

hl⁺⁺

HighLoad⁺⁺

Зачем знать алгоритмы

Андрей Аксенов
Sphinx Technologies Inc.

Highload++2009

hl⁺⁺

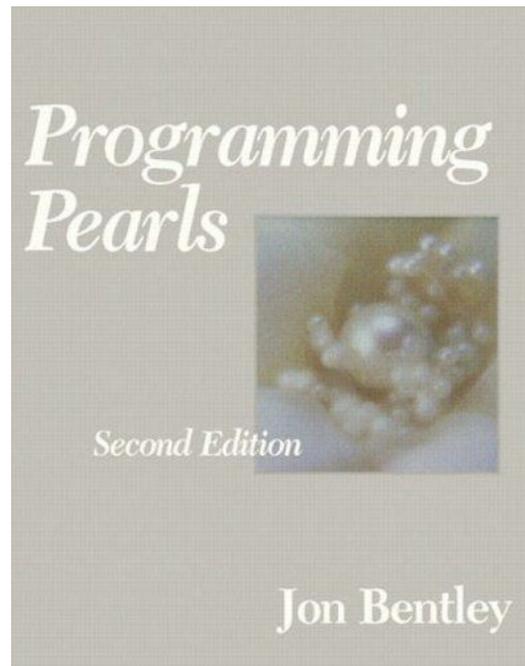
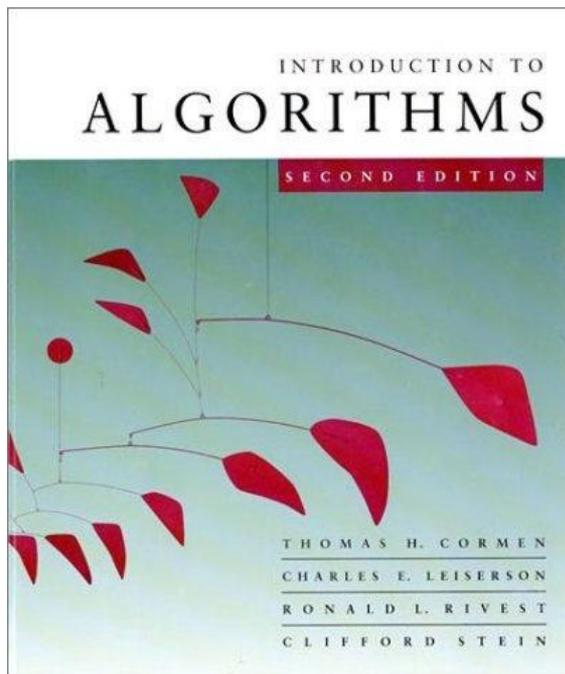
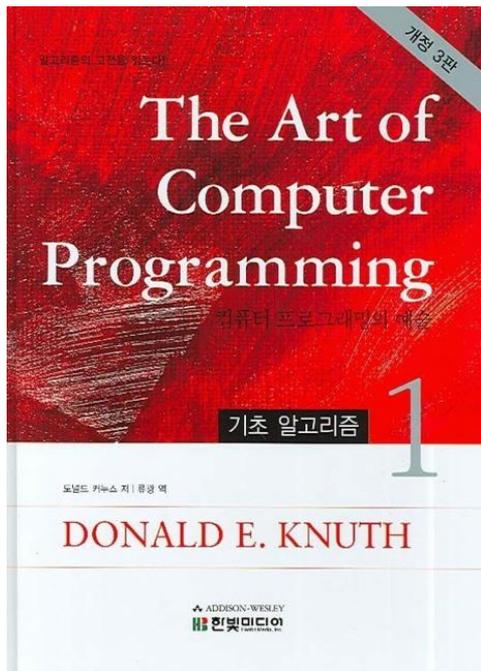
HighLoad⁺⁺



Who is Mr. Aksenov?

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

ЧЕСТНО ЛИСТАЛ ВСЕ!

hl⁺⁺

HighLoad⁺⁺

не прочитал ни одной :(

hl⁺⁺

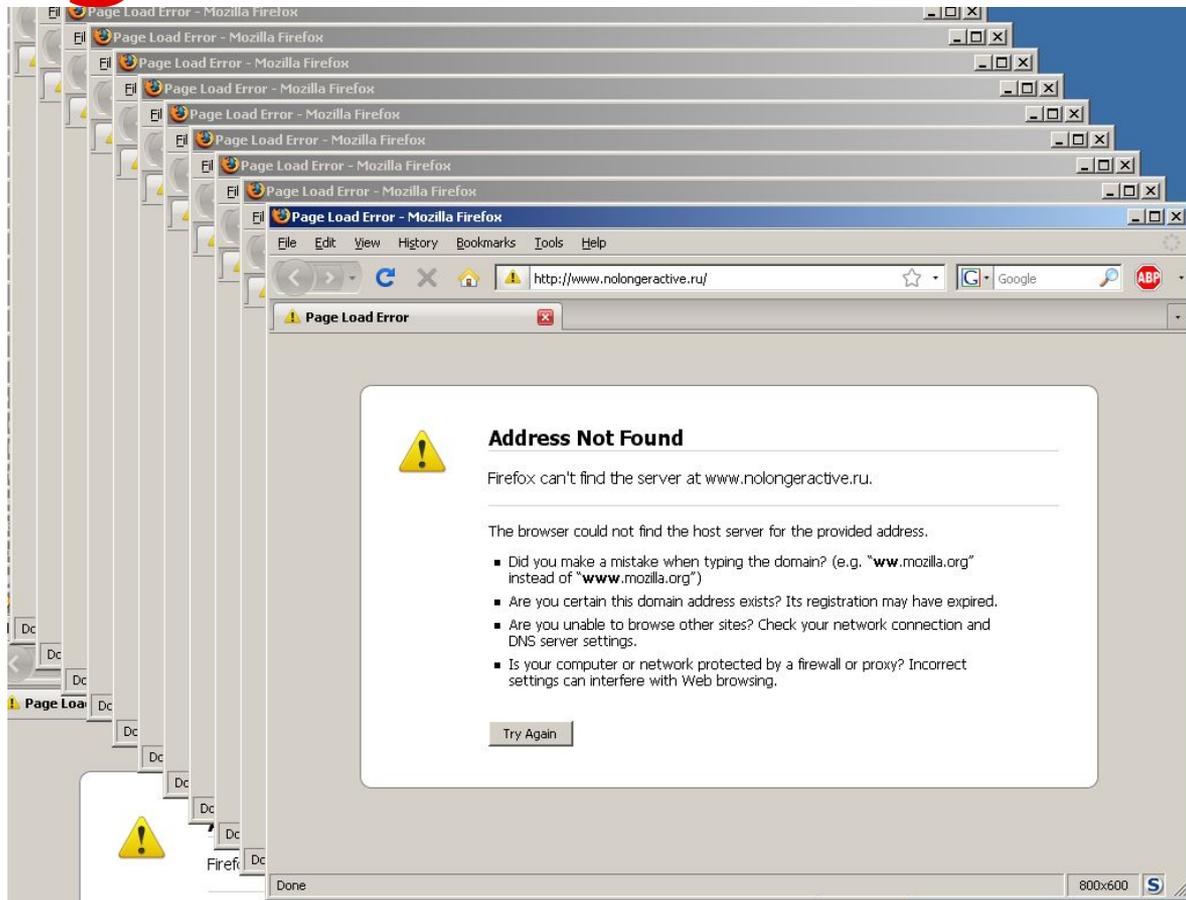
HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

делал веб-сайты и веб-движок



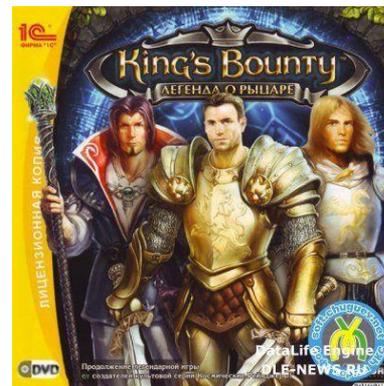
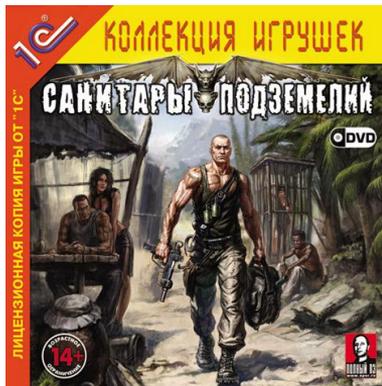
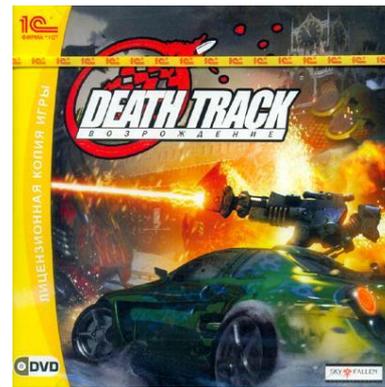
hl⁺⁺

HighLoad++

делал игры и 3D движок

hl++

HighLoad++



hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad++

делаю поисковой движок

hl⁺⁺

HighLoad⁺⁺



free open source search

hl⁺⁺

HighLoad⁺⁺

 Sphinx <<

free :(



hl⁺⁺

HighLoad⁺⁺



Black

hl⁺⁺

HighLoad⁺⁺

что могу рассказать?

