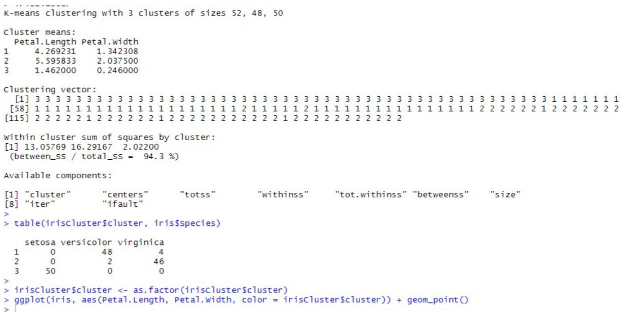
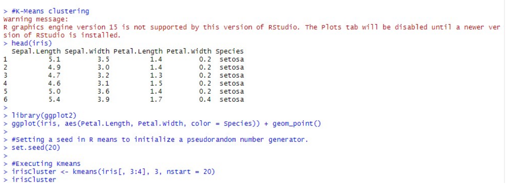
Q1. A

Output:

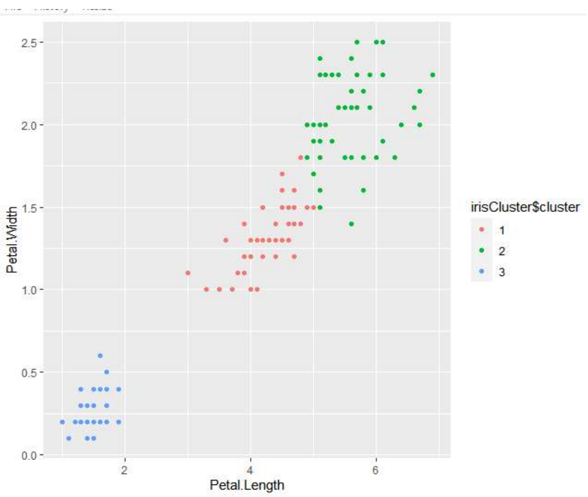


```
K-means clustering with 3 clusters of sizes 52, 48, 50

Cluster means:
  Petal.Length Petal.Width
1     4.269231    1.342308
2     5.595833    2.037500
3     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1
 [58] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2
[115] 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2

Within cluster sum of squares by cluster:
[1] 13.05769 16.29167  2.02200
 (between_SS / total_SS =  94.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
>
> table(irisCluster$cluster, iris$Species)

    setosa versicolor virginica
  1      0         48         4
  2      0          2        46
  3     50          0         0
>
> irisCluster$cluster <- as.factor(irisCluster$cluster)
> ggplot(iris, aes(Petal.Length, Petal.Width, color = irisCluster$cluster)) + geom_point()
> 
```



```
> #K-Means clustering
Warning message:
R graphics engine version 15 is not supported by this version of RStudio. The Plots tab will be disabled until a newer ver
sion of RStudio is installed.
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
>
> library(ggplot2)
> ggplot(iris, aes(Petal.Length, Petal.Width, color = Species)) + geom_point()
>
> #Setting a seed in R means to initialize a pseudorandom number generator.
> set.seed(20)
>
> #Executing Kmeans
> irisCluster <- kmeans(iris[, 3:4], 3, nstart = 20)
> irisCluster
```

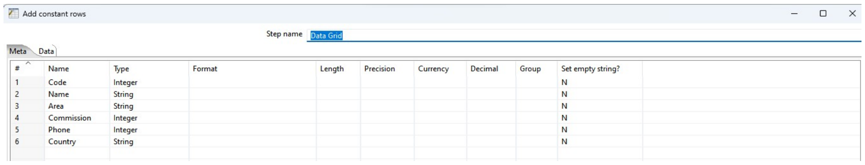
Graph:



Q1. B

Output:

