

Лекция № 5-6

**Работа с базами данных**

# Информационные системы

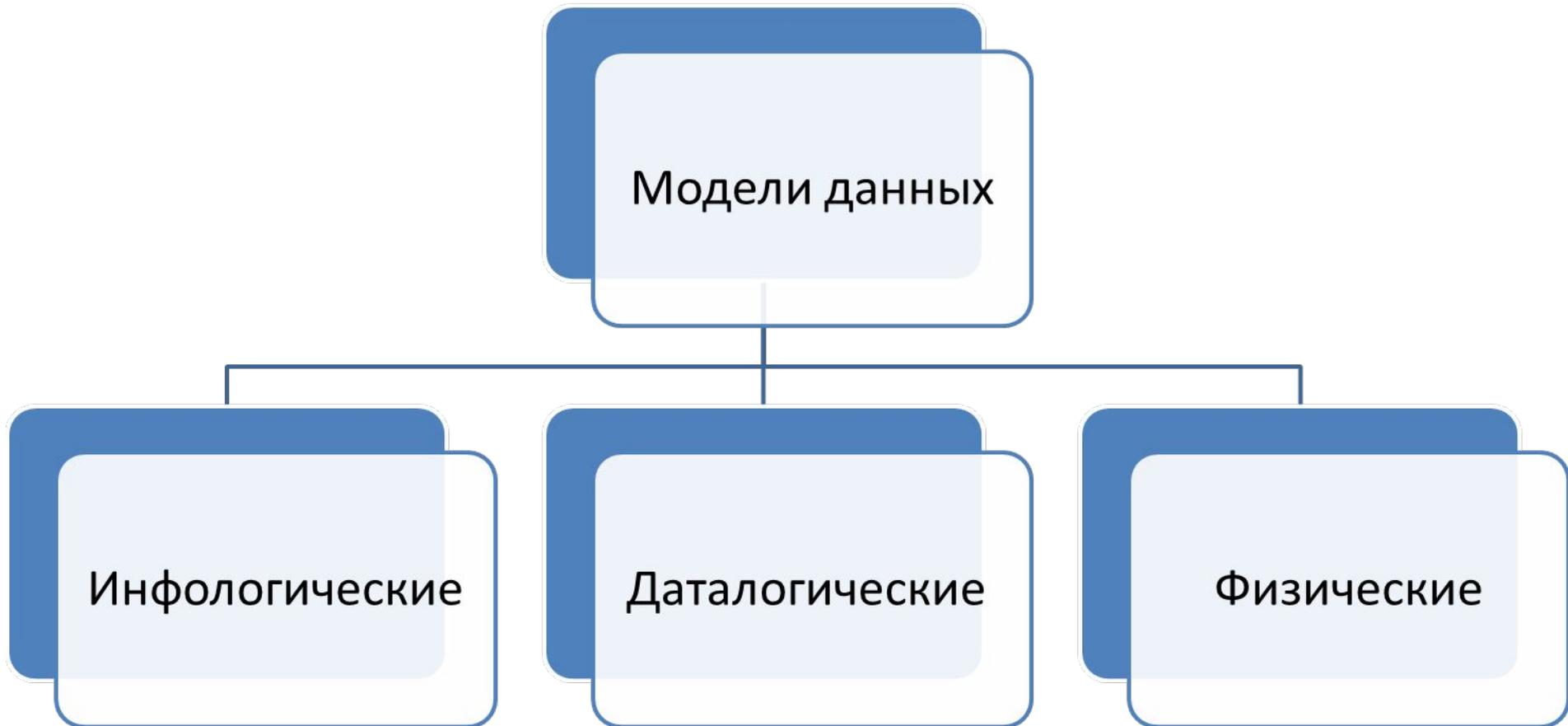
- В 21 веке одним из важнейших условий эффективного функционирования любой организации является наличие развитой автоматизированной информационной системы.
- **Информационная система**
  1. реализует автоматизированный сбор, обработку и манипулирование данными,
  2. содержит технические средства обработки данных,
  3. содержит программное обеспечение и обслуживающий персонал.
- Важнейшей структурной частью информационных систем являются модели данных и базы данных.

