Pasecinic Nichita

FAF – 192

21.10.2020

Databases Laboratory Work Nr.4

Title: TRANSACT-SQL: PROCEDURAL INSTRUCTIONS

Prerequisites: computer, connection to the network, book: Microsoft SQL Server 2017 by Vitalie Cotelea and Marian Cotelea, Soft: Microsoft SQL Management Server.

Tasks:

- Exercises 1, 2, 3, 4 from the SQL Lab Book

Chisinau 2020

1. Completați următorul cod pentru a afișa cel mai mare număr dintre cele trei numere prezentate:

```
DECLARE @N1 INT, @N2 INT, @N3 INT;
DECLARE @MAI_MARE INT;
SET @N1 = 60 * RAND();
SET @N2 = 60 * RAND();
SET @N3 = 60 * RAND();
-- Aici ar trebui plasate IF-urile ---
--------------------------------
PRINT @N1;
PRINT @N2;
PRINT @N3;
PRINT 'Mai mare = ' + CAST(@MAI_MARE AS VARCHAR(2));
```

Exercitiul 1:

```
declare @n1 int, @n2 int, @n3 int;
declare @maiMare int;
set @n1 = 60*rand();
set @n2 = 60*rand();
set @n3 = 60*rand();

if @n1 >= @n2 and @n1 >= @n3
	set @maiMare = @n1
else if @n2 >= @n1 and @n2 >= @n3
	set @maiMare = @n2
else if @n3 >= @n1 and @n3 >= @n1
	set @maiMare = @n3

print @n1;
print @n2;
print @n3;

print 'Cel mai mare nr: ' + cast(@maiMare as varchar(2));
```

2. Afișați primele zece date (numele, prenumele studentului) în funcție de valoarea notei (cu excepția notelor 6 și 8) a studentului la primul test al disciplinei *Baze de date*, folosind structura de alternativă *IF...ELSE*. Să se folosească variabilele.

Exercitiul 2:

```
declare @counter int,@sId int, @nume varchar(20), @prenume varchar(30), @mark int
set @counter = 10
set @sId = 0

while @counter <> 0
	begin
		select @nume = s.Nume_Student, @prenume = s.Prenume_Student, @mark = r.Nota
		from discipline as d, studenti as s, studenti_reusita as r
		where
			s.Id_Student = r.Id_Student and
			d.Disciplina = 'Baze de date' and
			r.Tip_Evaluare = 'Testul 1' and
			d.Id_Disciplina = r.Id_Disciplina and
			s.Id_Student = @sId + 100
		if @mark <> 8 and @mark <> 6
			begin
				print @nume + ' ' + @prenume + ' ' + ',Nota -> ' + cast(@mark
as varchar(2))
```

```
                -- update counter
                set @counter = @counter - 1
            end
            -- update studentId
            set @sId = @sId + 1
        end
```

3. Rezolvați aceeași sarcină, 1, apelând la structura selectivă *CASE*.

Exercitiul 1 modificat:

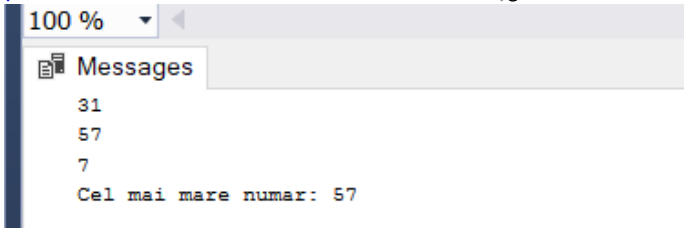
```
declare @n1 int, @n2 int, @n3 int, @maiMare int;

set @n1 = 60*rand();
set @n2 = 60*rand();
set @n3 = 60*rand();

select @maiMare =
       case
              when @n1 >= @n2 and @n1 >= @n3 then @n1
              when @n2 >= @n1 and @n2 >= @n3 then @n2
              when @n3 >= @n1 and @n3 >= @n1 then @n3
       end

print @n1;
print @n2;
print @n3;

print 'Cel mai mare numar: ' + cast(@maiMare as varchar(2))
```



4. Modificați exercițiile din sarcinile 1 și 2 pentru a include procesarea erorilor cu *TRY* și *CATCH*, și *RAISERRROR*.

Exercitiul 1 modificat:

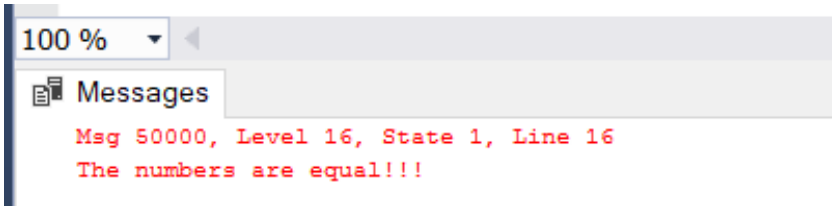
```
declare @n1 int, @n2 int, @n3 int, @maiMare int;

set @n1 = 0;
set @n2 = 0;
set @n3 = 0;

select @maiMare =
	case
		when @n1 >= @n2 and @n1 >= @n3 then @n1
		when @n2 >= @n1 and @n2 >= @n3 then @n2
		when @n3 >= @n1 and @n3 >= @n1 then @n3
	end

if @maiMare = @n1 and @maiMare = @n2 and @maiMare = @n3
	begin
		raiserror('The numbers are equal!!!', 16, 1)
	end

else
	begin
		print @n1;
		print @n2;
		print @n3;
		print 'Cel mai mare numar: ' + cast(@maiMare as varchar(2))
	endend catch
```



Ex 2 modificat:

```
declare @counter int,@sId int, @nume varchar(20), @prenume varchar(30), @mark int
set @counter = 10
set @sId = 0

while @counter <> 0
	begin
		select @nume = s.Nume_Student, @prenume = s.Prenume_Student, @mark = r.Nota
		from discipline as d, studenti as s, studenti_reusita as r
		where
			s.Id_Student = r.Id_Student and
			d.Disciplina = 'Baze de date' and
			r.Tip_Evaluare = 'Testul 1' and
			d.Id_Disciplina = r.Id_Disciplina and
			s.Id_Student = @sId + 100
		begin try
			if @mark <> 8 and @mark <> 6
				begin
```

```
                                print @nume + ' ' + @prenume + ' ' + ',Nota -> ' +
cast(@mark as varchar(2))
                                -- update counter
                                set @counter = @counter - 1
                        end
                end try
                begin catch
                        print 'A error occured!'
                end catch
                        -- update studentId
                        set @sId = @sId + 1
                end
```

Conclusion: In this work I learned and applied in practice Procedural Instructions of the SQL Transact, which have common things to other popular programming languages. Transact SQL extends the SQL standard with the implementation of programming structures.