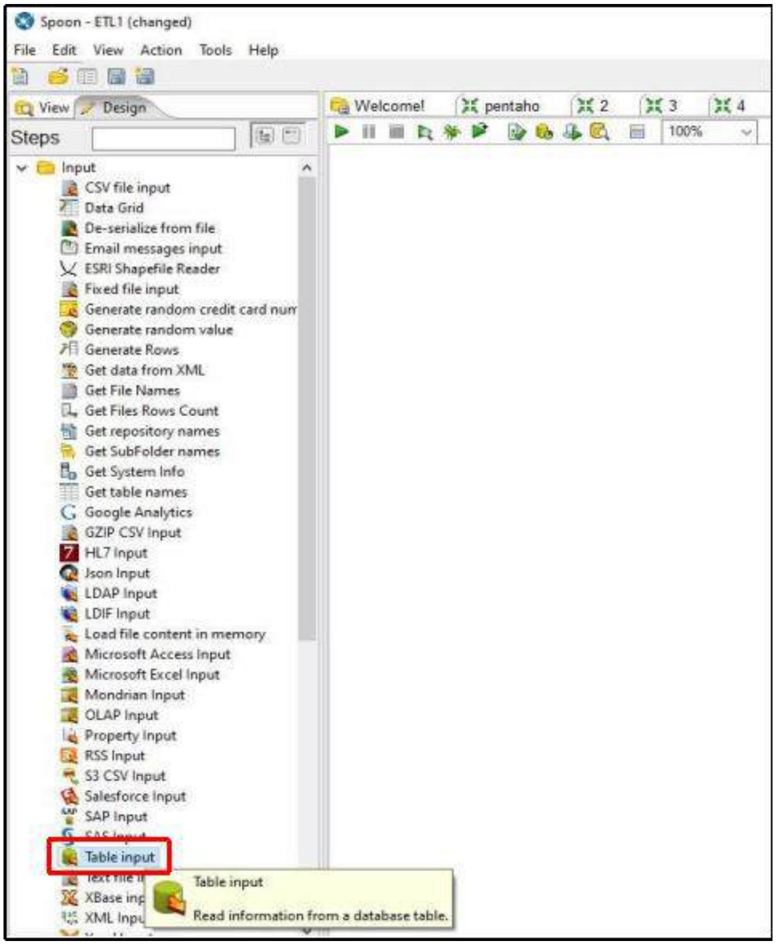
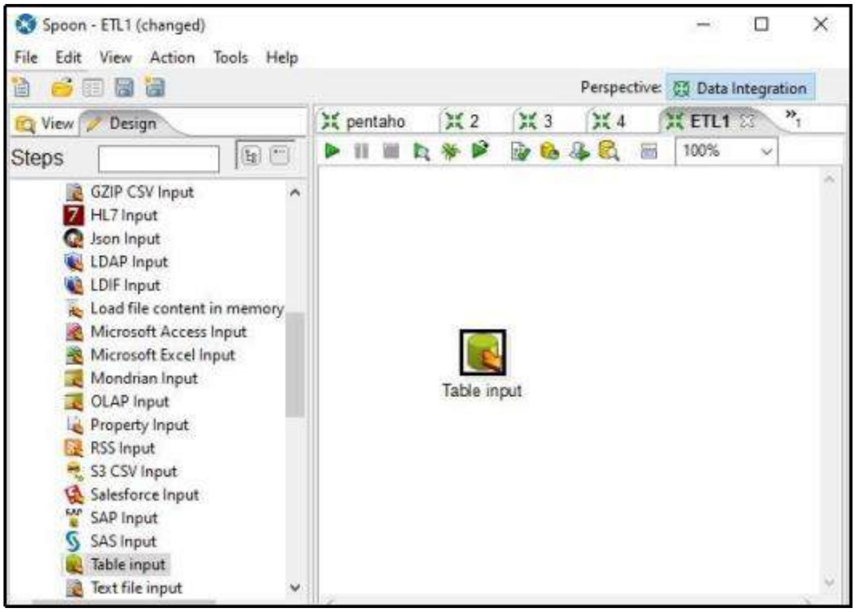


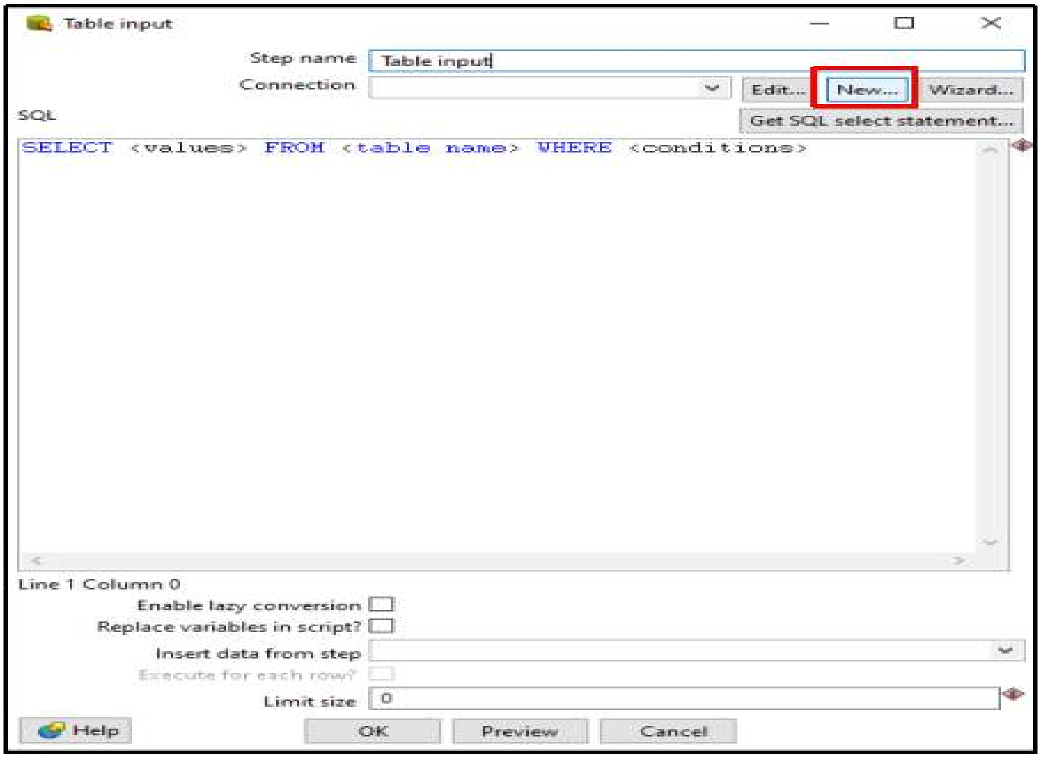
Table input

Read information from a database table.

Double click on table input



Click on New



Step name: Table input

```
SELECT <values> FROM <table name> WHERE <conditions>
```

Line 1 Column 0

Enable lazy conversion □

Replace variables in script? □

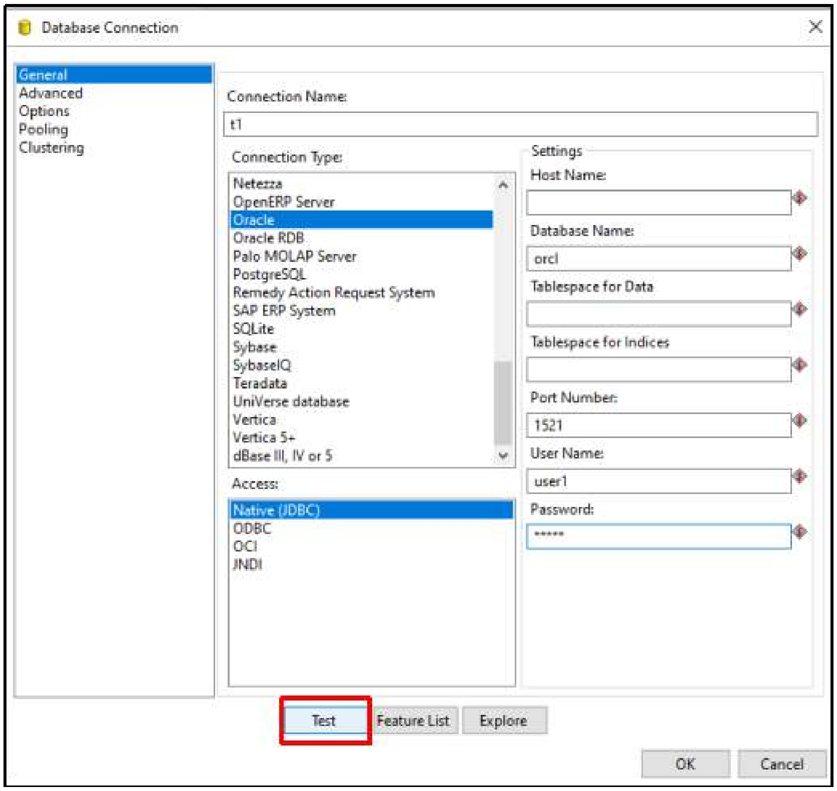
Insert data from step

Execute for each row? □

Limit size 0

Connect to the Database:

Fill in the details as below. Here enter User Name and Password same as your database username and password. Then click on Test.



Connection Name: t1

Host Name:

Database Name: orcl

Tablespace for Data:

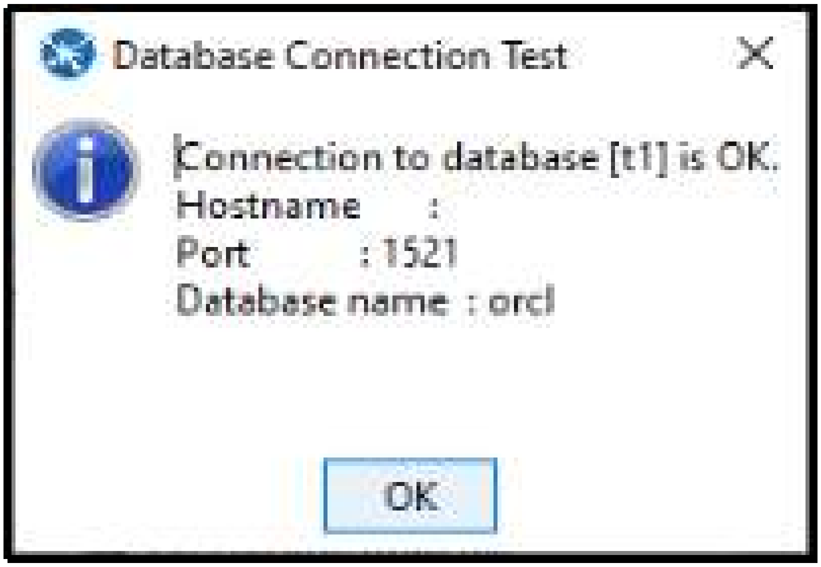
Tablespace for Indices:

Port Number: 1521

User Name: user1

Password: *****

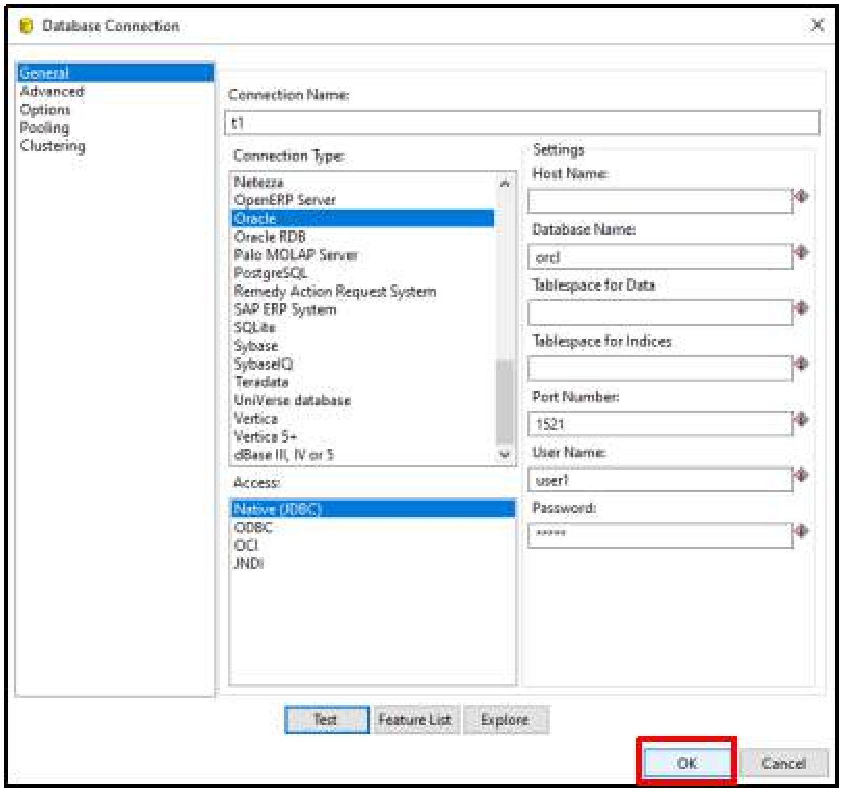
Click OK→



Database Connection Test

```
Connection to database [t1] is OK.
Hostname       :
Port           : 1521
Database name  : orcl
```

again Ok



Connection Name: t1

Connection Type: Oracle

Access: Native (JDBC)

Host Name:

Database Name: orcl

Tablespace for Data:

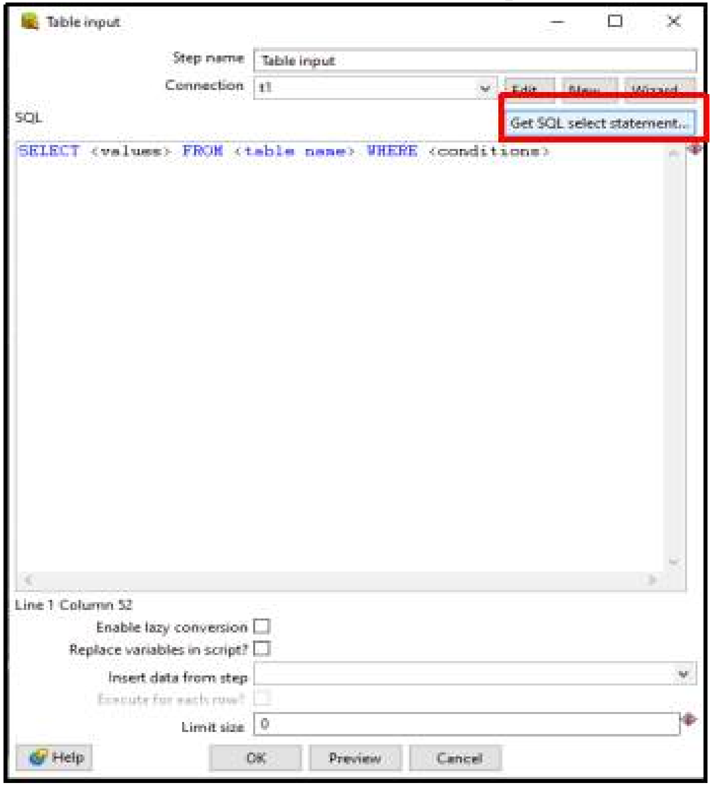
Tablespace for Indices:

Port Number: 1521

User Name: user1

Password: *****

Get SQL select statement… in table input window.



Step name: Table input

Connection: t1

SQL

```
SELECT <values> FROM <table name> WHERE <conditions>
```

Line 1 Column 52

Enable lazy conversion □

Replace variables in script? □

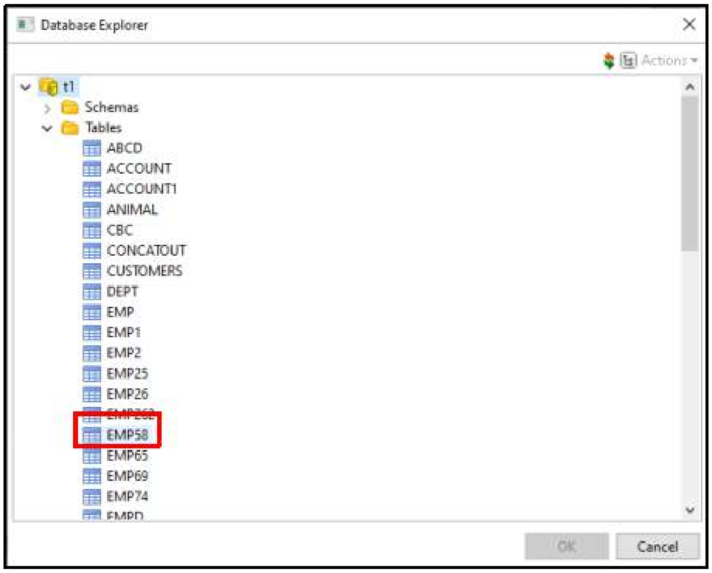
Insert data from step

Execute for each row? □

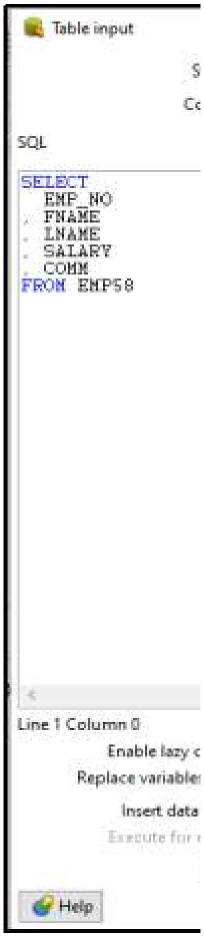
Limit size 0

Import table:

In t1, under tables, select the required table.



Click on OK than Click on Yes

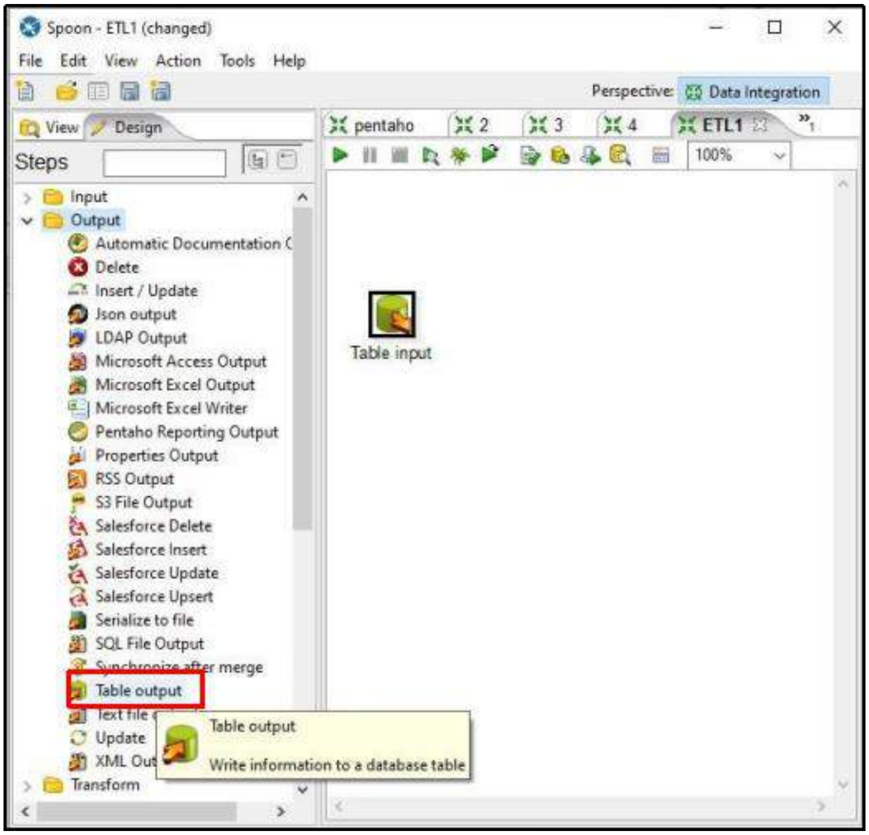


```
SELECT
  EMP_NO
, FNAME
, LNAME
, SALARY
, COMM
FROM EMP58
```

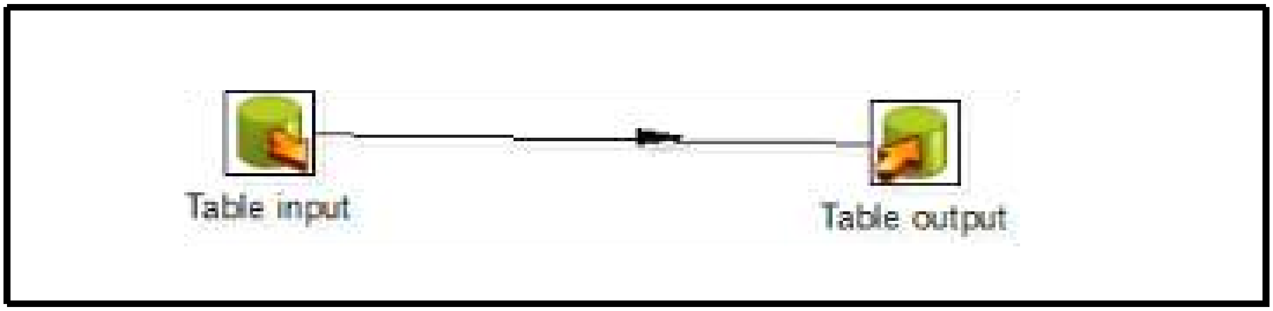
Preview in Table input window and click on OK

Step 4: Show output:

Drag and Drop Table Output



Hold the mouse pointer on Table input and select and drag the output connector to the Table output.



