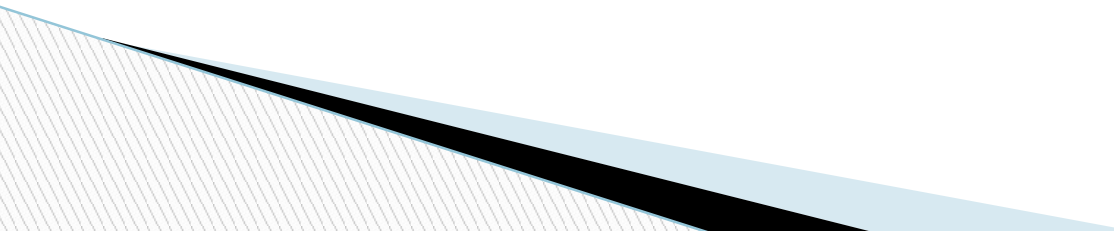


Базы данных

Основные понятия

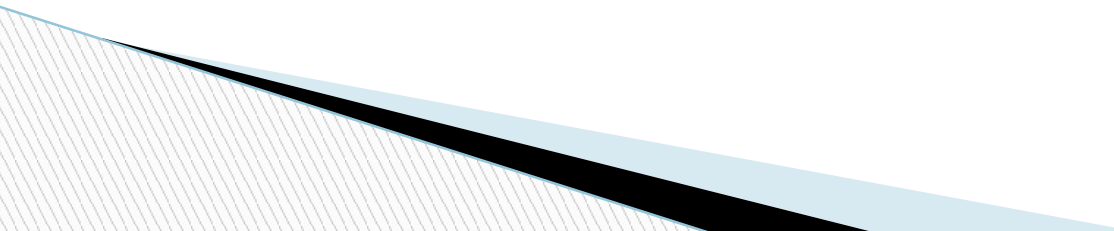


Банки данных имеют:

- вычислительную систему;
 - систему управления базами данных (СУБД);
 - одну или несколько баз данных (БД);
 - набор прикладных программ (приложений БД).
- 

▣ *База данных (БД)* обеспечивает хранение информации, а также удобный и быстрый доступ к данным. Она представляет собой совокупность данных различного характера, организованных по определенным правилам.

Информация в БД должна быть:

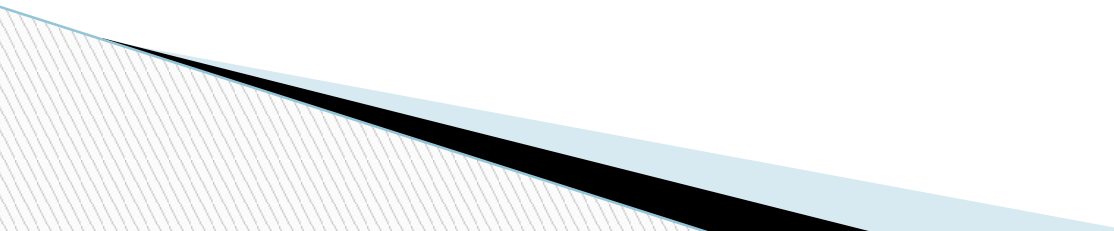
- непротиворечивой;
 - неизбыточной;
 - целостной.
- 

Система управления базой данных (СУБД)

- ▣ Это совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД.
- ▣ По характеру применения СУБД разделяют на **персональные** и **многопользовательские**.

- ▣ *Персональные СУБД* обеспечивают возможность создания локальных БД, работающих на одном компьютере. К персональным СУБД относятся Paradox, dBase, FoxPro, Access и др.
- ▣ *Многопользовательские СУБД* позволяют создавать информационные системы, функционирующие в архитектуре "клиент-сервер". Наиболее известными многопользовательскими СУБД являются Oracle, Informix, SyBase, Microsoft SQL Server, InterBase.