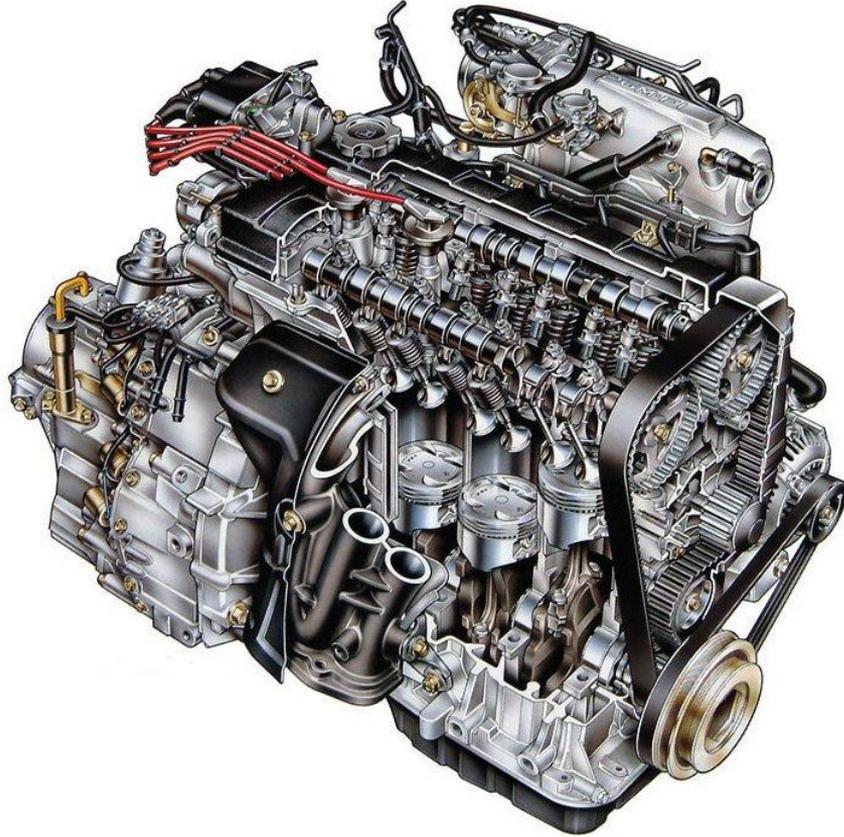
hl⁺⁺

HighLoad⁺⁺

как устроены всякие движки

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺

про движков СУБД (любой)

hl⁺⁺

HighLoad⁺⁺

Система **У**правления
Базой **Д**анных

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++', all contained within a red square.

HighLoad++

данные – это таблицы. из строк

hl++

HighLoad++

phpMyAdmin 3.2.2 - demo.phpmyadmin.net - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://demo.phpmyadmin.net/MAIN

phpMyAdmin 3.2.2 - demo.phpm...

Database: sakila (23)

Go

Database: sakila (23)

- actor
- actor_info
- address
- category
- city
- country
- customer
- customer_list
- film
- film_actor
- film_category
- film_list
- film_text
- inventory
- language
- nicer_but_slower_film_list
- payment
- rental
- sales_by_film_category
- sales_by_store
- staff
- staff_list
- store

	ID	na		code		address	zip code	phone	
<input type="checkbox"/>	218	VERA MCCOY	1168 Najafabad Parkway	40301	896649065861	Kabul	Afghanistan	active	1
<input type="checkbox"/>	441	MARIO CHEATHAM	1924 Shimonoseki Drive	52625	406784385440	Batna	Algeria	active	1
<input type="checkbox"/>	69	JUDY GRAY	1031 Daugavpils Parkway	59025	107137400143	Bchar	Algeria	active	2
<input type="checkbox"/>	176	JUNE CARROLL	757 Rustenburg Avenue	89668	506134035434	Skikda	Algeria	active	1
<input type="checkbox"/>	320	ANTHONY SCHWAB	1892 Nabereznyje Telnj Lane	28396	478229987054	Tafuna	American Samoa	active	2
<input type="checkbox"/>	528	CLAUDE HERZOG	486 Ondo Parkway	35202	105882218332	Benguela	Angola	active	1
<input type="checkbox"/>	383	MARTIN BALES	368 Hunuco Boulevard	17165	106439158941	Namibe	Angola	active	1
<input type="checkbox"/>	381	BOBBY BOUDREAU	1368 Maracabo Boulevard	32716	934352415130	South Hill	Anguilla	active	2
<input type="checkbox"/>	359	WILLIE MARKH							
<input type="checkbox"/>	560	JORDAN ARCHUI							
<input type="checkbox"/>	322	JASON MORRIE							
<input type="checkbox"/>	24	KIMBERLY LEE	96 Tafuna Way	99865	934730187245	Crdoba	Argentina	active	2
<input type="checkbox"/>	445	MICHEAL FORMAN	203 Tambaram Street	73942	41154950611	Escobar	Argentina	active	1
<input type="checkbox"/>	530	DARRYL ASHCRAFT	166 Jinchang Street	86760	717566026669	Ezeiza	Argentina	active	2
<input type="checkbox"/>	89	JULIA FLORES	1926 El Alto	75543	846225459260	La Plata	Argentina	active	1

Find: algo Next Previous Highlight all Match case

Done 1024x768

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

hl⁺⁺

HighLoad⁺⁺

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!675292816413!Antofagasta!Chile!acti
ve! NO # #KRISTEN CHAVEZ!1345 Oshawa Boulevard!32114!104491201771!Hino!Japan!ac
tive!  I #!BOB PFEIFFER!415 Pune Avenue!44274!203202500108!Xintai!China!acti
ve! l #!
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! #!JORDAN ARCHULETA!1229 Varanasi (Benares) Man
or!40195!817740355461 Avellaneda      Argentina!active! I #
!CASSANDRA WALTERS!!920 Kumbakonam Loop!75090!685010736240!Salinas
United States!active! N #!
TOMMY COLLAZO!!76 Kermanshah Manor!23343!762361821578!Qomsheh!Iran!active!  I #
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 #!
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 #! RAY HOULE!740 Udaipur Lane!33505!497288595103      Dzerzinsk!Russia
n Federation!active! T #!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%
00 i 0|| 3NDA      L↑ 01BRANDY GRAVES1944 Bamenda Way2464575975221996Warren
United Statesactive0▲↑  ♀↑♂ 08 w 0|| 3NDA      L↑100^JUAN FRALEY469 Nakh
on Sawan Street58866689199636560TeboksaryRussian Federationactive0▲♂  ♀±♀ 0@
u 0|| r 3NDA      L↑@ 3DARLENE ROSE1386 Nakhon Sawan Boulevard535023688991
74225PyongyangNorth Koreaactive0▲±♀♀±♀ ♀ 0H g 0|| 3NDA      L↑P ♀NANCY THOMA
S808 Bhopal Manor10672465887807014Yamuna NagarIndiaactive0▲↑♂♀±↑↑ 0P v 0|| T
3NDA      L↑' .CATHERINE CAMPBELL46 Pjatigorsk Lane23616262076994845MoscowRussian
Federationactive0▲  ♀±→ 0X z 0|| 3NDA      L↑p00JORDAN ARCHULETA122
9 Varanasi (Benares) Manor40195817740355461AvellanedaArgentinaactive0▲♂♂♀±←← 0'
p 0|| = 3NDA      L↑A 6GERALDINE PERKINS97 Shimoga Avenue44660177167004331
Tel Aviv-JaffaIsraelactive0▲↑♂♀±→
0h e 0|| † 3NDA      L↑P0:GEORGE LINTON1957 Yantai Lane59255704948322302Soroc
abaBrazilactive0▲  ♀±↑→ 0p o 0|| 3NDA      L↑a QANDREA HENDERSON320
Baiyin Parkway37307223664661973MahajangaMadagascaractive0▲♂♀♀±→ 0x i 0|| L
3NDA      L↑00KENT ARSENAULT32 Liaocheng Way1944410877354933Juiz de ForaBrazilact
ive0▲  ♀±←* 0A v 0|| T 3NDA      L↑L08ALBERTO HENNING502 Mandi Ba
hauddin Parkway15992618156722572BarcelonaVenezuelaactive0▲±♂♀±←♀ 00M b 0|| T 3
NDA      L↑L 3WALICIA MILLS1963 Moscow Place64863761379480249NagaonIndiaactive0▲±→
♀±↑
0P p 0|| L 3NDA      L↑p .CYNTHIA YOUNG1425 Shikarpur Manor65599678220867005M
unger (Monghyr)Indiaactive0▲±♀±±♂ 0|| j 0|| L 3NDA      L↑E0||MARIO CHEATHAM1924
Shimonoseki Drive52625406784385440BatnaAlgeriaactive0▲♂  ♀±*♂ 0a c 0|| r 3NDA
L!! 0>BRIAN WYMAN1769 Iwaki Lane25787556100547674BydgoszczPolandactive0▲±
1      2      3      4      5Print 6      7      8Goto 9Video 10

```

hl⁺⁺

HighLoad⁺⁺

это, кстати, данные – без индексов

hl⁺⁺

HighLoad⁺⁺

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! U
-
JANET PHILLIPS!1718 Valencia Street!37359!675292816413!Antofagasta!Chile
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! W
ve!
ve!
FRAN!
Tuxt!
Kamy!
or!
@!
Unit!
TOMM!
GO!
dom!
BILL!
BRIA!
LEON!
e!
BRE!
Q!
ve!
n Fe!
4966!
1
FRANCIS SIKES!355 Vitria de Santo Anto Way!81758!548003849552!San Juan Bautista
Tuxtepec!Mexico!active! R!
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! P!
TOMMY COLLAZO!!76 Kermanshah Manor!23343!762361821578!Qomsheh!Iran!active!
MARSHALL THORN!!1584 Ljubertsy Lane!22954!285710089439!Southampton!United Kin
gdom!active! N!
BILLIE HORTON!457 Tongliao Loop!56254!880756161823!Inegl!Turkey!active! N!
BRIAN WYMAN!1769 Iwaki Lane!25787!556100547674 Bydgoszcz!Poland!active!
S!
LEON BOSTIC!734 Tanshui Avenue!70664!366776723320 Florencia!Colomb
ia!active! T!
BRENT HARKINS!319 Plock Parkway!26101!854259976812!Sultanbeyli!Turkey!active! S
GRACE ELLIS!442 Rae Bareli Place!24321!886636413768!Duisburg!Germany!act
ive! Y!
RAY HOULE!740 Udaipur Lane!33505!497288595103 Dzerzinsk!Russia
n Federation!active! V!
GERALD FULTZ!45 Aparecida de Goiania Place!7431!65

