

Базы данных

Лекция 2

Возникновение БД

- Кипу – узелковая письменность инков: III тысячелетие до н. э.
- Клинопись – письменность шумеров, персов и т.д.: ~3300 г. до н. э.
- ...
- Перфокарты – в «интеллектуальных машинах» С.Н. Корсакова: 1832 год
- Реляционная модель данных – Э.Ф. Кодд (E.F. Codd) 1969-1970 гг.
- Современные БД

БД и СУБД

- **База данных (БД)**– организованная структура, предназначенная для хранения информации. Современные БД позволяют размещать в своих структурах не только данные, но и методы (т.е. программный код), с помощью которых происходит взаимодействие с потребителем или другими программно-аппаратными комплексами.
- **Системы управления базами данных (СУБД)** – комплекс программных средств, предназначенных для создания структуры новой базы, наполнения ее содержанием, редактирования содержимого и визуализации информации. Под **визуализацией информации базы** понимается отбор отображаемых данных в соответствии с заданным критерием, их упорядочение, оформление и последующая выдача на устройство вывода или передача по каналам связи.

Классификация по среде хранения

- Во вторичной памяти, или традиционная: средой постоянного хранения является периферийная энергонезависимая память (вторичная память) — как правило жёсткий диск. В оперативную память СУБД помещает лишь кеш и данные для текущей обработки.
- В оперативной памяти : все данные на стадии исполнения находятся в оперативной памяти.
- В третичной памяти: средой постоянного хранения является отсоединяемое от сервера устройство массового хранения (третичная память), как правило на основе магнитных лент или оптических дисков. Во вторичной памяти сервера хранится лишь каталог данных третичной памяти, файловый кеш и данные для текущей обработки; загрузка же самих данных требует специальной процедуры.

Классификация по содержанию

- Географическая
- Историческая
- Научная
- Мультимедийная
- Клиентская, и т.д.



Другие виды БД

- Пространственная: БД, в которой поддерживаются пространственные свойства сущностей предметной области. Такие БД широко используются в геоинформационных системах.
- Временная, или темпоральная: БД, в которой поддерживается какой-либо аспект времени, не считая времени, определяемого пользователем.
- Пространственно-временная БД: БД, в которой одновременно поддерживается одно или более измерений в аспектах как пространства, так и времени.
- Циклическая: БД, объём хранимых данных которой не меняется со временем, поскольку в процессе сохранения новых данных они заменяют более старые данные. Одни и те же ячейки для данных используются циклически.

Классификация по технологии обработки данных

- Централизованная база данных хранится в памяти одной вычислительной системы. Если эта вычислительная Система является компонентом сети ЭВМ, возможен распределенный доступ к такой базе. Этот способ использования баз данных часто применяют в локальных сетях ПК.
- Распределенная база данных состоит из нескольких, возможно, пересекающихся или даже дублирующих друг друга частей, хранимых в различных ЭВМ вычислительной сети. Работа с такой базой осуществляется с помощью системы управления распределенной базой данных (СУРБД).

Банк данных

- Совокупность БД и программы СУБД образует информационно-поисковую систему, называемую банком данных.
- По способу доступа к данным базы данных делятся на базы данных с локальным доступом и базы данных с удаленным (сетевым доступом).
- Системы централизованных баз данных с сетевым доступом предполагают различные архитектуры таких систем: файл-сервер; клиент-сервер.

Файл-сервер

- Архитектура систем БД с сетевым доступом предполагает выделение одной из машин сети в качестве центральной (сервер файлов). На такой машине хранится совместно используемая централизованная БД.
- Все другие машины сети выполняют функции рабочих станций, с помощью которых поддерживается доступ пользовательской системы к централизованной базе данных.
- Файлы базы данных в соответствии с пользовательскими запросами передаются на рабочие станции, где в основном и производится обработка. При большой интенсивности доступа к одним и тем же данным производительность такой информационной системы падает.
- Пользователи могут создавать также на рабочих станциях локальные БД, которые используются ими монопольно.

