

Министерство образования и науки Украины  
Харьковский национальный университет  
радиоэлектроники

Мазурова О.А., Широкопетлева М.С., Черепанова Ю.Ю.

ПО ЭВМ

## Системы баз данных

Программное обеспечение автоматизированных систем

2010

## Содержание

Введение.....	3
Теория.....	4
Концепция и технология баз данных.....	4
Системы баз данных.....	5
Информационные системы.....	18
Уровни моделирования ИС.....	22
Текущий контроль знаний.....	26
Информационные системы.....	26
Выводы.....	30
Словарь терминов.....	31
Перечень ссылок.....	33

## Системы баз данных

На сегодняшний день без данных не возможна ни одна деятельность; объём информации увеличивается с каждой минутой.

Без посторонней помощи человек уже не может сориентироваться в таком объеме информации. Поэтому и нужны компьютерные системы БД, которые позволяют хранить большие объемы информации и эффективно ее использовать.

БД создаются на предприятиях, организациях, малых фирмах, которые задействованы в любых сфере человеческой деятельности.

### Цели и задачи курса

Целью курса является изучение основных вопросов проектирования и организации баз данных (БД), изучение языка SQL, ознакомление с современными системами управления БД (СУБД), приобретения практических навыков проектирования, разработки и эксплуатации баз данных и информационных систем, созданных на их основе. В ходе изучения дисциплины вы должны приобрести знания и умения разработки концептуальных, логических и физических моделей данных; навыки выбора и использования средств управления информацией, которые удовлетворяют требованиям к решению поставленной задачи; знания, необходимые для выбора, обоснования и реализации наилучших решений по разработке информационных систем на основании баз данных

По результатам курса студенты должны знать:

- концепции моделирования предметных областей, которые характерны для различных моделей данных;
- структуру и принципы организации баз данных;
- этапы проектирования баз данных и информационных систем на их основе;
- средства работы с базами данных с использованием языка запросов SQL;
- структуру и принципы функционирования систем управления базами данных;
- функциональные возможности и средства использования СУБД MS Access;
- СУБД VDE Administrator и среду Delphi для разработки информационных систем.

## Теория

### Концепция и технология баз данных

#### Понятие БД и СУБД

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

Система БД включает в себя (см. рис. 1):

- данные, непосредственно сохраняемые в БД;
- аппаратное обеспечение;
- программное обеспечение;

- пользователей (прикладные программисты, конечные пользователи, администраторы данных и БД).



Рисунок 1 - Система БД

#### Информация, данные, знания

Восприятие реального мира связано с восприятием и осознанием различных явлений. С давних времен люди пытались описать эти явления, собирая о них информацию и передавая ее друг другу. Фиксация и передача информации осуществляется с помощью конкретного средства общения (с помощью естественного языка или изображений) с помощью живого человека или на конкретном носителе (камне, бумаге и т.п.). Обычно информация разделяется на данные (факты, явления, события, идеи или предметы) и их интерпретацию (семантику). С помощью естественного языка данные и их интерпретация фиксируются совместно, т.к. естественный язык

достаточно гибок для представления того и другого. Примером может служить утверждение "Стоимость билета 12.20". Здесь "12.20" - данное, а "Стоимость билета" - его семантика.

Применение ЭВМ для работы с данными обычно приводит к разделению данных и интерпретации. ЭВМ имеет дело главным образом с данными как таковыми. Большая часть интерпретирующей информации вообще не фиксируется в явной форме (ЭВМ не "знает", является ли "12.20" стоимостью билета или временем отправления транспорта). Одна из причин, по которой применение ЭВМ привело к отделению данных от интерпретации та, что ЭВМ не обладали достаточными возможностями для обработки текстов на естественном языке - основном языке интерпретации данных. Кроме того, стоимость памяти ЭВМ была первоначально весьма велика. Память использовалась для хранения самих данных, а интерпретация традиционно возлагалась на пользователя. Пользователь закладывал интерпретацию данных в свою программу, которая "знала", например, что шестое вводимое значение связано с временем отправления, а четвертое - со стоимостью билета.

### Основные понятия

**Информационные процессы** - процессы создания, сбора, обработки, накопления, хранения, поиска, распространения и потребления информации.

**Информация** - сведения о лицах, предметах, фактах, событиях и процессах независимо от способа их представления.

**Данные** - информация, отражающая определенное состояние некоторой предметной области в конкретной форме представления и содержащая лишь наиболее существенные с точки зрения целей и задач сбора и обработки информации элементы образа отражаемого фрагмента действительности.

**Знания** - закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области.

**Информационная система** - программная система, работающая на основании базы данных и обеспечивающая сбор, обработку, хранение, анализ и выдачу информации, необходимой для обеспечения управленческой деятельности и технологических процессов.

**База данных** - это компьютеризованная система, предназначенная для хранения информации и предоставления ее по требованию.

## Системы баз данных

### Препосылки появления баз 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. Становление этого направления способствовало интенсификации методов численного решения сложных математических задач, развитию класса языков программирования, ориентированных на удобную запись численных алгоритмов, становлению обратной связи с разработчиками новых архитектур ЭВМ.

Второе направление - это 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 достаточно сложную структуру. Классическими примерами информационных систем являются банковские системы, системы резервирования авиационных или железнодорожных билетов, мест в гостиницах и т.д.

На самом деле, второе направление возникло несколько позже первого. Это связано с тем, что на заре вычислительной техники компьютеры обладали ограниченными возможностями в части памяти. Понятно, что можно говорить о надежном и долговременном хранении информации только при наличии запоминающих устройств, сохраняющих информацию после выключения электрического питания. Оперативная память этим свойством обычно не обладает. В начале использовались два вида устройств внешней памяти: магнитные ленты и барабаны. При этом емкость магнитных лент была достаточно велика, но по своей физической природе они обеспечивали последовательный доступ к данным. Магнитные же барабаны (они больше всего похожи на современные магнитные диски с фиксированными головками) давали возможность произвольного доступа к данным, но были ограниченного размера.

Легко видеть, что указанные ограничения не очень существенны для чисто численных расчетов. Даже если программа должна обработать (или произвести) большой объем информации, при программировании можно продумать расположение этой информации во внешней памяти, чтобы программа работала как можно быстрее.

