Challenge Augustus 2023
Soldier Payment Rules

A solution with DT5GL/SQL by Jack Jansonius – 21 September 2023

Problem Statement (from the web site):

During different service periods a soldier may have the following characteristics:
Rank {private, corporal, sergeant, lieutenant, captain}
Profession {fighter, driver, cook, officer}
Service Type {active, reserve, retired}
Unit {HQ, paratroopers, marines, infantry}
Combat {yes, no}

Pay rate is determined by aggregating the amounts according to these rules:
Base rate is $1/hr.
Private $1/hr., corporal $2/hr., sergeant $3/hr., lieutenant $4/hr., captain $5/hr.
Fighter $2/hr., driver $1/hr., cook $1/hr., officer $3/hr.
Active $2/hr., Reserve $1/hr.
HQ $1/hr., others $2/hr.
Combat $5/hr., non-combat $0/hr.

The challenge: assemble a single timeline for the soldier over a given service period that shows his/her hourly pay rate in any given time. Flag any conflicting dates (e.g. can't be a sergeant and a lieutenant at the same time).

Additional challenge: What are all the different aggregated pay rates that apply and during which periods?

Comments on this solution:
Completely along the pattern of my solution for the March-2023 Permit Eligibility - challenge. So no difficult analysis of time periods with an equal sum of the characteristics, but going month by month through a desired period and displaying only those months in which the sum of the characteristics changes from a previous period.

Initially, overlaps and gaps in the characteristics were not taken into account, but a solution that does so did not take much effort; the extra "code" for this is indicated by dark blue in a second solution.

Tables in the database:

- soldier -

id	name	yearmonth_start	yearmonth_end
1	Not Retired	201501	201812
2	Retired	201501	201512
3	Regretting Retirement	201502	201712
4	Gaps and Overlaps	201501	201812

- service contract -

soldierid	type	date_from	date_to
1	1	2015-01-01	2015-06-30
1	2	2015-07-01	2015-11-30
1	1	2015-12-01	NULL
2	3	2015-01-01	NULL
3	1	2015-01-01	2015-06-30
3	3	2015-07-01	2015-11-30
3	2	2015-12-01	NULL
4	1	2015-01-01	2015-06-30
4	2	2015-07-01	2015-12-31
4	1	2015-12-01	NULL

- service -

id	description
1	active
2	reserve
3	retired

- service rate -

service_id	rate	date_from	date_to
1	2	2015-01-01	2016-06-30
1	2.5	2016-07-01	NULL
2	1	2015-01-01	2016-06-30
2	1.5	2016-07-01	NULL
3	0	2015-01-01	NULL

- base contract -

soldierid	type	date_from	date_to
1	1	2015-01-01	NULL
3	1	2015-01-01	NULL
4	1	2015-01-01	NULL

- base -

id	description
1	base

- base rate -

base_id	rate	date_from	date_to
1	1	2015-01-01	2016-06-30
1	1.25	2016-07-01	NULL

- rank contract -

soldierid	type	date_from	date_to
1	1	2015-01-01	2015-12-31
1	2	2016-01-01	2016-12-31
1	3	2017-01-01	NULL
3	5	2015-01-01	2015-12-31
3	4	2016-01-01	2016-12-31
3	3	2017-01-01	NULL
4	1	2015-01-01	2015-11-30
4	2	2016-02-01	2017-01-31
4	3	2017-01-01	NULL

- rank -

id	description
1	private
2	corporal
3	sergeant
4	lieutenant
5	captain

- rank rate -

rankid	rate	date_from	date_to
1	1	2015-01-01	2016-06-30
1	1.5	2016-07-01	NULL
2	2	2015-01-01	2016-06-30
2	2.5	2016-07-01	NULL
3	3	2015-01-01	2016-06-30
3	3.5	2016-07-01	NULL
4	4	2015-01-01	2016-06-30
4	4.5	2016-07-01	NULL
5	5	2015-01-01	2016-06-30
5	5.5	2016-07-01	NULL

- profession contract

soldierid	type	date_from	date_to
1	1	2015-01-01	2015-06-30
1	3	2015-07-01	2015-11-30
1	2	2015-12-01	2016-12-31
1	4	2017-01-01	NULL
3	1	2015-01-01	2015-06-30
3	2	2015-07-01	2015-11-30
3	3	2015-12-01	2016-12-31
3	4	2017-01-01	NULL
4	1	2015-01-01	2015-06-30
4	3	2015-07-01	2016-01-31
4	2	2015-12-01	2016-12-31
4	4	2017-01-01	NULL

- profession -

id	description
1	fighter
2	driver
3	cook
4	officer

- profession rate -

professionid	rate	date_from	date_to
1	2	2015-01-01	2016-06-30
1	2.5	2016-07-01	NULL
2	1	2015-01-01	2016-06-30
2	1.6	2016-07-01	NULL
3	1	2015-01-01	2016-06-30
3	1.5	2016-07-01	NULL
4	3	2015-01-01	2016-06-30
4	4	2016-07-01	NULL

