**Deccan Education Society’s**

**Navinchandra Mehta Institute of**

**Technology and Development**

# C E R T I F I C A T E

This is to certify that Miss. DHANASHREE LAXMAN KELASKAR of M.C.A. Semester I with Roll No.C23081 has completed 10 practicals of Advanced Database Management System under my supervision in this college during the year 2023-2024.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CO** | **R1**  **(Journal)** | **R2**  **(Performance during lab session)** | **R3**  **(Implementation using different problem solving techniques)** | **R4**  **(Mock Viva)** | **Attendance** |
| **CO1** |  |  |  |  |  |
| **CO2** |  |  |  |  |  |
| **CO3** |  |  |  |  |  |
| **CO4** |  |  |  |  |  |

Practical-in-charge Head of Department

MCA Department (NMITD)

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MCAL13 ADBMS Lab INDEX** | | | | |
| **Sr.No** | **Title** | **CO** | **Date** | **Sign** |
| 1 | Implementation of Partitions- Range, List.  Self-Learning Topics - Hash Partition, Composite partition | **CO1** |  |  |
| 2 | Analytical Queries  Roll\_Up, CUBE, First, Last, Lead, Lag, Rank and Dense Rank  Self-Learning Topics- Cume\_list, Percent\_rank | CO3 |  |  |
| 3 | Implementation of,  • Abstract Data Type  • Reference  Self-Learning Topics- Nested ADT, Inheritance | CO1 |  |  |
| 4 | ETL Transformation with Pentaho   1. Copy data from Source & store to Target 2. Adding Sequence 3. Adding Calculator 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 | CO2 |  |  |
| 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-**  Setw(), getw(), data(),rm()  **Attaching and Detaching data**  Reading data from the console  Loading data from different data sources(CSV,Excel) | CO4 |  |  |
| 6 | Data preprocessing techniques in R Naming and Renaming variables  Adding a new variables  Dealing with missing value  Dealing with categorical data  Data reduction using subsetting | CO4 |  |  |
| 7 | Implementation and analysis of Linear regression through graphical methods. | CO4 |  |  |
| 8 | Implementation and Analysis Classification algorithms like Naïve Bayesiam, K-Nearest Neighbour, ID3, C4.5 | CO4 |  |  |
| 9 | Implementation and analysis of Apriori Algorithm using Market Basket Analysis | CO4 |  |  |
| 10 | Implementation and analysis of clustering algorithms like K-means, Agglomarative | CO4 |  |  |

**Dhanashree L. Kelaskar**

**MCA-B**

**C23081**

**ADVANCED DATABASE MANAGEMENT SYSTEM**

**PRACTICAL NO. 1**

**AIM- Implementation of Partitions- Range, List.**

**Self-Learning Topics- Hash Partition, Composite partition**

**Range Partitioning**

**CODE-**

create table sales\_range

2 (salesman\_id number(5),

3 salesman\_name varchar2(30),

4 salesman\_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/YY

YY')),

9 Partition sales\_feb2001 values less than(To\_Date('01/03/2001','DD/MM/YY

YY')),

10 Partition sales\_mar2001 values less than(To\_Date('01/04/2001','DD/MM/YY

YY')),

11 Partition sales\_apr2001 values less than(To\_Date('01/05/2001','DD/MM/YY

YY'))

12 )

13 ;

Table created.

SELECT TABLE\_NAME, PARTITION\_NAME FROM USER\_TAB\_PARTITIONS WHERE TABLESPACE\_NAME='USERS';

|  |  |
| --- | --- |
| TABLE\_NAME | PARTITION\_NAME |
| SALES\_RANGE | SALES\_JAN2001 |
| SALES\_RANGE | SALES\_FEB2001 |
| SALES\_RANGE | SALES\_MAR2001 |
| SALES\_RANGE | SALES\_APR2001 |

4 rows selected.

SQL> insert into sales\_range values(1,’john smith’, 5000, to\_date(‘23/02/2001’,’DD/MM/YYYY’));

1 row created.

SQL> insert into sales\_range values(2,’mihir rana’ ,6000,to\_date(‘03/04/2001’,’DD/MM/YYYY’));

1 row created.

SQL> insert into sales\_range values(3,’riya rane’ ,8000,to\_date(‘11/03/2001’,’DD/MM/YYYY’));

1 row created.

SQL> insert into sales\_range values(4,'smith johns',8000,to\_date('13/01/2001','DD/MM/YYYY'));

1 row created.

SQL> insert into sales\_range values(5,'priya more',8000,to\_date('15/02/2001','DD/MM/YYYY'));

1 row created.

SQL>select \* from sales\_range;

|  |  |  |  |
| --- | --- | --- | --- |
| **SALESMAN\_ID** | **SALESMAN\_NAME** | **SALESMAN\_AMOUNT** | **SALES\_DAT** |
| 5 | Trupti Sawant | 9000 | 19-JAN-00 |
| 4 | smith johns | 8000 | 13-JAN-00 |
| 1 | john smith | 5000 | 23-FEB-00 |
| 6 | priya more | 8000 | 15-FEB-00 |
| 3 | riya rane | 8000 | 11-MAR-00 |
| 7 | smith joy | 7000 | 05-MAR-00 |
| 2 | mihir rana | 6000 | 03-APR-00 |
| 8 | jake gomez | 4000 | 05-APR-00 |

8 rows selected.

SQL>select \* from sales\_range partition(sales\_feb2001);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALESMAN\_AMOUNT | SALES\_DAT |
| 1 | john smith | 5000 | 23-FEB-00 |
| 6 | priya more | 8000 | 15-FEB-00 |

SQL> select \* from sales\_range partition(sales\_mar2001);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALESMAN\_AMOUNT | SALES\_DAT |
| 3 | riya rane | 8000 | 11-MAR-00 |
| 7 | smith joy | 7000 | 05-MAR-00 |

SQL> select \* from sales\_range partition(sales\_apr2001);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALESMAN\_AMOUNT | SALES\_DAT |
| 2 | mihir rana | 6000 | 03-APR-00 |
| 8 | jake gomez | 4000 | 05-APR-00 |

SQL> select \* from sales\_range partition(sales\_jan2001);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALESMAN\_AMOUNT | SALES\_DAT |
| 5 | Trupti Sawant | 9000 | 19-JAN-00 |
| 4 | smith johns | 8000 | 13-JAN-00 |

**List Partitioning**

**CODE-**

SQL>create table sales\_list

2 (salesman\_id number(5),

3 salesman\_name varchar2(20),

4 sales\_city varchar2(20),

5 sales\_date date)

6 PARTITION by list(sales\_city)

7 (PARTITION sales\_west values('andheri','borivali'),

8 PARTITION sales\_harbur values('chembur','vashi','panvel'),

9 PARTITION sales\_central values('thane','kalyan'),

10 PARTITION sales\_others values(default))

11 enable row movement;

Table created.

SQL>SELECT TABLE\_NAME, PARTITION\_NAME FROM USER\_TAB\_PARTITIONS WHERE TABLESPACE\_NAME='USERS';

|  |  |
| --- | --- |
| TABLE\_NAME | PARTITION\_NAME |
| SALES\_RANGE | SALES\_JAN2001 |
| SALES\_RANGE | SALES\_FEB2001 |
| SALES\_RANGE | SALES\_MAR2001 |
| SALES\_RANGE | SALES\_APR2001 |
| EMPLOYEE24 | E1 |
| EMPLOYEE24 | E2 |
| SALES\_LIST | SALES\_WEST |
| SALES\_LIST | SALES\_HARBUR |
| SALES\_LIST | SALES\_CENTRAL |
| SALES\_LIST | SALES\_OTHERS |

10 rows selected.

SQL>INSERT INTO sales\_list values(1,'mayank','andheri',to\_date('25/01/2001','dd/mm/yyyy'));

1 row created.

SQL> INSERT INTO sales\_list values(2,'mehul','vashi',to\_date('05/02/2001','dd/mm/yyyy'));

1 row created.

SQL> INSERT INTO sales\_list values(3,'nita','borivali',to\_date('07/03/2001','dd/mm/yyyy'));

1 row created.

SQL> INSERT INTO sales\_list values(4,'rina','chembur',to\_date('07/04/2001','dd/mm/yyyy'));

1 row created.

SQL> INSERT INTO sales\_list values(5,'neelam','chembur',to\_date('12/05/2001','dd/mm/yyyy'));

1 row created.

SQL> INSERT INTO sales\_list values(6,'sakshi','thane',to\_date('15/06/2001','dd/mm/yyyy'));

1 row created.

SQL> INSERT INTO sales\_list values(7,'meera','kalyan',to\_date('13/07/2001','dd/mm/yyyy'));

1 row created.

SQL>INSERT INTO sales\_list values(8,'yash','dadar',to\_date('01/08/2001','dd/mm/yyyy'));

1 row created.

SQL>INSERT INTO sales\_list values(9,'tina','navi mumbai',to\_date('07/09/2001','dd/mm/yyyy'));

1 row created.

select \* from sales\_list;

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_CITY | SALES\_DAT |
| 1 | mayank | andheri | 25-JAN-00 |
| 3 | nita | borivali | 07-MAR-00 |
| 2 | mehul | vashi | 05-FEB-00 |
| 4 | rina | chembur | 07-APR-00 |
| 5 | neelam | chembur | 12-MAY-00 |
| 6 | sakshi | thane | 15-JUN-00 |
| 7 | meera | kalyan | 13-JUL-00 |
| 8 | yash | dadar | 01-AUG-00 |
| 9 | tina | navi mumbai | 07-SEP-00 |

9 rows selected.

SQL>select \* from sales\_list partition(sales\_west);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_CITY | SALES\_DAT |
| 1 | mayank | andheri | 25-JAN-00 |
| 3 | nita | borivali | 07-MAR-00 |

SQL> select \* from sales\_list partition(sales\_central);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_CITY | SALES\_DAT |
| 6 | sakshi | thane | 15-JUN-00 |
| 7 | meera | kalyan | 13-JUL-00 |

SQL> select \* from sales\_list partition(sales\_harbur);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_CITY | SALES\_DAT |
| 2 | mehul | vashi | 05-FEB-00 |
| 4 | rina | chembur | 07-APR-00 |
| 5 | neelam | chembur | 12-MAY00 |

SQL>select \* from sales\_list partition(sales\_others);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_CITY | SALES\_DAT |
| 8 | yash | dadar | 01-AUG-00 |
| 9 | tina | navi mumbai | 07-SEP-00 |

**Hash Partitioning**

**CODE-**

**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;

Table created.

**SQL>** SELECT TABLE\_NAME, PARTITION\_NAME FROM USER\_TAB\_PARTITIONS WHERE TABLESPACE\_NAME='USERS';

|  |  |
| --- | --- |
| TABLE\_NAME | PARTITION\_NAME |
| SALES\_HASH | SYS\_P23 |
| SALES\_HASH | SYS\_P21 |
| SALES\_HASH | SYS\_P22 |
| SALES\_HASH | SYS\_P24 |

SQL> insert into sales\_hash values(101,'John',44000,12);

1 row created.

SQL> insert into sales\_hash values(102,'Neha',54000,10);

1 row created.

SQL> insert into sales\_hash values(103,'Rohan',45000,15);

1 row created.

SQL> insert into sales\_hash values(104,'Rahul',32001,06);

1 row created.

SQL> insert into sales\_hash values(105,'Dhanashree',56000,11);

1 row created.

SQL> insert into sales\_hash values(106,'Nash',42001,16);

1 row created.

SQL> insert into sales\_hash values(108,'Alis',54000,18);

1 row created.

SQL> select \* from sales\_hash;

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_AMOUNT | WEEK\_NO |
| 108 | Alis | 54000 | 18 |
| 104 | Rahul | 32001 | 6 |
| 102 | Neha | 54000 | 10 |
| 103 | Rohan | 45000 | 15 |
| 105 | Dhanashree | 56000 | 11 |
| 101 | John | 44000 | 12 |
| 106 | Nash | 42001 | 16 |

8 rows selected.

SQL> select \* from sales\_hash partition(sys\_p21);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_AMOUNT | WEEK\_NO |
| 108 | Alis | 54000 | 18 |

SQL> select \* from sales\_hash partition(sys\_p22);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_AMOUNT | WEEK\_NO |
| 104 | Rahul | 32001 | 6 |

SQL> select \* from sales\_hash partition(sys\_p23);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_AMOUNT | WEEK\_NO |
| 102 | Neha | 54000 | 10 |
| 103 | Rohan | 45000 | 15 |
| 105 | Dhanashree | 56000 | 11 |

SQL> select \* from sales\_hash partition(sys\_p24);

|  |  |  |  |
| --- | --- | --- | --- |
| SALESMAN\_ID | SALESMAN\_NAME | SALES\_AMOUNT | WEEK\_NO |
| 101 | John | 44000 | 12 |
| 106 | Nash | 42001 | 16 |

**Composite Partitioning**

**CODE-**

SQL>create table orders

2 (

3 order\_no NUMBER,

4 order\_date DATE,

5 prod NUMBER,

6 quantity NUMBER

7 )

8 PARTITION BY RANGE(order\_date)

9 SUBPARTITION BY HASH(prod) SUBPARTITIONS 4

10 (

11 PARTITION order1 VALUES LESS THAN(TO\_DATE('01/02/2023','DD/MM/YYYY')),

12 PARTITION order2 VALUES LESS THAN(TO\_DATE('01/03/2023','DD/MM/YYYY')),

13 PARTITION order3 VALUES LESS THAN(TO\_DATE('01/04/2023','DD/MM/YYYY')),

14 PARTITION order4 VALUES LESS THAN(TO\_DATE('01/05/2023','DD/MM/YYYY'))

15 );

**Table created.**

SQL> SELECT TABLE\_NAME, PARTITION\_NAME FROM USER\_TAB\_PARTITIONS WHERE TABLESPACE\_NAME='USERS';

|  |  |
| --- | --- |
| TABLE\_NAME | PARTITION\_NAME |
| ORDERS | ORDER1 |
| ORDERS | ORDER2 |
| ORDERS | ORDER3 |
| ORDERS | ORDER4 |

**4 rows selected.**

SQL>INSERT INTO orders

2 VALUES (11, TO\_DATE('11/02/2023','DD/MM/YYYY'),101,5);

**1 row created.**

SQL> INSERT INTO orders

2 VALUES (12, TO\_DATE('21/01/2023','DD/MM/YYYY'),102,4);

**1 row created.**

SQL> INSERT INTO orders

2 VALUES (13, TO\_DATE('15/03/2023','DD/MM/YYYY'),103,2);

**1 row created.**

SQL> INSERT INTO orders

2 VALUES (14, TO\_DATE('27/04/2023','DD/MM/YYYY'),104,8);

**1 row created.**

SQL> INSERT INTO orders

2 VALUES (15, TO\_DATE('16/01/2023','DD/MM/YYYY'),105,6);

**1 row created.**

SQL> INSERT INTO orders

2 VALUES (16, TO\_DATE('20/03/2023','DD/MM/YYYY'),106,3);

**1 row created.**

SQL> INSERT INTO orders

2 VALUES (17, TO\_DATE('20/02/2023','DD/MM/YYYY'),107,6);

**1 row created.**

SQL> select \* from orders;

|  |  |  |  |
| --- | --- | --- | --- |
| ORDER\_NO | ORDER\_DAT | PROD | QUANTITY |
| 12 | 21-JAN-23 | 102 | 4 |
| 15 | 16-JAN-23 | 105 | 6 |
| 17 | 20-FEB-23 | 107 | 6 |
| 11 | 11-FEB-23 | 101 | 5 |
| 13 | 15-MAR-23 | 103 | 2 |
| 16 | 20-MAR-23 | 106 | 3 |
| 14 | 27-APR-23 | 104 | 8 |

**7 rows selected.**

SQL>SELECT \* FROM orders PARTITION(order1);

|  |  |  |  |
| --- | --- | --- | --- |
| ORDER\_NO | ORDER\_DAT | PROD | QUANTITY |
| 12 | 21-JAN-23 | 102 | 4 |
| 15 | 16-JAN-23 | 105 | 6 |

SQL> SELECT \* FROM orders PARTITION(order2);

|  |  |  |  |
| --- | --- | --- | --- |
| ORDER\_NO | ORDER\_DAT | PROD | QUANTITY |
| 17 | 20-FEB-23 | 107 | 6 |
| 11 | 11-FEB-23 | 101 | 5 |

SQL> SELECT \* FROM orders PARTITION(order3);

|  |  |  |  |
| --- | --- | --- | --- |
| ORDER\_NO | ORDER\_DAT | PROD | QUANTITY |
| 13 | 15-MAR-23 | 103 | 2 |
| 16 | 20-MAR-23 | 106 | 3 |

SQL> SELECT \* FROM orders PARTITION(order4);

|  |  |  |  |
| --- | --- | --- | --- |
| ORDER\_NO | ORDER\_DAT | PROD | QUANTITY |
| 14 | 27-APR-23 | 104 | 8 |

**PRACTICAL NO.2**

**AIM- Analytical Queries - Roll\_Up, CUBE, First, Last, Lead, Lag, Rank and Dense Rank**

**Self-Learning Topics- Cume\_list, Percent\_rank**

**CODE-**

SQL>create table ssb1

2 (emp\_no number(5),

3 dep\_no number(10),

4 bdate date,

5 sal number(10),

6 comm number(10),

7 job varchar2(10));

**Table created.**

SQL> insert all

2 into ssb1 values(101,10,'12-Jan-82',22001,1000,'Clerk')

3 into ssb1 values(102,10,'13-Feb-83',22001,2001,'Clerk')

4 select \* from dual;

**2 rows created.**

SQL> insert all

2 into ssb1 values(103,10,'14-Mar-84',44000,200,'Clerk')

3 into ssb1 values(104,20,'15-Apr-87',55000,3000,'Manager')

4 into ssb1 values(105,20,'14-Jun-82',38000,4500,'Manager')

5 into ssb1 values(106,20,'15-Aug-88',44000,500,'Manager')

6 into ssb1 values(107,10,'31-Dec-81',58000,8000,'Manager')

7 into ssb1 values(108,20,'25-Jul-88',49000,700,'Clerk')

8 into ssb1 values(109,10,'27-Oct-85',46000,900,'Clerk')

9 into ssb1 values(110,20,'06-Dec-83',54000,5000,'Manager')

10 select \* from dual;

**8 rows created.**

SQL> select \* from ssb1;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMP\_NO | DEP\_NO | BDATE | SAL | COMM | JOB |
| 101 | 10 | 12-JAN-82 | 22001 | 1000 | Clerk |
| 102 | 10 | 13-FEB-83 | 22001 | 2001 | Clerk |
| 103 | 10 | 14-MAR-84 | 44000 | 200 | Clerk |
| 104 | 20 | 15-APR-87 | 55000 | 3000 | Manager |
| 105 | 20 | 14-JUN-82 | 38000 | 4500 | Manager |
| 106 | 20 | 15-AUG-88 | 44000 | 500 | Manager |
| 107 | 10 | 31-DEC-81 | 58000 | 8000 | Manager |
| 108 | 20 | 25-JUL-88 | 49000 | 700 | Clerk |
| 109 | 10 | 27-OCT-85 | 46000 | 900 | Clerk |
| 110 | 20 | 06-DEC-83 | 54000 | 5000 | Manager |

**10 rows selected.**

SQL> commit;

**Commit complete.**

**ROLLUP**

SQL> select dep\_no, job, count(\*), sum(sal)

2 from ssb1

3 group by rollup(dep\_no, job);

|  |  |  |  |
| --- | --- | --- | --- |
| DEP\_NO | JOB | COUNT(\*) | SUM(SAL) |
| 10 | Clerk | 4 | 134000 |
| 10 | Manager | 1 | 58000 |
| 10 |  | 5 | 192001 |
| 20 | Clerk | 1 | 49000 |
| 20 | Manager | 4 | 191000 |
| 20 |  | 5 | 240000 |
|  |  | 10 | 432001 |

**7 rows selected.**

**PARTIAL ROLLUP**

SQL>select dep\_no, job, sum(sal)

2 from ssb1

3 where dep\_no in(10,20)

4 group by dep\_no, rollup(job);

|  |  |  |
| --- | --- | --- |
| DEP\_NO | JOB | SUM(SAL) |
| 10 | Clerk | 134000 |
| 20 | Manager | 191000 |
| 10 | Manager | 58000 |
| 20 | Clerk | 49000 |
| 10 |  | 192001 |
| 20 |  | 240000 |

**6 rows selected.**

SQL>select dep\_no, job, count(\*),sum(sal)

2 from ssb1

3 group by job, rollup(dep\_no);

|  |  |  |  |
| --- | --- | --- | --- |
| DEP\_NO | JOB | COUNT(\*) | SUM(SAL) |
| 10 | Clerk | 4 | 134000 |
| 20 | Manager | 4 | 191000 |
| 10 | Manager | 1 | 58000 |
| 20 | Clerk | 1 | 49000 |
|  | Clerk | 5 | 183000 |
|  | Manager | 5 | 249000 |

**6 rows selected.**

**CUBE**

SQL> select dep\_no, job, count(\*), sum(sal)

2 from ssb1

3 group by cube(dep\_no, job);

|  |  |  |  |
| --- | --- | --- | --- |
| DEP\_NO | JOB | COUNT(\*) | SUM(SAL) |
|  |  | 10 | 432001 |
|  | Clerk | 5 | 183000 |
|  | Manager | 5 | 249000 |
| 10 |  | 5 | 192001 |
| 10 | Clerk | 4 | 134000 |
| 10 | Manager | 1 | 58000 |
| 20 |  | 5 | 240000 |
| 20 | Clerk | 1 | 49000 |
| 20 | Manager | 4 | 191000 |

**9 rows selected.**

**RANK**

SQL> select emp\_no, dep\_no, sal, comm,

2 rank() over(partition by dep\_no order by sal) as rank from ssb1;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMP\_NO | DEP\_NO | SAL | COMM | RANK |
| 101 | 10 | 22001 | 1000 | 1 |
| 102 | 10 | 22001 | 2001 | 1 |
| 103 | 10 | 44000 | 200 | 3 |
| 109 | 10 | 46000 | 900 | 4 |
| 107 | 10 | 58000 | 8000 | 5 |
| 105 | 20 | 38000 | 4500 | 1 |
| 106 | 20 | 44000 | 500 | 2 |
| 108 | 20 | 49000 | 700 | 3 |
| 110 | 20 | 54000 | 5000 | 4 |
| 104 | 20 | 55000 | 3000 | 5 |

**10 rows selected.**

**DENSE\_RANK**

SQL> select emp\_no, dep\_no, sal, comm,

2 dense\_rank() over(partition by dep\_no order by sal) as rank from ssb1;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EMP\_NO | DEP\_NO | SAL | COMM | RANK |
| 101 | 10 | 22001 | 1000 | 1 |
| 102 | 10 | 22001 | 2001 | 1 |
| 103 | 10 | 44000 | 200 | 2 |
| 109 | 10 | 46000 | 900 | 3 |
| 107 | 10 | 58000 | 8000 | 4 |
| 105 | 20 | 38000 | 4500 | 1 |
| 106 | 20 | 44000 | 500 | 2 |
| 108 | 20 | 49000 | 700 | 3 |
| 110 | 20 | 54000 | 5000 | 4 |
| 104 | 20 | 55000 | 3000 | 5 |

**10 rows selected.**

**LEAD**

SQL> select emp\_no, bdate,

2 lead(bdate, 1) over(order by bdate) as "next"

3 from ssb1;

|  |  |  |
| --- | --- | --- |
| EMP\_NO | BDATE | next |
| 107 | 31-DEC-81 | 12-JAN-82 |
| 101 | 12-JAN-82 | 14-JUN-82 |
| 105 | 14-JUN-82 | 13-FEB-83 |
| 102 | 13-FEB-83 | 06-DEC-83 |
| 110 | 06-DEC-83 | 14-MAR-84 |
| 103 | 14-MAR-84 | 27-OCT-85 |
| 109 | 27-OCT-85 | 15-APR-87 |
| 104 | 15-APR-87 | 25-JUL-88 |
| 108 | 25-JUL-88 | 15-AUG-88 |
| 106 | 15-AUG-88 |  |

**10 rows selected.**

**LAG**

SQL> select emp\_no, bdate,

2 lAG(bdate, 1) over(order by bdate) as "Previous"

