

# БАЗЫ ДАННЫХ

---

Лекция 7 Подзапросы  
Группировка данных

# Лекция 7

- Подзапросы
- Операции над множествами
- Группировки
- DML

# Подзапросы

- Конструкция SELECT, которая содержится в другой инструкции SELECT, называется подзапросом
- Первая инструкция SELECT подзапроса называется внешним запросом
- Внутренняя инструкция (или инструкции) SELECT, называется вложенным запросом
- Первым выполняется вложенный запрос, а его результат передается внешнему запросу

# Подзапросы

```
CREATE TABLE FACULTY
(
    FACULTY          CHAR(10)          NOT NULL,
    FACULTY_NAME     VARCHAR(50),
    CONSTRAINT PK_FACULTY PRIMARY KEY (FACULTY)
);
```

```
CREATE TABLE PROFESSION
(
    PROFESSION        CHAR(20),
    FACULTY           CHAR(10),
    PROFESSION_NAME   VARCHAR(100),
    QUALIFICATION     VARCHAR(50),
    CONSTRAINT PK_PROFESSION PRIMARY KEY (PROFESSION),
    CONSTRAINT FK_PROFESSION FOREIGN KEY (FACULTY)
        REFERENCES FACULTY (FACULTY)
);
```

# Подзапросы

```
select * from faculty;
```

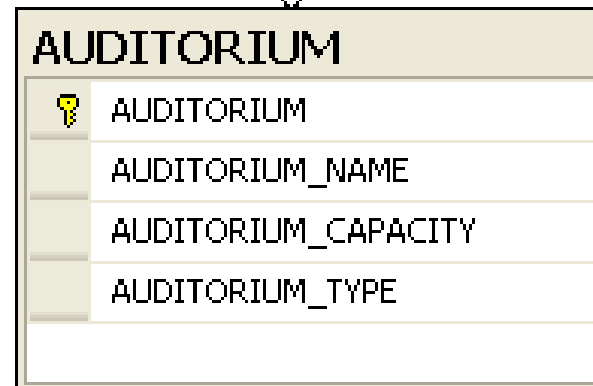
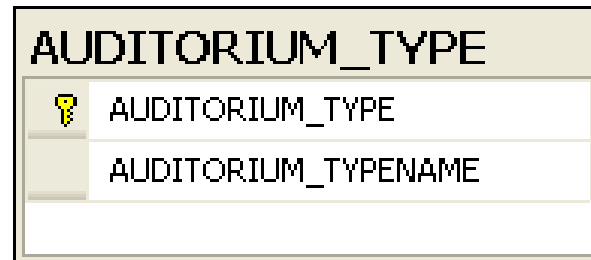
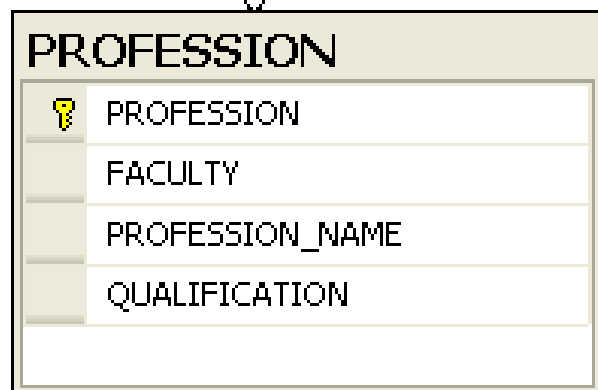
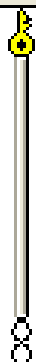
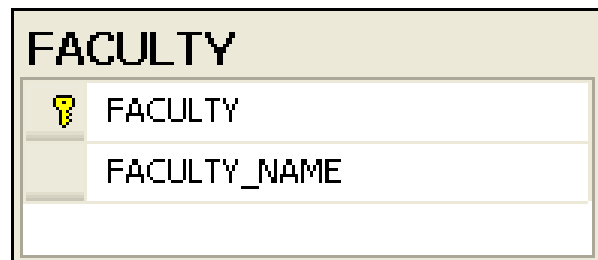
FACULTY	FACULTY_NAME
ИДиП	Издательское дело и полиграфия
ИЭФ	Инженерно-экономический факультет
ЛХФ	Лесохозяйственный факультет
ТОВ	Технология органических веществ
ТТЛП	Технология и техника лесной промыш...
ХТИТ	Химическая технология и техника

