

**Базы данных.  
Системы управления  
базами данных**

# Базы данных

---

**Базы данных** - важнейшая составная часть ***информационных систем***. Информационные системы предназначены для хранения и обработки больших объемов информации. Изначально такие системы существовали в письменном виде. Для этого использовались различные картотеки, папки, журналы, библиотечные каталоги и т.д. Любая информационная система должна выполнять три основные функции: *ввод данных, запросы по данным, составление отчетов*.

**База данных (БД)** – *поименованная совокупность структурированных данных о конкретных объектах некоторой предметной области*.

Иначе, **база данных** – реализованная с помощью компьютера информационная структура (модель), отражающая состояние объектов и их отношения.

Синоним – **банк данных**.

**Объект, атрибут, связь** – фундаментальные понятия ИС

# Системы управления базами данных

**Система управления базами данных (СУБД)** – комплекс программных и языковых средств для создания баз данных, поддержки их в актуальном состоянии и организации поиска в них необходимой информации.

## Этапы обобщённой технологии работы с СУБД:

1. Создание структуры таблиц базы данных.
2. Ввод и редактирование данных в таблицах:
  - с помощью таблиц;
  - с помощью экранных форм.
3. Обработка данных в таблицах (с помощью запросов):
  - запрос-выборка (для отбора данных);
  - запрос-изменение (для изменения, удаления, добавления).
4. Вывод информации из базы данных.

# Классификация БД

---

- **По характеру хранимой информации – фактографические и документальные.**  
Фактографические – это картотеки, в которых сведения представлены в строго определенном формате; документальные содержат информацию самого различного типа (текст., звук., граф., мультимедийного) – это архивы.
- **По способу хранения данных – централизованные** (на одном ПК) и **распределенные** (на разных ПК в локальных и глобальных комп. сетях).
- **По структуре организации данных – реляционные (табличные), иерархические и сетевые БД.**

# Типы моделей баз данных

---

- **Реляционные** (от латинского *relation* – отношение) состоят из одной и более взаимосвязанных двумерных таблиц.  
Их свойства:
  - записи могут иметь одинаковые значения в некоторых полях, кроме ключевого;
  - каждая запись имеет уникальное значение в ключевом поле;
  - несколько таблиц связаны ключами.
- **Иерархические** состоят из совокупности элементов, расположенных в порядке подчинения.  
Их свойства:
  - они имеют уровни, узлы, связи (образуют «дерево»);
  - несколько узлов уровня связаны только с одним узлом более высокого уровня;
  - иерархическое «дерево» имеет только одну вершину;
  - каждый уровень имеет своё имя.
- **Сетевые**  
Их свойства:
  - имеют уровни, узлы, связи;
  - связи между уровнями свободные, нет строгого подчинения.

# Функции СУБД:

---

- описание данных,
- первичный ввод,
- удаление данных,
- корректировка данных,
- упорядочение (сортировка данных),
- поиск информации (с помощью языка запросов),
- защита информации и разграничение доступа,
- резервное сохранение и восстановление БД,
- поддержка интерфейса с пользователем.

# Популярные СУБД

---

- **Ребус**
- **DBASE III**
- **FoxPro**
- **Access for Windows**
- **Paradox**
- **Lotus**
- **Oracle**

# Пример фактографической реляционной БД «Мои друзья»

---

Фамилия	Имя	Дата рождения	Телефон
---------	-----	---------------	---------

Простейшая РБД содержит одну таблицу, более сложная состоит из множества взаимосвязанных таблиц.



# Представление информации в реляционной БД

---

	Поле 1	Поле 2	...	Поле K-1	Поле K
Запись 1			...		
Запись 2			...		
	...		...		
Запись N			...		

Основной элемент БД – **запись** – строка таблицы, содержит информацию об одном объекте.

