

Скрытая сложность повседневной задачи: отображение табличных данных

Иван Пономарёв, МФТИ / КУРС

ponomarev@corchestra.ru

[@inponomarev](#)

Над чем мы работаем

- Платформа на Java
- PostgreSQL / MSSQL / Oracle — СУБД
- Jython — код бизнес-логики

Платформа даёт стандартные решения для создателей бизнес-приложений, в частности, элементы интерфейса и **грид**

Грид – это прокрутка...

name	socr	code	postalc...	gnin...	uno	oc...
(коттеджный поселок Мастрюки)	п	63028000006005400	443545	6330		36...
(Микрорайон А) 1-ая Северная	ул	63012000044006500	446402	6350		36...
(Микрорайон А) 2-ая Северная	ул	63012000044006600	446402	6350		36...
(Микрорайон А) 3-ая Северная	ул	63012000044006700	446402	6350		36...
(Микрорайон А) 4-ая Северная	ул	63012000044006800	446402	6350		36...
(Микрорайон А) 5-ая Северная	ул	63012000044006900	446402	6350		36...
(Микрорайон А) 6-ая Северная	ул	63012000044007000	446402	6350		36...
(Микрорайон А) 7-ая Северная	ул	63012000044007100	446402	6350		36...
(Микрорайон А) 8-ая Северная	ул	63012000044007200	446402	6350		36...
(Микрорайон Б) 1-ая Озерная	ул	63012000044007300	446402	6350		36...
(Микрорайон Б) 2-ая Озерная	ул	63012000044007400	446402	6350		36...
(Микрорайон Б) 3-ая Озерная	ул	63012000044007500	446402	6350		36...
(Микрорайон Б) 4-ая Озерная	ул	63012000044007600	446402	6350		36...
(Микрорайон Б) 5-ая Озерная	ул	63012000044007700	446402	6350		36...
(Микрорайон Б) 6-ая Озерная	ул	63012000044007800	446402	6350		36...
(Микрорайон Б) 7-ая Озерная	ул	63012000044007900	446402	6350		36...
(Микрорайон В) 1-ая Лесная	ул	63012000044008000	446402	6350		36...

Грид – это позиционирование...

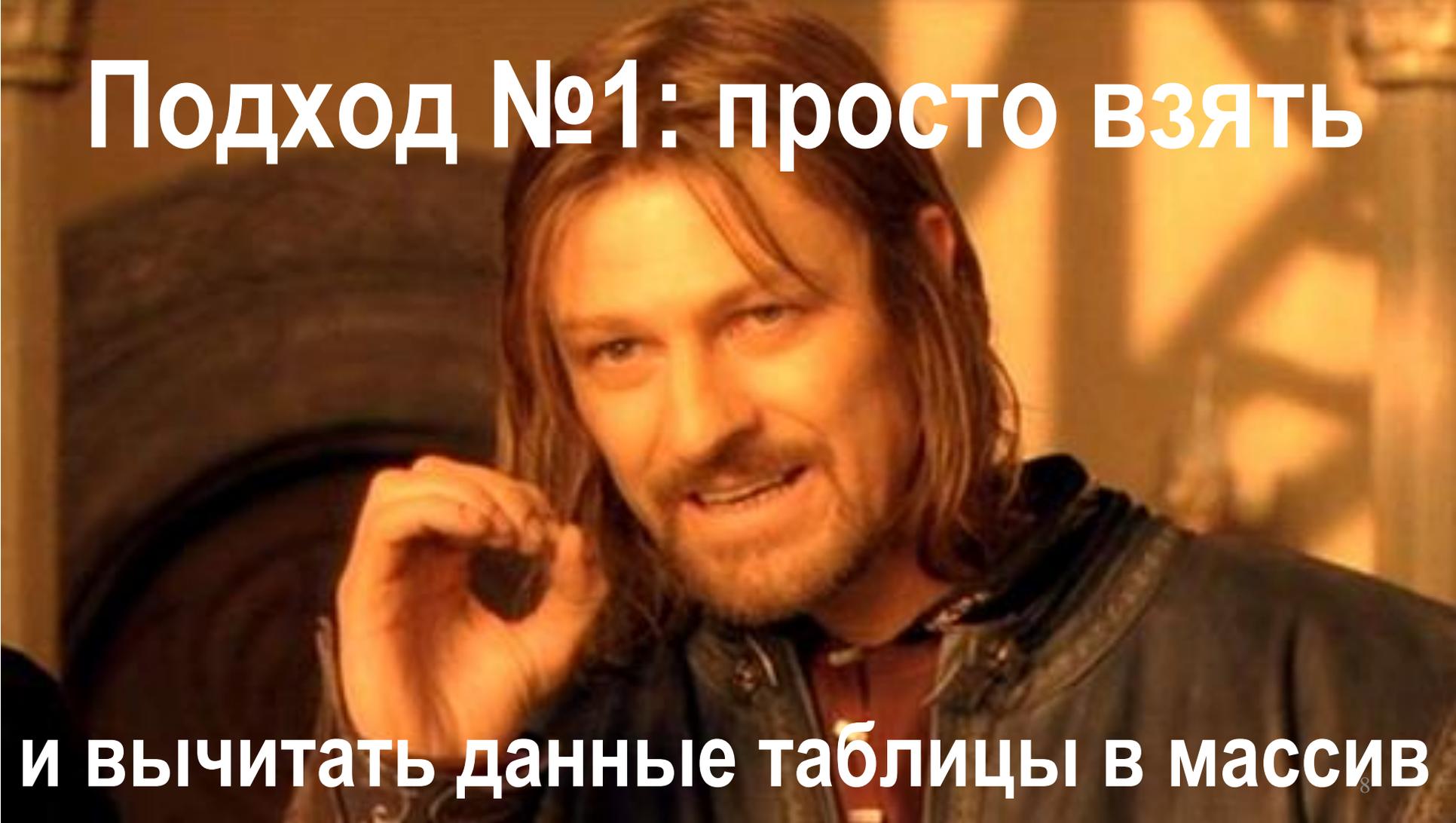
31018001000036351 >>>

name	socr	code	postalc...	gnin...	uno	oc...
(коттеджный поселок Мастрюки)	п	63028000006005400	443545	6330		36...
(Микрорайон А) 1-ая Северная	ул	63012000044006500	446402	6350		36...
(Микрорайон А) 2-ая Северная	ул	63012000044006600	446402	6350		36...
(Микрорайон А) 3-ая Северная	ул	63012000044006700	446402	6350		36...
(Микрорайон А) 4-ая Северная	ул	63012000044006800	446402	6350		36...
(Микрорайон А) 5-ая Северная	ул	63012000044006900	446402	6350		36...
(Микрорайон А) 6-ая Северная	ул	63012000044007000	446402	6350		36...
(Микрорайон А) 7-ая Северная	ул	63012000044007100	446402	6350		36...
(Микрорайон А) 8-ая Северная	ул	63012000044007200	446402	6350		36...
(Микрорайон Б) 1-ая Озерная	ул	63012000044007300	446402	6350		36...
(Микрорайон Б) 2-ая Озерная	ул	63012000044007400	446402	6350		36...
(Микрорайон Б) 3-ая Озерная	ул	63012000044007500	446402	6350		36...
(Микрорайон Б) 4-ая Озерная	ул	63012000044007600	446402	6350		36...
(Микрорайон Б) 5-ая Озерная	ул	63012000044007700	446402	6350		36...
(Микрорайон Б) 6-ая Озерная	ул	63012000044007800	446402	6350		36...
(Микрорайон Б) 7-ая Озерная	ул	63012000044007900	446402	6350		36...
(Микрорайон В) 1-ая Лесная	ул	63012000044008000	446402	6350		36...

Задача

- В нужные места на странице надо вставлять гриды, привязанные к данным таблиц.
- Быстро отображаться, быстро позиционироваться, быстро прокручиваться.
- Одинаково быстро на 10 записях и на 1 млн записей.

— А разве это сложно?

A close-up photograph of Aragorn from 'The Lord of the Rings'. He has long, wavy brown hair and a light beard. He is looking slightly to the left of the camera with a thoughtful expression, his right hand raised to his chin. The background is a warm, golden-brown color, possibly an interior setting with stone or wood.

Подход №1: просто взять

и вычитать данные таблицы в массив

Подход №1: вычитать в массив

- Прокрутка: доступ к «окну» == доступ к N-ному элементу массива — $O(1)$
- Переход к записи: бинарный поиск в отсортированном массиве — $O(\log(N))$

Проблемы:

- Первоначальная загрузка данных – $O(N)$



Проблемы:

- Первоначальная загрузка данных – $O(N)$
- «Тяжёлые» данные могут не «пролезть» в пропускную способность канала
- Данные изменяются со временем – надо периодически обновлять массив

Подход №2:

Переложим задачу на СУБД?

- **Хорошая новость:** извлечение по ключу — «лёгкая» операция

SELECT ... WHERE k >= ... LIMIT ...

Переход к записи по ключу работать будет быстро!

Подход №2:

Переложим задачу на СУБД?

Плохие новости:

- Позиционирование:

SELECT COUNT(*) WHERE k<... для определения диапазона полосы прокрутки и позиционирования.

- Прокрутка: выбираем записи с N-ной:

SELECT ... OFFSET ... LIMIT ...

Подход №2:

Переложим задачу на СУБД?

