

# БАЗЫ ДАННЫХ

---

Лекция 4

Архитектура базы данных Microsoft SQL Server

# Возможности СУБД

- разнообразные пользовательские интерфейсы
- физическая независимость данных
- логическая независимость данных
- оптимизация запросов
- целостность данных
- управление параллелизмом
- резервное копирование и восстановление
- безопасность баз данных

# Системные требования

- Память и дисковое пространство (min 512 Мб)
- ОС Windows 32 или 64 разряда
- Требования к сети

# Версии SQL Server

- Express Edition
- Workgroup Edition
- Standard Edition
- Web Edition
- Enterprise Edition
- Developer Edition
- Datacenter Edition
- Parallel Data Warehouse Edition

# Версии SQL Server

- Express Edition
  - Облегченная версия SQL Server
  - Предназначена для разработчиков приложений
  - По этой причине продукт содержит базовую программу
  - Поддерживает интеграцию общеязыковой среды выполнения CLR и собственный язык XML
  - SQL Server Management Express для SQL Server Express, <http://msdn.microsoft.com/express>

# Версии SQL Server

- Workgroup Edition
  - Предназначена для малого бизнеса и для использования на уровне отделов предприятия
  - Отсутствуют средства бизнес-аналитики (БА) и возможности обеспечения высокого уровня доступности
  - Поддерживает системы с двумя процессорами и максимум 2 Гб оперативной памяти

# Версии SQL Server

- Standard Edition
  - Версия предназначена для малого и среднего бизнеса
  - Содержит весь диапазон возможностей бизнес-аналитики, включая службы Analysis Services, Reporting Services и Integration Services.
  - Не содержит многих возможностей из версии Enterprise Edition
  - Поддерживает до 4х процессоров и 2 Тб оперативной памяти

# Версии SQL Server

- Web Edition
  - Версия предназначена для поставщиков веб-хостинга
  - Содержит службы отчетности Reporting Services
  - Поддерживается до 4х процессоров и 2 Тб оперативной памяти



# Версии SQL Server

- Enterprise Edition

- Предназначена для приложений, критичных по времени и с большим количеством пользователей
- Поддерживает секционирование данных, возможность получения мгновенных снимков состояния базы данных и онлайн-поддержку баз данных

# Версии SQL Server

- Developer Edition

- Содержит всю функциональность версии Enterprise Edition
- Лицензия разрешает использование только для разработки, тестирования и демонстрации на необходимом количестве систем
- Для использования другими разработчиками необходимо приобрести дополнительные лицензии

