

# Package ‘inesss’

February 5, 2021

**Title** Institut National Excellence Sante Services Sociaux

**Version** 0.5.0

**Description** Cette librairie fournit des fonctionnalités pour une variété de tâches propices au domaine de la santé et des outils pour visualiser les résultats.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.1.1

**Imports** askpass (>= 1.1),  
data.table (>= 1.13.4),  
DBI (>= 1.1.0),  
fs (>= 1.5.0),  
knitr (>= 1.30),  
lubridate (>= 1.7.9.2),  
miniUI (>= 0.1.1.1),  
parallel (>= 4.0.3),  
odbc (>= 1.3.0),  
readxl (>= 1.3.1),  
rmarkdown (>= 2.6),  
rstudioapi (>= 0.13),  
shiny (>= 1.5.0),  
shinydashboard (>= 0.7.1),  
shinyFiles (>= 0.9.0),  
stringr (>= 1.4.0),  
testthat (>= 3.0.1),  
writexl (>= 1.3.1)

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

## R topics documented:

comorbidity . . . . .	2
Comorbidity_diagn_codes . . . . .	3
Comorbidity_weights . . . . .	4
formulaire . . . . .	4

Pop_QC . . . . .	5
query_stat_gen1 . . . . .	6
RLS_convert . . . . .	8
RLS_list . . . . .	9
SQL_comorbidity . . . . .	9
SQL_comorbidity_diagn . . . . .	11
SQL_connexion . . . . .	13
SQL_obstetric . . . . .	13
SQL_stat_gen1 . . . . .	14
V_DEM_PAINT_MED_CM.SMED_COD_DIN . . . . .	16
V_DEM_PAINT_MED_CM.SMED_COD_SERV . . . . .	17
V_DENOM_COMNE_MED . . . . .	18
V_DES_COD . . . . .	18
V_PRODU_MED.NMED_NOM_MARQ_COMRC . . . . .	19

<b>Index</b>	<b>20</b>
--------------	-----------

---

comorbidity	<i>Comorbidity</i>
-------------	--------------------

---

## Description

Calcul des indicateurs de *Charlson*, *Elixhauser* et la combinaison des deux.

## Usage

```
comorbidity(
  dt,
  ID,
  DIAGN,
  DATE_DX,
  SOURCE,
  n1 = 30,
  n2 = 730,
  method = c("Charlson", "Elixhauser"),
  scores = "CIM10",
  confirm_sourc = list(`MED-ECHO` = 1, BDCU = 2, SMOD = 2)
)
```

## Arguments

dt	Dataset ayant au moins les quatre (4) colonnes ID, DIAGN, DATE_DX et SOURCE.
ID	Nom de la colonne indiquant le numéro de l'utilisateur, de l'individu.
DIAGN	Nom de la colonne indiquant le code d'un diagnostic. Voir <code>names(inesss::Comorbidity_diagn_codes)</code> .
DATE_DX	Nom de la colonne indiquant la date du diagnostic.
SOURCE	Nom de la colonne indiquant la provenance du diagnostic.
n1, n2	Nombre de jours dans le but de construire l'intervalle [n1,n2]. Pour qu'un code de diagnostic soit confirmé, il faut que <i>DIAGNi</i> soit suivi de <i>DIAGNj</i> (où $i < j$ ) et que le nombre de jours entre les deux soit dans l'intervalle [n1,n2].

method	Méthode de calcul des indicateurs. 'Charlson' ou 'Elixhauser'. Inscire les deux crée la colonne Combined.
scores	'CIM9' ou 'CIM10'. Nom de la colonne du dataset Comorbidity_weights à utiliser pour le calcul des indicateurs.
confirm_sourc	list indiquant la <i>confiance</i> des SOURCE. Si une SOURCE doit être confirmée par une autre dans l'intervalle [n1,n2], inscrire 2, sinon 1. Inscire les sources sous le format : confirm_sourc = list(source1 = 1, source2 = 2, source3 = 2, ...). confirm_sourc doit contenir toutes les valeurs uniques de la colonne SOURCE.

### Details

confirm\_sourc : Dans l'exemple confirm\_sourc = list(source1=1, source2=2, source3=2, ...), la source3 pourrait confirmer la source2 et vice-versa.

### Value

data.table avec les colonnes ID, Charlson (selon method), Elixhauser (selon method), Combined (selon method), et tous les codes de diagnosti indiquant leur poids.