```

```
view test_innodb.ibd - Far
C:\...0\data\sakila\test_innodb.ibd      DOS      163840      Col 0      64%
00 i 0|| 0NDA 0↓ 0IBRANDY GRAVES1944 Bamenda Way2464575975221996Warren
```

```
view test_innodb.ibd - Far
C:\...0\data\sakila\test_innodb.ibd      DOS      163840      Col 0      63%
s Place1290562127829280RajkotIndiaactive0♠♀
♀♠!!* 0a↑0A 0+ 0NGA 0♠↓ 0DWIGHT LOMBARDI1497 Fengshan Drive63022368738360376Ch
0NDA atsworthSouth Africaactive0♠♠ ♀♠♠♠♠ 0u gA 0- 0NGA
Fede >FELIX GAFFNEY1059 Yuncheng Avenue47498107092893983VilniusLithuaniaactive0 ♠♠♀♠♠
9 Va ▶ 0♠ aA 0. 0NGA 0♠ 0EJIMMIE EGGLESTON505 Madiun Boulevard97271970638808606
p WroclawPoland0♠♠♠♀♠♠♠ 0♠ hA 0/ 0NGA 0♠ 0EVERETT BANDA1741 Hoshiarpur
Tel Boulevard22372855066328617BilbaysEgyptactive0♠♠
♠♠♠▶ 0♠ vA 00 0NGA 0♠▶ 0JORDAN ARCHULETA1229 Varanasi (Benares) Manor40195817
abaB 740355461AvellanedaArgentinaactive0♠♠♠♠♀♠♠♠ ♠0♠ eA 01 0NGA 0♠ (IAN S
Bai TILL1894 Boa Vista Way77464239357986667GarlandUnited Statesactive0♠♠♠♀♠♠♠♠ 0♠_A
0NDA 02 0NGA 0♠♥ 0WALLACE SLONE1342 Sharja Way93655946114054231SokotoNigeriaact
ive0♠♠♠ ♀♠!!♠ 0♠+*♠A 03 0NGA 0♠♠KEN PREWITT1342 Abha Boulevard107149974536071
haud 16BucurestiRomaniaactive0♠♠ ♀♠+* 0p rA 08 0NGA 0♠▶ 0ALBERTO HENNI
NDA NG502 Mandi Bahauddin Parkway15992618156722572BarcelonaVenezuelaactive0♠♠♠♀♠♠♠♠
♀♠♠ 0♠♠ cA 09 0NGA 0♠♠
0p DAVE GARDINER1052 Pathankot Avenue77397128499386727LeshanChinaactive0♠♠♠♠♀♠♠♠♠ 0♠E
unge aA 0: 0NGA 0♠♠jIVAN CROMWELL1351 Sousse Lane37815203804046132MonclovaMexicoa
Shim ctive0♠♠♠♠♀♠♠▶ 0♠° iA 0; 0NGA 0♠♀JOHNNIE CHISHOLM1501 Pangkal Pinang A
venue943770864062795PlockPolandactive0♠♠♠♠♀♠♠!!* 0 iA 0< 0NGA 0♠♥%SIDNE
1 Y BURLESON1405 Hagonoy Avenue86587867287719310CzestochowaPolandactive0♠♠ ♠♀♠♠♠
♠♠♠♠♠♠A 0= 0NGA 0♠♀'BYRON BOX521 San Juan Bautista Tuxtepec Place95093844
0183485657hezgagzghanKazakstanactive0♠♠♠
```

hl⁺⁺

HighLoad⁺⁺

почему так?

разные стратегии хранения
строк

hl⁺⁺

HighLoad⁺⁺

MyISAM – в порядке
поступления
(в конец файла)

hl⁺⁺

HighLoad⁺⁺

InnoDB – хранит постранично,
внутри странички – в порядке
РК

hl⁺⁺

HighLoad⁺⁺

MyISAM – “обычное” хранение
InnoDB – т.н. “кластерное”

hl⁺⁺

HighLoad⁺⁺

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

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a red square background.

HighLoad++

```
SELECT * FROM users WHERE  
      id=123
```

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

```
SELECT * FROM users WHERE  
    lastname='Pupkin'
```

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

```
SELECT * FROM users WHERE  
    lastname LIKE 'Pu%'
```

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

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

The logo consists of the lowercase letters 'hl' followed by two plus signs '++', all in white, set against a solid red square background.

HighLoad++

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

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

как выполнять запрос?

hl⁺⁺

HighLoad⁺⁺

полный перебор? мееедленно

hl⁺⁺

HighLoad⁺⁺

нас спасут...

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

индексы!

алгоритмы уже спешат на
помощь!

hl⁺⁺

HighLoad⁺⁺

СМЫСЛ ЛЮБОГО ВИДА ИНДЕКСА?

hl⁺⁺

HighLoad⁺⁺

быстрый поиск по ключу(-ам)

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

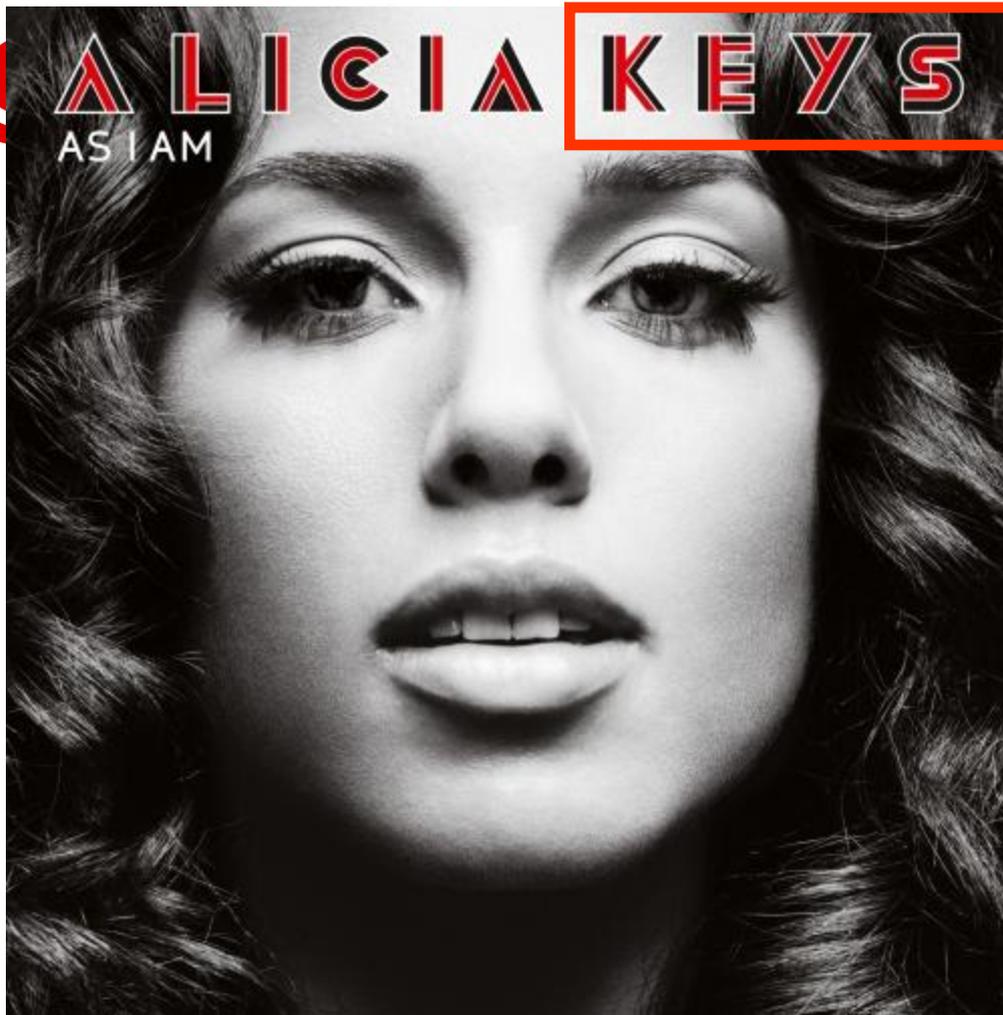


hl⁺⁺

High

ALICIA KEYS

AS I AM



hl⁺⁺

HighLoad⁺⁺

ВИДОВ ИНДЕКСОВ – ТОЖЕ МНОГО



HighLoad++

hash index

hl⁺⁺

HighLoad⁺⁺

R-tree index



HighLoad++

full-text index

hl⁺⁺

HighLoad⁺⁺

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

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

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

hl⁺⁺

HighLoad⁺⁺

дискует – страничками
(хорошо!)

