

БАЗЫ ДАННЫХ

Лекция 7.

Архитектура информационных систем
на основе баз данных

Архитектура информационных систем

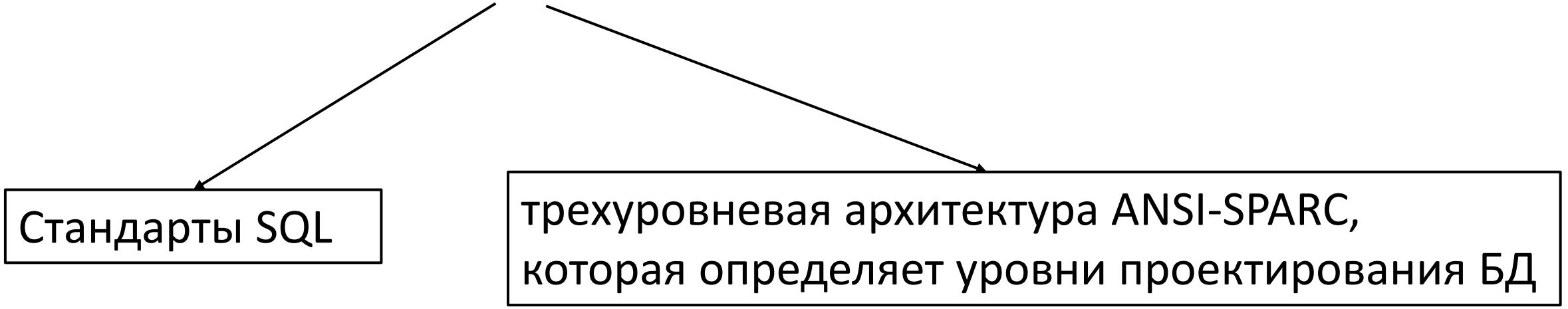


Архитектура информационных систем на основе БД

- При развитии БД предлагались различные способы реализации того, как должна быть устроена СУБД

Американский национальный институт стандартов

Стандарт ANSI для БД (American National Standards Institute)

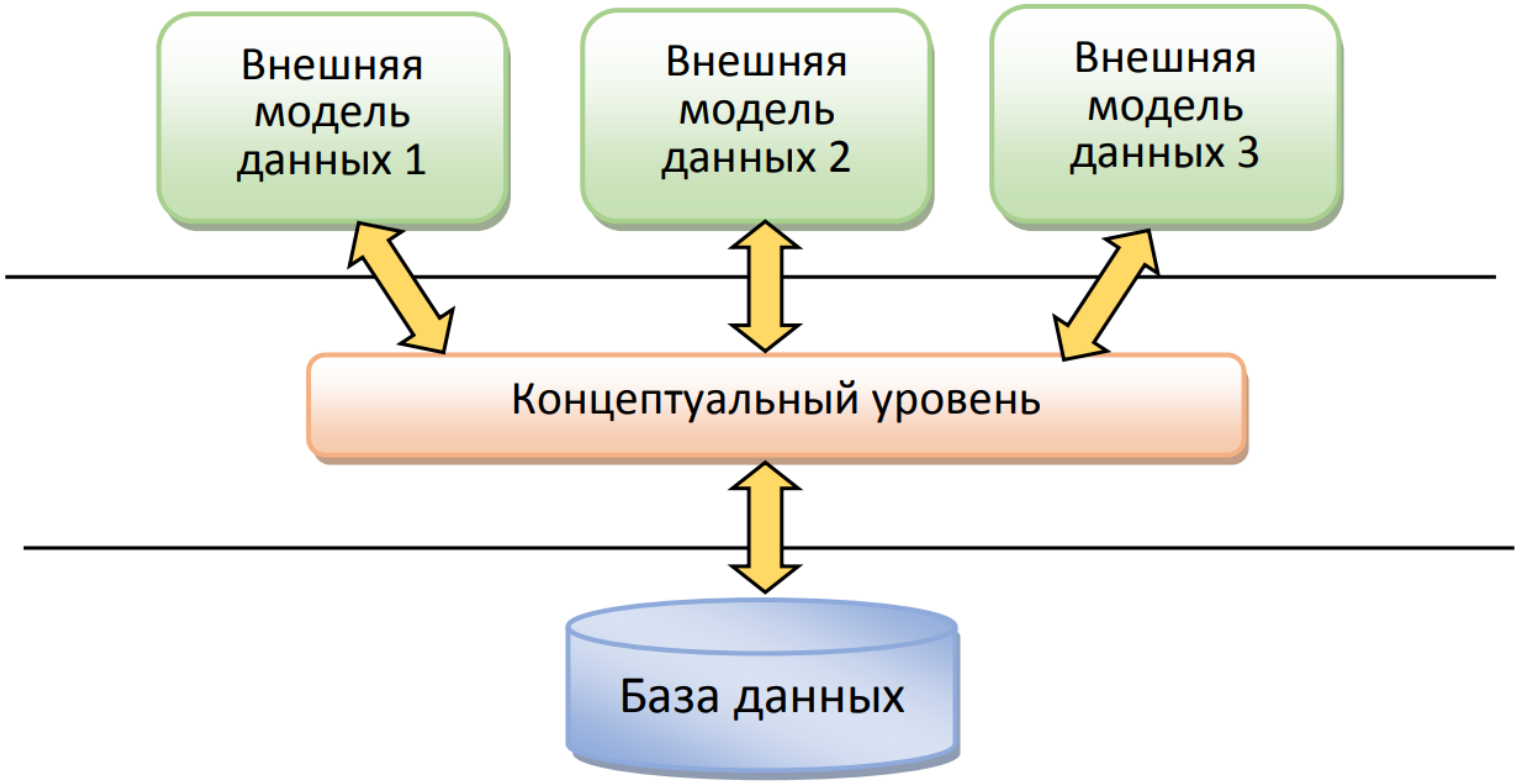


оказалась самым жизнеспособным вариантом

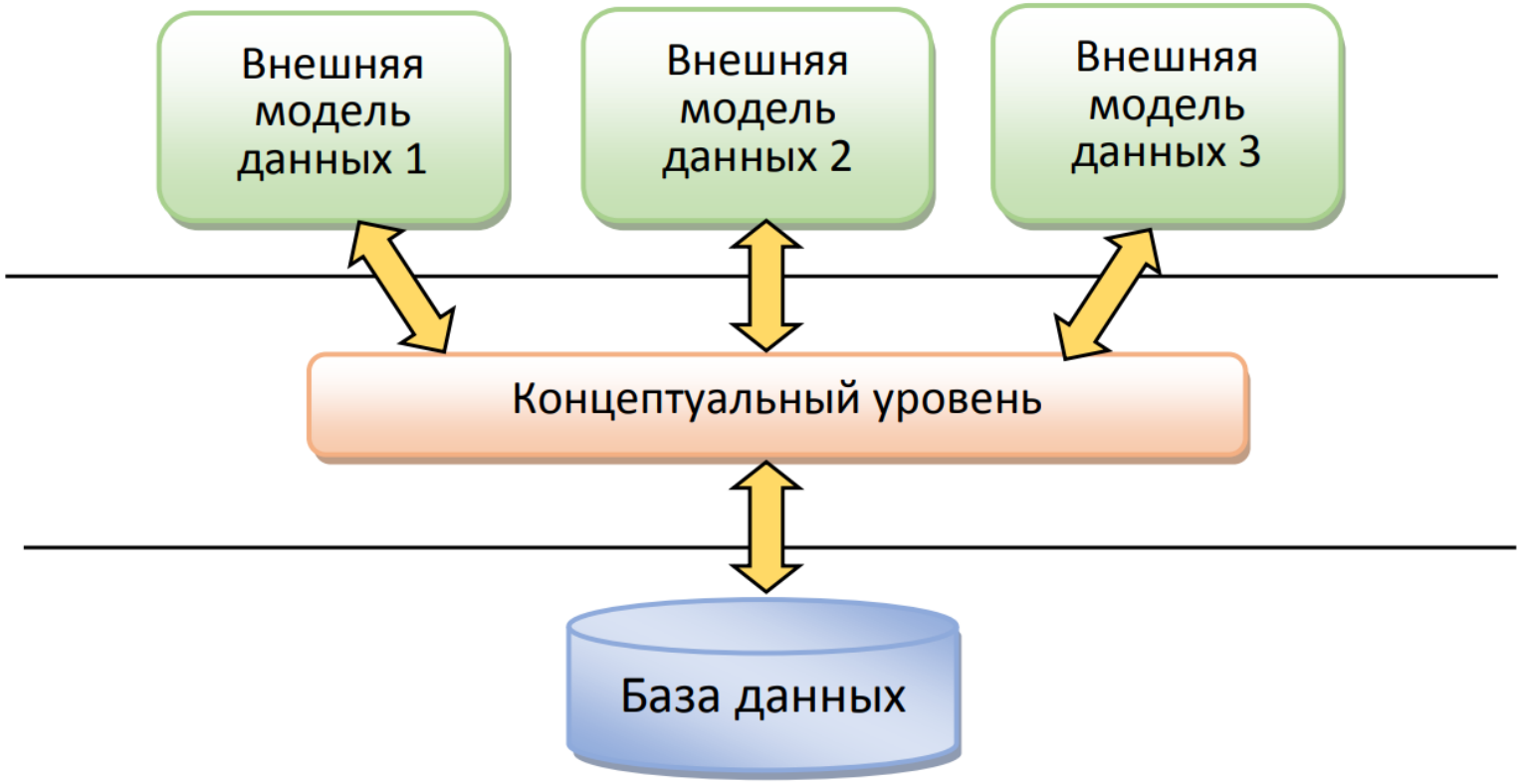
Архитектура информационных систем на основе БД

ANSI-SPARC: трехуровневая система организации БД

- выделяется три уровня представления данных:
  внешний, концептуальный и физический.