---

Comorbidity\_diagn\_codes

*Data - Codes diagnostics*

---

### Description

Codes SQL regex (se terminent par un '%') à utiliser lors de l'extraction des codes de diagnostics pour l'étude de la comorbidité.

### Usage

Comorbidity\_diagn\_codes

Charlson\_diagn\_codes

Elixhauser\_diagn\_codes

### Format

list(nom\_diagn = list(CIM9, CIM10)). 32 diagnostics.

list(nom\_diagn = list(CIM9, CIM10)). 16 diagnostics.

list(nom\_diagn = list(CIM9, CIM10)). 29 diagnostics.

### Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

### Source

Validation of the Combined Comorbidity Index of Charlson and Elixhauser to Predict 30-Day Mortality Across ICD-9 and ICD-10. Voir PDF.

---

Comorbidity\_weights      *Data - Poids des codes de diagnostics*

---

### Description

Data - Poids des codes de diagnostics

### Usage

```
data('Comorbidity_weights') / MaVariable <- Comorbidity_weights
```

### Format

Tableau de 4 variables et 32 observations.

**DIAGN** Identification du diagnostic (chr).

**DIAGN\_CODE** Code de diagnostic utilisé dans la programmation (chr).

**CIM9** Poids utilisé pour la 9<sup>e</sup> révision du CIM (int).

**CIM10** Poids utilisé pour la 10<sup>e</sup> révision du CIM (int).

### Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

### Source

Validation of the Combined Comorbidity Index of Charlson and Elixhauser to Predict 30-Day Mortality Across ICD-9 and ICD-10. Voir PDF.

---

formulaire

*Formulaire Interactif*

---

### Description

Permet d'exécuter des requêtes Excel à partir d'un formulaire interactif. La documentation complète du formulaire, *AIDE\_FORMULAIRE\_DATE.pdf*, est disponible [ici](#).

### Usage

```
formulaire()
```

### Details

#### Requêtes via Excel :

Il est conseillé d'utiliser les gabarits Excel pour éviter des erreurs de structures dans les tableaux d'arguments. Les fichiers Excel sont disponibles [ici](#).

---

Pop_QC	<i>Data - Estimations et projections de population comparables (1996-2041)</i>
--------	--

---

## Description

Tableau de la population québécoise par niveau géographique.

Ce fichier présente une série continue de données populationnelles comparables composée de la série des estimations (1996-2019) et de la série des projections (2020-2041) de population. Ces données tiennent compte de l'évolution de la population selon les plus récentes données observées de naissances, décès et mouvements migratoires.

Il est à noter que ces données de population sont présentées sur la base du découpage territorial du réseau de la santé et des services sociaux, soit pour les territoires suivants : le Québec, les réseaux universitaires intégrés de santé et de services sociaux (RUISSS), les régions sociosanitaires (RSS), les réseaux territoriaux de services (RTS), les réseaux locaux de services (RLS) et les centres locaux de services communautaires (CLSC).

## Usage

```
data('Pop_QC') / MaVariable <- Pop_QC.
```

## Format

Tableau de 8 variables et 2 595 320 observations :

**GEO** Niveau géographique : Québec, RUISSS, RSS, RTS, RLS, CLSC.

**CODE** Code du territoire.

**AN** Année.

**TYPE** Type de données : Estimations ou Projections.

**STATUT** Donnée révisée ou provisoire. NA indique que la donnée n'a pas été changée depuis la dernière publication.

**SEXE**

**AGE**

**POP** Population

## Details

La classe des colonnes est character lorsque c'est du texte ou integer lorsque c'est un nombre.

**Mise en ligne** : 25 février 2016.

**Dernière modification** : 24 avril 2020.

**Publication no** : EstimProjComp-ISQ.

La fiche d'information et technique de cette base de données est disponible avec le fichier Excel (voir *Source*).

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

## Source

MSSS Données de population.  
Fichier Excel utilisé.

---

query_stat_gen1	<i>Code SQL</i>
-----------------	-----------------

---

## Description

Générateur de code SQL pour la méthode stat\_gen1.

