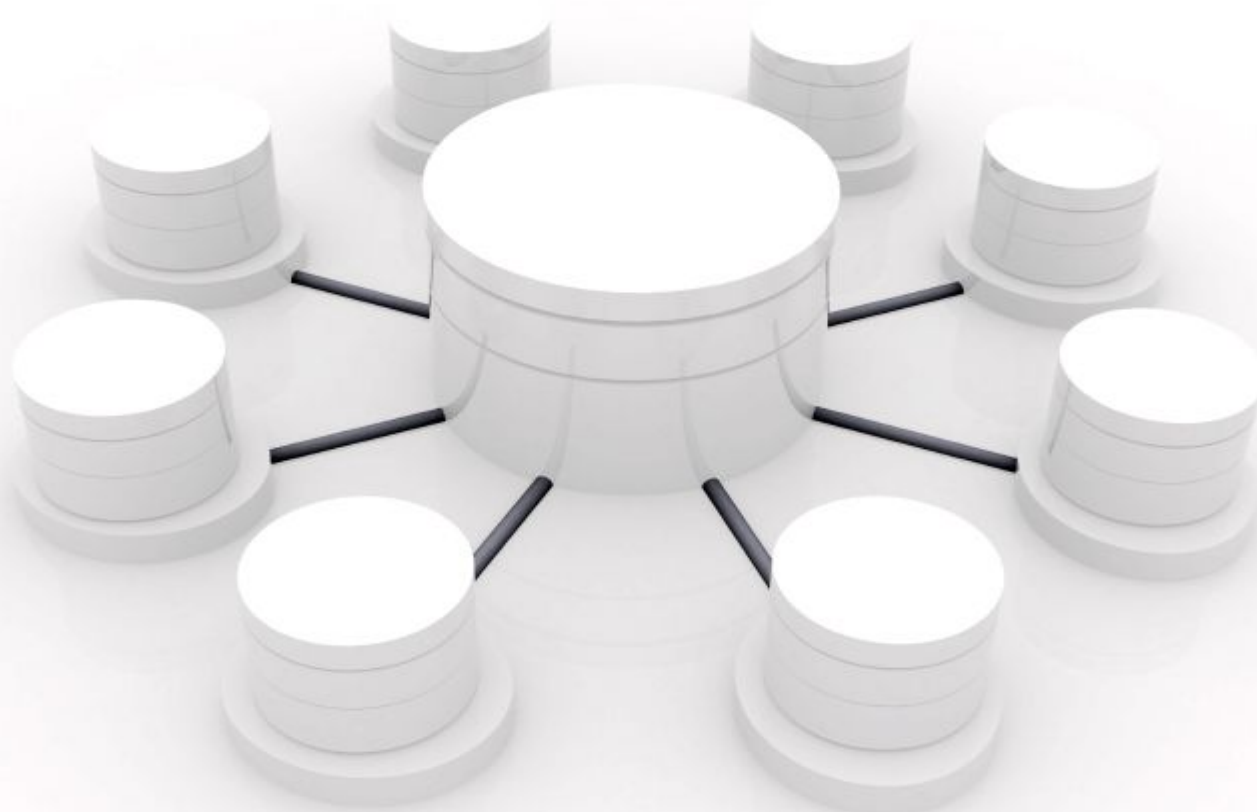


История развития и поколения СУБД.
Классификация СУБД. Архитектура СУБД. Основные
компоненты СУБД.



- История развития и поколения СУБД. Классификация СУБД. Архитектура СУБД. Схема ее работы и характеристика основных компонентов. Роль администраторов баз данных. Уровни представления информации в базах данных. Логическая и физическая независимость данных и средства ее обеспечения.

Этапы проектирования БД

- 1. **Системный анализ и словесное описание информационных объектов предметной области.**
- 2. **Информационно-логическое (инфологическое) проектирование** - создание инфологической модели предметной области – частично формализованного описания объектов предметной области в терминах некоторой семантической модели. Наиболее традиционная из них называется моделью сущности – связи (E/R- модель, entity-relationship), имеет графическую природу (прямоугольники, стрелки, ромбы).
- 3. **Выбор СУБД** и других инструментальных программных средств.
- 4. **Даталогическое (или логическое) проектирование**, т.е. описание БД в терминах принятой даталогической модели данных (наиболее распространена реляционная, т. е. E/R-модель преобразуем в реляционную).
- 5. **Физическое проектирование БД**, т.е. выбор эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы приложения.

Подходы к выбору состава и структуры предметной области

- 1. *Функциональный подход – он реализует принцип движения “ от задач ” и применяется тогда, когда заранее известны функции некоторой группы лиц и комплексов задач, для обслуживания информационных потребностей которых создается рассматриваемая БД. В этом случае мы можем четко выделить необходимый минимальный набор объектов предметной области, которые должны быть описаны.*
- 2. *Предметный подход – когда информационные потребности будущих пользователей БД жестко не фиксируются. Они могут быть многоаспектными и динамичными. БД, конструируемая при этом, называется предметной.*

Пример описания предметной области «Библиотека»

Пусть требуется разработать информационную систему **для автоматизации учета получения и выдачи книг в библиотеке**. Система должна предусматривать режимы ведения систематического каталога, отражающего перечень **областей знаний**, по которым имеются **книги** в библиотеке. Области знаний в систематическом каталоге могут иметь **уникальный внутренний номер** и **полное наименование**. Каждая книга может содержать сведения из нескольких областей знаний. Каждая книга в библиотеке может присутствовать в нескольких экземплярах. **Книга**, хранящаяся в библиотеке, характеризуется следующими параметрами:

- уникальный шифр (ISBN);
- название;
- фамилии авторов (могут отсутствовать);
- место издания (город);
- издательство;
- год издания;
- количество страниц;
- стоимость книги;
- количество экземпляров книги в библиотеке.

Книги могут иметь одинаковые названия, но они различаются по своему уникальному шифру (ISBN).

В библиотеке **ведется картотека читателей**.

На каждого читателя в картотеку заносятся следующие сведения:

- фамилия, имя, отчество;
- домашний адрес;
- телефон (будем считать, что у нас два телефона — рабочий и домашний);
- дата рождения.

Каждому читателю присваивается **уникальный номер читательского билета**.

Каждый читатель может одновременно держать на руках не более 5 книг. Читатель не должен одновременно держать более одного экземпляра книги одного названия.

Каждая книга в библиотеке может присутствовать в нескольких экземплярах. **Каждый экземпляр** имеет следующие характеристики:

- уникальный инвентарный номер;
- шифр книги, который совпадает с уникальным шифром из описания книг;
- место размещения в библиотеке.

В случае выдачи экземпляра книги читателю в библиотеке хранится специальный вкладыш (**листок читательского требования**), в котором должны быть записаны следующие сведения:

- номер билета читателя, который взял книгу;
- дата выдачи книги;
- дата возврата.

Предусмотреть следующие **ограничения на информацию в системе**:

- Книга может не иметь ни одного автора.
- В библиотеке должны быть записаны читатели не моложе 17 лет.
- В библиотеке присутствуют книги, изданные начиная с 1960 по текущий год.
- Каждый читатель может держать на руках не более 5 книг.
- Каждый читатель при регистрации в библиотеке должен дать телефон для связи: он может быть рабочим или домашним.
- Каждая область знаний может содержать ссылки на множество книг, но каждая книга может относиться к различным областям знаний.

С данной информационной системой должны работать следующие **группы пользователей**:

- библиотекари;
- читатели;
- администрация библиотеки.

При работе с системой **библиотекарь должен иметь возможность решать следующие задачи:**

- Принимать новые книги и регистрировать их в библиотеке.
- Относить книги к одной или к нескольким областям знаний.
-

Читатель должен иметь возможность решать следующие задачи:

- Просматривать системный каталог, то есть перечень всех областей знаний, книги по которым есть в библиотеке.
- По выбранной области знаний получить полный перечень книг, которые числятся в библиотеке.

Этот пример показывает, что перед началом разработки необходимо иметь точное представление о том, что же должно выполняться в нашей системе, какие пользователи в ней будут работать, какие задачи будет решать каждый пользователь. И это правильно, ведь когда мы строим здание, мы тоже заранее предполагаем: для каких целей оно предназначено, в каком климате оно будет стоять, на какой почве, и в зависимости от этого проектировщики могут предложить нам тот или иной проект. Но, к сожалению, очень часто по отношению к базам данных считается, что все можно определить потом, когда проект системы уже создан. Отсутствие четких целей создания БД может свести на нет все усилия разработчиков, и проект БД получится «плохим», неудобным, не соответствующим ни реально моделируемому объекту, ни задачам, которые должны решаться с использованием данной БД.

