

# Базы данных Введение

Бессарабов Н.В.

[bes@fpm.kubsu.ru](mailto:bes@fpm.kubsu.ru)

2019 г.

# Особенности курса

Знание немногих принципов освобождает от знания многих фактов. Р. Декарт

- Базы данных (БД) рассматриваются с одной стороны как отображения моделей бизнеса, с другой как компоненты информационных систем, с третьей стороны, учитываются особенности реализации систем управления базами данных (СУБД).
- Предлагается вариант семиотического подхода, что совсем не обычно для подобных курсов. Рассматривается семантика и прагматика данных, которые в подобных курсах почти не изучаются.
- В настоящее время СУБД используют большое количество моделей данных. Разобраться в этом зоопарке можно работая с такими моделями и моделями бизнеса, а также исследуя их морфизмы.
- Используемый инструментарий (СУБД Caché и Oracle, CASE-средство ERWin и модель WinRDBI) выбран так, чтобы за ограниченное время освоить шесть моделей данных и получить представление о моделях NoSQL. В частности, внимание, уделяемое Caché вызвано тем, что в ней удобно изучать пять моделей данных, а морфизмы трёх из них просто встроены в эту СУБД.

**Важно понимать, что без достаточно большой практической работы ничто не понимается до конца и не запоминается надолго.**

# Разделы курса

## Обязательные:

- Общие понятия.
- Модели данных. Семантика и прагматика данных. Понятие СУБД.
- Инфологические модели. Модель “Сущность – связь”.
- Иерархическая модель. Язык Caché ObjectScript.
- Реляционная модель.
- Нормализация.
- Языки SQL и QBE.
- Транзакции.
- Объектная и объектно-реляционная модели данных.
- Морфизмы объектных, реляционных и иерархических моделей.
- Хранение данных и доступ к ним. Планы исполнения.

## Необязательные:

- Представления о моделях NoSQL и NewSQL.
- Шаблоны/паттерны проектирования баз данных

# Что полезно изучить в университете и освоить самостоятельно

- Любые курсы по Oracle, особенно по SQL, PL/SQL, аналитическим функциям, основам администрирования, Apex. Можно изучать любые “большие” СУБД – DB2, SQL Server, PostgreSQL
- Язык и технологии Java, HTML5, CSS3, JavaScript, PHP, Python
- XML (XSL, XSLT, Xpath, Xquery, ...)
- Облачные вычисления, базы данных в облаках
- Администрирование баз данных
- Технологии программирования (читается не на всех направлениях)

Настоятельно рекомендую:

Глас вопиющего в пустыне!

- Принять участие в разработке серьёзного проекта (самое главное!)
- Изучать современную алгебру, в том числе теорию категорий, логики, в том числе, неклассические, и теорию систем (лучше по Урманцеву)
- Изучать английский язык (проблемы с литературой на русском)
- Готовиться к сдаче экзамена на сертификат Oracle (SQL, PL/SQL, MySQL, Java) или других СУБД.

Обеспеченность  
литературой  
(100 экз.)  
+ сайт ИНТУИТ

Учебное пособие:  
[www.intuit.ru](http://www.intuit.ru) (HTML,  
.fb2).

Слайды лекций  
рассчитаны не только  
на представление  
материала во время  
лекций, но  
представляют почти  
полный конспект.



**Основы информационных технологий**

Н. В. Бессарабов

**БАЗЫ ДАННЫХ**

**Модели, языки, структуры и семантика**

Учебное пособие

Москва

Национальный открытый университет «ИНТУИТ»

2013

# Цели лекции

В первой части лекции дано предварительное определение базы данных.

Вначале выясняется что такое “данные”, их “семантика” и “прагматика”. Наиболее распространены базы хранящие наборы записей. Определяются поля записей, наборы допустимых значений полей, называемых доменами, и схемы (типы) записей. Базы данных определяются как структурированные собрания записей, обладающие свойством сохраняемости и, может быть, свойством самоописания.

Во второй части изучены условия, определяющие допустимые значения данных. Их называют ограничениями целостности.

Определяются модели данных и системы управления базами данных.

Рассматриваются имеющиеся в бизнес-приложениях неопределённые значения.

В третьей части лекции БД рассматриваются как модели бизнеса. Именно с этих моделей начинается проектирование информационных систем, через отображения в них определяются некоторые семантики данных. С некоторыми понятиями, например, с аномалиями, просто невозможно разобраться до конца, не учитывая этот аспект.

Не смущайтесь, если что-то пока останется не выясненным до конца. Многое, но к сожалению, не всё, будет повторено и уточнено.

# Часть 1.

## Данные, семантика и прагматика Предварительное определение базы данных.

Здесь даются предварительные определения, необходимые для быстрого вхождения в тему. Под “базой данных” временно будем понимать любое (и не обязательно электронное) средство для хранения информации.

# Понятие данных

**Данные** – это представление фактов о предметной области информационной системы в виде структуры, состоящей из знаков.

**Вопрос:** Если имеются данные по какому-то объекту, процессу, то что ещё, кроме них, может понадобиться?

**Вариант вопроса:** Разве в базе данных имеется что-либо кроме этих самых данных?

**Ответ:** Много чего. Данные должны кем-то или чем-то распознаться, а затем интерпретироваться. Сами по себе они не могут быть использованы. Нужны семантики, то есть смыслы (интерпретации) данных, и прагматики, отражающие отношение интерпретатора к данным.

Уже для восприятия данных нужно представлять, какие знаки будем воспринимать. Иначе говоря, уметь распознавать эти знаки и как-то представлять систему знаков.

Пример: Пусть имеется знак: I

Можете без дополнительной информации сказать, что это: черта, буква латиницы, или римское число 1? Если, например, ожидается римское число, то неоднозначность восприятия отпадает.

Прагматику можно понимать как выбор одной из возможных семантик.

# Синтаксис, семантики и прагматики (1/2)

Пример: (анализ семантики по косвенным признакам): На вступительных экзаменах в вуз преподаватель, не получив ответа, просит студента хотя бы прочесть выражение вида  $2xy^2 + \dots$  и получает совсем неожиданную (для преподавателя) интерпретацию “два ху и два наверху”.

Разве не понятно, насколько абитуриент разобрался с многочленами, какой семантикой (смыслом) для него обладает запись “ $2xy^2 + \dots$ ” ?

Пример: Табло на здании банка – запись с тремя полями

USD	62.00	65.00
-----	-------	-------

Интерпретатор – человек. Некоторые варианты прочтения:

1. Человек не знает, что такое валюта и как её обменивать. Ответ: “?”
2. Знает, но интересуется только покупкой долларов (такая у него прагматика). Ответ: “Сегодня здесь продают доллар США за 65 рублей”. Здесь использована уже другая семантика.
3. Знает, что такое маржа и что она характеризует. Ответ: “3 рубля. А вчера было... Ожидание обмена валюты банком ... (повышается/ понижается)”. Здесь третья прагматика и задаваемая ею семантика.

Заметьте, что в последнем варианте для интерпретации потребовались дополнительные данные.

