Лабораторная работа 7.2

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 00:00:00.0000000 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 00:00:00.0000000 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: **CHAR** (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

Функции SQL производят действия с данными и возвращают результат. *Встроенные функции* 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
CONCAT(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.
COT(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
EXP(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	yy, yyyy
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:

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)

```
/****** Скрипт для команды SelectTopNRows из среды SSMS  ******/
SELECT TOP (1000) [FIO]
      ,[Data]
      ,[spez]
      ,[godpost]
  FROM [Ucheb_12_Ivanov].[dbo].[Student]
```

Результаты | Сообщения

	FIO	Data	spez	godpost
1	Александров Николай Николаевич	2001-01-01	физика	2018
2	Бельский Василий Владимир			
3	Белопольский Андрей Иванович	2004-10-21	Информатика	2022
4	Петров Иван Петрович	2004-07-17	физика	2022
5	Михайлов Николай Иванович	2005-12-09	математика	2023
6	Мироненко Светлана Владимировна	1998-12-09	Радиофизика	2016
7	Коваленко Владимир Иванович	2001-12-09	Информатика	2019
8	Виноградов Павел Гаврилович	2003-12-09	Химия	2021
9	Иванов Владимир Николаевич	1999-12-09	Радиофизика	2018
10	Семашкевич Виктория Михайловна	1998-12-09	Химия	2017

Нажмите, чтобы выделить столбец целиком

1) Вывести ФИО студентов и длину ФИО

```sql
use Ucheb_12_Ivanov

-- Вывести ФИО студентов и длину ФИО:
SELECT
        FIO
        ,LEN(FIO) AS Количество_символов
FROM
        Student
```

152 %

Результаты | Сообщения

	FIO	Количество_символов
1	Александров Николай Николаевич	30
2	Бельский Василий Владимирович	29
3	Белопольский Андрей Иванович	28
4	Петров Иван Петрович	20
5	Михайлов Николай Иванович	26
6	Мироненко Светлана Владимировна	31
7	Коваленко Владимир Иванович	27
8	Виноградов Павел Гаврилович	27
9	Иванов Владимир Николаевич	26
10	Семашкевич Виктория Михайловна	30

2) Вывести список студентов, убрать лишние пробелы в ФИО

```sql
--Вывести список студентов, убрать лишние пробелы в ФИО
SELECT
        TRIM(FIO) AS ФИО
        ,Data
        ,spez
        ,godpost
FROM
        Student
```

152 %

Результаты | Сообщения

	ФИО	Data	spez	godpost
1	Александров Николай Николаевич	2001-01-01	физика	2018
2	Бельский Василий Владимирович	2001-05-02	математика	2019
3	Белопольский Андрей Иванович	2004-10-21	Информатика	2022
4	Петров Иван Петрович	2004-07-17	физика	2022
5	Михайлов Николай Иванович	2005-12-09	математика	2023
6	Мироненко Светлана Владимировна	1998-12-09	Радиофизика	2016
7	Коваленко Владимир Иванович	2001-12-09	Информатика	2019
8	Виноградов Павел Гаврилович	2003-12-09	Химия	2021
9	Иванов Владимир Николаевич	1999-12-09	Радиофизика	2018
10	Семашкевич Виктория Михайловна	1998-12-09	Химия	2017

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

```
--Найти позиции буквы «о» в ФИО каждого студента. Вывести ФИО и позицию
SELECT
        FIO
    ,CHARINDEX('o',FIO) AS Позиция_о
    FROM
        Student
```

Результаты | Сообщения

FIO	Позиция_о
Александров Николай Николаевич	10
Бельский Василий Владимирович	26
Белопольский Андрей Иванович	4
Петров Иван Петрович	5
Михайлов Николай Иванович	7
Мироненко Светлана Владимировна	4
Коваленко Владимир Иванович	2
Виноградов Павел Гаврилович	4
Иванов Владимир Николаевич	5
Семашкевич Виктория Михайловна	16

4) Вывести ФИО и первые три буквы специализации каждого студента

```
--Вывести ФИО и первые три буквы специализации каждого студента
SELECT
        FIO
    ,LEFT(spez, 3) AS Спец_3
    FROM
        Student
```

