Частное учреждение образования
«Колледж бизнеса и права»

Специальность 2-40 01 01: «Программное обеспечение информационных технологий»	Дисциплина: «Базы данных и системы управления базами данных»
Составлена на основании учебной программы, утвержденной директором Колледжа бизнеса и права	

Лабораторная работа № 10

Инструкционо-технологическая карта

Тема: Использование встроенных функций СУБД в запросах
Цель работы:
- изучить основные группы функций СУБД;
- научиться применять различные группы встроенных функций в запросах.

Время выполнения: 2 часа

Краткие теоретические сведения

Кроме связывания таблиц и отбора данных оператор SELECT может использоваться для вычислений. В этом случае он имеет синтаксис:

SELECT <Выражение>

где <выражение> - какое-то математическое выражение или функция. Выражение имеет стандартный вид (как в Visual Basic), оно может включать в себя встроенные функции сервера.

Математические функции

- **ABS**(numeric) – модуль числа.
- **ACOS/ASIN/ATAN** (Float) – арккосинус, арксинус, арктангенс в радианах.
- **COS/SIN/TAN/COT** (Float) – косинус, синус, тангенс, котангенс.
- **CEILING** (Numeric) – наименьшее целое, большее или равное параметру в скобках.
- **DEGREES** (Numeric) – преобразует радианы в градусы.
- **EXP**(Float) – экспонента, е в степени х.
- **FLOOR**(Numeric) – наименьшее целое меньшее или равное выражению numeric.
- **LOG**(Float) – натуральный логарифм ln.

- **LOG10**(Float) – десятичный логарифм log10.
- **PI** () – число пи.
- **POWER** (Numeric,y) – возводит выражение Numeric в степень y.
- **RADIANS** (Numeric) – преобразует градусы в радианы.
- **RAND** () – генерирует случайное число типа данных Float, расположенное между нулем и единицей.
- **ROUND** (Numeric, Длина) – округляет выражение Numeric до заданной длины (количество знаков после запятой). Можно выполнять округление числа как после десятичной точки (Длина>0), так и до неё (Длина<0).
- **SIGN** (Numeric) – выводит знак числа +/- или ноль.
- **SQUARE** (Float) – вычисляет квадрат числа Float.
- **SQRT** (Float) – вычисляет квадратный корень числа Float.

Пример. На рисунке 1 показаны несколько примеров, демонстрирующих работу математических функций

