

Содержание:

image not found or type unknown



Введение

Базы данных использовались в вычислительной технике с незапамятных времен. В первых компьютерах использовались два вида внешних устройств – магнитные ленты и магнитные барабаны. Емкость магнитных лент была достаточно велика. Устройства для чтения-записи магнитных лент обеспечивали последовательный доступ к данным. Для чтения информации, которая находилась в середине или конце магнитной ленты, необходимо было сначала прочитать весь предыдущий участок. Следствием этого являлось чрезвычайно низкая производительность операций ввода-вывода данных во внешнюю память. Магнитные барабаны давали возможность произвольного доступа, но имели ограниченный объем хранимой информации.

Разумеется, говорить о какой-либо системе управления данными во внешней памяти, в тот момент не приходилось. Каждая прикладная программа, которой требовалось хранить данные во внешней памяти, сама определяла расположение каждого блока на магнитной ленте. Прикладная программа также брала на себя функции информационного обмена между оперативной памятью и устройствами внешней памяти с помощью программно-аппаратных средств низкого уровня. Такой режим работы не позволяет или очень затрудняет поддержку на одном носителе нескольких архивов долговременно хранимой информации. Кроме того, каждой прикладной программе приходилось решать проблемы именования частей данных и структуризации во внешней памяти.

История БД фактически началась с появлением магнитных дисков. Такие устройства внешней памяти обладали существенно большей емкостью, чем магнитная лента и барабаны, а также обеспечивали во много раз большую скорость доступа в режиме произвольной выборки. В отличие от современных систем управления, которые могут применяться для самых различных баз данных, подавляющее большинство ранее разработанных СУБД были тесно связаны с пользовательской базой для того, чтобы увеличить скорость работы, хоть и в

ущерб гибкости.

Первоначально СУБД применялись только в крупных организациях с мощной аппаратной поддержкой, необходимой для работы с большими объемами данных.

Современные авторы часто употребляют термины "банк данных" и "база данных" как синонимы, однако в общеотраслевых руководящих материалах по созданию банков данных Государственного комитета по науке и технике (ГКНТ), изданных в 1982 г., эти понятия различаются.

Там приводятся следующие определения банка данных, базы данных и СУБД:

Банк данных (Бнд) - это система специальным образом организованных данных - баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

База данных (БД) - именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Система управления базами данных (СУБД) - совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

Глава 1. Базы данных

1.1 Основные понятия баз данных

В современных базах данных хранятся не только данные, но и информация.

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.

Существует много систем управления базами данных. Они могут по-разному работать с разными объектами и предоставляют пользователю разные функции и средства. Большинство СУБД опираются на единый устоявшийся комплекс основных понятий.

1.2 Структура простейшей базы данных

Если в БД нет никаких данных (**пустая база**), то это все равно полноценная БД, т.к. она содержит информацию о структуре базы.

Структура базы определяет методы занесения данных и хранения их в базе. БД могут содержать различные объекты. 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 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.

1.3 Режимы работы с базами данных

Обычно с БД работают две категории исполнителей:

- 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.

СУБД имеет два режима: проектировочный и пользовательский.

В проектировочном режиме создаются и изменяются структура базы и ее объекты. В пользовательском используются ранее подготовленные объекты для наполнения БД или получения данных из нее.

1.4 Объекты базы данных

БД может содержать разные типы объектов. Каждая СУБД может реализовывать свои типы объектов.

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.

1.5 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 результирующую 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.

1.6 Системы программирования. Понятие 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.

В самом общем случае для создания программы на выбранном языке программирования нужно иметь следующие компоненты .

1. Текстовый редактор . Т.к. текст программы записывается с помощью слов, происходящих от английского языка, и символов для записи всевозможных операций, то формировать файл с исходным текстом программы можно в любом редакторе. 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 из отдельного файла 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 .

1.7 Экспертные системы

Экспертная система (ЭС) - expert system - особый класс систем искусственного интеллекта, включающий знания об определённой слабо структурированной и трудно формализуемой узкой предметной области и способная предлагать и объяснять разумные решения. Ее основным назначением является: интерпретация данных, диагностика состояния, мониторинг, прогнозирование, планирование и обучение. Все эти задачи по существу сводятся к распознаванию образов.

