

БАЗЫ ДАННЫХ

Лекция 9 Основы T-SQL

Пакет

- Пакет - это группа операторов T-SQL, которая обрабатывается сервером СУБД вместе

```
----- пакет 1 -----  
use TEMPDB  
go  
----- пакет 2 -----  
create table #A ([Номер] int identity (1,1));  
go  
----- пакет 3 -----  
insert into #A default values;  
go 3          -- повторить 3 раза  
----- пакет 4 -----  
select * from #A;  
go  
----- пакет 5 -----  
drop table #A;
```

Номер
1
2
3

DECLARE

- DECLARE - объявление используемых переменных
- Для каждой переменной указывается имя и тип

DECLARE

```
declare @x int = 1,  
        @y decimal(7,2), -- не инициализирована  
        @z float(4) = 2.1E-3;  
  
declare @a char(10),      -- не инициализирована  
        @b varchar(50) = 'Hello World!',  
        @c nvarchar(50) = N'Привет мир!';  
  
declare @d date = '20140107',  
        @e datetime = getdate(),  
        @f time;          -- не инициализирована  
  
declare @g uniqueidentifier = newid();  
  
declare @h table(  
                num int identity(1,1),  
                fil varchar(30) default 'XXX'  
                );  
  
insert @h default values; -- добавление одной строки  
select @x x, @y y, @z z;  
select @a a, @b b, @c c;  
select @d d, @e e, @f f;  
select @g g;  
select * from @h;
```

x	y	z
1	NULL	0.0021

a	b	c
NULL	Hello World!	Привет мир!

d	e	f
2014-01-07	2014-03-04 02:39:34.530	NULL

g
A341C7D3-B1B3-447F-B52C-E3EC6C2C9783

num	fil
1	XXX

Область видимости переменной

- Областью видимости переменной являются все инструкции между ее объявлением и концом пакета или хранимой процедуры, где она объявлена.

Операторы присваивания

- инициализировать в DECLARE
- присвоить значение SET
- присвоить значение SELECT

Операторы присваивания

```
declare    @y decimal(7,2) = 0.0,  
           @a char(10) ;  
set @y = 2.55;  
set @a = 'Hello';  
select @y y, @a a;  
go
```

y	a
2.55	Hello

```
declare    @y decimal(7,2) = 0.0,  
           @a char(10) ;  
select @y = 2.55, @a = 'Hello';  
select @y y, @a a;  
go
```

y	a
2.55	Hello

Оператор PRINT

```
declare
    @StudentsCount    int  = (select count(*) from STUDENT),
    @AuditoriumCount  int  = (select count(*) from AUDITORIUM),
    @TeacherCount     int  = (select count(*) from TEACHER);

select AUDITORIUM_TYPE [Тип аудитории],
       count(*)         [Количество аудиторий]
from AUDITORIUM group by AUDITORIUM_TYPE;
select PULPIT           [Кафедра],
       count(*)         [Количество дисциплин]
from SUBJECT group by PULPIT;

print 'Количество студентов      : ' + cast (@StudentsCount as varchar(10));
print 'Количество аудиторий      : ' + cast (@AuditoriumCount as varchar(10));
print 'Количество преподавателей : ' + cast (@TeacherCount as varchar(10));
go
```

Тип аудитории	Количество аудиторий
XXXX	1
ЛБ-К	5
ПК	13

Кафедра	Количество дисциплин
ИСиТ	14
ЛЗиДВ	1
ЛМиЛЗ	1

```
Количество студентов      : 178
Количество аудиторий      : 24
Количество преподавателей : 24
```


CAST и CONVERT

- Функции преобразования типов:
 - CAST(<выражение> AS <тип данных>)
 - CONVERT (<тип данных>, (<выражение> [, style])

CAST

Исходный тип	Целевой тип	Поведение
numeric	numeric	Округление
numeric	int	Усечение
numeric	money	Округление
money	int	Округление
money	numeric	Округление
float	int	Усечение
float	numeric	Округление
float	datetime	Округление
datetime	int	Округление

