Лабораторная работа 4.2

Встроенные функции

1. Теоретическая часть

На языке Transact-SQL используется множество различных *типов данных*. Всех их можно разделить на следующие группы:

Числовые типы данных: BIT (значение 0 или 1), **TINYINT** (от 0 до 255), **SMALLINT** (от

-32768 до 32767), INT (от -2147483648 до 2147483647), **BIGINT** (от -9223372036854775808 до 9223372036854775807), **DECIMAL** (числа с фиксированной точностью), **NUMERIC**: (аналогичен типу **DECIMAL**), **SMALLMONEY** (дробные значения от -214748.3648 до 214748.3647), **MONEY** (дробные значения от -922337203685477.5808 до 922337203685477.5807), **FLOAT** (от -1.79E+308 до 1.79E+308), REAL (числа от -340E+38 до 3.40E+38);

Типы данных, представляющие дату и время: DATE (дата от 01/01/0001 до 31/12/9999), TIME (время в диапазоне от 00:00:00.00000000 до 23:59:59.9999999), **DATETIME** (дата и время от 01/01/1753 до 31/12/9999), **DATETIME2** (дата и время от 01/01/0001 00:00:00.00000000 до 31/12/9999 23:59:59.9999999), **SMALLDATETIME** (дата и время от 01/01/1900 до 06/06/2079), **DATETIMEOFFSET** (дата и время от 01/01/0001 до 31/12/9999);

Строковые типы данных: СНАК (фиксированная строка длиной от 1 до 8000 символов), VARCHAR (переменная строка длиной от 1 до 8000 символов), NCHAR (Unicode – фиксированная строка длиной от 1 до 4000 символов), NVARCHAR (Unicode – переменная строка длиной от 1 до 4000 символов), TEXT и NTEXT (устаревшие, не рекомендуется использовать);

Бинарные типы данных: BINARY (фиксированные бинарные данные от 1 до 8000 байт), **VARBINARY** (переменные бинарные данные от 1 до 8000 байт), IMAGE (устаревшая, не рекомендуется использовать);

Другие типы данных: UNIQUEIDENTIFIER (уникальный идентификатор GUID), TIMESTAMP (номер версии строки в таблице), CURSOR (набор строк таблицы), HIERARCHYID (позиция в иерархии), SQL_VARIANT (данные любого типа), XML (документы или фрагменты XML), TABLE (таблица), GEOGRAPHY (географические данные, такие как широта и долгота), GEOMETRY (координаты на плоскости).

2.2. Встроенные функции Transact-SQL

 Φ ункции SQL производят действия с данными и возвращают результат. Встроенные Φ ункции делятся на три основные группы:

- *скалярные функции* обрабатывают одиночное значение и возвращают одно значение. Их можно использовать везде, где допускается применение выражений.
- *агрегатные функции* используются для получения обобщающих значений. Они, в отличие от скалярных функций, оперируют значениями столбцов множества строк; функции для списка значений.

Скалярные функции бывают следующих категорий:

- *строковые функции* выполняют определенные действия над строками и возвращают строковые или числовые значения;
- **числовые функции** возвращают числовые значения на основании заданных в аргументе значений того же типа;
- функции времени и даты выполняют различные действия над входными значениями времени и даты и возвращают строковое, числовое значение или значение в формате даты и времени;

– функции преобразования типа.

Список часто используемых строковых функций:

LEN(строка)	возвращает количество символов в заданной строке
· · · · ·	*
TRIM(строка) TRIM([символ FROM] строка)	удаляет символ пробела или другие заданные символы в начале и в конце строки.
LTRIM(строка)	удаляет начальные пробелы из заданной строки
RTRIM(строка)	удаляет конечные пробелы из заданной строки
CHARINDEX(подстрока, строка) CHARINDEX(подстрока, строка, начальная позиция)	возвращает индекс, по которому находится первое вхождение подстроки в строке.
PATINDEX('%шаблон%', строка)	возвращает индекс, по которому находится первое вхождение определенного шаблона в строке
LEFT(строка, число)	возвращает с начала строки определенное количество символов
RIGHT(строка, число)	возвращает с конца строки определенное количество символов
SUBSTRING(строка, начальная позиция, длина)	возвращает подстроку заданной длиной, начиная с данной позиции
REPLACE(строка, подстрока, замена)	заменяет одну подстроку другой
REVERSE(строка)	переворачивает строку наоборот
СОNCAT(строка1, строка2 [, строкаN]	объединяет заданные строки в одну
LOWER(строка)	переводит строку в нижний регистр
UPPER (строка)	переводит строку в верхний регистр
SPACE(число)	возвращает заданное количество пробелов
REPLICATE(строка, число)	повторяет значение строки указанное число раз
STUFF(строка, начальная позиция, количество, замена)	удаляет указанное количество символов первой строки в начальной позиции и вставляет на их место замену.