- **Имитация OFFSET:**

- **Oracle before 12:** **WITH** a **AS**

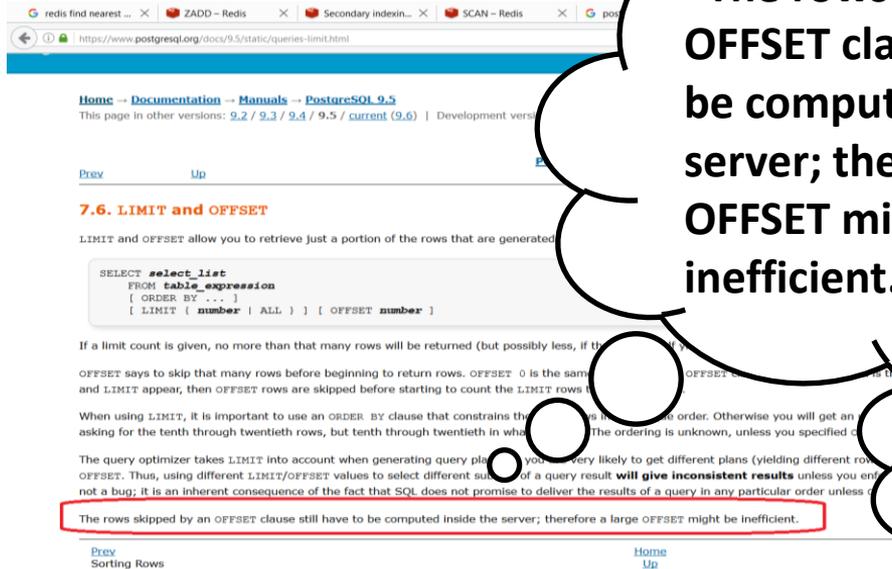
- (*YOUR_QUERY_HERE_WITH_ORDER_BY*)

- SELECT * FROM (SELECT a.*, ROWNUM rnum FROM a WHERE rownum <= limit) WHERE rnum >= offset ORDER BY rnum**

- **MS SQL Server before 2012:** **WITH** a **AS** (**SELECT ROW_NUMBER()**
OVER (ORDER BY %s) AS [limit_row_number], %s FROM %s) **SELECT ***
FROM a WHERE [limit_row_number] >= offset AND
[limit_row_number] < offset + limit

Проблемы:

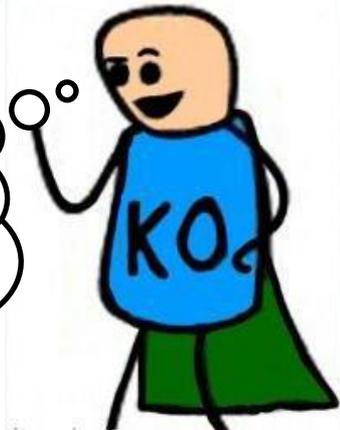
- COUNT и OFFSET — «дорогие» операции



The screenshot shows a web browser displaying the PostgreSQL 9.5 documentation page for LIMIT and OFFSET. The page title is "7.6. LIMIT and OFFSET". The text explains that LIMIT and OFFSET allow retrieving a portion of rows. A SQL query is shown: `SELECT select_list FROM table_expression [ORDER BY ...] [LIMIT (number | ALL)] [OFFSET number]`. A red box highlights the sentence: "The rows skipped by an OFFSET clause still have to be computed inside the server; therefore a large OFFSET might be inefficient."

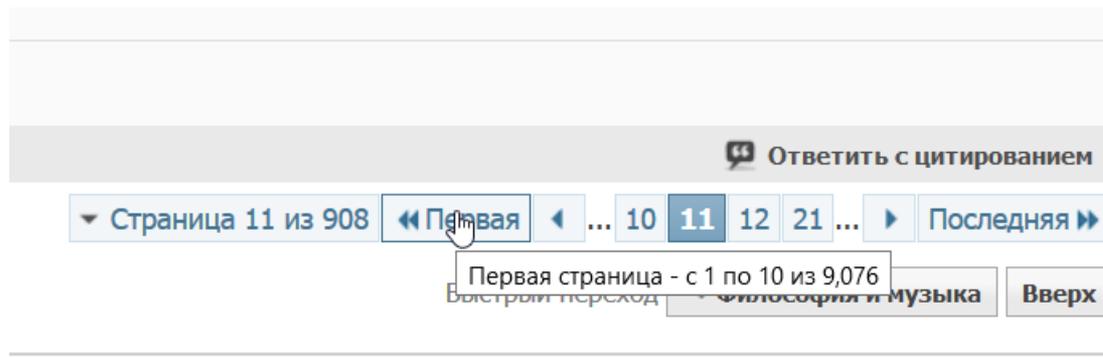
“The rows skipped by an OFFSET clause still have to be computed inside the server; therefore a large OFFSET might be inefficient.”

— Но чем мы ближе к концу набора, тем больше OFFSET!



– А зачем нам вообще грид со всеми записями?!

- Пагинация



– А зачем нам вообще грид со всеми записями?!

- Пагинация
- Фильтрация и показ только лимитированного количества записей

Browse history

From year (and earlier): From month (and earlier): Tag filter:

For any version listed below, click on its date to view it. For more help, see [Help:Page history](#) and [Help:Edit summary](#).

External tools: [Revision history statistics](#) · [Revision history search](#) · [Edits by user](#) · [Number of watchers](#) · [Page view statistics](#)

(cur) = difference from current version, (prev) = difference from preceding version,

m = [minor edit](#), [→](#) = [section edit](#), [←](#) = [automatic edit summary](#)

([newest](#) | [oldest](#)) View ([newer 500](#) | [older 500](#)) ([20](#) | [50](#) | [100](#) | [250](#) | [500](#))

Compare selected revisions

Java (programming language)

- [\(cur | prev\)](#) 14:44, 8 November 2014 [Gogoplata1234](#) ([talk](#) | [contribs](#)) . . (63,250 bytes) (+1) . . ([→External links](#))
- [\(cur | prev\)](#) 14:44, 8 November 2014 [Gogoplata1234](#) ([talk](#) | [contribs](#)) . . (63,249 bytes) (+102) . . ([→External links](#))

– А зачем нам вообще грид со всеми записями?!

- Пагинация
- Фильтрация и показ только лимитированного количества записей
- “Load more”

[And more...](#)



Today

Бывает и так:

Создать ▾

Поиск (Ctrl+F) × 🔍 ▾

Еще ▾

Дата ↓	Номер	Тип документа	Покупатель
📄✅ 18.08.2012 12:25:55	000000010	Продажа	Попов Б.В. ИЧП
📄✅ 03.09.2012 14:25:02	000000011	Заказ	Магазин "Продукты"
📄✅ 03.09.2012 14:25:15	000000011	Продажа	Магазин "Продукты"
📄✅ 04.09.2012 10:15:25	000000012	Заказ	Магазин "Мясная л...
📄✅ 04.09.2012 10:20:25	000000012	Продажа	Магазин "Мясная л...
📄✅ 05.09.2012 12:00:00	000000013	Заказ	Магазин "Мясная л...
📄✅ 05.09.2012 12:05:00	000000013	Продажа	Магазин "Мясная л...

⏪ ⏩ ⏴ ⏵

А где
полоса
прокрутки?

(Обсуждение темы:
<https://habrahabr.ru/post/280157/>)

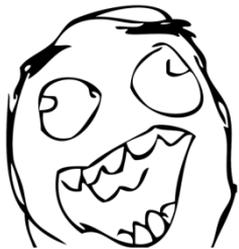
— Может, и не надо?



Почему нужен «честный» грид?