# Модели данных

- **Модель данных** – это набор правил, по которым организуются данные.
- Это очень простое определение можно уточнить.
- **Модель данных** – это некоторая абстракция, которая, будучи приложена к конкретным данным, позволяет пользователям и разработчикам трактовать их как информацию, то есть сведения, содержащие не только данные, но и взаимосвязи между ними.

# Классификация моделей данных

- Принято выделять три группы моделей данных:



# Классификация моделей данных – инфологическая модель

- **Инфологическая (семантическая) модель** – это обобщённое, не привязанное к какой-либо ЭВМ и СУБД описание предметной области. Это описание, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств объединяет частные представления о содержимом базы данных, полученные в результате опроса пользователей, и представления разработчиков о данных, которые могут потребоваться в будущих приложениях.
- Такая **человеко-ориентированная модель** полностью независима от физических параметров среды хранения данных. Поэтому инфологическая модель не должна изменяться до тех пор, пока она адекватно отражает предметную область, то есть до тех пор, пока не произошли изменения в предметной области.

# Классификация моделей данных – даталогическая и физическая модели

- **Даталогические модели** являются компьютерно-ориентированными, они поддерживаются конкретными СУБД. С их помощью СУБД даёт возможность пользователям осуществлять доступ к хранимым данным лишь не заботясь об их физическом расположении. Так как доступ к данным осуществляется с помощью конкретной СУБД, то даталогические модели описываются на *языке описания данных* используемой СУБД.
- **Физические модели.** Нужные данные отыскиваются СУБД на внешних запоминающих устройствах по *физической модели* данных. Физическая модель оперирует категориями, относящимися к организации внешней памяти и структурам хранения данных, которые используются в данной операционной среде.

# Информационные системы

Современной формой информационных систем являются **банки данных**, которые включают в свой состав

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

**Основными функциями банков данных** являются:

- )хранение данных и их защита;
- изменение (обновление, добавление и удаление) хранимых данных;
- поиск и отбор данных по запросам пользователей;
- .обработка данных и вывод результатов.

# ОСНОВНЫЕ ПОНЯТИЯ БАЗ

## ДАнных

### • БАЗЫ ДАнных И СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАнных

- **База данных (БД) — это организованная структура хранения данных и информации.**
- *Данные и информация — понятия взаимосвязанные, но не тождественные. Для получения информации из данных должен существовать метод ее получения.*
- Система управления базами данных (СУБД) размещает в своих структурах не только данные, но и методы (программный код), с помощью которых происходит взаимодействие с потребителем или с другими программно-аппаратными комплексами.
- **Система управления базами данных (СУБД) — это комплекс программных средств создания, ведения и совместного использования БД многими**

# ПРИМЕРЫ БД И СУБД

- **База данных крупного банка.** В ней есть все необходимые сведения о клиентах, об их адресах, кредитной истории, состоянии расчетных счетов, финансовых операциях и т. д.
- Доступ к этой базе имеется у достаточно большого количества сотрудников банка.
- Отсутствует лицо, которое имеет доступ *ко всей базе полностью* и при этом способно единолично вносить в нее произвольные изменения.
- База содержит *методы и средства*, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию.
- В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

# ПРИМЕРЫ БД И СУБД

- База данных лечебно-профилактического учреждения (ЛПУ).
- Содержит необходимые сведения о пациентах, их разнообразные персональные данные, истории болезней, состояние и физиологические показатели организма, рекомендации и назначения и т. д.
- Доступ к этой базе имеется у достаточно большого количества сотрудников ЛПУ – врачей, лаборантов, медсестер, медстатистиков, фармацевтов, финансистов, руководителей ЛПУ.
- Отсутствует лицо, которое имеет доступ *ко всей базе полностью* и при этом способно единолично вносить в нее произвольные изменения.
- База содержит *методы и средства*, позволяющие каждому из сотрудников оперировать только с теми данными, которые входят в его компетенцию.
- В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным сотрудникам, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и

***Важнейшим достоинством применения БД в информационных системах является обеспечение независимости данных от прикладных программ. Это позволяет не обременять пользователей проблемами представления данных на физическом уровне.***