# Подзапросы

```
select * from profession
```

PROFESSION	FACULTY	PROFESSION_NAME	QUALIFICATION
1-25 01 07	ИЭФ	Экономика и управление на предприятии	экономист-менеджер
1-25 01 08	ИЭФ	Бухгалтерский учет, анализ и аудит	экономист
1-36 01 08	ХТиТ	Конструирование и производство изделий из компози...	инженер-механик
1-36 05 01	ТТЛП	Машины и оборудование лесного комплекса	инженер-механик
1-36 07 01	ХТиТ	Машины и аппараты химических производств и предп...	инженер-механик
1-40 01 02	ИДиП	Информационные системы и технологии	инженер-программист-системотехник
1-46 01 01	ТТЛП	Лесоинженерное дело	инженер-технолог
1-47 01 01	ИДиП	Издательское дело	редактор-технолог
1-48 01 02	ТОВ	Химическая технология органических веществ, матер...	инженер-химик-технолог
1-48 01 05	ТОВ	Химическая технология переработки древесины	инженер-химик-технолог
1-54 01 03	ТОВ	Физико-химические методы и приборы контроля каче...	инженер по сертификации
1-75 01 01	ЛХФ	Лесное хозяйство	инженер лесного хозяйства
1-75 02 01	ЛХФ	Садово-парковое строительство	инженер садово-паркового строительства
1-89 02 02	ЛХФ	Туризм и природопользование	специалист в сфере туризма

# Подзапросы



# Подзапросы

Найти специальности, которые  
получают на  
Лесохозяйственном  
факультете?



# Подзапросы

- Найти код лесохозяйственного факультета
- Найти специальности по коду факультета

# Подзапросы

```
= SELECT FACULTY FROM FACULTY
```

```
WHERE FACULTY_NAME = 'Лесохозяйственный факультет';
```

%

Results



Messages

FACULTY

ЛХФ

```
= SELECT QUALIFICATION FROM PROFESSION
```

```
WHERE FACULTY = 'ЛХФ';
```

%

Results



Messages

QUALIFICATION

инженер лесного хозяйства

инженер садово-паркового строительства

специалист в сфере туризма

# Подзапросы

```
SELECT QUALIFICATION FROM PROFESSION  
WHERE FACULTY = (SELECT FACULTY FROM FACULTY  
WHERE FACULTY_NAME = 'Лесохозяйственный факультет');
```

Results		Messages	
QUALIFICATION			
инженер лесного хозяйства			
инженер садово-паркового строительства			
специалист в сфере туризма			

# Подзапросы

- FROM
- WHERE
- SELECT
- HAVING

# Подзапросы

- *Коррелируемый* подзапрос зависит от внешнего запроса и выполняется для каждой строки результирующего набора.
- *Независимый* подзапрос не зависит от внешнего запроса и выполняется только один раз, но результат его выполнения подставляется в каждую строку результирующего набора.
- Допускается применять только такие подзапросы, которые формируют скалярный результирующий набор

# Подзапросы

Найти студентов, оценки которых по предмету меньше средней оценки по этому предмету

# Подзапросы

```
SELECT * FROM PROGRESS;
```

Results			
Messages			
SUBJECT	IDSTUDENT	PDATE	NOTE
ОАиП	1000	2013-01-10	6
ОАиП	1001	2013-01-10	8
ОАиП	1002	2013-01-10	7
ОАиП	1003	2013-01-10	5
ОАиП	1005	2013-01-10	4
СУБД	1014	2013-01-12	5
СУБД	1015	2013-01-12	9
СУБД	1016	2013-01-12	5
СУБД	1017	2013-01-12	4
КГ	1018	2013-06-05	4
КГ	1019	2013-06-05	7
КГ	1020	2013-06-05	7
КГ	1021	2013-06-05	9
КГ	1022	2013-06-05	5
КГ	1023	2013-06-05	6
ОХ	1064	2013-01-01	6
ОХ	1065	2013-01-01	4
ОХ	1066	2013-01-01	9

```
SELECT * FROM STUDENT;
```

Results			
Messages			
IDSTUDENT	IDGROUP	NAME	
1000	1	Хартанович Екатерина Александровна	
1001	1	Горбач Елизавета Юрьевна	
1002	1	Зыкова Кристина Дмитриевна	
1003	1	Борисевич Ольга Анатольевна	
1004	1	Медведева Мария Андреевна	
1005	1	Шенец Екатерина Сергеевна	
1006	1	Шитик Алина Игоревна	
1007	2	Силюк Валерия Ивановна	
1008	2	Сергель Виолетта Николаевна	
1009	2	Добродей Ольга Анатольевна	
1010	2	Подольак Мария Сергеевна	
1011	2	Никитенко Екатерина Дмитриевна	
1012	3	Яцкевич Галина Иосифовна	