# Версии SQL Server

- Datacenter Edition

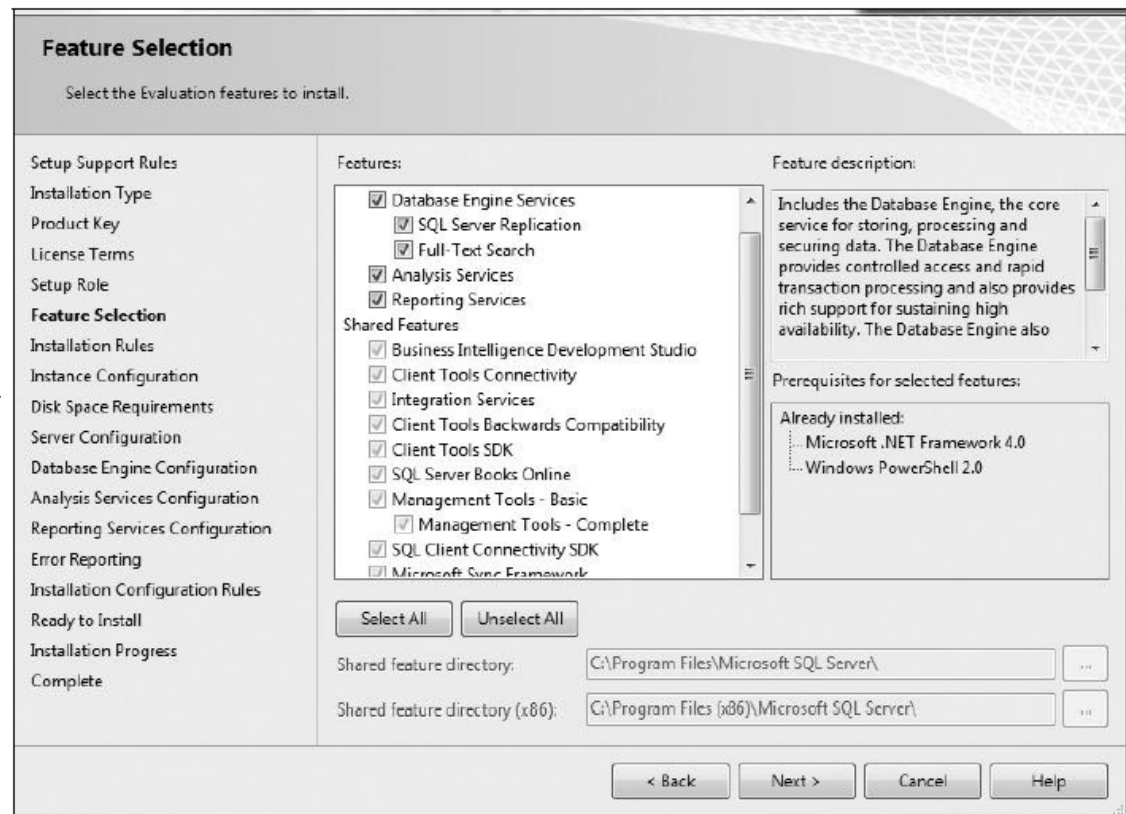
- Предназначена для поддержки масштабирования наивысшего уровня
- Нет ограничений по памяти
- Можно создавать до 25 экземпляров сервера
- Обеспечивается поддержка до 256 логических процессоров

# Версии SQL Server

- Parallel Data Warehouse Edition
  - Версия специализирована для хранилищ данных и поддерживает базы данных хранилищ данных размером от 10 Тб до 1 Пб (петабайт, 1 Пб = 1024 Тб)
  - Используется архитектура массово-параллельной обработки (МПО с возможностями высокопроизводительных вычислений HPC (High Performance Computing))

# Компоненты SQL Server

- Database Engine
- Analysis Services
- Reporting Services
- Integration Services
- Full-Text Search
- SQL Server Agent
- SQL Server Browser



# Экземпляр сервера БД

- по умолчанию (default)
- именованные (named)

# Режим проверки подлинности

- Режим проверки подлинности Windows
- Режим проверки подлинности SQL Server

# Базы данных

- Системные
- Пользовательские



# Системные БД

- Master
  - информация системного уровня: параметры настройки, о других базах данных и их файлах, учетные записи и др.
- Msdb
  - для планирования работы заданий
- Tempdb
  - временные объекты (таблицы, процедуры, курсоры)
- Model
  - шаблон базы данных
- Resource
  - содержит системные объекты, которые входят в состав SQL Server

# Системные БД

- создаются при инсталляции сервера
- используются сервером в процессе работы

# Хранение объектов баз данных

- страницы
- экстененты
- файлы
- файловые группы

# Хранение объектов баз данных

- Основной единицей хранилища данных является страница
- Размер страницы постоянен и составляет 8 Кб
- Каждая страница имеет заголовок размером в 96 байтов для системная информации
- Строки данных размещаются на странице сразу же после заголовка
- Виды страниц:
  - страницы данных
  - страницы индексов

# Хранение объектов баз данных

- Содержимое базы данных хранится в одном или нескольких файлах
- Каждый файл разделен на несколько страниц
- Каждую страницу таблицы или индекса можно однозначно идентифицировать, используя
  - идентификатор базы данных
  - идентификатор файла базы данных
  - номер страницы

# Хранение объектов баз данных

- Физическая единица дискового пространства, используемая для выделения памяти для таблиц и индексов, называется экстентом
- Размер экстента составляет 8 последовательно расположенных страниц или иначе 64 Кбайт