С другой стороны, для информационных систем, в которых потребность в текущих данных определяется пользователем, наличие только магнитных лент и барабанов неудовлетворительно. Представьте себе покупателя билета, который стоя у кассы должен дожидаться полной перемотки магнитной ленты. Одним из естественных требований к таким системам является средняя быстрота выполнения операций.

Именно требования к вычислительной технике со стороны нечисленных приложений вызвали появление съемных магнитных дисков с подвижными головками, что явилось революцией в истории вычислительной техники. Эти устройства внешней памяти обладали существенно большей емкостью, чем магнитные барабаны, обеспечивали удовлетворительную скорость доступа к данным в режиме произвольной выборки, а возможность смены дискового пакета на

устройстве позволяла иметь практически неограниченный архив данных. С появлением магнитных дисков началась история систем управления данными во внешней памяти.

Таким образом, предпосылками появления БД являются:

- появление запоминающих устройств, сохраняющих информацию после выключения электрического питания;

- смена магнитных лент и барабанов на съемные магнитные диски с подвижными головками.

### **Файловые системы и системы управления БД (СУБД)**

Решающим шагом в истории появления компьютерных БД явился переход к использованию централизованных систем управления файлами.

Файлы применяются для хранения текстовых данных: документов, текстов программ и т.д. Такие файлы обычно образуются и модифицируются с помощью различных текстовых редакторов, компиляторов (файлы - объектные модули). Структура файлов обычно очень проста: это либо последовательность записей, содержащих строки текста, либо последовательность байтов, среди которых встречаются специальные символы. С точки зрения прикладной программы файл - это именованная область внешней памяти, в которую можно записывать и из которой можно считывать данные.

Файловые системы обычно обеспечивают хранение слабо структурированной информации. 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.

Однако позже оказалось, что все достоинства систем управления файлами, тем не менее, не удовлетворяют всех потребностей информационных систем. И в истории развития БД появились системы управления базами данных (СУБД). Проведем сравнительный анализ файловых систем и СУБД с учетом следующим критериев:

- способ хранения информации;
- управление единицами информации;
- поддержка специальных языков работы с информацией;
- многопользовательский доступ к информации;
- защита информации.

### **Способ хранения информации файловыми системами**

Первая развитая файловая система была разработана фирмой IBM для ее серии 360. 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. Работать с файлами на уровне пользователя здесь было очень неудобно.



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 двигается 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.

Однако эта возможность обмениваться с магнитными дисками порциями меньше объема блока в настоящее время не используется в файловых системах. Это связано с двумя обстоятельствами. Во-первых, при выполнении обмена с диском аппаратура выполняет три основных действия: подвод головок к нужному цилиндру, поиск на дорожке нужного блока и собственно обмен с этим блоком. Из всех этих действий в среднем наибольшее время занимает первое. Поэтому существенный выигрыш в суммарном времени обмена за счет считывания или записывания только части блока получить практически невозможно. Во-вторых, для того, чтобы работать с частями блоков, файловая система должна обеспечить соответствующего размера буфера оперативной памяти, что существенно усложняет распределение оперативной памяти.

Поэтому во всех файловых системах явно или неявно выделяется некоторый базовый уровень, обеспечивающий работу с файлами, представляющими набор прямо адресуемых в адресном пространстве файла блоков. Размер этих логических блоков файла совпадает или кратен размеру физического блока диска и обычно выбирается равным размеру страницы виртуальной памяти, поддерживаемой аппаратурой компьютера совместно с операционной системой.

### **Потребности информационных систем в способах хранения информации**

Однако ситуация коренным образом отличается для информационных систем. Эти системы главным образом ориентированы 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.



Производились необходимые надстройки над файловыми системами (библиотеки программ), подобно тому, как это делается в компиляторах, редакторах и т.д.

Но поскольку информационные системы требуют сложных структур данных, эти дополнительные индивидуальные средства управления данными являлись существенной частью информационных систем и практически повторялись от одной системы к другой. Стремление выделить и обобщить общую часть информационных систем, ответственную за управление сложно структурированными данными, явилось первой побудительной причиной создания СУБД. Очень скоро стало понятно, что невозможно обойтись общей библиотекой программ, реализующей над стандартной базовой файловой системой более сложные методы хранения данных.

### **Управление единицами информации**

Для обсуждения этого вопроса рассмотрим пример. Предположим, что мы хотим реализовать простую информационную систему, поддерживающую учет сотрудников некоторой фирмы. Система должна выполнять следующие действия: выдавать списки сотрудников по отделам, поддерживать возможность перевода сотрудника из одного отдела в другой, приема на работу новых сотрудников и увольнения работающих. Для каждого отдела должна поддерживаться возможность получения имени руководителя этого отдела, общей численности отдела, общей суммы выплаченной в последний раз зарплаты и т.д. Для каждого сотрудника должна поддерживаться возможность выдачи номера удостоверения по полному имени сотрудника, выдачи полного имени по номеру удостоверения, получения информации о текущем соответствии занимаемой должности сотрудника и о размере его зарплаты.

Предположим, что мы решили основывать эту информационную систему на файловой системе и пользоваться при этом одним файлом, расширив базовые возможности файловой системы за счет специальной библиотеки функций. Поскольку минимальной информационной единицей в нашем случае является сотрудник, естественно потребовать, чтобы в этом файле содержалась одна запись для каждого сотрудника. Какие поля должна содержать такая запись? Полное имя сотрудника (СОТР\_ИМЯ), номер его удостоверения (СОТР\_НОМЕР), информацию о его соответствии занимаемой должности (для простоты, "да" или "нет") (СОТР\_СТАТ), размер зарплаты (СОТР\_ЗАРП), номер отдела (СОТР\_ОТД\_НОМЕР). Поскольку мы хотим ограничиться одним файлом, та же запись должна содержать имя руководителя отдела (СОТР\_ОТД\_РУК).

