Лекция 1. Часть 2.

Базы данных: основные понятия

Базы данных: основные понятия

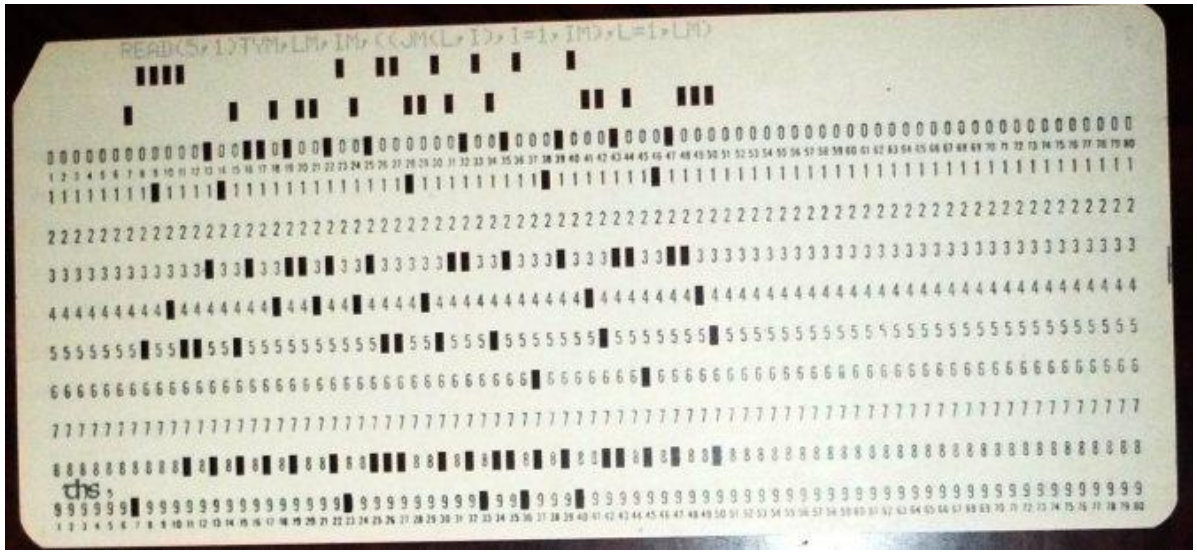
1. История возникновения баз данных. Файловая система как их прообраз.
2. Типы задач, для решения которых необходимы базы данных.

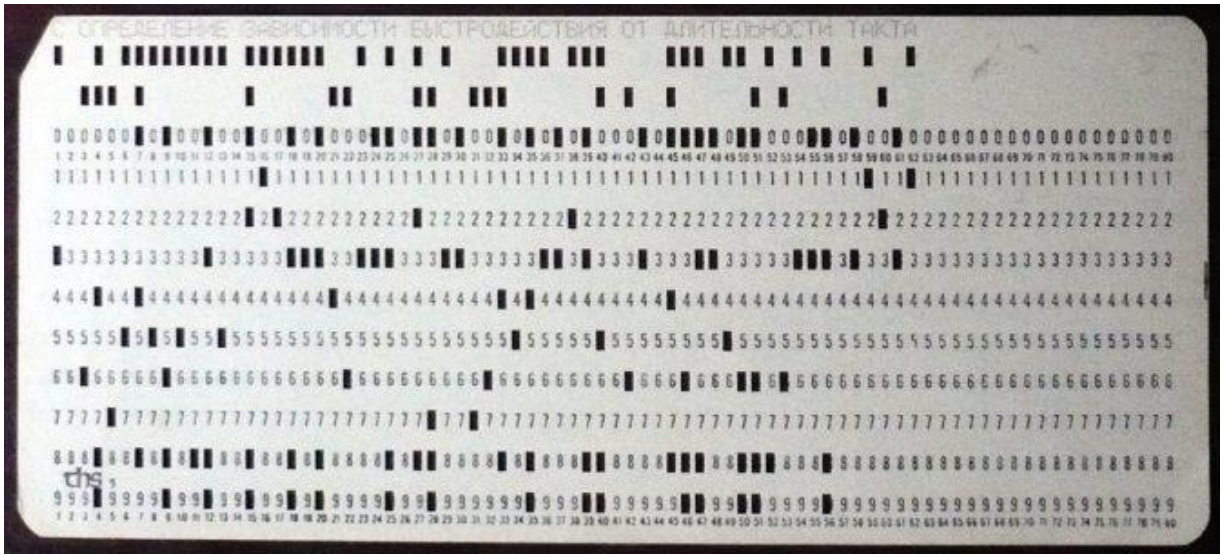
Классификация задач, решаемых на компьютере

