

Challenge March 2023

Permit Eligibility

A solution with DT5GL by Jack Jansonius – 2 May 2023

Problem Statement (from the web site):

As part of a regulatory process, a government agency wants to determine if an applicant is eligible for a resident permit using a simple rule: an applicant is eligible for a resident permit if the applicant has lived at an address while married and in that time period, they have shared the same address at least 7 of the last 10 years. This problem was proposed by [Trisotech](#).

Here is an example of input data:

A list of periods living at an address for applicant (From, To, Address):

2010-01-01,2015-12-31,"123 Main St, Anytown, USA"

2016-01-01,2020-12-31,"456 Oak St, Anytown, USA"

2021-01-01,2023-03-04,"789 Elm St, Anytown, USA"

A list of periods living at an address for spouse (From, To, Address):

2010-01-01,2015-12-31,"123 Main St, Anytown, USA"

2016-01-01,2020-12-31,"120 Maple St, Anytown, USA"

2021-01-01,2023-03-04,"789 Elm St, Anytown, USA"

A list of applicant and spouse marriage periods (From, To):

2010-01-01,2015-12-31

2021-01-01,2023-03-04

Tables in the database:

id	name	type
1	Applicant1	A
2	Spouse1	NULL
3	Applicant2	A
4	Spouse2	NULL
5	Applicant3	A
6	Spouse3	NULL
7	Applicant4	A
8	Spouse4	NULL
9	Applicant5	A
10	Spouse5.1	NULL
11	Spouse5.2	NULL

- person -

id	address
1	123 Main St, Anytown, USA
2	456 Oak St, Anytown, USA
3	789 Elm St, Anytown, USA
4	120 Maple St, Anytown, USA

- address -

personid	addressid	date from	date to
1	1	2010-01-01	2015-12-31
1	2	2016-01-01	2020-12-31
1	3	2021-01-01	NULL
2	1	2010-01-01	2015-12-31
2	4	2016-01-01	2020-12-31
2	3	2021-01-01	NULL
3	1	2010-01-01	2015-05-31
3	2	2016-01-01	2020-12-31
3	3	2021-01-01	NULL
4	1	2010-01-01	2015-12-31
4	4	2016-01-01	2020-12-31
4	3	2022-06-01	NULL
5	1	2010-01-01	2015-12-31
5	2	2016-04-08	2020-12-31
5	3	2021-01-01	2022-08-18
5	1	2023-02-12	NULL
6	1	2010-01-01	2015-12-31
6	4	2016-02-12	2020-12-31
6	3	2021-01-01	2022-08-18
6	1	2023-02-12	NULL
7	1	2010-01-01	2015-12-31
7	2	2016-01-01	2017-12-31
7	3	2018-01-01	2023-01-31
8	1	2010-01-01	2015-12-31
8	4	2016-01-01	2017-12-31
8	3	2018-01-01	2022-11-30
9	1	2015-01-01	2015-01-31
9	2	2016-01-01	2022-12-31
10	1	2015-01-01	2015-01-31
11	2	2016-01-01	2022-12-31

- person-address -

applicant	spous	date from	date to
1	2	2010-01-01	2015-12-31
1	2	2021-01-01	NULL
3	4	2010-01-01	2015-12-31
3	4	2022-01-01	NULL
5	6	2010-01-01	NULL
7	8	2010-01-01	2015-12-31
7	8	2018-01-01	2023-03-04
9	10	2010-01-01	2015-01-31
9	11	2016-02-01	2022-12-31

- married -

Implementation of the decision tables in DT5GL:

SQLite_database: "Database/Applicant.sqlite"

An applicant is eligible for a resident permit if the applicant has lived at an address while married and in that time period, they have shared the same address at least 7 of the last 10 years. But here it's not years or days but months that are counted: 7 of the last 10 years = 84 of the last 120 months

YearMonthEndCheck = 202303,
so reference months are from March 2013 to February 2023 (= exactly 10 years).
Reference day = the 15th of the month.

