

# Современные информационные технологии

---

Биологический институт  
Томский государственный  
университет

---

Лекция 4

**Базы данных**

**Дмитрий Владимирович**

**Курбатский**

**научный сотрудник НИЛБиЭМ Би ТГУ,**

**магистр биологии**

- Зоологический музей (к. 123)
- Кафедра зоологии позвоночных (к. 126)



# Студенту на заметку

- Сайт <http://citforum.ru/> - множество околокомпьютерных статей по всем аспектам, в одном месте и без рекламы.
- Сайт <http://sql.ru> – про базы данных и работу с ними.

# Блок 1

Общие понятия и классификация  
БД

# База данных (БД)

- организованная в соответствии с определёнными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.
- совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных.
- некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия.
- совместно используемый набор логически связанных данных (и описание этих данных), предназначенный для удовлетворения информационных потребностей организации.

и даже

- представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ) (Гражданский кодекс РФ, ст. 1260).

# Вариант необходимых свойств

- БД хранится и обрабатывается в вычислительной системе.
- Данные в БД логически структурированы (систематизированы).
- БД включает схему, или метаданные, описывающие логическую структуру БД в формальном виде (в соответствии с некоторой метамоделью).

# Модель данных

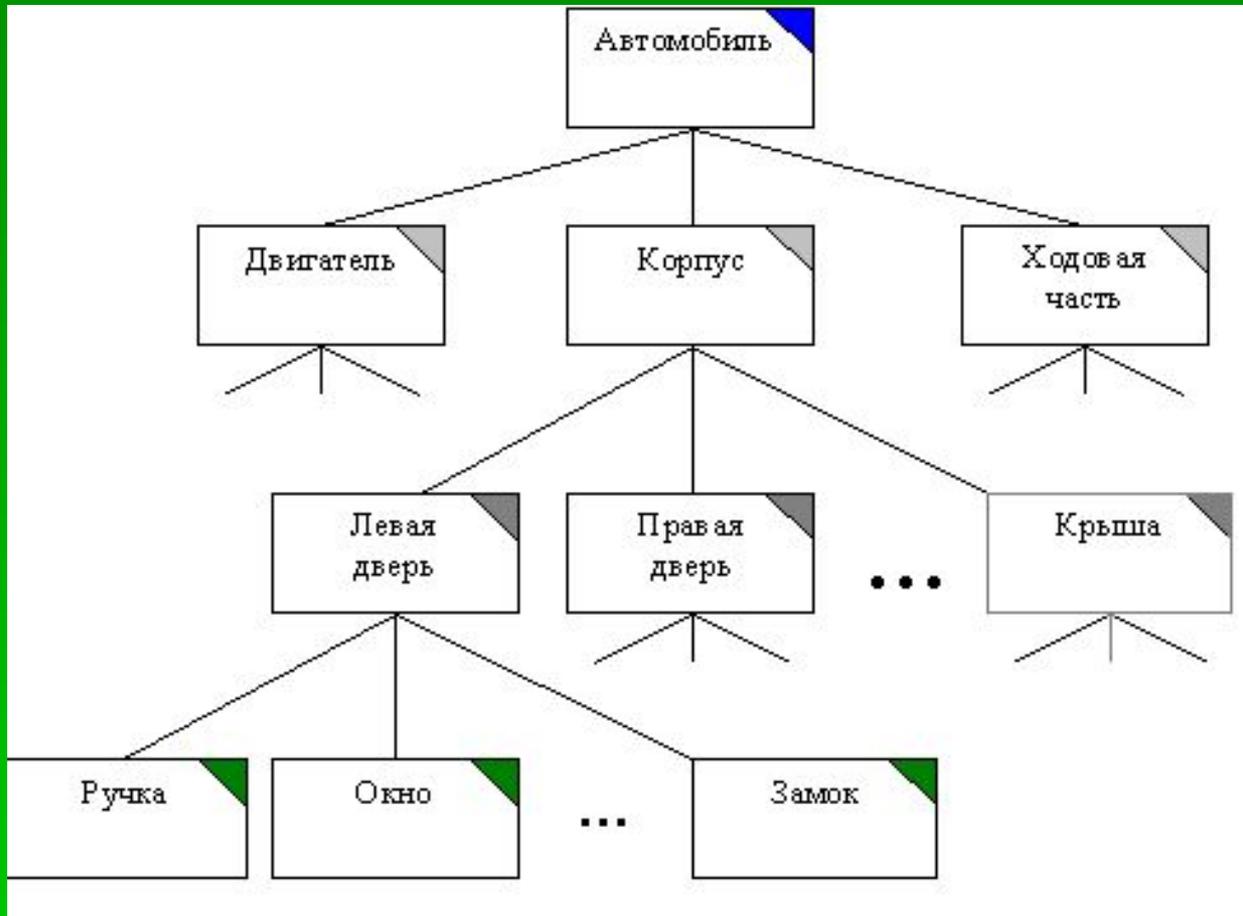
1. аспект структуры
  2. аспект манипуляции
  3. аспект целостности
- аспект защиты
  - аспект практической реализации и применения

# Классификация БД

## ...по модели данных

- Иерархическая
- Сетевая
- Реляционная
- Объектная и объектно-ориентированная
- Объектно-реляционная
- Функциональная
- и др.

