

Визуальное проектирование БД.

Создание таблиц

- База данных является базовым элементом SQL Server 2000 и своего рода контейнером, в котором располагаются объекты и данные. Любой объект должен принадлежать базе данных. Каждая база данных имеет свою систему безопасности, связанную с системой безопасности SQL Server 2000.

(1,2,3)

Создание базы данных возможно несколькими способами:

- средствами языка Transact-SQL;
- с помощью графической утилиты Enterprise Manager;
- с помощью мастера создания базы данных Create Database Wizard.

(3,4,13)

- Для управления базами данных SQL Server 2000 используется папка Databases, имеющаяся в каждой инсталляции. Непосредственно в этой папке перечисляется набор баз данных, созданных на сервере.

(6,7)

В MS SQL Server 2000 реализованы следующие
типы индексов

- кластерные индексы;
- не кластерные индексы;
- уникальные индексы.

(1,5,4)

Не кластерные индексы являются наиболее типичными индексами. В отличие от кластерных, они не перестраивают физическую структуру таблицы, а лишь организуют ссылки на соответствующие строки (указатели – row locator), которые включают в себя:

- информацию об идентификационном номере файла (ID_file), в котором храниться строка;
- идентификационный номер страницы данных;
- номер соответствующей строки на странице;
- содержимое столбца.

(1,5,4)

Уникальные индексы гарантируют уникальность значений в индексируемом столбце. Он является надстройкой для таблицы и может быть реализован как для кластерного, так и для не кластерного индекса.

Право на создание индекса имеет только владелец таблицы, и это право не может быть передано другому пользователю.

(1,5,4)

В SQL Server 2000 различают несколько типов хранимых

процедур

- **системные хранимые процедуры**, предназначенные для работы с системными данными;
- **расширенные хранимые процедуры**, представляющие собой динамически подключаемые программы, которые в операционной среде могут играть роль самостоятельного приложения;
- **пользовательские хранимые процедуры**, создаваемые на уровне алгоритмов обработки данных.

(1,5,6,8)

Триггеры (triggers) в SQL Server 2000 представляют собой набор команд Transact-SQL, выполняемых автоматически при осуществлении тех или иных модификаций данных в таблице.

Физически триггеры являются ни чем иным, как хранимыми процедурами специального типа. Каждый триггер связан с конкретной таблицей и запускается сервером автоматически каждый раз, когда пользователи пытаются произвести вставку, изменение или удаление данных.

(1,5,6,8)

Триггеры различаются по типу команд, на которые они реагируют

- **INSERT TRIGGER** – запускаются при попытке вставить данные с помощью команды INSERT;
- **UPDATE TRIGGER** – запускаются при попытке изменения данных с помощью команды UPDATE;
- **DELETE TRIGGER** – запускаются при попытке удаления данных с помощью команды DELETE.
- (1,5,6,8)

Интерфейс доступа к данным OLE DB

OLE DB — это архитектура компонентов базы данных, реализующая эффективный доступ по локальной сети и через Интернет к источникам данных многих типов, в том числе реляционным источникам данных, почтовым файлам, неформатированным текстовым файлам и электронным таблицам. Набор OLE-интерфейсов, обеспечивает универсальный доступ к данным различного формата.

В архитектуре OLE DB приложения, получающие доступ к данным, называют *потребителями данных*, например: Access, Visual Basic. Программы, обеспечивающие внутренний доступ к данным, называют *средствами доступа к базам данных*, например: Microsoft OLE DB Provider for Microsoft SQL Server (рис.) или Microsoft Jet 4.0 OLE DB Provider для доступа к базе данных Microsoft Access внешнего потребителя.

(1,2,7)

Файлы данных и группы файлов

Для хранения базы данных предназначен набор файлов, персональный для любой базы данных. Каждый файл может принадлежать только одной базе данных. В SQL Server 2000 существует два типа файлов базы данных:

- файлы данных;
- файлы журнала транзакций.

Файлы данных (data file) предназначены для хранения информации, находящейся в таблицах базы данных. Кроме того, в этих файлах также размещены процедуры, ограничения, триггеры, индексы и другая информация;

(1,2,7)

В файлы журнала транзакций (transaction log file) SQL Server 2000 записывает информацию о ходе выполнения транзакций. В них размещается информация о состоянии данных перед началом транзакции, о выполняемых изменениях, заблокированных ресурсах и другая сопутствующая информация.

