HighLoad++

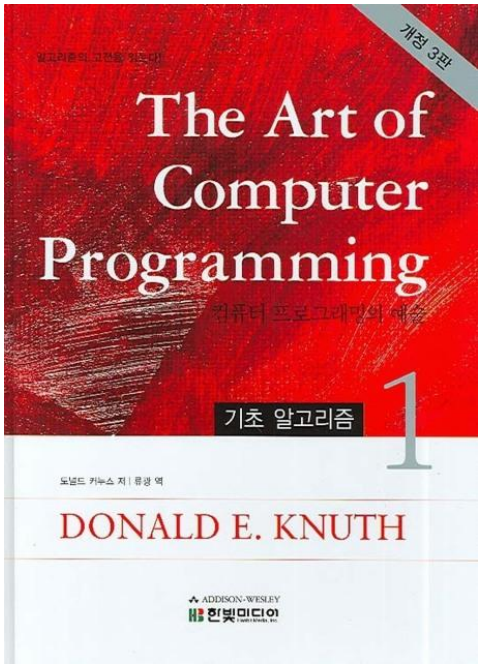
Зачем знать алгоритмы

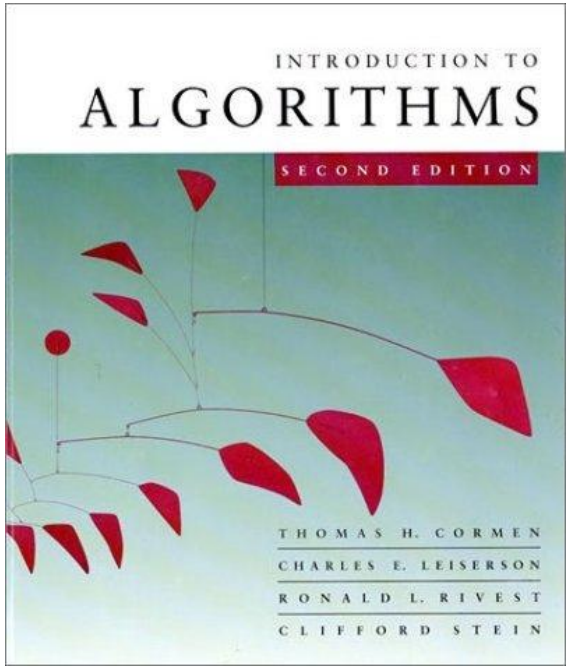
Андрей Аксенов
Sphinx Technologies Inc.

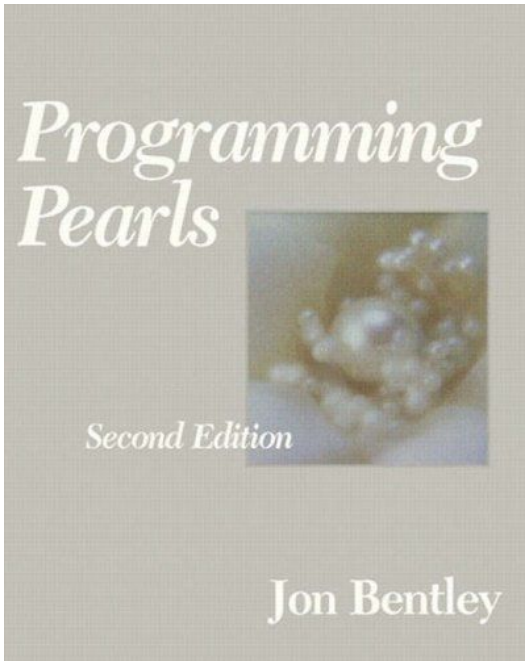
Highload++2009

Who is Mr. Aksenov?

HighLoad++


개정 3판
The Art of Computer Programming
기초 알고리즘
1
DONALD E. KNUTH
ADDISON-WESLEY


INTRODUCTION TO
ALGORITHMS
SECOND EDITION
THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN


Programming Pearls
Second Edition
Jon Bentley

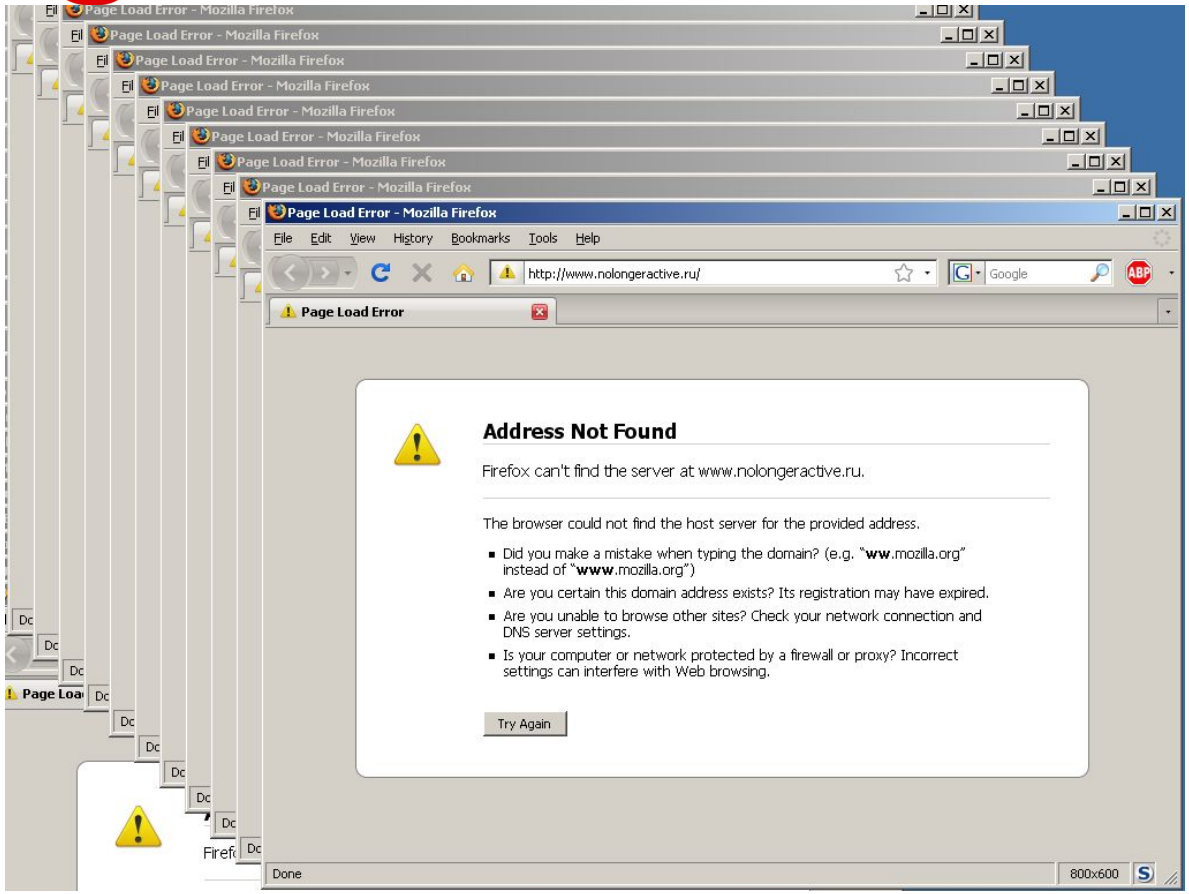
честно листал все!

не прочитал ни одной :(

HighLoad++

делал веб-сайты и веб-движок

HighLoad++


Page Load Error - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.nolongeractive.ru/
Google
Page Load Error
Address Not Found
Firefox can't find the server at www.nolongeractive.ru.
The browser could not find the host server for the provided address.
Did you make a mistake when typing the domain? (e.g. "ww.mozilla.org" instead of "www.mozilla.org")
Are you certain this domain address exists? Its registration may have expired.
Are you unable to browse other sites? Check your network connection and DNS server settings.
Is your computer or network protected by a firewall or proxy? Incorrect settings can interfere with Web browsing.
Try Again
Done
800x600

делал игры и 3D движок

hl++ HighLoad++


Коллекция игрушек
Магия Крови
SKY FALLEN

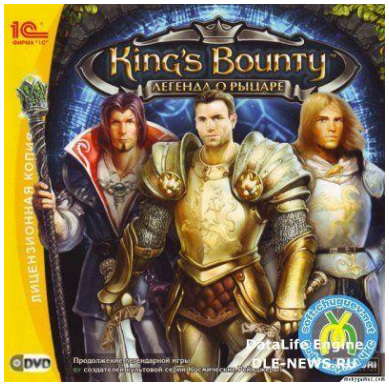

Время теней
Магия Крови
SKY FALLEN


Death Track
Возрождение
DVD


Коллекция игрушек
Санитары подземелий
DVD


Кодекс войны
Высшая раса
DVD


Ролевая игра по книге С.Лукьяненко и Н.Перумова
Не время для драконов
DVD


King's Bounty
Легенда о рыцаре
DVD


1С
КОЛЛЕКЦИЯ ИГРУШЕК
ЛИЦЕНЗИОННАЯ КОПИЯ ИГРЫ ОТ "1С"
2 CD
МАГИЯ КРОВИ
SKY FALLEN

<

HighLoad++

делаю поисковой движок

free open source search

hl++ HighLoad++

Sphinx <<

free :(

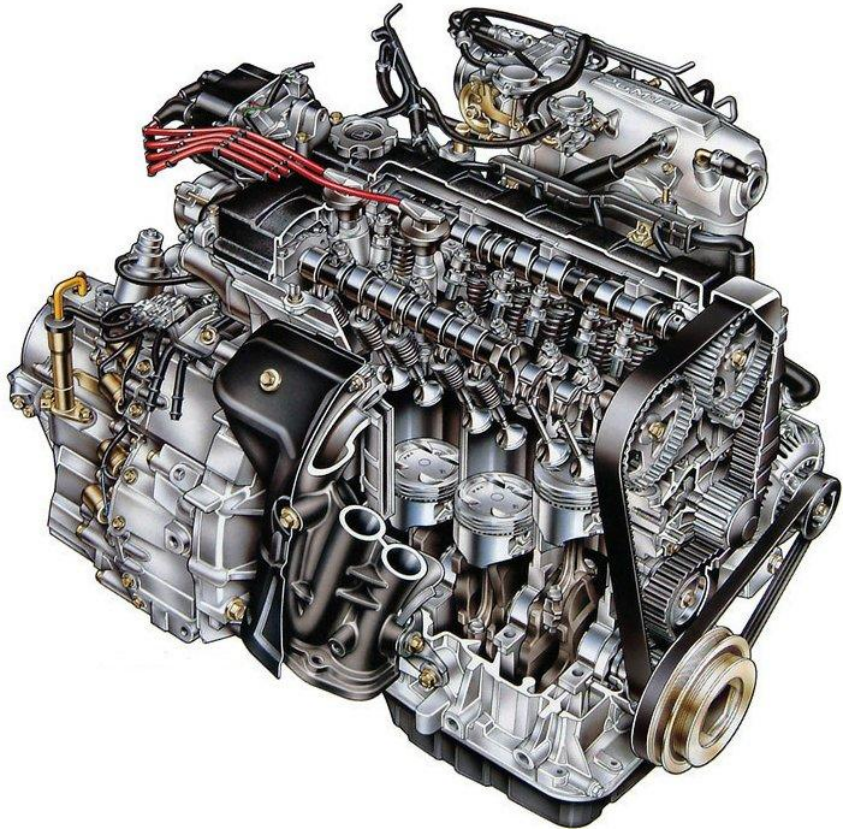
HighLoad++

что могу рассказать?

