

Кемеровский институт (филиал) РЭУ им. Г.В. Плеханова  
Экономический факультет  
Кафедра вычислительной техники и информационных  
технологий

---

# БАЗЫ ДАННЫХ

*Лебедева Т.Ф.*  
*2013 г.*

## Цели и задачи дисциплины:

*Цель* – формирование у студентов глубоких теоретических знаний в области управления, хранения и обработки данных, а также практических навыков по проектированию и реализации эффективных систем хранения и обработки данных.

*Задачи:*

- способствовать формированию системного базового представления, первичных знаний, умений и навыков студентов по принципам построения систем управления базами данных как научной и прикладной дисциплины, достаточных для дальнейшего продолжения образования и самообразования их в области вычислительной техники, информационных систем различного назначения;
- способствовать изучению и построению моделей организации данных, проектирование реляционных баз данных для различных предметных областей;
- способствовать получению базового уровня в проектировании и разработке баз данных в системах СУБД MS Access, MS SQL Server;
- способствовать представлению о роли и месте баз данных в автоматизированных системах, о назначении и основных характеристиках различных систем управления базами данных, их функциональных возможностях.

Дисциплина «Базы данных» относится к базовой (общепрофессиональной) части профессионального цикла.

■ Изучению курса предшествуют дисциплины: «Дискретная математика», «Теоретические основы информатики», «Программирование», «Вычислительные системы, сети и телекоммуникации», «Операционные системы, среды и оболочки», «Объектно-ориентированный анализ и программирование»

Требования к входным знаниям, умениям и компетенциям:

*Знать:*

■ – методы теории множеств, математической логики, теории графов, типовые алгоритмы поиска и сортировки данных, основные принципы работы запоминающих устройств, используемых для хранения данных в компьютерах, функции операционных систем по работе с файлами;

*Уметь:*

■ – применять математический аппарат при решении задач, приемы работы с файловыми системами операционных систем.

■ Успешное освоение курса позволяет перейти к изучению дисциплин: «Проектирование информационных систем», «СУБД Oracle для разработки экономических приложений», «Информационная безопасность», Итоговая государственная аттестация.

## основная литература

- ) Карпова, Т.С. Базы данных: модели, разработка, реализация [Текст]: учебник / Т.С. Карпова. – Спб.: Питер, 2002. – 304 с.
- ) Кузин, А.В. Базы данных [Текст]: учебное пособие для вузов /А.В. Кузин, С.В. Левонисова.-М.: Академия, 2010.-316 с.
- ) Лебедева, Т.Ф. Базы данных на основе кейс-метода [текст]: учебное пособие. /Т.Ф. Лебедева, Солопова А.Н. – Кемерово: Кеми РГТЭУ, 2011.-135 с.

## дополнительная литература

- ) Грабер, М. Введение в SQL [Текст] / М. Грабер ; пер. с англ - М.: Лори, 2000. – 382 с.
- ) Дейт, К. Дж. Введение в системы баз данных [текст]. /К. Дж. Дейт, 7-е изд. – М.: Вильямс, 2001. – 1072 с.
- ) Кириллов, В.В. Основы проектирования реляционных баз данных [электронный ресурс]  
<http://www.citforum.ru/database/dbguide/index.shtml>
- ) Кириллов, В.В. Структурированный язык запросов (SQL) [электронный ресурс]  
<[http://www.citforum.ru/database/sql\\_kg/index.shtml](http://www.citforum.ru/database/sql_kg/index.shtml)>.
- ) Кузнецов, С.Д. Введение в стандарты языка SQL [электронный ресурс]  
<http://www.citforum.ru/database/sqlbook/>

- 6) Кузнецов, С.Д. Основы современных баз данных / Информационно-аналитические материалы / Центр Информационных технологий МГУ, [электронный ресурс] <<http://citforum.ru/database/osbd/contents.shtml> >
- 7) Пушников, А.Ю. Введение в системы управления базами данных [электронный ресурс] <<http://www.citforum.ru/database/dblearn/index.shtml>>
- 8) Сергеев А. Access 2007. Новые возможности – Спб.: Питер, 2008. – 176 с.
- 9) Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: учебник для высших учебных заведений / Под ред. Проф. А.Д. Хомоненко. – Издание третье, дополненное и переработанное. – СПб.: КОРОНА принт. 2007. – 736 с.