Extra: a listing of the months in which the condition is met, for example:
Periods married and same address: [03/2013-12/2015], [01/2021-02/2023]

Attribute: YearMonthEndCheck Type: Integer
Equals: 202303
Attribute: NumberYearsInspected Type: Integer
Equals: 10

Table 0:

If:	0 1
'Next applicant present'	Y N
Then:	
NextApplicant is Selected	X
NextApplicant is NotSelected	X
#	
# Repeat until: NotSelected	

Proposition: 'Next applicant present'
Obtain_instance_from_database_view: applicant

Table 1:

If:	0 1 2
Next year in [firstYear-lastYear]	Y N N
month_ok >= 84	- Y N
Then:	
EvalYear is Selected	X
EvalYear is Finished_ok	X
EvalYear is Finished_nok	X
#	
# Repeat until: Finished_ok, Finished_nok	

Table 2:

If:	0 1 2 3
Next month in [firstMonth-lastMonth]	Y Y Y N
'Applicant is married on reference date'	Y Y N - ¹
applicant_address.id = spouse_address.id	Y N - -
Then:	
EvalMonth is Eligible	X
EvalMonth is NotEligible_1	X
EvalMonth is NotEligible_2	X
EvalMonth is Finished	X
#	
# Repeat until: Finished	

Proposition: 'Applicant is married on reference date'
Obtain_instance_from_database_view: married

¹ One reference date per month, namely the 15th, is somewhat rough, since the applicant must meet the condition for at least 14 days of a month for the month to count toward the final score of 84 months. With an additional condition in this decision table, it is easy to insert a second reference date per month so that, for example, the 5th and 25th of the month are polled.

```

# Determine range of years: [firstYear-lastYear]
Attribute: lastYear          Type: Integer
Equals: int(YearMonthEndCheck/100)

Attribute: firstYear         Type: Integer
Equals: lastYear - NumberYearsInspected

# Determine range of months within selected year: [firstMonth-lastMonth]
Attribute: checkMonth        Type: Integer
Equals: YearMonthEndCheck % 100

Attribute: firstMonth        Type: Integer2
Equals: checkMonth if year == firstYear else 1

Attribute: lastMonth         Type: Integer
Equals: checkMonth - 1 if year == lastYear else 12

Attribute: refdate           Type: Text
Equals: str(year) + "-" + zerofill(month,2) + "-15"

Attribute: applicant_address.id   Type: Integer
Attribute: spouse_address.id      Type: Integer

Attribute: current_month_ok_string Type: Text
Equals: zerofill(month,2) + "/" + str(year)
# zerofill(4,2) = "04"

```

rTable 3: specify ok_periodinfo\1

```

If:
year = lastYear          | 0 | 1 |
month = lastMonth        | Y | Y |
nr_current_month_ok = 0  | Y | N |
Then:
ok_periodinfo = ", [" + current_month_ok_string + "]" | X | |
ok_periodinfo = "-" + current_month_ok_string + "]" | | X |
# .....

```

Table 4: specify ok_periodinfo\2

```

If:
nr_current_month_ok = 0  | 0 | 1 | 2 |
period_textstring = ""  | Y | Y | N |
Then:
ok_periodinfo = "[" + current_month_ok_string | X | | |
ok_periodinfo = ", [" + current_month_ok_string | | X | |
ok_periodinfo = ""      | | | X |
# .....

```

Table 5: specify nok_periodinfo

```

If:
nr_current_month_ok = 0  | 0 | 1 | 2 |
nr_current_month_ok = 1  | Y | N | N |
Then:
nok_periodinfo = ""      | X | | |
nok_periodinfo = "]"     | | X | |
nok_periodinfo = "-" + last_month_ok_string + "]" | | | X |
# .....

```

² With this definition of firstMonth and lastMonth, the months to be checked for the past 10 years range from March 2013 to February 2023 if the variable YearMonthEndCheck is set to 202303. That check period can be moved up a month with firstMonth = checkMonth+1 if year == firstYear else 1 and lastMonth = checkMonth if year == lastYear else 12.

Database views

Database_view: applicant

With_attributes: id, name

Query:

```
SELECT id, name
FROM person
WHERE type = "A"
LIMIT 1 OFFSET %s
```

With_arguments: applicant.auto_index

Database_view: married

With_attributes: applicant_id, spouse_id, from, to

Query:

```
SELECT applicant_id, spouse_id, date_from, date_to
FROM married
WHERE applicant_id = %s
AND date_from <= '%s'
AND (date_to IS NULL OR date_to >= '%s' )3
```

With_arguments: applicant.id, refdate, refdate

Database_view: applicant_address⁴

With_attributes: id

Query:

```
SELECT
COALESCE((SELECT addressid
FROM person_address
WHERE personid = %s
AND date_from <= '%s'
AND (date_to IS NULL OR date_to >= '%s' )),
-1) AS addressid
```

With_arguments: applicant.id, refdate, refdate

Retrieves the addressid for the applicant on the reference day

and returns -1 if no address is found.

