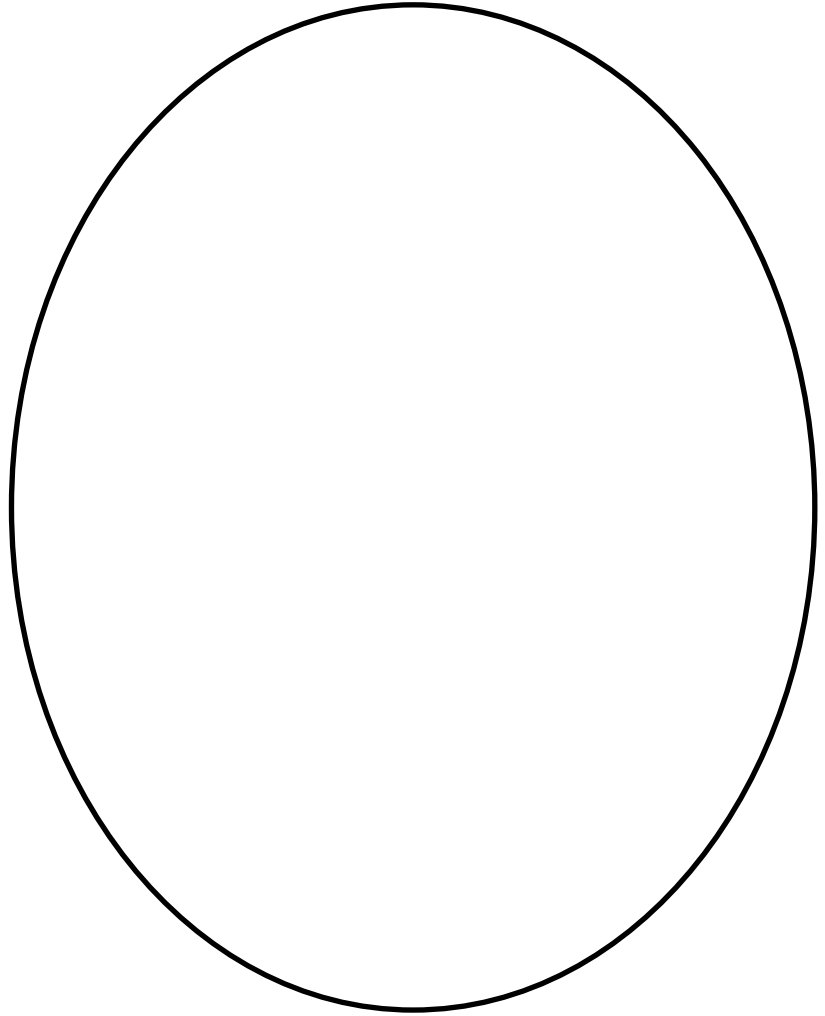
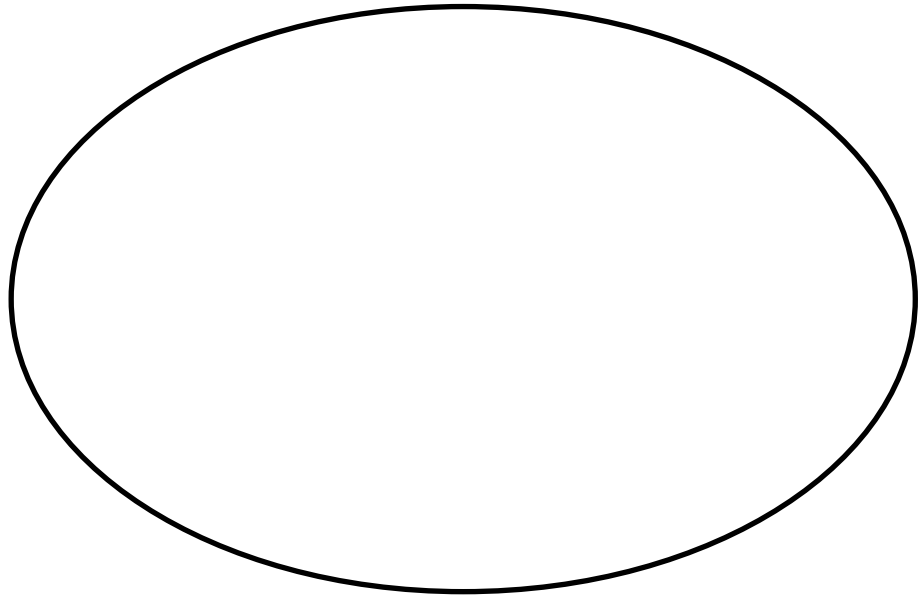
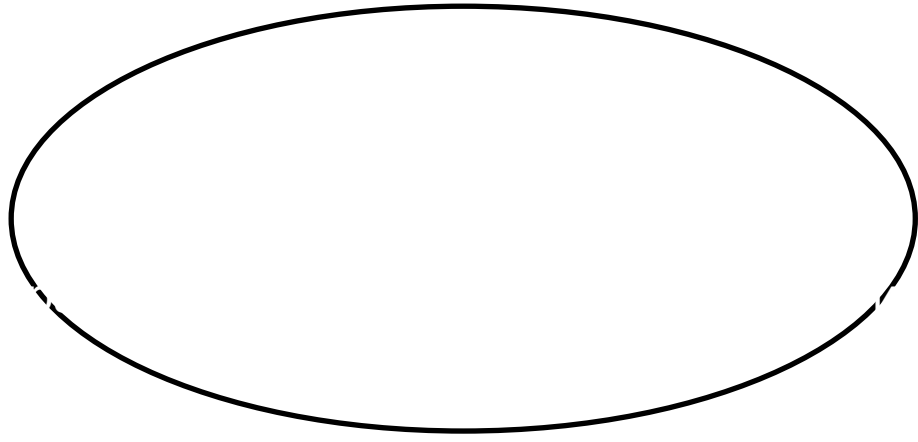


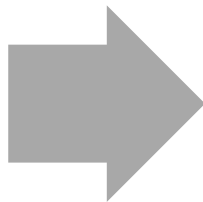
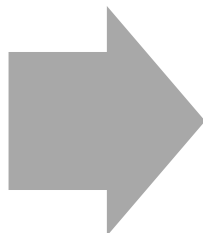
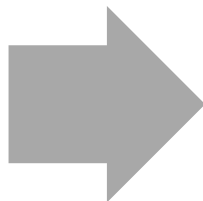
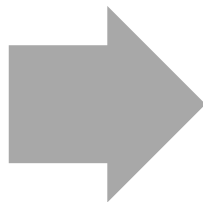
# Базы данных

## Управление данными

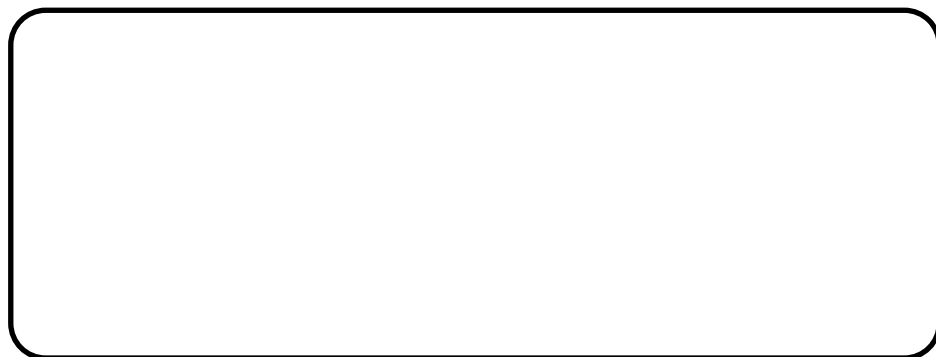
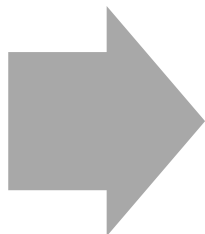
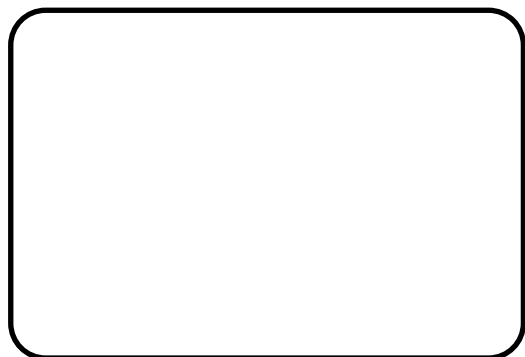
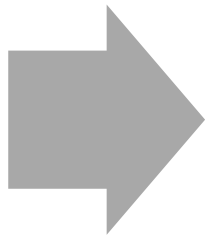
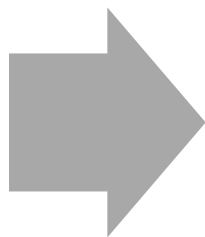




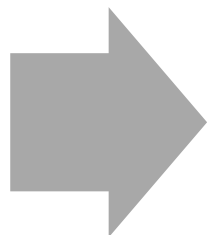
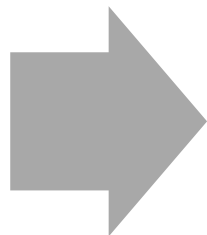
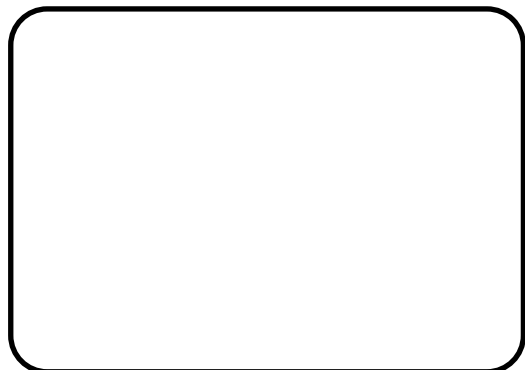
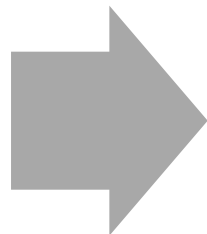
# Основные требования, предъявляемые к банкам данных



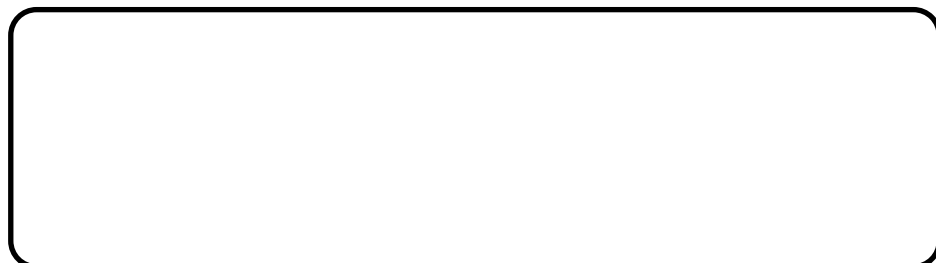
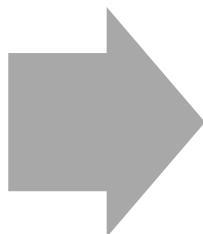
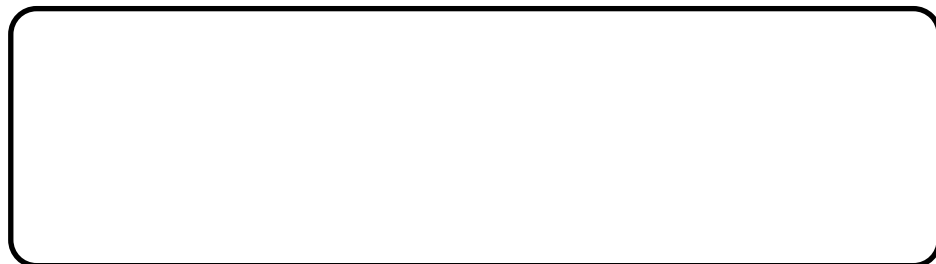
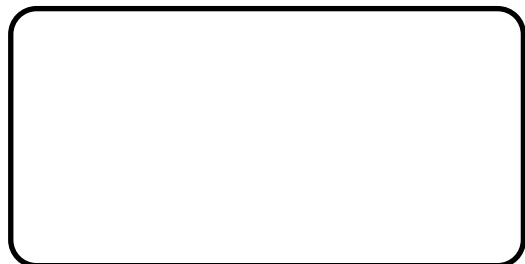
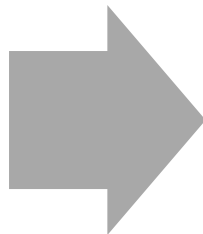
# Основные требования, предъявляемые к банкам данных



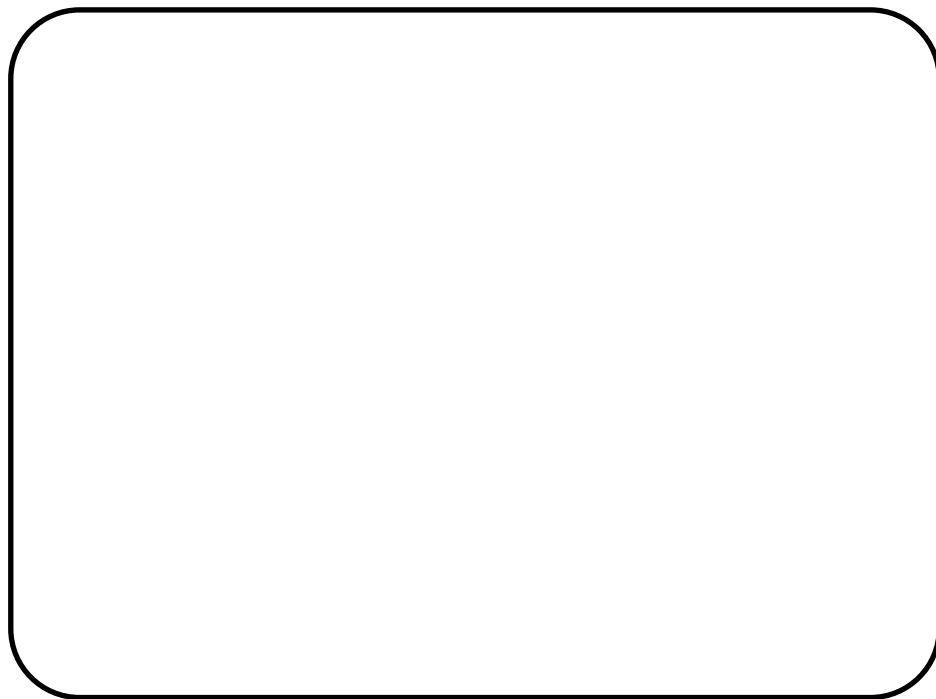
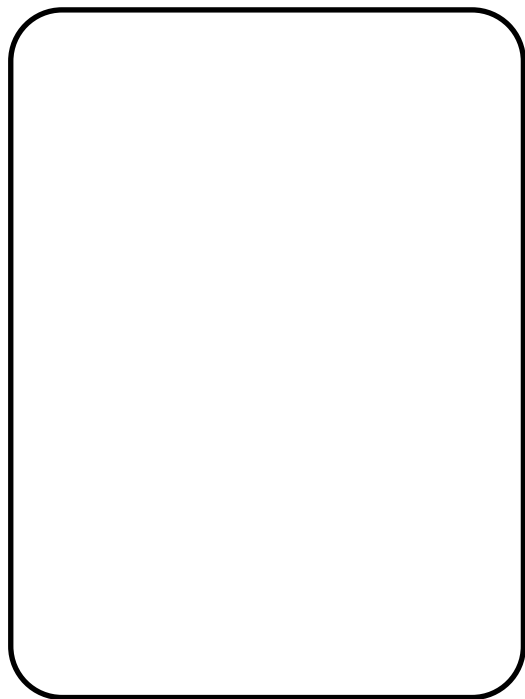
# Основные требования, предъявляемые к банкам данных

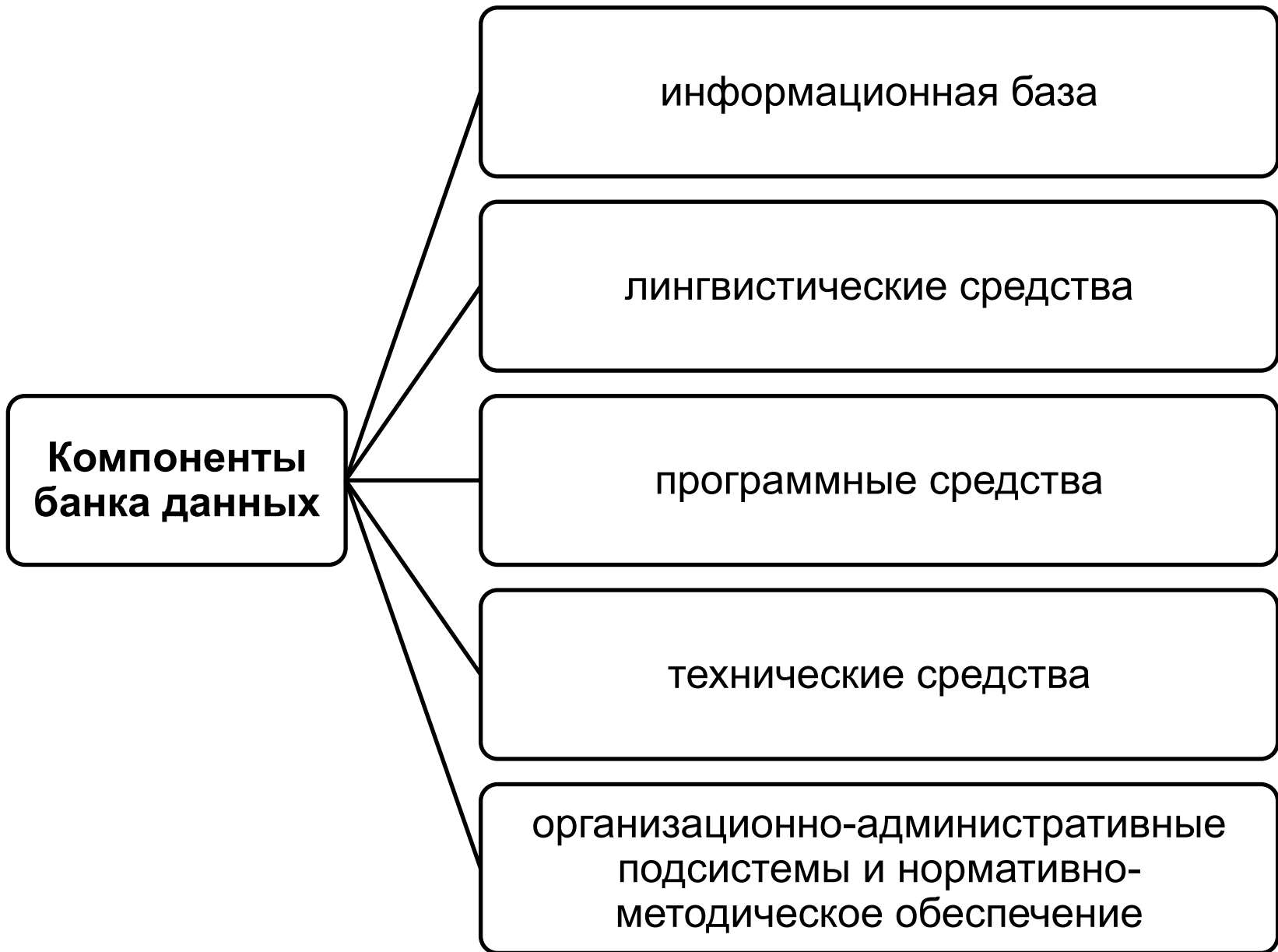


# Основные требования, предъявляемые к банкам данных



# Основные требования, предъявляемые к банкам данных







## **Информационная база**

Данные, отражающие состояние определенной предметной области и используемые информационной системой

# Уровни представления данных

- Концептуальный уровень
  - Обобщенный взгляд на данные с позиций предметной области
  - Пользователь затрагивает
  - Разработчик ИС организации (размещения) данных во внешней памяти
- Логический (внешний) уровень
  - Глобальное представление БД, определяет необходимые условия для организации хранения данных на внешних запоминающих устройствах
  - Прикладной программист
- Физический (внутренний) уровень
  - СУБД



# Лингвистические средства

В состав СУБД включаются

## Этапы работы с базой данных

- Описание БД
- Описание частей БД, необходимых для конкретных приложений (задач, групп задач)
- Программирование задач или описание запросов в соответствии с правилами конкретного языка и использованием языковых конструкций для обращения к БД
- Загрузка БД

## Работы с базой данных

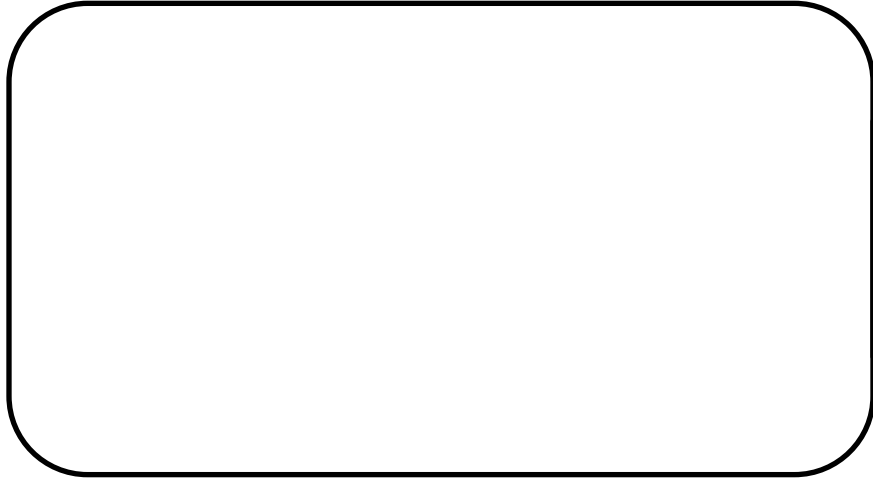
- определение схемы базы данных
- характеристики хранимых и виртуальных данных
- параметры организации хранения данных в памяти
- средства поддержки целостности базы данных, ограничения доступа, секретности

- средства запросов к базе данных и поддержания базы данных (добавление, удаление, обновление данных, создание и уничтожение БД, изменение определений БД, обеспечение запросов к справочнику БД)

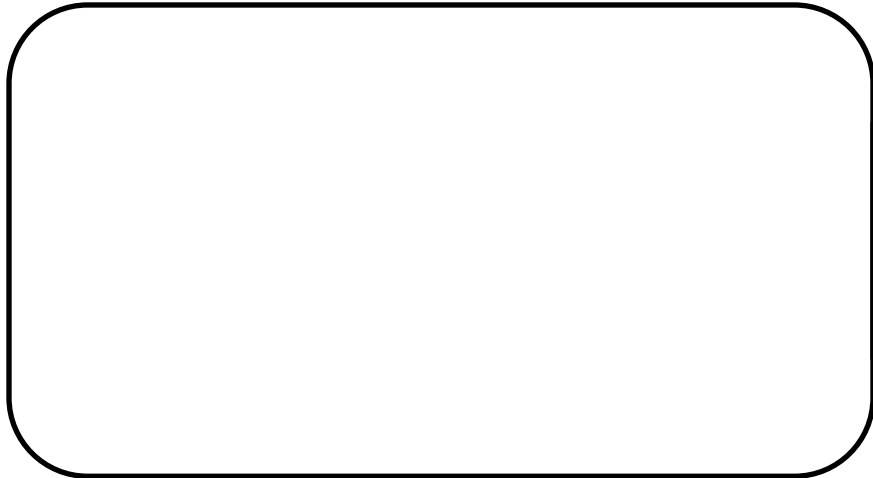
# Языковые средства работы с базой данных



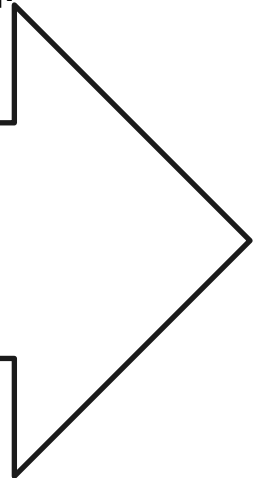
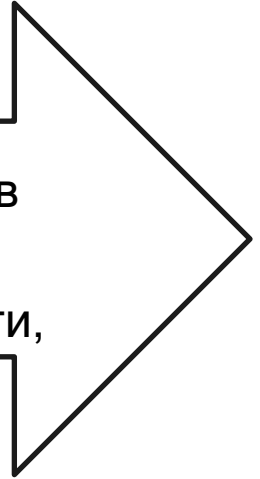
# Характеристики языков



•отражают возможности описания данных, средств представления запроса, обновления, поддержки целостности и секретности, включения в языки программирования, управления форматом ответов, средств запроса к словарю данных БД и т. д.