Результаты | Сообщения

FIO	Спец_3
Александров Николай Николаевич	физ
Бельский Василий Владимирович	мат
Белопольский Андрей Иванович	Инф
Петров Иван Петрович	физ
Михайлов Николай Иванович	мат
Мироненко Светлана Владимировна	Рад
Коваленко Владимир Иванович	Инф
Виноградов Павел Гаврилович	Хим
Иванов Владимир Николаевич	Рад
Семашкевич Виктория Михайловна	Хим

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

```
31
32      --Вывести список студентов, заменить специализацию «математика»
33      --на «прикладная математика»
34  SELECT
35          FIO
36          ,Data
37          ,REPLACE(spez, 'математика', 'прикладная математика') AS Спец
38          ,godpost
39  FROM
40          Student
41
```

Результаты | Сообщения

FIO	Data	Спец	godpost
Александров Николай Николаевич	2001-01-01	физика	2018
Бельский Василий Владимирович	2001-05-02	прикладная математика	2019
Белопольский Андрей Иванович	2004-10-21	Информатика	2022
Петров Иван Петрович	2004-07-17	физика	2022
Михайлов Николай Иванович	2005-12-09	прикладная математика	2023
Мироненко Светлана Владимировна	1998-12-09	Радиофизика	2016
Коваленко Владимир Иванович	2001-12-09	Информатика	2019
Виноградов Павел Гаврилович	2003-12-09	Химия	2021
Иванов Владимир Николаевич	1999-12-09	Радиофизика	2018
Семашкевич Виктория Михайловна	1998-12-09	Химия	2017

7) Вывести список студентов, специализацию на верхнем регистре

```
41  --Вывести список студентов, специализацию на верхнем регистре
42  SELECT
43  FIO
44          ,Data
45          ,UPPER(spez) AS Спец
46          ,godpost
47  FROM
48          Student
49
```

Результаты | Сообщения

FIO	Data	Спец	godpost
Александров Николай Николаевич	2001-01-01	ФИЗИКА	2018
Бельский Василий Владимирович	2001-05-02	МАТЕМАТИКА	2019
Белопольский Андрей Иванович	2004-10-21	ИНФОРМАТИКА	2022
Петров Иван Петрович	2004-07-17	ФИЗИКА	2022
Михайлов Николай Иванович	2005-12-09	МАТЕМАТИКА	2023
Мироненко Светлана Владимировна	1998-12-09	РАДИОФИЗИКА	2016
Коваленко Владимир Иванович	2001-12-09	ИНФОРМАТИКА	2019
Виноградов Павел Гаврилович	2003-12-09	ХИМИЯ	2021
Иванов Владимир Николаевич	1999-12-09	РАДИОФИЗИКА	2018
Семашкевич Виктория Михайловна	1998-12-09	ХИМИЯ	2017

8) Вывести ФИО студентов в правильном и обратном виде

```
--Вывести ФИО академиков в правильном и обратном виде
SELECT
        FIO
        ,REVERSE(FIO) AS ФИО_Обр
FROM
        Student
        Название
```

Результаты | Сообщения

	FIO	ФИО_Обр
	Александров Николай Николаевич	чивеалокиН йалокиН ворднаскелА
2	Бельский Василий Владимирович	чиворимидалВ йилисаВ йиксьлеБ
3	Белопольский Андрей Иванович	чивонавИ йерднА йиксьлополеБ
4	Петров Иван Петрович	чивортеП навИ вортеП
5	Михайлов Николай Иванович	чивонавИ йалокиН волйахиМ
6	Мироненко Светлана Владимировна	анворимидалВ аналтевС окненориМ
7	Коваленко Владимир Иванович	чивонавИ римидалВ окнелавоК
8	Виноградов Павел Гаврилович	чиволирваГ леваП водаргониВ
9	Иванов Владимир Николаевич	чивеалокиН римидалВ вонавИ
10	Семашкевич Виктория Михайловна	анволйахиМ яироткиВ чивекшамеС

9) Вывести каждую специализацию 4 раза в одной строке. Убрать дубликаты

```
--Вывести каждую специализацию 4 раза в одной строке. Убрать дубликаты
SELECT DISTINCT
REPLICATE(spez, 4) AS Спец_4
FROM
Student
```

Результаты | Сообщения

Спец_4
ИнформатикаИнформатикаИнформатикаИнформатика
математикаматематикаматематикаматематика
РадиофизикаРадиофизикаРадиофизикаРадиофизика
физикафизикафизикафизика
ХимияХимияХимияХимия