HighLoad++

как устроены всякие движки

HighLoad++



на пальцах, не по книжке!
(см. “не читатель”)

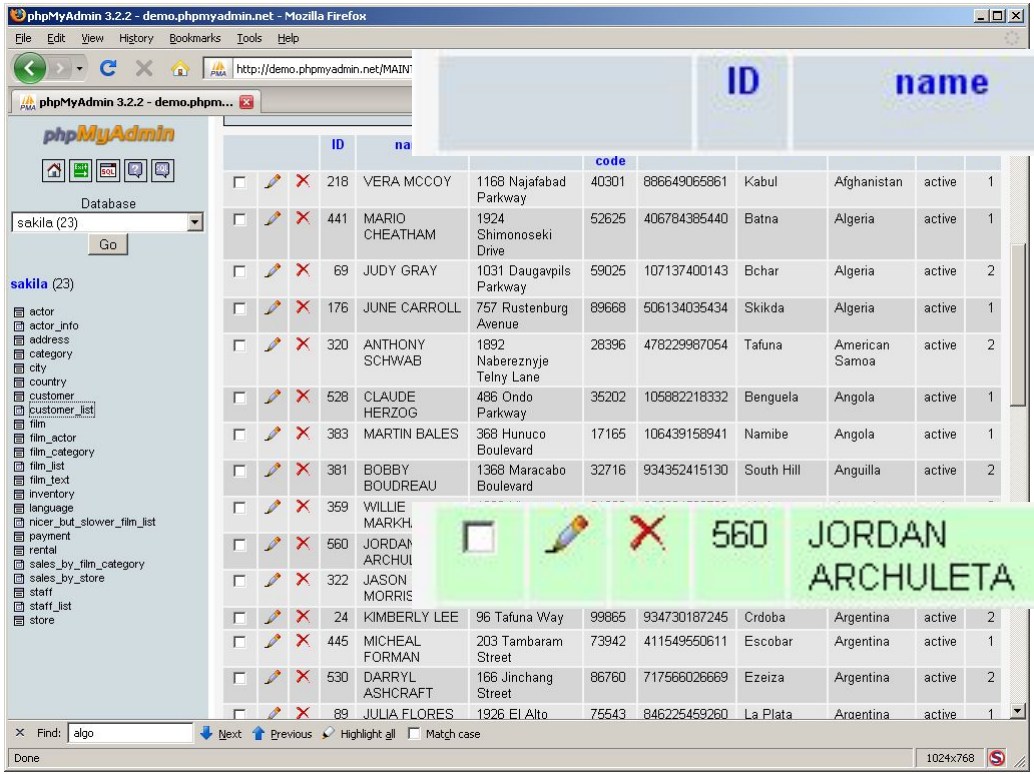
HighLoad++

про движок СУБД (любой)

Система Управления Базой Данных

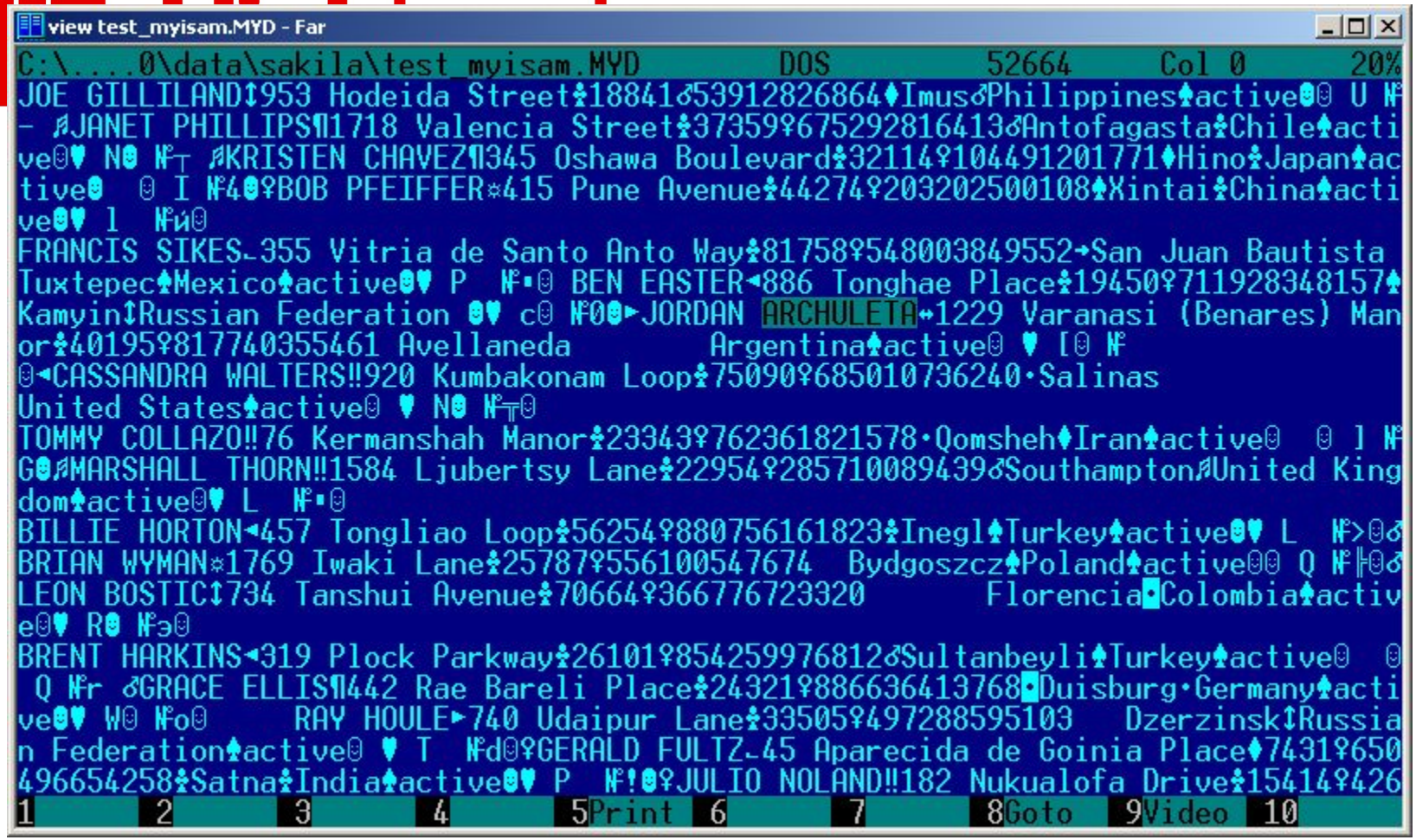
данные – это таблицы. из строк

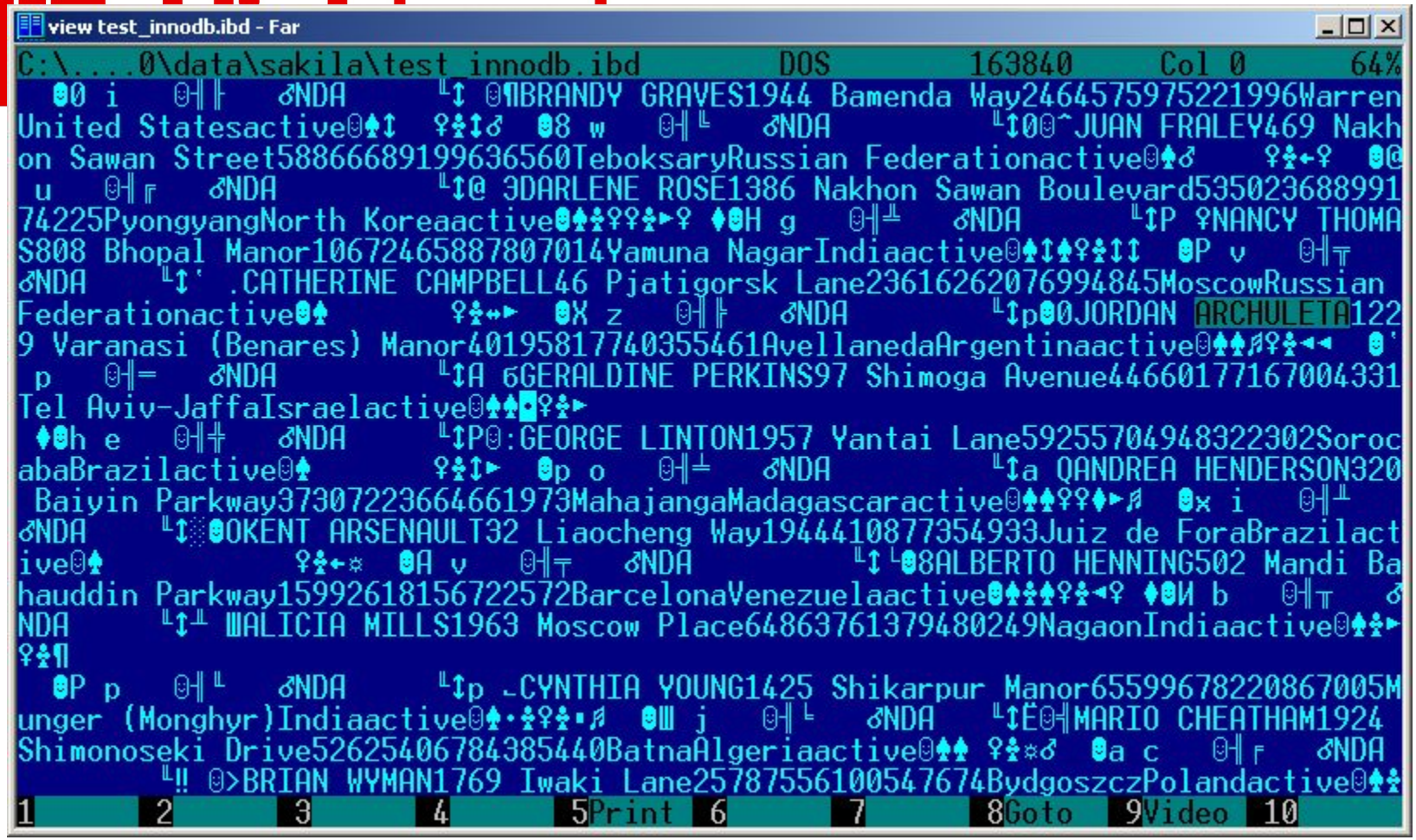
HighLoad++



	ID	name	address	zip code	phone	
	560	JORDAN ARCHULETA	1229 Varanasi (Benares) Manor	40195	817740355461	Ave

