**Sybase PowerDesigner**

**Physical Data Model Report**

***Model: soat[pg]***

***Report: Report 1***



Author: andres.sandoval

Version:

Date: 19/08/2016

Summary:

**Table of Contents**

**The 'Table of Contents' field needs to be updated!**

# Introduction

## Description

## Card of model soat[pg]

|  |  |
| --- | --- |
| Name | soat[pg] |
| Code | soat\_pg\_ |
| DBMS | PostgreSQL 9 |
| Comment |  |
| Database | <None> |

# Short model description

## List of diagrams

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |
| sys.kernel | sys\_kernel |

## Diagram app.soat



## Diagram sys.kernel



## List of tables

|  |  |
| --- | --- |
| Name | Code |
| \_action\_ | \_action\_ |
| \_ambulance\_use\_ | \_ambulance\_use\_ |
| \_app\_ | \_app\_ |
| \_canton\_ | \_canton\_ |
| \_catalog\_ | \_catalog\_ |
| \_department\_ | \_department\_ |
| \_doctor\_ | \_doctor\_ |
| \_enterprise\_ | \_enterprise\_ |
| \_expense\_form\_ | \_expense\_form\_ |
| \_expense\_form\_medicine\_detail\_ | \_expense\_form\_medicine\_detail\_ |
| \_expense\_form\_tariff\_detail\_ | \_expense\_form\_tariff\_detail\_ |
| \_form008\_has\_doctor\_ | \_form008\_has\_doctor\_ |
| \_form\_008\_ | \_form\_008\_ |
| \_fur\_ | \_fur\_ |
| \_medicine\_ | \_medicine\_ |
| \_objection\_ | \_objection\_ |
| \_oxygen\_use\_ | \_oxygen\_use\_ |
| \_parish\_ | \_parish\_ |
| \_patient\_ | \_patient\_ |
| \_profile\_ | \_profile\_ |
| \_profile\_has\_action\_ | \_profile\_has\_action\_ |
| \_province\_ | \_province\_ |
| \_tariff\_ | \_tariff\_ |
| \_user\_ | \_user\_ |

## List of references

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_canton\_references\_province\_\_ | \_\_fk\_canton\_references\_province\_\_ | \_province\_ | \_canton\_ | province\_identifier |  |  |
| \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_catalog\_ | \_doctor\_ | speciality\_identifier |  |  |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_expense\_form\_ | \_expense\_form\_medicine\_detail\_ | expense\_form\_identifier |  |  |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_medicine\_ | \_expense\_form\_medicine\_detail\_ | item\_identifier |  |  |
| \_\_fk\_expense\_form\_references\_form008\_\_ | \_\_fk\_expense\_form\_references\_form008\_\_ | \_form\_008\_ | \_expense\_form\_ | form008\_identifier |  |  |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_expense\_form\_ | \_expense\_form\_tariff\_detail\_ | expense\_form\_identifier |  |  |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_tariff\_ | \_expense\_form\_tariff\_detail\_ | item\_identifier |  |  |
| \_\_fk\_form008\_references\_patient\_\_ | \_\_fk\_form008\_references\_patient\_\_ | \_patient\_ | \_form\_008\_ | patient\_identifier |  |  |
| \_\_fk\_parish\_references\_canton\_\_ | \_\_fk\_parish\_references\_canton\_\_ | \_canton\_ | \_parish\_ | canton\_identifer |  |  |
| \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_doctor\_ | \_form008\_has\_doctor\_ | doctor\_identifier |  |  |
| \_\_form008\_has\_doctor\_references\_form008\_\_ | \_\_form008\_has\_doctor\_references\_form008\_\_ | \_form\_008\_ | \_form008\_has\_doctor\_ | form008\_identifier |  |  |
| \_\_fur\_references\_form008\_\_ | \_\_fur\_references\_form008\_\_ | \_form\_008\_ | \_fur\_ | form008\_identifier |  |  |
| \_\_fur\_references\_medic\_authorizer\_\_ | \_\_fur\_references\_medic\_authorizer\_\_ | \_doctor\_ | \_fur\_ | insured\_dead\_authorizer\_doctor |  |  |
| \_\_fur\_references\_medic\_in\_charge\_\_ | \_\_fur\_references\_medic\_in\_charge\_\_ | \_doctor\_ | \_fur\_ | insured\_dead\_in\_charge\_doctor |  |  |
| \_\_objection\_references\_form008\_\_ | \_\_objection\_references\_form008\_\_ | \_form\_008\_ | \_objection\_ | form008\_identifier |  |  |
| \_fk\_action\_reference\_action\_ | \_fk\_action\_reference\_action\_ | \_action\_ | \_action\_ | root\_identifier |  |  |
| \_fk\_action\_references\_app\_ | \_fk\_action\_references\_app\_ | \_app\_ | \_action\_ | app\_identifier |  |  |
| \_fk\_ambulance\_use\_references\_form008\_ | \_fk\_ambulance\_use\_references\_form008\_ | \_form\_008\_ | \_ambulance\_use\_ | form008\_identifier |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_tariff\_ | \_ambulance\_use\_ | ambulance\_km\_code |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_tariff\_ | \_ambulance\_use\_ | starting\_point\_code |  |  |
| \_fk\_catalog\_refereces\_app\_ | \_fk\_catalog\_refereces\_app\_ | \_app\_ | \_catalog\_ | app\_identifier |  |  |
| \_fk\_department\_references\_department\_coordination\_ | \_fk\_department\_references\_department\_coordination\_ | \_department\_ | \_department\_ | coordination\_identifier |  |  |
| \_fk\_department\_references\_enterprice\_ | \_fk\_department\_references\_enterprice\_ | \_enterprise\_ | \_department\_ | enterprise\_identifier |  |  |
| \_fk\_enterprise\_references\_app\_ | \_fk\_enterprise\_references\_app\_ | \_app\_ | \_enterprise\_ | app\_identifier |  |  |
| \_fk\_oxigen\_use\_references\_form008\_ | \_fk\_oxigen\_use\_references\_form008\_ | \_form\_008\_ | \_oxygen\_use\_ | form008\_identifier |  |  |
| \_fk\_profile\_has\_action\_references\_action\_ | \_fk\_profile\_has\_action\_references\_action\_ | \_action\_ | \_profile\_has\_action\_ | action identifier |  |  |
| \_fk\_profile\_has\_action\_references\_profile\_ | \_fk\_profile\_has\_action\_references\_profile\_ | \_profile\_ | \_profile\_has\_action\_ | profile identifier |  |  |
| \_fk\_profile\_references\_app\_ | \_fk\_profile\_references\_app\_ | \_app\_ | \_profile\_ | app identifier |  |  |
| \_fk\_user\_references\_department\_ | \_fk\_user\_references\_department\_ | \_department\_ | \_user\_ | department\_identifier |  |  |
| \_fk\_user\_references\_profile\_ | \_fk\_user\_references\_profile\_ | \_profile\_ | \_user\_ | profile identifier |  |  |

## List of views

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

## List of users

|  |  |
| --- | --- |
| Name | Code |
| app\_oper | app\_oper |
| app\_sys | app\_sys |
| inventory | inventory |
| kernel | kernel |
| soat | soat |

## List of databases

|  |  |
| --- | --- |
| Name | Code |
| app | app |

## List of tablespaces

|  |  |
| --- | --- |
| Name | Code |
| app | app |

## List of procedures

|  |  |
| --- | --- |
| Name | Code |
| change\_action\_state | change\_action\_state |
| change\_catalog\_state | change\_catalog\_state |
| change\_profile\_state | change\_profile\_state |
| change\_user\_password | change\_user\_password |
| change\_user\_state | change\_user\_state |
| get\_action | get\_action |
| get\_action\_by\_app\_id | get\_action\_by\_app\_id |
| get\_action\_by\_id | get\_action\_by\_id |
| get\_action\_script\_by\_handler | get\_action\_by\_handler |
| get\_active\_patient | get\_active\_patient |
| get\_all\_doctor | get\_all\_doctor |
| get\_all\_expense\_form\_detail | get\_all\_expense\_form\_detail |
| get\_all\_medicine | get\_all\_medicine |
| get\_all\_menu\_action\_by\_profile\_id\_root\_id | get\_all\_menu\_action\_by\_profile\_id\_root\_id |
| get\_ambulance\_use\_by\_form008\_identifier | get\_ambulance\_use\_by\_form008\_identifier |
| get\_ambulance\_use\_tariff | get\_ambulance\_use\_tariff |
| get\_app\_by\_id | get\_app\_by\_id |
| get\_asigned\_menu\_action\_by\_profile\_id\_root\_id | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| get\_catalog | get\_catalog |
| get\_catalog\_by\_id | get\_catalog\_by\_id |
| get\_catalog\_type | get\_catalog\_type |
| get\_coordination | get\_coordination |
| get\_department | get\_department |
| get\_department\_by\_enterprise\_id | get\_department\_by\_enterprise\_id |
| get\_department\_by\_id | get\_department\_by\_id |
| get\_doctor\_by\_form008\_identifier | get\_doctor\_by\_form008\_identifier |
| get\_doctor\_by\_id | get\_doctor\_by\_id |
| get\_enterprise | get\_enterprise |
| get\_expense\_form\_by\_form008\_identifier | get\_expense\_form\_by\_form008\_identifier |
| get\_form008 | get\_form008 |
| get\_form008\_by\_id | get\_form008\_by\_id |
| get\_fur\_by\_form008\_identifier | get\_fur\_by\_form008\_identifier |
| get\_objection\_by\_form008\_identifier | get\_objection\_by\_form008\_identifier |
| get\_oxygen\_use\_by\_form008\_identifier | get\_oxygen\_use\_by\_form008\_identifier |
| get\_oxygen\_use\_tariff | get\_oxygen\_use\_tariff |
| get\_patient | get\_patient |
| get\_patient\_by\_id | get\_patient\_by\_id |
| get\_patient\_by\_id2 | get\_patient\_by\_id2 |
| get\_profile | get\_profile |
| get\_profile\_by\_app\_id | get\_profile\_by\_app\_id |
| get\_profile\_by\_id | get\_profile\_by\_id |
| get\_tariff\_as\_medicine | get\_tariff\_as\_medicine |
| get\_user | get\_user |
| get\_user\_by\_id | get\_user\_by\_id |
| get\_vehicle | get\_vehicle |
| insert\_action | insert\_action |
| insert\_app | insert\_app |
| insert\_product | insert\_product |
| insert\_profile | insert\_profile |
| insert\_profile\_has\_action | insert\_profile\_has\_action |
| insert\_profile\_has\_action\_node | insert\_profile\_has\_action\_node |
| login\_user | login\_user |
| process\_ambulance\_use | process\_ambulance\_use |
| process\_catalog | process\_catalog |
| process\_department | process\_department |
| process\_enterprise | process\_enterprise |
| process\_expense\_form | process\_expense\_form |
| process\_form008 | process\_form008 |
| process\_form008\_has\_doctor | process\_form008\_has\_doctor |
| process\_fur | process\_fur |
| process\_medicine | process\_medicine |
| process\_objection | process\_objection |
| process\_oxygen\_use | process\_oxygen\_use |
| process\_patient | process\_patient |
| process\_user | process\_user |
| update\_action | update\_action |
| update\_profile | update\_profile |
| validate\_action\_access | validate\_action\_access |

# Full model description

## List of diagrams

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |
| sys.kernel | sys\_kernel |

## Diagram app.soat



## Diagram sys.kernel



## List of tables

|  |  |
| --- | --- |
| *Name* | *Code* |
| \_action\_ | \_action\_ |
| \_ambulance\_use\_ | \_ambulance\_use\_ |
| \_app\_ | \_app\_ |
| \_canton\_ | \_canton\_ |
| \_catalog\_ | \_catalog\_ |
| \_department\_ | \_department\_ |
| \_doctor\_ | \_doctor\_ |
| \_enterprise\_ | \_enterprise\_ |
| \_expense\_form\_ | \_expense\_form\_ |
| \_expense\_form\_medicine\_detail\_ | \_expense\_form\_medicine\_detail\_ |
| \_expense\_form\_tariff\_detail\_ | \_expense\_form\_tariff\_detail\_ |
| \_form008\_has\_doctor\_ | \_form008\_has\_doctor\_ |
| \_form\_008\_ | \_form\_008\_ |
| \_fur\_ | \_fur\_ |
| \_medicine\_ | \_medicine\_ |
| \_objection\_ | \_objection\_ |
| \_oxygen\_use\_ | \_oxygen\_use\_ |
| \_parish\_ | \_parish\_ |
| \_patient\_ | \_patient\_ |
| \_profile\_ | \_profile\_ |
| \_profile\_has\_action\_ | \_profile\_has\_action\_ |
| \_province\_ | \_province\_ |
| \_tariff\_ | \_tariff\_ |
| \_user\_ | \_user\_ |

### Table \_action\_

#### Card of table \_action\_

|  |  |
| --- | --- |
| Name | \_action\_ |
| Code | \_action\_ |

#### Server validation rule of the table \_action\_

%RULES%

#### Check constraint name of the table \_action\_

CKT\_\_ACTION\_

#### List of incoming references of the table \_action\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_action\_reference\_action\_ | \_fk\_action\_reference\_action\_ | \_action\_ | root\_identifier |  |  |
| \_fk\_profile\_has\_action\_references\_action\_ | \_fk\_profile\_has\_action\_references\_action\_ | \_profile\_has\_action\_ | action identifier |  |  |

#### List of outgoing references of the table \_action\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_action\_reference\_action\_ | \_fk\_action\_reference\_action\_ | \_action\_ | root\_identifier |  |  |
| \_fk\_action\_references\_app\_ | \_fk\_action\_references\_app\_ | \_app\_ | app\_identifier |  |  |

#### List of incoming traceability links of the table \_action\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_action\_ | get\_all\_menu\_action\_by\_profile\_id\_root\_id |
| \_action\_ | update\_action |
| \_action\_ | insert\_action |
| \_action\_ | insert\_profile\_has\_action\_node |
| \_action\_ | change\_action\_state |
| \_action\_ | get\_action\_script\_by\_handler |
| \_action\_ | validate\_action\_access |
| \_action\_ | get\_action\_by\_id |
| \_action\_ | insert\_action |
| \_action\_ | get\_action\_by\_app\_id |
| \_action\_ | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| \_action\_ | get\_action |

#### List of diagrams containing the table \_action\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_action\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| root\_identifier | root\_identifier |
| app\_identifier | app\_identifier |
| type | type |
| handler | handler |
| action | action |
| access | access |
| name | name |
| title | title |
| show\_in\_menu | show\_in\_menu |
| icon | icon |
| icon\_color | icon\_color |
| confirm | confirm |
| run\_mode | run\_mode |
| template | template |
| context\_states | context\_states |
| controller\_name | controller\_name |
| properties | properties |
| description | description |
| state | state |
| last updater | last\_updater |
| last update date | last\_update\_date |

#### List of indexes of the table \_action\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Unique | Cluster | Primary | Foreign Key | Alternate Key | Table |
| action\_unique\_index | action\_unique\_index | X |  |  |  |  | \_action\_ |

#### List of keys of the table \_action\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_ambulance\_use\_

#### Card of table \_ambulance\_use\_

|  |  |
| --- | --- |
| Name | \_ambulance\_use\_ |
| Code | \_ambulance\_use\_ |

#### Server validation rule of the table \_ambulance\_use\_

%RULES%

#### Check constraint name of the table \_ambulance\_use\_

CKT\_\_AMBULANCE\_USE\_

#### List of outgoing references of the table \_ambulance\_use\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_ambulance\_use\_references\_form008\_ | \_fk\_ambulance\_use\_references\_form008\_ | \_form\_008\_ | form008\_identifier |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_tariff\_ | ambulance\_km\_code |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_tariff\_ | starting\_point\_code |  |  |

#### List of incoming traceability links of the table \_ambulance\_use\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_ambulance\_use\_ | get\_ambulance\_use\_by\_form008\_identifier |
| \_ambulance\_use\_ | process\_ambulance\_use |

#### List of diagrams containing the table \_ambulance\_use\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_ambulance\_use\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| form008\_identifier | form008\_identifier |
| autorization\_number | autorization\_number |
| city | city |
| km\_out | km\_out |
| km\_in | km\_in |
| km\_diff | km\_diff |
| starting\_point\_code | starting\_point\_code |
| starting\_point\_description | starting\_point\_description |
| starting\_point\_uvr | starting\_point\_uvr |
| starting\_point\_fcm | starting\_point\_fcm |
| starting\_point\_value | starting\_point\_value |
| ambulance\_km\_code | ambulance\_km\_code |
| ambulance\_km\_description | ambulance\_km\_description |
| ambulance\_km\_uvr | ambulance\_km\_uvr |
| ambulance\_km\_fcm | ambulance\_km\_fcm |
| ambulance\_km\_value | ambulance\_km\_value |
| ambulance\_km\_subtotal | ambulance\_km\_subtotal |
| total | total |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_ambulance\_use\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_app\_

#### Card of table \_app\_

|  |  |
| --- | --- |
| Name | \_app\_ |
| Code | \_app\_ |

#### Server validation rule of the table \_app\_

%RULES%

#### Options of the table \_app\_

tablespace app

#### Check constraint name of the table \_app\_

CKT\_\_APP\_

#### List of tablespaces of the table \_app\_

|  |  |
| --- | --- |
| Name | Code |
| app | app |

#### List of incoming references of the table \_app\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_action\_references\_app\_ | \_fk\_action\_references\_app\_ | \_action\_ | app\_identifier |  |  |
| \_fk\_catalog\_refereces\_app\_ | \_fk\_catalog\_refereces\_app\_ | \_catalog\_ | app\_identifier |  |  |
| \_fk\_enterprise\_references\_app\_ | \_fk\_enterprise\_references\_app\_ | \_enterprise\_ | app\_identifier |  |  |
| \_fk\_profile\_references\_app\_ | \_fk\_profile\_references\_app\_ | \_profile\_ | app identifier |  |  |

#### List of incoming traceability links of the table \_app\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_app\_ | get\_action\_by\_id |
| \_app\_ | get\_user\_by\_id |
| \_app\_ | validate\_action\_access |
| \_app\_ | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| \_app\_ | get\_action\_by\_app\_id |
| \_app\_ | insert\_app |
| \_app\_ | get\_profile\_by\_id |
| \_app\_ | get\_app\_by\_id |
| \_app\_ | login\_user |

#### List of diagrams containing the table \_app\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_app\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| code | code |
| name | name |
| title | title |
| description | description |
| state | state |
| last updater | last\_updater |
| last update date | last\_update\_date |
| expiration\_date | expiration\_date |

#### List of keys of the table \_app\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_canton\_

#### Card of table \_canton\_

|  |  |
| --- | --- |
| Name | \_canton\_ |
| Code | \_canton\_ |

#### Server validation rule of the table \_canton\_

%RULES%

#### Check constraint name of the table \_canton\_

CKT\_\_CANTON\_

#### List of incoming references of the table \_canton\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_parish\_references\_canton\_\_ | \_\_fk\_parish\_references\_canton\_\_ | \_parish\_ | canton\_identifer |  |  |

#### List of outgoing references of the table \_canton\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_canton\_references\_province\_\_ | \_\_fk\_canton\_references\_province\_\_ | \_province\_ | province\_identifier |  |  |

#### List of incoming traceability links of the table \_canton\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_canton\_ | get\_patient\_by\_id2 |
| \_canton\_ | get\_fur\_by\_form008\_identifier |

#### List of diagrams containing the table \_canton\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_canton\_

|  |  |
| --- | --- |
| Name | Code |
| identifer | identifer |
| province\_identifier | province\_identifier |
| name | name |

#### List of keys of the table \_canton\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_catalog\_

#### Card of table \_catalog\_

|  |  |
| --- | --- |
| Name | \_catalog\_ |
| Code | \_catalog\_ |

#### Server validation rule of the table \_catalog\_

%RULES%

#### Check constraint name of the table \_catalog\_

CKT\_\_CATALOG\_

#### List of procedures of the table \_catalog\_

|  |  |
| --- | --- |
| Name | Code |
| get\_catalog\_by\_id | get\_catalog\_by\_id |

#### List of incoming references of the table \_catalog\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_doctor\_ | speciality\_identifier |  |  |

#### List of outgoing references of the table \_catalog\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_catalog\_refereces\_app\_ | \_fk\_catalog\_refereces\_app\_ | \_app\_ | app\_identifier |  |  |

#### List of incoming traceability links of the table \_catalog\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_catalog\_ | process\_catalog |
| \_catalog\_ | get\_catalog |
| \_catalog\_ | change\_catalog\_state |
| \_catalog\_ | get\_ambulance\_use\_tariff |
| \_catalog\_ | get\_oxygen\_use\_tariff |
| \_catalog\_ | get\_oxygen\_use\_tariff |
| \_catalog\_ | get\_catalog\_type |
| \_catalog\_ | get\_tariff\_as\_medicine |
| \_catalog\_ | get\_catalog\_by\_id |

#### List of diagrams containing the table \_catalog\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_catalog\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| app\_identifier | app\_identifier |
| type | type |
| name | name |
| value | value |
| description | description |
| state | state |
| last updater | last\_updater |
| last update date | last\_update\_date |

#### List of indexes of the table \_catalog\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Unique | Cluster | Primary | Foreign Key | Alternate Key | Table |
| idx\_catalog\_unique\_index2 | idx\_catalog\_unique\_index2 | X |  |  |  |  | \_catalog\_ |

#### List of keys of the table \_catalog\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_department\_

#### Card of table \_department\_

|  |  |
| --- | --- |
| Name | \_department\_ |
| Code | \_department\_ |

#### Server validation rule of the table \_department\_

%RULES%

#### Check constraint name of the table \_department\_

CKT\_\_DEPARTMENT\_

#### List of incoming references of the table \_department\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_department\_references\_department\_coordination\_ | \_fk\_department\_references\_department\_coordination\_ | \_department\_ | coordination\_identifier |  |  |
| \_fk\_user\_references\_department\_ | \_fk\_user\_references\_department\_ | \_user\_ | department\_identifier |  |  |