Клиент-сервер

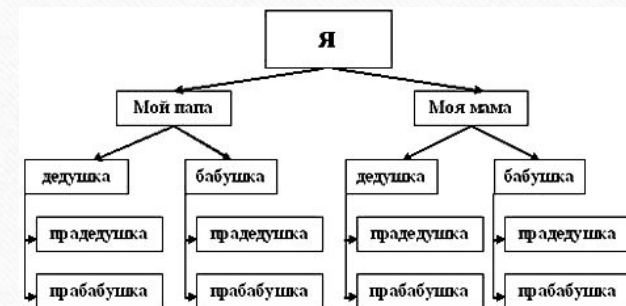
- В отличие от предыдущей системы, центральная машина (сервер базы данных), помимо хранения централизованной базы данных, должна обеспечивать выполнение основного объема обработки данных.
- Запрос на использование данных, выдаваемый клиентом (рабочей станцией), приводит к поиску и извлечению данных на сервере.
- Извлеченные данные транспортируются по сети от сервера к клиенту. Спецификой архитектуры клиент-сервер является использование языка запросов SQL.

Классификация по модели данных

- Иерархическая
- Объектная и объектно-ориентированная
- Объектно-реляционная
- Сетевая
- Функциональная
- Реляционная

Иерархическая СУБД

- В иерархической БД существует упорядоченность элементов в записи, один элемент считается главным, остальные – подчиненными.
- Данные в записи упорядочены в определенную последовательность как ступеньки лестницы, и поиск данных может осуществляться лишь последовательным «спуском» со ступеньки на ступеньку.
- Поиск какого-либо элемента данных в такой системе может оказаться довольно трудоемким из-за необходимости последовательно проходить несколько предшествующих иерархических уровней.
- Иерархическую БД образует каталог файлов, хранимых на диске; Такой же базой данных является родовое генеалогическое дерево.



Объектная СУБД

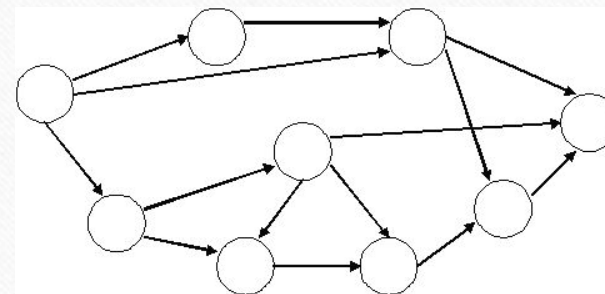
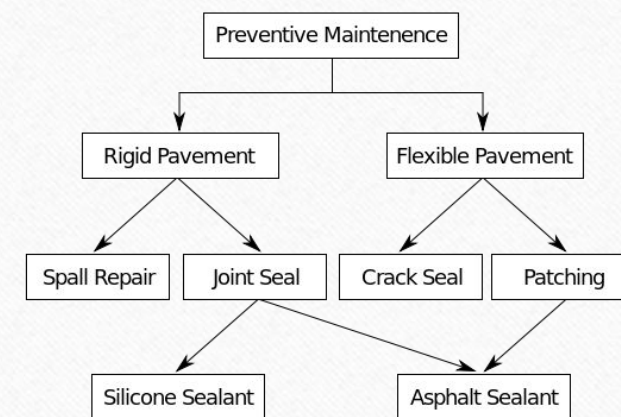
- ООСУБД являются системой управления базами данных, в которых информация представлена в виде объектов, используется в объектно-ориентированном программировании.
- Объектные базы данных отличаются от реляционных баз данных, являющихся таблично-ориентированными, и являются гибридом обоих подходов.
- Предназначены для работы с Delphi, Ruby, Python, Perl, Java, C #, Visual Basic .NET, C ++, Objective-C и Smalltalk или имеют собственные языки
- Доступ к данным может быть произведён быстрее, так как зачастую нет необходимости в присоединении таблиц, объект может быть получен непосредственно, без поиска, используя указатели. А язык программирования и схема базы данных используют одни те же определения типов.

Объектно-реляционная СУБД

- ОРСУБД – реляционная СУБД (РСУБД), поддерживающая некоторые технологии, реализующие объектно-ориентированный подход: объекты, классы и наследование реализованы в структуре баз данных и языке запросов.
- Объектно-реляционными СУБД являются, например, широко известные Oracle Database, Informix, DB2, PostgreSQL.

Сетевая модель данных

- Сетевая модель данных — логическая модель данных, являющаяся расширением иерархического подхода
- Эта база данных отличается большей гибкостью, так как в ней существует возможность устанавливать дополнительно к вертикальным иерархическим связям горизонтальные связи. Это облегчает процесс поиска требуемых элементов данных, так как уже не требует обязательного прохождения всех предшествующих ступеней.
- Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может быть любое число предков.