Список часто используемых числовых функций:

ABS(число)	возвращает абсолютное значение числа
CEILING(число)	возвращает наименьшее целое, большее или равное заданного числа.
FLOOR(число)	возвращает наибольшее целое число, меньшее или равное заданного числа
POWER(число, степень)	возвращает значение указанного выражения, возведенное в заданную степень
RAND([начальное значение])	возвращает псевдослучайное значение от 0 до 1
ROUND(число, точность)	возвращает число, округленное до указанной точности
SIGN(число)	возвращает положительное (+1), нулевое (0) или отрицательное (-1) значение, обозначающее знак заданного выражения

SQRT(число)	возвращает квадратный корень данного числа
SQUARE(число)	возвращает квадрат указанного числа
PI()	возвращает константное значение π
ACOS(число)	возвращает угол в радианах, косинус которого задан – арккосинус.
ASIN(число)	возвращает угол в радианах, синус которого задан – арксинус.
ATAN(число)	возвращает угол в радианах, тангенс которого задан – арктангенс.
COS(число)	возвращает косинус указанного угла в радианах.
SIN(число)	возвращает синус указанного угла в радианах.
TAN(число)	возвращает тангенс указанного угла в радианах.
СОТ(число)	возвращает котангенс указанного угла в радианах
DEGREES(число)	возвращает для значения угла в радианах соответствующее значение в градусах.
RADIANS(число)	возвращает для значения угла в градусах соответствующее значение в радианах
ЕХР(число)	возвращает экспонент заданного числа
LOG(число)	возвращает натуральный логарифм указанного числа
LOG(число, основа)	возвращает логарифм указанного числа
LOG10(число)	возвращает десятичный логарифм указанного числа

Список часто используемых функций времени и даты:

GETDATE()	возвращает текущую дату и время
CURRENT_TIMEZONE()	возвращает имя часового пояса
GETUTCDATE()	возвращает текущую дату и время по Гринвичу (UTC/GMT)
DAY (дата)	возвращает день месяца указанной даты
MONTH(дата)	возвращает номер месяца указанной даты
YEAR(дата)	возвращает год указанной даты
DATEPART(часть, дата)	возвращает целое число, представляющее указанную часть заданной даты
DATENAME(часть, дата)	возвращает строку символов, представляющую указанную часть заданной даты
DATEADD(часть, число, дата)	добавляет указанное целое число со знаком к части входного значения даты, а затем возвращает это измененное значение
DATEDIFF(часть, начальная дата, конечная дата)	возвращает разницу как целое число со знаком между частями заданных дат
EOMONTH(дата)	возвращает последний день месяца, заданной даты

Для функций времени и даты используются следующие аргументы как часть даты и времени:

Часть даты и времени	Сокращения
year	уу, уууу
quarter	qq, q
month	mm, m
dayofyear	dy, y
day	dd, d
week	wk, ww
weekday	dw
hour	hh
minute	mi, n
second	ss, s
millisecond	ms
microsecond	mcs
nanosecond	ns
tzoffset	tz
iso_week	isowk, isoww

Список часто используемых функций преобразования:

CAST(выражение AS тип)	преобразуют выражение в заданный тип	
CONVERT(тип, выражение [, стиль])		
ASCII(строка)	возвращает код ASCII первого символа указанного символьного выражения	
UNICODE(строка)	возвращает код Юникод первого символа указанного символьного выражения	
CHAR(число)	возвращает символ ASCII с указанным кодом	
NCHAR(число)	возвращает символ Юникода с указанным кодом	
STR(число)	возвращает символьные данные, преобразованные из числовых данных	