# Хранение объектов баз данных

- Все страницы данных имеют фиксированный размер (8 Кб) и состоят из следующих частей:
  - заголовка страницы (96 байтов)
  - пространства для данных
  - таблицы смещений строк

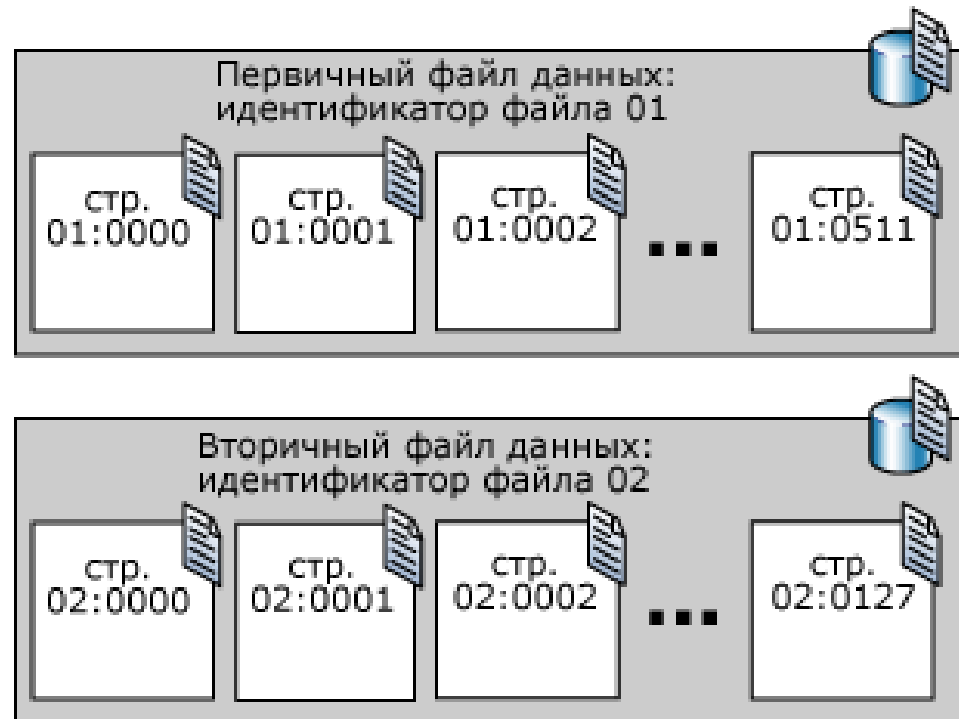
# Хранение объектов баз данных





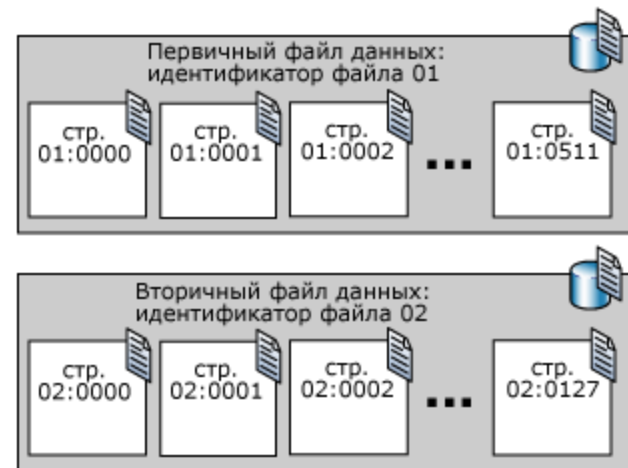
# Хранение данных на диске

- страницы
- экстенты
- файлы
- файловые группы



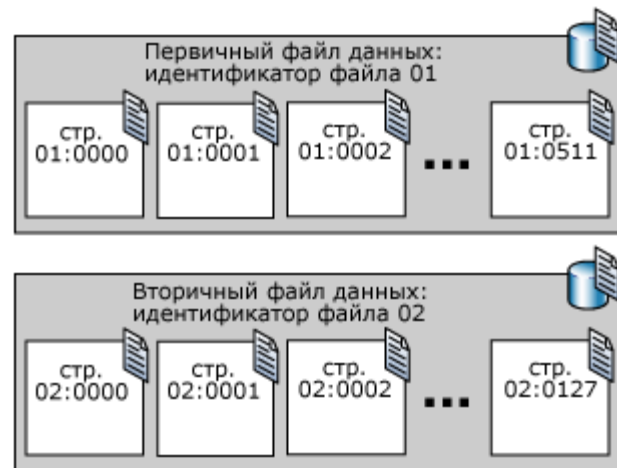
# Страницы

- Основной единицей хранилища данных является страница.
- Размер страницы постоянен и составляет 8 Кбайт.
- Каждая страница имеет заголовок 96 байтов, в котором хранится системная информация.
- Строки данных размещаются на странице сразу же после заголовка.
- Виды страниц:
  - страницы данных;
  - страницы индексов.



# Экстенты

- Физическая единица дискового пространства, используемая для выделения памяти для таблиц и индексов, называется экстендом.
- Размер экстента составляет восемь последовательно расположенных страниц или 64 Кбайт.
- Существует два следующих типа экстентов:
  - однородные экстенты;
