

SQLite КАК ХРАНИЛИЩЕ ПРОСТРАНСТВЕННЫХ ДАННЫХ

А.Г.Пискунов

Национальный авиационный университет, Киев, Украина

agp1@sql.ru

Работа посвящена исследованию возможностей встроенной реляционной базы данных SQLite для хранения пространственных данных, а, именно, различных видео- и фотоматериалов с учетом расположения как самих кадров, так и объектов, обнаруженных на них, на поверхности Земли. В связи с опытом использования реляционной базы данных (а именно Paradox 3.5) для хранения цифровых географических карт см. работы [10, 7, 6, 12, 11, 8] было решено использовать бесплатную распространяемую с открытым кодом систему управлениями реляционных баз данных - SQLite [3]. С целью исследования возможностей выбранной СУБД цифровые карты Украины (масштаба 1:500000, содержащую 25000 объектов и полмиллиона точек) и Киевской области (масштаба 1:200000, содержащую 65000 объектов и около миллиона точек) были переконвертированы в базы данных SQLite. Внутренняя схема и схема реляционных связей осталась практически такая же как в картографической базе данных (далее - КБД). Электронную версию статьи [8] с описанием КБД на основе СУБД Paradox 3.5 можно посмотреть по адресу [9]. Небольшие изменения внутренней схемы связаны с:

- введением суррогатных ключей, это поля *id*;
- дальнейшей нормализацией, а именно была удалена таблица *Decoder*, а вместо неё были добавлены две таблицы *Sign* (характеристики географических объектов) и *Code* (коды для некоторых характеристик географических объектов - у таких характеристик в поле *type* имеет значение *S*);
- использованием таблицы *SgmStr* для хранения точек сегментов, вместо поля типа blob (Binary Large Object - Большой Двоичный Объект);
- введением во внутреннюю схему двух индексов, которые будут обсуждаться ниже.

Текущая версия схемы реляционных связей представлена на 1 .

Как и в [7, 8] объект (запись в таблице *Object*) имеет тип локализации (точка, линия, многоугольник), код из классификатора [4], несколько характеристик соответственно коду объекта (таблица *ObjSign*), равно как и несколько ломаных, называемых сегментами, которые хранятся в таблице *Segment*. Точки сегментов хранятся в таблице *SgmStr*. Для уменьшения времени выполнения запроса к КБД и сегментам, и объектам дополнительно заранее был высчитан минимальный прямоугольник со сторонами параллельными осям координат, содержащий все точки образующие сегмент и/или объект. Функция, ставящая в соответствие сегменту или объекту такой прямоугольник далее будет обозначаться символом *box*, сам такой

