Assignment No	**Revision Session**
Title	**revision**
Objective	1) **Hash and list partitioning** 2) **K-nearest neighbors** 3) **Number Range**
Roll No	MCA2565

1) (SQL – Hash and List Partitioning)

Create a table and implement Hash partitioning and List partitioning. Insert records and display the partition structure.

Source Code :-

1. Create Table student with attributes id,name,dob and partition by list

```
SQL> create table Student(
  2  student_id number(5),
  3  student_name varchar(20),
  4  student_dob date)
  5  partition by list(student_name)
  6  (partition stu_divA values('a','b','c','d','e','f','g','h','i','j','k'),
  7  partition stu_divB values('n','o','p','q','r','s','t','u','v','w','x','y','z'));

Table created.
```

2. Insert values into student tables

```
SQL> insert into Student values(0121,'m',to_date('23-AUG-1963','dd-MON-yyyy'));
insert into Student values(0121,'m',to_date('23-AUG-1963','dd-MON-yyyy'))
            *
ERROR at line 1:
ORA-14400: inserted partition key does not map to any partition


SQL> insert into Student values(0121,'n',to_date('23-AUG-1963','dd-MON-yyyy'));

1 row created.

SQL> insert into Student values(0221,'j',to_date('15-SEP-1963','dd-MON-yyyy'));

1 row created.

SQL> insert into Student values(0321,'a',to_date('21-JUN-1953','dd-MON-yyyy'));

1 row created.

SQL> insert into Student values(0421,'h',to_date('10-JUl-1991','dd-MON-yyyy'));

1 row created.

SQL> insert into Student values(0521,'i',to_date('06-DEC-1977','dd-MON-yyyy'));

1 row created.

SQL> insert into Student values(0621,'z',to_date('26-JAN-1993','dd-MON-yyyy'));

1 row created.

SQL> insert into Student values(0621,'s',to_date('18-FEB-1985','dd-MON-yyyy'));
```

3. Show the values in partition A and B

```
SQL> select * from Student partition(stu_divA);

STUDENT_ID STUDENT_NAME          STUDENT_D
---------- --------------------  ---------
       221 j                     15-SEP-63
       321 a                     21-JUN-53
       421 h                     10-JUL-91
       521 i                     06-DEC-77

SQL> select * from Student partition(stu_divB);

STUDENT_ID STUDENT_NAME          STUDENT_D
---------- --------------------  ---------
       121 n                     23-AUG-63
       621 z                     26-JAN-93
       621 s                     18-FEB-85
```

4. Show partition in table separatly.

```
SQL> SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS
  2  WHERE TABLE_NAME='STUDENT';

TABLE_NAME
------------------------------------------------------------------------------
PARTITION_NAME
------------------------------------------------------------------------------
HIGH_VALUE
------------------------------------------------------------------------------
  NUM_ROWS
----------
STUDENT
STU_DIVA
'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k'



TABLE_NAME
------------------------------------------------------------------------------
PARTITION_NAME
------------------------------------------------------------------------------
HIGH_VALUE
------------------------------------------------------------------------------
  NUM_ROWS
----------
STUDENT
STU_DIVB
'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'
```

5. Write a command to add new partition called stu_null for the null values

```
SQL> alter table student add partition stu_null values(NULL);

Table altered.

SQL> SELECT TABLE_NAME,PARTITION_NAME,HIGH_VALUE,NUM_ROWS FROM USER_TAB_PARTITIONS
  2  WHERE TABLE_NAME='STUDENT';

TABLE_NAME
------------------------------------------------------------------------------
PARTITION_NAME
------------------------------------------------------------------------------
HIGH_VALUE
------------------------------------------------------------------------------
  NUM_ROWS
----------
STUDENT
STU_DIVA
'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k'


TABLE_NAME
------------------------------------------------------------------------------
PARTITION_NAME
------------------------------------------------------------------------------
HIGH_VALUE
------------------------------------------------------------------------------
  NUM_ROWS
----------
STUDENT
STU_DIVB
'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'
```

```
TABLE_NAME
------------------------------------------------------------------------------
PARTITION_NAME
------------------------------------------------------------------------------
HIGH_VALUE
------------------------------------------------------------------------------
  NUM_ROWS
----------
STUDENT
STU_NULL
NULL
```

6. Add values into stu_null table

```
SQL> insert into Student values(0621,'',to_date('18-FEB-1985','dd-MON-yyyy'));

1 row created.

SQL> insert into Student values(0621,'',to_date('01-may-1999','dd-MON-yyyy'));

1 row created.
```

```
SQL> select * from Student partition(stu_null);

STUDENT_ID STUDENT_NAME          STUDENT_D
---------- --------------------- ---------
       621                       18-FEB-85
       621                       01-MAY-99
```

7. Write a command to add new partition called stu_default for the default values
   Write a command to display records from the stu_default partition

```
SQL> alter table Student add partition stu_default values(DEFAULT)
  2  ;

Table altered.

SQL> select * from Student partition(stu_default);

no rows selected
```

