Пузиков А.Ю.

Аннотация.

Ключевые слова: базы данных, оптимизация селекторных запросов, индексы, бинарные деревья, КЛАДР, ФИАС.

Постановка задачи: В настоящее время проблема оптимизация запросов к базам данных возникает повсеместно. Многие структуры, к примеру КЛАДР или ФИАС – всероссийский классификатор адресов, содержат порядка миллиона записей, а значит оптимизация в подобных системах просто необходима. Такая БД будет полезна любому предприятию Омской области, в случае, если нужно будет быстро и точно получить данные о населенных пунктах РФ. При работе с БД КЛАДР основная таблица KLADR, которая представлена в **табл. 1**, где в основном стоят задачи выборки данных, так как внесение новых данных - это прерогатива гос. органов.

Таблица 1.

| Наименование (назначение) поля | Обозначение поля | Размер поля |
| --- | --- | --- |
| Порядковый номер (первичный ключ) | ID | 256 |
| Наименование | NAME | 40 |
| Сокращенное наименование типа объекта | SOCR | 10 |
| Код | CODE | 13 |
| Почтовый индекс | INDEX | 6 |
| Код ИФНС | GNINMB | 4 |
| Код территориального участка ИФНС | UNO | 4 |
| Код ОКАТО | OCATD | 11 |

Данная таблица содержит полный перечень всех географических местоположений РФ с их региональными идентификаторами (поле CODE). Структура кодового обозначения в поле CODE:

СС РРР ГГГ ППП АА, где

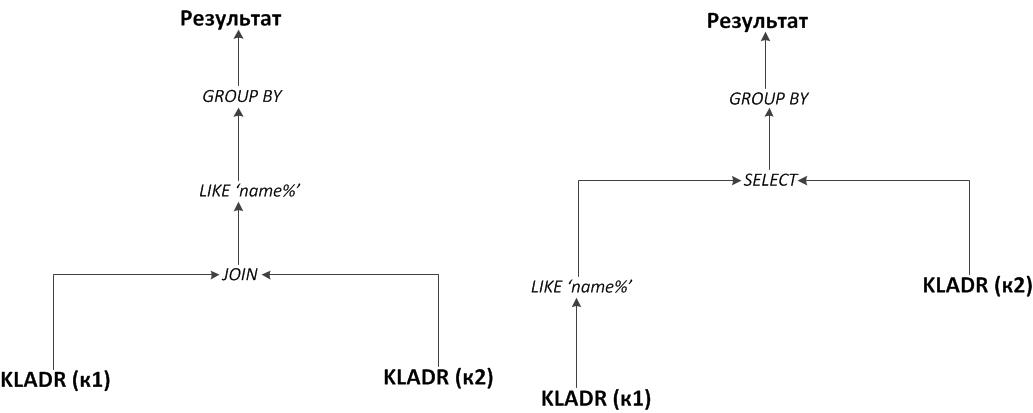
СС – код субъекта Российской Федерации (региона), коды регионов представлены в Приложении 2 к Описанию классификатора адресов Российской Федерации (КЛАДР), остальные коды в данный момент нам не будут интересны.

Особенностью таблицы является то, что самый старший территориальный объект всегда стоит раньше своих включений (его ID минимален для всего набора записей с определенным CODE). Среди запросов выборки можно особо выделить запросы, в которых требуется найти не только какой-либо географический объект, но также отобразить объект – родитель, а если их несколько, то всю иерархию. Как видно из содержания таблицы KLADR, такие запросы будут подразумевать подзапросы или операторы join к той же таблице.

Далее сформулируем запрос в формальном виде: «Получить названия населенных пунктов, начинающиеся с определенной последовательности символов и их типы вместе с названиями и типами территориальных объектов, в которые они включены». Во многих книгах предлагается при составной выборке отдавать предпочтение запросам с JOIN оператором. На языке SQL такой запрос выглядит так:

select k2.code as oblcode, k2.socr as socrobl,k2.name as nameobl, k1.SOCR as socrcity, k1.NAME as namecity, k1.code as citycode from KLADR as k1 left join kladr as k2 on k2.code = k1.code where k1.NAME like 'имя%' group by k2.code , k1.name , k1.socr

Но, если взглянуть на последовательность выполнения запроса, получим:



На рисунке 1 изображена ситуация А, при которой сначала выполняется операция join двух таблиц по полю code. Такая операция будет представлять собой декартово произведение размера *n* x *n*. Поэтому следует рассмотреть ситуацию Б, в которой анализ введённых данных происходит до поиска непосредственных родителей объекта. Запрос такого вида можно записать только с использования вложенных (nested) подзапросов:

select k2.SOCR as socrobl, k2.CODE as oblcode, k2.NAME as nameobl, k1.SOCR as socrcity , k1.NAME as namecity, k1.code as citycode from KLADR as k2,

(select k3.SOCR, k3.NAME, k3.code from KLADR as k3 where k3.NAME like 'имя%' )

as k1 where k2.code = k1.code group by k2.code , k1.name , k1.socr

В таблице 2 представлены результаты выполнения приведенных двух запросов с различными входными параметрами – среднее время выполнения в секундах. Лучшие параметры были получены при использовании вложенных запросов с комбинированным индексом (на основе бинарных деревьев) по полям NAME и CODE. Это произошло, несмотря на тот факт, что на внутренние подзапросы СУБД потратила 80-90% времени выполнения запроса.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Запрос | без индекса | индекс по NAME | индекс по CODE | индекс по NAME CODE | индекс по CODE NAME |
| join | 14,8705125 | 10,43575 | 2,91225 | > 10 сек. | 3,665 |
| nested | 3,871875 | 2,52875 | 2,925875 | 2,465 | 3,549 |

Если поддержка обновления базы в будущем не предусмотрена, что не рекомендуется, то для увеличения производительности следует:

1. поставить ограничения типов искомых объектов (поле SOCR);
2. разбить основной запрос на два запроса (отдельно населенный пункт и его область);
3. реорганизовать схему БД на несколько таблиц, отвечающих за конкретный тип географического объекта.

Библиографический список.

Пузиков Александр Юрьевич, магистр, инженер 2ой категории отдела информационных технологий, филиал АО «Гирпотрубопровод» - ОАО «ОмскГипротрубопровод», тел. 89618495271, адрес эл. почты [gooogleshandi@gmail.com](mailto:gooogleshandi@gmail.com),