MCAL13 ADBMS Lab INDEX				
Sr.No	**Title**	**CO**	**Date**	**Sign**
1	Implementation of Partitions: Range, List, Hash Partition Self-Learning Topics : Composite partition	CO1		
2	Analytical Queries Roll_Up, CUBE, First, Last, Lead, Lag, Rank and Dense Rank. Windowing functions ROWS-N PRECEDING AND FOLLOWING. Self-Learning Topics: Cume_list, Percent_rank	CO2		
3	Implementation of, • Abstract Data Type • Object table • Inheritance Self-Learning Topics: Nested ADT, Reference, Varray	CO1		
4	ETL Transformation with Pentaho 1. Copy data from Source & store to Target 2. Adding Sequence 3. Adding Calculator	CO3		

	4. Concatenation of Two Fields 5. Splitting of Two Fields 6. Number Range 7. String Operations 8. Sorting Data 9. Implement the Merge Join 10. Implement data validations on table data 11. Replace Strings 12. Splitting Fields to Rows			
5	Introduction to R, Install packages Loading packages ,Data types, checking variable type, printing variable and objects(Vector, Matrix, List, Factor, Data frame, Table)cbinding and rbinding Reading and Writing data: Setw(), getw(), data(),rm() Attaching and Detaching data. Reading data from the consol. Loading data from different data sources. (CSV, Excel) Self-Learning Topics: Operators, Conditional Statements and Loops, Functions, Loading data from Relational Databases, XML, Sorting, Date Conversion	CO4		

6	Data pre-processing techniques in R Naming and Renaming variables Adding a new variables Dealing with missing value ,Dealing with categorical data ,Data reduction using sub setting	CO4		
7	Implementation and analysis of Linear regression through graphical methods.	CO5		
8	Implementation and Analysis Classification algorithms like Naïve Bayesian, K-Nearest Neighbour, ID3, C4.5	CO5 .		
9	Implementation and analysis of Apriori Algorithm using Market Basket Analysis	CO5		
10	Implementation and analysis of clustering algorithms like K-means, Agglomerative	CO5		

Practical 1

Implementation of Partitions: Range, List, Hash Partition.

Self-Learning Topic: Composite partition Table range partition:

```
SQL> CREATE TABLE sales_range1
2    (salesman_id NUMBER(5),
3    salesman_name VARCHAR2(30),
4    sales_amount Number(10),
5    sales_date DATE)
6    PARTITION BY RANGE(sales_date)
7    (
8    PARTITION sales_jan2001 VALUES LESS
THAN(TO_DATE('01/02/2001','DD/MM/YYYY')),
9    PARTITION sales_feb2001 VALUES LESS
THAN(TO_DATE('01/03/2001','DD/MM/YYYY')),
10   PARTITION sales_mar2001 VALUES LESS
THAN(TO_DATE('01/04/2001','DD/MM/YYYY')),
11   PARTITION sales_apr2001 VALUES LESS
THAN(TO_DATE('01/05/2001','DD/MM/YYYY'))
12   ) ;

Table created.

SQL> SELECT TABLE_NAME, PARTITION_NAME FROM
USER_TAB_PARTITIONS WHERE
  2  TABLESPACE_NAME='USERS';

TABLE_NAME                PARTITION_NAME
------------------------------ ------------------------------ SALES_RANGE1
SALES_JAN2001
SALES_RANGE1               SALES_FEB2001
SALES_RANGE1               SALES_MAR2001
SALES_RANGE1               SALES_APR2001

4 rows selected.
```

```
SQL> insert into sales_range1 values(1, Yash, 25000,
TO_DATE('20/03/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into sales_range1 values(2, Yash, 30000,
TO_DATE('20/02/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into sales_range1 values(3, 'Rahul', 20000,
TO_DATE('20/01/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into sales_range1 values(4, 'Sam', 30000,
TO_DATE('22/04/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into sales_range1 values(5, 'Meera', 45000,
TO_DATE('18/02/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into sales_range1 values(6, 'Ajay', 34000,
TO_DATE('9/03/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into sales_range1 values(7, 'Keerthi', 24000,
TO_DATE('28/02/2001','DD/MM/YYYY'));

1 row created.

SQL> select * from sales_range1;
```

```
SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT SALES_DAT
----------- ------------------------------ ------------ ---------
3 Rahul                         20000 20-JAN-01
2 Yash                          30000 20-FEB-01
        5 Meera                         45000 18-FEB-01
        7 Keerthi                       24000 28-FEB-01
        1 Yash                          25000 20-MAR-01
        6 Ajay                          34000 09-MAR-01
        4 Sam                           30000 22-APR-01

7 rows selected.

SQL> Select * from sales_range1 partition(sales_jan2001);

SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT SALES_DAT
----------- ------------------------------ ------------ ---------
3 Rahul                         20000 20-JAN-01

SQL> Select * from sales_range1 partition(sales_feb2001);

SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT SALES_DAT
----------- ------------------------------ ------------ ---------
2 Yash                          30000 20-FEB-01
5 Meera                         45000 18-FEB-01
        7 Keerthi                       24000 28-FEB-01

SQL> Select * from sales_range1 partition(sales_mar2001);

SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT SALES_DAT
----------- ------------------------------ ------------ ---------
1 Yash                          25000 20-MAR-01
6 Ajay                          34000 09-MAR-01

SQL> Select * from sales_range1 partition(sales_apr2001); SALESMAN_ID
SALESMAN_NAME              SALES_AMOUNT SALES_DAT
----------- ------------------------------ ------------ ---------
```

4 Sam 30000 22-APR-01

LIST_PARTITIONING

```
SQL> CREATE TABLE newsales_list
2    (salesman_id NUMBER(5),
3    salesman_name VARCHAR2(30),
4    sales_city VARCHAR2(30),
5    sales_amount NUMBER(10),
6    sales_date DATE)
7    PARTITION BY LIST(sales_city)
8    (
9    PARTITION sales_west VALUES('Virar','Borivli'),
10   PARTITION sales_Harbur VALUES('Navi Mumbai','Panvel'),
11   PARTITION sales_central VALUES('Kalyan','Dombivli'),
12   PARTITION sales_other VALUES(DEFAULT)
13   )
14   enable row movement
15   ;

Table created.

SQL> SELECT TABLE_NAME,PARTITION_NAME FROM USER_TAB_PARTITIONS
WHERE
 TABLESPACE_NAME='USERS';

TABLE_NAME                PARTITION_NAME
------------------------------ ------------------------------ NEWSALES_LIST
SALES_WEST
NEWSALES_LIST             SALES_HARBUR
NEWSALES_LIST             SALES_CENTRAL
NEWSALES_LIST             SALES_OTHER

4 rows selected.

SQL> insert into newsales_list
values(1,'Yash','Kalyan','2000',TO_DATE('10/01/2001','DD/MM/YYYY'));
```

```
1 row created.

SQL> insert into newsales_list
values(2,Yash,'Panvel','4000',TO_DATE('15/04/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into newsales_list
values(3,'Nutan','Borivli','5000',TO_DATE('22/03/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into newsales_list
values(4,'Sam','Dombivli','2000',TO_DATE('12/03/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into newsales_list values(5,'Rahul','Navi
Mumbai','2000',TO_DATE('21/01/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into newsales_list
values(6,'Shyam','Ulhasnagar','10000',TO_DATE('4/03/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into newsales_list
values(7,'Ajay','Virar','8000',TO_DATE('5/01/2001','DD/MM/YYYY'));

1 row created.

SQL> select * from newsales_list;

SALESMAN_ID SALESMAN_NAME              SALES_CITY
```

---------- ---------- ----------

SALES_AMOUNT SALES_DAT
---------- ---------
3 Nutan Borivli
5000 22-MAR-01

7 Ajay Virar
8000 05-JAN-01

2 Arya Panvel
4000 15-APR-01

SALESMAN_ID SALESMAN_NAME SALES_CITY
---------- ---------- ----------
SALES_AMOUNT SALES_DAT
---------- ---------
5 Rahul Navi Mumbai
2000 21-JAN-01

1 Yash Kalyan
2000 10-JAN-01

4 Sam Dombivli
2000 12-MAR-01

SALESMAN_ID SALESMAN_NAME SALES_CITY
---------- ---------- ----------
SALES_AMOUNT SALES_DAT
---------- ---------
6 Shyam Ulhasnagar
10000 04-MAR-01

7 rows selected.

SQL> SELECT*FROM newsales_list partition(sales_west);

SALESMAN_ID SALESMAN_NAME SALES_CITY
---------- ---------- ----------
SALES_AMOUNT SALES_DAT
---------- ---------
3 Nutan Borivli
5000 22-MAR-01

7 Ajay Virar
8000 05-JAN-01

SQL> SELECT*FROM newsales_list partition(sales_harbur);

SALESMAN_ID SALESMAN_NAME SALES_CITY
---------- ---------- ----------
SALES_AMOUNT SALES_DAT
---------- ---------
2 Arya Panvel
4000 15-APR-01

5 Rahul Navi Mumbai
2000 21-JAN-01

SQL> SELECT*FROM newsales_list partition(sales_central);

SALESMAN_ID SALESMAN_NAME SALES_CITY
---------- ---------- ----------
SALES_AMOUNT SALES_DAT
---------- ---------
1 Yash Kalyan

2000 10-JAN-01

4 Sam Dombivli
2000 12-MAR-01

SQL> SELECT*FROM newsales_list partition(sales_other);

SALESMAN_ID SALESMAN_NAME SALES_CITY
---------- ---------- ----------
SALES_AMOUNT SALES_DAT
---------- ---------
6 Shyam Ulhasnagar
10000 04-MAR-01

Hash partioning

```
SQL> CREATE TABLE sales_hash
2  (salesman_id NUMBER(5),
3  salesman_name VARCHAR2(30),
4  sales_amount NUMBER(10),
5  week_no NUMBER(2) )
6  PARTITION BY HASH(salesman_id)
7  PARTITIONS 4
8  ;

Table created.
SQL> insert into sales_hash values(101,'Yash',30000,12);

1 row created.

SQL> insert into sales_hash values(102,'Adifa',42000,5);

1 row created.
```

```
SQL> insert into sales_hash values(103,'Sam',20000,10);

1 row created.

SQL> insert into sales_hash values(104,'Nutan',32000,8);

1 row created.

SQL> insert into sales_hash values(105,'Rahul',45000,12);

1 row created.

SQL> insert into sales_hash values(106,'Kartik',25000,4);

1 row created.

SQL> insert into sales_hash values(107,'Akash',33000,9);

1 row created.

SQL> select * from sales_hash;

SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT    WEEK_NO
----------- ------------------------------ ------------ ----------
104 Nutan                          32000         8        102
Adifa                         42000         5
      103 Sam                           20000        10
105 Rahul                         45000        12
      107 Akash                         33000         9
      101 Yash                          30000        12
106 Kartik                        25000         4

7 rows selected.

SQL>SELECT TABLE_NAME, PARTITION_NAME FROM USER_TAB_PARTITIONS
WHERE  TABLESPACE_NAME='SYSTEM';
```

```
TABLE_NAME                     PARTITION_NAME
------------------------------ ------------------------------
SALES_HASH                     SYS_P21
SALES_HASH                     SYS_P22
SALES_HASH                     SYS_P23
SALES_HASH                     SYS_P24

4 rows selected.

SQL> select*from sales_HASH partition(SYS_P21);

no rows selected

SQL> select*from sales_HASH partition(SYS_P22);

SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT    WEEK_NO
----------- ------------------------------ ------------ ----------
104 Nutan                           32000         8

SQL> select*from sales_HASH partition(SYS_P23);

SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT    WEEK_NO
----------- ------------------------------ ------------ ----------
102 Adifa                          42000         5
103 Sam                            20000        10
105 Rahul                          45000        12
      107 Akash                          33000         9

SQL> select*from sales_HASH partition(SYS_P24);

SALESMAN_ID SALESMAN_NAME              SALES_AMOUNT    WEEK_NO
----------- ------------------------------ ------------ ----------
      101 Yash                         30000        12
106 Kartik                         25000         4
```

Composite partioning

```
SQL> CREATE TABLE compositep
2    (
3    purchase_no NUMBER,
4    purchase_date DATE,
5    Product NUMBER,
6    Quantity NUMBER
7    )
8    PARTITION BY RANGE(purchase_date)
9    SUBPARTITION BY HASH(Product)SUBPARTITIONS 4
10   (
11   PARTITION order1 VALUES LESS
12   THAN(TO_DATE('01/02/2024','DD/MM/YYYY')),
13   PARTITION order2 VALUES LESS
14   THAN(TO_DATE('01/03/2024','DD/MM/YYYY')),
15   PARTITION order3 VALUES LESS
16   THAN(TO_DATE('01/04/2024','DD/MM/YYYY')),
17   PARTITION order4 VALUES LESS
18   THAN(TO_DATE('01/05/2024','DD/MM/YYYY'))
19   );

Table created.

SQL> SELECT TABLE_NAME,PARTITION_NAME FROM USER_TAB_PARTITIONS
  2  WHERE TABLESPACE_NAME='USERS';

TABLE_NAME                     PARTITION_NAME
------------------------------ ------------------------------ COMPOSITEP
ORDER1
COMPOSITEP                     ORDER2
COMPOSITEP                     ORDER3
COMPOSITEP                     ORDER4

4 rows selected.
```

```
SQL> INSERT INTO COMPOSITEP
  2   VALUES(01,TO_DATE('12/01/2024','DD/MM/YYYY'),101,2);
1   row created.

SQL> INSERT INTO COMPOSITEP
2   VALUES(02,TO_DATE('23/01/2024','DD/MM/YYYY'),102,6);

1   row created.

SQL> INSERT INTO COMPOSITEP
2   VALUES(03,TO_DATE('13/03/2024','DD/MM/YYYY'),103,5);

1   row created.


SQL> INSERT INTO COMPOSITEP
2   VALUES(04,TO_DATE('25/04/2024','DD/MM/YYYY'),104,1);

1   row created.

SQL> INSERT INTO COMPOSITEP
2   VALUES(05,TO_DATE('13/03/2024','DD/MM/YYYY'),105,7);

1 row created.

SQL> SELECT * FROM COMPOSITEP PARTITION(order1);

PURCHASE_NO PURCHASE_    PRODUCT   QUANTITY
----------- --------- ---------- ----------
2 23-JAN-24       102         6
        1 12-JAN-24       101         2

SQL> SELECT * FROM COMPOSITEP PARTITION(order2);

no rows selected
```

```
SQL> SELECT * FROM COMPOSITEP PARTITION(order3); PURCHASE_NO
PURCHASE_    PRODUCT   QUANTITY
----------- --------- ---------- ----------
3 13-MAR-24       103         5
        5 13-MAR-24       105         7

SQL> SELECT * FROM COMPOSITEP PARTITION(order4);

PURCHASE_NO PURCHASE_    PRODUCT   QUANTITY
-----------    ---------    ----------    ----------
4 25-APR-24       104         1
```

