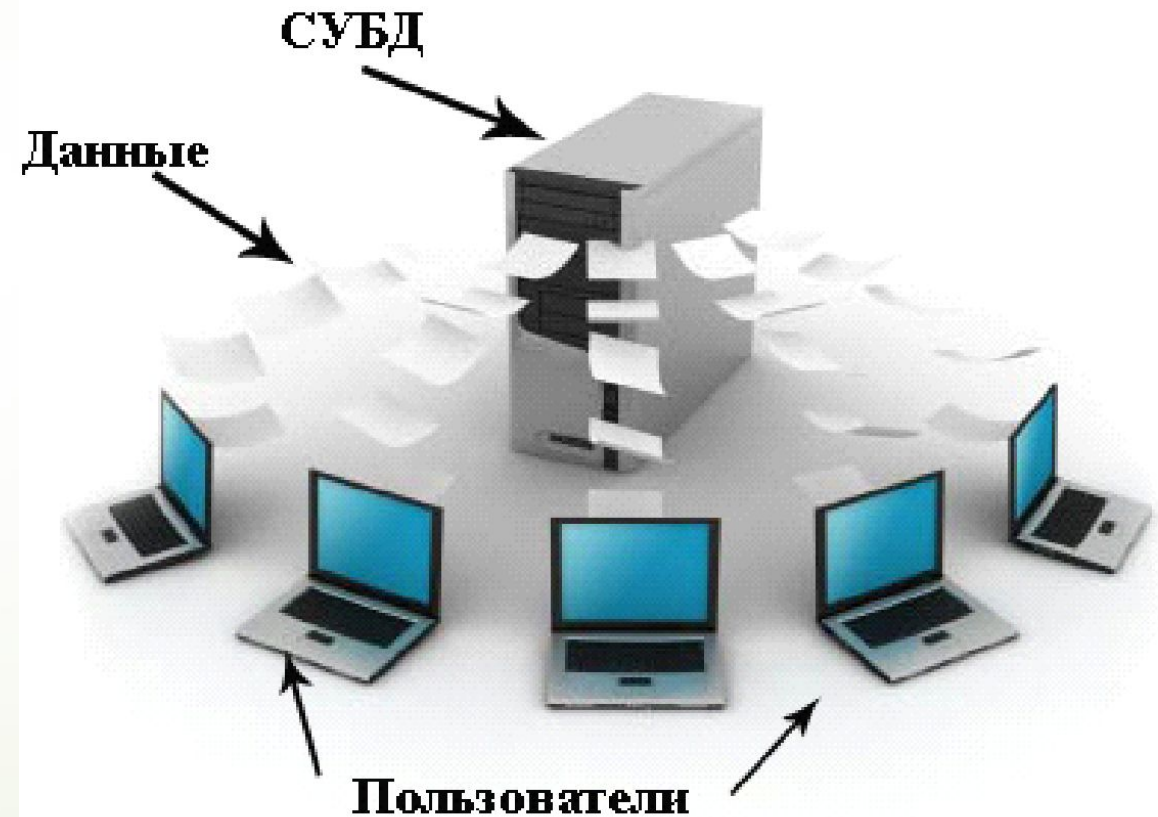


Лекция №1 Введение

1

- 1 Основные принципы, заложенные в методологию баз данных
- 2 История возникновения и этапы развития баз данных
- 3 Архитектуры СУБД
- 4 Краткий обзор СУБД
- 5 Классификация баз данных
- 6 Трехуровневая архитектура ANSI/SPARC



1 Основные принципы, заложенные в методологию БД

Информационная система (ИС) – система, реализующая автоматизированный сбор, обработку данных и манипулирование данными

Цель любой информационной системы – обработка данных об объектах реального мира

Современной формой организации ИС является **банк данных**.

Банк данных (Бнд) — это система специальным образом организованных данных — баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.

База данных (Бд) — именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Система управления базами данных (СУБд) — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования Бд многими пользователями.

Программы, с помощью которых пользователи работают с базой данных, называются **приложениями**.

2 История возникновения и этапы развития баз данных

3

Первый этап развития СУБД связан с организацией баз данных на больших машинах типа IBM 360/370, ЕС-ЭВМ и мини-ЭВМ типа PDP11 (фирмы Digital Equipment Corporation — DEC), разных моделях HP (фирмы Hewlett Packard).

