

# Администрирование информационных систем

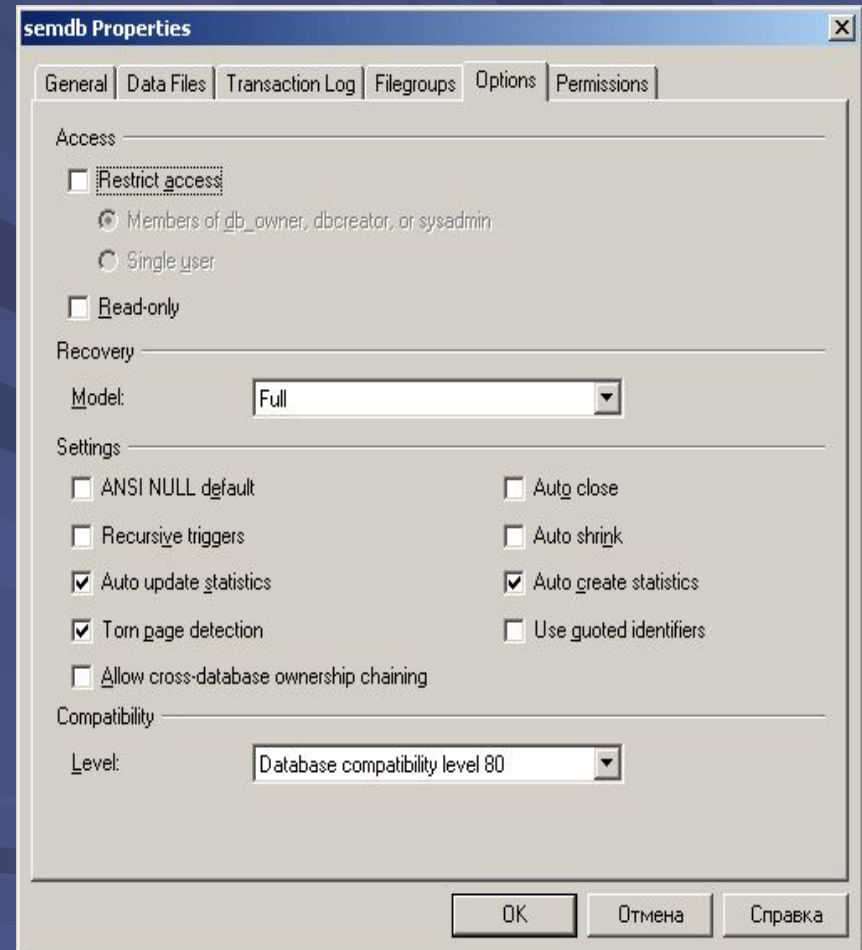
Администрирование БД  
лекция 5

# Настройка параметров БД

- Параметры БД определяют различные аспекты работы с системными или пользовательскими базами данных.
- Параметра пользовательских БД при создании наследуют параметры системной БД **model**.
- Выделяют следующие типы параметров:
  - управление автоматическим поведением БД (такие как автоматическое создание и обновление данных статистики);
  - выбор модели восстановления;
  - совместимость с ANSI;
  - режим доступа к БД (например только для чтения и доступа, предоставленный только для владельца БД – **dbo**).

# Просмотр значений параметров БД

- Просмотр установленных значения параметров БД можно с помощью:
  - SQL Server Enterprise Manager (рисунок справа)
  - или системной функции **DATABASEPROPERTYEX**
  - Например, для определения модели восстановления в БД semdb необходимо выполнить запрос:
    - `SELECT DATABASEPROPERTYEX('SEMDB','RECOVERY')`



# Изменение значений параметров БД

- Изменение значений параметров пользовательской БД выполняется либо путем изменения параметров в SQL Server Enterprise Manager, либо путем выполнения оператора Transact-SQL **ALTER DATABASE**.
- Например, для изменения модели восстановления на модель восстановления результатов импорта необходимо выполнить запрос:
  - **ALTER DATABASE SEMBD SET RECOVERY BULK\_LOGGED**

# Изменение размера базы данных

- После создания БД возникает необходимость в периодическом изменении ее размеров.
- При увеличении размеров БД возрастает нагрузка на систему, журналы транзакций увеличиваются быстрее.
- SQL Server поддерживает ряд механизмов управления изменениями БД: автоматические методы контроля размера БД, ручного изменения размеров файлов БД и журналов транзакций.