- unit contract

soldierid	type	date_from	date_to
1	1	2015-01-01	2015-12-31
1	1	2016-01-01	NULL
3	2	2015-01-01	2015-12-31
3	1	2016-01-01	NULL
4	1	2015-01-01	2015-11-30
4	1	2016-01-01	NULL

- unit - [1]

id	description
1	HQ
2	others

- unit rate -

unitid	rate	date_from	date_to
1	1	2015-01-01	2016-06-30
1	1.5	2016-07-01	NULL
2	2	2015-01-01	2016-06-30
2	2.5	2016-07-01	NULL

- combat contract

soldierid	type	date_from	date_to
1	2	2015-01-01	2015-03-30
1	1	2015-04-01	2015-06-30
1	2	2015-07-01	NULL
3	1	2015-01-01	2015-03-30
3	2	2015-04-01	2015-06-30
3	1	2015-07-01	NULL
4	2	2015-01-01	2015-03-30
4	1	2015-04-01	2015-06-30
4	2	2015-07-01	NULL

- combat -

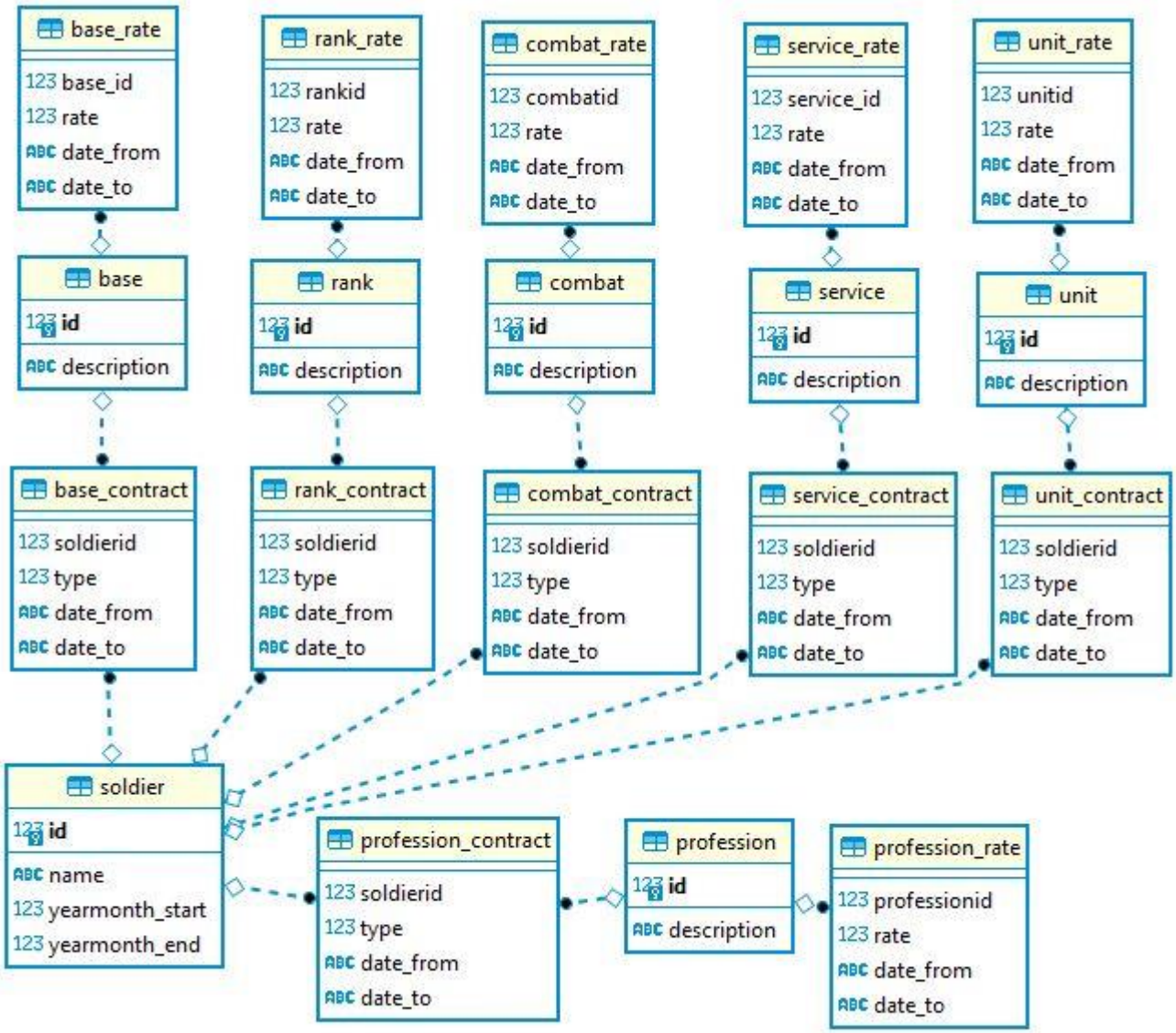
id	description
1	yes
2	no

- combat rate -

combatid	rate	date_from	date_to
1	5	2015-01-01	2016-06-30
1	5.5	2016-07-01	NULL
2	0	2015-01-01	NULL