Practical 2

Analytical Queries Roll_Up, CUBE, First, Last, Lead, Lag, Rank, Dense rank Windowing functions ROWS-N PRECEDING AND FOLLOWING:

```
SQL> CREATE TABLE Employeedata
2  (Emp_no NUMBER(5),
3  Dep_no NUMBER(5),
4  DOB DATE,
5  Salary NUMBER(10),
6  Comm NUMBER(8),
7  Job VARCHAR2(30)
8  )
9  ;

Table created.

SQL> INSERT into Employeedata
values(101,10,TO_DATE('03/09/2002','DD/MM/YYYY'),30000,2000,'Manager');

1 row created.

SQL> INSERT into Employeedata
values(102,10,TO_DATE('05/10/2002','DD/MM/YYYY'),3000,2000,'HR');

1 row created.

SQL> INSERT into Employeedata
values(103,10,TO_DATE('02/02/2003','DD/MM/YYYY'),30000,2000,'HR');

1 row created.

SQL> INSERT into Employeedata
values(104,20,TO_DATE('20/06/2003','DD/MM/YYYY'),40000,1000,'Technical')
;
1 row created.
```

```
SQL> INSERT into Employeedata
values(105,20,TO_DATE('20/08/2003','DD/MM/YYYY'),20000,1000,'Technical')
;
1 row created.

SQL> INSERT into Employeedata
values(106,30,TO_DATE('10/05/2003','DD/MM/YYYY'),20000,1000,'Sales');

1 row created.

SQL> INSERT into Employeedata
values(107,30,TO_DATE('10/07/2003','DD/MM/YYYY'),35000,2000,'Sales');

1 row created.

SQL> INSERT into Employeedata
values(108,20,TO_DATE('10/05/2003','DD/MM/YYYY'),20000,1000,'Sales');

1   row created.
```

ROLL UP:

```
SQL> SELECT Dep_no,Job,count(*),sum(salary)
2   from Employeedata
3   group by rollup(Dep_no,Job);

   DEP_NO JOB                       COUNT(*) SUM(SALARY)
---------- ------------------------------ ---------- -----------
      10 HR                            2      33000
      10 Manager                            1      30000
      10                               3      63000
      20 Sales                            1      20000
      20 Technical                          2      60000
      20                               3
80000
30 Sales                          2      55000
```

```
      30                          2      55000
8     198000
9     rows selected.
```

CUBE:

```
SQL>select dept_no,job,count(*),sum(salary)from Employeedata group by
2  cube(dept_no,job);

DEPT_NO JOB                 COUNT(*) SUM(SALARY)
---------- ------------------- ---------- -----------
                           4     157000
        Sales                2      22000
        Manager                 2     135000
      10                     1      70000
      10 Manager                1      70000
      20                     1      65000
      10 Manager                1      65000
      10                     1      10000
      10 HR                  1      10000
          20 Technical              1      65000
      30 Sales                1      10000
      30                     1      12000

      12 rows selected.
```

RANK:

```
SQL>select Emp_no,dept_no,salary,rank()over( partition by dept_no order
2 bysalary)as Rank from Employeedata ;

EMPNO    DEPT_NO    SALARY      RANK
---------- ---------- ---------- ----------
101       20     30000         3
102    10      3000         5
103    20     40000         1
```

105 20 20000 4
107 30 35000 2

5 rows selected.

FIRST:

```
SQL> select Dep_no,salary,
2 max(salary)keep(DENSE_RANK FIRST ORDER BY salary desc)
3 over(PARTITION BY Dep_no)"max"
4 from Employeedata;

DEP_NO        SALARY          max
--------------- ---------------- ------------------ 10
30000             30000
10                3000             3000
10                30000            30000
20                40000            40000
20                20000            20000
30                20000            20000
30                35000             35000
20                20000             20000
```

Windowing function ROWS N- PRECEEDING AND FOLLOWING.

```
create table emp10(   2
employee_id VARCHAR(10),
3  employee_name VARCHAR(20),
4  dep_name VARCHAR(20),
5  dep_id     NUMBER(10),
6  salary NUMBER(7)
7  );

Table created.

INSERT INTO emp10 VALUES('1001','Yash','IT',10,40000);
```

```
1 row created.

SQL> INSERT INTO emp10 VALUES('1002','Sneha','IT',10,30000);

1 row created.

SQL> INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,salary);
Enter value for employee_id: 1003
Enter value for employee_name: Priya
Enter value for dep_name: IT
Enter value for dep_id: 10 old
1: INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,salary) new
1: INSERT INTO emp10 VALUES('1003','Priya','IT',10,salary)

SQL> INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary);
Enter value for employee_id: 1003
Enter value for employee_name: Priya
Enter value for dep_name: IT
Enter value for dep_id: 10 Enter
value for salary: 20000 old   1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1003','Priya','IT',10,20000)

1 row created.

SQL> r
  1* INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary)
Enter value for employee_id: 1004
Enter value for employee_name: Rahul
Enter value for dep_name: Support
```

```
Enter value for dep_id: 30 Enter
value for salary: 15000 old   1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1004','Rahul','Support',30,15000)

1 row created.

SQL> r
  1* INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary)
Enter value for employee_id: 1005
Enter value for employee_name: Keerthi
Enter value for dep_name: Support
Enter value for dep_id: 30 Enter
value for salary: 18000 old   1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1005','Keerthi','Support',30,18000)

1 row created.

SQL> r
  1* INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary)
Enter value for employee_id: 1006
Enter value for employee_name: Soham
Enter value for dep_name: Sales
Enter value for dep_id: 20 Enter
value for salary: 35000 old   1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1006','Soham','Sales',20,35000)

1 row created.

SQL> r
```

```
  1* INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary)
Enter value for employee_id: 1007
Enter value for employee_name: Mansi
Enter value for dep_name: Sales
Enter value for dep_id: 20 Enter
value for salary: 30000 old   1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1007','Mansi','Sales',20,30000)

1 row created.

SQL> r
  1* INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary)
Enter value for employee_id: 1008
Enter value for employee_name: Rohit
Enter value for dep_name: Sales
Enter value for dep_id: 20 Enter
value for salary: 35000 old   1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1008','Rohit','Sales',20,35000)

1 row created.
r
  1* INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary)
Enter value for employee_id: 1009
Enter value for employee_name: Sam
Enter value for dep_name: Sales
Enter value for dep_id: 20 Enter
value for salary: 30000 old   1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1009','Sam','Sales',20,30000)
```

1 row created.

SQL> r
1* INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary)
Enter value for employee_id: 1010
Enter value for employee_name: Mohan
Enter value for dep_name: Sales
Enter value for dep_id: 20 Enter
value for salary: 30000 old 1:
INSERT INTO emp10
VALUES('&employee_id','&employee_name','&dep_name',&dep_id,&salary) new
1: INSERT INTO emp10 VALUES('1010','Mohan','Sales',20,30000)

1 row created.

SQL> select * from emp10;

EMPLOYEE_I	EMPLOYEE_NAME	DEP_NAME	DEP_ID	SALARY
1001	Yash	IT	10	40000
1002	Sneha	IT	10	30000
1003	Priya	IT	10	20000
1004	Rahul	Support	30	15000
1005	Keerthi	Support	30	18000
1006	Soham	Sales	20	35000
1007	Mansi	Sales	20	30000
1008	Rohit	Sales	20	35000
1009	Sam	Sales	20	30000
1010	Mohan	Sales	20	30000

10 rows selected. select
employee_id,employee_name,salary,sum(salary)over(partition by dep_id order
by dep_id rows 2 preceding)Total from emp10 order by dep_id; EMPLOYEE_I
EMPLOYEE_NAME SALARY TOTAL
---------- ---------- ---------- ----------

```
1002  Sneha                30000     30000
1003  Priya               20000     50000
1001  Yash               40000     90000
1009  Sam                 30000     30000
1010  Mohan                30000     60000
1006  Soham                35000     95000
1007  Mansi                30000     95000
1008  Rohit               35000    100000
1004  Rahul                15000     15000
1005  Keerthi              18000     33000

10 rows selected.
```

SQL> select employee_id,employee_name,salary,sum(salary)over(partition by dep_id order by dep_id rows 1 preceding)Total from emp10 order by dep_id;

```
EMPLOYEE_I EMPLOYEE_NAME          SALARY      TOTAL
---------- -------------------- ---------- ----------
1002  Sneha                30000     30000
1003  Priya               20000     50000
1001      Yash                40000     60000
1009  Sam                 30000     30000
1010  Mohan                30000     60000
1006  Soham                35000     65000
1007  Mansi                30000     65000
1008  Rohit               35000     65000
1004  Rahul                15000     15000
1005  Keerthi              18000     33000

10 rows selected.
```

SQL> select employee_id,dep_name,salary,sum(salary)over(partition by dep_id order by salary rows between 3 preceding and 1 following)Total_sal from emp10 order by dep_id,salary;

```
EMPLOYEE_I DEP_NAME              SALARY  TOTAL_SAL
```

---------- -------------------- ---------- ---------- 1003
IT 20000 50000
1002 IT 30000 90000
1001 IT 40000 90000
1010 Sales 30000 60000
1007 Sales 30000 90000
1009 Sales 30000 125000
1006 Sales 35000 160000
1008 Sales 35000 130000
1004 Support 15000 33000
1005 Support 18000 33000
10 rows selected.

Practical 3

Implementation of Abstract Data Type,Object table, Inheritance :

```
create type type_name As object(
2  fname VARCHAR(10),
3  mname VARCHAR(10),
4  lname VARCHAR(10));
 5
 6  /
```

Type created.

```
SQL> create type type_address As object(
2  street VARCHAR(15),
3  city VARCHAR(15),
4  pincode NUMBER(10));
5  /
```

Type created.

```
SQL> CREATE TABLE customer1
2  (
3  c_id NUMBER(5) PRIMARY KEY,
4  c_name type_name,
5  c_add type_address,
6  c_phoneno NUMBER(10)
7  );
```

Table created.

```
SQL> INSERT INTO customer1
  2  VALUES(1,type_name('Yash','K','Thomas),type_address('Hanuman
Nagar','Mumbai',400042),1244553278); ERROR:
ORA-01756: quoted string not properly terminated

SQL> INSERT INTO customer1
```

```
VALUES(1,type_name('Yash','K','Thomas'),type_address('Hanuman
Nagar','Mumbai',400042),1244553278);

1 row created.

SQL> INSERT INTO customer1
VALUES(2,type_name('Nutan','N','Magar'),type_address('Sai
nagar','Mumbai',400042),1233446634);

1 row created.
INSERT INTO customer1
VALUES(3,type_name('Akanksha','j','Jacob'),type_address('Hanuman
Nagar','Mumbai',400042),1244568823);

1 row created.

SQL> INSERT INTO customer1
VALUES(5,type_name('Rahul','s','Gupta),type_address('New
street','Pune',400035),1245562318); ERROR:
ORA-01756: quoted string not properly terminated


SQL> INSERT INTO customer1
VALUES(5,type_name('Rahul','s','Gupta'),type_address('New
street','Pune',400035),1245562318);

1 row created.

SQL> INSERT INTO customer1
VALUES(4,type_name('Uday','B','Singh'),type_address('sawant
road','Pune',400035),6643234167);

1 row created.
SQL> select * from customer1;

    C_ID
----------
```

C_NAME(FNAME, MNAME, LNAME)

C_ADD(STREET, CITY, PINCODE)

C_PHONENO

1
TYPE_NAME('Yash', 'K', 'Thomas')
TYPE_ADDRESS('Hanuman Nagar', 'Mumbai', 400042)
1244553278

C_ID

C_NAME(FNAME, MNAME, LNAME)

C_ADD(STREET, CITY, PINCODE)

C_PHONENO

2
TYPE_NAME('Nutan', 'N', 'Magar')
TYPE_ADDRESS('Sai nagar', 'Mumbai', 400042)
1233446634
C_ID

C_NAME(FNAME, MNAME, LNAME)

C_ADD(STREET, CITY, PINCODE)

C_PHONENO

3
TYPE_NAME('Akanksha', 'j', 'Jacob')
TYPE_ADDRESS('Hanuman Nagar', 'Mumbai', 400042)
1244568823

```
    C_ID
----------
C_NAME(FNAME, MNAME, LNAME)
--------------------------------------------------------------------------------
C_ADD(STREET, CITY, PINCODE)
--------------------------------------------------------------------------------
 C_PHONENO
----------
       5
TYPE_NAME('Rahul', 's', 'Gupta')
TYPE_ADDRESS('New street', 'Pune', 400035)
1245562318


    C_ID
----------
C_NAME(FNAME, MNAME, LNAME)
--------------------------------------------------------------------------------
C_ADD(STREET, CITY, PINCODE)
--------------------------------------------------------------------------------
 C_PHONENO
----------
       4
TYPE_NAME('Uday', 'B', 'Singh')
TYPE_ADDRESS('sawant road', 'Pune', 400035)
6643234167

select c.c_add.street from customer1 c where c_id=1;

C_ADD.STREET
---------------
Hanuman Nagar

SQL> select c.c_name.fname from customer1 c where c_id=1;
```

```
C_NAME.FNA
----------
Yash

SQL> select c_name from customer1;

C_NAME(FNAME, MNAME, LNAME)
-------------------------------------------------------------------------------- TYPE_NAME('Yash',
'K', 'Thomas')
TYPE_NAME('Nutan', 'N', 'Magar')
TYPE_NAME('Akanksha', 'j', 'Jacob')
TYPE_NAME('Rahul', 's', 'Gupta')
TYPE_NAME('Uday', 'B', 'Singh')

SQL> select c_id,c.c_name.lname from customer1 c;

    C_ID C_NAME.LNA
---------- ----------
1 Thomas
2 Magar
3 Jacob
       5 Gupta
4 Singh
SQL> select c.c_name.fname||' '||c.c_name.mname||' '||c.c_name.lname from
customer1 c;

C.C_NAME.FNAME||''||C.C_NAME.MNA
--------------------------------
Yash K Thomas
Nutan N Magar
Akanksha j Jacob
Rahul s Gupta
Uday B Singh
```

OBJECT TABLE:

```
SQL> create or replace type stud_type as object(roll_no number(5),name
varchar2(30));

2/

Type created.
SQL> create table students of stud_type;

Table created.

SQL> insert into students values(stud_type(1,'Yash'));

1 row created.
SQL> insert into students values(stud_type(2, 'Arya'));

1       row created.


SQL> select * from students;

ROLL _NO    NAME
------------   -------- 1
Yash
2       Arya

 SQL> select s.roll_no from students s;

  ROLL _NO
  ------------
  1
  2
```

Inheritance:

```
SQL> create or replace type ANIMAL_TY as object(Breed varchar2(25), Name
varchar2(25), BirthDate DATE);

2/
```

```
Type created.
SQL> create table ANIMAL of ANIMAL_TY;

Table created.
SQL> insert into ANIMAL values(ANIMAL_TY('Dog', 'Chase', '01-feb-24'));

1 row created.
SQL> insert into ANIMAL values(ANIMAL_TY('mule', 'husky', '03-apr-24'));

1 row created.
SQL> select REF(A) from ANIMAL A;
REF(A)
--------------------------------------------------------------------------------
00002802097EBOB876052742618FE51CDEBDA06F3D23D11FE87F3440E581
DC113847C9D2CE0041DCC90000

000028020920CF2D0218C94BD1899992C48FE3B17C23D11FE87F3440E581
DC113847C9D2CE0041DCC90001


SQL> create table KEEPER(Keeper_name varchar2(25), Animal_Kept REF
ANIMAL_TY);

Table created.
SQL> describe KEEPER;

Name                    Null?           Type
----------------------  -----------  ------------------------------------------- KEEPER_NAME
VARCHAR2(25)
ANIMAL_KEPT                     REF OF ANIMAL_TY


SQL> insert into KEEPER select 'John' REF(A) from ANIMAL A where Name='chase';

1 row created.
```

