Статистика в ретро-спективе

Егор Рогов
Postgres Pro

postgrespro.ru

О себе

Егор Рогов

Отдел образовательных программ Postgres Professional

- учебные курсы
- перевод документации и глоссарий
- демонстрационная база данных
- книги

edu@postgrespro.ru
http://postgrespro.ru/education

Зачем нужна статистика

Перебор планов

Оценка стоимости плана

Оценка кардинальности и селективности

Информация о стастических характеристиках данных

Что необходимо

От статистики

- актуальность
- **баланс объема и точности**

От планировщика

- учет имеющейся статистики для оценок

От разработчика

- аккуратные запросы

Статистика по таблице в целом

Количество строк и количество страниц

- pg_class: relpages, reltuples

Обновляется при удобном случае

- не только ANALYZE, но и VACUUM, CREATE INDEX

Оценка размера полной выборки из таблицы

Статистика по атрибутам

Нужно уметь оценивать селективность простых предикатов...

- атрибут = константа (!=)
- атрибут > константа (≥, <, ≤)
- и других

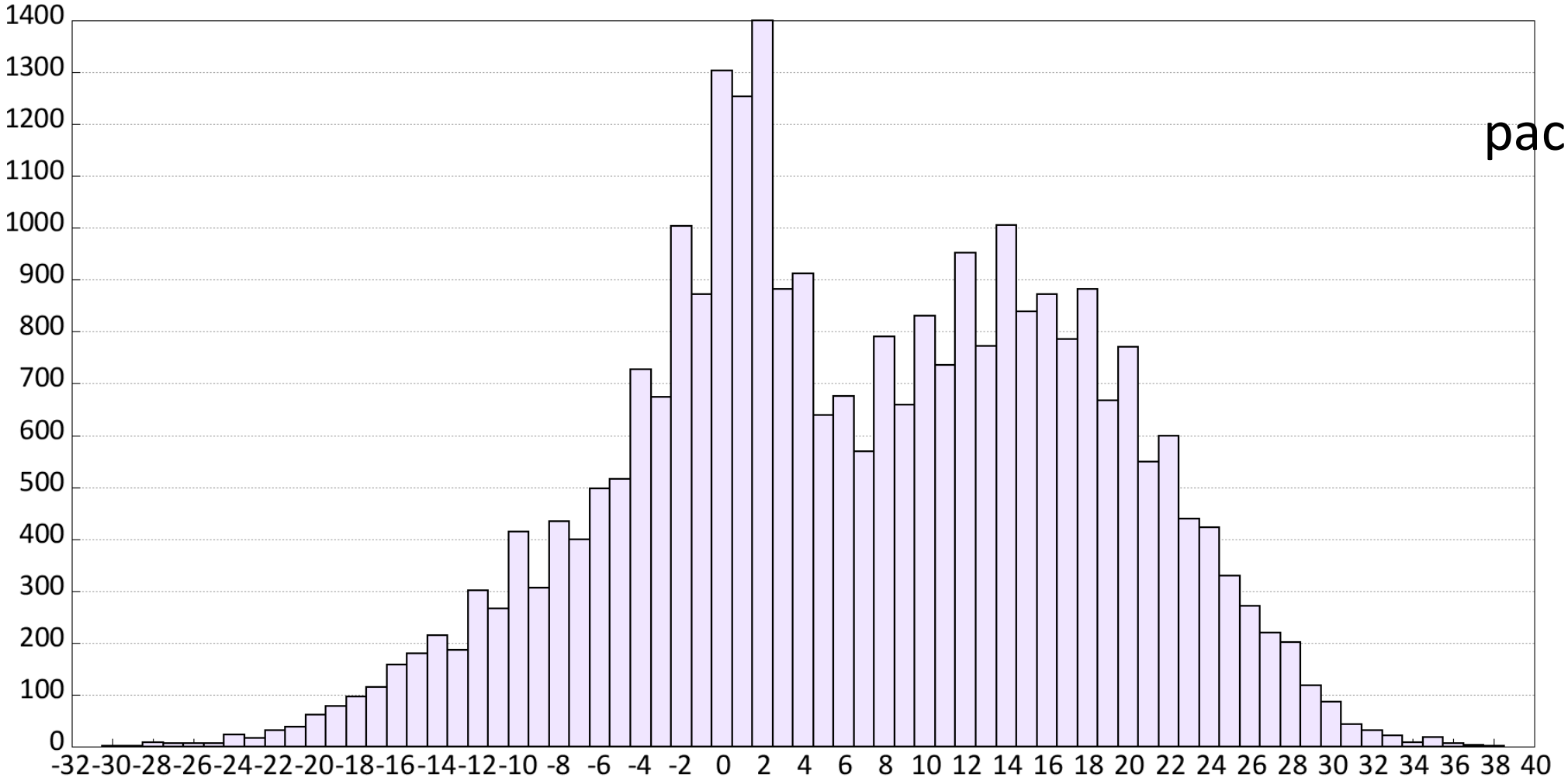
...а селективность сложных можно вычислить

- $sel_{\text{NOT } a} = 1 - sel_a$
- $sel_{a \text{ AND } b} = sel_a sel_b$ —— предположение о независимости *a* и *b*
- $sel_{a \text{ OR } b} = 1 - (1 - sel_a)(1 - sel_b)$

