

Использование реляционной модели при создании картографической базы данных

Предисловие

Обнаружив неподдельный интерес к вопросу хранения различного рода объектов в реляционных базах данных (см. по этому вопросу работы [10, 11, 12]), авторы решили повторно издать свой пример использования реляционной модели для хранения **любых** объектов с **любыми** характеристиками и их пространственными координатами.

Отбросив из рассмотрения атрибуты, хранящие рамку объекта, и таблицы для хранения координат объекта, мы автоматически получаем просто универсальное хранилище объектов. Единственный серьезный вопрос - это разработка подходящего классификатора. Таким образом предлагаемая схема может быть использована вместо или вместе со схемами авторов работ [10, 11, 12] и многих других. В частности, в работе [12] и по адресу [13] приводятся списки статей, связанных с рассматриваемой темой.

Специально хочется отметить, что в своих последующих проектах [14] авторам никогда не хотелось использовать предложенную схему в связи с потерей производительности и потерей возможностей использования условий ограничения целостности, которые есть в промышленных РСУБД. Лично нам кажется вполне достаточный такой подход, предложенный В.Котляревским в [12], как хранение неинтерпретируемых РСУБД и, возможно, часто меняющихся от проекта к проекту, атрибутов объектов в так называемых блоках (BLOB).

Статья была напечатана в кн.: Сборник научных трудов института кибернетики НАНУ "Применение компьютерных технологий в решении задач народного хозяйства", Киев, 1996.

Авторы выражают благодарность Дмитрию Орлову и Виктору Илюхину за помощь в реанимации статьи.

Картографическая база данных (КБД) является низкоуровневым инструментальным средством, разработанным авторами для использования в различных геоинформационных системах и оформленным в виде библиотеки программ, написанных на языке С. О КБД версии 2.0 было объявлено в [3]. При разработке авторы ориентировались на технологии, используемые в украинском аэрогеодезическом предприятии (УкрАГП). КБД предназначена для извлечения информации из классификаторов, принятых в УкрАГП ([5], [6]); извлечения, редактирования, удаления и добавления информации, описывающей объекты и явления, имеющие пространственные координаты. В работе описана концептуальная модель, внутренняя схема, схема реляционных связей и правила проверки целостности КБД. Также рассмотрены некоторые вопросы повышения эффективности геоинформационных систем.

Подготовка данных

В процессе дигитализации участков карты, создаются файлы определенных форматов, содержащих список описаний объектов, изображенных на данном участке карте. Каждое такое конкретное описание состоит из метрической и семантической информации.

Метрическая часть описания представляет собой последовательность точек, составляющую образ объекта. Эта последовательность точек может быть единственной точкой, или точкой с направлением, или ломаной линией или односвязным многоугольником. В ломаных линиях, получаемых соединением точек последовательности прямолинейными отрезками, не допускается самопересечений.

Семантическая часть описания представляет собой код класса географического объекта (последовательность из 8 цифр), тип его локализации (точка, линия или территория), список характеристик, допустимых для данного класса и значение характеристик. Семантическая часть описания объекта используются для правильной интерпретации и отображения соответствующей последовательности точек.

Отметим, что классификаторы УкрАГП имеют древовидную структуру. Причем в случае общегеографического классификатора первые сыновья корня дерева есть группы географических объектов (имеют специальное название - слои), вторые сыновья - некоторые объекты из данной группы, а последующие сыновья есть последовательное уточнение объекта родителя. При движении от корня дерева к листьям коды сыновей получают дописыванием справа в коде родителя ненулевых цифр.

Пример:

```
. . . .
30000000 - гидрография и гидротехнические сооружения
31110000 - океаны и моря
```

```
. . . .
31130000 - водохранилища и др. водоемы
31131000 - водохранилища
31132000 - пруды
31133000 - бассейны
```

```
. . . .
40000000 - населенные пункты
. . . .
```

В классификаторе для топографических планов дерево имеет 3 уровня ветвления и разделяет все коды (XXYYZZZZ) по разделам (левые две цифры XX), подразделам (следующие две цифры YY) и собственно по объектам (правые четыре цифры ZZZZ кода).

```
. . . .
09000000 - объекты гидротехн. и водн. транспорт
09010000 - каналы и канавы
09010100 - канал
```

```
. . . .
09020000 - плотины
```

```
. . . .
10000000 - объекты водоснабжения
. . . .
```

В отличие от файлов дигитайзера, КБД допускает хранение многосвязных объектов, вхождение последовательности точек в образ различных объектов, объединение нескольких объектов в один составной объект. Поэтому общая последовательность точек объекта разделяется на несколько контуров. Сами контуры, в свою очередь, разделяются на т.н. сегменты (последовательность точек, все точки которой, входят в состав одних и тех же объектов). Кроме этого сегменты используются для хранения и отображения очень длинных объектов.

