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»

**Выполнила:** студентка гр.914302

Медведская Ю.И.

**Проверила:** Лукашевич А. Э.

Минск 2022

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

**Цель практической работы: и**зучить 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:

1. Запрос для создания временной таблицы через переменную типа **TABLE** и запрос с условной конструкцией IF

declare @temp table

(

column1 varchar(30),

column2 int,

column3 varchar(30),

column4 int

)

insert into @temp values

('first row', 11, 'second varchar', 69),

('second row', 12, 'second varchar', 420),

('third row', 13, 'yet another varchar', 666),

('fourth row', 14, 'still varchar', 2022)

declare @counter int = 0

while @counter < 5

begin

insert into @temp values

('inserted usign while', @counter, '-', 0)

set @counter = @counter + 1

end

declare @odd bit = 0

if @odd = 1

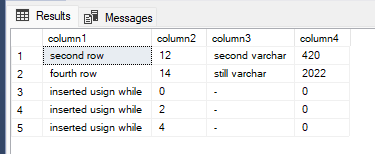
select \* from @temp

where column2 % 2 = 1

else

select \* from @temp

where column2 % 2 = 0;



1. Запрос с использованием цикла **WHILE**;

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

BEGIN

PRINT @с

IF (@с>40) AND (@с<50)

BREAK

ELSE

SET @с = @с+rand()\*10

CONTINUE

END

PRINT @с

DECLARE @d INT SET @d = 1 WHILE @d <10

BEGIN

PRINT @d

IF (@d>4) AND (@d<6)

BREAK

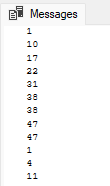
ELSE

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

CONTINUE

END

PRINT @d



1. запрос для создания скалярной функции;

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

DROP FUNCTION dbo.ISOweek;

GO

CREATE FUNCTION dbo.ISOweek (@NUMBER INT) RETURNS CHAR(150)

WITH EXECUTE AS CALLER AS

BEGIN

DECLARE @man int;

DECLARE @ISOweek char(150);

SET @man= @NUMBER

IF (@man<=10) SET @ISOweek='Число меньше или равно 10';

IF (@man>10 AND @man<30) SET @ISOweek='Число в диапозоне от 10 до 30';

IF (@man>=30) SET @ISOweek='Число больше или равно 30';

RETURN(@ISOweek);

END;

GO

SET DATEFIRST 1;

SELECT dbo.ISOweek(10) AS 'Число';



1. запрос для создания функции, которая возвращает табличное значение;

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

DROP FUNCTION db\_owner.ufn\_SalesByStore;

GO

CREATE FUNCTION db\_owner.ufn\_SalesByStore(@id int) RETURNS TABLE

AS RETURN (

SELECT t.Country AS 'Страна', t.City AS 'Город', SUM(d.NumberOfCompanys + d.NumberOfTechnique+d.NumberOfWeapons) AS 'Все'

FROM Base d, PlaceOfLocation t

WHERE d.IdOfBase =t.IdOfPlace and d.NumberOfCompanys>@id

GROUP BY t.Country,t.City

);

GO

SELECT \* from db\_owner.ufn\_SalesByStore(10);

1. запрос для создания процедуры без параметров

DROP PROCEDURE IF EXISTS Count\_Country;

go

CREATE PROCEDURE Count\_Country

AS

Select count(Country) AS 'Количество стран' from PlaceOfLocation

where Country='test'

Go

DROP PROCEDURE IF EXISTS Count\_Big\_Place;

go

CREATE PROCEDURE Count\_Big\_Place

AS

Select count(Country) AS 'Количество стран с большей площадью' from PlaceOfLocation

where Area>100

Go

EXECUTE Count\_Country

EXECUTE Count\_Big\_Place

1. запрос для создания процедуры c входным параметром

DROP PROCEDURE IF EXISTS Count\_Country\_Area;

go

CREATE PROCEDURE Count\_Country\_Area @area int

AS

Select count(Country) AS 'Количество стран' from PlaceOfLocation

where Country='test' and Area>=@area

Go

DROP PROCEDURE IF EXISTS Count\_Base;

go

CREATE PROCEDURE Count\_Base @number as Int

AS

Select count(IdOfBase) from Base

WHERE IdTypeOfTroop ='1' and NumberOfCompanys>=@number

Go

EXEC Count\_Country\_Area 6

EXEC Count\_Base 10

1. запрос для создания процедуры c входными параметрами и **RETURN**

DROP PROCEDURE IF EXISTS Check\_Country;