SQL> select * from KEEPER;

KEEPER_NAME

ANIMAL_KEPT
--- --- ---
JOHN

000022020820CF2D0218C94BD1899992C48FE3B17C23D11FE87F3440E581
DC113847C9D2CE

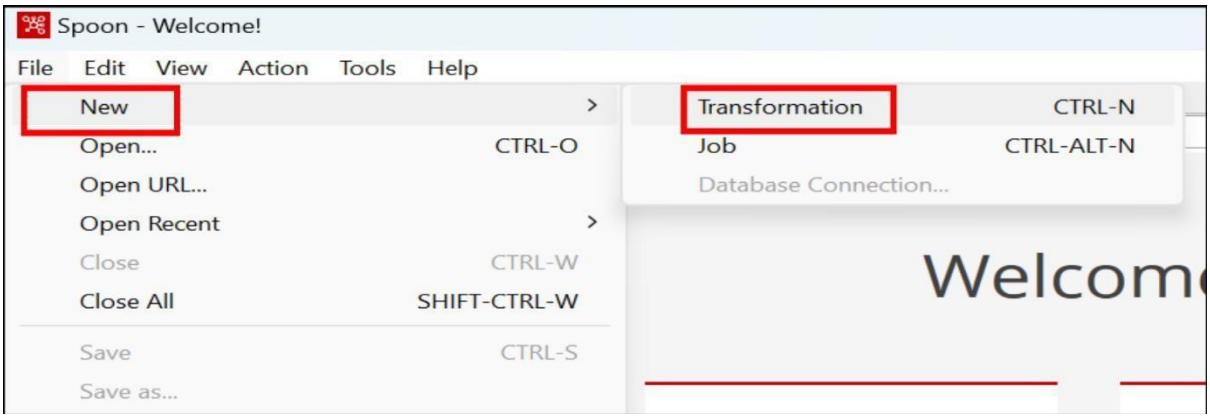
Practical 4

Implementation of ETL transformation with Pentaho:

Transformation1:

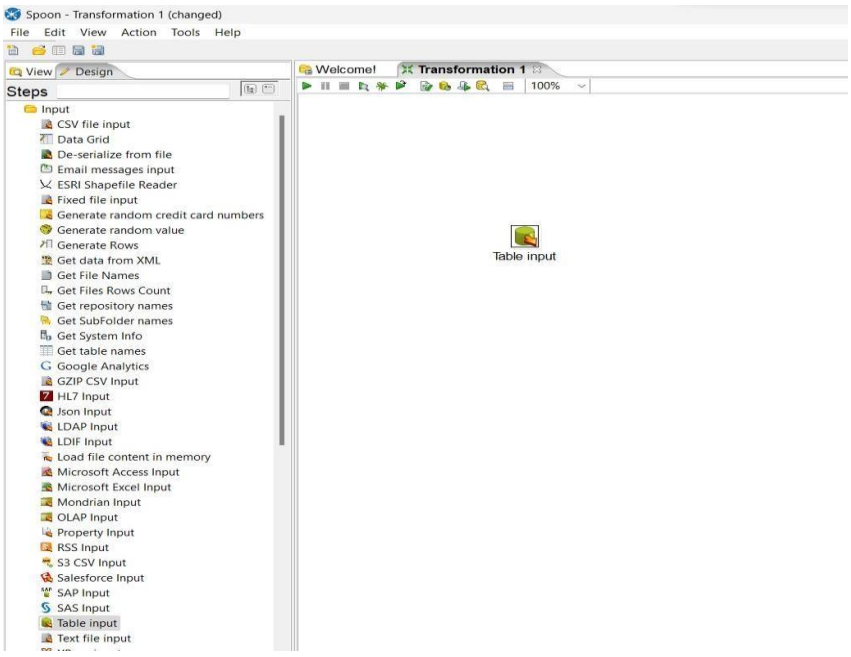
Copy Data from source to store in target

Step 1 : - In the data integration folder open “Spoon (Windows Batch File)”. Step 2 : - Go to File→New→Transformation.



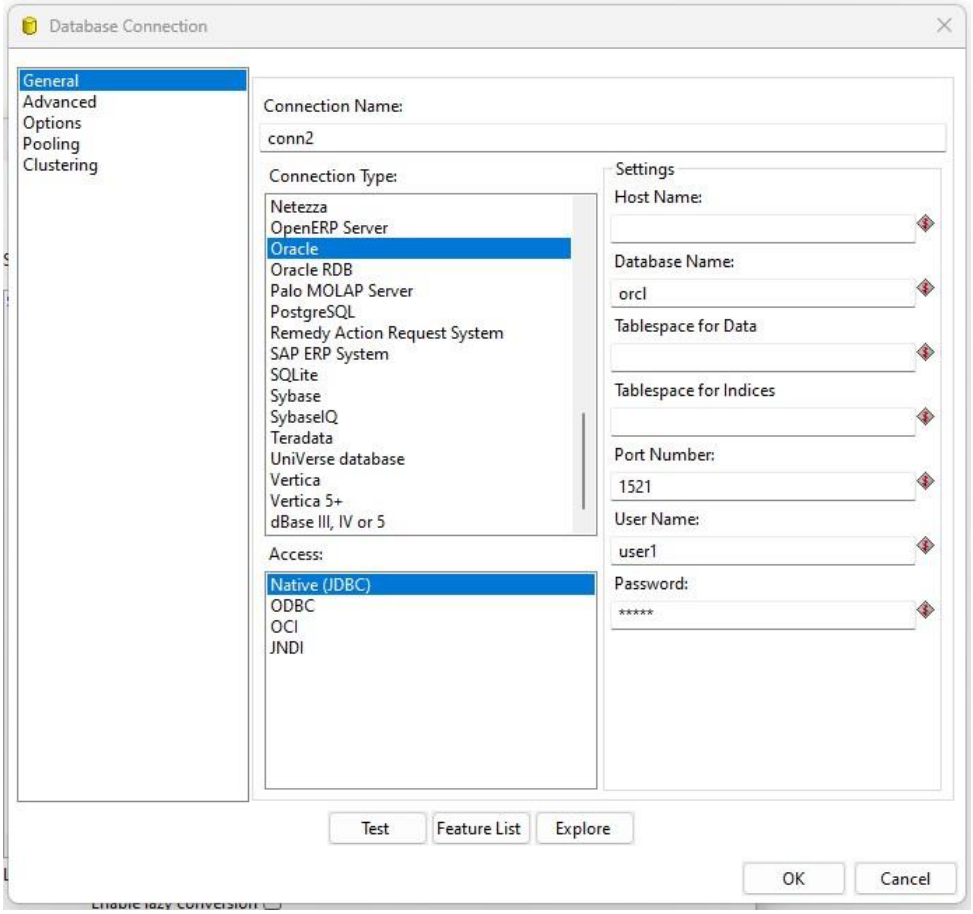
Step 3 : - Import SQL Table to Pentaho

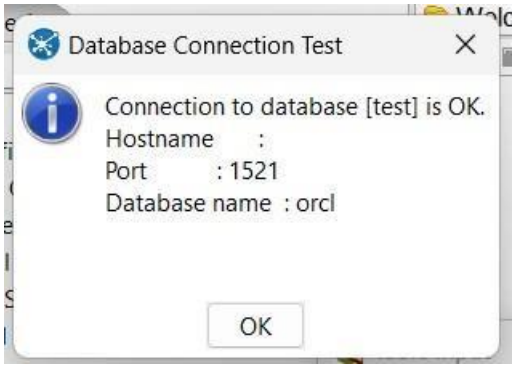
Design tab →Input folder → Drag and drop Table input