[1] A little mistake on my part: 'Others' is paratroopers, marines, infantry…

ER-Diagram (DBeaver/SQlite)



Implementation of the decision model in DT5GL (first solution):

```
SQLite_database: "Database/Soldier Payment.db"

# Reference day = the 15th of the month.

Table 0:
If:                                          | 0| 1|
'Next soldier present'                       | Y| N|
Then:
NextSoldier is Selected                      | X|  |
NextSoldier is Finished                      |  | X|
# .......
# Repeat until: Finished

Proposition: 'Next soldier present'
Obtain_instance_from_database_view: soldier

Table 1:
If:                                          | 0| 1|
Next year in [firstYear-lastYear]            | Y| N|
Then:
EvalYear is Selected                         | X|  |
EvalYear is Finished                         |  | X|
# .......
# Repeat until: Finished

Table 2:
If:                                          | 0| 1| 2| 3| 4|
Next month in [firstMonth-lastMonth]         | Y| Y| Y| Y| N|
service.description = "retired"              | Y| Y| N| N| -|
current_pay_rate = 0                         | Y| N| -| -| -|
current_pay_rate = pay_rate_this_month       | -| -| Y| N| -|
Then:
EvalMonth is Skip                            | X|  | X|  |  |
EvalMonth is NoPayRate                       |  | X|  |  |  |
EvalMonth is NewPayRate                      |  |  |  | X|  |
EvalMonth is Finished                        |  |  |  |  | X|
# .......
# Repeat until: Finished


# Range of years to be checked: [firstYear-lastYear]
Attribute: firstYear                 Type: Integer
Equals: int(soldier.yearmonth_start/100)

Attribute: lastYear                  Type: Integer
Equals: int(soldier.yearmonth_end/100)

# Range of months to be checked within selected year: [firstMonth-lastMonth]
Attribute: firstMonth                Type: Integer
Equals: soldier.yearmonth_start % 100 if year == firstYear else 1

Attribute: lastMonth                 Type: Integer
Equals: soldier.yearmonth_end % 100 if year == lastYear  else 12

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

Attribute: fromdate                  Type: Text
Equals: "01/" + zerofill(month,2) + "/" + str(year)

Attribute: soldier.yearmonth_start      Type: Integer
Attribute: soldier.yearmonth_end        Type: Integer
```

```
Attribute: pay_rate_this_month  Type: Real
Equals: base.rate + rank.rate + profession.rate + service.rate + unit.rate +
combat.rate

Attribute: base.rate        Type: Real
Attribute: rank.rate        Type: Real
Attribute: profession.rate  Type: Real
Attribute: service.rate     Type: Real
Attribute: unit.rate        Type: Real
Attribute: combat.rate      Type: Real



########################## Database views ############################

Database_view: soldier
With_attributes: id, name, yearmonth_start, yearmonth_end
Query:
 SELECT id, name, yearmonth_start, yearmonth_end
  FROM soldier
 LIMIT 1 OFFSET %s
With_arguments: soldier.auto_index


Database_view: base
With_attributes: description, rate
Query:
SELECT base.description, base_rate.rate
  FROM base_contract
       JOIN base ON base_contract.type = base.id
       JOIN base_rate ON base.id = base_rate.base_id
 WHERE base_contract.soldierid = %s AND
      '%s' BETWEEN base_contract.date_from
               AND COALESCE(base_contract.date_to,'9999-12-31') AND
      '%s' BETWEEN base_rate.date_from
               AND COALESCE(base_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: rank
With_attributes: description, rate
Query:
SELECT rank.description, rank_rate.rate
  FROM rank_contract
       JOIN rank ON rank_contract.type = rank.id
       JOIN rank_rate ON rank.id = rank_rate.rankid
 WHERE rank_contract.soldierid = %s AND
      '%s' BETWEEN rank_contract.date_from
               AND COALESCE(rank_contract.date_to,'9999-12-31') AND
      '%s' BETWEEN rank_rate.date_from
               AND COALESCE(rank_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: profession
With_attributes: description, rate
Query:
SELECT profession.description, profession_rate.rate
  FROM profession_contract
       JOIN profession ON profession_contract.type = profession.id
       JOIN profession_rate ON profession.id = profession_rate.professionid
 WHERE profession_contract.soldierid = %s AND
       '%s' BETWEEN profession_contract.date_from
                AND COALESCE(profession_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN profession_rate.date_from
                AND COALESCE(profession_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate
```

```
Database_view: service
With_attributes: description, rate
Query:
SELECT service.description, service_rate.rate
  FROM service_contract
       JOIN service ON service_contract.type = service.id
       JOIN service_rate ON service.id = service_rate.service_id
 WHERE service_contract.soldierid = %s AND
       '%s' BETWEEN service_contract.date_from
                AND COALESCE(service_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN service_rate.date_from
                AND COALESCE(service_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: unit
With_attributes: description, rate
Query:
SELECT unit.description, unit_rate.rate
  FROM unit_contract
       JOIN unit ON unit_contract.type = unit.id
       JOIN unit_rate ON unit.id = unit_rate.unitid
 WHERE unit_contract.soldierid = %s AND
       '%s' BETWEEN unit_contract.date_from
                AND COALESCE(unit_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN unit_rate.date_from
                AND COALESCE(unit_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: combat
With_attributes: description, rate
Query:[2]
SELECT combat.description, combat_rate.rate
  FROM combat_contract
       JOIN combat ON combat_contract.type = combat.id
       JOIN combat_rate ON combat.id = combat_rate.combatid
 WHERE combat_contract.soldierid = %s AND
       '%s' BETWEEN combat_contract.date_from
                AND COALESCE(combat_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN combat_rate.date_from
                AND COALESCE(combat_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate
```