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

Данные в первых ЭВМ

Перфокарты





5 Mb данных

62 500 перфокарт

Магнитные ленты

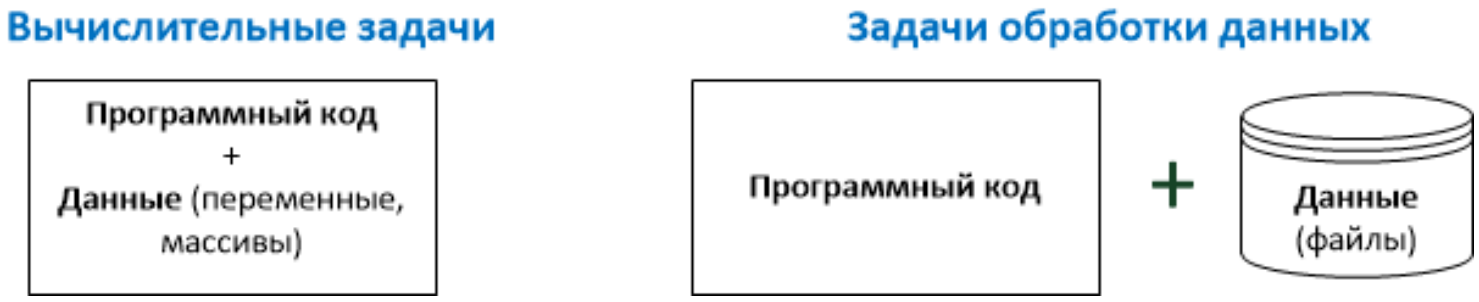
Первый HDD
IBM, 1956

5 Mb, 1 тонна

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).



Концепции баз данных – это результат развития файловых систем.

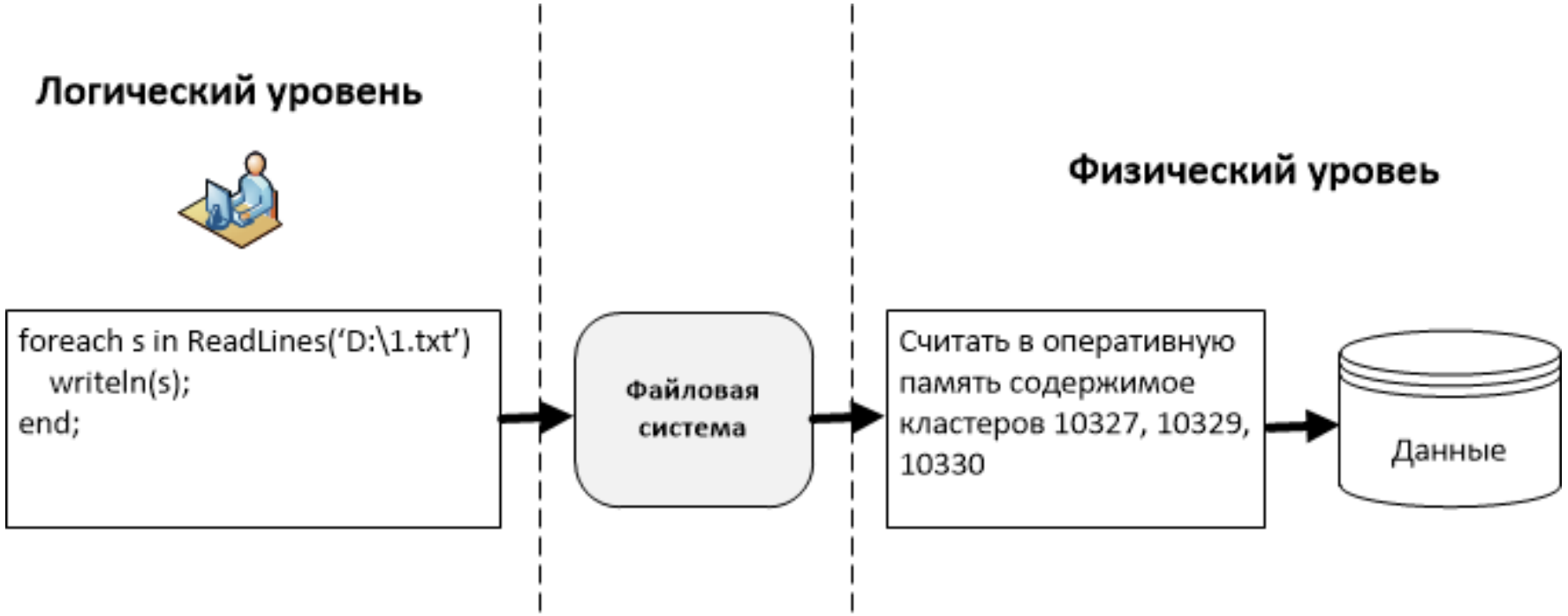
Логическая и физическая структура файлов

