Лекция 9

Базовые модели данных

1. Основные понятия моделей данных
2. Классификация моделей данных в ГИС
3. Иерархическая модель
4. Модель квадратомическое дерево
5. Реляционная модель
6. Модель «сущность-связь»
7. Сетевые модели

1. Основные понятия моделей данных

Под *моделью данных* понимают совокупность принципов организации данных.

Модель отражает наиболее общие свойства объекта или исследуемого процесса.

Информационные единицы составляющие основу организации моделей

знак **—** элементарная единица информации, являющаяся реализацией свойств объекта в заранее заданной, структурно организованной знаковой системе.

тип **—** совокупность моделей или объектов, объединенная общим набором признаков, или класс подобных знаков.

типизация **—** объединение данных по набору заданных признаков или выделение из множества данных тех, которые удовлетворяют заданным критериям (или признакам).

***сущность* —** элемент модели (совокупность атрибутов и знаков), описывающая законченный объект или понятие;

· ***атрибут* —** элементарное данное, описывающее свойства сущностей;

***атрибут данных* —** свойство данных;

· ***запись данных*** — формализованное представление сложной информационной модели без описания ее структуры. Запись может быть логической и физической;

· ***запись логическая* —** информационная единица, соответствующая одному шагу обработки информации;

· ***запись физическая* —** порция информации, которая является единицей обмена данными между внутренней и внешней памятью ЭВМ;

- ***Предметной областью*** называется подмножество (часть реального мира), на котором определяется набор данных и методы манипулирования с ними для решения конкретной задачи или исследований.

Для построения модели объекта в виде составляющих частей и определение связей между этими частями применяют методы (процедуры) ***абстракции***, которые тоже образуют целый ряд понятий:

- ***абстракция*** – процедура структуризации (типизации) данных. Различают два вида абстракции: *обобщение и агрегация*.

Обобщение в свою очередь подразделяется на две категории: ***собственно обобщение*** и ***классификация***.

- ***собственно обобщение*** – процедура соотнесения множества типов одному типу соотносится с понятием: «есть часть…»;
- ***классификация*** – процедура соотнесения множества знаков одному типу.

Агрегация – процедура конструирования объекта из других базовых объектов; соотносится с понятием «есть некоторые…».

Под агрегатными данными будем понимать набор данных для формирования объекта из его частей на основе процедур агрегации.

Процедура, обратная агрегации, называется ***пошаговой детализацией***. Она применяется для разбиения агрегатной модели на составные части.

Схема структурирования данных с применением процедур абстракции:
а) прямые процедуры; б) обратные процедуры

Пример построения модели на основе процедур обобщения

Пример построения модели на основе процедур агрегации

2. Классификация моделей данных в ГИС

Все модели данных делятся на статические и динамические модели

К статическим относятся модели инвариантные относительно времени.

Динамические модели не только допускают изменение параметров и структур во времени, но и служат для описания изменения процессов и моделей именно во времени.

Примерами таких моделей в ГИС могут служить два вида электронных карт: электронные карты в режиме разделения времени *(электронные атласы)* - статическая модель, электронные карты в реальном масштабе времени *(навигационные системы)* - динамическая модель.

Классификация моделей по степени типизации

Сильно типизированные – это модели, в которых большинство данных удовлетворяет неким условиям и ограничениям и может быть отнесено к узкому подклассу (типу).

Примером сильно типизированных данных в ГИС служат координатные (метрические) данные и все табличные данные.

Слабо типизированные – это модели, в которых данные разнородны по формату, структуре.

Примером слабо типизированных моделей в ГИС могут быть описательные характеристики (временные наборы данных).

Представление моделей

Выделяют табличные и графовые формы представления моделей.

Табличная форма дает представление модели или ее характеристик в виде одной или совокупности взаимосвязанных таблиц. При этом данные в ячейках таблицы не могут заноситься произвольно, они подчиняются определенным правилам, в частности, по столбцам располагают типизированные данные. Примером табличного представления модели кроме таблицы может служить логическая запись.

Пример Личность / Ф.И.О. / Возраст / Адрес / Социальное положение / Стаж / Зарплата / Специальность

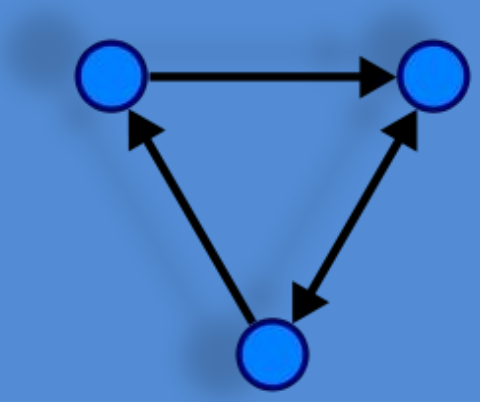
Графовая форма основана на построении модели в виде графической схемы, называемой *графом.* Эта схема включает элементы графа, называемые *вершинами (узлами)* и *ребрами (дугами).*

В отличие от произвольно нарисованной схемы графовая модель, как и табличная, строится по определенным правилам.

В частности, каждое ребро может быть *ориентировано,* если определен путь от одной вершины к другой.

Может быть *не ориентировано,* что соответствует возможному пути от одной вершины к другой в обоих направлениях.

Пример Простейший пример ориентированного графа - вектор в трехмерном пространстве, а неориентированного графа - кривая пути из одной точки в другую.



По форме отображения модели данных делятся на аналоговые и дискретные

Аналоговые **модели в свою очередь разбиваются на две группы:**
прямой и косвенной аналогии.

- **Модели *прямой аналогии* создаются на основе физического моделирования**

  Пример: **Аналоговые карты, модели судов, самолетов, гидротехнические сооружения и т.п.**

- **Модели *косвенной аналогии* - на основе математического моделирования (аналитического описания).**

  Пример: **Цифровая модель рельефа, построенная на основе аналитического описания поверхности.**

Дискретные модели строятся путем замены непрерывных функций набором дискретных значений аргументов и функций. Дискретность определяется *шагом квантования.*

Примером дискретных моделей являются большинство цифровых моделей, на основе которых впоследствии осуществляется аналоговое представление информации

В ГИС РАЗЛИЧАЮТ

БАЗОВЫЕ МОДЕЛИ

СПЕЦИАЛЬНЫЕ МОДЕЛИ

Базовые модели данных

Среди базовых моделей данных выделяют

- Квадратомическое дерево
- Модель «сущность-связь»
- Инфологическая модель
- Иерархическая модель
- Реляционная модель
- Сетевые модели

Специальные модели данных

Среди специальных моделей выделяют:

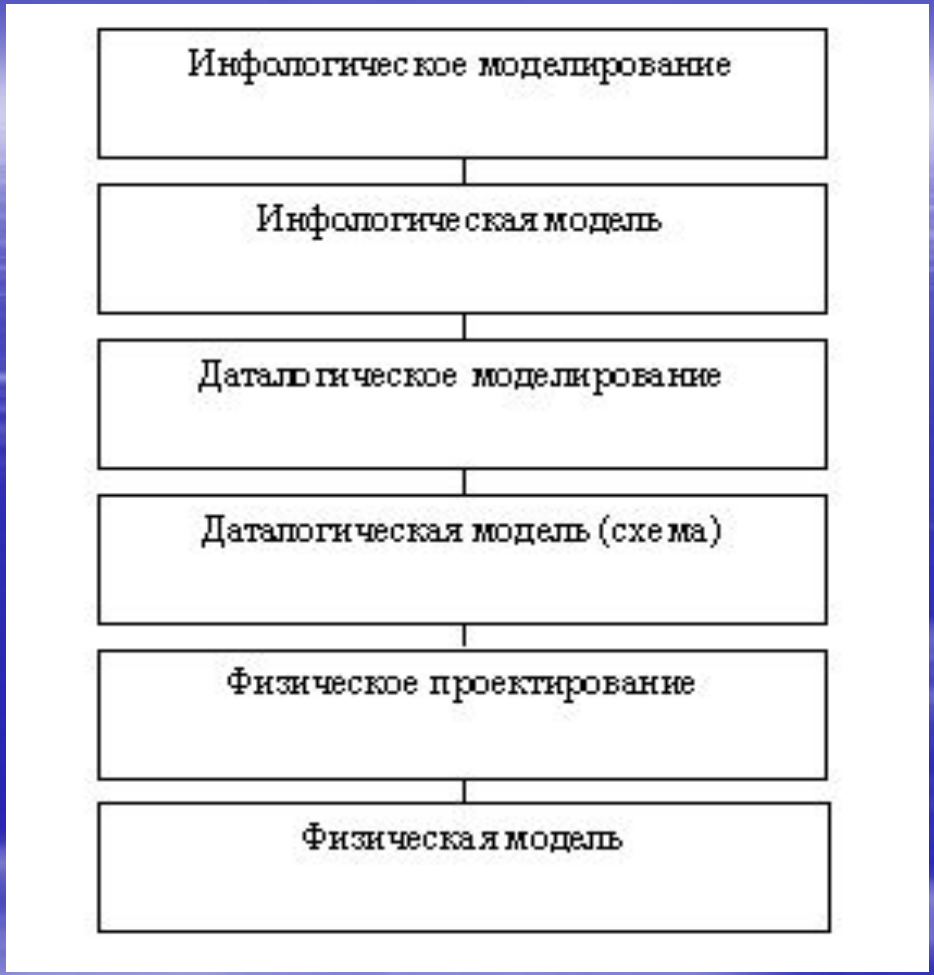
- Растровые
- Векторные
  - Векторные топологические
  - Векторные нетопологические

3. Иерархическая модель

Иерархическая модель относится к наиболее простым структурно определенным моделям.