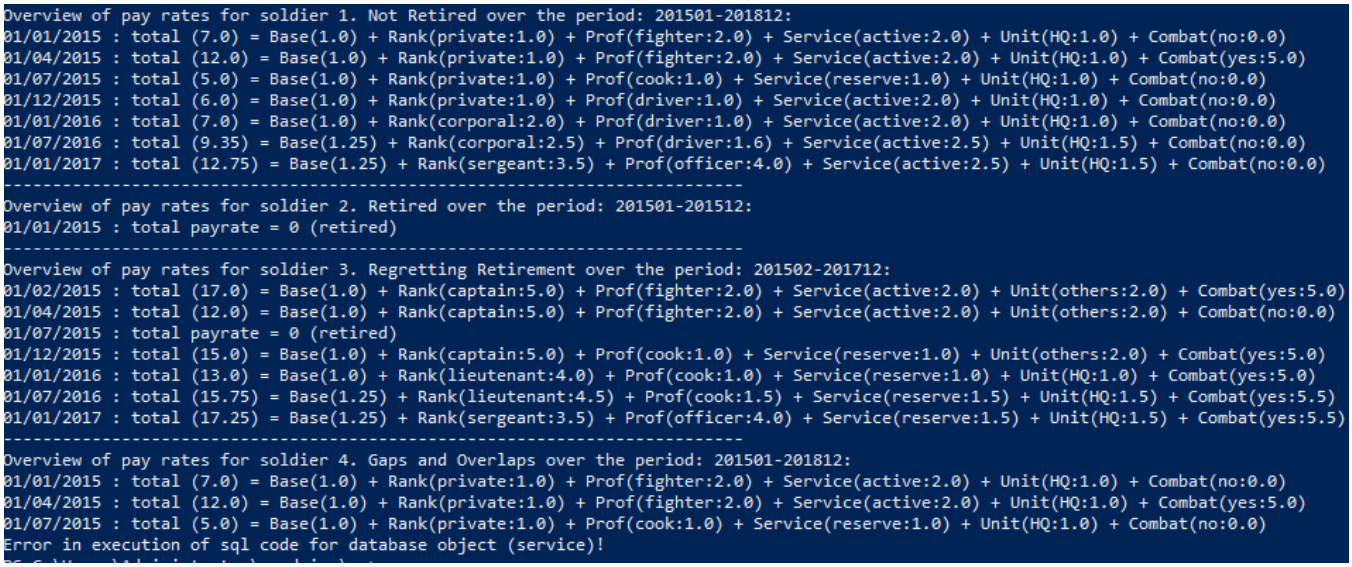
[2] SQL query as suggested by ChatGPT:

```
SELECT R.description AS RankDescription, COALESCE(RT.rate, 0.00) AS Rate
FROM Soldier S
INNER JOIN Contract C ON S.id = C.soldierid
INNER JOIN Rank R ON C.type = R.id
LEFT JOIN Rate RT ON R.id = RT.rankid
WHERE S.id = [SoldierID] -- Replace [SoldierID] with the ID of the soldier for whom you want information
retrieve
AND [Peildatum] BETWEEN C.date_from AND COALESCE(C.date_to, '9999-12-31')
AND [Peildatum] BETWEEN RT.date_from AND COALESCE(RT.date_to, '9999-12-31');
```

```
########################## GoalAttributes ############################

GoalAttribute: NextSoldier
Repeat_until:  Finished

Case: Finished
Print: "End!"

Case: Selected
Print: "Overview of pay rates for soldier %s. %s over the period: %s-%s:"
soldier.id soldier.name soldier.yearmonth_start soldier.yearmonth_end
>>: current_pay_rate = -1.00


GoalAttribute: EvalYear
Repeat_until:  Finished

Case: Finished
Print: "--------------------------------------------------------------------------
"

Case: Selected
Print: "#REM# -  "


GoalAttribute: EvalMonth
Repeat_until: Finished

Case: Finished
Print: "#REM# -  "

Case: Skip
Print: "#REM# -  "

Case: NoPayRate
Print: "%s : total payrate = 0 (retired) "  fromdate
>>: current_pay_rate = 0

Case: NewPayRate
Print: "%s : total (%s) = Base(%s) + Rank(%s:%s) + Prof(%s:%s) + Service(%s:%s) +
Unit(%s:%s) + Combat(%s:%s)" fromdate pay_rate_this_month base.rate
rank.description rank.rate profession.description profession.rate
service.description service.rate unit.description unit.rate combat.description
combat.rate
>>: current_pay_rate = pay_rate_this_month
```

