**Лабораторная работа № 14**

**Создание и управление курсорами**

Цель: Изучение назначения и типов курсоров, синтаксиса языка T – SQL для создания и открытия курсоров, выборки данных из курсора и изменения строк таблиц с помощью курсоров, удаления данных, закрытия и освобождения курсоров, а также приобретения навыков их применения и управления с помощью команд и системных хранимых процедур SQL Server.

Методические указания

Операции в реляционной базе данных выполняются над множеством строк. Например, набор строк, возвращаемый инструкцией SELECT, содержит все строки, которые удовлетворяют условиям, указанным в предложении WHERE инструкции. Такой полный набор строк, возвращаемых инструкцией, называется результирующим набором. Можно в результирующий набор не включать те или иные столбцы, применяя вертикальную фильтрацию. Результирующие наборы могут содержать сотни тысяч строк, и клиентские приложения не всегда справляются с таким объемом данных.

Приложения, особенно интерактивные, не всегда эффективно работают с результирующим набором как с единым целым. Им нужен механизм, позволяющий обрабатывать одну строку или небольшое их число за один раз. Для решения этой проблемы используются курсоры, которые представляют собой окна, налагаемые на результирующий набор данных и выделяющие требуемую часть данных. Курсоры являются расширением результирующих наборов, которые предоставляют такой механизм. Перемещая созданный курсор, можно получить доступ ко всем результирующим данным. Таким образом, курсоры SQL Server представляют собой механизм обмена данными между сервером и клиентом, который минимизирует ресурсы клиентского приложения.

Курсоры позволяют усовершенствовать обработку результатов:

позиционируясь на отдельные строки результирующего набора;

получая одну или несколько строк от текущей позиции в результирующем наборе;

поддерживая изменение данных в строках в текущей позиции результирующего набора;

поддерживая разные уровни видимости изменений, сделанных другими пользователями для данных, представленных в результирующем наборе;

предоставляя инструкциям Transact-SQL в скриптах, хранимых процедурах и триггерах доступ к данным результирующего набора.

Однако всегда, когда это возможно, следует избегать использования курсоров и применять команды SELECT, UPDATE, DELETE и INSERT.

Один курсор может базироваться на нескольких таблицах, расположенных в разных базах данных. Операция считывания определенных в курсоре данных называется выборкой (fetch). Если за одну операцию курсор позволяет выбрать несколько строк таблицы, то такой курсор называется блочным.

По способу просмотра данных курсоры бывают последовательные (forward only), которые обеспечивают просмотр строк только в одном направлении – от начала к концу, и прокручиваемые, которые допускают просмотр в обоих направлениях и переход к произвольной строке.

По представляемым возможностям курсоры делятся на четыре типа: статические, динамические, последовательные ключевые. Тип курсора определяется на стадии его создания и не может быть изменен.

Статический курсор (static cursor) называют также курсорами моментального снимка (snapshot cursor). При открытии такого курсора сервер выбирает все данные, соответствующие заданным критериям, и сохраняет результирующий набор строк в системной базе данных tempdb без изменения, если даже исходные строки и изменяются. Поэтому статический курсор всегда открывается в режиме “только для чтения”.

Динамические курсоры (dynamic cursor) противоположны статическим. При их использовании не создается полная копия исходных данных, а выполняется динамическая выборка данным из исходных таблиц только при обращении пользователя к тем или иным данным, при этом на время выборки соответствующие строки блокируются сервером. После выборки строк исходные строки могут изменяться пользователями, но эти изменения уже не отражаются в выбранных строках. С другой стороны, изменения в выбранных строках не будут видны другим пользователям, пока они не будут подтверждены (committed).

Последовательный курсор выбирает данные только от начала к его концу. Он не хранит результирующий набор. Строки считываются из базы данных, как только они выбираются в курсоре. Это позволяет отображать все изменения в базе данных. В курсоре видно самое последнее состояние данных.

Курсоры, зависящие от набора ключей (keyset-driven cursor), или ключевые курсоры, построены на основе уникальных идентификаторов. Множество всех уникальных идентификаторов (ключей) строк таблиц базы данных называется набором ключей. Сервер блокирует строки исходных таблиц только на время составления таблицы ключей. Ключевой курсор представляет набор ключей, идентифицирующих строки полного результирующего набора курсора. Набор ключей строится в системной базе данных tempdb.

При работе с курсорами можно выделить пять основных операций: создание курсора, открытие курсора, выборка из курсора и изменение строк данных с помощью курсора, закрытие курсора и освобождение курсора.

Синтаксис команды для создания курсора следующий:

DECLARE Имя курсора CURSOR [LOCAL/GLOBAL] [FORWARD\_ONLY\SCROLL] [STATIC\KEYSET\DINAMIC\FAST\_FORWARD] [READ\_ONLY\SCROLL\_LOCKS\OPTIMISTIC] [TYPE\_WARNING]

FOR select\_statement [FOR UPDATE [OF column\_name [,…n]]].