hl⁺⁺

HighLoad⁺⁺

используется – всеми

hl⁺⁺

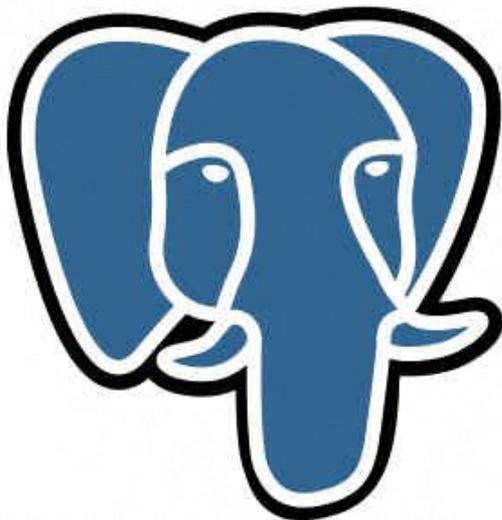
HighLoad⁺⁺



hl⁺⁺

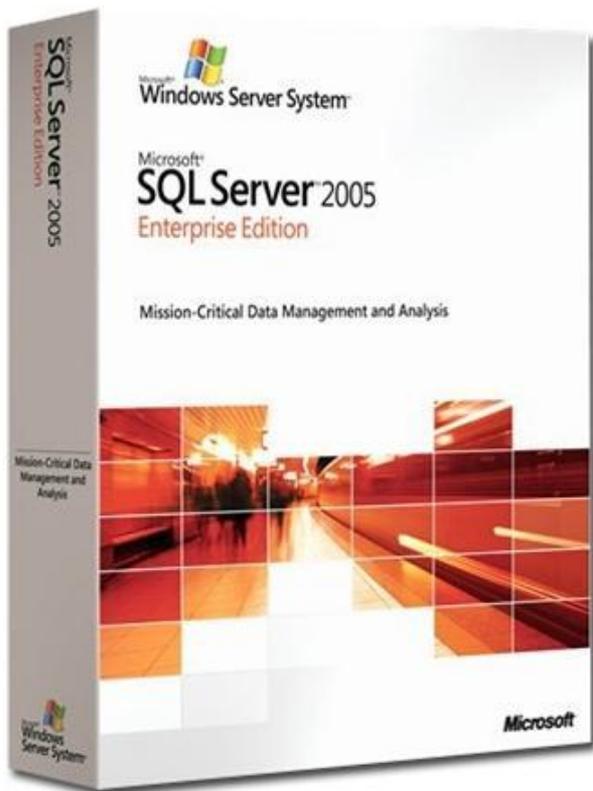
HighLoad⁺⁺

PostgreSQL



hl++

HighLoad++





HighLoad++

ORACLE®

hl⁺⁺

HighLoad⁺⁺

используется несмотря ни на что!!!



€



Sun
microsystems

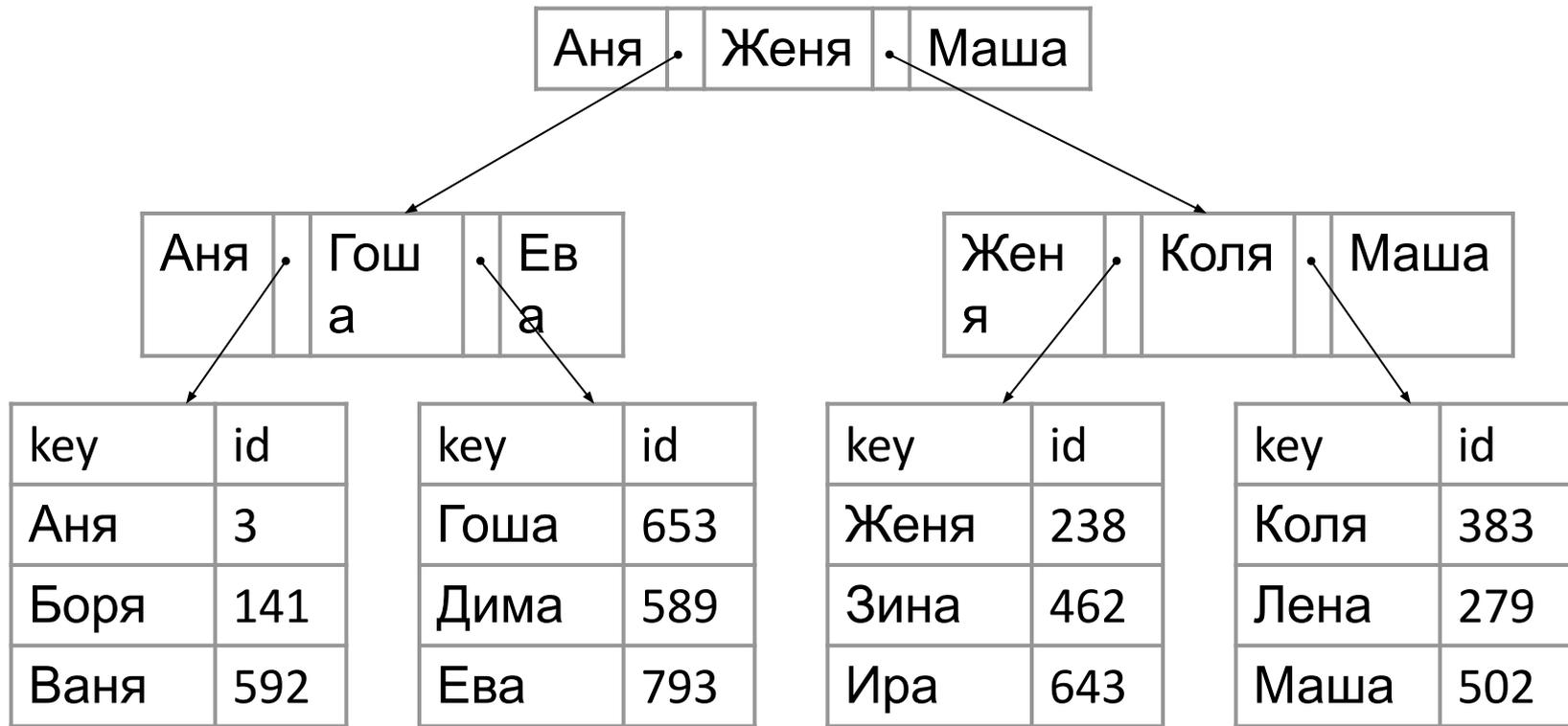
€

ORACLE®

hl⁺⁺

HighLoad⁺⁺

как устроено?



(B-дерево; странички; масштаб 1:72)

hl⁺⁺

HighLoad⁺⁺

два вида страничек

Промежуточные = ключи + указатели на другие странички



{ key1, ptr1, key2, ptr2, ..., keyN }

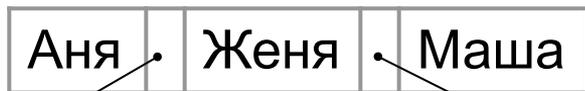
Листовые = ключи + соотв-е им данные (eg. row_offset)

key	id
Аня	3
Боря	141
Ваня	592

Аня	3	Бор	141	Ван	592
		я		я	

{ 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

hl⁺⁺

HighLoad⁺⁺

почему все используют этот
ужас?!