В этой модели данных связи между ее частями являются жесткими, а ее структурная диаграмма должна быть упорядоченным деревом. При этом, для описания различных уровней модели используют следующие понятия: корень, ствол, ветви, листья и лес.

Обобщенная иерархическая модель представляет собой описание процесса или системы, состоящей из совокупности уровней, связанных одной дугой.



Иерархическая модель проектирования ГИС

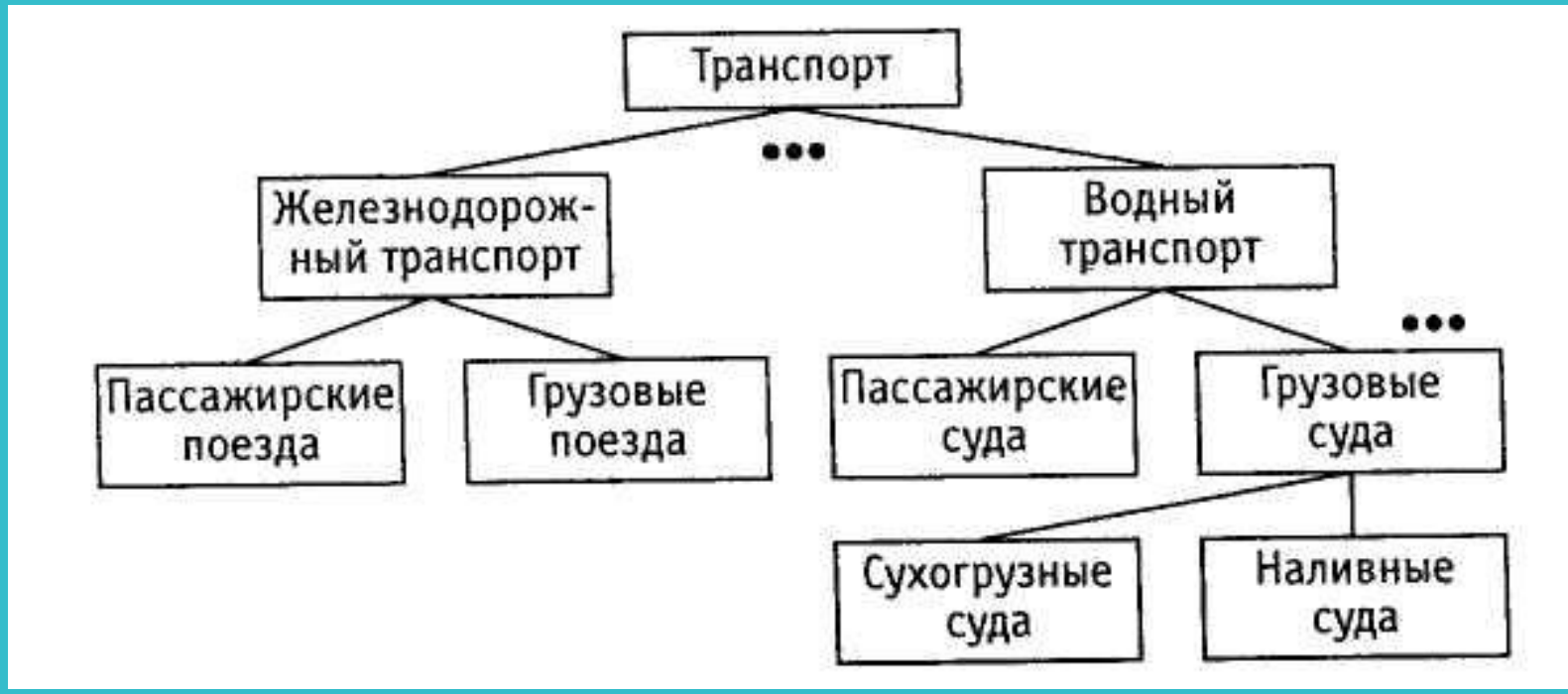
В иерархических моделях данных существует два внутренних ограничения:

1) все типы связей должны быть фукнциональными;
2) структура связей должна быть древовидной.

Следствием этих ограничений является необходимость соответствующей структуризации данных.

В силу функциональности связей запись может иметь не более одной исходной записи любого типа, т.е. связь должна иметь жесткий вид – **ОДИН КО МНОГИМ**.


Я
Мой папа
Моя мама
дедушка
бабушка
дедушка
бабушка
прадедушка
прадедушка
прадедушка
прадедушка
прабабушка
прабабушка
прабабушка
прабабушка


Транспорт
Железнодорож-
ный транспорт
Водный
транспорт
Пассажирские
поезда
Грузовые
поезда
Пассажирские
суда
Грузовые
суда
Сухогрузные
суда
Наливные
суда

Недостатком иерархической модели является снижение времени доступа при большом числе уровней, поэтому в ГИС не используются модели при большом числе уровней (более 10).

Однако, иерархические модели довольно устойчиво применяются для составления различного рода классификаторов.

3. Модель квадратомическое дерево

Данная модель имеет множество названий:

- Квадротомическая модель;

- Квадротомическое представление (данных);

- квадродерево;

- дерево квадратов;

- Q-дерево, 4-дерево (quadtree, quad tree, Q-tree).

Это модель является один из способов представления пространственных объектов в виде **иерархической древовидной структуры** основанный на декомпозиции пространства на квадратные участки (**квадратные блоки, квадранты**), каждый из которых делится рекурсивно на 4 вложенных до достижения некоторого уровня детальности представления (**разрешения**).

Квадратомическое дерево – структура, используемая для накопления и хранения географической информации.

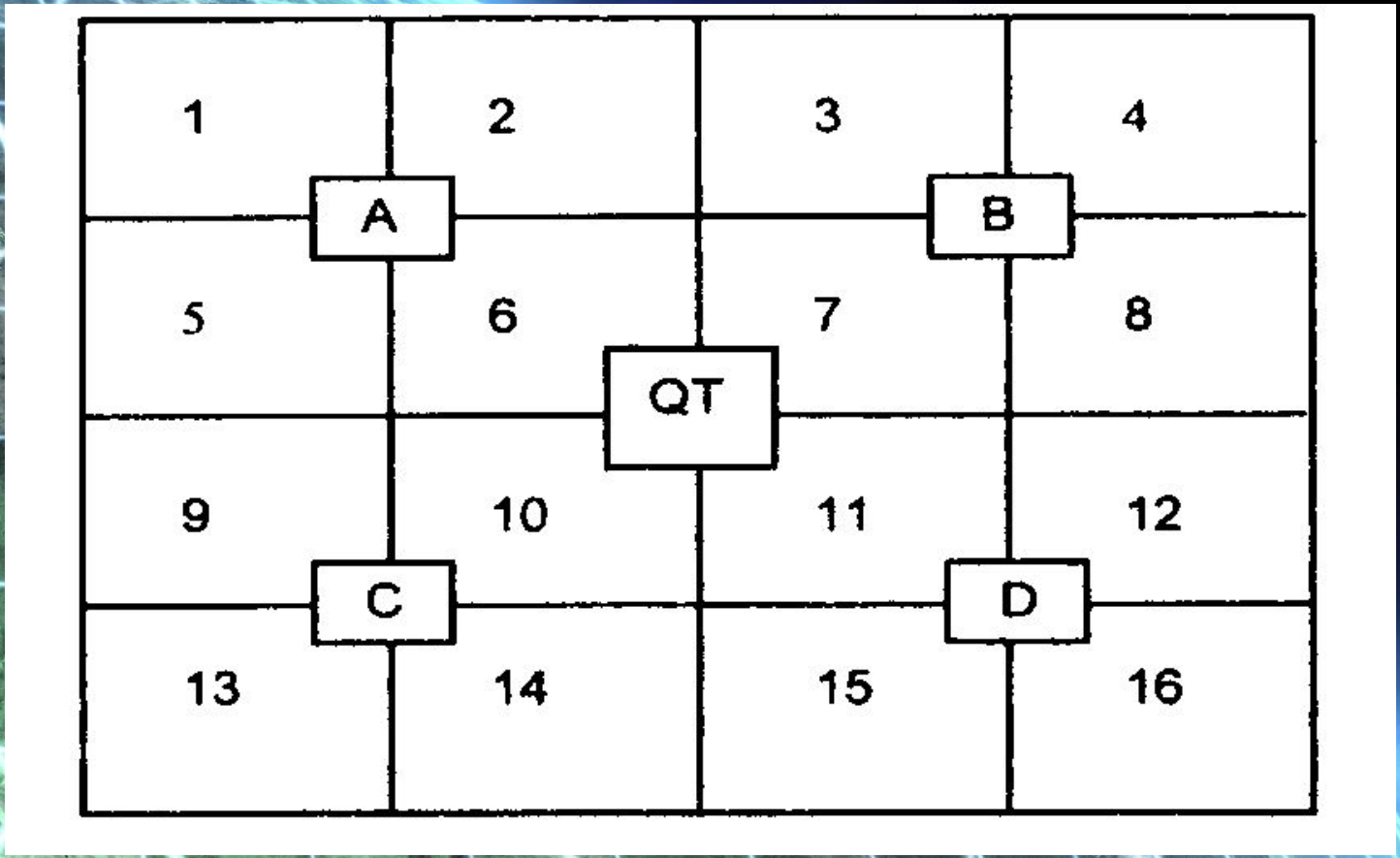
В этой структуре двухмерная геометрическая область рекурсивно подразделяется на квадраты, что определило название данной модели.

