

# Архитектура систем базы данных. СУБД

# Рассматриваемые вопросы:

- 1.Трехуровневая архитектура ANSI-SPARC
  - 1.1 Внешний уровень
  - 1.2 Концептуальный уровень
  - 1.3 Внутренний уровень
  - 1.4 Физический уровень
  - 1.5 Независимость от данных
  - 1.6 Отображения
  - 1.7 Схемы БД
- 2.Модели данных и концептуальное моделирование
- 3.Система управления базами данных (СУБД)
  - 2.1 Функции СУБД
  - 2.2 Языки БД
  - 2.3 Компоненты СУБД
- 4.Система управления передач данных
- 5.Многопользовательская обработка средствами СУБД. Архитектура клиент-сервер
  - 3.1 Телеобработка
  - 3.2 Файловый сервер
  - 3.3 Технология клиент-сервер
- 6.Системные каталоги
- 7.Служба IRDS, как стандарт словарей данных
- 8.Утилиты

# СУБД

**Основная цель системы управления базами данных (СУБД)** заключается в том, чтобы *предложить пользователю абстрактное представление данных, скрыв конкретные особенности хранения и*

Главной задачей проектирования базы данных должно быть **абстрактное и общее описание информационных потребностей** организации, которые должны найти свое отражение в создаваемой базе данных.

Поскольку база данных является **общим ресурсом**, то **каждому пользователю** может потребоваться **свое, отличное от других представление** о характеристиках информации, сохраняемой в базе данных.

# Трехуровневая архитектура ANSI-SPARC

Для удовлетворения потребности *коллективного использования* структур данных при *их индивидуальном представлении* разработана **архитектура ANSI-SPARC.**

Три уровня абстракции, уровни описания элементов данных, которые формируют трехуровневую архитектуру:

- Внешний уровень
- Концептуальный уровень
- Внутренний уровень

# Трехуровневая архитектура ANSI-SPARC

**Цель трехуровневой архитектуры** заключается в *отделении пользовательского представления базы данных от ее физического представления.*

**Причины** разделения на *три уровня:*

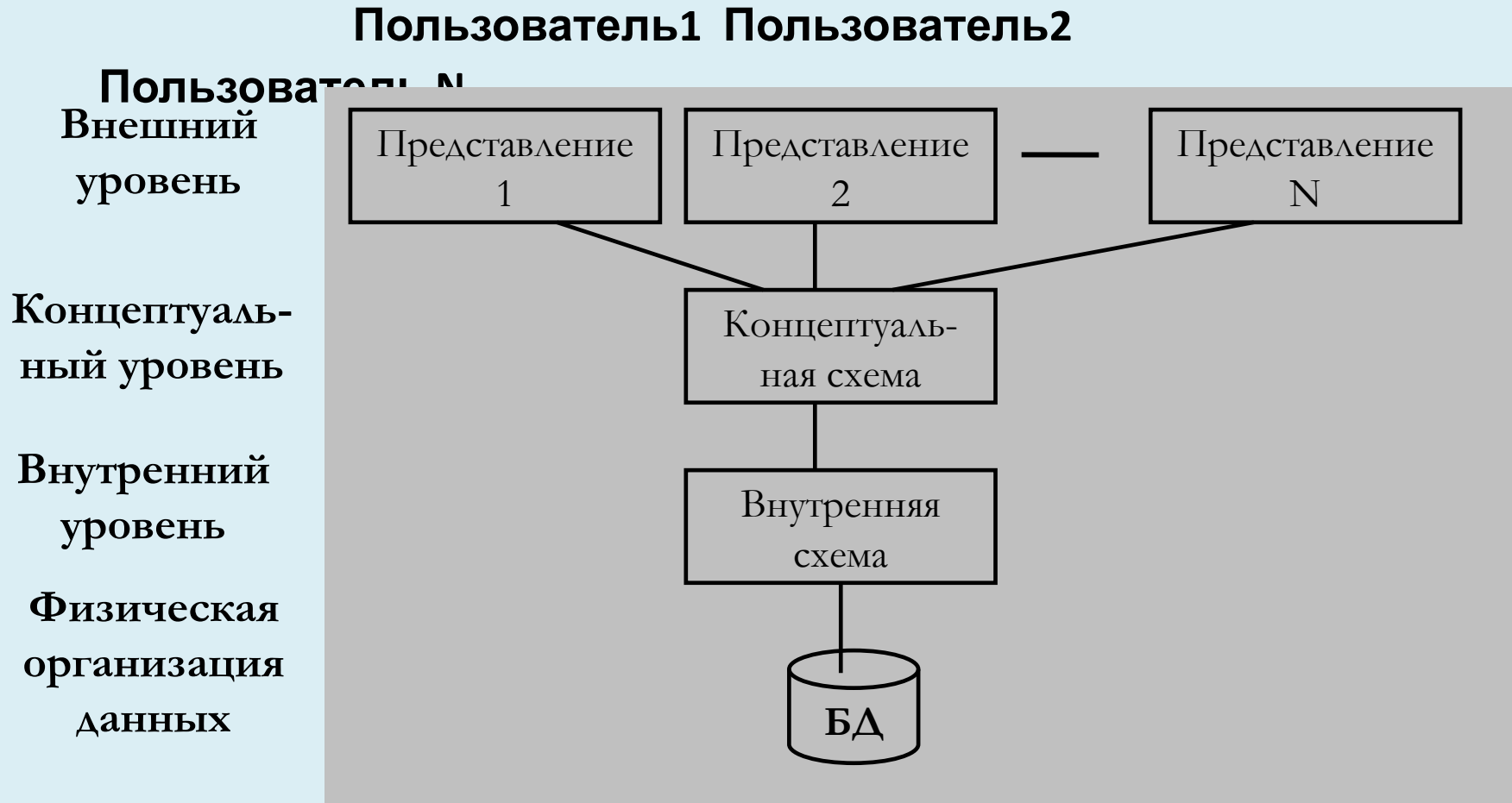
- *Каждый пользователь должен иметь возможность обращаться к одним и тем же данным, используя свое собственное представление о них; изменять свое представление о данных, причем это изменение не должно оказывать влияния на других пользователей.*
- *Пользователи не должны непосредственно иметь дело с такими подробностями физического хранения данных в базе, как индексирование и хеширование, т.е. взаимодействие пользователя с базой не должно зависеть от особенностей хранения в ней данных.*

# Трехуровневая архитектура ANSI-SPARC

**Причины** разделения на *три* уровня;.

- *Администратор БД (АБД) должен иметь возможность изменять структуру хранения данных в базе, не оказывая влияния на пользовательские представления.*
- *Внутренняя структура базы данных не должна зависеть от таких изменений физических аспектов хранения информации, как переключение на новое устройство хранения.*
- *АБД должен иметь возможность изменять концептуальную или глобальную структуру базы данных без какого-либо влияния на всех пользователей.*

# Трехуровневая архитектура ANSI-SPARC



# Внешний уровень

**Внешний уровень (external level)** – это *представление базы данных с точки зрения пользователей, описывает ту часть базы данных, которая относится к каждому пользователю.*

*СУБД и операционная система воспринимают данные на внутреннем уровне (internal level).*

Внешний уровень состоит из нескольких различных внешних представлений базы данных, имеет дело с представлением "реального мира", выраженным в наиболее удобной для него форме.

Внешнее представление содержит только те **сущности, атрибуты и связи** "реального мира", которые интересны пользователю.

**Другие сущности, атрибуты или связи**, которые ему неинтересны, также могут быть **представлены в базе данных**, но **пользователь** может даже **не подозревать об их существовании**.

**Различные представления могут по-разному отображать одни и те же данные.**

Некоторые **внешние представления могут включать производные или вычисляемые данные**, которые не хранятся в базе данных как таковые, а создаются по мере надобности.

Представления могут также включать **комбинированные или производные данные из нескольких объектов** .



# Концептуальный уровень

Промежуточным уровнем в трехуровневой архитектуре является **концептуальный уровень**

**Концептуальный уровень** – это *обобщающее представление базы данных, описывает то, какие данные хранятся в базе данных, а также связи, существующие между ними, содержит логическую модель всей базы данных и не содержит никаких сведений о методах хранения данных*

На концептуальном уровне представлены следующие компоненты:

- все сущности, их атрибуты и связи;
- накладываемые на данные ограничения;
- семантическая информация о данных;
- информация о мерах обеспечения безопасности и поддержки целостности данных.

# Внутренний уровень

**Внутренний уровень** – это низкоуровневое представление всей базы данных как базы, состоящей из некоторого множества экземпляров каждого из существующих типов внутренних записей.

**Внутреннее представление**, так же как внешнее и концептуальное, отделено от **физического уровня**, поскольку в нем не рассматриваются физические записи, обычно называемые *блоками* или *страницами*, и физические области устройства хранения, такие как цилиндры и дорожки.