Double Click on Table Output.

In the Table Output Window, give name to the Target table, check the check boxes and click on Get fields.

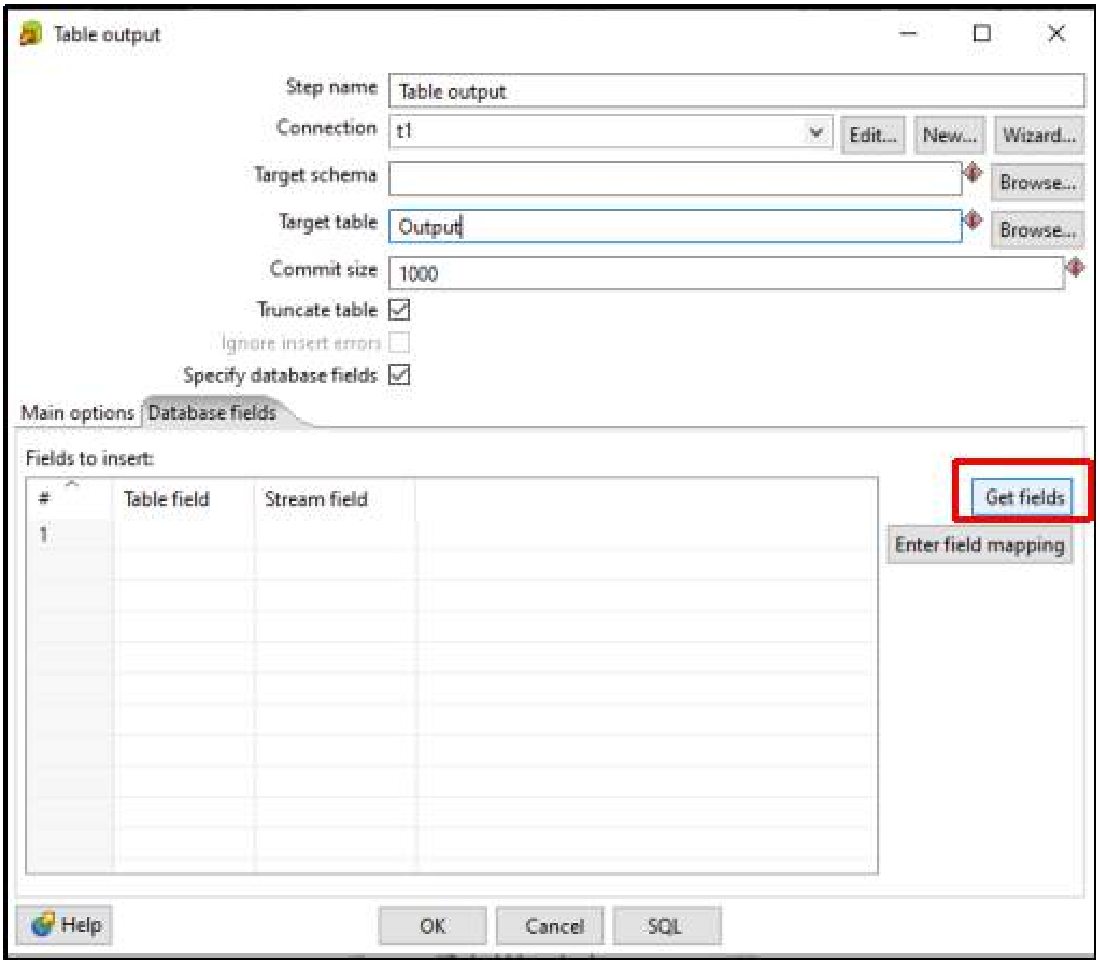


Table output

Step name: Table output

Connection: t1

Target schema:

Target table: Output

Commit size: 1000

Truncate table ☑

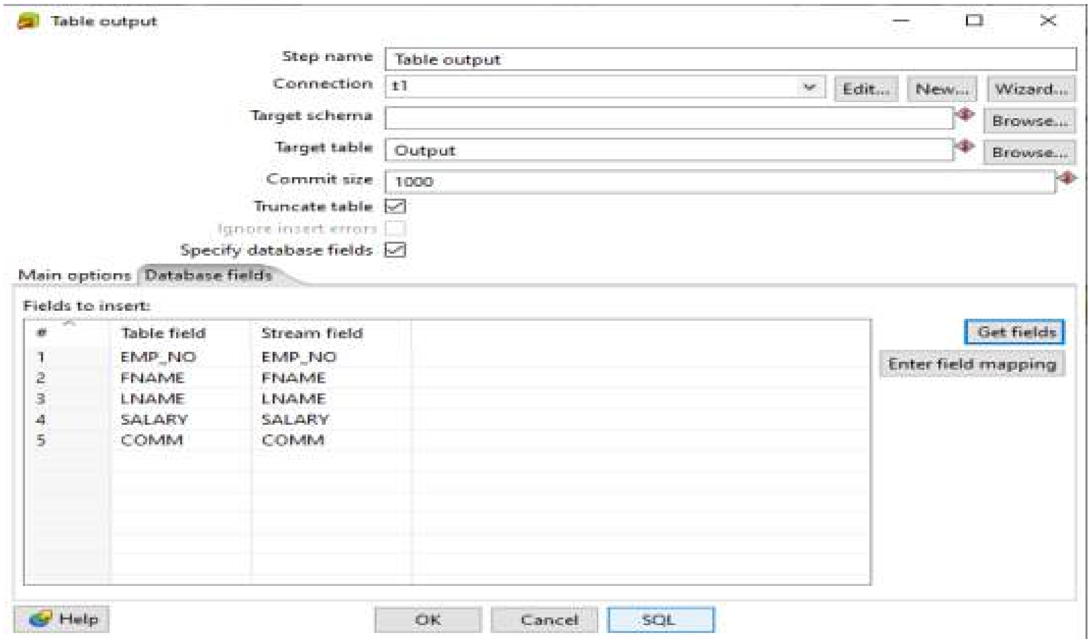
Ignore insert errors □

Specify database fields ☑

Fields to insert:

#	Table field	Stream field
1		

Click on SQL.



Step name: Table output

Connection: t1

Target schema:

Target table: Output

Commit size: 1000

Truncate table ☑

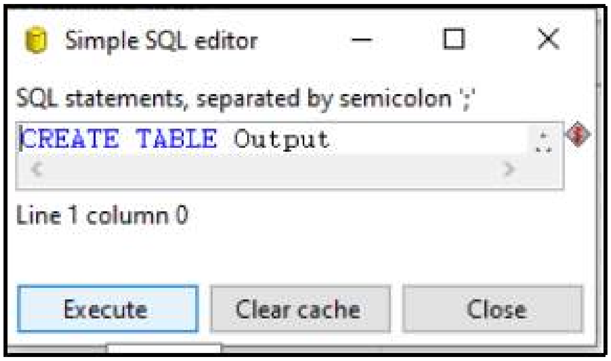
Ignore insert errors □

Specify database fields ☑

Fields to insert:

#	Table field	Stream field
1	EMP_NO	EMP_NO
2	FNAME	FNAME
3	LNAME	LNAME
4	SALARY	SALARY
5	COMM	COMM

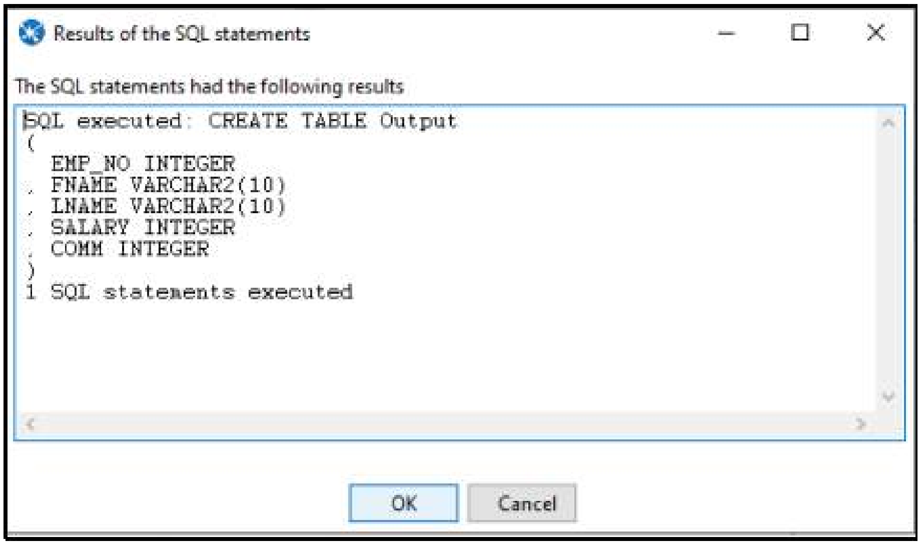
Click on Execute→OK



SQL statements, separated by semicolon ';'

```
CREATE TABLE Output
```

Line 1 column 0

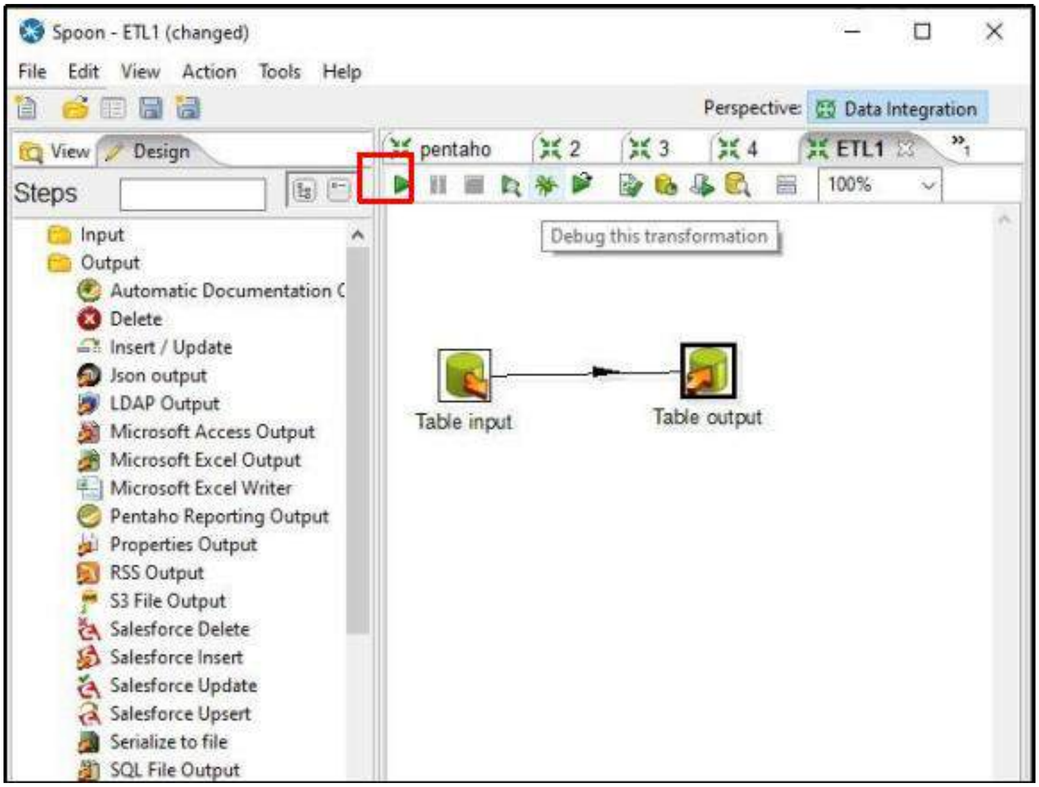


The SQL statements had the following results

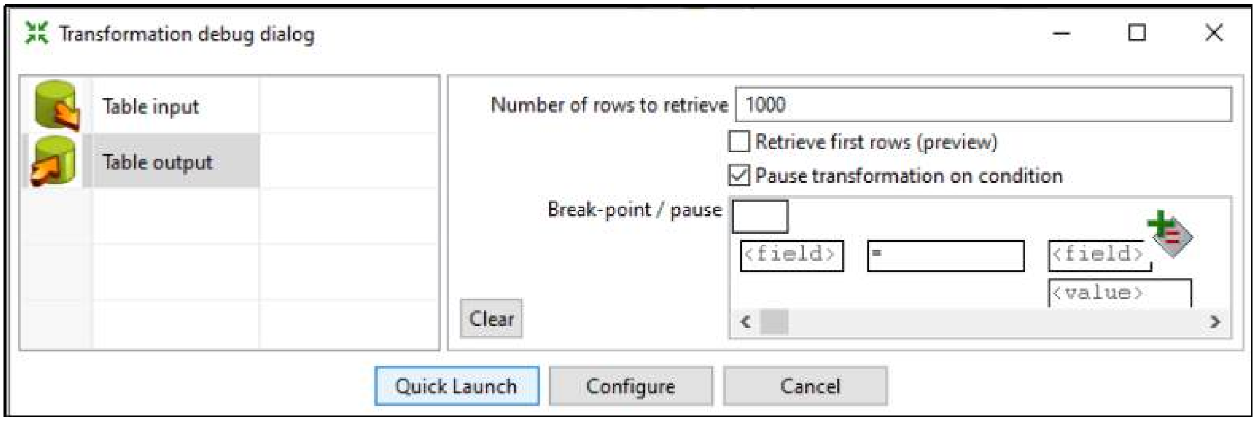
```
SQL executed: CREATE TABLE Output
(
  EMP_NO INTEGER
, FNAME VARCHAR2(10)
, LNAME VARCHAR2(10)
, SALARY INTEGER
, COMM INTEGER
)
1 SQL statements executed
```

Click on Close →OK

Then click on Debug this transformation.



Click on Quick Launch.



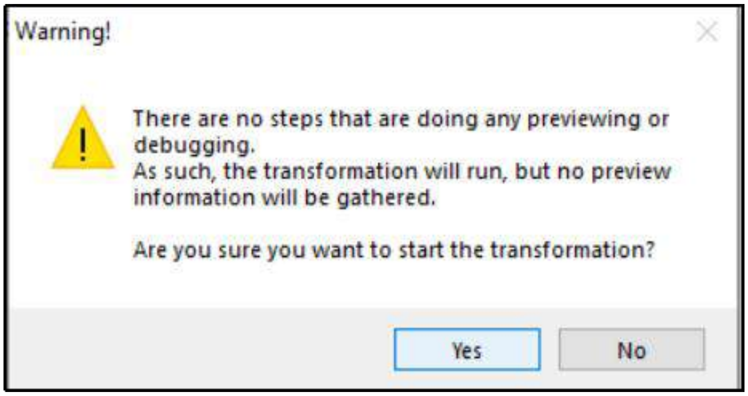
Number of rows to retrieve 1000

□ Retrieve first rows (preview)

☑ Pause transformation on condition

Break-point / pause <field> = <field> <value>

Click on Yes.

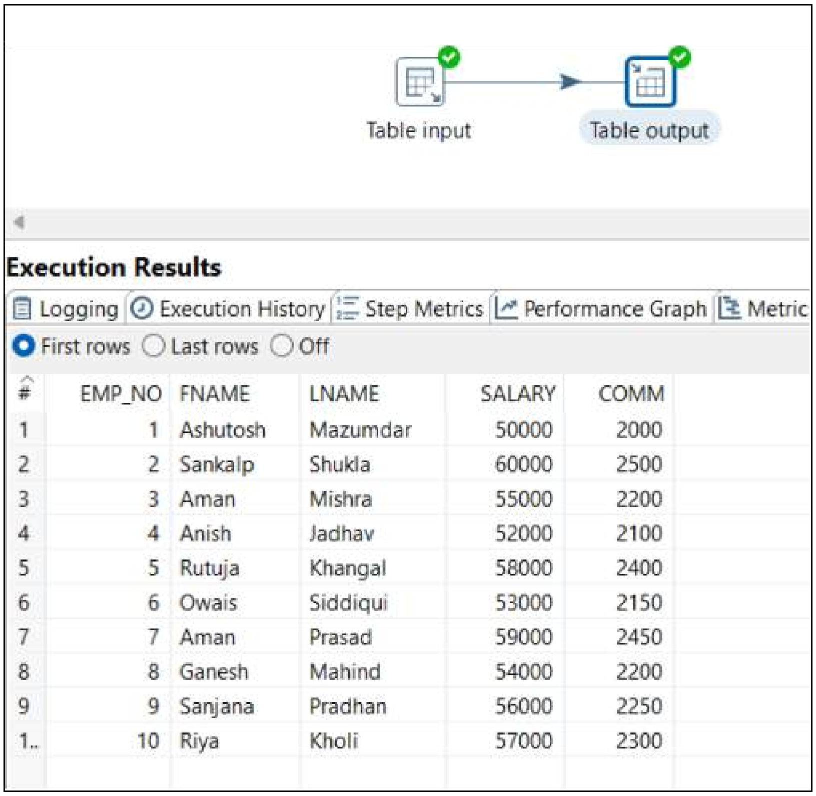


Warning!

There are no steps that are doing any previewing or debugging.
As such, the transformation will run, but no preview information will be gathered.

Are you sure you want to start the transformation?

If the Transformation is successful, you will see green ticks.



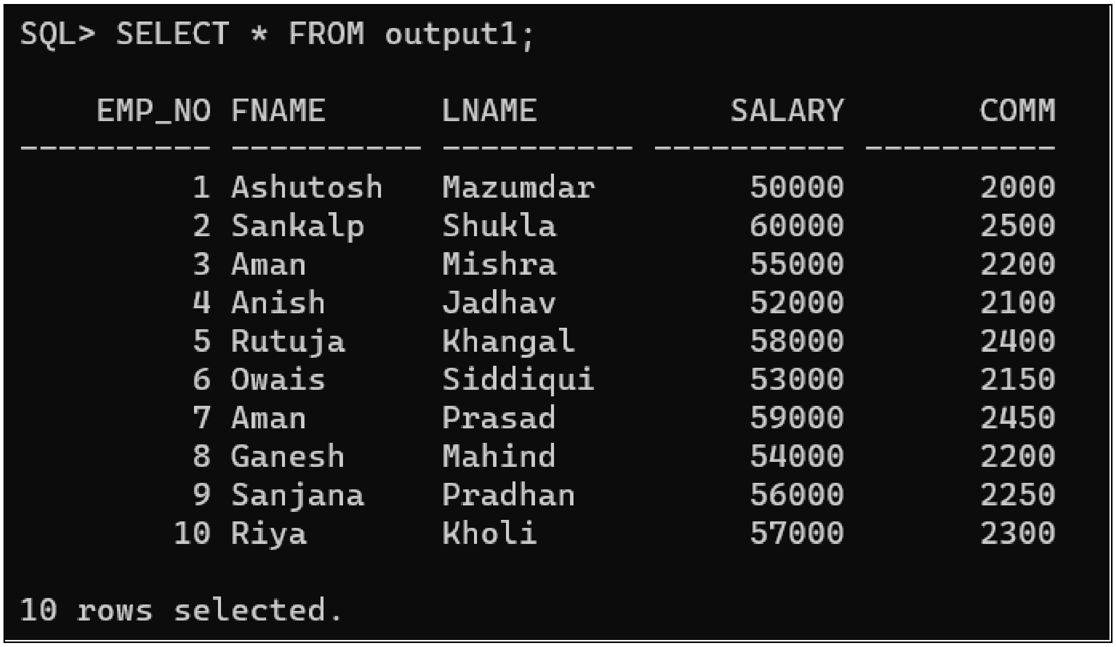
Execution Results

● First rows ○ Last rows ○ Off

#	EMP_NO	FNAME	LNAME	SALARY	COMM
1	1	Ashutosh	Mazumdar	50000	2000
2	2	Sankalp	Shukla	60000	2500
3	3	Aman	Mishra	55000	2200
4	4	Anish	Jadhav	52000	2100
5	5	Rutuja	Khangal	58000	2400
6	6	Owais	Siddiqui	53000	2150
7	7	Aman	Prasad	59000	2450
8	8	Ganesh	Mahind	54000	2200
9	9	Sanjana	Pradhan	56000	2250
1..	10	Riya	Kholi	57000	2300

Close → close
You will also see the table created in the database with the name same as the target table table name.

