

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра экономической информатики

ОТЧЕТ

Лабораторная работа №4
«Проектирование базы данных»
Вариант №1

Выполнил: студент гр.

914302 Борисенко А.И.

Проверил: Лукашевич А.Э.

Минск 2022

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)

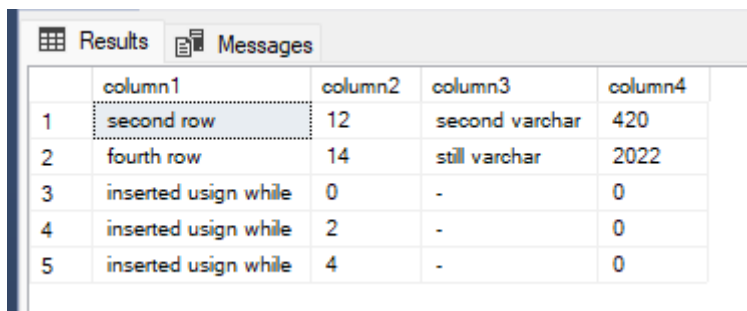
DECLARE @counter INT = 0

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;
```



	column1	column2	column3	column4
1	second row	12	second varchar	420
2	fourth row	14	still varchar	2022
3	inserted usign while	0	-	0
4	inserted usign while	2	-	0
5	inserted usign while	4	-	0

2. Запрос с использованием цикла **WHILE**

```
DECLARE @c INT SET @c = 1 WHILE @c < 100
BEGIN
    PRINT @c
    IF (@c > 40) AND (@c < 50)
        BREAK
    ELSE
        SET @c = @c + rand() * 10
    CONTINUE
END
PRINT @c
```

```

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
```

Messages
1
10
17
22
31
38
38
47
47
1
4
11

3. 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 диапазоне 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';

```

	Число
1	Число меньше или равно 10

4. 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.NumberOfCompanyys +
d.NumberOfTechnique+d.NumberOfWeapons) AS 'Все'
FROM Base d, PlaceOfLocation t
WHERE d.IdOfBase =t.IdOfPlace and d.NumberOfCompanyys>@id
GROUP BY t.Country,t.City
)

```

```
);
```

```
GO
```

```
SELECT * from db_owner.ufn_SalesByStore(10);
```

5. 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
```

6. Запрос для создания процедуры с 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
```

7. Запрос для создания процедуры с 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

```

8. 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

```

9. 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';

```

10. 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 'Итого'

```

11. 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

```

12. 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
```