

Системи і технології управління базами даних

Лекція на тему:

«Збережені процедури та
тригери»

Збережена процедура

Збережена процедура MS SQL Server – це набір операторів мови T-SQL.

Збережена процедура зберігається у базі даних і є об'єктом цієї бази, так само, як таблиці, уявлення, користувачі. Вона виконується на сервері, що значно пришвидшує роботу.

Синтаксис створення процедури

```
CREATE PROC [EDURE] procedure_name  
    [number]  
    [{@parameter data_type]  
    [VARYING] [= default] [OUTPUT] ][,...n]  
    AS sql_statement [...n]
```

У цьому синтаксисі жирним шрифтом виділені обов'язкові слова.

Пояснення параметрів

procedure name - назва процедури, яка не повинна збігатися з ключовими словами SQL;;

number – ціле число, яке дозволяє створювати групу процедур з однією назвою, але різними номерами; така група процедур може бути видалена одним оператором;

@parameter data type – параметр процедури та його тип; ім'я параметра завжди починається із символу @; тип параметра – допустимий тип даних SQL Server; значення параметра задається в момент виклику процедури; параметри – це локальні змінні, тобто в різних процедурах можуть бути параметри з однаковими назвами;

VARYING - використовується для параметрів, значення яких можуть змінюватися під час виконання процедури;

= default - задає значення параметра за замовченням;

OUTPUT – визначає параметр як вихідний параметр.

Приклад збереженої процедури

Процедура повертає всіх авторів, їхні статті, їхні публікації з 4 таблиць.
Ця процедура без параметрів.

```
USE pubs
```

```
IF EXISTS (SELECT name FROM sysobjects WHERE name =  
          'au_info_all' AND type = 'P') DROP PROCEDURE au_info_all
```

```
GO
```

```
CREATE PROCEDURE au_info_all
```

```
AS SELECT au_lname, au_fname, title, pub_name FROM authors a INNER  
JOIN titleauthor ta ON a.au_id = ta.au_id INNER JOIN titles t ON  
t.title_id = ta.title_id INNER JOIN publishers p ON t.pub_id = p.pub_id
```

```
GO
```

Виклик процедури

EXECUTE au_info_all

Або

EXEC au_info_all

Або

au_info_all

Приклад процедури з параметром

```
USE pubs
IF EXISTS (SELECT name FROM sysobjects WHERE name =
'au_info' AND type = 'P') DROP PROCEDURE au_info
GO
CREATE PROCEDURE au_info @lastname varchar(40),
@firstname varchar(20)
AS SELECT au_lname, au_fname, title, pub_name FROM
authors a INNER JOIN titleauthor ta ON a.au_id = ta.au_id
INNER JOIN titles t ON t.title_id = ta.title_id INNER JOIN
publishers p ON t.pub_id = p.pub_id WHERE au_fname =
@firstname AND au_lname = @lastnameGO
```

Виклик процедури з параметрами

```
EXECUTE au_info 'Dull', 'Ann'
```

Або

```
EXECUTE au_info @lastname = 'Dull', @firstname = 'Ann'
```

Або

```
EXECUTE au_info @firstname = 'Ann', @lastname = 'Dull'
```

Або

```
EXEC au_info 'Dull', 'Ann'
```

Або

```
EXEC au_info @lastname = 'Dull', @firstname = 'Ann'
```

Або

```
EXEC au_info @firstname = 'Ann', @lastname = 'Dull'
```

Або:

```
au_info 'Dull', 'Ann'–
```

Або

```
au_info @lastname = 'Dull', @firstname = 'Ann'–
```

Або

```
au_info @firstname = 'Ann', @lastname = 'Dull'
```

Видалення процедури

DROP PROCEDURE { *procedure* } [,...*n*]

Тригери

Тригери – це спеціальний тип збережених процедур, які запускаються автоматично при виконанні певних дій з таблицями бази даних. Кожний тригер прив'язується до конкретної таблиці. Коли користувач БД робить спробу, наприклад, змінити якусь таблицю, сервер автоматично запускає тригер, та, якщо він завершився успішно, дозволяється виконати зміни.