Step 5: Run query in SQL Plus.



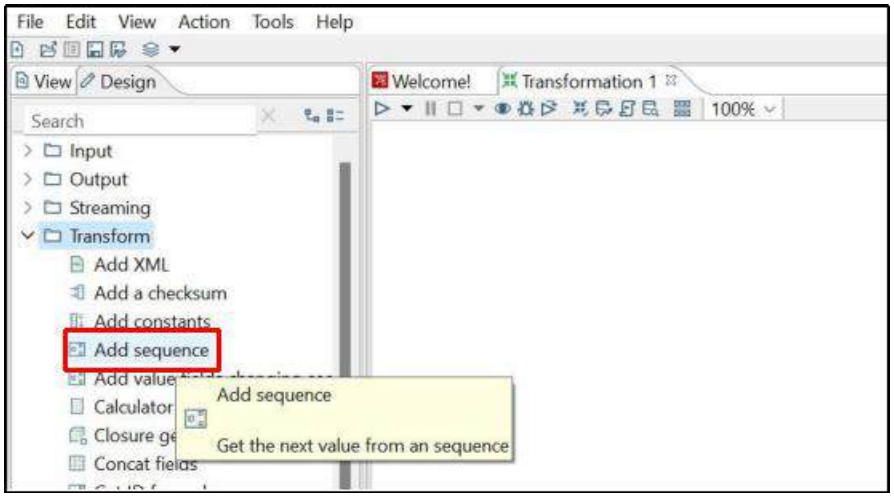
```
SQL> SELECT * FROM output1;

    EMP_NO FNAME      LNAME          SALARY       COMM
---------- ---------- ---------- ---------- ----------
         1 Ashutosh   Mazumdar        50000       2000
         2 Sankalp    Shukla          60000       2500
         3 Aman       Mishra          55000       2200
         4 Anish      Jadhav          52000       2100
         5 Rutuja     Khangal         58000       2400
         6 Owais      Siddiqui        53000       2150
         7 Aman       Prasad          59000       2450
         8 Ganesh     Mahind          54000       2200
         9 Sanjana    Pradhan         56000       2250
        10 Riya       Kholi           57000       2300

10 rows selected.
```

TRAMSFORMATION 2: Add sequence.

Step 1: Repeat Steps 2 and 3 from TRANSFORMATION 1.

Step 2: Perform transformation (Add sequence).

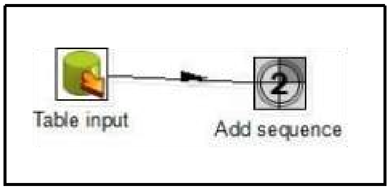
Drag and drop *Add Sequence* from the transform folder under the *Design* tab.



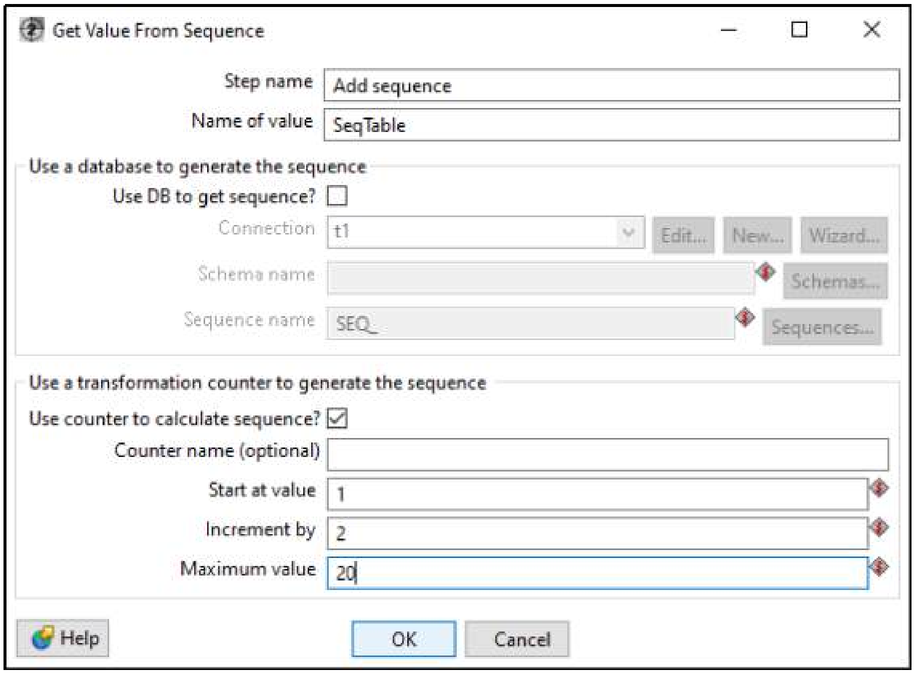
Add sequence

Get the next value from an sequence

Hold the mouse pointer on *Table input* and select and drag the output connector to the *Add sequence*.



Double click on Add sequence and fill in the details as shown below→Click on OK.



Step name: Add sequence

Name of value: SeqTable

Use a database to generate the sequence

Use DB to get sequence? □

Connection: t1

Schema name:

Sequence name: SEQ_

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☑

Counter name (optional):

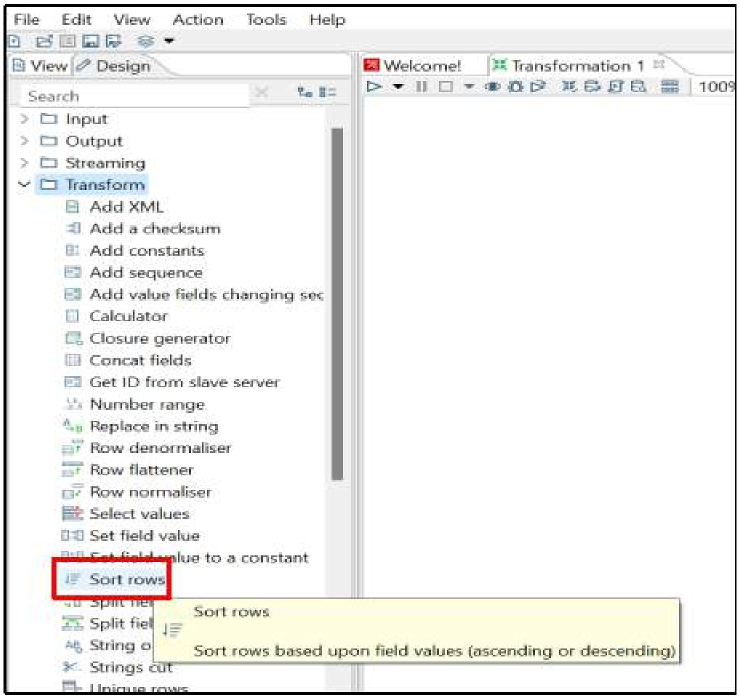
Start at value: 1

Increment by: 2

Maximum value: 20

Step 3: Perform transformation (Sort rows)

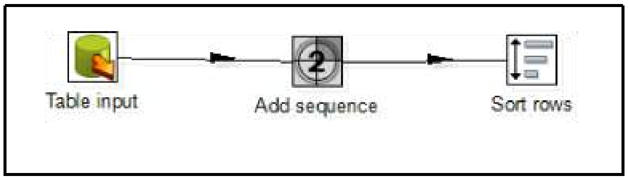
Drag and drop *Sort rows* from the transform folder under the *Design* tab.



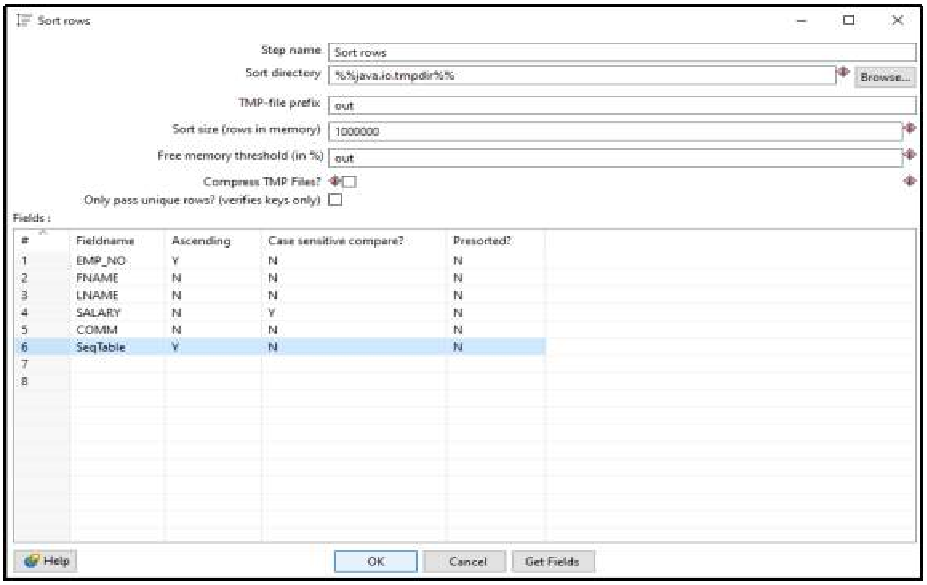
Sort rows

Sort rows based upon field values (ascending or descending)

Hold the mouse pointer on *Add sequence* and select and drag the output connector to the *Sort rows*.



Double click on Sort rows and fill in the details as shown below→Click on OK.


Sort rows

Step name: Sort rows

Sort directory: %%java.io.tmpdir%%

TMP-file prefix: out

Sort size (rows in memory): 1000000

Free memory threshold (in %): out

Compress TMP Files? □

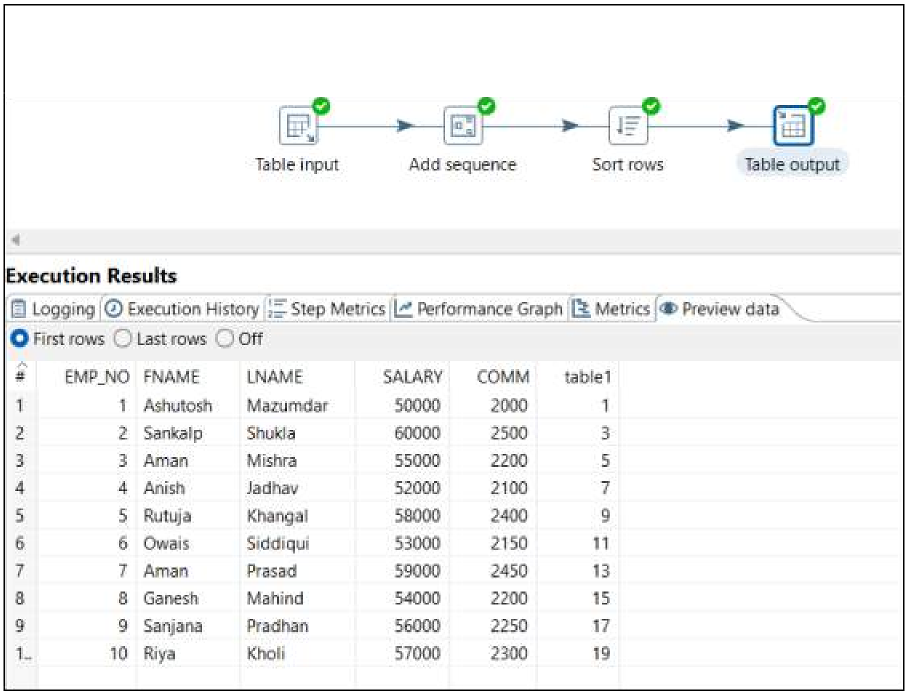
Only pass unique rows? (verifies keys only) □

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Presorted?
1	EMP_NO	Y	N	N
2	FNAME	N	N	N
3	LNAME	N	N	N
4	SALARY	N	Y	N
5	COMM	N	N	N
6	SeqTable	Y	N	N
7				
8				

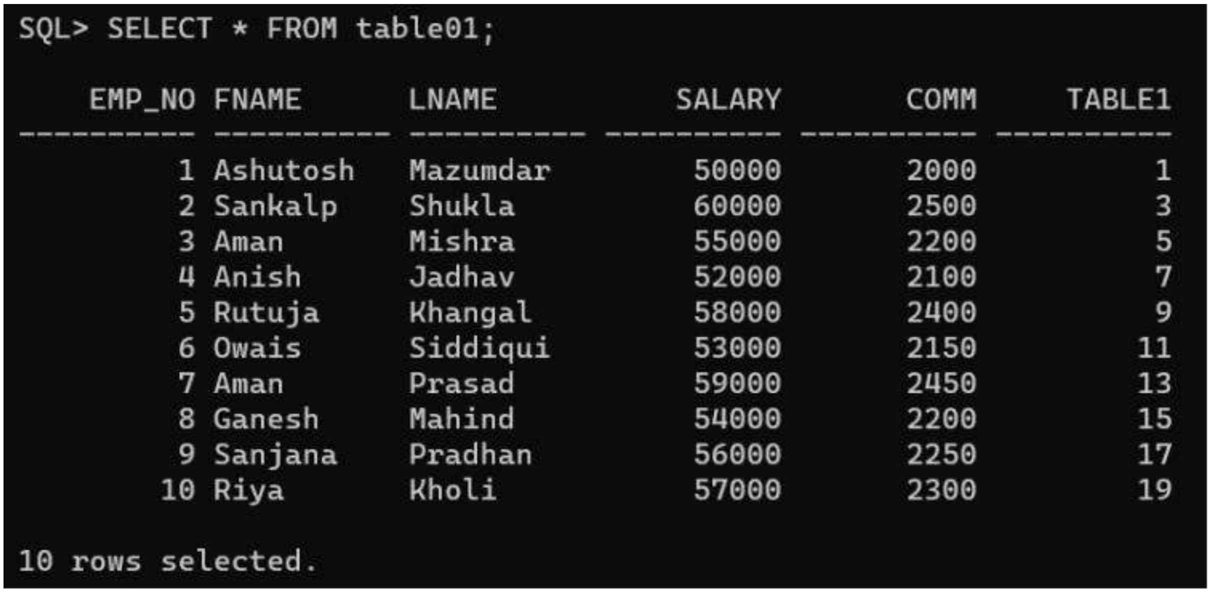
Step 4: Repeat Step 4 from TRANSFORMATION 1.

If the Transformation is successful, you will see green ticks



#	EMP_NO	FNAME	LNAME	SALARY	COMM	table1
1	1	Ashutosh	Mazumdar	50000	2000	1
2	2	Sankalp	Shukla	60000	2500	3
3	3	Aman	Mishra	55000	2200	5
4	4	Anish	Jadhav	52000	2100	7
5	5	Rutuja	Khangal	58000	2400	9
6	6	Owais	Siddiqui	53000	2150	11
7	7	Aman	Prasad	59000	2450	13
8	8	Ganesh	Mahind	54000	2200	15
9	9	Sanjana	Pradhan	56000	2250	17
1..	10	Riya	Kholi	57000	2300	19

Output in sql table remains the same:



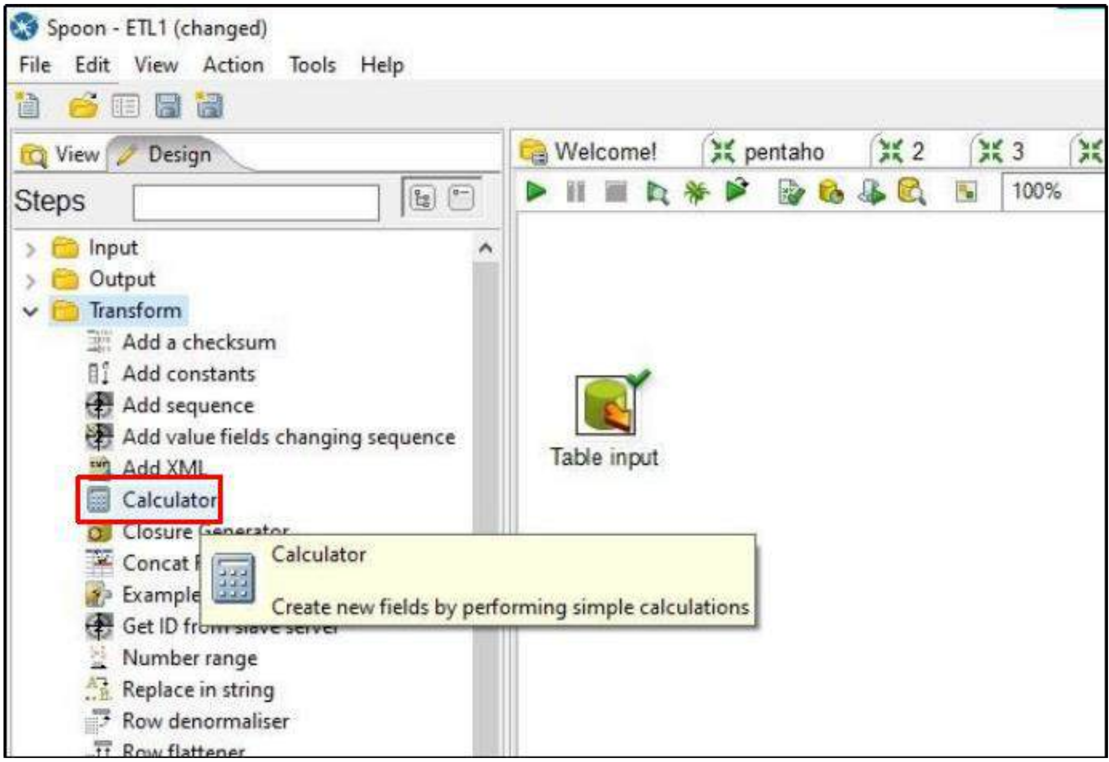
```
SQL> SELECT * FROM table01;

    EMP_NO FNAME      LNAME          SALARY       COMM     TABLE1
---------- ---------- ---------- ---------- ---------- ----------
         1 Ashutosh   Mazumdar        50000       2000          1
         2 Sankalp    Shukla          60000       2500          3
         3 Aman       Mishra          55000       2200          5
         4 Anish      Jadhav          52000       2100          7
         5 Rutuja     Khangal         58000       2400          9
         6 Owais      Siddiqui        53000       2150         11
         7 Aman       Prasad          59000       2450         13
         8 Ganesh     Mahind          54000       2200         15
         9 Sanjana    Pradhan         56000       2250         17
        10 Riya       Kholi           57000       2300         19

10 rows selected.
```

TRAMSFORMATION 3: Calculator

Repeat Steps 1 to 3 from TRANSFORMATION 1.

Step 4: Perform Transformation

Drag and drop *Calculator* from *Transform* folder under *Design* tab.

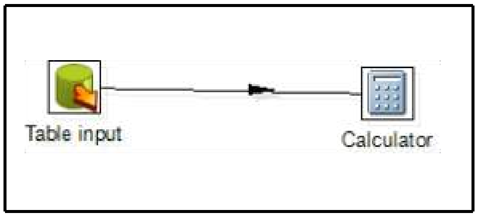


Calculator

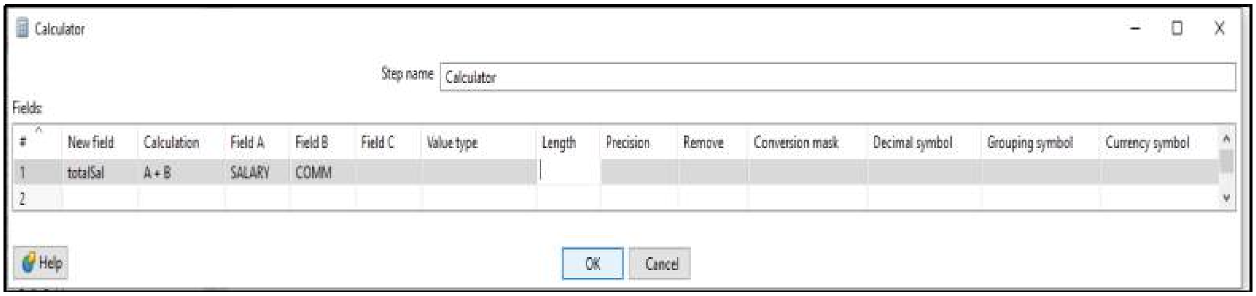
Create new fields by performing simple calculations

Hold the mouse Pointer on Table input and select and drag the output connector to the

Calculator



Double Click on *Calculator* and fill in the details as shown below.