Концептуальная схема КБД

Перейдем к формальному представлению концептуальной схемы КБД с помощью модели Чена "объект - связь" [1].

На концептуальной схеме:

- каждому прямоугольнику соответствует свой класс объектов Чена (это не класс в смысле классификатора, и не географический объект);
- классы связей написаны возле дуг, соединяющие прямоугольники, паре символов 1:1, 1:N, N:1, M:N - соответствующий тип класса связи (один к одному, один ко многим, многие ко одному, многие ко многим);
- список имен атрибутов класса объектов указывается под соответствующим прямоугольником.



Перечислим классы объектов концептуальной схемы:

1. Источник - информация о первоисточнике возникновения объекта КБД, позволяет узнать, откуда появился некоторый географический объект КБД. Так как формат файлов дигитайзера позволяет несколько раз начинать нумерацию объектов внутри файла с 1, был введен атрибут "индекс", служащий для точной идентификации объекта в файле.
2. Объект - список объектов, хранящихся в КБД.
3. Сегмент - список сегментов КБД. Атрибут "размерность" указывает как пользоваться атрибутом "значение" класса объектов точка. В сегменте значение точек может отсутствовать (как в подавляющем количестве сегментов КБД), иметь целый или вещественный тип. Сегменты с присутствующим значением могут использоваться для хранения таблично заданных функций (высота, глубина, уровень загрязнения и т.д.).
4. Точка - список точек сегмента.
5. Описание - список значений характеристики географического объекта, в соответствии с использованным классификатором.
6. Класс - список допустимых классов и возможных типов локализации объекта. Тип локализации принимает следующие значения линейный, площадной, точечный. Они обозначаются буквами L, P, T соответственно. Допустимые классы задаются используемым классификатором географических объектов.
7. Характеристика - список кодов, названий и типов характеристик. Тип характеристик общегеографического классификатора принимает следующие значения - десятичное число, символ, строка, пара десятичных чисел, целое; у промышленного классификатора - текст, вещественное, целое, дата.
8. Значение - список допустимых значений некоторых характеристик. Характеристики типа "символ" географического классификатора и некоторые характеристики типа "текст" промышленного классификатора (а именно - 21 - признак жилого-нежилая, 27 - материал, 35 - огнестойкость, 61 - состояние) имеют заданный набор допустимых значений.

Перечислим классы связей концептуальной схемы:

1. Соответствует - позволяет задать соответствие между объектом и сегментами КБД и объектом файла дигитайзера.
2. Изображает - указывает какие сегменты КБД участвуют в изображении объекта.
3. Описывают - связывают объект и значение его характеристик.
4. Составляет - задает список подобъектов, образующих составной объект.
5. Относится - указывает класс и тип локализации объекта.
6. Требуется - указывает, какие характеристики могут присутствовать в описании объекта заданного класса. Атрибут "обязана" задает обязательность присутствия данной характеристики в описании.
7. Имеет - определяет домен характеристики, т.е. связывает набор допустимых значений и характеристику.

Планирование реляционной схемы КБД

Главным требованием к КБД при разработке было требование обеспечить возможность быстрого отображения на экране дисплея прямоугольного участка карты в любых масштабах, сохраняя возможности ее редактирования. Следовательно, главным и единственным запросом (исключая поиски некоторых записей об объектах и сегментах по их номерам) должен бы быть следующий запрос:

Извлечь номера объектов заданных классов, заданного типа локализации, имеющих непустое пересечение с заданным прямоугольником на карте, >где классы задаются первыми левыми x цифрами кода, 0 <= x <= 8 (маска 6 кода), тип локализации - строчкой составленной из букв L, T, P длины от 1 до 3 символов. Прямоугольник задается координатами его левого верхнего и правого нижнего угла (рамка запроса).

Практически для прямой реализации такого запроса, КБД должна была бы читать информацию по меньшей мере из трех таблиц OBJECT, OBJSTR, SEGMENT (см. ниже внутреннюю схему КБД). Такая реализация была бы достаточно медленной. Для того, что бы добиться возможности построить ответ на запрос за один проход по одной таблице, в КБД было введено понятие рамки объекта и рамки сегмента.