Пример: погода в Москве

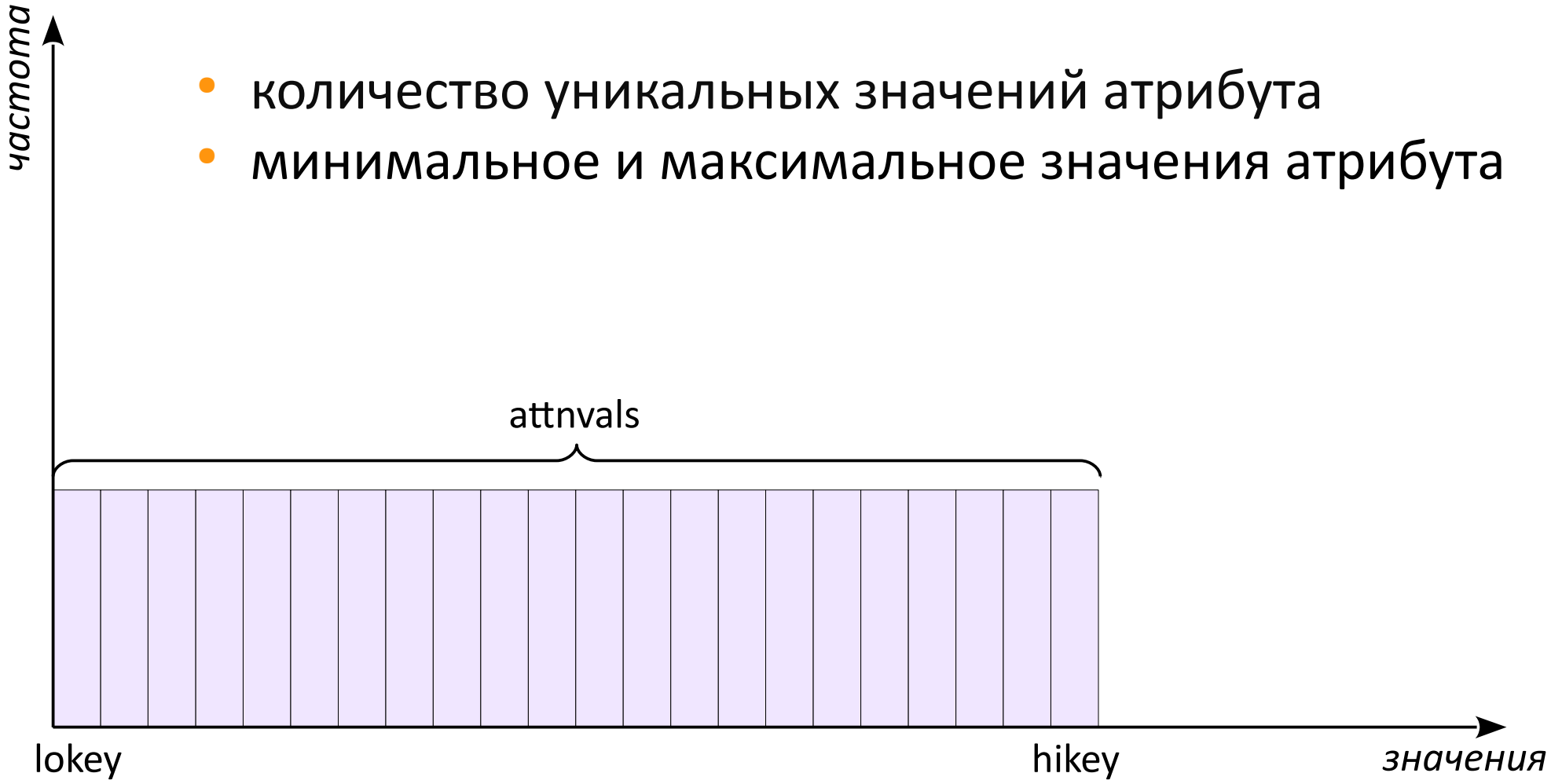
Среднесуточная температура за 10 лет

Пример: погода в Москве



Частотное распределение температур

Postgres95, 1996

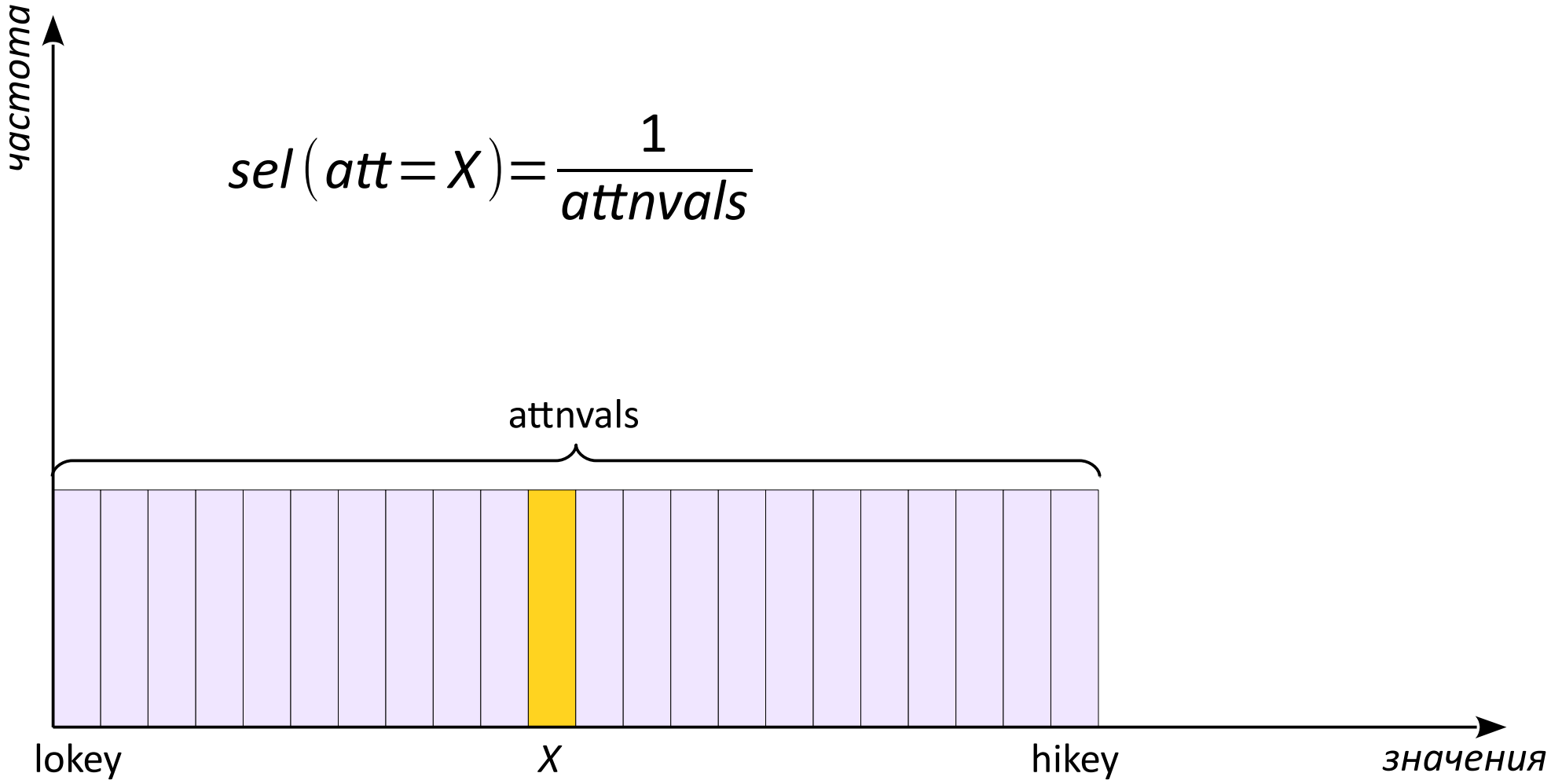


- количество уникальных значений атрибута
- минимальное и максимальное значения атрибута