CAST

K:																											
От:	binary	varbinary	char	varchar	nchar	nvarchar	datetime	smalldatetime	decimal	numeric	float	real	bigint	int(INT4)	smallint(INT2)	tinyint(INT1)	money	smallmoney	bit	timestamp	uniqueidentifier	image	ntext	text	sql_variant	xml	
binary																											
varbinary																											
char																											
varchar																											
nchar																											
nvarchar																											
datetime																											
smalldatetime																											
decimal																											
numeric																											
float																											
real																											
bigint																											
int(INT4)																											
smallint(INT2)																											
tinyint(INT1)																											
money																											
smallmoney																											
bit																											
timestamp																											
uniqueidentifier																											
image																											
ntext																											
text																											
sql_variant																											
xml																											

- Явное преобразование
- Неявное преобразование
- Преобразование недопустимо
- ★ Требуется явное CAST, чтобы предотвратить потерю при неявном преобразовании
- Неявные преобразования между типами XML-данных

CAST

```
-- numeric(p,s)    1<p<38, 0<=s<=p
```

```
| select 14/24 '-i-',  
  CAST (14/24 as numeric(5,3)) '-ii-',  
- CAST (14      as numeric(5,3))/CAST (24 as numeric(5,3)) '-iii-'
```

÷	-ii-	-iii-
0	0.000	0.583333333

CAST

```
declare @y1 numeric (8,3) = (select CAST(SUM(AUDITORIUM_CAPACITY) as numeric(8,3)) from AUDITORIUM),
        @y2 numeric (8,3), @y3 numeric (8,3), @y4 numeric (8,3)
if @y1 > 200
begin
    print 'общее количество мест больше 200'
    select @y2 = CAST((select COUNT(*)
                        from AUDITORIUM) as numeric(8,3)),
           @y3 = (select CAST(AVG(AUDITORIUM_CAPACITY) as numeric(8,3))
                  from AUDITORIUM),
           @y4 = (select CAST(COUNT(*) as numeric(8,3))
                  from AUDITORIUM where AUDITORIUM_CAPACITY > @y3)

    select @y1 '@y1 ', @y2 '@y2', @y3 '@y3', @y4 '@y4', @y4/@y2*100 '%'
end
else if @y1 > 100
    print 'общее количество мест от 100 до 200'
else if @y1 > 50
    print 'общее количество мест от 50 до 100'
else
    print 'общее количество мест меньше 50'
```

@y1	@y2	@y3	@y4	%
1564.000	24.000	71.000	0.000	0.00000000000000

CAST

```
declare @y1 numeric (8,3) = (select CAST(SUM(AUDITORIUM_CAPACITY) as numeric(8,3)) from AUDITORIUM),
        @y2 numeric (8,3), @y3 numeric (8,3), @y4 numeric (8,3)

if @y1 > 200
begin
    print 'общее количество мест больше 200'
    select @y2 = CAST((select COUNT(*)
                        from AUDITORIUM) as numeric(8,3)),
           @y3 = (select CAST(AVG(AUDITORIUM_CAPACITY) as numeric(8,3))
              from AUDITORIUM)

    set @y4 = (select CAST(COUNT(*) as numeric(8,3))
              from AUDITORIUM where AUDITORIUM_CAPACITY > @y3)

    select @y1 '@y1 ', @y2 '@y2', @y3 '@y3', @y4 '@y4', @y4/@y2*100 '%'
end
else if @y1 > 100
    print 'общее количество мест от 100 до 200'
else if @y1 > 50
    print 'общее количество мест от 50 до 100'
else
    print 'общее количество мест меньше 50'
```

@y1	@y2	@y3	@y4	%
1564.000	24.000	71.000	14.000	58.333333333300

Арифметические операции

```
declare @x int = 0, @x1 int = 1,  
        @y int = 1, @y1 int = 2,  
        @z numeric(7,2) = 2.0, @z1 numeric(5,1) = 33.0,  
        @w float(8) = 3.0, @w1 float(4) = 5.1E3;
```

```
-----  
set @x+=1;   print '@x-=1 = ' + cast(@y as varchar(10));  
set @y-=1;   print '@y-=1 = ' + cast(@y as varchar(10));  
set @z*=3.0; print '@z*=3.0 = ' + cast(@z as varchar(10));  
set @w/=2.0; print '@w/=2.0 = ' + cast(@w as varchar(10));  
set @z%=4;   print '@z/=4   = ' + cast(@z as varchar(10));
```

