

Практическая работа № 11

Программирование на SQL

1. Цель работы

1. Изучение переменных в T-SQL.
2. Изучение условных выражений.
3. Изучение циклов.

2. Теоретическая часть

Несмотря на то, что T-SQL – декларативный язык, у него есть расширение, позволяющее обрабатывать ошибки, создавать и выполнять хранимые процедуры и пользовательские функции, триггеры и сценарии с использованием локальных переменных, операторов присваивания, ветвлений и циклов.

Объявление переменной осуществляется с помощью оператора DECLARE. Упрощенный синтаксис команды имеет следующий вид:

```
DECLARE <@название> AS <тип>
```

Имена переменных в Transact-SQL начинаются с символа @.

Объявить сразу несколько переменных одним оператором DECLARE можно так:

```
DECLARE <@название1> AS <тип1>, ..., <@названиеN> AS <типN>
```

Ключевое слово AS необязательно.

При объявлении переменной можно ее инициализировать:

```
DECLARE <@название> AS <тип> = <значение>
```

Объявленным переменным можно присвоить различные значения с помощью оператора присваивания SET. Переменным должны присваиваться значения того типа данных, с каким они были объявлены. Упрощенный синтаксис команды имеет следующий вид:

```
SET <@название> = <значение>
```

Переменным можно присваивать скалярный результат выполнения запросов:

```
SET <@название> = (SELECT <значение> FROM <таблица>)
```

Неинициализированные переменные имеют значение NULL, их нельзя использовать в выражениях.

Переменным можно присваивать значения с помощью команды SELECT:

```
SELECT <@переменная1> = <столбец1>, ..., <@переменнаяN> = <столбецN> FROM  
<таблица>)
```

Значения переменных можно вывести с помощью команды PRINT. Синтаксис команды имеет следующий вид:

```
PRINT <сообщение>
```

Сообщение может быть символьной константой, переменной символьного типа, переменной, неявно преобразуемой в последовательность символов, или выражения, возвращающего символьный результат.

Значения переменных можно вывести с помощью команды SELECT. Синтаксис команды имеет следующий вид:

```
SELECT <@переменная1> [AS псевдоним1], ..., <@переменнаяN> [AS псевдонимN]
```

Для выполнения команды в зависимости от условия используется управляющая команда IF ... ELSE Инструкция, следующая за ключевым словом IF и его условием, выполняется только в том случае, если логическое выражение возвращает TRUE. Необязательное ключевое слово ELSE представляет другую инструкцию, которая выполняется, если условие IF не удовлетворяется и логическое выражение возвращает FALSE. Упрощенный синтаксис команды имеет следующий вид:

```
IF <условие>
[BEGIN]
<команды>
[END]
[ ELSE
[BEGIN]
<команды>
[END]
]
```

Условие должно возвращать только TRUE (ИСТИНА) или FALSE (ЛОЖЬ).

Если в блоке более чем одна команда, использование [BEGIN] ... [END] обязательно.

Для выполнения повторяющихся операций применяется цикл WHILE. Упрощенный синтаксис команды имеет следующий вид:

```
WHILE <условие>
[BEGIN]
<команды| BREAK | CONTINUE >
[END]
```

Команда BREAK приводит к выходу из цикла и вызывает инструкции, следующие за ключевым словом END, обозначающим конец цикла.

Команда CONTINUE пропускает все команды после себя до конца цикла и переводит цикл на следующий шаг.

3. Практическая часть

Таблица *Ученики*:

ИД	Фамилия	Предмет	Школа	Баллы
1	Иванова	Математика	Лицей	98,5
2	Петров	Физика	Лицей	99
3	Сидоров	Математика	Лицей	88
4	Полухина	Физика	Гимназия	78
5	Матвеева	Химия	Лицей	92
6	Касимов	Химия	Гимназия	68
7	Нурулин	Математика	Гимназия	81
8	Авдеев	Физика	Лицей	87
9	Никитина	Химия	Лицей	94
10	Барышева	Химия	Лицей	88

Пример 1: Даны числа а и b. Найти и вывести их сумму:

```
DECLARE @a INT, @b INT, @c INT
SET @a = 5
SET @b = 10
SET @c = @a + @b
PRINT @c
```