# Синтаксис, семантики и прагматики (2/2)

*Morris Ch. W. Foundations of the theory of signs. - Chicago, 1938.*

Источник: <http://niv.ru/doc/communications/pocheptsov/126.htm>



# Уточнения к предыдущему слайду

- Стрелки на предыдущем слайде это скорее некоторое соответствие, не всегда чётко определённое. Их не следует связывать с какими-либо математическими конструкциями.
- Означаемое может трактоваться двояко:
  - как денотат, то есть класс структур, предметов, явлений обозначаемых знаком; денотативное значение это понятийное ядро значения; денотат соответствует объёму понятия в логике;
  - как десигнат, идеальный объект, обозначенный данным знаком;
- Эти уточнения необходимы потому, что терминология в изучаемой нами области, как и во многих других, используется весьма фривольно. Часто под одним названием в разных частях одного документа имеются совсем разные вещи.

# Пример

Пример 1: Дана таблица с именем “Сотрудники” и шапкой

Табельный_номер	ФИО	Должность	Заработная плата
-----------------	-----	-----------	------------------

Что хранится в таблице? Каков смысл данных?

Оказывается, в нынешнем вид таблица бессмысленна.

Как в этом убедиться?

Отообразим таблицу (это наш знак) на модель бизнес-процессов (это у нас обозначаемое – денотат). Замечаем, что в бизнесе нет ничего вечного. Сотрудника сначала не было, затем его приняли, а далее может быть перевели, и не один раз, потом уволили, снова приняли и т.д. Наша таблица этого никак не отображает и потому бессмысленна.

Замечаем, что стоит попробовать как-то привязаться к дате или отрезку времени. Например, если изменить название, скажем, на “Сотрудники на 20.02.2019”, некоторый смысл появится.

# Как описать базу данных

База данных определяется:

- тем, какие данные в ней хранятся, и какие ограничения на них поддерживаются;
- какой синтаксис, какие семантики и прагматики используются;
- тем, как хранятся данные;
- тем, что и как спрашивают (или могут спросить);
- как интерпретируются результаты выборки;
- тем, кто, при каких условиях и когда может спрашивать.

Пример: Собрание книг со следующими особенностями:

1. отдано на ответственное хранение без права чтения;
2. книги на полках расположены бессистемно;
3. книги на полках расположены по возрастанию инвентарных номеров и снабжены каталогом, в котором карточки расположенные по темам;
4. имеется поисковая система, позволяющая вести поиск данных в заглавиях и/или в текстах книг;
5. имеется система организации и учета выдачи книг.

Замечание: варианты отличаются и прагматикой, и семантикой данных.

# Поля, записи, наборы записей

Данные часто хранятся в виде записей.

- **Записью** в базах данных называют минимальную уникально идентифицируемую единицу независимого хранения данных, образованную *иерархией полей*.
- **Поле записи** – именованный элемент данных, являющийся частью структуры записи базы данных или файла.
- Обычно, но не всегда, поля **типизированы**.
- Значения полей называются **элементами данных**.
- Для уникальной идентификации записей можно выделить одно или несколько полей в качестве **ключей**. Могут создаваться специальные идентификаторы не имеющие смысла в предметной области.
- **Схема записи** -- это описание внутренней структуры записи. Схема записи определяет связную последовательность полей, образующую дерево.

Замечание 1: В базах могут храниться данные, которые не удобно представлять типизированными записями. Это картографические, мультимедийные и другие слабо структурированные или неструктурированные данные.

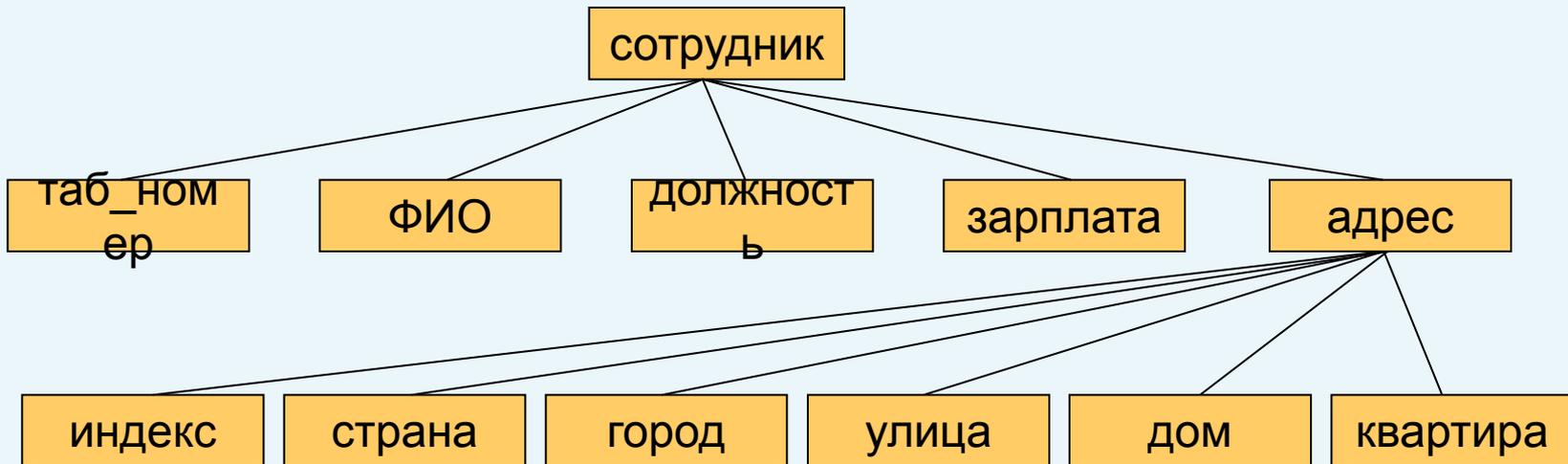
Замечание 2: Мы будем рассматривать только данные представляемые записями.

# Примеры записи, схемы записи

Пример: Запись “сотрудник”, с полем “адрес”

сотрудник(1101, Пирогов Олег Николаевич, лаборант, 2000,  
адрес(350033, Россия, Краснодар, Ставропольская, 140, 5))

Схема записи “сотрудник”:



# Типы данных

Разделим типы данных на три группы:

- **Простые** типы данных.
- **Структурированные** типы данных.
- **Ссылочные** типы данных.

Простые, иначе атомарные, или скалярные типы данных не обладают внутренней структурой. К простым типам относятся как минимум:

- строковые (с переменной и фиксированной длиной),
- численные (целый, вещественный),
- денежный (вещественный с двумя знаками после десятичной точки),
- интервальные типы (дата, время, временные метки),
- перечислимые типы.

Замечание: Не повезло в базах логическому типу. Очень часто он отсутствует. Поэтому приходится представлять его, например, символьным типом со значениями 1 как true и 0 как false.

# Структурированные типы данных

**Структурированные типы данных** образуются из составляющих компонентов, которые, в свою очередь, могут быть структурированы. Наиболее распространены структурированные типы: **массивы** и **записи**.

Пример: Структурированный тип данных запись «адрес», рассмотренный ранее на слайде 13

*сотрудник(таб\_номер, ФИО, должность, зарплата, адрес(индекс, страна, город, улица, дом, квартира))*

**Ссылочные типы данных** используются в объектных базах для определения ссылочных атрибутов, представляемых так называемыми объектными ссылками (OID и OREF). Пока мы подобными конструкциями не занимаемся.

Обратите внимание на то, что данные любых типов, за исключением структурированных, воспринимается как единое и неделимое целое. У данных структурированных типов такими неделимыми элементами являются листовые элементы дерева типа.

С точки зрения пользователя или других программ эти “неделимые” элементы могут иметь свою структуру, но система управления базами данных не может воспринимать её детали.

