do language plpgsql $$

declare

v\_state TEXT;

v\_msg TEXT;

v\_detail TEXT;

v\_hint TEXT;

v\_context TEXT;

begin

create table messages (

id\_message SERIAL NOT NULL PRIMARY KEY,

from\_user INTEGER NOT NULL REFERENCES public.user(id\_user),

to\_user INTEGER NOT NULL REFERENCES public.user(id\_user),

datasent TIMESTAMP NOT NULL,

dataread TIMESTAMP NOT NULL,

mes\_content TEXT NOT NULL,

attached\_files INTEGER

);

exception when others then

get stacked diagnostics

v\_state = returned\_sqlstate,

v\_msg = message\_text,

v\_detail = pg\_exception\_detail,

v\_hint = pg\_exception\_hint,

v\_context = pg\_exception\_context;

raise notice E'Got exception:

state : %

message: %

detail : %

hint : %

context: %', v\_state, v\_msg, v\_detail, v\_hint, v\_context;

raise notice E'Got exception:

SQLSTATE: %

SQLERRM: %', SQLSTATE, SQLERRM;

raise notice '%', message\_text; -- invalid. message\_text is contextual to GET STACKED DIAGNOSTICS only

end; $$;

Та код рішення котре виводить дана функція:

NOTICE: Got exception:

state : 42P07

message: relation "messages" already exists

detail :

hint :

context: SQL statement "create table messages (id\_message SERIAL, from\_user INTEGER, to\_user INTEGER, datasent TIMESTAMP, dataread TIMESTAMP, mes\_content TEXT, mes\_content TEXT, attached\_files INTEGER)"

PL/pgSQL function inline\_code\_block line 11 at SQL statement

NOTICE: Got exception:

SQLSTATE: 42P07

SQLERRM: relation "messages" already exists

ERROR: column "message\_text" does not exist

LINE 1: SELECT message\_text

^

QUERY: SELECT message\_text

CONTEXT: PL/pgSQL function inline\_code\_block line 33 at RAISE

SQL state: 42703

Дане середовище для управління реляційної бази даних допомагає услідкувати за багатьма недоліками, що можуть бути в результаті некоректної роботи компонентів програми.