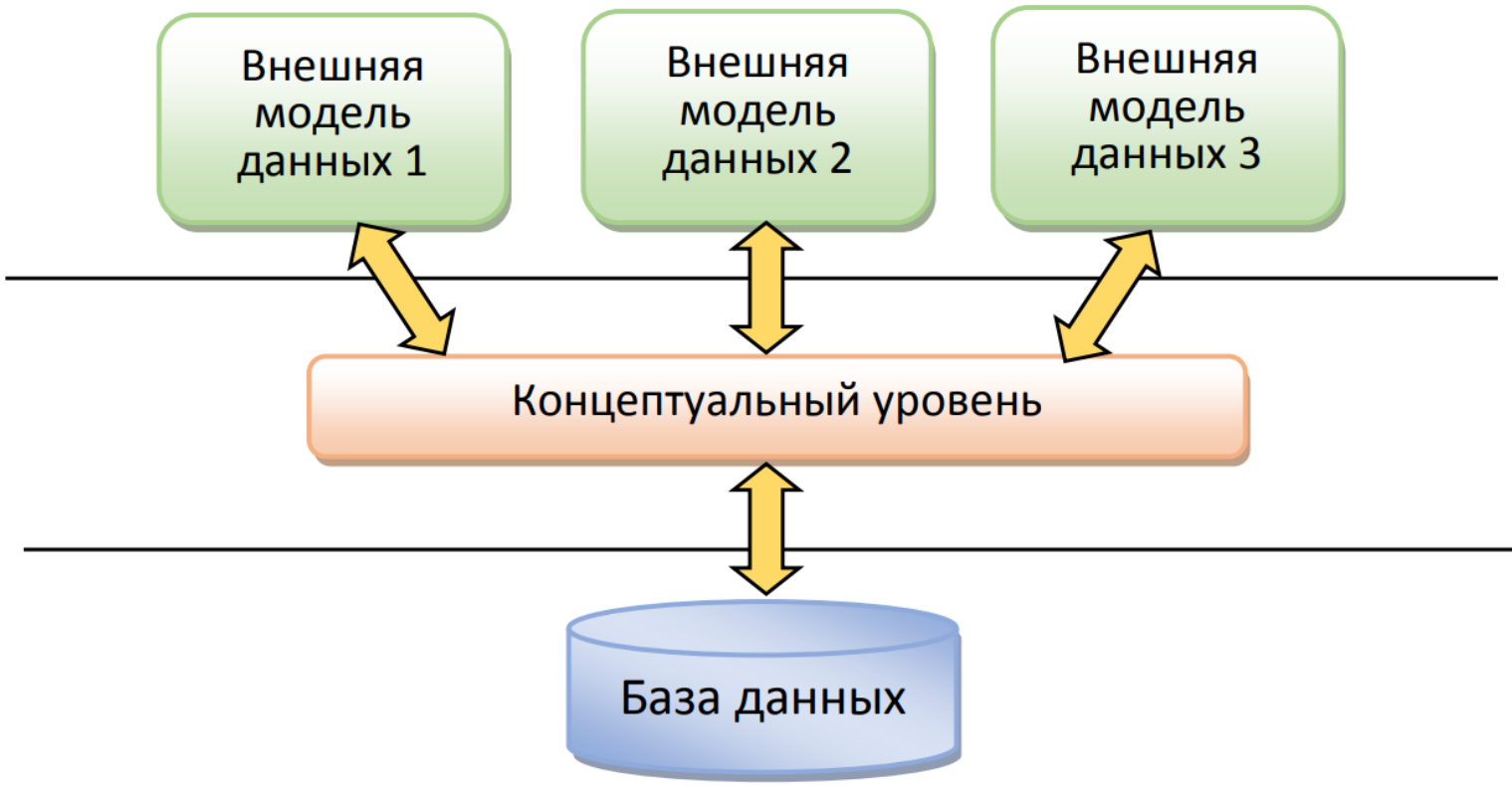


Трехуровневая архитектура СУБД



Концептуальная (логическая) схема определяет глобальный логический проект записей БД и связей между записями

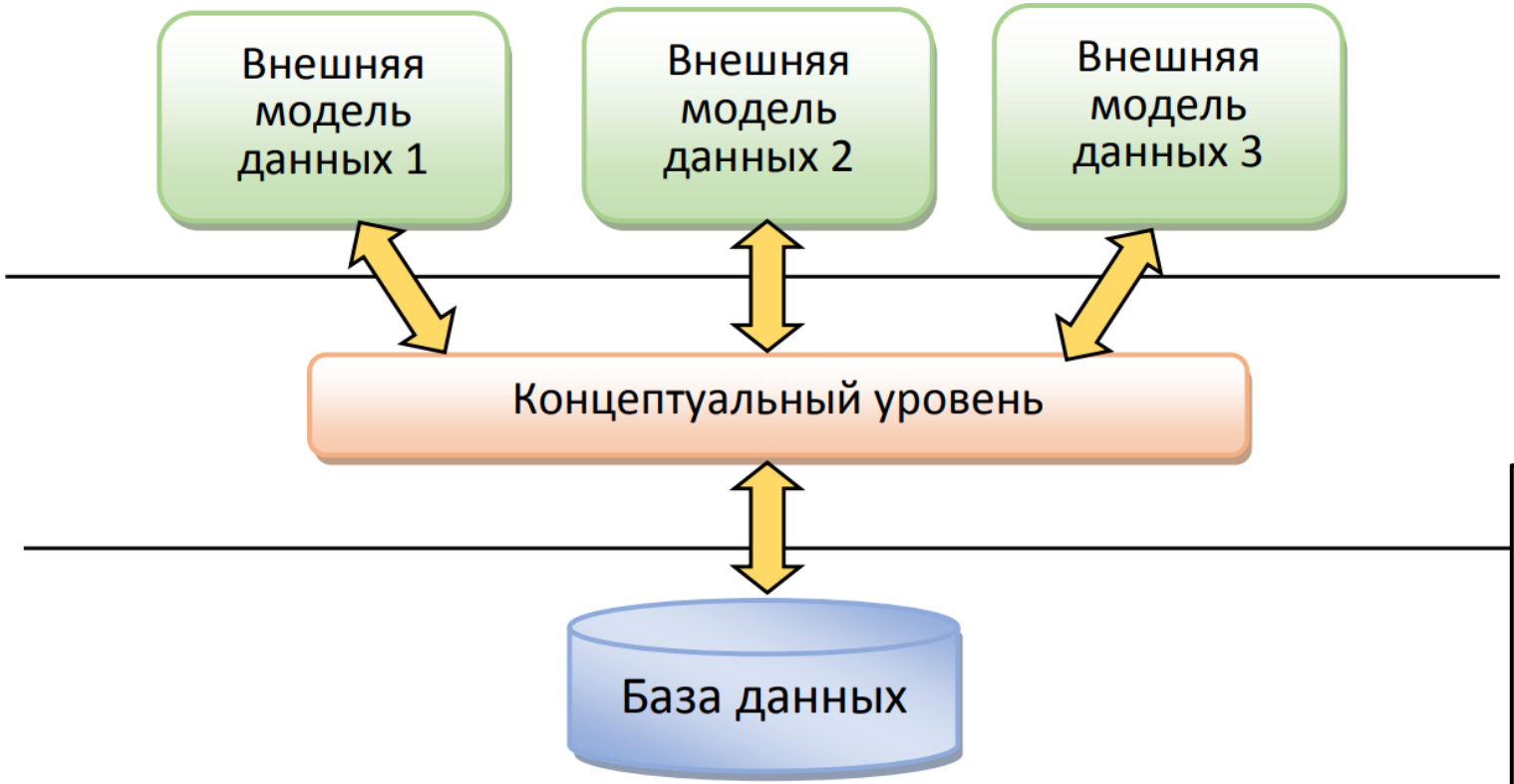
Трехуровневая архитектура СУБД



Внешние схемы определяют пользовательское представление данных в соответствии с различными представлениями различных групп пользователей

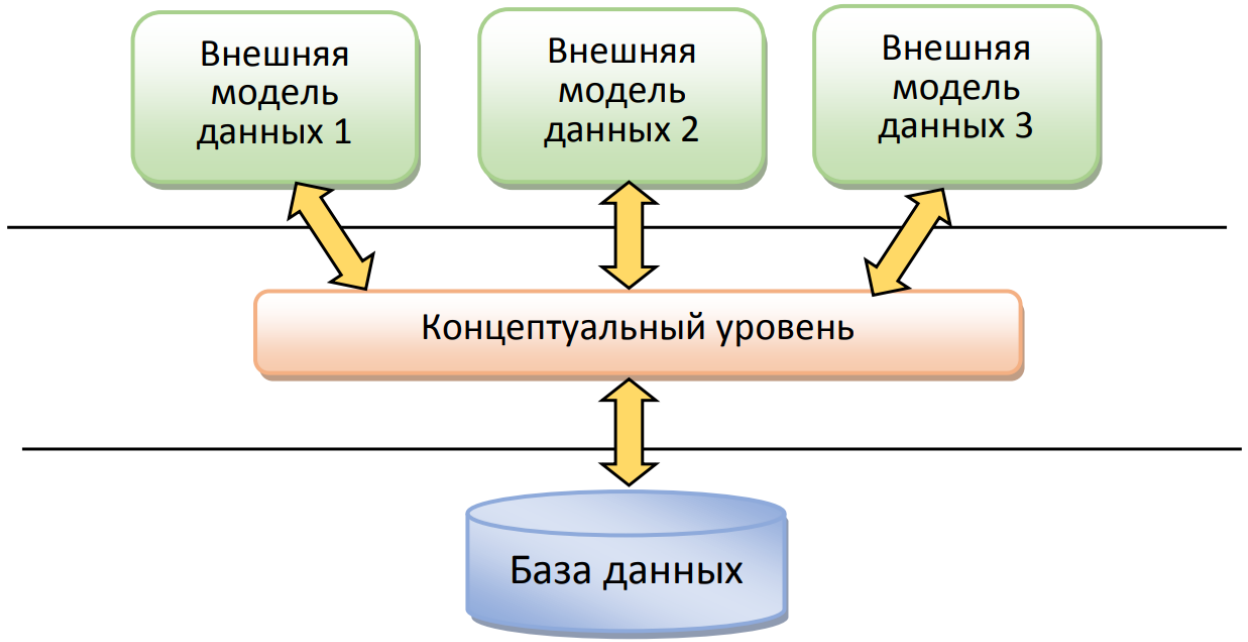
- могут весьма отличаться

Трехуровневая архитектура СУБД



Физическая схема описывает физическое размещение записей БД на устройствах памяти и в файлах, а также индексы, нужные для поддержки логических связей.