# Автоматическое увеличение размера файлов

- При создании пользовательской БД по умолчанию выбирается автоматическое увеличение размеров файлов данных и журналов транзакций.
- Однако при каждом увеличении файлов нагрузка на систему возрастает. Кроме того, увеличение дискового пространства для хранения файлов приводит к дефрагментации диска.
- Все это обуславливает необходимость контроля за объемом данных и журнала транзакций и отказ от модели автоматического увеличения размеров файлов данных и журнала транзакций.

# Изменение параметров автоматического увеличения размера файлов

- Изменение параметров автоматического увеличения размеров файлов данных и журнала транзакций выполняется с помощью графических средств SQL Server Enterprise Manager.
- Другой вариант – использование операторов Transact-SQL. Для отключения, например, автоматического увеличения файла данных БД semdb, необходимо выполнить запрос:
  - **ALTER DATABASE SEMDB  
MODIFY FILE ( NAME = 'semdb\_data', FILEGROWTH = 0 )**

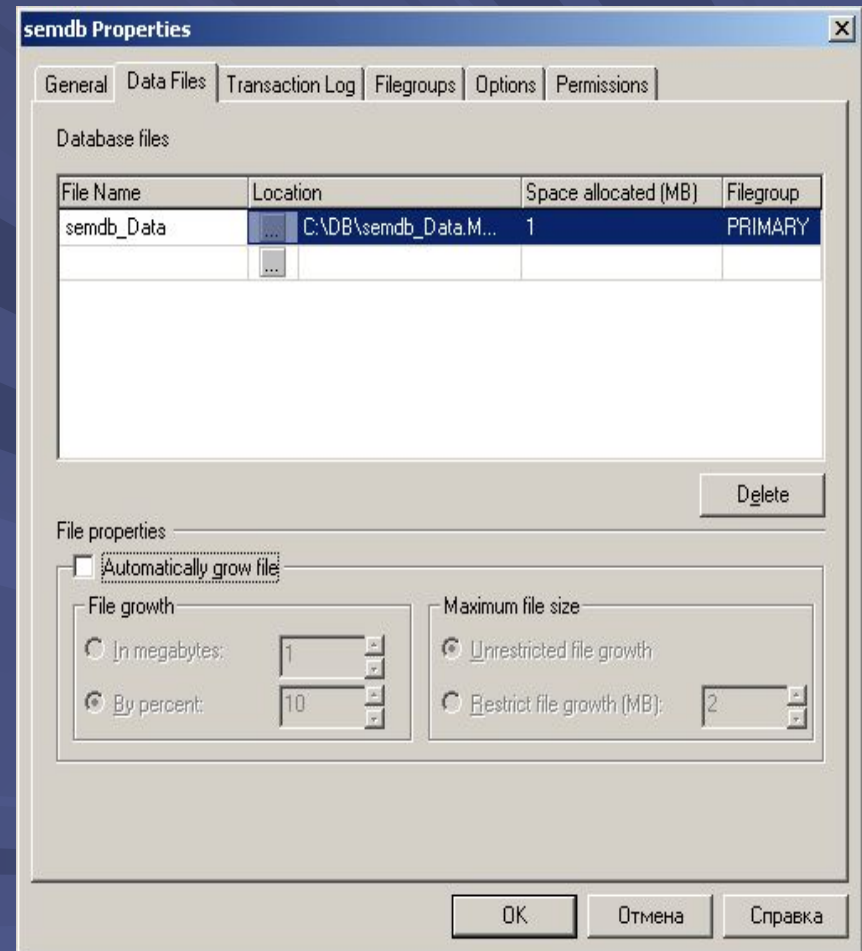
# Автоматическое уменьшение размеров файлов

- Размер файлов БД может быть автоматически не только увеличен, но и уменьшен.
- Такой режим может быть установлен, если файл данных или журнала транзакций имеет много свободного места.
- Однако режим автоматического уменьшения размера файлов не рекомендуется к использованию, поскольку это уменьшает производительность системы.