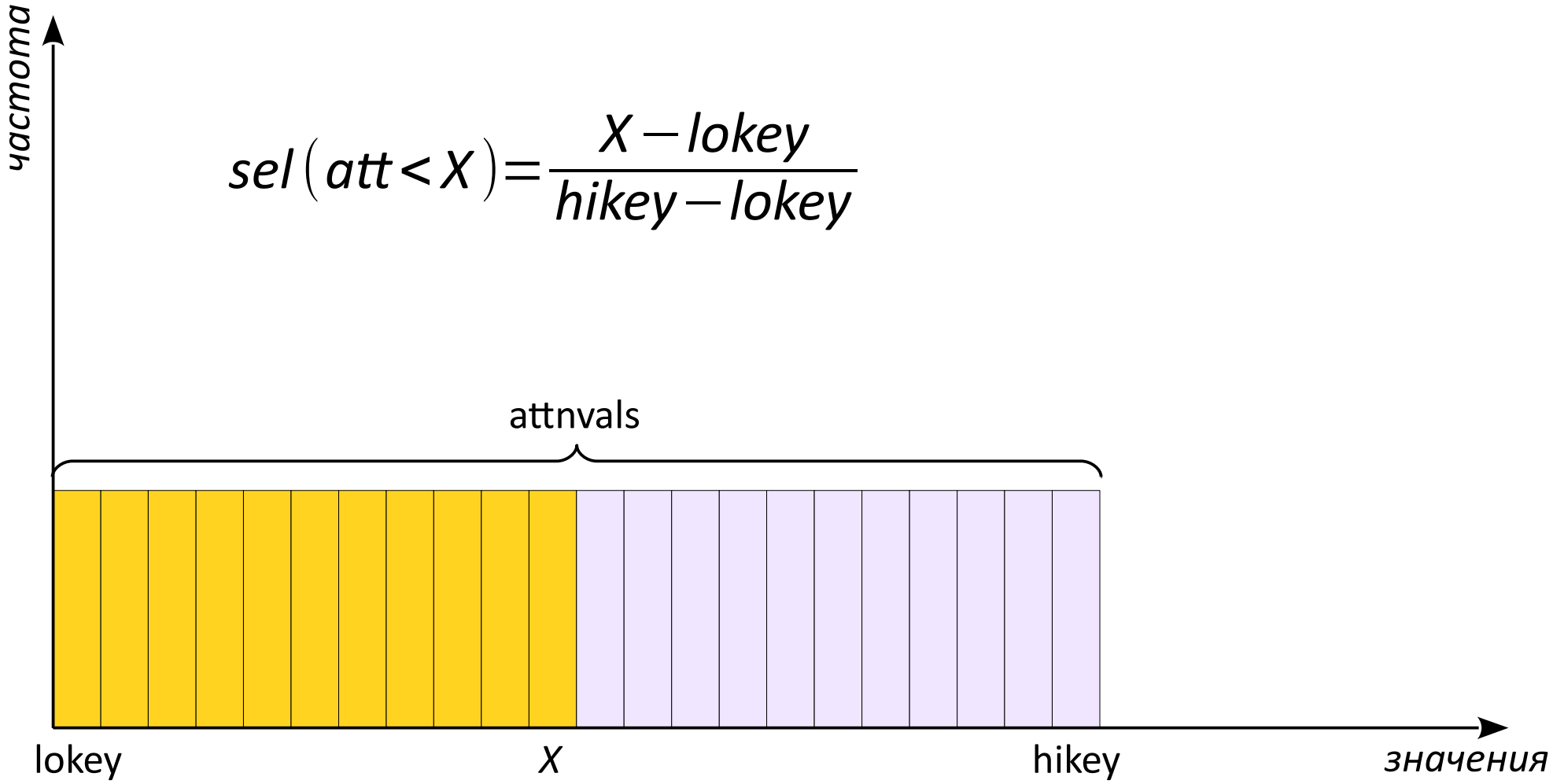
95 • 1996
6.2 • 1997
6.4 • 1998
6.5 • 1999
7.0 • 2000
7.1 • 2001
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Postgres95, 1996

$$sel(att = X) = \frac{1}{attnvals}$$



Postgres95, 1996



$$sel(att < X) = \frac{X - lokey}{hikey - lokey}$$

95 • 1996
6.2 • 1997
6.4 • 1998
6.5 • 1999
7.0 • 2000
7.1 • 2001
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Статистика

updateStats.tcl:
«this is kind of brute force and slow, but it works»

```
-- статистика заполняется вручную скриптами

95=> SELECT s.stalokey, s.stahikey
FROM pg_statistic s, pg_class c
WHERE s.starelid = c.oid AND c.relname = 'weather' AND s.staattnum = 4;

stalokey|stahikey
--------+--------
     -30|      38

95=> SELECT a.attnvals
FROM pg_attribute a, pg_class c WHERE a.attrelid = c.oid
AND c.relname = 'weather' AND a.attaattnum = 4;

attnvals
--------
      69
```

еще нет ни подзапросов, ни JOIN

95 • 1996
6.2 • 1997
6.4 • 1998
6.5 • 1999
7.0 • 2000
7.1 • 2001
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

И что получилось?..

```
95=> EXPLAIN SELECT * FROM weather WHERE temp = 0;

NOTICE:QUERY PLAN:
Index Scan on weather
```

кардинальность не отображается в плане
по формуле: 423, правильное значение: 1304

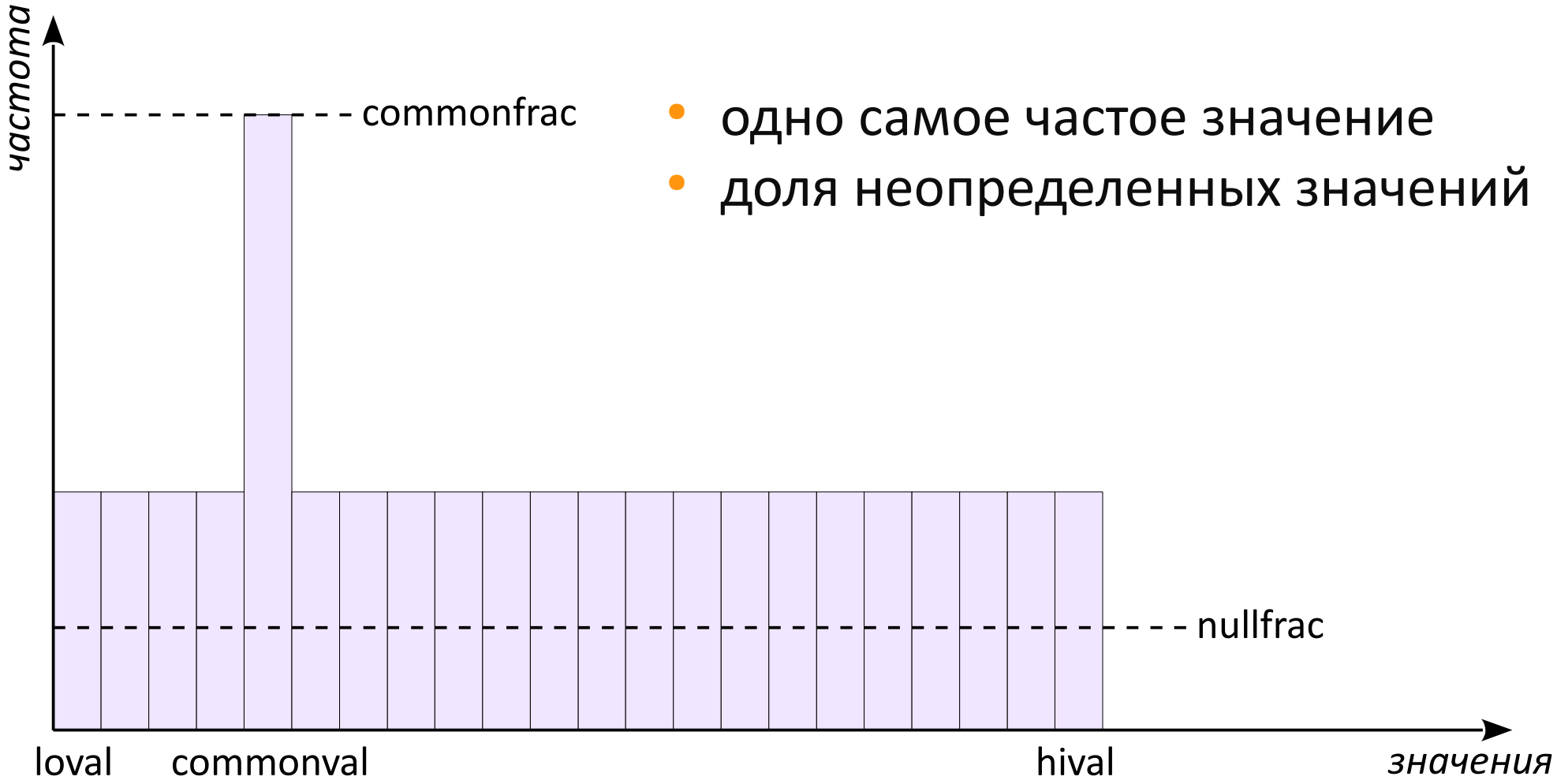
```
95=> EXPLAIN SELECT * FROM weather WHERE temp < 25;

NOTICE:QUERY PLAN:
Index Scan on weather
```