8. Command to add values ‘l’and ‘m’ in a partition stu_divA

```
SQL> alter table Student modify partition stu_divA add values ('l','m');

Table altered.
```

```
TABLE_NAME
------------------------------------------------------------------------------
PARTITION_NAME
------------------------------------------------------------------------------
HIGH_VALUE
------------------------------------------------------------------------------
  NUM_ROWS
----------
STUDENT
STU_DEFAULT
DEFAULT
```

9. Add new l and m values in div A

```
SQL> select * from Student partition(stu_divA);

STUDENT_ID STUDENT_NAME          STUDENT_D
---------- --------------------- ---------
       221 j                     15-SEP-63
       321 a                     21-JUN-53
       421 h                     10-JUL-91
       521 i                     06-DEC-77
       721 l                     01-JAN-00
       721 m                     02-DEC-86

6 rows selected.
```

2) (R – K-Nearest Neighbor)

Implement the K-Nearest Neighbor (KNN) classification algorithm in R and analyze the result.

Source Code :-

```
install.packages("class")
library(class)


table(iris$Species)
str(iris$Species)
head(iris)
ir=iris
train=ir[1:100,]
 ir1=ir[sample(nrow(ir)),]
head(ir1)
normalize<-function(x){
 return((x-min(x)/(max(x)-min(x))))
}
iris_n<-as.data.frame(lapply(ir1[,c(1,2,3,4)],normalize))
str(iris_n)
 iris_train<-iris_n[1:129,]
iris_test<-iris_n[130:150,]
iris_train_target<-iris[1:129,5]
iris_test_target<-iris[130:150,5]
iris_train_target
dim(iris_train)
dim(iris_test)
model<-knn(iris_train,iris_test,cl=iris_train_target,k=13)
model
```

table(iris_test_target,model)

Output :-

```
> table(iris$Species)

    setosa versicolor  virginica
        50         50         50
> str(iris$Species)
 Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> 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
> ir=iris
> train=ir[1:100,]
>   ir1=ir[sample(nrow(ir)),]
> head(ir1)
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
127          6.2         2.8          4.8         1.8  virginica
150          5.9         3.0          5.1         1.8  virginica
85           5.4         3.0          4.5         1.5 versicolor
109          6.7         2.5          5.8         1.8  virginica
21           5.4         3.4          1.7         0.2     setosa
87           6.7         3.1          4.7         1.5 versicolor
> normalize<-function(x){
+   return((x-min(x)/(max(x)-min(x))))
+ }
> iris_n<-as.data.frame(lapply(ir1[,c(1,2,3,4)],normalize))
> str(iris_n)
'data.frame':   150 obs. of  4 variables:
```

```
> str(iris_n)
'data.frame':   150 obs. of  4 variables:
 $ Sepal.Length: num  5.01 4.71 4.21 5.51 4.21 ...
 $ Sepal.Width : num  1.97 2.17 2.17 1.67 2.57 ...
 $ Petal.Length: num  4.63 4.93 4.33 5.63 1.53 ...
 $ Petal.Width : num  1.758 1.758 1.458 1.758 0.158 ...
>   iris_train<-iris_n[1:129,]
> iris_test<-iris_n[130:150,]
> iris_train_target<-iris[1:129,5]
> iris_test_target<-iris[130:150,5]
> iris_train_target
  [1] setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [11] setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [21] setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [31] setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [41] setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa     setosa
 [51] versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [61] versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [71] versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [81] versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
 [91] versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor versicolor
[101] virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica
[111] virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica
[121] virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica  virginica
Levels: setosa versicolor virginica
> dim(iris_train)
[1] 129   4
> dim(iris_test)
[1] 21  4
```

```
> model<-knn(iris_train,iris_test,cl=iris_train_target,k=13)
> model
 [1] setosa     setosa     setosa     versicolor virginica  setosa     virginica  setosa     setosa     setosa
[11] setosa     virginica  setosa     setosa     setosa     setosa     versicolor setosa     versicolor versicolor
[21] versicolor
Levels: setosa versicolor virginica
> table(iris_test_target,model)
                model
iris_test_target setosa versicolor virginica
      setosa          0          0         0
      versicolor      0          0         0
      virginica      13          5         3
>
>
```

3) (Pentaho – Number Range)

Design a Pentaho transformation using the Number Range step to categorize numerical values into ranges.

