

Математическая модель OLAP-кубов

Кузнецов Сергей Дмитриевич, ИСП РАН, kuzloc@ispras.ru
Кудрявцев Юрий Александрович, ВМиК МГУ, mail@y kud.com

Апрель 2008

Аннотация

В 1993 году Э. Коддом была предложена концепция OLAP-систем (Online Analytical Processing), включающая в себя 12 правил представления данных пользователю. Подобные системы, как следует из названия, предназначены для анализа данных в интерактивном режиме. В связи с этим основной задачей OLAP-средств является представление больших объемов данных в виде, удобном для анализа конечными пользователями. Представление данных в виде многомерных кубов на сегодняшний день является de facto стандартом пользовательской работы с большими массивами данных.

В данной статье вводятся основные понятия OLAP-систем, которые затем формализуются с использованием математического аппарата теории решеток. В рамках введенной формализации доказывается оптимальность (с точки зрения объема хранимых элементов) представления OLAP-кубов замкнутыми решетками или эквивалентными им Quotient-решетками.

Статья содержит следующие разделы:

- OLAP. Базовые понятия и терминология. Вводятся определения OLAP-кубов, отмечаются их основные требования и свойства.
- Некоторые определения теории решеток. В данном разделе приводятся необходимые в дальнейшем определения теории решеток.
- Математическая модель OLAP-данных. Вводится формальная модель OLAP-кубов, доказывается, что подобное представление является решеткой, и оптимальное с точки зрения хранения представление OLAP-кубов – замкнутые решетки.
- Выводы и направления дальнейших исследований.

1 OLAP. Базовые понятия и терминология.

Термин OLAP (Online Analytical Processing) был введен в 1993 Эдгаром Коддом [Cod93]. Цель OLAP систем – облегчение решения задач анализа данных. Кодд сформулировал 12 признаков OLAP-данных, и большинство современных OLAP средств отвечают этим постулатам. Однако 12 признаков в дальнейшем трансформировались в 4 ключевых определения, сформулированные Найджелом Пендзом (см. [Pen05]), на которые теперь ссылаются при определении OLAP-систем.

FASMI-тест. OLAP-система должна быть:

- Fast – быстрой, обеспечивать почти мгновенный отклик на большинство запросов

- Shared – многопользовательской, должен существовать механизм контроля доступа к данным, а также возможность одновременной работы многих пользователей
- Multidimensional – многомерной, данные должны представляться в виде многомерных кубов.
- Information – данные должны быть полны с точки зрения аналитика, т.е. содержать всю необходимую информацию.

Большинство существующих OLAP-средств удовлетворяют всем этим признакам. Однако в реализации подобных приложений возникает ряд проблем, прежде всего связанных с увеличением объема данных, которые необходимо хранить.

В 1995 группа исследователей во главе с Джимом Греем [GBLP95], проанализировав создававшиеся тогда пользовательские приложения баз данных, предложила расширение языка SQL – оператор CUBE. Этот оператор отвечает в SQL за создание многомерных кубов. Концепция многомерного представления данных является, наряду с моделью транзакций, одной из самых известных идей Кодда. В этой работе исследователи указали ряд эвристических рекомендаций по реализации новой структуры данных.

CUBE представляет собой обобщение операторов GROUP BY по всем возможным комбинациям измерений с разными уровнями агрегации данных. Каждая сгруппированная таблица относится к группе ячеек, описываемых кортежами из измерений, по которым формируется куб. Оператор, расширяющий SQL, называется CUBE BY (синтаксис такой же, как и у GROUP BY).

В стандарт SQL'99 был включен набор операторов для работы с OLAP-данными (запросы grouping set, rollup by, cube by, window by, rank, rownum и пр.).