Типи тригерів

INSERT TRIGGER. Тригери цього типу запускаються при спробі додавання запису за допомогою команди INSERT.

UPDATE TRIGGER. Тригери цього типу запускаються при спробі змінити дані за допомогою команди UPDATE.

DELETE TRIGGER. Тригери цього типу запускаються при спробі видалити записи за допомогою команди DELETE.

Важливі параметри тригера

Поведінку тригера визначають два параметри.

AFTER. Тригер виконується після успішного виконання команд, які викликали його. Такі тригери можуть бути визначені тільки для таблиць.

INSTEAD OF. Тригер викликається замість команд. Такі тригери можуть бути визначені як для таблиць, так і для уявлень.

Синтаксис створення тригера

```
CREATE TRIGGER trigger_name
  ON { table | view }
  [ WITH ENCRYPTION ]
  {
    { { FOR | AFTER | INSTEAD OF } [DELETE] [,] [INSERT] [,] [UPDATE] }

    [ WITH APPEND ]
    [ NOT FOR REPLICATION ]
    AS sql_statement [ ...n ]
  }
  |
  {
    [ { IF UPDATE ( column )
    [ { AND | OR } UPDATE ( column ) ]
      [ ...n ]
    } ]
    sql_statement [ ...n ]
  }
}
```

Пояснення синтаксису

trigger_name

Цей аргумент задає назву тригера. Воно повинно бути унікальним усередині БД.

Table | view

Назва таблиці або уявлення, до якої (якого) прив'язується тригер.

WITH ENCRYPTION

Задає шифрування тексту тригера.

AFTER

Якщо вказано цей аргумент, тригер буде запускатися тільки у разі успішного завершення команд. Цей аргумент приймається за замовченням.

INSTEAD OF

Якщо вказано цей аргумент, тригер буде виконуватися замість команд, для яких визначений цей тригер.

{ [DELETE] [,] [INSERT] [,] [UPDATE] }

Ця конструкція визначає, які команди призведуть до виконання тригера. Повинно бути вказано хоча б одне з цих значень.

Пояснення синтаксису

WITH APPEND

Цей аргумент має сенс для версій нижчих 7. Він дозволяє створення нового тригера без видалення попереднього того ж самого типу для однієї таблиці.

NOT FOR REPLICATION

Вказує, що тригер не може бути виконаний при запуску механізму реплікацій.

AS sql_statement [...n]

Ця конструкція визначає набір команд T-SQL, які будуть виконані при запуску тригера.

IF UPDATE (*column*)

Використання цього аргумента дозволяє виконати тригер при модифікації певного стовпця таблиці. Використовується тільки для команд INSERT, UPDATE.

{ AND | OR } UPDATE (*column*)

Попереднє можна використовувати для декількох стовпців.

Заборонені команди

Усередині тригера не можна використовувати такі команди:

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

Видалення тригерів

```
DROP TRIGGER { trigger } [ ,...n ]
```

Приклад тригера

- USE pubs
- IF EXISTS (SELECT name FROM sysobjects WHERE name = 'reminder' AND type = 'TR') DROP TRIGGER reminder
- GO
- CREATE TRIGGER reminderON titlesFOR INSERT, UPDATE, DELETE
- AS EXEC master..xp_sendmail 'MaryM', 'Don't forget to print a report for the distributors.'
- GO

Приклад триггера

- USE pubs
- IF EXISTS (SELECT name FROM sysobjects WHERE name = 'reminder' AND type = 'TR')
DROP TRIGGER reminder
- GO
- CREATE TRIGGER reminder
- ON Titles
- FOR INSERT, UPDATE AS RAISERROR (50009, 16,
10)
- GO