

Карпухина Анна

Вариант 6

Задание 1

```
2. Объявить переменную Regem1 типа строка длиной 100, а переменную Regem2 типа длинное целое.
3. Объявить переменную Regem1 типа динамическая строка с максимальной длиной 1000, а переменную Regem2 типа целое число.
4. Объявить переменную Regem1 типа строка длиной 30, а переменную Regem2 типа число с целой частью равной 10 и дробной частью равной 5.
5. Объявить переменную Regem1 типа дата/ время, а переменную Regem2 типа число в диапазоне от 0 до 255.
*/

DECLARE @a MONEY,
        @b NUMERIC(10,2),
        @text CHAR(100),
        @c BIGINT,
        @DINAMtext VARCHAR(1000),
        @d INT,
        @text1 CHAR(30),
        @E NUMERIC(10,5),
        @DT DATETIME,
        @num TINYINT
```

Задание 2

```
/*Работа с датой и временем задание 2|

1. Определить переменную Date1 типа дата/время. Присвоить ей значение даты 31.12.2009 в формате dd.mm.yyyy.
2. Определить переменную Date1 типа дата/время. Присвоить ей значение даты 31.12.2009 в формате mm.dd.yyyy.
3. Определить переменную Date1 типа дата/время. Присвоить ей значение даты 31.12.2009 в формате yyyy.mm.dd.

*/
SET DATEFORMAT dmy
DECLARE @Date1 DATETIME
SET @Date1 = '31.12.2009'

SET DATEFORMAT mdy
DECLARE @Date2 DATETIME
SET @Date2 = '12.31.2009'

SET DATEFORMAT ymd
DECLARE @Date3 DATETIME
SET @Date3 = '2009.12.31'
```

Задание 3

```
/*Преобразование типов переменных задание 3

1. Объявить переменные типа FLOAT, CHAR, TINYINT, BIT.
Присвоить значения, соответствующие типам.
Выполнить преобразование переменных типа FLOAT, CHAR, TINYINT, BIT в INT, DATETIME, BIT, FLOAT соответственно и вывести результат на экран.

2. Объявить переменные типа INT, DATETIME, NUMERIC, VARCHAR.
Присвоить значения, соответствующие типам.
Выполнить преобразование переменных типа INT, DATETIME, NUMERIC, VARCHAR в FLOAT, CHAR, FLOAT, INT соответственно и вывести результат на экран.

*/

DECLARE @p1 FLOAT, @p2 CHAR, @p3 TINYINT, @p4 BIT
SET @p1 = 1.57
SET @p2 = 'Анна'
SET @p3 = 2
SET @p4 = 'True'
SELECT CAST(@p1 AS INT) AS int1
SELECT CAST(@p2 AS DATETIME) AS datetime1
SELECT CAST(@p3 AS BIT) AS bit1
SELECT CAST(@p4 AS FLOAT) AS float1

DECLARE @q1 INT, @q2 DATETIME, @q3 NUMERIC, @q4 VARCHAR
SET @q1 = 5
SET @q2 = '1990-01-01 00:00:00'
SET @q3 = 584.72
SET @q4 = 'CTPOKA'
SELECT CAST(@q1 AS FLOAT) AS float1
SELECT CAST(@q2 AS CHAR) AS char1
SELECT CAST(@q3 AS FLOAT) AS float1
SELECT CAST(@q4 AS INT) AS int1
```

