

4 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»

Выполнил: студент гр.914302

Шпаковский М.Г.

Проверила: Лукашевич А. Э.

Минск 2022

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

Цель:

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.

Задание:

Для созданной базы данных, согласно номеру варианта, самостоятельно создать на языке P-SQL 10 запросов:

- 1 запрос для создания временной таблицы через переменную типа TABLE;
- 2 запроса с использованием условной конструкции IF;
- 2 запроса с использованием цикла WHILE;
- 1 запрос для создания скалярной функции;
- 1 запрос для создания функции, которая возвращает табличное значение;
- 2 запроса для создания процедуры без параметров;
- 2 запроса для создания процедуры с входным параметром;
- 2 запроса для создания процедуры с входными параметрами и RETURN;
- 2 запроса для создания процедуры обновления данных в таблице базы данных UPDATE;
- 2 запроса для создания процедуры извлечения данных из таблиц базы данных SELECT;

Выполнение:

1. запрос для создания временной таблицы через переменную типа TABLE;

```

use Car_dealership3;

DECLARE @mytable TABLE (
    [buyer_code] [int] IDENTITY(1,1) PRIMARY KEY,
    [surname] [nvarchar](50) NOT NULL,
    [name] [nvarchar](50) NOT NULL,
    [patronymic] [nvarchar](25) NULL,
    [passport_data] [nvarchar](25) NOT NULL,
    [address] [nvarchar](50) NULL)
INSERT INTO @mytable(surname, name, patronymic, passport_data, address)
VALUES ('Киров', 'Андрей', 'Васильевич', 'MP5830422', 'Киреевко 5, 77')
      ('Вароная', 'Алла', 'Михайловна', 'MP4234242', 'Белоусова 36'),
      ('Плющ', 'Олег', 'Васильевич', 'MP4738299', 'Е.Полоцкой 3'),
      ('Грушневая', 'Александра', 'Олеговна', 'MP5839994', 'Алибегова 23'),
      ('Никифорова', 'Надежда', 'Ивановна', 'MP5748499', 'Киреевко 5, 77')
SELECT * FROM @mytable

```

buyer_code	surname	name	patronymic	passport_data	address
1	Киров	Андрей	Васильевич	MP5830422	Киреевко 5, 77
2	Вароная	Алла	Михайловна	MP4234242	Белоусова 36
3	Плющ	Олег	Васильевич	MP4738299	Е.Полоцкой 3
4	Грушневая	Александра	Олеговна	MP5839994	Алибегова 23
5	Никифорова	Надежда	Ивановна	MP5748499	Киреевко 5, 77

2. запрос с использованием условной конструкции IF;

```

use [Car_dealership3];
DECLARE @a INT
DECLARE @str CHAR(50)
SET @a = (SELECT COUNT(*) FROM Buyers)
IF @a > 5 BEGIN
    SET @str = 'Количество покупателей больше 5' SELECT @str
END ELSE BEGIN
    SET @str = 'Покупателей'+str(@a) SELECT @str
END

```

Результаты	Сообщения
(Отсутствует имя столбца)	
1	Количество покупателей больше 5

```

DECLARE @Number INT;
SET @Number = 50;
IF @Number > 100
    PRINT 'The number is large.';
ELSE
    BEGIN
        IF @Number < 10
            PRINT 'The number is small.';
        ELSE
            PRINT 'The number is medium.';
    END ;
GO

```

20 %

Сообщения

The number is medium.

Время выполнения: 2022-11-28T14:31:42.6564671+03:00

3. запроса с использованием цикла WHILE;

```

WHILE ( SELECT AVG(salary) FROM dbo.Workers) < 600
BEGIN
    UPDATE dbo.Workers
    SET salary = Salary * 1.5;
    SELECT MAX (salary) FROM dbo.Workers
    IF ( SELECT MAX (salary) FROM dbo.Workers) > $500
        BREAK;
END

```

Результаты	Сообщения
(Отсутствует имя столбца)	
825	

4. запрос для создания скалярной функции;

```

GO
CREATE FUNCTION dbo.later (@DATE date) RETURNS CHAR(1)
WITH EXECUTE AS CALLER AS
BEGIN
    DECLARE @man int;
    DECLARE @res char(15);
    IF (@DATE>GETDATE()) SET @res='Позже';
    ELSE SET @res='раньше';
    RETURN(@res);
END;
GO
SELECT dbo.later('2022-11-02') AS 'время';

```

Результаты	Сообщения
время	
раньше	

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

```

GO
CREATE FUNCTION [dbo].[exp_workers]
(
    @minimal INT
)
RETURNS @tbl TABLE (name NVARCHAR(50), surname NVARCHAR(50), experience INT)
AS
BEGIN
    INSERT @tbl
    SELECT name, surname, experience
    FROM dbo.Workers
    WHERE experience > @minimal
    --возвращаем результат
    RETURN
END;
GO
SELECT * FROM dbo.exp_workers(4)