Функции нашей информационной системы требуют, чтобы обеспечивалась возможность многоключевого доступа к этому файлу по уникальным ключам (недублируемым в разных записях) СОТР\_ИМЯ и СОТР\_НОМЕР. Кроме того, должна обеспечиваться возможность выбора всех записей с общим значением СОТР\_ОТД\_НОМЕР, то есть доступ по неуникальному ключу. Для того, чтобы получить численность отдела или общий размер зарплаты, каждый раз при выполнении такой функции информационная система должна будет выбрать все записи о сотрудниках отдела и посчитать соответствующие общие значения.

Таким образом, даже для такой простой системы ее реализация на базе файловой системы, во-первых, требует создания достаточно сложной надстройки для многоключевого доступа к

файлам, и, во-вторых, вызывает требование существенной избыточности хранения (для каждого сотрудника одного отдела повторяется имя руководителя) и выполнение массовой выборки и вычислений для получения суммарной информации об отделах. Кроме того, если в ходе эксплуатации системы нам захочется, например, выдавать списки сотрудников, получающих заданную зарплату, то придется либо полностью просматривать файл, либо реструктуризовывать его, объявляя ключевым поле СОТР\_ЗАРП.

Первое, что приходит на ум, - это поддерживать два многоключевых файла: СОТРУДНИКИ и ОТДЕЛЫ. Первый файл должен содержать поля СОТР\_ИМЯ, СОТР\_НОМЕР, СОТР\_СТАТ, СОТР\_ЗАРП и СОТР\_ОТД\_НОМЕР, а второй - ОТД\_НОМЕР, ОТД\_РУК, ОТД\_СОТР\_ЗАРП (общий размер зарплаты) и ОТД\_РАЗМЕР (общее число сотрудников в отделе). Большинство неудобств, перечисленных в предыдущем абзаце, будут преодолены. Каждый из файлов будет содержать только недублируемую информацию, необходимости в динамических вычислениях суммарной информации не возникает. Но заметим, что при таком переходе наша информационная система должна обладать некоторыми новыми особенностями, сближающими ее с СУБД.

Прежде всего, система должна теперь знать, что она работает с двумя информационно связанными файлами (это шаг в сторону схемы базы данных), должна знать структуру и смысл каждого поля (например, что СОТР\_ОТД\_НОМЕР в файле СОТРУДНИКИ и ОТД\_НОМЕР в файле ОТДЕЛЫ означают одно и то же), а также понимать, что в ряде случаев изменение информации в одном файле должно автоматически вызывать модификацию во втором файле, чтобы их общее содержимое было согласованным. Например, если на работу принимается новый сотрудник, то необходимо добавить запись в файл СОТРУДНИКИ, а также соответствующим образом изменить поля ОТД\_ЗАРП и ОТД\_РАЗМЕР в записи файла ОТДЕЛЫ, описывающей отдел этого сотрудника.

Понятие согласованности данных является ключевым понятием баз данных. Фактически, если информационная система поддерживает согласованное хранение информации в нескольких файлах, можно говорить о том, что она поддерживает базу данных. Если же некоторая вспомогательная система управления данными позволяет работать с несколькими файлами, обеспечивая их согласованность, можно назвать ее системой управления базами данных. Уже только требование поддержания согласованности данных в нескольких файлах не позволяет обойтись библиотекой функций: такая система должна иметь некоторые собственные данные (метаданные) и даже знания, определяющие целостность данных.

### **Поддержка специальных языков работы с информацией**

Даже в нашем примере неудобно реализовывать такие запросы как "выдать общую численность отдела, в котором работает Петр Иванович Сидоров". Было бы гораздо проще, если бы система позволяла сформулировать такой запрос на близком пользователям языке. Такие языки называются языками запросов к базам данных.

Например: есть база данных о сотрудниках и их заданиях (см. рис.2)

Сотрудники			Задания		
Н_СОТР	ФАМ	Н_ОТД	Н_СОТР	Н_ПРО	Н_ЗАДАН
1	Иванов	1	1	1	1
2	Петров	1	1	2	1
3	Сидоров	2	2	1	2
			3	1	3
			3	2	2

Рисунок 2 - Фрагмент базы данных сотрудников

На языке SQL запрос можно было бы выразить в форме

```
SELECT ФАМ, Н_ЗАДАН FROM СОТРУДНИКИ, ЗАДАНИЯ
WHERE (СОТРУДНИКИ.Н_СОТР=ЗАДАНИЯ.Н_СОТР) AND (ЗАДАНИЯ.Н_ПРО=1);
```

На рис. 3 приведен результат выполнения запроса

ФАМ	Н_ЗАДАН
Иванов	1
Петров	2
Сидоров	3

Рисунок 3 - Результат выполнения запроса

Таким образом, при формулировании запроса СУБД позволит не задумываться о том, как будет выполняться этот запрос. Среди ее метаданных будет содержаться информация о том,

что поле СОТР\_ИМЯ является ключевым для файла СОТРУДНИКИ, а ОТД\_НОМЕР - для файла ОТДЕЛЫ, и система сама воспользуется этим.

### **Режим многопользовательского доступа**

Если операционная система поддерживает многопользовательский режим, вполне реальна ситуация, когда два или более пользователей одновременно пытаются работать с одним и тем же файлом. Если все эти пользователи собираются только читать файл, ничего страшного не произойдет. Но если хотя бы один из них будет изменять файл, для корректной работы этой группы требуется взаимная синхронизация.

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 (процесса), 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.

Таким образом, если опираться только на использование файлов, то для обеспечения корректности на все время модификации любого из двух файлов доступ других пользователей к этому файлу будет блокирован. Т.е., например, зачисление на работу Петра Ивановича Сидорова существенно затормозит получение информации о сотруднике Иване Сидоровиче Петрове, даже если они будут работать в разных отделах. Настоящие СУБД обеспечивают гораздо более тонкую синхронизацию параллельного доступа к данным.

### **Защита информации**

Далее, представьте себе, что в нашей первоначальной реализации информационной системы, основанной на использовании библиотек расширенных методов доступа к файлам, обрабатывается операция регистрации нового сотрудника. Следуя требованиям согласованного изменения файлов, информационная система вставила новую запись в файл СОТРУДНИКИ и собиралась модифицировать запись файла ОТДЕЛЫ, но именно в этот момент произошло аварийное выключение питания. Очевидно, что после перезапуска системы ее база данных будет находиться в рассогласованном состоянии. Потребуется выяснить это (а для этого нужно явно проверить соответствие информации с файлах СОТРУДНИКИ и ОТДЕЛЫ) и привести информацию в согласованное состояние. Настоящие СУБД берут такую работу на себя. Прикладная система не обязана заботиться о корректности состояния базы данных.