2 Многомерные кубы, определение и свойства

Рассмотрим базовую (фактическую) таблицу r , на основе которой будет строиться OLAP-куб. Множество атрибутов r условно делят на 2 группы:

1. Набор измерений (категорий, локаторов), которые служат критериями для анализа и определяют многомерное пространство OLAP-куба. За счет фиксации значений измерений получают срезы (гиперплоскости) куба. Каждый срез представляет собой запрос к данным, включающий агрегации.
2. Набор мер – функции, которые каждой точке пространства ставят в соответствие данные.

Из атрибутов r создаются измерения, содержащие проекцию r по атрибуту, с введенной иерархией (например, для таблицы, в которой хранятся фактические данные по продажам магазина, возможно наличие измерение под названием "Время содержащего иерархию вида "Год-Месяц-Неделя-День"). Куб представляет собой декартово произведение измерений, где для каждого элемента произведения назначен набор мер. В кубе введены отношения обобщения и специализации (roll-up/drill-down) по иерархиям измерений (подробнее об иерархиях см. 2.3). Ячейка высокого уровня иерархии может "спускаться"(drill-down) к ячейке низкого уровня (для примера 2.1 (R1,ALL,весна) может "спуститься"к ячейке (R1,книги,весна)) и наоборот, "подняться"(roll-up) (от (R1,книги,весна) к (R1,ALL,весна) по измерению "продукты").

Таблица 1: Фактические данные для примера

Регион	Продукт	Время года	Продажи
R1	книги	Весна	9
R1	Еда	Осень	3
R2	книги	Осень	6

Таблица 2: Куб для таблицы 1. Агрегирующая функция - AVG.

Регион	Продукт	Время года	AVG(Продажи)
R1	книги	Весна	9
R1	Еда	Осень	3
R2	книги	Осень	6
R1	книги	ALL	9
R1	ALL	Весна	9
ALL	книги	Весна	9
...
R2	ALL	ALL	6
ALL	Еда	ALL	3
ALL	ALL	Весна	9
ALL	ALL	ALL	6

2.1 Пример

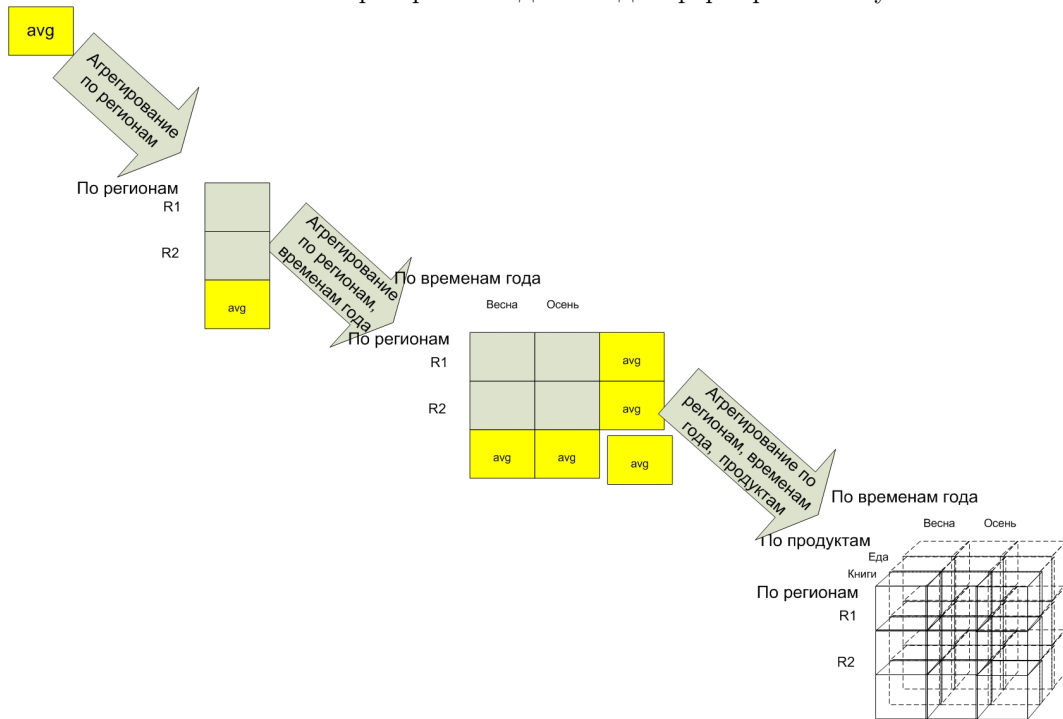
Рассмотрим пример, который будет в дальнейшем использоваться в этой статье.