Список часто используемых функций проверки значений:

Chucok tucho uchonosychioli	енисок чисто исполозуемых функции проверки зничении.	
ISDATE(выражение)	возвращает 1, если выражение имеет допустимое значение типа даты и времени, иначе возвращает значение 0	
ISNUMERIC(выражение)	возвращает 1, если выражение имеет допустимое значение числовой тип данных, иначе возвращает 0	
ISNULL(выражение, замена)	заменяет значение NULL указанным замещающим значением	
COALESCE(выражение[,n])	вычисляет аргументы по порядку и возвращает текущее значение первого выражения, изначально не вычисленного как NULL.	

Особое место среди встроенных скалярных функций языка SQL занимают функции вывода, которые являются разновидностью CASE-выражений. Функция CASE проверяет значение некоторого выражения, и в зависимости от результата проверки может возвращать тот или иной результат.

Выражение CASE имеет два формата:

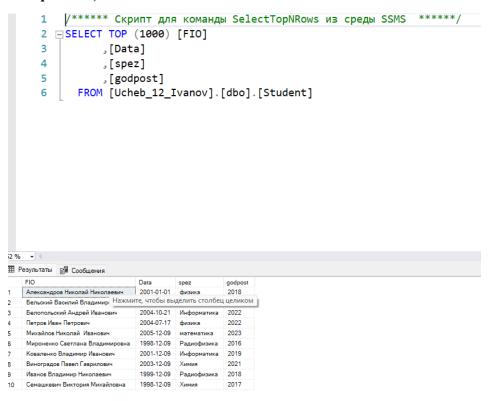
- простое выражение **CASE** для определения результата сравнивает выражение с набором простых выражений;
- поисковое выражение **CASE** для определения результата вычисляет набор логических выражений.

Оба формата поддерживают дополнительный аргумент ELSE.

Функция *IIF*(условие, выражение_если_истина, выражение_если_ложь) — возвращает одно из двух значений в зависимости от того, принимает логическое выражение значение true или false.

1. Практическая часть

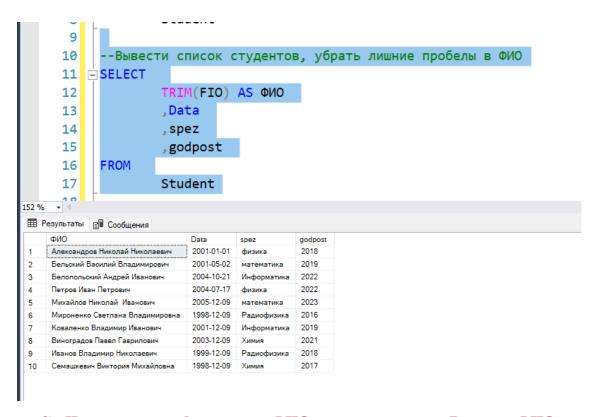
Воспользуемся из таблицей [Student] из предыдущих лабораторных работ (см. Лаб.раб. 4.1)



1) Вывести ФИО студентов и длину ФИО

```
1 _use Ucheb_12_Ivanov
            -- Вывести ФИО студентов и длину ФИО:
      4 SELECT
                      FIO
      5
       6
                      ,LEN(FIO) AS Количество_символов
            FROM
      7
                      Student
      8
       9
■ Результаты   Сообщения
    Петров Иван Петрович
Михайлов Николай Иванович
    Коваленко Владимир Иванович
    Виноградов Павел Гаврилович
```