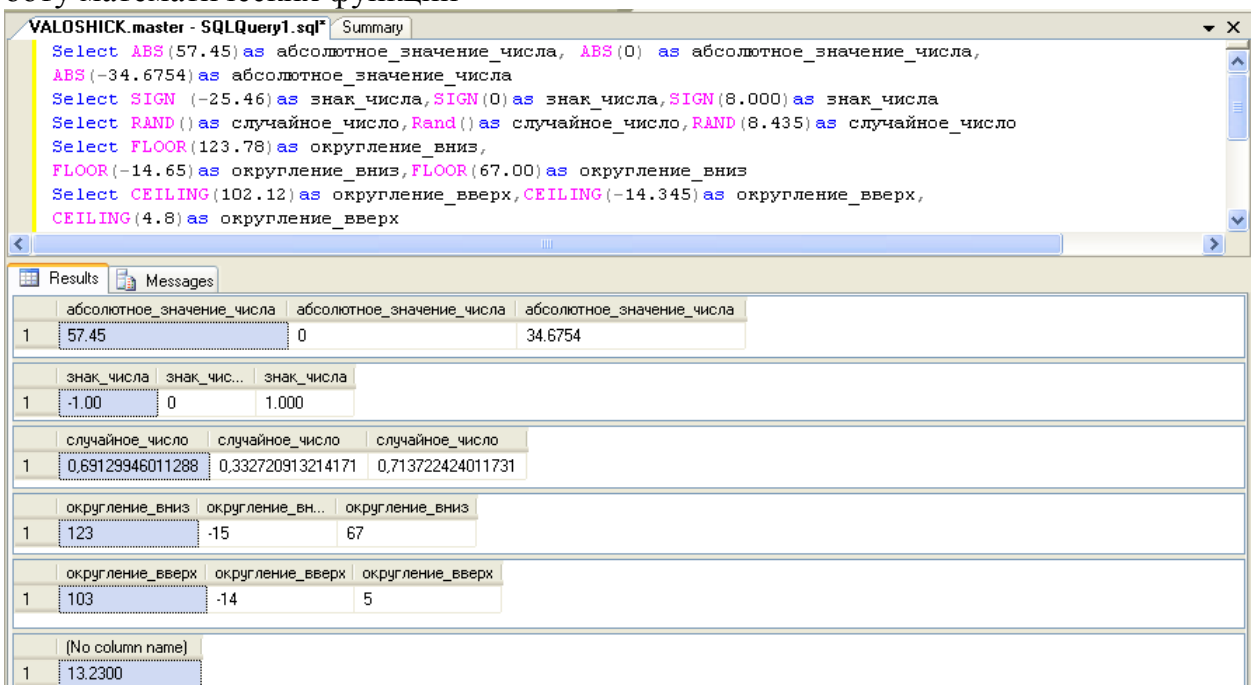

Рисунок 1 – Математические функции

Системные функции

Системные функции предназначены для получения информации о базе данных и ее содержимом. В SQL сервере существуют следующие системные функции:

- **COL_LENGTH**(таблица, поле) – выводит ширину поля.
- **DATALENGTH**(выражение) – выводит длину выражения.
- **GETAHSINULL**(имя БД) – выводит допустимо или недопустимо использовать в БД значение NULL.
- **IDENTINCR**(таблица) – выводит шаг увеличения поля счетчика в таблице.
- **IDENT_SEED**(таблица) – выводит начальное значение счетчиков в таблице.
- **ISDATE**(выражение) – выводит единицу, если выражение является датой и ноль, если не является.
- **ISNUMERIC**(выражение) – выводит единицу, если выражение является числовым и ноль, если не числовым.

- **NULIFF**(выражение1, выражение2) – выводит ноль если выражение1 равно выражению 2.

Агрегатные функции

Агрегатные функции – позволяют вычислять итоговые значения по полям таблицы.

- **COUNT**(поле) – выводит количество всех значений поля.
- **MAX**(поле) – выводит максимальное значение поля.
- **MIN**(поле) – выводит минимальное значение поля.
- **STDEV**(поле) – выводит средне квадратичное отклонение всех значений поля.
- **STDEVP**(поле) – выводит среднеквадратичное отклонение различных значений поля.
- **SUM**(поле) – суммирует все значений поля
- **TOP n [Percent]** – выводит n первых записей из таблица, либо n% записей из таблицы.
- **VAR**(поле) – выводит дисперсию всех значений поля.
- **VARP**(поле) – выводит дисперсию всех различных значений поля.

Функции работы с датами

Функции дат предназначены для работы с датами или времени. Существуют следующие функции дат:

- **DATEADD**(часть, число, date) – добавляет к дате date указанное значение дней, месяцев, часов и т.д. Первый аргумент определяет тип значения, которое необходимо добавить. Возможны следующие значения:
  - ✓ Year (yy или yyyy) – год;
  - ✓ Quarter (qq или q) – квартал;
  - ✓ Month(mm или m) – месяц;
  - ✓ Dayofyear(dy или y) – день (аналог Day);
  - ✓ Day (dd или d) – день;
  - ✓ Week(wk или ww) – неделя;
  - ✓ Hour (hh) – час;
  - ✓ Minute (mi или m) – минута;
  - ✓ Second(ss или s) – секунда;
  - ✓ Millisecond(ms) – миллисекунда.

Можно использовать как полные названия, так и сокращения.

Аргумент number содержит количество единиц, которое необходимо добавить к выбранному значению. Последний аргумент содержит значения типа datetime, date, smalldatetime.

Пример. На рисунке 2 показаны несколько примеров, демонстрирующих работу функции DATEADD

```
VALOSHICK.master - SQLQuery1.sql*   Summary
--Вычитание из указанной даты 08/13/78  2 года
Select DATEADD (year, -2,08/13/78 )as пример_1

--Прибавление к указанной дате 09/10/1978 10 дней и 35 минут
Select DATEADD (mi, 35,(DATEADD(day,10,09/10/1978)))as пример_2

--Прибавление к указанной   1/20/07 дате 5 дней
SELECT DATEADD(dd,5,11/20/07)  as прирер_3
```

Results | Messages

	пример_1
1	1898-01-01 00:00:00.000

	пример_2
1	1900-01-11 00:35:00.000

	прирер_3
1	1900-01-06 00:00:00.000

Рисунок 2 – Функция DATEADD

- **DATEDIFF**(часть, date1, date2) – возвращает разницу между указанными частями двух дат date1 и date2. Форматы для обозначения частей дат приведены выше, при описании функции DATEADD.