# Подзапросы

```
SELECT SUBJECT, AVG(NOTE) FROM PROGRESS GROUP BY SUBJECT;
```

Results

Messages

SUBJECT	(No column name)
КГ	6
ОАиП	6
ОХ	6
СУБД	5
ЭТ	6



```

SELECT NAME, SUBJECT, NOTE
FROM STUDENT S1 INNER JOIN PROGRESS P1
ON S1.IDSTUDENT = P1.IDSTUDENT
WHERE NOTE < (SELECT AVG(NOTE) FROM PROGRESS P2
              WHERE P1.SUBJECT = P2.SUBJECT
              GROUP BY SUBJECT)
ORDER BY NAME;

```

NAME	SUBJECT	NOTE
Абрамов Денис Дмитриевич	ЭТ	5
Борисевич Ольга Анатольевна	ОАиП	5
Бороховский Виталий Петрович	СУБД	2
Буянова Мария Александровна	КГ	4
Власик Евгения Викторовна	ЭТ	4
Горбач Елизавета Юрьевна	СУБД	3
Даниленко Максим Васильевич	ОХ	5
Кузмичева Анна Андреевна	ОХ	4
Мацкевич Надежда Валерьевна	СУБД	2
Мацкевич Надежда Валерьевна	КГ	5
Плешкун Милана Анатольевна	ОАиП	4
Плешкун Милана Анатольевна	СУБД	4
Шарапо Мария Владимировна	ОХ	4
Шенец Екатерина Сергеевна	ОАиП	4
Шенец Екатерина Сергеевна	СУБД	4

# Подзапросы

Может применяться со следующими операторами:

- операторами сравнения и EXISTS;
- оператором IN;
- операторами ANY и ALL;

# Подзапросы

```
select * from AUDITORIUM
|
select AUDITORIUM_TYPENAME
from AUDITORIUM_TYPE at
where exists (
    select * from AUDITORIUM aa
    where at.AUDITORIUM_TYPE = aa.AUDITORIUM_TYPE
    and aa.AUDITORIUM_CAPACITY > 60
)
```

AUDITORIUM_TYPENAME
Компьютерный класс
Лекционная
Лекционная с уст. компьютерами

# Подзапросы

```
select AUDITORIUM_TYPENAME
from AUDITORIUM_TYPE at
where not exists (
    select * from AUDITORIUM aa
    where at.AUDITORIUM_TYPE = aa.AUDITORIUM_TYPE
    and aa.AUDITORIUM_CAPACITY > 60
)
```

AUDITORIUM_TYPENAME
XXXXX
Химическая лаборатория
Спец. компьютерный класс

# Подзапросы

```
select f.FACULTY_NAME, p.QUALIFICATION, p.PROFESSION
from FACULTY f join (
    select FACULTY, QUALIFICATION, PROFESSION
    from PROFESSION
    where (QUALIFICATION like '%технолог%')
) p
on f.FACULTY = p.FACULTY
```

FACULTY_NAME	QUALIFICATION	PROFESSION
Технология и техника лесной промышленности	инженер-технолог	1-46 01 01
Издательское дело и полиграфия	редактор-технолог	1-47 01 01
Технология органических веществ	нженер-химик-технолог	1-48 01 02
Технология органических веществ	нженер-химик-технолог	1-48 01 05

# Подзапросы с IN

```
select FACULTY_NAME
from FACULTY
where FACULTY in (select FACULTY from PROFESSION where (QUALIFICATION like '%технолог%'))
```

---

FACULTY_NAME
Издательское дело и полиграфия
Технология органических веществ
Технология и техника лесной промышленности

```
select FACULTY_NAME
```

```
from FACULTY
```

```
where FACULTY not in (select FACULTY from PROFESSION where (QUALIFICATION like '%технолог%'))
```

---

FACULTY_NAME
Инженерно-экономический факультет
Лесохозяйственный факультет
Химическая технология и техника

# Подзапросы с ANY и ALL

**column\_name operator** [ANY|ALL]  
**query**

Оператор ANY возвращает значение TRUE, если результат вложенного запроса содержит хотя бы одну строку, удовлетворяющую условию сравнения.

