

Лабораторная работа 7.1 Программирование на SQL

1. Теоретическая часть

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.

2. Практическая часть

Создайте в тестовой БД таблицу **Table_uch1**

id	fo	predmet	ush	ball
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	Нестеров	математика	гимназия	81
8	Авдеев	физика	лицей	87
9	Никитина	химия	лицей	94
10	Барышев	химия	лицей	88

1) Даны числа а и b. Найти и вывести их сумму

```
4
3 DECLARE @a INT, @b INT, @c INT
4 SET @a = 5
5 SET @b = 10
6 SET @c = @a + @b PRINT @c
7
```

2 %
Сообщения
15

2) В таблице Table_uch1 найти разницу между наибольшими баллами среди лицейстов и гимназистов

```
7
8
9 --разница между наибольшими баллами среди лицейстов и гимназистов
10 DECLARE @licey FLOAT, @gimn FLOAT, @diff FLOAT
11
12 SET @licey = ( SELECT MAX(ball) From Table_uch1
13 WHERE ush = 'Лицей')
14
15 SET @gimn =( SELECT MAX(ball) From Table_uch1
16 WHERE ush = 'Гимназия')
17
18 SET @diff = ABS(@licey - @gimn)
19 PRINT @diff
20
```

2 %
Сообщения
18

3) Найти разницу между наибольшими и наименьшими баллам

```
20
21 --найти разницу между наибольшими и наименьшими баллами
22 DECLARE @maxp FLOAT, @minp FLOAT, @diff FLOAT
23 SET
24     @maxp = MAX(ball),
25     @minp = MIN(ball)
26 FROM
27     Table_uch1
28 SET @diff = @maxp - @minp
29 PRINT @diff
30
```

%
Сообщения
31

4) Дано случайное целое число меньше 1000, вывести его квадрат

```
31 --Дано случайное целое число меньше 1000, вывести его квадрат
32 DECLARE @a1 INT, @b1 INT
33 SET @a1=RAND()*1000
34 SET @b1 = SQUARE(@a1)
35 PRINT @a1
36 PRINT @b1
37
```

%
Сообщения
769
591361

5) Даны 2 случайных целых числа. Найти наибольшие из них

```
38 --Даны 2 случайных целых числа. Найти наибольшие из них
39
40 DECLARE @a2 INT=RAND() * 100, @b2 INT= RAND() * 100
41
42 IF @a2 > @b2
43     PRINT '@a2 = ' + CAST(@a2 AS VARCHAR(3))
44 ELSE
45     PRINT '@b2 = ' + CAST(@b2 AS VARCHAR(3))
46
```

Сообщения
@a2 = 53

Функция *CAST* преобразует выражение одного типа к другому. Она имеет следующую форму: *CAST(выражение AS тип_данных)*

6) Дано случайное целое число. Проверить, делится ли данное число на 3

```
46
47 --Дано случайное целое число. Проверить, делится ли данное число на 3
48 DECLARE @a3 INT = RAND() * 100
49 IF @a3 % 3 = 0
50     PRINT CAST(@a3 AS VARCHAR(3)) + ' делится на 3'
51 ELSE
52     PRINT CAST(@a3 AS VARCHAR(3)) + ' не делится на 3'
53
54
```

Сообщения
72 делится на 3

```
46
47 --Дано случайное целое число. Проверить, делится ли данное число на 3
48 DECLARE @a3 INT = RAND() * 100
49 IF @a3 % 3 = 0
50     PRINT CAST(@a3 AS VARCHAR(3)) + ' делится на 3'
51 ELSE
52     PRINT CAST(@a3 AS VARCHAR(3)) + ' не делится на 3'
53
54
```

Сообщения
32 не делится на 3

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

```
54 --Дано случайное целое число N (N < 1000).
55 --Если оно является степенью числа 5, то вывести «Да»,
56 --если не является – вывести «Нет»
57
58 DECLARE @a4 INT = RAND() * 1000
59
60 WHILE @a4 % 5 = 0
61     SET @a4 = @a4 / 5
62 IF @a4 = 1
63     PRINT 'Да'
64
65 ELSE
66     PRINT 'Нет'
67
```

