12. ПРОЦЕДУРА ↔ ФУНКЦИЯ

Процедура

- Позволяют объединить запросы и сохранить на сервере
- Быстрее
- В общем случае не имеет возвращаемого значения, но можно вернуть что-либо с помощью OUT
- Выполняется через CALL или EXECUTE
- Используются вместо запроса на выборку данных или изменение/удаление
- Позволяет ограничить доступ к БД, скрыть от пользователя структуру

Пример

```
CREATE TABLE Products
(
    Id INT IDENTITY PRIMARY KEY,
    ProductName NVARCHAR(30) NOT NULL,
    Manufacturer NVARCHAR(20) NOT NULL,
    ProductCount INT DEFAULT 0,
    Price MONEY NOT NULL
);
```

Процедура для извлечения данных из таблицы:

```
USE productsdb;
GO
CREATE PROCEDURE ProductSummary AS
SELECT ProductName AS Product, Manufacturer, Price
FROM Products
```

Также можно использовать BEGIN..END

Выполнение:

```
EXECUTE ProductSummary
```

Удаление

```
DROP PROCEDURE ProductSummary
```

Функция

- Можно использовать в операторах (н. в SELECT)
- Два типа возвращаемых значений: скаляр/таблица
- Выполняется через SELECT
- Используются вместо часто выполняющихся запросов

Пример (для той же таблицы)

Функция считает общую стоимость трёх продуктов (цена*кол-во)

```
USE productsdb;
GO
CREATE FUNCTION costCount(@id_1 INT,
@id_2 INT, @id_3 INT)
RETURNS INT
AS
BEGIN
  DECLARE @count INT = 0
  DECLARE @price INT = 0
  SELECT @count = SUM(ProductCount)
  FROM Products
  WHERE Id IN (@id_1, @id_2, @id_3)
  SELECT @price = SUM(Price)
  FROM Products
  WHERE Id IN (@id_1, @id_2, @id_3)
  RETURN @price*@count
END
```

Выполнение:

```
SELECT productsdb.dbo.costCount(
1, 2, 3) AS cost
```

Удаление

```
DROP FUNCTION costCount
```