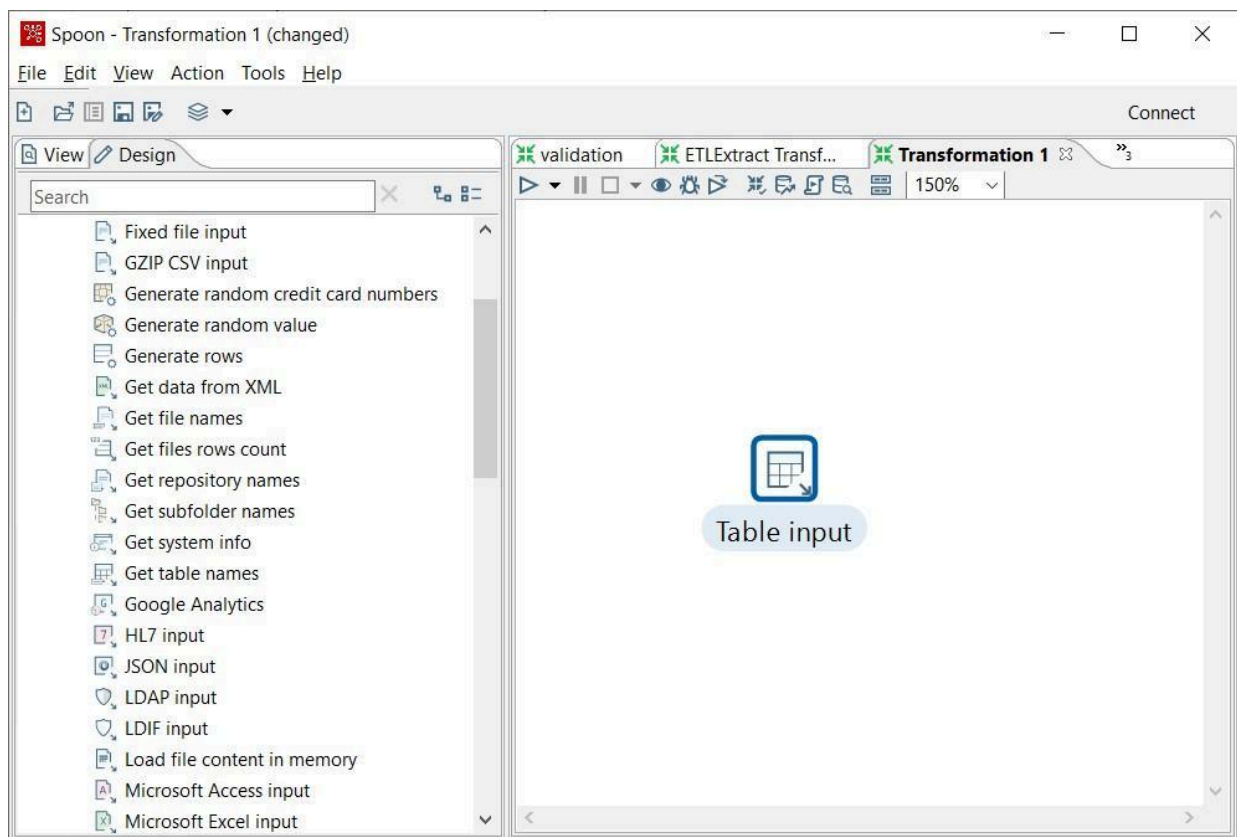


**Aim: Implementation of ETL transformation with pentaho like Copy data from source(table/excel/oracle) and store it to target (table/excel/oracle)**

1.Open new transformation

2.Click on Input and drag and drop table input on the screen.



### 3. Double click on this table\_input icon

Table input

Step name

Connection

SQL

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

Line 1 Column 0

Store column info in step meta ☐

Enable lazy conversion ☐

Replace variables in script? ☐

Insert data from step

Execute for each row? ☐

Limit size

<https://www.oracle.com/database/technologies/appdev/jdbc-downloads.html>

```
C:\Users\veera>sqlplus sys as sysdba
```

```
SQL*Plus: Release 21.0.0.0.0 - Production on Sat Oct 12 17:28:42 2024  
Version 21.3.0.0.0
```

```
Copyright (c) 1982, 2021, Oracle. All rights reserved.
```

```
Enter password:
```

```
Connected to:  
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production  
Version 21.3.0.0.0
```

```
SQL> SELECT instance_name, status FROM v$instance;
```

INSTANCE_NAME	STATUS
xe	OPEN

```
SQL> |
```

4. Click on new for creating a connection.

Database Connection

General  
Advanced  
Options  
Pooling  
Clustering

Connection name:  
Test

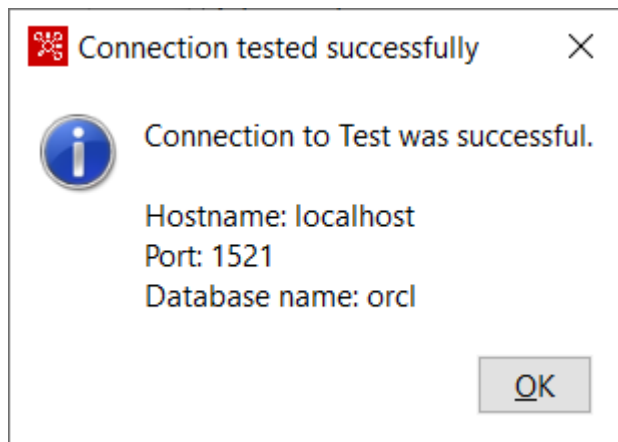
Connection type:  
Oracle  
Oracle RDB  
Palo MOLAP Server  
Pentaho Data Services  
PostgreSQL  
Redshift  
Remedy Action Request System  
SAP ERP System  
SQLite  
Snowflake  
SparkSQL  
Sybase  
Teradata

Access:  
Native (JDBC)  
ODBC  
OCI  
JNDI

Settings  
Host Name:  
localhost  
Database Name:  
orcl  
Tablespace for Data  
Tablespace for Indices  
Port Number:  
1521  
Username:  
system  
Password:  
.....

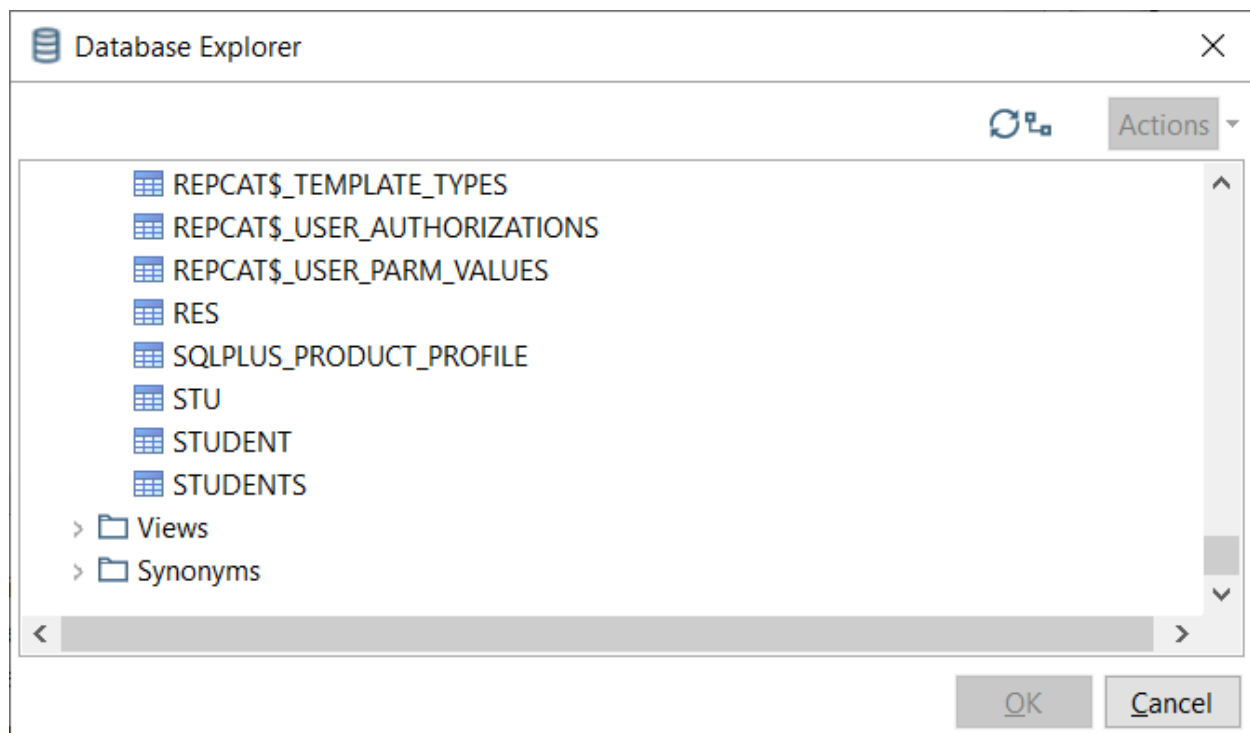
Test Feature List Explore OK Cancel

5. Click on test.



Click on OK.

Then click on get SQL select statement.



6. Select your table and click on OK.  
Then Click on Preview and then ok.  
Below given page will get open

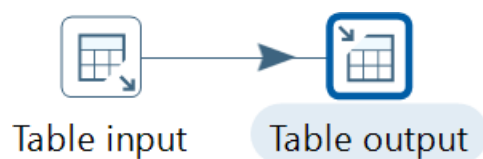
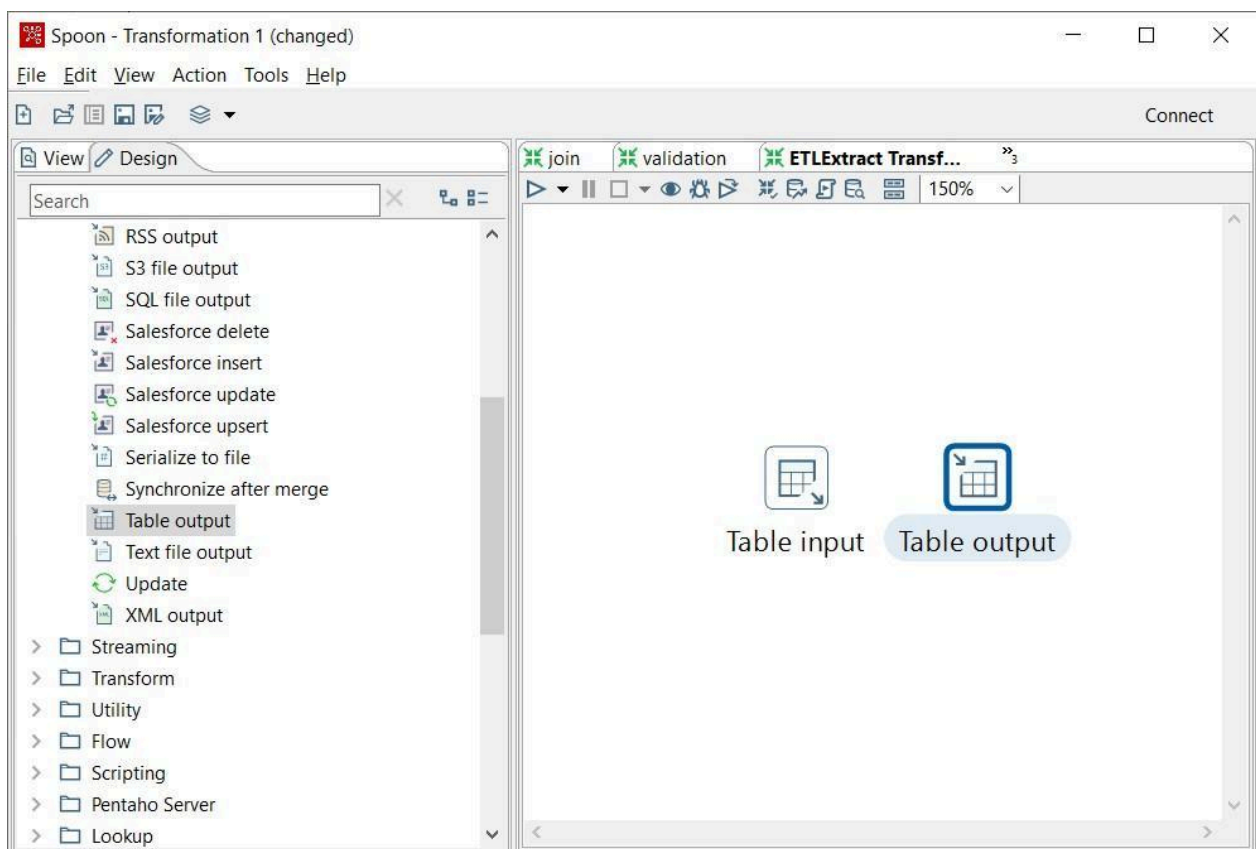
Rows of step: Table input (6 rows)

#	FNAME	LNAME	PERCENTAGE	AGE
1	Manoj	Singh	35.0	20.0
2	Meenal	Shah	42.0	21.0
3	Monica	kamble	62.0	21.0
4	Manoj	Singh	71.0	20.0
5	Manoj	Singh	83.0	20.0
6	Manoj	Singh	92.0	20.0

7. Click on close and then ok.

Data Input is Done Now we have to create a table for getting our Output. Go in Output and Drag & Drop table\_output.

8. Make a Connection between them.



9. Double click on table output.

Table output

Step name: Table output

Connection: Test [Edit... New... Wizard...]

Target schema: [Browse...]

Target table: stu [Browse...]

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	FNAME	FNAME
2	LNAME	LNAME
3	PERCENTAGE	PERCENTAGE
4	AGE	AGE

[Get fields]

Enter field mapping

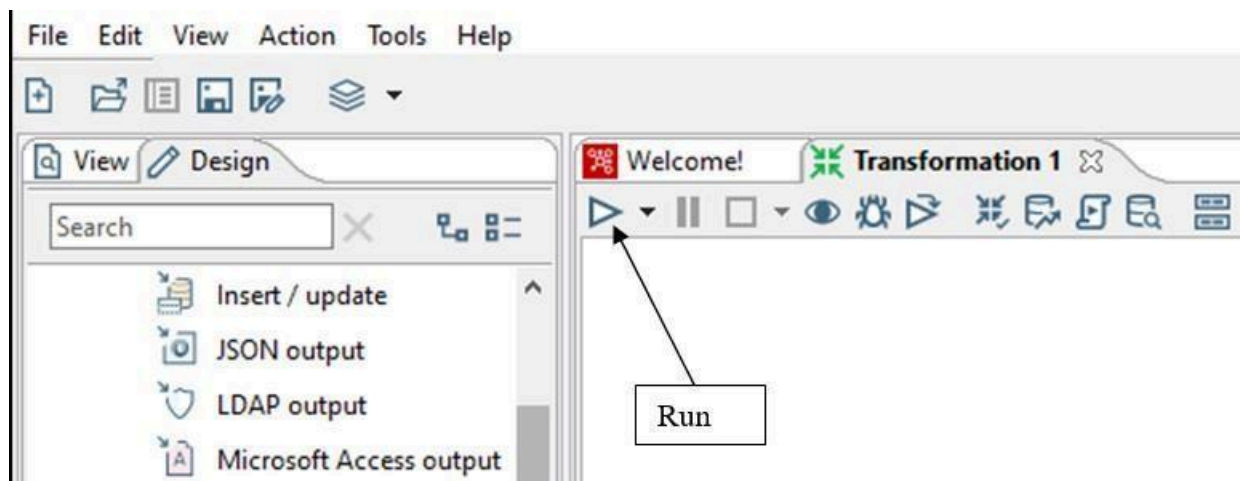
[Help] [OK] [Cancel] [SQL]

10. Give name to target table in which our input is going to get stored and specify database fields.

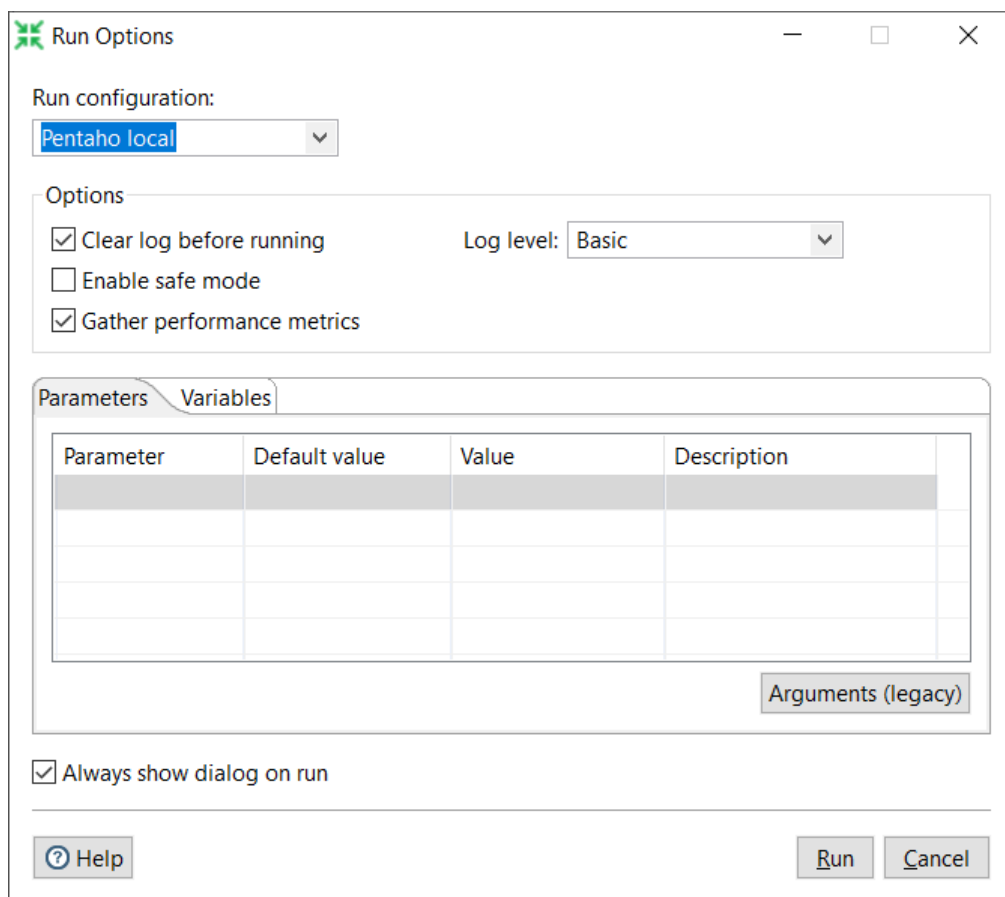
Click on SQL.