8. Даны случайные целые числа а и b. Найти наибольший общий делитель

(НОД)

VARCHAR - это строковый или символьный тип данных переменного размера. Вы можете хранить тут буквы, числа и символы. Начиная с SQL Server 2019, вы можете использовать полный диапазон символов Unicode при использовании колляции с поддержкой UTF-8. Вы можете объявить или переменные этого типа, используя **VARCHAR(n)**, где n обозначает размер строки в байтах. n меняется в диапазоне от 1 до 8000.

```
67
68 --Даны случайные целые числа а и b.
69 --Найти наибольший общий делитель (НОД)
70 DECLARE @a5 INT = RAND() * 1000, @b5 INT = RAND() * 1000
71 PRINT '@a5 = ' + CAST(@a5 AS VARCHAR(4))
72 PRINT '@b5 = ' + CAST(@b5 AS VARCHAR(4))
73
74 WHILE @a5 != @b5
75 BEGIN
76     IF @a5 > @b5
77     SET @a5 = @a5 - @b5 ELSE
78     SET @b5 = @b5 - @a5
79 END
80 PRINT 'НОД = ' + CAST(@a5 AS VARCHAR(4))
81
```

152 %

Сообщения

@a5 = 462
@b5 = 699
НОД = 3

Время выполнения: 2023-04-16T09:55:32.2938889+03:00

9. Даны два целых числа А и В (А < В). Найти сумму всех целых чисел от А до

В включительно

```
82
83 --Даны два целых числа А и В (А < В).
84 --Найти сумму всех целых чисел от А до В
85 DECLARE @a6 INT = 5, @b6 INT = 10, @s INT = 0
86 WHILE @a6 <= @b6
87 BEGIN
88     SET @s = @s + @a6
89     SET @a6 = @a6 + 1
90 END
91 PRINT 'Сумма = ' + CAST(@s AS VARCHAR(5))
92
93
```

6

Сообщения

Сумма = 45

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):

```
--Дано случайное целое число N (N < 100). Найти квадрат данного числа
--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, @S1 INT = 0
WHILE @M <= 2 * @N - 1
BEGIN
    SET @S1 = @S1 + @M
    PRINT @S1
    SET @M = @M + 2
END
```

%

Сообщения

1
4
9
16
25
36
49
64

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.

Функция *REPLICATE* (Transact-SQL)

Применено к: SQL Server Azure SQL Database Управляемый экземпляр SQL

Повторяет значение строки указанное число раз.

```
--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 @A7 INT = RAND() * 5, @C7 INT = 1
DECLARE @B7 INT = @A7 + RAND() * 5
PRINT '@A7 = ' + CAST(@A7 AS CHAR(1)) + ', @B7 = ' + CAST(@B7 AS CHAR(1))
WHILE @A7 <= @B7
BEGIN
    PRINT REPLICATE(@A7, @C7)
    SET @A7 = @A7 + 1
    SET @C7 = @C7 + 1
END
```

152 %

Сообщения

@A7 = 3, @B7 = 6
3
44
555
6666

Время выполнения: 2023-04-16T10:16:07.5078426+03:00

11. Напечатать те из двузначных чисел, которые делятся на 4, но не делятся на 6

```

117  --Напечатать те из двузначных чисел, которые делятся на 4, но не делятся на 6
118
119  DECLARE @A8 INT = 10
120  WHILE @A8 < 100
121  BEGIN
122      IF (@A8 % 4 = 0) AND (@A8 % 6 != 0)
123      PRINT @A8
124      SET @A8 = @A8 + 1
125  END
126