#### List of outgoing references of the table \_department\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_department\_references\_department\_coordination\_ | \_fk\_department\_references\_department\_coordination\_ | \_department\_ | coordination\_identifier |  |  |
| \_fk\_department\_references\_enterprice\_ | \_fk\_department\_references\_enterprice\_ | \_enterprise\_ | enterprise\_identifier |  |  |

#### List of incoming traceability links of the table \_department\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_department\_ | get\_user |
| \_department\_ | validate\_action\_access |
| \_department\_ | process\_department |
| \_department\_ | get\_department\_by\_enterprise\_id |
| \_department\_ | get\_department |
| \_department\_ | get\_coordination |
| \_department\_ | get\_department\_by\_id |

#### List of diagrams containing the table \_department\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_department\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| enterprise\_identifier | enterprise\_identifier |
| coordination\_identifier | coordination\_identifier |
| name | name |
| code | code |
| description | description |
| state | state |
| last updater | last\_updater |
| last update date | last\_update\_date |

#### List of indexes of the table \_department\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Unique | Cluster | Primary | Foreign Key | Alternate Key | Table |
| idx\_department\_unique | idx\_department\_unique | X |  |  |  |  | \_department\_ |

#### List of keys of the table \_department\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_doctor\_

#### Card of table \_doctor\_

|  |  |
| --- | --- |
| Name | \_doctor\_ |
| Code | \_doctor\_ |

#### Server validation rule of the table \_doctor\_

%RULES%

#### Check constraint name of the table \_doctor\_

CKT\_\_DOCTOR\_

#### List of incoming references of the table \_doctor\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_form008\_has\_doctor\_ | doctor\_identifier |  |  |
| \_\_fur\_references\_medic\_authorizer\_\_ | \_\_fur\_references\_medic\_authorizer\_\_ | \_fur\_ | insured\_dead\_authorizer\_doctor |  |  |
| \_\_fur\_references\_medic\_in\_charge\_\_ | \_\_fur\_references\_medic\_in\_charge\_\_ | \_fur\_ | insured\_dead\_in\_charge\_doctor |  |  |

#### List of outgoing references of the table \_doctor\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_catalog\_ | speciality\_identifier |  |  |

#### List of referencing views of the table \_doctor\_

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

#### List of incoming traceability links of the table \_doctor\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_doctor\_ | get\_doctor\_by\_form008\_identifier |
| \_doctor\_ | get\_doctor\_by\_id |
| \_doctor\_ | get\_all\_doctor |

#### List of diagrams containing the table \_doctor\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_doctor\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| speciality\_identifier | speciality\_identifier |
| identity\_number | identity\_number |
| first\_name | first\_name |
| last\_name | last\_name |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of indexes of the table \_doctor\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Unique | Cluster | Primary | Foreign Key | Alternate Key | Table |
| doctor\_uk | doctor\_uk | X |  |  |  |  | \_doctor\_ |

#### List of keys of the table \_doctor\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_enterprise\_

#### Card of table \_enterprise\_

|  |  |
| --- | --- |
| Name | \_enterprise\_ |
| Code | \_enterprise\_ |

#### Server validation rule of the table \_enterprise\_

%RULES%

#### Check constraint name of the table \_enterprise\_

CKT\_\_ENTERPRISE\_

#### List of procedures of the table \_enterprise\_

|  |  |
| --- | --- |
| Name | Code |
| process\_enterprise | process\_enterprise |

#### List of incoming references of the table \_enterprise\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_department\_references\_enterprice\_ | \_fk\_department\_references\_enterprice\_ | \_department\_ | enterprise\_identifier |  |  |

#### List of outgoing references of the table \_enterprise\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_enterprise\_references\_app\_ | \_fk\_enterprise\_references\_app\_ | \_app\_ | app\_identifier |  |  |

#### List of incoming traceability links of the table \_enterprise\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_enterprise\_ | get\_enterprise |
| \_enterprise\_ | process\_enterprise |
| \_enterprise\_ | process\_enterprise |
| \_enterprise\_ | get\_enterprise |

#### List of diagrams containing the table \_enterprise\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_enterprise\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| app\_identifier | app\_identifier |
| identity\_number | identity\_number |
| name | name |
| business\_name | business\_name |
| law\_representative | law\_representative |
| address | address |
| phone | phone |
| email | email |
| description | description |
| state | state |
| last updater | last\_updater |
| last update date | last\_update\_date |

#### List of indexes of the table \_enterprise\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Unique | Cluster | Primary | Foreign Key | Alternate Key | Table |
| identity\_number | identity\_number | X |  |  |  |  | \_enterprise\_ |

#### List of keys of the table \_enterprise\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_expense\_form\_

#### Card of table \_expense\_form\_

|  |  |
| --- | --- |
| Name | \_expense\_form\_ |
| Code | \_expense\_form\_ |

#### Server validation rule of the table \_expense\_form\_

%RULES%

#### Check constraint name of the table \_expense\_form\_

CKT\_\_EXPENSE\_FORM\_

#### List of incoming references of the table \_expense\_form\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_expense\_form\_medicine\_detail\_ | expense\_form\_identifier |  |  |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_expense\_form\_tariff\_detail\_ | expense\_form\_identifier |  |  |

#### List of outgoing references of the table \_expense\_form\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_expense\_form\_references\_form008\_\_ | \_\_fk\_expense\_form\_references\_form008\_\_ | \_form\_008\_ | form008\_identifier |  |  |

#### List of referencing views of the table \_expense\_form\_

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

#### List of incoming traceability links of the table \_expense\_form\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_expense\_form\_ | get\_expense\_form\_by\_form008\_identifier |
| \_expense\_form\_ | process\_expense\_form |

#### List of diagrams containing the table \_expense\_form\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_expense\_form\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| form008\_identifier | form008\_identifier |
| diagnostic | diagnostic |
| total | total |
| sppat | sppat |
| balance | balance |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_expense\_form\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_expense\_form\_medicine\_detail\_

#### Card of table \_expense\_form\_medicine\_detail\_

|  |  |
| --- | --- |
| Name | \_expense\_form\_medicine\_detail\_ |
| Code | \_expense\_form\_medicine\_detail\_ |

#### Server validation rule of the table \_expense\_form\_medicine\_detail\_

%RULES%

#### Check constraint name of the table \_expense\_form\_medicine\_detail\_

CKT\_\_EXPENSE\_FORM\_MEDICINE\_DET

#### List of outgoing references of the table \_expense\_form\_medicine\_detail\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_expense\_form\_ | expense\_form\_identifier |  |  |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_medicine\_ | item\_identifier |  |  |

#### List of incoming traceability links of the table \_expense\_form\_medicine\_detail\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_expense\_form\_medicine\_detail\_ | get\_all\_expense\_form\_detail |
| \_expense\_form\_medicine\_detail\_ | process\_expense\_form |

#### List of diagrams containing the table \_expense\_form\_medicine\_detail\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_expense\_form\_medicine\_detail\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| expense\_form\_identifier | expense\_form\_identifier |
| item\_identifier | item\_identifier |
| item\_code | item\_code |
| item\_description | item\_description |
| item\_type | item\_type |
| quantity | quantity |
| unit\_value | unit\_value |
| total\_value | total\_value |

#### List of keys of the table \_expense\_form\_medicine\_detail\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_expense\_form\_tariff\_detail\_

#### Card of table \_expense\_form\_tariff\_detail\_

|  |  |
| --- | --- |
| Name | \_expense\_form\_tariff\_detail\_ |
| Code | \_expense\_form\_tariff\_detail\_ |

#### Server validation rule of the table \_expense\_form\_tariff\_detail\_

%RULES%

#### Check constraint name of the table \_expense\_form\_tariff\_detail\_

CKT\_\_EXPENSE\_FORM\_TARIFF\_DETAI

#### List of outgoing references of the table \_expense\_form\_tariff\_detail\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_expense\_form\_ | expense\_form\_identifier |  |  |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_tariff\_ | item\_identifier |  |  |

#### List of incoming traceability links of the table \_expense\_form\_tariff\_detail\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_expense\_form\_tariff\_detail\_ | get\_all\_expense\_form\_detail |
| \_expense\_form\_tariff\_detail\_ | process\_expense\_form |

#### List of diagrams containing the table \_expense\_form\_tariff\_detail\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_expense\_form\_tariff\_detail\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| expense\_form\_identifier | expense\_form\_identifier |
| item\_identifier | item\_identifier |
| item\_code | item\_code |
| item\_description | item\_description |
| item\_type | item\_type |
| quantity | quantity |
| unit\_value | unit\_value |
| total\_value | total\_value |

#### List of keys of the table \_expense\_form\_tariff\_detail\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_form008\_has\_doctor\_

#### Card of table \_form008\_has\_doctor\_

|  |  |
| --- | --- |
| Name | \_form008\_has\_doctor\_ |
| Code | \_form008\_has\_doctor\_ |

#### Server validation rule of the table \_form008\_has\_doctor\_

%RULES%

#### Check constraint name of the table \_form008\_has\_doctor\_

CKT\_\_FORM008\_HAS\_DOCTOR\_

#### List of outgoing references of the table \_form008\_has\_doctor\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_doctor\_ | doctor\_identifier |  |  |
| \_\_form008\_has\_doctor\_references\_form008\_\_ | \_\_form008\_has\_doctor\_references\_form008\_\_ | \_form\_008\_ | form008\_identifier |  |  |

#### List of referencing views of the table \_form008\_has\_doctor\_

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

#### List of incoming traceability links of the table \_form008\_has\_doctor\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_form008\_has\_doctor\_ | process\_form008\_has\_doctor |
| \_form008\_has\_doctor\_ | get\_doctor\_by\_form008\_identifier |

#### List of diagrams containing the table \_form008\_has\_doctor\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_form008\_has\_doctor\_

|  |  |
| --- | --- |
| Name | Code |
| form008\_identifier | form008\_identifier |
| doctor\_identifier | doctor\_identifier |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_form008\_has\_doctor\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_form\_008\_

#### Card of table \_form\_008\_

|  |  |
| --- | --- |
| Name | \_form\_008\_ |
| Code | \_form\_008\_ |

#### Server validation rule of the table \_form\_008\_

%RULES%

#### Check constraint name of the table \_form\_008\_

CKT\_\_FORM\_008\_

#### List of incoming references of the table \_form\_008\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_expense\_form\_references\_form008\_\_ | \_\_fk\_expense\_form\_references\_form008\_\_ | \_expense\_form\_ | form008\_identifier |  |  |
| \_\_form008\_has\_doctor\_references\_form008\_\_ | \_\_form008\_has\_doctor\_references\_form008\_\_ | \_form008\_has\_doctor\_ | form008\_identifier |  |  |
| \_\_fur\_references\_form008\_\_ | \_\_fur\_references\_form008\_\_ | \_fur\_ | form008\_identifier |  |  |
| \_\_objection\_references\_form008\_\_ | \_\_objection\_references\_form008\_\_ | \_objection\_ | form008\_identifier |  |  |
| \_fk\_ambulance\_use\_references\_form008\_ | \_fk\_ambulance\_use\_references\_form008\_ | \_ambulance\_use\_ | form008\_identifier |  |  |
| \_fk\_oxigen\_use\_references\_form008\_ | \_fk\_oxigen\_use\_references\_form008\_ | \_oxygen\_use\_ | form008\_identifier |  |  |

#### List of outgoing references of the table \_form\_008\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_form008\_references\_patient\_\_ | \_\_fk\_form008\_references\_patient\_\_ | \_patient\_ | patient\_identifier |  |  |

#### List of referencing views of the table \_form\_008\_

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

#### List of incoming traceability links of the table \_form\_008\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_form\_008\_ | process\_form008 |
| \_form\_008\_ | get\_form008\_by\_id |
| \_form\_008\_ | get\_form008 |

#### List of diagrams containing the table \_form\_008\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_form\_008\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| code | code |
| patient\_identifier | patient\_identifier |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_form\_008\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_fur\_

#### Card of table \_fur\_

|  |  |
| --- | --- |
| Name | \_fur\_ |
| Code | \_fur\_ |

#### Server validation rule of the table \_fur\_

%RULES%

#### Check constraint name of the table \_fur\_

CKT\_\_FUR\_

#### List of outgoing references of the table \_fur\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fur\_references\_form008\_\_ | \_\_fur\_references\_form008\_\_ | \_form\_008\_ | form008\_identifier |  |  |
| \_\_fur\_references\_medic\_authorizer\_\_ | \_\_fur\_references\_medic\_authorizer\_\_ | \_doctor\_ | insured\_dead\_authorizer\_doctor |  |  |
| \_\_fur\_references\_medic\_in\_charge\_\_ | \_\_fur\_references\_medic\_in\_charge\_\_ | \_doctor\_ | insured\_dead\_in\_charge\_doctor |  |  |

#### List of referencing views of the table \_fur\_

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

#### List of incoming traceability links of the table \_fur\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_fur\_ | process\_fur |
| \_fur\_ | get\_fur\_by\_form008\_identifier |

#### List of diagrams containing the table \_fur\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_fur\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| form008\_identifier | form008\_identifier |
| patient\_condition\_walker | patient\_condition\_walker |
| patient\_condition\_occupant | patient\_condition\_occupant |
| accident\_address | accident\_address |
| accident\_province | accident\_province |
| accident\_canton | accident\_canton |
| accident\_parish | accident\_parish |
| accident\_neighborhood | accident\_neighborhood |
| accident\_date | accident\_date |
| accident\_hour | accident\_hour |
| accident\_minute | accident\_minute |
| accident\_report | accident\_report |
| vehicle\_insured | vehicle\_insured |
| vehicle\_not\_insured | vehicle\_not\_insured |
| vehicle\_not\_identified | vehicle\_not\_identified |
| vehicle\_coverage\_code | vehicle\_coverage\_code |
| vehicle\_license\_plate | vehicle\_license\_plate |
| vehicle\_soat\_number | vehicle\_soat\_number |
| vehicle\_insurance\_carrier | vehicle\_insurance\_carrier |
| vehicle\_police\_start\_date | vehicle\_police\_start\_date |
| vehicle\_police\_end\_date | vehicle\_police\_end\_date |
| health\_center\_admision\_date | health\_center\_admision\_date |
| health\_center\_admision\_hour | health\_center\_admision\_hour |
| health\_center\_admision\_minute | health\_center\_admision\_minute |
| health\_center\_history\_number | health\_center\_history\_number |
| health\_center\_egress\_date | health\_center\_egress\_date |
| health\_center\_ambulatory\_care | health\_center\_ambulatory\_care |
| health\_center\_hosp\_care | health\_center\_hosp\_care |
| health\_center\_hosp\_care\_days | health\_center\_hosp\_care\_days |
| health\_center\_admision\_report | health\_center\_admision\_report |
| health\_center\_egress\_report | health\_center\_egress\_report |
| health\_center\_referred\_fr | health\_center\_referred\_fr |
| health\_center\_referred\_fr\_city | health\_center\_referred\_fr\_city |
| health\_center\_referred\_fr\_date | health\_center\_referred\_fr\_date |
| health\_center\_referred\_to | health\_center\_referred\_to |
| health\_center\_referred\_to\_city | health\_center\_referred\_to\_city |
| health\_center\_referred\_to\_date | health\_center\_referred\_to\_date |
| insured\_dead | insured\_dead |
| insured\_dead\_basic\_reazon | insured\_dead\_basic\_reazon |
| insured\_dead\_direct\_reazon | insured\_dead\_direct\_reazon |
| insured\_dead\_date | insured\_dead\_date |
| insured\_dead\_hour | insured\_dead\_hour |
| insured\_dead\_minute | insured\_dead\_minute |
| insured\_dead\_in\_charge\_doctor | insured\_dead\_in\_charge\_doctor |
| insured\_dead\_authorizer\_doctor | insured\_dead\_authorizer\_doctor |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_fur\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_medicine\_

#### Card of table \_medicine\_

|  |  |
| --- | --- |
| Name | \_medicine\_ |
| Code | \_medicine\_ |

#### Server validation rule of the table \_medicine\_

%RULES%

#### Check constraint name of the table \_medicine\_

CKT\_\_MEDICINE\_

#### List of incoming references of the table \_medicine\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_expense\_form\_medicine\_detail\_ | item\_identifier |  |  |

#### List of incoming traceability links of the table \_medicine\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_medicine\_ | get\_all\_medicine |
| \_medicine\_ | process\_medicine |

#### List of diagrams containing the table \_medicine\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_medicine\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| code | code |
| name | name |
| type | type |
| value | value |
| stock | stock |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_medicine\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_objection\_

#### Card of table \_objection\_

|  |  |
| --- | --- |
| Name | \_objection\_ |
| Code | \_objection\_ |

#### Server validation rule of the table \_objection\_

%RULES%

#### Check constraint name of the table \_objection\_

CKT\_\_OBJECTION\_

#### List of outgoing references of the table \_objection\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_objection\_references\_form008\_\_ | \_\_objection\_references\_form008\_\_ | \_form\_008\_ | form008\_identifier |  |  |

#### List of referencing views of the table \_objection\_

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

#### List of incoming traceability links of the table \_objection\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_objection\_ | get\_objection\_by\_form008\_identifier |
| \_objection\_ | process\_objection |

#### List of diagrams containing the table \_objection\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_objection\_

|  |  |
| --- | --- |
| Name | Code |
| form008\_identifier | form008\_identifier |
| number | number |
| value | value |
| invoice | invoice |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_objection\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_oxygen\_use\_

#### Card of table \_oxygen\_use\_

|  |  |
| --- | --- |
| Name | \_oxygen\_use\_ |
| Code | \_oxygen\_use\_ |

#### Server validation rule of the table \_oxygen\_use\_

%RULES%

#### Check constraint name of the table \_oxygen\_use\_

CKT\_\_OXYGEN\_USE\_

#### List of outgoing references of the table \_oxygen\_use\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_oxigen\_use\_references\_form008\_ | \_fk\_oxigen\_use\_references\_form008\_ | \_form\_008\_ | form008\_identifier |  |  |

#### List of incoming traceability links of the table \_oxygen\_use\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_oxygen\_use\_ | get\_oxygen\_use\_by\_form008\_identifier |
| \_oxygen\_use\_ | process\_oxygen\_use |

#### List of diagrams containing the table \_oxygen\_use\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_oxygen\_use\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| form008\_identifier | form008\_identifier |
| start\_date | start\_date |
| start\_hour | start\_hour |
| start\_minute | start\_minute |
| end\_date | end\_date |
| end\_hour | end\_hour |
| end\_minute | end\_minute |
| diff\_hour | diff\_hour |
| diff\_minute | diff\_minute |
| total\_minute | total\_minute |
| lm | lm |
| oxygen\_code | oxygen\_code |
| oxygen\_description | oxygen\_description |
| uvr | uvr |
| fcm | fcm |
| value | value |
| total | total |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_oxygen\_use\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_parish\_

#### Card of table \_parish\_

|  |  |
| --- | --- |
| Name | \_parish\_ |
| Code | \_parish\_ |

#### Server validation rule of the table \_parish\_

%RULES%

#### Check constraint name of the table \_parish\_

CKT\_\_PARISH\_

#### List of outgoing references of the table \_parish\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_parish\_references\_canton\_\_ | \_\_fk\_parish\_references\_canton\_\_ | \_canton\_ | canton\_identifer |  |  |

#### List of incoming traceability links of the table \_parish\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_parish\_ | get\_patient\_by\_id2 |
| \_parish\_ | get\_fur\_by\_form008\_identifier |

#### List of diagrams containing the table \_parish\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_parish\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| canton\_identifer | canton\_identifer |
| name | name |

#### List of keys of the table \_parish\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_patient\_

#### Card of table \_patient\_

|  |  |
| --- | --- |
| Name | \_patient\_ |
| Code | \_patient\_ |

#### Server validation rule of the table \_patient\_

%RULES%

#### Check constraint name of the table \_patient\_

CKT\_\_PATIENT\_

#### List of procedures of the table \_patient\_

|  |  |
| --- | --- |
| Name | Code |
| get\_patient | get\_patient |
| process\_patient | process\_patient |

#### List of incoming references of the table \_patient\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_form008\_references\_patient\_\_ | \_\_fk\_form008\_references\_patient\_\_ | \_form\_008\_ | patient\_identifier |  |  |

#### List of referencing views of the table \_patient\_

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

#### List of incoming traceability links of the table \_patient\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_patient\_ | get\_form008 |
| \_patient\_ | get\_active\_patient |
| \_patient\_ | get\_patient\_by\_id2 |
| \_patient\_ | get\_patient |
| \_patient\_ | get\_patient |
| \_patient\_ | get\_patient\_by\_id |
| \_patient\_ | process\_patient |

#### List of diagrams containing the table \_patient\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_patient\_

|  |  |
| --- | --- |
| Name | Code |
| identificacion | identificacion |
| primer nombre | primer\_nombre |
| segundo nombre | segundo\_nombre |
| apellido materno | apellido\_materno |
| apellido paterno | apellido\_paterno |
| direccion | direccion |
| barrio | barrio |
| parroquia | parroquia |
| canton | canton |
| provincia | provincia |
| zona | zona |
| telefono | telefono |
| fecha\_nacimiento | fecha\_nacimiento |
| lugar nacimiento | lugar\_nacimiento |
| nacionalidad | nacionalidad |
| grupo\_cultural | grupo\_cultural |
| edad | edad |
| genero | genero |
| estado civil | estado\_civil |
| instruccion | instruccion |
| fecha admision | fecha\_admision |
| ocupacion | ocupacion |
| empresa\_trabajo | empresa\_trabajo |
| tipo\_seguro\_salud | tipo\_seguro\_salud |
| referido\_de | referido\_de |
| contacto emergencia | contacto\_emergencia |
| parentezco contacto emergencia | parentezco\_contacto\_emergencia |
| direccion contacto emergencia | direccion\_contacto\_emergencia |
| telefono contacto emergencia | telefono\_contacto\_emergencia |
| state | state |
| description | description |
| last update date | last\_update\_date |
| last updater | last\_updater |