11. Click on execute then ok and close. Go in database fields and click on get field

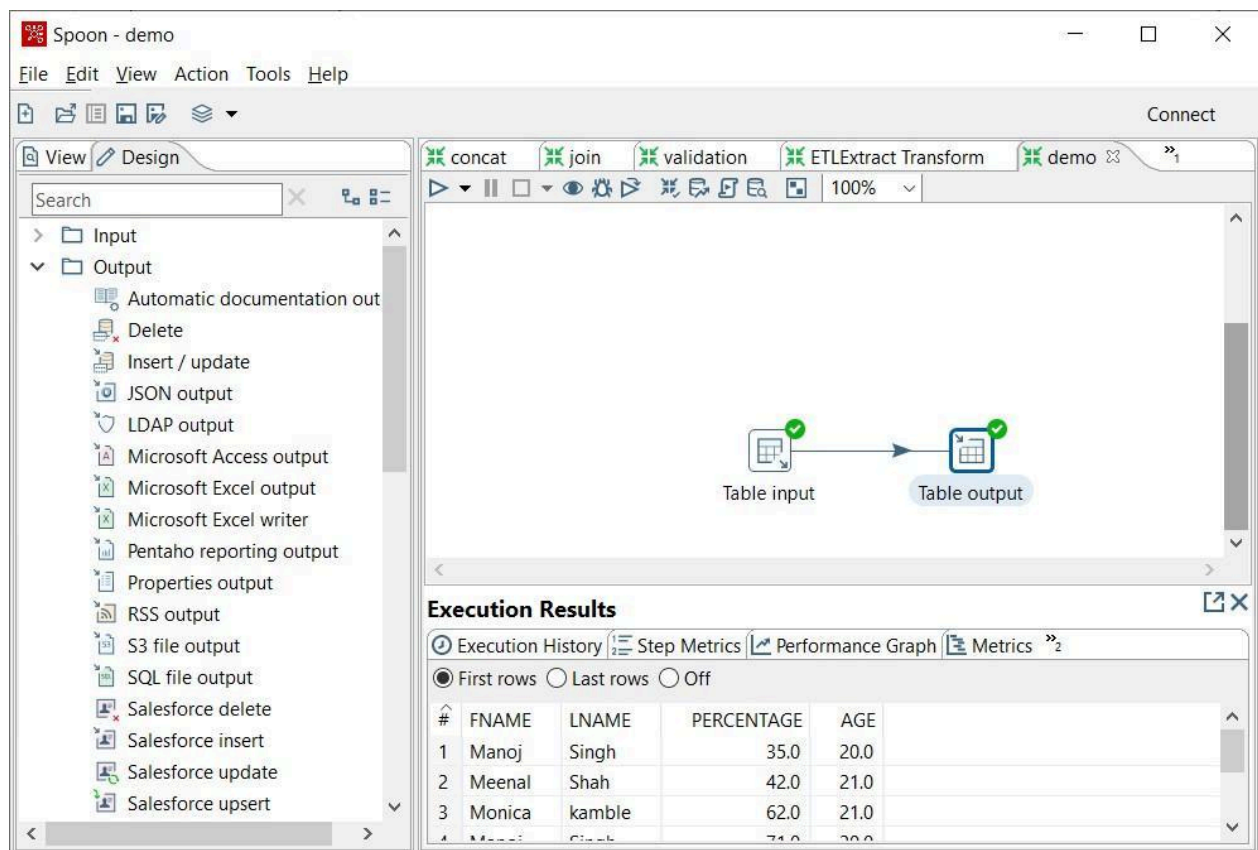
Click on ok and then run



12. Click on Run.





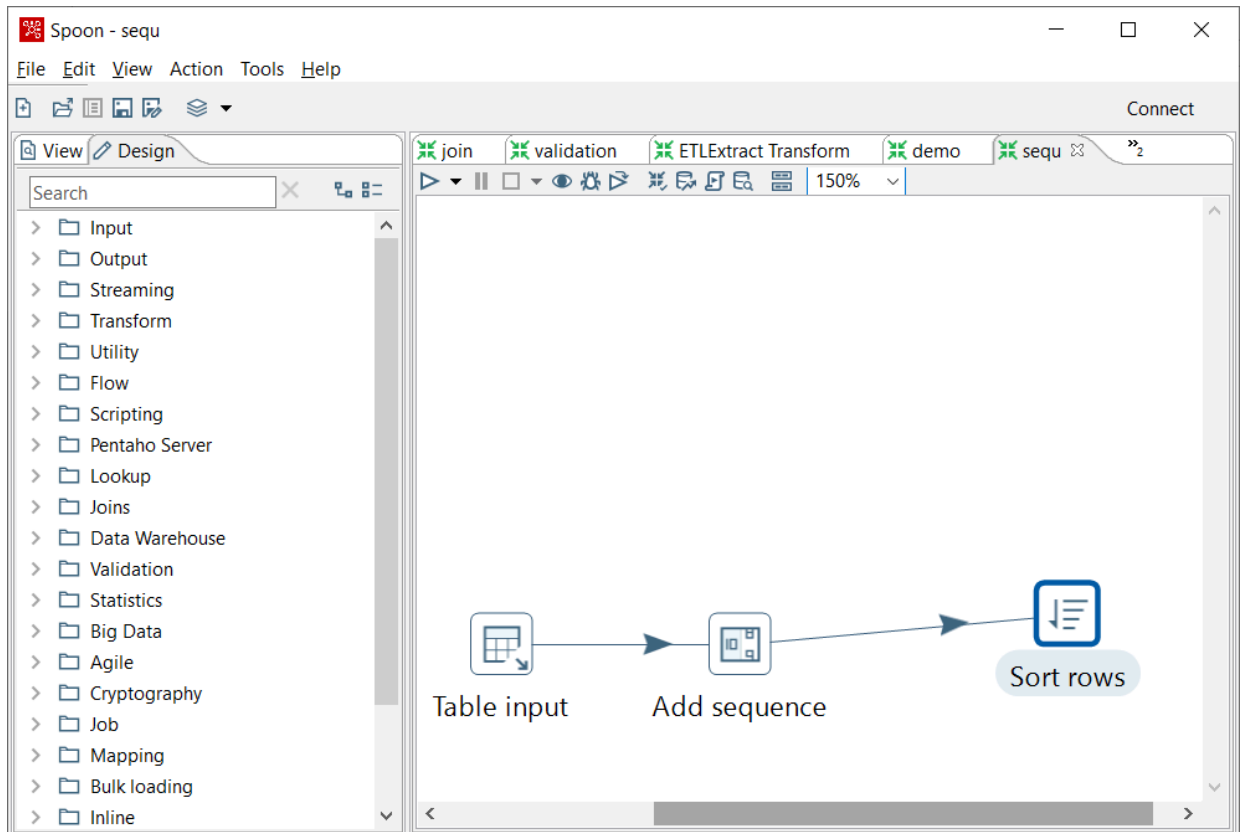


13. It is successfully done.

Now for checking our output go in SQL PLUS and type the table name which you have given in table\_output.

## Aim: Implementation of ETL transformation with Pentaho like Adding sequence.

1. Drag and drop a table input, sort rows and add sequence and make connection between them as shown



Input table:-

Table input

Step nameTable input

Connectionseq

Edit...New...Wizard...

SQL

Get SQL select statement...

```
SELECT
  ACCNO
  CUSTNAME
  BRANCH
  ACCTBAL
FROM SYSTEM.ACCOUNTS
```

Line 1 Column 0

Store column info in step meta☐

Enable lazy conversion☐

Replace variables in script?☐

Insert data from step

Execute for each row?☐

Limit size0

Help

OK

Preview

Cancel

## 2.Preview data

Rows of step: Table input (6 rows)

#	ACCNO	CUSTNAME	BRANCH	ACCTBAL
1	6.0	Mansi	Thane	80.0
2	4.0	Chutki	Bhuvaneshwar	2000.0
3	5.0	Nancy	Bhuvaneshwar	2000.0
4	1.0	Sugar	Bhuvaneshwar	5000.0
5	2.0	Arman	Fun	9998.0
6	3.0	Sanskar	Airport	7000.0

3. Double click on add sequence:-

**Add sequence**

Step name

Name of value

Use a database to generate the sequence

Use DB to get sequence? ☐

Connection  Edit... New... Wizard...

Schema name  Schemas...

Sequence name  Sequences...

Use a transformation counter to generate the sequence

Use counter to calculate sequence? ☒

Counter name (optional)

Start at value

Increment by

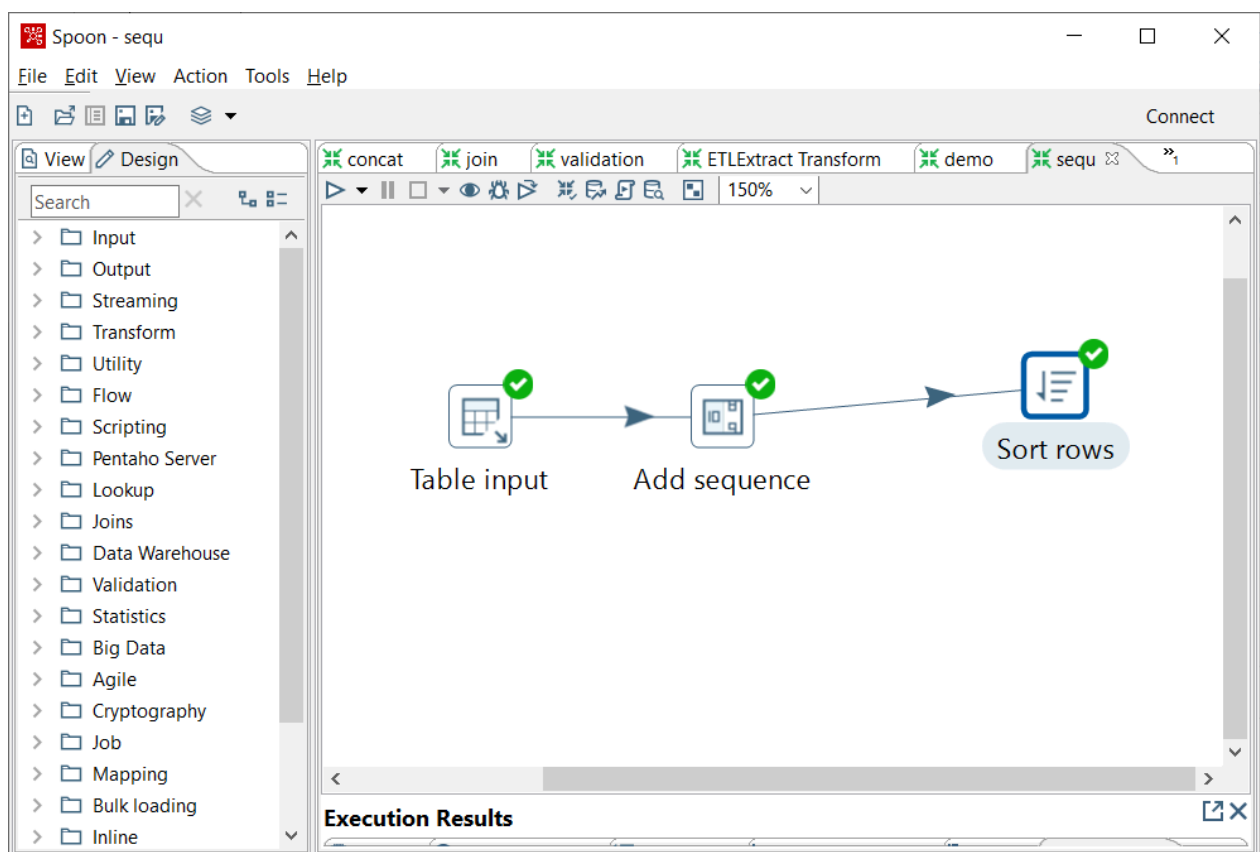
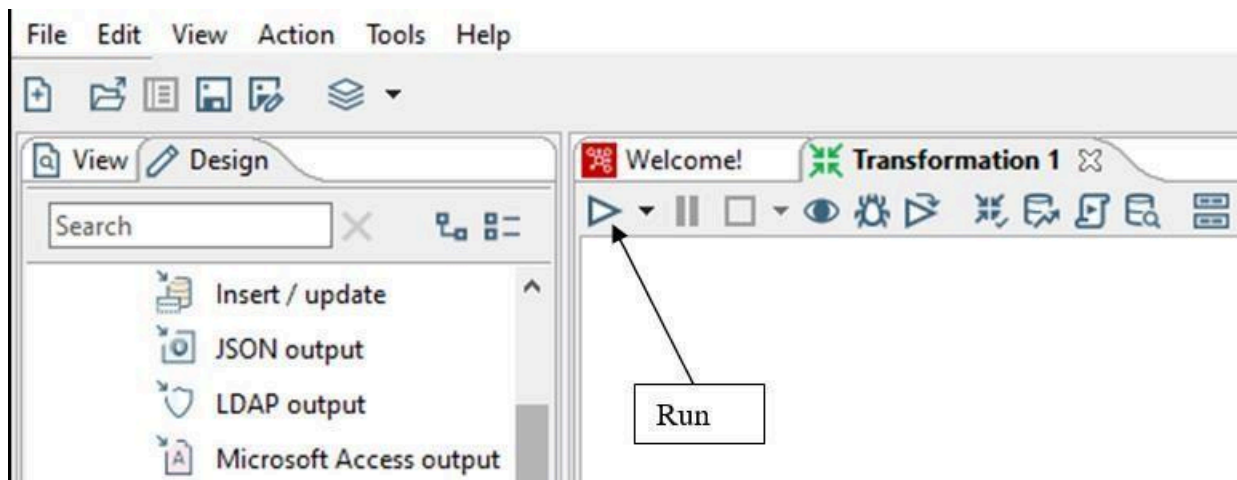
Maximum value

Help OK Cancel

#### 4. Double click on sort rows:-

[illegible]

5. Now click on run



6. Now click on preview data

The sequence according to the descending order of the ACCTBAL is shown

Execution Results

Logging Execution History Step Metrics Performance Graph Metrics Preview data

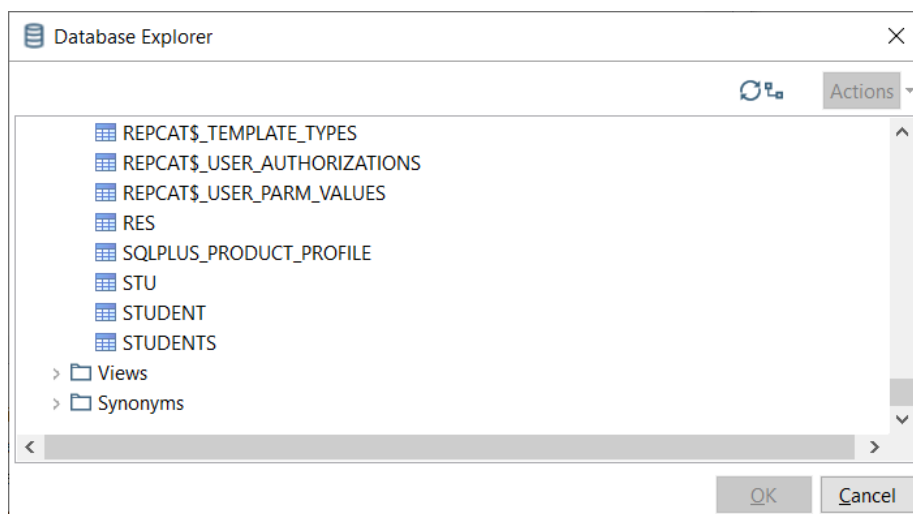
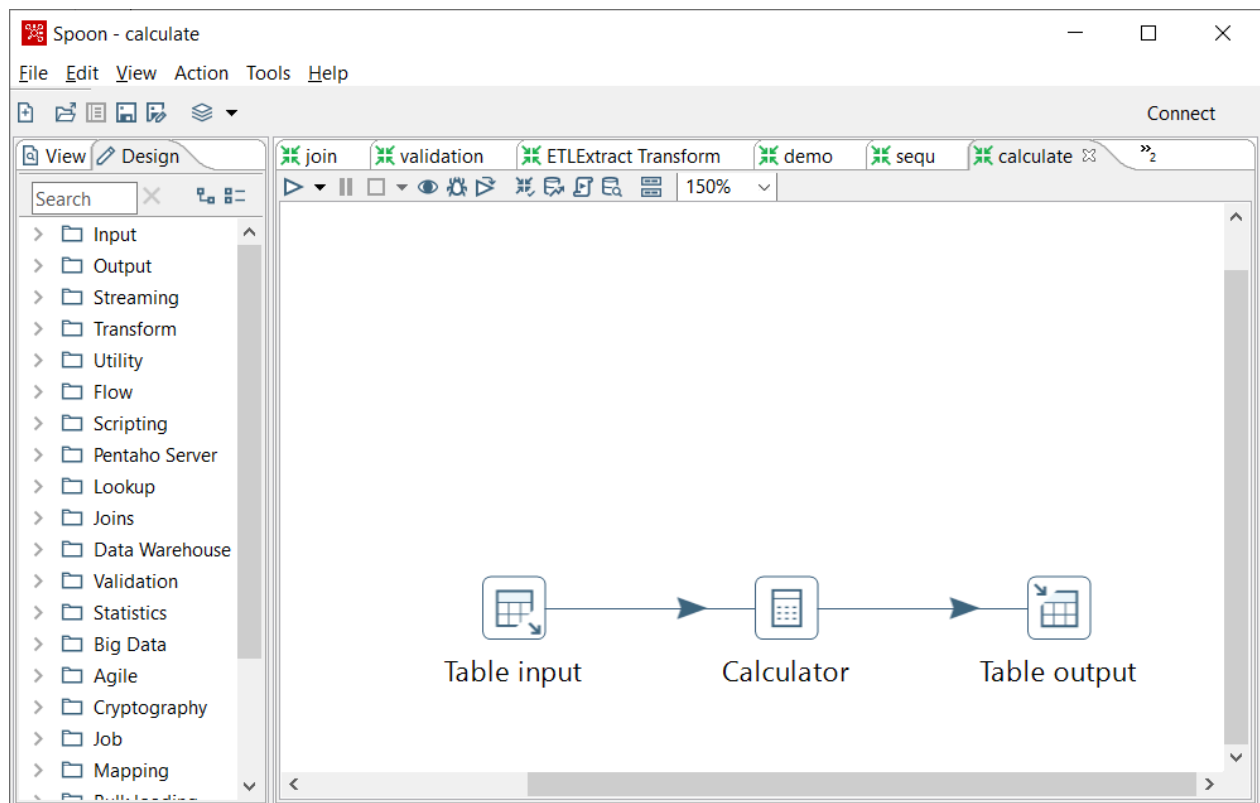
☒ First rows ☐ Last rows ☐ Off

#	ACCNO	CUSTNAME	BRANCH	ACCTBAL	sequence	
1	2.0	Arman	Fun	9998.0	5	
2	3.0	Sanskar	Airport	7000.0	6	
3	1.0	Sugar	Bhuvaneshwar	5000.0	4	
4	4.0	Chutki	Bhuvaneshwar	2000.0	2	
5	5.0	Nancy	Bhuvaneshwar	2000.0	3	
6	6.0	Mansi	Thane	80.0	1	

## Aim:-Implementation of ETL transformation with Pentaho like calculator.

1. Drag and drop an input table, a calculator and an output table

2. In input table go to get SQL select statement and select a table



3. Click ok

Then Preview->close->OK

4. Then click on calculator

Calculator

Step name:

☒ Throw an error on non existing files

Fields:

#	New field	Calculation	Field A	Field B	Field C	Value type	Length	Precision	Remove	Conversion mask	Decimal symbol	Grouping symbol	Currency symbol
1	Addition	A + B	AGE	PERCENTAGE		Integer			N				

5. Click OK

Double click on table Output.