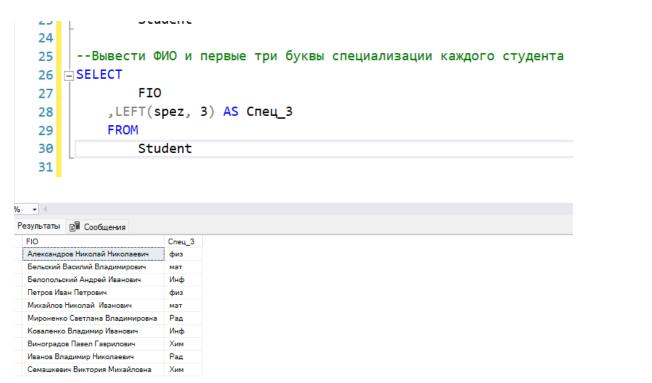
2) Вывести список студентов, убрать лишние пробелы в ФИО



3) Найти позиции буквы «о» в ФИО каждого студента. Вывести ФИО и позицию

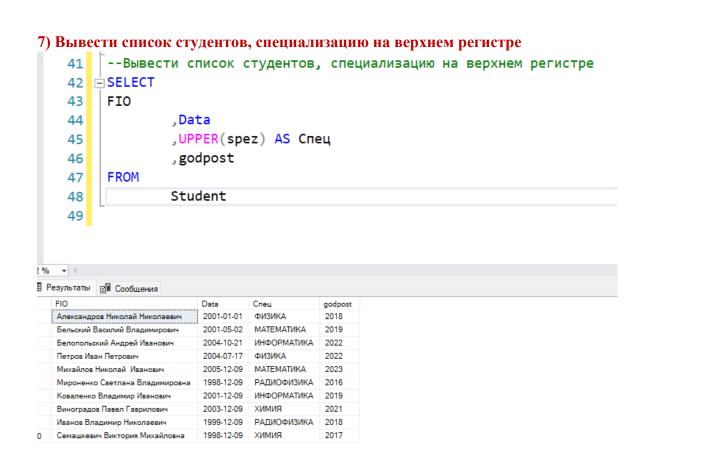
```
Student
    18
          --Найти позиции буквы «о» в ФИО каждого студента. Вывести ФИО и позицию
    19 SELECT
    20
                      FIO
                 , CHARINDEX('o', FIO) AS Позиция_о
    21
                 FROM
    22
                      Student
    23
    24
    25
% +
Результаты 📳 Сообщения
  FIO
                            Позиция о
                          10
 Александров Николай Николаевич
  Бельский Василий Владимирович
  Белопольский Андрей Иванович
  Петров Иван Петрович
  Михайлов Николай Иванович
  Мироненко Светлана Владимировна 4
  Коваленко Владимир Иванович
  Виноградов Павел Гаврилович
  Иванов Владимир Николаевич
  Семашкевич Виктория Михайловна 16
```

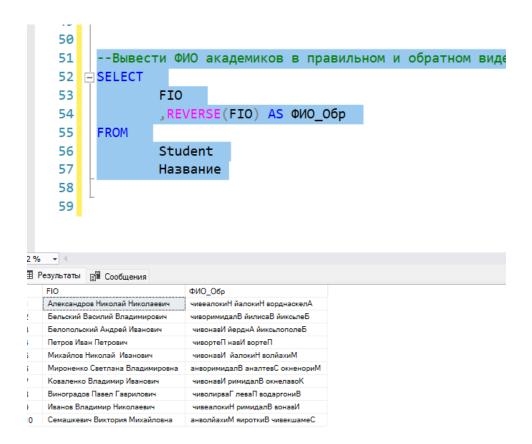
4) Вывести ФИО и первые три буквы специализации каждого студента