Файлы данных бывают двух ТИПОВ:

- Primary File (основной, или главный, файл);
- Secondary File (вторичный, или дополнительный, файл).
- Primary File (основной, или главный, файл);
- Secondary File (вторичный, или дополнительный, файл).

Одним из способов повышения надежности работы системы MS SQL Server 2000 является применение встроенного в систему механизма транзакций и контрольных точек и умелое его управление.

Транзакция – это одна или несколько последовательных команд языка Transact–SQL, образующих логически завершенный пакет и выполняемых как единое целое. Если по какой-либо причине хотя бы одна из команд пакета не выполняется, то происходит откат системы к состоянию, в котором она была до начала транзакции, и транзакция считается не выполненной.

Правила ASID

Обработка транзакций в любой системе управления базами данных должна производиться с соблюдением следующих правил ASID (Atomicity, Consistency, Isolation и Durability):

- **Atomicity** – **атомарность**, выполняемые в транзакции изменения либо выполняются все, либо не выполняются вовсе;
 - **Consistency** – **согласованность**, все данные после выполнения транзакции должны находиться в согласованном состоянии с соблюдением всех правил и ограничений целостности;
- (1,7,8)**

Правила ASID

- **Isolation** – **изолированность**, изменения данных, выполняемых различными одновременно работающими транзакциями, должны быть изолированы;
- **Durability** – **долговечность**, после завершения транзакции ничто не может вернуть систему в состояние, в котором она была до начала транзакции (происходит фиксация транзакции).

(1,7,8)

SQL Server 2000 поддерживает три вида определений транзакций:

- явное;
- автоматическое;
- Подразумеваемое.

(1,2,7)

Как и для таблиц, для представлений можно определить следующие права доступа:

- **SELECT** – просмотр данных;
- **INSERT** – добавление данных через представления;
- **UPDATE** – изменение данных в исходных таблицах;
- **DELETE** – удаление данных в исходных таблицах.
- (1,2,7)

Enterprise Manager является базовым инструментом при выполнении самых разнообразных задач:

- управления системой безопасностью;
- создания баз данных и их объектов;

Панель инструментов Enterprise Manager содержит:

- меню **Action** (Действие)- содержит тот же набор команд, что и контекстное меню объекта.
- **View** (вид)- позволяет изменить способ представления информации в результирующей панели Enterprise Manager
- **Tools** (сервис). становится доступным дополнительный инструментарий

(1,3)

Компонента SQLServerAgent

Компонента SQLServerAgent отвечает за автоматическое выполнение заданий и извещение системных операторов об ошибках в работе сервера. Запуск службы SQLServerAgent не обязателен.

С помощью службы SQL Server Agent можно выполнять запуск различных задач в определенное время, что при грамотном использовании может избавить администратора от большей части рутинной работы. Например, администратор может спланировать автоматическое выполнение операций резервного копирования и проверки целостности информации в базе данных во время наименьшей активности пользователей. При этом администратору не нужно находиться рядом и контролировать ход выполнения операций.