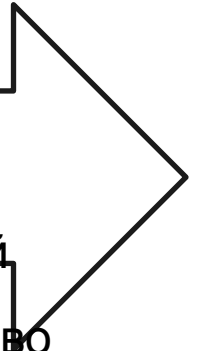
•определяются такими свойствами, как полнота, селективная мощность, простота изучения и использования, степень процедурности и модульности, унифицированность, производительность и эффективность



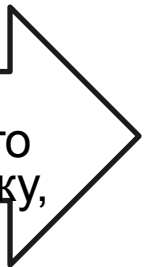
## Свойства языков запросов



- характеризует возможность выбора данных по разным критериям
- язык с большей селективной мощностью позволяет сформулировать большинство запросов так, что ответ на них содержит меньше ненужных данных



- субъективная оценка, охарактеризована степенью его близости к естественному языку, требуемым для его освоения временем и необходимым уровнем подготовки пользователя





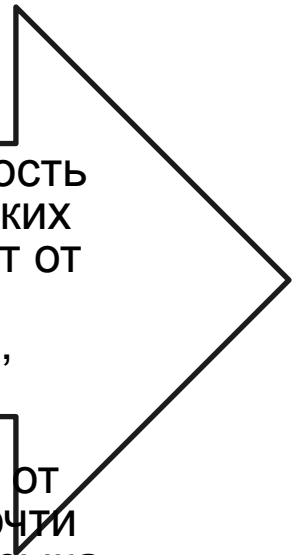
# Свойства языков запросов



- определяется присущими реляционной модели свойствами, в частности, полным отделением логической структуры данных от структур хранения и стратегий доступа



- характеризует возможность существования нескольких уровней языка и зависит от специфических свойств математической теории, лежащей в его основе
- языки, не обладающие модульностью, требуют от пользователя знания почти всего объема средств языка, что усложняет процесс их изучения

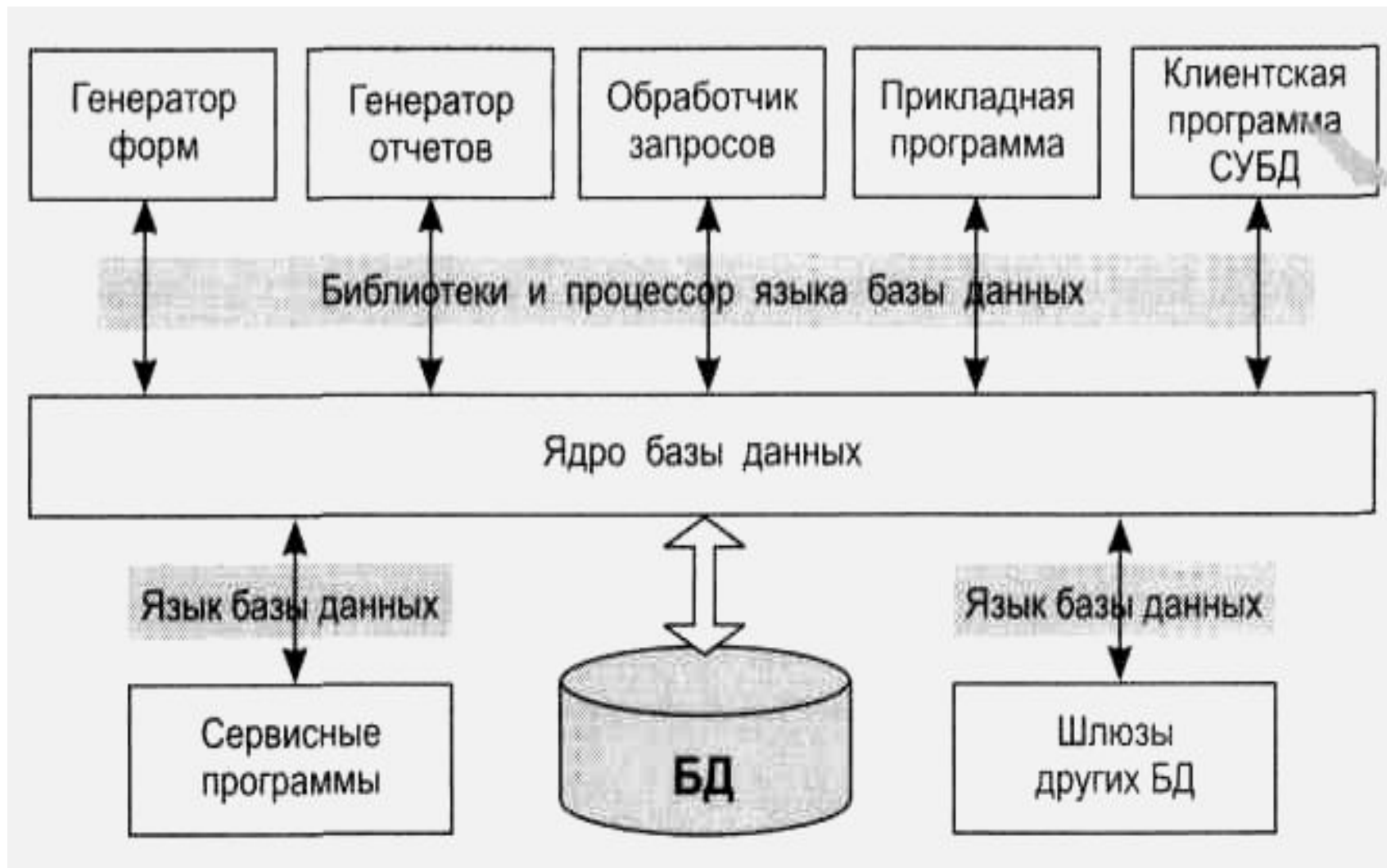


# Язык для работы с базами данных SQL (Structured Query Language)

- SQL
  - средства для спецификации и обработки запросов на выборку данных
  - функции по созданию, обновлению, управлению доступом
  - язык описания данных + язык манипулирования данными
  - внутренний язык баз данных, отражает особенности конкретной СУБД



## Программные средства



## Комплекс программных средств



# Централизованное управление данными обеспечивает:



# Технические средства

- **Технические средства БНД**

- ЭВМ

- Средства хранения данных

- Средства ввода данных

- Средства отображения данных

- Коммуникационные средства



## Организационно-административные подсистемы и нормативно-методическое обеспечение

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

# Пользователи баз данных

- Конечный пользователь
  - имеет дело только с внешним интерфейсом, поддерживаемым СУБД
  
- Администратор базы данных (АБД)
  - несет ответственность за проектирование и общее управление базой данных
  - определяет информационное содержание БД
  - идентифицирует объекты БД и моделирует базу, используя язык описания данных
  - решает все вопросы, связанные с размещением БД в памяти, выбором стратегии и ограничений доступа к данным
  - организует процессы загрузки, ведения и восстановления БД и многие другие действия, которые не могут быть полностью формализованы и автоматизированы



## Пользователи баз данных

- Прикладные программисты
  - имеют, как правило, в своем распоряжении один или несколько языков программирования, с помощью которых генерируются прикладные программы
  
- Администратор приложений
  - определяет для приложений подмодели данных
  
- (разные приложения обеспечиваются собственным «взглядом», но не на всю БД, а только на требуемую для конкретного приложения («видимую») ее часть)

# Типология баз данных

- Классификация по форме представляемой информации
  - Фактографические
  - Документальные
  - Мультимедийные
    - Картографические
    - Видео-, аудио-, графические

# Типология баз данных

- Классификация по типу хранимой (не мультимедийной) информации
  - Фактографические
    - Документальные
      - Полнотекстовые
      - Библиографически-реферативные
  - Лексикографические
    - Классификаторы
    - Кодификаторы
    - Словари
    - Тезаурусы
    - Рубрикаторы

# Типология баз данных

- Классификация по типу используемой модели данных
  - Иерархические
  - Сетевые
  - Реляционные



# Типология баз данных

- Классификация по топологии хранения
  - Локальные БД
  - Распределенные БД
- Классификация по типологии доступа и характеру использования
  - Специализированные
  - Интегрированные
- Классификация по функциональному назначению
  - Операционные
  - Справочно-информационные

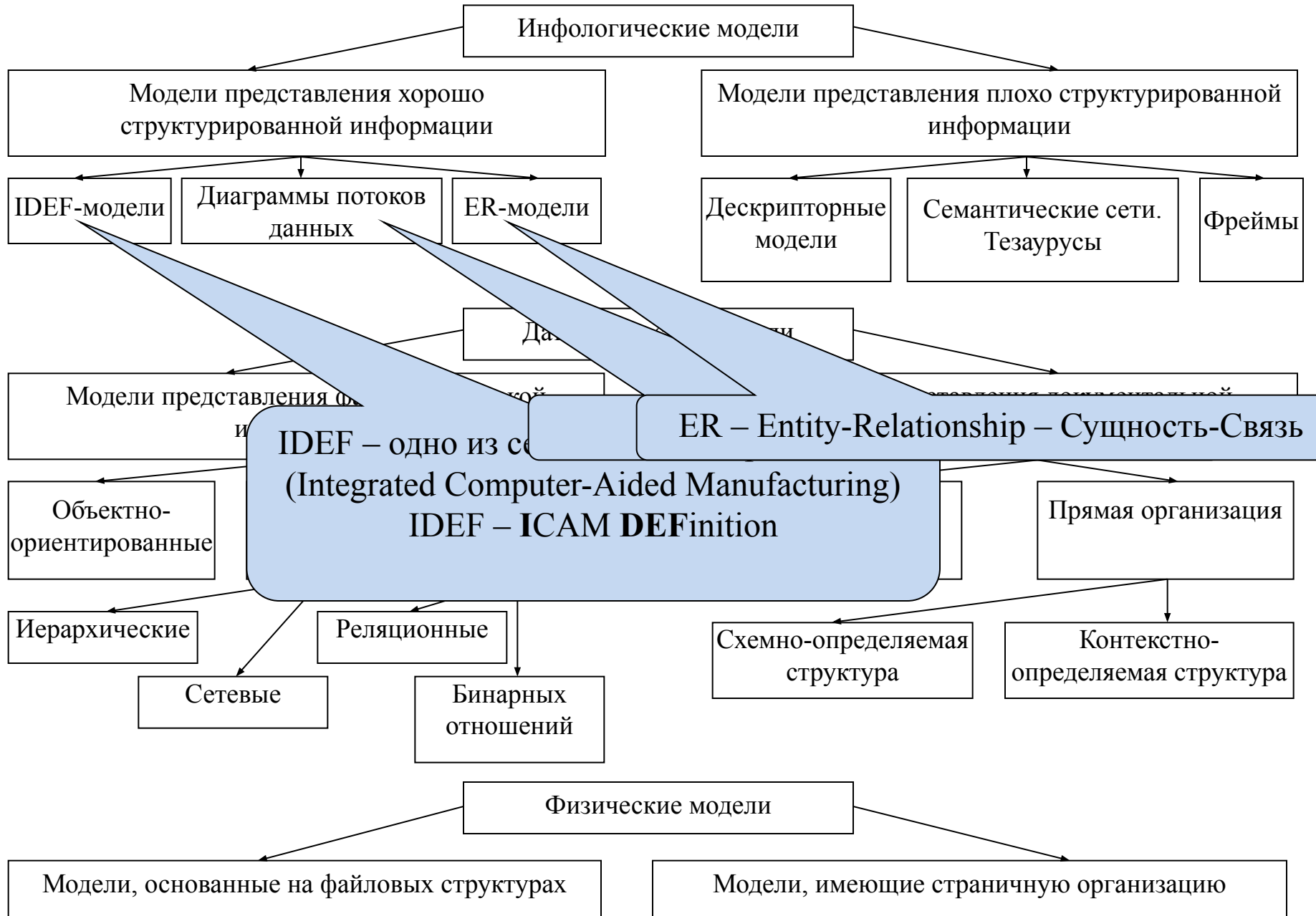
# Типология баз данных

- Классификация по сфере возможного применения
  - Универсальные
  - Специализированные
  
- Классификация по степени доступности
  - Общедоступные
  - С ограниченным доступом

# Типология баз данных с точки зрения информационных процессов

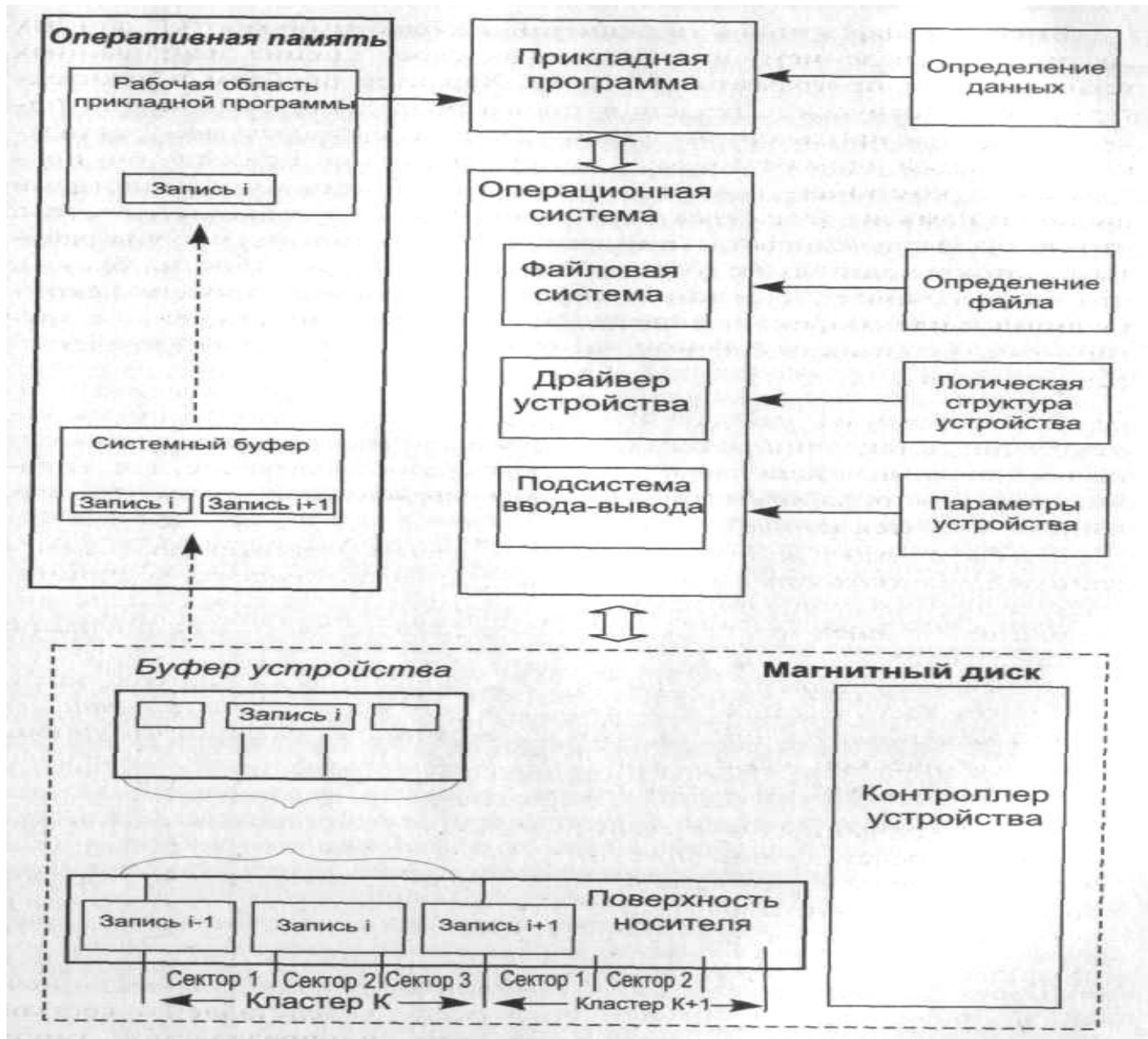
- БД могут относиться с различным уровнем информационных процессов
  - Уровень информационных технологий (ИТ)
    - БД определяется как взаимосвязанная совокупность файлов ОС, содержащих данные о предметной области решаемой задачи
    - основное внимание уделяется *физической структуре БД*
  - Уровень системы (ИС)
    - БД рассматривается как компонента, представляющая собой информационную модель предметной области
    - наиболее важной является проблема *логической структуры БД*
  - Уровень информационных ресурсов (ИР)
    - БД трактуется как элемент мировых ИР
    - основной характеристикой здесь является *содержание БД*

# Система моделей представления информации





# Примерная схема организации ввода-вывода



# Линейные файлы данных

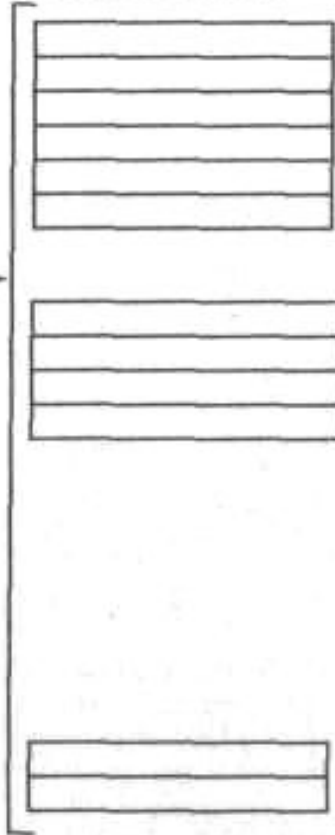


# Методы доступа к записям

Логический файл

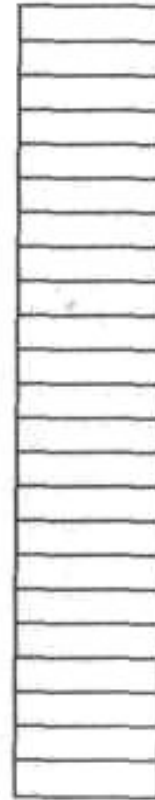


Последовательный набор данных

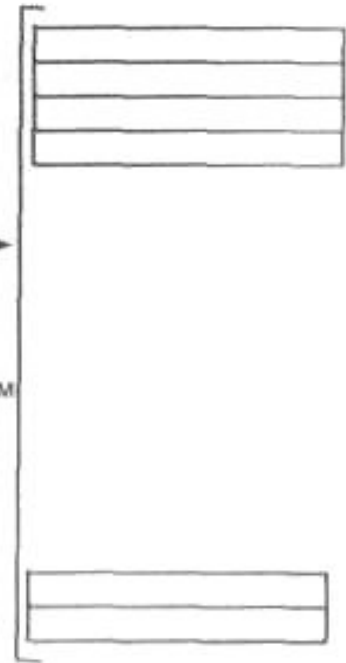


Метод доступа

Логический файл



Наборы данных прямого доступа

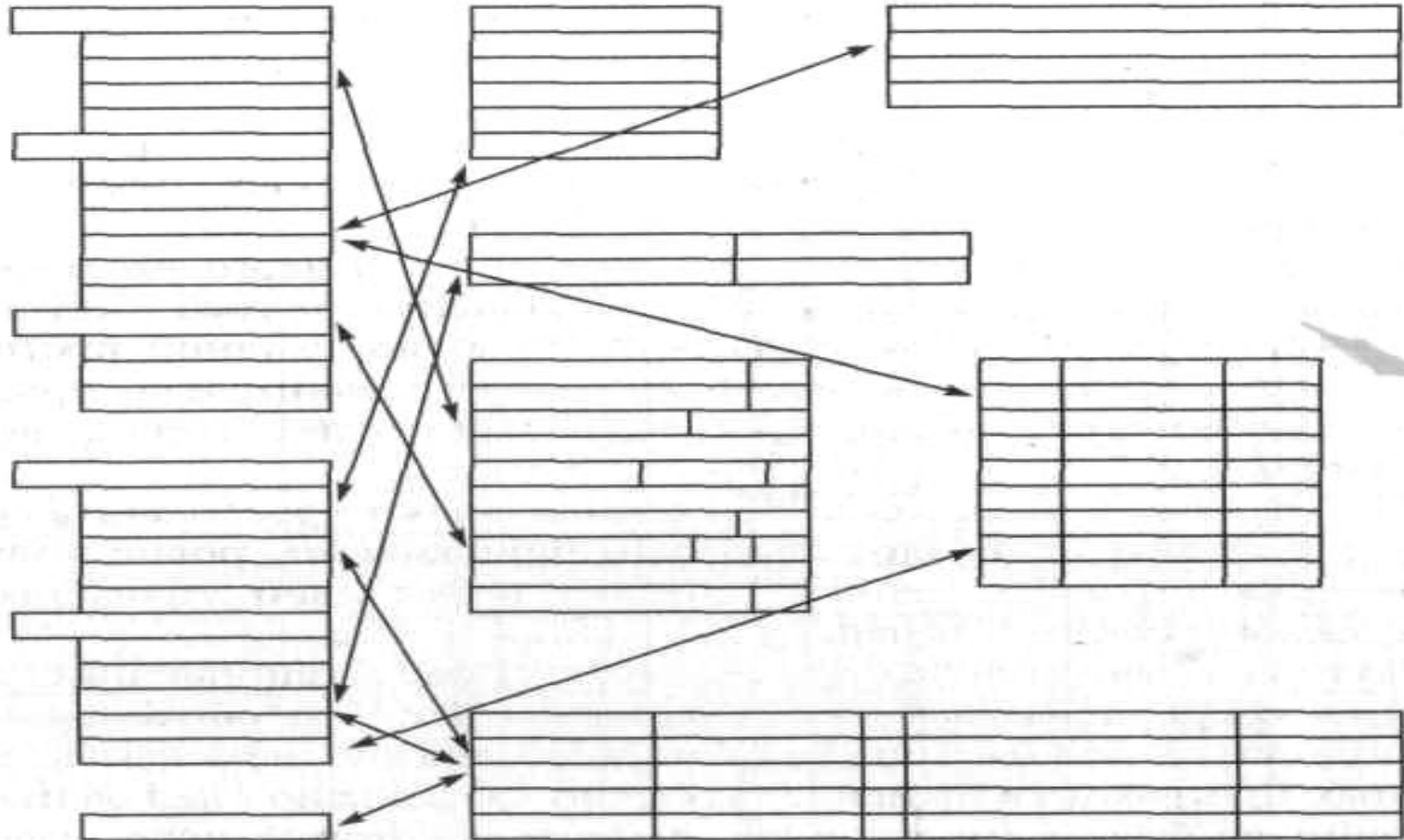


Метод доступа  
(с механизмом адресации)

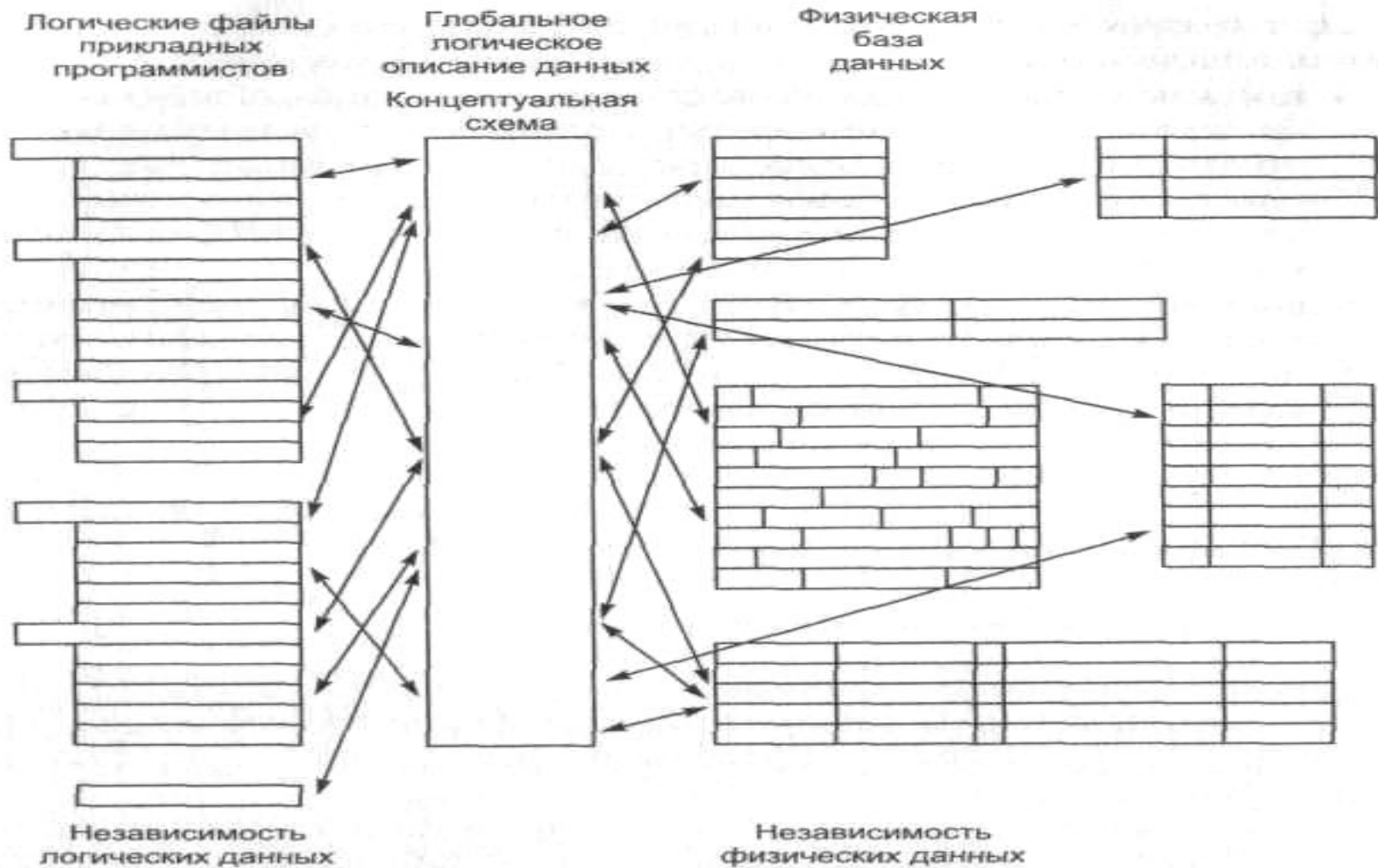
# Первые системы управления базами данных

