Учреждение образования “БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ”

Лабораторная работа №8. Основы программирования на T-SQL

Выполнил:

студент 2 курса, 7 группы

Бобрович Глеб Сергеевич

Минск 2022ПроТТеоре

|  |  |
| --- | --- |
| **Задание** |  |

|  |  |
| --- | --- |
| 1. Разработать T-SQL-скрипт, в котором:  − объявить переменные типа char, varchar, datetime, time, int, smallint, tinint, numeric(12, 5);  − первые две переменные проинициализировать в операторе объявления;  − присвоить произвольные значения следующим двум переменным с помощью оператора SET, одной из этих переменных присвоить значение, полученное в результате запроса SELECT;  − одну из переменных оставить без инициализации и не присваивать ей значения, оставшимся переменным присвоить некоторые значения с помощью оператора SELECT;  − значения одной половины переменных вывести с помощью оператора SELECT,значения другой половины переменных распечатать с помощью оператора PRINT.  Проанализировать результаты.  2. Разработать скрипт, в котором определяется общая вместимость аудиторий. Когда общая вместимость превышает 200, то вывести количество аудиторий, среднюю вместимость аудиторий, количество аудиторий, вместимость которых меньше средней, и процент таких аудиторий. Когда общая вместимость аудиторий меньше 200, то вывести сообщение о размере общей вместимости. | **Задание 1**  use ЛР8;  DECLARE @c char ='a',  @v varchar(4)='БГТУ',  @d datetime,  @t time,  @i int, --без иниц  @s smallint,  @ti tinyint,  @n numeric(12,5);  SET @d=GETDATE();  SELECT @t='12:59:34.21';  SELECT @c c, @v v, @d d, @t t;  SELECT @s=345, @ti=1, @n=1234567.12345;  print 's='+cast(@s as varchar(10));  print 'ti='+cast(@ti as varchar(10));  print 'n='+cast(@n as varchar(13));    **Задание 2**  USE ЛР8;  DECLARE @var1 int  DECLARE @var2 int  DECLARE @var3 int  DECLARE @var4 int  SELECT @var1 = SUM(AUDITORIUM\_CAPACITY) FROM AUDITORIUM  if @var1 > 200  begin  select @var2 = (select COUNT(\*) as [Количество] from AUDITORIUM),  @var3 = (select AVG(AUDITORIUM\_CAPACITY) as [Средняя] FROM AUDITORIUM)  set @var4 = (select COUNT(\*) as [Количество] from AUDITORIUM  where AUDITORIUM\_CAPACITY < @var3)  select @var2 'Кол-во ауд.', @var3 'Средняя вместимость',  @var4 'Кол-во ауд.< AVR',  100\*(cast(@var4 as float)/cast(@var2 as float)) 'Процент ауд.< AVR'  end  else select @var1; |
| 1. Разработать T-SQL-скрипт, который выводит на печать глобальные переменные:   − @@ROWCOUNT (число обработанных строк);  − @@VERSION (версия SQL Server);  − @@SPID (возвращает системный идентификатор процесса, назначенный сервером текущему подключению);  − @@ERROR (код последней ошибки);  − @@SERVERNAME (имя сервера);  − @@TRANCOUNT (возвращает уровень вложенности транзакции);  − @@FETCH\_STATUS (проверка результата считывания строк результирующего набора);  − @@NESTLEVEL (уровень вложенности текущей процедуры).  Проанализировать результат.  