

ANNEXE PROGRAMMES CODE SAS - GUIDE PÉDAGOGIQUE PRÉ-TRAITEMENT ET VÉRIFICATION D'UNE EXTRACTION SNDS

Ces annexes présentent les programmes codés en SAS illustrant les mécanismes de pré-traitement et de vérification d'une extraction de la base principale du SNDS.

Pour chaque programme, vous trouverez les éléments du support pédagogique associés.

Ces programmes sont adaptés dans le cadre d'un accès à la plateforme SNDS sur projet, ils peuvent être adaptés dans le cadre d'un accès permanent en modifiant les noms de variables, de tables et en ajoutant, si nécessaire, des conditions. Exemple : sur les dates de flux dans la table ER_PRS_F de DCIR.

Étapes génériques pour vérifier son extraction de données (slide 10)

1

Contrôler la cohérence de la livraison

Comparer le nombre de tables/lignes/colonnes mentionnées dans le fichier Excel de Bilan de livraison fourni par la CNAM (bilan d'extraction) avec le nombre de tables/lignes/colonnes effectivement présentes dans l'extraction reçue

/ Indiquer le chemin vers la bibliothèque contenant les tables livrées */*

LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; / nom de librairie et chemin à adapter, notamment si données situées sur système fils autre que plateforme SNDS. La librairie LIB_IN sera utilisée par défaut dans les autres programmes présentés */*

/ Calcul du nombre de tables */*

```
proc sql;
  select count(*) as nb_tables
  /*SASHELP.VTABLE est une vue du dictionnaire des tables DICTIONARY.TABLES donnant accès
  aux métadonnées des tables de la session courante */
  from sashelp.vtable
  where libname = 'LIB_IN';
quit;
```

/ Détails par table */*

```
proc sql;
  select
    libname,
    memname as table_name,
    nobs as nb_lignes,
    nvar as nb_colonnes
  /*SASHELP.VTABLE est une vue du dictionnaire des tables DICTIONARY.TABLES donnant accès
  aux métadonnées des tables de la session courante */
  from sashelp.vtable
  where libname = "LIB_IN"
  order by table_name;
quit;
```

/ 2 tableaux seront affichés dans une page HTML : 1 tableau avec le nombre total de tables figurant dans la bibliothèque, 1 autre avec le nombre de lignes et de colonnes pour chaque table. Vous pourrez comparer ces éléments avec ceux figurant dans le Bilan d'extraction */*

2

Dénombrer les BEN_NIR_PSA/NUM_ENQ et BEN_NIR_ANO/NUM_ENQ_ANO de la table centrale IR_BEN_R afin d'estimer approximativement le nombre de bénéficiaires du SNDS présents dans la cohorte livrée et comparer avec les effectifs attendus et indiqués dans le Bilan de livraison fourni par la CNAM

```
proc sql;
  select
    count(distinct(NUM_ENQ)) as NB_NUM_ENQ,
    count(distinct(NUM_ENQ_ANO)) as NB_NUM_ENQ_ANO
  from LIB_IN.IR_BEN_R_AAAAMM;
quit; run;
```

/ 1 tableau sera affiché dans une page HTML avec le nombre distinct de valeurs pour chacun des 3 identifiants */*

Etapes génériques pour vérifier son extraction de données (slide 11)

4

Lister et s'assurer que **les variables sensibles autorisées** (Ex : date de soins complètes, code de la commune de résidence, mois de naissance, date de décès complète, code de la commune de décès...) **soient présentes**, et que **les variables sensibles non autorisées soient absentes**. Dans le cas contraire, il faudra **les supprimer des données extraites**.

```
/*-> Indiquez le chemin vers le dossier d'extraction */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/.....";

/*=====
* 1)Données sensibles potentielles et variables associées
*=====*/
proc format;
  value $ID_P
'BEN_RES_COM','TRS_DEP_CDP','TRS_ARR_CDP','BDI_COD','COD_POST'='CODE_COMMUNE'
'DCD_COM_COD','DCD_LIE_LIB'='LIEU_DECES'
'PRS_HOS_DTD','EXE_SOI_DTF','EXE_SOI_DTD','PRE_PRE_DTD','TRS_DTD_DTE',
'TRS_DTF_DTE','DAT_HOS','SOR_DAT','ENT_DAT','NUM_DAT_AT','DAT_PEC','DAT_DEB_FAC',
'FAC_AMC_DT','FAC_AMC_DT_CTL',
'FAC_AMO_DT','FAC_AMO_DT_CTL','FAC_RAC_DT','FAC_RAC_DT_CTL',
'PAI_AMC_DT','PAI_AMC_DT_CTL','PAI_AMO_DT','PAI_AMO_DT_CTL','PAI_RAC_DT',
'PAI_RAC_DT_CTL','DAT_FIN_FAC','AUT_DEB_DTD','AUT_FIN_DTD','ENT_TYP_UN',
'SOR_TYP_UN'='DATE_SOINS'
'BEN_DCD_DTE'='DATE_DECES'
'BEN_NAI_MOI'='AN_MOIS_NAISS'
other = 'Autres';

/*=====
* 2)Affichage et vérification des variables sensibles dans LIB
*=====*/
proc sql;
  create table dico_sensible as
  select
    put(upcase(name), $ID_P.) as ID_CAT,
    memname as table_name,
    upcase(name) as column_name
  /* SASHELP.VCOLUMN est une vue du dictionnaire des colonnes DICTIONARY.COLUMNS donnant
  accès aux métadonnées des colonnes des tables de la session courante */
  from sashelp.vcolumn
  where libname = "LIB_IN"
  and calculated ID_CAT ne 'Autres'
  order by ID_CAT;
quit;
/* Crée une table contenant toutes les variables sensibles présentes dans les fichiers de la librairie
LIB_IN */
```



Vérifier la présence et le bon format des variables d'identifiants pseudonymisés (NUM_ENQ / BEN_NIR_PSA / NIR_ANO_17, BEN_IDT_ANO / NUM_ENQ_IDT et BEN_NIR_ANO / NUM_ENQ_ANO) et du rang gémellaire (BEN_RNG_GEM) dans les tables de référentiel des bénéficiaires (IR_BEN_R/IR_BEN_R_ARC) ainsi que dans les autres tables censées les contenir (Ex : ER_PRS_F, IR_IMB_R, T_MCOxxC...) pour permettre des jointures correctes entre ces tables.