Step name: Calculator

Fields:

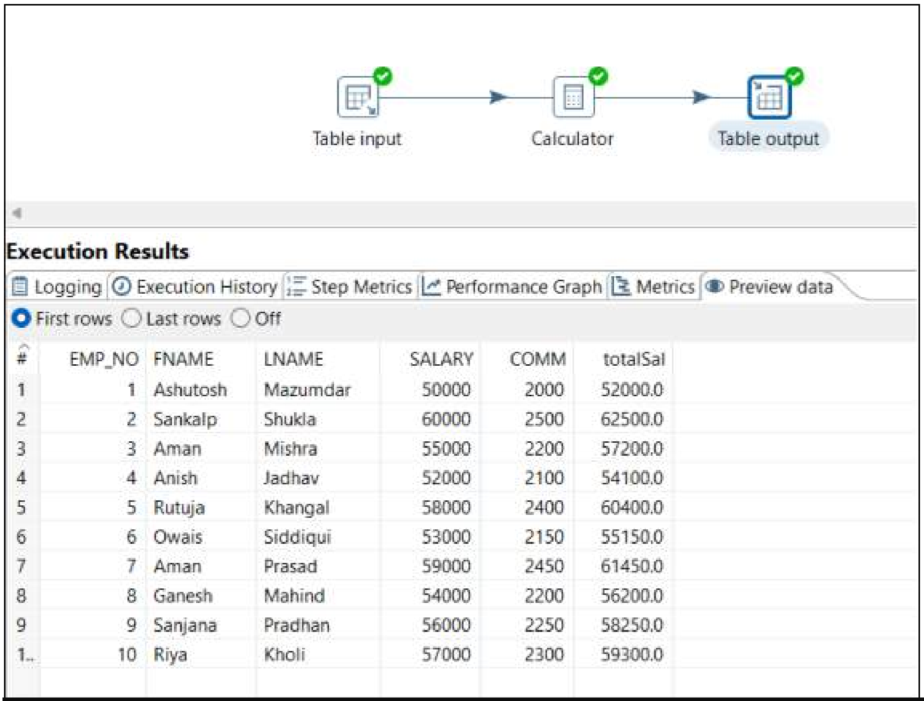
#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove	Conversion mask	Decimal symbol	Grouping symbol	Currency symbol
1	totalSal	A + B	SALARY	COMM									
2													

This will add the values in SALARY column and COMM column as result will be stored in TOTALSAL column.

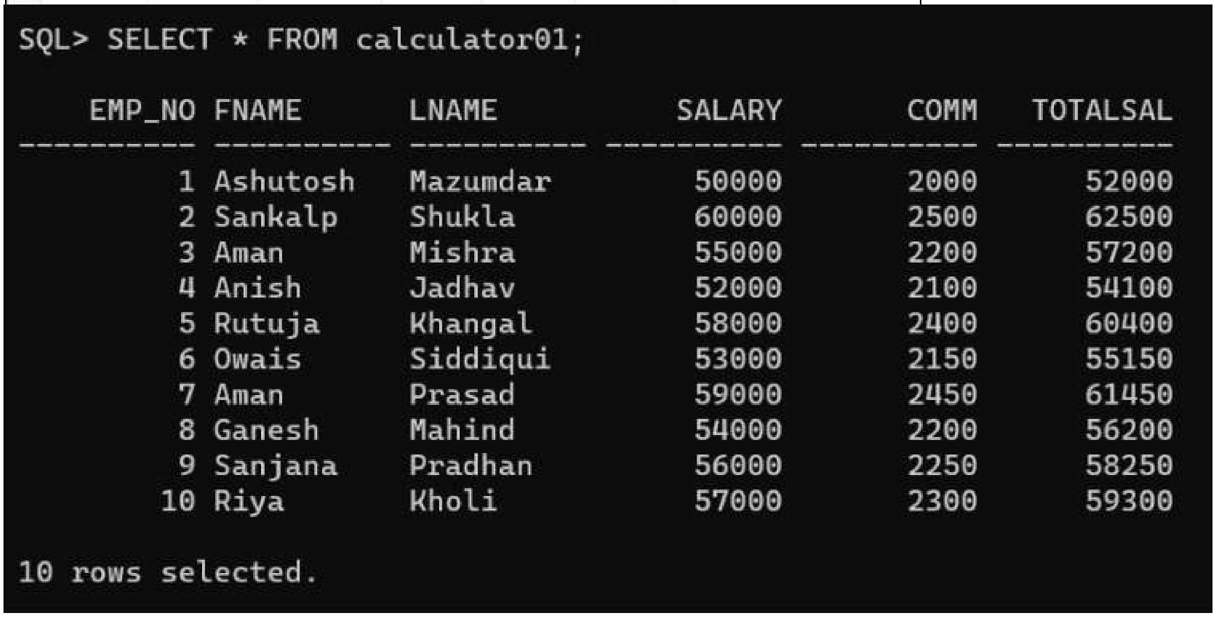
Click on OK.

Step 5: Repeat Step 4 from TRANSFORMATION 1.

If the Transformation is successful, you will see green ticks.



#	EMP_NO	FNAME	LNAME	SALARY	COMM	totalSal
1	1	Ashutosh	Mazumdar	50000	2000	52000.0
2	2	Sankalp	Shukla	60000	2500	62500.0
3	3	Aman	Mishra	55000	2200	57200.0
4	4	Anish	Jadhav	52000	2100	54100.0
5	5	Rutuja	Khangal	58000	2400	60400.0
6	6	Owais	Siddiqui	53000	2150	55150.0
7	7	Aman	Prasad	59000	2450	61450.0
8	8	Ganesh	Mahind	54000	2200	56200.0
9	9	Sanjana	Pradhan	56000	2250	58250.0
1..	10	Riya	Kholi	57000	2300	59300.0



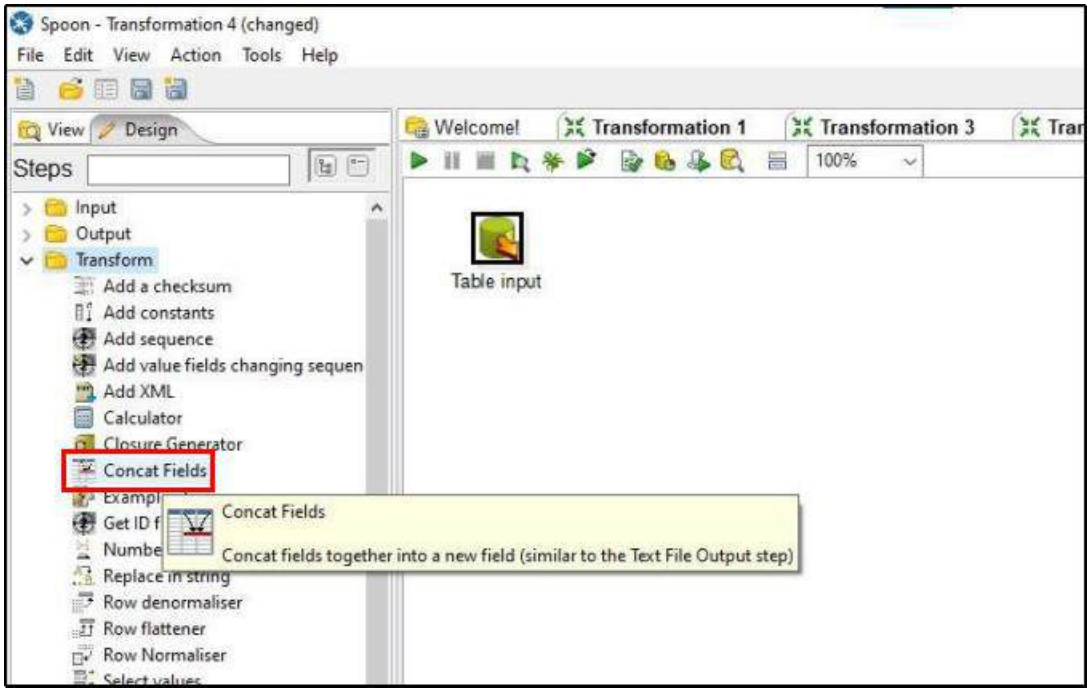
```
SQL> SELECT * FROM calculator01;

    EMP_NO FNAME      LNAME          SALARY       COMM   TOTALSAL
---------- ---------- ---------- ---------- ---------- ----------
         1 Ashutosh   Mazumdar        50000       2000      52000
         2 Sankalp    Shukla          60000       2500      62500
         3 Aman       Mishra          55000       2200      57200
         4 Anish      Jadhav          52000       2100      54100
         5 Rutuja     Khangal         58000       2400      60400
         6 Owais      Siddiqui        53000       2150      55150
         7 Aman       Prasad          59000       2450      61450
         8 Ganesh     Mahind          54000       2200      56200
         9 Sanjana    Pradhan         56000       2250      58250
        10 Riya       Kholi           57000       2300      59300

10 rows selected.
```

TRANSFORMATION 4: Concat Fields

Step 1: Repeat Steps 2 and 3 from TRANSFORMATION 1.

Step 2: Perform Transformation.

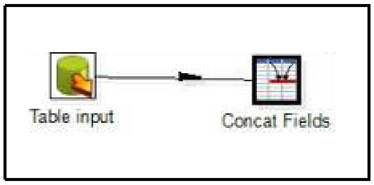
Drag and drop *Concat Fields* from *Transform* folder under *Design* tab.



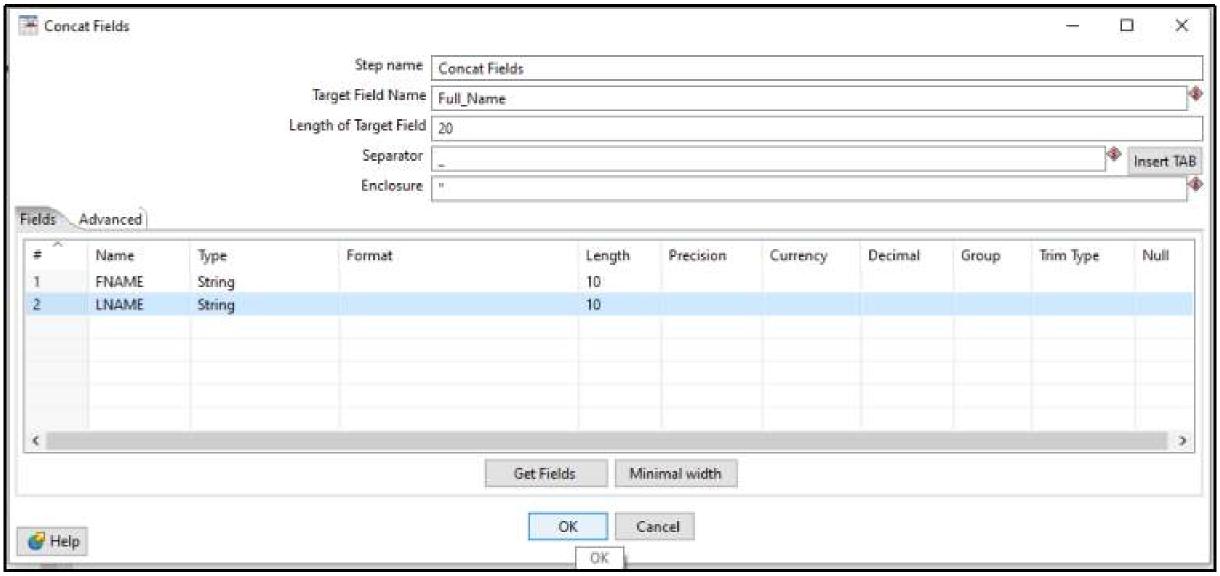
Concat Fields

Concat fields together into a new field (similar to the Text File Output step)

Hold the mouse Pointer on Table input and select and drag the output connector to the *Concat Fields*.



Double Click on *Concat Fields* and fill in the details as shown below→Click on OK.

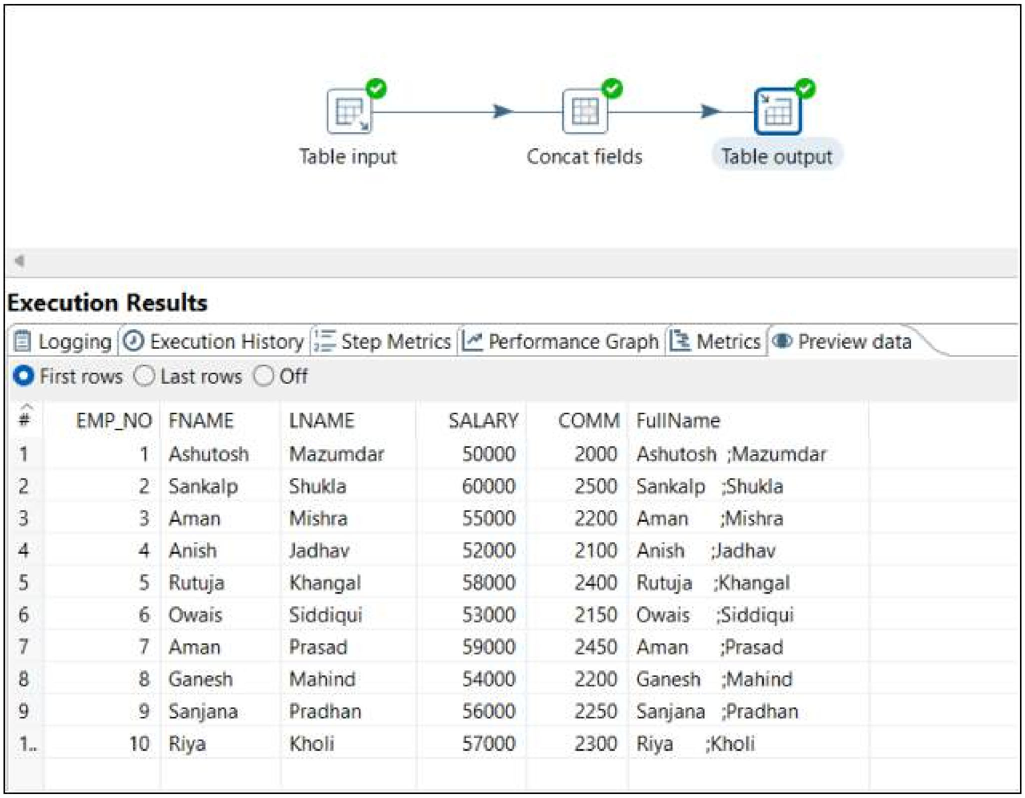


Step name: Concat Fields
Target Field Name: Full_Name
Length of Target Field: 20
Separator: _
Enclosure: "

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1	FNAME	String		10						
2	LNAME	String		10						

Step 3: Repeat Step 4 from TRANSFORMATION 1.

If the Transformation is successful, you will see green ticks.



#	EMP_NO	FNAME	LNAME	SALARY	COMM	FullName
1	1	Ashutosh	Mazumdar	50000	2000	Ashutosh ;Mazumdar
2	2	Sankalp	Shukla	60000	2500	Sankalp ;Shukla
3	3	Aman	Mishra	55000	2200	Aman ;Mishra
4	4	Anish	Jadhav	52000	2100	Anish ;Jadhav
5	5	Rutuja	Khangal	58000	2400	Rutuja ;Khangal
6	6	Owais	Siddiqui	53000	2150	Owais ;Siddiqui
7	7	Aman	Prasad	59000	2450	Aman ;Prasad
8	8	Ganesh	Mahind	54000	2200	Ganesh ;Mahind
9	9	Sanjana	Pradhan	56000	2250	Sanjana ;Pradhan
1..	10	Riya	Kholi	57000	2300	Riya ;Kholi

TRANSFORMATION 5: Number Range

SQL> CREATE TABLE percent05 (

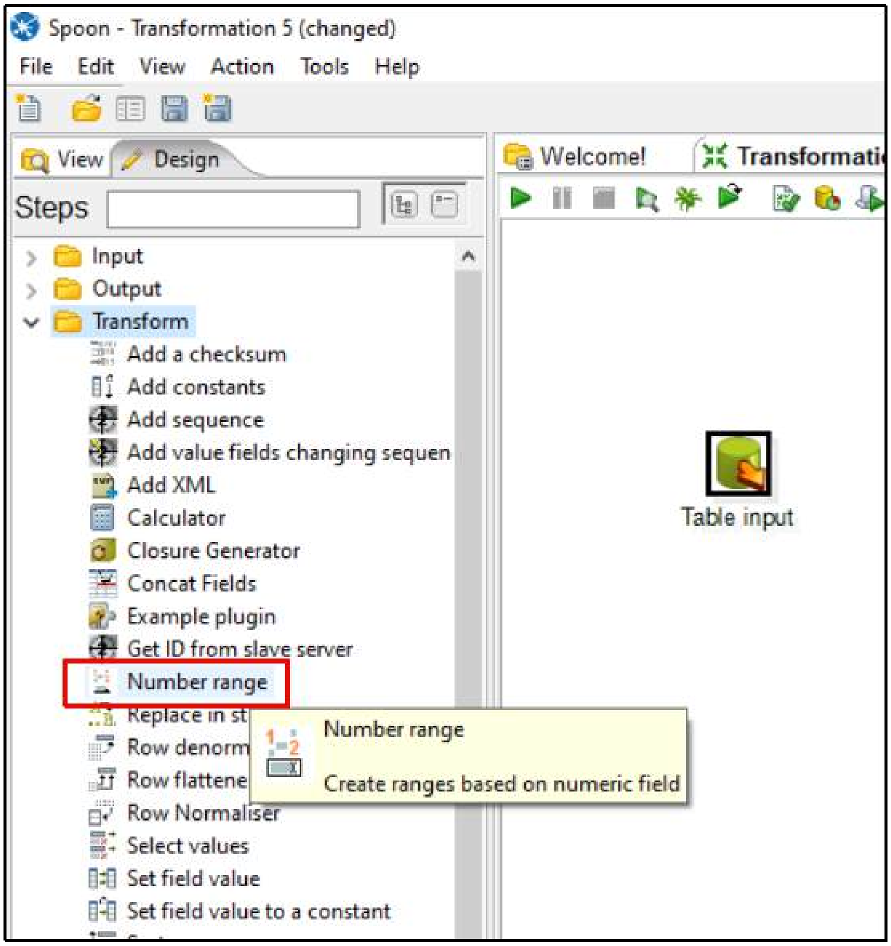
```
  ROLL_NO NUMBER, NAME

  VARCHAR2(50),

  PERCENTAGE Integer

  );

Table created.

SQL> INSERT ALL

INTO percent05 VALUES (1, 'Ashutosh', 85.5)

INTO percent05 VALUES (2, 'Rutuja', 92.3)

INTO percent05 VALUES (3, 'Sankalp', 78.9)

INTO percent05 VALUES (4, 'Ganesh', 94.7)

INTO percent05 VALUES (5, 'Aman', 87.2)

INTO percent05 VALUES (6, 'Riya', 91.0)

INTO percent05 VALUES (7, 'Pooja', 79.8)

SELECT * FROM dual;

6 rows created

SQL> select *from percent05;

   ROLL_NO NAME    PERCENTAGE




         1 Ashutosh                      86

         2 Rutuja                      92
```

3	Sankalp	79
4	Ganesh	95
5	Aman	87
6	Riya	91
7	Pooja	80

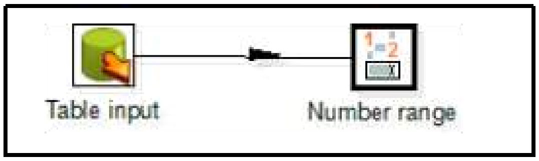
Step 1: Repeat Steps 2 and 3 from TRANSFORMATION 1.

Step 2: Perform Transformation.