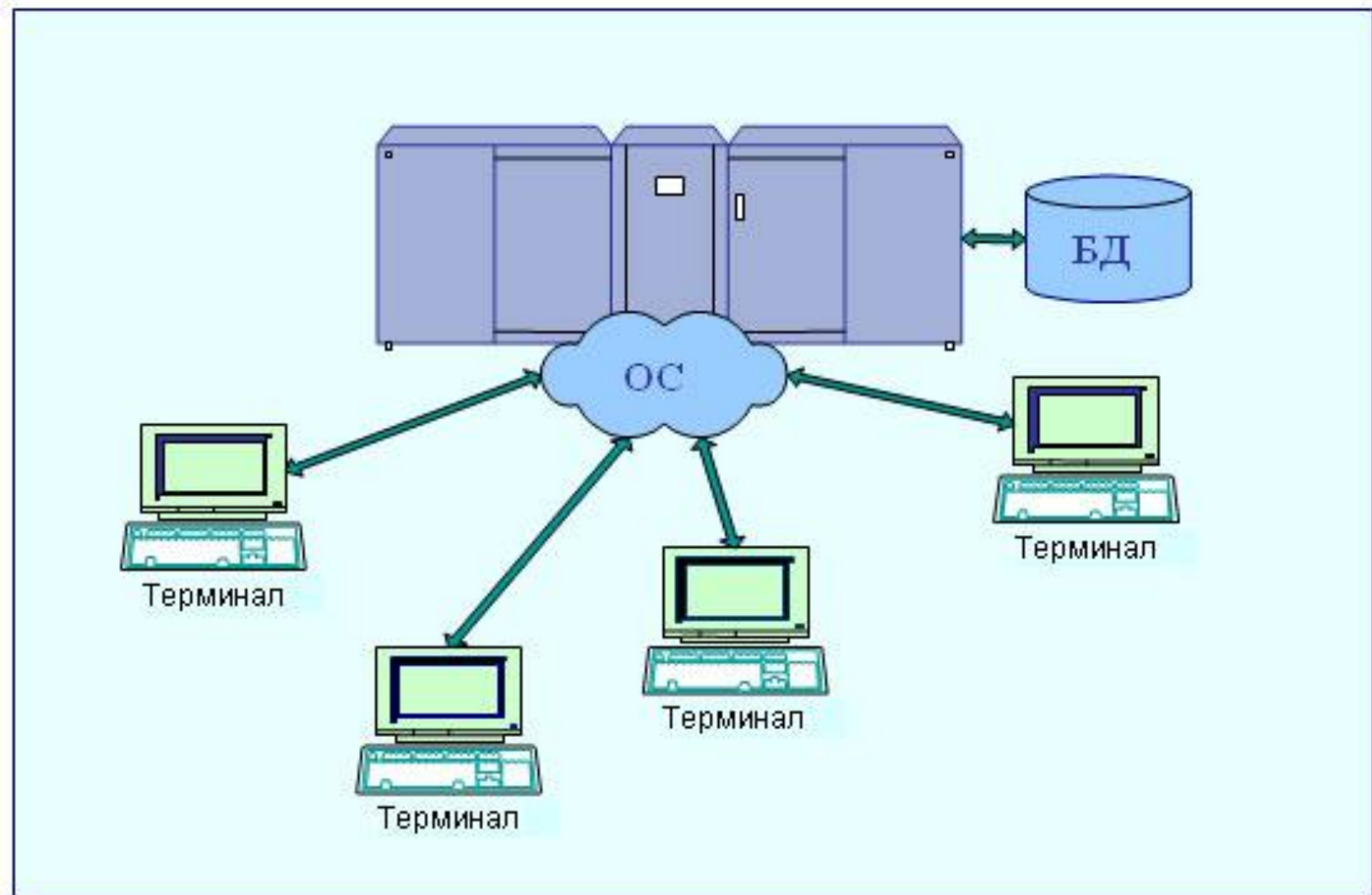


Рис. 1.1. Мэйнфреймовая архитектура

Второй этап- появление и развитие ПК

Особенности этого этапа состоят в следующем:

- Все СУБД были рассчитаны на создание БД в основном с монопольным доступом.
 - Большинство СУБД имели развитый и удобный пользовательский интерфейс и удобный инструментарий для разработки готовых приложений без программирования. Инструментальная среда состояла из готовых элементов приложения в виде шаблонов экранных форм, отчетов, этикеток (Labels), графических конструкторов запросов, которые достаточно просто могли быть собраны в единый комплекс.
 - При наличии высокоуровневых языков манипулирования данными, вроде реляционной алгебры и SQL, в настольных СУБД поддерживались низкоуровневые языки манипулирования данными на уровне отдельных строк таблиц.
 - В настольных СУБД отсутствовали средства поддержки ссылочной и структурной целостности базы данных. Эти функции должны были выполнять приложения, однако скудость средств разработки приложений иногда не позволяла это сделать, и эти функции должны были выполняться пользователем, требуя от него дополнительного контроля при вводе и изменении информации, хранящейся в БД.
 - Наличие монопольного режима работы фактически привело к вырождению функций администрирования БД.
 - В настоящий момент весьма положительная особенность — это сравнительно скромные требования к аппаратному обеспечению со стороны настольных СУБД. Вполне работоспособные приложения, разработанные, например, на Clipper, работали на PC 286.
- Яркие представители этого семейства — очень широко использовавшиеся до недавнего времени СУБД Dbase (DbaseIII+, DbaseIV), FoxPro, Clipper, Paradox.

Третий этап - появления сетей

5

Особенности данного этапа состоят в следующем:

- Практически все современные СУБД обеспечивают поддержку полной реляционной модели, а именно:
 - структурной целостности — допустимыми являются только данные, представленные в виде отношений реляционной модели;
 - языковой целостности, т. е. языков манипулирования данными высокого уровня (в основном SQL);
 - ссылочной целостности в течение всего времени функционирования системы.

4 этап-новая технология доступа к данным — Интранет

6

Основное отличие этого подхода от технологии «клиент-сервер» состоит в том, что отпадает необходимость использования специализированного клиентского программного обеспечения. Для работы с удаленной базой данных используется стандартный браузер Интернета, например Microsoft Internet Explorer (рис. 1.2).

