

Пользовательские функции и триггеры

Пользовательские функции

Скалярные функции

Возвращают одно значение любого типа данных, кроме `CURSOR`, `TIMESTAMP` и таблиц.

Синтаксис

```
CREATE FUNCTION имя_функции (@параметр тип_данных)

RETURNS тип_данных

AS

BEGIN

    -- тело функции

    RETURN (выражение);

END;
```

Пример

Функция, возвращающая квадрат числа:

```
CREATE FUNCTION Square(@Value int)
```

```
RETURNS int

AS

BEGIN

    RETURN @Value * @Value;

END;
```

Табличные функции

Возвращают таблицу. Делятся на:

- **Подставляемые:** содержат единственный оператор `SELECT`.
- **Многооператорные:** содержат несколько операторов T-SQL.

Синтаксис (подставляемая)

```
CREATE FUNCTION имя_функции (@параметр тип_данных)

RETURNS TABLE

AS

RETURN (SELECT выражение);
```

Синтаксис (многооператорная)

```
CREATE FUNCTION имя_функции (@параметр тип_данных)

RETURNS @Результат TABLE (определение_таблицы)

AS

BEGIN

    -- тело функции

    RETURN;

END;
```

Пример

Функция, возвращающая сотрудников старше заданного возраста:

```
CREATE FUNCTION GetOlderEmployees(@Age int)

RETURNS @Employees TABLE (Name nvarchar(50), Age int)

AS

BEGIN

    INSERT INTO @Employees

    SELECT Name, Age FROM Employees WHERE Age > @Age;

    RETURN;
```

```
END;
```

Триггеры

Назначение

Триггер — это хранимая процедура, которая выполняется автоматически при наступлении определенного события.

Типы триггеров

1. **DML-триггеры**: срабатывают при `INSERT`, `UPDATE`, `DELETE`.
2. **DDL-триггеры**: активируются командами `CREATE`, `ALTER`, `DROP`.
3. **INSTEAD OF**: выполняются вместо операции.
4. **AFTER**: выполняются после операции.

Синтаксис

```
CREATE TRIGGER имя_триггера  
  
ON таблица  
  
AFTER [INSERT, UPDATE, DELETE]  
  
AS  
  
BEGIN  
  
    -- тело триггера  
  
END;
```

Особенности выполнения триггеров

При вызове триггера будут выполнены операторы SQL, указанные после ключевого слова `AS`. Вы можете использовать несколько операторов, включая такие конструкции, как `IF` и `WHILE`.

Операторы, недопустимые в триггерах

В триггерах запрещено использование следующих операторов:

- `ALTER DATABASE`
- `CREATE DATABASE`
- `DISK INIT`
- `DISK RESIZE`
- `DROP DATABASE`
- `LOAD DATABASE`
- `LOAD LOG`
- `RECONFIGURE`
- `RESTORE DATABASE`
- `RESTORE LOG`

Пример 1: Триггер для логирования

```
CREATE TRIGGER LogInsert  
  
ON Employees  
  
AFTER INSERT  
  
AS
```

```
BEGIN

    INSERT INTO LogTable (Operation, Time)

    VALUES ('INSERT', GETDATE());

END;
```

Пример 2: INSTEAD OF

```
CREATE TRIGGER PreventDelete

ON Employees

INSTEAD OF DELETE

AS

BEGIN

    PRINT 'Удаление запрещено!';

END;
```

Временные таблицы

- `inserted` : содержит новые данные.
- `deleted` : содержит старые данные.

Хранятся в оперативной памяти, а не на диске

Пример: Триггер на обновление

```
CREATE TRIGGER CheckSalary

ON Employees

FOR UPDATE

AS

BEGIN

    DECLARE @OldSalary int, @NewSalary int;

    SELECT @OldSalary = Salary FROM deleted;

    SELECT @NewSalary = Salary FROM inserted;

    IF @NewSalary > @OldSalary * 1.2

    BEGIN

        PRINT 'Повышение зарплаты более чем на 20% запрещено';

        ROLLBACK;

    END;

END;
```