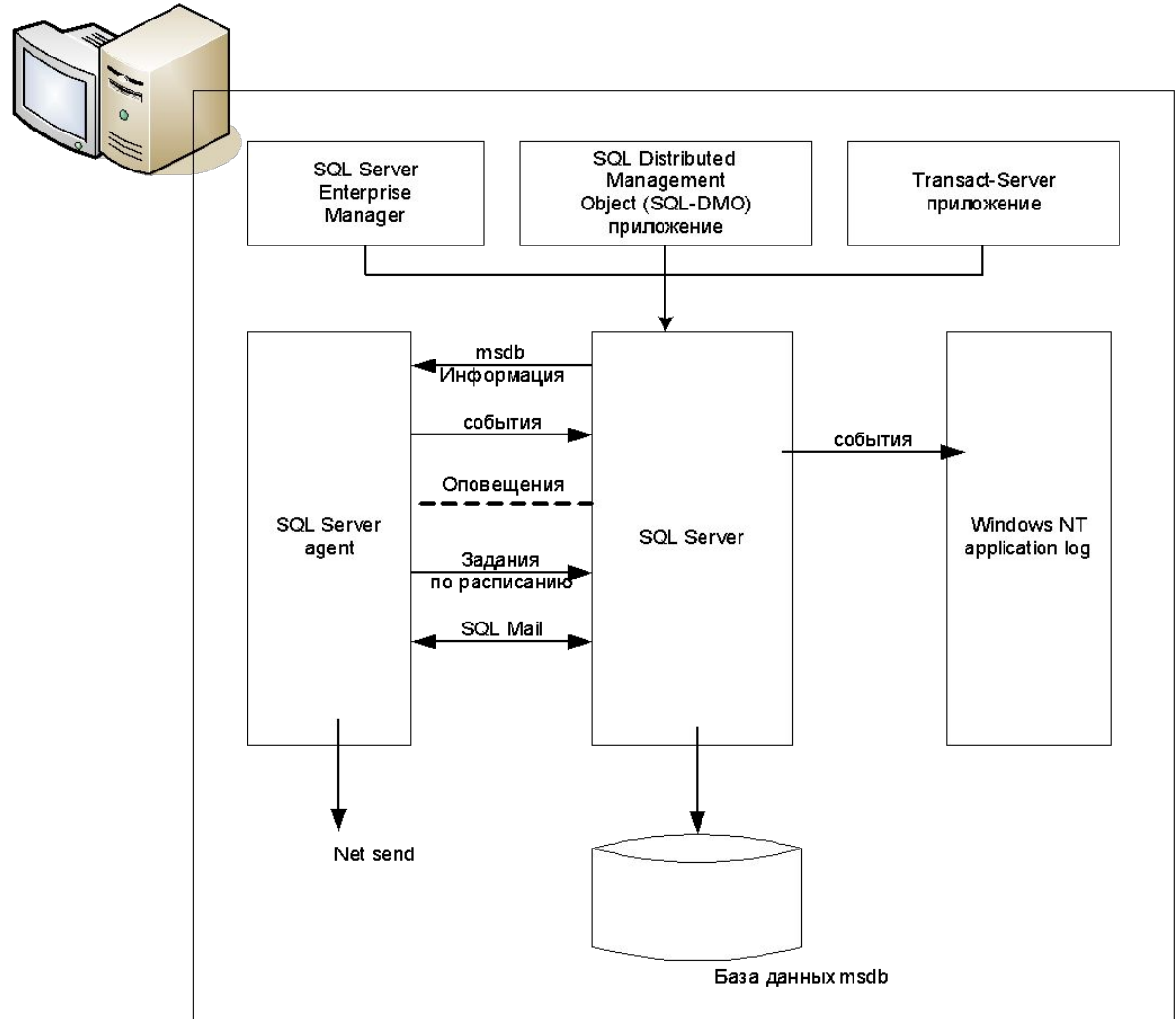
1,3,5,7

В работе службы SQLServerAgent применяются объекты трех типов: jobs (задания), operators (операторы), alerts (события). Объекты jobs (задания) описывают задачи, которые должны быть выполнены автоматически. Для каждого задания указывается одно или более расписаний (schedule) его запуска. Кроме того, задание может быть выполнено по требованию (on demand), т.е. вручную.

1,3,5,7

Компонента Microsoft Searchservice

Компонента Microsoft Searchservice (MSSearch) обеспечивает полнотекстовой поиск (Full-Text Search) нужной информации среди большого объема данных. На рисунке 5 изображено взаимодействие служб MSSearch и MSSQLServer. 1,3,5,7



Компонента Distributed Transaction Coordinator

Компонента Distributed Transaction Coordinator (MSDTC) координирует выполнение распределенных транзакций, выполняемых одновременно в различных базах данных и состоящих из нескольких локальных транзакций. Пользователи в одном запросе могут обращаться к различным базам данных, хранящимся на одном и том же или на разных серверах. Служба MSDTC автоматически отслеживает ситуации, в которых необходимо начать выполнение распределенных транзакций. В некоторых ситуациях пользователь может и не подозревать, что его транзакция выполняется как распределенная. Служба MSDTC скрывает от пользователя все действия по обработке распределенных транзакций. 1,3,5,7