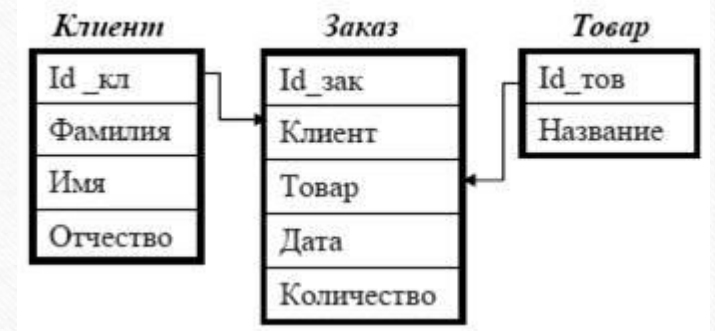


Функциональная модель данных

- Функциональная модель сообщает, какие функции (или операции) определены на этом объекте.
- Этот метод представляет функцию, сопоставляющую владельцу набор, т.е. многозначную функцию, такую как используемые в функциональной модели данных.
- Набор может иметь нуль, один или более элементов; иными словами, соответствующее множество может быть пустым или содержать единственный элемент или же несколько элементов.

Реляционная СУБД

- В реляционной БД под записью понимается строка прямоугольной таблицы. Элементы записи образуют столбцы этой таблицы (поля).
- Все элементы в столбце имеют одинаковый тип (числовой, символьный), а каждый столбец – неповторяющееся имя. Одинаковые строки в таблице отсутствуют.
- Преимущество таких БД – наглядность и понятность организации данных, скорость поиска нужной информации.
- Примером реляционной БД служит таблица на странице классного журнала, в которой записью является строка с данными о конкретном ученике, а имена полей (столбцов) указывают, какие данные о каждом ученике должны быть записаны в ячейках таблицы.
<http://habrahabr.ru/company/mailru/blog/266811/>



Структура простейшей базы данных

- **Структура базы** определяет методы занесения данных и хранения их в базе.
- Если в БД нет никаких данных (**пустая база**), то это все равно полноценная БД, т. к. она содержит информацию о структуре базы.
- Основными объектами БД являются таблицы. Простейшая база данных имеет хотя бы одну таблицу. Структура простейшей базы данных тождественно равна структуре ее таблицы.
- Структуру двумерной таблицы образуют столбцы и строки. Их аналогами в структуре простейшей базы данных являются **поля** и **записи** . Если записей в таблице нет, то ее структура образована набором полей. Изменив состав полей базовой таблицы (или их свойства), тем самым изменяем структуру данных, и, соответственно, получаем новую базу данных.

Поля БД

- Поля БД определяют групповые свойства данных, записываемых в ячейки, принадлежащие каждому из полей.
- Например, свойства полей могут быть такими: имя поля, тип поля, размер поля, формат поля, маска ввода, подпись, значение по умолчанию, условие на значение, обязательное поле, индексированное поле, пустые строки, и т.д.
- Типы данных: текстовый, числовой, денежный, дата/время, счетчик, поле мемо (большой объем текста), логический, поле объекта OLE (для мультимедийных объектов), гиперссылка, место подстановок.

Режимы работы с базами данных

- Проектировщики – разрабатывают структуру таблиц базы и согласовывают ее с заказчиком; разрабатывают объекты, предназначенные для автоматизации работы и ограничения функциональных возможностей работы с базой (из соображений безопасности);
- Пользователи – работают с базами данных, наполняют ее и обслуживают.
- Проектный режим СУБД – в проекционном режиме создаются и изменяются структура базы и ее объекты.
- Пользовательский режим СУБД – в пользовательском используются ранее подготовленные объекты для наполнения БД или получения данных из нее.

Объекты базы данных

- **Таблицы** – основные объекты любой БД, в которых хранятся все данные, имеющиеся в базе, и хранится сама структура базы (поля, их типы и свойства).
- **Отчеты** – предназначены для вывода данных, причем для вывода не на экран, а на печатающее устройство (принтер). В них приняты специальные меры для группирования выводимых данных и для вывода специальных элементов оформления, характерных для печатных документов (верхний и нижний колонтитулы, номера страниц, время создания отчета и другое).
- **Страницы** или страницы доступа к данным – специальные объекты БД, выполненные в коде HTML, размещаемые на web-странице и передаваемые клиенту вместе с ней. Сам по себе объект не является БД, посетитель может с ее помощью просматривать записи базы в полях страницы доступа. Т.о., страницы – интерфейс между клиентом, сервером и базой данных, размещенным на сервере.
- **Макросы и модули** – предназначены для автоматизации повторяющихся операций при работе с системой управления БД, так и для создания новых функций путем программирования. Макросы состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой. Модули создаются средствами внешнего языка программирования. Это одно из средств, с помощью которых разработчик БД может заложить в нее нестандартные функциональные возможности, удовлетворить специфические требования заказчика, повысить быстродействие системы управления, уровень ее защищенности.

