

- 
- 1. Понятие информационной системы (функции, состав). Определение базы данных. Определение СУБД. Этапы развития СУБД.



Понятие информационной системы (ИС)

Примеры информационных систем

- Банковские информационные системы,
- системы резервирования авиационных или железнодорожных билетов,
- системы резервирования мест в гостиницах и т.д.





Информационные системы

обеспечивают сбор, хранение, обработку, поиск и выдачу информации, необходимой в процессе решения задач из любой .области

Функции информационной системы



Поддержка надёжного хранения информации в памяти компьютера
Выполнение специфических для данного приложения преобразований информации и вычислений
Предоставление пользователям удобного интерфейса для поиска и выдачи информации

Поддержка надёжного хранения информации в памяти компьютера

Выполнение специфических для данного приложения преобразований информации и вычислений

Предоставление пользователям удобного интерфейса для поиска и выдачи информации

Состав ИС

ВС

- **вычислительная система**
- одна или несколько **баз данных**
- **СУБД**
- **набор прикладных программ.**

Вычислительная система (ВС) представляет собой совокупность взаимосвязанных и согласованно действующих ЭВМ и других устройств, обеспечивающих автоматизацию процессов приёма, обработки и выдачи информации потребителям.



Система управления базами данных (СУБД) - комплекс языковых и программных средств, предназначенный для создания, ведения и использования БД



База данных (БД) - совокупность специальным образом организованных данных, хранимых в памяти вычислительной системы и отображающих состояние объектов и их взаимосвязей в рассматриваемой предметной области.





Наборы прикладных программ могут создаваться в среде или вне среды СУБД - с помощью системы программирования, использующей средства доступа к БД



Этапы развития СУБД



История развития СУБД
насчитывает почти 50 лет. В
1968 году была введена в
эксплуатацию первая
промышленная СУБД система
IMS фирмы IBM.



IMS (IBM, 1968 г.)

IDMS (Cullinet, 1971 г.)

ADABAS (Software AG,
1969 г.)

ИНЭС (ВНИИСИ АН СССР,
1976 г.).



С конца 1990-х система поддерживает доступ на языке программирования Java, интерфейсы JDBC, обработку XML



Первый стандарт
ассоциации по языкам
обработки данных
(CODASYL) был
опубликован в 1975 году



В дальнейшее развитие теории баз данных большой вклад был сделан американским математиком Э. Ф. Коддом, который является создателем реляционной модели данных.

В 1981 году Э. Ф. Кодд получил за создание реляционной модели и реляционной алгебры престижную премию Тьюринга Американской ассоциации по вычислительной технике.



Первый этап развития СУБД связан с организацией БД на больших машинах типа IBM360/370.

БД хранились во **внешней памяти центральной ЭВМ**. Программы доступа к БД писались на различных языках. Интерактивный доступ обеспечивался с помощью консольных терминалов, которые служили устройствами ввода-вывода для центральной ЭВМ.



Эти системы можно было отнести к **системам распределенного доступа**, потому что база данных была централизованной, хранилась на устройствах внешней памяти одной центральной ЭВМ, а доступ к ней поддерживался от многих пользователей-задач.

Особенности этого этапа развития выражаются в следующем:

- Все СУБД базируются на мощных мультипрограммных операционных системах (MVS, SVM, RTE, OSRV, RSX, UNIX), поэтому в основном поддерживается работа с централизованной базой данных в режиме распределенного доступа.
- Функции управления распределением ресурсов в основном осуществляются операционной системой (ОС).
- Поддерживаются языки низкого уровня манипулирования данными, ориентированные на навигационные методы доступа к данным.



- Значительная роль отводится администрированию данных.

- Проводятся серьезные работы по обоснованию и формализации реляционной модели данных, и была создана первая система (System R), реализующая идеологию реляционной модели данных.

- Проводятся теоретические работы по оптимизации запросов и управлению распределенным доступом к централизованной БД, было введено понятие транзакции.

Появляются первые языки высокого уровня для работы с реляционной моделью данных. Однако отсутствуют стандарты для этих первых языков.



**Второй этап (появление ПК) -
настольные СУБД с
монопольным доступом.**

Характерные черты 2 этапа:

1) удобный пользовательский интерфейс

2) интерактивный режим работы с БД, как для описания БД, так и для проектирования запросов

3) развитый и удобный инструментарий для разработки готовых приложений без программирования



Многие из СУБД, относящихся к этому этапу не выдержали конкуренции и сейчас не поддерживаются. Оставшиеся в процессе развития стали приобретать черты многопользовательских СУБД, такие как механизмы совместного использования и защиты данных.

В качестве примера популярных сейчас настольных СУБД можно назвать **Microsoft Access** и **OpenOffice Base**.

Особенность:

Отсутствовали
инструментальные средства
администрирования БД

Недостатки:

**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 в этом случае эти функции должны были выполняться пользователем, требуя от него дополнительного контроля при вводе и изменении информации, хранящейся в БД.



Спрос на развитые удобные программы обработки данных заставлял поставщиков программного обеспечения поставлять все новые системы, которые принято называть настольными (desktop) СУБД.

Значительная конкуренция среди поставщиков заставляла совершенствовать эти системы, предлагая новые возможности, улучшая интерфейс и быстродействие систем, снижая их стоимость.

Наличие на рынке большого числа СУБД, выполняющих сходные функции, потребовало разработки методов экспорта - импорта данных для этих систем и открытия форматов хранения данных.



В этот период появлялись любители, которые вопреки здравому смыслу разрабатывали собственные СУБД, используя стандартные языки программирования.

Это был тупиковый вариант, потому что дальнейшее развитие показало, что перенести данные из нестандартных форматов в новые СУБД было гораздо труднее, а в некоторых случаях требовало таких трудозатрат, что легче было бы все разработать заново, но данные все равно надо было переносить на новую более перспективную СУБД.



Третий этап развития СУБД
связан с широким развитием
компьютерных сетей.