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

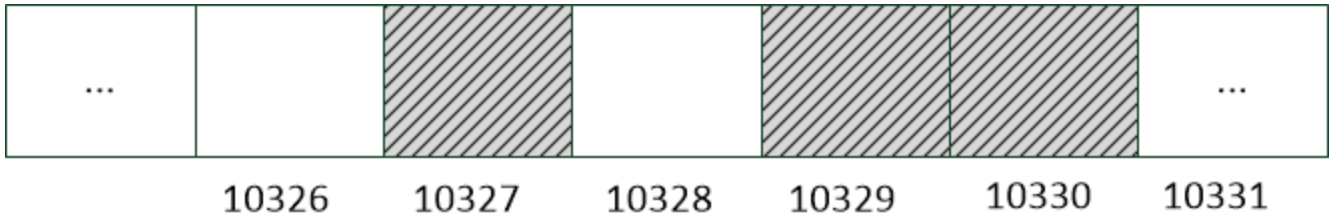
Дескриптор файла

Имя	*Мой файл.doc*
Дата создания	*01.02.2020*
Атрибуты	*Archive*
Первый кластер	*10327*
Размер	*9326*
…	…

Таблица размещения файлов

№ кластера	Статус
10326	*Сбойный*
10327	*10329*
10328	*Свободный*
10329	*10330*
10330	*Конец цепочки*
10331	*Свободный*
…	…

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**pdate **D**elete

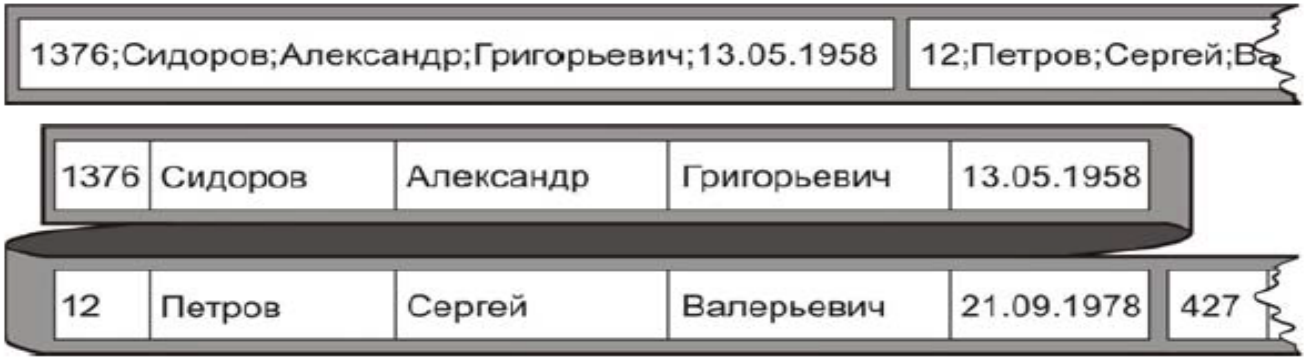
Плоские структурированные файлы

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:

