Лабораторный практикум

МДК 11.01 Технология разработки и защиты баз данных

SQL

Куропаткина О.П.

ЛАБОРАТОРНАЯ РАБОТА №9

ИНСТРУКЦИИ IF..ELSE, EXISTS, CASE, WHILE

Блок инструкций

Блок инструкций может состоять из одной или нескольких инструкций языка Transact-SQL. Каждый блок начинается с инструкции BEGIN и заканчивается инструкцией END, как это показано далее:

```
BEGIN
    statement_1
    statement_1
    ...
END
```

Блок можно разместить внутри инструкции IF, чтобы в зависимости от определенного условия разрешить исполнение одной или нескольких инструкций.

Инструкция IF

Инструкция IF языка Transact-SQL соответствует одноименной инструкции, поддерживаемой почти всеми языками программирования. Инструкция IF выполняет одну или несколько составляющих блок инструкций, если логическое выражение, следующее после ключевого слова IF, возвращает значение TRUE (истина). Если же инструкция IF содержит оператор ELSE, то при условии, что логическое выражение возвращает значение FALSE (ложь), выполняется вторая группа инструкций.

```
--------------------------------------------------------------------------
--                    Условная конструкция IF..ELSE
--------------------------------------------------------------------------

DECLARE @myVar nvarchar(20);

SET @myVar = 'Hello World!!!'

-- В условной конструкции IF, указываем выражение: @myVar is NULL
IF @myVar is NULL
    PRINT ('значение myVar не задано');    -- Если условие удовлетворяет истинности.
ELSE
    PRINT @myVar;                          -- ИНАЧЕ Если условие не удовлетворяет истинности.
```

108 %

Сообщения

```
Hello World!!!
```

Инструкция EXISTS

Операция EXISTS возвращает значение TRUE или FALSE, в зависимости от того, имеется ли соответствующие запросу SELECT данные.

```
19  ------------------------------------------------------------------------
20  --                          Операция EXISTS
21  ------------------------------------------------------------------------
22  -- Операция EXISTS возвращает значение TRUE или FALSE, в зависимости от того
23  -- имеется ли соответствующие запросу SELECT данные
24  USE ForEducationSQL
25  GO
26
27  IF EXISTS (SELECT * FROM Product WHERE Cost > 5000)
28      PRINT 'В базе данных имеются товары дороже 5000 руб.'
29  ELSE
30      PRINT 'Все товары базы данных дешевле 5000 руб.'
31
```

108 %

Сообщения

```
Все товары базы данных дешевле 5000 руб.
```

Инструкция CASE

Оценка списка условий и возвращение одного из нескольких возможных выражений результатов.

Выражение CASE имеет два формата:

- простое выражение CASE для определения результата сравнивает выражение с набором простых выражений;
- поисковое выражение CASE для определения результата вычисляет набор логических выражений.

Оба формата поддерживают дополнительный аргумент ELSE.

```
96  ------------------------------------------------------------------------
97  --                          Оператор CASE
98  ------------------------------------------------------------------------
99  -- Простой оператор CASE
100
101 DECLARE @myTinyVar TinyInt =3 ;
102
103 PRINT CASE @myTinyVar          -- входное выражение для CASE
104         WHEN 0 THEN 'zero'     -- если @myIntVar = 0 то выводим 'zero'
105         WHEN 1 THEN 'One'      -- если @myIntVar = 1 то выводим 'One'
106         WHEN 2 THEN 'Two'      -- если @myIntVar = 2 то выводим 'Two'
107         WHEN 3 THEN 'Three'    -- если @myIntVar = 3 то выводим 'Three'
108         ELSE 'More than three' -- если не сработало ни одно из условий то выводим 'More than three'
109       END                      --конец оператора CASE
110 GO
```

108 %

Сообщения

```
Three
```

```sql
--------------------------------------------------------------------------------
-- Поисковый оператор CASE
DECLARE @MyIntVar int;

SET @MyIntVar = 0;

PRINT CASE -- отсутствует входное выражение
        WHEN @MyIntVar IS NULL THEN 'переменная @MyIntVar пустая'  -- выражение в конструкции WHEN должно принимать булево значение
        WHEN @MyIntVar < 0 THEN 'переменная @MyIntVar меньше нуля'
        WHEN @MyIntVar > 0 THEN 'переменная @MyIntVar больше нуля'
        WHEN @MyIntVar > 3 THEN 'переменная @MyIntVar больше трех' -- данная строка никогда не будет выполнятся,
        -- так как на предыдущей строке была проверка на условиее больше нуля, а любое значение больше трех тоже больше нуля
        ELSE 'Непредвиденная ситуация'
      END
GO
```

```
108 %
Сообщения
Непредвиденная ситуация
```