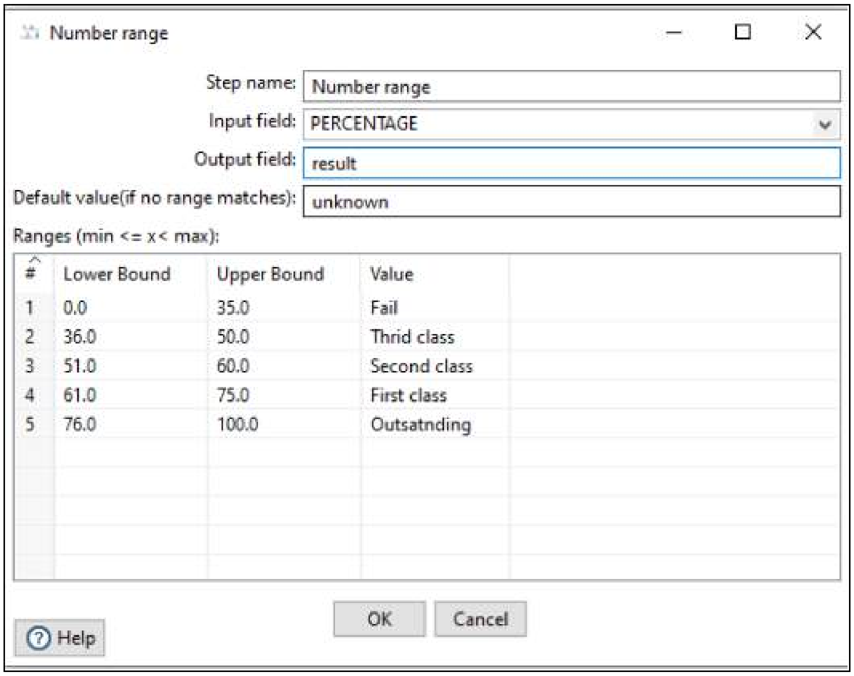
Drag and drop *Number Range* from *Transform* folder under *Design* tab.



Hold the mouse Pointer on Table input and select and drag the output connector to the Number range.



Double Click on *Number range* and fill in the details as shown below→Click on OK.



Step name: Number range

Input field: PERCENTAGE

Output field: result

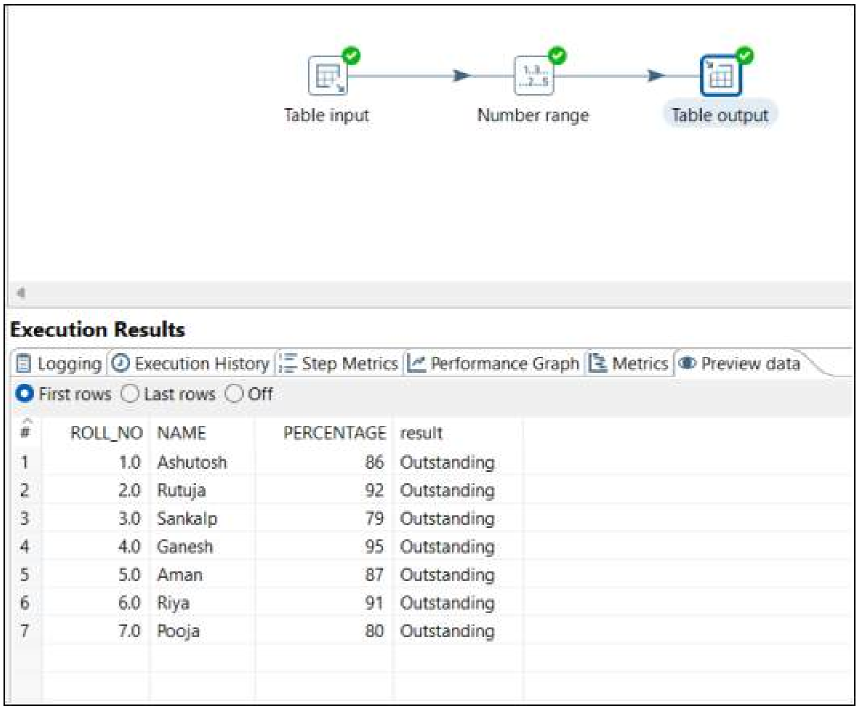
Default value(if no range matches): unknown

Ranges (min <= x< max):

#	Lower Bound	Upper Bound	Value
1	0.0	35.0	Fail
2	36.0	50.0	Thrid class
3	51.0	60.0	Second class
4	61.0	75.0	First class
5	76.0	100.0	Outsatnding

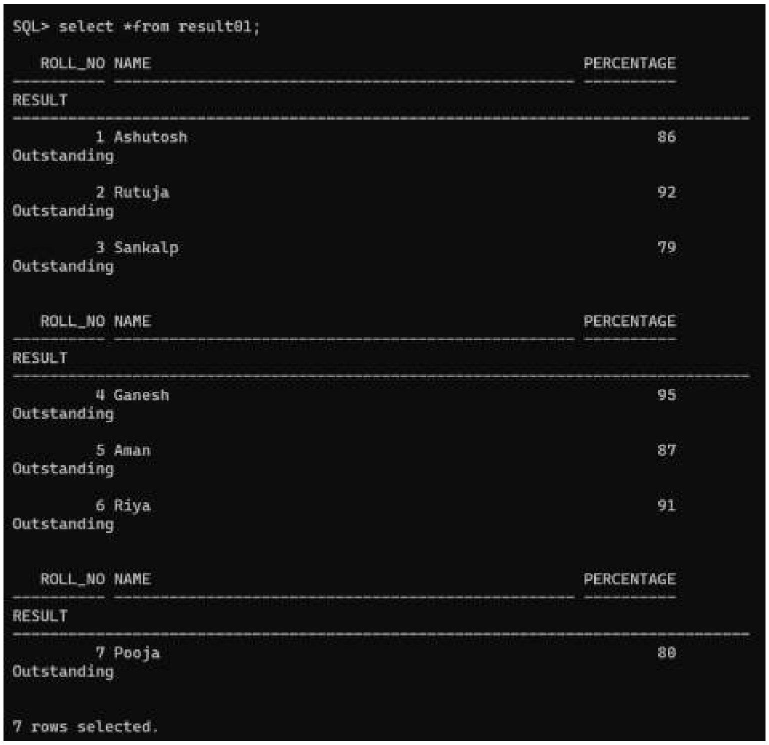
Step 3: Repeat Step 4 from TRANSFORMATION 1.

If the Transformation is successful, you will see green ticks.



#	ROLL_NO	NAME	PERCENTAGE	result
1	1.0	Ashutosh	86	Outstanding
2	2.0	Rutuja	92	Outstanding
3	3.0	Sankalp	79	Outstanding
4	4.0	Ganesh	95	Outstanding
5	5.0	Aman	87	Outstanding
6	6.0	Riya	91	Outstanding
7	7.0	Pooja	80	Outstanding

Step 4: Run SQL query.



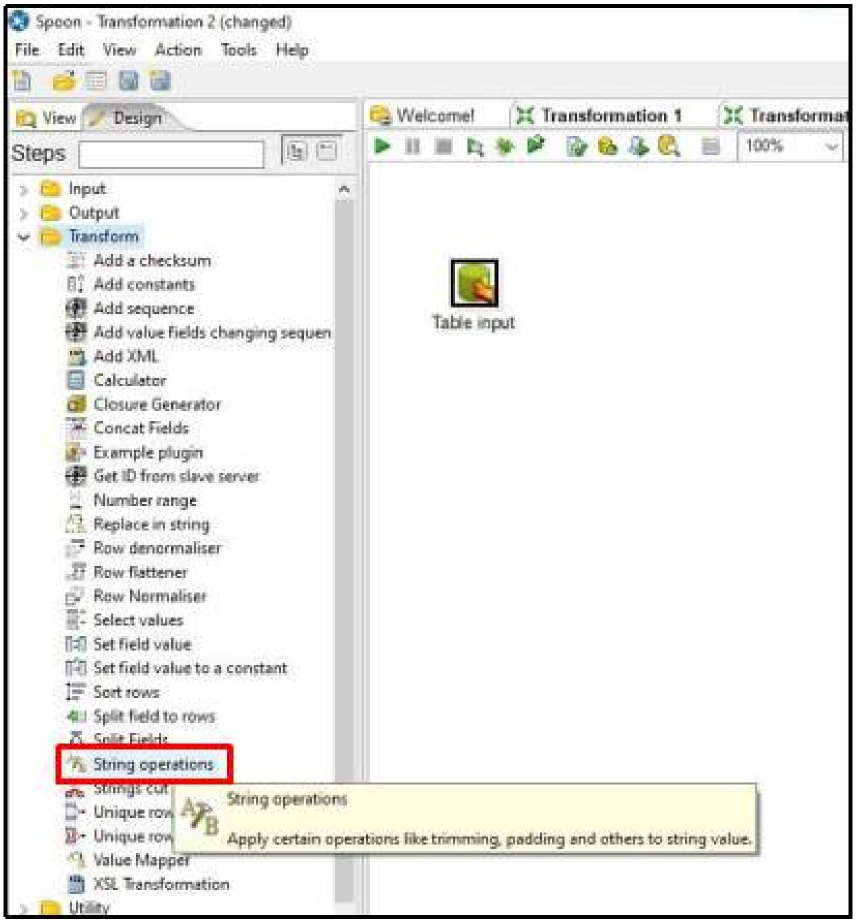
```
SQL> select *from result01;

   ROLL_NO NAME                                                PERCENTAGE
---------- -------------------------------------------------- ----------
RESULT
--------------------------------------------------------------------------------
         1 Ashutosh                                                    86
Outstanding

         2 Rutuja                                                      92
Outstanding

         3 Sankalp                                                     79
Outstanding


   ROLL_NO NAME                                                PERCENTAGE
---------- -------------------------------------------------- ----------
RESULT
--------------------------------------------------------------------------------
         4 Ganesh                                                      95
Outstanding

         5 Aman                                                        87
Outstanding

         6 Riya                                                        91
Outstanding


   ROLL_NO NAME                                                PERCENTAGE
---------- -------------------------------------------------- ----------
RESULT
--------------------------------------------------------------------------------
         7 Pooja                                                       80
Outstanding


7 rows selected.
```

TRANSFORMATION 6: String Operations

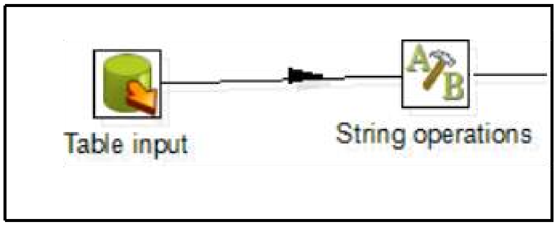
Step 1: Repeat Steps 2 and 3 from TRANSFORMATION 1.

Step 2: Perform Transformation.

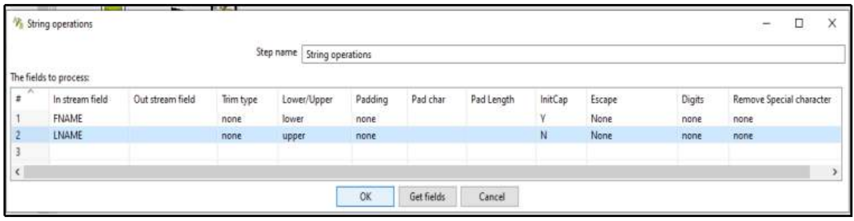
Drag and drop *Number Range* from *Transform* folder under *Design* tab.



Hold the mouse Pointer on Table input and select and drag the output connector to the String operations.



Double Click on *String operations* and fill in the details as shown below→Click on OK.



Step name: String operations

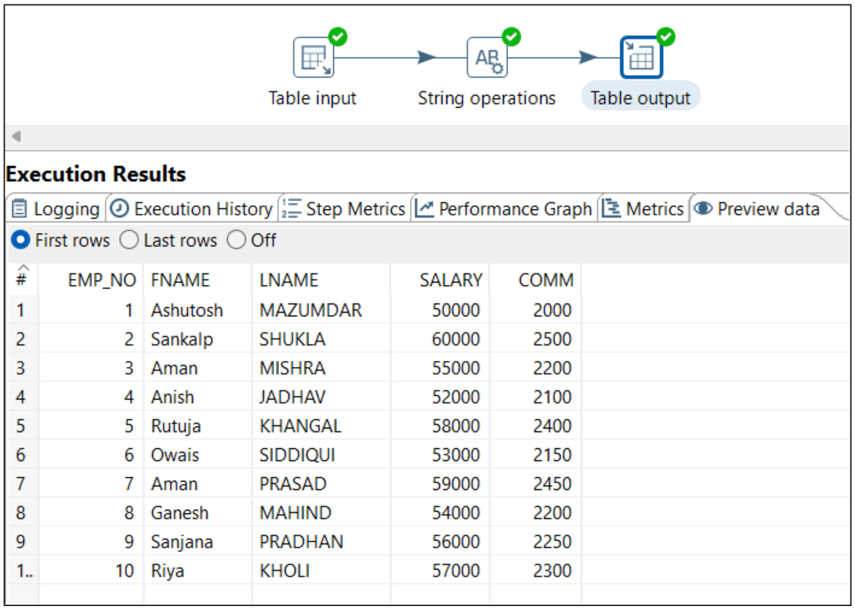
The fields to process:

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	FNAME		none	lower	none			Y	None	none	none
2	LNAME		none	upper	none			N	None	none	none
3											

Step 3:

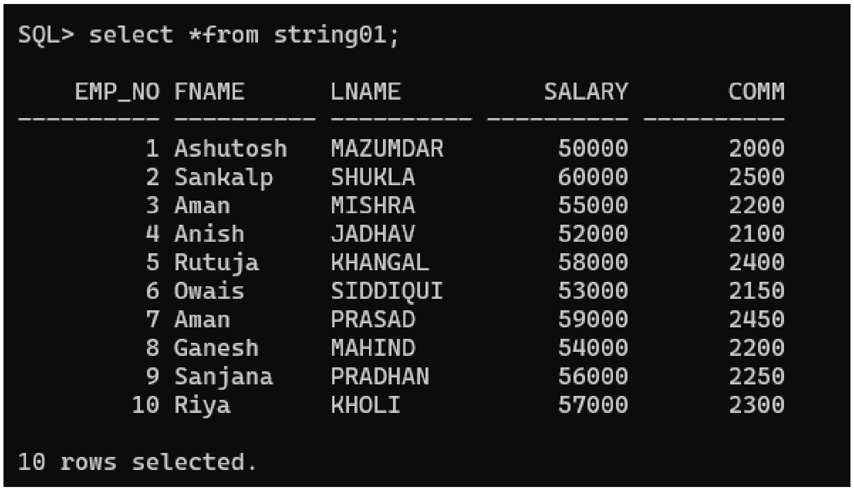
Repeat Step 4 from TRANSFORMATION 1.

If the Transformation is successful, you will see green ticks.



#	EMP_NO	FNAME	LNAME	SALARY	COMM
1	1	Ashutosh	MAZUMDAR	50000	2000
2	2	Sankalp	SHUKLA	60000	2500
3	3	Aman	MISHRA	55000	2200
4	4	Anish	JADHAV	52000	2100
5	5	Rutuja	KHANGAL	58000	2400
6	6	Owais	SIDDIQUI	53000	2150
7	7	Aman	PRASAD	59000	2450
8	8	Ganesh	MAHIND	54000	2200
9	9	Sanjana	PRADHAN	56000	2250
1..	10	Riya	KHOLI	57000	2300

Step 4: Run SQL query.



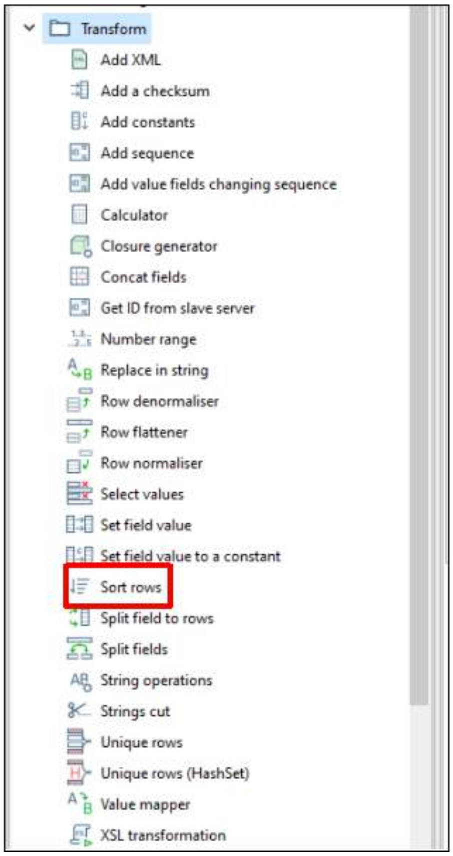
```
SQL> select *from string01;

    EMP_NO FNAME      LNAME          SALARY       COMM
---------- ---------- ---------- ---------- ----------
         1 Ashutosh   MAZUMDAR        50000       2000
         2 Sankalp    SHUKLA          60000       2500
         3 Aman       MISHRA          55000       2200
         4 Anish      JADHAV          52000       2100
         5 Rutuja     KHANGAL         58000       2400
         6 Owais      SIDDIQUI        53000       2150
         7 Aman       PRASAD          59000       2450
         8 Ganesh     MAHIND          54000       2200
         9 Sanjana    PRADHAN         56000       2250
        10 Riya       KHOLI           57000       2300

10 rows selected.
```

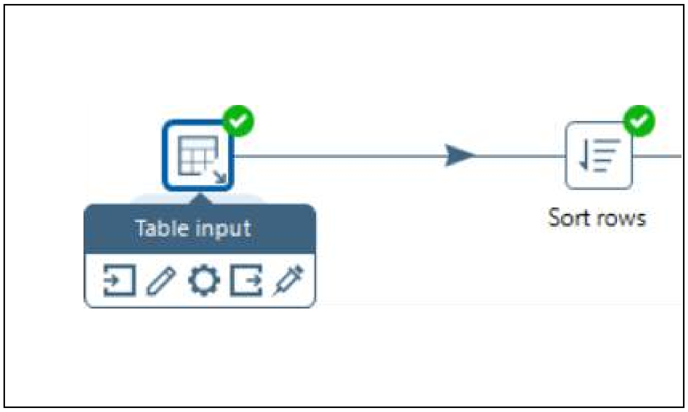
TRANSFORMATION 7: Sorting Data

Step 1: Repeat Steps 2 and 3 from TRANSFORMATION 1.

Step 2: Perform Transformation.

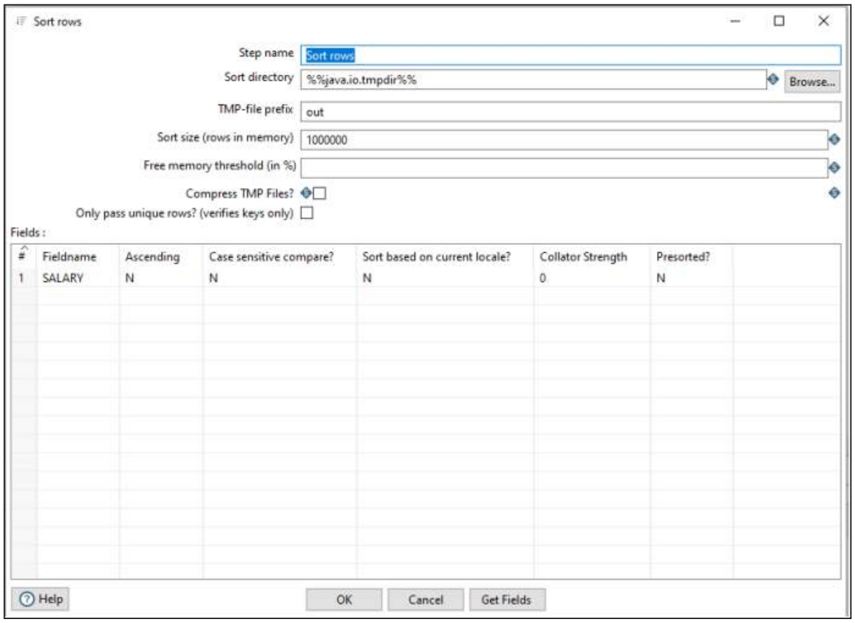
Drag and drop *Sort rows* from *Transform* folder under *Design* tab.