Трехуровневая архитектура СУБД



Цель создания:

необходимость воплощения независимости каждого уровня для изоляции программ от особенностей представления данных на более низком уровне.

// Отделение пользовательского представления базы данных от ее физического представления.

Основные подходы к хранению данных

Какая бы модель ни была использована
для хранения и обработки данных необходимо, чтобы выполнялись правила **полноты, непротиворечивости и целостности** данных.

Полнота данных

- база данных должна обеспечивать полное и адекватное описание предметной области;
- при этом должен соблюдаться принцип минимальной избыточности;
- особое внимание на полноту обращается на этапе проектирования базы данных.

Основные подходы к хранению данных

Непротиворечивость данных

- данные, которые хранятся в базе данных, должны проверяться на правильность при вводе;
- существует запрет на дублирование данных.

Целостность данных:

- при описании связей должна обеспечиваться правильность ссылок между таблицами (обеспечивается каскадным обновлением/удалением);
- блокировка модифицируемых записей при одновременной работе с БД;
- механизм транзакций (последовательность операций над БД), (возможность отката).

Классификация информационных систем



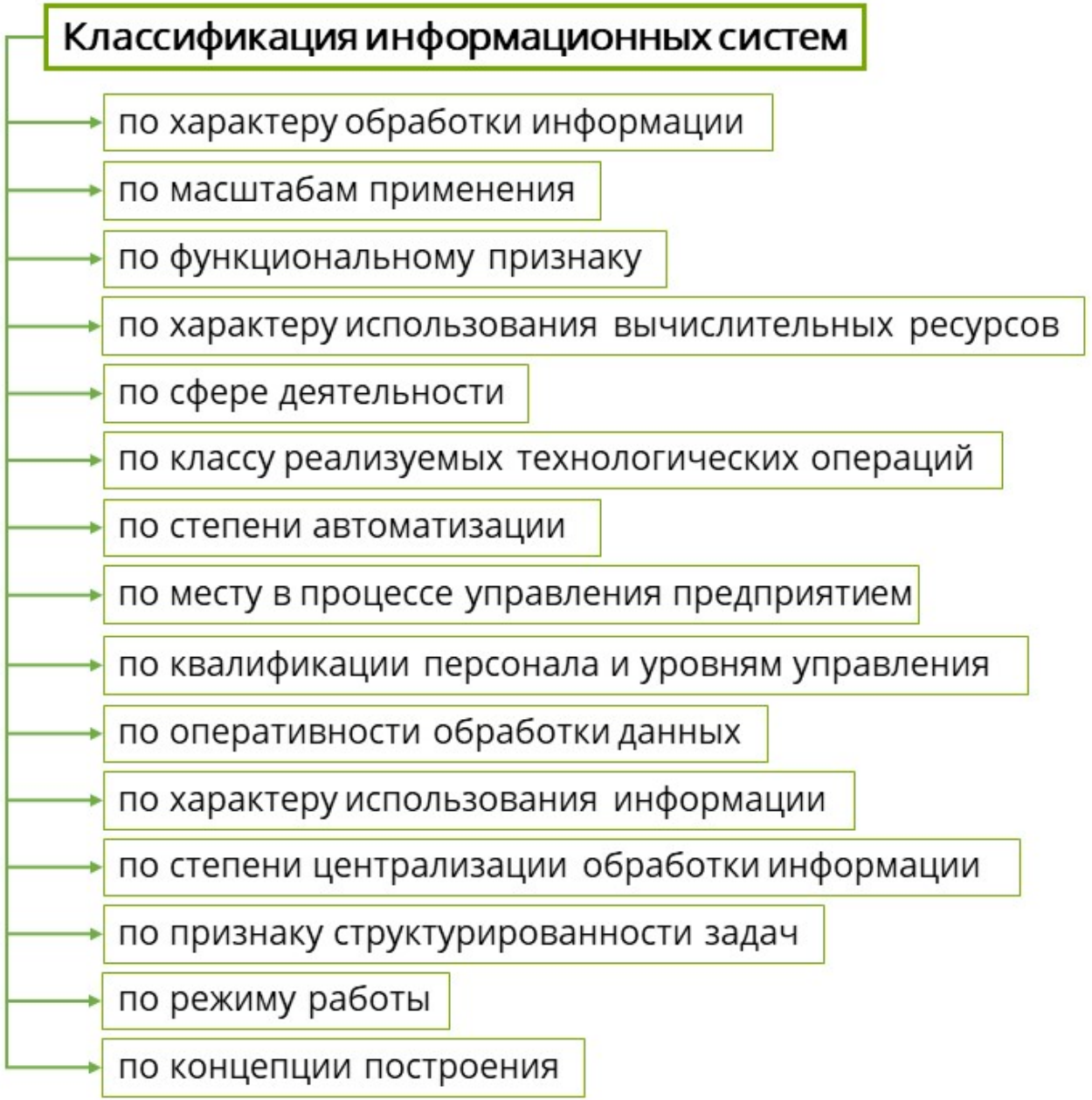
Классификация ИС

Информационная система (ИС) в самом широком смысле представляет собой **программный комплекс**, функции которого состоят в

- поддержке надежного хранения информации в памяти компьютера,
- выполнении специфических для данного приложения преобразований информации и/или вычислений,
- предоставлении пользователям удобного и легко осваиваемого интерфейса.

Обычно объемы информации, с которыми работают ИС, достаточно велики, а сама информация имеет достаточно сложную структуру.

Классификация ИС



Классификация ИС

По **способу организации хранения и доступа к данным**:

- локальные (автономные),
- централизованные,
- распределенные;



По способу организации хранения и доступа к данным

Локальная информационная система

- реализуется на автономном компьютере
- система может содержать несколько простых приложений, связанных общим информационным фондом (БД)
- рассчитана на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место.



По способу организации хранения и доступа к данным

Локальная информационная система

- создаются с помощью «настольных» СУБД

Настольные (несетевые) СУБД обрабатывают реляционные таблицы путем полной их загрузки в оперативную память компьютера. После обработки данных таблица полностью выгружается из оперативной памяти.

DBASE, Visual DBASE, Paradox, Microsoft FoxPro, Visual FoxPro, Access

- или с помощью файловой системы и диалоговой оболочки для ввода, редактирования и обработки данных.

По способу организации хранения и доступа к данным

Локальная информационная система

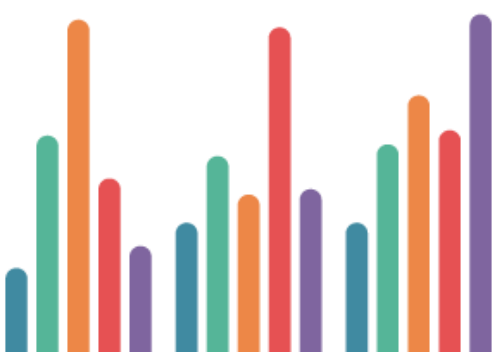
- дешевизна
- простота эксплуатации
- «персонализация рабочей среды», когда пользователь может выбрать себе инструменты для работы, наиболее соответствующие его потребностям.

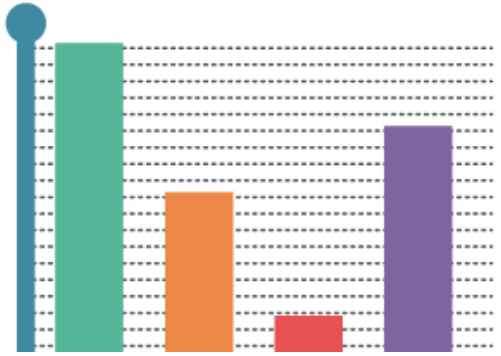


По способу организации хранения и доступа к данным

При выборе настольных СУБД следует обратить внимание на:

- наличие базового инструментария для построения таблиц, работы с данными и структурами в диалоговом режиме, сортировки и импорта записей;
- наличие форм как инструментального средствами для формирования структур элементов данных и экранов просмотра;
- наличие средств конструирования простых и сложных запросов и отчетов;

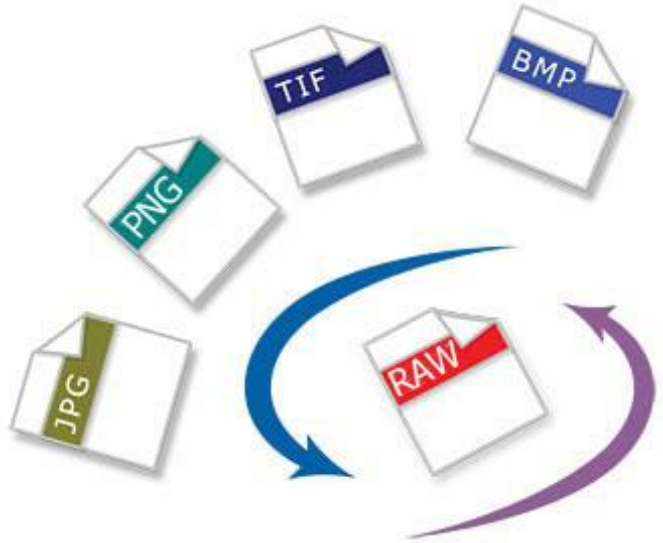




По способу организации хранения и доступа к данным

<u>При выборе настольных СУБД следует обратить внимание на:</u>

- наличие средств быстрой разработки программ и внешних интерфейсов;
- наличие средств связности характеризует связь с данными, хранящимися в файлах различных форматов;
- наличие средства для работы в Интернет (инструментарий для публикации статических и динамических документов, для ввода и извлечения данных из сети).