**Блоки (или страницы) устройства ввода-вывода** – это количество данных, передаваемых из вторичной памяти (памяти накопителя) в основную (оперативную) память за одну операцию ввода-вывода.

Внутреннее представление предполагает наличие **бесконечного линейного адресного пространства**.

# Внутренний уровень

Внутреннее представление описывается с помощью ***внутренней схемы***.

**Внутренняя схема** определяет *не только различные типы хранимых записей, но также существующие индексы, способы представления хранимых полей, физическую упорядоченность хранимых записей и т.д.*

**Внутренняя схема** формируется с использованием еще одного языка определения данных – *внутреннего*.

# Физический уровень

Ниже внутреннего уровня находится **физический уровень**, *который контролируется операционной системой под руководством СУБД.*

Состоит только из известных операционной системе элементов.

# Независимость от данных

*Основным назначением трехуровневой архитектуры является обеспечение **независимости от данных**, которая означает, что изменения на нижних уровнях никак не влияют на верхние уровни.*

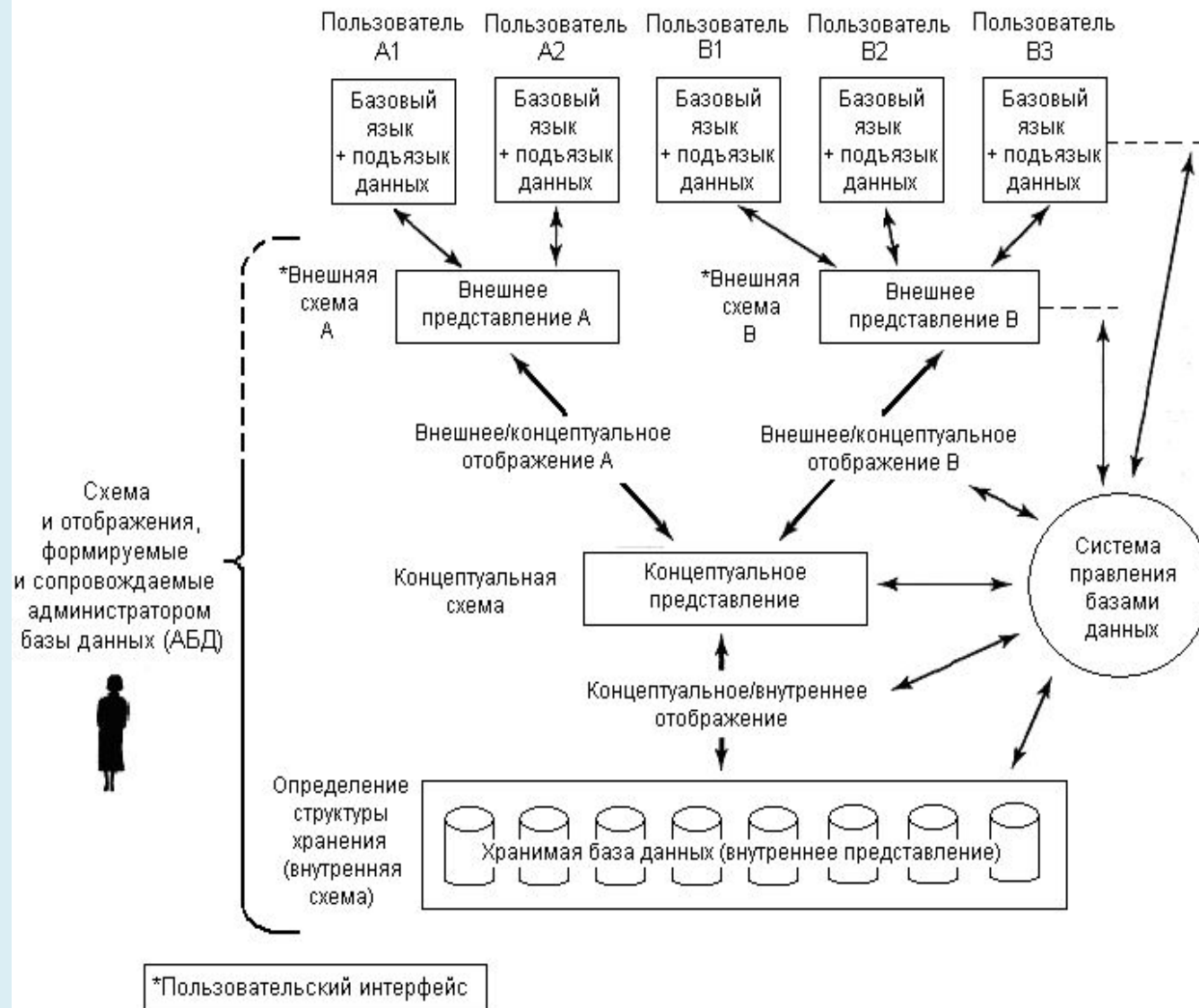
Различают два типа независимости от данных: **логическую** и **физическую**.

# Независимость от данных

**Логическая независимость от данных** означает полную защищенность внешних схем от изменений, вносимых в концептуальную схему.

**Физическая независимость от данных** означает защищенность концептуальной схемы от изменений, вносимых во внутреннюю схему.

# Независимость данных



# Отображения

Архитектура системы баз данных, кроме элементов самих трех уровней, включает определенные *отображения*:

- отображения *концептуального уровня на внутренний*
- *несколько отображений внешних уровней на концептуальный.*



# Отображения

## **Отображение «концептуальный-внутренний»**

устанавливает соответствие между концептуальным представлением и хранимой базой данных, т.е. описывает, *как концептуальные записи и поля представлены на внутреннем уровне.*

## **Отображение «внешний-концептуальный»**

определяет соответствие между некоторым внешним представлением и концептуальным представлением.

Отображение «концептуальный-внутренний» служит основой ***физической независимости от данных,***

а отображения «внешний-концептуальный» являются ключом ***к логической независимости от данных.***

