ПРЕДИСЛОВИЕ

За последние годы в нашей стране произошли значительные перемены, которые не могли не затронуть области информатики и вычислительной техники. Всего каких-нибудь десять лет назад работа с базами данных и электронными таблицами была уделом профессиональных программистов.

Системы управления базами данных (СУБД) не были предназначены для широкого пользователя.

Их основным потребителем был военно-промышленный комплекс.

С появлением огромного числа банков, акционерных обществ и частных компаний ситуация резко изменилась.

В настоящее время обработка и хранение информации являются важнейшими задачами.

Потеря информации или ее несвоевременное получение могут обернуться потерей денег. Именно этими обстоятельствами можно объяснить столь быстрый рост компьютерной техники и стремительное развитие электронных таблиц и систем управления базами данных в нашей стране и за рубежом.

Для оперативного, гибкого и эффективного управления предприятиями, фирмами и организациями различных форм собственности, телекоммуникационными средствами гражданского и военного назначения, информационно-вычислительными, экологическими, радиолокационными и радионавигационными системами широко внедряются системы автоматизированного управления, ядром которых являются базы данных (БД).

При большом объеме информации и сложности производимых с ней операций проблема эффективности средств организации хранения, доступа и обработки данных приобретает особое значение.

Важность и значимость баз данных в современной жизни определяют серьезные требования, предъявляемые к квалификации специалистов, создающих приложения на их основе.

ОСНОВЫ ТЕОРИИ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

Определение и назначение баз данных. Системы управлении базами данных

С самого начала развития вычислительной техники образовались **два основных направления** ее использования.

Первое направление — применение вычислительной техники для выполнения численных расчетов, которые слишком долго или вообще невозможно производить вручную.

Второе направление — использование средств вычислительной техники в автоматических или автоматизированных информационных системах.

В самом широком смысле **информационная система** представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и/или вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса.

Обычно объемы информации, с которыми приходится иметь дело таким системам, достаточно велики, а сама информация имеет довольно сложную структуру. Классическими примерами информационных систем являются банковские системы, системы резервирования авиационных или железнодорожных билетов, мест в гостиницах и т.д.

Второе направление возникло несколько позже первого. Это связано с тем, что на заре вычислительной техники **компьютеры обладали ограниченными возможностями**. Надежное и долговременное хранение информации возможно только при наличии запоминающих устройств, сохраняющих информацию после выключения электрического питания. Оперативная память этим свойством обычно не обладает.

Используемые в ранних ЭВМ два вида устройств внешней памяти — магнитные ленты и барабаны — были несовершенными. Магнитные ленты обладали достаточно большой емкостью, но по своей физической природе обеспечивали лишь последовательный доступ к данным. Магнитные барабаны, обеспечивая возможность произвольного доступа к данным, имели ограниченный размер. Появление новых носителей данных — в первую очередь, жестких дисков — дало толчок к работам по созданию информационных компьютерных систем.

Основу любой информационной системы составляет **база данных**, т.е. набор данных, организованных специальным образом.

В действующем в настоящее время Законе РФ «О правовой охране программ для электронных вычислительных машин и баз данных» от 23.09.92 № 3523-1 дастся следующее определение:

«**База данных -** это объективная форма представления и организации совокупности данных (например, статей, расчетов), систематизированных таким образом, чтобы эти данные могли быть найдены и обработаны с помощью ЭВМ».

Файл — это место фактического хранения информации. В файле различают структуру и собственно данные. Структура файла остается неизменной, а информация (данные) может изменяться при операциях обращения к нему.

В качестве основной структурообразующей единицы хранимых в файле данных принимается **хранимая запись**. Хранимые записи состоят из фиксированной совокупности полей, служащих для представления значений какого-либо типа (чисел, литерных строк, дат, булевских значений, денежных единиц и т.д.), и могут иметь формат фиксированной или переменной длины. Полям, как правило, присваиваются уникальные в данной базе имена, ассоциируемые с предметной областью.