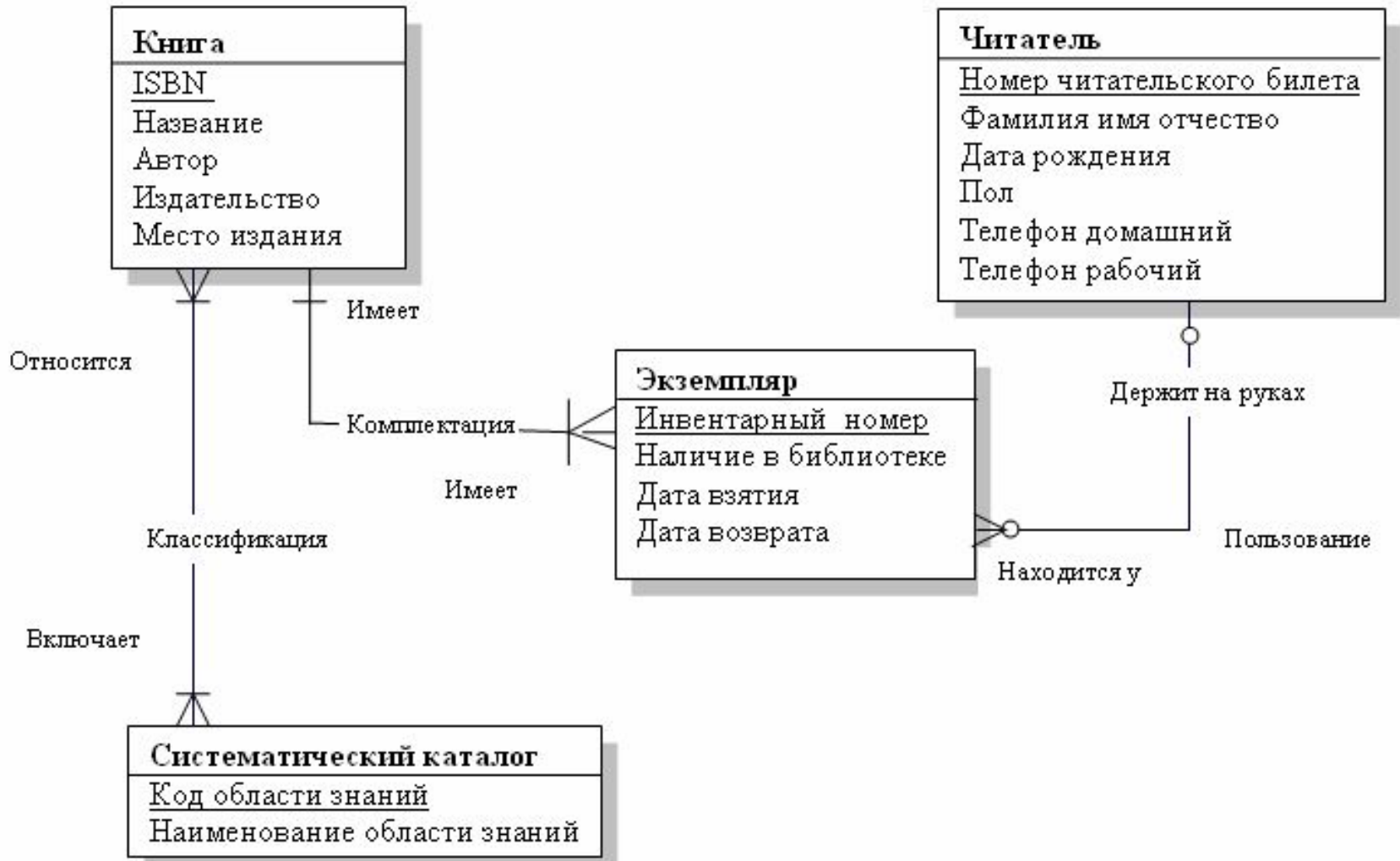
Лектор Георгица И.В.

Строим инфологическую модель



Уникальный шифр (ISBN);
Название;
Фамилии авторов (могут отсутствовать);
Место издания (город);
Издательство;
Год издания;
Количество страниц;
Стоимость книги;
Количество экземпляров книги в библиотеке.

Инфологическая модель БД «Библиотека»



Из методического указания к лабораторным работам

Общие требования к оформлению отчетов по лабораторным работам

Лабораторные работы выполняются индивидуально каждым студентом в соответствии с темой базы данных, выбранной из Приложения 1 и согласованной с преподавателем (тема может быть предложена студентом). Отчет по каждой лабораторной работе должен содержать:

- 1) номер и название лабораторной работы;
- 2) цель работы;
- 3) оборудование и программные средства;
- 4) формулировку индивидуального задания;
- 5) ход работы (описание этапов выполнения);
- 6) результаты выполнения лабораторной работы (уточняются в указаниях к выполнению лабораторных работ);
- 7) выводы и ответы на контрольные вопросы.

Лабораторная работа № 1 ИНФОЛОГИЧЕСКОЕ МОДЕЛИРОВАНИЕ

Цель работы: изучение этапов проектирования баз данных, создание инфологической модели базы данных с использованием модели "сущность-связь".

Оборудование: ПЭВМ IBM PC-AT 486 и выше.

Программные средства: электронное пособие по курсу "Базы данных".

Задание

В ходе выполнения лабораторной работы необходимо произвести системный анализ предметной области и создать инфологическую модель базы данных в соответствии с полученным индивидуальным заданием.

Программа работы

1. Изучить основы проектирования реляционных баз данных.
2. Разработать инфологическую модель базы данных в соответствии с полученным вариантом.
3. Составить отчет.(ER-модель).

Пояснения к работе

Проектирование базы данных начинается с этапа системного анализа и словесного описания информационных объектов предметной области.

Разработка инфологической модели предметной области является следующим этапом проектирования баз данных (БД). Цель инфологического моделирования – создание частично формализованного описания объектов исследуемой области в терминах некоторой семантической модели. В качестве инфологической модели выбрана модель *сущность-связь* Питера Чена (*entity-relationship model*, или *ER-модель*). ER-модель – это метод построения концептуальных моделей данных с использованием диаграмм, в котором основными

конструктивными элементами являются *сущности*, *связи* между ними и *атрибуты* (свойства) сущностей.

Сущность – объект реального мира, информацию о котором необходимо хранить в базе данных. (Понятия *тип сущности* и *экземпляр сущности* различны и соотносятся так же, как *тип переменной* и *переменная* в языках программирования).

Атрибут – характеристика сущности, имеющая имя, уникальное для конкретного типа сущности.

Связь – ассоциация между двумя или более сущностями.

Ключ – минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности.

При построении ER-модели предлагается использование графической интерпретации CASE-системы (*Computer Aided Software Engineering* - системы автоматизированного проектирования) *Power Designer* (рис. 1). В этой системе сущности обозначаются прямоугольниками, в которых содержится следующая информация: имя сущности, перечень атрибутов, в котором набор ключевых атрибутов подчеркнут. Сущности соединены связями.

При проектировании связей определяются три аспекта:

- 1) общее имя связи и имена ролей со стороны обеих сущностей;
- 2) множественность связи;
- 3) обязательность связи.



Рис. 1. Пример связывания сущностей *Студент* и *Преподаватель*.

Типы связи по множественности: *один-к-одному* (1:1), *один-ко-многим* (1:M), *многие-ко-многим* (M:M). Связь *один-к-одному* означает, что экземпляр одной сущности связан только с одним экземпляром другой сущности; связь *один-ко-многим* означает, что один экземпляр сущности может быть связан с несколькими экземплярами другой сущности; связь *многие-ко-многим* означает, что один экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и наоборот, один экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности (рис. 2).



Рис. 2. Пример моделирования связи *многие-ко-многим*.

Множественность графически изображается разветвлением линии связи. Обязательность связи обозначается следующим образом: перпендикуляр к линии связи - обязательная связь (хотя бы один экземпляр сущности *должен* участвовать в связи); кружок на линии связи - необязательная связь (экземпляр сущности *может* участвовать в связи).

Порядок чтения информации в ER-моделях:

имя сущности - имя роли со стороны этой сущности - имя связи - обязательность связи - множественность связи - имя другой сущности.

Пример фрагмента ER-модели, представленный на рис. 1, можно интерпретировать следующим образом.

Со стороны сущности *преподаватель*:

преподаватель - руководит - дипломным проектированием - не обязательно (т.е. не каждый преподаватель должен быть руководителем дипломного проектирования) - одного или более - студентов.

Со стороны сущности *студент*:

студент - пишет диплом - в процессе дипломного проектирования - обязательно под руководством - только одного - преподавателя.

Указания к выполнению

В отчете по лабораторной работе необходимо:

- кратко описать предметную область и круг потенциальных пользователей, чьи информационные потребности может обеспечивать проектируемая база данных;
- согласно этому описанию выделить сущности (не менее 3-х);
- для каждой сущности определить атрибуты, в том числе ключевые;
- охарактеризовать связи между сущностями, решить вопросы множественности и обязательности связи со стороны каждой сущности;
- начертить ER-модель базы данных согласно заданию.

Контрольные вопросы

1. Дайте определение понятий *информация* и *данные*, чем они отличаются?
2. Что такое модель данных?
3. Дайте классификацию моделей данных.
4. Что называют инфологической моделью данных?
5. Какие компоненты ER-модели вы знаете?
6. Приведите пример связи 1:M и M:M в разработанной ER-модели.
7. Что такое обязательность связи и как она изображается графически?
8. Как выглядит ER-модель проектируемой вами базы данных в графической нотации Питера Чена?