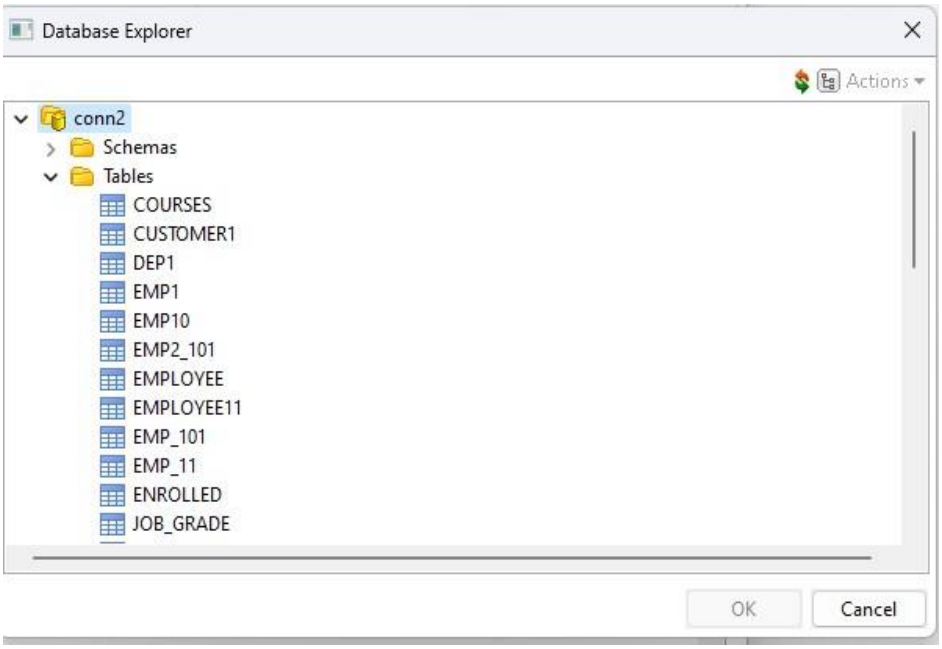


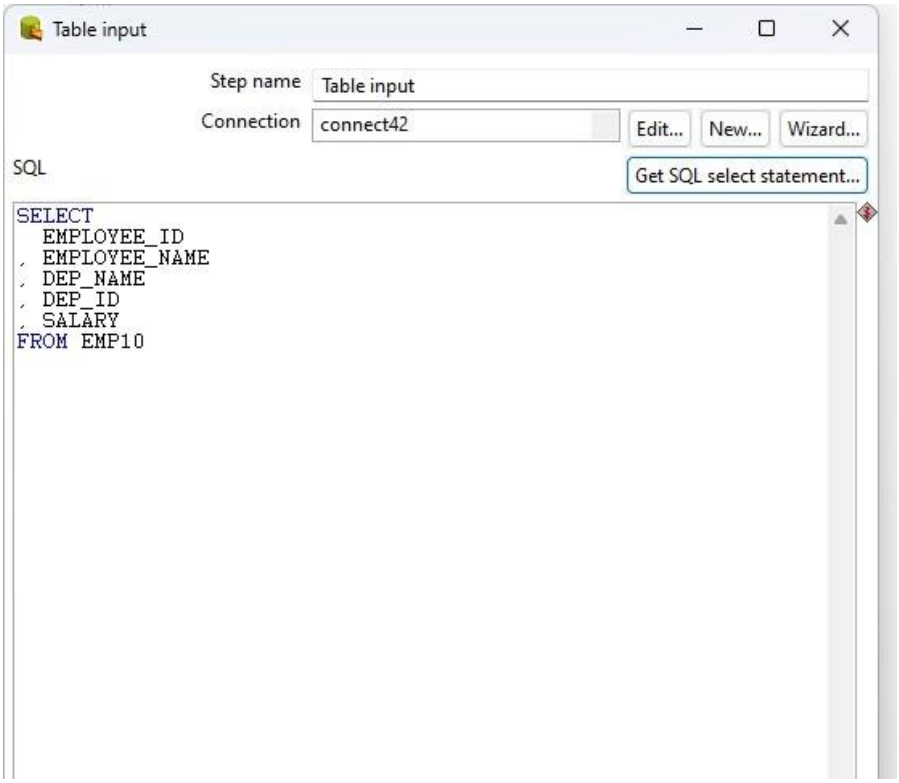
Click on new

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



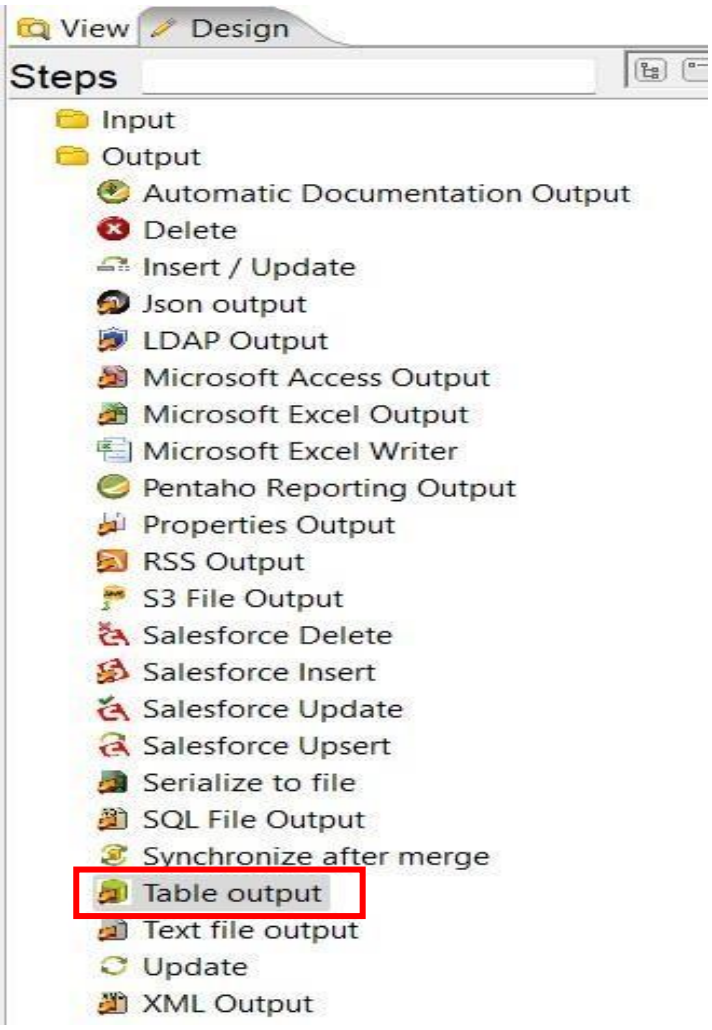






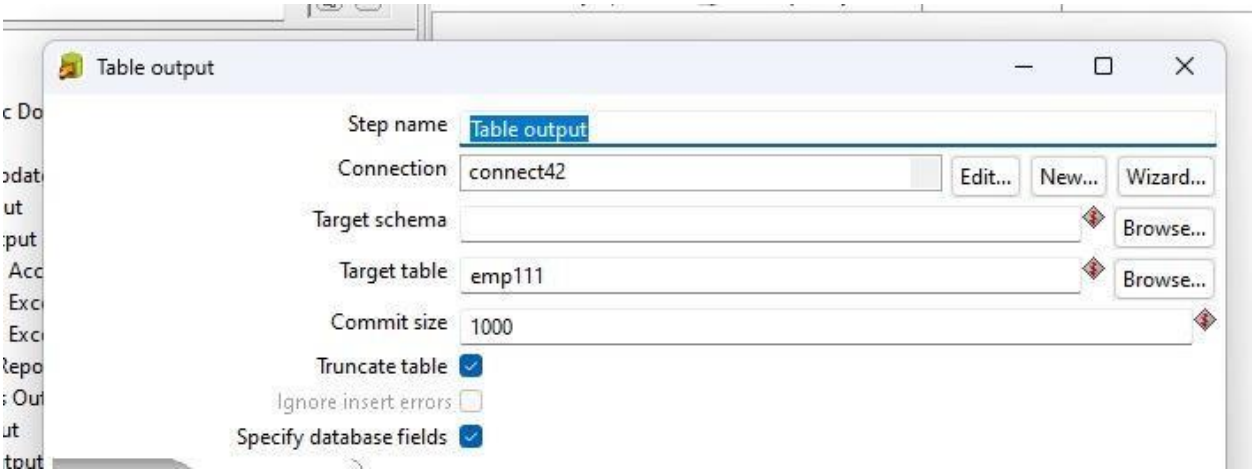
Click Close → OK

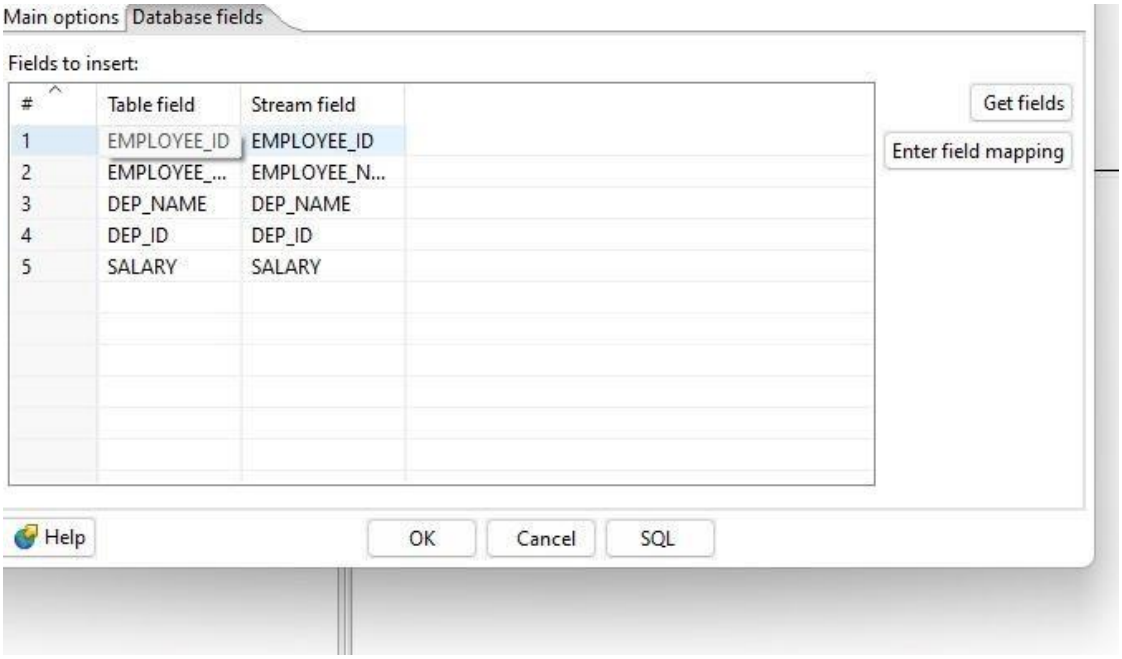
Show output: Drag and Drop Table Output



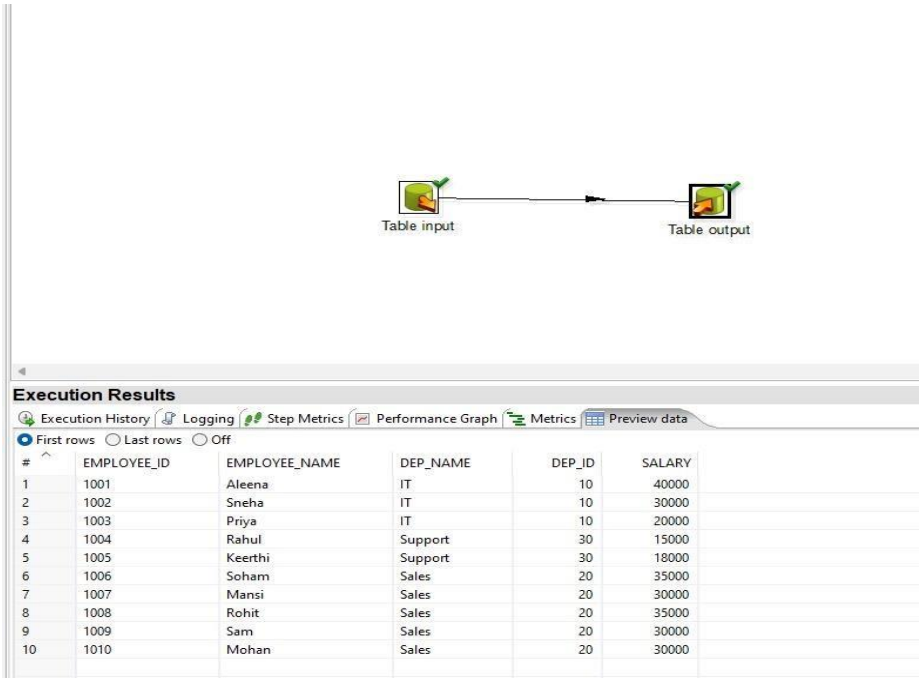
Connect input table and Output table.

Double click on Output table.



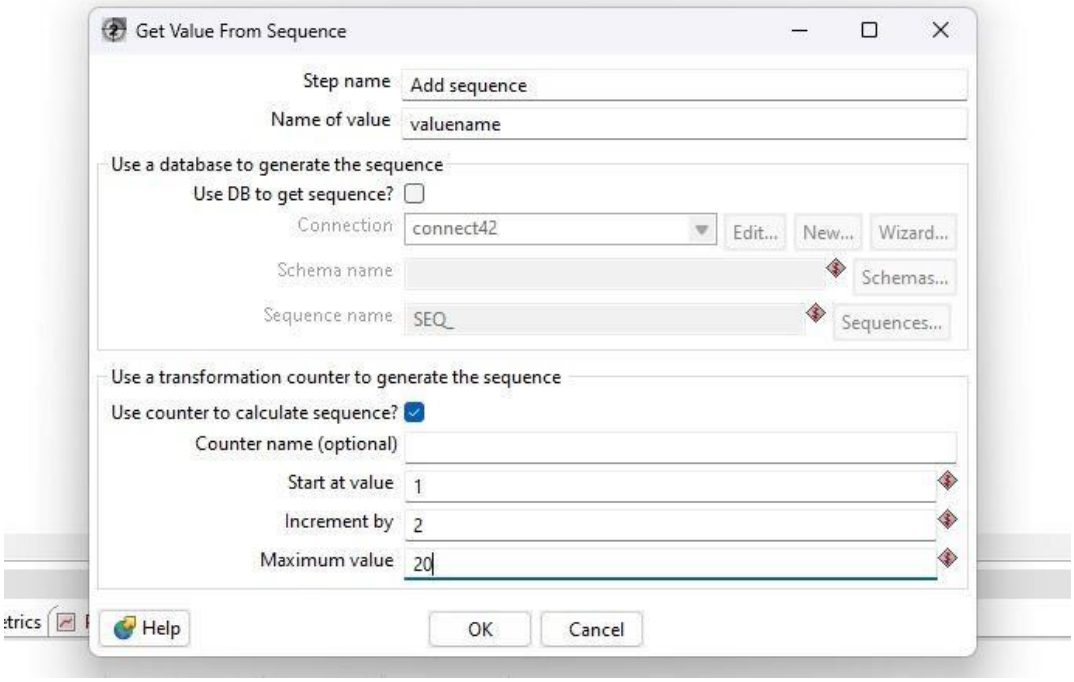


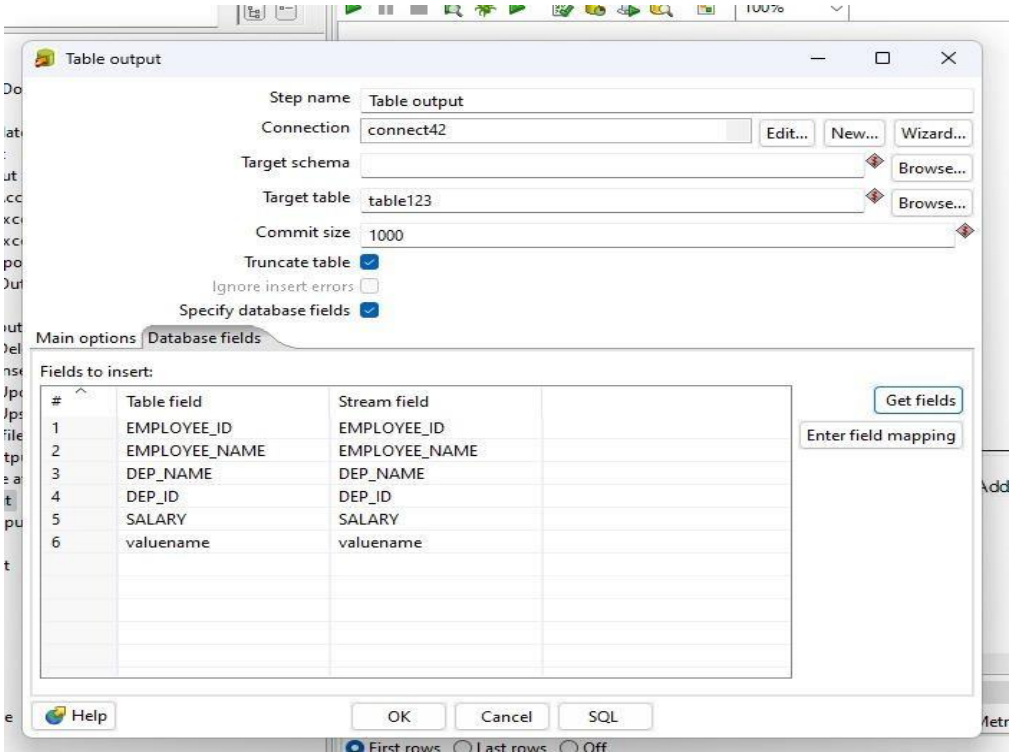
Click on SQL->EXECUTE.

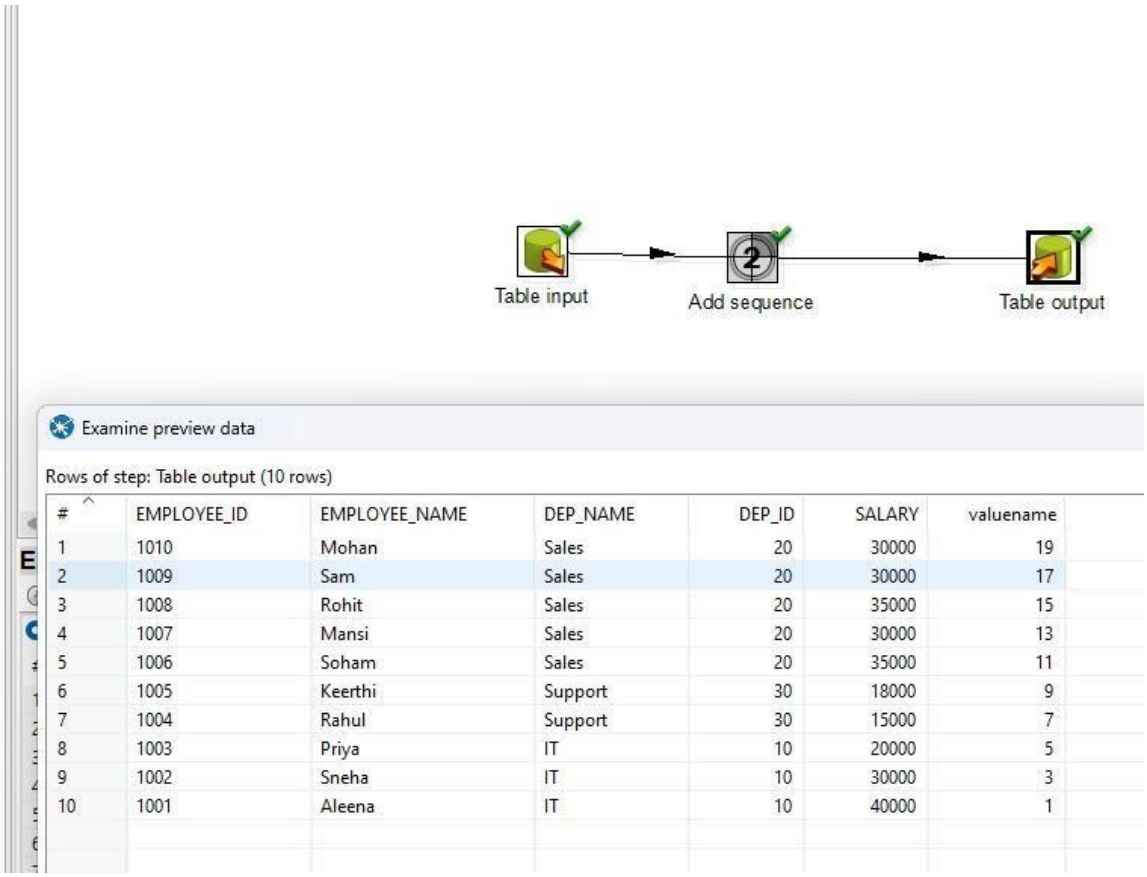


#	EMPLOYEE_ID	EMPLOYEE_NAME	DEP_NAME	DEP_ID	SALARY
1	1001	Aleena	IT	10	40000
2	1002	Sneha	IT	10	30000
3	1003	Priya	IT	10	20000
4	1004	Rahul	Support	30	15000
5	1005	Keerthi	Support	30	18000
6	1006	Soham	Sales	20	35000
7	1007	Mansi	Sales	20	30000
8	1008	Rohit	Sales	20	35000
9	1009	Sam	Sales	20	30000
10	1010	Mohan	Sales	20	30000