/-> Indiquez le chemin vers le dossier d'extraction */*

```
LIBNAME LIB_IN "%sysget(HOME)/sasdata/...";
```

/=====**

** 1) Format permettant d'identifier les variables correspondant aux identifiants patients*

/=====**

```
proc format;
```

```
value $IDENTIFIANT_PATIENT
```

```
'NUM_ENQ' , 'NUM_ENQ_IDT', 'NUM_ENQ_ANO', 'BEN_RNG_GEM', 'BEN_NIR_PSA', 'NIR_ANO_17',  
'BEN_IDT_ANO', 'BEN_NIR_ANO'='ID'
```

```
other = 'Autres';
```

/ ATTENTION : La variable NUM_ENQ_IDT (correspondant à BEN_IDT_ANO pour les accès permanents) n'existe pas par défaut dans les tables livrées (sauf dans les tables KI_CCI_R et KI_ECD_R). Il est recommandé de la créer (si possible) selon la règle suivante :*

- Utiliser NUM_ENQ_ANO lorsqu'il existe (identifiant unique) ;
- À défaut, utiliser la concaténation NUM_ENQ || BEN_RNG_GEM.

Attention : dans ce dernier cas, l'identifiant obtenu n'est pas toujours unique.

**/*

/=====**

** 2) Affichage et vérification des identifiants dans LIB*

/=====**

```
proc sql;
```

```
create table table_id as
```

```
select memname as table_name,/* Nom de la table*/
```

```
upcase(name) as column_name,/* Nom de la colonne*/
```

```
type, /* Type de la colonne*/
```

```
length, /* longueur */
```

```
format /* format SAS */
```

/ SASHELP.VCOLUMN est une vue du dictionnaire des colonnes DICTIONARY.COLUMNS donnant accès aux métadonnées des colonnes des tables de la session courante */*

```
from sashelp.vcolumn
```

```
where libname = "LIB_IN"
```

```
and put(upcase(name), $IDENTIFIANT_PATIENT.) ne 'Autres'
```

```
order by column_name,table_name;
```

```
quit;
```

*/**

Un tableau sera affiché présentant les différents identifiants patients disponibles dans les tables livrées. Pour chaque identifiant, le type, la longueur et le format seront précisés.

**/*



Vérifier la **quantité mensuelle de soins et de sujets dans chaque table** (ER_PRS_F, tables affinées de DCIR, T_MCOxxB, T_SSRxxB...) afin de contrôler si de fortes hausses ou baisses de quantité de soins sont observées dans les données extraites et ainsi identifier tout problème d'extraction ou de chargement des données. Des tableaux et graphiques réalisés sur la fréquence mensuelle des soins ou des sujets distincts peuvent être édités pour cela (voir diapositive suivante)

/ Exemple de graphique d'édition du nombre de soins mensuels, ici à partir de la table ER_PRS_F. Ceci peut être également fait :*

- *en reliant cette table avec ER_PHA_F pour les médicaments, ER_CAM_F pour les procédures, ER_BIO_F pour les actes de biologie...*
- *en remplaçant les tables ER_PRS_F_2020 à 2024 par T_MCO20C à T_MCO24C pour les séjours MCO du PMSI*

*⇒ A adapter selon la source des données et le nom des tables */*

```
options mprint mlogic nodate nonumber;
ods graphics on;
```

/ 1) Création de la table fusionnée */*

```
%macro union_tables_PRS(startann=, endann=);
proc sql;
create table ER_PRS_F_&startann._&endann as
%do year=&startann %to &endann;
select * from ER_PRS_F_&year
%if &year < &endann %then union all;
%end;
order by EXE_SOI_DTD;
quit;
%mend union_tables_PRS;
%union_tables_PRS(startann=2020, endann=2024);
```

/ 2) Restreindre la période et calculer le mois */*

```
%let start = '01JAN2020'd;
%let end = '31DEC2024'd;

data prs_filtered;
set ER_PRS_F_2020_2024;
where &start <= EXE_SOI_DTD <= &end; /* Si EXE_SOI_DTD est une date SAS */
mois = intnx('month', EXE_SOI_DTD, 0, 'b');
format mois yymmnn6.;
```

/ Remarque : Si c'est la variable est un datetime, utiliser :*

```
where &start <= datepart(EXE_SOI_DTD) <= &end;
mois = intnx('month', datepart(EXE_SOI_DTD), 0, 'b');
*/
run;
```

/ 3) Compter le nombre de soins par mois avec PROC FREQ */*

```
proc freq data=prs_filtered noprint;
tables mois / out=counts(drop=percent rename=(count=nb_soins));
```

```
run;

/* 4) Générer un calendrier mensuel complet pour avoir aussi les mois sans soins */

data calendrier;
do i = 0 to intck('month', &start, &end);
mois = intrnx('month', &start, i, 'b');
output;
end;
drop i;
format mois yymmn6.;
run;

/* 5) Joindre pour compléter les mois manquants avec 0 */

proc sql;
create table counts_full as
select a.mois,
coalesce(b.nb_soins, 0) as nb_soins
from calendrier as a
left join counts as b
on a.mois = b.mois
order by a.mois;
quit;

/* 6) Tracer la courbe */

title "Nombre de soins mensuels (ER_PRS_F) - 2020 à 2024";
proc sgplot data=counts_full;
series x=mois y=nb_soins / markers;
xaxis label="Mois" values=(intrnx('month', &start, 0, 'b') to intrnx('month', &end, 0, 'b'))
valuesformat=yyymm6. fitpolicy=stagger;
yaxis label="Nombre de soins";
run;
title;
```

Gestion des doublons entre DCIR, PMSI et CépiDC : date de décès (slide 22)

Différentes méthodes peuvent être envisagées pour déterminer la date de décès en cas d'informations discordantes. Pour chacune d'elles, il est conseillé de recenser l'information aussi largement que possible depuis les différentes tables contenant l'information de date de décès. La seconde étape consiste alors à choisir quelle information prioriser et conserver lorsque celle-ci est présente dans plusieurs tables et qu'elle n'est pas uniforme dans toutes les sources.

```
/* Veuillez préciser le chemin vers la librairie destinée à stocker la table des décès. */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/...";/* Chemin pour le stockage de la table des résultats */
```

```
/* Chemin vers les données livrées */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/...";
```

```
/*******/
/* 1- Creation table avec décès d'IR_BEN_R */
/*******/
```

```
PROC SQL;
  CREATE TABLE LIB_OUT.DECES_IR_BEN_R AS
  SELECT DISTINCT t1.NUM_ENQ_ANO,
    t1.NUM_ENQ,
    t1.BEN_RNG_GEM,
    t1.BEN_DCD_DTE
  FROM LIB_IN.IR_BEN_R_AAAAMM t1
  WHERE t1.BEN_DCD_DTE NE '01JAN1600'd;
QUIT;
```

```
/*******/
/* 2- Creation table avec décès de KI_CCI_R */
/*******/
```

```
PROC SQL;
  CREATE TABLE LIB_OUT.DECES_KI_CCI_R AS
  SELECT DISTINCT t1.NUM_ENQ_IDT,
    t1.BEN_DCD_DTE /* VARIABLES A REMPLACER EN COHERENCE AVEC LES
VARIABLES DES EXTRACTIONS */
  FROM LIB_IN.KI_CCI_R t1;
QUIT;
```

```
/*******/
/* 3- Creation table avec décès de MCO */
/*******/
```

```
/* Veuillez préciser le chemin des librairies contenant les tables sources,
ainsi que la librairie destinée à stocker les tables des résultats. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/sasdata1/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/sasdata1/Programmes formation test";/* Chemin
pour le stockage de la table des résultats */
```

```
/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
```