4. Разработать T-SQL-скрипты, выполняющие:  − вычисление значений переменной **z**    для различных значений исходных данных;  − преобразование полного ФИО студента в сокращенное (например, Макейчик Татьяна Леонидовна в Макейчик Т. Л.);  − поиск студентов, у которых день рождения в следующем месяце, и определение их возраста;  − поиск дня недели, в который студенты некоторой группы сдавали экзамен по СУБД.  5. Продемонстрировать конструкцию IF… ELSE на примере анализа данных таблиц базы данных **Х\_UNIVER**. | **Задание 3**  USE ЛР8;  select @@ROWCOUNT 'число обраб. строк',  @@VERSION 'версия SQL Server',  @@SPID 'системный иден. процесса',  @@ERROR 'код последней ошибки',  @@SERVERNAME 'имя сервера',  @@TRANCOUNT 'уровень вложенности транзакций',  @@FETCH\_STATUS 'проверка рез-та счит. строк рез.набора',  @@NESTLEVEL 'ур. влож-сти тек. процедуры';    **Задание 4**  USE ЛР8;  declare @tt int=3, @x float=4, @z float;  if (@tt>@x) set @z=power(SIN(@tt),2);  if (@tt<@x) set @z=4\*(@tt+@x);  if (@tt=@x) set @z=1-exp(@x-2);  print 'z='+cast(@z as varchar(10));  declare @ss varchar(100)=(select top 1 NAME from STUDENT)  select substring(@ss, 1, charindex(' ', @ss))  +substring(@ss, charindex(' ', @ss)+1,1)+'.'  +substring(@ss, charindex(' ', @ss, charindex(' ', @ss)+1)+1,1)+'.'  select NAME as 'Имя', 2019-YEAR(BDAY) as 'Возраст' --у кого др в след.месяце  from STUDENT  where MONTH(BDAY)=MONTH(getdate())+1; |
| 6. Разработать сценарий, в котором с помощью CASE анализируются оценки, полученные студентами некоторого факультета при сдаче экзаменов. | **Задание 6**  USE ЛР8;  select student.NAME as 'Имя', student.IDGROUP as 'Группа',  case  when progress.NOTE between 0 and 3 then 'Плохо'  when progress.NOTE between 4 and 6 then 'Хорошо'  when progress.NOTE between 7 and 8 then 'Отлично'  else 'Идеально'  end Оценка, COUNT(\*)[Количество]  from student, PROGRESS where student.IDGROUP=17  group by student.NAME, student.IDGROUP,  case  when progress.NOTE between 0 and 3 then 'Плохо'  when progress.NOTE between 4 and 6 then 'Хорошо'  when progress.NOTE between 7 and 8 then 'Отлично'  else 'Идеально'  end |
| 7. Создать временную локальную таблицу из трех столбцов и 10 строк, заполнить ее и вывести содержимое. Использовать оператор WHILE. | **Задание 7**  USE ЛР8;  CREATE table #LR7QUERY7  ( TIND int,  TFIELD varchar(50),  NUM int  );  SET nocount on; --не вывод сообщения о вводе строк  declare @ii int=0;  while @ii<10  begin  insert #LR7QUERY7(TIND, TFIELD, NUM)  values (floor(10\*RAND()), replicate('Field',2), floor(10\*RAND()));  set @ii=@ii+1;  end;  select \* from #LR7QUERY7;  drop table #LR7QUERY7; |
| 8. Разработать скрипт, демонстрирующий использование оператора RETURN. | **Задание 8**  USE ЛР8;  declare @x int=1;  print @x+1  print @x+2  RETURN  print @x+3 |
| 9. Разработать сценарий с ошибками, в котором используются для обработки ошибок блоки TRY и CATCH. Применить функции ERROR\_NUMBER (код последней ошибки), ERROR\_ES-SAGE (сообщение об ошибке), ERROR\_LINE (код последней ошибки), ERROR\_PROCEDURE (имя процедуры или NULL), ERROR\_SEVERITY (уровень серьезности ошибки), ERROR\_ STATE (метка ошибки). Проанализировать результат. | **Задание 9**  USE ЛР8;  print 'Query 9'  begin TRY  update PROGRESS set NOTE='5'  where NOTE='6'  end TRY  begin CATCH  print ERROR\_NUMBER() --код последней ошибки  print ERROR\_MESSAGE() --сообщение об ошибке  print ERROR\_LINE() --код последней ошибки  print ERROR\_PROCEDURE() --имя процедуры или NULL  print ERROR\_SEVERITY() --уровень серьезности ошибки  print ERROR\_STATE() --метка ошибки  end CATCH |