

Оптимизация запросов Статистика



Авторские права

© Postgres Professional, 2019 год.

Авторы: Егор Рогов, Павел Лузанов

Использование материалов курса

Некоммерческое использование материалов курса (презентации, демонстрации) разрешается без ограничений. Коммерческое использование возможно только с письменного разрешения компании Postgres Professional. Запрещается внесение изменений в материалы курса.

Обратная связь

Отзывы, замечания и предложения направляйте по адресу:

edu@postgrespro.ru

Отказ от ответственности

Компания Postgres Professional не несет никакой ответственности за любые повреждения и убытки, включая потерю дохода, нанесенные прямым или непрямым, специальным или случайным использованием материалов курса. Компания Postgres Professional не предоставляет каких-либо гарантий на материалы курса. Материалы курса предоставляются на основе принципа «как есть» и компания Postgres Professional не обязана предоставлять сопровождение, поддержку, обновления, расширения и изменения.

Базовая статистика

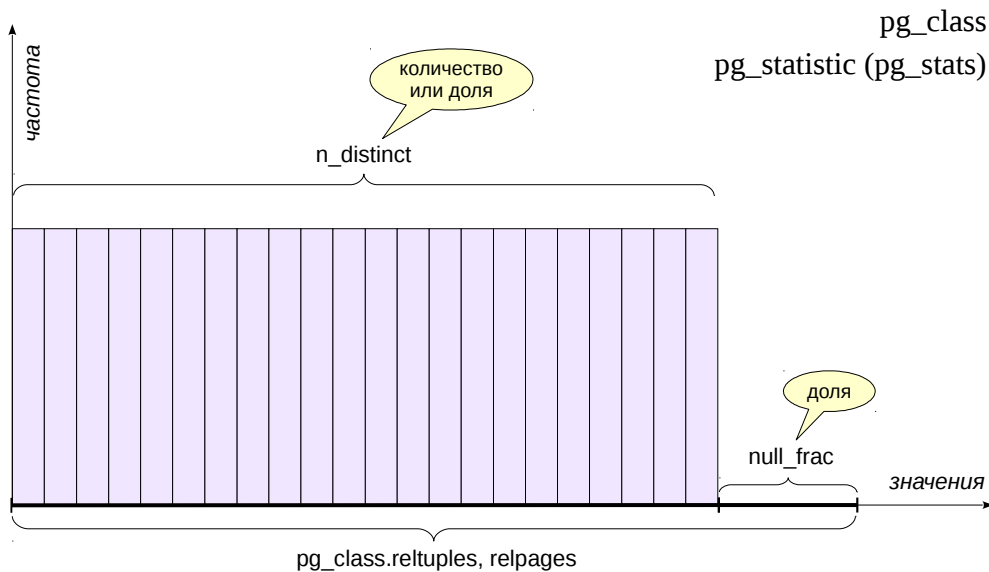
Наиболее частые значения и гистограммы

Расширенная статистика

Статистика для индексов по выражениям

Сбор статистики

Использование статистики для оценки кардинальности
и селективности



Теперь перейдем к вопросу статистики и ее использования для оценки кардинальности и селективности. Какая статистика собирается?

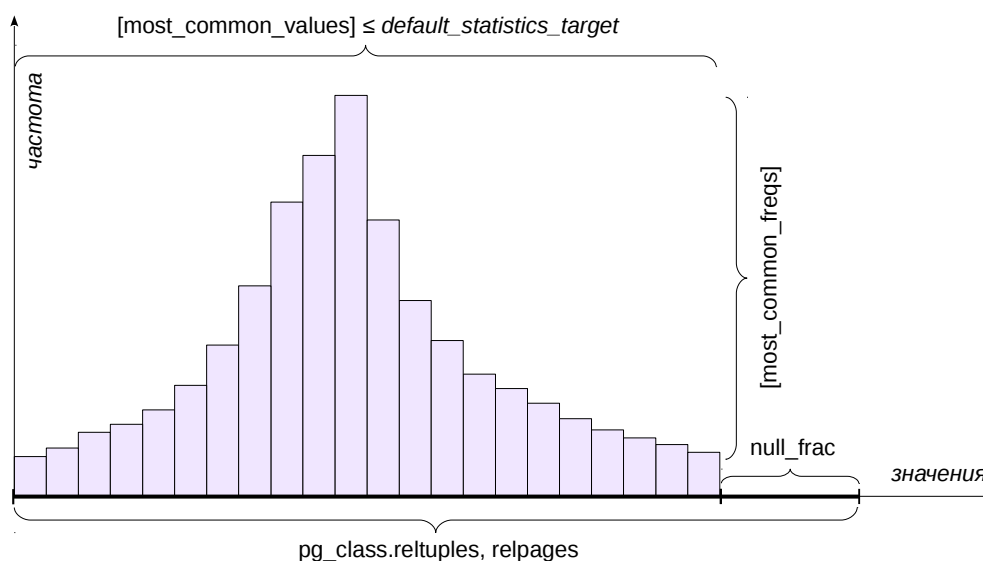
Во-первых, это число строк в таблице (или индексе) и размер объекта в страницах. Эта информация хранится в системном каталоге в таблице `pg_class`, поля `reltuples` и `relpages`.

