

# Formulaire

inesss v.1.1.0

2022-05-11

## Table des matières

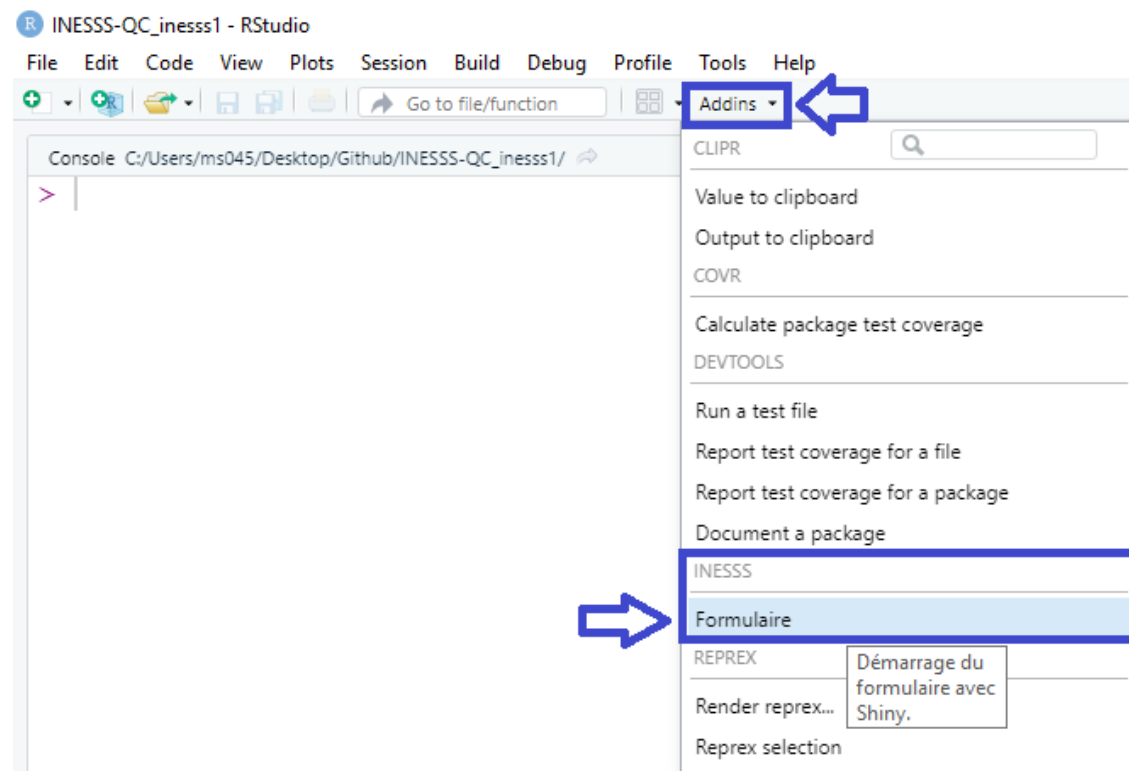
<b>1</b>	<b>Démarrer le formulaire</b>	<b>3</b>
<b>2</b>	<b>Connexion</b>	<b>4</b>
<b>3</b>	<b>Requêtes via Excel</b>	<b>5</b>
3.1	Naïf / Switch . . . . .	6
3.1.1	Arguments . . . . .	6
3.1.2	Détails . . . . .	7
3.1.3	Résultats . . . . .	7
3.1.3.1	Tableau des résultats . . . . .	7
3.1.3.2	Tableau des arguments . . . . .	8
3.1.3.3	Code SQL . . . . .	8
3.1.4	Exemples . . . . .	9
3.1.4.1	ex01 — GROUPER_PAR . . . . .	9
3.1.4.2	ex02 — DIN . . . . .	11
3.1.4.3	ex03 — Exclusion AHFS . . . . .	13
3.1.4.4	ex04 — Exclusion DENOM . . . . .	15
3.1.4.5	ex05 — Exclusion DIN . . . . .	17
3.1.4.6	ex06 — Age . . . . .	19
3.1.4.7	ex07 — Exclusion et Inclusion - CODE_SERV et CODE_LIST . . . . .	21
3.2	Statistiques générales . . . . .	23
3.2.1	Arguments . . . . .	23
3.2.2	Détails . . . . .	24
3.2.3	Résultats . . . . .	24
3.2.3.1	Tableau des résultats . . . . .	24
3.2.3.2	Tableau des arguments . . . . .	25
3.2.3.3	Code SQL . . . . .	25
3.2.4	Exemples . . . . .	26
3.2.4.1	ex02 — GROUPER_PAR . . . . .	26

---

3.2.4.2	ex03 — AHFS . . . . .	28
3.2.4.3	ex04 — DENOM . . . . .	29
3.2.4.4	ex05 — DIN . . . . .	30
3.2.4.5	ex06 — Age . . . . .	31
3.2.4.6	ex07 — Exclusion et Inclusion - CODE_SERV et CODE_LIST . . . . .	32

## 1 Démarrer le formulaire

À partir de *RStudio*, cliquer sur *Addins*, puis sur *Formulaire*.



## 2 Connexion

Établir une connexion entre *R* et *Teradata*.

1. Inscrire le nom d'utilisateur (commence généralement par *MS*) et le mot de passe.
2. Cliquer sur le bouton *Connexion*. S'il n'y a pas d'erreur, le mot *Connexion* suivi de la date et l'heure s'affichera. Modifier l'identifiant ou le mot de passe au besoin.
3. Cliquer sur le bouton *Déconnexion* (ou fermer la fenêtre du formulaire) pour terminer la connexion SQL.

**Attention :** Si le mot de passe est expiré, une fenêtre demandant un nouveau mot de passe apparaîtra **en arrière-plan**. Sélectionner cette fenêtre et entrer le nouveau mot de passe.