Если в качестве примера базы данных рассмотреть картотеку сотрудников некоторого абстрактного предприятия, то единицей хранимых данных может быть запись персональной информации по каждому сотруднику с полями: табельный номер (формат поля — целое число); фамилия, имя, отчество (формат поля — литерная строка определенной длины); дата рождения (формат поля — дата); заработная плата (формат поля — действительное число) и т.д.

Информационные системы ориентированы главным образом на хранение, выбор и модификацию постоянно существующей информации.

Структура информации зачастую очень сложна, и хотя структуры данных различны в разных информационных системах, между ними часто бывает много общего. На начальном этапе использования вычислительной техники для управления информацией проблемы структуризации данных решались индивидуально в каждой информационной системе.

Поскольку информационные системы содержат сложные структуры данных, дополнительные индивидуальные средства управления этими данными, являясь существенной частью информационных систем, практически повторялись от одной системы к другой. Стремление выделить общую часть информационных систем, ответственную за управление сложно структурированными данными, явилось первой побудительной причиной создания систем управления базами данных.

Компоненты наиболее полного варианта СУБД следующие:

- среда пользователя, дающая возможность непосредственного управления данными с клавиатуры;
- алгоритмический язык для программирования прикладных систем обработки данных, реализованный как интерпретатор (последний позволяет быстро создавать и сглаживать программы);

- компилятор для придания завершенной программе вида готового коммерческого продукта в форме независимого EXE-файла;
- программы-утилиты для быстрого программирования рутинных операций (генераторы отчетов, форм, таблиц, экранов, меню и других приложений).

Собственно СУБД — это инструментальная оболочка пользователя. Наличие в СУБД языка программирования позволяет создавать сложные системы обработки данных, ориентированные на конкретные задачи и конкретного пользователя.

Области применения баз данных

Автоматизированные информационные системы (АИС), основу которых составляют базы данных, появились в 60-х годах XX века в военной промышленности и бизнесе — там, где были накоплены значительные объемы полезных данных. Первоначально АИС были ориентированы лишь на работу с информацией фактического характера — числовыми или текстовыми характеристиками объектов. Затем по мере развития техники появилась возможность обработки текстовой информации на естественном языке.

Принципы хранения разных видов информации в АИС аналогичны, но алгоритмы ее обработки определяются характером информационных ресурсов. Соответственно различают два класса АИС: **документальные** и **фактографические**.

Документальные АИС служат для работы с документами на естественном языке. Наиболее распространенный тип документальных АИС — информационно-поисковые системы, предназначенные для накопления и подбора документов, удовлетворяющих заданным критериям. Эти системы могут выполнять просмотр и подборку монографий, публикаций в периодике, сообщений пресс-агентств, текстов законодательных актов и т.д.

Фактографические АИС оперируют фактическими сведениями, представленными в формализованном виде, и используются для решения задач обработки данных.

Обработка данных — специальный класс решаемых на ЭВМ задач, связанных с вводом, хранением, сортировкой, отбором и группировкой записей данных однородной структуры. К задачам этого класса относятся: учет товаров в магазинах и на складах; начисление зарплаты; управление производством, финансами, телекоммуникациями и т.п.

Различают фактографические АИС **оперативной обработки** данных, подразумевающие быстрое обслуживание относительно простых запросов от большою числа пользователей, и фактографические АИС **аналитической обработки**, ориентированные на выполнение сложных запросов, требующих проведения статистической обработки исторических (накопленных за некоторый промежуток времени) данных, моделирования процессов предметной области и прогнозирования развития этих процессов.

Таким образом, АИС применяются в следующих областях:

- организация хранилищ данных;
- системы анализа данных;
- системы принятия решений;
- мобильные и персональные базы данных:
- географические базы данных;
- мультимедиа базы данных;
- распределенные информационные системы;
- базы данных для всемирной сети World Wide Web.

Информационная модель данных и ее состав

Каждая информационная система в зависимости от назначения имеет дело с той или иной частью конкретного мира, которую принято называть ее **предметной областью**. Анализ предметной области является необходимым начальным этапом разработки любой информационной системы. Именно на этом этапе определяются информационные потребности всей совокупности пользователей будущей системы, которые, в свою очередь, предопределяют содержание ее базы данных.

