Базы данных
Введение

Бессарабов Н.В.

bes@fpm.kubsu.ru

2015 г.

Особенности курса

- Единый подход к любым базам данных (БД). Они рассматриваются с одной стороны как модели бизнеса, с другой как компоненты информационных систем, с третьей стороны, учитываются особенности реализации систем управления базами данных (СУБД).
- Как обычно, рассматривается синтактика. Много внимания уделяется семантике и прагматике БД, которыми в подобных курсах обычно почти не занимаются.
- В современных базах данных используется большое количество моделей данных. Разобраться в этом зоопарке можно рассматривая морфизмы между ними и морфизмы между моделями данных и моделями бизнеса. По духу это подход теории категорий, но соответствующие формализмы в курсе не рассматриваются.
- Предполагается довольно высокий уровень абстрактного мышения.
- Помните, что без достаточно большой практической работы ничто не понимается до конца и не запоминается надолго.
- Инструментарий: используются СУБД Caché и Oracle, CASE-средство ERWin и модель WinRDBI. Большое внимание, уделяемое Caché вызвано тем, что в ней реализуются морфизмы трёх моделей данных.

Что полезно изучить в университете и освоить самостоятельно

- Любые курсы по Oracle, особенно по SQL, PL/SQL, аналитическим функциям, основам администрирования, Apex
- Язык и технологии Java, HTML5, CSS3, JavaScript, PHP, Python
- Технологии программирования
- Семантика в базах данных
- Облачные вычисления, базы данных в облаках

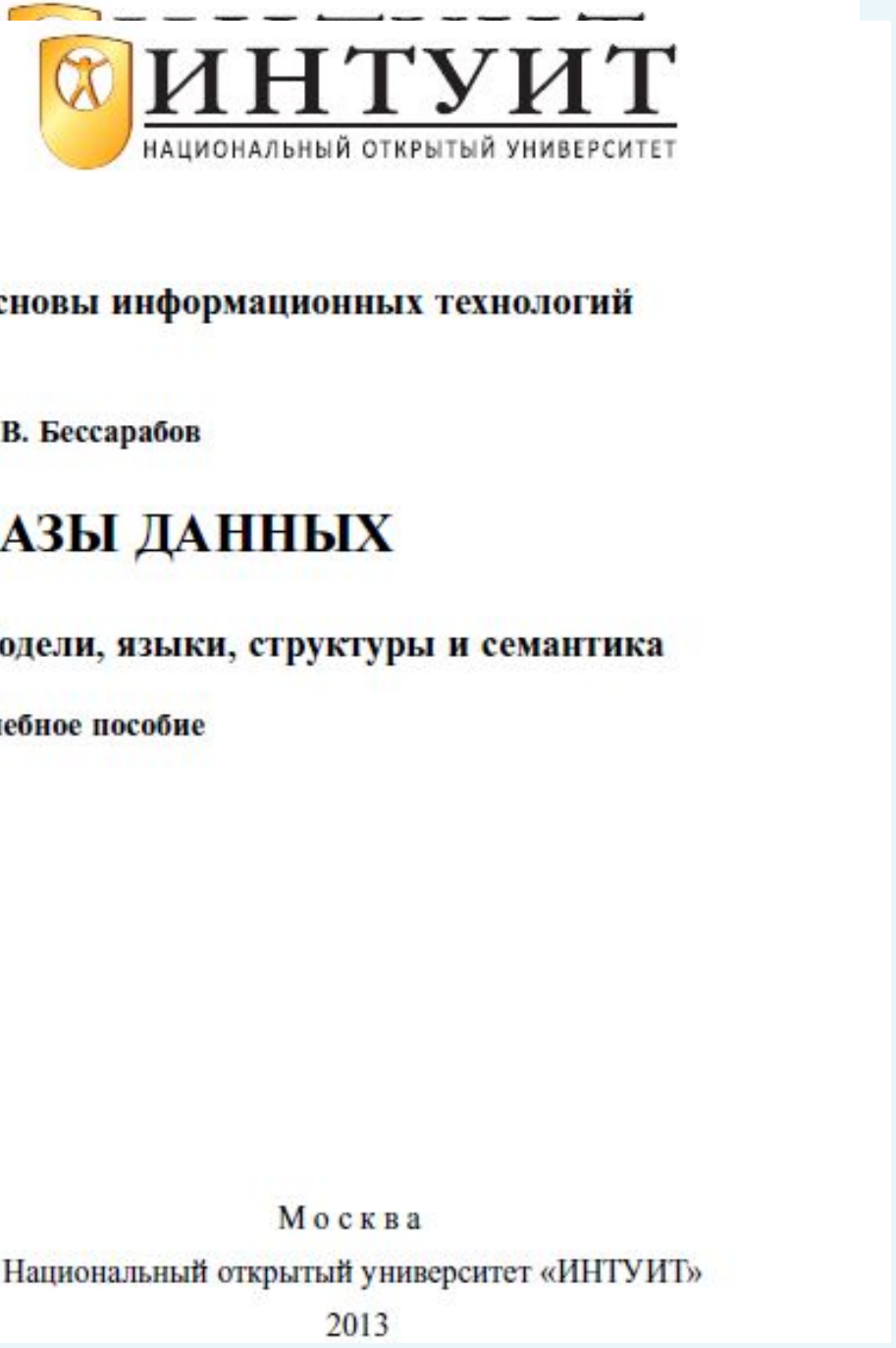
Настоятельно рекомендую:

- Принять участие в разработке серьёзного проекта, в том числе по грантам. Примеры тем: “Базы знаний встроенные в БД”, “Работа с БД на фрагменте естественного языка”, “База знаний по языкам SQL и PL/SQL”
- Изучать современную математику (разделы “Алгебры”, “Теория категорий”, “Логики”) и теорию систем (лучше по Урманцеву)
- Изучать английский язык
- Готовиться к сдаче экзамена на сертификат Oracle (SQL, PL/SQL, MySQL, Java)

Обеспеченность литературой (100 экз.) + сайт ИНТУИТ

Учебное пособие: www.intuit.ru (HTML, .fb2).