На рис. 4 приведено краткое сравнение файловых систем и СУБД.

Файловые системы	Системы управления БД
<u>Способ хранения информации</u>	
Хранится неструктурированная информация	Необходимо хранить сложно структурированную информацию
<u>Управление единицами хранения информации</u>	
В каждом файле хранится отдельная информация	Необходимо поддерживать согласованность информации в файлах
<u>Поддержка специальных языков работы с информацией</u>	
Не поддерживаются специальные языки	Реализована поддержка языков запросов к данным
<u>Многопользовательский доступ к информации</u>	
Не поддерживается реальная параллельная работа пользователей	Поддерживается реальная параллельная работа пользователей
<u>Защита информации</u>	
Не обеспечивается восстановление информации после сбоев	Поддерживается восстановление информации после мягких и жестких сбоев

Рисунок 4 - Сравнение файловых систем и СУБД

Таким образом, СУБД решают множество проблем, которые затруднительно или вообще невозможно решить при использовании файловых систем. При этом существуют приложения, для которых вполне достаточно файлов; приложения, для которых необходимо решать, какой уровень работы с данными во внешней памяти для них требуется, и приложения, для которых безусловно нужны базы данных.

Итак, существует несколько потребностей, которые не покрываются возможностями систем управления файлами: поддержание логически согласованного набора файлов; обеспечение языка манипулирования данными; восстановление информации после разного рода сбоев; реально параллельная работа нескольких пользователей. Можно считать, что если прикладная информационная система опирается на некоторую систему управления данными, обладающую этими свойствами, то эта система управления данными является системой управления базами данных (СУБД).

**СУБД** - система управления данными, обеспечивающая следующие функциональные возможности: поддержание логически согласованного набора файлов; обеспечение языка манипулирования данными; восстановление информации после разного рода сбоев; реально параллельная работа нескольких пользователей.

### Основные функции СУБД

Организация типичной СУБД и состав ее компонентов соответствует набору функций СУБД:

- управление данными во внешней памяти;

- управление буферами оперативной памяти;
- управление транзакциями;
- журнализация и восстановление БД после сбоев;
- поддержание языков БД.

Рассмотрим подробнее эти функции:

### **Управление данными во внешней памяти**

Эта функция включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для ускорения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователи в любом случае не обязаны знать, использует ли СУБД файловую систему, и если использует, то как организованы файлы. В частности, СУБД поддерживает собственную систему именования объектов БД.

### **Управление буферами оперативной памяти**

СУБД обычно работают с БД значительного размера; по крайней мере этот размер обычно существенно больше доступного объема оперативной памяти. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. При этом, даже если операционная система производит общесистемную буферизацию (как в случае ОС UNIX), этого недостаточно для целей СУБД, которая располагает гораздо большей информацией о полезности буферизации той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов.

#### **▼ Подробнее**

Заметим, что существует отдельное направление СУБД, которое ориентировано на постоянное присутствие в оперативной памяти всей БД. Это направление основывается на предположении, что в будущем объем оперативной памяти компьютеров будет настолько велик, что позволит не беспокоиться о буферизации. Пока эти работы находятся в стадии исследований

### **Управление транзакциями**

**Транзакция** - это последовательность операций над БД, рассматриваемых СУБД как единое



целое. Либо транзакция успешно выполняется, и СУБД фиксирует (COMMIT) изменения БД, произведенные этой транзакцией, во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД. Если вспомнить наш пример информационной системы с файлами СОТРУДНИКИ и ОТДЕЛЫ, то единственным способом не нарушить целостность БД при выполнении операции приема на работу нового сотрудника является объединение элементарных операций над файлами СОТРУДНИКИ и ОТДЕЛЫ в одну транзакцию. Таким образом, поддержание механизма транзакций является обязательным условием даже однопользовательских СУБД (если, конечно, такая система заслуживает названия СУБД). Но понятие транзакции гораздо более важно в многопользовательских СУБД. То свойство, что каждая транзакция начинается при целостном состоянии БД и оставляет это состояние целостным после своего завершения, делает очень удобным использование понятия транзакции как единицы активности пользователя по отношению к БД. При соответствующем управлении параллельно выполняющимися транзакциями со стороны СУБД каждый из пользователей может в принципе ощущать себя единственным пользователем СУБД (на самом деле, это несколько идеализированное представление, поскольку в некоторых случаях пользователи многопользовательских СУБД могут ощутить присутствие своих коллег).

### Журнализация

Одним из основных требований к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев: так называемые мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания), и жесткие сбои, характеризующиеся потерей информации на носителях внешней памяти. Примерами программных сбоев могут быть: аварийное завершение работы СУБД (по причине ошибки в программе или в результате некоторого аппаратного сбоя) или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается незавершенной. Первую ситуацию можно рассматривать как особый вид мягкого аппаратного сбоя; при возникновении последней требуется ликвидировать последствия только одной транзакции.

Понятно, что в любом случае для восстановления БД нужно располагать некоторой дополнительной информацией. Другими словами, поддержание надежности хранения данных в БД требует избыточности хранения данных, причем та часть данных, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД.

**Журнал** - это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. В разных СУБД изменения БД журналируются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), иногда - минимальной внутренней операции модификации

страницы внешней памяти; в некоторых системах одновременно используются оба подхода.

### **Поддержка языков БД**

Для работы с базами данных используются специальные языки, в целом называемые **языками баз данных**. В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language). Язык SQL позволяет определять схему реляционной БД и манипулировать данными. При этом именование объектов БД (для реляционной БД - именование таблиц и их столбцов) поддерживается на языковом уровне в том смысле, что компилятор языка SQL производит преобразование имен объектов в их внутренние идентификаторы на основании специально поддерживаемых служебных таблиц-каталогов. Внутренняя часть СУБД (ядро) вообще не работает с именами таблиц и их столбцов.