Testrun first solution:

```
Overview of pay rates for soldier 1. Not Retired over the period: 201501-201812:
01/01/2015 : total (7.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2015 : total (5.0) = Base(1.0) + Rank(private:1.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/12/2015 : total (6.0) = Base(1.0) + Rank(private:1.0) + Prof(driver:1.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/01/2016 : total (7.0) = Base(1.0) + Rank(corporal:2.0) + Prof(driver:1.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/07/2016 : total (9.35) = Base(1.25) + Rank(corporal:2.5) + Prof(driver:1.6) +
Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
01/01/2017 : total (12.75) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) +
Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
--------------------------------------------------------------------------
Overview of pay rates for soldier 2. Retired over the period: 201501-201512:
01/01/2015 : total payrate = 0 (retired)
--------------------------------------------------------------------------
Overview of pay rates for soldier 3. Regretting Retirement over the period: 201502-201712:
01/02/2015 : total (17.0) = Base(1.0) + Rank(captain:5.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(others:2.0) + Combat(yes:5.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(captain:5.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(others:2.0) + Combat(no:0.0)
01/07/2015 : total payrate = 0 (retired)
01/12/2015 : total (15.0) = Base(1.0) + Rank(captain:5.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(others:2.0) + Combat(yes:5.0)
01/01/2016 : total (13.0) = Base(1.0) + Rank(lieutenant:4.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2016 : total (15.75) = Base(1.25) + Rank(lieutenant:4.5) + Prof(cook:1.5) +
Service(reserve:1.5) + Unit(HQ:1.5) + Combat(yes:5.5)
01/01/2017 : total (17.25) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) +
Service(reserve:1.5) + Unit(HQ:1.5) + Combat(yes:5.5)
--------------------------------------------------------------------------
Overview of pay rates for soldier 4. Gaps and Overlaps over the period: 201501-201812:
01/01/2015 : total (7.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2015 : total (5.0) = Base(1.0) + Rank(private:1.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(HQ:1.0) + Combat(no:0.0)
Error in execution of sql code for database object (service)!
```