@x-=1 = 1

@y-=1 = 0

@z*=3.0 = 6.00

@w/=2.0 = 1.5

@z/=4 = 2.00

Арифметические операции

```
set @z=1233.2 + 15.01;      print '@z=1233.2 + 15.01 = ' + cast(@z as varchar(10));
set @w=12.33E3 - 15.01E4;   print '@w=12.33E3 - 15.01E2 = ' + cast(@w as varchar(10));
set @x=1233 * 15;           print '@x=1233 * 15 = ' + cast(@x as varchar(10));
set @y=1233.33 / 15;        print '@y=1233.33 / 15 = ' + cast(@y as varchar(10));
set @x=-100%11;             print '@x=100%10 = ' + cast(@x as varchar(10));
```

@z=1233.2 + 15.01 = 1248.21

@w=12.33E3 - 15.01E2 = -137770

@x=1233 * 15 = 18495

@y=1233.33 / 15 = 82

@x=100%10 = -1

Арифметические операции

```
select  @x1 = @x+@y, @y1 =@x-@y;  
print '@x1 = @x+@y = ' + cast(@x1 as varchar(10)) + ' ; @y1 =@x-@y = ' + cast(@y1 as varchar(10));  
select  @z1 = @x*@z1, @w1 =@w1/@y;  
print '@z1 = @x*@z1 = ' + cast(@z1 as varchar(10)) + ' ; @w1 =@w1/@y = ' + cast(@w1 as varchar(10))  
select  @w1 = @z%@x;   print '@w1 = @z%@x = ' + cast(@w1 as varchar(10));
```

@x1 = @x+@y = 81; @y1 =@x-@y = -83

@z1 = @x*@z1 = -33.0; @w1 =@w1/@y = 62.1951

@w1 = @z%@x = 0.21

Встроенные функции

```
print 'Округление           : '+ cast(round(12345.12345, 2) as varchar(12));
print 'Нижнее целое         : '+ cast(floor(24.5) as varchar(12));
print 'Верхнее целое        : '+ cast(ceiling(24.5) as varchar(12));
print 'Возведение в степень: '+ cast(power(12.0, 2) as varchar(12));
print 'Логарифм по exp      : '+ cast(log(144.0) as varchar(12));
print 'Корень квадратный    : '+ cast(sqrt(144.0) as varchar(12));
print 'Экспонента           : '+ cast(exp(4.96981) as varchar(12));
print 'Абсолютное значение : '+ cast(abs(-5) as varchar(12));
print 'Радиианы             : '+ cast(radians(180.0) as varchar(20));
print 'Число пи             : '+ cast(pi() as varchar(12));
print 'Градусы              : '+ cast(degrees(pi()) as varchar(20));
print 'Синус                : '+ cast(sin(pi()) as varchar(12));
print 'Косинус              : '+ cast(cos(pi()) as varchar(12));
```

Встроенные функции

Округление	: 12345.12000
Нижнее целое	: 24
Верхнее целое	: 25
Возведение в степень	: 144.0
Логарифм по eхр	: 4.96981
Корень квадратный	: 12
Экспонента	: 144
Абсолютное значение	: 5
Рadiany	: 3.141592653589793100
Число пи	: 3.14159
Градусы	: 180
Синус	: 1.22465e-016

Конкатенация

```
declare @y1 numeric (8,3) = (select CAST(SUM(AUDITORIUM_CAPACITY) as numeric(8,3)) from AUDITORIUM),
        @y2 numeric (8,3), @y3 numeric (8,3), @y4 numeric (8,3)

if @y1 > 200
begin
    print 'общее количество мест больше 200'
    select @y2 = CAST((select COUNT(*)
                        from AUDITORIUM) as numeric(8,3)),
           @y3 = (select CAST(AVG(AUDITORIUM_CAPACITY) as numeric(8,3))
                  from AUDITORIUM)

    set @y4 = (select CAST(COUNT(*) as numeric(8,3))
              from AUDITORIUM where AUDITORIUM_CAPACITY > @y3)

    print ' @y1 = ' + CAST(@y1 as varchar(20)) +
          ' @y2 = ' + CAST(@y2 as varchar(20)) +
          ' @y3 = ' + CAST(@y3 as varchar(20)) +
          ' @y4 = ' + CAST(@y4 as varchar(20)) +
          ' % = ' + CAST(@y4/@y2*100 as varchar(20))
end
else if @y1 > 100
    print 'общее количество мест от 100 до 200'
else if @y1 > 50
    print 'общее количество мест от 50 до 100'
else
    print 'общее количество мест меньше 50'
```