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



Цели лекции

В первой части лекции дано предварительное определение базы данных.

Вначале выясняется что такое “данные”, их “семантики” 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 только в бизнес-приложениях неопределённые значения.

В третьей части лекции БД рассматриваются как модели бизнеса. Этот подход очень важен и для студента, изучающего курс, и для постановщика задач создания информационных систем, содержащих базы данных. В некоторых вопросах (пример: так называемые, аномалии) невозможно разобраться до конца, не учитывая этот аспект.

Часть1.
Предварительное определение базы данных.
Семантика и прагматика

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 системы баз данных или информационной системы в форме, допускающей их хранение и обработку на компьютерах, передачу по каналам связи, а также восприятие человеком.

(М.Р. Когаловский)

Вопрос: Если имеются данные (вариант – полные, исчерпывающие данные) по какому-то объекту, процессу, то что ещё может понадобиться?

Ответ: Данные должны кем-то или чем-то интерпретироваться. Сами по себе они не могут быть использованы. Нужны семантики, то есть смыслы данных, и их прагматика, то есть отношение интерпретатора к данным. Можно понимать прагматику как выбор одной из возможных семантик.

Синтаксис, семантики и прагматика (1/2)

Пример 1 (анализ по косвенным признакам): На вступительных экзаменах в ВУЗ преподаватель, не получив ответа, просит студента хотя бы прочесть выражение вида $2xy^2+\ldots$ и получает совсем неожиданную интерпретацию “два ху и два наверху”.

Вопрос: разве не понятно, насколько абитуриент освоил школьную математику?

Пример 2: Табло на здании банка – запись с тремя полями

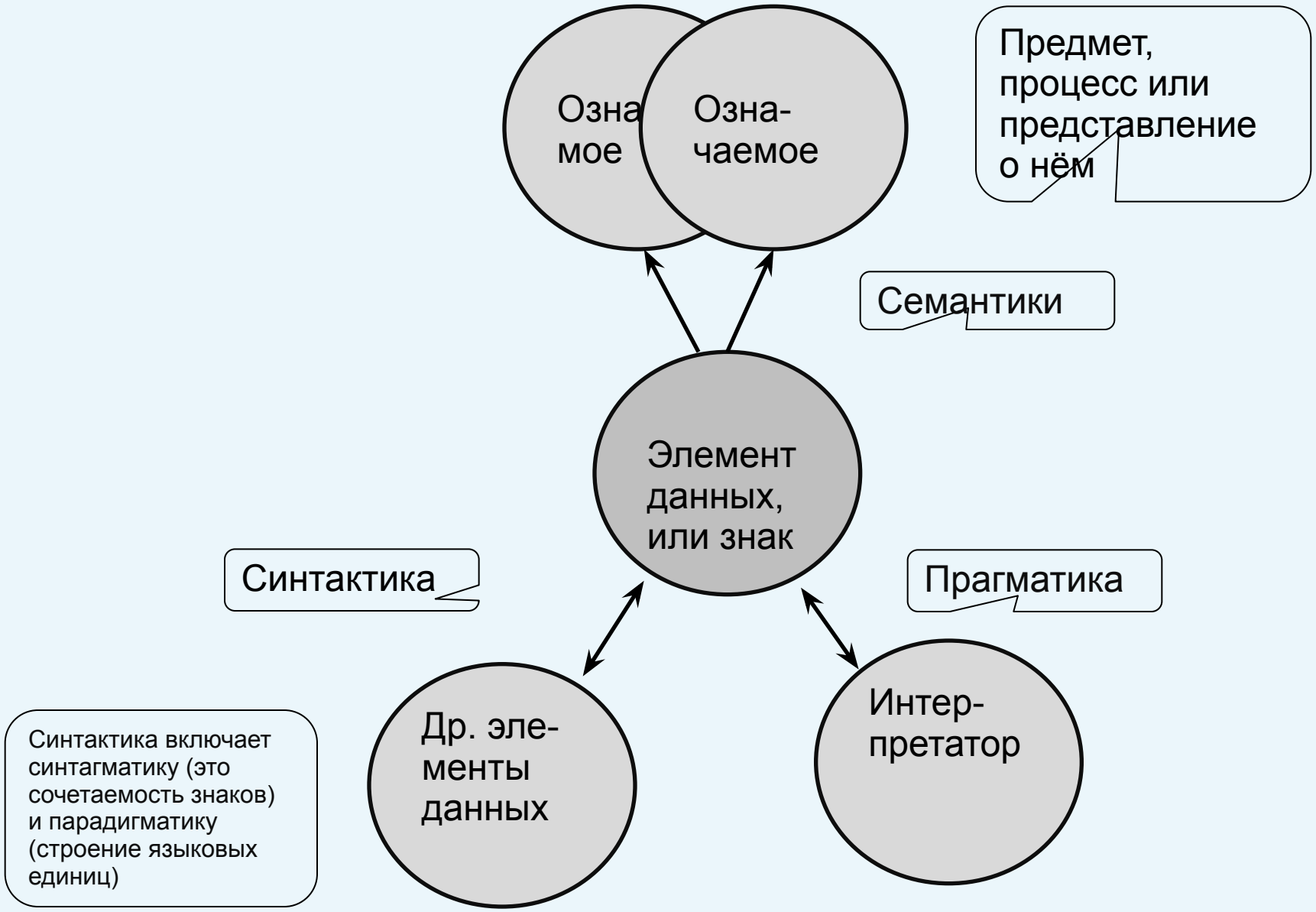
USD	**62.00**	**65.00**

Интерпретатор человек. Некоторые варианты прочтения:

1. Человек не знает, что такое валюта и как её обменивать. Ответ: “?”
2. Знает, но интересуется только покупкой (это 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 третьи семантика и прагматика.