Transformation 1) Split the Name Feild

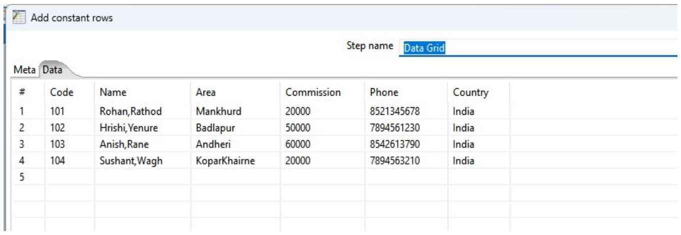


Add constant rows

Step name: Data Grid

Meta | Data

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Set empty string?
1	Code	Integer							N
2	Name	String							N
3	Area	String							N
4	Commission	Integer							N
5	Phone	Integer							N
6	Country	String							N

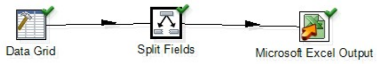


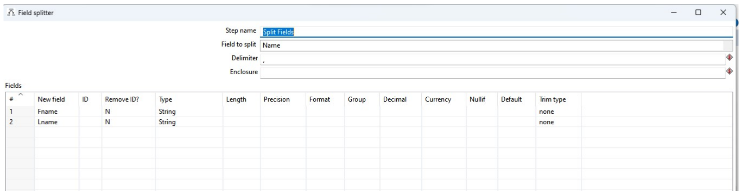
Add constant rows

Step name: Data Grid

Meta | Data

#	Code	Name	Area	Commission	Phone	Country
1	101	Rohan,Rathod	Mankhurd	20000	8521345678	India
2	102	Hrishi,Yenure	Badlapur	50000	7894561230	India
3	103	Anish,Rane	Andheri	60000	8542613790	India
4	104	Sushant,Wagh	KoparKhairne	20000	7894563210	India
5						





Field splitter

Step name: Split Fields

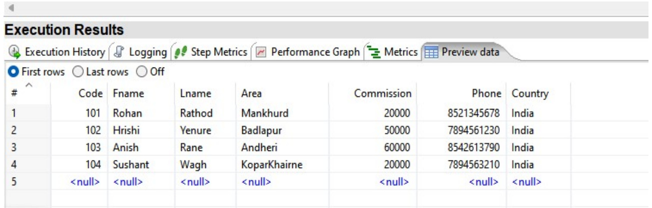
Field to split: Name

Delimiter: ,

Enclosure:

Fields

#	New field	ID	Remove ID?	Type	Length	Precision	Format	Group	Decimal	Currency	Nullif	Default	Trim type
1	Fname		N	String									none
2	Lname		N	String									none



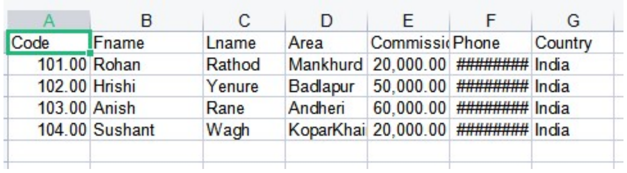
Execution Results

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

● First rows ○ Last rows ○ Off

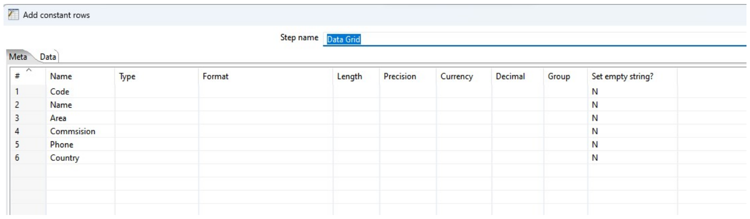
#	Code	Fname	Lname	Area	Commission	Phone	Country
1	101	Rohan	Rathod	Mankhurd	20000	8521345678	India
2	102	Hrishi	Yenure	Badlapur	50000	7894561230	India
3	103	Anish	Rane	Andheri	60000	8542613790	India
4	104	Sushant	Wagh	KoparKhairne	20000	7894563210	India
5	<null>	<null>	<null>	<null>	<null>	<null>	<null>