  - смешанные экстенты.



# Экстенты

- Однородные экстенты содержат данные одной таблицы или индекса
- Смешанные экстенты могут содержать данные до восьми таблиц или индексов.



# Свойства страниц данных

- Все типы страниц данных имеют фиксированный размер (8 Кбайт) и состоят из следующих частей:
- заголовка страницы;
- пространства для данных;
- таблицы смещений строк.



# Пространство для данных

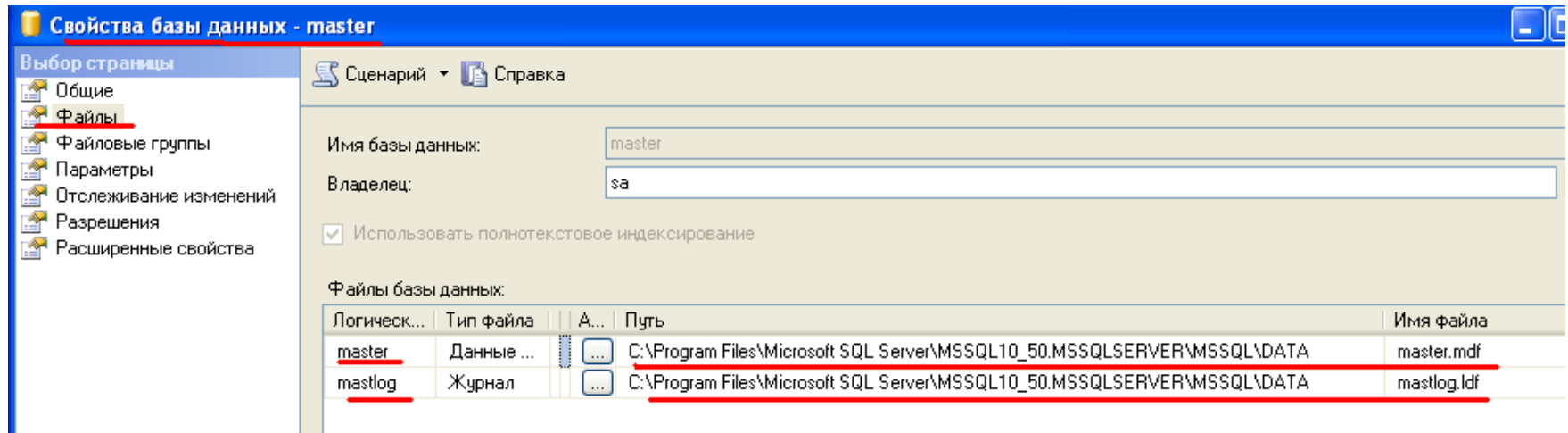
- Для каждой строки страницы создается
  - запись данных
  - запись в таблице смещения
- Строка данных не может выходить за пределы одной страницы
- Строки сохраняются в последовательном порядке

# Пространство для данных

- Если в странице нет места для новой строки той же таблицы, эта строка сохраняется в следующей странице цепочки страниц.
- Для таблиц, которые имеют только столбцы фиксированного размера, в каждой странице сохраняется одинаковое количество строк.
- Если в таблице есть хотя бы один столбец переменной длины, количество строк в странице может колебаться, и в странице сохраняется столько строк, сколько поместится.