# Домены

**Домен** можно считать уточнением типа данных. Домен определяет подмножество значений некоторого типа данных имеющих определенный смысл.

Домен должен иметь *уникальное* в пределах базы данных имя. Определяется он на некотором типе данных или на другом домене. Домен характеризуется *условием*, выделяющем подмножество данных описываемого домена.

Пример: Домен “возраст человека” характеризуется условием (возраст>0 и возраст<120).

На этом домене с помощью условий (возраст>21 и возраст< 45) можно определить домен “возраст сотрудника охранного предприятия”.

Замечание: К сожалению, в существующих системах управления базами данных домены не поддерживаются.

# Структура набора записей

## Набор записей

А как связываются наборы записей?

### схема записи

ФИО	Адрес	Телефон
Текст(20)	Текст(35)	Текст(12)

Имена полей

Типы полей

### данные

Иванов И.И.	Ставропольская 149	1-111-111
Петров П.П.	Ставропольская 153	2-222-222

← Запись 1

← Запись 2

...

Домены задают ограничения целостности

# Схема базы

Вспомним, что схема записи это описание ее структуры.

Описание базы или ее фрагмента принято называть **схемой базы/фрагмента**. Некоторые наборы записей в схеме базы могут быть связаны. Поэтому в схему базы включаются **связи**, представляемые *как часть схем связываемых записей, либо отдельным описанием*.

Пример: в двух наборах записей “сотрудник” и “отдел” со схемой *сотрудник(табельный\_номер, ФИО, должность, таб\_ном\_начальника, зарплата, комисионные, номер\_отдела)*

*отдел ( номер\_отдела, название\_отдела, город)*

свяжем эти наборы записей через поля “номер\_отдела”, имея в виду, что у каждого сотрудника в поле “номер\_отдела” должен стоять номер отдела, который имеется в одной из записей набора “отдел” и не может быть номера, не указанного в одной из записей набора “отдел”. Остается добавить схему связи:

*связь\_сотр\_отд(сотрудник.номер\_отдела, отдел.номер\_отдела)*

При моделировании реальных объектов бизнеса, следует уточнить многие подробности, например, как-то указать, что в отделе может не быть сотрудников, что один сотрудник не может работать в двух отделах и т.д. Этими ограничениями целостности мы займёмся позже.

# Данные и метаданные

**Метаданные** это данные специального вида, которые описывают структурные свойства всех или части хранимых данных. Часть метаданных – комментарии – интерпретируется человеком. Поскольку какие-то метаданные имеются всегда, база **самодокументируема**.

Понятно, что схема базы это существенная часть её метаданных.

**Метаданные** представляют часть **семантики** “основных” данных, зависящую от реализации базы.

## Степени структурированности

Будем считать, что записи сгруппированы в некоторые агрегаты, которые здесь будем называть наборами.

Если метаданные применимы ко всем данным их набора, то соответствующая модель данных называется **структурированной**.

Если отдельные записи набора могут содержать поля, которых нет у других записей набора, модель называется **полуструктурированной**.

Модель, в которой элементы метаданных могут описывать только одну или немногие записи, называется **неструктурированной**.

# Пример. Данные и метаданные

Данные содержатся в двух таблицах. Таблица T1 приведена ниже, таблица T2 на этом слайде не показана.

T1:

ФИО	Адрес	Телефон
Иванов И.И.	Ставропольская 149	2-111-111
Петров П.П.	Ставропольская 153	2-222-222

Часть метаданных может быть записаны в двух таблицах.

M1 содержит перечень таблиц, M2 – перечень столбцов

M1:

Ном_таб	Имя_таб
1	T1
2	T2

M2:

В M1 и M2 не все метаданные

Ном_таб	Ном_столб	Имя_столб
1	1	ФИО
1	2	Адрес
1	3	Телефон
2	1	Назв_отдела

Замечание: К метаданным относятся типы, комментарии, ограничения целостности и много других данных.

# Что такое база данных (узкое и неполное псевдоопределение)

Будем понимать под **базой данных** (БД) собрание **записей**, обладающее следующими свойствами:

- Записи интегрированы в некоторые структуры (наборы, таблицы со связями, деревья, сети, агрегированные структуры и т.д.), описываемые схемами.
- База обладает **персистентностью** – она сохраняет данные при выключении питания.
- База содержит **типы данных, метаданные, ограничения целостности и комментарии** (интерпретируемые человеком).
- Данные независимы от обрабатывающих их программ. Под независимостью данных от программ понимается возможность создать структуры данных, не обращая внимание на их обработку, а затем написать программные модули, обрабатывающие данные.

Ограничения целостности будут рассмотрены позже.

Важное замечание: В структурированной модели данных начать работать с данными можно только после определения схемы данных. В полуструктурированных и неструктурированных моделях схема изначально может отсутствовать, а затем достраиваться по мере заполнения данными и постоянно меняться.

# Часть 2. Модели данных. Базы данных и файловые системы.

# Ограничения целостности

**Ограничения целостности** это условия специального вида, которые должны выполняться для некоторого набора записей (хотя бы одной), для некоторой **подсхемы** или **схемы** всей базы данных. Выделяют **декларативные** и **процедурные** ограничения.

Декларативные ограничения описываются при создании схемы базы на декларативном языке её описания. Например, ограничение “первичный ключ” (“primary key”) означает, что, задав значения ключевых полей можно выбрать запись однозначно.

Процедурные ограничения определяются процедурами специального вида, называемыми обычно триггерами и реагирующие на события из некоторого набора. Что-то вроде демонов в Unix.

Процедурные ограничения записываются в специальном процедурном языке.

Замечание: Ограничения целостности – это ещё один элемент семантики данных, задающий условия, которым должны удовлетворять.

# Примеры декларативных ограничений целостности

## 1. Ограничение “Первичный ключ”

Если имеем дело только с людьми, у которых есть ИНН, то в наборе записей

*Сотрудник (ИНН, ФИО, Должность, Зарплата)*

поле “ИНН” может использоваться как первичный ключ так как значение ИНН уникально.

## 2. Ограничение типа “Check” (Проверка)

В наборе записей

*Сотрудник (ИНН, ФИО, Должность, Зарплата, Бонус)*

для каждой записи должно выполняться условие

$\text{Бонус} < 0.2 * \text{Зарплата}$

Замечание: Ограничения Check строятся на данных одной текущей записи.

# Пример процедурного ограничения целостности

Пример процедурного ограничения: В наборе записей *сотрудник(ИНН, ФИО, Должность, Зарплата, Бонус)* предусмотреть изменение поля “Зарплата” только в сторону уменьшения.

Почему это ограничение не декларативно? Потому, что назначаемая зарплата в базе данных пока еще не записана и отношение

“Новая\_зарплата” < “Старая\_зарплата”  
нельзя выразить через данные базы.

Замечание: Поддержание ограничений целостности требует **активности** базы и реализуется процедурами работающими подобно резидентным программам. Вообще, **активность базы** это её способность совершать действия сверх непосредственно указанных ей.

# Модели данных

Определение 1 (М.Р. Когаловский): **Модель данных** это “система типов данных, типов связей между ними и допустимых видов ограничений целостности, которые могут быть для них определены”.

Определение 2 (М.Р. Когаловский): **Модель данных** это “метамодель для описания моделей предметной области в среде выбранной СУБД”.

Таким образом по отношению к данным базы модель данных это **метаметамодель**.