Запросы и формы

- **Запросы** – служат для извлечения данных из таблиц и предоставления их пользователю в удобном виде. С их помощью выполняют отбор данных, их сортировку и фильтрацию. Можно выполнить преобразование данных по заданному алгоритму, создавать новые таблицы, выполнять автоматическое заполнение таблиц данными, импортированными из других источников, выполнять простейшие вычисления в таблицах и многое другое.
- Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную *результатирующую таблицу* (*моментальный снимок*) – образ отобранных из базовых таблиц полей и записей. Работа с образом происходит быстрее и эффективнее, нежели с таблицами, хранящимися на жестком диске.
- Обновление БД тоже можно осуществить посредством запроса. В базовые таблицы все данные вносятся в порядке поступления, т.е. они не упорядочены. Но по соответствующему запросу можно получить отсортированные и отфильтрованные нужным образом данные.
- **Формы** – средства для ввода данных, предоставляющие пользователю необходимые для заполнения поля. В них можно разместить специальные элементы управления (счетчики, раскрывающиеся списки, переключатели, флажки и прочее) для автоматизации ввода. Пример, заполнение определенных полей бланка. При выводе данных с помощью форм можно применять специальные средства их оформления.

Системы программирования

- **Система программирования** – инструментальное ПО, предназначенное для поддержки разработки программных систем на этапах программирования и отладки.
- Каждая система программирования должна иметь некоторый встроенный в нее язык программирования, предназначенный для общения с человеком – разработчиком программной системы.

Компоненты программы

- **1. Текстовый редактор** . Специализированные редакторы ориентированы на конкретный язык программирования. Подобные редакторы созданы для всех популярных языков программирования и дополнительно могут автоматически проверять правильность синтаксиса программы непосредственно во время ее ввода.
- **2. Компилятор** . Исходный текст с помощью **программы-компилятора** переводится в машинный код. На этом этапе создается промежуточный объектный код (двоичный файл, стандартное решение .OBJ).
- **3. Редактор связей и библиотеки функций** . Исходный текст большой программы, как правило, состоит из нескольких модулей. Каждый модуль состоит отдельный файл с объектным кодом, которые затем нужно объединить в единое целое. К ним нужно добавить машинный код подпрограмм, реализующих различные стандартные функции (например, вычисление \sin и \cos). Такие функции содержатся в библиотеках (файлы с расширением .LIB), которые поставляются вместе с компилятором. Объектный код обрабатывается специальной программой – **редактором связей** или **сборщиком** , который выполняет связывание объектных модулей и машинного кода стандартных функций, находя их в библиотеках, и формирует на выходе работоспособное приложение – **исполнимый код** для конкретной платформы. Если по каким-то причинам объектный модуль или нужная библиотека не найдены, то сборщик сообщает об ошибке и готовой программы не получается.
- **4. Исполнимый код** – это законченная программа, которую можно запустить на любом компьютере, где установлена операционная система, для которой эта программа создавалась. Как правило, итоговый файл имеет расширение .EXE или .COM.

Временная сложность

- В информатике временная сложность алгоритма определяет время работы, используемое алгоритмом, как функции от длины строки, представляющей входные данные.
- Например, если время, которое нужно алгоритму для выполнения работы, для всех входов длины n не превосходит $5n^3 + 3n$ для некоторого n (большого некоторого n_0), асимптотическая временная сложность равна $O(n^3)$.
- Поскольку производительность алгоритма может отличаться при входах одного и того же размера, обычно используется временная сложность наихудшего случая поведения алгоритма. Менее часто, и это обычно оговаривается специально, время измеряется как средняя сложность.

Временная сложность

Примеры алгоритмов	Время работы	Название
Поиск наименьшего или наибольшего элемента в неотсортированном массиве	$O(n)$	линейное время
Двоичный поиск	$O(\log n)$	логарифмическое время
Поиск в k -мерном дереве	$O(n^c)$ при $0 < c < 1$	дробная степень
Максимально быстрая сортировка сравнением	$O(n \log n)$	линейно-логарифмическое время
Сортировка пузырьком, сортировка вставками, прямая свёртка	$O(n^2)$	квадратичное время

Д/3:

- Webex,
- Skype,
- Outlook,
- Office365,
- Зеркальные серверы