

Pacote de Funções DOX (Schema) - PostgreSQL

Pensando em facilitar a busca de informações dentro do software por meio de instruções SQL, ao longo do tempo foram criadas várias funções e procedures que simplificaram ainda mais a busca de informações, seja por nossos consultores internos, ou por nossos clientes em suas automações.

A Ema disponibiliza um schema / package para criar no banco de dados que terá todas as funções e procedures.

Para isso, basta executar os scripts abaixo para o banco especificado:

- **Primeiramente, executar o script para criação do schema**

```
create schema pkg_ema authorization postgres; grant all on schema pkg_ema to postgres with grant option;
```

- **Em seguida, executar a criação das funções**

```
CREATE OR REPLACE FUNCTION pkg_ema.retorna_lista(in_lista text, delimitador character)
  RETURNS TABLE(out_id integer, out_str text)
AS $body$
declare
  dados record;
begin
  IN_LISTA = REPLACE(REPLACE(IN_LISTA, '(', ','), ')', '');

  IF DELIMITADOR = '' OR DELIMITADOR IS NULL THEN DELIMITADOR = ','; END IF;
  IF IN_LISTA <> '.' and IN_LISTA is not null and IN_LISTA <> '' and (IN_LISTA not like '/*%'
and IN_LISTA not like '%*/') THEN
    for dados in (select unnest(string_to_array(IN_LISTA, DELIMITADOR)) LISTA)
    LOOP
      BEGIN
        OUT_ID  = cast(dados.lista as integer);
```

```

        OUT_STR = dados.lista;
RETURN NEXT;

EXCEPTION
WHEN OTHERS then
    OUT_ID = 0;
OUT_STR = dados.lista;
    RETURN NEXT;
END;
END LOOP;
ELSE
OUT_ID = 0;
    OUT_STR = '0';
    RETURN NEXT;
END IF;

end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_lista(text, character) owner to postgres;

create or replace function pkg_ema.retorna_valor_variavel(xidprocesso in int4, xidvariavel in
int4) returns varchar as
$body$
    declare
    valor varchar(5000);
    begin
    select valoratual into valor
    from public.crm_processo_variavel
    where crm_processo_variavel.idprocesso = xidprocesso
    and crm_processo_variavel.idvariavel = xidvariavel ;
    return valor;
    end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_valor_variavel(int4, int4) owner to postgres;

create or replace function pkg_ema.retorna_valor_variavel_texto(xidprocesso in int4,
xidvariavel in int4) returns varchar as
$body$

```

```

declare
valor varchar(5000);
begin
select valoratual into valor
from public.crm_processo_variavel
where crm_processo_variavel.idprocesso = xidprocesso
and crm_processo_variavel.idvariavel = xidvariavel ;
return valor;
end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_valor_variavel_texto(int4, int4) owner to postgres;

```

```

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_clob(xidprocesso integer,
xidvariavel integer) RETURNS text as
$body$
declare
valor text;
begin
select valoratual into valor
from public.crm_processo_variavel
where crm_processo_variavel.idprocesso = xidprocesso
and crm_processo_variavel.idvariavel = xidvariavel ;
return trim(valor);
end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_valor_variavel_clob(int4, int4) owner to postgres;

```

```

create or replace function pkg_ema.retorna_valor_variavel_int(xidprocesso in int4,
xidvariavel in int4) returns int4 as
$body$
declare
valor int4;
begin
select cast(case when valoratual = '.' then '0' else valoratual end as int4)
into valor
from public.crm_processo_variavel
where crm_processo_variavel.idprocesso = xidprocesso

