CREATE TABLE public.ref\_type (

id INTEGER NOT NULL,

name VARCHAR(13),

CONSTRAINT ref\_type\_pkey PRIMARY KEY(id)

)

WITH (oids = false);

ALTER TABLE public.ref\_type

OWNER TO postgres;

CREATE TABLE public.object (

id SERIAL,

type\_id INTEGER,

parent\_id INTEGER,

CONSTRAINT object\_pkey PRIMARY KEY(id),

CONSTRAINT object\_fk FOREIGN KEY (parent\_id)

REFERENCES public.object(id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

NOT DEFERRABLE,

CONSTRAINT object\_fk1 FOREIGN KEY (type\_id)

REFERENCES public.ref\_type(id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

NOT DEFERRABLE

)

WITH (oids = false);

ALTER TABLE public.object

OWNER TO postgres;

CREATE TABLE public.area\_changing (

id SERIAL,

val\_1 REAL,

val\_2 REAL,

changing\_dt DATE,

obj\_id INTEGER,

area REAL,

CONSTRAINT area\_changing\_pkey PRIMARY KEY(id),

CONSTRAINT area\_changing\_fk FOREIGN KEY (obj\_id)

REFERENCES public.object(id)

ON DELETE RESTRICT

ON UPDATE RESTRICT

NOT DEFERRABLE

)

WITH (oids = false);

ALTER TABLE public.area\_changing

OWNER TO postgres;

CREATE OR REPLACE FUNCTION public.a (

p\_date date = '2021-09-01'::date

)

RETURNS TABLE (

id integer,

area real,

type text,

level text

) AS

$body$

DECLARE

BEGIN

return query

WITH RECURSIVE r AS (

SELECT po.id,

po.type\_id as type,

po.id::text as level

FROM public.object as po

WHERE po.parent\_id is null

UNION

SELECT o.id,

o.type\_id as type,

r.level || '.' || o.id as level

FROM public.object as o

JOIN r ON o.parent\_id = r.id)

SELECT r.id, area.area, rt.name::text, r.level

FROM r

left join public.ref\_type rt on rt.id = r.type

left join lateral (select ac.area from public.area\_changing ac where ac.obj\_id = r.id ORDER by ac.changing\_dt desc limit 1 )as area on true

order by r.level;

END;

$body$

LANGUAGE 'plpgsql'

VOLATILE

CALLED ON NULL INPUT

SECURITY INVOKER

PARALLEL UNSAFE

COST 100 ROWS 1000;

ALTER FUNCTION public.a (p\_date date)

OWNER TO postgres;

CREATE OR REPLACE FUNCTION public.b (

p\_date date

)

RETURNS TABLE (

id integer,

type text,

area real

) AS

$body$

DECLARE

BEGIN

return Query

select onlycircle.id, rt.name::text, area.area::REAL

from (

select \*

from public.object o

where o.type\_id = 2

) as onlycircle

join public.object op on op.id = onlycircle.parent\_id and op.type\_id = 1

left join public.ref\_type rt on rt.id = onlycircle.type\_id

left join lateral

(

select ac.area

from public.area\_changing ac

where ac.obj\_id = onlycircle.id

ORDER by ac.changing\_dt desc

limit 1

) as area on true ;

END;

$body$

LANGUAGE 'plpgsql'

VOLATILE

CALLED ON NULL INPUT

SECURITY INVOKER

PARALLEL UNSAFE

COST 100 ROWS 1000;

ALTER FUNCTION public.b (p\_date date)

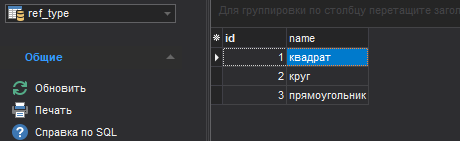
OWNER TO postgres;

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

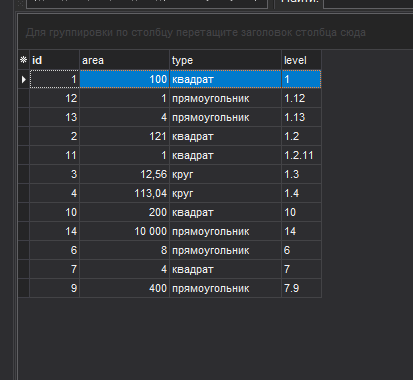
Изображение выглядит как текст, монитор, внутренний, экран

Автоматически созданное описание



select \*

from public.a('01.09.2021')



Level добавлен для удобности анализа данных и сортировки по нему (path)

select \*

from public.b('01.09.2021')

Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание