**Лабораторная работа № 10. Обработка результатов запросов с помощью курсоров**

**Вариант 12**

Цель: изучить курсоры, способы взаимодействия с ними и их виды; отработать навыки работы с курсорами.

Курсор является программной конструкцией, которая дает возможность пользователю обрабатывать строки результирующего набора запись за записью. Курсоры бывают локальные и глобальные (по умолчанию), статические и динамические (по умолчанию).

Курсор открывается с помощью оператора OPEN.

Оператор FETCH считывает одну строку из результирующего набора и продвигает указатель на следующую строку. Количество переменных в списке после ключевого слова INTO должно быть равно количеству столбцов результирующего набора, а порядок их должен соответствовать порядку перечисления столбцов в SELECT-списке.

После выполнения FETCH проверяется значение функции @@fetch\_status, которая возвращает значение 0, если оператор FETCH выполнен успешно; −1, если достигнут конец результирующего набора и строка не считывается; −2, если выбранная строка отсутствует в БД. В зависимости от полученного результата цикл продолжается и считывается следующая строка, или цикл заканчивается.

Курсор закрывается с помощью оператора CLOSE.

Первое задание ­ разработка запроса с использованием курсора – рисунок 1.1.

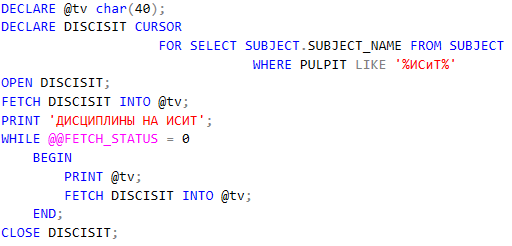


Рисунок 1.1 ­– Задание 1

Локальный курсор может применяться в рамках одного пакета и ресурсы, выделенные ему при объявлении, освобождаются сразу после завершения работы пакета.

Глобальный курсор может быть объявлен, открыт и использован в разных пакетах. Выделенные ему при объявлении ресурсы освобождаются только после выполнения оператора DEALLOCATE или при завершении сеанса пользователя.

Второе задание – разработка сценария, показывающего отличие локального курсора от глобального – рисунок 1.2.

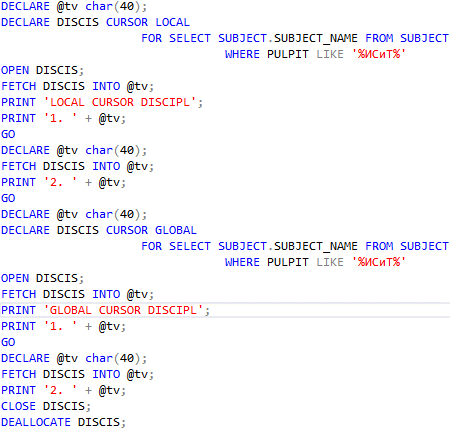


Рисунок 1.2 ­– Локальный и глобальный курсоры

А также результат выполнения данных курсоров – рисунок 1.3.

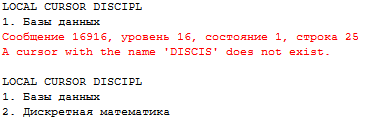


Рисунок 1.3 ­– Результат выполнения курсоров

Открытие статического курсора приводит к выгрузке результирующего набора во временную таблицу системной БД TEMPDB, и все дальнейшие операции осуществляются с этой таблицей.

После открытия курсора все текущие изменения в исходных таблицах не будут отражаться в результирующем наборе.

Значение функции @@CURSOR\_ROWS равно −n (количество записей) при асинхронной выборке, равно n при синхронной выборке, равно 0, если курсор не открыт. При выполнении курсора все изменения (UPDATE, DELETE и INSERT) в исходной таблице не отражаются на результате выборки строк.

Если заменить LOCAL STATIC на LOCAL DYNAMIC, то изменения будут отражаться в результирующем наборе.

Третье задание – разработка сценария, демонстрирующего отличие статических курсоров от динамических. Скрипт представлен на рисунке 1.4.

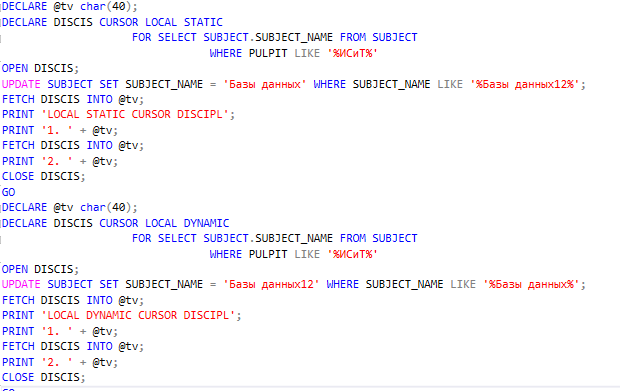


Рисунок 1.4 ­– Итоговый скрипт

Результат выполнения сценария третьего задания – рисунок 1.5.