Размер куба данных определяется по формуле $\prod_d (c_i + 1)$, где d - количество измерений ("столбцов"), c_i - размерность измерения, т.е. количество различных значений кортежей по этому измерению. Эквивалентный sql-запрос: `SelectCount(distinctdimension)fromcube_t(able)`, +1 отвечает за "значение" *ALL*, агрегирующее все возможные значения измерения.

2.2 Измерения

Измерения куба – это набор доменов, по которым создается многомерное пространство. Важной особенностью OLAP-моделей является разделение измерений на локаторы (задающие точки) и меры (задающие значение). Как отмечено в [Tho02], данное разделение может носить как условный, так и жесткий характер. В случае условного разделения измерения можно "разворачивать" как данные и как аналитики, создавая новую аналитику куба по продажам – "количество продаж". Таким образом, возрастает гибкость моделей и уровень абстракции. Однако этот подход, несмотря на свою привлекательность, сложен в реализации (в частности, отметим необходимость создания оптимальных алгоритмов хранения абстрактных типов данных) и, насколько нам известно, нигде промышленно не реализован. Теоретически, вкупе с моделированием решеток кубов логикой преди-

Рис. 1: Схема агрегирования данных для формирования куба



катов первого порядка, абстрагирование понятия "измерения" дает очень интересные результаты.

Локаторы куба отличаются иерархической структурой, и для получения значений мер на каждом уровне агрегирования вводятся агрегирующие функции.

2.3 Иерархии и агрегирование

Иерархичность данных – одно из важнейших свойств многомерных кубов. Иерархии призваны добавлять новые уровни в аналитическое пространство пользователя. Самым распространенным примером иерархии является "день–неделя–месяц–год". Между элементами разных уровней иерархии существуют отношения обобщения и специализации (rollup/drilldown).

Все иерархии можно разбить на 2 типа, о которых пойдет речь ниже. Основой разбиения будет служить расстояние d от корня ($\{ALL, ALL, \dots, ALL\}$) до листьев. В случае, если $d = const$, – иерархии называются *уровневыми (leveled)*, иначе – *несбалансированными (ragged)*.

Примеры типов иерархий:

Уровневые: день–месяц–год; улица – город – страна.

Несбалансированные: Организационная диаграмма, различная группировка продуктов.

2.4 Агрегирующие функции, меры и формулы

Неотъемлемой частью OLAP-модели является задание функций агрегирования. Поскольку цель OLAP – создание многоуровневой модели анализа, данные на уровнях, отличных от фактического, должны быть соответствующим образом агрегированы. Важно отметить, что по каждому измерению можно задавать собственную (и не одну) функцию агрегации. Таким образом, в случае куба с n измерениями функция агрегирования - это:

$$f(x) = [\{f_{1,1}, \dots, f_{1,k_1}\}, \dots, \{f_{n,1}, \dots, f_{n,k_n}\}]$$

где x – точка куба, а $f_{i,j}$ - j -ая функция агрегирования по i -ому измерению.

В [GC97] приведена следующая классификация агрегирующих функций с точки зрения сложности распараллеливания.

Таблица 3: Категории агрегирующих функций

Категория	Примеры
Дистрибутивные	Sum(), Count(), Minimum(), Maximum()
Алгебраические	Average(), Standard_Deviation(), Center_of_Mass(), MaxN(), MinN()
Холистические	Median(), Most_Frequent(), Rank()

Дистрибутивные функции позволяют разбивать входные данные и вычислять отдельные итоги, которые потом можно объединять.

