

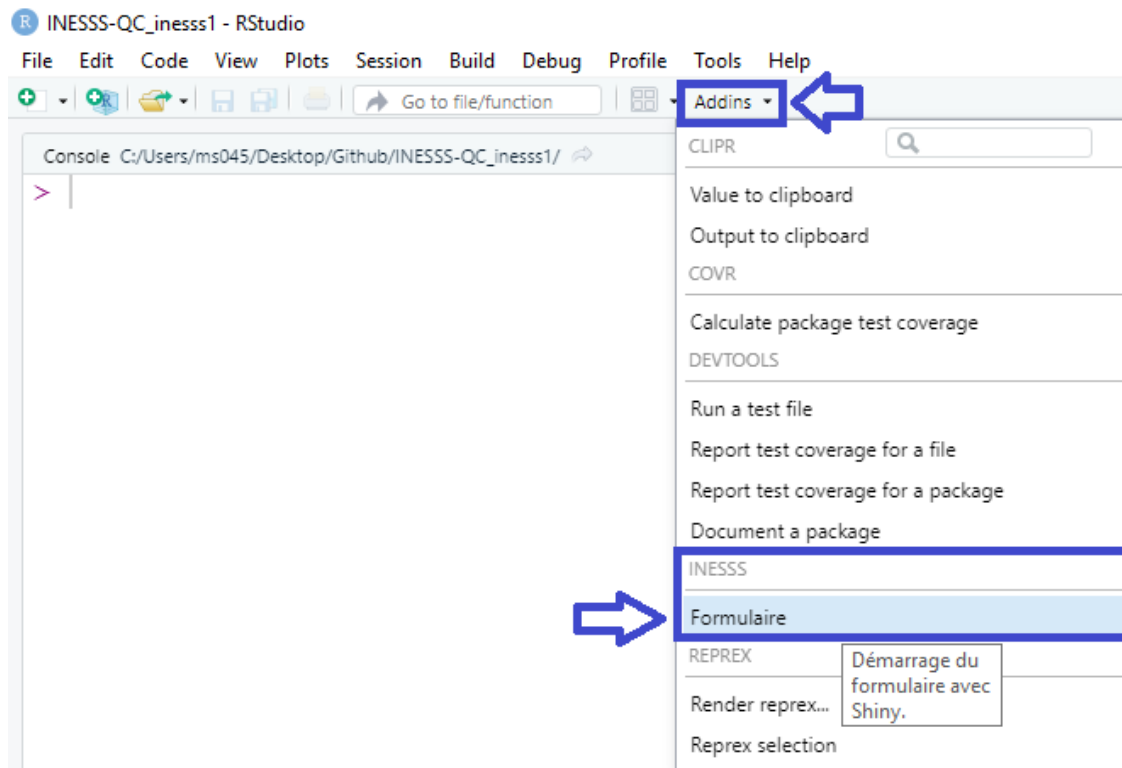
Formulaire

inesss v.0.7.3

2021-04-12

1 Démarrer le formulaire

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

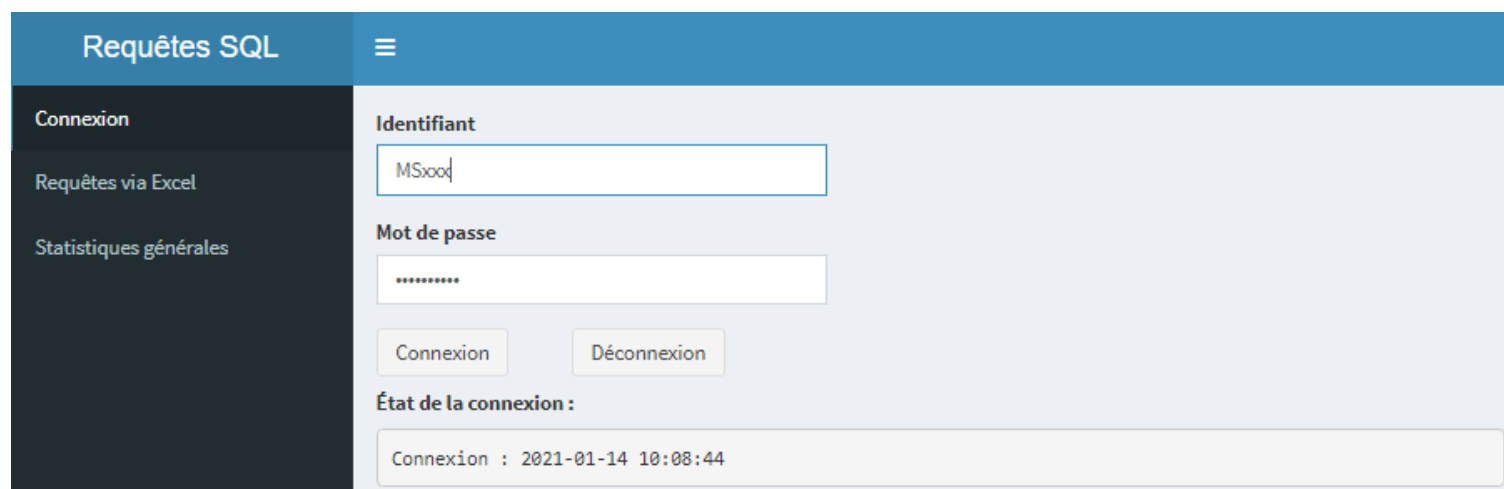


2 Connexion

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 screenshot shows a web application interface for SQL queries. On the left is a dark sidebar with the title 'Requêtes SQL' and three menu items: 'Connexion', 'Requêtes via Excel', and 'Statistiques générales'. The main area has a light blue header with a hamburger menu icon. Below the header, the 'Connexion' section contains a form with two input fields: 'Identifiant' (containing 'MSxxx') and 'Mot de passe' (masked with dots). Below these fields are two buttons: 'Connexion' and 'Déconnexion'. At the bottom, there is a section titled 'État de la connexion :' with a light gray box displaying the text '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 sous le bouton pour vérifier si c'est le bon fichier à utiliser.
3. Le programme vérifie si chaque onglet contient un tableau et si sa structure correspond à une méthode existante.
4. Si la structure est acceptée, le message «**Aucune erreur, exécution possible.**» s'affiche. Sinon, le message indique, pour chaque onglet, les erreurs à corriger.
5. Si la structure est acceptée, 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 noms des onglets sont les mêmes 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.

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.

