

```

1  import csv
2  import random
3
4  def run_internal():
5
6      fieldnames=['agent_id',
7                  'monthyear',
8                  'dept_id',
9                  'property_type_id',
10                 'totalCompletionTime',
11                 'totalDaysToLease',
12                 'totalScheduledMins',
13                 'totalAdherenceMins',
14                 'totalHandlingTime',
15                 'receivedCalls',
16                 'totalStartTime',
17                 'totalCollabProj',
18                 'totalIndProj',
19                 'totalStaff',
20                 'totalDevelopmentManpower',
21                 'totalInquiryResponseTime',
22                 'totalInquiries',
23                 'totalTaskDuration',
24                 'totalTasks',
25                 'totalPropertiesSold',
26                 'totalPropertiesLeased',
27                 'totalNewUnits',
28                 'salesAgentSatisfactionRating']
29
30     writer = csv.DictWriter(open("population/property_internal.csv", "w",newline=''),
31                             fieldnames=fieldnames)
32     print("Generating INTERNAL values.");
33     x=0;
34     for a in range(1,5):
35         for b in range(1,37):
36             for c in range(1,3):
37                 for d in range(1,11):
38                     o = random.randint(0,4);
39                     if(o > 0 & a != 2):
40                         writer.writerow(dict([
41                             ('agent_id', d),
42                             ('monthyear', b),
43                             ('dept_id', c),
44                             ('property_type_id',a + (1*random.randint(0,1))),
45                             ('totalCompletionTime', random.randint(1,300)),
46                             ('totalDaysToLease', random.randint(1,30)),
47                             ('totalScheduledMins', random.randint(360,480)),
48                             ('totalAdherenceMins', random.randint(0,150)),
49                             ('totalHandlingTime', random.randint(0,240)),
50                             ('receivedCalls', random.randint(0,500)),
51                             ('totalStartTime', random.randint(1,20)),
52                             ('totalCollabProj', random.randint(0,5)),
53                             ('totalIndProj', random.randint(0,100)),
54                             ('totalStaff', random.randint(0,5)),
55                             ('totalDevelopmentManpower', random.randint(0,100)),
56                             ('totalInquiryResponseTime', random.randint(0,120)),
57                             ('totalInquiries', random.randint(0,100)),
58                             ('totalTaskDuration', random.randint(0,12000)),
59                             ('totalTasks', random.randint(0,100)),
60                             ('totalPropertiesSold',
61                             random.randint(0,4)*(round(random.uniform(0.1,1.9)))),
62                             ('totalPropertiesLeased',
63                             random.randint(0,5)*(round(random.uniform(0.1,1.9)))),
64                             ('totalNewUnits',
65                             random.randint(0,1)*(round(random.uniform(0.1,1.9)))),
66                             ('salesAgentSatisfactionRating', random.randint(0,100))
67                             ]));

```

```

64         x=x+1;
65         #print("INTERNAL -> Row "+str(x)+"; ID Combination:
        "+str(a)+"-"+str(b)+"-"+str(c));
66     print("Generated "+str(x)+" rows for INTERNAL.");
67
68 def run_financial():
69
70     fieldnames=['monthyear',
71                 'property_type_id',
72                 'city_id',
73                 'dept_id',
74                 'agent_id',
75                 'opExpenses',
76                 'totalTransactions',
77                 'netProfit',
78                 'costsOfGoodsSold',
79                 'interest',
80                 'taxes',
81                 'revenue',
82                 'minSoldPrice',
83                 'nonCashExpenses',
84                 'earningsBeforeTaxesInterest',
85                 'totalManagedSpend',
86                 'totalMaintenanceCosts',
87                 'totalCostAvoid',
88                 'totalRentReadyCosts',
89                 'totalRentIncome',
90                 'totalRentCosts',
91                 'totalMarketingExpenses',
92                 'totalCallCosts',
93                 'totalCustAcquisitionCost',
94                 'totalTrxValue',
95                 'totalTrxCommission',
96                 'totalUncollectedTenantFees']
97
98     writer = csv.DictWriter(open("population/property_financial.csv", "w",newline=''),
99                             fieldnames=fieldnames)
100     print("Generating FINANCIAL values.");
101     y=0;
102     for a in range(1,37):
103         for b in range(1,5):
104             for c in range(1,4):
105                 for d in range(1,3):
106                     for e in range(1,11):
107                         netProfit = random.randint(0,500000);
108                         opEx = netProfit*1/5;
109                         CGS = netProfit*2/5;
110                         interest = netProfit*0.06;
111                         taxes = netProfit*0.15;
112                         revenue = netProfit-opEx-CGS-interest-taxes;
113                         endProfit = netProfit-taxes-interest;
114                         transactionsCount = random.randint(1,11);
115                         writer.writerow(dict([
116                             ('monthyear', a),
117                             ('property_type_id', b),
118                             ('city_id', c),
119                             ('dept_id', d),
120                             ('agent_id',e),
121                             ('opExpenses', opEx),
122                             ('totalTransactions', transactionsCount),
123                             ('netProfit', endProfit),
124                             ('costsOfGoodsSold', CGS),
125                             ('interest', interest),
126                             ('taxes', taxes),
127                             ('revenue', revenue),
128                             ('minSoldPrice', netProfit/20*(round(random.uniform(0.2,1.8),
129