### **программное обеспечение**

- Пакет программ Microsoft Office,
- Microsoft Access 2007;
- Microsoft SQL Server 2008

### **базы данных, информационно-справочные и поисковые системы**

- Библиотека Максима Мошкова. Научно-популярная литература [Электронный ресурс] // [http://http://libhttp://lib.http://lib.ru](http://libhttp://lib.http://lib.ru).
- ООО Научная образовательная библиотека [Электронный ресурс] // <http://elibrary.ru>
- Электронно-библиотечной системе «Инфра-М» [Электронный ресурс] // <http://student.rsute.ru/library>
- Информационно-коммуникационные технологии в образовании. Система федеральных образовательных порталов // [Электронный ресурс] <http://www.ict.edu.ru/lib/>
- Курсы по базам данных в Интернет-университете// [Электронный ресурс] <http://www.intuit.ru>

# Концепция баз данных

## Данные и ЭВМ

Восприятие реального мира можно соотнести с последовательностью разных, хотя иногда и взаимосвязанных, явлений. С давних времен люди пытались описать эти явления (даже тогда, когда не могли их понять). Такое описание называют *данными*.

Традиционно фиксация данных осуществляется с помощью конкретного средства общения, например, с помощью естественного языка на конкретном носителе (например, бумаге). Обычно данные (факты, явления, события, идеи или предметы) и их интерпретация (семантика) фиксируются совместно, так как естественный язык достаточно гибок для представления того и другого.

Примером может служить утверждение «Стоимость авиабилета 128». Здесь «128» – данное, а «Стоимость авиабилета» – его семантика.

Нередко данные и интерпретация разделены. Например, «Расписание движения самолетов» может быть представлено в виде таблицы (таблица 1.1), в верхней части которой (отдельно от данных) приводится их интерпретация. Такое разделение затрудняет работу с данными (попробуйте быстро получить сведения из нижней части таблицы).

# Концепция баз данных

Таблица 1.1 – Расписание движения самолетов

## Интерпретация

Дни недели	Пункт отправления	Время вылета	Пункт назначения	Время прибытия	Тип самолета	Стоимость билета
Данные						
2_4_7	Баку	21.12	Москва	0.52	ИЛ-86	115.00
3_6	Ереван	7.20	Киев	9.25	ТУ-154	92.00
2_6	Казань	22.40	Баку	23.50	ТУ-134	73.50
1 по 7	Киев	14.10	Москва	16.15	ТУ-154	57.00
2_3_5	Минск	10.50	Сочи	13.06	ИЛ-86	78.50
1_3_6	Москва	15.17	Баку	18.44	ИЛ-86	115.00
1 по 7	Москва	9.05	Киев	11.05	ТУ-154	57.00
1_3_5	Рига	21.53	Таллин	22.57	АН-24	21.50
3_6	Сочи	18.25	Баку	20.12	ТУ-134	44.00
2_4_6	Таллин	6.30	Рига	7.37	АН-24	21.50

# Концепция баз данных

## Данные и ЭВМ

Применение ЭВМ для ведения (*ведение данных* – термин, объединяющий действия по добавлению, удалению или изменению хранимых данных) и обработки данных обычно приводит к еще большему разделению данных и интерпретации.

ЭВМ имеет дело главным образом с данными как таковыми. Большая часть интерпретирующей информации вообще не фиксируется в явной форме (ЭВМ не «знает», является ли «21.50» стоимостью авиабилета или временем вылета).