3 from ssb1;

|  |  |  |
| --- | --- | --- |
| EMP\_NO | BDATE | Previous |
| 107 | 31-DEC-81 |  |
| 101 | 12-JAN-82 | 31-DEC-81 |
| 105 | 14-JUN-82 | 12-JAN-82 |
| 102 | 13-FEB-83 | 14-JUN-82 |
| 110 | 06-DEC-83 | 13-FEB-83 |
| 103 | 14-MAR-84 | 06-DEC-83 |
| 109 | 27-OCT-85 | 14-MAR-84 |
| 104 | 15-APR-87 | 27-OCT-85 |
| 108 | 25-JUL-88 | 15-APR-87 |
| 106 | 15-AUG-88 | 25-JUL-88 |

**10 rows selected.**

**FIRST**

SQL> select dep\_no, sal,

2 max(sal) keep(dense\_rank first order by sal desc)

3 over(partition by dep\_no)"max"

4 from ssb1;

|  |  |  |
| --- | --- | --- |
| DEP\_NO | SAL | max |
| 10 | 46000 | 58000 |
| 10 | 22001 | 58000 |
| 10 | 58000 | 58000 |
| 10 | 44000 | 58000 |
| 10 | 22001 | 58000 |
| 20 | 49000 | 55000 |
| 20 | 44000 | 55000 |
| 20 | 54000 | 55000 |
| 20 | 55000 | 55000 |
| 20 | 38000 | 55000 |

**10 rows selected.**

**LAST**

SQL> select dep\_no, sal,

2 min(sal) keep(dense\_rank last order by sal desc)

3 over(partition by dep\_no)"min"

4 from ssb1;

|  |  |  |
| --- | --- | --- |
| DEP\_NO | SAL | min |
| 10 | 46000 | 22001 |
| 10 | 22001 | 22001 |
| 10 | 58000 | 22001 |
| 10 | 44000 | 22001 |
| 10 | 22001 | 22001 |
| 20 | 49000 | 38000 |
| 20 | 44000 | 38000 |
| 20 | 54000 | 38000 |
| 20 | 55000 | 38000 |
| 20 | 38000 | 38000 |

**10 rows selected.**

**PRACTICAL NO.3**

**AIM- Implementation of - • Abstract Data Type**

**• Reference**

**Self-Learning Topics- Nested ADT, Inheritance**

**CODE-**

SQL>create type type\_name As object

2 (

3 fname varchar(20),

4 mname varchar(20),

5 Iname varchar(20)

6 );

7 /

**Type created.**

SQL> create type type\_address As object

2 (

3 street varchar(20),

4 city varchar(20),

5 pincode number(10)

6 );

7 /

**Type created.**

SQL>create table customer

2 (

3 c\_id number(5) primary key,

4 c\_name type\_name,

5 c\_add type\_address,

6 c\_phno number(10)

7 );

**Table created.**

SQL> insert into customer

2 values(1,type\_name('mina','r','rana'),

3 type\_address('bandra','mumbai',400501),12456789);

**1 row created.**

SQL>insert into customer

2 values(2,type\_name('sanjiv','v','rajiv'),

3 type\_address('dadar','mumbai',400502),12356787);

**1 row created.**

SQL> insert into customer

2 values(3,type\_name('sagar','b','malhotra'),

3 type\_address('vileparle','mumbai',400503),22354787);

**1 row created.**

SQL> insert into customer

2 values(4,type\_name('ansh','c','dhawan'),

3 type\_address('jogeshwari','mumbai',400504),2156787);

**1 row created.**

SQL> insert into customer

2 values(5,type\_name('ankit','a','mehra'),

3 type\_address('vileparle','mumbai',400505),2456982);

**1 row created.**

select \* from customer;

|  |  |
| --- | --- |
| C\_ID | 1 |
| C\_NAME(FNAME, MNAME, INAME) | TYPE\_NAME('mina', 'r', 'rana') |
| C\_ADD(STREET, CITY, PINCODE) | TYPE\_ADDRESS('bandra', 'mumbai', 400501) |
| C\_PHNO | 12456789 |

|  |  |
| --- | --- |
| C\_ID | 2 |
| C\_NAME(FNAME, MNAME, INAME) | TYPE\_NAME('sanjiv', 'v', 'rajiv') |
| C\_ADD(STREET, CITY, PINCODE) | TYPE\_ADDRESS('dadar', 'mumbai', 400502) |
| C\_PHNO | 12356787 |

|  |  |
| --- | --- |
| C\_ID | 3 |
| C\_NAME(FNAME, MNAME, INAME) | TYPE\_NAME('sagar', 'b', 'malhotra') |
| C\_ADD(STREET, CITY, PINCODE) | TYPE\_ADDRESS('vileparle', 'mumbai', 400503) |
| C\_PHNO | 22354787 |

|  |  |
| --- | --- |
| C\_ID | 4 |
| C\_NAME(FNAME, MNAME, INAME) | TYPE\_NAME('ansh', 'c', 'dhawan') |
| C\_ADD(STREET, CITY, PINCODE) | TYPE\_ADDRESS('jogeshwari', 'mumbai', 400504) |
| C\_PHNO | 2156787 |

|  |  |
| --- | --- |
| C\_ID | 5 |
| C\_NAME(FNAME, MNAME, INAME) | TYPE\_NAME('ankit', 'a', 'mehra') |
| C\_ADD(STREET, CITY, PINCODE) | TYPE\_ADDRESS('vileparle', 'mumbai', 400505) |
| C\_PHNO | 2456982 |

SQL>desc customer;

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| C\_ID | NOT NULL | NUMBER(5) |
| C\_NAME |  | TYPE\_NAME |
| C\_ADD |  | TYPE\_ADDRESS |
| C\_PHNO |  | NUMBER(10) |

SQL> set describe depth 2;

SQL> desc customer;

|  |  |  |
| --- | --- | --- |
| Name | Null? | Type |
| C\_ID | NOT NULL | NUMBER(5) |
| C\_NAME |  | TYPE\_NAME |
| FNAME |  | VARCHAR2(20) |
| MNAME |  | VARCHAR2(20) |
| INAME |  | VARCHAR2(20) |
| C\_ADD |  | TYPE\_ADDRESS |
| STREET |  | VARCHAR2(20) |
| CITY |  | VARCHAR2(20) |
| PINCODE |  | NUMBER(10) |
| C\_PHNO |  | NUMBER(10) |

SQL> select c.c\_add.street from customer c where c\_id=1;

|  |
| --- |
| C\_ADD.STREET |
| bandra |

SQL> select c.c\_name.fname from customer c where c\_id=1;

|  |
| --- |
| C\_NAME.FNAME |
| mina |

SQL> select c\_name from customer;

|  |
| --- |
| C\_NAME(FNAME MNAME INAME) |
| TYPE\_NAME('mina''r''rana') |
| TYPE\_NAME('sanjiv''v''rajiv') |
| TYPE\_NAME('sagar''b''malhotra') |
| TYPE\_NAME('ansh''c''dhawan') |
| TYPE\_NAME('ankit''a''mehra') |

SQL> select c.c\_name.fname||' '||c.c\_name.mname||' '||c.c\_name.iname

2 from customer c;

|  |
| --- |
| C.C\_NAME.FNAME||''||C.C\_NAME.MNAME||''||C.C\_NAME.INAME |
| mina r rana |
| sanjiv v rajiv |
| sagar b malhotra |
| ansh c dhawan |
| ankit a mehra |

**Nesting & Altering Existing Type**

SQL> ALTER TYPE type\_address add attribute(name type\_name)cascade;

**Type altered.**

SQL> CREATE TABLE cust2

2 (cust\_no INTEGER,

3 add1 type\_address);

**Table created.**

SQL> insert into cust2

2 values(101,type\_address('prabhadevi','mumbai',400012,

3 type\_name('dhanashree','r','Paste')));

**1 row created.**

SQL> insert into cust2

2 values(102,type\_address('thane','mumbai',400601,

3 type\_name('dhanashree','v','Kelaskar')));

**1 row created.**

SQL> insert into cust2

2 values(103,type\_address('sion','mumbai',400001,

3 type\_name('sanjana','m','pradhan')));

**1 row created.**

SQL> insert into cust2

2 values(104,type\_address('matunga','mumbai',400002,

3 type\_name('shruti','s','shah')));

**1 row created.**

SQL> insert into cust2

2 values(105,type\_address('ghatkopar','mumbai',400005,

3 type\_name('shamika','a','zagade')));

**1 row created.**

SQL>DESC cust2;

|  |
| --- |
| Name Null? Type |
| CUST\_NO NUMBER(38) |
| ADD1 TYPE\_ADDRESS |

SQL> SELECT cust\_no FROM cust2;

|  |
| --- |
| CUST\_NO |
| 101 |
| 102 |
| 103 |
| 104 |
| 105 |

SQL> SELECT cust\_no,c.add1.street,c.add1.name.fname FROM cust2 c;

|  |
| --- |
| CUST\_NO ADD1.STREET ADD1.NAME.FNAME |
| 101 prabhadevi dhanashree |
| 102 thane dhanashree |
| 103 sion sanjana |
| 104 matunga shruti |
| 105 ghatkopar shamika |

**OBJECT TABLE**

SQL> CREATE OR REPLACE type stud\_type AS object

2 (roll\_no NUMBER(5),

3 name VARCHAR2(30)

4 );

5 /

**Type created.**

SQL> CREATE TABLE students of stud\_type;

**Table created.**

SQL> INSERT INTO students VALUES(stud\_type(1,'john'));

**1 row created.**

SQL> INSERT INTO students VALUES(stud\_type(2,'smith'));

**1 row created.**

SQL> INSERT INTO students VALUES(stud\_type(3,'sam'));

**1 row created.**

SQL> INSERT INTO students VALUES(stud\_type(4,'george'));

**1 row created.**

SQL>INSERT INTO students VALUES(stud\_type(5,'laura'));

**1 row created.**

SQL>SELECT \* FROM students;

|  |
| --- |
| ROLL\_NO NAME |
| 1 john |
| 2 smith |
| 3 sam |
| 4 george |
| 5 laura |

SQL> SELECT roll\_no FROM students;

|  |
| --- |
| ROLL\_NO |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

SQL> SELECT s.roll\_no FROM students s;

|  |
| --- |
| ROLL\_NO |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

SQL>SELECT \* FROM students s WHERE s.name='smith';

|  |
| --- |
| ROLL\_NO NAME |
| 2 smith |

**REF & DREF FUNCTION**

SQL> CREATE OR REPLACE type ANIMAL\_TY AS OBJECT

2 (Breed VARCHAR2(25),

3 Name VARCHAR2(25),

4 BirthDate DATE);

5 /

**Type created.**

SQL> CREATE TABLE ANIMAL of ANIMAL\_TY;

**Table created.**

SQL> INSERT INTO ANIMAL VALUES(

2 ANIMAL\_TY('MULE','FRANCES','01-MAR-02'));

**1 row created.**

SQL> INSERT INTO ANIMAL VALUES(

2 ANIMAL\_TY('DOG','BENJI','03-SEP-01'));

**1 row created.**

SQL> INSERT INTO ANIMAL VALUES(

2 ANIMAL\_TY('HORSE','AKSHAY','10-OCT-11'));

**1 row created.**

SQL> INSERT INTO ANIMAL VALUES(

2 ANIMAL\_TY('CAT','JUI','04-JUNE-04'));

**1 row created.**

SQL> INSERT INTO ANIMAL VALUES(

2 ANIMAL\_TY('RABBIT','SUMEDH','15-FEB-02'));

**1 row created.**

**THE REF FUNCTION -**

SQL> SELECT REF(A) FROM ANIMAL A;

REF(A)

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

0000280209C6655DC417AD43D0AFB331B539926FCBAFEE57BDD30D4FC98A7D1317905DEBE700429D

990000

0000280209D6976F812411446EACB9596A6B89C348AFEE57BDD30D4FC98A7D1317905DEBE700429D

990001

0000280209EF43A3F378A54A84914B2936A450E6B2AFEE57BDD30D4FC98A7D1317905DEBE700429D

990002

00002802097A4D3708746541B580A7F109DFB2B334AFEE57BDD30D4FC98A7D1317905DEBE700429D

990003

REF(A)

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

0000280209468A20C1675B4319B4B728AA0A16485FAFEE57BDD30D4FC98A7D1317905DEBE700429D

990004

**USING THE DEREF FUNCTION-**

CREATE TABLE KEEPER

2 (KeeperName VARCHAR2(25),

3 AnimalKept REF ANIMAL\_TY);

**Table created.**

SQL> DESCRIBE KEEPER

|  |
| --- |
| Name Null? Type |
| KEEPERNAME VARCHAR2(25) |
| ANIMALKEPT REF OF ANIMAL\_TY |
| BREED VARCHAR2(25) |
| NAME VARCHAR2(25) |
| BIRTHDATE DATE |

SQL> INSERT INTO KEEPER

2 SELECT 'CATHERINE',

3 REF(A)

4 FROM ANIMAL A

5 WHERE Name='AKSHAY';

**1 row created.**

SQL> set describe depth 2;

SQL> DESCRIBE KEEPER

|  |
| --- |
| Name Null? Type |
| KEEPERNAME VARCHAR2(25) |
| ANIMALKEPT REF OF ANIMAL\_TY |
| BREED VARCHAR2(25) |
| NAME VARCHAR2(25) |
| BIRTHDATE DATE |

SQL> SELECT \* FROM KEEPER;

KEEPERNAME

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

ANIMALKEPT

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

CATHERINE

0000220208EF43A3F378A54A84914B2936A450E6B2AFEE57BDD30D4FC98A7D1317905DEBE7

SQL> SELECT KeeperName,DEREF(K.AnimalKept)

2 FROM KEEPER K;

KEEPERNAME

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

DEREF(K.ANIMALKEPT)(BREED, NAME, BIRTHDATE)

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

CATHERINE

ANIMAL\_TY('HORSE', 'AKSHAY', '10-OCT-11')

**PRACTICAL NO.4**

**AIM- ETL Transformation with Pentaho-**

1. **Copy data from Source & store to Target**
2. **Adding Sequence**
3. **Adding Calculator**
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**

**CODE-**

SQL> create table emp25

2 (

3 emp\_no numeric(5),

4 fname varchar2(10),

5 lname varchar2(10),

6 salary numeric(5),

7 comm numeric(5)

8 );

**Table created.**

SQL> insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm);

Enter value for emp\_no- 1

Enter value for fname- yash

Enter value for lname- shah

Enter value for salary- 25000

Enter value for comm- 500

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(1,'yash','shah',25000,500)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 2

Enter value for fname- priya

Enter value for lname- agarwal

Enter value for salary- 35000

Enter value for comm- 1000

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(2,'priya','agarwal',35000,1000)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 3

Enter value for fname- ritesh

Enter value for lname- deshmukh

Enter value for salary- 45000

Enter value for comm- 1500

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(3,'ritesh','deshmukh',45000,1500)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 4

Enter value for fname- riyal

Enter value for lname- nehru

Enter value for salary- 50000

Enter value for comm- 500

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(4,'riyal','nehru',50000,500)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 5

Enter value for fname- siddesh

Enter value for lname- mehra

Enter value for salary- 20010

Enter value for comm- 200

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(5,'siddesh','mehra',20010,200)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 6

Enter value for fname- raju

Enter value for lname- thakur

Enter value for salary- 35000

Enter value for comm- 1000

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(6,'raju','thakur',35000,1000)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 7

Enter value for fname- prerna

Enter value for lname- alkute

Enter value for salary- 40000

Enter value for comm- 1500

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(7,'prerna','alkute',40000,1500)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 8

Enter value for fname- shaam

Enter value for lname- malhotra

Enter value for salary- 60000

Enter value for comm- 2001

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(8,'shaam','malhotra',60000,2001)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 9

Enter value for fname- prisha

Enter value for lname- ketkar

Enter value for salary- 55000

Enter value for comm- 500

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(9,'prisha','ketkar',55000,500)

**1 row created.**

SQL> r

1\* insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

Enter value for emp\_no- 10

Enter value for fname- shashank

Enter value for lname- gokhale

Enter value for salary- 56000

Enter value for comm- 500

old 1- insert into emp25 values(&emp\_no,'&fname','&lname',&salary,&comm)

new 1- insert into emp25 values(10,'shashank','gokhale',56000,500)

**1 row created.**

SQL> commit ;

**Commit complete.**

SQL> select \* from emp25;

|  |
| --- |
| EMP\_NO FNAME LNAME SALARY COMM |
| 1 yash shah 25000 500 |
| 2 priya agarwal 35000 1000 |
| 3 ritesh deshmukh 45000 1500 |
| 4 riyal nehru 50000 500 |
| 5 siddesh mehra 20010 200 |
| 6 raju thakur 35000 1000 |
| 7 prerna alkute 40000 1500 |
| 8 shaam malhotra 60000 2001 |
| 9 prisha ketkar 55000 500 |
| 10 shashank gokhale 56000 500 |

**10 rows selected.**

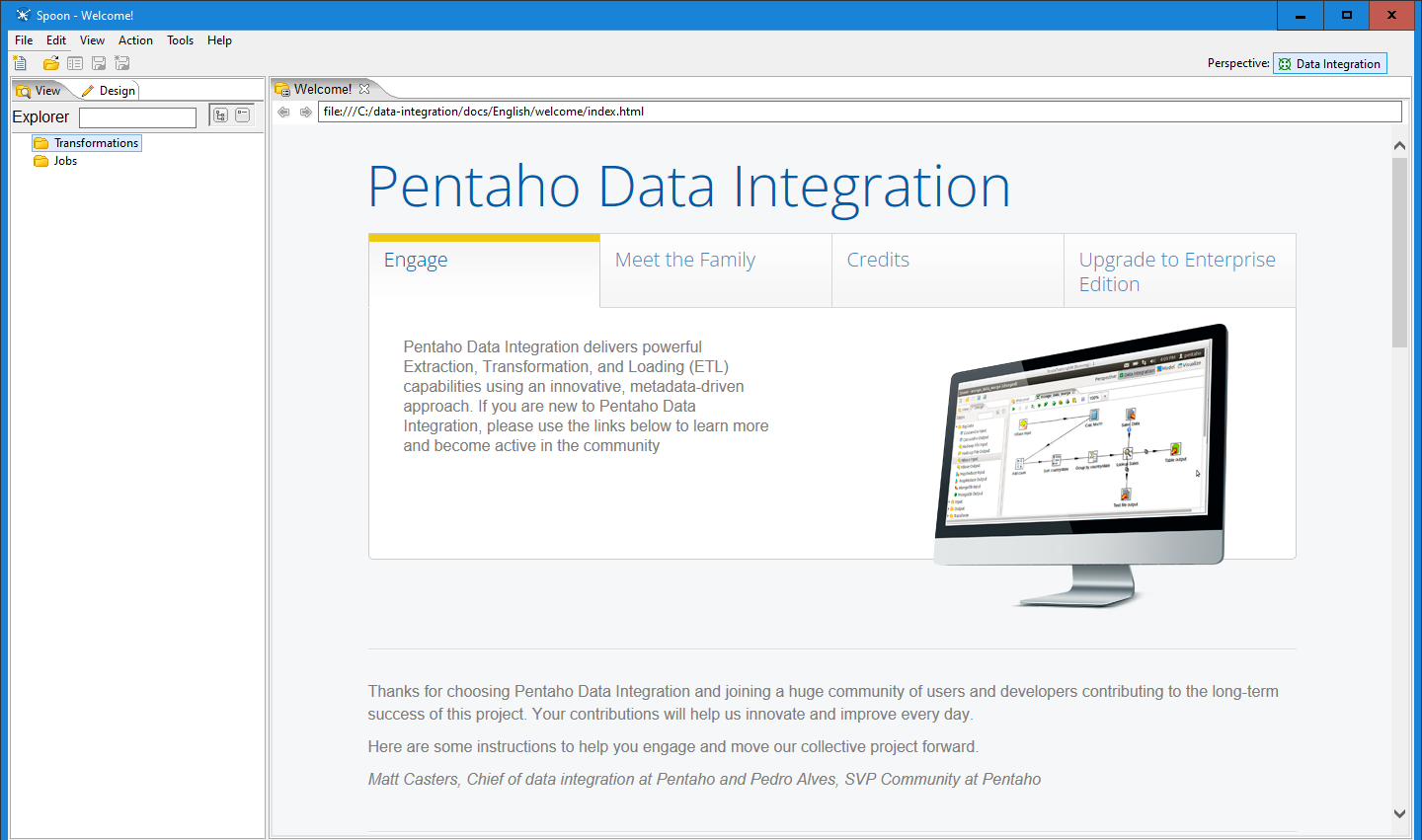
**TRANSFORMATION 1-**

STARTING WITH PENTAHO-

Step 1- In the C drive open data integration folder.

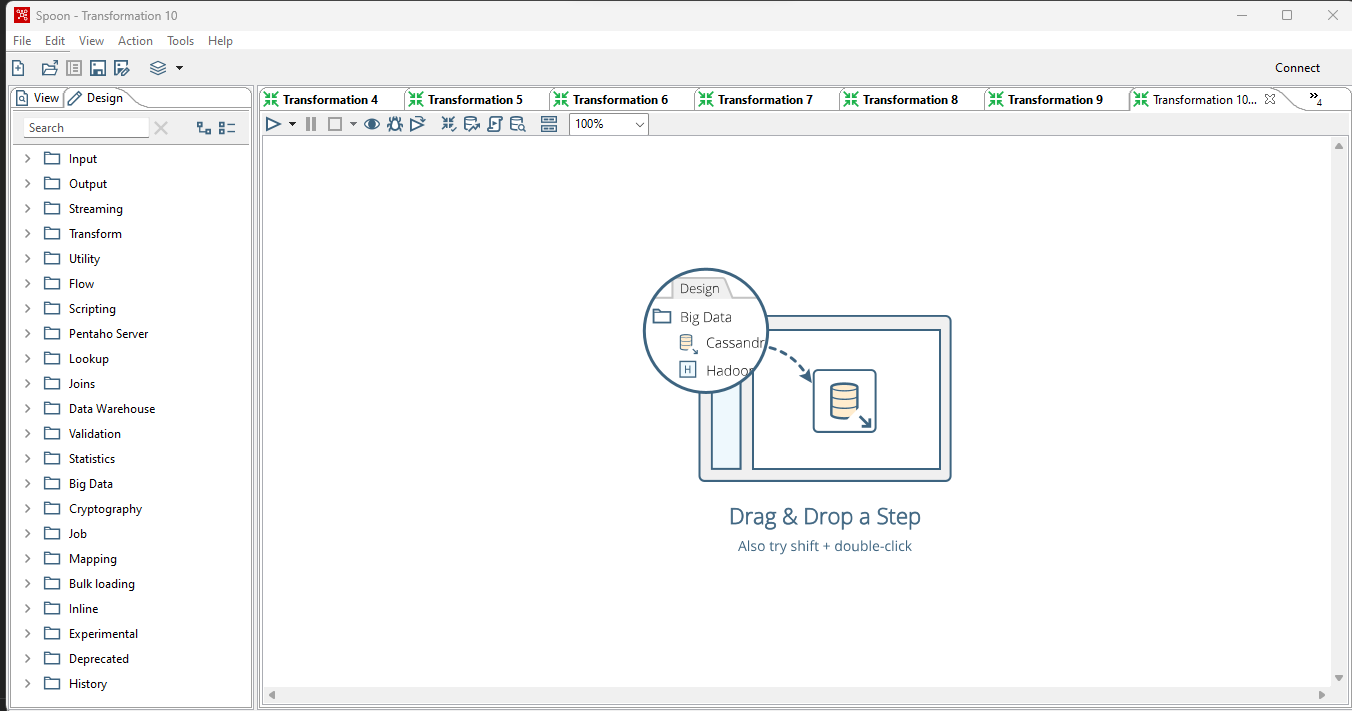
Step 2 - open “Spoon (Windows Batch File)”.