The image shows a light blue window for establishing a connection. It has two input fields: 'Identifiant' with the text 'MSxxx' and 'Mot de passe' with masked characters. Below these are two buttons: 'Connexion' and 'Déconnexion'. At the bottom, there is a section titled 'État de la connexion :' containing a text box that displays 'Connexion : 2021-01-14 10:08:44'.

### 3 Requetes via Excel

Exécution d'une ou plusieurs requêtes par l'intermédiaire d'un fichier Excel. Chaque onglet doit contenir un tableau indiquant les arguments d'une méthode. Il est fortement conseillé d'utiliser [les gabarits disponibles](#).

1. Cliquer sur le bouton *Sélectionner fichier Excel* pour sélectionner le fichier Excel contenant la ou les requêtes à exécuter.
2. Le répertoire du fichier sélectionné apparaît.
3. Le programme vérifie si chaque onglet contient un tableau et si sa structure correspond à une méthode existante.
4. Si la ou les structures sont acceptées, le message «**Aucune erreur, exécution possible.**» s'affiche. Sinon, le message indique, pour chaque onglet, la ou les erreurs à corriger.
5. Le bouton *Exécuter requêtes* apparaît. Ce bouton permet de sélectionner un répertoire de sauvegarde et d'inscrire le nom du fichier. Lorsque c'est fait, le programme exécute la ou les requêtes associées aux arguments de chaque onglet. Un message en bas à droite indique que l'exécution est en cours et disparaît lorsque c'est terminé.
6. Le fichier sauvegardé présente les résultats de la manière suivante :
  - Les onglets possèdent les mêmes noms que ceux du fichier contenant les arguments.
  - À gauche se trouve le tableau des résultats.
  - Au milieu, un rappel des arguments utilisés.
  - À droite, un exemple de code SQL utilisé dans Teradata pour créer le tableau des résultats (uniquement celui de la première période d'étude).

### 3.1 Naïf / Switch

Statistiques générales pour un ou des médicaments à partir d'une cohorte consommant ce(s) médicament(s) pour la première fois.

Un individu est considéré comme étant *Naïf* lorsqu'il a un traitement pour la première fois et qu'il n'a jamais eu d'autres traitements *de la même famille*. Pour obtenir les statistiques sur des patients qui prennent un médicament et qui sont naïfs à toute la classe, il faut établir l'utilisation antérieure de tous les médicaments de la classe (dont le traitement interrogé) et exclure tous les patients des cohortes analysées.

Un individu est considéré comme étant *Switch* lorsqu'il a un traitement pour la première fois, mais qu'il a eu un autre traitement dans le passé appartenant *à la même famille*.

Pour obtenir les statistiques sur des patients qui ont passé d'un traitement d'une même famille à une autre, il faut établir l'usage antérieur d'un traitement de la même classe (autre que le traitement interrogé) et se restreindre aux patients de la cohorte analysée. Cette requête est crédible seulement si l'utilisateur est certain que les médicaments de la même famille sont mutuellement exclusifs puisqu'on ne vérifie pas dans cette requête si le patient prenait le traitement interrogé dans la période antérieure.

Un individu *NaïfSwitch* est considéré comme étant un patient qui prend un traitement pour la première fois et pour lequel nous ne vérifions pas s'il s'agit d'une première consommation de tous les médicaments d'une même famille, ou encore s'il s'agit d'un changement de médicament dans cette famille. Pour obtenir les statistiques sur des patients *NaïfSwitch*, il faut établir l'usage antérieur du traitement en question et exclure ces patients de la cohorte analysée.

Vue utilisée : [V\\_DEM\\_PAINT\\_MED\\_CM](#).

#### 3.1.1 Arguments

- **METHODE** : Méthode à utiliser, ici `naif_switch1`.
- **DATE\_DEBUT** : Date(s) de début de la ou des périodes d'étude au format AAAA-MM-JJ. Doit contenir le même nombre de valeurs que `DATE_FIN`.
- **DATE\_FIN** : Date(s) de fin de la ou des périodes d'étude au format AAAA-MM-JJ. Doit contenir le même nombre de valeurs que `DATE_DEBUT`.
- **TYPE\_RX** : Type de code à analyser. Une valeur parmi :
  - `DENOM` : Code de dénomination commune.
  - `DIN` : Code d'identification du médicament.
- **CODE\_RX** : Codes à analyser selon `TYPE_RX`.
- **GROUPER\_PAR** : Afficher (aggréger) les résultats par :
  - `AHFS` : Résultats par code de classe AHFS.
  - `DENOM` : Résultats par code de dénomination commune.
  - `DIN` : Résultats par code d'identification du médicament.
  - `CodeList` : Résultats par code de catégories de liste de médicaments.
  - `CodeServ` : Résultats par code de service.
  - `Teneur` : Résultats par teneur du médicament.
  - `Format` : Résultats par format d'acquisition du médicament.
  - `Age` : Résultats par âge à une date précise. Voir argument `age_date`. L'âge est calculé à partir de la date de naissance disponible dans la vue `V_FICH_ID_BEN_CM`.