Краткая история развития БД

До 1963 года – Обработка файлов

	Описание наиболее значимых событий
1959	Образована Ассоциация по языкам систем обработки данных (<i>Conference on Data System Languages, CODASYL</i>) – организация, являющаяся автором языка программирования COBOL, языковых спецификаций сетевой модели данных, а также ряда важных концептуальных и аналитических материалов по технологиям баз данных

1963-1980 Эра обработки нереляционных баз данных.
Иерархическая и сетевая модели данных. Первое поколение СУБД

	Описание наиболее значимых событий
1963	Завершена разработка первой СУБД IDS (Integrated Data Storage) – Интегрированное хранилище данных, компания General Electric, под руководством Чарльза Бахмана (C. Bachman) – сетевая модель данных
1968- 1969	Компания IBM выпустила первую версию своей СУБД IDMS (Integrated Database Management System) для платформы IBM/360 – иерархическая модель данных

1980- н. время. Реляционные базы данных. Второе поколение СУБД

	Описание наиболее значимых событий
1970	Публикация работы Эдгара Кодда (E.Codd) в журнале « <i>Communication of A CM</i> », которая послужила основой разработки теории реляционных баз данных.
1975	Опубликован отчет рабочей группы ANSI/X3/SPARC о трехуровневой архитектуре СУБД
1976	Опубликована широко известная статья Питера Чена (P. Chen), в которой предложена модель данных «сущность-связь» – ER-модель
1978	Опубликован заключительный отчет исследовательской группы ANSI/X3/SPARC по базам данных
1979-1984	В компании IBM инициирован исследовательский проект System R* с целью изучения методов реализации систем распределенных реляционных баз данных

1980	Выпуск первых СУБД для мэйнфреймов: DB2 (корпорация IBM) , Oracle (Oracle)
1981	За цикл работ по созданию теории реляционных баз данных АСМ удостоила Э. Кодда Тьюринговской премии
1982	Первые СУБД для микрокомпьютеров
1986	Принят стандарт ANSI языка SQL-86
1987	Принят стандарт ISO SQL-87 — первый международный стандарт языка SQL
1991	Компания Microsoft выпустила Access
1992	Принят международный стандарт ISO SQL-92
1999	Принят стандарт ISO/IEC/ANSI SQL 1999 Издательство «Вильямс» (Киев, Москва, СПб) выпустило в русском переводе 6-е издание книги Криса Дейта «Введение в системы баз данных» (К. J. Date)

1985- настоящее время

В ответ на все возрастающую сложность приложений баз данных появились две новые разновидности систем: **объектно-ориентированные СУБД**, или **ОО СУБД** (Object-Oriented DBMS — OODBMS), и **объектно-реляционные СУБД**, или **ОР СУБД** (Object-Relational DBMS — ORDBMS). Реализации подобных моделей представляют собой СУБД **третьего поколения**.

	Описание наиболее значимых событий
1985	Развитие интереса к объектно-ориентированным СУБД (ООСУБД)
1995	Первые приложения баз данных для Интернета
1997	Применение XML к обработке баз данных
2000- н.в.	Новые стандарты, технологии, модели, архитектуры систем БД

Поколения СУБД. Их характеристика.

- ***К СУБД первого поколения относят СУБД на основе сетевой модели данных (их иногда называют CODASYL-системы) и системы на основе иерархических подходов.***
- ***СУБД второго поколения – реляционные***
- ***СУБД третьего поколения – объектно-реляционные и объектно-ориентированные.***

Критерии классификации СУБД

По степени универсальности (сфере применения) :

- СУБД общего назначения (СУБД ОН) и специализированные СУБД (СпСУБД).

По используемой модели данных

- иерархические, сетевые, реляционные; объектно-ориентированные СУБД.

По методам организации хранения и обработки данных :

- централизованные (локальные, файл – серверные, клиент-серверные) и распределённые СУБД.

По сфере применения

- справочные системы и системы обработки данных.

Классификация по масштабу систем:

- персональные; уровня группы, отдела, предприятия; корпоративные; географически распределенные.

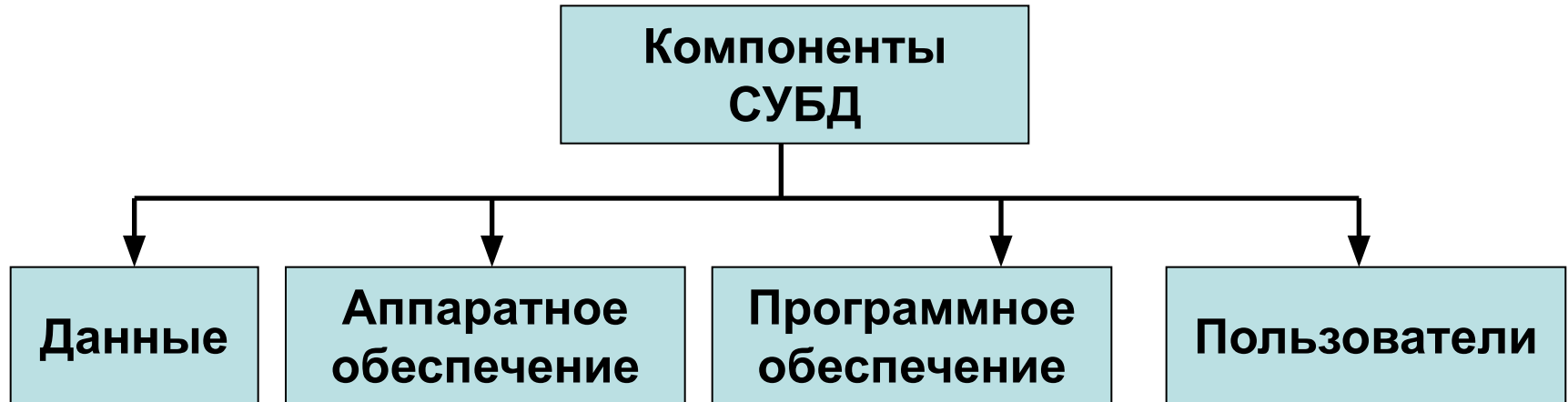
Что такое распределенная СУБД?

Распределенная СУБД – это СУБД, поддерживающая работу с распределенными базами данных. Одно из определений распределенной БД:

"Распределенная БД - это множество физических баз данных, которые выглядят для пользователя как одна логическая БД". К сожалению на сегодняшний день ни одна СУБД полностью не реализует это определение. Наиболее близко к его реализации подошли следующие СУБД:

- - Informix On-Line фирмы Informix Software;
- - Ingres Intelligent Database фирмы Ingres Corp;
- - Oracle (version 7) фирмы Oracle Corp;
- - Sybase System 10 фирмы Sybase Inc.

Основные компоненты СУБД и их состав

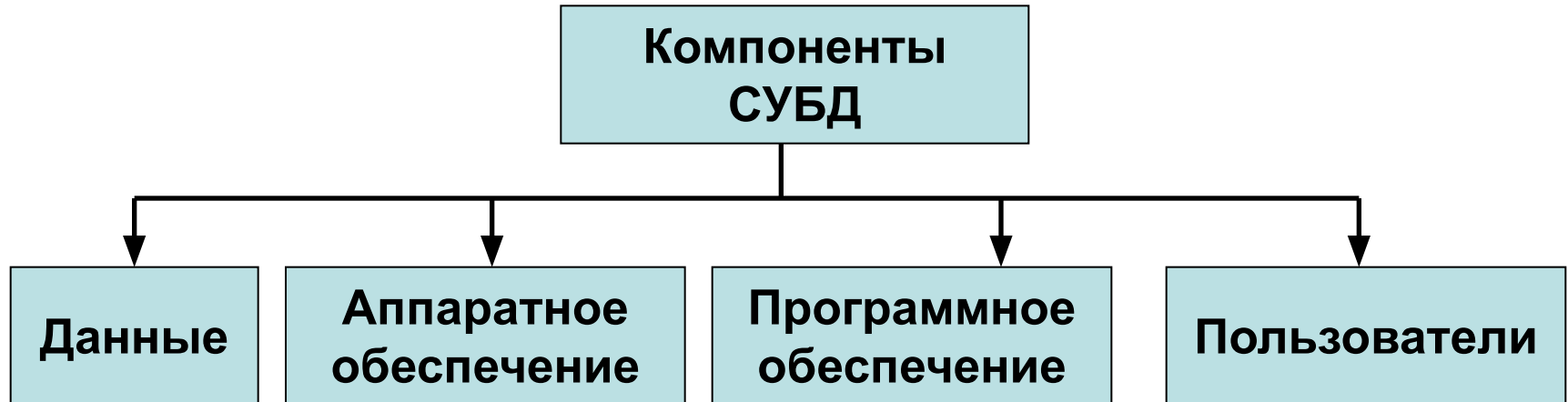


Данные должны быть интегрированными и общими.

Интегрирование – возможность представлять базу данных как объединение нескольких отдельных файлов данных полностью или частично не перекрывающихся.

Общие – возможность использования отдельных областей данных в БД несколькими различными пользователями, причем даже в одно и тоже время(одновременный доступ).