Существуют по крайней мере две исторические причины, по которым применение ЭВМ привело к отделению данных от интерпретации:

- ЭВМ не обладали достаточными возможностями для обработки текстов на естественном языке – основном языке интерпретации данных;
- стоимость памяти ЭВМ была первоначально весьма велика.



# Концепция баз данных

## Данные и ЭВМ

Память использовалась для хранения самих данных, а интерпретация традиционно возлагалась на пользователя. Пользователь закладывал интерпретацию данных в свою программу, которая «знала», например, что шестое вводимое значение связано со временем прибытия самолета, а четвертое – со временем его вылета. Это существенно повышало роль программы, так как вне интерпретации данные представляют собой не более чем совокупность битов на запоминающем устройстве.

Жесткая зависимость между *данными* и использующими их *программами* создает серьезные проблемы в ведении данных и делает использования их менее гибкими. Нередки случаи, когда пользователи одной и той же ЭВМ создают и используют в своих программах разные наборы данных, содержащие сходную информацию.

Разработчики прикладных программ (написанных, например, на Бейсике, Паскале или С) размещают нужные им данные в файлах, организуя их наиболее удобным для себя образом. При этом одни и те же данные могут иметь в разных приложениях совершенно разную организацию (разную последовательность размещения в записи, разные форматы одних и тех же полей и т.п.).

# Концепция баз данных

## Данные и ЭВМ

Обобществить такие данные чрезвычайно трудно: например, любое изменение структуры записи файла, производимое одним из разработчиков, приводит к необходимости изменения другими разработчиками тех программ, которые используют записи этого файла.

В книге У. Девиса (Операционные системы, М., Мир, 1980) приведен пример:

«Несколько лет назад почтовое ведомство (из лучших побуждений) пришло к решению, что все адреса должны обязательно включать почтовый индекс. Во многих вычислительных центрах это, казалось бы, незначительное изменение привело к ужасным последствиям. Добавление к адресу нового поля, содержащего шесть символов, означало необходимость внесения изменений в каждую программу, использующую данные этой задачи в соответствии с изменившейся суммарной длиной полей. Тот факт, что какой-то программе для выполнения ее функций не требуется знания почтового индекса, во внимание не принимался: если в некоторой программе содержалось обращение к новой, более длинной записи, то в такую программу вносились изменения, обеспечивающие дополнительное место в памяти».

# Концепция баз данных

## Данные и ЭВМ

Активная деятельность по отысканию приемлемых способов обобществления непрерывно растущего объема информации привела к созданию в начале 60-х годов специальных программных комплексов, называемых «Системы управления базами данных» (СУБД). Основная особенность СУБД – это наличие процедур для ввода и хранения не только самих данных, но и описаний их структуры. Файлы, снабженные описанием хранимых в них данных и находящиеся под управлением СУБД, стали называть банками данных, а затем базами данных (БД).

Пусть, например, требуется хранить расписание движения самолетов (таблица. 1.1) и ряд других данных, связанных с организацией работы аэропорта (БД «Аэропорт»). Представив, что мы используем для этого «русифицированную» СУБД, можно подготовить следующее описание расписания:

*СОЗДАТЬ ТАБЛИЦУ Расписание*

*(Номер\_рейса            Целое    Дни\_недели            Текст (8)  
Пункт\_отправления    Текст (24)    Время\_вылета            Время  
Пункт\_назначения    Текст (24)    Время\_прибытия        Время  
Тип\_самолета        Текст (8)    Стоимость\_билета        Валюта);*

*и ввести его вместе с данными в БД «Аэропорт».*

# Концепция баз данных

## Данные и ЭВМ

Язык запросов СУБД позволяет обращаться за выборкой данных как из программ, так и с терминалов). Сформировав запрос

*ВЫБРАТЬ Номер\_рейса, Дни\_недели, Время\_вылета ИЗ ТАБЛИЦЫ  
Расписание*

*ГДЕ Пункт\_отправления = 'Москва' И Пункт\_назначения = 'Киев' И  
Время\_вылета > 17;*

получим расписание «Москва-Киев» на вечернее время, а по запросу  
*ВЫБРАТЬ КОЛИЧЕСТВО (Номер\_рейса) ИЗ ТАБЛИЦЫ Расписание ГДЕ  
Пункт\_отправления = 'Москва' И Пункт\_назначения = 'Минск';*  
получим количество рейсов «Москва-Минск».

