January 2016 Challenge Decision Table For Vacation Days Calculation

Author David Durant RapidGen Software

The challenge is described here.

By using the principles of defining the unique conditions and actions and filling out the rules, simplifying the table and adding an ELSE rule we have a decision table for the January Challenge.

Red parts are not part of the table but give clarity

Initial Action						
Days_leave <- 22			RULES			
Condition	1	2	3	4	5	ELSE
AGE < 18	Y	N	N	N	N	ELSE
AGE >= 60	N	Y	-	N	N	-
SERVICE >= 30	N	-	Y	N	N	-
SERVICE >= 15	N	-	-	Y	N	-
AGE >= 45	N	-	-	-	Y	-
Action						
Days_leave + 5	X	X	X	.	.	.
Days_leave + 2	.	.	.	X	X	.
Days_leave + 3	.	X	X	.	.	.

Every employee receives at least 22 days.
Row 1: an initial action to assign 22 days regardless of anything else happening.

Only employees younger than 18 or at least 60 years, or employees with at least 30 years of service will receive 5 extra days. Condition rows 1-3 test on age less than 18 (rule 1), age at least 60 (rule 2) and at least 30 years of service (rule 3). Where any of these rules 1-3 satisfy, Days_leave (vacation days) is increased by 5 in the first action row.

Employees with at least 30 years of service is shown above as the third condition and is satisfied by rule 3. *Employees of age 60 or more* is shown as the second condition row satisfied by rule 2. Both of these receive 3 extra days, on top of possible additional days already supplied. Days_leave is increased by 3 in action row 3.

If the employee has at least 15 but less than 30 years of service, 2 extra days are given. These 2 days are also provided for employees of age 45 or more. The 2 extra days cannot be combined with the 5 extra days.

Because the 2 days cannot be combined with the 5 extra days we have a false (N) response to condition rows 1-3. Condition rows 4 and 5 test on service of 15 or more years and age more than 45. Where rules 4 and 5 satisfy respectively, Days_leave is increased by 2 (action row 2).

This is a multi-ruling table so that more than one rule can be satisfied. For example, if age is 61 and years of service 31 both rules 2 and 3 are satisfied but the result is the same in that the vacation days are 30.

The ELSE rule covers any rules not stated (and is normally an exception handler) and in this instance covers all the ages and service years that don't receive extra days.

This is how we would write the decision table by using RapidGen Software Decision Table Language – expertGenius RPL.

```
*DETAB Calculate_Days_leave
I Days_leave MV  22                    [Everybody gets 22 vacation days
C AGE       <  18       Y N N N N ELSE
  AGE       >= 60       N Y - N N -
  SERVICE   >= 30       N - Y N N -
  AGE       >= 45       N - - Y N -
  SERVICE   >= 15       N - - - Y -
A Days_leave + 5        X X X . . .
  Days_leave + 3        . X X . . .
  Days_leave + 2        . . . X X .
```

expertGenius RPL Decision Table functionality is a full programming language. It is capable of many things including a complete "Code, Compile, Execute and Go" modelling facility which makes outcomes fast and clear.

The full program with field definitions, inputs and output (display) and of course comments (following [ ) is listed below:

```
*DIC
 Days_leave %I2                    [Define an integer field
 AGE        %I2         !15-80     [set age limits 15-80
 SERVICE    %I2         !0-65      [set service limits
 Name       %V32                   [Define character string

*DETAB Calculate_Days_leave
I Days_leave     MV    22          [Everybody gets 22 vacation days
  INPUTP         Name;             [In this example we input employee's
  INPUTP         AGE;              [name, AGE
  INPUTP         SERVICE;          [and years of SERVICE

C AGE       <     18    Y N N N N ELSE [Is age less than 18
  AGE       >=    60    N Y - N N -   [Is age greater or equal to 60
  SERVICE   >=    30    N - Y N N -   [Is service years greater or equal to 30
  AGE       >=    45    N - - Y N -   [Is age greater or equal to 45
  SERVICE   >=    15    N - - - Y -   [Is service years greater or equal to 15

A Days_leave +    5     X X X . . .       [Add 5 to vacation days
  Days_leave +    3     . X X . . .       [Add 3 to vacation days
  Days_leave +    2     . . . X X .       [Add 2 to vacation days

  DISPLAY Name,' ',AGE,' ',SERVICE,' Vacation ',Days_leave  [show the result
  DISPLAY 'Rule ?'     1 2 3 4 5 E                          [show the rules

*GO
```

An alternative Decision Table where we have set the bounds of the age bands is as follows:

Initial Action			RULES						
Days_leave MV 22									
Condition	1	2	3	4	5	6	7	8	ELSE
AGE >= ?	-	18	18	18	45	45	45	60	ELSE
AGE < ?	18	45	45	45	60	60	60	-	-
SERVICE >= 15	-	N	Y	-	N	Y	-	-	-
SERVICE >= 30	-	N	N	Y	N	N	Y	-	-
Action									
Days_leave + 5	X	.	.	X	.	.	X	X	.
Days_leave + 2	.	.	X	.	X	X	.	.	.
Days_leave + 3	.	.	.	X	.	.	X	X	.

