

**Московский государственный университет экономики,  
статистики и информатики  
Московский государственный институт эконометрики,  
информатики, финансов и права**

---

**Диго С.М.**

## **Введение в банки данных**

**Учебное пособие по курсу  
“Базы данных”**

**Москва, 2002**

УДК 004.65  
ББК 018\*32.973  
Д 44

Диго С.М. Введение в банки данных (Учебное пособие по курсу “Базы данных”)  
/М. Московский международный институт эконометрики, информатики,  
финансов и права. 2002.

© Диго Светлана Михайловна, 2002.

© Московский международный институт эконометрики, информатики,  
финансов и права, 2002

## Содержание

1. Понятие банка данных .....	4
1.1. Основные понятия .....	4
1.2. Преимущества БнД.....	5
1.3. Пользователи БнД.....	6
1.4. Предпосылки широкого использования БнД.....	9
1.5. Требования к БнД .....	9
1.6. Недостатки БнД .....	10
2. Компоненты банка данных.....	11
2.1. Информационная компонента.....	11
2.2. Программные средства БнД .....	13
2.3. Языковые средства БнД .....	14
2.4. Технические средства БнД.....	19
2.5. Организационно-методические средства.....	23
2.6. Администраторы банка данных.....	23
2.7. Взаимодействие компонентов БнД.....	27
3. Классификация банков данных.....	29
3.1. Классификация баз данных .....	29
3.2. Классификация СУБД .....	37
3.3. Классификационные группировки, относящиеся к БнД в целом.....	40
4. Уровни моделей и этапы проектирования бд.....	43
4.1. Уровни моделей .....	43
4.2. Взаимосвязь этапов проектирования БД.....	45
4.3. Факторы, влияющие на проектирование БД.....	47
Контрольные вопросы .....	49
5. Литература.....	51

# 1. Понятие банка данных

## 1.1. Основные понятия

Банк данных (БнД) является современной формой организации хранения и доступа к информации. "Банк данных – это система специальным образом организованных данных (баз данных), программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных." [5]

В данном выше определении БнД, с одной стороны подчеркивается, что банк данных является сложной системой, включающей в себя все обеспечивающие подсистемы, необходимые для функционирования любой системы автоматизированной обработки данных.

С другой стороны, в этом определении также обозначены и основные *отличительные особенности банков данных*:

- Базы данных создаются обычно не для решения какой-либо одной задачи для одного пользователя, а для многоцелевого использования.

- Базы данных отражают определенную часть реального мира. Надо стремиться, чтобы вся информация, описывающая предметную область, фиксировалась в базе данных однократно, накапливалась и поддерживалась в актуальном состоянии централизованно, а все пользователи, которым эта информация нужна, должны иметь возможность работать с ней.

- Базы данных – это специальным образом организованные данные. Эти особенности в организации данных заключаются, прежде всего, в том, что БД представляют собой системы взаимосвязанных данных, единство и целостность которых поддерживается специальными программными средствами.

- Для функционирования БнД необходимо наличие специальных языковых и программных средств (называемых *СУБД – Система Управления Базами Данных*), облегчающих для пользователей выполнение всех операций, связанных с организацией хранения данных, их корректировки и доступа к ним.

Нельзя сказать, что в рассматриваемой нами сфере установилось терминологическое единство. Так, в англоязычной литературе понятие «банк данных» используется редко. В некоторых из этих источников используется понятие «система баз данных» (database system), которое по своему содержанию близко введенному понятию банка данных (система баз данных включает базу данных, систему управления базами данных, соответствующее оборудование и персонал) [1]. Согласно семантики русского языка "система баз данных" воспринимается уже,

чем то, что это понятие обозначает в действительности. Поэтому слово "банк" является в этом смысле лучше, так как "банк" привычно обозначает не только то, что хранится в нем, но и всю инфраструктуру (вспомните, хотя бы понятие «банк» как финансовое учреждение – это ведь не просто «куча денег»). Очевидно, что нельзя отождествлять понятие "база данных" и "банк данных", как это иногда происходит в некоторых литературных источниках.

Терминологические различия наблюдаются и при определении других понятий в области БД. Особенно это касается терминов, используемых в конкретных программных системах. В связи с тем, что терминология конкретных СУБД сильно различается, нельзя описать общие принципы построения БД, пользуясь терминологией какой-либо одной из них. В учебнике будут введены термины, которые, по мнению автора, в наибольшей степени соответствуют отображаемым ими понятиям.

Следует отметить, что использование тех или иных терминов зависит от аспекта рассмотрения изучаемой проблемы. Так, например, в [4] под базой данных понимается практически любая совокупность данных, которая может быть обработана с помощью ЭВМ. И это оправдано, так как права собственности и иные права не могут зависеть от того, при помощи какого программного средства созданы файлы, и какой у них способ организации. Но такое широкое толкование термина БД в курсе «Проектирование баз данных» приведет к нивелированию особенностей банков данных как особой информационной технологии.

## **1.2. Преимущества БД**

Особенности "банковской" организации данных определяют их основные преимущества перед "небанковской" организацией.

Наличие единого отображения определенной части реального мира позволяет обеспечить непротиворечивость и целостность информации, возможность обращаться к ней не только при решении заранее предопределенных задач, но и с нерегламентированными запросами. Интегрированное хранение сокращает избыточность хранимых данных, что приводит к сокращению затрат не только на создание и хранение данных, но и на поддержание их в актуальном состоянии.

Использование БД при правильной его организации должно существенно изменить деятельность организации, где он внедряется: привести к обеспечению большей доступности данных для всех категорий сотрудников, сокращению документооборота, возможности получения разнообразных по форме и содержанию документов, перераспределению функций между сотрудниками и изменению

характера выполняемых функций и, как следствие, улучшить всю систему управления предприятием.

Централизованное управление данными также дает целый ряд преимуществ. Использование СУБД обеспечивает высокое качество выполнения функций по управлению данными и облегчает процесс создания информационных систем (ИС).

Выделение специальной группы сотрудников, выполняющих функции по проектированию и развитию БД (администраторов БД), и освобождение от этих функций всех остальных пользователей не только приводит к снижению требований к остальным участникам процесса создания и функционирования БД, но и повышает качество разработок, так как вопросами организации данных занимается небольшое число профессионалов в этой области.

Преимуществом банков данных является также то, что они обеспечивают возможность более полной реализации принципа независимости прикладных программ от данных, чем это возможно при организации локальных файлов.

### **1.3. Пользователи БД**

В процессе создания и эксплуатации БД с ним взаимодействуют пользователи разных категорий (рис. 1.1). Базы данных создаются для удовлетворения потребностей *конечных пользователей*. Чаще всего – это специалисты конкретных предметных областей, использующие БД для выполнения своих профессиональных обязанностей. В последнее время БД все чаще используются и для удовлетворения *непроизводственных информационных потребностей*. Конечные пользователи – наиболее многочисленная группа пользователей. Нельзя недооценивать важности этой группы пользователей и не понимать их специфических особенностей для каждой из категорий конечных пользователей БД.

Специфическими пользователями БД являются сотрудники информационных служб. Они пользуются, в основном, метайнформацией. Часто бывает желательным, чтобы другая информация была для них закрыта. Кроме того, они используют и другие ресурсы БД для выполнения своих функций.

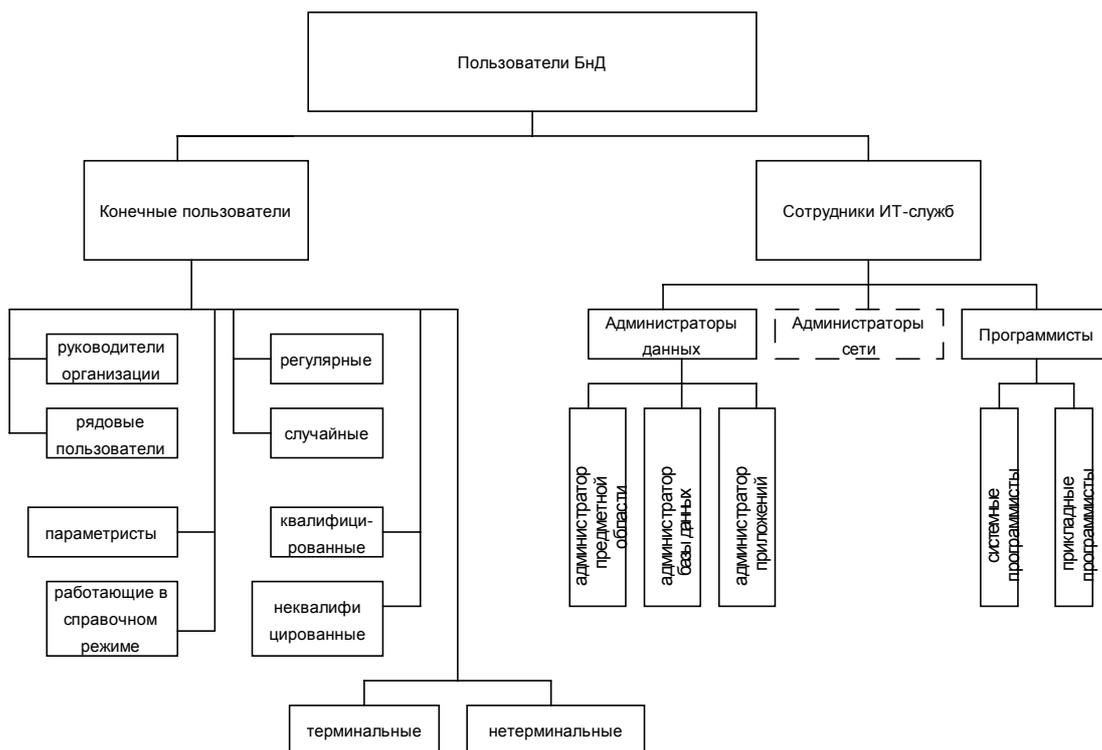


Рис. 1. 1. Пользователи БНД

Категория «Конечные пользователи» неоднородна: конечные пользователи различаются широтой информационных потребностей, квалификацией, режимами взаимодействия с БНД и др. Это могут быть случайные пользователи, обращающиеся к базе данных время от времени, а могут быть и регулярные пользователи. Конечные пользователи могут отличаться друг от друга и степенью владения вычислительной техникой. От конечных пользователей не должно требоваться каких-то специальных знаний в области вычислительной техники и языковых средств.

При создании БНД важно не только построение классификационной схемы, но и распределение реальных конечных пользователей по группам, так как от характеристики пользователей будут зависеть принимаемые проектные решения.

В связи с тем, что использование БНД оказывает влияние на все аспекты деятельности организации, особую роль играют руководители организации. Именно они должны обеспечить проведение единой информационной политики и организацию взаимодействия различных подразделений через общую базу данных. Они должны создавать подразделения, отвечающие за создание и функционирование БНД, определять функциональные обязанности сотрудников, которые существенно изменятся с внедрением БНД. Кроме того, руководители

организации выступают в качестве конечных пользователей с наиболее высоким приоритетом.

Отдельные пользователи в процессе работы с базой данных могут менять содержание БД – это так называемые пользователи-параметристы. Другие могут только использовать хранящуюся в БД информацию.

Пользователи могут взаимодействовать с БД как непосредственно (терминальные пользователи), так и через посредников.

Понятием «Конечные пользователи» определяется не только отдельное лицо или группа лиц, но и вычислительные процессы/задачи, а иногда и целые системы, взаимодействующие с БД.

В зависимости от особенностей создаваемого банка данных круг его конечных пользователей может существенно различаться

Категория «сотрудники информационных служб» также является неоднородной. В рамках курса «Базы данных» наибольший интерес для нас представляют «Администраторы БД» – лица, ответственные за создание БД и его надежное функционирование, за соблюдение регламента доступа к хранимым, за развитие БД.

Наличие в составе СУБД средств, ориентированных на разные категории пользователей, делает возможной работу с базой данных не только профессионалов в области обработки данных, но практически любого пользователя, причем это использование может быть как для их профессиональных целей, так и для удовлетворения потребности в информации в быту и т. п.

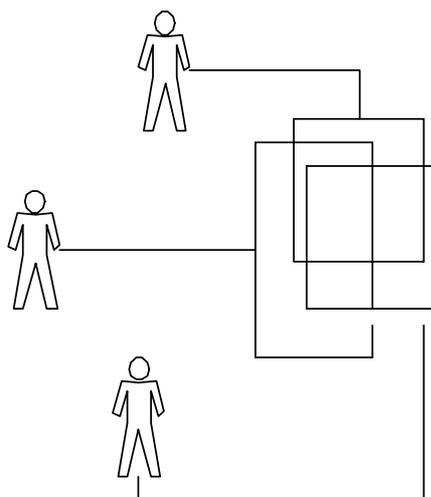


Рис.1.2. Пересечение информационных потребностей пользователей

#### **1.4. Предпосылки широкого использования БД**

Очевидные преимущества БД и объективные предпосылки их создания привели к широкому их использованию. К числу предпосылок применения БД относятся следующие:

- объекты реального мира находятся в сложной взаимосвязи между собой. Это приводит к необходимости, чтобы их информационное отражение также представляло единое взаимоувязанное целое;
- информационные потребности различных пользователей существенно пересекаются, что делает целесообразным использование единых баз данных и обеспечение доступа к ним разных пользователей (рис.1.2);
- функции создания и ведения информационного фонда и предоставления нужных данных тем или иным процессам являются универсальными, общими при решении разнообразных задач. Создание специализированных программных средств для управления данными приводит к повышению уровня выполнения этих функций и сокращению трудоемкости создания информационных систем;
- современный уровень развития технического и программного обеспечения, а также теории и практики построения информационных систем позволяют создавать эффективные БД.

#### **1.5. Требования к БД**

Особенности "банковской" организации данных позволяют сформулировать основные требования, предъявляемые к БД:

- адекватность отображения предметной области (полнота, целостность и непротиворечивость данных, актуальность информации, т. е. ее соответствие состоянию отображаемой реальной системы на данный момент времени);
- возможность взаимодействия пользователей разных категорий и в разных режимах; обеспечение высокой эффективности доступа для разных приложений;
- дружелюбность интерфейсов и малое время на освоение системы, особенно для конечных пользователей;
- обеспечение секретности и конфиденциальности для некоторой части данных; определение групп пользователей и их полномочий;

- обеспечение взаимной независимости программ и данных;
- обеспечение надежности функционирования БнД; защита данных от случайного и преднамеренного разрушения; возможность быстрого и полного восстановления данных в случае их разрушения; технологичность обработки данных;
- приемлемые характеристики функционирования БнД (стоимость обработки, время реакции системы на запросы, требуемые машинные ресурсы и др.).