Step 3- DOUBLE CLICK ON IT

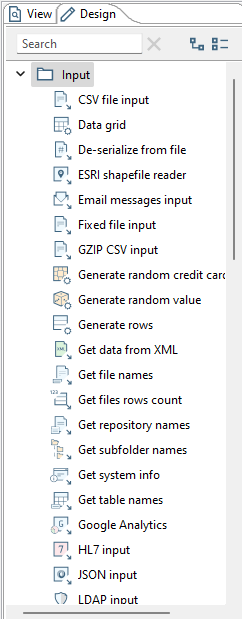


Step 4 - CLICK CLOSE ON TIPS

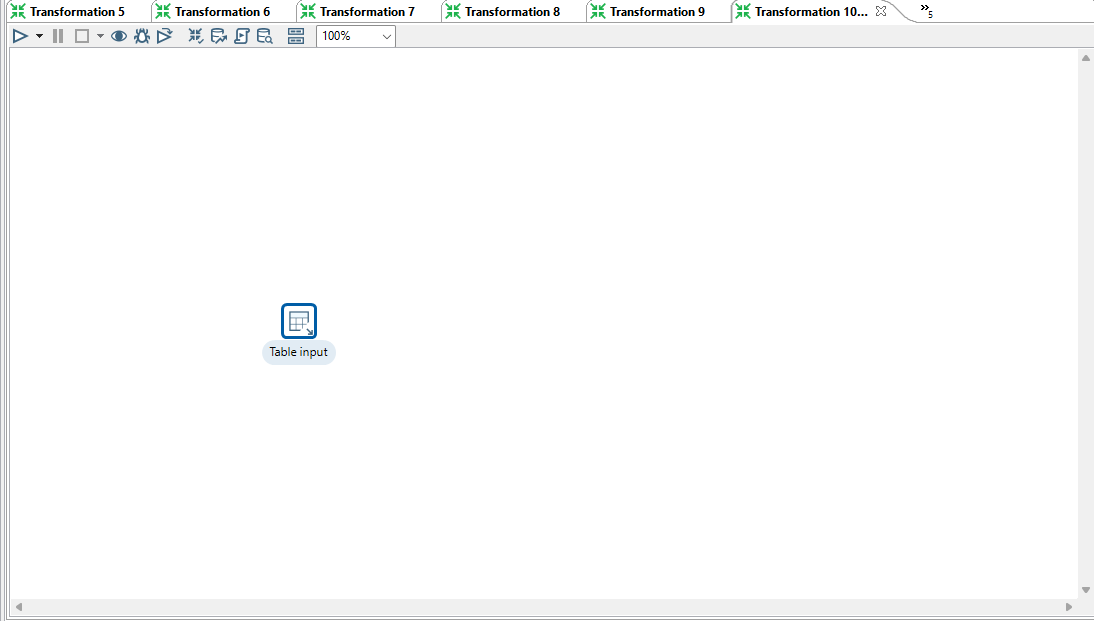
Step 5 -DOUBLE CLICK TRANSFORMATIONS



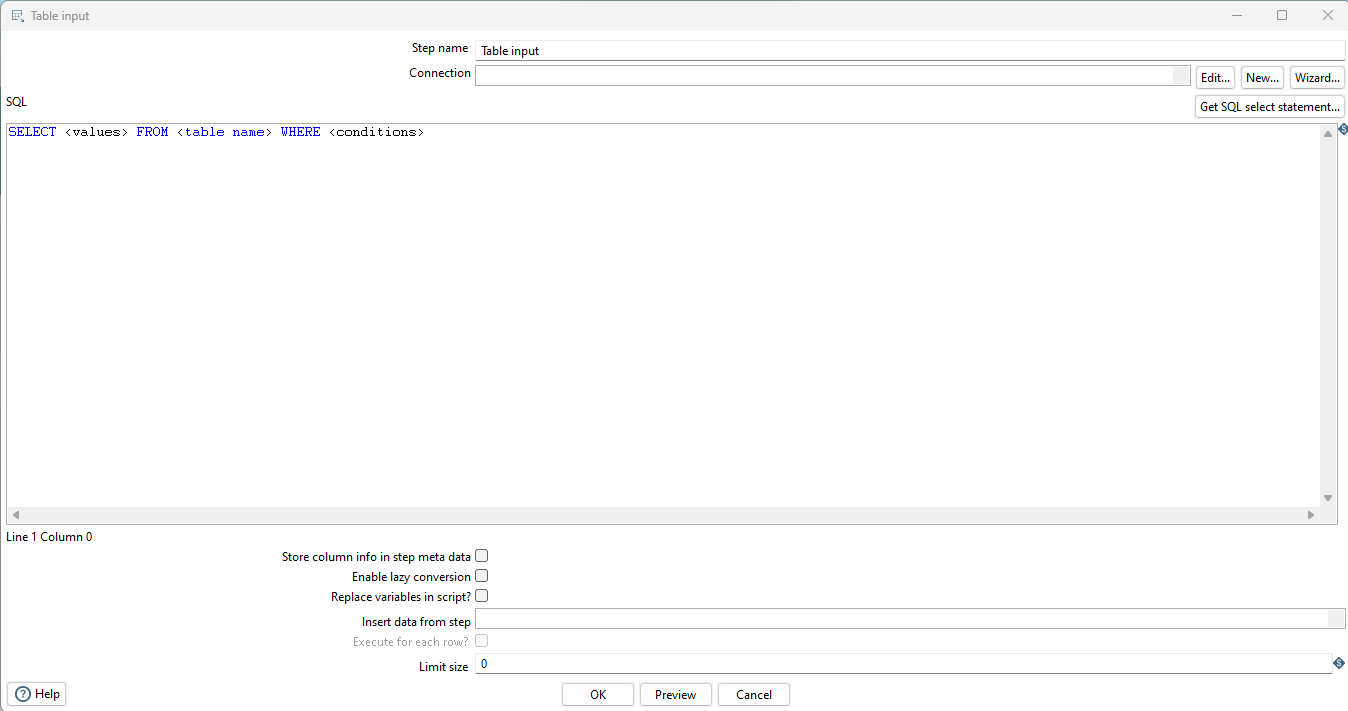
Step 6 - Open the drop down of INPUT.



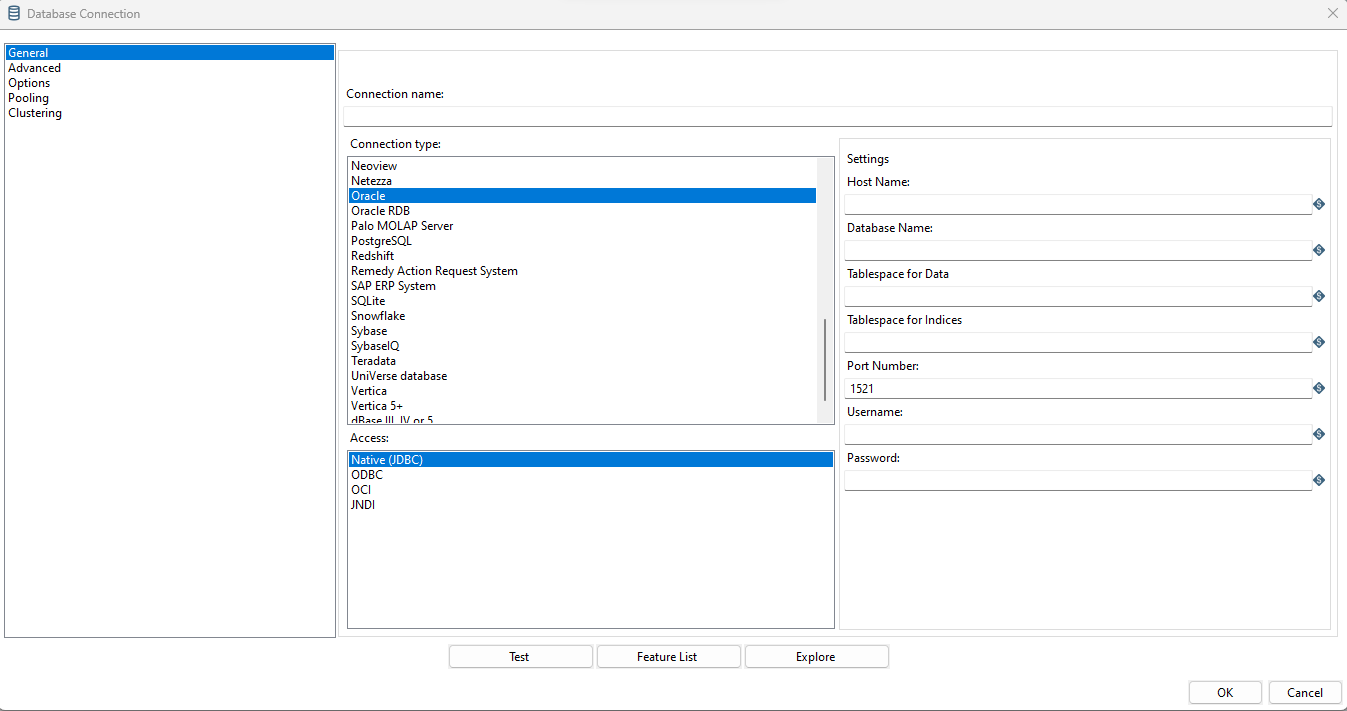
Step 7 - FIND TABLE INPUT and DRAG AND DROP IT ON THE SCREEN



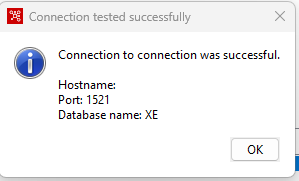
Step 8 - Double click on it.



Step 9 - CLICK ON NEW ON THE RIGHT SIDE OF CONNECTION.



Step 10 - GIVE CONNECTION NAME, DATABASE NAME, USERNAME AND PASSWORD and CLICK ON TEST



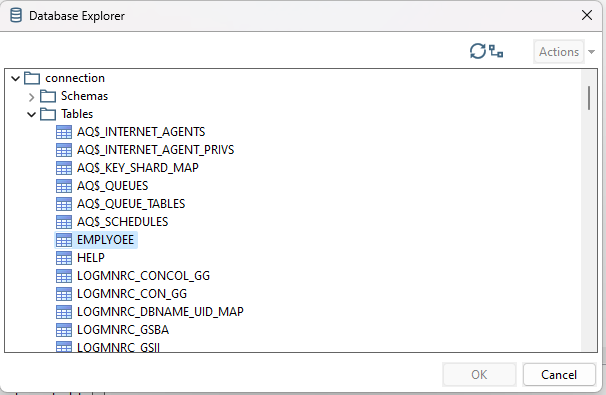
THIS MSG WILL BE DISPLAY

Step 11 - CLICK ON OK CLICK ON OK AGAIN

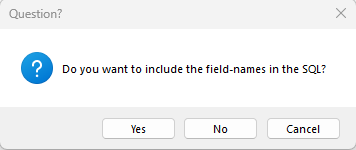
Step 12 - CLICK ON GET SQL STATEMENT.



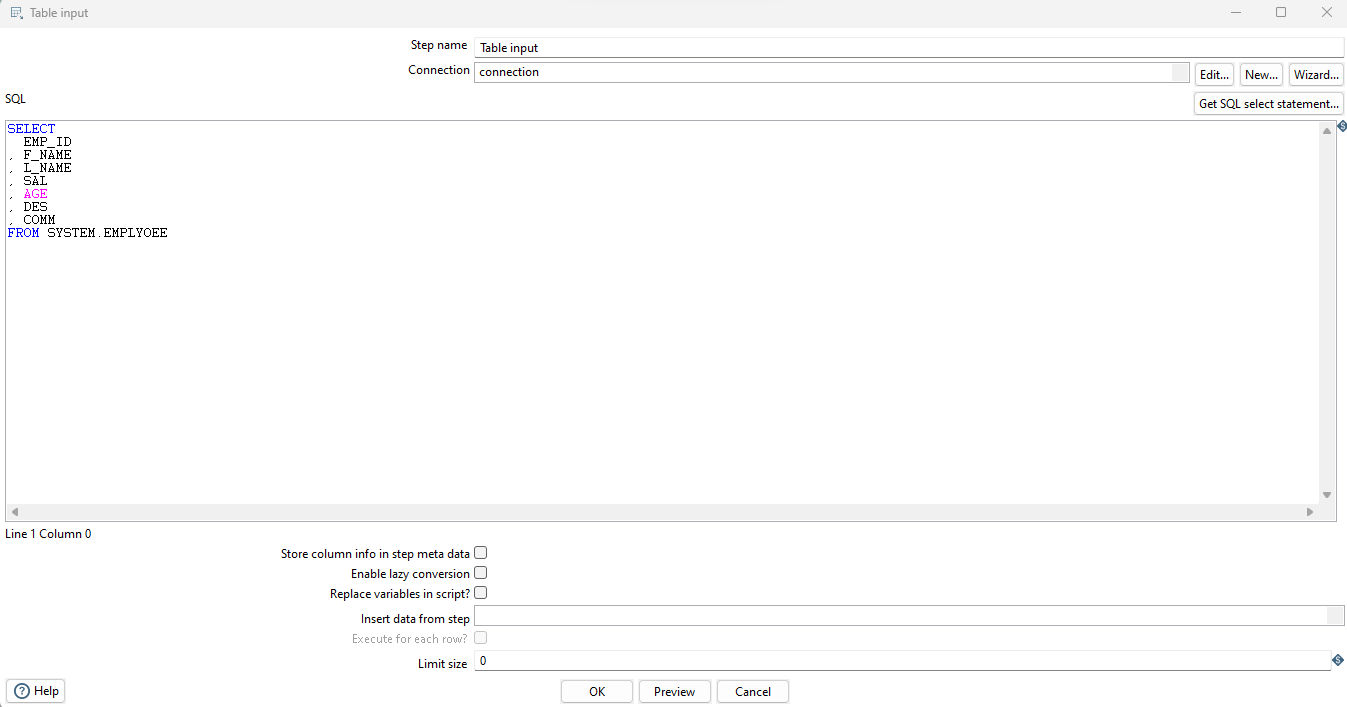
Step 13 - Open the dropdown list and find your table name.



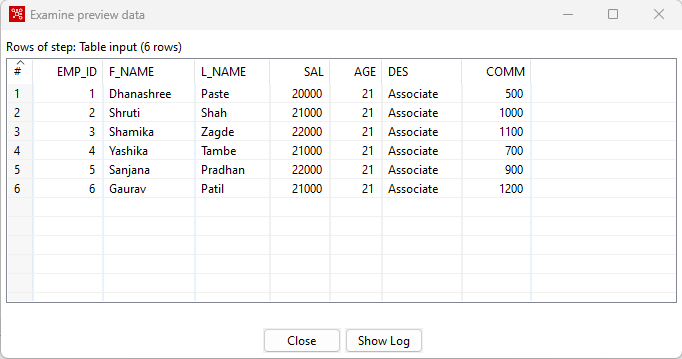
Step 14 - SELECT YOUR TABLENAME. CLICK OK AND CLICK ON GET SQL STATEMENT.



CLICK ON YES.

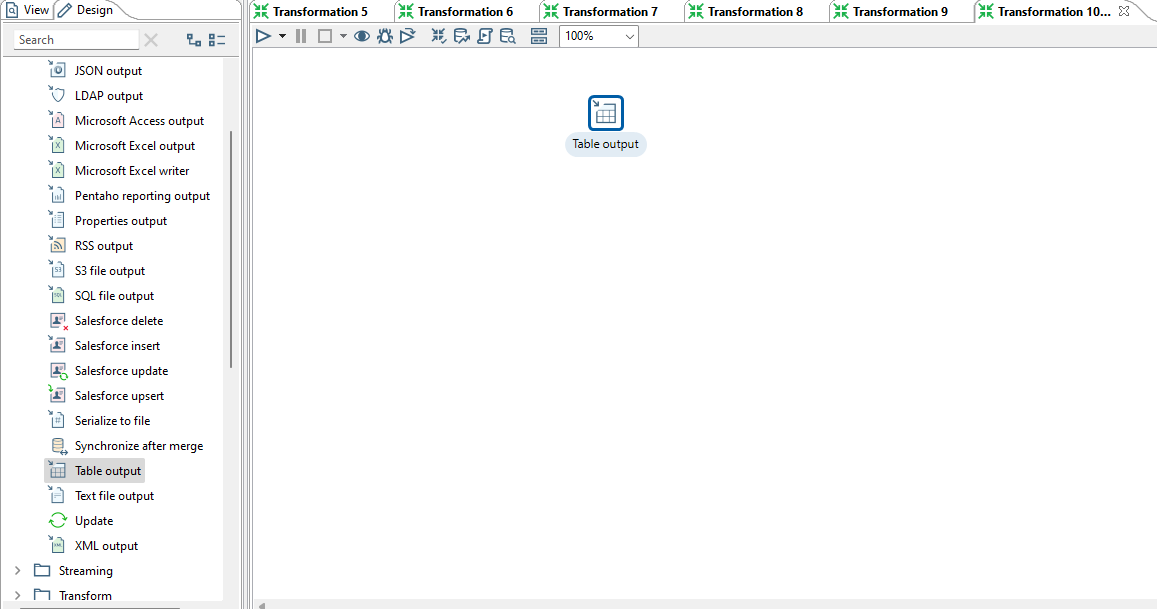


Step 15 - Click preview and Click ok.

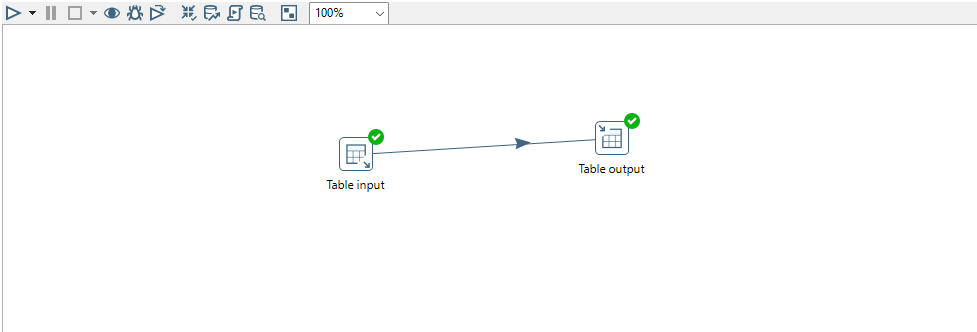


Step 16 - Click close. Click ok

Step 17 - Now open the dropdown of output and drag and drop table output.

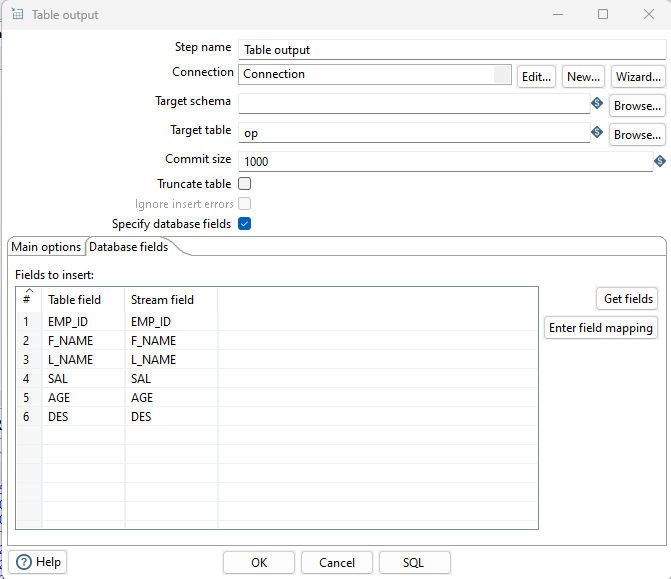


Step 18 - Hold the mouse Pointer on table input and select and drag the Output connector to the Table output.



Step 19 - Double click on it.

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



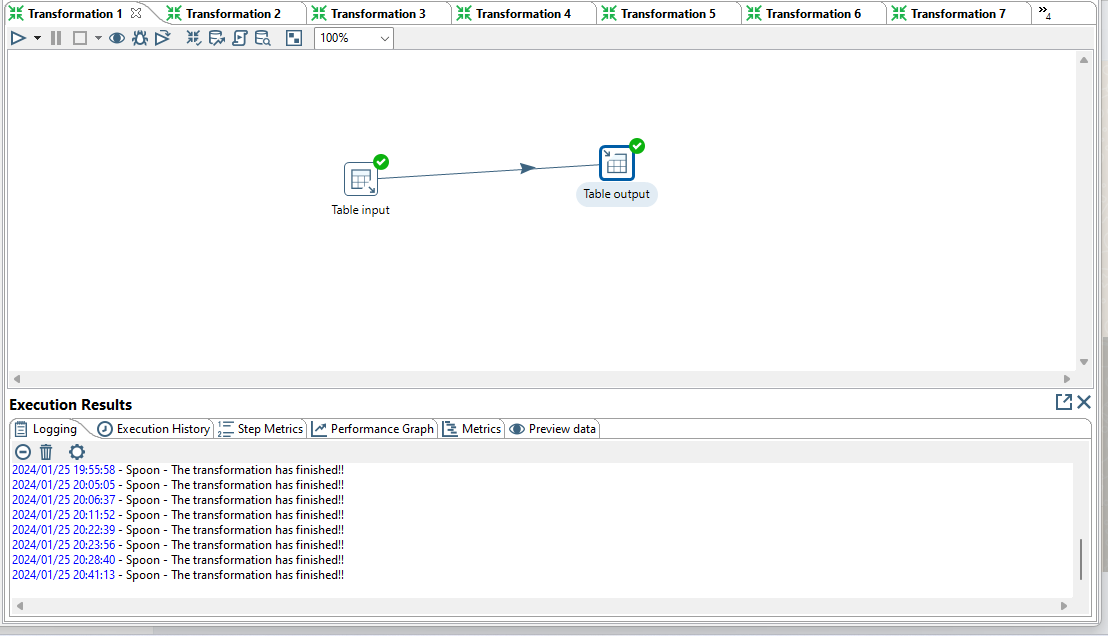
CLICK ON SQL

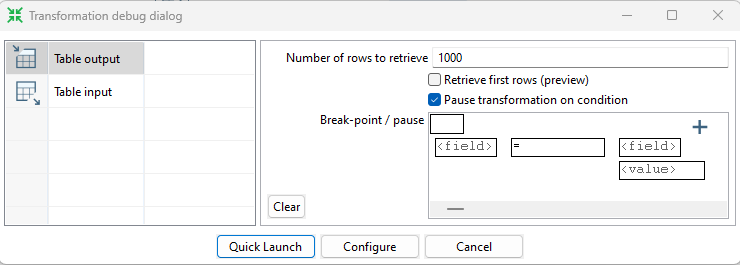
CLICK ON EXECUTE THAN OK.

CLICK ON CLOSE THAN OK.

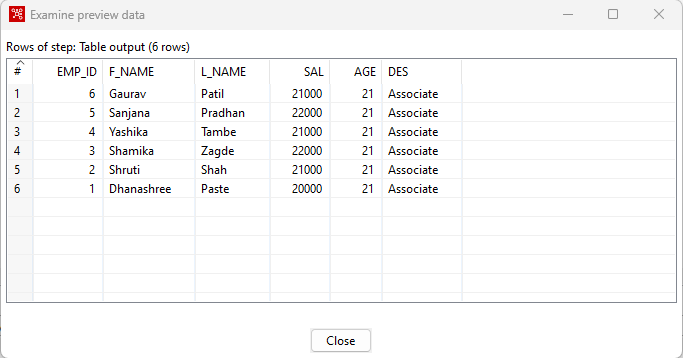
Then click on Debug this transformation.

Click on Quick Launch.

****

****

CLICK OK.



Go to SQL plus and view data from target table.

SQL> select \* from op;

EMP\_ID F\_NAME L\_NAME SAL AGE DES

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

1 Dhanashree Paste 20000 21 Associate

2 Shruti Shah 21000 21 Associate

3 Shamika Zagde 22000 21 Associate

4 Yashika Tambe 21000 21 Associate

5 Sanjana Pradhan 22000 21 Associate

6 Gaurav Patil 21000 21 Associate

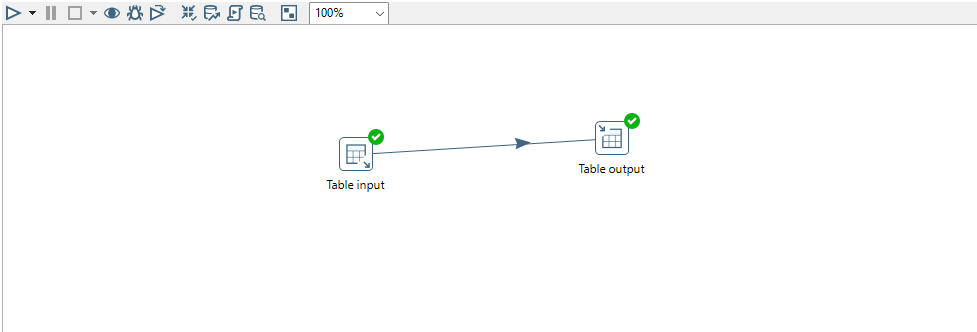
6 rows selected.