Для открытия курсора используется команда:

OPEN {{[GLOBAL] Имя курсора}\Имя переменной}.

Для выборки данных необходимо применять команду:

FETCH [[NEXT\PRIOR\FIRST\LAST\ABSOLUTE {n\@nvar}\ RELATIVE {n\@nvar} ] FROM ] {{[GLOBAL] Имя курсора}\Имя переменной} []INTO @ Имя переменной [,…n]]

Команды UPDATE, DELETE, CLOSE и DEALLOCATE позволят соответственно произвести изменение данных, удаление данных, закрытие и освобождение курсора.

Задания на лабораторную работу.

1. Создайте таблицу Clients со следующими атрибутами: Id, Name, Address.

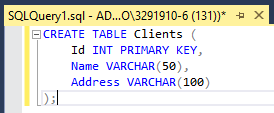


Рисунок 1 - создание таблицы

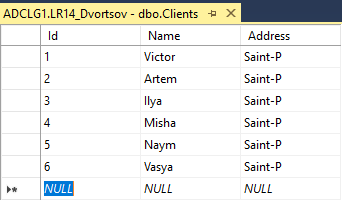


Рисунок 2 - заполнение таблицы данными

1. С помощью курсора создайте запрос, который будет выводить информацию о всех клиентах c нечетным Id.

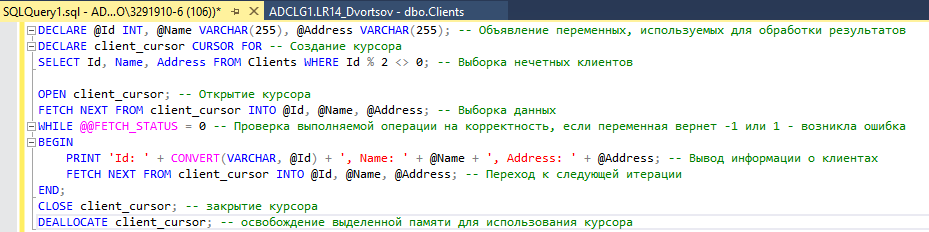


Рисунок 3 - создание курсора с выборкой данных

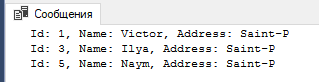


Рисунок 4 - вывод результата - id нечентых клиентов с их информацией

1. Создайте таблицу OlympMath с результатами Олимпиады по математике за 2019, 2020 и 2021 года. Таблица OlympMath состоит из следующих атрибутов: id, Name, res19, res20, res21.

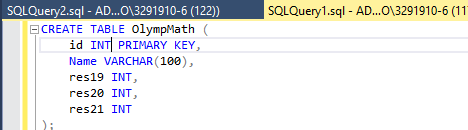


Рисунок 5 - создание таблицы

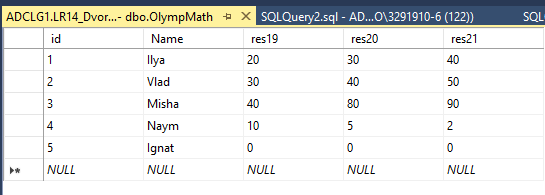


Рисунок 6 - заполнение таблицы данными

1. С помощью курсора создайте запрос, который будет находить и выводить лучший результат для каждого участника. То есть максимальный в строке.