БД становится доступна
одновременно многим
пользователям. **Поэтому важной
является проблема
согласованности данных.**



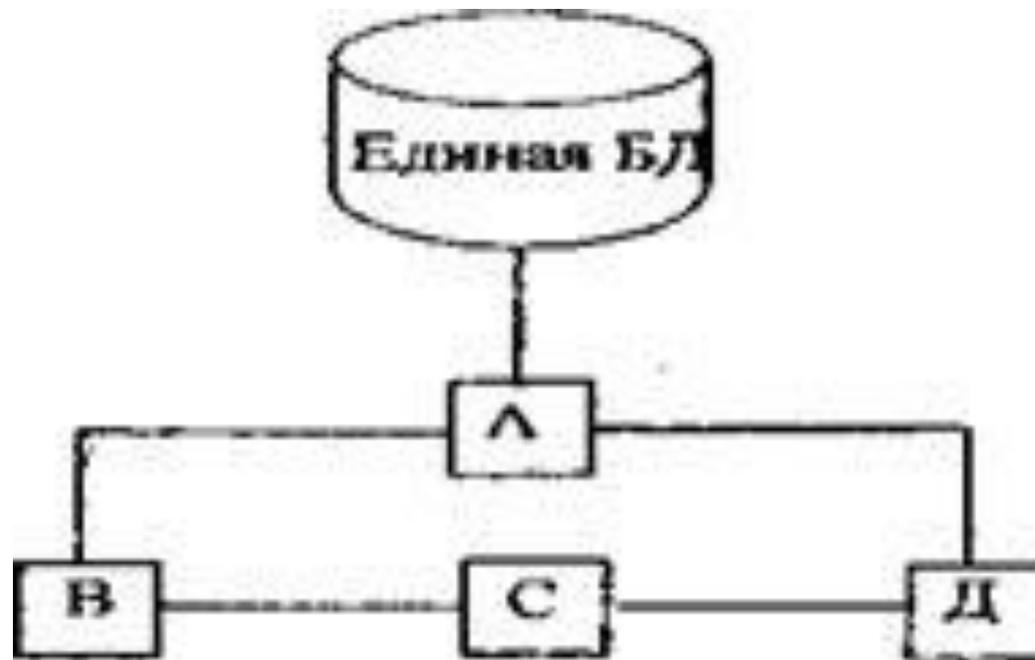
Задачи, связанные с **параллельной обработкой транзакций** –

последовательности операций над БД, переводящих ее из одного непротиворечивого состояния в другое непротиворечивое состояние.

Успешное решение этих задач приводит к появлению **распределенных БД и баз данных с распределенной обработкой**, позволяющих организовать параллельную обработку информации и поддержку целостности БД.

Системы распределенной обработки данных

Параллельный доступ нескольких пользователей к одной БД, расположенной на одном компьютере, соответствует режиму **распределенного доступа к централизованной БД**





Преимущество использования данного способа - простота поддержки базы данных в актуальном состоянии.

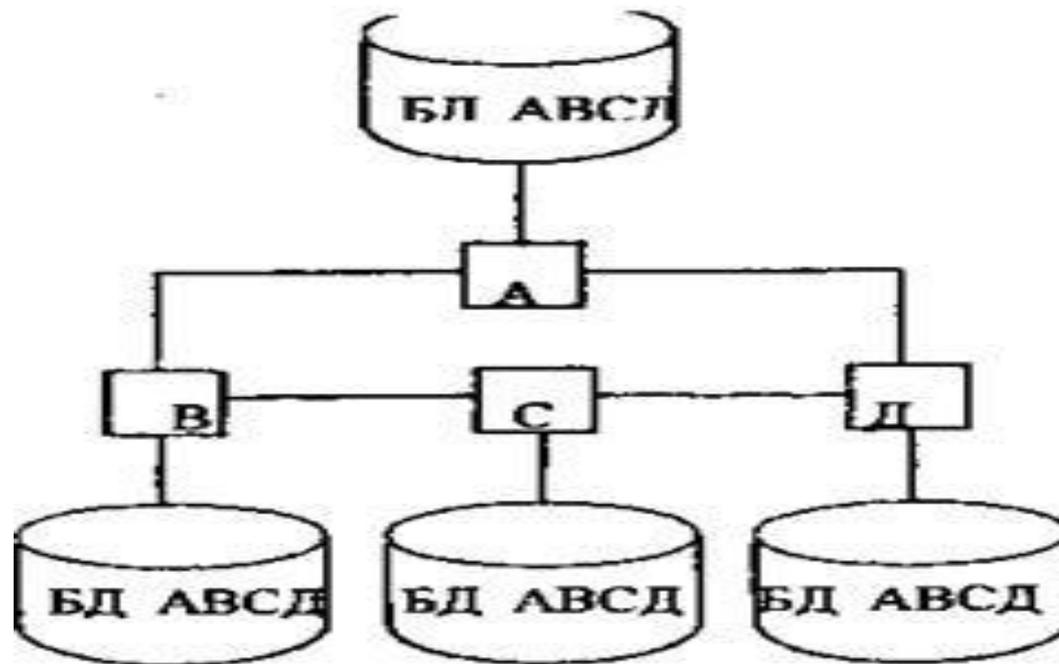
Недостатки

- ограниченность размера базы данных
- зависимость от размера памяти,
- все запросы производятся к единственному серверу с соответствующими затратами на стоимость связи и временную задержку

База данных может быть недоступной для удаленных пользователей при появлении ошибок связи и полностью выходит из строя при отказе центрального сервера.



Если БД распределена по нескольким компьютерам, расположенным в сети, и к ней возможен параллельный доступ, то мы имеем дело с параллельным доступом к распределенной БД. Такие системы называют **системами распределенных БД.**





Каждый клиент пользуется своей базой данных, которая может быть либо частью общей информационной базы, либо копией информационной базы данных в целом что приводит к ее дублированию для каждого клиента

Необходимость поддержки
многопользовательской работы с БД и
возможность децентрализованного
хранения данных потребовали
развития средств
администрирования БД с
реализацией общей концепции
средств защиты данных.



К третьему этапу относится разработка ряда стандартов языков описания и манипулирования данными начиная с **SQL89, SQL92, SQL99, SQL2003** и технологий по обмену данными между различными СУБД например, протокол ODBC.



К 3 этапу относится начало работ, связанных с концепцией **объектно-ориентированных БД** — (ООБД).



Представителями СУБД,
относящимся ко третьему этапу,
можно считать MS Access и все
современные серверы баз данных
ORACLE, MS SQL Server ,
Informix, IBM DB2, SQLBase и
другие современные серверы баз
данных, которых в настоящий
момент насчитывается несколько
десятков.



Большое влияние на СУБД оказало и развитие сети Интернет. При формировании веб-страниц в большинстве случаев задействуются СУБД и обслуживаемые ими БД.

Это привело к появлению ряда СУБД, чья популярность, в первую очередь, связана с их использованием при создании веб-приложений. Наиболее яркий пример – реляционная **СУБД MySQL.**



Объектные расширения
реализованы в трех
ведущих РСУБД – **Oracle,**
Informix и DB.

Принципы подхода к объектному расширению реляционной модели

значения атрибутов отношений не обязательно являются атомарными

значениями атрибутов отношений могут быть объекты

при построении таблиц (классов) может использоваться механизм наследования

классы включают операции



Четвертый этап характеризуется использованием технологии доступа к данным — интранет.