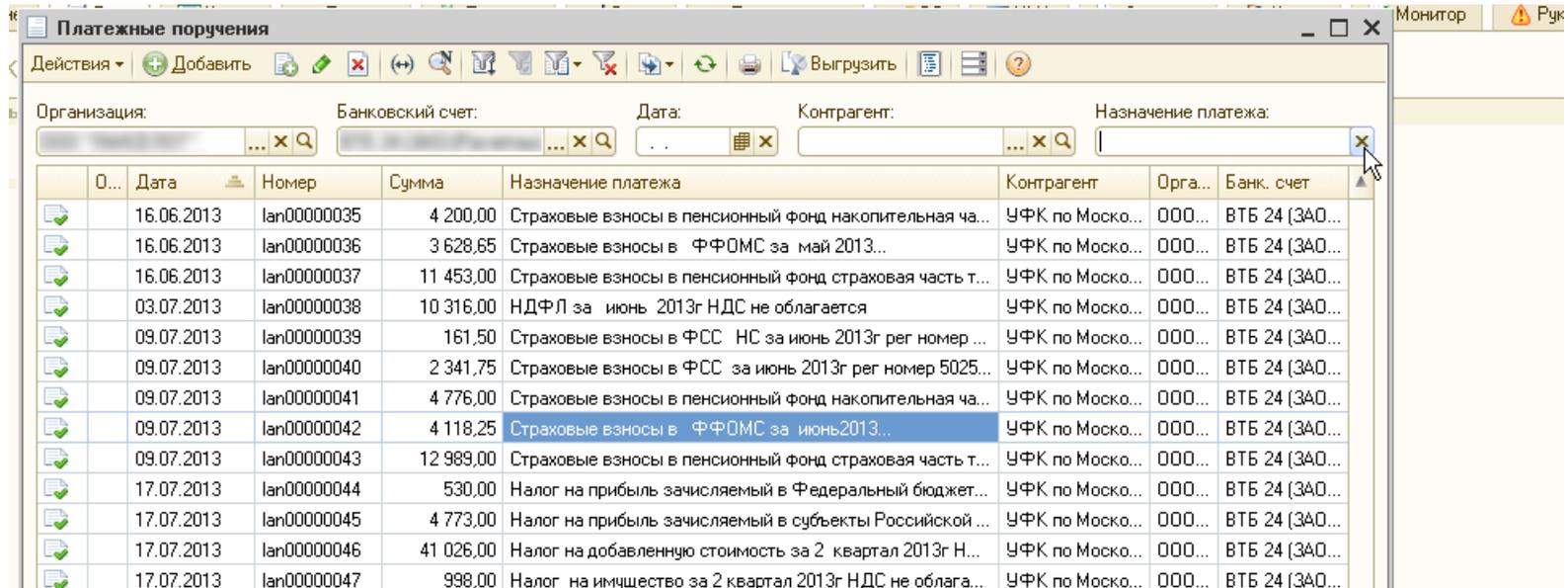
1. Пользователи любят «простыни» данных

	A	B	C	D	E	F	G	H	I
1	0.90983738	0.563257205	0.160316515	0.457405746	0.369878661	0.872872816	0.515589926	0.195094861	0.042931
2	0.542981632	0.595087871	0.76799871	0.021583036	0.040989111	0.24286559	0.988874346	0.063394357	0.689947
3	0.285343924	0.061461795	0.265644893	0.118469725	0.491143329	0.286727461	0.518108415	0.776093716	0.613382
4	0.698247699	0.005729436	0.724948119	0.796694362	0.092881722	0.675423692	0.319078579	0.250201724	0.341477
5	0.950188289	0.381581914	0.032495581	0.988311101	0.082168787	0.720500424	0.777373385	0.31168398	0.369843
6	0.578500621	0.542396827	0.665258157	0.001207081	0.344137258	0.279090066	0.31906997	0.270353815	0.221351
7	0.549158698	0.228342657	0.165990208	0.244790708	0.523717894	0.253973081	0.874552223	0.517722675	0.68537
8	0.501595505	0.328806588	0.901740079	0.465051919	0.699231313	0.456325928	0.12227123	0.894104638	0.305222
9	0.409792706	0.472870427	0.741041558	0.707458623	0.069852039	0.184184032	0.204891175	0.876914772	0.865390
10	0.369591708	0.376881093	0.076488033	0.701150703	0.6813041	0.23156523	0.952057645	0.690822191	0.747280
11	0.658742994	0.546816584	0.076042987	0.170513015	0.239139848	0.834982635	0.589965679	0.730445397	0.969821
12	0.805539126	0.820117717	0.493856624	0.67022981	0.862128947	0.655326942	0.374701671	0.005274284	0.633379
13	0.202498992	0.796463353	0.610873785	0.523595269	0.415965892	0.450622764	0.956110463	0.820536608	0.076016
14	0.843254074	0.892755123	0.699376138	0.173778596	0.284430371	0.177823059	0.215576667	0.647462841	0.632100
15	0.953722268	0.652791918	0.391285415	0.535375726	0.790207436	0.401798903	0.50606727	0.450118789	0.571827
16	0.309287626	0.116508069	0.705692078	0.57019095	0.547901005	0.498690475	0.749268829	0.477021562	0.139002
17	0.355271537	0.564699735	0.746027405	0.649178845	0.335571276	0.590417446	0.294089424	0.896030582	0.9211
18	0.348250065	0.17238299	0.703389808	0.873356533	0.288949755	0.657854212	0.602364188	0.253642084	0.717568
19	0.952232776	0.83254704	0.763956394	0.820107504	0.913199658	0.821872371	0.153022992	0.966869885	0.150394
20	0.145493053	0.499127855	0.374162902	0.508620825	0.869201668	0.361629016	0.699483556	0.962296654	0.610496
21	0.158405274	0.603124598	0.238687278	0.436033174	0.177229137	0.421493775	0.176565073	0.177646735	0.262379
22	0.05947856	0.937518047	0.817667692	0.695830101	0.428518253	0.923357015	0.014921334	0.06856301	0.183860
23	0.893381382	0.034188105	0.892410766	0.263377744	0.741296532	0.940262057	0.376371455	0.308555922	0.404358
24	0.530952234	0.800449911	0.741122982	0.406342676	0.144129174	0.018914167	0.244840265	0.92374018	0.10072
25	0.792497926	0.741675221	0.695526275	0.89711495	0.447454952	0.456679816	0.027557169	0.837701511	0.522445
26	0.877548549	0.691377509	0.174567646	0.953046054	0.668233577	0.864362997	0.242501972	0.069158766	0.877325



Почему нужен «честный» грид?

1. Пользователи любят «простыни» данных
2. Нужно видеть соседние записи



Платежные поручения

Действия ▾ + Добавить + [Иконки] Выгрузить [Иконки] ?

Организация: [Поиск] Банковский счет: [Поиск] Дата: [Календарь] Контрагент: [Поиск] Назначение платежа: [Поиск]

О...	Дата	Номер	Сумма	Назначение платежа	Контрагент	Орга...	Банк. счет
[Иконка]	16.06.2013	Ian00000035	4 200,00	Страховые взносы в пенсионный фонд накопительная ча...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	16.06.2013	Ian00000036	3 628,65	Страховые взносы в ФФОМС за май 2013...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	16.06.2013	Ian00000037	11 453,00	Страховые взносы в пенсионный фонд страховая часть т...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	03.07.2013	Ian00000038	10 316,00	НДФЛ за июнь 2013г НДС не облагается	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	09.07.2013	Ian00000039	161,50	Страховые взносы в ФСС НС за июнь 2013г рег номер ...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	09.07.2013	Ian00000040	2 341,75	Страховые взносы в ФСС за июнь 2013г рег номер 5025...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	09.07.2013	Ian00000041	4 776,00	Страховые взносы в пенсионный фонд накопительная ча...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	09.07.2013	Ian00000042	4 118,25	Страховые взносы в ФФОМС за июнь2013...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	09.07.2013	Ian00000043	12 989,00	Страховые взносы в пенсионный фонд страховая часть т...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	17.07.2013	Ian00000044	530,00	Налог на прибыль зачисляемый в Федеральный бюджет...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	17.07.2013	Ian00000045	4 773,00	Налог на прибыль зачисляемый в субъекты Российской ...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	17.07.2013	Ian00000046	41 026,00	Налог на добавленную стоимость за 2 квартал 2013г Н...	УФК по Моско...	000...	ВТБ 24 (ЗАО...
[Иконка]	17.07.2013	Ian00000047	998,00	Налог на имущество за 2 квартал 2013г НДС не облага...	УФК по Моско...	000...	ВТБ 24 (ЗАО...

Почему нужен «честный» грид?

1. Пользователи любят «простыни» данных
2. Нужно видеть соседние записи
3. Переход к связанному списку

Notes Links Refresh Clear Filter Find
 Show Attached Page

Item Ledger Entries -

Entry Type	Document Type	Document No.	Item No.	Description
Transfer	Direct Trans...	RPT055970	A1956138A	
Transfer	Direct Trans...	RPT055970	A1961821A	
Transfer	Direct Trans...	RPT055970	A1961821A	
Transfer	Direct Trans...	RPT055970	145866321	
Transfer	Direct Trans...	RPT055970	145866321	
Transfer	Direct Trans...	RPT055970	A1944968A	
Transfer	Direct Trans...	RPT055970	A1944968A	
Transfer	Direct Trans...	RPT055971	A1961829A	
Transfer	Direct Trans...	RPT055971	A1961829A	
Transfer	Direct Trans...	RPT055972	A1961894A	
Transfer	Direct Trans...	RPT055972	A1961894A	
Transfer	Direct Trans...	RPT055973	A1944968A	
Transfer	Direct Trans...	RPT055973	A1944968A	
Transfer	Direct Trans...	RPT055973	A1961829A	
Transfer	Direct Trans...	RPT055973	A1961829A	
Transfer	Direct Trans...	RPT055974	A2042891A	
Transfer	Direct Trans...	RPT055974	A2042891A	
Transfer	Direct Trans...	RPT055974	A1961837A	
Transfer	Direct Trans...	RPT055974	A1961837A	
Transfer	Direct Trans...	RPT055974	A1886320A	
Transfer	Direct Trans...	RPT055974	A1886320A	
Transfer	Direct Trans...	RPT055974	A1886320A	

Select - Item List

HOME
ACTIONS
NAVIGATE
REPORT

New

Edit

View

Delete

Sales Prices

Requisition Worksheet

Item Journal

Adjust Cost - Item Entries

Statistics

Item G/L Turnover

Inventory Availability

Inventory - Top 10 List

Price List

Show as List

Show as Chart

Notes

Links

New
Manage
Process
Report
View
Show Attached

Item List -

Type to filter (F3) No.

No filters applied

No.	Description	Su... Exist	Ass... BOM	Production BOM No.	Routing No.	Base Unit of Measure	Cost is A...	Unit Cost
A1944260A	MOUNTED C.BOARD, VPR	No	No			PCS	<input checked="" type="checkbox"/>	0,00
A1944508A	HB MOUNT	No	No			PCS	<input checked="" type="checkbox"/>	0,00
A1944565A	SS-S20//C(E2BR)	No	No			PCS	<input type="checkbox"/>	1 387,96
A1944589A	REMOTE COMMANDER (RM	No	No			PCS	<input type="checkbox"/>	1 501,85
A1944590A	REMOTE COMMANDER (RM	No	No			PCS	<input type="checkbox"/>	1 392,74
A1944591A	REMOTE COMMANDER (RM	No	No			PCS	<input type="checkbox"/>	426,84
A1944667A	MOUNTED C.BOARD, AU-	No	No			PCS	<input type="checkbox"/>	1 537,68
A1944671A	MOUNTED C.BOARD, VPR	No	No			PCS	<input type="checkbox"/>	27 697,50
A1944802A	MOUNTED C. BOARD,RE-	No	No			PCS	<input type="checkbox"/>	16 516,22
A1944805A	MOUNTED C.BOARD,DPR-	No	No			PCS	<input type="checkbox"/>	120 785,04
A1944967A	HDD(D500GB,HTS545050	No	No			PCS	<input type="checkbox"/>	2 576,29
A1944968A	HDD(D750GB,HTS541075	No	No			PCS	<input type="checkbox"/>	2 847,49

	2316200507	1	1	1	0,00	1 450,00	0,00	<input checked="" type="checkbox"/>	Transfer
	MOSCOW	-1	-1	0	0,00	-4 884,11	0,00	<input type="checkbox"/>	Transfer
	2316200507	1	1	1	0,00	4 884,11	0,00	<input checked="" type="checkbox"/>	Transfer

Как справляются другие ребята?

— С переменным успехом.

- PgAdmin3 — виснет
- SQL Server Management Studio — по умолчанию ограничивает число строк
- инструменты DB2, ERP Microsoft Navision (ок. 2003 г.) — сразу открывают и «крутят» любые таблицы

Наше решение

- Disclaimer:
 - Не «серебряная пуля»
 - Не всё просто с сортировкой и фильтрацией
 - Нужны правильные индексы, LIKE с осторожностью
- Но:
 - Неплохой компромисс между скоростью и удобством

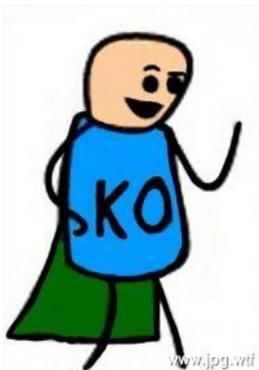
Временное упрощение задачи (потом ограничения снимем!)