hl⁺⁺

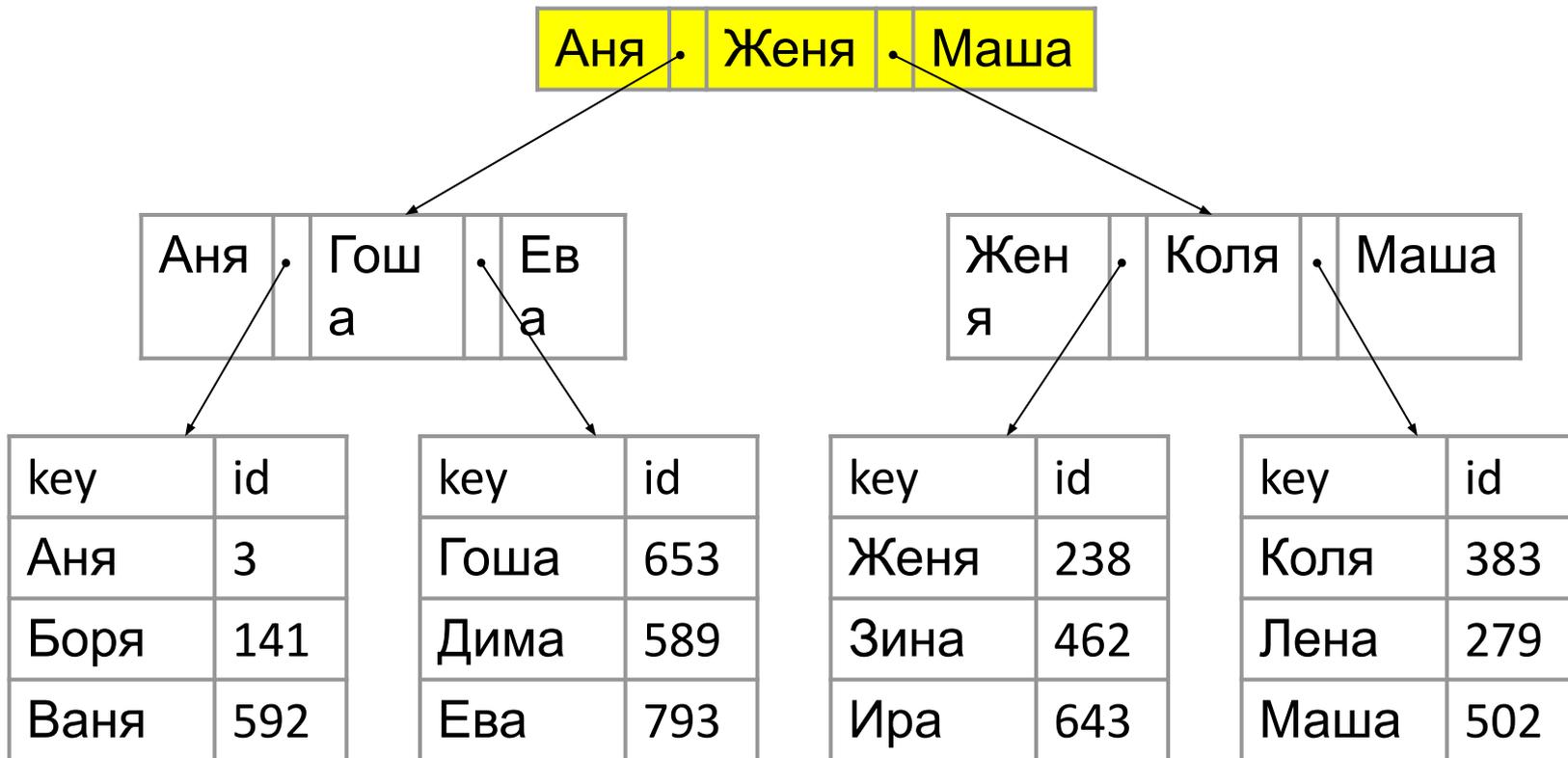
HighLoad⁺⁺

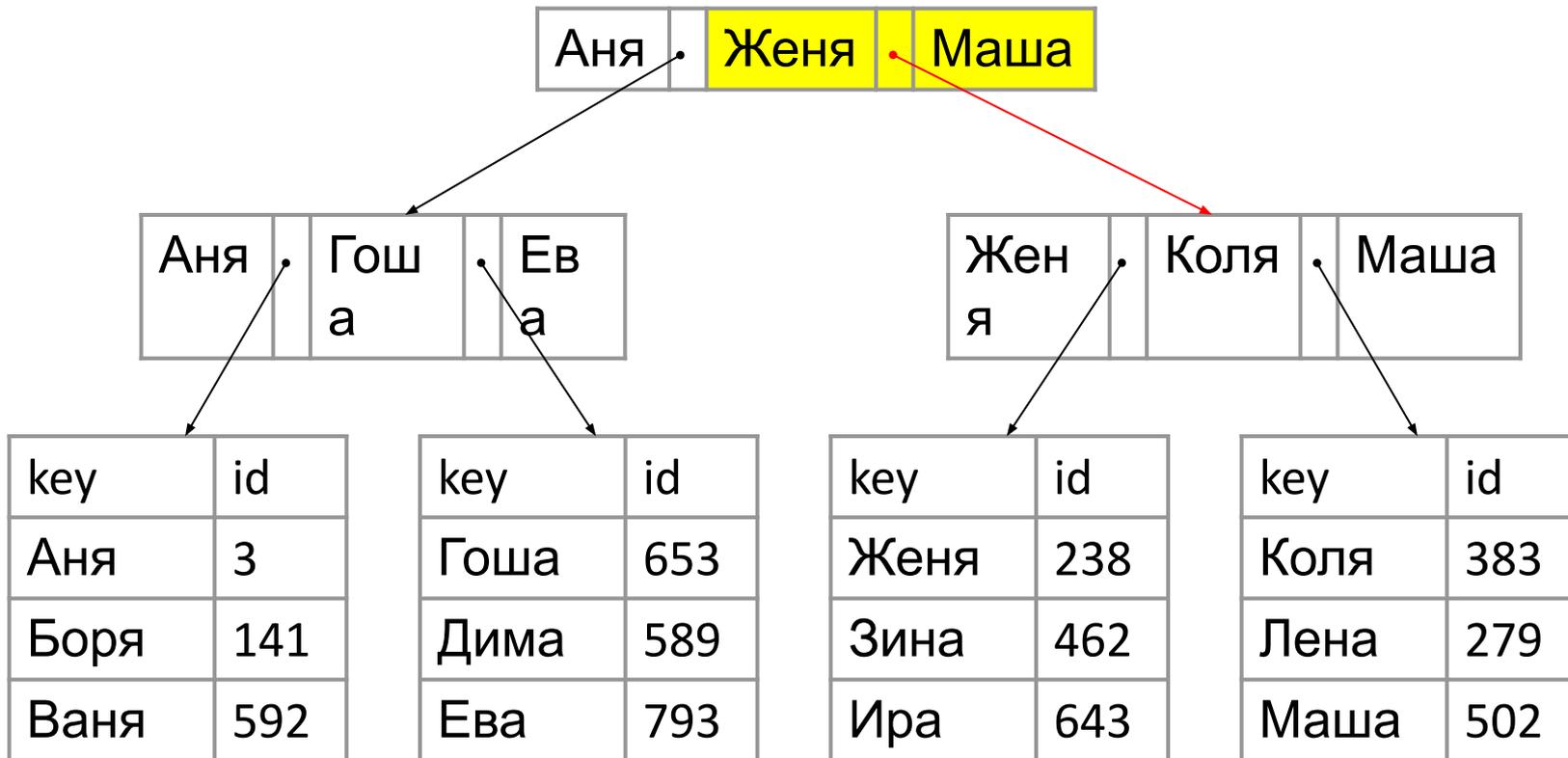
во-1х – легко искать по ключу

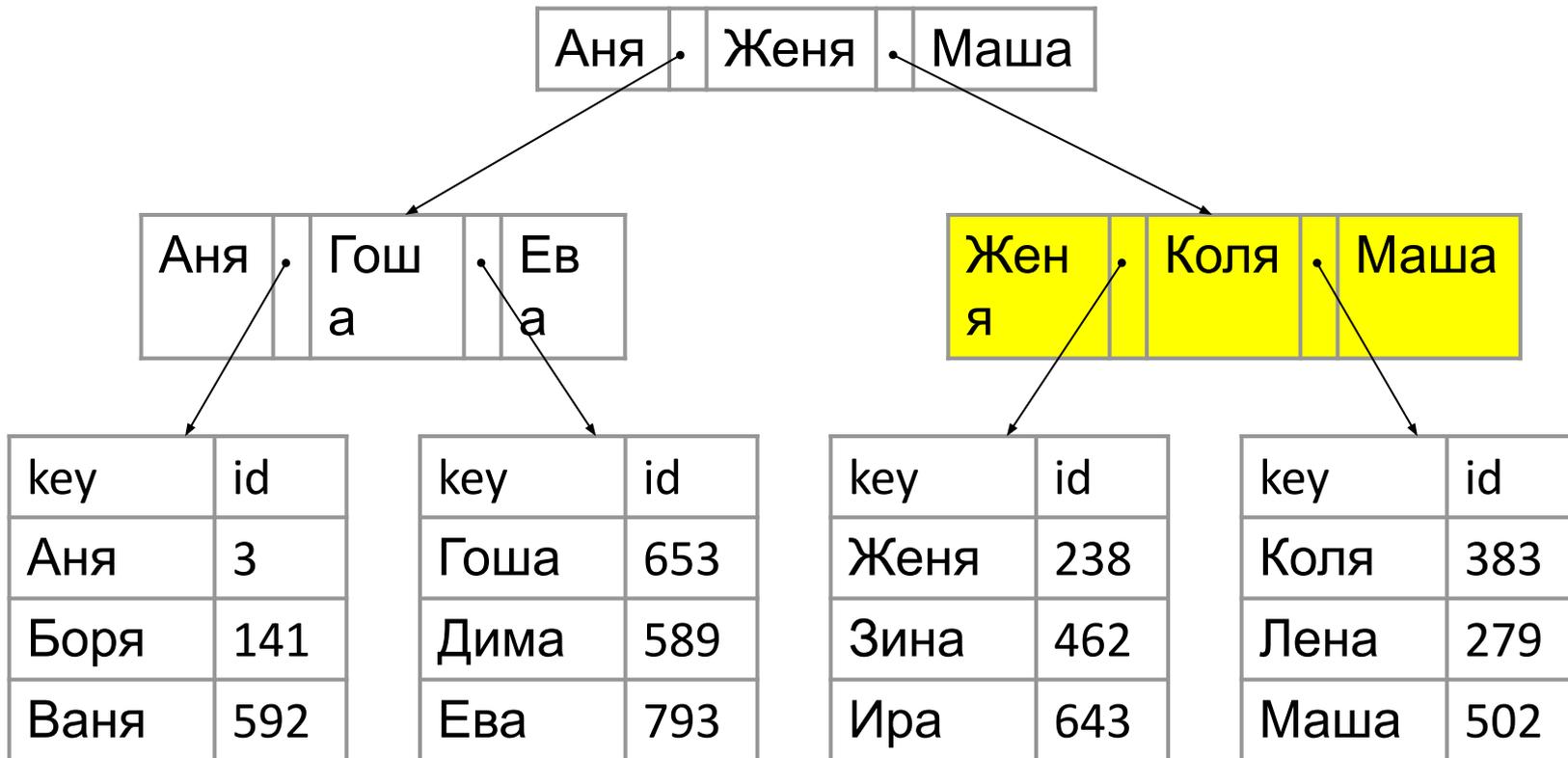
hl⁺⁺

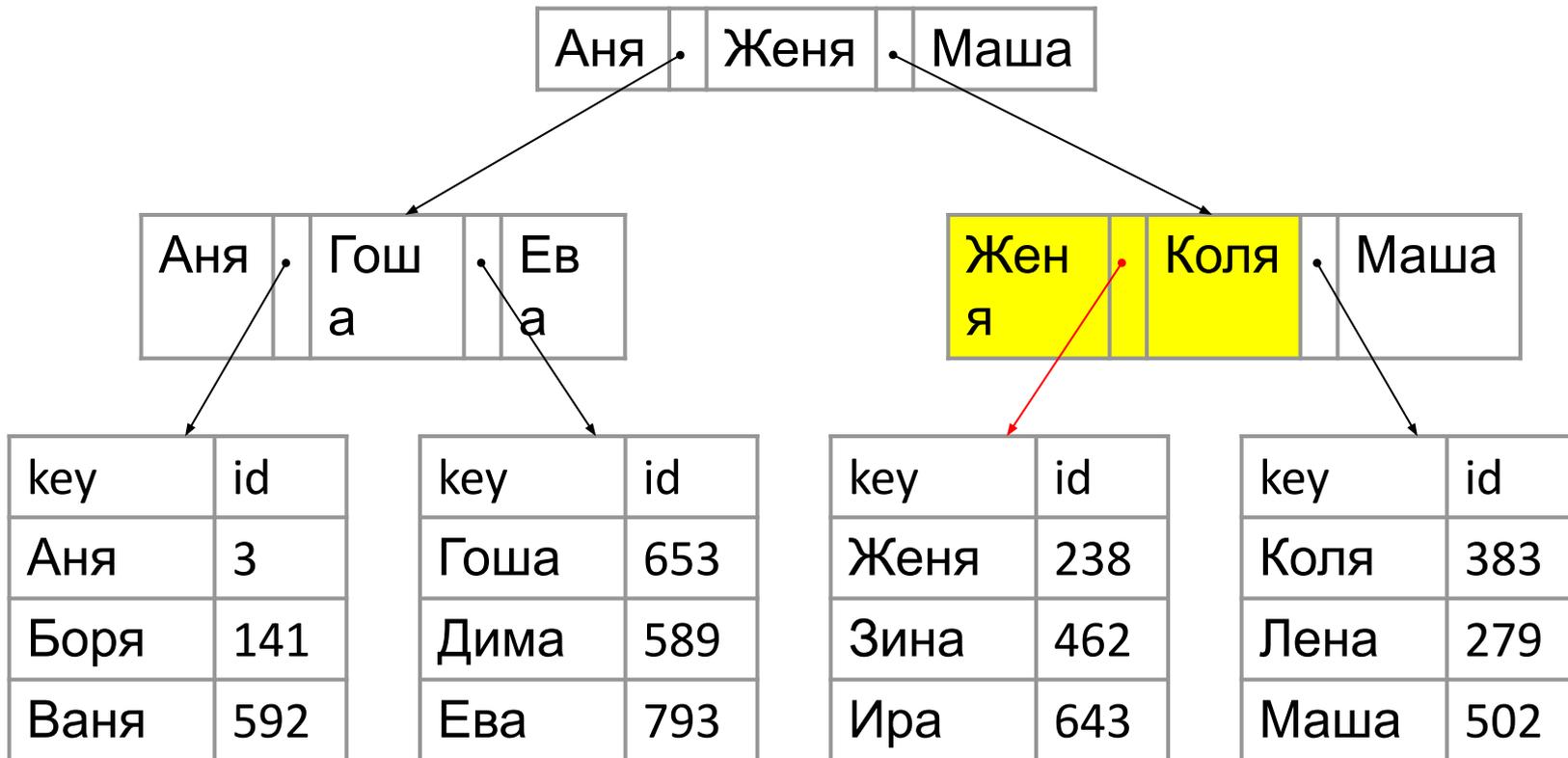
HighLoad⁺⁺

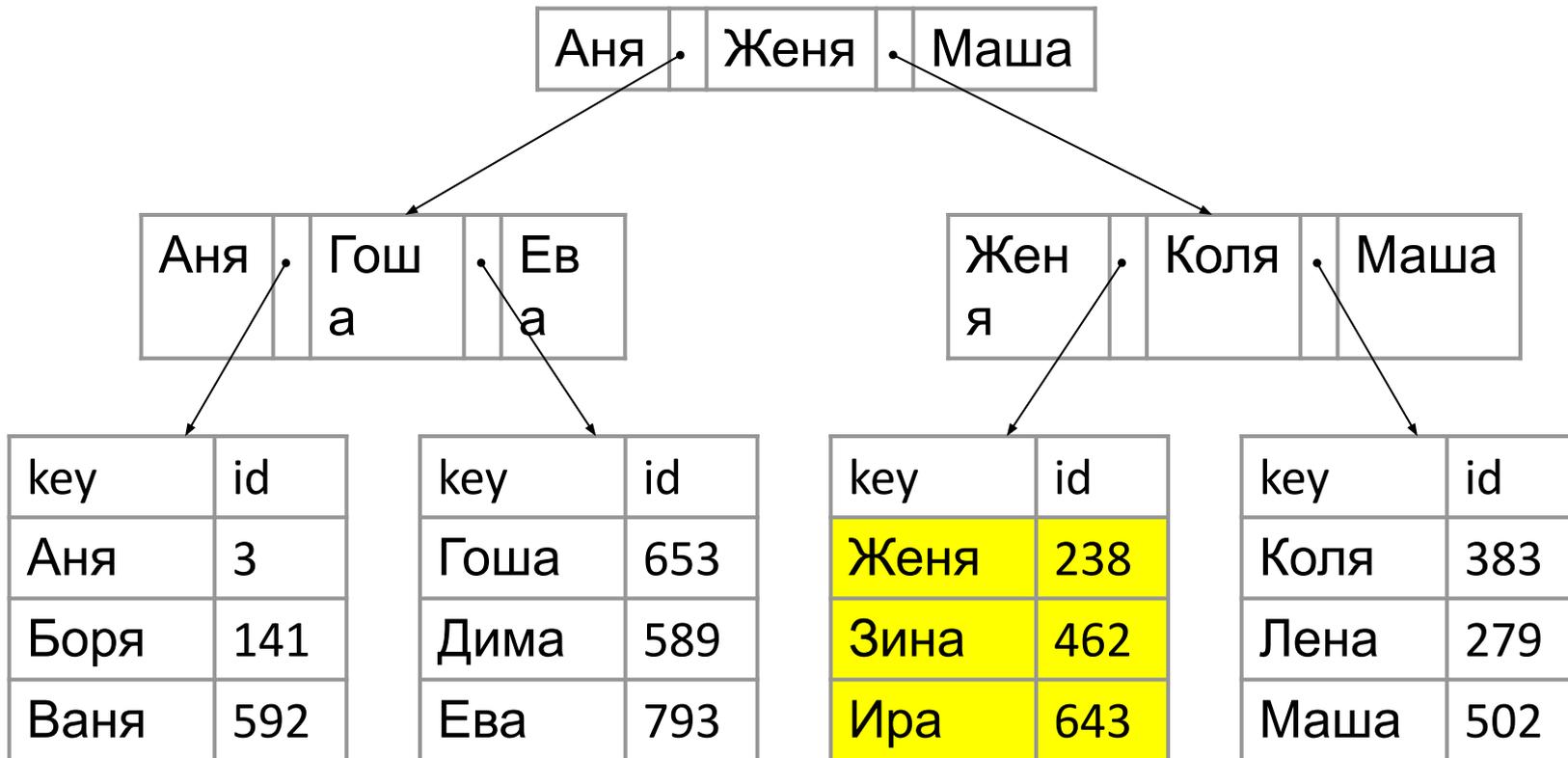
пример – ищем Зину

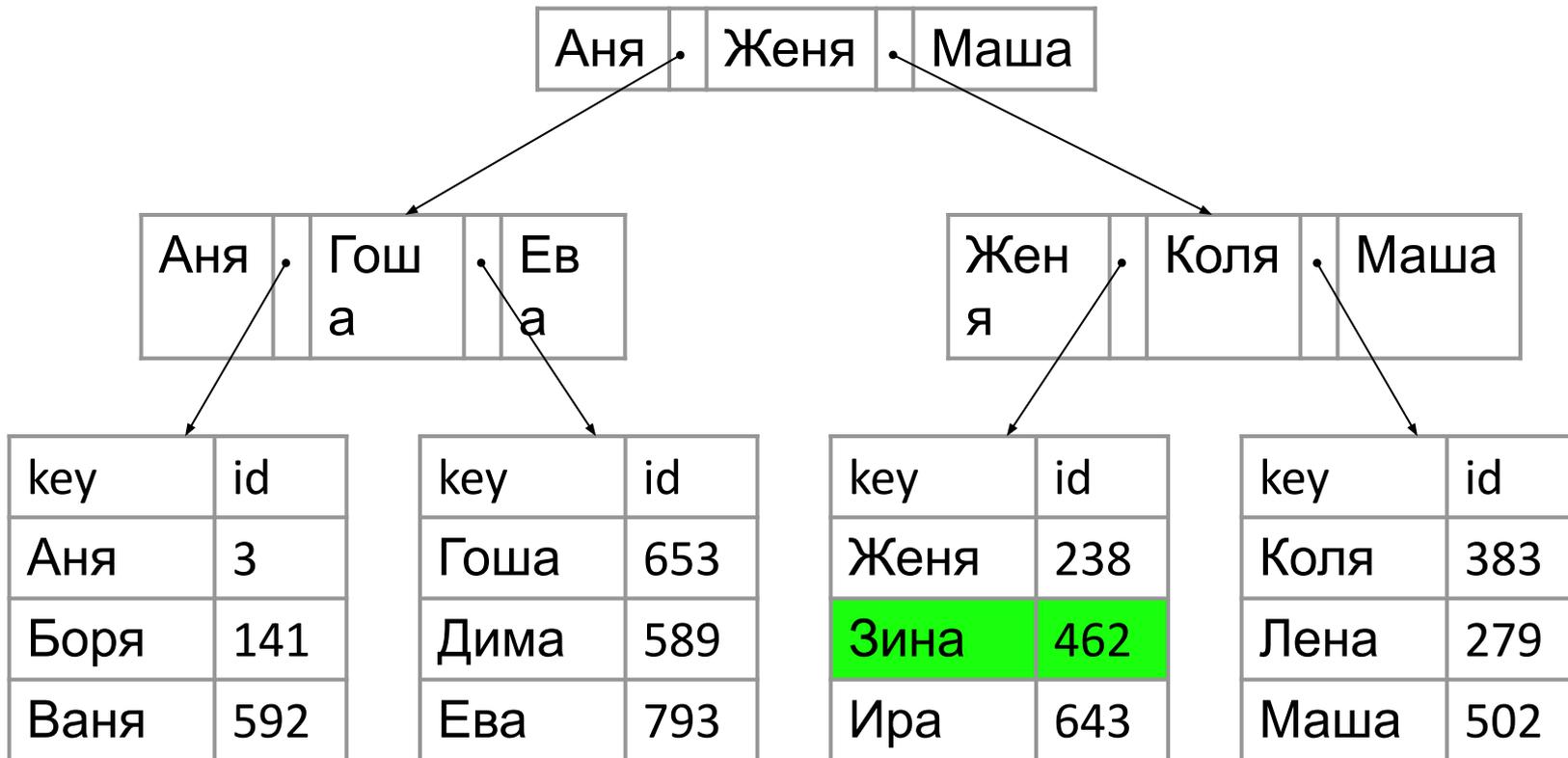












hl⁺⁺

HighLoad⁺⁺

ура – Зина нашлась!!!

hl⁺⁺

HighLoad⁺⁺

хорошо – поиск работает...

hl⁺⁺

HighLoad⁺⁺

...НО ОН ЧЁ, ВСЕГДА ТАКОЙ
РЕЗКИЙ?

hl⁺⁺

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

key	id
Коля	383
Лена	279
Маша	502

hl⁺⁺

HighLoad⁺⁺

почему все используют этот
ужас?!

hl⁺⁺

HighLoad⁺⁺

во-2х – легко обновлять

hl++

HighLoad++

странички обычно НЕ
ПОЛНЫ



key	id
Аня	3
?	?
?	?

hl⁺⁺

HighLoad⁺⁺

вставляем...



key	id
Аня	3
Боря	141
?	?

hl++

HighLoad++

вставляем...

key	id
Аня	3
Боря	141
Ваня	592

hl++

HighLoad++

вставляем... оп-па,
некуда!!



key	id
Аня	3
Боря	141
Ваня	592
Гоша	653

hl++

HighLoad++

создаем новую страничку

key	id
Аня	3
Боря	141
Ваня	592
Гоша	653

key	id
?	?
?	?
?	?
?	?

hl++

HighLoad++

создаем новую страничку

key	id
Аня	3
Боря	141
?	?

key	id
Ваня	592
Гоша	653
?	?

hl++

HighLoad++

...и суем "ее" в родителя

key	id
Аня	3
Боря	141
?	?

key	id
Ваня	592
Гоша	653
?	?

hl++

HighLoad++

это все – тоже трогаем max
2-3 странички

key	id
Аня	3
Боря	141
?	?

key	
Ваня	592
Гоша	653
?	?

hl⁺⁺

Hig



hl⁺⁺

HighLoad⁺⁺

B-tree Kung Fu!!!

hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺

усложним – добавим условий