**Поле** – столбец таблицы, содержит информацию об одном атрибуте объекта.

# Проектирование баз данных

---

Проектирование базы данных выполняется, как правило, коллективом разработчиков и включает следующие работы:

- анализ предметной области;
- проектирование и непосредственно кодирование (создание запросов и приложений);
- тестирование и сопровождение.

# Требования к современным базам данных

---

- адекватность базы данных предметной области;
- интегрированность данных;
- независимость данных;
- минимальная избыточность хранимых данных;
- целостность базы данных;
- обеспечение защиты от несанкционированного доступа или случайного уничтожения данных;
- гибкость и адаптивность структуры базы данных;
- динамичность данных и способность к расширению;
- возможность поиска по многим ключам.

# Коллектив специалистов, обслуживающих большие базы данных

---

**Администратор** - специалист, имеющий представление об информационных потребностях конечных пользователей и отвечающий за определение, загрузку, защиту и эффективность базы данных.

**Аналитики**, обладая знаниями закономерностей соответствующей предметной области, в контакте с конечными пользователями строят формальные (математические) модели для задач конечного пользователя, которые являются исходным представлением задачи для прикладного программиста.

**Прикладные программисты** на основе представления задачи, полученного аналитиками, разрабатывают прикладные программы для решения задач конечных пользователей.

**Системные программисты** обеспечивают работоспособность операционной системы, систем программирования и СУБД, разрабатывают сервисные программы.

# Характеристики БД:

---

- Полнота
- Правильная организация
- Актуальность
- Удобство для использования

Последствия неправильного проектирования

Проектирование баз данных осуществляется на двух уровнях – *физическом* и *логическом*. На физическом уровне решаются вопросы размещения данных на внешних носителях. Во многом эта работа выполняется СУБД автоматически без участия разработчика. На логическом уровне составляется общий список полей, который может насчитывать от единиц до тысяч. Описывают каждое поле по типу данных. Общий список полей разбивается на основные таблицы. Дальнейшее рассмотрение информационной структуры приводит к разбиению – **нормализации** – основных таблиц на более мелкие с целью избежания многократно повторяющихся данных в записях, что уменьшает объем памяти, занимаемый базой данных на диске, и обеспечивает непротиворечивость данных в БД.

Теория реляционных БД разработана **Коддом (1970-е гг.)**.

Эдгар Франк Кодд — британский учёный, работы которого заложили основы теории реляционных баз данных. Работая в компании IBM, он создал реляционную модель данных.

Процесс нормализации имеет итерационный (пошаговый) характер, осуществляется методом нормальных форм. Суть метода состоит в последовательном переводе таблицы из одной нормальной формы в другую, причем каждая последующая устраняет определенный вид функциональной зависимости между полями таблицы. Всего в теории описаны шесть нормальных форм, на практике чаще всего применяются первые три.

**Первая нормальная форма.** Отношение называется приведенным к первой нормальной форме, если все его атрибуты неделимы. Например, отношение, содержащее поле ФИО, не приведено к первой нормальной форме, если в запросах БД требуется выделить отдельно фамилию или имя. Разработчики БД изначально строят так исходное отношение, чтобы оно было в первой нормальной форме.

**Вторая нормальная форма.** Для приведения отношений ко второй нормальной форме введем понятие функциональной зависимости. Функциональная зависимость полей – это зависимость, при которой в строке определенному значению ключевого поля соответствует только одно значение не ключевого поля. Функционально не ключевое поле зависит от составного ключа, но не зависит от любого поля, входящего в составной ключ.

**Третья нормальная форма.** Третья нормальная форма позволяет устранить *транзитивную зависимость*. Транзитивная зависимость существует в отношении, если существуют два описательных поля, в которых первое зависит от ключа, а второе зависит от первого. Отношение находится в третьей нормальной форме, если оно находится во второй нормальной форме, и каждое не ключевое поле не транзитивно зависит от ключа.