- **TYPE\_RX\_RETRO** : Type de code à exclure. Si vide, prend la valeur de **TYPE\_RX**. Une valeur parmi :
  - **AHFS** : Code identifiant la classe de médicaments telle que déterminée par l'*American Hospital Formulary Service*.
  - **DENOM** : Code de dénomination commune.
  - **DIN** : Code d'identification du médicament.
- **RX\_RETROSPECT\_A\_EXCLUDE** : Traitement(s) à inclure dans la période rétrospective. Un individu qui a au moins un traitement durant la période rétrospective ne sera pas considéré comme *naïf* ou *switch*. La période rétrospective est construite à partir des dates de références (index) et de l'argument **NJOURS\_SANS\_CONSO** :  $[INDEX - NJOURS\_SANS\_CONSO; INDEX - 1]$ .
- **NJOURS\_SANS\_CONSO** : Nombre de jours avant la date de référence (index) où un individu ne doit pas avoir consommé **RX\_RETROSPECT\_A\_EXCLUDE**. Permet de créer la période rétrospective (voir **RX\_RETROSPECT\_A\_EXCLUDE**).
- **CODE\_SERV\_FILTRE** : Exclusion ou Inclusion des codes de services. Par défaut **Exclusion**. Si aucun filtre, laisser la colonne **CODE\_SERV** vide.
- **CODE\_SERV** : Le ou les codes de service à exclure ou à inclure (sinon laisser vide). Voir [SMED\\_COD\\_SERV\\_1](#).  
Les valeurs permises sont : 1, AD et L, M, M1 à M3.
- **CODE\_LIST\_SERV** : Exclusion ou Inclusion des codes de catégorie de listes de médicaments. Par défaut **Inclusion**. Si aucun filtre, laisser la colonne **CODE\_LIST** vide.
- **CODE\_LIST** : Le ou les codes de catégories de liste de médicaments à exclure ou à inclure (sinon laisser vide). Voir [SMED\\_COD\\_CATG\\_LISTE\\_MED](#).  
Les valeurs permises sont : 03, 40 et 41.
- **AGE\_DATE** : Date à laquelle on calcule l'âge si **GROUPER\_PAR** contient **Age**. Si **NULL**, prend la première valeur de **DATE\_DEBUT**.

### 3.1.2 Détails

- **AHFS** : Un code complet AHFS est composé de 6 caractères indiquant la classe (caractères 1 et 2), la sous-classe (caractères 3 et 4) et la sous-sous-classe (caractères 5 et 6). Pour rechercher toutes les combinaisons possibles, il suffit de remplacer une paire de caractères par --. Par exemple, le code AHFS 04---- reviendrait à chercher la classe 04 et toutes les sous-classes et sous-sous-classes qui existent.

### 3.1.3 Résultats

Un onglet résultat contient trois (3) éléments :

#### 3.1.3.1 Tableau des résultats

- **DATE\_DEBUT** : Indique la ou les dates de début de la période d'étude.
- **DATE\_FIN** : Indique la ou les dates de fin de la période d'étude.
- **AHFS\_CLA** : Seulement si **GROUPER\_PAR** contient **AHFS**. Code de la classe AHFS.
- **AHFS\_SCLA** : Seulement si **GROUPER\_PAR** contient **AHFS**. Code de la sous-classe AHFS.
- **AHFS\_SSCLA** : Seulement si **GROUPER\_PAR** contient **AHFS**. Code de la sous-sous-classe AHFS.
- **NOM\_AHFS** : Seulement si **GROUPER\_PAR** contient **AHFS**. Nom de la classe AHFS.
- **DENOM** : Seulement si **GROUPER\_PAR** contient **DENOM**. Code de dénomination commune.
- **NOM\_DENOM** : Seulement si **GROUPER\_PAR** contient **DENOM**. Nom de la dénomination commune.

- **DIN** : Seulement si GROUPER\_PAR contient DIN. Code d'identification du médicament.
- **NOM\_MARQ\_COMRC** : Seulement si GROUPER\_PAR contient DIN. Nom de la marque commerciale.
- **CODE\_SERV** : Seulement si GROUPER\_PAR contient CodeServ. Code de service,
- **CODE\_LIST** : Seulement si GROUPER\_PAR contient CodeList. Code de catégorie de listes de médicaments.
- **TENEUR** : Seulement si GROUPER\_PAR contient Teneur. Teneur du médicament.
- **FORMAT\_ACQ** : Seulement si GROUPER\_PAR contient Format. Format d'acquisition du médicament.
- **AGE** : Seulement si GROUPER\_PAR contient Age. Age de l'individu à la date AGE\_DATE.
- **MNT\_MED** : Montant autorisé par la RAMQ pour le médicament ou le produit. Il comprend la part du grossiste (s'il y a lieu) et la part du manufacturier. Voir [SMED\\_MNT\\_AUTOR\\_MED](#). [sum(SMED\_MNT\_AUTOR\_MED) as MNT\_MED].
- **MNT\_SERV** : Montant de frais de service autorisé par la RAMQ à la date du service. Voir [SMED\\_MNT\\_AUTOR\\_FRAIS\\_SERV](#). [sum(SMED\_MNT\_AUTOR\_FRAIS\_SERV) as MNT\_SERV].
- **MNT\_TOT** : Somme des variables MNT\_MED et MNT\_SERV.
- **COHORTE** : Nombre d'individus unique. [count(distinct SMED\_NO\_INDIV\_BEN\_BANLS) as COHORTE].
- **NBRE\_RX** : Nombre de demandes de paiement. [count(\*) as NBRE\_RX].
- **QTE\_MED** : Quantité totale des médicaments ou des fournitures dispensés. Voir [SMED\\_QTE\\_MED](#). [sum(SMED\_QTE\_MED) as QTE\_MED].
- **DUREE\_TX** : Durée de traitement totale des prescriptions en jours. Voir [SMED\\_NBR\\_JR\\_DUREE\\_TRAIT](#). [sum(SMED\_NBR\_JR\_DUREE\_TRAIT) as DUREE\_TX].

### 3.1.3.2 Tableau des arguments

- Table rappelant les arguments fournis par l'utilisateur à la section [Arguments](#)

### 3.1.3.3 Code SQL

- Exemple de code SQL généré par les arguments. S'il y a plusieurs périodes d'étude, seul le code de la première période est affiché.



