/\*\*\*

Title: step1\_dedup\_inpatient\_cases\_from\_HOAP

Description: Generates a data set of acute inpatient cases (=stays) from the HOAP.HOA "case" tables.

Eliminates duplicates and merges cases that are contiguous by 1 day.

Is 1st step in generating analytic data sets for readmission rate computation.

Version Control: https://dsghe.lacare.org/nblume/Readmissions/tree/master/Code/Data\_Acquisition\_and\_Understanding/Cloudera%20DSW/Iteration2/

Data Source: HOAP.HOA QNXT, CLM and ENC case tables.

Output: NATHALIE.prjrea\_step1\_inpatient\_cases

All under review

\*\*\*/

/\*

UNIQUE\_CASES

Purpose: To merge data from HOA's 3 case tables with priority QNXT>CLM>ENC

Notes: 1. Inpatient cases are identified with 'where srv\_cat = '01ip\_a' on the sace files.

This excludes more SNF inpatient stays than using substr(type\_bill,1,2) in ('11','12') on the hdr files.

2. Unique tupple (cin\_no, admi\_dt, dis\_dt) are selected with priority QNXT>CLM>ENC

3. There is a potential loss of Dx and Pr information when cases are deduped over cin\_no, admit\_dt and dis\_dt alone

\*/

create table NATHALIE.TMP\_UNIQUE\_CASES

as

-- select only 1 with same (cin\_no, admi\_dt, dis\_dt) tupple

-- add row number by cin\_no partition. Will be used at next setp.

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

(

select \*

from

( --add number rows inside partitions where each partition is a unique (cin\_no, admi\_dt, dis\_dt) tupple

select \*

, row\_number() over(partition by cin\_no, adm\_dt, dis\_dt order by source\_table asc, case\_id desc) as rownumber

from

( -- union of cases across 3 data tables: qnxt, clm, enc

select case\_id, adm\_dt, dis\_dt, cin\_no

, case\_dx1

, case\_pr1

, severity, aprdrg, dis\_status, provider, paid\_amt\_case, from\_er

, 1 as source\_table

from hoap.QNXT\_CASE\_INPSNF as Q

where srv\_cat = '01ip\_a'

union

select case\_id, adm\_dt, dis\_dt, cin\_no

, case\_dx1

, case\_pr1

, severity, aprdrg, dis\_status, provider, paid\_amt\_case, from\_er

, 2 as source\_table

from hoap.clm\_case\_inpsnf as C

where srv\_cat = '01ip\_a'

union

select case\_id, adm\_dt, dis\_dt, cin\_no

, case\_dx1

, case\_pr1

, severity, aprdrg, dis\_status, provider, null as paid\_amt\_case, from\_er

, 3 as source\_table

from hoap.ENC\_CASE\_INPSNF as E

where srv\_cat = '01ip\_a'

) AS ALL\_CASES

order by cin\_no, adm\_dt, dis\_dt

) ALL\_CASES\_PARTITIONED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, 4 as source\_table

, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

/\*

FUSE INITIAL STAY WITH TRANSFERS or 1d READMITS

Purpose: To merge rows that concern contiguous stays (also to capture cases that overlap in time). Contiguiity = discharge and admit are at most 1 day apart.

Notes: 1. Awkward implementation because looping is not permitted in Impala environment (see https://stackoverflow.com/questions/49523380/write-a-while-loop-in-impala-sql)

Therefore as long as you need to fuse admits, you need to hard-code the repetition of the search-and-fuse script below.

2. FS = "first stay" and SS = "second stay"

3. What is kept:

From FS in fusing contiguous stays: fromER, Dx1, Pr1, severity and aprdrg

From SS in fusing contiguous stays: discharge date from 2nd stay, and max(dis\_dt).

What is concatenated (both FS and SS values are kept): case\_id, source tables, provider.

\*/

-- Iteration 1

create table NATHALIE.TMP\_FUSED\_CASES\_1

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_UNIQUE\_CASES as FS

inner join

NATHALIE.TMP\_UNIQUE\_CASES as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

/\*

The next set of 'Iteration x' are identical to Iteration 1 excpet for the table names. If the logic of Iteration 1 is valid so is the logic of Iteration x.

\*/

-- Iteration 2

create table NATHALIE.TMP\_FUSED\_CASES\_2

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_1 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_1 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

-- Iteration 3

create table NATHALIE.TMP\_FUSED\_CASES\_3

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_2 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_2 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

-- Iteration 4

create table NATHALIE.TMP\_FUSED\_CASES\_4

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_3 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_3 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 5

create table NATHALIE.TMP\_FUSED\_CASES\_5

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_4 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_4 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 6

create table NATHALIE.TMP\_FUSED\_CASES\_6

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_5 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_5 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