# Изменение размера файлов данных вручную

- Отслеживая изменения свободного пространства в файлах данных, администратор имеет возможность оценить необходимость в изменении размеров файлов. Такие операции выполняются во время снижения нагрузки на систему.
- Для увеличения размера файлов можно воспользоваться средствами SQL Server Enterprise Manager, выбрав соответствующие закладки в контекстном меню соответствующей БД.



# Изменение размера файлов данных вручную

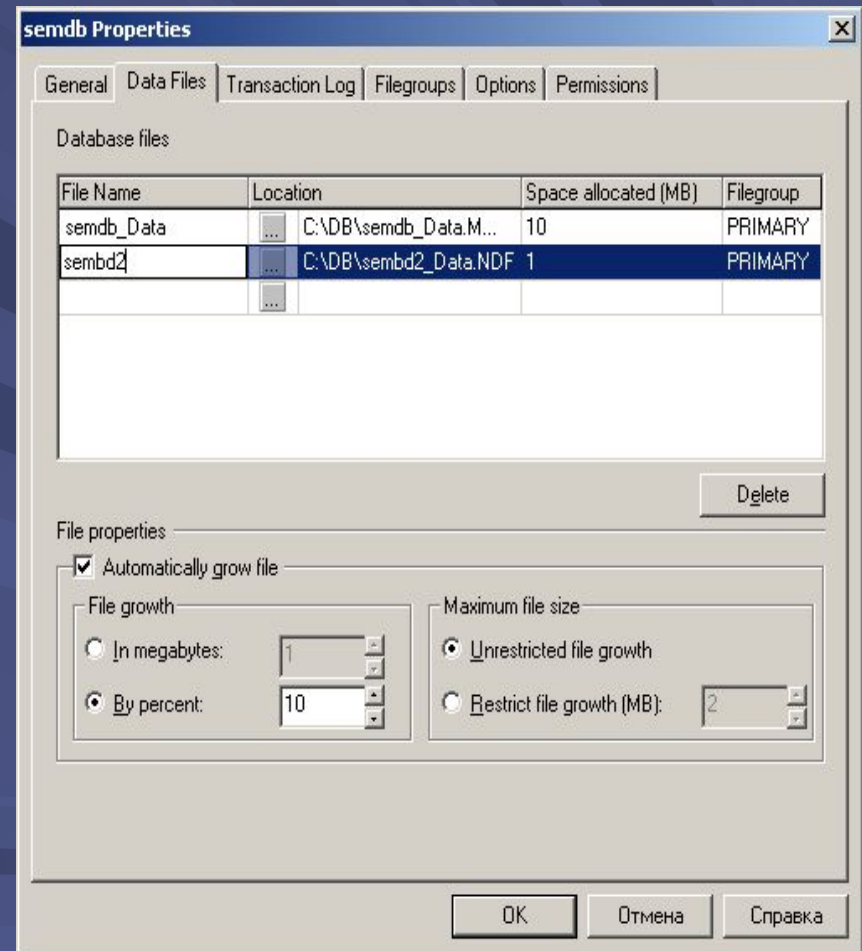
- При использовании средств Transact-SQL необходимо выполнить соответствующий запрос.
- Для установления нового размера файла данных в БД semdb размером 20Мб необходимо выполнить запрос:
  - **ALTER DATABASE SEMDB  
MODIFY FILE ( NAME = 'semdb\_data', SIZE = 25 )**
- Для уменьшения размера файла данных вручную используется оператор DBCC SHRINKFILE:
  - **USE SEMDB DBCC SHRINKFILE ( 'semdb\_data', 10 )**
  - Данный запрос уменьшит размер файла данных в БД semdb до 10 Мб.
- При выполнении оператора DBCC SHRINKFILE операция производится над текущей БД. Во время уменьшения размера файл БД уменьшается с конца. По умолчанию все использованные страницы в сокращаемой части файла данных перемещаются на свободное пространство в начале файла. Файл не может быть уменьшен до размера меньшего, чем объем данных или размера БД model.

