

ЛЕКЦИЯ 4.

НАСТОЛЬНЫЕ И СЕРВЕРНЫЕ СУБД. ДОСТУП К ДАННЫМ БД

- Чисто настольные (персональные) СУБД
- Сетевые многопользовательские версии настольных СУБД
- Серверные СУБД
- Механизмы доступа к данным
- Схема функционирования СУБД
- Обязанности администратора БД

Предпосылки появления СУБД

До появления персональных ЭВМ в 80–х г 20 века существовала централизованная обработка и хранение данных на компьютерах (мэйнфреймы) семейства IBM-360/370 , мини-эвм типа DEC PDP. Тогда использовались неинтеллектуальные терминалы, и основным средством пользовательского интерфейса были устройства считывания перфокарт и перфолент.

- Практически отсутствовала персонализация рабочей среды - все программное обеспечение хранилось централизованно и

Настольные (персональные) СУБД

- Появление первых ПЭВМ и быстрый рост индустрии ПК привели к созданию первых настольных СУБД и росту их популярности
- Настольные СУБД не содержат специальных средств, управляющих данными. Все взаимодействие с данными осуществляется с помощью файловых сервисов ОС.
- Обработка данных БД целиком и полностью выполняется в пользовательском приложении.
- В состав таких СУБД включены средства, позволяющие создать более или менее

Первая персональная СУБД:

dBase

- Первая промышленная версия СУБД dBase появилась в начале 80-х годов 20 в.
- Благодаря простоте в использовании, нетребовательности к ресурсам ПЭВМ эта СУБД приобрела немалую популярность.
- С 1986 г. СУБД dBase обладала комфортной по тем временам средой разработки и средствами манипуляции данными, поэтому быстро заняла лидирующее положение среди настольных СУБД.
- С 2000 года dBase стала некоммерческим продуктом с доступными исходными текстами.

Особенности СУБД dBase

- Хранение данных в dBase основано на принципе <одна таблица - один файл> (расширение *.dbf).
- MEMO-поля и BLOB-поля хранятся в отдельных файлах (расширение *.dbt). Индексы для таблиц хранятся в отдельных файлах.
- Формат данных dBase является открытым (это означает возможность простого редактирования данных), что позволило другим производителям заимствовать формат для создания dBase-подобных СУБД.

СУБД Paradox

- СУБД Paradox создана в 1985 году. К началу 90-х годов Paradox был весьма популярной СУБД.
- Однако, в отличие от dBase, формат данных Paradox является закрытым, поэтому для доступа к данным этого формата требуются специальные библиотеки, <знающие> этот формат, например популярная библиотека Paradox Engine.
- <Закрытый> формат данных Paradox, позволяет реализовать такой сервис СУБД, как защита таблиц и отдельных полей паролем, хранение некоторых правил ссылочной целостности в самих таблицах БД. Эти сервисы недоступны при использовании <открытых> форматов данных в СУБД типа dBase.

Microsoft Access

- СУБД MS Access появилась в начале 90-х г. Это была первая настольная реляционная СУБД для 16-разрядной версии Windows, ориентирована на пользователей Office, в том числе и не знакомых с программированием.
- Все данные, относящиеся к одной БД (таблицы, индексы, правила ссылочной целостности, список пользователей, а также формы и отчеты) хранятся в одном файле, что в целом удобно.
- СУБД Access может быть использована как в качестве настольной СУБД так и в качестве клиента серверной СУБД Microsoft SQL Server

Современные настольные СУБД

Популярные настольные СУБД сегодня :

- имеют визуальные средства проектирования запросов, форм, отчетов и приложений;
- предоставляют доступ к данным серверных СУБД;
- имеют средства публикации данных в Internet и поддерживают создание приложений для редактирования данных с помощью Web-браузеров;
- предоставляют возможность хранить описания правил ссылочной целостности внутри базы данных.