По способу организации хранения и доступа к данным

Централизованная информационная система

- хранится на центральном компьютере
- пользователи и прикладные программы имеют удаленный доступ к базе данных.



По способу организации хранения и доступа к данным

Централизованная информационная система

- хранится на центральном компьютере
- пользователи и прикладные программы имеют удаленный доступ к базе данных.

Преимущества централизованной ИС:

- минимальные затраты на корректировку,
- поэтому такая система предпочтительна,
  если важны требования к безопасности и целостности данных.

По способу организации хранения и доступа к данным

Централизованная информационная система

Недостатки централизованной ИС:

- сложность в обслуживании,
- увеличение времени отклика,
- затраты на передачу данных,
- неисправность центральной системы выводит из рабочего состояния всю сеть.

По способу организации хранения и доступа к данным

Централизованные базы данных реализуются на базе двух архитектур с сетевым доступом:

1. Архитектура «файл – сервер»

- Выделение одной из машин в сети в качестве центральной (*сервер файлов*), на которой хранится совместно используемая централизованная БД.



По способу организации хранения и доступа к данным

Централизованные базы данных:

1. Архитектура «файл – сервер»

- Выделение одной из машин в сети в качестве центральной (*сервер файлов*), на которой хранится совместно используемая централизованная БД.

- Остальные машины сети исполняют роль рабочих станций, на которых в основном и производится обработка данных, получаемых **в виде файлов** базы данных в соответствии с запросами пользователей;

По способу организации хранения и доступа к данным

Централизованные базы данных:

2. Архитектура «клиент – сервер»

- стала стандартом для современных СУБД
- *сервер* владеет и распоряжается информационными ресурсами системы, а *клиент* пользуется ими.



По способу организации хранения и доступа к данным

Централизованные базы данных:

2. Архитектура «клиент – сервер»

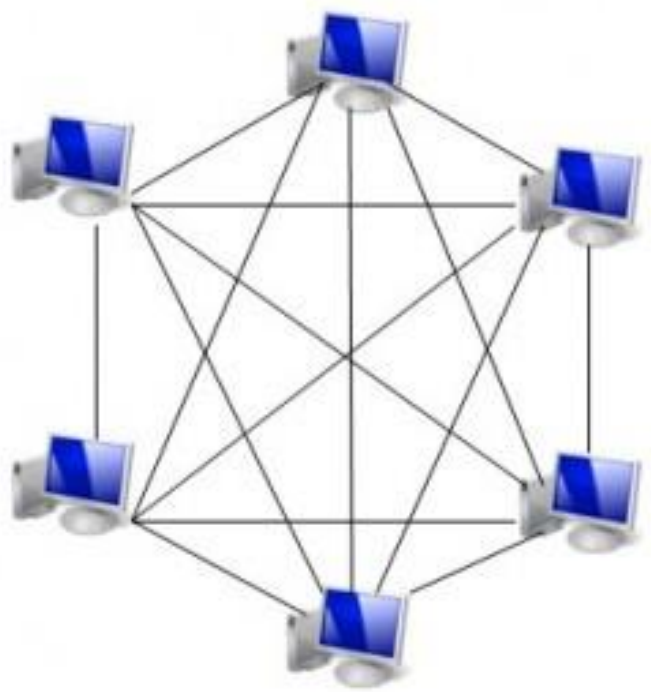
- стала стандартом для современных СУБД
- *сервер* владеет и распоряжается информационными ресурсами системы, а *клиент* пользуется ими.

- Сервер БД помимо хранения базы данных обеспечивает выполнение основного объема обработки данных.
- Запрос клиента (рабочей станции) порождает поиск и извлечение данных на сервере, которые затем транспортируются по сети к клиенту (в отличие от передаваемых файлов в предыдущей архитектуре).

По способу организации хранения и доступа к данным

Распределенная информационная система

- Предполагает хранение и управление данными в нескольких узлах компьютерной сети и передачу данных между ними в процессе выполнения запросов.
- На разных компьютерах могут храниться не только различные таблицы, но и разные фрагменты одной таблицы.
- Для пользователя не имеет значения, как организовано хранение данных.



По способу организации хранения и доступа к данным

Распределенная информационная система

Принято строить различными способами, как системы:

- с архитектурой «**клиент-сервер**»;
- с распределенными вычислениями;
- на основе Internet/Intranet-технологий.

Архитектура «клиент-сервер»



Модели архитектуры «клиент-сервер»

Компьютеры в сети не являются равноправными:

- Серверы владеют и распоряжаются информационно-вычислительными ресурсами (процессоры, файловая система, почтовая служба, служба печати, база данных)
- Клиенты имеют возможность обращаться к этим службам, пользуясь услугами серверов.

Конкретный сервер определяется видом ресурса, которым он владеет:

- Сервер БД — обслуживает запросы клиентов по обработке данных,
- файловой сервер – управляет файловой системой, и т. д.

Модели архитектуры «клиент-сервер»

- В сети один и тот же компьютер может выполнять роль как клиента, так и сервера.
- Этот же принцип распространяется и на **взаимодействие программ**. Если одна из них выполняет некоторые функции, предоставляя другим набор услуг, то такая программа выступает в качестве сервера. Программы, которые пользуются этими услугами, принято называть клиентами.

Ядро реляционной SQL-ориентированной СУБД – сервер БД (SQL-сервер)

Программа, обращающаяся для обработки данных — SQL-клиент.

Модели архитектуры «клиент-сервер»

Как программа СУБД реализует следующие основные функции:

- управление данными, находящимися в базе;
- обработка информации с помощью прикладных программ;
- представление информации в удобном для пользователя виде.

При размещении СУБД в сети возможны различные варианты распределения функций по узлам (модели архитектуры).

Двухзвенные модели распределения функций

Распределение функций СУБД между двумя узлами сети.

Сервер - обязательно присутствует функция управления данными,

Клиент - обязательно занимается вопросами представления информации.

Наиболее типичными вариантами разделения функций между компьютером-сервером и компьютером-клиентом :

- распределенное представление;
- удаленное представление (DataBase Server — DBS);
- удаленный доступ к данным (Remote Data Access — RDA);
- распределенная БД.

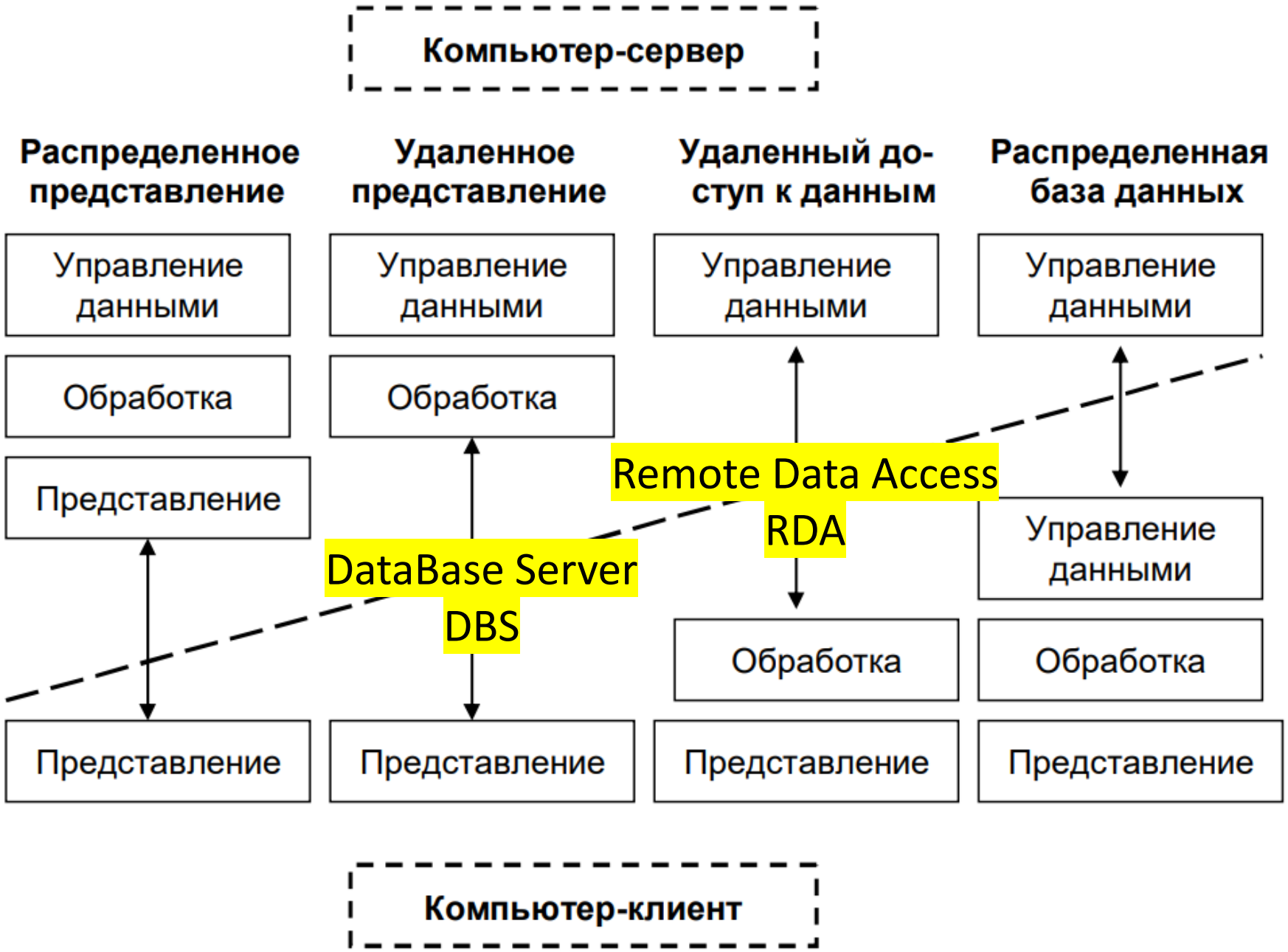
Двухзвенные модели архитектуры «клиент-сервер»