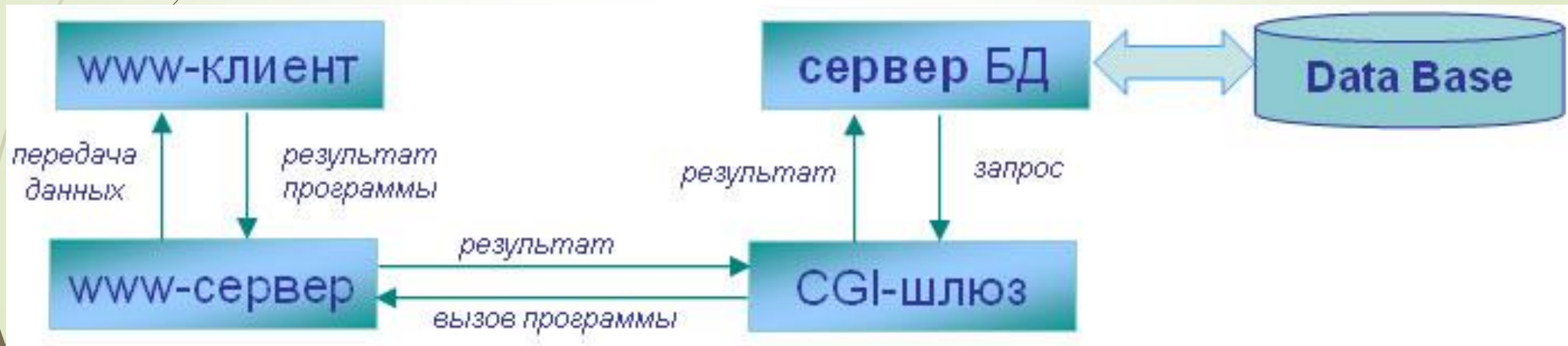


Рис. 1.2. Доступ к базам данных в архитектуре Интернет

2 Виды архитектур СУБД Централизованная архитектура

7

База данных в виде набора файлов находится на жестком диске компьютера.

На том же компьютере установлены СУБД и приложение для работы с БД

Пользователь запускает приложение. Используя предоставляемый приложением пользовательский интерфейс, он инициирует обращение к БД на выборку/обновление информации.

Все обращения к БД идут через СУБД, которая инкапсулирует внутри себя все сведения о физической структуре БД.

СУБД инициирует обращения к данным, обеспечивая выполнение запросов пользователя (осуществляя необходимые операции над данными).

Результат СУБД возвращает в приложение.

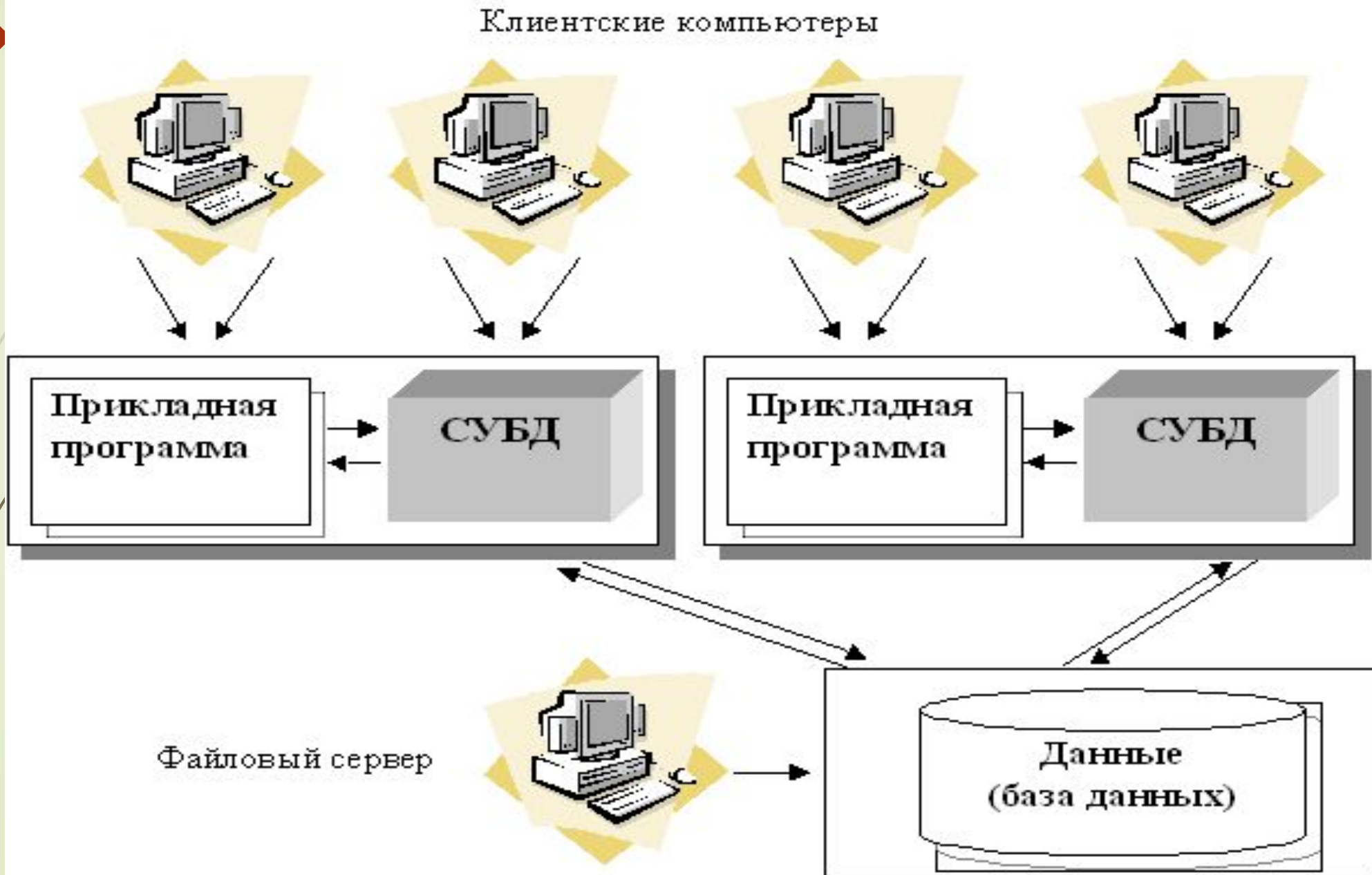
Приложение, используя пользовательский интерфейс, отображает результат