Vue utilisée : [V_DEM_PAINT_MED_CM](#).

3.1.1 Arguments

- **METHODE** : Indique la méthode à utiliser, ici `naïf_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** : Distinction entre le code de dénomination commune (DENOM) ou le code d'identification du médicament (DIN).
- **CODE_RX** : Codes DENOM ou DIN à analyser. **Inscrire les codes sous la forme d'un nombre entier.**
- **GROUPER_PAR** : Afficher (aggréger) les résultats par :
 - Codes : Résultats pour chaque `CODE_RX`.
- **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".

3.1.2 Résultats

Un onglet résultat contient trois (3) éléments :

3.1.2.1 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.
- **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]`.
- **DENOM / DIN** : Code(s) analysé(s) à l'intérieur de la période d'étude.
- **RX_RETROSPECT_A_EXCLURE** : Code(s) de médicament(s) qui n'ont *jamais* été consommé(s) durant la période rétrospective.
- **NJOURS_SANS_CONSO** : Nombre de jours qu'un individu ne doit pas avoir reçu de traitements avant sa date de référence (index) pour être considéré comme *naïf* ou *switch*.

3.1.2.2 Tableau des arguments

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

3.1.2.3 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.

3.1.3 Exemples

3.1.3.1 DENOM — GROUPER_PAR Codes

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
naif_switch1	2018-01-01	2018-12-31	DENOM	47873	Codes		365	Exclusion	1	Inclusion	
	2019-01-01	2019-12-31		47958							
				48213							

Tableau des résultats :