Database_view: spouse_address

With_attributes: id

Query:

```
SELECT
COALESCE((SELECT addressid
FROM person_address
WHERE personid = %s
AND date_from <= '%s'
AND (date_to IS NULL OR date_to >= '%s' )),
-2) AS addressid
```

With_arguments: married.spouse_id, refdate, refdate

Retrieves the addressid for the spouse on the reference day

and returns -2 if no address is found.

³ Of course, the end date of a marriage period or residential address does not necessarily have to be filled!

⁴ SQL query as suggested by ChatGPT. Coalesce(x,y) = y if x='Null' else x.

GoalAttributes

GoalAttribute: NextApplicant
Repeat_until: NotSelected

Case: NotSelected
Print: "End!"

Case: Selected
Print: "#REM# - "
>>: month_ok = 0 # Counter months ok, because married and living at
the same address.
>>: nok_married = 0 # Counter months not ok because not married.
>>: nok_address = 0 # Counter months not ok because married but not
living at the same address.

>>: nr_current_month_ok = 0 # Counter months ok within a subperiod of
consecutive ok months.
>>: period_textstring = "" # The text string in which the periods are
collected.
>>: last_month_ok_string = "" # Remember last ok month in format "mm/yyyy";
empty now for next applicant.

GoalAttribute: EvalYear
Repeat_until: Finished_ok, Finished_nok

Case: Finished_ok
Print: "Applicant %s (%s) is ELIGIBLE for a resident permit." applicant.name
applicant.id
Print: "Number of months married and same address.....: %s " month_ok
Print: " (= > sufficient for minimum of 84 months)"
Print: "Number of months married; not the same address: %s" nok_address
Print: "Number of months not married.....: %s" nok_married
Print: "Periods married and same address: %s" period_textstring
Print: "-----"

Case: Finished_nok
Print: "Applicant %s (%s) is NOT eligible for a resident permit." applicant.name
applicant.id
Print: "Number of months married and same address.....: %s " month_ok
Print: " (= > not sufficient for minimum of 84 months)"
Print: "Number of months married; not the same address: %s" nok_address
Print: "Number of months not married.....: %s" nok_married
Print: "Periods married and same address: %s" period_textstring
Print: "-----"

Case: Selected
Print: "#REM# - "

GoalAttribute: EvalMonth
Repeat_until: Finished

Case: Finished
Print: "#REM# - "

Case: Eligible
>>: month_ok = month_ok + 1
>>: period_textstring = period_textstring + ok_periodinfo
>>: nr_current_month_ok = nr_current_month_ok + 1
>>: last_month_ok_string = current_month_ok_string

Case: NotEligible_1
>>: nok_address = nok_address + 1
>>: period_textstring = period_textstring + nok_periodinfo
>>: nr_current_month_ok = 0

Case: NotEligible_2
>>: nok_married = nok_married + 1
>>: period_textstring = period_textstring + nok_periodinfo
>>: nr_ok = 0

Testrun solution without Extra:

Applicant Applicant1 (1) is NOT eligible for a resident permit.
Number of months married and same address.....: 60
 (=> not sufficient for minimum of 84 months)
Number of months married; not the same address: 0
Number of months not married.....: 60

Applicant Applicant2 (3) is NOT eligible for a resident permit.
Number of months married and same address.....: 36
 (=> not sufficient for minimum of 84 months)
Number of months married; not the same address: 12
Number of months not married.....: 72

Applicant Applicant3 (5) is NOT eligible for a resident permit.
Number of months married and same address.....: 55
 (=> not sufficient for minimum of 84 months)
Number of months married; not the same address: 65
Number of months not married.....: 0

Applicant Applicant4 (7) is ELIGIBLE for a resident permit.
Number of months married and same address.....: 93
 (=> sufficient for minimum of 84 months)
Number of months married; not the same address: 3
Number of months not married.....: 24

Applicant Applicant5 (9) is ELIGIBLE for a resident permit.
Number of months married and same address.....: 84
 (=> sufficient for minimum of 84 months)
Number of months married; not the same address: 22
Number of months not married.....: 14

End!
Time elapsed: 0:00:02.825050

Testrun solution with Extra:

Applicant Applicant1 (1) is NOT eligible for a resident permit.

