

# Введение в базы данных



# Оглавление:

Основные понятия

Информационная система

СУБД

База данных

Таблица БД

Бизнес-правила

Инструментальные средства для операций с БД система Delphi

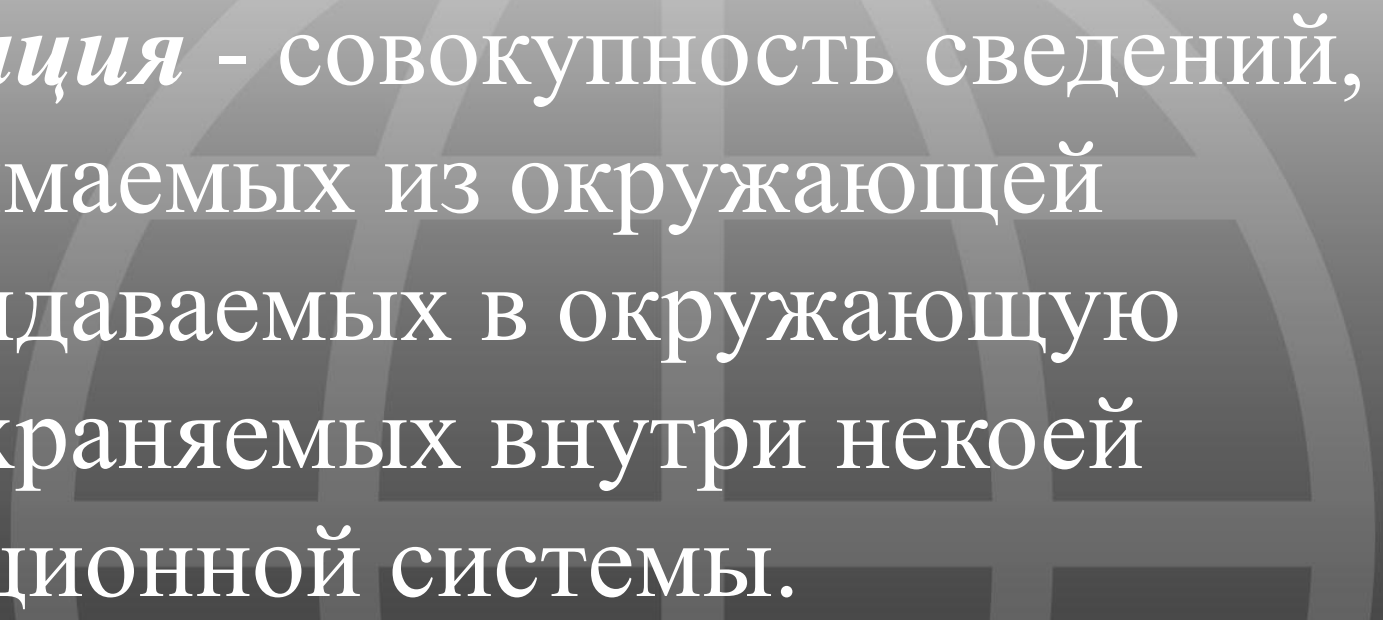
Проектирование БД

Пример создания БД



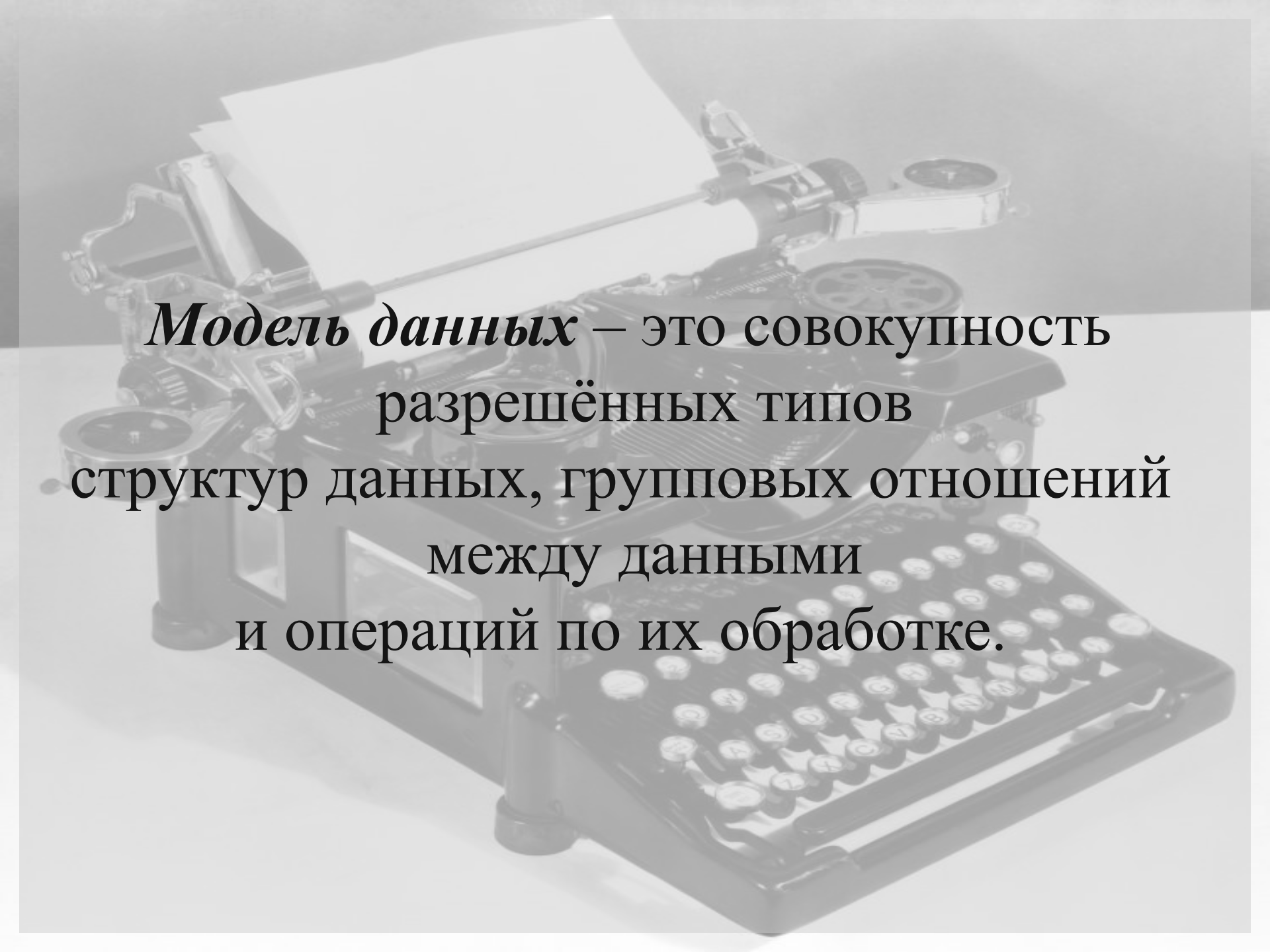
# Основные понятия



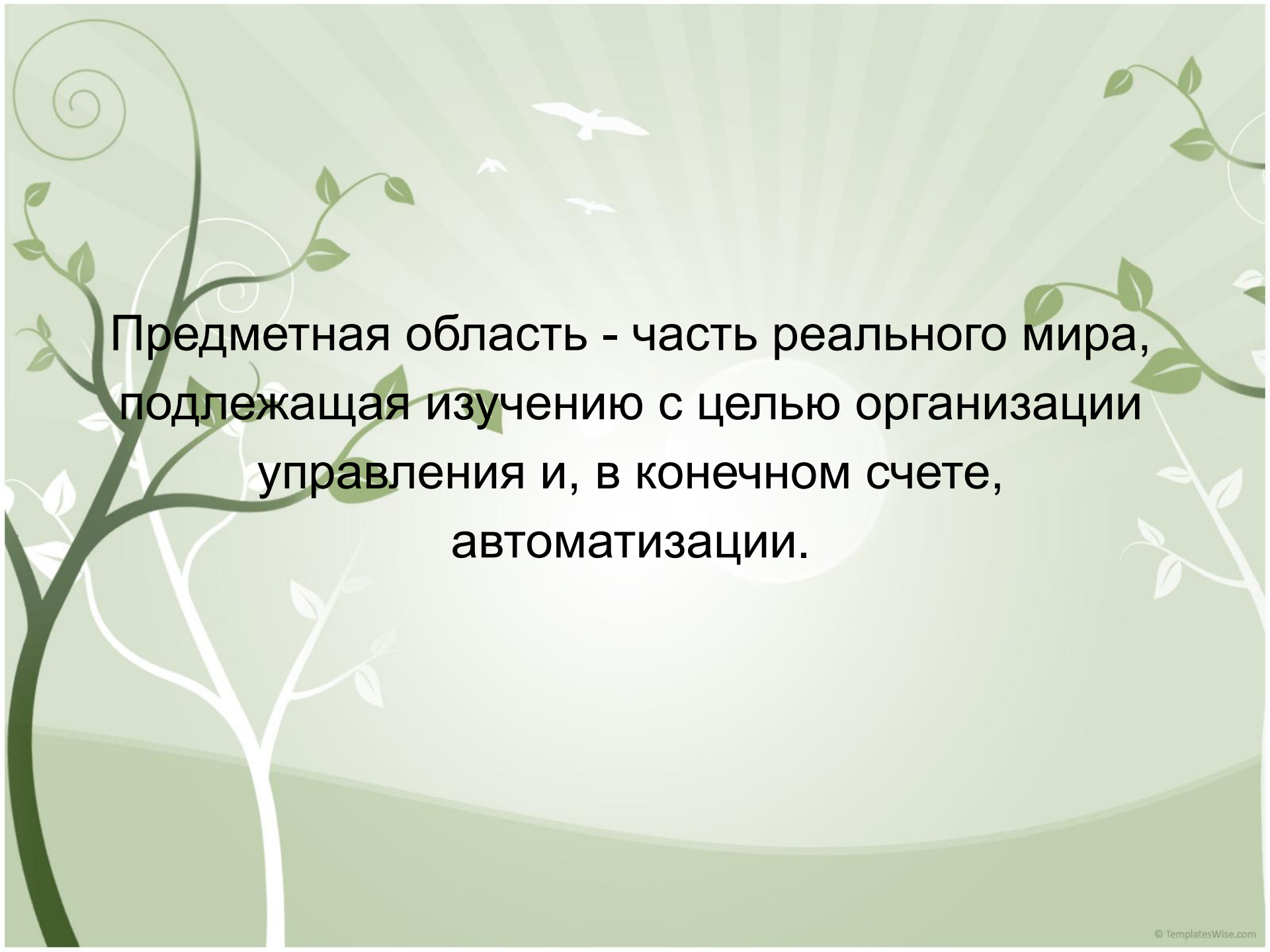


*Информация* - совокупность сведений, воспринимаемых из окружающей среды, выдаваемых в окружающую среду, сохраняемых внутри некоей информационной системы.

*Данные* - информация, представленная в виде, позволяющем упростить и автоматизировать ее сбор, хранение и дальнейшую обработку.

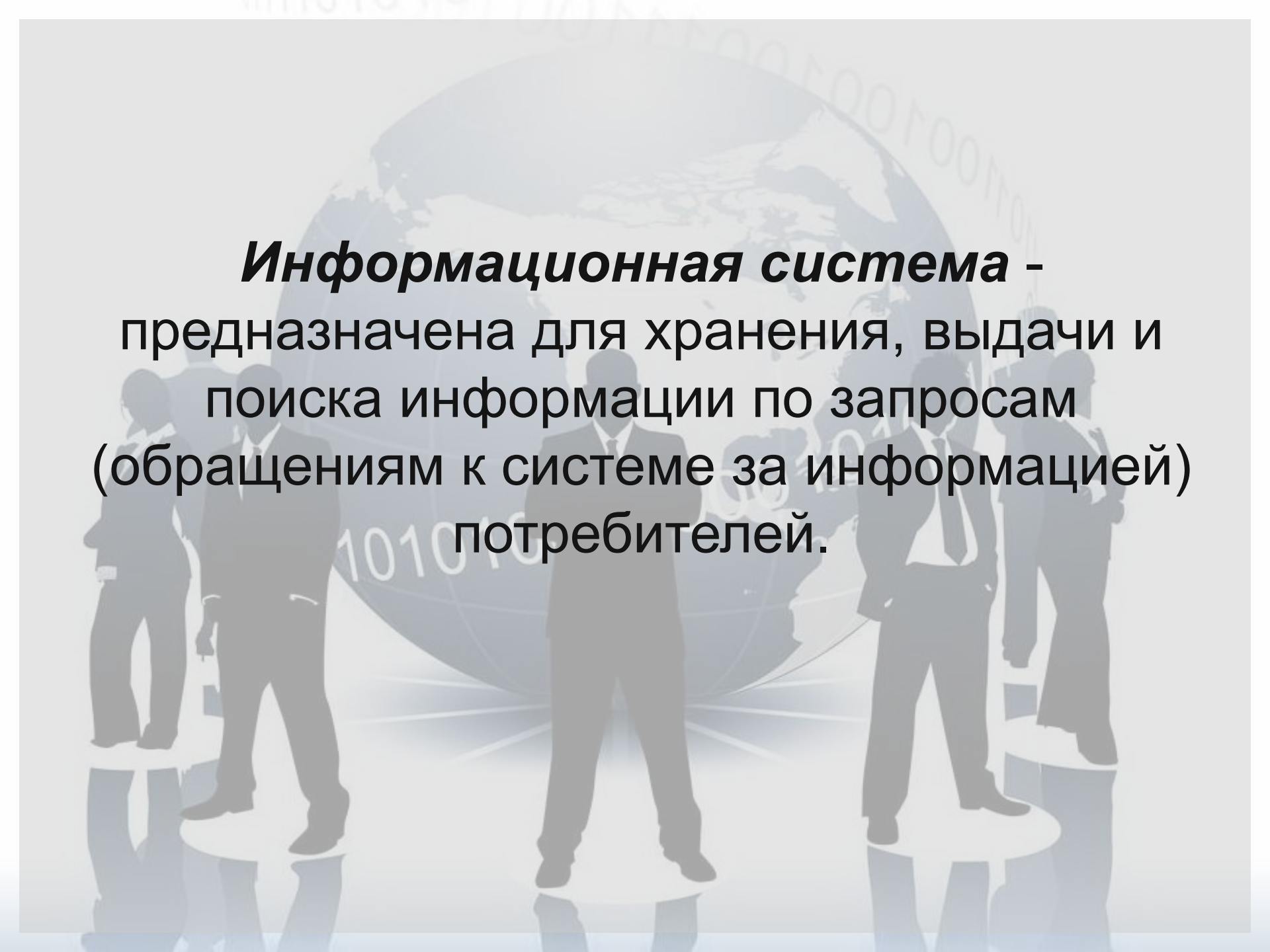


*Модель данных* – это совокупность разрешённых типов структур данных, групповых отношений между данными и операций по их обработке.



**Предметная область - часть реального мира,  
подлежащая изучению с целью организации  
управления и, в конечном счете,  
автоматизации.**




The background of the slide features a large, semi-transparent globe in the center. In the foreground, there are several silhouettes of business professionals in suits, some standing on small white circular platforms. The overall color scheme is light blue and white, with binary code (0s and 1s) scattered throughout the background.

***Информационная система*** -  
предназначена для хранения, выдачи и  
поиска информации по запросам  
(обращениям к системе за информацией)  
потребителей.



Различают 3 класса информационных систем по степени автоматизации:

1. Ручные(журналы, книги учёта);
2. Автоматизированные(работа системы + работа оператора);
3. Автоматические(работа системы).

A network diagram consisting of 15 white human icons connected by thin white lines on a light blue background. The icons are arranged in a roughly circular pattern, with lines connecting them to form a complex web of relationships. The text is overlaid on the left side of the diagram.


По степени распределённости различаются на:

- Локальные ИС (компоненты находятся на одном компьютере);
- Распределённые ИС (компоненты распределены по нескольким компьютерам).

- **Распределенная информационная система представляет собой множество баз данных, которые дистанционно удалены друг от друга и имеют ряд общих параметров.**
- **Они функционируют по общим правилам, которые определены централизованно одновременно для всех баз данных, включенных в информационную систему.**
- **Обмен информацией производится согласно правилам, которые также определены централизованно.**

Распределённые ИС, в свою очередь, разделяют на:

- файл-серверные ИС (ИС с архитектурой «файл-сервер»);
- клиент-серверные ИС (ИС с архитектурой «клиент-сервер»);