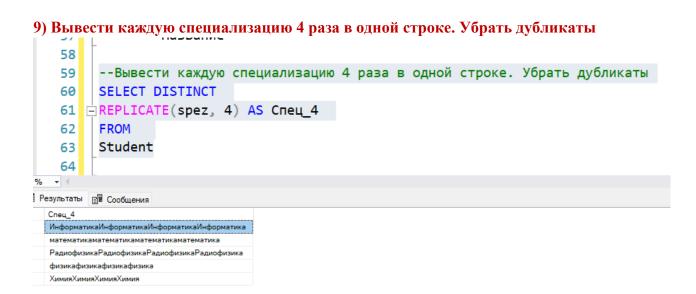


6) Вывести список студентов, заменить специализацию «математика» на «прикладная математика»

```
31
                 --Вывести список студентов, заменить специализацию «математика»
     32
     33
                  --на «прикладная математика»
     34
         SELECT
     35
                       FIO
     36
                        ,Data
     37
                        ,REPLACE(spez, 'математика', 'прикладная математика') AS Спец
                        , godpost
     38
            FROM
     39
                       Student
     40
     41
% + (
Результаты 🖺 Сообщения
                             Data
  Александров Николай Николаевич 2001-01-01 физика
                                                        2018
   Бельский Василий Владимирович
                             2001-05-02 прикладная математика
                                                        2019
                          2004-10-21 Информатика
   Белопольский Андрей Иванович
                                                        2022
   Петров Иван Петрович
                            2004-07-17 физика
                            2005-12-09 прикладная математика 2023
   Михайлов Николай Иванович
   Мироненко Светлана Владимировна 1998-12-09 Радиофизика 
Коваленко Владимир Иванович 2001-12-09 Информатика
   2019
                                                       2021
   Иванов Владимир Николаевич
                            1999-12-09 Радиофизика
                                                       2018
  Семашкевич Виктория Михайловна 1998-12-09 Химия
                                                        2017
```







10) Вывести абсолютное значение тригонометрических функций на точке π

```
65
    66
          --Вывести абсолютное значение тригонометрических функций на точке п
    67
          SELECT
                   ABS(COS(PI()))
                                       AS Косинус_Пи
    68
    69
                   ,ABS(SIN(PI())) AS Синус_Пи
                    ,ABS(TAN(PI())) AS Тангенс Пи
    70
    71
                    ,ABS(COT(PI())) AS KoTahrehc_Πu
    72
    73
% + (
Результаты 🖺 Сообщения
  Косинус_Пи Синус_Пи
                         Тангенс Пи
                                       КоТангенс Пи
          1,22464679914735E-16 1,22464679914735E-16 8,16561967659769E+15
```

11) Вывести число 132.456, округленное с точностью от 3 до -3

```
72
    73
          --Вывести число 132.456, округленное с точностью от 3 до -3
    74
       ĖSELECT
                   ROUND(123.456, 3)
                                          AS Okp3
    75
                   ,ROUND(123.456, 2)
    76
                                         AS Okp2
                  ,ROUND(123.456, 1)
    77
                                          AS Okp1
    78
                   ,ROUND(123.456, 0)
                                         AS Okp0
                  ,ROUND(123.456, -1) AS Oκp_1
    79
                   ,ROUND(123.456, -2)
    80
                                         AS Okp_2
    81
                   ,ROUND(123.456, -3)
                                         AS 0κp_3
    82
2 %
В Результаты В Сообщения
  Окр3 Окр2
             Окр1 Окр0
                       Окр 1 Окр 2
                                   Окр 3
  123.456 123.460 123.500 123.000 120.000 100.000 0.000
```

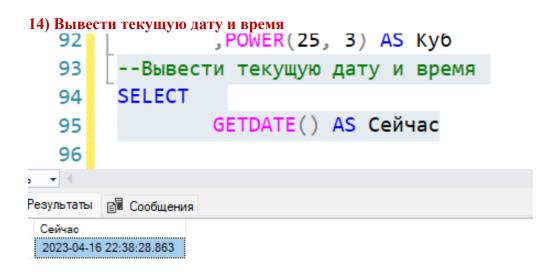
12) Вывести наименьшее целое число, которое больше или равно 123.456, и наибольшее целое число, которое меньше или равно 123.456

```
82
    83 🖃 -- Вывести наименьшее целое число, которое больше или равно 123.456,
          --и наибольшее целое число, которое меньше или равно 123.456
    84
    85
          SELECT
    86
                   CEILING(123.456) AS Больше
    87
                  ,FLOOR(123.456)
                                      AS Меньше
    88
Результаты В Сообщения
  Больше Меньше
       123
  124
```

13) Вывести квадратный корень, квадрат и куб числа 25

```
88
89
-- Вывести квадратный корень, квадрат и куб числа 25

SELECT
SQRT(25) AS Корень
SQUARE(25) AS Квадрат
92
3 POWER(25, 3) AS Куб
Результаты Сообщения
Корень Квадрат Куб
5 625 15625
```