Двухзвенные модели распределения функций



Двухзвенные модели распределения функций

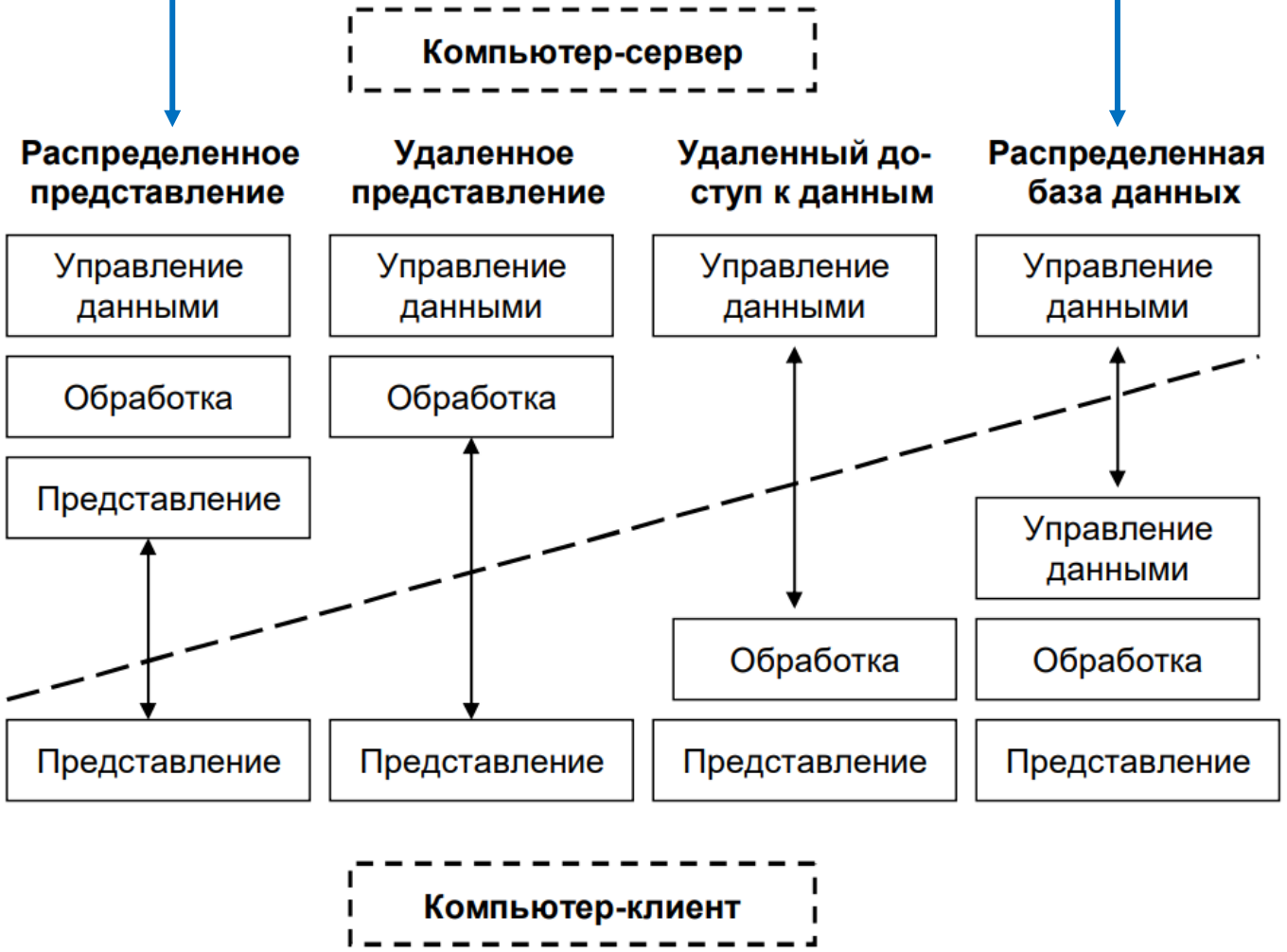


Двухзвенные модели распределения функций



Двухзвенные модели распределения функций

выполнение одной из двух функций одновременно на двух компьютерах



Двухзвенные модели распределения функций

выполнение одной из двух функций одновременно на двух компьютерах



строгое распределение функций между клиентом и сервером

Двухзвенные модели распределения функций

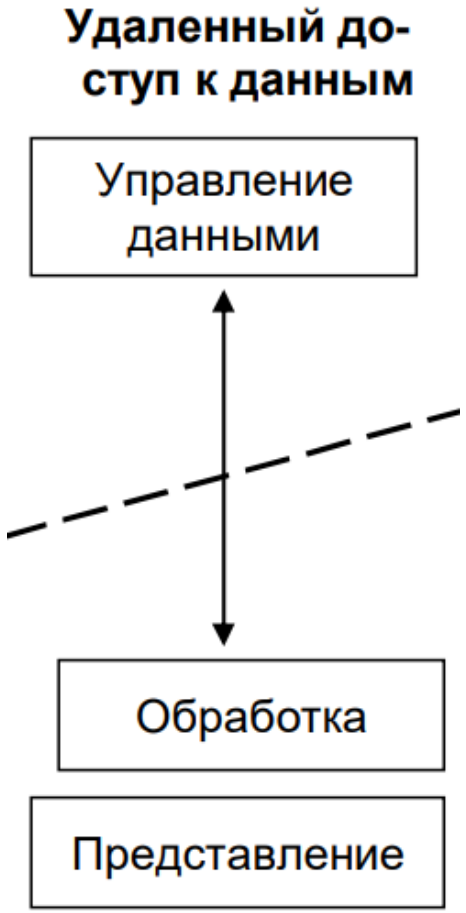


Двухзвенные модели распределения функций

Модель удаленного доступа к данным (Remote Data Access — RDA):

- сервис **управления данными** осуществляется сервером

- программы, реализующие функции **представления** информации и логику прикладной **обработки**, совмещены и выполняются на клиенте,



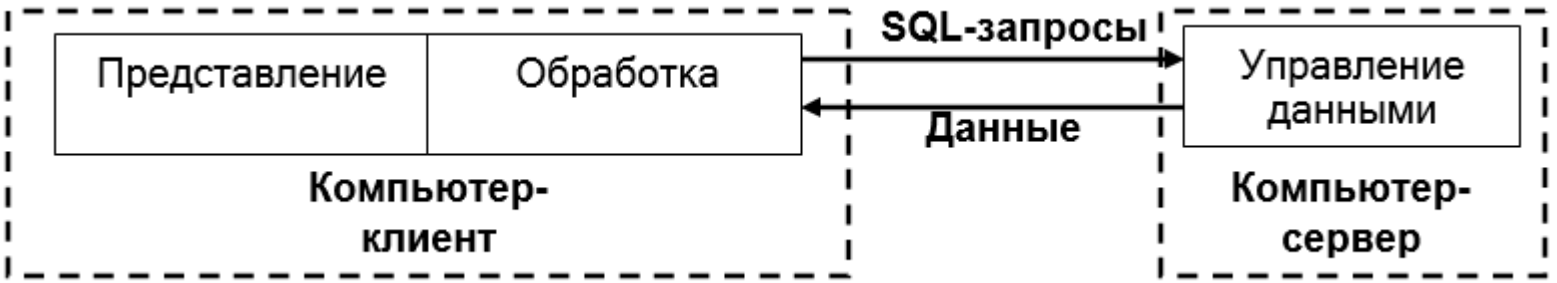
Двухзвенные модели распределения функций

Модель удаленного доступа к данным (Remote Data Access — RDA):

- обращение за сервисом управления данными происходит через среду передачи с помощью

операторов языка SQL
или вызовом функций специальной библиотеки API

(Application Programming Interface —
интерфейса прикладного программирования)



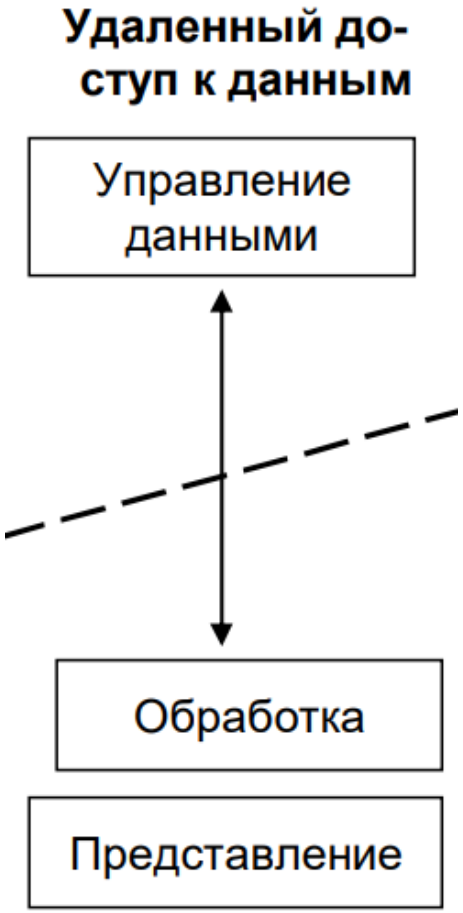
Двухзвенные модели распределения функций

Модель удаленного доступа к данным (Remote Data Access — RDA):

Основное достоинство :

Существует множество готовых СУБД, имеющих SQL-интерфейсы, и инструментальных средств, обеспечивающих быстрое создание программ клиентской части.