Loaded the data into Excel



A	B	C	D	E	F	G
Code	Fname	Lname	Area	Commissic	Phone	Country
101.00	Rohan	Rathod	Mankhurd	20,000.00	########	India
102.00	Hrishi	Yenure	Badlapur	50,000.00	########	India
103.00	Anish	Rane	Andheri	60,000.00	########	India
104.00	Sushant	Wagh	KoparKhai	20,000.00	########	India

Transformation 2) Add a Sequence:

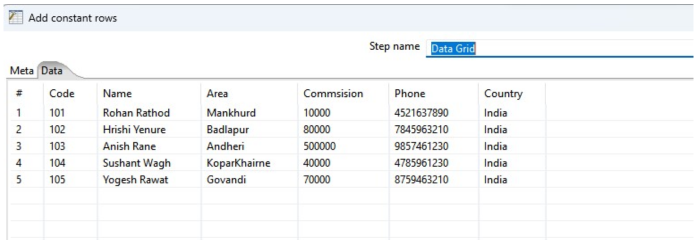


Add constant rows

Step name: Data Grid

Meta | Data

#	Name	Type	Format	Length	Precision	Currency	Decimal	Group	Set empty string?
1	Code								N
2	Name								N
3	Area								N
4	Commsision								N
5	Phone								N
6	Country								N

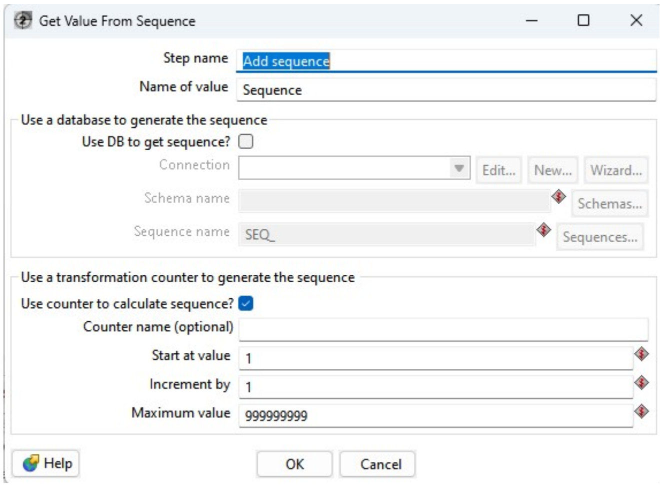


Add constant rows

Step name: Data Grid

Meta | Data

#	Code	Name	Area	Commsision	Phone	Country
1	101	Rohan Rathod	Mankhurd	10000	4521637890	India
2	102	Hrishi Yenure	Badlapur	80000	7845963210	India
3	103	Anish Rane	Andheri	500000	9857461230	India
4	104	Sushant Wagh	KoparKhairne	40000	4785961230	India
5	105	Yogesh Rawat	Govandi	70000	8759463210	India



Get Value From Sequence

Step name: Add sequence

Name of value: Sequence

Use a database to generate the sequence

Use DB to get sequence? □

Connection

Schema name

Sequence name: SEQ_

Use a transformation counter to generate the sequence

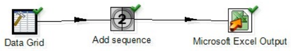
Use counter to calculate sequence? ☑

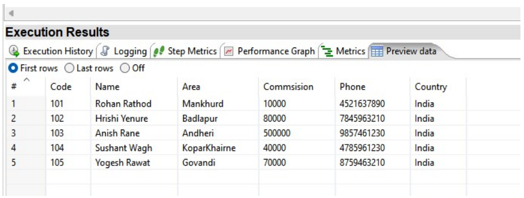
Counter name (optional)

Start at value: 1

Increment by: 1

Maximum value: 999999999



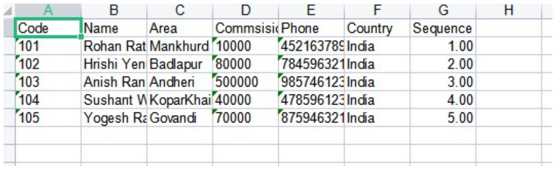


Execution Results

● First rows ○ Last rows ○ Off

#	Code	Name	Area	Commsision	Phone	Country
1	101	Rohan Rathod	Mankhurd	10000	4521637890	India
2	102	Hrishi Yenure	Badlapur	80000	7845963210	India
3	103	Anish Rane	Andheri	500000	9857461230	India
4	104	Sushant Wagh	KoparKhairne	40000	4785961230	India
5	105	Yogesh Rawat	Govandi	70000	8759463210	India