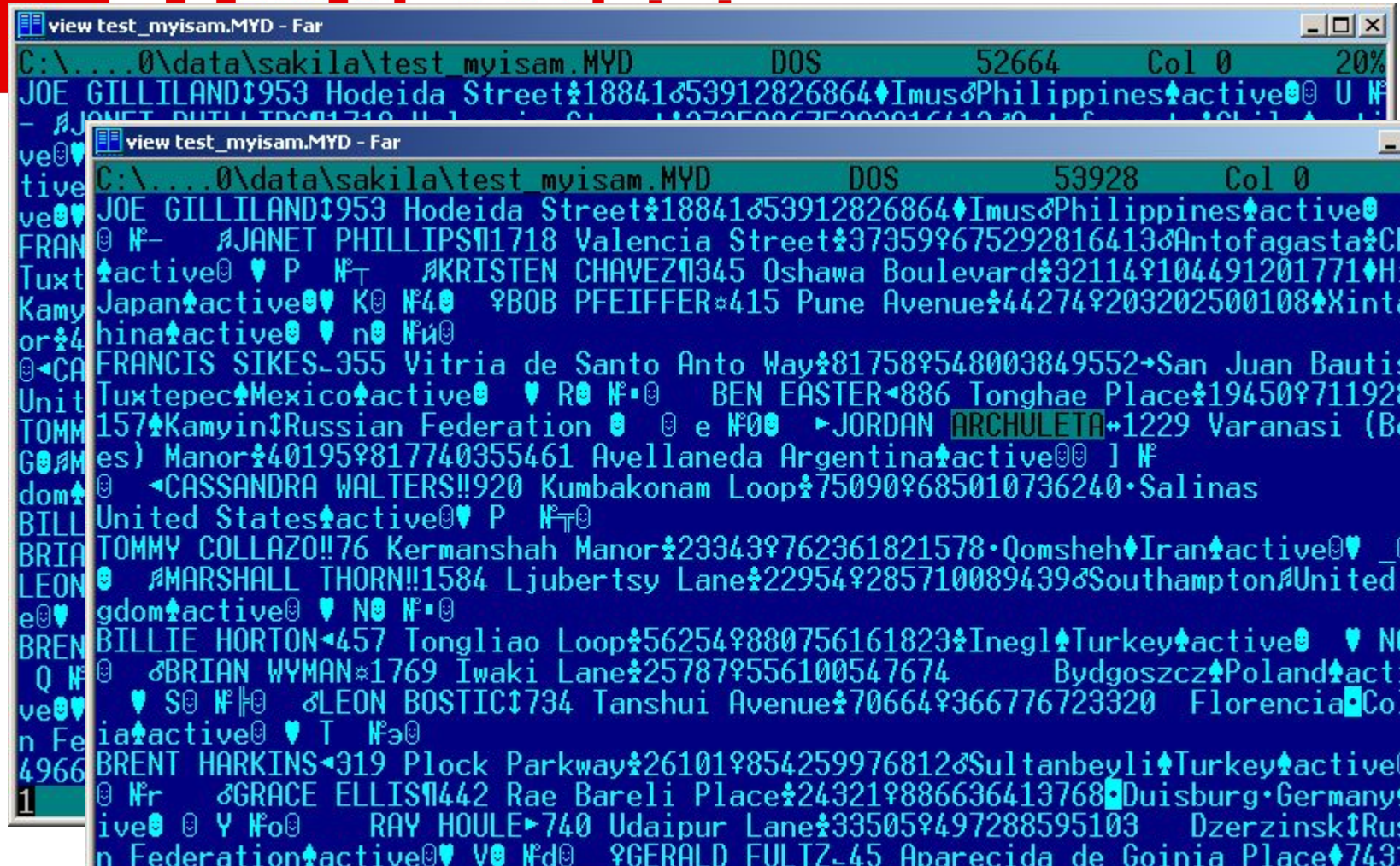
строки – нужно где-то хранить

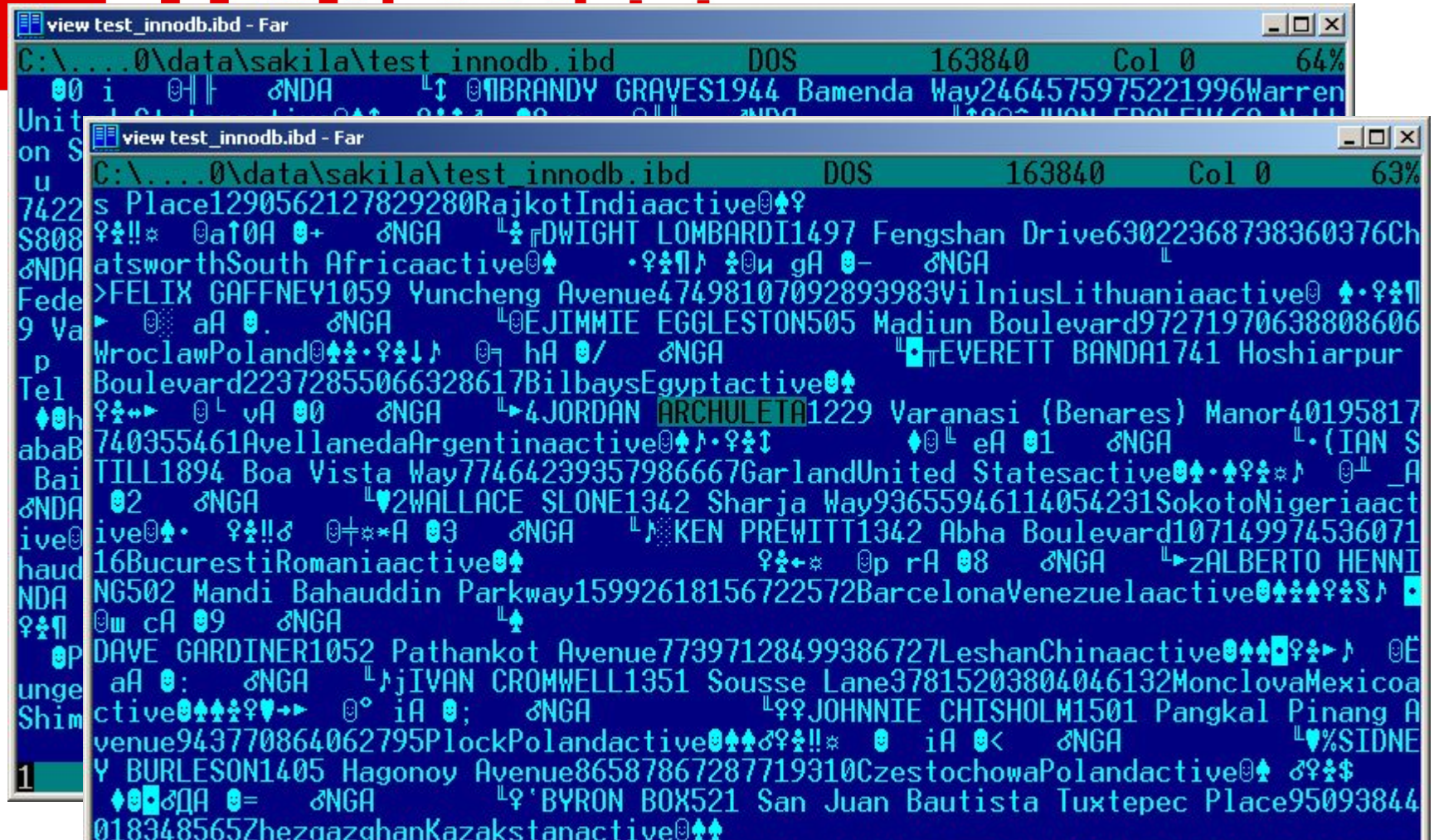

view test_myisam.MYD - Far
C:\....0\data\sakila\test_myisam.MYD DOS 52664 Col 0 20%
JOE GILLILAND↕953 Hodeida Street↨18841♂53912826864♦Imus♂Philippines♠active☻☺ U №
- ♬JANET PHILLIPS¶1718 Valencia Street↨37359↑67529216413♂Antofagasta↨Chile♠acti
ve☺♥ N☻ №┬ ♬KRISTEN CHAVEZ¶345 Oshawa Boulevard↨32114↑104491201771♦Hino↨Japan♠ac
tive☻ ☺ I №4☻↑BOB PFEIFFER☼415 Pune Avenue↨44274↑203202500108♠Xintai↨China♠acti
ve☻♥ l №й☺
FRANCIS SIKES←355 Vitria de Santo Anto Way↨81758↑548003849552→San Juan Bautista
Tuxtepec♠Mexico♠active☻♥ P №▪☺ BEN EASTER◄886 Tonghae Place↨19450↑711928348157♠
Kamyin↕Russian Federation ☻♥ c☺ №0☻►JORDAN ARCHULETA↔1229 Varanasi (Benares) Man
or↨40195↑817740355461 Avellaneda Argentina♠active☺ ♥ [☺ №
☺◄CASSANDRA WALTERS‼920 Kumbakonam Loop↨75090↑685010736240·Salinas
United States♠active☺ ♥ N☻ №┬☺
TOMMY COLLAZO‼76 Kermanshah Manor↨23343↑762361821578·Qomsheh♦Iran♠active☺ ☺] №
G☻♬MARSHALL THORN‼1584 Ljubertsy Lane↨22954↑285710089439♂Southampton♬United King
dom♠active☺♥ L №▪☺
BILLIE HORTON◄457 Tongliao Loop↨56254↑880756161823↨Inegl♠Turkey♠active☻♥ L №>☺♂
BRIAN WYMAN☼1769 Iwaki Lane↨25787↑556100547674 Bydgoszcz♠Poland♠active☺☺ Q №╠☺♂
LEON BOSTIC↕734 Tanshui Avenue↨70664↑366776723320 Florencia·Colombia♠activ
e☺♥ R☻ №ə☺
BRENT HARKINS◄319 Plock Parkway↨26101↑854259976812♂Sultanbeyli♠Turkey♠active☺ ☺
Q №r ♂GRACE ELLIS¶442 Rae Bareli Place↨24321↑886636413768·Duisburg·Germany♠acti
ve☻♥ W☺ №o☺ RAY HOULE►740 Udaipur Lane↨33505↑497288595103 Dzerzinsk↕Russia
n Federation♠active☺ ♥ T №d☺↑GERALD FULTZ←45 Aparecida de Goinia Place♦7431↑650
496654258↨Satna↨India♠active☻♥ P №!☻↑JULIO NOLAND‼182 Nukualofa Drive↨15414↑426
1 2 3 4 5Print 6 7 8Goto 9Video 10


view test_innodb.ibd - Far
C:\....0\data\sakila\test_innodb.ibd DOS 163840 Col 0 64%
1 2 3 4 5Print 6 7 8Goto 9Video 10

это, кстати, данные – без индексов

добавляем РК, и брюки превращаются…


view test_myisam.MYD - Far
C:\....0\data\sakila\test_myisam.MYD DOS 52664 Col 0 20%
JOE GILLILAND↕953 Hodeida Street♀18841♂53912826864♦Imus♂Philippines♠active
view test_myisam.MYD - Far
C:\....0\data\sakila\test_myisam.MYD DOS 53928 Col 0 20%
JOE GILLILAND↕953 Hodeida Street♀18841♂53912826864♦Imus♂Philippines♠active
♫JANET PHILLIPS¶1718 Valencia Street♀37359♀675292816413♂Antofagasta♀Chile
♠active
♫KRISTEN CHAVEZ¶345 Oshawa Boulevard♀32114♀104491201771♦Hino♀
Japan♠active
♀BOB PFEIFFER☼415 Pune Avenue♀44274♀203202500108♠Xintai♀C
hina♠active
FRANCIS SIKES←355 Vitria de Santo Anto Way♀81758♀548003849552→San Juan Bautista
Tuxtepec♠Mexico♠active
BEN EASTER◄886 Tonghae Place♀19450♀711928348
157♠Kamyin↕Russian Federation
►JORDAN ARCHULETA↔1229 Varanasi (Benar
es) Manor♀40195♀817740355461 Avellaneda Argentina♠active
◄CASSANDRA WALTERS‼920 Kumbakonam Loop♀75090♀685010736240•Salinas
United States♠active
TOMMY COLLAZO‼76 Kermanshah Manor♀23343♀762361821578•Qomsheh♦Iran♠active
♫MARSHALL THORN‼1584 Ljubertsy Lane♀22954♀285710089439♂Southampton♫United Kin
gdom♠active
BILLIE HORTON◄457 Tongliao Loop♀56254♀880756161823♀Inegl♠Turkey♠active
♂BRIAN WYMAN☼1769 Iwaki Lane♀2578♀556100547674 Bydgoszcz♠Poland♠active
♂LEON BOSTIC↕734 Tanshui Avenue♀70664♀366776723320 Florencia•Colomb
ia♠active
BRENT HARKINS◄319 Plock Parkway♀26101♀854259976812♂Sultanbeyli♠Turkey♠active
♂GRACE ELLIS¶442 Rae Bareli Place♀24321♀886636413768•Duisburg•Germany♠act
ive
RAY HOULE►740 Udaipur Lane♀33505♀497288595103 Dzerzinsk↕Russia
n Federation♠active
♀GERALD FULTZ←45 Aparecida de Goinia Place♦7431♀65


view test_innodb.ibd - Far
C:\....0\data\sakila\test_innodb.ibd DOS 163840 Col 0 64%
view test_innodb.ibd - Far
C:\....0\data\sakila\test_innodb.ibd DOS 163840 Col 0 63%

почему так?
разные стратегии хранения строк

MyISAM – в порядке поступления (в конец файла)

InnoDB – хранит постранично, внутри странички – в порядке PK

HighLoad++

☐		✕	560	JORDAN ARCHULETA	1229 Varanasi (Benares) Manor	40195	817740355461
☐		✕	561	IAN STILL	1894 Boa Vista Way	77464	239357986667
☐		✕	562	WALLACE SLONE	1342 Sharja Way	93655	946114054231

(InnoDB, **после** добавления PK)

MyISAM – “обычное” хранение

InnoDB – т.н. “кластерное”

умеем хранить –
теперь нужно **быстро** искать!