На рисунке показан фрагмент двухмерной области Qt, состоящей из 16 пикселей. Каждый пиксель обозначен цифрой. Вся область разбивается на четыре квадранта: A, B, C, D.

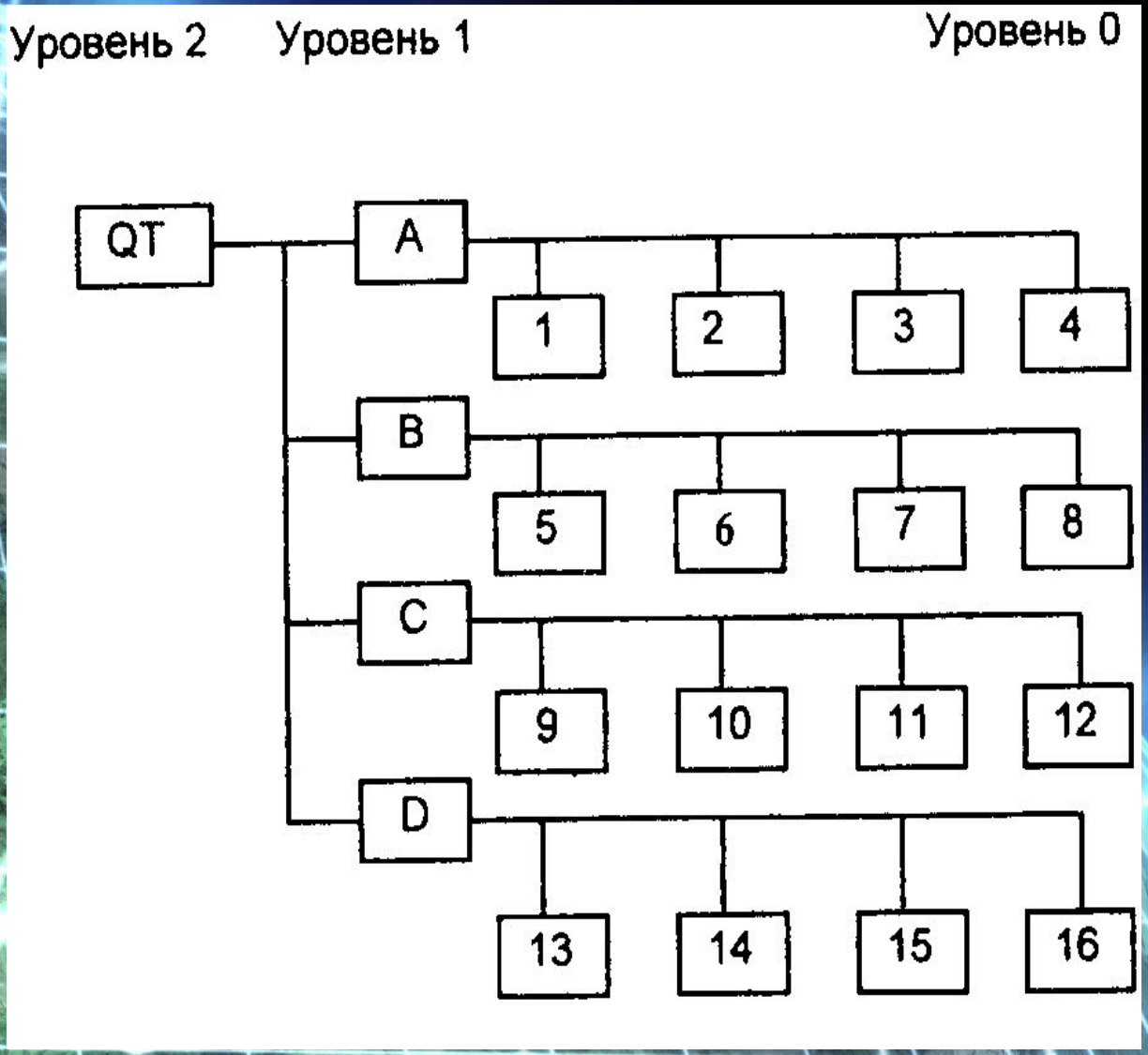
Каждый из четырех квадрантов является узлом квадратомического дерева.

Большой квадрант Qt становится узлом более высокого иерархического уровня квадратомического дерева, а меньшие квадранты появляются на более низких уровнях.

Двухмерная область Qt, состоящая из 16 пикселей



Квадратомическое дерево в виде E-структуры

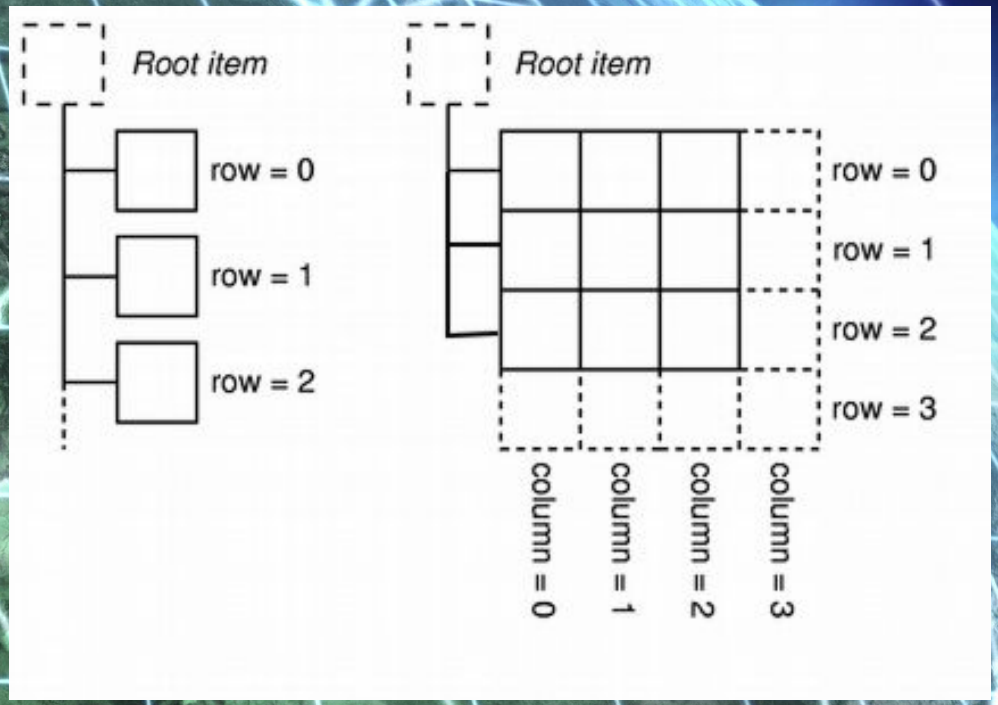


На этом слайде показано квадратомическое дерево, построенное по данным двухмерной области Qt из 16 пикселей.

Как видно, эта структура являет собой классический пример E-дерева.

Преимущество такой структуры состоит в том, что регулярное разделение обеспечивает накопление, восстановление и обработку данных простым и эффективным способом. Простота проистекает из геомет рической регулярности разбиения, а эффективность достигается за счет хранения только узлов с данными, которые

В модели списка можно пользоваться только одним индексным компонентом – номером строки, получить доступ к которому можно с помощью функции QModelIndex::row(). В модели таблицы используется два индексных компонента – номер строки и номер столбца, получить доступ к которым можно с помощью функции QModelIndex::row() и QModelIndex::column().