Implementation of the decision model in DT5GL (second solution):

```
SQLite_database: "Database/Soldier Payment.db"

# V2: detecting gaps and overlaps...
# Reference day = the 15th of the month.

Table 0:
If:                                          | 0| 1|
'Next soldier present'                       | Y| N|
Then:
NextSoldier is Selected                      | X|  |
NextSoldier is Finished                      |  | X|
# .......
# Repeat until: Finished

Proposition: 'Next soldier present'
Obtain_instance_from_database_view: soldier

Table 1:
If:                                          | 0| 1|
Next year in [firstYear-lastYear]            | Y| N|
Then:
EvalYear is Selected                         | X|  |
EvalYear is Finished                         |  | X|
# .......
# Repeat until: Finished

Table 2:
If:                                          | 0| 1| 2| 3| 4| 5|
Next month in [firstMonth-lastMonth]         | Y| Y| Y| Y| Y| N|
service.description = "retired"              | Y| Y| N| N| N| -|
current_pay_rate = 0                         | Y| N| -| -| -| -|
errorstring = ""                             | -| -| Y| Y| N| -|
current_pay_rate = pay_rate_this_month       | -| -| Y| N| -| -|
Then:
EvalMonth is Skip                            | X|  | X|  |  |  |
EvalMonth is NoPayRate                       |  | X|  |  |  |  |
EvalMonth is NewPayRate                      |  |  |  | X|  |  |
EvalMonth is Error                           |  |  |  |  | X|  |
EvalMonth is Finished                        |  |  |  |  |  | X|
# .......
# Repeat until: Finished


# Range of years to be checked: [firstYear-lastYear]
Attribute: firstYear                 Type: Integer
Equals: int(soldier.yearmonth_start/100)

Attribute: lastYear                  Type: Integer
Equals: int(soldier.yearmonth_end/100)

# Range of months to be checked within selected year: [firstMonth-lastMonth]
Attribute: firstMonth                Type: Integer
Equals: soldier.yearmonth_start % 100 if year == firstYear else 1

Attribute: lastMonth                 Type: Integer
Equals: soldier.yearmonth_end % 100 if year == lastYear  else 12

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

Attribute: fromdate                  Type: Text
Equals: "01/" + zerofill(month,2) + "/" + str(year)

Attribute: soldier.yearmonth_start      Type: Integer
Attribute: soldier.yearmonth_end        Type: Integer
```

```
Attribute: pay_rate_this_month  Type: Real
Equals: base.rate + rank.rate + profession.rate + service.rate + unit.rate +
combat.rate

Attribute: errorstring[3]
Equals: rankstring + professionstring + servicestring + unitstring + combatstring

Attribute: rankstring      Type: Text
Equals: "Missing Rank "    if rank.no == 0 else  \
        "Overlap in Rank " if rank.no > 1  else ""

Attribute: professionstring      Type: Text
Equals: "Missing Profession "    if profession.no == 0 else  \
        "Overlap in Profession " if profession.no > 1  else ""

Attribute: servicestring      Type: Text
Equals: "Missing Service "    if service.no == 0 else  \
        "Overlap in Service " if service.no > 1  else ""

Attribute: unitstring      Type: Text
Equals: "Missing Unit "    if unit.no == 0 else  \
        "Overlap in Unit " if unit.no > 1  else ""

Attribute: combatstring      Type: Text
Equals: "Missing Combat "    if combat.no == 0 else  \
        "Overlap in Combat " if combat.no > 1  else ""


Attribute: base.rate        Type: Real
Attribute: rank.rate        Type: Real
Attribute: profession.rate  Type: Real
Attribute: service.rate     Type: Real
Attribute: unit.rate        Type: Real
Attribute: combat.rate      Type: Real
Attribute: base.no          Type: Integer
Attribute: rank.no          Type: Integer
Attribute: profession.no    Type: Integer
Attribute: service.no       Type: Integer
Attribute: unit.no          Type: Integer
Attribute: combat.no        Type: Integer
```

[3] Why not a 'Type: Text' here? The answer is very simple: the attribute errorstring appears as a condition in the third decision table: errorstring = "", and on that basis the type could already be determined.

```
########################## Database views ############################

Database_view: soldier
With_attributes: id, name, yearmonth_start, yearmonth_end
Query:
 SELECT id, name, yearmonth_start, yearmonth_end
  FROM soldier
 LIMIT 1 OFFSET %s
With_arguments: soldier.auto_index

Database_view: base
With_attributes: description, rate, no
Query:
SELECT base.description, base_rate.rate, count(*)
  FROM base_contract
       JOIN base ON base_contract.type = base.id
       JOIN base_rate ON base.id = base_rate.base_id
 WHERE base_contract.soldierid = %s AND
      '%s' BETWEEN base_contract.date_from
               AND COALESCE(base_contract.date_to,'9999-12-31') AND
      '%s' BETWEEN base_rate.date_from
               AND COALESCE(base_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: rank
With_attributes: description, rate, no
Query:
SELECT rank.description, rank_rate.rate, count(*)
  FROM rank_contract
       JOIN rank ON rank_contract.type = rank.id
       JOIN rank_rate ON rank.id = rank_rate.rankid
 WHERE rank_contract.soldierid = %s AND
      '%s' BETWEEN rank_contract.date_from
               AND COALESCE(rank_contract.date_to,'9999-12-31') AND
      '%s' BETWEEN rank_rate.date_from
               AND COALESCE(rank_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: profession
With_attributes: description, rate, no
Query:
SELECT profession.description, profession_rate.rate, count(*)
  FROM profession_contract
       JOIN profession ON profession_contract.type = profession.id
       JOIN profession_rate ON profession.id = profession_rate.professionid
 WHERE profession_contract.soldierid = %s AND
       '%s' BETWEEN profession_contract.date_from
                AND COALESCE(profession_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN profession_rate.date_from
                AND COALESCE(profession_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: service
With_attributes: description, rate, no
Query:
SELECT service.description, service_rate.rate, count(*)
  FROM service_contract
       JOIN service ON service_contract.type = service.id
       JOIN service_rate ON service.id = service_rate.service_id
 WHERE service_contract.soldierid = %s AND
       '%s' BETWEEN service_contract.date_from
                AND COALESCE(service_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN service_rate.date_from
                AND COALESCE(service_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate
```

```
Database_view: unit
With_attributes: description, rate, no
Query:
SELECT unit.description, unit_rate.rate, count(*)
  FROM unit_contract
       JOIN unit ON unit_contract.type = unit.id
       JOIN unit_rate ON unit.id = unit_rate.unitid
 WHERE unit_contract.soldierid = %s AND
       '%s' BETWEEN unit_contract.date_from
                AND COALESCE(unit_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN unit_rate.date_from
                AND COALESCE(unit_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate


Database_view: combat
With_attributes: description, rate, no
Query:
SELECT combat.description, combat_rate.rate, count(*)
  FROM combat_contract
       JOIN combat ON combat_contract.type = combat.id
       JOIN combat_rate ON combat.id = combat_rate.combatid
 WHERE combat_contract.soldierid = %s AND
       '%s' BETWEEN combat_contract.date_from
                AND COALESCE(combat_contract.date_to,'9999-12-31') AND
       '%s' BETWEEN combat_rate.date_from
                AND COALESCE(combat_rate.date_to,'9999-12-31')
With_arguments: soldier.id, refdate, refdate
```

```
########################## GoalAttributes ############################

GoalAttribute: NextSoldier
Repeat_until:  Finished

Case: Finished
Print: "End!"

Case: Selected
Print: "Overview of pay rates for soldier %s. %s over the period: %s-%s:"
soldier.id soldier.name soldier.yearmonth_start soldier.yearmonth_end
>>: current_pay_rate = -1.00


GoalAttribute: EvalYear
Repeat_until:  Finished

Case: Finished
Print: "-------------------------------------------------------------------------"

Case: Selected
Print: "#REM# -  "


GoalAttribute: EvalMonth
Repeat_until: Finished

Case: Finished
Print: "#REM# -  "

Case: Skip
Print: "#REM# -  "

Case: NoPayRate
Print: "%s : total payrate = 0 (retired) "  fromdate
>>: current_pay_rate = 0

Case: NewPayRate
Print: "%s : total (%s) = Base(%s) + Rank(%s:%s) + Prof(%s:%s) + Service(%s:%s) +
Unit(%s:%s) + Combat(%s:%s)" fromdate pay_rate_this_month base.rate
rank.description rank.rate profession.description profession.rate
service.description service.rate unit.description unit.rate combat.description
combat.rate
>>: current_pay_rate = pay_rate_this_month

Case: Error
Print: "%s : %s"  fromdate  errorstring
```