GO

CREATE PROCEDURE Check\_Country @param int AS

IF (SELECT Country FROM PlaceOfLocation WHERE IdOfPlace = @param)

= 'USA'

RETURN 1 ELSE RETURN 2

DECLARE @return\_status int

EXECUTE @return\_status = Check\_Country 3 SELECT 'Return Status' = @return\_status

DROP PROCEDURE IF EXISTS Check\_Number;

GO

CREATE PROCEDURE Check\_Number @param int AS

IF (SELECT NumberOfCompanys FROM Base WHERE IdOfBase = @param)

= '11'

RETURN 1 ELSE RETURN 2

DECLARE @return\_status int

EXECUTE @return\_status = Check\_Number 4 SELECT 'Return Status' = @return\_status

1. запрос для создания процедуры обновления данных в таблице базы данных **UPDATE**

DROP PROC IF EXISTS Update\_Base

GO

CREATE PROC Update\_Base AS

UPDATE Base SET NumberOfTechnique = NumberOfCompanys+10

DROP PROC IF EXISTS uUpdate\_Base2

GO

CREATE PROC Update\_Base2 AS

UPDATE Base SET NumberOfTechnique = NumberOfTechnique-10

EXEC Update\_Base

EXEC Update\_Base2

1. запроса для создания процедуры извлечения данных из таблиц базы данных **SELECT**

DROP PROC IF EXISTS Select\_Country

go

CREATE PROC Select\_Country @e CHAR(250) AS

SELECT \* FROM PlaceOfLocation WHERE Country= @e

DROP PROC IF EXISTS Select\_City

go

CREATE PROC Select\_City @e char (250) AS

SELECT \* FROM PlaceOfLocation WHERE City= @e

EXEC Select\_Country 'Belarus'

EXEC Select\_City 'Minsk';

1. Создать функцию для выполнения четырех арифметических операций “+”, “- ”, “\*” и “/” над целыми операндами типа bigint, выполнив кодирование и проверку

drop function if exists Calculator

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

use [MilitaryReg(lr3)];

SELECT dbo.Calculator(4,5, '+') AS '+', dbo.Calculator(3,7, '\*') - dbo.Calculator(64,4,'/')\*2 AS 'Итог'

1. Создать функцию, возвращающую таблицу с динамическим набором столбцов, выполнив кодирование и тестирование

DROP FUNCTION IF EXISTS DYNTAB

go

CREATE FUNCTION DYNTAB (@State char(10))

RETURNS Table AS

RETURN SELECT City, Country, Adres FROM PlaceOfLocation WHERE Country = @state

SELECT \* FROM DYNTAB ('Belarus')

ORDER BY City, Country, Adres

1. Создать функцию, разбивающую входную строку на подстроки, используя в качестве разделителя пробелы, выполнив кодирование и тестирование

DROP FUNCTION IF EXISTS ParseStr

go

CREATE FUNCTION ParseStr (@String varchar (500))

RETURNS @tab1 TABLE

(

Number int IDENTITY (1,1) NOT NULL,

Substr nvarchar (30)

) AS

BEGIN

DECLARE @Str1 varchar (500), @Pos int

SET @Str1 = @String

WHILE 1 > 0

BEGIN

SET @Pos = CHARINDEX(' ', @Str1)

IF @POS > 0

BEGIN

INSERT INTO @tab1

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

set @Str1 = (SUBSTRING (@Str1, @Pos + 1, 500))

END

ELSE

BEGIN

INSERT INTO @tab1 VALUES (@Str1)

BREAK

END

END

RETURN

END

DECLARE @TestString varchar(500)

Set @TestString = 'SQL Server 2019'

SELECT \* FROM ParseStr (@TestString)

go