

# **Лекция 13. Работа с курсором. Применение курсора. Управление курсором.**

**Курсор** – это особый временный объект SQL, предназначенный для использования в программах и хранимых процедурах.

С его помощью можно в цикле пройти по результирующему набору строк запроса, по отдельности считывая и обрабатывая каждую его строку.

В хранимых процедурах с помощью курсоров можно выполнять сложные вычисления, которые трудно выразить с помощью синтаксиса инструкции SELECT.

# Microsoft SQL Server реально поддерживает два различных типа курсоров:

**курсоры Transact-SQL** и **курсоры API** (курсоры программного интерфейса приложений). Курсоры API создаются внутри приложения, использующего объекты Microsoft ActiveX Data Objects (ADO), OLE DB, ODBC или DB-Library.

Каждое из этих API поддерживает несколько отличающиеся функциональные возможности и использует различный синтаксис. Здесь мы не будем подробно обсуждать курсоры API; если вы планируете использовать их, обратитесь к соответствующей документации на API и языку программирования, который вы собираетесь применить.

*Курсоры Transact-SQL* создаются с помощью команды **DECLARE CURSOR**. Как объект курсора, так и множество строк, на которое он указывает, должны существовать на сервере.

Подобные курсоры называются серверными курсорами. Если вы используете серверный курсор из приложения, соединенного с SQL Server через сеть, каждая операция с курсором требует двустороннего сетевого взаимодействия. Библиотеки API-курсоров, поддерживающие серверные курсоры, поддерживают также клиентский курсор, который существует в клиентской системе и кэширует строки, которые он обрабатывает на клиенте.

# Реализация курсоров в среде MS SQL Server

SQL Server поддерживает три вида курсоров:

- *курсоры SQL* применяются в основном внутри триггеров, хранимых процедур и сценариев;
- *курсоры сервера* действуют на сервере и реализуют программный интерфейс приложений для ODBC, OLE DB, DB\_Library;
- *курсоры клиента* реализуются на самом клиенте. Они выбирают весь результирующий набор строк из сервера и сохраняют его локально, что позволяет ускорить операции обработки данных за счет снижения потерь времени на выполнение сетевых операций.

# ***Характеристики курсоров***

- **Отражение изменений**

Способность курсора отражать изменения в данных называется ***чувствительностью курсора***.

При создании вашего курсора могут быть независимо определены два вида чувствительности: изменения каких строк включаются во множество (членство множества) и отражение изменений в исходных строках.

- **Прокрутка**

Второй характеристикой курсора является способность осуществления прокрутки как вперед, так и назад, либо только вперед. Здесь имеет место извечная для программирования дилемма: скорость против гибкости. Последовательные курсоры (forward-only) работают значительно быстрее, но имеют меньшую гибкость.

- **Обновление**

Последней характеристикой, используемой для классификации курсоров, является возможность обновления строк курсором. Опять же, курсоры "только чтение" обычно более производительны, но имеют меньшую гибкость.

# Типы курсоров

Transact-SQL поддерживает четыре различных типа курсоров: статические, ключевые, динамические и курсоры быстрого доступа, или "пожарные" (firehose). Каждый тип курсора хранит различные данные относительно строк, на которые он указывает, кроме того, каждому типу курсора свойственны различные сочетания характеристик, рассмотренных в предыдущем разделе.

- **Статические курсоры**

Статический курсор делает как бы моментальный снимок данных, задаваемых оператором SELECT, и хранит их в базе данных tempdb. Он "не чувствует" изменений в структуре или в значениях данных, а поскольку любые модификации будут отражены только в копии, этот курсор всегда открывается в режиме "только чтение". Статические курсоры, однако, могут быть объявлены как последовательные или как прокручиваемые.

- **Ключевые курсоры**

Ключевой курсор копирует в базу tempdb только те столбцы, которые уникально идентифицируют каждую строку. Чтобы иметь возможность объявить ключевой курсор, каждая таблица, входящая в определение оператора SELECT, должна иметь уникальный индекс, который задает копируемый набор – ключ.

Ключевые курсоры могут быть как модифицируемыми, так и иметь режим "только для чтения". Они также могут быть прокручиваемыми или последовательными.