**TRANSFORMATION 2 - Sequence and Sort Transformation**

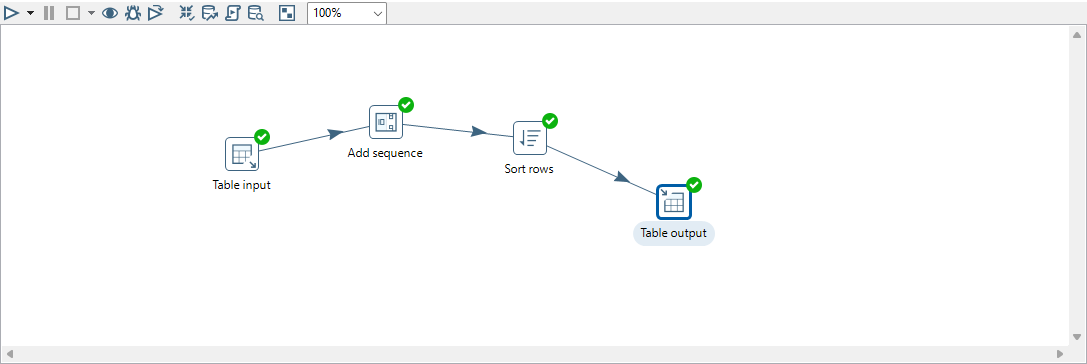
FOLLOW STEPS 1 TO 5 FROM 1ST TRANSFORMATION

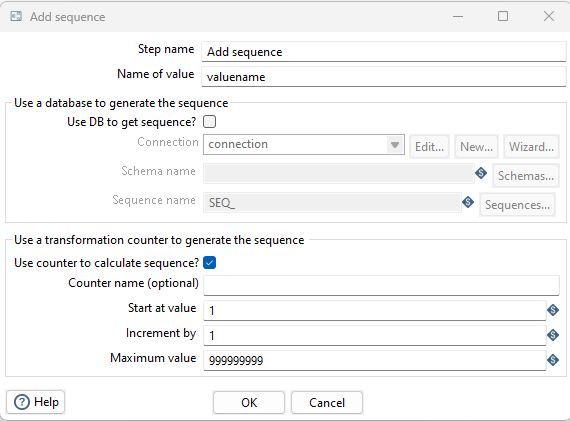
Step 2- Click on Transform and open the dropdown

Step 3- Drag Add Sequence and sort rows on the Panel.

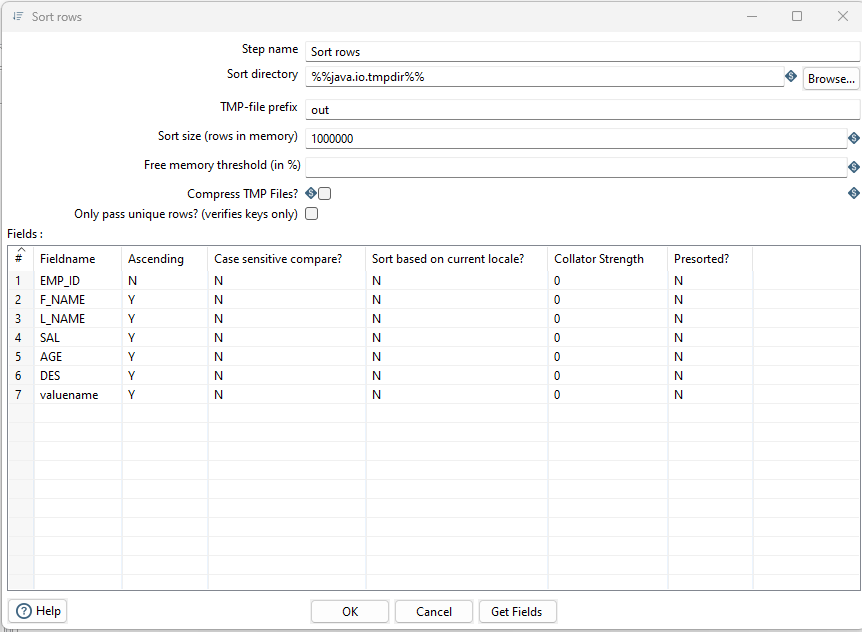


Step 4- Hold the mouse pointer on the Table Input and then drag the output connector to the sort rows and add seq.



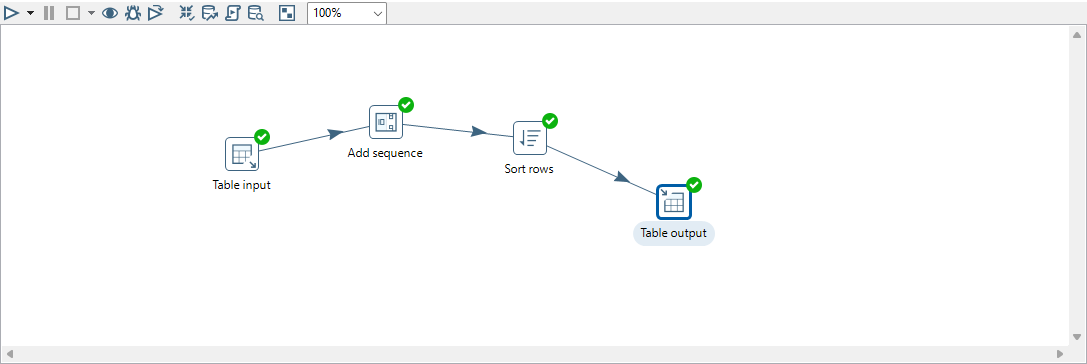


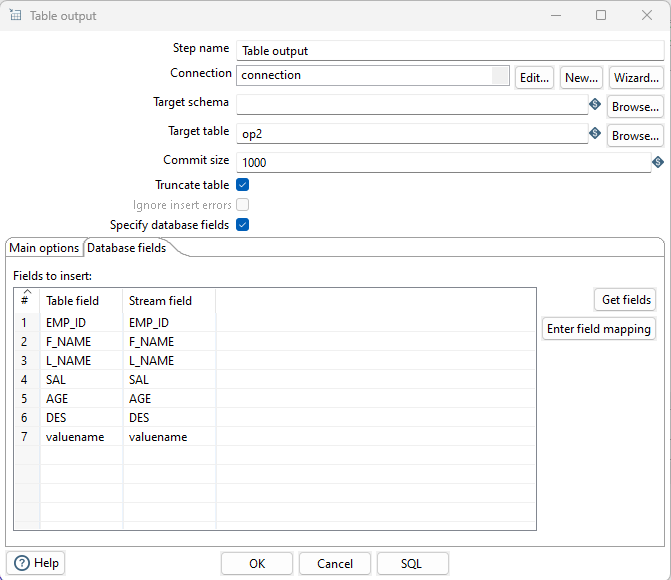
Step 5- Double click on the sort rows and than click on get fields



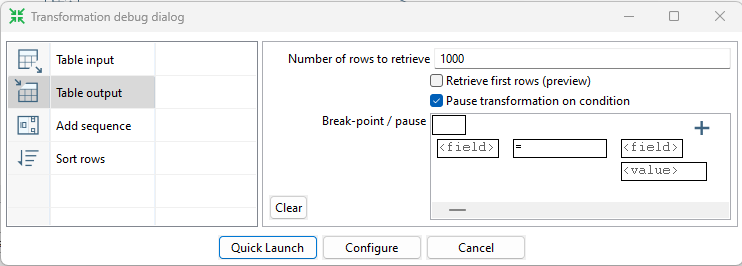
Edit the fields and press ok

For table Output Follow same steps as transformation 1.



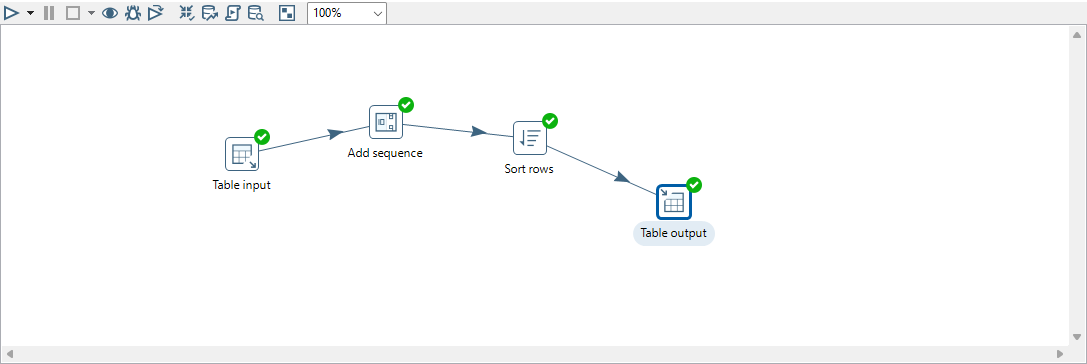
****

Step 6- Click on Debug the Transformation and Click on Quick Launch.



CLICK OK.

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



Go to SQL plus and view data from target table.

SQL> select \* from op2;

EMP\_ID F\_NAME L\_NAME SAL AGE DES VALUENAME

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

6 Gaurav Patil 21000 21 Associate

6

5 Sanjana Pradhan 22000 21 Associate

5

4 Yashika Tambe 21000 21 Associate

4

3 Shamika Zagde 22000 21 Associate

3

2 Shruti Shah 21000 21 Associate

2

1 Dhanashree Paste 20000 21 Associate

1

6 rows selected.

**Transformation 3- Concat the Fields**

Follow Step 1 to step 5 from 1st Transformation

Step 2 - Drag and drop the concat fields and double click on concat fields.

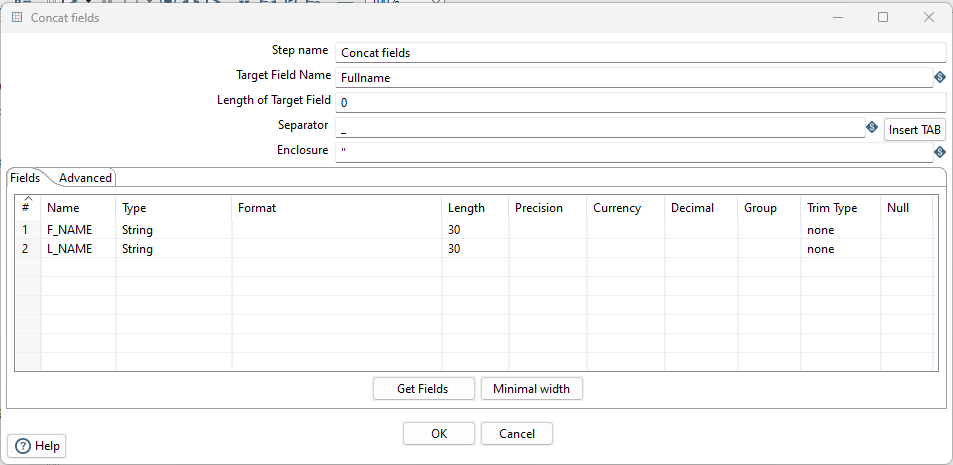
Write target step name as Full Name.

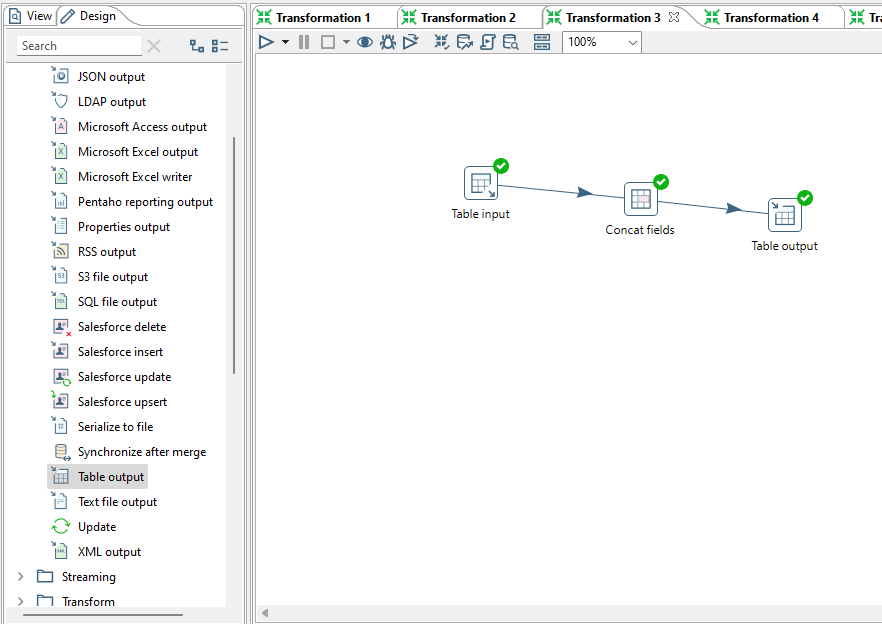
Click on get fields -> Fname and Lname are kept and others are discarded and their datatype

would be STRING. They are separated by ‘;’ and enclosed by ‘ ” ’. Then OK.

Drag and drop table output and double click on it. Check the boxes and select the tab database fields.

Click on get fields than SQL than Execute and lastly ok

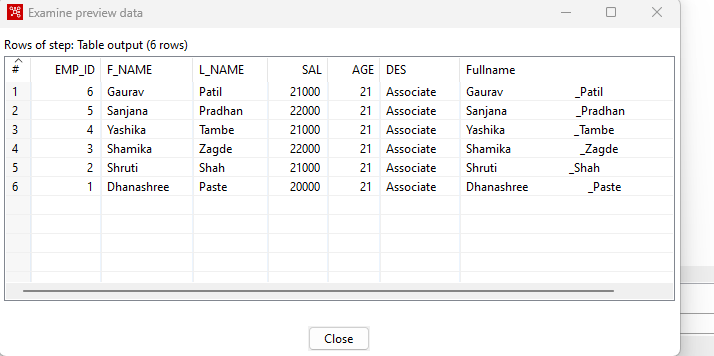




Click on spider icon for QUICK LAUNCH.

Go to SQL plus and view data from target table.

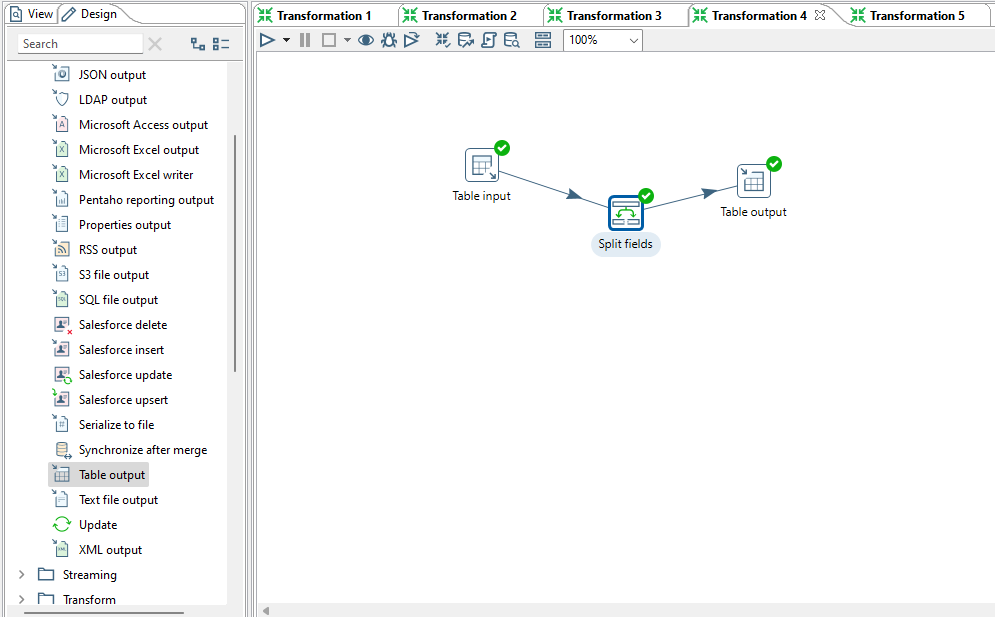
SQL> select \* from op3;

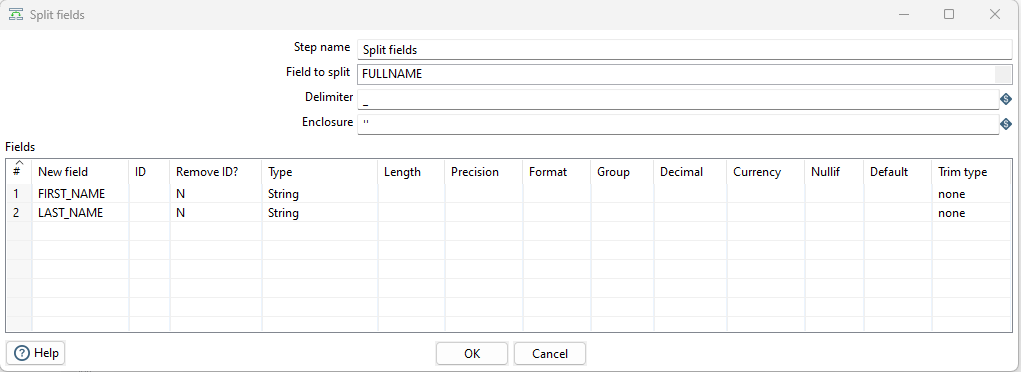


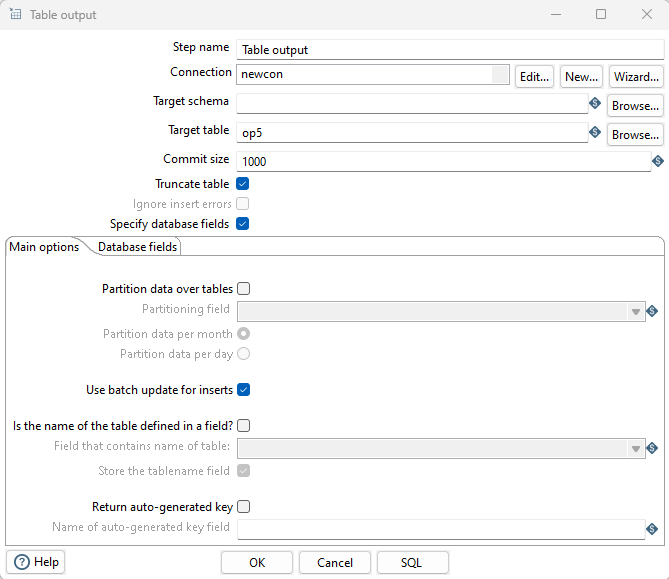
**Transformation 4 - Split rows**

Follow step 1 to 5 from 1st Transformation

**Step 6-** Drag and drop the split fields and Double click on it . Field split is use to split Full Name into first name and last name. Giving their Datatype and length of string.

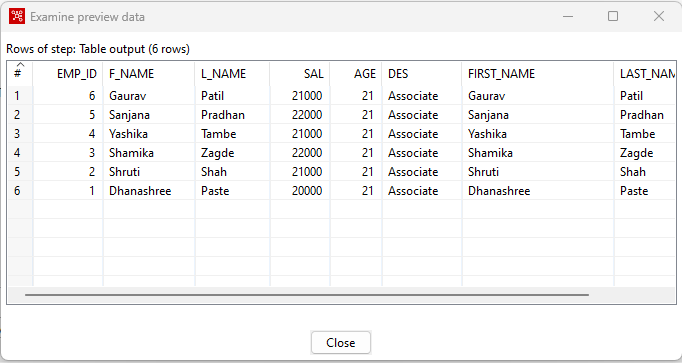






Drag and drop the table output. Double click on it. Write the target table name check the boxes and select the database fields Get fields

SQL> select \* from op5;

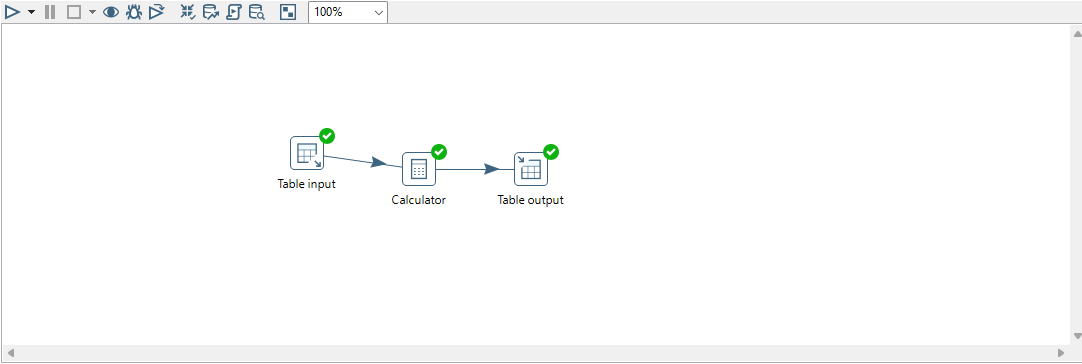


**TRAMSFORMATION 5- Calculator**

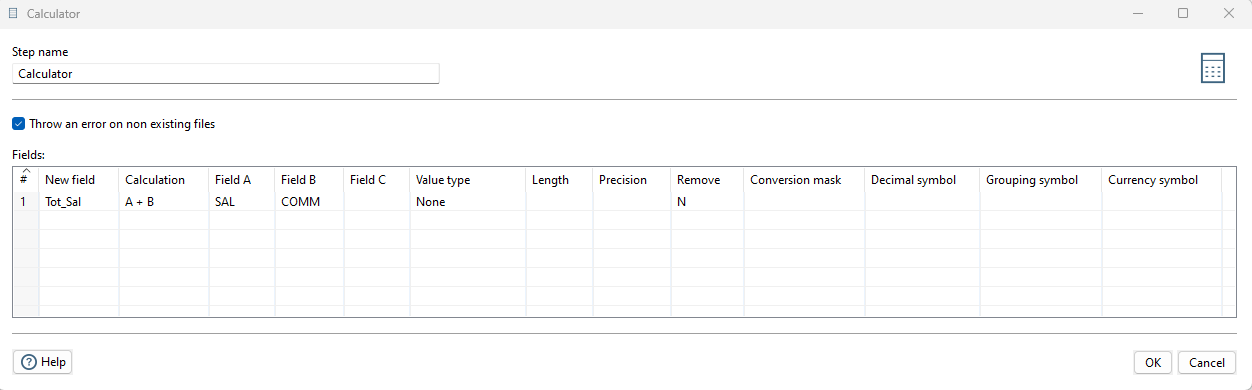
Repeat Steps 1 to 3 from TRANSFORMATION 1.

Drag and drop *Calculator* from Transformfolder .

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

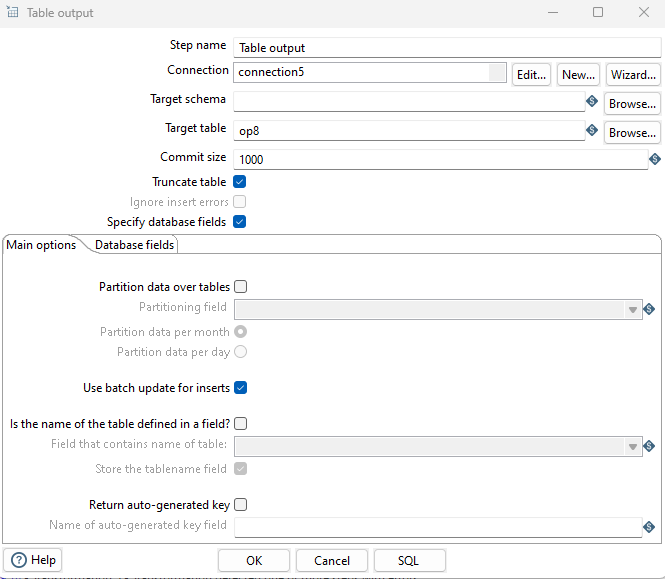


Double Click on *Calculator* and fill in the details as shown below.



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

Click on OK.