Рамкой назовем упорядоченную четверку чисел - координаты левого верхнего и правого нижнего угла минимального прямоугольника, содержащего все точки объекта или сегмента. Рамки должны быть **рассчитаны заранее** (до выполнения запроса) и хранятся в качестве атрибутов объектов и сегментов.

Следующее ослабление упомянутого запроса назовем КБД - запросом (он выполняется приблизительно в три раза быстрее ранее упомянутого, но требует дополнительной памяти для хранения рамок). Этот запрос реализован в КБД:

Извлечь номера объектов заданных классов, заданного типа локализации, имеющих непустое пересечение рамки с рамкой запроса.

Перед изложением внутренней схемы КБД для реляционной модели данных осталось отметить, что КБД обычно используется вместе с Системой Визуализации Карты (СВК) и Системой Исполнения Запросов (СИЗ). СИЗ - отвечает за исполнение более сложных запросов, чем КБД - запрос, СВК - занимается вычерчиванием объектов на устройствах вывода. Для обеспечения эффективной совместной работы этих систем во внутреннюю схему КБД были добавлены некоторые атрибуты, не имеющие отношения к концептуальной модели.

В частности, к сегменту добавлен атрибут "Направление обхода". Этот атрибут используется СВК для того, чтобы управлять порядком извлечения точек из КБД при построении образа объекта на экране.

Схема КБД

При перечислении совокупности файлов КБД для типов полей таблицы использовались следующие обозначения:

- N - обозначает длинное целое (диапазон значений -2147483647 <= .. <= 2147483647)
- S - короткое целое (-32767 <= .. <= 32767)
- Axx - строка длиной xx символов
- B - внешняя память произвольной длины (BLOB - Binary Large Object)
- поля, отмеченные символом "*" - составляют первичный ключ таблицы
- поля, отмеченные символом "+" - являются вторичным индексом

Таблица CELL

Таблица CELL - моделирует класс связи СОСТАВЛЯЕТ и задает состав объекта.

№пп	Поле	Тип	Ключ	Описание
1	obj	N	*	КБД-номер объекта
2	objCell	S	*	порядковый номер подобъекта
3	cell	N	+	КБД-номер подобъекта

Таблица ОБЪЕКТ

Таблица ОБЪЕКТ - моделирует класс объектов ОБЪЕКТ, класс связей ОТНОСИТСЯ и задает определение объекта.

№пп	Поле	Тип	Ключ	Описание
1	zoom	S	*	зарезервировано для СИЗ
2	type	A2	*	тип локализации
3	cipher	A8	*	географический код объекта
4	obj	N	*+	КБД-номер объекта
5	left	N		рамка объекта
6	top	N		рамка объекта
7	right	N		рамка объекта
8	bottom	N		рамка объекта
9	nmbrL	N		зарезервировано для СИЗ
10	drawMet	S		метод рисования, зарезервировано для СВК

Таблица OBJSIGN

Таблица OBJSIGN - моделирует связь ОПИСЫВАЕТ, объект ОПИСАНИЕ и задает описание объекта.

№пп	Поле	Тип	Ключ	Описание
1	obj	N	*	КБД-номер объекта
2	sign	S	*	номер характеристики
3	value	A10		значение характеристики
4	signRest	B		продолжение значения характеристики

Таблица OBJSTR

Таблица OBJSTR - моделирует связь ИЗОБРАЖАЕТ и задает структуру объекта.

№пп	Поле	Тип	Ключ	Описание
1	obj	N	*	КБД-номер объекта
2	objSgm	S	*	порядковый номер сегмента
3	sgm	N	+	КБД-номер сегмента
4	dir	S		направление обхода
5	loop	S		номер контура сегмента

Таблица SEGMENT

Таблица SEGMENT моделирует объект СЕГМЕНТ

№пп	Поле	Тип	Ключ	Описание
1	sgm	N	*	КБД-номер сегмента
2	left	N		рамка сегмента
3	top	N		рамка сегмента
4	right	N		рамка сегмента
5	bottom	N		рамка сегмента
6	dimension	S		размерность сегмента
7	sgmStr	B		структура сегмента, поле моделирует связь ОБРАЗУЕТ и объект ТОЧКА

Таблица _CONFORM

Таблица _CONFORM - моделирует связь СООТВЕТСТВУЕТ и объект ИСТОЧНИК

№пп	Поле	Тип	Ключ	Описание
1	plan	S	*	номер файла дигитайзера
2	planNo	S	*	номер объекта в файле
3	obj_sgm	A1	*+	признак КБД - номер объект или сегмент