# Подзапросы с ANY и ALL

```
select AUDITORIUM_TYPENAME
from AUDITORIUM_TYPE at
where 60 >= ANY(
    select AUDITORIUM_CAPACITY from AUDITORIUM aa
    where at.AUDITORIUM_TYPE = aa.AUDITORIUM_TYPE
)
```

AUDITORIUM_TYPENAME
Компьютерный класс
Лекционная



# Подзапросы с ANY и ALL

**column\_name operator [ANY|ALL] query**

Оператор ALL возвращает значение TRUE, если результат вложенного запроса возвращает все значения, обрабатываемого столбца, удовлетворяющие условию сравнения.

# Подзапросы с ANY и ALL

```
SELECT IDSTUDENT, SUBJECT, NOTE  
FROM PROGRESS  
WHERE NOTE >= ALL(SELECT NOTE FROM PROGRESS)  
ORDER BY IDSTUDENT;
```

%

Results

Messages

IDSTUDENT	SUBJECT	NOTE
1015	СУБД	9
1021	КГ	9
1023	СУБД	9
1057	ЭТ	9
1066	ОХ	9

# Соединения или подзапросы?

# Соединения или подзапросы?

- Подзапросы - когда требуется вычислить агрегатное значение "на лету" и использовать его в другом запросе для сравнения.
- `SELECT emp_no, enter_date`
- `FROM works_on`
- `WHERE enter_date = (SELECT min(enter_date) FROM works_on);`

# Соединения или подзапросы?

- Соединения - когда список выбора инструкции `SELECT` в запросе содержит столбцы более чем из одной таблицы.
- `SELECT employee, empid, name, job`
- `FROM employee INNER JOIN works_on`
- `ON employee.empid = works_on.empid`
- `AND enter_date = '11.03.2017';`

# Подзапросы

- Список выбора внутреннего подзапроса, которому предшествует операция сравнения, может содержать только одно выражение или название столбца, и подзапрос должен возвращать единственный результат.
- Тип данных столбца, указанного в конструкции where внешнего оператора, должен быть совместим с типом данных в столбце, указанным в списке выбора подзапроса

# Подзапросы

- Количество вложенных уровней для подзапросов не должно превышать 16
- Максимальное число подзапросов на каждой стороне объединения не больше 16

# Операции над множествами

- UNION
- UNION ALL
- INTERSECT
- EXCEPT



# UNION

```
select * from AUDITORIUM where AUDITORIUM_CAPACITY < 30  
union  
select * from AUDITORIUM where AUDITORIUM_CAPACITY > 60
```

AUDITORIUM	AUDITORIUM_NAME	AUDITORIUM_CAPACITY	AUDITORIUM_TYPE
206-1	206-1	15	ЛБ-К
301-1	301-1	15	ЛБ-К
413-1	413-1	15	ЛБ-К
?	???	90	ЛК
025-4	025-4	90	ЛК
103-4	103-4	90	ЛК
105-4	105-4	90	ЛК
107-4	107-4	90	ЛК
114-4	114-4	90	ЛК-К
132-4	132-4	90	ЛК
229-4	229-4	90	ЛК
304-4	304-4	90	ЛБ-К
314-4	314-4	90	ЛК
320-4	320-4	90	ЛК
408-2	408-2	90	ЛК
423-1	423-1	90	ЛБ-К
429-4	429-4	90	ЛК

# UNION ALL

```
select * from AUDITORIUM where AUDITORIUM_CAPACITY < 30  
union all  
select * from AUDITORIUM where AUDITORIUM_CAPACITY < 60
```

AUDITORIUM	AUDITORIUM_NAME	AUDITORIUM_CAPACITY	AUDITORIUM_TYPE
206-1	206-1	15	ЛБ-К
301-1	301-1	15	ЛБ-К
413-1	413-1	15	ЛБ-К
110-4	110-4	30	ЛК
111-4	111-4	30	ЛК
206-1	206-1	15	ЛБ-К
301-1	301-1	15	ЛБ-К
324-1	324-1	50	ЛК
413-1	413-1	15	ЛБ-К

# INTERSECT

```
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
  from AUDITORIUM where AUDITORIUM_CAPACITY < 30  
intersect  
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
  from AUDITORIUM where AUDITORIUM_CAPACITY > 60
```

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY
------------	-----------------	---------------------

```
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
  from AUDITORIUM where AUDITORIUM_CAPACITY > 30  
intersect  
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
  from AUDITORIUM where AUDITORIUM_CAPACITY < 90
```

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY
236-1	ЛК	60
313-1	ЛК	60
324-1	ЛК	50