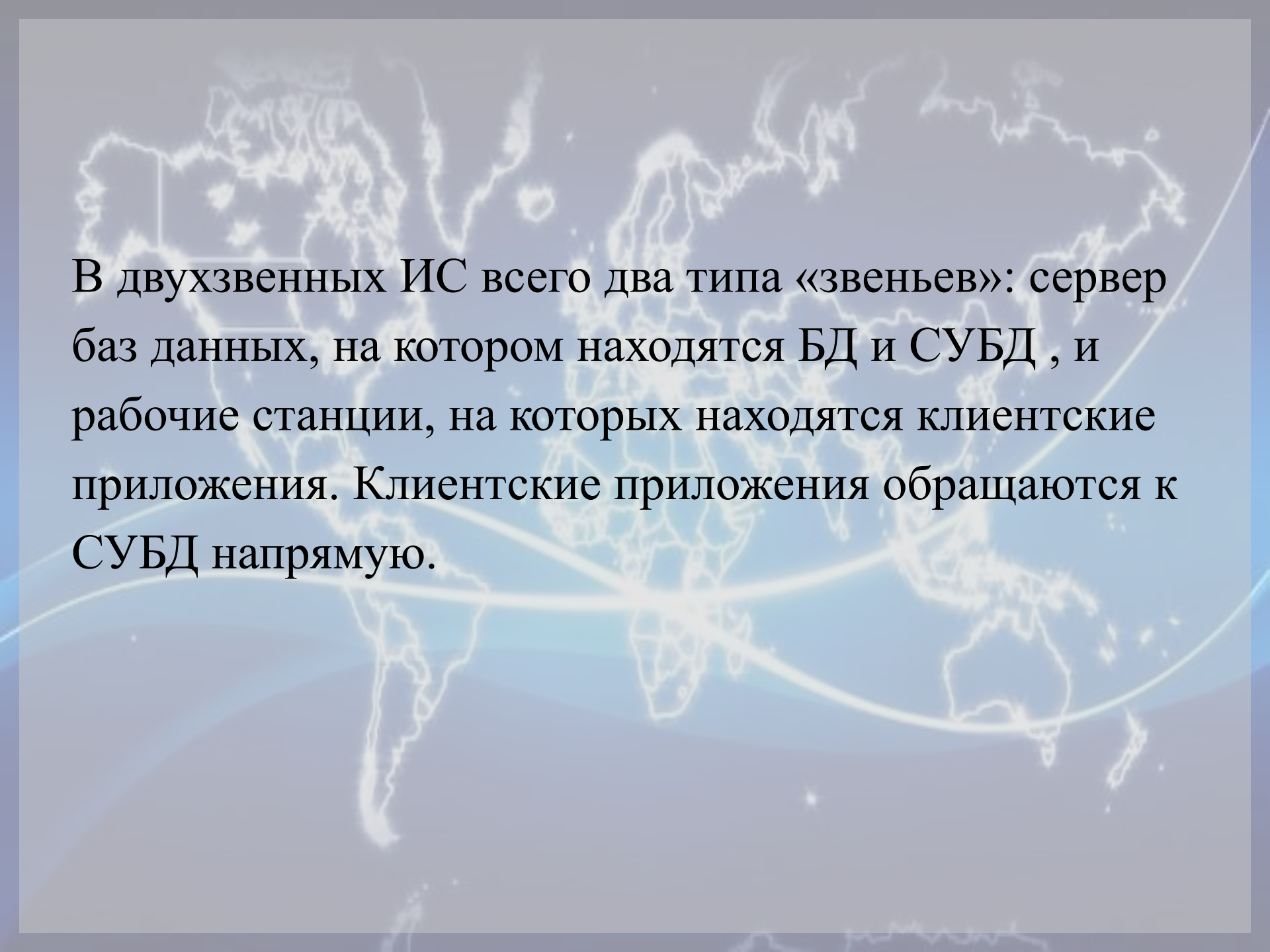
- 
- **В файл-серверных ИС** база данных находится на файловом сервере, а СУБД и клиентские приложения находятся на рабочих станциях.
  - **В клиент-серверных ИС** база данных и СУБД находятся на сервере, а на рабочих станциях находятся клиентские приложения.



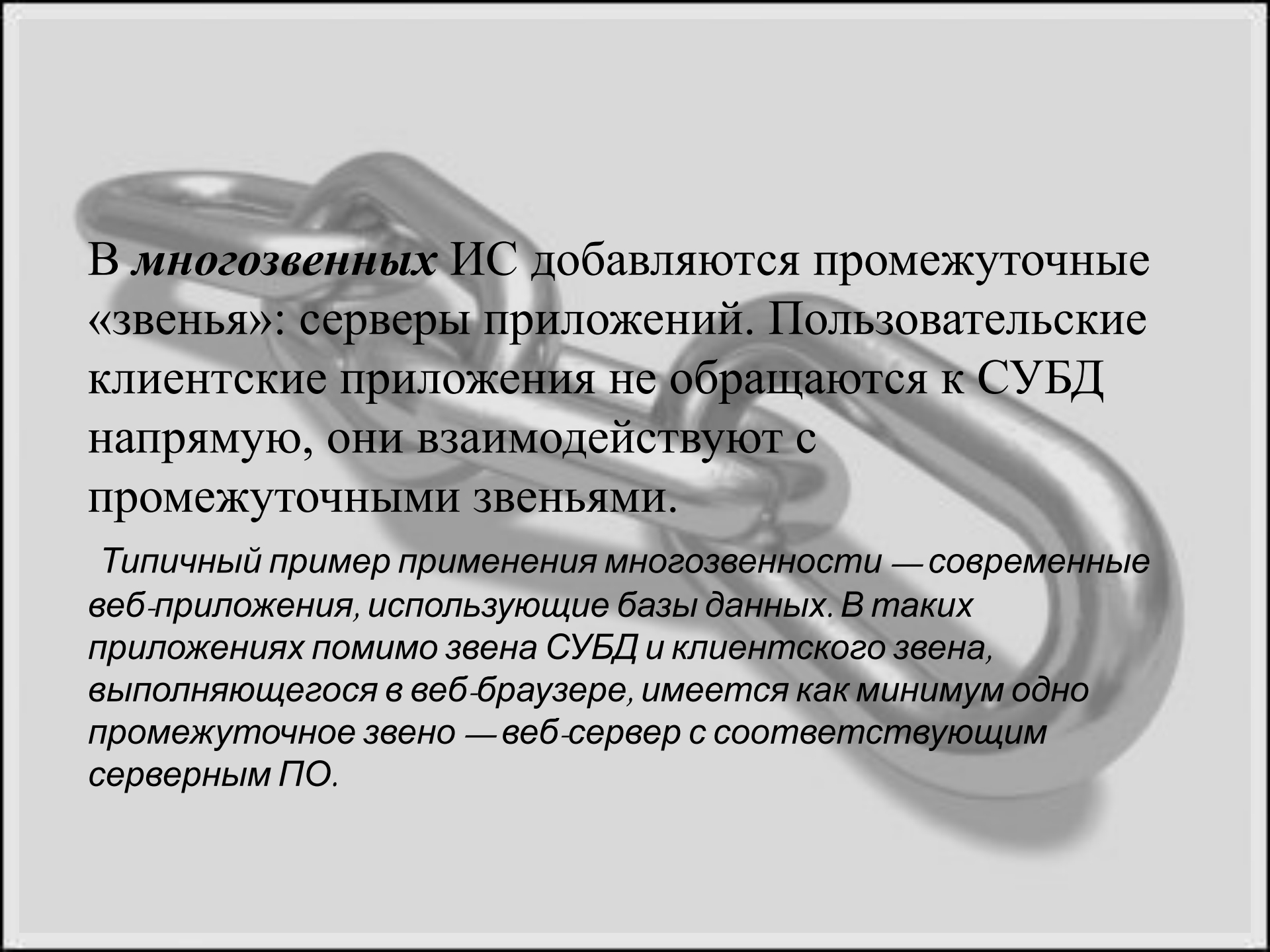
В свою очередь, клиент-серверные ИС разделяют на:

- двухзвенные
- многозвенные.





В двухзвенных ИС всего два типа «звеньев»: сервер баз данных, на котором находятся БД и СУБД, и рабочие станции, на которых находятся клиентские приложения. Клиентские приложения обращаются к СУБД напрямую.



В *многозвенных* ИС добавляются промежуточные «звенья»: серверы приложений. Пользовательские клиентские приложения не обращаются к СУБД напрямую, они взаимодействуют с промежуточными звеньями.

*Типичный пример применения многозвенности — современные веб-приложения, использующие базы данных. В таких приложениях помимо звена СУБД и клиентского звена, выполняющегося в веб-браузере, имеется как минимум одно промежуточное звено — веб-сервер с соответствующим серверным ПО.*

Современной формой информационных систем являются *банки данных*, включающие в свой состав следующие составляющие:

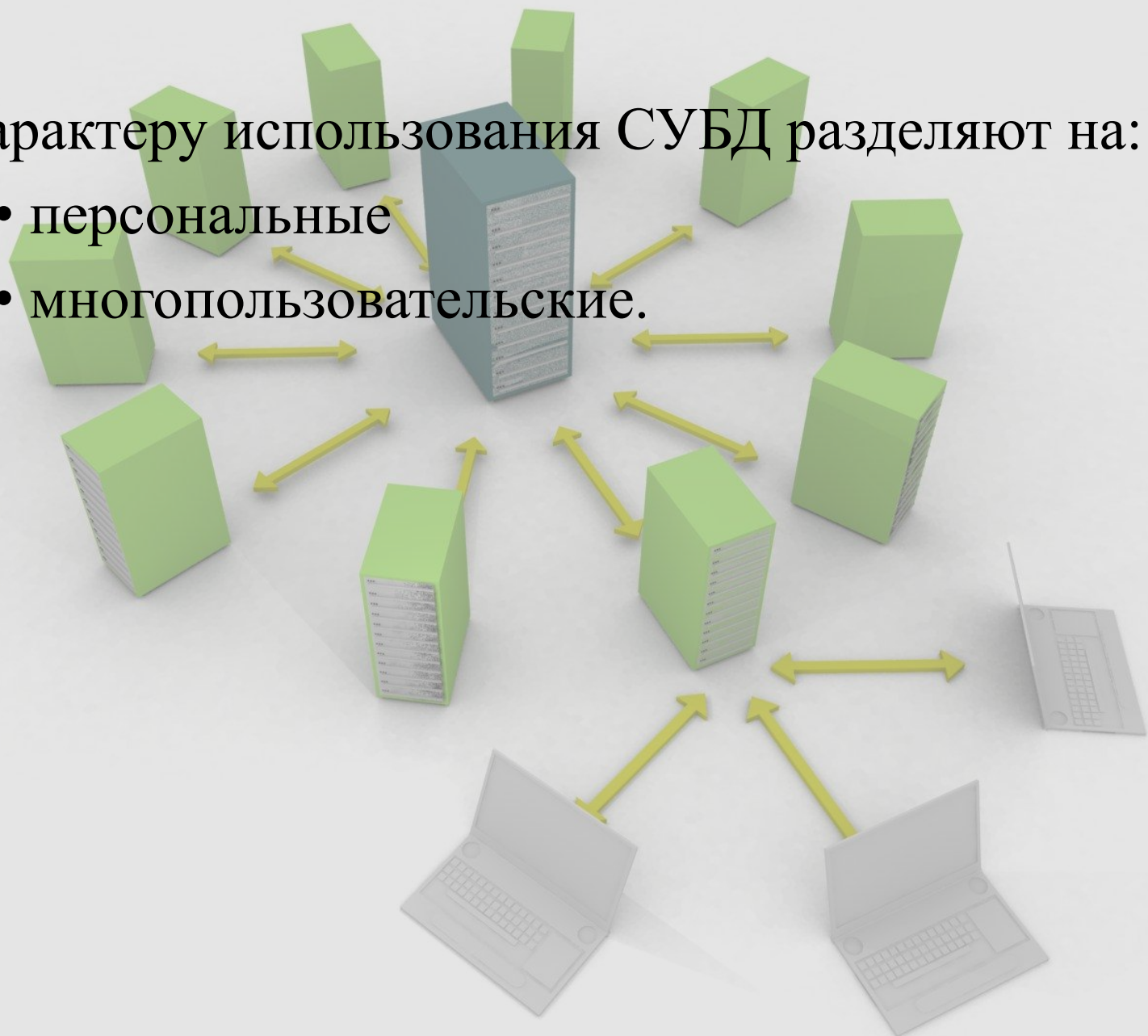
- вычислительную систему;
- систему управления базами данных (СУБД);
- одну или несколько баз данных (БД);
- набор прикладных программ (приложений БД).