Популярные персональные СУБД

СУБД	Производитель	URL
Visual dBase	dBase, Inc	http://www.dbase2000.com/
Paradox	Corel	http://www.corel.com/
Microsoft Access	Microsoft	http://www.microsoft.com/
Microsoft FoxPro	Microsoft	http://www.microsoft.com/
Microsoft Visual FoxPro	Microsoft	http://www.microsoft.com/
Microsoft Data Engine	Microsoft	http://www.microsoft.com/

Сетевые многопользовательские версии

- Следующим шагом в развитии персональных (настольных) СУБД было появление их сетевых многопользовательских версий, позволяющих обрабатывать данные, находящиеся на сетевом диске, многим пользователям одновременно.
- От чисто настольных СУБД такие версии отличаются наличием механизма блокировок частей файлов данных (содержащих одну или несколько записей таблицы), что позволяет обращаться к одному и тому же файлу нескольким пользователям

Проблема ссылочной

целостности

- Важной проблемой многопользовательских версий СУБД является возможность нарушения ссылочной целостности данных, так как единственным механизмом, контролирующим ее, является пользовательское приложение.
- Поэтому все пользовательские приложения должны содержать соответствующий программный код. Кроме того, доступ к файлам базы данных из любых других приложений должен быть

- В наиболее популярных настольных СУБД (например, Microsoft Access, Corel Paradox) программный код, контролирующей стандартную ссылочную целостность, содержится в библиотеках, используемых всеми приложениями, работающими с этой базой данных, а сама база данных при этом может содержать описание правил ссылочной целостности.

Недостатки сетевых версий

СУБД

- Недостатки сетевых многопользовательских версий СУБД проявляются не сразу, а лишь в процессе длительной эксплуатации, когда объем хранимых данных и число пользователей становятся достаточно велики - это приводит к снижению производительности приложений, и возникновению серьезных ошибок.
- Поскольку, вся реальная обработка данных в таких СУБД осуществляется в клиентском приложении, и любые библиотеки доступа к данным в этом случае также находятся в адресном пространстве клиентского приложения.

Перегрузка сети

- В сетевых версиях настольных СУБД при выполнении запросов данные, на основании которых выполняется такой запрос (это может быть одна или несколько таблиц целиком либо, один или несколько индексов и выбранные с их помощью части таблиц), должны быть доставлены в то же самое адресное пространство клиентского приложения.
- Это приводит к перегрузке сети при увеличении числа пользователей и объема данных, а также грозит иными неприятными последствиями например разрушением

Как с этим бороться?

- Известно много случаев, когда для решения подобных проблем закупалось дорогое сетевое оборудование с целью увеличения пропускной способности сети.
- Также применялись иные <ухищрения> наподобие хранения метаданных или наиболее часто используемых данных в клиентских приложениях или просто на локальных жестких дисках.
- Нередко также применялся подход, приводящий к децентрализации хранения данных: создавались локальные базы

Локальные БД

- Однотипные локальные БД (например для различных подразделений компании или для разных периодов: лет, кварталов, месяцев) облегчали работу, связанную с вводом данных, но повышали стоимость их статистической обработки и анализа - в этом случае нужно было обрабатывать данные из разных источников.
- Однако эти меры позволяли лишь отложить на время решение проблемы снижения производительности, но не устраняли главного недостатка сетевых версий систем **обработки данных в клиентском**

Серверные СУБД

- Следующим этапом развития СУБД для персональных компьютеров стали так называемые серверные СУБД.
- Радикальным решением проблемы сетевого трафика и иных проблем, возникающих при увеличении объема данных и числа пользователей, стал переход к архитектуре <клиент-сервер>, позаимствовавшей многие достоинства старой <мэйнфреймовой> модели вычислений, в частности, централизацию хранения и обработки данных.

Сервер баз данных

- Серверные СУБД реализуют принцип централизации хранения и обработки данных на одном выделенном компьютере в сети, где функционирует специальное приложение, называемое сервером баз данных.
- Сервер баз данных отвечает за работу с файлами БД, поддержку ссылочной целостности, резервное копирование, обеспечение авторизованного доступа к данным, протоколирование операций и, конечно, за выполнение пользовательских запросов на выборку и модификацию данных и метаданных

Клиентские приложения

- Клиентские приложения, являющиеся источниками запросов к базам данных, функционируют на рабочих станциях типа ПЭВМ в сети.
- Для клиентских приложений файлы, в которых хранятся данные, абсолютно бесполезны (и, при правильной организации доступа пользователей к файлам в сети, они вообще не должны быть доступны).
- Реально манипулировать файлами БД может только сервер баз данных, выполненный как приложение или сервис операционной