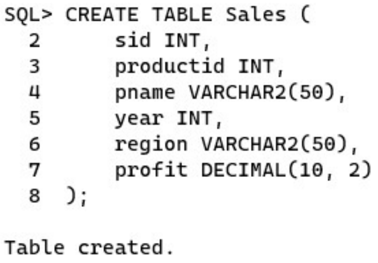
Loaded the data into Excel



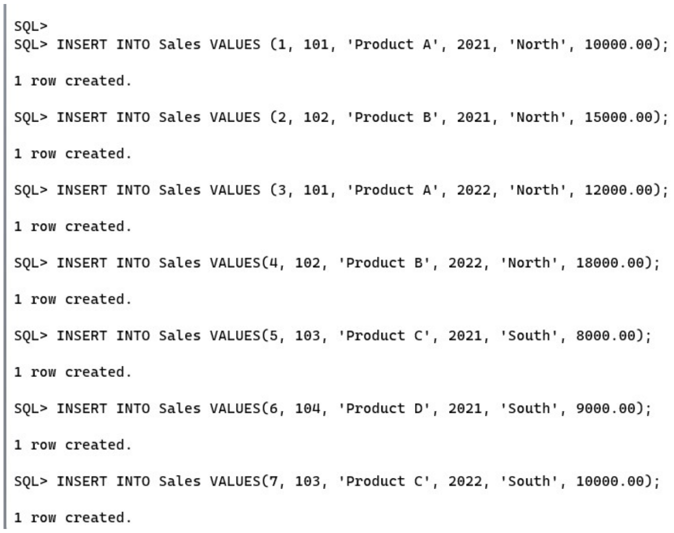
	A	B	C	D	E	F	G	H
	Code	Name	Area	Commsisi	Phone	Country	Sequence	
	101	Rohan Rat	Mankhurd	10000	452163789	India	1.00	
	102	Hrishi Yen	Badlapur	80000	784596321	India	2.00	
	103	Anish Ran	Andheri	500000	985746123	India	3.00	
	104	Sushant W	KoparKhai	40000	478596123	India	4.00	
	105	Yogesh Ra	Govandi	70000	875946321	India	5.00	

Q1. C)

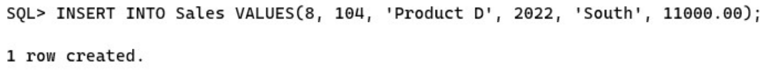
Output:



```
SQL> CREATE TABLE Sales (
  2      sid INT,
  3      productid INT,
  4      pname VARCHAR2(50),
  5      year INT,
  6      region VARCHAR2(50),
  7      profit DECIMAL(10, 2)
  8  );

Table created.
```

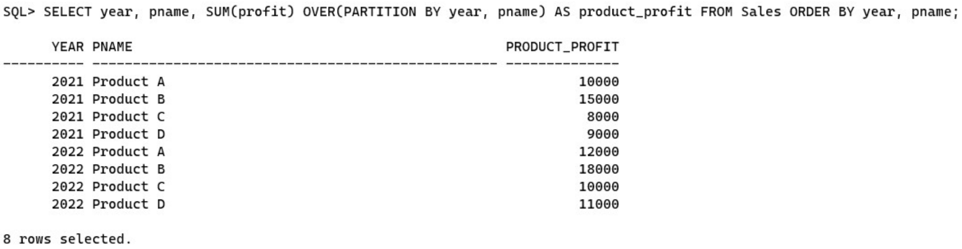


```
SQL>
SQL> INSERT INTO Sales VALUES (1, 101, 'Product A', 2021, 'North', 10000.00);

1 row created.

SQL> INSERT INTO Sales VALUES (2, 102, 'Product B', 2021, 'North', 15000.00);

1 row created.

SQL> INSERT INTO Sales VALUES (3, 101, 'Product A', 2022, 'North', 12000.00);

1 row created.

SQL> INSERT INTO Sales VALUES(4, 102, 'Product B', 2022, 'North', 18000.00);

1 row created.

SQL> INSERT INTO Sales VALUES(5, 103, 'Product C', 2021, 'South', 8000.00);

1 row created.

SQL> INSERT INTO Sales VALUES(6, 104, 'Product D', 2021, 'South', 9000.00);

1 row created.

SQL> INSERT INTO Sales VALUES(7, 103, 'Product C', 2022, 'South', 10000.00);

1 row created.
```



