### Карпухина Анна

### Вариант 6

#### Задание 1

# Задание 2

```
| December | Passive | Pa
```

```
1. Объявить переменные типа FLOAT, CHAR, TINYINT, BIT.
Присвоить значения, соответствующие типан.
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 типан.
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 = 'Anha'

SET @p3 = 2

SET @p4 = 'True'

SELECT CAST(@p1 AS INT) AS int1

SELECT CAST(@p1 AS INT) 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 @q3 = 584.72

SET @q3 = 584.72

SET @q3 = 584.72

SET ECCT CAST(@p3 AS FLOAT) AS float1

SELECT CAST(@p4 AS INT) AS into1
```

## 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 Регет и вывести на экран.
<sup>|</sup>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 = 'BPEM9 : '

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: Использую 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
∃SELECT CASE Код_должности WHEN 1 THEN 'Один'
                                WHEN 2 THEN 'Два'
                                WHEN 3 THEN 'Tpu'
                                WHEN 4 THEN 'Четыре'
                                WHEN 5 THEN 'Пять'
                                ELSE ''
          END AS Код_должности_text
FROM Сотрудники1
```

```
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
GU

DECLARE @QUANTITY INT, @TEXT CHAR(100)

SET @QUANTITY = (SELECT COUNT(*) FROM Сотрудники1)

SET @TEXT = 'В ТАБЛИЦЕ БОЛЬШЕ 4 ЗАПИСЕЙ'

DIF @QUANTITY>=4
      SELECT @TEXT
USE Радиостанции
□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))
```

```
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
 G0
□ALTER FUNCTION GetSegU
 (@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
```