Для работы с удаленной базой данных используется стандартный браузер

- 
-
- Сети интранет построены на том же аппаратно-программном обеспечении, принципах и протоколах, что и сеть Интернет.
 - В общем случае под сетью интранет понимают выделенную часть сети Интернет, в которой выполняется Web-приложение



При этом встроенный в загружаемые пользователем HTML-страницы код, отслеживает все действия пользователя и транслирует их в низкоуровневые SQL-запросы к базе данных, выполняя, таким образом, ту работу, которой в технологии клиент-сервер занимается клиентская программа.



Простые задачи обработки данных, не связанные со сложными алгоритмами, требующими согласованного изменения данных во многих взаимосвязанных объектах, достаточно просто и эффективно могут быть построены по данной архитектуре.

В этом случае для подключения нового пользователя к возможности использовать данную задачу не требуется установка дополнительного клиентского программного обеспечения.



Наиболее именитый
производитель серверных СУБД
– это корпорация **Oracle**,
выпустившая в 1979 г. первую
коммерческую реляционную
СУБД Oracle v2 и с тех пор
являющаяся ключевым
производителем в области
серверов БД.

Существенное место на рынке занимает корпорация **IBM**, выпускающая реляционную СУБД DB2 и иерархическую СУБД IMS. Приобретя в 2001 г. подразделение корпорации **Informix**, IBM добавила в свою линейку продуктов одноименную СУБД.



Заметное место занимает корпорация **Microsoft** с ее серверным продуктом **MS SQL Server** и настольной СУБД **Access**, входящей в пакет Microsoft Office.

Несмотря на то, что MS SQL Server выпускается только для ОС семейства Windows, популярность данной платформы, поддержка в средствах разработки Microsoft и широкие возможности самой СУБД привели к ее широкому распространению.



Основанная в 1984 г. компания **Sybase** может быть также названа одним из пионеров в области разработки реляционных СУБД. В конце 1980-х – начале 1990-х гг. Sybase вела разработку SQL Server в альянсе с Microsoft, но в дальнейшем продукты стали независимыми. На сегодняшний день в линейке продуктов **Sybase** есть реляционный сервер БД Adaptive Server Enterprise, встраиваемая реляционная СУБД SQL Anywhere и нереляционная СУБД с "поколоночным" хранением данных Sybase IQ, предназначенная для задач аналитической обработки данных и построения хранилищ данных.

Среди приверженцев свободно распространяемого ПО широкую популярность приобрела СУБД **MySQL**, изначально разрабатывавшаяся созданной в Швеции компанией **MySQL AB**. В настоящее время у MySQL лидирующие позиции в качестве СУБД, используемой в области веб-разработки. В 2008 г. компания MySQL AB была приобретена Sun Microsystems, а в 2010 г. уже саму Sun приобрела **Oracle**. Сейчас выпускаются как коммерческие, так и бесплатно распространяемая версия MySQL (MySQL Community Edition).



Каждая СУБД реализует одну из моделей баз данных для логической структуризации используемых данных. Эти модели являются главным критерием того, как будет работать и управлять информацией приложение.

Существует несколько таких моделей, среди которых самой популярной является реляционная.



Хотя реляционная модель является мощной и гибкой, есть ситуации, решения которых она предложить не может.

Тут на помощь придёт сравнительно новая модель, называемая NoSQL. Из-за того, что эти системы не используют строгую структуризацию данных, они предлагают большую свободу действий при обработке информации.



Безмодельный (NoSQL) подход

NoSQL-способ структуризации данных заключается в избавлении от ограничений при хранении и использовании информации.

Они не основываются на одной модели, а каждая база данных в зависимости от целей использует различные модели.

Разница между SQL- и NoSQL-подходами:

Структура и тип хранящихся данных: SQL/реляционные базы данных требуют наличия однозначно определённой структуры хранения данных, а NoSQL базы данных таких ограничений не ставят.

Запросы: РСУБД реализуют SQL-стандарты, поэтому из них можно получать данные при помощи языка SQL. Каждая NoSQL база данных реализует свой способ работы с данными.



Надёжность: когда речь заходит о надёжности, SQL базы данных однозначно впереди.



Существует довольно много различных моделей и функциональных систем для NoSQL баз данных:

Хранилище ключ-значение - **Redis, MemcacheDB** и т.д. (обычно хранят данные в памяти)

Распределённое хранилище - **Cassandra, HBase** и т.д. (предназначены для очень больших объёмов данных).

Документо-ориентированные СУБД - **MongoDB, Couchbase** и т.д. (предназначены для хранения иерархических структур данных – документов)



2. Архитектура информационной системы

Архитектура файл-сервер.

составными компонентами файл-серверной архитектуры являются

- **Файловый сервер,**
- **клиентские места,**
- **сетевая инфраструктура.**



Файловый сервер
функционирует под управлением
специализированного
программного обеспечения -
сетевой операционной системы.



Как следует из самого термина файл-сервер, весь обмен между клиентскими рабочими местами и сервером осуществляется на уровне файлов.

При этом сервер не обладает никакой информацией о содержимом файлов, поэтому всю обработку данных производит клиент.



Поскольку в архитектуре файл-сервер впервые появляются клиент и сервер, то, строго говоря, в ней тоже используются клиент-серверные технологии, если брать самое широкое значение этого термина.

Однако, из-за того, что взаимодействие клиента и сервера осуществляется на слишком низком уровне - уровне файлов, **в информационных приложениях, созданных в этой архитектуре, вся обработка данных ведется на клиенте.**

Особенности:

- ~~Централизованное хранение данных.~~
- По запросам пользователей **файлы** **базы** **данных** передаются на персональные компьютеры (ПК), где и производится их обработка.

Недостатки архитектуры:

- передача избыточных данных: вне зависимости от того, сколько записей из базы данных требуется пользователю, файлы базы данных передаются целиком.
- высокая интенсивность передачи обрабатываемых данных.

- 
- потенциальные проблемы с сохранностью данных при одновременном внесении изменений с разных мест.
 - **принципиальная невозможность гарантировать со стороны сервера целостность информации в базе данных, поскольку их обработка осуществляется каждым клиентом по отдельности.**

Архитектура клиент-сервер

составными компонентами клиент-серверной архитектуры являются

- **сервер,**
- **клиентские места,**
- **сетевая инфраструктура.**



Однако, в отличие от предыдущего случая, сервер здесь является уже не сервером файлов, а **сервером баз данных.**



Информационная система в архитектуре клиент-сервер создаётся, поддерживается и функционирует под управлением сервера БД, например, **Microsoft SQL Server** или **Oracle Server**.



Сервер базы данных обеспечивает выполнение основного объема обработки данных. Формируемые пользователем или приложением запросы поступают к серверу БД в виде инструкций языка SQL. Сервер базы данных выполняет поиск и извлечение нужных данных, которые затем передаются на компьютер пользователя.

Достоинством такого подхода в сравнении предыдущим является заметно меньший объем передаваемых данных.