В состав языковых средств современных СУБД входят:

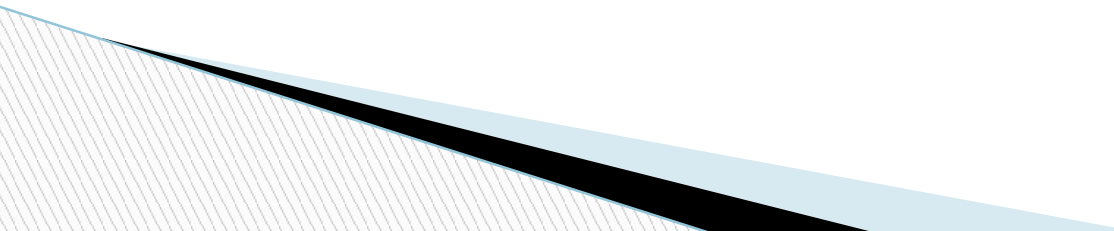
- ▣ Язык описания данных, предназначенный для описания логической структуры данных;
 - ▣ язык манипулирования данными, обеспечивающий выполнение основных операций над данными - ввод, модификацию и выборку;
 - ▣ язык структурированных запросов (SQL, Structured Query Language), обеспечивающий управление структурой БД и манипулирование данными, а также являющийся стандартным средством доступа к удаленным БД;
 - ▣ язык запросов по образцу (QBE, Query By Example), обеспечивающий визуальное конструирование запросов к БД.
- 

▣ *Прикладные программы*, или *приложения*, служат для обработки данных, содержащихся в БД. Пользователь осуществляет управление БД и работу с ее данными именно с помощью приложений, которые также называют *приложениями БД*.

Модели данных

- База данных содержит данные, используемые какой-либо прикладной информационной системой (например, системами «Сирена» или «Экспресс» продажи авиа- и железнодорожных билетов).

В зависимости от вида организации данных различают следующие основные модели представления данных в базе:

- иерархическую;
 - сетевую;
 - реляционную;
 - объектно-ориентированную.
- 

- В *иерархической* модели данные представляются в виде древовидной (иерархической) структуры. Подобная организация данных удобна для работы с иерархически упорядоченной информацией, однако при оперировании данными со сложными логическими связями иерархическая модель оказывается слишком громоздкой.

- В *сетевой* модели данные организуются в виде произвольного графа. Недостатком сетевой модели является жесткость структуры и высокая сложность ее реализации.
- Кроме того, значительным недостатком *иерархической* и *сетевой* моделей является то, что структура данных задается на этапе проектирования БД и не может быть изменена при организации доступа к данным.

- В *объектно-ориентированной* модели отдельные записи базы данных представляются в виде объектов. Между записями базы данных и функциями их обработки устанавливаются взаимосвязи с помощью механизмов, подобных соответствующим средствам объектно-ориентированных языков программирования. Объектно-ориентированные модели сочетают особенности сетевой и реляционной моделей и используются для создания крупных БД со сложными структурами данных.

▣ *Реляционная модель*, предложенная в 70-х годах XX века сотрудником фирмы IBM Эдгаром Коддом, получила название от английского термина relation (отношение). Реляционная БД представляет собой совокупность таблиц, *связанных отношениями*. Достоинствами реляционной модели данных являются простота, гибкость структуры, удобство реализации на компьютере, наличие теоретического описания. Большинство современных БД для персональных компьютеров являются реляционными.

Базы данных и приложения

В зависимости от взаимного расположения приложения и БД можно выделить:

- ▣ локальные БД;
- ▣ удалённые БД.

Для выполнения операций с локальными БД разрабатываются и используются локальные приложения, а для операций с удалёнными БД – клиент-серверные приложения.

Расположение БД в значительной степени влияет на разработку приложения, обрабатывающего содержащиеся в этой базе данные.

Виды приложений:

- ▣ приложения, использующие локальные базы данных, называют *одноуровневыми* (однозвенными) приложениями, поскольку приложение и базы данных образуют единую файловую структуру;
- ▣ приложения, использующие удаленные базы данных, разделяют на *двухуровневые* (двухзвенные) и *многоуровневые* (многозвенные). *Двухуровневые* приложения содержат клиентскую и серверную части;
- ▣ *многоуровневые* (обычно трехуровневые) приложения кроме клиентской и серверной частей имеют дополнительные части. К примеру, в трехуровневых приложениях имеются клиентская часть, сервер приложений и сервер базы данных.