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

Синтаксис, семантики и прагматика (2/2)



Как описать базу данных

База данных определяется:

- тем, что в ней хранится;
- тем, как оно хранится;
- 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: семантику данных примера пока не рассматриваем; то есть, считаем, что смысл данных определён однозначно и не нуждается в уточнении.

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.**
- **Схема записи** -- это описание внутренней структуры записи. Схема записи определяет связную последовательность полей, образующую дерево.
- **Поле записи** – именованный элемент данных, являющийся частью структуры записи базы данных или файла.
- Обычно, но не всегда, поля типизированы.
- Значения полей называются **элементами данных.**
- Требование уникальной идентификации записей может реализоваться выделением одного или нескольких полей в качестве **ключей.**

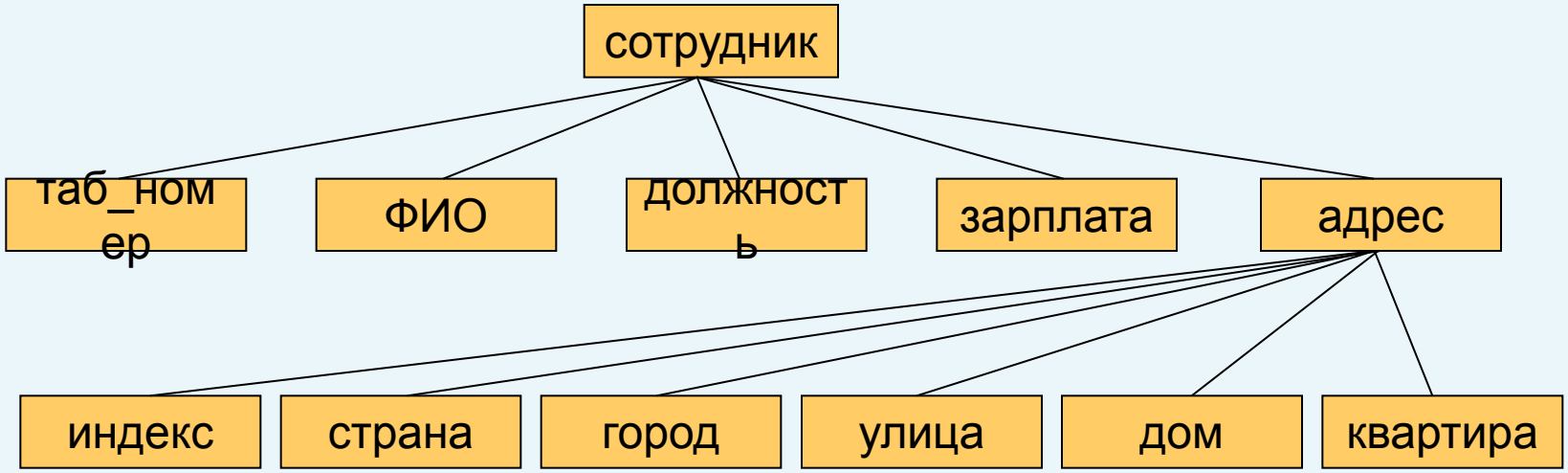
Замечание: В базах могут храниться данные, которые не удобно представлять записями. Это картографические, мультимедийные и другие плохо структурированные данные.

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 распространенные структурированные типы: **массивы** и **записи.**

Пример: Структурированный тип данных запись «адрес», рассмотренный ранее на слайде 7

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.

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 *вычислимого* типа данных “возраст (человека)” характеризуется условием (возраст>0 и возраст<120). На нем 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

