**REQUETE TYPE DANS LA TABLE DES PRESTATIONS DU DCIR**

|  |  |
| --- | --- |
| **Objectif** | Décrire une requête type de sélection de prestations impliquant une table d'actes affinés de codage avec sélection de données concernant les bénéficiaires.**Avertissement**: le programme de sélection des prestations sélectionne des remboursements de médicaments et devra être adapté en fonction des prestations d’intérêt. |
| **Création** | Direction DATA de Santé publique France |
| **Date de création** | 30/04/2018 |
| **Date de dernière mise-à-jour** | 30/04/2018 |

Sommaire

[1. Identification des tables et des variables 2](#_Toc513480008)

[2. Recommandations pour les requêtes 4](#_Toc513480009)

[3. Programmes Type 5](#_Toc513480010)

[3.1. Sélection des prestations : exemple des remboursements de médicaments 5](#_Toc513480011)

[3.2. Sélection de données sociodémographiques dans les référentiels des bénéficiaires 13](#_Toc513480012)

1. Identification des tables et des variables

Tables d’actes affinés de codage disponibles

-**ER\_PHA\_F** (médicaments)

-**ER\_BIO\_F** (actes de biologie)

-**ER\_CAM\_F** (actes médicaux)

-**ER\_UCD\_F** (médicaments rétrocédés + en sus du GHS)

-**ER\_LOT\_F** (lots de medicaments)

-**ER\_TIP\_F** (dispositifs médicaux)

Table des prestations

**ER\_PRS\_F**

Pseudo référentiel

**IR\_BEN\_R**

*2 clés de jointure*

*9 clés de jointure*

* La table des prestations (**ER\_PRS\_F**) est centrale dans le DCIR. Les variables généralement sélectionnées dans cette table sont :
	+ Nature de la prestation (*PRS\_NAT\_REF*)
	+ Dates (par exemple dates de début (*EXE\_SOI\_DTD*) et de fin de prestation (*EXE\_SOI\_DTF*))
	+ variables concernant le PS exécutant ou prescripteur (par exemple spécialité médicale : *PSE\_SPE\_COD* (PS exécutant) et *PSP\_SPE\_COD* (PS prescripteur))
	+ CMU (Couverture Maladie Universelle Complémentaire) au moment de la prestation (*BEN\_CMU\_TOP*)
	+ Régime d’affiliation à l’Assurance Maladie au moment de la prestation (généré à partir d’ORG\_AFF\_BEN)
	+ Département ou commune de résidence au moment de la prestation (BEN\_RES\_DPT et *BEN\_RES\_COM*)

NB : la sélection des prestations peut se faire sur une valeur particulière de *PRS\_NAT\_REF* (nature de la prestation) comme par exemple ‘visite chez un médecin généraliste’ et dans ce cas la jointure s’effectue sur deux tables seulement (**ER\_PRS\_F** et **IR\_BEN\_R**).

* De la table du pseudo-référentiel des bénéficiaires (**IR\_BEN\_R**), on retient généralement les variables socio-démographiques (la table contient les informations les plus récentes) et l’identifiant unique *BEN\_NIR\_ANO* en plus des 2 clés de jointure (*BEN\_NIR\_PSA* et *BEN\_RNG\_GEM*).

Cependant, on peut garder uniquement les variables de jointure (*BEN\_NIR\_PSA* et *BEN\_RNG\_GEM*) et conserver les variables socio-démographiques associées à des prestations déjà sélectionnées (cf programme type joint) ou récupérer les variables socio-démographiques de l’année 20aa dans la table **EXTRACTION\_PATIENTS20aaTR** du répertoire CONSOPAT.

* La table d’actes affinés de codage précise la nature des remboursements liés à la prestation, comme par exemple la liste des médicaments liée à la prestation(**ER\_PHA\_F**) ou les actes médicaux (**ER\_CAM\_F**). Le document « [DCIR ce qu’il faut savoir](https://intranet_vosespaces.ansp.local/ds/bddt/SNIIRAM1/_Informations%20g%C3%A9n%C3%A9rales_/2015_ce_quil_faut_savoir_DCIR_avril2016.docx)» décrit ces tables en détail.