@y1 = 1564.000 @y2 = 24.000 @y3 = 71.000 @y4 = 14.000 % = 58.333333333300

Строковые функции

```
print 'Подстрока          : '+' substring('1234567890', 3,2);
print 'Удалить пробелы слева : '+' 'X'+ ltrim('      67890');
print 'Удалить пробелы справа : '+' rtrim('12345      ') +'X';
print 'Нижний регистр       : '+' lower ('ВЕРХНИЙ РЕГИСТР');
print 'Верхний регистр      : '+' upper ('нижний регистр');
print 'Заменить            : '+' replace('1234512345', '5', 'X');
print 'Строка пробелов     : '+' 'X'+ space(5) +'X';
print 'Повторить строку     : '+' replicate('12', 5);
print 'Найти по шаблону     : '+' cast (patindex ('%Y_Y%', '123456YxY7890') as varchar(5));
```

```
Подстрока          : 34
Удалить пробелы слева : X67890
Удалить пробелы справа : 12345X
Нижний регистр       : верхний регистр
Верхний регистр      : НИЖНИЙ РЕГИСТР
Заменить            : 1234X1234X
Строка пробелов     : X      X
Повторить строку     : 1212121212
Найти по шаблону     : 7
```

Функции даты и времени

```
declare @dt datetime = getdate(), -- текущая дата
        @ed datetime = dateadd(d, 200, getdate()); -- текущая дата +200 дней

print 'Текущая дата' : '+ convert(varchar(12), @dt, 103);
print 'Текущая дата+200 дней' : '+ convert(varchar(12), @ed, 103);
print 'День' : '+ convert(varchar(12), day(@dt));
print 'День недели (1-вс)' : '+ convert(varchar(12), datepart(dw, @dt));
print 'Неделя' : '+ convert(varchar(12), datepart(wk, @dt));
print 'Месяц' : '+ convert(varchar(12), month(@dt));
print 'Квартал' : '+ convert(varchar(12), datepart(q, @dt));
print 'Год' : '+ convert(varchar(12), year(@dt));
print '+1 день' : '+ convert(varchar(12), dateadd(d, 1, @dt), 103);
print '+1 неделя' : '+ convert(varchar(12), dateadd(wk, 1, @dt), 103);
print '+1 месяц' : '+ convert(varchar(12), dateadd(m, 1, @dt), 103);
print '+1 квартал' : '+ convert(varchar(12), dateadd(q, 1, @dt), 103);
print '+1 год' : '+ convert(varchar(12), dateadd(y, 1, @dt), 103);
print 'Разница в днях' : '+ convert(varchar(12), datediff(d, @dt, @ed));
print 'Разница в неделях' : '+ convert(varchar(12), datediff(wk, @dt, @ed));
print 'Разница в месяцах' : '+ convert(varchar(12), datediff(m, @dt, @ed));
print 'Разница в годах' : '+ convert(varchar(12), datediff(yy, @dt, @ed));
```

Функции даты и времени

- DATEADD (datepart, number, date)
 - datepart:
 - year - yy, yyyy; quarter - qq, q; month - mm, m;
 - dayofyear - dy, y; day - dd, d; week - wk, ww;
 - weekday - dw, w; hour – hh; minute - mi, n;
 - second - ss, s
-
- DATEPART (datepart, date)
- DATEDIFF (datepart , startdate , enddate)

Функции даты и времени

Текущая дата	: 05/03/2014
Текущая дата+200 дней	: 21/09/2014
День	: 5
День недели	: 4
Неделя	: 10
Месяц	: 3
Квартал	: 1
Год	: 2014
+1 день	: 06/03/2014
+1 неделя	: 12/03/2014
+1 месяц	: 05/04/2014
+1 квартал	: 05/06/2014
+1 год	: 06/03/2014
Разница в днях	: 200
Разница в неделях	: 29
Разница в месяцах	: 6
Разница в годах	: 0