# EXCEPT

```
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
      from AUDITORIUM where AUDITORIUM_CAPACITY < 50  
except  
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
      from AUDITORIUM where AUDITORIUM_CAPACITY < 60
```

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY
------------	-----------------	---------------------

```
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
      from AUDITORIUM where AUDITORIUM_CAPACITY < 60  
except  
select AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY  
      from AUDITORIUM where AUDITORIUM_CAPACITY < 50
```

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY
324-1	ПК	50

# Операции над множествами

- Приоритет:
  - INTERSECT
  - UNION и EXCEPT

# Группировки

- GROUP BY
- GROUP BY HAVING
- GROUP BY WITH CUBE
- GROUP BY WITH ROLLUP

# GROUP BY

```
select AUDITORIUM_TYPE, COUNT(*) as 'количество'  
from AUDITORIUM group by AUDITORIUM_TYPE;
```

AUDITORIUM_TYPE	количество
ЛБ-К	5
ЛК	16
ЛК-К	1

# GROUP BY

```
select AUDITORIUM_TYPE, COUNT(*) as 'количество'  
from AUDITORIUM  
where AUDITORIUM_CAPACITY > 15  
group by AUDITORIUM_TYPE
```

AUDITORIUM_TYPE	количество
ЛБ-К	2
ЛК	16
ЛК-К	1

```
select AUDITORIUM_CAPACITY, COUNT(*) as 'количество'  
from AUDITORIUM  
where AUDITORIUM_TYPE != 'ЛБ-К'  
group by AUDITORIUM_CAPACITY
```

AUDITORIUM_CAPACITY	количество
30	2
50	1
60	2
90	12



# GROUP BY HAVING

```
select AUDITORIUM_CAPACITY, COUNT(*)  
  from AUDITORIUM  
  group by AUDITORIUM_CAPACITY  
  having AUDITORIUM_CAPACITY > 15
```

AUDITORIUM_CAPACITY	(Отсутствует имя столбца)
30	2
50	1
60	2
90	14

# GROUP BY HAVING

```
select AUDITORIUM_CAPACITY, COUNT(*), SUM(AUDITORIUM_CAPACITY)
from AUDITORIUM
group by AUDITORIUM_CAPACITY
```

AUDITORIUM_CAPACITY	(Отсутствует имя столбца)	(Отсутствует имя столбца)
15	3	45
30	2	60
50	1	50
60	2	120
90	14	1260

```
select AUDITORIUM_CAPACITY, COUNT(*)
from AUDITORIUM
group by AUDITORIUM_CAPACITY
having SUM(AUDITORIUM_CAPACITY) >= 60
```

AUDITORIUM_CAPACITY	(Отсутствует имя столбца)
30	2
60	2
90	14

# GROUP BY WITH CUBE

```
select AUDITORIUM_TYPE, COUNT(*) as 'количество'  
from AUDITORIUM group by AUDITORIUM_TYPE;
```

AUDITORIUM_TYPE	количество
ЛБ-К	5
ЛК	16
ЛК-К	1

```
select AUDITORIUM_TYPE, COUNT(*) as 'количество'  
from AUDITORIUM group by AUDITORIUM_TYPE with cube
```

AUDITORIUM_TYPE	количество
ЛБ-К	5
ЛК	16
ЛК-К	1
NULL	22

# GROUP BY WITH CUBE

```
select AUDITORIUM_TYPE, AUDITORIUM_CAPACITY, COUNT(*) as 'количество'  
from AUDITORIUM  
group by AUDITORIUM_TYPE, AUDITORIUM_CAPACITY
```

AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	количество
ЛБ-К	15	3
ЛК	30	2
ЛК	50	1
ЛК	60	2
ЛБ-К	90	2
ЛК	90	11
ЛК-К	90	1

# GROUP BY WITH CUBE

```
select AUDITORIUM_TYPE, AUDITORIUM_CAPACITY, COUNT(*) as 'количество'  
from AUDITORIUM  
group by AUDITORIUM_TYPE, AUDITORIUM_CAPACITY with cube .
```

AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	количество
ЛБ-К	15	3
NULL	15	3
ЛК	30	2
NULL	30	2
ЛК	50	1
NULL	50	1
ЛК	60	2
NULL	60	2
ЛБ-К	90	2
ЛК	90	11
ЛК-К	90	1
NULL	90	14
NULL	NULL	22
ЛБ-К	NULL	5
ЛК	NULL	16
ЛК-К	NULL	1