I will not repeat all the information above but note that rule 2 covers all those who get the standard 22 days and no extra days. Rule 2 could be included in the ELSE rule but by stating rule 2 separately this means the ELSE rule can act as an exception handler, although in this instance there is no action given.

Only employees younger than 18 or at least 60 years, or employees with at least 30 years of service will receive 5 extra days.
Condition rows 1-4 test on age and service. Where rules 1, 4, 7 and 8 satisfy, Days_leave is increased by 5 in the first action row. Ignoring the under 18s when rules 4, 7 and 8 satisfy then Days_leave is increased by 3 in action row 3.

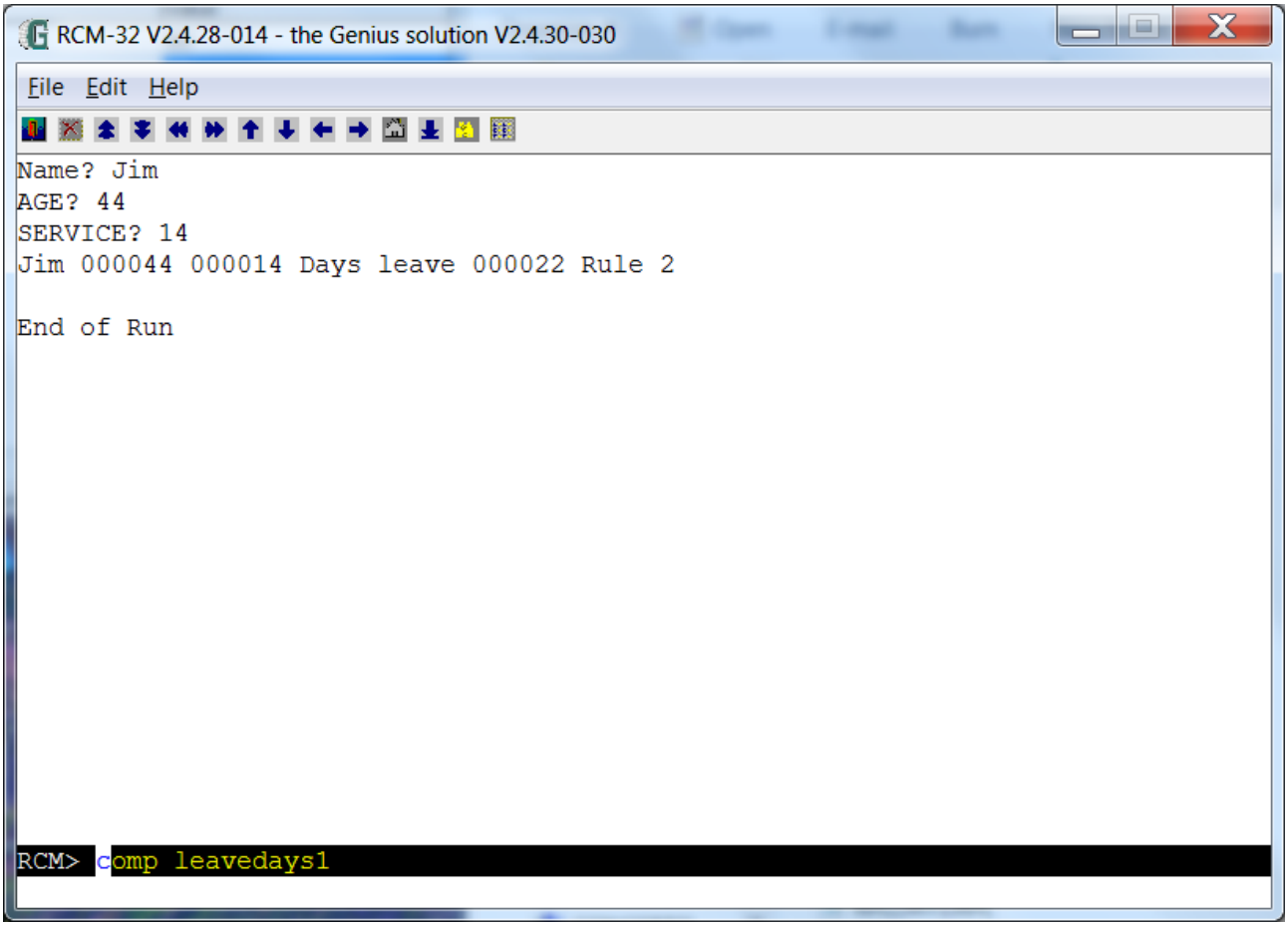
If the employee has at least 15 but less than 30 years of service, 2 extra days are given. These 2 days are also provided for employees of age 45 or more. The 2 extra days cannot be combined with the 5 extra days.
Condition rows 1-4 test on age and service. Where rules 3, 5 and 6 satisfy, Days_leave is increased by 2 in action row 2

It is unlikely that rule 4 will be satisfied as the age of the employee would have to be 14 or less when they joined.

