

# БАЗЫ ДАННЫХ

---

Лекция 11 Процедуры и функции

# Процедуры и функции

- Пакет (batch)
- Подпрограмма (routine)
  - Процедура
  - Функция
- Системные
- Пользовательские

# Пакет

- Пакет — это последовательность инструкций, которые отправляются системе базы данных для совместного их выполнения
- Преимущество – одновременное исполнение всех инструкций позволяет получить улучшение производительности

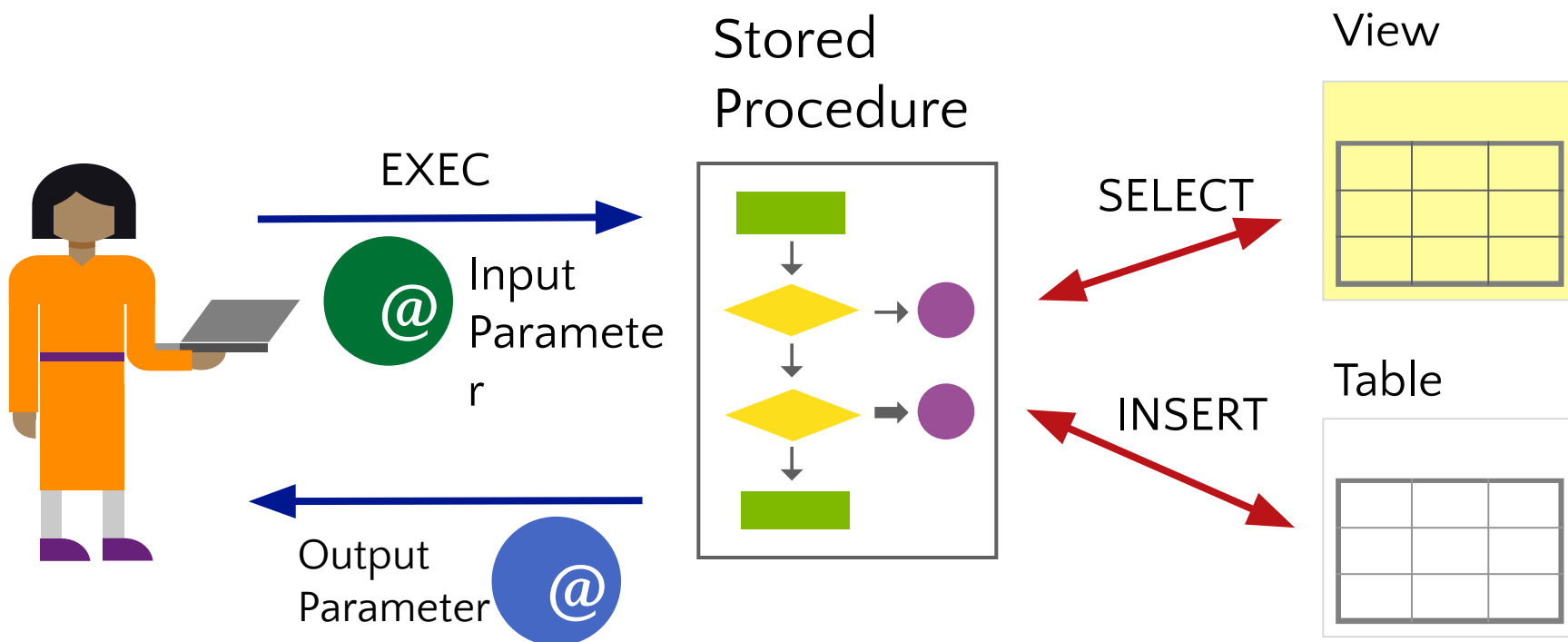
# Пакет

- Ограничение на единственную инструкцию:
- CREATE VIEW
- CREATE PROCEDURE
- CREATE TRIGGER

# Хранимые процедуры

- Хранимая процедура – объект базы данных
- Хранимая процедура – поименованный блок (BEGIN...END) операторов, хранящийся в базе данных в откомпилированном виде

# Хранимые процедуры



# Хранимые процедуры

- Принимает входные параметры
- Принимаете и формирует выходные параметры

# Хранимые процедуры

- Целочисленное значение, возвращаемое к точке вызова с помощью оператора RETURN
- Один или более результирующих наборов, сформированных операторами SELECT
- Содержимое стандартного выходного потока, полученного при выполнении операторов PRINT



