Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра «Проектирования информационно-компьютерных систем»

**ОТЧЕТ**

по лабораторной работе №4

по дисциплине «Системы и методы управления базами данных»

На тему: «ОСНОВЫ ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ ВСТРОЕННОГО ЯЗЫКА TRANSACT-SQL В MICROSOFT SQL SERVER»

**Выполнил:** Студент гр.914302

Недбальский И.Д.

**Проверила:** Лукашевич А. Э.

Минск 2022

**Название:** ОСНОВЫ ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ ВСТРОЕННОГО ЯЗЫКА TRANSACT-SQL В MICROSOFT SQL SERVER

**Цель практической работы:** изучить используемый в реляционных СУБД встроенный язык программирования Transact-SQL для написания программ в MS SQL Server. Изучить правила построения идентификаторов, правила объявления переменных и их типов. Изучить принципы работы с циклами и ветвлениями. Изучить работу с переменными типа Table. Изучить синтаксис и семантику функций и хранимых процедур Transact– SQL: способов их идентификации, методов задания и спецификации параметров и возвращаемых значений, и вызовов функций и хранимых процедур.

Скрипт функций, согласно варианту:

use university

DECLARE @str CHAR(30)

SELECT @str = Surname FROM Students

SELECT @str

DECLARE @a INT

SET @a = (SELECT COUNT(\*) FROM Groups)

SELECT @a

DECLARE @mytable TABLE(id INT, myname CHAR (255) DEFAULT 'Ââåäèòå íàçâàíèå')

INSERT @mytable SELECT Id\_facultet, Name\_facultet FROM Facultet

SELECT \* FROM @mytable

use university

SELECT AVG(Salary) From Teachers

DECLARE @s float

SET @s = (SELECT AVG(Salary) From Teachers)

SET @s = @s \* 123.34;

Select @s;

Select Stipendia from Students

DECLARE @stp int

SET @stp = (SELECT SUM(Stipendia) From Students)

Select @stp;

select \* from groups

where Name\_group != '914301' and Name\_group != '914302'

DECLARE @k INT

SET @k = (SELECT COUNT(\*) FROM Kafedra)

SELECT @k

DECLARE @temp TABLE(id INT, date\_updated DATETIME, age BIGINT, surname CHAR (255))

INSERT @temp VALUES (1,'22/10/2001 23:00:00', 24, 'Ñòóäåíò')

INSERT @temp SELECT Id\_student, Date\_updated, Age, Surname FROM Students

SELECT \* FROM @temp

DECLARE @n INT

DECLARE @res CHAR(30)

SET @n = (SELECT COUNT(\*) FROM kafedra)

IF @n >10 BEGIN

SET @res = 'Êîëè÷åñòâî êàôåäð áîëüøå 10' SELECT @res

END ELSE BEGIN

SET @res = 'Êîëè÷åñòâî êàôåäð = ' + str(@n) SELECT @res

END

DECLARE @f INT

DECLARE @fac\_res CHAR(30)

SET @f = (SELECT COUNT(\*) FROM Facultet)

IF @f >=2 AND @f <= 4 BEGIN

SET @fac\_res = '' SELECT @fac\_res

END ELSE BEGIN

SET @fac\_res = 'Âñåãî ' + str(@f) + ' ôàêóëüòåòîâ' SELECT @fac\_res

END

SELECT Year(Date\_of\_birth) FROM Students;

SELECT AVG(Year(Date\_of\_birth)) FROM Students

DECLARE @d int

DECLARE @d\_res CHAR(30)

SET @d = (SELECT AVG(Year(Date\_of\_birth)) FROM Students)

IF @d >='1980' AND @d <= '1999' BEGIN

SET @d\_res = '' SELECT @d\_res

END ELSE BEGIN

SET @d\_res = 'Ñð. ãîä ðîæäåíèÿ = ' + str(@d) SELECT @d\_res

END

DECLARE @p INT SET @p = 1 WHILE @p <100

BEGIN

PRINT @p -- âûâîä íà ýêðàí çíà÷åíèÿ ïåðåìåííîé

IF (@p>40) AND (@p<50)

BREAK --âûõîä è âûïîëíåíèå 1-é êîìàíäû çà öèêëîì

ELSE

SET @p = @p+rand()\*10

CONTINUE

END

PRINT @p

DECLARE @i INT

DECLARE @count INT

DECLARE @kaf TABLE(id INT, kafedra\_name varchar(50))

INSERT @kaf SELECT Id\_kafedra, Name\_kafedra FROM Kafedra

SELECT @i = Id\_kafedra FROM Kafedra

SET @count = (Select Count(\*) From Kafedra)