Основными функциями сервера БД являются:

- хранение и резервное копирование данных;
- выполнение пользовательских запросов на выборку и модификацию данных;
- поддержка ссылочной целостности данных согласно определённым в базе данных правилам;
- реализация бизнес-правил информационной системы;
- обеспечение авторизованного доступа к данным на основе проверки прав и привилегий пользователей;
- протоколирование операций и ведение журнала транзакций.



Важнейшим преимуществом архитектуры клиент-сервер является возможность хранения бизнес-правил на сервере, что позволяет избежать дублирования кода в различных клиентских приложениях, использующих общую базу данных.

3. Классификация СУБД

В качестве основных классификационных признаков можно использовать следующие:

- **вид программы,**
- **характер использования,**
- **модель данных.**





К СУБД относятся следующие основные виды программ:

- **полнофункциональные СУБД;**
- **серверы БД**

Полнофункциональные СУБД (ПФСУБД)

представляют собой традиционные СУБД, которые сначала появились для больших машин, затем для мини-машин и для ПЭВМ.

К ПФСУБД относятся, например, такие пакеты как: **Clarion Database Developer, DataBase, Dataplex, dBase IV, Microsoft Access, Microsoft FoxPro, Paradox.**



Обычно ПФСУБД имеют развитый интерфейс, позволяющий с помощью команд меню создавать и модифицировать структуры таблиц, вводить данные, формировать запросы, разрабатывать отчеты, выводить их на печать и т. п.



Для создания запросов и отчетов не обязательно программирование, а можно пользоваться языком QBE.

Многие ПФСУБД включают средства программирования для профессиональных разработчиков.

Серверы БД

Серверы БД реализуют функции управления данными, запрашиваемыми другими (клиентскими) программами обычно с помощью операторов SQL.

Примеры серверов БД:

- **MS SQL Server,**
- InterBase ,
- **Oracle,**
- **IBM DB2.**
- Sybase,
- Informix,
- Ingress,
- Postgres,
- Cache,
- Firebird,
- Teradata и т. д.



Среди них выделяются лидеры,
занимающие вместе более 90%
мирового рынка СУБД:

**Oracle, Microsoft SQL Server и
IBM DB2.**



2. По **характеру использования СУБД** делят на персональные и многопользовательские.



К персональным СУБД относятся

- **Visual FoxPro,**
- **Paradox,**
- **Clipper,**
- **dBase,**
- **MS Access и др.**



Многопользовательские системы включают в себя сервер БД и клиентскую часть и, как правило, могут работать в неоднородной вычислительной среде (с разными типами ЭВМ и операционными системами).

К многопользовательским СУБД относятся все серверы БД.



- 
- По используемой **модели данных СУБД (как и БД)**,
разделяют на
 - **иерархические,**
 - **сетевые,**
 - **реляционные,**
 - **объектно-ориентированные.**
- Некоторые СУБД могут одновременно поддерживать несколько моделей данных.



Большая часть СУБД поддерживает реляционную модель данных.

Однако она не вполне удовлетворяет сегодняшним требованиям, предъявляемым к скорости обработки запросов.

Крупнейшие разработчики СУБД встраивают в свои продукты поддержку объектной модели программирования.



Компании IBM и Oracle переработали ядра своих СУБД (**DB2** и **ORACLE**) и добавили в него объектные свойства.

По-видимому, рынок корпоративных систем в ближайшее время останется за гибридными объектно-реляционными СУБД.



Наиболее полно современному
состоянию вычислительных
систем соответствуют объектные
базы данных, представленные,
например, постреляционными
СУБД **Cache, MongoDB**



По соображениям совместимости с прежними наработками, лидеры индустрии СУБД предлагают смешанный подход - объектно-реляционный.



Известны прогнозы, предсказывающие, что на смену универсальным коммерческим СУБД, которые становятся все более **громоздкими, дорогостоящими и медленно работающими**, придут специализированные СУБД, предназначенные для решения ограниченного круга задач и лишённые этих недостатков.



Однако эти прогнозы, как правило, являются несостоятельными. По мнению специалистов в области разработки информационных систем пользователи выбирают СУБД, ориентируясь на совокупность параметров. Обычно учитываются такие характеристики, как **обеспечение высокой надежности работы, безопасность, простота управления, простота разработки, возможность работы с большими БД.**



Важна и распространённость
данной СУБД в стране, наличие
обученных специалистов
(администраторов, разработчиков),
наличие большого числа удачных
внедрений СУБД.

По этим параметрам лидерами
остаются **MS SQL Server** и **Oracle**.



По прогнозам специалистов в ближайшие 5-7 лет универсальные коммерческие СУБД будут лидировать на рынке программных продуктов. При этом необходимо учитывать, что появление важных новых возможностей у одного из производителей заставляет остальных также их реализовывать.

Так анализ новых возможностей последних версий MS SQL Server показывает, что большинство из них были реализованы у Oracle недавно или несколько лет назад.



Так, настоящее время и в Microsoft SQL Server и в Oracle поддерживаются развитые средства оптимизации запросов, режим неблокирующего чтения.

Как в Oracle, так и в MS SQL Server встроены функции файловой системы, которой можно пользоваться для хранения обычных файлов.

Средства поддержки мультимедийных типов данных (геоинформационных, аудио- и видеоданных) первоначально были встроены СУБД компаний Oracle и DB2, а, начиная с версии SQL Server 2008, Microsoft также включил в свою систему поддержку геоинформационных данных.

NoSQL подход

Причиной появления NoSQL СУБД в первую очередь стало резкое увеличение объемов информационных хранилищ и усложнение связей между документами

Информация в базах данных перестала быть изолированной и стала менее структурированной. Иногда вообще невозможно жестко описать структуру хранимой информации.



В NoSQL базах в отличие от реляционных структура данных не регламентирована (или слабо типизированна) — в отдельной строке или документе можно добавить произвольное поле без предварительного декларативного изменения структуры всей таблицы

Разница между SQL- и NoSQL-подходами:

Структура и тип хранящихся данных: SQL/реляционные базы данных требуют наличия однозначно определённой структуры хранения данных, а NoSQL базы данных таких ограничений не ставят.

Запросы: РСУБД реализуют SQL-стандарты, поэтому из них можно получать данные при помощи языка SQL. Каждая NoSQL база данных реализует свой способ работы с данными.



Примеры СУБД

**MemcacheDB, Redis, Amazon
DynamoDB** - хранилища
«ключ-значение»

**CouchDB, Couchbase,
MarkLogic, MongoDB, eXist,
Berkeley DB XML** - документо-
ориентированные СУБД



**4. Основные категории
пользователей:
администраторы БД,

конечные пользователи.
Основные функции
администратора БД.**



○ **Конечные пользователи** –
категория пользователей, для
которых и создаётся БД.