# Иерархическая БД

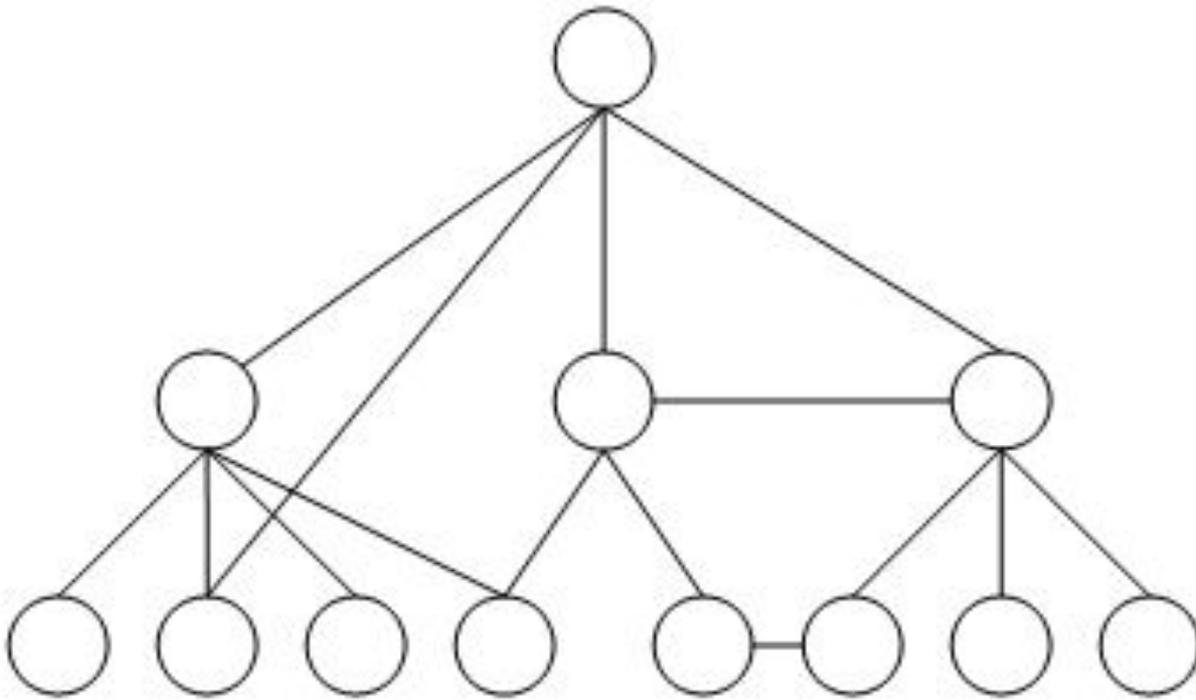


- Information Management System (IMS)
- Time-Shared Date Management System (TDMS)
- Mark IV Multi - Access Retrieval System
- System 2000
- InterSystems Caché
- Google App Engine Datastore API

в операционных системах:

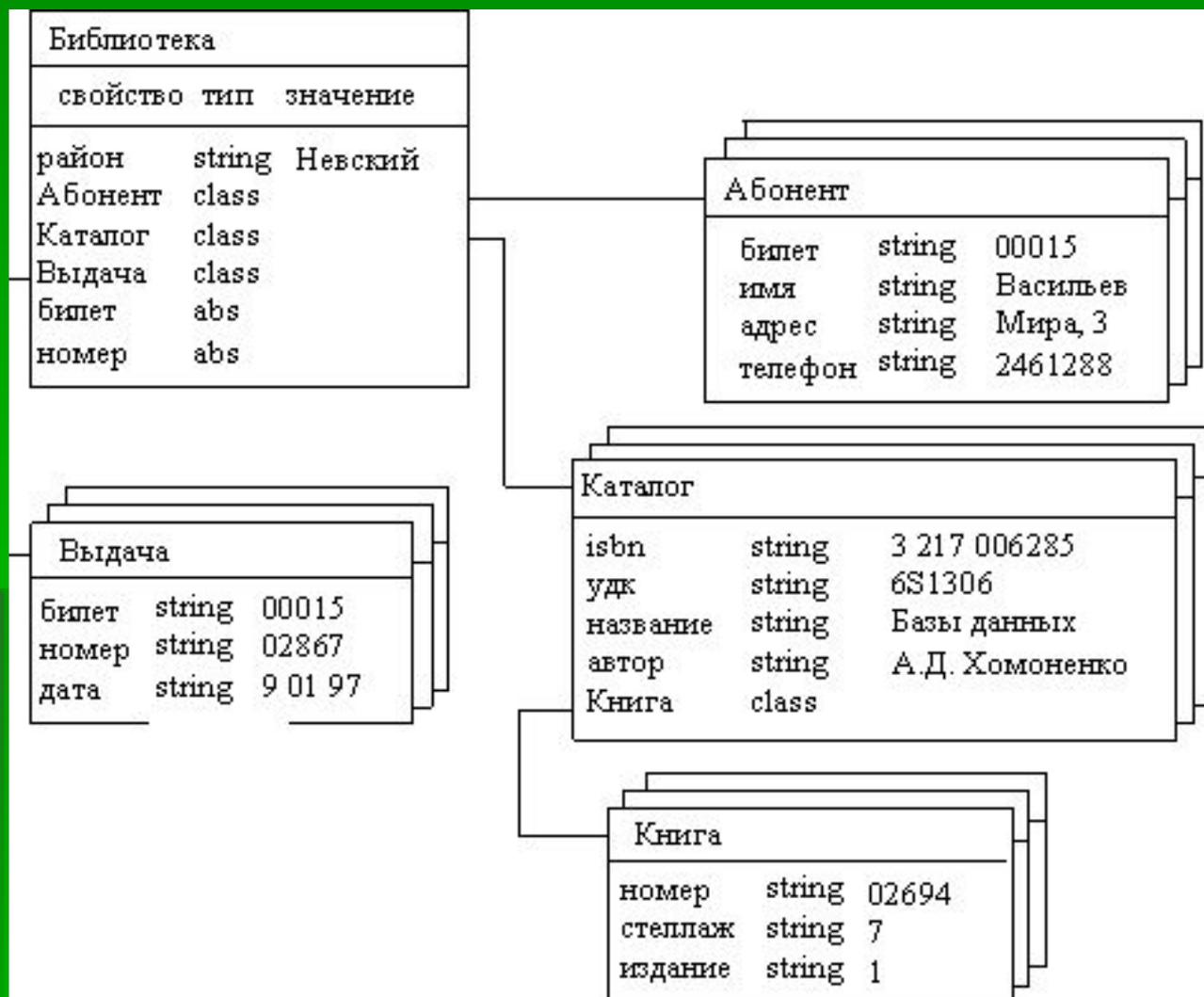
- Серверы каталогов LDAP и Active Directory
- иерархические файловые системы
- Реестр Windows.