Язык SQL содержит специальные средства определения ограничений целостности БД. Опять же, ограничения целостности хранятся в специальных таблицах-каталогах, и обеспечение контроля целостности БД производится на языковом уровне, т.е. при компиляции операторов модификации БД компилятор SQL на основании имеющихся в БД ограничений целостности генерирует соответствующий программный код. Специальные операторы языка SQL позволяют определять так называемые представления БД, фактически являющиеся хранимыми в БД запросами (результатом любого запроса к реляционной БД является таблица) с именованными столбцами. Наконец, авторизация доступа к объектам БД производится также на основе специального набора операторов SQL. Идея состоит в том, что для выполнения операторов SQL разного вида пользователь должен обладать различными полномочиями. Пользователь, создавший таблицу БД, обладает полным набором полномочий для работы с этой таблицей.

### **Типовая организация современной СУБД**

Логически в современной реляционной СУБД можно выделить внутреннюю часть - ядро СУБД (часто его называют Data Base Engine), компилятор языка БД (обычно SQL), подсистему поддержки времени выполнения, набор утилит (см. рис. 5).



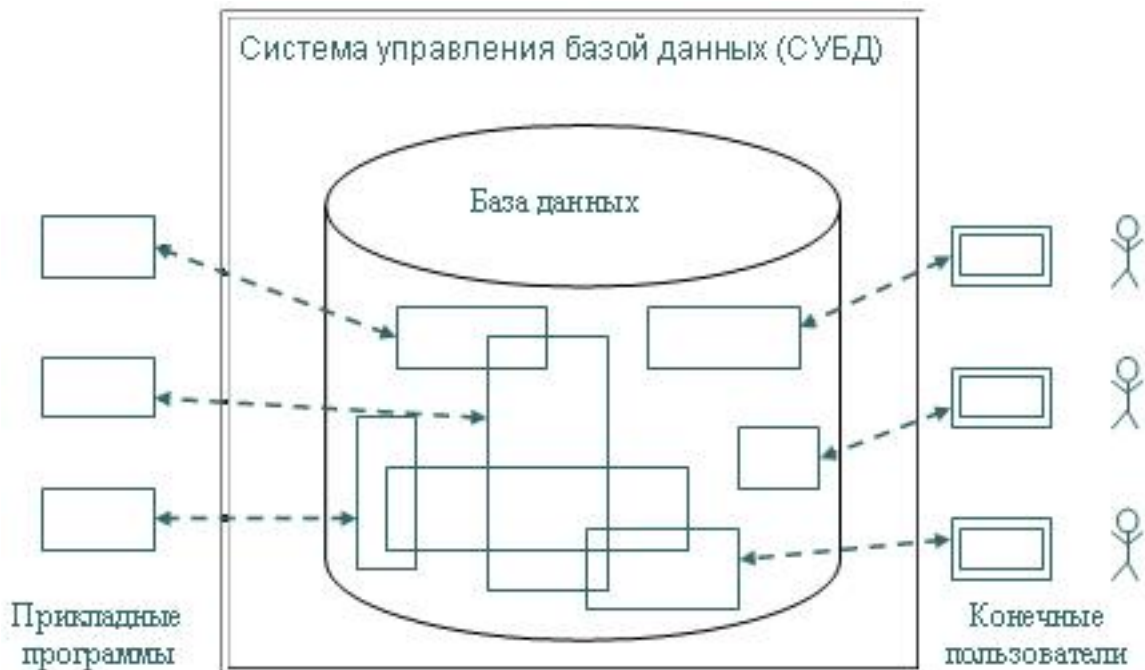


Рисунок 5 - Упрощенная схема информационной системы

Ядро СУБД отвечает за управление данными во внешней памяти, управление буферами оперативной памяти, управление транзакциями и журнализацию. Соответственно, можно выделить такие компоненты ядра, как менеджер данных, менеджер буферов, менеджер транзакций и менеджер журнала. Ядро СУБД обладает собственным интерфейсом, не доступным пользователям напрямую и используемым в программах, производимых компилятором SQL (или в подсистеме поддержки выполнения таких программ) и утилитах БД. Ядро СУБД является основной резидентной частью СУБД. При использовании архитектуры "клиент-сервер" ядро является основной составляющей серверной части системы.

Основной функцией компилятора языка БД является компиляция операторов языка БД в некоторую выполняемую программу. Основной проблемой реляционных СУБД является то, что языки этих систем (а это, как правило, SQL) являются непроцедурными, т.е. в операторе такого языка специфицируется некоторое действие над БД, но эта спецификация не является процедурой, а лишь описывает в некоторой форме условия совершения желаемого действия. Поэтому компилятор должен решить, каким образом выполнять оператор языка прежде, чем произвести программу. Результатом компиляции является выполняемая программа, представляемая в некоторых системах в машинных кодах, но более часто в выполняемом внутреннем машинно-независимом коде. В последнем случае реальное выполнение оператора производится с привлечением подсистемы поддержки времени выполнения, представляющей собой, по сути дела, интерпретатор этого внутреннего языка.

Наконец, в отдельные утилиты БД обычно выделяют такие процедуры, которые слишком накладно выполнять с использованием языка БД, например, загрузка и выгрузка БД, сбор

статистики, глобальная проверка целостности БД и т.д. Утилиты программируются с использованием интерфейса ядра СУБД, а иногда даже с проникновением внутрь ядра.

Преимущества БД по сравнению с традиционными бумажными технологиями:

- компактность (сравните память персонального компьютера и многокомнатный архив отдела кадров, которые хранят БД по сотрудникам предприятия);
- скорость (доступ к компьютерной информации намного быстрее, чем поиск по всевозможным бумажным журналам и книгам записи);
- низкие трудозатраты (для работы с компьютерной БД достаточно одного человека);
- применимость (компьютерная БД легко доступна для любого отдела предприятия).

Основное отличие компьютерных БД от бумажных технологий то, что компьютерные данные являются:

- интегрированными, т.е. данные хранятся как объединение нескольких отдельных файлов данных, полностью или частично не перекрывающихся;
- общими, т.е. одни и те же области БД могут использоваться одновременно (одновременный доступ) несколькими пользователями.