# Изменение размера файла журнала транзакций

- Для работы БД важно наличие свободного пространства в журнале транзакций.
- При использовании модели восстановления отдельных операций или модели восстановления результатов копирования необходимо выполнение регулярного резервного копирования файлов журнала транзакций.
- Если журналы транзакций заполняются слишком быстро необходимо либо чаще копировать журналы, либо позволить SQL Server автоматически увеличивать размер журнала.
- Изменение размеров файла журнала транзакций БД выполняется подобно изменению размеров файлов данных.

# Создание дополнительных файлов

- При нехватке места на одном диске прибегают к созданию дополнительных файлов данных и журналов транзакций, расположенных на других дисках.
- При создании дополнительных файлов БД можно воспользоваться SQL Server Enterprise Manager.
- По умолчанию все дополнительные файлы данных добавляются в основную группу файлов.



# Создание дополнительных файлов

- Создание дополнительных файлов возможно с помощью операторов Transact-SQL. Для добавления дополнительного файла данных в БД SEMDB необходимо задать:
  - **ALTER DATABASE SEMDB**  
**ADD FILE (NAME = 'SEMDB2', FILENAME = 'C:\DB\SEMDB2.ndf', SIZE = 10 , MAXSIZE = 25, FILEGROWTH = 5)**

# Конфигурирование дисковой подсистемы

- Для обеспечения максимальной производительности, отказоустойчивости и минимизации времени восстановления данных при сбое, необходимо разработать стратегию размещения файлов данных, журналов транзакций и БД tempdb.
- При выборе размещения журнала транзакций необходимо учитывать основное назначение журналов – обеспечение восстановления данных в случае отказа дисков с размещенными на них файлами данных. Отсюда следует основное условие – размещаются файлы транзакций, как правило, на дисках отличных от дисков с файлами данных БД.
- Следующий шаг оптимизации производительности – размещение журналов транзакций на отдельных дисках для каждой БД.
- Еще один способ оптимизации производительности – использование системы RAID 1 (зеркальное хранение данных).

# Конфигурирование дисковой подсистемы

- При выборе дисковой системы для хранения файлов данных основные условия – обеспечение недопустимости потери данных и минимизация времени простоя.
- Пути решения – размещения файлов данных и журналов транзакций на разных дисках, размещение файлов данных на разных дисках (особенно для больших БД).
- Использование RAID позволяет увеличить производительность системы.
- Если в системе требуется высокая производительность операций чтения, то рекомендуется использование RAID 5.
- Если требуется высокая производительность операций записи, то рекомендуется использование RAID 0, или RAID 10.

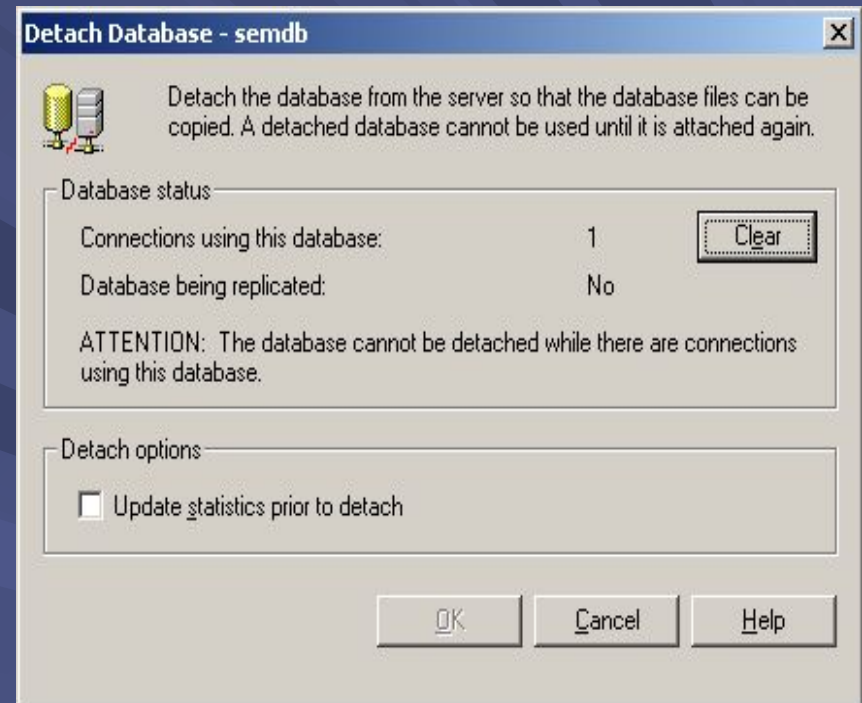
# Конфигурирование дисковой подсистемы

- При выборе дисковой подсистемы для размещения БД `tembd` необходимо помнить, что данная БД используется для временного хранения рабочих файлов.
- Оптимизация производительности данной БД предполагает возможность обеспечения обработки большого числа операций чтения и записи.
- Оптимизация производительности — размещение БД `tembd` на отдельном диске и использование отдельного дискового контроллера.