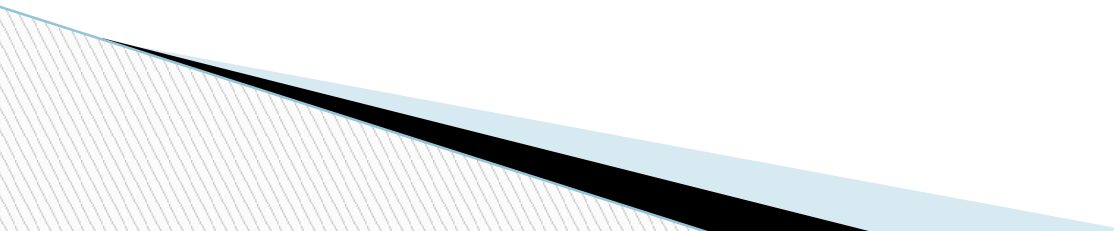
Механизмы доступа к локальным и удалённым БД

- **BDE** (Borland Database Engine — процессор баз данных фирмы Borland), предоставляющий развитый интерфейс для взаимодействия с базами данных;
- **ADO** (ActiveX Data Objects — объекты данных ActiveX) осуществляет доступ к информации с помощью OLE DB (Object Linking and Embedding Data Base - связывание и внедрение объектов баз данных);
- **dbExpress** обеспечивает быстрый доступ к информации в базах данных с помощью набора драйверов;
- **InterBase** реализует непосредственный доступ к базам данных InterBase.

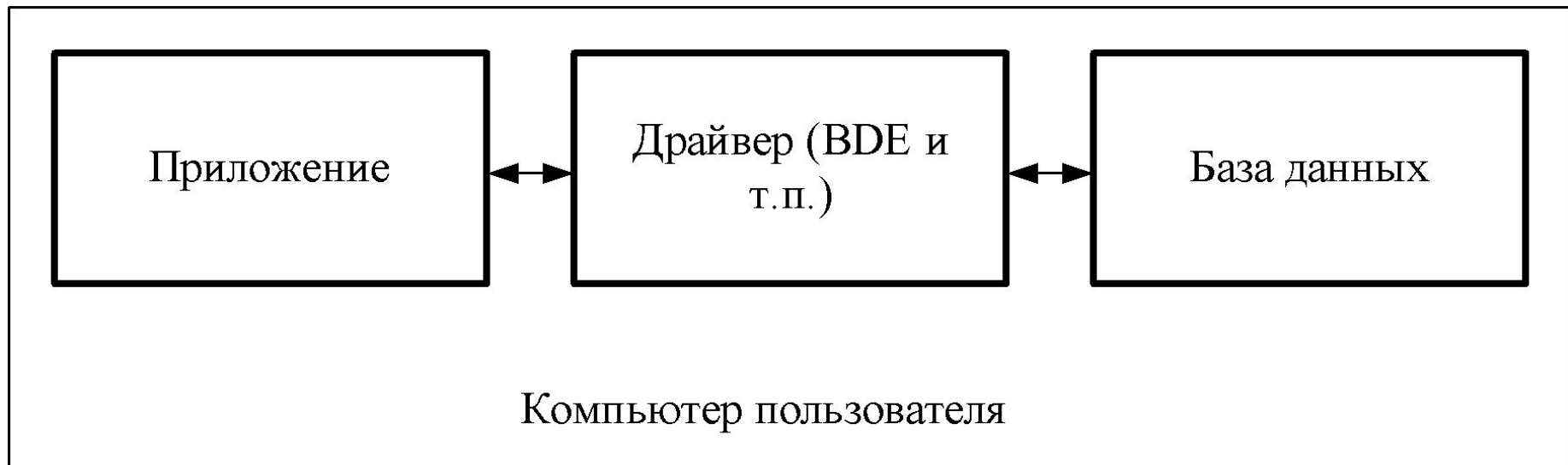
ADO

- ▣ Механизм ADO доступа к информации базы данных является стандартом фирмы Microsoft. Использование этой технологии подразумевает использование настраиваемых провайдеров данных. Технология ADO обеспечивает универсальный механизм доступа из приложений к информации источников данных. Эта технология основана на стандартных интерфейсах COM, являющихся системным механизмом Windows. Это позволяет удобно распространять приложения баз данных без вспомогательных библиотек.