по формуле: 23631, правильное значение: 27855
реально — 9739 (треть строк), ошибку заметили в 7.0

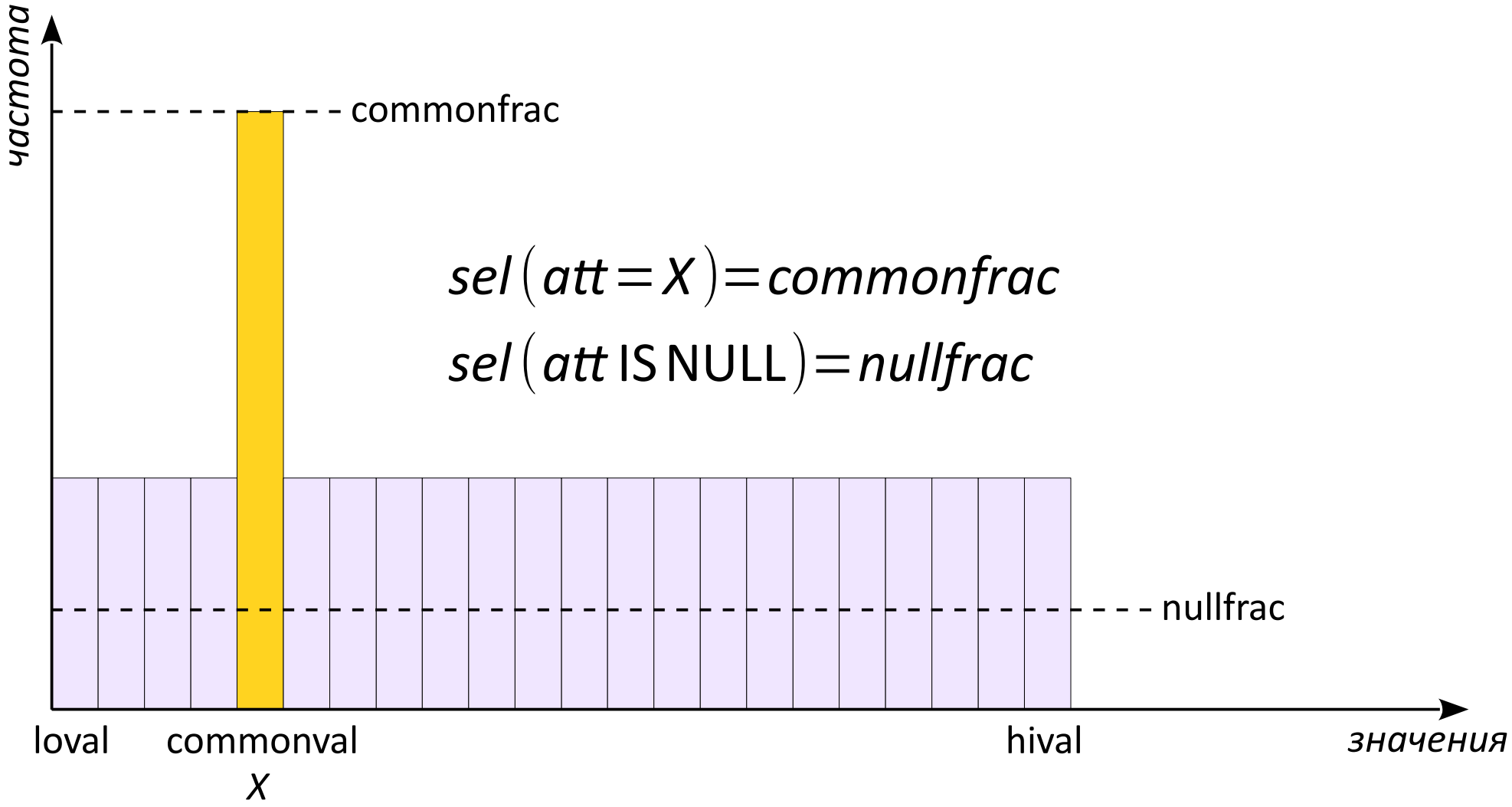
95 • 1996
6.2 • 1997
6.4 • 1998
6.5 • 1999
7.0 • 2000
7.1 • 2001
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 7.0, 2000