*pour une table donnée, ou dans une boucle. */*

```
/*=====*/
/* =====(1) Définition de la macro ===== */
/*=====*/
```

```
%macro EXTRACTION_MCO_DECES(annees=);
```

```
%let fin_geo = '130780521', '130783236', '130783293', '130784234',
'130804297','600100101','750041543', '750100018', '750100042', '750100075', '750100083',
'750100091', '750100109', '750100125', '750100166', '750100208', '750100216',
'750100232', '750100273', '750100299' , '750801441', '750803447', '750803454',
'910100015', '910100023', '920100013', '920100021', '920100039', '920100047', '920100054',
'920100062', '930100011', '930100037', '930100045', '940100027',
'940100035', '940100043', '940100050', '940100068', '950100016', '690783154',
'690784137', '690784152', '690784178', '690787478', '830100558' ;
```

```
/* Suppression de la table finale si elle existe déjà */
```

```
proc datasets lib=LIB_OUT nolist;
  delete table_deces_mco;
quit;
```

```
/* Nombre d'annees a traiter */
```

```
%let n = %sysfunc(countw(&annees));
```

```
/* Boucle sur les anneess */
```

```
%do i = 1 %to &n;
```

```
  /* Recuperation de l'année */
```

```
  %let annee = %scan(&annees, &i);
```

```
  %let an = %substr(&annee, 3, 2);
```

```
  %put NOTE: === Traitement &annee ===;
```

```
  /* Nom de la table de travail */
```

```
  %let table_work = table_mco_deces&annee;
```

```
  /* Suppression de la table de travail si elle existe */
```

```
  proc datasets lib=work nolist;
    delete &table_work;
  quit;
```

```
  /* ----- CREATION DE LA TABLE ----- */
```

```
  PROC SQL;
```

```
      CREATE TABLE work.&table_work as
      SELECT DISTINCT t2.NUM_ENQ,
      t1.SOR_MOD,
      t2.EXE_SOI_DTF as BEN_DCD_DTE
      FROM LIB_IN.T_MCO&an.B t1 INNER JOIN LIB_IN.T_MCO&an.C t2
      ON (t1.ETA_NUM = t2.ETA_NUM AND t1.RSA_NUM = t2.RSA_NUM)
      WHERE t1.SOR_MOD="9"
```

```

        AND t2.eta_num not in (&fin_geo)
        AND t2.NIR_RET='0' AND t2.NAI_RET='0' AND t2.SEX_RET='0' AND
t2.SEJ_RET='0' AND t2.FHO_RET='0' AND t2.PMS_RET='0' AND t2.DAT_RET='0' AND
t2.COH_NAI_RET='0' AND t2.COH_SEX_RET='0'
        AND t1.grg_ghm not like "90%" AND t1.typ_gen_rsa='0'
        AND t1.ent_mod <> '0' and t1.sor_mod <> '0';

QUIT;

/* Creation de la table finale */
%if not %sysfunc(exist(LIB_OUT.table_decès_mco)) %then %do;

    data LIB_OUT.table_decès_mco;
        set work.&table_work;
    run;

%end;

/* Ajout des tables*/
%else %do;

    data LIB_OUT.table_decès_mco;
        set LIB_OUT.table_decès_mco work.&table_work;
    run;

%end;

%end;

/* Fin macro */
%mend EXTRACTION_MCO_DECES;

/* ===== */
/* =====(2) Exemple d'appel de la macro ===== */
/* ===== */

%EXTRACTION_MCO_DECES(annees=2020 2021); /* indiquer ici toutes les années 20xx en lien
avec les fichiers T_MCOxxB extraits dans la librairie LIB_IN */

/*****/
/* Une fois les tables DECES_IR_BEN_R, DECES_KI_CCI_R et TABLE_DECES_MCO créées, vous
pourrez les fusionner par un identifiant commun entre les tables (NUM_ENQ, NUM_ENQ_ANO,
NUM_ENQ_IDT) afin d'avoir toutes les dates dans un seul fichier */
/*****/

/* recréation NUM_ENQ_IDT pour fusion avec DECES_KI_CCI */
DATA work.DECES_IR_BEN_R_IDT;
SET LIB_OUT.DECES_IR_BEN_R;

length NUM_ENQ_IDT $30;

IF not missing(NUM_ENQ_ANO) THEN
NUM_ENQ_IDT = NUM_ENQ_ANO;
ELSE

```

```

NUM_ENQ_IDT = cats(NUM_ENQ, put(BEN_RNG_GEM,2.));
RUN;

PROC SQL;
CREATE TABLE work.TOTAL_DECES AS
SELECT DISTINCT NUM_ENQ, NUM_ENQ_ANO, NUM_ENQ_IDT
FROM work.DECES_IR_BEN_R_IDT

UNION

SELECT DISTINCT
" AS NUM_ENQ,
" AS NUM_ENQ_ANO,
NUM_ENQ_IDT
FROM LIB_OUT.DECES_KI_CCI_R

UNION

SELECT DISTINCT
NUM_ENQ,
" AS NUM_ENQ_ANO, /* pour créer les variables à vide */
" AS NUM_ENQ_IDT /* pour créer les variables à vide */
FROM LIB_OUT.TABLE_DECES_MCO;
QUIT;

PROC SQL;
CREATE TABLE LIB_OUT.DECES_IRBEN_KICCI_MCO AS
SELECT DISTINCT
p.NUM_ENQ,
p.NUM_ENQ_ANO,
p.NUM_ENQ_IDT,

ir.BEN_DCD_DTE AS DCD_IR_DTE,
ki.BEN_DCD_DTE AS DCD_KI_DTE,
mco.BEN_DCD_DTE AS DCD_MCO_DTE,
ir.BEN_RNG_GEM

FROM work.TOTAL_DECES p

LEFT JOIN work.DECES_IR_BEN_R_IDT ir
ON p.NUM_ENQ = ir.NUM_ENQ
OR p.NUM_ENQ_IDT = ir.NUM_ENQ_IDT

LEFT JOIN LIB_OUT.DECES_KI_CCI_R ki
ON p.NUM_ENQ_IDT = ki.NUM_ENQ_IDT

LEFT JOIN LIB_OUT.TABLE_DECES_MCO mco
ON p.NUM_ENQ = mco.NUM_ENQ;
QUIT;