# СУБД



По характеру использования СУБД разделяют на:

- персональные
- многопользовательские.





A stack of three floppy disks is placed on a computer keyboard. The keyboard is light-colored, and the keys are visible. The floppy disks are stacked vertically, with the top one slightly offset. The background is a soft, out-of-focus blue and white.

***Персональная СУБД*** обеспечивает  
возможность создания локальных БД,  
работающих на одном компьютере.

*К персональным СУБД относятся Paradox, dBase, FoxPro,  
Access и другие.*





***Многопользовательские СУБД*** позволяют создавать информационные системы, функционирующие в архитектуре "клиент-сервер".

*К многопользовательским СУБД относятся Oracle, Informix, SyBase, Microsoft SQL Server, InterBase и др.*

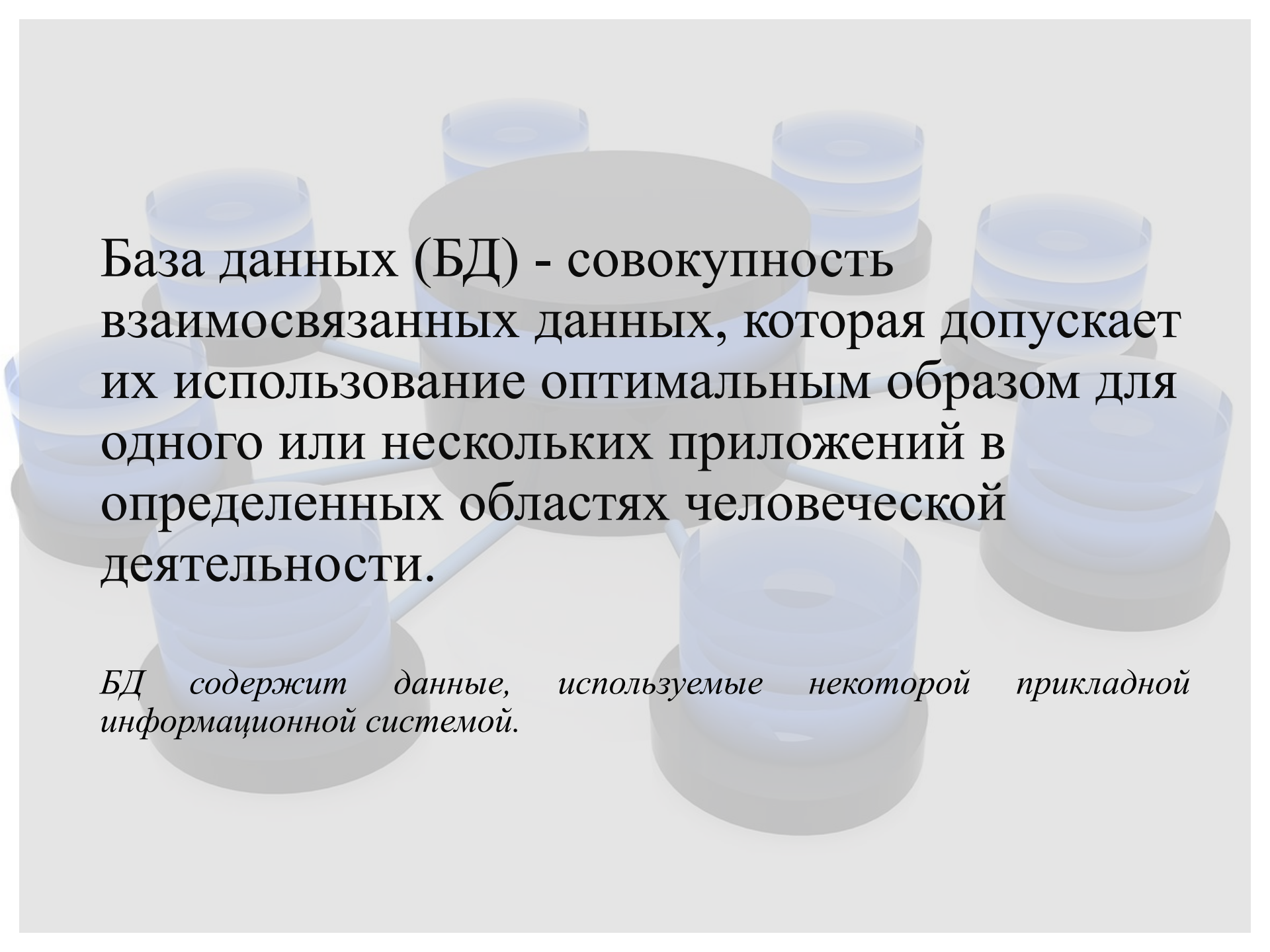


В состав языковых средств современных СУБД входят:

- 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 (Structured Query Language, SQL), обеспечивающий управление структурой БД и манипулирование данными, а также являющийся стандартным средством доступа к удаленным БД;
- язык запросов по образцу (Query By Example, QBE), обеспечивающий визуальное конструирование запросов к БД.

## Современная СУБД содержит следующие компоненты:

- ядро, которое отвечает за управление данными во внешней и оперативной памяти, и журнализацию (сохранение информации, необходимой для восстановления базы данных в предыдущее консистентное состояние в случае логических или физических отказов);
- процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;
- подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД
- а также сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.



База данных (БД) - совокупность взаимосвязанных данных, которая допускает их использование оптимальным образом для одного или нескольких приложений в определенных областях человеческой деятельности.

*БД содержит данные, используемые некоторой прикладной информационной системой.*



В зависимости от взаимного расположения приложения и БД можно выделить:

- локальные БД;
- удаленные БД.



*Локальные БД* располагаются на том же компьютере, что и работающие с ними приложения. В этом случае информационная система имеет локальную архитектуру.

Для доступа к локальной БД процессор баз данных BDE использует стандартные драйверы, которые позволяют работать с форматами БД dBase, Paradox, FoxPro, а также с текстовыми файлами.





***Удаленная БД*** размещается на компьютере-сервере сети, а приложение, осуществляющее работу с этой БД, находится на компьютере пользователя.

*В этом случае речь идет об архитектуре клиент-сервер, когда информационная система делится на неоднородные части — сервер и клиент БД.*

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

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

# Иерархическая БД

*Иерархической* называется БД, объекты которой распределены по иерархии и имеют разные уровни.