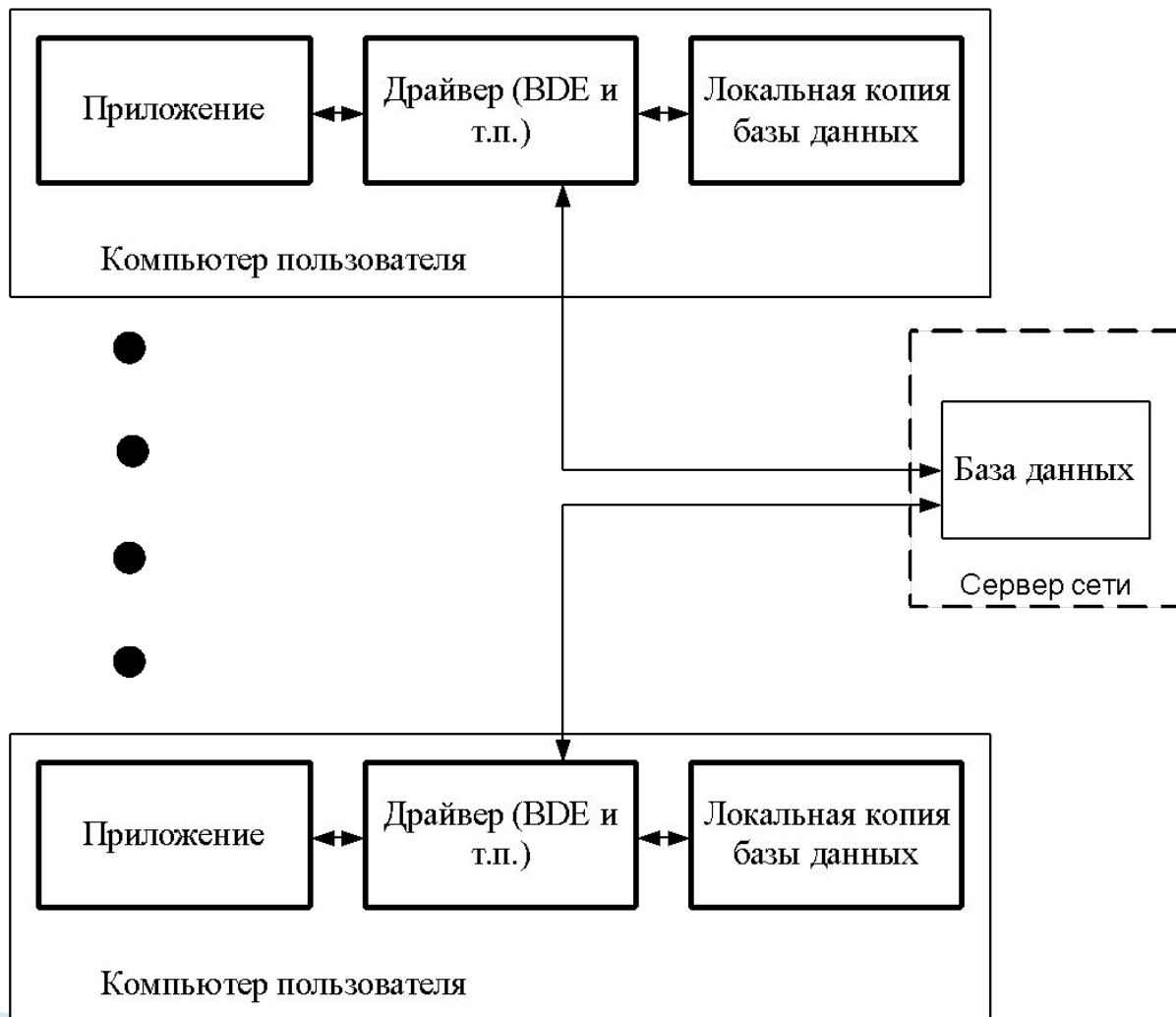
Варианты архитектуры

- Локальная архитектура
 - Архитектура «файл-сервер»
 - Архитектура «клиент-сервер»
 - Трёхуровневая архитектура «клиент-сервер»
- 