Замечание: в обоих определениях не полностью учтена семантика и никак не учитывается прагматика. Но временно, как рабочие определения, их можно принять.

Теперь понятно, что к определению базы, данному на слайде 21 следует добавить еще свойство:

- **База данных создается в рамках одной или нескольких моделей данных.**

Замечание: В современных СУБД число моделей м.б. до 3 -5-ти. <sup>28</sup>

# Составные части модели данных

По Дейту [3], реляционная модель, которая нами будет рассмотрена в следующих лекциях, состоит из трех частей:

- структурной,
- целостной,
- манипуляционной.

Заметьте, что семантика и прагматика в этом раскладе в явном виде отсутствуют.

Перечисленные аспекты могут выделяться в любой модели данных, но не всегда реализуются явно.



В наше определение базы (но не модели) данных следует добавить:

- способность создавать и поддерживать схемы,
- работу с ограничениями целостности,
- манипулирование данными.

Забегая вперед, отметим, что структурная часть реляционной модели, образуется отношениями и связями между ними. Единственная структура данных в реляционной модели это описанные выше  $n$ -арные отношения, связанные бедным набором связей типов 1:1 и 1: $n$ .

Семантика в базах данных реализуется как минимум типами данных, метаданными и ограничениями целостности.

# Файловые системы и базы данных

Для долговременного хранения больших объемов данных в настоящее время используют файловые системы и базы данных. И хотя данные в базе хранятся в файлах, механизмы хранения данных в базе существенно отличны. Поэтому файловые системы и базы данных имеют свои области применения.

В первом приближении можно считать, что **базы данных это надстройки над файловой системой**, обеспечивающие работу со сложными структурами данных без явного использования операций с файлами.

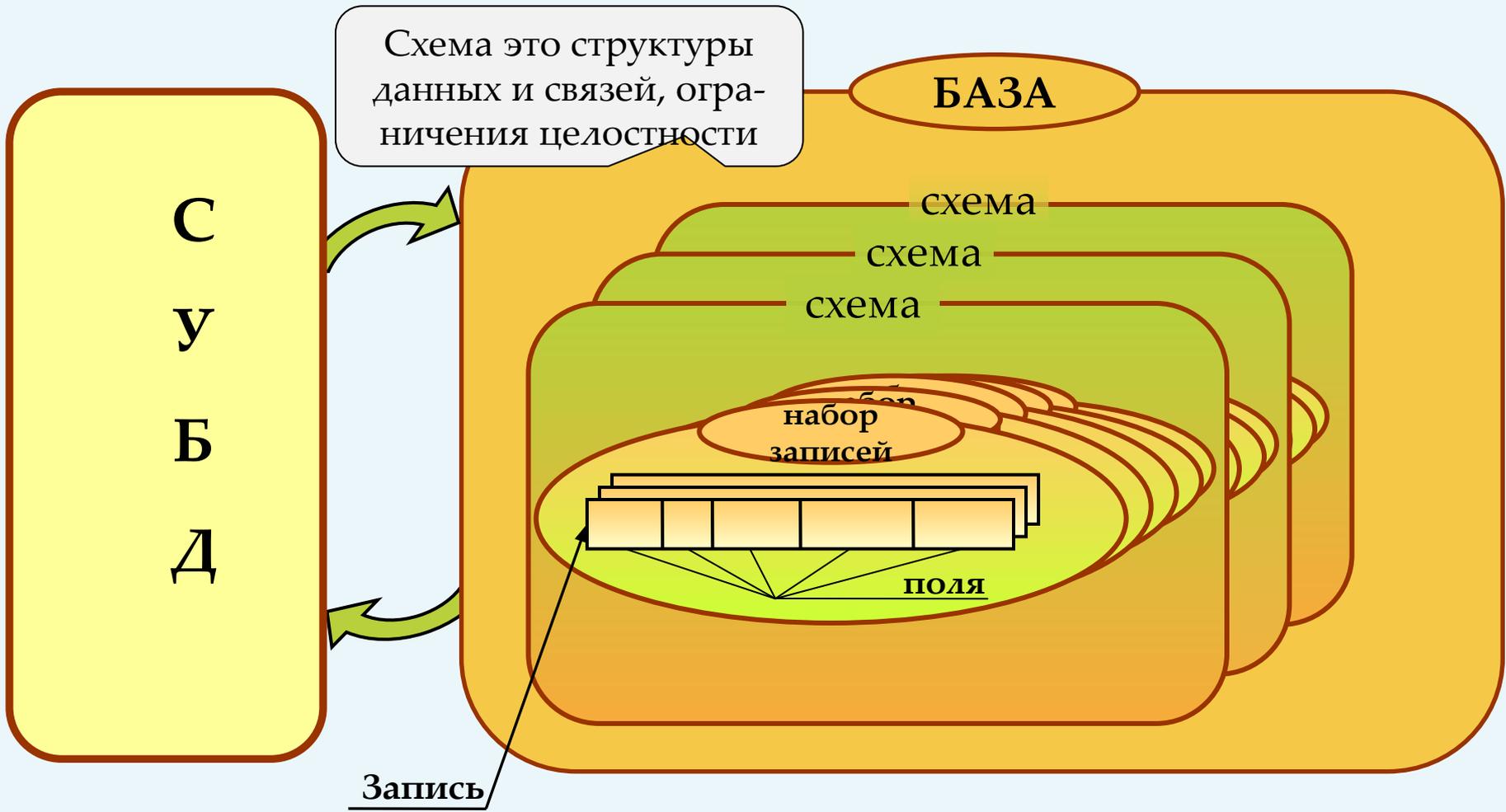
# Системы управления базами данных

Термин: Система управления базами данных (СУБД), в английском Database Management System (DBMS).

Определение: СУБД это “программная система, предназначенная для создания и хранения базы данных на основе некоторой модели данных, обеспечения логической и физической целостности содержащихся в ней данных, надежного и эффективного использования ресурсов (данных, пространства памяти и вычислительных ресурсов), предоставления к ней санкционированного доступа для приложений и конечных пользователей, а также для поддержки функций администратора базы данных”.  
(М.Р. Когаловский).

Замечание: Во многих современных больших СУБД используют как правило три модели данных – табличную, одну из объектных и иерархическую (обычно для работы с XML).

# Упрощённое представление базы данных



# Часть 3. База данных как модель бизнеса

# Модельный подход

Под бизнесом в дальнейшем изложении будем понимать любую деятельность, не обязательно связанную с извлечением прибыли.

Бизнес это набор бизнес-процессов, как-то связанных между собой.

Из-за сложности предметной области модели бизнеса многослойны.