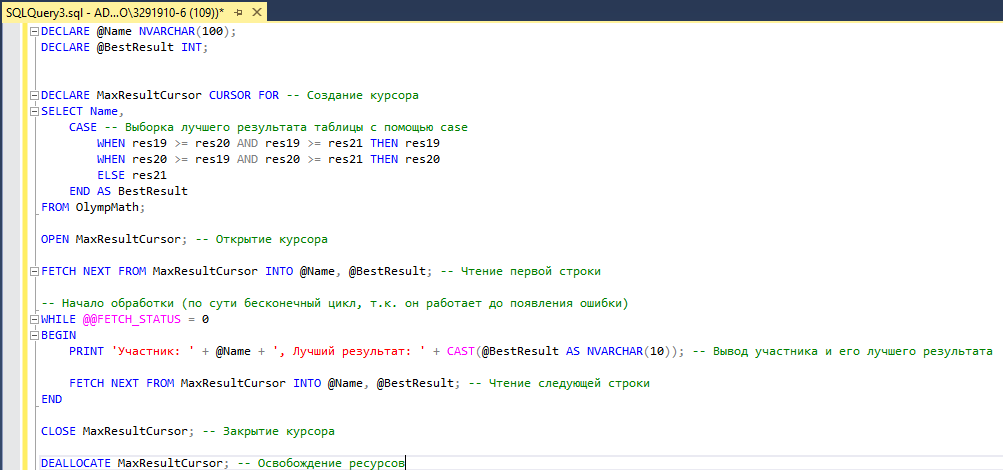


Рисунок 7 - код курсора

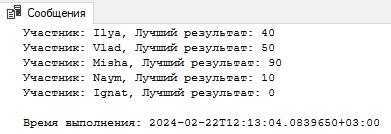
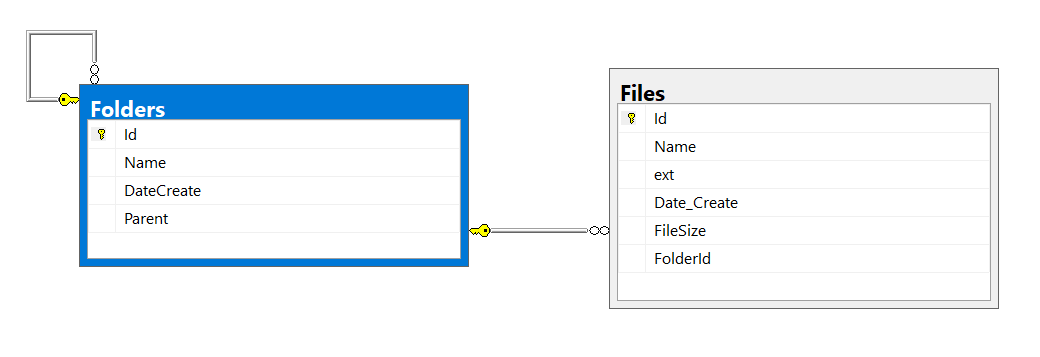


Рисунок 8 – результат

1. Создайте две таблицы со следующей структурой (папки и файлы):



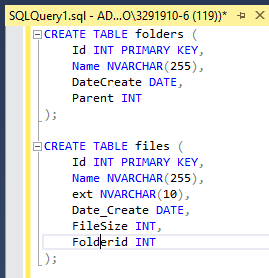


Рисунок 9 - создание 2-х таблиц - folders и files

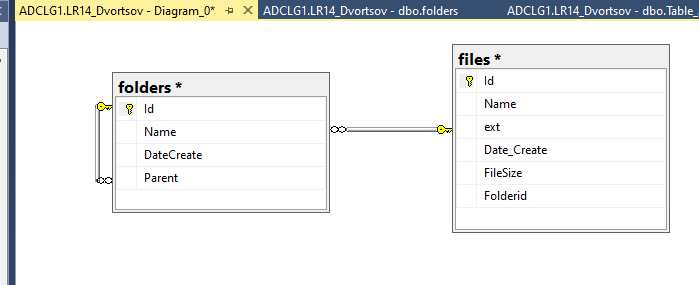


Рисунок 10 - добавление диаграммы

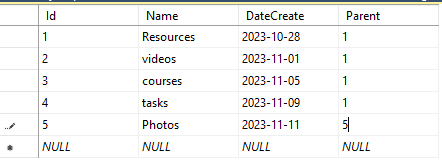


Рисунок 11 - заполнение таблицы folders

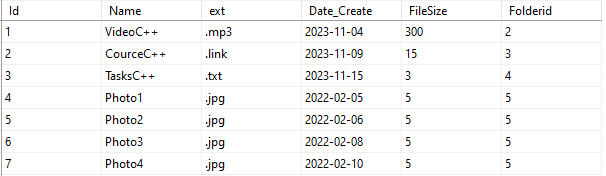


Рисунок 12 - заполнение таблицы files

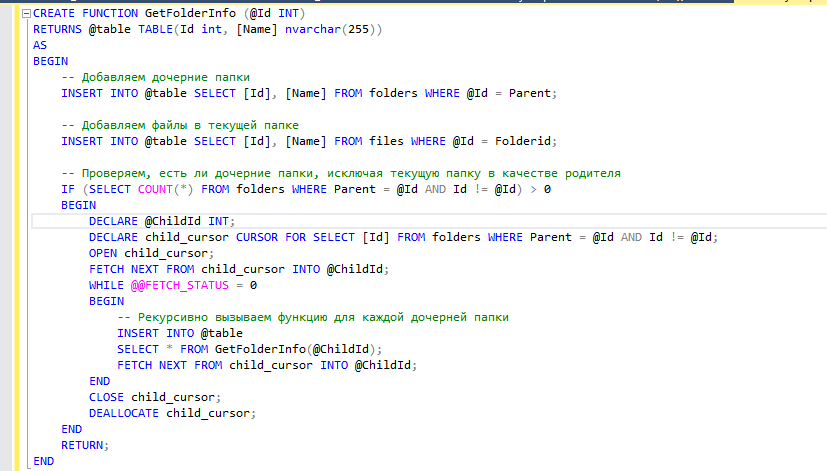


Рисунок 13 - код функции



Рисунок 14 - код запроса

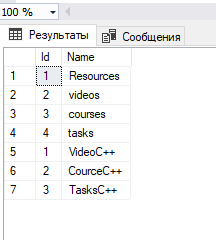


Рисунок 15 - результат