Популярные серверные СУБД

СУБД	Производитель	Url
Oracle	Oracle Corp.	http://www.oracle.com/
Microsoft SQL Server	Microsoft	http://www.microsoft.com/
Informix	Informix	http://www.informix.com/
Sybase	Sybase	http://www.sybase.com/
DB2	IBM	http://www-4.ibm.com/

Преимущества серверных СУБД

1. Одним из важнейших преимуществ архитектуры <клиент-сервер> является снижение сетевого трафика при выполнении запросов. Например, при необходимости выбора пяти записей из таблицы, содержащей миллион записей, клиентское приложение посылает серверу запрос, который сервером баз данных анализируется на корректность и, если запрос корректен, компилируется, оптимизируется и выполняется. После этого результат запроса (те самые пять записей, а вовсе не вся таблица) передается обратно

2. Вторым преимуществом архитектуры <клиент-сервер> является возможность хранения бизнес-правил (например, правил ссылочной целостности или ограничений на значения данных) на сервере, что позволяет избежать дублирования кода в различных клиентских приложениях, использующих общую базу данных. В этом случае любое редактирование данных, в том числе и редактирование средствами, не предусмотренными разработчиками информационной системы (например, различными утилитами администрирования сервера), может быть произведено только с учетом этих правил.

- 3. Третье преимущество серверных СУБД обусловлено тем, что часть кода, связанного с обработкой данных, также может быть реализована в виде хранимых процедур сервера, что позволяет еще больше <облегчить> клиентское приложение, а это означает, что требования к рабочим станциям могут быть не столь высоки. Это в конечном итоге снижает стоимость информационной системы даже при использовании дорогостоящих серверной СУБД и аппаратного обеспечения.

4. Современные серверные СУБД обладают широкими возможностями управления пользовательскими привилегиями и правами доступа к различным объектам базы данных. Как правило, в базе данных хранятся сведения о ее пользователях, их паролях и привилегиях, а каждый объект базы данных принадлежит какому-либо пользователю.

Большинство серверных СУБД поддерживает так называемые роли, представляющие собой совокупность прав на доступ к тем или иным объектам базы данных.

5. Современные серверные СУБД обладают также широкими возможностями резервного копирования и архивации данных, а нередко и оптимизации выполнения запросов.

Они также, как правило, предоставляют возможность параллельной обработки данных, особенно в случае использования многопроцессорных компьютеров в качестве сервера баз данных.

Доступ к данным баз данных

- Известно, что за организацию, размещение и оперирование с данными во внешней памяти отвечает операционная система ЭВМ, точнее, ее компонент который называется файловой системой.
- Файловая система не знает внутренней смысловой логики организации данных в файлах и оперирует с ними как с однородной совокупностью байтов.
- За организацию и поддержание логической структуры данных, доступ к данным при их обработке в оперативной и внешней памяти отвечает СУБД.

Механизмы доступа к данным

- При выборе СУБД для АИС необходимо понимать, с помощью каких средств разработки будет создаваться АИС на основе данной СУБД, а также о том, каким образом разработанные приложения будут манипулировать данными.
- От того, правильно ли выбран механизм доступа к данным, зависит очень многое, в частности производительность приложений, возможность применения тех или иных функциональных особенностей данной СУБД, простота разработки пользовательского интерфейса и ряд других факторов

Категории механизмов доступа к данным

Существует несколько способов доступа к данным баз данных из средств разработки и клиентских приложений, которые условно можно разделить на **две категории**:

- использование клиентского механизма (API или клиентских СОМ-объектов);
- применение универсальных механизмов доступа к данным.

Клиентские механизмы доступа

- **API.** Подавляющее большинство СУБД содержит в своем составе библиотеки, предоставляющие специальный *прикладной программный интерфейс (Application Programming Interface, API)* для доступа к данным этой СУБД. Обычно такой интерфейс представляет собой набор функций, вызываемых из клиентского приложения.
- **Клиентские COM-объекты.** Windows-версии клиентского ПО наиболее популярных серверных СУБД, в частности Microsoft SQL Server, Oracle, Informix, содержат также **COM-серверы**, предоставляющие объекты для доступа к данным и метаданным СУБД.