Пример. Подсчитать количество дней прошедших между двумя датами.

```
VALOSHICK.master - SQLQuery1.sql*   Summary
Select DATEDIFF(day,20-11-2007,25-11-2007) as разность
```

Results | Messages

	разность
1	5

Рисунок 3 – Функция DATEDIFF

- **DATENAME**(часть, date) – выделяет из даты date указанную часть и возвращает ее в символьном формате. Формат частей дат приведен в описании функции DATEADD.

Пример. На рисунке 4 показаны несколько примеров, демонстрирующих работу функции DATENAME

```
VALOSHICK.master - SQLQuery1.sql*   Summary
Select DATENAME(month,20-07-2007) as результат
```

Results | Messages

	результат
1	July

Рисунок 4 – Функция DATENAME

- **DATEPART**(часть, date) – выделяет из даты указанную часть и возвращает ее в числовом формате. Формат частей дат приведен в описании функции DATEADD.

Пример. Вывести количество месяцев в дате 20-07-2011

Select DATEPART(month,20-07-2011)

Результат: 7

- **DAY**(date) – выводит количество дней в заданной дате;
- **MONTH** (date) – выводит количество месяцев в заданной дате;
- **YEAR**(date) – выводит количество лет в заданной дате;

Пример. На рисунке 5 показаны несколько примеров, демонстрирующих работу функций DAY, MONTH, YEAR

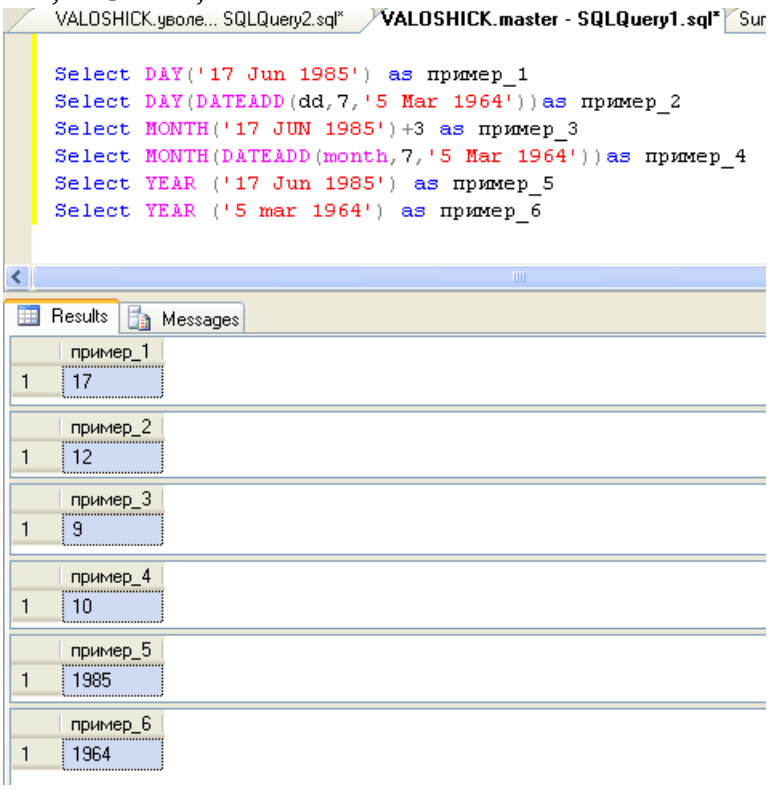

Рисунок 5 – Функции DAY, MONTH, YEAR

- **GETDATE()** – выводит текущее системное время.