ЭС - интеллектуальная система, предназначенная для оказания консультационной помощи специалистам, работающим в некоторой предметной области. Выделяют два типа экспертных систем.

ЭС первого типа . Предназначены для специалистов, чей профессиональный уровень не слишком высок. В базах знаний таких систем содержатся знания, полученные от специалистов экстра-класса.

ЭС второго типа . Предназначены для специалистов высокой квалификации, выполняя для них значительную часть рутинных операций и просмотр больших массивов информации. В экспертной системе обязательно присутствует система объяснений.

Системы управления базами данных позволяют объединять большие объемы информации и обрабатывать их, сортировать, делать выборки по определенным критериям и т. п.

Глава 2. Система управления базами данных

2.1 История развития СУБД

Рост производительности персональных вычислительных машин спровоцировал развитие СУБД, как отдельного класса. К середине 60-х годов прошлого века уже существовало большое количество коммерческих СУБД. Интерес к базам данных увеличивался все больше, так что данная сфера нуждалась в стандартизации.

Автор комплексной базы данных Integrated Data Store Чарльз Бахман (Charles Bachman) организовал целевую группу DTG (Data Base Task Group) для утверждения особенностей и организации стандартов БД в рамках CODASYL - группы, которая отвечала за стандартизацию языка программирования COBOL. Уже в 1971 году был представлен свод утверждений и замечаний, который был назван Подход CODASYL, и спустя некоторое время появились первые успешные коммерческие продукты, изготовленные с учетом замечаний вышеупомянутой рабочей группы. В 1968 году отметилась и компания IBM, которая представила собственную СУБД под названием IMS. Фактически данный продукт представлял собой компиляцию утилит, которые использовались с системами System/360 на шаттлах Аполлон. Решение было разработано согласно коцпетам CODASYL, но при этом была применена строгая иерархия для структуризации данных. В свою очередь в варианте CODASYL за базис была взята сетевая СУБД. Оба варианта, между тем, были приняты сообществом позднее как классические варианты организации работы СУБД, а сам Чарльз Бахман в 1973 году получил премию Тьюринга за работу Программист как навигатор. В 1970 году сотрудник компании IBM Эдгар Кодд, работавший в одном из отделений Сан Хосе (США), в котором занимались разработкой систем хранения, написал ряд статей, касающихся навигационных моделей СУБД. Заинтересовавшись вопросом он разработал и изложил несколько инновационных подходов касательно оптимальной организации систем управления БД. Работа Кодда внесла значительный вклад в развитие СУБД и является действительным основоположником теории реляционных баз данных. Уже в 1981 году Э.Ф.Кодд создал реляционную модель данных и применил к ней операции реляционной алгебры.

2.2 Классификация

В зависимости от архитектуры построения системы управления базами СУБД могут подразделяться на следующие типы:

1. Иерархические
2. Многомерные
3. Реляционные
4. Сетевые

5.Объектно-ориентированные

6.Объектно-реляционные

7.[править]

2.3 Файловые системы

Представим себе, что имеется некоторый носитель информации определенной емкости, устройство для чтения-записи на этот носитель в режиме произвольного доступа и прикладные программы, которые используют конкретный носитель для ввода-вывода информации во внешнюю память. В этом случае, каждая прикладная программа должна знать где и в каком месте хранятся необходимые данные. Так как прикладных программ больше, чем носителей информации, то несколько прикладных программ могут использовать один накопитель. Что произойдет, если одной из прикладных программ потребуется дозаписать свои данные на диск? Может произойти наложение: ситуация в которой данные одной программы будут перезаписаны другой программой. Важным шагом в развитии информационных систем явился переход к использованию централизованных систем управления файлами. С точки зрения прикладной программы, файл – именованная область внешней памяти, в которую можно записывать данные, и из которой можно их считывать. Для того чтобы была возможность считать информацию из какой либо области внешней памяти необходимо знать имя этого сектора(имя файла), размер самой области и его физическое расположение. Сама система управления файлами выполняет следующие функции:

- распределение внешней памяти;
- отображение имен файлов в соответствующие адреса во внешней памяти;
- обеспечение доступа к данным.