#### List of keys of the table \_patient\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_profile\_

#### Card of table \_profile\_

|  |  |
| --- | --- |
| Name | \_profile\_ |
| Code | \_profile\_ |

#### Server validation rule of the table \_profile\_

%RULES%

#### Check constraint name of the table \_profile\_

CKT\_\_PROFILE\_

#### List of incoming references of the table \_profile\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_profile\_has\_action\_references\_profile\_ | \_fk\_profile\_has\_action\_references\_profile\_ | \_profile\_has\_action\_ | profile identifier |  |  |
| \_fk\_user\_references\_profile\_ | \_fk\_user\_references\_profile\_ | \_user\_ | profile identifier |  |  |

#### List of outgoing references of the table \_profile\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_profile\_references\_app\_ | \_fk\_profile\_references\_app\_ | \_app\_ | app identifier |  |  |

#### List of incoming traceability links of the table \_profile\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_profile\_ | insert\_profile |
| \_profile\_ | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| \_profile\_ | validate\_action\_access |
| \_profile\_ | update\_profile |
| \_profile\_ | change\_profile\_state |
| \_profile\_ | get\_user |
| \_profile\_ | get\_user\_by\_id |
| \_profile\_ | login\_user |
| \_profile\_ | insert\_profile |
| \_profile\_ | get\_profile\_by\_app\_id |
| \_profile\_ | get\_profile\_by\_id |
| \_profile\_ | get\_profile |

#### List of diagrams containing the table \_profile\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_profile\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| app identifier | app\_identifier |
| name | name |
| description | description |
| state | state |
| last updater | last\_updater |
| last update date | last\_update\_date |

#### List of keys of the table \_profile\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_profile\_has\_action\_

#### Card of table \_profile\_has\_action\_

|  |  |
| --- | --- |
| Name | \_profile\_has\_action\_ |
| Code | \_profile\_has\_action\_ |

#### Server validation rule of the table \_profile\_has\_action\_

%RULES%

#### Check constraint name of the table \_profile\_has\_action\_

CKT\_\_PROFILE\_HAS\_ACTION\_

#### List of outgoing references of the table \_profile\_has\_action\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_profile\_has\_action\_references\_action\_ | \_fk\_profile\_has\_action\_references\_action\_ | \_action\_ | action identifier |  |  |
| \_fk\_profile\_has\_action\_references\_profile\_ | \_fk\_profile\_has\_action\_references\_profile\_ | \_profile\_ | profile identifier |  |  |

#### List of incoming traceability links of the table \_profile\_has\_action\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_profile\_has\_action\_ | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| \_profile\_has\_action\_ | get\_all\_menu\_action\_by\_profile\_id\_root\_id |
| \_profile\_has\_action\_ | insert\_profile\_has\_action\_node |
| \_profile\_has\_action\_ | validate\_action\_access |
| \_profile\_has\_action\_ | insert\_profile\_has\_action |

#### List of diagrams containing the table \_profile\_has\_action\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_profile\_has\_action\_

|  |  |
| --- | --- |
| Name | Code |
| profile identifier | profile\_identifier |
| action identifier | action\_identifier |

#### List of keys of the table \_profile\_has\_action\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_province\_

#### Card of table \_province\_

|  |  |
| --- | --- |
| Name | \_province\_ |
| Code | \_province\_ |

#### Server validation rule of the table \_province\_

%RULES%

#### Check constraint name of the table \_province\_

CKT\_\_PROVINCE\_

#### List of incoming references of the table \_province\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_canton\_references\_province\_\_ | \_\_fk\_canton\_references\_province\_\_ | \_canton\_ | province\_identifier |  |  |

#### List of incoming traceability links of the table \_province\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_province\_ | get\_patient\_by\_id2 |
| \_province\_ | get\_fur\_by\_form008\_identifier |

#### List of diagrams containing the table \_province\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_province\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| name | name |

#### List of keys of the table \_province\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_tariff\_

#### Card of table \_tariff\_

|  |  |
| --- | --- |
| Name | \_tariff\_ |
| Code | \_tariff\_ |

#### Server validation rule of the table \_tariff\_

%RULES%

#### Check constraint name of the table \_tariff\_

CKT\_\_TARIFF\_

#### List of incoming references of the table \_tariff\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Child Table | Foreign Key Columns | Parent Role | Child Role |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_expense\_form\_tariff\_detail\_ | item\_identifier |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_ambulance\_use\_ | ambulance\_km\_code |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_ambulance\_use\_ | starting\_point\_code |  |  |

#### List of incoming traceability links of the table \_tariff\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_tariff\_ | get\_oxygen\_use\_tariff |
| \_tariff\_ | get\_tariff\_as\_medicine |
| \_tariff\_ | get\_ambulance\_use\_tariff |

#### List of diagrams containing the table \_tariff\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the table \_tariff\_

|  |  |
| --- | --- |
| Name | Code |
| code | code |
| type | type |
| description | description |
| uvr1 | uvr1 |
| fcm1 | fcm1 |
| valor1 | valor1 |
| uvr2 | uvr2 |
| fcm2 | fcm2 |
| valor2 | valor2 |
| uvr3 | uvr3 |
| fcm3 | fcm3 |
| valor3 | valor3 |

#### List of keys of the table \_tariff\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

### Table \_user\_

#### Card of table \_user\_

|  |  |
| --- | --- |
| Name | \_user\_ |
| Code | \_user\_ |

#### Server validation rule of the table \_user\_

%RULES%

#### Check constraint name of the table \_user\_

CKT\_\_USER\_

#### List of outgoing references of the table \_user\_

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Code | Parent Table | Foreign Key Columns | Parent Role | Child Role |
| \_fk\_user\_references\_department\_ | \_fk\_user\_references\_department\_ | \_department\_ | department\_identifier |  |  |
| \_fk\_user\_references\_profile\_ | \_fk\_user\_references\_profile\_ | \_profile\_ | profile identifier |  |  |

#### List of incoming traceability links of the table \_user\_

|  |  |
| --- | --- |
| Linked Object | Source Object |
| \_user\_ | get\_user |
| \_user\_ | change\_user\_password |
| \_user\_ | process\_user |
| \_user\_ | change\_user\_state |
| \_user\_ | login\_user |
| \_user\_ | validate\_action\_access |
| \_user\_ | get\_user\_by\_id |
| \_user\_ | process\_user |

#### List of diagrams containing the table \_user\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of columns of the table \_user\_

|  |  |
| --- | --- |
| Name | Code |
| identifier | identifier |
| login | login |
| profile identifier | profile\_identifier |
| department\_identifier | department\_identifier |
| name | name |
| last\_name | last\_name |
| email | email |
| password | password |
| description | description |
| state | state |
| last updater | last\_updater |
| last update date | last\_update\_date |

#### List of indexes of the table \_user\_

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Code | Unique | Cluster | Primary | Foreign Key | Alternate Key | Table |
| user\_unique\_index | user\_unique\_index | X |  |  |  |  | \_user\_ |

#### List of keys of the table \_user\_

|  |  |  |
| --- | --- | --- |
| Name | Code | Primary |
| Key\_1 | key\_1 | X |

## List of references

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Name* | *Code* | *Parent Table* | *Child Table* | *Foreign Key Columns* | *Parent Role* | *Child Role* |
| \_\_fk\_canton\_references\_province\_\_ | \_\_fk\_canton\_references\_province\_\_ | \_province\_ | \_canton\_ | province\_identifier |  |  |
| \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_\_fk\_doctor\_references\_catalog\_specialty\_\_ | \_catalog\_ | \_doctor\_ | speciality\_identifier |  |  |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ | \_expense\_form\_ | \_expense\_form\_medicine\_detail\_ | expense\_form\_identifier |  |  |
| \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ | \_medicine\_ | \_expense\_form\_medicine\_detail\_ | item\_identifier |  |  |
| \_\_fk\_expense\_form\_references\_form008\_\_ | \_\_fk\_expense\_form\_references\_form008\_\_ | \_form\_008\_ | \_expense\_form\_ | form008\_identifier |  |  |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ | \_expense\_form\_ | \_expense\_form\_tariff\_detail\_ | expense\_form\_identifier |  |  |
| \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ | \_tariff\_ | \_expense\_form\_tariff\_detail\_ | item\_identifier |  |  |
| \_\_fk\_form008\_references\_patient\_\_ | \_\_fk\_form008\_references\_patient\_\_ | \_patient\_ | \_form\_008\_ | patient\_identifier |  |  |
| \_\_fk\_parish\_references\_canton\_\_ | \_\_fk\_parish\_references\_canton\_\_ | \_canton\_ | \_parish\_ | canton\_identifer |  |  |
| \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_\_form008\_has\_doctor\_references\_doctor\_\_ | \_doctor\_ | \_form008\_has\_doctor\_ | doctor\_identifier |  |  |
| \_\_form008\_has\_doctor\_references\_form008\_\_ | \_\_form008\_has\_doctor\_references\_form008\_\_ | \_form\_008\_ | \_form008\_has\_doctor\_ | form008\_identifier |  |  |
| \_\_fur\_references\_form008\_\_ | \_\_fur\_references\_form008\_\_ | \_form\_008\_ | \_fur\_ | form008\_identifier |  |  |
| \_\_fur\_references\_medic\_authorizer\_\_ | \_\_fur\_references\_medic\_authorizer\_\_ | \_doctor\_ | \_fur\_ | insured\_dead\_authorizer\_doctor |  |  |
| \_\_fur\_references\_medic\_in\_charge\_\_ | \_\_fur\_references\_medic\_in\_charge\_\_ | \_doctor\_ | \_fur\_ | insured\_dead\_in\_charge\_doctor |  |  |
| \_\_objection\_references\_form008\_\_ | \_\_objection\_references\_form008\_\_ | \_form\_008\_ | \_objection\_ | form008\_identifier |  |  |
| \_fk\_action\_reference\_action\_ | \_fk\_action\_reference\_action\_ | \_action\_ | \_action\_ | root\_identifier |  |  |
| \_fk\_action\_references\_app\_ | \_fk\_action\_references\_app\_ | \_app\_ | \_action\_ | app\_identifier |  |  |
| \_fk\_ambulance\_use\_references\_form008\_ | \_fk\_ambulance\_use\_references\_form008\_ | \_form\_008\_ | \_ambulance\_use\_ | form008\_identifier |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ | \_tariff\_ | \_ambulance\_use\_ | ambulance\_km\_code |  |  |
| \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ | \_tariff\_ | \_ambulance\_use\_ | starting\_point\_code |  |  |
| \_fk\_catalog\_refereces\_app\_ | \_fk\_catalog\_refereces\_app\_ | \_app\_ | \_catalog\_ | app\_identifier |  |  |
| \_fk\_department\_references\_department\_coordination\_ | \_fk\_department\_references\_department\_coordination\_ | \_department\_ | \_department\_ | coordination\_identifier |  |  |
| \_fk\_department\_references\_enterprice\_ | \_fk\_department\_references\_enterprice\_ | \_enterprise\_ | \_department\_ | enterprise\_identifier |  |  |
| \_fk\_enterprise\_references\_app\_ | \_fk\_enterprise\_references\_app\_ | \_app\_ | \_enterprise\_ | app\_identifier |  |  |
| \_fk\_oxigen\_use\_references\_form008\_ | \_fk\_oxigen\_use\_references\_form008\_ | \_form\_008\_ | \_oxygen\_use\_ | form008\_identifier |  |  |
| \_fk\_profile\_has\_action\_references\_action\_ | \_fk\_profile\_has\_action\_references\_action\_ | \_action\_ | \_profile\_has\_action\_ | action identifier |  |  |
| \_fk\_profile\_has\_action\_references\_profile\_ | \_fk\_profile\_has\_action\_references\_profile\_ | \_profile\_ | \_profile\_has\_action\_ | profile identifier |  |  |
| \_fk\_profile\_references\_app\_ | \_fk\_profile\_references\_app\_ | \_app\_ | \_profile\_ | app identifier |  |  |
| \_fk\_user\_references\_department\_ | \_fk\_user\_references\_department\_ | \_department\_ | \_user\_ | department\_identifier |  |  |
| \_fk\_user\_references\_profile\_ | \_fk\_user\_references\_profile\_ | \_profile\_ | \_user\_ | profile identifier |  |  |

### Reference \_\_fk\_canton\_references\_province\_\_

#### Card of reference \_\_fk\_canton\_references\_province\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_canton\_references\_province\_\_ |
| Code | \_\_fk\_canton\_references\_province\_\_ |
| Child Table | \_canton\_ |
| Parent Table | \_province\_ |
| Foreign Key Columns | province\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_canton\_references\_province\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_canton\_references\_province\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = province\_identifier | identifier | province\_identifier |

### Reference \_\_fk\_doctor\_references\_catalog\_specialty\_\_

#### Card of reference \_\_fk\_doctor\_references\_catalog\_specialty\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_doctor\_references\_catalog\_specialty\_\_ |
| Code | \_\_fk\_doctor\_references\_catalog\_specialty\_\_ |
| Child Table | \_doctor\_ |
| Parent Table | \_catalog\_ |
| Foreign Key Columns | speciality\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_doctor\_references\_catalog\_specialty\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_doctor\_references\_catalog\_specialty\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = speciality\_identifier | identifier | speciality\_identifier |

### Reference \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_

#### Card of reference \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ |
| Code | \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_ |
| Child Table | \_expense\_form\_medicine\_detail\_ |
| Parent Table | \_expense\_form\_ |
| Foreign Key Columns | expense\_form\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_expense\_form\_medicine\_detail\_references\_expense\_form\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = expense\_form\_identifier | identifier | expense\_form\_identifier |

### Reference \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_

#### Card of reference \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ |
| Code | \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_ |
| Child Table | \_expense\_form\_medicine\_detail\_ |
| Parent Table | \_medicine\_ |
| Foreign Key Columns | item\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_expense\_form\_medicine\_detail\_references\_medicine\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = item\_identifier | identifier | item\_identifier |

### Reference \_\_fk\_expense\_form\_references\_form008\_\_

#### Card of reference \_\_fk\_expense\_form\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_expense\_form\_references\_form008\_\_ |
| Code | \_\_fk\_expense\_form\_references\_form008\_\_ |
| Child Table | \_expense\_form\_ |
| Parent Table | \_form\_008\_ |
| Foreign Key Columns | form008\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_expense\_form\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_expense\_form\_references\_form008\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = form008\_identifier | identifier | form008\_identifier |

### Reference \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_

#### Card of reference \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ |
| Code | \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_ |
| Child Table | \_expense\_form\_tariff\_detail\_ |
| Parent Table | \_expense\_form\_ |
| Foreign Key Columns | expense\_form\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_expense\_form\_tariff\_detail\_references\_expense\_form\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = expense\_form\_identifier | identifier | expense\_form\_identifier |

### Reference \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_

#### Card of reference \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ |
| Code | \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_ |
| Child Table | \_expense\_form\_tariff\_detail\_ |
| Parent Table | \_tariff\_ |
| Foreign Key Columns | item\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_expense\_form\_tariff\_detail\_references\_tariff\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| code = item\_identifier | code | item\_identifier |

### Reference \_\_fk\_form008\_references\_patient\_\_

#### Card of reference \_\_fk\_form008\_references\_patient\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_form008\_references\_patient\_\_ |
| Code | \_\_fk\_form008\_references\_patient\_\_ |
| Child Table | \_form\_008\_ |
| Parent Table | \_patient\_ |
| Foreign Key Columns | patient\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_form008\_references\_patient\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_form008\_references\_patient\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identificacion = patient\_identifier | identificacion | patient\_identifier |

### Reference \_\_fk\_parish\_references\_canton\_\_

#### Card of reference \_\_fk\_parish\_references\_canton\_\_

|  |  |
| --- | --- |
| Name | \_\_fk\_parish\_references\_canton\_\_ |
| Code | \_\_fk\_parish\_references\_canton\_\_ |
| Child Table | \_parish\_ |
| Parent Table | \_canton\_ |
| Foreign Key Columns | canton\_identifer |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fk\_parish\_references\_canton\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fk\_parish\_references\_canton\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifer = canton\_identifer | identifer | canton\_identifer |

### Reference \_\_form008\_has\_doctor\_references\_doctor\_\_

#### Card of reference \_\_form008\_has\_doctor\_references\_doctor\_\_

|  |  |
| --- | --- |
| Name | \_\_form008\_has\_doctor\_references\_doctor\_\_ |
| Code | \_\_form008\_has\_doctor\_references\_doctor\_\_ |
| Child Table | \_form008\_has\_doctor\_ |
| Parent Table | \_doctor\_ |
| Foreign Key Columns | doctor\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_form008\_has\_doctor\_references\_doctor\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_form008\_has\_doctor\_references\_doctor\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = doctor\_identifier | identifier | doctor\_identifier |

### Reference \_\_form008\_has\_doctor\_references\_form008\_\_

#### Card of reference \_\_form008\_has\_doctor\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | \_\_form008\_has\_doctor\_references\_form008\_\_ |
| Code | \_\_form008\_has\_doctor\_references\_form008\_\_ |
| Child Table | \_form008\_has\_doctor\_ |
| Parent Table | \_form\_008\_ |
| Foreign Key Columns | form008\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_form008\_has\_doctor\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_form008\_has\_doctor\_references\_form008\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = form008\_identifier | identifier | form008\_identifier |

### Reference \_\_fur\_references\_form008\_\_

#### Card of reference \_\_fur\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | \_\_fur\_references\_form008\_\_ |
| Code | \_\_fur\_references\_form008\_\_ |
| Child Table | \_fur\_ |
| Parent Table | \_form\_008\_ |
| Foreign Key Columns | form008\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fur\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fur\_references\_form008\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = form008\_identifier | identifier | form008\_identifier |

### Reference \_\_fur\_references\_medic\_authorizer\_\_

#### Card of reference \_\_fur\_references\_medic\_authorizer\_\_

|  |  |
| --- | --- |
| Name | \_\_fur\_references\_medic\_authorizer\_\_ |
| Code | \_\_fur\_references\_medic\_authorizer\_\_ |
| Child Table | \_fur\_ |
| Parent Table | \_doctor\_ |
| Foreign Key Columns | insured\_dead\_authorizer\_doctor |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fur\_references\_medic\_authorizer\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fur\_references\_medic\_authorizer\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = insured\_dead\_authorizer\_doctor | identifier | insured\_dead\_authorizer\_doctor |

### Reference \_\_fur\_references\_medic\_in\_charge\_\_

#### Card of reference \_\_fur\_references\_medic\_in\_charge\_\_

|  |  |
| --- | --- |
| Name | \_\_fur\_references\_medic\_in\_charge\_\_ |
| Code | \_\_fur\_references\_medic\_in\_charge\_\_ |
| Child Table | \_fur\_ |
| Parent Table | \_doctor\_ |
| Foreign Key Columns | insured\_dead\_in\_charge\_doctor |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_fur\_references\_medic\_in\_charge\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_fur\_references\_medic\_in\_charge\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = insured\_dead\_in\_charge\_doctor | identifier | insured\_dead\_in\_charge\_doctor |

### Reference \_\_objection\_references\_form008\_\_

#### Card of reference \_\_objection\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | \_\_objection\_references\_form008\_\_ |
| Code | \_\_objection\_references\_form008\_\_ |
| Child Table | \_objection\_ |
| Parent Table | \_form\_008\_ |
| Foreign Key Columns | form008\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_\_objection\_references\_form008\_\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_\_objection\_references\_form008\_\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = form008\_identifier | identifier | form008\_identifier |

### Reference \_fk\_action\_reference\_action\_

#### Card of reference \_fk\_action\_reference\_action\_

|  |  |
| --- | --- |
| Name | \_fk\_action\_reference\_action\_ |
| Code | \_fk\_action\_reference\_action\_ |
| Child Table | \_action\_ |
| Parent Table | \_action\_ |
| Foreign Key Columns | root\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_action\_reference\_action\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_action\_reference\_action\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = root\_identifier | identifier | root\_identifier |

### Reference \_fk\_action\_references\_app\_

#### Card of reference \_fk\_action\_references\_app\_

|  |  |
| --- | --- |
| Name | \_fk\_action\_references\_app\_ |
| Code | \_fk\_action\_references\_app\_ |
| Child Table | \_action\_ |
| Parent Table | \_app\_ |
| Foreign Key Columns | app\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_action\_references\_app\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_action\_references\_app\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = app\_identifier | identifier | app\_identifier |

### Reference \_fk\_ambulance\_use\_references\_form008\_

#### Card of reference \_fk\_ambulance\_use\_references\_form008\_

|  |  |
| --- | --- |
| Name | \_fk\_ambulance\_use\_references\_form008\_ |
| Code | \_fk\_ambulance\_use\_references\_form008\_ |
| Child Table | \_ambulance\_use\_ |
| Parent Table | \_form\_008\_ |
| Foreign Key Columns | form008\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_ambulance\_use\_references\_form008\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_fk\_ambulance\_use\_references\_form008\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = form008\_identifier | identifier | form008\_identifier |

### Reference \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_

#### Card of reference \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_

|  |  |
| --- | --- |
| Name | \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ |
| Code | \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_ |
| Child Table | \_ambulance\_use\_ |
| Parent Table | \_tariff\_ |
| Foreign Key Columns | ambulance\_km\_code |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_fk\_ambulance\_use\_references\_tariff\_ambulance\_km\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| code = ambulance\_km\_code | code | ambulance\_km\_code |

### Reference \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_

#### Card of reference \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_

|  |  |
| --- | --- |
| Name | \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ |
| Code | \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_ |
| Child Table | \_ambulance\_use\_ |
| Parent Table | \_tariff\_ |
| Foreign Key Columns | starting\_point\_code |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_fk\_ambulance\_use\_references\_tariff\_starting\_point\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| code = starting\_point\_code | code | starting\_point\_code |