```

Supprimer les informations erronées au sein de son extraction (slide 27)

La variable **BEN_AMA_COD** dans la table ER_PRS_F de DCIR contient l'âge du bénéficiaire à la date du soin. Cette variable est calculée comme le délai entre l'année et le mois de la date du soin EXE_SOI_DTD (ou à défaut de la date de liquidation) et l'année et le mois de naissance du bénéficiaire. Elle est exprimée en mois / années révolu(e)s :

- **bénéficiaires < 2 ans : BEN_AMA_COD = 1 000 + âge en mois MM** (MM = [00-23 mois], c'est à dire l'âge en mois révolus)
Exemple : Un nourrisson a la valeur BEN_AMA_COD = 1001 jusqu'à ses 2 mois, etc. jusqu'à 1023 quand il a 23 mois. Ensuite, il a la valeur BEN_AMA_COD = 2 à partir du jour de son 2ème anniversaire.
- **bénéficiaires ≥ 2 ans : BEN_AMA_COD = XXX** (XXX représente l'âge en années révolues)
Exemple : une personne a la valeur BEN_AMA_COD = 25 le jour de son 25ème anniversaire, et aura la valeur BEN_AMA_COD = 26 lors de son 26ème anniversaire.

```
/* Veuillez préciser le chemin vers la librairie destinée à stocker les tables */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/...";
/* Chemin vers les données livrées */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/...";

/* =====(1) Définition de la macro ===== */
%macro ER_PRS_F_AGE_CAT(annees=);

/* Nombre d'annees a traiter */
%let n = %sysfunc(countw(&annees));
/* Boucle sur les annees */
%do i = 1 %to &n;

/* Extraction de l'annee */
%let annee = %scan(&annees, &i);
/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
delete ER_PRS_F_AGE_CAT_&annee.;
quit;
%put NOTE: === Traitement ER_PRS_F_&annee ===;
PROC SQL;
CREATE TABLE LIB_OUT.ER_PRS_F_AGE_CAT_&annee. AS
SELECT
t1.*,
/* 1) Âge en années */
CASE
WHEN t1.ben_ama_cod BETWEEN 1000 AND 1024
THEN (t1.ben_ama_cod - 1000) / 12
ELSE t1.ben_ama_cod
END AS ben_ama_cod_en_annees,
/* 2) Catégorisation de l'âge */
CASE
WHEN t1.ben_ama_cod BETWEEN 1000 AND 1024
THEN "< 2 ans"
WHEN t1.ben_ama_cod BETWEEN 2 AND 110
THEN ">= 2 ans"
ELSE "NV" /* valeur aberrante (Non valide) */
END AS ben_ama_cod_cat LENGTH=20
FROM LIB_IN.ER_PRS_F_&annee. AS t1
;
QUIT;
```

```

        %end;

%mend ER_PRS_F_AGE_CAT;

/* =====(2) Exemple d'appel de la macro =====
*/

%ER_PRS_F_AGE_CAT(annees=2022);

/*
Cette macro crée des copies des tables ER_PRS_F_AAAA sur une période donnée
et ajoute deux variables :

- ben_ama_cod_en_annees : âge exprimé en années, permettant de traiter de façon homogène
les âges.

- ben_ama_cod_cat : variable permettant de distinguer les individus de moins de 2 ans
(ben_ama_cod de 1000 à 1024) des individus âgés de 2 ans et plus.
*/

```

Appliquer des filtres dans son extraction (données du PMSI) (slide 31)

Séjours/séances : il est recommandé d'appliquer systématiquement plusieurs filtres

MCO

- Exclure les FINESS géographiques APHP, APHM et HCL dont les remontées sont en doublons sur leur FINESS juridique avant 2017 inclus (via la table T_MCOxxB) ;
- Exclure les séjours avec un groupage GHM en erreur, dans la table T_MCOxxB, correspondant à :
 - Des séjours avec informations manquantes ou incohérentes (diagnostics, âge, date de sortie, etc.) ne permettant pas de grouper le séjour dans un GHM et de lui attribuer un tarif ;
 - De « faux » séjours générés automatiquement pour des besoins de facturation (dialyse, activité externe, passage aux urgences, forfaits, etc.) des ES ex-OQN ;
- Exclure les « faux » résumés de sortie anonymes (RSA) générés automatiquement (dialyses, activités externes de médecins salariés, ATU, FFM, SE...)
- Exclure les prestations inter établissements (PIE) via la table T_MCOxxB : transfert temporaire (< 2 jours) d'un patient dans un autre ES pour réaliser une prestation. La même prestation figure en double dans les remontées des 2 ES, mais aucune valorisation ne sera calculée pour l'ES prestataire ;
- Exclure les séjours avec des clés de chaînage incorrectes sur les informations des bénéficiaires via les variables codes retours XXX_RET de la table T_MCOxxC (**Attention** : le nombre de ces variables peut varier selon l'année)

```
SELECT C.NIR_ANO_17, B.ETA_NUM, B.RSA_NUM, C.EXE_SOI_DTD, C.EXE_SOI_DTF
FROM T_MCOaaB AS B
INNER JOIN T_MCOaaC AS C
  ON B.ETA_NUM = C.ETA_NUM
  AND B.RSA_NUM = C.RSA_NUM
```

```
/* Exclusion des FINESS géographiques APHP, APHM et HCL en doublons entre 2005 et 2017 */
WHERE B.ETA_NUM NOT IN ('130780521', '130783236', '130783293', '130784234', '130804297',
'600100101', '750041543', '750100018', '750100042', '750100075', '750100083', '750100091',
'750100109', '750100125', '750100166', '750100208', '750100216', '750100232', '750100273',
'750100299', '750801441', '750803447', '750803454', '910100015', '910100023', '920100013',
'920100021', '920100039', '920100047', '920100054', '920100062', '930100011', '930100037',
'930100045', '940100027', '940100035', '940100043', '940100050', '940100068', '950100016',
'690783154', '690784137', '690784152', '690784178', '690787478', '830100558')
```

```
/* Exclusion des séjours avec un groupage en erreur */
AND B.GRG_GHM NOT LIKE '90%'
```

```
/* Exclusion des faux séjours avec RSA */
AND TYP_GEN_RSA = '0'
```

```
/* Exclusion des PIE */
AND (B.SEJ_TYP <> 'B' OR B.SEJ_TYP IS NULL)
```

```
/* Exclusion des clés de chaînage incorrectes sur les informations des bénéficiaires */
```

```
/* Variables disponibles depuis 2005 */
AND C.NIR_RET = '0' AND C.NAI_RET = '0' AND C.SEX_RET = '0' AND C.SEJ_RET = '0' AND
C.FHO_RET = '0' AND C.PMS_RET = '0'
```

```
/* Variable disponible depuis 2006 */
AND C.DAT_RET = '0'
```

```
/* Variables disponibles depuis 2013 */
AND C.COH_NAI_RET = '0' AND C.COH_SEX_RET = '0' ;
```