# Структура СУБД

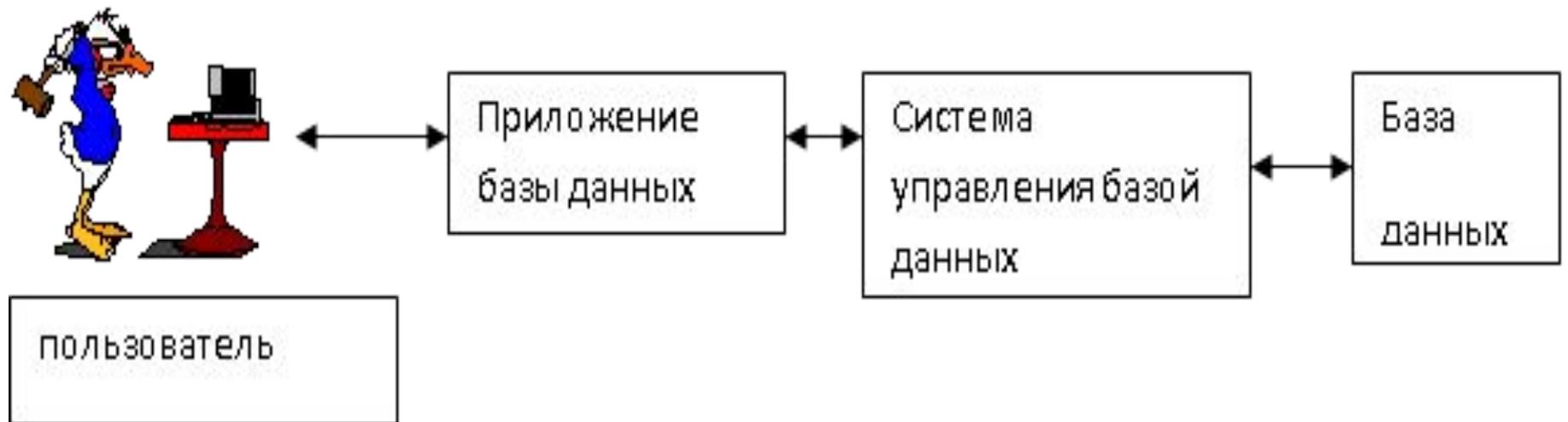


Рис. 1.1. Компоненты системы базы данных

# Структура СУБД

СУБД - интерфейс между прикладными программами и базой данных, обеспечивающий независимость данных.

## Основные функции СУБД.

- . Определение структуры создаваемой базы данных, ее инициализация и проведение начальной загрузки
- . Предоставление пользователям возможности манипулирования данными (выборка необходимых данных, выполнение вычислений, визуализация).
- . Обеспечение независимости прикладных программ и данных.
- . Защита логической целостности баз данных.
- . Защита физической целостности баз данных.
- . Управление полномочиями пользователей на доступ к базам данных.
- Синхронизация работы нескольких пользователей.
- Управление ресурсами среды хранения.
- Поддержка деятельности системного персонала.

- По характеру использования СУБД делят на *персональные* и *многопользовательские*.
- ***Многопользовательские СУБД*** включают в себя сервер БД и клиентскую часть, работают в неоднородной вычислительной среде — допускаются разные типы ЭВМ и различные операционные системы.
- ***Персональные СУБД*** обеспечивают возможность создания персональных БД и недорогих приложений, работающих с ними, и при необходимости создания приложений, работающих с сервером БД.

# МОДЕЛИ СУБД

В настоящее время наиболее распространены 3 модели управления базами данных

1. Иерархическая
2. Сетевая
3. Реляционная

Кроме них используются также и объектно-ориентированная модель

# Иерархические базы данных

- . В этой модели имеется один главный объект (корень) и остальные - подчиненные - объекты, находящиеся на разных уровнях иерархии. Взаимосвязи объектов образуют иерархическое дерево с одним корневым объектом.
- Иерархическая БД состоит из упорядоченного набора нескольких экземпляров одного типа дерева. Автоматически поддерживается целостность ссылок между предками и потомками. Основное правило: никакой потомок не может существовать без своего родителя

# Иерархические базы данных

## Пример

« Доку... » Деактивация радиоактивных отходов - 2005 Периодическая система элементов Д. И. Мендел... Поиск: Деактивация радиоактив...