DATE_DEBUT	DATE_FIN	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX	DENOM	RX_RETROSPECT_A_EXCLURE
2018-01-01	2018-12-31	5 498 472,55	15 099,27	5 513 571,82	312	1 919	140 867,000	48 235	47873	47873; 47958; 48213
2018-01-01	2018-12-31	6 073 116,77	16 672,18	6 089 788,95	353	2 160	211 831,000	53 356	47958	47873; 47958; 48213
2018-01-01	2018-12-31	374 855,88	893,87	375 749,75	31	98	12 480,000	3 120	48213	47873; 47958; 48213
2019-01-01	2019-12-31	6 280 951,54	18 280,86	6 299 232,40	348	2 301	148 889,000	55 379	47873	47873; 47958; 48213
2019-01-01	2019-12-31	7 172 793,62	21 048,11	7 193 841,73	398	2 595	250 186,000	64 110	47958	47873; 47958; 48213
2019-01-01	2019-12-31	1 334 109,69	3 439,77	1 337 549,46	81	391	45 656,000	11 474	48213	47873; 47958; 48213

Code SQL : Seulement le code SQL de la première période et du premier code d'analyse est affiché.

```
with ALL_USER as (
  select SMED_NO_INDIV_BEN_BANLS as ID,
         min(SMED_DAT_SERV) as DATE_INDEX
  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 ('47873')
        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(A.ID)
  from ALL_USER As A left join PROD.V_DEM_PAIMT_MED_CM As V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where V.SMED_DAT_SERV between (A.DATE_INDEX - 365) and (A.DATE_INDEX - 1)
        and V.SMED_COD_DENOM_COMNE in ('47873', '47958', '48213')
        and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)
        and V.SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH 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,
       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 NAIF_SWITCH As N left join PROD.V_DEM_PAIMT_MED_CM as V
  on N.ID = V.SMED_NO_INDIV_BEN_BANLS
where V.SMED_DAT_SERV between '2018-01-01' and '2018-12-31'
      and V.SMED_COD_DENOM_COMNE in ('47873')
      and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)
      and V.SMED_NBR_JR_DUREE_TRAIT > 0;
```

3.1.3.2 GROUPER_PAR NULL — Plusieurs codes pour un même type de traitement.

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
naïf_switch1	2018-01-01	2018-12-31	DENOM	47873			365	Exclusion	1	Inclusion	
	2019-01-01	2019-12-31		47958							
				48213							

Tableau des résultats :

DATE_DEBUT	DATE_FIN	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX	DENOM	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO
2018-01-01	2018-12-31	12 186 634,21	33 312,52	12 219 946,73	696	4 253	372 598,000	106 800	47873; 47958; 48213	47873; 47958; 48213	365
2019-01-01	2019-12-31	15 112 678,21	43 625,64	15 156 303,85	827	5 388	454 837,000	133 826	47873; 47958; 48213	47873; 47958; 48213	365

Code SQL : Seule la première période est affichée.

```
with ALL_USER as (
  select SMED_NO_INDIV_BEN_BANLS as ID,
         min(SMED_DAT_SERV) as DATE_INDEX
  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 ('47873', '47958', '48213')
         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(A.ID)
  from ALL_USER As A left join PROD.V_DEM_PAIMT_MED_CM As V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where V.SMED_DAT_SERV between (A.DATE_INDEX - 365) and (A.DATE_INDEX - 1)
         and V.SMED_COD_DENOM_COMNE in ('47873', '47958', '48213')
         and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)
         and V.SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH 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,
       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 NAIF_SWITCH As N left join PROD.V_DEM_PAINT_MED_CM as V  
on N.ID = V.SMED_NO_INDIV_BEN_BANLS  
where V.SMED_DAT_SERV between '2018-01-01' and '2018-12-31'  
and V.SMED_COD_DENOM_COMNE in ('47873', '47958', '48213')  
and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)  
and V.SMED_NBR_JR_DUREE_TRAIT > 0;
```

3.1.3.3 DIN

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
naif_switch1	2018-01-01 2019-01-01	2018-12-31 2019-12-31	DIN	2241927			365	Exclusion	1	Inclusion	

Tableau des résultats :

DATE_DEBUT	DATE_FIN	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX	DIN	RX_RETROSPECT_A_EXCLURE	NJOURS_SANS_CONSO
2018-01-01	2018-12-31	3 038 118,83	6 409,15	3 044 527,98	286	660	62 900,000	11 903	2241927	2241927	365
2019-01-01	2019-12-31	2 986 888,06	6 500,12	2 993 388,18	273	648	61 890,000	10 740	2241927	2241927	365