### 3.1.4 Exemples

#### 3.1.4.1 ex01 — GROUPER\_PAR

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	TYPE_RX_RETRO	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
naïf_switch1	2018-01-01	2018-12-31	DENOM	39	AHFS			365	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31		47092	DENOM								
				47135	DIN								
					CodeList								
					CodeServ								
					Teneur								
					Format								
					Age								

Code SQL : Seulement le code SQL de la première période est affiché.

```
with ALL_USER as (
  select
    SMED_NO_INDIV_BEN_BANLS as ID,
    min(SMED_DAT_SERV) as DATE_INDEX
  from V_DEM_PAIMT_MED_CM
  where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
    and SMED_COD_DENOM_COMNE in ('00039')
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
  group by ID
),
OLD_USER as (
  select distinct ID
  from ALL_USER as A left join V_DEM_PAIMT_MED_CM as V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where SMED_DAT_SERV between (DATE_INDEX - 365) and (DATE_INDEX - 1)
    and SMED_COD_DENOM_COMNE in ('00039')
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH_COHORT as (
  select ID from ALL_USER
  except
  select ID from OLD_USER
)
select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_CLA_AHF as AHFS_CLA,
```

```

SMED_COD_SCLA_AHF as AHFS_SCLA,
SMED_COD_SSCLA_AHF as AHFS_SSCLA,
SMED_COD_DENOM_COMNE as DENOM,
SMED_COD_DIN as DIN,
SMED_COD_SERV_1 as CODE_SERV,
SMED_COD_CATG_LISTE_MED as CODE_LIST,
SMED_COD_TENR_MED as TENEUR,
SMED_COD_FORMA_ACQ_MED as FORMAT_ACQ,
(cast(to_date('2018-01-01') as int) - cast(F.BENF_DAT_NAISS as int)) / 10000 as AGE,
sum(SMED_MNT_AUTOR_MED) as MNT_MED,
sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
count(*) as NBRE_RX,
sum(SMED_QTE_MED) as QTE_MED,
sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM as D left join PROD.V_FICH_ID_BEN_CM as F
  on D.SMED_NO_INDIV_BEN_BANLS = F.BENF_NO_INDIV_BEN_BANLS
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
  and SMED_NO_INDIV_BEN_BANLS in (select ID from NAIF_SWITCH_COHORT)
  and SMED_COD_DENOM_COMNE in ('00039')
  and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
  and SMED_NBR_JR_DUREE_TRAIT > 0
group by AHFS_CLA, AHFS_SCLA, AHFS_SSCLA, DENOM, DIN, CODE_SERV, CODE_LIST, TENEUR, FORMAT_ACQ, AGE
order by AHFS_CLA, AHFS_SCLA, AHFS_SSCLA, DENOM, DIN, CODE_SERV, CODE_LIST, TENEUR, FORMAT_ACQ, AGE;

```

## 3.1.4.2 ex02 — DIN

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPE_PAR	TYPE_RX_RETRO	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
naif_switch1	2018-01-01	2018-12-31	DIN	30848	DIN			365	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31		585092									

Code SQL : Seulement le code SQL de la première période est affiché.

```

with ALL_USER as (
    select
        SMED_NO_INDIV_BEN_BANLS as ID,
        min(SMED_DAT_SERV) as DATE_INDEX
    from V_DEM_PAINT_MED_CM
    where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
        and SMED_COD_DIN in (30848, 585092)
        and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
        and SMED_NBR_JR_DUREE_TRAIT > 0
    group by ID
),
OLD_USER as (
    select distinct ID
    from ALL_USER as A left join V_DEM_PAINT_MED_CM as V
        on A.ID = V.SMED_NO_INDIV_BEN_BANLS
    where SMED_DAT_SERV between (DATE_INDEX - 365) and (DATE_INDEX - 1)
        and SMED_COD_DIN in (30848, 585092)
        and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
        and SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH_COHORT as (
    select ID from ALL_USER
    except
    select ID from OLD_USER
)
select
    '2018-01-01' as DATE_DEBUT,
    '2018-12-31' as DATE_FIN,
    SMED_COD_DIN as DIN,
    sum(SMED_MNT_AUTOR_MED) as MNT_MED,
    sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
    sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
    count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,

```

```
count(*) as NBRE_RX,  
sum(SMED_QTE_MED) as QTE_MED,  
sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX  
from PROD.V_DEM_PAINT_MED_CM  
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'  
and SMED_NO_INDIV_BEN_BANLS in (select ID from NAIF_SWITCH_COHORT)  
and SMED_COD_DIN in (30848, 585092)  
and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)  
and SMED_NBR_JR_DUREE_TRAIT > 0  
group by DIN  
order by DIN;
```

## 3.1.4.3 ex03 — Exclusion AHFS

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPE_PAR	TYPE_RX_RETRO	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
naïf_switch1	2018-01-01	2018-12-31	DENOM	47092	DENOM	AHFS	04----	365	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31		47135			08--16 122436						