Appliquer des filtres dans son extraction (données du PMSI) (slide 32)

Activité externe : pour travailler sur l'activité externe des établissements de soins ex-DG en MCO, il est recommandé de

MCO

- Exclure les **FINESS géographiques APHP, APHM et HCL** dont les remontées sont en doublons sur leur FINESS juridique avant 2017 inclus via la table [T_MCOaaCSTC](#) ;
- Exclure les **séjours avec des clés de chaînage incorrectes** sur les informations des bénéficiaires via les variables codes retours de la table [T_MCOaaCSTC](#).

```
SELECT A.NIR_ANO_17, A.ETA_NUM, A.SEQ_NUM, A.EXE_SOI_DTD, A.EXE_SOI_DTF
FROM T_MCOaaCSTC AS A
```

```
/* Exclusion des FINESS géographiques APHP, APHM et HCL en doublons entre 2005 et 2017 */
WHERE A.ETA_NUM NOT IN ('130780521', '130783236', '130783293', '130784234',
'130804297','600100101', '750041543', '750100018', '750100042', '750100075', '750100083',
'750100091', '750100109', '750100125', '750100166', '750100208', '750100216', '750100232',
'750100273', '750100299', '750801441', '750803447', '750803454', '910100015', '910100023',
'920100013', '920100021', '920100039', '920100047', '920100054', '920100062', '930100011',
'930100037', '930100045', '940100027', '940100035', '940100043', '940100050', '940100068',
'950100016', '690783154', '690784137', '690784152', '690784178', '690787478', '830100558')
```

```
/* Exclusion des clés de chaînage incorrectes sur les informations des bénéficiaires */
```

```
/* Variables disponibles depuis 2008 */
AND A.NIR_RET = '0' AND A.NAI_RET = '0' AND A.SEX_RET = '0' AND A.IAS_RET = '0'
```

```
/* Variable disponible depuis 2009 */
AND A.ENT_DAT_RET = '0' ;
```

Appliquer des filtres dans son extraction (données du PMSI) (slide 33)

Séjours/séquences : Il est recommandé d'appliquer systématiquement plusieurs filtres

HAD

- Exclure les sous-séquences qui ne seront pas valorisées (i.e. les sous-séquences avec une erreur de groupage) via la table T_HADaaGRP : WHERE GHT_NUM <> '99';
- Exclure les séjours associés à des clés de chaînage incorrectes sur les informations des bénéficiaires via les variables codes retours de la table T_HADaaC :
 - Variables disponibles depuis 2005 : WHERE NIR_RET = '0' AND NAI_RET = '0' AND SEX_RET = '0' AND SEJ_RET = '0' AND FHO_RET = '0' AND PMS_RET = '0' AND DAT_RET = '0';
 - Variables disponibles depuis 2013 : AND COH_NAI_RET = '0' AND COH_SEX_RET = '0';
- Dans ce champ d'activité, les **ES APHP, APHM et HCL** n'ont pas remonté leur activité en double sur leur FINESS géographique.

```

/*****/
/* Macro permettant d'exclure les sous-séquences qui ne seront pas valorisées, i.e. les
sous-séquences présentant une erreur de groupage, à partir de la table T_HADaaGRP */
/*****/

/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/..."; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro NETTOYAGE_T_HAD_AA_GRP(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
  delete tables_had_grp_nettoyee;
quit;

/* Nombre d'annees a traiter */
%let n = %sysfunc(countw(&annees));

/* Boucle sur les anneess */
%do i = 1 %to &n;

  /* Extraction de l'annee */
  %let annee = %scan(&annees, &i);
  %let an = %substr(&annee, 3, 2);

  %put NOTE: === Traitement &annee ===;

  /* Nom de la table de travail */
  %let table_work = table_had_grp&annee;

```



```

/* Suppression de la table de travail si elle existe */
proc datasets lib=work nolist;
    delete &table_work;
quit;

/* Creation ou ajout dans la table WORK.table_had_grpYYYY */
proc sql;
    %if %sysfunc(exist(work.&table_work)) = 0 %then %do;
        create table work.&table_work as
    %end;
    %else %do;
        insert into work.&table_work
    %end;

    select *
    from LIB_IN.T_HAD&an.GRP t1
    where t1.GHT_NUM <> '99'
    ;
quit;

/* Ajout dans la table finale LIB_OUT.tables_had_grp_nettoyee */
%if %sysfunc(exist(LIB_OUT.tables_had_grp_nettoyee)) = 0 %then %do;

    data LIB_OUT.tables_had_grp_nettoyee;
        set work.&table_work;
    run;

%end;
%else %do;

    /* FUSION DES TABLES */
    data LIB_OUT.tables_had_grp_nettoyee;
        set LIB_OUT.tables_had_grp_nettoyee work.&table_work;
    run;

%end;

/* Suppression de la table de travail */
proc datasets lib=work nolist;
    delete &table_work;
quit;

%end;

%mend NETTOYAGE_T_HAD_AA_GRP;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/* ===== */

%NETTOYAGE_T_HAD_AA_GRP(annees=2016 2017);

```

```

/*****/
/* Macro permettant d'exclure les séjours associés à des clés de chaînage incorrectes sur les
informations des bénéficiaires, identifiées à partir des variables de codes retour de la table
T_HADaaC */
/*****/

/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/... "; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro NETTOYAGE_T_HAD_AA_C(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
  delete tables_had_c_nettoyeur;
quit;

/* Nombre d'annees a traiter */
%let n = %sysfunc(countw(&annees));

/* Boucle sur les anneess */
%do i = 1 %to &n;

  /* Recuperation de l'annee */
  %let annee = %scan(&annees, &i);
  %let an = %substr(&annee, 3, 2);

  %put NOTE: === Traitement &annee ===;

  /* Nom de la table de travail */
  %let table_work = table_had_c&annee;

  /* Suppression de la table de travail si elle existe */
  proc datasets lib=work nolist;
    delete &table_work;
  quit;

  /* ----- CREATION DU FILTRE ----- */

  /* Filtres valides pour toutes les anneess depuis 2005 */
  %let filtres =
    NIR_RET = '0' AND
    NAI_RET = '0' AND
    SEX_RET = '0' AND
    SEJ_RET = '0' AND