# GROUP BY WITH ROLLUP

```
select AUDITORIUM_TYPE, COUNT(*) as 'количество'  
from AUDITORIUM group by AUDITORIUM_TYPE;
```

AUDITORIUM_TYPE	количество
ЛБ-К	5
ЛК	16
ЛК-К	1

```
select AUDITORIUM_TYPE, COUNT(*) as 'количество'  
from AUDITORIUM group by AUDITORIUM_TYPE with rollup;
```

AUDITORIUM_TYPE	количество
ЛБ-К	5
ЛК	16
ЛК-К	1
NULL	22

# GROUP BY WITH ROLLUP

```
select AUDITORIUM_TYPE, AUDITORIUM_CAPACITY, COUNT(*) as 'количество'  
from AUDITORIUM  
group by AUDITORIUM_TYPE, AUDITORIUM_CAPACITY with rollup
```

AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	количество
ЛБ-К	15	3
ЛБ-К	90	2
ЛБ-К	NULL	5
ЛК	30	2
ЛК	50	1
ЛК	60	2
ЛК	90	11
ЛК	NULL	16
ЛК-К	90	1
ЛК-К	NULL	1
NULL	NULL	22

# INSERT

```
exec SP_HELP [SUBJECT]
```

Column_name	Type	Computed	Length	Prec	Scale	Nullable	Trim Trailing Blanks	Fixed Len Null In Source
SUBJECT	char	no	10			no	no	no
SUBJECT_NAME	varchar	no	100			yes	no	yes
PULPIT	char	no	20			yes	no	yes

```
insert into SUBJECT values ('СУБД', 'Системы управления базами данных', 'ИСиТ');
insert into SUBJECT values ('БД', 'Базы данных', 'ИСиТ'),
                           ('ИНФ', 'Информационные технологии', 'ИСиТ'),
                           ('ОАиП', 'Основы алгоритмизации и программирования', 'ИСиТ'),
                           ('ПЗ', 'Представление знаний в компьютерных системах', 'ИСиТ');
```



# INSERT

```
exec SP_HELP GROUPS
```

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource
IDGROUP	int	no	4	10	0	no	(n/a)	(n/a)
FACULTY	char	no	10			yes	no	yes
PROFESSION	char	no	20			yes	no	yes
YEAR_FIRST	smallint	no	2	5	0	yes	(n/a)	(n/a)
COURSE	int	yes	4	10	0	yes	(n/a)	(n/a)

Identity	Seed	Increment	Not For Replication
IDGROUP	1	1	0

# INSERT

```
INSERT INTO AUDITORIUM (AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY, AUDITORIUM_NAME)
VALUES ('311-4', '%%%', NULL, '311-4');

INSERT INTO AUDITORIUM (AUDITORIUM, AUDITORIUM_TYPE, AUDITORIUM_CAPACITY, AUDITORIUM_NAME)
VALUES ('311-3', '%%%', DEFAULT, '311-3');

SELECT * FROM AUDITORIUM WHERE AUDITORIUM LIKE '311%';
```

%			
Results  Messages			
AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	AUDITORIUM_NAME
311-3	%%%	1	311-3
311-4	%%%	NULL	311-4

# INSERT

```
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]
```

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

```
insert [#Результаты экзамена] ([оценка], [дисциплина], [дата экзамена], [Ф.И.О. студента])
select pr.NOTE, sb.SUBJECT_NAME, pr.PDATE, st.NAME
from STUDENT st join PROGRESS pr on st.IDSTUDENT = pr.IDSTUDENT
      join [SUBJECT] sb on pr.[SUBJECT] = sb.[SUBJECT];
```

# INSERT

```
use BSTU
go
create table [#Журнал]
(
    [Идентификатор] bigint identity (1,1),
    [Операция]       char(8),
    [Дата и время]   datetime2,
    [Ключ]           varchar(100)
)
go
insert into  TEACHER      (TEACHER,    TEACHER_NAME, PULPIT)
output      'INSERT', SYSDATETIME(), inserted.TEACHER into [#Журнал]
values      ('АКНВЧ',      'Акунович Станислав Иванович', 'ИСиТ'),
            ('КЛСНВ',      'Колесников Виталий Леонидович', 'ИСиТ'),
            ('ПРМНН',      'Петров Олег Викторович', 'ИСиТ')
```

# INSERT

```
use BSTU
go
select * from [#Журнал];
```

Идентификатор	Операция	Дата и время	Ключ
1	INSERT	2014-01-12 01:29:30.9693593	АКНВЧ
2	INSERT	2014-01-12 01:29:30.9693593	КПСНВ
3	INSERT	2014-01-12 01:29:30.9693593	ГРМН