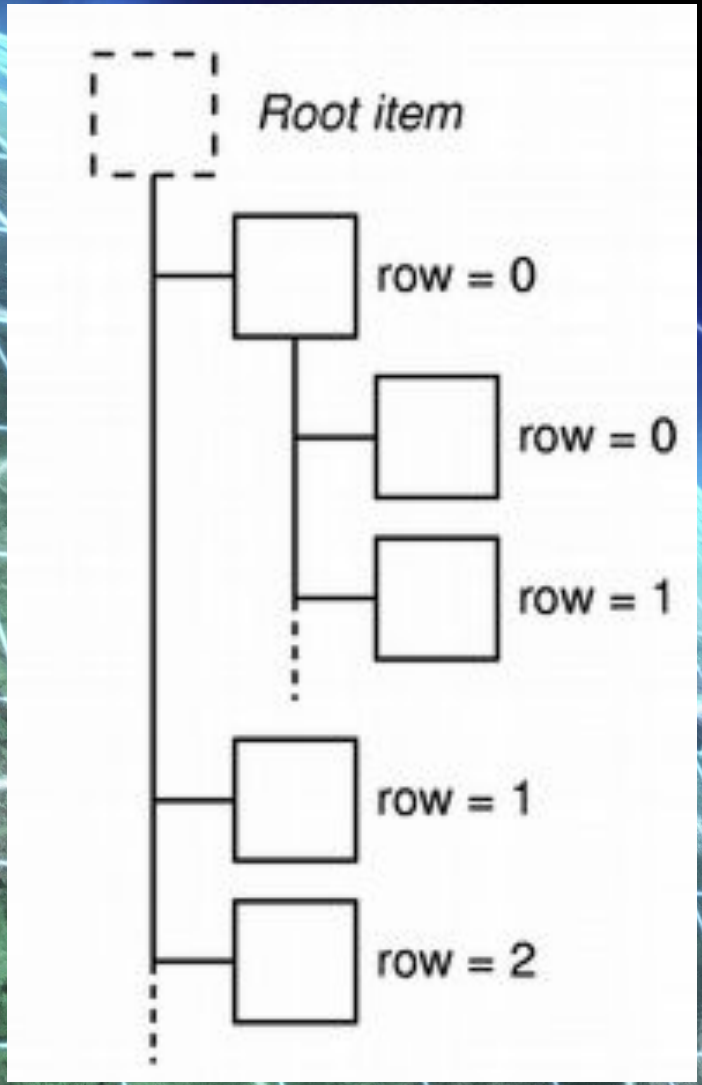
Модель дерева подобна модели таблицы при следующих отличиях.

Как и в модели таблицы, родительский элемент элементов верхнего уровня является корневым, однако родительский элемент любого другого элемента занимает другое место в иерархии элементов.

Доступ к родительским элементам можно получить при помощи функции QModelIndex::parent(). Каждый элемент имеет свои ролевые данные и может иметь или не иметь дочерние элементы, каждый из которых является таким же элементом.

Поскольку любой элемент может иметь дочерние элементы, такую структуру данных можно определить рекурсивно (в виде дерева).

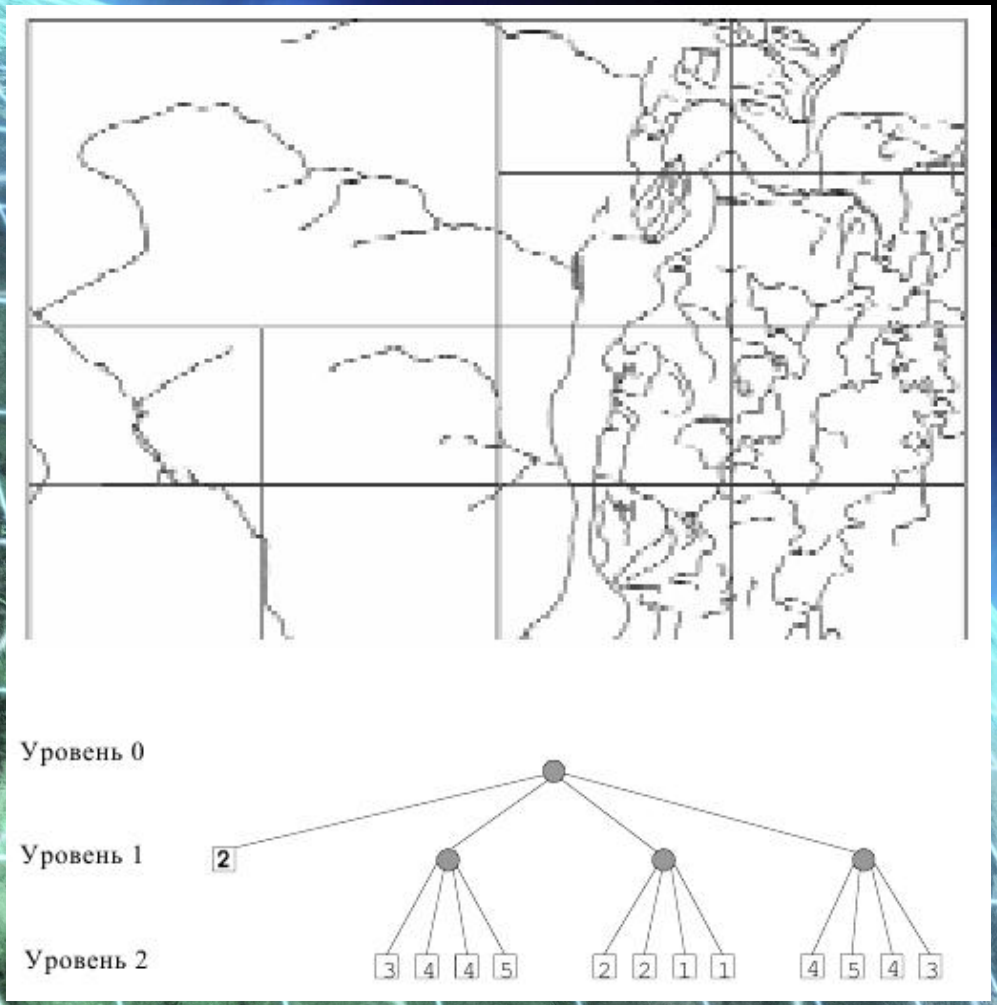
Модель дерева



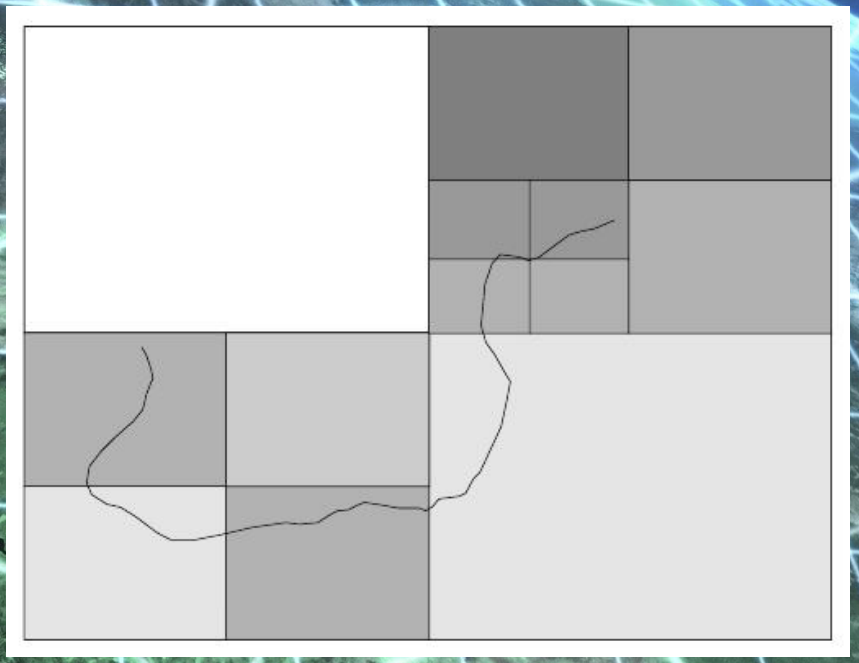
Модели, основанные на квадратомических деревьях, обеспечивают расчеты площадей, центроидные определения, распознавание образов, выявление связанных компонентов, определение соседства, преобразование расстояний, разделение изображений, сглаживание данных и усиление краевых эффектов.

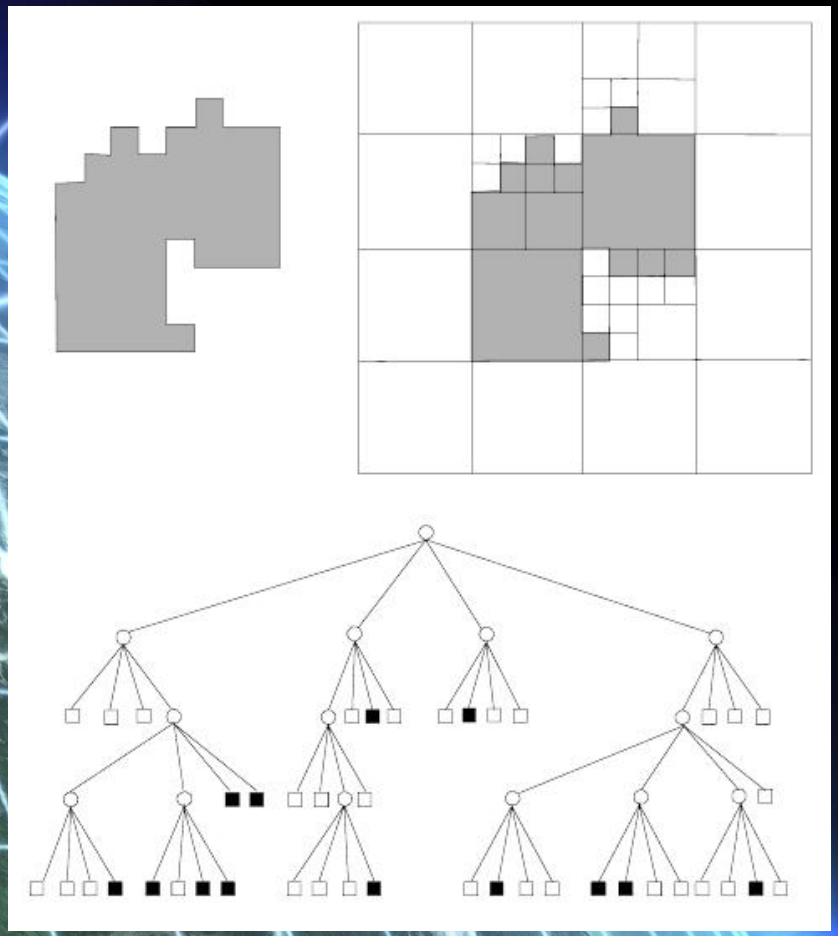
Вследствие этого появилась возможность ис пользовать квадратомические деревья для хранения географических данных. Однако при этом требуется развитие процедур для превращения растровых данных в формат квадратомического дерева и усовершенствование техники линейного кодирования.