1. Первичный ключ таблицы состоит из одного поля k с типом INT
2. Сортировка по первичному ключу

```
CREATE TABLE test (  
    k INT NOT NULL PRIMARY KEY,  
    descr VARCHAR(20)  
);
```

Мысленный эксперимент

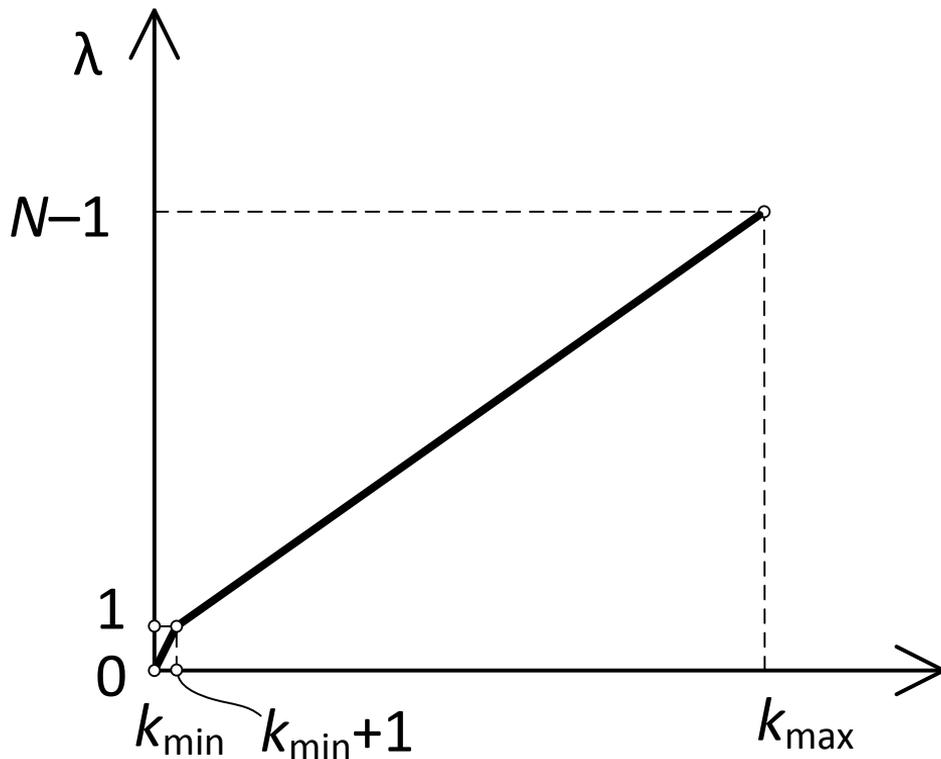
1. У 1-й записи $k = 0$.
 2. У 1 000-й записи $k = 10\,000$.
 3. Записи отсортированы по k .
- Вопрос: какое *примерно* значение k будет иметь запись с номером 500?



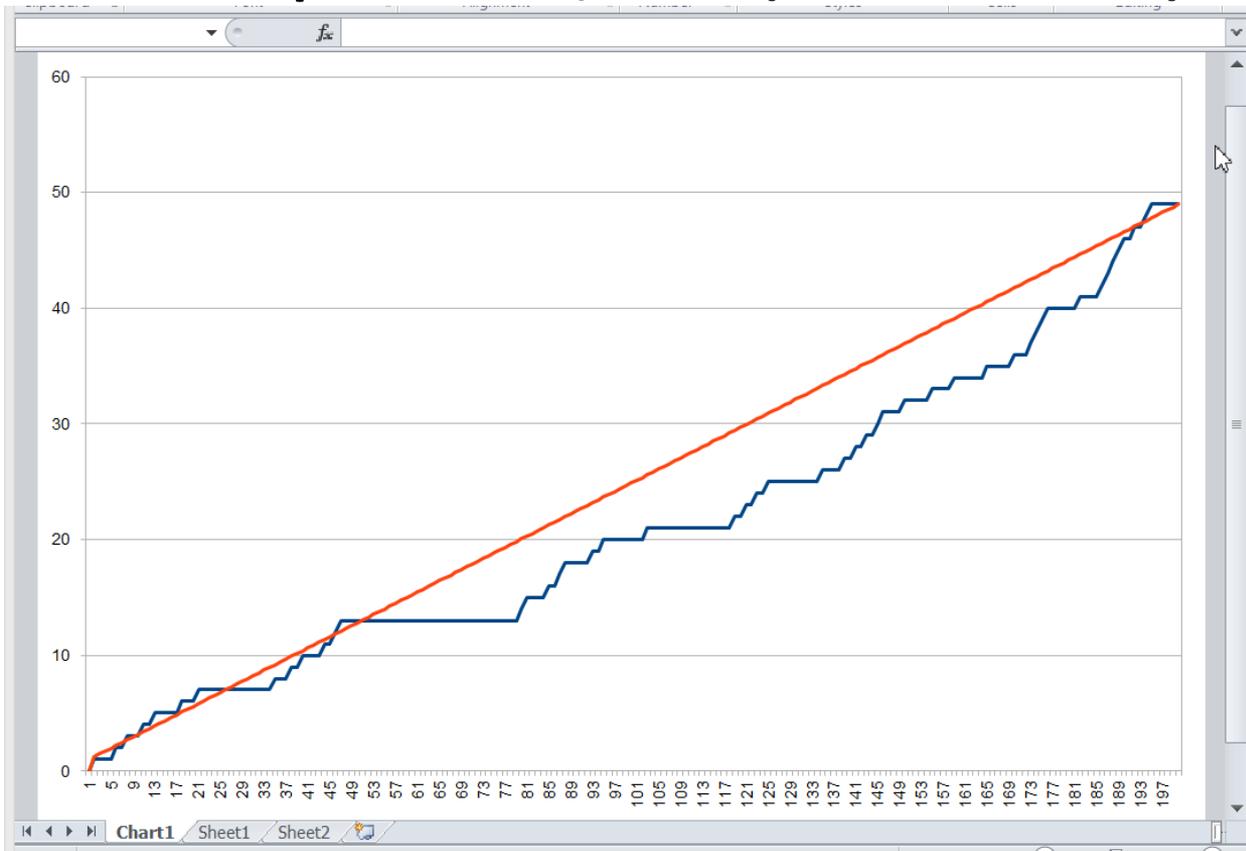
Ответ: наверное,
что-то около $\frac{k_{max} + k_{min}}{2} = 5000\dots$

Идея

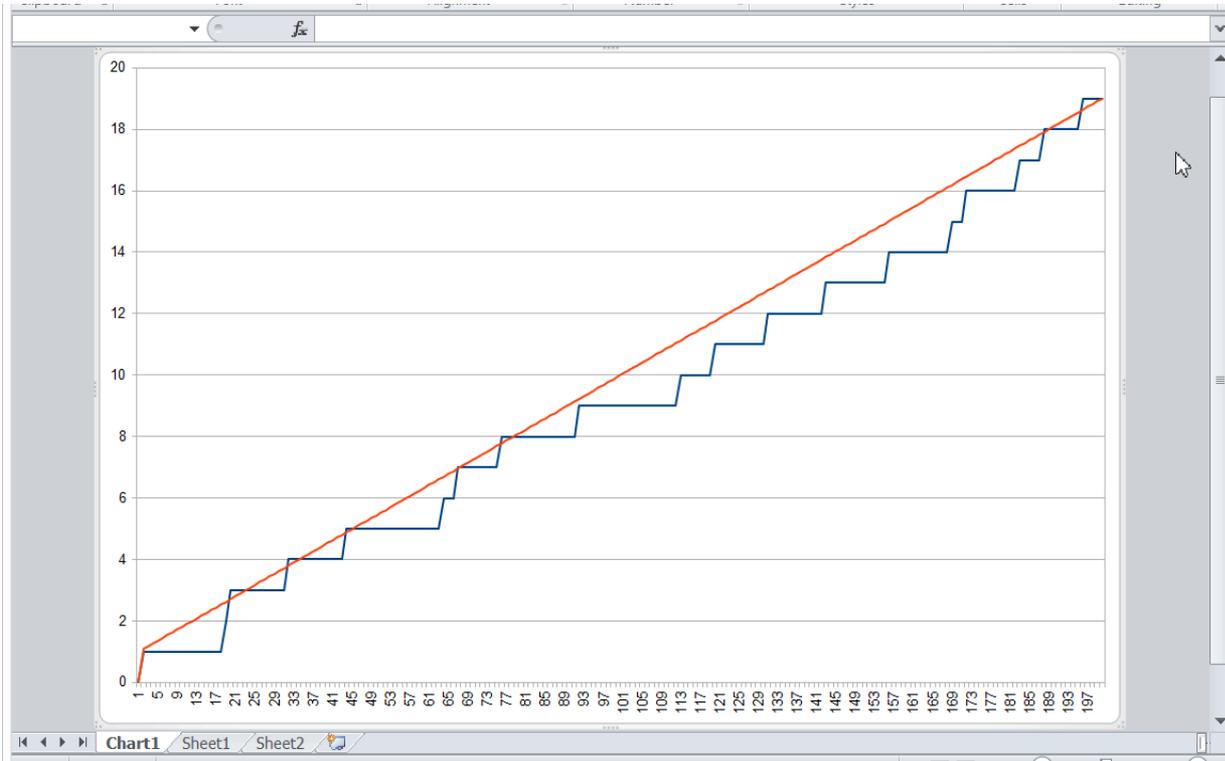
«Угадаем» взаимосвязь между ключом записи k и её порядковым номером λ при помощи линейной интерполяции?