WHILE @count <10

BEGIN

SET @i= @i +1

SET @count= @count+1

INSERT @kaf VALUES (@i, 'Èìÿ íåèçâåñòíî')

END

SELECT \* FROM @kaf

USE University;

GO

IF OBJECT\_ID (N'dbo.ISOweek', N'FN') IS NOT NULL

DROP FUNCTION dbo.ISOweek;

GO

CREATE FUNCTION dbo.ISOweek (@DATE date) RETURNS CHAR(15)

WITH EXECUTE AS CALLER AS

BEGIN

DECLARE @man int;

DECLARE @ISOweek char(15);

SET @man= MONTH(@DATE)

IF (@man=1) SET @ISOweek='ßíâàðü';

IF (@man=2) SET @ISOweek='Ôåâðàëü';

IF (@man=3) SET @ISOweek='Ìàðò';

IF (@man=4) SET @ISOweek='Àïðåëü';

IF (@man=5) SET @ISOweek='Ìàé';

IF (@man=6) SET @ISOweek='Èþíü';

IF (@man=7) SET @ISOweek='Èþëü';

IF (@man=8) SET @ISOweek='Àâãóñò';

IF (@man=9) SET @ISOweek='Ñåíòÿáðü';

IF (@man=10) SET @ISOweek='Îêòÿáðü';

IF (@man=11) SET @ISOweek='Íîÿáðü';

IF (@man=12) SET @ISOweek='Äåêàáðü';

RETURN(@ISOweek);

END;

GO

SET DATEFIRST 1;

SELECT dbo.ISOweek('12.04.2004') AS 'Ìåñÿö';

USE University;

GO

IF OBJECT\_ID (N'ufn\_SalesByStore', N'IF') IS NOT NULL

DROP FUNCTION DEKAN.ufn\_SalesByStore;

GO

CREATE FUNCTION DEKAN.ufn\_SalesByStore(@storeid int) RETURNS TABLE

AS RETURN (

SELECT d.Name\_kafedra AS "Êàôåäðà", t.Position AS "Äîëæíîñòü",

SUM(t.Salary + t.RISE) AS "Ñóììà çàðïëàòû" FROM KAFEDRA d, TEACHERS t

WHERE d.Id\_kafedra =t.Id\_kafedra and t.salary>@storeid

GROUP BY d.Name\_kafedra, t.Position);

GO

SELECT \* from dekan.ufn\_SalesByStore(99);

--Çàïðîñ8 | Ïîëüçîâàòåëüñêàÿ ôóíêöèÿ, êîòîðàÿ âîçâðàùàåò ðåçóëüòàò â âèäå òàáëèöû,

--âûâîäèò âñåõ ó÷àùèõñÿ ñòóäåíòîâ ïî êàôåäðàì ñ óêàçàíèåì êóðñà.

--Ïðè ýòîì ôóíêöèÿ èìååò îäèí ïàðàìåòð @city, ñ ïîìîùüþ êîòîðîãî åñòü îãðàíè÷åíèå íà ãîðîä ïðîæèâàíèÿ Ìèíñê.

USE University;

GO

IF OBJECT\_ID (N'ufn\_kaf\_students', N'IF') IS NOT NULL

DROP FUNCTION DEKAN.ufn\_kaf\_students;

GO

CREATE FUNCTION DEKAN.ufn\_kaf\_students(@city varchar(50)) RETURNS TABLE

AS RETURN (

SELECT d.Name\_kafedra AS "Êàôåäðà", s.Surname AS "Ôàìèëèÿ", g.Course AS "Êóðñ"

FROM Students s

INNER JOIN Groups g ON s.id\_group=g.id\_group

INNER JOIN Kafedra d ON g.id\_kafedra=d.Id\_kafedra

WHERE s.City like @city);

GO

SELECT \* from dekan.ufn\_kaf\_students('Ìèíñê');

GO

CREATE PROCEDURE Count\_Assistent

AS

Select count(position) from TEACHERS

where position='Àññèñòåíò'

EXECUTE Count\_Assistent

GO

CREATE PROCEDURE Count\_Assistent\_Salary @Sum\_salary as Int

AS

Select count(position) from TEACHERS

WHERE position='Àññèñòåíò' and SALARY>=@Sum\_salary

EXEC Count\_Assistent\_Salary 100

GO

CREATE PROCEDURE Count\_Assistent\_Salary\_Title @Sum\_salary as Int, @Title as varchar(15)

AS

Select count(\*) from TEACHERS

WHERE position like @Title and SALARY>=@Sum\_salary

EXEC Count\_Assistent\_Salary\_Title 100, '%íò%'