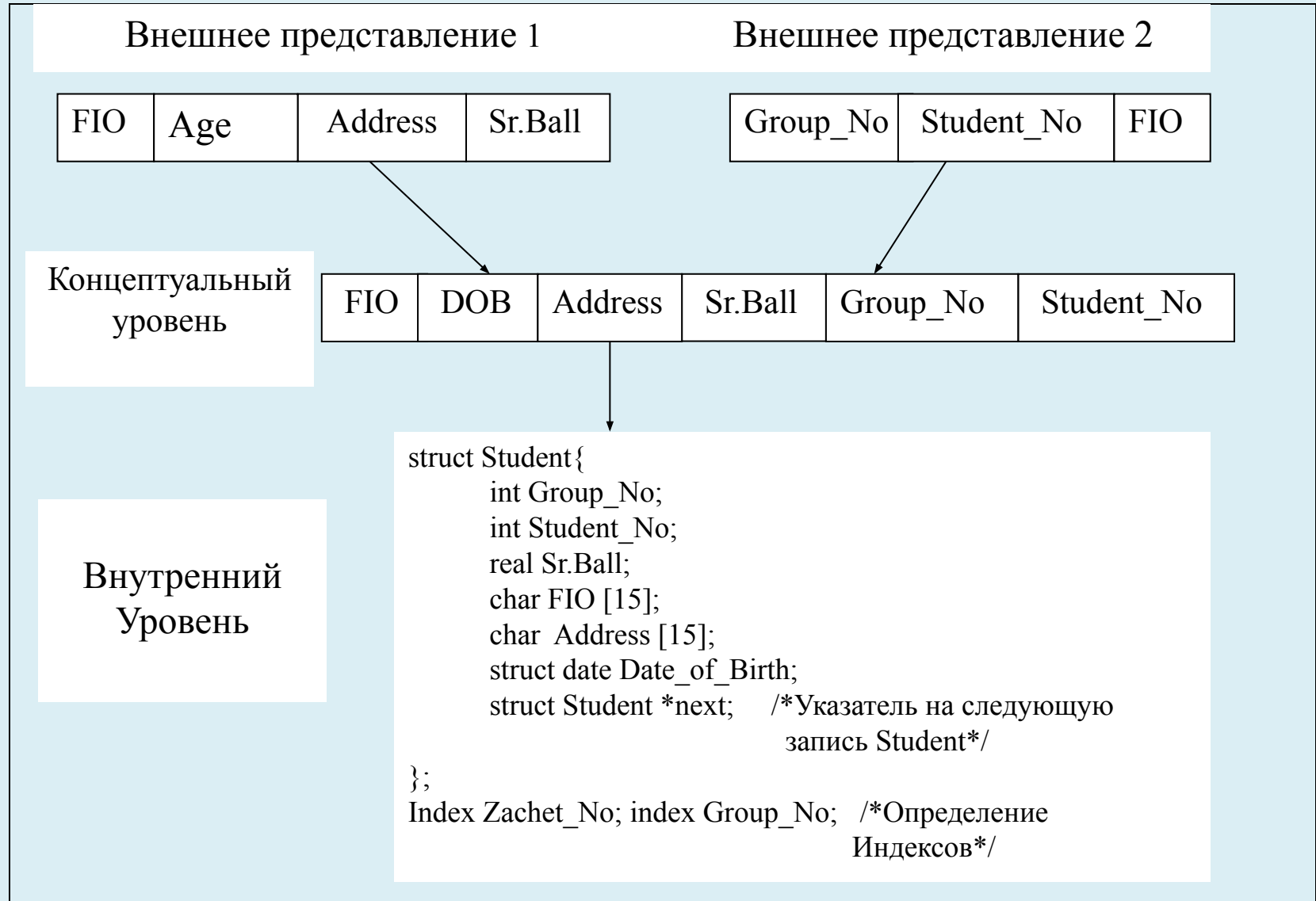
# Схемы БД

*Общее описание базы данных называется*  
**схемой базы данных.**

**Существуют три различных типа схем базы данных:**

1. На самом *высоком* уровне имеется несколько **внешних схем** или *подсхем*, которые соответствуют разным представлениям данных.
2. **Концептуальная схема** *описывает все элементы данных и связи между ними, с указанием необходимых ограничений поддержки целостности данных.*
3. **Внутренняя схема** - *полное описание внутренней модели данных, содержит определения хранимых записей, методы представления, описания полей данных, сведения об индексах и выбранных схемах кэширования.*

# Различия между тремя уровнями представления данных



# Различия между тремя уровнями представления данных

**Описанием базы данных** является *схема базы данных, которая создается в процессе ее проектирования, причем предполагается, что она изменяется достаточно редко.*

*Совокупность информации, хранящейся в базе данных в любой определенный момент времени, называется **состоянием базы данных.***

**Схема базы данных** иногда называется *содержанием базы данных, а ее состояние — детализацией.*

# Модели данных

**Модель данных** - это интегрированный набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные

**Модель данных рассматривается как сочетание трех компонентов:**

- *Структурная часть* - набор правил, по которым может быть построена база данных.
- *Управляющая часть*, определяющая типы допустимых операций с данными.
- *Набор ограничений поддержки целостности данных* (необязательно), гарантирующих корректность используемых данных.

# Модели данных

Для отображения в терминах архитектуры ANSI-SPARC идентифицируют следующие три связанные модели данных:

- *внешнюю модель данных*, отображающую представления каждого существующего в организации типа пользователей, которую иногда называют *предметной областью* (Universe of Discourse-UoD);
- *концептуальную модель данных*, отображающую логическое (или обобщенное) представление о данных, не зависящее от типа выбранной СУБД;
- *внутреннюю модель данных*, отображающую концептуальную схему определенным образом, понятным выбранной целевой СУБД.

# Модели данных

## 1. Объектные (object-based) модели данных:

- *Модель типа “сущность-связь” или ER-модель (Entity-Relationship model)*
- *Семантическая модель*
- *Функциональная модель*
- *Объектно-ориентированная модель*

## 2. Модели данных на основе записей (record-based):

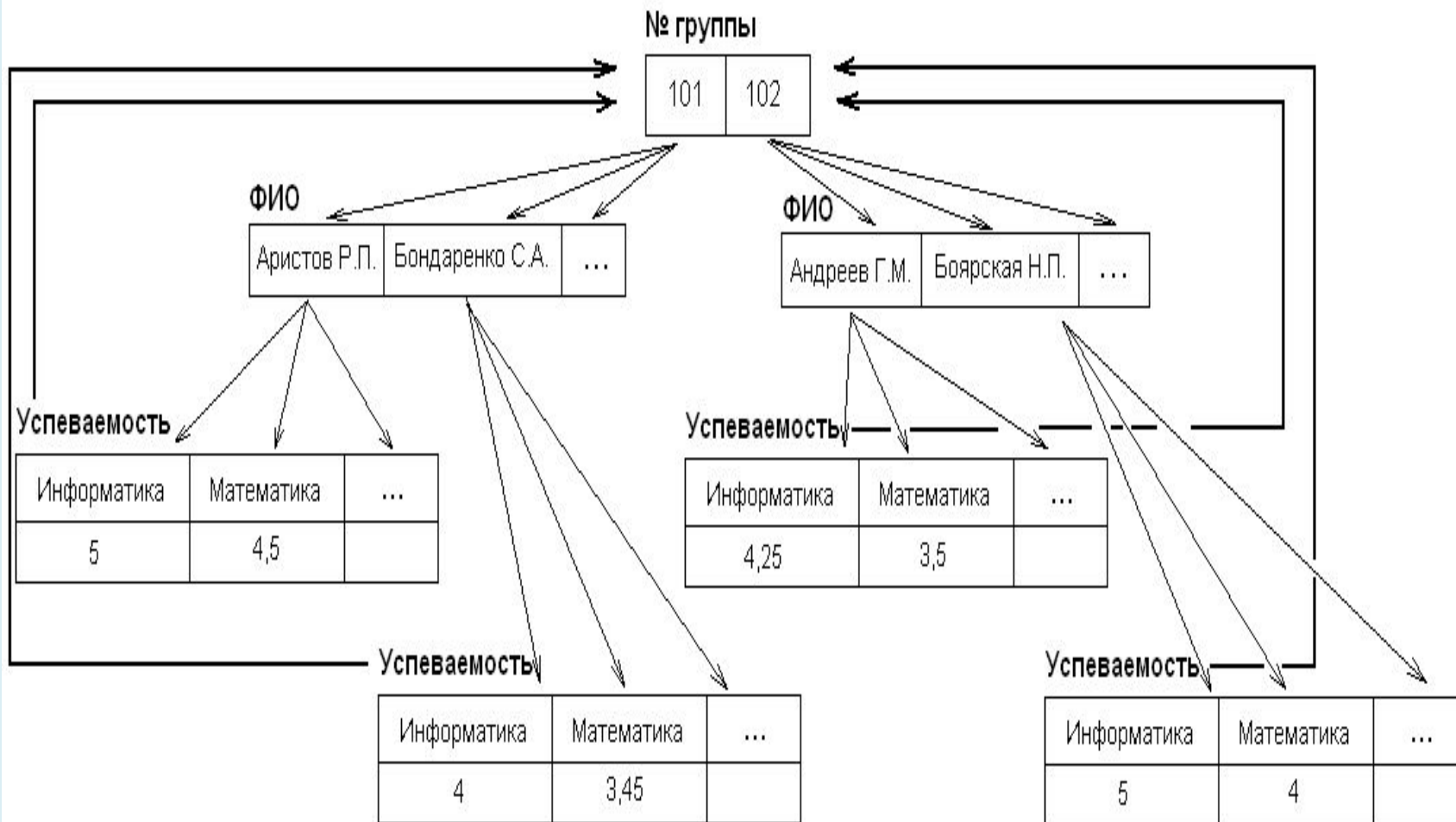
- *Реляционная модель данных (relational data model)*
- *Сетевая модель данных (network data model)*
- *Иерархическая модель данных (hierarchical data model)*