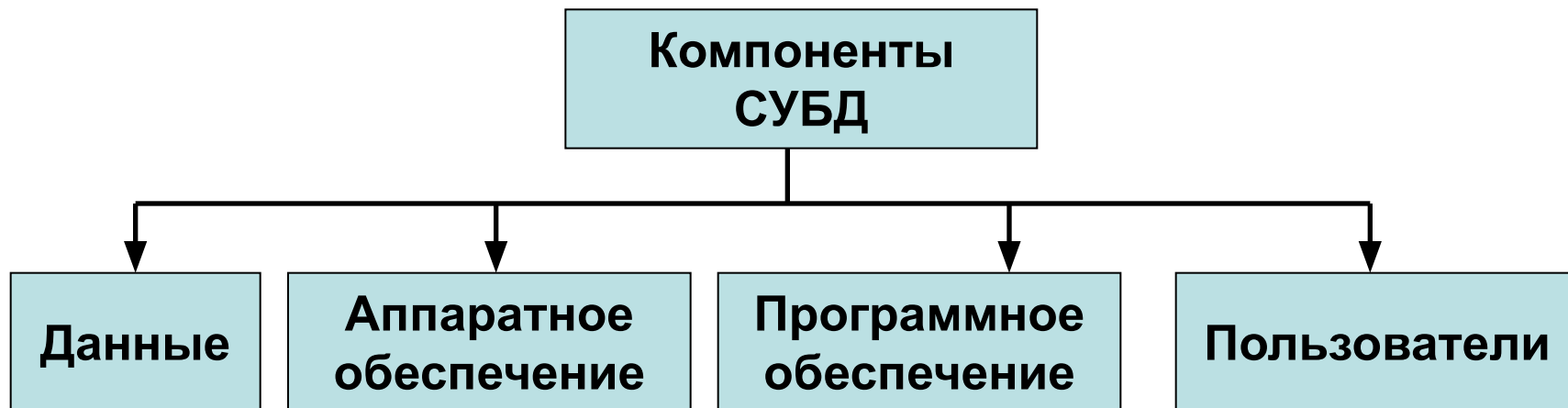
Основные компоненты СУБД и их состав



Накопители для хранения информации (обычно диски с перемещаемыми головками) вместе с подсоединенными устройствами ввода-вывода, контроллерами устройств, каналами ввода-вывода и т.д.

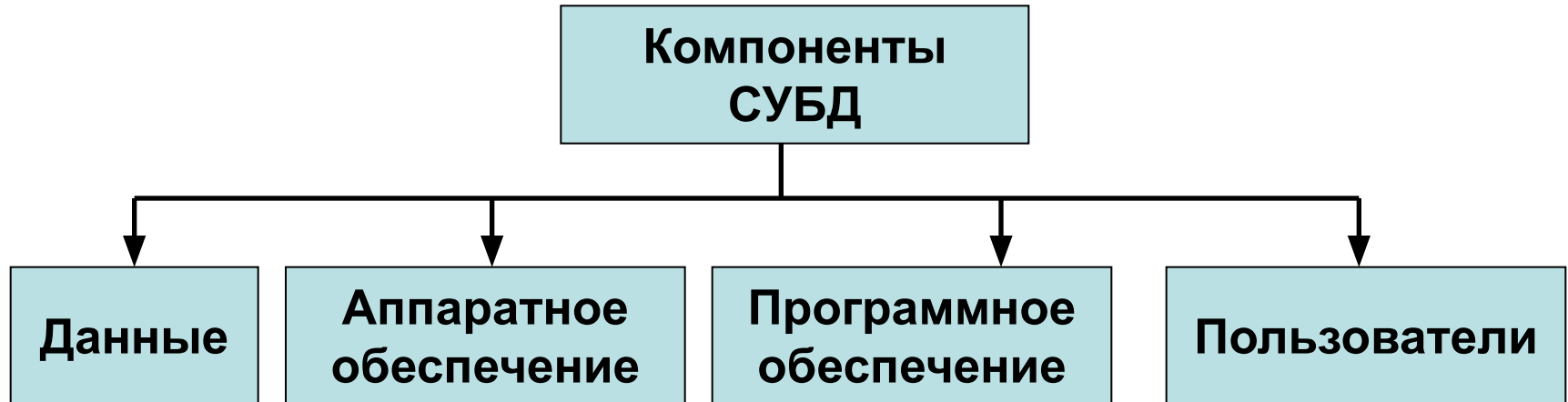
Процессор или процессоры вместе с основной памятью, которая используется для поддержки работы программного обеспечения системы

Основные компоненты СУБД и их состав.



Диспетчер базы данных (database manager), или система управления базами данных СУБД (database management system (DBMS)). СУБД предоставляет пользователю возможность рассматривать БД как объект более высокого уровня по сравнению с аппаратным обеспечением, а также поддерживает выражаемые в терминах высокого уровня пользовательские запросы (SQL). Кроме СУБД, в программном обеспечении – утилиты, средства разработки приложений, средства проектирования, генераторы отчетов и другие.

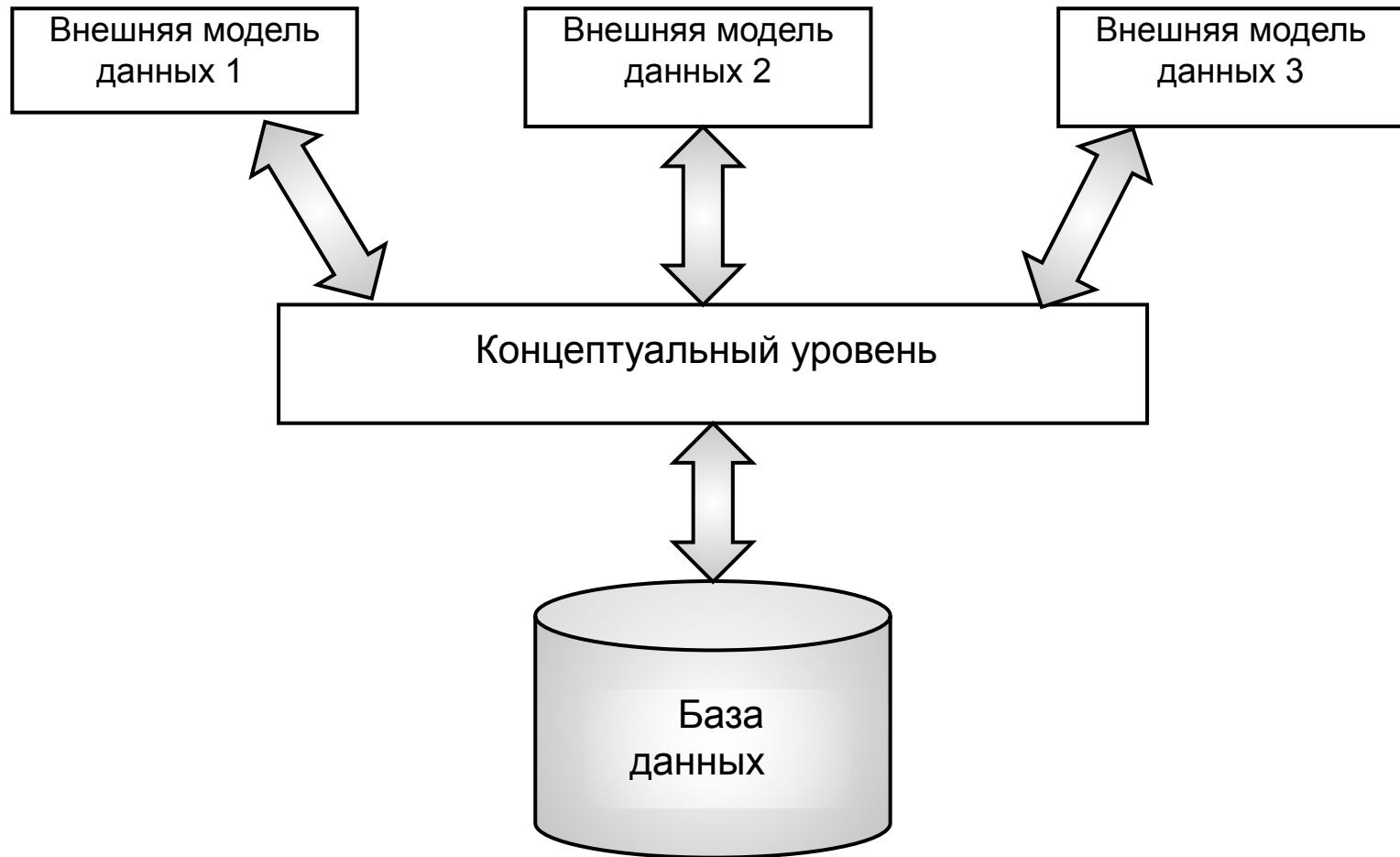
Основные компоненты СУБД и их состав



Работающие с базами данных пользователи обладают различными знаниями, навыками и сталкиваются с решением различных задач:

- конечные пользователи;*
- разработчики баз данных;*
- разработчики приложений;*
- администраторы баз данных.*

Схема трехуровневой архитектуры ANSI для СУБД. Описание назначений уровней.



Логическая и физическая независимость уровней при работе с данными.

Эта архитектура позволяет обеспечить логическую (между уровнями 1 и 2) и физическую (между уровнями 2 и 3) независимость при работе с данными.

Логическая независимость предполагает возможность изменения одного приложения без корректировки других приложений, работающих с этой же базой данных. Физическая независимость предполагает возможность переноса хранимой информации с одних носителей "на другие при сохранении работоспособности всех приложений, работающих с данной базой данных. Это именно то, чего не хватало при использовании файловых систем.