Рассмотрение особенностей реализации отдельных систем управления файлами выходит за рамки данной темы. На данном этапе достаточно знать, что прикладные программы видят файл как линейную последовательность записей и могут выполнить над ним ряд операций. Основные операции с файлами в СУФ:

- создать файл (определенного типа и размера)

- открыть ранее созданный файл
- прочитать из файла определенную запись
- изменить запись
- добавить запись в конец файла
- [править]

2.4 СУБД крупных ЭВМ

Данный этап развития связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и различных моделях фирмы Hewlett Packard. В таком случае информация хранилась во внешней памяти центральной ЭВМ.

Пользователями баз данных были фактически задачи, запускаемые в основном в пакетном режиме. Интерактивный режим доступа обеспечивался с помощью консольных терминалов, которые не обладали собственными вычислительными ресурсами (процессором, оперативной памятью, внешней памятью) и служили только устройствами ввода-вывода для центральной ЭВМ. Программы доступа к БД писались на различных языках программирования и запускались как обычные числовые программы. Особенности данного этапа:

Все СУБД базируются на мощных мультипрограммных ОС (Unix и др.).

- Поддерживается работа с централизованной БД в режиме распределенного доступа. Функции управления распределением ресурсов выполняются операционной системой.
- Поддерживаются языки низкого манипулирования данными, ориентированные на навигационные методы доступа к данным. Значительная роль отводится администрированию данных.
- Проводятся серьезные работы по обоснованию и формализации реляционной модели данных. Была создана первая система (System R), реализующая идеологию реляционной модели данных.
- Проводятся теоретические работы по оптимизации запросов и управлению распределенным доступом к централизованной БД, было введено понятие транзакции.
- Большой поток публикаций по всем вопросам теории БД. Результаты научных исследований активно внедряются в коммерческие СУБД.

- Появляются первые языки высокого уровня для работы с реляционной моделью данных (SQL), однако отсутствуют стандарты для этих языков.
- [править]

2.5 Настольные СУБД

Компьютеры стали ближе и доступнее каждому пользователю. Исчез благоговейный страх рядовых пользователей перед непонятными и сложными языками программирования. Появилось множество программ, предназначенных для работы неподготовленных пользователей. Простыми и понятными стали операции копирования файлов и переноса информации с одного компьютера на другой, распечатка текстов, таблиц и других документов. Системные программисты были отодвинуты на второй план. Каждый пользователь мог себя почувствовать полным хозяином этого мощного и удобного устройства, позволяющего автоматизировать многие аспекты собственной деятельности. И, конечно, это сказалось и на работе с базами данных. Новоявленные СУБД позволяли хранить значительные объемы информации, они имели удобный интерфейс для заполнения, встроенные средства для генерации различных отчетов. Эти программы позволяли автоматизировать многие учетные функции, которые раньше велись вручную. Постоянное снижение цен на персональные компьютеры сделало такое ПО доступным не только для организаций и фирм, но и для отдельных пользователей. Компьютеры стали инструментом для ведения документации и собственных учетных функций. Это все сыграло как положительную, так и отрицательную роль в области развития баз данных. Кажущаяся простота и доступность персональных компьютеров и их программного обеспечения породила множество дилетантов. Много было создано систем-однодневок, которые не отвечали законам развития и взаимосвязи реальных объектов. Однако доступность персональных компьютеров заставила пользователей из многих областей знаний, которые ранее не применяли вычислительную технику в своей деятельности, обратиться к ним. И спрос на развитые удобные программы обработки данных заставлял поставщиков программного обеспечения поставлять все новые системы, которые принято называть настольными СУБД. Значительная конкуренция среди поставщиков заставляла совершенствовать эти конфигурации, предлагая новые возможности, улучшая интерфейс и быстродействие систем, снижая их стоимость. Наличие на рынке большого числа СУБД, выполняющих сходные функции, потребовало разработки методов экспорта-импорта данных для этих систем и открытия