# Хранимые процедуры

Результирующий набор хранимой процедуры может быть использован в качестве источника строк для INSERT

# Хранимые процедуры

- Допускается применение :
  - Основных DDL, DML и TCL-операторов
  - Конструкций TRY/CATCH
  - Курсоров
  - Временных таблиц

# Хранимые процедуры

- Не допускается применение :
  - CREATE or ALTER FUNCTION
  - CREATE or ALTER TRIGGER
  - CREATE or ALTER PROCEDURE
  - CREATE or ALTER VIEW
  - USE databasename

# Хранимые процедуры

```
CREATE TABLE TLESSON
(
  TLESSON          CHAR(10) NOT NULL,
  TLESSON_NAME    VARCHAR(30),
  CONSTRAINT PK_TLESSON PRIMARY KEY (TLESSON)
);
```

TLESSON	TLESSON_NAME
ДП	Дипломное проектирование
ЗП	Практическое занятие
ЗЧ	Зачет
КП	Курсовое проектирование
КЧ	Классный час
ЛК	Лекция
ЛР	Лабораторная работа
ПД	Преддипломная практика
ПИ	Инженерная практика
ПП	Программистская практика
ПР	Производственная практи...
СМ	Семинар
ФК	Занятие физкультурой
ЭК	Экзамен

# Хранимые процедуры

```
drop procedure PrintTLESSON
```

```
create procedure PrintTLESSON
```

```
as
```

```
declare @lt char(10), @ln varchar(30)
```

```
declare ctl cursor local read_only for select * from TLESSON order by tlesson
```

```
begin
```

```
print '----- Типы занятий (TLESSON)-----'
```

```
open ctl
```

```
fetch ctl into @lt, @ln
```

```
while @@FETCH_STATUS = 0
```

```
begin
```

```
print @lt+' '+@ln
```

```
fetch ctl into @lt, @ln
```

```
end
```

```
close ctl
```

```
end
```

```
exec PrintTLESSON
```

```
----- Типы занятий (TLESSON)-----
```

```
ДП          Дипломное проектирование
```

```
ЗП          Практическое занятие
```

```
ЗЧ          Зачет
```

```
КП          Курсовое проектирование
```

```
КЧ          Классный час
```

```
ЛК          Лекция
```

```
ЛР          Лабораторная работа
```

```
ПД          Преддипломная практика
```

```
ПИ          Инженерная практика
```

```
ПП          Программисткая практика
```

```
ПР          Производственная практика
```

```
СМ          Семинар
```

```
ФК          Занятие физкультурой
```

```
ЭК          Экзамен
```

# Хранимые процедуры

```
drop procedure SelectTLESSON
```

```
create procedure SelectTLESSON
```

```
as
```

```
begin
```

```
    select * from TLESSON
```

```
end
```

```
exec SelectTLESSON
```

TLESSON	TLESSON_NAME
ДП	Дипломное проектирование
ЗП	Практическое занятие
ЗЧ	Зачет
КП	Курсовое проектирование
КЧ	Классный час
ЛК	Лекция
ЛР	Лабораторная работа
ПД	Преддипломная практика
ПИ	Инженерная практика
ПП	Программисткая практика
ПР	Производственная практика
СМ	Семинар
ФК	Занятие физкультурой
ЭК	Экзамен

# Хранимые процедуры

```
declare @x int, @y int
print '1)@@ROWCOUNT = '+CAST(@@ROWCOUNT as varchar(10));
select * from TLESSON
print '2)@@ROWCOUNT = '+CAST(@@ROWCOUNT as varchar(10));
print '3)@@ROWCOUNT = '+CAST(@@ROWCOUNT as varchar(10));
set @x = 1
print '4)@@ROWCOUNT = '+CAST(@@ROWCOUNT as varchar(10));
select @x = 2, @y = 3
print '5)@@ROWCOUNT = '+CAST(@@ROWCOUNT as varchar(10));
```