Пример. На рисунке 6 показан пример, демонстрирующий работу функции GETDATE()

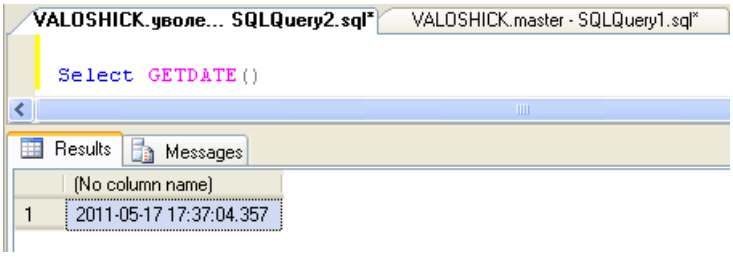

Рисунок 6 – ункция GETDATE()

Строковые функции

Строковые функции позволяют позволяют производить операции с одной или несколькими строками.

- **‘Строка1’+ ‘Строка2’** присоединяет Строку1 к Строке2.
- **ASCII**(Char) – возвращает ASCII код с самого левого символа выражения Char.

Пример. На рисунке 7 показан пример, демонстрирующий работу функции ASCII

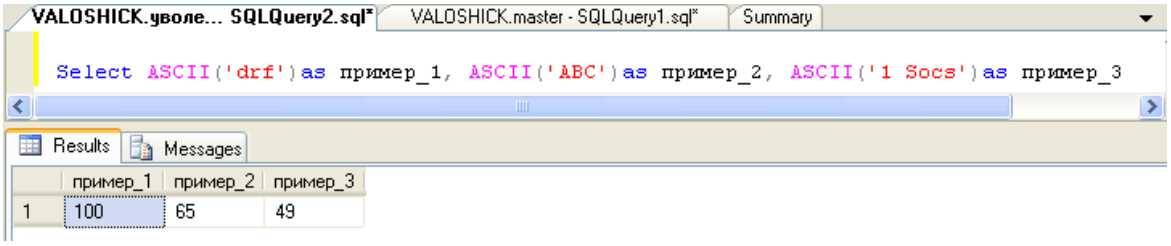

Рисунок 7 – Функция ASCII

- **CHAR**(Int) – возвращает символ, ASCII-кол которого, соответствует указанному значению. Функция CHAR выполняет действие, обратное функции ASCII.
- **CHARINDEX**(Образец, Выражение) – выполняет поиск подстроки Образец в строке Выражение. Функция возвращает порядковый номер символа, с которого начинается первое вхождение подстроки в строку. Для поиска второго и последующих вхождений подстроки дополнительно можно указать стартовую позицию, с которой будет начат поиск.

Пример. На рисунке 8 показаны примеры, демонстрирующие работу функции CHARINDEX

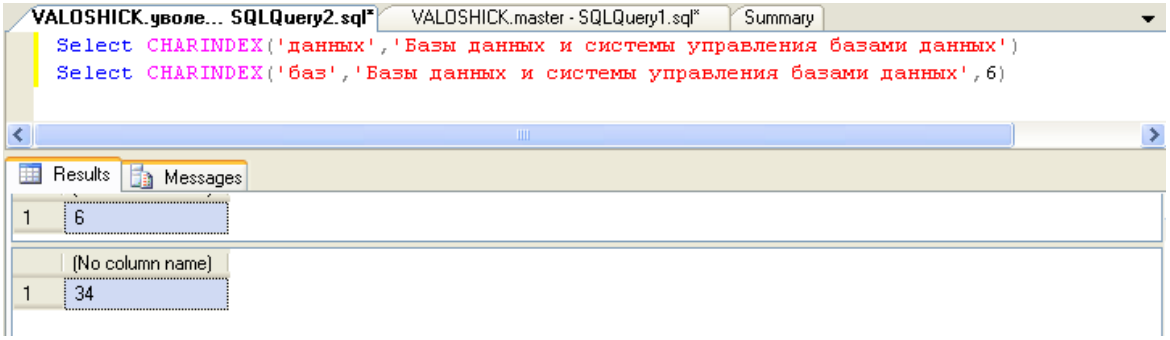

Рисунок 8 – Функция CHARINDEX

- **DIFFERENCE**(Выражение1, Выражение2) – возвращает число в диапазоне от 0 до 4, по которому можно судить о совпадении звучания двух строк: 0 – выражения абсолютно различны; 4 – выражения абсолютно идентичны. Оба выражения типа данных Char.