Code SQL : Seule la première période est affichée.

```

with ALL_USER as (
  select SMED_NO_INDIV_BEN_BANLS as ID,
         min(SMED_DAT_SERV) as DATE_INDEX
  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 ('2241927')
        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(A.ID)
  from ALL_USER As A left join PROD.V_DEM_PAIMT_MED_CM As V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where V.SMED_DAT_SERV between (A.DATE_INDEX - 365) and (A.DATE_INDEX - 1)
        and V.SMED_COD_DENOM_COMNE in ('2241927')
        and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)
        and V.SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH 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,
       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 NAIF_SWITCH As N left join PROD.V_DEM_PAINT_MED_CM as V  
on N.ID = V.SMED_NO_INDIV_BEN_BANLS  
where V.SMED_DAT_SERV between '2018-01-01' and '2018-12-31'  
and V.SMED_COD_DENOM_COMNE in ('2241927')  
and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)  
and V.SMED_NBR_JR_DUREE_TRAIT > 0;
```

3.1.3.4 RX_RETROSPECT_A_EXCLUDE conteant plus de valeurs que CODES (avec ou sans les memes valeurs)

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	RX_RETROSPECT_A_EXCLUDE	NJOURS_SANS_CONSO	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
naïf_switch1	2018-01-01	2018-12-31	DIN	2257238		2042479	365	Exclusion	1	Inclusion	
	2019-01-01	2019-12-31		2272903		2042487					
						2257238					
						2272903					
						2317192					
						2317206					

Tableau des résultats :

DATE_DEBUT	DATE_FIN	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX	DIN	RX_RETROSPECT_A_EXCLUDE	NJOURS_SANS_CONSO
2018-01-01	2018-12-31	84 957,05	50 647,43	135 604,48	1 193	5 329	6 432,000	165 302	2257238; 2272903	2042479; 2042487; 2257238; 2272903; 2317192; 2317206	365
2019-01-01	2019-12-31	87 930,89	53 169,44	141 100,33	1 301	5 510	6 657,000	172 053	2257238; 2272903	2042479; 2042487; 2257238; 2272903; 2317192; 2317206	365

Code SQL : Seule la première période est affichée.

```
with ALL_USER as (
  select SMED_NO_INDIV_BEN_BANLS as ID,
         min(SMED_DAT_SERV) as DATE_INDEX
  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 ('2257238', '2272903')
        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(A.ID)
  from ALL_USER As A left join PROD.V_DEM_PAIMT_MED_CM As V
    on A.ID = V.SMED_NO_INDIV_BEN_BANLS
  where V.SMED_DAT_SERV between (A.DATE_INDEX - 365) and (A.DATE_INDEX - 1)
        and V.SMED_COD_DENOM_COMNE in ('2042479', '2042487', '2257238', '2272903', '2317192', '2317206')
        and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)
        and V.SMED_NBR_JR_DUREE_TRAIT > 0
),
NAIF_SWITCH 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,  
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 NAIF_SWITCH As N left join PROD.V_DEM_PAIMT_MED_CM as V  
on N.ID = V.SMED_NO_INDIV_BEN_BANLS  
where V.SMED_DAT_SERV between '2018-01-01' and '2018-12-31'  
and V.SMED_COD_DENOM_COMNE in ('2257238', '2272903')  
and (V.SMED_COD_SERV_1 not in ('1') or V.SMED_COD_SERV_1 is null)  
and V.SMED_NBR_JR_DUREE_TRAIT > 0;
```

3.2 Statistiques générales

Statistiques d'un ou de plusieurs codes de médicaments selon certains critères.

Vue utilisée : [V_DEM_PAINT_MED_CM](#).

3.2.1 Arguments

- **METHODE** : Indique la 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** : Distinction entre le code de dénomination commune (DENOM) ou le code d'identification du médicament (DIN).
- **CODE_RX** : Codes DENOM ou DIN à analyser. **Inscrire les codes sous la forme d'un nombre entier.**
- **GROUPE_PAR** : Afficher (aggréger) les résultats par :
 - Codes : Résultats pour chaque **CODE_RX**.
 - Teneur : Résultats par teneur du médicament. Voir [SMED_COD_TENR_MED](#).
 - Format : Résultats par format d'acquisition du médicament. Voir [SMED_COD_FORMA_ACQ_MED](#).
- **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".