Code SQL : Seulement le code SQL de la première période est affiché.

```

with ALL_USER as (
    select
        SMED_NO_INDIV_BEN_BANLS as ID,
        min(SMED_DAT_SERV) as DATE_INDEX
    from V_DEM_PAINT_MED_CM
    where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
        and SMED_COD_DENOM_COMNE in ('47092', '47135')
        and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
        and SMED_NBR_JR_DUREE_TRAIT > 0
    group by ID
),
OLD_USER as (
    select distinct ID
    from ALL_USER as A left join V_DEM_PAINT_MED_CM as V
        on A.ID = V.SMED_NO_INDIV_BEN_BANLS
    where SMED_DAT_SERV between (DATE_INDEX - 365) and (DATE_INDEX - 1)
        and (SMED_COD_CLA_AHF = '04'
            or SMED_COD_CLA_AHF || SMED_COD_SSCLA_AHF = '0816'
            or SMED_COD_CLA_AHF || SMED_COD_SCLA_AHF || SMED_COD_SSCLA_AHF = '122436')
        and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
        and SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH_COHORT as (
    select ID from ALL_USER
    except
    select ID from OLD_USER
)
select
    '2018-01-01' as DATE_DEBUT,
    '2018-12-31' as DATE_FIN,
    SMED_COD_DENOM_COMNE as DENOM,
    sum(SMED_MNT_AUTOR_MED) as MNT_MED,
    sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,

```

```
sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
count(*) as NBRE_RX,
sum(SMED_QTE_MED) as QTE_MED,
sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
      and SMED_NO_INDIV_BEN_BANLS in (select ID from NAIF_SWITCH_COHORT)
      and SMED_COD_DENOM_COMNE in ('47092', '47135')
      and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
      and SMED_NBR_JR_DUREE_TRAIT > 0
group by DENOM
order by DENOM;
```

## 3.1.4.4 ex04 — Exclusion DENOM

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPE_PAR	TYPE_RX_RETRO	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
naïf_switch1	2018-01-01	2018-12-31	DENOM	47092	DENOM	DENOM	47092	365	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31		47135			47135						
							47136						

Code SQL : Seulement le code SQL de la première période est affiché.

```

with ALL_USER as (
  select
    SMED_NO_INDIV_BEN_BANLS as ID,
    min(SMED_DAT_SERV) as DATE_INDEX
  from V_DEM_PAINT_MED_CM
  where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
    and SMED_COD_DENOM_COMNE in ('47092', '47135')
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
  group by ID
),
OLD_USER as (
  select distinct ID
  from ALL_USER as A left join V_DEM_PAINT_MED_CM as V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where SMED_DAT_SERV between (DATE_INDEX - 365) and (DATE_INDEX - 1)
    and SMED_COD_DENOM_COMNE in ('47092', '47135', '47136')
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH_COHORT as (
  select ID from ALL_USER
  except
  select ID from OLD_USER
)
select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DENOM_COMNE as DENOM,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
  count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,

```

```
count(*) as NBRE_RX,  
sum(SMED_QTE_MED) as QTE_MED,  
sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX  
from PROD.V_DEM_PAINT_MED_CM  
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'  
and SMED_NO_INDIV_BEN_BANLS in (select ID from NAIF_SWITCH_COHORT)  
and SMED_COD_DENOM_COMNE in ('47092', '47135')  
and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)  
and SMED_NBR_JR_DUREE_TRAIT > 0  
group by DENOM  
order by DENOM;
```



## 3.1.4.5 ex05 — Exclusion DIN

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPE_PAR	TYPE_RX_RETRO	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
naïf_switch1	2018-01-01 2019-01-01	2018-12-31 2019-12-31	DENOM	47092	DENOM DIN	DIN	2083523 2084082 2240331 2453312	365	Exclusion	1	Inclusion		

Code SQL : Seulement le code SQL de la première période est affiché.

```

with ALL_USER as (
  select
    SMED_NO_INDIV_BEN_BANLS as ID,
    min(SMED_DAT_SERV) as DATE_INDEX
  from V_DEM_PAINT_MED_CM
  where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
    and SMED_COD_DENOM_COMNE in ('47092')
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
  group by ID
),
OLD_USER as (
  select distinct ID
  from ALL_USER as A left join V_DEM_PAINT_MED_CM as V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where SMED_DAT_SERV between (DATE_INDEX - 365) and (DATE_INDEX - 1)
    and SMED_COD_DIN in (2083523, 2084082, 2240331, 2453312)
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH_COHORT as (
  select ID from ALL_USER
  except
  select ID from OLD_USER
)
select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DENOM_COMNE as DENOM,
  SMED_COD_DIN as DIN,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,

```

```
sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
count(*) as NBRE_RX,
sum(SMED_QTE_MED) as QTE_MED,
sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
      and SMED_NO_INDIV_BEN_BANLS in (select ID from NAIF_SWITCH_COHORT)
      and SMED_COD_DENOM_COMNE in ('47092')
      and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
      and SMED_NBR_JR_DUREE_TRAIT > 0
group by DENOM, DIN
order by DENOM, DIN;
```

## 3.1.4.6 ex06 — Age

## Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPE_PAR	TYPE_RX_RETRO	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
naïf_switch1	2018-01-01	2018-12-31	DIN	30848	DIN			365	Exclusion	1	Inclusion		2018-06-05
	2019-01-01	2019-12-31		585092	Age								

**Code SQL :** Seulement le code SQL de la première période est affiché.

```
with ALL_USER as (
  select
    SMED_NO_INDIV_BEN_BANLS as ID,
    min(SMED_DAT_SERV) as DATE_INDEX
  from V_DEM_PAINT_MED_CM
  where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
    and SMED_COD_DIN in (30848, 585092)
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
  group by ID
),
OLD_USER as (
  select distinct ID
  from ALL_USER as A left join V_DEM_PAINT_MED_CM as V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where SMED_DAT_SERV between (DATE_INDEX - 365) and (DATE_INDEX - 1)
    and SMED_COD_DIN in (30848, 585092)
    and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
    and SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH_COHORT as (
  select ID from ALL_USER
  except
  select ID from OLD_USER
)
select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DIN as DIN,
  (cast(to_date('2018-06-05') as int) - cast(F.BENF_DAT_NAISS as int)) / 10000 as AGE,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
```

```
count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,  
count(*) as NBRE_RX,  
sum(SMED_QTE_MED) as QTE_MED,  
sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX  
from PROD.V_DEM_PAINT_MED_CM as D left join PROD.V_FICH_ID_BEN_CM as F  
on D.SMED_NO_INDIV_BEN_BANLS = F.BENF_NO_INDIV_BEN_BANLS  
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'  
and SMED_NO_INDIV_BEN_BANLS in (select ID from NAIF_SWITCH_COHORT)  
and SMED_COD_DIN in (30848, 585092)  
and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)  
and SMED_NBR_JR_DUREE_TRAIT > 0  
group by DIN, AGE  
order by DIN, AGE;
```