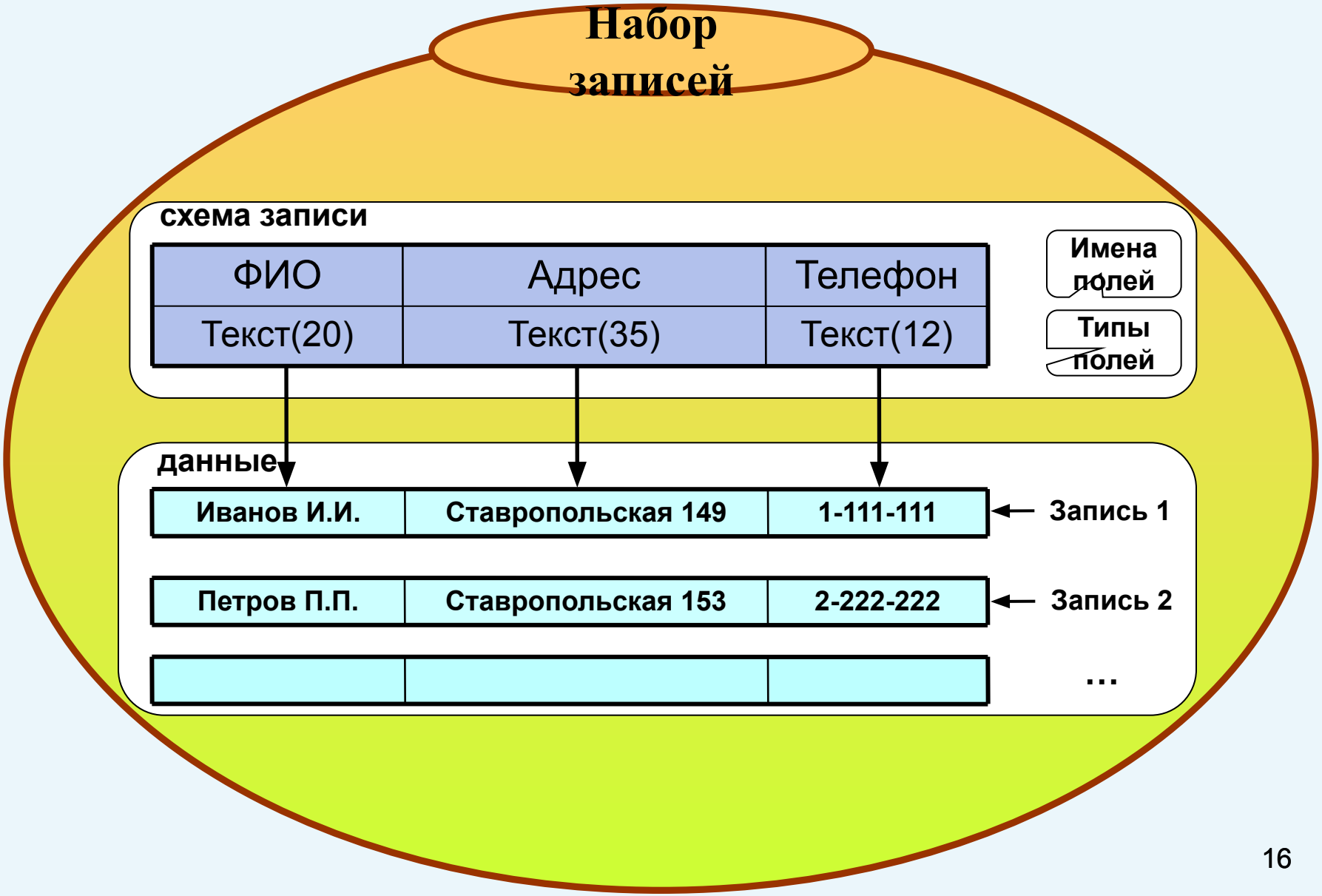


Схема базы

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. Этим мы займёмся позже.

Данные и метаданные

Метаданные это данные специального вида, которые описывают структурные свойства данных, хранимых в базе. Поскольку какие-то метаданные имеются всегда, база обладает свойством **самодокументируемости**.

Понятно, что схема базы это некоторая существенная часть метаданных базы.

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

Пример. Данные и метаданные

Данные содержатся в двух таблицах. Таблица Т1, приведена ниже, таблица Т2 на этом слайде не показана.

Т1:

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. М1 содержит перечень таблиц, М2 – перечень столбцов

М1:

Ном_таб	Имя_таб
1	Т1
2	Т2

М2:

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

Что такое база данных (узкое и неполное псевдоопределение)

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

- Записи интегрированы в некоторые структуры (таблицы со связями, деревья, сети и т.д.), описываемые схемами;
- База обладает **персистентностью** - способностью хранить данные.
- База как правило содержит **метаданные;**
- Данные независимы от обрабатывающих их программ.

Замечание: Под независимостью в простейшем случае понимается возможность создать структуры данных, не обращая внимание на их обработку, а затем написать программные модули, обрабатывающие данные.

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

Ограничения целостности

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

Декларативные ограничения описываются заданием некоторого свойства при создании схемы базы. Например, ограничение “первичный ключ” (“primary key”) означает, что значения указанных в определении ключа полей записи определяют ее однозначно.

Процедурные ограничения могут быть определены только через процедуры специального вида, называемые триггерами.

Замечание: Ограничения целостности – это ещё один элемент семантики.

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.

2. Ограничение типа “Check” (Проверка)
В наборе записей
Сотрудник (ИНН, ФИО, Должность, Зарплата, Бонус)
для каждой записи должно выполняться условие
Бонус<0.2*Зарплата
Замечание: Ограничения Check строятся на данных одной 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**.

Теперь понятно, что к определению базы, данному в предыдущем слайде следует добавить еще свойство:

- База данных создается в рамках одной или нескольких моделей данных

Составные части модели данных

По Дейту [3], реляционная модель состоит из трех частей:

- структурной,
- целостной,
- манипуляционной.

Заметьте, что семантика и прагматика в этом раскладе отсутствуют.

Перечисленные аспекты могут выделяться в любой модели данных, но не всегда реализуются явно. Поэтому в определение базы данных следует добавить:

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

Структурная часть реляционной модели образуется отношениями и связями между ними. Единственная структура данных в реляционной модели это рассмотренные выше n-арные отношения, связанные бедным набором связей типов 1:1 и 1:n.

Подробно реляционная модель и связи между наборами записей (отношениями) будут рассмотрены в следующих лекциях.

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 надстройка над файловой системой, обеспечивающая работу со сложными структурами данных без явного использования операций с файлами.