SELECT * FROM users WHERE id=123

SELECT * FROM users WHERE lastname='Pupkin'

SELECT * FROM users WHERE lastname LIKE ‘Pu%’

SELECT * FROM goods WHERE MATCH('ipod') ORDER BY price ASC

SELECT * FROM users WHERE sex='F' AND age>=18 AND age<=25

HighLoad++

как выполнять запрос?

полный перебор? мееедленно

нас спасут…

HighLoad++

индексы!

алгоритмы уже спешат на помощь!

смысл **любого** вида индекса?

быстрый поиск по ключу(-ам)

HighLoad++


Grand Wailea Resort.
HOTEL & SPA
Maui
Grand Wailea Resort.
HOTEL & SPA
Maui
General Information
Address
Phone
Reservations
Concierge
Dining at the Grand Wailea
Spa Grande

HighLoad++

HighLoad++


ALICIA KEYS
AS I AM

видов индексов – тоже много

hash index

R-tree index

full-text index

индекс общего назначения –
типично **B-tree**

поиск – по равенству,
диапазону
(чисел, строк, и т.п.)

дискует – страничками
(хорошо!)

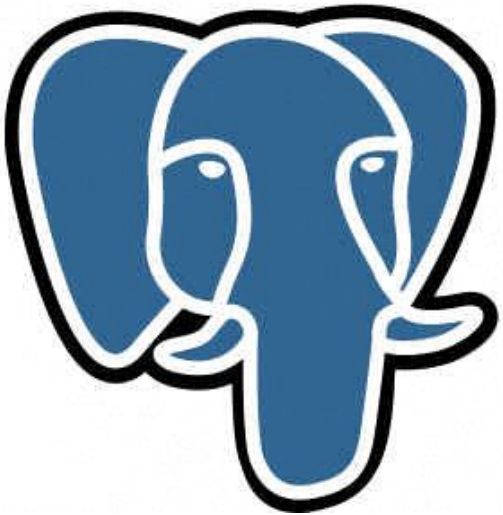
используется – всеми


hl++
HighLoad++

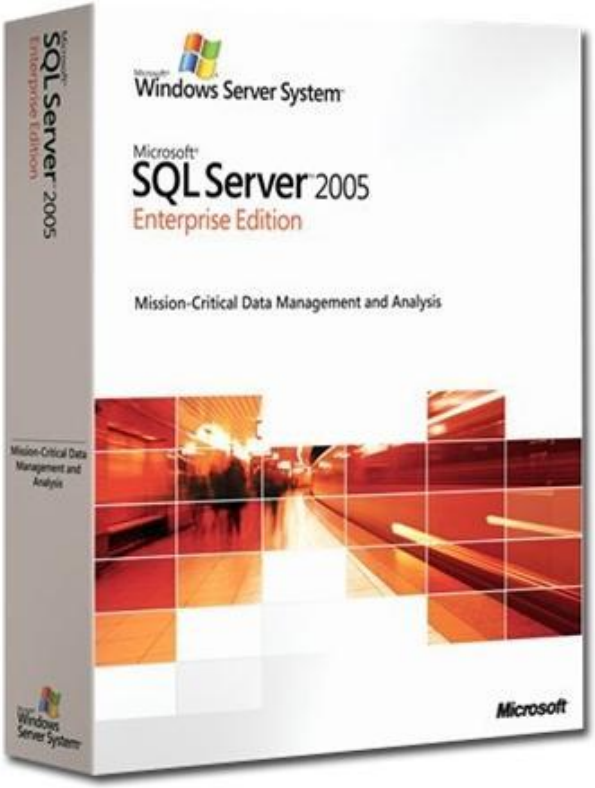

MySQL®

HighLoad++

PostgreSQL



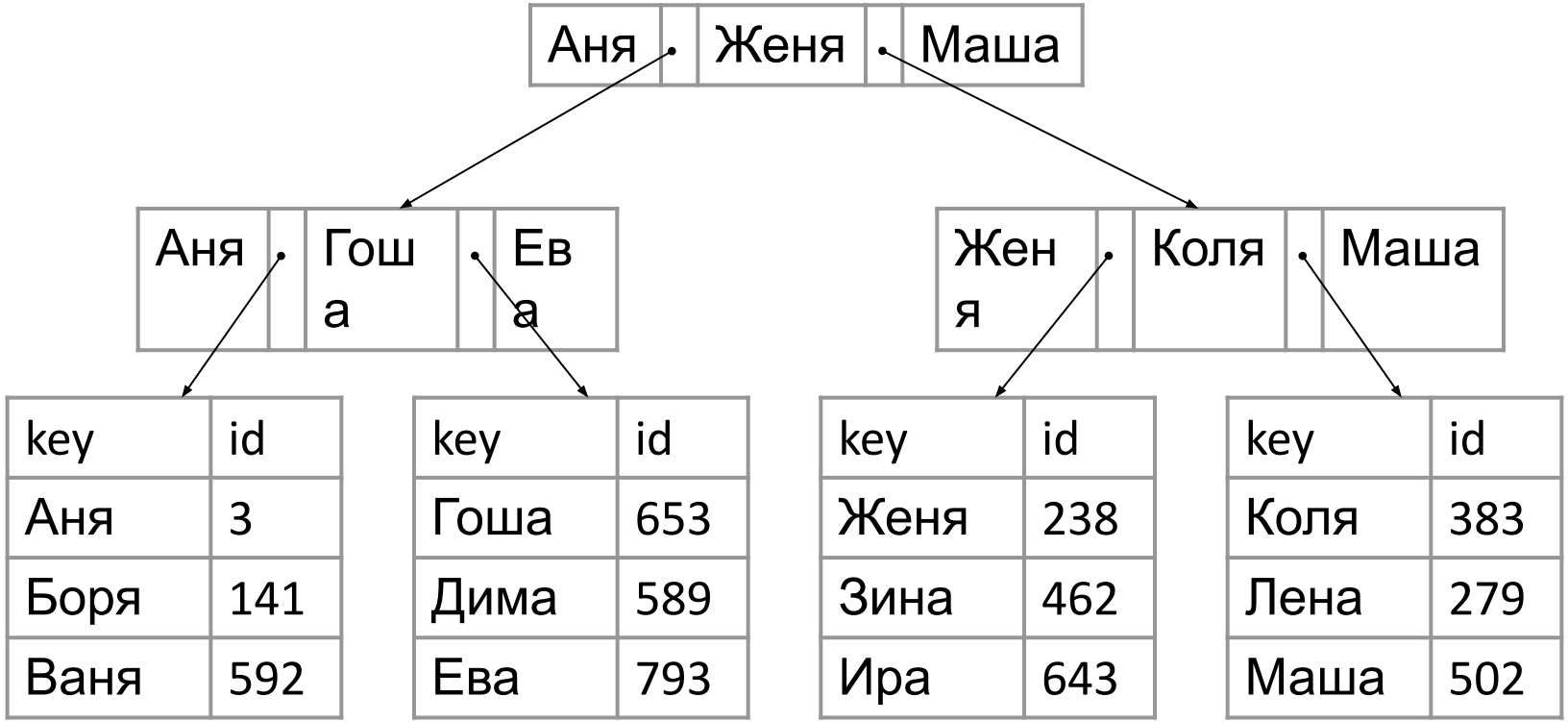
HighLoad++


Windows Server System
Microsoft
SQL Server 2005
Enterprise Edition
Mission-Critical Data Management and Analysis
Microsoft

HighLoad++

используется несмотря ни на что!!!

как устроено?



(B-дерево; странички; масштаб 1:72)

два вида страничек

Промежуточные = ключи + указатели на другие странички

{ key1, ptr1, key2, ptr2, …, keyN }

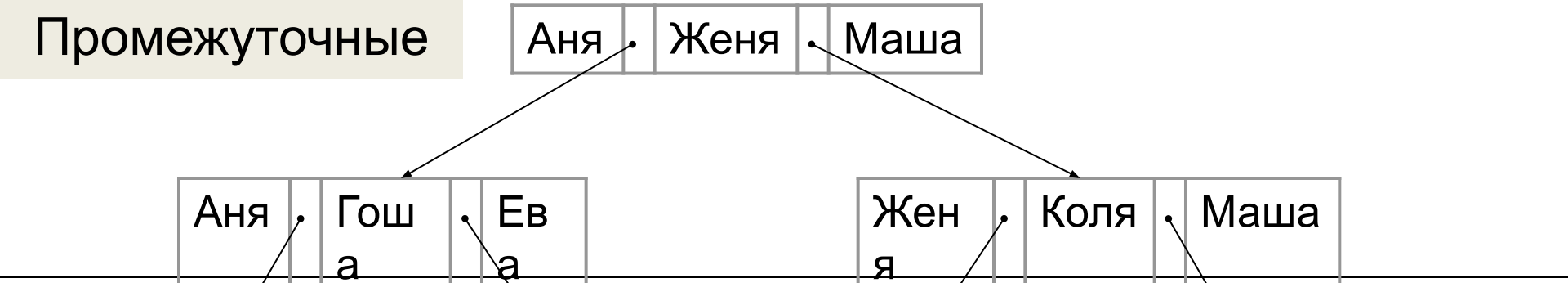
Листовые = ключи + соотв-е им данные (eg. row_offset)