Задание 4

```
/*
Использование функций для работы с типом дата/время задание 4

1. Вывести на экран название текущего месяца и текущее время.

2. Разобрать на отдельные составляющие текущую дату и время и вывести значения на экран
в следующем порядке (вместо многоточий): "Сегодня: День = ..., Месяц = ..., Год = ..., Часов = ..., Минут = ..., Секунд= ..."

3. В исходный текст, сохраненный в переменной Perem,
после слова " время " вставить текущее время. Результат сохранить в той же переменной Perem и вывести на экран.
*/

SELECT DATENAME(MONTH,GETDATE())+ ', ' + CAST(DATEPART(HOUR,GETDATE())AS CHAR(4))+
': ' + CAST (DATEPART(MINUTE,GETDATE())AS CHAR(4))+': ' + CAST (DATEPART(SECOND,GETDATE())AS CHAR(4))

SELECT 'Сегодня: День = ' + CAST(DATENAME(DAY, GETDATE()) AS CHAR(2))+', ' +
'Месяц = '+DATENAME(MONTH, GETDATE())+', ', 'Год =' +CAST(DATENAME(YEAR, GETDATE()) AS CHAR(4))+', ' +
'Часов =' +CAST(DATENAME(HOUR, GETDATE())AS CHAR(2))+', ' + 'Минут =' +CAST(DATENAME(MINUTE, GETDATE())AS CHAR(2))+', ' +
'Секунд=' + CAST(DATENAME(SECOND, GETDATE())AS CHAR(2))

DECLARE @Perem CHAR(10)
SET @Perem = 'ВРЕМЯ : '
SELECT @Perem + CAST(DATENAME(HOUR, GETDATE())AS CHAR(2))+': ' +CAST(DATENAME(MINUTE, GETDATE())AS CHAR(2))+': ' +
CAST(DATENAME(SECOND, GETDATE())AS CHAR(2))
/*
```

Задание 5

```
/*
Простое выражение Case: задание 5

Задание: Используя простое выражение Case, произвести вывод измененных значений одного из полей таблицы
базы данных своего варианта (использовать не менее 3 альтернатив изменения и предусмотреть использование ELSE).
При необходимости изменить значения полей в используемой таблице.
*/

USE Радиостанции
GO

SELECT CASE Код_должности WHEN 1 THEN 'Один'
                           WHEN 2 THEN 'Два'
                           WHEN 3 THEN 'Три'
                           WHEN 4 THEN 'Четыре'
                           WHEN 5 THEN 'Пять'
                           ELSE ''
END AS Код_должности_text

FROM Сотрудники1
```

Задание 6

```
/*
```

Поисковое выражение CASE: задание 6

Задание: Использую поисковое выражение Case, произвести замену значений одного из полей таблицы базы данных своего варианта (использовать не менее 3 альтернатив изменения и предусмотреть использование ELSE). При необходимости изменить значения полей в используемой таблице.

```
*/
```

```
USE Радиостанции
```

```
GO
```

```
SELECT CASE WHEN Код_должности = 1 THEN 'Один'
           WHEN Код_должности = 2 THEN 'Два'
           WHEN Код_должности = 3 THEN 'Три'
           WHEN Код_должности = 4 THEN 'Четыре'
           WHEN Код_должности = 5 THEN 'Пять'
           ELSE ''
END AS Код_должности_text1

FROM Сотрудники1
```

Задание 7

```
/*
```

Условная конструкция IF задание 7

1. Подсчитать количество записей в одной из таблиц базы данных своего варианта.
Если их в таблице от ?? до ?? (задать значения), то ничего не сообщать,
в противном случае вывести сообщение вида "В таблице ... записей" (вместо многоточия поставить точное количество записей).

2. Подсчитать сумму значений одного из полей таблицы базы данных своего варианта.
Если полученная сумма в диапазоне от ?? до ?? (задать значения), то ничего не сообщать,
в противном случае вывести сообщение вида "Сумма по полю ????? = ..." (вместо многоточия поставить точную сумму)

```
*/
```

```
USE Радиостанции
```

```
GO
```

```
DECLARE @QUANTITY INT, @TEXT CHAR(100)
SET @QUANTITY = (SELECT COUNT(*) FROM Сотрудники1)
SET @TEXT = 'В ТАБЛИЦЕ БОЛЬШЕ 4 ЗАПИСЕЙ'
```

```
IF @QUANTITY >= 4
```

```
BEGIN
```

```
    SELECT @TEXT
```

```
END
```

```
USE Радиостанции
```

```
GO
```

```
DECLARE @QUANTITY INT, @TEXT CHAR(100)
SET @QUANTITY = (SELECT SUM(Возраст) FROM Сотрудники1)
SET @TEXT = 'Сумма по полю'
```

```
IF @QUANTITY >= 150
```

```
BEGIN
```

```
    SELECT @TEXT + CAST (@QUANTITY AS CHAR(3))
```

```
END
```

Задание 8

Цикл WHILE задание 8

Вычислить сумму членов последовательности.
Формула общего члена последовательности определяется номером варианта.
Вычисление вести до заданной точности вычисления (задается в программе).
Точность вычисления определяется как разность двух членов последовательности.