4	dbNo	N	+	КБД номер
---	------	---	---	-----------

Таблица _PLANS

Таблица _PLANS - моделирует объект ИСТОЧНИК

№пп	Поле	Тип	Ключ	Описание
1	name	A8	*	имя файла дигитайзера
2	lineNo	S	*	индекс
3	plan	S	+	номер файла дигитайзера

Таблица \$CLASS

Таблица \$CLASS - моделирует объект КЛАСС

№пп	Поле	Тип	Ключ	Описание
1	Cipher	A9	*	код класса и тип локализации
2	Cnt	A50		название класса

Таблица \$CLSSIGN

Таблица \$CLSSIGN - моделирует связь ТРЕБУЕТ

№пп	Поле	Тип	Ключ	Описание
1	Cipher	A8	*	код класса
2	Sign	S	*	код характеристики
3	MainSign	A1		признак обязательного присутствия

Таблица \$DECODER

Таблица \$DECODER - моделирует объект ХАРАКТЕРИСТИКА, связь ИМЕЕТ и объект ЗНАЧЕНИЕ

№пп	Поле	Тип	Ключ	Описание
1	Sign	S	*	код характеристики
2	Value	A12	*	допустимое значение характеристики
3	Type	A1		тип характеристики
4	Cnt	A65		название характеристики или значения характеристики

Схема реляционных связей КБД



В таблице OBJSIGN, в случае характеристик с заданным (перечисленным) множеством значений, для ссылки на таблицу \$DECODER используется пара (sign, value), или, в противном случае, - просто поле sign

Перечислим множество F функциональных зависимостей КБД. (Несколько перечисленных в скобках атрибутов фактически являются одним атрибутом). Заметим, что используя аксиомы Армстронга [1], можно показать, что F минимально покрывает множество всех функциональных зависимостей КБД.

- obj, objCell -> cell : объект, порядковый номер подобъекта -> подобъект;
- obj -> (left, top, right, bottom) : объект -> рамка объекта;
- obj -> zoom : объект -> зарезервировано для СИЗ;
- obj -> nmbr : объект -> зарезервировано для СИЗ;
- obj -> drawMet : объект -> метод отображения, зарезервировано для СВК;
- obj -> type : объект -> тип локализации объекта;
- obj -> cipher : объект -> код объекта;
- obj, objSgm -> sgm : объект, порядковый номер сегмента -> сегмент;
- obj, objSgm -> dir : объект, порядковый номер сегмента -> направление обхода сегмента;
- obj, objSgm -> loop : объект, порядковый номер сегмента -> контур сегмента;
- obj, sign -> (value, signRest) : объект, характеристика -> значение характеристики;
- sgm -> (left, top, right, bottom) : сегмент -> рамка сегмента;
- sgm -> dimension : сегмент -> размерность сегмента;
- эта зависимость скрыта в поле sgmStr - sgm, sgmPoint -> (vert, horiz, value) : сегмент, порядковый номер точки -> вертикальная координата, горизонтальная координата, значение в точке;
- cipher, sign -> mainSign : код, характеристика -> признак обязательного присутствия характеристики в описании объекта;
- cipher -> cnt : код -> описание кода;
- sign -> type : характеристика -> тип характеристики, описание характеристики;
- sign -> cnt : характеристика -> тип характеристики, описание характеристики;
- sign, value -> cnt : характеристика, значение характеристики -> описание значения;
- name, lineNo -> plan : имя файла, индекс -> номер планшета;

21. plan, planNo, sgm_obj -> dbNo : номер планшета, номер объекта в планшете, признак объект или сегмент -> КБД - номер объекта или сегмента;

Основные отношения КБД относительно F находится в третьей нормальной форме (3NF) и в нормальной форме Бойса - Кодда (BCNF). Из дополнительных отношений (начинаются на символ "_" и символ "\$"), в третьей нормальной форме и в нормальной форме Бойса - Кодда не находится отношение \$DECODER. Во избежание чрезмерного усложнения реализации, процесс нормализации отношений не был доведен до конца.

Трудностей логического порядка, связанных с недостаточной нормализацией таблиц, не должно возникать, так как предполагается, что информация таблиц классификатора (имя таблицы начинается на \$) подвергается изменениям достаточно редко. Дополнительно заметим, что внутренняя схема КБД не имеет многозначных зависимостей, не имеет избыточности и имеет наименьшее из возможных среди эквивалентных схем в BCNF число имен отношений. Поэтому можно считать, что схема КБД "достаточно близка" к оптимальной схеме в BCNF (в смысле [2]).