Эти запросы не потеряют актуальности и при расширении таблицы:  
*ДОБАВИТЬ В ТАБЛИЦУ Расписание Длительность\_полета Целое;*  
как это было с программами обработки почтовых адресов при введении почтового индекса.

Однако за все надо расплачиваться: на обмен данными через СУБД требуется большее время, чем на обмен аналогичными данными прямо из файлов, специально созданных для того или иного приложения.

## Поколения СУБД и направления исследований в области БД

Принято выделять три поколения СУБД:

**I. Поколение.** Сетевые и иерархические системы БД, широко распространенные в 70-е годы, получили название - системы БД первого поколения. Это были первые системы, предлагавшие развитую функциональность СУБД в рамках единой системы, с языками определения и манипулирования данными для набора записей. Назовем некоторые наиболее общие характеристики ранних систем:

- Эти системы активно использовались в течение многих лет, дольше, чем используется какая-либо из реляционных СУБД.
- Все ранние системы не основывались на каких-либо абстрактных моделях. Понятие модели данных фактически вошло в обиход специалистов в области БД только вместе с реляционным подходом.
- В ранних системах доступ к БД производился на уровне записей. Пользователи этих систем осуществляли явную навигацию в БД, на основе языков программирования, расширенных функциями СУБД.
- Доступ к данным на уровне записей заставляли пользователя самого производить всю оптимизацию доступа к БД, без поддержки системы.
- После появления реляционных систем большинство ранних систем было оснащено «реляционными» интерфейсами.

Поколения СУБД и направления исследований в области БД

**II. Поколение.** В 80-е годы системы первого поколения были существенно потеснены современным семейством реляционных СУБД, называемых системами БД второго поколения. Типичные представители многопользовательских профессиональных систем второго поколения – DB2, INGRES, ORACLE, Informix и др.

В нашей стране представление о реляционных СУБД у большинства программистов сложилось на основе опыта использования систем на платформе персональных компьютеров, таких как dBASE, FoxBASE, FoxPro, Paradox, Clipper, Clarion, а позже Access.

Очень часто персональные СУБД использовались (да и сейчас кое-где используются) для автоматизации таких задач, например, в финансовой сфере, которые требуют многопользовательских профессиональных систем.

Реляционные СУБД и сейчас являются наиболее популярными в сфере разработки бизнес-приложений.

Однако существует широкий класс приложений, для которых технология реляционных систем БД не является вполне удовлетворительной:

- технология программирования;
- системы, основанные на знаниях, и
- мультимедийные системы;
- системы автоматизации проектирования (САПР);
- геоинформационные системы (ГИС);
- издательские системы;
- системы дистанционного обучения;
- системы электронной коммерции и др.

Это прежде всего связано с примитивностью структур данных, лежащих в основе реляционной модели данных. Плоские нормализованные отношения универсальны и теоретически достаточны для представления данных любой предметной области. Однако в нетрадиционных приложениях в базе данных появляются сотни, если не тысячи таблиц, над которыми постоянно выполняются дорогостоящие операции соединения, необходимые для воссоздания сложных структур данных, присущих предметной области.

## Поколения СУБД и направления исследований в области БД

Другим серьезным ограничением реляционных систем являются их относительно слабые возможности по части представления семантики приложения.

Осознавая эти ограничения и недостатки реляционных систем, исследователи в области баз данных выполняют многочисленные проекты, основанные на идеях, выходящих за пределы реляционной модели данных.