Подобная архитектура использовалась в первых версиях СУБД DB2, Oracle, Ingres.

2 Виды архитектур Архитектура "файл-сервер"

8



2 Виды архитектур СУБД Архитектура "файл-сервер"

9

Работа построена следующим образом:

- База данных в виде набора файлов находится на жестком диске специально выделенного компьютера (файлового сервера).
- Существует локальная сеть, состоящая из клиентских компьютеров, на каждом из которых установлены СУБД и приложение для работы с БД.
- На каждом из клиентских компьютеров пользователи имеют возможность запустить приложение. Используя предоставляемый приложением пользовательский интерфейс, он инициирует обращение к БД на выборку/обновление информации.
- Все обращения к БД идут через СУБД, которая инкапсулирует внутри себя все сведения о физической структуре БД, расположенной на файловом сервере.
- СУБД инициирует обращения к данным, находящимся на файловом сервере, в результате которых часть файлов БД копируется на клиентский компьютер и обрабатывается, что обеспечивает выполнение запросов пользователя (осуществляются необходимые операции над данными).
- При необходимости (в случае изменения данных) данные отправляются назад на файловый сервер с целью обновления БД.
- Результат СУБД возвращает в приложение.
- Приложение, используя пользовательский интерфейс, отображает результат выполнения запросов.

2 Виды архитектур СУБД Архитектура "файл-сервер"

10

В литературе указываются следующие основные недостатки данной архитектуры:

- При одновременном обращении множества пользователей к одним и тем же данным производительность работы резко падает, т.к. необходимо дождаться пока пользователь, работающий с данными, завершит свою работу. В противном случае возможно затирание исправлений, сделанных одними пользователями, изменениями других пользователей.
- Вся тяжесть вычислительной нагрузки при доступе к БД ложится на приложение клиента, так как при выдаче запроса на выборку информации из таблицы вся таблица БД копируется на клиентскую машину и выборка осуществляется на клиенте. Таким образом, неоптимально расходуются ресурсы клиентского компьютера и сети. В результате возрастает сетевой трафик и увеличиваются требования к аппаратным мощностям пользовательского компьютера.
- В БД на файл-сервере гораздо проще вносить изменения в отдельные таблицы, минуя приложения, непосредственно из инструментальных средств (например, из утилиты Database Desktop фирмы Borland для файлов Paradox и dBase); Все это позволяет говорить о низком уровне безопасности – как с точки зрения хищения и нанесения вреда, так и с точки зрения внесения ошибочных изменений.
- Недостаточно развитый аппарат транзакций служит потенциальным источником ошибок в плане нарушения смысловой и ссылочной целостности информации при одновременном внесении изменений в одну и ту же запись.

2 Виды архитектур СУБД Технология "клиент – сервер"

11

Архитектура "клиент – сервер" разделяет функции приложения пользователя (называемого клиентом) и сервера.

Приложение-клиент формирует запрос к серверу, на котором расположена БД, на структурном языке запросов SQL (Structured Query Language), являющемся промышленным стандартом в мире реляционных БД.