Функции API

- В случае сетевых версий **настоельных СУБД** функции **API** обеспечивают чтение/запись файлов базы данных.
- В случае **серверных СУБД** функции **API** инициируют передачу запросов серверу баз данных и получение от сервера результатов выполнения запросов или кодов ошибок.
- Библиотеки, содержащие API для доступа к данным серверной СУБД, обычно входят в состав ее клиентского программного обеспечения, устанавливаемого на компьютерах в сети, где функционируют клиентские приложения.

Достоинства и недостатки API

- Использование клиентского API (или клиентских СОМ-объектов) является наиболее очевидным (и самым **эффективным** с точки зрения производительности) способом манипуляции данными в приложении.
- Однако в этом случае созданное приложение сможет использовать данные только СУБД этого производителя, и замена СУБД на другую повлечет за собой **переписывание** значительной части кода клиентского приложения, поскольку клиентские API и объектные модели не подчиняются никаким стандартам и различны для разных СУБД.

Универсальные механизмы

- Универсальные механизмы доступа к данным обычно реализованы в виде библиотек и модулей, называемых драйверами или провайдерами, содержащих стандартный набор функций или классов, реализованных на основе функций клиентского API конкретных СУБД
- Достоинством универсальных механизмов является возможность применения одного и того же абстрактного API, или COM-серверов, компонентов, классов для доступа к разным типам СУБД.
- Недостаток: за подобную универсальность приходится платить невозможностью доступа к уникальной функциональности, специфичной для конкретной СУБД, снижением производительности приложений

Популярные универсальные

механизмы Наиболее популярными среди

универсальных механизмов доступа к данным можно назвать следующие:

- Open Database Connectivity (**ODBC**).
- **OLE DB**.
- ActiveX Data Objects (**ADO**).
- Borland Database Engine (BDE).

Универсальные механизмы ODBC, OLE DB и ADO фирмы Microsoft представляют собой по существу промышленные стандарты.

ODBC (Open Database Connectivity)

- С помощью ODBC можно манипулировать данными любой СУБД (и даже данными, не имеющими прямого отношения к базам данных, например данными в файлах электронных таблиц или в текстовых файлах), если для них имеется ODBC-драйвер.
- Для манипуляции данными используются как непосредственные вызовы ODBC API, так и другие универсальные механизмы доступа к данным, например OLE DB, ADO, BDE, реализующие стандартные функции на основе вызовов ODBC API в драйверах или провайдерах, специально предназначенных для работы с любыми ODBC-источниками.

OLE DB и ADO

- *OLE DB и ADO* – части универсального механизма доступа к данным Microsoft (*Microsoft Universal Data Access*), позволяющая осуществить доступ как к реляционным, так и к нереляционным источникам данных, таким как файловая система, данные электронной почты, многомерные хранилища данных и др.
- *OLE DB и ADO* становятся все более популярным способом доступа к данным, так как входят в состав таких широко используемых продуктов, как Microsoft Office и Microsoft Internet Explorer, а также включены в ядро операционных систем семейства Windows.

Схема функционирования серверной СУБД

- Работа системы управления базами данных связана с использованием трех уровней описаний данных: описание физической организации данных, описание общей логической структуры данных, описание данных прикладной программы (подмоделей).
- Представим последовательность действий, которые должна выполнить система управления базой данных, в процессе формирования записи соответствующей подмодели данных прикладной программы

СХЕМА РАБОТЫ СУБД



1. Прикладная программа Пр1 формирует запрос на чтение записи. Если обращение осуществляется к конкретной записи, то формируется значение ключа записи. Если записи обрабатываются последовательно, то ключей не требуется.
2. Система управления базой данных на основании описания подмодели данных для данной прикладной программы исследует вопрос о правомочности обращения к данным из программы.