3.2.2 Résultats

Un onglet résultat contient trois (3) éléments :

3.2.2.1 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.
- **DENOM / DIN** : Code(s) analysé(s) à l'intérieur de la période d'étude. Si on ne demande pas les résultats par **Codes**, cette colonne se situera à la fin du tableau et tous les codes seront indiqués dans une même cellule.
- **NOM_DENOM / NOM_MARQ_COMRC** : Indique le nom de la dénomination commune si c'est un DENOM ou le nom de la marque de commerce si c'est un DIN.
Conditionnel : demander les résultats par **Codes**.
- **TENEUR** : Teneur du médicament. Conditionnel : demander les résultats par **Teneur**.
- **FORMAT_ACQ** : Format d'acquisition du médicament. Conditionnel : demander les résultats par **Format**.
- **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]`.

3.2.2.2 Tableau des arguments

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

3.2.2.3 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.

3.2.3 Exemples

3.2.3.1 GROUPER_PAR : Aucune variable.

Arguments :

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

Tableau des résultats :

DATE_DEBUT	DATE_FIN	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX	DENOM
2018-01-01	2018-12-31	6 601 042,70	24 388 771,29	30 989 813,99	434 363	3 865 192	286 658 817,730	51 528 479	39; 47092; 47135
2019-01-01	2019-12-31	7 945 756,10	24 998 247,75	32 944 003,85	445 568	3 962 187	291 918 813,840	51 838 096	39; 47092; 47135

Code SQL : Seule la première période est affichée.

```
select '2018-01-01' as DATE_DEBUT,
       '2018-12-31' as DATE_FIN,
       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 ('00039', '47092', '47135')
       and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null);
```


3.2.3.2 GROUPER_PAR : Codes – DENOM.**Arguments :**

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

Tableau des résultats :

DATE_DEBUT	DATE_FIN	DENOM	NOM_DENOM	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX
2018-01-01	2018-12-31	39	acétaminophène	4 298 016,11	21 154 864,49	25 452 880,60	409 770	3 400 521	277 405 408,730	42 559 050
2018-01-01	2018-12-31	47092	bézafrabrate	576 893,95	126 665,06	703 559,01	1 247	17 827	360 738,000	362 826
2018-01-01	2018-12-31	47135	losartan potassique	1 726 132,64	3 107 241,74	4 833 374,38	30 477	446 844	8 892 671,000	8 606 603
2019-01-01	2019-12-31	39	acétaminophène	4 367 678,41	21 824 084,78	26 191 763,19	419 985	3 502 576	282 496 667,790	43 238 159
2019-01-01	2019-12-31	47092	bézafrabrate	538 942,56	124 316,34	663 258,90	1 184	17 431	345 836,000	347 911
2019-01-01	2019-12-31	47135	losartan potassique	3 039 135,13	3 049 846,63	6 088 981,76	31 585	442 180	9 076 310,050	8 252 026

Code SQL : Seule la première période est affichée.

```

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 ('00039', '47092', '47135')
      and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
group by DENOM
order by DENOM;

```

3.2.3.3 GROUPER_PAR : Codes – DIN.**Arguments :**

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

Tableau des résultats :

DATE_DEBUT	DATE_FIN	DIN	NOM_MARQ_COMRC	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX
2018-01-01	2018-12-31	30848	Depo-Provera	2 966,34	627,93	3 594,27	23	72	565,000	3 791
2018-01-01	2018-12-31	585092	Depo-Provera	1 108 837,90	326 632,80	1 435 470,70	12 711	37 327	38 070,000	2 914 122
2019-01-01	2019-12-31	30848	Depo-Provera	3 570,17	609,40	4 179,57	16	68	680,000	3 068
2019-01-01	2019-12-31	585092	Depo-Provera	1 090 991,97	339 924,77	1 430 916,74	12 691	37 268	37 858,000	2 938 446

Code SQL : Seule la première période est affichée.

```

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', 'AD') or SMED_COD_SERV_1 is null)
group by DIN
order by DIN;

