

Decision Model for DMCommunity March-2024 Challenge “Analyze Employees”

Problem

The problem was [formulated](#) as follows:

“Let’s help an HR office create a rules-based service to analyze its employees. Each employee has a unique name, age, gender, marital status, one or more locations (places of residence), number of children, salary, and probably more attributes. This information is coming to the service as a JSON request such as in this [file](#). Your service should find answers to the following questions:

- What is the current total number of employees?
- How many children do all employees have? How many children does the average employee have?
- What is an average salary? What are the maximal and minimal salaries?
- How many employees are single?
- In which states do the employees have residences?
- How many people are inside 20% of highest-paid employees? Who are these high-paid employees?
- List employees living in certain zip codes.

Your decision service should be flexible for adding more questions that analyze its employees.”

OpenRules-based Solution

We created the OpenRules project “AnalyzeEmployees” which now can be downloaded along with other OpenRules [samples](#). We downloaded the provided test file in JSON format and placed it in the subfolder “data” in the file “Test1.json” (you can see it in the Appendix at the end of this document).

Then we created the glossary with all input decision variables corresponding to the provided JSON file. We added the business concept “Results” which contains output decision variables (with red background) for the answers to all the above questions:

Glossary glossary			
Variable Name	Business Concept	Attribute	Type
Company Name	Company	companyName	String
Employees		employees	Employee[]
Selected Zip Codes		selectedZipCodes	String[]
Name	Employee	name	String
Age		age	int
Gender		gender	String
Marital Status		maritalStatus	String
Locations		locations	Location[]
Number of Children		children	int
Salary		salary	double
Location Id	Location	id	String
Street		street	String
Zip Code		zipCode	String
State		state	String

Total Number of Employees	Results	totalNumberOfEmployees	int
Total Number of Children		totalNumberOfChildren	int
Average Number of Children per Employee		averageNumberOfChildren	double
Average Salary		averageSalary	double
Max Salary		maxSalary	double
Min Salary		minSalary	double
Total Salary		totalSalary	double
High Salary		highSalary	double
Number of High-Paid Employees		numberOfHighPaidEmployees	int
High-Paid Employees		highPaidEmployees	Employee[]
Number of Single Employees		numberOfSingleEmployees	int
Employees at Selected Zip Codes		employeesAtSelectedZipCodes	String[]
Distinct Employee States		distinctStatesOfEmployees	List<String>

Before writing any rules, we created a test table in Excel that uses the provided JSON file:

DecisionTest testCases				
#	ActionUseJson	ActionExpect	ActionExpect	ActionExpect
Test ID	JSON File	High Salary	Number of High-Paid Employees	Employees at Selected Zip Codes
1	Test1.json	176000	3	Warner, Stevens, Smith, Doe

To answer the questions we could use decision tables with iterations. For example, to calculate “Total Number of Employees” and other “totals” we could use the “for each” loop:

Decision CalculateTotals [for each Employee in Employees]					
Action		Action		Action	
Total Number of Employees		Total Number of Children		Total Salary	
+	1	+	Number of Children	+	Salary

However, we wanted to avoid using loops and instead utilize OpenRules natural language expressions with predefined functions. So, we created the decision table that assigns the results of the functions in the column “Value” to the decision variables in the column “Variable”:

Decision ApplyFunctions	
ActionAssign	
Variable	Value
Total Number of Employees	Count of Employees
Total Number of Children	Sum of Number of Children of Employees
Average Number of Children per Employee	Total Number of Children / Total Number of Employees
Average Salary	Average of Salary of Employees
Max Salary	Max of Salary of Employees
Min Salary	Min of Salary of Employees
High Salary	Max Salary * 0.8
Distinct Employee States	Distinct of State of Locations of Employees
Total Salary	Sum of Salary of Employees

We expect that this table is self-explanatory. The expression “**Count of Employees**” calculates the total number of elements in the collection of Employees (in this case defined as an array). The expression “**Average of Salary of Employees**” calculates the average salary among all employees.