## 3.1.4.7 ex07 — Exclusion et Inclusion - CODE\_SERV et CODE\_LIST

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPE_PAR	TYPE_RX_RETRO	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
naïf_switch1	2018-01-01	2018-12-31	DENOM	39	DENOM			365	Exclusion	1	Inclusion	40	
	2019-01-01	2019-12-31		47092						AD		41	
				47135									

Code SQL : Seulement le code SQL de la première période est affiché.

```

with ALL_USER as (
    select
        SMED_NO_INDIV_BEN_BANLS as ID,
        min(SMED_DAT_SERV) as DATE_INDEX
    from V_DEM_PAINT_MED_CM
    where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
        and SMED_COD_DENOM_COMNE in ('00039', '47092', '47135')
        and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
        and SMED_COD_CATG_LISTE_MED in ('40', '41')
        and SMED_NBR_JR_DUREE_TRAIT > 0
    group by ID
),
OLD_USER as (
    select distinct ID
    from ALL_USER as A left join V_DEM_PAINT_MED_CM as V
        on A.ID = V.SMED_NO_INDIV_BEN_BANLS
    where SMED_DAT_SERV between (DATE_INDEX - 365) and (DATE_INDEX - 1)
        and SMED_COD_DENOM_COMNE in ('00039', '47092', '47135')
        and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
        and SMED_COD_CATG_LISTE_MED in ('40', '41')
        and SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH_COHORT as (
    select ID from ALL_USER
    except
    select ID from OLD_USER
)
select
    '2018-01-01' as DATE_DEBUT,
    '2018-12-31' as DATE_FIN,
    SMED_COD_DENOM_COMNE as DENOM,
    sum(SMED_MNT_AUTOR_MED) as MNT_MED,
    sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,

```

```
sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
count(*) as NBRE_RX,
sum(SMED_QTE_MED) as QTE_MED,
sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
and SMED_NO_INDIV_BEN_BANLS in (select ID from NAIF_SWITCH_COHORT)
and SMED_COD_DENOM_COMNE in ('00039', '47092', '47135')
and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
and SMED_COD_CATG_LISTE_MED in ('40', '41')
and SMED_NBR_JR_DUREE_TRAIT > 0
group by DENOM
order by DENOM;
```

## 3.2 Statistiques générales

Statistiques d'un ou de plusieurs codes de médicaments selon certains critères. Vue utilisée : [V\\_DEM\\_PAIMT\\_MED\\_CM](#).

### 3.2.1 Arguments

- **METHODE** : Méthode à utiliser, ici `stat_gen1`.
- **DATE\_DEBUT** : Date(s) de début de la ou des périodes d'étude au format AAAA-MM-JJ. Doit contenir le même nombre de valeurs que `DATE_FIN`.
- **DATE\_FIN** : Date(s) de fin de la ou des périodes d'étude au format AAAA-MM-JJ. Doit contenir le même nombre de valeurs que `DATE_DEBUT`.
- **TYPE\_RX** : Type de code à analyser. Une valeur parmi :
  - **AHFS** : Code identifiant la classe de médicaments telle que déterminée par l'*American Hospital Formulary Service*.
  - **DENOM** : Code de dénomination commune.
  - **DIN** : Code d'identification du médicament.
- **CODE\_RX** : Codes à analyser selon **TYPE\_RX**.
- **GROUPER\_PAR** : Afficher (aggréger) les résultats par :
  - **AHFS** : Résultats par code de classe AHFS.
  - **DENOM** : Résultats par code de dénomination commune.
  - **DIN** : Résultats par code d'identification du médicament.
  - **CodeList** : Résultats par code de catégories de liste de médicaments.
  - **CodeServ** : Résultats par code de service.
  - **Teneur** : Résultats par teneur du médicament.
  - **Format** : Résultats par format d'acquisition du médicament.
  - **Age** : Résultats par âge à une date précise. Voir argument `age_date`. L'âge est calculé à partir de la date de naissance disponible dans la vue `V_FICH_ID_BEN_CM`.
- **TYPE\_RX\_RETRO** : Type de code à exclure. Si vide, prend la valeur de **TYPE\_RX**. Une valeur parmi :
  - **AHFS** : Code identifiant la classe de médicaments telle que déterminée par l'*American Hospital Formulary Service*.
  - **DENOM** : Code de dénomination commune.
  - **DIN** : Code d'identification du médicament.
- **RX\_RETROSPECT\_A\_EXCLUDE** : Traitement(s) à inclure dans la période rétrospective. Un individu qui a au moins un traitement durant la période rétrospective ne sera pas considéré comme *naïf* ou *switch*. La période rétrospective est construite à partir des dates de références (index) et de l'argument `NJOURS_SANS_CONSO` : `[INDEX - NJOURS_SANS_CONSO; INDEX - 1]`.
- **NJOURS\_SANS\_CONSO** : Nombre de jours avant la date de référence (index) où un individu ne doit pas avoir consommé **RX\_RETROSPECT\_A\_EXCLUDE**. Permet de créer la période rétrospective (voir **RX\_RETROSPECT\_A\_EXCLUDE**).
- **CODE\_SERV\_FILTRE** : Exclusion ou Inclusion des codes de services. Par défaut **Exclusion**. Si aucun filtre, laisser la colonne `CODE_SERV` vide.
- **CODE\_SERV** : Le ou les codes de service à exclure ou à inclure (sinon laisser vide). Voir [SMED\\_COD\\_SERV\\_1](#).  
Les valeurs permises sont : 1, AD et L, M, M1 à M3.