Инструкция WHILE

Инструкция WHILE выполняет одну или несколько заключенных в блок инструкций, на протяжении времени, пока (while) логическое выражение возвращает значение TRUE (истина). Иными словами, если выражение возвращает TRUE, выполняется инструкция или блок инструкций, после чего снова осуществляется проверка выражения. Этот процесс повторяется до тех пор, пока выражение не возвратит значение FALSE (ложь).

```sql
--------------------------------------------------------------------------------
-- Оператор WHILE. Организация циклов

DECLARE @myVar int;
SET @myVar = 0;

    WHILE (@myVar < 11) -- условие выполнения цикла, пока условие истинно выполняется цикл.
    BEGIN
        PRINT 'Текущее Значение ' + CAST (@myVar as varchar);
        SET @myVar = @myVar + 1;
    END
 GO
 --------------------------------------------------------------------------------
```

```
108 %
Сообщения
Текущее Значение 0
Текущее Значение 1
Текущее Значение 2
Текущее Значение 3
Текущее Значение 4
Текущее Значение 5
Текущее Значение 6
Текущее Значение 7
Текущее Значение 8
Текущее Значение 9
Текущее Значение 10
```

Блок внутри инструкции WHILE может содержать одну или две необязательных инструкций, применяемых для управления выполнением инструкций внутри блока: BREAK или CONTINUE.

Инструкция CONTINUE останавливает выполнение только текущей инструкции в блоке и начинает выполнять его с самого начала.

```
142 DECLARE @myVar int;
143 SET @myVar =0;
144
145     WHILE @myVar < 11
146     BEGIN
147         PRINT 'Текущее Значение ' + CAST (@myVar as varchar);
148         IF @myVar =5
149             BEGIN
150                 SET @myVar = @myVar + 2;
151                 CONTINUE; -- Прерывает дальнейшее выполнение текущей итерации и возвращается в начало цикла WHILE
152             END;
153         SET @myVar = @myVar + 1;
154     END
155 GO
```

108 %

Сообщения

```
Текущее Значение 0
Текущее Значение 1
Текущее Значение 2
Текущее Значение 3
Текущее Значение 4
Текущее Значение 5
Текущее Значение 7
Текущее Значение 8
Текущее Значение 9
Текущее Значение 10
```

Инструкция BREAK останавливает выполнение инструкций внутри блока и начинает исполнение инструкций, следующих сразу же после этого блока.

```
158 DECLARE @myVar int;
159 SET @myVar = 0;
160
161     WHILE @myVar < 21
162     BEGIN
163         PRINT 'Текущее Значение ' + CAST (@myVar as varchar);
164         IF @myVar = 7
165         BEGIN
166             PRINT '@myVar = 7! Прерывание цикла!'
167             BREAK; -- Оператор прерывания цикла (не рекомендуется использовать)
168         END
169         SET @myVar = @myVar + 1;
170     END
171 GO
```

108 %

Сообщения

```
Текущее Значение 0
Текущее Значение 1
Текущее Значение 2
Текущее Значение 3
Текущее Значение 4
Текущее Значение 5
Текущее Значение 6
Текущее Значение 7
@myVar = 7! Прерывание цикла!
```

ДАЛЬШЕ САМИ:

1. **Если в базе данных есть клиенты, у которых день рождения в текущем месяце, выведите сообщение «В текущем месяце есть именинники среди клиентов компании!». В противном случае, выведите соответствующее сообщение.**
2. **Дополните предыдущие условие и выведите <u>количество</u> именинников в текущем месяце.**

3. **Если общая сумма от продаж товаров (количество проданного товара * стоимость) менее 200 000 руб., вывести сообщение «Выручка составила – 'Сумма продаж'. Нужно больше продаж!». Иначе - «Выручка составила – 'Сумма продаж'. Не плохо, но могли бы и лучше!».**
4. **Выведите список товаров: Название, Стоимость, Актуальность (Если атрибут IsActive = TRUE, то вывести «Актуален», если FALSE – «Не для продажи)**
5. **Выведите список услуг: Название, Стоимость с учетом скидки, Длительность (Если услуги длится менее 60 минут - Непродолжительная услуга, от 60 до 120 минут – Услуга средней длительности, более 120 минут – Длительная услуга), Длительность услуги в минутах. Отсортируйте список по стоимости.**
6. **Выведите список клиентов: Фамилия И.О., Пол, Возраст, Примечание (если клиент записан более 3ех раз на услугу – «Постоянный клиент», от 1 до 3ех – «Иногда заглядывает», 0 – «Ни одной услуги не оказано. Как так?»**

Не забывайте, что у всех атрибутов должны быть наименования.

7. **С использованием цикла добавьте в таблицу «Категория товара» 10 строк по шаблону: Категория 1, Категория 2 … Категория 10.**
8. **Рассчитайте среднюю стоимость товара. При помощи конструкции WHILE увеличьте стоимость на 20% тех товаров, стоимость которых ниже средней.**