## Usage

```
query_stat_gen1(
  debut,
  fin,
  type_Rx = "DENOM",
  codes,
  group_by = "Codes",
  code_serv = c("1", "AD"),
  code_serv_filtre = "Exclusion",
  code_list = NULL,
  code_list_filtre = "Inclusion",
  ...
)
```

## Arguments

debut	Date de début de la période d'étude au format AAAA-MM-JJ (une seule valeur).
fin	Date de fin de la période d'étude au format AAAA-MM-JJ (une seule valeur).
type_Rx	Indique le type de code analysé : <ul style="list-style-type: none"> <li>'DENOM' : Code de dénomination commune (SMED_COD_DENOM_COMNE).</li> <li>'DIN' : Code d'identification du médicament (SMED_COD_DIN).</li> </ul>
codes	Vecteur comprenant le ou les codes d'analyse au format numérique, sans zéros.
group_by	Regrouper (aggréger) les résultats par : <ul style="list-style-type: none"> <li>'Codes' : Résultats par code analysé.</li> <li>'Teneur' : Résultats par teneur de médicament (SMED_COD_TENR_MED) incluant les valeurs absentes.</li> <li>'Format' : Résultats par format d'acquisition du médicament (SMED_COD_FORMA_ACQ_MED) incluant les valeurs absentes.</li> </ul>
code_serv	Vecteur de type character comprenant le ou les codes de service (SMED_COD_SERV_1) à exclure ou à inclure, sinon inscrire NULL.
code_serv_filtre	'Inclusion' ou 'Exclusion' des codes de service code_serv. Incrire code_serv = NULL s'il n'y a pas de filtre à appliquer.
code_list	Vecteur de type character comprenant le ou les codes de catégories de liste de médicaments (SMED_COD_CATG_LISTE_MED) à exclure ou à inclure, sinon inscrire NULL.

code\_list\_filtre

'Inclusion' ou 'Exclusion' des codes de catégories de liste de médicaments  
code\_list. Inscrire code\_list = NULL s'il n'y a pas de filtre à appliquer.

## Details

### Méthode stat\_gen1 :

Statistiques descriptives tirées de la vue V\_DEM\_PAINT\_MED\_CM.

group\_by : Revient à utiliser la commande group by dans le code SQL.

code\_serv\_filtre, code\_list\_filtre :

"Exclusion" : Inclus les NULL, 'Inclusion' exclus les NULL.

## Value

Chaîne de caractères à utiliser dans une requête SQL.

## Examples

```
### Avantages d'utiliser cat() si c'est pour afficher du code dans la console.
# Avec cat()
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222), group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))
# Sans cat()
query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222), group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
)

### GROUP_BY
# group_by = NULL : resultats par periode d'etude.
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = NULL,
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))
# group_by = 'Codes' : resultats par code d'analyse.
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))
# group_by = c('Teneur', 'Format') : resultats par teneur et format.
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = c('Teneur', 'Format'),
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion'
))
```

```
### Exclusion VS Inclusion
cat(query_stat_gen1(
  debut = '2020-01-01', fin = '2020-12-31',
  type_Rx = 'DENOM', codes = c(47092, 47135, 48222),
  group_by = NULL,
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion',
  code_list = c('40', '41'), code_list_filtre = 'Inclusion'
))
```

RLS\_convert

*Data - Correspondance RLS Loi 10*

## Description

Établir la correspondance des RLS avant et après l'adoption de la loi 10.

## Usage

```
data('RLS_convert') / MaVariable <- RLS_convert.
```

## Format

Tableau de 2 variables et 84 observations :

**RLS14** Code de RLS **avant** l'adoption de la loi 10.

**RLS15** Code de RLS **après** l'adoption de la loi 10.

## Details

Certains RLS ne peuvent être convertis, car leur valeur se retrouve avant et après l'adoption de la loi 10.

`attr(RLS_convert, "RLS_exclus")` indique les quatre (4) RLS exclus : 611, 612, 1611, 1612.

`attr(RLS_convert, "RLS_exclus_value")` renvoie un tableau indiquant les valeurs avant et après l'adoption de la loi 10 pour ces quatre (4) RLS.

L'attribut `MaJ` indique la dernière mise à jour ou la date de création du tableau.

## Source

Correspondance Etablissement Public Loi 10.

Fichier Excel utilisé.



---

RLS_list	<i>Data - Liste des RLS</i>
----------	-----------------------------

---

### Description

Vecteur contenant la liste des 93 RLS plus 3 valeurs utiles lors d'analyse : 1001, 1701, 1801.

### Usage

```
data('RLS_list') / MaVariable <- RLS_list.
```

### Format

Vecteur integer de 96 nombres.

### Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

---

SQL_comorbidity	<i>Comorbidity</i>
-----------------	--------------------

---

### Description

Extraction des codes de diagnostics CIM pour ensuite calculer les indicateurs de Charlson et Elixhauser.