### Reference \_fk\_catalog\_refereces\_app\_

#### Card of reference \_fk\_catalog\_refereces\_app\_

|  |  |
| --- | --- |
| Name | \_fk\_catalog\_refereces\_app\_ |
| Code | \_fk\_catalog\_refereces\_app\_ |
| Child Table | \_catalog\_ |
| Parent Table | \_app\_ |
| Foreign Key Columns | app\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_catalog\_refereces\_app\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_catalog\_refereces\_app\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = app\_identifier | identifier | app\_identifier |

### Reference \_fk\_department\_references\_department\_coordination\_

#### Card of reference \_fk\_department\_references\_department\_coordination\_

|  |  |
| --- | --- |
| Name | \_fk\_department\_references\_department\_coordination\_ |
| Code | \_fk\_department\_references\_department\_coordination\_ |
| Child Table | \_department\_ |
| Parent Table | \_department\_ |
| Foreign Key Columns | coordination\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_department\_references\_department\_coordination\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_department\_references\_department\_coordination\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = coordination\_identifier | identifier | coordination\_identifier |

### Reference \_fk\_department\_references\_enterprice\_

#### Card of reference \_fk\_department\_references\_enterprice\_

|  |  |
| --- | --- |
| Name | \_fk\_department\_references\_enterprice\_ |
| Code | \_fk\_department\_references\_enterprice\_ |
| Child Table | \_department\_ |
| Parent Table | \_enterprise\_ |
| Foreign Key Columns | enterprise\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_department\_references\_enterprice\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_department\_references\_enterprice\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = enterprise\_identifier | identifier | enterprise\_identifier |

### Reference \_fk\_enterprise\_references\_app\_

#### Card of reference \_fk\_enterprise\_references\_app\_

|  |  |
| --- | --- |
| Name | \_fk\_enterprise\_references\_app\_ |
| Code | \_fk\_enterprise\_references\_app\_ |
| Child Table | \_enterprise\_ |
| Parent Table | \_app\_ |
| Foreign Key Columns | app\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_enterprise\_references\_app\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_enterprise\_references\_app\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = app\_identifier | identifier | app\_identifier |

### Reference \_fk\_oxigen\_use\_references\_form008\_

#### Card of reference \_fk\_oxigen\_use\_references\_form008\_

|  |  |
| --- | --- |
| Name | \_fk\_oxigen\_use\_references\_form008\_ |
| Code | \_fk\_oxigen\_use\_references\_form008\_ |
| Child Table | \_oxygen\_use\_ |
| Parent Table | \_form\_008\_ |
| Foreign Key Columns | form008\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_oxigen\_use\_references\_form008\_

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of reference joins of the reference \_fk\_oxigen\_use\_references\_form008\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = form008\_identifier | identifier | form008\_identifier |

### Reference \_fk\_profile\_has\_action\_references\_action\_

#### Card of reference \_fk\_profile\_has\_action\_references\_action\_

|  |  |
| --- | --- |
| Name | \_fk\_profile\_has\_action\_references\_action\_ |
| Code | \_fk\_profile\_has\_action\_references\_action\_ |
| Child Table | \_profile\_has\_action\_ |
| Parent Table | \_action\_ |
| Foreign Key Columns | action identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_profile\_has\_action\_references\_action\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_profile\_has\_action\_references\_action\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = action identifier | identifier | action identifier |

### Reference \_fk\_profile\_has\_action\_references\_profile\_

#### Card of reference \_fk\_profile\_has\_action\_references\_profile\_

|  |  |
| --- | --- |
| Name | \_fk\_profile\_has\_action\_references\_profile\_ |
| Code | \_fk\_profile\_has\_action\_references\_profile\_ |
| Child Table | \_profile\_has\_action\_ |
| Parent Table | \_profile\_ |
| Foreign Key Columns | profile identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_profile\_has\_action\_references\_profile\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_profile\_has\_action\_references\_profile\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = profile identifier | identifier | profile identifier |

### Reference \_fk\_profile\_references\_app\_

#### Card of reference \_fk\_profile\_references\_app\_

|  |  |
| --- | --- |
| Name | \_fk\_profile\_references\_app\_ |
| Code | \_fk\_profile\_references\_app\_ |
| Child Table | \_profile\_ |
| Parent Table | \_app\_ |
| Foreign Key Columns | app identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_profile\_references\_app\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_profile\_references\_app\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = app identifier | identifier | app identifier |

### Reference \_fk\_user\_references\_department\_

#### Card of reference \_fk\_user\_references\_department\_

|  |  |
| --- | --- |
| Name | \_fk\_user\_references\_department\_ |
| Code | \_fk\_user\_references\_department\_ |
| Child Table | \_user\_ |
| Parent Table | \_department\_ |
| Foreign Key Columns | department\_identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_user\_references\_department\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_user\_references\_department\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = department\_identifier | identifier | department\_identifier |

### Reference \_fk\_user\_references\_profile\_

#### Card of reference \_fk\_user\_references\_profile\_

|  |  |
| --- | --- |
| Name | \_fk\_user\_references\_profile\_ |
| Code | \_fk\_user\_references\_profile\_ |
| Child Table | \_user\_ |
| Parent Table | \_profile\_ |
| Foreign Key Columns | profile identifier |
| Parent Role |  |
| Child Role |  |

#### List of diagrams containing the reference \_fk\_user\_references\_profile\_

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

#### List of reference joins of the reference \_fk\_user\_references\_profile\_

|  |  |  |
| --- | --- | --- |
| Display Name | Parent Table Column | Child Table Column |
| identifier = profile identifier | identifier | profile identifier |

## List of views

|  |  |
| --- | --- |
| *Name* | *Code* |
| monthly\_statistics | monthly\_statistics |

### View monthly\_statistics

#### Card of view monthly\_statistics

|  |  |
| --- | --- |
| Name | monthly\_statistics |
| Code | monthly\_statistics |

#### SQL query of the view monthly\_statistics

select

row\_number() OVER () "NRO",

f8.last\_update\_date "FECHA",

to\_char(f8.last\_update\_date, 'Mon') "MES",

fur.vehicle\_insurance\_carrier "ASEGURADORA",

f8.code "TRAMITE",

fur.health\_center\_admision\_date "FECHA DE INGRESO",

fur.health\_center\_egress\_date "FECHA DE EGRESO",

coalesce(fur.health\_center\_hosp\_care\_days, 0) "DIAS ESTADIA",

coalesce(obj.number, 'S/N') "RECLAMO Nº",

p.identificacion "CEDULA DE CIUDADANIA",

p.primer\_nombre || ' ' || p.segundo\_nombre || ' ' || p.apellido\_paterno || ' ' || p.apellido\_materno "PACIENTE",

p.edad "EDAD",

coalesce(d.first\_name || ' ' || d.last\_name, 'Ninguno') "MEDICO",

ef.total "PLANILLADO",

obj.value "OBJETADO",

ef.total - obj.value "FACTURADO",

obj.invoice "FACTURA",

ef.diagnostic "DIAGNOSTICO",

case fur.insured\_dead

when 1 then 'FALLECIDO'

else 'VIVO'

end "CONDICION DE ALTA",

f8.description "OBSERVACIONES"

from

soat.\_form\_008\_ f8

LEFT join soat.\_fur\_ fur on f8.identifier = fur.form008\_identifier

LEFT join soat.\_objection\_ obj on f8.identifier = obj.form008\_identifier

INNER JOIN soat.\_patient\_ p on f8.patient\_identifier = p.identificacion

LEFT join soat.\_form008\_has\_doctor\_ f8hd on f8.identifier = f8hd.form008\_identifier

LEFT join soat.\_doctor\_ d on f8hd.doctor\_identifier = d.identifier

LEFT join soat.\_expense\_form\_ ef on f8.identifier = ef.form008\_identifier;

#### List of referenced objects of the view monthly\_statistics

|  |  |
| --- | --- |
| Name | Code |
| \_doctor\_ | \_doctor\_ |
| \_expense\_form\_ | \_expense\_form\_ |
| \_form008\_has\_doctor\_ | \_form008\_has\_doctor\_ |
| \_form\_008\_ | \_form\_008\_ |
| \_fur\_ | \_fur\_ |
| \_objection\_ | \_objection\_ |
| \_patient\_ | \_patient\_ |

#### List of diagrams containing the view monthly\_statistics

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

#### List of columns of the view monthly\_statistics

|  |  |  |
| --- | --- | --- |
| Name | Code | Data Type |
| NRO | NRO |  |
| FECHA | FECHA |  |
| MES | MES |  |
| ASEGURADORA | ASEGURADORA |  |
| TRAMITE | TRAMITE |  |
| FECHA DE INGRESO | FECHA DE INGRESO |  |
| FECHA DE EGRESO | FECHA DE EGRESO |  |
| DIAS ESTADIA | DIAS ESTADIA |  |
| RECLAMO Nº | RECLAMO Nº |  |
| CEDULA DE CIUDADANIA | CEDULA DE CIUDADANIA |  |
| PACIENTE | PACIENTE |  |
| EDAD | EDAD |  |
| MEDICO | MEDICO |  |
| PLANILLADO | PLANILLADO |  |
| OBJETADO | OBJETADO |  |
| FACTURADO | FACTURADO |  |
| FACTURA | FACTURA |  |
| DIAGNOSTICO | DIAGNOSTICO |  |
| CONDICION DE ALTA | CONDICION DE ALTA |  |
| OBSERVACIONES | OBSERVACIONES |  |

## List of users

|  |  |
| --- | --- |
| *Name* | *Code* |
| app\_oper | app\_oper |
| app\_sys | app\_sys |
| inventory | inventory |
| kernel | kernel |
| soat | soat |

### User app\_oper

#### Card of user app\_oper

|  |  |
| --- | --- |
| Name | app\_oper |
| Code | app\_oper |

#### Description of the user app\_oper

Usuario de produccion para APP

### User app\_sys

#### Card of user app\_sys

|  |  |
| --- | --- |
| Name | app\_sys |
| Code | app\_sys |

#### List of extended inverse collections of the user app\_sys

|  |  |
| --- | --- |
| Parent | Inverse Collection Name |
| User 'inventory' | Schemas |
| User 'soat' | Schemas |

### User inventory

#### Card of user inventory

|  |  |
| --- | --- |
| Name | inventory |
| Code | inventory |

#### List of Owner

|  |  |
| --- | --- |
| Name | Code |
| app\_sys | app\_sys |

### User kernel

#### Card of user kernel

|  |  |
| --- | --- |
| Name | kernel |
| Code | kernel |

#### List of extended inverse collections of the user kernel

|  |  |
| --- | --- |
| Parent | Inverse Collection Name |
| Reference '\_fk\_action\_references\_app\_' | Inverse of Schema child |
| Reference '\_fk\_action\_references\_app\_' | Inverse of Schema parent |

#### List of tables of the user kernel

|  |  |
| --- | --- |
| Name | Code |
| \_action\_ | \_action\_ |
| \_app\_ | \_app\_ |
| \_catalog\_ | \_catalog\_ |
| \_department\_ | \_department\_ |
| \_enterprise\_ | \_enterprise\_ |
| \_profile\_ | \_profile\_ |
| \_profile\_has\_action\_ | \_profile\_has\_action\_ |
| \_user\_ | \_user\_ |

#### List of indexes of the user kernel

|  |  |
| --- | --- |
| Name | Code |
| action\_unique\_index | action\_unique\_index |
| identity\_number | identity\_number |
| user\_unique\_index | user\_unique\_index |

#### List of procedures of the user kernel

|  |  |
| --- | --- |
| Name | Code |
| change\_action\_state | change\_action\_state |
| change\_catalog\_state | change\_catalog\_state |
| change\_profile\_state | change\_profile\_state |
| change\_user\_password | change\_user\_password |
| change\_user\_state | change\_user\_state |
| get\_action | get\_action |
| get\_action\_by\_app\_id | get\_action\_by\_app\_id |
| get\_action\_by\_id | get\_action\_by\_id |
| get\_action\_script\_by\_handler | get\_action\_by\_handler |
| get\_all\_menu\_action\_by\_profile\_id\_root\_id | get\_all\_menu\_action\_by\_profile\_id\_root\_id |
| get\_app\_by\_id | get\_app\_by\_id |
| get\_asigned\_menu\_action\_by\_profile\_id\_root\_id | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| get\_catalog | get\_catalog |
| get\_catalog\_by\_id | get\_catalog\_by\_id |
| get\_catalog\_type | get\_catalog\_type |
| get\_coordination | get\_coordination |
| get\_department | get\_department |
| get\_department\_by\_enterprise\_id | get\_department\_by\_enterprise\_id |
| get\_department\_by\_id | get\_department\_by\_id |
| get\_enterprise | get\_enterprise |
| get\_profile | get\_profile |
| get\_profile\_by\_app\_id | get\_profile\_by\_app\_id |
| get\_profile\_by\_id | get\_profile\_by\_id |
| get\_user | get\_user |
| get\_user\_by\_id | get\_user\_by\_id |
| insert\_action | insert\_action |
| insert\_app | insert\_app |
| insert\_profile | insert\_profile |
| insert\_profile\_has\_action | insert\_profile\_has\_action |
| insert\_profile\_has\_action\_node | insert\_profile\_has\_action\_node |
| login\_user | login\_user |
| process\_catalog | process\_catalog |
| process\_department | process\_department |
| process\_enterprise | process\_enterprise |
| process\_user | process\_user |
| update\_action | update\_action |
| update\_profile | update\_profile |
| validate\_action\_access | validate\_action\_access |

### User soat

#### Card of user soat

|  |  |
| --- | --- |
| Name | soat |
| Code | soat |

#### List of Owner

|  |  |
| --- | --- |
| Name | Code |
| app\_sys | app\_sys |

#### List of tables of the user soat

|  |  |
| --- | --- |
| Name | Code |
| \_ambulance\_use\_ | \_ambulance\_use\_ |
| \_canton\_ | \_canton\_ |
| \_doctor\_ | \_doctor\_ |
| \_expense\_form\_ | \_expense\_form\_ |
| \_expense\_form\_medicine\_detail\_ | \_expense\_form\_medicine\_detail\_ |
| \_expense\_form\_tariff\_detail\_ | \_expense\_form\_tariff\_detail\_ |
| \_form008\_has\_doctor\_ | \_form008\_has\_doctor\_ |
| \_form\_008\_ | \_form\_008\_ |
| \_fur\_ | \_fur\_ |
| \_medicine\_ | \_medicine\_ |
| \_objection\_ | \_objection\_ |
| \_oxygen\_use\_ | \_oxygen\_use\_ |
| \_parish\_ | \_parish\_ |
| \_patient\_ | \_patient\_ |
| \_province\_ | \_province\_ |
| \_tariff\_ | \_tariff\_ |

#### List of procedures of the user soat

|  |  |
| --- | --- |
| Name | Code |
| get\_active\_patient | get\_active\_patient |
| get\_all\_doctor | get\_all\_doctor |
| get\_all\_expense\_form\_detail | get\_all\_expense\_form\_detail |
| get\_all\_medicine | get\_all\_medicine |
| get\_ambulance\_use\_by\_form008\_identifier | get\_ambulance\_use\_by\_form008\_identifier |
| get\_ambulance\_use\_tariff | get\_ambulance\_use\_tariff |
| get\_doctor\_by\_form008\_identifier | get\_doctor\_by\_form008\_identifier |
| get\_doctor\_by\_id | get\_doctor\_by\_id |
| get\_expense\_form\_by\_form008\_identifier | get\_expense\_form\_by\_form008\_identifier |
| get\_form008 | get\_form008 |
| get\_form008\_by\_id | get\_form008\_by\_id |
| get\_fur\_by\_form008\_identifier | get\_fur\_by\_form008\_identifier |
| get\_objection\_by\_form008\_identifier | get\_objection\_by\_form008\_identifier |
| get\_oxygen\_use\_by\_form008\_identifier | get\_oxygen\_use\_by\_form008\_identifier |
| get\_oxygen\_use\_tariff | get\_oxygen\_use\_tariff |
| get\_patient | get\_patient |
| get\_patient\_by\_id | get\_patient\_by\_id |
| get\_patient\_by\_id2 | get\_patient\_by\_id2 |
| get\_tariff\_as\_medicine | get\_tariff\_as\_medicine |
| process\_ambulance\_use | process\_ambulance\_use |
| process\_expense\_form | process\_expense\_form |
| process\_form008 | process\_form008 |
| process\_form008\_has\_doctor | process\_form008\_has\_doctor |
| process\_fur | process\_fur |
| process\_medicine | process\_medicine |
| process\_objection | process\_objection |
| process\_oxygen\_use | process\_oxygen\_use |
| process\_patient | process\_patient |

#### List of views of the user soat

|  |  |
| --- | --- |
| Name | Code |
| monthly\_statistics | monthly\_statistics |

## List of databases

|  |  |
| --- | --- |
| *Name* | *Code* |
| app | app |

### Database app

#### Card of database app

|  |  |
| --- | --- |
| Name | app |
| Code | app |
| Comment | Base de datos de APP |

#### Options of the database app

app

#### List of linked tablespaces of the database app

|  |  |
| --- | --- |
| Name | Code |
| app | app |

## List of tablespaces

|  |  |
| --- | --- |
| *Name* | *Code* |
| app | app |

### Tablespace app

#### Card of tablespace app

|  |  |
| --- | --- |
| Name | app |
| Code | app |
| Options |  |
| Database | <None> |

#### List of objects of the tablespace app

|  |
| --- |
| Short Description |
| Database 'app' |
| Table 'kernel'.'\_app\_' |

## List of procedures

|  |  |
| --- | --- |
| *Name* | *Code* |
| change\_action\_state | change\_action\_state |
| change\_catalog\_state | change\_catalog\_state |
| change\_profile\_state | change\_profile\_state |
| change\_user\_password | change\_user\_password |
| change\_user\_state | change\_user\_state |
| get\_action | get\_action |
| get\_action\_by\_app\_id | get\_action\_by\_app\_id |
| get\_action\_by\_id | get\_action\_by\_id |
| get\_action\_script\_by\_handler | get\_action\_by\_handler |
| get\_active\_patient | get\_active\_patient |
| get\_all\_doctor | get\_all\_doctor |
| get\_all\_expense\_form\_detail | get\_all\_expense\_form\_detail |
| get\_all\_medicine | get\_all\_medicine |
| get\_all\_menu\_action\_by\_profile\_id\_root\_id | get\_all\_menu\_action\_by\_profile\_id\_root\_id |
| get\_ambulance\_use\_by\_form008\_identifier | get\_ambulance\_use\_by\_form008\_identifier |
| get\_ambulance\_use\_tariff | get\_ambulance\_use\_tariff |
| get\_app\_by\_id | get\_app\_by\_id |
| get\_asigned\_menu\_action\_by\_profile\_id\_root\_id | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| get\_catalog | get\_catalog |
| get\_catalog\_by\_id | get\_catalog\_by\_id |
| get\_catalog\_type | get\_catalog\_type |
| get\_coordination | get\_coordination |
| get\_department | get\_department |
| get\_department\_by\_enterprise\_id | get\_department\_by\_enterprise\_id |
| get\_department\_by\_id | get\_department\_by\_id |
| get\_doctor\_by\_form008\_identifier | get\_doctor\_by\_form008\_identifier |
| get\_doctor\_by\_id | get\_doctor\_by\_id |
| get\_enterprise | get\_enterprise |
| get\_expense\_form\_by\_form008\_identifier | get\_expense\_form\_by\_form008\_identifier |
| get\_form008 | get\_form008 |
| get\_form008\_by\_id | get\_form008\_by\_id |
| get\_fur\_by\_form008\_identifier | get\_fur\_by\_form008\_identifier |
| get\_objection\_by\_form008\_identifier | get\_objection\_by\_form008\_identifier |
| get\_oxygen\_use\_by\_form008\_identifier | get\_oxygen\_use\_by\_form008\_identifier |
| get\_oxygen\_use\_tariff | get\_oxygen\_use\_tariff |
| get\_patient | get\_patient |
| get\_patient\_by\_id | get\_patient\_by\_id |
| get\_patient\_by\_id2 | get\_patient\_by\_id2 |
| get\_profile | get\_profile |
| get\_profile\_by\_app\_id | get\_profile\_by\_app\_id |
| get\_profile\_by\_id | get\_profile\_by\_id |
| get\_tariff\_as\_medicine | get\_tariff\_as\_medicine |
| get\_user | get\_user |
| get\_user\_by\_id | get\_user\_by\_id |
| get\_vehicle | get\_vehicle |
| insert\_action | insert\_action |
| insert\_app | insert\_app |
| insert\_product | insert\_product |
| insert\_profile | insert\_profile |
| insert\_profile\_has\_action | insert\_profile\_has\_action |
| insert\_profile\_has\_action\_node | insert\_profile\_has\_action\_node |
| login\_user | login\_user |
| process\_ambulance\_use | process\_ambulance\_use |
| process\_catalog | process\_catalog |
| process\_department | process\_department |
| process\_enterprise | process\_enterprise |
| process\_expense\_form | process\_expense\_form |
| process\_form008 | process\_form008 |
| process\_form008\_has\_doctor | process\_form008\_has\_doctor |
| process\_fur | process\_fur |
| process\_medicine | process\_medicine |
| process\_objection | process\_objection |
| process\_oxygen\_use | process\_oxygen\_use |
| process\_patient | process\_patient |
| process\_user | process\_user |
| update\_action | update\_action |
| update\_profile | update\_profile |
| validate\_action\_access | validate\_action\_access |

### Procedure change\_action\_state

#### Card of procedure change\_action\_state

|  |  |
| --- | --- |
| Name | change\_action\_state |
| Code | change\_action\_state |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure change\_action\_state

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifiers varchar(64)[],

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

\_\_singleStateArr varchar(32)[] = '{activada, desactivada, eliminada}';

\_\_multipleStateArr varchar(32)[] = '{activadas, desactivadas, eliminadas}';