Считается, что содержимое КБД является корректным, если она удовлетворяет правилам проверки целостности.

Перечислим список правил:

1. Номера объектов (поле obj) в таблице Object являются уникальными.
2. В таблицах (Object, Segment), содержащих рамки, левая координата (поле left) должна быть меньше правой (right);
3. В таблицах, содержащих рамки, нижняя координата (bottom) должна быть меньше верхней (top);
4. В таблицах (для objstr - поле objSgm, для cell - objCell), содержащих порядковые номера, порядковые номера являются конечным, сплошным подмножеством натурального ряда, начинающегося с 1 (для objstr - сегментов в объекте, для cell - подобъектов в объекте, для sgmStr - точек в сегменте);
5. Код объекта должен содержаться в классификаторе;
6. Код характеристики должен содержаться в классификаторе;
7. Каждый сегмент входит в структуру объекта не более одного раза;
8. Каждый подобъект входит в состав объекта не более одного раза;
9. Каждая характеристика входит в описание объекта не более одного раза;
10. Записи описания, структуры или состава объекта могут находиться только при наличии записи определения данного объекта;
11. Определение объекта влечет обязательное наличие записей описания объекта и записей структуры объекта и/или записей состава объекта;
12. В структуру объекта должны входить ссылки только на реально существующие в КБД сегменты;
13. В состав объекта должны входить ссылки только на реально существующие в КБД объекты;
14. В состав объекта не должна входить рекурсивная ссылка на себя как на подобъект (ни прямая, ни через другие объекты).

Опыт применения КБД

КБД версии 2.0 использовалась при создании редактора электронных карт ([4]), на основе которого было создано несколько малых ГИС (геоинформационных систем) для карт разных масштабов, работающих под управлением MS-DOS. В частности, программы Генплан (кадастр промышленного предприятия) и Земельный Кадастр используют карту масштаба 1 : 500 и крупнее; информационно - справочные системы на основе карт Украины, Киевской области и Киева - масштабы 1 : 500000, 1 : 200000, 1 : 25000, соответственно.

Опыт эксплуатации КБД позволил выделить следующие характерные черты:

1. Количество операций ВСТАВИТЬ ЗАПИСЬ крайне мало по сравнению с количеством имеющихся в таблице записей. Например, карта крупного завода содержит 80000 объектов, поэтому ручное редактирование карты не может оказать существенного влияния на содержимое КБД;
2. Пусть для некоторого запроса к карте функция F отображает порядковый номер объекта, удовлетворяющего запросу, в ключ, используемый для поиска метрической части объекта. Эта функция оказалась кусочно монотонно - возрастающей. Причем количество монотонных участков мало, а разность $F(x+1) - F(x) = d$ обычно имеет небольшую величину и очень часто равна 1, где x и x+1 принадлежат монотонному участку.
3. Функция зависимости количества сегментов карты от диаметра сегмента (расстояние между двумя наиболее удаленными точками объекта) имеет экспоненциально убывающий характер. В приводимой ниже таблице не подсчитаны сегменты, состоящие из одной точки. Одноточечных сегментов приблизительно 40000.

Таблица распределения многоточечных сегментов по длинам (в точках)

точек в сегменте меньше чем	общее количество сегментов	процентов от количества всех сегментов
20	35167	87
40	3075	8
60	1045	3
80	445	1
100	226	1
120	133	0
140	79	0
160	55	0
180	45	0
200	24	0
220	24	0
240	9	0
260	8	0
280	7	0
300	3	0
320	4	0
340	1	0
360	3	0
500	2	0

Повышение производительности

Для повышения производительности была создана специализированная подсистема управления таблицей (СПУТС), содержащей сегменты, вместо ранее написанной с использованием универсальной СУБД PARADOX ENGINE.