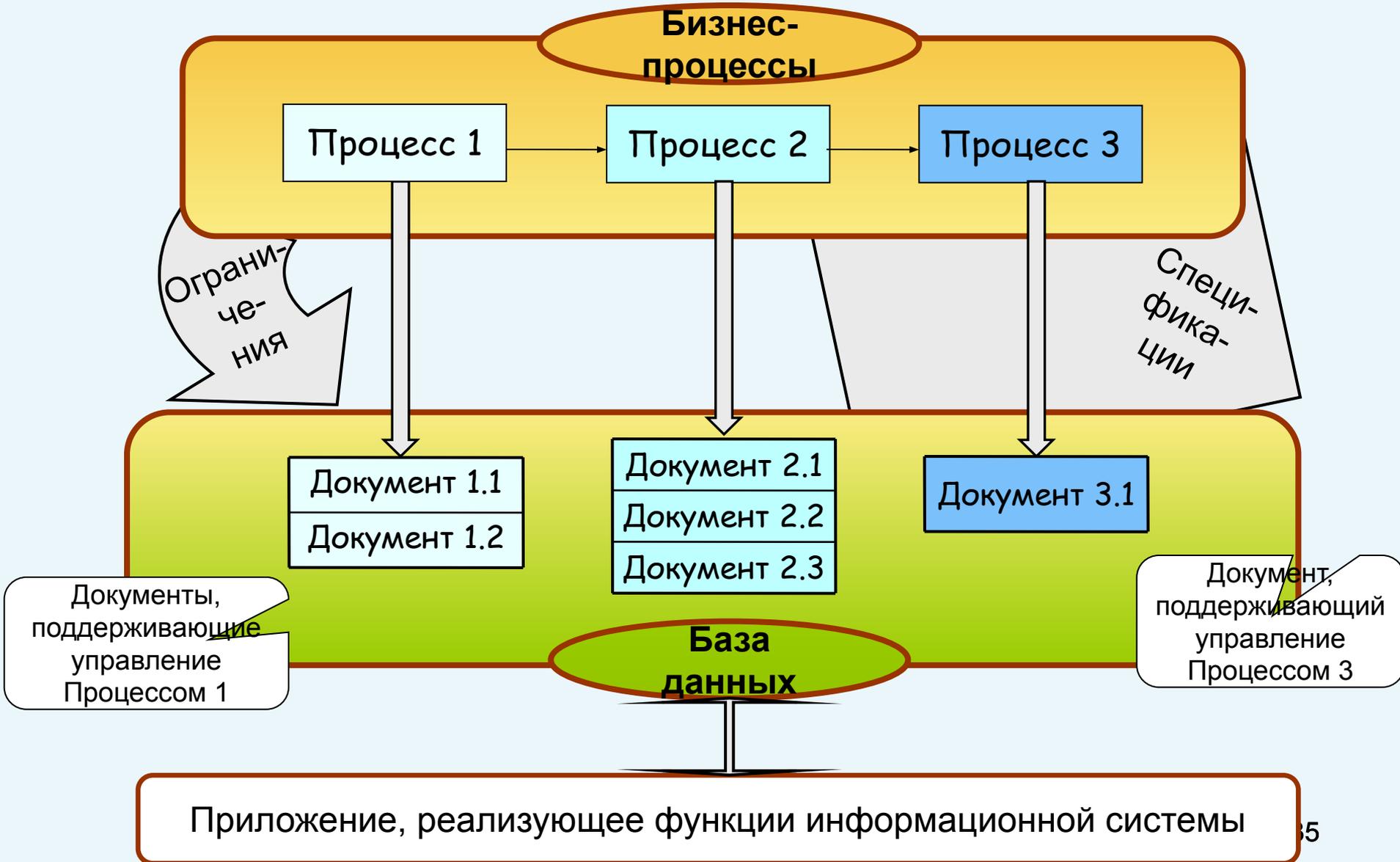
При создании информационных систем необходимо построить модель бизнес-процессов (в UML это как минимум, диаграммы прецедентов, деятельностей и последовательностей), а затем переходить к концептуальной модели базы данных. База данных это модель бизнеса, очевидно, не полная. Но она должна как можно полнее отобразить модель бизнеса.

Модельный подход очень важен и для студента, изучающего базы данных, и для постановщика задач создания информационных систем. Например, при изучении аномалий невозможно разобраться с ними до конца, не учитывая модельный аспект.

Существуют системы, для управления которыми достаточно регулировать потоки документов регламентирующих и сопровождающих бизнес-процессы. Такая документо-ориентированная система в чрезвычайно упрощённом виде представлена на следующем слайде.

# База данных как модель бизнеса

(пример документарного подхода)



# Процесс моделирования

Условно выделим этапы процесса моделирования данных:

**Первый этап:** выбор целей моделирования. Цель определяет сложность модели и возможности дальнейшего изменения модели (рефакторинга).

**Второй этап:** построение концептуальной модели. Выявляются существенные объекты, исключаются второстепенные. Задаются типы данных и домены. Результат – концептуальная схема модели. Возможно, проводится рефакторинг бизнеса.

**Третий этап:** создание логической модели.

**Четвертый этап:** создание физической модели.

**Пятый этап:** аппаратная реализация.

В современных технологиях разработки эти этапы могут выполняться не последовательно!

Концептуальная модель данных описывает данные исходя из требований бизнеса и теоретически может никак не учитывать последующие реализации. Правда, в жизни всё сложнее и хороший постановщик задачи учитывает возможности будущих реализаций.

Логическая (даталогическая) модель представляет схему данных для реализации в СУБД некоторого класса, например, табличных.

Физическая модель реализуется в конкретной СУБД.

Аппаратная реализация учитывает особенности платформы, на которой реализуется база данных.

# Общие требования к моделям

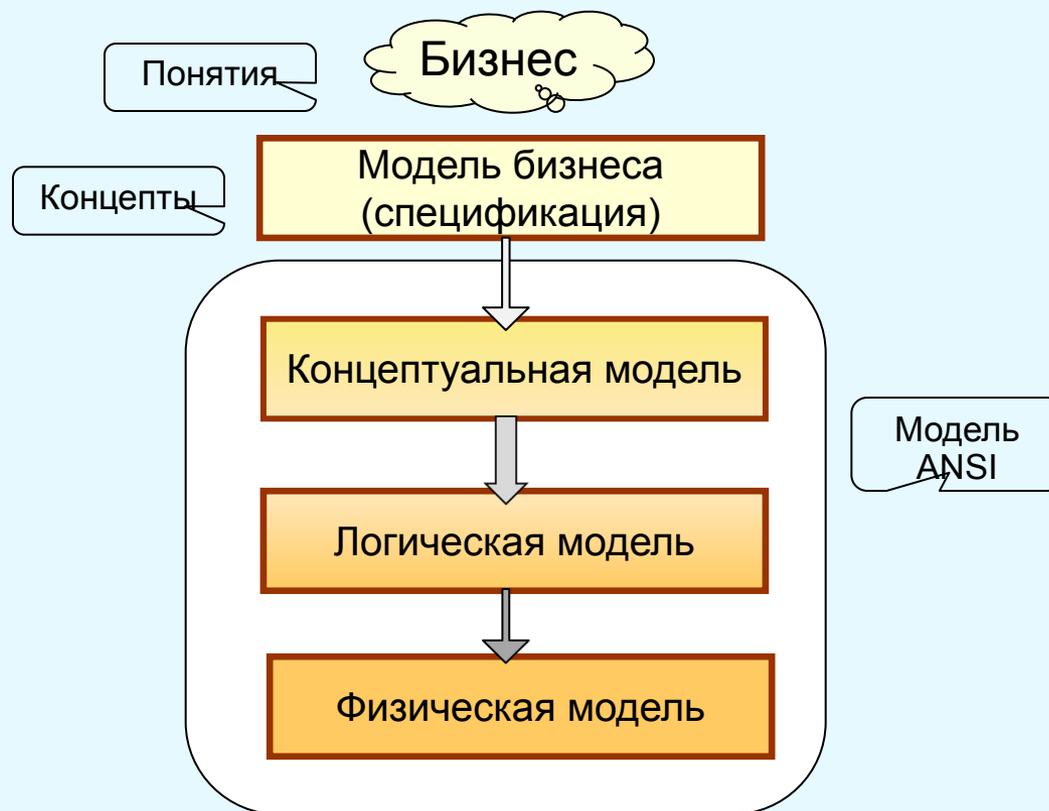
Требования к любым моделям:

- адекватность – точное, в некотором смысле, отображение объекта;
- полнота – предоставление возможности работать со всей необходимой информацией;
- непротиворечивость;
- трудоемкость разработки должна обеспечить реализацию в рамках выделенных ресурсов (денег, имеющихся кадров, выделенного времени и программных средств).