Testrun second solution:

```
Overview of pay rates for soldier 1. Not Retired over the period: 201501-201812:
01/01/2015 : total (7.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2015 : total (5.0) = Base(1.0) + Rank(private:1.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/12/2015 : total (6.0) = Base(1.0) + Rank(private:1.0) + Prof(driver:1.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/01/2016 : total (7.0) = Base(1.0) + Rank(corporal:2.0) + Prof(driver:1.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/07/2016 : total (9.35) = Base(1.25) + Rank(corporal:2.5) + Prof(driver:1.6) +
Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
01/01/2017 : total (12.75) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) +
Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
------------------------------------------------------------------------------
Overview of pay rates for soldier 2. Retired over the period: 201501-201512:
01/01/2015 : total payrate = 0 (retired)
------------------------------------------------------------------------------
Overview of pay rates for soldier 3. Regretting Retirement over the period: 201502-201712:
01/02/2015 : total (17.0) = Base(1.0) + Rank(captain:5.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(others:2.0) + Combat(yes:5.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(captain:5.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(others:2.0) + Combat(no:0.0)
01/07/2015 : total payrate = 0 (retired)
01/12/2015 : total (15.0) = Base(1.0) + Rank(captain:5.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(others:2.0) + Combat(yes:5.0)
01/01/2016 : total (13.0) = Base(1.0) + Rank(lieutenant:4.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2016 : total (15.75) = Base(1.25) + Rank(lieutenant:4.5) + Prof(cook:1.5) +
Service(reserve:1.5) + Unit(HQ:1.5) + Combat(yes:5.5)
01/01/2017 : total (17.25) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) +
Service(reserve:1.5) + Unit(HQ:1.5) + Combat(yes:5.5)
------------------------------------------------------------------------------
Overview of pay rates for soldier 4. Gaps and Overlaps over the period: 201501-201812:
01/01/2015 : total (7.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2015 : total (5.0) = Base(1.0) + Rank(private:1.0) + Prof(cook:1.0) +
Service(reserve:1.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/12/2015 : Missing Rank Overlap in Profession Overlap in Service Missing Unit
01/01/2016 : Missing Rank Overlap in Profession
01/02/2016 : total (7.0) = Base(1.0) + Rank(corporal:2.0) + Prof(driver:1.0) +
Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/07/2016 : total (9.35) = Base(1.25) + Rank(corporal:2.5) + Prof(driver:1.6) +
Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
01/01/2017 : Overlap in Rank
01/02/2017 : total (12.75) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) +
Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
------------------------------------------------------------------------------
End!
Time elapsed: 0:00:01.922821
```

```
Overview of pay rates for soldier 1. Not Retired over the period: 201501-201812:
01/01/2015 : total (7.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) + Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) + Service(active:2.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2015 : total (5.0) = Base(1.0) + Rank(private:1.0) + Prof(cook:1.0) + Service(reserve:1.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/12/2015 : total (6.0) = Base(1.0) + Rank(private:1.0) + Prof(driver:1.0) + Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/01/2016 : total (7.0) = Base(1.0) + Rank(corporal:2.0) + Prof(driver:1.0) + Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/07/2016 : total (9.35) = Base(1.25) + Rank(corporal:2.5) + Prof(driver:1.6) + Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
01/01/2017 : total (12.75) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) + Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
--------------------------------------------------------------------------------
Overview of pay rates for soldier 2. Retired over the period: 201501-201512:
01/01/2015 : total payrate = 0 (retired)
--------------------------------------------------------------------------------
Overview of pay rates for soldier 3. Regretting Retirement over the period: 201502-201712:
01/02/2015 : total (17.0) = Base(1.0) + Rank(captain:5.0) + Prof(fighter:2.0) + Service(active:2.0) + Unit(others:2.0) + Combat(yes:5.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(captain:5.0) + Prof(fighter:2.0) + Service(active:2.0) + Unit(others:2.0) + Combat(no:0.0)
01/07/2015 : total payrate = 0 (retired)
01/12/2015 : total (15.0) = Base(1.0) + Rank(captain:5.0) + Prof(cook:1.0) + Service(reserve:1.0) + Unit(others:2.0) + Combat(yes:5.0)
01/01/2016 : total (13.0) = Base(1.0) + Rank(lieutenant:4.0) + Prof(cook:1.0) + Service(reserve:1.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2016 : total (15.75) = Base(1.25) + Rank(lieutenant:4.5) + Prof(cook:1.5) + Service(reserve:1.5) + Unit(HQ:1.5) + Combat(yes:5.5)
01/01/2017 : total (17.25) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) + Service(reserve:1.5) + Unit(HQ:1.5) + Combat(yes:5.5)
--------------------------------------------------------------------------------
Overview of pay rates for soldier 4. Gaps and Overlaps over the period: 201501-201812:
01/01/2015 : total (7.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) + Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/04/2015 : total (12.0) = Base(1.0) + Rank(private:1.0) + Prof(fighter:2.0) + Service(active:2.0) + Unit(HQ:1.0) + Combat(yes:5.0)
01/07/2015 : total (5.0) = Base(1.0) + Rank(private:1.0) + Prof(cook:1.0) + Service(reserve:1.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/12/2015 : Missing Rank Overlap in Profession Overlap in Service Missing Unit
01/01/2016 : Missing Rank Overlap in Profession
01/02/2016 : total (7.0) = Base(1.0) + Rank(corporal:2.0) + Prof(driver:1.0) + Service(active:2.0) + Unit(HQ:1.0) + Combat(no:0.0)
01/07/2016 : total (9.35) = Base(1.25) + Rank(corporal:2.5) + Prof(driver:1.6) + Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
01/01/2017 : Overlap in Rank
01/02/2017 : total (12.75) = Base(1.25) + Rank(sergeant:3.5) + Prof(officer:4.0) + Service(active:2.5) + Unit(HQ:1.5) + Combat(no:0.0)
--------------------------------------------------------------------------------
End!
Time elapsed: 0:00:01.922821
```