Double click on Sort row and fill the

following fields And than click on Ok

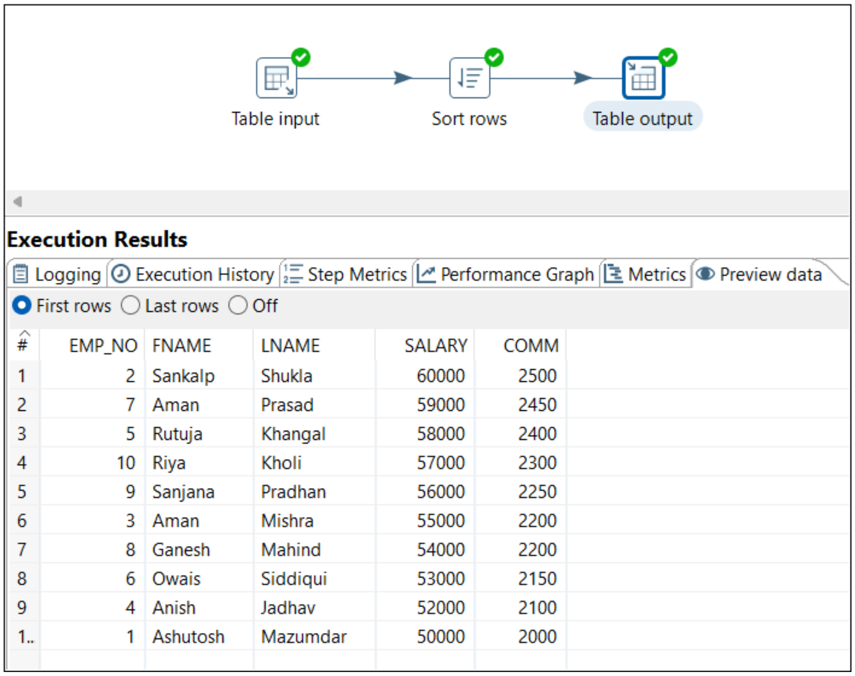


Step name: Sort rows
Sort directory: %%java.io.tmpdir%%
TMP-file prefix: out
Sort size (rows in memory): 1000000
Free memory threshold (in %):
Compress TMP Files? □
Only pass unique rows? (verifies keys only) □

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	SALARY	N	N	N	0	N

Perform Output same as we done in early tranformations.



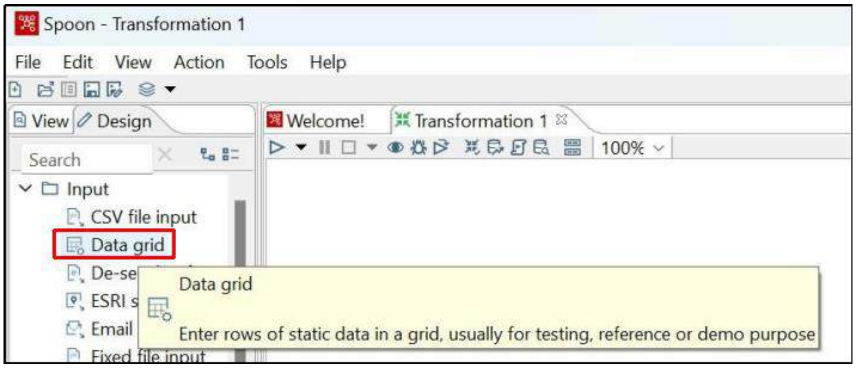
Execution Results

● First rows ○ Last rows ○ Off

#	EMP_NO	FNAME	LNAME	SALARY	COMM
1	2	Sankalp	Shukla	60000	2500
2	7	Aman	Prasad	59000	2450
3	5	Rutuja	Khangal	58000	2400
4	10	Riya	Kholi	57000	2300
5	9	Sanjana	Pradhan	56000	2250
6	3	Aman	Mishra	55000	2200
7	8	Ganesh	Mahind	54000	2200
8	6	Owais	Siddiqui	53000	2150
9	4	Anish	Jadhav	52000	2100
1..	1	Ashutosh	Mazumdar	50000	2000

RANSFORMATION 8: Merge Join

Step 1: Drag and drop 2 *Data Grid* from Input folder under *Design* tab.

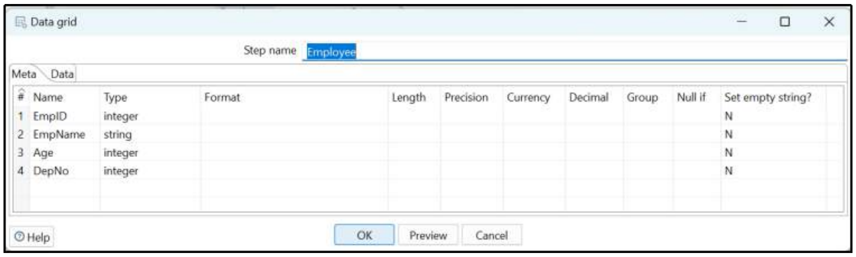


Data grid

Enter rows of static data in a grid, usually for testing, reference or demo purpose

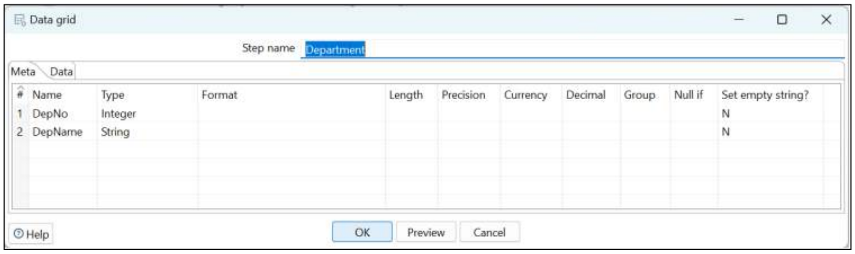
Rename them as Employee and Department.

Step 2: Double click on them and insert records into respective grids→ Click on OK.



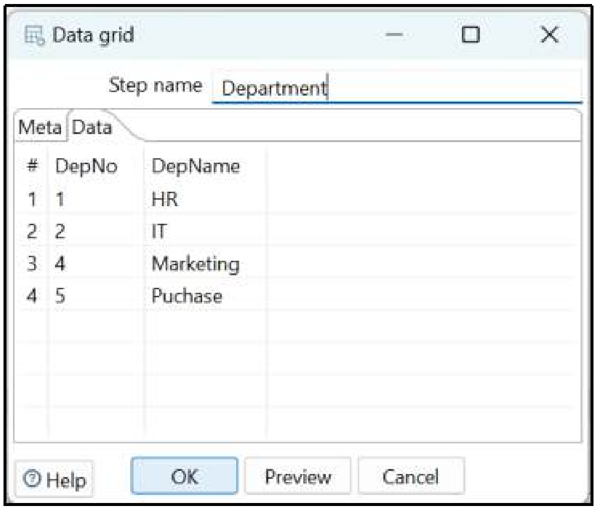
Step name Employee

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Null if	Set empty string?
1	EmpID	integer								N
2	EmpName	string								N
3	Age	integer								N
4	DepNo	integer								N



Step name Department

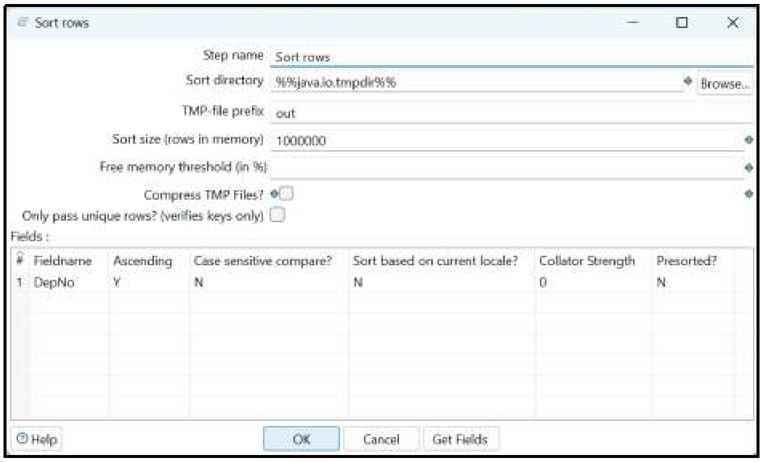
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Null if	Set empty string?
1	DepNo	Integer								N
2	DepName	String								N



Step name Department

#	DepNo	DepName
1	1	HR
2	2	IT
3	4	Marketing
4	5	Puchase

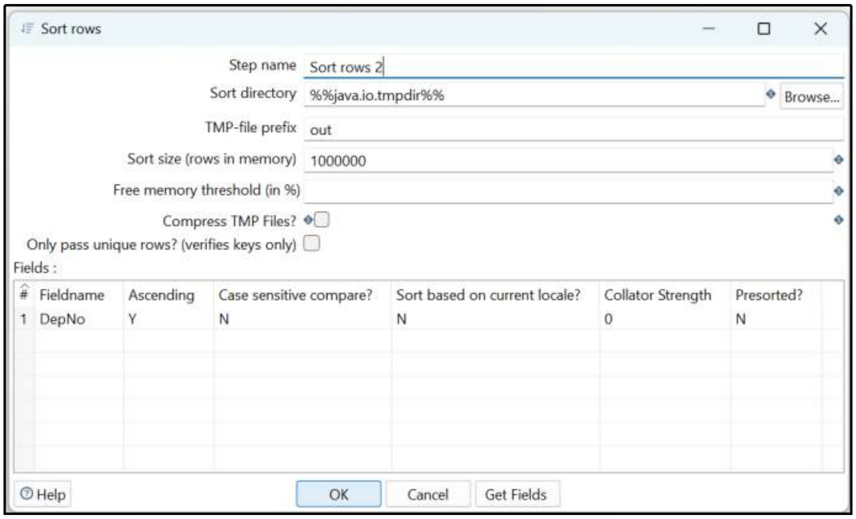
Step 2: Perform Sort rows transformation for both data grids respectively. Click on OK.



Step name: Sort rows
Sort directory: %%java.io.tmpdir%% Browse...
TMP-file prefix: out
Sort size (rows in memory): 1000000
Free memory threshold (in %):
Compress TMP Files? □
Only pass unique rows? (verifies keys only) □

Fields :

#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	DepNo	Y	N	N	0	N



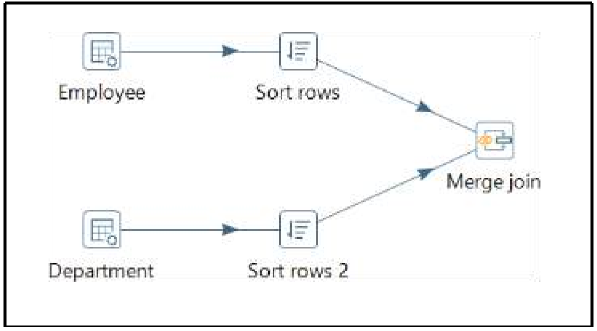
Step name: Sort rows 2
Sort directory: %%java.io.tmpdir%% Browse...
TMP-file prefix: out
Sort size (rows in memory): 1000000
Free memory threshold (in %):
Compress TMP Files? □
Only pass unique rows? (verifies keys only) □

Fields :

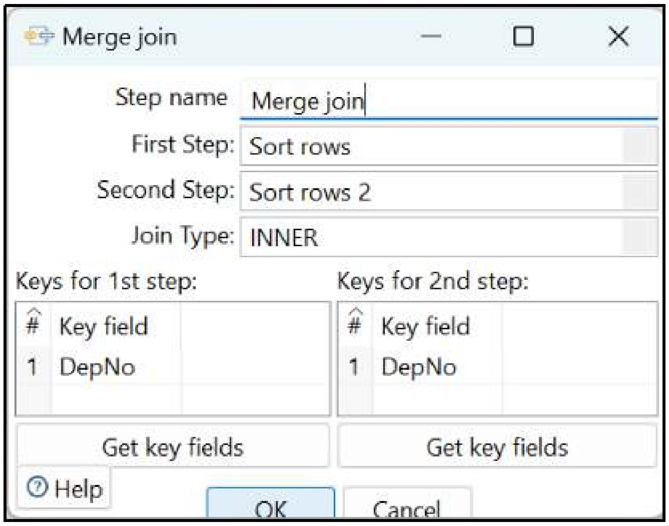
#	Fieldname	Ascending	Case sensitive compare?	Sort based on current locale?	Collator Strength	Presorted?
1	DepNo	Y	N	N	0	N

Step 3: Drag and Drop Merge join from joins folder under Design tab.

Hold the mouse Pointer on both the sort rows and select and drag the output connector to the Merge join as shown below.



Step 4: Double click on Merge join and fill in the details as shown below to perform INNER join→Click on OK.



Step name: Merge join

First Step: Sort rows

Second Step: Sort rows 2

Join Type: INNER

Keys for 1st step:

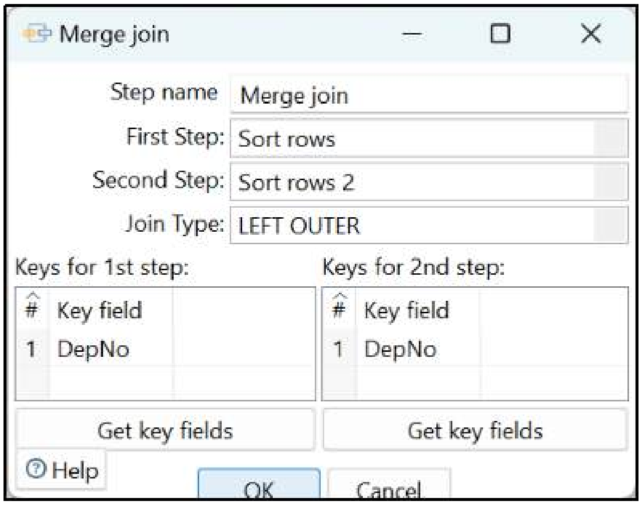
#	Key field
1	DepNo

Keys for 2nd step:

#	Key field
1	DepNo

Debug the transformation and perform Quick launch

Step 5: Double click on Merge join and fill in the details as shown below to perform LEFT OUTER joinàClick on OK.



Step name: Merge join

First Step: Sort rows

Second Step: Sort rows 2

Join Type: LEFT OUTER

Keys for 1st step:

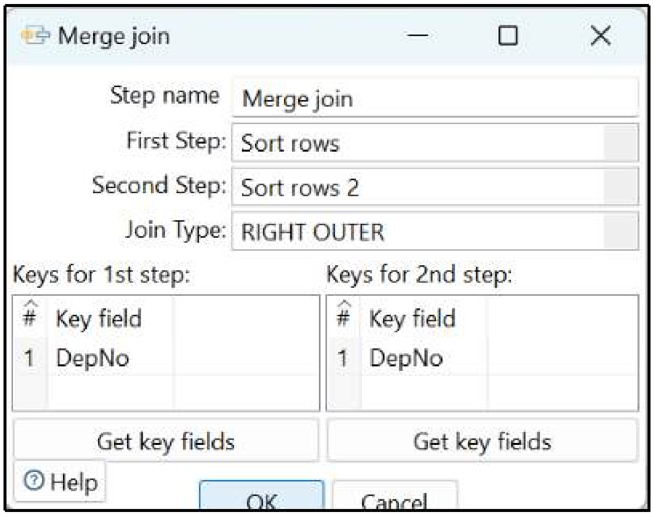
#	Key field
1	DepNo

Keys for 2nd step:

#	Key field
1	DepNo

Debug the transformation and perform Quick launch

Step 6: Double click on Merge join and fill in the details as shown below to perform RIGHT OUTER join→Click on OK.



Step name: Merge join

First Step: Sort rows

Second Step: Sort rows 2

Join Type: RIGHT OUTER

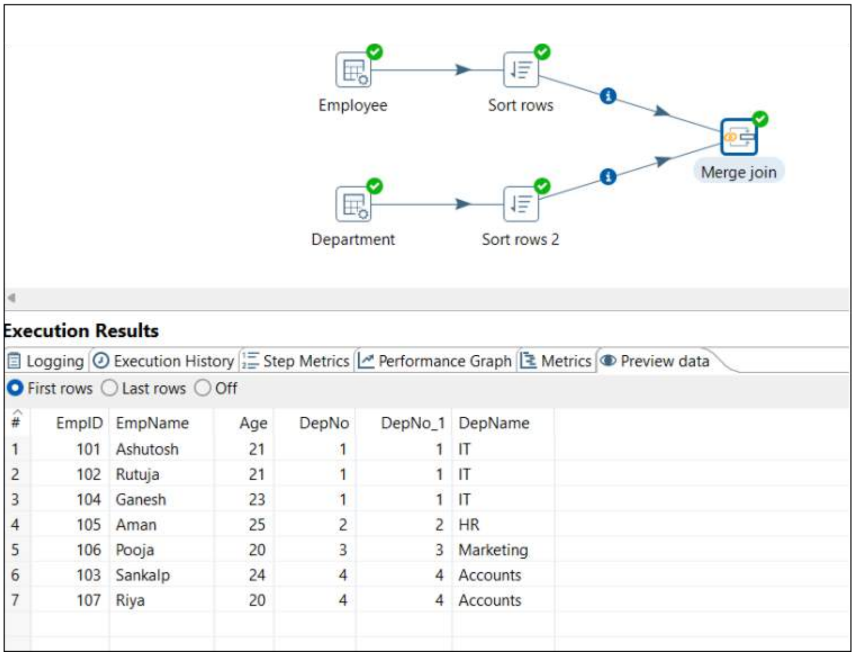
Keys for 1st step:

#	Key field
1	DepNo

Keys for 2nd step:

#	Key field
1	DepNo

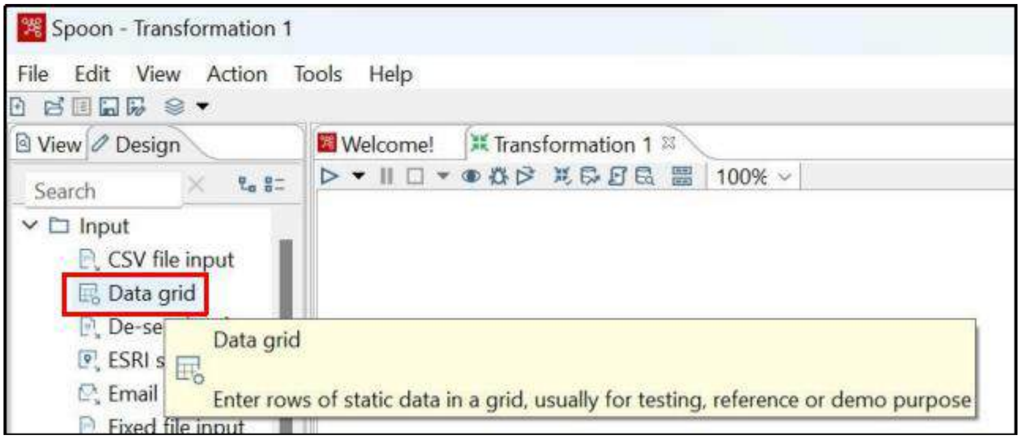
Debug the transformation and perform Quick launch



#	EmpID	EmpName	Age	DepNo	DepNo_1	DepName
1	101	Ashutosh	21	1	1	IT
2	102	Rutuja	21	1	1	IT
3	104	Ganesh	23	1	1	IT
4	105	Aman	25	2	2	HR
5	106	Pooja	20	3	3	Marketing
6	103	Sankalp	24	4	4	Accounts
7	107	Riya	20	4	4	Accounts

TRANSFORMATION 9: Data validations

Step 1: Drag and drop *Data Grid* from Input folder under *Design* tab.