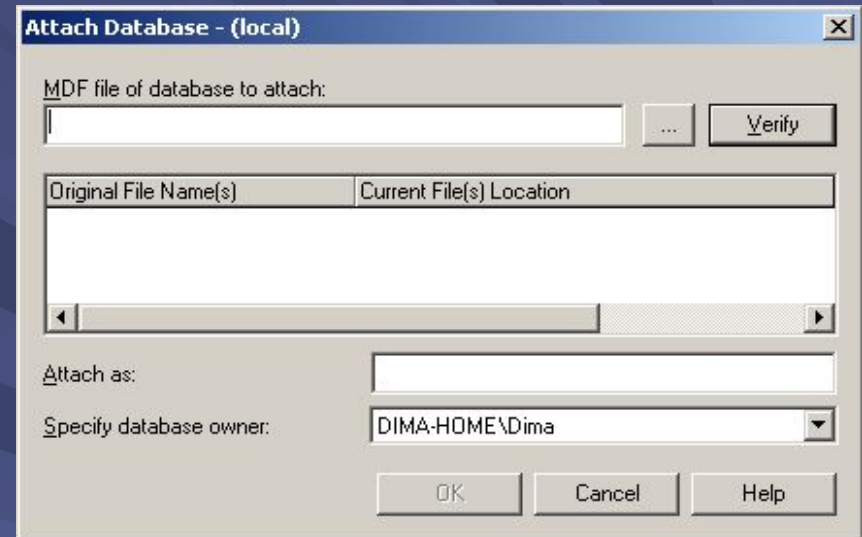
# Отсоединение и присоединение БД

- Для отсоединения БД с помощью SQL Server Enterprise Manager используется пункт меню Все задачи\Detach Database.
- Если с БД работают пользователи принудительное отключение выполняется с помощью кнопки Clear.
- Рекомендуется также обновить статистических сведений о БД.



# Отсоединение и присоединение БД

- После отсоединения можно переместить физические файлы БД на новые носители хранения информации.
- Для присоединения БД используется пункт контекстного меню Все задачи\Attach Database.
- В диалоговом окне указывается путь к основному файлу данных. Основной файл содержит информацию о размещении других файлов БД.
- В поле Current File Location для каждого перемещенного файла указывается его новое размещение.



# Отсоединение и присоединение БД

- Для отсоединения БД можно использовать и операторы Transact-SQL. Для этого применяется системная хранимая процедура `sp_detach_db`:
  - `sp_detach_db SEMDB, TRUE` – выполняет отсоединение БД `semdb`.
- Для присоединения используется системная хранимая процедура `sp_attach_db`:
  - `sp_attach_db SEMDB, @filename1 = 'C:\DB\Semdb_data.mdf'`

# Импорт и преобразование данных

- Заполнение пользовательской БД часто включает в себя импорт данных из внешних источников.
- Перед импортом необходимо выполнить следующие подготовительные действия:
  - оценить степень согласованности данных внешнего источника;
  - определить, потребуется ли изменение структуры таблиц (добавление новых полей);
  - определить необходимость в изменении формата данных;
  - определить, является ли импорт единовременным или периодическим;
  - определить режим доступа к внешним источникам.

# Преобразование данных посредством служб DTS

- В случае необходимости изменения импортируемых данных можно воспользоваться временными таблицами в SQL Server или непосредственно в процессе импорта.
- Изменение данных в процессе импорта называется DTS-трансформацией.
- DTS-трансформация предполагает, что данные обрабатываются посредством одной или нескольких операций или функций. При этом в источнике данные не изменяются.