```

```

    and crm_processo_variavel.idvariavel = xidvariavel ;
    return valor;
end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_valor_variavel_int(int4, int4) owner to postgres;

create or replace function pkg_ema.retorna_valor_variavel_valor(xidprocesso in int4,
xidvariavel in int4) returns decimal(15,2) as
$body$
    declare
        valor decimal(15,2) ;
    begin
        select replace(replace(case when valoratual = '.' then '0' else valoratual end, '.', ''),
',', ',.')
        into valor
        from public.crm_processo_variavel
        where crm_processo_variavel.idprocesso = xidprocesso
        and crm_processo_variavel.idvariavel = xidvariavel ;
        return valor;
    end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_valor_variavel_valor(int4, int4) owner to postgres;

create or replace function pkg_ema.retorna_valor_variavel_data(xidprocesso in int4,
xidvariavel in int4) returns date as
$body$
    declare
        valor date;
    begin
        if (xidprocesso > 0 and xidvariavel > 0) then
            begin
                select case when valoratual <> '.' then to_date(valoratual,'dd/mm/yyyy HH24:MI')
                else to_date('01-01-1900','dd/mm/yyyy HH24:MI') end into valor
                from public.crm_processo_variavel
                where crm_processo_variavel.idprocesso = xidprocesso
                and crm_processo_variavel.idvariavel = xidvariavel ;
            end;
        else

```

```

    valor := to_date('01/01/1900','dd/mm/yyyy HH24:MI');
end if;
return valor;
end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_valor_variavel_data(int4, int4) owner to postgres;

create or replace function pkg_ema.retorna_valor_variavel_datahr(xidprocesso in int4,
xidvariavel in int4) returns timestamp as
$body$
    declare
        valor timestamp;
    begin
        if (xidprocesso > 0 and xidvariavel > 0) then
            begin
                select case when valoratual <> '.' then to_timestamp(valoratual,'dd/mm/yyyy HH24:MI')
                    else to_timestamp('01-01-1900','dd/mm/yyyy HH24:MI') end into valor
                from public.crm_processo_variavel
                where.crm_processo_variavel.idprocesso = xidprocesso
                and.crm_processo_variavel.idvariavel = xidvariavel ;
            end;
        else
            valor := to_timestamp('01/01/1900','dd/mm/yyyy HH24:MI');
        end if;
        return valor;
    end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_valor_variavel_datahr(int4, int4) owner to postgres;

create or replace function pkg_ema.retorna_coluna_grade(xidprocesso int4, xidatividade int4,
xidformulario int4, xidgrade int4, xidvalor int4)
returns character varying as
$body$
    declare
        leitura varchar(4000);
    begin
        select
            resposta into leitura

```

```

from public.crm_processo_grade_valor
where idprocesso = xidprocesso
and idformulario = xidformulario
and idatividade = xidatividade
and idgrade = xidgrade
and idvalor = xidvalor
and idrepeticao = (select max(idrepeticao)
from crm_processo_grade_valor xx
where xx.idatividade = crm_processo_grade_valor.idatividade
and xx.idprocesso = crm_processo_grade_valor.idprocesso
and xx.idformulario = crm_processo_grade_valor.idformulario);
return(trim(leitura));
end;
$body$

language plpgsql volatile cost 100;
alter function pkg_ema.retorna_coluna_grade(int4, int4, int4, int4, int4) owner to postgres;

create or replace function pkg_ema.retorna_coluna_grade_texto(xidprocesso int4, xidatividade
int4, xidformulario int4, xidgrade int4, xidvalor int4)
returns character varying as
$body$
declare
leitura varchar(4000);
begin
select
resposta into leitura
from public.crm_processo_grade_valor
where idprocesso = xidprocesso
and idformulario = xidformulario
and idatividade = xidatividade
and idgrade = xidgrade
and idvalor = xidvalor
and idrepeticao = (select max(idrepeticao)
from crm_processo_grade_valor xx
where xx.idatividade = crm_processo_grade_valor.idatividade
and xx.idprocesso = crm_processo_grade_valor.idprocesso
and xx.idformulario = crm_processo_grade_valor.idformulario);
return(trim(leitura));
end;
$body$