```

3.2.3.4 GROUPER_PAR : Codes et Format.

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
stat_gen1	2018-01-01	2018-12-31	DENOM	39	Codes	Exclusion	1	Inclusion	
	2019-01-01	2019-12-31		47092	Format		AD		
				48222					

Tableau des résultats :

DATE_DEBUT	DATE_FIN	DENOM	NOM_DENOM	FORMAT_ACQ	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX
2018-01-01	2018-12-31	39	acétaminophène	NA	338,75	145,33	484,08	2	17	2 890,000	435
2018-01-01	2018-12-31	48222	cabozantinib	NA	206 184,00	203,69	206 387,69	15	22	660,000	660
2018-01-01	2018-12-31	39	acétaminophène	1	310 355,07	569 574,10	879 929,17	20 700	78 891	9 282 756,920	966 069
2018-01-01	2018-12-31	47092	bézafibrate	1	576 893,95	126 665,06	703 559,01	1 247	17 827	360 738,000	362 826
2018-01-01	2018-12-31	39	acétaminophène	2	3 987 322,29	20 585 145,06	24 572 467,35	394 483	3 321 613	268 119 761,810	41 592 546
2019-01-01	2019-12-31	39	acétaminophène	NA	222,41	100,20	322,61	1	12	2 040,000	327
2019-01-01	2019-12-31	48222	cabozantinib	NA	2 272 397,60	2 121,05	2 274 518,65	59	224	7 274,000	7 109
2019-01-01	2019-12-31	39	acétaminophène	1	316 500,94	597 863,42	914 364,36	21 370	82 247	9 633 602,800	1 009 935
2019-01-01	2019-12-31	47092	bézafibrate	1	538 942,56	124 316,34	663 258,90	1 184	17 431	345 836,000	347 911
2019-01-01	2019-12-31	48222	cabozantinib	1	97 227,90	61,82	97 289,72	7	7	330,000	210
2019-01-01	2019-12-31	39	acétaminophène	2	4 050 781,32	21 226 068,38	25 276 849,70	404 336	3 420 311	272 859 660,990	42 227 812
2019-01-01	2019-12-31	39	acétaminophène	3	173,74	52,78	226,52	4	6	1 364,000	85

Code SQL : Seule la première période est affichée.

```
select '2018-01-01' as DATE_DEBUT,
       '2018-12-31' as DATE_FIN,
       SMED_COD_DENOM_COMNE as DENOM,
       SMED_COD_FORMA_ACQ_MED as FORMAT_ACQ,
       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 ('00039', '47092', '48222')
       and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
group by DENOM, FORMAT_ACQ
order by DENOM, FORMAT_ACQ;
```

3.2.3.5 GROUPER_PAR : Teneur.**Arguments :**

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
stat_gen1	2018-01-01	2018-12-31	DENOM	47092	Teneur	Exclusion	1	Inclusion	
	2019-01-01	2019-12-31		48222			AD		

Tableau des résultats :

DATE_DEBUT	DATE_FIN	TENEUR	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX	DENOM
2018-01-01	2018-12-31	28426	46 860,00	45,89	46 905,89	3	5	150,000	150	48222
2018-01-01	2018-12-31	34160	28 116,00	28,08	28 144,08	3	3	90,000	90	48222
2018-01-01	2018-12-31	38308	131 208,00	129,72	131 337,72	11	14	420,000	420	48222
2018-01-01	2018-12-31	53192	576 893,95	126 665,06	703 559,01	1 247	17 827	360 738,000	362 826	47092
2019-01-01	2019-12-31	28426	633 564,30	465,90	634 030,20	17	46	2 040,000	1 530	48222
2019-01-01	2019-12-31	34160	846 787,40	845,94	847 633,34	32	90	2 714,000	2 875	48222
2019-01-01	2019-12-31	38308	889 273,80	871,03	890 144,83	40	95	2 850,000	2 914	48222
2019-01-01	2019-12-31	53192	538 942,56	124 316,34	663 258,90	1 184	17 431	345 836,000	347 911	47092

Code SQL : Seule la première période est affichée.

```

select '2018-01-01' as DATE_DEBUT,
       '2018-12-31' as DATE_FIN,
       SMED_COD_TENR_MED as TENEUR,
       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 ('00039', '47092', '48222')
      and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
group by TENEUR
order by TENEUR;