```

```

FHO_RET = '0' AND
PMS_RET = '0' AND
DAT_RET = '0';

/* Filtrés à appliquer sur des variables à partir de 2013 */
%if &annee. >= 2013 %then %do;
    %let filtres = &filtres AND COH_NAI_RET = '0' AND COH_SEX_RET = '0';
%end;

/* ----- CREATION DE LA TABLE ----- */

proc sql;
    create table work.&table_work as
    select
        &annee as annee_ext,
        t1.*
    from LIB_IN.T_HAD&an.C t1
    where &filtres
    ;
quit;

/* Création de la table finale */
%if not %sysfunc(exist(LIB_OUT.tables_had_c_nettoyee)) %then %do;

    data LIB_OUT.tables_had_c_nettoyee;
        set work.&table_work;
    run;

%end;

/* Ajout des tables */
%else %do;

    data LIB_OUT.tables_had_c_nettoyee;
        set LIB_OUT.tables_had_c_nettoyee work.&table_work;
    run;

%end;

/* Suppression de la table temporaire */
proc datasets lib=work nolist;
    delete &table_work;
quit;

%end;

/* Fin macro */
%mend NETTOYAGE_T_HAD_AA_C;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/*===== */

```

```
%NETTOYAGE_T_HAD_AA_C(annees=2012 2013);
```

Activité externe : pour travailler sur l'activité externe (disponible avec le chaînage des bénéficiaires à partir de 2013), il est recommandé de :

SMR

- Exclure les ACE associés à des clés de chaînage incorrectes sur les informations des bénéficiaires via les variables codes retour de la table T_SSRaaCSTC : WHERE NIR_RET = '0' AND NAL_RET = '0' AND SEX_RET = '0' AND IAS_RET = '0' AND ENT_DAT_RET = '0';
- Dans ce champ d'activité, les ES APHP, APHM et HCL n'ont pas remonté leur activité en double sur leur FINESS géographique.

```

/*****
/* Macro permettant d'exclure les ACE associés à des clés de chaînage incorrectes sur les
informations des bénéficiaires, identifiées via les variables de codes retour de la table
T_SSRaaCSTC */
*****/
/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/..."; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro NETTOYAGE_T_SSR_AA_CSTC(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
  delete tables_smr_cstc_nettoyee;
quit;

/* Nombre d'annees a traiter */
%let n = %sysfunc(countw(&annees));

/* Boucle sur les anneess */
%do i = 1 %to &n;

  /* Recuperation de l'annee */
  %let annee = %scan(&annees, &i);
  %let an = %substr(&annee, 3, 2);

  %if &annee < 2013 %then %do;
    %put ERROR: "Les filtres ne s'appliquent qu'a partir de 2013.";
    %return;
  %end;

  %put NOTE: === Traitement &annee ===;

```

```

/* Nom de la table de travail */
%let table_work = table_smr_cstc&annee;

/* Suppression de la table de travail si elle existe */
proc datasets lib=work nolist;
    delete &table_work;
quit;

/* ----- CREATION DE LA TABLE ----- */

proc sql;
    create table work.&table_work as
    select
        &annee as annee_ext,
        t1.*
    from LIB_IN.T_SSR&an.CSTC t1
        where NIR_RET = '0' AND NAI_RET = '0' AND SEX_RET = '0' AND IAS_RET = '0' AND
ENT_DAT_RET = '0'
    ;
quit;

/* Création de la table finale */
%if not %sysfunc(exist(LIB_OUT.tables_smr_cstc_nettoyee)) %then %do;

    data LIB_OUT.tables_smr_cstc_nettoyee;
        set work.&table_work;
    run;

%end;

/* Ajout des tables*/
%else %do;

    data LIB_OUT.tables_smr_cstc_nettoyee;
        set LIB_OUT.tables_smr_cstc_nettoyee work.&table_work;
    run;

%end;

/* Suppression de la table temporaire */
proc datasets lib=work nolist;
    delete &table_work;
quit;

%end;

/* Fin macro */
%mend NETTOYAGE_T_SSR_AA_CSTC;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/*===== */

```

```
%NETTOYAGE_T_SSR_AA_CSTC(annees=2013 2014);
```

Appliquer des filtres dans son extraction (données du PMSI) (slide 34)

Séjours : il est recommandé d'appliquer systématiquement plusieurs filtres

SMR

- Exclure les résumés hebdomadaires de sortie anonymes (RHA)** qui ne seront pas valorisés, (i.e. avec une erreur de groupage) via la table T_SSRaaB
 - Variable disponible depuis 2013 : `WHERE GRG_GME NOT LIKE '90%'`;
- Exclure les séjours ou parties de séjours** qui ne seront pas valorisés (i.e. avec une erreur de groupage) via la table T_SSRaaGME
 - Variable disponible depuis 2012 : `WHERE GME_COD NOT LIKE '90%'`;
 - et après 2012 : `WHERE GME NOT LIKE '90%'`;
- Exclure les « faux » RHA générés automatiquement** pour les besoins de facturation (déjà exclus via les filtres précédents) via la table T_SSRaaB
 - Variable disponible depuis 2015 : `WHERE TYP_GEN_RHA IN ('0', '4')`;
- Exclure les RHA de l'année précédente** (i.e. RHA répétés dans le PMSI de l'année N pour les séjours non clos et non valorisés en N-1) via la table T_SSRaaB : `WHERE RIGHT(MOI_ANN, 4) = annee`;
- Exclure les séjours associés à des clés de chaînage incorrectes sur les informations des bénéficiaires** via les variables codes retours de la table T_SSRaaC :
 - Variables disponibles depuis 2005 : `WHERE NIR_RET = '0' AND NAL_RET = '0' AND SEX_RET = '0' AND SEJ_RET = '0' AND FHO_RET = '0' AND PMS_RET = '0' AND DAT_RET = '0'`;
 - Variables disponibles depuis 2013 : `AND COH_NAL_RET = '0' AND COH_SEX_RET = '0'`;
- Dans ce champ d'activité, les **ES APHP, APHM et HCL n'ont pas remonté leur activité en double sur leur FINESS géographique.**

```

/*****
/*
Macro permettant, à partir de la table T_SSRaaB, d'exclure :
- les RHA non valorisables, i.e. présentant une erreur de groupage ;
- les « faux » RHA générés automatiquement pour les besoins de facturation ;
- les RHA de l'année précédente, i.e. répétés dans le PMSI de l'année N pour des séjours non clos
et non valorisés en N-1.
*/
*****/

/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/..."; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro nettoyage_t_ssr_aa_b(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
  delete tables_smr_b_nettoyee;
quit;

/* Nombre d'annees a traiter */
%let n = %sysfunc(countw(&annees));