-- Iteration 7

create table NATHALIE.TMP\_FUSED\_CASES\_7

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_6 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_6 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 8

create table NATHALIE.TMP\_FUSED\_CASES\_8

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_7 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_7 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 9

create table NATHALIE.TMP\_FUSED\_CASES\_9

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_8 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_8 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 10

create table NATHALIE.TMP\_FUSED\_CASES\_10

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_9 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_9 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 11

create table NATHALIE.TMP\_FUSED\_CASES\_11

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_10 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_10 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 12

create table NATHALIE.TMP\_FUSED\_CASES\_12

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_11 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_11 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 13

create table NATHALIE.TMP\_FUSED\_CASES\_13

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_12 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_12 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 14

create table NATHALIE.TMP\_FUSED\_CASES\_14

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_13 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_13 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 15

create table NATHALIE.TMP\_FUSED\_CASES\_15

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_14 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_14 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

--Iteration 16

create table NATHALIE.TMP\_FUSED\_CASES\_16

as

select \*

, row\_number() over (order by cin\_no asc, adm\_dt asc, dis\_dt asc) as rownumber2

from

( --PADDING\_ADDED

select \*

from

( -- ROWNUMER\_ADDED // order newly engineered cases above cases that are transfers and whose admit date is later

select \*

, row\_number() over(partition by cin\_no, dis\_dt order by adm\_dt asc) as rownumber

from

( -- VALS\_UPDATED // use stay\_interval to decide whether to replace some FS values with their SS analogs

select

case

when stay\_interval < 2 then ss\_case\_id

else fs\_case\_id

end as case\_id

, cin\_no, adm\_dt

, case

when stay\_interval < 2 then ss\_dis\_dt

else fs\_dis\_dt

end as dis\_dt

, from\_er, case\_dx1, case\_pr1, severity, aprdrg, dis\_status, provider

, case

when (stay\_interval < 2 and fs\_case\_id != ss\_case\_id) then fs\_paid\_amt\_case + ss\_paid\_amt\_case

else fs\_paid\_amt\_case

end as paid\_amt\_case

, case

when stay\_interval < 2 then concat(fs\_source\_table, ', ', ss\_source\_table)

else fs\_source\_table

end as source\_table

, stay\_interval

from

( --'INTERVAL\_ADDED' // add interval between 1st discharge date and 2nd admit date to create subquery called 'interval\_added'

select

FS.case\_id as fs\_case\_id, FS.cin\_no, FS.adm\_dt, FS.dis\_dt as fs\_dis\_dt, FS.from\_er, FS.case\_dx1, FS.case\_pr1, FS.severity

, FS.aprdrg, FS.dis\_status, FS.provider, FS.paid\_amt\_case as fs\_paid\_amt\_case, cast(FS.source\_table as varchar(1)) as fs\_source\_table

, concat(FS.case\_id, ', ', SS.case\_id) as ss\_case\_id

-- Keep whichever is later: discharge date from FS or from SS.

, case

when datediff(SS.dis\_dt, FS.dis\_dt) < 0

then FS.dis\_dt

else SS.dis\_dt

end as ss\_dis\_dt

, SS.dis\_status as ss\_dis\_status, concat(FS.provider, ', ', SS.provider) as ss\_provider, SS.paid\_amt\_case as ss\_paid\_amt\_case --the paid values will be added at the next level of nesting

, concat(cast(FS.source\_table as varchar(1)), ', ', cast(SS.source\_table as varchar(1))) as ss\_source\_table

, case

when FS.cin\_no = SS.cin\_no

then datediff(SS.adm\_dt, FS.dis\_dt)

else null

end as stay\_interval

from

NATHALIE.TMP\_FUSED\_CASES\_15 as FS

inner join

NATHALIE.TMP\_FUSED\_CASES\_15 as SS

ON SS.rownumber2 = FS.rownumber2 + 1

) AS INTERVAL\_ADDED

) AS VALS\_UPDATED

) AS ROWNUMER\_ADDED

union

-- Adding a dummy row at the end of the file as padding for the next step. In the next step, the last row is sheared off

-- when the table is joined with itself with an offset of 1 row.

(

select \*

from (

select null as case\_id, '1900-01-01' as adm\_dt, '1900-01-01' as dis\_dt, 'ZZZZZ' as cin\_no, null as case\_dx1, null as case\_pr1

, null as severity, null as aprdrg, null as dis\_status, null as provider, null as paid\_amt\_case, null as from\_er, '4' as source\_table

, 99 as stay\_interval, 1 as rownumber

) PADDING

)

) ALL\_CASES\_PADDED

where rownumber = 1

;

/\*

End of "loop"

If there are still more records reduced (being consolidated), then will need to run more iterations till no more reduction in total number of cases

\*/

/\*

Save the last iteration

\*/

drop table if exists nathalie.prjrea\_step1\_inpatient\_cases;

create table nathalie.prjrea\_step1\_inpatient\_cases

as

select \*

from NATHALIE.TMP\_FUSED\_CASES\_16

;

/\*

CLEAN UP

\*/

drop table if exists NATHALIE.TMP\_UNIQUE\_CASES;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_1;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_2;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_3;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_4;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_5;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_6;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_7;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_8;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_9;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_10;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_11;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_12;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_13;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_14;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_15;

drop table if exists NATHALIE.TMP\_FUSED\_CASES\_16;