Write a program to create a dictionary with product id as the key and product details as values for inventory management system ¶

- **Detail Structure of Product in Dictionary**

```
{ key --> product_id : values --> [ product_name, category, price, quantity, discount ] }
```

Importing Libraries

In [1]:

```python
import json as js
import numpy as np
import pandas as pd
import os
from IPython.display import clear_output
```

Read and Write function for json Files

In [2]:

```python
# reads json file and return dictionary

def read_file(products=True):

    fr = None

    # Opening records.json file in read mode if product is True
    if products: fr = open('records.json','r')

    # Opening transaction.json file in read mode if product is False
    else: fr = open('transaction.json','r')

    if fr == None: print('Error')

    # Loading json file into dictionary
    r = js.loads(fr.read())

    # Closing the file
    fr.close()

    # Return dictionary
    return r
```

In [3]:

```
#Write json file for given dictionaty input

def write_file(var,products=True):

    fw = None

    # Opening records.json file in write mode if product is True
    if products: fw = open('records.json','w')

    # Opening transaction.json file in write mode if product is False
    else: fw = open('transaction.json','w')

    # Dumping dictionary into json file
    fw.write(js.dumps(var))

    # Closing the file
    fw.close()
```

Product view of IMS

In [4]:

```
def product_views():

    # Dictionary for view creation of product details
    view = { i : [] for i in ['Id','Name', 'Category', 'Price', 'Quantity', 'D

    # Read Product Details
    products = read_file()

    # Loop for every id of products
    for id_ in products.keys():

        # Getting list of product detail throung id
        product = products[ id_ ]

        # Getting 'Name', 'Category', 'Price', 'Quantity', 'Discount' from eac
        Name = product[0]
        Category = product[1]
        Price = product[2]
        Quantity = product[3]

        Discount = product[4]

        # Loading it in view
        view['Id'].append(id_)
        view['Name'].append(Name)
        view['Category'].append(Category)
        view['Price'].append(Price)
        view['Quantity'].append(Quantity)
        view['Discount'].append(Discount)

    return pd.DataFrame( view ) # Return DataFrame format

product_views()
```

Out[4]:

	Id	Name	Category	Price	Quantity	Discount
0	100	dairy milk	chocalate	25	10	10
1	101	5-star	chocalate	15	6	5
2	102	kit-kat	chocalate	20	30	15
3	103	parle-g	biscuit	10	996	0
4	104	monaco	biscuit	5	1076	0
5	105	tiger	biscuit	5	10788	0
6	106	marie-gold	biscuit	10	10443	0
7	107	marie-lite	biscuit	10	1087	5
8	108	50-50	biscuit	5	1098	0
9	109	20-20	biscuit	5	1000	0
10	110	oreo	biscuit	25	2000	0
11	111	parle-gold	biscuit	20	460	0
12	112	mazha	soft drink	25	890	0
13	113	thums-up	soft drink	25	2350	0
14	114	mirinda	soft drink	25	1200	15
15	115	vanila flever	icecream	50	2349	10
16	116	mango flever	icecream	20	7890	4
17	117	chocalate flever	icecream	20	1230	5
18	118	kurkure	snake	10	4560	6
19	119	taka-tak	snake	10	1000	6
20	120	tedhe-medhe	snake	10	1010	1
21	121	maggie	snake	40	1500	8
22	122	uncle-chips	snake	11	500	1
23	123	tomato-chips	snake	35	600	10
24	124	fruiti	soft drink	25	700	0
25	125	spirit	soft drink	25	8000	10
26	126	cadbury	chocalate	25	5689	10
27	127	galaxy	chocalate	25	1230	10
28	128	nestle	chocalate	25	1483	10
29	129	amul	chocalate	25	6500	10
30	130	ferro rocher	chocalate	25	10	10
31	131	hershey	chocalate	25	10	10
32	132	eclair	chocalate	2	100	0
33	134	tata tea 250g	drinks	136	234	10
34	135	sugar	foodgrain	30	8495	0

	Id	Name	Category	Price	Quantity	Discount
35	136	foodex besan ladoo box 200g	sweets	115	8966	15
36	133	godrej jersey ghee 100ml	milk products	82	78	12

Print Details of Product

In [5]:

```
def print_product(prod):
    print('Product Details'.center(40,'-'))
    print('Name     :',prod[0])
    print('Category :',prod[1])
    print('Price    :',prod[2])
    print('Quantity :',prod[3])
    print('Discount :',prod[4])
```