```

```

language plpgsql volatile cost 100;
alter function pkg_ema.retorna_coluna_grade_texto(int4, int4, int4, int4, int4) owner to
postgres;

create or replace function pkg_ema.retorna_coluna_grade_clob(xidprocesso int4, xidatividade
int4, xidformulario int4, xidgrade int4, xidvalor int4)
returns text as
$body$
declare
    leitura text;
begin
    select
        resposta into leitura
    from public.crm_processo_grade_valor
    where idprocesso = xidprocesso
        and idformulario = xidformulario
        and idatividade = xidatividade
        and idgrade = xidgrade
        and idvalor = xidvalor
        and idrepeticao = (select max(idrepeticao)
            from crm_processo_grade_valor xx
            where xx.idatividade = crm_processo_grade_valor.idatividade
            and xx.idprocesso = crm_processo_grade_valor.idprocesso
            and xx.idformulario = crm_processo_grade_valor.idformulario);
    return(trim(leitura));
end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_coluna_grade_texto(int4, int4, int4, int4, int4) owner to
postgres;

create or replace function pkg_ema.retorna_coluna_grade_int(xidprocesso int4, xidatividade
int4, xidformulario int4, xidgrade int4, xidvalor int4)
returns int4 as
$body$
declare
    leitura int4;

begin
    /*leitura := 0 ;

```

```

select (select to_number(decode(respostainteiro,'.',0,respostainteiro)) from dual)*/

select respostainteiro into leitura
from public.crm_processo_grade_valor
where idprocesso = xidprocesso
and idformulario = xidformulario
and idatividade = xidatividade
and idgrade = xidgrade
and idvalor = xidvalor
and idrepeticao = (select max(idrepeticao)
                    from crm_processo_grade_valor xx
                    where xx.idatividade = crm_processo_grade_valor.idatividade
                    and xx.idprocesso = crm_processo_grade_valor.idprocesso);

return(leitura);

end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_coluna_grade_int(int4, int4, int4, int4, int4) owner to
postgres;

create or replace function pkg_ema.retorna_coluna_grade_valor(xidprocesso int4, xidatividade
int4, xidformulario int4, xidgrade int4, xidvalor int4)
returns numeric(15,2) as
$body$
declare
    leitura numeric(15,2);
begin
    select respostavalor into leitura
    from public.crm_processo_grade_valor
    where idprocesso = xidprocesso
    and idformulario = xidformulario
    and idatividade = xidatividade
    and idgrade = xidgrade
    and idvalor = xidvalor
    and idrepeticao = (select max(idrepeticao)
                      from crm_processo_grade_valor xx
                      where xx.idatividade = crm_processo_grade_valor.idatividade
                      and xx.idprocesso = crm_processo_grade_valor.idprocesso);

```



```

    return(leitura);
end;
$body$

language plpgsql volatile cost 100;
alter function pkg_ema.retorna_coluna_grade_valor(int4, int4, int4, int4, int4) owner to
postgres;

create or replace function pkg_ema.retorna_coluna_grade_data( xidprocesso int4, xidatividade
int4, xidformulario int4, xidgrade int4, xidvalor int4)
returns date as
$body$
declare
    leitura date;
begin
    select respostadatahora into leitura
    from public.crm_processo_grade_valor
    where idprocesso = xidprocesso
    and idformulario= xidformulario
    and idatividade = xidatividade
    and idgrade  = xidgrade
    and idvalor  = xidvalor
    and idrepeticao = (select max(idrepeticao)
        from crm_processo_grade_valor xx
        where xx.idatividade = crm_processo_grade_valor.idatividade
        and xx.idprocesso  = crm_processo_grade_valor.idprocesso
        and xx.idformulario = crm_processo_grade_valor.idformulario);
    return(leitura); --fazer o trim depois

end;
$body$

language plpgsql volatile cost 100;
alter function pkg_ema.retorna_coluna_grade_data(int4, int4, int4, int4, int4) owner to
postgres;

create or replace function pkg_ema.retorna_coluna_grade_datahr( xidprocesso int4,
xidatividade int4, xidformulario int4, xidgrade int4, xidvalor int4)
returns timestamp as
$body$
declare