Пример. На рисунке 9 показаны примеры, демонстрирующие работу функции DIFFERENCE

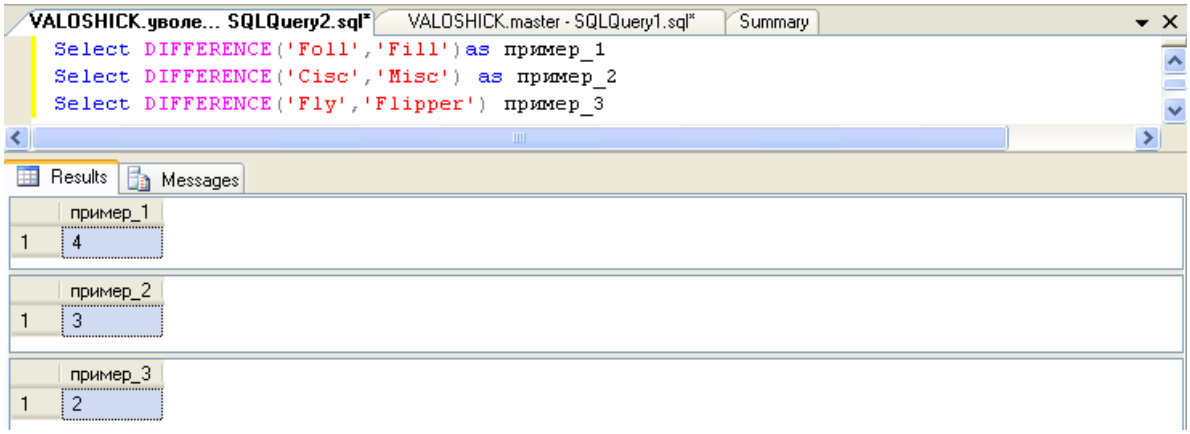

Рисунок 9 – Функция DIFFERENCE

- **LEFT**(Char, Int) – возвращает указанное количество символов строки, начиная с самого левого символа.

- **RIGHT**(Char, Int) – возвращает указанное количество символов с правого края строки.

Пример. На рисунке 10 показаны примеры, демонстрирующие работу функций RIGHT и LEFT

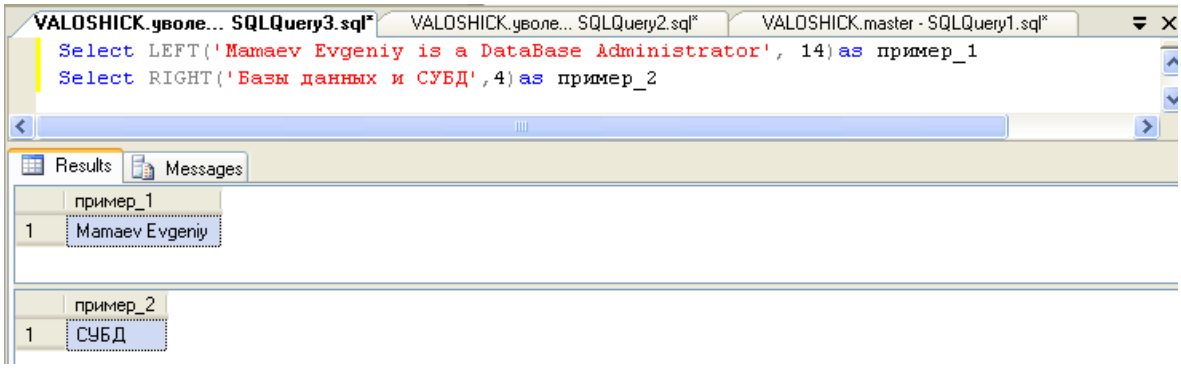

Рисунок 10 – Функции RIGHT и LEFT

- **LTRIM**(Char) – удаляет из строки Char пробелы слева.
- **RTRIM**(Char) – удаляет из строки Char пробелы справа.
- **NCHAR**(Int) – выводит выражение Int в формате Unicode.
- **REPLACE**(Строка1, Строка2, Строка3) – заменяет все вхождения введенной подстроки Строка2 на указанное значение Строка3.