### **1.6. Недостатки БнД**

Недостатки БнД вытекают из их достоинств. Создание интегрированной системы, естественно, сложнее, чем создание множества локальных систем. Как следствие, предъявляются высокие требования к квалификации разработчиков БнД. В результате интеграции возможна некоторая потеря эффективности отдельных приложений (но общая эффективность всей системы будет выше). Для управления данными требуются специализированное программное обеспечение, которое, в зависимости от класса системы, может быть сравнительно дорогим, предъявляющим повышенные требования к техническим средствам. Эксплуатация распределенных корпоративных БнД – процесс сложный и дорогостоящий.

Но, несмотря на некоторые недостатки, присущие такой форме организации данных, преимущества БнД значительно превосходят их недостатки. Кроме того, имеется очень широкий круг СУБД разных классов и технологий их использования. Правильный выбор системы позволит свести отрицательные последствия к минимуму.

## 2. Компоненты банка данных

Банк данных является сложной человеко-машинной системой, включающей различные взаимосвязанные и взаимозависимые компоненты (рис. 1.3).

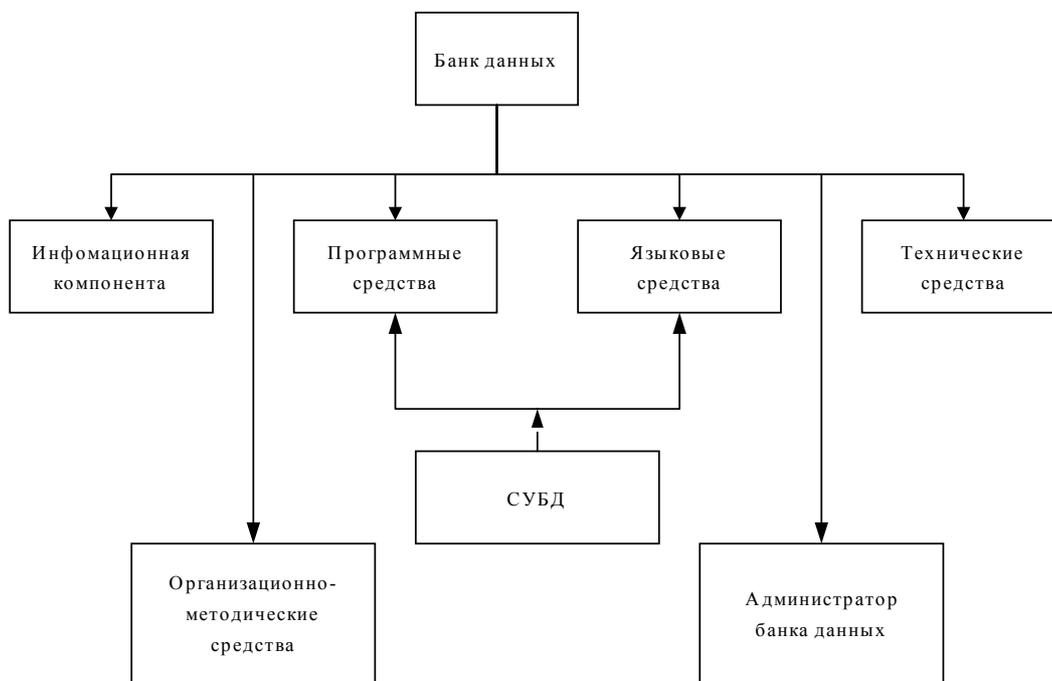


Рис. 1.3. Компоненты банка данных

### 2.1. Информационная компонента.

Ядром БНД является база данных. *База данных* – это поименованная совокупность взаимосвязанных данных, находящихся под управлением СУБД.

Существует множество определений базы данных. Некоторые из них имеют право на существование. Другие устарели и не соответствуют современным представлениям о БД. Так, в ранних определениях баз данных указывалось на их избыточность, отсутствие дублирование данных в них. На самом деле это не так. В базах данных может наблюдаться дублирование информации. Оно может быть вызвана спецификой используемой модели данных, не позволяющей полностью устранить дублирование, или технологическими причинами (обеспечение большей надежности, сокращение времени реакции системы и др.). Но это должна быть управляемая избыточность, причины и цели возникновения которой известны администратору базы данных и управляются как им, так и СУБД.

В настоящее время действует закон "О правовой охране программ для электронных вычислительных машин и баз данных" [4]. В этом законе дается следующее определение базы данных, "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"[4, ст.1]. Учитывая назначение этого закона, вполне естественно, что сделан иной акцент, чем в данном нами определении БД; определение, используемое в тексте данного Закона, является более широким, чем приведенное нами.

Мы в качестве рабочего будем пользоваться данным нами в п. 1.1 определением, и не всякие файлы будем считать базами данных.

В технической документации некоторых СУБД, а также в некоторых литературных источниках в состав БД включаются не только собственно хранимые данные о предметной области, но и описания БД. Более правильно описания баз данных считать самостоятельными компонентами БНД, даже если они и хранятся вместе с самими данными.

Описания баз данных относятся к *метаинформации*, т. е. информации об информации. Описание баз данных часто называют *схемой*. Кроме того, в БНД могут присутствовать описания отдельных частей базы данных с точки зрения конкретных пользователей. Такое описание называется *подсхемой*.

Кроме описания баз данных в состав метаинформации, хранимой в БНД, может включаться информация о предметной области, необходимая для проектирования автоматизированной информационной системы, о пользователях БНД, о проектных решениях и некоторая другая информация.

Централизованное хранилище метаинформации называется *словарем данных*. В литературе используются также термины словарь-справочник, энциклопедия, репозиторий. В некоторых источниках выявляются различия между этими терминами, в других они используются как синонимы. Для данного уровня рассмотрения для нас эти различия несущественны.

Роль словарной системы особенно возрастает при использовании средств автоматизированного проектирования информационных систем. Для большинства из них репозиторий является ядром всей системы. Кроме того, роль репозитория особо значима в распределенных системах.

К банку данных не относятся немашинные документы, служащие источниками информации, вводимой в БД, файлы входной и выходной информации, архивные файлы, выходные документы. Однако многие СУБД включают в свой состав языковые средства для описания этих компонент. В этом случае сами описания, используемые в процессе функционирования БНД, будут входить в его состав.

Как уже отмечалось выше, терминология, используемая в разных системах и разных литературных источниках, существенно различается. Так, в ранних версиях многих «настольных» реляционных СУБД вообще не использовался «механизм» базы данных (т. е. создавались фактически отдельные реляционные таблицы, каждая из которых запоминалась в отдельном файле базы данных). Во многих литературных источниках каждый такой файл стали называть базой данных, что не правильно.

В некоторых системах, например, Access, под БД понимают совокупность разных объектов: таблиц, запросов, форм, отчетов, макросов и модулей, т. е. понятие базы данных расширено, и включает в себя практически все информационные компоненты, созданные для конкретного приложения. В других системах, в частности, в Paradox, для обозначения подобной совокупности взаимосвязанных объектов используется понятие «семейство», что, очевидно, терминологически более правильно.

При работе с конкретной системой надо, прежде всего, уточнить терминологию, используемую в ней.

## 2.2. Программные средства БД

Программные средства БД представляют собой сложный комплекс, обеспечивающий взаимодействие всех частей информационной системы при ее функционировании (рис 1.4).

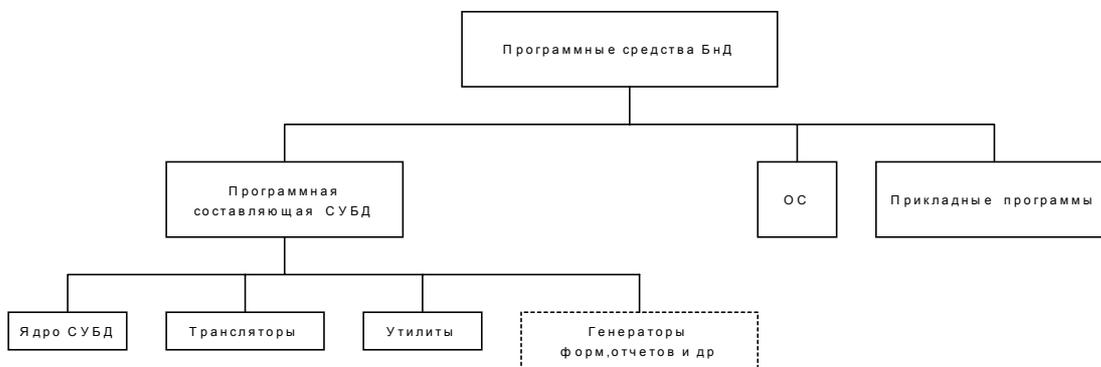


Рис. 1.4. Программные средства БД

Основу программного обеспечения БД представляют программные компоненты СУБД. Среди них можно выделить ядро СУБД, обеспечивающее создание БД, организацию ввода, обработки и хранения данных, т. е. именно то, что называется «управлением данными», а также другие компоненты, обеспечивающие настройку системы, средства тестирования, утилиты, обеспечивающие выполнение вспомогательных функций, таких как восстановление баз данных, сбор статистики о функционировании БД и др. Важной компонентой СУБД являются трансляторы или компиляторы для используемых ею языковых средств.

В состав большинства СУБД включены программные компоненты, позволяющие автоматизировать проектирование систем обработки информации (генераторы отчетов, меню и др.). Строго говоря, эти функции не являются непосредственно функциями по управлению данными, но большинство современных программных средств, которые продолжают называться СУБД, выходят за названные рамки и фактически являются мощными комплексными инструментальными средствами, позволяющими автоматизировать процесс создания информационных систем.

подавляющее большинство СУБД работает в среде универсальных операционных систем (ОС) и взаимодействует с ОС при обработке обращений к БД. Поэтому можно считать, что ОС также входит в состав БД.

Для удовлетворения конкретных потребностей пользователей пишутся соответствующие программы<sup>1</sup>, которые представляют прикладное программное обеспечение БД.

В зависимости от используемых технологий создания и функционирования систем могут появляться те или иные дополнительные компоненты. Так, при использовании Case-технологий будут присутствовать соответствующие программные компоненты, поддерживающие проектирование и перепроектирование системы.

При работе в архитектуре клиент-сервер программные средства будут подразделяться на соответствующие компоненты: клиентская часть, обеспечивающая интерфейс пользователя с системой, серверная часть, реализующая обработку запроса на сервере, и связанная часть, обеспечивающая взаимодействие элементов в сети.

Программные средства, используемые при создании и эксплуатации БД, будут также зависеть от масштаба БД, требований, предъявляемых к нему.

### **2.3. Языковые средства БД**

Языковые средства СУБД являются важнейшей компонентой банков данных, так как, в конечном счете, они обеспечивают интерфейс пользователей разных категорий с банком данных. Набор используемых языков средств широк и разнообразен. Языковые средства, используемые в БД, можно классифицировать по разным признакам (рис. 1.5).

---

<sup>1</sup> Понятие «программа» здесь трактуется широко. Это могут быть, например, и экранные формы, созданные с использованием визуальных средств, и запросы, написанные на любом языке запросов.

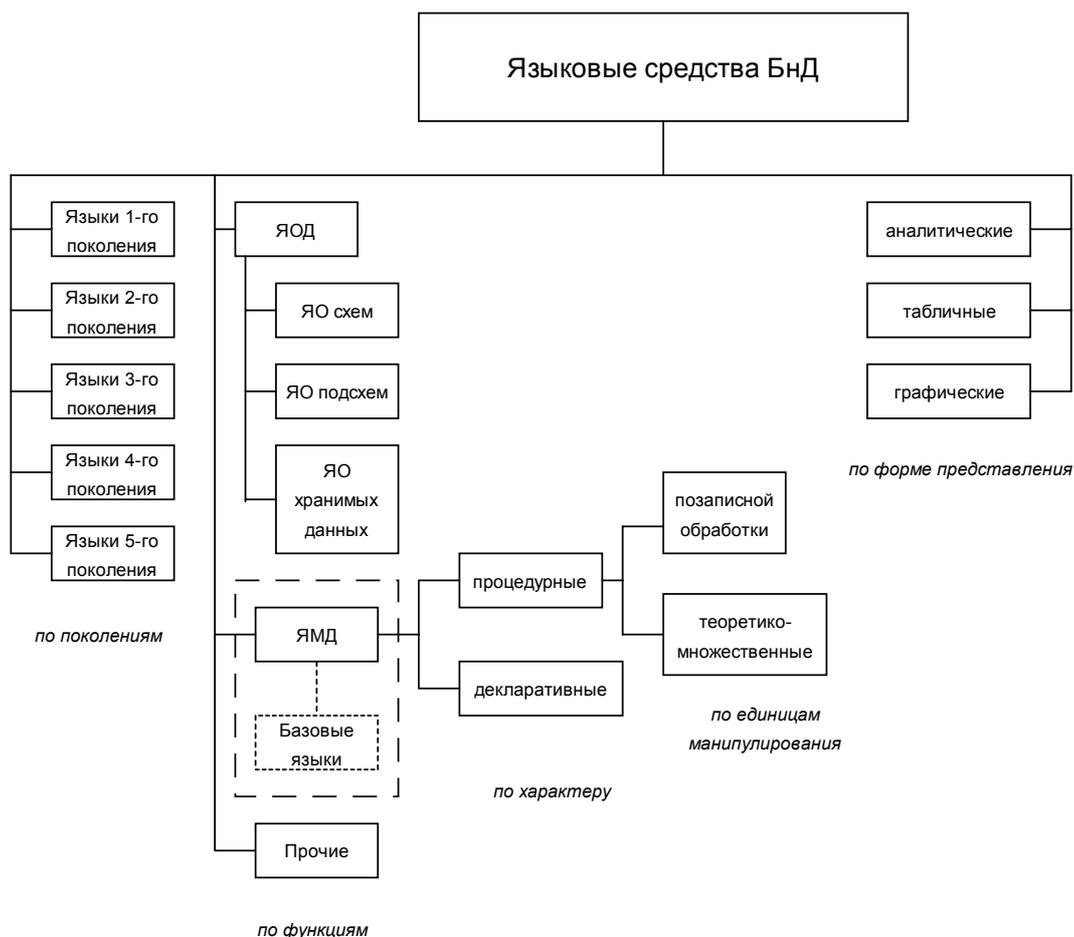


Рис. 1.5. Классификация языковых средств БД

Языковые средства большинства современных СУБД относятся к языкам четвертого поколения (к первому поколению языков относят машинные языки, ко второму – символические языки ассемблера, к третьему – алгоритмические языки типа PL, COBOL и т.п., которые в 60-е годы назывались языками высокого уровня, но уровень которых гораздо ниже, чем у языков четвертого поколения. Имеются еще и языки пятого поколения, к которому относят языки систем искусственного интеллекта).