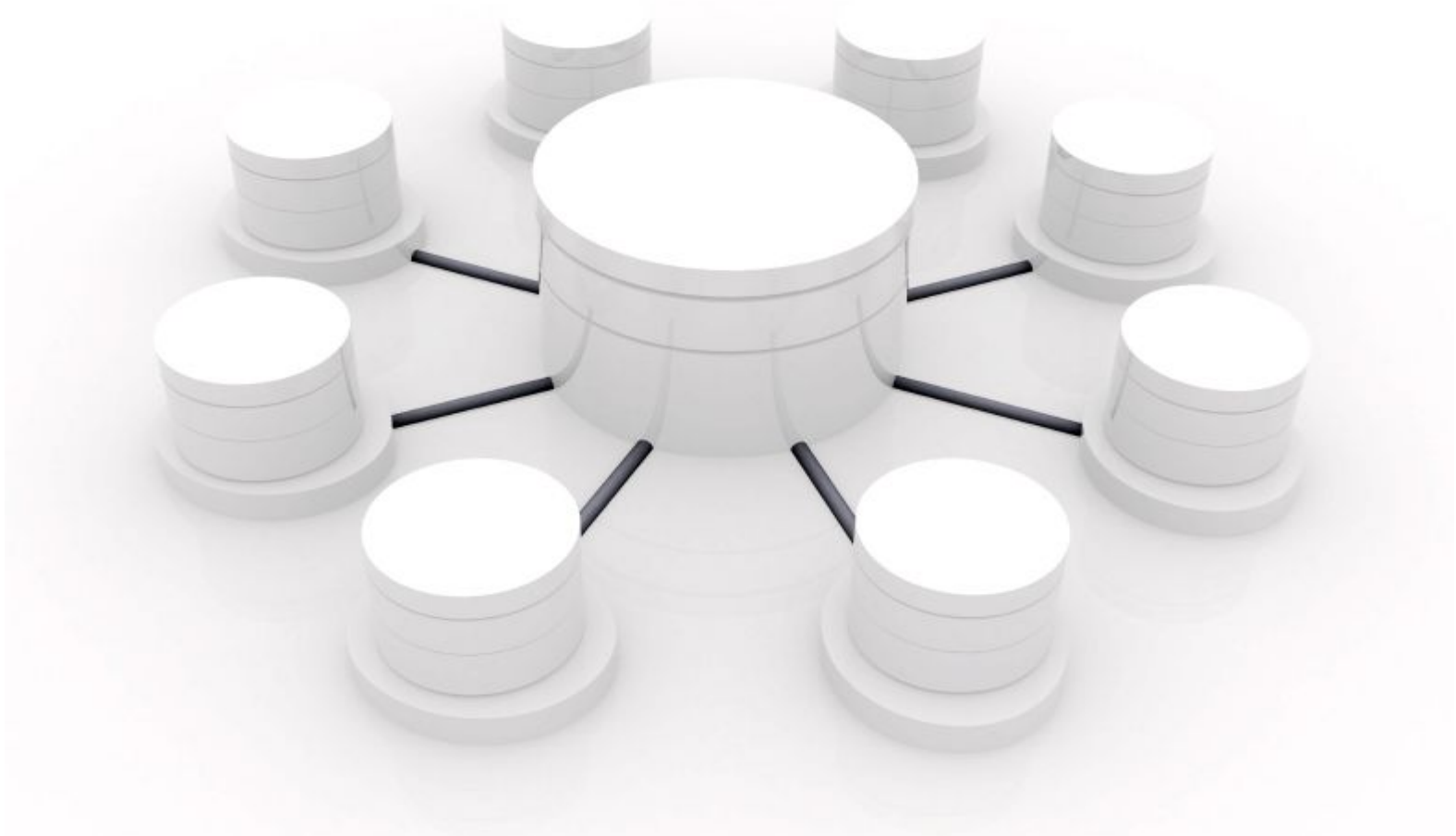
Выделение концептуального уровня позволило разработать аппарат централизованного управления базой данных.

Определение схемы и подсхемы БД.

С понятием «трехуровневая архитектура баз данных» связаны понятия «схема» и «подсхема».

Описание общей логической структуры базы данных называют *схемой*. Ее называют иногда *общей моделью* данных. **На основе одной схемы можно составить много различных подсхем** (в зависимости от требований пользователей к БД).

Модель данных. Классификация моделей данных.
Инфологическая модель.



Определение понятия «модель»

Модель - это такой материальный или мысленно представляемый объект, который в процессе познания (созерцания, анализа и синтеза) замещает объект-оригинал.

- Модель — это упрощенное представление реального устройства, процесса, явления.
- Процесс построения и исследования моделей называется **моделированием**, облегчает изучение имеющихся в реальном устройстве (процессе, явлении) свойств и закономерностей. Применяют для нужд познания (созерцания, анализа, синтеза).

Определение понятия «модель данных»

Модель данных — это некоторая интерпретация данных, связанная с этапом проектирования БД, которая трактуется как сведения, имеющие определенную структуру.

Модель данных – это логическое определение объектов, связанное с этапом проектирования БД.

Из «Энциклопедии технологий баз данных» М. Р. Когаловского

М.087 Модель данных **Data Model**

Аббрев. англ.: DM

1. Совокупность правил структурирования данных в базах данных, допустимых операций над ними и ограничений целостности, которым они должны удовлетворять. Механизмы каждой СУБД конструируются на основе той или иной модели данных, реализующей комплекс языковых средств определения данных и манипулирования данными, воплощающих ее концепции. Устаревшая трактовка этого понятия интерпретирует модель данных как структуру конкретной базы данных.
2. Система типов данных, типов связей между ними и допустимых видов ограничений целостности, которые могут быть для них определены. Здесь имеется в виду современное понимание типа данных как носителя свойств, определяющих и состояние экземпляров типа, и их поведение.
3. Метамодель для описания моделей предметной области в среде выбранной СУБД.

532

Модель данных CODASYL

CODASYL Data Model

Хрестоматийная модель данных, относящаяся к категории так называемых *графовых моделей*. Представляет собой разновидность сетевой модели данных, имеющую дело со структурами данных, конструируемыми из *записей*, которые подобны по своей структуре записям языка COBOL и связей "1:1" причем

Модель данных внешняя (ANSI/SPARC)

External Data Model

Поддерживаемая используемой СУБД модель данных внешнего уровня архитектуры систем баз данных. В терминах этой модели описываются внешние схемы конкретных баз данных.

Модель данных внутренняя (ANSI/SPARC)

Internal Data Model

Поддерживаемая используемой СУБД модель данных внутреннего уровня архитектуры систем баз данных. В терминах этой модели описываются внутренние схемы конкретных баз данных.

Модель данных временная

Temporally-Oriented Data Model

Модель данных, предусматривающая поддержку *концепции времени*. Развитые модели такого рода поддерживают концепцию *двумерного времени*. При этом различаются *действительное время* — время, когда факт имел место в реальности, и *время транзакции* — время, когда факт помещается в систему. Вместе с тем возможно использование и третьего аспекта времени — *времени, определяемого пользователем*

Модель данных графовая **Graph-Oriented Data Model**

Модель данных, в которой допустимые структуры данных могут быть представлены в виде графа общего или какого-либо специального вида, например в виде дерева. Необходимую группу операций в языке манипулирования данными, основанном на ней, представляют *навигационные операции*. Операции над данными здесь имеют *указательный* характер (record-by-record). К графовым моделям относятся иерархическая и сетевая модели данных CODASYL.

Модель данных иерархическая (HDB) **Hierarchical Data Model**

Модель данных, в основе которой используется *иерархическая древовидная структура данных*. Вершинами этой структуры являются записи соответствующего уровня, получаемые также *сегментами*, состоящие из простых элементов. При этом родительской записью являются записи более высокого уровня. При этом родительской записью являются записи более высокого уровня.

Модель данных концептуальная (ANSI/SPARC) **Infological Data Model**

Поддерживаемая используемой СУБД модель данных *концептуального уровня* архитектуры систем баз данных. В терминах этой модели описываются концептуальные конкретные базы данных.

Модель данных многомерная (MDM) **Multi-Dimensional Data Model**

Син. русск.: Модель измерений

Модель данных, оперирующая *многомерными представлениями данных* (в виде гиперкуба). Разновидности многомерной модели стали широко использоваться в середине 1990-х годов в связи с развитием технологий OLAP.

Модель данных объектная {ODM}

Object Data Model

Модель данных объектно-ориентированная {ODM}

Object-Oriented Data Model

См. Модель данных объектная.

Модель данных объектно-реляционная {ODM, RDM}

Object-Relation Data Model

Гибридная модель данных, сочетающая возможности реляционных моделей с объектными свойствами данных.

Модель данных, основанная на значениях

Value-Based Data Model

Син. англ.: Value-Oriented Data Model

Модель данных, в которой не поддерживается концепция *индивидуальности объектов*. К моделям этого класса относят реляционную модель данных, а также логические модели.

Модель данных, основанная на индивидуальности

Identity-Based Data Model

См. Объектно-ориентированная модель данных.

Модель данных реляционная {RDM}

Relational Data Model

Модель данных реляционная расширенная {RDM}

Extended Relational Data Model

Модель данных семантическая

Semantic Data Model

Модель данных сетевая

Network Data Model

Модель данных “сущностей-связей” {ER}

Entity-Relationship Data Model

Син. русск.: ER-модель

Опубликованная П. Ченом (P. Chen) в 1976 г. модель данных, кратко называемая *ER-моделью* и предназначенная по замыслу автора для описания *модели предметной области* в процессе проектирования базы данных. Однако эта модель была использована позднее в ряде экспериментальных СУБД в качестве модели данных внешнего уровня системы. Основными элементами ER-модели являются именованные *множества сущностей*, *множества связей* между ними, которые могут быть двуместными или многоместными, ориентированными или неориентированными. Сущности и связи обладают *атрибутами*. В модели вводится ограничение целостности данных, ассоциируемое с двумя множествами сущностей, — *зависимость по существованию*. Это ограничение является близким по смыслу ограничению целостности по ссылкам в реляционной модели.