```

```

128         ('nonCashExpenses', opEx/8*(round(random.uniform(0.1,1.9), 2))),
129         ('earningsBeforeTaxesInterest', netProfit),
130         ('totalManagedSpend', opEx/8*(round(random.uniform(0.4,1.6),
131         2))),
132         ('totalMaintenanceCosts',
133         opEx/8*(round(random.uniform(0.8,1.2), 2))),
134         ('totalCostAvoid', random.randint(0,10)),
135         ('totalRentReadyCosts', opEx/16*(round(random.uniform(0.9,1.1),
136         2))),
137         ('totalRentIncome', opEx/8),
138         ('totalRentCosts', opEx/16*(round(random.uniform(0.8,1.2), 2))),
139         ('totalMarketingExpenses',
140         opEx/8*(round(random.uniform(0.4,1.6), 2))),
141         ('totalCallCosts', opEx/8*(round(random.uniform(0.2,1.98), 2))),
142         ('totalCustAcquisitionCost',
143         opEx/8*(round(random.uniform(0.1,1.9), 2))),
144         ('totalTrxValue', netProfit/transcationsCount),
145         ('totalTrxCommission', netProfit/transcationsCount*0.2),
146         ('totalUncollectedTenantFees', random.randint(0,5000))
147     ]));
148     y=y+1;
149     #print("FINANCIAL -> Row "+str(y)+"; ID Combination:
150     "+str(a)+"-"+str(b)+"-"+str(c));
151     print("Generated "+str(y)+" rows for FINANCIAL.");
152
153 def run_organizational():
154
155     fieldnames=['agent_id',
156                 'monthyear',
157                 'dept_id',
158                 'Certifications',
159                 'totalTrainingsConducted',
160                 'totalPromotions',
161                 'totalAwards',
162                 'totalComplaintsResolved',
163                 'totalComplaints',
164                 'totalDealsClosed',
165                 'empPerfRating',
166                 'totalResolvedTasks',
167                 'totalTasks',
168                 'totalReworkedTasks',
169                 'totalRequests',
170                 'totalRequestsAddressingTime',
171                 'totalYearsOfExperience',
172                 'totalReassignments',
173                 'totalEmployees',
174                 'totalTrainedEmployees']
175
176     writer = csv.DictWriter(open("population/property_organizational.csv",
177     "w",newline=''), fieldnames=fieldnames)
178     print("Generating ORGANIZATIONAL values.");
179     y=0;
180     for a in range(1,11):
181         for b in range(1,37):
182             for c in range(1,3):
183                 tasks = random.randint(0,150);
184                 unresolved = tasks*(round(random.uniform(0,0.4), 2));
185                 resolved = tasks - unresolved;
186                 reworked = tasks*(round(random.uniform(0,0.2), 2));
187                 requests = tasks*(round(random.uniform(1,1.4), 2));
188                 complaints = tasks*(round(random.uniform(1,1.4), 2));
189                 compresolved = complaints*(round(random.uniform(0.75,1), 2));
190                 writer.writerow(dict([
191                     ('agent_id', a),
192                     ('monthyear', b),
193                     ('dept_id',c),
194                     ('Certifications',random.randint(0,1)),