**III. Поколение.** Термин «системы следующего (или третьего) поколения» вошел в жизнь после опубликования группой известных специалистов в области БД «Манифеста систем баз данных третьего поколения».

В целом можно сказать, что СУБД следующего поколения - это прямые наследники реляционных систем. В число требований к СУБД третьего поколения входят:

- полнота системы типов, поддерживаемых в СУБД;
- поддержка иерархии и наследования типов;
- возможность управления сложными объектами и т.д.



## Поколения СУБД и направления исследований в области БД

Основными направлениями систем третьего поколения, обладающими некоторыми разными характеристиками, являются:

- *Ориентация на расширенную реляционную модель.*

Основные характеристики:

- ✓ максимальное следование (насколько это возможно с учетом новых требований) известным принципам организации СУБД;
- ✓ абстрактные типы данных;
- ✓ поддержка исторической информации и темпоральных запросов; отказ от требований нормализации.

Одной из наиболее известных СУБД этого направления является система Postgres. В Postgres реализованы многие интересные средства: поддерживается темпоральная модель хранения и доступа к данным; допускается хранение в полях отношений данных абстрактных, определяемых пользователями типов.

## Поколения СУБД и направления исследований в области БД

- *Генерация систем баз данных, ориентированных на приложения.*  
Основная характеристика:

- ✓ создание собственно не системы, а генератора систем, наиболее полно соответствующих потребностям приложений.

Решение достигается путем создания наборов модулей со стандартизованными интерфейсами. Существуют два экспериментальных прототипа «генерационных» систем - Genesis и Exodus. Обе эти системы основаны прежде всего на принципах модульности и точного соблюдения установленных интерфейсов. По сути дела, системы состоят из минимального ядра (развитой файловой системы в случае Exodus) и технологического механизма программирования дополнительных модулей. В проекте Exodus этот механизм основывается на системе программирования E, которая является простым расширением C++, поддерживающим стабильное хранение данных во внешней памяти. Вместо готовой СУБД предоставляется набор "полуфабрикатов" с согласованными интерфейсами, из которых можно сгенерировать систему, максимально отвечающую потребностям приложения

## Поколения СУБД и направления исследований в области БД

- *Системы баз данных, основанные на правилах.*

К этому направлению относятся дедуктивные БД.

Основная характеристика:

- ✓ достижение расширяемости системы и ее адаптации к нуждам конкретных приложений путем использования стандартного механизма управления правилами.

По сути дела, система представляет собой некоторый интерпретатор системы правил и набор модулей-действий, вызываемых в соответствии с этими правилами. Можно изменять наборы правил (существует специальный язык задания правил) или изменять действия, подставляя другие модули с тем же интерфейсом. По определению, дедуктивная БД состоит из двух частей: экстенциональной, содержащей факты, и интенциональной, содержащей правила для логического вывода новых фактов на основе экстенциональной части и запроса пользователя. Отметим лишь, что обычно языки запросов и определения интенциональной части БД являются логическими (поэтому дедуктивные БД часто называют логическими). Имеется прямая связь дедуктивных БД с базами знаний

## Поколения СУБД и направления исследований в области БД

- *Объектно-ориентированные СУБД.*

Направление объектно-ориентированных баз данных (ООБД) возникло сравнительно давно. Публикации появлялись уже в середине 1980-х. Однако наиболее активно это направление развивается в последние годы. В настоящее время ведется очень много экспериментальных и производственных работ в области объектно-ориентированных СУБД. На рынке представлены — система GemStone компании Servio, ONTOS компании Ontos, ObjectStore компании Object Design, O2, ORION, Iris.

Кроме того, системы управления реляционными базами данных, разработанные компаниями Oracle, Microsoft, Borland, Informix, включали объектно-ориентированные средства. Многие из этих продуктов появились еще во второй половине 80-х годов, и сегодня, по прошествии времени разработки они все еще не вступили в пору зрелости; в этом — одна из причин того, что по сей день мировой рынок реальных приложений не торопится принимать системы ООБД.