Transformation 2:
Adding Sequence

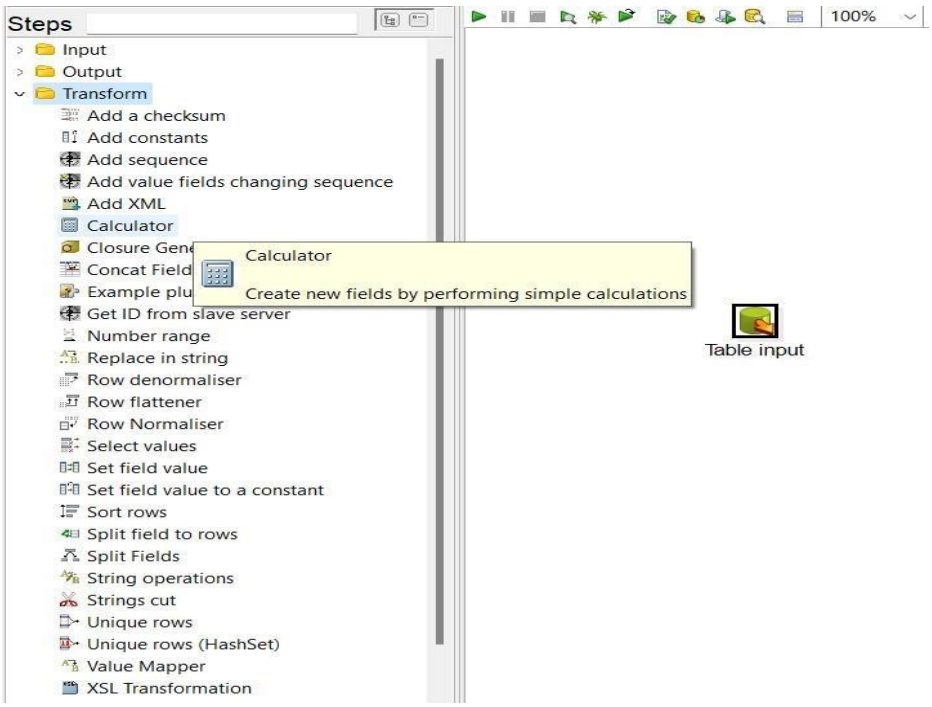


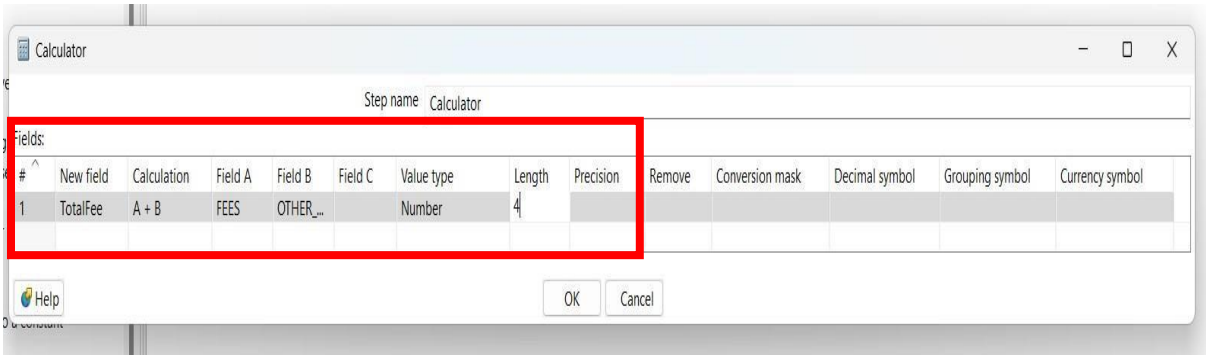


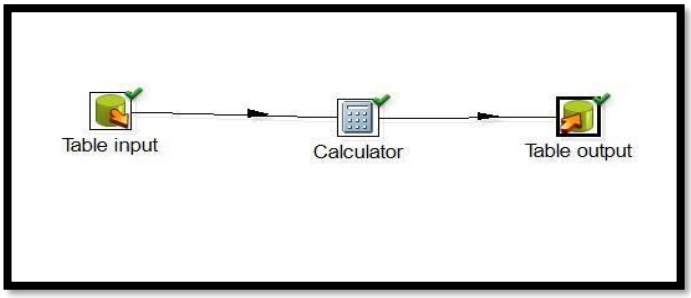


#	EMPLOYEE_ID	EMPLOYEE_NAME	DEP_NAME	DEP_ID	SALARY	valuename
1	1010	Mohan	Sales	20	30000	19
2	1009	Sam	Sales	20	30000	17
3	1008	Rohit	Sales	20	35000	15
4	1007	Mansi	Sales	20	30000	13
5	1006	Soham	Sales	20	35000	11
6	1005	Keerthi	Support	30	18000	9
7	1004	Rahul	Support	30	15000	7
8	1003	Priya	IT	10	20000	5
9	1002	Sneha	IT	10	30000	3
10	1001	Aleena	IT	10	40000	1

Transformaton 3:
Adding calculator.





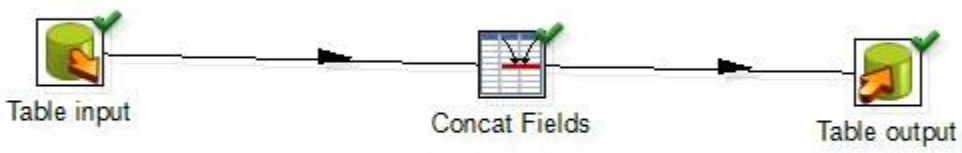


Examine preview data

Rows of step: Calculator (10 rows)

#	ROLL_NO	FNAME	LNAME	FEES	OTHER_FEES	MARKS	TotalFee
1	20	Kane	Williamson	2000	380	57	2380.00
2	19	Tonny	Stark	2000	650	84	2650.00
3	18	Tom	Lathon	2000	250	65	2250.00
4	17	Steve	Hope	2000	950	75	2950.00
5	16	Pat	Rock	2000	850	61	2850.00
6	15	John	Gross	2000	950	55	2950.00
7	14	Prithvi	Shaw	2000	650	78	2650.00
8	13	Ros	Taylor	2000	400	90	2400.00

Transformation4:
Concatenation of two fields



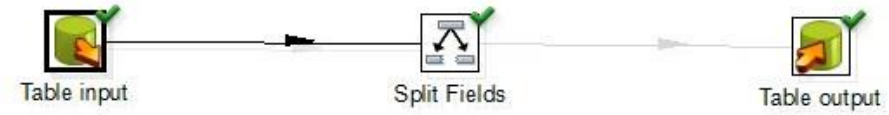
Execution Results

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

First rows | Last rows | Off

#	ROLLNO	FNAME	LNAME	FEES	MARKS	fullname
1	101	Shruti	Kamble	70000	92	Shruti ;Kamble
2	102	Sejal	Chatoriya	100000	80	Sejal ;Chatoriya
3	103	Arya	Pillai	100000	70	Arya ;Pillai
4	104	Prajakta	Jadhav	90000	82	Prajakta ;Jadhav
5	105	Tushar	Bharadwaj	60000	99	Tushar ;Bharadwaj

Transformation 5
Splitting of two fields

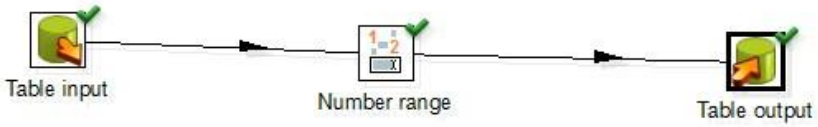


Execution History | Logging | Step Metrics | Performance Graph | Metrics

First rows | Last rows | Off

#	ROLLNO	FNAME	LNAME	FEES	MARKS
1	101	Shruti	Kamble	70000	92
2	102	Sejal	Chatoriya	100000	80
3	103	Arya	Pillai	100000	70
4	104	Prajakta	Jadhav	90000	82
5	105	Tushar	Bharadwaj	60000	99

Transformation 6
Number range



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

First rows | Last rows | Off

#	ROLLNO	FNAME	LNAME	FEES	MARKS	grade
1	101	Shruti	Kamble	70000	92	O+
2	102	Sejal	Chatoriya	100000	80	O
3	103	Arya	Pillai	100000	70	A+
4	104	Prajakta	Jadhav	90000	82	O
5	105	Tushar	Bharadwaj	60000	99	O+

Transformation 7
String Operation

First rows | Last rows | Off

#	ROLLNO	FNAME	LNAME	FEES	MARKS	first	last
1	101	Shruti	Kamble	70000	92	Shruti	KAMBLE
2	102	Sejal	Chatoriya	100000	80	Sejal	CHATORIYA
3	103	Arya	Pillai	100000	70	Arya	PILLAI
4	104	Prajakta	Jadhav	90000	82	Prajakta	JADHAV
5	105	Tushar	Bharadwaj	60000	99	Tushar	BHARADWAJ

Transformation 8
Sorting data

Rows of step: Table output (7 rows)

#	ROLLNO	FNAME	LNAME	FEES	MARKS	value
1	107	Aditya	Raj	90000	89	7
2	106	Annu	Sharma	50000	78	6
3	105	Tushar	Bharadwaj	60000	99	5
4	104	Prajakta	Jadhav	90000	82	4
5	103	Arya	Pillai	100000	70	3

Practical 5

Introduction to R, Install packages, Loading packages, Data types, checking variable type, printing variable and objects(Vector, Matrix, List, Factor, Data frame, Table)c-binding and rbinding Reading and Writing data:Setwd(), getwd(), data(),rm()Attaching and Detaching data. Reading data from the consol. Loading data from different data sources:

Introduction to R.

myString <- "Hello, World!" print
(myString)

```
> myString <- "Hello, World!"
> print ( myString)
[1] "Hello, World!"
>
```

setwd("C:/Program Files") getwd()
dir()

```
> setwd("C:/Program Files")
> getwd()
[1] "C:/Program Files"
> dir()
 [1] "Android"                      "Apache Software Foundation"
 [6] "Cisco Packet Tracer 7.3.0"    "Common Files"
[11] "Electronic Arts"              "Epic Games"
[16] "HPCommRecovery"               "HPPrintScanDoctor"
[21] "Java"                         "Malwarebytes"
[26] "Microsoft"                    "Microsoft Analysis Services"
[31] "Microsoft SQL Server"         "Microsoft Update Health Tools"
[36] "ModifiableWindowsApps"        "MongoDB"
[41] "Netease"                      "nodejs"
[46] "PPSSPP"                       "QGIS 3.8"
[51] "Uninstall Information"        "Windows Defender"
[56] "Windows Photo Viewer"         "Windows Sidebar"
```

Creating and assigning a variable.

x<-1 class(x)
print(x) x<-
'c'
is.character(

x)
is.integer(x)
y<-'2.14'
as.integer(y)

```
> x<-1
> class(x)
[1] "numeric"
> print(x)
[1] 1
> x<-'c'
> is.character(x)
[1] TRUE
> is.integer(x)
[1] FALSE
> y<-'2.14'
> as.integer(y)
[1] 2
```

Creating vectors x<-3
y<-vector("logical",length=10)
length(x) y<-c(4,5,6)
5*x

```
> x<-3
> y<-vector("logical",length=10)
> length(x)
[1] 1
> y<-c(4,5,6)
> 5*x
[1] 15
```

Creating Matrix:Two dimensional Array
m<-matrix(c(11,12,13,55,60,65,66,72,78), nrow=3,ncol=3) m
dim(m) attributes(m)

```
> m<-matrix(c(11,12,13,55,60,65,66,72,78), nrow=3,ncol=3)
> m
     [,1] [,2] [,3]
[1,]   11   55   66
[2,]   12   60   72
[3,]   13   65   78
> dim(m)
[1] 3 3
> attributes(m)
$dim
[1] 3 3
```

Retrieving values

matrix(c(11,12,13,55,60,65,66,72,78),nrow=3,ncol=3,byrow = TRUE) m
m[1,2] #first row second column value
m[2,] # second row m[,2] #second
column m[c(1,2),] #more than one row
m[,c(1,2)] #more than one column

```
> matrix(c(11,12,13,55,60,65,66,72,78),nrow=3,ncol=3,byrow = TRUE)
     [,1] [,2] [,3]
[1,]   11   12   13
[2,]   55   60   65
[3,]   66   72   78
> m
     [,1] [,2] [,3]
[1,]   11   55   66
[2,]   12   60   72
[3,]   13   65   78
> m[1,2] #first row second column value
[1] 55
> m[2,] # second row
[1] 12 60 72
> m[,2] #second column
[1] 55 60 65
> m[c(1,2),] #more than one row
     [,1] [,2] [,3]
[1,]   11   55   66
[2,]   12   60   72
> m[,c(1,2)] #more than one column
     [,1] [,2]
[1,]   11   55
[2,]   12   60
[3,]   13   65
```

cbind and rbind

x<-c(1,2,3) y<-
c(11,12,13)
cbind(x,y) rbind(x,y)

```
> x<-c(1,2,3)
> y<-c(11,12,13)
> cbind(x,y)
     x  y
[1,] 1 11
[2,] 2 12
[3,] 3 13
> rbind(x,y)
  [,1] [,2] [,3]
x    1    2    3
y   11   12   13
```

Operations on matrix. p<-3*m

p

n<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3)

q<- m+n q

o<-matrix(c(4,5,6,14,15,16),nrow=3,ncol=2) o

r<-m%*% o

r

mdash<-t(m) mdash

mdash<-t(m) mdash

```
> p<-3*m
> p
     [,1] [,2] [,3]
[1,]   33  165  198
[2,]   36  180  216
[3,]   39  195  234
> n<-matrix(c(4,5,6,14,15,16,24,25,26),nrow=3,ncol=3)
> q<- m+n
> q
     [,1] [,2] [,3]
[1,]   15   69   90
[2,]   17   75   97
[3,]   19   81  104
> o<-matrix(c(4,5,6,14,15,16),nrow=3,ncol=2)
> o
     [,1] [,2]
[1,]    4   14
[2,]    5   15
[3,]    6   16
> r<-m%*% o
> r
     [,1] [,2]
[1,]  715 2035
[2,]  780 2220
[3,]  845 2405
> mdash<-t(m)
> mdash
     [,1] [,2] [,3]
[1,]   11   12   13
[2,]   55   60   65
[3,]   66   72   78
> mdash<-t(m)
> mdash
     [,1] [,2] [,3]
[1,]   11   12   13
[2,]   55   60   65
[3,]   66   72   78
```

Determinant

s<-matrix(c(4,5,6,14,15,16,24,25,26), nrow=3,ncol=3,byrow=TRUE)
s_det<-det(s) s_det

```
> s<-matrix(c(4,5,6,14,15,16,24,25,26), nrow=3,ncol=3,byrow=TRUE)
> s_det<-det(s)
> s_det
[1] 1.110223e-14
```

List: thislist <- list("apple", "banana",
"cherry") thislist

thislist <- list("apple", "banana", "cherry")
thislist[1] <- "mango" thislist length(thislist)
"apple" %in% thislist append(thislist,
"orange") append(thislist, "kiwi", after = 2)
thislist
thislist <- list("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
(thislist)[2:5]

list1 <- list("a", "b", "c") list2
<- list(1,2,3)
list3 <- c(list1,list2) #use c function to combine two list list3

```
> thislist <- list("apple", "banana", "cherry")
> thislist
[[1]]
[1] "apple"

[[2]]
[1] "banana"

[[3]]
[1] "cherry"

> thislist <- list("apple", "banana", "cherry")
> thislist[1] <- "mango"
> thislist
[[1]]
[1] "mango"

[[2]]
[1] "banana"

[[3]]
[1] "cherry"

> length(thislist)
[1] 3
> "apple" %in% thislist
[1] FALSE
```

```
> append(thislist, "orange")
[[1]]
[1] "mango"

[[2]]
[1] "banana"

[[3]]
[1] "cherry"

[[4]]
[1] "orange"

> append(thislist, "kiwi", after = 2)
[[1]]
[1] "mango"

[[2]]
[1] "banana"

[[3]]
[1] "kiwi"

[[4]]
[1] "cherry"

> thislist
[[1]]
[1] "mango"

[[2]]
[1] "banana"

[[3]]
[1] "cherry"
```

```
> thislist <- list("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
> (thislist)[2:5]
[[1]]
[1] "banana"

[[2]]
[1] "cherry"

[[3]]
[1] "orange"

[[4]]
[1] "kiwi"

> list1 <- list("a", "b", "c")
> list2 <- list(1,2,3)
> list3 <- c(list1,list2) #use c function to combine two list
> list3
[[1]]
[1] "a"

[[2]]
[1] "b"

[[3]]
[1] "c"

[[4]]
[1] 1

[[5]]
[1] 2

[[6]]
[1] 3
```