```

%

Результаты

Сообщения

name	surname	experience
Ольга	Гришина	5
Виктор	Блюрин	5

6. запрос для создания процедуры без параметров;

```
CREATE PROCEDURE Count_Colors
AS
Select count(color) from Cars
Go
```

```
CREATE PROCEDURE Count_Olegs
AS
Select count(*) from Workers
WHERE name='Олег'
Go
```

```
EXECUTE Count_Colors
EXECUTE Count_Olegs
```

Результаты	Сообщения
(Отсутствует имя столбца)	
5	

Результаты	Сообщения
(Отсутствует имя столбца)	
1	

7. запрос для создания процедуры с входным параметром;

```
ALTER PROCEDURE Count_Worker_Salary @Sum_salary as Int
AS
Select count(experience) from Workers
WHERE experience=3 and SALARY>=@Sum_salary
Go
```

```
EXEC Count_Worker_Salary 600
```

Результаты	Сообщения
(Отсутствует имя столбца)	
3	

8. запрос для создания процедуры с входными параметрами и RETURN;

```
CREATE PROCEDURE is_oleg @param int AS
IF (SELECT name FROM Workers WHERE worker_code = @param)
= 'Олег'
RETURN 1 ELSE RETURN 0
DECLARE @return_status int
EXECUTE @return_status = is_oleg 3 SELECT 'Это олег' = @return_status
```

Результаты	Сообщения
Это олег	
0	

```

Create PROCEDURE is_color_any @param NVARCHAR AS
IF ( SELECT count(color) FROM dbo.Cars) > 0
RETURN 1 ELSE RETURN 0

DECLARE @return_color int
EXECUTE @return_color = is_color_any 'Red' SELECT 'is color' = @return_color

```

%

Результаты Сообщения

is color
1

9. запрос для создания процедуры обновления данных в таблице базы данных UPDATE;

```

CREATE PROC update_salary AS
UPDATE Workers SET salary = salary-10

EXEC update_salary

```

109 %

Сообщения

(затронута строк: 7)

Время выполнения: 2022-11-28T15:56:32.4996480+03:00

10. запроса для создания процедуры извлечения данных из таблиц базы данных SELECT;

```

ALTER PROC select_name @k CHAR(10) AS
SELECT * FROM Workers WHERE name=@k

select_name 'Анна'

```

109 %

Результаты Сообщения

	worker_code	surname	name	patronymic	experience	salary
1	7	Панасенко	Анна	Кирилловна	3	612.5

```

CREATE PROC select_color @e CHAR(10) AS
SELECT * FROM Cars WHERE color=@e

select_color 'Red'

```

109 %

Результаты Сообщения

	car_code	car_name	brand_code	production_year	color
1	1	ferrari laferrari	5	2013	Red
2	5	ford mustang	3	2017	Red

СОЗДАНИЕ, ИЗМЕНЕНИЕ, ПРИМЕНЕНИЕ И УДАЛЕНИЕ ФУНКЦИЙ И ХРАНИМЫХ ПРОЦЕДУР

Цель:

Изучение синтаксиса и семантики функций и хранимых процедур Transact– SQL: способов их идентификации, методов задания и спецификации параметров и возвращаемых значений, кодирования тела и вызовов функций и хранимых процедур, применение команд для создания, изменения и удаления системных и пользовательских как скалярных, так и табличных (с одной Inline или несколькими Multi – statement командами в теле) функций, системных, пользовательских, временных (локальных или глобальных) и расширенных хранимых процедур, а также приобретение навыков программирования, отладки, тестирования и включения в группу или подключения библиотеки функций и хранимых процедур.

Выполнение:

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

```
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 [Car_dealership3];
SELECT dbo.Calculator(4,5, '+') AS '+', dbo.Calculator(3,7, '*') - dbo.Calculator(64,4, '/') * 2 AS 'Итог'
```

109 %

Результаты Сообщения

	+	Итог
1	9	-11

Задание 2. Создать функцию, возвращающую таблицу с динамическим набором столбцов, выполнив кодирование и тестирование.

```
CREATE FUNCTION DYNTAB (@State char(10))
RETURNS Table AS
RETURN SELECT Name, Surname, Patronymic FROM Workers WHERE Name = @state

SELECT * FROM DYNTAB ('Олег')
ORDER BY Name, Surname, Patronymic
```

109 %

Результаты Сообщения

	Name	Surname	Patronymic
1	Олег	Панасенко	Михайлович

Задание 3. Создать функцию, разбивающую входную строку на подстроки, используя в качестве разделителя пробелы, выполнив кодирование и тестирование:

```
CREATE FUNCTION ParseFunction (@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 ParseFunction (@TestString)
```

82 %

Результаты Сообщения

	Number	Substr
1	1	SQL
2	2	Server
3	3	2019

Вывод:

В ходе данной лабораторной работы были изучены основы программирования с помощью языка transact-sql в Microsoft server, а также создание, изменение, применение и удаление функций и хранимых процедур. По результатам лабораторной работы был составлен отчет.