Логические файлы  
прикладных программ

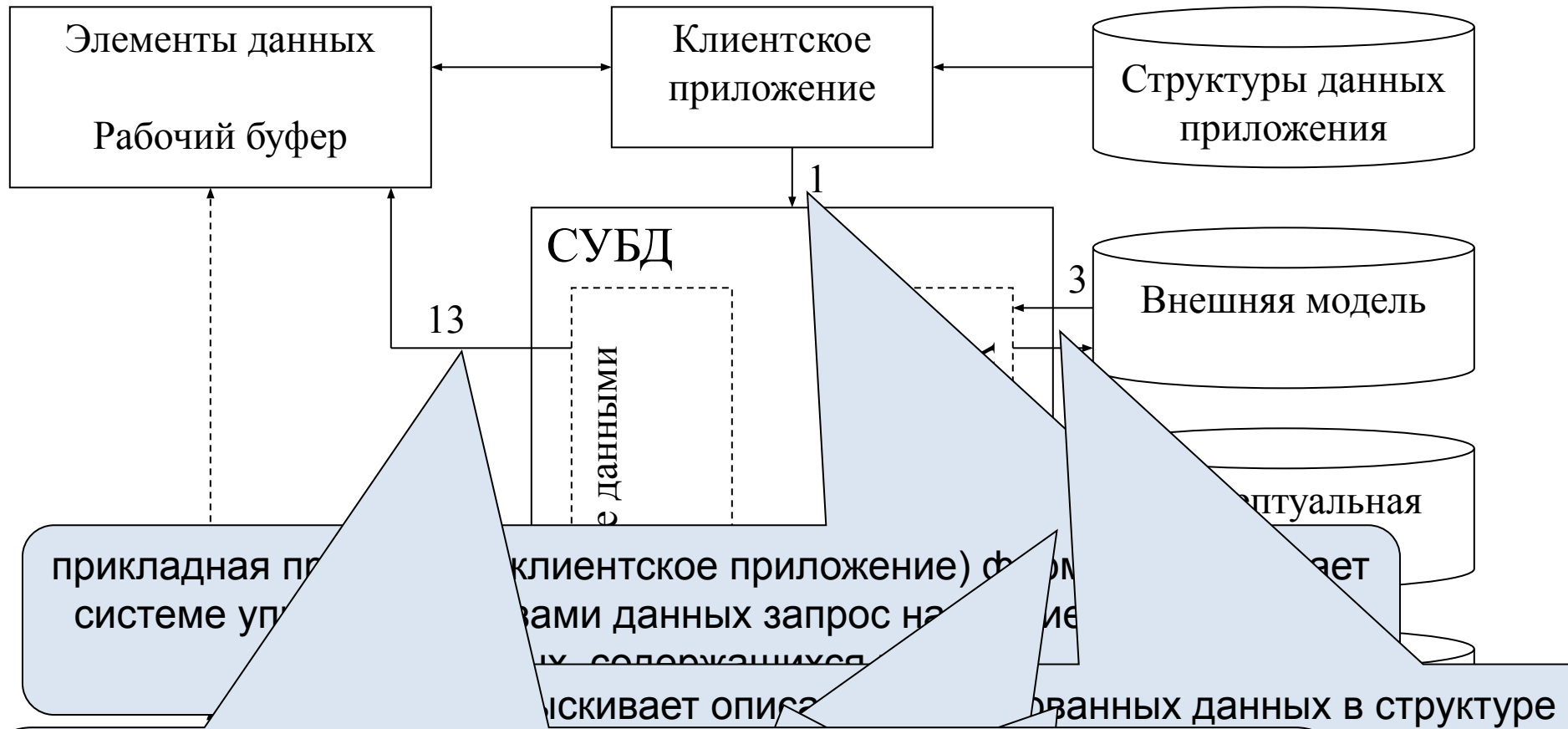
Физическая база данных



# Два уровня независимости данных



# Схема обработки запроса на выборку данных из БД

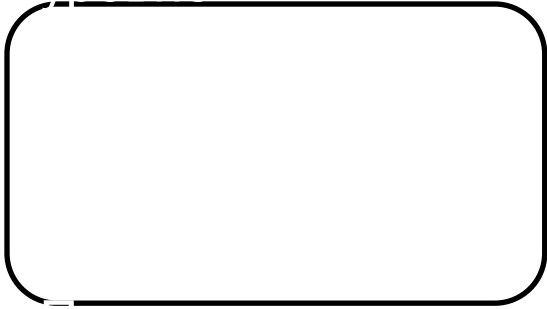


прикладная программа  
системе управления

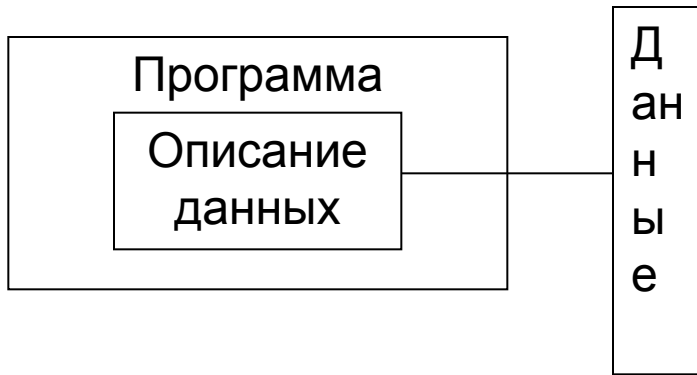
клиентское приложение) формирует  
запрос на выборку данных содержащих  
описания данных в структуре  
модель)

СУБД выделяет необходимую логическую запись, осуществляет форматные преобразования, обусловленные различиями описаний на глобальном и прикладном уровнях, и передает для функциональной обработки приложением данные в рабочий буфер, выделяемый прикладной программой или самой СУБД

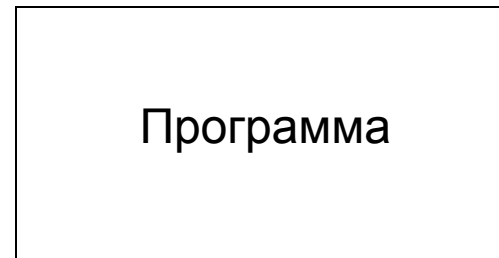
# форматы, структуры данных



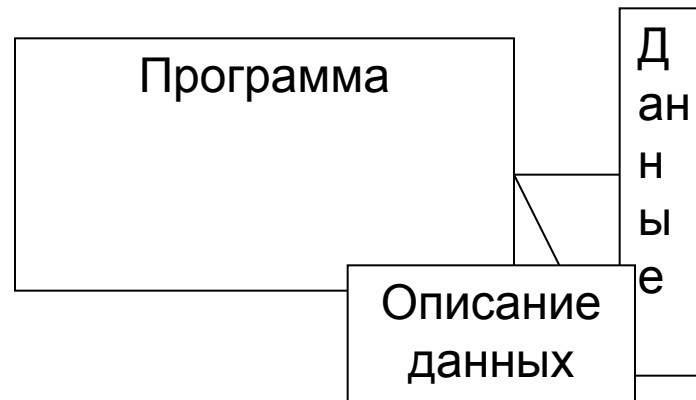
# Варианты размещения данных и их описания



В прикладной программе



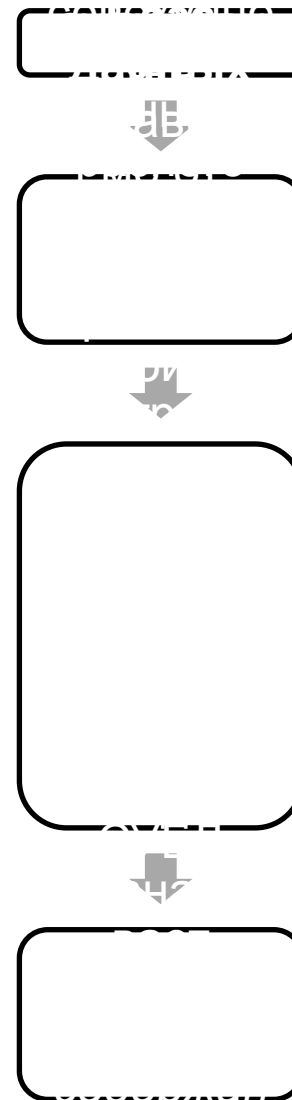
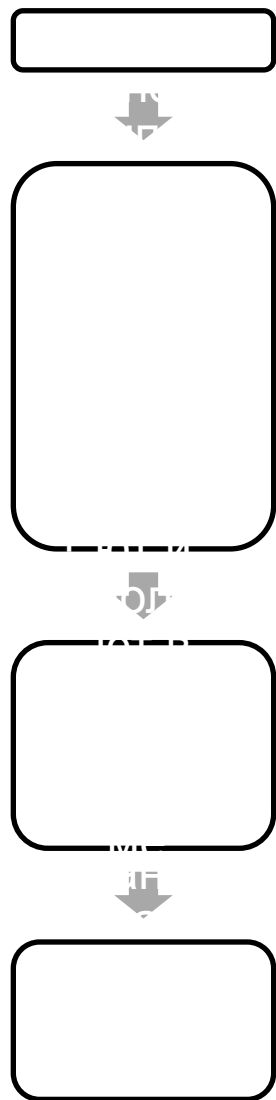
В файле данных



Отдельным набором данных (словарь данных)



# Основные отличительные особенности обработки данных, характерные для файловых систем и систем управления базами данных



# Основные задачи обработки данных, решаемые на основе концепций баз данных, сводятся к следующим вопросам

1. Как организовать хранение данных?

1. Как организовать хранение данных?

2. Как обеспечить целостность данных?

2. Как обеспечить целостность данных?

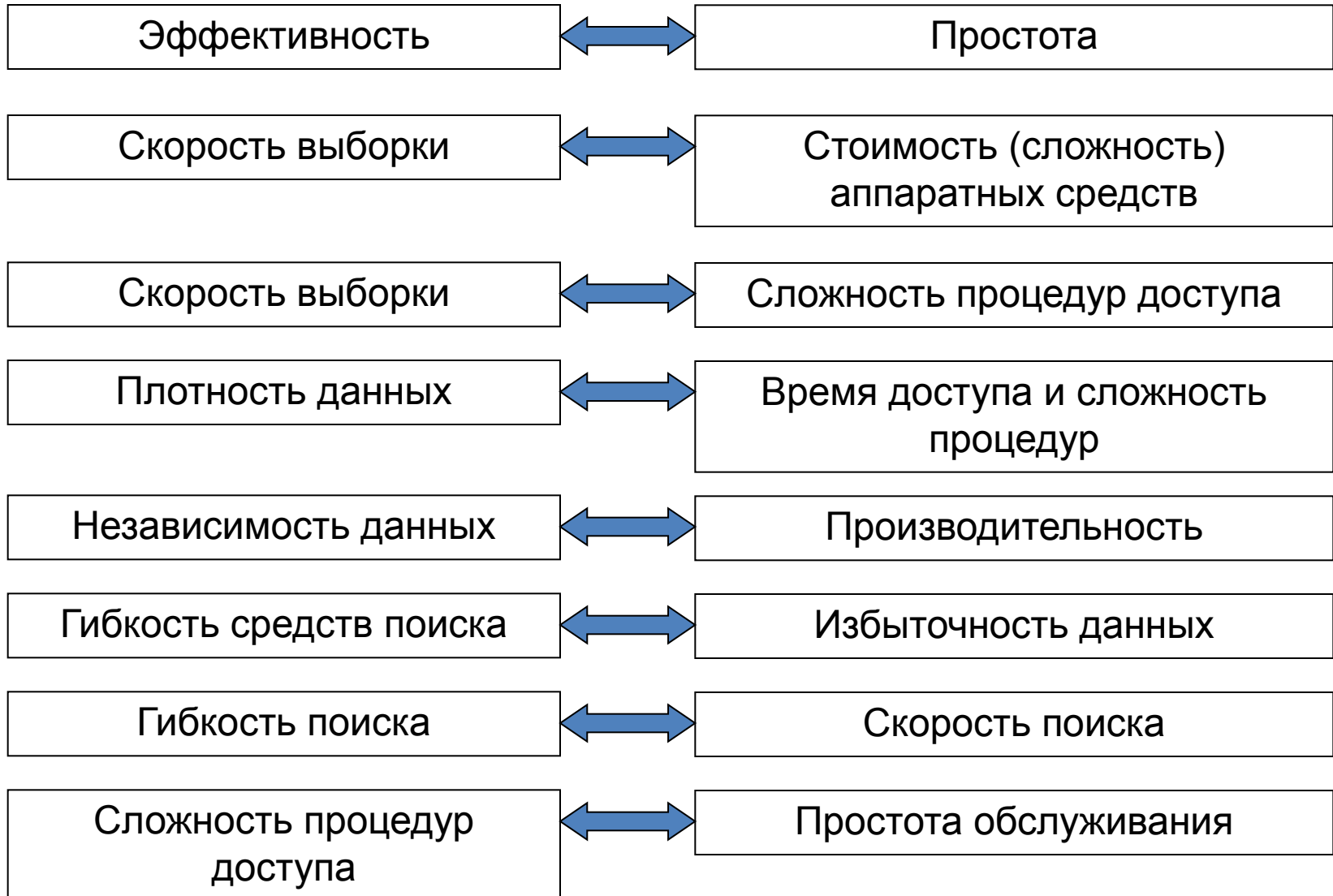
3. Как обеспечить безопасность данных?

3. Как обеспечить безопасность данных?

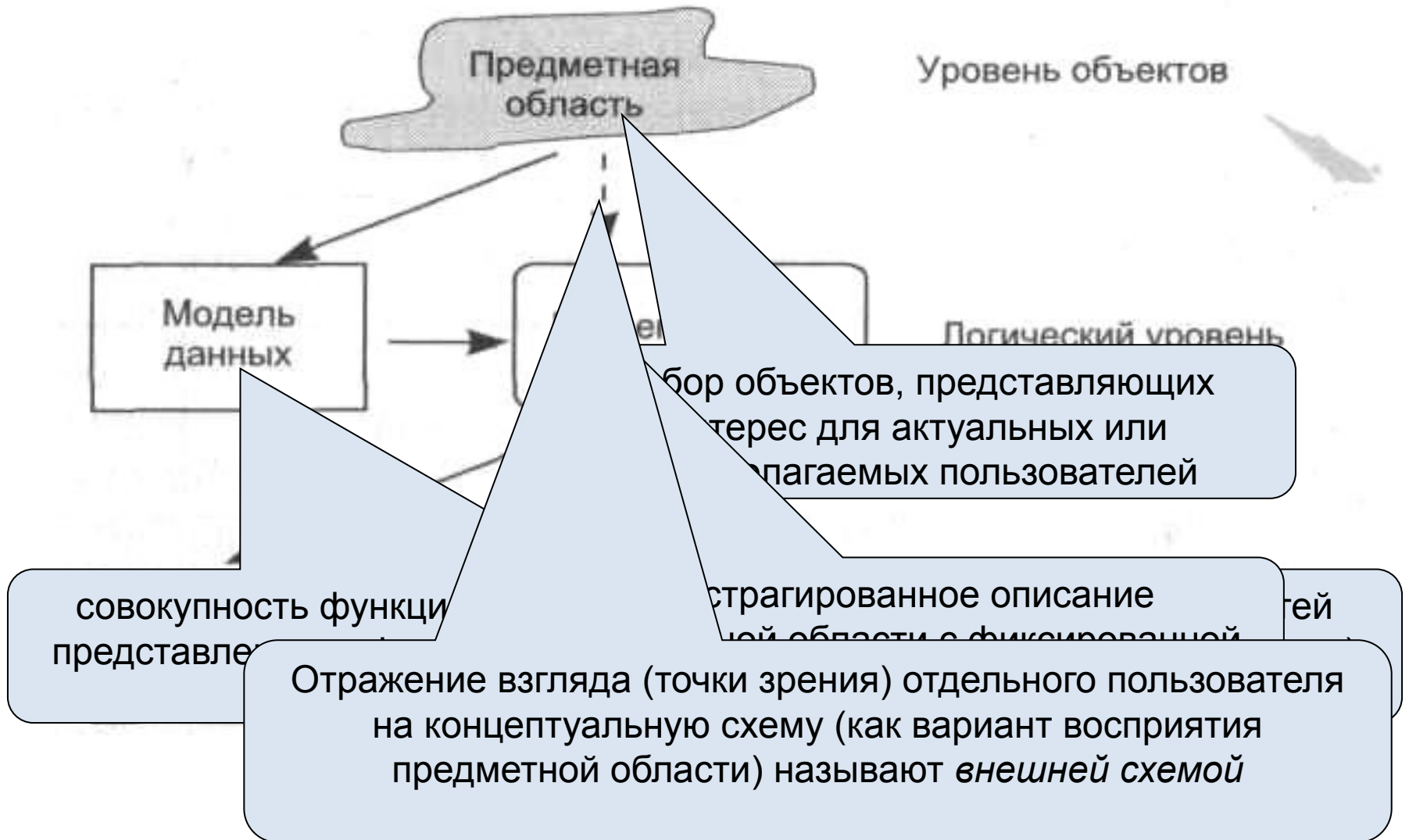
4. Как обеспечить доступность данных?

4. Как обеспечить доступность данных?

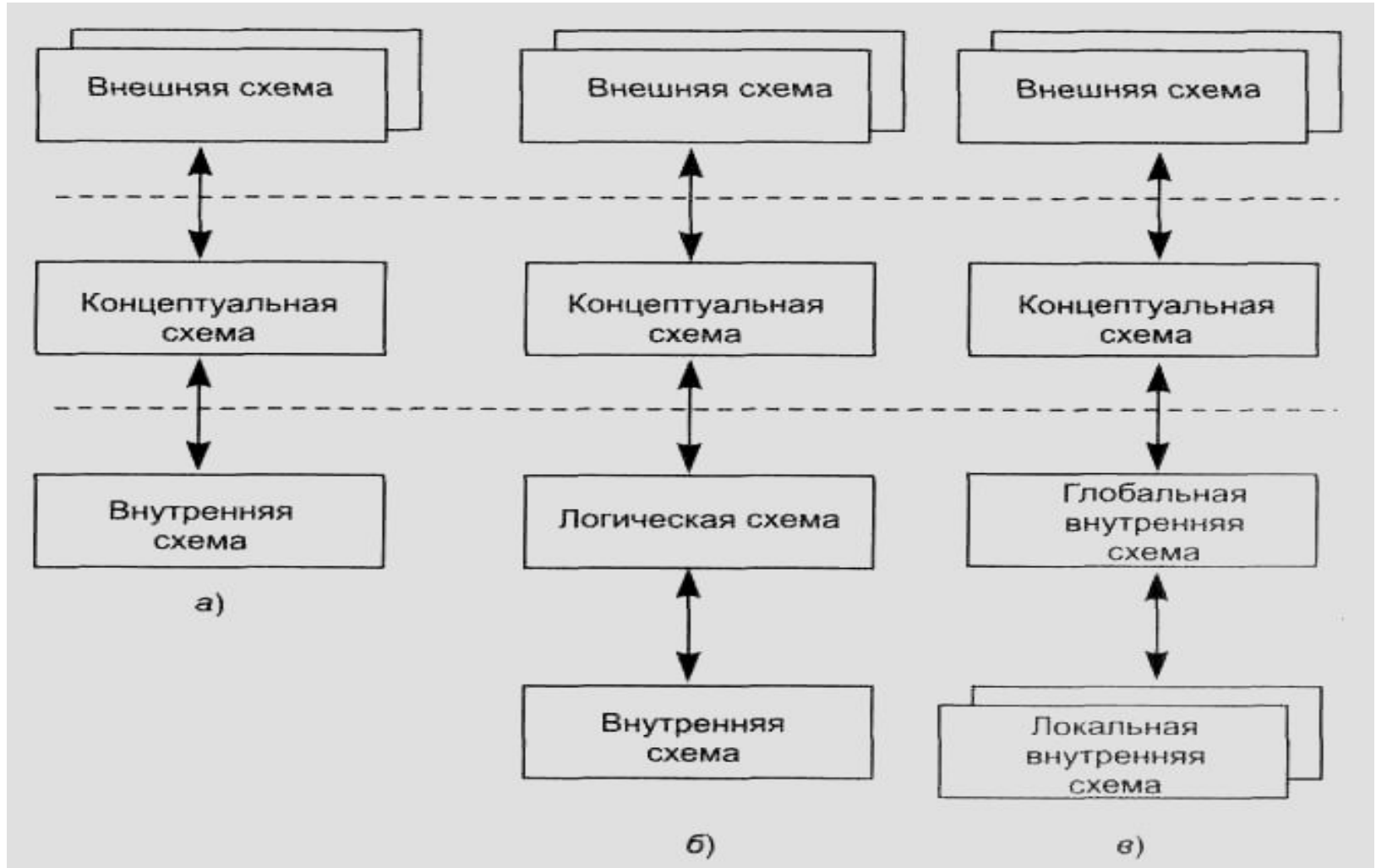
**Создание базы данных — это попытка найти компромисс сразу по нескольким направлениям и сочетаниям нескольких взаимообратных факторов (с точки зрения прагматики)**



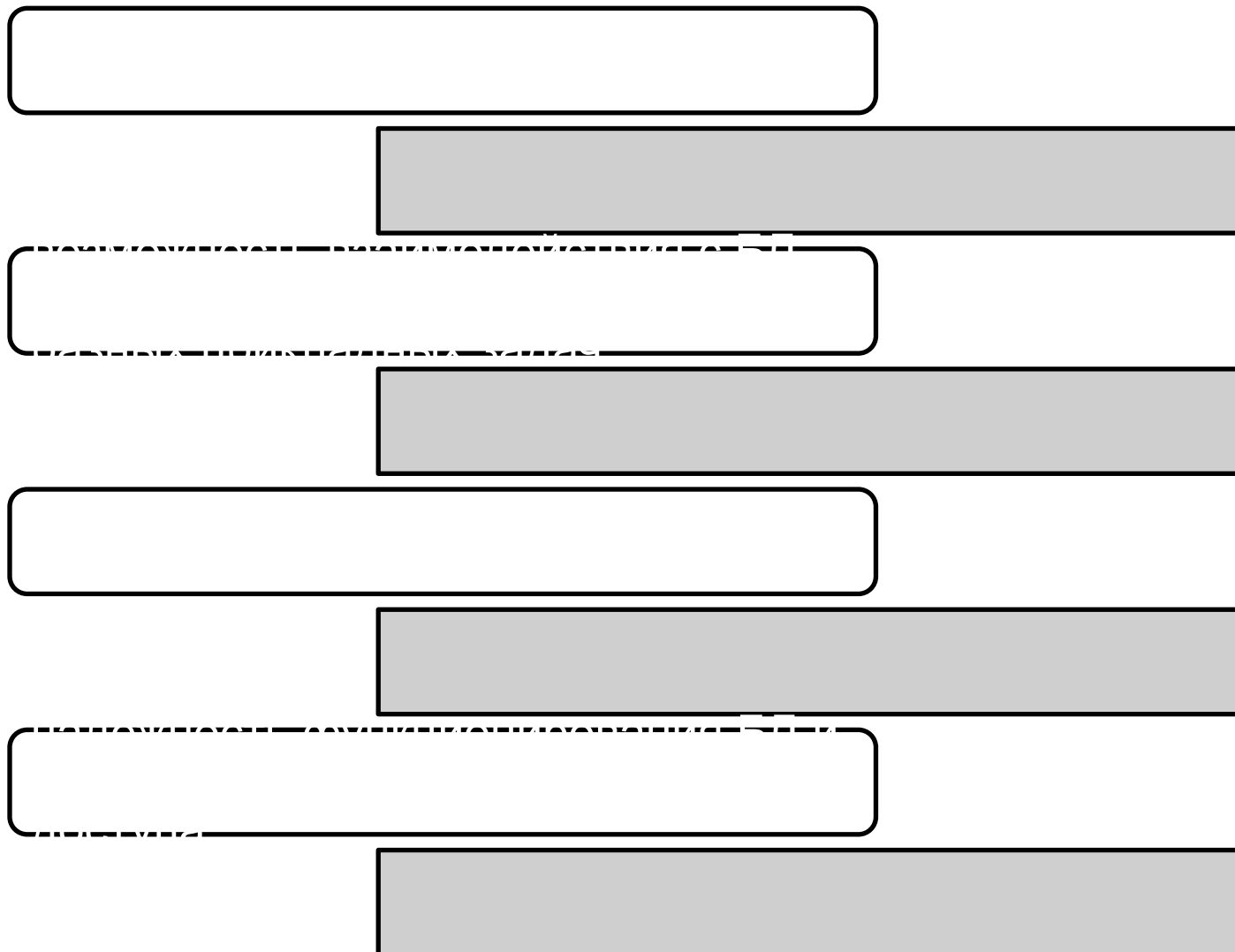
# Соотношение понятий концептуальной и внутренней схем



# Варианты решений трехуровневого представления



# Трехуровневая архитектура обеспечивает выполнение основных требований, предъявляемых к системам баз данных



**Трехуровневая архитектура имеет следующие достоинства с точки зрения пользователей различных категорий**

An empty rounded rectangular box with a black border, intended for text input.A solid gray rectangular box with a black border, likely representing a placeholder for a specific benefit.An empty rounded rectangular box with a black border, intended for text input.A solid gray rectangular box with a black border, likely representing a specific benefit.An empty rounded rectangular box with a black border, intended for text input.A solid gray rectangular box with a black border, likely representing a specific benefit.