```
В/*

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 (
         Date_Birth DATETIME NOT NULL
  DECLARE @DATE DATETIME
  DECLARE @N INT
  SET @N=1
  SET @DATE='28.01.2001'
 ⊨WHILE @N<28
         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 ТЕМР1. Заполнить таблицу ТЕМР2 данными
из имеющейся таблицы и вывести результат на экран.
2. Создать табличную переменную с названием ТЕМР2 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;
IF OBJECT_ID (N'TEMP1', N'U') IS NOT NULL
DROP TABLE #TEMP1;
IF OBJECT_ID (N'TEMP2', N'U') IS NOT NULL
DROP TABLE #TEMP2;
G0
CREATE TABLE #TEMP1
(Код_сотрудника
                     INT IDENTITY(1,1),
Должность VARCHAR(40),
Заработок INT NOT NULL
)
INSERT INTO #TEMP1 (Должность, Заработок) VALUES ('Звукорежиссер', 30000)
```

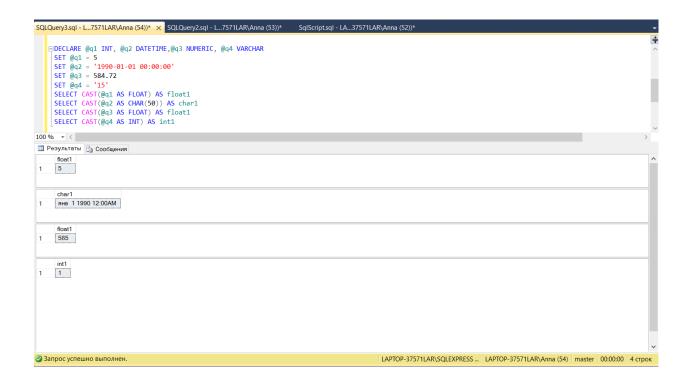
```
INSERT INTO #TEMP1 (Должность, Заработок) VALUES ('Уборщик',10000)
INSERT INTO #TEMP1 (Должность, Заработок) VALUES ('Флорист',20000)
CREATE TABLE #TEMP2
                        INT IDENTITY(1,1),
(Код сотрудника
Должность VARCHAR(40),
Заработок INT NOT NULL
INSERT INTO #TEMP2 SELECT Должность, Заработок 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
∃EXEC STAFF
CREATE PROCEDURE STAFF W @MAKC_BO3PACT INT
∃AS SELECT MAX(Возраст)
FROM Сотрудники1
WHERE Bospact<=@MAKC_BOSPACT
 GO
≡EXEC STAFF_W 50
ALTER PROCEDURE STAFF D @MUH_BO3PACT_INT, @MAKC_BO3PACT_INT, @CPEДНИЙ_BO3PACT_INT_OUTPUT
∃AS SELECT @СРЕДНИЙ_ВОЗРАСТ=AVG(Возраст)
FROM Сотрудники1
WHERE BO3PACT BETWEEN @MUH_BO3PACT AND @MAKC_BO3PACT
∃DECLARE @СРЕДНИЙ_ВОЗРАСТ INT
EXEC STAFF_D 18,60, @СРЕДНИЙ_ВОЗРАСТ OUTPUT
PRINT 'СРЕДНИЙ ВОЗРАСТ ОТ 18 ДО 60 PABEH'+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
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
```

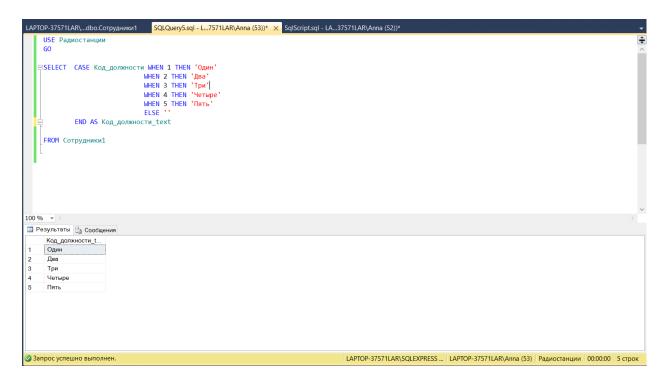
```
DECLARE @SUM FLOAT
DECLARE @N INT
DECLARE @VAL FLOAT
SET @N=1
SET @VAL=1
SET @SUM=0
WHILE @VAL > 0.01
BEGIN
SET @VAL = (SELECT dbo.GetSeqU(@N+1))-(SELECT dbo.GetSeqU(@N))
SET @SUM=@SUM+@VAL
SET @N=@N+1;
END
SELECT @SUM
USE Радиостанции
IF OBJECT_ID (N'FUNC1', N'U') IS NOT NULL
DROP FUNCTION FUNC1;
GO
CREATE FUNCTION FUNC1(@FIO CHAR(10))
RETURNS TABLE
AS
RETURN (SELECT *
FROM Сотрудники1
WHERE ΦИΟ=@FIO AND Bospact=54)
GO
SELECT * FROM FUNC1('Cemrob')
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
G0
-- COЗДАНИЕ DDL ТРИГГЕРА
CREATE TRIGGER WORK--создание триггера
ON DATABASE
FOR DROP_TABLE
AS
PRINT 'HEBO3MOЖНО УДАЛИТЬ ТАБЛИЦУ'
ROLLBACK;
G0
DROP TABLE People
```

USE Радиостанции

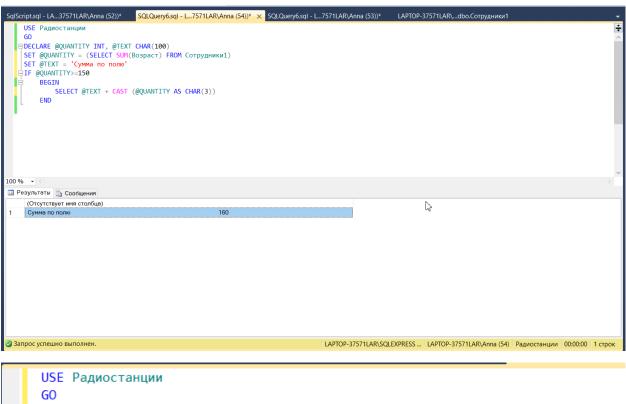
```
ENABLE TRIGGER WORK--активация триггера
ON DATABASE
GO
DROP TABLE People
USE Радиостанции
DISABLE TRIGGER WORK--отключение триггера и удаление таблицы
ON DATABASE
DROP TABLE People
USE Радиостанции
G0
DROP TRIGGER WORK--удаление триггера
ON DATABASE
--COЗДАНИЕ DML ТРИГГЕРА
USE Радиостанции
GO
CREATE 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)</pre>
       BEGIN
              RAISERROR('Ошибка!',5,1)
              ROLLBACK TRAN
       END
G0
```

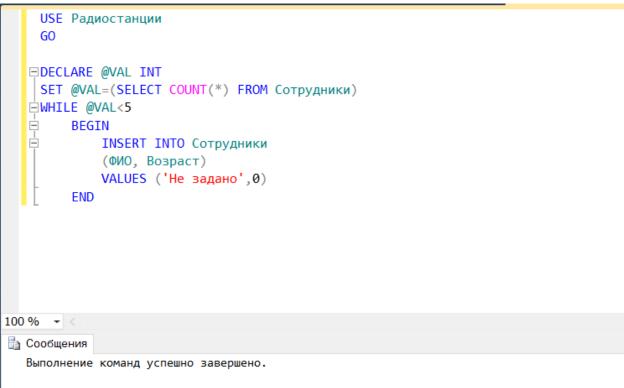
Запросы с демонстрацией вывода результата.