# DELETE

```
select count(*) [количество строк в таблице PROGRESS] from PROGRESS;  
delete from PROGRESS;  -- удаление всех строк  
select count(*) [количество строк в таблице PROGRESS] from PROGRESS;
```

количество строк в таблице PROGRESS
34

количество строк в таблице PROGRESS
0

# DELETE

```
select
(select count(*) from PROGRESS) [кол. строк в PROGRESS],
(select count(*)
 from PROGRESS p join STUDENT s on p.IDSTUDENT = s.IDSTUDENT
      join GROUPS g on s.IDGROUP = g.IDGROUP
      and g.FACULTY = 'ИдиП') [кол. строк соединилось]
```

кол. строк в PROGRESS	кол. строк соединилось
34	23

```
select count(*) [количество строк в таблице PROGRESS] from PROGRESS;
delete PROGRESS
from PROGRESS p join STUDENT s on p.IDSTUDENT = s.IDSTUDENT
      join GROUPS g on s.IDGROUP = g.IDGROUP
      and FACULTY = 'ИдиП';
select count(*) [количество строк в таблице PROGRESS] from PROGRESS;
```

количество строк в таблице PROGRESS
34

количество строк в таблице PROGRESS
11

# DELETE

```
select top(0)
    '1234567890' [оператор],
    sysdatetime() [дата и время],
    * into #LOG_PROGRESS from PROGRESS; -- создание временной таблицы

delete PROGRESS
output 'DELETE' [оператор], sysdatetime() [дата удаления], deleted.* into #LOG_PROGRESS
from PROGRESS p join STUDENT s on p.IDSTUDENT = s.IDSTUDENT
    join GROUPS g on s.IDGROUP = g.IDGROUP
where g.FACULTY = 'ИЭФ';
select * from #LOG_PROGRESS;
```

оператор	дата и время	SUBJECT	IDSTUDENT	PDATE	NOTE
DELETE	2014-02-16 23:33:46.6569140	ЭТ	1155	2013-01-15	7
DELETE	2014-02-16 23:33:46.6569140	ЭТ	1156	2013-01-15	8
DELETE	2014-02-16 23:33:46.6569140	ЭТ	1157	2013-01-15	9
DELETE	2014-02-16 23:33:46.6569140	ЭТ	1158	2013-01-15	4
DELETE	2014-02-16 23:33:46.6569140	ЭТ	1159	2013-01-15	5



# TRUNCATE

```
select count(*) [кол. строк в TEACHER] from TEACHER;  
truncate table TEACHER;  
select count(*) [кол. строк в TEACHER] from TEACHER;
```

кол. строк в TEACHER
39

кол. строк в TEACHER
0

Запись в журнал транзакций:

DELETE

TRUNCATE

# UPDATE

```
select top(5) * from AUDITORIUM;  
update AUDITORIUM set AUDITORIUM_CAPACITY = 100;  
select top(5) * from AUDITORIUM;
```

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	AUDITORIUM_NAME
02Б-4	ЛК	90	02Б-4
103-4	ЛК	90	103-4
105-4	ЛК	90	105-4
107-4	ЛК	90	107-4
110-4	ЛК	30	110-4

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	AUDITORIUM_NAME
02Б-4	ЛК	100	02Б-4
103-4	ЛК	100	103-4
105-4	ЛК	100	105-4
107-4	ЛК	100	107-4
110-4	ЛК	100	110-4

# UPDATE

Операция	Назначение
<b>=</b>	значение
<b>= default</b>	значение по умолчанию
<b>+=</b>	сумма (конкатенация для символьных данных)
<b>-=</b>	разница
<b>*=</b>	произведение
<b>/=</b>	частное
<b>%=</b>	остаток от деления
<b>&amp;=</b>	результат побитового логического AND
<b>^=</b>	результат побитового логического XOR
<b> =</b>	результат побитового логического OR

# UPDATE

```
select top(3) * from AUDITORIUM;  
update AUDITORIUM set AUDITORIUM_CAPACITY *= 1.2,  
                      AUDITORIUM_NAME += ' +20%';  
select top(3) * from AUDITORIUM;
```

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	AUDITORIUM_NAME
02Б-4	ЛК	90	02Б-4
103-4	ЛК	90	103-4
105-4	ЛК	90	105-4