De la table des prestations affinées, on ne retient généralement que le code affiné et la quantité correspondante

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nom de la table**  | **Code affiné** | **Quantité** | **Remarque** | **Référentiel associé** |
| **ER\_PHA\_F** | *PHA\_PRS\_C13* | *PHA\_ACT\_QSN* |   | IR\_PHA\_R |
| **ER\_BIO\_F** | *BIO\_PRS\_IDE* | *BIO\_ACT\_QSN* |  | IR\_BIO\_R |
| **ER\_CAM\_F** | *CAM\_PRS\_IDE* |  | Il n’y a pas de variable quantité dans la table ER\_CAM\_F. Prendre PRS\_ACT\_QTE de la table ER\_PRS\_F.Pensez à prendre en compte CAM\_ACT\_COD (code activité) + CAM\_TRT\_PHA (code phase de traitement). | - |
| **ER\_UCD\_F** | *UCD\_UCD\_COD* | *UCD\_DLV\_NBR* |  | - |
| **ER\_TIP\_F** | *TIP\_PRS\_IDE* | *TIP\_ACT\_QSN* |  | NT\_LPP |

La sélection des codes affinés peut se faire directement dans la requête en listant le ou les codes souhaités ou en fusionnant la table de prestations affinées avec un référentiel pour effectuer une sélection sur un groupe plus important de codes.

Par exemple : fusionner la table de prestations affinées de pharmacie : ER\_PHA\_R avec le référentiel pharmacie IR\_PHA\_R afin de faire une sélection sur un code ATC particulier regroupant plusieurs codes.

ER\_PHA\_F

ER\_PRS\_F

IR\_BEN\_R

IR\_PHA\_R

1. Recommandations pour les requêtes
* La fusion des tables se fait par une procédure SQL.
* Il est recommandé de ne garder que les variables nécessaires.
* Ne pas sélectionner systématiquement les 9 variables constituant les clés de jointure entre la table **ER\_PRS\_F** et les tables affinées,

Ne pas systématiquement sélectionner en double les variables en fonction de leur provenance. (par exemple : variable *BEN\_SEX\_COD* dans **IR\_BEN\_R** et **ER\_PRS\_F**)

* Lorsque l’on veut sélectionner des visites ou des consultations dans **ER\_PRS\_F**, il faut supprimer les enregistrements correspondant à des majorations et des compléments en utilisant le filtre *CPL\_MAJ\_TOP* < 2.
* Pour le moment, l’information sur les soins externes réalisés en établissements publics n’est pas exhaustive, et sauf cas particulier, il convient d’appliquer un filtre pour les exclure (*DPN\_QLF* = 71 AND *PRS\_DPN\_QLP* = 71)
* Les tables du DCIR sont des tables volumineuses. Les prestations alimentent le DCIR par mois de flux et non en fonction de la date d’exécution de la prestation de soins. Pour ces deux raisons, la requête peut être exécutée par mois de flux et, si l’on veut les prestations sur une année complète, la requête doit être répétée 18 fois (année entière plus les 6 mois suivants). En procédant ainsi, on connait par exemple plus de 98% des délivrances de médicaments de l’année portées au remboursement.
* Prise en compte des régularisations.

Dans les tables de prestations ou d’actes affinés de codage il existe des lignes de régularisation. Elles correspondent à des annulations de prestations avec ou non enregistrement d’une nouvelle ligne en remplacement de la ligne annulée. La ligne de régularisation est en tout point identique à la ligne initiale mais avec une quantité d’actes négative. En pratique, il faut donc sommer les quantités des lignes identiques de prestations et ne garder que les lignes dont la somme est strictement positive.

1. Programmes Type
	1. Sélection des prestations : exemple des remboursements de médicaments

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\* SELECTION DES REMBOURSEMENTS DE MEDICAMENTS \*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\* DANS DCIR ARCHIVE ET DCIR \*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* DONNEES en entrée

 oravue.er\_prs\_f\_&an ou oravue.er\_prs\_f

 oravue.er\_pha\_f\_&an ou oravue.er\_pha\_f

 oravue.IR\_PHA\_R

 &table\_population en entrée (facultatif)

/\* DONNEES en sortie

 &table\_sortie à définir

/\* PLAN du PROGRAMME ;

 - Définition par l'utilisateur des paramètres suivants:

 \* Choix de la librairie de données en sortie

 \* options obs=0 ou MAX (respectivement pour tester le

 programme ou lancer la sélection)

 \* choix des codes de médicaments

 \* liste des variables à sélectionner

 \* table de population à croiser (facultatif)

 - Macro outils

 - Sélection des médicaments

 - Copie de la table de population (facultatif) dans ORAUSER

 - Macro de sélection des médicaments pour un mois de flux :