The most tricky question “In which states do the employees have residences?” would require two nested loops: one over Employees and another – over Locations inside each Employee. So, the expression “**Distinct of State of Locations of Employees**” will calculate the number of distinct states (without duplications) where employees have residencies without nested loops.

However, we do not have standard functions for everything. For instance, when calculating the lists of high-paid employees and single employees we still need to use a decision table with a loop over Employees:

Decision IterateOverEmployees [for each Employee in Employees]										
Condition		Condition		Action		Action		Action		ActionExecute
Salary of Employee		Marital Status of Employee		Number of High-Paid Employees		High-Paid Employees		Number of Single Employees		Rules
>=	High Salary			+	1	Add	Employee			
		Is	Single					+	1	
										EmployeesAtZipCodes

To answer the last question “List employees living in certain zip codes” this table will invoke another decision table that for each employee iterates over his/her locations:

Decision EmployeesAtZipCodes [for each Location in Locations of Employee]			
Condition		Action	
Selected Zip Codes		Employees at Selected Zip Codes	
Include	Zip Code of Location	+	Name of Employee

Note that “Selected Zip Codes” are defined in the [JSON test](#).

At the very high level, our decision model will call the following decision table:

Decision AnalyzeEmployees
ActionExecute
Decision
ApplyFunctions
IterateOverEmployees

Deployed REST Decision Service

With one button push, we deployed this model as a local REST service and tested it using POSTMAN. Here are the results:

The screenshot shows a Postman interface for a POST request to `http://localhost:8080/analyze-employees`. The request body is in JSON format, containing a company object with a list of employees and their locations.

```

1  {
2  |  "company" : {
3  |    "companyName" : "ABC",
4  |    "employees" : [
5  |      {
6  |        "name" : "Robinson",
7  |        "age" : 25,
8  |        "gender" : "Female",
9  |        "maritalStatus" : "Married",
10 |        "locations" : [
11 |          {
12 |            "id" : "RobinsonLoc1",
13 |            "street" : "Main Str",
14 |            "zipCode" : "08831",

```

```
Body v 200 OK 13 ms 692 B Save as example
Pretty Raw Preview Visualize JSON
1 {
2   "decisionStatusCode": 200,
3   "rulesExecutionTimeMs": 3.7387,
4   "response": {
5     "results": {
6       "totalNumberOfEmployees": 12,
7       "totalNumberOfChildren": 17,
8       "averageNumberOfChildren": 1.0,
9       "averageSalary": 134583.33333333334,
10      "maxSalary": 220000.0,
11      "minSalary": 40000.0,
12      "totalSalary": 1615000.0,
13      "highSalary": 176000.0,
14      "numberOfHighPaidEmployees": 3,
15     "highPaidEmployees": [...],
82    ],
83    "numberOfSingleEmployees": 5,
84    "employeesAtSelectedZipCodes": [
85      "Warner",
86      "Stevens",
87      "Smith",
88      "Doe"
89    ],
90    "distinctStatesOfEmployees": [
91      "NJ",
92      "FL",
93      "CA"
94    ]
95  }
96 }
```

