Вопрос

Определите рейтинг пользователей по количеству созданных операций в каждом из месяцев 2018 года. Выведите USER\_ID, дату в формате (MM.YYYY), кол-во созданных операций, рейтинг пользователя (1,2 и так далее - без пропусков, пользователи, создавшие одинаковое количество операций имеют одинаковый рейтинг).

select

USER\_ID,

rep\_date,

cnt,

rownum

from (select

OPERATIONS.USER\_ID as USER\_ID,

to\_char(REPORTS.DAT, *'MM.YYYY'*) as rep\_date,

count(OPERATIONS.id) as cnt

from REPORTS

join OPERATIONS on REPORTS.ID = OPERATIONS.REPORT\_ID

where to\_char(REPORTS.DAT, *'YYYY'*) = *'2018'*

group by (to\_char(REPORTS.DAT, *'MM.YYYY'*), OPERATIONS.USER\_ID)

order by count(OPERATIONS.id) )

# **Как оптимизировать запросы в SQL?**

В этой статье хочу рассказать о некоторых приемах, позволяющих значительно ускорить работу с SQL операторами. *(Источник: личный опыт)*

[Обсудить](https://vc.ru/newtechaudit/113408-kak-optimizirovat-zaprosy-v-sql?comments)

29 052 просмотров

Вот некоторые из них:

**1.** Используйте конкретные имена столбцов после оператора select, вместо «\*» – это позволит увеличить быстроту отработки запроса и уменьшению сетевого трафика.

**2.**Сведите к минимуму использование подзапросов.

Например, запрос

Select Column\_A

From Table\_1

Where Column\_B = (Select max (Column\_B From Table\_2)

And Column\_C = (Select max (Column\_C From Table\_2)

And Column\_D = ‘position\_2’

выглядит значительно хуже на фоне аналогичного запроса:

Select Column\_A

From Table\_1

Where (Column\_B, Column\_C) = (Select max (Column\_B), max (Column\_C)

From Table\_2)

**3.**Используйте оператор IN аккуратно, поскольку на практике он имеет низкую производительность и может быть эффективен только при использовании критериев фильтрации в подзапросе.

**4.**Соединение таблиц в запросе также является критичным: в случае, когда соединение таблиц происходит в правильном порядке, то общее число строк, необходимых к обработке, значительно сократится.

При соединении основной и уточняющей таблиц убедитесь, что первой будет основная таблица, в противном случае вы рискуете получить обработку гораздо большего числа строк, чем необходимо.

**5.** При соединении таблиц EXIST предпочтительнее distinct (таблицы отношения «один-ко-многим»).

6. Избыточность при работе с SQL – это критичная необходимость, используйте в разделе WHERE как можно больше ограничивающих условий.

Например, если указан

WHERE Column\_А=Column\_В and Column\_А=425

вы сможете вывести результат, где Column\_В=425, однако при задании условий

WHERE Column\_А=Column\_В and Column\_B=Column\_C

оператор не сможет определить, что Column\_A=Column\_C.

**7.** Пишите простые запросы. Больше упрощайте. Оптимизатор может не справиться со слишком сложными операторами. Кроме того, иногда выполнение нескольких простых до невозможности операторов дает лучший результат по сравнению со сложными и позволяет добиться лучшей эффективности.

**8.**Помните, что одного и того же результата можно добиться разными способами. Например, оператор MINUS выполняется гораздо быстрее, чем запросы с оператором WHERE NOT EXIST. Запрос с данным оператором в самом общем виде выглядит следующим образом:

Select worker\_id

From workers

MINUS

Select worker\_id

From orders

Этот пример показывает все значения worker\_id, которые содержаться в таблице workers, не в таблице orders. Другими словами, если бы значение worker\_id одновременно присутствовало в таблицах workers и orders, то значение worker\_id не вывелось в результат, поскольку нет конкретики, содержание какой именно таблицы вывести как результат отработки запроса.

**9.**Оформляйте повторяющиеся коды в пользовательскую процедуру. Это может значительно ускорить работу, уменьшить сетевой трафик.

Таким образом, рассмотренные нами моменты работы с SQL операторами и запросами значительно ускоряют работу с СУБД.

В заключение хочется отметить, что очень важно при работе с SQL – мыслить шире, чем границы поставленной перед вами задачи

Всегда старайтесь оптимизировать запрос, какой бы не была мощной инфраструктура, даже ее производительность может снизиться при выполнении неоптимизированных запросов. Учитывайте все необходимые условия при работе с операторами таким образом, чтобы нагрузка на базу была минимальной.