Насколько хороша линейная интерполяция? (50 из 200)



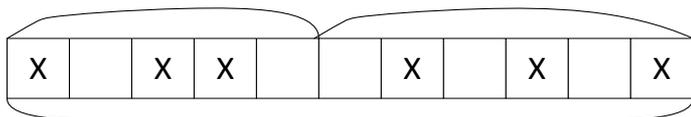
Случай 10 из 200 — похуже...



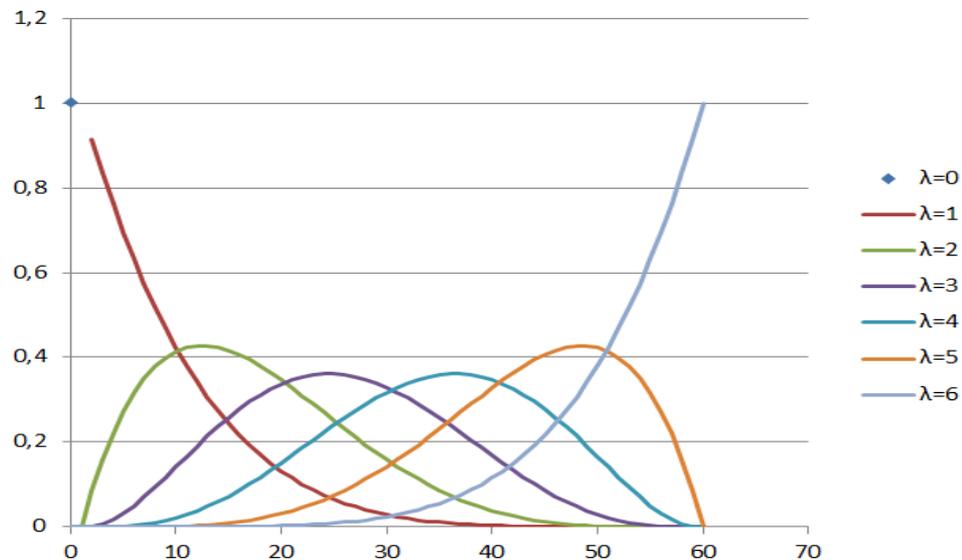
Вероятность наличия ровно λ записей с ключом $<k$

$$C_{\lambda-1}^{k-k_{\min}-1}$$

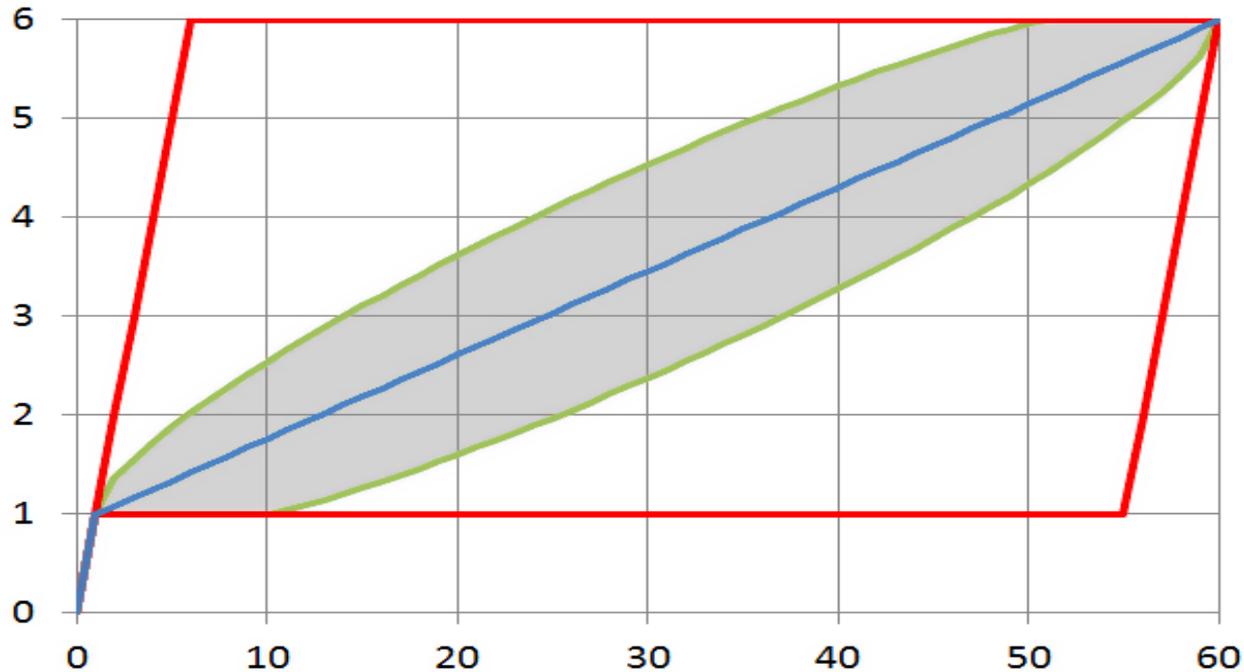
$$C_{\lambda_{\max}-\lambda}^{k_{\max}-(k-k_{\min})}$$



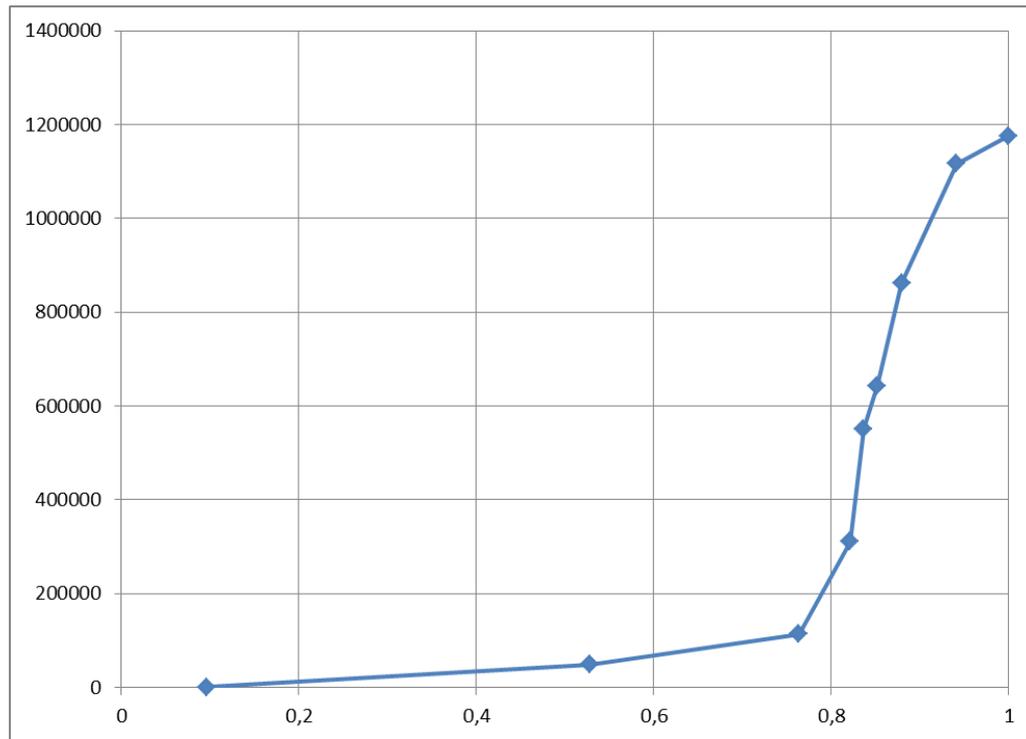
$$C_{\lambda_{\max}-1}^{k_{\max}-k_{\min}-1}$$



Теоретическая оценка границ, мат. ожидания и СКО



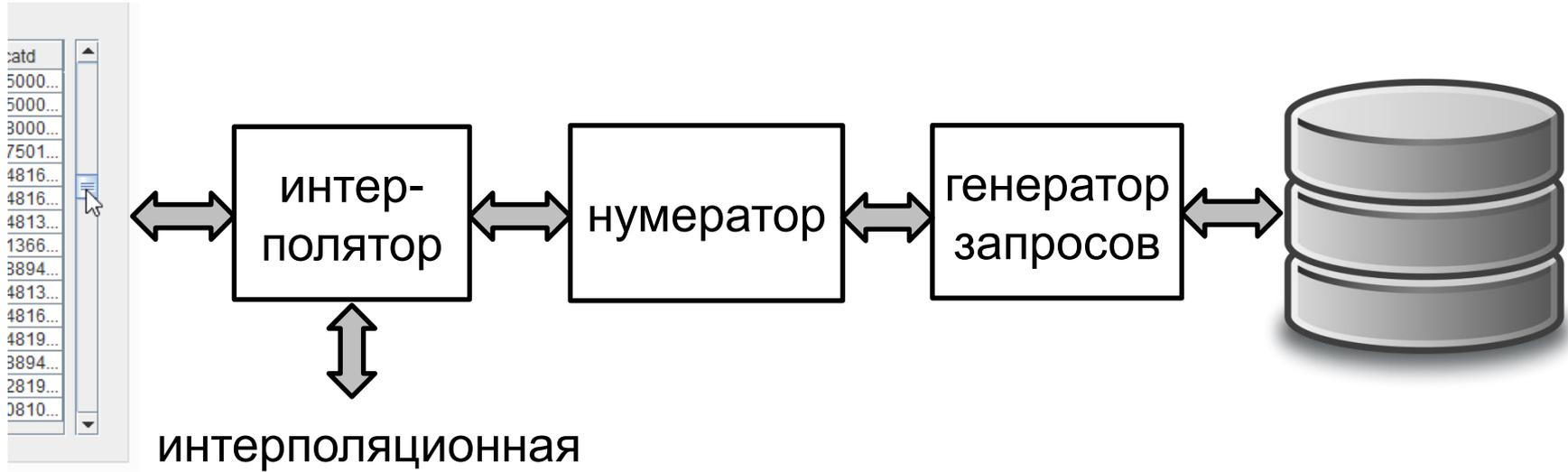
Неравномерное распределение ключей — кусочная интерполяция!