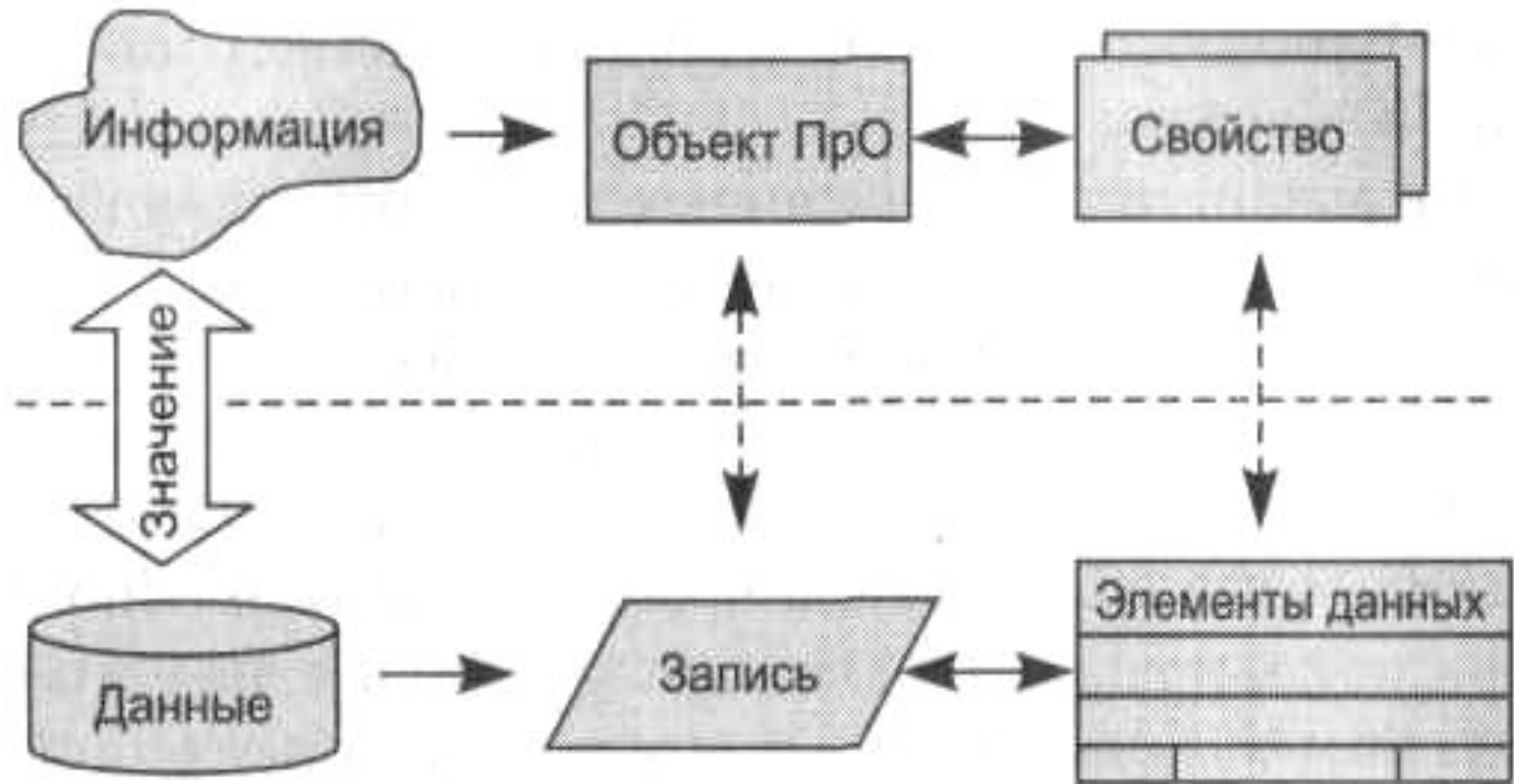
# Идентификация объектов и записей

## Основные понятия

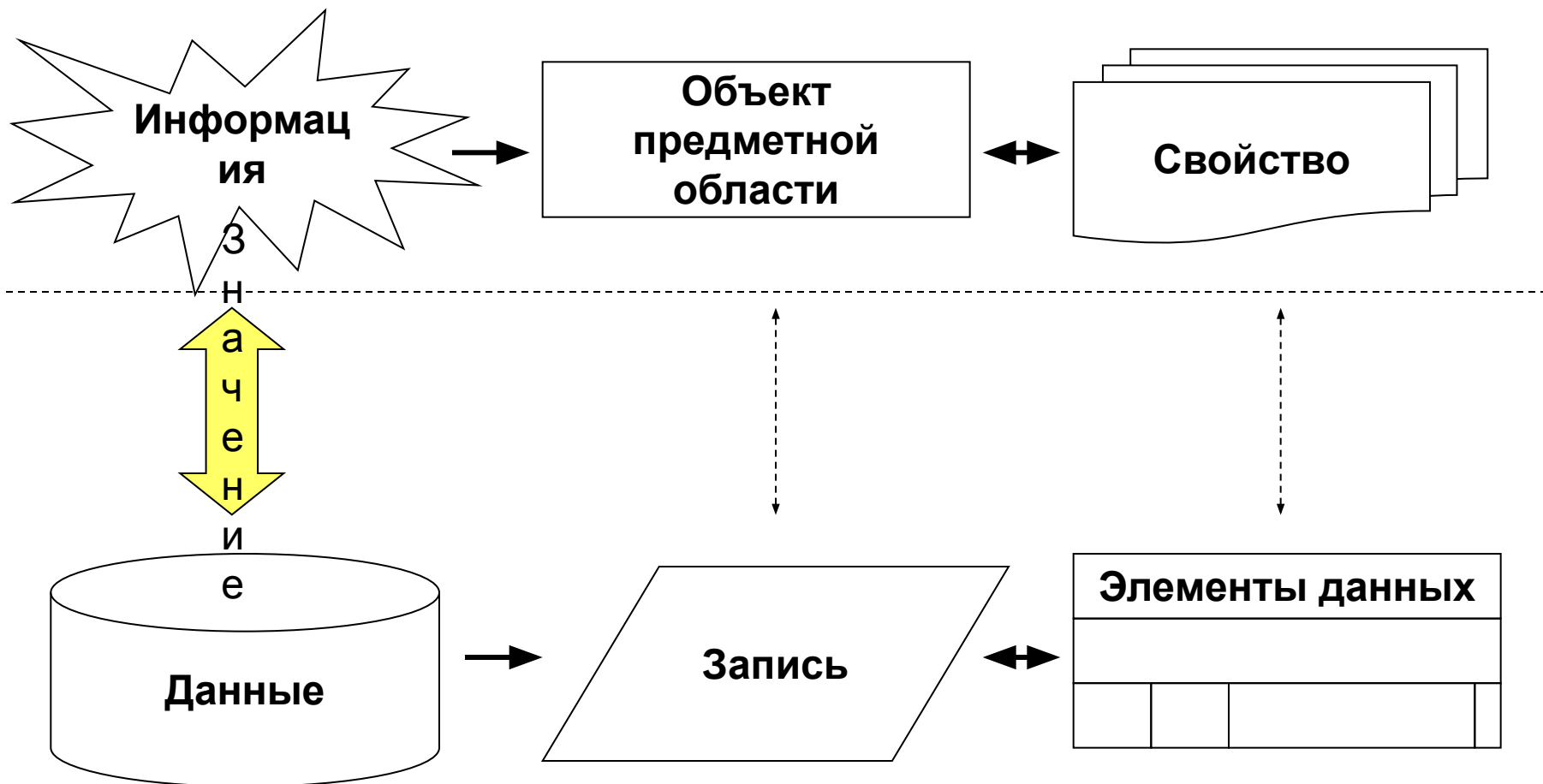
- в задачах обработки информации атрибуты *именуют* (обозначают) и приписывают им *значения*
- *информацию о свойствах* каждого из объектов рассматриваемой совокупности надо сохранять (записывать) как *данные*, чтобы при решении задач их можно было найти и выполнить необходимые преобразования
- любое состояние объекта характеризуется совокупностью актуализированных атрибутов (имеющих некоторое из значений в этот момент времени), которые фиксируются на некотором материальном носителе в виде *записи* — совокупности (группы) формализованных *элементов данных* (значений атрибутов, представленных в том или ином формате)



## Атрибутивный способ идентификации (используется для хорошо структурированной информации)



# Атрибутивный способ идентификации (используется для хорошо структурированной информации)



## Поиск записей

- В зависимости от уровня программного обеспечения прикладной программист может использовать следующие способы
  - Задать машинный адрес данных и в соответствии с физическим форматом записи прочитать значение
  - Это случай, когда программист должен быть «навигатором»
- Сообщить системе имя записи или элемента данных, которые он хочет получить, и возможно, организацию набора данных.
- В этом случае система сама произведет выборку, но для этого она должна будет использовать вспомогательную информацию о структуре данных и организации набора

**В качестве ключа, обеспечивающего доступ к записи, можно использовать идентификатор — отдельный элемент данных**

# Поиск записей

- **Первичный (главный) ключ**

- идентифицирует запись единственным образом

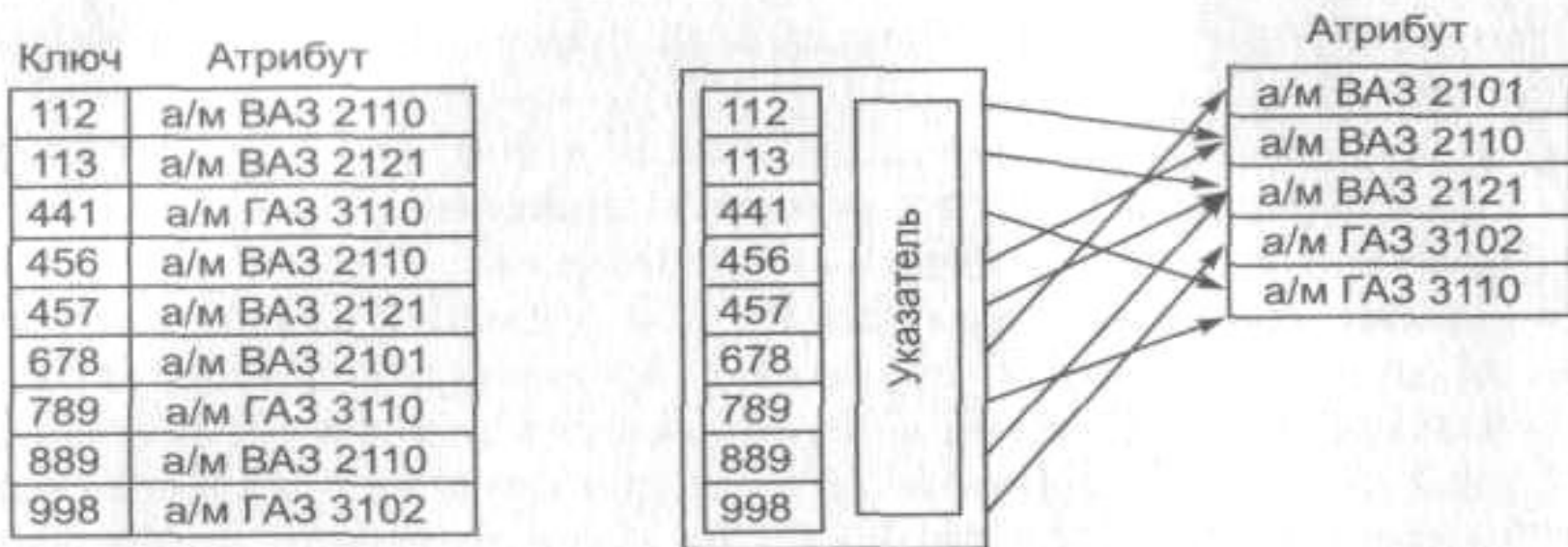
- **Вторичный (альтернативный) ключ**

- идентифицирует некоторую группу записей, имеющих определенное общее свойство

- **Сцепленный ключ**

- несколько элементов данных, которые в совокупности, обеспечат уникальность идентификации каждой записи набора данных

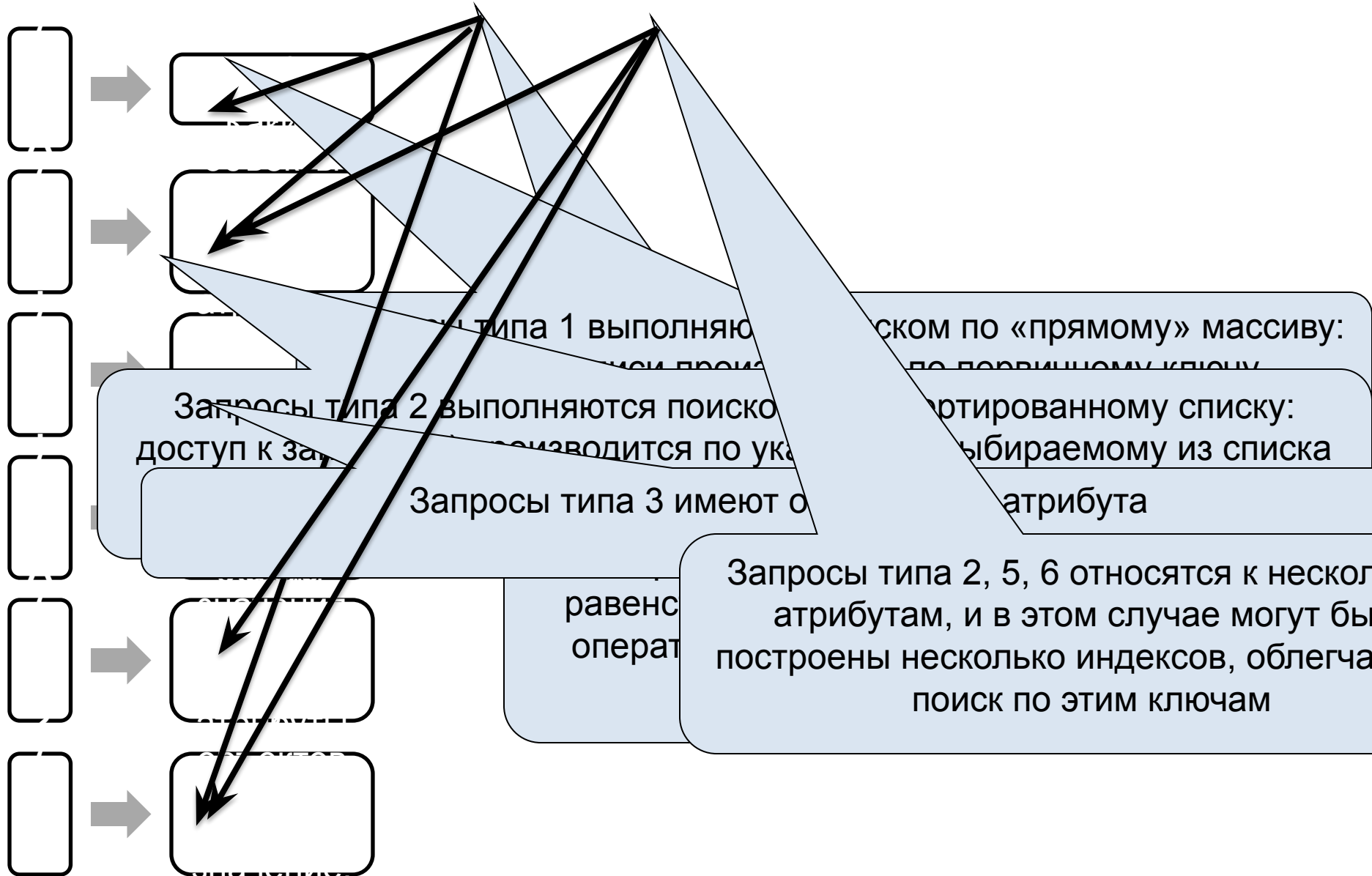
## Способы хранения ключа и атрибута



## Инвертируемый список

а/м ВАЗ 2110	678
а/м ВАЗ 2110	112, 456, 889
а/м ВАЗ 2121	113, 457
а/м ГАЗ 3102	998
а/м ГАЗ 3110	441, 789

# Типология простых (атомарных) запросов:

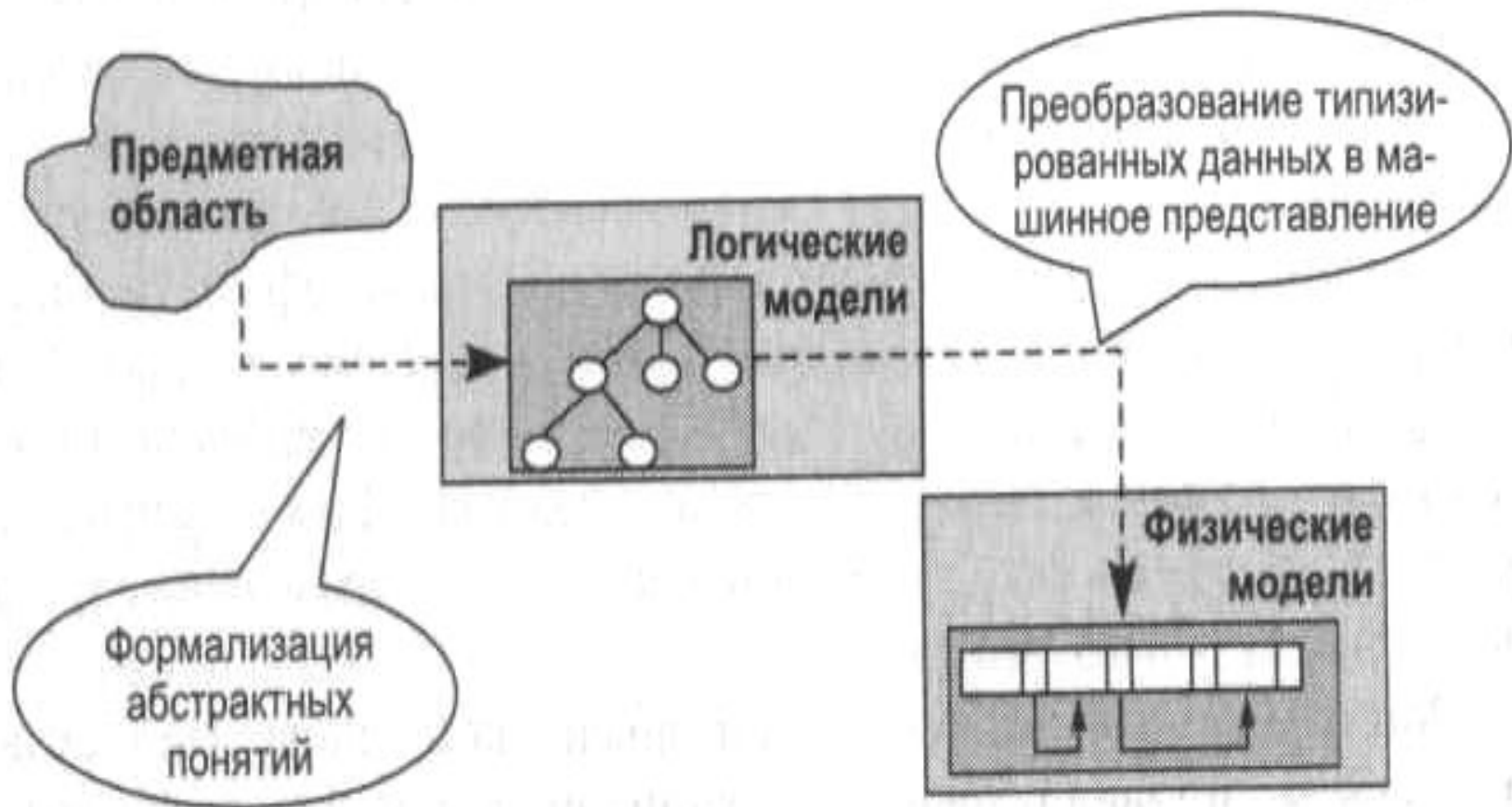


**Составные условия поиска могут использовать несколько простых логическими (булевыми) операторами**

**используются**  
**логических файлов.** В этих  
системах могут располагаться в

**используются**  
**логических файлов.** В этих  
отрицено наличие

# Этапы преобразования представлений предметной области





# Модели данных и баз данных

- Модель данных - обеспечивает на формальном абстрактном уровне конкретные способы представления объектов и связей
  - средство спецификации типов данных и их организации, разрешенных в конкретной БД
  - основа разработки общей методологии построения баз данных
  - основа минимизации влияния эволюции баз данных на уже существующие прикладные программы и работу конечных пользователей
  - основа разработки семейства языков запросов и языков манипулирования данными
  - основа архитектуры СУБД
  - основа изучения динамических свойств различных организаций данных
- Модель базы данных
  - определяет ясную границу между логическим и физическим аспектами управления базой данных (*независимость данных*)
  - обеспечивает конечным пользователям и программистам, создающим БД, возможность и средства общего понимания смысла данных (*коммуникабельность*)
  - определяет языковые понятия высокого уровня, обеспечивающие возможность выполнения однотипных операций над большими совокупностями записей (в общем случае разнотипных данных) как единую операцию (*обработка множеств*)

# Структуры данных

## • Структура информации

- схематичная форма представления сложных композиционных объектов и связей реальной предметной области, выделяемых как актуально необходимые для решения прикладных задач
- это неотъемлемое свойство информации (сведений, сигналов, воспринимаемых субъектом) о некоторой совокупности объектов предметной области в контексте практической задачи (решаемой субъектом)

## • Структура данных

- атрибутивная форма представления свойств и связей предметной области, ориентированная на выражение описания данных средствами формальных языков
- это определение информационных массивов (состава и взаимосвязей данных на логическом уровне, соответствующих характеру информации и видам соответствующих преобразований)

## • Структура записей

- целесообразная (учитывающая особенности физической среды) реализация способов хранения данных и организации доступа к ним как на уровне отдельных записей, так и их элементов
- это определение структуры физической памяти: выделение, освобождение и защита областей физического носителя, способы адресации и пересылки

# Линейные структуры

- **Массивы**

- совокупность однотипных элементов
- число элементов массива известно до его размещения, что позволяет строить гибкие многомерные системы адресации

- **Последовательности**

- представляет собой совокупность однотипных элементов
- число элементов до размещения неизвестно

- **Таблицы**

- последовательности, обычно представляемые строками – совокупностями разнотипных элементов.
- множество записей, каждая из которых представляет набор поименованных полей

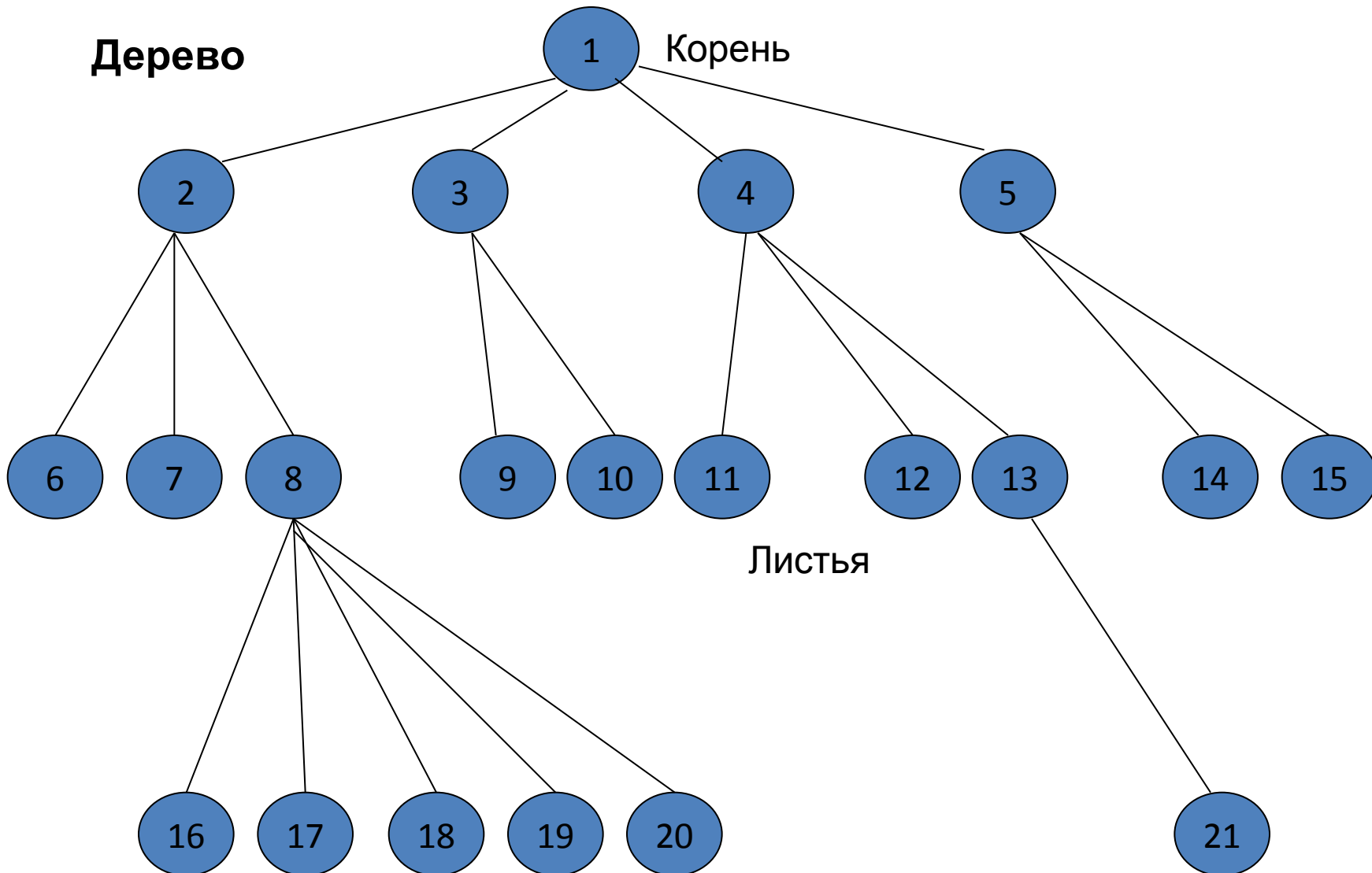
# Нелинейные структуры

- совокупность однотипных элементов
- порядок выборки элементов может отличаться от порядка следования в памяти, определенного при размещении
- число элементов до размещения неизвестно

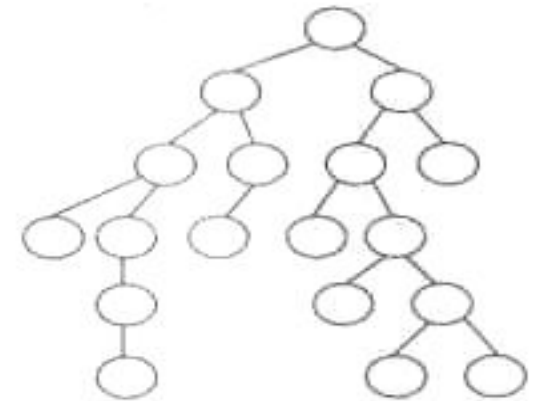
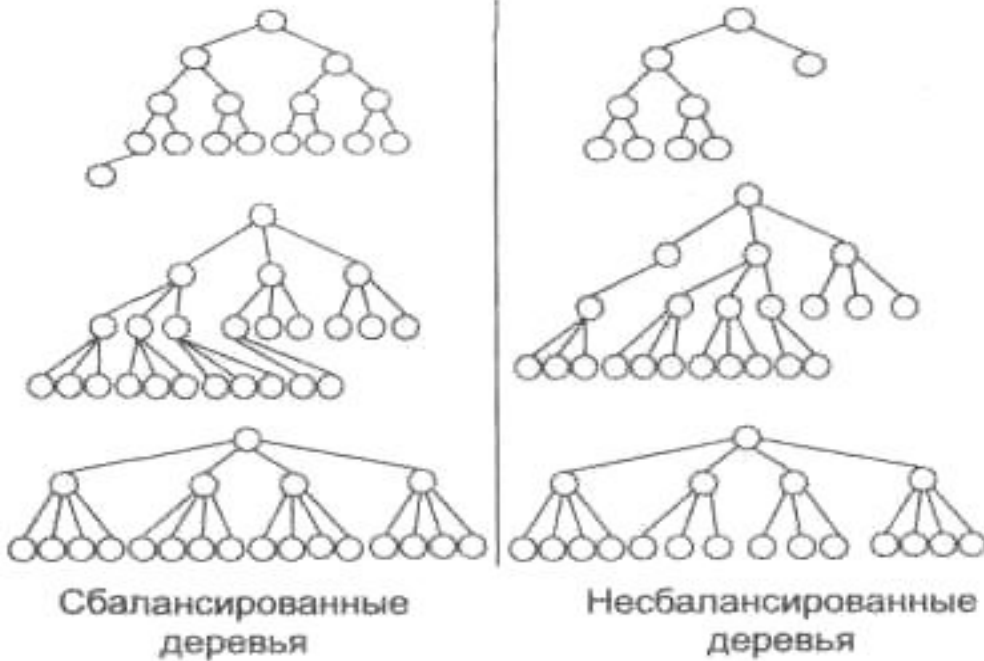
- представляет собой иерархию элементов, называемых *узлами*
- на самом верхнем уровне иерархии имеется только один узел — корень
- каждый узел, кроме корня, связан с одним узлом на более высоком уровне, называемым *исходным узлом* для данного узла
- Каждый элемент может быть связан с одним или несколькими