Write the target table

Click on get fields and get SQL statements and execute

Table output

Step name:

Connection:

Target schema:

Target table:

Commit size:

Truncate table: ☐

Ignore insert errors: ☐

Specify database:

Simple SQL editor

SQL statements, separated by semicolon ';'

```
ALTER TABLE stu ADD ( Addition INTEGER);
```

Line 1 column 0

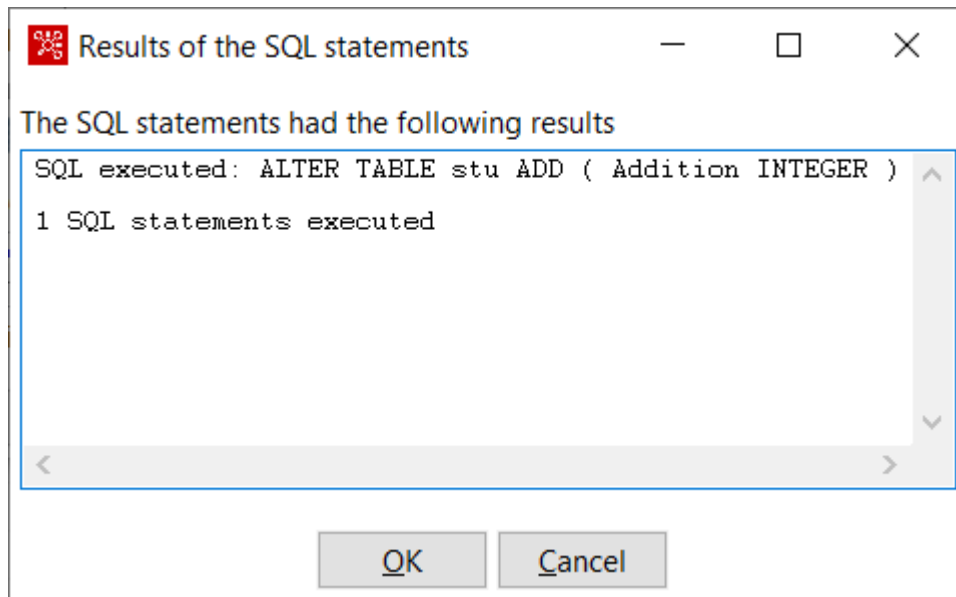
Table output - Database fields

Fields to insert:

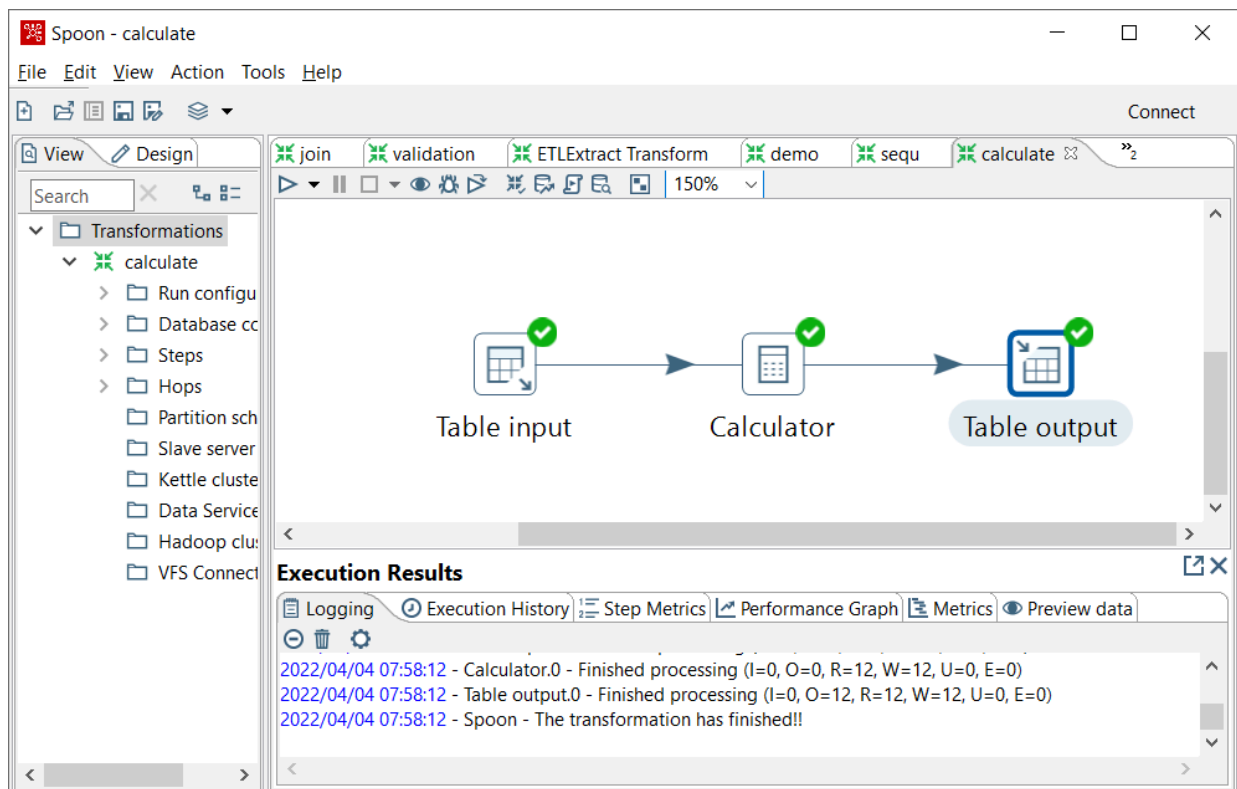
#	Table field	Stream field
1	FNAME	FNAME
2	LNAME	LNAME
3	PERCENTAGE	PERCENTAGE
4	AGE	AGE
5	Addition	Addition



## 6. Click ok



## 7. Then run the transformation.



8.Now click on preview data

#### Execution Results

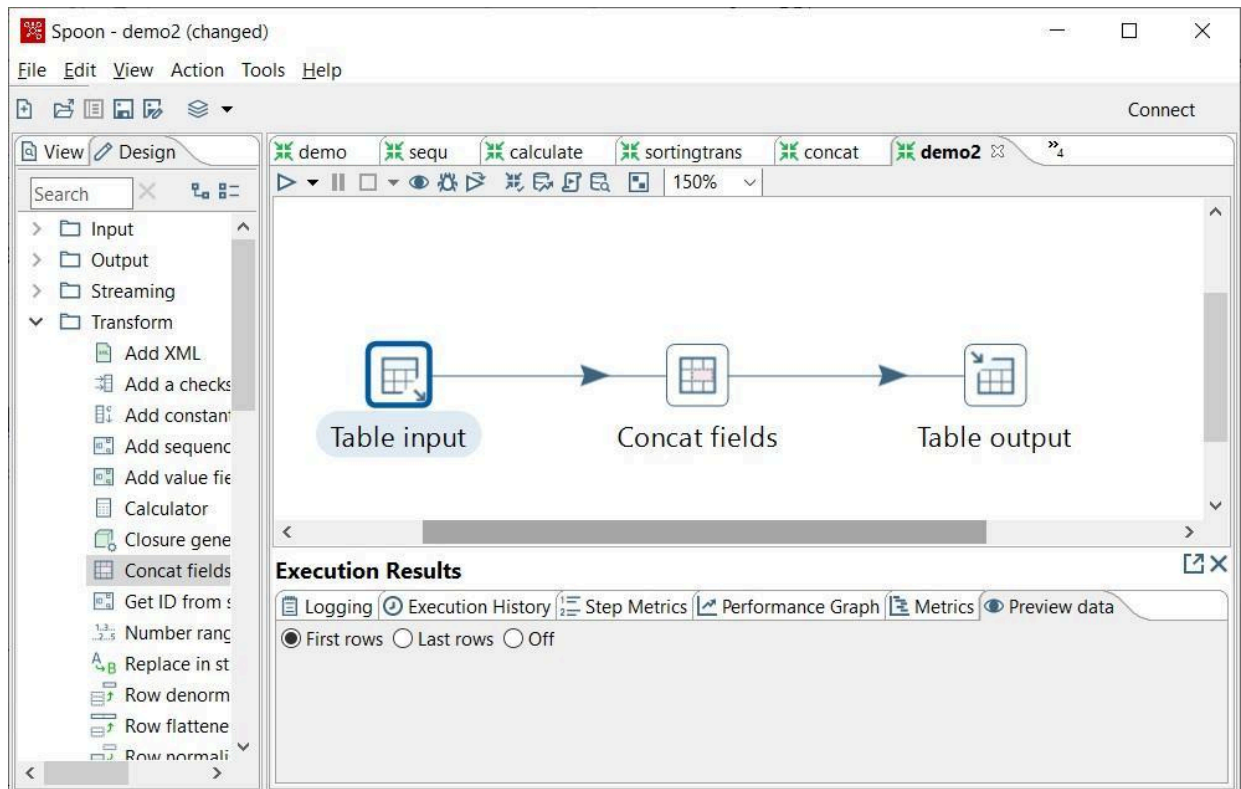
Logging	Execution History	Step Metrics	Performance Graph	Metrics	Preview data
<input checked="" type="radio"/> First rows <input type="radio"/> Last rows <input type="radio"/> Off					
#	FNAME	LNAME	PERCENTAGE	AGE	Addition
1	Manoj	Singh	35.0	20.0	55
2	Meenal	Shah	42.0	21.0	63
3	Monica	kamble	62.0	21.0	83
4	Manoj	Singh	71.0	20.0	91
5	Manoj	Singh	83.0	20.0	103
6	Manoj	Singh	92.0	20.0	112
7	Manoj	Singh	35.0	20.0	55
8	Meenal	Shah	42.0	21.0	63
9	Monica	kamble	62.0	21.0	83

9.Now Let's see the output in SQL PLUS

FNAME	LNAME	PERCENTAGE	AGE	ADDITION
Manoj	Singh	92	20	
Manoj	Singh	35	20	55
Meenal	Shah	42	21	63
Monica	kamble	62	21	83
Manoj	Singh	71	20	91
Manoj	Singh	83	20	103
Manoj	Singh	92	20	112
Manoj	Singh	35	20	55
Meenal	Shah	42	21	63
Monica	kamble	62	21	83

## Aim: Implementation of ETL transformation with Pentaho like Concatenate fields and split rows.

1. Drag and drop the table input, concat fields and table output and make connection between them as shown



2. Double click on table input and do the steps performed in previous practicals similarly.

Database Connection

General  
Advanced  
Options  
Pooling  
Clustering

Connection name:  
Split

Connection type:  
Oracle  
Oracle RDB  
Palo MOLAP Server  
Pentaho Data Services  
PostgreSQL  
Redshift  
Remedy Action Request System  
SAP ERP System  
SQLite  
Snowflake  
SparkSQL  
Sybase

Access:  
Native (JDBC)  
ODBC  
OCI  
JNDI

Settings  
Host Name:  
localhost  
Database Name:  
orcl  
Tablespace for Data  
Tablespace for Indices  
Port Number:  
1521  
Username:  
system  
Password:  
.....

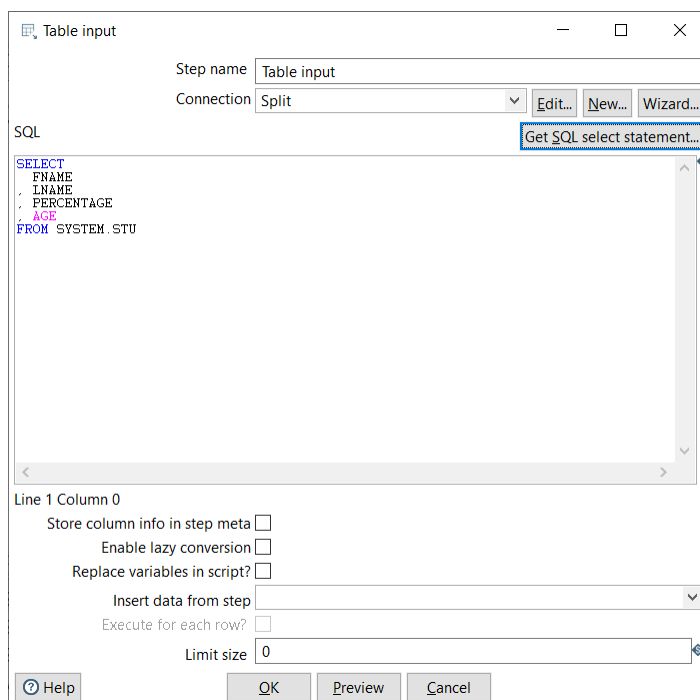
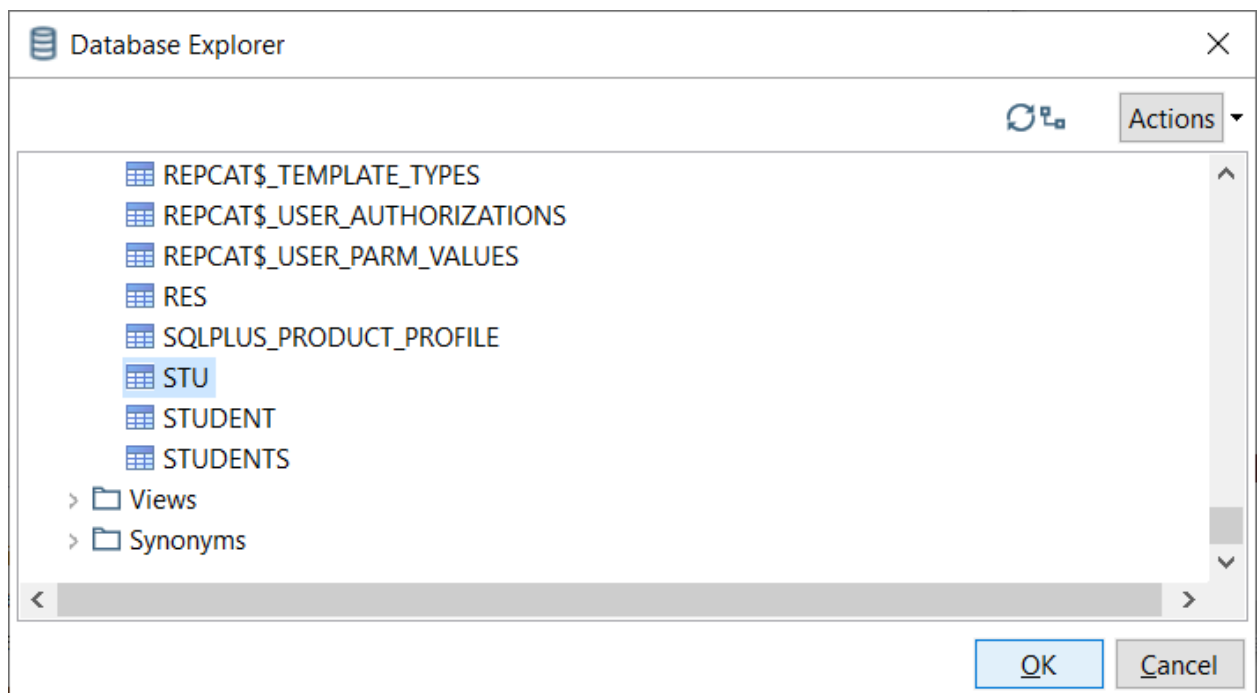
Test Feature List Explore OK Cancel

Connection tested successfully

Connection to Split was successful.

Hostname: localhost  
Port: 1521  
Database name: orcl

OK



Rows of step: Table input (7 rows)

#	FNAME	LNAME	PERCENTAGE	AGE
1	Manoj	Singh	35.0	20.0
2	Meenal	Shah	42.0	21.0
3	Ajay	Mane	53.0	22.0
4	Monica	kamble	62.0	21.0
5	Manoj	Singh	71.0	20.0
6	Manoj	Singh	83.0	20.0
7	Manoj	Singh	92.0	20.0

3. Now double click on concat fields and fill the fiels as given. Remove the semicolon from the separator and put space.

And click OK.