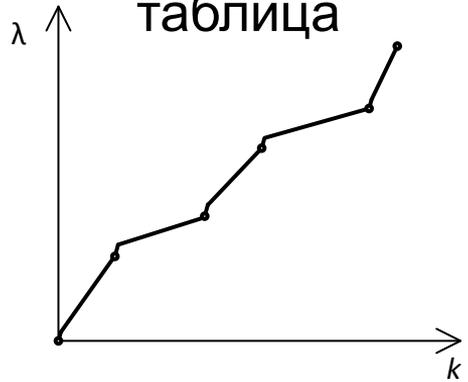
Снимаем ограничения

- Ключ составной
 - нумеруем возможные комбинации так, чтобы описывались (большим) целым числом
- Нужна сортировка
 - добавляем сортировочное поле к началу списка ключевых
- Ключ содержит не только INT-поля
 - нумеруем значения (большими) целыми!

Компоненты алгоритма



интерполяционная
таблица



Пример

- Справочник улиц КЛАДР
- 1 175 430 записей
- ORDER BY name, code
- **CREATE INDEX** ix_street **ON** kladr.street (name, code)

Пользователь прокрутил записи...

БД

запрос
ближайшей записи

Нумератор

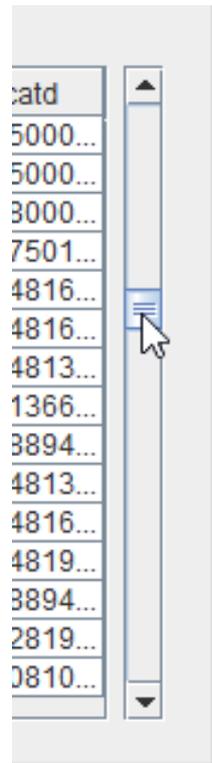
вычисление
ключа

Интерполятор

вычисление
порядкового номера

Полоса
прокрутки

изменилась
позиция бегунка

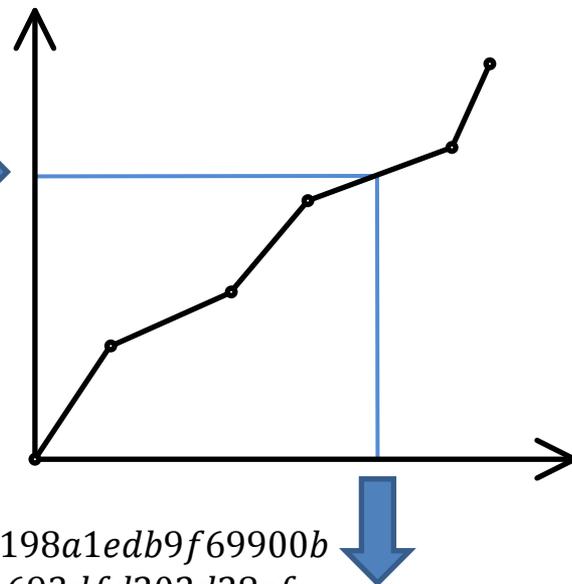


$$\lambda = 584600$$

Интерполяцией находим порядковый номер ключа



$\lambda = 584600$ →



$k \approx 5714fdd05a1ecb7198a1edb9f69900b$
 $d1327480a800fffbb693dfd202d28cf$
 $826094773573a1630cdba9d519d7$
 $d12e577968ca7fd8210df9c34811b7d06175068c$

Вычисляем примерные значения полей ключа по примерному номеру

$k \approx$ 5714fdd05a1ecb7198a1edb9f69900bd1327480a800
fffb693dfd202d28cf826094773573a1630cdba9d51
9d7d12e577968ca7fd8210df9c34811b7d06175068c

БД

запрос
ближайшей записи

Нумератор

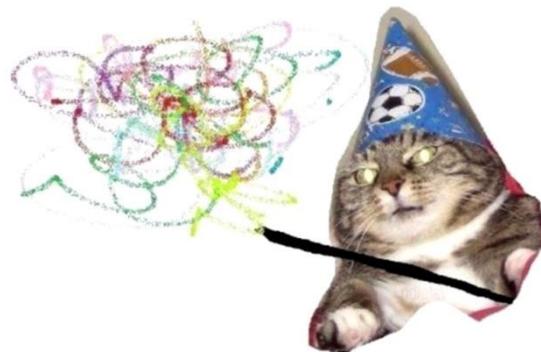
вычисление
ключа

Интерполятор

вычисление
порядкового номера

Полоса
прокрутки

изменилась
позиция бегунка



name \approx Мд'Uкр ®\$-'•ды9uj?;!AOZ#У>:"ZG818*\$%o9Ювэ'
code \approx я\$зес\$}0,&-»гФ4%эR

Запрашиваем записи, ближайшие к найденным примерным значениям полей

БД

запрос
ближайшей записи

Нумератор

вычисление
ключа

Интерполятор

вычисление
порядкового номера

Полоса
прокрутки

изменилась
позиция бегунка

name: Мд'Uкр ®\$-'•ды9uj?|AOZ#У>:"ZG818*\$%o9Ювэ'
code: я\$зес\$}0,&-»гФ4%эR

SELECT ... FROM ... WHERE
("name", "code") >= (?, ?) LIMIT ...

	name character varying(40)	socr character varying(10)	code character vary
1	МДС-136	ул	27004000006
2	Мдуль (Зеленая)	мкр	47005002000
3	Мебельная	ул	01008000041
4	Мебельная	ул	01008000043
5	Мебельная	ул	02000001000
6	Мебельная	ул	02000004000

Прокрутка в реальном времени

31018001000036351 >>>

name	socr	code	postalc...	gnin...	uno	oc...
(коттеджный поселок Мастрюки)	п	63028000006005400	443545	6330		36...
(Микрорайон А) 1-ая Северная	ул	63012000044006500	446402	6350		36...
(Микрорайон А) 2-ая Северная	ул	63012000044006600	446402	6350		36...
(Микрорайон А) 3-ая Северная	ул	63012000044006700	446402	6350		36...
(Микрорайон А) 4-ая Северная	ул	63012000044006800	446402	6350		36...
(Микрорайон А) 5-ая Северная	ул	63012000044006900	446402	6350		36...
(Микрорайон А) 6-ая Северная	ул	63012000044007000	446402	6350		36...
(Микрорайон А) 7-ая Северная	ул	63012000044007100	446402	6350		36...
(Микрорайон А) 8-ая Северная	ул	63012000044007200	446402	6350		36...
(Микрорайон Б) 1-ая Озерная	ул	63012000044007300	446402	6350		36...
(Микрорайон Б) 2-ая Озерная	ул	63012000044007400	446402	6350		36...
(Микрорайон Б) 3-ая Озерная	ул	63012000044007500	446402	6350		36...
(Микрорайон Б) 4-ая Озерная	ул	63012000044007600	446402	6350		36...
(Микрорайон Б) 5-ая Озерная	ул	63012000044007700	446402	6350		36...
(Микрорайон Б) 6-ая Озерная	ул	63012000044007800	446402	6350		36...
(Микрорайон Б) 7-ая Озерная	ул	63012000044007900	446402	6350		36...
(Микрорайон В) 1-ая Лесная	ул	63012000044008000	446402	6350		36...

Позиционирование

>>>

name	socr	code	postalcode	gninmb	uno	ocatd
за Автохозяй...	сад	5900500100...	617120	5933		57212501000
за Автохозяй...	сад	5900500100...	617120	5933		57212501000
за Автохозяй...	сад	5900500100...	617120	5933		57212501000
за Автохозяй...	сад	5900500100...	617120	5933		57212501000
за Автохозяй...	сад	5900500100...	617120	5933		57212501000
за Автохозяй...	сад	5900500100...	617120	5933		57212501000
За админист...	гск	6600001401...	624238	6681		65478000000
За админист...	гск	6600001401...	624238	6681		65478000000
За админист...	гск	6600001401...	624238	6681		65478000000
За валом	ул	6300300001...	446245	6330		36204816001
за газоразда...	сад	5900500100...	617120	5933		57212501000
за газоразда...	сад	5900500100...	617120	5933		57212501000
За д.N18г	гск	2900200100...	165152	2907		11205501000
За депо	ул	6300000100...	443036	6311		36401364000
За депо	ул	6603000100...	623900	6676		65254501000
За деревней	тер	3800600001...	666418	3827	3824	25206804001
За деревней	тер	3800600001...	666411	3827	3824	25206807000
За деревней	м	3800600002...	666411	3827	3824	25206807002
За дружбу	сад	6100000100...	344000	6165		60401000000
За ельником	м	3800600002...	666411	3827	3824	25206807002

Позиционирование

42000010000057000							>>>
name	socr	code	postalcode	gninmb	uno	ocatl	
Чаадаева	ул	42000010...	652517	4212		32419000...	
Чаадаева	ул	42000010...	652517	4212		32419000...	

БД

обращение
к таблице по ключу

запрос реального
номера записи

Нумератор

вычисление
порядкового номера

Интерполятор

позиция бегунка
(обр. интерп.)

Полоса
прокрутки

изменение позиции
бегунка

```
SELECT ... FROM ... WHERE  
("name", "code") >= (?, ?) LIMIT  
...
```

```
SELECT COUNT(*) FROM ...  
WHERE ("name", "code") < (?, ?)
```

Вычисляем порядковый номер ключа

БД

обращение
к таблице по ключу

Нумератор

вычисление
порядкового номера

Интерполятор

позиция бегунка
(обр. интерп.)