Creating data frame: student_id<-c(1,2,3)
student_names<-c("Yash","Abhi","Riya") position<-
c("First","Second","Third")
data<-data.frame(student_id,student_names,position)
data data$student_id nrow(data) ncol(data)
names(data)

```
> student_id<-c(1,2,3)
> student_names<-c("Aleena","Abhi","Riya")
> position<-  c("First","Second","Third")
> data<-data.frame(student_id,student_names,position)
> data
  student_id student_names position
1          1        Aleena    First
2          2          Abhi   Second
3          3          Riya    Third
> data$student_id
[1] 1 2 3
> nrow(data)
[1] 3
> ncol(data)
[1] 3
> names(data)
[1] "student_id"    "student_names" "position"
```

Table command

smoke <- matrix(c(51,43,22,92,28,21,68,22,9),ncol=3,byrow=TRUE)
colnames(smoke) <- c("High","Low","Middle") rownames(smoke) <-
c("current","former","never") smoke <- as.table(smoke) smoke

```
> smoke <- matrix(c(51,43,22,92,28,21,68,22,9),ncol=3,byrow=TRUE)
> colnames(smoke) <- c("High","Low","Middle")
> rownames(smoke) <- c("current","former","never")
> smoke <- as.table(smoke)
> smoke
        High Low Middle
current   51  43     22
former    92  28     21
never     68  22      9
```

Practical 6

Implementation of Data preprocessing techniques in R:

1. Naming and Renaming variables
2. adding a new variable.
3. Dealing with missing data.
4. Dealing with categorical data.
5. Data reduction using sub setting

```
data1<-mtcars head(data1,5)
install.packages("dplyr")
data1=rename(data1,horse_power=hp)
data1
data1$new_hp1<-data1$horse_power*0.5
colnames(data1) data1
```

```
> data1<-mtcars
> head(data1,5)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
> data1=rename(data1,horse_power=hp)
> data1
                   mpg cyl  disp horse_power drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0         110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0         110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0          93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0         110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0         175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0         105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0         245 3.21 3.570 15.84  0  0    3    4
> data1$new_hp1<-data1$horse_power*0.5
> colnames(data1)
 [1] "mpg"        "cyl"        "disp"       "horse_power" "drat"
 [6] "wt"         "qsec"       "vs"         "am"          "gear"
[11] "carb"       "new_hp1"
> data1
                   mpg cyl  disp horse_power drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0         110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0         110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0          93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0         110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0         175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0         105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0         245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7          62 3.69 3.190 20.00  1  0    4    2
```

data2=read.table("C:/Rpractical/missing.csv",sep=",") data2

data2=read.csv(file="C:/Rpractical/missing.csv",col.names=c("s.no","name", "salary","DoJ","desg")) data2

```
> data2=read.table("C:/Rpractical/missing.csv",sep=",")
> data2
   V1       V2      V3         V4         V5
1   1     Rick  623.30 01/01/2012         IT
2   2      Dan  515.20 23/09/2013 Operations
3   3 Michelle  611.00 15/11/2014         IT
4   4     Ryan  729.00 11/05/2014         HR
5  NA     Gary  843.25 27/03/2015    Finance
6   6     Nina      NA 21/05/2013         IT
7   7    Simon  632.80 30/07/2013 Operations
8   8     Guru  722.50 17/06/2014    Finance
9   9     John      NA 21/05/2012
10 10     Rock  600.80 30/07/2013         HR
11 11     Brad 1032.80 30/07/2013 Operations
12 12     Ryan  729.00 11/05/2014         HR
> data2=read.csv(file="C:/Rpractical/missing.csv",col.names=c("s.no","name","salary","Do
J","desg"))
> data2
   s.no     name  salary        DoJ       desg
1     2      Dan  515.20 23/09/2013 Operations
2     3 Michelle  611.00 15/11/2014         IT
3     4     Ryan  729.00 11/05/2014         HR
4    NA     Gary  843.25 27/03/2015    Finance
5     6     Nina      NA 21/05/2013         IT
6     7    Simon  632.80 30/07/2013 Operations
7     8     Guru  722.50 17/06/2014    Finance
8     9     John      NA 21/05/2012
9    10     Rock  600.80 30/07/2013         HR
10   11     Brad 1032.80 30/07/2013 Operations
11   12     Ryan  729.00 11/05/2014         HR
```

Imputation: install.packages("Hmisc")

x=c(1,2,3,NA,4) x<-impute(x,fun=mean)

x

x<-impute(x,fun=median)

x

```
> library(Hmisc)

Attaching package: 'Hmisc'

The following objects are masked from 'package:dplyr':

    src, summarize

The following objects are masked from 'package:base':

    format.pval, units

> x=c(1,2,3,NA,4)
> x<-impute(x,fun=mean)
> x
   1    2    3    4    5
 1.0  2.0  3.0 2.5*  4.0
> x<-impute(x,fun=median)
> x
   1    2    3    4    5
 1.0  2.0  3.0 2.5*  4.0
```

Categorical Data: gender_vector<-c("Male","Female","Female","Male")
class(gender_vector)
factor_gender<-factor(gender_vector) class(factor_gender)
day<-c("evening","morning","night","midday") factor_day<-
factor(day,order=TRUE,levels=c("evening","morning","night","midday"))
factor_day

```
> gender_vector<-c("Male","Female","Female","Male")
> class(gender_vector)
[1] "character"
> factor_gender<-factor(gender_vector)
> class(factor_gender)
[1] "factor"
> day<-c("evening","morning","night","midday")
> factor_day<-factor(day,order=TRUE,levels=c("evening","morning","night","midday"))
> factor_day
[1] evening morning night   midday
Levels: evening < morning < night < midday
```

Create numeric to factor age<-c(20,30,35,45,55)
salary<-c(10000,25000,45000,24000,20000) gender<-
c("male","female","male","male","female") emp<-data.frame(age,salary,gender)
emp
#cut func-used to divide numeric vector into different ranges wfact<-
cut(emp$age,3,labels=c("young","medium","aged")) table(wfact)

```
> age<-c(20,30,35,45,55)
> salary<-c(10000,25000,45000,24000,20000)
> gender<-c("male","female","male","male","female")
> emp<-data.frame(age,salary,gender)
> emp
  age salary gender
1  20  10000   male
2  30  25000 female
3  35  45000   male
4  45  24000   male
5  55  20000 female
> #cut func-used to divide numeric vector into different ranges
> wfact<-cut(emp$age,3,labels=c("young","medium","aged"))
> table(wfact)
wfact
 young medium   aged
     2      1      2
```

Practical 7

Implementation and analysis of Linear regression through graphical methods:

data<-mtcars data

```
> data<-mtcars
> data
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
```

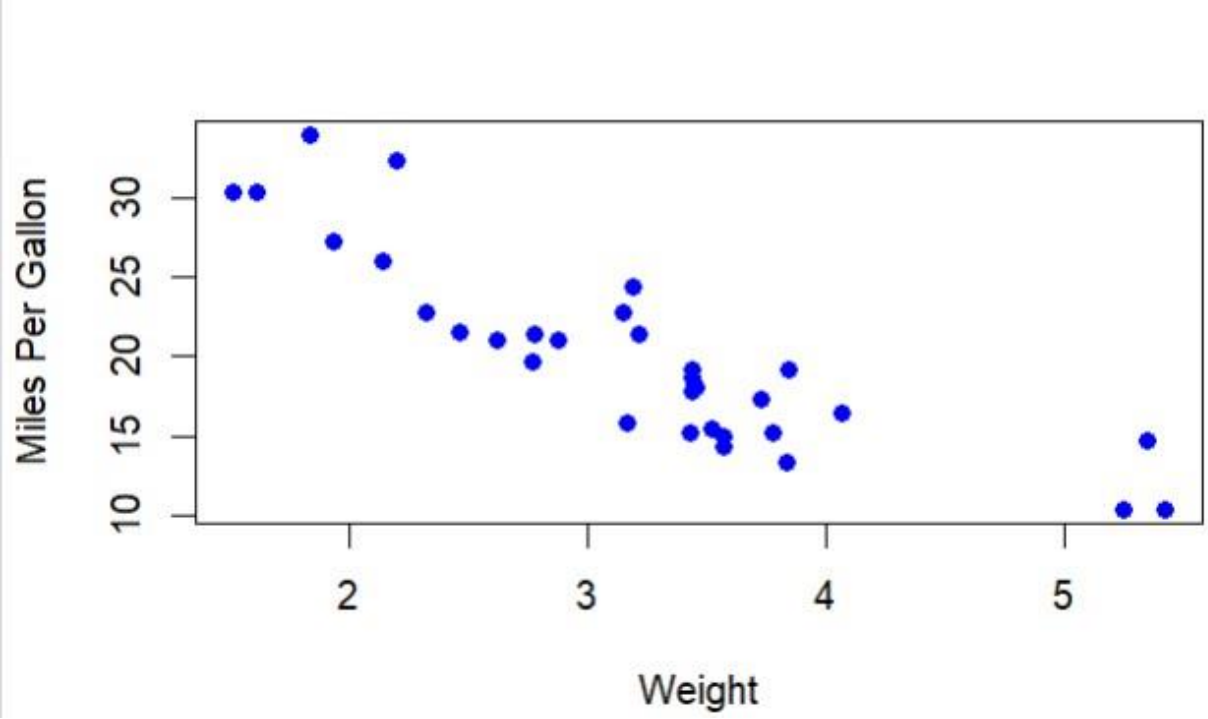
head(data)

model <- lm(mpg ~ wt, data = mtcars) summary(model)

```
> head(data)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> model <- lm(mpg ~ wt, data = mtcars)
```

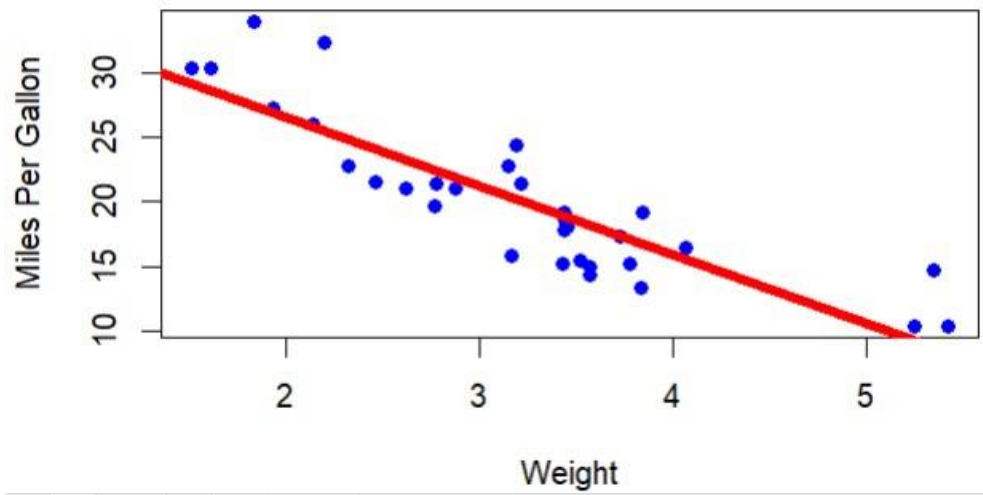
```
> summary(model)

Call:
lm(formula = mpg ~ wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10

> plot(mtcars$wt, mtcars$mpg, xlab = "Weight", ylab = "Miles Per Gallon", pch = 16, col = "blue")
>
```

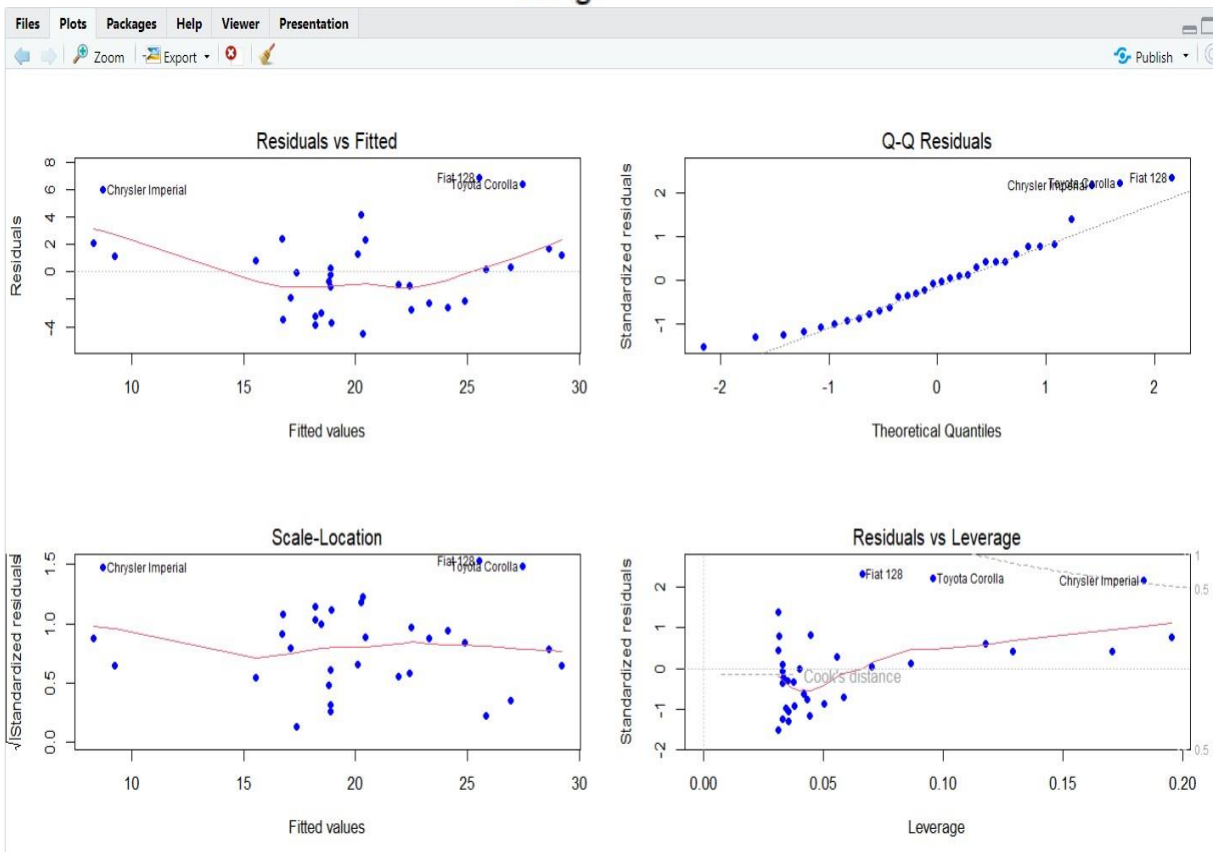
plot(mtcars$wt, mtcars$mpg, xlab = "Weight", ylab = "Miles Per Gallon", pch = 16, col = "blue")



abline(model, col = "red", lwd = 5) par(mfrow = c(2, 2))

```
plot(model, pch = 16, col = "blue") par(mfrow =
c(1, 1))

plot(model, pch = 16, col = "blue")
```