```
□IF OBJECT_ID (N'Date_of_Birth', N'U') IS NOT NULL
 DROP TABLE Date_of_Birth;
 G0
□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
Ė
         INSERT INTO Date_of_Birth
         VALUES (@DATE+@N)
         SET @N=@N+1
     END
SELECT * FROM Date of Birth
```

	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
_		2001 02 02 00 00 00

Запрос успешно выполнен.

Создание временной таблицы через переменную типа TABLE

- 1. Создать локальную временную таблицу с названием ТЕМР1. Заполнить таблицу ТЕМР2 данными из имеющейся таблицы и вывести результат на экран.
- 2. Создать табличную переменную с названием ТЕМР2 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;
```

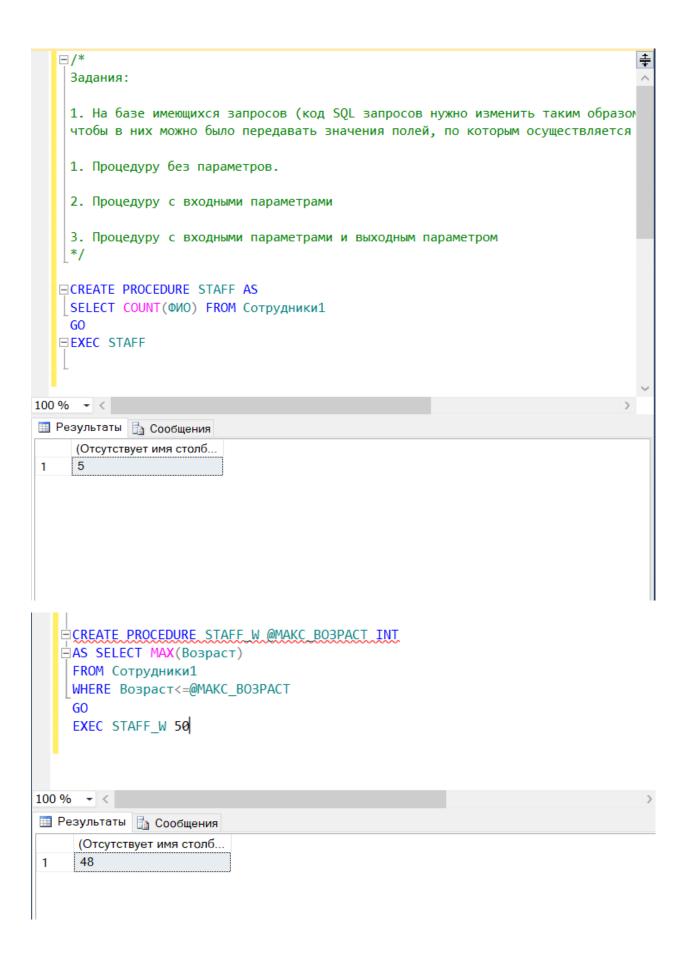
```
IF OBJECT ID (N'TEMP2', N'U') IS NOT NULL
DROP TABLE #TEMP2;
GO
CREATE TABLE #TEMP1
(Код сотрудника INT IDENTITY(1,1),
Должность VARCHAR(40),
Заработок INT NOT NULL
INSERT INTO #TEMP1 (Должность, Заработок) VALUES ('Звукорежиссер', 30000)
INSERT INTO #TEMP1 (Должность, Заработок) VALUES ('Уборщик',10000)
INSERT INTO #TEMP1 (Должность, Заработок) VALUES ('Флорист', 20000)
CREATE TABLE #TEMP2
                          INT IDENTITY(1,1),
(Код_сотрудника
Должность VARCHAR(40),
Заработок INT NOT NULL
INSERT INTO #TEMP2 SELECT Должность, Заработок FROM #TEMP1
SELECT *FROM #TEMP2
   Создание временной таблицы через переменную типа TABLE
   1. Создать локальную временную таблицу с названием ТЕМР1. Заполнить таблицу ТЕМР2 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),

Заработок INT NOT NULL
 🛄 Результаты 🔓 Сообщения
  Код сотрудн. Должность Зарабо
1 Звукорежиссер 30000
2 Уборщик 10000
3 Флорист 20000
```



```
🛱 ALTER PROCEDURE STAFF D @MUH_BO3PACT_INT, @MAKC_BO3PACT_INT, @CPEДНИЙ_BO3PACT_INT_OUTPUT
  ➡AS SELECT @СРЕДНИЙ_ВОЗРАСТ=AVG(Возраст)
   FROM Сотрудники1
   WHERE BOSPACT BETWEEN @MUH_BOSPACT AND @MAKC_BOSPACT
   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 РАВЕНЗ2
     USE Радиостанции
     G0
   □IF OBJECT_ID (N'FUNC1', N'U') IS NOT NULL
     DROP FUNCTION FUNC1;
     G0
   □CREATE FUNCTION FUNC1(@FIO CHAR(10))
     RETURNS TABLE
     AS
     RETURN (SELECT *
     FROM Сотрудники1
     WHERE ΦИΟ=@FIO AND Возраст=54)
     G0
     SELECT * FROM FUNC1('Cemrob')
91 %
🖽 Результаты 🚹 Сообщения
       Код_сотрудн...
                        Код_должно...
                                         ФИО
                                                  Возра...
                                                           Пол
 1
                        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
   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
       G0
     □ENABLE TRIGGER WORK--активация триггера
       ON DATABASE
       G0
     ■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 G0 □DROP TRIGGER WORK--удаление триггера ON DATABASE 91 % - < 🛅 Сообщения

Выполнение команд успешно завершено.

```
-- COЗДАНИЕ 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.