Карпова Т.С. Базы данных: модели, разработка, реализация

Понятие «данные» в концепции баз данных — это набор конкретных значений, параметров, характеризующих объект, условия, ситуацию или любые другие факторы. Примеры данных: Петров Николай Степанович, \$30 и т. д. Данные не обладают определенной структурой, данные становятся информацией тогда, когда пользователь задает им определенную структуру, то есть осознает их смысловое содержание. Поэтому центральным понятием в области баз данных является понятие модели. Не существует однозначного определения этого термина, у разных авторов эта абстракция определяется с некоторыми различиями, но тем не менее можно выделить нечто общее в этих определениях.

Модель данных — это некоторая абстракция, которая, будучи приложима к конкретным данным, позволяет пользователям и разработчикам трактовать их уже как информацию, то есть сведения, содержащие не только данные, но и взаимосвязь между ними.

На рис. 2.3 представлена классификация моделей данных.

В соответствии с рассмотренной ранее трехуровневой архитектурой мы сталкиваемся с понятием модели данных по отношению к каждому уровню. И действительно, физическая модель данных оперирует категориями, касающимися организации внешней памяти и структур хранения, используемых в данной операционной среде. В настоящий момент в качестве физических моделей используются различные методы размещения данных, основанные на файловых структурах: это организация файлов прямого и последовательного доступа, индексных файлов и инвертированных файлов, использующих различные методы кэширования, взаимосвязанных файлов. Кроме того, современные СУБД широко используют страничную организацию данных. Физические модели данных, основанные на страничной организации, являются наиболее перспективными.



Рис. 2.3. Классификация моделей данных

Наибольший интерес вызывают модели данных, используемые на концептуальном уровне. По отношению к ним внешние модели называются подсхемами и используют те же абстрактные категории, что и концептуальные модели данных.

Кроме трех рассмотренных уровней абстракции при проектировании БД существует еще один уровень, предшествующий им. Модель этого уровня должна выражать информацию о предметной области в виде, независимом от используемой СУБД. Эти модели называются *инфологическими*, или *семантическими*, и отражают в естественной и удобной для разработчиков и других пользователей форме информационно-логический уровень абстрагирования, связанный с фиксацией и описанием объектов предметной области, их свойства и их взаимосвязей.

Инфологические модели данных используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения, а *даталогические* модели уже поддерживаются конкретной СУБД.

Документальные модели данных соответствуют представлению о слабоструктурированной информации, ориентированной в основном на свободные форматы документов, текстов на естественном языке.

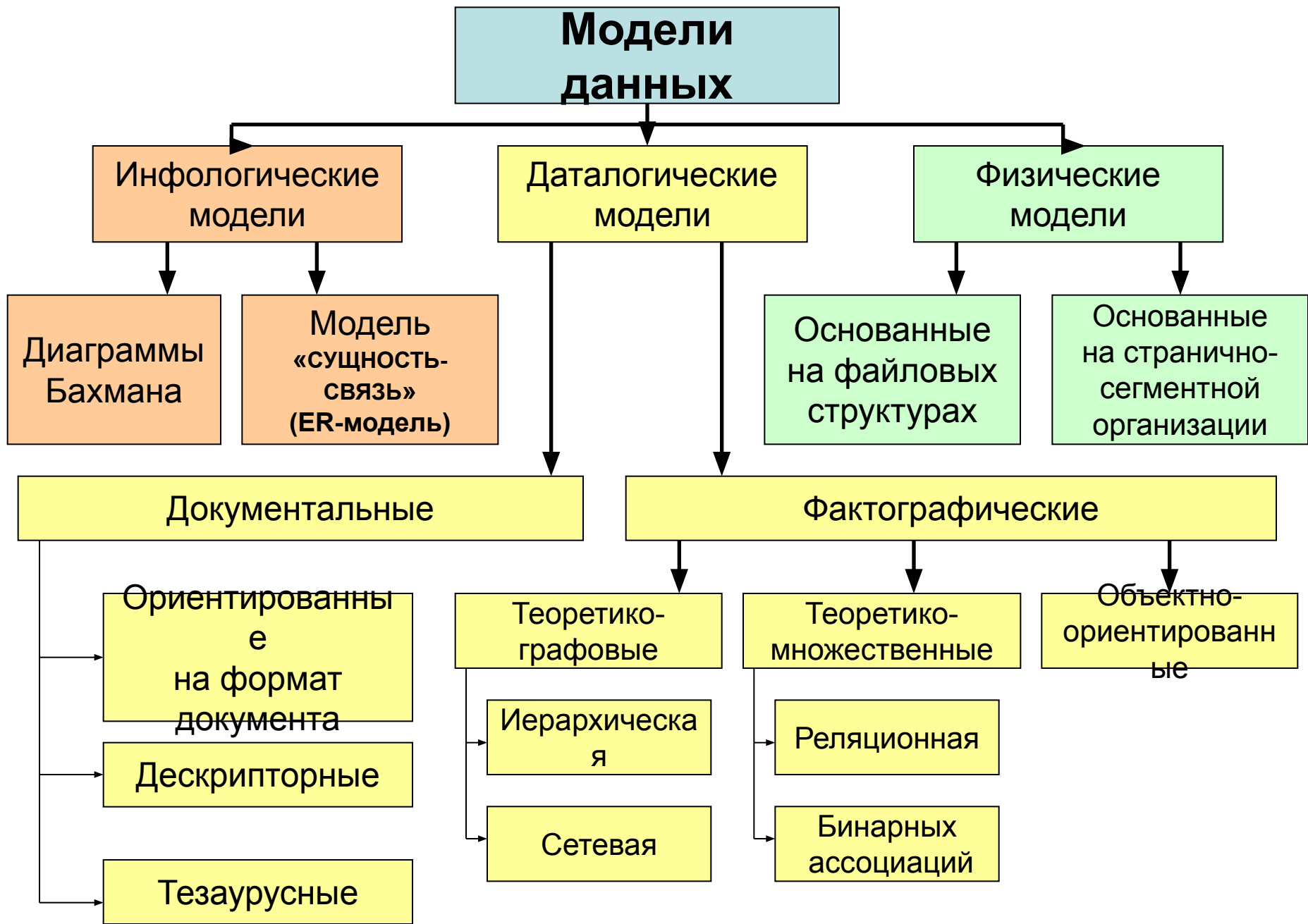
Модели, основанные на языках разметки документов, связаны, прежде всего, со стандартным общим языком разметки — SGML (Standart Generalised Markup Language), который был утвержден ISO в качестве стандарта еще в 80-х годах. Этот язык предназначен для создания других языков разметки, он определяет допустимый набор тегов (ссылок), их атрибуты и внутреннюю структуру документа. Контроль за правильностью использования тегов осуществляется при помощи специального набора правил, называемых DTD-описаниями, которые используются программой клиента при разборе документа. Для каждого класса документов определяется свой набор правил, описывающих грамматику соответствующего языка разметки. С помощью SGML можно описывать структурированные данные, организовывать информацию, содержащуюся в документах, представлять эту информацию в некотором стандартизованном формате. Но ввиду некоторой своей сложности SGML использовался в основном для описания синтаксиса других языков (наиболее известным из которых является HTML), и немногие приложения работали с SGML-документами напрямую.