- **CODE\_LIST\_SERV** : Exclusion ou Inclusion des codes de catégorie de listes de médicaments. Par défaut Inclusion. Si aucun filtre, laisser la colonne CODE\_LIST vide.
- **CODE\_LIST** : Le ou les codes de catégories de liste de médicaments à exclure ou à inclure (sinon laisser vide). Voir [SMED\\_COD\\_CATG\\_LISTE\\_MED](#). Les valeurs permises sont : 03, 40 et 41.
- **AGE\_DATE** : Date à laquelle on calcule l'âge si GROUPER\_PAR contient Age. Si NULL, prend la première valeur de DATE\_DEBUT.

### 3.2.2 Détails

- **AHFS** : Un code complet AHFS est composé de 6 caractères indiquant la classe (caractères 1 et 2), la sous-classe (caractères 3 et 4) et la sous-sous-classe (caractères 5 et 6). Pour rechercher toutes les combinaisons possibles, il suffit de remplacer une paire de caractères par --. Par exemple, le code AHFS 04---- reviendrait à chercher la classe 04 et toutes les sous-classes et sous-sous-classes qui existent.

### 3.2.3 Résultats

Un onglet résultat contient trois (3) éléments :

#### 3.2.3.1 Tableau des résultats

- **DATE\_DEBUT** : Indique la ou les dates de début de la période d'étude.
- **DATE\_FIN** : Indique la ou les dates de fin de la période d'étude.
- **AHFS\_CLA** : Seulement si GROUPER\_PAR contient AHFS. Code de la classe AHFS.
- **AHFS\_SCLA** : Seulement si GROUPER\_PAR contient AHFS. Code de la sous-classe AHFS.
- **AHFS\_SSCLA** : Seulement si GROUPER\_PAR contient AHFS. Code de la sous-sous-classe AHFS.
- **NOM\_AHFS** : Seulement si GROUPER\_PAR contient AHFS. Nom de la classe AHFS.
- **DENOM** : Seulement si GROUPER\_PAR contient DENOM. Code de dénomination commune.
- **NOM\_DENOM** : Seulement si GROUPER\_PAR contient DENOM. Nom de la dénomination commune.
- **DIN** : Seulement si GROUPER\_PAR contient DIN. Code d'identification du médicament.
- **NOM\_MARQ\_COMRC** : Seulement si GROUPER\_PAR contient DIN. Nom de la marque commerciale.
- **CODE\_SERV** : Seulement si GROUPER\_PAR contient CodeServ. Code de service,
- **CODE\_LIST** : Seulement si GROUPER\_PAR contient CodeList. Code de catégorie de listes de médicaments.
- **TENEUR** : Seulement si GROUPER\_PAR contient Teneur. Teneur du médicament.
- **FORMAT\_ACQ** : Seulement si GROUPER\_PAR contient Format. Format d'acquisition du médicament.
- **AGE** : Seulement si GROUPER\_PAR contient Age. Age de l'individu à la date AGE\_DATE.
- **MNT\_MED** : Montant autorisé par la RAMQ pour le médicament ou le produit. Il comprend la part du grossiste (s'il y a lieu) et la part du manufacturier. Voir [SMED\\_MNT\\_AUTOR\\_MED](#). [sum(SMED\_MNT\_AUTOR\_MED) as MNT\_MED].
- **MNT\_SERV** : Montant de frais de service autorisé par la RAMQ à la date du service. Voir [SMED\\_MNT\\_AUTOR\\_FRAIS\\_SERV](#). [sum(SMED\_MNT\_AUTOR\_FRAIS\_SERV) as MNT\_SERV].
- **MNT\_TOT** : Somme des variables MNT\_MED et MNT\_SERV.
- **COHORTE** : Nombre d'individus unique. [count(distinct SMED\_NO\_INDIV\_BEN\_BANLS) as COHORTE].
- **NBRE\_RX** : Nombre de demandes de paiement. [count(\*) as NBRE\_RX].
- **QTE\_MED** : Quantité totale des médicaments ou des fournitures dispensés. Voir [SMED\\_QTE\\_MED](#). [sum(SMED\_QTE\_MED) as QTE\_MED].



- **DUREE\_TX** : Durée de traitement totale des prescriptions en jours. Voir [SMED\\_NBR\\_JR\\_DUREE\\_TRAIT](#). `[sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX]`.

#### 3.2.3.2 Tableau des arguments

- Table rappelant les arguments fournis par l'utilisateur à la section [Arguments](#)

#### 3.2.3.3 Code SQL

- Exemple de code SQL généré par les arguments. S'il y a plusieurs périodes d'étude, seul le code de la première période est affiché.

