```
import csv
import random

def run_internal():

    fieldnames=['agent_id',
                'monthyear',
                'dept_id',
                'property_type_id',
                'totalCompletionTime',
                'totalDaysToLease',
                'totalScheduledMins',
                'totalAdherenceMins',
                'totalHandlingTime',
                'receivedCalls',
                'totalStartTime',
                'totalCollabProj',
                'totalIndProj',
                'totalStaff',
                'totalDevelopmentManpower',
                'totalInquiryResponseTime',
                'totalInquiries',
                'totalTaskDuration',
                'totalTasks',
                'totalPropertiesSold',
                'totalPropertiesLeased',
                'totalNewUnits',
                'salesAgentSatisfactionRating']

    writer = csv.DictWriter(open("population/property_internal.csv", "w",newline=''), fieldnames=fieldnames)
    print("Generating INTERNAL values.");
    x=0;
    for a in range(1,5):
        for b in range(1,37):
            for c in range(1,3):
                for d in range(1,11):
                    o = random.randint(0,4);
                    if(o > 0 & a != 2):
                        writer.writerow(dict([
                                ('agent_id', d),
                                ('monthyear', b),
                                ('dept_id', c),
                                ('property_type_id',a + (1*random.randint(0,1))),
                                ('totalCompletionTime', random.randint(1,300)),
                                ('totalDaysToLease', random.randint(1,30)),
                                ('totalScheduledMins', random.randint(360,480)),
                                ('totalAdherenceMins', random.randint(0,150)),
                                ('totalHandlingTime', random.randint(0,240)),
                                ('receivedCalls', random.randint(0,500)),
                                ('totalStartTime', random.randint(1,20)),
                                ('totalCollabProj', random.randint(0,5)),
                                ('totalIndProj', random.randint(0,100)),
                                ('totalStaff', random.randint(0,5)),
                                ('totalDevelopmentManpower', random.randint(0,100)),
                                ('totalInquiryResponseTime', random.randint(0,120)),
                                ('totalInquiries', random.randint(0,100)),
                                ('totalTaskDuration', random.randint(0,12000)),
                                ('totalTasks', random.randint(0,100)),
                                ('totalPropertiesSold',
                                random.randint(0,4)*(round(random.uniform(0.1,1.9)))),
                                ('totalPropertiesLeased',
                                random.randint(0,5)*(round(random.uniform(0.1,1.9)))),
                                ('totalNewUnits',
                                random.randint(0,1)*(round(random.uniform(0.1,1.9)))),
                                ('salesAgentSatisfactionRating', random.randint(0,100))
                        ]));
```

```
                        x=x+1;
                        #print("INTERNAL -> Row "+str(x)+"; ID Combination:
                        "+str(a)+"-"+str(b)+"-"+str(c));
    print("Generated "+str(x)+" rows for INTERNAL.");

def run_financial():

    fieldnames=['monthyear',
                'property_type_id',
                'city_id',
                'dept_id',
                'agent_id',
                'opExpenses',
                'totalTransactions',
                'netProfit',
                'costsOfGoodsSold',
                'interest',
                'taxes',
                'revenue',
                'minSoldPrice',
                'nonCashExpenses',
                'earningsBeforeTaxesInterest',
                'totalManagedSpend',
                'totalMaintenanceCosts',
                'totalCostAvoid',
                'totalRentReadyCosts',
                'totalRentIncome',
                'totalRentCosts',
                'totalMarketingExpenses',
                'totalCallCosts',
                'totalCustAcquisitionCost',
                'totalTrxValue',
                'totalTrxCommission',
                'totalUncollectedTenantFees']

    writer = csv.DictWriter(open("population/property_financial.csv", "w",newline=''),
    fieldnames=fieldnames)
    print("Generating FINANCIAL values.");
    y=0;
    for a in range(1,37):
        for b in range(1,5):
            for c in range(1,4):
