**Тема 31. Программирование**

**1.Применение переменных.**

**2.Выполнение условных вычислений.**

**3.Циклическое повторение процессов.**

1. Иногда задачи оказываются более сложными и требуют применения нескольких операторов, условных вычислений и промежуточного манипулирования данными.

В T-SQL поддерживаются некоторые из основных принципов программирования и программных конструкций, чаще они применяются с хранимыми процедурами, триггерами и другими средствами БД.

Это применение переменных, выполнение условных вычислений, циклическое повторение процессов.

Правила и требования к переменным:

1. Имена всех переменных должны начинаться со знака @, глобальных -@@.
2. Перед тем, как пользоваться переменная должна быть объявлена с помощью DECLARE.
3. При объявлении указывается тип.
4. Несколько переменных можно объявить в отдельных операторах DECLARE или через запятую в одном.
5. Сделать переменные «необъявленными» нельзя. Локальные переменные существуют пока не завершится процесс.

Например

DECLARE @age int;

DECLARE @firstName char(20);

DECLARE @lastName char(20);

Когда переменные объявляются, то они не содержат никаких значений (т.е. NULL).

Чтобы присвоить им значения используют SET.

SET @lastName=’\_\_\_\_\_\_\_\_\_’;

SET @firstName=’\_\_\_\_\_\_\_\_\_’;

SET @age=\_\_\_;

Необходимо при инициализации установить либо значения по умолчанию, либо первоначальные значения.

Можно присваивать и значения с помощью SELECT, например

SELECT @age=\_\_\_;

SET или SELECT?

С помощью SET можно присвоить значение только одной переменной. SET записывают несколько раз для каждой переменной. А в одном SELECT можно присвоить несколько значений.

Например, выведем содержимое переменных для проверки:

SELECT @lastName, @firstName, @age;

SURNAME NAME 20

SELECT возвращает содержимое переменных, а не их имена. Можно в SELECT использовать AS, также оператор PRINT.

Например

PRINT @lastName+ ‘,’+@firstName,

PRINT @age;

SURNAME ,NAME

20

Чтобы вывести в одну строку:

PRINT ‘Age:’+Convert(char, @age);

Проверить содержимое переменных можно с помощью отладчика, в состав входит доступная для изучения документация на отладчик.

Допустим требуется выполнить два запроса:

1-получить сведения о конкретном заказчике,

2-возврат заказов по датам

SELECT cust\_name, cust\_email

FROM customers

WHERE cust\_id=10001;

SELECT order\_num,order\_date

FROM orders

WHERE cust\_id=10001

ORDER BY order\_date;

В таком случае для другого заказчика необходимо обновлять условие!

--объявим переменную @cust\_id

DECLARE @cust\_id INT;

SET @cust\_id=10001;

--Получить имя заказчика и адрес его почты

SELECT cust\_name, cust\_email

FROM customers

WHERE cust\_id=@cust\_id

--Получить историю его заказов

SELECT order\_num,order\_date

FROM orders

WHERE cust\_id=@cust\_id

ORDER BY order\_date;

Имена переменных не следует заключать в одиночные кавычки!

2. В T-SQL предоставляется возможность писать код с вариантами выбора дальнейших действий во время выполнения (IF).

Например, необходимо обновить данные заказа, если день рабочий и не обновлять , если воскресенье.

Функция

GetDate ()-текущая дата, время

DatePart (dw, GetDate());-текущий день недели

--Объявить переменные

DECLARE @open BIT;

--Сегодня рабочий день?

IF DatePart(dw, GetDate())=7

SET @Open=0;

ELSE

SET @Open=1;

--Вывести результат

SELECT @Open AS OpenForBusiness

OpenForBusiness

1

3. В SQL поддерживается организация циклов, способных неоднократно повторять блок кода по мере надобности.

Циклы организуются с помощью оператора WHILE. Часто оператор применяется в сочетании с курсорами.

Например

DECLARE @counter INT;

SET @counter =1;

WHILE @counter<=10

BEGIN

PRINT @counter;

SET @counter=@counter+1;

END

Результат:

1

2

3

4

5

6

7

8

9

10

BREAK –прерывание цикла, CONTINUE-перезапуск обработки с начала цикла.