```

```

    leitura timestamp;
begin
    select respostadatahora into leitura
    from public.crm_processo_grade_valor
    where idprocesso = xidprocesso
    and idformulario= xidformulario
    and idatividade = xidatividade
    and idgrade = xidgrade
    and idvalor = xidvalor
    and idrepeticao = (select max(idrepeticao)
        from crm_processo_grade_valor xx
        where xx.idatividade = crm_processo_grade_valor.idatividade
        and xx.idprocesso = crm_processo_grade_valor.idprocesso
        and xx.idformulario = crm_processo_grade_valor.idformulario);
return(leitura);

end;
$body$
language plpgsql volatile cost 100;
alter function pkg_ema.retorna_coluna_grade_datahr(int4, int4, int4, int4, int4) owner to
postgres;

```

Chamando as Funções

-

Retorna lista

	RETORNO	
SQL	OUT_ID (Saída Integer)	OUT_STR (Saída Text)
select * from pkg_ema.retorna_lista('/*IDCLIENTE*/','');	0	0
select * from pkg_ema.retorna_lista('0','');	0	0
select * from pkg_ema.retorna_lista('','');	0	0
select * from pkg_ema.retorna_lista('(9,7)','','');	9 7	9 7
select * from pkg_ema.retorna_lista('(10;74)','','');	10 74	10 74

	RETORNO	
select * from pkg_ema.retorna_lista('91:5:S:SS','');	91 5 0 0	91 5 S SS
select * from pkg_ema.retorna_lista('ASD,ASDA,ASDA','');	0 0 0	ASD ASDA ASDA

- **Variável:**

```
select pkg_ema.retorna_valor_variavel(idprocesso, 1) as padrao,
       pkg_ema.retorna_valor_variavel_texto(idprocesso, 2) as texto,
       pkg_ema.retorna_valor_variavel_int(idprocesso, 3) as inteiro,
       pkg_ema.retorna_valor_variavel_valor(idprocesso, 4) as valor,
       pkg_ema.retorna_valor_variavel_data(idprocesso, 5) as data,
       pkg_ema.retorna_valor_variavel_datahr(idprocesso, 6) as datahr,
       pkg_ema.retorna_valor_variavel_clob(idprocesso, 7) as clob
from crm_processo where idprocesso = 0/*IDPROCESSO*/
```

- **Grade:**

```
select pkg_ema.retorna_coluna_grade(x.idprocesso, x.idatividade, x.idformulario, 1,
x.idvalor) as padrao,
       pkg_ema.retorna_coluna_grade_texto(x.idprocesso, x.idatividade, x.idformulario, 2,
x.idvalor) as texto,
       pkg_ema.retorna_coluna_grade_int(x.idprocesso, x.idatividade, x.idformulario, 3,
x.idvalor) as inteiro,
       pkg_ema.retorna_coluna_grade_valor(x.idprocesso, x.idatividade, x.idformulario, 4,
x.idvalor) as valor,
       pkg_ema.retorna_coluna_grade_data(x.idprocesso, x.idatividade, x.idformulario, 5,
x.idvalor) as data,
       pkg_ema.retorna_coluna_grade_datahr(x.idprocesso, x.idatividade, x.idformulario, 6,
x.idvalor) as datahr,
       pkg_ema.retorna_coluna_grade_clob(x.idprocesso, x.idatividade, x.idformulario, 7,
x.idvalor) as clob
from crm_processo_grade_valor x
where idprocesso = 0/*IDPROCESSO*/
and idatividade = 0/*IDATIVIDADE*/
```

```
and idformulario = 0/*IDFORMULARIO*/  
and idrepeticao = (select max(idrepeticao)  
                  from crm_processo_grade_valor  
                  where idprocesso = x.idprocesso  
                  and idatividade = x.idatividade)  
and idgrade = 1 order by idvalor
```

Revisão #6

Criado 3 December 2021 16:04:58 por Nicolly Andrielly

Atualizado 26 October 2023 17:55:55 por Nicolly Andrielly