```

```

/* Boucle sur les annees */
%do i = 1 %to &n;

    /* Recuperation de l'annee */
    %let annee = %scan(&annees, &i);
    %let an = %substr(&annee, 3, 2);

    %put NOTE: === Traitement &annee ===;

    /* Nom de la table de travail */
    %let table_work = table_smr_b&annee;

    /* Suppression de la table de travail si elle existe */
    proc datasets lib=work nolist;
        delete &table_work;
    quit;

    /* ----- CREATION DE LA TABLE ----- */

    proc sql;
        create table work.&table_work as
        select
            &annee as annee_ext,
            t1.*
        from LIB_IN.T_SSR&an.B t1
        where substr(MOI_ANN, 3, 4) = "&annee."

            %if &annee >= 2013 %then %do;
                AND GRG_GME NOT LIKE '90%'
            %end;

            %if &annee >= 2015 %then %do;
                AND TYP_GEN_RHA IN ('0','4')
            %end;
        ;
    quit;

    /* Création de la table finale */
    %if not %sysfunc(exist(LIB_OUT.tables_smr_b_nettoyee)) %then %do;

        data LIB_OUT.tables_smr_b_nettoyee;
            set work.&table_work;
        run;

    %end;

    /* Ajout des tables */
    %else %do;

        data LIB_OUT.tables_smr_b_nettoyee;
            set LIB_OUT.tables_smr_b_nettoyee work.&table_work;
        run;

    %end;

```



```

/* Suppression de la table temporaire */
proc datasets lib=work nolist;
  delete &table_work;
quit;

%end;

/* Fin macro */
%mend nettoyage_t_ssr_aa_b;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/*===== */

%nettoyage_t_ssr_aa_b(annees=2013);



---



/*****/
/* Macro permettant d'exclure les séjours ou parties de séjours non valorisables, i.e. présentant
une erreur de groupage, à partir de la table T_SSRaaGME */
/*****/

/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/..."; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro nettoyage_t_ssr_aa_gme(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
  delete tables_smr_gme_nettoyees;
quit;

/* Nombre d'années à traiter */
%let n = %sysfunc(countw(&annees));

/* Boucle sur les années */
%do i = 1 %to &n;

  /* Recuperation de l'année */
  %let annee = %scan(&annees, &i);
  %let an = %substr(&annee, 3, 2);

```

```

        %if &annee < 2012 %then %do;
            %put ERROR: "Les filtres sur les tables T_SSRaaGME ne s'appliquent qu'a partir de
2012.";
            %return;
        %end;

        %put NOTE: === Traitement &annee ===;

        /* Nom de la table de travail */
        %let table_work = table_smr_gme&annee;

        /* Suppression de la table de travail si elle existe */
        proc datasets lib=work nolist;
            delete &table_work;
        quit;

        /* ----- CREATION DE LA TABLE ----- */
        /* Changement dans le nom de la variable GME entre 2012 et 2013 */
        %let var_gme = %sysfunc(ifc(&annee = 2012, GME, GME_COD));
        proc sql;
            create table work.&table_work as
            select
                &annee as annee_ext,
                t1.*
            from LIB_IN.T_SSR&an.GME t1
            where &var_gme not like '90%';
        quit;

        /* Creation de la table finale */
        %if not %sysfunc(exist(LIB_OUT.tables_smr_gme_nettoyees)) %then %do;

            data LIB_OUT.tables_smr_gme_nettoyees;
                set work.&table_work;
            run;

        %end;

        /* Ajout des tables*/
        %else %do;

            data LIB_OUT.tables_smr_gme_nettoyees;
                set LIB_OUT.tables_smr_gme_nettoyees work.&table_work;
            run;

        %end;

        /* Suppression de la table temporaire */
        proc datasets lib=work nolist;
            delete &table_work;
        quit;

    %end;

    /* Fin macro */

```

```
%mend nettoyage_t_ssr_aa_gme;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/*===== */

%nettoyage_t_ssr_aa_gme(annees=2012);



---


/*****/
/* Macro permettant d'exclure les séjours associés à des clés de chaînage incorrectes sur les
informations des bénéficiaires, identifiées via les variables de codes retour de la table T_SSRaaC */
/*****/

/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/..."; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro NETTOYAGE_T_SSR_AA_C(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
  delete tables_ssr_c_nettoyees;
quit;

/* Nombre d'années à traiter */
%let n = %sysfunc(countw(&annees));

/* Boucle sur les annees */
%do i = 1 %to &n;

  /* Recuperation de l'année */
  %let annee = %scan(&annees, &i);
  %let an = %substr(&annee, 3, 2);

  %put NOTE: === Traitement &annee ===;

  /* Nom de la table de travail */
  %let table_work = table_ssr_c&annee;

  /* Suppression de la table de travail si elle existe */
  proc datasets lib=work nolist;
    delete &table_work;
```

```
quit;

/* ----- CREATION DU FILTRE ----- */

/* Filtres valides pour toutes les années depuis 2005 */
%let filtres =
  NIR_RET = '0' AND
  NAI_RET = '0' AND
  SEX_RET = '0' AND
  SEJ_RET = '0' AND
  FHO_RET = '0' AND
  PMS_RET = '0' AND
  DAT_RET = '0';

/* Filtres à appliquer sur des variables à partir de 2013 */
%if &annee. >= 2013 %then %do;
  %let filtres = &filtres AND COH_NAI_RET = '0' AND COH_SEX_RET = '0';
%end;

/* ----- CREATION DE LA TABLE ----- */

proc sql;
  create table work.&table_work as
  select
    &annee as annee_ext,
    t1.*
  from LIB_IN.T_SSR&an.C t1
  where &filtres
  ;
quit;

/* Creation de la table finale */
%if not %sysfunc(exist(LIB_OUT.tables_ssr_c_nettoyees)) %then %do;

  data LIB_OUT.tables_ssr_c_nettoyees;
    set work.&table_work;
  run;

%end;

/* Ajout des tables */
%else %do;

  data LIB_OUT.tables_ssr_c_nettoyees;
    set LIB_OUT.tables_ssr_c_nettoyees work.&table_work;
  run;

%end;

/* Suppression de la table temporaire */
proc datasets lib=work nolist;
  delete &table_work;
quit;
```

```
%end;

/* Fin macro */
%mend NETTOYAGE_T_SSR_AA_C;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/*===== */

%NETTOYAGE_T_SSR_AA_C(annees=2012 2013);
```

Appliquer des filtres dans son extraction (données du PMSI) (slide 35)