Concat fields

Step name: Concat fields

Target Field Name: FullName

Length of Target Field: 0

Separator: |

Enclosure: "

Insert TAB

Insert "

Fields Advanced

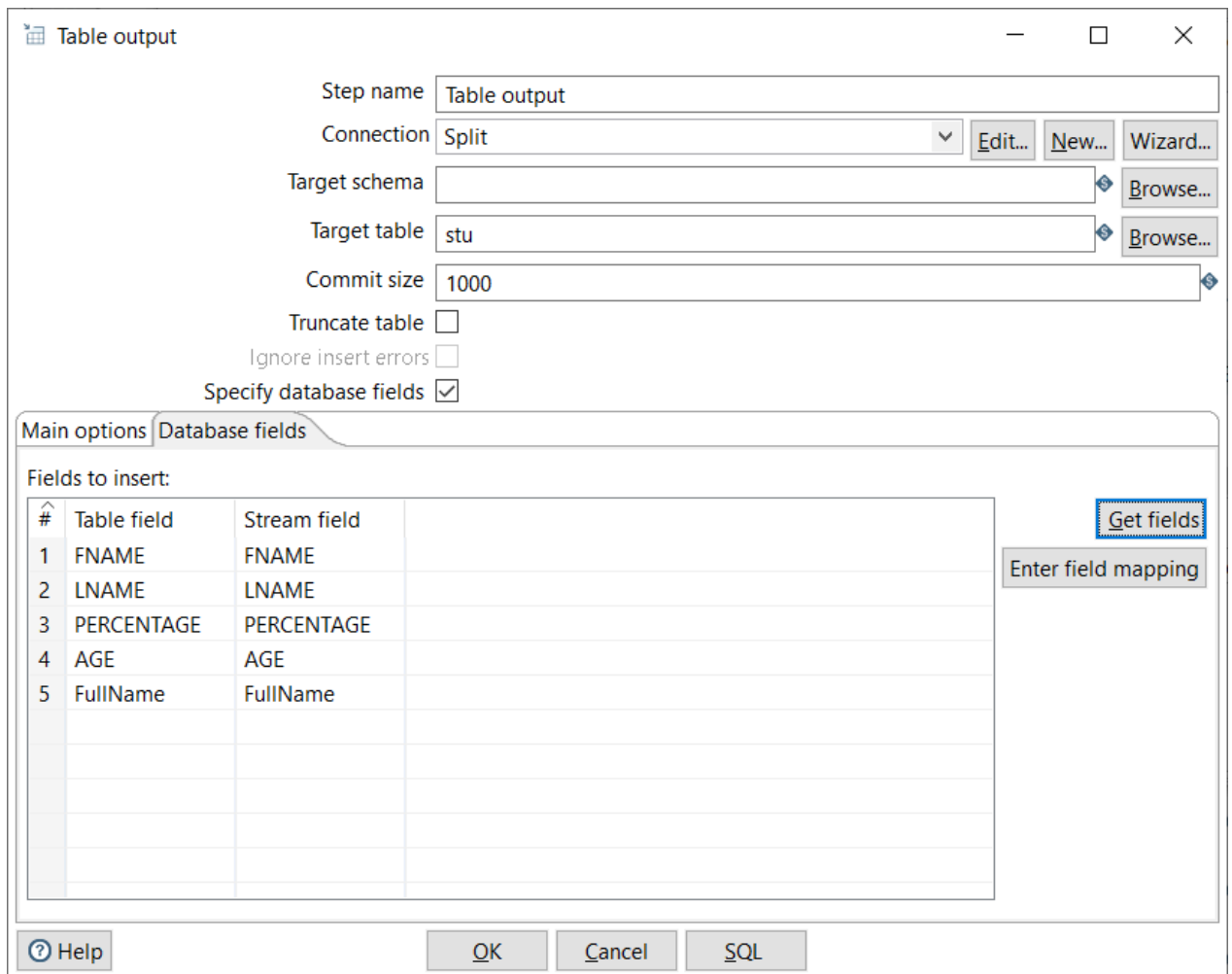
#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim Type	Null
1	FNAME	String								
2	LNAME	String								

Get Fields Minimal width

Help OK Cancel

4. Now double click on table output

And perform the steps of the table output as done before



The 'Table output' window is used to configure the output of a table. It includes fields for Step name, Connection, Target schema, Target table, Commit size, Truncate table, Ignore insert errors, and Specify database fields. The 'Database fields' tab is active, showing a table of fields to insert and a 'Get fields' button.

Step name: Table output

Connection: Split

Target schema:

Target table: stu

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options | Database fields

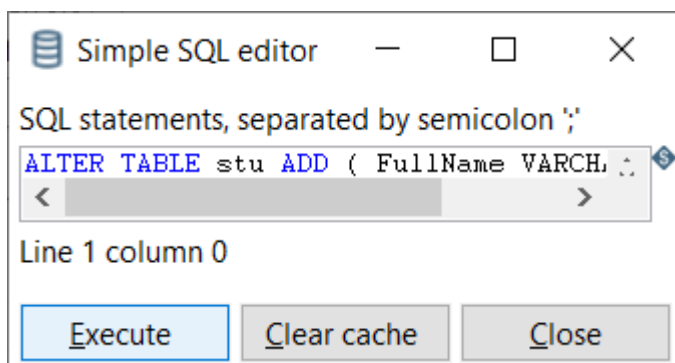
Fields to insert:

#	Table field	Stream field
1	FNAME	FNAME
2	LNAME	LNAME
3	PERCENTAGE	PERCENTAGE
4	AGE	AGE
5	FullName	FullName

Get fields

Enter field mapping

Help OK Cancel SQL



The 'Simple SQL editor' window displays SQL statements, separated by semicolon. The current statement is 'ALTER TABLE stu ADD ( FullName VARCHAR(255) );'. The editor includes a text area, a status bar showing 'Line 1 column 0', and buttons for Execute, Clear cache, and Close.

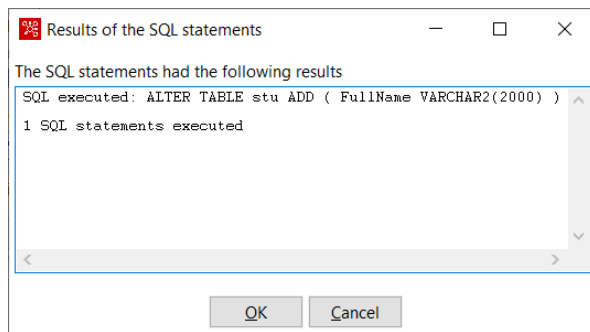
Simple SQL editor

SQL statements, separated by semicolon ';'

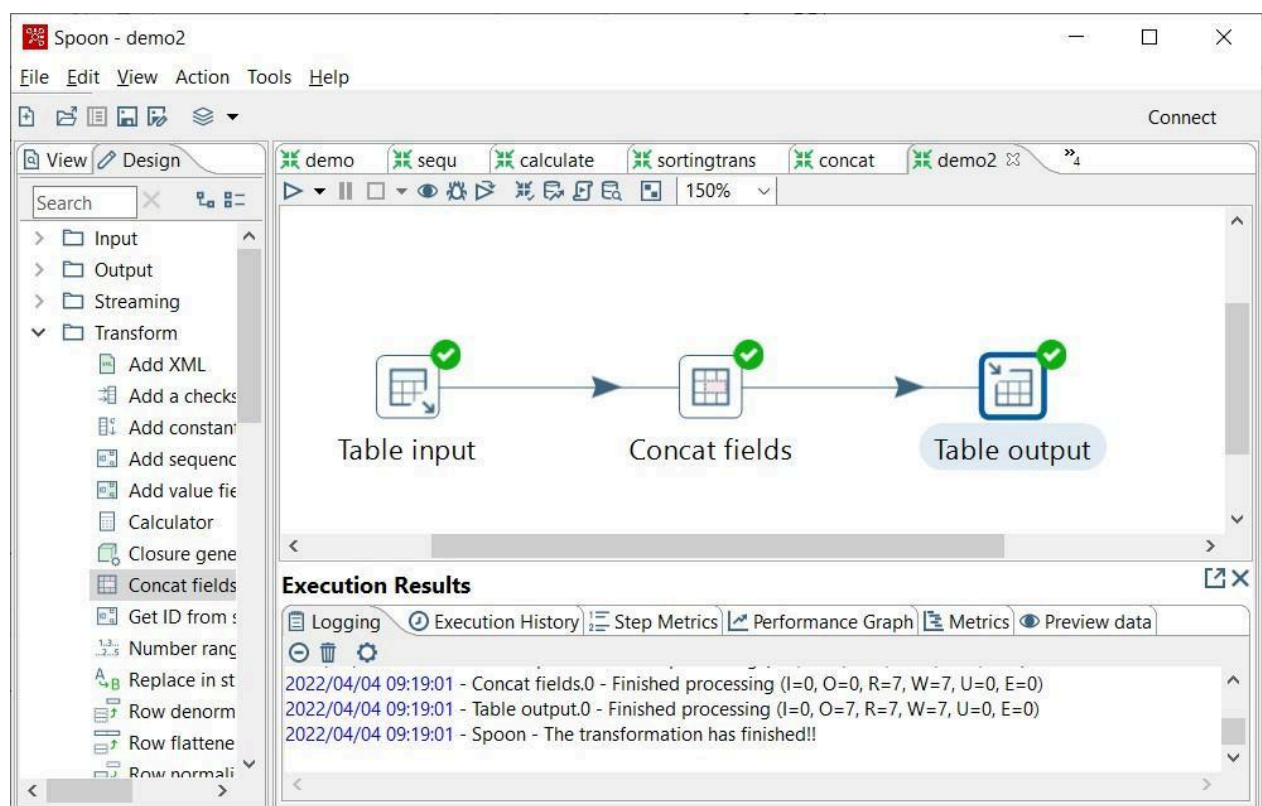
ALTER TABLE stu ADD ( FullName VARCHAR(255) );

Line 1 column 0

Execute Clear cache Close



## 5. Now run the transformation



## 6. Now click on table output and then preview data

**Execution Results**

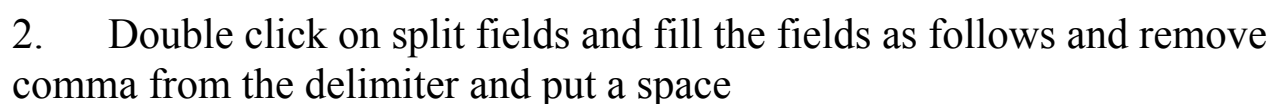
Logging Execution History Step Metrics Performance Graph Metrics Preview data

☒ First rows ☐ Last rows ☐ Off

#	FNAME	LNAME	PERCENTAGE	AGE	FullName
1	Manoj	Singh	35.0	20.0	Manoj Singh
2	Meenal	Shah	42.0	21.0	Meenal Shah
3	Ajay	Mane	53.0	22.0	Ajay Mane
4	Monica	kamble	62.0	21.0	Monica kamble
5	Manoj	Singh	71.0	20.0	Manoj Singh
6	Manoj	Singh	83.0	20.0	Manoj Singh
7	Manoj	Singh	92.0	20.0	Manoj Singh
8	Manoj	Singh	35.0	20.0	Manoj Singh
9	Meenal	Shah	42.0	21.0	Meenal Shah
1..	Ajay	Mane	53.0	22.0	Ajay Mane
1..	Monica	kamble	62.0	21.0	Monica kamble
1..	Manoj	Singh	71.0	20.0	Manoj Singh
1..	Manoj	Singh	83.0	20.0	Manoj Singh
1..	Manoj	Singh	92.0	20.0	Manoj Singh



1. Add a split rows in between concat fields and table output from the above practical



**Split fields**

Step name:

Field to split:

Delimiter:

Enclosure:

**Fields**

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group	Decimal	Currency	Nullif	Default	Trim type
1	FN		N	String									
2	LN		N	String									

3. Now double click on table output and click on get fields and click on SQL->execute

Table output

Step name: Table output

Connection: Split

Target schema:

Target table: stu

Commit size: 1000

Truncate table: ☐

Ignore insert errors: ☐

Specify database fields: ☒

Main options | Database fields

Fields to insert:

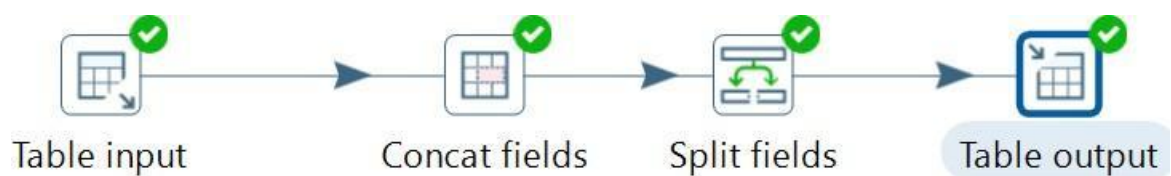
#	Table field	Stream field
1	FNAME	FNAME
2	LNAME	LNAME
3	PERCENTA...	PERCENTAGE
4	AGE	AGE
5	FN	FN
6	LN	LN

Get fields

Enter field mapping

Help OK Cancel SQL

4. Now run the transformation



5. Click on table output and click on preview data

### Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

First rows

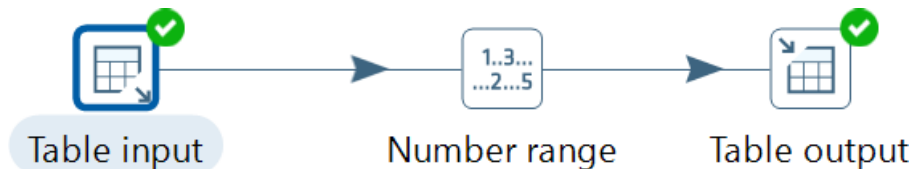
Last rows

Off

#	FNAME	LNAME	PERCENTAGE	AGE	FN	LN	
1	Manoj	Singh	35.0	20.0	Manoj	Singh	
2	Meenal	Shah	42.0	21.0	Meenal	Shah	
3	Ajay	Mane	53.0	22.0	Ajay	Mane	
4	Monica	kamble	62.0	21.0	Monica	kamble	
5	Manoj	Singh	71.0	20.0	Manoj	Singh	
6	Manoj	Singh	83.0	20.0	Manoj	Singh	
7	Manoj	Singh	92.0	20.0	Manoj	Singh	
8	Manoj	Singh	35.0	20.0	Manoj	Singh	
9	Meenal	Shah	42.0	21.0	Meenal	Shah	

## Aim: Implementation of ETL transformation with Pentaho like Number Range

1. Add number range between table output and table input and make connection between them as shown



2. Perform the steps of the input table as done in above practical
3. Double click on Number Range and fill the fields as given below and then click ok

The screenshot shows the 'Number range' configuration window. It contains the following fields and options:

- Step name: Number range
- Input field: PERCENTAGE (dropdown menu)
- Output field: range
- Default value(if no range): unknown

Below these fields is a table titled 'Ranges (min <= x < max):'.

#	Lower Bound	Upper Bound	Value
4	61	70	first
5	71	80	distinction
6	81	100	outstanding

At the bottom of the window are buttons for 'Help', 'OK', and 'Cancel'.

4. Now double click on table output and type the target able and click on get fields.

Then go to SQL->execute->ok->close

**Table output**

Step name:

Connection:

Target schema:

Target table:

Commit size:

Truncate table: ☐

Ignore insert errors: ☐

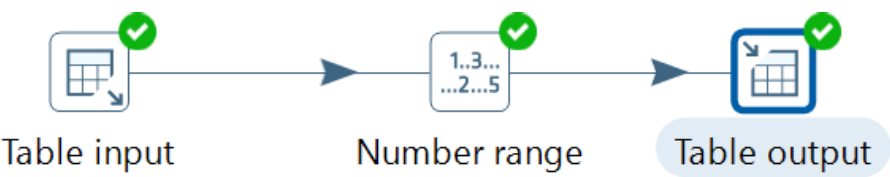
Specify database fields: ☒

**Main options** | **Database fields**

Fields to insert:

#	Table field	Stream field
1	FNAME	FNAME
2	LNAME	LNAME
3	PERCENTA...	PERCENTAGE
4	AGE	AGE
5	range	range

5. Now run the transformation



6. Click on table output and then preview data

Each student is assigned a certain range based on their percentage

### Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

First rows

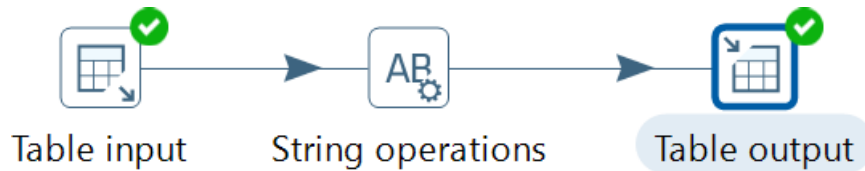
Last rows

Off

#	FNAME	LNAME	PERCENTAGE	AGE	range	
1	Manoj	Singh	35.0	20.0	fail	
2	Meenal	Shah	42.0	21.0	third	
3	Ajay	Mane	53.0	22.0	second	
4	Monica	kamble	62.0	21.0	first	
5	Manoj	Singh	71.0	20.0	distinction	
6	Manoj	Singh	83.0	20.0	outstanding	
7	Manoj	Singh	92.0	20.0	outstanding	
8	Manoj	Singh	35.0	20.0	fail	
9	Meenal	Shah	42.0	21.0	third	

## Aim: Implementation of ETL transformation with Pentaho like String operations.

1. Add string operations in between table input and table output.



2. Perform the steps in the table input as done before.

3. Double click on string operations and fill the fields as given and then click ok.

#	In stream field	Out stream field	Trim type	Lower/Upper	Padding	Pad char	Pad Length	InitCap	Escape	Digits	Remove Special character
1	FNAME			upper							
2	LNAME			lower							