## Информационные системы

**Информационная система** (ИС) 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.

### Аппаратное обеспечение ИС

- накопители для хранения информации (диски с перемещаемыми головками) вместе с устройствами ввода-вывода, контроллерами устройств, каналами ввода-вывода;
- процессор вместе с оперативной памятью, которая используется для работы программного обеспечения.

### Программное обеспечение ИС

- система управления базами данных (database management system (DBMS)) или диспетчер базы данных (database manager);

- утилиты восстановления БД, копирования БД и т.д.;
- структурированный язык запросов SQL;
- прикладное программное обеспечение.

### Архитектура информационной системы

Архитектура ИС может быть разделена на три основных уровня: внутренний, концептуальный и внешний. Внутренний уровень наиболее близок к физической памяти, т.е. связан со способом фактического хранения данных; внешний уровень наиболее близок к пользователям, т.е. связан с тем, как отдельные пользователи представляют себе эти данные; концептуальный уровень есть между двумя другими. Если внешний уровень связан с частными представлениями пользователей, то концептуальный уровень можно представить себе определяющим обобщенное представление пользователей (см.рис.6).



Рисунок 6 - Архитектура ИС

Теперь рассмотрим различные компоненты системы более подробно.

Пользователями системы являются либо прикладные программисты, либо пользователи, работающие с удаленных терминалов и имеющие разный уровень профессиональной подготовки. Каждый пользователь обеспечивается рабочей областью, которая используется для приема или передачи данных между пользователем и базой данных.

Как уже отмечалось, отдельного пользователя, интересует только часть всей БД, кроме того, представление пользователя об этой части базы будет до некоторой степени абстрактным, в сравнении с тем как данные физически хранятся. Пользователь представляет БД посредством внешней модели. Внешняя модель, таким образом, является информационным содержанием БД в том виде, в каком его представляет конкретный пользователь. Вообще говоря, внешняя модель состоит из различных экземпляров различных типов внешних записей. Каждая внешняя модель определяется посредством внешней схемы, которая в основном состоит из описаний всех типов

внешних записей этой внешней модели.

Теперь обратимся к концептуальному уровню. Концептуальная модель есть представление полного информационного содержания БД в несколько абстрактной форме по сравнению со способом физического хранения данных. Концептуальная модель состоит из множества экземпляров различных типов концептуальных записей. Концептуальная запись не обязана быть такой же, как внешняя запись или как хранимая запись. Концептуальная модель определяется посредством концептуальной схемы, которая включает определение каждого типа концептуальных записей.

Таким образом, концептуальная модель есть представление общего содержания БД, а концептуальная схема-определение этого представления.

Третьим уровнем архитектуры является внутренний уровень. Внутренняя модель есть представление самого низкого уровня всей БД; он состоит из различных экземпляров типов внутренних записей.

Термин "внутренняя запись" применяется для конструкции, которую мы называем хранимой записью. Таким образом, внутренняя модель является еще одним шагом в сторону от физического уровня, так как она не строится в терминах физических записей или блоков. Внутренняя модель описывается посредством внутренней схемы, которая не только определяет различные типы хранимых записей, но и то, какие индексы существуют, как представлены хранимые поля, какова физическая последовательность хранимых записей и т.д.

## Пользователи ИС

Естественно, что проект информационной системы надо начинать с анализа предметной области и выявления требований к ней отдельных пользователей (сотрудников организации, для которых создается база данных). Подробнее этот процесс будет рассмотрен ниже, а здесь отметим, что проектирование обычно поручается человеку (группе лиц) - *администратору базы данных (АБД)*. Им может быть как специально выделенный сотрудник организации, так и будущий пользователь ИС, достаточно хорошо знакомый с машинной обработкой данных.

Объединяя частные представления о содержимом базы данных, полученные в результате опроса пользователей, и свои представления о данных, которые могут потребоваться в будущих приложениях, АБД сначала создает обобщенное неформальное описание создаваемой базы данных. Это описание, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающим над проектированием базы данных, называют *инфологической моделью данных*.

Такая человеко-ориентированная модель полностью независима от физических параметров среды хранения данных. В конце концов этой средой может быть память человека, а не ЭВМ. Поэтому инфологическая модель не должна изменяться до тех пор, пока какие-то изменения в реальном мире не потребуют изменения в ней некоторого определения, чтобы эта модель продолжала отражать предметную область.

Таким образом, пользователями ИС являются:

- прикладные программисты - отвечают за написание прикладных программ, использующих БД (используют языки программирования C++ (Builder), Pascal (Delphi) и т.п.);

- конечные пользователи - работают с системами БД непосредственно через прикладное приложение или через встроенное приложение (процессор языка запросов);

- администраторы:

1) администратор данных (АД) - владеет сведениями об информации, необходимой для описания объектов или процессов предметной области; определяет группу лиц, имеющих доступ к БД, их полномочия. Т.е. это нетехнический специалист, хорошо разбирающийся в предметной области;

2) администратор базы данных (АБД) - обладает техническими знаниями об аппаратном и программном обеспечении БД; отвечает за поддержание целостности БД, ее восстановление и своевременное копирование, обеспечивает технические аспекты защиты данных от несанкционированного доступа.

### **Жизненный цикл ИС**

В общем случае, **жизненный цикл ИС** можно представить следующим образом:

- проектирование («бумажная» стадия жизни системы) - выполняется посредством изучения предметной области и требований, предъявляемых к информационной системе. Производится выбор:

- 1) структуры данных и стратегии их хранения в памяти ЭВМ;
- 2) технологии обслуживания ИС и взаимодействия с ней конечных пользователей;
- 3) технических и стандартных программных средств;

- реализация - материализация проекта, т.е. перенесение его в память ЭВМ. На этапе выполняются:

- 1) разрабатывается и отлаживается программное обеспечение ИС;
- 2) создается отладочная БД;
- 3) тестируется и корректируется технология обслуживания ИС;

- эксплуатация - начинается с наполнения системы реальной информацией. Включает также:

- 1) обеспечение защиты данных;

- 2) организация коллективного использования данных;
- 3) анализ и управление эффективностью системы;
- 4) совершенствование и последующее развитие ИС.