### Usage

```
SQL_comorbidity(
  conn,
  uid,
  pwd,
  dt,
  ID,
  DATE_INDEX,
  method = c("Charlson", "Elixhauser"),
  CIM = c("CIM9", "CIM10"),
  scores = "CIM10",
  lookup = 2,
  n1 = 30,
  n2 = 730,
  dt_source = c("V_DIAGN_SEJ_HOSP_CM", "V_SEJ_SERV_HOSP_CM", "V_EPISO_SOIN_DURG_CM",
    "I_SMOD_SERV_MD_CM"),
  dt_desc = list(V_DIAGN_SEJ_HOSP_CM = "MED-ECHO", V_SEJ_SERV_HOSP_CM = "MED-ECHO",
    V_EPISO_SOIN_DURG_CM = "BDCU", I_SMOD_SERV_MD_CM = "SMOD"),
  confirm_sourc = list(`MED-ECHO` = 1, BDCU = 2, SMOD = 2),
  obstetric_exclu = TRUE,
  verbose = TRUE
)
```

## Arguments

conn	Variable contenant la connexion entre R et Teradata. Voir <a href="#">SQL_connexion</a> . Évite d'utiliser les arguments uid et pwd.
uid	Nom de l'identifiant pour la connexion SQL Teradata.
pwd	Mot de passe associé à l'identifiant. Si NULL, le programme demande le mot passe. Cela permet de ne pas afficher le mot de passe dans un script.
dt	Tableau ayant au moins deux colonnes : ID et DATE_INDEX.
ID	Nom de la colonne contenant l'identifiant unique de l'utilisateur.
DATE_INDEX	Nom de la colonne contenant la date index de chaque usager.
method	Extraire les diagnostics associés aux méthodes de calcul <i>Charlson</i> , <i>Elixhauser</i> ou les deux.
CIM	'CIM9', 'CIM10' ou les deux. Permet de filtrer les codes de diagnostics selon le numéro de révision de la <i>Classification statistique internationale des maladies et des problèmes de santé connexes</i> (CIM).
scores	'CIM9' ou 'CIM10'. Nom de la colonne du dataset Comorbidity_weights à utiliser pour le calcul des indicateurs.
lookup	Nombre entier. Années à analyser avant la date index de chaque individu.
n1	Nombre de jours dans le but de construire l'intervalle [n1,n2]. Pour qu'un code de diagnostique soit confirmé, il faut que <i>DIAGNi</i> soit suivi de <i>DIAGNj</i> (où $i < j$ ) et que le nombre de jours entre les deux soit dans l'intervalle [n1,n2].
n2	Nombre de jours dans le but de construire l'intervalle [n1,n2]. Pour qu'un code de diagnostique soit confirmé, il faut que <i>DIAGNi</i> soit suivi de <i>DIAGNj</i> (où $i < j$ ) et que le nombre de jours entre les deux soit dans l'intervalle [n1,n2].
dt_source	Vecteur comprenant la ou les bases de données où aller chercher l'information. Voir <i>Details</i> .
dt_desc	list décrivant les bases de données demandées dans dt_source au format list(BD = 'MaDescription'). Voir <i>Details</i> .
confirm_sourc	list indiquant la <i>confiance</i> des SOURCE. Si une SOURCE doit être confirmée par une autre dans l'intervalle [n1,n2], inscrire 2, sinon 1. Inscrire les sources sous le format : confirm_sourc = list(source1 = 1, source2 = 2, source3 = 2, ...). confirm_sourc doit contenir toutes les valeurs uniques de la colonne SOURCE.
obstetric_exclu	TRUE ou FALSE. Si l'on doit exclure (TRUE) les diabètes et les hypertension de type gestationnel. Voir <i>Détails</i> .
verbose	TRUE ou FALSE. Affiche le temps qui a été nécessaire pour extraire les diagnostics d'une source (dt_source). Utile pour suivre le déroulement de l'extraction.

## Details

conn, uid, pwd : Pour se connecter à Teradata, utiliser conn ou la combinaison uid et pwd.

dt : Si un ID a plus d'une date index, seule la première, la plus ancienne, sera conservée.

obstetric\_exclu : Lorsqu'un cas de diabète ou d'hypertension a lieu 120 jours avant ou 180 jours après un événement obstétrique, on les considère de type gestationnel. Ces cas sont alors exclus de l'analyse.

dt\_source :

- **V\_DIAGN\_SEJ\_HOSP\_CM** : Cette structure contient tous les diagnostics associés à un séjour hospitalier.
- **V\_SEJ\_SERV\_HOSP\_CM** : Cette structure contient les séjours dans un service effectués par l'individu hospitalisé.
- **V\_EPISO\_SOIN\_DURG\_CM** : Cette structure contient les épisodes de soins des départements d'urgence de la province.
- **I\_SMOD\_SERV\_MD\_CM** : Cette vue retourne différentes informations se rapportant aux Services rendus à l'acte par des médecins.