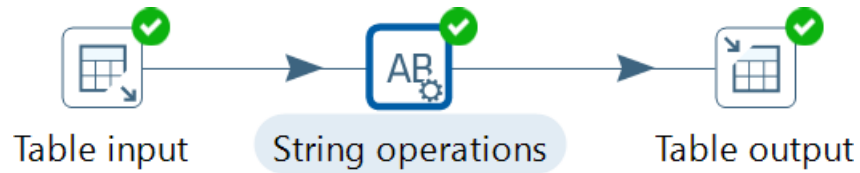
Buttons: Help, OK, Get fields, Cancel

4. Go to table output and then click on get fields and then SQL->execute->ok

#	Table field	Stream field
1	FNAME	FNAME
2	LNAME	LNAME
3	PERCENTAGE	PERCENTAGE
4	AGE	AGE

Buttons: Help, OK, Cancel, SQL

5. Now run the transformation.



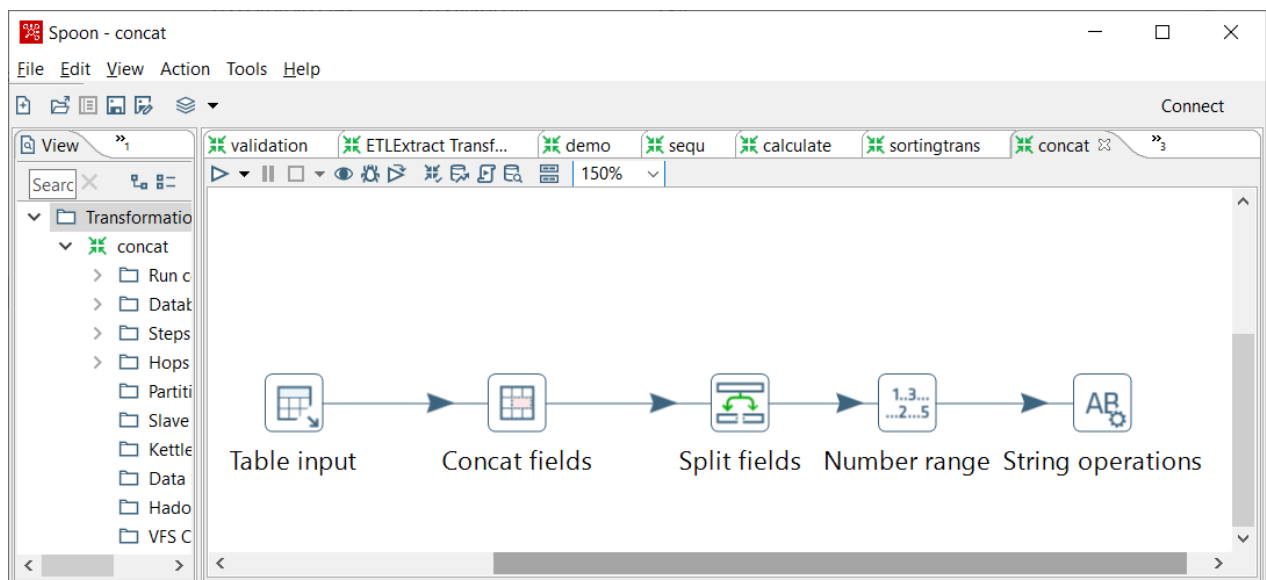
## Execution Results

Logging	Execution History	Step Metrics	Performance Graph	Metrics	Preview data
<input checked="" type="radio"/> First rows <input type="radio"/> Last rows <input type="radio"/> Off					
#	FNAME	LNAME	PERCENTAGE	AGE	
1	MANOJ	singh	35.0	20.0	
2	MEENAL	shah	42.0	21.0	
3	AJAY	mane	53.0	22.0	
4	MONICA	kamble	62.0	21.0	
5	MANOJ	singh	71.0	20.0	
6	MANOJ	singh	83.0	20.0	
7	MANOJ	singh	92.0	20.0	
8	MANOJ	singh	35.0	20.0	
9	MEENAL	shah	42.0	21.0	



## Aim: Implementation of ETL transformation with Pentaho using multiple transformations.

1. Drag and drop a table input, concat fields, split fields, number range and string operations and make connection between them as shown.



2. Double click on table input and perform the steps as done in above practicals.

The screenshot shows the 'Database Connection' dialog box. The 'General' tab is selected. The 'Connection name' is 'Split'. The 'Connection type' is 'Oracle'. The 'Access' is 'Native (JDBC)'. The 'Settings' section includes: 'Host Name' (localhost), 'Database Name' (orcl), 'Tablespace for Data' (empty), 'Tablespace for Indices' (empty), 'Port Number' (1521), 'Username' (system), and 'Password' (masked with dots). At the bottom, there are buttons for 'Test', 'Feature List', 'Explore', 'OK', and 'Cancel'.

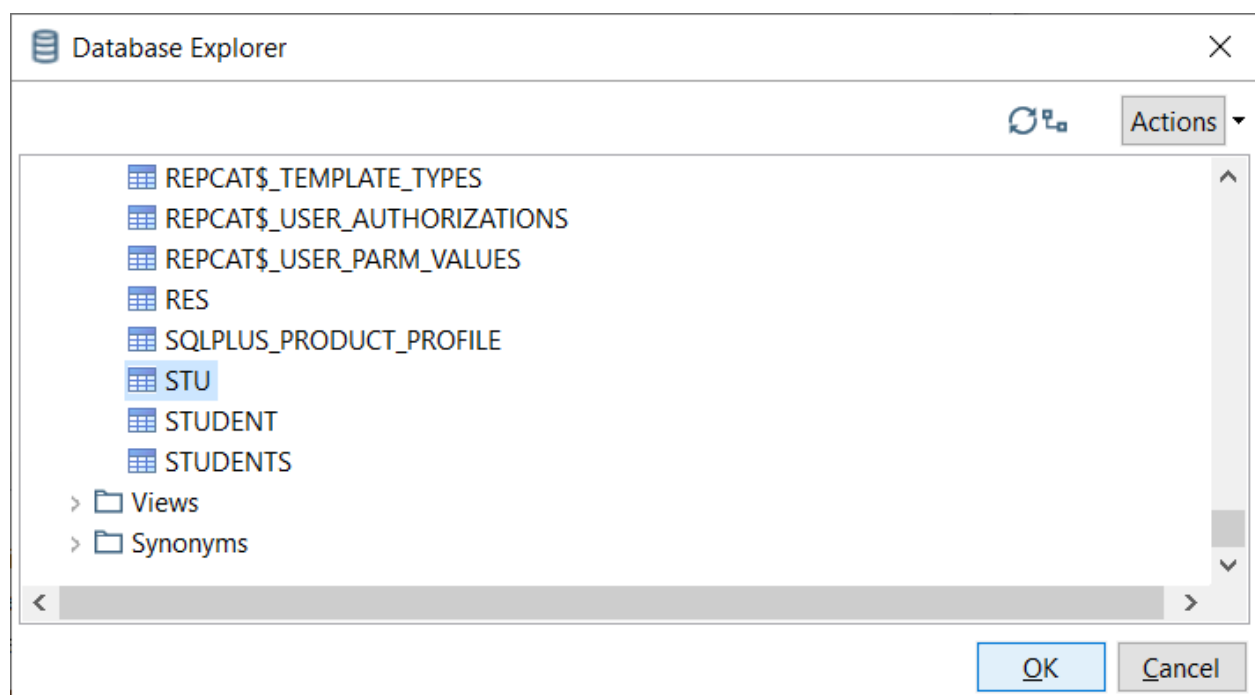
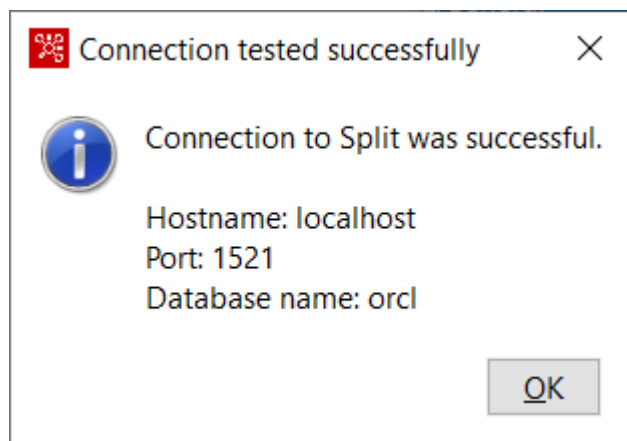


Table input

Step name

Table input

Connection

Split

Edit...

New...

Wizard...

SQL

Get SQL select statement...

```

SELECT
  FNAME
  LNAME
  PERCENTAGE
  AGE
FROM SYSTEM.STU

```

Line 1 Column 0

Store column info in step meta

Enable lazy conversion

Replace variables in script?

Insert data from step

Execute for each row?

Limit size

0

Help

OK

Preview

Cancel

Rows of step: Table input (7 rows)

#	FNAME	LNAME	PERCENTAGE	AGE
1	Manoj	Singh	35.0	20.0
2	Meenal	Shah	42.0	21.0
3	Ajay	Mane	53.0	22.0
4	Monica	kamble	62.0	21.0
5	Manoj	Singh	71.0	20.0
6	Manoj	Singh	83.0	20.0
7	Manoj	Singh	92.0	20.0

3. Now double click on concat fields and fill the fields as given. Remove the semicolon from the separator and put space and click ok.

[illegible]

4. Double click on split fields and fill the fields as follows and remove comma from the delimiter and put a space

**Split fields**

Step name:

Field to split:

Delimiter:

Enclosure:

**Fields**

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group	Decimal	Currency	Nullif	Default	Trim type
1	FN		N	String									
2	LN		N	String									

5. Double click on Number Range and fill the fields as given below and then click ok.

**Number range**

Step name:

Input field:

Output field:

Default value(if no range)

Ranges (min <= x < max):

#	Lower Bound	Upper Bound	Value
4	61	70	first
5	71	80	distinction
6	81	100	outstanding

6. Double click on string operations and fill the fields as given and then click ok.

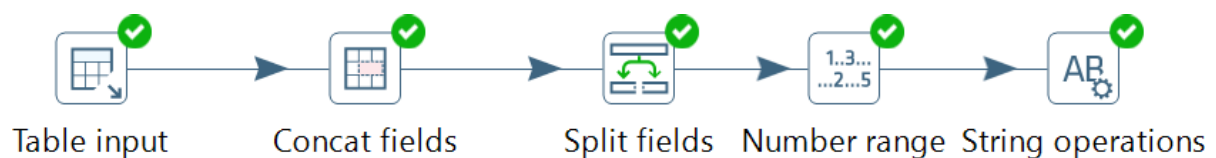
**String operations**

Step name:

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			upper							
2	LNAME			lower							

7. Now run the transformation



8.Now click on concat fields and then preview data.

#### Execution Results

Logging	Execution History	Step Metrics	Performance Graph	Metrics	Preview data
<input checked="" type="radio"/> First rows	<input type="radio"/> Last rows	<input type="radio"/> Off			
#	FNAME	LNAME	PERCENTAGE	AGE	FullName
1	Manoj	Singh	35.0	20.0	Manoj Singh
2	Meenal	Shah	42.0	21.0	Meenal Shah
3	Ajay	Mane	53.0	22.0	Ajay Mane
4	Monica	kamble	62.0	21.0	Monica kamble
5	Manoj	Singh	71.0	20.0	Manoj Singh
6	Manoj	Singh	83.0	20.0	Manoj Singh
7	Manoj	Singh	92.0	20.0	Manoj Singh
8	Manoj	Singh	35.0	20.0	Manoj Singh
9	Meenal	Shah	42.0	21.0	Meenal Shah
1..	Ajay	Mane	53.0	22.0	Ajay Mane
1..	Monica	kamble	62.0	21.0	Monica kamble
1..	Manoj	Singh	71.0	20.0	Manoj Singh
1..	Manoj	Singh	83.0	20.0	Manoj Singh
1..	Manoj	Singh	92.0	20.0	Manoj Singh

9.Now click on split fields and then preview data.

#### Execution Results

Logging

Execution History

Step Metrics

Performance Graph

Metrics

Preview data

☒ First rows
 ☐ Last rows
 ☐ Off

#	FNAME	LNAME	PERCENTAGE	AGE	FN	LN
1	Manoj	Singh	35.0	20.0	Manoj	Singh
2	Meenal	Shah	42.0	21.0	Meenal	Shah
3	Ajay	Mane	53.0	22.0	Ajay	Mane
4	Monica	kamble	62.0	21.0	Monica	kamble
5	Manoj	Singh	71.0	20.0	Manoj	Singh
6	Manoj	Singh	83.0	20.0	Manoj	Singh
7	Manoj	Singh	92.0	20.0	Manoj	Singh
8	Manoj	Singh	35.0	20.0	Manoj	Singh
9	Meenal	Shah	42.0	21.0	Meenal	Shah

10. Click on number range and then preview data

Each student is assigned a certain range based on their percentage

### Execution Results

Logging	Execution History	Step Metrics	Performance Graph	Metrics	Preview data
<input checked="" type="radio"/> First rows <input type="radio"/> Last rows <input type="radio"/> Off					
#	FNAME	LNAME	PERCENTAGE	AGE	range
1	Manoj	Singh	35.0	20.0	fail
2	Meenal	Shah	42.0	21.0	third
3	Ajay	Mane	53.0	22.0	second
4	Monica	kamble	62.0	21.0	first
5	Manoj	Singh	71.0	20.0	distinction
6	Manoj	Singh	83.0	20.0	outstanding
7	Manoj	Singh	92.0	20.0	outstanding
8	Manoj	Singh	35.0	20.0	fail
9	Meenal	Shah	42.0	21.0	third

11. Click on string operations and then preview data

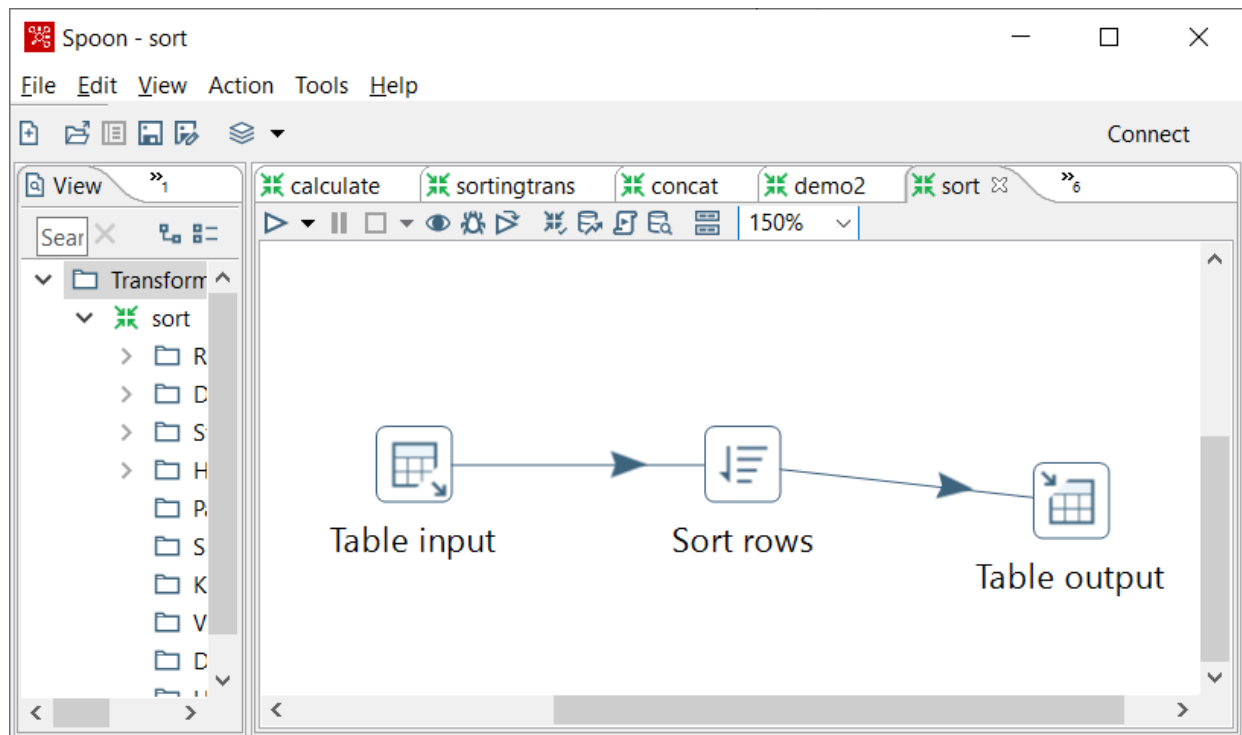
As shown fname is in uppercase and lname is in lowercase.

### Execution Results

Logging	Execution History	Step Metrics	Performance Graph	Metrics	Preview data
<input checked="" type="radio"/> First rows <input type="radio"/> Last rows <input type="radio"/> Off					
#	FNAME	LNAME	PERCENTAGE	AGE	
1	MANOJ	singh	35.0	20.0	
2	MEENAL	shah	42.0	21.0	
3	AJAY	mane	53.0	22.0	
4	MONICA	kamble	62.0	21.0	
5	MANOJ	singh	71.0	20.0	
6	MANOJ	singh	83.0	20.0	
7	MANOJ	singh	92.0	20.0	
8	MANOJ	singh	35.0	20.0	
9	MEENAL	shah	42.0	21.0	

## **Aim: Implementation of ETL transformation with Pentaho using sort rows.**

1. Drag and drop the table input, sort rows and table output. Make connection between them as shown.





2. Double click on table and perform the steps done in previous practicals.

Table input

Step name

Table input

Connection

sor

Edit...

New...

Wizard...

SQL

Get SQL select statement...

```

SELECT
  ACCNO
, CUSTNAME
, BRANCH
, ACCTBAL
FROM SYSTEM.ACCOUNTS
        
```

Line 1 Column 0

Store column info in step meta

☐

Enable lazy conversion

☐

Replace variables in script?

☐

Insert data from step

Execute for each row?

☐

Limit size

0

Help

OK

Preview

Cancel

Rows of step: Table input (6 rows)

#	ACCNO	CUSTNAME	BRANCH	ACCTBAL
1	6.0	Mansi	Thane	80.0
2	4.0	Chutki	Bhuvaneshwar	2000.0
3	5.0	Nancy	Bhuvaneshwar	2000.0
4	1.0	Sugar	Bhuvaneshwar	5000.0
5	2.0	Arman	Fun	9998.0
6	3.0	Sanskar	Airport	7000.0

3. Now double click on sort rows and fill the fields as follows and then click ok.

Sort rows

Step name:

Sort directory:

TMP-file prefix:

Sort size (rows in memory):

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	ACCTBAL	N	N	N	0	N

4. Double click on table output->write the target table->database fields->check specify get fields->SQL->Execute->ok->close.