Языки четвертого поколения создавались по принципу: "люди стоят дороже, чем машины" [2]. При их проектировании используются следующие принципы:

1. Принцип минимума работы: язык должен обеспечить минимум усилий, чтобы "заставить" машину работать.
2. Принцип минимума мастерства: работа должна быть так проста, как только это возможно; она не должна быть уделом избранных и быть понятной лишь посвященным.
3. Принцип естественности языка, упразднения "иностранного" синтаксиса и мнемоники. Язык не должен требовать от пользователей значительных усилий в изучении синтаксиса или содержать много мнемонических или иных обозначений, которые быстро забываются.

4. Принцип минимума времени. Язык должен позволять без существенной задержки реализовывать возникающие потребности в доступе к информации и ее обработке.
5. Принцип минимума ошибок. Технология должна быть спроектирована таким образом, чтобы минимизировать ошибки человека, а уж если они возникли, то, по возможности, "выловить" их автоматически.
6. Принцип минимума поддержки. Механизм языков четвертого поколения должен позволить легко вносить изменения в имеющиеся приложения.
7. Принцип максимума результата. Языки четвертого поколения предоставляют пользователям мощный инструмент для решения разнообразных задач.

На рис. 1.6 представлены компоненты языка четвертого поколения. Как мы видим, здесь представлены все основные «генераторы», наличие которых уже стало традиционным для СУБД разных классов.

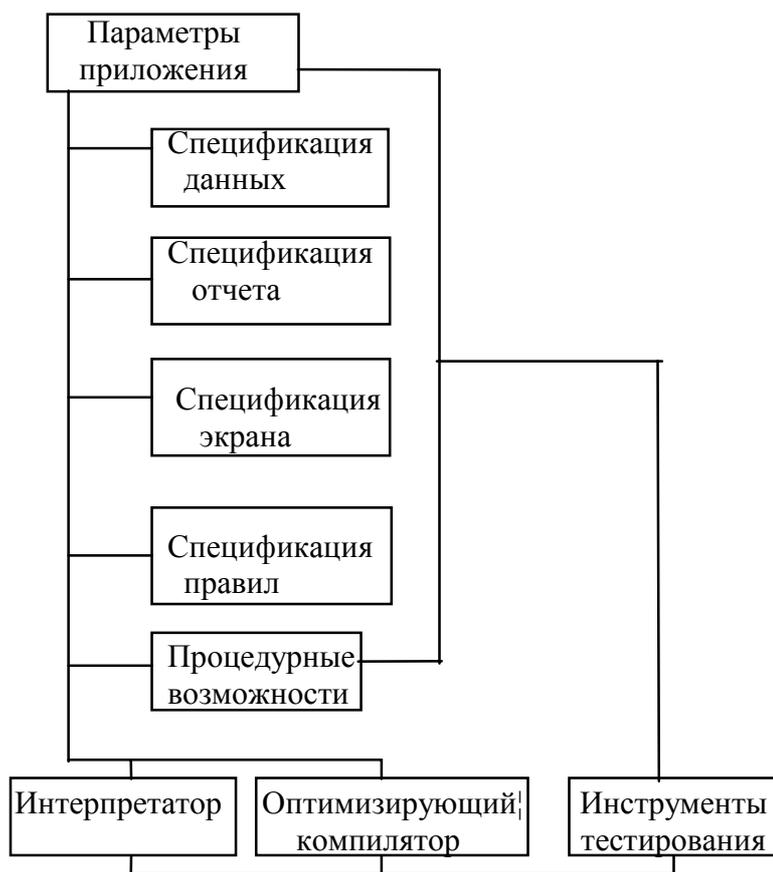


Рис. 1.6. Компоненты языка четвертого поколения

Можно выделить две концепции развития языковых средств: концепцию разделения и концепцию интеграции. При использовании концепции разделения различают *языки описания данных* (ЯОД) и *языки*

*манипулирования данными* (ЯМД). Назначение каждого из этих подклассов ясно из их названия.

Иногда в особую группу выделяют *языки запросов* (ЯЗ). Первоначально под языками запросов понимали языки высокого уровня, ориентированные на конечного пользователя, предназначенные для формирования запросов к БД (в такой трактовке их можно считать одной из разновидностей ЯМД). Однако сейчас ЯЗ понимается шире (см. гл. 3); многие ЯЗ включают в себя еще и возможности описания данных и корректировки БД.

В составе языков описания данных в зависимости от особенностей СУБД поддерживаются все или некоторые из следующих языков: *язык описания схем* (ЯОС), *язык описания подсхем* (ЯОПС), *язык описания хранимых данных* (ЯОХД), *языки описания внешних данных* (входных, выходных). В некоторых СУБД и сами эти разновидности языков, и создаваемые с их помощью элементы ИС являются самостоятельными компонентами, в других – некоторые из них могут объединены.

Языки манипулирования данными разделяются на две большие группы: *процедурные* и *непроцедурные*. При пользовании процедурными языками надо указать, какие действия и над какими объектами необходимо выполнить, чтобы получить результат. В непроцедурных языках указывается, что надо получить в ответе, а не как этого достичь.

Процедурные языки могут различаться по основным информационным единицам, которыми они манипулируют. Это могут быть языки, ориентированные на позаписную обработку данных, и языки, ориентированные на операции над множеством записей. Так, операции реляционной алгебры оперируют целиком отношением, а не каждой его записью.

Примерами непроцедурных языков являются языки, основанные на реляционном исчислении. Представителем языков, основанных на реляционном исчислении кортежей, является широко используемый язык запросов SQL. Табличный язык QBE также является непроцедурным языком.

Языковые средства предназначаются для пользователей разных категорий: конечных пользователей, системных аналитиков, профессиональных программистов. Повышение уровня языковых средств, их дружелюбности приводит к тому, что все большее число функций выполняется пользователями-непрограммистами самостоятельно, без посредников.

**По своим функциональным возможностям** выделяют следующие категории языков [2]:

1. Языки, обеспечивающие только возможности запросов. Они обеспечивают вывод требуемых данных на экран или печать в нужном формате. В настоящее время используется редко.

2. Комплексные языки запросов/обновлений. Более развитые языки, которые позволяют формулировать сложные запросы,

относящиеся к нескольким взаимосвязанным записям, а также обновлять данные также легко, как и формулировать запросы. Используя их, пользователи могут создавать свои собственные файлы.

3. Генераторы отчетов. Они позволяют выбирать нужные данные из файлов или баз данных и форматировать их в виде требуемых форм документов.

4. Графические языки. Использование графических средств в настоящее время постоянно расширяется. Они позволяют выводить данные в виде различных графиков и диаграмм, а также использовать другие изобразительные возможности. Также как генераторы отчетов, графические языки позволяют осуществлять отбор информации из файлов или баз данных по различным критериям, а также выполнять арифметические и логические манипуляции с данными.

5. Инструментальные средства поддержки решений. Языки этого типа предназначены для создания систем принятия решений. Это могут быть системы типа "что-если", системы, выполняющие временной или трендовый анализ и другие. Возможно использование как универсальных, так и проблемно-ориентированных средств.

6. Генераторы приложений. Языковые средства, предназначенные для генерации приложений, обеспечивают возможность описания непроцедурным путем требуемой обработки информации и дальнейшей автоматической генерации программ.

7. Машиноориентированные языки спецификаций. Фактически являются генераторами приложений, дальнейшим их развитием. В отличие от генераторов приложений языки спецификаций более универсальны и позволяют специфицировать приложения разных типов.

8. Языки очень высокого уровня. В большинстве случаев приложения строятся при помощи непроцедурных языков. Однако некоторые языки являются процедурными (например, NOMAD), но программирование на них значительно короче, чем, например, на COBOLe.

9. Параметризированные пакеты прикладных программ. Эта категория программных средств известна давно и "четвертое поколение" относится к таким ППП, которые допускают легкую модификацию самого пакета, позволяют пользователям генерировать собственные отчеты, запросы к базе данных, и т.д.

10. Языки приложений. Многие языки четвертого поколения являются универсальными языками. Другие – спроектированы для специфических приложений. Примерами таких языков являются языки для управления финансами, управления работой станков с программным управлением и т.д.

**По форме представления** различают *аналитические, табличные и графические* языковые средства. Классификация языковых средств по форме представления относится как к языкам описания данных, так и к языкам манипулирования данными. Так, описание таблицы с

использованием команды CREATE TABLE языка SQL является примером аналитической формы ЯОД, а описание такой же таблицы в Access и большинстве других настольных СУБД – пример табличной формы описания. В качестве примеров табличной и аналитической формы задания запросов можно привести языки QBE и SQL соответственно.

Достаточно часто бывает, что в рамках одной СУБД для одних и тех же целей могут использоваться языки разных типов. Так, например, во многих СУБД (dBase, FoxPro, и др.) для манипулирования данными могут использоваться:

1. табличный язык запросов типа QBE
2. язык SQL – аналитический ЯЗ, относящийся к классу языков исчисления кортежей
3. процедурный язык программирования (для указанных выше систем dBase, FoxPro это язык xBase, часть операторов которого реализуют операции реляционной алгебры, а другая часть, более значительная по количеству операторов и функций, представляет собой нереляционные операции, обеспечивающие позаписную обработку файлов, организацию циклической и условной обработки, ввод-вывод данных, корректировку, возможность работы с переменными памяти и другие возможности).

Описание данных в этих системах может быть представлено в табличном виде, либо, если определение данных происходит средствами SQL или с использованием операторов языка xBase, в аналитическом виде.

Кроме упомянутых языковых средств, эти системы включают в себя генераторы экранных форм, отчетов и приложений, а также язык разветвленной иерархической системы "меню", позволяющей пользователю выбрать те действия, которые он желает выполнить.

Часто СУБД обеспечивают автоматическое преобразование «текстов» с одного языка на другой. Так, например, многие СУБД, такие как Access, FoxPro и др., используют языки запросов табличного типа не только для непосредственной реализации запросов, но и как средство для более простого описания запроса и последующего автоматического преобразование его на язык SQL.

#### **2.4. Технические средства БД.**

В качестве технических средств для банков данных (рис. 1.7) чаще всего используются универсальные ЭВМ, периферийные средства для ввода информации в базу данных и отображения выводимой информации. Иногда используются дополнительные технические средства для хранения больших объемов данных на внешних носителях. Если банк данных реализуется в сети, то необходимы соответствующие технические средства для обеспечения ее работы.

Состав и тип технических средств, на которых реализуются БНД, зависит от многих факторов, основными из которых являются: технические характеристики оборудования, используемые технологии обработки данных, масштаб системы, временные ограничения на время реакции системы, сложность обработки, стоимостные характеристики и др.

Первоначально БНД реализовывались в основном на больших ЭВМ, а для доступа к БД использовались терминалы. В связи со значительным и постоянным улучшением характеристик персональных ЭВМ появилась возможность реализовать банки данных и на машинах этого класса. Но сначала характеристики персональных ЭВМ были недостаточными, чтобы в полной мере реализовать идеологию банков данных. Стала наблюдаться некоторая раздробленность информационных систем, что, в свою очередь, привело к бурному развитию сетевых технологий и использованию соответствующих технических средств.

Существуют и специализированные технические средства, предназначенные для создания и эксплуатации банков данных (машины баз данных), но они не нашли широкого распространения.

В последние годы некоторые фирмы (Oracle, Sun) активно развивают идею использования так называемых «сетевых компьютеров». Эти компьютеры представляют собой дешевые рабочие станции без дисковых накопителей<sup>2</sup>, которые будут работать в сети и использовать и программные средства, и данные, которые находятся на сервере.

Использование сетевых компьютеров предполагает обязательное применение мощных ЭВМ в качестве серверов, предъявляет высокие требования к организации хранения данных, к качеству каналов связи. При этом во многом становится predetermined технологией обработки данных (особенно в части распределения функций между клиентом и сервером). Использование сетевых компьютеров обусловлено не столько тем, чтобы сэкономить за счет использования более дешевых компьютеров, сколько желанием упорядочить использование программных средств, упростить систему обработки информации в целом, облегчить и удешевить поддержку системы.

Недостатки такого подхода являются:

- очень большая зависимость от «центральной» системы, потеря самостоятельности конечными пользователями
- уязвимость системы
- невозможность/неэффективность обеспечения потребности всех пользователей таким образом (хотя потребности пользователей и пересекаются, но степень пересечения может быть разной; кроме того,

---

<sup>2</sup> позднее появились предложения создавать сетевые компьютеры с собственными накопителями.

не исключается наличие сугубо персональных данных; хранение таких данных на отдаленном сервере приводит к непроизводительным расходам)

- очень высокие требования к серверной части системы.

Другим новым явлением является использование карманных ПК в качестве коммуникационных устройств для доступа к корпоративным данным в глобальных сетях.

Характеристики карманных компьютеров существенно улучшаются. Для них создается соответствующее программное обеспечение, позволяющее использовать их для мобильных пользователей, работающих в общей системе (тиражирование и синхронизация данных). Эти компьютеры являются более легкими, что немало важно для мобильных пользователей, а также стоят дешевле, чем переносные ПК. Ведущие производители СУБД приспособливают свои крупномасштабные серверные системы для доступа из карманных ПК.