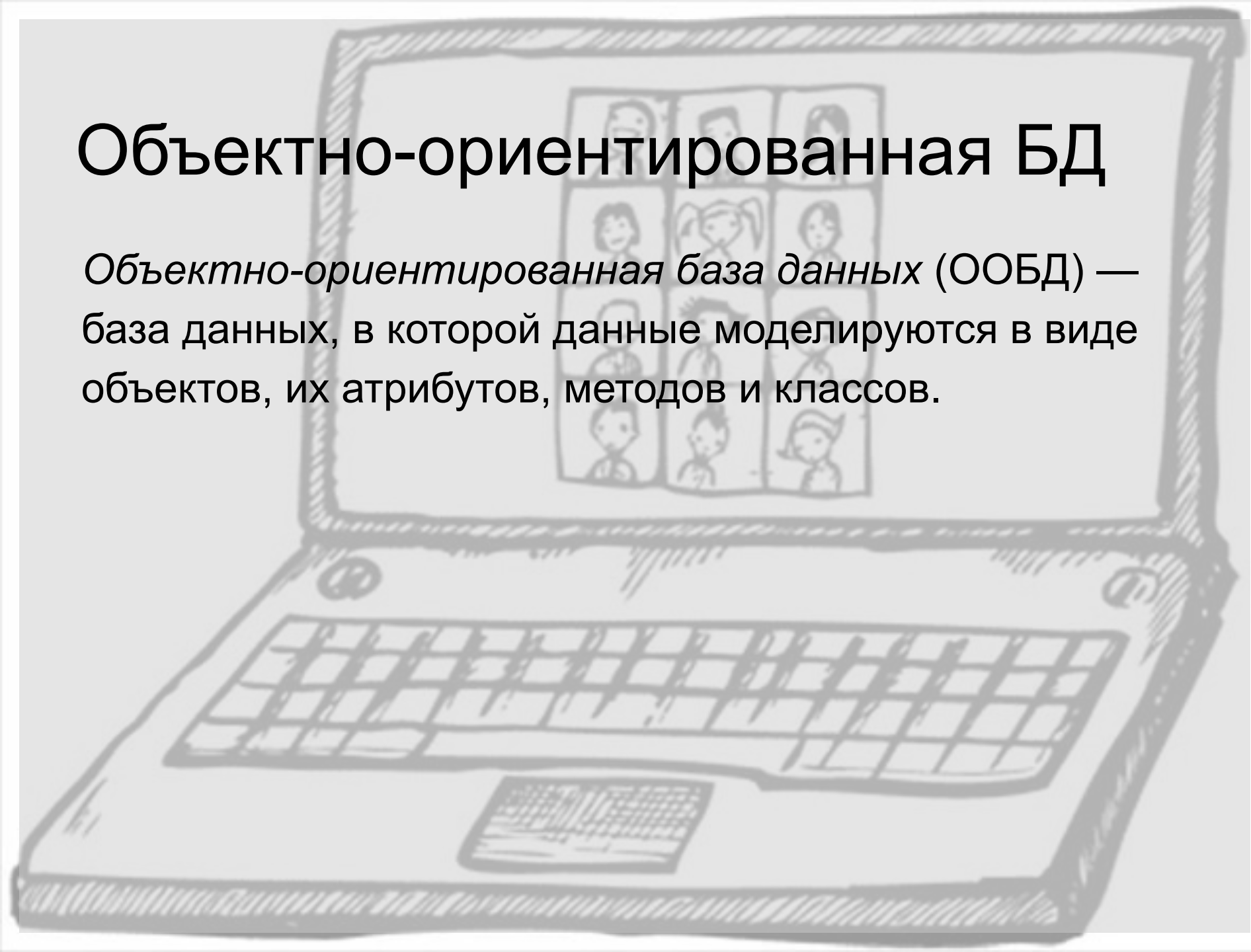
# Сетевая БД

The background features a network diagram with human icons. At the top, there are several blue icons. Below them are green icons, and further down are yellow icons. At the bottom center is a red icon. Large, 3D-style arrows connect these groups, pointing from the blue icons to the green, from the green to the yellow, and from the yellow to the red icon. The arrows have a gradient from light to dark.

*Сетевой* называется БД, к данным которой могут получить доступ одновременно несколько пользователей по средствам сети.

# Объектно-ориентированная БД

*Объектно-ориентированная база данных (ООБД)* — база данных, в которой данные моделируются в виде объектов, их атрибутов, методов и классов.





# Реляционная БД

*Реляционной* считается такая база данных, в которой все данные представлены для пользователя в виде прямоугольных таблиц значений данных, и все операции над базой данных сводятся к манипуляциям с таблицами.





# Объектно-реляционная БД

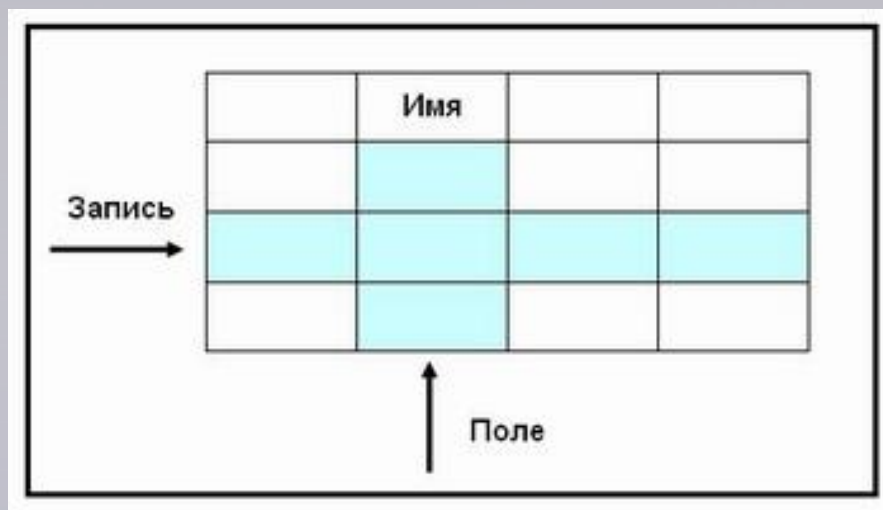
Это реляционная БД, поддерживающая некоторые технологии, реализующие объектно-ориентированный подход.



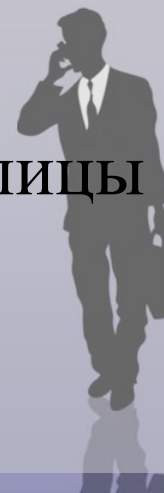
*Реляционная БД* состоит из взаимосвязанных таблиц. Каждая таблица содержит информацию об объектах одного типа, а совокупность всех таблиц образует единую БД.



Каждая таблица БД состоит из *строк и столбцов* и предназначена для хранения данных об однотипных объектах информационной системы.



Строка таблицы называется записью, столбец таблицы - полем. Каждое поле должно иметь уникальное в пределах таблицы имя.

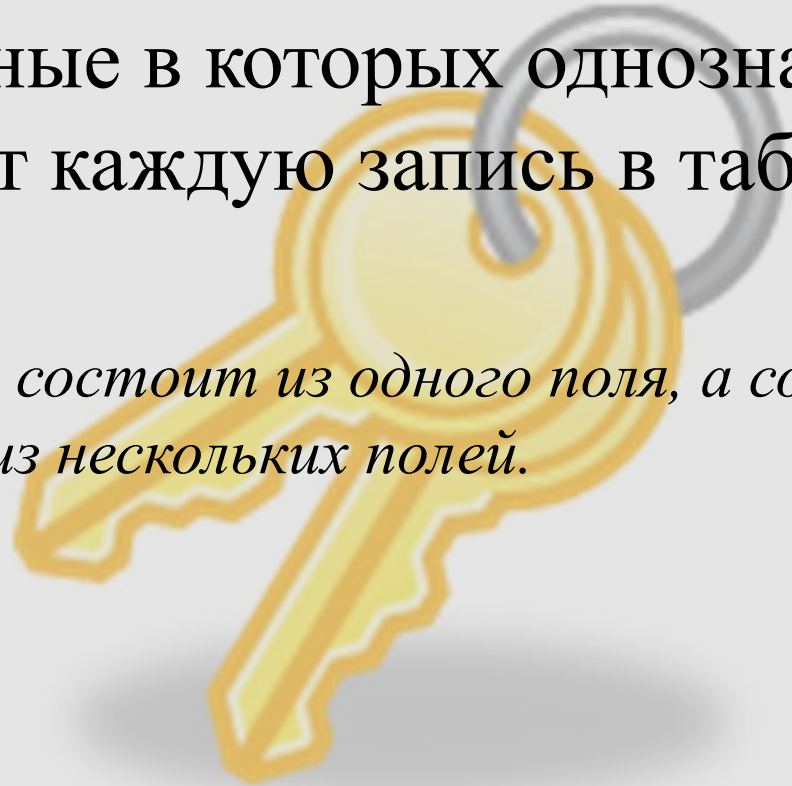


Основу таблицы составляет описание ее полей, каждая таблица должна иметь хотя бы одно поле. Понятие ***структуры таблицы*** является более широким и включает в себя:

- описание полей;
- ключ;
- индексы;
- ограничения на значения;
- ограничения ссылочной целостности между таблицами;
- пароли.

