##

Please maximize the window to see the indentation and comments correctly.

Metadata of the dataset:
1. event_epoch_time
2. user_id
3. device_id
4. user_agent
5. pizza_name
6. isCheeseBurst
7. Size
8. AddedToppings(colon separated string)
9. Price
10. Coupon Code
11. Order_Event
12. isVeg

##
##
Answer 1:
This algorithm removes all such records with event_epoch_time, user_id, device_id, user_agent as NULL and returns only those records with proper valid information in data fields.

#Ignore key here , value will be the entire record.
Map(key,value)

1. LS = split(value, '\t') //Here,'\t' is to denote tab
2. if((LS[1] != NULL) AND (LS[2] != NULL) AND (LS[3] != NULL) AND (LS[4] != NULL)) //here each value in first 4 columns is compared against Null, if any of these
//equals NULL , the record is not returned.
3. write(key, value)

##
##
Answer 2:
This reads the user agent field value and returns OS Version and Platform.

#Ignore key here , value will be the entire record.
Map(key,value)

1. LS = spilt(value, '\t')
2. if(LS[4] ! = NULL) //First check will be if user agent field has a non-null value
3. userAgent = split(LS[4] , ':'); //here, split is done on colon(:)
4. OSVersion = userAgent[1]; //First value in array obtained is OS Version
5. Platform = userAgent[2]; //Second value in array obtained is Platform
6. write(OSVersion , Platform);

##
##
Answer 3:

****
Task1: Find out the number of veg and non-veg pizzas sold

#Ignore key here , value will be the entire record.
Map(key,value)

1. vegPizza = 0 , nonvegPizza = 0;
2. LS = split(value, '\t')
3. if(LS[12] != NULL) //First check will be if isVeg field has a non-null value
4. if(LS[12] == 'Y') //Since this field has only 2 values - Y or N , if Y -> increment veg pizza count otherwise increment nonveg pizza count
5. vegPizza = getCounter("vegPizza").incrementBy(1);
6. else nonvegPizza = getCounter("nonvegPizza").incrementBy(1);
7. write("Veg :" , vegPizza)
8. write("Nonveg :" , nonvegPizza)

****
Task2: Find out the size wise distribution of pizzas sold

#Ignore key here , value will be the entire record.
Map(key, value)

1. medium = 0 , regular = 0 , large = 0;
2. LS = split(value, '\t')
3. if(LS[7] != NULL) //First check will be if Size field has a non-null value
4. if(LS[7] == 'M') //Check if size is M
5. medium = getCounter("medium").incrementBy(1); //increment medium size sold counter
6. else if (LS[7] == 'R') //If size is R, increment regular size sold counter
7. regular = getCounter("regular").incrementBy(1);
8. else large = getCounter("large").incrementBy(1); //Since this field has only 3 values - M, R , L. If is not M /R , it will be L.
9. write("Medium Pizzas :" , medium);
10.write("Regular Pizzas :" , regular);
11.write("Large Pizzas :" , large);

****
Task3: Find out how many cheese burst pizzas were sold

#Ignore key here , value will be the entire record.
Map(key,value)

1. cheeseBurst = 0;
2. LS = split(value, '\t')
3. if(LS[6] != NULL) //First check will be if isCheeseBurst field has a non-null value
4. if(LS[6] == 'Y') //This field has only 2 values - Y , N. If Y , increment count of cheeseBurst pizza, do not do anything with value 'N'
5. cheeseBurst = getCounter("cheeseBurst").incrementBy(1);
6. write("CheeseBurst Pizzas :" , cheeseBurst);

****
Task4:Find out how many small cheese burst pizzas were sold

#Ignore key here , value will be the entire record.
Map(key,value)

1. smallCheeseBurst = 0;
2. LS = split(value, '\t')
3. if(LS[6] != NULL) //First check will be if isCheeseBurst field has a non-null value
4. if(LS[6] == 'Y' AND LS[7] =='R') //If Yes , then check if the size is Regular(Small), if both are true, increment counter.
5. smallCheeseBurst = getCounter("smallCheeseBurst").incrementBy(1);
6. write("Small CheeseBurst Pizzas :" smallCheeseBurst);
7. if(smallCheeseBurst == 0) write("Correct") //Ideally, the count should be 0. If not , something is wrong with data or with algorithm.
8. else write("Something went wrong !!")