TLESSON	TLESSON_NAME
ДП	Дипломное проектирование
ЗП	Практическое занятие
ЗЧ	Зачет
КП	Курсовое проектирование
КЧ	Классный час
ЛК	Лекция
ЛР	Лабораторная работа
ПД	Преддипломная практика
ПИ	Инженерная практика
ПП	Программисткая практика
ПР	Производственная практика
СМ	Семинар
ФК	Занятие физкультурой
ЭК	Экзамен

```
1)@@ROWCOUNT = 0
```

```
(строк обработано: 14)
```

```
2)@@ROWCOUNT = 14
```

```
3)@@ROWCOUNT = 0
```

```
4)@@ROWCOUNT = 1
```

```
5)@@ROWCOUNT = 1
```

```
|
```

# Хранимые процедуры

```
create procedure SelectTLESSON  
as  
  declare @rc int = 0  
begin  
  select * from TLESSON  
  if @@ROWCOUNT = 0 set @rc = -1  
  return @rc  
end
```

Возможен  
только  
числовой код  
возврата

```
declare @code int = 77  
exec @code = SelectTLESSON  
print 'код возврата = ' + CAST(@code as varchar(10))
```

```
(строк обработано: 14)  
код возврата = 0
```



# Хранимые процедуры

```
create procedure GetExamResults -- создать процедуру с именем GetExamResults
as                               -- хранится в БД BSTU
begin -- формирует результирующий набор
    select st.NAME, pr.PDATE, sb.SUBJECT_NAME, pr.NOTE
        from STUDENT st join PROGRESS pr on st.IDSTUDENT = pr.IDSTUDENT
            join [SUBJECT] sb on pr.[SUBJECT] = sb.[SUBJECT];
end;
go;
execute GetExamResults; -- вызов процедуры
```

NAME	PDATE	SUBJECT_NAME	NOTE
Манакова Анастасия Владимировна	2013-01-10	Основы алгоритмизации и программирования	6
Хартанович Екатерина Александровна	2013-01-10	Основы алгоритмизации и программирования	8
Горбач Елизавета Юрьевна	2013-01-10	Основы алгоритмизации и программирования	7
Зыкова Кристина Дмитриевна	2013-01-10	Основы алгоритмизации и программирования	5

```
use BSTU
go
insert [#Результаты экзамена] execute GetExamResults;
```

# ROW\_NUMBER() OVER

```
select ROW_NUMBER() over (order by TEACHER), TEACHER, TEACHER_NAME from TEACHER
```

(Отсутствует имя столбца)	TEACHER	TEACHER_NAME
1	?	Неизвестный
2	АКНВЧ	Акунович Станислав Иванович
3	БЗБРДВ	Безбородов Владимир Степанович
4	БРКВЧ	Бракович Андрей Игорьевич
5	БРНВСК	Барановский Станислав Иванович
6	БРТШВЧ	Барташевич Святослав Александрович
7	ГРМН	Герман Олег Витольдович
8	ГРН	Гурин Николай Иванович
9	ДДК	Дедко Александр Аркадьевич
10	ДМДК	Демидко Марина Николаевна
11	ЕЩНК	Ещенко Людмила Семеновна
12	ЖЛК	Жиляк Надежда Александровна
13	ЖРСК	Жарский Иван Михайлович
14	ЗВГЦВ	Звягинцев Вячеслав Борисович
15	КБЛ	Кабайло Александр Серафимович
16	КЛСНВ	Колесников Леонид Валерьевич
17	ЛБХ	Лабоха Константин Валентинович
18	ЛЩНК	Лашенко Анатолий Пвалович
19	МХВ	Мохов Сергей Петрович
20	МШКВСК	Машковский Владимир Петрович
21	НВРВ	Неверов Александр Васильевич
22	НСКВЦ	Насковец Михаил Трофимович
23	ПРКПЧК	Прокочук Николай Романович
24	ПСТВЛВ	Пустовалова Наталия Николаевна

# ROW\_NUMBER() OVER

```
select ROW_NUMBER() over (partition by PULPIT order by TEACHER),  
       PULPIT, TEACHER, TEACHER_NAME from TEACHER
```