\_\_length int4;

BEGIN

\_\_length := (select count(\*) from kernel.\_action\_ where identifier = any(\_identifiers));

if(\_\_length <= 0) then

RAISE NOTICE 'No hay acciones para procesar!!';

return false;

end if;

update

kernel.\_action\_

set

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = any(\_identifiers);

if (\_\_length = 1) then

RAISE NOTICE 'Acción %!!', \_\_singleStateArr[\_state + 1];

else

RAISE NOTICE '% acciones %!!', \_\_length, \_\_multipleStateArr[\_state + 1];

end if;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure change\_action\_state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure change\_action\_state

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure change\_catalog\_state

#### Card of procedure change\_catalog\_state

|  |  |
| --- | --- |
| Name | change\_catalog\_state |
| Code | change\_catalog\_state |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure change\_catalog\_state

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifiers int4[],

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

\_\_singleStateArr varchar(32)[] = '{activado, desactivado, eliminado}';

\_\_multipleStateArr varchar(32)[] = '{activados, desactivados, eliminados}';

\_\_length int4;

BEGIN

\_\_length := (select count(\*) from kernel.\_catalog\_ where identifier = any(\_identifiers));

if(\_\_length <= 0) then

RAISE NOTICE 'No hay elementos del catalogo para procesar!!';

return false;

end if;

update

kernel.\_catalog\_

set

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = any(\_identifiers);

if (\_\_length = 1) then

RAISE NOTICE 'Elemento del catalogo %!!', \_\_singleStateArr[\_state + 1];

else

RAISE NOTICE '% elementos del catalogo %!!', \_\_length, \_\_multipleStateArr[\_state + 1];

end if;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure change\_catalog\_state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure change\_catalog\_state

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure change\_profile\_state

#### Card of procedure change\_profile\_state

|  |  |
| --- | --- |
| Name | change\_profile\_state |
| Code | change\_profile\_state |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure change\_profile\_state

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

\_\_stateArr varchar(32)[] = '{activado, desactivado, eliminado}';

\_\_oldState int2;

BEGIN

\_\_oldState := (select state from kernel.\_profile\_ where identifier = \_identifier);

update

kernel.\_profile\_

set

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

if(\_\_oldState = 2) then

RAISE NOTICE 'Perfíl recuperado!!';

else

RAISE NOTICE 'Perfíl %!!', \_\_stateArr[\_state + 1];

end if;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure change\_profile\_state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure change\_profile\_state

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure change\_user\_password

#### Card of procedure change\_user\_password

|  |  |
| --- | --- |
| Name | change\_user\_password |
| Code | change\_user\_password |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure change\_user\_password

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_password varchar(32),

in \_last\_updater varchar(64)

) RETURNS boolean AS $$

DECLARE

BEGIN

update kernel.\_user\_ set

password = \_password,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

raise notice 'Contraseña actualizada!!';

return true;

EXCEPTION

when others then

raise notice '%', SQLERRM;

return false;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure change\_user\_password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_user\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure change\_user\_password

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure change\_user\_state

#### Card of procedure change\_user\_state

|  |  |
| --- | --- |
| Name | change\_user\_state |
| Code | change\_user\_state |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure change\_user\_state

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifiers int4[],

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

\_\_singleStateArr varchar(32)[] = '{activado, desactivado, eliminado}';

\_\_multipleStateArr varchar(32)[] = '{activados, desactivados, eliminados}';

\_\_length int4;

BEGIN

\_\_length := (select count(\*) from kernel.\_user\_ where identifier = any(\_identifiers));

if(\_\_length <= 0) then

RAISE NOTICE 'No hay usuarios para procesar!!';

return false;

end if;

update

kernel.\_user\_

set

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = any(\_identifiers);

if (\_\_length = 1) then

RAISE NOTICE 'Usuario %!!', \_\_singleStateArr[\_state + 1];

else

RAISE NOTICE '% usuarios %!!', \_\_length, \_\_multipleStateArr[\_state + 1];

end if;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure change\_user\_state

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_user\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure change\_user\_state

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_action

#### Card of procedure get\_action

|  |  |
| --- | --- |
| Name | get\_action |
| Code | get\_action |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_action

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_app\_identifier int4,

\_filter json,

\_pattern text,

\_limit int4,

\_offset int4

)

RETURNS TABLE(

count bigint,

identifier varchar(64),

type varchar(32),

handler varchar(64),

action varchar(64),

access varchar(32),

name varchar(32),

title varchar(64),

show\_in\_menu int2,

icon varchar(32),

icon\_color varchar(32),

run\_mode varchar(32),

state int2

) AS $$

DECLARE

\_\_typeArr varchar(32)[] = '{Etiqueta, Accion, Dependencia}';

\_\_accessArr varchar(32)[] = '{Privado, Registrado, Heredado, Público}';

\_\_runModeArr varchar(32)[] = '{Pestaña, Dialogo, Oculto}';

BEGIN

\_pattern := '%' || \_pattern || '%';

RETURN QUERY(

SELECT

count(\*) over() "count",

a.identifier,

\_\_typeArr[a.type + 1] "type",

a.handler,

a.action,

\_\_accessArr[a.access + 1] "access",

a.name,

a.title,

a.show\_in\_menu,

a.icon,

a.icon\_color,

\_\_runModeArr[a.run\_mode + 1] "run\_mode",

a.state

FROM

kernel.\_action\_ a

where

a.app\_identifier = \_app\_identifier and

a.state::varchar(1) like '%' || (\_filter->>'state')::varchar(1) || '%' and

a.show\_in\_menu::varchar(1) like '%' || (\_filter->>'show\_in\_menu')::varchar(1) || '%' and

a.type::varchar(1) like '%' || (\_filter->>'type')::varchar(1) || '%' and

a.access::varchar(1) like '%' || (\_filter->>'access')::varchar(1) || '%' and

coalesce(a.run\_mode::varchar(1), '') like '%' || (\_filter->>'run\_mode')::varchar(1) || '%' and

(

a.handler ilike \_pattern or

a.action ilike \_pattern or

a.name ilike \_pattern or

a.title ilike \_pattern

)

order by a.handler, a.action asc

offset nullif(\_offset, 0)

limit nullif(\_limit, 0)

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_action

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_action

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_action\_by\_app\_id

#### Card of procedure get\_action\_by\_app\_id

|  |  |
| --- | --- |
| Name | get\_action\_by\_app\_id |
| Code | get\_action\_by\_app\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_action\_by\_app\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4

)

RETURNS SETOF kernel.\_action\_ AS $$

SELECT

a.\*

FROM

kernel.\_action\_ a

inner join kernel.\_app\_ app on app.identifier = a.app\_identifier

where

app.identifier = \_app\_identifier and

a.state = 0

order by a.handler, a.action asc;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_action\_by\_app\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_action\_by\_app\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_action\_by\_id

#### Card of procedure get\_action\_by\_id

|  |  |
| --- | --- |
| Name | get\_action\_by\_id |
| Code | get\_action\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_action\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier varchar(32),

in \_app\_identifier int4

)

RETURNS SETOF kernel.\_action\_ AS $$

SELECT

a.\*

FROM

kernel.\_action\_ a

inner join kernel.\_app\_ app on app.identifier = a.app\_identifier

where

a.identifier = \_identifier and

app.identifier = \_app\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_action\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_action\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_action\_script\_by\_handler

#### Card of procedure get\_action\_script\_by\_handler

|  |  |
| --- | --- |
| Name | get\_action\_script\_by\_handler |
| Code | get\_action\_by\_handler |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_action\_script\_by\_handler

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4,

in \_handler varchar(64)

)

RETURNS SETOF kernel.\_action\_ AS $$

SELECT

a.\*

FROM

kernel.\_action\_ a

where

a.app\_identifier = \_app\_identifier and

a.state != 2 and

a.handler = \_handler

order by a.action asc;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_action\_script\_by\_handler

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_action\_script\_by\_handler

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_active\_patient

#### Card of procedure get\_active\_patient

|  |  |
| --- | --- |
| Name | get\_active\_patient |
| Code | get\_active\_patient |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_active\_patient

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% ()

RETURNS SETOF soat.\_patient\_ AS $$

select

p.\*

from

soat.\_patient\_ p

where

p.state = 0

order by p.last\_update\_date desc, p.primer\_nombre, p.segundo\_nombre, p.apellido\_paterno, p.apellido\_materno;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_active\_patient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_patient\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_active\_patient

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_all\_doctor

#### Card of procedure get\_all\_doctor

|  |  |
| --- | --- |
| Name | get\_all\_doctor |
| Code | get\_all\_doctor |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_all\_doctor

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% ()

RETURNS setof soat.\_doctor\_ AS $$

select d.\* from soat.\_doctor\_ d where d.state = 0;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_all\_doctor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_doctor\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_all\_doctor

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_all\_expense\_form\_detail

#### Card of procedure get\_all\_expense\_form\_detail

|  |  |
| --- | --- |
| Name | get\_all\_expense\_form\_detail |
| Code | get\_all\_expense\_form\_detail |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_all\_expense\_form\_detail

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_expense\_form\_identifier int4

)

RETURNS setof soat.\_expense\_form\_tariff\_detail\_ AS $$

select \* from soat.\_expense\_form\_tariff\_detail\_ where expense\_form\_identifier = \_expense\_form\_identifier

union all

select \* from soat.\_expense\_form\_medicine\_detail\_ where expense\_form\_identifier = \_expense\_form\_identifier

order by item\_type, item\_description;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_all\_expense\_form\_detail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_expense\_form\_medicine\_detail\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_expense\_form\_tariff\_detail\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_all\_expense\_form\_detail

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_all\_medicine

#### Card of procedure get\_all\_medicine

|  |  |
| --- | --- |
| Name | get\_all\_medicine |
| Code | get\_all\_medicine |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_all\_medicine

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% ()

RETURNS SETOF soat.\_medicine\_ AS $$

select \* from soat.\_medicine\_ where state = 0;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_all\_medicine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_medicine\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_all\_medicine

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_all\_menu\_action\_by\_profile\_id\_root\_id

#### Card of procedure get\_all\_menu\_action\_by\_profile\_id\_root\_id

|  |  |
| --- | --- |
| Name | get\_all\_menu\_action\_by\_profile\_id\_root\_id |
| Code | get\_all\_menu\_action\_by\_profile\_id\_root\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_all\_menu\_action\_by\_profile\_id\_root\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_profile\_identifier int4,

in \_app\_identifier int4,

in \_root\_identifier varchar(32)

)

RETURNS TABLE(

identifier VARCHAR(64),

type int2,

access INT2,

name VARCHAR(32),

title VARCHAR(64),

icon VARCHAR(32),

icon\_color VARCHAR(32),

state INT2,

has\_children int2,

assigned int2

) AS $$

DECLARE

BEGIN

RETURN QUERY(

SELECT

a.identifier,

a.type,

a.access,

a.name,

a.title,

a.icon,

a.icon\_color,

a.state,

case (select count(\*) from kernel.\_action\_ a2 where a2.root\_identifier = a.identifier and a2.type != 2 and a2.state = 0)

when 0 then 0

else 1

end::int2 has\_children,

case a.type

when 0 then 0

else

case (select count(\*) from kernel.\_profile\_has\_action\_ pha where pha.action\_identifier = a.identifier and pha.profile\_identifier = \_profile\_identifier)

when 0 then 0

else 1

end

end::int2 assigned

FROM

kernel.\_action\_ a

WHERE

coalesce(a.root\_identifier, '') = \_root\_identifier and

a.state = 0 and

a.type != 2

ORDER BY

a.name ASC

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_all\_menu\_action\_by\_profile\_id\_root\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_has\_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_all\_menu\_action\_by\_profile\_id\_root\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_ambulance\_use\_by\_form008\_identifier

#### Card of procedure get\_ambulance\_use\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | get\_ambulance\_use\_by\_form008\_identifier |
| Code | get\_ambulance\_use\_by\_form008\_identifier |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_ambulance\_use\_by\_form008\_identifier

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_form008\_identifier int4

)

RETURNS SETOF soat.\_ambulance\_use\_ AS $$

select \* from soat.\_ambulance\_use\_ au where au.form008\_identifier = \_\_form008\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_ambulance\_use\_by\_form008\_identifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_ambulance\_use\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_ambulance\_use\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_ambulance\_use\_tariff

#### Card of procedure get\_ambulance\_use\_tariff

|  |  |
| --- | --- |
| Name | get\_ambulance\_use\_tariff |
| Code | get\_ambulance\_use\_tariff |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_ambulance\_use\_tariff

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% ()

RETURNS table(

code numeric,

type varchar(360),

description varchar(360),

uvr decimal(19,5),

fcm decimal(19,5),

value decimal(19,5)

) AS $$

DECLARE

\_\_type varchar(360);

\_\_fcm TEXT;

BEGIN

\_\_type = (select c.value from kernel.\_catalog\_ c where c.type = 'TARIFF\_PARAMETERS' and c.name = 'AMBULANCE\_TYPE');

\_\_fcm = (select c.value from kernel.\_catalog\_ c where c.type = 'TARIFF\_PARAMETERS' and c.name = 'CURRENT\_FCM');

if(\_\_type is null) then

\_\_type := 'SERVICIO DE TRANSPORTE Y ASISTENCIA SANITARIA - CONTENIDO DE SERVICIO DE TRANSPORTE ASISTENCIA SANITARIO%';

end if;

if(\_\_fcm is null) then

\_\_fcm := '1';

end if;

if(\_\_fcm = '1') then

return query(select t.code, t.type, t.description, t.uvr1, t.fcm1, t.valor1 from soat.\_tariff\_ t where t.type like \_\_type);

elsif (\_\_fcm = '2') then

return query(select t.code, t.type, t.description, t.uvr2, t.fcm2, t.valor2 from soat.\_tariff\_ t where t.type like \_\_type);

else

return query(select t.code, t.type, t.description, t.uvr3, t.fcm3, t.valor3 from soat.\_tariff\_ t where t.type like \_\_type);

end if;

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_ambulance\_use\_tariff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_tariff\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_ambulance\_use\_tariff

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_app\_by\_id

#### Card of procedure get\_app\_by\_id

|  |  |
| --- | --- |
| Name | get\_app\_by\_id |
| Code | get\_app\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_app\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (\_identifier int)

RETURNS SETOF kernel.\_app\_ AS

$BODY$

select

a.\*

from

kernel.\_app\_ a

where

a.identifier = \_identifier;

$BODY$

LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_app\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_app\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_asigned\_menu\_action\_by\_profile\_id\_root\_id

#### Card of procedure get\_asigned\_menu\_action\_by\_profile\_id\_root\_id

|  |  |
| --- | --- |
| Name | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| Code | get\_asigned\_menu\_action\_by\_profile\_id\_root\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_asigned\_menu\_action\_by\_profile\_id\_root\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_profile\_identifier int4,

in \_app\_identifier int4,

in \_root\_identifier varchar(32)

)

RETURNS SETOF kernel.\_action\_ AS $$

SELECT

act.\*

FROM

kernel.\_action\_ act

inner join kernel.\_profile\_has\_action\_ pa on act.identifier = pa.action\_identifier

inner join kernel.\_profile\_ p on pa.profile\_identifier = p.identifier and p.identifier = \_profile\_identifier and p.state = 0

inner join kernel.\_app\_ app on p.app\_identifier = app.identifier and act.app\_identifier = app.identifier and app.identifier = \_app\_identifier and app.state = 0

where

act.state = 0 and

coalesce(act.root\_identifier, '') = \_root\_identifier and

act.show\_in\_menu = 1

order by act.name;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_asigned\_menu\_action\_by\_profile\_id\_root\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_has\_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_asigned\_menu\_action\_by\_profile\_id\_root\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_catalog

#### Card of procedure get\_catalog

|  |  |
| --- | --- |
| Name | get\_catalog |
| Code | get\_catalog |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_catalog

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_app\_identifier int4,

\_filter json,

\_pattern text,

\_limit int4,

\_offset int4

)

RETURNS TABLE(

count bigint,

identifier int4,

type varchar(32),

name varchar(64),

value text,

state int2

) AS $$

DECLARE

BEGIN

RETURN QUERY(

select

count(c.\*) over(),

c.identifier,

c.type,

c.name,

c.value,

c.state

from

kernel.\_catalog\_ c

where

c.app\_identifier = \_app\_identifier and

c.type like '%' || (\_filter->>'type')::varchar(32) || '%' and

c.name ilike '%' || \_pattern || '%' and

c.state::varchar(1) like '%' || (\_filter->>'state')::varchar(1) || '%'

order by c.type, c.name, c.value asc

offset nullif(\_offset, 0)

limit nullif(\_limit, 0)

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_catalog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_catalog

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_catalog\_by\_id

#### Card of procedure get\_catalog\_by\_id

|  |  |
| --- | --- |
| Name | get\_catalog\_by\_id |
| Code | get\_catalog\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_catalog\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4

)

RETURNS SETOF kernel.\_catalog\_ AS $$

SELECT

c.\*

FROM

kernel.\_catalog\_ c

where

c.identifier = \_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_catalog\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_catalog\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_catalog\_type

#### Card of procedure get\_catalog\_type

|  |  |
| --- | --- |
| Name | get\_catalog\_type |
| Code | get\_catalog\_type |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_catalog\_type

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4

)

RETURNS SETOF kernel.\_catalog\_ AS $$

SELECT DISTINCT

NULL::int4 identifier,

NULL::int4 app\_identifier,

c.type,

NULL::varchar(64) "name",

NULL::text "value",

NULL::text description,

NULL::int2 state,

NULL::int4 last\_updater,

NULL::timestamp last\_update\_date

FROM

kernel.\_catalog\_ c

where

c.app\_identifier = \_app\_identifier

order by

c.type asc;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_catalog\_type

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_catalog\_type

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_coordination

#### Card of procedure get\_coordination

|  |  |
| --- | --- |
| Name | get\_coordination |
| Code | get\_coordination |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_coordination

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_enterprise\_identifier int4

)

RETURNS SETOF kernel.\_department\_ AS $$

SELECT

d.\*

FROM

kernel.\_department\_ d

WHERE

d.enterprise\_identifier = \_enterprise\_identifier and

d.coordination\_identifier is null and

d.state = 0;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_coordination

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_department\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_coordination

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_department

#### Card of procedure get\_department

|  |  |
| --- | --- |
| Name | get\_department |
| Code | get\_department |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_department

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_enterprise\_identifier int4,

in \_filter json,

in \_pattern text,

in \_limit int4,

in \_offset int4

)

RETURNS TABLE(

count bigint,

identifier int4,

type varchar(64),

name varchar(64),

code varchar(64),

coordination\_name varchar(64),

state int2

) AS $$

DECLARE

BEGIN

\_pattern := '%' || \_pattern || '%';

RETURN QUERY(

select

count(\*) over(),

d.identifier,

case

when d.coordination\_identifier is null then 'Coordinación'

else 'Departamento'

end::varchar(64),

d.name,

d.code,

coalesce(c.name, 'Ninguna'),

d.state

from

kernel.\_department\_ d

left join kernel.\_department\_ c on d.coordination\_identifier = c.identifier and c.state = 0 and c.enterprise\_identifier = \_enterprise\_identifier

where

d.enterprise\_identifier = \_enterprise\_identifier and

d.state::varchar(1) like '%' || (\_filter->>'state')::varchar(1) || '%' and

coalesce(c.identifier, -1)::varchar(1) like '%' || (\_filter->>'coordination')::varchar(1) || '%' and

(

d.name ilike \_pattern or

d.code ilike \_pattern

)

order by d.name asc

offset nullif(\_offset, 0)

limit nullif(\_limit, 0)

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_department

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_department\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_department

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_department\_by\_enterprise\_id

#### Card of procedure get\_department\_by\_enterprise\_id

|  |  |
| --- | --- |
| Name | get\_department\_by\_enterprise\_id |
| Code | get\_department\_by\_enterprise\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_department\_by\_enterprise\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_enterprise\_identifier int4

)

RETURNS SETOF kernel.\_department\_ AS $$

SELECT

d.\*

FROM

kernel.\_department\_ d

where

d.enterprise\_identifier = \_enterprise\_identifier and

d.state = 0;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_department\_by\_enterprise\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_department\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_department\_by\_enterprise\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_department\_by\_id

#### Card of procedure get\_department\_by\_id

|  |  |
| --- | --- |
| Name | get\_department\_by\_id |
| Code | get\_department\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_department\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4

)

RETURNS SETOF kernel.\_department\_ AS $$

SELECT

d.\*

FROM

kernel.\_department\_ d

where

d.identifier = \_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_department\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_department\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_department\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_doctor\_by\_form008\_identifier

#### Card of procedure get\_doctor\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | get\_doctor\_by\_form008\_identifier |
| Code | get\_doctor\_by\_form008\_identifier |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_doctor\_by\_form008\_identifier

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_form008\_identifier int4

)

RETURNS setof soat.\_doctor\_ AS $$

select

d.\*

from

soat.\_doctor\_ d

inner join soat.\_form008\_has\_doctor\_ fhd on d.identifier = fhd.doctor\_identifier

where

d.state = 0 and

fhd.form008\_identifier = \_\_form008\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_doctor\_by\_form008\_identifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_doctor\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_form008\_has\_doctor\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_doctor\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_doctor\_by\_id

#### Card of procedure get\_doctor\_by\_id

|  |  |
| --- | --- |
| Name | get\_doctor\_by\_id |
| Code | get\_doctor\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_doctor\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_identifier int4

)

RETURNS setof soat.\_doctor\_ AS $$

select

d.\*

from

soat.\_doctor\_ d

where

d.identifier = \_\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_doctor\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_doctor\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_doctor\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_enterprise

#### Card of procedure get\_enterprise

|  |  |
| --- | --- |
| Name | get\_enterprise |
| Code | get\_enterprise |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_enterprise

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% ()