Practical 8

Implementation and Analysis Classification algorithms like Naïve Bayesian, K-Nearest Neighbour, ID3, C4.5:

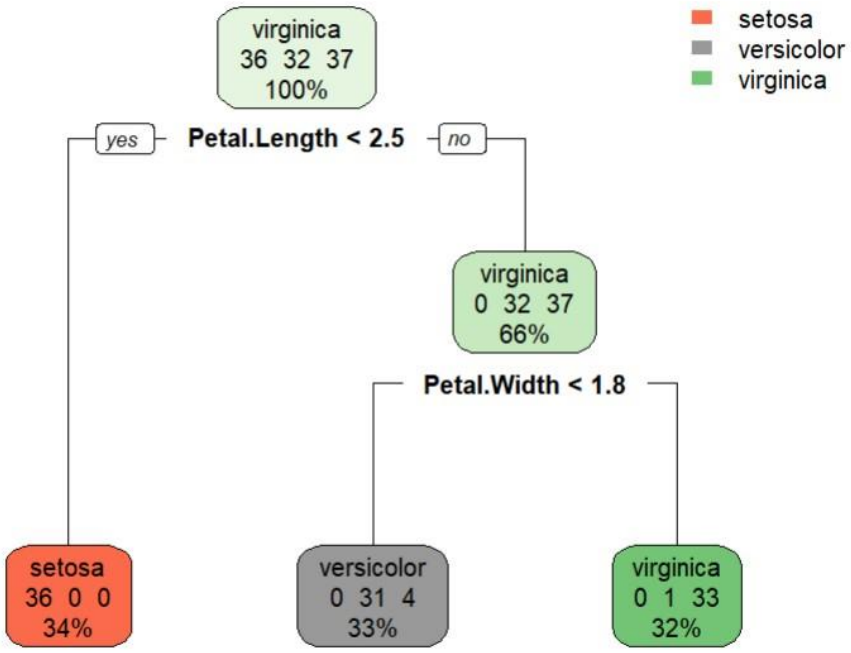
```
library(class)
library(e1071)
library(rpart)
library(rpart.plot)
library(C50)
library(ggplot2)
library(ROCR)  data(iris)
set.seed(123)
##decision tree
train_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[train_indices, ]  test_data <- iris[-
train_indices, ]
dt_model <- rpart(Species ~ ., data = train_data, method = "class")
dt_pred <- predict(dt_model, test_data, type = "class")  dt_accuracy <-
sum(dt_pred == test_data$Species) / grow(test_data)  cat("Decision
Tree Accuracy:", dt_accuracy, "\n")
```

```
> data(iris)
> set.seed(123)
> ##decision tree
> train_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
> train_data <- iris[train_indices, ]
> test_data <- iris[-train_indices, ]
> dt_model <- rpart(Species ~ ., data = train_data, method = "class")
> dt_pred <- predict(dt_model, test_data, type = "class")
> dt_accuracy <- sum(dt_pred == test_data$Species) / nrow(test_data)
> cat("Decision Tree Accuracy:", dt_accuracy, "\n")
Decision Tree Accuracy: 0.9777778
```

ID 3 :

```
id3_model <- rpart(Species ~ ., data = train_data, method = "class")

id3_pred <- predict(id3_model, test_data, type = "class")  id3_accuracy <-
sum(id3_pred == test_data$Species) / nrow(test_data)
cat("ID3 Accuracy:", id3_accuracy, "\n")
rpart.plot(id3_model, type = 2, extra = 101)
```

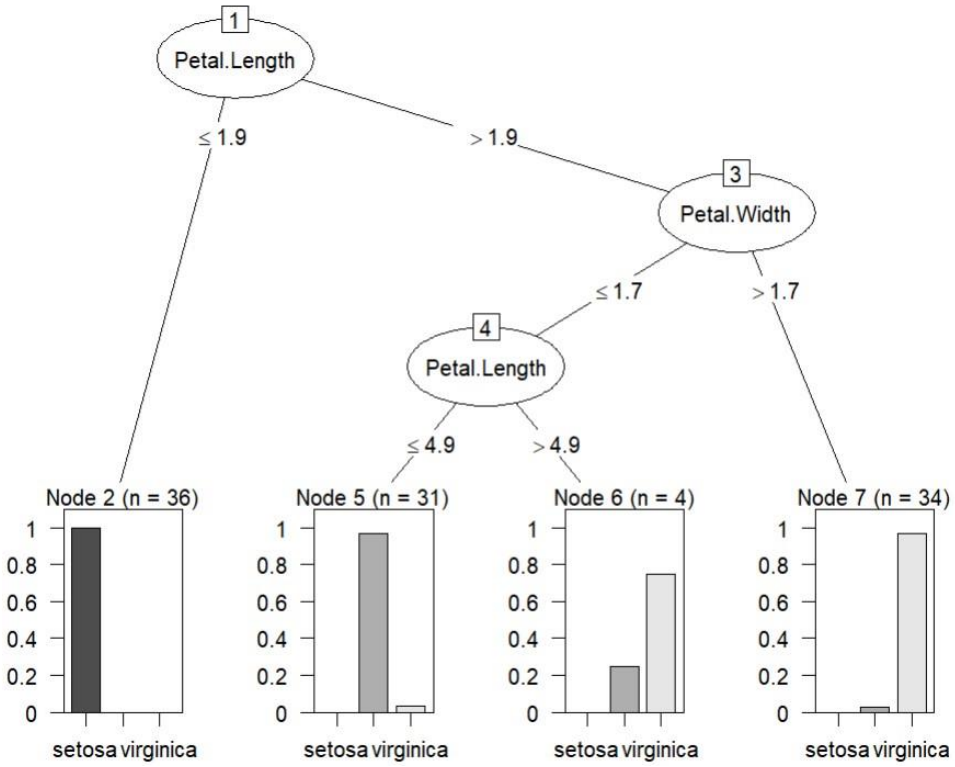
```
>   id3_model <- rpart(Species ~ ., data = train_data, method = "class")
> id3_pred <- predict(id3_model, test_data, type = "class")
> id3_accuracy <- sum(id3_pred == test_data$Species) / nrow(test_data)
> cat("ID3 Accuracy:", id3_accuracy, "\n")
ID3 Accuracy: 0.9777778
> rpart.plot(id3_model, type = 2, extra = 101)
```



C4.5

c45_model <- C5.0(train_data[, -5], train_data$Species) c45_pred <- predict(c45_model, newdata = test_data[, -5]) c45_accuracy <- sum(c45_pred == test_data$Species) / nrow(test_data) cat("C4.5 Accuracy:", c45_accuracy, "\n")
print(summary) plot(c45_model)

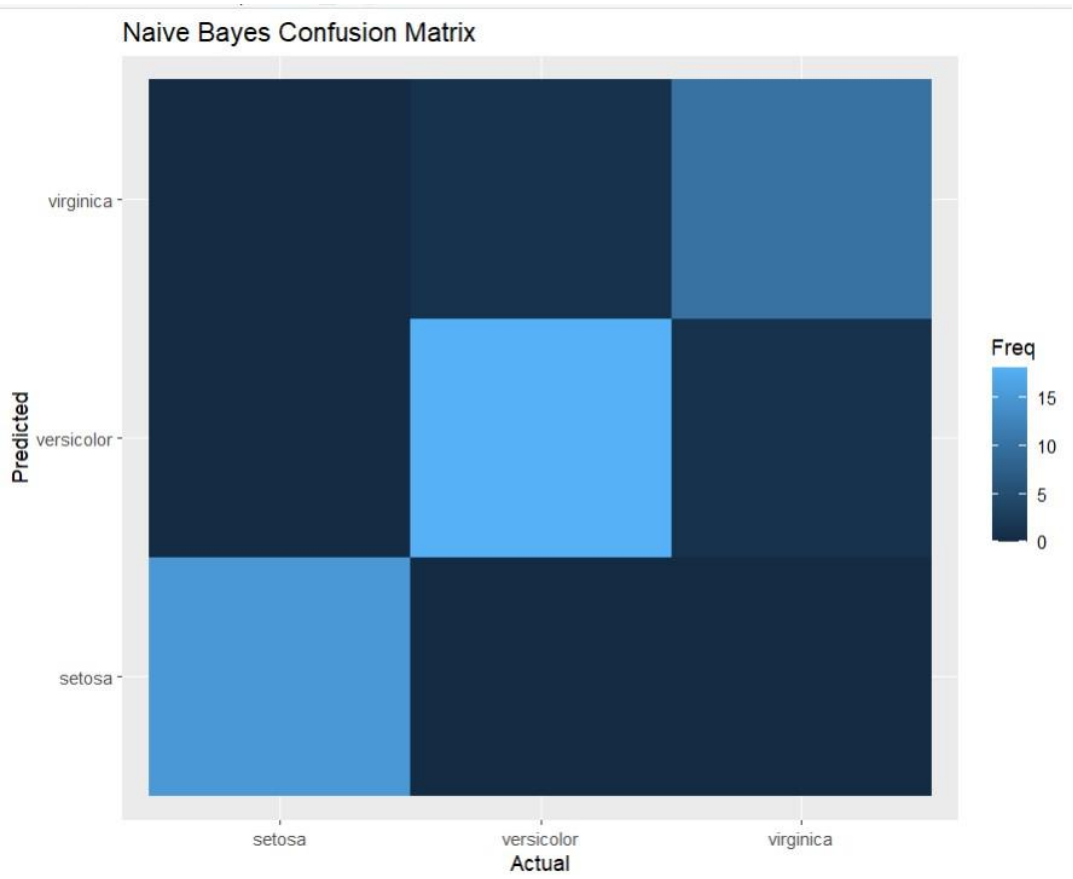
```
> c45_model <- C5.0(train_data[, -5], train_data$Species)
> c45_pred <- predict(c45_model, newdata = test_data[, -5])
> c45_accuracy <- sum(c45_pred == test_data$Species) / nrow(test_data)
> cat("C4.5 Accuracy:", c45_accuracy, "\n")
C4.5 Accuracy: 0.9777778
```



Naive bayes theorem

```
nb_model <- naiveBayes(Species ~ ., data = train_data)
train_indices <- sample(1:nrow(iris), 0.7 * nrow(iris))
train_data <- iris[train_indices, ]  test_data <- iris[-
train_indices, ]
 nb_pred <- predict(nb_model, test_data[, -5])
 nb_accuracy <- sum(nb_pred == test_data$Species) / nrow(test_data)
cat("Naive Bayes Accuracy:", nb_accuracy, "\n")  summary(nb_model)
nb_table <- table(Actual = test_data$Species, Predicted = nb_pred)  nb_table
 ggplot(as.data.frame.table(nb_table), aes(x = Actual, y = Predicted, fill = Freq)) +
geom_tile() + labs(title = "Naive Bayes Confusion Matrix", x = "Actual", y =
"Predicted")
```

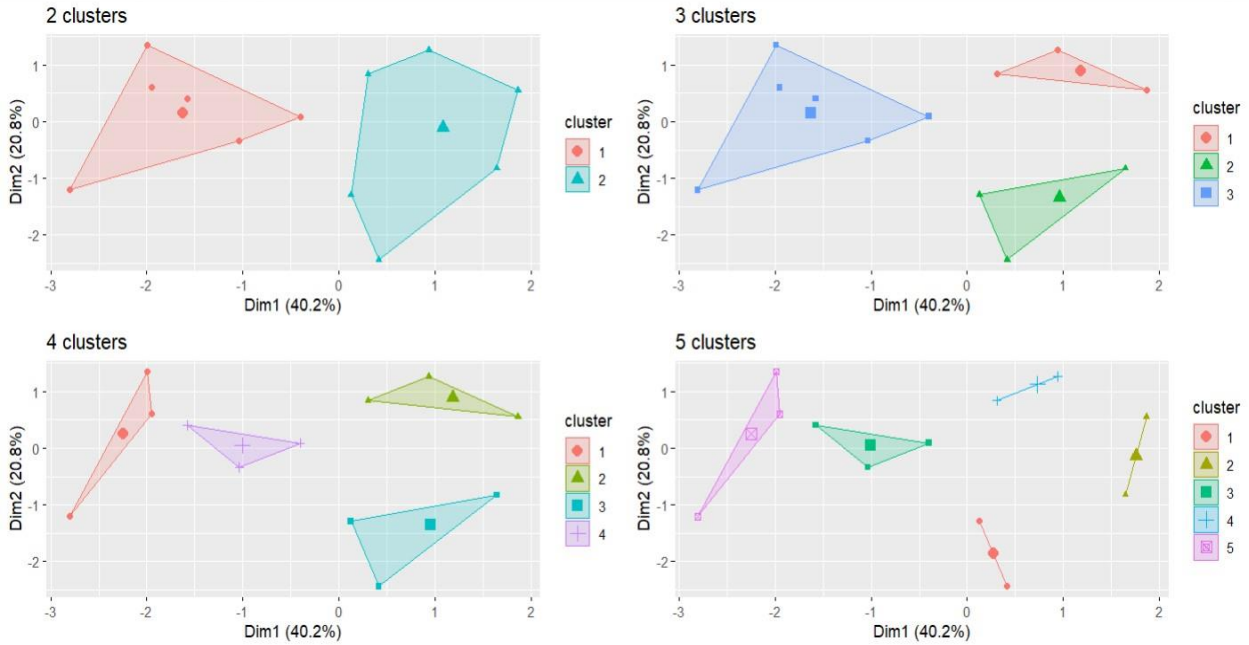
```
> train_data <- iris[train_indices, ]
> test_data <- iris[-train_indices, ]
> nb_pred <- predict(nb_model, test_data[, -5])
> nb_accuracy <- sum(nb_pred == test_data$Species) / nrow(test_data)
> cat("Naive Bayes Accuracy:", nb_accuracy, "\n")
Naive Bayes Accuracy: 0.9555556
> summary(nb_model)
          Length Class  Mode
apriori   3      table  numeric
tables    4      -none- list
levels    3      -none- character
isnumeric 4      -none- logical
call      4      -none- call
> nb_table <- table(Actual = test_data$Species, Predicted = nb_pred)
> nb_table
            Predicted
Actual       setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         18         1
  virginica       0          1        10
> ggplot(as.data.frame.table(nb_table), aes(x = Actual, y = Predicted, fill = Freq)) + geom_tile() + labs(title =
"Naive Bayes Confusion Matrix", x = "Actual", y = "Predicted")
```