10) Вывести абсолютное значение тригонометрических функций на точке π

```
--Вывести абсолютное значение тригонометрических функций на точке π
SELECT
        ABS(COS(PI()))   AS Косинус_Пи
       ,ABS(SIN(PI()))   AS Синус_Пи
       ,ABS(TAN(PI()))   AS Тангенс_Пи
       ,ABS(COT(PI()))   AS КоТангенс_Пи
```

Результаты | Сообщения

	Косинус_Пи	Синус_Пи	Тангенс_Пи	КоТангенс_Пи
1	1	1,22464679914735E-16	1,22464679914735E-16	8,16561967659769E+15

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

```
--Вывести число 132.456, округленное с точностью от 3 до -3
SELECT
         ROUND(123.456, 3)   AS Окр3
        ,ROUND(123.456, 2)   AS Окр2
        ,ROUND(123.456, 1)   AS Окр1
        ,ROUND(123.456, 0)   AS Окр0
        ,ROUND(123.456, -1)   AS Окр_1
        ,ROUND(123.456, -2)   AS Окр_2
        ,ROUND(123.456, -3)   AS Окр_3
```

Результаты | 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

```
--Вывести наименьшее целое число, которое больше или равно 123.456,
--и наибольшее целое число, которое меньше или равно 123.456
SELECT
         CEILING(123.456)  AS Больше
        ,FLOOR(123.456)    AS Меньше
```

Результаты | Сообщения

	Больше	Меньше
1	124	123

13) Вывести квадратный корень, квадрат и куб числа 25

```sql
--  Вывести квадратный корень, квадрат и куб числа 25
SELECT
         SQRT(25)      AS Корень
        ,SQUARE(25)    AS Квадрат
        ,POWER(25, 3) AS Куб
```

Результаты | Сообщения

Корень	Квадрат	Куб
5	625	15625

14) Вывести текущую дату и время

```sql
        ,POWER(25, 3) AS Куб
--Вывести текущую дату и время
SELECT
        GETDATE() AS Сейчас
```

Результаты | Сообщения

Сейчас
2023-04-16 22:38:28.863

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

```sql
SELECT
        GETDATE() AS Сейчас
--Вывести день, месяц, год,
--час, минуту, секунду,
--номер квартала, номер недели,
--день года, день недели для текущей даты и времени

SELECT
        DAY(GETDATE()) AS День
        ,MONTH(GETDATE()) AS Месяц
        ,YEAR(GETDATE()) AS Год
        ,DATEPART(HOUR, GETDATE()) AS Час
        ,DATEPART(MINUTE, GETDATE()) AS Минута
        ,DATEPART(SECOND, GETDATE()) AS Секунда
        ,DATEPART(QUARTER, GETDATE()) AS Квартал
        ,DATEPART(WEEK, GETDATE()) AS Неделя
        ,DATEPART(DAYOFYEAR, GETDATE()) AS День_года
        ,DATEPART(WEEKDAY, GETDATE()) AS День_недели
```

152 %

Результаты | Сообщения

	День	Месяц	Год	Час	Минута	Секунда	Квартал	Неделя	День_года	День_недели
1	16	4	2023	22	42	8	2	16	106	1

16) Вывести дату 100 дней назад от текущей

```sql

--Вывести дату 100 дней назад от текущей
SELECT
        DATEADD(DAY, -100, GETDATE()) AS День_100_Назад
```

Результаты | 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

```sql
--Иванов И.И. родился 8 июля 1895 года,
--скончался 12 апреля 1971 года. Вывести количество прожитых дней
SELECT
        DATEDIFF(DAY, '18950708', '19710412') AS Количество_прожитых_дней
```

Результаты | Сообщения

Количество_прожитых_дней
27671

18) Вывести ФИО и время года рождения каждого студента

```sql
--Вывести ФИО и время года рождения каждого студента
SELECT
        FIO
    , CASE MONTH(Data)
            WHEN 3 THEN 'Весна'
            WHEN 4 THEN 'Весна'
            WHEN 5 THEN 'Весна'
            WHEN 6 THEN 'Лето'
            WHEN 7 THEN 'Лето'
            WHEN 8 THEN 'Лето'
            WHEN 9 THEN 'Осень'
            WHEN 10 THEN 'Осень'
            WHEN 11 THEN 'Осень'
            ELSE 'Зима'
            END AS Времени_года
FROM Student
```