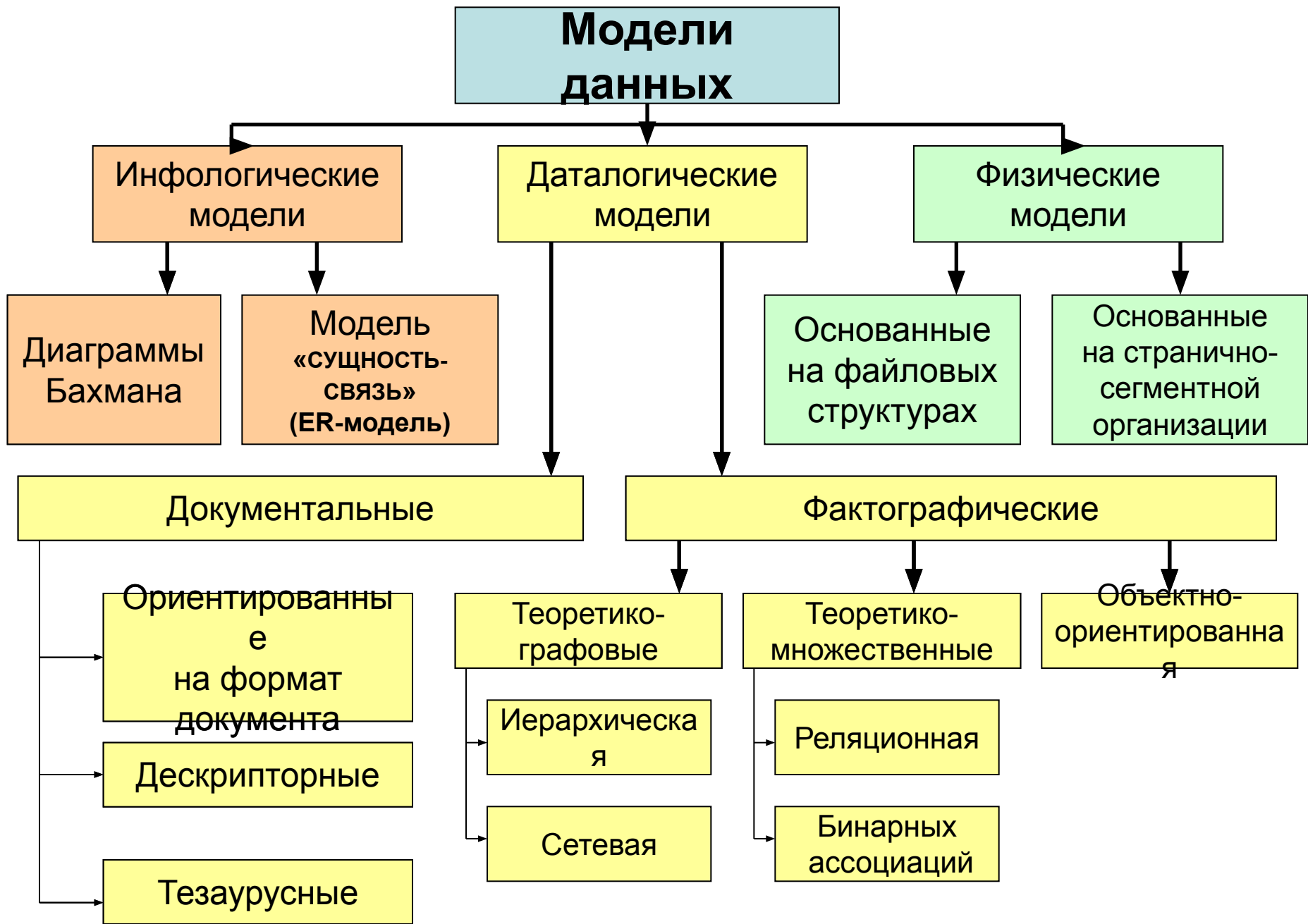
Гораздо более простой и удобный, чем SGML, язык HTML позволяет определять оформление элементов документа и имеет некий ограниченный набор инструкций — тегов, при помощи которых осуществляется процесс разметки. Инструкции HTML в первую очередь предназначены для управления процессом вывода содержимого документа на экран программы-клиента и определяют этим самым способ представления документа, но не его структуру. В качестве элемента гипертекстовой базы данных, описываемой HTML, используется текстовый файл, который может легко передаваться по сети с использованием протокола HTTP. Эта особенность, а также то, что HTML является открытым стандартом и огромное количество пользователей имеет возможность применения возможности этого языка для оформления своих документов, безусловно, повлияли на рост популярности HTML и сделали его сегодня главным механизмом представления информации в Интернете.

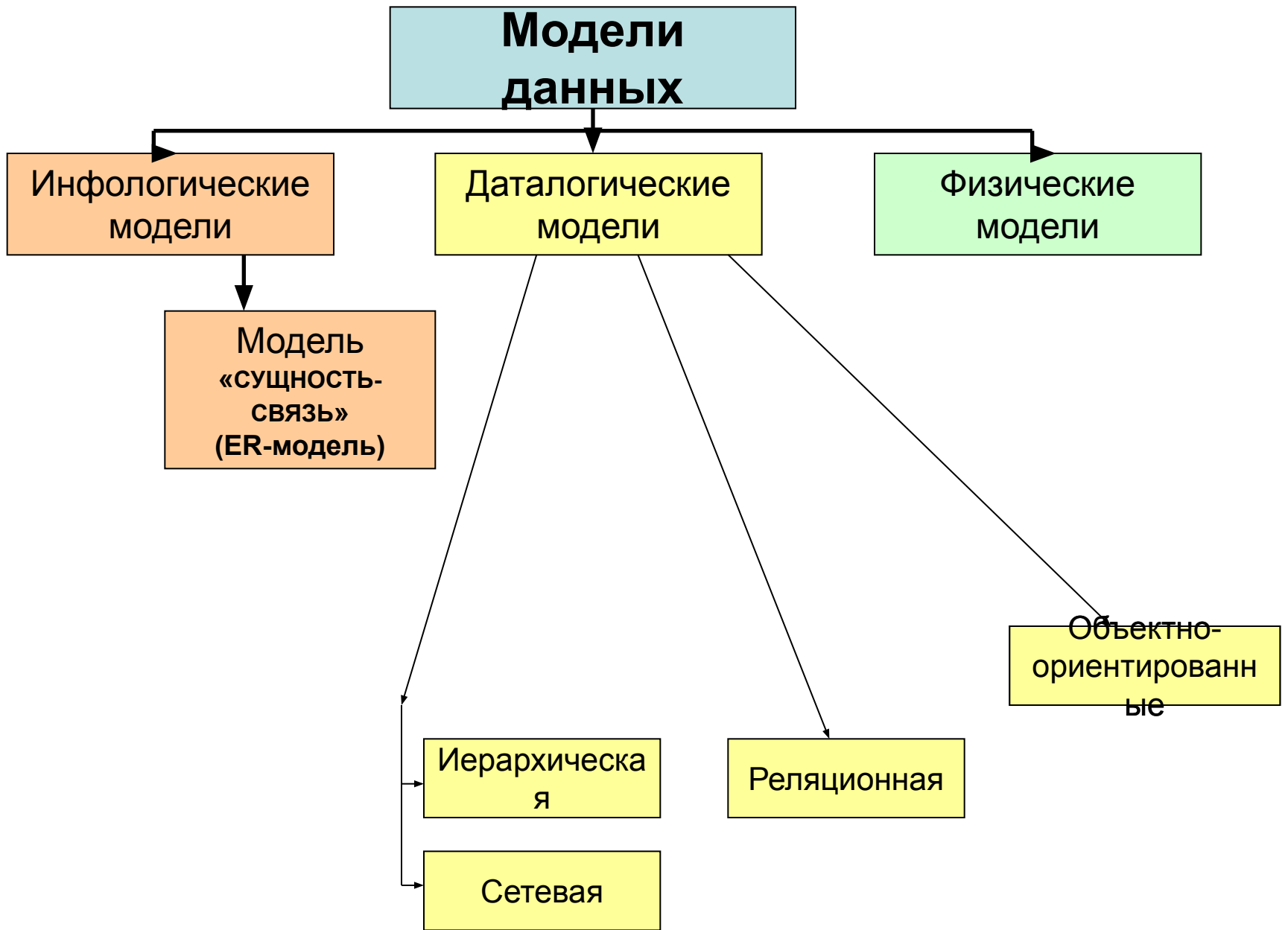
Однако HTML сегодня уже не удовлетворяет в полной мере требованиям, предъявляемым современными разработчиками к языкам подобного рода. И ему на смену был предложен новый язык гипертекстовой разметки, мощный, гибкий и, одновременно с этим, удобный язык XML. В чем же заключаются его достоинства?

XML (Extensible Markup Language) — это язык разметки, описывающий целый класс объектов данных, называемых XML-документами. Он используется в качестве средства для описания грамматики других языков и контроля за правильностью составления документов. То есть сам по себе XML не содержит никаких тегов, предназначенных для разметки, он просто определяет порядок их создания.

Тезаурусные модели основаны на принципе организации словарей, содержат определенные языковые конструкции и принципы их взаимодействия в заданной грамматике. Эти модели







Модели данных

Инфологические модели

Модель «сущность-связь» (ER-модель)

Даталогические модели

Иерархическая

Сетевая

Реляционная

Объектно-ориентированная

Физические модели

Инфологическое моделирование связано со 2-м этапом проектирования БД: созданием формализованного описания предметной области

Логическое (или даталогическое) моделирование осуществляется после этапа выбора СУБД. Этот тип модели полностью зависит от типа модели, поддерживаемой выбранной системой.

Физическое моделирование заключается в выборе эффективного размещения БД на внешних носителях для обеспечения наиболее эффективной работы.

Предметная область
(часть реального мира отображаемая в базе данных)



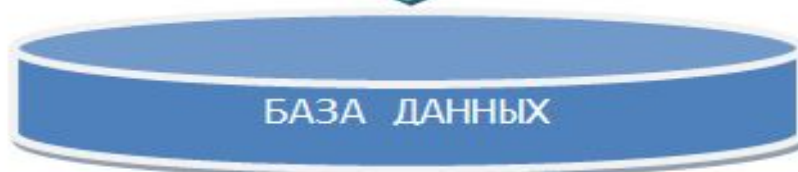
ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ
Формализованное, обобщенное, не привязанное к каким-либо СУБД описание предметной области



ДАТАЛОГИЧЕСКАЯ МОДЕЛЬ ДАННЫХ
Описание на языке конкретной СУБД



ФИЗИЧЕСКАЯ МОДЕЛЬ ДАННЫХ
Описание хранимых данных на уровне операционной системы



Модели описания, используемые СУБД

Инфологическая модель предметной области – это частично формализованное описание объектов предметной области в терминах некоторой семантической модели. Более традиционная из них называется моделью сущность – связь (E/R- модель, entity-relationship), имеет графическую природу (прямоугольники, стрелки, ромбы).

Модель «сущность – связь»

- *E/R-модель (или модель сущность – связь) создана Питером Ченом в 1976 году.*

E/R-модель стала фактическим стандартом при инфологическом моделировании БД по следующим причинам.