Вся остальная статистика собирается отдельно для каждого столбца и хранится в таблице `pg_statistic`. Но смотреть проще в представление `pg_stats`, которое показывает информацию в более удобном виде.

Поле `null_frac` содержит долю null-значений в таблице (от 0 до 1).

Поле `n_distinct` хранит число уникальных значений в столбце. Если значение `n_distinct` отрицательно, то модуль этого числа показывает долю уникальных значений. Например, `-1` означает, что все значения уникальны (типичный случай для первичного ключа).

<https://postgrespro.ru/docs/postgresql/10/planner-stats#id-1.5.13.5.3>



Если бы все данные были всегда распределены равномерно, то есть все значения встречались бы с одинаковой частотой, этой информации было бы почти достаточно (нужен еще минимум и максимум).

Но в реальности неравномерные распределения встречаются очень часто. Поэтому собирается еще и следующая информация:

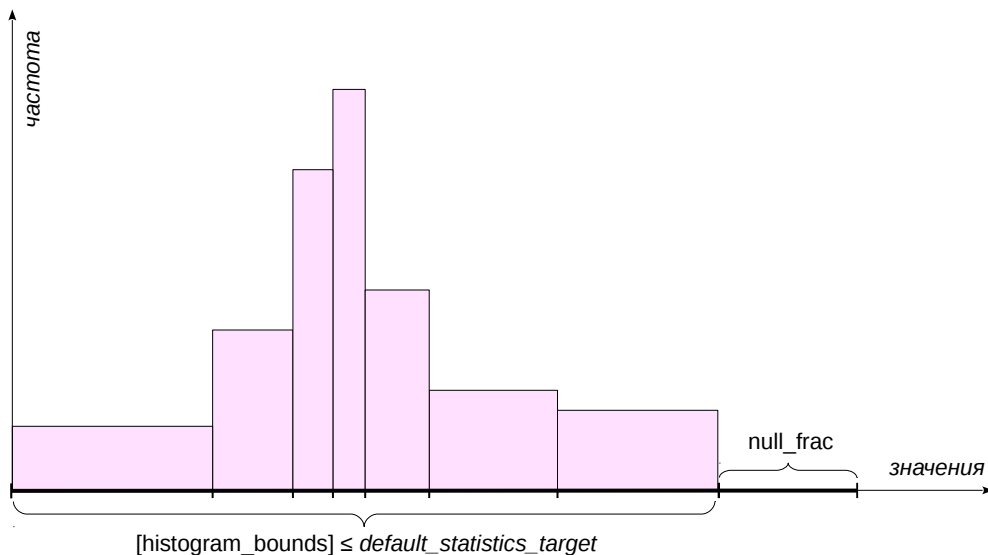
- массив наиболее частых значений — поле `most_common_values`,
- массив частот этих значений — поле `most_common_freqs`.

Если нам надо оценить селективность условия *поле = значение*, мы находим *значение* в массиве `most_common_values` и берем частоту из `most_common_freqs` — это и будет селективность.

Если надо оценить селективность условия *поле < значение*, находим в `most_common_values` все значения, меньшие искомого *значения*, и суммируем частоты из `most_common_freqs`.

Все это прекрасно работает, пока число различных значений не очень велико. Максимальный размер задается параметром `default_statistics_target`.