6. $(-1)^{(N+1)}/(2*N)!$

*/

USE Радиостанции

GO

ALTER FUNCTION GetSeqU

(@N int)

RETURNS float

AS

BEGIN

DECLARE @D FLOAT

--6. $(-1)^{(N+1)}/(2*N)$

SET @D=POWER((-1),(@N+1)/(2*@N));

RETURN @D

END

DECLARE @SUM FLOAT

DECLARE @N INT

DECLARE @VAL FLOAT

SET @N=1

SET @VAL=1

SET @SUM=0

WHILE @VAL > 0.01

BEGIN

SET @VAL = (SELECT dbo.GetSeqU(@N+1))-(SELECT dbo.GetSeqU(@N))

SET @SUM=@SUM+@VAL

SET @N=@N+1;

END

SELECT @SUM

Задание 9

/*

Задание: Для одной из таблиц базы данных своего варианта, задание 9|
имеющей ключевое поле с автоматическим наращиванием значения,
определить количество записей в таблице. Пока записей меньше ?? (задать значение),
делать в цикле добавление записи в таблицу с автоматическим наращиванием значения ключевого поля,
а вместо значения одного из полей вставлять значение 'Не задано'.

*/

USE Радиостанции

GO

DECLARE @VAL INT

SET @VAL=(SELECT COUNT(*) FROM Сотрудники)

WHILE @VAL<5

BEGIN

INSERT INTO Сотрудники

(ФИО, Возраст)

VALUES ('Не задано',0)

END

Задание 10

```

/*
Задание: Создать таблицу и заполнить одно из ее полей датами Задание 10
от даты вашего рождения до даты, отличающейся на количество дней, равных дню Вашего рождения
(дата рождения 15 марта 1990, от этой даты заполнить 15 дней).
Количество дней получить из заданной даты рождения.
*/
USE Радиостанции;
GO

IF OBJECT_ID (N'Date_of_Birth', N'U') IS NOT NULL
    DROP TABLE Date_of_Birth;
GO
CREATE TABLE Date_of_Birth (
    ID_Date INT IDENTITY(1,1),
    Date_Birth DATETIME NOT NULL
)
DECLARE @DATE DATETIME
DECLARE @N INT
SET @N=1
SET @DATE='28.01.2001'

WHILE @N<28
BEGIN
    INSERT INTO Date_of_Birth
    VALUES (@DATE+@N)
    SET @N=@N+1
END

SELECT * FROM Date_of_Birth

```

Задание 11

/*
Создание временной таблицы через переменную типа TABLE Задание 11

1. Создать локальную временную таблицу с названием TEMP1. Заполнить таблицу TEMP2 данными из имеющейся таблицы и вывести результат на экран.

2. Создать табличную переменную с названием TEMP2 и полями типа длинное целое со счетчиком, дата/время, целое, строка. Добавить в нее две записи с данными и вывести результат на экран.

```

/*
USE Радиостанции;
GO

IF OBJECT_ID (N'TEMP1', N'U') IS NOT NULL

DROP TABLE #TEMP1;

GO
IF OBJECT_ID (N'TEMP2', N'U') IS NOT NULL

DROP TABLE #TEMP2;

GO

CREATE TABLE #TEMP1
(Код_сотрудника INT IDENTITY(1,1),
Должность VARCHAR(40),
Зарботок INT NOT NULL
)

INSERT INTO #TEMP1 (Должность, Зарботок) VALUES ('Звукорежиссер', 30000)