## Value

data.table :

- ID : Colonne contenant l'identifiant unique de l'utilisateur.
- Charlson : Indicateur, seulement si method contient 'Charlson'.
- Elixhauser : Indicateur, seulement si method contient 'Elixhauser'.
- Combined : Indicateur, seulement si method contient 'Charlson' et 'Elixhauser'.
- Tous les diagnostics ainsi que leur poids (score).

---

SQL\_comorbidity\_diagn *Extraction - Codes diagn comorbidity*

---

## Description

Extraction SQL des diagnostics pour l'étude de la comorbidité.

## Usage

```
SQL_comorbidity_diagn(
  conn,
  uid,
  pwd,
  cohort,
  debut,
  fin,
  method = c("Charlson", "Elixhauser"),
  CIM = c("CIM9", "CIM10"),
  dt_source = c("V_DIAGN_SEJ_HOSP_CM", "V_SEJ_SERV_HOSP_CM", "V_EPISO_SOIN_DURG_CM",
    "I_SMOD_SERV_MD_CM"),
  dt_desc = list(V_DIAGN_SEJ_HOSP_CM = "MED-ECHO", V_SEJ_SERV_HOSP_CM = "MED-ECHO",
    V_EPISO_SOIN_DURG_CM = "BDCU", I_SMOD_SERV_MD_CM = "SMOD"),
  verbose = TRUE
)
```

## Arguments

conn	Variable contenant la connexion entre R et Teradata. Voir <a href="#">SQL_connexion</a> . Évite d'utiliser les arguments uid et pwd.
uid	Nom de l'identifiant pour la connexion SQL Teradata.
pwd	Mot de passe associé à l'identifiant. Si NULL, le programme demande le mot passe. Cela permet de ne pas afficher le mot de passe dans un script.
cohort	Cohorte d'étude. Vecteur comprenant les numéros d'identification des individus à conserver.
debut	Date de début de la période d'étude au format AAAA-MM-JJ.
fin	Date de fin de la période d'étude au format AAAA-MM-JJ.
method	Extraire les diagnostics associés aux méthodes de calcul <i>Charlson</i> , <i>Elixhauser</i> ou les deux.
CIM	'CIM9', 'CIM10' ou les deux. Permet de filtrer les codes de diagnostics selon le numéro de révision de la <i>Classification statistique internationale des maladies et des problèmes de santé connexes</i> (CIM).
dt_source	Vecteur comprenant la ou les bases de données où aller chercher l'information. Voir <i>Details</i> .
dt_desc	list décrivant les bases de données demandées dans dt_source au format list(BD = 'MaDescription'). Voir <i>Details</i> .
verbose	TRUE ou FALSE. Affiche le temps qui a été nécessaire pour extraire les diagnostics d'une source (dt_source). Utile pour suivre le déroulement de l'extraction.

## Details

conn, uid, pwd : Pour se connecter à Teradata, utiliser conn ou la combinaison uid et pwd.

dt\_source :

- **V\_DIAGN\_SEJ\_HOSP\_CM** : Cette structure contient tous les diagnostics associés à un séjour hospitalier.
- **V\_SEJ\_SERV\_HOSP\_CM** : Cette structure contient les séjours dans un service effectués par l'individu hospitalisé.
- **V\_EPISO\_SOIN\_DURG\_CM** : Cette structure contient les épisodes de soins des départements d'urgence de la province.
- **I\_SMOD\_SERV\_MD\_CM** : Cette vue retourne différentes informations se rapportant aux Services rendus à l'acte par des médecins.

method : Voir les listes [Comorbidity\\_diagn\\_codes](#) pour connaître les codes de diagnostics extraits.

## Value

data.table de 4 variables :

- ID : Numéro d'identification de l'utilisateur.
- DATE\_DX : Date de diagnostic.
- DIAGN : Code descriptif des diagnostics provenant de `diagn_codes`.
- SOURCE : Indique d'où provient l'information. Une valeur parmi dt\_source.