Это могут быть случайные пользователи, обращающиеся к БД время от времени за получением некоторой информации. Регулярными пользователями могут быть сотрудники фирмы, работающие со специально разработанными для них программами.

- Администратор БД. На начальной стадии разработки отвечает за оптимальную организацию БД для одновременной работы множества конечных пользователей.

На стадии эксплуатации администратор отвечает за корректность работы системы в многопользовательском режиме.

○ **Разработчики и администраторы приложений.**

Эта группа пользователей функционирует во время проектирования, создания и реорганизации БД. Администраторы приложений координируют работу разработчиков. Разработчики конкретных приложений работают с той частью информации из БД, которая требуется для конкретного приложения.



5. Особенности жизненного цикла систем баз данных. Фазы жизненного цикла. Фаза анализа и проектирования. Фаза функционирования. Работы, выполняемые на каждой из этих фаз.



Жизненный цикл базы данных - совокупность этапов, которые проходит база данных на своём пути от создания до окончания использования.

- 
1. Исследование и анализ проблемы, для решения которой создаётся база данных.
 2. Построение модели данных.
 3. Нормализация полученных моделей.

- 
-
4. Выбор физического способа хранения и эксплуатации базы данных.
 5. Проектирование входных и выходных форм.
 6. Разработка интерфейса приложения.
 7. Функциональное наполнение приложения.

- 
8. Отладка: проверка на корректность работы функционального наполнения системы.
 9. Тестирование: тест на корректность ввода-вывода данных, тест на максимальное количество активных сессий и т. д.
 10. Ввод в эксплуатацию: отладка инфраструктуры, обучение пользователей и персонала.



11. При необходимости - добавление входных и выходных форм и дополнительной функциональности.

12. Вывод из эксплуатации: перенос данных в новую СУБД.



6. Модели клиент-сервер в технологии БД.

**Структура типового интерактивного приложения, работающего с БД.
Двухуровневые модели.
Достоинства и недостатки**



Реализация архитектуры клиент - сервер, применительно к разработке БД позволяет более полно использовать ресурсы сети. Нагрузка равномерно распределяется между компьютером сервером и компьютером клиентом, который также как и сервер обладает собственными ресурсами.



Основной принцип технологии клиент – сервер применительно к технологии БД заключается в разделении функций стандартного интерактивного приложения на следующие группы:

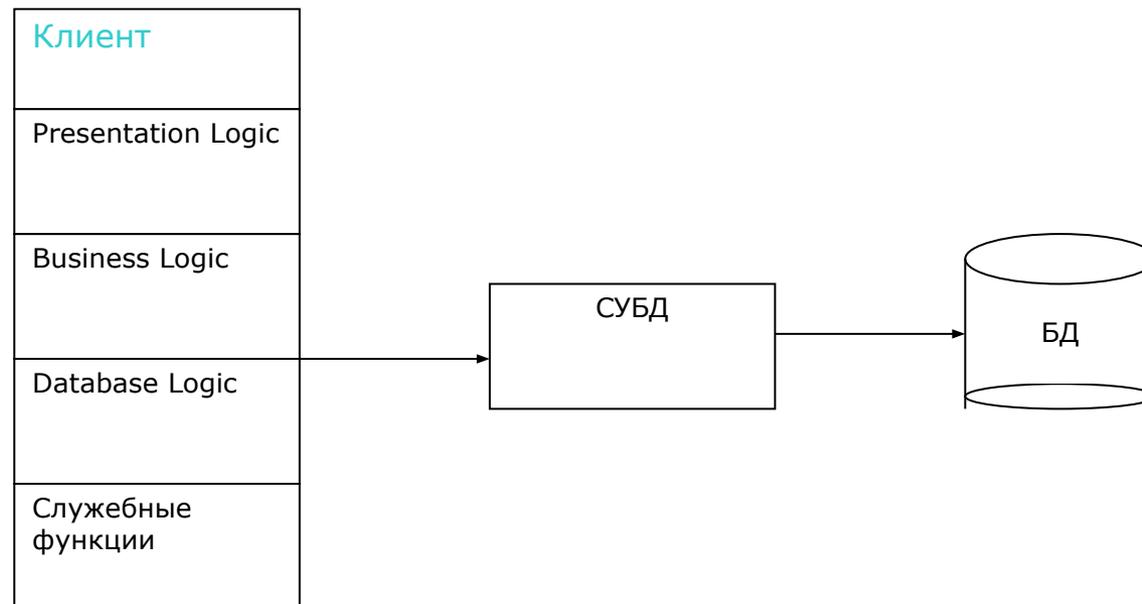
- Функции ввода и отображения данных.
-

- Прикладные функции, определяющие основные алгоритмы решения задач приложения.

- Функции обработки данных внутри приложения.

- Служебные функции, играющие роль связок между функциями первых 3-х групп.

Структура типового интерактивного приложения, работающего с БД



Презентационная логика – это часть приложения, определяющая то, что пользователь видит на экране. Сюда относятся, интерфейсные экранные формы, а также все, что выводится пользователю на экран, как результаты решения промежуточных задач или справочная информация.

Основными задачами презентационной логики являются:

- формирование экранных изображений;
- управление экраном;
- обработка движений мыши и нажатия клавиш клавиатуры.



Бизнес- логика или логика

приложений -это часть кода приложения, которая определяет собственно алгоритмы решения задач приложения. Обычно этот код пишется с помощью различных языков программирования





Логика обработки данных - это часть кода приложения, которая связана с обработкой данных внутри приложения. Данными управляет собственно СУБД.

Для обеспечения доступа к данным используются язык запросов и средства манипулирования данными языка SQL.



- 
-
- **Процессор управления данными** – это собственно СУБД, которая обеспечивает хранение и управление БД.



В централизованной архитектуре эти функции располагаются в единой среде и комбинируются внутри исполняемой программы.

В децентрализованной архитектуре эти задачи могут быть по-разному распределены между серверным и клиентским процессами.