- одно самое частое значение
- доля неопределенных значений

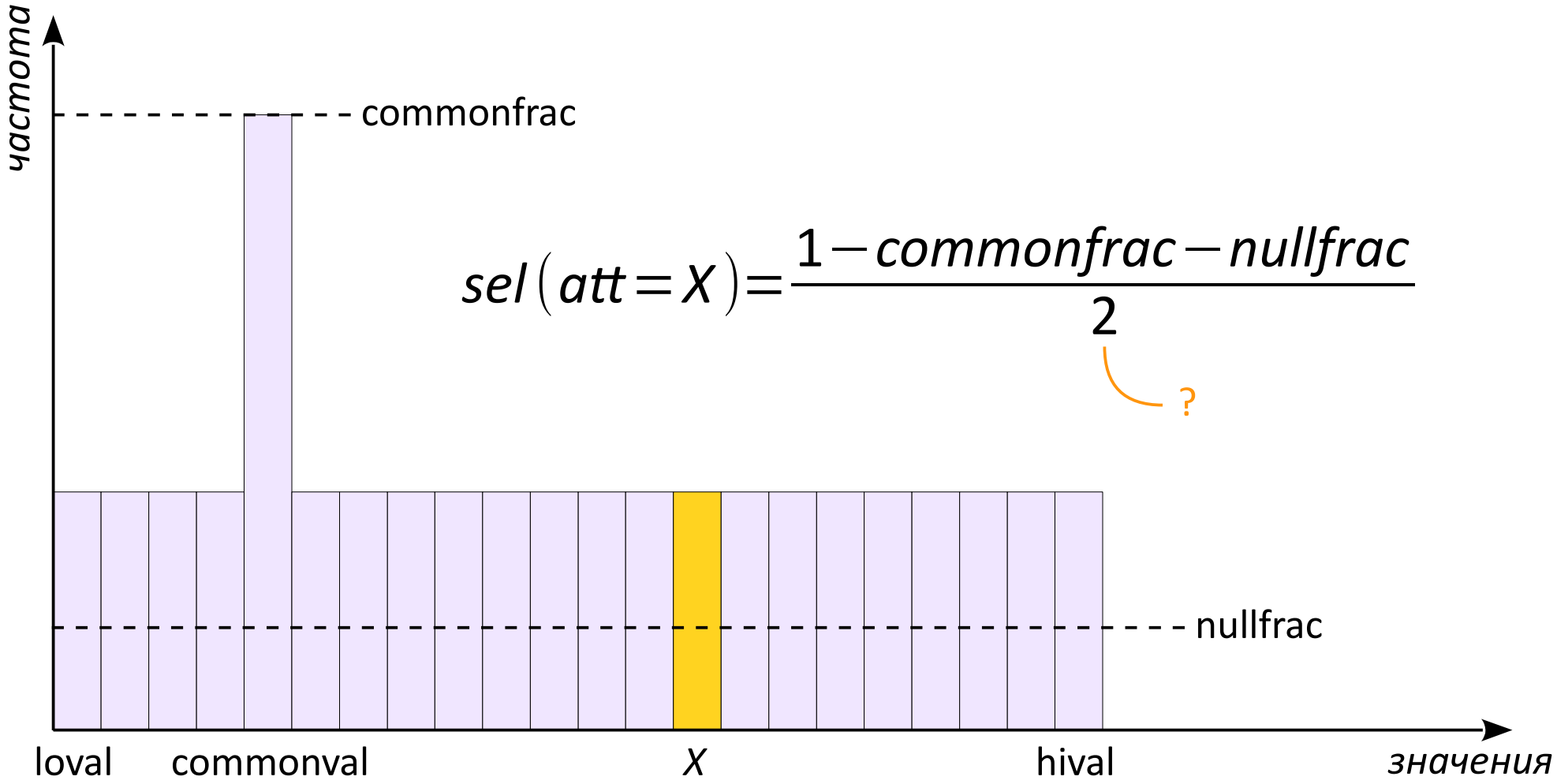
PostgreSQL 7.0, 2000



95 • 1996

7.0 • 2000
7.1 • 2001
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 7.0, 2000



Частое значение, доля NULL-ов

просто ANALYZE еще нет

```
7.0=# VACUUM ANALYZE weather;

7.0=# SELECT stacommonval, stacommonfrac, staloval, stahival,
stanullfrac
FROM pg_statistic
WHERE starelid = (SELECT oid FROM pg_class WHERE relname = 'weather')
AND staattnum = 4;
```

REGCLASS еще нет

```
 stacommonval | stacommonfrac | staloval | stahival | stanullfrac
--------------+---------------+----------+----------+-------------
 -12          |     0.0103026 | -30      | 38       |           0
```

правильное значение: 2

*«We use a three-bucket cache to get the most frequent item. This method works perfectly for columns with unique values, and columns with only two unique values, plus nulls. It becomes **less perfect** as the number of unique values increases and their distribution in the table becomes more random»*

95 • 1996
7.0 • 2000
7.1 • 2001
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Более корректный прогноз

```
7.0=# EXPLAIN SELECT * FROM weather WHERE temp = -12;

NOTICE:  QUERY PLAN:
Index Scan on weather  (... rows=301 ...)

7.0=# EXPLAIN SELECT * FROM weather WHERE temp = 0;

NOTICE:  QUERY PLAN:
Index Scan on weather  (... rows=30 ...)
```

правильное значение: 1304

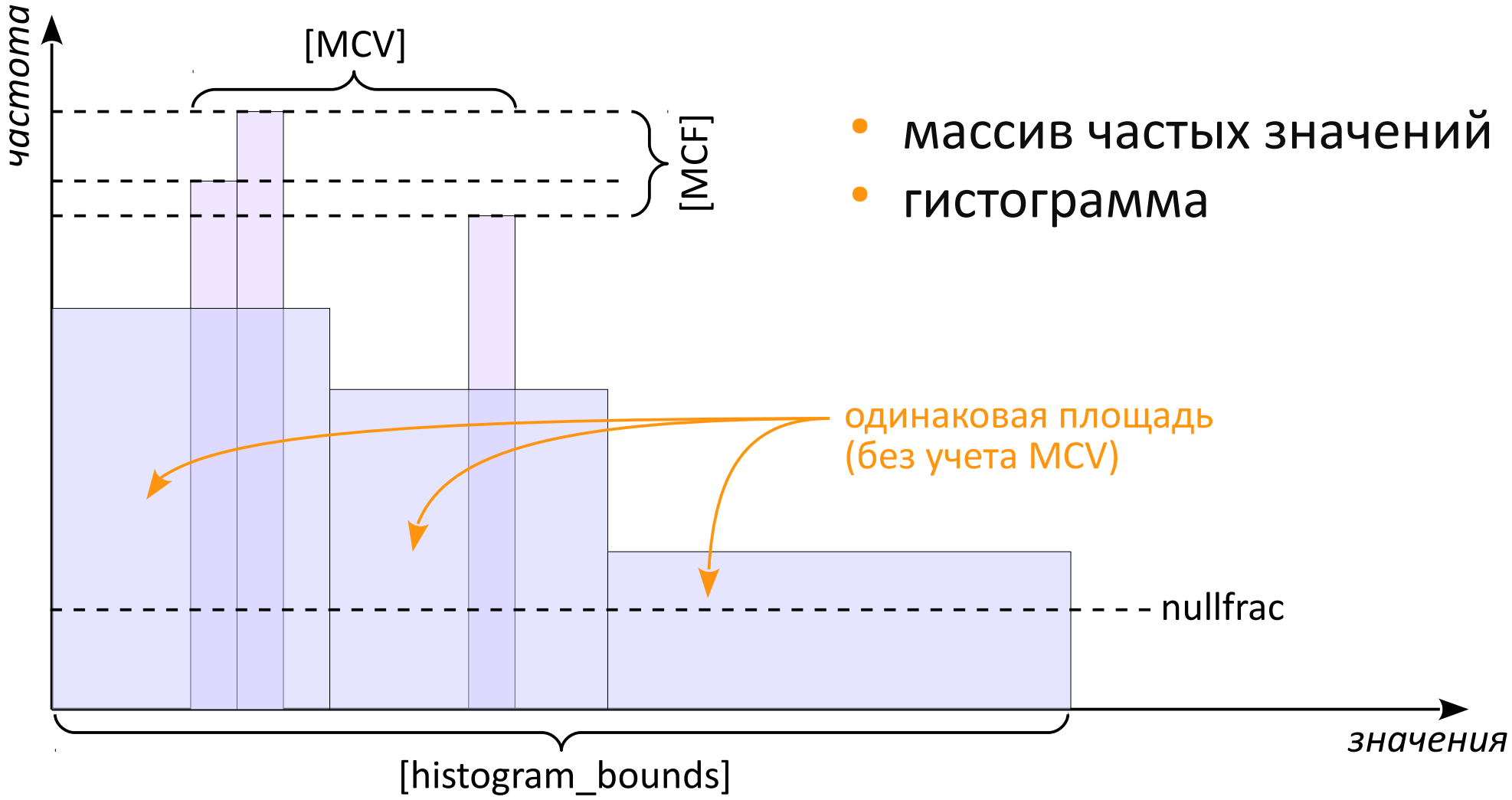
```
7.0=# EXPLAIN SELECT * FROM weather WHERE temp < 25;

NOTICE:  QUERY PLAN:
Seq Scan on weather  (... rows=23631 ...)
```

