Q.1) A) APRIORI ALGORITHM WITH SUPPORT 70% AND CONFIDENCE 80%.

1. LOAD GROCERIES DATASET AS TRANSACTIONS.

```
> setwd("C:\\Users\\user\\Desktop\\23")
> getwd()
[1] "C:/Users/user/Desktop/23"
> mba_data<-read.csv("C:\\Users\\user\\Desktop\\23\\groceries.csv")
> trans <- split(mba_data$Products, mba_data$Customer_Id,"transactions")
> head(trans)
$`1`
[1] "bread"  "butter" "eggs"   "milk"

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

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

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

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

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

> library(arules)
Loading required package: Matrix

Attaching package: 'arules'

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

    abbreviate, write
```

2. APPLY APRIORI ALGORITHM :

```
> rules = apriori(trans, parameter=list(support=0.7, confidence=0.8))
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen
        0.8    0.1    1 none FALSE            TRUE       5     0.7      1     10
 target  ext
  rules TRUE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 10

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[15 item(s), 15 transaction(s)] done [0.00s].
sorting and recoding items ... [1 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 done [0.00s].
writing ... [1 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
Warning message:
In asMethod(object) : removing duplicated items in transactions
> inspect(rules)
     lhs     rhs        support confidence coverage lift count
[1] {}  => {bread} 0.8     0.8          1        1    12
```
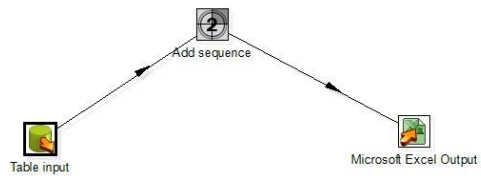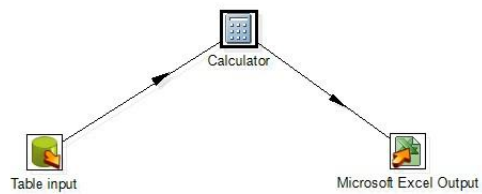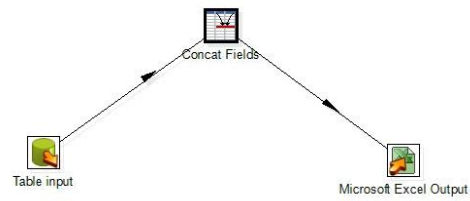
Q.2) A) PENTAHO :

Q.2) B)

Create table sales(sid,ssname,salesregion) with 3 partition using list partition based on region

```
SQL> CREATE TABLE sales(sid INTEGER,ssname VARCHAR(20),salesregion VARCHAR(20))
  2  PARTITION BY LIST(salesregion)
  3  (PARTITION p1 VALUES('MAHARASHTRA'),
  4  PARTITION p2 VALUES('GOA'),
  5  PARTITION p3 VALUES('KERALA'));
```

a) Describe the structure of the table and display the partition table.

```
SQL> DESC sales;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 SID                                                NUMBER(38)
 SSNAME                                             VARCHAR2(20)
 SALESREGION                                        VARCHAR2(20)
```

b)Insert minimum 6 records and display the content from each partition.

```
SQL> INSERT INTO sales VALUES(1,'Mango','MAHARASHTRA');

1 row created.

SQL> INSERT INTO sales VALUES(2,'Orange','GOA');

1 row created.

SQL> INSERT INTO sales VALUES(3,'Apple','GOA');

1 row created.

SQL> INSERT INTO sales VALUES(4,'Guava','MAHARASHTRA');

1 row created.

SQL> INSERT INTO sales VALUES(5,'Banana','MAHARASHTRA');

1 row created.

SQL> INSERT INTO sales VALUES(6,'Strawberry','KERALA');

1 row created.




SQL> SELECT * FROM sales;

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         1 Mango                MAHARASHTRA
         4 Guava                MAHARASHTRA
         5 Banana               MAHARASHTRA
         2 Orange               GOA
         3 Apple                GOA
         6 Strawberry           KERALA

6 rows selected.

SQL> SELECT * FROM sales PARTITION(p1);

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         1 Mango                MAHARASHTRA
         4 Guava                MAHARASHTRA
         5 Banana               MAHARASHTRA

SQL> SELECT * FROM sales PARTITION(p2);

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         2 Orange               GOA
         3 Apple                GOA

SQL> SELECT * FROM sales PARTITION(p3);

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         6 Strawberry           KERALA
```