This is how we would write the decision table by using Rapidgen Software Decision Table Language – expertGenius RPL.

```
*DETAB Calculate_Days_leave
I Days_leave      MV    22          [Everybody gets 22 days
C AGE       >=    ?           -  18 18 18 45 45 45 60 ELSE
  AGE       <     ?          18 45 45 45 60 60 60 -   -
  SERVICE   >=    15          -  N  Y  Y  N  Y  Y  -   -
  SERVICE   >=    30          -  N  N  Y  N  N  Y  -   -
A Days_leave +    5           X  .  .  X  .  .  X  X   .   [+5 extra days
  Days_leave +    2           .  .  X  .  X  X  .  .   .   [+2 extra days
  Days_leave +    3           .  .  .  X  .  .  X  X   .   [+3 extra days
```

Here we have a screen shot of the program running with inputs and outputs shown:



While the above Decision Table is acceptable there probably should be further checks on age and service, as it assumes data input is valid. The decision table does not handle invalid data. For example, age could be 20 and service 30 which although invalid would satisfy rule 4. An age of 15 or even 13 may or may not be acceptable so checks for minimum and even maximum ages could be made. Also, that an age of a person is greater than their years of service or alternatively take the minimum age from their real age and check that the result is more than their years of service.

Of course if you know your data is valid then there should be no problem but unfortunately data can become corrupted.

The following Decision Table includes the condition to check that the years of service cannot be more than they should be for the person's age.

```
*DETAB Calculate_Days_leave
I Days_leave      MV    22          [Everybody gets 22 days
  Minimum_age     MV    15          [Set minimum age
  Service_age     MV    AGE         [Set Service age to be age
  Service_age     -     Minimum_age [less the minimum age for comparison
C SERVICE   <=    Service_age Y  Y  Y  Y  Y  Y  Y  Y  ELSE
  AGE       >=    ?           -  18 18 18 45 45 45 60  -
  AGE       <     ?           18 45 45 45 60 60 60 -   -
  SERVICE   >=    15          -  N  Y  Y  N  Y  Y  -   -
  SERVICE   >=    30          -  N  N  Y  N  N  Y  -   -
A Days_leave +=   5           X  .  .  X  .  .  X  X   .   [+5 extra days
  Days_leave +=   2           .  .  X  .  X  X  .  .   .   [+2 extra days
  Days_leave +=   3           .  .  .  X  .  .  X  X   .   [+3 extra days
  DISPLAY         'Error'     .  .  .  .  .  .  .  .   X   [show error
```

If data is stored in a different format then AGE and SERVICE should be evaluated. In the following example these values are calculated from date of birth and date joining. However, years of service evaluation could be affected by taking a sabbatical, unpaid leave, maternity or even leaving and re-joining the company which have not been covered here.

```
*DETAB Calculate_Days_leave
I Days_leave      MV          22    [Everybody gets 22 days
  Chardate1       MV_DAT      'Today' [setup todays date in CCYYMMDD form
  INPUTP    Date.joined;            [In this example we input employee's
  CALL      Evaluate_service_years [joining date and date of birth then
  INPUTP    Date.of.birth;          [we call routines to calculate
  CALL      Evaluate_age            [their years of service and their age
  Minimum_age     MV    15          [Set minimum age
  Service_age     MV    AGE         [Set Service age to be age
  Service_age     -     Minimum_age [less the minimum age for comparison

C SERVICE   <=    Service_age Y  Y  Y  Y  Y  Y  Y  Y  ELSE
  AGE       >=    ?           -  18 18 18 45 45 45 60  -
  AGE       <     ?           18 45 45 45 60 60 60 -   -
  SERVICE   >=    15          -  N  Y  Y  N  Y  Y  -   -
  SERVICE   >=    30          -  N  N  Y  N  N  Y  -   -

A Days_leave +    5           X  .  .  X  .  .  X  X   .   [+5 extra days
  Days_leave +    2           .  .  X  .  X  X  .  .   .   [+2 extra days
  Days_leave +    3           .  .  .  X  .  .  X  X   .   [+3 extra days


*DETAB Evaluate_service_years
I Chardate2       MV_DAT Date.joined     [Date.joined in format CCYYMMDD
  SERVICE         MV     Chardate1:N4    [Take current year and subtract
  SERVICE         -      Chardate2:N4    [joining year to give SERVICE
C Chardate1+4:N4  <      Chardate2+4:N4  Y ELSE [adjust SERVICE by 1 year
A SERVICE         -      1               X -    [if joining MMDD > today
```

```
*DETAB Evaluate_age
I Chardate2       MV_DAT Date.of.birth    [Date.of.birth in CCYYMMDD form
  AGE             MV     Chardate1:N4     [Take current year and subtract
  AGE             -      Chardate2:N4     [birth year to give AGE
C Chardate1+4:N4  <      Chardate2+4:N4   Y ELSE [adjust AGE by 1 year if
A AGE             -      1                X -    [birthday not happened yet
```

An example of our Callable Decision Table Program where Decision Tables can be embedded and tested in C or C++ programs.