Членство в ключевом курсоре фиксируется на момент объявления курсора. Если в процессе открытого состояния курсора добавляется строка, удовлетворяющая условию отбора, она не будет добавлена во множество.

# Типы курсоров

- **Динамические курсоры**

Динамический курсор ведет себя так, как если бы при каждом обращении к строке повторно выполнялся оператор SELECT. Динамические курсоры отражают изменения, связанные как с членством, так и со значениями исходных данных, независимо от того, сделаны ли эти изменения внутри курсора, либо внесены другим пользователем.

Для динамических курсоров действует одно ограничение: используемый для определения курсора оператор SELECT может содержать фразу ORDER BY только в том случае, если имеется индекс, включающий в себя столбцы, используемые в фразе ORDER BY. Если вы объявляете ключевой курсор с использованием фразы ORDER BY, не оперирующей индексом, SQL Server преобразует курсор в ключевой.

- **Курсоры быстрого доступа**

SQL Server поддерживает специальную оптимизированную форму не прокручиваемого курсора, допускающего только чтение. Этот вид курсора объявляется с использованием ключевого слова FAST\_FORWARD, и чаще всего его называют "пожарным" курсором (firehose).

"Пожарные" курсоры очень эффективны, но при их использовании имеются два важных ограничения. Во-первых, если в операторе определения SELECT курсора вы использовали столбцы с типом данных text, ntext или image, а также фразу TOP, SQL Server преобразует курсор в ключевой.

Во-вторых, если оператор SELECT, который вы использовали для определения курсора, содержит таблицы, имеющие триггеры, и таблицы, не имеющие триггеров, курсор преобразуется в статический. Триггеры представляют собой сценарии Transact-SQL, которые автоматически исполняются сервером при выполнении для таблицы операторов Data Manipulation Language (DML).

Чтобы ...	Синтаксис оператора SQL
Создать курсор	DECLARE имя_курсора CURSOR [видимость] [прокрутка] [тип] [блокировка] [TYPE_WARNING] FOR оператор_выборки [FOR UPDATE [OF имена_столбцов]]
Открыть курсор	OPEN [GLOBAL] курсор_или_переменная
Закрыть курсор	CLOSE [GLOBAL] курсор_или_переменная
Освободить курсор	DEALLOCATE [GLOBAL] курсор_или_переменная
Использовать простую команду FETCH	FETCH курсор_или_переменная
Извлечь строку с записью ее в переменные	FETCH курсор_или_переменная INTO список_переменных
Осуществить выборку строки по ее абсолютной позиции	FETCH {FIRST   LAST   ABSOLUTE n) FROM курсор_или_переменная
Осуществить выборку строки по ее относительной позиции	FETCH {NEXT   PRIOR   RELATIVE n) FROM курсор_или_переменная
Выполнить позиционное обновление	UPDATE таблица_или_представление SET список_для_модификации WHERE CURRENT OF курсор_или_переменная
Выполнить позиционное удаление	DELETE таблица_или_представление WHERE CURRENT OF курсор_или_переменная
Использовать функцию CURSOR_STATUS для запроса статуса курсора	CURSOR_STATUS ('local', 'локальный_курсor') или CURSOR_STATUS ('global', 'глобальный_курсor') или CURSOR_STATUS ('variable', 'курсорная_переменная')



# Синтаксис

## ISO Syntax

```
DECLARE cursor_name [ INSENSITIVE ] [ SCROLL ] CURSOR
    FOR select_statement
    [ FOR { READ ONLY | UPDATE [ OF column_name [ ,...n ] ] } ]
[;]
```

## Transact-SQL Extended Syntax

```
DECLARE cursor_name CURSOR [ LOCAL | GLOBAL ]
    [ FORWARD_ONLY | SCROLL ]
    [ STATIC | KEYSET | DYNAMIC | FAST_FORWARD ]
    [ READ_ONLY | SCROLL_LOCKS | OPTIMISTIC ]
    [ TYPE_WARNING ]
    FOR select_statement
    [ FOR UPDATE [ OF column_name [ ,...n ] ] ]
[;]
```

# Задание

- Использование в форме процедур, функций и триггеров.
- Создание 5 курсоров