**Ключ** представляет собой комбинацию полей, данные в которых однозначно определяют каждую запись в таблице.

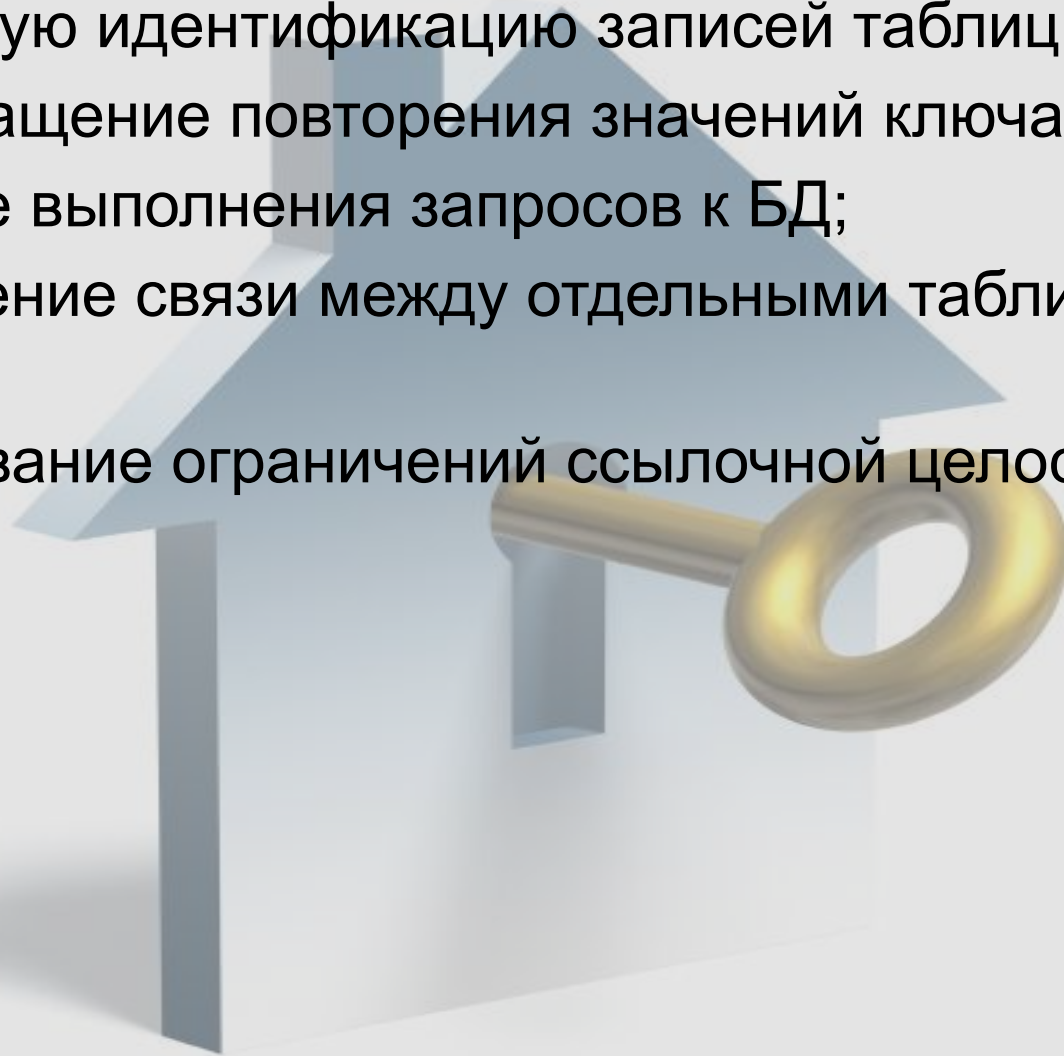
*Простой ключ состоит из одного поля, а составной (сложный) — из нескольких полей.*





# Ключ обеспечивает:

- однозначную идентификацию записей таблицы;
- предотвращение повторения значений ключа;
- ускорение выполнения запросов к БД;
- установление связи между отдельными таблицами БД;
- использование ограничений ссылочной целостности.



Таблицы различных форматов имеют свои особенности построения ключей. Вместе с тем существуют и общие правила, состоящие в следующем:

- Ключ должен быть уникальным. У составного ключа значения отдельных полей (но не всех одновременно) могут повторяться.
- Ключ должен быть достаточным и не избыточным, т. е. не содержать поля, которые можно удалить без нарушения уникальности ключа.
- В состав ключа не могут входить поля некоторых типов, например, графическое поле или поле комментария.

*Индекс*, как и ключ, строится по полям таблицы, однако он может допускать повторение значений составляющих его полей, в этом состоит его основное отличие от ключа.

Поля, по которым построен индекс, называют индексными. Простой индекс состоит из одного поля, а составной — из нескольких полей.

Создание индекса называют индексированием таблицы.

# Использование индекса обеспечивает:

- увеличение скорости доступа (поиска) к данным;
- сортировку записей;
- установление связи между отдельными таблицами БД;
- использование ограничений ссылочной целостности.

Использование индекса повышает скорость доступа к данным в таблице за счет того, что доступ выполняется не последовательным, а индексно-последовательным методом.

**Сортировка** представляет собой упорядочивание записей по полю или группе полей в порядке возрастания или убывания их значений. Индекс служит для сортировки таблиц по индексным полям.



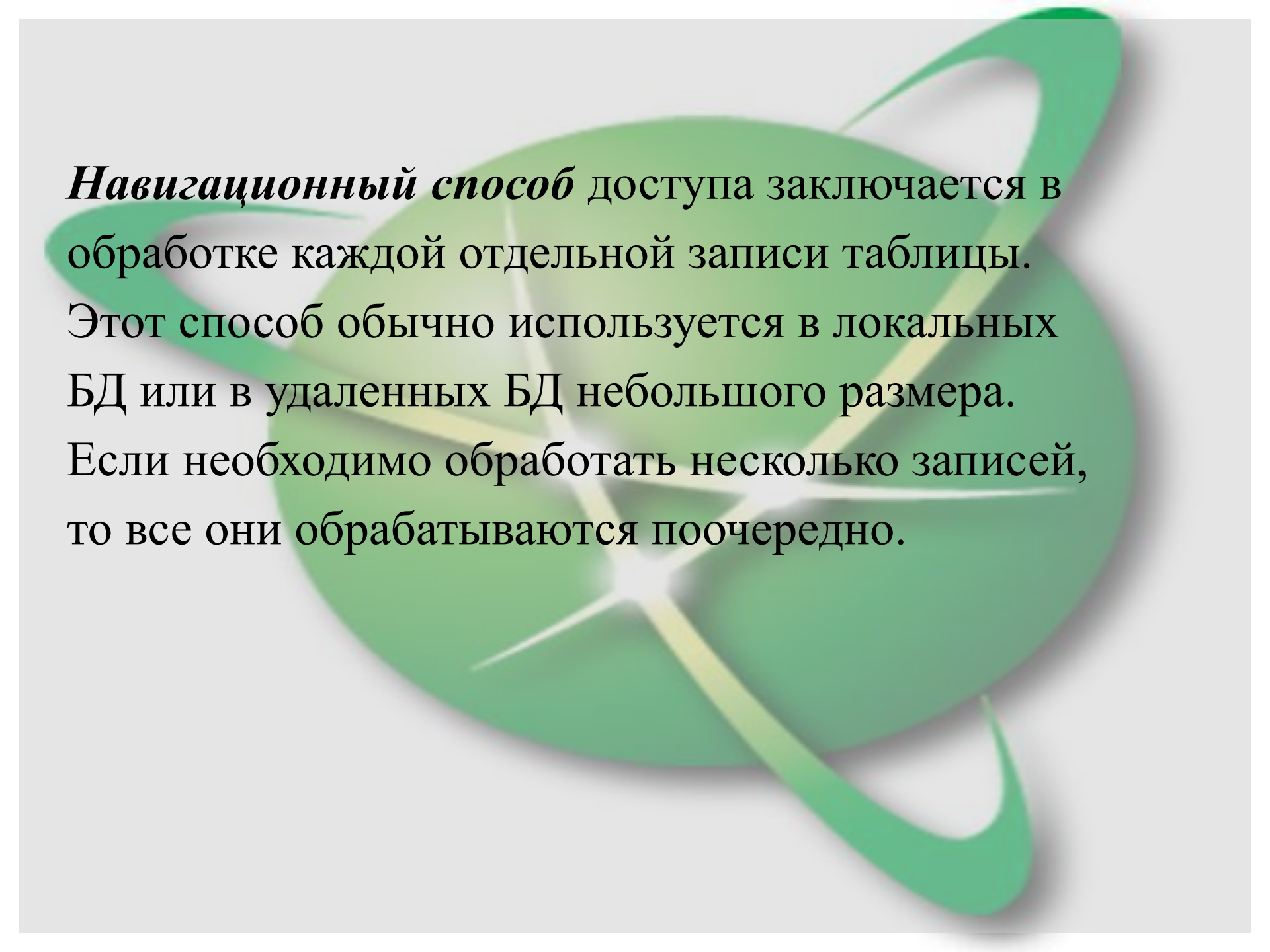
**СУБД** представляет собой совокупность языковых и программных средств, предназначенных для создания, ведения и использования БД.



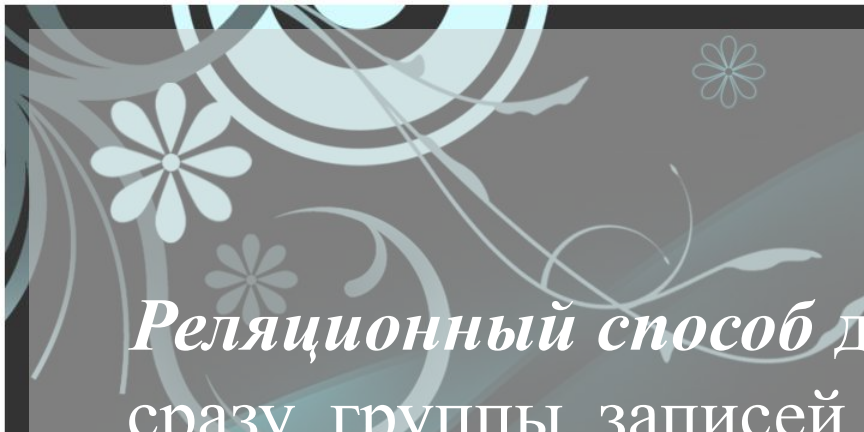
# Способы доступа к данным

При выполнении операций с таблицами используется один из следующих способов доступа к данным:

- навигационный;
- реляционный.