## Поколения СУБД и направления исследований в области БД

Широкое использование Интернет в различных сферах деятельности ставит перед разработчиками СУБД следующие проблемы:

- *Интеграция текста, данных, кода и потоков.* В области СУБД основное внимание всегда уделялось организации, хранению, анализу и выборке структурированных данных. Развитие Web-приложений продемонстрировало важность более сложных типов данных: текстов, изображений, темпоральных, аудио- и видеоданных.
- *Интеграция информации.* Типичным подходом к интеграции информации в масштабах предприятия является построение хранилищ (DataWarehouse) витрин (data mart) данных на основе извлечения операционных данных, их трансформации к единой схеме и загрузке данных в хранилище (процедура *ETL - extraction, transformation, loading*). Этот подход пригоден для использования на предприятии с несколькими десятками операционных баз данных, находящихся под единым контролем.

## Поколения СУБД и направления исследований в области БД

В Интернет требуется производить интеграцию информации между несколькими предприятиями. Как правило, организации не позволяют в массовом порядке извлекать данные из своих операционных баз данных, к ним можно будет адресовать лишь одиночные запросы. В результате потребуются интеграция, возможно, миллионов информационных источников «на лету». В связи с этим существует множество нерешенных проблем: семантическая неоднородность; неполнота и неточность данных; ограниченность доступа к конфиденциальным данным и т.д.

- *Мультимедийные запросы.* Очевидно, что объем мультимедийных данных (изображения, аудио, видео и т.д.) значительно возрастает. Проблемой сообщества баз данных является создание простых способов анализа, обобщения, поиска и обозрения электронных подборок мультимедийной информации, относящейся к некоторому объекту.

## Терминология в СУБД

В общепромышленных руководящих материалах по созданию банков данных Государственного комитета по науке и технике, изданных в 1982 г., приводятся следующие определения основных понятий:

- **База данных (БД)** - именованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области (ПО).
- **Предметная область (ПО)** – часть реального мира, подлежащая автоматизации с целью организации управления. Она представлена множеством фрагментов, каждый из которых характеризуется объектами, процессами и множеством пользователей.
- **Банк данных (БнД)** – это система специальным образом организованных данных – баз данных, программных, технических, языковых, организационно-методических средств, предназначенных для обеспечения централизованного накопления и коллективного многоцелевого использования данных.
- **Системы управления базами данных (СУБД)** – совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

## Терминология в СУБД

- Программы, с помощью которых пользователи работают с БД, называются **приложениями**.

### СУБД должна обеспечивать:

- возможность представления внутренней структуры данных;
- физическую и логическую независимость данных;
- минимальную избыточность данных;
- возможность быстрого поиска;
- эффективные языки запросов к данным;
- требования безопасности, надежности, конфиденциальности, целостности:
  - данные должны быть защищены от искажения, хищения, разрушения;
  - данные должны быть восстанавливаемыми;
  - данные должны быть контролируемы;
  - должна быть установлена процедура идентификации пользователей;
  - должна быть организована система санкционированного доступа;
  - должен быть установлен контроль за действиями пользователя с целью обнаружения ошибочных операций.



## Терминология в СУБД

СУБД должна предоставлять доступ к данным любым пользователям, включая и тех, которые практически не имеют и (или) не хотят иметь представления о:

- физическом размещении в памяти данных и их описаний;
- механизмах поиска запрашиваемых данных;
- проблемах, возникающих при одновременном запросе одних и тех же данных многими пользователями (прикладными программами);
- способах обеспечения защиты данных от некорректных обновлений и (или) несанкционированного доступа;
- поддержании БД в актуальном состоянии и множестве других функций СУБД.

## Терминология в СУБД

### Вопросы для самоконтроля к главе 1

- 1) Что понимается под ведением данных?
- 2) Можно ли использовать термины «база данных» и «банк данных» как эквивалентные?
- 3) Какие функции по отношению к пользователю выполняет СУБД?
- 4) Что включают требования надежности и безопасности БД?
- 5) Чем характеризуются БД первого поколения?