1	ИСиТ	?	Неизвестный
2	ИСиТ	АКНВЧ	Акунович Станислав Иванович
3	ИСиТ	БРКВЧ	Бракович Андрей Игорьевич
4	ИСиТ	ГРМН	Герман Олег Витольдович
5	ИСиТ	ГРН	Гурин Николай Иванович
6	ИСиТ	ДДК	Дедко Александр Аркадьевич
7	ИСиТ	ЖЛК	Жиляк Надежда Александровна
8	ИСиТ	КБЛ	Кабайло Александр Серафимович
9	ИСиТ	КЛСНВ	Колесников Леонид Валерьевич
10	ИСиТ	ЛЩНК	Лашенко Анатолий Пвалович
11	ИСиТ	ПСТВЛВ	Пустовалова Наталия Николаевна
12	ИСиТ	РМНК	Романенко Дмитрий Михайлович
13	ИСиТ	СМЛВ	Смелов Владимир Владиславович
14	ИСиТ	УРБ	Урбанович Павел Павлович
1	ЛВ	ЛБХ	Лабоха Константин Валентинович
1	ЛЗидВ	ЗВГЦВ	Звягинцев Вячеслав Борисович
1	ЛМиЛЗ	МХВ	Мохов Сергей Петрович
1	ЛПиСПС	ДМДК	Демидко Марина Николаевна
1	ЛУ	МШКВСК	Машковский Владимир Петрович
1	МиЭП	НВРВ	Неверов Александр Васильевич
1	ОВ	РВКЧ	Ровкач Андрей Иванович
1	ОХ	БЗБРДВ	Безбородов Владимир Степанович
1	ПОиСОИ	БРТШВЧ	Барташевич Святослав Александрович
2	ПОиСОИ	ЮДНКВ	Юденков Виктор Степанович
1	ТЛ	НСКВЦ	Насковец Михаил Трофимович

# WITH

```
with RRR as
```

```
(  
select ROW_NUMBER() over (order by TEACHER) rn, TEACHER, TEACHER_NAME from TEACHER  
)
```

```
select * from RRR where rn < 10
```

n	TEACHER	TEACHER_NAME
1	?	Неизвестный
2	АКНВЧ	Акунович Станислав Иванович
3	БЗБРДВ	Безбородов Владимир Степанович
4	БРКВЧ	Бракович Андрей Игорьевич
5	БРНВСК	Барановский Станислав Иванович
6	БРТШВЧ	Барташевич Святослав Александрович
7	ГРМН	Герман Олег Витольдович
8	ГРН	Гурин Николай Иванович
9	ДДК	Дедко Александр Аркадьевич

# Передача параметров

```
create procedure SelectTEACHER
  @p char(10),
  @n int = 5,
  @c int output
as
  declare @rc int = 0
begin
  begin try
    select * from (
      select ROW_NUMBER() over (order by TEACHER) rn, TEACHER, TEACHER_NAME
      from TEACHER
      where PULPIT = @p
      ) rrr where rrr.rn < @n
    set @c = (select COUNT(*) from TEACHER)
  end try
  begin catch
    set @rc = -1
  end catch
  return @rc
end
```

# Передача параметров

```
declare @code int, @ttt int
exec @code = SelectTEACHER 'ИСиТ', default, @ttt output
print 'код возврата = ' + CAST(@code as varchar(10))
print 'output = ' + CAST(@ttt as varchar(10))
```

n	TEACHER	TEACHER_NAME
1	?	Неизвестный
2	АКНВЧ	Акунович Станислав Иванович
3	БРКВЧ	Бракович Андрей Игорьевич
4	ГРМН	Герман Олег Витольдович

(строк обработано: 4)

код возврата = 0

output = 29

# Передача параметров

```
declare @code int, @ttt int
exec @code = SelectTEACHER 'ИСиТ', 2, @ttt output
print 'код возврата = ' + CAST(@code as varchar(10))
print 'output = ' + CAST(@ttt as varchar(10))
```

---

m	TEACHER	TEACHER_NAME
1	?	Неизвестный

(строк обработано: 1)

код возврата = 0

output = 29

# Хранимые процедуры

+  Диаграммы баз данных

+  Таблицы

+  Представления

+  Синонимы

-  Программирование

-  Хранимые процедуры


+  Системные хранимые процедуры

+  dbo.PrintTLESSON

-  dbo.SelectTEACHER

-  Параметры

 @p (char(10), Ввод, Нет значения по умолчанию)

 @n (int, Ввод, По умолчанию)

