**ЛАБОРАТОРНА РОБОТА №17**

ЗБЕРЕЖЕНІ ПРОЦЕДУРИ

*Мета:* основною метою даної лабораторної роботи є засвоєння студентами принципів створення, використання та управління збереженими процедурами в середовищі T-SQL, що використовується в системі управління базами даних Microsoft SQL Server. Збережені процедури є важливим інструментом для ефективної роботи з базами даних, оскільки вони дозволяють централізувати та стандартизувати обробку даних на стороні сервера.

Основні цілі лабораторної роботи включають:

1. Розуміння концепції збережених процедур: студенти повинні зрозуміти, що таке збережені процедури, які є їх основні характеристики та переваги використання в порівнянні з іншими методами виконання SQL-запитів.

2. Навички створення процедур: розробка навичок правильного кодування та структурування збережених процедур, включаючи декларацію параметрів, обробку помилок та оптимізацію виконання.

3. Управління та використання збережених процедур: практичне застосування створених процедур для вирішення типових завдань обробки даних, з можливістю модифікації та оптимізації існуючих процедур.

4. Безпека в збережених процедурах: освоєння практик програмування з урахуванням аспектів безпеки, таких як контроль доступу та уникнення SQL-ін'єкцій через правильне використання параметризації запитів.

**ТЕОРЕТИЧНІ ОСНОВИ**

Збережені процедури в T-SQL, що використовується в Microsoft SQL Server, є складовою частиною програмування сервера баз даних. Вони дозволяють об'єднати кілька операторів SQL в один виконавчий блок, який може бути викликаний додатком або іншими процедурами в базі даних. Цей розділ містить детальний огляд ключових концептів, які стосуються збережених процедур.

1. Основи збережених процедур: збережені процедури – це набори SQL-команд, які компілюються один раз і зберігаються у виконавчому форматі на сервері. Це забезпечує значне підвищення продуктивності, оскільки SQL Server може повторно використовувати вже оптимізований план виконання для подальших викликів.

2. Параметризація: збережені процедури можуть приймати параметри, що дозволяє їм виконувати оператори SQL з різними вхідними даними. Це робить їх надзвичайно гнучкими і потужними інструментами для повторного використання коду.

3. Безпека і контроль доступу: використання збережених процедур допомагає підвищити безпеку застосунків, обмежуючи прямий доступ до даних і забезпечуючи виконання лише дозволених операцій. Це також дозволяє реалізувати детальний контроль доступу до логіки обробки даних.

4. Обробка транзакцій: збережені процедури часто використовуються для управління транзакціями в базах даних. Вони можуть інкапсулювати логіку обробки транзакцій, забезпечуючи атомарність, консистентність, ізольованість і стійкість операцій (ACID).

5. Оптимізація виконання: компіляція і кешування виконавчих планів збережених процедур сприяють значному зниженню навантаження на сервер при частих запитах до бази даних. Оптимізація плану виконання базується на статистиці, яка може бути оновлена для підтримки актуальності виконання.

6. Уникнення SQL-ін'єкцій: збережені процедури зменшують ризик SQL-ін'єкцій через використання параметрів, які не дозволяють виконання шкідливого коду. Параметри обробляються як дані, а не як частина SQL-коду.

Приклади та особливості використання збережених процедур в T-SQL

***1. Проста збережена процедура***

Цей приклад демонструє створення базової збереженої процедури, яка виводить список клієнтів з певної категорії:

CREATE PROCEDURE GetCustomersByCategory

@CategoryID INT

AS

BEGIN

SELECT Name, Email FROM Customers WHERE CategoryID = @CategoryID;

END;

Ця процедура може бути викликана таким чином:

EXEC GetCustomersByCategory @CategoryID = 1;

***2. Збережена процедура з транзакцією***

Наступний приклад включає обробку транзакцій для забезпечення цілісності даних при оновленні інформації в кількох таблицях:

CREATE PROCEDURE UpdateCustomerOrders

@CustomerID INT,

@NewStatus VARCHAR(50)

AS

BEGIN

BEGIN TRANSACTION;

TRY

UPDATE Customers SET Status = @NewStatus WHERE ID = @CustomerID;

UPDATE Orders SET Status = @NewStatus WHERE CustomerID = @CustomerID;

COMMIT TRANSACTION;

CATCH

ROLLBACK TRANSACTION;

THROW;

END TRY

END;

**Особливості Використання**

1. Повторне використання коду: збережені процедури дозволяють централізувати бізнес-логіку, що сприяє її повторному використанню. Це покращує консистентність і зменшує дублювання коду.