# Реляционная модель данных

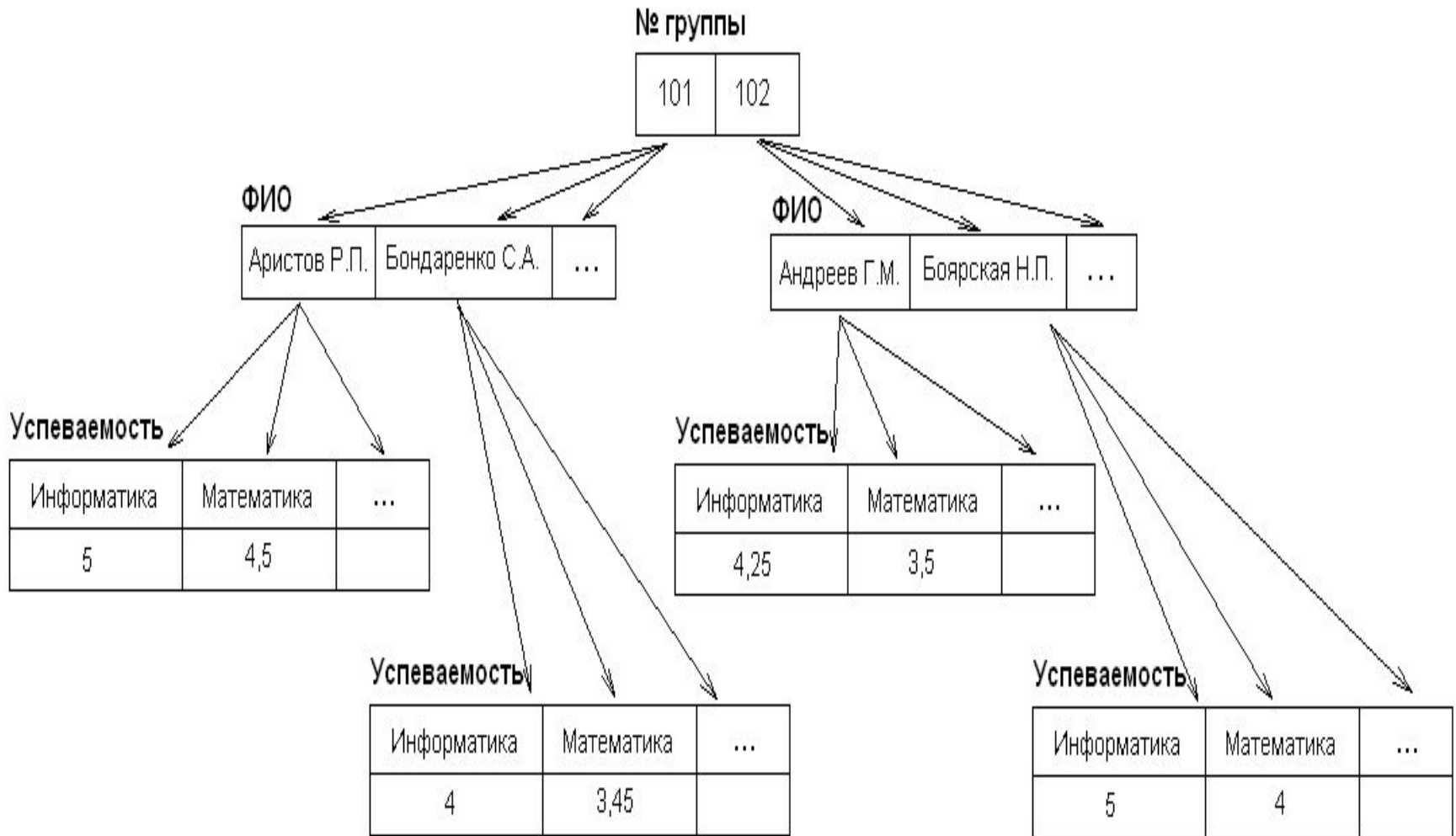
Group_No	Student_No	FIO	DOB	Sr.Ball
101	1	Аристов Р.П.	30.04.1986	4,5
101	2	Бондаренко С.А.	12.09.1987	4,5
101	3	Борисова Е.И.	08.06.1987	4,25
101	4	Макова Н.В.	01.06.1986	4,75
102	1	Боярская Н.П.	10.09.1987	4,5
102	2	Федоров Д.К.	15.03.1987	4,25
102	3	Сидоров И.Р.	29.07.1987	4,5
103	1	Андреев Г.М.	05.02.1986	4,25
103	2	Петров О.К.	25.08.1987	4,75
104	1	Иванов К.К.	03.03.1986	4,5
104	7	Григорьева Е.А.	16.10.1986	4,75
105	1	Серов И.Г.	27.04.1987	4,5
105	12	Антоненко К. П.	09.07.1986	4,5



# Сетевая модель данных



# Иерархическая модель данных



# Модели данных

## 3. Физические модели данных.

- *Обобщающая модель (unifying model)*
- *Модель памяти кадров (frame model)*

Первые две модели используются для описания данных на концептуальном и внешнем уровнях, а последняя — на внутреннем уровне.

**Физические модели данных** описывают то, как данные хранятся в компьютере, представляя информацию о структуре записей, их упорядоченности и существующих путях доступа

# Концептуальное моделирование

**Концептуальное моделирование, или концептуальное проектирование базы данных - это процесс конструирования модели использования информации на некотором предприятии, он не зависит от таких подробностей, как используемая СУБД, прикладные программы, языки программирования или любые другие вопросы физической организации информации.**

# Система управления Базами Данных

## Система управления базой данных (СУБД)

представляет собой *программное обеспечение*, которое управляет всем доступом к базе данных.

### Концептуально это происходит следующим образом:

1. Пользователь выдает запрос на доступ к данным, применяя определенный подъязык данных (обычно это язык SQL).
2. СУБД перехватывает этот запрос и анализирует его.
3. СУБД просматривает внешнюю схему для этого пользователя, соответствующее отображение «внешний-концептуальный», концептуальную схему, отображение «концептуальный-внутренний» и определения структур хранения.
4. СУБД выполняет необходимые операции в хранимой базе данных.

# Функции СУБД

- Хранение, извлечение и обновление данных
- Каталог, доступный конечным пользователям
- Поддержка транзакций
- Службы управления параллельной работой
- Службы восстановления
- Службы контроля доступа к данным
- Поддержка обмена данными
- Службы поддержки целостности данных
- Службы поддержки независимости от данных
- Вспомогательные службы

# Функции СУБД

## **1. Хранение, извлечение и обновление данных.**

*СУБД должна предоставлять пользователям возможность сохранять, извлекать и обновлять данные в базе данных.*

# Функции СУБД

## 2. Каталог доступный конечным пользователям

*СУБД должна иметь доступный конечным пользователям каталог, в котором хранится описание элементов данных.*

*В системном каталоге хранятся следующие сведения:*

- имена, типы и размеры элементов данных;*
- имена связей;*
- накладываемые на данные ограничения поддержки целостности;*
- имена санкционированных пользователей, которым предоставлено право доступа к данным;*
- внешняя, концептуальная и внутренняя схемы и отображения между ними;*
- статистические данные, например частота транзакций и счетчики обращений к объектам базы данных.*



# Функции СУБД

Системный каталог позволяет достичь *определенных преимуществ*, перечисленных ниже:

- Информация о данных может быть *централизованно собрана и сохранена*, что позволит *контролировать доступ к этим данным*, как и к любому другому ресурсу.
- Можно *определить смысл данных*, что поможет другим пользователям понять их предназначение.
- *Упрощается сообщение*, так как сохраняются точные определения смысла данных. В системном каталоге также могут быть указаны один или несколько пользователей, которые являются владельцами данных или обладают правом доступа к ним.
- Благодаря централизованному хранению *избыточность и противоречивость описания отдельных элементов данных могут быть*

# Функции СУБД

- Внесенные в базу данных *изменения* могут быть *запротоколированы*.
- *Последствия любых изменений* могут быть *определены еще до их внесения*, поскольку в системном каталоге зафиксированы все существующие элементы данных, установленные между ними связи, а также все их пользователи.
- *Меры обеспечения безопасности* могут быть *дополнительно усилены*.
- Появляются *новые возможности организации поддержки целостности данных*.
- Может выполняться *аудит сохраняемой информации*.

# Функции СУБД

## 3. Поддержка транзакций

**Транзакция** представляет собой набор действий, выполняемых отдельным пользователем или прикладной программой, с целью доступа или изменения содержимого базы данных.

СУБД должна иметь механизм, который гарантирует выполнение либо всех операций обновления данной транзакции, либо ни одной из них..

# Функции СУБД

**4. Сервисы управления параллельностью**  
СУБД имеет механизм, который *гарантирует корректное обновление базы данных при параллельном выполнении операций обновления многими пользователями.*

*Одна из основных целей создания и использования СУБД заключается в том, чтобы множество пользователей могло осуществлять параллельный доступ к совместно обрабатываемым данным.*

Параллельный доступ сравнительно просто организовать, если все пользователи выполняют только чтение данных, поскольку в этом случае они не могут помешать друг другу. Однако когда два или больше пользователей одновременно получают доступ к базе данных, конфликт с нежелательными последствиями легко может возникнуть, например, если хотя бы один из них попытается обновить данные.

# Функции СУБД

## 5. Сервисы восстановления

СУБД предоставляет *средства восстановления базы данных* на случай *какого-либо ее повреждения или разрушения*

## 6. Сервисы контроля доступа к данным

СУБД имеет *механизм, гарантирующий возможность доступа к базе данных только санкционированных пользователей.*

# Функции СУБД

## 7. Поддержка обмена данными

СУБД обладает *способностью к интеграции с коммуникационным программным обеспечением.*

Современная СУБД работает в сетевом режиме, *получая запросы в виде сообщений обмена данными и аналогичным образом отвечая на них.* Такие попытки передачи данных управляются *менеджером обмена данными.*

*Распределенная обработка достигается за счет реализации механизма поддержки представлений или подсхем*

# Функции СУБД

## 8. Вспомогательные службы

СУБД предоставляет *некоторый набор различных вспомогательных служб.*

*Вспомогательные утилиты* обычно предназначены для оказания помощи АБД в *эффективном администрировании базы данных.*

Минимальный перечень утилит:

- *Утилиты импортирования и экспортирования;*
- *Средства мониторинга;*
- *Программы статистического анализа;*
- *Инструменты реорганизации индексов;*
- *Инструменты сборки мусора и перераспределения памяти.*

# Языки баз данных

**Язык определения данных DDL** - *описательный язык, который позволяет АБД или пользователю описать и поименовать сущности, необходимые для работы некоторого приложения, а также связи, имеющиеся между различными сущностями*

На практике существует один общий язык DDL, который позволяет задавать спецификации, как минимум, для внешней и концептуальной схем.