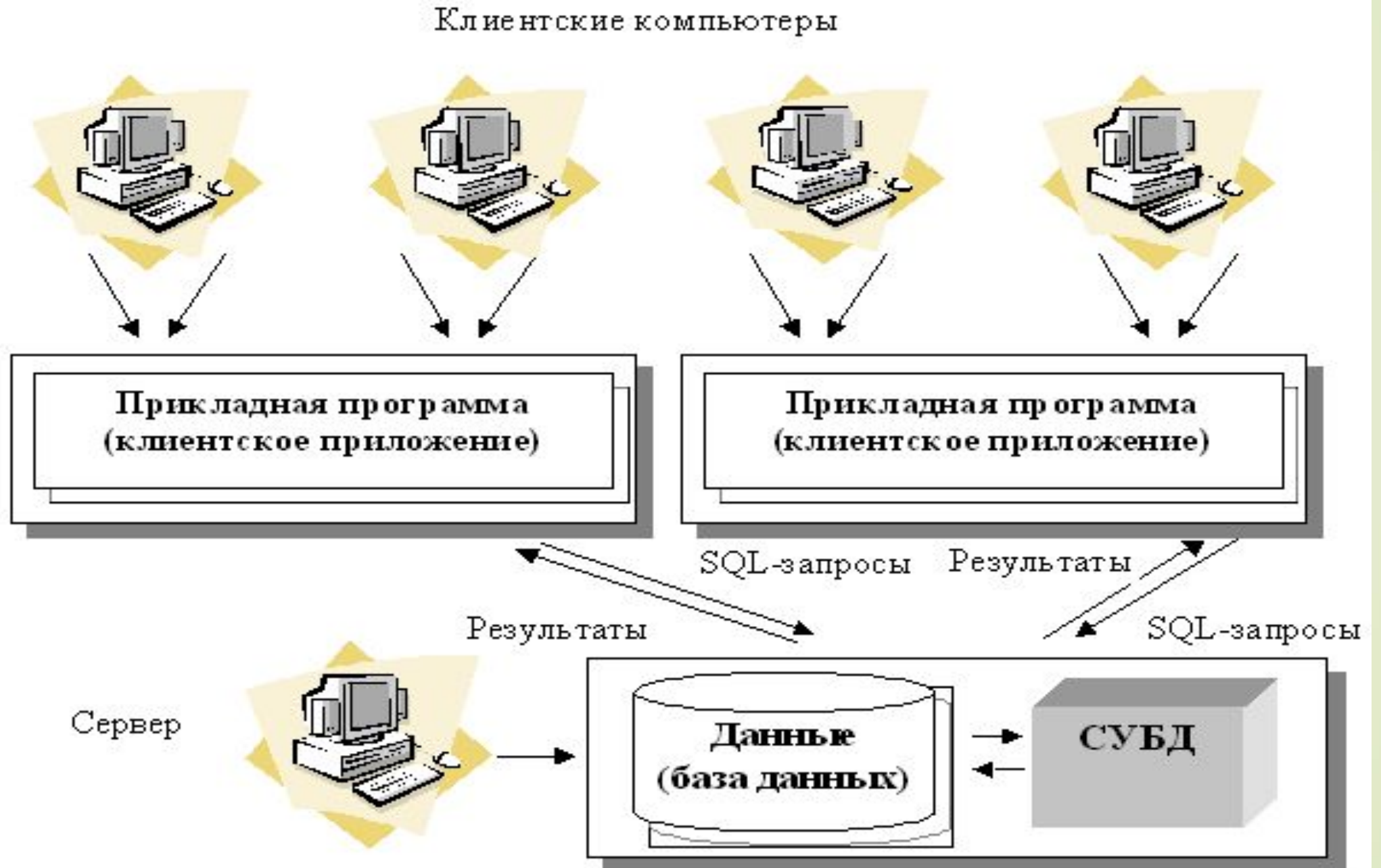
Удаленный сервер принимает запрос и переадресует его SQL-серверу БД.

SQL-сервер – специальная программа, управляющая удаленной базой данных. SQL-сервер обеспечивает интерпретацию запроса, его выполнение в базе данных, формирование результата выполнения запроса и выдачу его приложению-клиенту. При этом ресурсы клиентского компьютера не участвуют в физическом выполнении запроса; клиентский компьютер лишь отправляет запрос к серверной БД и получает результат, после чего интерпретирует его необходимым образом и представляет пользователю.

Так как клиентскому приложению посылается результат выполнения запроса, по сети "путешествуют" только те данные, которые необходимы клиенту. В итоге снижается нагрузка на сеть. Архитектура системы представлена на [рис. 1.5](#).

2 Виды архитектур СУБД Технология "клиент – сервер"

12



2 Виды архитектур СУБД Технология "клиент – сервер"

13

Функции приложения-клиента:

- Посылка запросов серверу.
- Интерпретация результатов запросов, полученных от сервера.
- Представление результатов пользователю в некоторой форме (интерфейс пользователя).

Функции серверной части:

- Прием запросов от приложений-клиентов.
- Интерпретация запросов.
- Оптимизация и выполнение запросов к БД.
- Отправка результатов приложению-клиенту.
- Обеспечение системы безопасности и разграничение доступа.
- Управление целостностью БД.
- Реализация стабильности многопользовательского режима работы.

2 Виды архитектур СУБД Технология "клиент – сервер"

14

Рассмотрим основные достоинства данной архитектуры по сравнению с архитектурой "файл-сервер":

- Существенно уменьшается сетевой трафик.
- Уменьшается сложность клиентских приложений (большая часть нагрузки ложится на серверную часть), а, следовательно, снижаются требования к аппаратным мощностям клиентских компьютеров.
- Наличие специального программного средства – SQL-сервера – приводит к тому, что существенная часть проектных и программистских задач становится уже решенной.
- Существенно повышается целостность и безопасность БД.