## Классификация моделей данных

При выполнении основных функций СУБД должна использовать различные описания данных. Естественно, что проект базы данных надо начинать с анализа предметной области и выявления требований к ней отдельных пользователей (сотрудников организации, для которых создается база данных). Проектирование обычно поручается человеку (группе лиц) – *администратору базы данных (АБД)*. Им может быть как специально выделенный сотрудник организации, так и будущий пользователь БД, достаточно хорошо знакомый с машинной обработкой данных.

Объединяя частные представления о содержимом БД, полученные в результате опроса пользователей, и свои представления о данных, которые могут потребоваться в будущих приложениях, АБД сначала создает обобщенное неформальное описание создаваемой базы данных. Это описание, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающим над проектированием базы данных, называют *инфологической моделью данных* (рис. 2.1).

# Модели данных

## Трехуровневая архитектура



## Классификация моделей данных

Такая человеко-ориентированная модель полностью независима от физических параметров среды хранения данных. Поэтому инфологическая модель не должна изменяться до тех пор, пока какие-то изменения в реальном мире не потребуют изменения в ней некоторого определения, чтобы эта модель продолжала отражать предметную область. Остальные модели, показанные на рис. 2.1, являются компьютеро-ориентированными. С их помощью СУБД дает возможность программам и пользователям осуществлять доступ к хранимым данным лишь по их именам, не заботясь о физическом расположении этих данных. Нужные данные отыскиваются СУБД на внешних запоминающих устройствах по *физической* модели данных.

Так как указанный доступ осуществляется с помощью конкретной СУБД, то модели должны быть описаны на *языке описания данных* этой СУБД. Такое описание, создаваемое АБД по инфологической модели данных, называют *даталогической* или *концептуальной* моделью данных.

## Классификация моделей данных

**Модель данных** это совокупность структур данных, взаимосвязей и операций их обработки.

Трехуровневая архитектура (инфологический, даталогический и физический уровни) позволяет обеспечить *независимость хранимых данных* от использующих их программ. АБД может при необходимости переписать хранимые данные на другие носители информации и (или) реорганизовать их физическую структуру, изменив лишь физическую модель данных. АБД может подключить к системе любое число новых пользователей (новых приложений), дополнив, если надо, даталогическую модель. Указанные изменения физической и даталогической моделей не будут замечены существующими пользователями системы (окажутся «прозрачными» для них), так же как не будут замечены и новые пользователи. Следовательно, независимость данных обеспечивает возможность развития системы баз данных без разрушения существующих приложений.



Рис. 2.2 Классификация моделей данных

## Классификация моделей данных

*Инфологические* модели данных используются на ранних стадиях проектирования для описания структур данных в процессе разработки приложения, а дата-логические 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. Для каждого класса документов определяется свой набор правил, описывающих грамматику соответствующего языка разметки. 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.



## Классификация моделей данных

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. 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.

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.

## Классификация моделей данных

*Тезаурусные модели* основаны на принципе организации словарей, содержат определенные языковые конструкции и принципы их взаимодействия в заданной грамматике. Эти модели используются в системах-переводчиках, особенно многоязыковых переводчиках.

*Дескрипторные модели* — самые простые из документальных моделей, и широко использовались на ранних стадиях использования документальных БД. В этих моделях каждому документу соответствовал дескриптор — описатель. Дескриптор имел жесткую структуру и описывал документ в соответствии с теми характеристиками, которые требуются для работы с документами в разрабатываемой документальной БД. Например, для БД, содержащей описание патентов, дескриптор содержал название области, к которой относился патент, номер патента, дату выдачи патента и еще ряд ключевых параметров, которые заполнялись для каждого патента. Обработка информации в таких БД велась исключительно по дескрипторам,, а не по самому тексту патента.