

ЛЕКЦИЯ №1 ВВЕДЕНИЕ. ЦЕЛИ И ЗАДАЧИ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЕ. ТРЕБОВАНИЯ К БАЗАМ ДАННЫХ.

Введение	1
1.1 Файловые системы	1
1.2 Понятие БД и СУБД	3
1.3 Уровни абстракции в СУБД. Функции абстрактных данных	5
1.4 Функции СУБД	6
1.5 Экспертные системы и базы знаний	6

Введение

Исторически сложившееся развитие вычислительных систем обусловило необходимость хранения в электронном (машиночитаемом) виде все большего количества информации. Одновременно с совершенствованием и дальнейшим развитием вычислительных систем росли объемы информации, подлежащей обработке и хранению. Сложности, возникшие при решении на практике задач структурированного хранения и эффективной обработки возрастающих объемов информации, стимулировали исследования в соответствующих областях. Задачи хранения и обработки данных были формализованы. Была создана теоретическая база для решения задач такого класса, результатом реализации на практике которой стали системы, предназначенные для организации обработки, хранения и предоставления доступа к информации. Позже такие системы стали называть системами баз данных.

Одновременно с развитием систем баз данных, происходило интенсивное развитие средств вычислительной техники, используемой для работы с большими объемами информации. Вычислительная мощность и, особенно, объемы запоминающих устройств первых вычислительных систем были недостаточны для хранения и обработки информации в объемах, необходимых на практике.

По мере развития систем баз данных, менялись принципы организации данных в них: первоначально данные представлялись на основе иерархической, а в последствии сетевой модели. В конце 1970-х – начале 1980-х годов начали появляться первые реляционные продукты. В настоящее время системы баз данных на основе реляционной модели занимают лидирующее положение, несмотря на заявления многих исследователей о скором переходе к объектно-ориентированным системам. В настоящее время объектно-ориентированные системы, тем не менее, развиваются, хотя темпы их развития и сдерживаются медленным принятием соответствующих стандартов. Кроме того, многие коммерческие реляционные системы приобретают объектно-ориентированные черты. На основании этого, можно предположить, что в будущем объектно-ориентированные системы будут постепенно вытеснять реляционные.

В настоящее время ведутся исследования в следующих направлениях:

1. дедуктивные системы;
2. экспертные системы;
3. расширяемые системы;
4. объектно-ориентированные системы.

1.1 Файловые системы

Историческим шагом явился переход к использованию централизованных систем управления файлами. С точки зрения прикладной программы файл - это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные. Правила именования файлов, способ доступа к данным, хранящимся в файле, и

структура этих данных зависят от конкретной системы управления файлами и, возможно, от типа файла. Система управления файлами берет на себя распределение внешней памяти, отображение имен файлов в соответствующие адреса во внешней памяти и обеспечение доступа к данным.

Первая развитая файловая система была разработана фирмой IBM для ее серии 360. К настоящему времени она очень устарела, и мы не будем рассматривать ее подробно. Заметим лишь, что в этой системе поддерживались как чисто последовательные, так и индексно-последовательные файлы, а реализация во многом опиралась на возможности только появившихся к этому времени контроллеров управления дисковыми устройствами. Если учесть к тому же, что понятие файла в OS/360 было выбрано как основное абстрактное понятие, которому соответствовал любой внешний объект, включая внешние устройства, то работать с файлами на уровне пользователя было очень неудобно. Требовался целый ряд громоздких и перегруженных деталями конструкций. Все это хорошо знакомо программистам среднего и старшего поколения, которые прошли через использование отечественных аналогов компьютеров IBM.

1.1.1 Структуры файлов

Дальше мы будем говорить о более современных организациях файловых систем. Начнем со структур файлов. Прежде всего, практически во всех современных компьютерах основными устройствами внешней памяти являются магнитные диски с подвижными головками, и именно они служат для хранения файлов. Такие магнитные диски представляют собой пакеты магнитных пластин (поверхностей), между которыми на одном рычаге движется пакет магнитных головок. Шаг движения пакета головок является дискретным, и каждому положению пакета головок логически соответствует цилиндр магнитного диска. На каждой поверхности цилиндр "высекает" дорожку, так что каждая поверхность содержит число дорожек, равное числу цилиндров. При разметке магнитного диска (специальном действии, предшествующем использованию диска) каждая дорожка размечается на одно и то же количество блоков таким образом, что в каждый блок можно записать по максимуму одно и то же число байтов. Таким образом, для произведения обмена с магнитным диском на уровне аппаратуры нужно указать номер цилиндра, номер поверхности, номер блока на соответствующей дорожке и число байтов, которое нужно записать или прочитать от начала этого блока.