Локальная архитектура



Архитектура «файл-сервер»

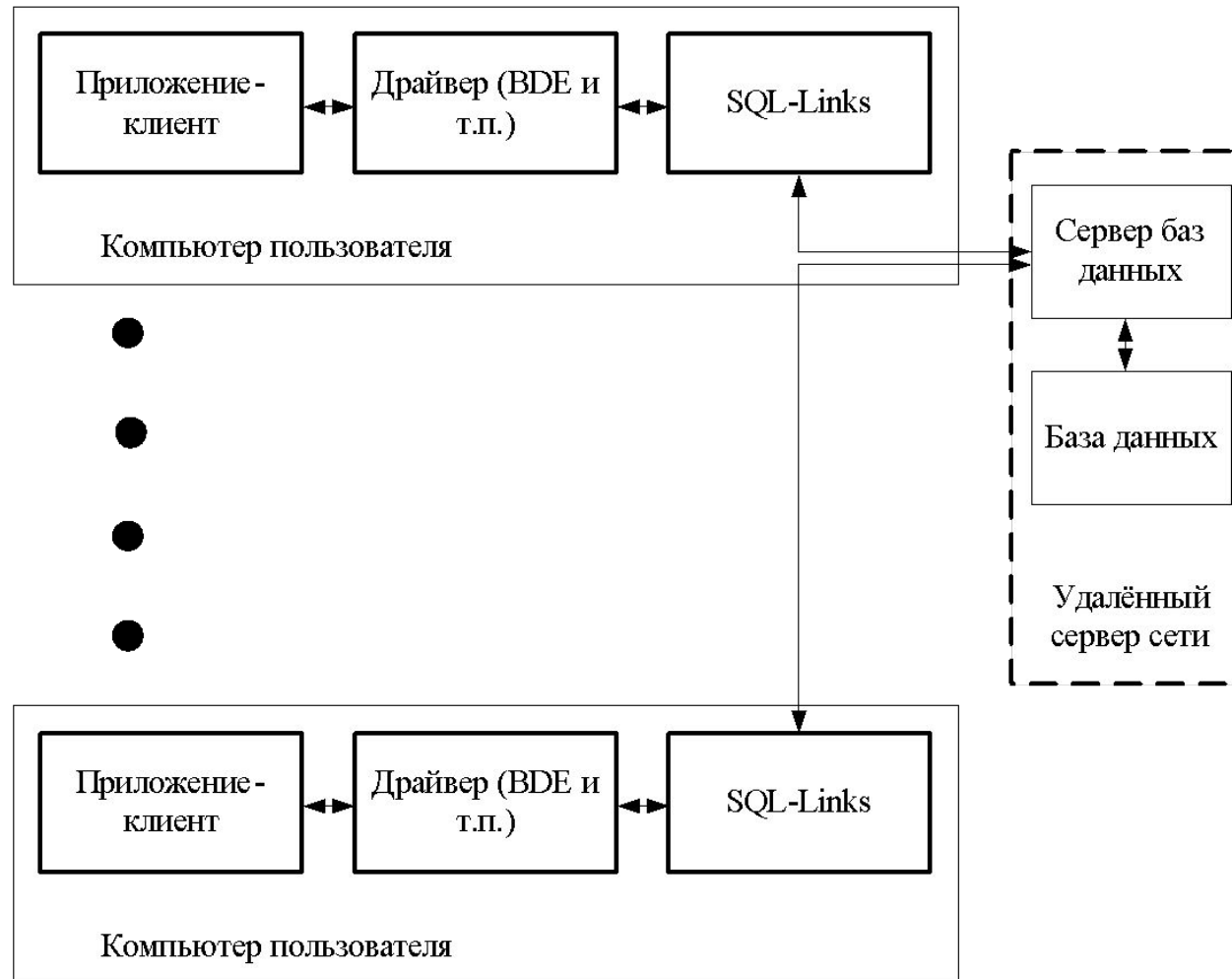


- Архитектура "файл-сервер" обычно применяется в сетях с небольшим количеством пользователей, для ее реализации подходят персональные СУБД, например, Paradox или dBase. **Достоинствами** этой архитектуры являются простота реализаций, а также то, что приложение фактически разрабатывается в расчете на одного пользователя и не зависит от компьютера сети, на который оно устанавливается.

