

Задача.

Подготовить выгрузку, содержащую информацию, за период нарушения ПДД с 2018-04-01 по 2018-04-20 с детализацией до даты нарушения. Выгрузка должна содержать атрибуты:

- Дата нарушения ПДД;
- Число нарушений ПДД всего;
- Число нарушений ПДД у которых результирующий код обработки нарушения ПДД равен 0;
- Число нарушений ПДД по которым была операция обработки тип 29;
- Число нарушений ПДД по которым не было операции обработки тип 29, а результирующий код обработки нарушения ПДД равен 0;

Для выгрузки информации можно использовать следующие запросы:

Запрос с использованием подзапросов:

SELECT

```
viol_date,  
(SELECT COUNT(*) FROM viols WHERE DATE(time_check) = viol_date) AS total_viol,  
(SELECT COUNT(*) FROM viols WHERE DATE(time_check) = viol_date AND refuse_code = 0) AS  
zero_refuse_viol,  
(SELECT COUNT(*) FROMopers WHERE oper_code = 29 AND tr_viol_id IN (SELECT tr_viol_id FROM viols  
WHERE DATE(time_check) = viol_date)) AS type29_viol,  
(SELECT COUNT(*) FROM viols WHERE DATE(time_check) = viol_date AND refuse_code = 0 AND tr_viol_id  
NOT IN (SELECT tr_viol_id FROMopers WHERE oper_code = 29)) AS zero_refuse_without_type29_viol  
FROM  
(SELECT DISTINCT DATE(time_check) AS viol_date FROM viols WHERE DATE(time_check) BETWEEN '2018-  
04-01' AND '2018-04-20') AS dates  
ORDER BY viol_date;
```

В представленном запросе мы используем 4 подзапроса, реализующие каждый из требуемых атрибутов.

Первый подзапрос возвращает общее количество нарушений для конкретной даты. Оборачиваем `time_check` в функцию `DATE` для того, чтобы преобразовать дату в формат (YYY-MM-DD), отбросив часы, минуты, секунды.

Второй подзапрос возвращает количество нарушений у которых результирующий код `refuse_code` равен нулю.

Третий подзапрос возвращает количество нарушений по которым операция обработки `oper_code` равна 29.

Четвертый подзапрос возвращает количество нарушений по котором не было операции обработки 29, а результирующий код равен 0. В данном подзапросе мы обращаемся сначала к таблице `viols`, выбираем `refuse_code` = 0, а затем формируем еще один подзапрос который возвращает все `tr_viol_id` которые не содержатся в таблице `opers` с `oper_code` = 29.

Затем группируем подзапросы в основной, который выводит информацию между датами 2018-04-01 и 2018-04-20. В конце сортируем даты по возрастанию с помощью ключевого слова `ORDER BY`.

Данный запрос является не оптимальным, так как каждый раз приходится обращаться к дате `viol_date`, что может привести к долгому времени выполнения.

Запрос с использованием CTE:

WITH

```
violations AS (  
    SELECT  
        DATE(time_check) AS viol_date,  
        COUNT(*) AS total_viol,  
        COUNT(*) FILTER (WHERE refuse_code = 0) AS zero_refuse_viol,  
        COUNT(*) FILTER (WHERE exists(SELECT tr_viol_id FROM opers WHERE tr_viol_id = v.tr_viol_id AND  
oper_code = 29)) AS type29_viol,  
        COUNT(*) FILTER (WHERE NOT exists(SELECT tr_viol_id FROM opers WHERE tr_viol_id = v.tr_viol_id  
AND oper_code = 29) AND refuse_code = 0) AS zero_refuse_without_type29_viol  
    FROM viols AS v  
    GROUP BY viol_date  
)  
SELECT * FROM violations  
WHERE viol_date >= '2018-04-01' AND viol_date < '2018-04-21'  
ORDER BY viol_date;
```

В представленном запросе мы создаем общую таблицу выражений с именем `violations`, в которой применяем функцию **DATE** к полю `time_check`, тем самым, возвращаем дату нарушения, а не полный временной формат.

Далее мы используем агрегатные функции **COUNT** и **FILTER** для подсчета необходимых нарушений. Считаем общее количество нарушений.

Затем количество нарушений, у которых результирующий код равен 0.

Далее, с использованием подзапроса к таблице `opers`, возвращаем все нарушения, по которым операция обработки равна 29.

Затем таким же образом возвращаем все нарушения, которые не содержатся в запросе, аналогичном предыдущему, и добавляем условие требуемого результирующего кода. В конце обращаемся к данной таблице и фильтруем результаты запроса, используя условие **WHERE**, чтобы выбрать только те записи, которые соответствуют периоду времени с 2018-04-01 до 2018-04-21. И наконец, мы сортируем результаты по возрастанию даты нарушения с помощью ключевого слова **ORDER BY**.

Данный запрос выглядит более лаконичным, чем предыдущий, однако использование общей таблицы выражений (CTE) является избыточным для решения поставленной задачи. Кроме того, так как CTE таблица хранится в оперативной памяти, это может стать проблемой при работе с очень большой базой данных.

Запрос с использованием Join:

```
SELECT
  DATE(time_check) AS viol_date,
  COUNT(*) AS total_viol,
  COUNT(CASE WHEN v.refuse_code = 0 THEN 1 END) AS zero_refuse_viol,
  COUNT(CASE WHEN o.oper_code = 29 THEN 1 END) AS type29_viol,
  COUNT(CASE WHEN o.oper_code IS NULL AND v.refuse_code = 0 THEN 1 END) AS
zero_refuse_without_type29_viol
FROM
  viols AS v
  LEFT JOINopers AS o ON v.tr_viol_id = o.tr_viol_id AND o.oper_code = 29
WHERE time_check >= '2018-04-01' AND time_check < '2018-04-21'
GROUP BY viol_date
ORDER BY viol_date
```

В этом запросе мы выбираем данные из таблицы `viols` и делаем `LEFT JOIN` с таблицей `opers`. Используем именно LEFT так как в таблице `opers` могут содержаться не все даты нарушений. Затем мы применяем функцию `DATE` к полю `time_check`, аналогично предыдущему запросу. Далее мы используем агрегатную функцию `CASE WHEN`, вместо `FILTER` для подсчета необходимых нарушений.

Затем мы фильтруем результаты запроса, используя условие `WHERE`, чтобы выбрать только те записи, которые соответствуют периоду времени с 2018-04-01 до 2018-04-21.

Далее мы группируем данные по дате нарушения с помощью ключевого слова `GROUP BY`. И наконец, мы сортируем результаты по возрастанию даты нарушения с помощью ключевого слова `ORDER BY`.

Данный запрос является наиболее оптимальным для решения поставленной задачи, а так же выглядит более читаемым, чем предыдущие.