Разработка модели это разновидность инженерных задач, для которых характерна невозможность достижения оптимума по всем параметрам одновременно. Поэтому любое приемлемое решение – это некоторый набор допустимых компромиссов.

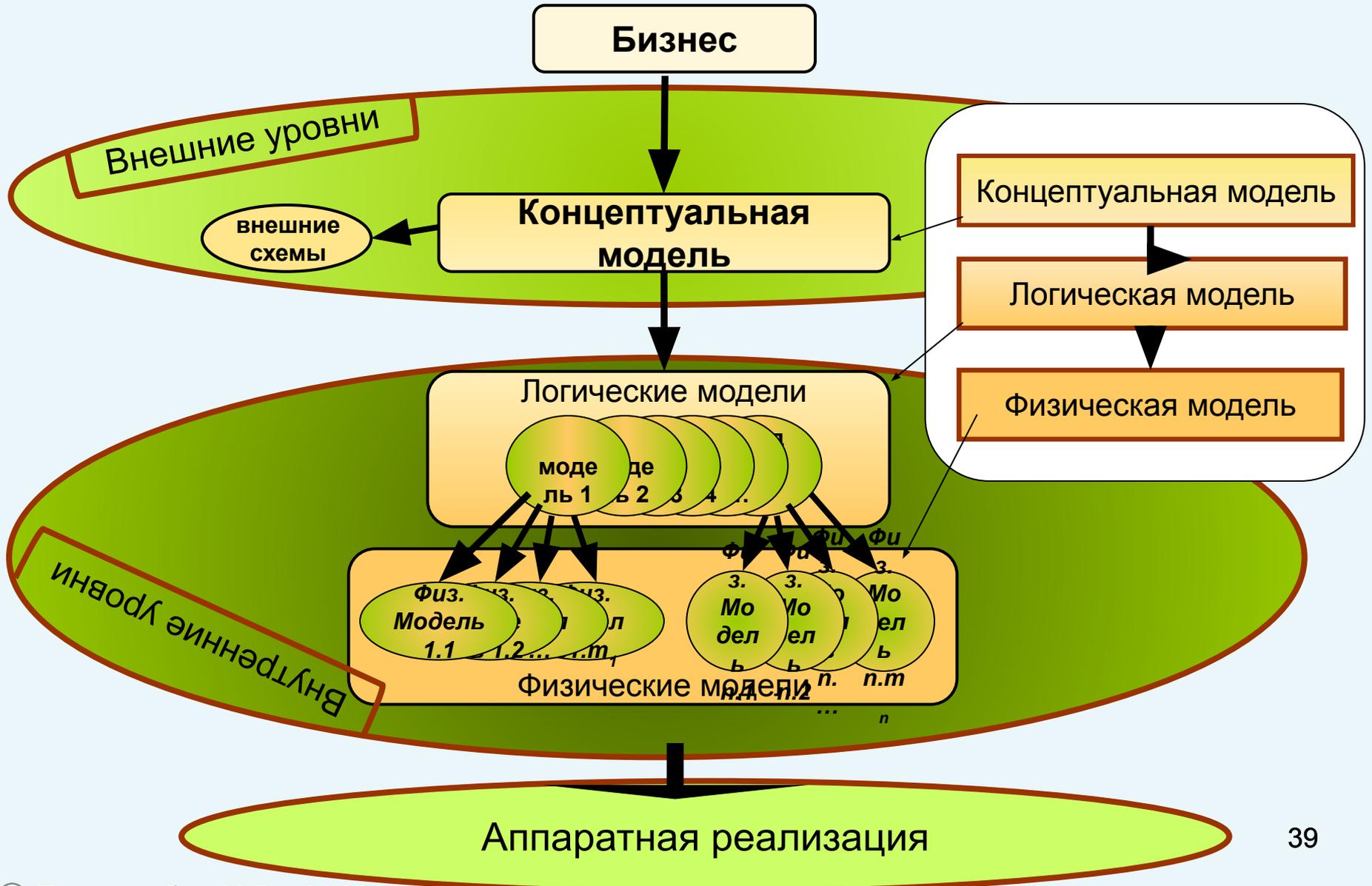
Замечание: справиться со сложностью системы можно используя декомпозиции и расслоения моделей. На следующем слайде представлена трёхуровневая модель ANSI/ISO.

# Трёхуровневая модель ANSI



- Отображение внешней модели в концептуальную
- Отображение концептуальной модели в логическую
- Отображение логической модели в физическую