# Файлы базы данных

- первичный файл (.mdf)
- вторичные файлы (.ndf)
- файлы журнала транзакций (.log)





# Файловые группы

- Файловые группы – поименованный набор файлов БД
- Все файлы БД, кроме файлов журнала транзакций, распределены по файловым группам
  - Первичные
  - Вторичные
- Используются для упрощения администрирования
  - Секционирование
  - Резервное копирование и восстановление

# Создание базы данных

- CREATE database BSTU
- ON PRIMARY -- первичная файловая группа
- (name = N'BSTU\_mdf', -- логическое имя файла
- filename = N'D:\BD\BSTU\_mdf.mdf', -- имя файла в ОС
- size = 10240Kb, -- первоначальный размер файла
- maxsize = UNLIMITED, -- максимальный размер файла
- filegrowth = 1024Kb) -- приращение
- LOG ON -- журнал транзакций
- ( name = N'BSTU\_log', -- логическое имя файла
- filename = N'D:\BD\BSTU\_log.ldf', -- физический файл
- size = 10240Kb, -- первоначальный размер
- maxsize = 2048Gb, -- максимальный размер
- filegrowth = 10%) -- приращение
-

# Журнал транзакций

- Сначала запись идет в журнал транзакций, потом в файл данных
- Файл журнала транзакций используется для восстановления данных БД в случае аварийного завершения работы сервера

# Файловые группы

- Во вторичных файловых группах могут располагаться только вторичные файлы.
- В первичной файловой группе помимо обязательного первичного файла тоже могут быть расположены вторичные файлы.
- При создании таблиц и индексов дисковая память для них автоматически отводится в файловой группе по умолчанию.
- Для размещения в другой файловой группе следует явно указывать ее имя в операторе CREATE, создающем таблицу или индекс.

# Пример

- CREATE database BSTU **ON PRIMARY**
- (name = N'BSTU\_1', filename = N'D:\BD\BSTU\_1.mdf',
- (size = 10240Kb, maxsize = UNLIMITED, filegrowth = 1024Kb),
- (name = N'BSTU\_2', filename = N'D:\BD\BSTU\_2.ndf', size = 10240KB, maxsize = 1Gb, filegrowth = 25%),
- **FILEGROUP FG1**
- (name = N'BSTU\_fg1\_1', filename = N'D:\BD\BSTU\_3.ndf',
- size = 10240Kb, maxsize = 1Gb, filegrowth = 25%),
- (name = N'BSTU\_fg1\_2', filename = N'D:\BD\BSTU\_4.ndf',
- size = 10240Kb, maxsize = 1Gb, filegrowth = 25%)
- **LOG ON**
- (name = N'BSTU\_log', filename = N'D:\BD\BSTU\_log.ldf',
- size = 10240Kb, maxsize = 2048Gb, filegrowth = 10%)

# Изменение файловых групп

- alter database BSTU add filegroup G1;
- alter database BSTU add file
- (name = N'BSTU1', filename = N'C:\BSTU1.ndf',
- size = 3072KB, maxsize = unlimited, filegrowth = 1024KB
- ) to filegroup G1;
- alter database BSTU modify filegroup G1 default;

# Параметры БД

- параметры автоматических действий,
- параметры курсора,
- параметры внешних действий,
- параметры восстановления,
- параметры SQL,
- параметры моментальных снимков,
- параметры компонента Service Broker,
- параметры доступности БД.