# Сетевая модель данных



- СООБЗ  
Cerebrum
- ИСУБД  
CronosPRO
- dbVista
- Caché
- GT.M

# Объектная и объектно-реляционные БД



- Jasmine
- Versant
- POET

## ■■■ по среде постоянного хранения

- Во вторичной памяти, или традиционная (conventional database)
  - средой постоянного хранения является периферийная энергонезависимая память
  - как правило жёсткий диск
  - в оперативной памяти лишь кеш и данные для текущей обработки
- В оперативной памяти (in-memory database, memory-resident database, main memory database)
  - все данные на стадии исполнения находятся в оперативной памяти.
- В третичной памяти (tertiary database)
  - средой постоянного хранения является отсоединяемое от сервера устройство массового хранения
  - как правило на основе магнитных лент (в прошлом) или оптических дисков
  - во вторичной памяти сервера хранится лишь каталог данных третичной памяти, файловый кеш и данные для текущей обработки; загрузка же самих данных требует специальной процедуры

## ■ ■ ■ по содержанию

- Географическая
- Историческая
- Научная
- Мультимедийная
- и др.

в частности,

- каталоги музеев и коллекций
  - Зоомузей ТГУ
  - ИСиЭЖ СО РАН

## ■ ■ ■ По степени распределённости

- Централизованная
  - или сосредоточенная (centralized database)
  - БД, полностью поддерживаемая на одном компьютере.
- Распределённая
  - (distributed database)
  - БД, составные части которой размещаются в различных узлах компьютерной сети в соответствии с каким-либо критерием.
  - Бывает:
    - Неоднородная (heterogeneous distributed database): фрагменты распределённой БД в разных узлах сети поддерживаются средствами более одной СУБД
    - Однородная (homogeneous distributed database): фрагменты распределённой БД в разных узлах сети поддерживаются средствами одной и той же СУБД.
    - Фрагментированная, или секционированная (partitioned database): методом распределения данных является фрагментирование (партиционирование, секционирование), вертикальное или горизонтальное.
    - Тиражированная (replicated database): методом распределения данных является тиражирование (репликация).

# ■ ■ ■ ПО ДИНАМИКЕ ДАННЫХ

- Пространственная
  - spatial database
  - БД, в которой поддерживаются пространственные свойства сущностей предметной области.
  - в ГИС
- Временная, или темпоральная
  - temporal database
  - БД, в которой поддерживается какой-либо аспект времени, не считая времени, определяемого пользователем.
- Пространственно-временная
  - spatial-temporal database
  - БД, в которой одновременно поддерживается одно или более измерений в аспектах как пространства, так и времени.
- Циклическая
  - round-robin database
  - БД, объём хранимых данных которой не меняется со временем, поскольку в процессе сохранения данных одни и те же записи используются циклически.

## ■ ■ ■ по размеру

- маленькие
- побольше
- ещё больше
- большие (терабайты)
- сверхбольшие (петабайты)

# СУБД

- Система управления базами данных
- *database management system (DBMS)*
- – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

# Функции СУБД

- управление данными во внешней памяти (на дисках);
- управление данными в оперативной памяти с использованием дискового кэша;
- журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- поддержка языков БД (язык определения данных, язык манипулирования данными).

# Компоненты СУБД

- ядро
  - отвечает за управление данными во внешней и оперативной памяти, за журнализацию
- процессор языка базы данных
  - обеспечивает оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода
- подсистема поддержки времени исполнения (runtime)
  - интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД
- сервисные программы (внешние утилиты)
  - обеспечивают ряд дополнительных возможностей по обслуживанию информационной системы

# Классификации СУБД

- По модели данных
  - Иерархические
  - Сетевые
  - Реляционные
  - Объектно-ориентированные
  - Объектно-реляционные
- По степени распределённости
  - Локальные СУБД
  - Распределённые СУБД
- По способу доступа к БД
  - Файл-серверные
  - Клиент-серверные
  - Встраиваемые