Полоса
прокрутки

изменение позиции
бегунка

name: Чаадаева

code: 42000010000057000



$k = 6049b5af6a5108492876ad372a3c5d2f6b86bebe0d7$
 $ae760e3e88d6973e5789bec40d43bfa454ed1ce24722$
 $8a08fbdf31e97b52f22dfa722022c07aa000000000$

Интерполяцией находим примерную позицию бегунка

БД

обращение
к таблице по ключу

Нумератор

вычисление
порядкового номера

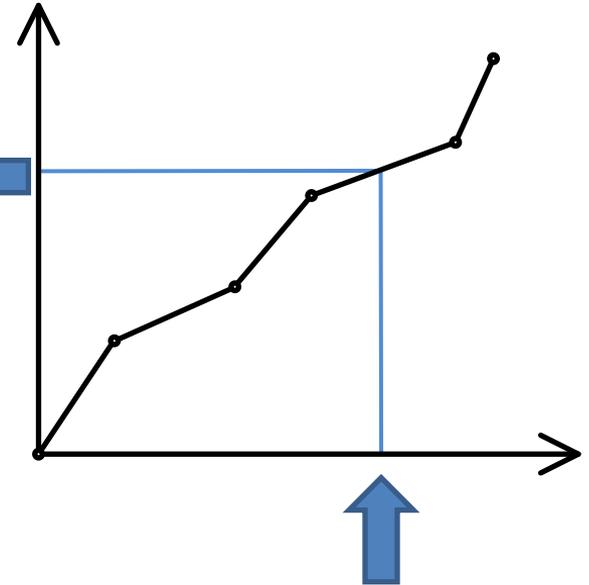
Интерpolator

позиция бегунка
(обр. интерп.)

Полоса
прокрутки

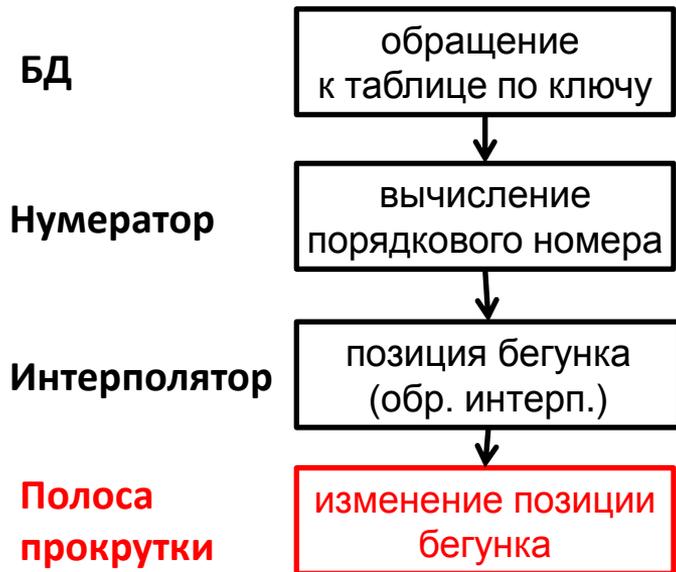
изменение позиции
бегунка

$\lambda \approx 1104307$



$\kappa = 6049b5af6a5108492876ad372a3c5d2f$
 $6b86bebe0d7ae760e3e88d6973e5789bec4$
 $0d43bfa454ed1ce247228a08fbd31e97b52$
 $f22dfa722022c07aa000000000$

Выставляем бегунок на позицию



itd
100...
100...
100...
370...
100...
338...
344...
100...
100...
100...
308...
382...
100...
100...
328...
346...
340...
375...
375...
362...

запрос реального номера записи

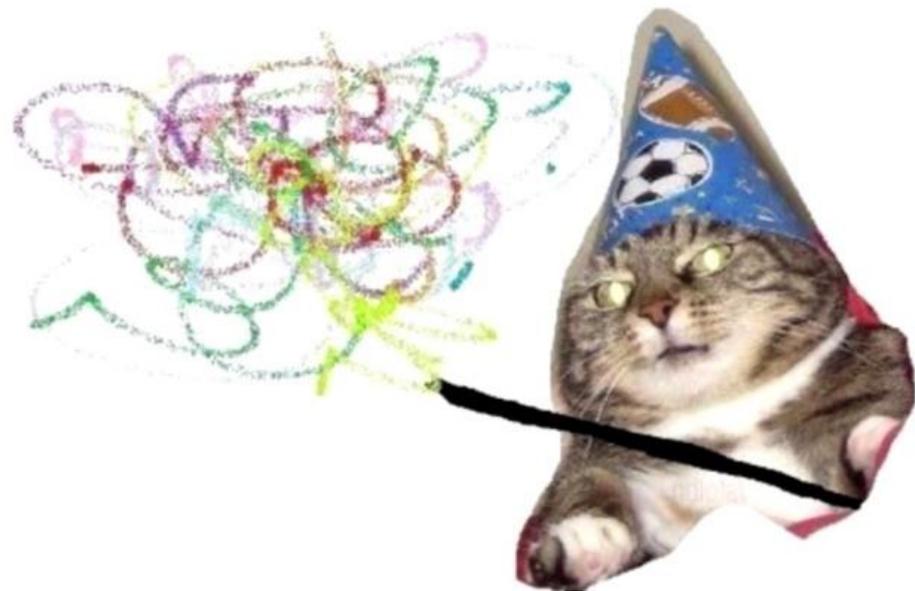


$\lambda = 1029031$

$\lambda \approx 1104307$

Осталось разобраться в одном вопросе...

Как работает нумератор?

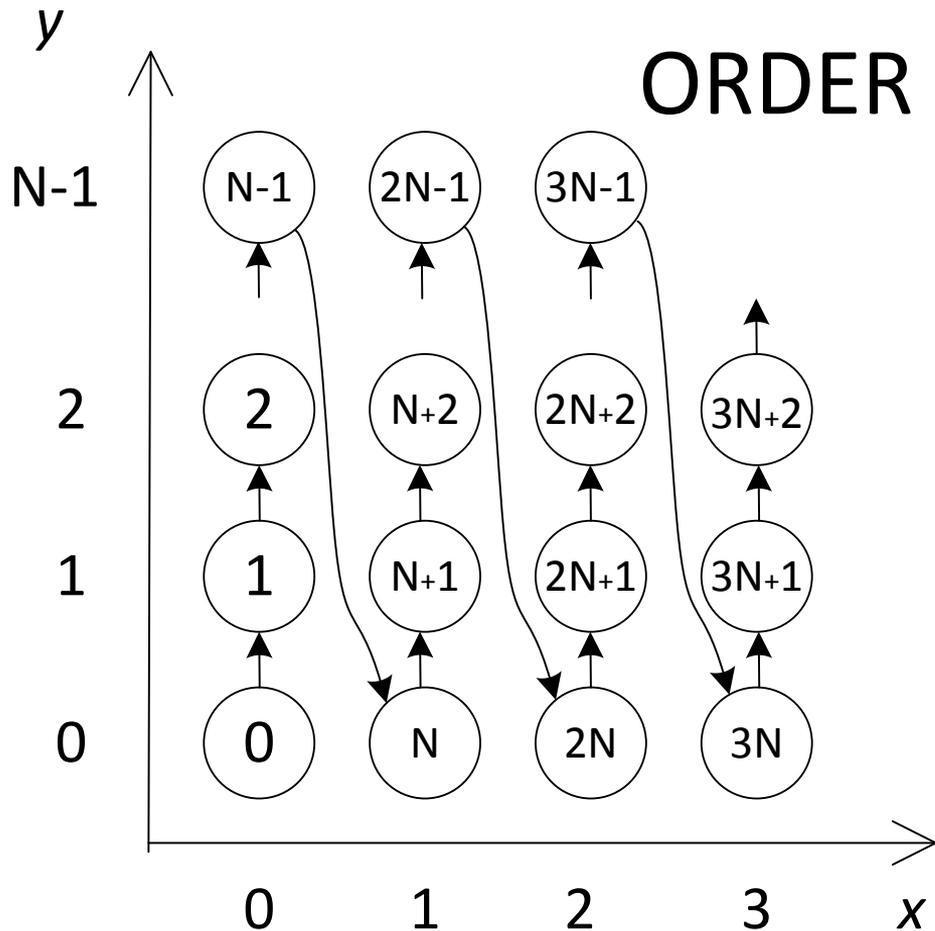


Функция нумератора

- ord — обратимая функция
- тип аргумента — широкий диапазон типов или наборов значений (у которых есть только сравнение, Comparable)
- тип результата — BigInteger (в нём есть арифметика, нужная для интерполяции)
- $k < l \Leftrightarrow ord(k) < ord(l)$
- $(k_1, \dots, k_n) < (l_1, \dots, l_n) \Leftrightarrow ord(k_1, \dots, k_n) < ord(l_1, \dots, l_n)$

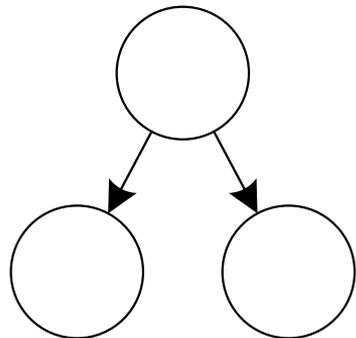
- С BIT и INT всё просто — они нумеруют сами себя
- Составные ключи?
- Строки?

Составной ключ/сортировка: ORDER BY X, Y



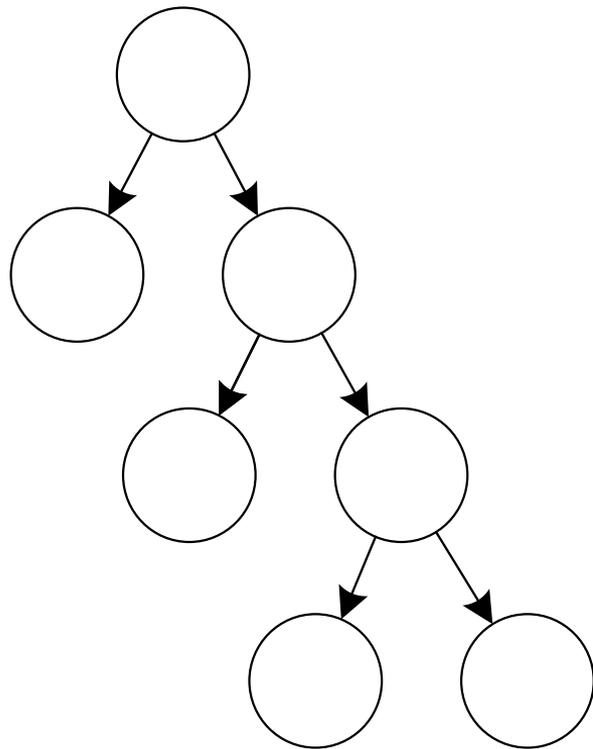
- Туда: $\kappa = Nx + y$
- Обратно:
 1. $x = \left\lfloor \frac{\kappa}{N} \right\rfloor$
 2. $y = \kappa \bmod N$

Составной ключ



Если умеем работать
с парами значений —

Составной ключ

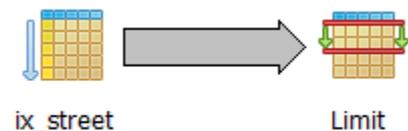
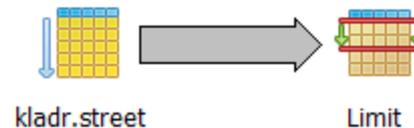
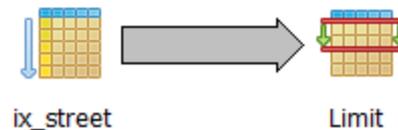


Если умеем работать
с парами значений —

— то можем работать
и с произвольными N-ками
значений

Как отыскивать записи от составного ключа?