К числу недостатков можно отнести более высокие финансовые затраты на аппаратное и программное обеспечение, а также то, что большое количество клиентских компьютеров, расположенных в разных местах, вызывает определенные трудности со своевременным обновлением клиентских приложений на всех компьютерах-клиентах. Тем не менее, архитектура "клиент – сервер" хорошо зарекомендовала себя на практике, в настоящий момент существует и функционирует большое количество БД, построенных в соответствии с данной архитектурой.

3 Краткий обзор современных СУБД. Файл-серверные СУБД

15

Способы доступа к БД:

- **Файл-серверные СУБД**
- **Клиент-серверные СУБД**
- **Встраиваемые СУБД**

Настольные и файл-серверные СУБД используются для сравнительно небольших задач (небольшой объем обрабатываемых данных, малое количество пользователей).

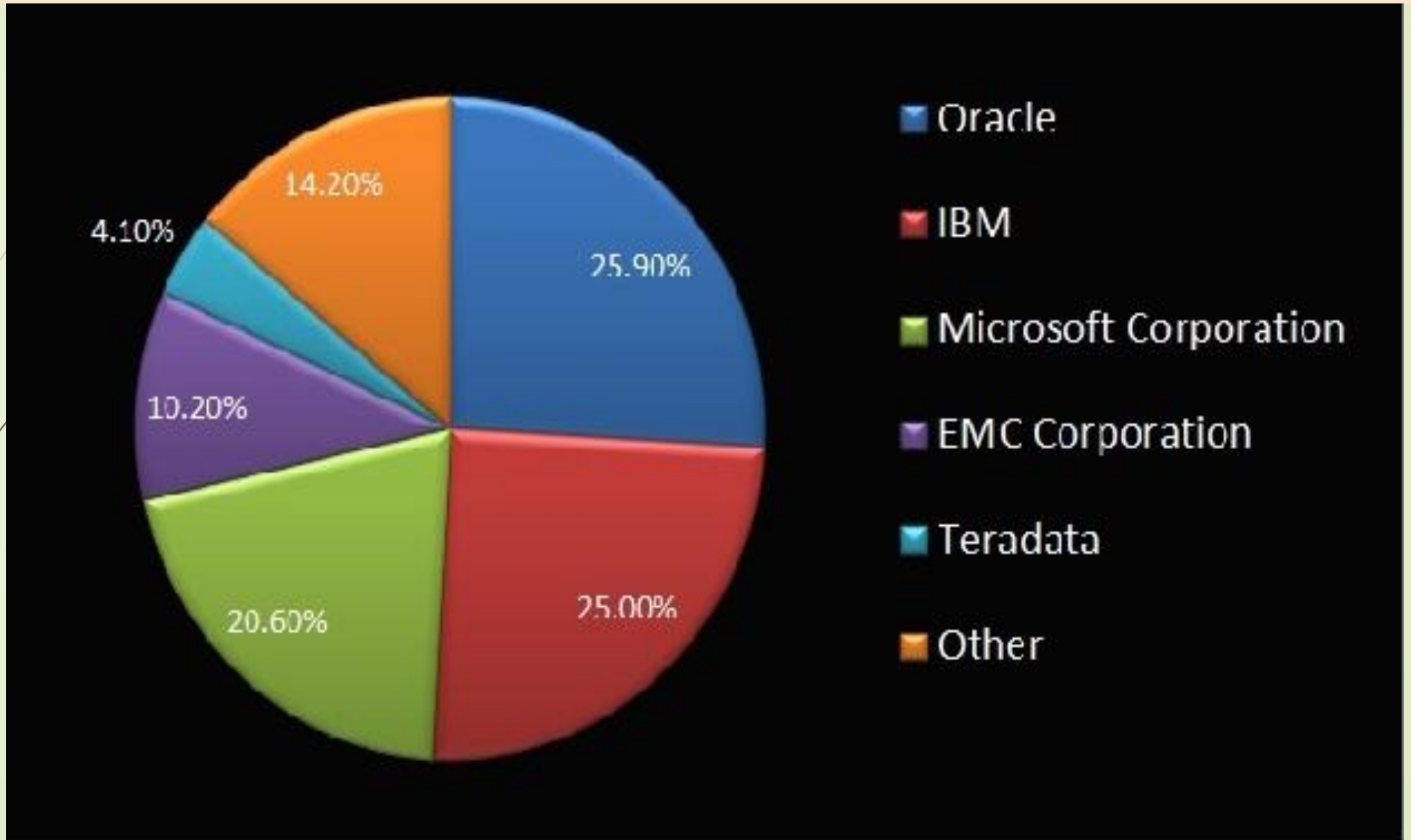
На данный момент файл-серверная технология считается устаревшей, а её использование в крупных информационных системах — недостатком.

Тем не менее, такие системы имеют достаточно обширную область применения там, где объемы данных не являются катастрофически большими, частота обновлений не бывает слишком высокой, организация территориально обычно расположена в одном небольшом здании, количество пользователей колеблется от одного до 10–15 человек. В подобных условиях использование настольных СУБД для управления информационными системами является вполне оправданным

В первую очередь широкое распространение получила система управления

3 Краткий обзор современных СУБД. Клиент-серверные СУБД

16



3 Краткий обзор современных СУБД. Клиент-серверные СУБД

17

В клиент-серверных СУБД (Microsoft SQL Server, Oracle, Firebird, PostgreSQL, InterBase, MySQL и др.) вся обработка данных ведётся на сервере.