Разбиение пространства карты с помощью квадродерева

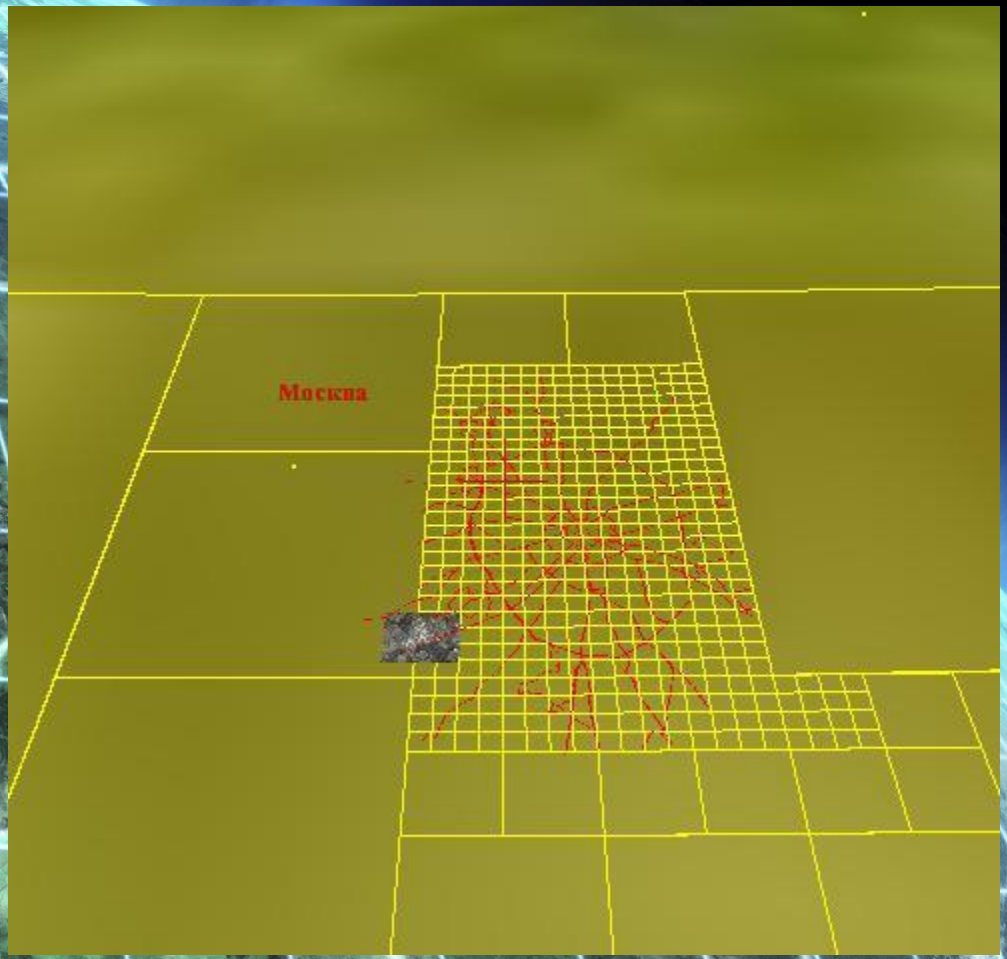


Определение средневзвешенной густоты бассейна реки

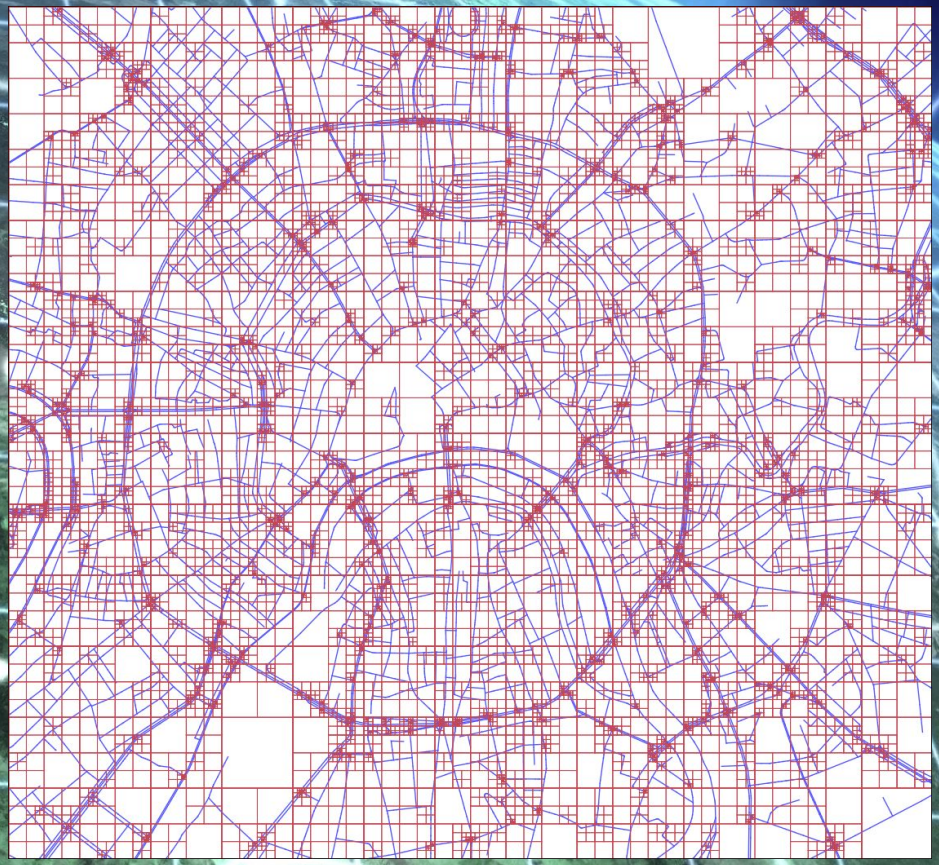


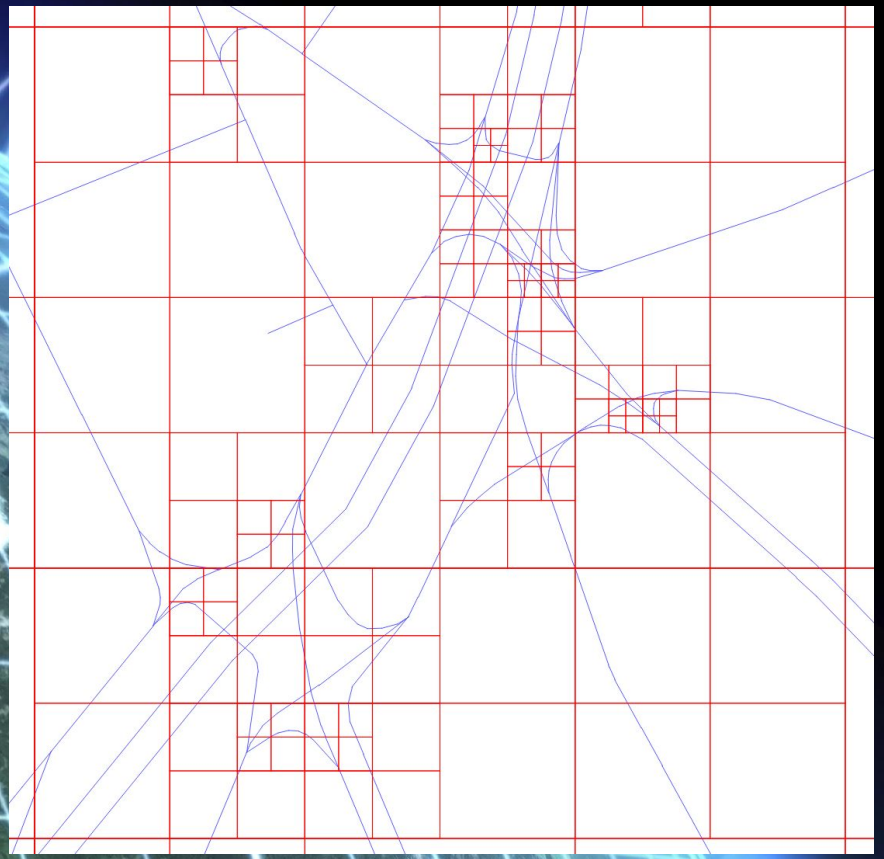


Квадродерево представления векторной информации



Примеры визуализации квадродеревьев





Визуализация карты Москвы по принципу квадродерева

Пример матричного квадродерева

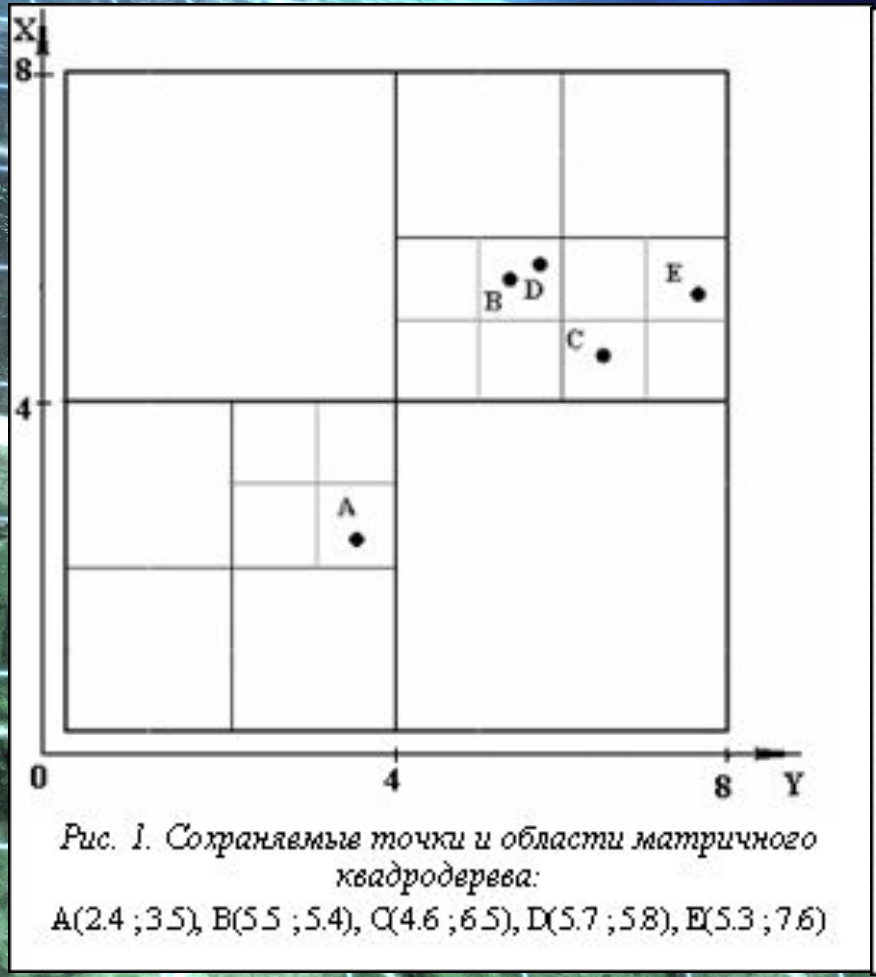