1) большинство современных CASE-средств содержат инструментальные средства для описания данных в формализме этой модели;

2) разработаны методы автоматического преобразования проекта БД из E/R-модели в реляционную, при этом преобразование выполняется в даталогическую модель, соответствующую конкретной СУБД.

Компоненты E/R-модели:

1. **Сущность** — это реальный или представляемый набор однотипных объектов, информация о котором характеризует предметную область. В системе существует множество экземпляров данной сущности (если проводить аналогию с ООП, то множества сущностей – класс, каждая сущность – объект (экземпляр класса)).

2. **Атрибуты** – значения, описывающие свойства сущности. Набор атрибутов, однозначно идентифицирующий конкретный экземпляр сущности, называется **ключевым**.

3. **Связи** – бинарные ассоциации, показывающие, каким образом сущности соотносятся или взаимодействуют между собой. Связь может существовать между двумя разными сущностями или между сущностью и ей же самой. Если есть связь между двумя сущностями, то она определяет взаимосвязь между экземплярами одной и другой сущности.

Типы связей в ER-модели

С точки зрения множественности:

- *Один к одному (1:1),*
- *один ко многим (1:M),*
- *многие ко многим (M:M).*

С точки зрения обязательности:

- *Обязательная (экземпляр первой сущности ДОЛЖЕН быть связан с экземпляром второй сущности)*
- *Необязательная (экземпляр первой сущности МОЖЕТ быть связан с экземпляром второй сущности)*

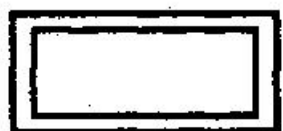
Инфологическая модель БД «Библиотека»



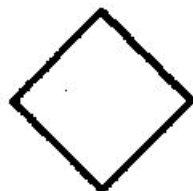
*Рассмотрим трактовку
инфологической модели ее
создателем, Питером Ченом.*



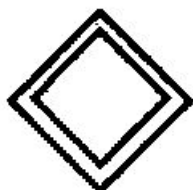
Класс сущности



Класс слабой сущности



Тип связи



Определяющий тип связи



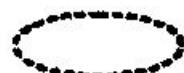
Атрибут



Ключевой атрибут



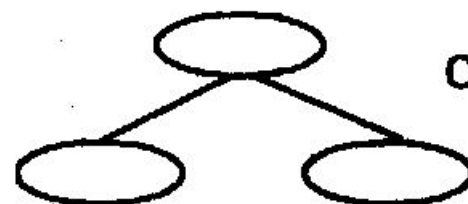
Дискриминатор
или частичный ключ



Производный атрибут



Многозначный атрибут



Составной атрибут

Обозначения кардинальности

1 Не более одной связанной сущности

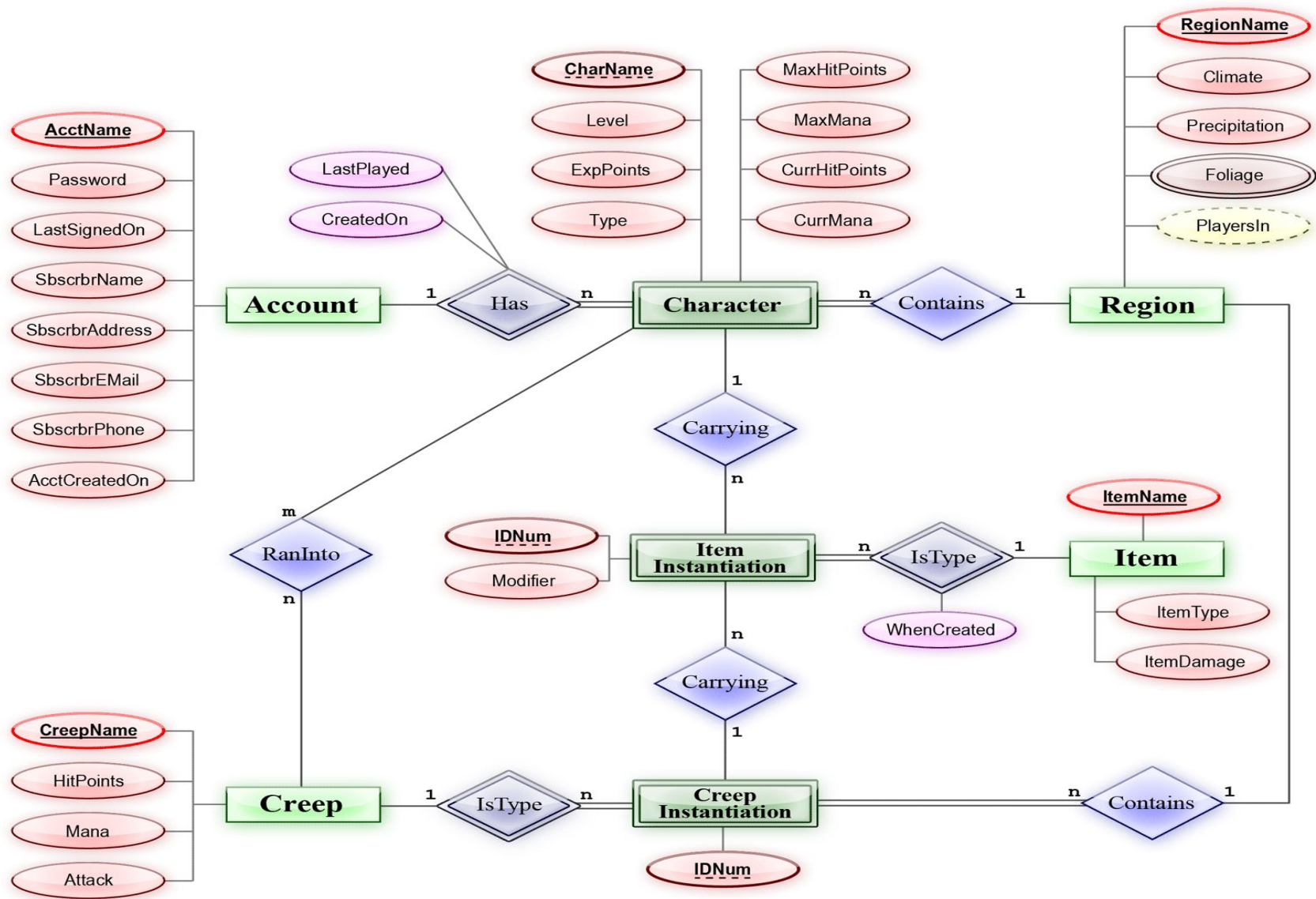
M Несколько (ноль и более) связанных сущностей

$i \dots j$ По меньшей мере i , но не более j связанных сущностей

==== Должен участвовать в связи

===== Может участвовать в связи

Начертите инфологическую модель выбранной предметной области в графической нотации Питера Чена



Вопросы к первому модулю (дисциплина «Базы данных»)

1. Назовите две основных области использования вычислительной техники.
2. Кратко охарактеризуйте основные этапы эволюции концепций БД.
3. Дайте определения следующих фундаментальных понятий теории баз данных:
 4. – информация;
 5. – данные;
 6. – база данных
 7. – система управления базами данных (СУБД);
 8. – информационная система.
9. Поколения СУБД. Их характеристика.
10. Критерии классификации СУБД.
11. Перечислите СУБД в зависимости от используемой модели данных.
12. Что такое распределенная СУБД?
13. Назовите СУБД, поддерживающие клиент-серверную архитектуру.
14. Попробуйте самостоятельно классифицировать СУБД «1С» по изученным критериям.
15. Перечислите основные функции СУБД.
16. Назовите основные компоненты СУБД и их состав.
17. Чем, по вашему мнению, отличаются понятия «администратор баз данных» и «администратор данных»?
18. Начертите схему трехзвенной архитектуры ANSI для СУБД. Опишите назначение уровней.
19. Что предполагает логическая и физическая независимость уровней при работе с данными.
20. Дайте определение схемы и подсхемы БД.
21. Дайте определение понятия «модель данных».
22. Как модели данных связаны с этапами проектирования базы данных?
23. Какой способ инфологического моделирования используется в лабораторной работе №1?
24. Какие теоретико-графовые модели вы знаете?
25. Перечислите этапы проектирования БД.
26. Каковы основные требования к проектируемой БД?
27. Охарактеризуйте функциональный и предметный подходы к описанию предметной области.

Вопросы к первому модулю (дисциплина «Базы данных»)

28. Дайте определение инфологической модели.
29. Назовите автора и год создания метода инфологического моделирования «сущность-связь».
30. Перечислите основные компоненты ER-модели.
31. Дайте определение понятий:– сущность;
32. – связь;
33. – атрибут;
34. – ключевой атрибут.
35. Какие типы связей вы знаете?
36. Начертите инфологическую модель выбранной предметной области в графической нотации Питера Чена.
37. Назовите СУБД, поддерживающую иерархическую модель данных.
38. Назовите основные элементы иерархической модели данных.
39. Граф какого вида лежит в основе сетевой модели данных?
40. Назовите основные элементы сетевой модели данных.
41. Назовите автора реляционной модели данных.
42. Назовите автора фундаментальных трудов в теории реляционных баз данных.
43. Что такое CODASYL?
44. Кого считают автором сетевой модели данных?
45. Что такое IDS и IDMS ?