****
Task5: Find out the number of cheese burst pizzas whose cost is below Rs 500

#Ignore key here , value will be the entire record.
Map(key,value)

1. cheeseBurstBelow500 = 0;
2. LS = split(value, '\t')
3. if(LS[6] != NULL) //First check will be if isCheeseBurst field has a non-null value
4. if(LS[6] == 'Y' AND LS[9] < 500) //If yes, check the price. If price is less than 500, then only increment, otherwise don't.
5. cheeseBurstBelow500 = getCounter("cheeseBurstBelow500").incrementBy(1);
6. write("CheeseBurst Pizzas Below 500 :" , cheeseBurstBelow500);

##########
Answer 4:
updated algorithms for the tasks in point-3 without using getCounter method.

****
Task1: Find out the number of veg and non-veg pizzas sold

#Ignore key here , value will be the entire record.
Map(key,value)

1. vegPizza = 0 , nonvegPizza = 0;
2. LS = split(value, '\t')
3. if(LS[12] != NULL) //First check will be if isVeg field has a non-null value
4. if(LS[12] == 'Y') //Since this field has only 2 values - Y or N , if Y -> increment veg pizza count otherwise increment nonveg pizza count

```
5.              vegPizza = vegPizza + 1;
6.      else  nonvegPizza = nonvegPizza + 1;
7. write("Veg :"  , vegPizza)
8. write("Nonveg :" , nonvegPizza)
```

****

Task2: Find out the size wise distribution of pizzas sold

#Ignore key here , value will be the entire record.
Map(key, value)

```
1. medium = 0 , regular = 0 , large = 0;
2. LS = split(value, '\t')
3.   if(LS[7] != NULL)                    //First check will be if Size field has a non-null value
4.      if(LS[7] == 'M')                  //Check if size is M
5.              medium = medium + 1;              //increment medium size sold count
6.      else if (LS[7] == 'R')                  //If size is R, increment regular size sold count
7.              regular = regular + 1;
8.      else   large = large + 1;                //Since this field has only 3 values - M, R , L. If is not M /R , it will be L.
9. write("Medium Pizzas :" , medium );
10.write("Regular Pizzas :" , regular);
11.write("Large Pizzas :" , large );
```

****

Task3: Find out how many cheese burst pizzas were sold

#Ignore key here , value will be the entire record.
Map(key,value)

```
1. cheeseBurst = 0;
2. LS = split(value, '\t')
3.    if(LS[6] != NULL)                    //First check will be if isCheeseBurst field has a non-null value
4.        if(LS[6] == 'Y')                 //This field has only 2 values - Y , N. If Y , increment count of cheeseBurst
pizza, do not do anything with value 'N'
5.            cheeseBurst = cheeseBurst + 1 ;
6. write("CheeseBurst Pizzas :" , cheeseBurst);
```

****

Task4:Find out how many small cheese burst pizzas were sold

#Ignore key here , value will be the entire record.
Map(key,value)

```
1. smallCheeseBurst = 0;
2. LS = split(value, '\t')
3.    if(LS[6] != NULL)                    //First check will be if isCheeseBurst field has a non-null value
4.        if(LS[6] == 'Y' AND LS[7] =='R')              //If Yes , then check if the size is Regular(Small), if both are
true, increment count.
5.            smallCheeseBurst = smallCheeseBurst + 1;
```

6. write("Small CheeseBurst Pizzas :" smallCheeseBurst);
7. if(smallCheeseBurst == 0) write("Correct") //Ideally, the count should be 0. If not, something is wrong with data or with algorithm.
8. else write("Something went wrong !!")

****

Task5: Find out the number of cheese burst pizzas whose cost is below Rs 500

#Ignore key here , value will be the entire record.
Map(key,value)

1. cheeseBurstBelow500 = 0;
2. LS = split(value, '\t')
3. if(LS[6] != NULL) //First check will be if isCheeseBurst field has a non-null value
4. if(LS[6] == 'Y' AND LS[9] < 500) //If yes, check the price. If price is less than 500, then only increment, otherwise don't.
5. cheeseBurstBelow500 = cheeseBurstBelow500 + 1;
6. write("CheeseBurst Pizzas Below 500 :" , cheeseBurstBelow500);

##########