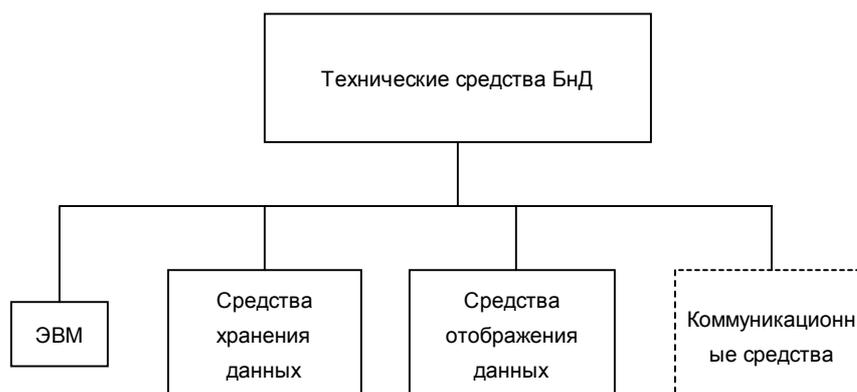


Рис. 1.7. Технические средства БнД

Тип используемых ЭВМ будет зависеть от масштаба создаваемой системы (Рис. 1.8). В настоящее время в подавляющем большинстве случаев БнД реализуются в сетевой среде с использованием множества разнотипных ЭВМ, причем их состав постоянно меняется в процессе эксплуатации банка данных.

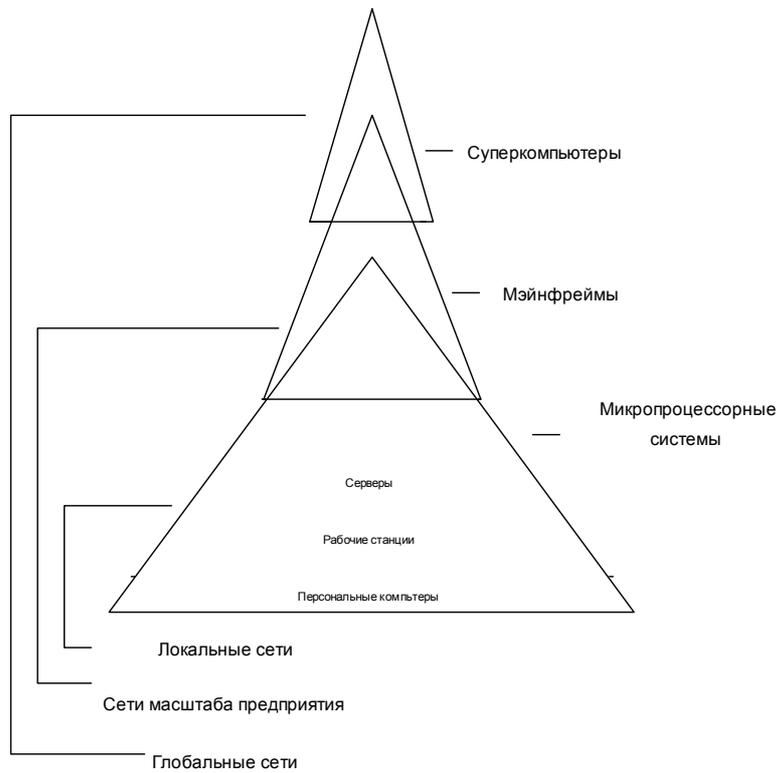


Рис. 1. 8. Взаимосвязь используемых ЭВМ и технологии организации информационной системы

Технические средства БД не ограничиваются только ЭВМ. Сюда входит весь комплекс технических средств хранения, отображения и передачи информации. Особую роль для обеспечения эффективного и надежного функционирования банка данных играют средства хранения информации. Память в БД обычно организуется в виде многоуровневой системы. Необходимо обращать внимание не только на выбор запоминающих устройств для организации хранения данных, предназначенных для оперативного доступа к ним, но и хранения архивных данных.

В банках данных, также как и во всех других информационных системах, выполняются операции по вводу, хранению, обработке и выводу информации. При выполнении каждой из этих операций могут использоваться различные технологии и, как следствие, различные технические и программные средства для их поддержания.

## 2.5. Организационно-методические средства.

Организационно-методические средства банка данных представляют собой различные инструкции, методические и регламентирующие материалы, предназначенные для пользователей разных категорий, взаимодействующих с банком данных. Это могут быть инструкции конечным пользователям по работе с базой данных, документы, определяющая права доступа и регламент работы; сюда же отнесем и методики проектирования баз.

## 2.6. Администраторы банка данных.

Функционирование БнД невозможно без участия специалистов, обеспечивающих создание, функционирование и развитие БнД. Такая группа специалистов называется администратором банка данных (АБД). Эта группа специалистов считается составной частью банка данных.

В зависимости от сложности и объема банка данных, от особенностей используемой СУБД служба администрации банка данных может различаться как по составу и квалификации специалистов, так и по количеству работающих в этой службе.

**Функции администратора банка данных.** АБнД выполняют работы по созданию и обеспечению функционирования БнД на протяжении всех этапов жизненного цикла системы. В составе группы администраторов банка данных можно выделить различные подгруппы в зависимости от выполняемых ими функций. Численность группы администрации, выполняемые ими функции, будут в значительной степени зависеть от масштаба банка данных, специфики хранимой в нем информации, типа банка данных, особенностей используемых программных средств и некоторых других факторов.

В составе администрации БнД должны быть системные аналитики, проектировщики структур данных и внешнего по отношению к банку данных информационного обеспечения, проектировщики технологических процессов обработки данных, системные и прикладные программисты, операторы, специалисты по техническому обслуживанию. Если речь идет о коммерческом банке данных, то важную роль здесь будут играть специалисты по маркетингу.

Администраторы банка данных выполняют большой круг разнообразных функций. Далее в учебнике мы будем подробно рассматривать некоторые из них. Сейчас же просто перечислим основные из этих функций:

1. Анализ предметной области: описание предметной области, выявление ограничений целостности, определение статуса информации, определение потребностей пользователей, определение статуса пользователей, определение соответствия "данные –

- пользователь", определение объемно-временных характеристик обработки данных.
2. Проектирование структуры базы данных: определение состава и структуры информационных единиц, составляющих базу данных, задание связей между ними, выбор методов упорядочения данных и методов доступа к информации, описание структуры БД на ЯОД.
  3. Задание ограничений целостности при описании структуры базы данных и процедур обработки БД: задание ограничений целостности, присущих предметной области, определение ограничений целостности, вызванных структурой базы данных, разработка процедур обеспечения целостности БД при вводе и корректировке данных, обеспечение ограничений целостности при параллельной работе пользователей в многопользовательском режиме.
  4. Первоначальная загрузка и ведение базы данных: разработка технологии первоначальной загрузки и ведения (изменения, добавления, удаления записей) БД, проектирование форм ввода, создание программных модулей, подготовка исходных данных, ввод и контроль ввода.
  5. Защита данных от несанкционированного доступа.
    - 5.1. Обеспечение парольного входа в систему: регистрация пользователей, назначение и изменение паролей.
    - 5.2. Обеспечение защиты конкретных данных: определение прав доступа групп пользователей и отдельных пользователей, определение допустимых операций над данными для отдельных пользователей, выбор/создание программно-технологических средств защиты данных; шифрование информации с целью защиты данных от несанкционированного использования.
    - 5.3. Тестирование средств защиты данных.
    - 5.4. Фиксация попыток несанкционированного доступа к информации.
    - 5.5. Исследование возникающих случаев нарушения защиты данных и проведение мероприятий по их предотвращению.
  6. Защита данных от разрушений. Одним из способов защиты от потери данных является резервирование. Используется как при физической порче файла, так и в случае, если в БД внесены нежелательные необратимые изменения.
  7. Обеспечение восстановления БД: разработка программно-технологических средств восстановления БД, организация ведения системных журналов.
  8. Анализ обращений пользователей к БД: сбор статистики обращений пользователей к БД, ее хранение и анализ (кто из пользователей, к какой информации, как часто обращался, какие выполнял операции, время выполнения запросов, анализ причин безуспешных (в т. ч. и аварийных) обращений к БД).

9. Анализ эффективности функционирования БД и развитие системы: анализ показателей функционирования системы (время обработки, объем памяти, стоимостные показатели), реорганизация и реструктуризация баз данных, изменение состава баз данных, развитие программных и технических средств.
10. Работа с пользователями: сбор информации об изменениях в предметной области, об оценке пользователями работы БД, определение регламента работы пользователей с БД, обучение и консультирование пользователей.
11. Подготовка и поддержание системных программных средств: сбор и анализ информации о СУБД и других ПП, приобретение программных средств, их установка, проверка работоспособности, поддержание системных библиотек, развитие программных средств.
12. Организационно-методическая работа: выбор или создание методики проектирования БД, определение целей и направлений развития системы, планирование этапов развития БД, разработка и выпуск организационно-методических материалов.

**Связи администратора банка данных.** В процессе своей деятельности администратор БД взаимодействует с другими категориями пользователей банка данных, а также и с "внешними" специалистами, не являющимися пользователями БД (рис.1.9).

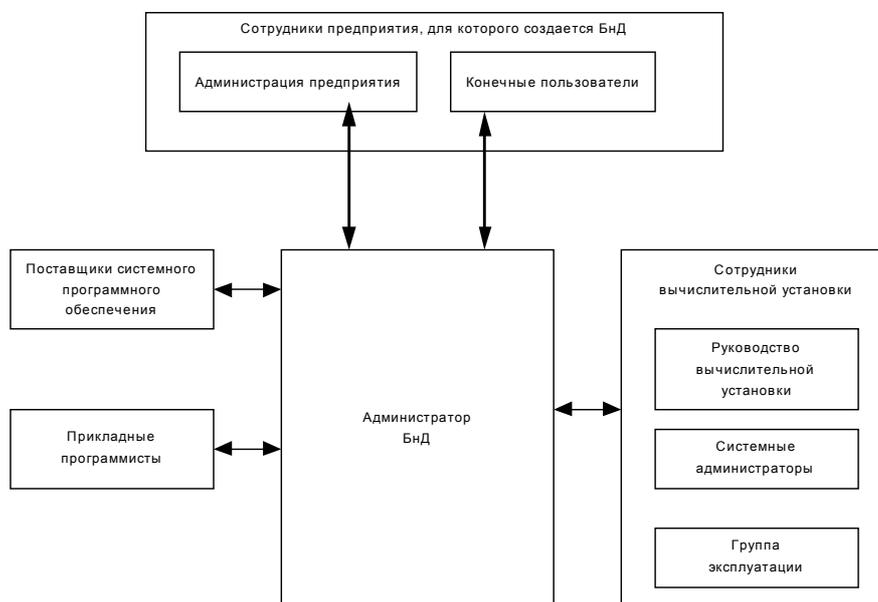


Рис. 1.9. Взаимодействие АДБД с другими категориями пользователей

Прежде всего, если банк данных создается для информационного обслуживания какого-либо предприятия или организации, то необходимы контакты с администрацией этой организации. Как

указывалось выше, внедрение БнД приводит к большим изменениям не только системы обработки данных, но и всей системы управления организацией. Естественно, что такие большие проекты не могут быть выполнены без активного участия и поддержки руководителей организации. Руководство организации должно быть ознакомлено с возможностями, предоставляемыми БнД, проинформировано об их преимуществах и недостатках, а также проблемах, вызываемых созданием и функционированием БнД.

Так как база данных является динамическим информационным отображением предметной области, то желательно, чтобы администратор БнД в свою очередь был своевременно информирован о перспективах развития объекта, для которого создается информационная система.

Руководством организации и администратором БнД должны быть согласованы цели, основные направления и сроки создания БнД и его развития, очередность подключения пользователей.

Очень тесная связь у АБД на всех этапах жизненного цикла БнД наблюдается с конечными пользователями. Это взаимодействие начинается на начальных стадиях проектирования системы, когда изучаются потребности пользователей, уточняются особенности предметной области, и постоянно поддерживается как на протяжении процесса проектирования, так и функционирования системы.

Следует отметить, что в последнее время наблюдается активное перераспределение функций между конечными пользователями и администраторами банка данных. Это, прежде всего, связано с развитием языковых и программных средств, ориентированных на конечных пользователей. Сюда относятся простые и одновременно мощные языки запросов, а также средства автоматизации проектирования.

Если банк данных функционирует в составе какой-либо включающей его автоматизированной информационной системы (например, в АСУ), то АБД должен работать в контакте со специалистами по обработке данных в этой системе.

Администраторы БнД взаимодействуют и с внешними по отношению к нему группами специалистов и, прежде всего, поставщиками СУБД и ППП, администраторами других БнД.

БнД часто создаются специализированными проектными коллективами на основе договора на разработку информационной системы в целом или БнД как самостоятельного объекта проектирования. В этом случае служба администрации БнД должна создаваться как в организации-разработчике, так и в организации-заказчике.

**Средства администратора современных СУБД.** На эффективность работы БнД оказывают влияние множество внешних и внутренних факторов. Возрастание сложности и масштабов БнД,

высокая «цена» неправильных или запоздалых решений по администрированию БД, высокие требования к квалификации специалистов делают актуальной задачу использования развитых средствах автоматизированного (или даже автоматического) администрирования БД.

Средства администрирования включены в состав всех СУБД. Особенно развиты эти средства в корпоративных СУБД. Кроме того, появился целый класс специализированного программного обеспечения: средства DBA (DataBase Administration - администрирование базы данных).

### Типичные функции средств DBA

Мониторинг работы БД, реакция на нештатные ситуации	Наблюдение за объектами БД, анализ, сопоставление характеристик	Оптимизация хранения данных, оптимизация работы сервера	Сопровождение БД, файлов, табличных пространств, откатных сегментов
Слежение за использованием ресурсов, выдача статистики	Планирование необходимых вычислительных мощностей	Анализ свободного пространства, устранение дефрагментации	Перенос таблицы на новое пространство, в другую СУБД, на другой компьютер
Обнаружение и исправление возникающих неполадок	Задание пороговых значений для слежения за нужными объектами	Наблюдение за параметрами, влияющими на производительность БД	Перенос содержимого базы данных в другую СУБД

### 2.7. Взаимодействие компонентов БД.

На рис. 1.10 представлена упрощенная схема взаимодействия компонентов БД в процессе создания и эксплуатации системы. Создание БД начинается с проектирования БД и ее описания на ЯОД (1). На этапе проектирования структуры БД могут использоваться как методики «ручного» проектирования, так и CASE-средства, автоматически генерирующие описания БД. Полученные описания должны быть введены в БД и запомнены в соответствии с требованиями конкретной СУБД (2,3). После того, как описание базы данных сохранено, в базу данных могут вводиться данные (4). При этом СУБД использует метаинформацию, зафиксированную в словаре

данных. Заполненная БД может использоваться для извлечения из нее нужной пользователям информации (5). При формулировании запросов используется информация, содержащаяся в схемах и подсхемах. В результате выполнения запроса выходные данные в том или виде выдаются пользователю (6). Кроме собственно затребованных данных при выполнении операций с БД часто выдается та или иная диагностическая информация (7). Для обеспечения надежности функционирования БД необходимо выполнять соответствующие процедуры, в частности осуществлять журнализацию (8) выполняемых действий с БД, регулярно архивировать данные (9).