Предметная область конкретной информационной системы рассматривается, прежде всего, как некоторая совокупность реальных объектов, которые представляют интерес для ее пользователей. Примерами объектов предметной области могут служить персональные ЭВМ, программные продукты и их пользователи. Каждый из этих объектов обладает определенным набором свойств (атрибутов). Так, например, компьютер характеризуется названием фирмы-производителя, идентификатором модели, типом микропроцессора, объемом оперативной и внешней памяти, типом графической карты и т.д.

Информационный объект — это описание некоторой сущности предметной области, т.е. реального объекта, процесса, явления или события. Информационный объект (сущность) образуется совокупностью логически взаимосвязанных атрибутов (свойств), представляющих собой качественные и количественные характеристики объекта (сущности).

Между объектами предметной области могут существовать связи, имеющие различный содержательный смысл. Эти связи могут быть **обязательными** или **факультативными** (необязательными).

Если вновь порожденный объект оказывается по необходимости связанным с каким-либо объектом предметной области, то между этими двумя объектами существует обязательная связь. В противном случае связь является факультативной.

Совокупность объектов предметной области и связей между ними характеризует **структуру предметной области**.

Множество объектов предметной области, значения атрибутов объектов и связи между ними могут изменяться во времени. Изменения могут сводиться к появлению новых или исключению из рассмотрения некоторых существующих объектов в предметной области, установлению новых или разрушению существующих связей между ними. Следовательно, с каждым моментом времени можно сопоставить некоторое *состояние предметной области.*

Информационно-логическая модель (ИЛМ) — это совокупность информационных объектов (сущностей) предметной области и связей между ними.

Процесс создания информационной модели начинается с определения концептуальных требований будущих пользователей БД.

Концептуальная модель представляет собой интегрированные концептуальные требования всех пользователей к базе данных данной предметной области. При этом усилия разработчика должны быть направлены в основном на структуризацию данных, принадлежащих будущим пользователям БД и выявление взаимосвязей между ними.

Возможно, что отраженные в концептуальной модели взаимосвязи между объектами окажутся впоследствии нереализуемыми средствами выбранной СУБД, что потребует ее изменения. Версия концептуальной модели, которая может быть реализована конкретной СУБД, называется **логической моделью**.

Логическая модель, отражающая логические связи между атрибутами объектов вне зависимости от их содержания и среды хранения, может быть реляционной, иерархической или сетевой.

Типы взаимосвязей в модели

На практике часто используются связи, устанавливающие различные виды соответствия между объектами «связанных» типов, — это **один к одному** (1:1), **один ко многим** (1 :M), **многие ко многим** (M:M).

Связь **один к одному** означает, что каждому экземпляру первого объекта (A) соответствует только один экземпляр второго объекта (B) и, наоборот, каждому экземпляру второго объекта (B) соответствует только один экземпляр первого объекта (A).

Связь **один ко многим** означает, что каждому экземпляру одного объекта (А) может соответствовать несколько экземпляров другого объекта (В), а каждому экземпляру второго объекта (В) может соответствовать только один экземпляр первого объекта (А).

Связь **многие ко многим** означает, что каждому экземпляру одного объекта (А) могут соответствовать несколько экземпляров второго объекта (В) и, наоборот, каждому экземпляру второго объекта (В) могут соответствовать тоже несколько экземпляров первого объекта (А).

Пример. Рассмотрим совокупность следующих информационных объектов:
СТУДЕНТ (Номер студента, ФИО, Дата рождения, Номер группы);
СТИПЕНДИЯ (Номер студента, Размер стипендии);
ГРУППА (Номер группы, Специальность);
ПРЕПОДАВАТЕЛЬ (Код преподавателя, ФИО, Должность).

Здесь информационные объекты СТУДЕНТ и СТИПЕНДИЯ связаны отношением один к одному, так как каждый студент может иметь только одну стипендию и каждая стипендия может быть назначена только одному студенту.