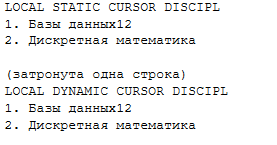


Рисунок 1.5 ­– Задание 3

По умолчанию для курсора установлен атрибут SCROLL, позволяющий применять оператор FETCH с дополнительными опциями позиционирования.

Перед каждым названием выводится номер строки результирующего набора, определяемый функцией ROW\_NUMBER(). Выбор строки определяется соответствующим ключевым словом в операторе FETCH.

ключевые слова FETCH: FIRST (первая строка), LAST(последняя строка), NEXT (следующая строка за текущей), PRIOR (предыдущая строка от текущей), ABSOLUTE 3 (третья строка от начала), ABSOLUTE −3 (третья строка от конца), RELATIVE 5 (пятая строка вперед от текущей), RELATIVE −5 (пятая строка назад от текущей).

В 4 задании – запрос со свойствами навигации в результирующем наборе – рисунок 1.6.

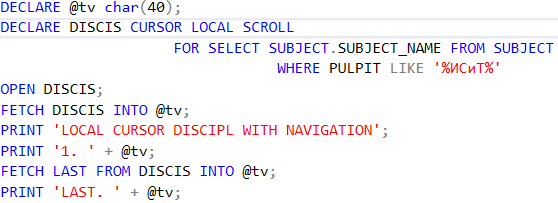


Рисунок 1.6 ­– Задание 4

Результат сценария четвертого задания – рисунок 1.7.



Рисунок 1.7 – Результат задания 4

Курсоры с установленным свойством FOR UPDATE помимо чтения данных из строк с помощью оператора FETCH, могут эти строки изменять или удалять с помощью операторов UPDATE и DELETE, если в секции WHERE эти операторы используют операцию CURRENT OF, для которой указывается имя курсора. Такой формат операторов позволяет удалять или изменять строки в таблице, соответствующих текущей позиции курсора в результирующем наборе.

Задание 5: создать курсор, демонстрирующий применение конструкции CURRENT OF в секции WHERE с использованием операторов UPDATE и DELETE - Рисунок 1.8.

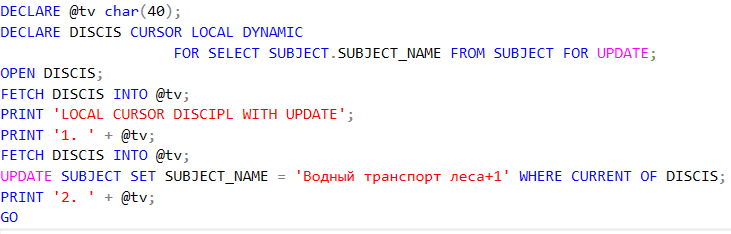


Рисунок 1.8 – Пятое задание

Шестое задание: Разработать SELECT-запрос, с помощью которого из таблицы PROGRESS удаляются строки, содержащие информацию о студентах, получивших оценки ниже 4 – рисунок 1.9.

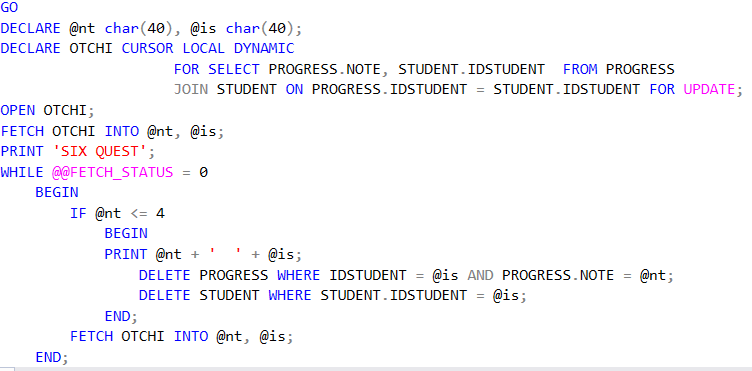


Рисунок 1.9 – Итоговый запрос

**Вывод**

В ходе лабораторной работы были изучены курсоры, а также способы взаимодействия с ними; были получены навыки по созданию, использованию и взаимодействию с курсорами; были изучены виды курсоров и их отличия.

.