```

3.2.3.6 Trimestres

Arguments :

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPER_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
stat_gen1	2018-01-01	2018-03-31	DENOM	47092	Codes	Exclusion	1	Inclusion	
	2018-04-01	2018-06-30		48222			AD		
	2018-07-01	2018-09-30							
	2018-10-01	2018-12-31							

Tableau des résultats :

DATE_DEBUT	DATE_FIN	DENOM	NOM_DENOM	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX
2018-01-01	2018-03-31	47092	bézafrabrate	140 391,38	31 559,79	171 951,17	1 070	4 323	87 458,000	87 905
2018-04-01	2018-06-30	47092	bézafrabrate	145 115,47	31 977,52	177 092,99	1 089	4 459	90 397,000	91 092
2018-07-01	2018-09-30	47092	bézafrabrate	143 815,05	31 236,10	175 051,15	1 081	4 489	89 587,000	89 993
2018-10-01	2018-12-31	47092	bézafrabrate	147 572,05	31 891,65	179 463,70	1 067	4 556	93 296,000	93 836
2018-10-01	2018-12-31	48222	cabozantinib	206 184,00	203,69	206 387,69	15	22	660,000	660

Code SQL : Seule la première période est affichée.

```

select '2018-01-01' as DATE_DEBUT,
       '2018-03-31' 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 '2018-03-31'
       and SMED_COD_DENOM_COMNE in ('47092', '48222')
       and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
group by DENOM
order by DENOM;

```

3.2.3.7 Exclusion codes de services et Inclusion codes listes.**Arguments :**

METHODE	DATE_DEBUT	DATE_FIN	TYPE_RX	CODE_RX	GROUPE_PAR	CODE_SERV_FILTRE	CODE_SERV	CODE_LIST_FILTRE	CODE_LIST
stat_gen1	2018-01-01	2018-12-31	DENOM	39		Exclusion	1	Inclusion	03
	2019-01-01	2019-12-31		47092			AD		40
				48222					

Tableau des résultats :

DATE_DEBUT	DATE_FIN	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX	DENOM
2018-01-01	2018-12-31	5 081 094,06	21 281 733,24	26 362 827,30	410 763	3 418 370	277 766 806,730	42 922 536	39; 47092; 48222
2019-01-01	2019-12-31	7 276 246,47	21 950 583,99	29 226 830,46	420 976	3 520 238	282 850 107,790	43 593 389	39; 47092; 48222

Code SQL : Seule la première période est affichée.

```

select '2018-01-01' as DATE_DEBUT,
       '2018-12-31' as DATE_FIN,
       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 ('00039', '47092', '48222')
       and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
       and SMED_COD_CATG_LISTE_MED in ('03', '40');

```

4 Statistiques générales

Formulaire interactif permettant l'inscription d'arguments, l'exécution de la requête et la sauvegarde des résultats à partir de la vue [V_DEM_PAINT_MED_CM](#).

Requêtes SQL

Connexion

Requêtes via Excel

Statistiques générales

Arguments

Nombre de périodes

2

Période 1

2018-01-01 au 2018-12-31

Période 2

2019-01-01 au 2019-12-31

Nombre de Codes Rx

3

Type de Code Rx

DENOM

Code Rx 1

47092

Code Rx 2

47135

Code Rx 3

48222

Grouper par

☒ Codes
☐ Teneur
☐ Format

Codes de Services

Exclusion

☒ 1: REFUS D'ORDONNANCE
☒ AD : AJUSTEMENT DE LA DOSE D'UN MEDICAMENT
☐ L, M, M1 à M3 : PREPARATION MAGISTRALE

Codes Liste Médicament

Inclusion

☐ 03 : Liste des médicaments en pharmacie
☐ 40 : Liste des médicaments de patients d'exception en pharmacie
☐ 41 : Liste des médicaments d'exception en pharmacie

Exécuter Requête

Réinitialiser Arguments

Résultats

Show 25 entries

Search:

DATE_DEBUT	DATE_FIN	DENOM	NOM_DENOM	MNT_MED	MNT_SERV	MNT_TOT	COHORTE	NBRE_RX	QTE_MED	DUREE_TX
2018-01-01	2018-12-31	47092	bézafrilate	576 893,95 \$	126 665,06 \$	703 559,01 \$	1 247	17 827	360 738,000	362 826
2018-01-01	2018-12-31	47135	losartan potassique	1 726 132,64 \$	3 107 241,74 \$	4 833 374,38 \$	30 477	446 844	8 892 671,000	8 606 603
2018-01-01	2018-12-31	48222	cabozantinib	206 184,00 \$	203,69 \$	206 387,69 \$	15	22	660,000	660
2019-01-01	2019-12-31	47092	bézafrilate	538 942,56 \$	124 316,34 \$	663 258,90 \$	1 184	17 431	345 836,000	347 911
2019-01-01	2019-12-31	47135	losartan potassique	3 039 135,13 \$	3 104 237,72 \$	6 143 372,85 \$	31 626	445 477	9 076 310,050	8 252 026
2019-01-01	2019-12-31	48222	cabozantinib	2 369 625,50 \$	2 182,87 \$	2 371 808,37 \$	59	231	7 604,000	7 319

DATE_DEBUT

DATE_FIN

DENOM

NOM_DENOM

MNT_MED

MNT_SERV

MNT_TOT

COHORTE

NBRE_RX

QTE_MED

DUREE_TX

Showing 1 to 6 of 6 entries

Previous 1 Next

Sauvegarder Résultats en Excel

Requête SQL

```

select '2018-01-01' as DATE_DEBUT,
       '2018-12-31' as DATE_FIN,
       SMED_COD_DENOM_COMME 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_BANIS) as COHORTE,
       count(*) as NBRE_RX,
       sum(SMED_QTE_MED) as QTE_MED,
       sum(SMED_NBR_3R_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_COMME in ('47092', '47135', '48222')
   and (SMED_COD_SERV_1 not in ('1', 'AD') or SMED_COD_SERV_1 is null)
group by DENOM
order by DENOM;

```

23

4.1 Arguments

- **Nombre de périodes** : Nombre entier. Indique le nombre de périodes d'étude à analyser.
- **Période X** : Sélectionner ou inscrire manuellement la date de début et de fin de la période d'étude *X*.
- **Nombre de Codes Rx** : Nombre entier. Indique le nombre de Code Rx à analyser.
- **Type de Code Rx** : DENOM ou DIN. Indique le type de Code Rx à analyser.
 - DENOM : Code de dénomination commune. Voir [SMED_COD_DENOM_COMNE](#).
 - DIN : Code d'identification du médicament. Voir [SMED_COD_DIN](#).
- **Code Rx X** : Le(s) code(s) de médicament à analyser. Inscrire les codes sous la forme d'un nombre entier.
- **Grouper par** :
 - Codes : Résultats pour chaque Code Rx.
 - Teneur : Résultats par teneur du médicament. Voir [SMED_COD_TENR_MED](#).
 - Format : Résultats par format d'acquisition du médicament. Voir [SMED_COD_FORMA_ACQ_MED](#).
- **Codes de Service** : Exclusion ou inclusion des codes de services. Il n'y aura pas de filtre si aucune case n'est cochée. Voir [SMED_COD_SERV_1](#).
- **Code Liste Médicament** : Exclusion ou inclusion des codes de catégories de listes de médicaments sélectionnés. Il n'y aura pas de filtre si aucune case n'est cochée. Voir [SMED_COD_CATG_LISTE_MED](#).
- **Bouton Exécuter Requête** : Exécution de la requête SQL à partir des arguments sélectionnés. Une fois complété, le tableau des résultats s'affiche sous la section *Résultats*.
- **Bouton Réinitialiser Arguments** : Réinitialisation complète de la page (arguments par défaut).

4.2 Résultats

- Tableau des résultats en fonction des arguments sélectionnés.
- **Bouton Sauvegarder Résultats en Excel** : Ouvre une fenêtre permettant de sauvegarder au format Excel le tableau des résultats. Il faut choisir le répertoire de sauvegarde et le nom du fichier.

4.3 Requête SQL

- Code SQL utilisé pour créer le tableau des résultats en fonction des arguments sélectionnés. Seul le code de la première période est affiché.

4.4 Exemples

- Voir section *Requête via Excel > Statistiques générales > Exemples*.