- where $(k_1, k_2) \geq (K_1, K_2)$
(работает только в Postgres)
- where $k_1 > K_1$ or $(k_1 = K_1$ and $(k_2 \geq K_2))$
не использует индекс!
- where $k_1 \geq K_1$ and $(k_1 > K_1$ or $(k_2 \geq K_2))$



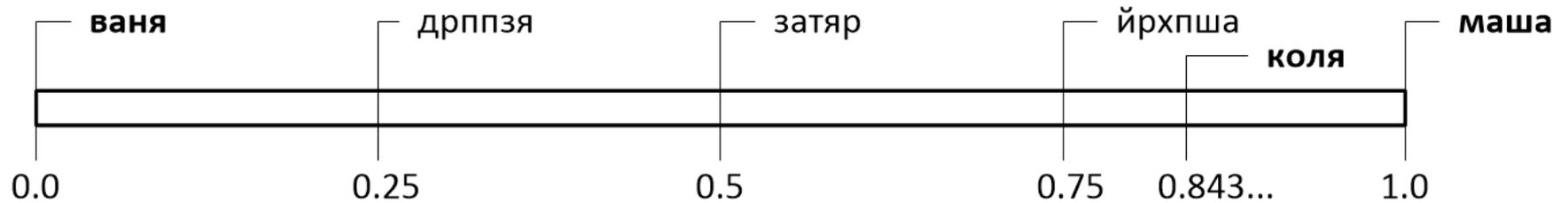
Нумерация строк

0

ВАНЯ



Интерполяция строк



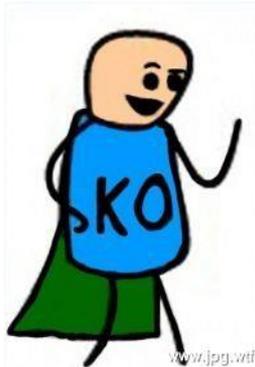
Сколько всего строк длины не больше t
в алфавите из a символов?

(a^t — неправильный ответ,
это число строк длины *ровно* t).

Сколько всего строк длины не больше m
в алфавите из a символов?

Пустая строка, строка из одной буквы, из двух букв...

$$1 + a + a^2 + \dots + a^m =$$

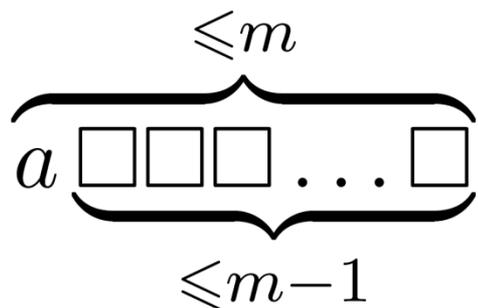


$$= \frac{a^{m+1} - 1}{a - 1} \approx O(a^m)$$

Каков порядковый номер строки $(c_0, c_1, \dots, c_{l-1})$ в словарно-упорядоченном пространстве строк длины не больше m ?

" — 0, 'а' — 1, 'б' — ?

2 — неправильный ответ: 'абак', 'араб', 'арка'...



$$ord(c_0) = 1 + \frac{a^m - 1}{a - 1} c_0$$

Каков порядковый номер строки $(c_0, c_1, \dots, c_{l-1})$ в словарно упорядоченном пространстве строк длины не больше m ?

- Общая формула:

$$F = l + \sum_{i=0}^{l-1} \frac{a^{m-i} - 1}{a - 1} c_i$$

- Постаравшись, можно найти алгоритм обратного вычисления строки

Ничего не упустили?..

Пишем тест:

ИЖЕВСК	297544850130486
--------	-----------------

ИРКУТСК	309090873152681
---------	-----------------

ЙОШКАР-ОЛА	343626219328684
------------	-----------------

Реальность:

SELECT name **FROM** city **ORDER BY** name
(PostgreSQL с настройками по умолчанию)

122	И
123	Й
124	Ижевск
125	Йошкар-Ола
126	Иркутск
127	К

Правила сортировки (collation rules)

`java.text.RuleBasedCollator` extends `java.text.Collator`

Результат «подгонки» правил к PostgreSQL:

`...<в,В<г,Г<д,Д<е,Е;ё,Ё<ж,Ж<з,З<и,И;й,Й<к,К<л,Л...`

`java.text.CollationKey`

Строка с точки зрения Collation Rule

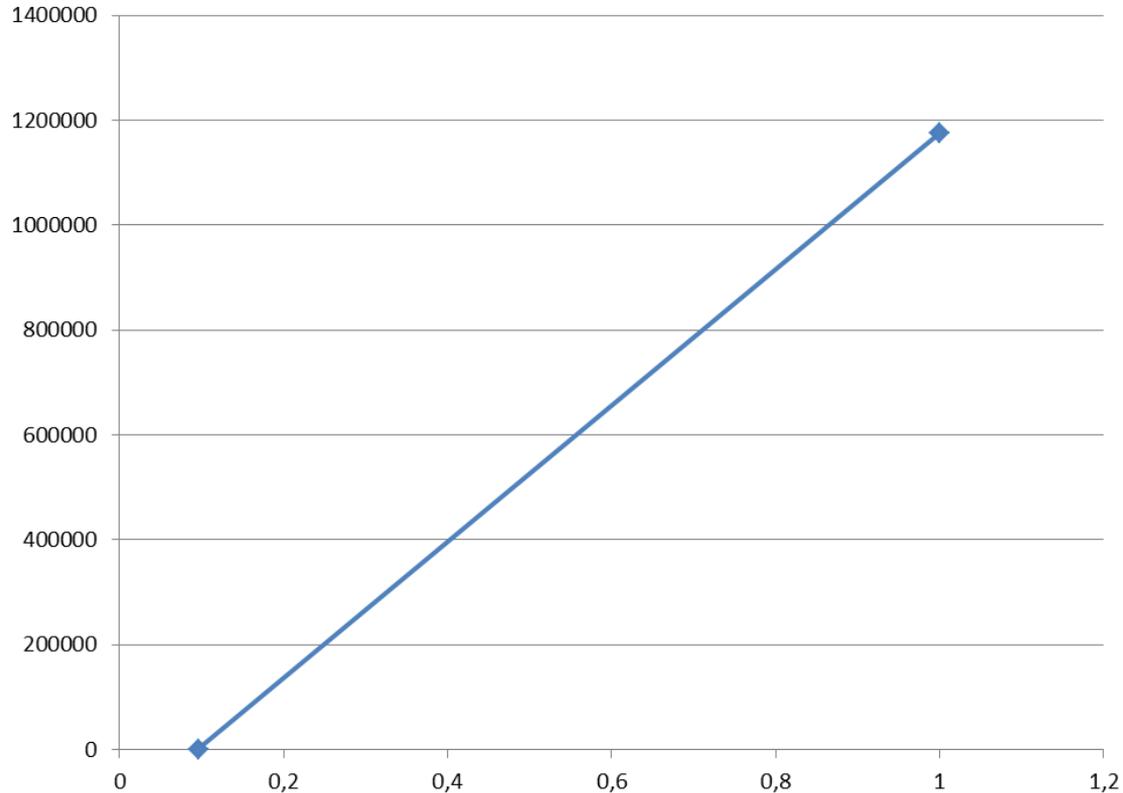
Й	о	ш	к	а	р	-	О	л	а
64	69	73	65	5C	6B	01	69	66	5C
01	00	00	00	00	00	00	00	00	00
01	00	00	00	00	00	00	01	00	00

Насколько всё это быстро?

JMH-замер:

- 10000 ops/sec — для 200 символов
- 30000 ops/sec — для 50 символов

Первоначальное заполнение интерполяционной таблицы



А также не упомянуты...

- Обработка и сортировка NULL-значений
- Режим прокрутки на малый шаг
- Анализ «качества» интерполяционных точек

...и другие мелкие, но важные детали

“God is in the detail.”

Выводы

1. Создавать
фреймворк — дело
трудное
и неблагодарное



Выводы

2. Всегда помните
про O-оценку
сложности алгоритмов



Выводы

3. По возможности,
избегайте COUNT
и OFFSET



Выводы

4. SQL не всемогущ. Ему можно и нужно «помогать»
5. Помните, что сортировка строк существенно зависит от Collation. Следите за текущим Collation.
6. **«Приблизительное» решение бывает намного быстрее «точного» и практически столь же хорошо**

Ссылки

- **Демо-пример:**
<https://github.com/inponomarev/lyragrid-demo>
- **Платформа C-Orchestra:** <http://corchestra.ru>
«Реализация грида»
 - <https://habrahabr.ru/post/278773/>
 - <https://habrahabr.ru/post/279083/>
- **«Быстро/неудобно — медленно/удобно»**
 - <https://habrahabr.ru/post/280157/>



Вопросы?

ponomarev@corchestra.ru
[@inponomarev](#)