GO

CREATE PROCEDURE

Count\_Assistent\_Itogo @Sum\_salary Int, @Title Char(15) , @Itogo Int OUTPUT AS

Select @Itogo = count(\*) from TEACHERS

WHERE SALARY>=@Sum\_salary AND position LIKE @Title

Declare @q As int

EXEC Count\_Assistent\_Itogo 100, '%íò%', @q output select @q

GO

CREATE PROCEDURE checkname @param int AS

IF (SELECT Surname FROM STUDENTS WHERE Id\_student = @param)

RETURN 1 ELSE RETURN 2

DECLARE @return\_status int

EXECUTE @return\_status = checkname 3 SELECT 'Return Status' = @return\_status

GO

CREATE PROC update\_proc AS

UPDATE STUDENTS SET stipendia = stipendia-50

Select stipendia from students;

EXEC update\_proc

GO

CREATE PROC select\_zavkaf @fio CHAR(10) AS

SELECT \* FROM kafedra WHERE fio\_zavkaf=@fio

EXEC select\_zavkaf 'Äîà À.Â.'

--Çàïðîñ9 | Ïðîöåäóðà update\_proc\_rise ñ âõîäíûì ïàðàìåòðîì è çíà÷åíèåì ïî óìîë÷àíèþ @p real = 0.5

-- äëÿ óâåëè÷åíèÿ çíà÷åíèÿ íàäáàâêè ê çàðïëàòå â òàáëèöå TEACHER â çàäàííîå êîëè÷åñòâî ðàç:

GO

CREATE PROC update\_proc\_rise @p float = 0.5 AS

UPDATE TEACHERS SET rise = rise + rise\*@p

Select rise from teachers;

EXEC update\_proc\_rise 1.5

GO

CREATE PROC count\_teacher

@d1 DATE, @d2 DATE, @c INT OUTPUT

AS

SELECT @c=count(Id\_teacher) from teachers

WHERE Date\_hire BETWEEN @d1 AND @d2 SET @c = ISNULL(@c,0)

DECLARE @c2 INT

EXEC count\_teacher '01/01/2006', '31/12/2008', @c2 OUTPUT SELECT @c2

--Çàäàíèå1 | Ôóíêöèÿ äëÿ âûïîëíåíèÿ ÷åòûðåõ àðèôìåòè÷åñêèõ îïåðàöèé “+”, “- ”, “\*” è “/”

--íàä öåëûìè îïåðàíäàìè òèïà bigint

GO

CREATE FUNCTION Calculator (@Opd1 bigint,

@Opd2 bigint,

@Oprt char(1) = '\*') RETURNS bigint

AS BEGIN

DECLARE @Result bigint SET @Result =

CASE @Oprt

WHEN '+' THEN @Opd1 + @Opd2 WHEN '-' THEN @Opd1 - @Opd2

WHEN '\*' THEN @Opd1 \* @Opd2 WHEN '/' THEN @Opd1 / @Opd2 ELSE 0

END

Return @Result END

GO

SELECT dbo.Calculator(4,5, '+'),

dbo. Calculator(3,7, '\*')- dbo.Calculator(64,4,'/')\*2

--Çàäàíèå2 | Ôóíêöèÿ, âîçâðàùàþùàÿ òàáëèöó ñ äèíàìè÷åñêèì íàáîðîì ñòîëáöîâ

GO

CREATE FUNCTION DYNTAB (@State char(15))

RETURNS Table AS

RETURN SELECT surname, name, city FROM students WHERE city = @state

SELECT \* FROM DYNTAB ('Ìèíñê')

ORDER BY surname

--Çàäàíèå3 | Ôóíêöèÿ, ðàçáèâàþùàÿ âõîäíóþ ñòðîêó íà ïîäñòðîêè,

-- èñïîëüçóÿ â êà÷åñòâå ðàçäåëèòåëÿ ïðîáåëû

GO

CREATE FUNCTION Parse (@String nvarchar (500))

RETURNS @tabl TABLE

(Number int IDENTITY (1,1) NOT NULL,

Substr nvarchar (30)) AS

BEGIN

DECLARE @Str1 nvarchar (500), @Pos int SET @Str1 = @String

WHILE 1>0 BEGIN

SET @Pos = CHARINDEX(' ', @Str1) IF @POS>0

BEGIN

INSERT INTO @tabl

VALUES (SUBSTRING (@Str1,1,@Pos)) END

ELSE BEGIN

INSERT INTO @tabl VALUES (@Str1) BREAK

END END RETURN END

DECLARE @TestString nvarchar (500)

Set @TestString = 'SQL Server 2019'

SELECT \* FROM Parse ('SQL Server 2019')