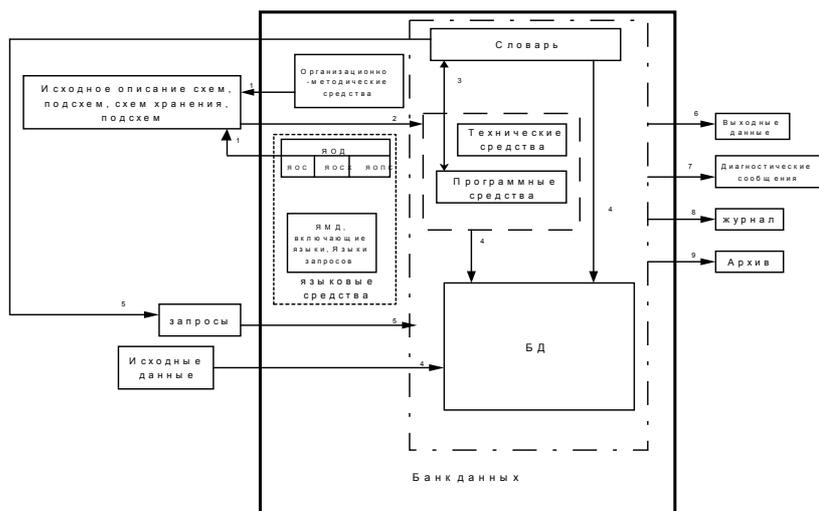


Рис. 1.10. Схема взаимодействия компонентов БД

### 3. Классификация банков данных

Банки данных являются сложными системами, и их классификация может быть произведена как для всего банка данных в целом, так и для каждой его компоненты отдельно; классификация для каждого из компонентов может быть проведена по множеству разных признаков (рис. 1.12).

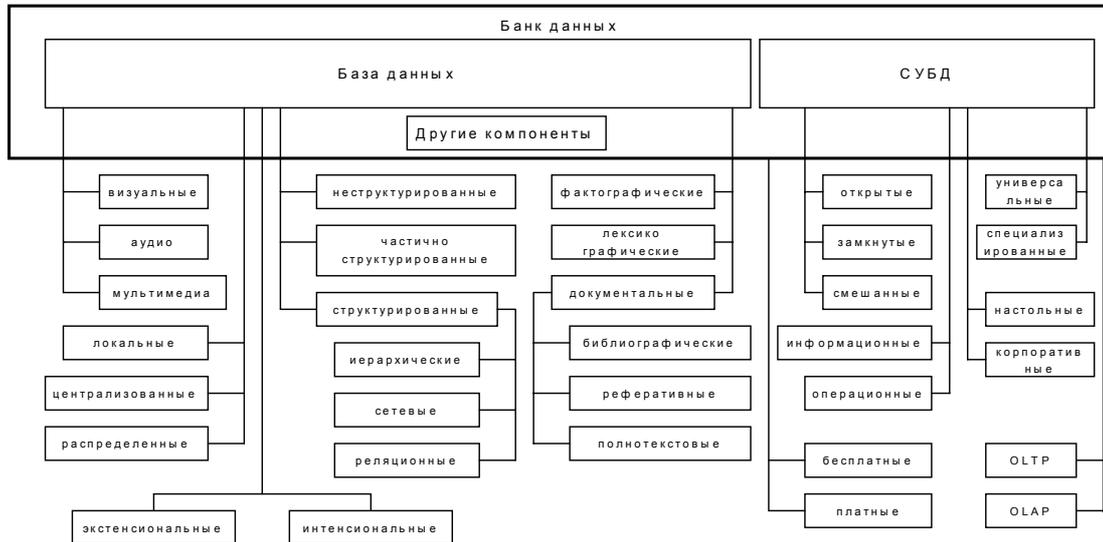


Рис. 1.12. Классификация БД

#### 3.1. Классификация баз данных

Центральной компонентой банка данных является база данных, и большинство классификационных признаков относятся именно к ней. **По форме представления информации** различают *визуальные* - и *аудио* системы, а также системы *мультимедиа*. Эта классификация показывает, в каком виде информация храниться в БД и выдается из баз данных пользователям: в виде изображения, звука или имеется возможность использования разных форм отображения информации. Понятие "изображение" здесь используется в широком смысле: это может быть символьный текст, неподвижное графическое изображение (рисунки, чертежи и т.п.), фотографии, географические карты, движущие изображения. Классификация способов представления информации являет собой самостоятельную проблему и здесь не рассматривается.

**По характеру организации данных** БД могут быть разделены на *неструктурированные*, *частично структурированные* и *структурированные*. Этот классификационный признак относится к информации, представленной в символьном виде. К неструктурированным БД могут быть отнесены базы, организованные в виде семантических сетей. Частично структурированными можно считать базы данных в виде обычного текста или гипертекстовые системы. Структурированные БД требуют предварительного

проектирования и описания структуры БД. Только после этого базы данных такого типа могут быть заполнены данными.

Структурированные БД в свою очередь **по типу используемой модели** делятся на *иерархические, сетевые, реляционные, смешанные и мультимодельные*.

Классификация по типу модели распространяется не только на базы данных, но и на СУБД.

В структурированных БД обычно различают несколько уровней информационных единиц, входящих одна в другую. Число этих уровней может быть различным даже для систем, относящихся к одному и тому же классу. Большинство структурированных систем поддерживают уровень поля, записи и файла. Эти информационные единицы могут называться в разных системах по-разному, но суть остается одной и той же, а именно: *полю* соответствует наименьшая семантическая единица информации; совокупность полей или и иных, более сложных информационных единиц, если они допустимы в конкретной СУБД, образуют *запись*, а множество однотипных записей представляют *файл базы данных*. В последнее время большинство СУБД в явном виде поддерживают и уровень *базы данных*, как совокупности взаимосвязанных файлов БД.

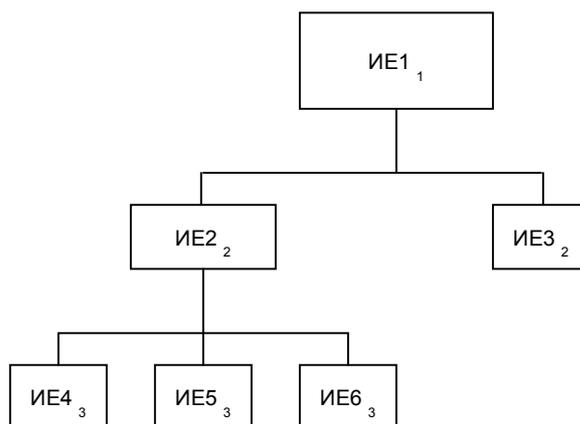


Рис. 1. 13. Схема иерархической модели

В иерархических (Рис. 1.13) и сетевых (Рис. 1.14) моделях между информационными единицами (записями разных файлов) могут задаваться связи. Как видно из приведенных схем, графическое представление иерархической модели представляет собой граф типа «дерево». В такой модели имеется одна вершина – корень дерева, являющаяся входом в структуру. Каждая вершина, отличная от корня, может иметь только одну исходную вершину и, в общем случае, сколько угодно порожденных вершин.

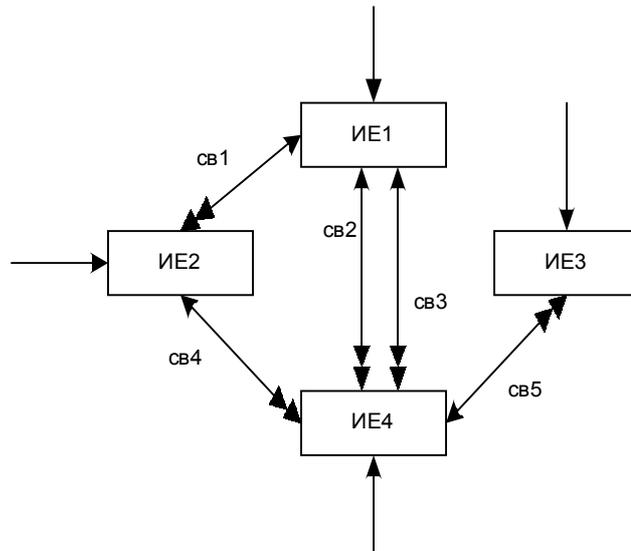


Рис. 1. 14. Схема сетевой модели

Графическое представление сетевой модели представляет собой граф типа «сеть». Входом в такую структуру может являться любая вершина. Каждая вершина может иметь как несколько порожденных, так и несколько исходных вершин. Между парой вершин может быть объявлено несколько связей. Подавляющее большинство СУБД поддерживает простые сетевые структуры, т. е. между каждой парой типов записей поддерживается отношение 1:М. Направление и характер связи в сетевых моделях не является очевидными, как в случае иерархической модели, поэтому при изображении структуры БД направление связи должно быть указано.

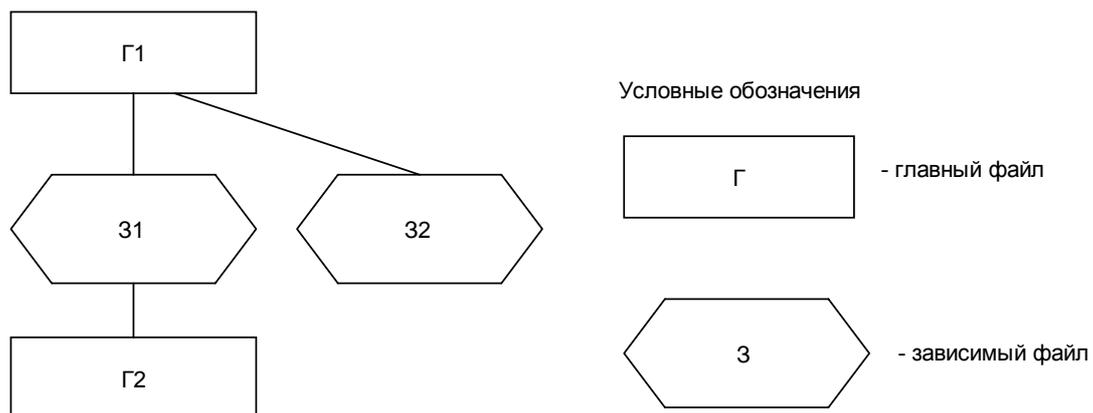


Рис. 1. 15. Схема сетевой модели с разнотипными файлами

Кроме сетевых моделей с равноправными файлами существуют сетевые модели с разнотипными файлами. В таких моделях различают главные (основные) и зависимые файлы (рис. 1.15). Вход в структуру возможен только через главные файлы. Связываться между собой могут только записи разных типов.

Связи в иерархических и сетевых моделях описываются при проектировании БД. Чаще всего эти связи при хранении данных в БД передаются посредством адресных указателей. Иерархические и сетевые модели БД не накладывают ограничения на тип внутризписной структуры. В принципе она может быть любой, как простой линейной (т. е. состоять только из простых полей, следующих в записи последовательно друг за другом), так и сложной иерархической, включающей в себя различные составные единицы информации (векторы, повторяющиеся группы и т. п.). Конкретные же СУБД накладывают ограничения на допустимые в них информационные единицы, характер связей между ними, порядок их расположения в записи, а также часто имеют и различные количественные ограничения.

Особое место среди структурированных систем занимают системы, построенные на использовании *инвертированных файлов*. Особенность организации данных в них состоит в том, что собственно хранимые данные и информация о связях между логически и физически отделены друг от друга. Основные данные в этих системах хранятся в файлах, записи которых могут иметь сложную структуру. Вся управляющая информация сосредоточена в ассоциаторе. Логическая связь между файлами устанавливается посредством компонента ассоциатора, называемого сетью связи. На рис. 1.16 схематически представлен принцип установления связей в таких системах. Реально, связи устанавливаются не непосредственно с элементами связи, как это изображено на рисунке, а через преобразователь адреса. В системах, построенных на инвертированных файлах, можно передавать связь типа М:М между записями файлов (что не позволяют никакие другие системы). Отделение ассоциативной информации от собственно хранимых данных позволяет изменять связи, не изменяя при этом самих файлов.

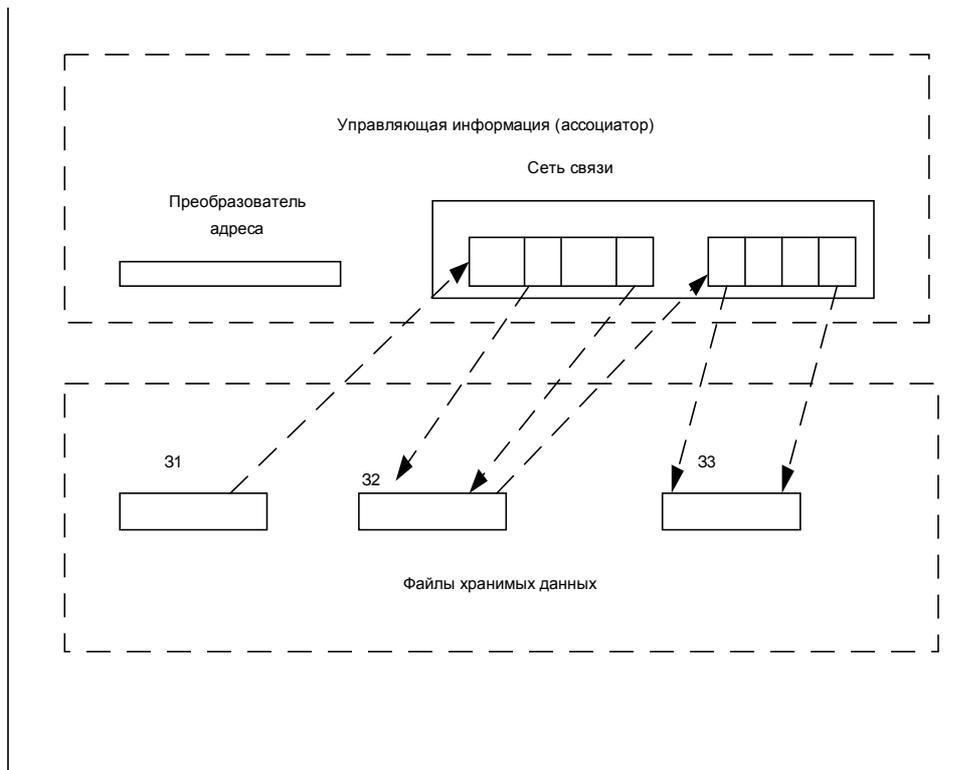


Рис. 1.16. Схема организации данных в системах, основанных на инвертированных файлах

В реляционных моделях (в отличие от иерархических и сетевых) связи между файлами БД определяются динамически в момент выполнения запроса. Эти связи определяются по равенству значений соответствующих полей (полей связи), содержащихся в каждом из связанных файлов.