***Навигационный способ*** доступа заключается в обработке каждой отдельной записи таблицы. Этот способ обычно используется в локальных БД или в удаленных БД небольшого размера. Если необходимо обработать несколько записей, то все они обрабатываются поочередно.

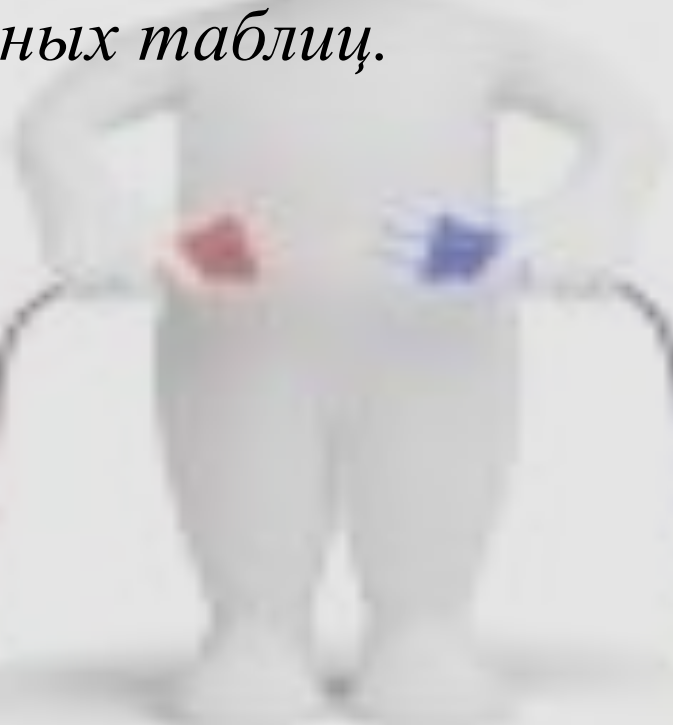


*Реляционный способ* доступа основан на обработке сразу группы записей, при этом если необходимо обработать одну запись, то обрабатывается группа, состоящая из одной записи.

Реляционный способ доступа основывается на SQL-запросах, поэтому его также называют SQL-ориентированным. Этот способ доступа ориентирован на выполнение операций с удаленными БД, хотя его можно использовать также и для локальных БД.

# Связь между таблицами

*В частном случае БД может состоять из одной таблицы, однако обычно реляционная БД состоит из взаимосвязанных таблиц.*





Организация связи (отношений) между таблицами называется *связыванием или соединением* таблиц.



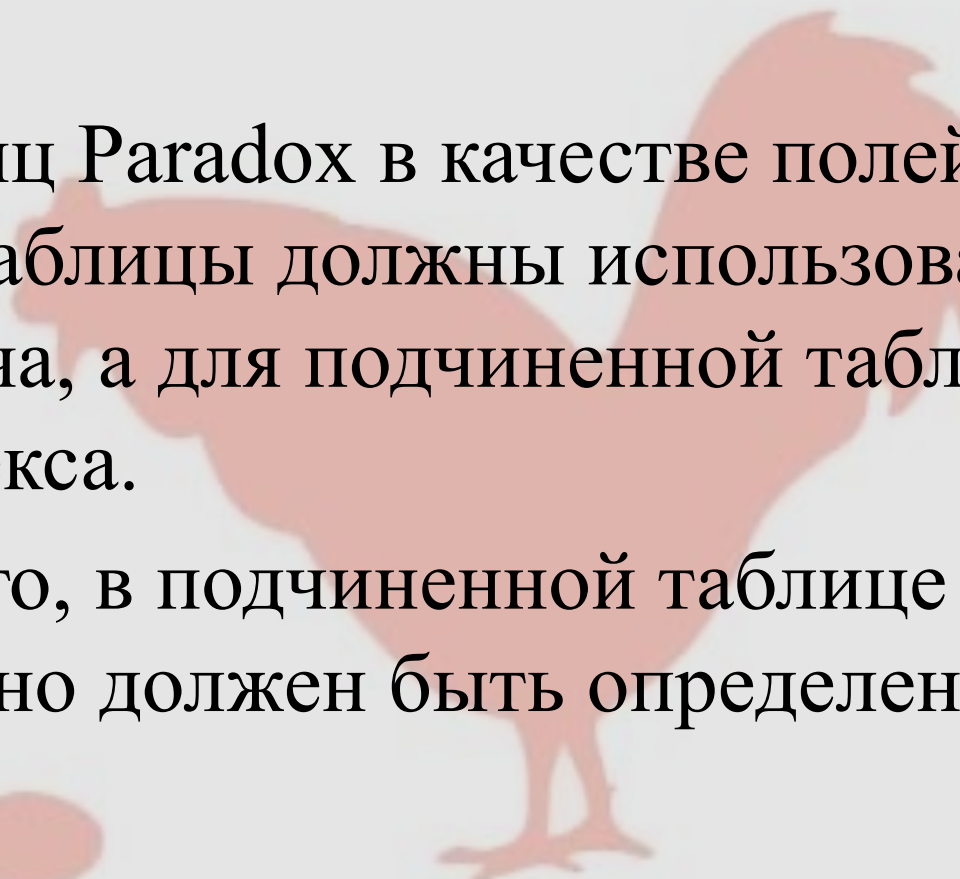
# Правила связывания таблиц

- Для связывания таблиц используются поля связи. Поля связи обязательно должны быть индексированными.
- В подчиненной таблице для связи с главной таблицей берется индекс, который также называется внешним ключом.
- Состав полей этого индекса должен полностью или частично совпадать с составом полей индекса главной таблицы.

## Частные случаи

Использование индексов зависит от формата связываемых таблиц. Так, для таблиц dBase индексы строятся по одному полю и нет деления на ключ и индексы.

Для организации связи в главной и подчиненной таблицах выбираются индексы, составленные по полям совпадающего типа.



Для таблиц Paradox в качестве полей связи главной таблицы должны использоваться поля ключа, а для подчиненной таблицы — поля индекса.

Кроме того, в подчиненной таблице обязательно должен быть определен ключ.

PARADOX

# Схема связи таблиц в Paradox

M_code	...
Ключевое поле	...

D_number	D_code	...
Ключевое поле	Индексное поле	...

# Пояснение с схеме

В главной таблице определен ключ, построенный по полю M\_Code автоинкрементного типа. В подчиненной таблице определен ключ по полю D\_number также автоинкремент-ного типа и индекс, построенный по полю D\_code целочисленного типа. Связь между таблицами устанавливается по полям D\_code и M\_code. Индекс по полю DCode является внешним ключом. Подобное именованние полей облегчает ориентацию в их названиях, особенно при большом количестве таблиц.



Связь между таблицами определяет отношение подчиненности, при котором одна таблица является, а вторая . Саму связь (отношение) называют связь "главный-подчиненный", "родительский дочерний" или "мастер - детальный". Существуют следующие виды СВЯЗИ:

- отношение "один - к - одному«;
- отношение "один - ко - многим«;
- отношение "много - к - одному«;
- отношение "много - ко - многим".



# Механизм транзакций

Информация БД в любой момент времени должна быть целостной и непротиворечивой. Одним из путей обеспечения этого является использование механизма транзакций. Транзакция представляет собой выполнение последовательности операций.




## Использование механизма транзакций необходимо:

- при выполнении последовательности взаимосвязанных операций с БД;
- при многопользовательском доступе к БД.



## При использовании этого механизма возможны две ситуации.

- Успешно завершены все операции. В этом случае транзакция считается успешной, и все изменения в БД, которые были произведены в рамках транзакции отдельными операциями, утверждаются. В результате БД переходит из одного целостного состояния в другое.
- Неудачно завершена хотя бы одна операция. При этом вся транзакция считается неуспешной, и результаты выполнения всех операций (даже успешно выполненных) отменяются. В результате происходит возврат БД в состояние, в котором она находилась до начала транзакции.



Транзакция может быть *неявной* или *явной*.  
*Неявная* транзакция стартует автоматически, а по завершении также автоматически подтверждается или отменяется.

*Явной* транзакцией управляет программист с использованием компонента Database и/или средств SQL.

# Бизнес-правила

*Бизнес - правила* представляют собой механизмы управления БД и предназначены для поддержания БД в целостном состоянии, а также для выполнения ряда других действий.