Информационные объекты ГРУППА и СТУДЕНТ связаны отношением один ко многим, так как одна группа может включать в себя много студентов, в то время как каждый студент может обучаться только в одной группе.

Информационные объекты СТУДЕНТ и ПРЕПОДАВАТЕЛЬ связаны отношением многие ко многим, так как один студент может обучаться у многих преподавателей и один преподаватель может обучать многих студентов.

Нормализация баз данных

Одни и те же данные могут группироваться в таблицы различными способами. Группировка атрибутов в отношениях должна быть рациональной, т.е. минимизирующей дублирование данных и упрощающей процедуры их обработки и обновления. Устранение избыточности данных, являющееся одной из важнейших задач при проектировании баз данных, обеспечивается **нормализацией**.

Нормализация — это формальный аппарат ограничений на формирование таблиц (отношений), который позволяет устранить дублирование, обеспечивает непротиворечивость хранимых данных и уменьшает трудозатраты на ведение (ввод, корректировку) базы данных.

Процесс нормализации заключается в разложении (декомпозиции) исходных отношений БД на более простые отношения. При этом на каждой ступени этого процесса схемы отношений приводятся в нормальные формы. Для каждой ступени нормализации имеются наборы ограничений, которым должны удовлетворять отношения БД. Тем самым удаляется из таблиц базы избыточная неключевая информация.

Средства ускоренного доступа к данным

Чтобы пользователь чувствовал себя комфортно, время ожидания ответа на запрос к БД не должно превышать нескольких секунд. В связи с этим требованием специально разрабатываются методы ускорения выборки, позволяющие обойтись без полного перебора строк при выполнении реляционных операций модификации отношений и отбора данных.

Наиболее эффективны методы индексирования и хеширования значений ключей отношения.

Индексирование — логическая сортировка строк таблицы — заключается в создании вспомогательных файлов, содержащих упорядоченные списки значений ключей отношения со ссылками на строку отношения, в которой они находятся.

Индексные файлы занимают дополнительную память, но резко ускоряют поиск благодаря применению метода половинного деления.

Для одного отношения может быть создано несколько индексов. Кроме того, можно создать индекс для нескольких отношений, если они содержат одинаковые атрибуты, что позволит ускорить выполнение операций соединения этих отношений. Индексы позволяют находить строки, в которых значения ключевых полей совпадают с заданным значением или принадлежат заданному интервалу.

Хеширование (hashing) — использование хэш-функций, которые вычисляют вес строки таблицы по значению ее ключевых атрибутов. Результат вычисления хэш-функции — целое число в диапазоне физических номеров строк таблицы.

Идеальная хэш-функция должна давать разные значения веса для разных ключевых атрибутов. Но это не всегда возможно. На практике обычно используют простые хэш-функции, например f(к) = к mod р. где к — целое число, первичный ключ отношения; р — простое целое число; mod — операция, вычисляющая остаток при целочисленном делении. Если ключевой атрибут — строка символов, то для вычисления f(к) выбирается один из методов преобразования строки в число, например вычисление контрольной суммы.

Для организации доступа к данным при хешировании создается таблица с пустыми строками, которая заполняется следующим образом:

- по первичному ключу новой строки вычисляется значение хэш-функции f(k) и результат трактуется как номер строки в созданной таблице;
- если строка уже занята, производится проверка следующих строк по специальному алгоритму до тех пор, пока не будет обнаружено свободное место.

Аналогично производится поиск нужной строки:

- если после вычисления f(к) на месте в таблице, которое соответствует вычисленному значению, оказывается пустая строка, значит, искомой строки просто нет;
- если значение ключа совпало с искомым, поиск заканчивается;
- если же значение ключа не совпало с искомым, проверяются следующие строки таблицы до обнаружения строки с нужным ключом (в этом случае искомая строка найдена) или пустой строки (в этом случае искомая строка отсутствует).

Если таблица заполнена не более чем на 60 %, то для размещения новой или поиска существующей строки необходимо проверить в среднем не более двух ячеек. Хеширование используют для поиска строк.