2. Підвищення продуктивності: оскільки збережені процедури компілюються і кешуються на сервері, вони часто виконуються швидше, ніж еквівалентні запити, відправлені безпосередньо з клієнтської сторони.

3. Безпека: використання збережених процедур зменшує ризики, пов'язані з SQL-ін'єкціями, оскільки параметри обробляються як дані, а не як частина SQL-команди. Також можливе додаткове обмеження доступу до даних шляхом надання прав на виконання процедур замість прямого доступу до таблиць.

4. Масштабування: збережені процедури допомагають оптимізувати і масштабувати застосунки, що працюють з великими обсягами даних, за рахунок ефективного управління транзакціями та запитами.

Використання збережених процедур є важливим аспектом розробки ефективних і безпечних застосунків баз даних. Вони не тільки сприяють кращому управлінню кодом і даними, але й забезпечують високий рівень абстракції та контролю над складними операціями обробки даних.

**ЗАВДАННЯ**

***Завдання 1***

Створіть збережену процедуру.

Процедура приймає параметри:

- ідентифікатор клієнта.

- ідентифікатор співробітника.

- ідентифікатор виробника.

- ідентифікатор товару.

- кількість одиниць.

На першому етапі, процедура перевіряє наявність зазначеної кількості одиниць товару на складі.

Далі, на другому етапі, проводиться розрахунок суми замовлення (PRICE \* @qty).

На третьому етапі, якщо сума кредитного ліміту компанії більша або дорівнює сумі замовлення - зменшується кількість одиниць товару на складі, сума замовлення вираховується з кредитного ліміту компанії та замовлення вноситься до таблиці замовлень.

На четвертому етапі сума поточних продажів службовця збільшується на суму замовлення.

На п'ятому етапі сума поточних продажів офісу даного співробітника збільшується на суму замовлення.

На завершальному етапі процедура виводить повідомлення про статус операції.

**ДОДАТКОВІ МАТЕРІАЛИ:**

1. <https://www.w3schools.com/sql/sql_stored_procedures.asp>
2. <https://learn.microsoft.com/en-us/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=sql-server-ver16>
3. <https://learn.microsoft.com/uk-ua/sql/relational-databases/native-client-odbc-stored-procedures/running-stored-procedures?view=sql-server-2016>

**КОНТРОЛЬНІ ЗАПИТАННЯ**

1. Що таке збережена процедура в контексті SQL Server? Опишіть визначення збереженої процедури, її основні характеристики та призначення в управлінні базами даних.

2. Які переваги використання збережених процедур порівняно з звичайними SQL-запитами? Наведіть приклади, як збережені процедури можуть покращити продуктивність, безпеку та управління кодом.

3. Як створити збережену процедуру в SQL Server? Опишіть кроки та синтаксис для створення збереженої процедури, включно з декларацією параметрів і виконавчим блоком коду.

4. Як можна викликати збережену процедуру? Деталізуйте процес виклику збереженої процедури з різних додатків або інших процедур, включаючи передачу параметрів.

5. Які є особливості управління транзакціями в збережених процедурах? Розгляньте, як збережені процедури використовуються для контролю транзакцій і як це впливає на цілісність даних.

6. Як збережені процедури впливають на безпеку бази даних? Обговоріть механізми, які використовують збережені процедури для підвищення безпеки, зокрема захист від SQL-ін'єкцій.

7. Як оптимізувати виконання збереженої процедури? Розкрийте методи та практики для оптимізації продуктивності збережених процедур, включно з індексацією та плануванням запитів.

8. Які можливості для логування та аудиту надають збережені процедури? Визначте, як збережені процедури можуть бути використані для збору даних про виконання та моніторингу доступу до бази даних.