The logo consists of the lowercase letters 'hl' followed by two plus signs '++', all in white, set against a solid red square background.

HighLoad++

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

hl⁺⁺

HighLoad⁺⁺

индекс “в лоб” по sex?

hl⁺⁺

HighLoad⁺⁺

они ВСЕ подходят по условию
'F'!

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

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

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

hl⁺⁺

HighLoad⁺⁺

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	12369
F	12370
F	12371
F	12372
F	12373
F	12374
F	12375
F	12376



hl⁺⁺

HighLoad⁺⁺

неселективный индекс –
косяк и западло!

hl⁺⁺

HighLoad⁺⁺

sex='F' AND age>=18 AND age<=25

индекс “по лбу” по age?

key	id
13	12345
13	12346
13	12347
14	12348

key	id
17	12353
18	12354
18	12355
18	12356

key	id
25	12361
25	12362
27	12363
29	12364

НО – ВДРУГ ЭТО МУЖИКИ?!

15	12351
16	12352

21	12359
22	12360

53	12367
64	12368

hl⁺⁺

HighLoad⁺⁺

мужики нам не нужны!!!

hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺

СПИСОК НУЖНЫХ СТРОК – ЯСЕН СРАЗУ

hl⁺⁺

HighLoad⁺⁺

чтений с диска – минимум
скорости – максимум

hl⁺⁺

HighLoad⁺⁺

бонус – сортировка по age

hl⁺⁺

HighLoad⁺⁺

кстати, про сортировку...



HighLoad++

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

hl⁺⁺

HighLoad⁺⁺

как выполнять? есть варианты

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

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

ИНДЕКС

данные

е

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

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