В зависимости от характера распределения функций можно выделить следующие модели распределений:

- **удаленная презентация;** (вся презентация на клиенте – все остальное на сервере)

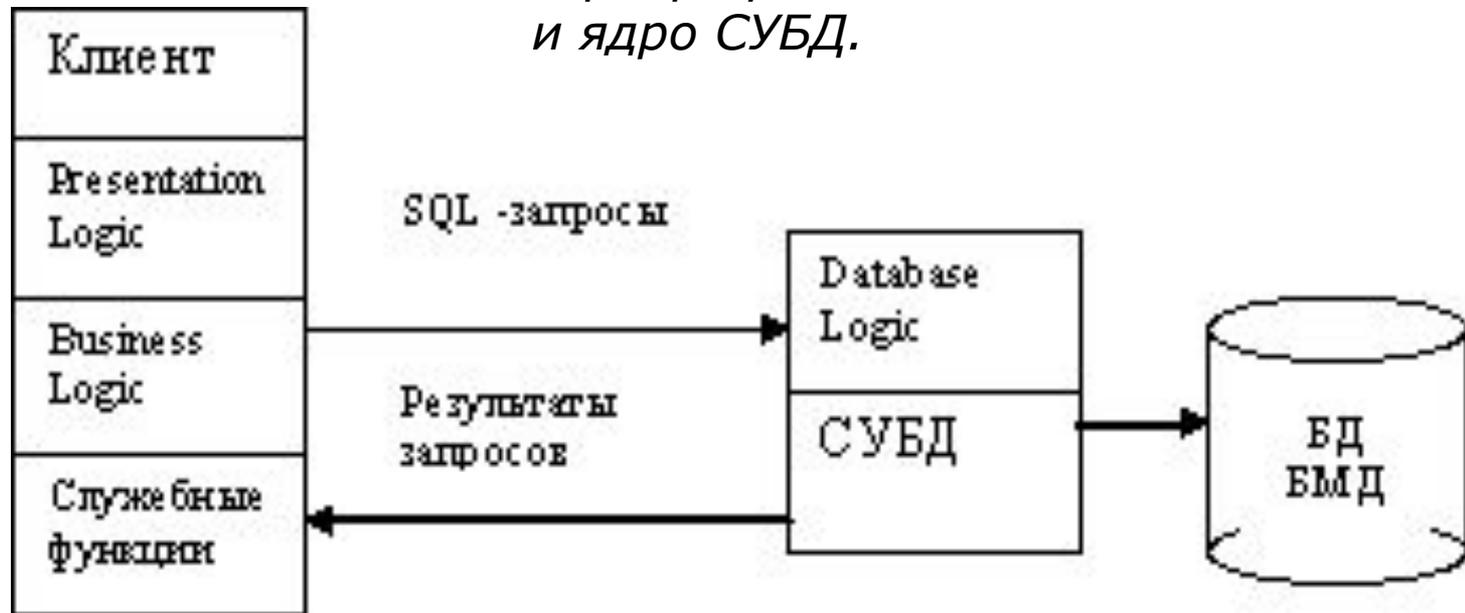
- 
- **распределенная бизнес-логика;**
(презентация и часть бизнес-логики на клиенте – остальное на сервере)
 - **распределенное управление данными;** (презентация, бизнес-логика, и часть управления данными на клиенте);
 - **Удаленный доступ к данным**
(презентационная и бизнес-логика на клиенте - остальное на сервере).

Двухуровневые модели

- Двухуровневая модель предполагает распределение всех указанных функций между 2-мя процессами, которые выполняются на 2-х платформах – клиенте и сервере.

Модель удаленного доступа к данным (RDA)

В этой модели презентационная логика и бизнес-логика располагаются на клиенте. На сервере располагается база данных и ядро СУБД.



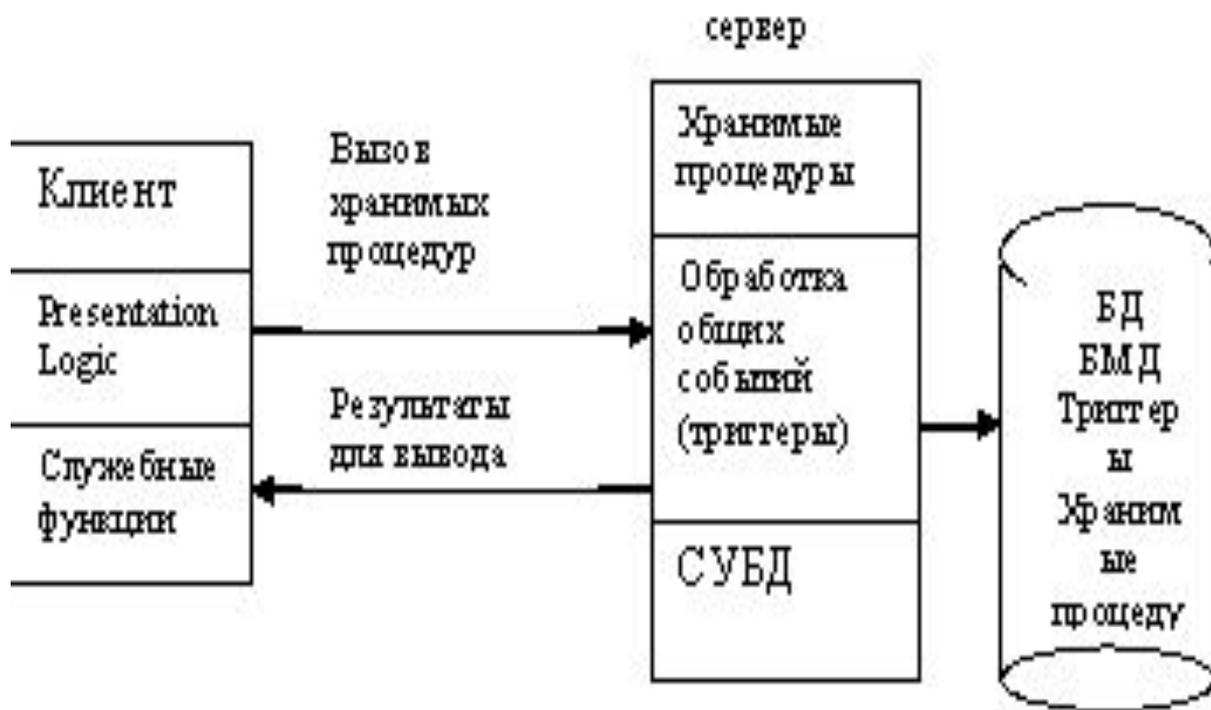
Достоинства:

- **наличие большого числа готовых СУБД, имеющих SQL-интерфейсы и инструментальных средств, обеспечивающих создание клиентских приложений.**
- **Резкое уменьшение загрузки сети, т.к. по сети передаются запросы SQL, в ответ на запросы клиент получает не блоки файлов, а только данные, релевантные запросу.**
- **унификация интерфейса клиент-сервер, стандартом при общении клиента и сервера становится язык SQL.**

Недостатки:

- достаточно высокая загрузка системы передачи данных, вследствие того, что вся логика сосредоточена в приложении, а обрабатываемые данные расположены на удаленном узле.
- При повторении аналогичных функций в разных приложениях **код соответствующей бизнес-логики должен быть повторен для каждого клиентского приложения.**
- Системы **неудобны с точки зрения модификации и сопровождения.** Даже при незначительном изменении функций системы требуется переделка всей прикладной части.

Модель сервера БД





Модель сервера БД отличается тем, что функции компьютера клиента ограничиваются представлением информации, в то время как прикладные функции обеспечиваются приложением, находящимся на компьютере-сервере.