# Языки баз данных

**Язык управления данными (DML)** - язык, содержащий набор операторов для поддержки основных операций манипулирования содержащимися в базе данными.

**Процедурные языки DML** - языки, которые позволяют сообщить системе о том, какие данные необходимы, и точно указать, как их можно извлечь.

**Непроцедурные языки DML** – языки, которые позволяют указать лишь то, какие данные требуются, но, не то, как их следует извлекать.

# Языки баз данных

## Типы языков четвертого поколения:

1. *Языки представления информации*, например языки запросов или генераторы отчетов;
2. *Специализированные языки*, например языки электронных таблиц и баз данных;
3. *Генераторы приложений*, которые при создании приложений обеспечивают определение, вставку, обновление или извлечение сведений из базы данных;
4. *Языки очень высокого уровня*, предназначенные для генераций кода приложений.

В качестве языков четвертого поколения можно указать SQL и QBE.

# Языки баз данных

## Другие типы 4GL-языков:

**Генератор форм** представляет собой интерактивный инструмент, предназначенный для быстрого создания шаблонов ввода и отображения данных в экранных формах, позволяет пользователю определить внешний вид экранной формы, ее содержимое и место расположения на экране.

**Генератор отчетов** является инструментом создания отчетов на основе хранимой в базе данных информации. Существует два основных типа генераторов отчетов: языковой и визуальный

**Генератор графического представления данных** - этот генератор представляет собой инструмент, предназначенный для извлечения информации из базы данных и отображения ее в виде диаграмм с графическим представлением существующих тенденций и связей.

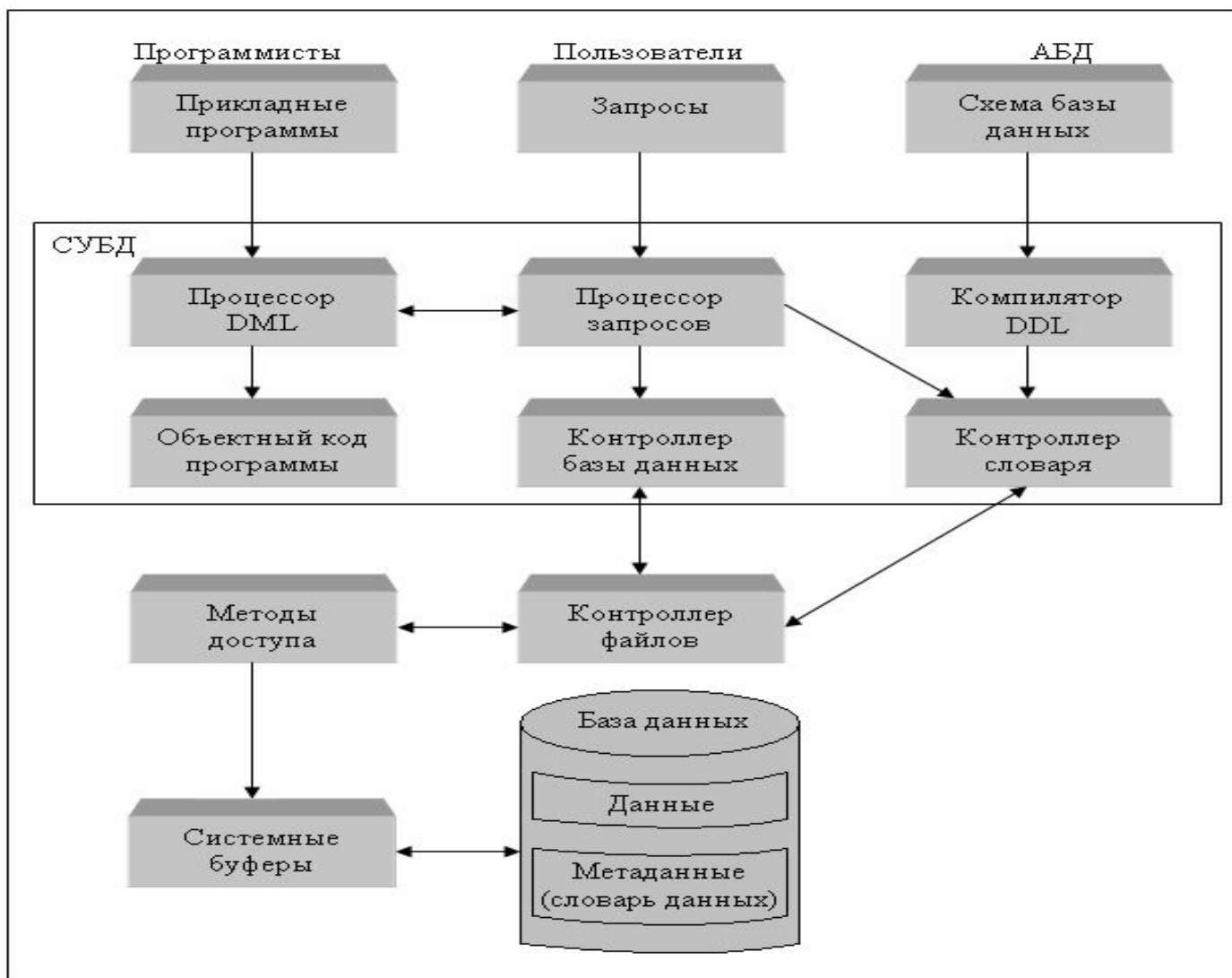
**Генераторы прикладных программ** - генератор прикладных программ представляет собой инструмент для создания программ, взаимодействующих с базой данных.

# Компоненты СУБД

СУБД состоит *из нескольких программных компонентов (модулей)*, каждый из которых предназначен для выполнения специфической операции.

- Процессор запросов
- Контроллер базы данных
- Контроллер файлов
- Препроцессор языка DML
- Компилятор языка DDL
- Контроллер словаря