Table output

Step name:

Connection:

Target schema:

Target table:

Commit size:

Truncate table ☐

Ignore insert errors ☐

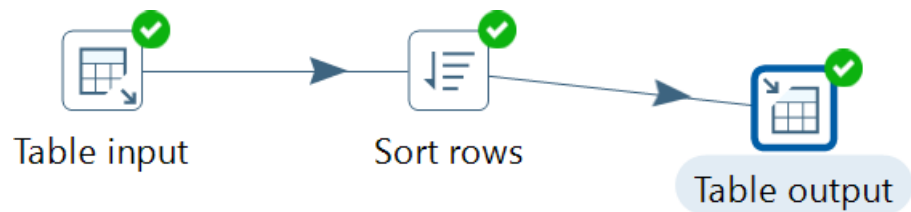
Specify database fields ☒

Main options Database fields

Fields to insert:

#	Table field	Stream field
1	ACCNO	ACCNO
2	CUSTNAME	CUSTNAME
3	BRANCH	BRANCH
4	ACCTBAL	ACCTBAL

5. Now run the transformation



6. Now click on table output and then click on preview data

The account balance column is sorted in descending order by default as sort rows sorts the data in descending order by default.

<input checked="" type="radio"/> First rows <input type="radio"/> Last rows <input type="radio"/> Off				
#	ACCNO	CUSTNAME	BRANCH	ACCTBAL
1	2.0	Arman	Fun	9998.0
2	3.0	Sanskar	Airport	7000.0
3	1.0	Sugar	Bhuvaneshwar	5000.0
4	4.0	Chutki	Bhuvaneshwar	2000.0
5	5.0	Nancy	Bhuvaneshwar	2000.0
6	6.0	Mansi	Thane	80.0

**Aim: Implementation of ETL transformation with Pentaho like CSV File.**

## .CSV file

- 1.From the input select “.CSV INPUT TABLE”
- 2.From output select table output (Drag and drop both)

And make connection between them



3. double click on csv file input ->browse->choose a csv file->get fields->delete the value column->delete select->close->ok

## Delete all rows except the year row.

CSV file input

Step name: CSV file input

Filename: D:\annual-enterprise-survey-2020-financial-year-provisional-csv.csv Browse...

Delimiter: , Insert TAB

Enclosure: "

NIO buffer size: 50000

Lazy conversion? ☒

Header row present? ☒

Add filename to result ☐

The row number field name (optional):

Running in parallel? ☐

New line possible in fields? ☐

Format: mixed

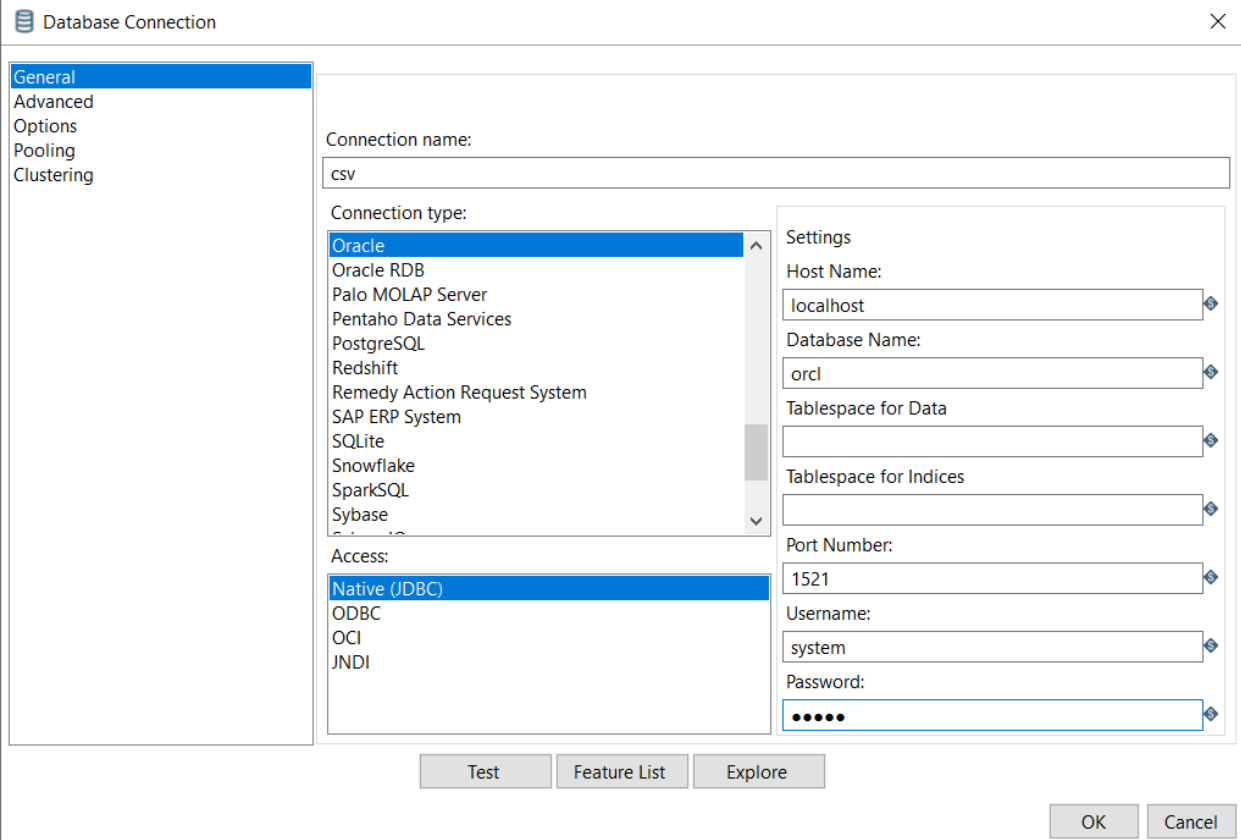
File encoding:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Trim type
1	Year	Integer	#	15	0	\$	.	,	none

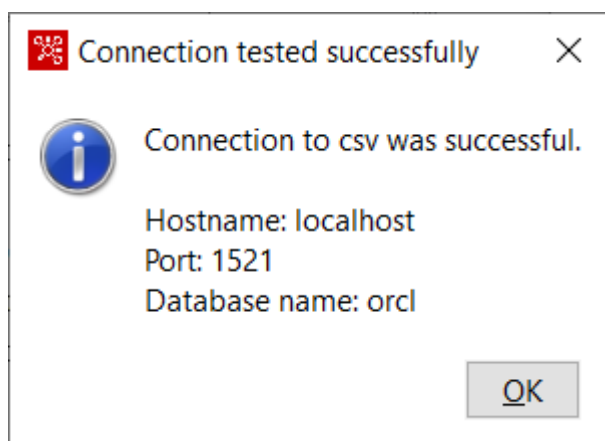
Help
OK
Get Fields
Preview
Cancel

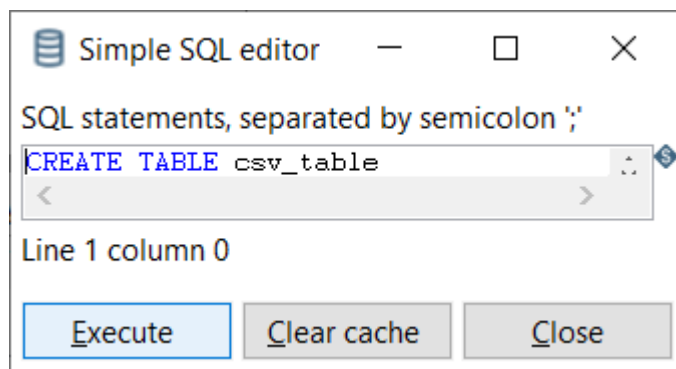
[illegible][illegible]

5. Now create a new connection in table output->type the fields as given->test->ok->SQL->execute->ok->ok->close.



The 'Database Connection' dialog box is shown with the 'General' tab selected. The 'Connection name' field contains 'csv'. The 'Connection type' dropdown menu is open, showing a list of database types with 'Oracle' selected. The 'Access' dropdown menu is also open, showing 'Native (JDBC)' selected. The 'Settings' section on the right contains the following fields: 'Host Name' (localhost), 'Database Name' (orcl), 'Tablespace for Data' (empty), 'Tablespace for Indices' (empty), 'Port Number' (1521), 'Username' (system), and 'Password' (masked with dots). At the bottom, there are buttons for 'Test', 'Feature List', 'Explore', 'OK', and 'Cancel'.





6. Now run the transformation



7. Click on table output and then click on preview data.

### Execution Results

Logging	Execution History	Step Metrics	Performance Graph	Metrics	Preview data
<input checked="" type="radio"/> First rows <input type="radio"/> Last rows <input type="radio"/> Off					
#	Year				
1	2020				
2	2020				
3	2020				
4	2020				
5	2020				
6	2020				
7	2020				
8	2020				
9	2020				
1..	2020				
1..	2020				
1..	2020				
1..	2020				

## Aim: Implementation of ETL transformation with Pentaho like merge join.

1. Take a table input and do the steps as done in previous. include one table in it

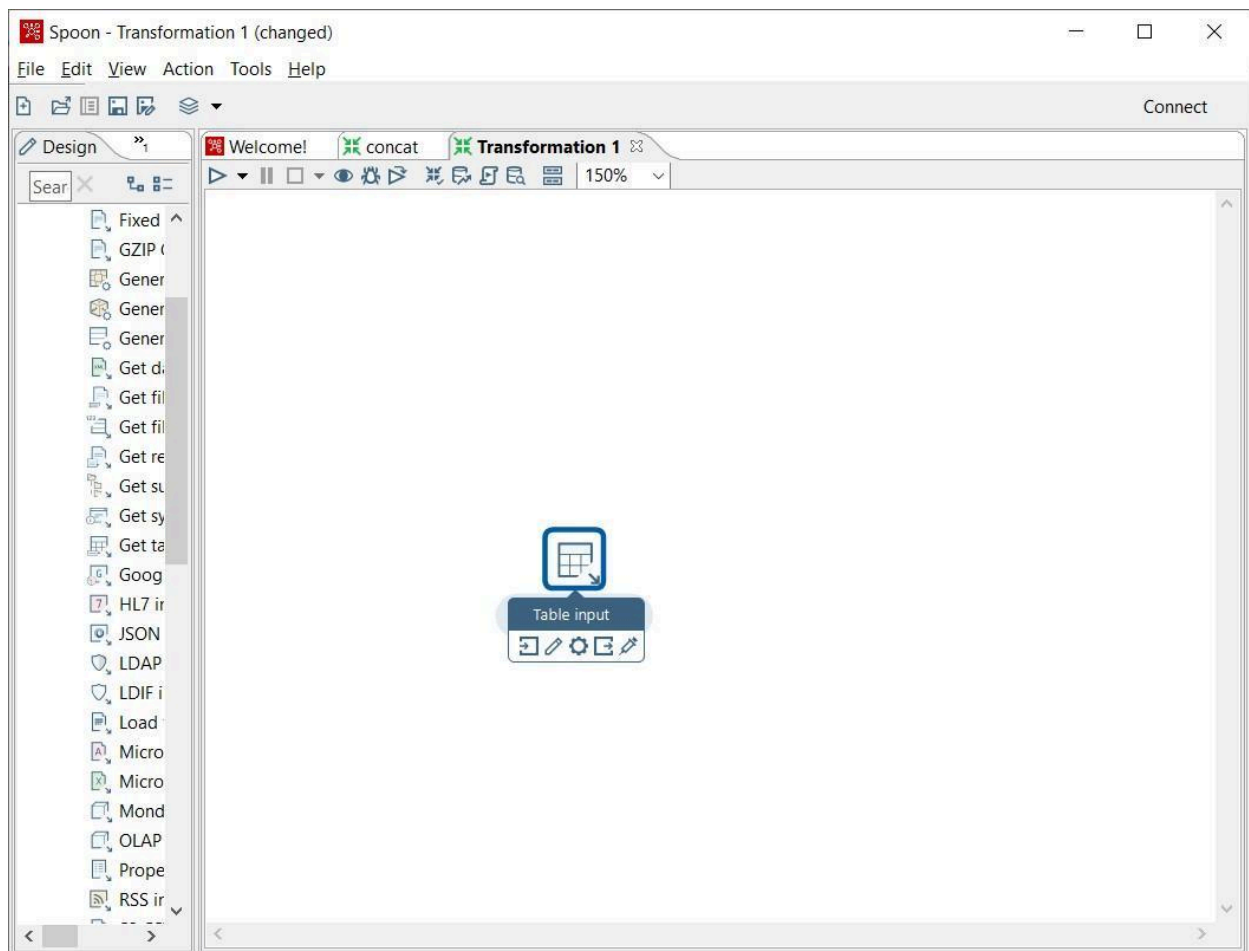




Table input

Step name

Employee

Connection

Edit...

New...

Wizard...

SQL

Get SQL select statement...

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

Line 1 Column 52

Store column info in step meta

☐

Enable lazy conversion

☐

Replace variables in script?

☐

Insert data from step

Execute for each row?

☐

Limit size

0

Help

OK

Preview

Cancel

Database Connection

General

Advanced

Options

Pooling

Clustering

Connection name:

Test

Connection type:

Oracle

Oracle RDB

Palo MOLAP Server

Pentaho Data Services

PostgreSQL

Redshift

Remedy Action Request System

SAP ERP System

SQLite

Snowflake

SparkSQL

Sybase

...

Access:

Native (JDBC)

ODBC

OCI

JNDI

Settings

Host Name:

localhost

Database Name:

orcl

Tablespace for Data

Tablespace for Indices

Port Number:

1521

Username:

system

Password:

.....

Test

Feature List

Explore

OK

Cancel

Connection tested successfully

i

Connection to Test was successful.  
Hostname: localhost  
Port: 1521  
Database name: orcl

OK

Table input

Step name

Employee

Connection

Test

Edit...

New...

Wizard...

SQL

Get SQL select statement...

```

SELECT
EMPID
EMPNO
DEPTNO
SALARY
FROM SYSTEM.EMPLOYEE

```

Line 1 Column 0

Store column info in step meta

Enable lazy conversion

Replace variables in script?

Insert data from step

Execute for each row?

Limit size

0

Help

OK

Preview

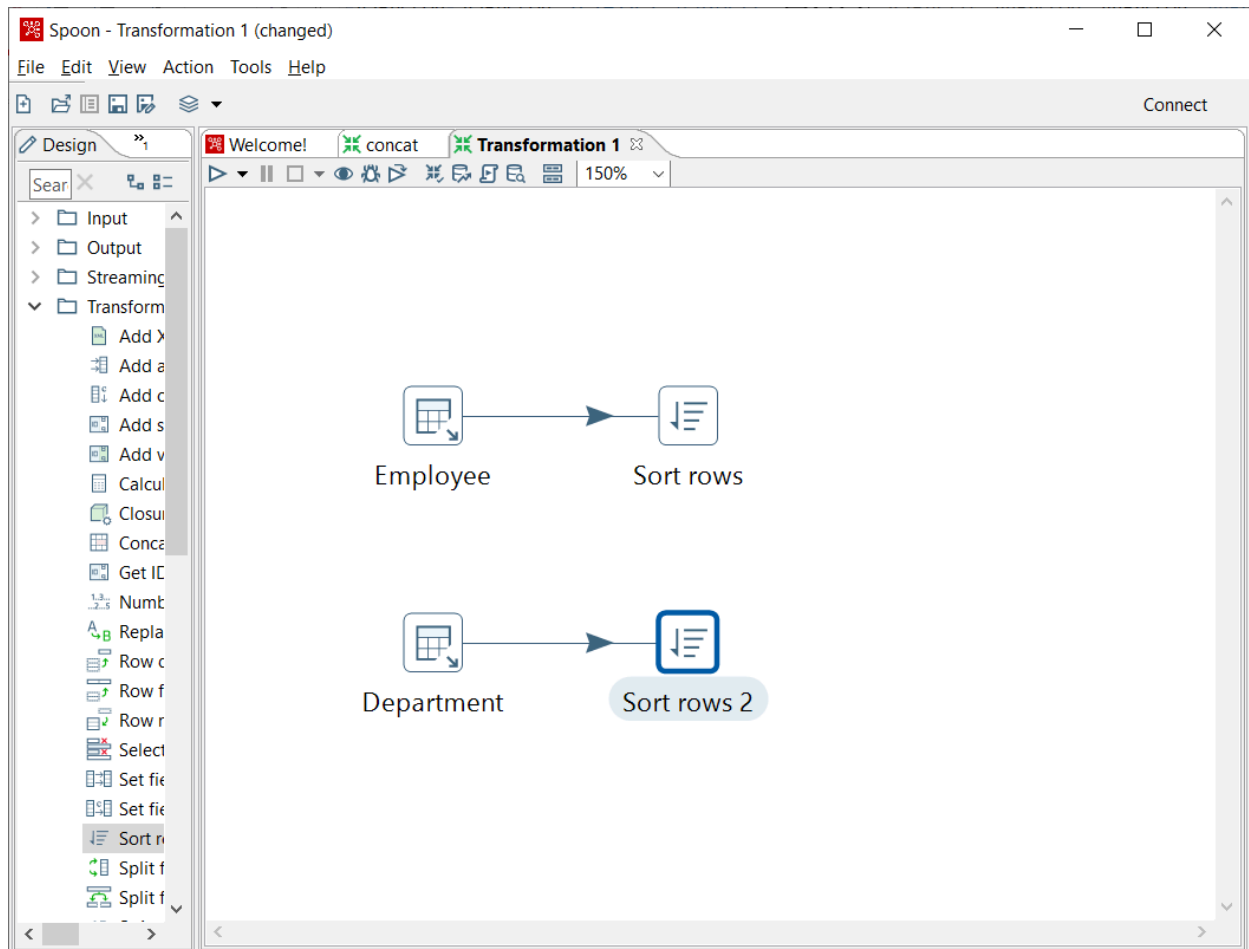
Cancel

## Examine preview data

Rows of step: Employee (7 rows)

#	EMPID	EMPNO	DEPNO	SALARY
1	1000	Ajay Singh	PD	15000
2	1001	Bharti Shetty	QC	25000
3	1002	Sheetal Roy	PD	35000
4	1003	Meena Pawar	Support	30000
5	1004	Rajlaxmi Pal	Support	4000
6	1005	Vaibhav Vaidya	QC	5000
7	1006	Nancy	QC	5000

2. Take another table input and connect both Statement->test->tables of them to different sort rows



3. Double click on another table input->Get SQL->test->tables->department->ok->yes->preview->close->ok

The 'Table input' dialog box is shown with the following details:

- Step name:** Department
- Connection:** Test
- Buttons:** Edit..., New..., Wizard...
- SQL:** A text area containing the SQL query:

```
SELECT
  DEPTNO
, NO_OF_EMPLOYEES
FROM SYSTEM.DEPARTMENT
```
- Get SQL select statement...:** A button next to the SQL text area.
- Line 1 Column 0:** Metadata options including:
  - Store column info in step meta ☐
  - Enable lazy conversion ☐
  - Replace variables in script? ☐
  - Insert data from step
  - Execute for each row? ☐
  - Limit size
- Buttons:** Help, OK, Preview, Cancel

The 'Database Explorer' window shows a list of database objects:

- DEF\$\_PUSHED\_TRANSACTIONS
- DEF\$\_TEMP\$LOB
- DEPARTMENT
- EMP
- EMPLOYEE
- HELP
- LOGMNRC\_DBNAME\_UID\_MAP
- LOGMNRC\_GSII
- LOGMNRC\_GTCS
- LOGMNRC\_GTLO
- LOGMNRC\_GTAS PART MAP

Buttons: OK, Cancel



Examine preview data

Rows of step: Department (3 rows)

#	DEPTNO	NO_OF_EMPLOYEES
1	PD	200.0
2	QC	300.0
3	Support	300.0

//Sorting for employee

[Note :-In Employee table and department table key “dept no” is common, so we can perform join]

```
SQL> desc employee;
Name                                     Null?      Type
-----
EMPID                                   VARCHAR2(10)
EMPNO                                   VARCHAR2(20)
DEPTNO                                  VARCHAR2(10)
SALARY                                  NUMBER(7)

SQL> desc department;
Name                                     Null?      Type
-----
DEPTNO                                  VARCHAR2(10)
NO_OF_EMPLOYEES                         NUMBER
```

4. Double click on sort connected to employee table input->fill the fields as given below->ok

Sort rows

Step name

Emp\_sort

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	DEPTNO	Y	N	N	0	N

Help

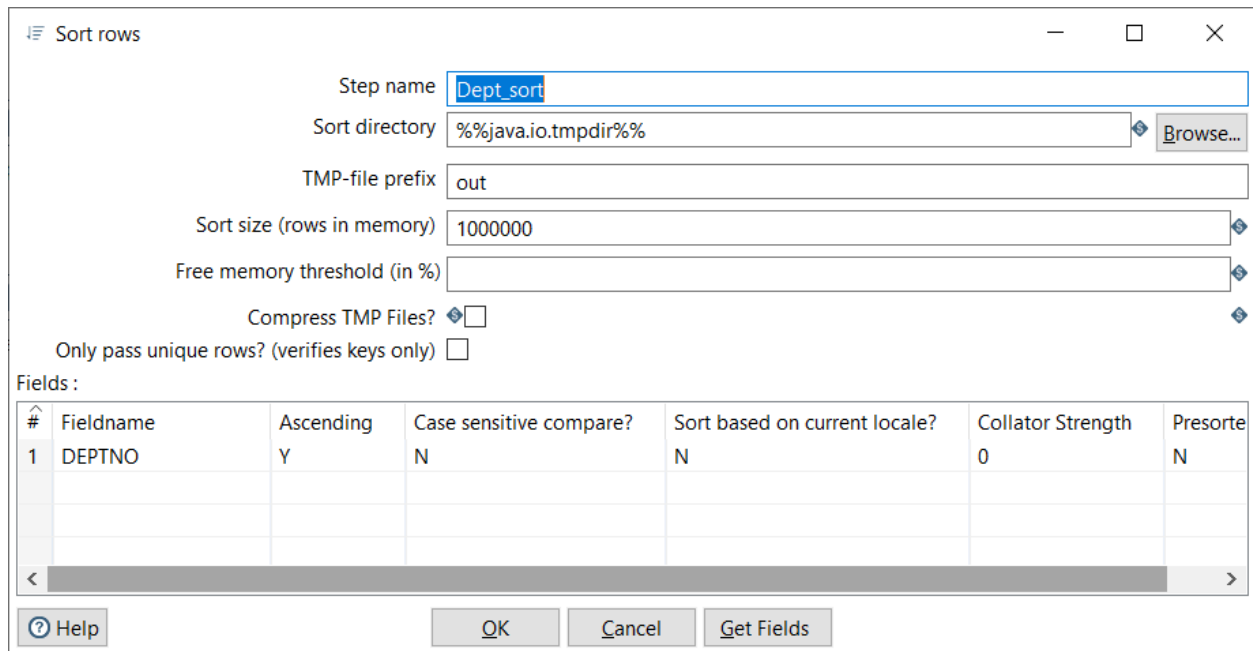
OK

Cancel

Get Fields

//Sorting for department

5. Double click on sort connected to department table input->fill the fields as given below->ok



The 'Sort rows' dialog box is shown with the following fields and values:

- Step name: Dept\_sort
- 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) ☐

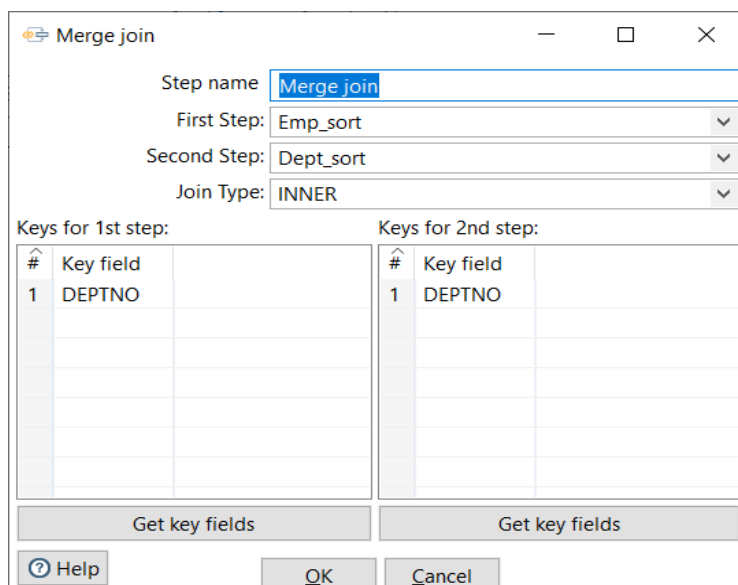
Fields:

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

Buttons: Help, OK, Cancel, Get Fields

//To perform Join

6. Drag and drop merge join from transform->Make connection from both the sorts to merge join->Double click on merge join and fill the fields as given below->ok->close



The 'Merge join' dialog box is shown with the following fields and values:

- Step name: Merge join
- First Step: Emp\_sort
- Second Step: Dept\_sort
- Join Type: INNER

Keys for 1st step:

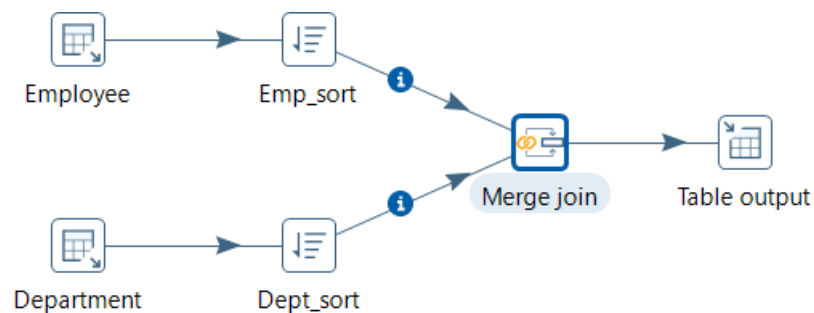
#	Key field
1	DEPTNO

Keys for 2nd step:

#	Key field
1	DEPTNO

Buttons: Get key fields, Get key fields, Help, OK, Cancel

7. Take a table output and make connection from merge join to table output



//To put it in output table

8. Double click on table output->target table:-merge->specify database fields->get fields->SQL->Execute->ok->close->ok.

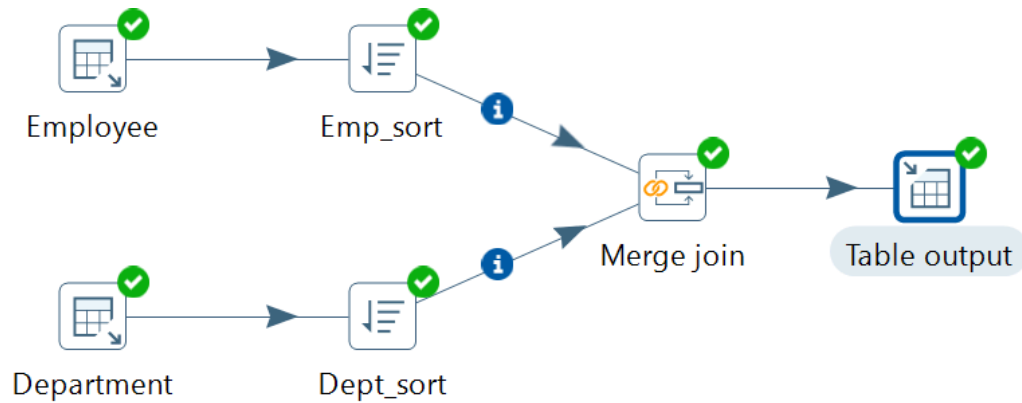
The screenshot shows the 'Table output' configuration window. The 'Step name' is 'Table output', the 'Connection' is 'Test', and the 'Target table' is 'merge'. The 'Commit size' is set to 1000. The 'Specify database fields' checkbox is checked. The 'Database fields' tab is active, showing a table with 6 rows of field mappings. The 'Get fields' button is visible, and the 'Enter field mapping' button is also present.

#	Table field	Stream field
1	EMPID	EMPID
2	EMPNO	EMPNO
3	DEPTNO	DEPTNO
4	SALARY	SALARY
5	DEPTNO_1	DEPTNO_1
6	NO_OF_EM...	NO_OF_EMPL...



9. Now run the transformation

Go to table output->preview data.



Execution Results						
Logging Execution History Step Metrics Performance Graph Metrics Preview data						
First rows Last rows Off						
#	EMPID	EMPNO	DEPTNO	SALARY	DEPTNO_1	NO_OF_EMPLOYEES
1	1000	Ajay singh	PD	15000	PD	200.0
2	1002	sheetal Roy	PD	35000	PD	200.0
3	1001	Bharti Shetty	QC	25000	QC	300.0
4	1003	Meena Powar	Support	30000	Support	300.0

10. Now in SQL plus type

> select \* from merge;

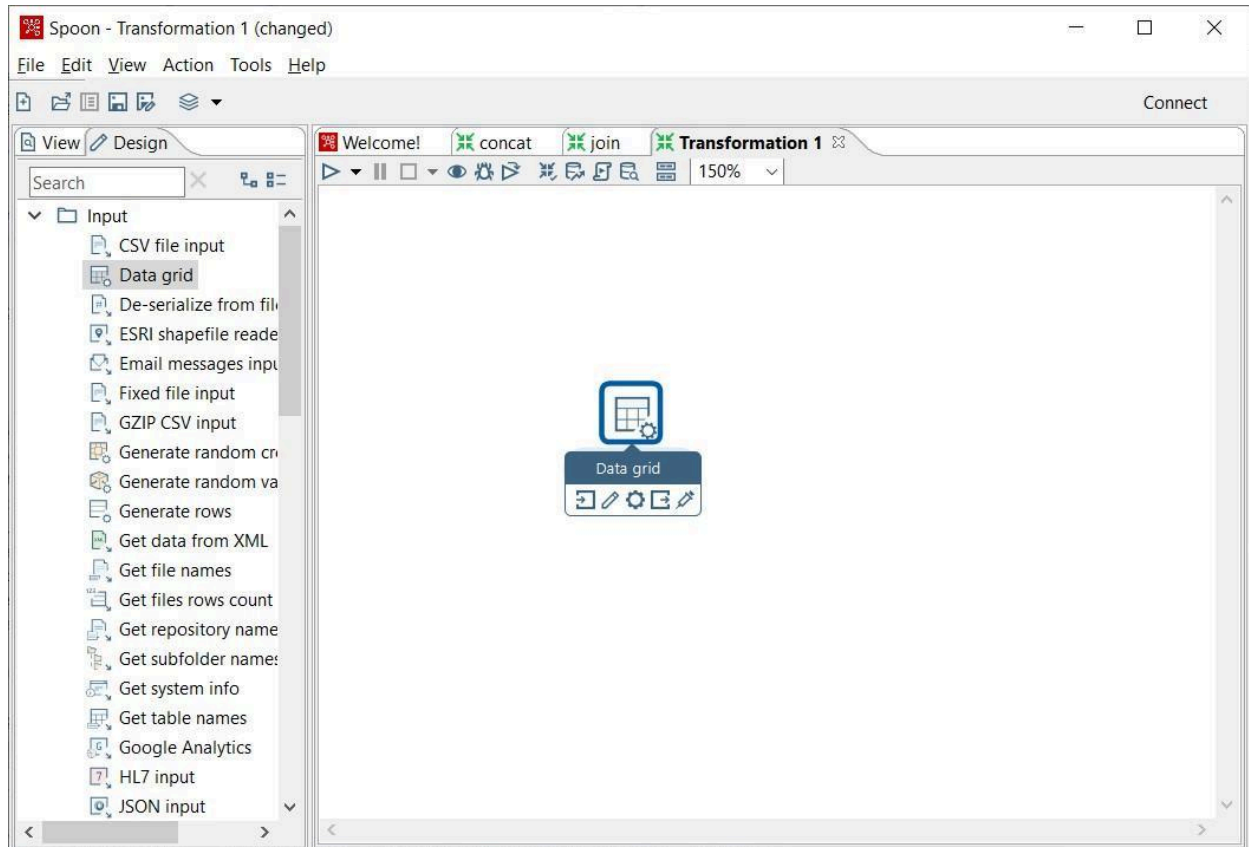
Here both the tables have merged.

```
SQL> select*from merge;
```

EMPID	EMPNO	DEPTNO	SALARY	DEPTNO_1	NO_OF_EMPLOYEES
1000	Ajay singh	PD	15000	PD	200
1002	sheetal Roy	PD	35000	PD	200
1001	Bharti Shetty	QC	25000	QC	300
1003	Meena Powar	Support	30000	Support	300

## Aim: Implementation of ETL transformation with Pentaho using validation.

1. Drag and drop a data grid from input.



2. Double click on the data grid and fill the fields in the meta part as given below.

The screenshot shows the 'Data grid' configuration dialog box. The 'Step name' is 'product'. The 'Meta' tab is selected, showing a table with the following fields:

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Set empty string?
1	prod_name	String							
2	prod_qty	Integer							
3	check_status	String							
4									

Buttons at the bottom: Help, OK, Preview, Cancel.

3. Then click on the data part and fill the fields as given below.

Data grid

Step name

product

Meta

Data

#	prod_name	prod_qty	check_status
1	Soap	20	shifted
2	Medicine	100	cancelled
3	Stationary	30	shifted
4			
5			

?

Help


OK

Preview

Cancel

[Now we will perform validation on check\_status column. So here we just want to display records whose check\_status is shifted.]

4.preview->ok->close->ok.

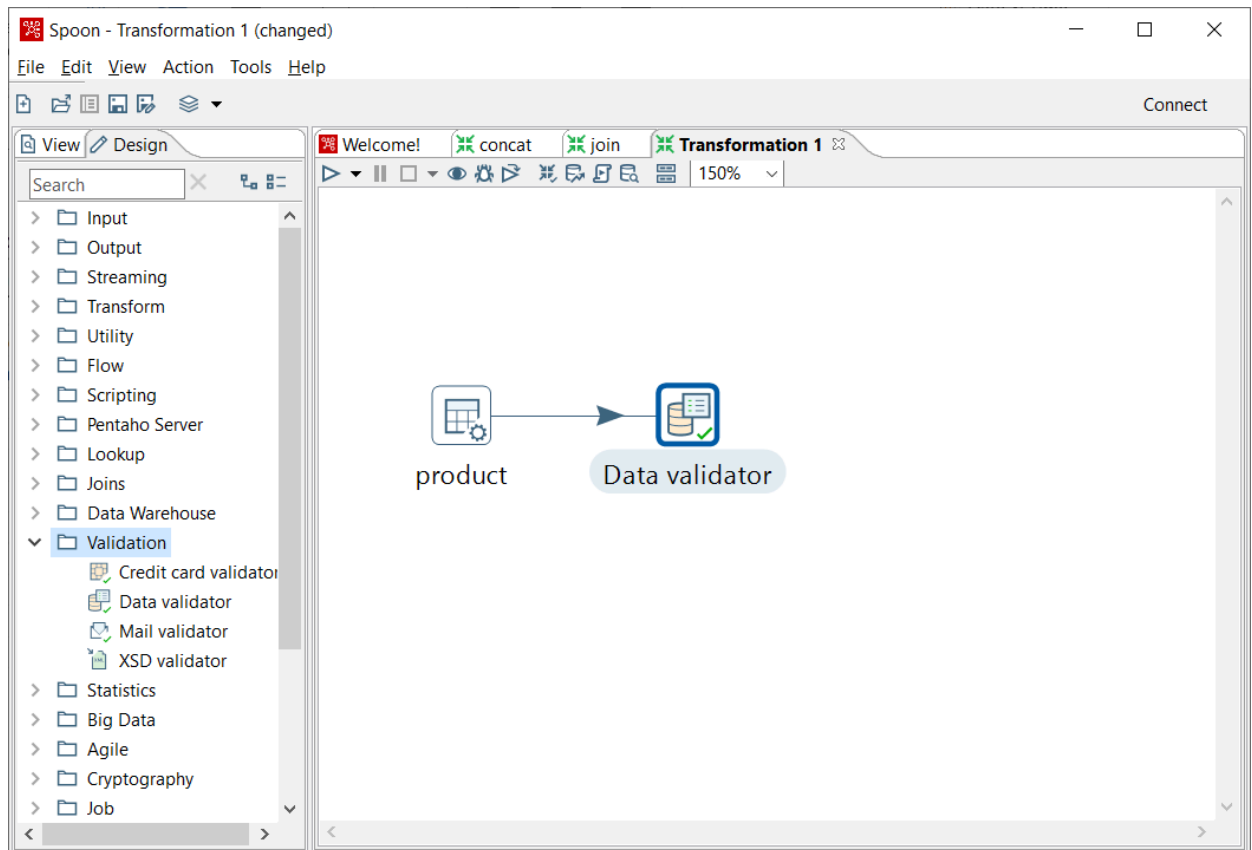
 Examine preview data

Rows of step: product (3 rows)