15) Вывести день, месяц, год, час, минуту, секунду, номер квартала, номер недели, день года, день недели для текущей даты и времени

```
SELECT
     94
                  GETDATE() AS Сейчас
     95
     96
        ⊟--Вывести день, месяц, год,
          --час, минуту, секунду,
     97
          --номер квартала, номер недели,
     98
     99
          --день года, день недели для текущей даты и времени
    100
    101 SELECT
    102
                   DAY(GETDATE()) AS День
                   ,MONTH(GETDATE()) AS Месяц
    103
                   ,YEAR(GETDATE()) AS Год
    104
                   ,DATEPART(HOUR, GETDATE()) AS Yac
    105
    106
                   ,DATEPART(MINUTE, GETDATE()) AS Минута
                   ,DATEPART(SECOND, GETDATE()) AS Секунда
    107
                   ,DATEPART(QUARTER, GETDATE()) AS Квартал
    108
                   ,DATEPART(WEEK, GETDATE()) AS Неделя
    109
                   ,DATEPART(DAYOFYEAR, GETDATE()) AS День_года
    110
    111
                   ,DATEPART(WEEKDAY, GETDATE()) AS День_недели
    112
152 %
Результаты В Сообщения
   День Месяц Год Час Минута Секунда Квартал Неделя День_года День_недели
          2023 22 42
  16 4
                      8 2
                                   16
                                         106
```

16) Вывести дату 100 дней назад от текущей

```
112
113
114
115
116

--Вывести дату 100 дней назад от текущей
SELECT

DATEADD(DAY, -100, GETDATE()) AS День_100_Назад

Результаты
В Сообщения

День_100_Назад
2023-01-06 22:44:42.000
```

17) Иванов И.И. родился 8 июля 1895 года, скончался 12 апреля 1971 года. Вывести количество прожитых дней

18) Вывести ФИО и время года рождения каждого студента

```
120
  121
        --Вывести ФИО и время года рождения каждого студента
  122 SELECT
                FIO
  123
           , CASE MONTH(Data)
  124
                    WHEN 3 THEN 'Becha'
  125
                   WHEN 4 THEN 'Becha'
  126
                   WHEN 5 THEN 'Becha'
  127
                   WHEN 6 THEN 'Лето'
  128
                   WHEN 7 THEN 'Лето'
  129
                  WHEN 8 THEN 'Лето'
  130
                  WHEN 9 THEN 'Ocenb'
  131
                   WHEN 10 THEN 'Осень'
  132
                  WHEN 11 THEN 'Осень'
  133
                   ELSE 'Зима'
  134
                   END AS Времени года
  135
  136 FROM Student
% -
Результаты 🖺 Сообщения
 FIO
  Александров Николай Николаевич Зима
  Белопольский Андрей Иванович
  Петров Иван Петрович
                    Лето
  Михайлов Николай Иванович
                    Зима
  Мироненко Светлана Владимировна Зима
  Коваленко Владимир Иванович
  Виноградов Павел Гаврилович
                   Зима
  Иванов Владимир Николаевич
                    Зима
0 Семашкевич Виктория Михайловна Зима
19) Вывести список студкетов. Для каждого студента, в зависимости от возраста, при
поступлении «молодой» или «старше» в дополнительном столбце
 137 🚊 --Вывести список студкетов.
 138
         --Для каждого студента, в зависимости от возраста,
 139
         --при поступлении «молодой» или «старше» в дополнительном столбце
 140 SELECT
 141
                   FIO
                   ,Data
 142
 143
                   , spez
 144
                   ,godpost
                   ,IIF(godpost - Year(Data) <= 18, 'Молодой','старше')
 145
         AS Возраст_при_поступлении
 146
```