3. На основе общей модели данных СУБД привязывает подмодель данных к модели и определяет, какие элементы данных необходимы.
4. На основе описания физической организации данных определяется, какие физические записи следует считать в системные буферы, чтобы сформировать затем требуемые данные.
5. Операционной системе выдается задание на чтение требуемых физических данных.

6. Работают программы, реализующие методы доступа к данным операционной системы.
7. Из внешних запоминающих устройств запрошенные физические записи перемещаются в системные буферы.
8. СУБД на основе описания общей модели данных, а также описания подмоделей программы формирует логическую запись в соответствии с подмоделью. При этом реализуется необходимое преобразование данных, вызванное разным определением данных в модели

9. Данные из системных буферов СУБД передаются в рабочую область прикладной программы Пр1.
10. Система управлений базой данных формирует и передает прикладной программе код возврата или информацию о своей работе в процессе обслуживания ее запроса.
11. Прикладная программа приступает к функциональной обработке переданных ей данных.

ФУНКЦИОНАЛЬНЫЕ БЛОКИ СУБД

- Процессор описания и поддержания структуры БД (ядро СУБД)
- Процессор запросов к БД
- Монитор транзакций
- Интерфейс ввода данных
- Интерфейс запросов
- Интерфейс выдачи сведений
- Генератор отчетов

АВТОМАТИЗИРОВАННЫЕ БАНКИ ДАННЫХ

- Известно, что основой любого управления является информация о состоянии объекта. Поэтому центром любой автоматизированной системы являются данные: их организация, ведение, хранение, использование.
- Меняются техника, программное хозяйство, но данные остаются, работа с ними является делом достаточно дорогим, и именно поэтому важны системные принципы организации данных, которые и положены в основу создания банков данных.

Состав банка данных

- Под *автоматизированным банком* данных понимается организационно-техническая система, представляющая собой совокупность баз данных пользователей, технических и программных средств формирования и ведения этих баз и коллектива специалистов, обеспечивающих функционирование системы.

Составные части банка данных



Составными частями любого банка данных являются база данных, система управления базой данных (СУБД), администратор банка данных, прикладное программное обеспечение

Основные функции банка данных

- адекватное информационное отображение предметной области,
- обеспечение хранения, обновления и выдачи необходимых данных пользователям в требуемой форме.

Администратор баз данных

- Администратора базы данных следует рассматривать как необходимый структурный элемент автоматизированного банка данных.
- Администратор базы данных — это лицо (или группа лиц), которое имеет глобальное представление об организации данных в системе и несет ответственность за их сохранность.
- Администратору баз данных отводится важная роль — ответственность за общее управление системой баз данных.

Обязанности администратора БД

- ***1. Определение информационного содержания базы данных***

Администратор базы данных с учетом запросов пользователей к базе принимает решение о том, какая информация должна содержаться в базе данных, т.е. определяет информационное содержание базы данных и задает их общую логическую организацию, так называемую модель данных.

- **2. *Определение структуры памяти и стратегии (механизмов) доступа.***

Администратор базы данных должен решить, каким образом данные представляются в базе данных, т.е. разработать физическую организацию данных.

- **3 *Взаимодействие с пользователем.***

Пользователи при участии администратора имеют возможность корректно определить собственный «взгляд» на базу данных, что выражается в задании подмодели данных.

- **4 Определение стратегии отказа и восстановления.**

Работа автоматизированной системы, использующей банк данных, существенно зависит от его успешного функционирования. В случае повреждения по какой-либо причине всей базы данных или ее некоторой части необходимо предусмотреть возможность восстановления данных с минимальной задержкой и влиянием на сохранившуюся часть базы данных. Администратор должен определить и реализовать соответствующую стратегию

- ***5 Модернизация и эффективность работы базы данных.***

Администратор базы данных ответствен за такую организацию системы, которая обеспечит максимальную эффективность ее функционирования, а также за выполнение всех модернизаций базы данных, направленных на более полное удовлетворение требований пользователей.

Средства администрирования СУБД

- Для выполнения своих функций администратор базы данных использует набор вспомогательных программ.
- Эти программы составляют существенную часть системы управления базами данных.
- К ним относятся, например, программы, ведения системного журнала, хранящего сведения о каждом обращении в базу данных, программы восстановления БД и программы анализа статистики использования данных.