```

```

INSERT INTO #TEMP1 (Должность, Зарплата) VALUES ('Уборщик', 10000)
INSERT INTO #TEMP1 (Должность, Зарплата) VALUES ('Флорист', 20000)

CREATE TABLE #TEMP2
(Код_сотрудника INT IDENTITY(1,1),
Должность VARCHAR(40),
Зарплата INT NOT NULL
)

INSERT INTO #TEMP2 SELECT Должность, Зарплата FROM #TEMP1
SELECT *FROM #TEMP2

```

Задание 12

Задания 12:

1. На базе имеющихся запросов (код SQL запросов нужно изменить таким образом, чтобы в них можно было передавать значения полей, по которым осуществляется поиск) создать:

1. Процедуру без параметров.
2. Процедуру с входными параметрами
3. Процедуру с входными параметрами и выходным параметром
*/

```

CREATE PROCEDURE STAFF_AS
SELECT COUNT(ФИО) FROM Сотрудники1
GO
EXEC STAFF

CREATE PROCEDURE STAFF_W @МАКС_ВОЗРАСТ INT
AS SELECT MAX(Возраст)
FROM Сотрудники1
WHERE Возраст<=@МАКС_ВОЗРАСТ
GO
EXEC STAFF_W 50

ALTER PROCEDURE STAFF_D @МИН_ВОЗРАСТ INT, @МАКС_ВОЗРАСТ INT, @СРЕДНИЙ_ВОЗРАСТ INT OUTPUT
AS SELECT @СРЕДНИЙ_ВОЗРАСТ=AVG(Возраст)
FROM Сотрудники1
WHERE Возраст BETWEEN @МИН_ВОЗРАСТ AND @МАКС_ВОЗРАСТ
GO

DECLARE @СРЕДНИЙ_ВОЗРАСТ INT
EXEC STAFF_D 18,60, @СРЕДНИЙ_ВОЗРАСТ OUTPUT
PRINT 'СРЕДНИЙ ВОЗРАСТ ОТ 18 ДО 60 РАВЕН'+CAST (@СРЕДНИЙ_ВОЗРАСТ AS CHAR(5))

```

Задание 13

/*

Создание и использование функций. задание 13

1. Создать скалярную функцию и использовать обращение к ней в запросе.
2. Создать линейную табличную функцию и обращение к ней.

*/

```

USE Радиостанции
GO

```

```

ALTER FUNCTION GetSeqU
(@N int)
RETURNS float
AS
BEGIN
DECLARE @D FLOAT
--6. (-1)^(N+1)/(2*N)
SET @D=POWER((-1), (@N+1)/(2*@N));
RETURN @D
END

```

```

DECLARE @SUM FLOAT
DECLARE @N INT
DECLARE @VAL FLOAT
SET @N=1
SET @VAL=1
SET @SUM=0
WHILE @VAL > 0.01
BEGIN
SET @VAL = (SELECT dbo.GetSeqU(@N+1))-(SELECT dbo.GetSeqU(@N))
SET @SUM=@SUM+@VAL
SET @N=@N+1;
END
SELECT @SUM

```

```

USE Радиостанции
GO

```

```

IF OBJECT_ID (N'FUNC1', N'U') IS NOT NULL

```

```

DROP FUNCTION FUNC1;

```

```

GO

```

```

CREATE FUNCTION FUNC1(@FIO CHAR(10))
RETURNS TABLE
AS
RETURN (SELECT *
FROM Сотрудники1
WHERE ФИО=@FIO AND Возраст=54)
GO

```

```

SELECT * FROM FUNC1('Семгов')

```

Задание 14

```

/*

```

Создание и использование триггеров. Задание 14

1. Создать DDL - триггер в базе данных варианта задания для события изменения таблицы. Продемонстрировать работу триггера на примере изменения таблицы базы данных.
 2. Создать DML – триггер для проверки бизнес-правила в базе данных варианта задания. Триггер должен проверять возможность изменения записей одной таблицы на основе проверки записей другой таблицы
- ```

*/

