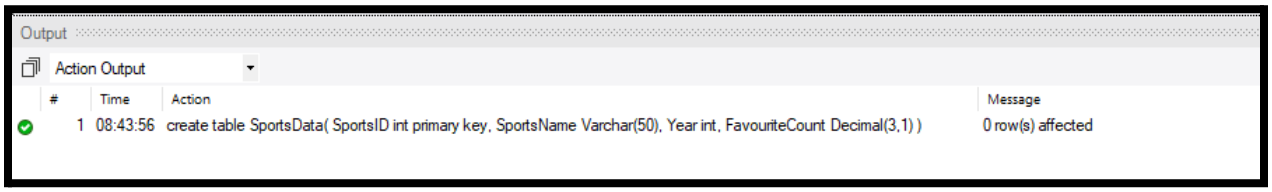
SEM I - ADDM Practical-07-01-2026

Name : Mansij Metar
Roll no: 91
Seat NO: 10211075

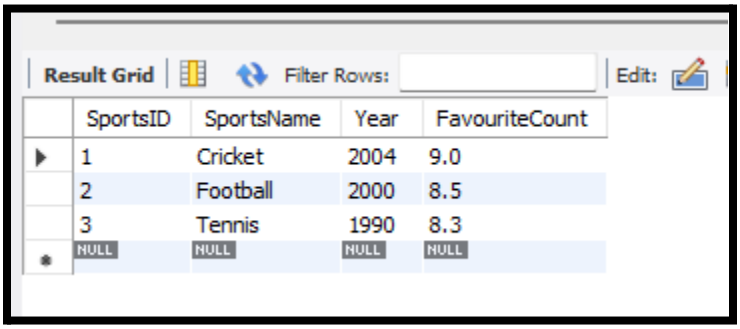
A) 1)Create a Table Sports Data with columns : SportsID, SportName, Year and favouriteCount
2)Insert % rows into the sportsData table
3)Display Sports with a favouriteCount above 8 out of 10
4)Delete a sports released in 2000
5) Update the FavouriteCount of all sports by adding 0.5

```
1) CREATE TABLE SportsData (
   SportsID INT PRIMARY KEY,
   SportName VARCHAR(50),
   Year INT,
   FavouriteCount DECIMAL(3,1)
);
```




```
2) INSERT INTO SportsData (SportsID, SportName, Year, FavouriteCount)
VALUES
(1, 'Cricket', 1995, 9.0),
(2, 'Football', 2002, 8.5),
(3, 'Tennis', 2000, 7.5),
(4, 'Basketball', 2010, 8.8),
(5, 'Hockey', 1998, 6.9);
```

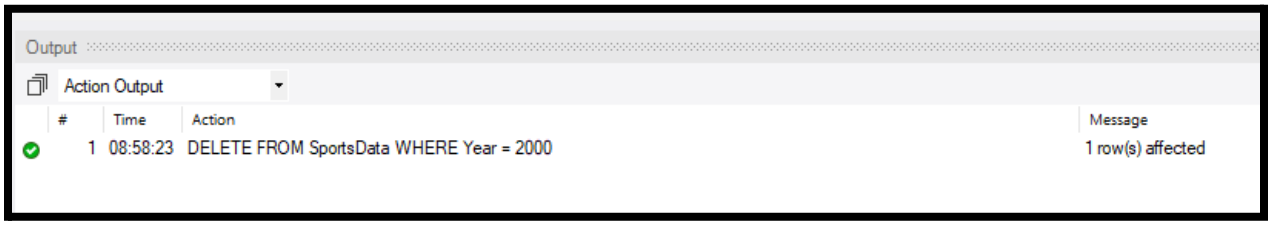
```
SELECT *
FROM SportsData
WHERE FavouriteCount > 8;
```



Result Grid | Filter Rows: | Edit:

SportsID	SportsName	Year	FavouriteCount
1	Cricket	2004	9.0
2	Football	2000	8.5
3	Tennis	1990	8.3
NULL	NULL	NULL	NULL

```
DELETE FROM SportsData
WHERE Year = 2000;
```



Output

Action Output

#	Time	Action	Message
1	08:58:23	DELETE FROM SportsData WHERE Year = 2000	1 row(s) affected

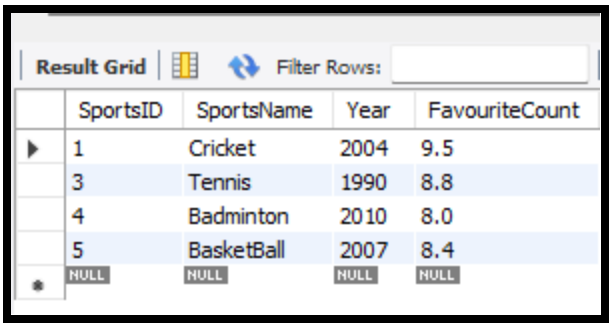
```
UPDATE SportsData
SET FavouriteCount = FavouriteCount + 0.5;
```