Упорядочить ▾ Просмотр ▾ Общий доступ ▾ Печать Электронная почта Новая папка

Избранное

- Компьютер
- Загрузки
- Моя музыка

Рабочий стол

- Библиотеки
- Видео
- Документы**
- Изображения
- Музыка

Домашняя группа

- vita
- AppData
- Application Data
- Contacts
- Cookies
- Desktop
- Local Settings
- NetHood
- PrintHood
- Recent
- Roaming
- SendTo
- Главное меню
- Загрузки
- Избранное

**Библиотека "Документы"** Упорядочить: Папка ▾  
Деактивация радиоактивных отходов - 2005 Периодическая система элементов Д. И. Мен...

Имя	Дата изменения	Тип	Размер
1	14.01.2014 21:48	Файл	2 КБ
11060_html_1aa9deb3	14.01.2014 21:48	Файл "GIF"	1 КБ
47704_190_23413-383966-06b	14.01.2014 21:48	Файл "JPEG"	29 КБ
48918_200	14.01.2014 21:48	Файл "GIF"	96 КБ
49558_190_2312363_vb	14.01.2014 21:48	Файл "JPG"	31 КБ
50691_32387_T8[2]	14.01.2014 21:48	Файл "GIF"	88 КБ
50937_200_34-510-502	14.01.2014 21:48	Файл "JPEG"	30 КБ
50945_40381_200 (1)	14.01.2014 21:48	Файл "GIF"	343 КБ
51137_70_50440_200	14.01.2014 21:48	Файл "JPEG"	7 КБ
51147_190_20023	14.01.2014 21:48	Файл "JPG"	30 КБ
51160_20191_5	14.01.2014 21:48	Файл "GIF"	97 КБ
51462_27019_v2485	14.01.2014 21:48	Файл "GIF"	65 КБ
2299824_vb	14.01.2014 21:48	Файл "JPG"	9 КБ
2352554_vb	14.01.2014 21:48	Файл "JPG"	15 КБ
2545005_vb	14.01.2014 21:48	Файл "JPG"	12 КБ
<b>2597671_vb</b>	14.01.2014 21:48	Файл "JPG"	8 КБ
2699441_vb	14.01.2014 21:48	Файл "JPG"	11 КБ
adv_out(1).php	14.01.2014 21:48	Файл "PHP"	9 КБ
adv_out(2).php	14.01.2014 21:48	Файл "PHP"	17 КБ
adv_out	14.01.2014 21:48	Файл сценария JS...	8 КБ
adv_out.php	14.01.2014 21:48	Файл "PHP"	9 КБ
box	14.01.2014 21:48	CSS-документ	6 КБ
dl	14.01.2014 21:48	Файл сценария JS...	7 КБ

2597671\_vb Дата съемки: Укажите дату и время Оценка: ☆☆☆☆☆ Размер: 7,75 КБ Авторы: Добавьте автора  
Файл "JPG" Ключевые слова: Добавьте ключевое сл... Размеры: 200 x 199 Название: Добавьте название

WI-FI\_65  
Доступ к Интернету

# Иерархические базы данных

- + Эффективное использование памяти ЭВМ
- + Неплохие показатели времени выполнения операций над данными
- - Громоздкость для обработки информации с достаточно сложными логическими связями.
- - Сложность понимания для обычного пользователя