Средства разработки чаще всего поддерживают графический интерфейс пользователя в MS Windows, стандарт интерфейса ODBC и средства автоматической генерации кода.



Двухзвенные модели распределения функций

Модель удаленного доступа к данным (Remote Data Access — RDA):

Недостатки модели:

- Довольно высокая загрузка системы передачи данных вследствие того, что вся логика сосредоточена в приложении, а обрабатываемые данные расположены на удаленном узле (активный обмен).

- Системы, построенные на основе RDA, неудобны с точки зрения разработки, модификации и сопровождения:

- при незначительном изменении функций системы требуется переделка всей прикладной ее части, возможно дублирование кода приложений

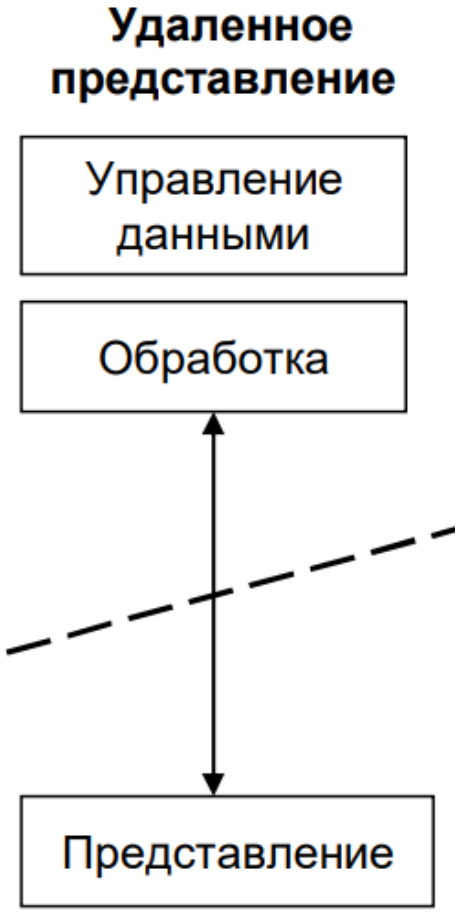
Двухзвенные модели распределения функций



Двухзвенные модели распределения функций

Модель удаленного представления (DataBase Server – DBS):

- сервис **управления данными** осуществляется сервером
- **прикладные** функции обеспечиваются приложением, находящимся на компьютере-сервере.
- функции компьютера-клиента ограничиваются функциями **представления** информации

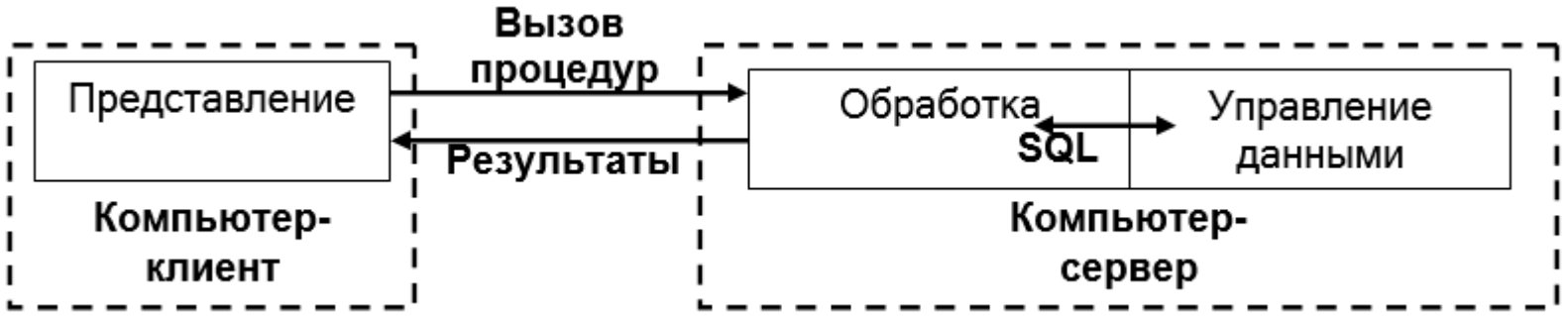


Двухзвенные модели распределения функций

Модель удаленного представления (DataBase Server – DBS):

Перед сервером БД стоят следующие задачи:

- В каждый момент времени БД должна отражать текущее состояние предметной области, данные в БД должны быть непротиворечивыми.
- Необходим постоянный контроль за состоянием БД, отслеживание всех изменений и адекватная реакция на них. Возникновение некоторой ситуации в БД должно четко и оперативно влиять на ход выполнения прикладной задачи.
- Одна из важных проблем СУБД — контроль типов данных, в том числе нестандартных



Двухзвенные модели распределения функций

Модель удаленного представления (DataBase Server – DBS):

- Централизованный контроль в модели сервера баз данных выполняется с использованием механизма **триггеров**.
- Триггеры срабатывают при возникновении определенного события в БД. Триггеры могут вызывать **хранимые процедуры**.
- И хранимые процедуры, и триггеры хранятся в словаре БД, они могут быть использованы несколькими клиентами, что существенно уменьшает дублирование алгоритмов обработки данных в разных клиентских приложениях.

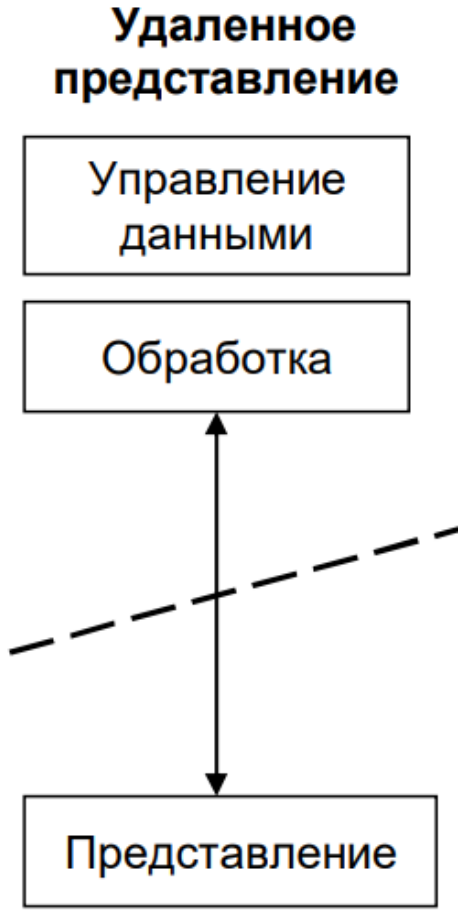
Двухзвенные модели распределения функций

Модель удаленного представления (DataBase Server – DBS):

Достоинства DBS-модели:

- возможность хорошего централизованного администрирования приложений на этапах разработки, сопровождения и модификации,

- эффективное использование вычислительных и коммуникационных ресурсов.