Add new product in Inventory

In [6]:

```
def get_id(ids):

    # intialize value result and offset
    result = 0
    offset = 1

    # loop values by enumerating with "sorted ids" as "i"  and "enumerate valu
    for k,i in enumerate(ids,100):

        # set result equals to k
        result = k

        # if there is missing values in sorted ids then this condition will sa
        if k != int(i):

            # set offset to zero and break loop
            offset = 0
            break

    return result + offset  # return sum of result and offset
```

In [7]:

```
def add_product():

    # Read Product Details
    data = read_file()

    # creating list of ids getting from product details in integer
    ids = np.array( list(data.keys()), dtype = int )

    # Getting new id for new product and passing sorted ids
    new_id = get_id( sorted(ids) )


    # Print the message
    print( '*' * 40, 'Adding new product in Inventory', '-' * 40, sep='\n' )

    #getting inputs of new product
    Name     = input('Enter the Name: ').lower()
    Category = input('Enter the Category: ').lower()
    Price    = float(input('Enter the Price: '))
    Quantity = float(input('Enter the Quantity: '))
    Discount = float(input('Enter the Discount: '))

    print( '-' * 40 )

    # printing new product id of product
    print('Product id of new product :',new_id)

    # fetching data of new product
    data[new_id] = [ Name, Category, Price, Quantity, Discount ]


    # print confirm details of new product
    print_product(data[new_id])

    print( '*' * 40 )

    # saving product details
    write_file(data)

add_product()
```

```
****************************************
Adding new product in Inventory
----------------------------------------
Enter the Name: Bisleri Club Soda: 750 ml
Enter the Category: drinks
Enter the Price: 20
Enter the Quantity: 1324
Enter the Discount: 10
----------------------------------------
Product id of new product : 137
------------Product Details-------------
Name     : bisleri club soda: 750 ml
Category : drinks
Price    : 20.0
Quantity : 1324.0
Discount : 10.0
****************************************
```

Remove product from Inventory

In [8]:

```
def remove_product():

    # Read Product Details
    data = read_file()

    print( '*' * 40 )

    # Getting ids of product to be remove
    id_ = input('Enter the product id: ')

    # Getting product from ids if no product return false
    product = data.get(id_,False)

    # Condition of getting product detaisl
    if product:

        #print product details
        print_product(product)

        print( '-' * 40 )

        # check quantity of product before removing
        if int(product[3]) != 0:

            # print warning message and getting input if you want to proceed
            if input("This product has some quantities Do you want to remove i

                # print final message of loss
                print("This product has loss of Rs", float(product[2]) * int(p

                # removing that product
                data.pop(id_)

                # saving product details in file
                write_file(data)

                #final removal message
                print('Deleted Sucessfully !!')

            #Terminate process
            else:

                #print message
                print("Operation Terminated!!!")

        # Product has zero quantity
        else:

            # removing that product
            data.pop(id_)

            # saving product details in file
            write_file(data)

             #final removal message
            print('Deleted Sucessfully !!')
    # Product of that product id does exist
    else: print("This product does't exist")

    print('*' * 40)


remove_product()
```

```
****************************************
Enter the product id: 132
------------Product Details-------------
Name     : eclair
Category : chocalate
Price    : 2
Quantity : 100
Discount : 0
----------------------------------------
This product has some quantities Do you want to remove it ? (Say Yes/No): Yes
This product has loss of Rs 200.0
Deleted Sucessfully !!
****************************************
```

Product Purchase by customer and Generate Bill

transcation format

```
{
    transation_id_1 : {
        'name':
        'products': [ (product_name, quantiy, price, discount), (product_2),.... ]
    },
    transation_id_1 : { ... },
    ...,
    'id': max_id
}
```