а

результат

hl⁺⁺

HighLoad⁺⁺

read_index – мало и быстро

hl⁺⁺

HighLoad⁺⁺

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

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

hl⁺⁺

HighLoad⁺⁺

`read_random_rows` – медленно!
 $10K * 5-10 \text{ ms/seek} = 50-100 \text{ sec} \dots$

hl⁺⁺

HighLoad⁺⁺

sort_rows – обычно быстро

$10K * 0.1-1 \text{ KB/row} = 1-10 \text{ MB (in RAM)}$

hl⁺⁺

HighLoad⁺⁺

мораль – все зло от random
rows!

hl⁺⁺

HighLoad⁺⁺

(еще от `sort_rows`, если их
МНОГО)

hl⁺⁺

HighLoad⁺⁺



The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

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

hl⁺⁺

HighLoad++

направо –

read_fat_index(WHERE) +
sort_index(ORDER) + LIMIT +
read_less_rows + sort_rows

hl⁺⁺

HighLoad⁺⁺

нужен утолщенный
INDEX (sex, age, hotness)

hl⁺⁺

HighLoad⁺⁺

ВМЕСТЕ С ПОИСКОМ ПО sex, age –
сразу узнаем hotness (+40 KB)

hl⁺⁺

HighLoad⁺⁺

```
sort ( 10K * [ hotness, rowptr ] )
```

hl⁺⁺

HighLoad⁺⁺

read_rows – почти не нужен
(10 строк результата...)

hl⁺⁺

HighLoad⁺⁺

sort_rows – вообще не нужен



HighLoad++

PROFIT?

hl⁺⁺

HighLoad⁺⁺

не панацея – даже в теории

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

INDEX (sex, age, hotness)

WHERE sex='F' ORDER BY hotness LIMIT 10

hl⁺⁺

HighLoad⁺⁺

в теории – обработать 50%

индекса

затем – прочитывать 10 строк (пф!)

hl⁺⁺

HighLoad⁺⁺

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

The logo consists of the lowercase letters 'hl' in white on a red square background, followed by two plus signs '++' in white.