# Основные компоненты типичной системы управления базами данных

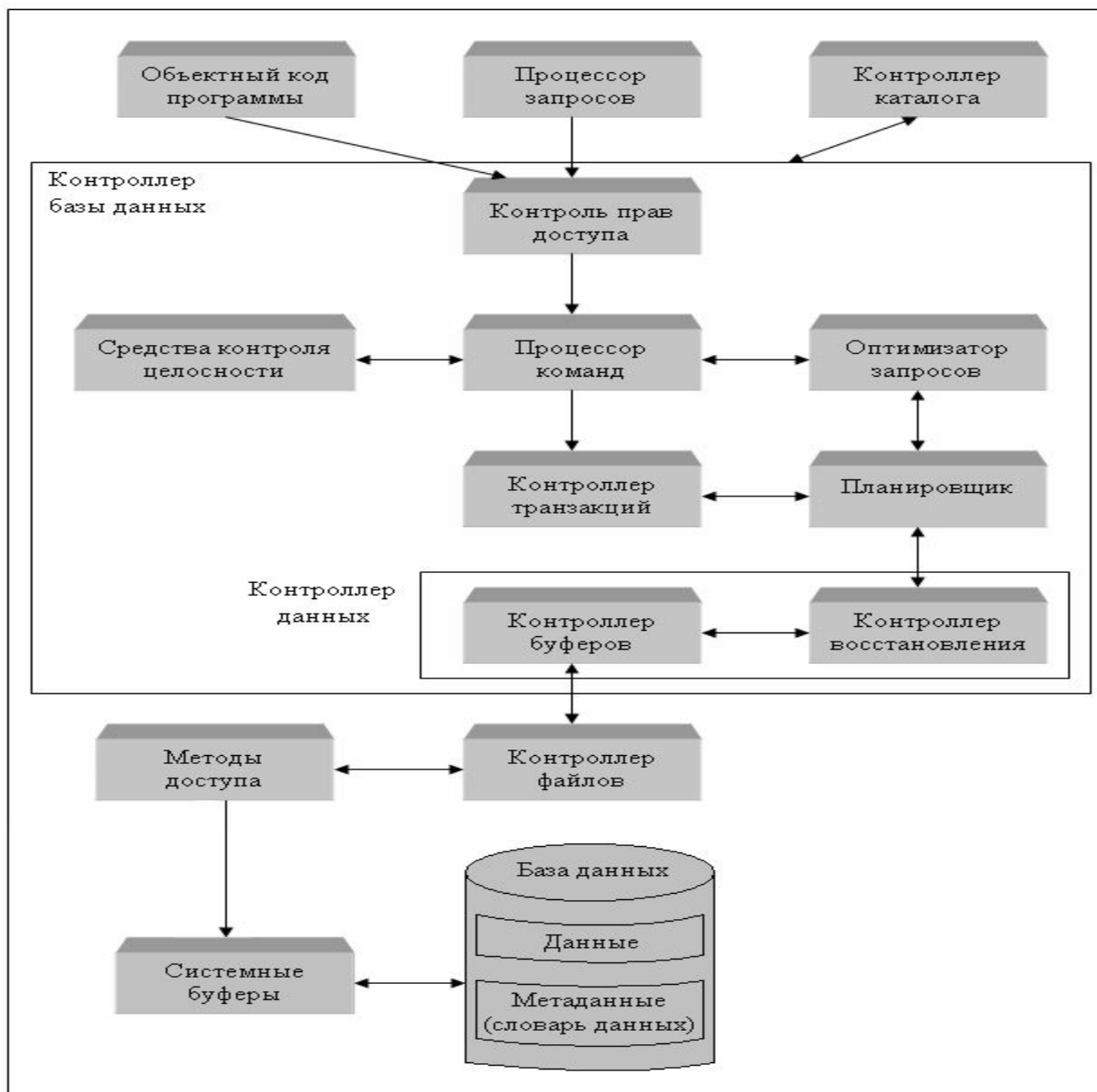


# Компоненты контроллера базы данных

Основные программные компоненты, входящие в состав контроллера базы данных:

- Контроль прав доступа
- Процессор команд
- Средства контроля целостности
- Оптимизатор запросов
- Контроллер транзакций
- Планировщик
- Контроллер восстановлений
- Контроллер буферов

# Компоненты контроллера базы данных



# Система управления передачей данных

**Запрос конечного пользователя** к базе данных передается от рабочей станции пользователя к некоторому интерактивному приложению, а от него – к СУБД в форме *коммуникационных сообщений*.

**Ответы пользователю** также передаются в форме подобных сообщений.

*Передача таких сообщений происходит под управлением еще одного программного компонента – диспетчера передачи данных.*



# Система управления передачей данных

Диспетчер передачи данных и СУБД должны иметь согласованную совместную работу, как равноправные компоненты программного продукта более высокого уровня, называемого **системой базы данных и передачи данных** (*DataBase/Data-Communication – DB/DC*).

# Архитектура многопользовательских СУБД

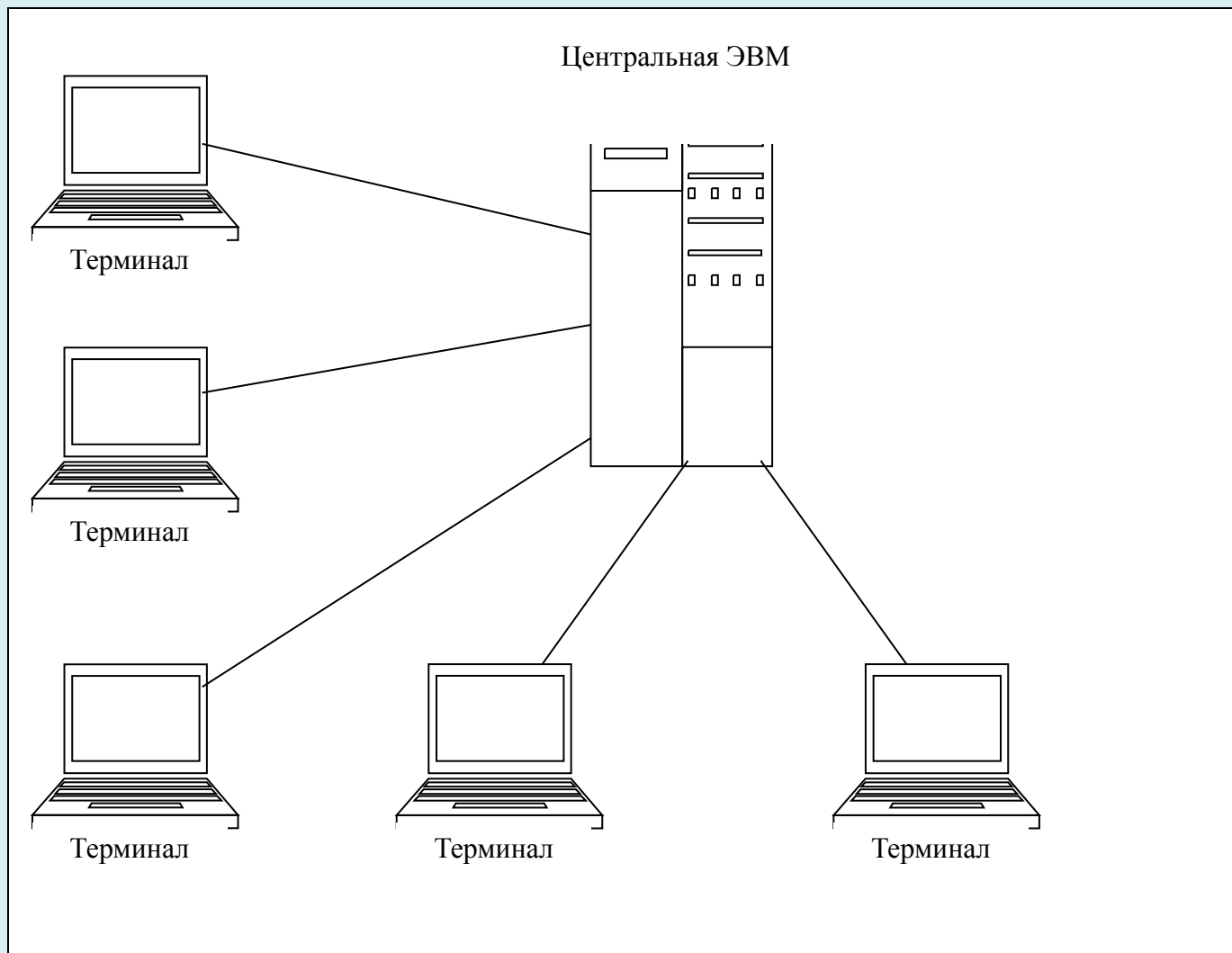
В настоящее время существуют такие *основные типовые архитектурные решения*, используемыми при реализации *многопользовательских СУБД*, а именно:

- Телеобработка
- Файловый сервер
- Технология "клиент-сервер".

# Телеобработка

**Телеобработка** – схема, при которой один компьютер с единственным процессором соединен с несколькими терминалами. При этом вся обработка выполняется с помощью этого компьютера.