```

```

USE Радиостанции
GO
--СОЗДАНИЕ DDL ТРИГГЕРА
CREATE TRIGGER WORK--создание триггера
ON DATABASE
FOR DROP_TABLE
AS
PRINT 'НЕВОЗМОЖНО УДАЛИТЬ ТАБЛИЦУ'
ROLLBACK;
GO

```

```

DROP TABLE People

```

```

USE Радиостанции

```

GO

```
ENABLE TRIGGER WORK--активация триггера
ON DATABASE
GO
```

```
DROP TABLE People
```

```
USE Радиостанции
GO
```

```
DISABLE TRIGGER WORK--отключение триггера и удаление таблицы
ON DATABASE
DROP TABLE People
```

```
USE Радиостанции
GO
```

```
DROP TRIGGER WORK--удаление триггера
ON DATABASE
```

--СОЗДАНИЕ DML ТРИГГЕРА

```
USE Радиостанции
GO
```

```
CREATE TRIGGER WORK1--создание триггера
ON Сотрудники1
AFTER INSERT AS
DECLARE @KODE_S INT, @KODE_D INT
SELECT @KODE_S=Код_сотрудника, @KODE_D=Код_должности
FROM Сотрудники1
```

```
IF (@KODE_S>5 OR @KODE_D<5)
 BEGIN
 RAISERROR('Ошибка!',5,1)
 ROLLBACK TRAN
 END
GO
```

Запросы с демонстрацией вывода результата.



SQLQuery3.sql - L...7571LAR\Anna (54))\* x SQLQuery2.sql - L...7571LAR\Anna (53))\* SqlScript.sql - LA...37571LAR\Anna (52))\*

```

DECLARE @q1 INT, @q2 DATETIME, @q3 NUMERIC, @q4 VARCHAR
SET @q1 = 5
SET @q2 = '1990-01-01 00:00:00'
SET @q3 = 584.72
SET @q4 = '15'
SELECT CAST(@q1 AS FLOAT) AS float1
SELECT CAST(@q2 AS CHAR(50)) AS char1
SELECT CAST(@q3 AS FLOAT) AS float1
SELECT CAST(@q4 AS INT) AS int1

```

100 %

Результаты Сообщения

|   | float1 |
|---|--------|
| 1 | 5      |

|   | char1              |
|---|--------------------|
| 1 | янв 1 1990 12:00AM |

|   | float1 |
|---|--------|
| 1 | 585    |

|   | int1 |
|---|------|
| 1 | 1    |

Запрос успешно выполнен. | LAPTOP-37571LAR\SQLEXPRESS ... | LAPTOP-37571LAR\Anna (54) | master | 00:00:00 | 4 строк

LAPTOP-37571LAR\...dbo.Сотрудники1 SQLQuery5.sql - L...7571LAR\Anna (53))\* x SqlScript.sql - LA...37571LAR\Anna (52))\*

```

USE Радиостанции
GO
SELECT CASE Код_должности WHEN 1 THEN 'Один'
 WHEN 2 THEN 'Два'
 WHEN 3 THEN 'Три'
 WHEN 4 THEN 'Четыре'
 WHEN 5 THEN 'Пять'
 ELSE ''
END AS Код_должности_text
FROM Сотрудники1

```

100 %

Результаты Сообщения

|   | Код_должности_t... |
|---|--------------------|
| 1 | Один               |
| 2 | Два                |
| 3 | Три                |
| 4 | Четыре             |
| 5 | Пять               |

Запрос успешно выполнен. | LAPTOP-37571LAR\SQLEXPRESS ... | LAPTOP-37571LAR\Anna (53) | Радиостанции | 00:00:00 | 5 строк

SQLQuery5.sql - L...7571LAR\Anna (54))\* LAPTOP-37571LAR\...dbo.Сотрудники1 SQLQuery5.sql - L...7571LAR\Anna (53))\* x SqlScript.sql - LA...37571LAR\Anna (52))\*

```

SELECT CASE WHEN Код_должности = 1 THEN 'Один'
 WHEN Код_должности = 2 THEN 'Два'
 WHEN Код_должности = 3 THEN 'Три'
 WHEN Код_должности = 4 THEN 'Четыре'
 WHEN Код_должности = 5 THEN 'Пять'
 ELSE ''
END AS Код_должности_text1

FROM Сотрудники1