## Уровни моделирования ИС

### Предметная область

При выполнении основных функций СУБД должна использовать различные описания данных. Рассмотрим порядок создания этих описаний.

При разработке базы данных (ИС) обычно выделяется несколько уровней моделирования, при помощи которых происходит переход от предметной области к конкретной реализации базы данных средствами конкретной СУБД. Можно выделить следующие уровни: сама предметная область, концептуальная модель предметной области, логическая модель данных, физическая модель данных, собственно база данных и приложения.

Предметная область - это часть реального мира, данные о которой мы хотим отразить в базе данных. Например, в качестве предметной области можно выбрать бухгалтерию какого-либо предприятия, отдел кадров, банк, магазин и т.д. Предметная область бесконечна и содержит как существенно важные понятия и данные, так и малозначащие или вообще не значащие данные. Так, если в качестве предметной области выбрать учет товаров на складе, то понятия "накладная" и "счет-фактура" являются существенно важными понятиями, а то, что сотрудница, принимающая накладные, имеет двоих детей - это для учета товаров неважно. Однако, с точки зрения отдела кадров данные о наличии детей являются существенно важными. Таким образом, важность данных зависит от выбора предметной области.

### Концептуальная модель предметной области

**Концептуальная модель предметной области** - это наши знания о предметной области в виде понятий (концептов). Знания могут быть как в виде неформальных знаний в мозгу эксперта, так и выражены формально при помощи каких-либо средств. В качестве таких средств могут выступать текстовые описания предметной области, наборы должностных инструкций, правила ведения дел в компании и т.п. Опыт показывает, что текстовый способ представления модели предметной области крайне неэффективен. Гораздо более информативными и полезными при разработке баз данных являются описания предметной области, выполненные при помощи специализированных графических нотаций. Имеется большое количество методик описания предметной области. Концептуальная модель БД - отражает информационное содержание данных, как основных понятий и отношений между ними. Концептуальная модель не затрагивает физического состояния данных, в том числе архитектуры данных, методов доступа, форматов

физических данных.

На рис. 7 приведен фрагмент концептуальной модели предметной области "Предприятие".

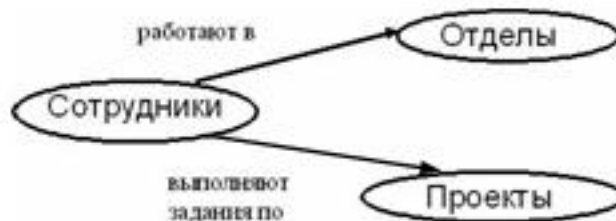


Рисунок 7 - Концептуальная модель ИС "Предприятие"

Из наиболее известных методик исследования предметных областей и построения концептуальных моделей можно назвать системный анализ. Также существует целый ряд методик, учитывающих принципы системного анализа, - методика структурного анализа SADT и основанная на нем IDEF0, диаграммы потоков данных Гейна-Сарсона, методика объектно-ориентированного анализа UML, и др. Концептуальная модель предметной области описывает скорее процессы, происходящие в предметной области и данные, используемые этими процессами. От того, насколько правильно смоделирована предметная область, зависит успех дальнейшей разработки приложений.

**Модель данных** - инструментарий для отображения предметной области, определяется:

- допустимой организацией данных;
- ограничениями целостности (семантикой);
- множеством операций, допустимых над объектами модели данных.

### Логическая модель данных

На следующем, более низком уровне находится логическая модель данных предметной области.

Логическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью. Примеры понятий - "сотрудник", "отдел", "проект", "зарплата". Примеры взаимосвязей между понятиями - "сотрудник числится ровно в одном отделе", "сотрудник может выполнять несколько проектов", "над одним проектом может работать несколько сотрудников". Примеры ограничений - "возраст сотрудника не менее 16 и не более 60 лет".

Можно выделить три основные виде логических моделей:



- иерархическую модель;
- сетевую модель;
- реляционную модель.

Логическая модель данных для СУБД реляционного типа представляет собой схему базы данных, приведенную на рисунке 8



Рисунок 8 - Пример логической модели данных

Логическая модель данных является начальным прототипом будущей базы данных. Логическая модель строится в терминах информационных единиц, но *без привязки к конкретной СУБД*. Предварительным средством разработки логической модели данных в настоящий момент являются различные варианты инфологических (информационно-логических) моделей - **ER-диаграмма (Entity-Relationship, диаграммы сущность-связь)**. Одну и ту же ER-модель можно преобразовать как в реляционную модель данных, так и в модель данных для иерархических и сетевых СУБД, или в постреляционную модель данных.

Решения, принятые на предыдущем уровне, при разработке инфологической модели предметной области, определяют некоторые границы, в пределах которых можно развивать логическую модель данных, в пределах же этих границ можно принимать различные решения.

Для логической модели данных характерно то, что выполняя все основные требования, предъявляемые СУБД, не поддерживается ориентация на конкретную СУБД, что реализуется в физической модели данных.

### Физическая модель данных



На еще более низком уровне находится физическая модель данных.

**Физическая модель данных** описывает данные средствами конкретной СУБД. Ограничения, имеющиеся в логической модели данных, реализуются различными средствами СУБД, например, при помощи индексов, декларативных ограничений целостности, триггеров, хранимых процедур. При этом опять-таки решения, принятые на уровне логического моделирования определяют некоторые границы, в пределах которых можно развивать физическую модель данных. Точно также, в пределах этих границ можно принимать различные решения. Например, отношения, содержащиеся в логической модели данных, должны быть преобразованы в таблицы, но для каждой таблицы можно дополнительно объявить различные индексы, повышающие скорость обращения к данным. Многое тут зависит от конкретной СУБД.

Если физическая модель данных реализована средствами реляционной СУБД, то отношения, разработанные на стадии формирования логической модели данных, преобразуются в таблицы, атрибуты становятся столбцами таблиц, для ключевых атрибутов создаются уникальные индексы, домены преобразуются в типы данных, принятые в конкретной СУБД.