# Связанные понятия

- Информационная система
- База знаний
- Хранилище данных

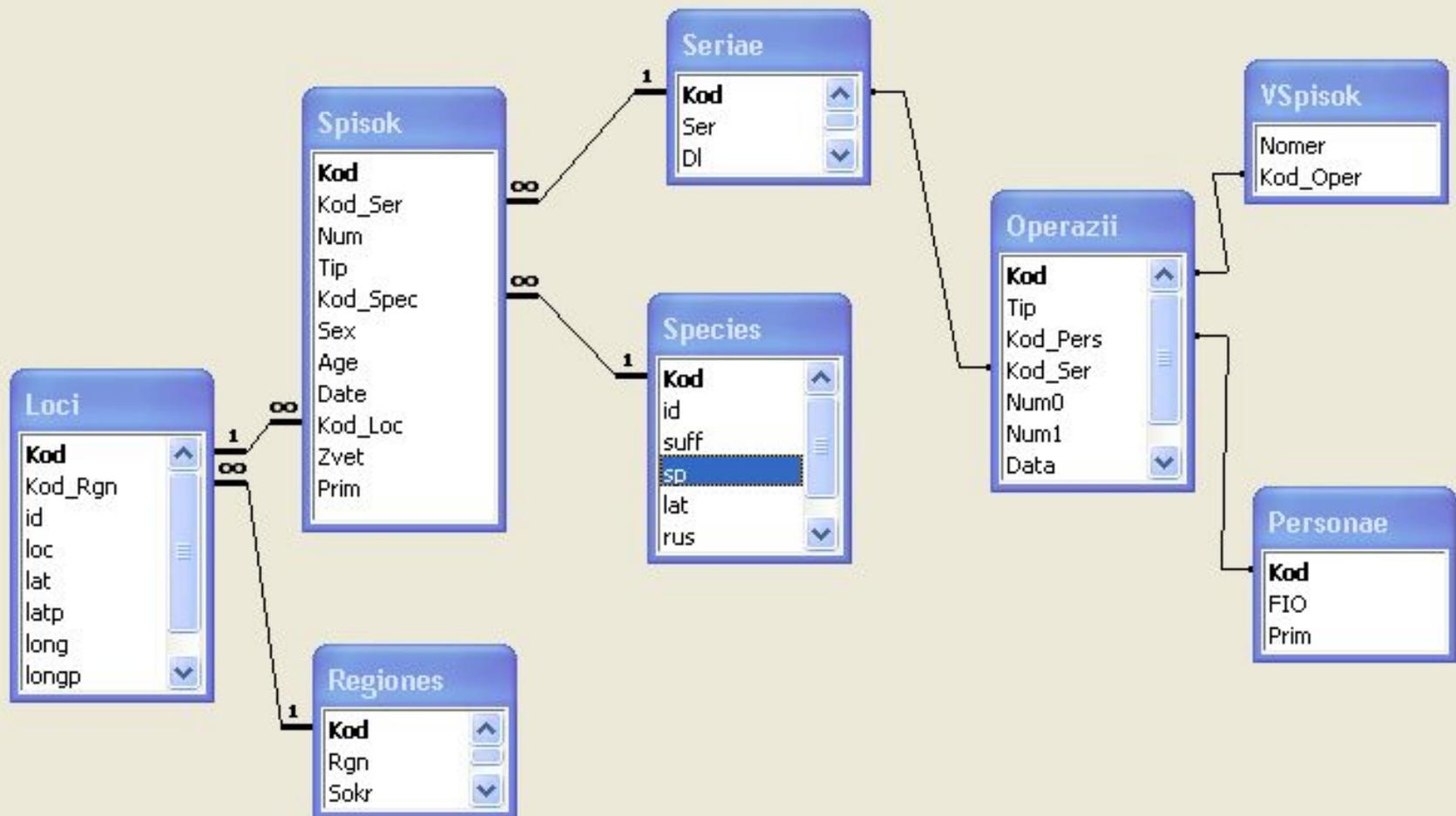
# Блок 2

Реляционные БД

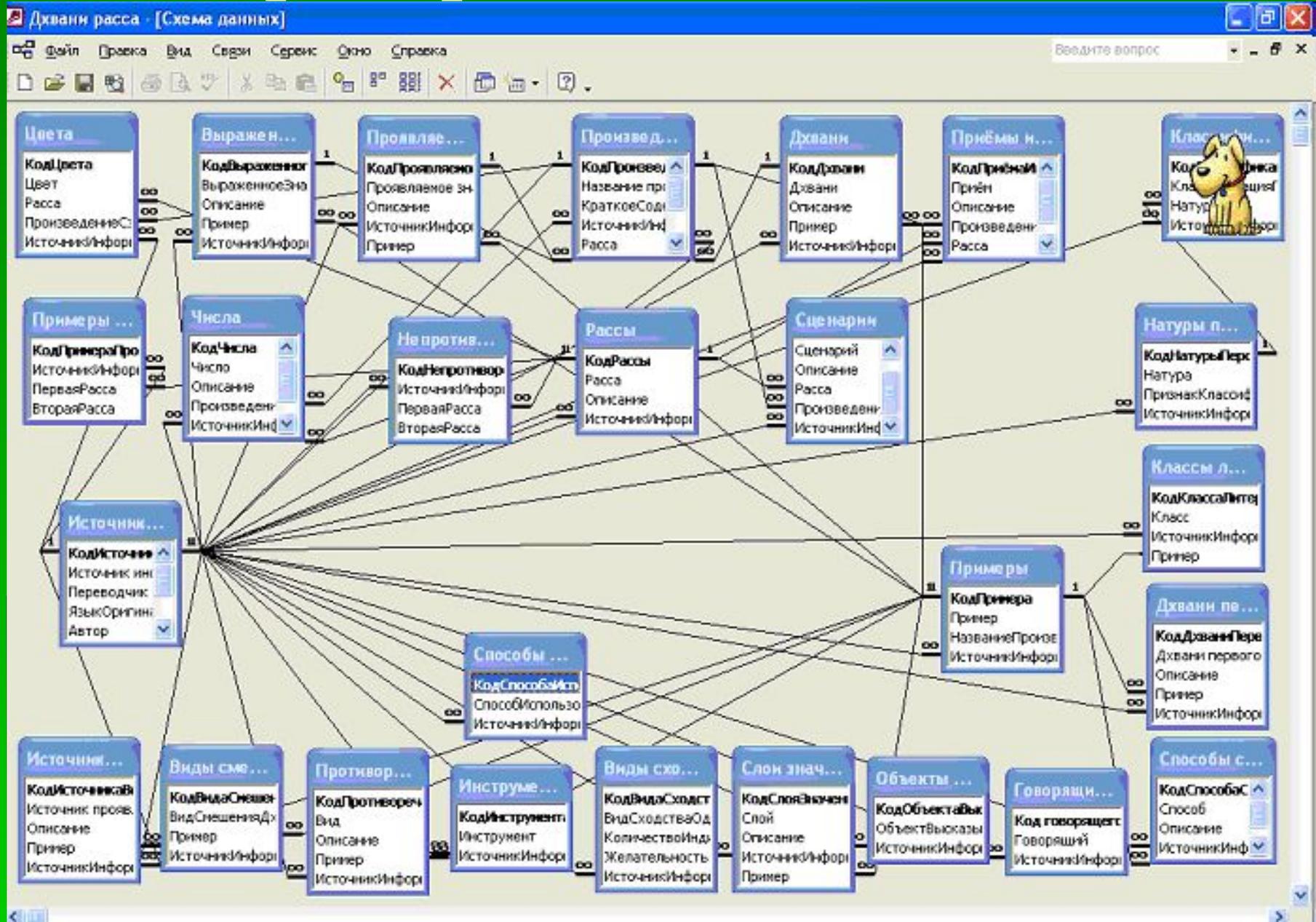
Компоненты БД, СУБД

Некоторые СУБД

# Реляционные БД



# Ещё пример



# Аспекты реляционных БД

---

1. структура данных (объекты данных)
2. целостность данных
3. обработка данных (операторы)

# Свойства реляционных БД

1. В таблице нет двух одинаковых строк.
2. Таблица имеет столбцы, соответствующие атрибутам отношения.
3. Каждый атрибут в отношении имеет уникальное имя.
4. Порядок строк в таблице произвольный.

# 12 правил Кодда

правило 0: Основное правило (Foundation Rule): Реляционная СУБД должна быть способна полностью управлять базой данных, используя связи между данными:

- Чтобы быть реляционной системой управления базами данных (СУБД), система должна использовать исключительно свои реляционные возможности для управления базой данных.

правило 1: Явное представление данных (The Information Rule):

- Информация должна быть представлена в виде данных, хранящихся в ячейках. Данные, хранящиеся в ячейках, должны быть атомарны. Порядок строк в реляционной таблице не должен влиять на смысл данных.

правило 2: Гарантированный доступ к данным (Guaranteed Access Rule):

- Доступ к данным должен быть свободен от двусмысленности. К каждому элементу данных должен быть гарантирован доступ с помощью комбинации имени таблицы, первичного ключа строки и имени столбца.

правило 3: Полная обработка неизвестных значений (Systematic Treatment of Null Values):

- Неизвестные значения NULL, отличные от любого известного значения, должны поддерживаться для всех типов данных при выполнении любых операций. Например, для числовых данных неизвестные значения не должны рассматриваться как нули, а для символьных данных — как пустые строки.

# 12 правил Кодда

правило 4: Доступ к словарю данных в терминах реляционной модели (Active On-Line Catalog Based on the Relational Model):

- Словарь данных должен сохраняться в форме реляционных таблиц, и СУБД должна поддерживать доступ к нему при помощи стандартных языковых средств, тех же самых, которые используются для работы с реляционными таблицами, содержащими пользовательские данные.

правило 5: Полнота подмножества языка (Comprehensive Data Sublanguage Rule):

- Система управления реляционными базами данных должна поддерживать хотя бы один реляционный язык, который