# Параметры БД

- **ALTER DATABASE ... SET .....**

```
use MASTER
```

```
go
```

```
alter database BSTU set AUTO_CLOSE off          -- автоматическое закрытие БД
```

```
alter database BSTU set AUTO_CREATE_STATISTICS on -- автоматический сбор статистики
```

```
alter database BSTU set AUTO_SHRINK on           -- автоматическое сжатие файлов
```

```
go
```



# Параметры БД

- `AUTO_CLOSE` – закрытие БД при отсутствии с ней соединений.
- `AUTO_CREATE_STATISTICS` – включить или отключить автоматический сбор статистики для оптимизатора запросов
- `AUTO_SHRINK` – необходимость периодического сжатия файлов БД

# Параметры БД

```
use MASTER
go

select  case DATABASEPROPERTYEX('BSTU', 'IsAutoClose')
        when 0 then 'off'          -- если = 0, то вернуть 'off'
        else 'on'                  -- иначе вернуть 'on'
        end  IsAutoClose,
        case DATABASEPROPERTYEX('BSTU', 'IsAutoShrink')
        when 0 then 'off'
        else 'on'
        end  IsAutoShrink,
        case DATABASEPROPERTYEX('BSTU', 'IsAutoCreateStatistics')
        when 0 then 'off'
        else 'on'
        end  IsAutoCreateStatistics
go
```

IsAutoClose	IsAutoShrink	IsAutoCreateStatistics
off	on	on

# Принципы проектирования логической схемы базы данных

Уровень представления	Пользователи
Логический уровень	Проектировщики
Физический уровень	Системные администраторы

# Проектирование логической схемы базы данных

- **Системный администратор:** БД – это набор файлов операционной системы, имеющих определенную физическую структуру, формат и организацию.
- **Проектировщик:** БД – это **логическая схема данных**, представляющая собой набор структурированных данных с заданными ограничениями и связями.
- **Пользователь:** БД представлена в виде подмножества **таблиц** или **представлений**, набора **процедур** и функций на языке **T-SQL** и пр.

# Проектирование логической схемы базы данных

- Теория реляционных БД предполагает независимость трех уровней друг от друга.
- Физическое размещение данных никак не должно ограничивать возможности проектирования логической схемы данных.
- Представления пользователей о БД не зависят от структуры логической схемы.
- В реальности – частичная независимость уровней друг от друга.

# Размещение таблиц в файловой

группе

```
create table AUDITORIUM -- аудитории вуза
(
    AUDITORIUM          char(20)    -- идентификатор аудитории (304-1, 401-4,...)
                        constraint AUDITORIUM_PK primary key,
    AUDITORIUM_TYPE     char(10)    -- тип аудитории (ЛК, ЛБ-К, ...)
                        constraint AUDITORIUM_AUDITORIUM_TYPE_FK foreign key
                        references AUDITORIUM_TYPE(AUDITORIUM_TYPE),
    AUDITORIUM_CAPACITY int         -- вместимость (макс. кол. слушателей)
                        constraint AUDITORIUM_CAPACITY_CHECK default 1
                        check (AUDITORIUM_CAPACITY between 1 and 300),
    AUDITORIUM_NAME     varchar(50) -- текстовое наименование аудитории (комментарий
) on FG1; -- в файловой группе FG1
```