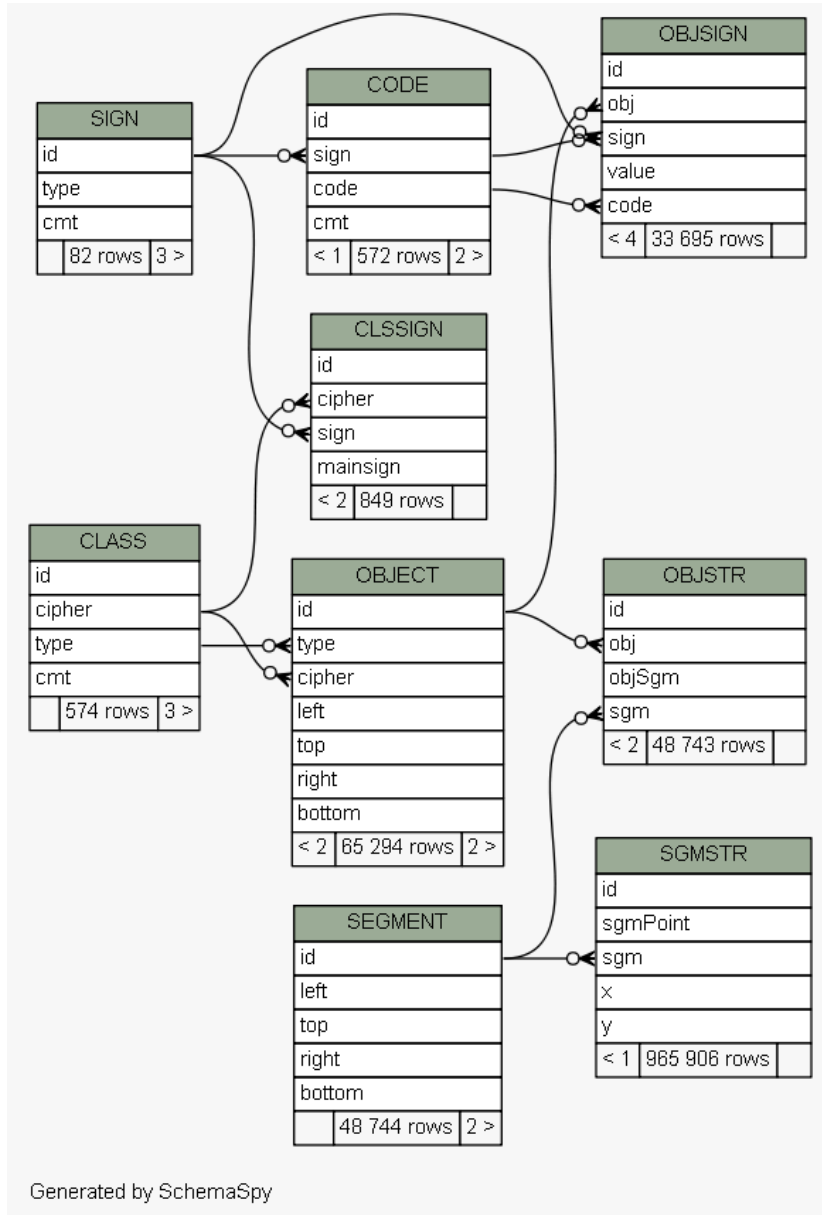


Рис. 1: Схема реляционных связей

прямоугольник будет называться рамка. Далее, через Q обозначим прямоугольник, содержащий точки объектов, которые должны быть отрисованы на экране в текущем запросе.

Версия редактора электронных карт, описанная в [11], сначала находила множество записей из таблицы *Object*:

$$O = \{o | box(o) \cap Q \neq \emptyset\}$$

для объектов, рамки которых имеют непустое пересечение с прямоугольником запроса. Затем, по полученному множеству записей об объектах, строила множество записей из таблицы *Segment* (с запоминанием способа рисования данного сегмента) пересечение рамок которых с прямоугольником запроса тоже оказывалось не пустым. Далее, для каждого сегмента из поля *sgmStr* таблицы *Segment* извлекались координаты его точек и сегмент отрисовывался на экране.

Текущая версия обозревателя КБД, написанная на языке C# с подключением ADO - провайдера SQLite [2], иначе выполняет поиск требуемых объектов для рисования.

Во-первых, в момент события загрузка окна (грубо говоря, в момент запуска приложения) один раз выполняется запрос:

```
select s.id, s.left, s.top, s.right, s.bottom, o.cipher
from segment s, objstr os, object o
where s.id = os.sgm
and os.obj = o.id
```

который создает список S всех номеров сегментов из КБД с географическим кодом (см. [4]) того объекта, в который входит сегмент. В дальнейшем код определяет способ отрисовки данного сегмента. Собственно время выполнения этого первого запроса не превышает двух секунд.

Во-вторых, при каждом изменении прямоугольника Q , для сегментов из S , пересечение рамок которых с прямоугольником запроса не пусто, выбираются их точки:

```
select x, y
from SgmStr
where sgm = @sgm
order by sgmPoint
```

и полностью строится хеш-таблица T из пар (номер_сегмента, список_его_точек).

По окончании построения таблицы T выполняется отрисовка карты. На рисунке 2 представлено окно с прорисованной картой Киевской области.