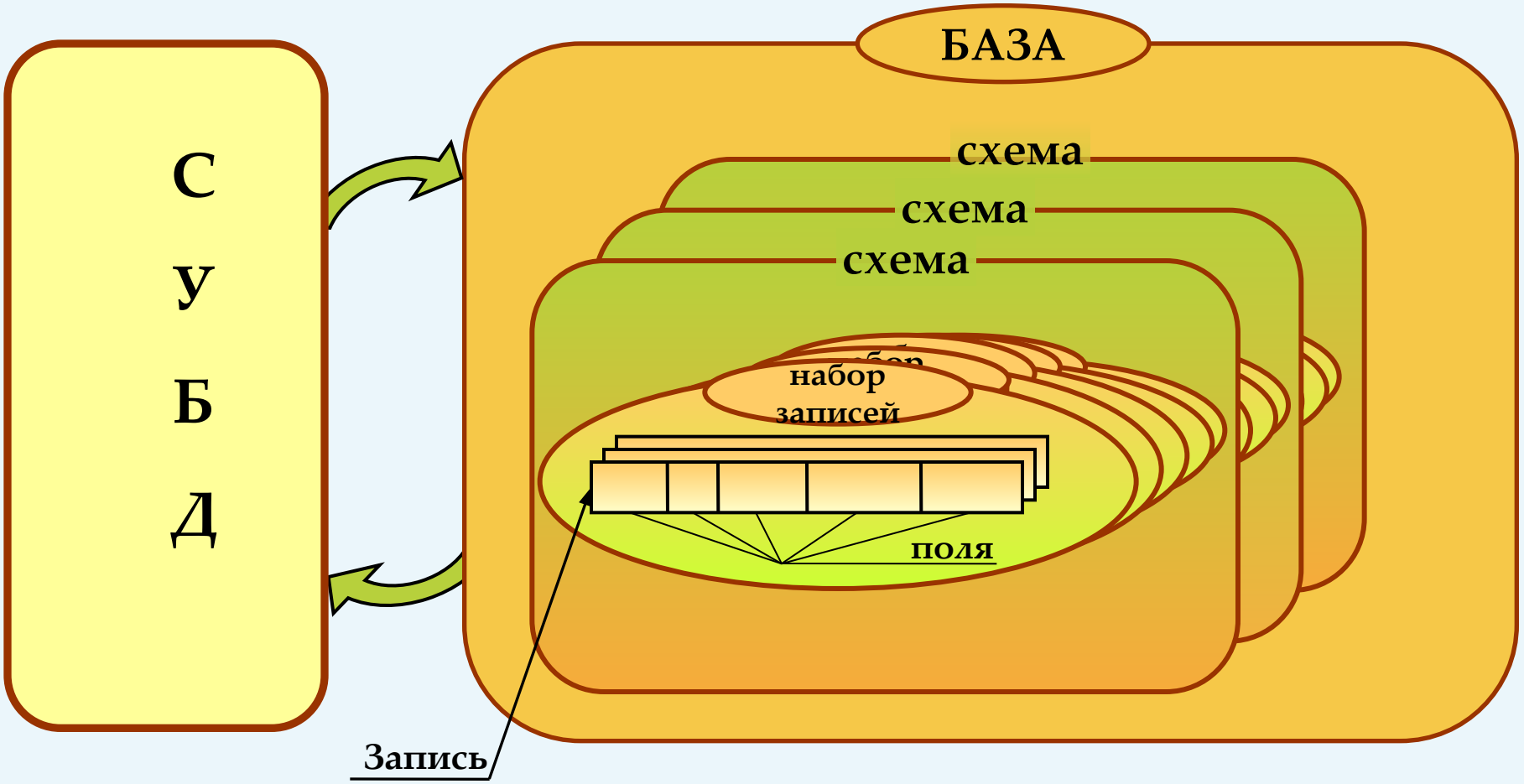
Системы управления базами данных

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

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: В 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

Модельный подход очень важен и для студента, изучающего курсы баз данных и баз знаний, и для постановщика задач создания информационных систем. В некоторых вопросах (пример -- так называемые, аномалии) невозможно разобраться до конца, не учитывая модельный аспект.

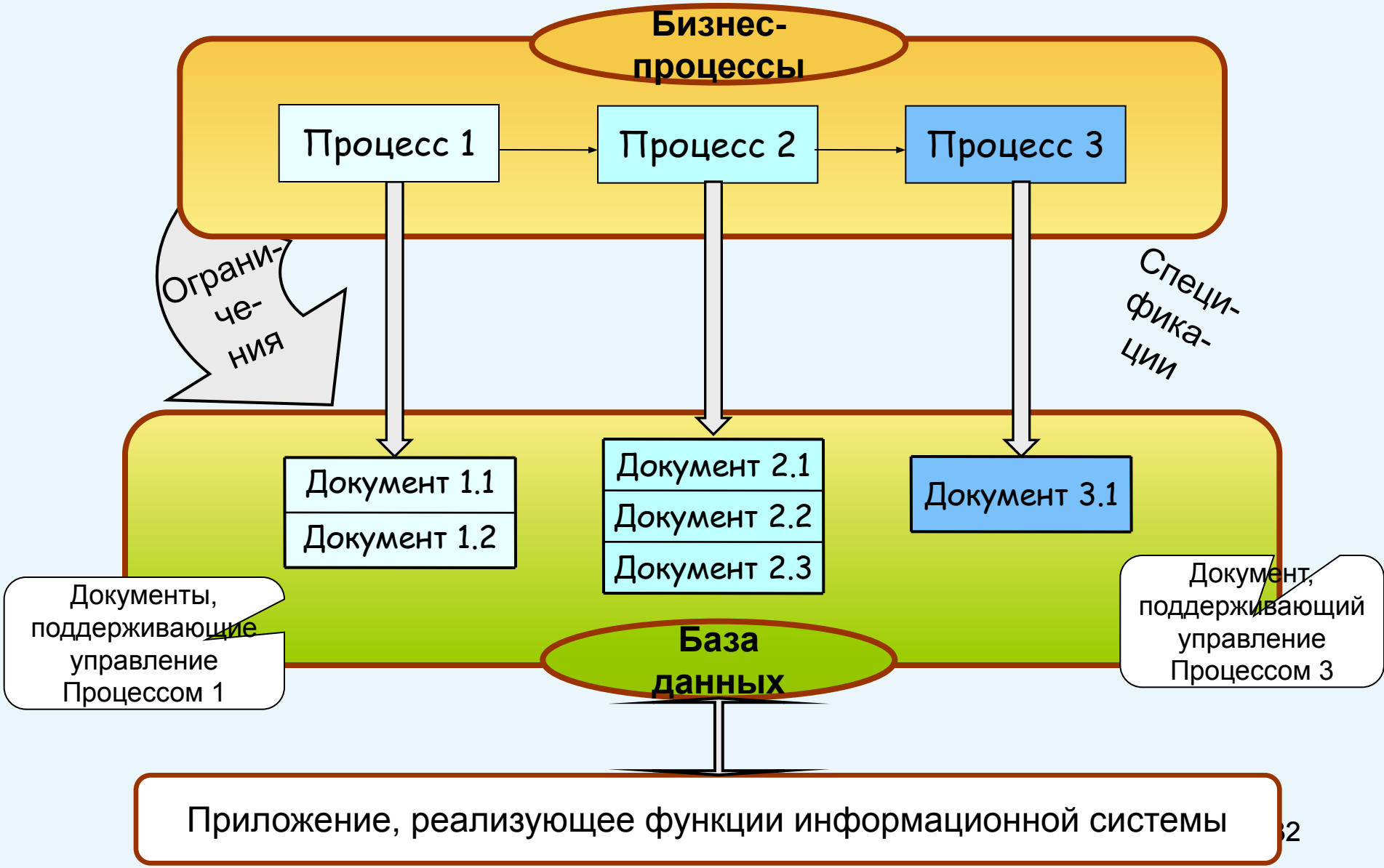
Под бизнесом в дальнейшем изложении будем понимать любую деятельность, не обязательно связанную с извлечением прибыли.