RETURNS SETOF kernel.\_enterprise\_ AS $$

select

e.\*

from

kernel.\_enterprise\_ e

where

e.state = 0;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_enterprise

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_enterprise\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_enterprise\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure get\_enterprise

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_expense\_form\_by\_form008\_identifier

#### Card of procedure get\_expense\_form\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | get\_expense\_form\_by\_form008\_identifier |
| Code | get\_expense\_form\_by\_form008\_identifier |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_expense\_form\_by\_form008\_identifier

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_form008\_identifier int4

)

RETURNS SETOF soat.\_expense\_form\_ AS $$

select \* from soat.\_expense\_form\_ where form008\_identifier = \_form008\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_expense\_form\_by\_form008\_identifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_expense\_form\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_expense\_form\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_form008

#### Card of procedure get\_form008

|  |  |
| --- | --- |
| Name | get\_form008 |
| Code | get\_form008 |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_form008

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_filter json,

\_pattern text,

\_limit int4,

\_offset int4

)

RETURNS TABLE(

count bigint,

identifier int4,

code varchar(8),

"last\_update\_date" varchar(10),

patient varchar(260),

state int2

) AS $$

DECLARE

\_\_stateFilter varchar(3) = '%' || (\_filter->>'state')::varchar(1) || '%';

BEGIN

\_pattern := '%' || \_pattern || '%';

RETURN QUERY(

SELECT

count(\*) over() "count",

f.identifier,

f.code,

to\_char(f.last\_update\_date, 'DD/MM/YYYY')::varchar(10) "last\_update\_date",

(p.primer\_nombre || ' ' || p.segundo\_nombre || ' ' || p.apellido\_paterno || ' ' || p.apellido\_materno || ' (' || p.identificacion || ')')::varchar(260) patient,

f.state

FROM

soat.\_form\_008\_ f

inner join soat.\_patient\_ p on f.patient\_identifier = p.identificacion

where

f.state::varchar(1) like \_\_stateFilter and

(f.code ilike \_pattern or p.primer\_nombre ilike \_pattern or p.segundo\_nombre ilike \_pattern or p.apellido\_paterno ilike \_pattern or p.apellido\_materno ilike \_pattern or p.identificacion ilike \_pattern)

order by f.last\_update\_date desc

offset nullif(\_offset, 0)

limit nullif(\_limit, 0)

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_form008

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_form\_008\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_patient\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_form008

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_form008\_by\_id

#### Card of procedure get\_form008\_by\_id

|  |  |
| --- | --- |
| Name | get\_form008\_by\_id |
| Code | get\_form008\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_form008\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_identifier int4

)

RETURNS setof soat.\_form\_008\_ AS

$$

select

\*

from

soat.\_form\_008\_

where

identifier = \_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_form008\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_form\_008\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_form008\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_fur\_by\_form008\_identifier

#### Card of procedure get\_fur\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | get\_fur\_by\_form008\_identifier |
| Code | get\_fur\_by\_form008\_identifier |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_fur\_by\_form008\_identifier

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_form008\_identifier int4

)

RETURNS setof soat.\_fur\_ AS $$

SELECT fur.identifier, form008\_identifier, patient\_condition\_walker, patient\_condition\_occupant,

accident\_address, p.name, c.name, pr.name,

accident\_neighborhood, accident\_date, accident\_hour, accident\_minute,

accident\_report, vehicle\_insured, vehicle\_not\_insured, vehicle\_not\_identified,

vehicle\_coverage\_code, vehicle\_license\_plate, vehicle\_soat\_number,

vehicle\_insurance\_carrier, vehicle\_police\_start\_date, vehicle\_police\_end\_date,

health\_center\_admision\_date, health\_center\_admision\_hour, health\_center\_admision\_minute,

health\_center\_history\_number, health\_center\_egress\_date, health\_center\_ambulatory\_care,

health\_center\_hosp\_care, health\_center\_hosp\_care\_days, health\_center\_admision\_report,

health\_center\_egress\_report, health\_center\_referred\_fr, health\_center\_referred\_fr\_city,

health\_center\_referred\_fr\_date, health\_center\_referred\_to, health\_center\_referred\_to\_city,

health\_center\_referred\_to\_date, insured\_dead, insured\_dead\_basic\_reazon,

insured\_dead\_direct\_reazon, insured\_dead\_date, insured\_dead\_hour,

insured\_dead\_minute, insured\_dead\_in\_charge\_doctor, insured\_dead\_authorizer\_doctor,

state, description, last\_update\_date, last\_updater

FROM soat.\_fur\_ fur

inner join soat.\_province\_ p on accident\_province::int4 = p.identifier

inner join soat.\_canton\_ c on accident\_canton::int4 = c.identifier

inner join soat.\_parish\_ pr on accident\_parish::int4 = pr.identifier

where form008\_identifier = \_\_form008\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_fur\_by\_form008\_identifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_canton\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_fur\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_parish\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_province\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_fur\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_objection\_by\_form008\_identifier

#### Card of procedure get\_objection\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | get\_objection\_by\_form008\_identifier |
| Code | get\_objection\_by\_form008\_identifier |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_objection\_by\_form008\_identifier

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_form008\_identifier int4

)

RETURNS SETOF soat.\_objection\_ AS $$

select \* from soat.\_objection\_ where form008\_identifier = \_form008\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_objection\_by\_form008\_identifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_objection\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_objection\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_oxygen\_use\_by\_form008\_identifier

#### Card of procedure get\_oxygen\_use\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | get\_oxygen\_use\_by\_form008\_identifier |
| Code | get\_oxygen\_use\_by\_form008\_identifier |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_oxygen\_use\_by\_form008\_identifier

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_form008\_identifier int4

)

RETURNS SETOF soat.\_oxygen\_use\_ AS $$

select \* from soat.\_oxygen\_use\_ ou where ou.form008\_identifier = \_\_form008\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_oxygen\_use\_by\_form008\_identifier

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_oxygen\_use\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_oxygen\_use\_by\_form008\_identifier

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_oxygen\_use\_tariff

#### Card of procedure get\_oxygen\_use\_tariff

|  |  |
| --- | --- |
| Name | get\_oxygen\_use\_tariff |
| Code | get\_oxygen\_use\_tariff |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_oxygen\_use\_tariff

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% ()

RETURNS table(

code numeric,

type varchar(360),

description varchar(360),

uvr decimal(19,5),

fcm decimal(19,5),

value decimal(19,5)

) AS $$

DECLARE

\_\_oxygenCatalogCode varchar(64) = 'CODE';

\_\_oxygenCatalogType varchar(32) = 'OXYGEN\_TARIFF';

\_\_oxygenCatalogDescription varchar(64) = 'DESCRIPTION';

\_\_oxygenCatalogUVR varchar(64) = 'UVR';

\_\_oxygenCatalogFCM varchar(64) = 'FCM';

\_\_code text;

\_\_description text;

\_\_uvr text;

\_\_fcm text;

BEGIN

\_\_code = (select c.value from kernel.\_catalog\_ c where c.type = \_\_oxygenCatalogType and c.name = \_\_oxygenCatalogCode);

\_\_description = (select c.value from kernel.\_catalog\_ c where c.type = \_\_oxygenCatalogType and c.name = \_\_oxygenCatalogDescription);

\_\_uvr = (select c.value from kernel.\_catalog\_ c where c.type = \_\_oxygenCatalogType and c.name = \_\_oxygenCatalogUVR);

\_\_fcm = (select c.value from kernel.\_catalog\_ c where c.type = \_\_oxygenCatalogType and c.name = \_\_oxygenCatalogFCM);

if(\_\_code is null) then

\_\_code := '241208';

end if;

if(\_\_description is null) then

\_\_description := 'CALCULO DE CONSUMO DE OXIGENO POR MINUTOS';

end if;

if(\_\_uvr is null) then

\_\_uvr := '3';

end if;

if(\_\_fcm is null) then

\_\_fcm := '0.01';

end if;

return query(select \_\_code::numeric, null::varchar(360), \_\_description::varchar(360), \_\_uvr::decimal(19,5), \_\_fcm::decimal(19,5), (\_\_uvr::decimal(19,5) \* \_\_fcm::decimal(19,5)) );

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_oxygen\_use\_tariff

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] |  |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_tariff\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure get\_oxygen\_use\_tariff

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_patient

#### Card of procedure get\_patient

|  |  |
| --- | --- |
| Name | get\_patient |
| Code | get\_patient |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_patient

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_filter json,

\_pattern text,

\_limit int4,

\_offset int4

)

RETURNS TABLE(

count bigint,

identificacion varchar(64),

primer\_nombre varchar(260),

state int2

) AS $$

DECLARE

\_\_stateFilter varchar(3) = '%' || (\_filter->>'state')::varchar(1) || '%';

BEGIN

\_pattern := '%' || \_pattern || '%';

RETURN QUERY(

SELECT

count(\*) over() "count",

p.identificacion,

(p.primer\_nombre || ' ' || p.segundo\_nombre || ' ' || p.apellido\_paterno || ' ' || p.apellido\_materno)::varchar(260) primer\_nombre,

p.state

FROM

soat.\_patient\_ p

where

p.state::varchar(1) like \_\_stateFilter and

(p.primer\_nombre ilike \_pattern or p.segundo\_nombre ilike \_pattern or p.apellido\_paterno ilike \_pattern or p.apellido\_materno ilike \_pattern or p.identificacion ilike \_pattern)

order by p.last\_update\_date desc, p.primer\_nombre, p.segundo\_nombre, p.apellido\_paterno, p.apellido\_materno

offset nullif(\_offset, 0)

limit nullif(\_limit, 0)

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_patient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_patient\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_patient\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure get\_patient

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_patient\_by\_id

#### Card of procedure get\_patient\_by\_id

|  |  |
| --- | --- |
| Name | get\_patient\_by\_id |
| Code | get\_patient\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_patient\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identificacion VARCHAR(64)

)

RETURNS SETOF soat.\_patient\_ AS $$

select

p.\*

from

soat.\_patient\_ p

where

p.identificacion = \_identificacion;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_patient\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_patient\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_patient\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_patient\_by\_id2

#### Card of procedure get\_patient\_by\_id2

|  |  |
| --- | --- |
| Name | get\_patient\_by\_id2 |
| Code | get\_patient\_by\_id2 |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_patient\_by\_id2

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identificacion VARCHAR(64)

)

RETURNS SETOF soat.\_patient\_ AS $$

SELECT identificacion, primer\_nombre, segundo\_nombre, apellido\_materno,

apellido\_paterno, direccion, barrio, pr.name, c.name, p.name,

zona, telefono, fecha\_nacimiento, lugar\_nacimiento, nacionalidad,

grupo\_cultural, edad, genero, estado\_civil, instruccion, fecha\_admision,

ocupacion, empresa\_trabajo, tipo\_seguro\_salud, referido\_de, contacto\_emergencia,

parentezco\_contacto\_emergencia, direccion\_contacto\_emergencia,

telefono\_contacto\_emergencia, state, description, last\_update\_date,

last\_updater

FROM soat.\_patient\_

inner join soat.\_province\_ p on provincia::int4 = p.identifier

inner join soat.\_canton\_ c on canton::int4 = c.identifier

inner join soat.\_parish\_ pr on parroquia::int4 = pr.identifier

where

identificacion = \_identificacion;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_patient\_by\_id2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_canton\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_parish\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_patient\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_province\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_patient\_by\_id2

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_profile

#### Card of procedure get\_profile

|  |  |
| --- | --- |
| Name | get\_profile |
| Code | get\_profile |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_profile

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_app\_identifier int4,

\_filter json,

\_pattern text,

\_limit int4,

\_offset int4

)

RETURNS TABLE(

count bigint,

identifier int4,

name varchar(64),

state int2

) AS $$

DECLARE

BEGIN

RETURN QUERY(

select

count(\*) over(),

p.identifier,

p.name,

p.state

from

kernel.\_profile\_ p

where

p.app\_identifier = \_app\_identifier and

p.state::varchar(1) like '%' || (\_filter->>'state')::varchar(1) || '%' and

p.name ilike '%' || \_pattern || '%'

order by p.name asc

offset nullif(\_offset, 0)

limit nullif(\_limit, 0)

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_profile

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_profile\_by\_app\_id

#### Card of procedure get\_profile\_by\_app\_id

|  |  |
| --- | --- |
| Name | get\_profile\_by\_app\_id |
| Code | get\_profile\_by\_app\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_profile\_by\_app\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4

)

RETURNS SETOF kernel.\_profile\_ AS $$

select

p.\*

from

kernel.\_profile\_ p

where

p.app\_identifier = \_app\_identifier and

p.state = 0

order by

p.name;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_profile\_by\_app\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_profile\_by\_app\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_profile\_by\_id

#### Card of procedure get\_profile\_by\_id

|  |  |
| --- | --- |
| Name | get\_profile\_by\_id |
| Code | get\_profile\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_profile\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier integer,

in \_app\_identifier integer

)

RETURNS SETOF kernel.\_profile\_ AS $$

select

p.\*

from kernel.\_profile\_ p

inner join kernel.\_app\_ a on a.identifier = p.app\_identifier

where

p.state != 2 and

p.identifier = \_identifier and

a.identifier = \_app\_identifier;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_profile\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_profile\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_tariff\_as\_medicine

#### Card of procedure get\_tariff\_as\_medicine

|  |  |
| --- | --- |
| Name | get\_tariff\_as\_medicine |
| Code | get\_tariff\_as\_medicine |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_tariff\_as\_medicine

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% ()

RETURNS table(

identifier numeric,

code numeric,

name varchar(360),

type varchar(360),

value decimal(19,5)

) AS $$

DECLARE

\_\_type varchar(360);

\_\_fcm TEXT;

BEGIN

\_\_type = (select c.value from kernel.\_catalog\_ c where c.type = 'TARIFF\_PARAMETERS' and c.name = 'AMBULANCE\_TYPE');

\_\_fcm = (select c.value from kernel.\_catalog\_ c where c.type = 'TARIFF\_PARAMETERS' and c.name = 'CURRENT\_FCM');

if(\_\_type is null) then

\_\_type := 'SERVICIO DE TRANSPORTE Y ASISTENCIA SANITARIA - CONTENIDO DE SERVICIO DE TRANSPORTE ASISTENCIA SANITARIO%';

end if;

if(\_\_fcm is null) then

\_\_fcm := '1';

end if;

if(\_\_fcm = '1') then

return query(select t.code identifier, t.code code, t.description "name", t.type "type", t.valor1 "value" from soat.\_tariff\_ t where t.type not like \_\_type);

elsif (\_\_fcm = '2') then

return query(select t.code identifier, t.code code, t.description "name", t.type "type", t.valor2 "value" from soat.\_tariff\_ t where t.type not like \_\_type);

else

return query(select t.code identifier, t.code code, t.description "name", t.type "type", t.valor3 "value" from soat.\_tariff\_ t where t.type not like \_\_type);

end if;

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_tariff\_as\_medicine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_tariff\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_tariff\_as\_medicine

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure get\_user

#### Card of procedure get\_user

|  |  |
| --- | --- |
| Name | get\_user |
| Code | get\_user |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_user

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_filter json,

\_pattern text,

\_limit int4,

\_offset int4

)

RETURNS table(

count bigint,

identifier int4,

login varchar(64),

name varchar(128),

profile\_name varchar(64),

department\_name varchar(150),

email varchar(64),

state int2

) AS $$

DECLARE

BEGIN

\_pattern := '%' || \_pattern || '%';

RETURN QUERY(

select

count(\*) over(),

u.identifier,

u.login,

(u.name || ' ' || u.last\_name)::varchar(128),

coalesce(p.name, 'Ninguno'),

coalesce(d.name, 'Ninguno'),

u.email,

u.state

from

kernel.\_user\_ u

left join kernel.\_department\_ d on d.identifier = u.department\_identifier and d.state = 0

left join kernel.\_profile\_ p on p.identifier = u.profile\_identifier and p.state = 0

where

u.state::varchar(1) like '%' || (\_filter->>'state')::varchar(1) || '%' and

(

u.login ilike \_pattern or

u.name || ' ' || u.last\_name ilike \_pattern or

coalesce(p.name, 'Ninguno') ilike \_pattern or

coalesce(d.name, 'Ninguno') ilike \_pattern or

u.email ilike \_pattern

)

order by u.name, u.last\_name asc

offset nullif(\_offset, 0)

limit nullif(\_limit, 0)

);

EXCEPTION

WHEN OTHERS THEN NULL;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_department\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_user\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_user

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_user\_by\_id

#### Card of procedure get\_user\_by\_id

|  |  |
| --- | --- |
| Name | get\_user\_by\_id |
| Code | get\_user\_by\_id |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_user\_by\_id

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_app\_identifier int4

)

RETURNS SETOF kernel.\_user\_ AS $$

select

u.\*

from kernel.\_user\_ u

inner join kernel.\_profile\_ p on p.identifier = u.profile\_identifier

inner join kernel.\_app\_ a on a.identifier = p.app\_identifier and a.identifier = \_app\_identifier

where

u.identifier = \_identifier;

$$

LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure get\_user\_by\_id

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_user\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure get\_user\_by\_id

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure get\_vehicle

#### Card of procedure get\_vehicle

|  |  |
| --- | --- |
| Name | get\_vehicle |
| Code | get\_vehicle |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure get\_vehicle

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_filter text,

\_state char(1),

\_limit integer,

\_trashed char(1)

)

RETURNS SETOF vehicle.\_vehicle\_ AS $$

SELECT

v.identifier,

v.code,

v.type\_identifier,

v.year,

v.plate,

v.brand\_identifier,

v.model,

v.color\_identifier,

v.engine,

v.chassis,

v.reference\_value,

v.appraised\_value,

v.state,

v.description,

v.active,

v.trashed,

v.author,

v.creation\_date,

v.last\_updater,

v.last\_update\_date,

v.dustman,

v.trash\_date

FROM

vehicle.\_vehicle\_ v

inner join kernel.\_catalog\_ t on v.type\_identifier = t.identifier

where

v.trashed::varchar(1) like '%' || \_trashed || '%' and

v.active::varchar(1) like '%' || \_state || '%' and

(

t.name ilike '%' || \_filter || '%' or

v.code ilike '%' || \_filter || '%' or

v.year ilike '%' || \_filter || '%' or

v.plate ilike '%' || \_filter || '%' or

v.model ilike '%' || \_filter || '%' or

v.engine ilike '%' || \_filter || '%'

)

order by v.code asc

limit

case \_limit

when 0 then (select count(\*) from vehicle.\_vehicle\_)

else \_limit

end;

$$ LANGUAGE '%ProcLanguage%';

### Procedure insert\_action

#### Card of procedure insert\_action

|  |  |
| --- | --- |
| Name | insert\_action |
| Code | insert\_action |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure insert\_action

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4,

in \_root\_identifier varchar(64),

in \_type int2,

in \_handler varchar(64),

in \_action varchar(64),

in \_access int2,

in \_name varchar(32),

in \_title varchar(64),

in \_show\_in\_menu int2,

in \_icon varchar(32),

in \_icon\_color varchar(32),

in \_confirm int2,

in \_run\_mode int2,

in \_template text,

in \_context\_states varchar(64),

in \_controller\_name varchar(64),

in \_properties text,

in \_description text,

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

\_\_identifier\_seq bigint;

\_\_identifier varchar(64);

BEGIN

--

--normalizacion--

--

if(\_type = 0) then --etiqueta

\_\_identifier\_seq := nextval('kernel.action\_identifier\_sequence');

\_\_identifier := md5(\_\_identifier\_seq::text) || md5(\_\_identifier\_seq::text);

\_handler := NULL;

\_action := NULL;

\_access := 0;

--\_title := NULL;

\_show\_in\_menu := 1;

\_confirm := NULL;

\_run\_mode := NULL;

\_template := NULL;

\_context\_states := NULL;

\_controller\_name := NULL;

\_properties := '{}';

else --accion o dep

\_\_identifier := md5(\_handler) || md5(\_action);

if exists (select 1 from kernel.\_action\_ where identifier = \_\_identifier) then

raise exception 'La combinación de "manejador" y "acción" ya está asociada a otra acción!!';

end if;

if(\_run\_mode != 1) then

\_properties := '{}';

end if;

if(\_run\_mode = 2) then

\_template := NULL;

\_controller\_name := NULL;

end if;

end if;

insert into kernel.\_action\_(

identifier,

root\_identifier,

app\_identifier,

type,

handler,

action,

access,

name,

title,

show\_in\_menu,

icon,

icon\_color,

confirm,

run\_mode,

template,

context\_states,

controller\_name,

properties,

description,

state,

last\_updater

)values(

\_\_identifier,

NULLIF(\_root\_identifier, ''),

\_app\_identifier,

\_type,

\_handler,

\_action,

\_access,

\_name,

\_title,

\_show\_in\_menu,

\_icon,

\_icon\_color,

\_confirm,

\_run\_mode,

\_template,

\_context\_states,

\_controller\_name,

\_properties,

\_description,

\_state,

\_last\_updater

);

RAISE NOTICE 'Accion "%" ingresada!!', \_name;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure insert\_action

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_action\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure insert\_action

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure insert\_app

#### Card of procedure insert\_app

|  |  |
| --- | --- |
| Name | insert\_app |
| Code | insert\_app |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure insert\_app

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_name varchar(64),

in \_title varchar(64),

in \_description text,

in \_active int2

) RETURNS void AS

$BODY$

insert into kernel.\_app\_(

code,

name,

title,

description,

active

)values(

'APP-APP-' || date\_part('year', now()) || '-' || '#####',

\_name,

\_title,

\_description,

\_active

);

$BODY$

LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure insert\_app

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_app\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure insert\_app

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure insert\_product

#### Card of procedure insert\_product

|  |  |
| --- | --- |
| Name | insert\_product |
| Code | insert\_product |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure insert\_product

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4,

in \_name varchar(64),

in \_price decimal(19, 5),

in \_description text,

in \_active int2,

in \_author varchar(64)

)