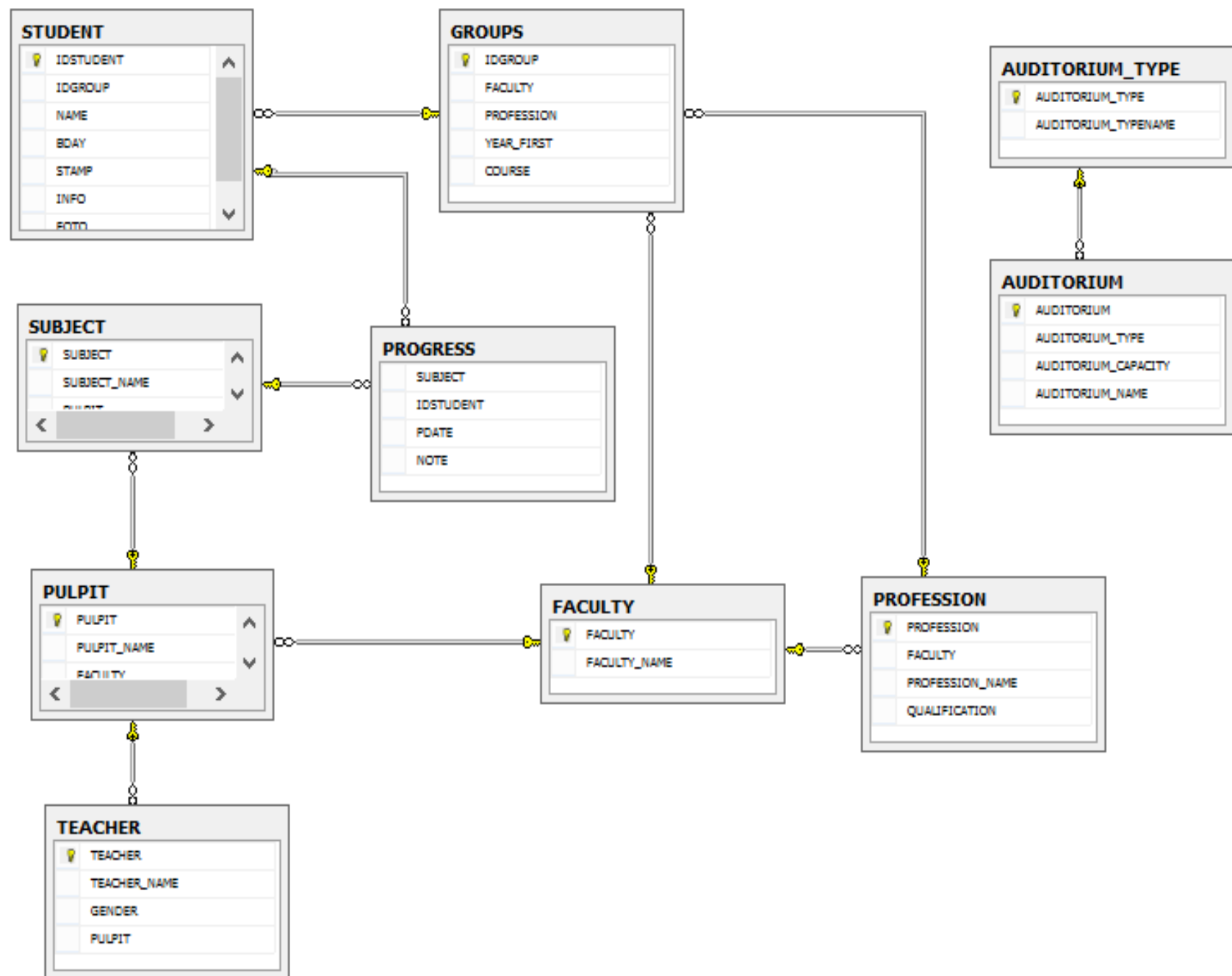
# Размещение таблиц в файловой группе

```
create table SHEDULE_TEACHER    -- расписание преподавателя
(
    [DATETIME]    smalldatetime,    -- дата и время занятий
    TEACHER        char(10),        -- преподаватель
    [SUBJECT]      char(10),        -- дисциплина
    AUDITORIUM     char(10),        -- аудитория
    constraint PK_SHEDULE_TEACHER primary key([DATETIME],TEACHER)
) on FG2
```

# Удаление таблиц

- **DROP TABLE ...**
- 
- Проблема удаления таблицы:
  - Пользователь не имеет достаточных прав на удаление
  - Таблица заблокирована транзакцией другого сеанса
  - На первичный ключ удаляемой таблицы ссылается внешний ключ другой таблицы





# Изменение структуры таблицы

- Добавление или удаление столбцов
- Добавление или удаление ограничений
- Изменение точности или типа числовых данных, размерности символьных данных, ...