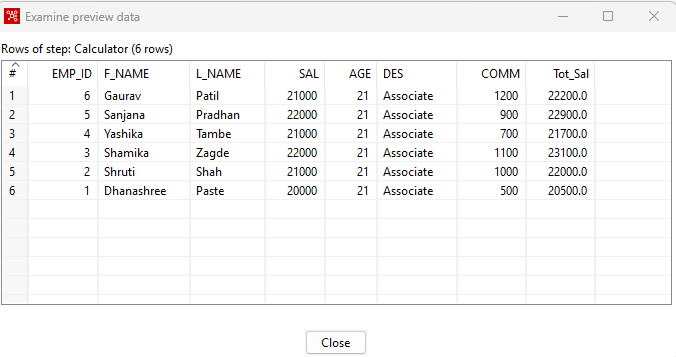


Step 5-Repeat Step 4 from TRANSFORMATION 1.

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

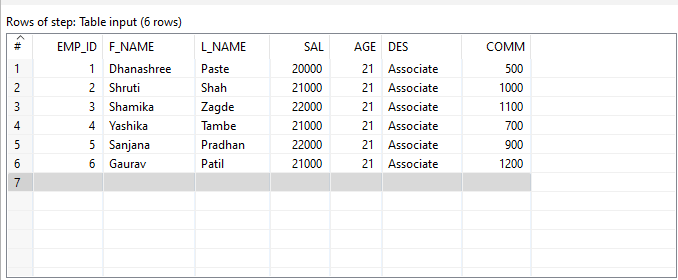
Step 6-Run SQL query.

SQL> SELECT \* FROM op8;



**TRANSFORMATION 6- Number Range**

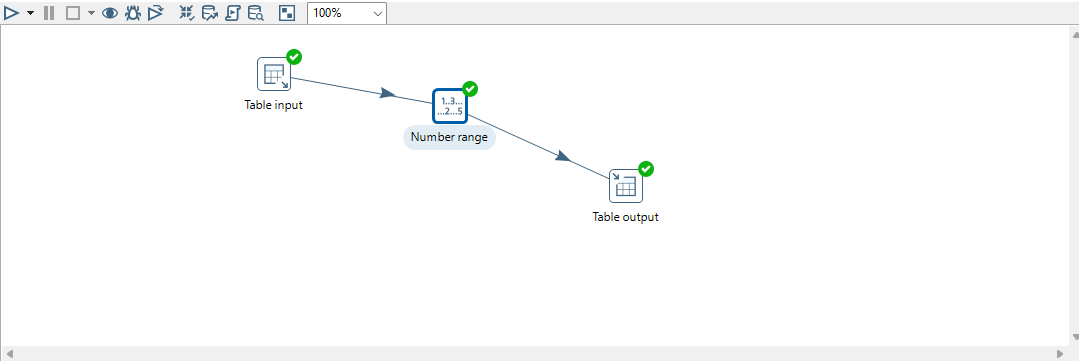
Input –



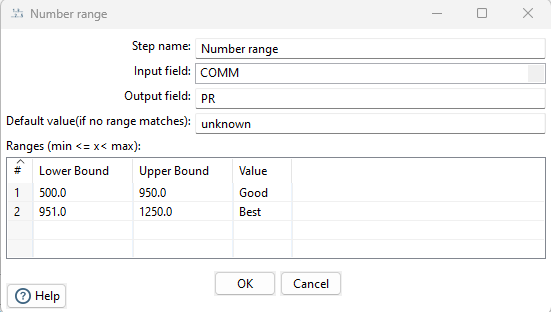
Step 1- Repeat Steps 2 and 3 from TRANSFORMATION 1.

Step 2- Perform Transformation. Drag and drop Number Rangefrom Transformfolder.

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

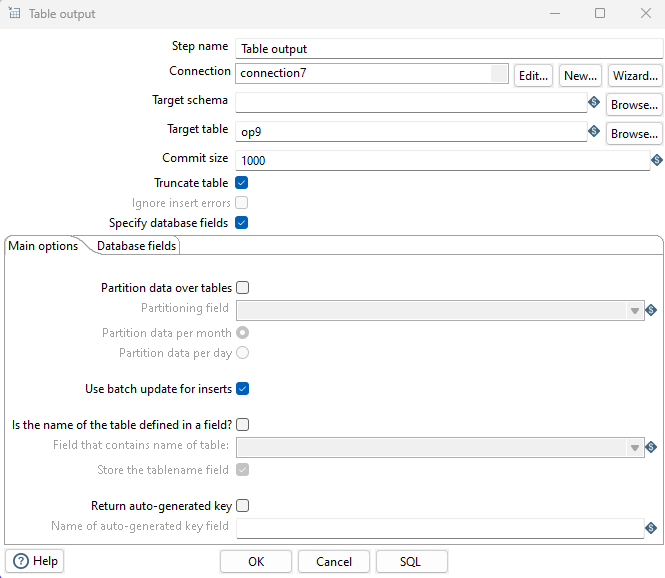


Double Click on Number range.

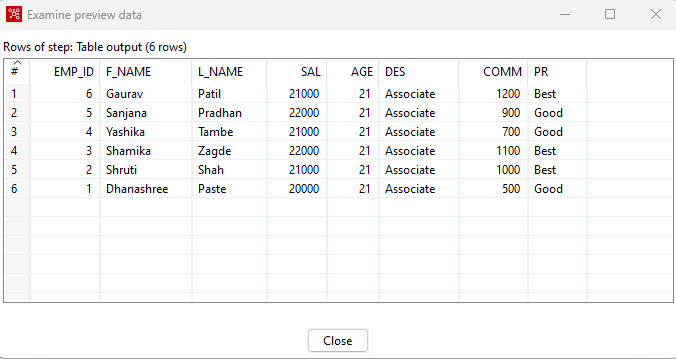


**Step 3-** Repeat Step 4 from TRANSFORMATION 1.

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



Step 4- Run SQL query.

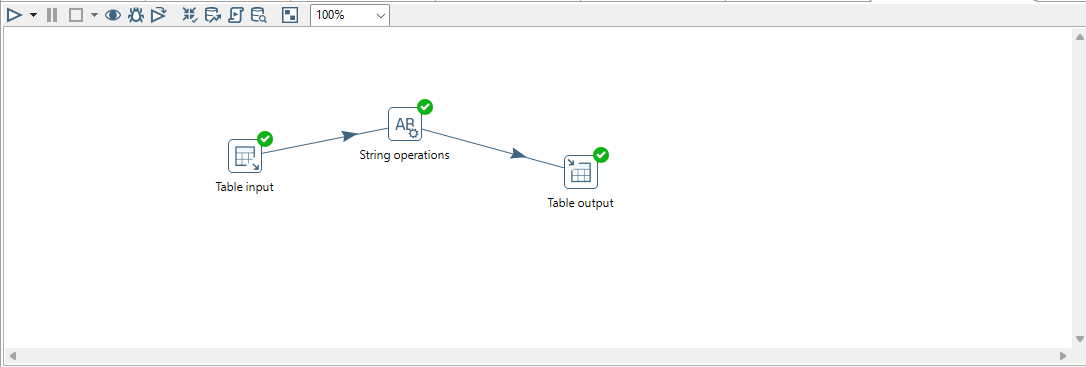


**TRANSFORMATION 7- String Operations**

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

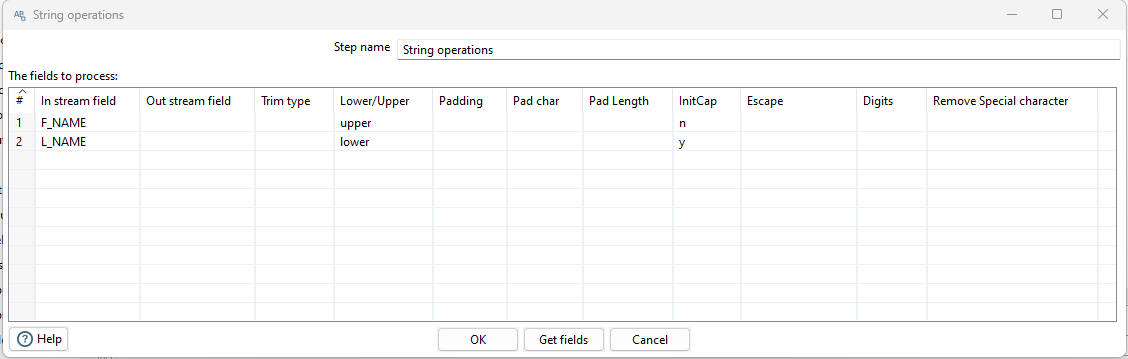
Step 2- Perform Transformation.

Drag and drop Number Range from Transformfolder.



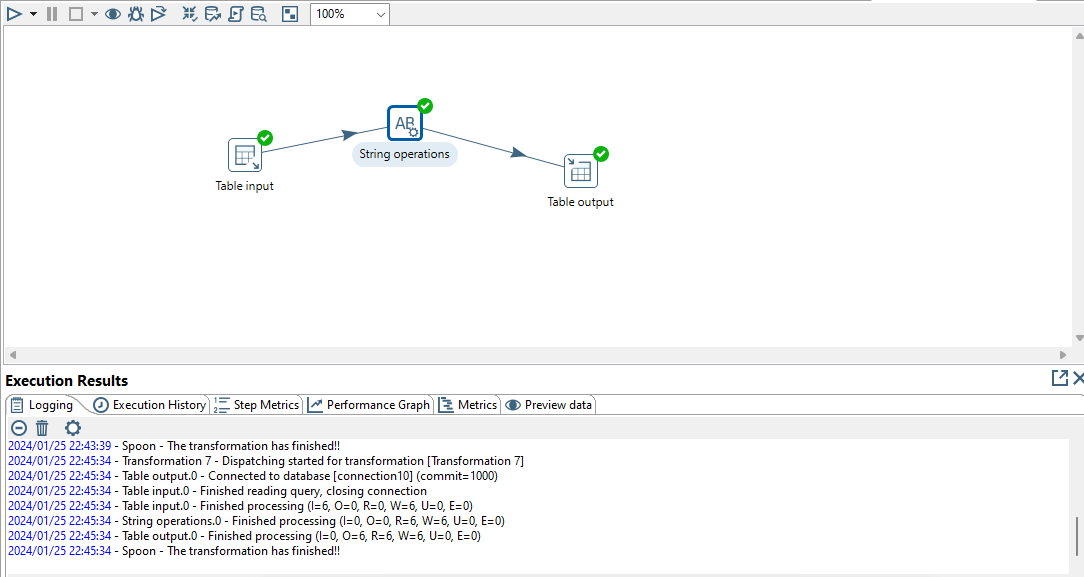
Hold the mouse Pointer on Table input and select and drag the output connector to the String operations.

Double Click on String operations.

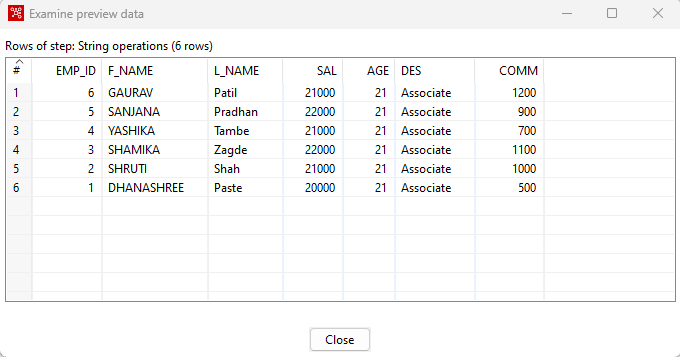


Step 3- Repeat Step 4 from TRANSFORMATION 1.

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

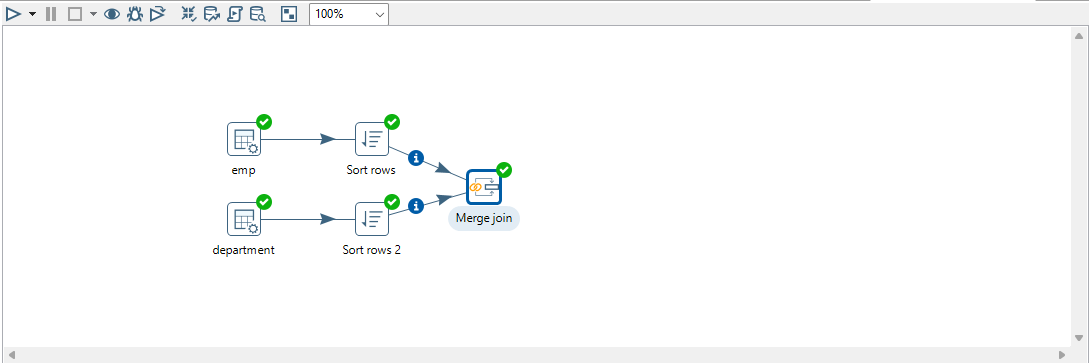


Step 4- Run SQL query.

****

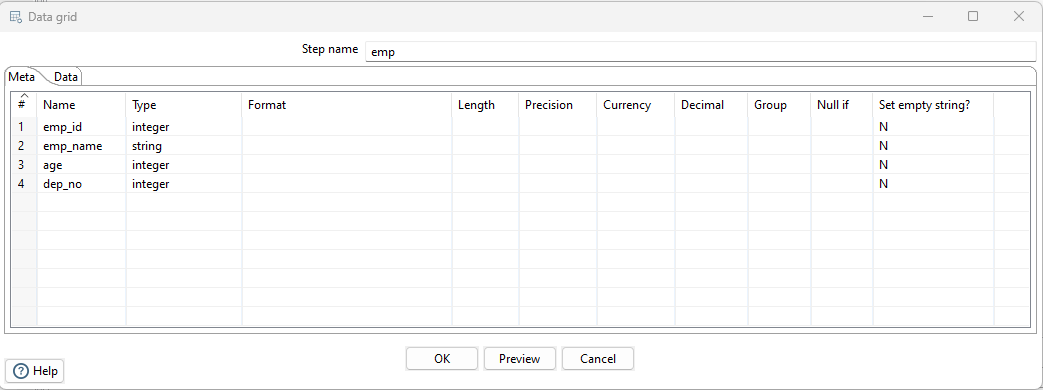
**TRANSFORMATION 8- Merge Join**

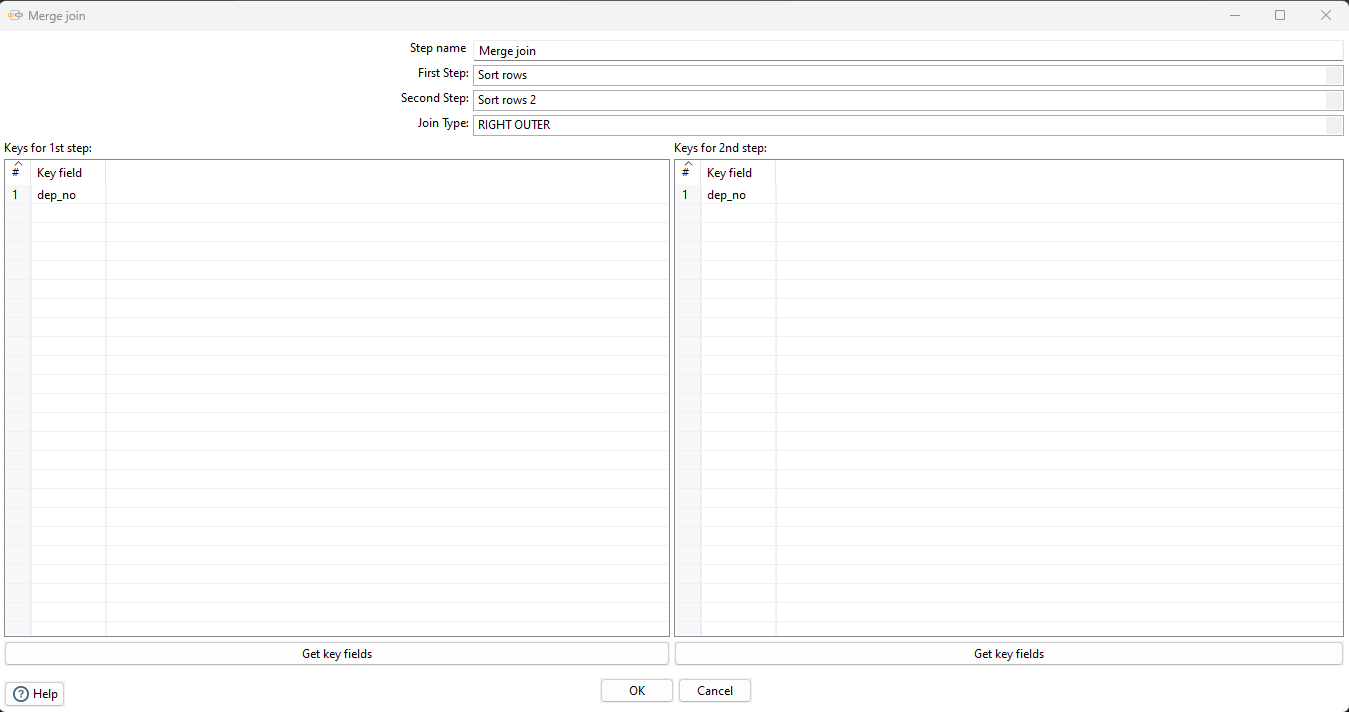
Step 1- Drag and drop 2 Data Gridfrom Input folder



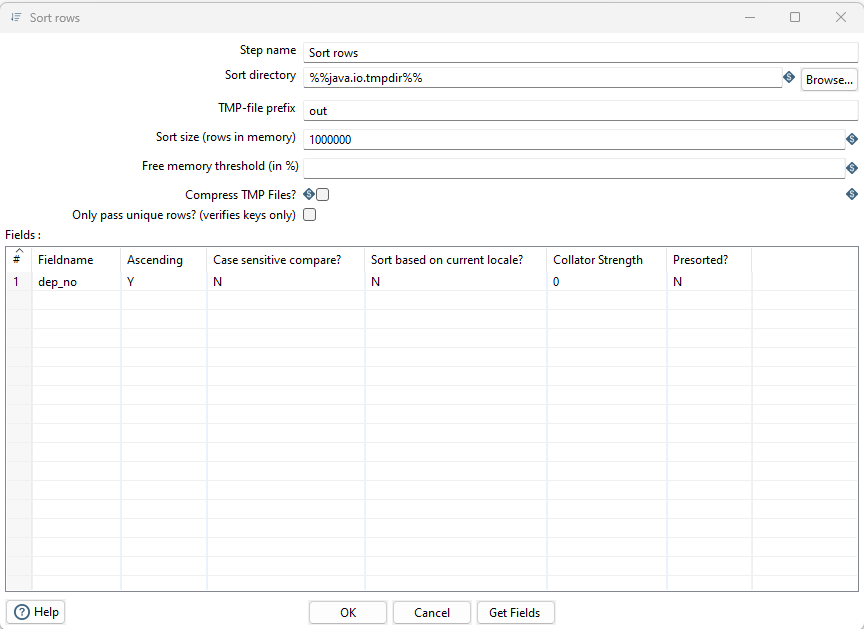
Rename them as Employee and Department.

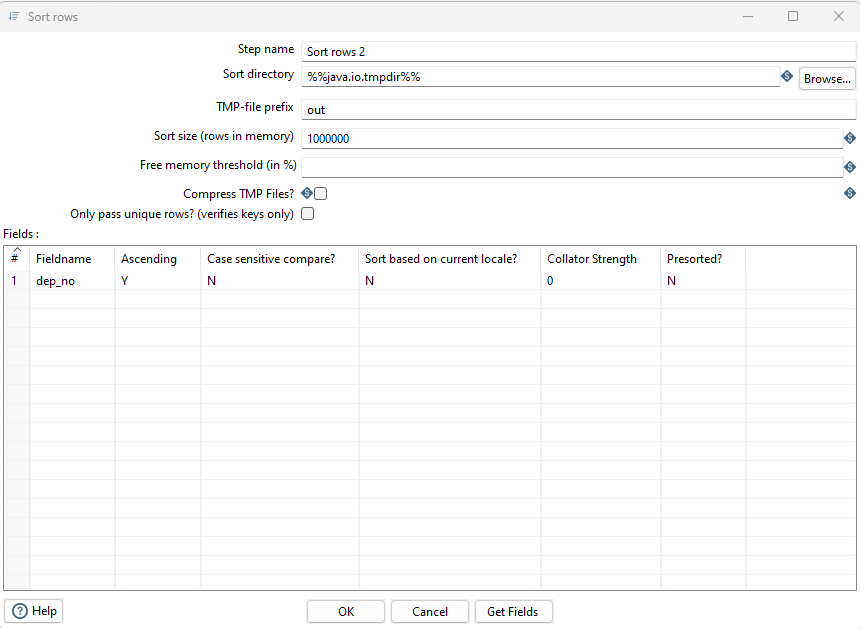
Step 2- Double click on them and insert records into respective grids. Click on OK.





Step 3- Perform Sort rows transformation for both data grids respectively. Click on OK.



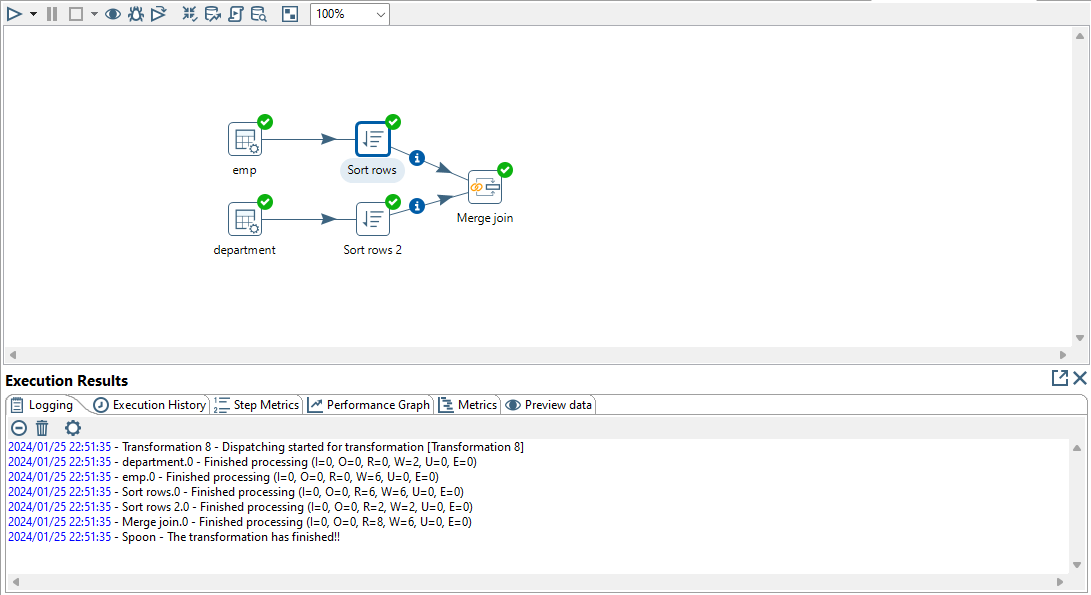


Step 4- Drag and Drop Merge join from joins folder

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

Step 5- Double click on Merge join and fill in the details as shown below to perform INNER join. Click on OK.

Debug the transformation and perform Quick launch.