key	id
Аня	3
Боря	141
Ваня	592

Аня	3	Боря	141	Ваня	592

{ key1, data1, key2, data2, … }

Промежуточные



key	id
Аня	3
Боря	141
Ваня	592

key	id
Гоша	653
Дима	589
Ева	793

key	id
Женя	238
Зина	462
Ира	643

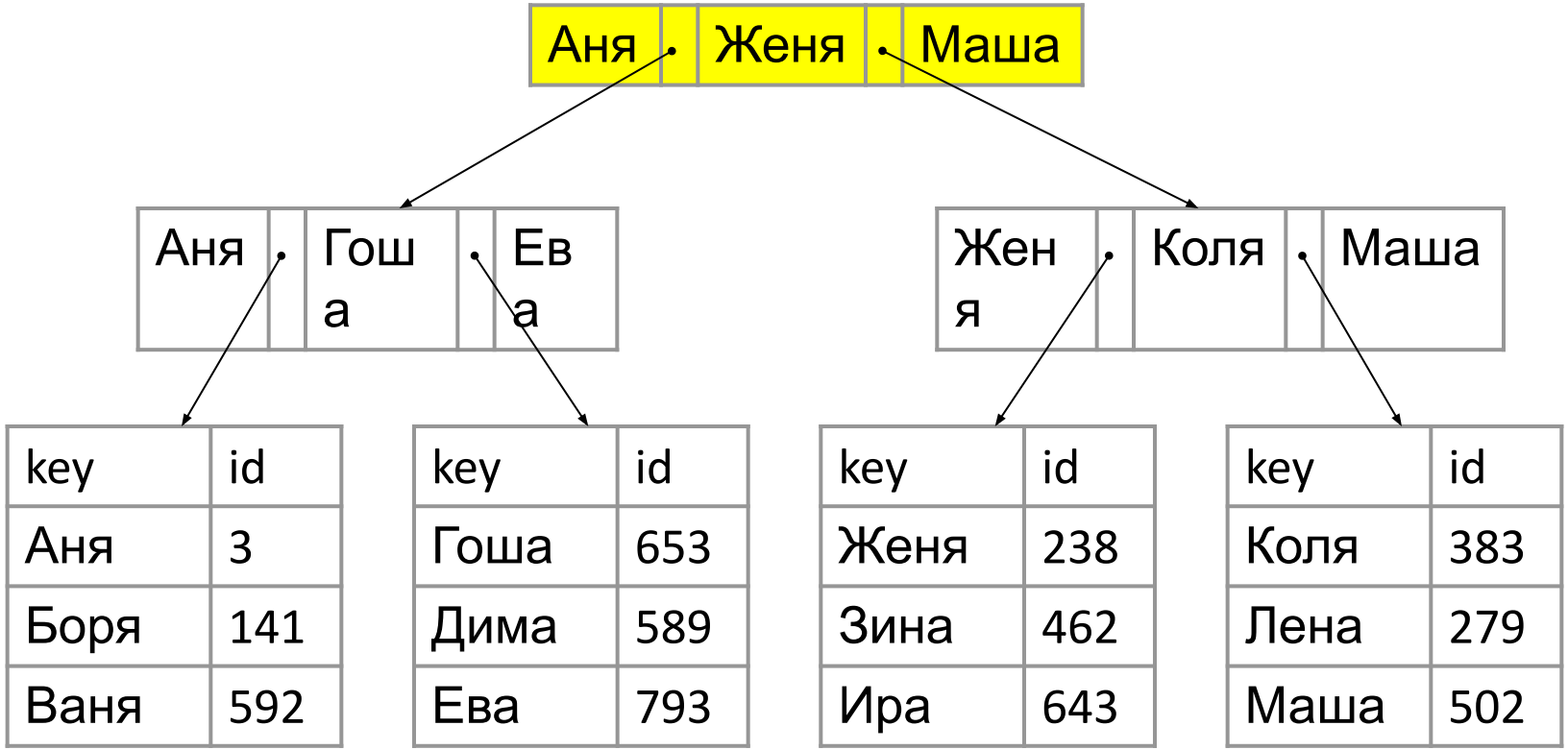
key	id
Коля	383
Лена	279
Маша	502

Листовые

почему все используют этот ужас?!