Пример 2: В таблице «Ученики» найти разницу между наибольшими баллами среди лицейстов и гимназистов:

```
DECLARE @licey FLOAT, @gimn FLOAT, @diff FLOAT
SET @licey = (
    SELECT
        MAX(Баллы)
    FROM
        Ученики
    WHERE
        Школа = 'Лицей'
)
SET @gimn = (
    SELECT
        MAX(Баллы)
    FROM
        Ученики
    WHERE
        Школа = 'Гимназия'
)
SET @diff = ABS(@licey - @gimn)
PRINT @diff
```

Пример 3: В таблице «Ученики» найти разницу между наибольшими и наименьшими баллами:

```
DECLARE @maxp FLOAT, @minp FLOAT, @diff FLOAT
SELECT
    @maxp = MAX(Баллы),
    @minp = MIN(Баллы)
FROM
    Ученики
SET @diff = @maxp - @minp
PRINT @diff
```

Пример 4: Дано случайное целое число меньше 1000. Вывести его квадрат:

```
DECLARE @a INT = RAND() * 1000, @b INT
SET @b = SQUARE(@a)
PRINT @b
```

Пример 5: Даны случайные целые числа a и b. Найти наибольшие из них:

```
DECLARE @a INT = RAND() * 100, @b INT = RAND() * 100
IF @a > @b
    PRINT '@a = ' + CAST(@a AS VARCHAR(3))
ELSE
    PRINT '@b = ' + CAST(@b AS VARCHAR(3))
```

Пример 6: Дано случайное целое число a. Проверить, делится ли данное число на 3:

```
DECLARE @a INT = RAND() * 100
IF @a % 3 = 0
    PRINT CAST(@a AS VARCHAR(3)) + ' делится на 3'
ELSE
    PRINT CAST(@a AS VARCHAR(3)) + ' не делится на 3'
```

Пример 7: Дано случайное целое число N ($N < 1000$). Если оно является степенью числа 5, то вывести «Да», если не является – вывести «Нет»:

```
DECLARE @a INT = RAND() * 1000
WHILE @a % 5 = 0
    SET @a = @a / 5
IF @a = 1
    PRINT 'Да'
ELSE
    PRINT 'Нет'
```

Пример 8: Даны случайные целые числа а и b. Найти наибольший общий делитель (НОД):

```
DECLARE @a INT = RAND() * 1000, @b INT = RAND() * 1000
PRINT '@a = ' + CAST(@a AS VARCHAR(4))
PRINT '@b = ' + CAST(@b AS VARCHAR(4))

WHILE @a != @b
BEGIN
    IF @a > @b
        SET @a = @a - @b
    ELSE
        SET @b = @b - @a
END
PRINT 'НОД = ' + CAST(@a AS VARCHAR(4))
```

Пример 9: Даны два целых числа А и В ($A < B$). Найти сумму всех целых чисел от А до В включительно:

```
DECLARE @a INT = 5, @b INT = 10, @s INT = 0
WHILE @a <= @b
BEGIN
    SET @s = @s + @a
    SET @a = @a + 1
END
PRINT 'Сумма = ' + CAST(@s AS VARCHAR(5))
```

Пример 10: Дано случайное целое число N ($N < 100$). Найти квадрат данного числа, используя для его вычисления следующую формулу:

$$N^2 = 1 + 3 + 5 + \dots + (2 \cdot N - 1)$$

После добавления к сумме каждого слагаемого выводить текущее значение суммы (в результате будут выведены квадраты всех целых чисел от 1 до N):

```
DECLARE @N INT = RAND() * 10, @M INT = 1, @S INT = 0
WHILE @M <= 2 * @N - 1
BEGIN
    SET @S = @S + @M
    PRINT @S
    SET @M = @M + 2
END
```

Пример 11: Даны случайные целые числа A и B ($A < B$). Вывести все целые числа от A до B включительно; при этом число A должно выводиться 1 раз, число $A + 1$ должно выводиться 2 раза и т.д.:

```
DECLARE @A INT = RAND() * 5, @C INT = 1
DECLARE @B INT = @A + RAND() * 5
PRINT '@A = ' + CAST(@A AS CHAR(1)) + ', @B = ' + CAST(@B AS CHAR(1))
WHILE @A <= @B
BEGIN
    PRINT REPLICATE(@A, @C)
    SET @A = @A + 1
    SET @C = @C + 1
END
```