Эта модель является **более технологичной**, чем модель удалённого доступа к данным.

- 
- Основу данной модели составляет механизм хранимых процедур как средство программирования SQL сервера,
 - механизм **триггеров** как механизм отслеживания текущего состояния информационного хранилища,
 - механизм **ограничений на пользовательские типы данных**, который называется механизмом поддержки доменной структуры.



Централизованнный контроль в модели сервера БД выполняется с использованием механизма **триггеров.**



Триггеры также являются частью БД.

Термин триггер взят из электроники и отражает механизм отслеживания специальных событий, которые связаны с состоянием БД. Триггер является как бы некоторым тумблером, который срабатывает при возникновении некоторого события в БД.

Ядро СУБД проводит мониторинг всех событий, которые вызывают созданные и описанные триггеры в БД, и при наступлении соответствующего события сервер запускает соответствующий триггер. Триггер представляет собой некоторую программу, которая выполняется над БД. Триггеры могут вызывать хранимые процедуры.



В данной модели сервер является
активным, потому что не только клиент, но и сам сервер, используя механизм триггеров, может быть инициатором обработки данных в БД.



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.



Достоинства модели - ВОЗМОЖНОСТЬ
хорошего централизованного
администрирования приложений на
этапах разработки, сопровождения и
модификации, а также **эффективное**
использование вычислительных и
коммуникационных ресурсов.



Один из **недостатков** модели связан с ограничениями средств разработки хранимых процедур. Основное ограничение – привязка операторов хранимых процедур к используемой СУБД.

Язык написания хранимых процедур является процедурным расширением языка SQL.

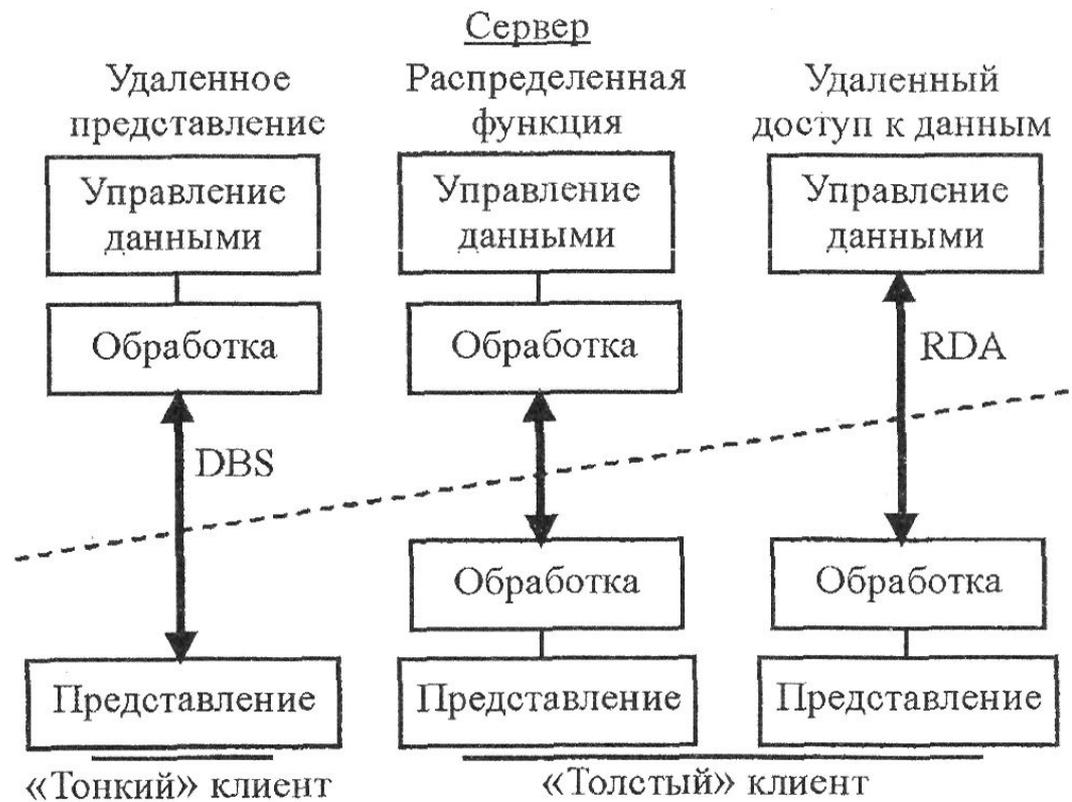
Другой **недостаток** – очень большая загрузка сервера.



Если большая часть бизнес логики приложения _____ перенесена на на сервер, то требования к клиентам в этой модели резко уменьшаются.

Иногда такую модель называют моделью с **тонким клиентом**, в отличие от предыдущих, где на клиента возлагались гораздо более серьезные задачи.

Модели архитектуры клиент-сервер



Модель распределенной БД

Распределенная база данных предполагает хранение данных на нескольких узлах сети, обработку данных и их передачу между этими узлами в процессе выполнения запросов.

Разбиение данных в распределенной базе данных может достигаться путем **хранения различных таблиц на разных компьютерах** или **хранения разных фрагментов одной таблицы на разных компьютерах**.



Любой узел способен независимо обрабатывать запросы пользователей, требующие доступа к локально сохраняемым данным (т. е. каждый узел обладает определенной степенью автономности), а также способен обрабатывать данные, сохраняемые на других компьютерах сети.