# Просмотр структуры таблицы

```
use BSTU
```

```
go
```

```
exec SP_HELP TEACHER;
```

# Просмотр структуры таблицы

Name	Owner	Type	Created_datetime
TEACHER	dbo	usertable	2014-01-09 15:24:54.093

Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLen/NullInSource
TEACHER	char	no	10			no	no	no
TEACHER_NAME	varchar	no	100			yes	no	yes
GENDER	char	no	1			yes	no	yes
PULPIT	char	no	20			yes	no	yes

constraint_type	constraint_name	delete_action	update_action	status_enabled	status_for_replication	constraint_keys
CHECK on column GENDER	TEACHER_GENDER_CH	(n/a)	(n/a)	Enabled	Is_For_Replication	(([GENDER]='x' OR [GENDER]='m')
PRIMARY KEY (clustered)	TEACHER_PK	(n/a)	(n/a)	(n/a)	(n/a)	TEACHER
FOREIGN KEY	TEACHER_PULPIT_FK	No Action	No Action	Enabled	Is_For_Replication	PULPIT
						REFERENCES BSTU.dbo.PULPI...

# Изменение структуры таблицы

```
use BSTU
go
select *, len(TEACHER_NAME) 'len(TEACHER_NAME)' from TEACHER order by TEACHER
```

TEACHER	TEACHER_NAME	GENDER	PULPIT	len(TEACHER_NAME)
AKHB4	Акунович Станислав Иванович	NULL	ИСиТ	27
APC	Арсентьев Виталий Арсентьевич	NULL	ПОиСОИ	29
ATP	Атрощенко Олег Александрович	NULL	ЛУ	28
БЗБРДВ	Безбородов Владимир Степанович	NULL	ОХ	30
БПР	Белов Валерий Павлович	NULL	ПОиСОИ	23

```
use BSTU
go
alter table TEACHER -- изменение столбца
    alter column TEACHER_NAME varchar(50) not null;
alter table TEACHER -- добавление нового столбца
    add BDATE date not null default '19600101'
```

# Изменение структуры таблицы

```
use BSTU
go
select *, len(TEACHER_NAME) 'len(TEACHER_NAME)' from TEACHER order by TEACHER
```

TEACHER	TEACHER_NAME	GENDER	PULPIT	BDATE	len(TEACHER_NAME)
AKHB4	Акунович Станислав Иванович	NULL	ИСиТ	1960-01-01	27
APC	Арсентьев Виталий Арсентьевич	NULL	ПОиСОИ	1960-01-01	29
ATP	Атрощенко Олег Александрович	NULL	ЛУ	1960-01-01	28

# Системный каталог

- Системный каталог состоит из таблиц, описывающих структуру объектов базы данных
- К системным базовым таблицам нельзя обращаться напрямую



# Представления каталога

- sys.objects
- sys.columns
- sys.database\_principals

```
USE sample;  
SELECT object_id, principal_id, type  
FROM sys.objects  
WHERE name = 'employee';
```



# Динамические административные представления и функции

- Возвращают информацию о состоянии сервера, которую можно применить для отслеживания и настройки производительности системы

◆ `sys.dm_db_*` —

◆ `sys.dm_tran_*` —

◆ `sys.dm_io_*` —

◆ `sys.dm_exec_*` —

# Информационная схема

- Состоит из представлений, доступных только для чтения, которые предоставляют информацию обо всех таблицах, представлениях и столбцах компонента Database Engine, к которым имеется доступ.
  - `information_schema.tables`
  - `information_schema.columns`

# Системные хранимые процедуры

- Системные хранимые процедуры применяются для выполнения административных и пользовательских задач:
  - переименование объектов базы данных,
  - идентификация пользователей
  - мониторинг авторизации и ресурсов
- 
- `sp_help`
  - `sp_configure`

# Системные функции

- OBJECT\_ID(object\_name)
- OBJECT\_NAME(object\_id)
- USER\_ID([user\_name])
- USER\_NAME([user\_id])
- DB\_ID([db\_name])
- DB\_NAME([db\_id])

# Функции свойств

- Функции свойств возвращают свойства объектов баз данных, типов данных и файлов.
  - Функции свойств поддерживают десятки свойств в виде параметров, которые можно явно указывать.
  - Почти все функции свойств возвращают одно из следующих значений: 0, 1 или NULL.
- 
- OBJECTPROPERTY(id, property)
  - COLUMNPROPERTY(id, column, property)
  - FILEPROPERTY(filename, property)
  - TYPEPROPERTY(type, property)

# Вопросы?