во-1х – легко искать по ключу

пример – ищем Зину

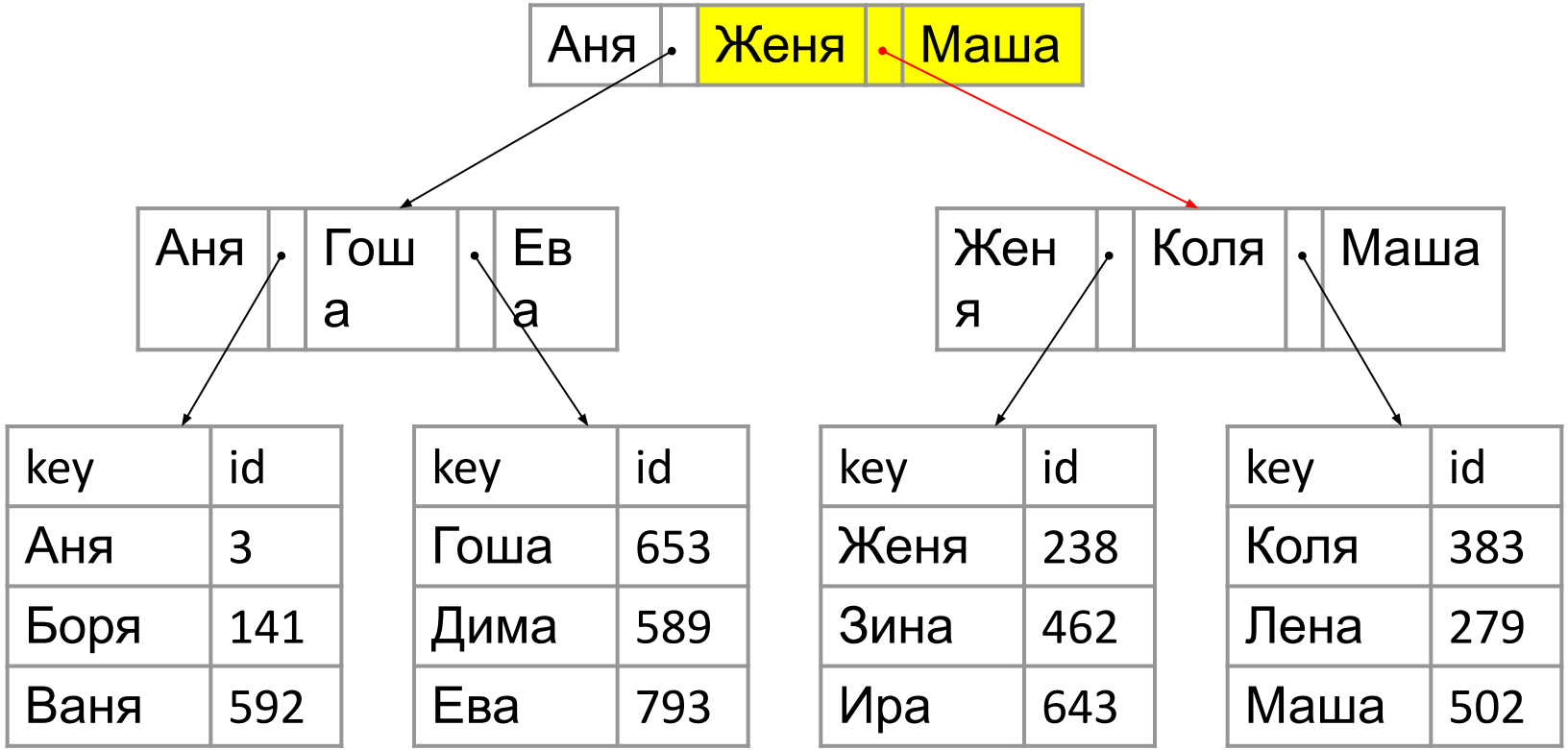


key	id
Аня	3
Боря	141
Ваня	592

key	id
Гоша	653
Дима	589
Ева	793

key	id
Женя	238
Зина	462
Ира	643

key	id
Коля	383
Лена	279
Маша	502

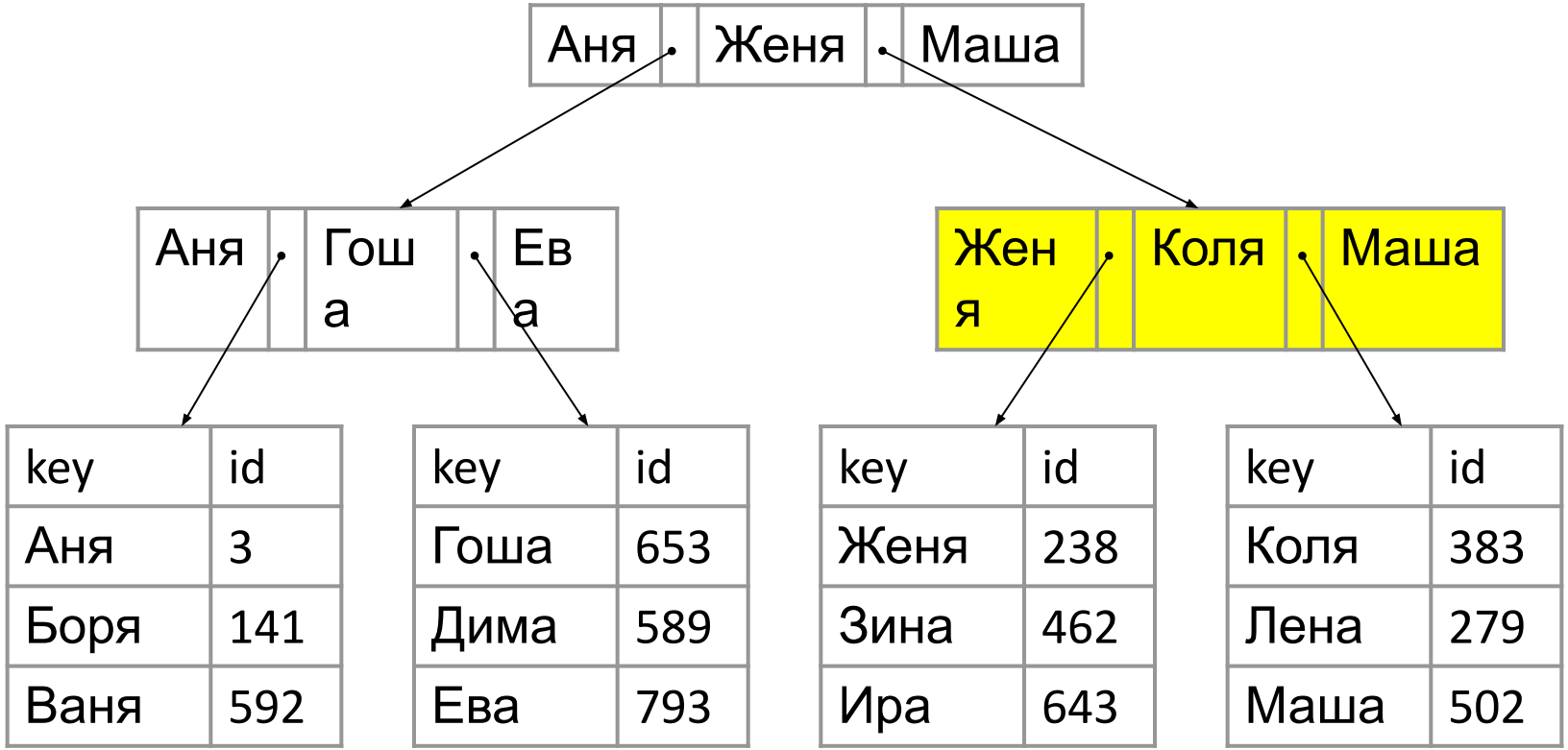


key	id
Аня	3
Боря	141
Ваня	592

key	id
Гоша	653
Дима	589
Ева	793

key	id
Женя	238
Зина	462
Ира	643

key	id
Коля	383
Лена	279
Маша	502

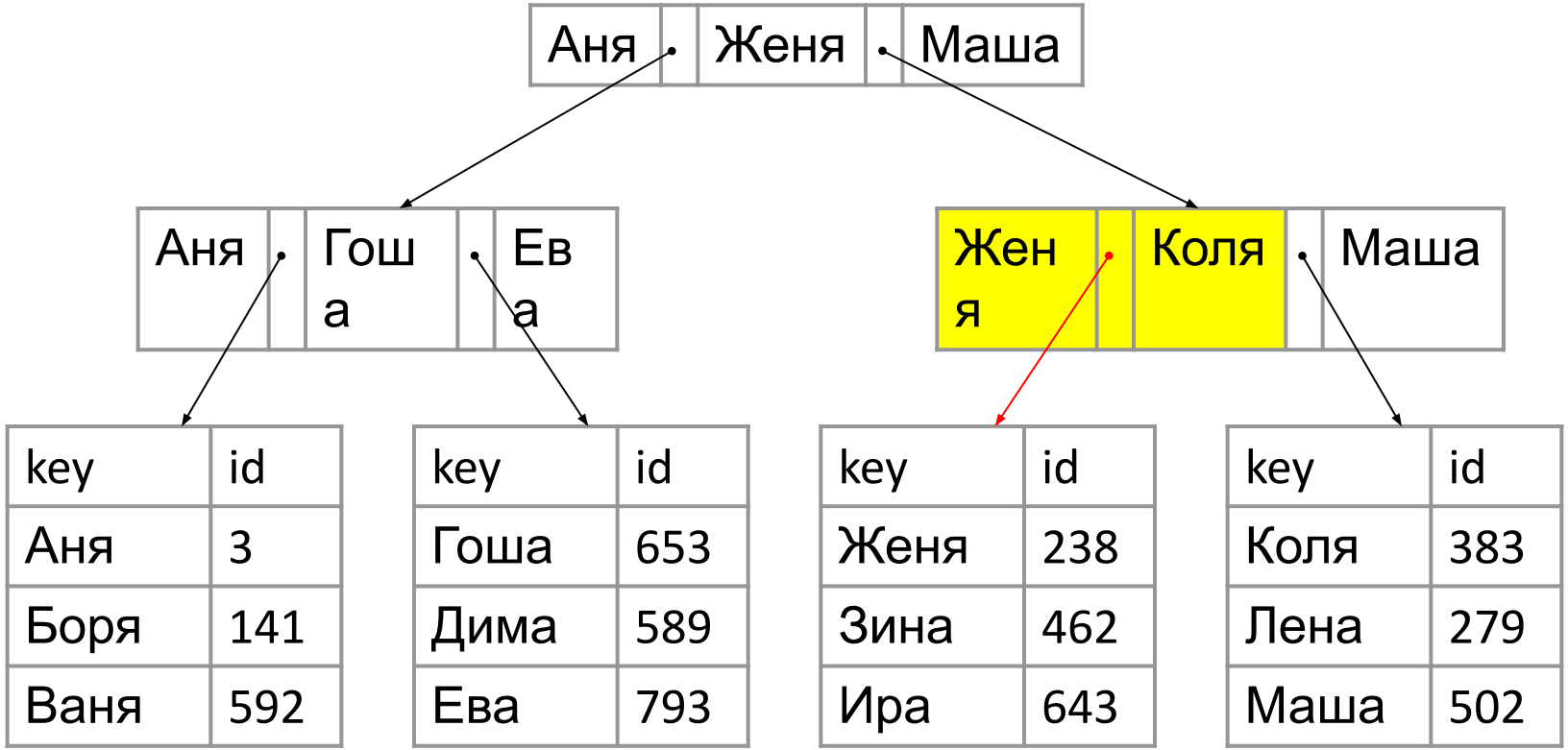


key	id
Аня	3
Боря	141
Ваня	592

key	id
Гоша	653
Дима	589
Ева	793

key	id
Женя	238
Зина	462
Ира	643

key	id
Коля	383
Лена	279
Маша	502

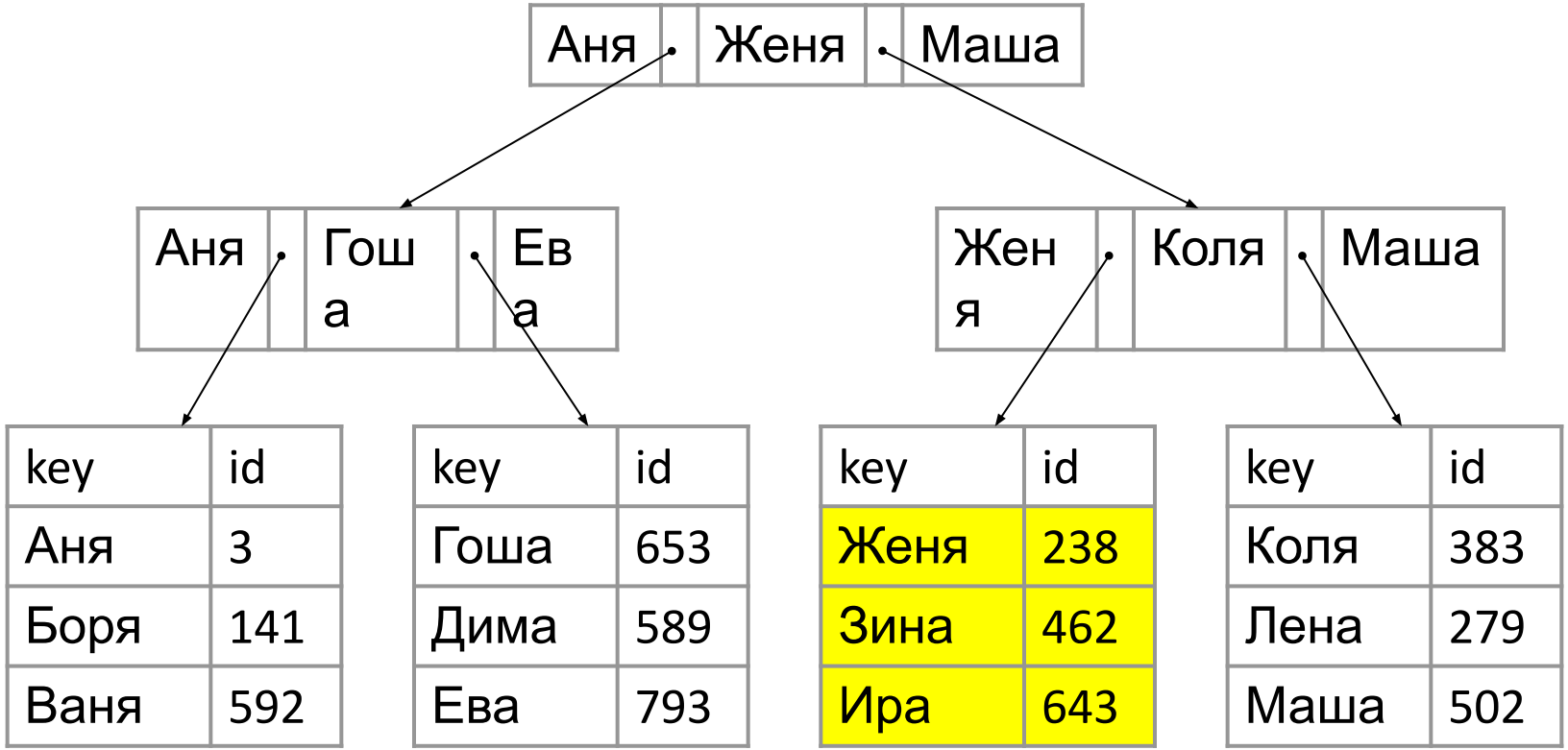


key	id
Аня	3
Боря	141
Ваня	592

key	id
Гоша	653
Дима	589
Ева	793

key	id
Женя	238
Зина	462
Ира	643

key	id
Коля	383
Лена	279
Маша	502

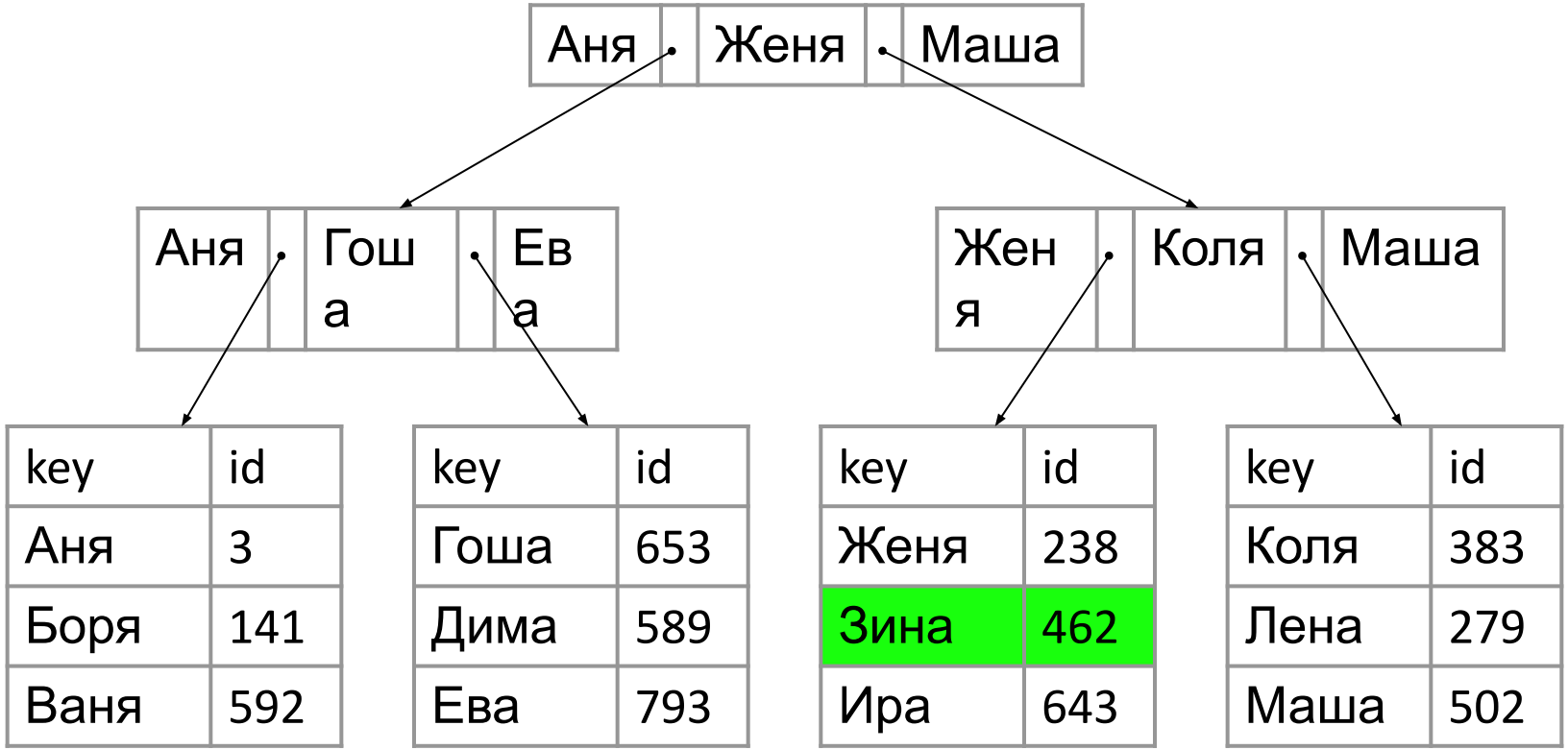


key	id
Аня	3
Боря	141
Ваня	592

key	id
Гоша	653
Дима	589
Ева	793

key	id
Женя	238
Зина	462
Ира	643

key	id
Коля	383
Лена	279
Маша	502


Аня
Женя
Маша
Аня
Гоша
Ева
Женя
Коля
Маша
key
id
Аня
3
Боря
141
Ваня
592
key
id
Гоша
653
Дима
589
Ева
793
key
id
Женя
238
Зина
462
Ира
643
key
id
Коля
383
Лена
279
Маша
502

ура – Зина нашлась!!!

хорошо – поиск работает…

…но он чё, всегда такой резкий?