1.1.2 Именованние файлов

Остановимся коротко на способах именования файлов. Все современные файловые системы поддерживают многоуровневое именование файлов за счет поддержания во внешней памяти дополнительных файлов со специальной структурой - каталогов. Каждый каталог содержит имена каталогов и/или файлов, содержащихся в данном каталоге. Таким образом, полное имя файла состоит из списка имен каталогов плюс имя файла в каталоге, непосредственно содержащем данный файл. Разница между способами именования файлов в разных файловых системах состоит в том, с чего начинается эта цепочка имен.

В этом отношении имеются два крайних варианта. Во многих системах управления файлами требуется, чтобы каждый архив файлов (полное дерево справочников) целиком располагался на одном дисковом пакете (или логическом диске, разделе физического дискового пакета, представляемом с помощью средств операционной системы как отдельный диск). В этом случае полное имя файла начинается с имени дискового устройства, на котором установлен соответствующий диск.

1.1.3 Защита файлов

Поскольку файловые системы являются общим хранилищем файлов, принадлежащих, вообще говоря, разным пользователям, системы управления файлами

должны обеспечивать авторизацию доступа к файлам. В общем виде подход состоит в том, что по отношению к каждому зарегистрированному пользователю данной вычислительной системы для каждого существующего файла указываются действия, которые разрешены или запрещены данному пользователю. Существовали попытки реализовать этот подход в полном объеме. Но это вызывало слишком большие накладные расходы как по хранению избыточной информации, так и по использованию этой информации для контроля правомочности доступа.

1.1.4 Режим многопользовательского доступа.

Исторически в файловых системах применялся следующий подход. В операции открытия файла (первой и обязательной операции, с которой должен начинаться сеанс работы с файлом) помимо прочих параметров указывался режим работы (чтение или изменение). Если к моменту выполнения этой операции от имени некоторой программы А файл уже находился в открытом состоянии от имени некоторой другой программы В (правильнее говорить "процесса", но мы не будем вдаваться в терминологические тонкости), причем существующий режим открытия был несовместимым с желаемым режимом (совместимы только режимы чтения), то в зависимости от особенностей системы программе А либо сообщалось о невозможности открытия файла в желаемом режиме, либо она блокировалась до тех пор, пока программа В не выполнит операцию закрытия файла.

1.2 Понятие БД и СУБД

Система баз данных – это компьютеризированная система основная задача которой – хранение информации и предоставление доступа к ней по требованию.

Система баз данных включает в себя

(ри
с. 0.1):

1. данные, непосредственно сохраняемые в базе данных;
2. аппаратное обеспечение;
3. программное обеспечение;
4. пользователей:
 - а) прикладные программисты;
 - б) конечные пользователи;
 - в) администраторы баз данных.

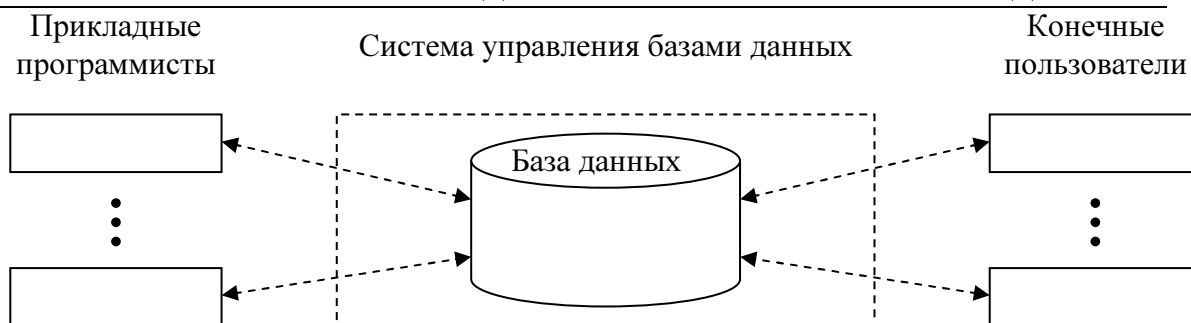


рис. 0.1 Система баз данных.

Данные в базе данных являются интегрированными и, как правило, общими. Понятие интегрированных данных подразумевает возможность представления базы данных как объединение нескольких отдельных файлов данных, полностью или частично не перекрывающихся. Понятие общие подразумевает возможность использования отдельных областей данных, в базе данных несколькими различными пользователями.