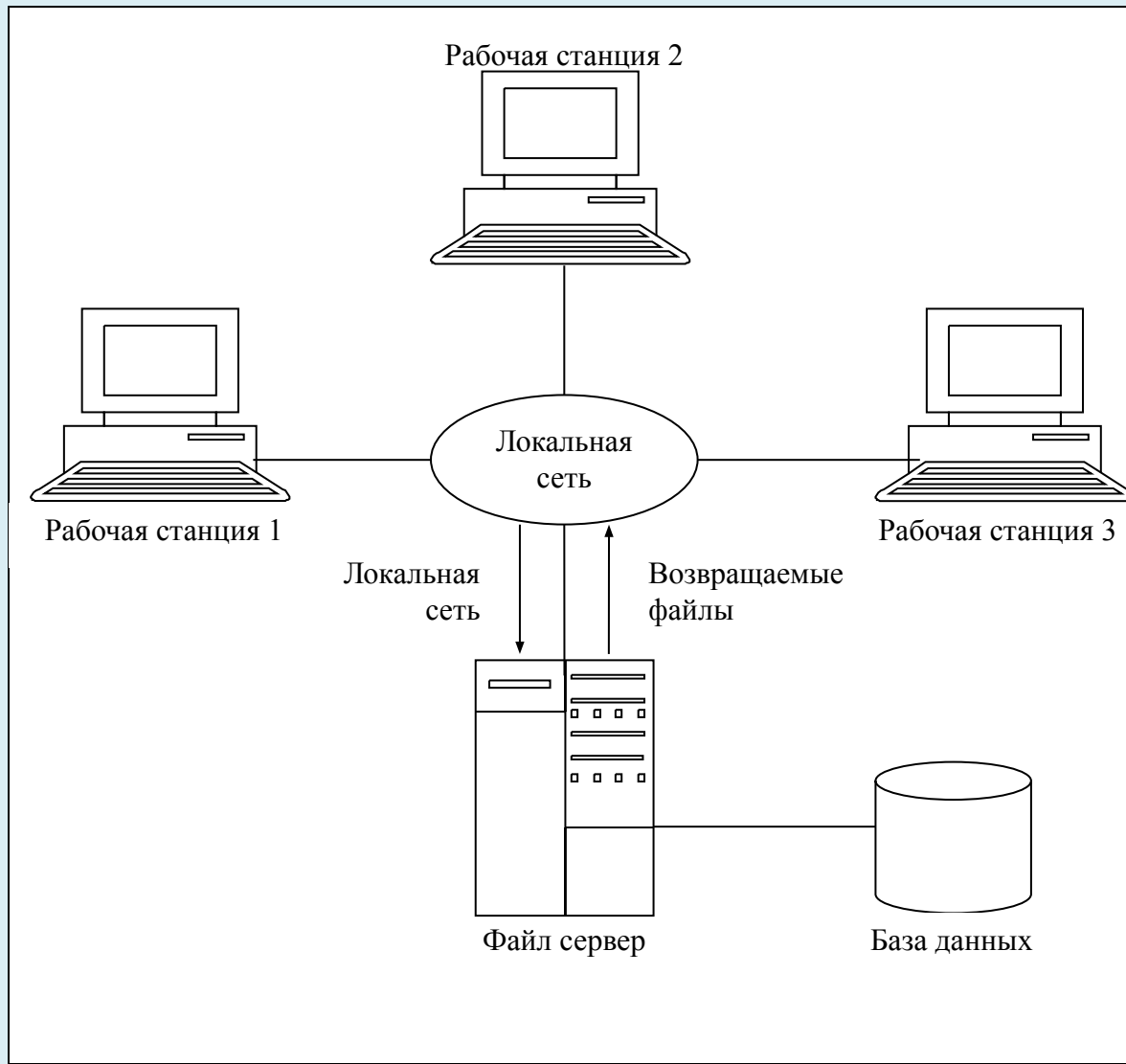
# Топология архитектуры телеобработки



# Файловый сервер

В среде файлового сервера обработка данных распределена в сети, обычно представляющей собой локальную вычислительную сеть (ЛВС). Файловый сервер содержит файлы, необходимые для работы приложений и самой СУБД. Эти приложения и СУБД размещены и функционируют на отдельных рабочих станциях.

# Архитектура с использованием файлового сервера



# Архитектура с использованием файлового сервера

Архитектура с использованием файлового сервера обладает следующими *основными недостатками*:

- *большой объем сетевого трафика,*
- *на каждой рабочей станции должна находиться полная копия СУБД,*
- *управление параллельностью, восстановлением и целостностью усложняется, поскольку доступ к одним и тем же файлам могут осуществлять сразу несколько экземпляров СУБД.*

# Архитектура клиент-сервер

***"Клиент/сервер" означает такой способ взаимодействия программных компонентов, при котором они образуют единую систему.***

***Существует некий клиентский процесс, требующий определенных ресурсов, а также серверный процесс, который эти ресурсы предоставляет.***



# Архитектура клиент-сервер

**Клиент** принимает от пользователя запрос, проверяет синтаксис и генерирует запрос к базе данных на языке SQL или другом языке базы данных, который соответствует логике приложения. Затем он передает сообщение серверу, ожидает поступления ответа и форматирует полученные данные для представления их пользователю.

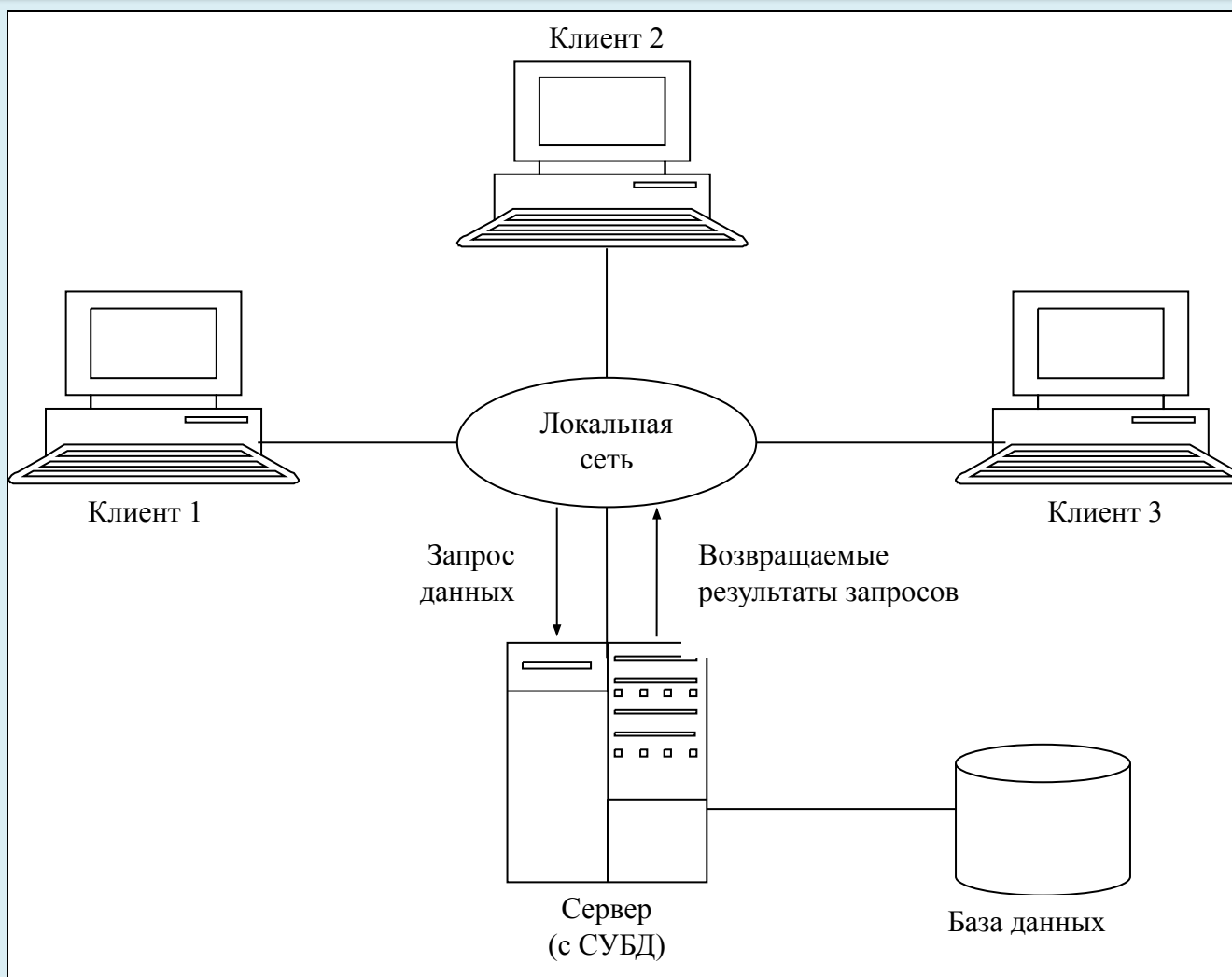
**Сервер** принимает и обрабатывает запросы к базе данных, а затем передает полученные результаты обратно клиенту. Такая обработка включает проверку полномочий клиента, обеспечение требований целостности, поддержку системного каталога, а также выполнение запроса и обновление данных, при этом поддерживается управление параллельностью и восстановлением.