точное значение: 27855

95 • 1996
7.0 • 2000
7.1 • 2001
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

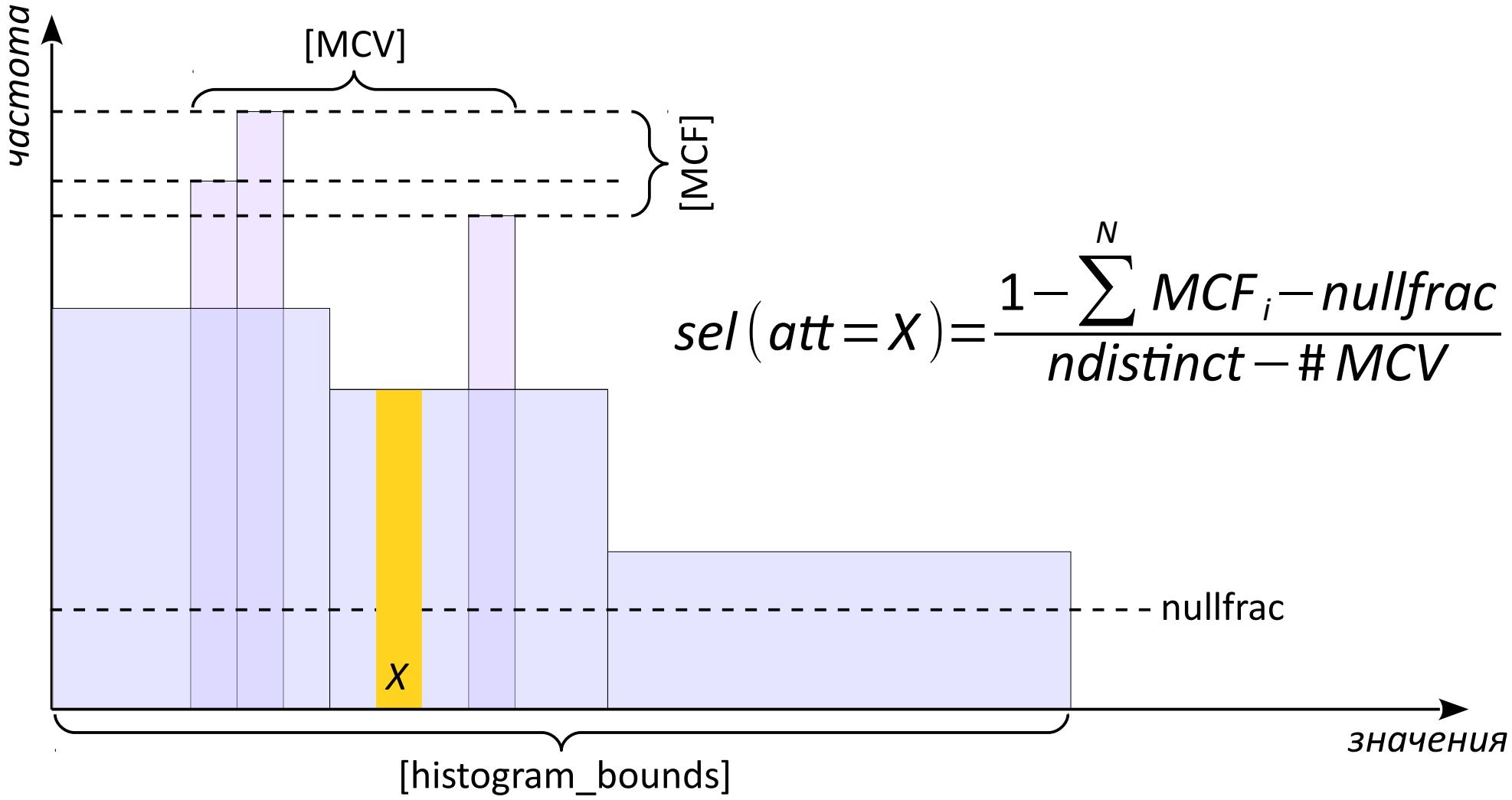
PostgreSQL 7.2, 2002



- массив частых значений
- гистограмма

95 • 1996
7.0 • 2000
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

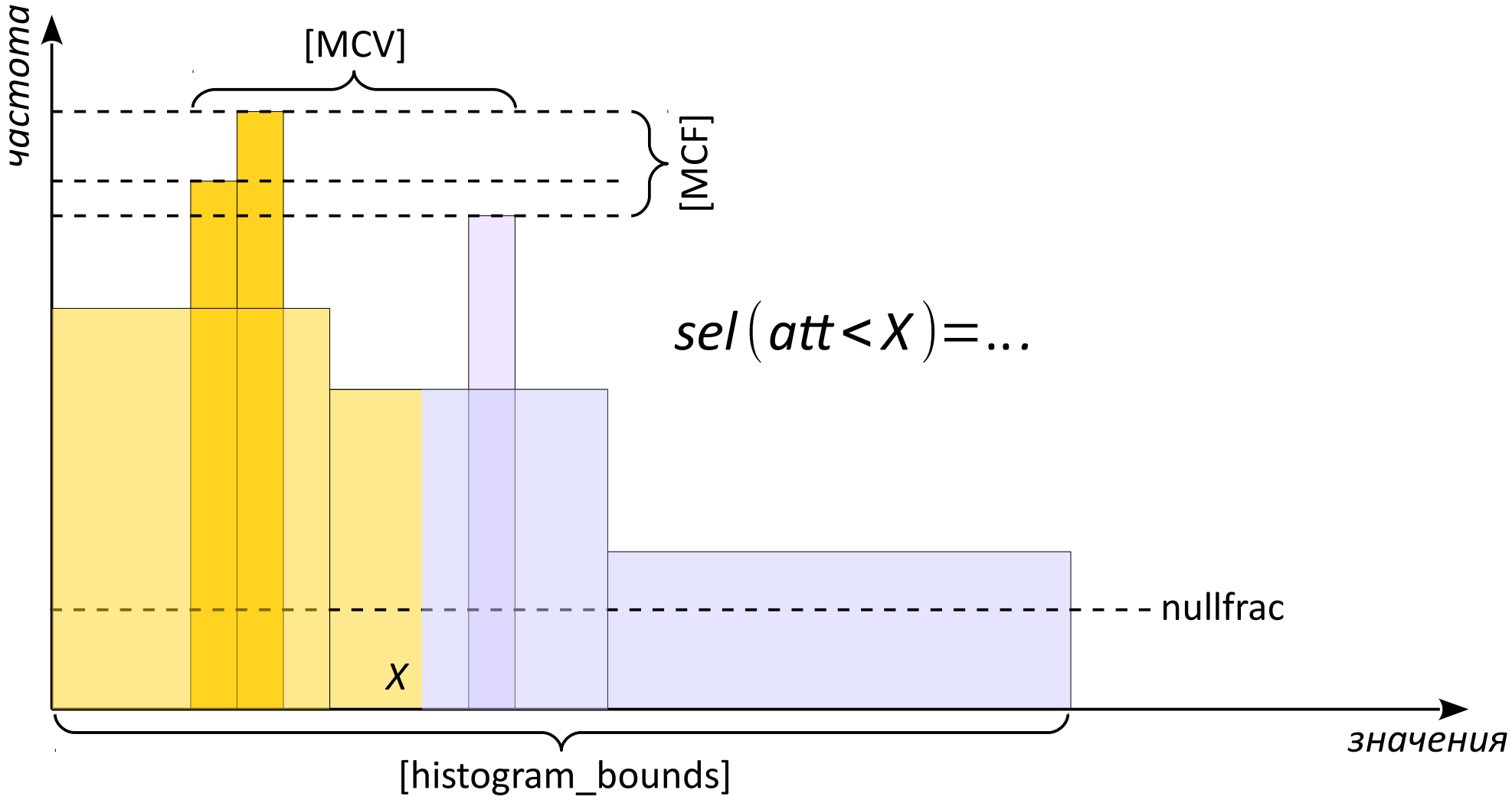
PostgreSQL 7.2, 2002



$$sel(att = X) = \frac{1 - \sum_{N} MCF_i - nullfrac}{ndistinct - \#MCV}$$

95 • 1996
7.0 • 2000
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 7.2, 2002



95 • 1996
7.0 • 2000
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Частые значения и гистограмма

теперь отдельно от VACUUM

```
7.2=# ANALYZE weather;

7.2=# SELECT most_common_vals, most_common_freqs
FROM pg_stats
WHERE tablename = 'weather' AND attname = 'temp';

-[ RECORD 1 ]-----+-----------------------------------------
most_common_vals  | {2,1,0,14,4,-2,12,18,-1,16}
most_common_freqs | {0.0483333,0.0436667,0.0433333,0.039,0.0336667,...
```

сделали для удобства

```
7.2=# SELECT histogram_bounds FROM pg_stats
WHERE tablename = 'weather' AND attname = 'temp';

-[ RECORD 1 ]----+--------------------------------
histogram_bounds | {-28,-10,-6,-3,6,9,13,17,20,24,37}
```

95 • 1996
7.0 • 2000
7.2 • 2002
7.4 • 2003
• 2004
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Корректный прогноз

```
7.2=# EXPLAIN SELECT * FROM weather WHERE temp = 0;
NOTICE:  QUERY PLAN:
Seq Scan on weather  (... rows=1269 ...)

7.2=# EXPLAIN SELECT * FROM weather WHERE temp = 25;
NOTICE:  QUERY PLAN:
Seq Scan on weather  (... rows=351 ...)

7.2=# EXPLAIN SELECT * FROM weather WHERE temp < 25;
NOTICE:  QUERY PLAN:
Seq Scan on weather  (... rows=27571 ...)
```

PostgreSQL 8.0, 2005

Статистика для индексов по выражениям

Поддержка сбора статистики для произвольных типов

analyze.c:
«Call the type-specific typanalyze function.
If none is specified, use std_typanalyze()»