Пример 12: Напечатать те из двузначных чисел, которые делятся на 4, но не делятся на 6:

```
DECLARE @A INT = 10
WHILE @A < 100
BEGIN
    IF (@A % 4 = 0) AND (@A % 6 != 0)
        PRINT @A
    SET @A = @A + 1
END
```

Пример 13: Даны два целых числа D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, следующей за указанной:

```
DECLARE @D INT = 31, @M INT = 12
SET @D = CASE
    WHEN @M IN (1, 3, 5, 7, 8, 10, 12) AND @D = 31 THEN 1
    WHEN @M IN (4, 6, 9, 11) AND @D = 30 THEN 1
    WHEN @M = 2 AND @D = 29 THEN 1
    ELSE @D + 1
END
SET @M = CASE
    WHEN @D = 1 AND @M = 12 THEN 1
    WHEN @D = 1 AND @M < 12 THEN @M + 1
    ELSE @M
END

PRINT CAST(@D AS VARCHAR(2)) + '/' + CAST(@M AS VARCHAR(2))
```

Пример 14: Вывести слово «Нижевартовск» на экран столько раз, сколько в нем букв:

```
DECLARE @L INT, @N CHAR(13) = 'Нижевартовск'
SET @L = LEN(@N)
```

```

WHILE @L > 0
BEGIN
    PRINT @N
    SET @L = @L - 1
END

```

Пример 15: Напишите код для вывода на экран с помощью цикла:

```

НишневартовскксвотравенжиН
Нишневартовс свотравенжиН
Нишневартов  вотравенжиН
Нишневарто   отравенжиН
Нишневарт    травенжиН
Нишневар     равенжиН
Нишнева      авенжиН
Нишнев       венжиН
Нишне        енжиН
Нижн         нжиН
Ниж          жиН
Ни           иН
Н           Н
Ни          иН
Ниж         жиН
Нижн        нжиН
Нишне       енжиН
Нишнев      венжиН
Нишнева     авенжиН
Нишневар    равенжиН
Нишневарт   травенжиН
Нишневарто  отравенжиН
Нишневартов вотравенжиН
Нишневартовс свотравенжиН
НишневартовскксвотравенжиН

```

```

DECLARE @L INT, @M INT, @N CHAR(13)
SET @N = 'Нишневартовск'
SET @L = LEN(@N)
SET @M = @L
WHILE @L > 0
BEGIN
    PRINT LEFT(@N, @L) + SPACE(2 * (@M - @L)) + RIGHT(REVERSE(@N), @L)
    SET @L = @L - 1

```

```

END
SET @L = 2
WHILE @L <= @M
BEGIN
    PRINT LEFT(@N, @L) + SPACE(2 * (@M - @L)) + RIGHT(REVERSE(@N), @L)
    SET @L = @L + 1
END

```

4. Задание

1. Даны числа А и В. Найти и вывести их произведение.
2. В таблице «Ученики» найти разницу между средними баллами лицеистов и гимназистов.
3. В таблице «Ученики» проверить на четность количество строк.
4. Дано четырехзначное число. Вывести сумму его цифр.
5. Даны случайные целые числа а, b и с. Найти наименьшее из них.
6. Дано случайное целое число а. Проверить, делится ли данное число на 11.
7. Дано случайное целое число N ($N < 1000$). Если оно является степенью числа 3, то вывести «Да», если не является – вывести «Нет».
8. Даны случайные целые числа а и b. Найти наименьший общий кратный (НОК).
9. Даны два целых числа А и В ($A < B$). Найти сумму квадратов всех целых чисел от А до В включительно.
10. Найти первое натуральное число, которое при делении на 2, 3, 4, 5, и 6 дает остаток 1, но делится на 7.
11. Вывести свою фамилию на экран столько раз, сколько в нем букв.
12. Напишите код для вывода на экран с помощью цикла:

```

Н
иНи
жиНиж
нжиНижн
енжиНижне
венжиНижнев
авенжиНижнева
равенжиНижневар
травенжиНижневарт
отравенжиНижневарто
вотравенжиНижневартов
свотравенжиНижневартовс
ксвотравенжиНижневартовск

```