Перечислим ее свойства:

1. В отличие от универсальных СУБД, она не записывает в таблицу пустые блоки для дальнейших вставок.
2. Для доступа к таблице сегментов в СПУТС используется файл индекса на основе модифицированной версии В*-дерева [7,8], в которой учитывается кусочно монотонно - возрастающий характер функции F. В этой версии ссылки на записи содержатся только в листьях дерева. Повышения производительности удалось достичь за счет специальной стратегии планирования работы страничного пула индекса. В данном случае поиск записи по ключу всегда осуществляется начиная с корня дерева индекса по направлению к листьям. Таким образом, доступ к узлам верхних уровней дерева индекса происходит чаще, чем к узлам нижних уровней. Точнее, если узел в среднем имеет k сыновей, то желательно, чтобы приоритет его страницы в оперативной памяти был в среднем в k раз выше приоритета страниц уровня его сыновей. С учетом вышесказанного была построена приоритетная схема управления вытеснением страниц индекса, которая объединяет приоритетное планирование страниц и метод вытеснения страниц LRU [9] (вытесняется наиболее давно используемая страница данного уровня дерева индекса). Кроме того подсистема управления страничным пулом позволяет оперативно изменять приоритеты страниц, когда точно известно, что данная страница не понадобится в ближайшем будущем. Эти особенности планирования страничного пула эффективно используются подпрограммами управления индексом и позволяют достичь существенного повышения производительности поиска в индексе (иногда в несколько раз). Отметим, что повышение производительности наблюдается не только при поиске в индексе, но и при его создании. Наибольшее увеличение производительности наблюдается при полностью упорядоченных запросах. В случае чисто случайной последовательности запросов производительность практически не хуже, чем в случае использования одной только стратегии LRU.
3. Координаты точек сегмента, в зависимости от его диаметра, записываются 8-, 12-, 16- или 32- битными числами. Это приводит к более плотному хранению сегментов в файле записей переменной длины, повышает вероятность нахождения требуемого сегмента в буфере системы и снижает количество чтения страниц диска. В упомянутой карте завода 20% сегментов записаны 8 - битными числами (диаметр < 256), 57% - 12 - битными числами (диаметр < 4096), 23% - 16 - битными числами (диаметр < 64768) и всего 10 сегментов записаны 32 - битными числами;
4. КБД 3.0 позволяет переносить в таблицу SEGMENT начальный отрезок памяти (одинаковой длины для каждой записи таблицы), хранящейся в файле записей переменной длины (BLOB). Для большого количества коротких сегментов отпадает необходимость открывать запись в файле записей переменной длины, читать ее в свой буфер и закрывать.

Такая модификация КБД привела к существенному уменьшению общего размера файлов, содержащих сегменты, и, приблизительно, двухкратному повышению общей производительности системы. В частности, задача Генплан, построенная с КБД 3.0, на карте завода выполняет типичный запрос по отображению основного окна (на компьютере АТ - 486DX2-66 со средней скоростью передачи (чтение + запись) информации на диске 1.1 Мбайт в секунду) в среднем за 10 секунд.

Литература

1. Ульман Дж. Основы систем баз данных. Москва, Финансы и статистика, 1983.- 334 с.;
2. Голосов А.О., Цаленко М.Ш. Схемы реляционных баз данных: теория нормализации и построение нормальных форм. // Прикладная информатика. Вып. 2, М., Финансы и статистика, 1983, с. 92 - 119.
3. Пискунов А.Г., Федченко В.А. и др. Внутренняя схема системы управления электронными картами. // Конференция стран СНГ. Контроль и управление в технических системах., Тезисы докладов, Севастополь, 1993.
4. Пискунов А.Г., Федченко В.А., Малышко В.Г. Архитектура редактора электронных карт. // Тезисы 1-й Украинской конференции по автоматическому управлению. Автоматика - 94. Киев, 1994.
5. Классификатор графических изображений. Редакционно - издательский отдел, Военно - топографическое управление, ЕСККиКиИ, Москва, 1988;
6. Руководство по созданию цифровых карт местности по картографическим материалам. Редакционно - издательский отдел, Военно - топографическое управление, ЕСККиКиИ, Москва, 1989;
7. Кнут Д., Искусство программирования для ЭВМ. т. 3. Москва, Мир, 1978.- 844 с.
8. Сибуй М., Ямамото Т., Алгоритмы обработки данных. Москва, Мир, 1986.- 218 с.
9. Цикритзис Д., Бернштейн Ф., Операционные системы. Москва, Мир, 1977.- 336 с.
10. А.Тенцер. [База данных - хранилище объектов \(часть 1\)](#), 26.04.2002.
11. Scott W. Ambler, [Mapping Objects To Relational Databases - White Paper](#), . 26-FEB-1999
12. В.Котляревский, [ООП в РСУБД](#)
13. <http://foxserver.nvisual.com/articles.shtml>
14. <http://i.com.ua/~agp1>