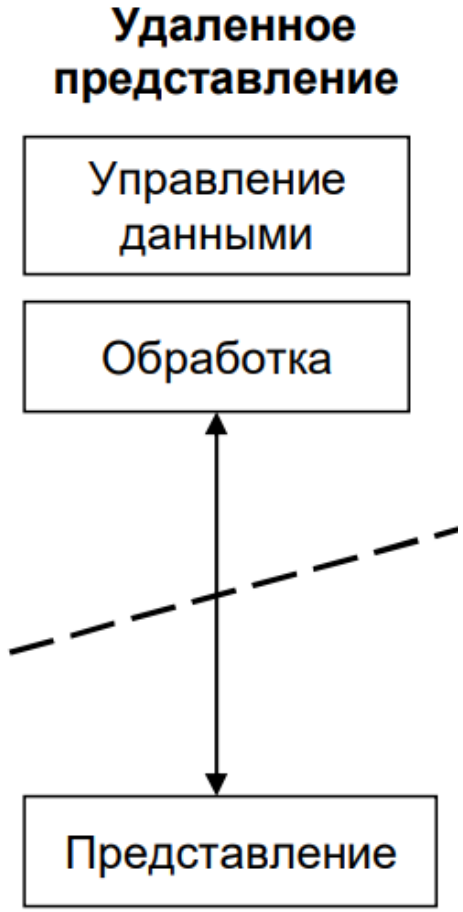


Двухзвенные модели распределения функций

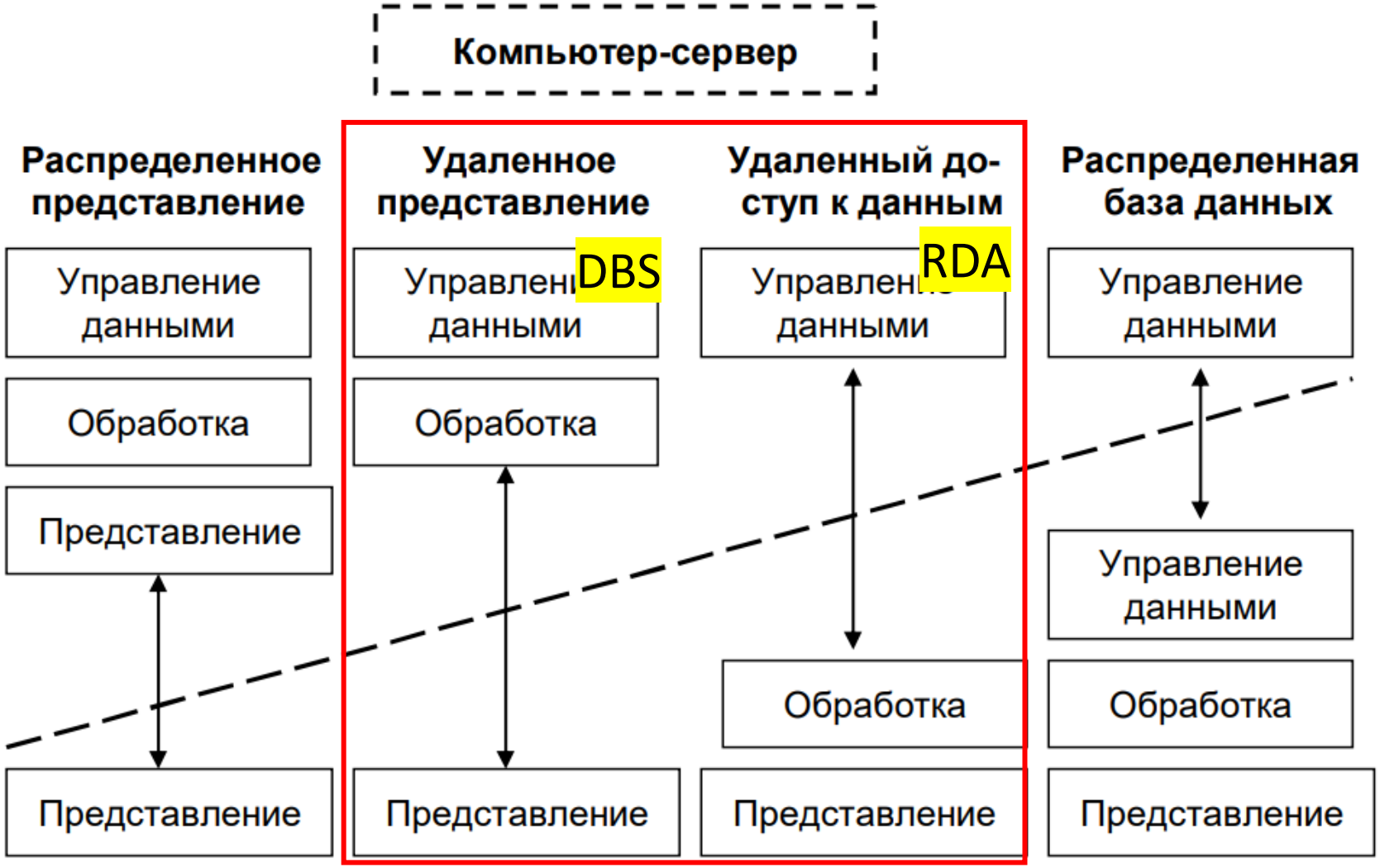
Модель удаленного представления (DataBase Server – DBS):

Недостатки DBS-модели:

- сильная привязка операторов хранимых процедур к конкретной СУБД,
- низкая эффективность использования вычислительных ресурсов ЭВМ, поскольку не удается организовать управление входным потоком запросов к программам компьютера-сервера, а также обеспечить перемещение процедур на другие компьютеры-серверы.



Двухзвенные модели распределения функций



На практике чаще используется разумный синтез RDA- и DBS-моделей для построения многопользовательских информационных систем.

Двухзвенные модели распределения функций



Двухзвенные модели распределения функций



Двухзвенные модели распределения функций

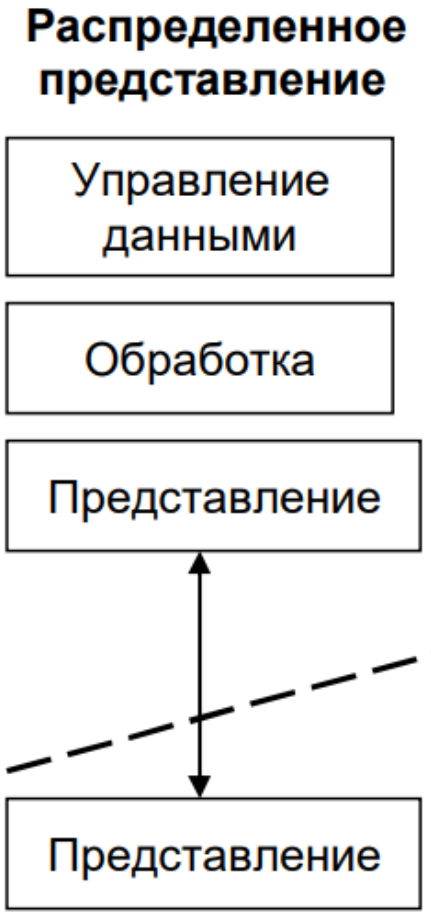


Двухзвенные модели распределения функций

Модель распределенного представления

- Мощный компьютер-сервер, клиентская часть системы практически вырождена.

- Функцией клиента – просто **отображение информации** на экране монитора и связь с основным компьютером через локальную сеть.



Двухзвенные модели распределения функций

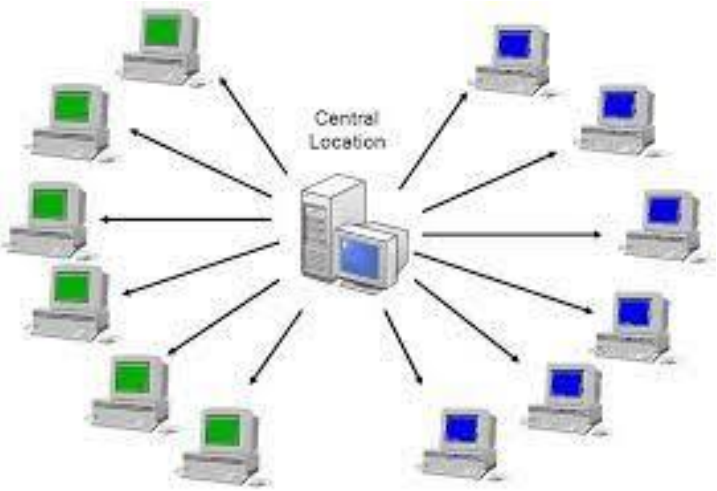
Модель распределенного представления

СУБД подобного рода могут иметь место в сетях, поддерживающих работу так называемых **X-терминалов**.

Хост-машина должен иметь достаточную мощность, чтобы обслуживать несколько X-терминалов.

X-терминал тоже должен обладать достаточно быстрым процессором и иметь достаточный объем оперативной памяти (дисковые накопители отсутствуют).

Все ПО находится на хост-машине. ПО X-терминала, выполняющее функции управления представлением и сетевые функции, загружается по сети с сервера при включении X-терминала.



Двухзвенные модели распределения функций

Модель распределенного представления

Достоинства модели:

- простота обслуживания и управления доступом к системе,
- относительная дешевизна (невысокая стоимости оконечных терминалов).

Недостатки модели:

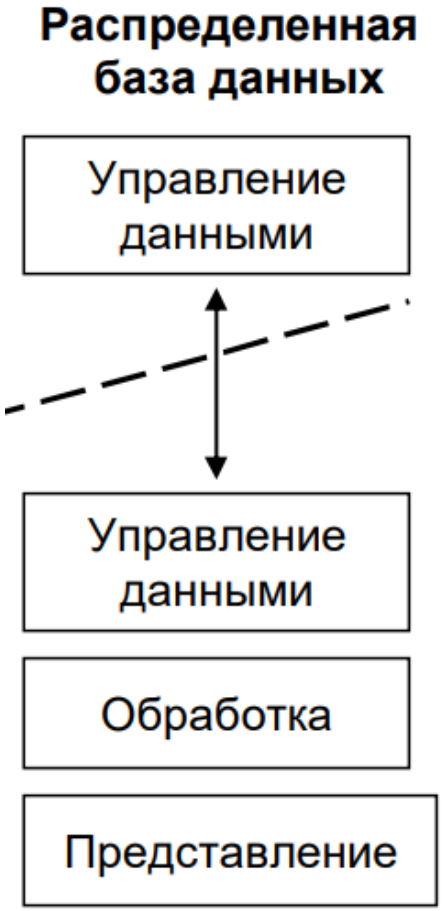
- уязвимость системы при невысокой надежности центрального узла,
- высокие требования к серверу по производительности при большом числе клиентов.

Двухзвенные модели распределения функций



Двухзвенные модели распределения функций

Модель распределенной базы данных



Мощный **компьютер-клиент**, причем данные хранятся на компьютере-клиенте и на компьютере-сервере.

Взаимосвязь обеих баз данных может быть двух разновидностей:

а) в локальной и удаленной базах хранятся отдельные части единой БД;

б) локальная и удаленная БД являются синхронизируемыми друг с другом копиями.