HighLoad++

– В жизни – под 1000 (а не 3) записей на страничку
– Два уровня страничек – 1000*1000 – **миллион**
– Три уровня – **миллиард**…
– Итого **2-3 странички max** – практически всегда

Аня	3
Боря	141
Ваня	592

Гоша	653
Дима	589
Ева	793

Женя	238
Зина	462
Ира	643

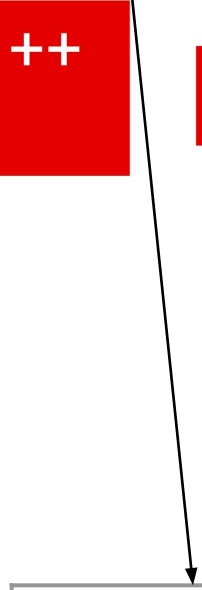
Коля	383
Лена	279
Маша	502

почему все используют этот ужас?!

во-2х – легко обновлять

key	id
Аня	3
?	?
?	?

странички обычно НЕ полны

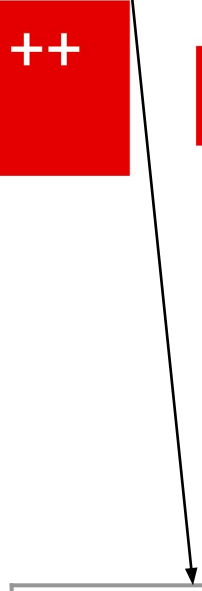


key	id
Аня	3
Боря	141
?	?

вставляем…

key	id
Аня	3
Боря	141
Ваня	592

вставляем…



key	id
Аня	3
Боря	141
Ваня	592
Гоша	653

вставляем… оп-па, некуда!!

создаем новую страничку

key	id
Аня	3
Боря	141
Ваня	592
Гоша	653

key	id
?	?
?	?
?	?
?	?

создаем новую страничку

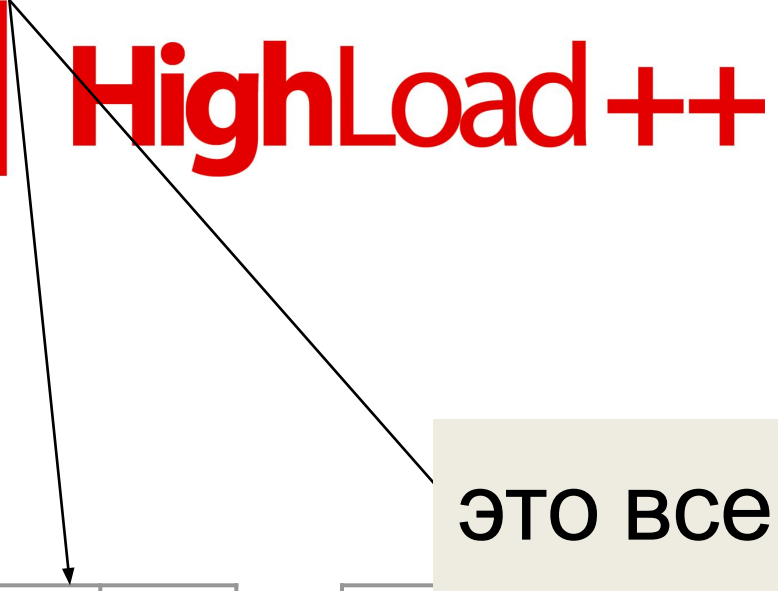
key	id
Аня	3
Боря	141
?	?

key	id
Ваня	592
Гоша	653
?	?

…и суем “ее” в родителя

key	id
Аня	3
Боря	141
?	?

key	id
Ваня	592
Гоша	653
?	?



key	id
Аня	3
Боря	141
?	?

key	
Ваня	592
Гоша	653
?	?

это все – тоже трогаем max

2-3 странички

hl++ Hig

HighLoad++

B-tree Kung Fu!!!

вернемся к запросам?

SELECT * FROM users WHERE id=123

1. “Ищем Зину” (rowoffset по id=123)
2. seek(rowoffset) в файле строк (.MYD)
3. read(rowdata) из файла
4. и… все – результат готов

усложним – добавим условий

SELECT * FROM users WHERE sex='F' AND age>=18 AND age<=25

индекс “в лоб” по sex?

key	id
F	12345
F	12346
F	12347
F	12348
F	12349
F	12350
F	12351
F	12352

key	id
F	12353
F	12354
F	12355
F	12356
F	12357
F	12358
F	12359
F	12360

key	id
F	12361
F	12362
F	12363
F	12364
F	12365
F	12366
F	12367
F	12368

key	id
F	1
F	1
F	1
F	1
F	1
F	1
F	1
F	1

key	id
F	12345
F	12346
F	12347
F	12348
F	12349
F	12350
F	12351
F	12352

key	id
F	12353
F	12354
F	12355
F	12356
F	12357
F	12358
F	12359
F	12360

key	id
F	12361
F	12362
F	12363
F	12364
F	12365
F	12366
F	12367
F	12368

key	i
F	1
F	1
F	1
F	1
F	1
F	1
F	1
F	1

они ВСЕ подходят по условию ‘F’!

HighLoad++

…и нам надо прочитать с диска (!) 5,000,000+ строк…

…и для каждой лично проверить паспорт и age>=18 and age<=25?!

key	id
F	12345
F	12346
F	12347
F	12348
F	12349
F	12350
F	12351
F	12352

	id
F	12353
F	12354
F	123[illegible]
F	[illegible]
F	[illegible]357
F	12358
F	12359
F	12360

	id
F	12361
F	12362
F	[illegible]2363
F	[illegible]2364
F	[illegible]2365
F	12366
F	12367
F	12368

key	id
F	1
F	1
F	1
F	1
F	1
F	1
F	1
F	1

неселективный индекс –

косяк и западло!

sex=‘F’ AND age>=18 AND age<=25

индекс “по лбу” по age?

key	id
13	12345
13	12346
13	12347
14	12348
15	12351
16	12352

key	id
17	12353
18	12354
18	12355
18	12356
21	12359
22	12360

key	id
25	12361
25	12362
27	12363
29	12364
53	12367
64	12368

но – вдруг это мужики?!

мужики нам не нужны!!!

либо опять читать **ненужные** строки (мужиков) – либо…

покрывающий (covering)
индекс
по **обоим** полям

key	id
F, 13	12345
F, 17	12346
F, 19	12347
F, 21	12348
F, 21	12349
F, 22	12350
F, 23	12351
F, 23	12352

key	id
F, 25	12353
F, 27	12354
F, 31	12355
F, 33	12356
F, 42	12357
M, 11	12358
M, 13	12359
M, 17	12360

key	id
M, 19	12361
M, 23	12362
M, 29	12363
M, 31	12364
M, 37	12365
M, 41	12366
M, 43	12367
M, 47	12368

список нужных строк – ясен сразу

чтений с диска – минимум
скорости – максимум

бонус – сортировка по age

кстати, про сортировку…

SELECT * FROM users WHERE sex='F' AND age>=18 AND age<=25 **ORDER BY hotness DESC LIMIT 10**

как выполнять? есть варианты

HighLoad++

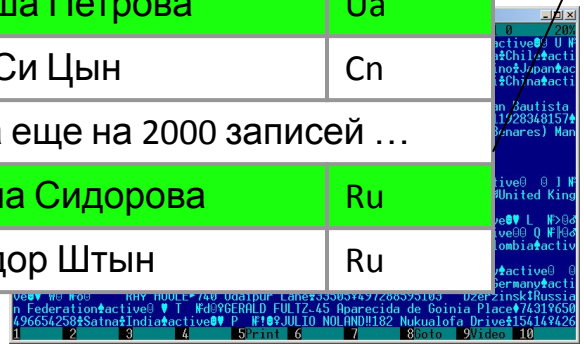
налево – read_index(WHERE) + read_rows + sort_rows(ORDER)

индекс

key	id
F, 13	12345
F, 17	12346
F, 19	12347
F, 21	12348
F, 21	12349
F, 22	12350
F, 23	12351
F, 27	12354

данные

id	name	Country
12350	Лена Иванова	Ru
24523	Шараф Худайбердыев	Uz
… дырка на 1000 записей …		
24624	Мацал Кошек	Cz
12347	Маша Петрова	Ua
80356	Ли Си Цын	Cn
… дырка еще на 2000 записей …		
12351	Нина Сидорова	Ru
10756	Федор Штын	Ru



сортировка

результат

read_index – мало и быстро

10K*(1+4+8 bytes) = 130 KB

5-10 ms/seek, 50+ MB/s read

read_**random**_rows – медленно!

10K*5-10 ms/seek = 50-100 sec…

sort_rows – обычно быстро

10K*0.1-1 KB/row = 1-10 MB (in RAM)

мораль – все зло от random rows!

(еще от sort_rows, если их много)

HighLoad++

… sex='F' AND age>=18 AND age<=25
ORDER BY hotness DESC LIMIT 10

направо –
read_**fat**_index(WHERE) +
sort_index(ORDER) + LIMIT +
read_**less**_rows + sort_rows

нужен утолщенный
INDEX (sex, age, hotness)

вместе с поиском по sex, age – сразу узнаем hotness (+40 KB)

sort (10K * [hotness, rowptr])

read_rows – почти не нужен
(10 строк результата…)

sort_rows – вообще не нужен

PROFIT?

не панацея – даже в теории

INDEX (sex, age, hotness)
WHERE sex='F' ORDER BY hotness LIMIT 10

в теории – обработать 50%

индекса

затем – прочитать 10 строк (пф!)

INDEX (sex, age, hotness)

1M rows, ~20 MB, 50% = ~10 MB

INDEX (sex, age, hotness)

WHERE sex='F' AND hotness>0 ORDER BY age LIMIT 10

в теории – читаем **индекс**
линейно –
пока не заполним limit

что на практике?

HighLoad++

welcome to real world

```
CREATE TABLE usertest (
    id INTEGER PRIMARY KEY NOT NULL,
sex ENUM ('m','f'),
    age INTEGER NOT NULL,
    hotness INTEGER NOT NULL,
    name VARCHAR(255) NOT NULL,
INDEX(sex,age,hotness) )
```

500,000 записей
56 MB данных (.ibd файл)
32 MB innodb_buffer_pool

age \in [18.. 80]
hotness \in [-5.. 5]

```
mysql> explain select * from usertest
    where sex='f' and age>=18 and age<=25
    order by hotness desc limit 10 \G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: usertest
         type: ref
possible_keys: sex
          key: sex
      key_len: 2
          ref: const
         rows: 25119
        Extra: Using where; Using filesort
1 row in set (0.00 sec)
```

filesort – **НЕ** про временный файл

filesort – про “сортировку строк”

Using where – проверка условия **НЕ** по индексу – ?!!

запрос проще, **точно** по индексу?

```
mysql> explain select * from usertest
    where sex='f' and age=18
    order by hotness desc limit 10 \G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: usertest
         type: ref
possible_keys: sex
          key: sex
      key_len: 6
          ref: const,const
         rows: 10386
        Extra: Using where (bug #30733, 30 aug 2007?)
1 row in set (0.00 sec)
```

проверим экспериментально!!!

```
mysql> select * from usertest where sex='f' and
    age>=18 and age<=25
    order by hotness desc limit 10;
...
10 rows in set (23.05 sec)

mysql> select * from usertest where sex='f' and
    age=18
    order by hotness desc limit 10;
...
10 rows in set (0.05 sec)
```

причина?

```
mysql> explain select * from usertest
    where sex='f' and age>=18 and age<=25
    order by hotness desc limit 10 \G
*************************** 1. row ***************************
           id: 1
  select_type: SIMPLE
        table: usertest
         type: ref
possible_keys: sex
          key: sex
      key_len: 2      ← используется только начало, поле "sex"!!!
          ref: const
         rows: 25119
        Extra: Using where; Using filesort        ← так и есть :(
1 row in set (0.00 sec)
```

MySQL **не умеет** сортировать элементы **индекса** :(

сортировка “по индексу” –
только если индекс
гарантирует порядок

1) в куске индекса sex=F,
18<=age<=25
порядок hotness desc НЕ
гарантирован

2) optimizer лажанул, 18<=age<=25 считается НЕ по индексу (а могло бы)

(теория говорит – можно лучше!)

