Хранимые процедуры

Хранимые процедуры работают в среде SQL Server. Хранимые процедуры могут возвращать значения и модифицировать их. В хранимые процедуры можно передавать значения и получать посредством них результат работы, совершенно не обязательно имеющий отношение к рабочей таблице. Процедура может вычислять результат в процессе работы.

Преимущество использования процедур:

1. Высокая производительность
2. Преимущество разработки систем в архитектуре клиент-сервер
3. Уровень безопасности
4. Усиление правил сервера, работающих с данными

Хранимые процедуры компилируются при первом выполнении и сохраняются в системной таблице текущей БД. При компиляции они оптимизируются. Оптимизированные хранимые процедуры могут значительно улучшить производительность системы.

Создание хранимых процедур

Синтаксис

CREATE PROCEDURE [владелец,] имя_процедуры

[; число] [@имя_параметра тип_данных]

Пример

create procedure all_employees

as select * from employees

exec all_employees - вызов

Новую процедуру можно создать только в текущей БД.

Параметры в процедурах

Имеется возможность передавать внутрь хранимых процедур значения, с которыми она должна работать. Основные принципы применения параметров в хранимых процедурах:

1. может быть определен 1 или несколько параметров
2. имя параметра начинается с символа @
3. имена параметров являются локальными в той процедуре, где они используются

Пример

create procedure proc7(@p1 char(15), @p2 char(20), @p3 int) as insert into workers values(@p1, @p2, @p3)

exec proc7 'Petrov', Sules, 3333

Хранимая процедура не может быть модифицирована непосредственно, поэтому сначала следует удалить ее с помощью команды **DROP PROCEDURE,** а затем создать заново.

Инструкции управления ходом выполнения процедур

IF ... ELSE

Синтаксис

IF выражение

инструкция

[ESLE]

[IF выражение инструкция]

Пример

if exists (select * from workers

where badge = 1234)

print 'entry avialable'

if exists (select * from employees

where name = 'Petrov')

print 'employees present'

else print 'employees not found'

BEGIN ... END

Эти ключевые слова применяются для обозначения набора инструкций Transact-SQL, которые являются одним блоком.

Пример

if exists (select * from workers

where badge = 1234)

begin

print 'entry avialable'

select name, department from employees where badge = 1234

end

WHILE

Это ключевое слово применяется для определения условий, при истинности которых выполняется одна или несколько инструкций Transact SQL.

WHILE

<логическое выражение >

<инструкция SQL >

Пример

declare @x int - объявление переменной

select @x = 1

while @x <5

begin

print 'x<5'

select @x = @x+1

end

Определение и использование переменных

Переменные определяются с помощью инструкции **DECLARE**, где устанавливается тип данных переменной. Начальное значение для переменной устанавливается с помощью инструкции **SELECT**

Синтаксис

DECLARE @имя_переменной тип_данных

Для определения значения локальной переменной применяется инструкция **SELECT**

Синтаксис

SELECT @имя_переменной = выражение

Примеры

```
declare @mynam int

select @mynam = count(*) from Workers

declare @mychar char(2)

select @mychar = convert(char(2), @mynam)

declare @mess char(40)

select @mess = 'There are' + @mychar +'rows in the table workers'

print @mess
```

Выражение CASE

Применяется для селективного выбора на основании нескольких опций

Пример

```
select name, division =

case department

when "Sales" then "Sales @ Harket"
```

when "Field Service" then "Support"

when "Logistics" then "Parts"

else "Other department"

end, badge

from company

Триггеры

Триггеры - это методы, с помощью которых разработчик приложений SQL Server и аналитик базы данных может обеспечить целостность базы данных. Триггеры дают возможность базе данных поддерживать условные правила, не полагаясь на программное обеспечение приложений.

SQL Server принимает правила и установки по умолчанию перед записью информации в БД. Эти средства представляют собой некий предварительный фильтр для информации и могут предотвратить определенное действие в отношении данных на основании своей роли - контроль ввода и изменение данных.

Триггер - это постфильтр, применяемый после обновления данных и воздействия правил и установок по умолчанию.

Синтаксис

CREATE TRIGGER [владелец.] имя триггера

ON [владелец.] имя таблицы

FOR { INSERT, UPDATE, DELETE} [WITH ENCRYPTION] AS инструкции SQL

Когда выполняется триггер, SQL Server создает специальную таблицу, куда он помещает данные, вызвавшие выполнение триггера. Эта таблица имеет название ... для операций обновления и добавления и delete для операций удаления. Поскольку триггеры выполняются после завершения операций, строки в таблице ... всегда дублируют одну или несколько записей базовой таблицы.