Number of months married and same address.....: 60

(=> not sufficient for minimum of 84 months)

Number of months married; not the same address: 0

Number of months not married.....: 60

Periods married and same address: [03/2013-12/2015], [01/2021-02/2023]

Applicant Applicant2 (3) is NOT eligible for a resident permit.

Number of months married and same address.....: 36

(=> not sufficient for minimum of 84 months)

Number of months married; not the same address: 12

Number of months not married.....: 72

Periods married and same address: [03/2013-05/2015], [06/2022-02/2023]

Applicant Applicant3 (5) is NOT eligible for a resident permit.

Number of months married and same address.....: 55

(=> not sufficient for minimum of 84 months)

Number of months married; not the same address: 65

Number of months not married.....: 0

Periods married and same address: [03/2013-12/2015], [01/2021-08/2022], [02/2023]

Applicant Applicant4 (7) is ELIGIBLE for a resident permit.

Number of months married and same address.....: 93

(=> sufficient for minimum of 84 months)

Number of months married; not the same address: 3

Number of months not married.....: 24

Periods married and same address: [03/2013-12/2015], [01/2018-11/2022]

Applicant Applicant5 (9) is ELIGIBLE for a resident permit.

Number of months married and same address.....: 84

(=> sufficient for minimum of 84 months)

Number of months married; not the same address: 22

Number of months not married.....: 14

Periods married and same address: [01/2015], [02/2016-12/2022]

End!

Time elapsed: 0:00:02.074774

Details test cases

Testcase 1 (website):

person:

id	name	type
1	Applicant1	A
2	Spouse1	NULL

address:

id	address
1	123 Main St, Anytown, USA
2	456 Oak St, Anytown, USA
3	789 Elm St, Anytown, USA
4	120 Maple St, Anytown, USA

person-address:

personid	addressid	date from	date to
1	1	2010-01-01	2015-12-31
1	2	2016-01-01	2020-12-31
1	3	2021-01-01	NULL
2	1	2010-01-01	2015-12-31
2	4	2016-01-01	2020-12-31
2	3	2021-01-01	NULL

married:

applicant	spous	date from	date to
1	2	2010-01-01	2015-12-31
1	2	2021-01-01	NULL

Result:

Applicant Applicant1 (1) is NOT eligible for a resident permit.

Number of months married and same address.....: 60

(=> not sufficient for minimum of 84 months)

Number of months married; not the same address: 0

Number of months not married.....: 60

Periods married and same address: [03/2013-12/2015], [01/2021-02/2023]

Testcase 2:

person:

id	name	type
3	Applicant2	A
4	Spouse2	NULL

person-address:

personid	addressid	date from	date to
3	1	2010-01-01	2015-05-31
3	2	2016-01-01	2020-12-31
3	3	2021-01-01	NULL
4	1	2010-01-01	2015-12-31
4	4	2016-01-01	2020-12-31
4	3	2022-06-01	NULL

married:

applicant	spous	date from	date to
3	4	2010-01-01	2015-12-31
3	4	2022-01-01	NULL

Result:

Applicant Applicant2 (3) is NOT eligible for a resident permit.

Number of months married and same address.....: 36

(=> not sufficient for minimum of 84 months)

Number of months married; not the same address: 12

Number of months not married.....: 72

Periods married and same address: [03/2013-05/2015], [06/2022-02/2023]

Testcase 3:

person:

id	name	type
5	Applicant3	A
6	Spouse3	NULL

address:

id	address
1	123 Main St, Anytown, USA
2	456 Oak St, Anytown, USA
3	789 Elm St, Anytown, USA
4	120 Maple St, Anytown, USA

person-address:

personid	addressid	date from	date to
5	1	2010-01-01	2015-12-31
5	2	2016-04-08	2020-12-31
5	3	2021-01-01	2022-08-18
5	1	2023-02-12	NULL
6	1	2010-01-01	2015-12-31
6	4	2016-02-12	2020-12-31
6	3	2021-01-01	2022-08-18
6	1	2023-02-12	NULL

married:

applicant	spous	date from	date to
5	6	2010-01-01	NULL

Result:

Applicant Applicant3 (5) is NOT eligible for a resident permit.