Собственно база данных и информационная система. И, наконец, как результат предыдущих этапов появляется собственно сама база данных. База данных реализована на конкретной программно-аппаратной основе, и выбор этой основы позволяет существенно повысить скорость работы с базой данных. Например, можно выбирать различные типы компьютеров, менять количество процессоров, объем оперативной памяти, дисковые подсистемы и т.п. Очень большое значение имеет также настройка СУБД в пределах выбранной программно-аппаратной платформы.

Но опять решения, принятые на предыдущем уровне - уровне физического проектирования, определяют границы, в пределах которых можно принимать решения по выбору программно-аппаратной платформы и настройки СУБД. Таким образом, ясно, что решения, принятые на каждом этапе моделирования и разработки базы данных, будут сказываться на дальнейших этапах. Поэтому особую роль играет принятие правильных решений на ранних этапах моделирования.

## Текущий контроль знаний

### Информационные системы

#### ИС

На этапе проектирования ИС выполняются следующие шаги:

- Выбор структуры данных и стратегии их хранения.
- Выбор технологии обслуживания ИС и взаимодействия с ней конечных пользователей.
- Выбор технических и стандартных программных средств.
- Все вышеперечисленное

#### ИС

Укажите правильную последовательность для следующих элементов жизненного цикла ИС.

	Реализация - материализация проекта.
	Эксплуатация
	Проектирование

#### ИС

Компьютеризованная система, предназначенная для хранения информации и предоставления ее по требованию. что это?

- Информационная система.

- Система управления базами данных.
- База данных.

## ИС

Что такое информационная система?

- Компьютеризованная система, предназначенная для хранения информации и предоставления ее по требованию
- Программный комплекс, обеспечивающий следующие функциональные возможности: поддержание логически согласованного набора файлов; обеспечение языка манипулирования данными; восстановление информации после разного рода сбоев; реально параллельная работа нескольких пользователей.
- 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

## ИС

Какие признаки характерны для файловых систем?

- Хранение отдельной самостоятельной информации в каждом файле.
- Поддержка языков запросов к БД
- Невозможность реальной параллельной работы нескольких пользователей.
- Хранение неструктурированной информации.
- Поддержка восстановления информации после разного рода сбоев.
- Необходимость хранения сложно структурированной информации.

## ИС

Определите корректную последовательность уровней моделирования.

Концептуальная модель
Предметная область
Физическая модель
База данных и приложения
Логическая модель

### ИС

Какие признаки характерны для систем управления БД?

- Поддержка согласованности информации в файлах
- Поддержка языков запросов к БД
- Необходимость хранения сложно структурированной информации.
- Хранение неструктурированной информации.
- Параллельная работа нескольких пользователей не поддерживается.
- Хранение отдельной самостоятельной информации в каждом файле.

### ИС

Что такое данные?

- Информация, отражающая определенное состояние некоторой предметной области в конкретной форме представления и содержащая лишь наиболее существенные с точки зрения целей и задач сбора и обработки информации элементы образа отражаемого фрагмента действительности
- Закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области

- Сведения о лицах, предметах, фактах, событиях и процессах, независимо от способа их представления.

## ИС

Администратор БД - это пользователь ИС, который ...

- владеет сведениями об информации, необходимой для описания объектов или процессов предметной области; определяет группу лиц, имеющих доступ к БД, их полномочия
- обладает техническими знаниями об аппаратном и программном обеспечении БД; отвечает за поддержание целостность БД, ее восстановление и своевременное копирование, обеспечивает технические аспекты защиты данных от несанкционированного доступа
- работает с системами БД непосредственно через прикладное приложение или через встроенное приложение
- отвечает за написание прикладных программ, использующих БД

## ИС

Что такое предметная область?

- Область, ограниченная предметами.
- Чувств реального мира, данные о которой мы хотим отразить в БД.
- Область жизни и деятельности человека

## **Выводы по теме "Системы баз данных"**

В данном разделе были рассмотрены основные расхождения между файловыми системами и системами управления базами данных; базовые функции и приводится типичная организация СУБД и информационных систем, как систем, основанных на базах данных; введены основные понятия реляционных баз данных, рассмотрены уровни моделирования при проектировании информационных систем и основные компоненты и организацию ИС.

## Словарь терминов

### База данных

Компьютеризованная система, предназначенная для хранения информации и предоставления ее по требованию

### Данные

отдельные факты, характеризующие объекты, процессы и явления предметной области, а также их свойства.

### Журнал

это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью, в которую поступают записи обо всех изменениях основной части БД.

### Запрос

команда, которая адресуется Базе Данных для вывода требуемой информации из таблиц.

команда, которую вы даете вашей программе базы данных, и которая сообщает ей, чтобы она вывела определенную информацию из таблиц в память.

### Знания

закономерности предметной области (принципы, связи, законы), полученные в результате практической деятельности и профессионального опыта, позволяющие специалистам ставить и решать задачи в этой области

### Информационная система

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.

### **Информационные процессы**

процессы создания, сбора, обработки, накопления, хранения, поиска, распространения и потребления информации.

### **Информация**

сведения о лицах, предметах, фактах, событиях и процессах независимо от способа их представления.

### **Модель предметной области**

это наши знания о предметной области. Модель предметной области описывает скорее процессы, происходящие в предметной области и данные, используемые этими процессами. От того, насколько правильно смоделирована предметная область, зависит успех дальнейшей разработки приложений.

### **Предметная область**

часть реального мира, данные о которой мы хотим отразить в базе данных.

### **Реляционная база данных**

набор отношений, имена которых совпадают с именами схем отношений в схеме БД.

### **СУБД**

система управления данными, обеспечивающая следующие функциональные возможности: поддержание логически согласованного набора файлов; обеспечение языка манипулирования данными; восстановление информации после разного рода сбоев; реально параллельная работа нескольких пользователей.

Смотрите также:

система управления базами данных



## **Перечень ссылок**

### **Источники, использованные в материалах**

Кириллов В.В. Основы проектирования реляционных баз данных . СПб. Санкт-Петербургский Государственный институт точной механики и оптики. . -с.

### **Основная литература**

Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки данных. Вводный курс. Москва. Гелиос АРБ. 2002. -368с.