  - (а) имеет линейный синтаксис,
  - (б) может использоваться как интерактивно, так и в прикладных программах,
  - (в) поддерживает операции определения данных, определения представлений, манипулирования данными (интерактивные и программные), ограничители целостности, управления доступом и операции управления транзакциями (begin, commit и rollback).

правило 6: Возможность модификации представлений (View Updating Rule):

- Каждое представление должно поддерживать все операции манипулирования данными, которые поддерживают реляционные таблицы: операции выборки, вставки, модификации и удаления данных.

правило 7: Наличие высокоуровневых операций управления данными (High-Level Insert, Update, and Delete):

- Операции вставки, модификации и удаления данных должны поддерживаться не только по отношению к одной строке реляционной таблицы, но по отношению к любому множеству строк.

# 12 правил Кодда

правило 8: Физическая независимость данных (Physical Data Independence):

- Приложения не должны зависеть от используемых способов хранения данных на носителях, от аппаратного обеспечения компьютеров, на которых находится реляционная база данных.

правило 9: Логическая независимость данных (Logical Data Independence):

- Представление данных в приложении не должно зависеть от структуры реляционных таблиц. Если в процессе нормализации одна реляционная таблица разделяется на две, представление должно обеспечить объединение этих данных, чтобы изменение структуры реляционных таблиц не сказывалось на работе приложений.

правило 10: Независимость контроля целостности (Integrity Independence):

- Вся информация, необходимая для поддержания целостности, должна находиться в словаре данных. Язык для работы с данными должен выполнять проверку входных данных и автоматически поддерживать целостность данных.

правило 11: Дистрибутивная независимость (Distribution Independence):

- База данных может быть распределённой, может находиться на нескольких компьютерах, и это не должно оказывать влияние на приложения. Перенос базы данных на другой компьютер не должен оказывать влияния на приложения.

правило 12: Согласование языковых уровней (The Nonsubversion Rule):

- Если используется низкоуровневый язык доступа к данным, он не должен игнорировать правила безопасности и правила целостности, которые поддерживаются языком более высокого уровня.