проверяем дальше!

```
mysql> explain select * from usertest
    where sex='f' order by hotness desc limit 10 \G
...
          key: sex
      key_len: 2
          ref: const
         rows: 226072
        Extra: Using where; Using filesort


mysql> explain select * from usertest
    where sex='f' order by hotness desc limit 10 \G
...
10 rows in set (20.25 sec)
```

и последний запрос

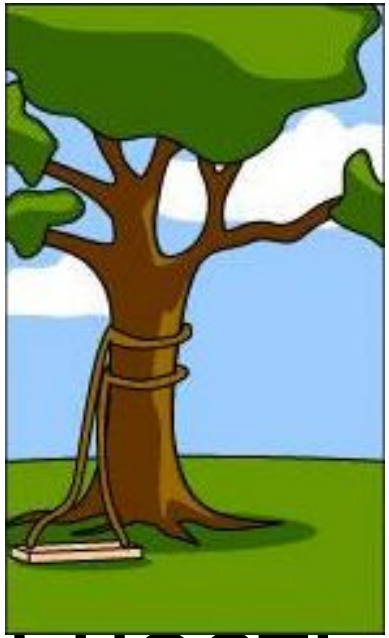
```
mysql> explain select * from usertest where sex='f' and hotness>0
order by age asc limit 10 \G
...
          key: sex
      key_len: 2
          ref: const
         rows: 226072
        Extra: Using where; Using filesort


mysql> select * from usertest where sex='f' and hotness>0 order by
age asc limit 10;
...
10 rows in set (0.25 sec)
```

ИТОГО

Sex	Age	Hotness	Time	Optimal
Const=F	Range=18..25	Order	23.05	HELL NO
Const=F	Const=18	Order	0.05	OK
Const=F	–	Order	20.25	HELL NO
Const=F	Order	Cond>0	0.25	KINDA OK

теория была оптимистична…



…реальность внесла коррективы.

не учли **детали реализации**

1. нету “досортировки” индекса (MySQL specific)

2. limit обрабатывается… так себе (MySQL specific)

3. optimizer ошибается (везде и у всех)

HighLoad++

про ошибки optimizer и спасительный full-scan

```
mysql> select * from usertest where sex='f' order by hotness desc
limit 10;
...
10 rows in set (20.25 sec)
```

```
mysql> select * from usertest ignore index(sex) where sex='f'
order by hotness desc limit 10;
...
10 rows in set (0.55 sec)
```

10,000 x 10 ms = 100 sec

100,000 x 1KB / 50 M/s = 2 sec

мораль: random IO **очень** плохо
(водка яд водка яд водка яд)

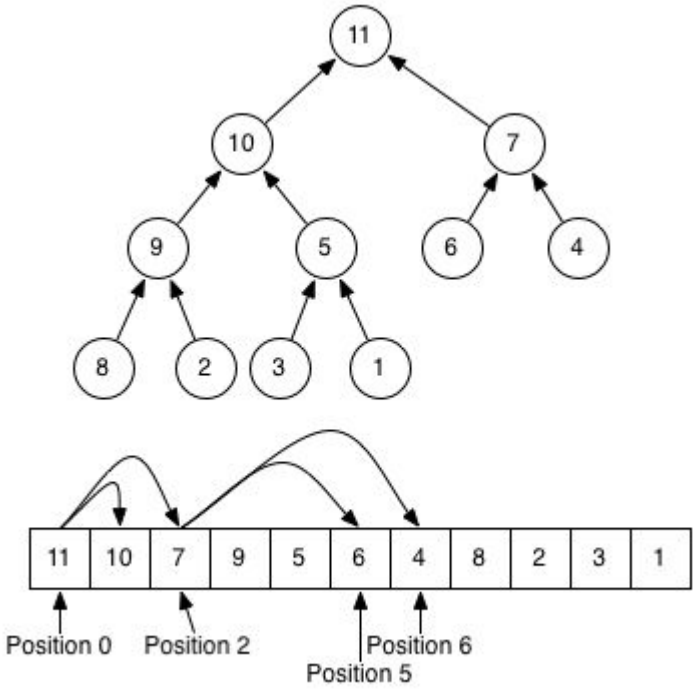
про “обработку” limit

теория – приоритетная очередь

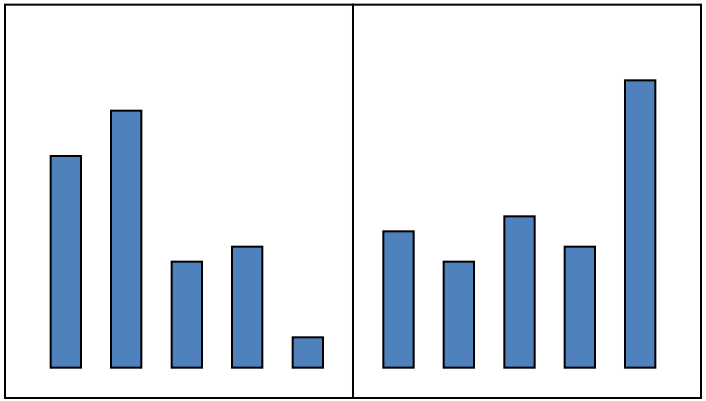
HighLoad++

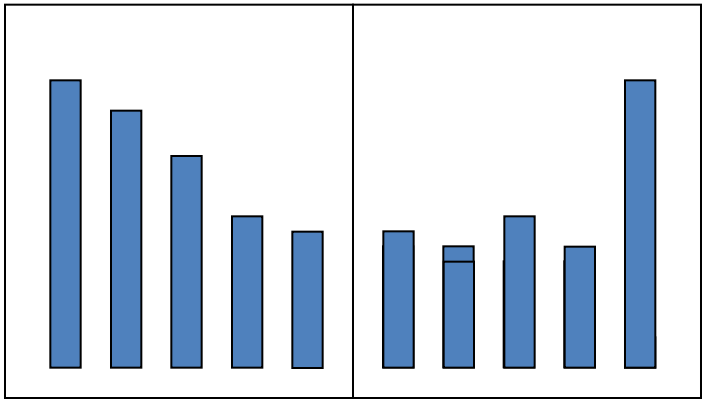
приоритетная!!

технически – heap


11
10
7
9
5
6
4
8
2
3
1
11 10 7 9 5 6 4 8 2 3 1
Position 0
Position 2
Position 5
Position 6

или просто double buffer





ключевое свойство –
в памяти храним только top-N

LIMIT 10 – надо хранить 10 строк

LIMIT 130,10 – надо 140

практика – MySQL vs. LIMIT

выбрать и отсортировать **ВСЕ** (*)

* – всегда, когда индекс не гарантирует точный порядок

выбрать OK – избежать нельзя

HighLoad++

сортировать **все** плохо…

HighLoad++

лишний удар по CPU/RAM/IO :(

как убирать mysql сортировку?

строить более другие индексы

ставить более другой софт

умеет и “обычный” поиск!

трюки про WHERE вместо LIMIT

(я не пробовал, но говорят, возможно)

…именно в таком порядке.

более практический пример?

HighLoad++

импортируем дамп Wikipedia

XML дамп → 2 толстые таблицы

хочется – а) одну б) тонкую!

```
INSERT INTO mycontent
SELECT t.old_id, p.page_id,
UNIX_TIMESTAMP(p.page_touched), p.page_len,
p.page_title, COMPRESS(t.old_text)
FROM text t, page p
   WHERE t.old_id=p.page_latest
   AND page_namespace=0 AND page_is_redirect=0;
```

15 GB text, 0.5 GB page, ~4.5M rows

tps=~200, bi/bo=~1 MB/sec
~200 MB .MYD in ~20 mins, ETA 10+ hrs

```
mysql> EXPLAIN SELECT t.old_id, ... \G
*********************** 1. row ***********************
          table: page
           type: ref
            key: name_title
            ref: const
           rows: 4435392
          Extra: Using where
*********************** 2. row ***********************
          table: text
           type: eq_ref
            key: PRIMARY
            ref: wiki.p.page_latest
           rows: 1
```

что хотим?

scan 15 GB text, join 0.5 GB page

почему не выходит?

… FROM text t, page p WHERE t.old_id=p.page_latest

решение – index(page_latest)

еще **пришлось** STRAIGHT_JOIN
(optimizer опять лажанул!)

результат – 40 минут, включая CREATE INDEX

hl++ HighLoad++

так зачем же знать алгоритмы?

“did we learn something today?”

HighLoad++

как устроено B-дерево

как работает индекс

как работают выборки

зачем нужны full-scans

как работает сортировка с LIMIT

чего можно добиться в идеале – в теории

…и как оно, бывает, не работает – на практике!

а толку?!

чего ждать от БД

HighLoad++

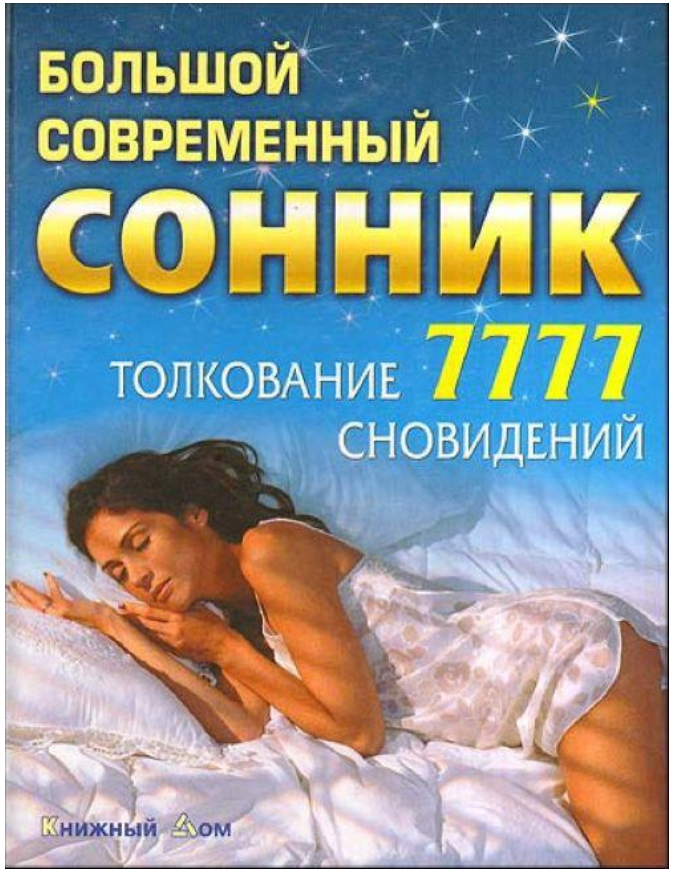
чего **не** ждать

HighLoad++

как и что тестировать

как объяснять потом результаты

HighLoad++


БОЛЬШОЙ
СОВРЕМЕННЫЙ
СОННИК
ТОЛКОВАНИЕ 7777
СНОВИДЕНИЙ
Книжный Дом

в итоге –
как заставлять таки работать

HighLoad++

…**БЫСТРО** работать.

“это все” (с) вопросы?!