Рис. 1. Сохраняемые точки и области матричного квадродерева:

A(2.4 ; 3.5), B(5.5 ; 5.4), C(4.6 ; 6.5), D(5.7 ; 5.8), E(5.3 ; 7.6)

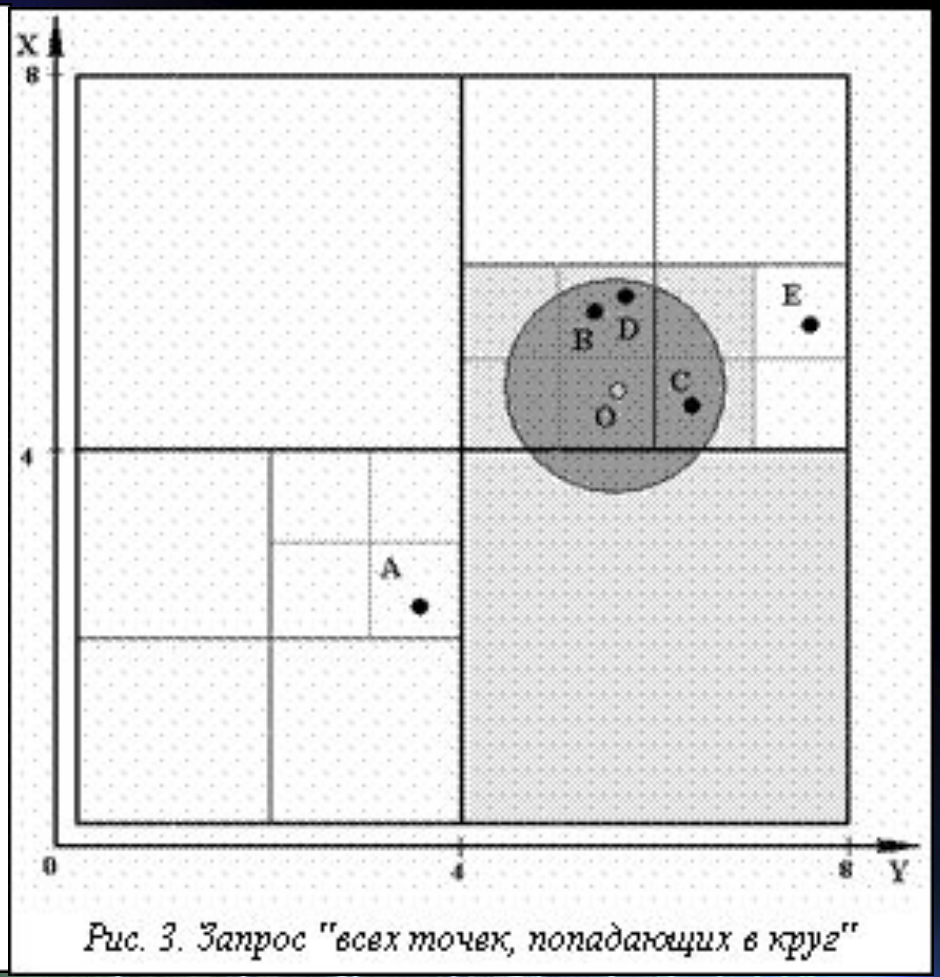
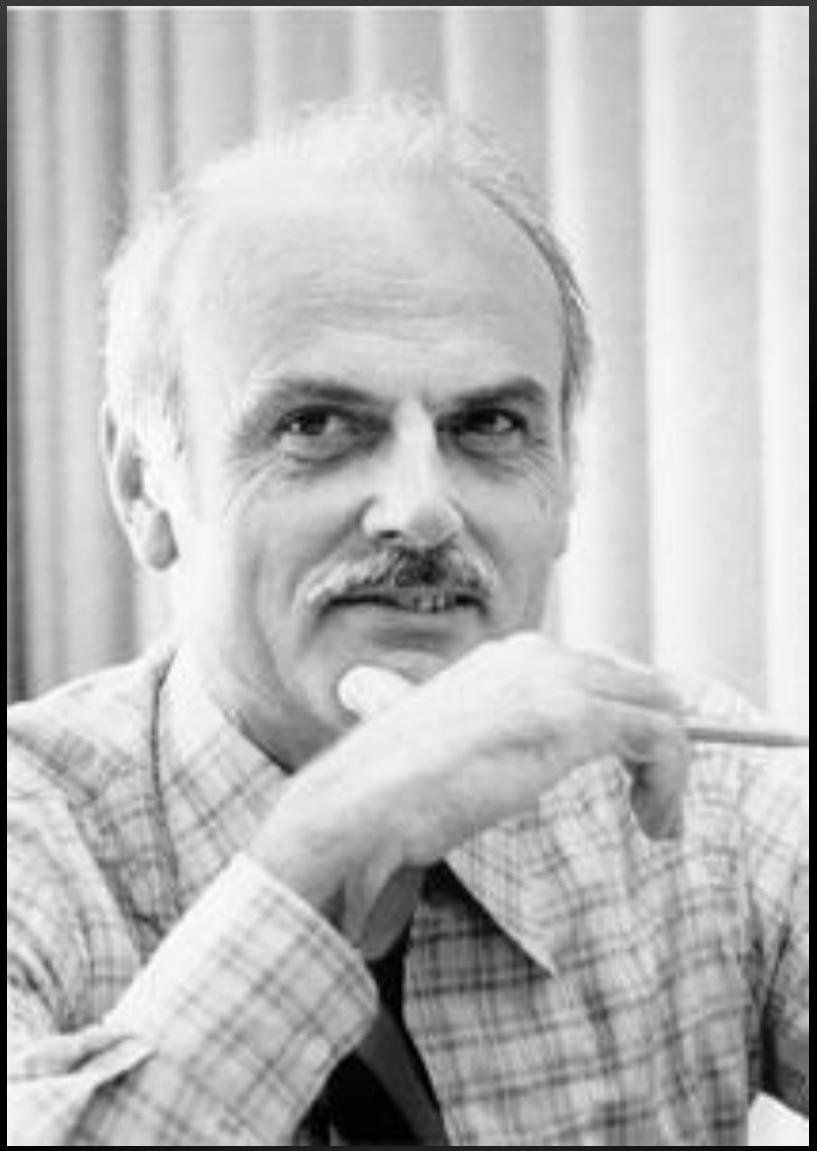


Рис. 3. Запрос "всех точек, попадающих в круг"

4. Реляционная модель данных

Реляционная модель данных – логическая модель данных. Представляет интерес как наиболее математически проработанная.