# Связанные термины и понятия

- Отношение
- Заголовок
- Атрибут
- Кортеж
- Домен
- Атом
- Декомпозиция
- Нормализация
- Ключ
- Индекс
- Реляционная алгебра

# Пример таблицы данных

Species : таблица							
	Kod	Код	Доп.	Sp.	Латынь	Русское	
+	67	66		<input type="checkbox"/>	<i>Branta canadensis</i>	Канадская казарка	
+	68	67		<input type="checkbox"/>	<i>Branta leucopsis</i>	Белошекая казарка	
+	69	68		<input type="checkbox"/>	<i>Branta bernicla</i>	Черная казарка	
+	70	69		<input type="checkbox"/>	<i>Branta nigricans</i>	Американская казарка	
+	71	70		<input type="checkbox"/>	<i>Rufibrenta ruficollis</i>	Краснозобая казарка	
▶	+	72	71	<input type="checkbox"/>	<i>Anser anser</i>	Серый гусь	
+	73	72		<input type="checkbox"/>	<i>Anser albifrons</i>	Белолобый гусь	
+	74	73		<input type="checkbox"/>	<i>Anser erythropus</i>	Пискулька	
+	75	74		<input type="checkbox"/>	<i>Anser fabalis</i>	Гуменник	
+	76	75		<input type="checkbox"/>	<i>Chen caerulescens</i>	Белый гусь	
+	77	75 a		<input type="checkbox"/>	<i>Chen rossii</i>	Гусь Росса	
+	78	76		<input type="checkbox"/>	<i>Philacte canagica</i>	Белошей	
+	79	77		<input type="checkbox"/>	<i>Eulabeia indica</i>	Горный гусь	
+	80	78		<input type="checkbox"/>	<i>Cygnopsis cygnoides</i>	Сухонос	
+	81	79		<input type="checkbox"/>	<i>Cygnus olor</i>	Лебедь-шипун	
+	82	80		<input type="checkbox"/>	<i>Cygnus cygnus</i>	Лебедь-кликун	
+	83	81		<input type="checkbox"/>	<i>Cygnus bewickii</i>	Малый лебедь	
+	84	82		<input type="checkbox"/>	<i>Cygnus columbianus</i>	Американский лебедь	
+	85	83		<input type="checkbox"/>	<i>Tadorna ferruginea</i>	Огарь	
+	86	84		<input type="checkbox"/>	<i>Tadorna tadorna</i>	Пеганка	
+	87	85		<input type="checkbox"/>	<i>Tadorna cristata</i>	Хохлатая пеганка	
+	88	86		<input type="checkbox"/>	<i>Anas platyrhynchos</i>	Кряква	
+	89	87		<input type="checkbox"/>	<i>Anas poecilorhyncha</i>	Черная кряква	
+	90	88		<input type="checkbox"/>	<i>Anas crecca</i>	Чирок-свиистунок	
+	91	88 a		<input type="checkbox"/>	<i>Anas carolinensis</i>	Зеленокрылый чирок	
+	92	89		<input type="checkbox"/>	<i>Anas formosa</i>	Клоктун	
+	93	90		<input type="checkbox"/>	<i>Anas falcata</i>	Касатка	
+	94	91		<input type="checkbox"/>	<i>Anas strepera</i>	Серая утка	
+	95	92		<input type="checkbox"/>	<i>Anas penelope</i>	Свизь	

Запись: 72 из 831

# Нормализация

- Первая нормальная форма (1NF)
- Вторая нормальная форма (2NF)
- Третья нормальная форма (3NF)
- Нормальная форма Бойса — Кодда (BCNF)
- Четвёртая нормальная форма (4NF)
- Пятая нормальная форма (5NF)
- Доменно-ключевая нормальная форма (DKNF)
- Шестая нормальная форма (6NF)



# Студенту на заметку

- Первые 3 формы нормализации в примерах:

<http://club.shelek.ru/viewart.php?id=177>

# Первая нормальная форма

Наим.	Город	Адрес	Эл. почта	Вид	Конт. лица
Поршнево й з-д	Владими р	Ул. 2-я Кольцевая, 17	info@plunger.ru	Поставщ ик	Иванов И.И., зам. дир., тел (3254)76-15-95 Петров П.П., нач. отд. сбыта, тел (3254)76-15-35
ООО Вымпел	Курск	Ул. Гоголя, 25	rennon@mail.r u	Клиент	Сидоров С.С., директор, тел. (7634)66-65-38
ИЧП Альфа	Владими р	Ул. Пушкинская, 37, оф. 565	alpha323@list.r u	Клиент	Васильев В.В., директор, тел (3254)74-57-45

# Первая нормальная форма

Наим.	Город	Адрес	Эл. почта	Вид
Поршневой з-д	Владимир	Ул. 2-я Кольцевая, 17	info@plunger.ru	Поставщик
Поршневой з-д	Владимир	Ул. 2-я Кольцевая, 17	info@plunger.ru	Поставщик
ООО Вымпел	Курск	Ул. Гоголя, 25	pennon@mail.ru	Клиент
ИЧП Альфа	Владимир	Ул. Пушкинская, 37, оф. 565	alpha323@list.ru	Клиент



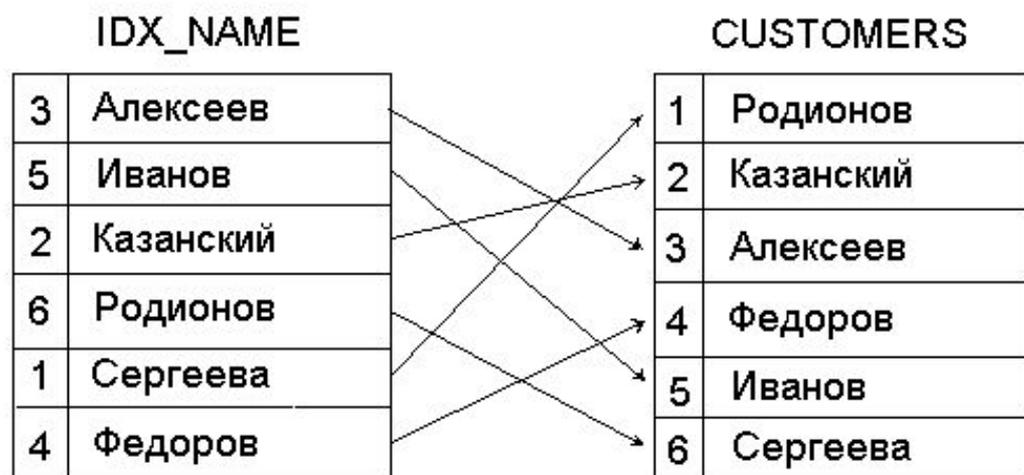
Должность	Ф.И.О.	Код города	Тел.
зам. дир.	Иванов И. И.	3254	76-15-9 5
нач. отд. сбыта	Петров П. П.	3254	76-15-3 5
директор	Сидоров С.С.	7634	66-65-3 8
директор	Васильев В.В.	3254	74-57-4 5