Функции даты и времени

```
declare @t time(7) = sysdatetime(), -- текущее время
        @et time(7) = dateadd(mi, 12*60, getdate()),
        @dt datetime2 = sysdatetime(); -- текущая дата и время

print 'Текущее время' : '+' + convert(varchar(12), @t);
print 'Текущие дата и время' : '+' + convert(varchar(20), @dt, 120);
print 'Текущие время + 12час' : '+' + convert(varchar(20), @et);
print 'Часы' : '+' + convert(varchar(20), datepart(hh, @t));
print 'Минуты' : '+' + convert(varchar(20), datepart(mi, @t));
print 'Секунды' : '+' + convert(varchar(20), datepart(ss, @t));
print 'Миллисекунды' : '+' + convert(varchar(20), datepart(ms, @t));
print 'Текущее время + 1час' : '+' + convert(varchar(20), dateadd(hh, 1, @t));
print 'Текущее время + 1мин' : '+' + convert(varchar(20), dateadd(mi, 1, @t));
print 'Текущее время + 1сек' : '+' + convert(varchar(20), dateadd(ss, 1, @t));
print 'Разница в днях' : '+' + convert(varchar(20), datediff(d, @t, @et));
print 'Разница в час' : '+' + convert(varchar(20), datediff(hh, @t, @et));
print 'Разница в мс' : '+' + convert(varchar(20), datediff(ms, @t, @et));
```

```
Текущее время : 11:11:38.248
Текущие дата и время : 2014-03-05 11:11:38
Текущие время + 12час : 23:11:38.2470000
Часы : 11
Минуты : 11
Секунды : 38
Миллисекунды : 248
Текущее время + 1час : 12:11:38.2489297
Текущее время + 1мин : 11:12:38.2489297
Текущее время + 1сек : 11:11:39.2489297
Разница в днях : 0
Разница в час : 12
Разница в мс : 43199999
```

Системные функции

```
select @@ROWCOUNT      'кол. обработанных записей',
       @@ERROR           'код ошибки',
       @@SERVERNAME      'имя сервера',
       @@IDENTITY        'последнее значение IDENTITY',
       @@SPID            'сист. идентификатор',
       @@VERSION          'версия SQL Server'

select @@TRANCOUNT       'уровень вложенности транзакции',
       @@FETCH_STATUS    'статус команды выборки',
       @@NESTLEVEL       'уровень вложенности'

go
```

кол. обработанных записей	код ошибки	имя сервера	последнее зн...	сист. идент...	версия SQL Server
1	0	SMW-IIS-SQL	NULL	52	Microsoft SQL Server 2008 R2 (RTM) - 10.50.1600...

уровень вложенности транзакции	статус команды выборки	уровень вложенности
0	0	0

Оператор RETURN

```
print 'Метка 1';  
return;          -- выход из пакета  
print 'Метка 2'; -- не выполняется  
go  
print 'Метка 3';
```

```

print 'пакет1: начало-----';
declare @x0 int = 0;
print 'пакет1:@x0 = ' + cast(@x0 as char(1));
begin
    declare @x1 int = 1;
    print 'блок1: начало -----';
    print 'блок1:@x0 = ' + cast(@x0 as char(1));
    print 'блок1:@x1 = ' + cast(@x1 as char(1));
    begin
        declare @x2 int = 2;
        print '    блок2: начало -----';
        print '    блок2:@x0 = ' + cast(@x0 as char(1));
        print '    блок2:@x1 = ' + cast(@x1 as char(1));
        print '    блок2:@x2 = ' + cast(@x2 as char(1));
        print '    блок2: конец -----';
    end;
    begin
        print '    блок3: начало -----';
        print '    блок3:@x0 = ' + cast(@x0 as char(1));
        print '    блок3:@x1 = ' + cast(@x1 as char(1));
        print '    блок3:@x2 = ' + cast(@x2 as char(1));
        return; -- выход из пакета
        print '    блок3: конец -----';
        print ''
    end;
    print 'блок1: конец -----';
end;
print 'пакет1: конец-----';
go
print 'пакет2: начало-----';
print 'пакет2: конец-----';