# Сетевые базы данных

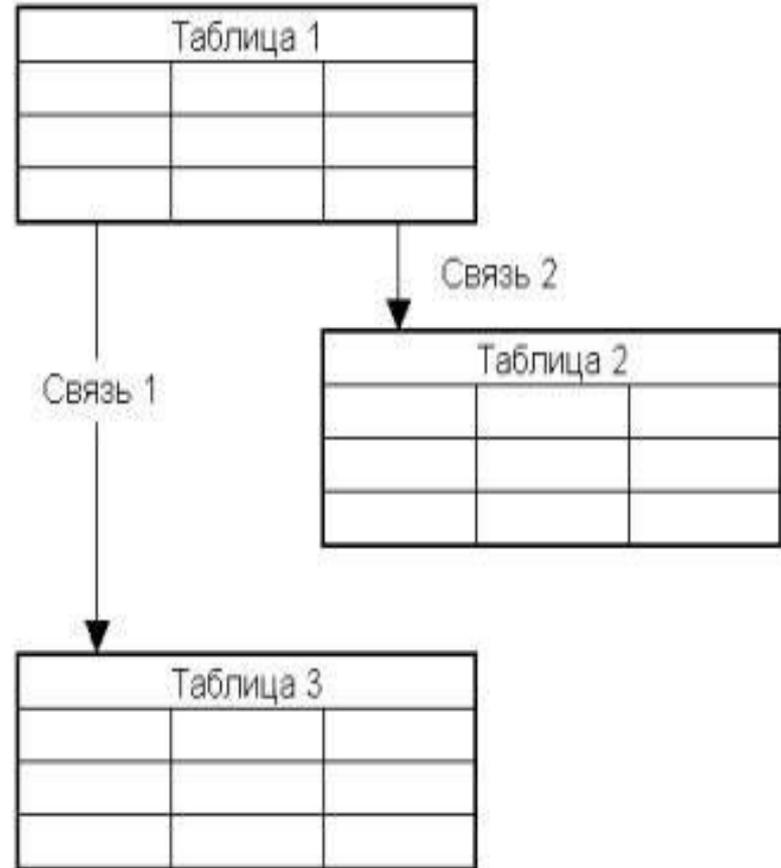
В сетевой структуре данных потомок может иметь любое число предков; любой объект может быть одновременно и главным, и подчиненным, и может участвовать в образовании любого числа взаимосвязей с другими объектами.

Сетевая БД состоит из набора записей и набора связей между этими записями.

- + Возможность эффективной реализации по затратам памяти и оперативности обработки
- Сложность и жесткость БД
- Понижен контроль целостности данных
- Большое количество связей

# Реляционные базы данных

- **Реляционная модель** использует представление данных в виде таблиц (реляций, связей). В ее основе лежит математическое понятие теоретико-множественного отношения: она базируется на реляционной алгебре и теории отношений.
- **Реляционная модель ориентирована на организацию данных в виде двумерных таблиц.**



# Реляционные базы данных

Каждая реляционная таблица представляет собой двумерный массив и обладает следующими свойствами:

- \* каждый элемент таблицы — один элемент данных
- \* все ячейки в столбце таблицы однородные, то есть все элементы в столбце имеют одинаковый тип (числовой, символьный и т. д.)
- \* каждый столбец имеет уникальное имя
- \* одинаковые строки в таблице отсутствуют
- \* порядок следования строк и столбцов может быть произвольным

# Реляционные базы данных

## *Достоинства реляционной модели:*

- \* простота и доступность для понимания пользователем. Единственной используемой информационной конструкцией является "таблица";
- \* строгие правила проектирования, базирующиеся на математическом аппарате;
- \* полная независимость данных. Изменения в прикладной программе при изменении реляционной БД минимальны;
- \* для организации запросов и написания прикладного ПО нет необходимости знать конкретную организацию БД во внешней памяти.

## *Недостатки реляционной модели:*

- \* далеко не всегда предметная область может быть представлена в виде "таблиц";
- \* в результате логического проектирования появляется множество "таблиц". Это приводит к трудности понимания структуры данных;
- \* БД занимает относительно много внешней памяти;
- \* относительно низкая скорость доступа к данным.

# ориентированные базы данных (ОО БД)

Объектно-ориентированные модели данных - используются понятия класса, объекта, метода.

В объектно-ориентированной базе данных хранятся не только данные, но и методы их обработки в виде программного кода.

**Хорошо подходят для приложений со сложными типами данных**

- Специализированные медицинские системы с программами компьютерного моделирования
- Системы составных документов, объединяющих текст, графику и электронные таблицы – электронные медицинские карты.

**Свойство ОО БД - понятийная сложность.**

# ориентированные базы данных (ОО БД). Пример.

ОО БД естественным образом отражают иерархию разнородных данных.

Весь документ представлен как один объект, состоящий из разнородных объектов, которые, в свою очередь, состоят из еще более мелких объектов.