Впервые была предложена Эдгаром Коддом, известным исследователем в области баз данных, в 1969 году, когда он был сотрудником фирмы IBM. Впервые основные концепции этой модели были опубликованы в 1970.



ЭДГАР КОДД

(1923-2003)

Реляционная модель данных представляет собой хранилище данных, организованных в виде двумерных таблиц.

С таблицами знакомы все, являются основным элементом баз данных. Однако таблицы это лишь внешнее отражение сложных внутренних структур БД. Говоря точнее, таблица – результат вывода данных на экран, на принтер. Почти всегда таблица – результат поиска, отбора. На экране мы видим не все данные, а лишь их часть в удобной табличной форме.

Основными понятиями реляционной модели являются:

атрибут

кортеж

отношение

Атрибуты

Это самые простые элементы структуры таблицы. В таблице мы их видим как названия столбцов.

Атрибуты по сути это множество имён столбцов. **Множество именно в математическом смысле. То есть, во-первых, *уникальное*, во-вторых, *неупорядоченное*.**

Уникальность **атрибутов обеспечивается именованием. Система должна следить за тем, чтобы не было двух одинаковых. Поскольку таблиц много, то обычно спереди добавляется приставка - имя таблицы.**

Неупорядоченность **никак специально не обеспечивается. Обычно в реальности атрибуты хранятся в том порядке, в каком были созданы. Однако любой другой порядок также имеет право на существование, поэтому уместно относиться к набору атрибутов именно как ко множеству.**

Атрибуты различаются по типам.

Наиболее известные из них:

- **числовой;**
- **текстовой;**
- **логический.**

Есть и другие типы, в том числе и производные.

Тип должен соблюдаться для всех значений атрибута.

Например: в таблице-каталоге скважин могут быть следующие атрибуты:

- ✔ **Номер/индекс скважины ID;**
- ✔ **Координаты X и Y;**
- ✔ **Высота Z;**
- ✔ **Глубина H и др.**

Пример атрибутов в реляционной модели

	Координаты, мм	
№ сваи	X	Y
1	24261,8	66320,3
2	28979,1	58158,7
3	28979,1	59758,7
4	29121,1	56574,8
5	29257,8	61322,6
6	29681,8	55079
7	29903,8	62783,6
8	30399,2	58608,3
9	30519,4	53718,6
10	30565	60207,8
11	30617	57019,6
12	30818,2	64093,5
13	31100,6	61711,5
14	31201,4	55534,3
15	31584,4	52526,2

Кортежи

Это аналоги строк в таблице.

Каждый кортеж содержит несколько элементов по числу атрибутов таблицы, каждый элемент – одно значение, соответствующее одному атрибуту.

Для разных атрибутов, разумеется, будут разные типы данных, но для одного и того же атрибута тип строго соблюдается в разных кортежах таблицы.

Итак, кортеж – набор значений, но не просто обособленных, а значений, для каждого из которых известно, какому столбцу они принадлежат, какому атрибуту. Поэтому удобно считать, что кортежи содержат пары – имя атрибута и значение.

В примере для каталога скважин можно записать строку-кортеж так:

ID	*X*	*Y*	*Z*	*H*
151K	*7541203*	*5724619*	*31,6*	*17,5*

Основные типы данных в реляционной модели те же, что и в программировании:

- **целочисленный INTEGER;**
- **дробночисленный (с плавающей точкой) FLOAT;**
- **текстовой (символьный) различной длины CHAR, VARCHAR;**
- **логический (да/нет) LOGICAL;**
- **временной (дата/время) (DATE/TIME).**

Однако любых математических типов будет недостаточно, чтобы построить целостную базу данных и избежать несоответствий. Например, координаты XY в системе Гаусса-Крюгера должны быть миллионы метров – не меньше и не больше. Высота Z не может быть выше 10 км.

Это помогает не только отсекать возможные ошибки, но и заранее сузить область определения, задать ей практичные рамки. Такое пользовательское описание данных очень близко к понятию *домена*.

Домен это потенциально возможное множество значений. **Domain** в переводе означает «область», здесь смысл не расходится с переводом.

Домен является **множеством**, хотя в общем случае его значения нельзя просто перечислить. Зато всегда можно понять, в домене данное значение или нет.

Домен имеет **границу**, данные делятся на возможные и невозможные. Как и для множества, это не означает, что количество элементов конечное. Вышесказанное характеризует такое свойство домена как **ограниченность**.

Второе свойство домена – **уникальность**. Можно сравнить одни элементы с другими и избежать дубликатов. Для одного отдельного домена это само собой разумеется.

Первичный ключ (primary key)

Это очень важное понятие, можно сказать «ключевое». Теоретически это набор значений, который однозначно идентифицирует данный кортеж. Точнее сказать, набор атрибутов отношения, минимально необходимый для идентификации.

Первичный ключ может быть *простой* – из одной колонки, и составной – из нескольких.

Первичный ключ составляет стержень таблицы, и любая СУБД имеет технические средства для его реализации. После назначения колонок первичным ключом уникальность по нему отслеживается автоматически. Система не позволит создать две строки с одинаковыми значениями первичного ключа, например, вписать еще один замер в той же скважине за ту же дату

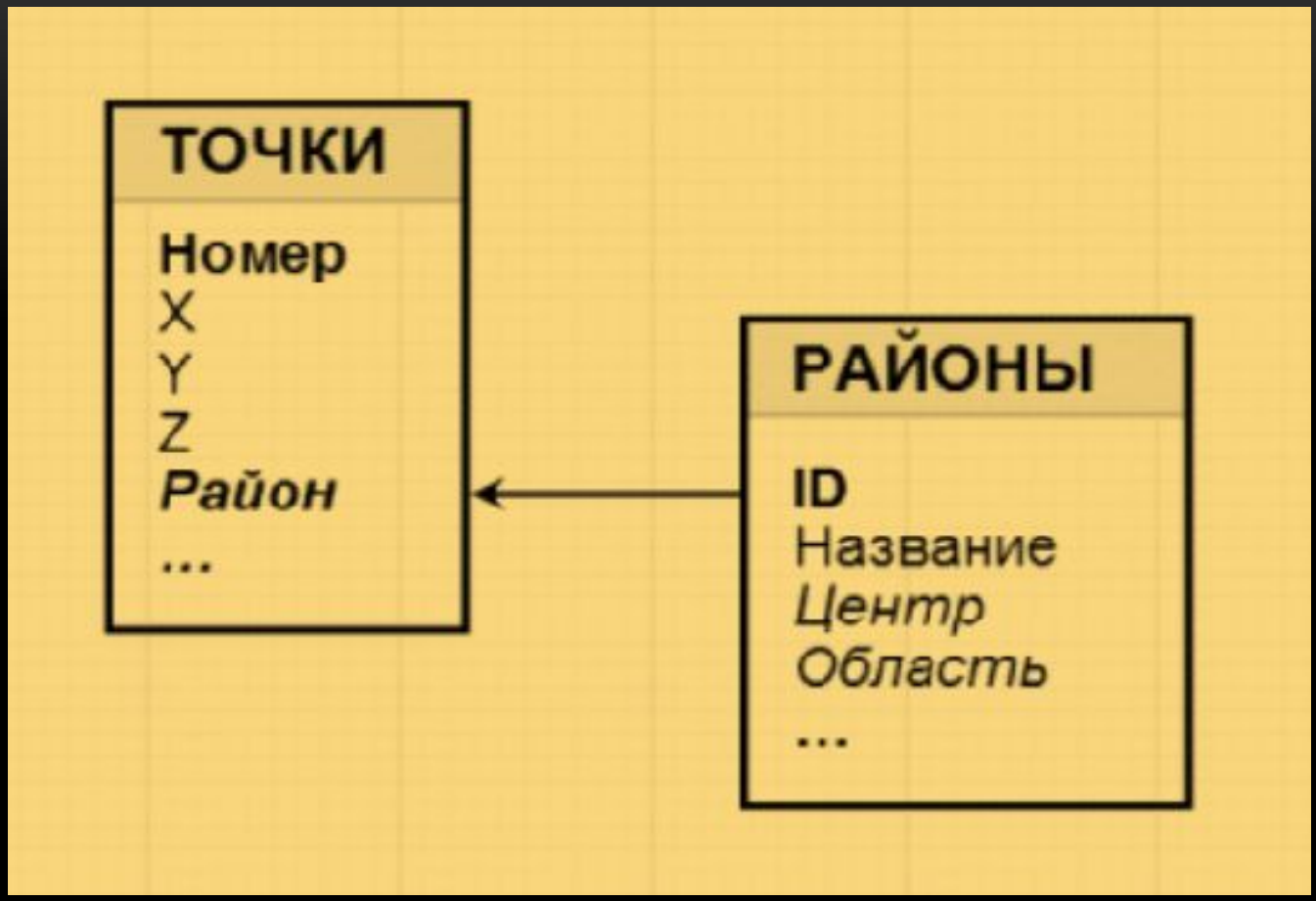
Пример: для таблицы-каталога скважин первичный ключ – номер (индекс) скважины, для таблицы замеров грунтовых вод первичный ключ должен состоять из номера скважины и даты замера. Для одной и той же скважины в разных строках дата должна быть разной, и наоборот.