Приложения-клиенты при этом посылают запросы на обработку и получение данных из СУБД и получают ответы; приложения-клиенты не имеют непосредственного доступа к файлам данных. Все промышленные СУБД на данный момент являются именно клиент-серверными.

Встраиваемые СУБД (SQLite, Firebird Embedded, Microsoft SQL Server Compact и др.) поставляются в составе готового программного продукта, не требуя процедуры самостоятельной установки. Встраиваемые СУБД предназначены для локального хранения данных приложения и не рассчитаны на коллективное использование в сети. К примеру, встраиваемая бесплатная СУБД SQLite широко используется в известной мобильной ОС Android, разработанной в компании Google, и во многих мобильных приложениях.

3 Краткий обзор современных СУБД. Схема лицензирования

18

□ Бесплатные СУБД

□ Коммерческие промышленные СУБД (большинство производителей предлагают также бесплатную версию)

Файл-серверные и встраиваемые СУБД практически все являются бесплатными, из бесплатных клиент-серверных СУБД наиболее известные: **Firebird, PostgreSQL и MySQL**.

□ Плюсы **свободных СУБД**: бесплатно, менее требовательны к ресурсам ПК, обладают богатым функционалом и хорошей производительностью при грамотной настройке, достаточно надёжны.

□ Минусы: никто не даст гарантии, что через определенное время проект не перестанет существовать, т.к. его поддерживает сообщество энтузиастов, также сложно найти техническую поддержку.

□ Плюсы **коммерческих СУБД**: хорошая задокументированность, высокая производительность, масштабируемость, надёжность, поддерживаемость, наличие встроенных инструментов для разработки и администрирования. Вероятность того, что компания Oracle, Microsoft или IBM перестанут поддерживать свои системы, стремится к нулю.

□ Минусы: они более требовательны к ресурсам, чем бесплатные аналоги, стоят денег и немалых.

3 Краткий обзор современных СУБД. Схема лицензирования

	Бесплатные версии	Ограничения
Microsoft	SQL Server 2005 Express Edition	Размер БД - до 4 Гб, количество баз не ограничено, не более 1 Гб ОП и только 1 процессор (ядро). Поддерживаемые платформы: только Windows 2005 – только x86, 2008 – x86 и x64.
	SQL Server 2008 Express	
	SQL Server 2008 R2 Express Edition	Размер БД - до 10 Гб, количество баз не ограничено, не более 1 Гб ОП и только 1 процессор (ядро).
	SQL Server 2012 Express	Поддерживаемые платформы: только Windows x86 и x64.
Oracle	Oracle Database 11g Express Edition, (Oracle Database XE)	Суммарно до 11Гб БД, использует не более 1Гб ОП и только 1 процессор (ядро). Поддерживаемые платформы: Windows x86, Linux x64.
IBM	IBM DB2 Express-C	Размер базы не ограничен, используется до 4Гб ОП и до 2-х процессоров. Поддерживаемые платформы: Windows x86 и x64, Linux x86 и x64, Unix x86 и x64, Solaris x86 и x64,

4 Критерии выбора СУБД пользователем:

20

- особенность задачи (предметной области);
- модель данных;
- тип создаваемой ИС (локальная или сетевая, централизованное или распределенное хранение данных);
- объем БД;
- масштабируемость по пользователям.

5 Классификация баз данных

21



5 Классификация баз данных

22

Классификация баз данных по виду хранимых объектов

Ориентированы на хранение слабо структурированной информации
Основные операции: <ul style="list-style-type: none">• поиск текста по фрагменту,• поиск документа по ключевым словам и сочетаниям• отслеживание статуса документа• мониторинг движения документа
Lotus Notes XML



Ориентированы на хранение множества взаимосвязанных фактов
Основные операции: <ul style="list-style-type: none">• Запросы на поиск взаимосвязанной информации• Вычисление новой информации из имеющейся• Выполнение расчетов• Составление множества отчетов с вычислением и обобщением данных
IMS, IDS, dbVista, Informix, Oracle, Ms Access, Ms SQL Server, SQL Base, FoxPro, Paradox, Cach, MySQL, Clipper