К аппаратному обеспечению системы относятся накопители для хранения информации, вместе с устройствами ввода-вывода, контролерами устройств и т.д.; вычислительная техника, используемая для поддержки работы ПО системы.

Программное обеспечение является промежуточным слоем между собственно физической базой данных и пользователями системы и называется диспетчером базы данных или системой управления базами данных, СУБД (DBMS). Все запросы пользователей обрабатываются СУБД.

Таким образом, **СУБД** – это специализированное программное обеспечение, предоставляющее пользователю базы данных возможность работать с ней, не вникая в детали хранения информации на уровне программного обеспечения.

Прикладные программисты – отвечают за написание прикладных программ, использующих базу данных.

Конечные пользователи – работают с базой данных непосредственно, через рабочую станцию или терминал. Конечный пользователь может получить доступ к базе данных используя соответствующее прикладное ПО.

Администраторы базы данных – технические специалисты, осуществляющие создание БД, технический контроль работы СУБД и др. операции. Администраторы базы данных отвечают за реализацию решений администратора данных. Администратор данных решает, какие данные необходимо хранить в БД, обеспечивает поддержание порядка при обслуживании и использовании хранимых в БД данных.

Функции администратора базы данных:

определение концептуальной схемы. Администратор БД определяет какие именно данные необходимо сохранять в БД. Этот процесс обычно называют логическим (или концептуальным) проектированием БД. После определения содержимого БД на абстрактном уровне, администратор БД создает соответствующую концептуальную схему, с помощью концептуального ЯОД.

Определение внутренней схемы. Администратор БД решает, как данные должны быть представлены в хранимой БД. Этот процесс называют физическим проектированием. После завершения физического проектирования администратор БД с помощью внутреннего ЯОД должен создать соответствующую структуру хранения, а также определить отображение между внутренней и концептуальной схемой.

ЛЕКЦИЯ №1 ВВЕДЕНИЕ. ЦЕЛИ И ЗАДАЧИ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЕ. ТРЕБОВАНИЯ К БАЗАМ ДАННЫХ.

Взаимодействие с пользователями. Администратор БД обеспечивает пользователей необходимыми им данными. Для этого администратор БД должен написать (или оказать пользователям помощь в написании) необходимых внешних схем. Кроме этого необходимо определить отображение между внешней и концептуальной схемами.

1. Определение правил безопасности и целостности.
2. Определение процедур резервного копирования и восстановления.
3. Управление производительностью и реагирование на изменяющиеся требования.

База данных состоит из некоторого набора постоянных данных, которые используются прикладными системами какого-либо предприятия. Под словом "постоянные" подразумеваются данные, которые отличаются от других, более изменчивых данных, таких, как промежуточные данные и вообще все транзитные данные. "Постоянные" данные на самом деле могут недолго оставаться таковыми, поскольку данные в БД должны отражать об изменчивых объектах реального мира и отношениях между ними.

Использование баз данных для хранения информации позволяет организовать централизованное управление данными, что обеспечивает следующие преимущества:

1. возможность сокращения избыточности;
2. возможность устранения (до некоторой степени) противоречивости;
3. возможность общего доступа к данным;
4. возможность соблюдения стандартов;
5. возможность введения ограничений для обеспечения безопасности
6. возможность обеспечения целостности данных;
7. возможность сбалансировать противоречивые требования;
8. возможность обеспечения независимости данных. Поскольку программное обеспечение отделяется от данных, хранимых БД, изменения, вносимые в структуру БД, в большинстве случаев не приводят к необходимости внесения радикальных изменений в программное обеспечение.

1.3 Уровни абстракции в СУБД. Функции абстрактных данных

Существует 3 уровня архитектуры СУБД :

1. Внутренний уровень - наиболее близкий к физическому хранению. Он связан со способами хранения информации на физических устройствах хранения;
2. Внешний уровень - наиболее близкий к пользователям. Он связан со способами представления данных для отдельных пользователей;
3. Концептуальный уровень - является промежуточным между двумя первыми. Этот уровень связан с обобщенными представлениями пользователей, в отличие от внешнего уровня, связанного с индивидуальными представлениями пользователей.

1.3.1 Внешний уровень – внешнее представление

Внешний уровень - индивидуальный уровень пользователя. Пользователь может быть как прикладным программистом, так и конечным пользователем с любым уровнем профессиональной подготовки. Каждый пользователь имеет свой язык общения с СУБД. Для программиста - это какой-либо язык программирования, для пользователя - язык запросов или язык, основанный на формах и меню. Любой из этих языков включает подязык данных, т.е. множество операторов всего языка, связанное только с объектами и операциями баз данных. Т.о. подязык данных встроен в базовый язык пользователя, который также обеспечивает на связанные с БД возможности.