Двухзвенные модели распределения функций

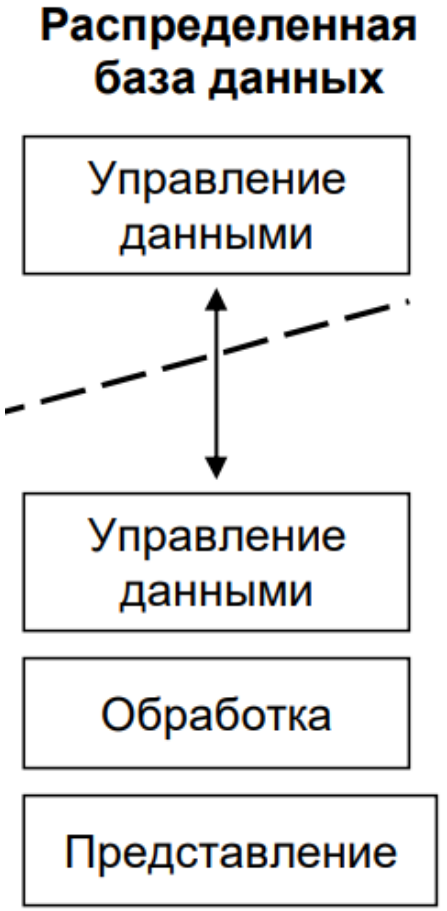
Модель распределенной базы данных

Достоинства модели:

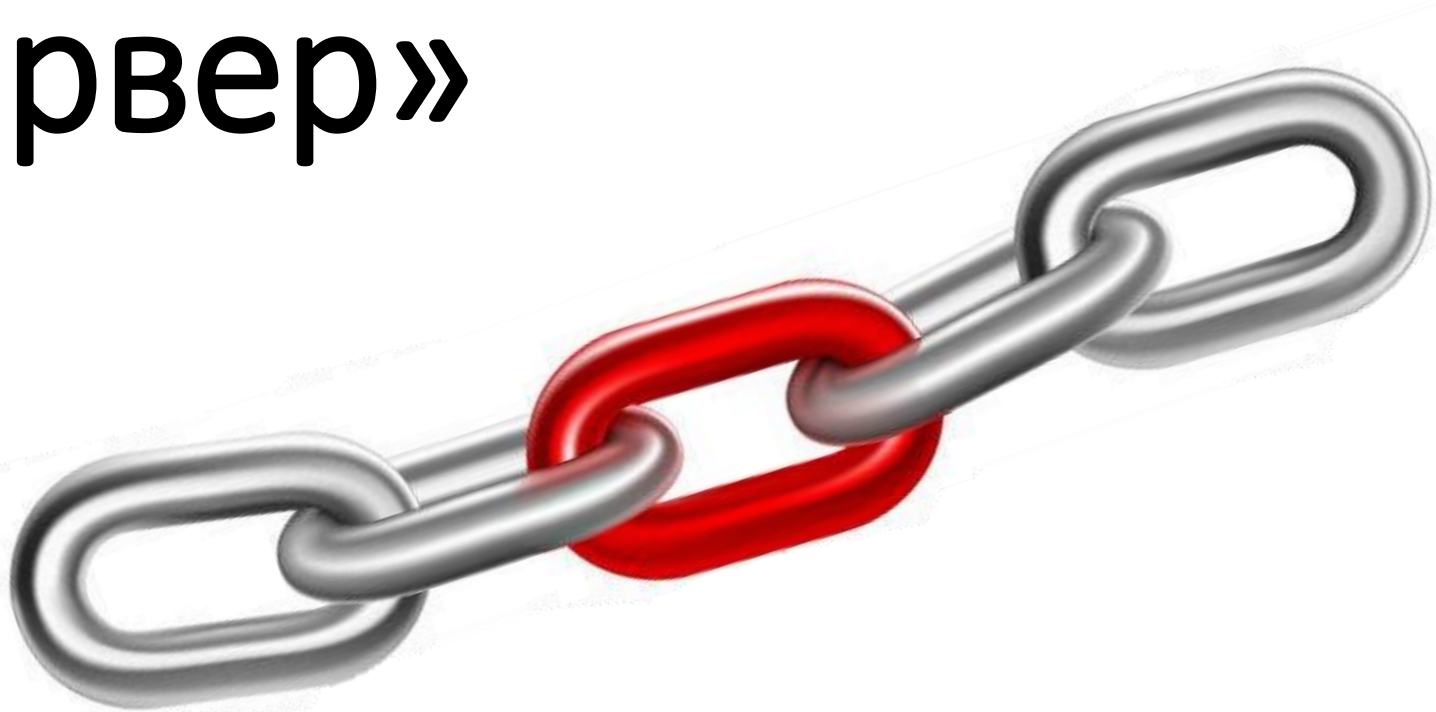
- гибкость создаваемых на ее основе ИС, позволяющих клиенту обрабатывать локальные и удаленные БД;
- при наличии механизмов координации соответствия копий система обладает высокой «живучестью»

Недостатки модели:

- высокие затраты при выполнении большого числа одинаковых приложений на компьютерах-клиентах



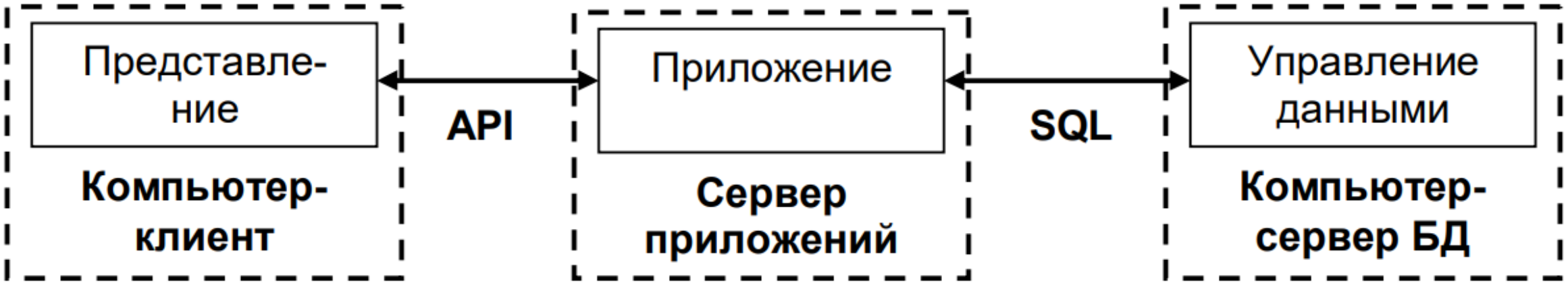
Трехзвенная модель архитектуры «клиент-сервер»



Трехзвенная модель распределения функций

Каждая из 3 функций приложения реализуется на отдельном компьютере.
Возможно распределение функций на большее число компьютеров.

Модель сервера приложений, или AS-модель (Application Server)



Трехзвенная модель распределения функций

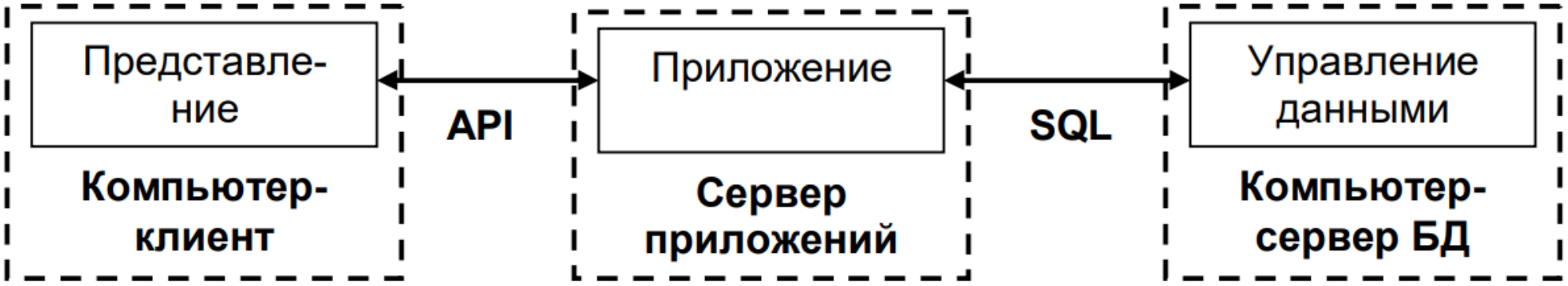
Модель сервера приложений, или AS-модель (Application Server)

Достоинство:

- гибкость и универсальность вследствие разделения функций приложения на три независимые составляющие.
  Часто эта модель более эффективна, чем двухзвенные.

Недостаток :

- более высокие затраты ресурсов компьютеров на обмен информацией между компонентами приложения, чем у двухзвенных моделей.



Стандарт ODBC

- При построении ИС типа клиент-сервер возникает проблема доступа со стороны СУБД или приложений, разработанных в одной среде, к данным, порожденным другой СУБД.

Стандартный интерфейс ODBC

(**Open Database Connectivity** — совместимость открытых баз данных)

Основное назначение:

обеспечение унифицированного доступа
к локальным и удаленным БД различных производителей.

Стандарт ODBC

- Интерфейс ODBC обеспечивает взаимную совместимость серверных и клиентских компонентов доступа к данным.

- Для реализации унифицированного доступа к различным СУБД, было введено понятие **драйвера ODBC** (динамически загружаемая библиотека).

- PostgreSQL ODBC Driver: psqlodbc

Стандарт ODBC

При построении ODBC-архитектура содержит четыре компонента:

1. Приложение вызывает функции ODBC для выполнения SQL-инструкций, получает и интерпретирует результаты;

2. Менеджер драйверов загружает ODBC-драйверы, когда этого требует приложение;

3. ODBC-драйверы обрабатывают вызовы функций ODBC, передают операторы SQL СУБД и возвращают результат в приложение;

4. Источник данных (data source) — объект, скрывающий СУБД, детали сетевого интерфейса, расположение и полное имя БД и т.д.