---

SQL_connexion	<i>Connexion Teradata</i>
---------------	---------------------------

---

**Description**

Connexion entre R et SQL Teradata.

**Usage**

```
SQL_connexion(uid, pwd = NULL, dsn = "PEI_PRD", encoding = "latin1")
```

**Arguments**

uid	Identifiant.
pwd	Mot de passe. Si NULL, le mot de passe est demandé lors de l'exécution.
dsn	<b>Data Source Name.</b> Par défaut 'PEI_PRD'.
encoding	'latin1' ou 'UTF-8'. Encodage de la base de données. Par défaut 'latin1'.

**Details**

Voir ?odbc::dbConnect.

**Value**

Connexion Teradata, sinon NULL.

**Examples**

```
## Not run:
conn <- SQL_connexion('abc007')
conn <- SQL_connexion(uid = 'abc007', pwd = 'MonMotDePasse', dsn = 'PEI_PRD')

## End(Not run)
```

---

SQL_obstetric	<i>Extraction - Codes diagn obstetriques</i>
---------------	--

---

**Description**

Extraction des événements obstétriques.

**Usage**

```
SQL_obstetric(
  conn,
  uid,
  pwd,
  cohort,
  debut,
  fin,
  CIM = c("CIM9", "CIM10"),
  dt_source = c("V_DIAGN_SEJ_HOSP_CM", "V_SEJ_SERV_HOSP_CM", "V_EPISO_SOIN_DURG_CM",
    "I_SMOD_SERV_MD_CM"),
  dt_desc = list(V_DIAGN_SEJ_HOSP_CM = "MED-ECHO", V_SEJ_SERV_HOSP_CM = "MED-ECHO",
    V_EPISO_SOIN_DURG_CM = "BDCU", I_SMOD_SERV_MD_CM = "SMOD"),
  verbose = TRUE
)
```

**Details**

dt\_source :

- **V\_DIAGN\_SEJ\_HOSP\_CM** : Cette structure contient tous les diagnostics associés à un séjour hospitalier.
- **V\_SEJ\_SERV\_HOSP\_CM** : Cette structure contient les séjours dans un service effectués par l'individu hospitalisé.
- **V\_EPISO\_SOIN\_DURG\_CM** : Cette structure contient les épisodes de soins des départements d'urgence de la province.
- **I\_SMOD\_SERV\_MD\_CM** : Cette vue retourne différentes informations se rapportant aux Services rendus à l'acte par des médecins.

---

SQL\_stat\_gen1

*Statistiques générales*

---

**Description**

Tableau indiquant les statistiques générales d'un ou de plusieurs codes de médicaments selon certains critères.

**Usage**

```
SQL_stat_gen1(
  conn,
  uid,
  pwd,
  debut,
  fin,
  type_Rx = "DENOM",
  codes,
  group_by = "Codes",
  code_serv = c("1", "AD"),
  code_serv_filtre = "Exclusion",
```

```

    code_list = NULL,
    code_list_filtre = "Inclusion",
    ...
)

```

## Arguments

conn	Variable contenant la connexion entre R et Teradata. Voir <a href="#">SQL_connexion</a> . Évite d'utiliser les arguments uid et pwd.
uid	Nom de l'identifiant pour la connexion SQL Teradata.
pwd	Mot de passe associé à l'identifiant. Si NULL, le programme demande le mot passe. Cela permet de ne pas afficher le mot de passe dans un script.
debut	Date de début de la période d'étude au format AAAA-MM-JJ (une seule valeur).
fin	Date de fin de la période d'étude au format AAAA-MM-JJ (une seule valeur).
type_Rx	Indique le type de code analysé : <ul style="list-style-type: none"> <li>'DENOM' : Code de dénomination commune (SMED_COD_DENOM_COMNE).</li> <li>'DIN' : Code d'identification du médicament (SMED_COD_DIN).</li> </ul>
codes	Vecteur comprenant le ou les codes d'analyse au format numérique, sans zéros.
group_by	Regrouper (aggréger) les résultats par : <ul style="list-style-type: none"> <li>'Codes' : Résultats par code analysé.</li> <li>'Teneur' : Résultats par teneur de médicament (SMED_COD_TENR_MED) incluant les valeurs absentes.</li> <li>'Format' : Résultats par format d'acquisition du médicament (SMED_COD_FORMA_ACQ_MED) incluant les valeurs absentes.</li> </ul>
code_serv	Vecteur de type character comprenant le ou les codes de service (SMED_COD_SERV_1) à exclure ou à inclure, sinon inscrire NULL.
code_serv_filtre	'Inclusion' ou 'Exclusion' des codes de service code_serv. Inscrire code_serv = NULL s'il n'y a pas de filtre à appliquer.
code_list	Vecteur de type character comprenant le ou les codes de catégories de liste de médicaments (SMED_COD_CATG_LISTE_MED) à exclure ou à inclure, sinon inscrire NULL.
code_list_filtre	'Inclusion' ou 'Exclusion' des codes de catégories de liste de médicaments code_list. Inscrire code_list = NULL s'il n'y a pas de filtre à appliquer.