                for d in range(1,3):
                    for e in range(1,11):
                        netProfit = random.randint(0,500000);
                        opEx = netProfit*1/5;
                        CGS = netProfit*2/5;
                        interest = netProfit*0.06;
                        taxes = netProfit*0.15;
                        revenue = netProfit-opEx-CGS-interest-taxes;
                        endProfit = netProfit-taxes-interest;
                        transcationsCount = random.randint(1,11);
                        writer.writerow(dict([
                        ('monthyear', a),
                        ('property_type_id', b),
                        ('city_id', c),
                        ('dept_id', d),
                        ('agent_id',e),
                        ('opExpenses', opEx),
                        ('totalTransactions', transcationsCount),
                        ('netProfit', endProfit),
                        ('costsOfGoodsSold', CGS),
                        ('interest', interest),
                        ('taxes', taxes),
                        ('revenue', revenue),
                        ('minSoldPrice', netProfit/20*(round(random.uniform(0.2,1.8),
                        2))),
```

```
                        ('nonCashExpenses', opEx/8*(round(random.uniform(0.1,1.9), 2))),
                        ('earningsBeforeTaxesInterest', netProfit),
                        ('totalManagedSpend', opEx/8*(round(random.uniform(0.4,1.6),
                        2))),
                        ('totalMaintenanceCosts',
                        opEx/8*(round(random.uniform(0.8,1.2), 2))),
                        ('totalCostAvoid', random.randint(0,10)),
                        ('totalRentReadyCosts', opEx/16*(round(random.uniform(0.9,1.1),
                        2)) ),
                        ('totalRentIncome', opEx/8),
                        ('totalRentCosts', opEx/16*(round(random.uniform(0.8,1.2), 2)) ),
                        ('totalMarketingExpenses',
                        opEx/8*(round(random.uniform(0.4,1.6), 2))),
                        ('totalCallCosts', opEx/8*(round(random.uniform(0.2,1.98), 2))),
                        ('totalCustAcquisitionCost',
                        opEx/8*(round(random.uniform(0.1,1.9), 2))),
                        ('totalTrxValue', netProfit/transcationsCount),
                        ('totalTrxCommission', netProfit/transcationsCount*0.2),
                        ('totalUncollectedTenantFees', random.randint(0,5000))
                        ]));
                        y=y+1;
                        #print("FINANCIAL -> Row "+str(y)+";  ID Combination:
                        "+str(a)+"-"+str(b)+"-"+str(c));
    print("Generated "+str(y)+" rows for FINANCIAL.");

def run_organizational():

    fieldnames=['agent_id',
                'monthyear',
                'dept_id',
                'Certifications',
                'totalTrainingsConducted',
                'totalPromotions',
                'totalAwards',
                'totalComplaintsResolved',
                'totalComplaints',
                'totalDealsClosed',
                'empPerfRating',
                'totalResolvedTasks',
                'totalTasks',
                'totalReworkedTasks',
                'totalRequests',
                'totalRequestsAddressingTime',
                'totalYearsOfExperience',
                'totalReassignments',
                'totalEmployees',
                'totalTrainedEmployees']

    writer = csv.DictWriter(open("population/property_organizational.csv",
    "w",newline=''), fieldnames=fieldnames)
    print("Generating ORGANIZATIONAL values.");
    y=0;
    for a in range(1,11):
        for b in range(1,37):
            for c in range(1,3):
                tasks = random.randint(0,150);
                unresolved = tasks*(round(random.uniform(0,0.4), 2));
                resolved = tasks - unresolved;
                reworked = tasks*(round(random.uniform(0,0.2), 2));
                requests = tasks*(round(random.uniform(1,1.4), 2));
                complaints = tasks*(round(random.uniform(1,1.4), 2));
                compresolved = complaints*(round(random.uniform(0.75,1), 2));
                writer.writerow(dict([
                        ('agent_id', a),
                        ('monthyear', b),
                        ('dept_id',c),
                        ('Certifications',random.randint(0,1)),
```

```
                        ('totalTrainingsConducted',random.randint(0,3)),
                        ('totalPromotions',random.randint(0,1)),