```

BEGIN END

```

пакет1: начало-----
пакет1:@x0 = 0
блок1: начало -----
блок1:@x0 = 0
блок1:@x1 = 1
    блок2: начало -----
    блок2:@x0 = 0
    блок2:@x1 = 1
    блок2:@x2 = 2
    блок2: конец -----
    блок3: начало -----
    блок3:@x0 = 0
    блок3:@x1 = 1
    блок3:@x2 = 2
пакет2: начало-----
пакет2: конец-----

```

BEGIN END

```
declare @x1 int = 0

set @x1 = (select SUM(AUDITORIUM_CAPACITY) from AUDITORIUM)

if @x1 > 200
    begin
        print 'общее количество мест больше 200'
        select CAST (COUNT(*) as numeric(5,3)) /
            CAST( (select COUNT(*) from AUDITORIUM) as numeric(5,3)) *
                100 as '%'
            from AUDITORIUM
            where AUDITORIUM_CAPACITY > (select AVG(AUDITORIUM_CAPACITY) from AUDITORIUM);
    end
else if @x1 > 100
    print 'общее количество мест от 100 до 200'
else if @x1 > 50
    print 'общее количество мест от 50 до 100'
else
    print 'общее количество мест меньше 50'
```

%

58.333333300

IF-ELSE

```
if exists(select * from PULPIT)
    print 'в таблице PULPIT есть строки'
else
    print 'в таблице PULPIT нет строк'
go
```

в таблице PULPIT есть строки

```
if (select SUM(AUDITORIUM_CAPACITY) from AUDITORIUM) > 200
    print 'общее количество мест больше 200'
else
    print 'общее количество мест меньше 200'
go
```

общее количество мест больше 200

IF-ELSE

```
declare @x1 int = 0
```

```
set @x1 = (select SUM(AUDITORIUM_CAPACITY) from AUDITORIUM)
```

```
if @x1 > 200
```

```
    print 'общее количество мест больше 200'
```

```
else if @x1 > 100
```

```
    print 'общее количество мест от 100 до 200'
```

```
else if @x1 > 50
```

```
    print 'общее количество мест от 50 до 100'
```

```
else
```

```
    print 'общее количество мест меньше 50'
```

```
go
```

CASE

```
declare @z1 numeric (8,3) = (select CAST(SUM(AUDITORIUM_CAPACITY) as numeric(8,3)) from AUDITO
print 'общее количество мест ' +
(
  case
    when @z1 > 200 then 'больше 200'
    when @z1 > 100 then 'от 100 до 200'
    else 'меньше 50'
  end
)
```

общее количество мест больше 200

CASE

```
select case (select COUNT(*) from TEACHER where TEACHER_NAME LIKE '%Владимир%')  
  when 0 then 'нет'  
  when 1 then 'один'  
  when 2 then 'два'  
  else 'тьма'  
end 'Владимир'
```

Владимир

тьма

```
select * from TEACHER where TEACHER_NAME LIKE '%Владимир%'
```

TEACHER	TEACHER_NAME	PULPIT
БЗБРДВ	Безбородов Владимир Степанович	ОХ
МШКВСК	Машковский Владимир Петрович	ЛУ
СМЛВ	Смелов Владимир Владиславович	ИСИТ

WHILE

```
select SUM(AUDITORIUM_CAPACITY) from AUDITORIUM
```

(Отсутствует имя столбца)

1564

```
while (select SUM(AUDITORIUM_CAPACITY) from AUDITORIUM) < 2000
```

```
begin
```

```
    update AUDITORIUM set AUDITORIUM_CAPACITY = AUDITORIUM_CAPACITY*1.05
```

```
end
```

```
select SUM(AUDITORIUM_CAPACITY) from AUDITORIUM
```

(Отсутствует имя столбца)

2015

WHILE

```
declare @i int = 0;
while @i < 5          -- логическое выражение
    set @i += 1;      -- тело цикла
print '@i = ' + cast(@i as varchar(3));
```

@i = 5

```
declare @i int = 0;
while @i < 5          -- логическое выражение
begin                -- тело цикла
    set @i += 1;
    print '@i = ' + cast(@i as varchar(3));
end;
```