# Отношения **relations**

- один к одному **one-to-one**
- один ко многим **one-to-many**
- многие ко многим **many-to-many**

# Индексирование



- Карточка: рег. №№ : KRegN (407)
- Карточка: русское : KRTaxon (149)
- Карточка: хранение : KFM (10)
- Коллекторы : Coll (128)
- Наборщики : KSSz (4)
- Определители : Det (17)
- По возрасту : Age (6)
- По полу : Sex (3)
- Пол/возр. доп. : SA\_Dop (1)
- Препараторы : Praer (22)
- Примечания сверщика : Prim (16)
- Прочее : Etc (14)
- Регистрационные №№ : Reg\_N (407)
- Сохранность : Sohran (4)
- Таксоны: Виды : Taxon (124)
- Таксоны: Виды (рус.) : Taxon\_R (140)
- Таксоны: Классы : Classis (5)
- Таксоны: Отряды : Ordo (29)
- Таксоны: Перечни охраняемых видов : OSpisok...
- Таксоны: Семейства : Familia (44)
- Таксоны: Типы : Typus (2)
- Типы коллекций : Koll\_Tip (1)
- Типы хранения : Hran\_Tip (4)
- Удалённые записи : KDel (1)
  - False (163)
- Формы хранения : Forma (11)
  - влажный препарат (22)
  - мумия (3)
  - сверить (47)
  - скелет (13)
  - сухой препарат, имаго (57)
  - тушка (113)
  - часть скелета (1)
  - череп (29)
  - чучело (125)
  - шкура без черепа (7)

# Ключи

- По типу
  - первичный
  - вторичные
  - внешний
- По строению
  - простой
  - составной
- По происхождению
  - естественный
  - суррогатный

# Пример таблицы данных

Species : таблица							
	Kod	Код	Доп.	Sp.	Латынь	Русское	
+	67	66		<input type="checkbox"/>	<i>Branta canadensis</i>	Канадская казарка	
+	68	67		<input type="checkbox"/>	<i>Branta leucopsis</i>	Белошекая казарка	
+	69	68		<input type="checkbox"/>	<i>Branta bernicla</i>	Черная казарка	
+	70	69		<input type="checkbox"/>	<i>Branta nigricans</i>	Американская казарка	
+	71	70		<input type="checkbox"/>	<i>Rufibrenta ruficollis</i>	Краснозобая казарка	
▶	+	72	71	<input type="checkbox"/>	<i>Anser anser</i>	Серый гусь	
+	73	72		<input type="checkbox"/>	<i>Anser albifrons</i>	Белолобый гусь	
+	74	73		<input type="checkbox"/>	<i>Anser erythropus</i>	Пискулька	
+	75	74		<input type="checkbox"/>	<i>Anser fabalis</i>	Гуменник	
+	76	75		<input type="checkbox"/>	<i>Chen caerulescens</i>	Белый гусь	
+	77	75	a	<input type="checkbox"/>	<i>Chen rossii</i>	Гусь Росса	
+	78	76		<input type="checkbox"/>	<i>Philacte canagica</i>	Белошей	
+	79	77		<input type="checkbox"/>	<i>Eulabeia indica</i>	Горный гусь	
+	80	78		<input type="checkbox"/>	<i>Cygnopsis cygnoides</i>	Сухонос	
+	81	79		<input type="checkbox"/>	<i>Cygnus olor</i>	Лебедь-шипун	
+	82	80		<input type="checkbox"/>	<i>Cygnus cygnus</i>	Лебедь-кликун	
+	83	81		<input type="checkbox"/>	<i>Cygnus bewickii</i>	Малый лебедь	
+	84	82		<input type="checkbox"/>	<i>Cygnus columbianus</i>	Американский лебедь	
+	85	83		<input type="checkbox"/>	<i>Tadorna ferruginea</i>	Огарь	
+	86	84		<input type="checkbox"/>	<i>Tadorna tadorna</i>	Пеганка	
+	87	85		<input type="checkbox"/>	<i>Tadorna cristata</i>	Хохлатая пеганка	
+	88	86		<input type="checkbox"/>	<i>Anas platyrhynchos</i>	Кряква	
+	89	87		<input type="checkbox"/>	<i>Anas poecilorhyncha</i>	Черная кряква	
+	90	88		<input type="checkbox"/>	<i>Anas crecca</i>	Чирок-свиистунок	
+	91	88	a	<input type="checkbox"/>	<i>Anas carolinensis</i>	Зеленокрылый чирок	
+	92	89		<input type="checkbox"/>	<i>Anas formosa</i>	Клоктун	
+	93	90		<input type="checkbox"/>	<i>Anas falcata</i>	Касатка	
+	94	91		<input type="checkbox"/>	<i>Anas strepera</i>	Серая утка	
+	95	92		<input type="checkbox"/>	<i>Anas penelope</i>	Свизь	

Запись: 72 из 831

# Блок 3

Компоненты СУБД

БД и программирование

Язык SQL

# Компоненты СУБД

- Отношение (таблица)
- Представление
- Снимок
- Хранимая процедура
- Триггер
- Курсор
- Индекс
- Запрос