# Проектирование структур баз данных.

В настоящее время в большинстве современных баз данных используется реляционная модель, описанная **Коддом** в строгих математических терминах в 1970 г.

**Реляционная модель** ориентирована на организацию данных в виде двумерных таблиц и базируется на двух утверждениях:

- база данных должна состоять из таблиц и только из таблиц. Только содержимое таблиц определяет операции БД;

- описание данных и манипуляции над ними должны быть независимыми от способа хранения данных на нижнем уровне.

# СУРБД

Системы управления реляционными базами данных (**СУРБД**) должны обеспечивать свою собственную систему управления, основанную только на логическом представлении данных. СУРБД включает

- Язык для оперирования с реляционными структурами - структурированный язык запросов SQL (Structured Query Language).
- При использовании реляционной модели СУБД пользователь работает с логической структурой данных.
- В большинстве настольных СУРБД пользователь не имеет доступа ко внутреннему представлению данных, СУРБД самостоятельно работают с файлами. Пользователь определяет только отношения и связи между ними.

# Реляционная таблица. Свойства

1. Каждая реляционная таблица представляет собой двумерный массив.
2. Каждая таблица состоит из однотипных строк и имеет уникальное имя.
3. Строки имеют фиксированное число полей (столбцов) и значений (множественные поля и повторяющиеся группы недопустимы). Иначе говоря, в каждой позиции таблицы на пересечении строки и столбца всегда имеется в точности одно значение или ничего.
4. Строки таблицы обязательно отличаются друг от друга хотя бы единственным значением, что позволяет однозначно идентифицировать любую строку такой таблицы.

# Реляционная таблица. Свойства

5. Столбцам таблицы однозначно присваиваются имена, и в каждом из них размещаются однородные значения данных (даты, фамилии, целые числа или денежные суммы).
6. Полное информационное содержание базы данных представляется в виде явных значений данных и такой метод представления является единственным. В частности, не существует каких-либо специальных "связей" или указателей, соединяющих одну таблицу с другой.
7. При выполнении операций с таблицей ее строки и столбцы можно обрабатывать в любом порядке безотносительно к их информационному содержанию. Этому способствует наличие имен таблиц и их столбцов, а также возможность выделения любой их строки или любого набора строк с указанными признаками

# Реляционная таблица. Первичный ключ

Каждая таблица в БД должна иметь **первичный ключ**.

**Первичный ключ** – это поле или группа полей, которые гарантируют уникальность записи.

- При разработке таблицы в качестве первичного ключа следует выбирать столько полей, сколько требуется для того, чтобы каждая запись таблицы была уникальной.
- Одни таблицы содержат одно поле, уникально идентифицирующее каждую запись – **простой ключ**.
- Другие таблицы могут потребовать первичного ключа, состоящего из комбинации полей - **составного ключа (composite key)**.

# Реляционная таблица. Первичный ключ

Даже если таблица имеет составной первичный ключ, он может быть только один.

- **Построение первичного ключа является обязательным.**
- Данные часто имеют **естественный ключ (natural key)**.
- При выборе ключа надо проявлять осторожность, так как некоторые данные только кажутся уникальными.
- Например, фамилия и имя, наименование фирмы и дата заказа.
- Если данные не содержат естественного первичного ключа, то он должен быть создан - **искусственный ключ (artificial key)**.

# Реляционная таблица. Первичный ключ

Реляционная теория требует, чтобы данные унифицировались уникально по трём критериям:

- .таблицей, где хранится этот элемент данных;
- .названием поля в этой таблице;
- .значением первичного ключа для записи.

# Реляционная таблица. Связь

- **Связь - это функциональная зависимость между сущностями.** Если между некоторыми сущностями существует связь, то факты из одной сущности ссылаются или некоторым образом связаны с фактами из другой сущности.
- Поддержание непротиворечивости функциональных зависимостей между сущностями называется ссылочной целостностью.
- Поскольку связи находятся “внутри” реляционной модели, реализация ссылочной целостности может выполняться как приложением, так и самой СУБД (с помощью механизмов декларативной ссылочной целостности и триггеров)

# Реляционная таблица. Связь

**Связи могут быть представлены пятью основными характеристиками:**

- 1. тип связи** (идентифицирующая, не идентифицирующая, полная/ неполная категория, неспецифическая связь);
- 2. родительская сущность;**
- 3. дочерняя (зависимая) сущность;**
- 4. мощность связи** (cordiallity);
- 5. допустимость пустых (null) значений.**

Использование ненормализованных таблиц может привести к нарушению целостности данных в БД.