 CREA\_TABLE

 - Macro de compilation des mois de flux : COMPILATION

 - Macro de suppression des régularisations : REGULARISATIONS

 - Macro TAB\_MEDICAMENTS qui exécute les macros CREA\_TABLE et

 COMPILATION pour chaque mois de flux et REGULARISATION sur la

 table finale

 - L'utilisateur lance la Macro TAB\_MEDICAMENTS en choisissant

 \* le mois et année de début d'exécution des soins (format aaaamm)

 \* le mois et année de fin d'exécution des soins (format aaaamm)

 \* le nom de la table de sortie

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\* Paramètres à changer : répertoire, liste des variables, des codes \*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*déclaration de la librairie \*/

/\* exemple : sous répertoire PGM\_tmt de l’espace partage ESP\_PART\*/

%LET fichier = %SYSGET (HOME)/sasdata ;

LIBNAME ma\_lib "&fichier/sasdata/ESP\_PART/PGM\_tmt" ;

OPTIONS OBS =0 ; /\*pour tester la requête ou OPTIONS OBS = MAX pour l’exécuter sur les

 données \*/

/\*Critères de sélection des médicaments à extraire du référentiel pharmacie (cf communiqué du 24 janvier 2012)\*/

%LET codes = pha\_cip\_c13 in ( 3400933226558, 3400934744198, 3400926939939 ) ;

/\* EXEMPLE 1 : Sélection par CIP 13

 pha\_cip\_c13 in (3400933226558,3400934744198,3400926939939)

 EXEMPLE 2 : Sélection par CIP 7

 pha\_prs\_ide in ( 3876282, 3517778, 3383272 )

 EXEMPLE 3 : Sélection par code ATC

 pha\_atc\_c03 like 'A10' and not(pha\_atc\_c07='A10BX06')

 EXEMPLE 4 : Sélection par principe actif

 pha\_nom\_pa like %METFORMINE%

 exemple 5 : Sélection par CIP 7 ou ATC 7\*

 pha\_prs\_ide in ( 3876282, 3517778, 3383272 ) or pha\_atc\_C07 in ("J01AA02", "J01FA01" ,"J01MA01") \*/

/\*liste des variables à conserver **(Ne pas changer les préfixes p., pha., ir.)**\*/

/\* variables de la table des prestations\*/

%LET liste\_var\_prs =

 p.ben\_nir\_psa,

 p.ben\_rng\_gem,

 p.exe\_soi\_dtd,

 p.ben\_res\_dpt,

 p.org\_aff\_ben,

 p.pse\_spe\_cod,

 p.psp\_spe\_cod

 /\*et autres variables selon le besoin\*/;

/\* variables de la table d’actes affinée = ici pharmacie ER\_PHA\_F\*/

%LET liste\_var\_pha =

 pha.pha\_prs\_c13 ,

 pha.pha\_prs\_ide

 /\*et autres variables selon le besoin\*/;

/\* variables du référentiel = ici référenciel pharmacie IR\_PHA\_R\*/

%LET liste\_var\_ir =

 ir.pha\_atc\_c03,

 ir.pha\_atc\_c07

 /\*et autres variables selon le besoin\*/;

/\*1ère année de la table ER\_PRS\_F\*/

%LET an\_DCIR = 2013 ;

/\***Facultatif**: Pour extraire les remboursements de médicaments d'une population déjà constituée, et non de la population française entière

Attention: 1 ligne = 1 individu

Variables obligatoires : BEN\_NIR\_PSA, BEN\_RNG\_GEM

Si vous n'avez pas de table de population au départ: %let table\_pop= , sinon =nom de la table\*/

%LET table\_pop = ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\* MACRO OUTILS \*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 9 cles techniques \*/

%MACRO CLE\_TEC ( p, d ) ;

&p..DCT\_ORD\_NUM = &d..DCT\_ORD\_NUM and

&p..FLX\_DIS\_DTD = &d..FLX\_DIS\_DTD and

&p..FLX\_EMT\_NUM = &d..FLX\_EMT\_NUM and

&p..FLX\_EMT\_ORD = &d..FLX\_EMT\_ORD and

&p..FLX\_EMT\_TYP = &d..FLX\_EMT\_TYP and

&p..FLX\_TRT\_DTD = &d..FLX\_TRT\_DTD and

&p..ORG\_CLE\_NUM = &d..ORG\_CLE\_NUM and

&p..prs\_ORD\_NUM = &d..prs\_ORD\_NUM and

&p..REM\_TYP\_AFF = &d..REM\_TYP\_AFF

%MEND ;