Другой отличительной чертой реляционных моделей является ограничение на внутрizaписную структуру: записи имеют линейную структуру и могут содержать только простые поля (рис. 1.17).



Рис. 1.17. Схема реляционной модели

Эти отличительные особенности играют решающую роль при проектировании структуры БД.

Восьмидесятые годы были временем интенсивного развития реляционных систем. В 1992 году<sup>3</sup> уровень продаж реляционных СУБД

<sup>3</sup> Первая коммерческая реляционная СУБД была выпущена фирмой Oracle в 1979 году.

впервые превысил уровень продаж нереляционных СУБД. Но до 90% данных предприятий хранилось к этому моменту в нереляционных базах данных на мэйнфреймах.

По *типу хранимой информации* БД делятся на *документальные, фактографические* и *лексикографические*. Среди документальных баз различают *библиографические, реферативные* и *полнотекстовые*. К лексикографическим базам данных относятся различные словари (классификаторы, многоязычные словари, словари основ слов и т. п.).

В системах фактографического типа в БД хранится информация об интересующих пользователя объектах предметной области в виде «фактов» (например, биографические данные о сотрудниках, данные о выпуске продукции производителями и п. т.); в ответ на запрос пользователя выдается требуемая ему информация об интересующем его объекте/объектах или сообщение о том, что искомая информация отсутствует в БД.

В документальных БД единицей хранения является какой-либо документ (например, текст закона или статьи) и пользователю в ответ на его запрос выдается либо ссылка на документ, либо сам документ, в котором он может найти интересующую его информацию.

БД документального типа могут быть организованы по-разному: без хранения и с хранением самого исходного документа на машинных носителях. К системам первого типа можно отнести библиографические и реферативные БД, а также БД-указатели, «отсылающие» к источнику информации. Системы, в которых предусмотрено хранение полного текста документа, так и называются *полнотекстовыми*.

В системах документального типа целью поиска может быть не только какая-то информация, хранящаяся в документах, но и сами документы. Так, возможны запросы типа «сколько документов было создано за определенный период времени» и т. п. Часто в критерий поиска в качестве признаков включаются «дата принятия документа», «кем принят» и другие «выходные данные» документов.

Специфической разновидностью баз данных являются *базы данных форм документов*. Они обладают некоторыми чертами документальных систем (ищется документ, а не информация о конкретном объекте, форма документа имеет название, по которому обычно и осуществляется ее поиск), так и специфическими особенностями (документ ищется не с целью извлечь из него информацию, а с целью использования его в качестве «шаблона»).

В последние годы активно развивается объектно-ориентированный подход к созданию информационных систем. Объектные базы данных организованы как объекты и ссылки к объектам. Объект представляет собой данные и правила, которые оперируют этими данными. Объект включает метод, который является частью определения объекта и запоминается вместе с объектом. В объектных базах данных данные запоминаются как объекты,

классифицированные по типам классов и организованные в иерархическое семейство классов. Класс – коллекция объектов с одинаковыми свойствами. Объекты принадлежат классу. Классы организованы в иерархии.

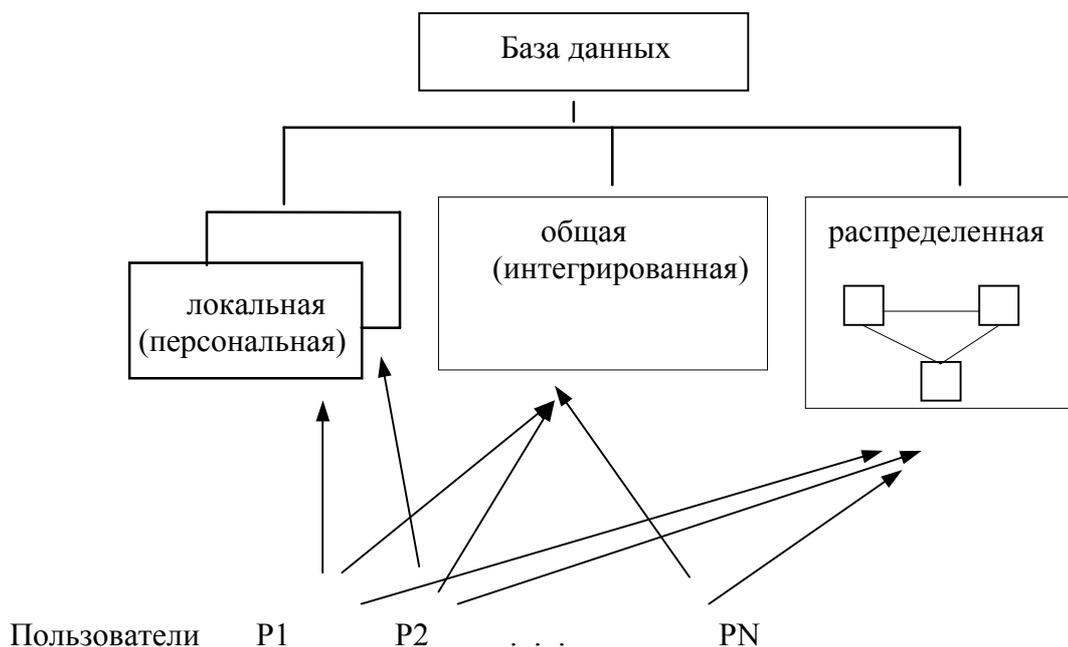


Рис. 1.18. Классификация БД по характеру хранения данных и обращения к ним

По **характеру организации хранения** данных и обращения к ним различают *локальные* (персональные), *общие* (интегрированные, централизованные) и *распределенные* базы данных (Рис. 1.18).

*Персональная база данных* – это база данных, предназначенная для локального использования одним пользователем. Локальные БД могут создаваться каждым пользователем самостоятельно, а могут извлекаться из общей БД.

*Интегрированные и распределенные* БД предполагают возможность одновременного обращения нескольких пользователей к одной и той же информации (многопользовательский, параллельный режим доступа). Это привносит специфические проблемы при их проектировании и в процессе эксплуатации БД. *Распределенные БД* кроме этого имеют характерные особенности, связанные с тем, что физически разные части БД могут быть расположены на разных ЭВМ, а логически, с точки зрения пользователя они должны представлять собой единое целое.

Технологии, которые на первый взгляд вроде бы находятся на разных концах спектра (локальная и распределенная обработка), на самом деле очень близки и различаются практически тем, как поддерживается связь между отдельными частями БД. В случае локальных систем поддержание этой связи не является

централизованной, а в случае распределенных БнД – должна поддерживаться СУБД. Технологией, позволяющей совмещать идеи локальной работы и централизованного поддержания единой БД, является технология тиражирования, при которой средства СУБД позволяют тиражировать отдельные части общей БД, локально использовать их, а потом «согласовывать» отдельные фрагменты БД в рамках единой базы данных.

Концепции централизованной и распределенной обработки данных также не так сильно различаются между собой, как кажется на первый взгляд. Так называемые клиент-серверные системы с «тонким клиентом» очень близки к централизованным базам данных.

Банк данных является сложной человеко-машинной системой, и распределяться по узлам сети могут не только БД, но и другие компоненты БнД. Причем сама БД при этом может быть и не распределенной (например, при обеспечении многопользовательского доступа к централизованной БД в сети). Поэтому будем различать два понятия: распределенные БД и распределенные БнД. При этом под *распределенным БнД* будем понимать банк данных, в котором распределена хотя бы одна любая из его компонент.

Различают *экстенциональные* (ЭБД) и *интенциональные* БД. Интенциональная база данных (ИБД) строится с помощью правил, определяющих ее содержание, а не с помощью явного хранения данных в БД, как в экстенциональных БД.

Например, пусть имеется ЭБД, содержащая таблицу ЛИЧНОСТЬ (PERSON), которая содержит сведения о личности, и среди полей которой есть поля ФАМИЛИЯ\_ИМЯ\_ОТЧЕСТВО (FIO), ПОЛ (SEX). Мы можем построить в этой ЭБД вторую таблицу РОДИТЕЛЬ (PARENT), которая содержит поля ФАМИЛИЯ\_ИМЯ\_ОТЧЕСТВО родителя (FIO) и ИМЯ\_РЕБЕНКА (CHILD). С помощью правил мы можем определить, например, отношение ОТЕЦ (FATHER), просто указав, что отец это родитель, у которого пол – мужской. На ПРОЛОГе это отношение можно определить следующим образом:

$father(X,Y):= person(X,male), parent(X,Y).$

Если выполнить это правило, то получится отношение, которое содержит подмножество кортежей таблицы PARENT, таких, для которых верно указанное условие. Пользователю эти данные выдадутся в виде обычного отношения.

Данное определение ЭБД и ИБД можно расширить и на другой (не реляционный) тип БД, и другой способ задания правил. В более общем виде можно сказать, что информацию можно передать и в виде данных, и в виде программ (строго говоря, программы тоже являются данными, но в русском языке нет подходящего термина, который можно было бы здесь употребить вместо слова "данные").

БД классифицируются *по объему*. Особое место здесь занимают так называемые *очень большие базы данных*. Это вызвано тем, что для

больших баз данных по иному стоят вопросы обеспечения эффективности хранения информации и обеспечения ее обработки.

### 3.2. Классификация СУБД

Рассмотрим теперь ряд классификационных признаков, относящихся к СУБД. *По языкам общения* СУБД делятся на *открытые, замкнутые и смешанные*. *Открытые системы* – это системы, в которых для обращения к базам данных используются универсальные языки программирования. *Замкнутые системы* имеют собственные языки общения с пользователями БД.

По числу уровней в архитектуре различают одноуровневые, двухуровневые, трехуровневые системы. В принципе возможно выделение и большего числа уровней. Под архитектурным уровнем СУБД понимают функциональный компонент, механизмы которого служат для поддержки некоторого уровня абстракции данных (логический и физический уровень, а также "взгляд" пользователя – внешний уровень).

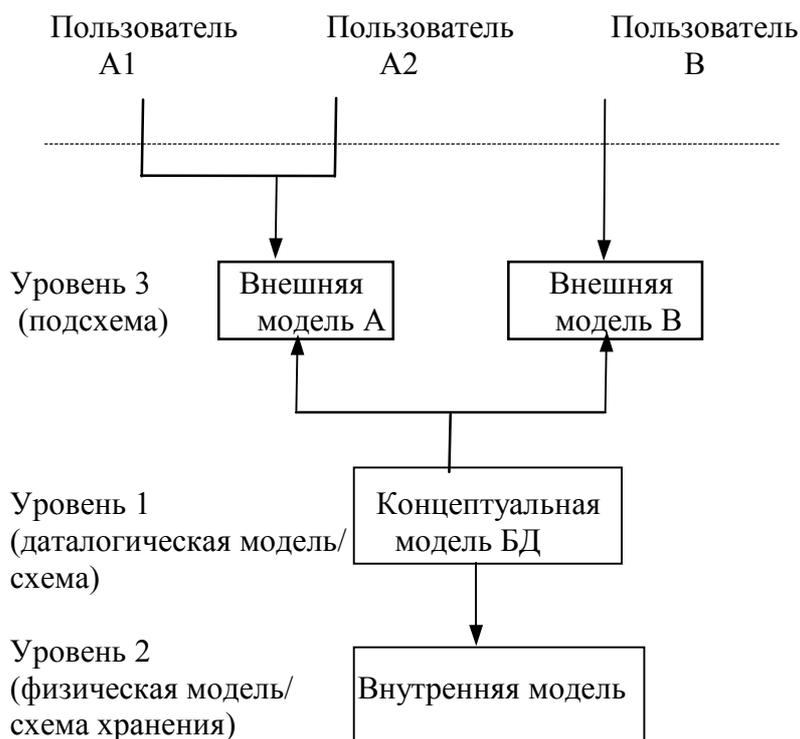


Рис 1.19. Классификация СУБД по числу уровней в архитектуре (пример трехуровневой архитектуры)

На рис. 1.19 сделана попытка совместить терминологию, встречающуюся в разных литературных источниках. В литературе широко используются понятия "внешняя", "концептуальная" и

внутренняя" модель/уровень, а также "логический" и "физический"<sup>4</sup> уровень, а кроме того "внешняя схема", "подсхема", "схема хранения", просто "схема" и проч. Понятие схема с тем или иным уточнением обычно относится к описанию соответствующего уровня описания данных.

Нумерация уровней на рисунке условна, но тем не менее отражает их значимость (внутренняя модель может быть построена только на основе концептуальной; эти два уровня могут быть совмещены, но поддерживаются СУБД всегда; внешний уровень в архитектуре СУБД может отсутствовать).

**По выполняемым функциям** СУБД делятся на *информационные* и *операционные*. Информационные СУБД позволяют организовать хранение информации и доступ к ней. Для выполнения более сложной обработки необходимо писать специальные программы. Операционные СУБД выполняют достаточно сложную обработку, например, автоматически позволяют получать агрегированные показатели, не хранящиеся непосредственно в базе данных, могут изменять алгоритмы обработки и т.д.