- любой элемент может быть связан с любым другим элементом

**Дерево**



# Деревья

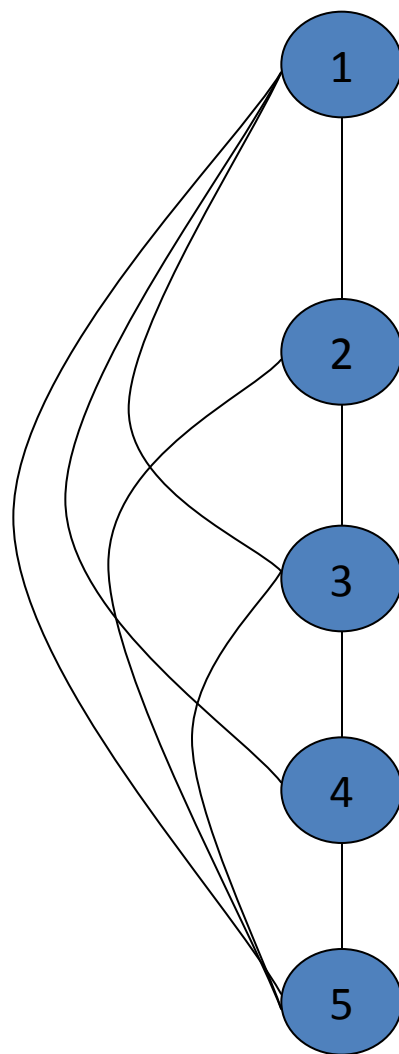
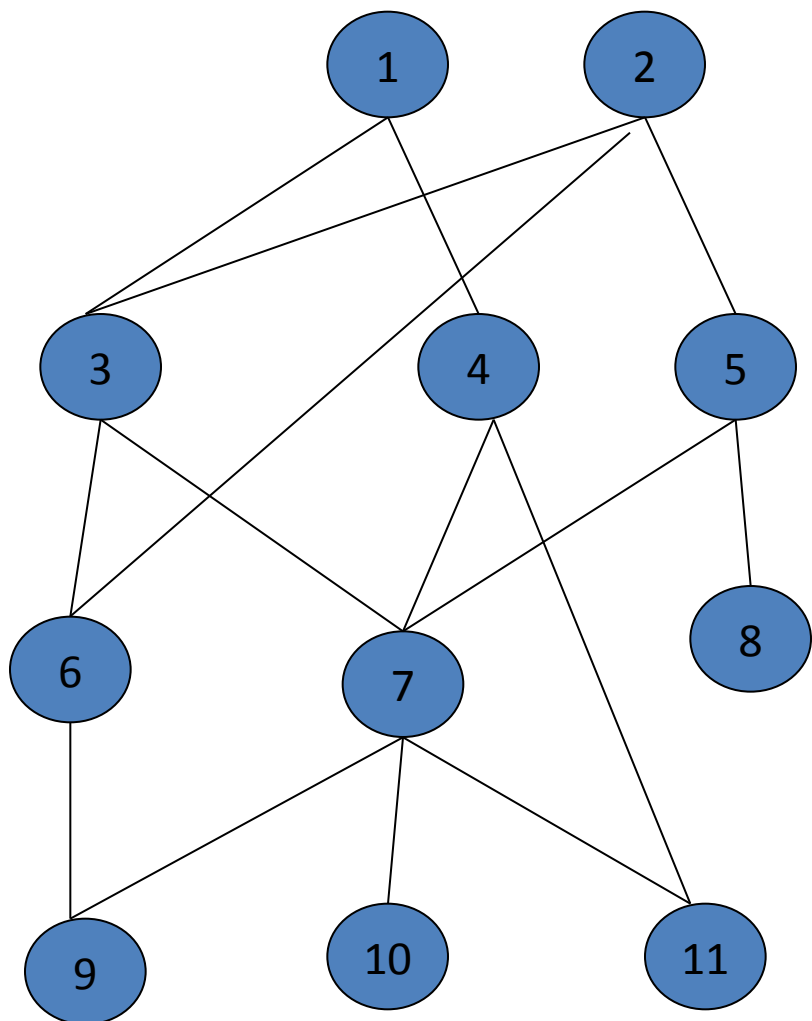


Пример несбалансированного двоичного дерева

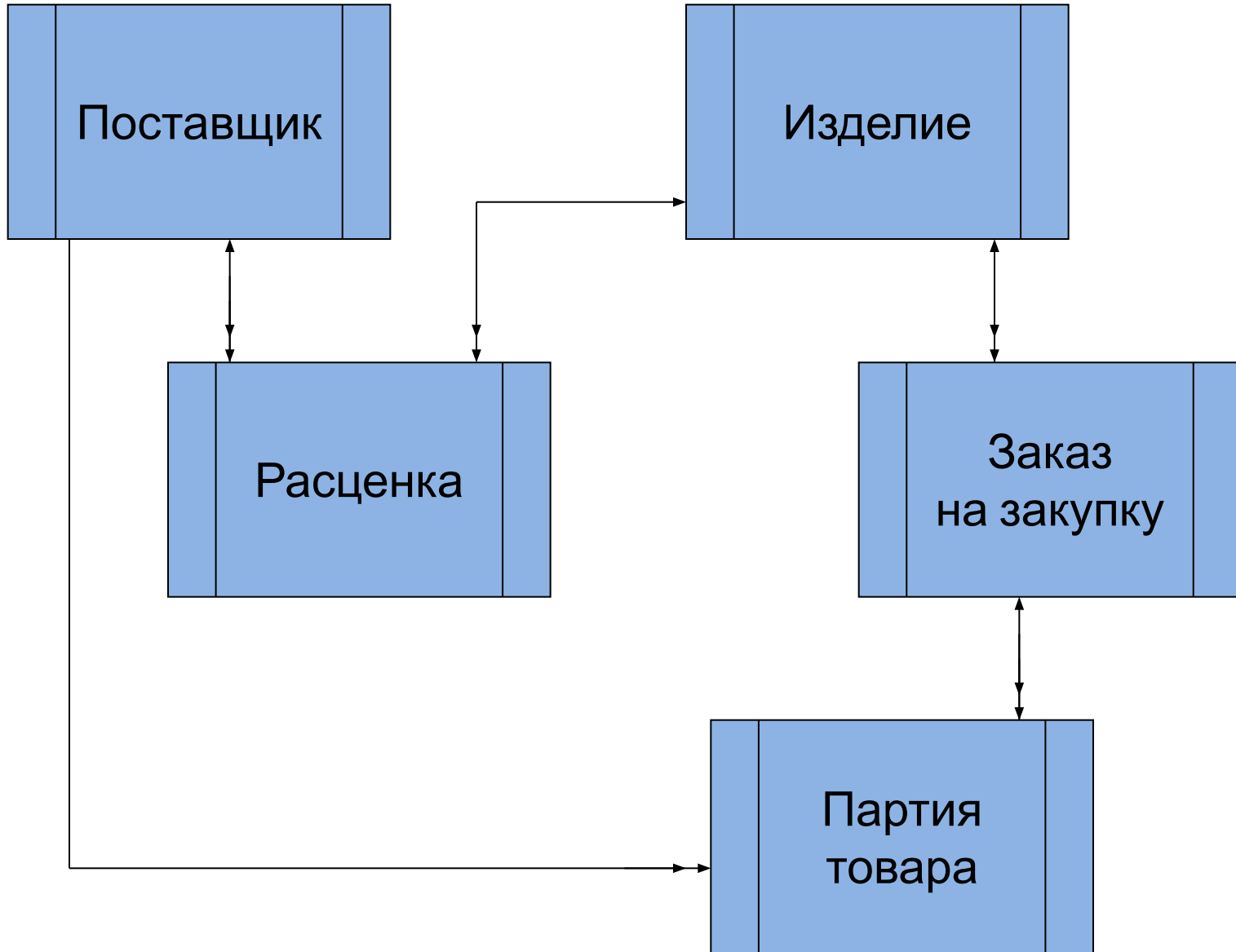
*Сбалансированное дерево* в каждом узле имеет одинаковое число ветвей, причем процесс включения новых ветвей в узлы дерева идет сверху вниз, а на каждом уровне дерева — слева направо

*Двоичные деревья* — это особая категория сбалансированных древовидных структур, в которой допускается не более двух ветвей для одного узла

# Сетевые структуры

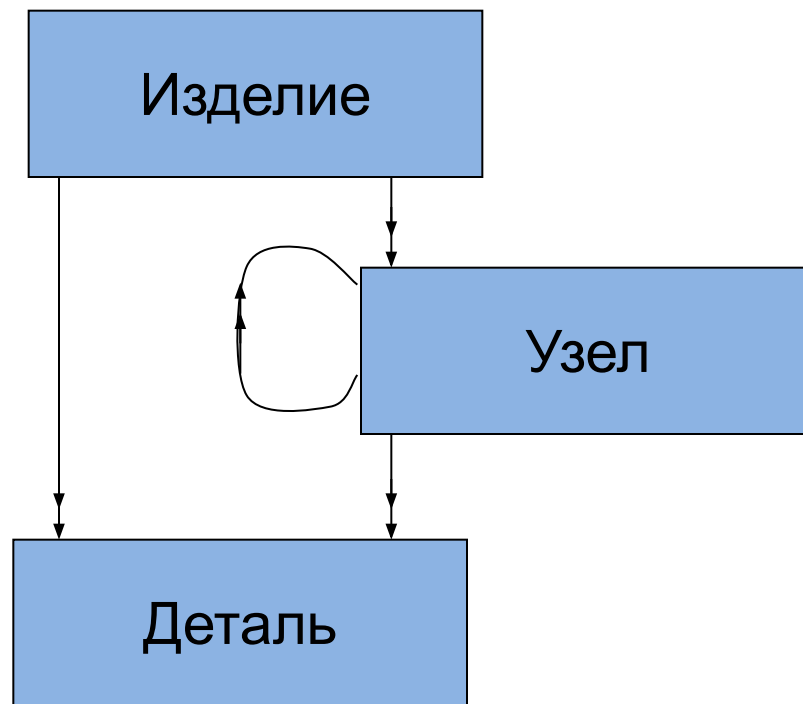
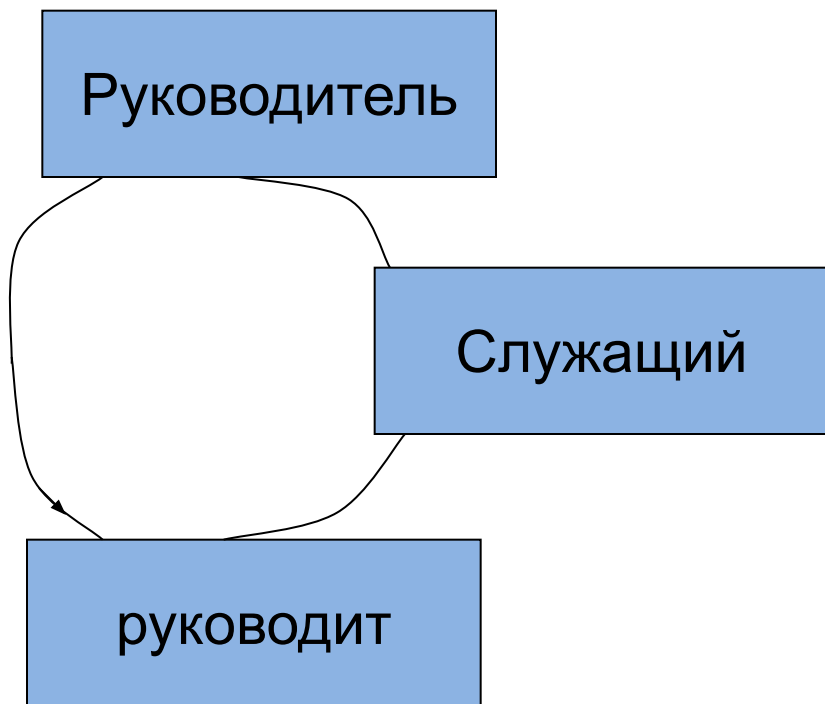


# Пример простой сетевой структуры

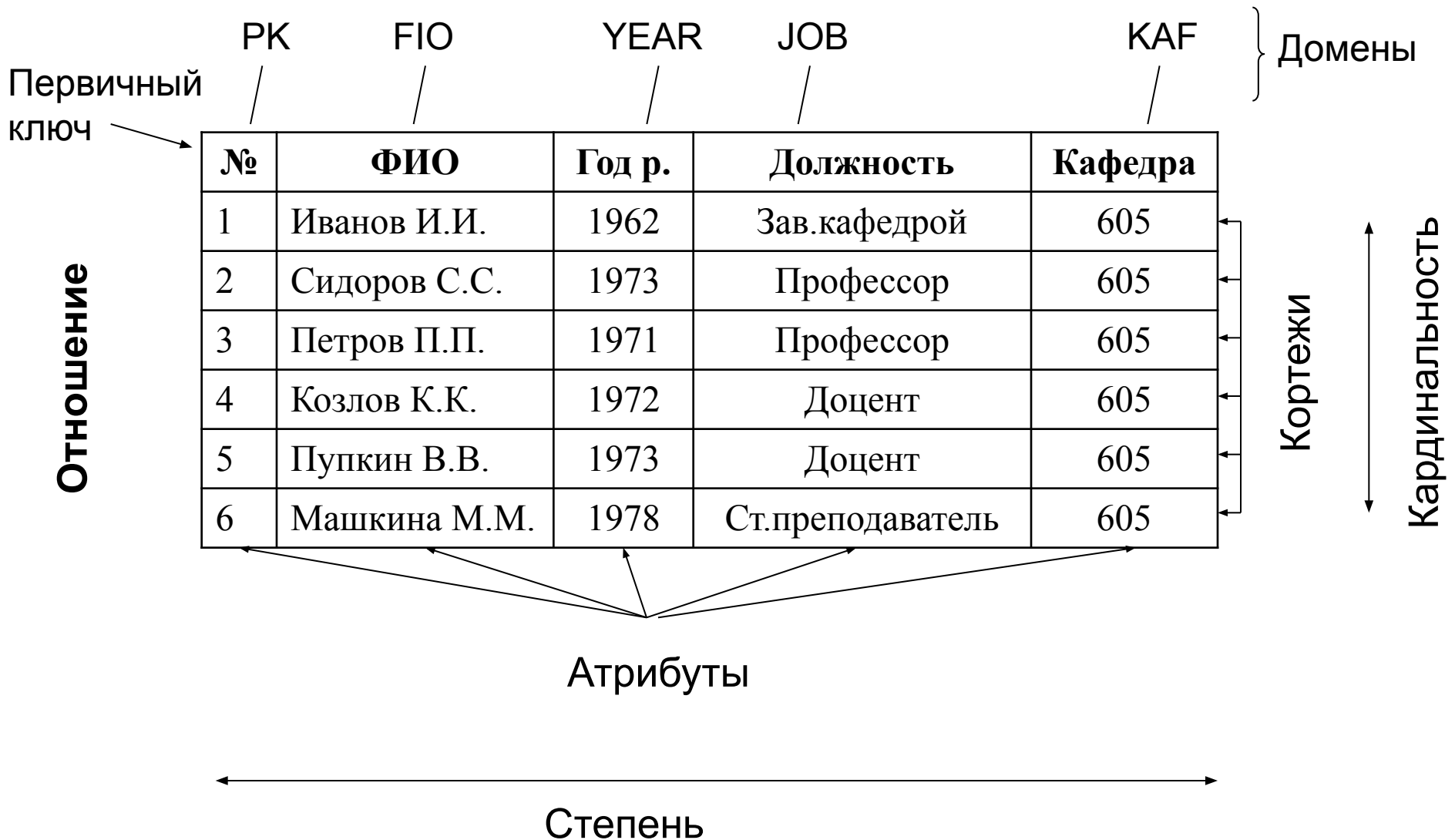




## Пример сетевой структуры с петлей



# Основные понятия реляционной модели



Реляционная модель предъявляет к таблицам следующие **требования**:

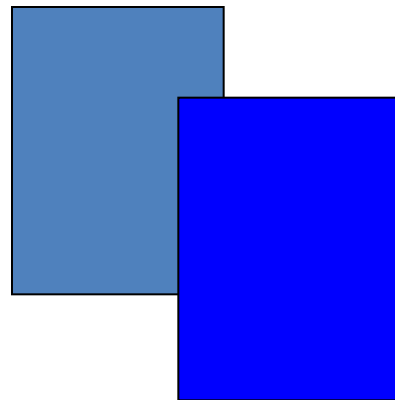
- 1) данные в ячейках таблицы должны быть структурно неделимыми ;
- 2) данные в одном столбце должны быть одного типа;
- 3) каждый столбец должен быть уникальным (недопустимо дублирование столбцов);
- 4) столбцы размещаются в произвольном порядке;
- 5) строки размещаются в таблице также в произвольном порядке;
- 6) столбцы имеют уникальные наименования.

# Объединение

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Сидоров В.А.	1975	Профессор	606
Машкина М.М.	1978	Доцент	605
Дыркин Д.А.	1969	Ассистент	607
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Сидоров В.А.	1975	Профессор	606
Машкина М.М.	1978	Доцент	605
Дыркин Д.А.	1969	Ассистент	607
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605

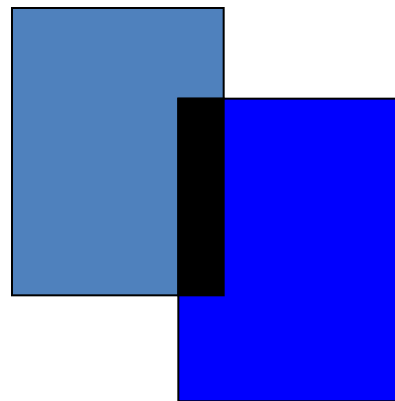


## Пересечение

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Сидоров В.А.	1975	Профессор	606
Машкина М.М.	1978	Доцент	605
Дыркин Д.А.	1969	Ассистент	607
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606

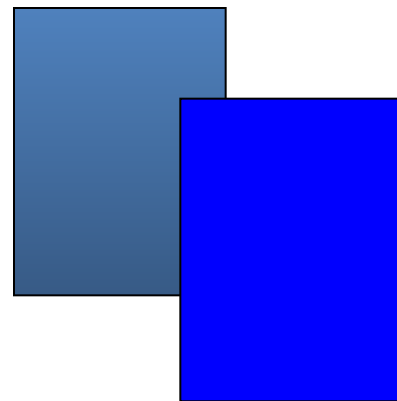


# Разность

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Сидоров В.А.	1975	Профессор	606
Машкина М.М.	1978	Доцент	605
Дыркин Д.А.	1969	Ассистент	607
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605



# Произведение

<b>Job</b>
Профессор
Доцент
Ст.преподаватель
Ассистент

<b>Kaf</b>
606
605

<b>Job</b>	<b>Kaf</b>
Профессор	605
Профессор	606
Доцент	605
Доцент	606
Ст.преподаватель	605
Ст.преподаватель	606
Ассистент	605
Ассистент	606

# Выборка

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Сидоров В.А.	1975	Профессор	606
Машкина М.М.	1978	Доцент	605
Дыркин Д.А.	1969	Ассистент	607
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605

Kaf = 605

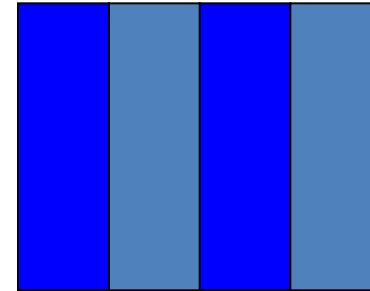


<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Машкина М.М.	1978	Доцент	605
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605



# Проекция

<b>FIO</b>	<b>Year</b>	<b>Job</b>	<b>Kaf</b>
Сидоров В.А.	1975	Профессор	606
Машкина М.М.	1978	Доцент	605
Дыркин Д.А.	1969	Ассистент	607
Петрова М.И.	1965	Ст.преподаватель	607
Сидорчук В.В.	1971	Ассистент	606
Пупкина Л.В.	1972	Доцент	605
Иванова И.И.	1974	Доцент	605



<b>FIO</b>	<b>Job</b>
Сидоров В.А.	Профессор
Машкина М.М.	Доцент
Дыркин Д.А.	Ассистент
Петрова М.И.	Ст.преподаватель
Сидорчук В.В.	Ассистент
Пупкина Л.В.	Доцент
Иванова И.И.	Доцент

## Соединение

<b>ФИО</b>	<b>Job</b>
Сидоров В.А.	Профессор
Машкина М.М.	Доцент
Дыркин Д.А.	Ассистент
Петрова М.И.	Ст.преподаватель
Сидорчук В.В.	Ассистент
Пупкина Л.В.	Доцент
Иванова И.И.	Доцент

<b>Job</b>	<b>ZP</b>
Профессор	10000
Доцент	8000
Ст.преподаватель	5000
Ассистент	4000

<b>ФИО</b>	<b>Job</b>	<b>ZP</b>
Сидоров В.А.	Профессор	10000
Машкина М.М.	Доцент	8000
Дыркин Д.А.	Ассистент	4000
Петрова М.И.	Ст.преподаватель	5000
Сидорчук В.В.	Ассистент	4000
Пупкина Л.В.	Доцент	8000
Иванова И.И.	Доцент	8000

# Деление

Делимое



Должность
Зав. каф
Профессор
Доцент
Ст. препод.
Ассистент

Посредник



Должность	Кафедра
Зав. каф	22
Профессор	22
Доцент	22
Зав. каф	23
Доцент	23
Ст. препод.	24
Ассистент	24

Делитель



Кафедра
22

Кафедра
22
23

Деление

Должность
Зав. каф
Профессор
Доцент

Должность
Зав. каф
Профессор

# Физические модели баз данных

## Организация данных на машинных носителях



An empty rounded rectangular box with a black border and rounded corners. It is positioned above a gray shadowed area that extends across the width of the box.



An empty rounded rectangular box with a black border and rounded corners. It is positioned above a gray shadowed area that extends across the width of the box.



An empty rounded rectangular box with a black border and rounded corners. It is positioned above a gray shadowed area that extends across the width of the box.

# Физические модели баз данных

## Типы записей

- совокупность элементов или агрегатов данных, воспринимаемая и обычно физически отдельно размещаемая в рабочей области памяти прикладной программой как единое целое
- последовательность записей *в логике обработки образует файл*

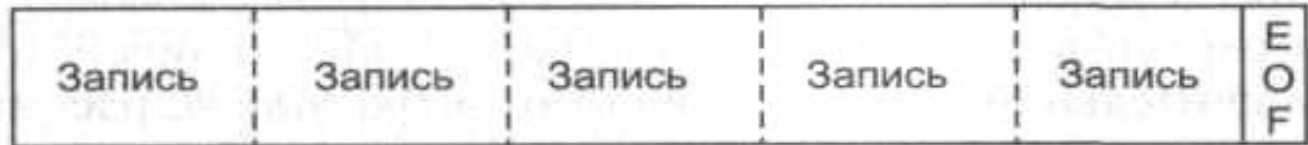
- совокупность данных, которые размещаются в файле обычно на внешнем носителе, и могут быть считаны или записаны как единое целое одной *командой ввода-вывода*
- файл — это последовательность физических записей, размещаемых в линейном пространстве носителя, но в общем случае не обязательно в линейном порядке

# Способы организации файлов данных

Файл

Типы записей

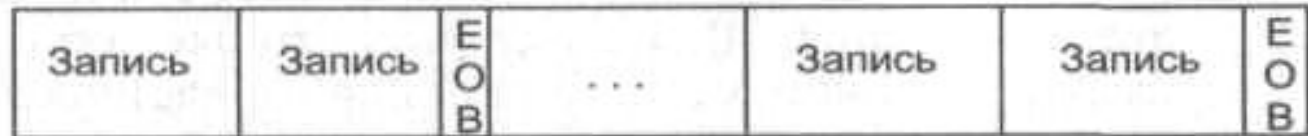
Поток-ориентированный



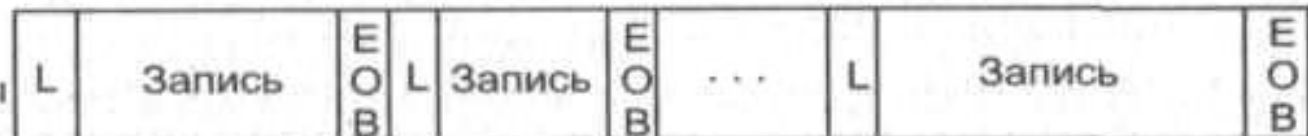
Записей фиксированной длины



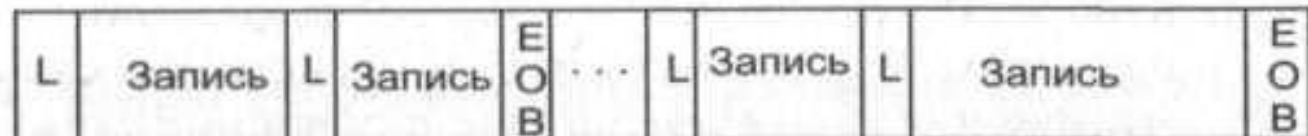
С блокировкой записей фиксированной длины



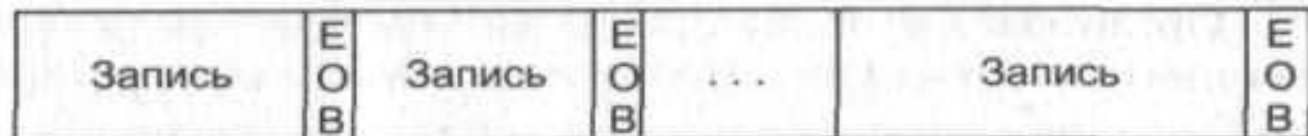
Записей переменной длины



С блокировкой записей переменной длины



Записей неопределенной длины



# Физическое представление с разделением данных и связей

Ф.И.О.	Год	Должность	Кафедра
Иванов И.И.	1948	Зав.кафедрой	22
Сидоров С.С.	1953	Профессор	22
Пупкин В.В.	1945	Профессор	22
Цветкова С.С.	1960	Доцент	22
Козлов К.К.	1959	Доцент	22
Петров П.П.	1960	Ст.преподаватель	22
Лютикова Л.Л.	1977	Ассистент	22
Сидорчук А.В.	1950	Зав.кафедрой	23
Пряников А.К	1944	Профессор	23
Сумкин Ф.Б.	1958	Доцент	23
Соловьев Х.В.	1958	Доцент	23
Воробьева В.В.	1959	Ст.преподаватель	23
Дыркин В.С.	1966	Ассистент	23
Машкина М.М.	1976	Ассистент	23

1	Воробьева В.В.	1	1944	1	Ассистент
2	Дыркин В.С.	2	1945	2	Доцент
3	Иванов И.И.	3	1948	3	Зав.кафедрой
4	Козлов К.К.	4	1950	4	Профессор
5	Лютикова Л.Л.	5	1953	5	Ст.преподаватель
6	Машкина М.М.	6	1958		
7	Петров П.П.	7	1959	1	22
8	Пряников А.К	8	1960	2	23
9	Пупкин В.В.	9	1966		
10	Сидоров С.С.	10	1976		
11	Сидорчук А.В.	11	1977		
12	Соловьев Х.В.				
13	Сумкин Ф.Б.				
14	Цветкова С.С.				