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

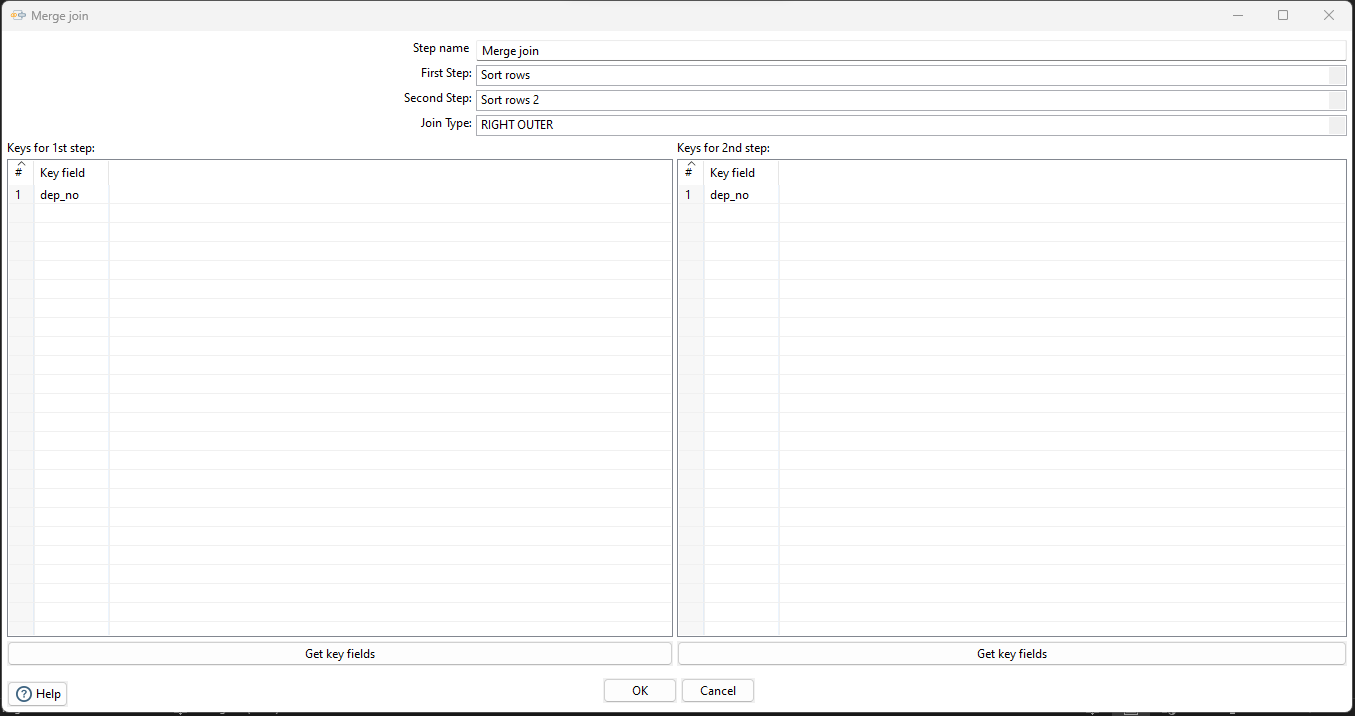
Debug the transformation and perform Quick launch

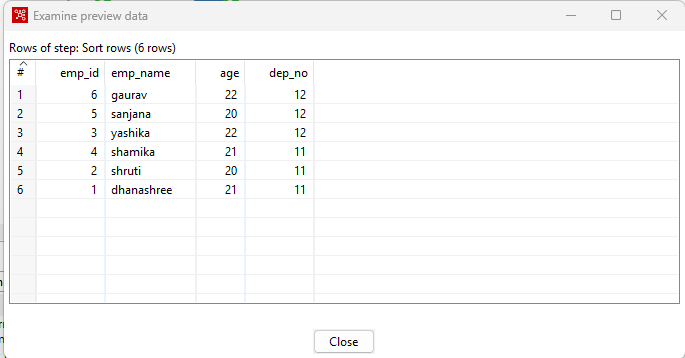
Step 7- Double click on Merge join and fill in the details as shown below to perform RIGHT OUTER join. Click on OK.

Debug the transformation and perform Quick launch

Step 8- Double click on Merge join and fill in the details as shown below to perform FULL OUTER join. Click on OK.

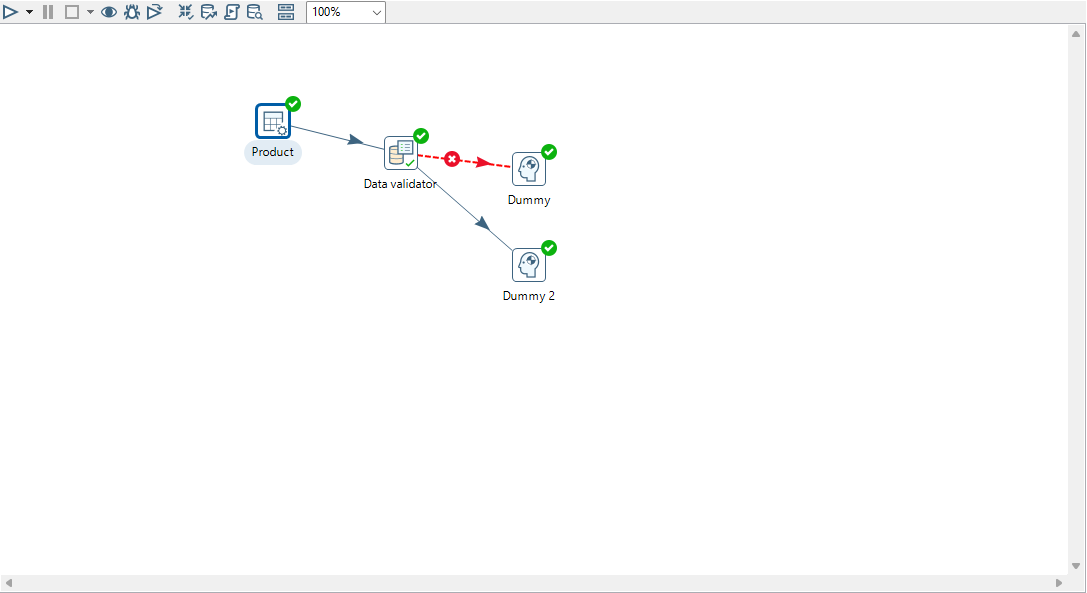
Debug the transformation and perform Quick launch





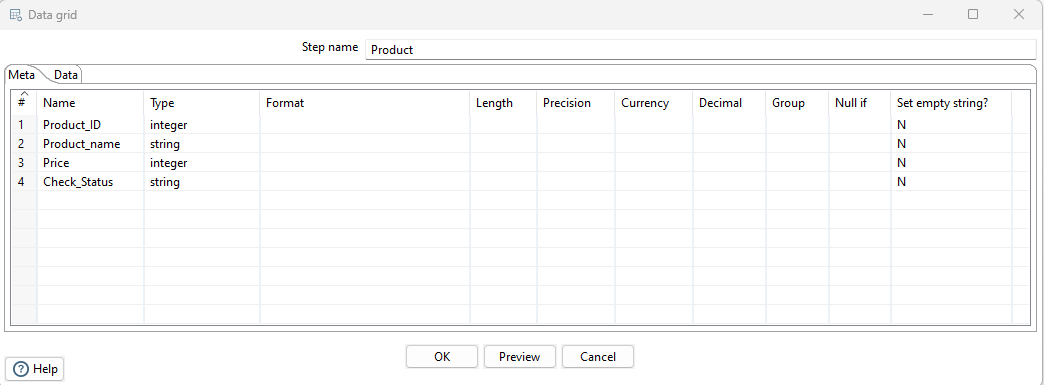
**TRANSFORMATION 9- Data validations**

Step 1- Drag and drop Data Gridfrom Input

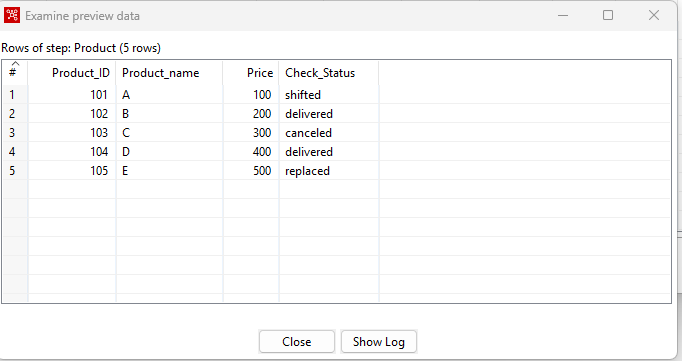


Rename it as Product.

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



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

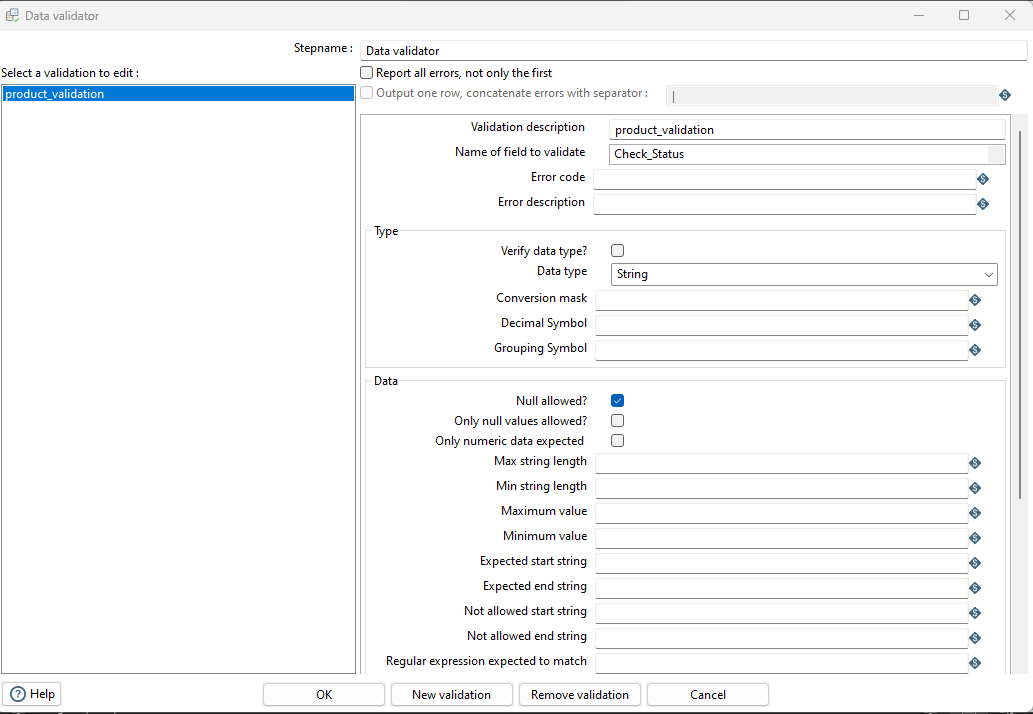


Hold the mouse pointer on Product data grid and select and drag the output connector to the Data validator.

Double click on Data validator. New Validation.

Give Validation Name and click OK.

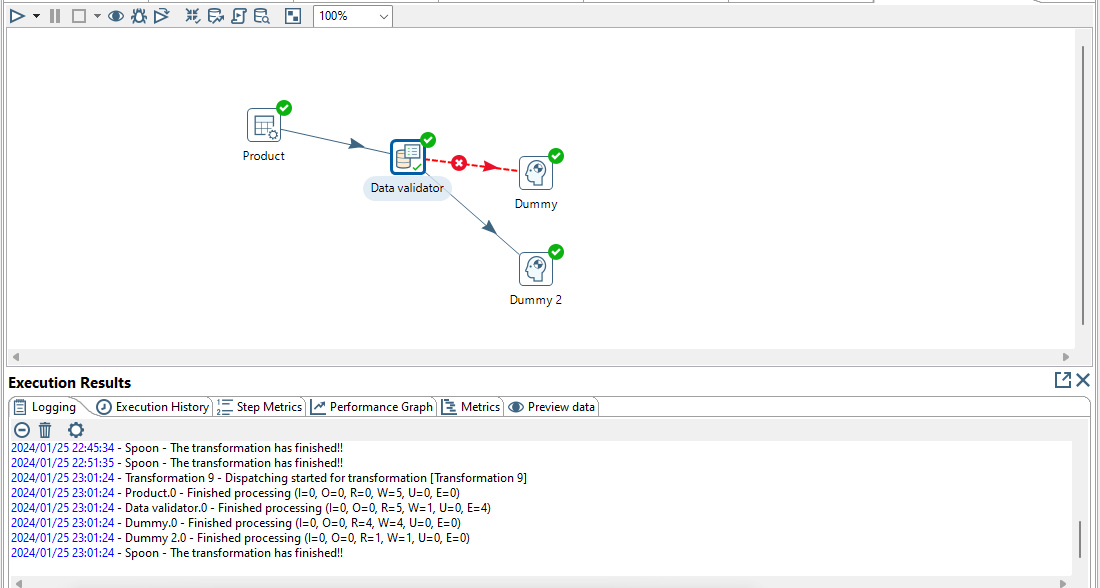
select the validation to edit .



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

Click on OK.

Step 4- Drag and drop Dummy from Flow



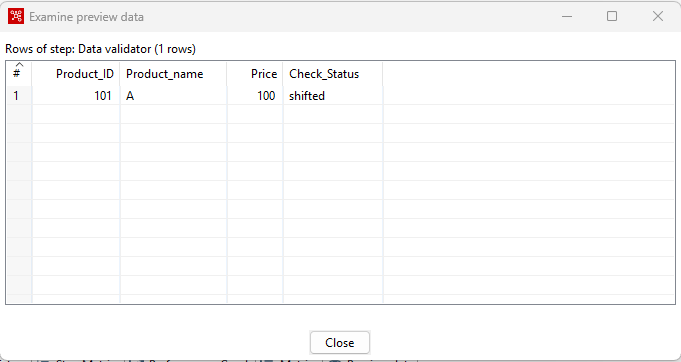
Hold the mouse pointer on Data validator and select and drag the output connector to the Dummy. Select Main output of step.



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



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



**PRACTICAL NO. 5**

**AIM -** **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- Setw(), getw(), data(),rm()**

**Attaching and Detaching data**

**Reading data from the console**

**Loading data from different data sources(CSV,Excel)**

**Install packages, Loading packages**

> install.packages("dplyr")

> library(dplyr)

**Data types, checking variable type, printing variable and objects (Vector, Matrix, List, Factor, Data frame, Table)**

> x=5

> print(x)

[1] 5

> class(x)

[1] "numeric"

> y="c"

> print(y)

[1] "c"

> is.character(y)

[1] TRUE

> is.integer(y)

[1] FALSE

**Vector operations**

> x=c(11,27,33)

> y=c(4,5,6)

> x\*y

[1] 44 135 198

> x-y

[1] 7 22 27

> x+y

[1] 15 32 39

**Cbinding(column bind) and rbinding(row bind)**

> x=c(1,2,3)

> y=c(4,5,6)

> cbind(x,y)

x y

[1,] 1 4

[2,] 2 5

[3,] 3 6

> rbind(x,y)

[,1] [,2] [,3]

x 1 2 3

y 4 5 6

**Matrix operations**

**# using matrix() function**

> m= matrix(c(1,2,3,4,5,6,7,8,9),nrow = 3,ncol=3)

> m

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

> dim(m)

[1] 3 3

> m= matrix(c(1,2,3,4,5,6,7,8,9),nrow = 3,ncol=3,byrow = TRUE)

> m

[,1] [,2] [,3]

[1,] 1 2 3

[2,] 4 5 6

[3,] 7 8 9

**# matrix multiplication**

> p=3\*m

> p

[,1] [,2] [,3]

[1,] 3 6 9

[2,] 12 15 18

[3,] 21 24 27

> n=matrix(c(4,5,6,13,14,15,24,25,26),nrow = 3,ncol = 3)

> q=m+n

> q

[,1] [,2] [,3]

[1,] 5 15 27

[2,] 9 19 31

[3,] 13 23 35

> mdash = t(m)

> mdash

[,1] [,2] [,3]

[1,] 1 4 7

[2,] 2 5 8

[3,] 3 6 9

**Determinant**

> s = matrix(c(2,3,4,14,15,16,21,22,23),nrow=3,ncol=3,byrow = TRUE)

> s\_det = det(s)

> s\_det

[1] 2.109424e-15

> r\_det = det(m)

> r\_det

[1] 6.661338e-16

**Dataframe**

> student\_id = c(1,2,3)

> student\_names = c("Mina","Rina","Seema")

> position = c("First","Second","Third")

> data = data.frame(student\_id,student\_names,position)

> data

student\_id student\_names position

1 1 Mina First

2 2 Rina Second

3 3 Seema Third

> data$student\_id

[1] 1 2 3

> names(data)

[1] "student\_id"

[2] "student\_names"

[3] "position"

**2-Dimensional table in R**

> 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

**Reading and Writing data From csv**

> getwd()

[1] "C-/Users/Dell/OneDrive/Documents"

> emo<-read.csv("new.csv")

> emo

X a b pi

1 1 12 45 3.142

> data <- read.table("new.csv",sep = ",",header = T)

> data

employee salary

1 mina 5000

2 rina 10000

3 seema 15000

#dimension

> dim(data)

[1] 3 2

> head(data,2)

employee salary

1 mina 5000

2 rina 10000

> tail(data,2)

employee salary

2 rina 10000

3 seema 15000

**Reading and Writing data from Excel.**

> install.packages("readxl")

> library(readxl)

> excel\_file <-"C-/Users/Dell/OneDrive/Documents/excel.xlsx"

> data <- read\_excel(excel\_file)

> data

# A tibble- 5 × 3

rollno name marks

*<dbl>* *<chr>* *<dbl>*

1 1 shruti 90

2 2 riya 80

3 3 sakshi 60

4 4 mira 94

5 5 sameer 85

**Write file in directory**

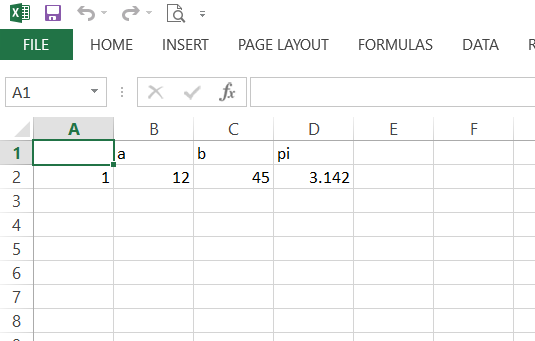
> z=data.frame(a=12,b=45,pi=3.142)

> print(z)

a b pi

1 12 45 3.142

> write.csv(z,file = "new.csv")



**PRACTICAL NO. 6**

**AIM- Data preprocessing techniques in R-  
Naming and Renaming variables**

**Adding a new variables**

**Dealing with missing value**

**Dealing with categorical data**

**Data reduction using subsetting**

>install.packages(“dplyr”)

setwd(“F-/R”)

getwd()

> data <- data.frame(ID=1-5,Name=c("smit","john","bob","alice","charlie"),Age=c(25,30,22,NA,28),Score=c(85,92,78,65,89))

> data

ID Name Age Score

1 1 smit 25 85

2 2 john 30 92

3 3 bob 22 78

4 4 alice NA 65

5 5 charlie 28 89

**## Renaming columns with dplyr--rename()**

require(dplyr)

> data = rename(data, UserID=ID,FULLNAME=Name)

> data

UserID FULLNAME Age Score

1 1 smit 25 85

2 2 john 30 92

3 3 bob 22 78

4 4 alice NA 65

5 5 charlie 28 89

> data$Grade <- ifelse(data$Score >= 90, "A", ifelse(data$Score >= 80, "B", "C"))

> data

UserID FULLNAME Age Score Grade

1 1 smit 25 85 B

2 2 john 30 92 A

3 3 bob 22 78 C

4 4 alice NA 65 C

5 5 charlie 28 89 B

> data <- na.omit(data)

> data

UserID FULLNAME Age Score Grade

1 1 smit 25 85 B

2 2 john 30 92 A

3 3 bob 22 78 C

5 5 charlie 28 89 B

> data$Gender <- factor(c("Male", "Female", "Male", "Male"))

> data

UserID FULLNAME Age Score Grade Gender

1 1 smit 25 85 B Male

2 2 john 30 92 A Female

3 3 bob 22 78 C Male

5 5 charlie 28 89 B Male

> colnames(data)

[1] "UserID" "FULLNAME" "Age"

[4] "Score" "Grade" "Gender"

Adding new variable

> my\_data1$new\_hp1 <- my\_data1$hp \* 0.5

> colnames(my\_data1)

[1] "mpg" "cyl" "disp" "hp"

[5] "drat" "new\_hp1"

#naming variable

V1 V2 V3

101 MINA 45000

102 RINA 53000

103 SEEMA 28000

104 VINA NA

105 NIMA 34000

> data2 = read.csv(file="new.csv", col.names=c("Sno","NAME","SALARY"))

> data2

Sno NAME SALARY

1 101 MINA 45000

2 102 RINA 53000

3 103 SEEMA 28000

4 104 VINA NA

5 105 NIMA 34000

//dealing with missing values

UserID FULLNAME Age Score Grade

1 1 smit 25 85 B

2 2 john 30 92 A

3 3 bob 22 78 C

4 4 alice NA 65 C

5 5 charlie 28 89 B

> data$Age = ifelse(is.na(data$Age),ave(data$Age,FUN = function(x)mean(x,na.rm = TRUE)),data$Age)

> data

UserID FULLNAME Age Score Grade Gender

1 1 smit 25 85 B Male

2 2 john 30 92 A Female

3 3 bob 22 78 C Male

5 5 charlie 28 89 B Male

//dealing with categorical data

df$Country =factor(df$Country,levels=c(‘france’,’spain’,’germany’),labels=c(1.0,2.0,3.0))

//data reduction using subsetting

> data

V1 V2 V3

1 101 MINA 45000

2 102 RINA 53000

3 103 SEEMA 28000

4 104 VINA NA

5 105 NIMA 34000

//column selection

> subset\_data <- data[, c("V2", "V3")]

> subset\_data

V2 V3

1 MINA 45000

2 RINA 53000

3 SEEMA 28000

4 VINA NA

5 NIMA 34000

//using subset function

> subset\_data <- subset(data, V2 == "MINA")

> subset\_data

V1 V2 V3

1 101 MINA 45000

**PRACTICAL NO.7**

**AIM- Implementation and analysis of Linear regression through graphical methods.**

**Simple linear regression**

> ads<-read.csv("advertise.csv")

> view(ads)

> nrow(ads)

[1] 200

> ncol(ads)

[1] 4

> colnames(ads)

[1] "TV" "Radio" "Newspaper"

[4] "Sales"

> TV<-ads$TV

> Sales<-ads$Sales

> plot(TV,Sales)

> plot (TV, Sales, pch=16, cex=1, col='blue', main='TV vs Sales', xlab='TV', ylab='sales')

> model <- lm(Sales ~ TV)

> summary(model)

Call-

lm(formula = Sales ~ TV)

Residuals-

Min 1Q Median 3Q Max

-8.3860 -1.9545 -0.1913 2.0671 7.2124

Coefficients-

Estimate Std. Error t value

(Intercept) 7.032594 0.457843 15.36

TV 0.047537 0.002691 17.67

Pr(>|t|)

(Intercept) <2e-16 \*\*\*

TV <2e-16 \*\*\*

---

Signif. codes-

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’

0.1 ‘ ’ 1

Residual standard error- 3.259 on 198 degrees of freedom

Multiple R-squared- 0.6119, Adjusted R-squared- 0.6099

F-statistic- 312.1 on 1 and 198 DF, p-value- < 2.2e-16

> coefficients(model)

(Intercept) TV

7.03259355 0.04753664

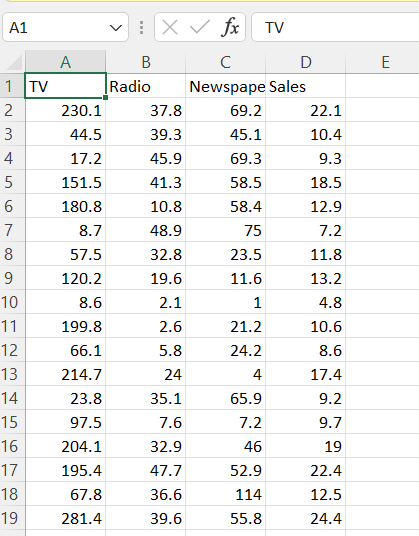
> coef(model)