```

100 %

Результаты Сообщения

|   | Код_должности_te... |
|---|---------------------|
| 1 | Один                |
| 2 | Два                 |
| 3 | Три                 |
| 4 | Четыре              |
| 5 | Пять                |

Запрос успешно выполнен. LAPTOP-37571LAR\SQLEXPRESS ... LAPTOP-37571LAR\Anna (53) Радиостанции 00:00:00 5 строк

SQLQuery7.sql - L...7571LAR\Anna (54))\* x SQLQuery6.sql - L...7571LAR\Anna (53))\* LAPTOP-37571LAR\...dbo.Сотрудники1 SqlScript.sql - LA...37571LAR\Anna (52))\*

```

USE Радиостанции
GO
DECLARE @QUANTITY INT, @TEXT CHAR(100)
SET @QUANTITY = (SELECT COUNT(*) FROM Сотрудники1)
SET @TEXT = 'В ТАБЛИЦЕ БОЛЬШЕ 4 ЗАПИСЕЙ'
IF @QUANTITY >= 4
BEGIN
 SELECT @TEXT
END

```

100 %

Результаты Сообщения

|   | (Отсутствует имя столбца)  |
|---|----------------------------|
| 1 | В ТАБЛИЦЕ БОЛЬШЕ 4 ЗАПИСЕЙ |

Запрос успешно выполнен. LAPTOP-37571LAR\SQLEXPRESS ... LAPTOP-37571LAR\Anna (54) Радиостанции 00:00:00 1 строк

SqlScript.sql - LA...37571LAR\Anna (52))\* SQLQuery6.sql - L...7571LAR\Anna (54))\* SQLQuery6.sql - L...7571LAR\Anna (53))\* LAPTOP-37571LAR\...dbo.Сотрудники1

```
USE Радиостанции
GO
DECLARE @QUANTITY INT, @TEXT CHAR(100)
SET @QUANTITY = (SELECT SUM(Возраст) FROM Сотрудники1)
SET @TEXT = 'Сумма по полю'
IF @QUANTITY >= 150
BEGIN
 SELECT @TEXT + CAST (@QUANTITY AS CHAR(3))
END
```

100 %

Результаты Сообщения

| (Отсутствует имя столбца) |                   |
|---------------------------|-------------------|
| 1                         | Сумма по полю 160 |

Запрос успешно выполнен. LAPTOP-37571LAR\SQLEXPRESS ... LAPTOP-37571LAR\Anna (54) Радиостанции 00:00:00 1 строк

```
USE Радиостанции
GO
DECLARE @VAL INT
SET @VAL=(SELECT COUNT(*) FROM Сотрудники)
WHILE @VAL<5
BEGIN
 INSERT INTO Сотрудники
 (ФИО, Возраст)
 VALUES ('Не задано',0)
END
```

100 %

Сообщения

Выполнение команд успешно завершено.

```

IF OBJECT_ID (N'Date_of_Birth', N'U') IS NOT NULL
 DROP TABLE Date_of_Birth;

GO

CREATE TABLE Date_of_Birth (
 ID_Date INT IDENTITY(1,1),
 Date_Birth DATETIME NOT NULL
)

DECLARE @DATE DATETIME
DECLARE @N INT
SET @N=1
SET @DATE='28.01.2001'

WHILE @N<28
BEGIN
 INSERT INTO Date_of_Birth
 VALUES (@DATE+@N)
 SET @N=@N+1
END

SELECT * FROM Date_of_Birth

```

100 %

Результаты Сообщения

|   | ID_Date | Date_Birth              |
|---|---------|-------------------------|
| 1 | 1       | 2001-01-29 00:00:00.000 |
| 2 | 2       | 2001-01-30 00:00:00.000 |
| 3 | 3       | 2001-01-31 00:00:00.000 |
| 4 | 4       | 2001-02-01 00:00:00.000 |
| 5 | 5       | 2001-02-02 00:00:00.000 |
| 6 | 6       | 2001-02-03 00:00:00.000 |

Запрос успешно выполнен.