 @c (int, Ввод-вывод, Нет значения по умолчанию)

 Возвращает целое значение

+  dbo.SelectTLESSON

+  Функции



# Хранимые процедуры

- ALTER PROCEDURE
- DROP PROCEDURE
- sp\_rename

# Хранимые процедуры

```
use BSTU
go
drop procedure GET_PULPITS; -- удаление процедуры
go
create procedure GET_PULPITS -- создание процедуры
as begin
    declare @rc int = (select count(*) from PULPIT) ;
    select * from PULPIT;
    return @rc; -- код возврат
end;
```

# Хранимые процедуры

```
use BSTU
go
alter procedure GET_PULPITS  -- модификация процедуры
    @f varchar(20),
    @c int output
as begin
    declare @rc int = (select count(*) from PULPIT) ;
    print 'параметры: @f = '+@f +', @c = '+ cast(@c as varchar(3));
    select * from PULPIT where FACULTY = @f;
    set @c = @@rowcount;
    return @rc;  -- код возврат
end;
```

# Динамический SQL

- `sp_executesql`

```
DECLARE @sqlcode AS NVARCHAR(256) =  
N'<code_to_run>';  
EXEC sys.sp_executesql @statement = @sqlcode;
```

```
DECLARE @sqlcode AS NVARCHAR(256) =  
N'SELECT GETDATE() AS dt';  
EXEC sys.sp_executesql @statement = @sqlcode;
```

# Функции

- Встроенные
  - Математические
  - Строковые
  - Работа с датами
- Пользовательские

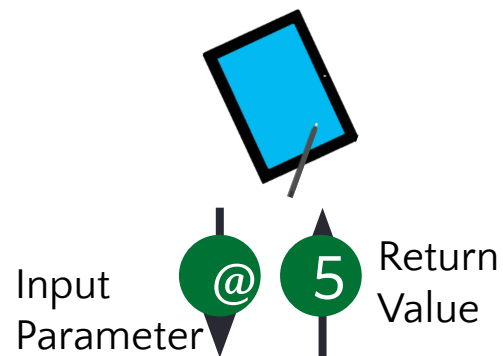
# Функции

- Не допускается применение:
  - DDL-операторов
  - DML-операторов изменяющих данные (INSERT, DELETE, UPDATE)
  - конструкций TRY/CATCH
  - транзакций

# Функции

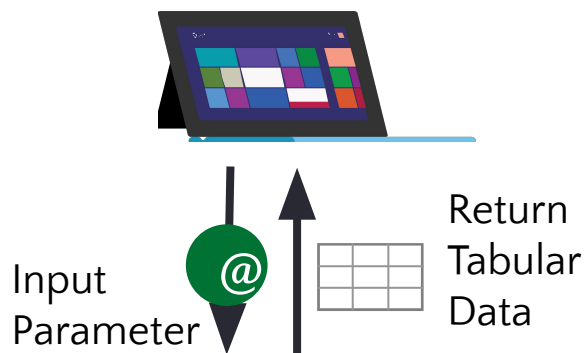
- Скалярные
- Inline
- Multi-Statement

# ФУНКЦИИ



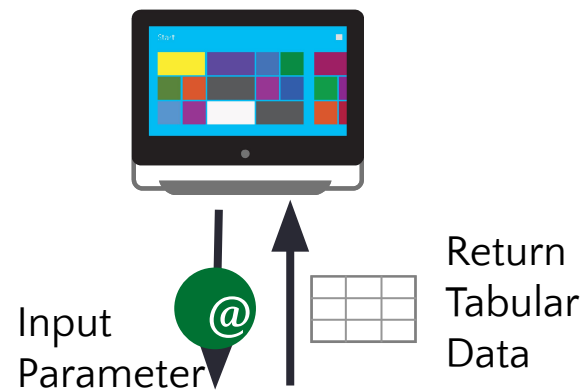
```
SELECT ...  
RETURN
```

Scalar  
Function



```
SELECT ...  
RETURN
```

Inline Table-  
Valued  
Function



```
INSERT ...  
SELECT ...  
RETURN
```

Multi-statement  
Table-Valued  
Function



# Функции

- Скалярные
  - возвращает одно значение
  - нельзя timestamp, cursor, text, ntext, image, table