Output

Action Output

#	Time	Action	Message
1	09:00:46	update sportsdata Set favouriteCount=FavouriteCount + 0.5	4 row(s) affected Rows matched: 4 Changed: 4 Warnings: 0

Final Output



Result Grid | Filter Rows:

SportsID	SportsName	Year	FavouriteCount
1	Cricket	2004	9.5
3	Tennis	1990	8.8
4	Badminton	2010	8.0
5	BasketBall	2007	8.4
NULL	NULL	NULL	NULL

B) Given the following DataFrame:

Data = {'Name.': ['Tom','Jerry','Mickey','Donald']'
'Age':[21,None,24,None]'
Marks': [80,75,None ,90]}

After Converting to a Dataframe,write a python program to :

1. **Fill missing values in Age with the median age.**
2. **Drop rows where Marks is missing**

CODE:

```
import pandas as pd
Data = {
   'Name': ['Tom', 'Jerry', 'Mickey', 'Donald'],
   'Age': [21, None, 24, None],
   'Marks': [80, 75, None, 90]
}
print(Data)
```

OUTPUT:

```
{'Name': ['Tom', 'Jerry', 'Mickey', 'Donald'], 'Age': [21, None, 24, None], 'Marks': [80, 75, None, 90]}
```

```
df = pd.DataFrame(Data)
print(df)
```

OUTPUT:

```
     Name   Age  Marks
0     Tom  21.0   80.0
1   Jerry   NaN   75.0
2  Mickey  24.0    NaN
3  Donald   NaN   90.0
```

```
df['Age'] = df['Age'].fillna(df['Age'].median())
```

print(df[‘Age’])

OUTPUT:

```
0    21.0
1    22.5
2    24.0
3    22.5
Name: Age, dtype: float64
```

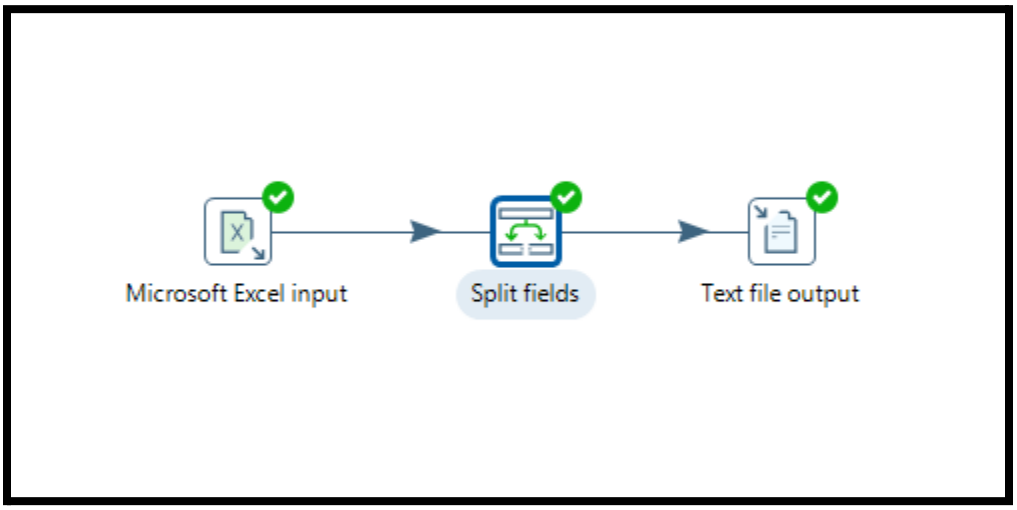
df = df.dropna(subset=['Marks'])

print(df)

OUTPUT:

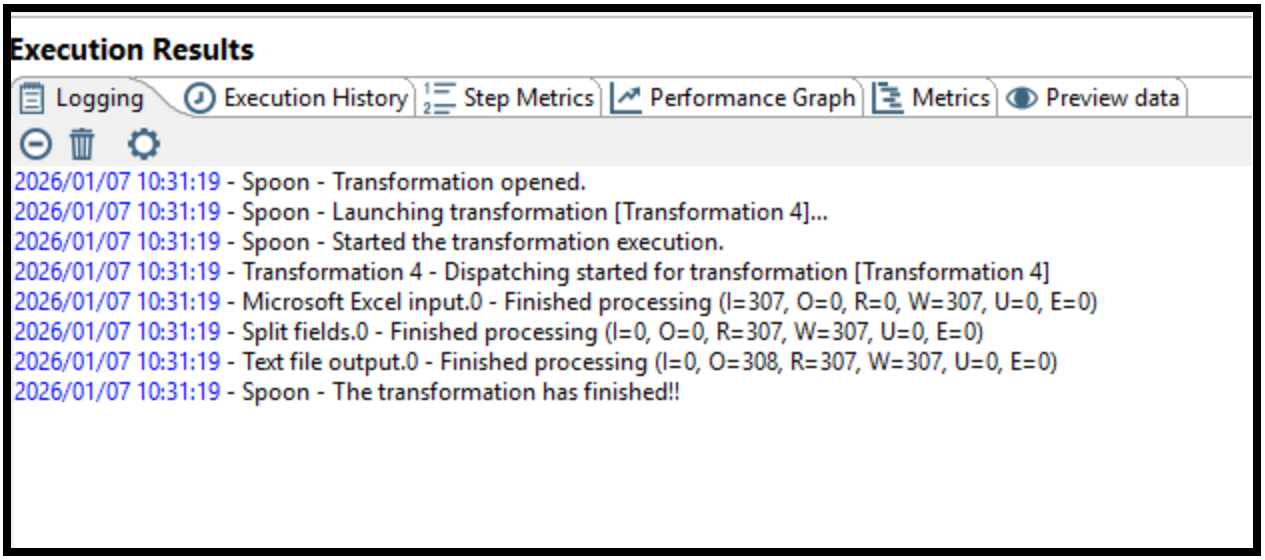
	Name	Age	Marks
0	Tom	21.0	80.0
1	Jerry	22.5	75.0
3	Donald	22.5	90.0

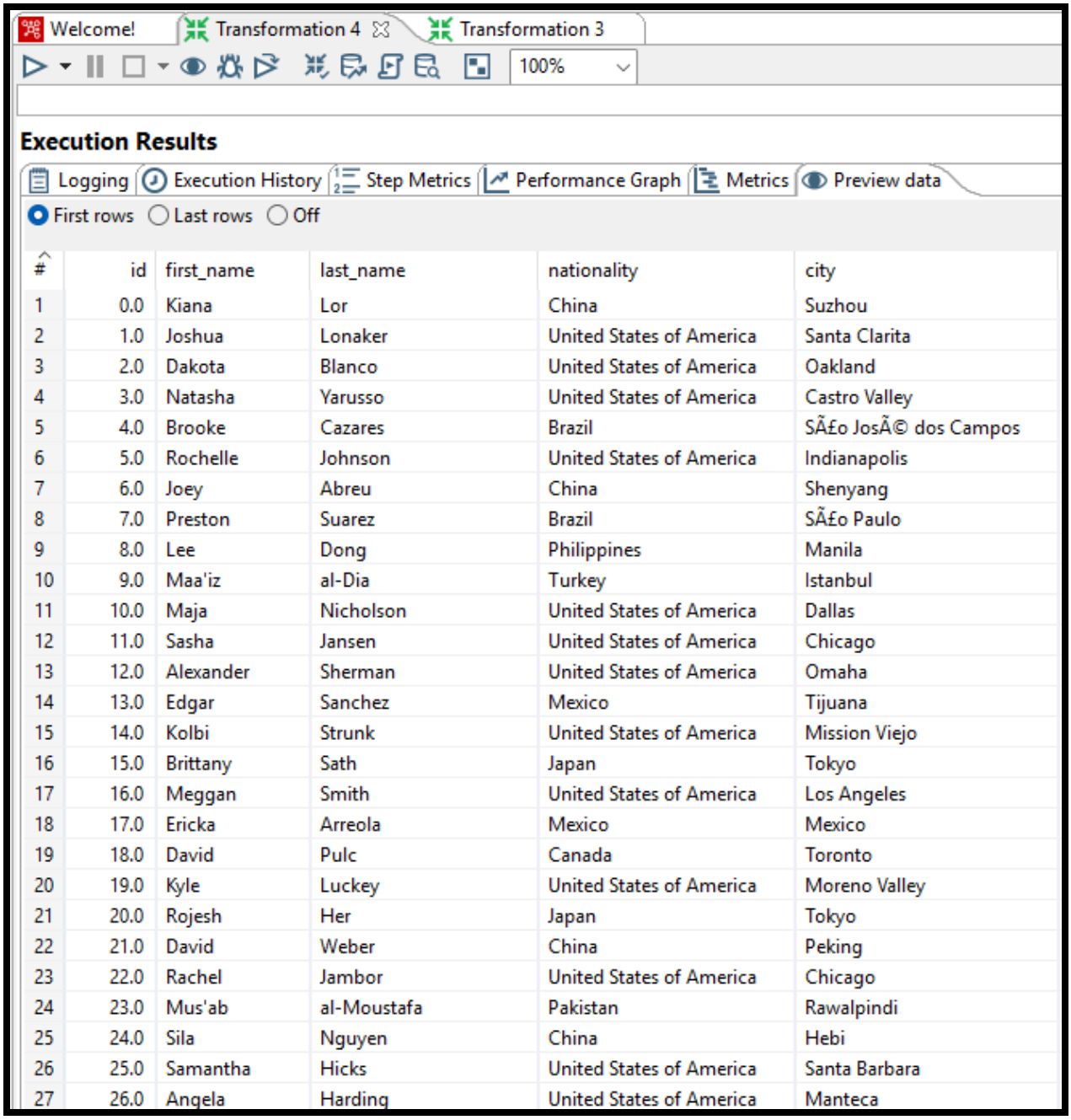
C) Retrieve data from any file (Excel) and store it after splitting full name column to firstname and lastname using pentaho