/\*

Создание временной таблицы через переменную типа TABLE

1. Создать локальную временную таблицу с названием TEMP1. Заполнить таблицу TEMP2 данными из имеющейся таблицы и вывести результат на экран.

2. Создать табличную переменную с названием TEMP2 и полями типа длинное целое со счетчиком, дата/время, целое, строка. Добавить в нее две записи с данными и вывести результат на экран.

\*/

USE Радиостанции;  
GO

IF OBJECT\_ID (N'TEMP1', N'U') IS NOT NULL

DROP TABLE #TEMP1;

```

GO
IF OBJECT_ID (N'TEMP2', N'U') IS NOT NULL

DROP TABLE #TEMP2;

GO

CREATE TABLE #TEMP1
(Код_сотрудника INT IDENTITY(1,1),
Должность VARCHAR(40),
Зарплата INT NOT NULL
)

INSERT INTO #TEMP1 (Должность, Зарплата) VALUES ('Звукорежиссер', 30000)
INSERT INTO #TEMP1 (Должность, Зарплата) VALUES ('Уборщик', 10000)
INSERT INTO #TEMP1 (Должность, Зарплата) VALUES ('Флорист', 20000)

CREATE TABLE #TEMP2
(Код_сотрудника INT IDENTITY(1,1),
Должность VARCHAR(40),
Зарплата INT NOT NULL
)

INSERT INTO #TEMP2 SELECT Должность, Зарплата FROM #TEMP1
SELECT *FROM #TEMP2

```

100 %

Создание временной таблицы через переменную типа TABLE

1. Создать локальную временную таблицу с названием TEMP1. Заполнить таблицу TEMP2 данными из имеющейся таблицы и вывести результат на экран.
2. Создать табличную переменную с названием TEMP2 и полями типа длинное целое со счетчиком, дата/время, целое, строка. Добавить в нее две записи с данными и вывести результат на экран.

```

/*
USE Радиостанции;
GO

IF OBJECT_ID (N'TEMP1', N'U') IS NOT NULL
DROP TABLE #TEMP1;
GO
IF OBJECT_ID (N'TEMP2', N'U') IS NOT NULL
DROP TABLE #TEMP2;
GO

CREATE TABLE #TEMP1
(Код_сотрудника INT IDENTITY(1,1),
Должность VARCHAR(40),
Зарплата INT NOT NULL

```

Результаты

|   | Код_сотрудн... | Должность     | Зарбот... |
|---|----------------|---------------|-----------|
| 1 | 1              | Звукорежиссер | 30000     |
| 2 | 2              | Уборщик       | 10000     |
| 3 | 3              | Флорист       | 20000     |

```
/*
Задания:

1. На базе имеющихся запросов (код SQL запросов нужно изменить таким образом
чтобы в них можно было передавать значения полей, по которым осуществляется

1. Процедуру без параметров.

2. Процедуру с входными параметрами

3. Процедуру с входными параметрами и выходным параметром
*/

CREATE PROCEDURE STAFF AS
SELECT COUNT(ФИО) FROM Сотрудники1
GO
EXEC STAFF
```

100 %

Результаты   Сообщения

|   | (Отсутствует имя столб... |
|---|---------------------------|
| 1 | 5                         |

```
CREATE PROCEDURE STAFF_W @МАКС_ВОЗРАСТ INT
AS SELECT MAX(Возраст)
FROM Сотрудники1
WHERE Возраст<=@МАКС_ВОЗРАСТ
GO
EXEC STAFF_W 50
```