# Архитектура клиент-сервер

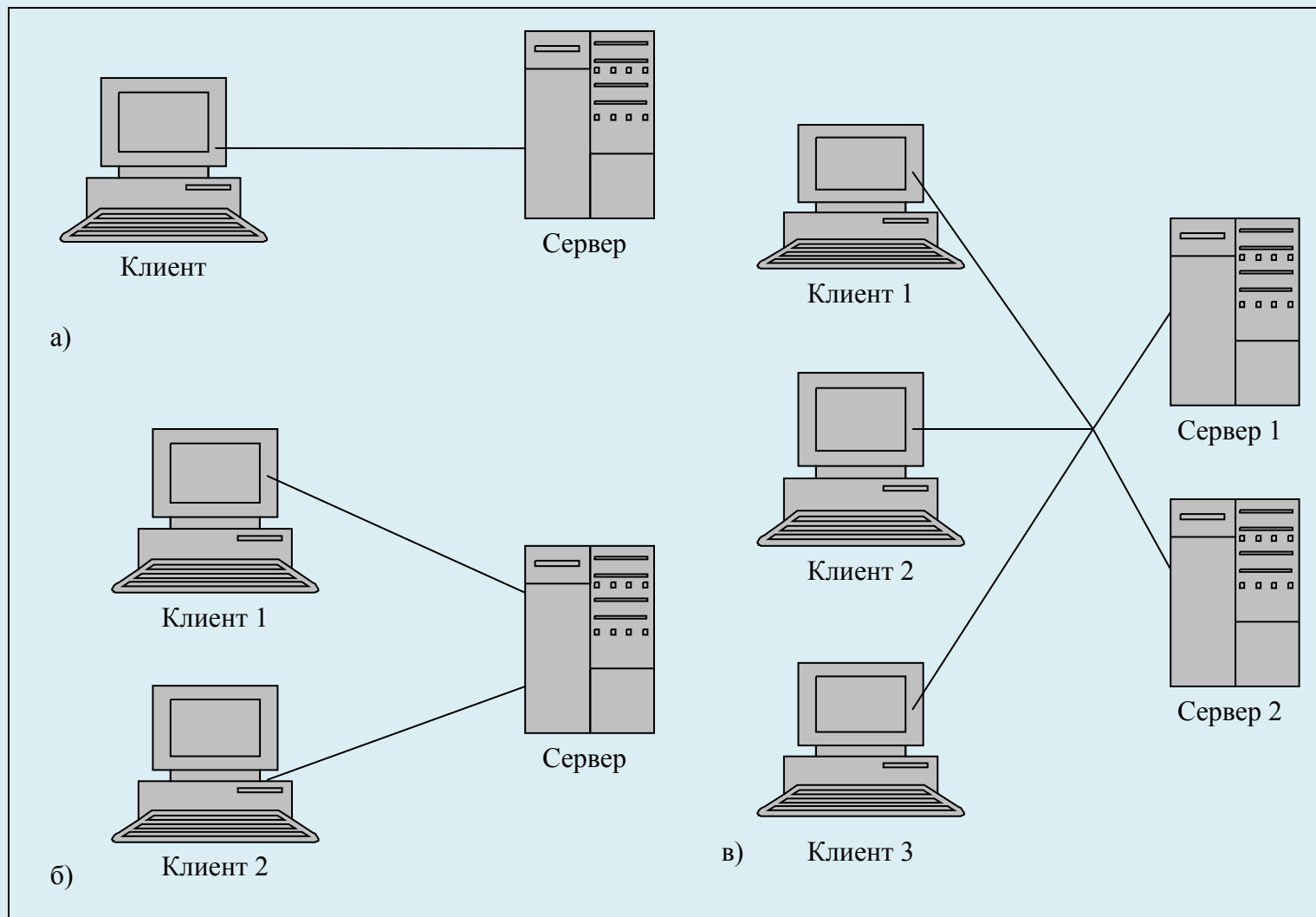
## Операции сервера

- Принимает и обрабатывает запросы к БД от клиента
- Проверяет полномочия пользователей
- Гарантирует соблюдение ограничений целостности
- Выполняет запросы обновления и возвращает результат клиенту
- Поддерживает системный каталог
- Обеспечивает параллельный доступ к БД

# Общая схема построения систем с архитектурой «клиент-сервер»



# Альтернативные топологии систем с архитектурой «клиент-сервер»



# Функции, выполняемые в среде “клиент/сервер”

Клиент	Сервер
Управляет пользовательским интерфейсом	Принимает и обрабатывает запросы к базе данных со стороны клиентов
Принимает и проверяет синтаксис введенного пользователем запроса	Проверяет полномочия пользователей
Выполняет приложение	Гарантирует соблюдение ограничений целостности
Генерирует запрос к базе данных и передает его серверу	Выполняет запросы/обновления и возвращает результаты клиенту
Отображает полученные данные пользователю	Поддерживает системный каталог Обеспечивает параллельный доступ к БД Обеспечивает управление восстановлением

# Системные каталоги

**Системный каталог** – это хранилище данных, которые описывают сохраняемую в базе данных информацию, т.е. метаданные, или "данные о данных".

## Система словаря данных может быть:

**Активной** - система всегда согласуется со структурой базы данных, поскольку она автоматически поддерживается этой системой.

**Пассивной** - система может противоречить состоянию базы данных из-за инициируемых пользователями изменений.

Если словарь данных является частью базы данных, то он называется **интегрированным словарем данных**.

# Служба IRDS, как стандарт словарей данных

**Служба IRDS** представляет собой *программный инструмент, предназначенный для управления информационными ресурсами организации, а также для их документирования.*

Служба IRDS включает:

- *определение таблиц, содержащих словарь данных,*
- *операций, которые могут быть использованы для доступа к этим таблицам.*

# Служба IRDS, как стандарт словарей данных

**Стандарты IRDS** определяют набор правил хранения информации в словаре данных и доступа к ней, преследуя при этом три следующие цели:

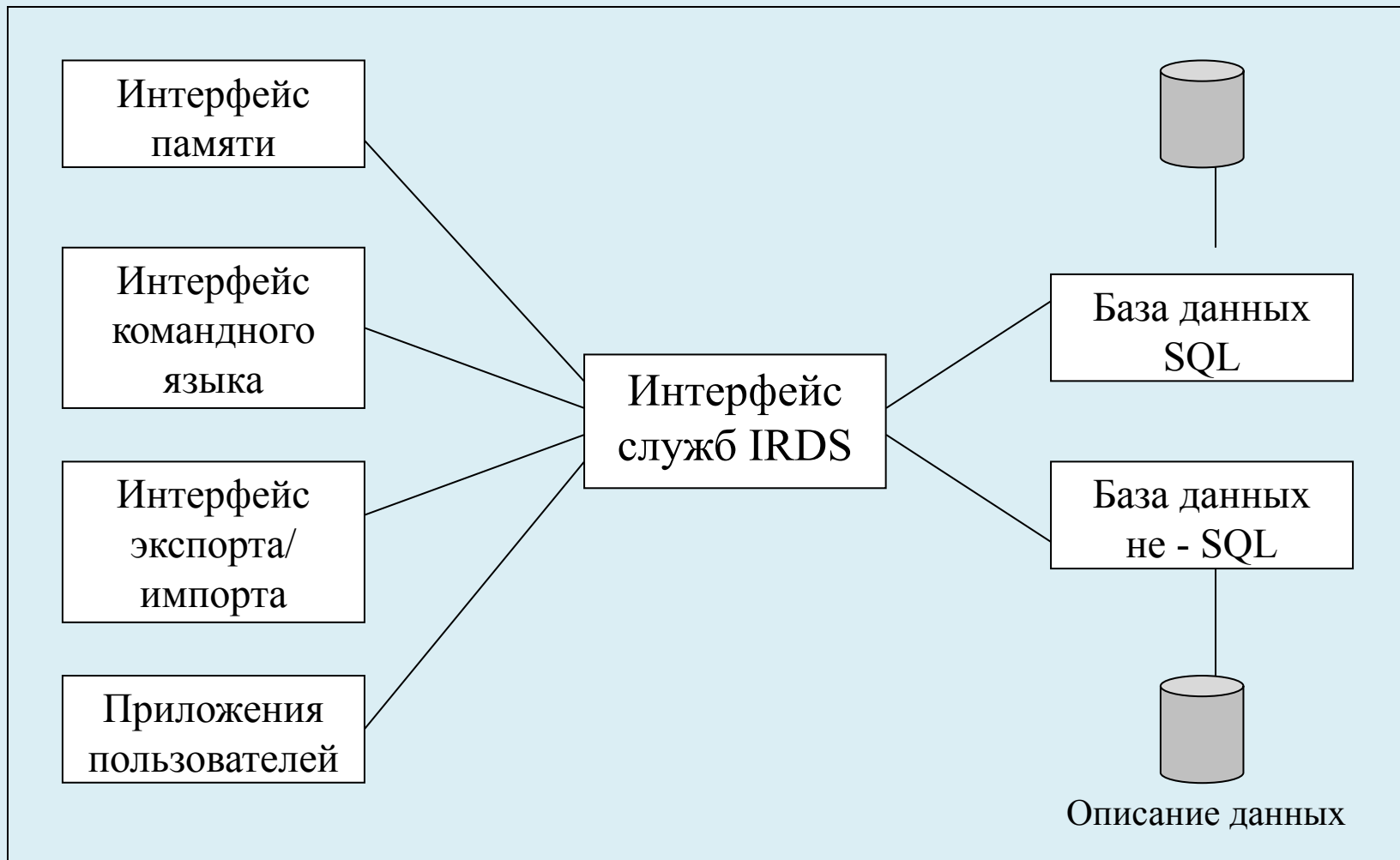
- *расширяемость данных;*
- *целостность данных;*
- *контролируемый доступ к данным.*

**Интерфейс сервисов** может вызываться со стороны таких типов пользовательских интерфейсов, как:

- *панельный интерфейс;*
- *файлы экспорта/импорта;*
- *командный язык;*
- *прикладные программы*



# Интерфейс IRDS - сервисов



# УТИЛИТЫ

*Утилиты – это программы, разработанные для администратора баз данных и используемые им при решении различных административных задач.*

## Примеры утилит различных типов:

- Инструменты загрузки
- Инструменты выгрузки-перезагрузки
- Инструменты реорганизации
- Статистические инструменты
- Инструменты анализа