# Требования, предъявляемые к базам данных

1. Описания должны быть понятны пользователю, не проектировавшему базу
2. Однажды принятые способы представления данных должны допускать присоединение новых элементов данных без изменения существующих схем данных и прикладных программ
3. СУБД должны позволять эффективно обрабатывать произвольные запросы к базе данных



# Модели и этапы проектирования баз данных

1. Проектирование базы данных — это упорядоченный формализованный процесс создания *системы взаимосвязанных описаний*, т. е. таких моделей предметной области, которые связывают (фиксируют) хранимые в базе данные с объектами предметной области, описываемыми этими данными
1. Такие описания реализуются, например, в виде *схем*

# Модели и этапы проектирования баз данных

1. Проектирование начинается с анализа предметной области и выявления функциональных и других требований к проектируемой системе
1. Проектирование обычно выполняется человеком (группой людей) — системным аналитиком (а на практике чаще администратором базы данных), которым может быть как специально выделенным сотрудником, так и будущим пользователем базы данных, достаточно хорошо знакомым с машинной обработкой данных

# Модели и этапы проектирования баз данных

1. Объединяя отдельные представления о содержимом базы данных, полученные в результате опроса пользователей, и свои представления о данных, которые могут потребоваться для решения практических задач, системный аналитик сначала создает обобщенное неформальное описание создаваемой базы данных.
2. Это описание, выполненное с использованием естественного языка, математических выражений, таблиц, графов и других средств, понятных всем людям, работающим над проектированием базы данных, называют *инфологической моделью*

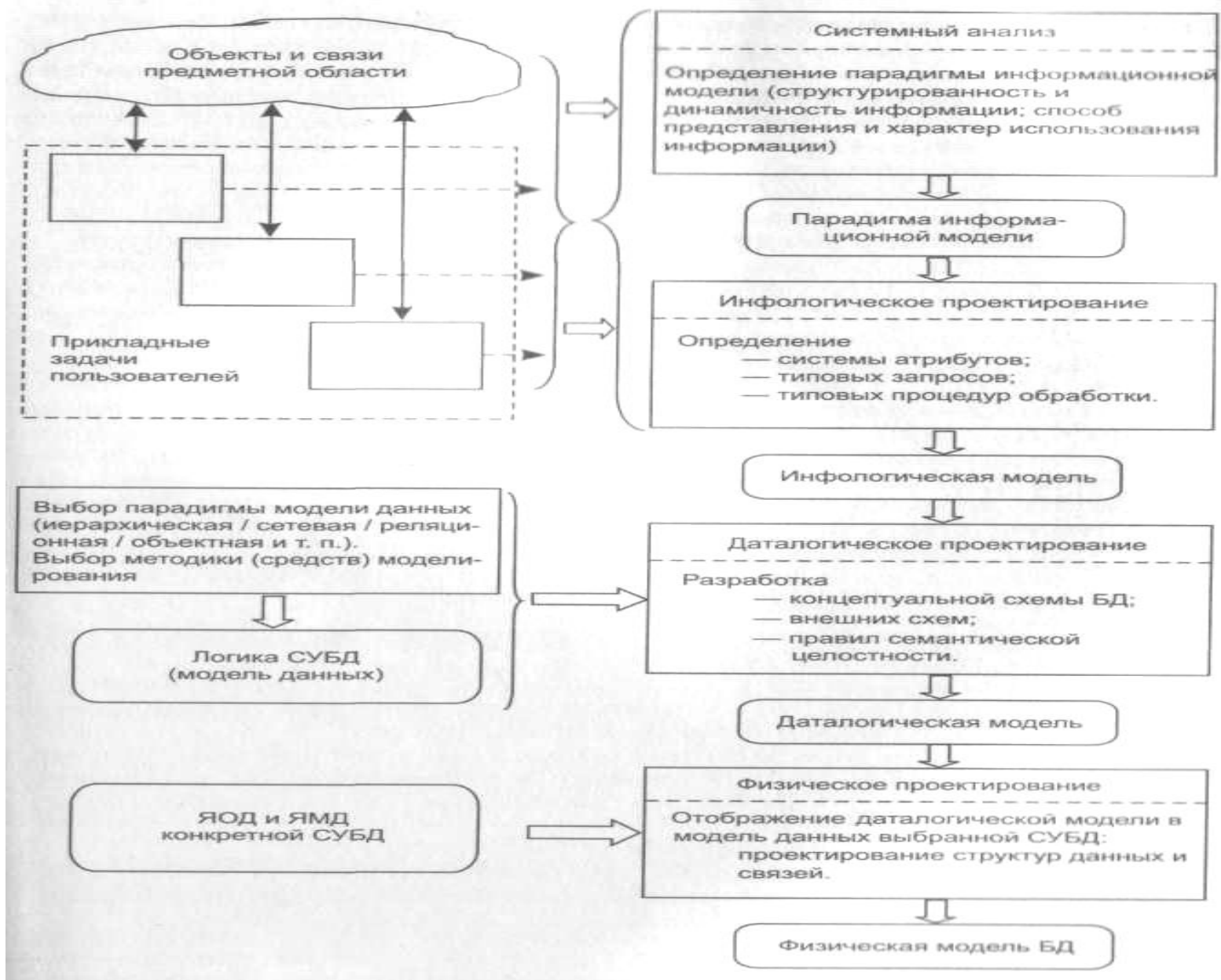
# Модели и этапы проектирования баз данных

1. Инфологическая человеко-ориентированная модель практически полностью независима от физических параметров среды хранения данных, которой может быть как память человека, так и ЭВМ
1. Инфологическая модель не изменяется до тех пор, пока какие-то изменения в реальном мире (той его части, которая отнесена к предметной области) не потребуют изменения в модели соответствующего фрагмента описания, чтобы эта модель продолжала адекватно отражать предметную область.

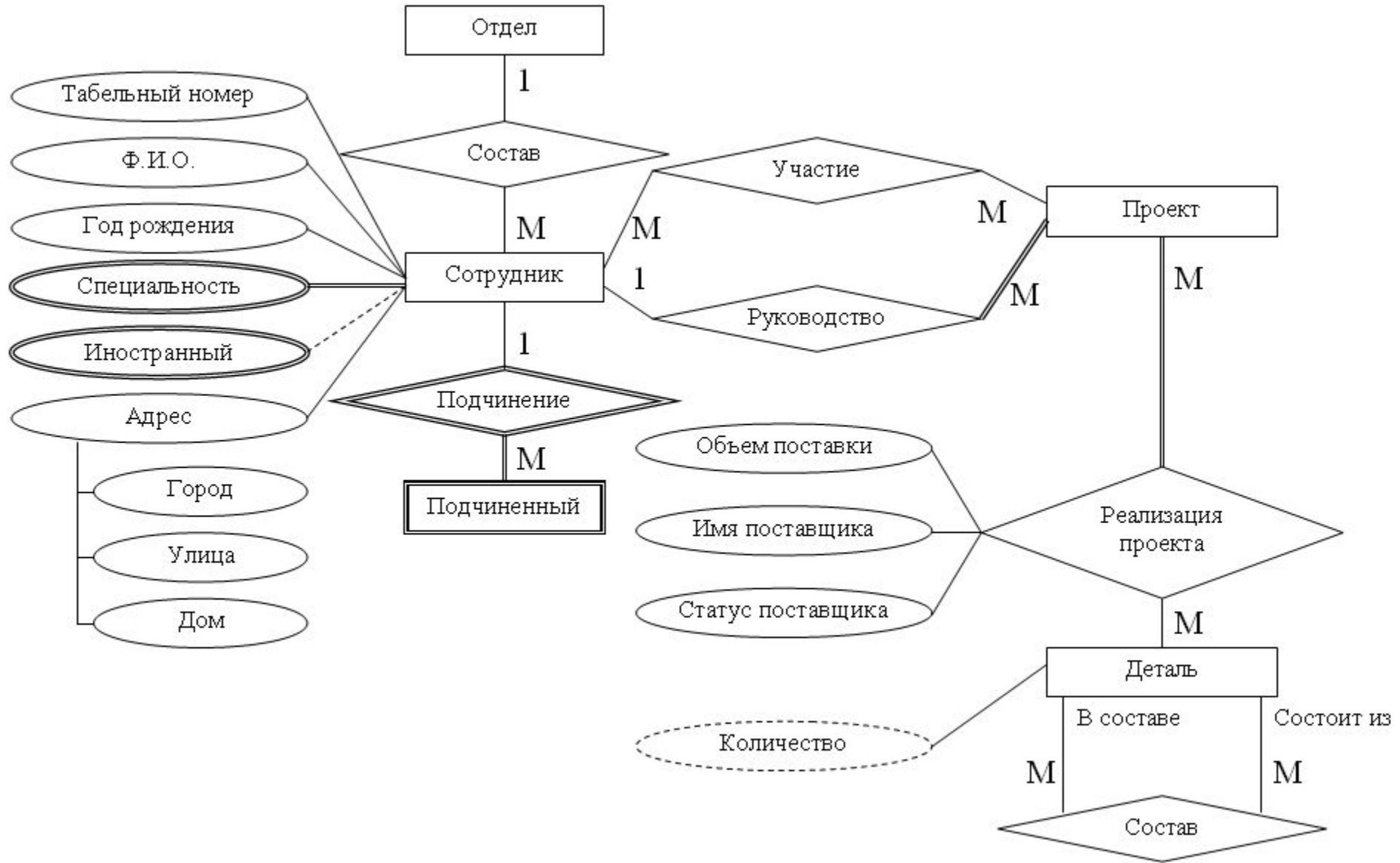
# Модели и этапы проектирования баз данных

1. Все модели, кроме инфологической, являются машинно-ориентированными
1. С их помощью СУБД дает возможность программам и пользователям осуществлять доступ к хранимым данным лишь по их именам, не заботясь о физическом расположении этих данных
1. Так как доступ к данным осуществляется с помощью конкретной СУБД, то модели должны быть представлены на языке описания данных этой СУБД. Такое описание, создаваемое по инфологической модели данных, называют *даталогической моделью* данных.
1. Для размещения и поиска данных на внешних запоминающих устройствах СУБД использует *физическую модель* данных.

# Стадии и объекты процесса проектирования



# Пример ER-диаграммы



## Сущности

Сущность, с помощью которой моделируется класс однотипных объектов, определяется как «предмет, который может быть четко идентифицирован».

Сущность должна *определяться* таким *набором атрибутов*, который позволял бы различать отдельные экземпляры сущности.

Сущность имеет *имя*, уникальное в пределах модели. При этом *имя сущности* — это *имя типа*, а не некоторого конкретного экземпляра.

Сущности подразделяются на *сильные* и *слабые*.



## Свойства

Свойство - *характер связи* свойства с сущностью.

Свойство может быть *множественным* или *единичным*

Свойство может быть *простым* или *составным*

В некоторых случаях полезно различать *базовые* и *производные* свойства

Если наличие некоторого свойства для всех экземпляров сущности не является обязательным, то такое свойство называется *условным*

Значения свойств могут быть постоянными – *статическими* или *динамическими*

Свойство может быть *неопределенным*

Свойство может рассматриваться как *ключевое*

## Связи

Связь определяется как «ассоциация, объединяющая несколько сущностей»

Сущности, объединяемые связью, называются *участниками*.  
Степень связи определяется количеством участников связи

Если каждый экземпляр сущности участвует, по крайней мере, в одном экземпляре связи, то такое участие этой сущности называется *полным* (или *обязательным*); в противном случае — *неполным* (или *необязательным*)

Количественный характер участия экземпляров сущностей (один или многие) задается *типом связи* (или *мощностью связи*).  
Возможны следующие типы: «один к одному» (1:1), «один ко многим» (1:M), «многие ко многим» (M:M).

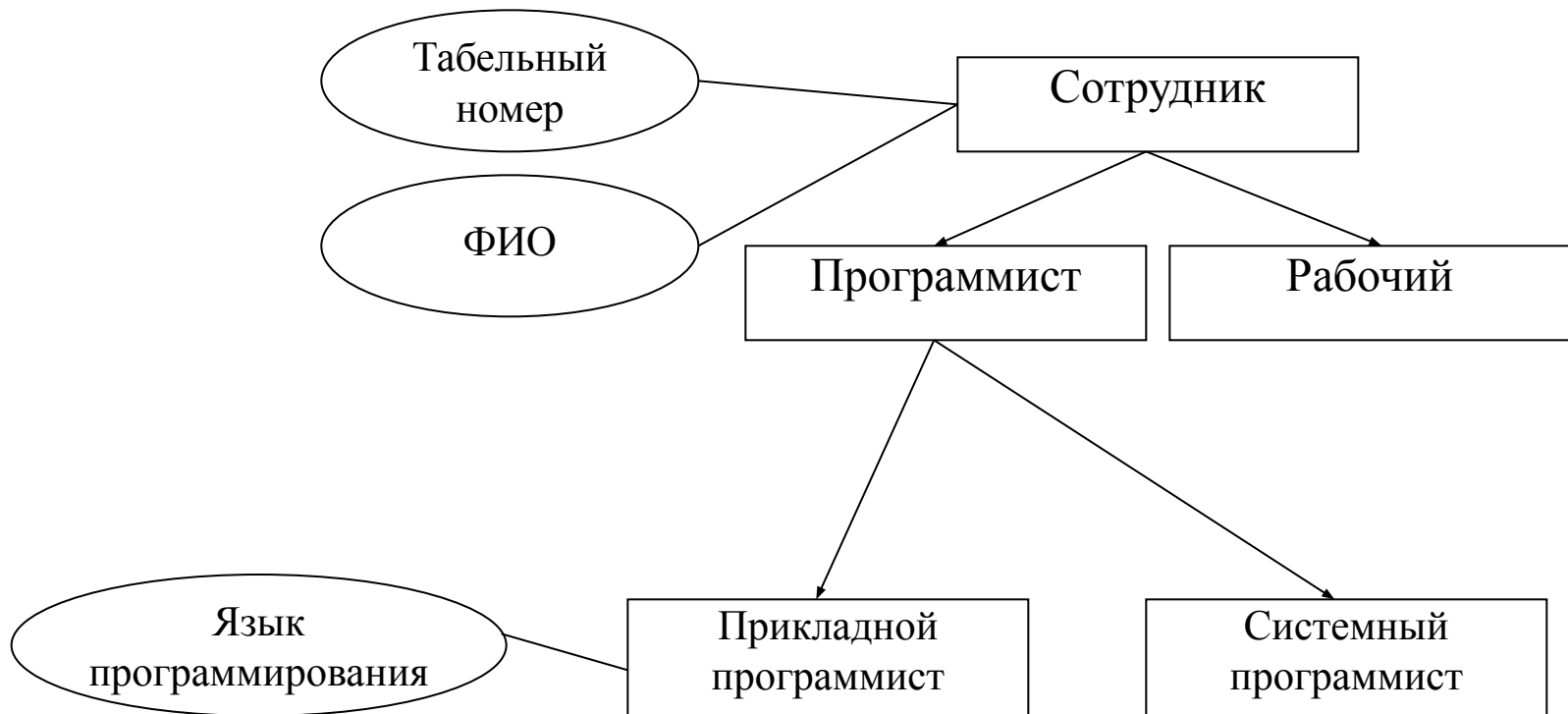
Отношение «часть — целое» используются для представления *составных объектов*.

## Супертипы

Отношение «род – вид» используется для представления *обобщенных объектов*.

Сущность может быть расщеплена на два или более *взаимоисключающих подтипов*, каждый из которых включает общие атрибуты и/или связи.

Сущность, на основе которой определяются подтипы, называется *супертипом*.



## Нотация ER-диаграмм

**Сущности.** Каждый тип сущности в ER-диаграммах представляется в виде прямоугольника, содержащего имя сущности.

В качестве имени обычно используются существительные (или обороты существительного) в единственном числе.

Для отражения сущностей слабых типов используются прямоугольники, стороны которых рисуются двойными линиями.

## Нотация ER-диаграмм

**Свойства.** Свойства служат для уточнения, идентификации, характеристики или выражения состояния сущности или связи. Свойства отображаются в виде эллипсов, содержащих имя свойства. Эллипс соединяется с соответствующей сущностью или связью линией.

Имена ключевых свойств подчеркиваются

Контур эллипса рисуется двойной линией, если свойство многозначное

Контур эллипса рисуется штриховой линией, если свойство производное

Эллипс соединяется пунктирной линией, если свойство условное

Если свойство составное, то составляющие его свойства отображаются другими эллипсами, соединенными с эллипсом составного

## Нотация ER-диаграмм

Связь – это графически изображаемая ассоциация, устанавливаемая между сущностями. Каждый тип связи на ER-диаграмме отображается в виде ромба с именем связи внутри. И в качестве имени обычно используются отглагольные существительные.

Стороны ромба рисуют двойными линиями, если это связь сущности слабого типа с сущностью, от которой она зависит.

Участники связи соединены со связью линиями. Двойная линия обозначает полное участие сущности в связи с данной стороны.

Связь может быть модифицирована указанием роли.

Тип связи указывается индексами «1» или «М» над соответствующей линией

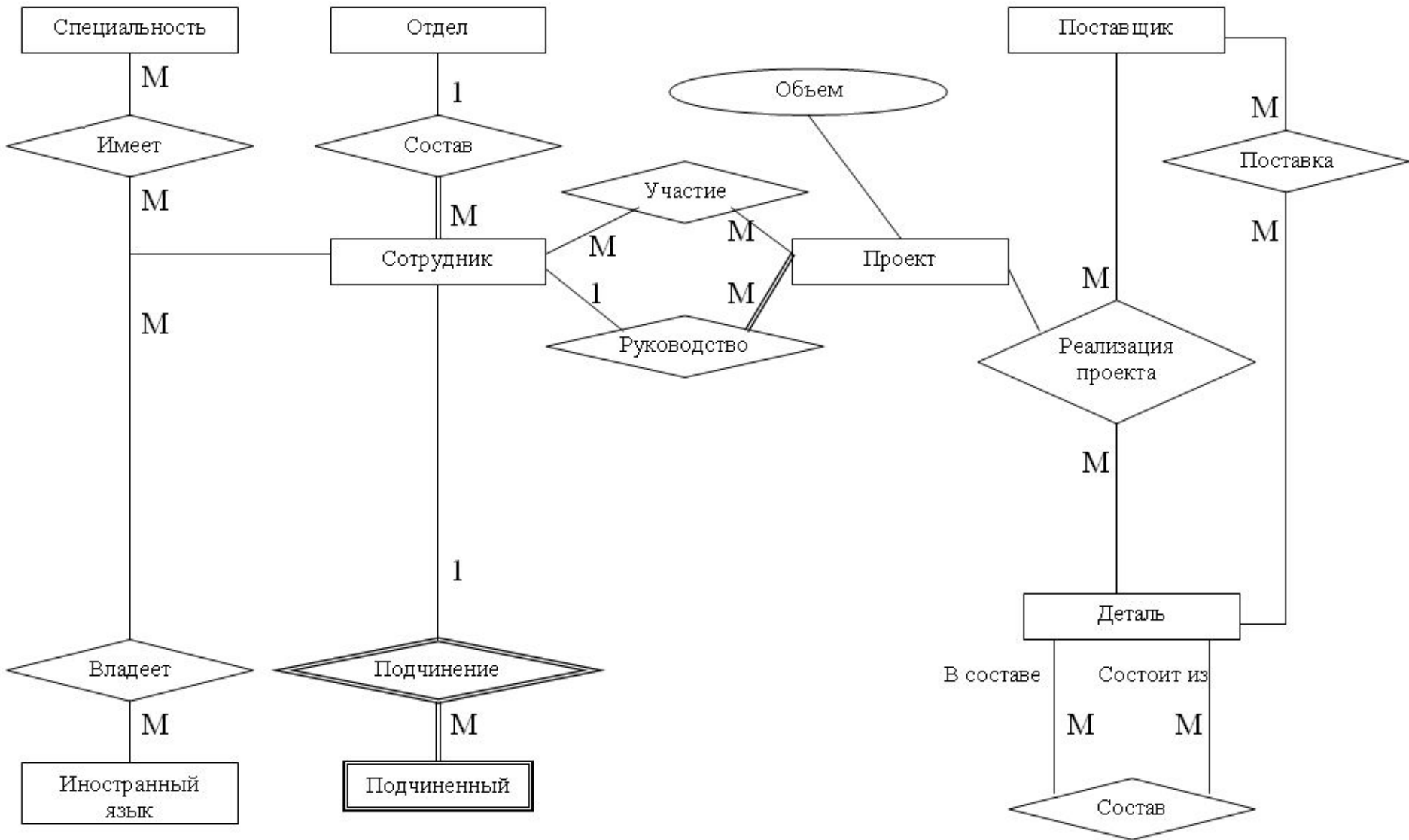
## Нормальные формы ER-диаграмм

В *первой нормальной форме* ER-диаграммы устраняются повторяющиеся атрибуты или группы атрибутов, т. е. производится выявление неявных сущностей, «замаскированных» под атрибуты.

Во *второй нормальной форме* устраняются атрибуты, зависящие только от части уникального идентификатора. Эта часть уникального идентификатора определяет отдельную сущность.

В *третьей нормальной форме* устраняются атрибуты, зависящие от атрибутов, не входящих в уникальный идентификатор. Эти атрибуты являются основой отдельной сущности.

# Пример ER-диаграммы в третьей нормальной форме



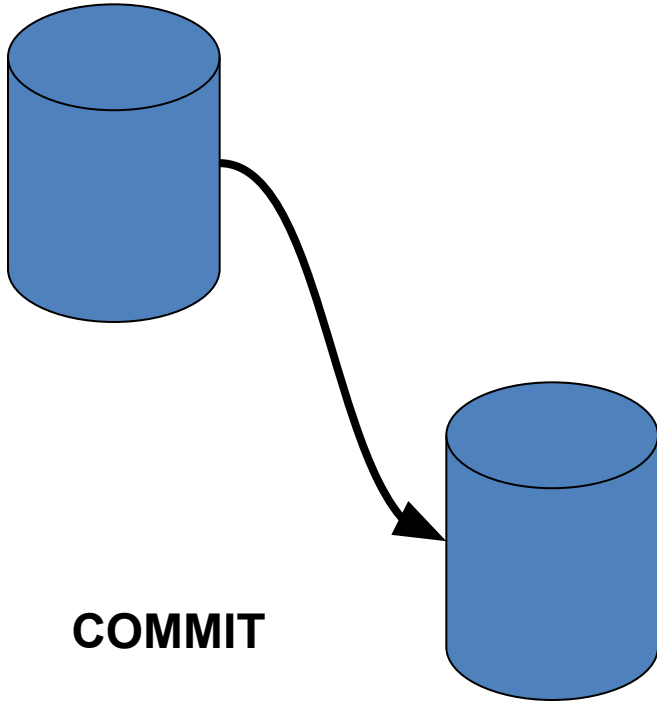


# Транзакции

*Транзакция* – неделимая с точки зрения воздействия на БД последовательность операторов манипулирования данными, такая, что:

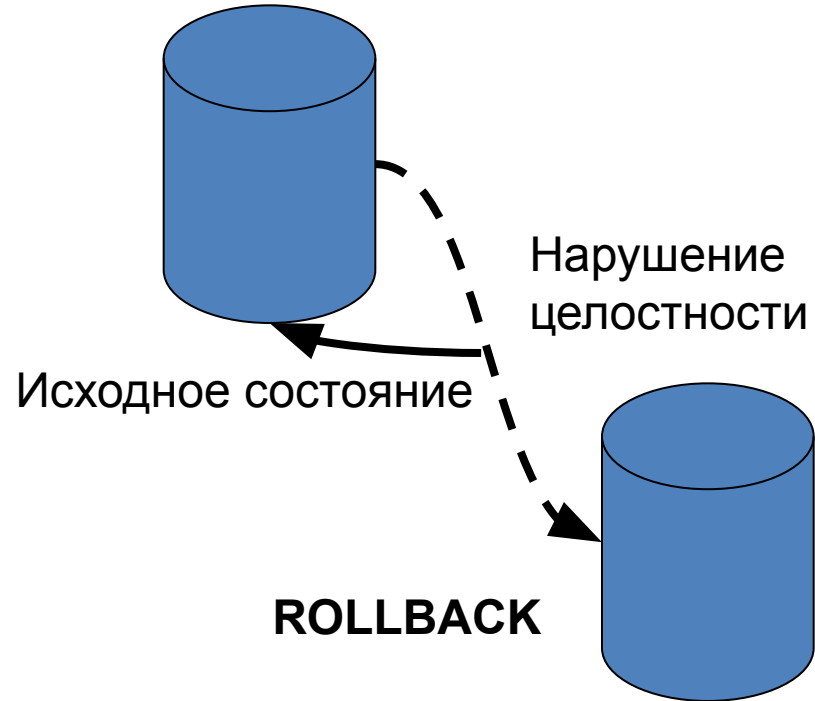
- 1) либо результаты всех операторов, входящих в транзакцию, отображаются в БД;
- 2) либо воздействие всех этих операторов полностью отсутствует.