Тонкий момент представляют «большие» значения. Чтобы не увеличивать размер `pg_statistic` и не нагружать планировщик бесполезной работой, значение исключается из статистики и анализа, если его размер превышает 1 КБ. В самом деле, если в поле хранятся такие большие значения, скорее всего они уникальны и не имеют шансов попасть в `most_common_values`.



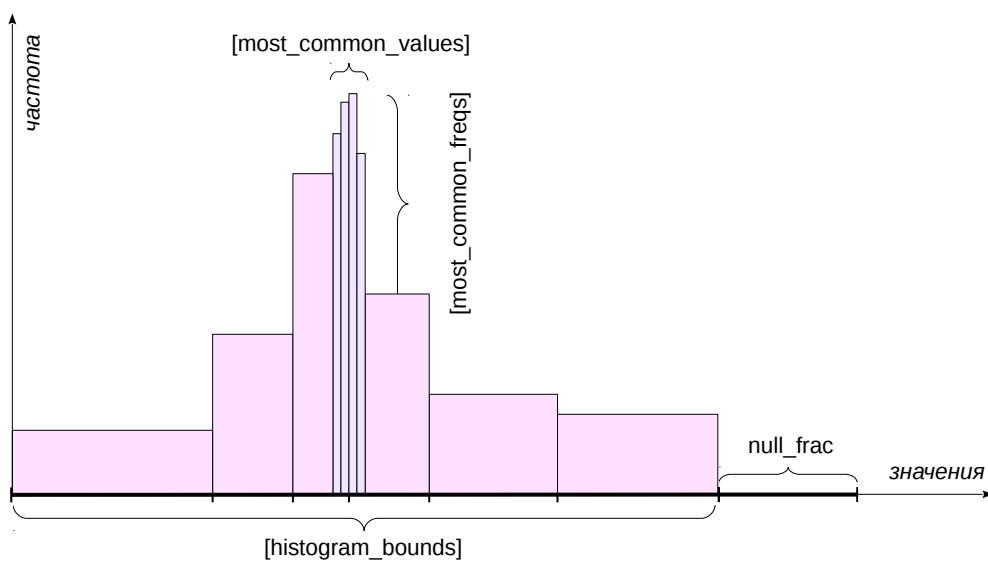
Если число различных значений слишком велико, чтобы записать их в массив, на помощь приходит гистограмма. Гистограмма состоит из нескольких корзин, в которые помещаются значения. Количество корзин ограничено тем же параметром *default_statistics_target*.

Ширина корзин выбирается так, чтобы в каждую попало примерно одинаковое число значений (на рисунке это выражается в одинаковой площади прямоугольников).

При таком построении достаточно хранить только массив крайних значений каждой корзины — поле *histogram_bounds*. Частота одной корзины равна $1/(\text{число корзин})$.

Оценить селективность условия *поле < значение* можно как $N/(\text{общее число корзин})$, где N — число корзин, лежащих слева от *значения*. Оценку можно улучшить, добавив часть корзины, в которую попадает само *значение*.

Если же надо оценить селективность условия *поле = значение*, то гистограмма в этом не может помочь, и приходится довольствоваться предположением о равномерном распределении и брать в качестве оценки $1/n_distinct$.



Но обычно два подхода объединяются: строится список наиболее частых значений, а все остальные значения покрываются гистограммой.

При этом гистограмма строится так, что в ней не учитываются значения, попавшие в список. Это позволяет улучшить оценки.

Упорядоченность (использовать битовую карту?)

`pg_stats.correlation`
(1 — по возрастанию, 0 — хаотично, -1 — по убыванию)

Видимость (использовать только индексное сканирование?)

`pg_class.relallvisible`

Средний размер значения в байтах (оценка памяти)

`pg_stats.avg_width`

Информация об элементах массивов, tsvector и т. п.

`pg_stats.most_common_elems`
`pg_stats.most_common_elem_freqs`
`pg_stats.elem_count_histogram`

Еще несколько значений статистики.

В поле `correlation` записывается показатель упорядоченности значений на диске. Если значения хранятся строго по возрастанию, значение будет близко к единице; если по убыванию — к минус единице. Чем более хаотично расположены данные на диске, чем ближе значение к нулю. Именно это поле использует оптимизатор, когда выбираем между сканированием битовой карты и обычным индексным сканированием.