Уровни грунтовых вод		
Скважина	***Дата замера***	***Уровень***
151к	***01.06.1999***	***13.1***
151к	***08.06.1999***	***14.2***
119	***01.06.1999***	***6.4***
119	***05.06.1999***	***8.3***

Внешний ключ (Foreign key)

Служит для связи таблиц. Это значения из одной таблицы, по которым можно однозначно привязаться к другой. Точнее говоря, для отношения внешний ключ - это опять-таки набор определенных заранее атрибутов.

Пример: в таблице точек наблюдений может быть атрибут «Административный Район», где для каждой точки проставлен код района, которому она принадлежит. Имеется таблица-справочник административных районов, в которой каждый район описан отдельной строкой. Для каждой точки по коду района можно найти его название и другие характеристики. Можно вообще соединить две таблицы в одну по этим ключам. Говорят, что атрибут «Район» – внешний ключ, ссылающийся на другую таблицу. Колонка ID в той, второй таблице, должна быть обязательно первичным ключом, иначе могут случайно сыскаться два одинаковых кода района в разных строках и система даст сбой, не сумеет однозначно привязаться.



Внешний ключ

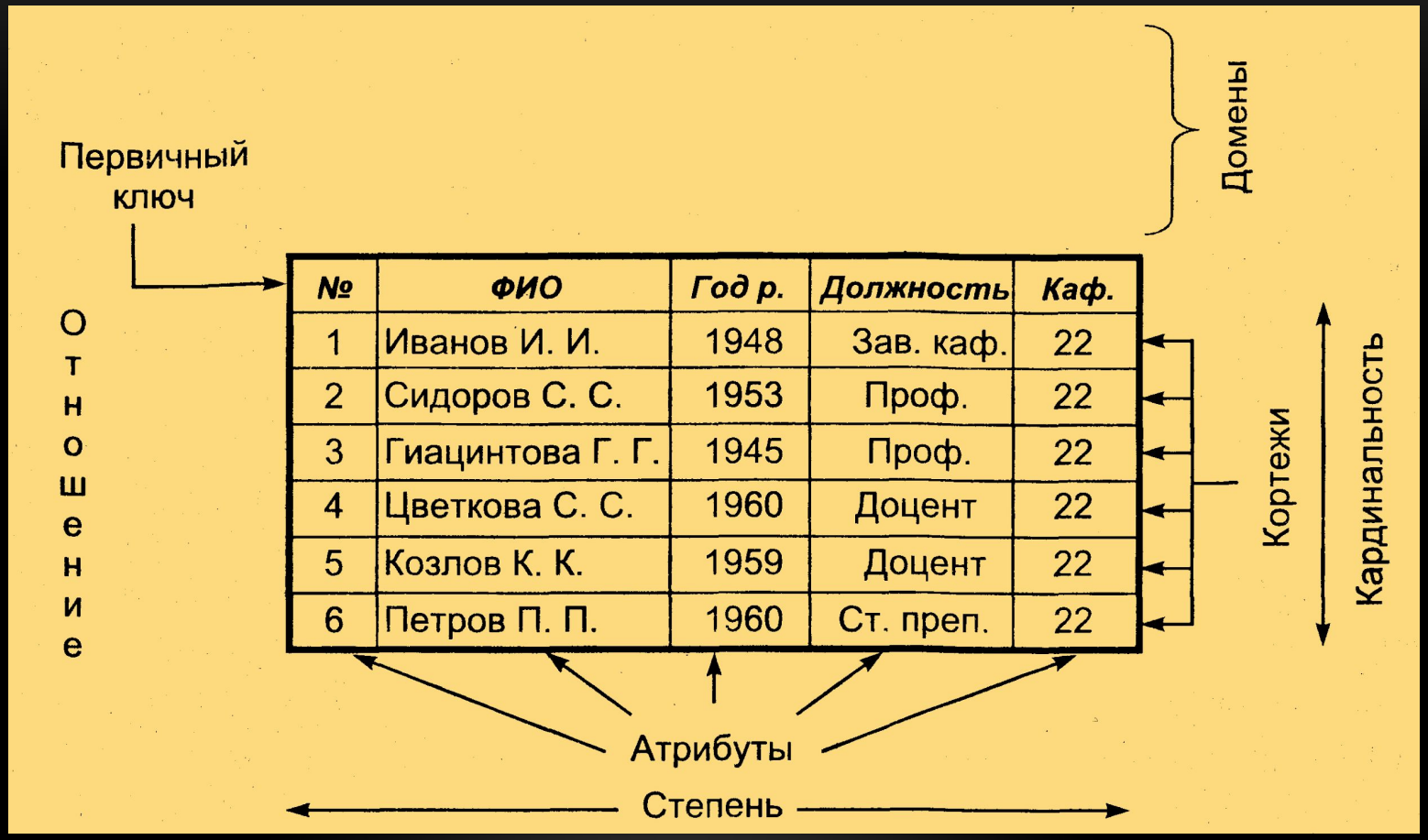
Внешний ключ должен ссылаться на первичный ключ другой таблицы. В своей таблице он может быть обычным атрибутом, а может входить в состав первичного ключа, это заранее не известно.

Например, если в таблице точек нумерация не сквозная по области, а порайонная, то атрибут «Район» логичным образом войдет в первичный ключ. К его внешней функции это не будет иметь прямого отношения.

Понятно, что первичный ключ может быть составным, а внешний ключ? Тоже может.

Например, коды административных районов не обязаны быть уникальными, и могут повторяться в разных областях (субъектах федерации). Тогда первичный ключ таблицы районов будет «Код района» и «Код области», и на диаграмме связей мы увидим между таблицами две линии. То же может быть и с номерами скважин, выполненных разными субподрядчиками: словом, система ключей достаточно гибкая, чтобы отражать любые варианты идентификации объектов предметной области.

ОСНОВНЫЕ КОМПОНЕНТЫ РЕЛЯЦИОННЫХ МОДЕЛЕЙ ДАННЫХ ГИС



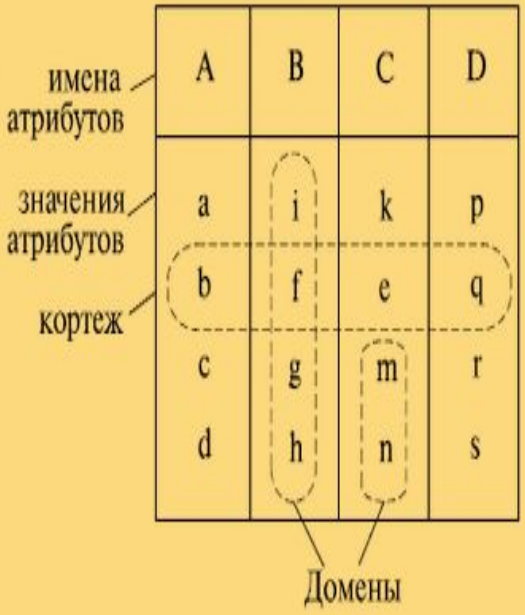
ОСНОВНЫЕ МОДЕЛИ РЕЛЯЦИОННЫХ ДАННЫХ ГИС

«Транзисторы»

Тип транзистора	β	$I_{k\,max}$, А	P_k, Вт	C_k, пФ
ТК903А	4,0	3	30	180
КТ904А	3,5	0,8	5	72
КТ907Б	3,0	1,0	13,5	20
КТ911А	5,2	0,4	3	10
...	...	...	...	...

а)

«Имя отношения»



б)

Выделяют три составные части реляционной модели данных:

- структурную
- манипуляционную
- целостную

Структурная часть модели

Определяет, что единственной структурой данных является нормализованное парное отношение.

Отношения удобно представлять в форме таблиц, где каждая строка есть кортеж, а каждый столбец – атрибут, определенный на некотором домене.

Манипуляционная часть модели

Определяет два фундаментальных механизма манипулирования данными – реляционная алгебра и реляционное исчисление.

Основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка реляционных БД: язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

Целостная часть модели

определяет требования целостности сущностей и целостности ссылок.

Первое требование состоит в том, что любой кортеж любого отношения отличим от любого другого кортежа этого отношения, т.е. другими словами, любое отношение должно обладать первичным ключом.

Требование целостности по ссылкам, или требование внешнего ключа состоит в том, что для каждого значения внешнего ключа, появляющегося в ссылающемся отношении, в отношении, на которое ведет ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным (т.е. ни на что не указывать).

Недостатки реляционной модели

Строгость структур страдает негибкостью. Реляционная модель волей-неволей задает строгую однотипность объектов в таблице, тогда как в реальности все они разные. Пропущенные значения помогают примириться с обязательным набором атрибутов, но постоянно стоит дилемма – вынести объекты другого типа за рамки или рамки расширить?

Сильная зависимость структур от данных. Возможности работы реляционными способами, особенно структуры данных, сильно зависят от состава данных, от их статистики.

Несвязность кортежей оборачивается неэффективностью для простых массивов. При наличии в предметной области тесных связей и упорядоченностей объектов становятся бессмысленными и БД, и вообще табличная форма.