Séjours : il est recommandé
d'appliquer systématiquement
plusieurs filtres

RIM-P

- Exclure les séquences indiquées comme « sortie d'essai » jusqu'en 2016 car elles ne sont pas considérées comme des hospitalisations via la table T_RIPaaRSA : `WHERE SEQ_IND <> 'E'` ;
- Exclure les « faux » résumés de sortie anonymes (RSA) générés automatiquement pour les besoins de facturation via la table T_RIPaaRSA
 - Variable disponible depuis 2015 : `WHERE TYP_GEN_RSA = '0'` ;
- Exclure les séjours associés à des clés de chaînage incorrectes sur les informations des bénéficiaires via les variables codes retours de la table T_RIPaaC :
 - Variables disponibles depuis 2007 : `WHERE NIR_RET = '0' AND NAL_RET = '0' AND SEX_RET = '0' AND SEJ_RET = '0' AND FHO_RET = '0' AND PMS_RET = '0' AND DAT_RET = '0'` ;
 - Variables disponibles depuis 2013 : `AND COH_NAL_RET = '0' AND COH_SEX_RET = '0'` ;
- Dans ce champ d'activité, les ES APHP, APHM et HCL n'ont pas remonté leur activité en double sur leur FINESS géographique.

```

/*****/
/* Macro permettant, à partir de la table T_RIPaaRSA, d'exclure :
- les séquences indiquées comme « sortie d'essai » jusqu'en 2016, car elles ne sont pas
  considérées comme des hospitalisations ;
- les « faux » résumés de sortie anonymes (RSA) générés automatiquement pour les besoins de
  facturation. */
/*****/

/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
  ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/..."; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
  Vous pouvez copier-coller ce code et appeler directement la macro
  pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro nettoyage_t_rip_aa_RSA(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
  delete tables_rip_rsa_nettoyes;
quit;

/* Nombre d'années à traiter */
%let n = %sysfunc(countw(&annees));

/* Boucle sur les années */
%do i = 1 %to &n;

  /* Récupération de l'année */
  %let annee = %scan(&annees, &i);

```

```

%let an = %substr(&annee, 3, 2);

%if &annee < 2007 %then %do;
    %put ERROR: "Les filtres ne s'appliquent qu'a partir de 2007.";
    %return;
%end;

%put NOTE: === Traitement &annee ===;

/* Nom de la table de travail */
%let table_work = table_rip_rsa&annee;

/* Suppression de la table de travail si elle existe */
proc datasets lib=work nolist;
    delete &table_work;
quit;

/* ----- CREATION DU FILTRE ----- */
%let filtres = ;

%if &annee. < 2016 %then %do;
%let filtres = SEQ_IND <> 'E';
%end;
%else %if (&annee >= 2015 and &annee <= 2016) %then %do;
    %let filtres = SEQ_IND <> 'E' AND TYP_GEN_RSA = '0';
%end;
%else %if &annee. > 2016 %then %do;
    %let filtres = TYP_GEN_RSA = '0';
%end;

/* ----- CREATION DE LA TABLE ----- */
proc sql;
    create table work.&table_work as
    select
        &annee as annee_ext,
        t1.*
    from LIB_IN.T_RIP&an.RSA t1
    where &filtres;
quit;

/* Creation de la table finale */
%if not %sysfunc(exist(LIB_OUT.tables_rip_rsa_nettoyees)) %then %do;

    data LIB_OUT.tables_rip_rsa_nettoyees;
        set work.&table_work;
    run;

%end;

/* Ajout des tables */
%else %do;

    data LIB_OUT.tables_rip_rsa_nettoyees;
        set LIB_OUT.tables_rip_rsa_nettoyees work.&table_work;
    run;

```

```

%end;

/* Suppression de la table temporaire */
proc datasets lib=work nolist;
    delete &table_work;
quit;

%end;

/* Fin macro */
%mend nettoyage_t_rip_aa_RSA;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/*===== */

%nettoyage_t_rip_aa_RSA(annees=2015);



---


/*****/
/* Macro permettant d'exclure les séjours associés à des clés de chaînage incorrectes sur les
informations des bénéficiaires, identifiées via les variables de codes retour de la table T_RIPaaC */
/*****/

/* Veuillez préciser le chemin des librairies contenant les tables à filtrer,
ainsi que la librairie destinée à stocker les tables filtrées. */
LIBNAME LIB_IN "%sysget(HOME)/sasdata/..."; /* Librairie des tables d'extraction */
LIBNAME LIB_OUT "%sysget(HOME)/sasdata/..."; /* Chemin pour le stockage de la table des
résultats */

/* Veuillez ne pas modifier le corps de la macro (1).
Vous pouvez copier-coller ce code et appeler directement la macro
pour une table donnée, ou dans une boucle. */

/*===== */
/* =====(1) Définition de la macro ===== */
/*===== */

%macro NETTOYAGE_T_RIP_AA_C(annees=);

/* Suppression de la table finale si elle existe déjà */
proc datasets lib=LIB_OUT nolist;
    delete tables_rip_c_nettoyes;
quit;

/* Nombre d'annees a traiter */
%let n = %sysfunc(countw(&annees));

/* Boucle sur les annees */
%do i = 1 %to &n;

    /* Recuperation de l'annee */
    %let annee = %scan(&annees, &i);

```



```

%let an = %substr(&annee, 3, 2);

    %if &annee < 2007 %then %do;
        %put ERROR: "Le filtres ne s'appliquent qu'a partir de 2007.";
        %return;
    %end;

%put NOTE: === Traitement &annee ===;

/* Nom de la table de travail */
%let table_work = table_rip_c&annee;

/* Suppression de la table de travail si elle existe */
proc datasets lib=work nolist;
    delete &table_work;
quit;

/* ----- CREATION DU FILTRE ----- */

/* Filtres valides pour toutes les annees depuis 2005 */
%let filtres =
    NIR_RET = '0' AND
    NAI_RET = '0' AND
    SEX_RET = '0' AND
    SEJ_RET = '0' AND
    FHO_RET = '0' AND
    PMS_RET = '0' AND
    DAT_RET = '0';

/* Filtres à appliquer sur des variables a partir de 2013 */
%if &annee. >= 2013 %then %do;
    %let filtres = &filtres AND COH_NAI_RET = '0' AND COH_SEX_RET = '0';
%end;

/* ----- CREATION DE LA TABLE ----- */

proc sql;
    create table work.&table_work as
    select
        &annee as annee_ext,
        t1.*
    from LIB_IN.T_RIP&an.C t1
    where &filtres
    ;
quit;

/* Creation de la table finale */
%if not %sysfunc(exist(LIB_OUT.tables_rip_c_nettoyees)) %then %do;

    data LIB_OUT.tables_rip_c_nettoyees;
        set work.&table_work;
    run;

%end;

```

```

/* Ajout des tables*/
%else %do;

    data LIB_OUT.tables_rip_c_nettoyees;
        set LIB_OUT.tables_rip_c_nettoyees work.&table_work;
    run;

%end;

/* Suppression de la table temporaire */
proc datasets lib=work nolist;
    delete &table_work;
quit;

%end;

/* Fin macro */
%mend NETTOYAGE_T_RIP_AA_C;

/*===== */
/* =====(2) Exemple d'appel de la macro ===== */
/*===== */

%NETTOYAGE_T_RIP_AA_C(annees=2022);

```