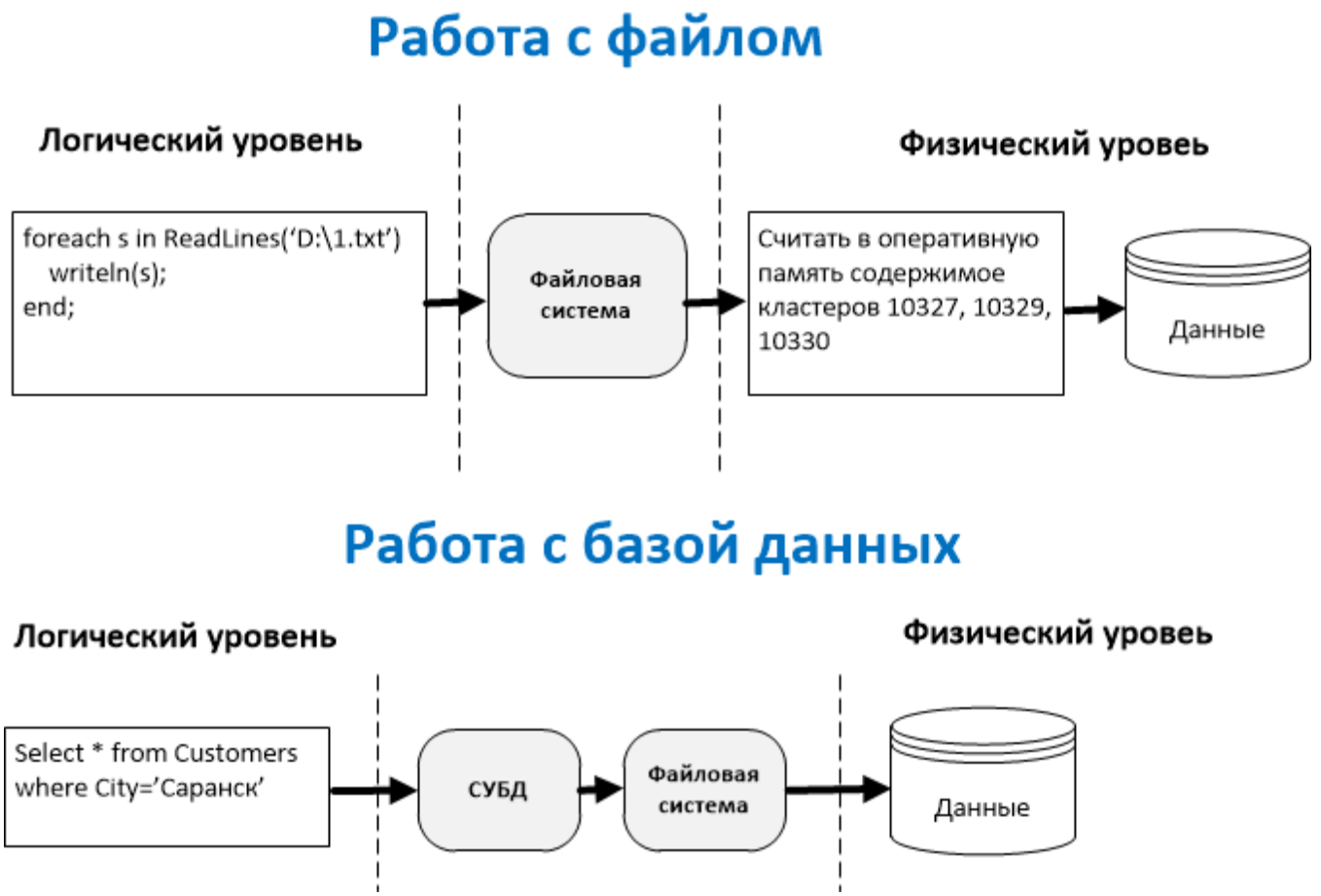
- заголовка, где хранится информация о структуре записей (имена и размерность полей) и их количестве;
- области данных из записей фиксированной длины.

<table>
<tr><td>Записи из пяти полей.
1-е поле: Id, целое, длина 10 байт.
2-е поле: Lastname, символьное, длина 30 байт.
…</td><td>Запись 1</td><td>Запись 2</td><td>…</td></tr>
<tr><td>Заголовок</td><td colspan="3">Данные</td></tr>
</table>

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



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	Боль-нич.

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 Zippy	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.