In [9]:

```
def bill_view(carts,m):

    # print the Title of Bill
    print('Sr',"Name".ljust(m), "Quantity",'MRP',"Price","Total")

    print('-'*(m+28))

    #Setting intial Values
    total_saving  = 0
    overall_total = 0

    # setting sr no to 1
    sr = 1

    # where 'n' is Name, 'q' is Quantity, 'd' is discount
    for n, q, mrp, d in carts:

        # price of product = MRP - Discount
        price          = str( float(mrp) - float(d) )

        # Total = Price of Product * Quantity
        total          = str( float(price) * int(q) )

        # Saving = Discount * Quantity
        total_saving  += float(d) * int(q)
        overall_total += float(total)

        # Print everything
        print( str(sr).ljust(2), n.ljust(m), str(q).ljust(8), str(mrp).ljust(3

        # update sr no
        sr += 1

    print('-'*(m+28))

    # print Total Saving and Total
    print('Total Saving:',str(total_saving).ljust(m+1),"Total:",overall_total)




def purchase_product():

    # Read Product Details
    data = read_file()

    #Setting Cart as empty list
    carts = []

    #For print purpose
    max_len = 4

    while True:
        print( '*' * (max_len+28) )

        # Carts is not empty then print the bill
        if carts != []: bill_view(carts,max_len)
        # Enter the product id to be purchase or enter 'pay' or 'done' for pay
        prod_id = input('Scan Product or Enter the product id: ')


        # You enter prod id then this condition satistfies
        if not (prod_id.lower() in 'exit,pay,done'):

            # Enter the quantity of that product
            quant   = int(input('Enter the quantity'))

            print('-'*40)


            # getting products of that product id
            prods   = data[prod_id]

            # fetching products detail as name, quantity mention, mrp, discoun
            prod    = (prods[0], quant, prods[2], prods[-1] )

            # update max_len for printing purpose
            max_len = max( len(prod[0]), max_len)

            # updating the inventory and deduce the quantity of that product
            if int(data[prod_id][3]) > quant:

                # updating the product quantity
                data[prod_id][3] =  int(data[prod_id][3]) - quant

                # updating carts
                carts.append( prod )

            # if quantity is insufficenet then no product exist
            else: print('Not exist')


        # Payment of Product
        if prod_id.lower() in 'paydone':

            # Emter the name of product
            name          = input('Enter your name: ')

            # Read Transcation Details
            trans         = read_file(False)

            # Fetching new id
            new_id        = trans['id']

            # Fetching bill in new id
            trans[new_id] = { 'name': name, 'products': carts }

            # Updating new id
            trans['id']   = str(int(new_id) + 1)

            # Saving Transcation Details
            write_file(trans,False)
            # Saving Product Details
            write_file(data)

            # break statement
            break

        # exit fuction
        if prod_id.lower() == 'exit':break


        # print purpose
        clear_output()


purchase_product()
```

```
*******************************************************
Sr Name                        Quantity MRP Price Total
-------------------------------------------------------
1  parle-g                     4        10  10.0  40.0
2  parle-gold                  5        20  20.0  100.0
3  foodex besan ladoo box 200g 3        115 100.0 300.0
4  mango flever                8        20  16.0  128.0
-------------------------------------------------------
Total Saving: 77.0                         Total: 568.0
Scan Product or Enter the product id: pay
Enter your name: Manas Patil
```

Save Every Bill in a CSV File

In [10]:
```
def save_bill():

    # Read Transaction Details
    data = read_file(False)

    # print the message details of acessing file and names of file
    print('Following Bills in Folder named  "Bills" are as follows')

    # Getting all transaction  id and bills detail
    for id_, bill in data.items():

        # if id is not a number
        if id_ == 'id': continue

        # Fetching Bill product details
        products = bill['products']

        # seting overall total and saving
        overall_total    = 0
        saving           = 0

        # Updating products variable of each product
        for sr in range(len(products)):

            # Sr no in each products
            products[sr]     = [sr+1] + products[sr]

            # Saving = Discount * quantity
            saving          += float(products[sr][-1] * products[sr][-3])

            # Price = MRP - Discount ---> Discount changes to Price
            products[sr][-1] = products[sr][-2] - products[sr][-1]

            # Total = Price * quantity
            total            = float(products[sr][-1] * products[sr][-3])

            # Summing Overall total
            overall_total   += total

            # append total of each product
            products[sr].append( total )

        # Last section of total saving and Overall Total
        products.append( ['Total Saving',saving,'','','Total',total] )

        # converting in dataframe
        df = pd.DataFrame(products,columns=['Sr','Name','Quantity','MRP','Pric

        # File name of that bill as --> 'bill_no_id_name.csv'
        file_name = 'Bill_no_' + id_ + '_' + '_'.join(bill['name'].split()) +

        # if Bills folders not exist then create that product
        if 'Bills' not in os.listdir(): os.mkdir('Bills')

        # Saving Bill from datafrome to csv format
        df.to_csv('Bills/'+ file_name,index=False)
        #printing file
        print(file_name)

save_bill()
```

```
Following Bills in Folder named  "Bills" are as follows
Bill_no_1000_Ashish_Jangra.csv
Bill_no_1001_Pratik_Maurya.csv
Bill_no_1002_Lokesh_Parab.csv
Bill_no_1003_Suyash_Salvi.csv
Bill_no_1004_Manas_Patil.csv
```