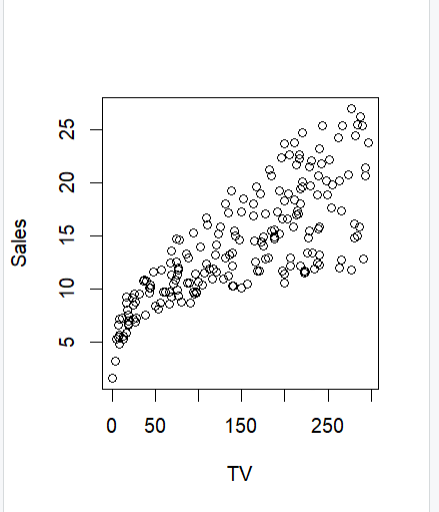
(Intercept) TV

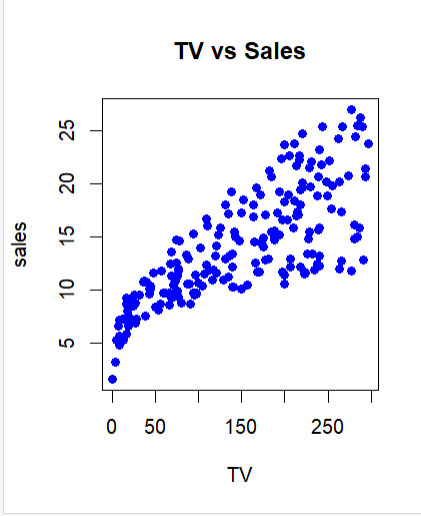
7.03259355 0.04753664

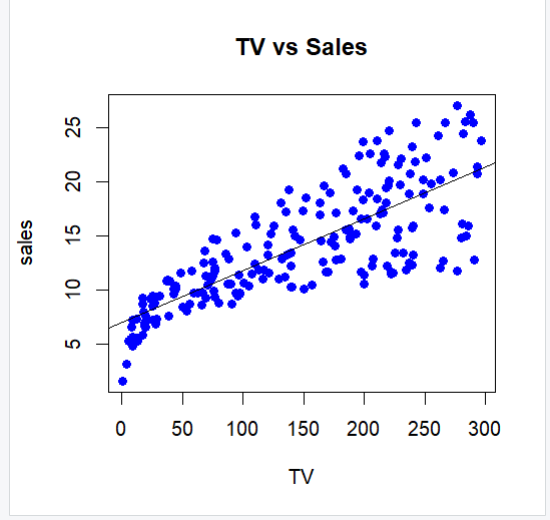
> abline(model)

Dataset= advertise.csv









**Multi linear regression**

> data\_in=read.csv("advertise.csv")

> plot(data\_in$Newspaper,data\_in$Radio)

> P1 = runif(200)

> P1

[1] 0.0718534596 0.0758784539 0.9222587277

[4] 0.7499774054 0.1503735969 0.2838246610

[7] 0.9785188290 0.8787262591 0.3438672579

[10] 0.3526551875 0.8047882700 0.3405104557

[13] 0.1412136271 0.5762494607 0.4737610647

[16] 0.2484424885 0.4147537174 0.0056585434

[19] 0.3936397876 0.9434619204 0.3435946400

[22] 0.5555662666 0.8041276808 0.4496011410

[25] 0.9906881901 0.0917285227 0.6126693289

[28] 0.9444318064 0.9826621644 0.7924660577

[31] 0.4344289922 0.1676183362 0.5687589620

[34] 0.0046076139 0.6681444086 0.6069936533

[37] 0.3835228358 0.6826846679 0.1123220637

[40] 0.7169814680 0.1272434620 0.7826690862

[43] 0.9523283544 0.3626450344 0.3412123281

[46] 0.0835960840 0.0093247723 0.1760357560

[49] 0.2458270874 0.7390013568 0.7118976754

[52] 0.9554161574 0.8945319587 0.4371296256

[55] 0.6708509389 0.9434681854 0.8498086254

[58] 0.0322956932 0.1206930012 0.3882001814

[61] 0.7699926670 0.1149553277 0.5990590823

[64] 0.8557556961 0.6993191498 0.8917943623

[67] 0.9299933126 0.1842395314 0.7463169664

[70] 0.0278463247 0.8933500338 0.5594351145

[73] 0.7746621536 0.6806199970 0.0622334110

[76] 0.0357349622 0.4041062221 0.4179408501

[79] 0.1341650309 0.5983146222 0.6875028438

[82] 0.3615968798 0.2985305712 0.4024941782

[85] 0.2175296345 0.3936405797 0.6535055810

[88] 0.3359900315 0.3702572128 0.6917497492

[91] 0.1941598633 0.9088638939 0.2890069799

[94] 0.5808701103 0.3605921306 0.7353006317

[97] 0.0315086176 0.7852203520 0.9099880380

[100] 0.1623231519 0.5748855583 0.0008972671

[103] 0.7080400428 0.3407407177 0.4848918049

[106] 0.8968255341 0.6527075702 0.3155160600

[109] 0.7452309884 0.0337528375 0.7196539391

[112] 0.0005203295 0.1405336151 0.5523303279

[115] 0.6868955598 0.6525590194 0.1423030191

[118] 0.1943323237 0.2457600238 0.0856786692

[121] 0.1751525914 0.6409379386 0.2107034749

[124] 0.4233466652 0.3144124495 0.4923673542

[127] 0.7714040992 0.1167143292 0.9367503780

[130] 0.4447615247 0.2563915108 0.8354442741

[133] 0.2247440561 0.1475371264 0.7120942909

[136] 0.5198035322 0.3623575440 0.4151975873

[139] 0.7389401216 0.7718217841 0.5535938707

[142] 0.1714756796 0.6958106435 0.2044118950

[145] 0.1759140829 0.2982776645 0.4577589822

[148] 0.1963324037 0.1760300738 0.0303095630

[151] 0.7955773568 0.8261671287 0.1418796696

[154] 0.4612462260 0.4423987814 0.6180045202

[157] 0.9111423036 0.6099810982 0.1385558334

[160] 0.4849685228 0.5949992624 0.8787579117

[163] 0.3118338524 0.6085016814 0.4645263716

[166] 0.7888774413 0.4370650381 0.9105486884

[169] 0.5499497552 0.9765654884 0.0071500591

[172] 0.5305178089 0.7410718543 0.8941168149

[175] 0.0079936041 0.6846924922 0.6269047991

[178] 0.8379436601 0.9467985439 0.2978515294

[181] 0.2898408629 0.6805284668 0.0684977174

[184] 0.3997992843 0.7015431772 0.3662630143

[187] 0.1039355290 0.6310499883 0.7278663479

[190] 0.9492182476 0.2688524560 0.6376093037

[193] 0.1780390081 0.8934126536 0.3818269379

[196] 0.6200961547 0.7518629848 0.5195188674

[199] 0.3443898580 0.7561842182

> p2 = order(P1)

> p2

[1] 112 102 34 18 171 175 47 70 150 97

[11] 58 110 76 75 183 1 2 46 120 26

[21] 187 39 62 128 59 41 79 159 113 13

[31] 153 117 134 5 100 32 142 121 145 149

[41] 48 193 68 91 118 148 144 123 85 133

[51] 119 49 16 131 191 6 93 181 180 146

[61] 83 163 125 108 88 12 104 45 21 9

[71] 199 10 95 82 137 44 186 89 195 37

[81] 60 19 86 184 84 77 17 138 78 124

[91] 31 167 54 155 130 24 147 154 165 15

[101] 105 160 126 198 136 172 169 114 141 22

[111] 72 33 101 14 94 161 80 63 36 164

[121] 158 27 156 196 177 188 192 122 116 107

[131] 87 35 55 182 74 38 176 115 81 90

[141] 143 65 185 103 51 135 40 111 189 96

[151] 139 50 173 109 69 4 197 200 61 127

[161] 140 73 42 98 166 30 151 23 11 152

[171] 132 178 57 64 8 162 66 71 194 174

[181] 53 106 92 99 168 157 3 67 129 20

[191] 56 28 179 190 43 52 170 7 29 25

> training\_ds = data\_in[p2[1-150],]

> test\_ds = data\_in[p2[150-200],]

> multiple\_line\_regression = lm(Newspaper ~ Radio + TV, data = training\_ds)

> abline(multiple\_line\_regression)

> summary(multiple\_line\_regression)

Call-

lm(formula = Newspaper ~ Radio + TV, data = training\_ds)

Residuals-

Min 1Q Median 3Q Max

-40.264 -15.355 -1.676 12.950 76.502

Coefficients-

Estimate Std. Error t value Pr(>|t|)

(Intercept) 14.62790 4.29932 3.402 0.000861

Radio 0.59229 0.11440 5.177 7.28e-07

TV 0.01759 0.02017 0.872 0.384687

(Intercept) \*\*\*

Radio \*\*\*

TV

---

Signif. codes-

0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error- 20.62 on 147 degrees of freedom

Multiple R-squared- 0.1606, Adjusted R-squared- 0.1492

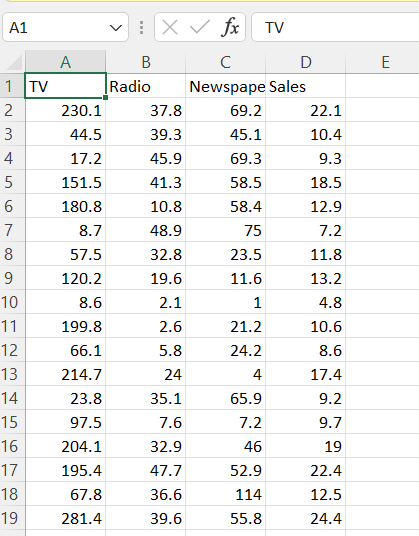
F-statistic- 14.06 on 2 and 147 DF, p-value- 2.578e-06

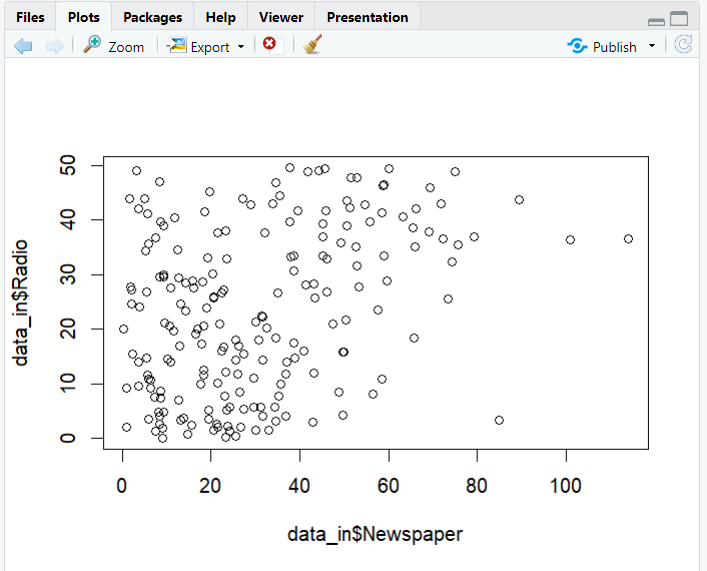
> plot(multiple\_line\_regression)

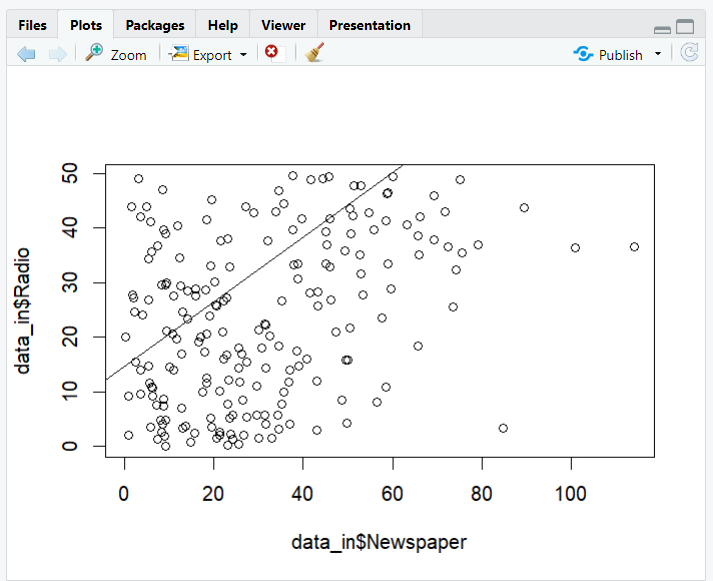
> pred\_value = predict(multiple\_line\_regression, newdata = test\_ds)

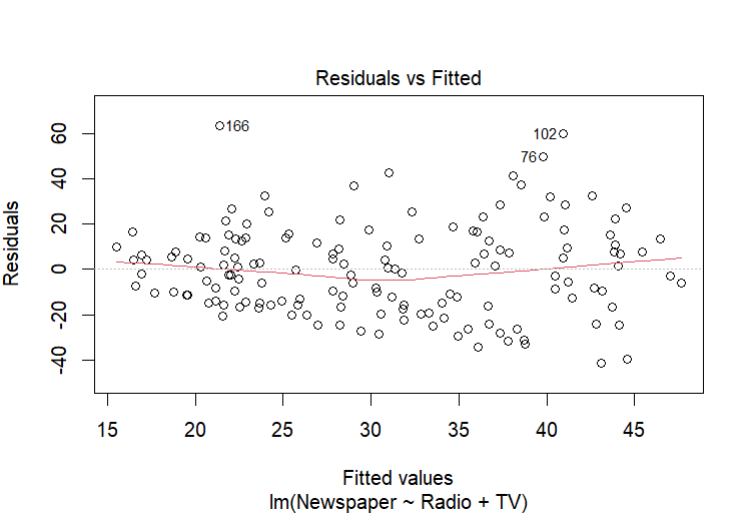
> test\_ds$Predicted\_values = pred\_value

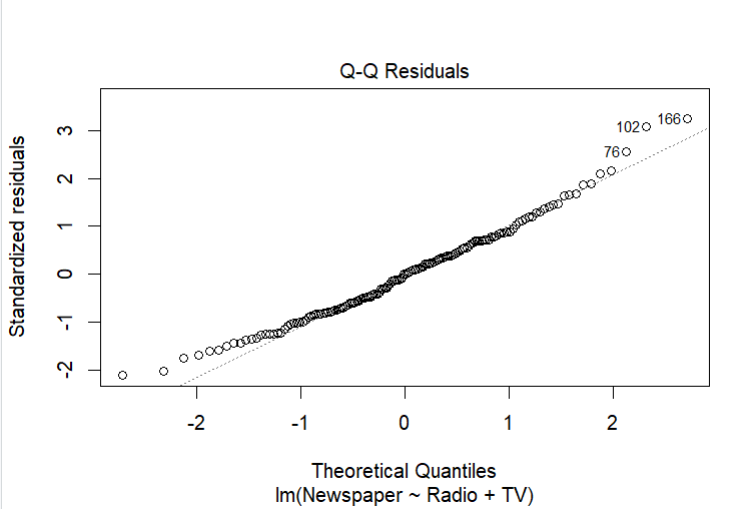
Dataset = advertise.csv

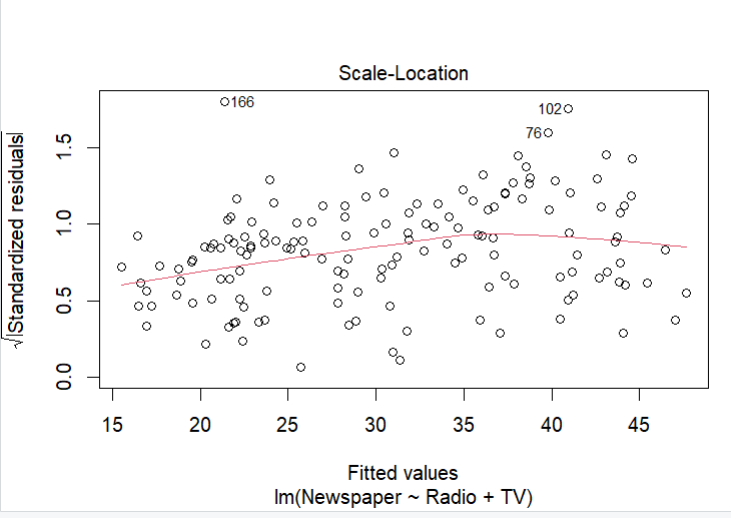






****

****

****

**PRACTICAL NO. 8**

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

**Naïve Bayesian**

install.packages("e1071")

install.packages("klaR")

install.packages("caret")

library(e1071)

library(klaR)

library(caret

> data("iris")

> 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

> unique(iris$Species)

[1] setosa versicolor virginica

Levels- setosa versicolor virginica

> pairs(iris[1-4], main = "Iris Data (red=setosa, green=versicolor, blue=virginica)",

+ pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])

>pairs(iris[1-4], main = "Iris Data (red=setosa, green=versicolor, blue=virginica)",

+ pch = 21, bg = c("red", "green3", "blue")[unclass(iris$Species)])

> index <- sample(nrow(iris), floor(nrow(iris) \* 0.7))

> train <- iris[index,]

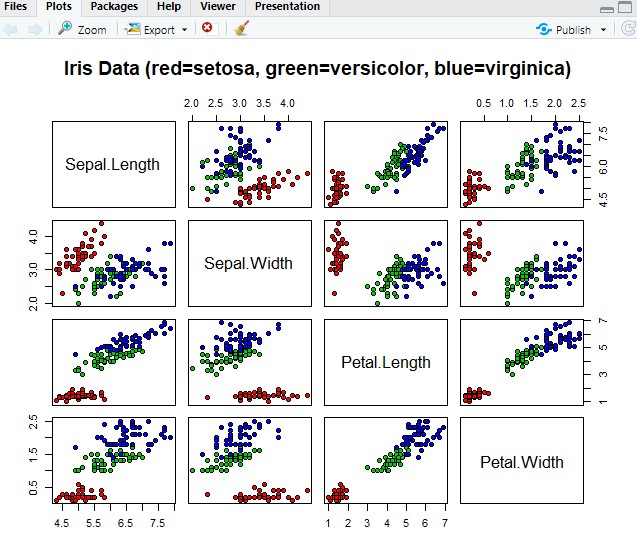
> test <- iris[-index,]

> xTrain <- train[, -5]

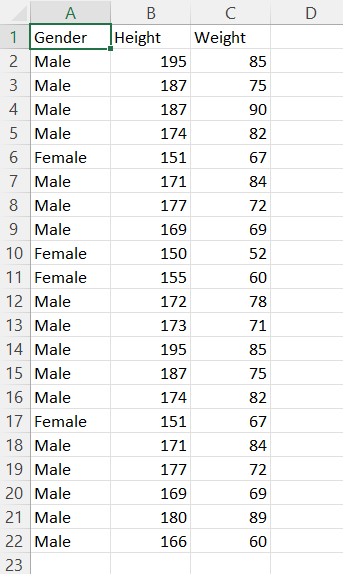
> yTrain <- train$Species

> xTest <- test[, -5]

> yTest <- test$Species



**Decision Tree**



> data\_in <- read.csv("new.csv")

> install.packages("rpart")

> install.packages("rpart.plot")

> library(rpart)

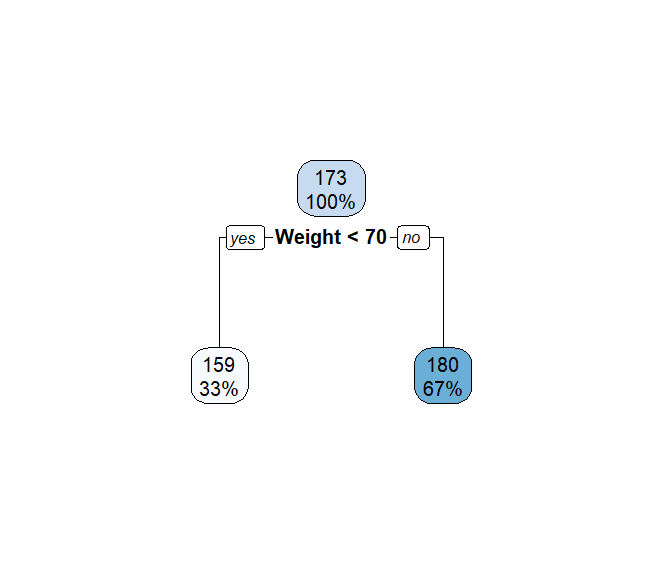
> library(rpart.plot)

> tree <- rpart(Height ~ Gender + Weight, data = data\_in)

> a <- data.frame(Gender = c("Male"), Weight = c(85))

> result <- predict(tree, a)

> rpart.plot(trees)



**K Nearest Neighbour**

> head(iris)

Sepal.Length Sepal.Width Petal.Length

1 5.1 3.5 1.4

2 4.9 3.0 1.4

3 4.7 3.2 1.3

4 4.6 3.1 1.5

5 5.0 3.6 1.4

6 5.4 3.9 1.7

Petal.Width Species

1 0.2 setosa

2 0.2 setosa

3 0.2 setosa

4 0.2 setosa

5 0.2 setosa

6 0.4 setosa

> library(class)

> library(ggplot2)

> set.seed(123)

> train\_indices <- sample(1-nrow(iris), 0.7 \* nrow(iris))

> train\_data <- iris[train\_indices, ]

> test\_data <- iris[-train\_indices, ]

> knn\_pred <- knn(train = train\_data[, -5], test = test\_data[, -5], cl = train\_data$Species, k = 3)

> knn\_accuracy <- sum(knn\_pred == test\_data$Species) / nrow(test\_data)

> cat("K-Nearest Neighbors Accuracy-", knn\_accuracy, "\n")

K-Nearest Neighbors Accuracy- 0.9777778

> knn\_table <- table(Actual = test\_data$Species, Predicted = knn\_pred)

> knn\_df <- as.data.frame.matrix(knn\_table)

> colnames(knn\_df) <- c('Actual', 'Predicted', 'Freq')

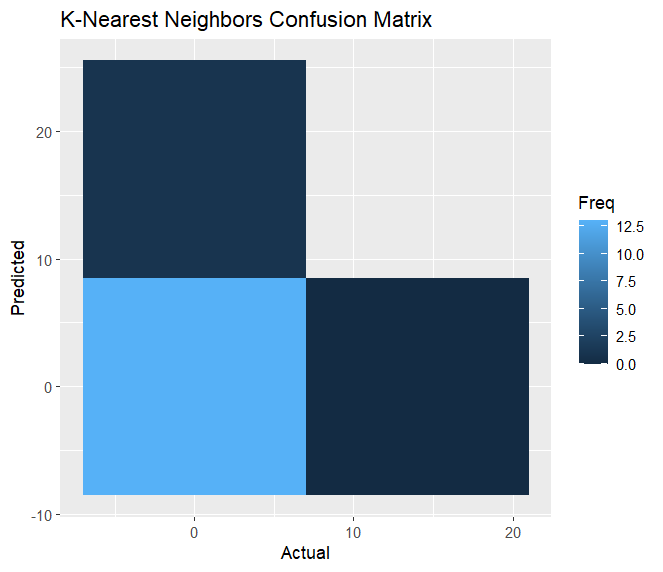
>ggplot(knn\_df, aes(x=Actual,y=predicted, fill=freq))+

+ geom\_tile()+

+ labs(title = “K-Nearest Neighbors Confusion Matrix”,

+ x=”Actual”,

+ y=”predicted”)



**ID3**

> library(rpart)

> library(rpart.plot)

> id3\_model <- rpart(Species ~ ., data = train\_data, method = "class")

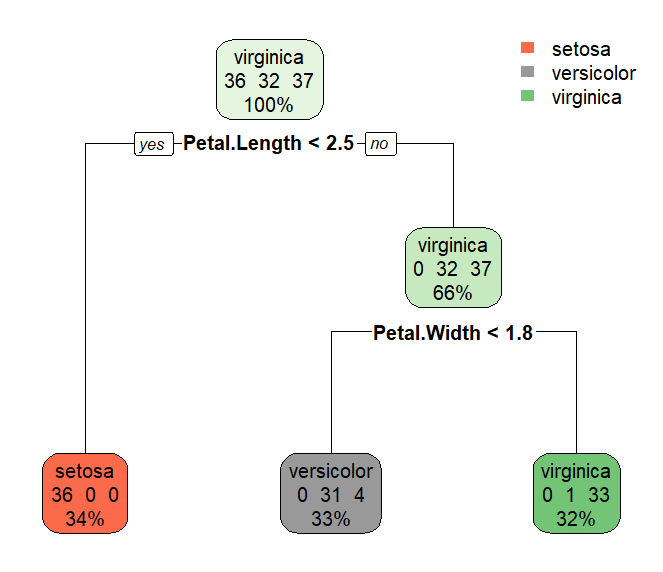
> id3\_pred <- predict(id3\_model, test\_data, type = "class")

> 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**

> library(C50)

> c45\_model <- C5.0(train\_data[, -5], train\_data$Species)

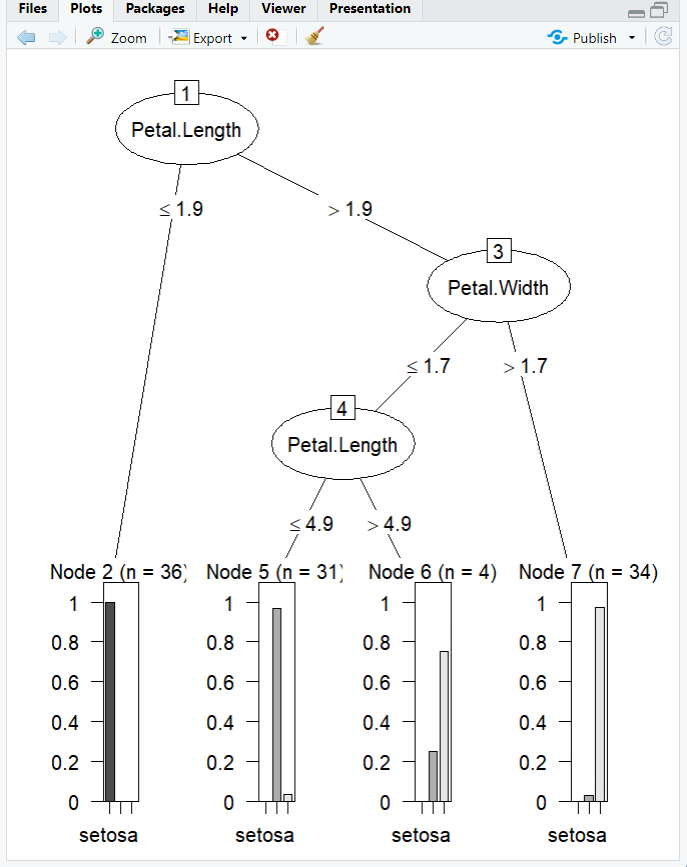
> c45\_pred <- predict(c45\_model, newdata = test\_data[, -5])

> c45\_accuracy <- sum(c45\_pred == test\_data$Species) / nrow(test\_data)

> cat("C4.5 Accuracy-", c45\_accuracy, "\n")

C4.5 Accuracy- 0.9777778

> plot(c45\_model)



**PRACTICAL NO. 9**

**AIM- Implementation and analysis of Apriori Algorithm using Market Basket Analysis.**

> install.packages("arules")

> library(arules)

> install.packages("arulesViz")

> library(arulesViz)

> install.packages("RColorBrewer")

> library(RColorBrewer)

> data(package="arules")

> data("Groceries")

> rules<-apriori(Groceries, parameter = list(supp=0.01,conf = 0.2))

Apriori

Parameter specification-

confidence minval smax arem aval

0.2 0.1 1 none FALSE

originalSupport maxtime support

TRUE 5 0.01

minlen maxlen target ext

1 10 rules TRUE

Algorithmic control-

filter tree heap memopt load sort

0.1 TRUE TRUE FALSE TRUE 2

verbose

TRUE

Absolute minimum support count- 98

set item appearances ...[0 item(s)] done [0.00s].

set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].

sorting and recoding items ... [88 item(s)] done [0.00s].

creating transaction tree ... done [0.00s].

checking subsets of size 1 2 3 4 done [0.00s].

writing ... [232 rule(s)] done [0.00s].

creating S4 object ... done [0.00s].