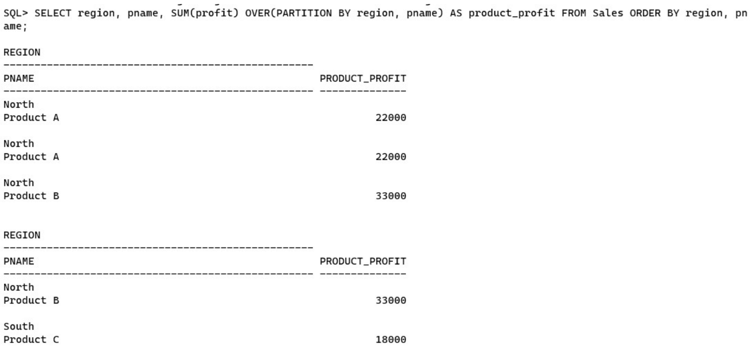
```
SQL> INSERT INTO Sales VALUES(8, 104, 'Product D', 2022, 'South', 11000.00);

1 row created.
```

Display product wise profit for each year

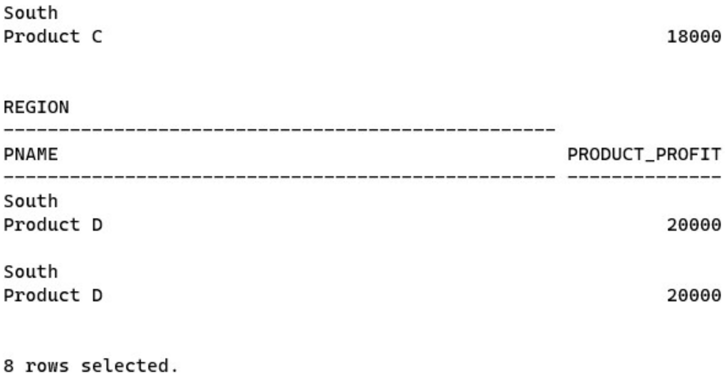


```
SQL> SELECT year, pname, SUM(profit) OVER(PARTITION BY year, pname) AS product_profit FROM Sales ORDER BY year, pname;

      YEAR PNAME                                              PRODUCT_PROFIT
---------- -------------------------------------------------- --------------
      2021 Product A                                                   10000
      2021 Product B                                                   15000
      2021 Product C                                                    8000
      2021 Product D                                                    9000
      2022 Product A                                                   12000
      2022 Product B                                                   18000
      2022 Product C                                                   10000
      2022 Product D                                                   11000

8 rows selected.
```