HighLoad++

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

hl⁺⁺

HighLoad⁺⁺

что на практике?

hl⁺⁺

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) )
```

hl⁺⁺

HighLoad⁺⁺

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”

hl⁺⁺

HighLoad⁺⁺

Using where – проверка условия
NE по индексу – ?!!

hl⁺⁺

HighLoad⁺⁺

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)
```

hl⁺⁺

HighLoad⁺⁺

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)

hl⁺⁺

HighLoad⁺⁺

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)
```

hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺

сортировка “по индексу” –
только если индекс
гарантирует порядок

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

2) optimizer лажанул, $18 \leq \text{age} \leq 25$
считается НЕ по индексу (а могло
бы)

hl⁺⁺

HighLoad⁺⁺

(теория говорит – можно
лучше!)

hl⁺⁺

HighLoad⁺⁺

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)
```

hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺



теория была оптимистична...

hl⁺⁺

HighLoad⁺⁺



...реальность внесла
коррективы.

hl⁺⁺

HighLoad⁺⁺

не учили детали реализации

1. нету “досортировки”
индекса
(MySQL specific)

2. limit обрабатывается... так
себе
(MySQL specific)

hl⁺⁺

HighLoad⁺⁺

3. optimizer ошибается
(везде и у всех)

hl⁺⁺

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)
```

hl⁺⁺

HighLoad++

$10,000 \times 10 \text{ ms} = 100 \text{ sec}$

$100,000 \times 1\text{KB} / 50 \text{ M/s} = 2 \text{ sec}$

hl⁺⁺

HighLoad⁺⁺

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

hl⁺⁺

HighLoad⁺⁺

про “обработку” limit

hl⁺⁺

HighLoad⁺⁺

теория – приоритетная
очередь

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺



приоритетная!!



TWICE BANK
TWICE BANK

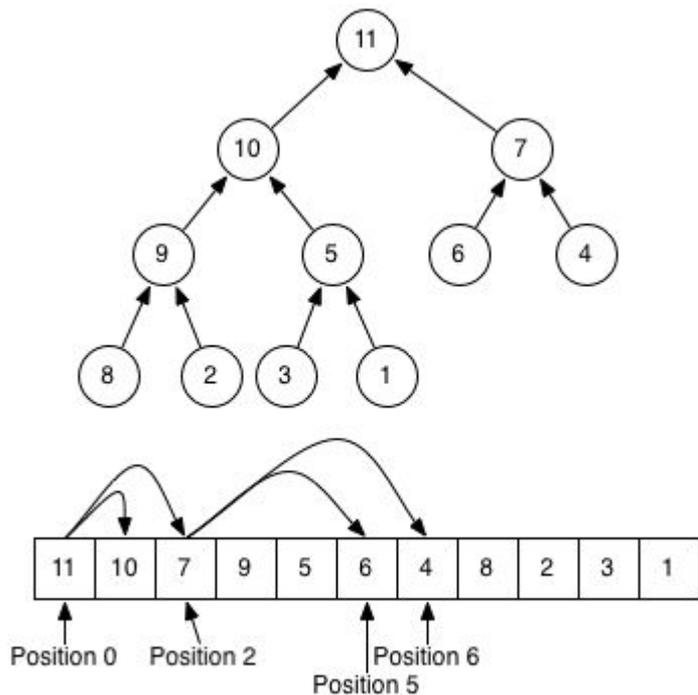
hl⁺⁺

HighLoad⁺⁺

технически – hear

hl++

HighLoad++



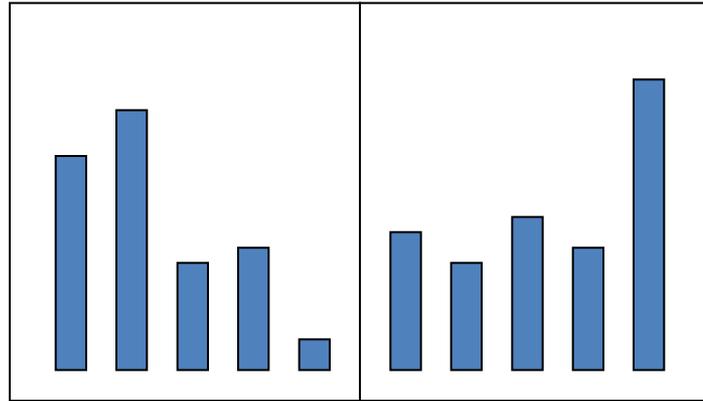
hl⁺⁺

HighLoad⁺⁺

или просто double buffer

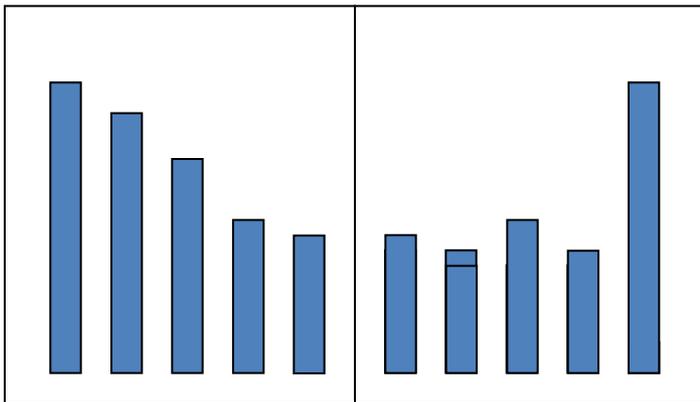


HighLoad++



hl⁺⁺

HighLoad++



The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

**КЛЮЧЕВОЕ СВОЙСТВО –
в памяти храним только top-N**

The logo consists of the lowercase letters 'hl' in white, followed by two plus signs '++' in white, all set against a solid red square background.

HighLoad++

LIMIT 10 – надо хранить 10
строк

hl⁺⁺

HighLoad⁺⁺

LIMIT 130,10 – надо 140

hl⁺⁺

HighLoad⁺⁺

практика – MySQL vs. LIMIT

hl⁺⁺

HighLoad⁺⁺

выбрать и отсортировать **ВСЕ**
(*)

* – всегда, когда индекс не гарантирует точный порядок

hl⁺⁺

HighLoad⁺⁺

выбрать ОК – избежать нельзя

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

сортировать **все** плохо...

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

лишний удар по CPU/RAM/IO :(

hl⁺⁺

HighLoad⁺⁺

как убирать mysql сортировку?

hl⁺⁺

HighLoad⁺⁺

строить более другие индексы

hl⁺⁺

HighLoad⁺⁺

ставить более другой софт

hl⁺⁺

HighLoad⁺⁺



умеет и “обычный” поиск!

hl⁺⁺

HighLoad⁺⁺

трюки про WHERE вместо LIMIT
(я не пробовал, но говорят, возможно)

hl⁺⁺

HighLoad⁺⁺

...ИМЕННО В ТАКОМ ПОРЯДКЕ.

hl⁺⁺

HighLoad⁺⁺

более практический пример?

hl⁺⁺

HighLoad⁺⁺

импортируем дамп Wikipedia



hl⁺⁺

HighLoad⁺⁺

XML дампы → 2 толстые
таблицы
хочется – а) одну б) тонкую!

hl⁺⁺

HighLoad⁺⁺

```
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
```

hl⁺⁺

HighLoad⁺⁺

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
```

hl⁺⁺

HighLoad++

решение – `index(page_latest)`

hl⁺⁺

HighLoad⁺⁺

еще пришлось STRAIGHT_JOIN
(optimizer опять лажанул!)

hl⁺⁺

HighLoad⁺⁺

результат – 40 минут, включая
CREATE INDEX

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

так зачем же знать алгоритмы?

hl⁺⁺

HighLoad⁺⁺

“did we learn something today?”

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

как устроено B-дерево

hl⁺⁺

HighLoad⁺⁺

как работает индекс

hl⁺⁺

HighLoad⁺⁺

как работают выборки

hl⁺⁺

HighLoad++

зачем нужны full-scans

hl⁺⁺

HighLoad⁺⁺

как работает сортировка с
LIMIT

hl⁺⁺

HighLoad⁺⁺

чего можно добиться в идеале
– в теории

hl⁺⁺

HighLoad⁺⁺

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



hl⁺⁺

HighLoad++

а толку?!

hl⁺⁺

HighLoad⁺⁺

чего ждать от БД

hl⁺⁺

HighLoad++

чего не ждать

hl⁺⁺

HighLoad⁺⁺

как и что тестировать

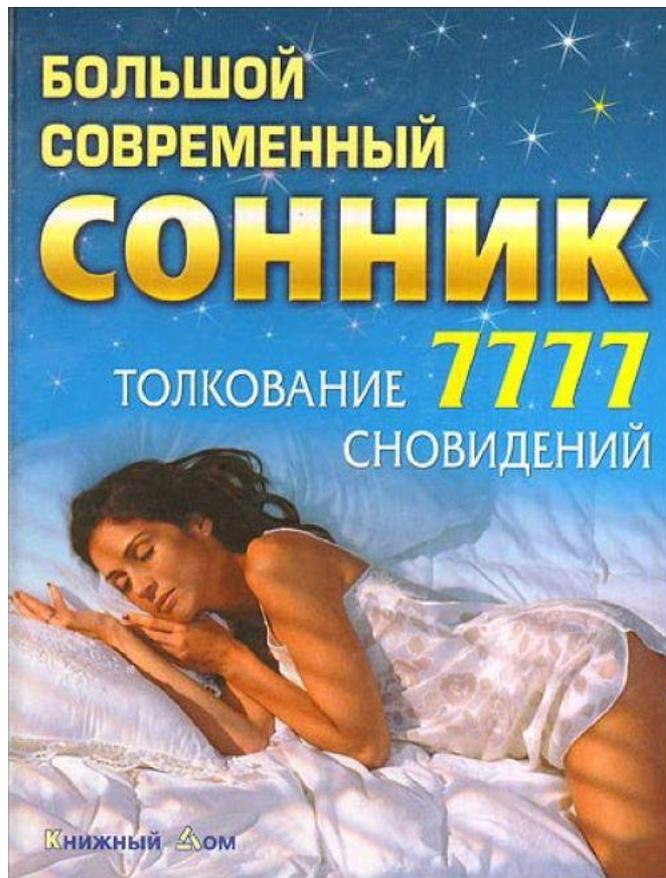
hl⁺⁺

HighLoad⁺⁺

как объяснять потом результаты

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

В итоге –

как заставлять таки работать

hl⁺⁺

HighLoad⁺⁺



hl⁺⁺

HighLoad⁺⁺

...**БЫСТРО** работать.

hl⁺⁺

HighLoad⁺⁺



“это все” (с) вопросы?!