/\* Lorsque l’on a besoin de fusionner une table personnelle avec une table oracle, il faut créer une table personnelle dans ORAUSER, avant cela il faut s’assurer qu’une table avec le même nom n’existe pas déjà ; le cas échéant il faut supprimer cette table existante \*/

%MACRO KILL\_ORAUSER ( table ) ;

%IF %SYSFUNC( exist( ORAUSER.&table )) %THEN %DO ;

 PROC SQL;

 DROP TABLE ORAUSER.&table;

 QUIT;

%END ;

%MEND ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\* CHOIX DES MEDICAMENTS DANS LE REFERENTIEL PHARMACIE \*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%KILL\_ORAUSER( liste\_medicaments ) ;

PROC SQL;

 CREATE TABLE ORAUSER.liste\_medicaments AS

 SELECT \* /\*ou variables à choisir\*/

 FROM ORAVUE.IR\_PHA\_R

 WHERE &codes

 ;

QUIT ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\* OPTION : CHOIX d'UNE TABLE de POPULATION \*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%MACRO table\_pop() ;

%IF &table\_pop ne %THEN %DO ;

 %PUT Table de population = &table\_pop. ;

 %KILL\_ORAUSER(pop) ;

 DATA orauser.pop ;

 SET &table\_pop ;

 RUN ;

 %MACRO croisement\_tab\_pop() ;

 inner join orauser.pop pop

 on pop.ben\_nir\_psa=p.ben\_nir\_psa and pop.ben\_rng\_gem=p.ben\_rng\_gem

 %MEND ;

%END ;

%ELSE %DO ;

 %PUT no\_table ;

 %MACRO croisement\_tab\_pop() ;

 %MEND ;

%END ;

%MEND ;

%table\_pop ;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\* SELECTION DES PRESTATIONS \*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\* Macro de création d'une table de prestations mensuelle \*\*\*/

%MACRO CREA\_TABLE ( FLX\_DIS\_DTD ) ;

%PUT ### Mois de flux ( &FLX\_DIS\_DTD ) Début %SYSFUNC( datetime(),datetime. ) ;

/\* Suppression de la table de prestations mensuelle si existante \*/

%IF %SYSFUNC ( exist( table\_mois )) %THEN %DO ;

 PROC DELETE DATA = table\_mois ;

 RUN ;

%END ;

PROC SQL;

CREATE TABLE table\_mois as

SELECT &liste\_var\_prs,

 &liste\_var\_pha,

 &liste\_var\_ir,

 pha.pha\_act\_qsn

FROM &table\_prs as p %croisement\_tab\_pop /\*ne fait rien si pas de table de population à

 croiser\*/

 INNER JOIN &table\_pha as pha

 ON %CLE\_TEC (p,pha)

 INNER JOIN orauser.liste\_medicaments as ir

 ON &CONDITION\_JOINTURE\_MEDICAMENTS

WHERE p.flx\_dis\_dtd ="&FLX\_DIS\_DTD:0:0:0"dt

AND (p.exe\_soi\_dtd between "&DATE\_EXE\_DEB:0:0:0"dt and "&DATE\_EXE\_FIN:0:0:0"dt ) /\* sélection de la période d’exécution du soin\*/

AND (p.DPN\_QLF <> 71 AND p.PRS\_DPN\_QLP <> 71) /\*exclusion des soins externes

 transmis pour info\*/

 AND (p.CPL\_MAJ\_TOP<2) ; /\*exclusion des majorations\*/

QUIT ;

%PUT ### CREA\_TABLE ( &FLX\_DIS\_DTD ) Fin %SYSFUNC( datetime(),datetime. ) ;

%MEND CREA\_TABLE ;

/\*\*\* Macro Compilation des tables mensuelles \*\*\*\*/

%MACRO COMPILATION() ;

%IF %SYSFUNC(exist( compil )) eq 0 %THEN %DO ;

proc datasets nolist memtype=data;

change table\_mois=compil;

quit;

%END ;

%ELSE %DO;

proc append base=compil data=table\_mois FORCE;run;

proc delete data=table\_mois;run;

%END;

%MEND COMPILATION ;

/\*\*\* Macro de suppression des régularisations \*\*\*/

%MACRO REGULARISATIONS( table\_sortie );