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

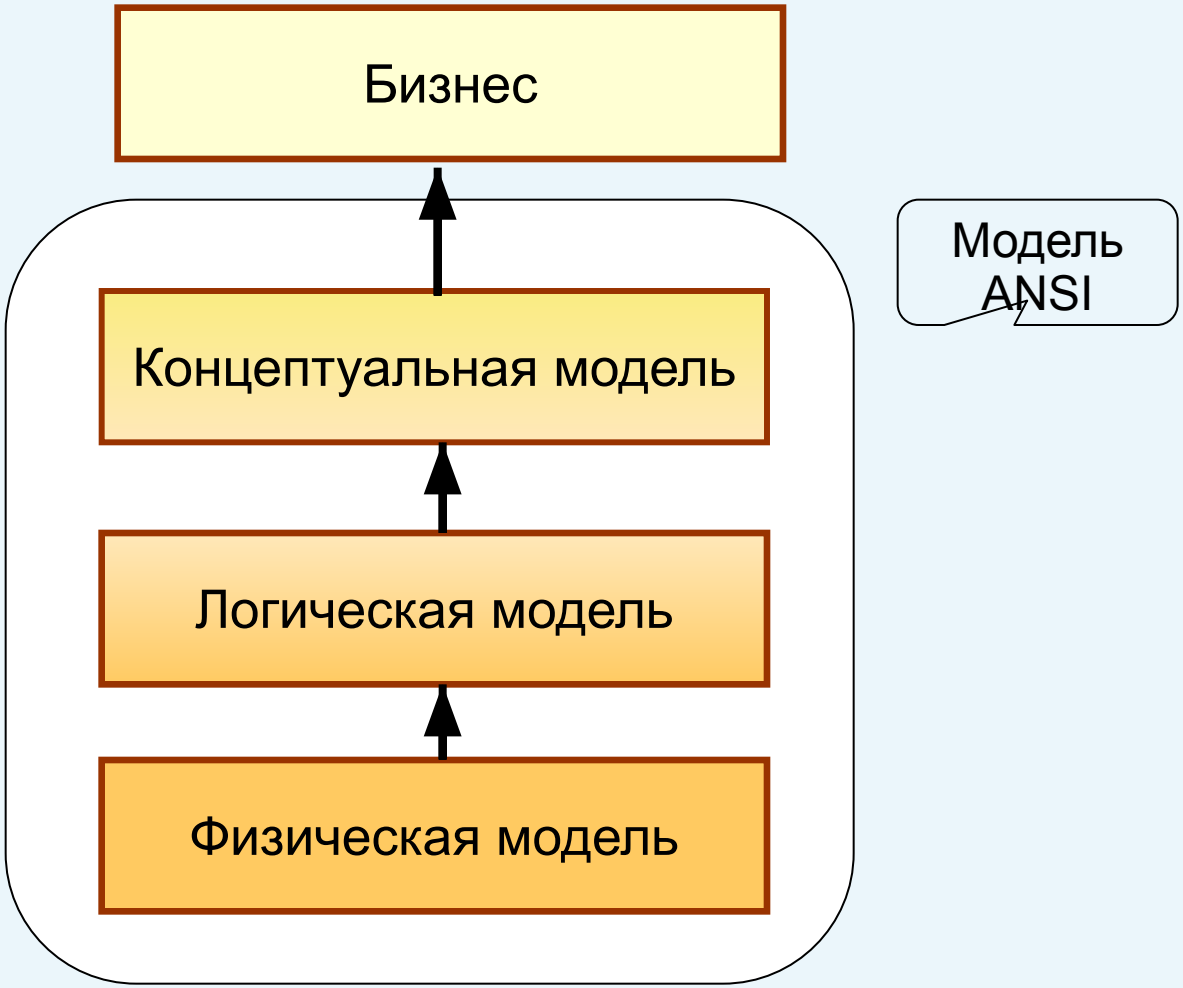
Существуют системы, для управления которыми достаточно управлять потоками документов, регламентирующих и сопровождающих бизнес-процессы. Такая документо - ориентированная система представлена на следующем слайде.