Представление отдельного пользователя о БД на внешнем уровне архитектуры называют внешним представлением. Т.о. внешнее представление - это содержимое БД, каким его видит отдельный пользователь. Например, сотрудник отдела кадров видит БД

как набор записей о сотрудниках плюс набор записей о подразделениях. В общем случае внешнее представление состоит из множества экземпляров каждого типа внешней записи, которые не обязательно совпадают с хранимыми записями. Подъязык данных пользователя определен в терминах внешних записей. Каждое внешнее представление определяется средствами внешней схемы, которая, в основном, состоит из определений каждого типа записей во внешнем представлении.

1.3.2 Концептуальный уровень – концептуальное представление

Концептуальное представление - это представление всей информации БД в несколько более абстрактной форме по сравнению с физическим способом хранения данных. Концептуальное представление представляет данные такими, какими они есть на самом деле, а не такими, какими их вынужден видеть пользователь в рамках определенного языка. Концептуальное представление состоит из множества экземпляров каждого типа концептуальной записи, хотя в некоторых системах в способы концептуального представления данных могут быть другими - например, в виде объектов и связей между ними. Концептуальное представление определяется средствами концептуальной схемы, которая состоит из определений каждого типа концептуальных записей. При определении концептуальной схемы используется концептуальный язык определения данных, определения которого относятся только к содержанию информации. Концептуальное представление, т.о. обеспечивает независимость данных от способа их хранения.

1.3.3 Внутренний уровень – внутреннее представление

Внутреннее представление - это представление нижнего уровня всей БД. Оно состоит из множества экземпляров каждого типа внутренней записи. Внутренняя запись соответствует хранимой записи. Внутреннее представление не связано с физическим уровнем и в нем не рассматриваются физические записи. Внутреннее представление предполагает существование бесконечного линейного адресного пространства. Подробности отображения этого пространства на физические устройства хранения не включены в общую архитектуру из-за сильной зависимости от системы.

Внутреннее представление описывается с помощью внутренней схемы, которая описывается с помощью внутреннего языка определения данных.

Между тремя уровнями представлений имеются два уровня отображений. Отображения концептуального уровня на внутренний и внешнего уровня на концептуальный. Отображения сохраняют независимость данных случае внесения в структуру БД изменений.

1.4 Функции СУБД

1. Определение данных. СУБД должна допускать определения данных (внешние схемы, концептуальную и внутреннюю схемы, соответствующие отображения). Для этого СУБД включает в себя языковый процессор для различных языков определений данных.
2. Обработка данных. СУБД должна обрабатывать запросы пользователя на выборку, а также модификацию данных. Для этого СУБД включает в себя компоненты процессора языка обработки данных.
3. Безопасность и целостность данных. СУБД должна контролировать запросы и пресекать попытки нарушения правил безопасности и целостности.
4. Восстановление данных и дублирование. СУБД должна обеспечить восстановление данных после сбоев.

5. Словарь данных. СУБД должна обеспечить функцию словаря данных. Сам словарь можно считать системной базой данных, которая содержит данные о данных пользовательской БД, т.е. содержит определения других объектов системы. Словарь интегрирован в определяемую им БД и, поэтому, содержит описание самого себя.
6. Производительность. СУБД должна выполнять свои функции с максимальной производительностью.

1.5 Экспертные системы и базы знаний

В последнее время появилась необходимость хранения и использования слабоструктурированных данных, каковыми являются, например, человеческие знания в экспертных системах.

Экспертная система – система искусственного интеллекта, включающая знания об определенной слабо структурированной и трудно формализуемой узкой предметной области и способная предлагать и объяснять пользователю разумные решения. Экспертная система состоит из базы знаний, механизма логического вывода и подсистемы объяснений.

База знаний – семантическая модель, описывающая предметную область и позволяющая отвечать на такие вопросы из этой предметной области, ответы на которые в явном виде не присутствуют в базе. База знаний является основным компонентом интеллектуальных и экспертных систем.

Для хранения баз знаний в современных экспертных системах используются либо промышленные СУБД и специализированное промежуточное ПО, либо специализированное ПО.

В настоящем курсе основное внимание уделяется проектированию БД и организации хранения в них структурированной информации. Проектирование и создание баз знаний будет подробно рассматриваться в других курсах, связанных с изучением интеллектуальных систем.