форматов хранения данных. Но и в этот период появлялись любители, которые вопреки здравому смыслу разрабатывали собственные СУБД, используя стандартные языки программирования. Это был тупиковый вариант, потому что дальнейшее развитие показало, что перенести данные из нестандартных форматов в новые СУБД было гораздо труднее, а в некоторых случаях требовало таких трудозатрат, что легче было бы все разработать заново, но данные все равно надо было переносить на новую более перспективную СУБД. И это тоже было результатом недооценки тех функции, которые должна была выполнять СУБД. Особенности этого этапа следующие:

Стандартизация высокоуровневых языков манипулирования данными (разработка и внедрение стандарта SQL92 во все СУБД).

Все СУБД были рассчитаны на создание БД в основном с монопольным доступом. И это понятно. Компьютер персональный, он не был подсоединен к сети, и база данных на нем создавалась для работы одного пользователя. В редких случаях предполагалась последовательная работа нескольких пользователей, например, сначала оператор, который вводил бухгалтерские документы, а потом главбух, который определял проводки, соответствующие первичным документам.

Большинство СУБД имели развитый и удобный пользовательский интерфейс. В большинстве существовал интерактивный режим работы с БД как в рамках описания БД, так и в рамках проектирования запросов. Кроме того, большинство СУБД предлагали развитый и удобный инструментарий для разработки готовых приложений без программирования.

Во всех настольных СУБД поддерживался только внешний уровень представления реляционной модели, то есть только внешний табличный вид структур данных.

При наличии высокоуровневых языков манипулирования данными типа реляционной алгебры и SQL в настольных СУБД поддерживались низкоуровневые языки на уровне отдельных строк таблиц.

В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных. Эти функции должны были выполнять приложения, однако скудость средств разработки приложений иногда не позволяла это сделать, и в этом случае эти функции должны были выполняться пользователем, требуя от него дополнительного контроля при вводе и изменении информации, хранящейся в БД.

Наличие монопольного режима работы фактически привело к вырождению функций администрирования БД.

Сравнительно скромные требования к аппаратному обеспечению со стороны настольных СУБД. Вполне работоспособные приложения, разработанные, например, на Clipper, работали на PC 286. В принципе, их даже трудно назвать полноценными СУБД. Яркие представители этого семейства — очень широко использовавшиеся до недавнего времени СУБД Dbase (DbaseIII+, DbaseIV), FoxPro, Clipper, Paradox.

Заключение

Преимущества использования БД

Рассмотрим, **какие преимущества получает пользователь при использовании БД как безбумажной технологии:**

- ***Компактность***

Информация хранится в БД, нет необходимости хранить многотомные бумажные картотеки

- ***Скорость***

Скорость обработки информации (поиск, внесение изменений) компьютером намного выше ручной обработки

- ***низкие трудозатраты***

Нет необходимости в утомительной ручной работе над данными

- ***применимость***

Всегда доступна свежая информация

Дополнительные преимущества появляются при использовании БД в многопользовательской среде, поскольку становится возможным осуществлять централизованное управление данными.

Современные системы управления базами данных обеспечивают как физическую (независимость от способа хранения и метода доступа), так и логическую независимость данных (возможность изменения одного приложения без изменения остальных приложений, работающих с этими же данными).

Современные СУБД дают возможность включать в них не только текстовую и графическую информацию, но и звуковые фрагменты и даже видеоклипы.

Простота использования СУБД позволяет создавать новые базы данных, не прибегая к программированию, а пользуясь только встроенными функциями. СУБД обеспечивают правильность, полноту и непротиворечивость данных, а также удобный доступ к ним.

Популярные СУБД - **FoxPro, Access for Windows, Paradox**. Для менее сложных применений вместо СУБД используются **информационно-поисковые системы (ИПС)**, которые выполняют следующие функции:

- хранение большого объема информации;
- быстрый поиск требуемой информации;
- добавление, удаление и изменение хранимой информации;
- вывод ее в удобном для человека виде.

Список литературы

1. Интернет