AUDITORIUM	AUDITORIUM_TYPE	AUDITORIUM_CAPACITY	AUDITORIUM_NAME
02Б-4	ЛК	108	02Б-4 +20%
103-4	ЛК	108	103-4 +20%
105-4	ЛК	108	105-4 +20%

# UPDATE

```
select * from TEACHER
update TEACHER set  TEACHER_NAME += (' '+GENDER),
                    PULPIT = default, GENDER = default;
select * from TEACHER;
```

TEACHER	TEACHER_NAME	GENDER	PULPIT
ГРН	Гурин Николай Иванович	м	ИСиТ
ДЛН	Долинина Татьяна Николаевна	ж	СБУАиА
ДМДК	Демидко Марина Николаевна	м	ЛПиСПС
ДТК	Дятко Александр Аркадьевич	м	ИСиТ
ЕЩНК	Ещенко Людмила Семеновна	ж	ТНВиО...
ЖЛК	Жиляк Надежда Александровна	ж	ИСиТ

TEACHER	TEACHER_NAME	GENDER	PULPIT
ГРН	Гурин Николай Иванович м	м	NULL
ДЛН	Долинина Татьяна Николаевна ж	м	NULL
ДМДК	Демидко Марина Николаевна м	м	NULL
ДТК	Дятко Александр Аркадьевич м	м	NULL
ЕЩНК	Ещенко Людмила Семеновна ж	м	NULL
ЖЛК	Жиляк Надежда Александровна ж	м	NULL

# UPDATE

```
select distinct pr.IDSTUDENT, pr.[SUBJECT], pr.PDATE
into [#Механики_с_оценкой_<_10]
from STUDENT s join GROUPS g      on s.IDGROUP = g.IDGROUP
      join PROGRESS pr on pr.IDSTUDENT = s.IDSTUDENT
      join PROFESSION p on p.PROFESSION = g.PROFESSION
where p.QUALIFICATION like '%механик%' and pr.NOTE < 10;
```

```
select t.IDSTUDENT, t.[SUBJECT], t.PDATE, pr.NOTE
from PROGRESS pr join [#Механики_с_оценкой_<_10] t
      on pr.IDSTUDENT = t.IDSTUDENT and
      pr.[SUBJECT] = t.[SUBJECT] and
      pr.PDATE = t.PDATE
```

IDSTUDENT	SUBJECT	PDATE	NOTE
1064	OX	2013-01-19	6
1065	OX	2013-01-19	4
1066	OX	2013-01-19	9
1067	OX	2013-01-19	5
1068	OX	2013-01-19	8
1069	OX	2013-01-19	4

# UPDATE

```
update PROGRESS set NOTE+=1
from PROGRESS pr join [#Механики_с_оценкой_<_10] t
    on pr.IDSTUDENT = t.IDSTUDENT and
       pr.[SUBJECT] = t.[SUBJECT] and
       pr.PDATE = t.PDATE;
```

```
select t.IDSTUDENT, t.[SUBJECT], t.PDATE, pr.NOTE
from PROGRESS pr join [#Механики_с_оценкой_<_10] t
    on pr.IDSTUDENT = t.IDSTUDENT and
       pr.[SUBJECT] = t.[SUBJECT] and
       pr.PDATE = t.PDATE;
```

IDSTUDENT	SUBJECT	PDATE	NOTE
1064	OX	2013-01-19	7
1065	OX	2013-01-19	5
1066	OX	2013-01-19	10
1067	OX	2013-01-19	6
1068	OX	2013-01-19	9
1069	OX	2013-01-19	5

# UPDATE

```
-
select top(0) 'XXXXXXX' [операция],
              AUDITORIUM [аудитория] ,
              AUDITORIUM_CAPACITY [вместимость до],
              AUDITORIUM_CAPACITY [вместимость после]
into [#Регистрация изменений] -- создать таблицу
from AUDITORIUM;

update AUDITORIUM set AUDITORIUM_CAPACITY*=1.2
output 'UPDATE' [операция], inserted.AUDITORIUM [аудитория],
       deleted.AUDITORIUM_CAPACITY [до],
       inserted.AUDITORIUM_CAPACITY [после] into [#Регистрация изменений]
where AUDITORIUM_TYPE = 'ЛБ-К';
select * from [#Регистрация изменений];
```

операция	аудитория	вместимость до	вместимость после
UPDATE	206-1	15	18
UPDATE	301-1	15	18
UPDATE	304-4	90	108
UPDATE	413-1	15	18
UPDATE	423-1	90	108



# Вопросы?