# Пример нормализации БД:

**ПОЛИКЛИНИКА** (ФАМИЛИЯ, ДАТА\_РОЖДЕНИЯ,  
УЧАСТОК, ВРАЧ, ДАТА ПОСЕЩЕНИЯ, ДИАГНОЗ)

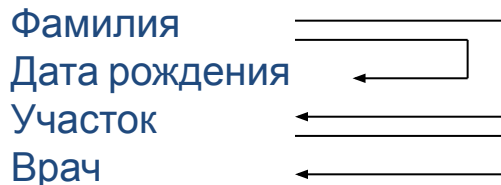
- **первая нормальная форма**. Ключ записей подчеркнут. Согласно определению 2-й нормальной формы все неключевые поля должны функционально зависеть от полного ключа. Это ДИАГНОЗ. Таблицу разбиваем на две:

**ПОСЕЩЕНИЯ**(ФАМИЛИЯ, ДАТА ПОСЕЩЕНИЯ, ДИАГНОЗ)

**ПАЦИЕНТЫ**(ФАМИЛИЯ, ДАТА\_РОЖДЕНИЯ, УЧАСТОК, ВРАЧ)

- **вторая нормальная форма**

Присутствует транзитивная зависимость:



**ПОСЕЩЕНИЯ**(ФАМИЛИЯ, ДАТА ПОСЕЩЕНИЯ, ДИАГНОЗ)

**ПАЦИЕНТЫ**(ФАМИЛИЯ, ДАТА\_РОЖДЕНИЯ, УЧАСТОК)

**ВРАЧИ**(УЧАСТОК, ВРАЧ)

- **третья нормальная форма**

**A6 (2013).** В фрагменте базы данных представлены сведения о родственных отношениях. Определите на основании приведенных данных фамилию и инициалы бабушки Ивановой А.И.

ID	Фамилия_И.О.	Пол
71	Иванов Т.М.	М
85	Петренко И.Т.	М
13	Черных И.А.	Ж
42	Петренко А.И.	Ж
23	Иванова А.И.	Ж
96	Петренко Н.Н.	Ж
82	Черных А.Н.	М
95	Цейс Т.Н.	Ж
10	Цейс Н.А.	М
	...	

ID_Родителя	ID_Ребенка
23	71
13	23
85	23
82	13
95	13
85	42
82	10
95	10
...	...

- 1) Петренко А.И.    2) Черных И.А.    3) Цейс Т.Н.    4) Петренко Н.Н.

## **A16 (1999).** Представлена база данных «Волшебные страны»

<b>Страна</b>	<b>Население</b>	<b>Площадь</b>
НАРНИЯ	148	46,9
ОЗ	155	95,3
ШВАМБРАНИЯ	132	53,5
ЛУКОМОРЬЕ	199	47,7
ЗАЗЕРКАЛЬЕ	211	76,2

После проведения сортировки сведения о НАРНИИ переместились на 1 строку вниз. Сортировка производилась:

- 1) В порядке возрастания по полю СТРАНА
- 2) В порядке убывания по полю ПЛОЩАДЬ
- 3) В порядке возрастания по полю ПЛОЩАДЬ
- 4) В порядке убывания по полю СТРАНА
- 5) В порядке возрастания по полю НАСЕЛЕНИЕ

**A23 (2000).** Представлена база данных «Расписание уроков»:

№	День	N_урока	Кабинет	Предмет	Преп	Класс
1	пн	2	32	матем	Голубева	9а
2	пн	2	21	физика	Иванова	10а
3	вт	4	25	литер	Зайцев	8б
4	вт	3	25	литер	Зайцев	8а
5	чт	4	31	физика	Зайцева	10б
6	пт	3	32	матем	Голубева	8а
7	чт	2	41	химия	Панина	9а
8	пн	1	28	матем	Петров	10а
9	вт	1	41	химия	Панина	10б

Запросу, содержащему выражение **N\_урока > 2** **и** **класс > '8а'** удовлетворяют только записи

1) 1, 2, 3, 4, 5, 6, 7    2) 3, 4, 5, 6    3) 3    4) 5    5) 3, 4, 5