KNN

install.packages("cluster")
install.packages("factoextra")
install.packages("gridextra") library(cluster)
data<-animals data<-na.omit(data) head(data,n=10)
kn<-kmeans(data,centers=2,nstart=25) kn2<-kmeans(data,centers=3,nstart=25)
kn3<-kmeans(data,centers=4,nstart=25) kn4<-kmeans(data,centers=5,nstart=25)
plot1<-fviz_cluster(kn,geom="point",data=data)+ggtitle("2 clusters") plot2<-
fviz_cluster(kn2,geom="point",data=data)+ggtitle("3 clusters") plot3<-
fviz_cluster(kn3,geom="point",data=data)+ggtitle("4 clusters") plot4<-
fviz_cluster(kn4,geom="point",data=data)+ggtitle("5 clusters") library(gridExtra)
grid.arrange(plot1,plot2,plot3,plot4,nrow=2)

```
> data<-animals
> data<-na.omit(data)
> head(data,n=10)
    war fly ver end gro hai
ant   1   1   1   1   2   1
bee   1   2   1   1   2   2
cat   2   1   2   1   1   2
cpl   1   1   1   1   1   2
chi   2   1   2   2   2   2
cow   2   1   2   1   2   2
duc   2   2   2   1   2   1
eag   2   2   2   2   1   1
ele   2   1   2   2   2   1
fly   1   2   1   1   1   1
> kn<-kmeans(data,centers=2,nstart=25)
> kn2<-kmeans(data,centers=3,nstart=25)
> kn3<-kmeans(data,centers=4,nstart=25)
> kn4<-kmeans(data,centers=5,nstart=25)
> plot1<-fviz_cluster(kn,geom="point",data=data)+ggtitle("2 clusters")
> plot2<-fviz_cluster(kn2,geom="point",data=data)+ggtitle("3 clusters")
> plot3<-fviz_cluster(kn3,geom="point",data=data)+ggtitle("4 clusters")
> plot4<-fviz_cluster(kn4,geom="point",data=data)+ggtitle("5 clusters")
> library(gridExtra)
> grid.arrange(plot1,plot2,plot3,plot4,nrow=2)
```


2 clusters
3 clusters
4 clusters
5 clusters
Dim1 (40.2%)
Dim2 (20.8%)
cluster

Practical 9

Implementation and analysis of Apriori Algorithm using Market Basket:

Analysis. setwd<-("C:/Rpractical") getwd()
mba_data<-read.csv("C:/Rpractical/data_apriori.csv") mba_data

```
> setwd<-("C:/Rpractical")
> getwd()
[1] "C:/Program Files"
> mba_data<-read.csv("C:/Rpractical/data_apriori.csv")
> mba_data
   Customer_Id  Products
1            1     bread
2            1    butter
3            1      eggs
4            1      milk
5            4     bread
6            4    butter
7            4      eggs
8            4      milk
9            4      soda
10           2      beer
11           2     bread
12           2    cheese
13           2     chips
14           2      mayo
15           2      soda
16           3     bread
17           3    butter
```

trans<-split(mba_data$Product,mba_data$Customer_Id,"transactions") trans

```
> trans<-split(mba_data$Product,mba_data$Customer_Id,"transactions")
> trans
$`1`
[1] "bread"  "butter" "eggs"   "milk"

$`2`
[1] "beer"   "bread"  "cheese" "chips"  "mayo"   "soda"

$`3`
[1] "bread"   "butter"  "eggs"    "milk"    "oranges"

$`4`
[1] "bread"  "butter" "eggs"   "milk"   "soda"

$`5`
[1] "buns"    "chips"   "beer"    "mustard" "pickels" "soda"

$`6`
[1] "bread"     "butter"    "chocolate" "eggs"      "milk"

$`7`
[1] "banana"    "chocolate" "eggs"      "milk"      "oranges"
```

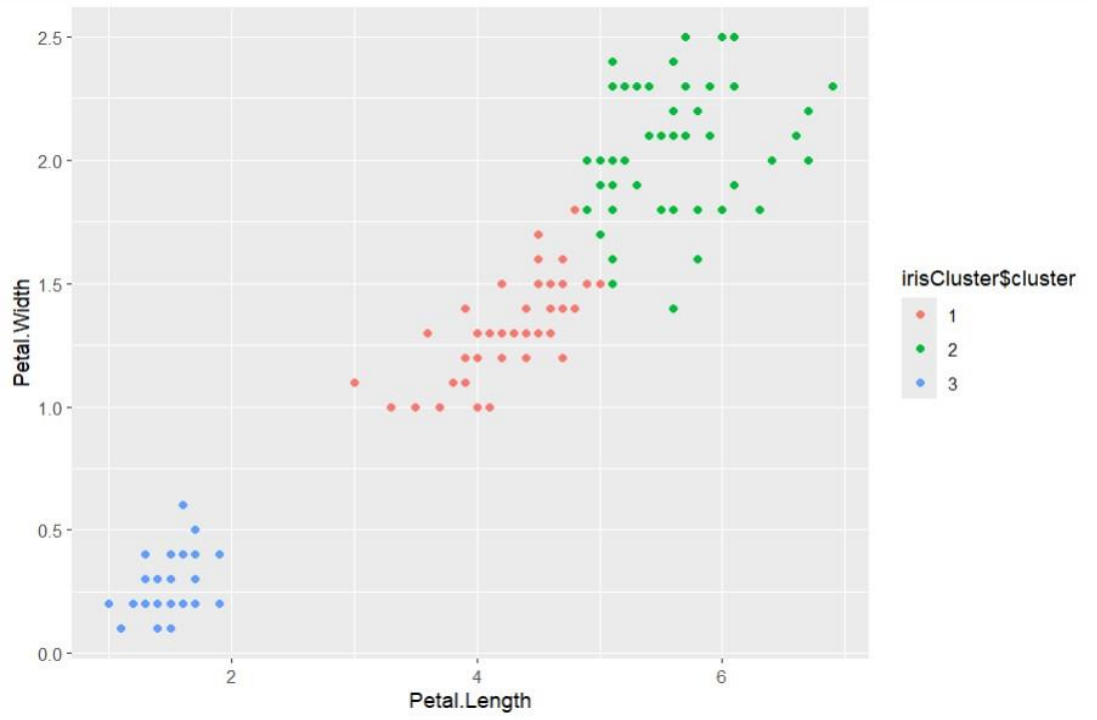
rules=apriori(trans,parameter = list(support=0.5,confidence=0.9,maxlen=3,minlen=2)) inspect(rules)

```
> inspect(rules)
     lhs               rhs      support   confidence coverage  lift     count
[1]  {eggs}         => {milk}   0.6000000 1          0.6000000 1.666667 9
[2]  {milk}         => {eggs}   0.6000000 1          0.6000000 1.666667 9
[3]  {butter}       => {bread}  0.6000000 1          0.6000000 1.250000 9
[4]  {butter, eggs} => {milk}   0.5333333 1          0.5333333 1.666667 8
[5]  {butter, milk} => {eggs}   0.5333333 1          0.5333333 1.666667 8
[6]  {bread, eggs}  => {milk}   0.5333333 1          0.5333333 1.666667 8
[7]  {bread, milk}  => {eggs}   0.5333333 1          0.5333333 1.666667 8
[8]  {butter, eggs} => {bread}  0.5333333 1          0.5333333 1.250000 8
[9]  {bread, eggs}  => {butter} 0.5333333 1          0.5333333 1.666667 8
[10] {butter, milk} => {bread}  0.5333333 1          0.5333333 1.250000 8
[11] {bread, milk}  => {butter} 0.5333333 1          0.5333333 1.666667 8
```

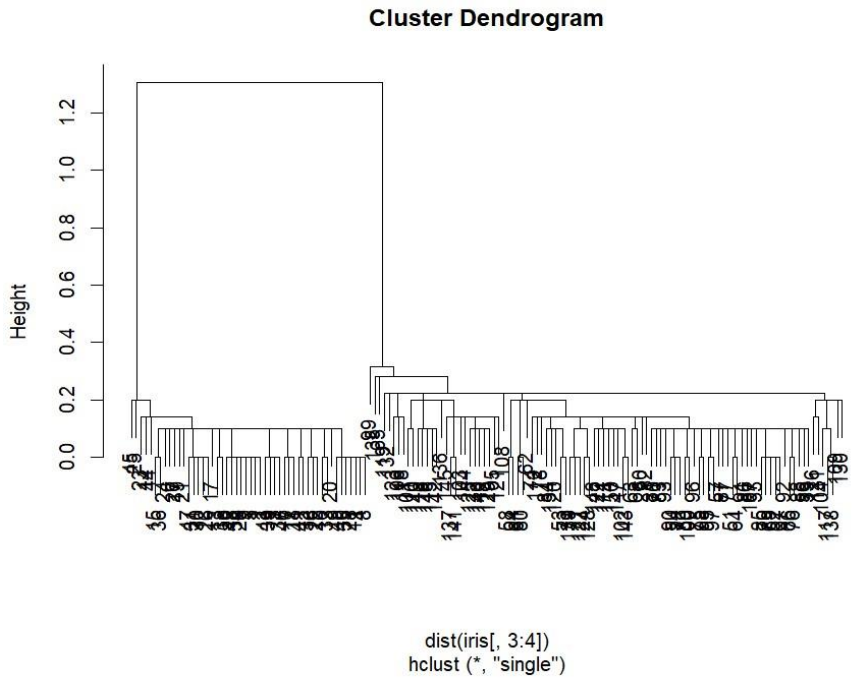
Practical 10

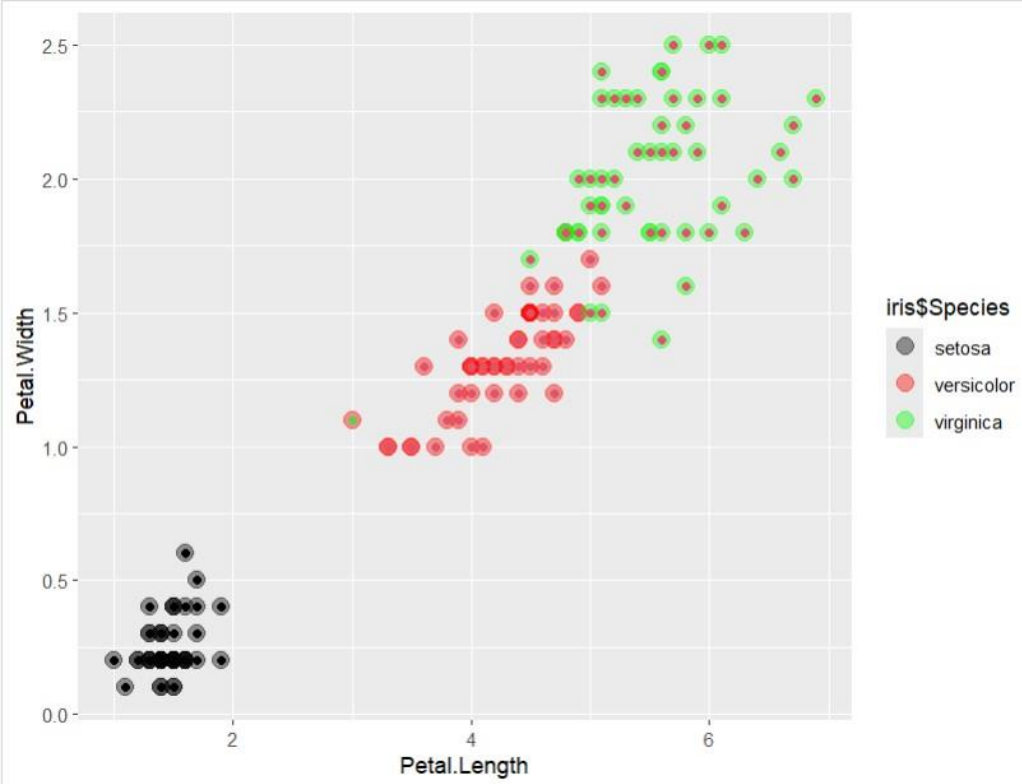
Implementation and analysis of clustering algorithms like K-means, Agglomerative:

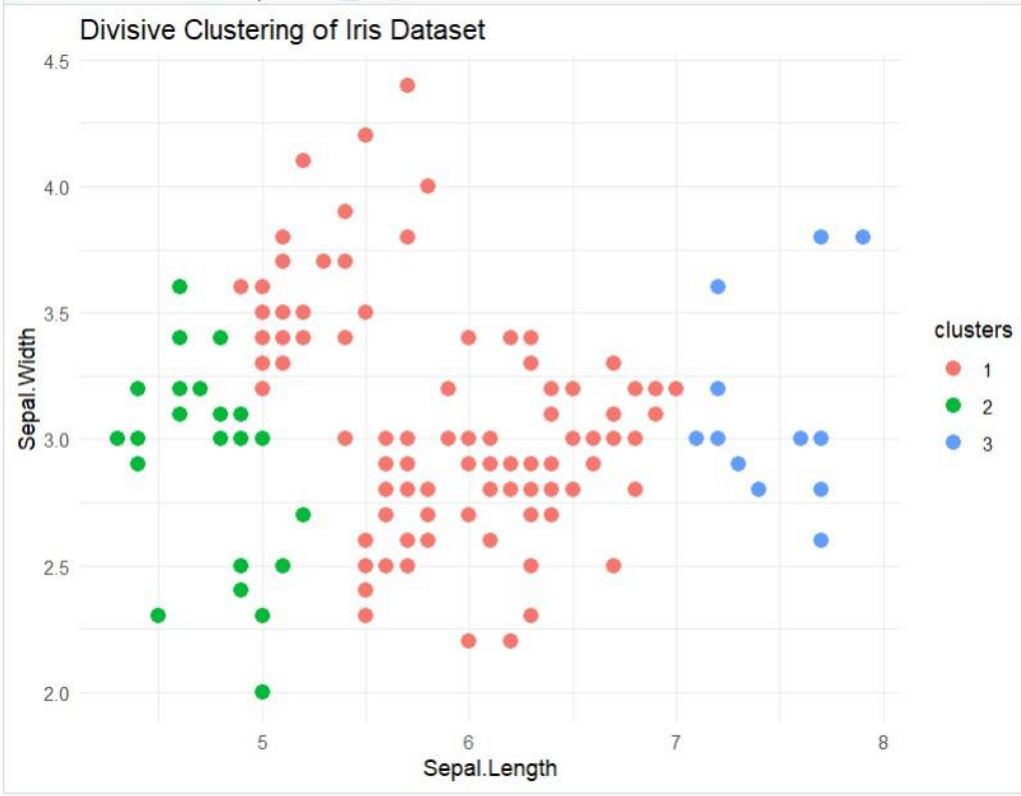
```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> irisCluster
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
 [60] 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 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2
[119] 2 1 2 2 2 2 2 2 1 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"         "iter"
[9] "ifault"
> irisCluster$cluster <- as.factor(irisCluster$cluster)
> ggplot(iris, aes(Petal.Length, Petal.Width, color =irisCluster$cluster)) + geom_point()
```



```
> 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
> clusters <- hclust(dist(iris[, 3:4]), method = 'single') ##you can change the method to avera
ge also
> plot(clusters)
> clusterCut <- cutree(clusters, 3)
> table(clusterCut, iris$Species)

clusterCut setosa versicolor virginica
         1     50          0         0
         2      0         49        50
         3      0          1         0
> ggplot(iris, aes(Petal.Length, Petal.Width, color = iris$Species)) +geom_point(alpha = 0.4, s
ize = 3.5) + geom_point(col = clusterCut) + scale_color_manual(values = c('black','red', 'gree
n'))
```




Petal.Width
Petal.Length
iris$Species
setosa
versicolor
virginica


Divisive Clustering of Iris Dataset
Sepal.Width
Sepal.Length
clusters
1
2
3