

Лекция 1. Часть 2.

**Базы данных: основные
понятия**

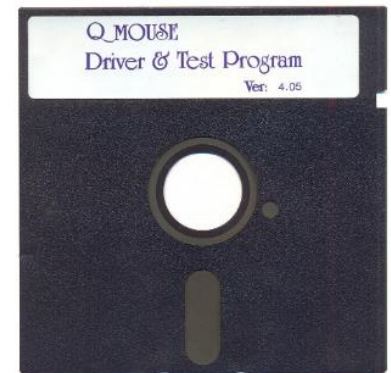
Базы данных: основные понятия

1. История возникновения баз данных.
2. Файловая система как прообраз баз данных и СУБД.

Кратенькая история



- 3600 до н.э. - глиняные таблички у древних шумеров;
- 1966 г. - появление MUMPS;
- 1968 г. - первая промышленная СУБД система IMS фирмы IBM;
- 1972 г. - компания IBM начала исследовательский проект по разработке РСУБД;
- 1979 г. - выход первой РСУБД Oracle;
- 1980 г. - появление dBase II;
- 1981 г. - появление IBM PC;
- 1982 г. - выход DB2 фирмы IBM;
- 1993 г. - выход Intel Pentium, Seagate Medalist 425xe 428.1MB HDD;
- 1994 г. - выход Oracle для PC;
- 1995 г. - появление MySQL;
- 1996 г. - появление PostgreSQL;
- 2005 г. - появление Hadoop;
- 2009 г. - появление MongoDB.



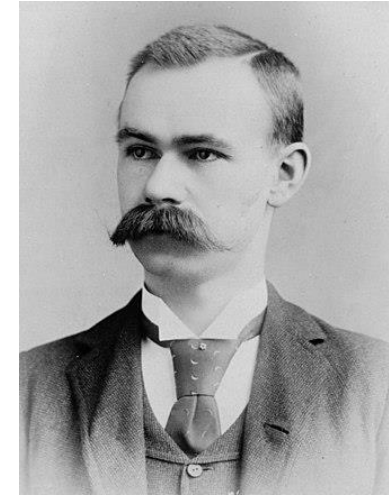
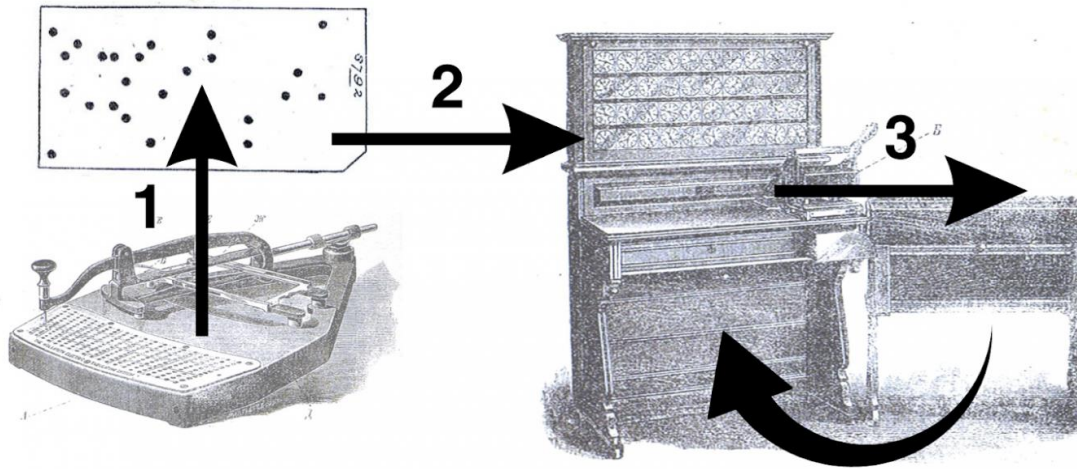
Хранение и обработка данных до компьютеров

Данные до компьютеров



Данные до компьютеров

В 1880-х годах создал **табулятор** – электромеханическую машину, которая могла считывать и сортировать статистические записи, закодированные на перфокартах.



Герман Холлерит
(1860-1929),
американский инженер и
изобретатель

Использовались для обработки результатов переписи населения США (1890, 1900), Канады, Австрии.

Данные до компьютеров

1896 – Холлерит создал компанию
TMC (Tabular Machine Company)

1911 – продал компанию, она вошла в
CTR (Computing-Tabulating-Recording
Company)

1924 – CTR переименована в IBM
(International Business Machines
Corporation)

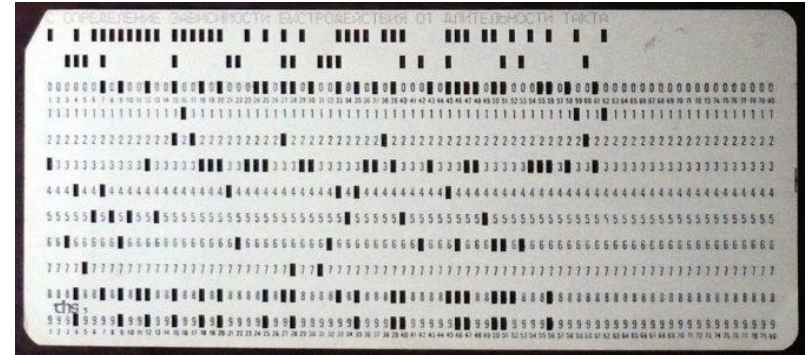


Герман Холлерит за табулятором. 1894.
Колумбийский университет.

Перфокарты и табуляторы IBM



Табуляторы IBM в Администрации социального обеспечения США (SSA). Балтимор, 1936 год



80 байт

Появление компьютеров (ЭВМ, электронная вычислительная машина)

Механизация вычислений



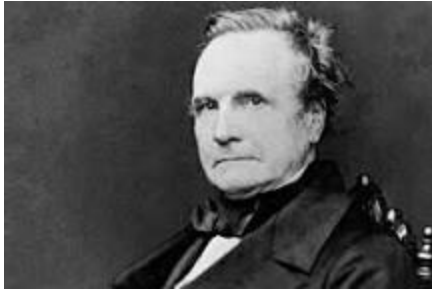
Суммирующая машина Паскаля (1652 год)



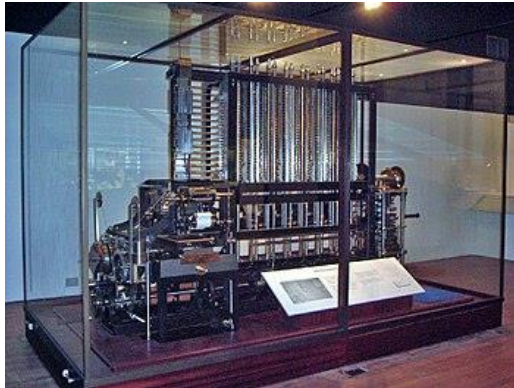
Арифмометр Лейбница (1673 год)

Выполняется только одно действие,
промежуточные результаты записывались на
бумаге

Автоматизация вычислений

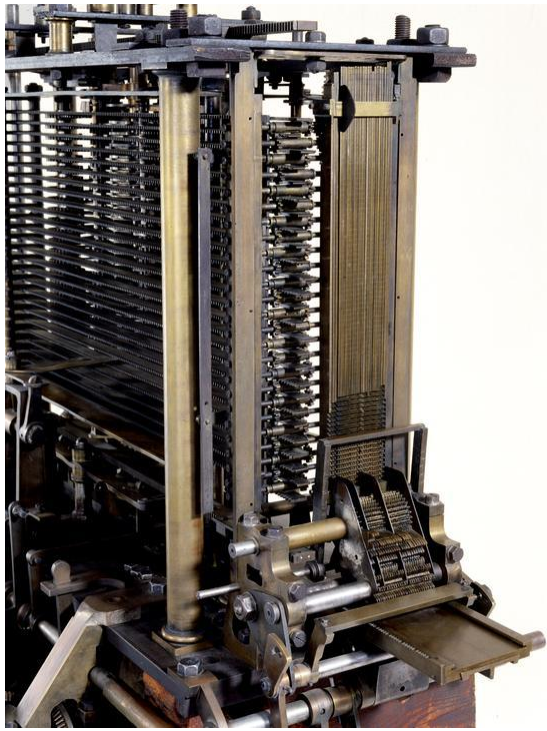


Чарльз Бэббидж (1791-1871),
Англия



Разностная машина Бэббиджа (проект
1822 года) для составления таблиц
значений функций

Автоматизация вычислений

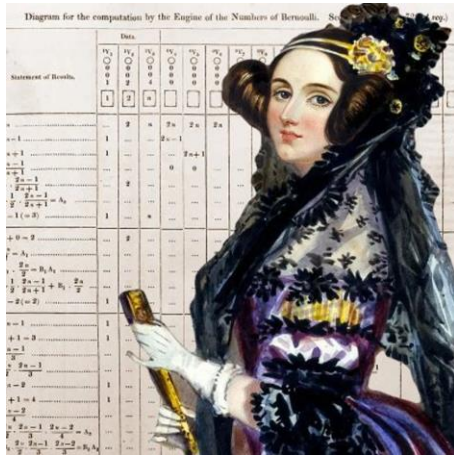


Аналитическая машина Бэббиджа (идея 1834 года) - прообраз цифровых компьютеров:

- *Склад* для сохранения значений переменных и результатов вычислений (память)
- *Мельница* для выполнения арифметических действий над переменными (арифметическое устройство)
- Устройство управления последовательностью операций
- Механизм ввода исходных данных и инструкций с перфокарт
- Механизм печати результатов

"Сам процесс вычисления осуществляется с помощью алгебраических формул, записанных на перфорированных картах. Вся умственная работа сводится к написанию формул, пригодных для вычислений, производимых машиной, и неких простых указаний, в какой последовательности эти вычисления должны производиться" Луи Менабреа

Первая программа в мире



Ада Лавлейс (Байрон) (1817-1852), Англия

- Перевела и значительно дополнила работу Луи Менабреа.
- В комментариях привела алгоритм вычисления чисел Бернулли на аналитической машине Бэббиджа.
- Использовала термины "рабочая ячейка" (переменная) и "цикл".

"Суть и предназначение машины изменятся от того, какую информацию мы в нее вложим. Машина сможет писать музыку, рисовать картины и покажет науке такие пути, которые мы никогда и нигде не видели"

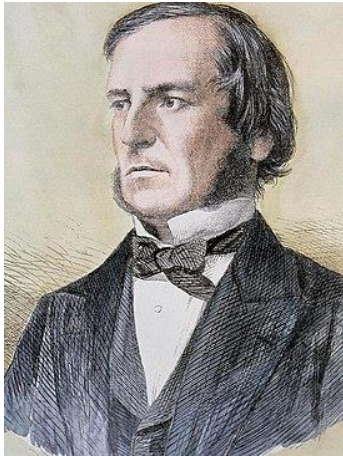
Двоичная система счисления



Готфрид Вильгельм Лейбниц (1646-1716),
Германия

- Создал комбинаторику как науку.
- Заложил основы математической логики
- Описал двоичную систему счисления с цифрами 0 и 1.

Логика и математика



Джордж Буль (1815-1864), Англия

- Выполнение символических математических операций не только над числами, но и над множествами.
- Законы алгебры логики позволяют упрощать логические выражения точно так же, как элементарная алгебра упрощает числовые.

Логические переменные: 0 и 1

Логические операторы: AND (И), OR (ИЛИ), NOT (НЕ), XOR (Исключающее ИЛИ)

Логические выражения можно реализовать с помощью телеграфных реле или переключателей, проводов и лампочек.

Алгебра логики

Логические переменные: 0 (ложь) и 1 (истина)

Логические операторы: AND (И), OR (ИЛИ), NOT (НЕ), XOR (Исключающее ИЛИ)

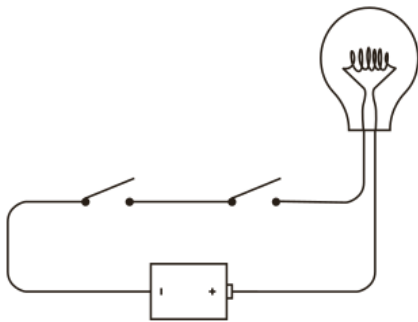
название	для И	для ИЛИ
двойного отрицания	$\overline{\overline{A}} = A$	
исключения третьего	$A \cdot \overline{A} = 0$	$A + \overline{A} = 1$
операции с конст.	$A \cdot 0 = 0, A \cdot 1 = A$	$A + 0 = A, A + 1 = 1$
повторения	$A \cdot A = A$	$A + A = A$
поглощения	$A \cdot (A + B) = A$	$A + A \cdot B = A$
переместительный	$A \cdot B = B \cdot A$	$A + B = B + A$
сочетательный	$A \cdot (A \cdot B) = (A \cdot B) \cdot C$	$A + (B + C) = (A + B) + C$
распределительный	$A + B \cdot C = (A + B) \cdot (A + C)$	$A \cdot (B + C) = A \cdot B + A \cdot C$
законы де Моргана	$\overline{A \cdot B} = \overline{A} + \overline{B}$	$\overline{A + B} = \overline{A} \cdot \overline{B}$

Логика и числа

- Группы логических переменных могут представлять числа в двоичной форме
- Логические операции в случае с двоичными числами могут объединяться для арифметических расчетов

Булевы выражения и окружающий мир

Логические выражения можно реализовать с помощью телеграфных реле или переключателей, проводов и лампочек.



Левый переключатель

Разомкнут
Разомкнут
Замкнут
Замкнут

Правый переключатель

Разомкнут
Замкнут
Разомкнут
Замкнут

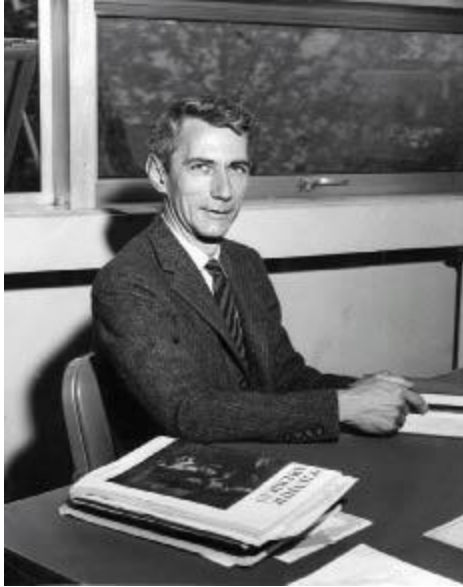
Лампочка

Не горит
Не горит
Не горит
Горит

Последовательное соединение переключателей	0	1
	0	0
1	0	1

И	0	1
0	0	0
1	0	1

Булева алгебра и электрические цепи



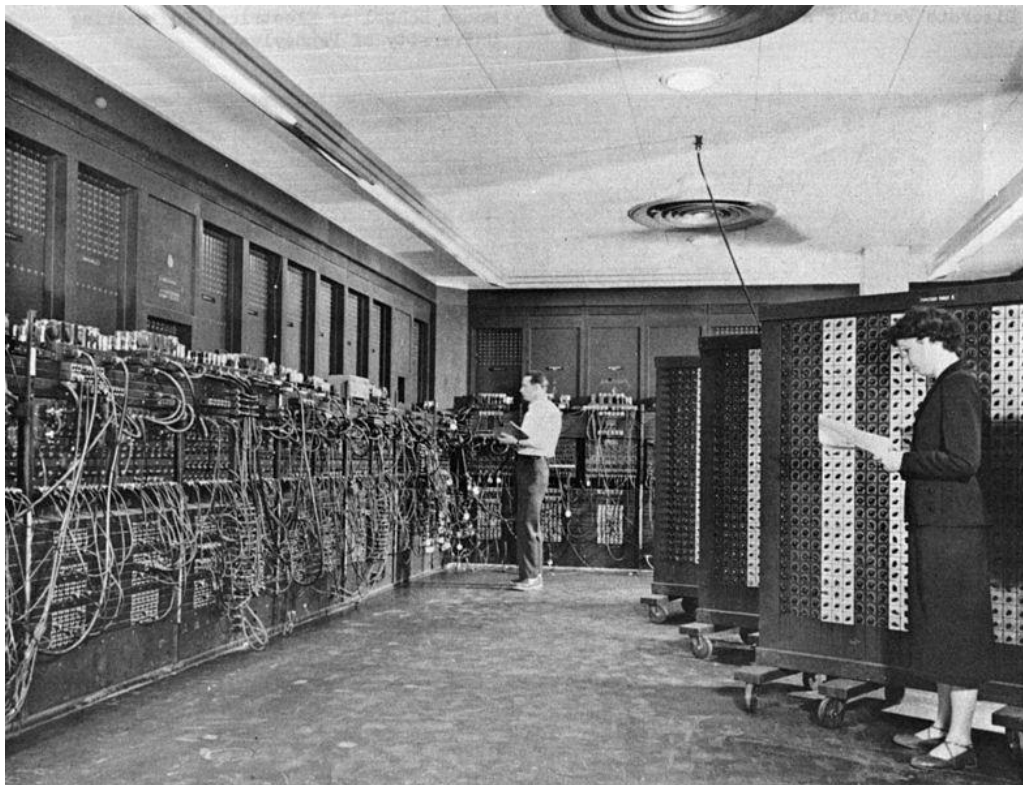
Клод Шеннон (1916-2001), США

- Указал на возможность реализовать логические выражения в виде электрических цепей (реле и переключатели)
- Ввел слово "бит" для обозначения двоичной цифры
- Предложил использовать бит как единицу измерения количества информации

Тело и душа компьютера

Нематериальный мир, законы математики	Материальный мир, законы физики
Компьютеры работают в двоичной системе из 0 и 1	Компьютеры - это электронные устройства, построенные на основе электрических цепей и цифровых схем

Первые компьютерные системы: ограниченный набор исполняемых команд и программ



1945 год. Компьютер ENIAC (Electronic Numerical Integrator and Computer)

Площадь — 167 кв.м.
Вес — 27 тонн

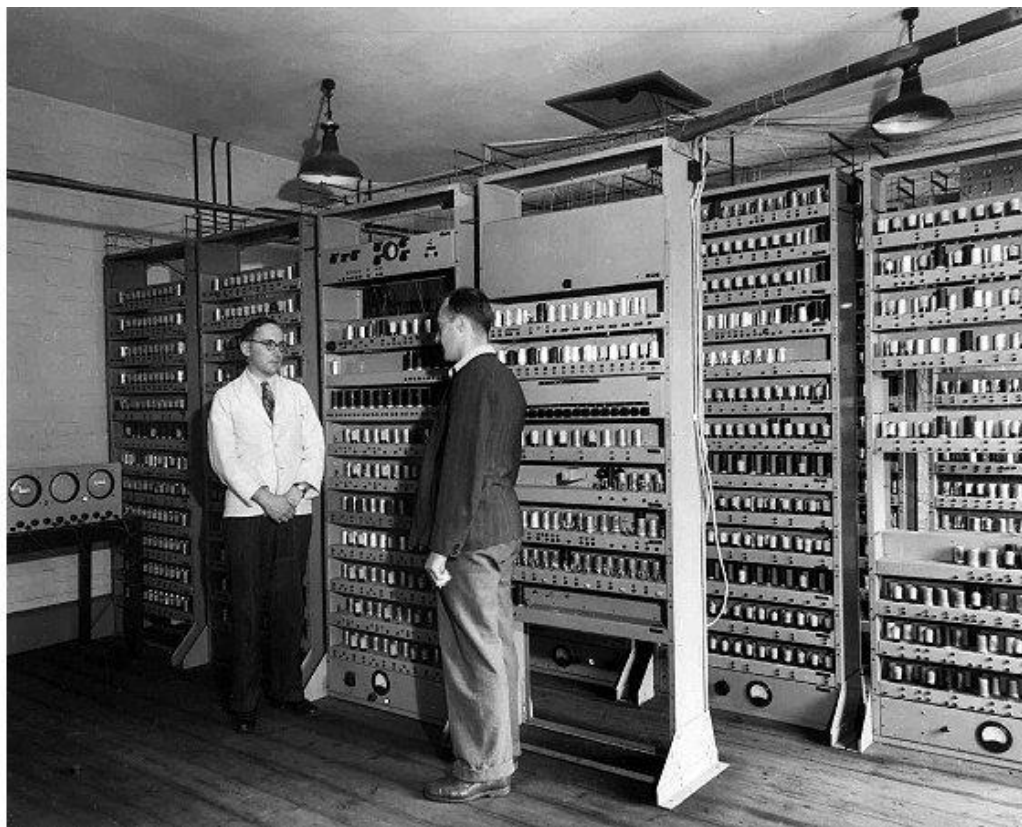
18 000 электронных ламп

Десятичная арифметика

Ввод/вывод данных — перфокарты

Программирование — ручная установка переключателей

Первый компьютер с архитектурой фон Неймана



**1949 год. Компьютер
EDSAC (Electronic Delay
Storage Automatic Computer)
Кембриджский университет**

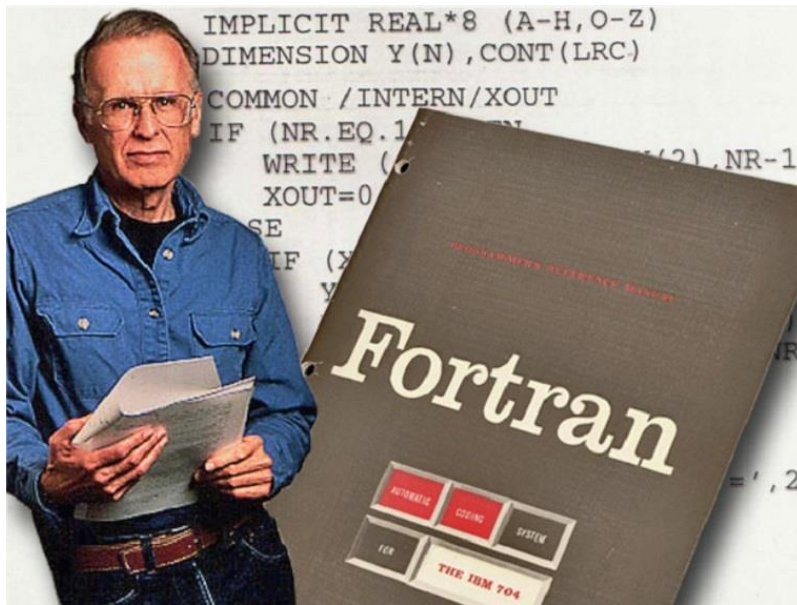
Площадь — 20 кв.м.

3 000 электронных ламп

Поддерживал язык ассемблера и подпрограммы.

В 1952 году на нем была написана первая игра «Крестики-нолики».

Fortran – первый высокоуровневый ЯП



1954 год

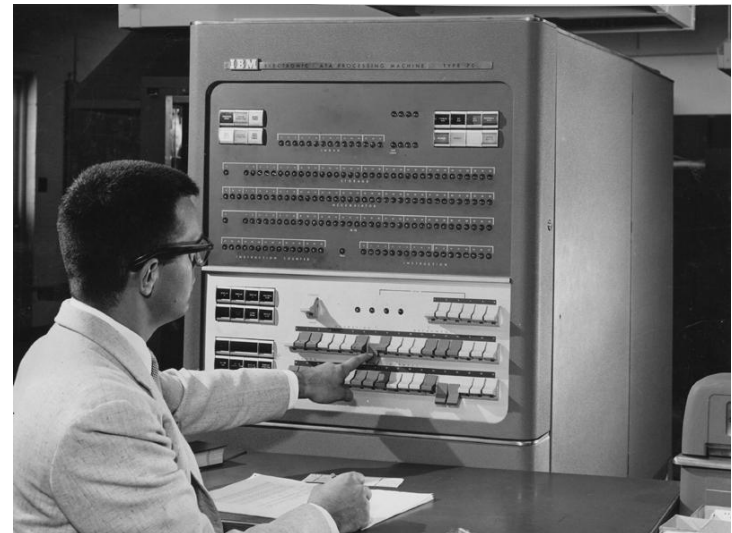
Джон Бэкус (John Backus)

1924-2007

Разрабатывался в фирме IBM
для компьютера IBM 704

Formula Translation

- Оператор присваивания
- Массивы
- Оператор цикла DO



Данные в ЭВМ

Данные в первых ЭВМ





5 Мб данных

62 500 перфокарт

Магнитные ленты



Первый HDD
IBM, 1956

5 Mb, 1 тонна



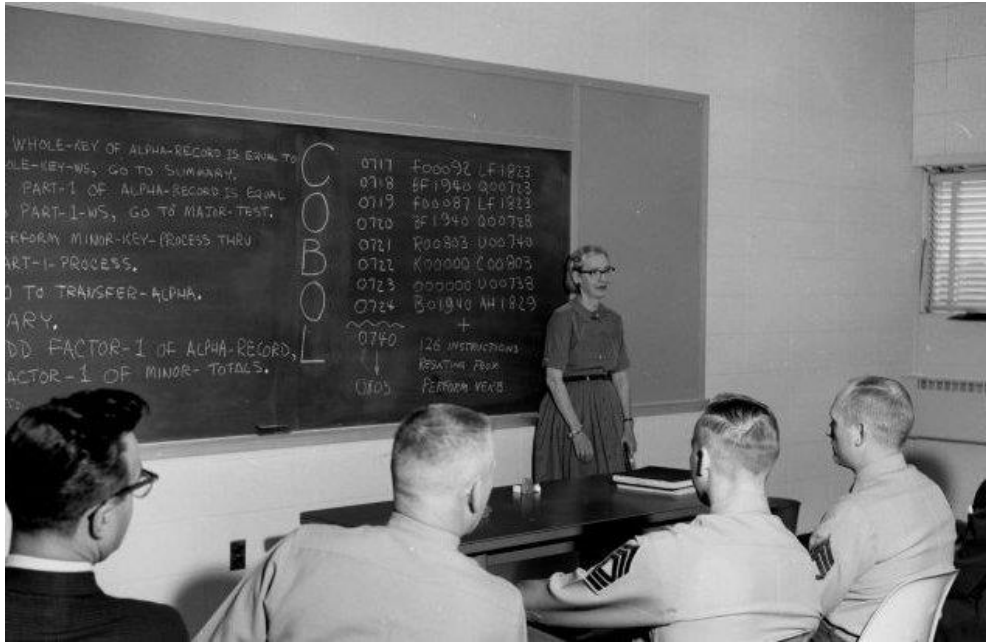
Сравнительная таблица основных параметров ЭВМ

Параметры сравнения	Поколения ЭВМ			
	Первое	Второе	Третье	Четвертое
Период времени	1946 - 1959	1960 - 1969	1970 - 1979	С 1980 г.
Элементная база (для УУ, АЛУ)	Электронные (или электрические) лампы	Полупроводники (транзисторы)	Интегральные схемы	Большие интегральные схемы (БИС)
Основной тип ЭВМ	Большие		Малые (мини)	Микро ЭВМ
Основные устройства ввода	Пульт, перфоленточный, перфокарточный ввод	Добавился алфавитно-цифровой дисплей, клавиатура	Алфавитно-цифровой дисплей, клавиатура	Цветной графический дисплей, сканер, клавиатура
Основные устройства вывода	Алфавитно-цифровое печатающее устройство (АЦПУ), перфоленточный вывод		Графопостроитель, принтер	
Внешняя память	Магнитные ленты, барабаны, перфоленты, перфокарты	Добавился магнитный диск	Перфоленты, магнитный диск	Магнитные и оптические диски
Ключевые решения в ПО	Универсальные языки программирования, трансляторы	Пакетные операционные системы, оптимизирующие трансляторы	Интерактивные операционные системы, структурированные языки программирования	Дружественность ПО, сетевые операционные системы
Режим работы ЭВМ	Однопрограммный	Пакетный	Разделения времени	Персональная работа и сетевая обработка данных
Цель использования ЭВМ	Научно-технические расчеты	Технические и экономические расчеты	Управление и экономические расчеты	Телекоммуникации, информационное обслуживание

Классификация задач, решаемых на компьютере

Тип	Представление данных	Алгоритм вычислений
Вычислительные задачи	Простое	Сложный
Традиционные задачи обработки данных (невычислительные)	Сложное	Простой
Современные задачи обработки данных	Сложное	Сложный

COBOL – первый язык для бизнес-приложений



1959 год

Грейс Хоппер – руководитель проекта (язык Flow-Matic)

Common **B**usines
Oriented **L**anguage

- Структуры данных (записи)
- Файлы

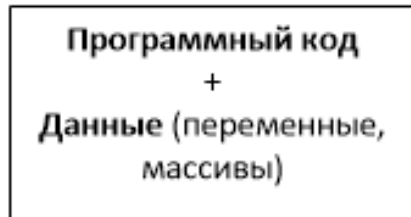
```
PERFORM UNTIL EndOfStudentFile
    ADD 1 TO MonthCount(MOBirth)
    READ StudentFile
        AT END SET EndOfStudentFile TO TRUE
    END-READ
END-PERFORM
```

Эволюция представления данных: от ФС к БД

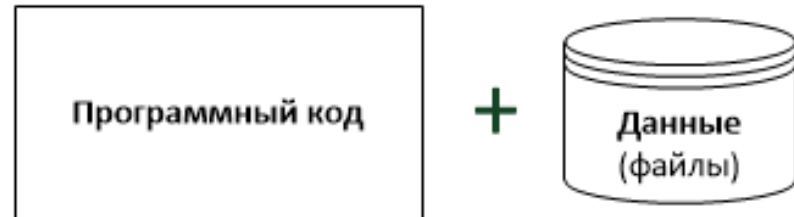
1950-60 годы: данные отделяются от программ и обособляются в файлы.

Первые коммерческие программы – для ведения бухгалтерии (file folder = папка для бумаг).

Вычислительные задачи



Задачи обработки данных



Концепции баз данных – это результат развития файловых систем.

Логическая и физическая структура файлов

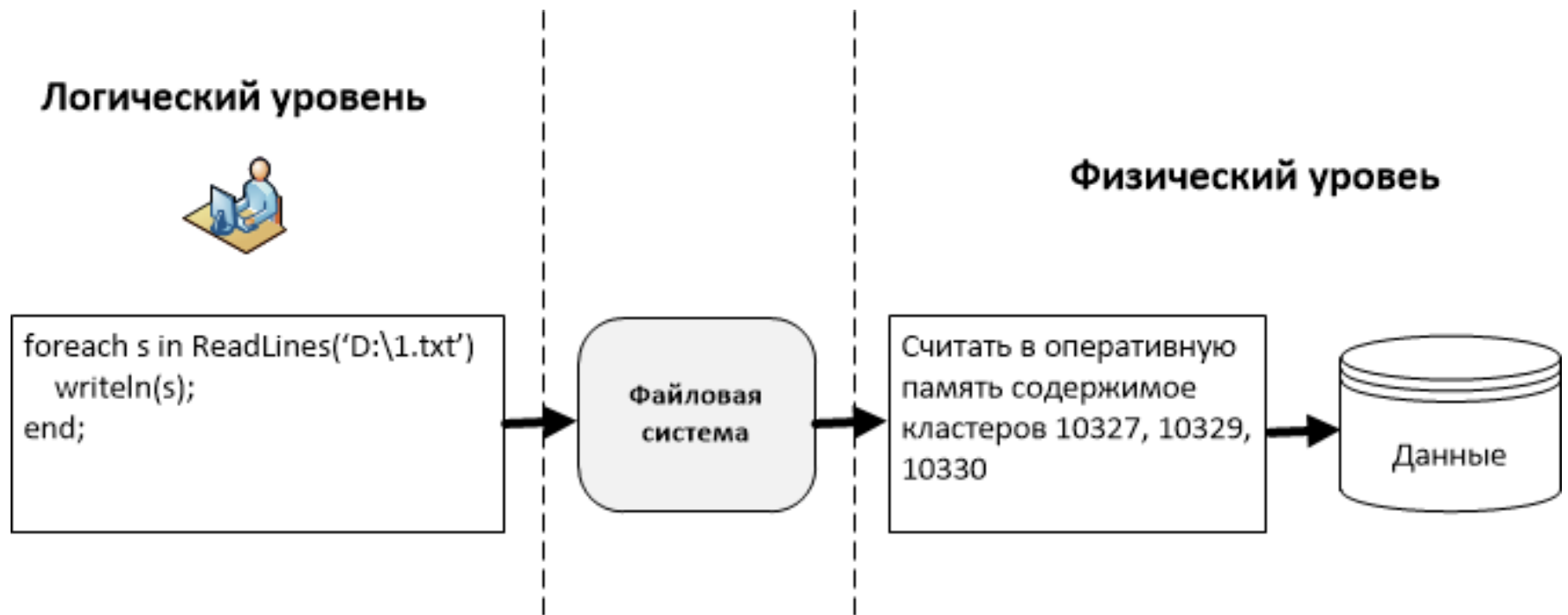
Для прикладной программы файл – это **именованная область внешней памяти**, в которую можно записывать и из которой можно считывать данные.



Файл на внешнем носителе – это **цепочка кластеров** (физических записей).

Файловая система – для абстрагирования от данных

Пользователь/программист имеет дело только с логическими данными (более удобная форма), он не касается деталей фактического низкоуровневого размещения данных.



C:\Документы\Мой файл.doc

Каталог файлов

Дескриптор файла 1
...
Дескриптор файла N

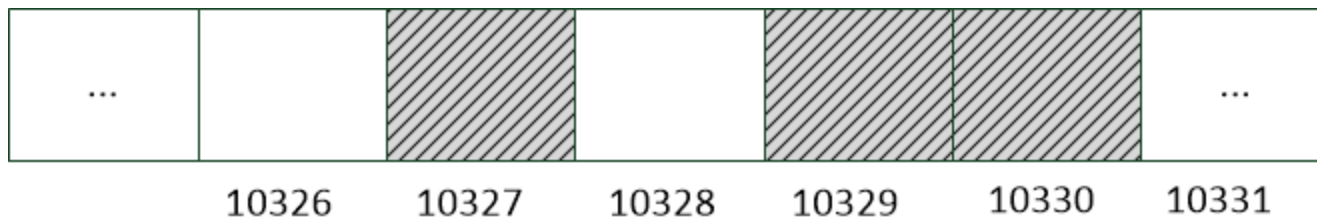
Дескриптор файла

Имя	<i>Мой файл.doc</i>
Дата создания	<i>01.02.2020</i>
Атрибуты	<i>Archive</i>
Первый кластер	<i>10327</i>
Размер	<i>9326</i>
...	...

Таблица размещения файлов

№ кластера	Статус
10326	<i>Сбойный</i>
10327	<i>10329</i>
10328	<i>Свободный</i>
10329	<i>10330</i>
10330	<i>Конец цепочки</i>
10331	<i>Свободный</i>
...	...

File Allocation Table (FAT)



Кластеры на диске

Файловая система – часть операционной системы, включающая:

- Совокупность всех файлов на диске с их **физической организацией**.
- Структуры данных управления файлами (каталоги файлов, дескрипторы файлов, таблицы распределения файлов, ...), т. е. **логическая организация файловых структур**.
- Комплекс системных **программных средств**, реализующих управление файлами (создание, уничтожение, чтение, запись, поиск и другие операции над файлами).

Напишем консольное приложение для работы с данными в файле:

- Фамилия, имя, пол, возраст студентов
- CRUD-операции

Create **R**ead **U**ppdate **D**elete

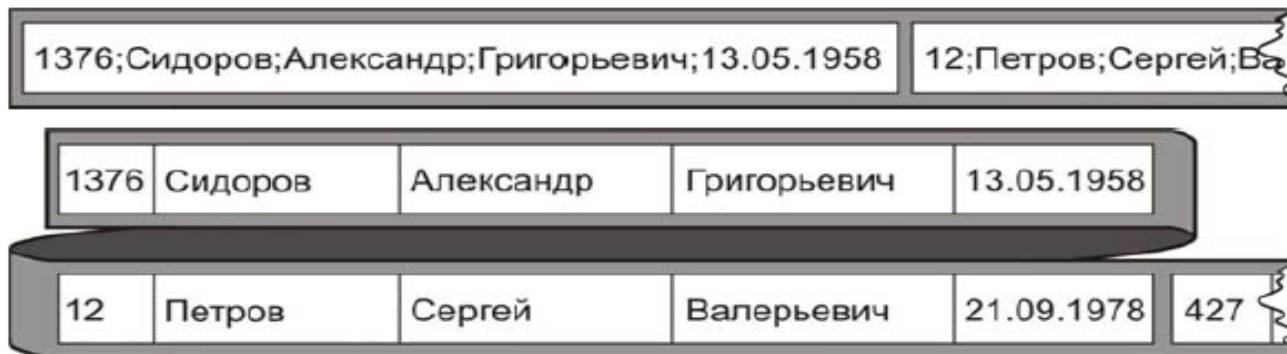
Плоские структурированные файлы

Логические записи

id	surname	name	patronymic	birthday
1376	Сидоров	Александр	Григорьевич	13.05.1958
12	Петров	Сергей	Валерьевич	21.09.1978
...

Физические записи

- Список полей с символами-разделителями.
- Список из полей фиксированной длины, равной ширине соответствующих столбцов таблицы.



Структурированные данные - поддержка в ЯП

COBOL (Common Business-Oriented Language), 1959 год

Pascal позволяет определять составные типы данных (записи), создавать переменные таких типов и сохранять их в файле.

```
Type Person = Record
  id: LongInt;
  surname: string[30];
  name: string[30];
  patronymic: string[30];
  birthday: string[10];
end;
var f: file of Person;
```

Файл содержит логические записи, состоящие из полей.

Напишем консольное приложение для работы с данными в файле:

- Фамилия, имя, пол, возраст студентов
- CRUD-операции

Free Pascal, 310 строк

Плоские структурированные файлы

Плоские файлы – прообраз баз данных

- В файле содержатся только данные, информации о записях нет.
- Структура записей задается в прикладной программе.

Недостатки

- Жесткая привязка приложения к физической структуре файла.
- Каждый новый отчет потребует изменение приложения.
- Поддержка безопасности в программном коде.
- Поддержка целостности данных в программном коде.
- Администрирование (резервные копии, восстановление данных) в программном коде.
- Организация многопользовательского режима в программном коде.

Файлы с метаданными

Файл состоит из:

- заголовка, где хранится информация о структуре записей (имена и размерность полей) и их количестве;
- области данных из записей фиксированной длины.

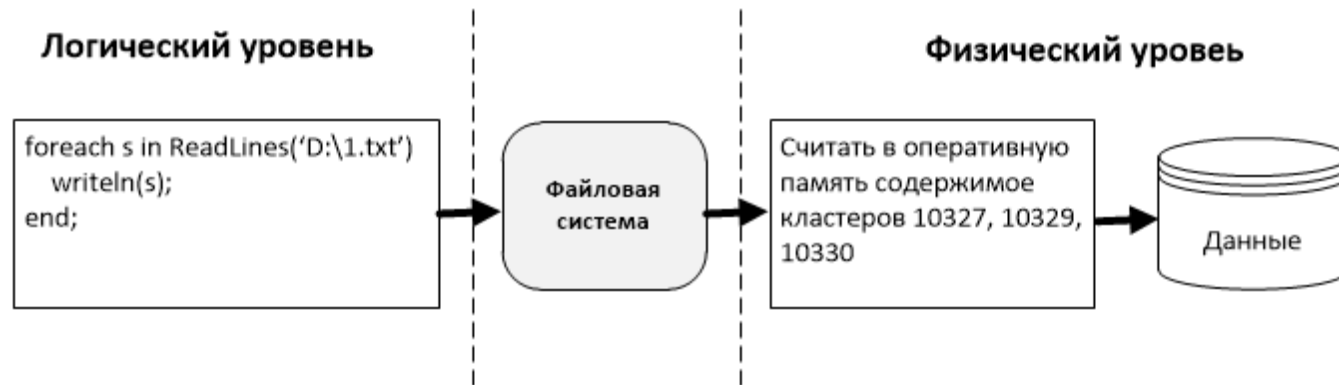
<i>Записи из пяти полей. 1-е поле: Id, целое, длина 10 байт. 2-е поле: Lastname, символьное, длина 30 байт. ...</i>	<i>Запись 1</i>	<i>Запись 2</i>	<i>...</i>
Заголовок	Данные		

Преимущество: структура данных описывается в самих файлах с данными, а не в прикладных программах.

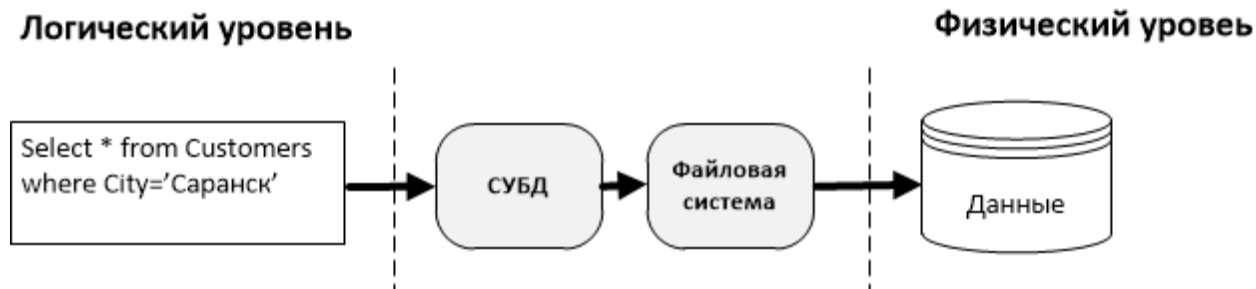
Файлы DBF – стандартный формат для ранних баз данных на персональных компьютерах.

СУБД – следующий шаг в абстрагировании пользователя/программиста от данных

Работа с файлом



Работа с базой данных



Перепишем наше CRUD-приложение с использованием СУБД SQLite

Free Pascal, 287 строк

Данные в нескольких источниках (файлах)

Задача 1. Кадровый учет. Файл:

Фамилия	Имя	Отчество	Дата рожд-я	Место жит-ва	Должность	Оклад
Иванов	Иван	Иванович	01.02.1985	г. Саранск	Инженер	30000

Задача 2. Начисление заработной платы. Файл:

Фамилия	Имя	Отчество	Оклад	Месяц	Отработано, дней	Сумма
Иванов	Иван	Иванович	30000	Февраль	10	15000

Задача 3. Учет больничных. Файл:

Фамилия	Имя	Отчество	Оклад	Месяц	На больн-м, дней	Сумма
Иванов	Иван	Иванович	30000	Февраль	6	8500

Данные в нескольких источниках (файлах)

Задача 1. Кадровый учет. Файл:

Фамилия	Имя	Отчество	Дата рожд-я	Место жит-ва	Должность	Оклад
Иванов	Иван	Иванович	01.02.1985	г. Саранск	Инженер	30000

Задача 2. Начисление заработной платы. Файл:

Фамилия	Имя	Отчество	Оклад	Месяц	Отработано, дней	Сумма
Иванов	Иван	Иванович	30000	Февраль	10	15000

Задача 3. Учет больничных. Файл:

Фамилия	Имя	Отчество	Оклад	Месяц	На больн-м, дней	Сумма
Иванов	Иван	Иванович	30000	Февраль	6	8500

Информация дублируется – это плохо!

Данные в нескольких источниках (файлах)

Задача 1. Кадровый учет. Файл:

Фамилия	Имя	Отчество	Дата рожд-я	Место жит-ва	Должность	Оклад
Иванов	Иван	Иванович	01.02.1985	г. Саранск	Инженер	30000

Задача 2. Начисление заработной платы. Файл:

Фамилия	Имя	Отчество	Оклад	Месяц	Отработано, дней	Сумма
Иванов	Иван	Иванович	30000	Февраль	10	15000

Задача 3. Учет больничных. Файл:

Фамилия	Имя	Отчество	Оклад	Месяц	На больн-м, дней	Сумма
Иванов	Иван	Иванович	30000	Февраль	6	8500

Информация **дублируется** – данные могут стать противоречивыми.

Данные нужно **интегрировать**, хранить в одном месте.

Общая информационная база

Вариант 1. Объединить все в одном файле:

ФИО	Дата рожд-я	Место жит-ва	Должность	Оклад	Месяц	Отработано, дней	На больн-м, дней	Зарплата	Больнич.
-----	-------------	--------------	-----------	-------	-------	------------------	------------------	----------	----------

Общая информационная база

Вариант 1. Объединить все в одном файле:

ФИО	Дата рожд-я	Место жит-ва	Должность	Оклад	Месяц	Отработано, дней	На больн-м, дней	Зарплата	Больнич.
-----	-------------	--------------	-----------	-------	-------	------------------	------------------	----------	----------

Недостатки:

- Останется дублирование данных внутри файла.
- Сильно возрастет время решения задачи 3.

Общая информационная база

Вариант 1. Объединить все в одном файле:

ФИО	Дата рожд-я	Место жит-ва	Должность	Оклад	Месяц	Отработано, дней	На больн-м, дней	Зарплата	Больнич.
-----	-------------	--------------	-----------	-------	-------	------------------	------------------	----------	----------

Недостатки:

- Останется дублирование данных внутри файла.
- Сильно возрастет время решения задачи 3.

Вариант 2. Два файла:

Фамилия	Имя	Отчество	Дата рожд-я	Место жит-ва	Должность	Оклад
---------	-----	----------	-------------	--------------	-----------	-------

ФИО	Оклад	Месяц	Отработано, дней	На больн-м, дней	Зарплата	Больничный
-----	-------	-------	------------------	------------------	----------	------------

Вариант 3. Два файла:

Табельный номер	Фамилия	Имя	Отчество	Дата рождения	Место жит-ва	Должность	Оклад
-----------------	---------	-----	----------	---------------	--------------	-----------	-------

Табельный номер	Оклад	Месяц	Отработано, дней	На больн-м, дней	Зарплата	Больничный
-----------------	-------	-------	------------------	------------------	----------	------------

База данных как новый вид данных

База данных – совокупность взаимосвязанных хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений.

ISO, 2015: База данных - совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними, причем такое собрание данных, которое поддерживает одну или более областей применения.

- Базы данных нужны при использовании общих данных несколькими задачами (программами).
- Основной критерий оптимальности функционирования базы данных – время выполнения запросов пользователей к данным.

Стоимость типовых операций



Стоимость операции	нс (ns)	мкс (μs)	мс (ms)
Получение значения из L1	0.5		
Ошибка предсказания перехода в CPU	5		
Получение значения из L2	7		
Mutex lock/unlock	25		
Получение значения из RAM	100		
Сжатие 1Кб методом Zipru	3 000	3	
Отправка 1Кб через 1Гбит/сек сеть	10 000	10	
Чтение 4Кб с SSD (случайный доступ)	150 000	150	
Чтение 1Мб из RAM (последовательный доступ)	250 000	250	
Round trip внутри одного датацентра	500 000	500	
Чтение 1Мб из SSD (последовательный доступ)	1 000 000	1 000	1
Позиционирование HDD	10 000 000	10 000	10
Чтение 1Мб из HDD (последовательный доступ)	20 000 000	20 000	20
Round trip между США и Нидерландами	150 000 000	150 000	150

<https://gist.github.com/jboner/2841832>

Основной критерий оптимальности функционирования базы данных – время выполнения запросов к данным.

Выводы

- Базы данных — результат развития файловых систем.
- Информация о структуре данных должны храниться в базе данных, а не в приложении.
- Дублирование информации в системе – **зло**, т. к. можно легко нарушить логическую целостность (непротиворечивость) данных.
- Информация в базе данных разделяется на связанные друг с другом части. Сделать это можно по-разному. Нужно минимизировать дублирование данных.