/\*liste des variables (hors variables quantité) sans les suffixes\*/

%let liste\_var\_regul=%qsysfunc(tranwrd(%quote(&liste\_var\_prs),p.,)) , %qsysfunc(tranwrd(%quote(&liste\_var\_pha),pha., )), %qsysfunc(tranwrd(%quote(&liste\_var\_ir),ir., ));

PROC SQL ;

CREATE TABLE &table\_sortie as

SELECT distinct

 &liste\_var\_regul,

 sum(pha\_act\_qsn) as pha\_qt

FROM compil

GROUP BY &liste\_var\_regul

HAVING CALCULATED pha\_qt>0 ;

QUIT ;

%MEND REGULARISATIONS;

/\*\*\* Exécution des macros précédentes pour chaque mois de flux\*\*\*/

%MACRO TAB\_MEDICAMENTS( Date\_DEB= ,Date\_FIN= ,table\_sortie= ) ;

/\*DEFINITION DES PARAMETRES\*/

%LET AN\_DEB = %SUBSTR( &DATE\_DEB,1,4 ) ;

%LET mois\_deb = %SUBSTR( &DATE\_DEB,5,2 ) ;

%LET AN\_FIN = %SUBSTR( &DATE\_FIN,1,4 ) ;

%LET mois\_fin = %SUBSTR( &DATE\_FIN,5,2 ) ;

/\*dates de soin en format SAS\*/

%LET EXE\_DEB = %SYSFUNC( mdy( &mois\_deb,1,&AN\_DEB )) ;

%LET EXE\_FIN0 = %SYSFUNC( mdy( &mois\_fin,1,&an\_fin )) ;

%LET EXE\_FIN = %EVAL( %SYSFUNC( intnx( month,&EXE\_FIN0, 1 ))-1 ) ; /\*pour aller jusqu'au dernier jour du mois de la date de fin donnée\*/

/\*dates de soin en format date9\*/

%LET DATE\_EXE\_DEB = %SYSFUNC( putn( &EXE\_DEB,date9. )) ;

%LET DATE\_EXE\_FIN = %SYSFUNC( putn( &EXE\_FIN,date9. )) ;

 /\*suppression de la table compilée des mois extraits si ce n'est pas la 1ère exécution\*/

%IF %SYSFUNC(exist(compil)) %THEN %DO;

 PROC DELETE DATA=compil;

 RUN ;

%END;

/\*BOUCLE MENSUELLE DE SELECTION DES PRESTATIONS\*/

%LET iterDeb = 1;

%LET iterFin = %SYSFUNC( intck( month,&EXE\_DEB,&EXE\_FIN )) ;/\*Nombre de mois entre le début et la fin des soins \*/

/\*BOUCLE SUR LES MOIS\*/

%DO mois=&iterDeb %TO &iterFin + 7 ;

 %LET FLX\_DIS\_DTD = %SYSFUNC( intnx( MONTH,&EXE\_DEB,&mois ), date9. ) ;

 %LET an = %SYSFUNC( year( %SYSFUNC( intnx( MONTH,&EXE\_DEB,&mois-1 )))) ;

/\*jointure par code CIP13 ou code CIP7 selon l'année\*/

 %IF &an<**2014** %THEN %LET CONDITION\_JOINTURE\_MEDICAMENTS=(pha.pha\_prs\_ide=ir.pha\_prs\_ide);

 %ELSE %LET CONDITION\_JOINTURE\_MEDICAMENTS=(pha.pha\_prs\_c13=ir.pha\_cip\_c13);

 /\*DCIR archivé\*/

 %IF &an<&an\_DCIR %THEN %DO;

 %LET table\_prs = oravue.er\_prs\_f\_&an ;

 %LET table\_pha = oravue.er\_pha\_f\_&an ;

 %END;

 /\*DCIR courant\*/

 %IF &AN >= &an\_DCIR %THEN %DO;

 %LET table\_prs = oravue.er\_prs\_f ;

 %LET table\_pha = oravue.er\_pha\_f ;

 %END;

 /\*affichage dans le journal\*/

 %PUT "Valeurs des paramètres" ;

 %PUT DATE\_EXE\_DEB = &DATE\_EXE\_DEB ;

 %PUT DATE\_EXE\_FIN = &DATE\_EXE\_FIN ;

 %PUT an = &an ;

 %PUT &table\_prs ;