Далее, так как для поиска точек при выполнении запроса в редакторе электронных карт, программа последовательно выбирала записи из таблицы *Object*, затем - с использованием первичных ключей - рамок и точек сегментов из таблицы *Segment*, то нужда в использовании дополнительных индексов отсутствовала. Текущий обозреватель карт для нормального выполнения обоих запросов требует введение двух индексов:

- create index objsgm_sgm on ObjStr (sgm);
- ускоряет выполнение первого запроса для построения списка S ;

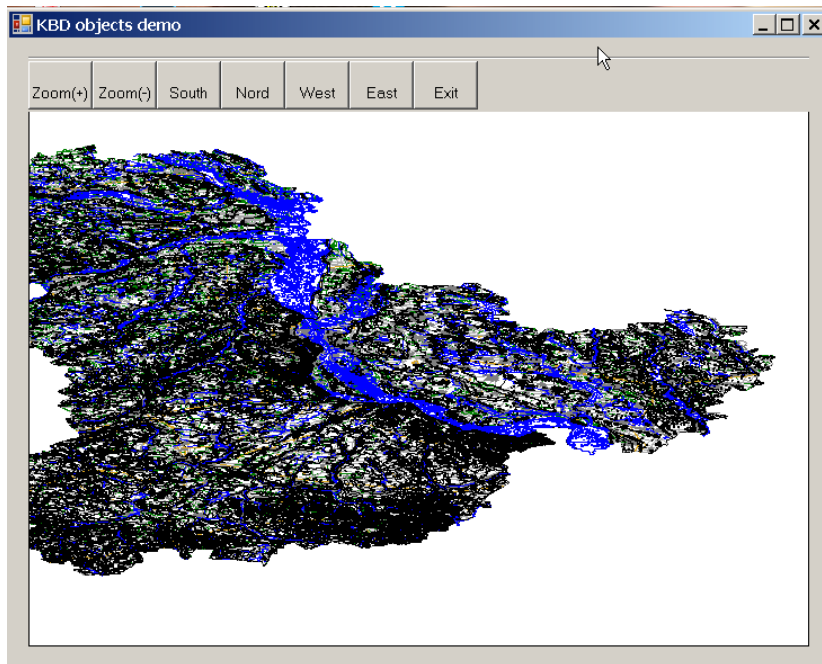


Рис. 2: Полная отрисовка карты Киевской области

- create index sgmenti on SgmStr (sgm, sgmPoint);
- ускоряет выполнение второго запроса для построения хеш-таблицы T .

Так как построение хеш-таблицы каждый раз выполняется полностью, то одним из способов ускорения времени реакции является создание специального кэша на базе R -дерева [5], который не будет перестраиваться целиком при каждом изменении запроса Q и, возможно, фоновой подкачки новых сегментов, как это предлагалось делать в [1].

Обсуждаемый обозреватель КБД на весьма средних компьютерах производства 2005-2007 годов выполняет выборку данных для полной отрисовки карты Украины за 15 секунд, а выборку данных для полной отрисовки карты Киевской области за 30 секунд. Такое время реакции программы трактуется как вполне приемлемое. Кроме того, были поставлены эксперименты по переносу точек сегментов из отдельной таблицы в поле типа blob таблицы *Segment*. Это приводило к существенному (5-8 раз) повышению производительности обозревателя на рассматриваемом количестве точек (до миллиона). Причем скорость выполнения запроса в случае с таблицей *SgmStr*, и в случае без неё падала линейно в зависимости от количества выбираемых точек. Результаты экспериментов собраны на графике 3, где на оси X отмечается количество точек, на оси Y - время выполнения запроса в секундах.

На практике предполагается брать растровые карты из доступных источников. К таким относится, например, проект OpenStreetMap см. [14].