# Определение отношений между таблицами

- Следующий шаг при разработке БД в реляционной модели, это установление связей между таблицами. Чем грамотней заданы связи между таблицами, тем проще извлекать информацию из базы данных.
- При описании отношений подразумевается связь между записями разных таблиц. Например, если упоминается **связь типа одна-ко-многим**, то имеется в виду, что одна запись некоторой таблицы связана со многими записями другой таблицы. Ни в коем случае не имеется в виду связь одной таблицы со многими таблицами

# Определение отношений между таблицами

- Простейшая связь между записями таблиц - это **одна-к-одной**. Связь такого типа осуществляется, когда связываемые таблицы имеют одинаковый первичный ключ. Чаще всего этот тип связи используется при наличии таблицы с большим числом полей, некоторые из которых являются второстепенными не столь значимыми.
- Наиболее распространён тип связи **одна-ко-многим**. Например, клиент и заказы: один клиент может сделать много заказов. Поля, по которым осуществляются связи, не являются свободными, то есть не могут иметь произвольные значения. Например, в заказе должен быть упомянут клиент, который есть в таблице «Клиенты». С точки зрения таблицы «Клиенты» поле «ФИО клиента» может быть произвольным, так как не зависит от полей других таблиц.

# Определение отношений между таблицами

- Если связаны все ключевые поля одной таблицы и часть ключевых полей другой таблицы, то тип связи может быть только одна-ко-многим.
- Тип связи **много-ко-многим** возникает, если связаны поля, частично входящие в первичный ключ одной и другой таблицы. Например, поле «Наименование продукта» в таблице «Заказы» и поле «Наименование продукта» в таблице «Отчисления». Продукт может быть заказан несколькими клиентами, а отчисления по продукту идут разным специалистам за каждую продажу продукта (если таблица «Отчисления» имеет два поля в первичном ключе - название продукта и специалист или название продукта и менеджер)

## Access

- MS Access наиболее популярная СУРБД для разработки настольных баз данных (БД).
- Набор мастеров позволяет полностью автоматизировать создание пользовательского интерфейса
- Множество средств разработки помогает сократить время создания законченного приложения. Полная интеграция с другими приложениями MS Office – также большой плюс Access. Встроенный язык Visual Basic for Application (VBA) позволяет выполнять нестандартные задачи.

# Access

Microsoft Access предоставляет несколько **средств создания** каждого из **основных объектов базы**. Эти средства можно классифицировать как:

- **ручные** - разработка объектов в режиме Конструктора;
- **автоматизированные** - разработка с помощью программ-мастеров;
- **автоматические** – средства ускоренной разработки простейших объектов.

# Объекты базы данных

**Таблицы** - основные структурные элементы базы данных. Предназначены для хранения данных в виде **записей** (строк) и **полей** (столбцов). Данные в отдельной таблице обычно относятся к определенной категории. Таблицы хранят структуру базы: поля, их типы и свойства.

**Запросы** – объекты, позволяющие получить нужные данные из одной или нескольких таблиц.

**Формы** – объекты, предназначенные, в основном, для ввода данных. В форме можно разместить элементы управления (счётчики, раскрывающиеся списки, флажки, переключатели и др.), облегчающие ввод и изменение данных в полях. Кроме того формы можно использовать для отображения данных.

# Объекты базы данных

**Отчёты** – объекты, предназначенные для вывода данных на печать.

**Макрос** – макрокоманда или набор макрокоманд, используемый для автоматического выполнения некоторых операций.

**Модуль** – это совокупность описаний, инструкций и процедур, сохраненная под общим именем. Модули создаются средствами внешнего языка программирования, в данном случае языка Visual Basic for Application (VBA 6.0.). Позволяют реализовать нестандартные функциональные возможности, повысить уровень защищённости и др.

**На этом все...**

**Благодарю  
за внимание !!!**