**По сфере возможного применения** различают *универсальные* и *специализированные*, обычно проблемно-ориентированные СУБД.

Системы управления базами данных поддерживают разные типы данных. Набор типов данных, допустимых в разных СУБД, различен. Кроме того, ряд СУБД позволяет разработчику добавлять новые типы данных и новые операции над этими данными. Такие системы называются *расширяемыми* системами баз данных (РСБД).

Дальнейшим развитием концепции РСБД являются *системы объектно-ориентированных баз данных (СООБД)*, обладающие достаточно мощными выразительными возможностями, чтобы непосредственно моделировать сложные объекты.

**По «мощности»** СУБД делятся на *«настольные»* и *«корпоративные»*. Характерными чертами настольных СУБД являются сравнительно невысокие требования к техническим средствам, ориентация на конечного пользователя, низкая стоимость.

Корпоративные СУБД обеспечивают работу в распределенной среде, высокую производительность, поддержку коллективной работы при проектировании систем, имеют развитые средства администрирования и более широкие возможности поддержания целостности.

---

<sup>4</sup> Во многих современных CASE-средствах концептуальной моделью называется ER-модель предметной области, а физической - модель, поддерживаемую конкретной СУБД. Если первое еще можно считать удачным использованием термина (так как ER-модель действительно отражает общую «концепцию» системы, то второе - крайне неудачно, так как ни о какой «физике» речь здесь не идет.

В связи с выше перечисленными чертами корпоративных СУБД очевидно, что эти системы сложны, дороги, требуют значительных вычислительных ресурсов.

### Сравнение «настольных» и «корпоративных» СУБД

Табл. 1.1

Критерий	насто́льн ые	корпоративны е
Простота использования	+	
Стоимость программного обеспечения	+	
Стоимость эксплуатации	+	
Функциональные возможности, в т. ч.: <ul style="list-style-type: none"> <li>■ возможности администрирования</li> <li>■ возможности работы с Интернет/Интранет</li> </ul> и др.		+
Надежность функционирования		+
Поддерживаемые объемы данных		+
Быстродействие		+
Возможности масштабирования		+
Работа в гетерогенной среде		+

Системы обоих классов интенсивно развиваются, причем некоторые тенденции развития присущи каждому из этих классов. Прежде всего, это использование высокоуровневых средств разработки приложений (что раньше было присуще, в основном, настольным системам), рост производительности и функциональных возможностей, работа в локальных и глобальных сетях и др.

Наиболее известными из корпоративных СУБД являются Oracle, Informix, Sybase, MS SQL Server, Progress и некоторые другие.

Наблюдается связь между классом СУБД и используемой операционной системой. Системы под UNIX позиционируются как корпоративные распределенные системы. Сейчас в этот сектор «пробивается» Windows NT и заменяющая ее Windows 2000.

*По ориентации на преобладающую категорию пользователей можно выделить СУБД для разработчиков и для конечных пользователей.* Системы, относящиеся к первому классу, должны иметь качественные компиляторы и позволять создавать «отчуждаемые» программные продукты, обладать развитыми средствами отладки, включать средства документирования проекта и обладать другими возможностями, позволяющими создавать эффективные сложные

системы. Основными требованиями, предъявляемыми к системам, ориентированным на конечного пользователя, являются: удобство интерфейса, высокий уровень языковых средств, наличие интеллектуальных модулей подсказок, повышенная защита от непреднамеренных ошибок («защита от дурака») и т. п.

### 3.3. Классификационные группировки, относящиеся к БД в целом

Следующая группа признаков классификации связана с **банком данных в целом**. По условиям предоставления услуг различают *бесплатные* и *платные* банки данных. Платные БД в свою очередь делятся на *бесприбыльные* и *коммерческие*. Бесприбыльные банки данных функционируют на принципе самокупаемости и не ставят своей целью получение прибыли. Это обычно БД социально значимой информации, имеющей широкий круг пользователей, или научной, библиотечной информации. Основной целью создания коммерческих банков данных является получение прибыли от информационной деятельности.

Информационные системы различаются *по характеру преобладающей обработки информации*. В одних в основном реализуется большое число достаточно простых запросов (такие системы получили название *OLTP* (On-Line Transaction Processing) – *системы оперативной обработки транзакций*). В других, напротив, требуется сложная аналитическая обработка данных (для такого класса систем стал использоваться термин *OLAP* (On-line Analytical Processing)).

Термин OLAP является сравнительно новым и в разных литературных источниках трактуется иногда по-разному. Этот термин часто отождествляют с поддержкой принятия решений (DSS (Decision Support Systems) – системы поддержки принятия решения). А в качестве синонима для последнего термина используют Data Warehousing – хранилища (склады) данных, понимая под этим набор организационных решений, программных и аппаратных средств для обеспечения аналитиков информацией на основе данных из систем обработки транзакций нижнего уровня и других источников.

«Склады данных» позволяют обрабатывать данные, накопленные за длительные периоды времени. Эти данные являются разнородными (и не обязательно структурированными). Для «складов данных» присущ многомерный характер запросов. Огромные объемы данных, сложность структуры как данных, так и запросов требует использования специальных методов доступа к информации.

В других источниках понятие Системы Поддержки Принятия Решений (СППР) считается более широким. Хранилища данных и средства оперативной аналитической обработки могут служить одними из компонентов архитектуры СППР.

Иногда различают "OLAP в узком смысле" – это системы, которые обеспечивают только выборку данных в различных разрезах, и "OLAP в широком смысле", или просто OLAP, включающей в себя:

- поддержку нескольких пользователей, редактирующих БД;
- функции моделирования, в том числе вычислительные механизмы получения производных результатов, а также агрегирования и объединения данных;
- прогнозирование, выявление тенденций и статистический анализ.

Естественно, что каждый из этих типов ИС требует специфической организации данных, а так же специальных программных средств, обеспечивающих эффективное выполнение стоящих задач.

Для обеспечения быстрой обработки данных при их анализе используются разнообразные приемы. Одним из них является организация данных в виде так называемых многомерных БД (MDD). Информация в MDD хранится не в виде индексированных записей в таблицах, а в форме логически упорядоченных массивов. Единой общепризнанной многомерной модели хранения данных не существует. В MDD отсутствует стандартизованный метод доступа к данным, и они могут отвечать требованиям специфической аналитической обработки данных.

*Табл. 1.2*

### Сравнение OLTP и OLAP

Характеристика	OLTP	OLAP
Преобладающие операции	Ввод данных, поиск	Анализ данных
Характер запросов	Много простых транзакций	Сложные транзакции
Хранимые данные	Оперативные, детализированные	Охватывающие большой период времени, агрегированные
Вид деятельности	Оперативная, тактическая	Аналитическая, стратегическая
Тип данных	Структурированные	Разнотипные

Хранилища данных могут быть разбиты на два типа: корпоративные хранилища данных (enterprise data warehouses) и киоски данных (data marts).

Корпоративные хранилища данных содержат информацию, относящуюся ко всей корпорации и собранную из множества оперативных источников для консолидированного анализа. Обычно такие хранилища охватывают целый ряд аспектов деятельности

корпорации и используются для принятия как тактических, так и стратегических решений.

Киоски данных содержат подмножество корпоративных данных и строятся для отделов или подразделений внутри организации. Киоски данных часто строятся силами самого отдела и охватывают конкретный аспект, интересующий сотрудников данного отдела. Киоск данных может получать данные из корпоративного хранилища (зависимый киоск), или, что более распространено, данные могут поступать непосредственно из оперативных источников (независимый киоск).

Киоски и хранилища данных строятся по сходным принципам и используют практически одни и те же технологии.

**По степени доступности** БД делятся на *общедоступные* и *с ограниченным кругом пользователей*.

**По охвату** БД могут классифицироваться в свою очередь в разных «разрезах»:

- *территориальный*
  - ◆ *всемирный*
  - ◆ ...
  - ◆ *страна*
  - ◆ ...
  - ◆ *город*
  - ◆ ...
- *временной*
- *ведомственный*
- *проблемный (тематический)*

Территориальный и ведомственный признаки классификации могут относиться не только к информации, хранящейся БД, но и к кругу обслуживаемых пользователей.

**По характеру взаимодействия с пользователями** (кто инициализирует действия) БД делятся на:

- активные БД
- пассивные БД.

В пассивных БД ведущая роль принадлежит пользователю. В активных – система может самостоятельно менять поведение. В последнее время термин «активная база данных» стал часто использоваться для систем, использующих триггеры.

**По форме собственности** БД делятся на:

- государственные
- негосударственные
  - ◆ частные
  - ◆ групповые
  - ◆ личные.

В литературе встречаются и другие аспекты классификации банков данных, но названные являются наиболее значимыми.

## 4. Уровни моделей и этапы проектирования бд

### 4.1. Уровни моделей

В базе данных отражается информация об определенной предметной области. *Предметной областью* называется часть реального мира, представляющая интерес для данного исследования.

В автоматизированных информационных системах отражение предметной области обеспечивается посредством информационной модели. Мы будем рассматривать далее вопросы проектирования баз данных для СУБД, поддерживающих структурированные модели данных. В зависимости от аспекта рассмотрения (уровня абстракции) различают модели данных нескольких уровней. Число реально выделенных и самостоятельно поддерживаемых уровней моделей будет зависеть от особенностей СУБД.

Чаще всего выделяют три уровня моделей: логический, физический и внешний.

*Даталогическая* (datalogical) модель (ДЛМ) базы данных является моделью логического уровня и представляет собой отображение логических связей между элементами данных безотносительно к среде хранения. Эта модель строится в терминах информационных единиц, допустимых в той конкретной СУБД, в среде которой мы проектируем базу данных. Этап создания ДЛМ называется *даталогическим проектированием*. Описание логической структуры базы данных на языке СУБД называется *схемой*.

Для привязки даталогической модели к среде хранения используется модель данных физического уровня (для краткости часто называемая *физической моделью*). Эта модель определяет используемые запоминающие устройства, способы физической организации данных в среде хранения. Модель физического уровня также строится с учетом возможностей, предоставляемых СУБД. Описание физической структуры базы данных называется *схемой хранения*. Соответствующий этап проектирования БД называется физическим проектированием. СУБД обладают разными возможностями по физической организации данных, в связи с чем сложность и трудоемкость физического проектирования, набор выполняемых шагов различаются для конкретных систем. К числу работ, выполняемых на этапе физического проектирования, относятся: выбор типа носителя, способа организации данных, методов доступа, определение размера физического блока, управление размещением данных на внешнем носителе, управление свободной памятью, определение целесообразности сжатия данных и используемых методов сжатия, оценка физической модели данных. К физическому проектированию относятся и проблемы, связанные с буферизацией

Независимо от того, поддерживаются в явном виде отдельно модели логического и физического уровня, с точки зрения методологии

все равно можно выделить эти уровни моделей и соответствующие им этапы проектирования баз данных.

В некоторых СУБД, помимо описания общей логической структуры базы данных, имеется возможность описать логическую структуру БД с точки зрения конкретного пользователя. Такая модель называется *внешней*, а ее описание называется *подсхемой*. Если СУБД "поддерживает" схему, схему хранения и подсхему, то она является СУБД с трехуровневой архитектурой.

Внешняя модель не всегда является точным подмножеством схемы. Некоторые СУБД допускают различия в типах данных, определенных в схеме и подсхеме, и обеспечивают их преобразование, позволяют задавать различный логический порядок следования элементов в схеме и подсхеме, обеспечивают введение в подсхему виртуальных полей и т. д.. Если определена подсхема, то пользователь имеет доступ только к тем данным, которые отражены в соответствующей подсхеме, что является одним из способов защиты информации от несанкционированного доступа.

В подсхемах часто задается не только логическая структура части базы данных с точки зрения конкретного пользователя (приложения), но и допустимые режимы обработки в рамках этой подсхемы, что служит дополнительным механизмом защиты информации от разрушения.

Использование аппарата подсхем облегчает работу пользователя, так как он должен знать структуру не всей базы данных, а только той ее части, которая имеет непосредственное отношение к нему.

В тех случаях, когда СУБД в явном виде не поддерживает подсхемы, перечисленные функции могут выполнять другие компоненты системы. Близким к понятию подсхемы является понятие *view* (взгляд), которое в настоящее время широко используется в англоязычной литературе по реляционным СУБД.

Выше мы говорили о трех уровнях моделей, которые поддерживаются СУБД. Но для того, чтобы спроектировать структуру базы данных, необходима исходная информация о предметной области. Желательно, чтобы эта информация была представлена в формализованном виде. Такое формализованное описание предметной области будем называть *инфологической (infological) моделью предметной области (ИЛМ)*<sup>5</sup> или *концептуальной моделью (КМ)*. Информация, требуемая для проектирования БД, мало зависит от особенностей СУБД. Более того, для проектирования ИС с "небанковской" организацией (но использующей структурированное представление данных) обычно требуется та же исходная информация. Поэтому *концептуальная схема* представляет собой описание

---

<sup>5</sup> В предыдущих изданиях учебника [ , ] для этого уровня моделирования использовался термин «инфологическое». Последнее время в литературных источниках этот термин используется редко. В данном учебном пособии термины «инфологическое» и «концептуальное» моделирование используются как синонимы.

предметной области, выполненное без жесткой ориентации на используемые в дальнейшем программные и технические средства. Концептуальная схема должна отражать специфику предметной области, а не структуру БД. Иногда в концептуальную схему добавляют информацию, отображающую чисто языковые характеристики, такие как наличие синонимов, длина реквизитов и др. Это, скорее всего, вызвано следующими основными причинами: 1) нежеланием вводить еще один уровень моделей, 2) трудностью отделения языковых проблем от других, так как анализируемая предметная область обычно представлена в какой-либо знаковой системе, и анализу обычно подвергается именно это представление, а не непосредственно сама ПО.