RETURNS void AS $$

insert into [%OWNER%.]\_product\_(

app\_identifier,

name,

code,

price,

description,

active,

author

)values(

\_app\_identifier,

\_name,

'FACIG-PROD-' || date\_part('year', now()) || '-' || lpad(cast((select (count(\*) + 1) from product.\_product\_) as text), 5, '0'),

\_price,

\_description,

\_active,

\_author

);

$$ LANGUAGE '%ProcLanguage%';

### Procedure insert\_profile

#### Card of procedure insert\_profile

|  |  |
| --- | --- |
| Name | insert\_profile |
| Code | insert\_profile |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure insert\_profile

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4,

in \_name varchar(64),

in \_description text,

in \_state int2,

in \_last\_updater int4,

in \_actions varchar[]

)

RETURNS BOOLEAN AS $$

DECLARE

\_\_identifier int4;

BEGIN

if exists (select 1 from kernel.\_profile\_ where name = \_name) then

raise exception 'Ya existe un perfíl con el mismo nombre';

end if;

insert into kernel.\_profile\_(

app\_identifier,

name,

description,

state,

last\_updater

)values(

\_app\_identifier,

\_name,

\_description,

\_state,

\_last\_updater

) RETURNING identifier into \_\_identifier;

if not kernel.insert\_profile\_has\_action(\_\_identifier, \_actions) then

raise exception 'No se pudieron insertar las acciones del perfíl';

end if;

RAISE NOTICE 'Perfíl "%" ingresado!!', \_name;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure insert\_profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure insert\_profile

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure insert\_profile\_has\_action

#### Card of procedure insert\_profile\_has\_action

|  |  |
| --- | --- |
| Name | insert\_profile\_has\_action |
| Code | insert\_profile\_has\_action |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure insert\_profile\_has\_action

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_profile\_identifier int4,

in \_actions varchar[]

)

RETURNS BOOLEAN AS $$

DECLARE

\_\_action\_identifier varchar(64);

BEGIN

delete from kernel.\_profile\_has\_action\_ where profile\_identifier = \_profile\_identifier;

foreach \_\_action\_identifier in array \_actions

loop

if not kernel.insert\_profile\_has\_action\_node(\_profile\_identifier, \_\_action\_identifier) then

return false;

end if;

end loop;

return true;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure insert\_profile\_has\_action

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_profile\_has\_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure insert\_profile\_has\_action

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure insert\_profile\_has\_action\_node

#### Card of procedure insert\_profile\_has\_action\_node

|  |  |
| --- | --- |
| Name | insert\_profile\_has\_action\_node |
| Code | insert\_profile\_has\_action\_node |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure insert\_profile\_has\_action\_node

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_profile\_identifier int4,

in \_action\_identifier varchar(64)

)

RETURNS BOOLEAN AS $$

DECLARE

\_\_dep\_actions varchar[];

\_\_root varchar(64);

BEGIN

--validate if action exists

if not exists (select 1 from kernel.\_action\_ where identifier = \_action\_identifier) then

raise exception 'La accion id = %, no existe en la tabla de acciones..!!', \_action\_identifier;

end if;

--insert current node

IF NOT EXISTS (SELECT 1 FROM kernel.\_profile\_has\_action\_ WHERE profile\_identifier = \_profile\_identifier and action\_identifier = \_action\_identifier) THEN

INSERT INTO kernel.\_profile\_has\_action\_(profile\_identifier, action\_identifier) VALUES (\_profile\_identifier, \_action\_identifier);

END IF;

--insert current node deps

insert into kernel.\_profile\_has\_action\_(

profile\_identifier,

action\_identifier

) select

\_profile\_identifier,

a.identifier

from

kernel.\_action\_ a

where

a.root\_identifier = \_action\_identifier and

a.state = 0 and

a.type = 2 and

a.access = 0 and

a.identifier not in ( select pha.action\_identifier from kernel.\_profile\_has\_action\_ pha where pha.profile\_identifier = \_profile\_identifier);

--if has parent, return insert parent action

\_\_root := (select root\_identifier from kernel.\_action\_ WHERE identifier = \_action\_identifier);

if(\_\_root is not null) then

return kernel.insert\_profile\_has\_action\_node(\_profile\_identifier, \_\_root);

end if;

RETURN TRUE;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure insert\_profile\_has\_action\_node

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_has\_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure insert\_profile\_has\_action\_node

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure login\_user

#### Card of procedure login\_user

|  |  |
| --- | --- |
| Name | login\_user |
| Code | login\_user |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure login\_user

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_login varchar(32),

in \_password varchar(32),

in \_app\_identifier int4

)

RETURNS SETOF kernel.\_user\_ AS

$BODY$

select

u.\*

from kernel.\_user\_ u

inner join kernel.\_profile\_ p on p.identifier = u.profile\_identifier

inner join kernel.\_app\_ a on a.identifier = p.app\_identifier

where

u.state != 2 and

md5(u.login) = \_login and

u.password = \_password and

a.identifier = \_app\_identifier;

$BODY$

LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure login\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_user\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure login\_user

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure process\_ambulance\_use

#### Card of procedure process\_ambulance\_use

|  |  |
| --- | --- |
| Name | process\_ambulance\_use |
| Code | process\_ambulance\_use |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_ambulance\_use

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_form008\_identifier int4,

in \_\_autorization\_number varchar(64),

in \_\_city varchar(64),

in \_\_km\_out decimal(19, 5),

in \_\_km\_in decimal(19, 5),

in \_\_km\_diff decimal(19, 5),

in \_\_starting\_point\_code numeric,

in \_\_starting\_point\_description varchar(360),

in \_\_starting\_point\_uvr decimal(19, 5),

in \_\_starting\_point\_fcm decimal(19, 5),

in \_\_starting\_point\_value decimal(19, 5),

in \_\_ambulance\_km\_code numeric,

in \_\_ambulance\_km\_description varchar(360),

in \_\_ambulance\_km\_uvr decimal(19, 5),

in \_\_ambulance\_km\_fcm decimal(19, 5),

in \_\_ambulance\_km\_value decimal(19, 5),

in \_\_ambulance\_km\_subtotal decimal(19, 5),

in \_\_total decimal(19, 5),

in \_\_description text,

in \_\_last\_updater int4

)

RETURNS boolean AS $$

DECLARE

BEGIN

insert into soat.\_ambulance\_use\_(

form008\_identifier,

autorization\_number,

city,

km\_out,

km\_in,

km\_diff,

starting\_point\_code,

starting\_point\_description,

starting\_point\_uvr,

starting\_point\_fcm,

starting\_point\_value,

ambulance\_km\_code,

ambulance\_km\_description,

ambulance\_km\_uvr,

ambulance\_km\_fcm,

ambulance\_km\_value,

ambulance\_km\_subtotal,

total,

description,

last\_updater

)values(

\_\_form008\_identifier,

\_\_autorization\_number,

\_\_city,

\_\_km\_out,

\_\_km\_in,

\_\_km\_diff,

\_\_starting\_point\_code,

\_\_starting\_point\_description,

\_\_starting\_point\_uvr,

\_\_starting\_point\_fcm,

\_\_starting\_point\_value,

\_\_ambulance\_km\_code,

\_\_ambulance\_km\_description,

\_\_ambulance\_km\_uvr,

\_\_ambulance\_km\_fcm,

\_\_ambulance\_km\_value,

\_\_ambulance\_km\_subtotal,

\_\_total,

\_\_description,

\_\_last\_updater

);

RAISE NOTICE 'Registro ingresado';

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_ambulance\_use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_ambulance\_use\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure process\_ambulance\_use

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_catalog

#### Card of procedure process\_catalog

|  |  |
| --- | --- |
| Name | process\_catalog |
| Code | process\_catalog |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_catalog

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_app\_identifier int4,

in \_type varchar(32),

in \_name varchar(64),

in \_value text,

in \_description text,

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

BEGIN

if exists (select 1 from kernel.\_catalog\_ where identifier = \_identifier) then

update kernel.\_catalog\_ set

type = \_type,

name = \_name,

value = \_value,

description = \_description,

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

raise notice 'Catalogo actualizado!!';

else

insert into kernel.\_catalog\_(

app\_identifier,

type,

name,

value,

description,

state,

last\_updater

)values(

\_app\_identifier,

\_type,

\_name,

\_value,

\_description,

\_state,

\_last\_updater

);

raise notice 'Catalogo ingresado!!';

end if;

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_catalog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_catalog\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_catalog

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure process\_department

#### Card of procedure process\_department

|  |  |
| --- | --- |
| Name | process\_department |
| Code | process\_department |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_department

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_enterprise\_identifier int4,

in \_coordination\_identifier int4,

in \_name varchar(64),

in \_description text,

in \_state int2,

in \_last\_updater int4

)

RETURNS BOOLEAN AS $$

DECLARE

BEGIN

if exists (select 1 from kernel.\_department\_ where identifier = \_identifier) then

update

kernel.\_department\_

set

coordination\_identifier = nullif(\_coordination\_identifier, 0),

name = \_name,

description = \_description,

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

RAISE NOTICE 'Departamento ingresado!!';

else

insert into kernel.\_department\_(

enterprise\_identifier,

coordination\_identifier,

name,

code,

description,

state,

last\_updater

)values(

\_enterprise\_identifier,

nullif(\_coordination\_identifier, 0),

\_name,

'TMP',

\_description,

\_state,

\_last\_updater

) returning identifier into \_identifier;

update kernel.\_department\_ set code = 'DEPTO-' || date\_part('year', now()) || '-' || lpad(\_identifier::text, 5, '0') where identifier = \_identifier;

RAISE NOTICE 'Departamento ingresado!!';

end if;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_department

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_department\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_department

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure process\_enterprise

#### Card of procedure process\_enterprise

|  |  |
| --- | --- |
| Name | process\_enterprise |
| Code | process\_enterprise |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_enterprise

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_app\_identifier int4,

in \_identity\_number varchar(13),

in \_name varchar(64),

in \_business\_name varchar(64),

in \_law\_representative varchar(64),

in \_address varchar(64),

in \_phone varchar(64),

in \_email varchar(64),

in \_description text,

in \_state int2,

in \_last\_updater int4

)

RETURNS BOOLEAN AS $$

DECLARE

BEGIN

if not exists (select 1 from kernel.\_enterprise\_ where identifier = \_identifier) then

insert into kernel.\_enterprise\_(

app\_identifier,

identity\_number,

name,

business\_name,

law\_representative,

address,

phone,

email,

description,

state,

last\_updater

)values(

\_app\_identifier,

\_identity\_number,

\_name,

\_business\_name,

\_law\_representative,

\_address,

\_phone,

\_email,

\_description,

\_state,

\_last\_updater

);

else

update

kernel.\_enterprise\_

set

identity\_number = \_identity\_number,

name = \_name,

business\_name = \_business\_name,

law\_representative = \_law\_representative,

address = \_address,

phone = \_phone,

email = \_email,

description = \_description,

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

end if;

RAISE NOTICE 'Datos actualizados!!';

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_enterprise

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_enterprise\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_enterprise\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure process\_enterprise

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure process\_expense\_form

#### Card of procedure process\_expense\_form

|  |  |
| --- | --- |
| Name | process\_expense\_form |
| Code | process\_expense\_form |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_expense\_form

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_form008\_identifier int4,

in \_diagnostic text,

in \_total decimal(19,5),

in \_sppat decimal(19,5),

in \_balance decimal(19,5),

in \_description text,

in \_last\_updater int4,

in \_tariff\_detail json[],

in \_medicine\_detail json[]

)

RETURNS boolean AS $$

DECLARE

\_\_expenseFormIdentifier int4;

\_\_tmpJson json;

BEGIN

insert into soat.\_expense\_form\_(

form008\_identifier,

diagnostic,

total,

sppat,

balance,

description,

last\_updater

)values(

\_form008\_identifier,

\_diagnostic,

\_total,

\_sppat,

\_balance,

\_description,

\_last\_updater

) returning identifier into \_\_expenseFormIdentifier;

FOREACH \_\_tmpJson SLICE 0 IN ARRAY \_tariff\_detail

LOOP

insert into soat.\_expense\_form\_tariff\_detail\_(

expense\_form\_identifier,

item\_identifier,

item\_code,

item\_description,

item\_type,

quantity,

unit\_value,

total\_value

)values(

\_\_expenseFormIdentifier,

(\_\_tmpJson->>'item\_identifier')::numeric,

(\_\_tmpJson->>'item\_code')::varchar(64),

(\_\_tmpJson->>'item\_description')::varchar(360),

(\_\_tmpJson->>'item\_type')::varchar(360),

(\_\_tmpJson->>'quantity')::decimal(19,5),

(\_\_tmpJson->>'unit\_value')::decimal(19,5),

(\_\_tmpJson->>'total\_value')::decimal(19,5)

);

END LOOP;

FOREACH \_\_tmpJson SLICE 0 IN ARRAY \_medicine\_detail

LOOP

insert into soat.\_expense\_form\_medicine\_detail\_(

expense\_form\_identifier,

item\_identifier,

item\_code,

item\_description,

item\_type,

quantity,

unit\_value,

total\_value

)values(

\_\_expenseFormIdentifier,

(\_\_tmpJson->>'item\_identifier')::int4,

(\_\_tmpJson->>'item\_code')::varchar(64),

(\_\_tmpJson->>'item\_description')::varchar(360),

(\_\_tmpJson->>'item\_type')::varchar(360),

(\_\_tmpJson->>'quantity')::decimal(19,5),

(\_\_tmpJson->>'unit\_value')::decimal(19,5),

(\_\_tmpJson->>'total\_value')::decimal(19,5)

);

END LOOP;

RAISE NOTICE 'Planilla de gastos generada con éxito';

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_expense\_form

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_expense\_form\_ | Table | soat[pg] | soat[pg] |  |
| \_expense\_form\_medicine\_detail\_ | Table | soat[pg] | soat[pg] |  |
| \_expense\_form\_tariff\_detail\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure process\_expense\_form

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_form008

#### Card of procedure process\_form008

|  |  |
| --- | --- |
| Name | process\_form008 |
| Code | process\_form008 |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_form008

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_patient\_identifier varchar(64),

in \_last\_updater INT4

)

RETURNS int4 AS $$

DECLARE

\_\_identifier int4;

\_\_code int4;

BEGIN

\_\_code := (select count(\*) + 1 from soat.\_form\_008\_ where date\_trunc('year', now()) = date\_trunc('year', last\_update\_date));

insert into soat.\_form\_008\_(

patient\_identifier,

code,

last\_updater

) values(

\_patient\_identifier,

date\_part('year', now()) || '-' || lpad(\_\_code, 3, '0'),

last\_updater

) returning identifier into \_\_identifier;

return \_\_identifier;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN 0;

end;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_form008

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_form\_008\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_form008

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_form008\_has\_doctor

#### Card of procedure process\_form008\_has\_doctor

|  |  |
| --- | --- |
| Name | process\_form008\_has\_doctor |
| Code | process\_form008\_has\_doctor |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_form008\_has\_doctor

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_form008\_identifier int4,

in \_\_doctor\_identifier int4,

in \_\_last\_updater int4

)

RETURNS boolean AS $$

DECLARE

BEGIN

if not exists(select 1 from soat.\_form008\_has\_doctor\_ where form008\_identifier = \_\_form008\_identifier ) then

insert into soat.\_form008\_has\_doctor\_(

form008\_identifier,

doctor\_identifier,

last\_updater

)values(

\_\_form008\_identifier,

\_\_doctor\_identifier,

\_\_last\_updater

);

else

update

soat.\_form008\_has\_doctor\_

set

doctor\_identifier = \_\_doctor\_identifier,

last\_updater = \_\_last\_updater,

last\_update\_date = now()

where

form008\_identifier = \_\_form008\_identifier;

end if;

RAISE NOTICE 'Medico asignado a formulario';

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_form008\_has\_doctor

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_form008\_has\_doctor\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_form008\_has\_doctor

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_fur

#### Card of procedure process\_fur

|  |  |
| --- | --- |
| Name | process\_fur |
| Code | process\_fur |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_fur

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

\_\_form008\_identifier int4,

\_\_patient\_condition\_walker int2,

\_\_patient\_condition\_occupant int2,

\_\_accident\_address text,

\_\_accident\_province varchar(64),

\_\_accident\_canton varchar(64),

\_\_accident\_parish varchar(64),

\_\_accident\_neighborhood varchar(64),

\_\_accident\_date date,

\_\_accident\_hour int4,

\_\_accident\_minute int4,

\_\_accident\_report text,

\_\_vehicle\_insured int2,

\_\_vehicle\_not\_insured int2,

\_\_vehicle\_not\_identified int2,

\_\_vehicle\_coverage\_code varchar(64),

\_\_vehicle\_license\_plate varchar(64),

\_\_vehicle\_soat\_number varchar(64),

\_\_vehicle\_insurance\_carrier varchar(64),

\_\_vehicle\_police\_start\_date date,

\_\_vehicle\_police\_end\_date date,

\_\_health\_center\_admision\_date date,

\_\_health\_center\_admision\_hour int4,

\_\_health\_center\_admision\_minute int4,

\_\_health\_center\_history\_number varchar(64),

\_\_health\_center\_egress\_date date,

\_\_health\_center\_ambulatory\_care int2,

\_\_health\_center\_hosp\_care int2,

\_\_health\_center\_hosp\_care\_days int4,

\_\_health\_center\_admision\_report text,

\_\_health\_center\_egress\_report text,

\_\_health\_center\_referred\_fr varchar(64),

\_\_health\_center\_referred\_fr\_city varchar(64),

\_\_health\_center\_referred\_fr\_date date,

\_\_health\_center\_referred\_to varchar(64),

\_\_health\_center\_referred\_to\_city varchar(64),

\_\_health\_center\_referred\_to\_date date,

\_\_insured\_dead int2,

\_\_insured\_dead\_basic\_reazon text,

\_\_insured\_dead\_direct\_reazon text,

\_\_insured\_dead\_date date,

\_\_insured\_dead\_hour int4,

\_\_insured\_dead\_minute int4,

\_\_insured\_dead\_in\_charge\_doctor int4,

\_\_insured\_dead\_authorizer\_doctor int4,

\_\_last\_updater int4

)

RETURNS boolean AS $$

DECLARE

BEGIN

if exists (select 1 from soat.\_fur\_ where form008\_identifier = \_\_form008\_identifier) then

RAISE NOTICE 'No se puede ingresar el FUR, ya existe un formulario vinculado';

return false;

else

INSERT INTO soat.\_fur\_(

form008\_identifier,

patient\_condition\_walker,

patient\_condition\_occupant,

accident\_address,

accident\_province,

accident\_canton,

accident\_parish,

accident\_neighborhood,

accident\_date,

accident\_hour,

accident\_minute,

accident\_report,

vehicle\_insured,

vehicle\_not\_insured,

vehicle\_not\_identified,

vehicle\_coverage\_code,

vehicle\_license\_plate,

vehicle\_soat\_number,

vehicle\_insurance\_carrier,

vehicle\_police\_start\_date,

vehicle\_police\_end\_date,

health\_center\_admision\_date,

health\_center\_admision\_hour,

health\_center\_admision\_minute,

health\_center\_history\_number,

health\_center\_egress\_date,

health\_center\_ambulatory\_care,

health\_center\_hosp\_care,

health\_center\_hosp\_care\_days,

health\_center\_admision\_report,

health\_center\_egress\_report,

health\_center\_referred\_fr,

health\_center\_referred\_fr\_city,

health\_center\_referred\_fr\_date,

health\_center\_referred\_to,

health\_center\_referred\_to\_city,

health\_center\_referred\_to\_date,

insured\_dead,

insured\_dead\_basic\_reazon,

insured\_dead\_direct\_reazon,

insured\_dead\_date,

insured\_dead\_hour,

insured\_dead\_minute,

insured\_dead\_in\_charge\_doctor,

insured\_dead\_authorizer\_doctor,

last\_updater

)VALUES(

\_\_form008\_identifier,

\_\_patient\_condition\_walker,

\_\_patient\_condition\_occupant,

\_\_accident\_address,

\_\_accident\_province,

\_\_accident\_canton,

\_\_accident\_parish,

\_\_accident\_neighborhood,

\_\_accident\_date,

\_\_accident\_hour,

\_\_accident\_minute,

\_\_accident\_report,

\_\_vehicle\_insured,

\_\_vehicle\_not\_insured,

\_\_vehicle\_not\_identified,

\_\_vehicle\_coverage\_code,

\_\_vehicle\_license\_plate,

\_\_vehicle\_soat\_number,

\_\_vehicle\_insurance\_carrier,

\_\_vehicle\_police\_start\_date,

\_\_vehicle\_police\_end\_date,

\_\_health\_center\_admision\_date,

\_\_health\_center\_admision\_hour,

\_\_health\_center\_admision\_minute,

\_\_health\_center\_history\_number,

\_\_health\_center\_egress\_date,

\_\_health\_center\_ambulatory\_care,

\_\_health\_center\_hosp\_care,

\_\_health\_center\_hosp\_care\_days,

\_\_health\_center\_admision\_report,

\_\_health\_center\_egress\_report,

\_\_health\_center\_referred\_fr,

\_\_health\_center\_referred\_fr\_city,

\_\_health\_center\_referred\_fr\_date,

\_\_health\_center\_referred\_to,

\_\_health\_center\_referred\_to\_city,

\_\_health\_center\_referred\_to\_date,

\_\_insured\_dead,

\_\_insured\_dead\_basic\_reazon,

\_\_insured\_dead\_direct\_reazon,

\_\_insured\_dead\_date,

\_\_insured\_dead\_hour,

\_\_insured\_dead\_minute,

\_\_insured\_dead\_in\_charge\_doctor,

\_\_insured\_dead\_authorizer\_doctor,

\_\_last\_updater

);

RAISE NOTICE 'Formulario actualizado';

return true;

end if;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_fur

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_fur\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_fur