@i = 1
@i = 2
@i = 3
@i = 4
@i = 5

```

set nocount on;  -- не выводить: row(s) affected
declare @ac int  = (select sum(AUDITORIUM_CAPACITY)
                    from AUDITORIUM
                    where AUDITORIUM_TYPE in ('ЛК', 'ЛБ-К'));
print 'Общая вместимость аудиторий (ЛК,ЛБ-К) : ' + cast(@ac as varchar(6));
while (@ac < 1500)
begin
    update AUDITORIUM set AUDITORIUM_CAPACITY *= 1.1  where AUDITORIUM_TYPE = 'ЛК';
    update AUDITORIUM set AUDITORIUM_CAPACITY *= 1.05 where AUDITORIUM_TYPE = 'ЛБ-К';
    set @ac  = (select sum(AUDITORIUM_CAPACITY)
                from AUDITORIUM
                where AUDITORIUM_TYPE in ('ЛК', 'ЛБ-К'));
    print space(38) + ': ' + cast(@ac as varchar(6));
end;
print 'Общая вместимость аудиторий (ЛК,ЛБ-К) : ' + cast(@ac as varchar(6));

```

```

Общая вместимость аудиторий (ЛК,ЛБ-К) : 1245
                                     : 1355
                                     : 1465
                                     : 1586
Общая вместимость аудиторий (ЛК,ЛБ-К) : 1586

```

WHILE

```
declare @i int = 0, @x int;
while (@i < 5)
begin
    set @x = @i;
    set @i+=1;
    print 'A' + cast(@x as varchar(2));
    if @x = 2 continue;      -- В НАЧАЛО
    if @x = 3 break;         -- ВЫЙТИ
    print 'B' + cast(@x as varchar(2));
end;
```

A0
B0
A1
B1
A2
A3

TRY/ CATCH

```
declare @x int = 3, @y int = 0, @z int;
begin try
    print 'begin try:A'
    set @z = @x/@y;           -- деление на 0
    print 'begin try:B'      -- если ошибка - не выполняется
end try
begin catch                  -- обработка ошибки
    print 'begin catch:C'
end catch
```

```
begin try:A
begin catch:C
```

TRY/ CATCH

```
begin try
```

```
update TEACHER set TEACHER = 'YPB' where TEACHER = 'CMJB'
```

```
end try
```

```
begin catch
```

```
    print 'catch'
```

```
    print ERROR_NUMBER()
```

```
    print ERROR_MESSAGE()
```

```
    print ERROR_LINE()
```

```
    print ERROR_PROCEDURE()
```

```
    print ERROR_SEVERITY()
```

```
    print ERROR_STATE()
```

```
end catch
```

```
catch
```

```
2627
```

Нарушение "PK_TEACHER" ограничения PRIMARY KEY. Невозможно вставить повторяющийся ключ в объект "dbo.TEACHER".

3

14

1

RAISERROR

```
declare @x int = 3, @y int = 0, @z int;
begin try
    if @y = 0
        raiserror ('Деление на нуль', 10, 1);
    else set @z = @x/@y;          -- деление на 0
    if exists(select * from PULPIT where PULPIT = 'ИСИТ')
        raiserror ('Нарушение РК', 11, 2);
    else insert PULPIT(PULPIT) values('ИСИТ');
    print 'begin try:B'
end try
begin catch                    -- обработка ошибки
    print 'номер ошибки : ' + cast(error_number() as varchar(6));
    print 'сообщение : ' + error_message();
    print 'уровень : ' + cast(error_severity() as varchar(6));
    print 'метка : ' + cast(error_state() as varchar(8));
    print 'номер строки : ' + cast(error_line() as varchar(8));
    if error_procedure() is not null
        print 'имя процедуры : ' + error_procedure();
end catch
```

Деление на нуль
номер ошибки : 50000
сообщение : Нарушение РК
уровень : 11
метка : 2
номер строки : 7

WAITFOR

```
declare @ct char(8)
set @ct= (
    select CAST(DATEPART(hh, DATEADD(mi, 1, SYSDATETIME())) as CHAR(2))
        + ':' +
        CAST(DATEPART(mi, DATEADD(mi, 1, SYSDATETIME())) as CHAR(2))
    )
select SYSDATETIME(), @ct
waitfor time @ct
select SYSDATETIME()
```