```

152 %

Сообщения

16
20
28
32
40
44
52
56
64
68
76
80
88
92

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

```

125 | END
126 | --Даны два целых числа D (день) и M (месяц), определяющие правильную дату невисокосного года.
127 | --Вывести значения D и M для даты, следующей за указанной
128 | DECLARE @D INT = 31, @M INT = 12
129 | SET @D = CASE
130 |     WHEN @M IN (1, 3, 5, 7, 8, 10, 12) AND @D = 31 THEN 1
131 |     WHEN @M IN (4, 6, 9, 11) AND @D = 30 THEN 1
132 |     WHEN @M = 2 AND @D = 29 THEN 1
133 |     ELSE @D + 1
134 | END
135 | SET @M = CASE
136 |     WHEN @D = 1 AND @M = 12 THEN 1
137 |     WHEN @D = 1 AND @M < 12 THEN @M + 1
138 |     ELSE @M
139 | END
140 | PRINT CAST(@D AS VARCHAR(2)) + '/' + CAST(@M AS VARCHAR(2))
141 |

```

13. Вывести слово «Нижевартовск» на экран столько раз, сколько в нем букв

[illegible]

15. Напишите код для вывода на экран с помощью цикла

ФУНКЦИЯ *REVERSE* (Transact-SQL)

Возвращает строковое значение, где символы переставлены в обратном порядке справа налево.

Синтаксис

REVERSE (*string_expression*)

LEFT(строка, число) возвращает с начала строки определенное количество символов

RIGHT(строка, число) возвращает с конца строки определенное количество символов

SPACE: возвращает строку, которая содержит определенное количество пробелов

Напишите код для вывода на экран с помощью цикла:

НижневартовсксвотравенжиН
Нижневартовс свотравенжиН
Нижневартов вотравенжиН
Нижневарто отравенжиН
Нижневарт травенжиН
Нижневар равенжиН
Нижнева авенжиН
Нижнев венжиН
Нижне енжиН
Нижн нжиН
Ниж жиН
Ни
иН Н
Н
Ни иН
Ниж жиН
Нижн нжиН
Нижне енжиН
Нижнев венжиН
Нижнева авенжиН
Нижневар равенжиН
Нижневарт травенжиН
Нижневарто отравенжиН
Нижневартов вотравенжиН
Нижневартовс свотравенжиН
НижневартовсксвотравенжиН

```
--  
151 --Напишите код для вывода на экран с помощью цикла  
152 DECLARE @L2 INT, @M2 INT, @N2 CHAR(13)  
153 SET @N2 = 'Нижневартовск'  
154 SET @L2 = LEN(@N2)  
155 SET @M2 = @L2  
156 WHILE @L2 > 0  
157 BEGIN  
158     PRINT LEFT(@N2, @L2) + SPACE(2 * (@M2 - @L2)) + RIGHT(REVERSE(@N2), @L2)    SET @L2 = @L2 - 1  
159 END  
160 SET @L2 = 2  
161 WHILE @L2 <= @M2  
162 BEGIN  
163     PRINT LEFT(@N2, @L2) + SPACE(2 * (@M2 - @L2)) + RIGHT(REVERSE(@N2), @L2)  
164     SET @L2 = @L2 + 1  
165 END  
166
```

152 %

Сообщения

Нижневарто отравенжиН
Нижневарт травенжиН
Нижневар равенжиН
Нижнева авенжиН
Нижнев венжиН
Нижне енжиН
Нижн нжиН
Ниж жиН
Ни иН
иН Н
Н
Ни иН
Ниж жиН
Нижн нжиН
Нижне енжиН
Нижнев венжиН
Нижнева авенжиН
Нижневар равенжиН
Нижневарт травенжиН
Нижневарто отравенжиН
Нижневартов вотравенжиН

Самостоятельная работа

1. Даны числа A и B . Найти и вывести их произведение.
2. В таблице «Ученики» найти разницу между средними баллами лицеев и гимназистов.
3. В таблице «Ученики» проверить на четность количество строк.
4. Дано четырехзначное число. Вывести сумму его цифр.
5. Даны случайные целые числа a , b и c . Найти наименьшее из них.
6. Дано случайное целое число a . 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 a и b . Найти наименьший общий кратный (НОК).
9. Даны два целых числа A и B ($A < B$). Найти сумму квадратов всех целых чисел от A до B 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.