Display product wise profit for each region

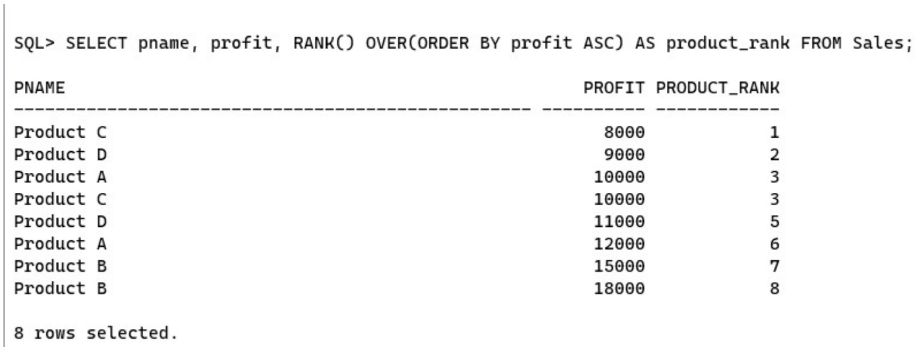


```
SQL> SELECT region, pname, SUM(profit) OVER(PARTITION BY region, pname) AS product_profit FROM Sales ORDER BY region, pn
ame;

REGION
--------------------------------------------------
PNAME                                              PRODUCT_PROFIT
-------------------------------------------------- --------------
North
Product A                                                   22000

North
Product A                                                   22000

North
Product B                                                   33000


REGION
--------------------------------------------------
PNAME                                              PRODUCT_PROFIT
-------------------------------------------------- --------------
North
Product B                                                   33000

South
Product C                                                   18000
```



```
South
Product C                                                   18000


REGION
--------------------------------------------------
PNAME                                              PRODUCT_PROFIT
-------------------------------------------------- --------------
South
Product D                                                   20000

South
Product D                                                   20000


8 rows selected.
```

Assign a sequential order(in ascending order), to products based on profit.

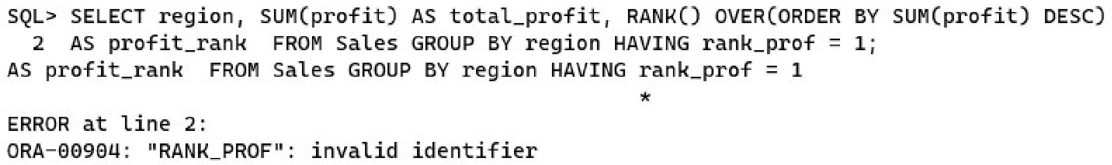


```
SQL> SELECT pname, profit, RANK() OVER(ORDER BY profit ASC) AS product_rank FROM Sales;

PNAME                                                  PROFIT PRODUCT_RANK
-------------------------------------------------- ---------- ------------
Product C                                                8000            1
Product D                                                9000            2
Product A                                               10000            3
Product C                                               10000            3
Product D                                               11000            5
Product A                                               12000            6
Product B                                               15000            7
Product B                                               18000            8

8 rows selected.
```

Display the name of region which is having highest profit, using Rank function.

SELECT region, SUM(profit) AS total_profit, RANK() OVER(ORDER BY SUM(profit) DESC)

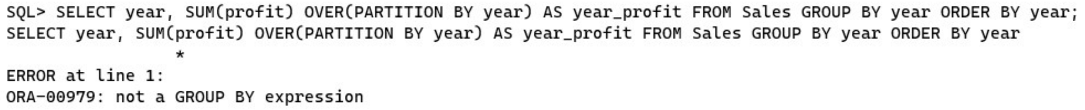
AS profit_rank FROM Sales GROUP BY region HAVING rank_prof = 1;



```
SQL> SELECT region, SUM(profit) AS total_profit, RANK() OVER(ORDER BY SUM(profit) DESC)
  2  AS profit_rank  FROM Sales GROUP BY region HAVING rank_prof = 1;
AS profit_rank  FROM Sales GROUP BY region HAVING rank_prof = 1
                                                  *
ERROR at line 2:
ORA-00904: "RANK_PROF": invalid identifier
```

Display year wise total profit

SELECT year, SUM(profit) OVER(PARTITION BY year) AS year_profit FROM Sales GROUP BY year ORDER BY year;



```
SQL> SELECT year, SUM(profit) OVER(PARTITION BY year) AS year_profit FROM Sales GROUP BY year ORDER BY year;
SELECT year, SUM(profit) OVER(PARTITION BY year) AS year_profit FROM Sales GROUP BY year ORDER BY year
                 *
ERROR at line 1:
ORA-00979: not a GROUP BY expression
```