(Отсутствует имя столбца)	(Отсутствует имя столбца)
2011-03-11 02:16:19.2777872	2:17

(Отсутствует имя столбца)
2011-03-11 02:17:00.0016880

WAITFOR

```
select SYSDATETIME(), @ct  
waitfor time @ct  
select SYSDATETIME()
```

```
declare @i int = 0  
select SYSDATETIME()  
while @i < 5  
begin  
    waitfor delay '00:00:05'  
    select SYSDATETIME()  
    set @i = @i+1  
end
```

2011-03-11 02:26:11.0584448

(Отсутствует имя столбца)

2011-03-11 02:26:16.0604000

(Отсутствует имя столбца)

2011-03-11 02:26:21.0613520

(Отсутствует имя столбца)

2011-03-11 02:26:26.0623040

(Отсутствует имя столбца)

2011-03-11 02:26:31.0632560

(Отсутствует имя столбца)

2011-03-11 02:26:36.0652112

Понятие курсора

- Курсор – программная конструкция, которая служит для хранения результата запроса и для обработки строк результирующего набора запись за записью

Понятие курсора

1. Курсор объявляется в операторе DECLARE.
2. Курсор открывается с помощью оператора OPEN.
3. С помощью оператора FETCH считывается одна или несколько строк результирующего набора, связанного с курсором SELECT-оператора, и обрабатывается нужным образом. Результат каждого считывания проверяется с помощью системной функции @@FETCH_STATUS.
4. Курсор закрывается оператором CLOSE.
5. Если курсор глобальный, то он должен быть освобожден с использованием оператора DEALLOCATE.

Понятие курсора

```
declare @tid char(10), @tnm char(40), @tgn char(1);  
declare c_teacher cursor global                                -- 1  
        for select TEACHER, TEACHER_NAME, GENDER            -- 1  
        from TEACHER where PULPIT = 'ПОИСОИ';                -- 1
```

Понятие курсора

```
open c_teacher;                                -- 2
fetch c_teacher into @tid, @tnm, @tgn;        -- 3
while @@fetch_status = 0                      -- 3
begin
    print @tid + ' ' + @tnm + ' ' + @tgn;
    fetch c_teacher into @tid, @tnm, @tgn;    -- 3
end;
close c_teacher;                              -- 4
deallocate c_teacher;                         -- 5
```

БРТШВЧ	Барташевич Святослав Александрович	М
ЮДНКВ	Юденков Виктор Степанович	М
.....

Программные объекты

- Хранимые процедуры
- Скалярные и табличные функции
- Триггеры

Хранимые процедуры

```
use BSTU
go
create procedure GET_PULPITS
as begin                                -- начало кода
    select * from PULPIT;              -- код процедуры
end;                                    -- конец кода
```

```
use BSTU
go
-- допускается сокращение ключевого
-- слова EXECUTE
exec GET PULPITS; -- вызов процедуры
```

PULPIT	PULPIT_NAME	FACULTY
БФ	Белорусской филологии	ИДиП
ИСиТ	Информационных систем и технологий	ИДиП
ЛВ	Лесоводства	NULL
ЛЗидВ	Лесозащиты и лесосиноведения	NULL

ФУНКЦИИ

```
create function COUNT_STUDENTS(@faculty varchar(20)) returns int
as begin
    declare @rc int = 0;
    set @rc = (
        select count(idstudent)
        from FACULTY f join GROUPS g on f.FACULTY = g.FACULTY
        join STUDENT s on s.IDGROUP = g.IDGROUP
        where f.FACULTY = @faculty
    );
    return @rc;
end;
```

Функции

```
declare @c int = dbo.COUNT_STUDENTS('ИдиП');  
print 'количество студентов на факультете ИдиП: ' +  
      cast(@c as varchar(4));
```

количество студентов на факультете ИдиП: 64

Триггеры

```
- create trigger AUD_AFTER_UPDA on Товары after UPDATE  
as print 'AUD_AFTER_UPDATE_A';  
return;
```

Вопросы?