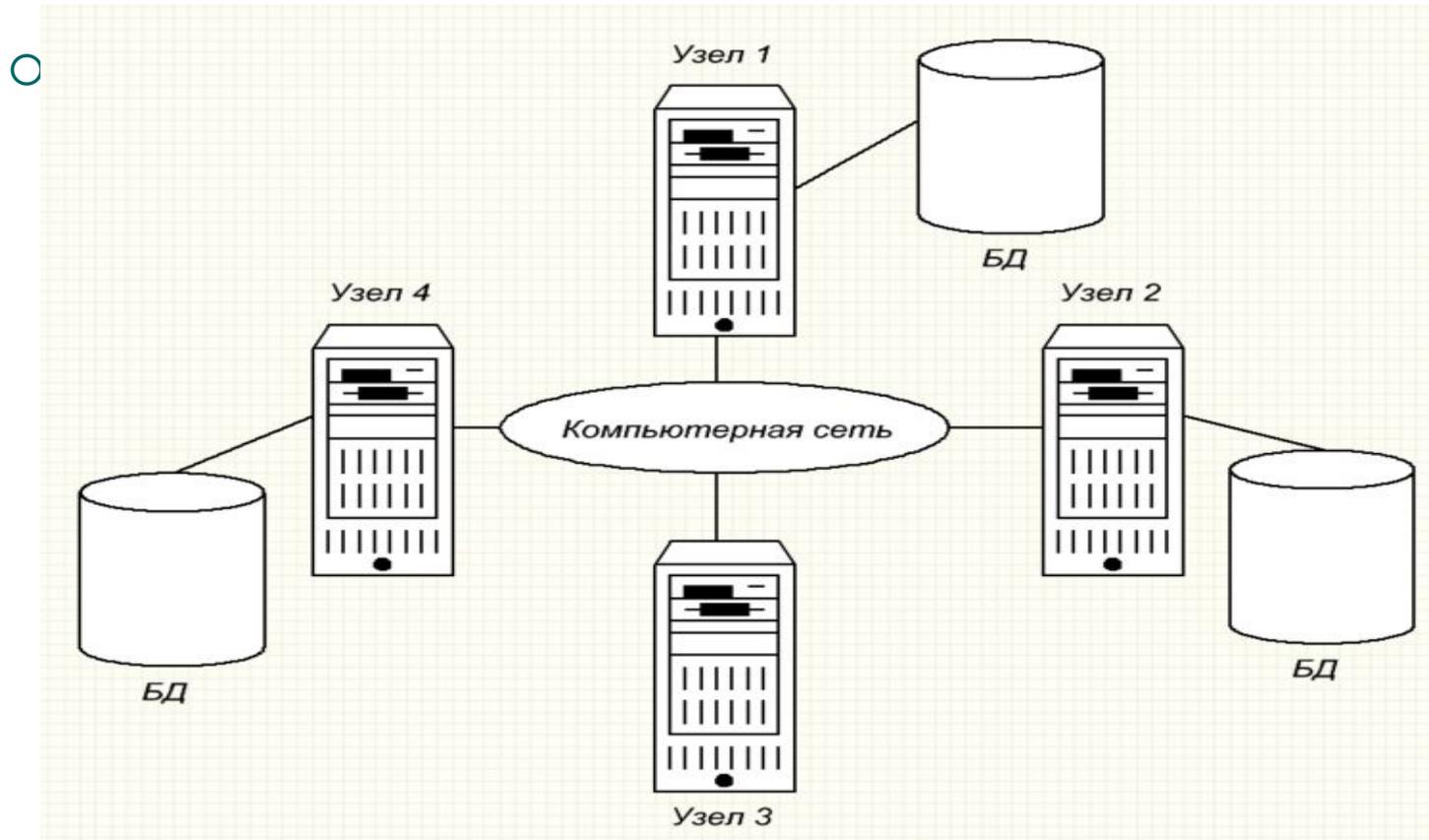


Для пользователя (или прикладной программы) не должно иметь значения, каким образом распределены данные между компьютерами.

Работа с распределенной базой данных должна осуществляться так же, как и с централизованной. Взаимосвязь обеих БД может быть 2-х разновидностей:

- а) в локальной и удаленной базах хранятся отдельные части единой БД;
- б) локальная и удаленная БД являются синхронизируемыми друг с другом копиями.

Топология распределенной БД



Преимущества распределенных баз данных

Данные также обычно распределены, поскольку каждая организационная единица на предприятии создает и обрабатывает собственные данные, относящиеся к ее деятельности.

Распределенная система позволяет структуре базы данных отображать структуру предприятия – локальные данные могут храниться локально, в соответствии с логической принадлежностью, тогда как к удаленным данным доступ может осуществляться по мере необходимости.

Недостатки:

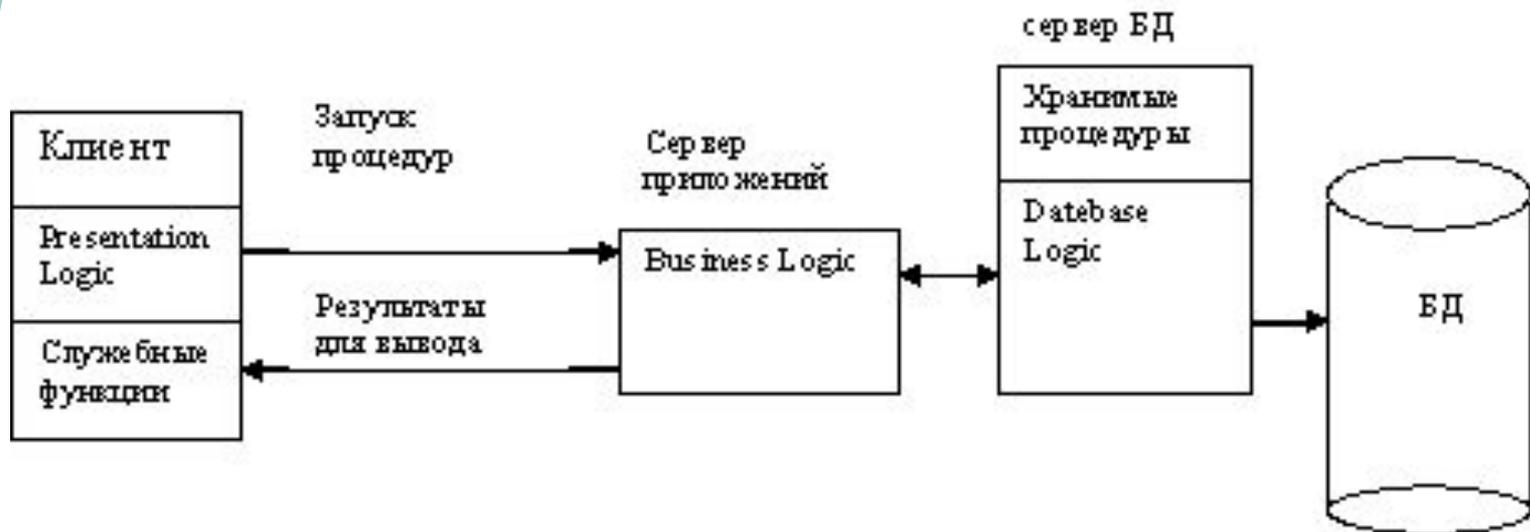
- повышенная сложность распределенных систем, по крайней мере, с технической точки зрения. Для того чтобы скрыть от пользователя сложность системы, требуется весьма тщательная ее проработка.



Модель сервера приложений

Эта модель является расширением 2-хуровневой модели и в ней вводится дополнительный промежуточный уровень между клиентом и сервером.

Этот промежуточный уровень содержит один или несколько серверов приложений.





В этой модели компоненты делятся между тремя исполнителями:

Клиент обеспечивает логику представления, включая графический пользовательский интерфейс;

клиент может запускать локальный код приложения клиента, который может содержать обращения к локальной БД, находящейся на компьютере- клиенте.



Серверы БД в этой модели занимаются исключительно функциями СУБД: обеспечивают создания и ведения БД, обеспечивают функции хранилищ БД, кроме того, на них возлагаются функции создания резервных копий, восстановления БД после сбоев, управления выполнением транзакций.