Source Code :-

Number range

Step name: Number range

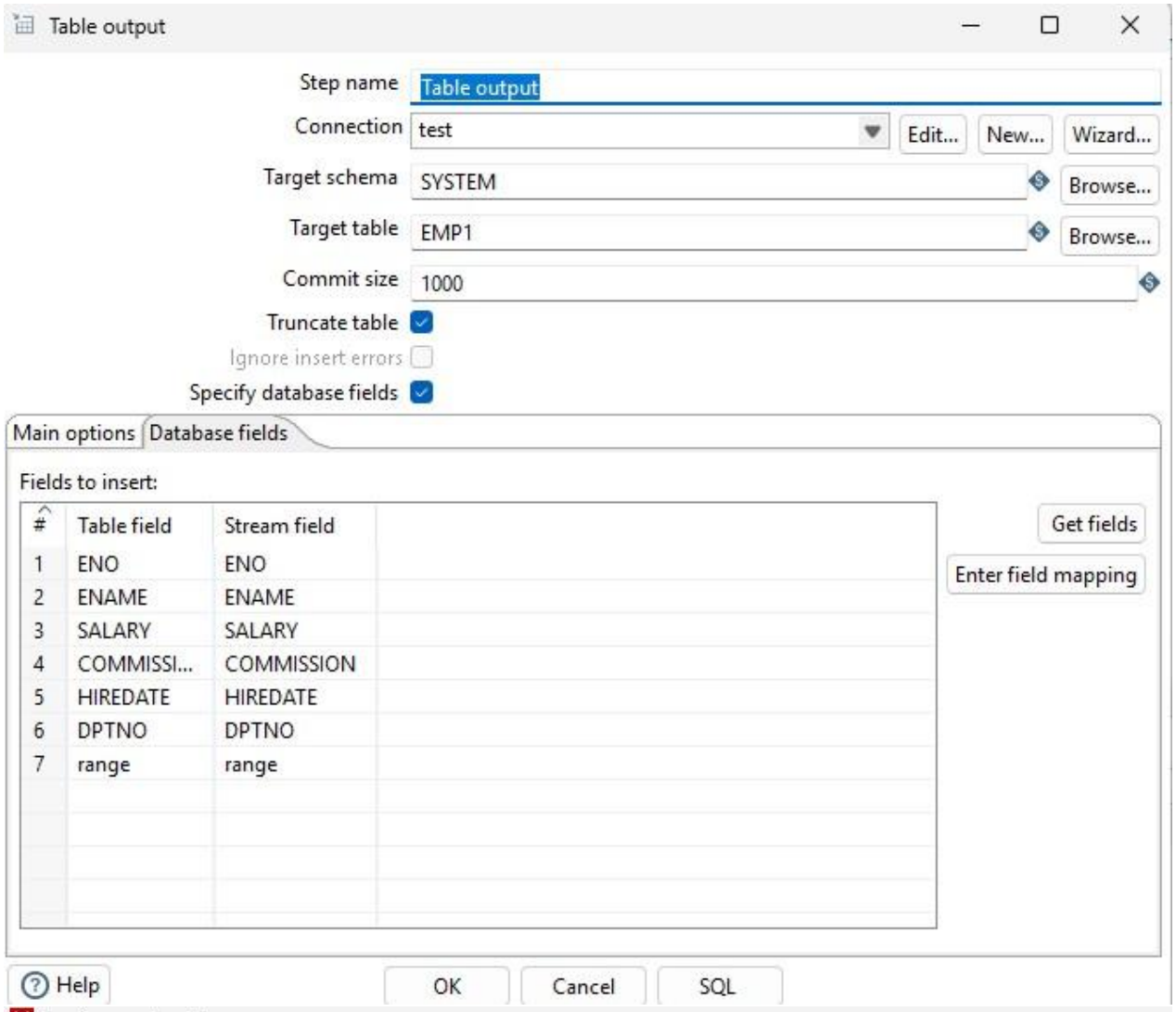
Input field: ENO

Output field: range

efault value(if no range matches): unknown

anges (min <= x< max):

#	Lower Bound	Upper Bound	Value
1		5.0	Less than 5
2	5.0	10.0	5-10
3	10.0		More than 10



#	Table field	Stream field
1	ENO	ENO
2	ENAME	ENAME
3	SALARY	SALARY
4	COMMISSI...	COMMISSION
5	HIREDATE	HIREDATE
6	DPTNO	DPTNO
7	range	range

Examine preview data

Rows of step: Table output (10 rows)

#	ENO	ENAME	SALARY	COMMISSION	HIREDATE	DPTNO	range
1	10.0	**********Xavier	39000.0	10000	1990/07/30 00:00:00.000000000	9.0	More than 10
2	9.0	**********Max	35000.0	9000	1986/05/05 00:00:00.000000000	10.0	5-10
3	8.0	**********Sameer	20000.0	1000	1996/10/18 00:00:00.000000000	2.0	5-10
4	7.0	**********Yadhnesh	50000.0	<null>	2001/04/12 00:00:00.000000000	6.0	5-10
5	6.0	**********Banner	12000.0	<null>	2012/03/13 00:00:00.000000000	1.0	5-10
6	5.0	**********Mary	18000.0	3000	1999/11/01 00:00:00.000000000	5.0	5-10
7	4.0	**********Harry	15000.0	3000	2005/11/12 00:00:00.000000000	6.0	Less than 5
8	3.0	**********Lex	30000.0	8000	1985/02/25 00:00:00.000000000	12.0	Less than 5
9	2.0	**********Alex	25000.0	5000	1996/01/12 00:00:00.000000000	5.0	Less than 5
10	1.0	**********Atharva	80000.0	50000	2000/12/02 00:00:00.000000000	5.0	Less than 5