

**Бази даних та інформаційні системи**

**ЛАБОРАТОРНА РОБОТА №1**

**Збережені процедури СКБД PostgreSQL**

Виконав:

Ст. Прізвище Ім'я

Група .....

**Тема:** Вивчення поняття збережених процедур СКБД PostgreSQL v.11 /користувацьких функцій на процедурній мові PL/pgSQL попередніх версій PostgreSQL/.

**Мета роботи:** Ознайомлення з метою, перевагами та недоліками використання збережених процедур СКБД PostgreSQL, процесом їх розробки та застосування.

### Теоретичний матеріал:

У PostgreSQL представлені функції чотирьох видів:

- функції на мові запитів (функції, написані на SQL)
- функції на процедурних мовах (функції, написані, наприклад, на PL/pgSQL, PL/Tcl або PL/Python)
- внутрішні функції
- функції на мові C

Функції будь-яких видів можуть приймати в якості аргументів (параметрів) базові типи, складені типи або їх поєднання. Крім того, будь-які функції можуть повертати значення базового або складеного типу. Також можна визначити функції, які повертають набори базових або складених значень.

Функції багатьох видів можуть також приймати або повертати певні псевдотипи (наприклад, поліморфні типи).

Процедура представляє собою об'єкт бази даних, подібний до функції. Відмінність полягає в тому, що

- *процедура не повертає значення, і тому для неї не визначається тип повернення (хоча є спосіб повернути значення через вхідні параметри типу **inout**)*
- *процедура викликається явно, оператором **CALL**, тоді як функція викликається в складі запиту або команди DML*
- *починаючи з 11 версії в PostgreSQL представлені збережені процедури, які підтримують транзакції, тобто можна розпочати транзакцію, зафіксувати або відкотити її (на відміну від визначених користувачем функцій, що не можуть виконувати транзакції).*

Все, що говориться про створення користувацьких функцій, може бути застосовано і до процедур, за винятком того, що для процедур використовується команда **CREATE PROCEDURE**, не визначається тип результату, і до них не належать деякі властивості, наприклад, строгість (**strict**).

Загалом, збережена процедура - це іменований набір операторів SQL та певної процедурної мови, який зберігається на сервері у скомпільованому вигляді.

Отже, процедура - це блок з однієї або більше команд. Це може бути не просто один запит, а ціла програма, з власною логікою (оператори IF, цикли). Процедура може приймати задалегідь певні змінні і використовувати їх в своїх розрахунках,

завдяки чому, результат роботи процедури може бути динамічним і буде залежати від певних умов і/або стану одержуваних значень змінних.

У процедурі можна:

- використовувати оператори, які виконують будь-які операції DML в базі даних (вибірка, вставка, зміна або видалення даних), включаючи можливість виклику інших вбудованих процедур;
- використовувати вхідні параметри;
- повертати статус виконання для відображення вдалого або помилкового виконання;
- повертати декілька значень у формі вихідних параметрів.

Збережена процедура дозволяє згрупувати блок обчислень і послідовність запитів *всередині* сервера бази даних, таким чином, ми отримуємо силу процедурної мови і простоту використання SQL за значної економії накладних витрат на клієнт-серверну взаємодію

- виключаються додаткові звернення між клієнтом і сервером
- проміжні непотрібні результати не передаються між сервером і клієнтом
- є можливість уникнути численних аналізів одного запиту

В результаті це призводить до значного збільшення продуктивності в порівнянні з додатком, що не використовує збережені процедури.

Але крім переваг є ще деякі застереження:

- повільніша розробка програмного забезпечення, оскільки PL/pgSQL вимагає спеціалізованих навичок, якими не володіють багато розробників.
- складно керувати версіями і важко налагоджувати.
- ускладнюється переносимість до інших систем керування базами даних .

Онлайн-ресурси для ознайомлення з теоретичним матеріалом теми «Збережені процедури СКБД PostgreSQL»:

1. [PostgreSQL Documentation](https://www.postgresql.org/docs/)  
<https://www.postgresql.org/docs/>
2. [PostgreSQLTutorial.com website. PLpgsql](https://www.postgresqltutorial.com/postgresql-plpgsql/)  
<https://www.postgresqltutorial.com/postgresql-plpgsql/>
3. [PostgreSQLTutorial.com website. PostgreSQL-create-procedure](https://www.postgresqltutorial.com/postgresql-plpgsql/postgresql-create-procedure/)  
<https://www.postgresqltutorial.com/postgresql-plpgsql/postgresql-create-procedure/>

Перелік розділів та понять, з якими необхідно ознайомитись для виконання завдання лабораторної роботи (документація PostgreSQL v11 і вище).

## Глава 43. PL/pgSQL - процедурна мова SQL

### **Огляд**

Переваги використання PL/pgSQL

Підтримувані типи даних аргументів і значень

### **Структура PL / pgSQL**

#### **Оголошення**

Оголошення параметрів функції

ALIAS (*нове\_ім'я ALIAS FOR старе\_ім'я*)

Спадкування типів даних (*змінна %TYPE*)

Типи кортежів (*ім'я ім'я\_таблиці%ROWTYPE*)

Тип record (*ім'я RECORD*)

### **Вирази**

#### **Основні оператори**

Присвоювання

Виконання команди, що не повертає результат (*PERFORM запит*)

Виконання запиту, що повертає один рядок (*STRICT*)

Виконання динамічно сформованих команд (*EXECUTE*)

Статус виконання команди (*GET DIAGNOSTICS / FOUND*)

Не робити нічого (*NULL*)

#### **Керуючі структури (галуження)**

Команди для повернення значення з функції

Завершення процедури

Виклик процедури (*CALL*)

Умовні оператори

Прості цикли

Цикл за результатами запиту

Цикл за елементами масиву

Обробка помилок (*EXCEPTION*)

Отримання інформації про місце виконання (*GET DIAGNOSTICS stack = PG\_CONTEXT*)

#### **Курсори**

Оголошення курсорних змінних

Відкриття курсору

Використання курсорів

Обробка курсору в циклі

#### **Повідомлення і помилки**

Виведення повідомлень і помилок (*RAISE*)

Налагоджувальні перевірки в функції PL/pgSQL (*ASSERT*)

## Хід роботи

1. Опрацювати теоретичний матеріал.
2. Розробити збережену процедуру (або користувацьку функцію на процедурній мові PL/pgSQL для версій PostgreSQL нижче 11) відповідно до власних потреб роботи зі створюваною базою даних.
3. У збереженій процедурі необхідно використати:
  - оголошення вхідних та вихідних параметрів функції, псевдоніми, змінні та константи;
  - присвоювання змінній типу RECORD та команди PERFORM, STRICT або EXECUTE за потребою;
  - керуючі структури (IF/CASE/FOR ... щось за потребою);
  - команди обробки помилок EXCEPTION та виведення повідомлень RAISE – обов’язково;
  - використати курсор.
4. Оформити звіт про виконання лабораторної роботи, який має містити:
  - титульну сторінку;
  - тему, мету та завдання лабораторної роботи;
  - короткий перелік та опис використаних при створенні збереженої процедури команд, структур, понять;
  - навести скріни екрану з кодом збереженої процедури та отриманим результатом її виконання.
5. Завантажити в канал “БД. Лабораторна робота” свої команди в Teams.