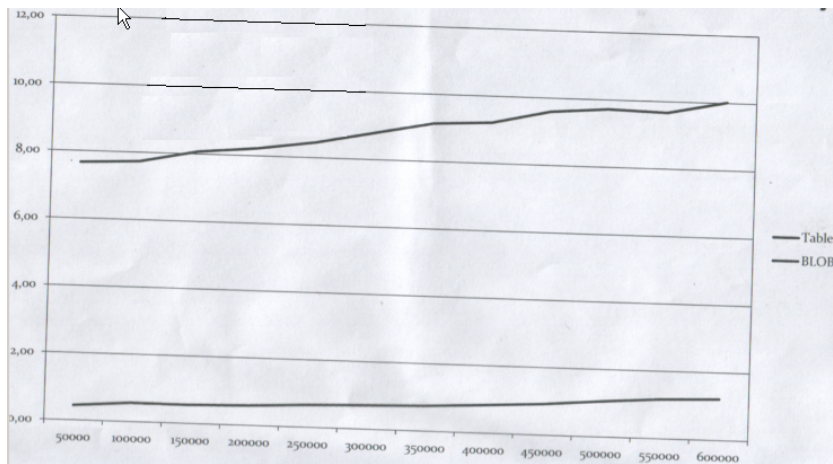


Рис. 3: Время отработки запроса к КБД

При этом, в КБД будет храниться только координаты снимков и объектов, обнаруженных на этих снимках. Предполагается, что общее число записей для хранения требуемой информации будет меньше, чем в двух рассмотренных картах. Вопрос, совмещения растровых карт проекта OpenStreetMap и пространственных данных из КБД прорабатывался в [13]. Таким образом, производительность обозревателя карт на основе СУБД SQLite признана вполне удовлетворительной для предполагаемого использования.

Литература

- [1] . Оптимізація запитів до інформаційних систем обробки геопросторових даних. // Тези XIV міжнародної науково-практичної конференції молодих учених і студентів «ПОЛІТ.Сучасні проблеми науки. Інформаційно-діагностичні системи». - Київ: НАУ - 2015 - 147 с. 4
- [2] ADO-провайдер System.Data.SQLite. <http://system.data.sqlite.org/index.html/doc/trunk/www/downloads.wiki>. 3
- [3] Встраиваемая система управления базами данных SQLite. <http://sqlite.org/>. 1
- [4] Классификатор графических изображений. / Редакционно - издательский отдел, Военно - топографическое управление, ЕСКИККИИ, Москва, 1988. 1, 3
- [5] Антонин Гуттман. R-дерево. http://iipo.tu-bryansk.ru/fileadmin/user_upload/trubakov/book/3_2_1_R_Tree.pdf. 4
- [6] Пискунов А.Г. Архитектура редактора электронных карт / А.Г. Пискунов, В.О. Федченко, В.Г. Малышко. / Тезисы 1-й Украинской конференции по автоматическому управлению « Автоматика-94», Киев - 1994. 1
- [7] Пискунов А.Г. Внутренняя схема системы управления электронными картами / А.Г. Пискунов, В.О. Федченко. / Конференция стран СНГ «Контроль и управление в технических системах». Тезисы докладов. – Винница - 1993. 1
- [8] Пискунов А.Г. Использование реляционной модели при создании картографической базы данных / А.Г. Пискунов, В.О. Федченко, М.В. Николаев. / Сборник научных трудов института кибернетики НАНУ «Применение компьютерных технологий в решении задач народного хозяйства». – Киев - 1996. 1
- [9] Пискунов А.Г. Использование реляционной модели при создании картографической базы данных / А.Г. Пискунов, В.О. Федченко, М.В. Николаев. <http://agp1.hx0.ru/kbd.html>. 1
- [10] Пискунов А.Г. Концептуальная схема системы управления электронными картами / А.Г. Пискунов, В.О. Федченко. / Семинар «Планирование экономического развития регионов с учетом их экологической безопасности», тезисы докладов. – Севастополь - 1993. 1

- [11] Пискунов А.Г. Повышение производительности системы управления электронными картами / А.Г. Пискунов, В.О. Федченко, М.В. Николаев. / Тезисы 2-й Украинской конференции по автоматическому управлению «Автоматика-95», Львов - 1995. [1](#), [3](#)
- [12] Пискунов А.Г. Система преобразования форматов карт / А.Г. Пискунов, В.Д. Барановский, А.А Мокринцев. / Тезисы 2-й Украинской конференции по автоматическому управлению «Автоматика-95», Львов - 1995. [1](#)
- [13] Соснев Л.М. Використання відкритих сервісів геопросторових даних в інформаційних системах на прикладі OpenStreetMap / В.Д. Зівакін, А.О.Колесник, Л.М.Соснев. // Тези XIV міжнародної науково-практичної конференції молодих учених і студентів «ПОЛІТ.Сучасні проблеми науки. Інформаційно-діагностичні системи». - Київ: НАУ - 2015 - 147 с. [5](#)
- [14] Стив Кост. Некоммерческий картографический проект OSM. <http://www.openstreetmap.org>. [4](#)