# Средства преобразования данных

- В SQL Server имеется несколько средств экспорта/импорта данных:

Средство	Описание
Службы DTS	Графическое средство импорта, экспорта и модификации данных. Работают с разнообразными источниками. DTS создают пакеты обработки, которые можно выполнять по расписанию.
Утилита bcp	Утилита командной строки, предназначенная для загрузки данных из текстового файла в таблицу или представление SQL Server
Оператор BULK INSERT	Предназначен для загрузки данных из текстового файла формата ANSI в таблицу SQL Server. Оператор может быть включен в пакет DTS

# Службы DTS

- Службы DTS – мощный набор графических утилит и программируемых объектов для импорта, экспорта и преобразования данных. К числу утилит относятся:
  - Мастер DTS импорта/экспорта – dtswiz.exe
  - Конструктор DTS Designer
  - Средства выполнения DTS пакетов
- DTS пакет состоит из отдельных этапов, называемых *задачами DTS*.

# Типы подключений DTS

- Для пакета DTS необходимо указать действительный источник и приемник данных. В процессе выполнения пакет также может подключаться к дополнительным источникам данных.
- DTS пакеты используют следующие типы подключений:
  - Соединение с источником данных – подключение к стандартной БД, OLE DB подключение.
  - Соединение с файлом – соединение с текстовым файлом. Формат файла определяется свойствами подключения.
  - Канальное соединение – соединение с промежуточным файлом, который содержит строку для открытия OLE DB – соединения. При этом параметры соединения можно разместить в отдельном файле и редактировать строку соединения, не изменяя DTS пакета.



# Задачи DTS

- Пакеты DTS могут выполнять множество задач как последовательно, так и параллельно.
- Задача DTS – это отдельная единица работы по переносу и преобразованию данных.

# Задачи копирования и управления данными

- Bulk Insert – быстрая загрузка данных в таблицу или представление, но при ее выполнении не регистрируются строки при вставке которых произошла ошибка.
- Execute SQL – позволяет выполнять операторы T-SQL, например удаление таблиц или запуск хранимых процедур
- Copy SQL Server Object – копирует объекты (метаданные о БД) из одного экземпляра SQL Server в другой
- Transfer Database Object – набор задач, копирующих информацию уровня сервера.

# Задачи преобразования данных

- Transform Data – копирование, преобразование и вставка данных из источника в приемник, простейшая реализация канала данных.
- Data Driven Query – выборка, настройка и выполнение одного или нескольких операторов T-SQL для данных записи.

# Задачи, функционирующие как задания

- Active X Script – выполнение сценария ActiveX
- Dynamic Properties – выборка данных из внешнего источника и задание полученных значений указанным свойствам пакета.
- Execute Package – выполнение в процессе обработки других пакетов DTS
- Execute Process – запуск исполняемой программы или пакетного файла.
- File Transfer Protocol – загрузка данных с удаленного сервера или Интернет-ресурса.
- Send mail – отправка почтового сообщения.

# Ход обработки пакета DTS

- Службы DTS упорядочивают задачи при помощи констант предшествования и этапов.
- Этапы определяют последовательность выполнения задач пакета.
- Управление этой последовательностью осуществляется с помощью констант предшествования. Константы последовательно связывают все задачи пакета.
- Задачи без констант предшествования выполняются параллельно.
- Используются следующие константы предшествования:
  - Unconditional – если вторая задача связана с первой посредством данной константы, то она будет ожидать завершения первой и будет выполнена независимо от успеха или неудачи первой задачи;
  - On Success – если вторая задача связана с первой данным условием, то она будет ожидать первой и выполнится только в случае успеха первой задачи;
  - On Failure – если вторая задача связана с первой данным условием, то она будет ожидать окончания первой задачи и выполнится только в случае ошибки при выполнении первой.

# Варианты хранения DTS пакетов

- Пакет DTS можно хранить в различных местах сервера БД или вне ее:
  - БД SQL Server – в виде таблицы БД msdb на любом экземпляре SQL Server (по умолчанию)
  - Депозитарий Meta Data Services – в депозитарии служб.
  - Файл Visual Basic – пакет хранится в виде кода Visual Basic и доступен для редактирования посредством Visual Basic или C++.
  - Структурированный файл хранилища – хранится в виде отдельного файла.