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_medicine

#### Card of procedure process\_medicine

|  |  |
| --- | --- |
| Name | process\_medicine |
| Code | process\_medicine |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_medicine

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_name varchar(360),

in \_value decimal(19,5),

in \_stock decimal(19,5),

in \_last\_updater int4

)

RETURNS boolean AS $$

DECLARE

\_\_code varchar(64);

\_\_identifier int4;

BEGIN

if not exists (select 1 from soat.\_medicine\_ where identifier = \_identifier) then

insert into soat.\_medicine\_(

code,

name,

type,

value,

stock,

last\_updater

)values(

'NULL',

\_name,

'FARMACIA',

\_value,

\_stock,

\_last\_updater

) returning identifier into \_\_identifier;

\_\_code := 'MED-' || LPAD(\_\_identifier::text, 5, '0')::VARCHAR(60);

update soat.\_medicine\_ set code = \_\_code where identifier = \_\_identifier;

end if;

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_medicine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_medicine\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_medicine

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_objection

#### Card of procedure process\_objection

|  |  |
| --- | --- |
| Name | process\_objection |
| Code | process\_objection |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_objection

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_form008\_identifier int4,

in \_number varchar(64),

in \_value decimal(19,5),

in \_invoice varchar(64),

in \_description text,

in \_last\_updater int4

)

RETURNS boolean AS $$

DECLARE

BEGIN

if not exists(select 1 from soat.\_objection\_ where form008\_identifier = \_form008\_identifier) then

insert into soat.\_objection\_(

form008\_identifier,

number,

value,

invoice,

description,

last\_updater

)values(

\_form008\_identifier,

\_number,

\_value,

\_invoice,

\_description,

\_last\_updater

);

RAISE NOTICE 'Objeción registrada...';

else

update soat.\_objection\_ set

number = \_number,

value = \_value,

invoice = \_invoice,

description = \_description,

last\_update\_date = now(),

last\_updater = \_last\_updater

where

form008\_identifier = \_form008\_identifier;

RAISE NOTICE 'Objeción actualizada...';

end if;

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_objection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_objection\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_objection

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_oxygen\_use

#### Card of procedure process\_oxygen\_use

|  |  |
| --- | --- |
| Name | process\_oxygen\_use |
| Code | process\_oxygen\_use |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_oxygen\_use

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_\_form008\_identifier int4,

in \_\_start\_date date,

in \_\_start\_hour int,

in \_\_start\_minute int,

in \_\_end\_date date,

in \_\_end\_hour int,

in \_\_end\_minute int,

in \_\_diff\_hour int,

in \_\_diff\_minute int,

in \_\_total\_minute int,

in \_\_lm decimal(19, 5),

in \_\_oxygen\_code numeric,

in \_\_oxygen\_description varchar(360),

in \_\_uvr decimal(19, 5),

in \_\_fcm decimal(19, 5),

in \_\_value decimal(19, 5),

in \_\_total decimal(19, 5),

in \_\_description text,

in \_\_last\_updater int4

)

RETURNS boolean AS $$

DECLARE

BEGIN

insert into soat.\_oxygen\_use\_(

form008\_identifier,

start\_date,

start\_hour,

start\_minute,

end\_date,

end\_hour,

end\_minute,

diff\_hour,

diff\_minute,

total\_minute,

lm,

oxygen\_code,

oxygen\_description,

uvr,

fcm,

value,

total,

description,

last\_updater

)values(

\_\_form008\_identifier,

\_\_start\_date,

\_\_start\_hour,

\_\_start\_minute,

\_\_end\_date,

\_\_end\_hour,

\_\_end\_minute,

\_\_diff\_hour,

\_\_diff\_minute,

\_\_total\_minute,

\_\_lm,

\_\_oxygen\_code,

\_\_oxygen\_description,

\_\_uvr,

\_\_fcm,

\_\_value,

\_\_total,

\_\_description,

\_\_last\_updater

);

RAISE NOTICE 'Registro ingresado';

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_oxygen\_use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_oxygen\_use\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure process\_oxygen\_use

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_patient

#### Card of procedure process\_patient

|  |  |
| --- | --- |
| Name | process\_patient |
| Code | process\_patient |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_patient

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identificacion VARCHAR(64),

in \_primer\_nombre VARCHAR(64),

in \_segundo\_nombre VARCHAR(64),

in \_apellido\_materno VARCHAR(64),

in \_apellido\_paterno VARCHAR(64),

in \_direccion VARCHAR(128),

in \_barrio VARCHAR(64),

in \_parroquia VARCHAR(64),

in \_canton VARCHAR(64),

in \_provincia VARCHAR(64),

in \_zona VARCHAR(64),

in \_telefono VARCHAR(64),

in \_fecha\_nacimiento date,

in \_lugar\_nacimiento VARCHAR(64),

in \_nacionalidad VARCHAR(64),

in \_grupo\_cultural VARCHAR(64),

in \_edad INT2,

in \_genero CHAR(1),

in \_estado\_civil VARCHAR(64),

in \_instruccion VARCHAR(64),

in \_fecha\_admision DATE,

in \_ocupacion VARCHAR(64),

in \_empresa\_trabajo VARCHAR(64),

in \_tipo\_seguro\_salud VARCHAR(64),

in \_referido\_de VARCHAR(64),

in \_contacto\_emergencia VARCHAR(64),

in \_parentezco\_contacto\_emergencia VARCHAR(64),

in \_direccion\_contacto\_emergencia VARCHAR(128),

in \_telefono\_contacto\_emergencia VARCHAR(64),

in \_description TEXT,

in \_last\_updater INT4

)

RETURNS boolean AS $$

DECLARE

BEGIN

if not exists(select 1 from soat.\_patient\_ where identificacion = \_identificacion ) then

INSERT INTO soat.\_patient\_(

identificacion,

primer\_nombre,

segundo\_nombre,

apellido\_materno,

apellido\_paterno,

direccion,

barrio,

parroquia,

canton,

provincia,

zona,

telefono,

fecha\_nacimiento,

lugar\_nacimiento,

nacionalidad,

grupo\_cultural,

edad,

genero,

estado\_civil,

instruccion,

fecha\_admision,

ocupacion,

empresa\_trabajo,

tipo\_seguro\_salud,

referido\_de,

contacto\_emergencia,

parentezco\_contacto\_emergencia,

direccion\_contacto\_emergencia,

telefono\_contacto\_emergencia,

description,

last\_updater

)VALUES (

\_identificacion,

\_primer\_nombre,

\_segundo\_nombre,

\_apellido\_materno,

\_apellido\_paterno,

\_direccion,

\_barrio,

\_parroquia,

\_canton,

\_provincia,

\_zona,

\_telefono,

\_fecha\_nacimiento,

\_lugar\_nacimiento,

\_nacionalidad,

\_grupo\_cultural,

\_edad,

\_genero,

\_estado\_civil,

\_instruccion,

\_fecha\_admision,

\_ocupacion,

\_empresa\_trabajo,

\_tipo\_seguro\_salud,

\_referido\_de,

\_contacto\_emergencia,

\_parentezco\_contacto\_emergencia,

\_direccion\_contacto\_emergencia,

\_telefono\_contacto\_emergencia,

\_description,

\_last\_updater

);

RAISE NOTICE 'Paciente ingresado';

else

update soat.\_patient\_ set

primer\_nombre = \_primer\_nombre,

segundo\_nombre = \_segundo\_nombre,

apellido\_materno = \_apellido\_materno,

apellido\_paterno = \_apellido\_paterno,

direccion = \_direccion,

barrio = \_barrio,

parroquia = \_parroquia,

canton = \_canton,

provincia = \_provincia,

zona = \_zona,

telefono = \_telefono,

fecha\_nacimiento = \_fecha\_nacimiento,

lugar\_nacimiento = \_lugar\_nacimiento,

nacionalidad = \_nacionalidad,

grupo\_cultural = \_grupo\_cultural,

edad = \_edad,

genero = \_genero,

estado\_civil = \_estado\_civil,

instruccion = \_instruccion,

fecha\_admision = \_fecha\_admision,

ocupacion = \_ocupacion,

empresa\_trabajo = \_empresa\_trabajo,

tipo\_seguro\_salud = \_tipo\_seguro\_salud,

referido\_de = \_referido\_de,

contacto\_emergencia = \_contacto\_emergencia,

parentezco\_contacto\_emergencia = \_parentezco\_contacto\_emergencia,

direccion\_contacto\_emergencia = \_direccion\_contacto\_emergencia,

telefono\_contacto\_emergencia = \_telefono\_contacto\_emergencia,

description = \_description,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identificacion = \_identificacion;

RAISE NOTICE 'Paciente actualizado';

end if;

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

end;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_patient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_patient\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure process\_patient

|  |  |
| --- | --- |
| Name | Code |
| app.soat | app\_soat |

### Procedure process\_user

#### Card of procedure process\_user

|  |  |
| --- | --- |
| Name | process\_user |
| Code | process\_user |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure process\_user

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_login varchar(64),

in \_profile\_identifier int4,

in \_department\_identifier int4,

in \_name varchar(64),

in \_last\_name varchar(64),

in \_email varchar(64),

in \_password varchar(32),

in \_description text,

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

BEGIN

\_department\_identifier := nullif(\_department\_identifier, 0);

if exists (select 1 from kernel.\_user\_ where identifier = \_identifier) then

if exists (select 1 from kernel.\_user\_ where login = \_login and identifier != \_identifier) then

raise notice 'El login % ya esta asignado a otro usuario!!', \_login;

return false;

end if;

if exists (select 1 from kernel.\_user\_ where email = \_email and identifier != \_identifier) then

raise notice 'El email % ya esta asignado a otro usuario!!', \_login;

return false;

end if;

update kernel.\_user\_ set

login = \_login,

profile\_identifier = \_profile\_identifier,

department\_identifier = \_department\_identifier,

name = \_name,

last\_name = \_last\_name,

email = \_email,

description = \_description,

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

if (char\_length(\_password) = 32) then

update kernel.\_user\_ set

password = \_password

where identifier = \_identifier;

end if;

raise notice 'Usuario actualizado!!';

else

if exists (select 1 from kernel.\_user\_ where login = \_login) then

raise notice 'El login % ya esta asignado a otro usuario!!', \_login;

return false;

end if;

if exists (select 1 from kernel.\_user\_ where email = \_email) then

raise notice 'El email % ya esta asignado a otro usuario!!', \_login;

return false;

end if;

insert into kernel.\_user\_(

login,

profile\_identifier,

department\_identifier,

name,

last\_name,

email,

password,

description,

state,

last\_updater

)values(

\_login,

\_profile\_identifier,

\_department\_identifier,

\_name,

\_last\_name,

\_email,

\_password,

\_description,

\_state,

\_last\_updater

);

raise notice 'Usuario ingresado!!';

end if;

return true;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure process\_user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_user\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_user\_ | Table | soat[pg] | soat[pg] |  |

#### List of diagrams containing the procedure process\_user

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure update\_action

#### Card of procedure update\_action

|  |  |
| --- | --- |
| Name | update\_action |
| Code | update\_action |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure update\_action

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_app\_identifier int4,

in \_identifier varchar(64),

in \_root\_identifier varchar(64),

in \_type int2,

in \_handler varchar(64),

in \_action varchar(64),

in \_access int2,

in \_name varchar(32),

in \_title varchar(64),

in \_show\_in\_menu int2,

in \_icon varchar(32),

in \_icon\_color varchar(32),

in \_confirm int2,

in \_run\_mode int2,

in \_template text,

in \_context\_states varchar(64),

in \_controller\_name varchar(64),

in \_properties text,

in \_description text,

in \_state int2,

in \_last\_updater int4

) RETURNS boolean AS $$

DECLARE

\_\_new\_identifier\_seq bigint;

\_\_new\_identifier varchar(64);

BEGIN

if not exists (select 1 from kernel.\_action\_ where identifier = \_identifier and type = \_type) then --el tipo de accion ha cambiado--

if(\_type = 0) then --etiqueta

\_\_new\_identifier\_seq := nextval('kernel.action\_identifier\_sequence');

\_\_new\_identifier := md5(\_\_identifier\_seq::text) || md5(\_\_identifier\_seq::text);

else --accion o dep

\_\_new\_identifier := md5(\_handler) || md5(\_action);

if exists (select 1 from kernel.\_action\_ where identifier = \_\_new\_identifier) then

raise exception 'La combinación de "manejador" y "acción" ya está asociada a otra acción!!';

end if;

end if;

else

if(\_type = 0) then --etiqueta

\_\_new\_identifier := \_identifier;

else --accion o dep

\_\_new\_identifier := md5(\_handler) || md5(\_action);

if((\_identifier != \_\_new\_identifier) and exists (select 1 from kernel.\_action\_ where identifier = \_\_new\_identifier)) then

raise exception 'La combinación de "manejador" y "acción" ya está asociada a otra acción!!';

end if;

end if;

end if;

---

--normalizacion--

--

if(\_type = 0) then --etiqueta

\_handler := NULL;

\_action := NULL;

\_access := 0;

--\_title := NULL;

\_show\_in\_menu := 1;

\_confirm := NULL;

\_run\_mode := NULL;

\_template := NULL;

\_context\_states := NULL;

\_controller\_name := NULL;

\_properties := '{}';

else --accion o dep

if(\_run\_mode != 1) then

\_properties := '{}';

end if;

if(\_run\_mode = 2) then

\_template := NULL;

\_controller\_name := NULL;

end if;

end if;

update kernel.\_action\_ set

identifier = \_\_new\_identifier,

root\_identifier = NULLIF(\_root\_identifier, ''),

app\_identifier = \_app\_identifier,

type = \_type,

handler = \_handler,

action = \_action,

access = \_access,

name = \_name,

title = \_title,

show\_in\_menu = \_show\_in\_menu,

icon = \_icon,

icon\_color = \_icon\_color,

confirm = \_confirm,

run\_mode = \_run\_mode,

template = \_template,

context\_states = \_context\_states,

controller\_name = \_controller\_name,

properties = \_properties,

description = \_description,

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

RAISE NOTICE 'Acción actualizada con éxito!!';

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure update\_action

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure update\_action

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure update\_profile

#### Card of procedure update\_profile

|  |  |
| --- | --- |
| Name | update\_profile |
| Code | update\_profile |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure update\_profile

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_identifier int4,

in \_name varchar(64),

in \_description text,

in \_state int2,

in \_last\_updater int4,

in \_actions varchar[]

)

RETURNS BOOLEAN AS $$

DECLARE

BEGIN

if exists (select 1 from kernel.\_profile\_ where name = \_name and identifier != \_identifier) then

raise exception 'Ya existe un perfíl con el mismo nombre';

end if;

update

kernel.\_profile\_

set

name = \_name,

description = \_description,

state = \_state,

last\_updater = \_last\_updater,

last\_update\_date = now()

where

identifier = \_identifier;

if not kernel.insert\_profile\_has\_action(\_identifier, \_actions) then

raise exception 'No se pudieron actualizar las acciones del perfíl';

end if;

RAISE NOTICE 'Perfíl "%" actualizado!!', \_name;

RETURN TRUE;

EXCEPTION

WHEN OTHERS THEN

RAISE NOTICE '%', SQLERRM;

RETURN FALSE;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure update\_profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure update\_profile

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |

### Procedure validate\_action\_access

#### Card of procedure validate\_action\_access

|  |  |
| --- | --- |
| Name | validate\_action\_access |
| Code | validate\_action\_access |
| Parent | Physical Data Model 'soat[pg]' |
| Type | Function |

#### Text of the procedure validate\_action\_access

DROP TYPE IF EXISTS [%OWNER%.]validate\_action\_access\_tp CASCADE;

CREATE TYPE [%OWNER%.]validate\_action\_access\_tp as(success int2, content text, callback text);

CREATE OR REPLACE FUNCTION [%OWNER%.]%FUNC% (

in \_action\_identifier varchar(64),

in \_user\_identifier int4,

in \_profile\_identifier int4,

in \_department\_identifier int4,

in \_app\_identifier int4,

in \_session int2

)

RETURNS [%OWNER%.]validate\_action\_access\_tp AS $$

DECLARE

\_\_val\_return [%OWNER%.]validate\_action\_access\_tp;

\_\_app kernel.\_app\_%rowtype;

\_\_user kernel.\_user\_%rowtype;

\_\_profile kernel.\_profile\_%rowtype;

\_\_department kernel.\_department\_%rowtype;

\_\_coordination kernel.\_department\_%rowtype;

\_\_action kernel.\_action\_%rowtype;

BEGIN

\_\_val\_return.success := 0;

--VALIDAR APP

select \* into \_\_app from kernel.\_app\_ where identifier = \_app\_identifier;

if \_\_app is null then

if (\_session = 1) then

\_\_val\_return.content := 'La aplicacion no existe, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

else

\_\_val\_return.content := 'La aplicacion no existe'::text;

end if;

return \_\_val\_return;

elsif (\_\_app.state = 1 or \_\_app.state = 2) then

if (\_session = 1) then

\_\_val\_return.content := 'La aplicacion no esta activa, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

else

\_\_val\_return.content := 'La aplicacion esta inactiva'::text;

end if;

return \_\_val\_return;

elsif (\_\_app.state = 3 or (date\_trunc('day', now()) >= date\_trunc('day', \_\_app.expiration\_date))) then

update kernel.\_app\_ set state = 3 where identifier = \_app\_identifier and state != 3;

if (\_session = 1) then

\_\_val\_return.content := 'El periodo de prueba de la aplicacion a terminado, para disfrutar de la version completa contacte a su administrador, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

else

\_\_val\_return.content := 'El periodo de prueba de la aplicacion a terminado, para disfrutar de la version completa contacte a su administrador'::text;

end if;

return \_\_val\_return;

end if;

--VALIDAR ACCION

select \* into \_\_action from kernel.\_action\_ where identifier = \_action\_identifier and state != 2;

if \_\_action is null then

\_\_val\_return.content := 'La peticion no existe'::text;

return \_\_val\_return;

elsif \_\_action.state != 0 then

\_\_val\_return.content := 'La peticion esta inactiva'::text;

return \_\_val\_return;

elsif \_\_action.access = 2 then

---SI LA ACCION ES DE ACCESO HEREDADO VALIDAMOS LA ACCION PADRE

return kernel.validate\_action\_access(\_\_action.root\_identifier, \_user\_identifier, \_profile\_identifier, \_department\_identifier, \_app\_identifier, \_session);

elsif \_\_action.access = 3 then

---SI LA ACCION ES DE ACCESO PUBLICO NO NECESITAMOS VALIDAR MAS

\_\_val\_return.content := 'Accion de uso público'::text;

\_\_val\_return.success := 1;

return \_\_val\_return;

end if;

--VALIDAR SESSION

if (\_session != 1) then

\_\_val\_return.content := 'No has iniciado sesion'::text;

return \_\_val\_return;

end if;

--VALIDAR USUARIO

select \* into \_\_user from kernel.\_user\_ where identifier = \_user\_identifier and state != 2;

if \_\_user is null then

\_\_val\_return.content := 'El usuario con el que iniciaste sesión ya no existe, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

elsif \_\_user.state != 0 then

\_\_val\_return.content := 'El usuario con el que iniciaste sesión se desactivó, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

elsif \_\_user.profile\_identifier != \_profile\_identifier then

\_\_val\_return.content := 'Tu perfil de usuario ha cambiado, se cerrará la sesión para aplicar la nueva configuración asignada'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

elsif \_\_user.department\_identifier != \_department\_identifier then

\_\_val\_return.content := 'Tu departamento ha cambiado, se cerrará la sesión para aplicar la nueva configuración asignada'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

end if;

--VALIDAR PERFIL

select \* into \_\_profile from kernel.\_profile\_ where identifier = \_profile\_identifier and state != 2;

if \_\_profile is null then

\_\_val\_return.content := 'El perfil ligado al usuario con el que iniciaste sesión ya no existe, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

elsif \_\_profile.state != 0 then

\_\_val\_return.content := 'El perfil ligado al usuario con el que iniciaste sesión se desactivó, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

end if;

--VALIDAR DEPARTAMENTO

select \* into \_\_department from kernel.\_department\_ where identifier = \_department\_identifier;

if \_\_department is not null then

if \_\_department.state = 2 then

\_\_val\_return.content := 'El departamento ligado al usuario con el que iniciaste sesión ya no existe, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

elsif \_\_department.state != 0 then

\_\_val\_return.content := 'El departamento ligado al usuario con el que iniciaste sesión se desactivó, se cerrará la sesión'::text;

\_\_val\_return.callback := 'logout'::text;

return \_\_val\_return;

end if;

end if;

-- SI LA ACCION ES PARA USUARIOS REGISTRADOS

if \_\_action.access = 1 then

\_\_val\_return.content := 'Accion para usuarios registrados'::text;

\_\_val\_return.success := 1;

return \_\_val\_return;

end if;

--VALIDAR ACCESO A LA ACCION

if not exists (select 1 from kernel.\_profile\_has\_action\_ where action\_identifier = \_action\_identifier and profile\_identifier = \_profile\_identifier) then

\_\_val\_return.content := 'No tienes acceso a la petición'::text;

return \_\_val\_return;

end if;

\_\_val\_return.success := 1;

return \_\_val\_return;

EXCEPTION

WHEN OTHERS THEN

\_\_val\_return.success := 0;

\_\_val\_return.content := 'Ocurrió un error inesperado'::text;

\_\_val\_return.callback := null;

return \_\_val\_return;

END;

$$ LANGUAGE '%ProcLanguage%';

#### List of outgoing traceability links of the procedure validate\_action\_access

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Linked Object | Linked Object Type | Linked Model | Linked Package | Link Type |
| \_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_app\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_department\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_profile\_has\_action\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |
| \_user\_ | Table | soat[pg] | soat[pg] | DBCreateAfter (computed) |

#### List of diagrams containing the procedure validate\_action\_access

|  |  |
| --- | --- |
| Name | Code |
| sys.kernel | sys\_kernel |