**Тема30. Курсоры**

1. **Использование курсоров**

SQL-запросы, связанные с извлечением данных, работают с наборами строк, которые называют **результирующими,** они соответствуют условиям выборки, их может быть «0» или больше. При использовании простых конструкций SELECT невозможно получить первую, следующую строку или предыдущие 10 строк. Это особенность функционирования реляционной СУБД.

**Курсор** - это запрос к базе данных, хранящийся на сервере, выборка, полученная в результате выполнения SELECT.

После того, как курсор сохранен, приложения могут «прокручивать» строки в прямом или обратном порядке.

Чаще всего СУБД предоставляет следующие возможности с курсорами:

1. помечать курсор для чтения (нельзя обновлять, удалять данные);
2. задавать направление выполняемых операций (лево, право, первая, последняя, абсолютное, относительное положение и т.д.);
3. помечать столбцы как редактируемые, а другие как не редактируемые;
4. указывать области видимости (курсор может быть доступен только для запроса, посредством которого он был создан или для всех запросов);
5. создавать копию полученных данных путем указания СУБД (чтобы данные не изменялись в период работы курсора, открытием курсора и обращением к нему).

Курсоры используются в основном интерактивными приложениями, позволяющими прокручивать записи, менять их.

В Access не поддерживаются, в SQLite другой синтаксис. Курсоры бесполезны в веб-приложениях (ASP, ASP.Net, ColdFusion, PHP, Python, Ruby, JSP).

1. **Создание курсоров**

Прежде чем курсор может быть использован, его следует объявить(определить). При этом никакие данные не извлекаются.

После объявления курсор нужно открыть для получения данных- это извлечение строк согласно инструкции SELECT, заданной при объявлении.

После того, как курсор заполнен данными, из него могут быть извлечены необходимые строки.

Курсор должен быть закрыт (возможно освободить занимаемые им ресурсы - оператор DEALLOCATE удаляет курсор, выполняется автоматически SQL Server, когда курсор выходит из области действия).

Потом его можно открывать, закрывать, извлекать из него строки сколько нужно раз.

Определение курсора:

DECLARE CustCursor CURSOR

FOR

SELECT \* FROM STUDENT

WHERE STIPEND IS NULL;

Такой курсор является частью приложения, позволяющего отделению определять тех, кто не получает стипендию.

Могут использоваться в DB2, MariaDB, MySQL, SQL Server.

OPEN CURSOR CustCursor; - открытие курсора.

Выполняется запрос и полученные строки сохраняются для последующего просмотра.

**Для Oracle**

DECLARE TYPE CustCursor IS REF CURSOR

RETURN Customers&ROWTYPE;--извлечение первой строки

DECLARE CustRecord Customers&ROWTYPE

BEGIN

OPEN CustCursor;

FETCH CustCursor INTO CustRecord;--извлекает текущую (первую строку) и записывает ее в переменную CustRecord с полученными данными ничего не делается

CLOSE CustCursor;

DECLARE TYPE CustCursor is REF Cursor

RETURN Customers&ROWTYPE;

DECLARE CustRecord Customers&RowTYPE

BEGIN

OPEN CustCursor;

LOOP

FETCH CustCursor INTO CustRecord;

EXIT WHEN CustCursor %NOTFOUND;

…

END LOOP;

CLOSE CustCursor;

END;

Инструкция FETCH находится в цикле LOOP, поэтому она выполняется снова и снова.

Строка EXIT WHEN CustCursor %NOTFOUND; означает, что цикл должен быть завершен, когда больше не останется строк для извлечения.

Сам код обработки **…**.

**Для SQL Server**

DECLARE @cust\_id CHAR(10), @student\_id,

@cust\_name CHAR(50), @student\_name,

@cust\_adress CHAR(50), student\_city

OPEN CustCursor

FETCH NEXT FROM CustCursor

INTO @student\_id, @student\_name, @student\_city

WHILE @@FETCH\_STATUS=0

BEGIN

…

FETCH NEXT FROM CustCursor

INTO @student\_id, @student\_name, @student\_city

END

CLOSE CustCursor;

Здесь переменные объявляются для каждого извлекаемого столбца, а FETCH осуществляют выборку строк и сохраняют их значения в этих переменных.

Доступ к содержимому курсора можно получить с помощью инструкции-оператора FETCH.

Она задает, какие строки можно получить, откуда они должны быть извлечены и где их хранить. Оператор FETCH NEXT извлекает следующую строку из таблицы. FETCH NEXT употребляется чаще всего, иногда употребляют и другие формы PRIOR, FIRST, LAST.

FETCH ABSOLUTE извлечение строки по конкретному номеру с начала таблицы, FETCH RELATIVE извлечение строки по конкретному номеру с текущей строки в таблице.

Цикл WHILE нужен для последующей обработки каждой строки, извлечение данных из курсора происходит до тех пор, пока удовлетворяется заданное условие. Всякий раз при использовании FETCH внутренняя функция @@FETCH\_STATUS получает код состояния выборки, она возвращает код состояния 0, если выполнение FETCH завершится успешно, иначе отрицательный код состояния. Цикл WHILE продолжается до тех пор, пока удовлетворяется заданное в нем условие

WHILE @@FETCH\_STATUS=0. В теле цикла можно организовать обработку извлекаемых данных (после оператора BEGIN и перед следующим FETCH).

1. **Закрытие курсоров**

По окончании работы с курсорами их следует закрывать, в СУБД SQL Server требуется, чтобы ресурсы, занятые курсором были освобождены.

CLOSE CustCursor

DEALLOCATE CURSOR CustCursor—закрывает курсор

После того, как курсор закрыт, к нему нельзя обратиться, не открыв снова. Но его не нужно объявлять заново при повторном использовании, достаточно выполнить OPEN.