c)Add new partition and insert 2 records and display content from each partiton.

```
SQL> ALTER TABLE sales ADD PARTITION p4 VALUES('GUJARAT');

Table altered.

SQL> INSERT INTO sales VALUES(7,'Kiwi','GUJARAT');

1 row created.

SQL> INSERT INTO sales VALUES(8,'Grapes','GUJARAT');

1 row created.

SQL> SELECT * FROM sales;

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         1 Mango                MAHARASHTRA
         4 Guava                MAHARASHTRA
         5 Banana               MAHARASHTRA
         2 Orange               GOA
         3 Apple                GOA
         6 Strawberry           KERALA
         7 Kiwi                 GUJARAT
         8 Grapes               GUJARAT


SQL> SELECT * FROM sales PARTITION(p1);

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         1 Mango                MAHARASHTRA
         4 Guava                MAHARASHTRA
         5 Banana               MAHARASHTRA

SQL> SELECT * FROM sales PARTITION(p2);

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         2 Orange               GOA
         3 Apple                GOA

SQL> SELECT * FROM sales PARTITION(p3);

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         6 Strawberry           KERALA

SQL> SELECT * FROM sales PARTITION(p4);

       SID SSNAME               SALESREGION
---------- -------------------- --------------------
         7 Kiwi                 GUJARAT
         8 Grapes               GUJARAT
```

Q.3) A) Create an ADT passenger_type with fields PID,PName,Address,Destination,Age. Create an object table passenger_dtls of type passenger.Insert some meaningful record and display.

SQL> create type passenger_type as object

  2 (

  3   PID number(6),

```
  4   PName varchar2(20),

  5   Address varchar2(20),

  6   Destination varchar2(20),

  7   Age number(6)

  8   );

  9   /
```
Type created.


```
SQL> create table passenger

 2   (

 3   Passenger_dtls passenger_type

 4   );
```
Table created.

SQL> insert into passenger values(passenger_type('1','Aman','Ghatkopar','London','22'));
1 row created.

SQL> insert into passenger values(passenger_type('2','Aakash','Thane','Sweden','20'));
1 row created.

SQL> insert into passenger values(passenger_type('3','Deepak','Chembur','Germany','19'));
1 row created.

SQL> select * from passenger;
PASSENGER_DTLS(PID, PNAME, ADDRESS, DESTINATION, AGE)

PASSENGER_TYPE(1, 'Aman', 'Ghatkopar', 'London', 22)
PASSENGER_TYPE(2, 'Aakash', 'Thane', 'Sweden', 20)
PASSENGER_TYPE(3, 'Deepak', 'Chembur', 'Germany', 19)

Q.3) B) Show the usage of First and Last Analytical Functions in oracle.
```
SQL> create table studs(

 2   id int,

 3   name varchar(20),

 8   );
```

```
  4  department varchar(20),

  5  major varchar(20),

  6  joining_date date,

  7  marks int );
Table created.
SQL> insert into studs values(1,'aman','science','IT',to_date('3/2/2022','dd/mm/yyyy'),95);

1 row created.

SQL> insert into studs

values(2,'prasad','science','physics',to_date('4/2/2022','dd/mm/yyyy'),92);

1 row created.

SQL> select * from studs;

      ID NAME            DEPARTMENT        MAJOR

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

JOINING_D     MARKS

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

      1 aman            science        IT

03-FEB-22      95


      2 prasad          science        physics

04-FEB-22      92
SQL> select department,marks,

  2  max(marks)keep(DENSE_RANK FIRST ORDER BY marks desc)

  3  over(PARTITION BY department)"max"

  4  from studs;

DEPARTMENT            MARKS     max

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

science               92     95

science               95     95


SQL> select department,marks,

  2  min(marks)keep(DENSE_RANK LAST ORDER BY marks desc)

  3  over(PARTITION BY department)"min"

  4  from studs;

DEPARTMENT            MARKS     min
```

```
-------------------- ---------- ----------
science              92         92
science              95         92
```