 %PUT FLX\_DIS\_DTD = &FLX\_DIS\_DTD ;

%PUT CONDITION\_JOINTURE\_MEDICAMENTS=&CONDITION\_JOINTURE\_MEDICAMENTS;

 %CREA\_TABLE(&FLX\_DIS\_DTD) ;

 %COMPILATION() ;

%END ;

 %REGULARISATIONS( &table\_sortie ) ;

%MEND TAB\_MEDICAMENTS;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*lancement de la macro de sélection des prestations \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

OPTIONS MPRINT ;

 %TAB\_MEDICAMENTS( Date\_DEB = 201401,Date\_FIN = 201404,table\_sortie = table\_medicaments ) ;

* 1. Sélection de données sociodémographiques dans les référentiels des bénéficiaires

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\* SELECTION DE DONNEES SOCIODEMOGRAPHIQUES \*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\* DANS LES REFERENTIELS DES BENEFICIAIRES \*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* DONNEES en entrée

 &table\_entree à définir

 ORAVUE.IR\_BEN\_R

 ORAVUE.IR\_BEN\_R\_ARC

/\* DONNEES en sortie

 &table\_sortie à définir

/\* PLAN du PROGRAMME ;

 - Définition par l'utilisateur des paramètres suivants:

 \* table\_entree= table de population dont on veut connaître

 les caractéristiques

 \* table\_sortie= table après croisement avec les référentiels

 - Liste des variables d'intérêt dans les référentiels

 - Copie de la table entrée dans ORAUSER

 - Croisement des données avec IR\_BEN\_R

 - Croisement des données avec IR\_BEN\_R\_ARC (référentiel archivé)

 - Compilation des deux tables

 - Conservation de la dernière information mise à jour pour chaque

 BEN\_NIR\_PSA/BEN\_RNG\_GEM

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\* CROISEMENT AVEC LE REFERENTIEL DES BENEFICIAIRES \*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let table\_entree=table\_medicaments;

%let table\_sortie=table\_IR;

/\*copie de la table dans ORAUSER\*/

%KILL\_ORAUSER( &table\_entree) ;

PROC SQL;

 CREATE TABLE ORAUSER.&table\_entree AS

 SELECT \* /\*ou variables à choisir, au minimum BEN\_NIR\_PSA, BEN\_RNG\_GEM\*/

 FROM &table\_entree

 ;

QUIT ;

/\*Croisement avec les référentiels

- IR\_BEN\_R pour les individus ayant consommé au moins 1 fois depuis le 1er janvier 2013

- IR\_BEN\_R\_ARC pour les individus dont les derniers remboursements sont antérieurs à 2013\*/

%let liste\_var\_IR\_BEN =

 ben\_nai\_moi,

 ben\_sex\_cod,

 ben\_dcd\_dte, /\*sur profil 108\*/

 ben\_nir\_ano,

 ben\_idt\_ano,

 ben\_dte\_maj,

 max\_trt\_dtd

 ;

proc sql ;

/\*Croisement avec IR\_BEN\*/

create table tab\_IR\_BEN (compress=yes) as

select a.\*,&liste\_var\_IR\_BEN

from ORAUSER.&table\_entree a , ORAVUE.IR\_BEN\_R b

where a.ben\_nir\_psa = b.ben\_nir\_psa and

 a.ben\_rng\_gem = b.ben\_rng\_gem

;

/\*Croisement avec IR\_BEN\_R\_ARC\*/

create table tab\_IR\_BEN\_ARC (compress=yes) as

select a.\*, &liste\_var\_IR\_BEN

from ORAUSER.&table\_entree a , ORAVUE.IR\_BEN\_R\_ARC b

where a.ben\_nir\_psa = b.ben\_nir\_psa and

 a.ben\_rng\_gem = b.ben\_rng\_gem;

quit;

/\*Compilation des 2 tables\*/

data tab\_IR;set tab\_IR\_BEN (in=a) tab\_IR\_BEN\_ARC(in=b);

if b then source='IR\_BEN\_ARC';if a then source='IR\_BEN';run;

/\*Conservation de l'information la plus récente par ben\_nir\_psa/ben\_rng\_gem\*/

proc sort data=tab\_IR;by ben\_nir\_psa ben\_rng\_gem descending ben\_dte\_maj descending max\_trt\_dtd;run;

proc sort data=tab\_IR nodupkey out=&table\_sortie; by ben\_nir\_psa ben\_rng\_gem;run;