5 Классификация баз данных

23

Классификация баз данных по допустимым операциям обработки информации



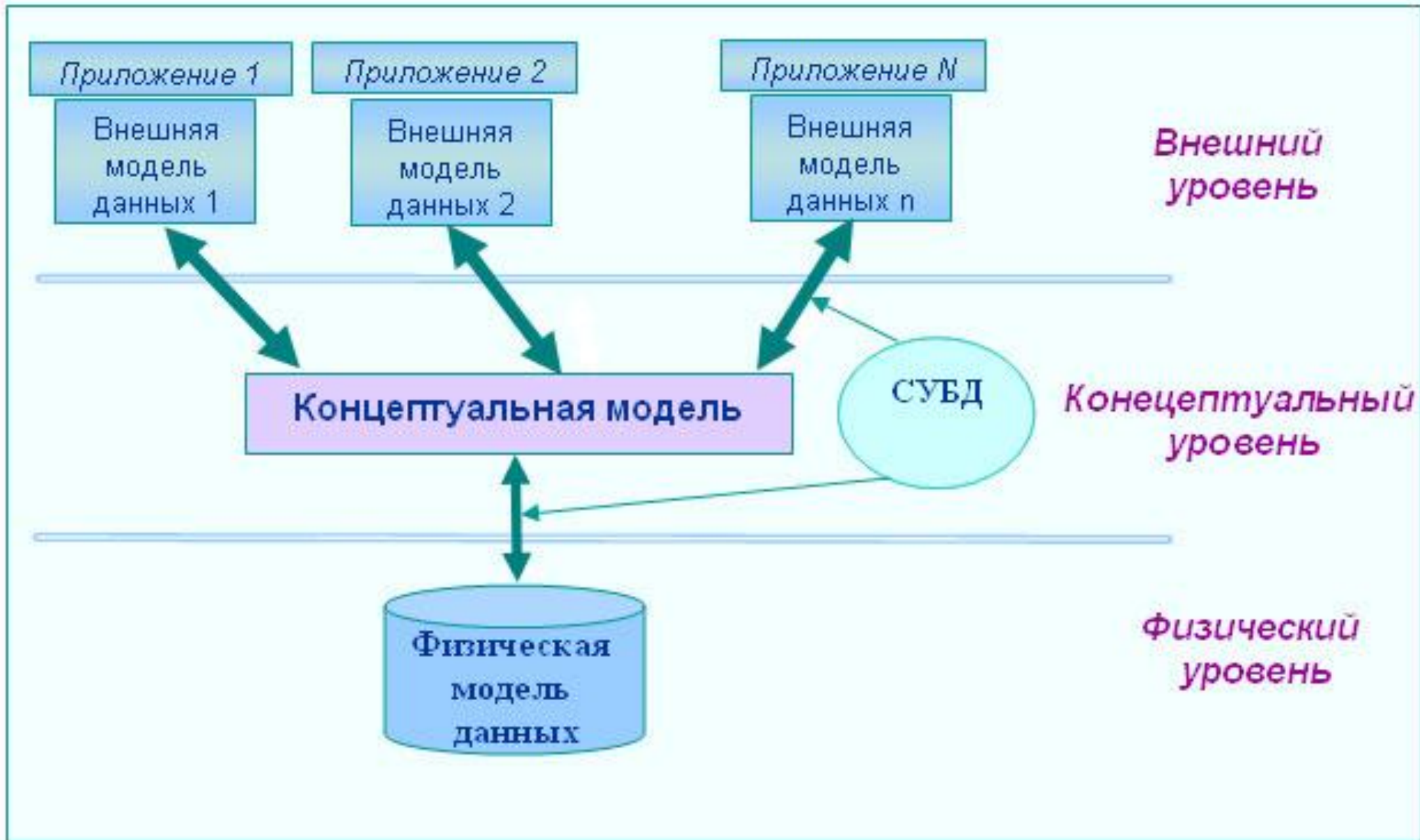
6 Трехуровневая архитектура ANSI/SPARC

24

- ▣ **Уровень внешних моделей** — самый верхний уровень, где каждая модель имеет свое «видение» данных. Этот уровень определяет точку зрения на БД отдельных приложений. Каждое приложение видит и обрабатывает только те данные, которые необходимы именно этому приложению. Например, система распределения работ использует сведения о квалификации сотрудника, но ее не интересуют сведения об окладе, домашнем адресе и телефоне сотрудника, и наоборот, именно эти сведения используются в подсистеме отдела кадров.
- ▣ **Концептуальный уровень** — центральное управляющее звено, здесь база данных представлена в наиболее общем виде, который объединяет данные, используемые всеми приложениями, работающими с данной базой данных. Фактически концептуальный уровень отражает обобщенную модель предметной области (объектов реального мира), для которой создавалась база данных. Как любая модель, концептуальная модель отражает только существенные, с точки зрения обработки, особенности объектов реального мира.
- ▣ **Физический уровень** — собственно данные, расположенные в файлах или в страничных структурах, расположенных на внешних носителях информации.

6 Трехуровневая архитектура ANSI/SPARC

25



6 Трехуровневая архитектура ANSI/SPARC

26

Концептуальный уровень (conceptual level) представления данных предназначен для отображения внешнего уровня на внутренний. Фактически, это полное представление требований к данным со стороны организации, которое не зависит от способа их хранения.

На концептуальном уровне представлены следующие компоненты:

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

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

Внутренний уровень – это физическое представление базы данных в компьютере. Этот уровень описывает, как информация хранится в базе данных.

6 Трехуровневая архитектура ANSI/SPARC

27

Внутренний уровень описывает физическую реализацию базы данных и предназначен для достижения оптимальной производительности и обеспечения экономичности использования дискового пространства..

На внутреннем уровне хранится следующая информация:

- распределение дискового пространства для хранения данных и индексов;
- описание подробностей сохранения записей (с указанием реальных размеров сохраняемых элементов данных);
- сведения о размещении записей;
- сведения о сжатии данных и выбранных методах их шифрования.

Ниже внутреннего уровня находится **физический уровень** (physical level), который контролируется операционной системой, но под руководством СУБД.