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Индекс по выражению

```
8.0=# EXPLAIN SELECT * FROM weather WHERE extract('month' FROM day)=1;
                QUERY PLAN
-----------------------------------------
 Seq Scan on weather  (... rows=147 ...)

8.0=# CREATE INDEX weather_mon ON weather(extract('month' FROM day));
8.0=# ANALYZE weather;

8.0=# EXPLAIN SELECT * FROM weather WHERE extract('month' FROM day)=1;
                    QUERY PLAN
-----------------------------------------------
 Seq Scan on weather  (... rows=2435 ...)
```

Bitmap Scan появится в 8.1

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.2 • 2006
• 2007
8.3 • 2008
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 8.4, 2009

Статистика для полнотекстового поиска
(встроен в ядро в 8.3, до того был модулем contrib/tsearch2)

- наиболее частые элементы tsvector

Применение

- селективность оператора @@
- способ выполнения запроса '*частое* & *редкое*'

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Частые лексемы

```
8.4=# SELECT stavalues1, stanumbers1
FROM pg_statistic ——— в 8.4 еще не добавили в pg_stats
WHERE starelid = 'mail_messages'::regclass AND staattnum = 7
AND stakind1 = 4;

-[ RECORD 1 ]----------------------------------------------
stavalues1  | {like,make,need,one,think,tom,use,would,write,wrote}
stanumbers1 | {0.39,0.33,0.34,0.35,0.42,0.34,0.49,0.45,0.35,0.66}
```

Корректный прогноз

```
8.4=# EXPLAIN SELECT * FROM mail_messages
WHERE tsv @@ to_tsquery('wrote');
                        QUERY PLAN
-----------------------------------------------------
 Seq Scan on mail_messages  (... rows=231636 ...)
```

361 в 8.3

```
8.4=# EXPLAIN SELECT * FROM mail_messages
WHERE tsv @@ to_tsquery('wrote & tattoo');
                        QUERY PLAN
-----------------------------------------------------
 Bitmap Heap Scan on mail_messages  (... rows=1158 ...)
```