## Details

Pour se connecter à Teradata, utiliser conn ou la combinaison uid et pwd.

## Exemples

```

## Not run:
### GROUP_BY
# Aucun group by
dt1 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = NULL
)

```

```

# Codes - DENOM
dt2 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = 'Codes'
)
# Codes - DIN
dt3 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DIN', codes = c(30848, 585092), group_by = 'Codes'
)
# Codes et Format
dt4 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = c('Codes', 'Format')
)
# Teneur
dt5 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = 'Teneur'
)

### Exclusion & Inclusion
dt6 <- SQL_stat_gen1(
  conn = conn,
  debut = c('2017-01-01', '2018-01-01'), fin = c('2017-12-31', '2018-12-31'),
  type_Rx = 'DENOM', codes = c(39, 47092, 48222), group_by = 'Codes',
  code_serv = c('1', 'AD'), code_serv_filtre = 'Exclusion',
  code_list = c('03', '40', '41'), code_list_filtre = 'Inclusion'
)

## End(Not run)

```

---

V\_DEM\_PAIMT\_MED\_CM.SMED\_COD\_DIN

*Data - COD\_DIN*


---

## Description

Codes d'identification des médicaments (SMED\_COD\_DIN) qui sont présents dans la vue V\_DEM\_PAIMT\_MED\_CM.

## Usage

V\_DEM\_PAIMT\_MED\_CM.SMED\_COD\_DIN

## Format

Tableau de 3 variables :

**DIN** Code d'identification du médicament (SMED\_COD\_DIN). integer.

**DEBUT** Première année où le code a été inscrit. integer.

**FIN** Dernière année où le code a été inscrit. integer.



**Details**

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

**Source**

V\_DEM\_PAINT\_MED\_CM.

---

V\_DEM\_PAINT\_MED\_CM. SMED\_COD\_SERV  
*Data - COD\_SERV*

---

**Description**

Codes de services (SMED\_COD\_SERV\_X) qui sont présent dans la vue V\_DEM\_PAINT\_MED\_CM.

**Usage**

V\_DEM\_PAINT\_MED\_CM. SMED\_COD\_SERV

**Format**

Tableau de 5 variables :

**COD\_SERV** Codes de services pouvant être inscrit dans les colonnes SMED\_COD\_SERV\_1, SMED\_COD\_SERV\_2 et SMED\_COD\_SERV3. character.

**SERV\_1** Première et dernière année que le code de service a été inscrit dans la colonne SMED\_COD\_SERV\_1. Si NA, le code n'a jamais été inscrit dans cette colonne. character.

**SERV\_2** Première et dernière année que le code de service a été inscrit dans la colonne SMED\_COD\_SERV\_2. Si NA, le code n'a jamais été inscrit dans cette colonne. character.

**SERV\_3** Première et dernière année que le code de service a été inscrit dans la colonne SMED\_COD\_SERV\_3. Si NA, le code n'a jamais été inscrit dans cette colonne. character.

**COD\_SERV\_DESC** Description du code de service tirée de la variable COD\_SERV\_DESC de la vue V\_PARAM\_SERV\_MED. character.

**Details**

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

**Source**

V\_DEM\_PAINT\_MED\_CM.

---

V_DENOM_COMNE_MED	<i>Data</i>
-------------------	-------------

---

### Description

Description des codes de dénomination commune.

### Usage

V\_DENOM\_COMNE\_MED

### Format

Tableau de 7 variables :

**DENOM** Code de dénomination commune (NMED\_COD\_DENOM\_COMNE). character.

**DATE\_DEBUT** Date à laquelle cette dénomination commune est apparue pour la première fois (NMED\_DD\_DENOM\_COMNE). Date.