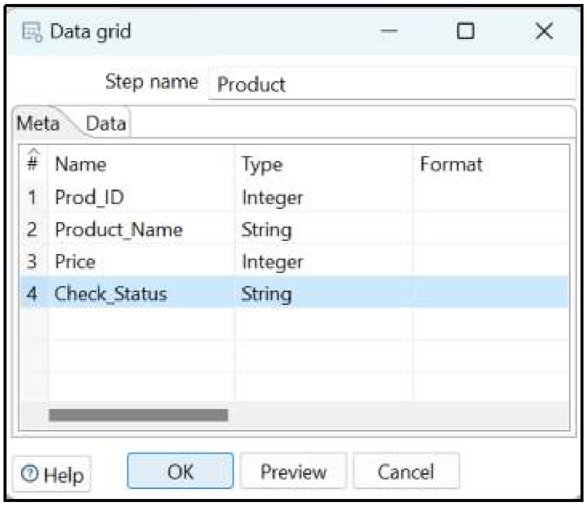


Data grid

Enter rows of static data in a grid, usually for testing, reference or demo purpose

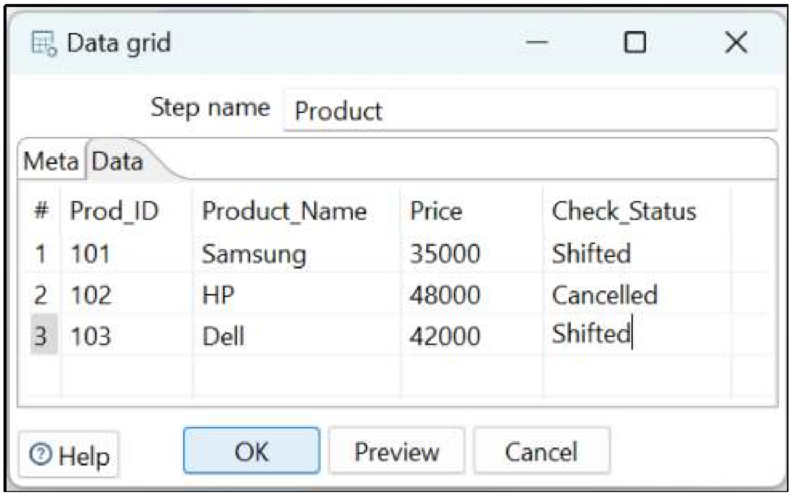
Rename it as Product.

Step 2: Double click on Product data grid and insert records as shown below→ Click on OK.



Step name: Product

#	Name	Type	Format
1	Prod_ID	Integer	
2	Product_Name	String	
3	Price	Integer	
4	Check_Status	String	

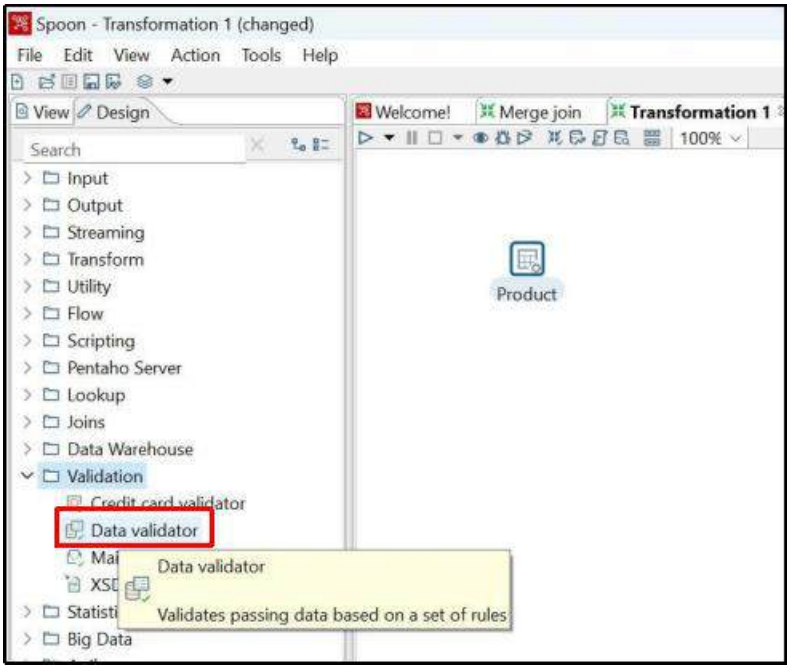


Data grid

Step name: Product

#	Prod_ID	Product_Name	Price	Check_Status
1	101	Samsung	35000	Shifted
2	102	HP	48000	Cancelled
3	103	Dell	42000	Shifted

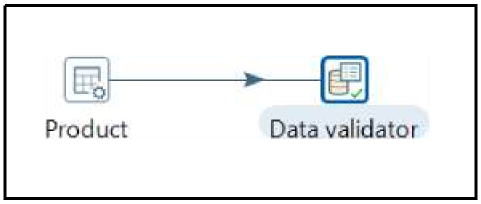
Step 3: Drag and drop Data validator from Validation folder under Design tab



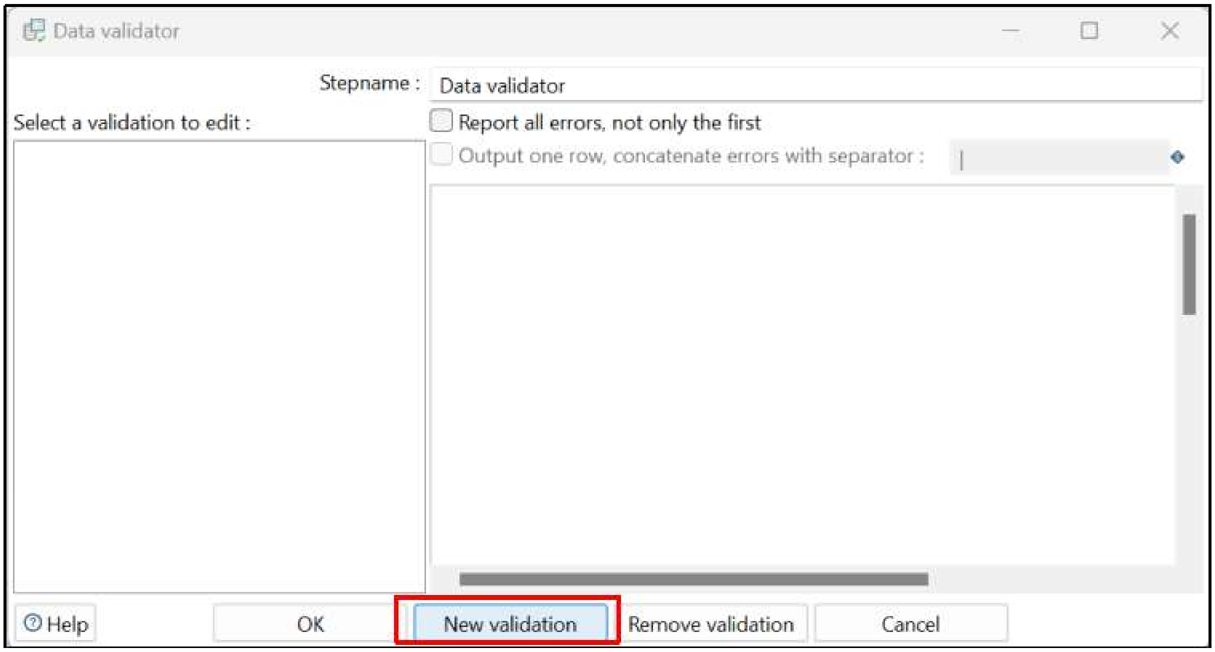
Data validator

Validates passing data based on a set of rules

Hold the mouse pointer on Product data grid and select and drag the output connector to the Data validator.



Double click on Data validator→New Validattion



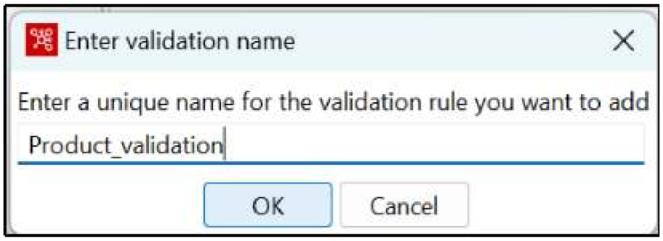
Stepname : Data validator

Select a validation to edit :

□ Report all errors, not only the first

□ Output one row, concatenate errors with separator : |

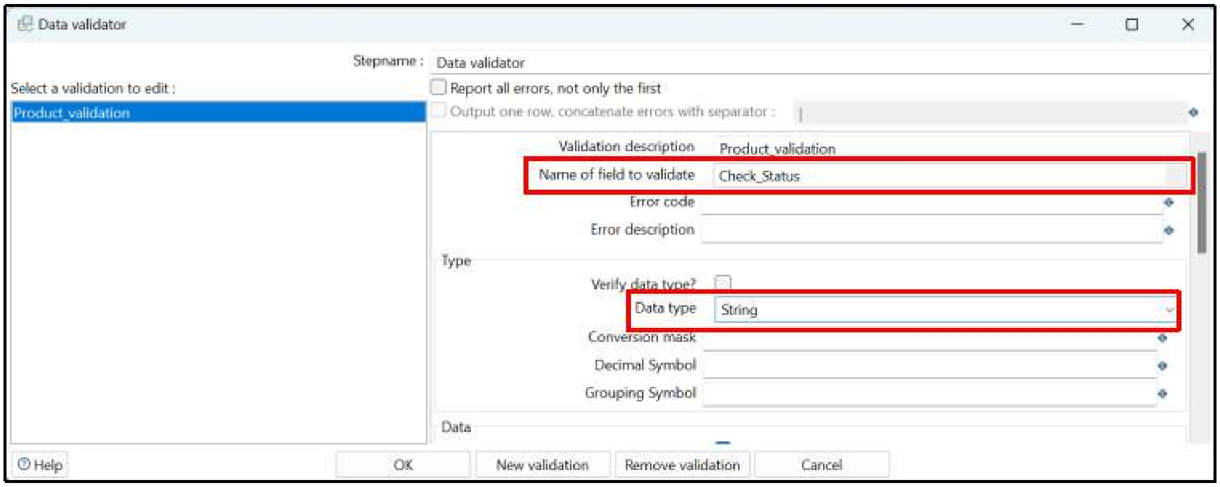
Give Valiadtion Name and click OK.



Enter a unique name for the validation rule you want to add

Product_validation

Select the validation to edit and fill in the details as shown below.



Data validator

Stepname : Data validator

Select a validation to edit : Product_validation

□ Report all errors, not only the first

□ Output one row, concatenate errors with separator :

Validation description: Product_validation

Name of field to validate: Check_Status

Error code:

Error description:

Type

Verify data type? □

Data type: String

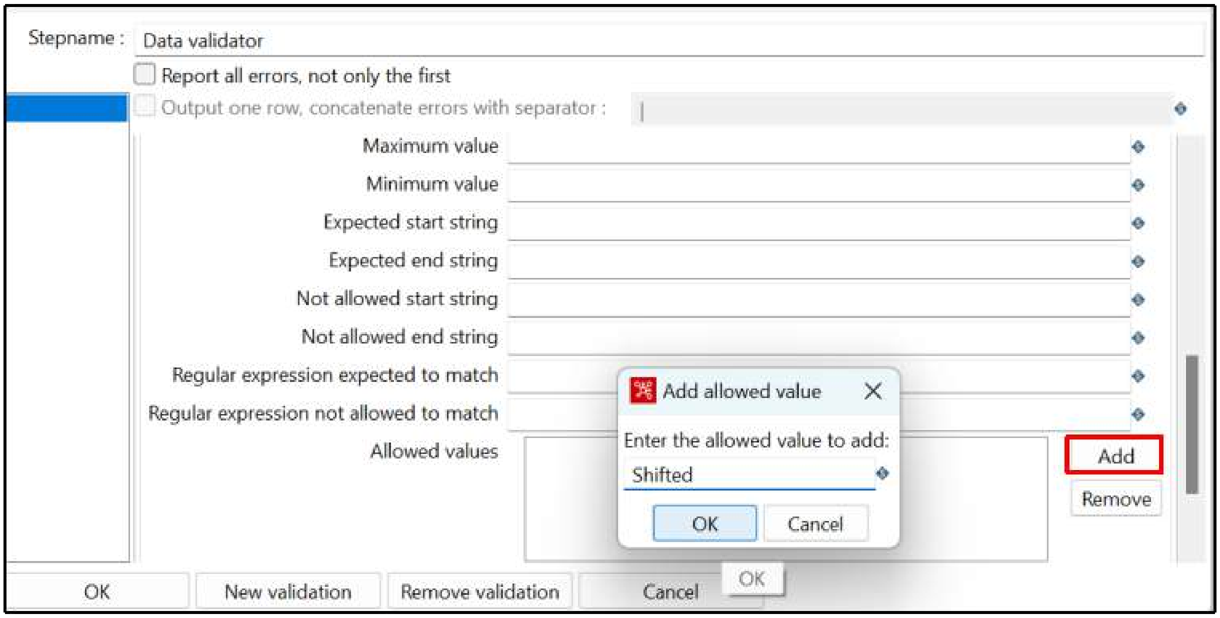
Conversion mask:

Decimal Symbol:

Grouping Symbol:

Data

Click on add to set validation, set it 65 to Shifted and press Enter and click on ok.



Stepname : Data validator

□ Report all errors, not only the first

□ Output one row, concatenate errors with separator :

Maximum value:

Minimum value:

Expected start string:

Expected end string:

Not allowed start string:

Not allowed end string:

Regular expression expected to match:

Regular expression not allowed to match:

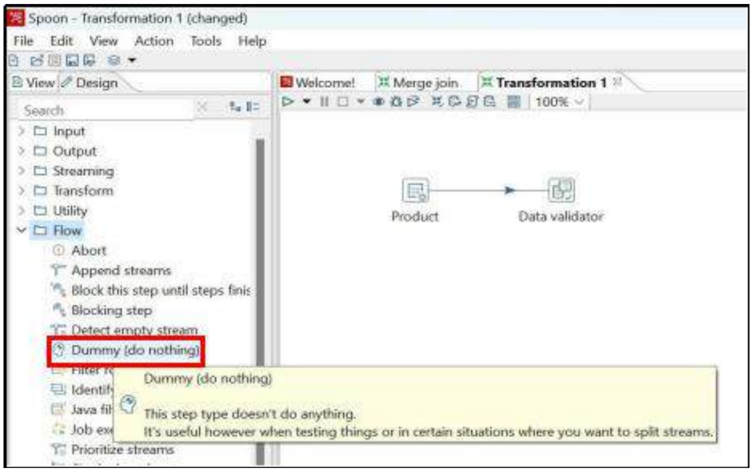
Allowed values:

Add allowed value

Enter the allowed value to add: Shifted

Click on OK.

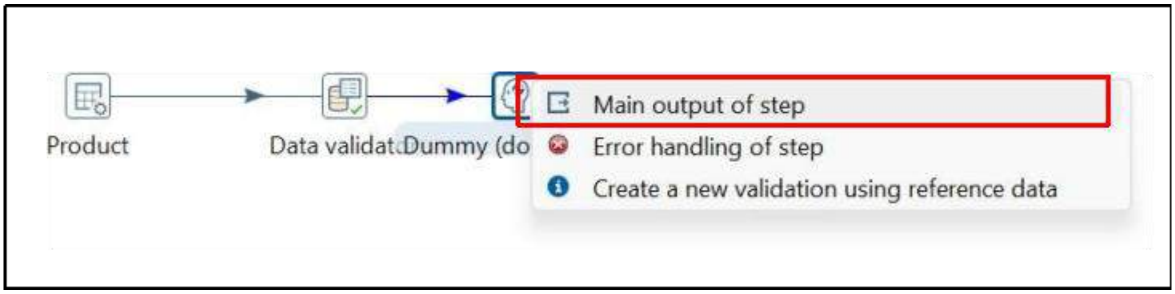
Step 4: Drag and drop Dummy from Flow folder under the Design tab.



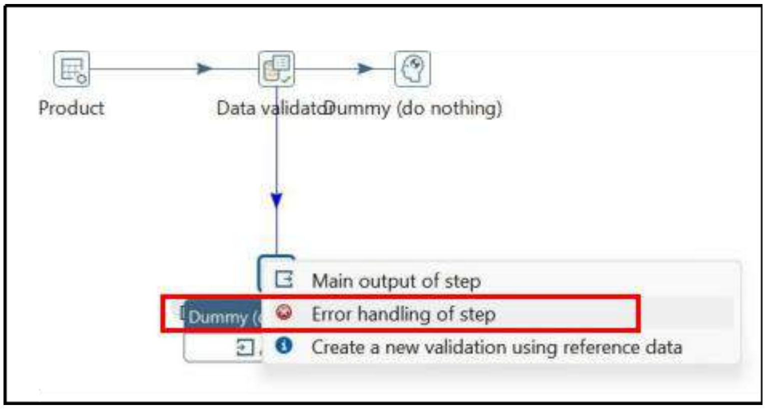
Dummy (do nothing)

This step type doesn't do anything.
It's useful however when testing things or in certain situations where you want to split streams.

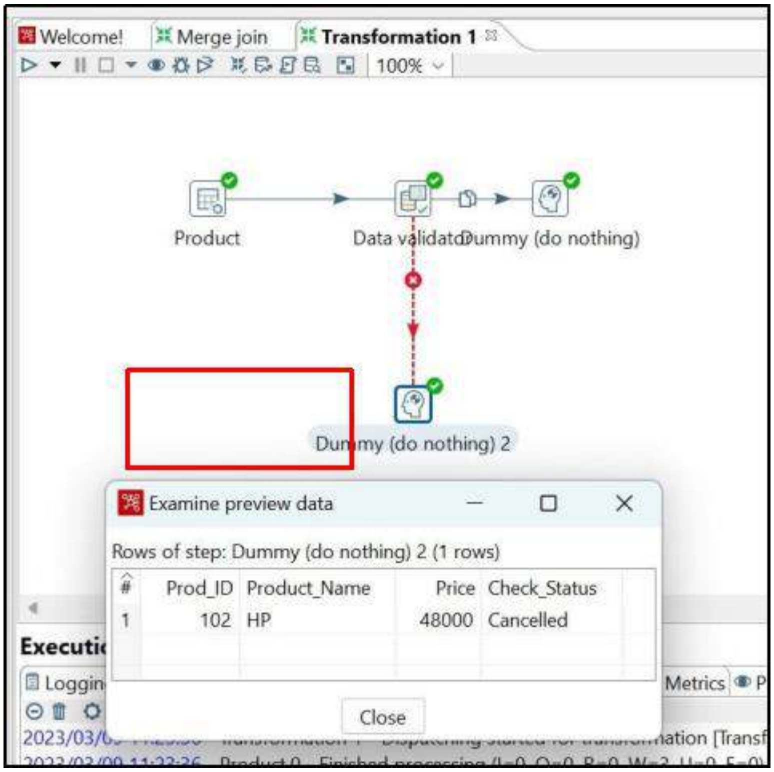
Hold the mouse pointer on Data validator and select and drag the output connector to the Dummy. Select Main output of step.



Step 5: Drag and drop another Dummy from Flow folder under the Design tab and connect it to the data validator. Select Error handling of step. In the next window click in Copy.


Product
Data validatDummy (do nothing)
Dummy (
Main output of step
Error handling of step
Create a new validation using reference data

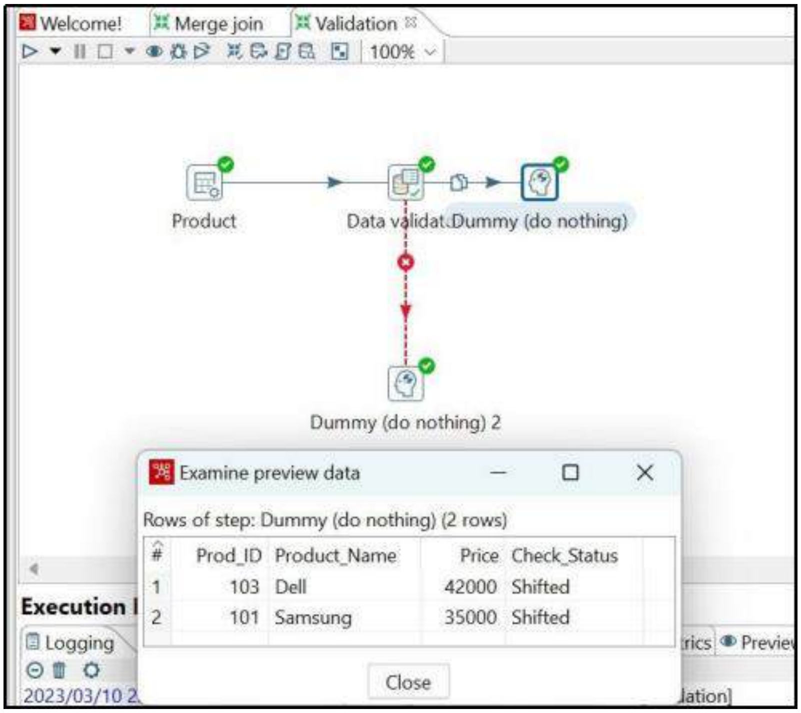
Step 6: Quick Launch the transformation selecting one dummy file each.