Пример. На рисунке 11 показан пример, демонстрирующий работу функции REPLACE

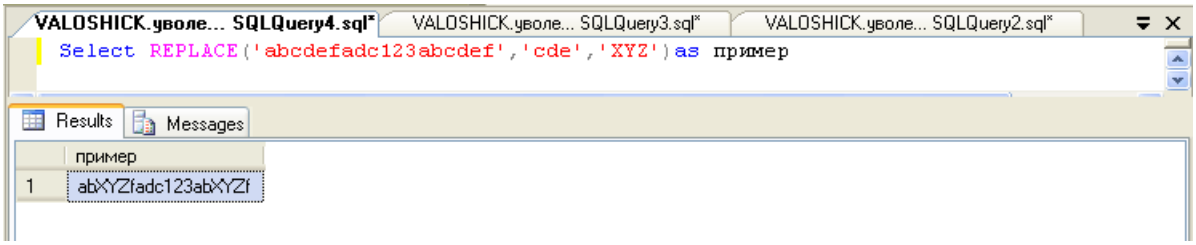

Рисунок 11 – Функции REPLACE

- **REPLICATE**(Char, Int) – выполняет повторение строки Char Int раз.
- **REVERSE**(Char) – возвращает строку, символы которой записаны в обратном порядке по отношению к исходной строке.

Пример. На рисунке 12 показаны примеры, демонстрирующие работу функций REPLICATE и REVERSE

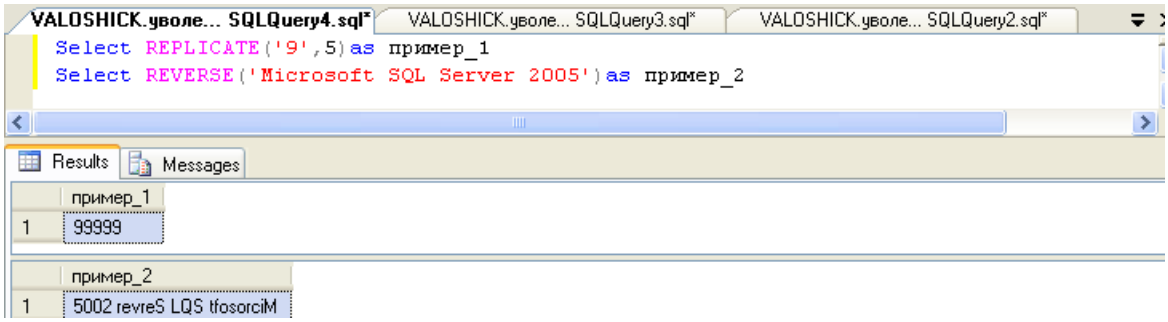

Рисунок 12 – Функции REPLICATE и REVERSE

- **SPACE**(Int) – возвращает указанное количество пробелов.
- **STR**(Float) – выполняет конвертирование значения числового типа в символьный формат.
- **STUFF**(Выражение1, Начало, Длина, Выражение2) – удаляет из Выражения1 начиная с позиции символа Начало количество символов равное параметру Длина, вместо них подставляет Выражение2.

- **SUBSTRING**(Выражение, Начало, Длина) – возвращает для троки Выражение подстроку указной длины, начиная с заданного символа.

Пример. На рисунке 13 показаны примеры, демонстрирующие работу функций STUFF и SUBSTRING

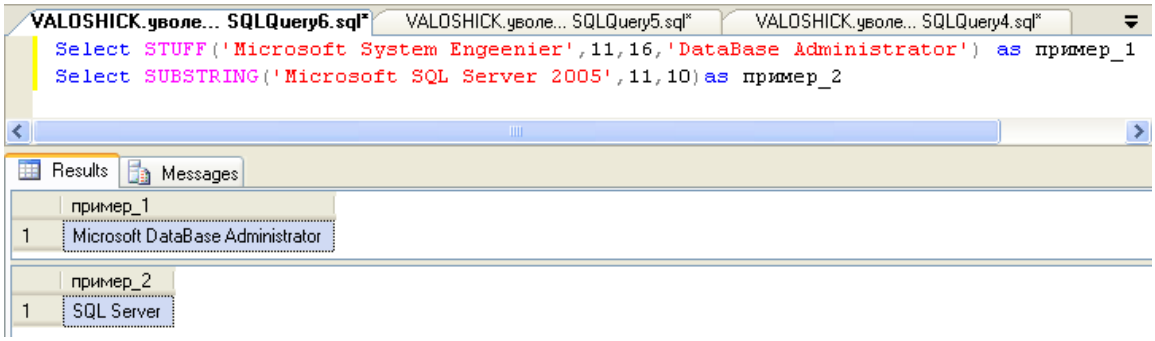

Рисунок 13 – Функции STUFF и SUBSTRING

- **UNICODE**(Char) – возвращает Unicode-код самого левого символа строки.
- **LOWER**(Char) – переводит все символы строки в нижний регистр.
- **UPPER**(Char) – переводит все символы строки в верхний регистр.