Appendix. Test file in JSON format:

```
{
  "company": {
    "companyName": "ABC",
    "employees": [ {
      "name": "Robinson",
      "age": 25,
      "gender": "Female",
      "maritalStatus": "Married",
      "locations": [ {
        "id": "RobinsonLoc1",
        "street": "Main Str",
        "zipCode": "08831",
```

```
"state" : "NJ"
}, {
  "id" : "RobinsonLoc2",
  "street" : "Ocean Drive",
  "zipCode" : "33019",
  "state" : "FL"
}],
"children" : 2,
"salary" : 220000.0
}, {
  "name" : "Warner",
  "age" : 45,
  "gender" : "Male",
  "maritalStatus" : "Married",
  "locations" : [ {
    "id" : "WarnerLoc",
    "street" : "Maple Street",
    "zipCode" : "08817",
    "state" : "NJ"
  } ],
  "children" : 0,
  "salary" : 150000.0
}, {
  "name" : "Stevens",
  "age" : 24,
  "gender" : "Male",
  "maritalStatus" : "Single",
  "locations" : [ {
    "id" : "StevensLoc",
    "street" : "Regency Drive",
    "zipCode" : "08817",
    "state" : "NJ"
  } ],
  "children" : 0,
  "salary" : 135000.0
}, {
  "name" : "White",
  "age" : 32,
  "gender" : "Female",
  "maritalStatus" : "Married",
  "locations" : [ {
    "id" : "WhiteLoc",
    "street" : "Green Road",
    "zipCode" : "33019",
    "state" : "FL"
  } ],
  "children" : 2,
  "salary" : 195000.0
```

```
}, {
  "name" : "Smith",
  "age" : 46,
  "gender" : "Male",
  "maritalStatus" : "Single",
  "locations" : [ {
    "id" : "SmithLoc",
    "street" : "Maple Street",
    "zipCode" : "90027",
    "state" : "CA"
  } ],
  "children" : 1,
  "salary" : 120000.0
}, {
  "name" : "Green",
  "age" : 28,
  "gender" : "Female",
  "maritalStatus" : "Married",
  "locations" : [ {
    "id" : "GreenLoc",
    "street" : "Green Road",
    "zipCode" : "33019",
    "state" : "FL"
  } ],
  "children" : 3,
  "salary" : 140000.0
}, {
  "name" : "Brown",
  "age" : 32,
  "gender" : "Male",
  "maritalStatus" : "Married",
  "locations" : [ {
    "id" : "BrownLoc",
    "street" : "Green Road",
    "zipCode" : "33019",
    "state" : "FL"
  } ],
  "children" : 2,
  "salary" : 65000.0
}, {
  "name" : "Klaus",
  "age" : 54,
  "gender" : "Male",
  "maritalStatus" : "Married",
  "locations" : [ {
    "id" : "KlausLoc",
    "street" : "Green Road",
    "zipCode" : "33019",
```

```
    "state" : "FL"
  }],
  "children" : 1,
  "salary" : 85000.0
}, {
  "name" : "Houston",
  "age" : 47,
  "gender" : "Female",
  "maritalStatus" : "Single",
  "locations" : [ {
    "id" : "HoustonLoc",
    "street" : "Green Road",
    "zipCode" : "33019",
    "state" : "FL"
  } ],
  "children" : 2,
  "salary" : 135000.0
}, {
  "name" : "Long",
  "age" : 29,
  "gender" : "Male",
  "maritalStatus" : "Married",
  "locations" : [ {
    "id" : "LongLoc",
    "street" : "Green Road",
    "zipCode" : "33019",
    "state" : "FL"
  } ],
  "children" : 3,
  "salary" : 40000.0
}, {
  "name" : "Short",
  "age" : 22,
  "gender" : "Male",
  "maritalStatus" : "Single",
  "locations" : [ {
    "id" : "ShortLoc",
    "street" : "Green Road",
    "zipCode" : "33019",
    "state" : "FL"
  } ],
  "children" : 0,
  "salary" : 120000.0
}, {
  "name" : "Doe",
  "age" : 21,
  "gender" : "Female",
  "maritalStatus" : "Single",
```



```
"locations" : [ {
  "id" : "DoeLoc1",
  "street" : "Green Road",
  "zipCode" : "33019",
  "state" : "FL"
}, {
  "id" : "DoeLoc2",
  "street" : "Morgan Street",
  "zipCode" : "33020",
  "state" : "FL"
}, {
  "id" : "DoeLoc3",
  "street" : "Lyric Ave",
  "zipCode" : "90027",
  "state" : "CA"
} ],
"children" : 1,
"salary" : 210000.0
}],
"selectedZipCodes" : [ "08817", "90027" ]
}
}
```