# Хранимые процедуры

```
drop function countTEACHER
```

```
create function countTEACHER(@p char(10)) returns int  
begin  
  declare @rc int = (select COUNT(*) from TEACHER where PULPIT = @p)  
  return @rc  
end
```

```
declare @x int = dbo.countTEACHER('ИСИТ')  
print ' преподавателей ИСИТ = ' + cast(@x as varchar(10))  
select dbo.countTEACHER('ИСИТ')
```

(Отсутствует имя столбца)

14

# Функции

- Inline

- одна команда SELECT
- возврат table

```
create function dbo.fnPROFESSION() returns table
as return select PROFESSION, FACULTY, QUALIFICATION
from PROFESSION;
```

```
select * from fnPROFESSION()
```

PROFESSION	FACULTY	QUALIFICATION
1-25 01 07	ИЭФ	экономист-менеджер
1-25 01 08	ИЭФ	экономист

# ФУНКЦИИ

```
alter function dbo.fnPROFESSION(@f varchar(20) = null, @like_q varchar(30) = null)
returns table
as return select PROFESSION, FACULTY, QUALIFICATION
from PROFESSION
where FACULTY = isnull(@f, FACULTY) and
      1 = case
            when @like_q is null then 1
            when QUALIFICATION like @like_q then 1
            else 0
          end;
end;
```

# Функции

```
select * from fnPROFESSION(default, default);
```

PROFESSION	FACULTY	QUALIFICATION
1-25 01 07	ИЭФ	экономист-менеджер

```
select * from fnPROFESSION('ИЭФ', default);
```

PROFESSION	FACULTY	QUALIFICATION
1-25 01 07	ИЭФ	экономист-менеджер
1-25 01 08	ИЭФ	экономист

# Функции

```
select * from fnPROFESSION(default, 'экономика');
```

PROFESSION	FACULTY	QUALIFICATION
1-25 01 07	ИЭФ	экономист-менеджер
1-25 01 08	ИЭФ	экономист

```
select * from fnPROFESSION('ХТИТ', 'механика');
```

PROFESSION	FACULTY	QUALIFICATION
1-36 01 08	ХТИТ	инженер-механик
1-36 07 01	ХТИТ	инженер-механик

# Функции

```
select count(*) 'кол. механиков на ХТИТ'  
from GROUPS g join fnPROFESSION('ХТИТ', '%механик%') p  
    on g.PROFESSION = p.PROFESSION  
    join STUDENT s  
    on s.IDGROUP = g.IDGROUP
```

кол. механиков на ХТИТ
21

```
select f.FACULTY, p.QUALIFICATION  
    from FACULTY f left outer join fnPROFESSION(default, '%механик%') p  
    on f.FACULTY = p.FACULTY
```

FACULTY	QUALIFICATION
ИДиП	инженер-электромеханик
ИЭФ	NULL
ЛХФ	NULL
ТОВ	NULL
ТТЛП	инженер-механик
УТТ	

# Функции

- Multi-Statement
- возвращает table
- несколько команд



# Функции

```
create function FACULTY_REPORT (@c int)
returns @fr table
(
    [Факультет]                varchar(50),
    [Количество кафедр]       int,
    [Количество групп]        int,
    [Количество студентов]    int,
    [Количество специальностей] int
)
```

```
as begin
    declare cc cursor static for
        select FACULTY from FACULTY
        where dbo.COUNT_STUDENTS(FACULTY, default) > @c;
    declare @f varchar(30);
    open cc;
    fetch cc into @f;
    while @@fetch_status = 0
    begin
        insert @fr values(
            @f,
            (select count(PULPIT) from PULPIT where FACULTY = @f),
            (select count(IDGROUP) from GROUPS where FACULTY = @f),
            dbo.COUNT_STUDENTS(@f, default),
            (select count(PROFESSION) from PROFESSION where FACULTY = @f)
        );
        fetch cc into @f;
    end;
    return;
end;
```

# Функции

```
select * from FACULTY_REPORT (28)
```

Факультет	Количество кафедр	Количество групп	Количество студентов	Количество специальностей
ИДиП	5	11	64	3
ИЭФ	3	5	29	2

# Хранимые процедуры

- DROP FUNCTION
- ALTER FUNCTION

Вопросы?