> inspect(rules[1.10])

lhs rhs support

[1] {} => {whole milk} 0.255516

confidence coverage lift count

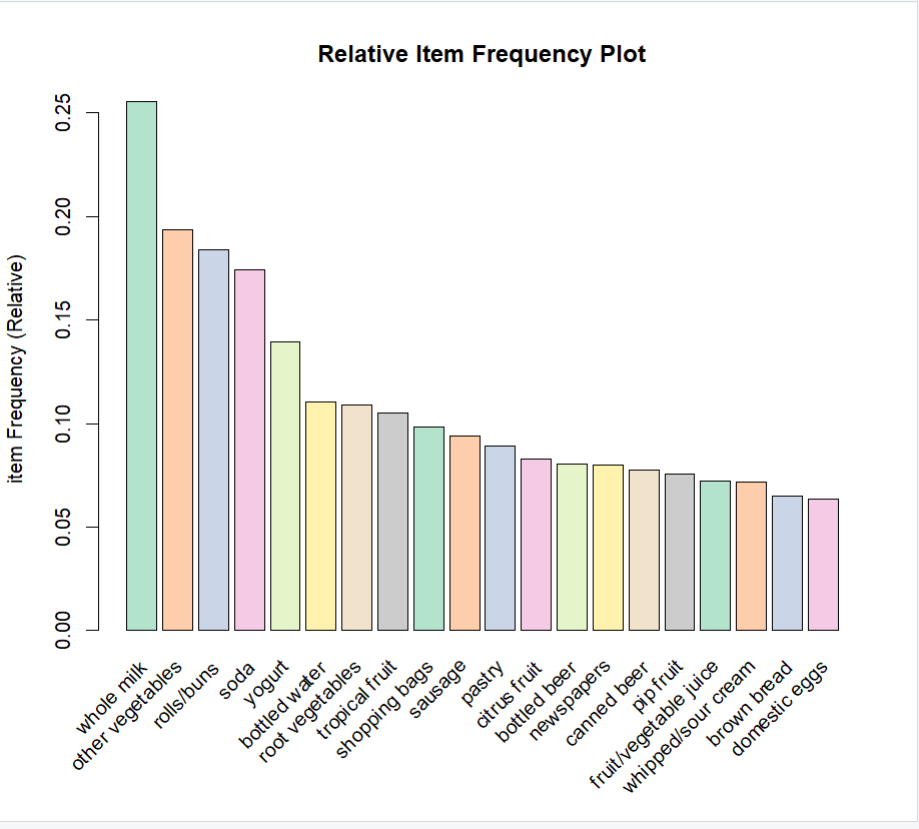
[1] 0.255516 1 1 2513

> arules--itemFrequencyPlot(Groceries,topN = 20,col=brewer.pal(8,'Kelaskarl2'),

+ main = 'Relative Item Frequency Plot',

+ type = "relative",

+ ylab = "item Frequency (Relative)")

****

**PRACTICAL NO. 10**

**AIM- Implementation and analysis of clustering algorithms like K-means, Agglomarative.**

**K means Clustering**

> install.packages("cluster")

> install.packages("factoextra")

> library(cluster)

> library(factoextra)

> data()

> df<-animals

> df<-na.omit(df)

> df<-scale(df)

> head(df)

war fly ver

ant -1.1832160 -0.5825753 -1.6020820

bee -1.1832160 1.6020820 -1.6020820

cat 0.7888106 -0.5825753 0.5825753

cpl -1.1832160 -0.5825753 -1.6020820

chi 0.7888106 -0.5825753 0.5825753

cow 0.7888106 -0.5825753 0.5825753

end gro hai

ant -0.6831301 0.6831301 -0.9036961

bee -0.6831301 0.6831301 1.0327956

cat -0.6831301 -1.3662601 1.0327956

cpl -0.6831301 -1.3662601 1.0327956

chi 1.3662601 0.6831301 1.0327956

cow -0.6831301 0.6831301 1.0327956

> head(df,n=10)

war fly ver

ant -1.1832160 -0.5825753 -1.6020820

bee -1.1832160 1.6020820 -1.6020820

cat 0.7888106 -0.5825753 0.5825753

cpl -1.1832160 -0.5825753 -1.6020820

chi 0.7888106 -0.5825753 0.5825753

cow 0.7888106 -0.5825753 0.5825753

duc 0.7888106 1.6020820 0.5825753

eag 0.7888106 1.6020820 0.5825753

ele 0.7888106 -0.5825753 0.5825753

fly -1.1832160 1.6020820 -1.6020820

end gro hai

ant -0.6831301 0.6831301 -0.9036961

bee -0.6831301 0.6831301 1.0327956

cat -0.6831301 -1.3662601 1.0327956

cpl -0.6831301 -1.3662601 1.0327956

chi 1.3662601 0.6831301 1.0327956

cow -0.6831301 0.6831301 1.0327956

duc -0.6831301 0.6831301 -0.9036961

eag 1.3662601 -1.3662601 -0.9036961

ele 1.3662601 0.6831301 -0.9036961

fly -0.6831301 -1.3662601 -0.9036961

> k2<-kmeans(df,centers = 2, nstart =25)

> str(k2)

List of 9

$ cluster - Named int [1-15] 2 2 1 2 1 1 1 1 1 2 ...

..- attr(\*, "names")= chr [1-15] "ant" "bee" "cat" "cpl" ...

$ centers - num [1-2, 1-6] 0.7888 -1.1832 -0.0971 0.1456 0.5826 ...

..- attr(\*, "dimnames")=List of 2

.. ..$ - chr [1-2] "1" "2"

.. ..$ - chr [1-6] "war" "fly" "ver" "end" ...

$ totss - num 84

$ withinss - num [1-2] 31.6 24

$ tot.withinss- num 55.7

$ betweenss - num 28.3

$ size - int [1-2] 9 6

$ iter - int 1

$ ifault - int 0

- attr(\*, "class")= chr "kmeans"

> k2

K-means clustering with 2 clusters of sizes 9, 6

Cluster means-

war fly ver

1 0.7888106 -0.09709588 0.5825753

2 -1.1832160 0.14564382 -0.8738629

end gro hai

1 0.4554200 0.227710 0.1721326

2 -0.6831301 -0.341565 -0.2581989

Clustering vector-

ant bee cat cpl chi cow duc eag ele fly

2 2 1 2 1 1 1 1 1 2

her liz man rab wha

2 2 1 1 1

Within cluster sum of squares by cluster-

[1] 31.62424 24.02727

(between\_SS / total\_SS = 33.7 %)

Available components-

[1] "cluster" "centers"

[3] "totss" "withinss"

[5] "tot.withinss" "betweenss"

[7] "size" "iter"

[9] "ifault"

> fviz\_cluster(k2,data=df)

> k3<-kmeans(df,center =3, nstart = 25)

> k4<-kmeans(df,center =4, nstart = 25)

> k5<-kmeans(df,center =5, nstart = 25)

> p1<-fviz\_cluster(k2, geom = "point", data = df) +ggtitle("k2")

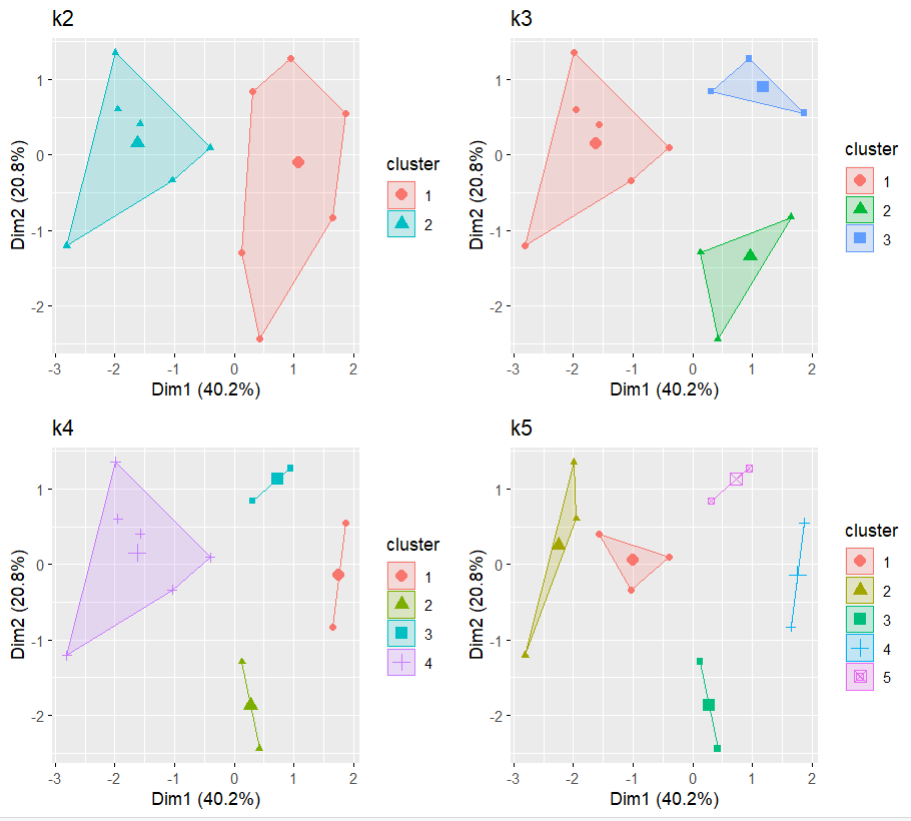
> p2<-fviz\_cluster(k3, geom = "point", data = df) +ggtitle("k3")

> p3<-fviz\_cluster(k4, geom = "point", data = df) +ggtitle("k4")

> p4<-fviz\_cluster(k5, geom = "point", data = df) +ggtitle("k5")

> library(gridExtra)

> grid.arrange(p1,p2,p3,p4, nrow=2)

****

**Agglomerative Hierarchical clustering**

**Complete Linkage**

> library(tidyverse)

> library(cluster)

> library(factoextra)

> df <- animals

> df <- na.omit(df)

> df <- scale(df)

> head(df)

war fly ver end gro

ant -1.1832160 -0.5825753 -1.6020820 -0.6831301 0.6831301

bee -1.1832160 1.6020820 -1.6020820 -0.6831301 0.6831301

cat 0.7888106 -0.5825753 0.5825753 -0.6831301 -1.3662601

cpl -1.1832160 -0.5825753 -1.6020820 -0.6831301 -1.3662601

chi 0.7888106 -0.5825753 0.5825753 1.3662601 0.6831301

cow 0.7888106 -0.5825753 0.5825753 -0.6831301 0.6831301

hai

ant -0.9036961

bee 1.0327956

cat 1.0327956

cpl 1.0327956

chi 1.0327956

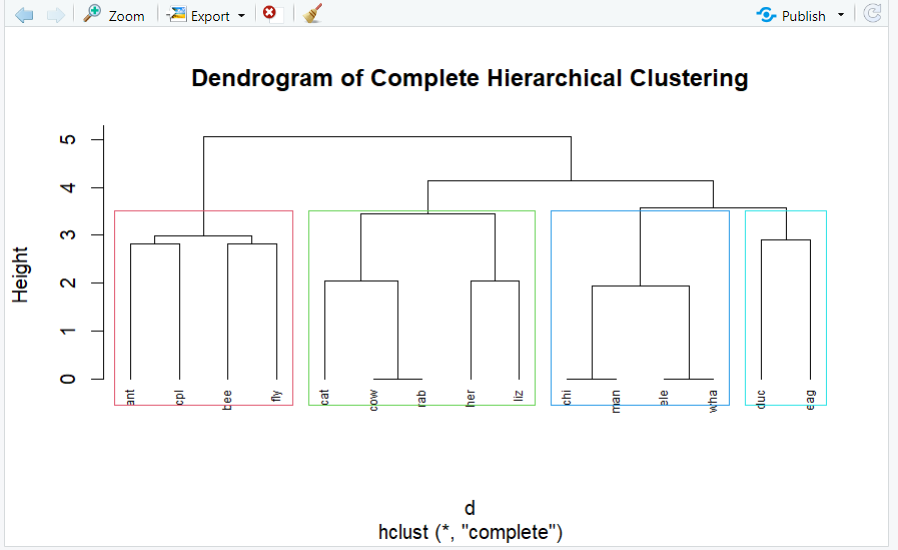
cow 1.0327956

> d <- dist(df, method = "euclidean")

> hc1 <- hclust(d, method = "complete")

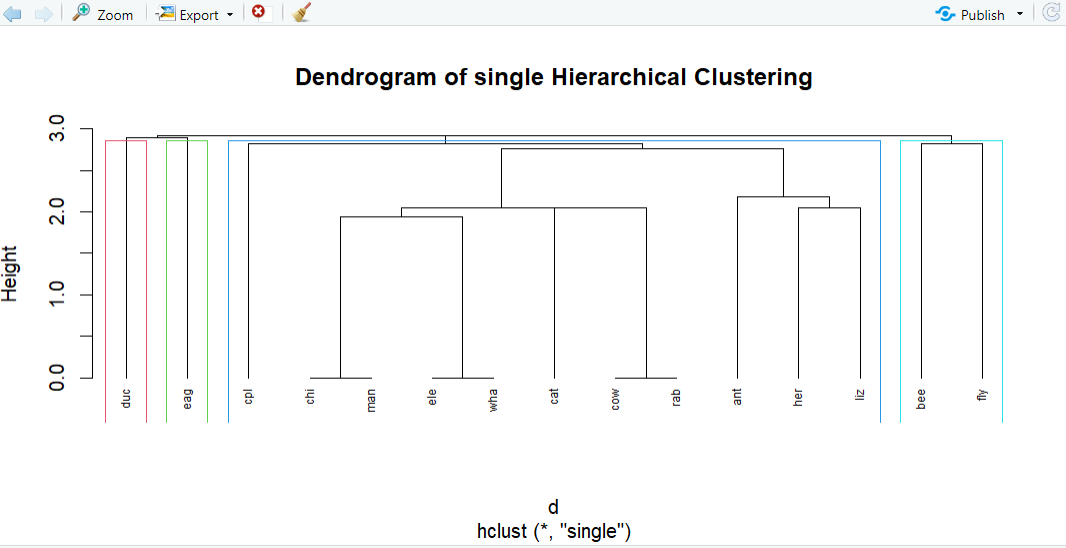
> plot(hc1, cex = 0.6, hang = -1, main = "Dendrogram of Complete Hierarchical Clustering")

> rect.hclust(hc1, k = 4, border = 2-5)

****

**Single Linkage**

|  |
| --- |
| > library(tidyverse)  > library(cluster)  > library(factoextra)  > data()  > df<-animals  > df<-na.omit(df)  > df<-scale(df)  > head(df)  war fly ver end gro  ant -1.1832160 -0.5825753 -1.6020820 -0.6831301 0.6831301  bee -1.1832160 1.6020820 -1.6020820 -0.6831301 0.6831301  cat 0.7888106 -0.5825753 0.5825753 -0.6831301 -1.3662601  cpl -1.1832160 -0.5825753 -1.6020820 -0.6831301 -1.3662601  chi 0.7888106 -0.5825753 0.5825753 1.3662601 0.6831301  cow 0.7888106 -0.5825753 0.5825753 -0.6831301 0.6831301  hai  ant -0.9036961  bee 1.0327956  cat 1.0327956  cpl 1.0327956  chi 1.0327956  cow 1.0327956  > d<- dist(df, method = "euclidean")  > hcl<-hclust(d,method = "single")  > plot(hcl, cex = 0.6, hang = -1, main="Dendrogram of singke Hierarchical Clustering")  > rect.hclust(hcl, k=4, border = 2-5)  > plot(hcl, cex = 0.6, hang = -1, main="Dendrogram of single Hierarchical Clustering")  > rect.hclust(hcl, k=4, border = 2-5) |

****

**Average Linkage**

> library(tidyverse)

> library(cluster)

> library(factoextra)

> df <- animals

> df <- na.omit(df)

> df <- scale(df)

> head(df)

war fly ver end gro

ant -1.1832160 -0.5825753 -1.6020820 -0.6831301 0.6831301

bee -1.1832160 1.6020820 -1.6020820 -0.6831301 0.6831301

cat 0.7888106 -0.5825753 0.5825753 -0.6831301 -1.3662601

cpl -1.1832160 -0.5825753 -1.6020820 -0.6831301 -1.3662601

chi 0.7888106 -0.5825753 0.5825753 1.3662601 0.6831301

cow 0.7888106 -0.5825753 0.5825753 -0.6831301 0.6831301

hai

ant -0.9036961

bee 1.0327956

cat 1.0327956

cpl 1.0327956

chi 1.0327956

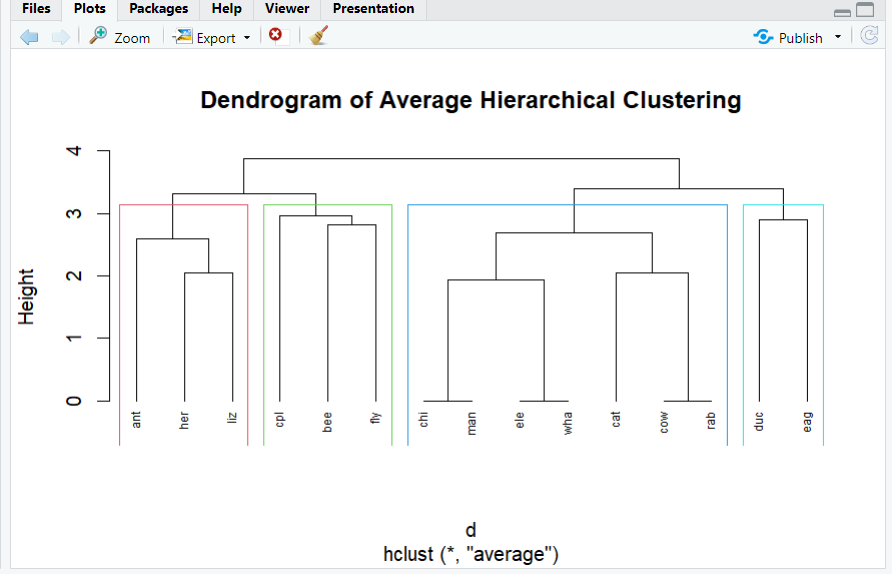
cow 1.0327956

> d <- dist(df, method = "euclidean")

> hcl <- hclust(d, method = "average")

> plot(hcl, cex = 0.6, hang = -1, main = "Dendrogram of Average Hierarchical Clustering")

> rect.hclust(hcl, k = 4, border = 2-5)

****

**Divisive clustering**

> library(tidyverse)

> library(cluster)

> library(factoextra)

> df <- animals

> df <- na.omit(df)

> df <- scale(df)

> head(df)

war fly ver end

ant -1.1832160 -0.5825753 -1.6020820 -0.6831301

bee -1.1832160 1.6020820 -1.6020820 -0.6831301

cat 0.7888106 -0.5825753 0.5825753 -0.6831301

cpl -1.1832160 -0.5825753 -1.6020820 -0.6831301

chi 0.7888106 -0.5825753 0.5825753 1.3662601

cow 0.7888106 -0.5825753 0.5825753 -0.6831301

gro hai

ant 0.6831301 -0.9036961

bee 0.6831301 1.0327956

cat -1.3662601 1.0327956

cpl -1.3662601 1.0327956

chi 0.6831301 1.0327956

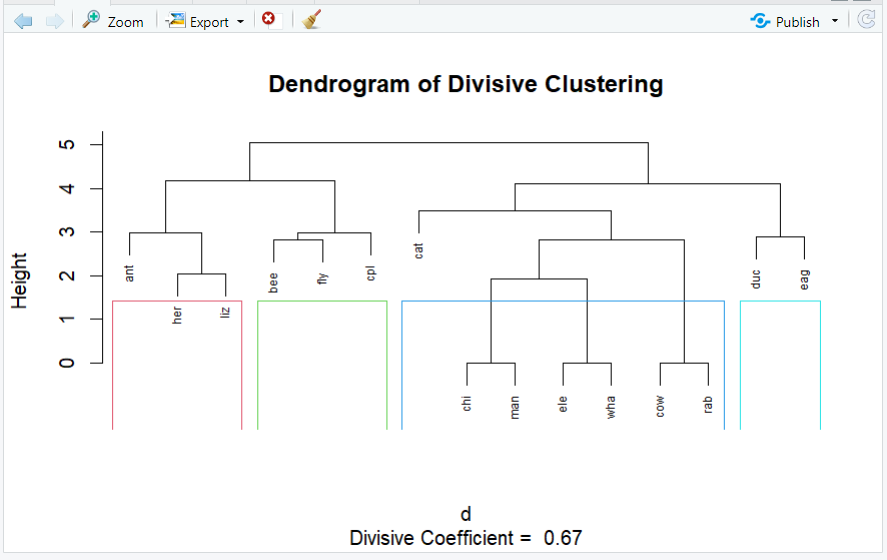
cow 0.6831301 1.0327956

> d <- dist(df, method = "euclidean")

> divisive\_clusters <- diana(d, diss = TRUE)

> plot(divisive\_clusters, which.plots = 2, cex = 0.6, main = "Dendrogram of Divisive Clustering")

> rect.hclust(divisive\_clusters, k = 4, border = 2-5)

****