Недостатки:

- Пользователь работает со своей локальной копией БД, данные в которой обновляются при каждом запросе к какой-либо из таблиц. При этом с сервера пересылается новая копия всей таблицы, данные которой затребованы. Таким образом, если пользователю необходимо несколько записей таблицы, с сервера по сети пересылается вся таблица. В результате циркуляции в сети больших объемов избыточной информации резко возрастает нагрузка на сеть, что приводит к соответствующему снижению её быстродействия и производительности информационной системы в целом;
- В связи с тем, что на каждом компьютере имеется своя копия БД, изменения, сделанные в ней одним пользователем, в течение некоторого времени являются неизвестными другим пользователям. Поэтому требуется постоянное обновление БД. Кроме того, возникает необходимость синхронизации работы отдельных пользователей, связанная с блокировкой в таблицах записей, которые в данный момент редактирует другой пользователь;
- Управление БД осуществляется с разных компьютеров, поэтому в значительной степени затруднена организация управления доступом, соблюдения конфиденциальности и поддержания целостности БД.

Архитектура «клиент-сервер»

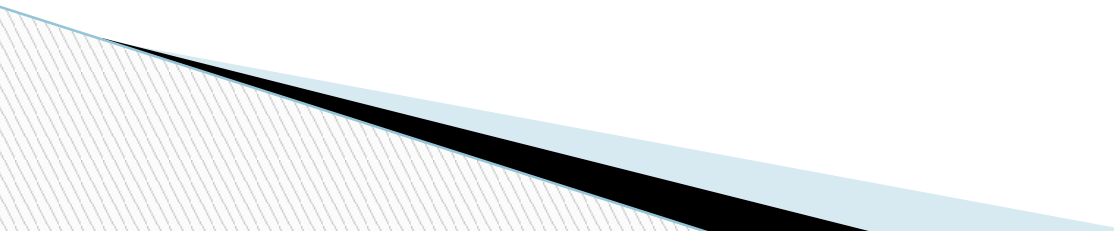


Достоинства:

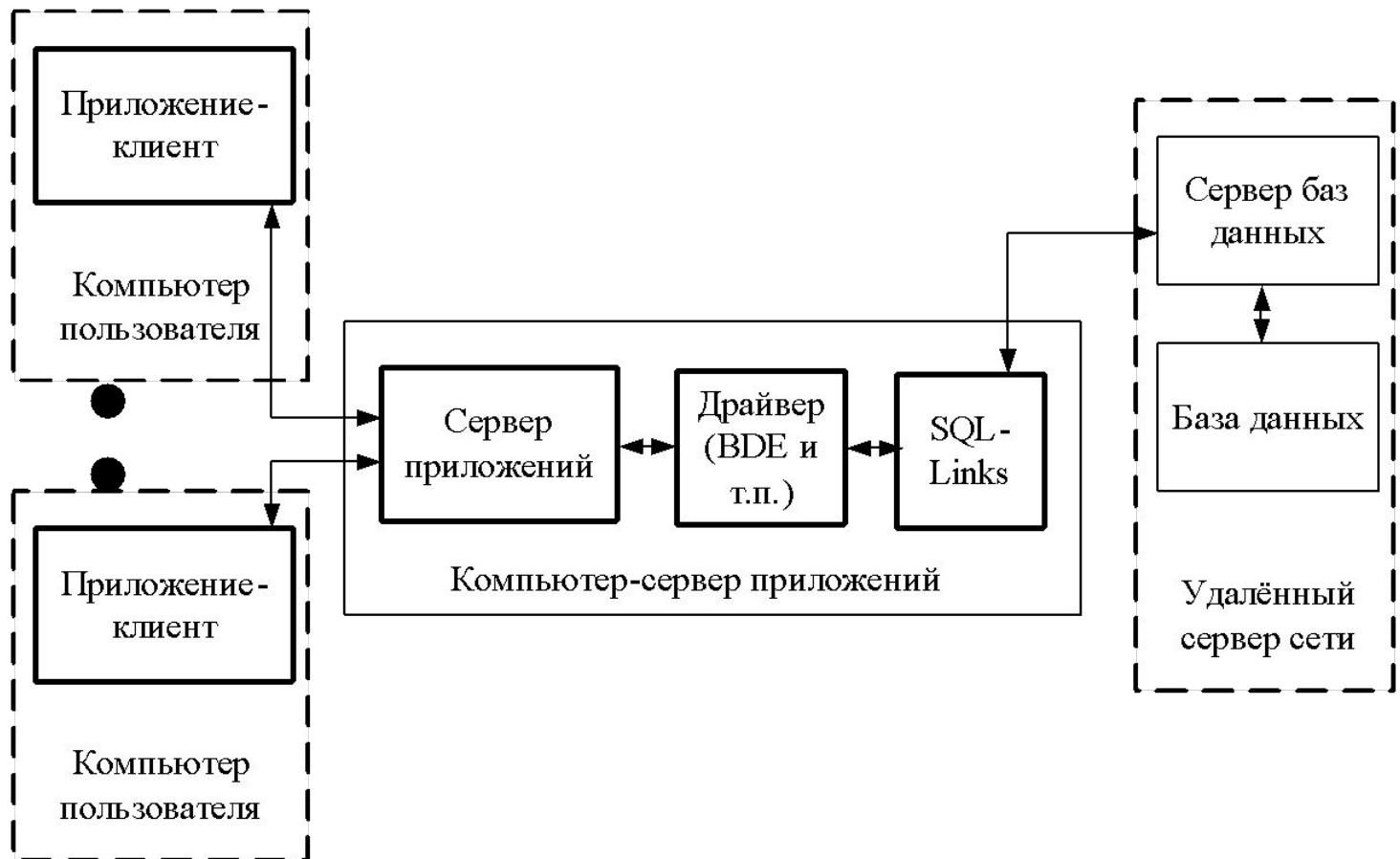
- ❑ Снижение нагрузки на сеть, поскольку в ней циркулирует нужная информация.
- ❑ Повышение безопасности информации, связанная с тем. Что обработка запросов всех клиентов выполняется единой программой, расположенной на сервере. Сервер устанавливает общие для всех пользователей правила использования БД, управляет режимами доступа клиентов к данным, запрещая. В частности, одновременное изменение одной записи различными пользователями.
- ❑ Уменьшение сложности клиентских приложений за счет отсутствия в них кода, связанного с контролем БД и разграничением доступа к ней.

- Для реализации архитектуры «клиент-сервер» обычно используются многопользовательские СУБД, например, Oracle или Microsoft SQL Server. Подобные СУБД также называют *промышленными*, т. к. они позволяют создать информационную систему организации или предприятия с большим числом пользователей. Промышленные СУБД являются сложными системами и требуют мощной вычислительной техники и соответствующего обслуживания. Обслуживание выполняет специалист (или группа специалистов).
- Доступ приложения к промышленным СУБД осуществляется через драйверы SQL-Links.

Задачи системного администратора:

- Защита БД;
 - поддержание целостности БД;
 - обучение и подготовка пользователей;
 - загрузка данных из других БД;
 - тестирование данных;
 - резервное копирование и восстановление;
 - внесение изменений в информационную систему.
- 

Трёхуровневая архитектура «клиент-сервер»



Преимущества:

- ▣ разгрузка сервера от выполнения части операций, перенесенных на сервер приложений;
 - ▣ уменьшение размера клиентских приложений за счет разгрузки их от лишнего кода;
 - ▣ единое поведение всех клиентов;
 - ▣ упрощение настройки клиентов - при изменении общего кода сервера приложений автоматически изменяется поведение приложений-клиентов.
- 