База данных как модель бизнеса

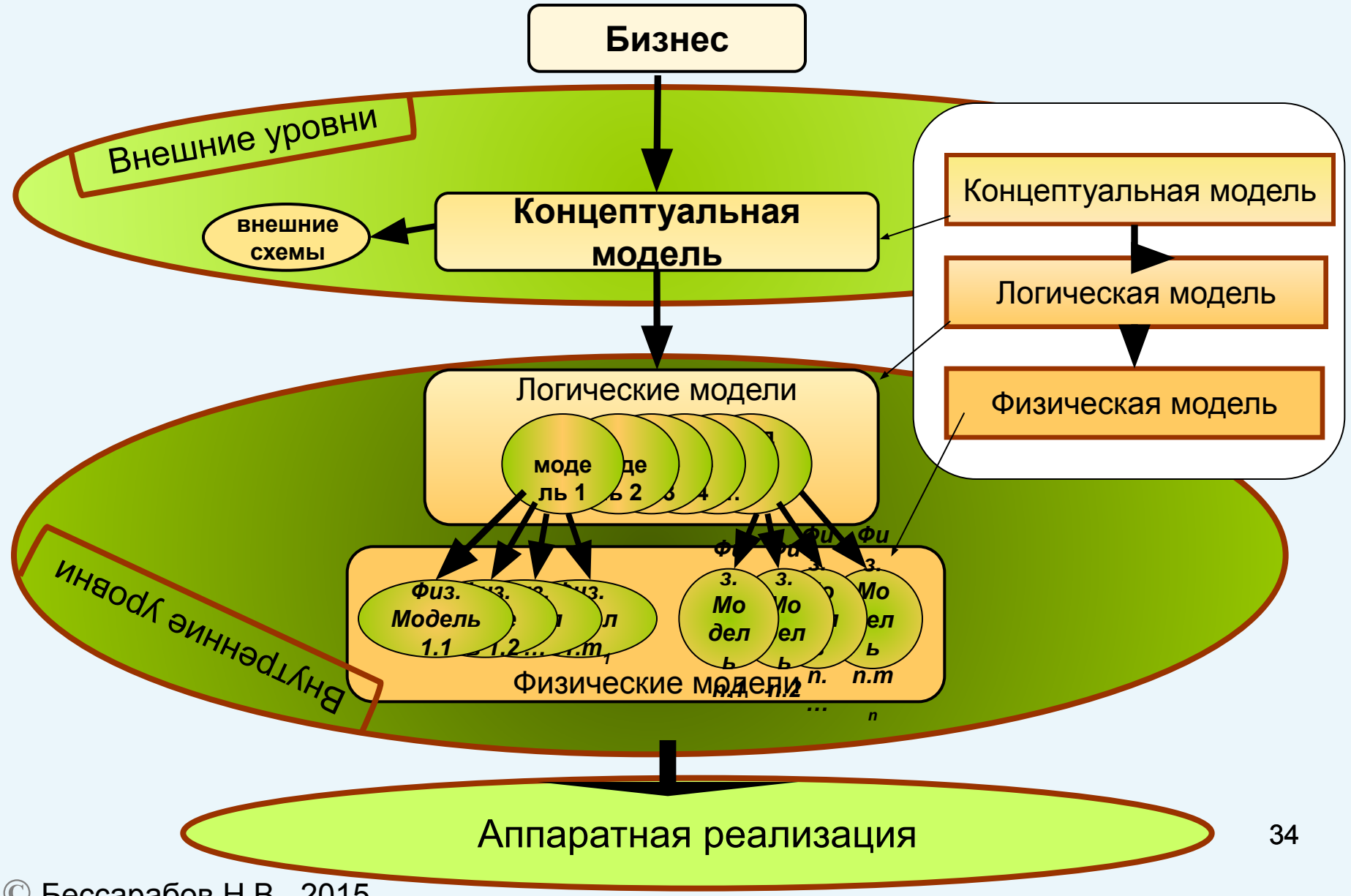
(пример документарного подхода)



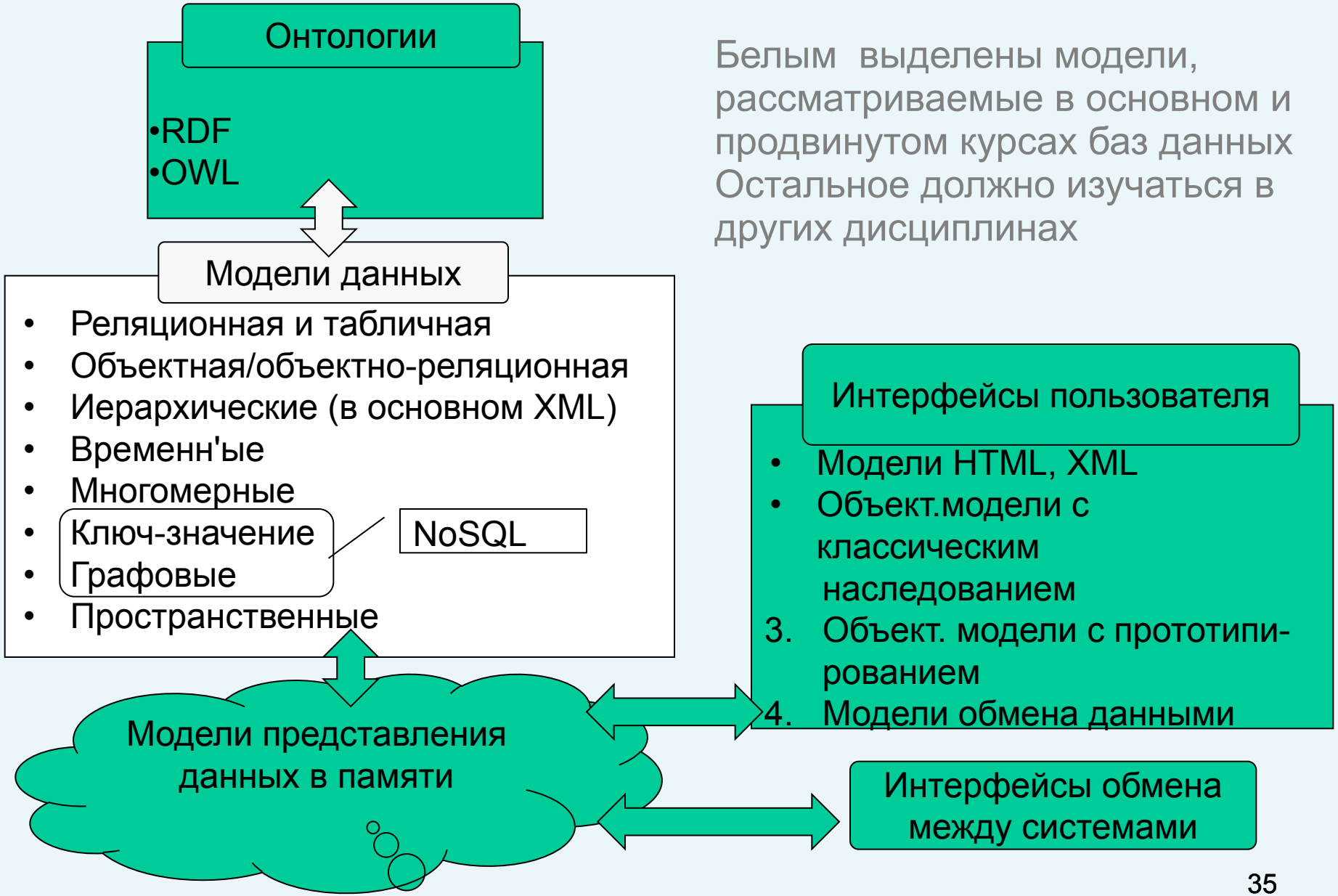
Трёхуровневая модель ANSI



Трёхуровневая модель (в экземплярах)



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
3. SQL*Plus
4. C/C++
5. HTML
6. JavaScript
7. PHP

Сейчас

1. SQL
2. PL/SQL
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)
- **Третичная** (массивы дисков магнитных или оптических, другие оптические носители) – емкость практически не ограничена. Время обращения секунды, десятки секунд или минуты.