147

FROM Student

```
149 SELECT
                  FIO
  150
  151
                  , Data
                  , CASE
  152
                       WHEN (MONTH(Data)=3 AND DAY(Data) >= 21)
  153
  154
         OR (MONTH(Data)=4 AND DAY(Data) <= 20) THEN 'OBEH'
                       WHEN (MONTH(Data)=4 AND DAY(Data) >= 21)
  155
         OR (MONTH(Data)=5 AND DAY(Data) <= 21) THEN 'Телец'
  156
  157
                       WHEN (MONTH(Data)=5 AND DAY(Data) >= 22)
         OR (MONTH(Data)=6 AND DAY(Data) <= 21) THEN 'Близнецы'
  158
                       WHEN (MONTH(Data)=6 AND DAY(Data) >= 22)
  159
  160
         OR (MONTH(Data)=7 AND DAY(Data) <= 22) THEN 'Pak'
                      WHEN (MONTH(Data)=7 AND DAY(Data) >= 23)
  161
         OR (MONTH(Data)=8 AND DAY(Data) <= 21) THEN 'Лев'
  162
                       WHEN (MONTH(Data)=8 AND DAY(Data) >= 22)
  163
         OR (MONTH(Data)=9 AND DAY(Data) <= 23) THEN 'Дева'
  164
                       WHEN (MONTH(Data)=9 AND DAY(Data) >= 24)
  165
         OR (MONTH(Data)=10 AND DAY(Data) <= 23) THEN 'Весы'
  166
  167
                      WHEN (MONTH(Data)=10 AND DAY(Data) >= 24)
         OR (MONTH(Data)=11 AND DAY(Data) <= 22) THEN 'Скорпион'
  168
                       WHEN (MONTH(Data)=11 AND DAY(Data) >= 23)
  169
         OR (MONTH(Data)=12 AND DAY(Data) <= 22) THEN 'Стрелец'
  170
                       WHEN (MONTH(Data)=12 AND DAY(Data) >= 23)
  171
  172
         OR (MONTH(Data)=1 AND DAY(Data) <= 20) THEN 'Kosepor'
                       WHEN (MONTH(Data)=1 AND DAY(Data) >= 21)
  173
         OR (MONTH(Data)=2 AND DAY(Data) <= 19) THEN 'Водолей'
  174
                       WHEN (MONTH(Data)=2 AND DAY(Data) >= 20)
  175
         OR (MONTH(Data)=3 AND DAY(Data) <= 20) THEN 'Рыбы'
  176
                  END AS Знак_зодиака
  177
  178
         FROM Student
% +
Результаты 🗐 Сообщения
                       Data
                              Знак_зодиака
  Александров Николай Николаевич 2001-01-01 Козерог
  Бельский Василий Владимирович
                       2001-05-02
                               Телец
  Белопольский Андрей Иванович
                       2004-10-21
                               Весы
  Петров Иван Петрович
                       2004-07-17 Pak
  Михайлов Николай Иванович
                       2005-12-09 Стрелец
                       1998-12-09 Стрелец
  Мироненко Светлана Владимировна
  Коваленко Владимир Иванович
                       2001-12-09 Стрелец
                       2003-12-09 Стрелец
  Виноградов Павел Гаврилович
  Иванов Владимир Николаевич
                       1999-12-09 Стрелец
  Семашкевич Виктория Михайловна
                       1998-12-09 Стрелец
```

2. Самостоятельная работа

- 1. Вывести список студентов, отсортированный по количеству символов в ФИО.
- 2. Вывести список студентов, убрать лишние пробелы в ФИО.
- 3. Найти позиции «ов» в ФИО каждого студентов. Вывести ФИО и номер позиции.

- 4. Вывести ФИО и последние две буквы специализации для каждого студента.
- 5. Вывести список студентов, ФИО в формате Фамилия и Инициалы.
- 6. Вывести список специализаций в правильном и обратном виде. Убрать дубликаты.
- 7. Вывести свою фамилию в одной строке столько раз, сколько вам лет.
- 8. Вывести абсолютное значение функций $sin^2\left(\frac{\pi}{2}\right) cos\left(\frac{3\pi}{2}\right)$ с точностью два знака после десятичной запятой.
- 9. Вывести количество дней до конца семестра.
- 10. Вывести количество месяцев от вашего рождения.
- 12. Вывести список специализаций без повторений. Для каждой специализации вывести «длинный» или «короткий», в зависимости от количества символов.