100 %

Результаты   Сообщения

|   | (Отсутствует имя столб... |
|---|---------------------------|
| 1 | 48                        |

```

ALTER PROCEDURE STAFF_D @МИН_ВОЗРАСТ INT, @МАКС_ВОЗРАСТ INT, @СРЕДНИЙ_ВОЗРАСТ INT OUTPUT
AS SELECT @СРЕДНИЙ_ВОЗРАСТ=AVG(Возраст)
FROM Сотрудники1
WHERE Возраст BETWEEN @МИН_ВОЗРАСТ AND @МАКС_ВОЗРАСТ
GO

DECLARE @СРЕДНИЙ_ВОЗРАСТ INT
EXEC STAFF_D 18,60, @СРЕДНИЙ_ВОЗРАСТ OUTPUT
PRINT 'СРЕДНИЙ ВОЗРАСТ ОТ 18 ДО 60 РАВЕН'+CAST (@СРЕДНИЙ_ВОЗРАСТ AS CHAR(5))

```

100 % <

Сообщения

СРЕДНИЙ ВОЗРАСТ ОТ 18 ДО 60 РАВЕН32

```

USE Радиостанции
GO

IF OBJECT_ID (N'FUNC1', N'U') IS NOT NULL
DROP FUNCTION FUNC1;
GO

CREATE FUNCTION FUNC1(@FIO CHAR(10))
RETURNS TABLE
AS
RETURN (SELECT *
FROM Сотрудники1
WHERE ФИО=@FIO AND Возраст=54)
GO

SELECT * FROM FUNC1('Семгов')

```

91 % <

Результаты Сообщения

|   | Код_сотрудн... | Код_должно... | ФИО    | Возра... | Пол     |
|---|----------------|---------------|--------|----------|---------|
| 1 | 2              | 1             | Семгов | 54       | мужской |

```

/*
Создание и использование триггеров.

1. Создать DDL - триггер в базе данных варианта задания для события изменения таблицы.
Продемонстрировать работу триггера на примере изменения таблицы базы данных.

2. Создать DML - триггер для проверки бизнес-правила в базе данных варианта задания.
Триггер должен проверять возможность изменения записей одной таблицы на основе проверки записей другой таблицы
*/

USE Радиостанции
GO

CREATE TRIGGER WORK--
ON DATABASE
FOR DROP_TABLE
AS
PRINT 'НЕВОЗМОЖНО УДАЛИТЬ ТАБЛИЦУ'
ROLLBACK;
GO

DROP TABLE People

```

91 % <

Сообщения

НЕВОЗМОЖНО УДАЛИТЬ ТАБЛИЦУ  
Сообщение 3609, уровень 16, состояние 2, строка 2  
Транзакция завершилась в триггере. Выполнение пакета прервано.

```

DROP TABLE People

USE Радиостанции
GO

ENABLE TRIGGER WORK--активация триггера
ON DATABASE
GO

DROP TABLE People

```

91 % <

Сообщения

НЕВОЗМОЖНО УДАЛИТЬ ТАБЛИЦУ  
Сообщение 3609, уровень 16, состояние 2, строка 2  
Транзакция завершилась в триггере. Выполнение пакета прервано.



```
DISABLE TRIGGER WORK--отключение триггера и удаление таблицы
ON DATABASE
DROP TABLE People
```

91 % <



Сообщения

Выполнение команд успешно завершено.

```
USE Радиостанции
GO
```

```
DROP TRIGGER WORK--удаление триггера
ON DATABASE
```

91 % <



Сообщения

Выполнение команд успешно завершено.

```
--СОЗДАНИЕ DML ТРИГГЕРА
USE Радиостанции
GO

CREATE TRIGGER WORK1--создание триггера
ON Сотрудники1
AFTER INSERT AS
DECLARE @KODE_S INT, @KODE_D INT
SELECT @KODE_S=Код_сотрудника, @KODE_D=Код_должности
FROM Сотрудники1
IF (@KODE_S>5 OR @KODE_D<5)
BEGIN
 RAISERROR('Ошибка!',5,1)
 ROLLBACK TRAN
END
GO
```

91 %



Сообщения

Выполнение команд успешно завершено.