# Трёхуровневая модель (в экземплярах)

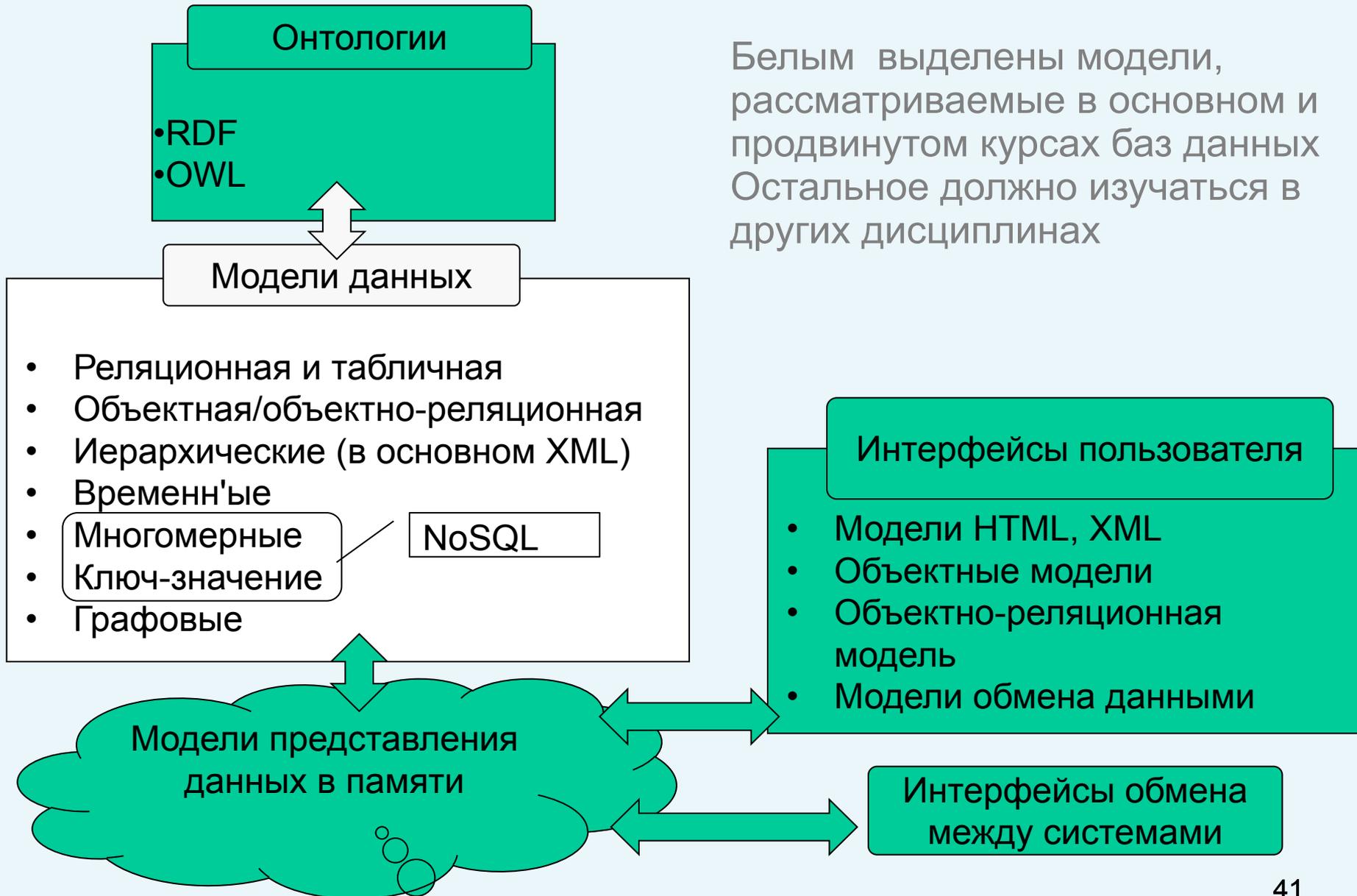


# Какие модели описывают бизнес. Схема Закмана

		Данные ЧТО	Функции КАК	Дислока- ция, сеть ГДЕ	Люди КТО	Время КОГДА	Мотивация ПОЧЕМУ	
Бизнес-руководители	Планировщик	Список важных понятий и объектов	Список основных бизнес-процессов	Территориальное расположение	Ключевые организации	Важнейшие события	Бизнес-цели и стратегии	Сфера действия (контекст)
	Владелец, менеджер	Концептуальная модель данных	Модель бизнес-процессов	Схема логистики	Модель потока работ (workflow)	Мастер-план реализации	Бизнес-план	Модель предприятия
ИТ-менеджеры и разработчики	Конструктор, архитектор	Логические модели данных	Архитектура приложений	Модель распределенной архитектуры	Архитектура интерфейса пользователя	Структура процессов	Роли и модели бизнес-правил	Модель системы
	Проектировщик	Физическая модель данных	Системный проект	Технологич. архитектура	Архитектура презентации	Структуры управления	Описания бизнес-правил	Технологическая (физическая) модель
	Разработчик	Описание структуры данных	Программный код	Сетевая архитектура	Архитектура безопасности	Определение временных привязок	Реализация бизнес-логики	Детали реализации
		Данные	Работающие программы	Сеть	Реальные люди, организации	Бизнес-события	Работающие бизнес-стратегии	Работающее предприятие
		Данные	Функции, Процессы	Расположение систем	Люди, организации	Время, расписание	Мотивация	

Это наши модели

# Модели и метамодели



Белым выделены модели, рассматриваемые в основном и продвинутом курсах баз данных. Остальное должно изучаться в других дисциплинах.

# Сколько языков программирования нужно знать для работы с ИС?

Пример Oracle. Сейчас и 20 лет назад.

1995 год

1. SQL
2. PL/SQL или TSQL
3. SQL\*Plus
4. C/C++
5. HTML
6. JavaScript
7. PHP

Сейчас

1. SQL
2. PL/SQL или TSQL
3. SQL\*Plus
4. C/C++
5. HTML
6. JavaScript + пакеты
7. PHP

8. Язык Java + технологии
9. XML (XSL, XPath, Schema, RELAX NG, XQuery, XMI)
10. Ruby (+ Rails) ??

11. Языки для представления семантики (RDF, OWL, ...)
12. Языки, специфичные для предметной области (DSL, MDA) ??

# Аппаратная реализация

Виды памяти используемой базами данных:

- **Первичная** (оперативная) память – емкость до единиц гигабайт. Время обращения десятки или сотни наносекунд ( $10^{-8} - 10^{-7}$  с). НЕ СОХРАНЯЕТ ИНФОРМАЦИЮ ПРИ ПЕРЕРЫВАХ В ПИТАНИИ!!
- **Вторичная** (как правило, жесткий магнитный диск) – емкость от сотен гигабайт до единиц терабайт. Время обращения сотые доли секунды ( $10^{-2}$  с).

Ожидаемая замена на флеш-память изобретённую в 1984 году Фудзио Масуокой в компании Тошиба. В наст. время ресурс записи 10-100 тыс. раз. Скорость чтения десятки микросекунд и менее.

Скорость передачи данных 300 - 500 Мбайт/с. Скорость стирания от единиц до сотен мс.

- **Третичная** (массивы дисков магнитных или оптических, другие оптические носители) – емкость практически не ограничена. Время обращения секунды, десятки секунд или минуты. (см. Дартмут. Ун-т)

Свойства современных запоминающих устройств во многом определяют структуру и функции СУБД.

В соответствии с традицией термин  
“память” означает “оперативная  
память”

# Проблема быстродействия

При работе с немедленно сохраняемыми данными **запросы** к базе будут выполняться недопустимо медленно.

Выход из положения: В первичной памяти создается кэш буферов достаточно большой емкости. Если кроме информации используемой в данный момент удастся извлечь информацию, которая понадобится в ближайшем будущем, и сначала искать информацию в кэше, то число обращений к диску резко сократится.

Показатель

$\text{Hit\_ratio} = (\text{число\_обращений\_в\_кэш}) / (\text{число\_обращений\_к\_данным})$

должен быть как можно ближе к 1, например  $\gg 0,95$ .

Примечание: В современных СУБД применяется сложная система буферов.

# Заключение

Что Вы должны освоить прослушав эту лекцию:

- Общие представления о синтактике, семантике и прагматике
- Понятие “запись”. Схемы записи. Наборы записей.
- Классификация типов данных
- Понятие “домен”
- Понятия “модель данных”, “структурированные”, “полуструктурированные” и “неструктурированные” данные
- Представление о базе данных. Понятие “СУБД”
- Схема базы и набора записей. Связи наборов записей
- Ограничения целостности, декларативные и процедурные ограничения
- Понятия “данные”, “метаданные” и “модели данных”
- База как модель бизнеса
- Модель ANSI