Исходное состояние

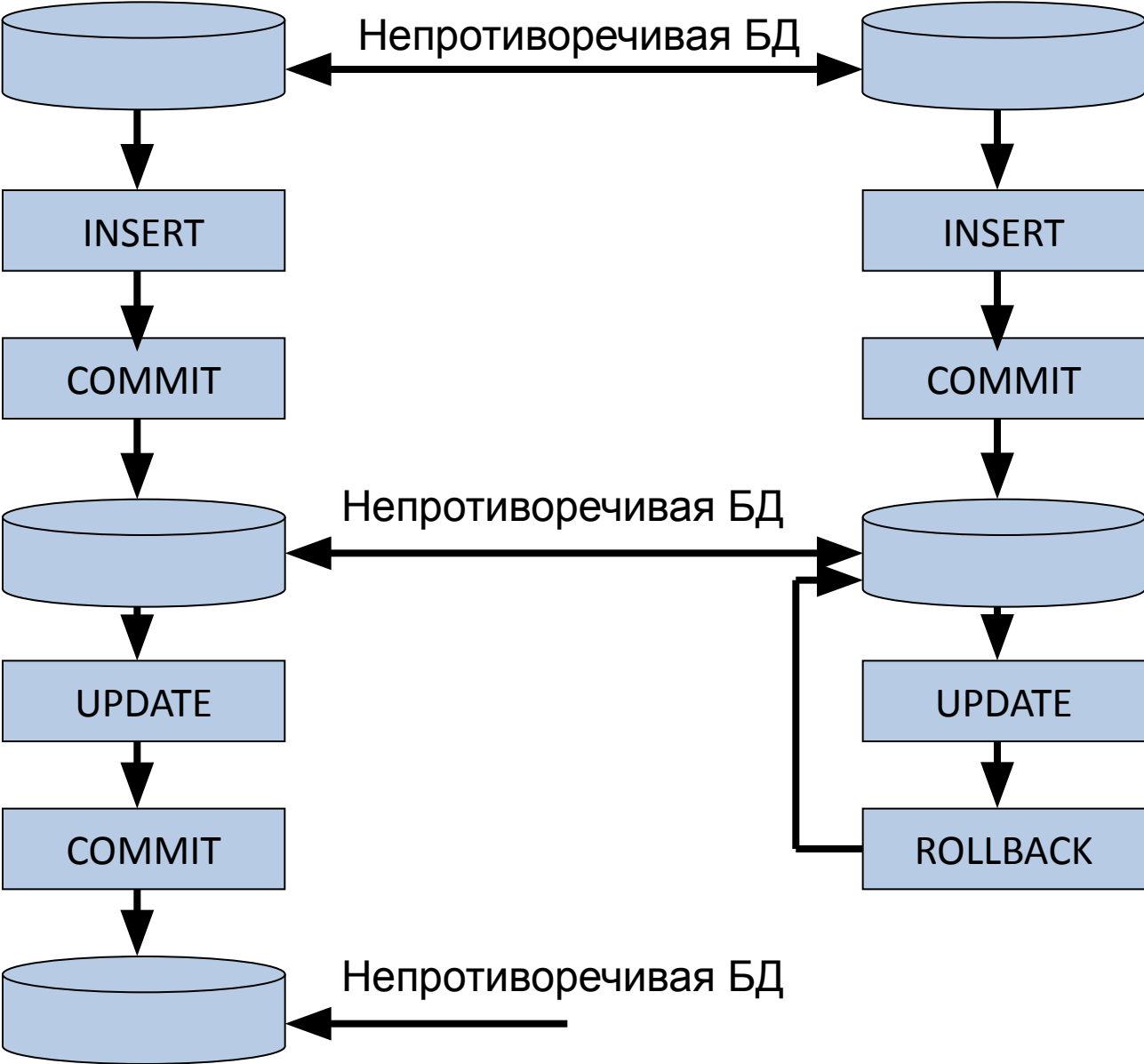


Измененная БД

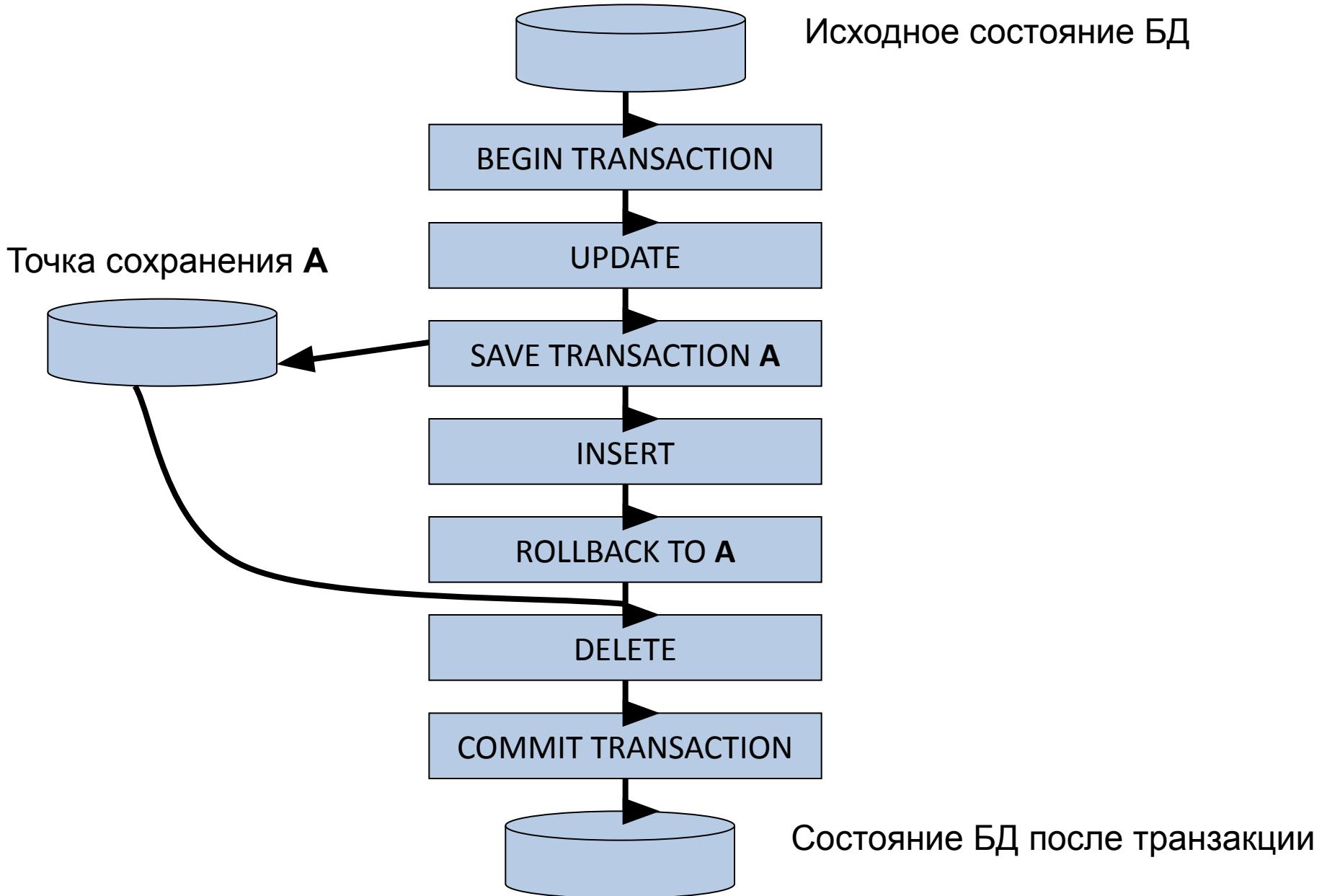
Исходное состояние



# Модель автоматического выполнения транзакций



# Модель управляемого выполнения транзакций



# Журнал транзакций

Протокол журнализации (и управления буферизацией) действует по правилу Write Ahead Log (WAL) — «пиши сначала в журнал», и состоит в том, что если требуется сохранить во внешней памяти измененный объект базы данных, то перед этим нужно гарантировать сохранение во внешней памяти журнала записи о его изменении.

## Общие требования к системе восстановления данных в составе СУБД

1. Пользователь не должен осуществлять рестарт транзакций или повторный ввод данных. Восстановление должно проходить на базе транзакции с помощью отмены или изменения отдельных транзакций.
2. Быстрое восстановление данных обеспечивается генерацией данных, используемых для восстановления.
3. При выполнении процедур автоматизированного восстановления пользователь не должен анализировать состав данных и выбирать сами процедуры.

# Сервисные программные средства для восстановления

**Программы ведения системного журнала** регистрируют операции над БД: описание соответствующей транзакции, код пользователя, текст входного сообщения, тип изменения БД, адреса изменяемых данных вместе с их значениями до и после изменения.

**Программы архивации** используются для регулярного получения копий БД для последующего ее восстановления.

**Программы восстановления** применяются для возврата БД или некоторых ее частей и состояние, предшествующее возникновению отказа. При этом используют архивную копию БД и системный журнал.

**Программы отката** ликвидируют последствия выполнения определенной транзакции в БД.

**Программы записи контрольных точек и повторного исполнения** позволяют ускорить восстановление.

# «Проблема пропавшего изменения»

Кладовщик 1

Запрос количества  
пива на складе для  
ID\_Сорт = 5  
Ответ: 30

ID_Сорт	...	Количество
...	...	...
5	...	30
...	...	...

Заполнение столбца  
ID\_Сорт таблицы  
«Пиво» с  
Количеством 25

Изменение  
значения столбца  
Количество и  
занесение нового  
значения (5) в  
строку таблицы

ID_Сорт	...	Количество
...	...	...
5	...	5
...	...	...

ID_Сорт	...	Количество
...	...	...
5	...	12
...	...	...

Кладовщик 2

Запрос количества  
пива на складе для  
ID\_Сорт = 5  
Ответ: 30

Заполнение столбца  
ID\_Сорт таблицы  
«Пиво» с  
Количеством 18

Изменение  
значения столбца  
Количество и  
занесение нового  
значения (12) в  
строку таблицы



# Проблема чтения «грязных данных» (dirty data)

Кладовщик 1

Запрос количества  
пива на складе для  
ID\_Сорт = 5  
Ответ: 30

ID_Сорт	...	Количество
...	...	...
5	...	30
...	...	...

Заполнение столбца  
ID\_Сорт таблицы  
«Пиво» с  
количеством 30

Увеличение  
значения столбца  
Количество на 30 и  
занесение нового  
значения (60) в  
строку таблицы

ID_Сорт	...	Количество
...	...	...
5	...	60
...	...	...

ROLLBACK  
(возврат к  
исходному  
состоянию)

ID_Сорт	...	Количество
...	...	...
5	...	30
...	...	...

Кладовщик 2

Запрос количества  
пива на складе с  
ID\_Сорт=5  
Ответ: 60

Заполнение столбца  
ID\_Сорт таблицы  
«Пиво» с  
количеством 60

**!! ОШИБКА !!**



# Проблема чтения несогласованных данных

Кладовщик 1

Запрос количества  
пива на складе для  
ID\_Сорт = 5  
Ответ: 30

ID_Сорт	...	Количество
...	...	...
5	...	30
...	...	...



Заполнение столбца  
ID\_Сорт таблицы  
«Пиво» с  
количеством 30



Увеличение  
значения столбца  
Количество на 30 и  
занесение нового  
значения (60) в  
строку таблицы



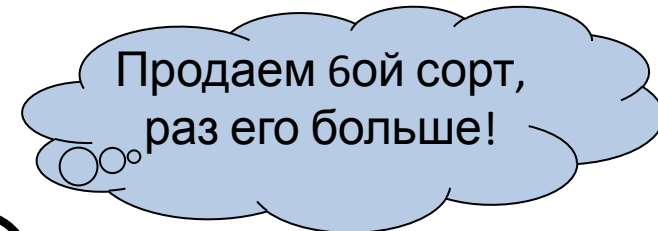
ID_Сорт	...	Количество
...	...	...
5	...	60
...	...	...

Кладовщик 2

Запрос количества  
пива на складе с  
ID\_Сорт=5  
Ответ: 30



Запрос количества  
пива на складе с  
ID\_Сорт=6  
Ответ: 50

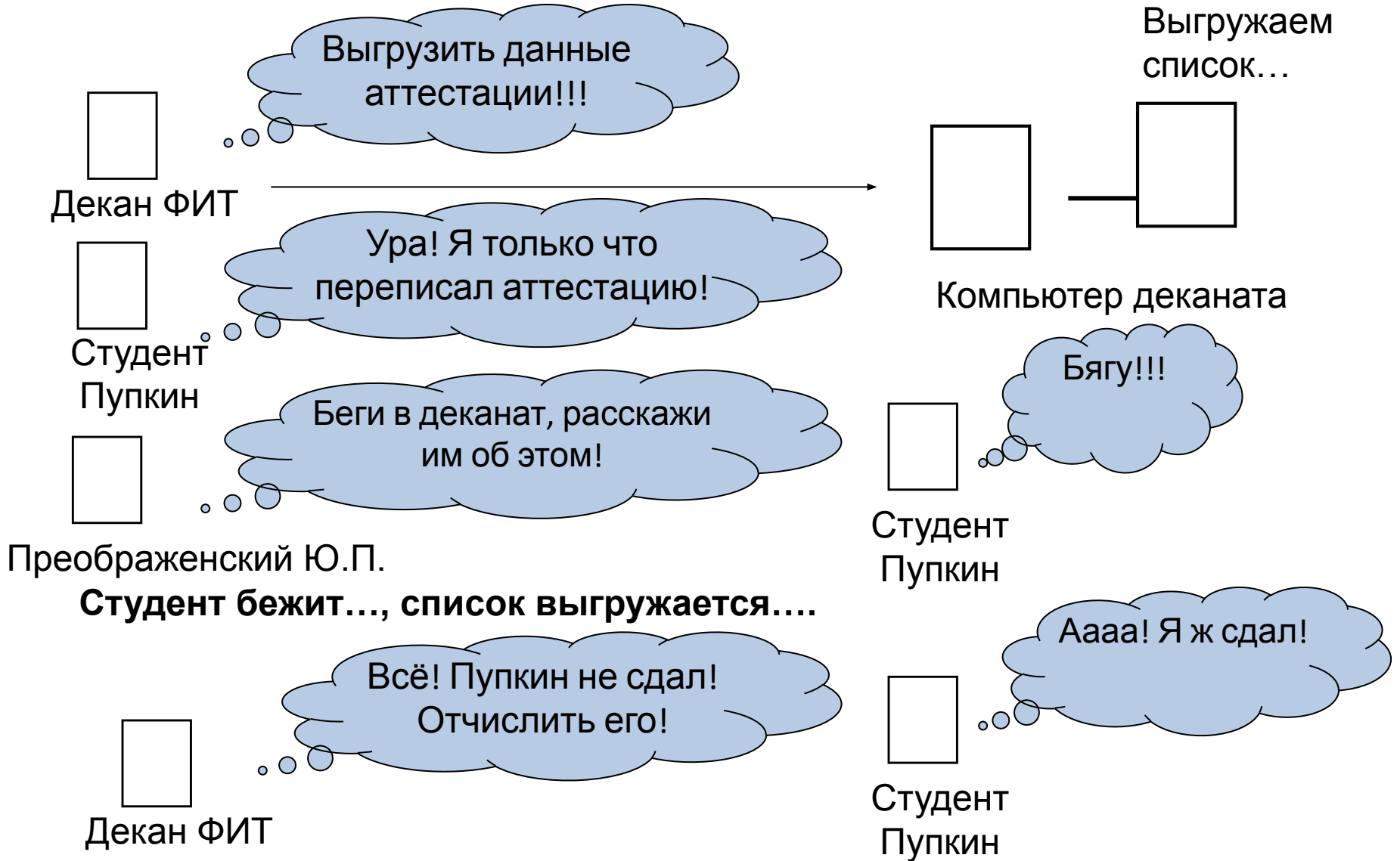


Запрос количества  
пива на складе с  
ID\_Сорт=5  
Ответ: 60



**Не то продали!  
Бестолковые  
менеджеры!**

# Проблема «строк-призраков»



**Список выгружается еще раз.... Пупкин сдал... но его уже отчислили...**

# Сериализация транзакций

Метод *сериализации транзакций* — это механизм их выполнения по такому плану, когда результат совместного выполнения транзакций эквивалентен результату некоторого последовательного выполнения этих же транзакций.

Между транзакциями могут существовать следующие виды конфликтов:

Транзакция 2 пытается изменить объект, измененный незакончившейся Транзакцией 1 (**W-W — конфликт**);

Транзакция 2 пытается изменить объект, прочитанный незакончившейся Транзакцией 1 (**R-W — конфликт**);

Транзакция 2 пытается читать объект, измененный незакончившейся Транзакцией 1 (**W-R — конфликт**).

# Захват и освобождение объекта

Выделяются два основных режима захватов:

**совместный режим** — S (Shared), означающий разделяемый захват объекта и необходимый для выполнения операции чтения объекта;

**монопольный режим** — X (exclusive), означающий монопольный захват объекта и необходимый для выполнения операций записи, удаления и модификации.

# Потенциально возможные объекты для захвата

В контексте реляционных баз данных возможны следующие варианты:

**файл** - физический (с точки зрения базы данных) объект, область хранения нескольких отношений и, возможно, индексов

**таблица** - логический объект, соответствующий множеству записей данного отношения;

**страница данных** - физический объект, хранящий записи одного или нескольких отношений, индексную или служебную информацию;

**запись** - элементарный физический объект базы данных.

# Правила ACID

**Транзакция** — это законченный блок обращений к базе данных и некоторых действий над ней, для которого гарантируется выполнение четырех условий, так называемых свойств **ACID** (**Atomicity, Consistency, Isolation, Durability**):

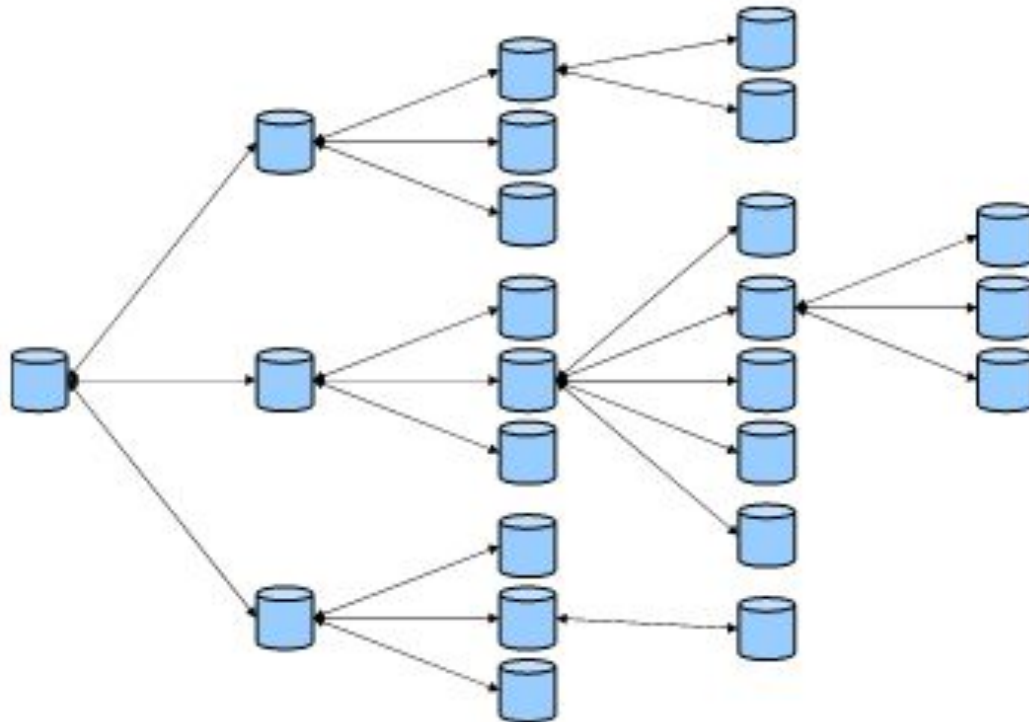
**атомарность** — операции транзакции образуют неразделимый атомарный блок с определенным началом и концом. Этот блок либо выполняется от начала до конца, либо не выполняется вообще. Если в процессе выполнения транзакции произошел сбой, происходит откат к исходному состоянию

**согласованность** — по завершении транзакции все задействованные объекты находятся в согласованном состоянии

**изолированность** — одновременный доступ транзакций различных приложений к разделяемым объектам координируется таким образом, чтобы эти транзакции не влияли друг на друга

**долговременность** — все изменения данных, осуществленные в процессе выполнения транзакции, не могут быть потеряны

# Распределенные базы дан.....



# Основные особенности распределенных баз данных

1. Репликация

1. Удаленные транзакции

1. Распределенные запросы



# OLTP

(Online Transaction Processing)

## **транзакционная система**

обработка транзакций в реальном  
времени

Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы минимальное время отклика.

## **Типовой запрос к OLTP-системе:**

«Каков был уровень импорта товара в январе 2012 года?»

## **Невозможны аналитические запросы вида:**

«Будет ли получена от этого прибыль?»

«Какие клиенты наиболее выгодны с позиции таможенных платежей и почему?»

# Использование OLTP

- OLTP-системы оперативной обработки транзакций, характеризуются большим количеством изменений, одновременным обращением множества пользователей к одним и тем же данным для выполнения разнообразных операций - чтения, записи, удаления или модификации данных. Для нормальной работы множества пользователей применяются блокировки и транзакции. Эффективная обработка транзакций и поддержка блокировок входят в число важнейших требований к системам оперативной обработки транзакций.
- OLTP-системы предназначены для ввода, структурированного хранения и обработки информации (операций, документов) в режиме реального времени.

# Системы OLTP характеризуются:

- поддержкой большого числа пользователей;
- малым временем отклика на запрос;
- относительно короткими запросами;
- короткими транзакциями;
- участие в запросах небольшого числа таблиц.

**Критическим для OLTP-приложений является скорость и надежность выполнения коротких операций обновления данных.**

# Требования к OLTP

- Сильно нормализованные модели данных;
- При возникновении ошибки транзакция должна целиком откатиться и вернуть систему к состоянию, которое было до начала транзакции;
- Обработка данных в реальном времени.

Что происходит?  
Ключевые показатели  
эффективности

Что произошло?  
Отчеты,  
Консолидация

Что произойдет?  
Прогнозирование

Что будет если ...?  
Моделирование

Почему это  
произошло?  
Анализ

Что я хочу чтобы  
произошло?  
Планирование,  
Бюджетирование



# Хранилища данных



Каково среднее значение промежутка времени между выставлением счета и оплатой его участником ВЭД в текущем и прошедшем году для разных групп участников ВЭД ?».



Данные практически не обновляются, а лишь накапливаются.

A screenshot of a data table with a grid interface. The table has several columns and rows of data. The columns are labeled with numbers 1 through 10. The rows are labeled with letters A through J. The data cells contain various numbers and colors (yellow, green, red, blue).