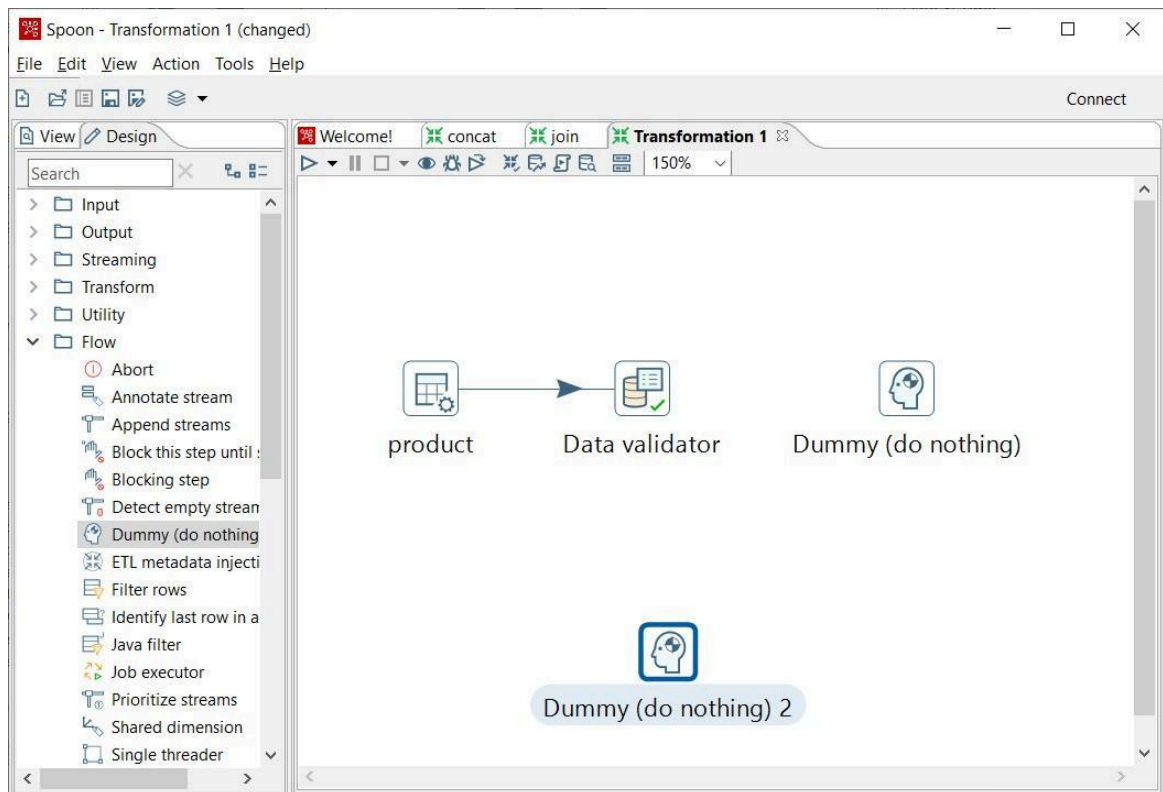
#	prod_name	prod_qty	check_status
1	Soap	20	shifted
2	Medicine	100	cancelled
3	Stationary	30	shifted

//To perform validation

1. Drag and drop a data validator from validation and make connection between the data rid and the data validator.



2. Drag and drop two dummy(do nothing) from flow.

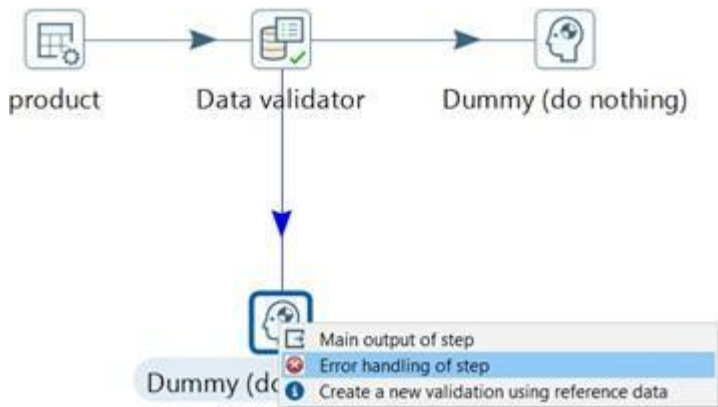


3. Make one connection from data validator to one dummy(do nothing). Click on the "main output of the step".

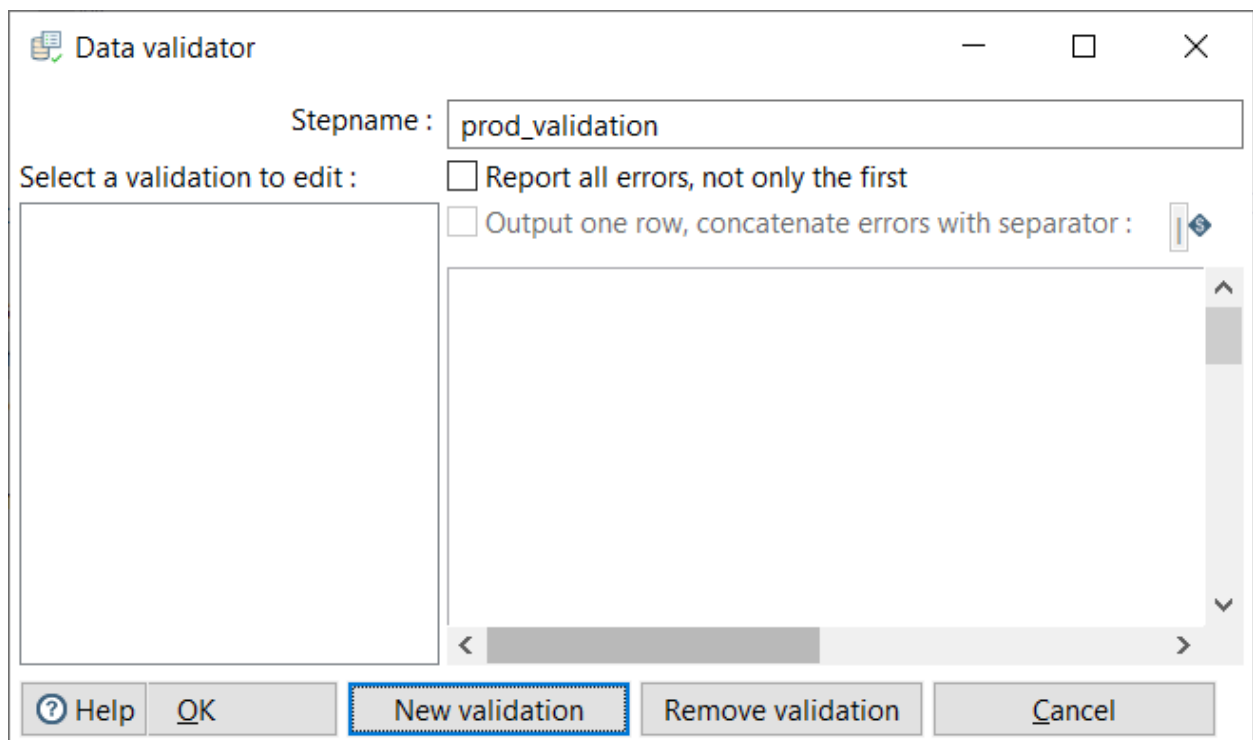



4. Make the second connection from data validator to the second dummy(do nothing).

Click on “Error handling of the step”.



5. Now double click on data validator->step name:-prod\_validation\_new validation->Give any name eg:-shift\_validation->ok.




Enter validation name

×

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

shift\_validation

OK

Cancel

6.shift\_validation->Name of field to validate:-check\_status, data type:-string->add->shifted->ok.

Data validator

— □ ×

Stepname : prod\_validation

Select a validation to edit :

shift\_validation

☐ Report all errors, not only the first  
☐ Output one row, concatenate errors with separator : |

Validation description

shift\_validation

Name of field to validate

check\_status

Error code

Error description

Verify data type?

☐

Data type

String

Conversion mask

Decimal Symbol

Grouping Symbol

Null allowed?

☒

Only null values allowed?

☐

Only numeric data expected

☐

Max string length

Min string length

Maximum value

Minimum value

Expected start string

Expected end string

Not allowed start string


Help

OK

New validation

Remove validation

Cancel


Add allowed value

×

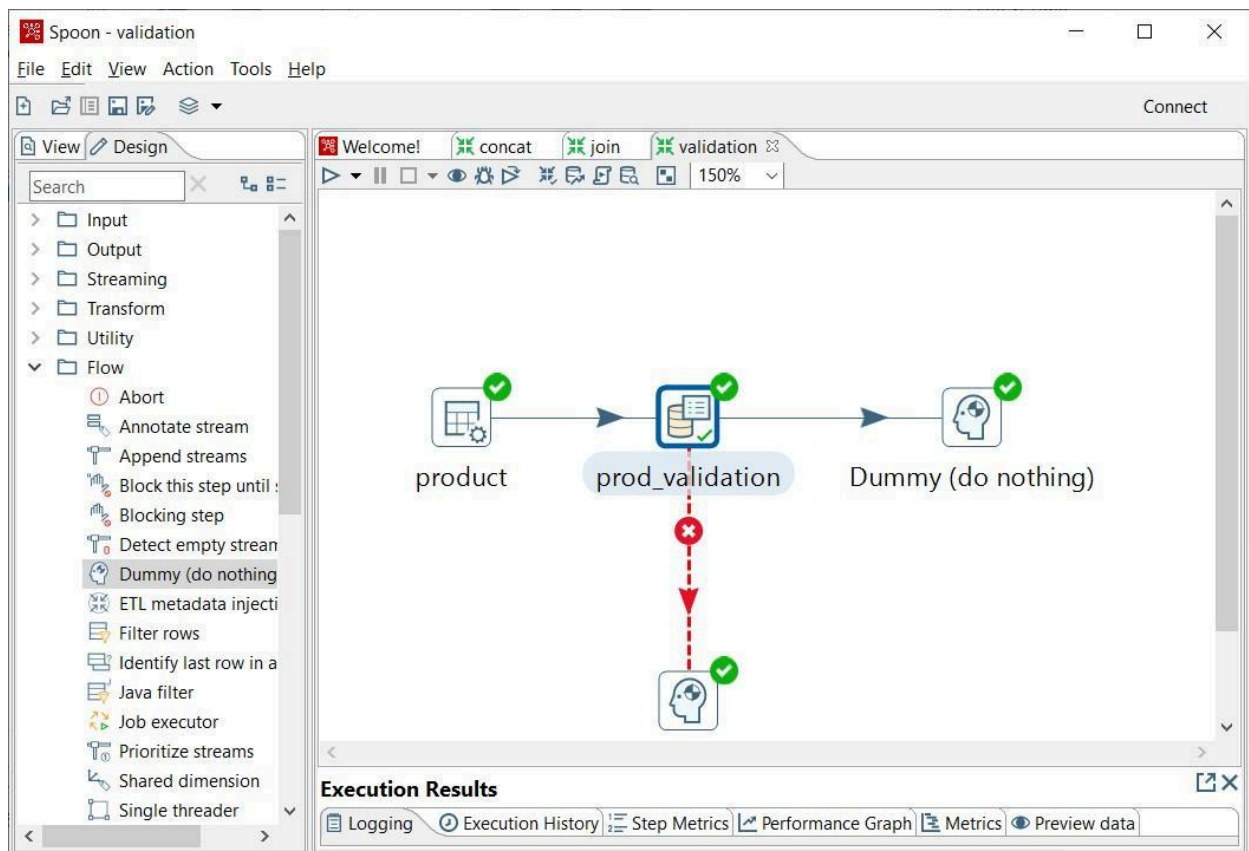
Enter the allowed value to add:

shifted

OK

Cancel

7. Now run the transformation.



8. Click on preview data.

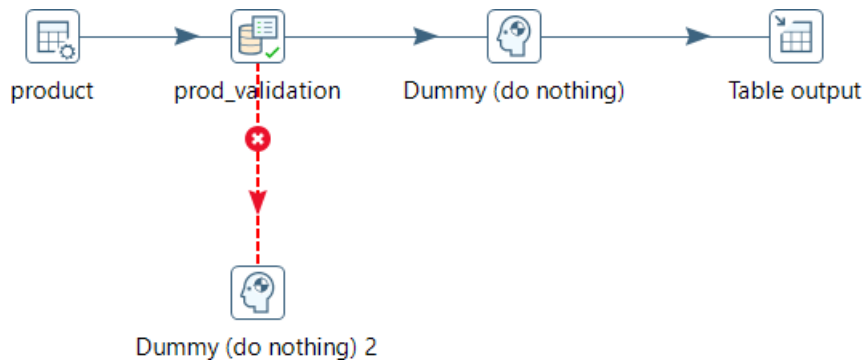
Here as we can see only check status of shifted is showing.

[illegible]



9. Similarly we can store the result in the output table.

Drag and drop a table output and connect it to the dummy(do nothing) of the main output.



10. Double click on table output->target table:-  
prod\_validation->specify database fields->database fields->get  
fields->SQL->Execute->ok->close.

Table output configuration window:

- Step name: Table output
- Connection: product
- Target schema:
- Target table: prod\_validation
- Commit size: 1000
- Truncate table: ☐
- Ignore insert errors: ☐
- Specify database fields: ☒

Simple SQL editor dialog:

SQL statements, separated by semicolon ';' :

```
CREATE TABLE prod_validation
```

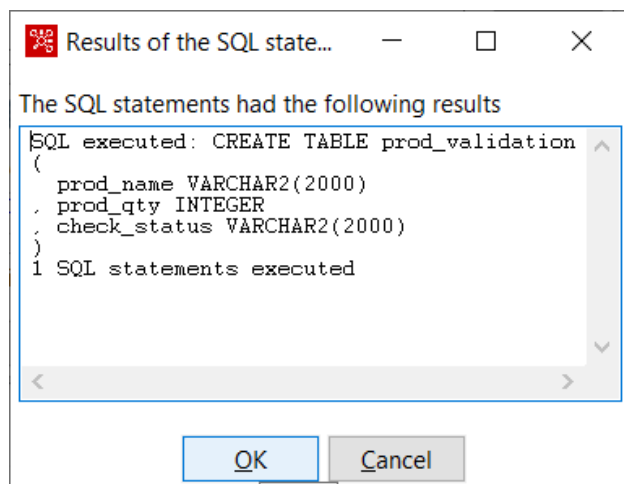
Line 1 column 0

Buttons: Execute, Clear cache, Close

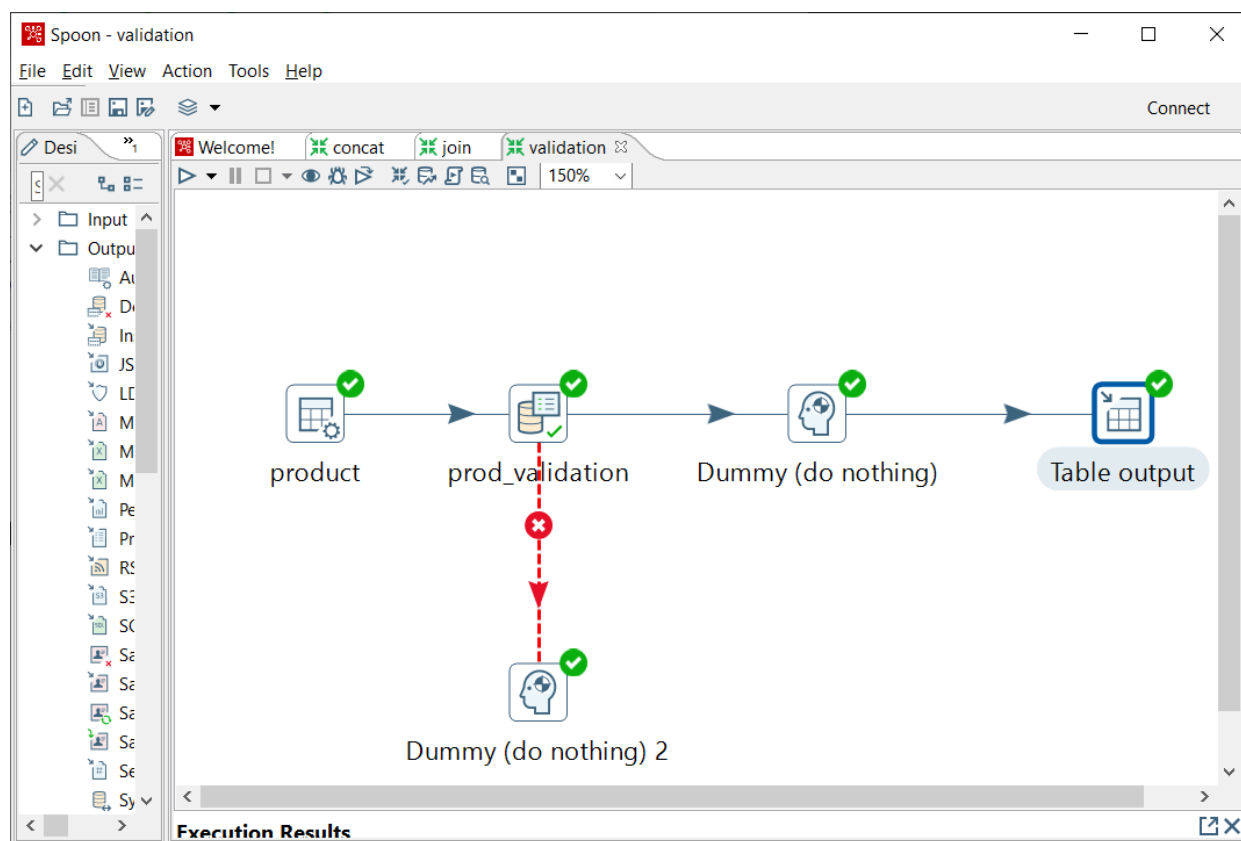
Table output configuration window (Database fields tab):

#	Table field	Stream field
1	prod_name	prod_name
2	prod_qty	prod_qty
3	check_status	check_status

Buttons: Execute, Get fields, Enter field mapping



12. Now again run the transformation



13. Click on preview data of the table output.

[illegible]