### 3.2.4 Exemples

#### 3.2.4.1 ex02 — GROUPER\_PAR

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
stat_gen1	2018-01-01	2018-12-31	DENOM	39	AHFS	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31		47092	DENOM					
				47135	DIN					
					CodeList					
					CodeServ					
					Teneur					
					Format					
					Age					

**Code SQL :** Seulement le code SQL de la première période est affiché.

```
select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_CLA_AHF as AHFS_CLA,
  SMED_COD_SCLA_AHF as AHFS_SCLA,
  SMED_COD_SSCLA_AHF as AHFS_SSCLA,
  SMED_COD_DENOM_COMNE as DENOM,
  SMED_COD_DIN as DIN,
  SMED_COD_SERV_1 as CODE_SERV,
  SMED_COD_CATG_LISTE_MED as CODE_LIST,
  SMED_COD_TENR_MED as TENEUR,
  SMED_COD_FORMA_ACQ_MED as FORMAT_ACQ,
  (cast(to_date('2018-01-01') as int) - cast(F.BENF_DAT_NAISS as int)) / 10000 as AGE,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
  count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
  count(*) as NBRE_RX,
  sum(SMED_QTE_MED) as QTE_MED,
  sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM as D left join PROD.V_FICH_ID_BEN_CM as F
  on D.SMED_NO_INDIV_BEN_BANLS = F.BENF_NO_INDIV_BEN_BANLS
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
  and SMED_COD_DENOM_COMNE in ('00039', '47092', '47135')
  and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
  and SMED_NBR_JR_DUREE_TRAIT > 0
```

```
group by AHFS_CLA, AHFS_SCLA, AHFS_SSCLA, DENOM, DIN, CODE_SERV, CODE_LIST, TENEUR, FORMAT_ACQ, AGE  
order by AHFS_CLA, AHFS_SCLA, AHFS_SSCLA, DENOM, DIN, CODE_SERV, CODE_LIST, TENEUR, FORMAT_ACQ, AGE;
```

## 3.2.4.2 ex03 — AHFS

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
stat_gen1	2018-01-01	2018-12-31	AHFS	04----	AHFS	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31		08--12						
				122426						

Code SQL : Seulement le code SQL de la première période est affiché.

```

select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DENOM_COMNE as DENOM,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
  count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
  count(*) as NBRE_RX,
  sum(SMED_QTE_MED) as QTE_MED,
  sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
      and SMED_COD_DENOM_COMNE in ('04----', '08--12', '122426')
      and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
      and SMED_NBR_JR_DUREE_TRAIT > 0
group by DENOM
order by DENOM;

```

## 3.2.4.3 ex04 — DENOM

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
stat_gen1	2018-01-01	2018-12-31	DENOM	39, 47092, 47135	DENOM	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31			DIN					

Code SQL : Seulement le code SQL de la première période est affiché.

```

select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DENOM_COMNE as DENOM,
  SMED_COD_DIN as DIN,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
  count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
  count(*) as NBRE_RX,
  sum(SMED_QTE_MED) as QTE_MED,
  sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
  and SMED_COD_DENOM_COMNE in ('39, 47092, 47135')
  and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
  and SMED_NBR_JR_DUREE_TRAIT > 0
group by DENOM, DIN
order by DENOM, DIN;

```

## 3.2.4.4 ex05 — DIN

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
stat_gen1	2018-01-01	2018-12-31	DIN	30848	DIN	Exclusion	1	Inclusion		
	2019-01-01	2019-12-31		585092						

Code SQL : Seulement le code SQL de la première période est affiché.

```

select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DIN as DIN,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
  count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
  count(*) as NBRE_RX,
  sum(SMED_QTE_MED) as QTE_MED,
  sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
  and SMED_COD_DIN in (30848, 585092)
  and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
  and SMED_NBR_JR_DUREE_TRAIT > 0
group by DIN
order by DIN;

```

## 3.2.4.5 ex06 — Age

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
stat_gen1	2018-01-01	2018-12-31	DIN	30848	DIN	Exclusion	1	Inclusion		2018-01-01
	2019-01-01	2019-12-31		585092	Age					

Code SQL : Seulement le code SQL de la première période est affiché.

```

select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DIN as DIN,
  (cast(to_date('2018-01-01') as int) - cast(F.BENF_DAT_NAISS as int)) / 10000 as AGE,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
  count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
  count(*) as NBRE_RX,
  sum(SMED_QTE_MED) as QTE_MED,
  sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAINT_MED_CM as D left join PROD.V_FICH_ID_BEN_CM as F
  on D.SMED_NO_INDIV_BEN_BANLS = F.BENF_NO_INDIV_BEN_BANLS
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
  and SMED_COD_DIN in (30848, 585092)
  and (SMED_COD_SERV_1 not in ('1') or SMED_COD_SERV_1 is null)
  and SMED_NBR_JR_DUREE_TRAIT > 0
group by DIN, AGE
order by DIN, AGE;

```

## 3.2.4.6 ex07 — Exclusion et Inclusion - CODE\_SERV et CODE\_LIST

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST	AGE_DATE
stat_gen1	2018-01-01	2018-12-31	DENOM	39, 47092, 47135	DENOM	Exclusion	1	Inclusion	40	
	2019-01-01	2019-12-31					AD		41	

Code SQL : Seulement le code SQL de la première période est affiché.

```

select
  '2018-01-01' as DATE_DEBUT,
  '2018-12-31' as DATE_FIN,
  SMED_COD_DENOM_COMNE as DENOM,
  sum(SMED_MNT_AUTOR_MED) as MNT_MED,
  sum(SMED_MNT_AUTOR_FRAIS_SERV) as MNT_SERV,
  sum(SMED_MNT_AUTOR_FRAIS_SERV + SMED_MNT_AUTOR_MED) as MNT_TOT,
  count(distinct SMED_NO_INDIV_BEN_BANLS) as COHORTE,
  count(*) as NBRE_RX,
  sum(SMED_QTE_MED) as QTE_MED,
  sum(SMED_NBR_JR_DUREE_TRAIT) as DUREE_TX
from PROD.V_DEM_PAIMT_MED_CM
where SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
  and SMED_COD_DENOM_COMNE in ('39, 47092, 47135')
  and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
  and SMED_COD_CATG_LISTE_MED in ('40', '41')
  and SMED_NBR_JR_DUREE_TRAIT > 0
group by DENOM
order by DENOM;

```