

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 Мирошник А.А.
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 supermarket;
```

--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;

```
DECLARE @mytable TABLE (id INT, myname CHAR(20) DEFAULT 'Иванов Иван')
```

```
INSERT INTO @mytable(id) VALUES (1)
```

```
INSERT INTO @mytable(id, myname) VALUES (2,'Игорь Троцкий')
```

```
SELECT * FROM @mytable;
```

-- 2 запроса с использованием условной конструкции IF;

```
DECLARE @a INT
```

```
DECLARE @str CHAR(30)
```

```
SET @a = (SELECT COUNT(summ) FROM product_sale)
```

```
IF @a > 4 BEGIN
```

```
SET @str = 'Сумма больше 4' SELECT @str
```

```
END ELSE BEGIN
```

```
SET @str = 'Сумма = ' + str(@a) SELECT @str
```

```
END;
```

```
DECLARE @b INT
```

```

DECLARE @str1 CHAR(90)
SET @b = (SELECT SUM(summ) FROM product_sale)
IF @b > 60 BEGIN
SET @str1 = 'Суммарная стоимость больше 60' SELECT @str1
END ELSE BEGIN
SET @str1 = 'Суммарная стоимость машин = ' + str(@b) SELECT @str1
END;

```

-- 2 запроса 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 @k INT SET @k = 1
DECLARE @d INT SET @d = (SELECT amount FROM product_sale WHERE product_id=@k) WHILE @d > 6000
BEGIN
PRINT @d
IF (@d > 6000) AND (@d < 6500)
BREAK
ELSE
SET @k = @k + 1
SET @d = (SELECT amount FROM product_sale WHERE product_id=@k)
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 (@PRICE int) RETURNS CHAR(15)
```

```
WITH EXECUTE AS CALLER AS
```

```
BEGIN
```

```
DECLARE @man int;
```

```
DECLARE @ISOweek char(15);
```

```
SET @man= @PRICE
```

```
IF (@man<=10000) SET @ISOweek='Дешево';
```

```
IF (@man>10000 AND @man<30000) SET @ISOweek='Нормально';
```

```
IF (@man>=30000) SET @ISOweek='Дорого';
```

```
RETURN(@ISOweek);
```

```
END;
```

```
GO
```

```
SET DATEFIRST 1;
```

```
SELECT dbo.ISOweek(300000) AS 'Месяц';
```

```
--      2 запроса для создания процедуры без параметров;
```

```
DROP PROCEDURE IF EXISTS Count_Halnazar;
```

```
go
```

```
CREATE PROCEDURE Count_Halnazar
```

```
AS
```

```
Select count(name_client) from client
```

```
where name_client='Halnazar'
```

```
Go
```

```
DROP PROCEDURE IF EXISTS Count_Adress;
```

```
go
```

```
CREATE PROCEDURE Count_Adress
```

```
AS
```

```
Select count(address) from client
```

```
where adress = 'Barcelona'
```

```
Go
```

```
EXECUTE Count_Halnazar
```

```
EXECUTE Count_Adress
```

```
--      2 запроса для создания процедуры с входными параметрами и RETURN;
```

```
DROP PROCEDURE IF EXISTS checkname;
```

```
GO
```

```
CREATE PROCEDURE checkname @param int AS
```

```
IF (SELECT amount FROM product_sale WHERE ID = @param)
```

```
= 1
```

```
RETURN 1 ELSE RETURN 2
```

```
DROP PROCEDURE IF EXISTS checkage;
```

GO

CREATE PROCEDURE checkage @param int AS

IF (SELECT prime FROM product_sale WHERE ID = @param)

= 2

RETURN 1 ELSE RETURN 2

DECLARE @return_status int

EXECUTE @return_status = checkage 1 SELECT 'Return Status' = @return_status

-- 2 запроса для создания процедуры обновления данных в таблице базы данных UPDATE;

DROP PROC IF EXISTS update_proc11

GO

CREATE PROC update_proc11 AS

UPDATE product_sale SET amount = amount+30

DROP PROC IF EXISTS update_proc22

GO

CREATE PROC update_proc22 AS

UPDATE product_sale SET amount = amount-300

EXEC update_proc11

EXEC update_proc22