Number of months married and same address.....: 55

(=> not sufficient for minimum of 84 months)

Number of months married; not the same address: 65

Number of months not married.....: 0

Periods married and same address: [03/2013-12/2015], [01/2021-08/2022], [02/2023]

Testcase 4:

person:

id	name	type
7	Applicant4	A
8	Spouse4	NULL

person-address:

personid	addressid	date from	date to
7	1	2010-01-01	2015-12-31
7	2	2016-01-01	2017-12-31
7	3	2018-01-01	2023-01-31
8	1	2010-01-01	2015-12-31
8	4	2016-01-01	2017-12-31
8	3	2018-01-01	2022-11-30

married:

applicant	spous	date from	date to
7	8	2010-01-01	2015-12-31
7	8	2018-01-01	2023-03-04

Result:

Applicant Applicant4 (7) is ELIGIBLE for a resident permit.

Number of months married and same address.....: 93

(=> sufficient for minimum of 84 months)

Number of months married; not the same address: 3

Number of months not married.....: 24

Periods married and same address: [03/2013-12/2015], [01/2018-11/2022]

Testcase 5:

person:

id	name	type
9	Applicant5	A
10	Spouse5.1	NULL
11	Spouse5.2	NULL

address:

id	address
1	123 Main St, Anytown, USA
2	456 Oak St, Anytown, USA
3	789 Elm St, Anytown, USA
4	120 Maple St, Anytown, USA

person-address:

personid	addressid	date from	date to
9	1	2015-01-01	2015-01-31
9	2	2016-01-01	2022-12-31
10	1	2015-01-01	2015-01-31
11	2	2016-01-01	2022-12-31

married:

applicant	spous	date from	date to
9	10	2010-01-01	2015-01-31
9	11	2016-02-01	2022-12-31

Result:

Applicant Applicant5 (9) is ELIGIBLE for a resident permit.
Number of months married and same address.....: 84
(=> sufficient for minimum of 84 months)
Number of months married; not the same address: 22
Number of months not married.....: 14
Periods married and same address: [01/2015], [02/2016-12/2022]

What's new in DT5GL?

In a first design of a solution to this challenge, I used variables in a temporary table of the database being used. This construction looked like this:

```
Initial_database_table: init_results
Query:
    CREATE TEMP TABLE results AS
    SELECT 0 AS month_ok,
           0 AS nok_married,
           0 AS nok_address
End_Query

Database_view: results
With_attributes: month_ok, nok_married, nok_address
Query:
SELECT *
    FROM results
End_Query
```

Thus, they are the variables that track how many months meet the condition "married and living at the same address," namely month_ok and how many months do not meet it, due to "not married" (nok_married) and months that do not meet it, due to "married and not living at the same address (nok_address).

The goal attribute eval_month then increments the variables in question with simple sql statements:

```
Case: Eligible
>SQL: "UPDATE results "
<SQL: "    SET month_ok = month_ok + 1 "
```

```
Case: NotEligible_1
>SQL: "UPDATE results "
<SQL: "    SET nok_address = nok_address + 1 "
```

```
Case: NotEligible_2
>SQL: "UPDATE results "
<SQL: "    SET nok_married = nok_married + 1 "
```

Through the database view 'results' mentioned above, the variables are always retrievable, for example as a condition in decision table 1: results.month_ok >= 84.

Until recently, this approach worked fine, but now that DT5GL is being integrated into the Dutch Datawarehouse tool Grip op Data (<https://www.gripopdata.nl/>) and is being used to label and enrich tens of millions of records for millions of customers at a government organization, performance is an issue, so I was asked to replace these external variables - for once - with internal variables.

Although parallel processing of this one-time labeling process⁵ remains necessary to keep the entire throughput time under control, the 25% performance gain achieved was well worth it!

The combination of decision tables and SQL realized with DT5GL appears to work very well for organizations that get stuck in complexity when working with SQL alone, or supplement SQL with all kinds of 3GL constructs.⁶

⁵ Stated process is performed with PostgreSQL and Oracle.

⁶ Conform: <https://dmcommunity.org/2021/09/02/is-sql-for-business-or-it/>

Replacing external variables in the database with internal variables does look very simple:

Case: Eligible

```
>>: month_ok = month_ok + 1
```

Case: NotEligible_1

```
>>: nok_address = nok_address + 1
```

Case: NotEligible_2

```
>>: nok_married = nok_married + 1
```

And the condition in decision table 1 is now even simpler: $\text{month_ok} \geq 84$