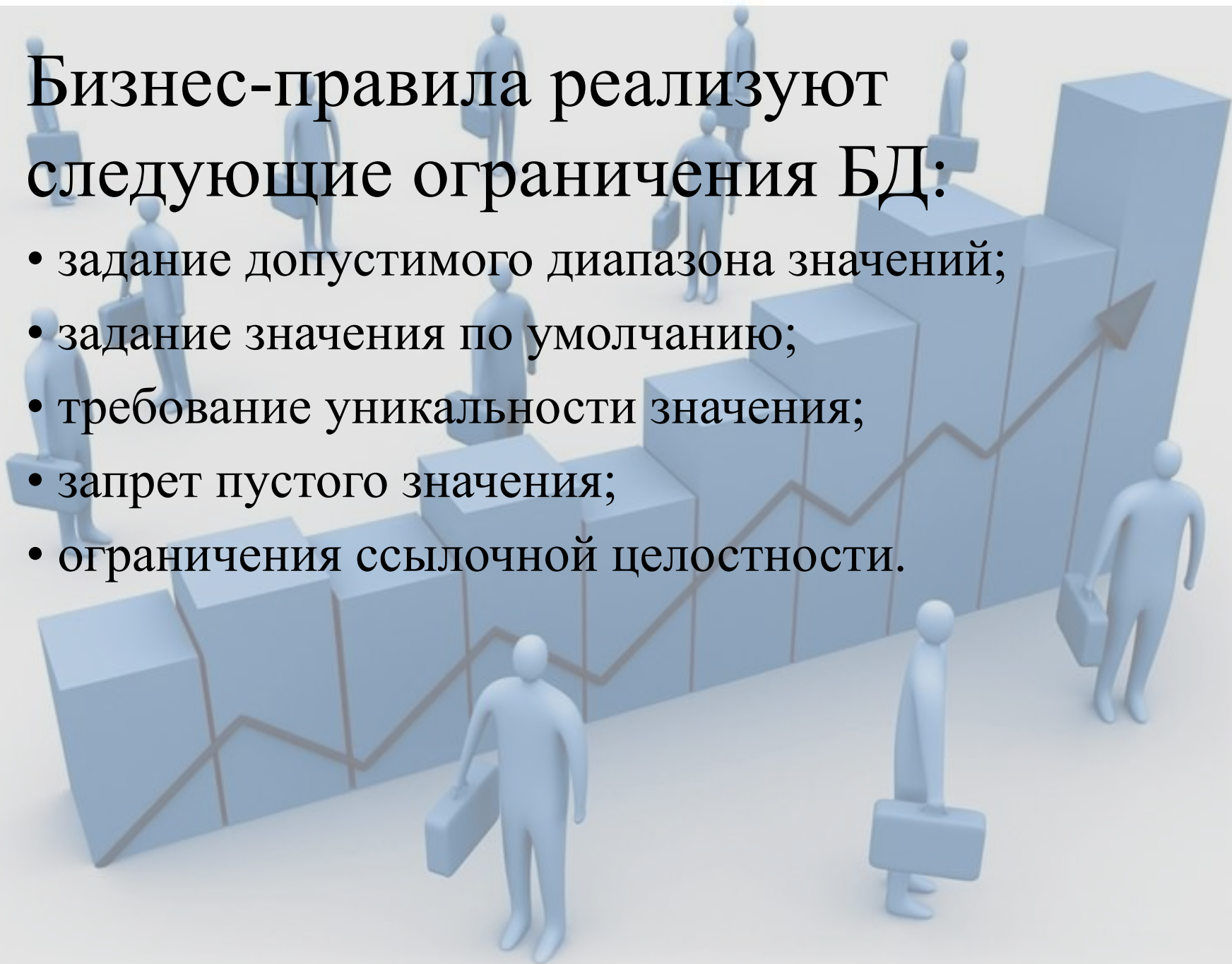


Rules



# Бизнес-правила реализуют следующие ограничения БД:

- задание допустимого диапазона значений;
- задание значения по умолчанию;
- требование уникальности значения;
- запрет пустого значения;
- ограничения ссылочной целостности.



*Инструментальные средства для операций с БД система Delphi* предлагает набор инструментальных средств, перечисленных ниже:

- **Borland Database Engine (BDE)** — процессор баз данных, который представляет собой набор динамических библиотек и драйверов, предназначенных для организации доступа к БД из Delphi-приложений. BDE является центральным звеном при организации доступа к данным.
- **BDE Administrator** — утилита для настройки различных параметров BDE.
- **Database Desktop** — программа создания и редактирования таблиц, SQL-запросов и запросов.
- **SQL Explorer** — Проводник БД, позволяющий просматривать и редактировать БД и словари данных.
- **SQL Builder** — программа визуального конструирования SQL-запросов.
- **Data Pump** — программа для переноса данных между БД.

# Проектирование БД

- Концептуальное
- Логическое
- Физическое





***Концептуальное проектирование*** - сбор, анализ и редактирование требований к данным. Для этого осуществляются следующие мероприятия:

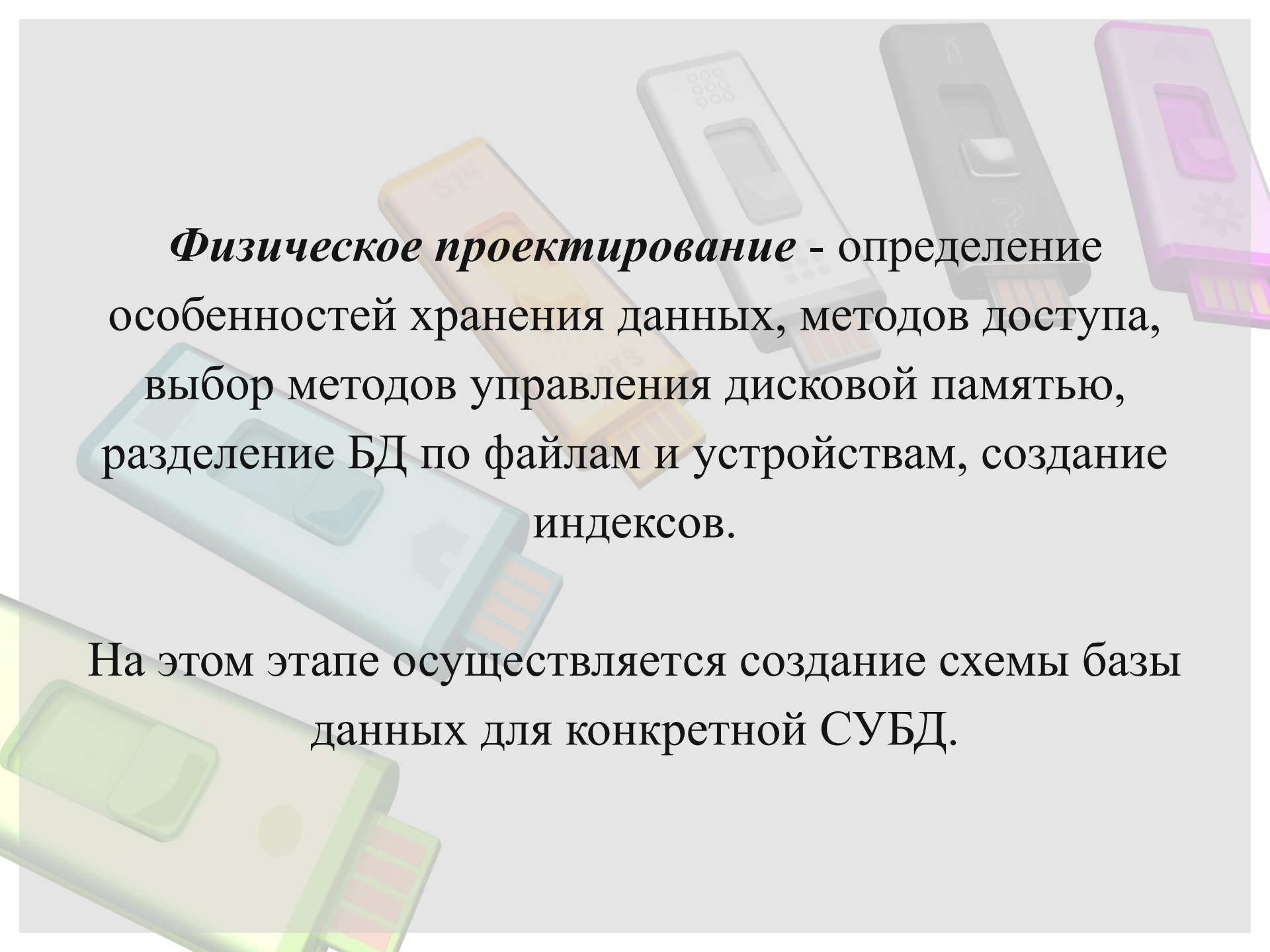
- обследование предметной области, изучение ее информационной структуры
- выявление всех фрагментов, каждый из которых характеризуется пользовательским представлением, информационными объектами и связями между ними, процессами над информационными объектами
- моделирование и интеграция всех представлений

По окончании данного этапа получаем концептуальную модель. Часто она представляется в виде модели "сущность-связь".

**Логическое проектирование** — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных.

*Для каждой таблицы указывается ее название, перечень полей и первичный ключ. Идентифицируются связи между таблицами. В рамках логического проектирования БД могут формулироваться ограничения целостности, приниматься решения о создании индексов и т. д.*

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.



***Физическое проектирование*** - определение особенностей хранения данных, методов доступа, выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, создание индексов.

На этом этапе осуществляется создание схемы базы данных для конкретной СУБД.



# Этапы проектирования БД

**КОНЦЕПТУАЛЬНЫЙ  
УРОВЕНЬ**  
сущности  
атрибуты  
связи

Представление аналитика

**ЛОГИЧЕСКИЙ УРОВЕНЬ**  
записи  
элементы данных  
связи между записями

Представление  
программиста

**ФИЗИЧЕСКИЙ УРОВЕНЬ**  
группирование данных  
индексы  
методы доступа

Представление  
администратора

физическое проектирование

# Пример создания БД