Пример. На рисунке 14 показаны примеры, демонстрирующие работу функций LOWER и UPPER

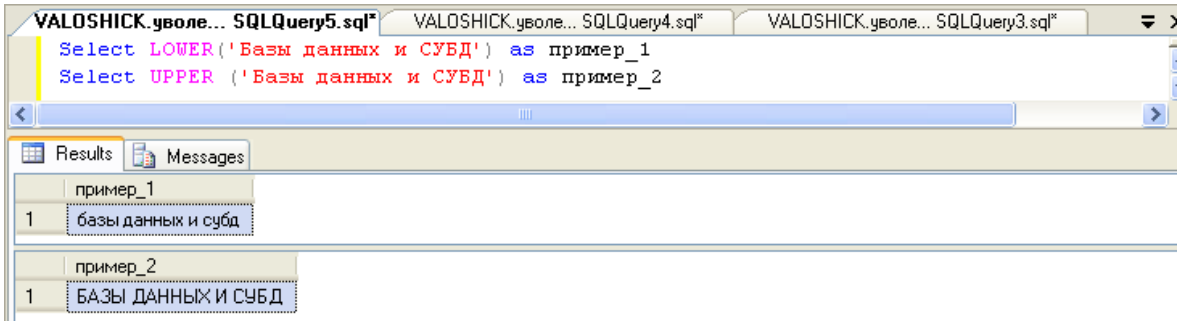

Рисунок 14 – Функции LOWER и UPPER

Логические функции

- **And** – возвращает True если аргументы имеют значение True.
- **Not** – возвращает True, если аргумент имеет значение False.
- **Or** – возвращает True, если любой из аргументов имеет значение True.

Статистичекие функции

С помощью статистических функций этоого типа можно получить информацию о состоянии системы и статистическую информацию о работе сервера.

- **@@CPU_ BSY** возвращает количество миллисекунд, которое затратил центральный процессор во время последнего запуска SQL Server
- **@@IO_BUSY** возвращает количество миллисекунд, которое система SQL Server ожидала выполнения операций ввода-вывода во время последнего запуска.
- **@@TIMETICKS** возвращает количество микросекунд в одном такте процессора
- **@@PACK_SENT** возвращает количество пакетов, которое было отправлено SQL Server клиентам со времени последнего запуска.
- **@@PACK_RECEIVED** возвращает количество пакетов, которое было принято SQL Server от клиентов со времени последнего запуска.

- **@@PACKET_ERRORS** возвращает количество неправильных пакетов, принятых SQL Server со времени последнего запуска.
- **@@IDLE** возвращает количество миллисекунд, прошедшее со времени последнего последнего запуска SQL Server.
- **@@TOTAL_READ** возвращает количество операций физического чтения диска, которое выполнила система SQL Server со времени последнего запуска.
- **@@TOTAL_WRITE** возвращает количество операций физической записи на диск, которое выполнила система SQL Server со времени последнего запуска.
- **@@TOTAL_ERRORS** возвращает количество ошибок дискогового ввода-вывода, произошедших со времени последнего запуска SQL Server.

Порядок выполнения работы

1. Изучить теоретический материла.
2. Выполнить все примеры и проверить результаты запросов.
3. Получить у преподавателя задание для индивидуальной работы.
4. Ответить на контрольные вопросы.

Контрольные вопросы

1. Перечислите и опишите математические функции: ABS, ACOS/ASIN/ATAN, COS/SIN/TAN/COT, EXP, LOG, PI, SQUARE, SQRT.
2. Перечислите и опишите основные агрегатные функции: COUNT, MAX, MIN, SUM.
3. Перечислите и опишите функции работы с датами: GETDATE(), DAY, MONTH, YEAR, DATEADD, DATEDIFF, DATENAME, DATEPART.
4. Перечислите и опишите основные логические функции: And, Not, Or.

Преподаватель С.В. Банцевич

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № от « »_____201_
Председатель ЦК Т.Г.Багласова