Результаты | Сообщения

FIO	Времени_года
Александров Николай Николаевич	Зима
Бельский Василий Владимирович	Весна
Белопольский Андрей Иванович	Осень
Петров Иван Петрович	Лето
Михайлов Николай Иванович	Зима
Мироненко Светлана Владимировна	Зима
Коваленко Владимир Иванович	Зима
Виноградов Павел Гаврилович	Зима
Иванов Владимир Николаевич	Зима
Семашкевич Виктория Михайловна	Зима

19) Вывести список студкетов. Для каждого студента, в зависимости от возраста, при поступлении «молодой» или «старше» в дополнительном столбце

```sql
--Вывести список студкетов.
--Для каждого студента, в зависимости от возраста,
--при поступлении «молодой» или «старше» в дополнительном столбце
SELECT
        FIO
        ,Data
        ,spez
        ,godpost
        ,IIF(godpost - Year(Data) <= 18, 'Молодой','старше')
AS Возраст_при_поступлении
FROM Student
```

Результаты | Сообщения

FIO	Data	spez	godpost	Возраст_при_поступлении
Александров Николай Николаевич	2001-01-01	физика	2018	Молодой
Бельский Василий Владимирович	2001-05-02	математика	2019	Молодой
Белопольский Андрей Иванович	2004-10-21	Информатика	2022	Молодой
Петров Иван Петрович	2004-07-17	физика	2022	Молодой
Михайлов Николай Иванович	2005-12-09	математика	2023	Молодой
Мироненко Светлана Владимировна	1998-12-09	Радиофизика	2016	Молодой
Коваленко Владимир Иванович	2001-12-09	Информатика	2019	Молодой
Виноградов Павел Гаврилович	2003-12-09	Химия	2021	Молодой
Иванов Владимир Николаевич	1999-12-09	Радиофизика	2018	старше
Семашкевич Виктория Михайловна	1998-12-09	Химия	2017	старше

20) Вывести ФИО, дату рождения и знак зодиака каждого студента

```sql
SELECT
        FIO
        , Data
        , CASE
            WHEN (MONTH(Data)=3  AND DAY(Data) >= 21)
OR (MONTH(Data)=4  AND DAY(Data) <= 20) THEN 'Овен'
            WHEN (MONTH(Data)=4  AND DAY(Data) >= 21)
OR (MONTH(Data)=5  AND DAY(Data) <= 21) THEN 'Телец'
            WHEN (MONTH(Data)=5  AND DAY(Data) >= 22)
OR (MONTH(Data)=6  AND DAY(Data) <= 21) THEN 'Близнецы'
            WHEN (MONTH(Data)=6  AND DAY(Data) >= 22)
OR (MONTH(Data)=7  AND DAY(Data) <= 22) THEN 'Рак'
            WHEN (MONTH(Data)=7  AND DAY(Data) >= 23)
OR (MONTH(Data)=8  AND DAY(Data) <= 21) THEN 'Лев'
            WHEN (MONTH(Data)=8  AND DAY(Data) >= 22)
OR (MONTH(Data)=9  AND DAY(Data) <= 23) THEN 'Дева'
            WHEN (MONTH(Data)=9  AND DAY(Data) >= 24)
OR (MONTH(Data)=10 AND DAY(Data) <= 23) THEN 'Весы'
            WHEN (MONTH(Data)=10 AND DAY(Data) >= 24)
OR (MONTH(Data)=11 AND DAY(Data) <= 22) THEN 'Скорпион'
            WHEN (MONTH(Data)=11 AND DAY(Data) >= 23)
OR (MONTH(Data)=12 AND DAY(Data) <= 22) THEN 'Стрелец'
            WHEN (MONTH(Data)=12 AND DAY(Data) >= 23)
OR (MONTH(Data)=1  AND DAY(Data) <= 20) THEN 'Козерог'
            WHEN (MONTH(Data)=1  AND DAY(Data) >= 21)
OR (MONTH(Data)=2  AND DAY(Data) <= 19) THEN 'Водолей'
            WHEN (MONTH(Data)=2  AND DAY(Data) >= 20)
OR (MONTH(Data)=3  AND DAY(Data) <= 20) THEN 'Рыбы'
        END AS Знак_зодиака
FROM Student
```

Результаты | Сообщения

FIO	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	Рак
Михайлов Николай Иванович	2005-12-09	Стрелец
Мироненко Светлана Владимировна	1998-12-09	Стрелец
Коваленко Владимир Иванович	2001-12-09	Стрелец
Виноградов Павел Гаврилович	2003-12-09	Стрелец
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.