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

В соответствии с традицией термин “память” означает “первичная 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 удается извлечь информацию, которая понадобится в ближайшем будущем, и сначала искать информацию в кэше, то число обращений к диску резко сократится. Показатель

Hit_ratio = (число_обращений_в_кэш)/(число_обращений_к_данным)

должен быть как можно ближе к 1, например >0,95.

Примечание: В современных СУБД применяется сложная система буферов.

Заключение

Что Вы должны освоить прослушав эту лекцию:

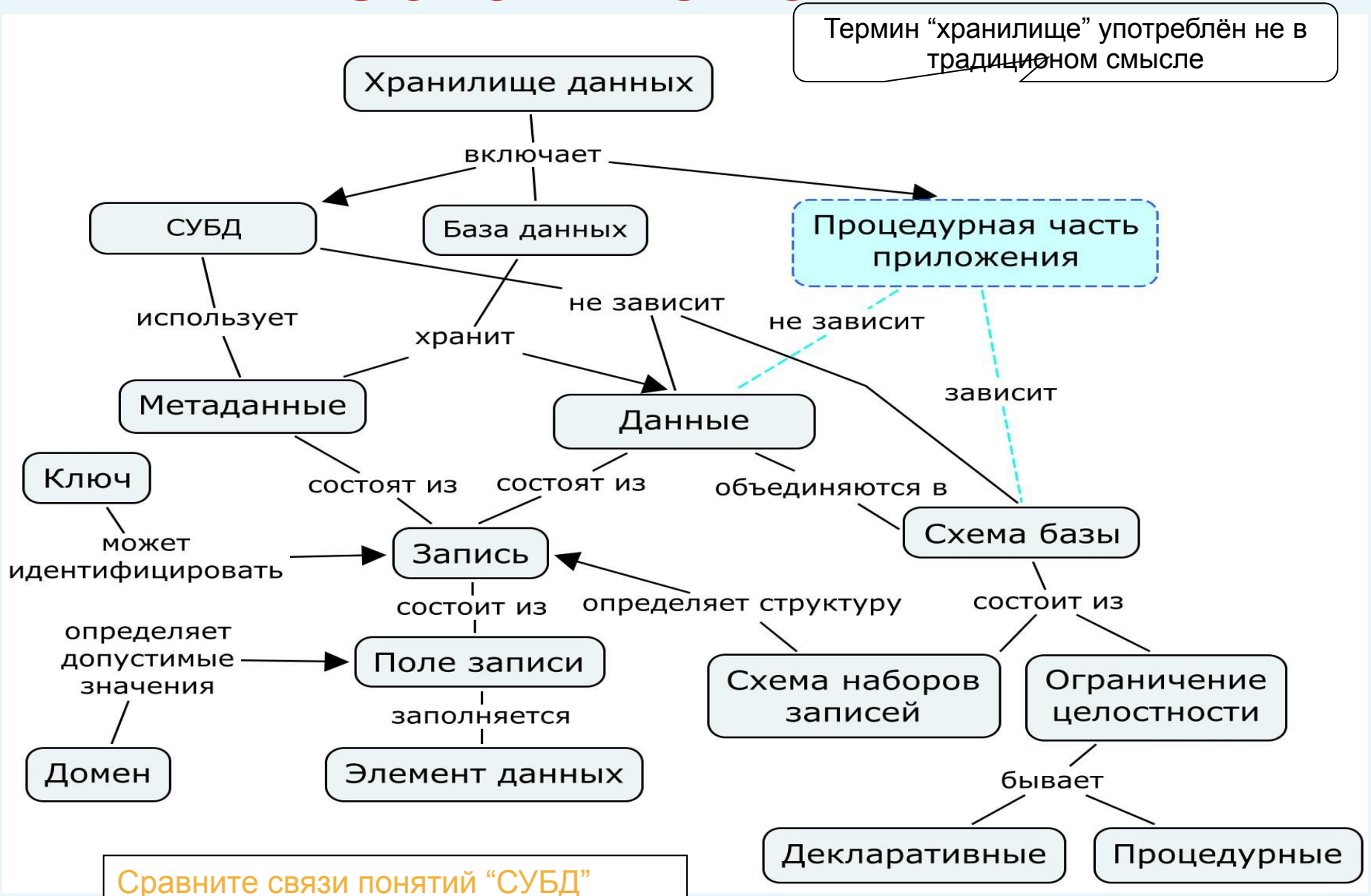
- Общие представления о синтактике, семантике и прагматике
- Понятие “запись”. Схемы записи. Наборы записей.
- Классификация типов данных
- Понятие “домен”
- Представление о базе данных
- Схема базы. Связи наборов записей
- Ограничения целостности, декларативные и процедурные ограничения
- Понятия “данные”,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

В дальнейшем весь материал лекции будет рассмотрен подробнее.

Литература

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

Основные понятия



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

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 системы баз данных или информационной системы в форме, допускающей их хранение и обработку на компьютерах, передачу по каналам связи, а также восприятие человеком (М.Р. Когаловский)
- **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. Бывают:
  - **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;
  - тем, что и как об этом спрашивают (или могут спросить);
  - 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.

Замечание: Термин “хранилище” употреблён не в традициоином смысле. Обычно хранилище это собрание данных, в основном предназначенное для анализа деятельности.