Original dataset

A	B	C	D
id	name	nationality	city
0	Kiana Lor	China	Suzhou
1	Joshua Lonaker	United States of America	Santa Clarita
2	Dakota Blanco	United States of America	Oakland
3	Natasha Yarusso	United States of America	Castro Valley
4	Brooke Cazares	Brazil	SÃ£o JosÃ© dos Campos
5	Rochelle Johnson	United States of America	Indianapolis
6	Joey Abreu	China	Shenyang
7	Preston Suarez	Brazil	SÃ£o Paulo
8	Lee Dong	Philippines	Manila
9	Maa'iz al-Dia	Turkey	Istanbul
10	Maja Nicholson	United States of America	Dallas
11	Sasha Jansen	United States of America	Chicago
12	Alexander Sherman	United States of America	Omaha
13	Edgar Sanchez	Mexico	Tijuana
14	Kolbi Strunk	United States of America	Mission Viejo


Execution Results
Logging
Execution History
Step Metrics
Performance Graph
Metrics
Preview data
2026/01/07 10:31:19 - Spoon - Transformation opened.
2026/01/07 10:31:19 - Spoon - Launching transformation [Transformation 4]...
2026/01/07 10:31:19 - Spoon - Started the transformation execution.
2026/01/07 10:31:19 - Transformation 4 - Dispatching started for transformation [Transformation 4]
2026/01/07 10:31:19 - Microsoft Excel input.0 - Finished processing (I=307, O=0, R=0, W=307, U=0, E=0)
2026/01/07 10:31:19 - Split fields.0 - Finished processing (I=0, O=0, R=307, W=307, U=0, E=0)
2026/01/07 10:31:19 - Text file output.0 - Finished processing (I=0, O=308, R=307, W=307, U=0, E=0)
2026/01/07 10:31:19 - Spoon - The transformation has finished!!



#	id	first_name	last_name	nationality	city
1	0.0	Kiana	Lor	China	Suzhou
2	1.0	Joshua	Lonaker	United States of America	Santa Clarita
3	2.0	Dakota	Blanco	United States of America	Oakland
4	3.0	Natasha	Yarusso	United States of America	Castro Valley
5	4.0	Brooke	Cazares	Brazil	SÃ£o JosÃ© dos Campos
6	5.0	Rochelle	Johnson	United States of America	Indianapolis
7	6.0	Joey	Abreu	China	Shenyang
8	7.0	Preston	Suarez	Brazil	SÃ£o Paulo
9	8.0	Lee	Dong	Philippines	Manila
10	9.0	Maa'iz	al-Dia	Turkey	Istanbul
11	10.0	Maja	Nicholson	United States of America	Dallas
12	11.0	Sasha	Jansen	United States of America	Chicago
13	12.0	Alexander	Sherman	United States of America	Omaha
14	13.0	Edgar	Sanchez	Mexico	Tijuana
15	14.0	Kolbi	Strunk	United States of America	Mission Viejo
16	15.0	Brittany	Sath	Japan	Tokyo
17	16.0	Meggan	Smith	United States of America	Los Angeles
18	17.0	Ericka	Arreola	Mexico	Mexico
19	18.0	David	Pulc	Canada	Toronto
20	19.0	Kyle	Luckey	United States of America	Moreno Valley
21	20.0	Rojesh	Her	Japan	Tokyo
22	21.0	David	Weber	China	Peking
23	22.0	Rachel	Jambor	United States of America	Chicago
24	23.0	Mus'ab	al-Moustafa	Pakistan	Rawalpindi
25	24.0	Sila	Nguyen	China	Hebi
26	25.0	Samantha	Hicks	United States of America	Santa Barbara
27	26.0	Angela	Harding	United States of America	Manteca