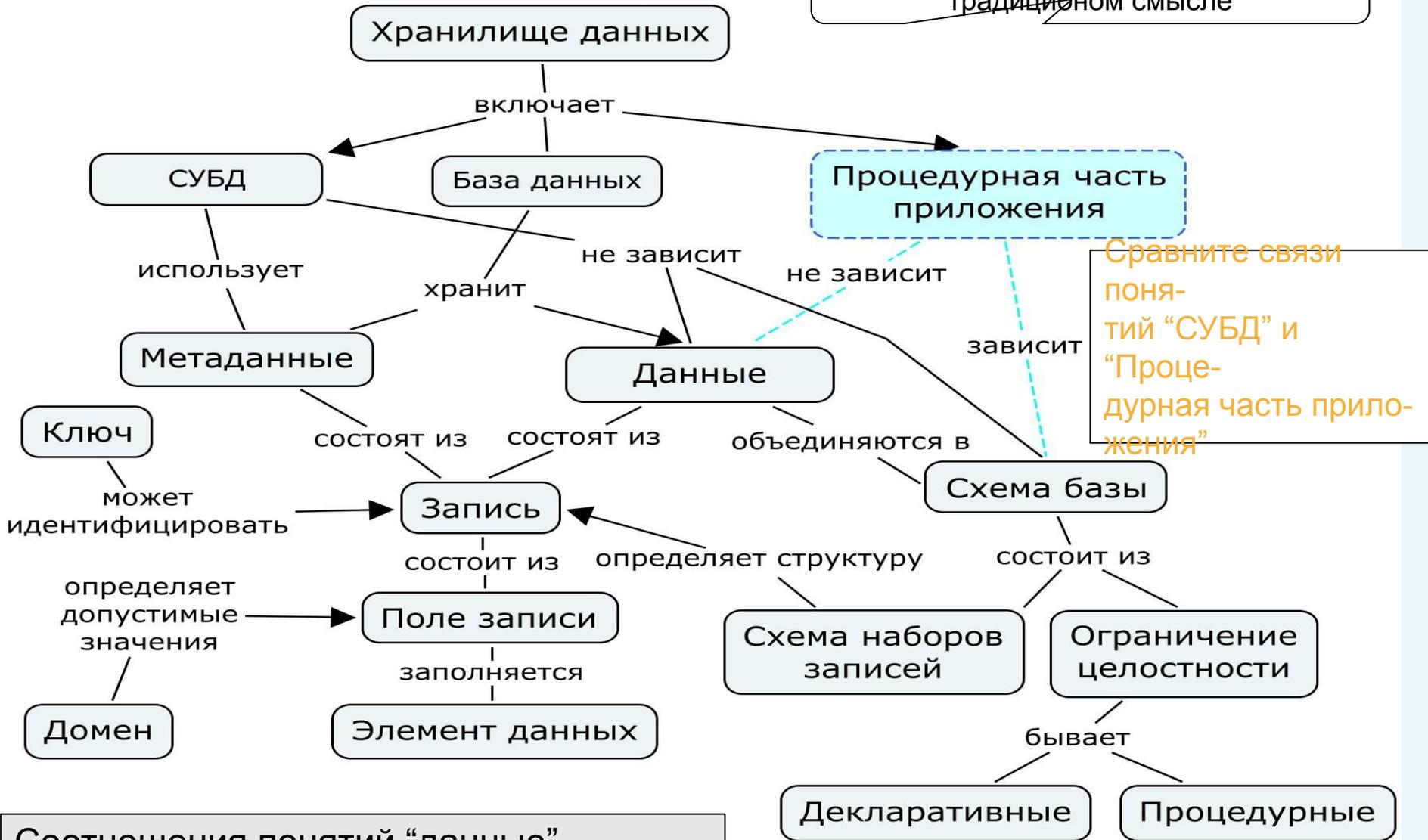
В дальнейшем весь этот материал будет рассмотрен подробнее.

# Литература

1. Когаловский М. Р. Энциклопедия технологий баз данных – М.: Финансы и статистика, 2002.-800 с.

# Основные понятия баз данных

Термин “хранилище” употреблён не в традиционном смысле



Сравните связи понятий “СУБД” и “Процедурная часть приложения”

Соотношения понятий “данные”, “семантика” и “прагматика” см. слайд 10

# Словарь студента (1/4)

- ❑ **База данных** - собрание записей, обладающее следующими свойствами:
  - записи интегрированы в некоторые структуры;
  - база как правило содержит **метаданные**;
  - база обладает **персистентностью** (способностью к сохранению);
  - данные независимы от обрабатывающих их программ (для баз не использующих объекты).
- ❑ **Данные** – это представление фактов о предметной области информационной системы в виде структуры, состоящей из знаков.
- ❑ **Семантики** это смыслы данных
- ❑ **Прагматики**, отражают отношение интерпретатора к данным. Прагматика задаёт выбор одной из возможных семантик.
- ❑ **Домен** – подмножество значений некоторого типа данных, имеющих определенный смысл.
- ❑ **Записью** в базах данных называют минимальную уникально идентифицируемую единицу независимого хранения данных, образованную иерархией **полей**.

# Словарь студента (2/4)

- ❑ **Ключ записи** – элемент или множество элементов данных (полей записей), значения которых однозначно идентифицируют один или несколько экземпляров записей этого типа в базе данных.
- ❑ **Метаданные** - это данные специального вида, которые описывают структурные свойства данных, хранимых в базе.
- ❑ **Модель данных** - это “система типов данных, типов связей между ними и допустимых видов ограничений целостности, которые могут быть для них определены” (М.Р. Когаловский). Составные части модели данных:
  - **структурная часть** - то есть типы, отношения и связи между ними;
  - **целостная часть** – ограничения целостности;
  - **манипуляционная часть** – языки для манипулирования данными и запросов к базе.
- ❑ **Неопределённое значение (NULL)** – означает отсутствие заданного значения, но не пустое значение.
- ❑ **Ограничения целостности** - это условия специального вида, которые должны выполняться для всей схемы или некоторой подсхемы базы данных. Выделяют:
  - **декларативные** ограничения целостности;
  - **процедурные** ограничения целостности.

# Словарь студента (3/4)

- ❑ **Поле записи** – именованный элемент данных, являющийся частью структуры записи базы данных или файла.
- ❑ **СУБД** - это “программная система, предназначенная для создания и хранения базы данных на основе некоторой модели данных, обеспечения логической и физической целостности содержащихся в ней данных, надежного и эффективного использования **ресурсов** (данных, пространства памяти и вычислительных ресурсов), предоставления к ней санкционированного доступа для приложений и конечных пользователей, а также для поддержки функций администратора базы данных” (М.Р. Когаловский).
- ❑ **Схема (тип) записи** - это описание внутренней структуры записи. Схема записи определяет связную последовательность **полей**, образующую дерево.
- ❑ **Схема базы** – описание базы данных.
- ❑ **Тип данных** – именованное потенциальное множество значений данных заданной структуры. Выделяем три группы типов данных:
  - простой;
  - структурированный;
  - ссылочный

# Словарь студента (4/4)

- **Хранилище данных** это все, что хранит данные. Хранилище определяется:
  - тем, что в нем хранится;
  - тем, как оно хранится;
  - какие семантики и прагматики используются;
  - тем, что и как об этом спрашивают (или могут спросить);
  - как интерпретируются результаты выборки;
  - тем, кто, как и когда может спрашивать.

Замечание: В этой лекции мы используем нетрадиционное представление о хранилище. В конце курса будет дано традиционное понимание хранилища как источника данных для решения задач анализа данных, в том числе поиска закономерностей, и принятия решений.

- **Элементами данных** называются значения полей.

# Задачи для решения в аудитории

1. В таблице emp определите количество сотрудников в отделе №30

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17.12.80	800	-	20
7499	ALLEN	SALESMAN	7698	20.02.81	1600	300	30
7521	WARD	SALESMAN	7698	22.02.81	1250	500	30
7566	JONES	MANAGER	7839	02.04.81	2975	-	20
7654	MARTIN	SALESMAN	7698	28.09.81	1250	1400	30
7698	BLAKE	MANAGER	7839	01.05.81	2850	-	30
7782	CLARK	MANAGER	7839	09.06.81	2450	-	10

2. Какие данные хранятся в emp? Дайте текст комментария одной фразой.
3. Определите концепт “посетитель сайта”. Задайте его атрибуты.