1 decision table rather than 3?

```
Table 0:
If:                                    | 0| 1| 2| 3| 4| 5| 6| 7| 8| 9|
'Next soldier present'                 | Y| N| -| -| -| -| -| -| -| -|
Next year in [firstYear-lastYear]      | -| -| Y| N| -| -| -| -| -| -|
Next month in [firstMonth-lastMonth]   | -| -| -| -| Y| Y| Y| Y| Y| N|
service.description = "retired"        | -| -| -| -| Y| Y| N| N| N| -|
current_pay_rate = 0                   | -| -| -| -| Y| N| -| -| -| -|
errorstring = ""                       | -| -| -| -| -| -| Y| Y| N| -|
current_pay_rate = pay_rate_this_month | -| -| -| -| -| -| Y| N| -| -|
Then:
NextSoldier is Selected                | X|  |  |  |  |  |  |  |  |  |
NextSoldier is Finished                |  | X|  |  |  |  |  |  |  |  |
EvalYear is Selected                   |  |  | X|  |  |  |  |  |  |  |
EvalYear is Finished                   |  |  |  | X|  |  |  |  |  |  |
EvalMonth is Skip                      |  |  |  |  | X|  | X|  |  |  |
EvalMonth is NoPayRate                 |  |  |  |  |  | X|  |  |  |  |
EvalMonth is NewPayRate                |  |  |  |  |  |  |  | X|  |  |
EvalMonth is Error                     |  |  |  |  |  |  |  |  | X|  |
EvalMonth is Finished                  |  |  |  |  |  |  |  |  |  | X|
# .......
```