Поле `pg_class.relallvisible` хранит количество страниц таблицы, которые содержат только актуальные версии строк (эта информация обновляется вместе с картой видимости). Если количество недостаточно велико, планировщик может отказаться от исключительно индексного сканирования в пользу сканирования по битовой карте.

В поле `avg_width` сохраняется средний размер строки в байтах для расчета необходимого для операции объема памяти.

В полях `most_common_elems`, `most_common_elem_freqs` и `elem_count_histogram` для таких составных типов, как массивы или `tsvector`, хранится распределение не самих значений, а их элементов. Это позволяет более точно планировать запросы с участием полей *не в первой нормальной форме*.

<https://postgrespro.ru/docs/postgresql/10/view-pg-stats>

Статистика значений в нескольких столбцах

`pg_statistic_ext`

Функциональные зависимости

зависимости одних столбцов от других

улучшает оценку условий с коррелированными предикатами

Уникальные значения

число уникальных значений в нескольких столбцах

улучшает оценку группировки

Начиная с версии 10 в PostgreSQL можно создать специальный объект для сбора расширенной статистики по нескольким столбцам таблицы. Собранные информация хранится в отдельной таблице системного каталога `pg_statistic_ext`.

В настоящее время таким образом можно собирать два вида статистики.

Во-первых, информация о функциональных зависимостях между столбцами. Допустим, данные в одном столбце определяются (полностью или частично) значением другого столбца. Если в запросе указать условия на оба столбца, оценка кардинальности окажется заниженной: планировщик решит, что оба условия уменьшают число строк в выборке. В таких случаях расширенная статистика может улучшить оценку.

Во-вторых, информация о числе уникальных значений в нескольких столбцах. Такая информация позволяет улучшить оценку кардинальности группировки по нескольким столбцам.

<https://postgrespro.ru/docs/postgresql/10/planner-stats#PLANNER-STATS-EXTENDED>

Размер таблицы (reltuples, relpages)

операции DDL

очистка

анализ

Остальная статистика

анализ

Настройка

`default_statistics_target = 100`

`ALTER TABLE ...`

`ALTER COLUMN ...`

`SET STATISTICS`

{ число значений в списке
число корзин в гистограмме
размер выборки при анализе

Информация о размере объекта (reltuples, relpages в pg_class) заполняется некоторыми DDL-операциями (CREATE INDEX, CREATE TABLE AS SELECT), а затем обновляется при очистке и анализе. Эти значения не будут точными, поскольку просматривается только часть страниц. Поэтому планировщик масштабирует значение в соответствии с отклонением реального размера файла данных от relpages.

Вся остальная статистика собирается при анализе таблицы. Обычно это происходит при автоанализе (рассматривается в курсе DBA2).

Вообще вся статистика не точна, да и не должна быть точной. Обычно достаточно попадания в порядок, чтобы выбрать приемлемый план. Настройка точности собираемой статистики всего одна — это параметр `default_statistics_target` (по умолчанию 100).

Он определяет и максимальный размер списка `most_common_values`, и максимальное число корзин в гистограмме. Значение этого параметра можно задавать индивидуально для каждого столбца. Уменьшение параметра (вплоть до 0) ускоряет работу планировщика, увеличение — повышает точность.

Этот же параметр определяет размер случайной выборки, которая используется при анализе. Установлено, что размер выборки, обеспечивающий хорошую точность оценок, практически не зависит от размера таблицы. Поэтому вычисляется максимум из значения глобального параметра и значений для каждого столбца (но не меньше 100), и анализируется в 300 раз больше случайных строк.

<https://postgrespro.ru/docs/postgresql/10/row-estimation-examples>



Характеристики данных собираются в виде статистики

Статистика нужна для оценки кардинальности

Кардинальность используется для оценки стоимости

Стоимость позволяет выбрать оптимальный план

Основа успеха —

адекватная статистика и корректная кардинальность

1. Создайте индекс на таблице билетов (tickets) по имени пассажира (passenger_name).
2. Какая статистика имеется для этой таблицы?
3. Объясните оценку кардинальности и выбор плана выполнения следующих запросов:
 - а) выборка всех билетов,
 - б) выборка билетов на имя ALEKSANDR IVANOV,
 - в) выборка билетов на имя ANNA VASILEVA,
 - г) выборка билета с идентификатором 0005432000284.