точное значение: 1
361 в 8.3

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.0 • 2010
9.1 • 2011
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 9.2, 2012

Статистика для массивов

- наиболее частые элементы массивов
- гистограмма длин массивов

Применение

- селективность операторов <@, @> (вхождение) и && (пересечение)

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

Корректный прогноз

```
9.2=# EXPLAIN SELECT * FROM weather_daily WHERE temps && ARRAY[0];
                              QUERY PLAN
-----------------------------------------------------------------
 Bitmap Heap Scan on weather_daily  (... rows=608 ...)
```

18 в 8.4

```
9.2=# EXPLAIN SELECT * FROM weather_daily WHERE temps && ARRAY[25];
                              QUERY PLAN
-----------------------------------------------------------------
 Bitmap Heap Scan on weather_daily  (... rows=264 ...)
```

18 в 8.4

GIN-индекс
(или GiST с расширением intarray)

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 9.3, 2013

Статистика для диапазонного типа

- гистограмма границ диапазонов
- гистограмма длин диапазонов

Применение

- селективность операторов неравенства, «слева», «справа», «включение элемента», «включение диапазона»

Корректный прогноз

```
9.3=# EXPLAIN SELECT * FROM weather_daily
WHERE temp @> 0;

                      QUERY PLAN
-------------------------------------------------
 Seq Scan on weather_daily  (... rows=694 ...)
```

точное значение: 710, 4 в 9.2

```
9.3=# EXPLAIN SELECT * FROM weather_daily
WHERE temp <@ '[20,30]'::int4range;

                      QUERY PLAN
-------------------------------------------------
 Seq Scan on weather_daily  (... rows=50 ...)
```

точное значение: 43, 1217 в 9.2

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
9.4 • 2014
• 2015
9.5 • 2016
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 10, 2017

Расширенная (многоколоночная) статистика

- функциональная зависимость — проблема коррелированных предикатов
- количество уникальных значений — оценка кардинальности при группировании

Ошибки прогноза

```
10=# EXPLAIN SELECT * FROM weather
WHERE day = '2019-01-01' AND month = 1;
                QUERY PLAN
-----------------------------------------
 Seq Scan on weather  (... rows=1 ...)

10=# EXPLAIN SELECT day, temp, count(*)
FROM weather GROUP BY day, temp;
                QUERY PLAN
-------------------------------------
 HashAggregate  (... rows=3652 ...)
```

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
10 • 2017
11 • 2018
12 • 2019

Dependencies, ndistinct

```
10=# CREATE STATISTICS weather_day_month(dependencies)
     ON day, month FROM weather;
10=# CREATE STATISTICS weather_day_temp(ndistinct)
     ON day, temp FROM weather;

10=# ANALYZE weather;

10=# SELECT stxkind, stxndistinct, stxdependencies
FROM pg_statistic_ext ——— новая таблица
WHERE stxrelid = 'weather'::regclass;
```

stxkind	stxdependencies	stxndistinct
{f}	{"1 => 3": 1.000000}	
{d}		{"1, 4": 18465}

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
10 • 2017
11 • 2018
12 • 2019

Корректный прогноз

```
10=# EXPLAIN SELECT * FROM weather
WHERE day = '2019-01-01' AND month = 1;
                QUERY PLAN
-----------------------------------------
 Seq Scan on weather  (... rows=8 ...)

10=# EXPLAIN SELECT day, temp, count(*)
FROM weather GROUP BY day, temp;
                QUERY PLAN
-------------------------------------
 HashAggregate  (... rows=18465 ...)
```

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
10 • 2017
11 • 2018
12 • 2019

PostgreSQL 12, 2019

Наиболее частые комбинации значений
для многоколоночной статистики

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
10 • 2017
12 • 2019

Ошибка прогноза

```
12=# EXPLAIN SELECT * FROM weather WHERE month = 7 AND temp = 14;
                          QUERY PLAN
-----------------------------------------------------------
 Seq Scan on weather  (... rows=85 ...)

12=# EXPLAIN SELECT * FROM weather WHERE month = 7 AND temp = 0;
                          QUERY PLAN
-----------------------------------------------------------
 Seq Scan on weather  (... rows=111 ...)
```

условие не учитывается

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
10 • 2017
12 • 2019

MCV

```
12=# CREATE STATISTICS weather_stx(mcv) ON month, temp FROM weather;
12=# ANALYZE weather;

12=# SELECT m.*
FROM pg_statistic_ext
JOIN pg_statistic_ext_data ON oid = stxoid,
     pg_mcv_list_items(stxdmcv) m
WHERE stxname = 'weather_stx';
```

данные выделены в отдельную таблицу

если были бы независимы

index	values	nulls	frequency	base_frequency
0	{9,12}	{f,f}	0.0105079408543263	0.0026795547219246 3
1	{12,0}	{f,f}	0.00989184008762 3	0.00378867856170 2
2	{11,2}	{f,f}	0.00876232201533 40	0.00393638678981 5
3	{12,1}	{f,f}	0.00862541073384 4	0.00364050171611468
4	{3,0}	{f,f}	0.00842004381161 00	0.00378867856170 2

Корректный прогноз

```
12=# EXPLAIN SELECT * FROM weather WHERE month = 7 AND temp = 14;
                          QUERY PLAN
-----------------------------------------------------------
 Seq Scan on weather  (... rows=132 ...)

12=# EXPLAIN SELECT * FROM weather WHERE month = 7 AND temp = 0;
                          QUERY PLAN
-----------------------------------------------------------
 Seq Scan on weather  (... rows=1 ...)
```

95 • 1996
7.0 • 2000
7.2 • 2002
8.0 • 2005
8.4 • 2009
9.2 • 2012
9.3 • 2013
10 • 2017
12 • 2019

Что и зачем настраивать

Одна общая ручка: statistics target

- размер случайной выборки для анализа (x 300)
- размер списков наиболее частых значений и элементов
- размер гистограмм

Зачем крутить

- увеличить выборку (неточный или нестабильный результат)
- увеличить точность прогноза

Абсолютно точная статистика никогда требуется

Что и зачем настраивать

Расширенная статистика

- CREATE STATISTICS ...

Создается вручную, поддерживается автоматически

Зачем создавать

- исправление ошибок прогноза из-за коррелированных данных

Как настраивать

Глобально

- ALTER SYSTEM SET *default_statistics_target = ...*

На уровне сеанса (для экспериментов)

- SET *default_statistics_target = ...*

На уровне столбца или столбцов

- ALTER TABLE … ALTER COLUMN … SET STATISTICS …
- ALTER INDEX … ALTER COLUMN … SET STATISTICS …
- ALTER STATISTICS … SET STATISTICS …

уже в PostgreSQL 13

Спасибо за внимание

Хорошая мысля приходит опосля?

- edu@postgrespro.ru

Postgres Professional

http://postgrespro.ru/

+7 495 150 06 91

info@postgrespro.ru

postgrespro.ru