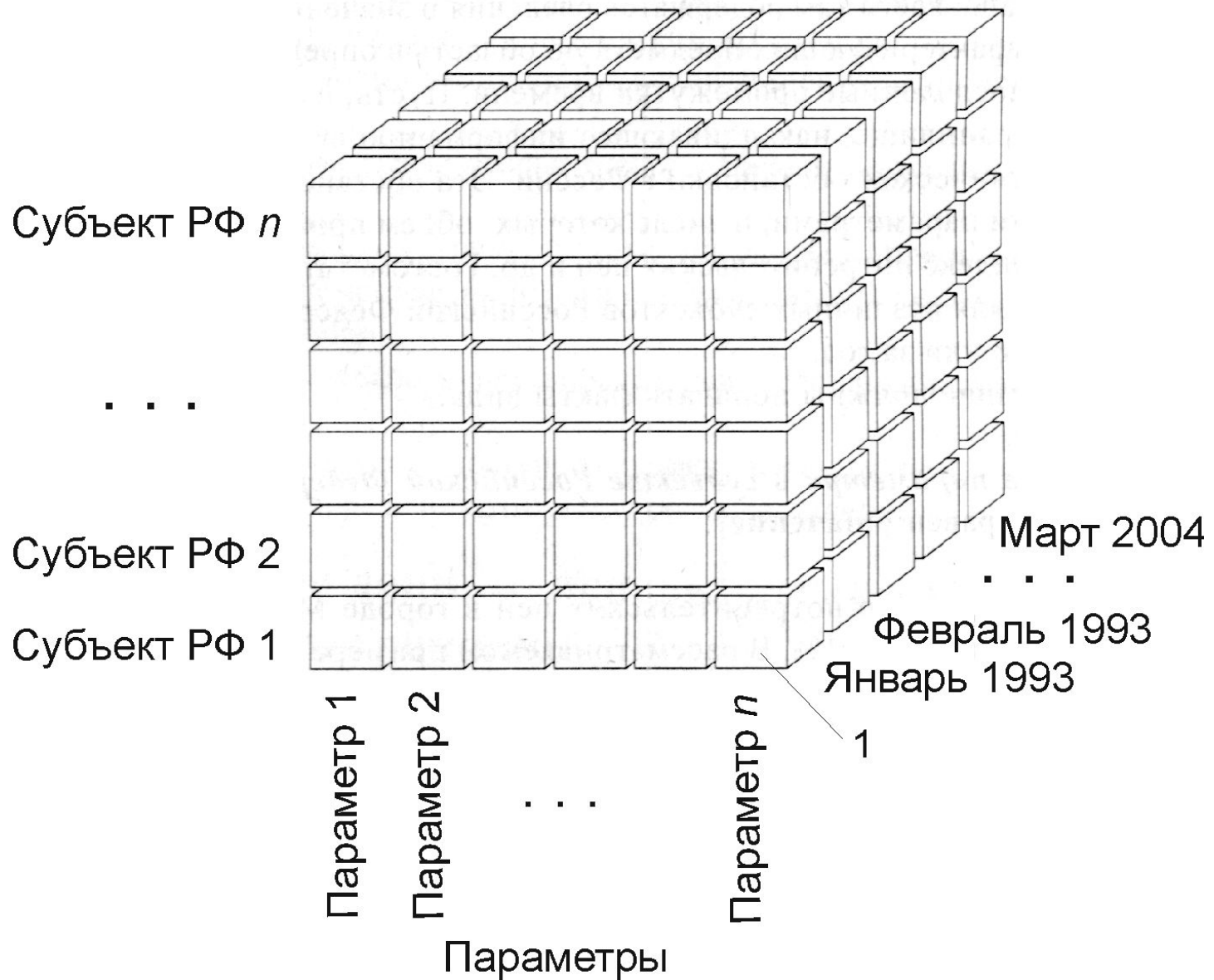
	1	2	3	4	5	6	7	8	9	10
A	1	2	3	4	5	6	7	8	9	10
B	11	12	13	14	15	16	17	18	19	20
C	21	22	23	24	25	26	27	28	29	30
D	31	32	33	34	35	36	37	38	39	40
E	41	42	43	44	45	46	47	48	49	50
F	51	52	53	54	55	56	57	58	59	60
G	61	62	63	64	65	66	67	68	69	70
H	71	72	73	74	75	76	77	78	79	80
I	81	82	83	84	85	86	87	88	89	90
J	91	92	93	94	95	96	97	98	99	100

Необходима хронологическая упорядоченность данных



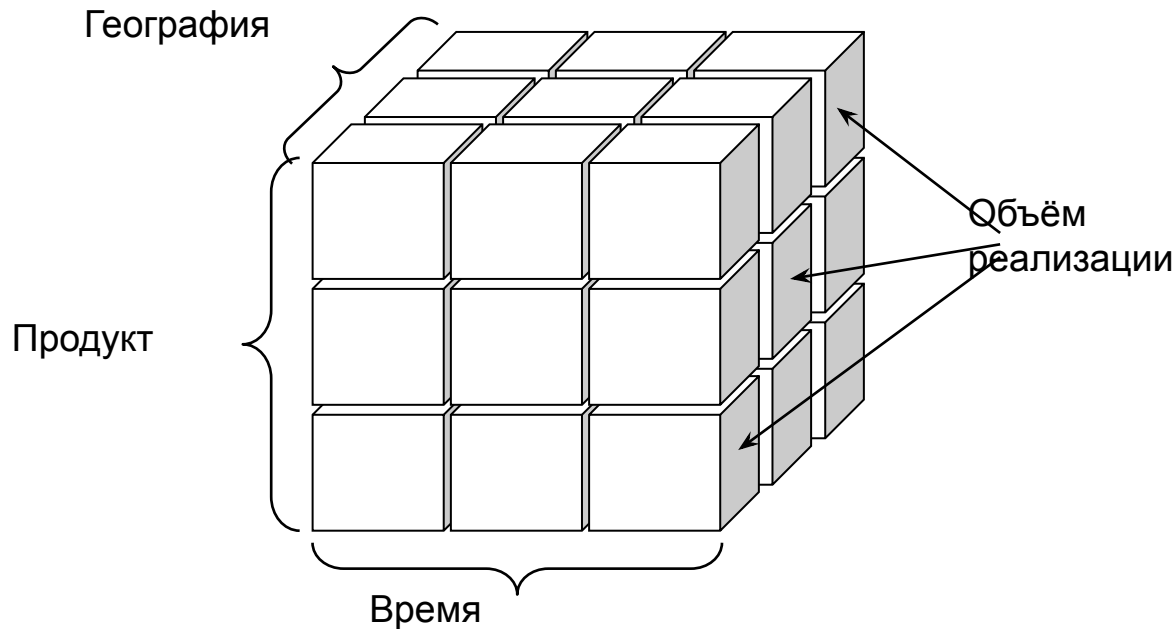
При запросах импорта нет нужды учитывать «каждый контейнер», достаточно иметь агрегированную информацию за прошлый сезон/прошлый год/несколько лет

# Хранилища данных





# *OLAP-технология: быстрый многомерный анализ больших объемов информации для использования в широком контексте управления бизнесом*



## **Измерения:**

- Время
- Продукт
- География

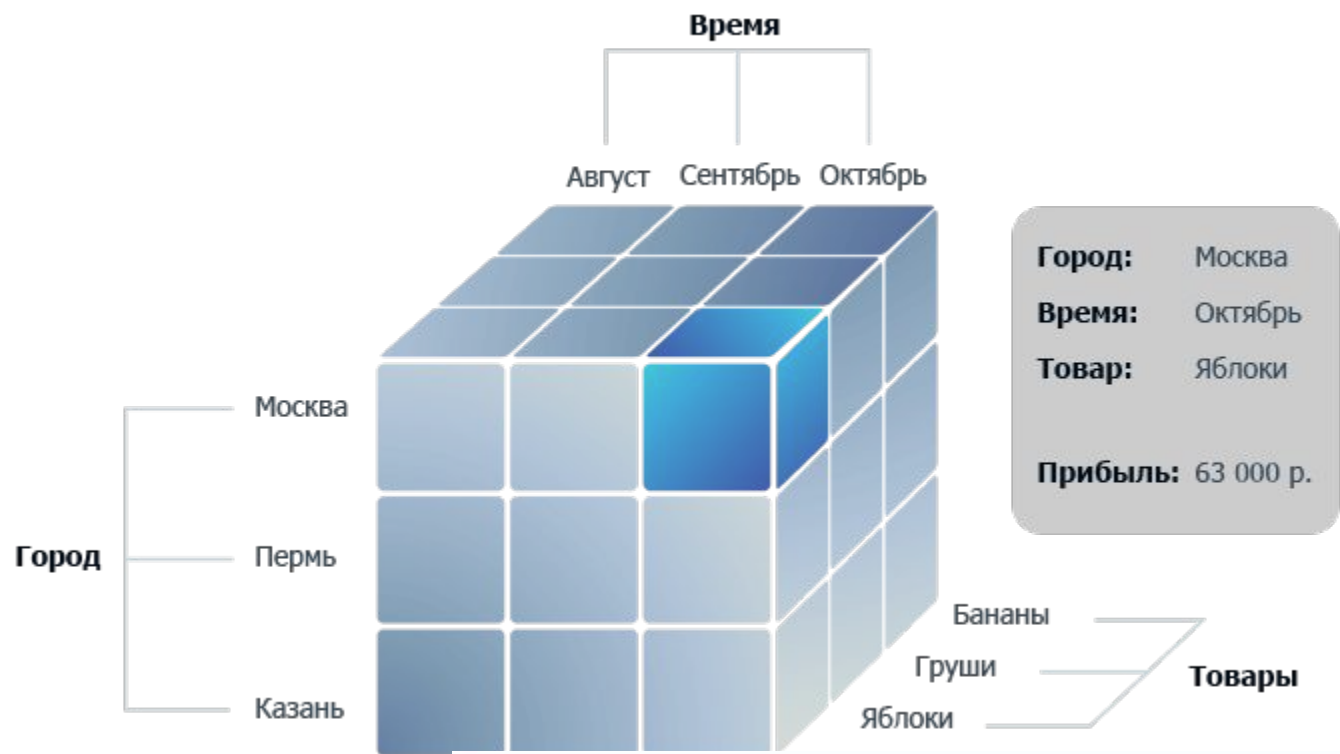
## **Мера:**

- Объем реализации

**Измерения** – факторы, влияющие на деятельность предприятия

**Меры** – количественные показатели деятельности предприятия

**Гиперкуб** – совокупность измерений и мер



		Август	Сентябрь	Октябрь	
<b>Москва</b>	Яблоки	20 000	21 000	22 000	63 000
	Груши	25 000	27 000	29 000	81 000
	Бананы	30 000	33 000	36 000	99 000
	<b>Итого:</b>	75 000	81 000	87 000	243 000
<b>Пермь</b>	Яблоки	18 000	19 000	20 000	57 000
	Груши	20 000	22 000	24 000	66 000
	Бананы	14 000	17 000	30 000	51 000
	<b>Итого:</b>	52 000	58 000	64 000	174 000
<b>Казань</b>	Яблоки	12 000	13 000	14 000	39 000
	Груши	18 000	20 000	22 000	60 000
	Бананы	22 000	25 000	28 000	75 000
	<b>Итого:</b>	52 000	58 000	64 000	174 000
<b>Итого</b>		<b>179 000</b>	<b>197 000</b>	<b>215 000</b>	<b>591 000</b>

# Хранилища данных

Основные операции над кубами данных:

1. Сечение

2. Вращение

3. Свертка

4. Детализация

1	2	3
1	4	7
Январь Февраль	5	8
... Июнь Июль	6 Год	9
... Ноябрь Декабрь		

# OLAP

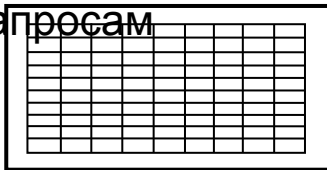
(Online Analytical Processing)

## аналитическая обработка в реальном времени

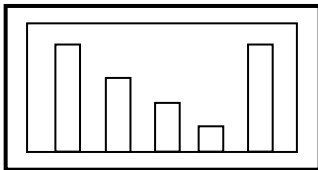
технология обработки данных, заключающаяся в подготовке суммарной (агрегированной) информации на основе больших массивов данных, структурированных по многомерному принципу

это совокупность концепций, принципов и требований, лежащих в основе программных продуктов, облегчающих аналитикам доступ к данным

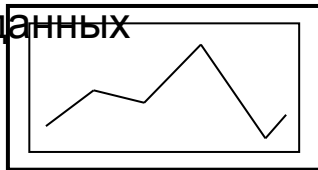
Отчеты по  
запросам



Анализ

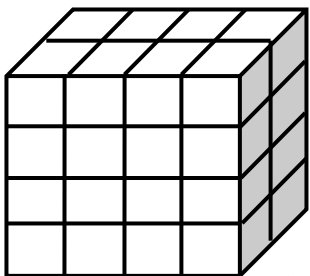


Извлечение  
данных



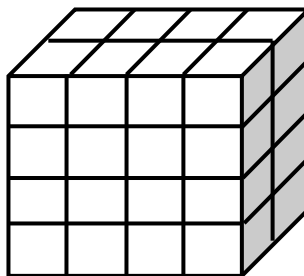
*Инструменты  
пользователя*

OLAP Сервер



Вывод

OLAP Сервер



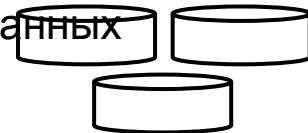
*OLAP-Механизмы*

Мониторинг

Администрирование



Хранилище  
данных

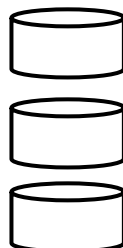


*Хранилище  
данных*

Словарь мета-данных



Витрины данных

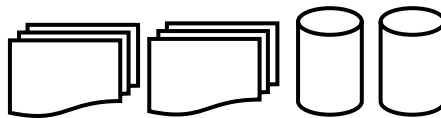


Операционные БД

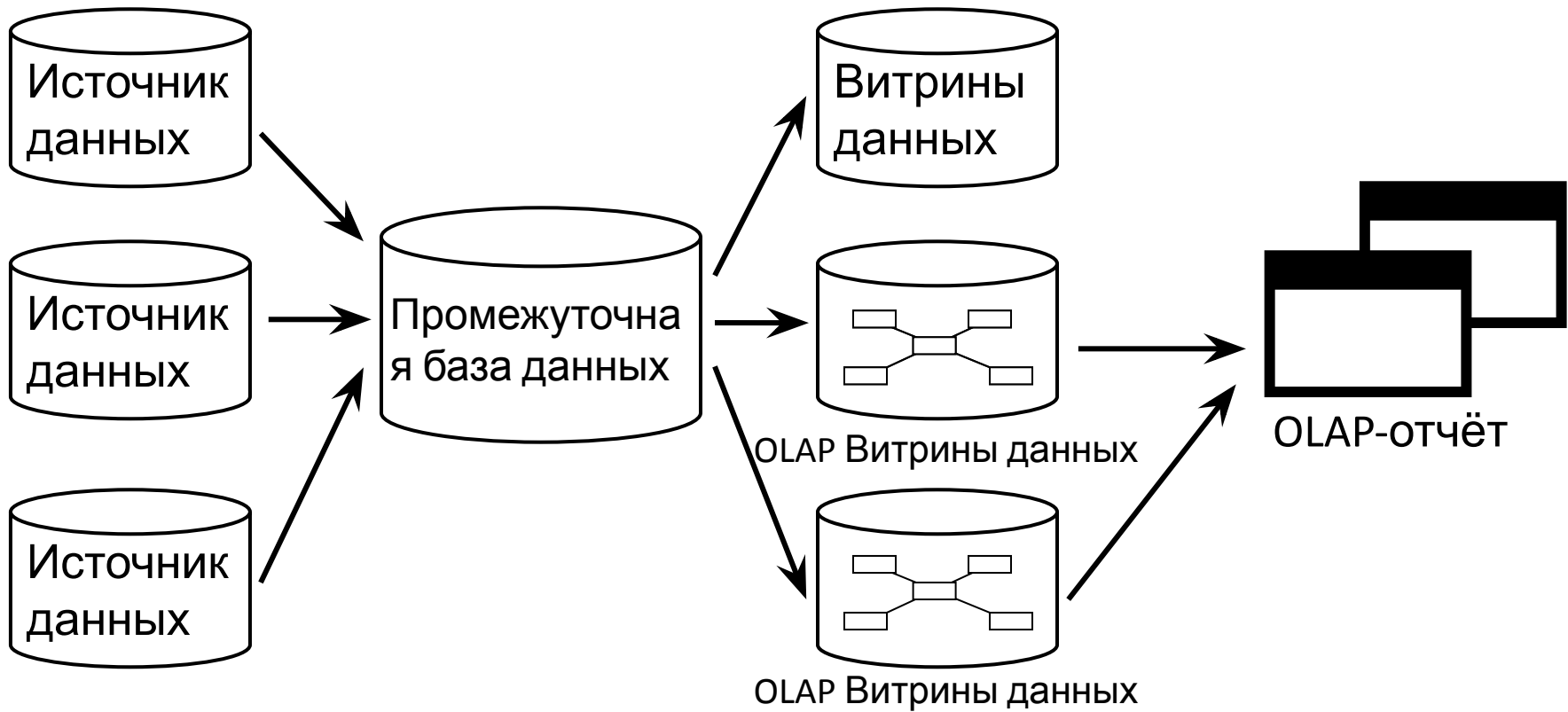


Выделение  
Очистка  
Преобразование  
Загрузка  
Обновление

Внешние источники

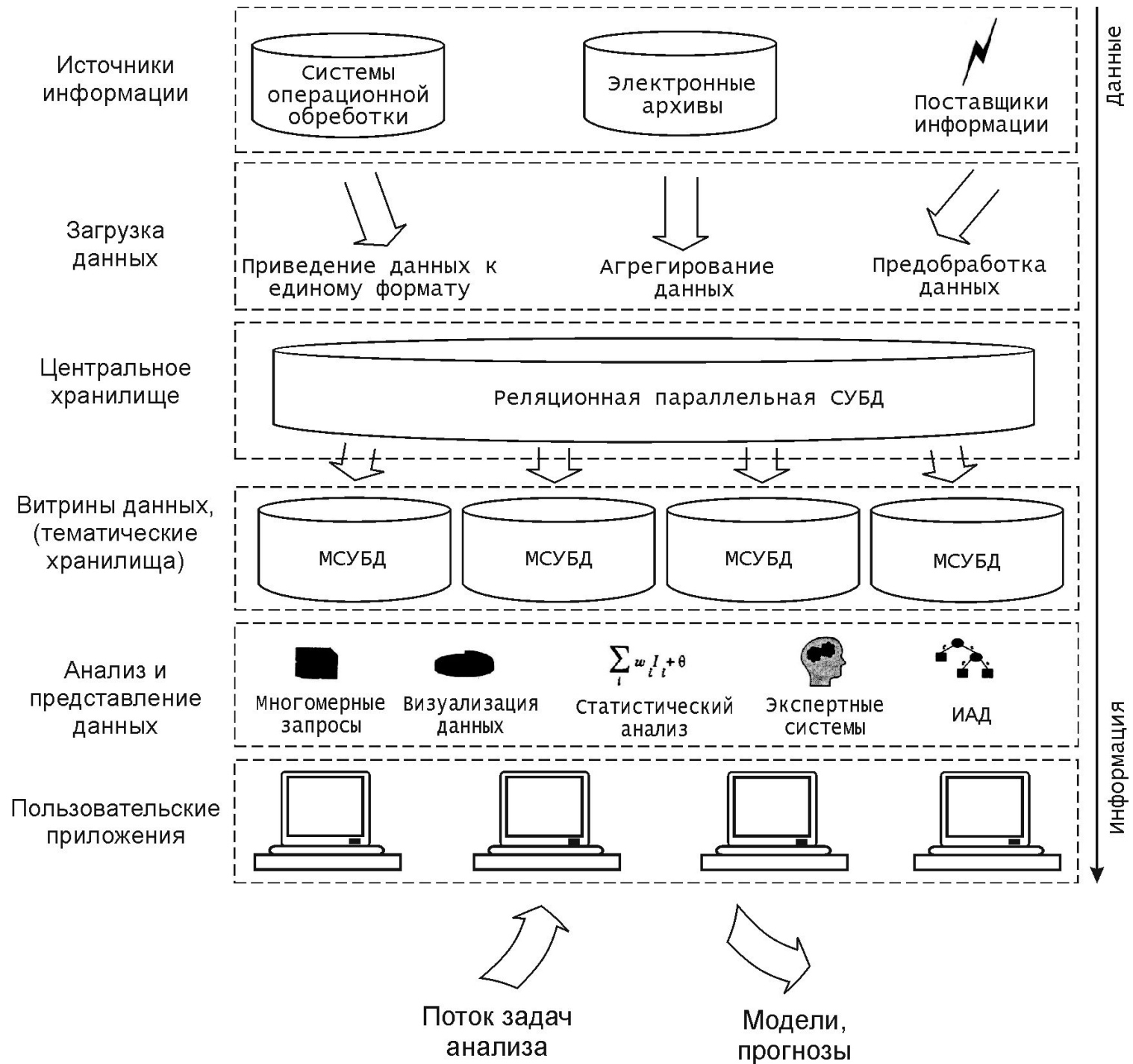


*Интеграция и  
обработка  
данных*



# Критерии OLAP (FASMI)

- **Fast** (*Быстрый*). Приложение OLAP должно обеспечивать минимальное время доступа к аналитическим данным - в среднем порядка 5 секунд;
- **Analysis** (*Анализ*). Приложение OLAP должно давать пользователю возможность осуществлять числовой и статистический анализ;
- **Shared** (*Разделяемый доступ*). Приложение OLAP должно предоставлять возможность работы с информацией многим пользователям одновременно;
- **Multidimensional** (*Многомерность*);
- **Information** (*Информация*). Приложение OLAP должно давать пользователю возможность получать нужную информацию, в каком бы электронном хранилище данных она не находилась.



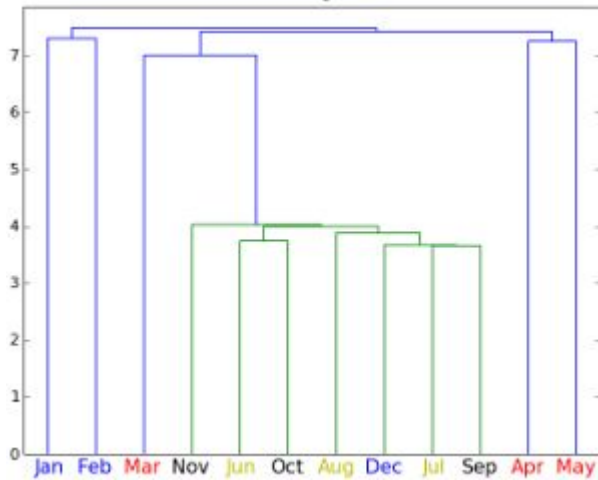
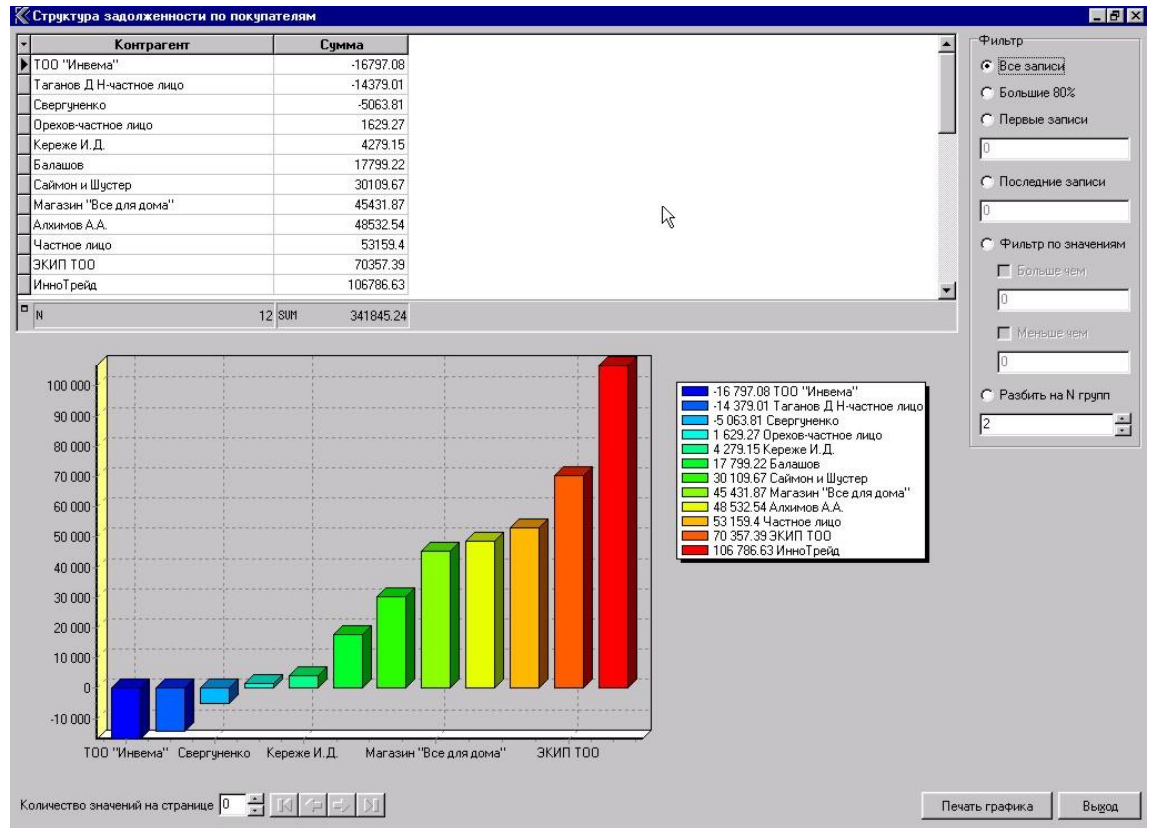
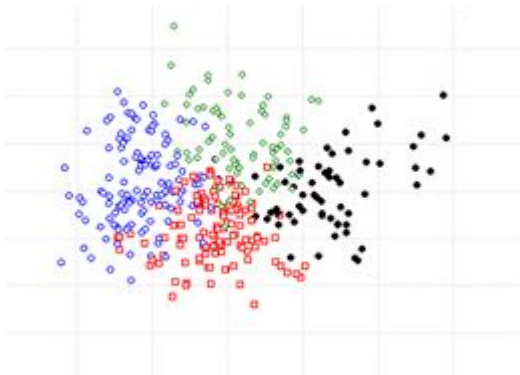


# Методы аналитической обработки данных в хранилище

Традиционные статистические методы регрессионного, факторного, дисперсионного анализа, анализа временных рядов, а также методы, основанные на искусственном интеллекте (нейронные сети, нечеткую логику, генетические алгоритмы, методы извлечения знаний)

Средства анализа данных в СППР на основе хранилищ данных используются для решения следующих **задач**:

- 1) выделение в данных групп сходных по некоторым признакам записей (кластерный анализ);
- 2) нахождение и аппроксимация зависимостей, связывающих анализируемые параметры или события, а также поиск параметров, наиболее значимых в терминах конкретной задачи;
- 3) поиск данных, существенно отклоняющихся от выявленных закономерностей (анализ аномалий);
- 4) прогнозирование развития объектов различной природы на основе хранящейся ретроспективной информации об их состоянии в прошлом.



Показатель	План	Факт	Изменение
Объем реализации продукции, ед.	5700	4850	-850
Цена реализации, млн. руб.	5,0	5,2	0,2
Себестоимость изделия, млн. руб.	4,0	4,5	0,5
В том числе удельные переменные расходы, млн. руб.	2,8	3,1	0,3
Сумма постоянных затрат, млн. руб.	6840	6790	-50
Прибыль, млн. руб.	5700	3395	-2305
Издержки, млн. руб.	22800	21825	-975
Рентабельность, %	25,00	15,56	-9,44
Изменение рентабельности за счет:			
количества реализованной продукции			-6,24
цены реализации			4,75
удельных переменных затрат			-8,21
суммы постоянных затрат			0,26
<b>Итого</b>			<b>-9,44</b>