                        ('totalAwards',random.randint(0,1)),
                        ('totalComplaintsResolved',compresolved),
                        ('totalComplaints',complaints),
                        ('totalDealsClosed',random.randint(0,10)),
                        ('empPerfRating',random.randint(0,5)),
                        ('totalResolvedTasks',resolved),
                        ('totalTasks',tasks),
                        ('totalReworkedTasks',reworked),
                        ('totalRequests',requests),
                        ('totalRequestsAddressingTime',random.randint(0,200)),
                        ('totalYearsOfExperience',random.randint(0,20)),
                        ('totalReassignments',random.randint(0,2)),
                        ('totalEmployees',random.randint(0,2)),
                        ('totalTrainedEmployees',random.randint(0,2))
                ]));
                y=y+1;
                #print("ORGANIZATIONAL -> Row "+str(y)+";  ID Combination:
                "+str(a)+"-"+str(b)+"-"+str(c));
    print("Generated "+str(y)+" rows for ORGANIZATIONAL.");

def run_customer():

    fieldnames=['monthyear',
                'agent_id',
                'property_type_id',
                'city_id',
                'dept_id',
                'totalTransactions',
                'totalComplaints',
                'abandonedCalls',
                'receivedCalls',
                'totalCustomers',
                'totalHandlingTime',
                'totalDeals',
                'totalDealsClosed',
                'totalProductCost',
                'totalAvailableProperties',
                'totalLeaseInquiries',
                'totalCustomerRatings',
                'totalUnitTurnoverMonths',
                'totalPropertyManagementFees',
                'totalProperties',
                'totalTenants',
                'totalWaitTime',
                'totalCloseDealDays',
                'totalCustomerComments']

    writer = csv.DictWriter(open("population/property_customers.csv", "w",newline=''),
    fieldnames=fieldnames)
    print("Generating CUSTOMER values.");
    y=0;
    for a in range(1,37):
        for b in range(1,11):
            for c in range(1,5):
                for d in range(1,4):
                    for e in range(1,3):
                        calls = random.randint(0,200);
                        writer.writerow(dict([
                                ('monthyear', a),
                                ('agent_id', b),
                                ('property_type_id', c),
                                ('city_id', d),
                                ('dept_id', e),
                                ('totalTransactions', random.randint(0,15)),
                                ('totalComplaints', random.randint(0,10)),
```

```
                                    ('abandonedCalls', calls*2/5),
                                    ('receivedCalls', calls),
                                    ('totalCustomers', random.randint(0,5)),
                                    ('totalHandlingTime', calls * random.randint(1,30)),
                                    ('totalDeals', random.randint(0,3)),
                                    ('totalDealsClosed', random.randint(0,2)),
                                    ('totalProductCost', random.randint(1000000,2000000)),
                                    ('totalAvailableProperties', random.randint(1,3)),
                                    ('totalLeaseInquiries', random.randint(100,300)),
                                    ('totalCustomerRatings', random.randint(1,5)),
                                    ('totalUnitTurnoverMonths', random.randint(12,60)),
                                    ('totalPropertyManagementFees',
                                    random.randint(11000,100000)),
                                    ('totalProperties', random.randint(100,200)),
                                    ('totalTenants', random.randint(2,5)),
                                    ('totalWaitTime', random.randint(0,10)),
                                    ('totalCloseDealDays', random.randint(5,20)),
                                    ('totalCustomerComments', random.randint(0,20))
                            ]));
                            y=y+1;
                            #print("CUSTOMER -> Row "+str(y)+";  ID Combination:
                            "+str(a)+"-"+str(b)+"-"+str(c)+"-"+str(d)+"-"+str(e));
        print("Generated "+str(y)+" rows for CUSTOMER.");


    ########################################

    run_financial();
    run_internal();
    run_organizational();
    run_customer();

```