```

```

188         ('totalTrainingsConducted', random.randint(0,3)),
189         ('totalPromotions', random.randint(0,1)),
190         ('totalAwards', random.randint(0,1)),
191         ('totalComplaintsResolved', compresolved),
192         ('totalComplaints', complaints),
193         ('totalDealsClosed', random.randint(0,10)),
194         ('empPerfRating', random.randint(0,5)),
195         ('totalResolvedTasks', resolved),
196         ('totalTasks', tasks),
197         ('totalReworkedTasks', reworked),
198         ('totalRequests', requests),
199         ('totalRequestsAddressingTime', random.randint(0,200)),
200         ('totalYearsOfExperience', random.randint(0,20)),
201         ('totalReassignments', random.randint(0,2)),
202         ('totalEmployees', random.randint(0,2)),
203         ('totalTrainedEmployees', random.randint(0,2))
204     ]));
205     y=y+1;
206     #print("ORGANIZATIONAL -> Row "+str(y)+"; ID Combination:
207         "+str(a)+"-"+str(b)+"-"+str(c));
208
209     print("Generated "+str(y)+" rows for ORGANIZATIONAL.");
210
211 def run_customer():
212
213     fieldnames=['monthyear',
214                 'agent_id',
215                 'property_type_id',
216                 'city_id',
217                 'dept_id',
218                 'totalTransactions',
219                 'totalComplaints',
220                 'abandonedCalls',
221                 'receivedCalls',
222                 'totalCustomers',
223                 'totalHandlingTime',
224                 'totalDeals',
225                 'totalDealsClosed',
226                 'totalProductCost',
227                 'totalAvailableProperties',
228                 'totalLeaseInquiries',
229                 'totalCustomerRatings',
230                 'totalUnitTurnoverMonths',
231                 'totalPropertyManagementFees',
232                 'totalProperties',
233                 'totalTenants',
234                 'totalWaitTime',
235                 'totalCloseDealDays',
236                 'totalCustomerComments']
237
238     writer = csv.DictWriter(open("population/property_customers.csv", "w", newline=''),
239                             fieldnames=fieldnames)
240     print("Generating CUSTOMER values.");
241     y=0;
242     for a in range(1,37):
243         for b in range(1,11):
244             for c in range(1,5):
245                 for d in range(1,4):
246                     for e in range(1,3):
247                         calls = random.randint(0,200);
248                         writer.writerow(dict([
249                             ('monthyear', a),
250                             ('agent_id', b),
251                             ('property_type_id', c),
252                             ('city_id', d),
253                             ('dept_id', e),
254                             ('totalTransactions', random.randint(0,15)),
255                             ('totalComplaints', random.randint(0,10)),

```

```

253         ('abandonedCalls', calls*2/5),
254         ('receivedCalls', calls),
255         ('totalCustomers', random.randint(0,5)),
256         ('totalHandlingTime', calls * random.randint(1,30)),
257         ('totalDeals', random.randint(0,3)),
258         ('totalDealsClosed', random.randint(0,2)),
259         ('totalProductCost', random.randint(1000000,2000000)),
260         ('totalAvailableProperties', random.randint(1,3)),
261         ('totalLeaseInquiries', random.randint(100,300)),
262         ('totalCustomerRatings', random.randint(1,5)),
263         ('totalUnitTurnoverMonths', random.randint(12,60)),
264         ('totalPropertyManagementFees',
random.randint(11000,100000)),
265         ('totalProperties', random.randint(100,200)),
266         ('totalTenants', random.randint(2,5)),
267         ('totalWaitTime', random.randint(0,10)),
268         ('totalCloseDealDays', random.randint(5,20)),
269         ('totalCustomerComments', random.randint(0,20))
270     ]));
271     y=y+1;
272     #print("CUSTOMER -> Row "+str(y)+"; ID Combination:
"+str(a)+"-"+str(b)+"-"+str(c)+"-"+str(d)+"-"+str(e));
273     print("Generated "+str(y)+" rows for CUSTOMER.");
274
275
276
277     #####
278
279     run_financial();
280     run_internal();
281     run_organizational();
282     run_customer();
283

```