# Дополнительные механизмы СУБД

- Транзакция
- Журнализация
- Секционирование
- Репликация

# Некоторые СУБД

- Старые
  - Clipper
  - FoxPro
  - IBM DB2
  - dBase
  - Paradox
- От Micro\$oft
  - Visual FoxPro
  - Microsoft Access
  - MS SQL Server
- Встраиваемые
  - BerkeleyDB
  - Firebird Embedded
  - OpenEdge
  - Microsoft SQL Server Compact
  - SQLite
- Бесплатные
  - MySQL
  - PostgreSQL
- Прочие
  - Caché
  - Firebird
  - Informix
  - Interbase
  - Oracle
  - Sybase Adaptive Server Enterprise
  - ЛИНТЕР

# Типы данных БД

## (на примере PostgreSQL)

- Численные типы
  - Целые
  - С фиксированной точкой
  - С плавающей точкой
  - Денежный тип (отличается специальным форматом вывода, а в остальном аналогичен числам с фиксированной точкой с двумя знаками после запятой)
- Символьные типы произвольной длины
- Двоичные типы (включая BLOB)
- Типы «дата/время» (полностью поддерживающие различные форматы, точность, форматы вывода, включая последние изменения в часовых поясах)
- Булев тип
- Перечисление
- Геометрические примитивы
- Сетевые типы
  - IP и IPv6-адреса
  - CIDR-формат
  - MAC-адрес
- UUID-идентификатор
- XML-данные
- Массивы
- Идентификаторы объектов БД
- Псевдотипы

# Ещё о данных

- Тип Variant
- Пустые и нулевые значения:
  - NULL
  - 0
  - пустая строка ""
- Операторы
  - =, ==, ===
  - Like
  - BETWEEN

# Язык запросов

- – искусственный язык, на котором делаются запросы к базам данных и другим информационным системам, особенно к информационно-поисковым системам.
- Примеры:
  - SQL
  - Language Integrated Query — расширение для некоторых языков программирования в .NET Framework
  - XQuery
  - XPath
  - поисковые запросы

# Связанные понятия

- Регулярные выражения
- Релевантность

# SQL

- Structured Query Language
- язык структурированных запросов



# Студенту на заметку

- 3 варианта динамики данных
  - создание
  - удаление
  - модификация
- а также отбор и выдача результатов
- и обмен *между*

# Возможности SQL

- Изменение структуры данных
  - таблицы
  - поля
  - связи, индексы и пр.
- Изменение данных
  - создание (добавление)
  - модификация
  - удаление
- Поиск и выборка данных
  - расчёты
  - транспонирование
- Управление пользователями и доступом к БД
- Управление работой СУБД
  - транзакции

# Состав **SQL**

- операторы
  - инструкции
  - вычисляемые функции
- 
- таблицы
  - поля
  - связи и объединения

# Операторы SQL

- операторы определения данных (Data Definition Language, DDL)
  - CREATE создает объект БД
  - ALTER изменяет объект
  - DROP удаляет объект
- операторы манипуляции данными (Data Manipulation Language, DML)
  - SELECT считывает данные, удовлетворяющие заданным условиям
  - INSERT добавляет новые данные
  - UPDATE изменяет существующие данные
  - DELETE удаляет данные
- операторы определения доступа к данным (Data Control Language, DCL)
  - GRANT предоставляет пользователю (группе) разрешения на определенные операции с объектом
  - REVOKE отзывает ранее выданные разрешения
  - DENY задает запрет, имеющий приоритет над разрешением
- операторы управления транзакциями (Transaction Control Language, TCL)
  - COMMIT применяет транзакцию.
  - ROLLBACK откатывает все изменения, сделанные в контексте текущей транзакции.
  - SAVEPOINT делит транзакцию на более мелкие участки.

# Структура запроса **SQL** на выборку данных

---

SELECT [ALL | DISTINCT] возвращаемый\_столбец, ... | \*

FROM спецификатор\_таблицы, ...

WHERE условие\_поиска

GROUP BY имя\_столбца, ...

HAVING условие\_поиска

ORDER BY спецификатор\_сортировки, ...

# Объединения таблиц

- JOIN
  - INNER
  - OUTER (Left, Right)
  - FULL
- UNION

# Примеры запросов **SQL**

- `SELECT StName FROM Students WHERE GrNo IN (1, 3, 5, 10);`
- `SELECT * FROM Students WHERE StName LIKE 'Натал_я';`
- и др.

# Преимущества и недостатки SQL

## ■ Преимущества

- Независимость от конкретной СУБД
- Наличие стандартов
- Декларативность

## ■ Недостатки

- Несоответствие реляционной модели данных
  - Повторяющиеся строки
  - Неопределённые значения (nulls)
  - Явное указание порядка колонок слева направо
  - Колонки без имени и дублирующиеся имена колонок
  - Отсутствие поддержки свойства «=»
  - Использование указателей
  - Высокая избыточность
- Сложность
- Отступления от стандартов
- Сложность работы с иерархическими структурами

# Связанные понятия

- SQL-SQL-инъекции

# Взаимодействие БД и ОС

---

- ODBC
  - Open Database Connectivity
- DAO
  - Data Access Objects
- ADO
  - ActiveX Data Objects
- ADOdb
- BDE
  - Borland Database Engine



# Студенту на заметку

## ■ API

- интерфейс программирования приложений
- *или* интерфейс прикладного программирования
- *application programming interface*
- набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах

# Объекты ADO

- Connection (представляет подключение к удалённому источнику данных)
- Recordset (представляет набор строк, полученный от источника данных)
- Command (используется для выполнения команд и SQL-запросов с параметрами)
- Record (может представлять одну запись объекта Recordset или же иерархическую структуру, состоящую из текстовых данных)
- Stream (используется для чтения и записи потоковых данных, например, документов XML или двоичных объектов)
- Errors (представляет ошибки)
- Fields (представляет столбцы таблицы базы данных)
- Parameters (представляет набор параметров SQL-инструкции)
- Properties (представляет набор свойств объекта)