Examine preview data

Rows of step: Dummy (do nothing) 2 (1 rows)

#	Prod_ID	Product_Name	Price	Check_Status
1	102	HP	48000	Cancelled



Examine preview data

Rows of step: Dummy (do nothing) (2 rows)

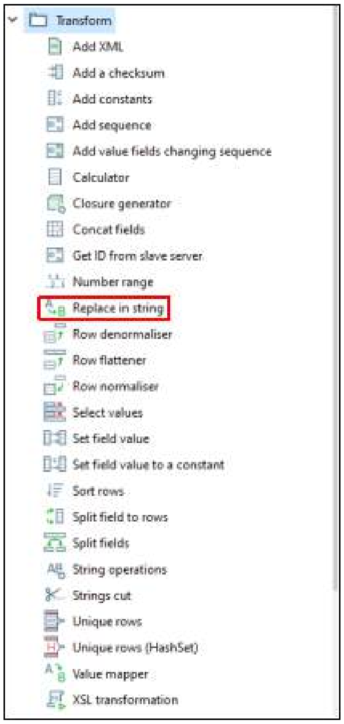
#	Prod_ID	Product_Name	Price	Check_Status
1	103	Dell	42000	Shifted
2	101	Samsung	35000	Shifted

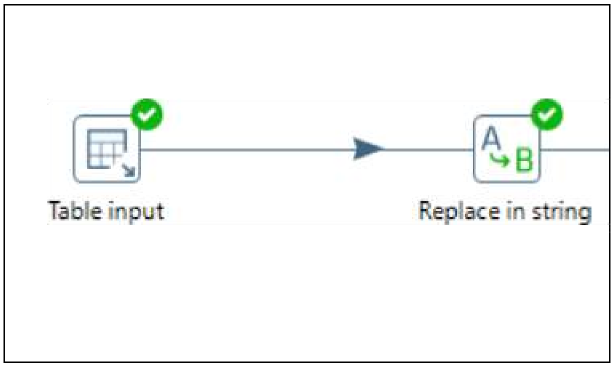
TRANSFORMATION10: Replace Strings

Step 1: Repeat Steps 2 and 3 from TRANSFORMATION 1 (Import output table of replace in strings transformation as Table input).

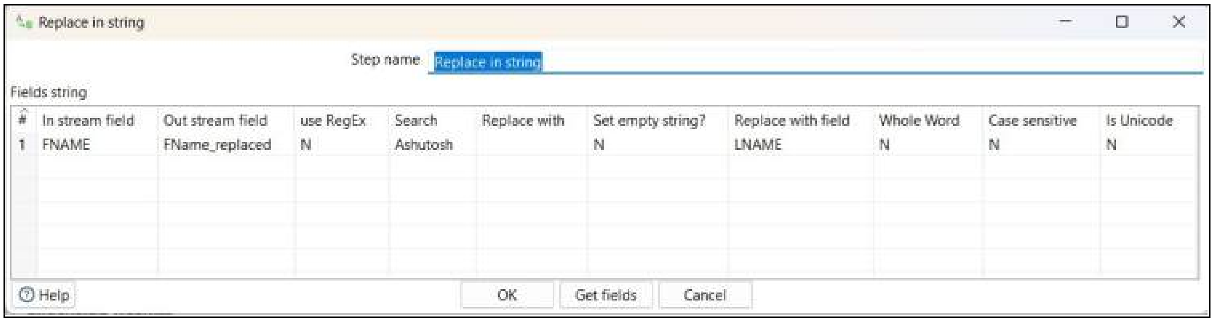
Step 2: Perform Transformation.

Drag and drop replace in strings from *Transform* folder under *Design* tab.





Step 3: Double click on Replace in String and fill the following fields

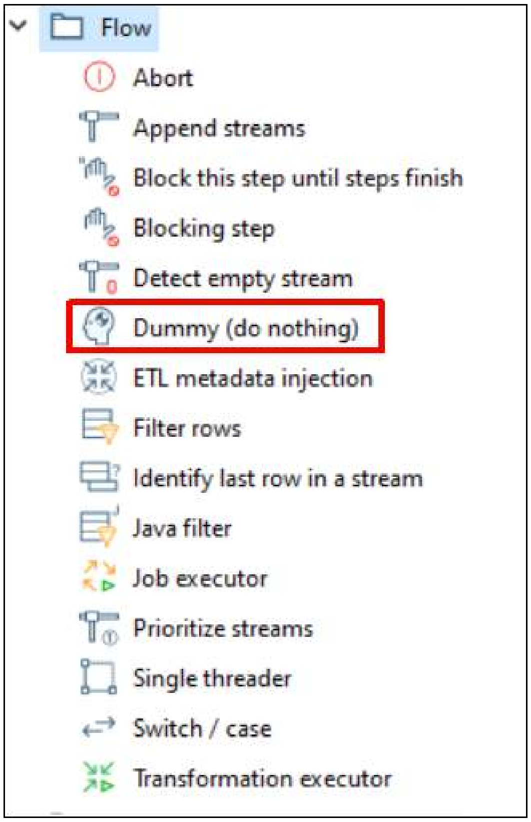


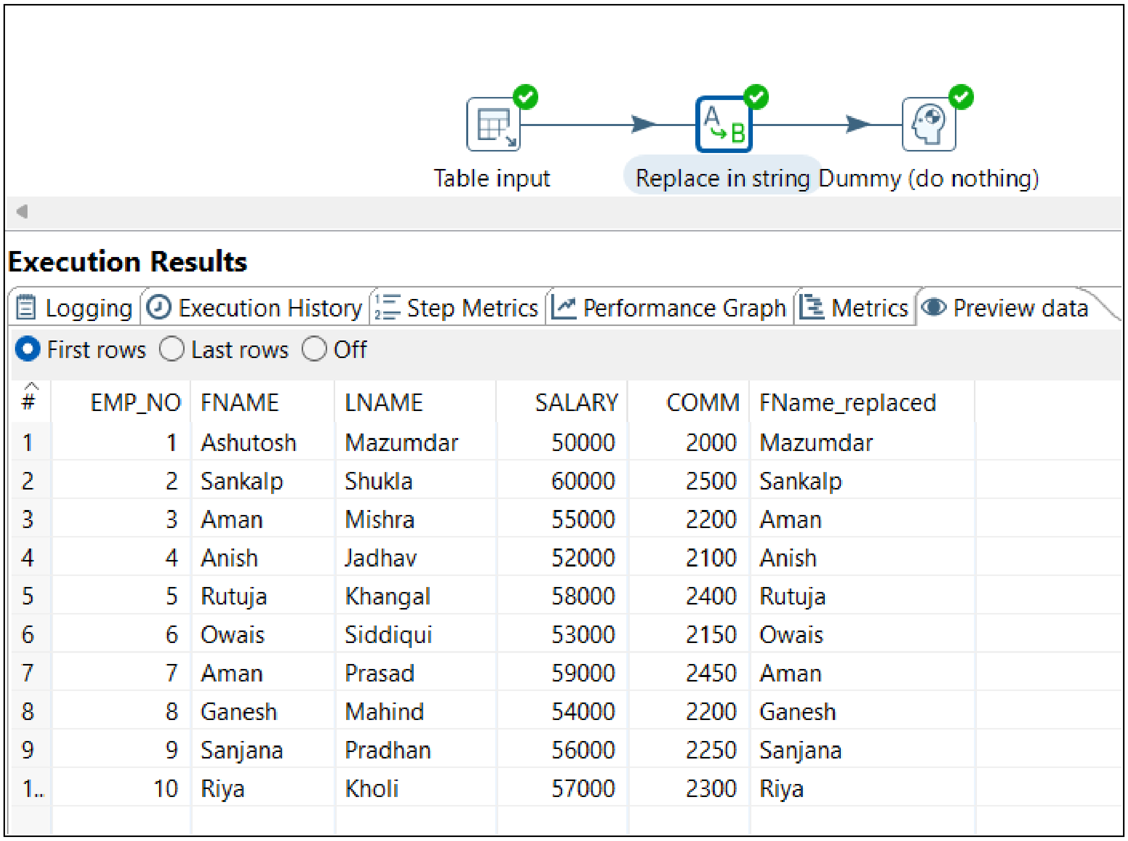
Step name Replace in string

Fields string

#	In stream field	Out stream field	use RegEx	Search	Replace with	Set empty string?	Replace with field	Whole Word	Case sensitive	Is Unicode
1	FNAME	FName_replaced	N	Ashutosh		N	LNAME	N	N	N

From Flow Double click or Drag Dummy





Execution Results

Logging | Execution History | Step Metrics | Performance Graph | Metrics | Preview data

● First rows ○ Last rows ○ Off

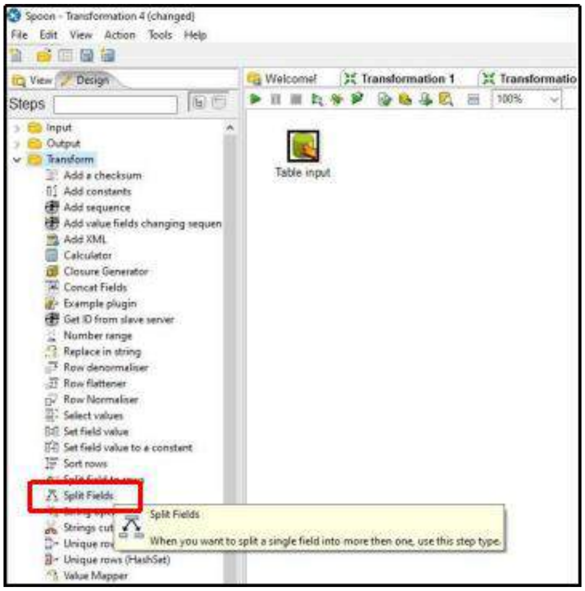
#	EMP_NO	FNAME	LNAME	SALARY	COMM	FName_replaced
1	1	Ashutosh	Mazumdar	50000	2000	Mazumdar
2	2	Sankalp	Shukla	60000	2500	Sankalp
3	3	Aman	Mishra	55000	2200	Aman
4	4	Anish	Jadhav	52000	2100	Anish
5	5	Rutuja	Khangal	58000	2400	Rutuja
6	6	Owais	Siddiqui	53000	2150	Owais
7	7	Aman	Prasad	59000	2450	Aman
8	8	Ganesh	Mahind	54000	2200	Ganesh
9	9	Sanjana	Pradhan	56000	2250	Sanjana
1..	10	Riya	Kholi	57000	2300	Riya

TRANSFORMATION 11: Split Fields.

Step 1: Repeat Steps 2 and 3 from TRANSFORMATION 1 (Import output table of concat fields transformation as Table input).

Step 2: Perform Transformation.

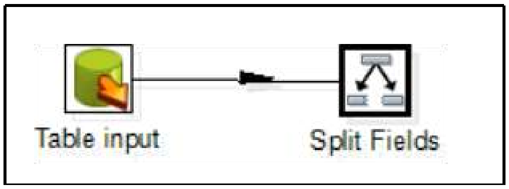
Drag and drop *Concat Fields* from *Transform* folder under *Design* tab.



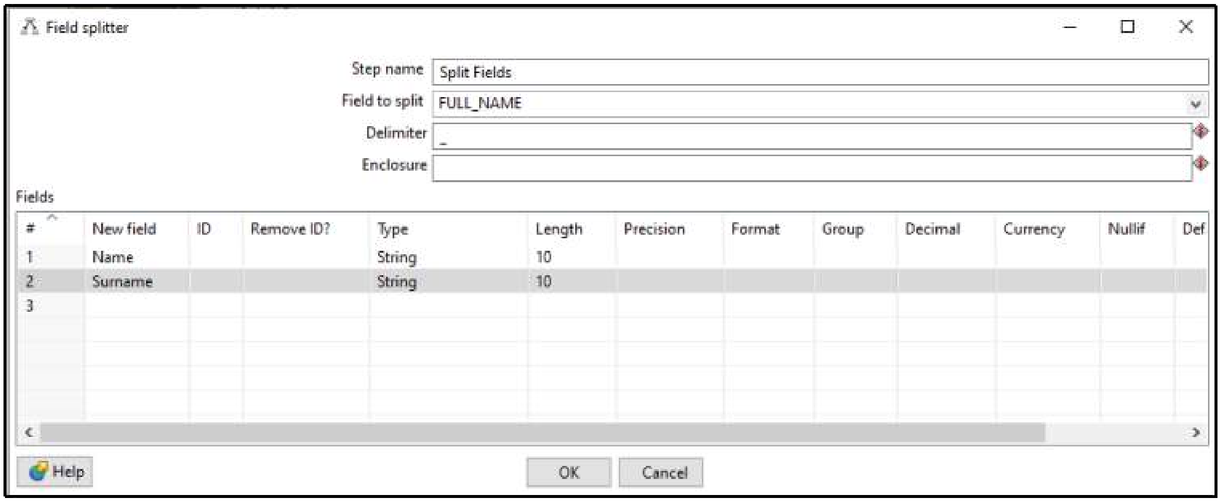
Split Fields

When you want to split a single field into more then one, use this step type.

Hold the mouse Pointer on Table input and select and drag the output connector to the Split Fields.



Double Click on *Split Fields* and fill in the details as shown below→Click on OK.



Step name: Split Fields

Field to split: FULL_NAME

Delimiter: _

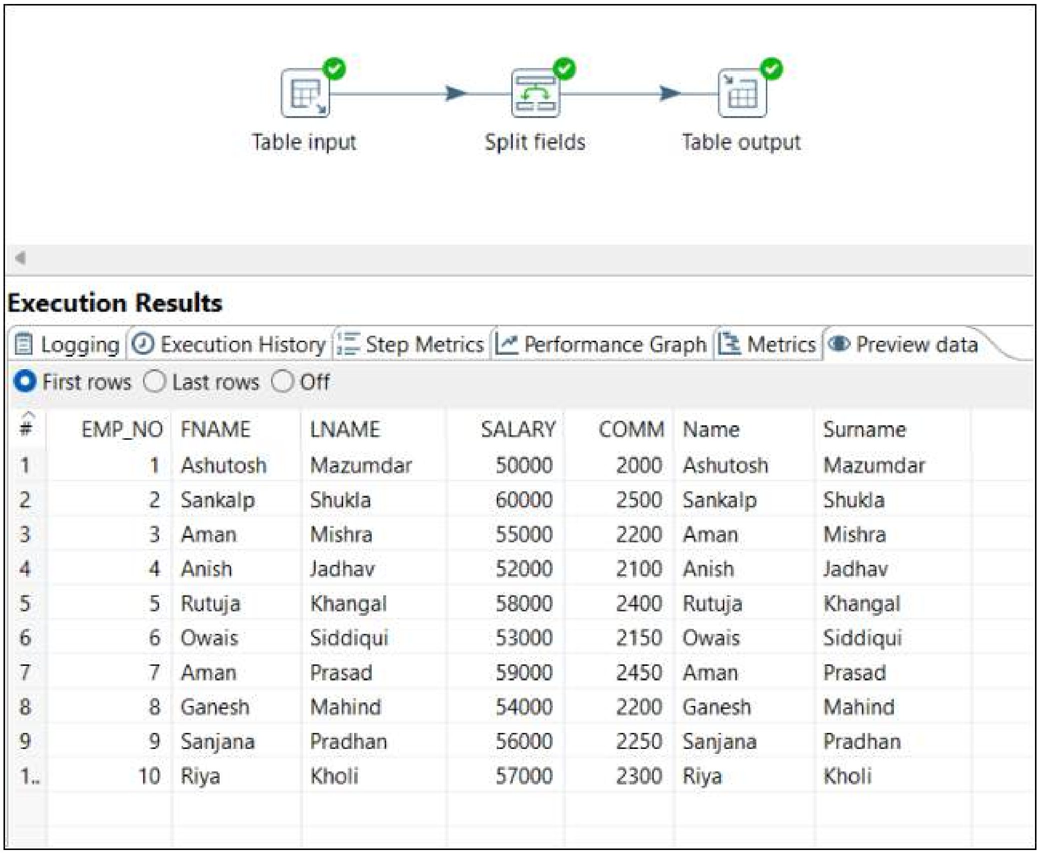
Enclosure:

Fields

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group	Decimal	Currency	Nullif	Def
1	Name			String	10							
2	Surname			String	10							
3												

Step 3: Repeat Step 4 from TRANSFORMATION 1.

If the Transformation is successful, you will see green ticks.

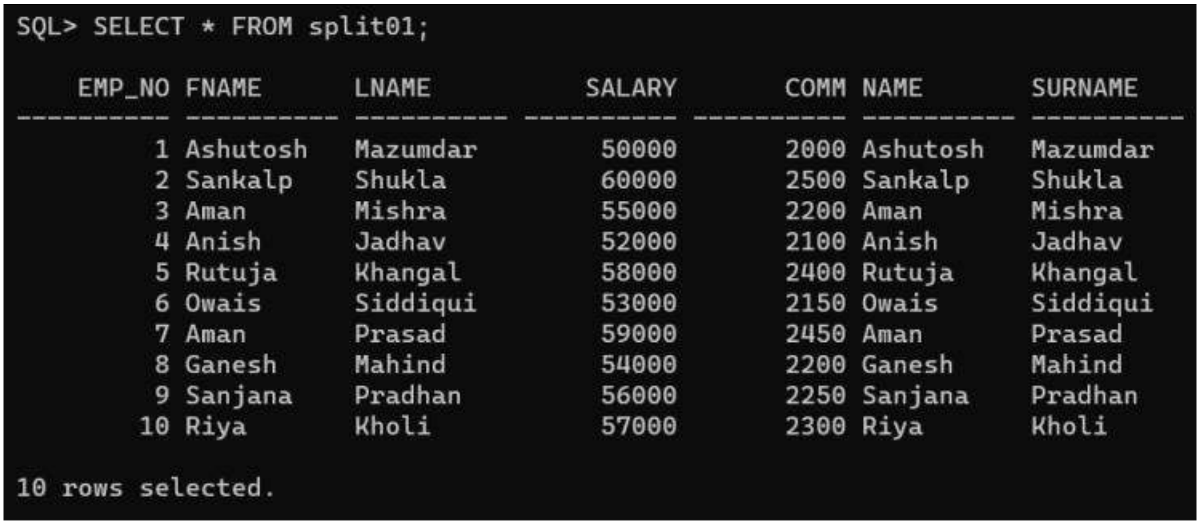


Execution Results

● First rows ○ Last rows ○ Off

#	EMP_NO	FNAME	LNAME	SALARY	COMM	Name	Surname
1	1	Ashutosh	Mazumdar	50000	2000	Ashutosh	Mazumdar
2	2	Sankalp	Shukla	60000	2500	Sankalp	Shukla
3	3	Aman	Mishra	55000	2200	Aman	Mishra
4	4	Anish	Jadhav	52000	2100	Anish	Jadhav
5	5	Rutuja	Khangal	58000	2400	Rutuja	Khangal
6	6	Owais	Siddiqui	53000	2150	Owais	Siddiqui
7	7	Aman	Prasad	59000	2450	Aman	Prasad
8	8	Ganesh	Mahind	54000	2200	Ganesh	Mahind
9	9	Sanjana	Pradhan	56000	2250	Sanjana	Pradhan
1..	10	Riya	Kholi	57000	2300	Riya	Kholi

Step 4: Run SQL query.



```
SQL> SELECT * FROM split01;

    EMP_NO FNAME      LNAME          SALARY       COMM NAME       SURNAME
---------- ---------- ---------- ---------- ---------- ---------- ----------
         1 Ashutosh   Mazumdar        50000       2000 Ashutosh   Mazumdar
         2 Sankalp    Shukla          60000       2500 Sankalp    Shukla
         3 Aman       Mishra          55000       2200 Aman       Mishra
         4 Anish      Jadhav          52000       2100 Anish      Jadhav
         5 Rutuja     Khangal         58000       2400 Rutuja     Khangal
         6 Owais      Siddiqui        53000       2150 Owais      Siddiqui
         7 Aman       Prasad          59000       2450 Aman       Prasad
         8 Ganesh     Mahind          54000       2200 Ganesh     Mahind
         9 Sanjana    Pradhan         56000       2250 Sanjana    Pradhan
        10 Riya       Kholi           57000       2300 Riya       Kholi

10 rows selected.
```