**DATE\_FIN** Date à laquelle la dénomination commune a cessé d'être utilisée (NMED\_DF\_DENOM\_COMNE). Date.

**NOM\_DENOM** Nom de la dénomination commune du médicament (NMED\_NOM\_DENOM\_COMNE). character.

**NOM\_DENOM\_SYNON** Synonyme du nom de la dénomination commune du médicament (NMED\_NOM\_DENOM\_COMNE\_SYNON). character.

**NOM\_DENOM\_ANGLAIS** Nom anglais de la dénomination commune du médicament (NMED\_NOM\_ANGL\_DENOM\_COMNE). character.

**NOM\_DENOM\_SYNON\_ANGLAIS** Synonyme du nom anglais de la dénomination commune du médicament (NMED\_NOM\_ANGL\_DENOM\_SYNON). character.

### Details

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

### Source

V\_DENOM\_COMNE\_MED.

---

V_DES_COD	<i>Data</i>
-----------	-------------

---

### Description

Domaine de valeurs pour les différents codes de l'environnement informationnel.

### Usage

V\_DES\_COD

**Format**

Tableau de 5 variables :

**CODE** Valeurs codifiées que peut prendre un élément (CODE\_VAL\_COD). character.

**TYPE\_CODE** Nom identifiant un élément de données (CODE\_NOM\_COD). character.

**CODE\_DESC** Description du code (CODE\_DES). character.

**DATE\_DEBUT** Date de début de la période d'application (CODE\_DD\_DES\_COD). Date.

**DATE\_FIN** Date de fin de la période d'application (CODE\_DF\_DES\_COD). Date.

**Details**

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

**Source**

V\_DES\_COD.

---

V\_PRODU\_MED.NMED\_NOM\_MARQ\_COMRC

*Data - NOM\_MARQ\_COMRC*

---

**Description**

Nom sous lequel est commercialisé un produit pharmaceutique.

**Usage**

V\_PRODU\_MED.NMED\_NOM\_MARQ\_COMRC

**Format**

Tableau de 5 variables :

**DENOM** Code de dénomination commune (NMED\_COD\_DENOM\_COMNE). character.

**DIN** Code d'identification du médicament (NMED\_COD\_DIN). integer.

**NOM\_MARQ\_COMRC** Nom sous lequel est commercialisé un produit pharmaceutique (NMED\_NOM\_MARQ\_COMRC). character.

**DATE\_DEBUT** Date de début d'une occurrence de cette table (NMED\_DD\_PRODU\_MED). Date.

**DATE\_FIN** Date de fin d'une occurrence de cette table (NMED\_DF\_PRODU\_MED). Date.

**Details**

L'attribut MaJ indique la dernière mise à jour ou la date de création du tableau.

**Source**

V\_PRODU\_MED.

# Index

## \* datasets

- Comorbidity\_diagn\_codes, [3](#)
- Comorbidity\_weights, [4](#)
- Pop\_QC, [5](#)
- RLS\_convert, [8](#)
- RLS\_list, [9](#)
- V\_DEM\_PAINT\_MED\_CM.SMED\_COD\_DIN, [16](#)
- V\_DEM\_PAINT\_MED\_CM.SMED\_COD\_SERV, [17](#)
- V\_DENOM\_COMNE\_MED, [18](#)
- V\_DES\_COD, [18](#)
- V\_PRODU\_MED.NMED\_NOM\_MARQ\_COMRC, [19](#)

Charlson\_diagn\_codes  
(Comorbidity\_diagn\_codes), [3](#)  
comorbidity, [2](#)  
Comorbidity\_diagn\_codes, [3](#), [12](#)  
Comorbidity\_weights, [4](#)

Elixhauser\_diagn\_codes  
(Comorbidity\_diagn\_codes), [3](#)

formulaire, [4](#)

Pop\_QC, [5](#)

query\_stat\_gen1, [6](#)

RLS\_convert, [8](#)  
RLS\_list, [9](#)

SQL\_comorbidity, [9](#)  
SQL\_comorbidity\_diagn, [11](#)  
SQL\_connexion, [10](#), [12](#), [13](#), [15](#)  
SQL\_obstetric, [13](#)  
SQL\_stat\_gen1, [14](#)

V\_DEM\_PAINT\_MED\_CM.SMED\_COD\_DIN, [16](#)  
V\_DEM\_PAINT\_MED\_CM.SMED\_COD\_SERV, [17](#)  
V\_DENOM\_COMNE\_MED, [18](#)  
V\_DES\_COD, [18](#)  
V\_PRODU\_MED.NMED\_NOM\_MARQ\_COMRC, [19](#)