#### 4.2. Взаимосвязь этапов проектирования БД.

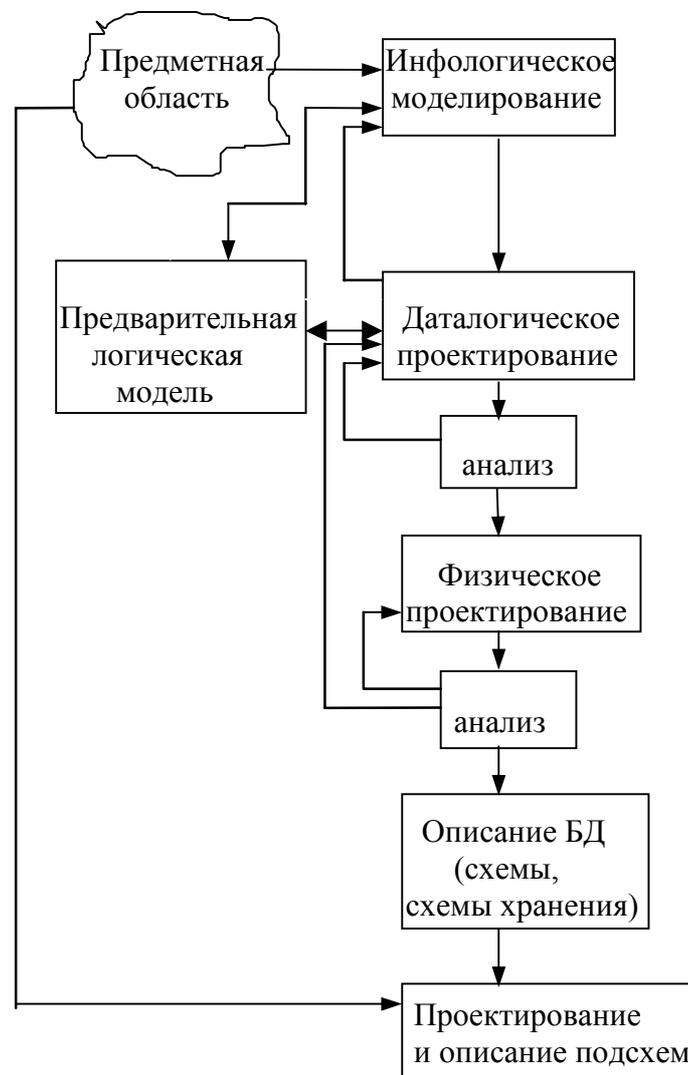


Рис. 1. 22. Взаимосвязь этапов проектирования БД

Начальным шагом проектирования ИС является построение инфологической модели предметной области. Предварительная инфологическая модель строится еще на предпроектной стадии и затем

уточняется на более поздних стадиях проектирования. Затем на ее основе строится даталогическая модель. Физическая и внешняя модель после этого могут строиться в любой последовательности по отношению друг к другу, в том числе и параллельно. На рис. 1.20 изображена взаимосвязь этапов проектирования БД. Как видно из рисунка, при проектировании БД возможен возврат на предыдущие уровни. При этом возможны два вида возвратов: первый тип обусловлен необходимостью пересмотра результата проектирования (например, для улучшения полученных характеристик, "обхода" ограничений и т. п.), второй тип вызван необходимостью уточнения предыдущей модели (обычно – инфологической) с целью получения дополнительной информации для проектирования или при выявлении противоречий в модели.

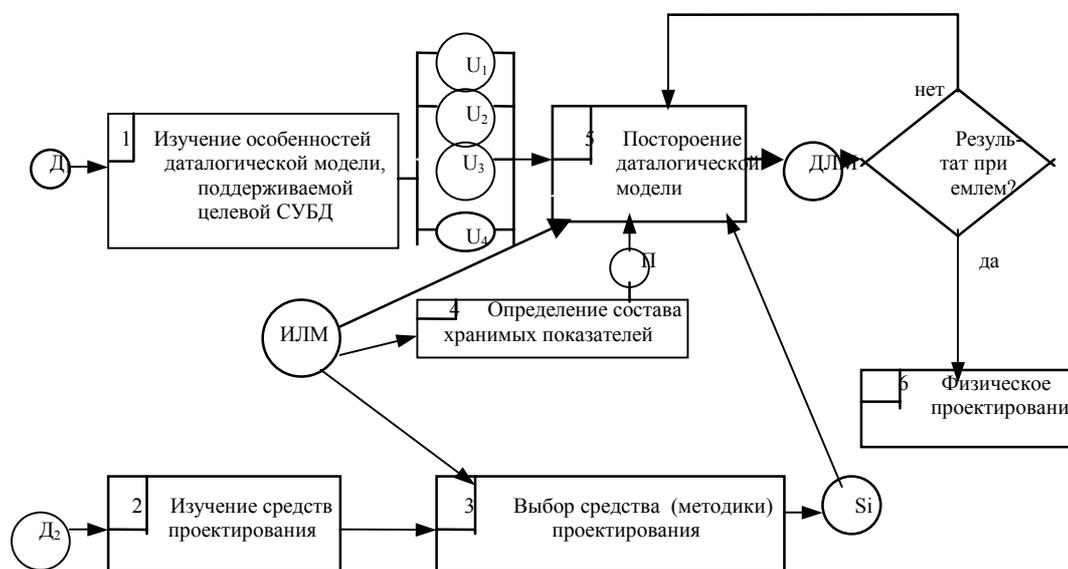


Рис. 1.21. Технологическая сеть проектирования для этапа даталогического проектирования

- Д1 - документация по СУБД
- Д2 - документация по средствам проектирования
- U1 - набор допустимых даталогических конструкций
- U2 - операторы ЯМД
- U3 - ограничения, налагаемые СУБД на ДЛМ
- U4 - возможности физической организации данных
- ДЛМ - даталогическая модель
- ИЛМ - инфологическая модель
- П - перечень хранимых показателей
- Si - выбранное средство проектирования

На рис. 1.21 и 1.22 изображены укрупненные технологические сети проектирования для этапов даталогического и физического проектирования.<sup>6</sup> Как видно из этих рисунков, результат предыдущего этапа проектирования используется на входе следующего этапа.

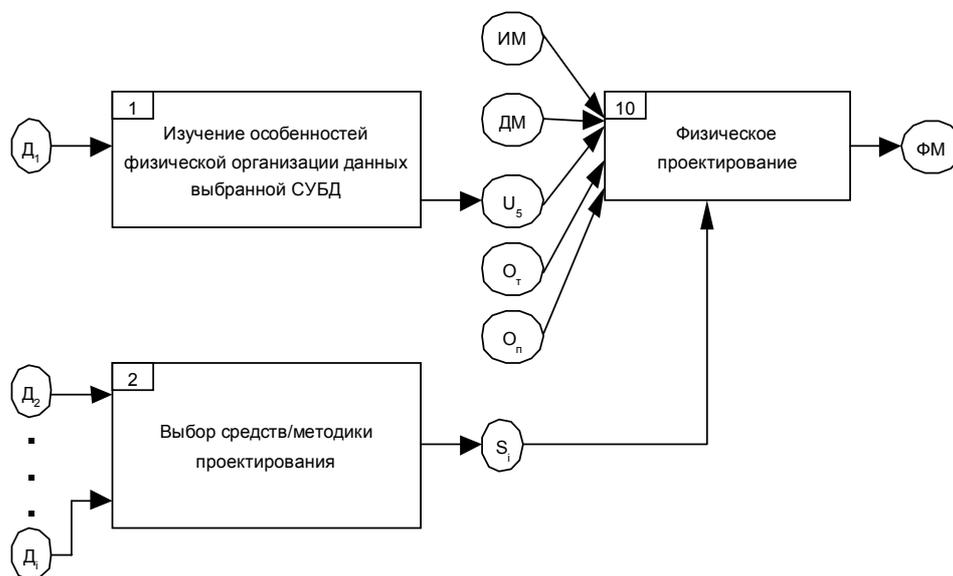


Рис. . . Технологическая сеть проектирования для этапа физического проектирования

- Д<sub>1</sub> - документация по СУБД
- Д<sub>2</sub> – Д - документация по средствам проектирования
- U<sub>5</sub> - возможности физической организации данных
- ДМ- даталогическая модель
- ИМ- инфологическая модель
- S<sub>i</sub> - выбранное средство проектирования
- O<sub>т</sub> - ограничения на используемые технические средства
- O<sub>п</sub> – ограничения со стороны пользователей/процессов

### 4.3. Факторы, влияющие на проектирование БД

Как было отмечено выше, на стадии инфологического моделирования должна быть собрана и представлена в надлежащем виде вся информация, необходимая и достаточная для дальнейшего проектирования БД. Для того, чтобы было понятно, какая информация должна фиксироваться при описании предметной области, перечислим основные из факторов, оказывающих влияние на проектирование структуры БД:

1. специфика предметной области

<sup>6</sup> Следует обратить внимание на различие терминологии, используемой как в литературных источниках, так и в конкретных CASE-системах. Так, во многих CASE-системах [ ] ER-модель предметной области называется концептуальной схемой, а представление логической структуры целевой базы данных - физической моделью.

- 1.1. особенности отображаемых объектов, характер связи между объектами предметной области.
- 1.2. «размер» системы (объем хранимых данных)
2. особенности требуемой обработки информации
  - 2.1. характеристика запросов (критерий поиска, частота запроса; состав реквизитов, выдаваемых в ответ, упорядоченность ответа, частота совместного использования реквизитов и т. п.)
  - 2.2. требования к защите информации
  - 2.3. ограничения по времени реакции системы на каждый из запросов, что в свою очередь определяется несколькими факторами, такими как: режим выполнения запроса (интерактивный, пакетный, в реальном масштабе времени), статус запроса и др.
3. Характеристика пользователей системы
  - 3.1. важность/статус, приоритеты,
  - 3.2. число пользователей
  - 3.3. распределение функций между пользователями; степень пересечения информационных потребностей пользователей
  - 3.4. приоритеты пользователей в оценке значимости факторов, влияющих на проектирование БД
  - 3.5. Технология обработки данных
  - 3.6. возможность/необходимость работы в распределенной среде, в том числе необходимость поддерживать связь с «мобильными» компьютерами
  - 3.7. «доступные» технологии обработки данных
4. Состояние существующей системы обработки информации
  - 4.1. наличие существующей автоматизированной системы обработки информации
  - 4.2. объем имеющихся «наработок»
  - 4.3. наличие технических и программных средств, их состояние
  - 4.4. соотношение объемов «существующей» и «новой» частей проектируемой системы
  - 4.5. затраты на перевод имеющейся системы на новую основу
5. Возможности, предоставляемые используемыми (выбранными для реализации проекта) техническими и программными средствами
  - 5.1. поддерживаемые структуры данных; ограничения, накладываемые программным обеспечением
  - 5.2. ограничения по объему памяти
  - 5.3. быстродействие технических средств
  - 5.4. «производительность» Программного Обеспечения
  - 5.5. особенности языков манипулирования данными
6. Трудоемкость проектирования
7. Финансовые возможности
8. Квалификация кадров

- 8.1. разработчиков
  - 8.2. пользователей
  - 9. Используемые методики проектирования
    - 9.1. наличие средств автоматизации проектирования
    - 9.2. используемый алгоритм проектирования
  - 10. Субъективные факторы
    - 10.1. мода
    - 10.2. привычки и предпочтения
- Более подробно влияние некоторых из перечисленных выше факторов будет рассмотрено далее, по мере изложения вопросов проектирования БД.

### **Контрольные вопросы**

1. Дайте определение банка данных.
2. Назовите основные преимущества банков данных.
3. Назовите основные недостатки банков данных.
4. Каковы предпосылки создания БД?
5. Какие требования предъявляются к банкам данных?
6. Какие компоненты включаются в состав банка данных?
7. Что называется Системой Управления Базой данных?
8. Что называется базой данных?
9. Дайте классификацию языковых средств СУБД.
10. Какие поколения языковых средств Вы знаете? Дайте краткую характеристику языковым средствам каждого из поколений.
11. Назовите принципы, по которым построены языки четвертого поколения.
12. Перечислите компоненты языка четвертого поколения.
13. Приведите примеры процедурных и не процедурных языков. В чем основные отличия между языками этих классов.
14. Назовите основные отличительные особенности банков данных.
15. Какие технические средства необходимы для реализации банка данных?
16. Какие типы ЭВМ чаще всего используются для реализации банков данных?
17. Перечислите основные признаки классификации банков данных.
18. В чем разница между системами со структурированными и неструктурированными базами данных?
19. Охарактеризуйте основные классы СУБД.
20. СУБД каких классов являются в настоящее время наиболее распространенными?
21. Каковы основные тенденции развития СУБД наблюдаются в настоящее время?
22. Сравните системы типа OLTP и OLAP.
23. Перечислите основные отличия корпоративных и настольных СУБД.

24. Сравните локальные, интегрированные и распределенные БД.
25. Перечислите этапы проектирования баз данных.
26. Что называется схемой, подсхемой и внешней схемой?
27. Какую роль выполняет подсхема? Какие преимущества дает ее использование?
28. Что называется словарем данных, репозиторием?
29. Охарактеризуйте взаимодействие компонентов БнД при работе с системой.
30. Что называется инфологической моделью?
31. Является ли инфологическое моделирование этапом, присущим только проектированию баз данных?
32. Какая информация является исходной для построения концептуальной модели?
33. Кто должен создавать концептуальную модель и почему?
34. Какие требования предъявляются к инфологической модели?
35. Что называется даталогической моделью?
36. Какая информация является исходной для построения даталогической модели?
37. Какие вопросы решаются на стадии даталогического моделирования?
38. Изобразите технологическую сеть проектирования для стадии даталогического моделирования.
39. Что называется физической моделью?
40. Какая информация является исходной для построения физической модели?
41. Какие вопросы решаются на стадии физического моделирования?
42. Какие факторы влияют на проектирование БД?
43. Перечислите основные категории пользователей банков данных.
44. Кого называют конечными пользователями?
45. Кого называют администраторами банка данных?
46. Перечислите основные функции администратора банка данных.
47. В каком порядке должны выполняться этапы проектирования БнД?

## 5. Литература

1. Hansen Gary W., Hansen James V. Database Management and Design. – New Jerse: Pretice-Hall, Inc., - 1985. – 420 с.
2. Martin James. Fours-Generation Languages, V. 2.- Pretice-Hall, Inc., - 1986, - 500 с.
3. Диго С. М. Проектирование и использование баз данных. – М.: Финансы и статистика, 1995.
4. Закон "О правовой охране программ для электронных вычислительных машин и баз данных" № 3523-1, 23.09.92
5. Общеотраслевые руководящие материалы по созданию банков данных. М.:ГКНТ, 1982