Алгебраические функции возможно представить в виде комбинации из дистрибутивных функций (например, Average() можно представить, как $\frac{sum()}{count()}$)

Холистические функции невозможно вычислять на частичных данных или представлять каким-либо образом.

3 Некоторые определения из теории решеток

Приведем некоторые необходимые в дальнейшем изложении определения из теории решеток. Определения приводятся по книге [Г.81].

3.1 Частично-упорядоченное множество, решетка

Определение 1. Частично-упорядоченным множеством (чум) называется множество, на котором определено бинарное отношение \leq , удовлетворяющее для всех x, y, z следующим правилам:

1. $x \leq x$ (рефлексивность)
2. если $x \leq y$ и $y \leq x$, то $x = y$ (антисимметричность)
3. если $x \leq y$ и $y \leq z$, то $x \leq z$ (транзитивность)

Если для чум (A, \leq) верно, что $\forall x, y \in A : x \leq y$ или $y \leq x$, то такое множество называется линейно-упорядоченным множеством, или *цепью*.

Пусть $H \subseteq P$ и $a \in P$. Тогда a называется *верхней границей* подмножества H , если $h \leq a$ для всех $h \in H$. Верхняя граница a подмножества H называется его верхней гранью или *супремумом*, если $a \leq b$ для любой верхней границы b подмножества H . Будем писать $a = \sup H$ или $a = \vee H$. Понятие *нижней границы* и *нижней грани*, или *инфинума*, вводятся аналогично; инфинум обозначается $\inf H$ или $\wedge H$.

Определение 2. Чум (A, \leq) называется **решеткой**, если $\forall x, y \in L$ существуют $\sup(x, y)$ и $\inf(x, y)$.

Мы будем использовать обозначения

$$a \wedge b = \inf\{a, b\}$$

и

$$a \vee b = \sup\{a, b\}$$

и называть \wedge – *пересечением*, а \vee – *объединением*. В теории решеток они называются *бинарными операциями*, являющимися отображениями $A^2 \rightarrow A$.

Операции \wedge и \vee обладают следующими свойствами.

1. Идемпотентность: $a \wedge a = a$, $a \vee a = a$.
2. Коммутативность: $a \wedge b = b \wedge a$ и $a \vee b = b \vee a$
3. Ассоциативность: $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ и $(a \vee b) \vee c = a \vee (b \vee c)$

3.2 Описание решеток. Изоморфизм решеток. Оператор замыкания.

Существует несколько способов описания решеток, но мы остановимся на описании частичного порядка, поскольку данный способ достаточно просто модифицируется для анализа OLAP-кубов. Для примеров в этом подразделе будем использовать решетку $T_l = (\{0, a, b, 1\}, \leq) : 0 \leq a \leq 1, 0 \leq b \leq 1$.

Определение 3. Будем говорить, что в чум (A, \leq) , элемент a покрывает b или b покрывается элементом a (обозначение: $a \succ b$ или $b \prec a$ соответственно), если:

- $a \succ b$ и
- не существует элемент x , такой, что $a \succ x \succ b$

Следующая лемма показывает, что отношение покрытия будет определять частичный порядок на множестве.

Лемма 1. Если (A, \leq) – конечное чум, то $a \leq b$ тогда и только тогда, когда $a = b$ или когда существует конечная последовательность элементов x_0, \dots, x_{n-1} , такая, что $x_0 = a$, $x_{n-1} = b$ и $x_i \prec x_{i+1}$, для $0 \leq i < n - 1$.

Отношение покрытия в решетке T_l будет выглядеть следующим образом:

$$\prec = \{(0, a), (0, b), (a, 1), (b, 1)\}$$

Определение 4. Решетки $A_l = (A, \leq)$ и $B_l = (B, \leq)$, называются изоморфными ($A_l \cong B_l$), если существует взаимно однозначное отображение $\theta : A_l \rightarrow B_l$, называемое изоморфизмом, такое что

$$a \leq b \text{ в } A \iff \theta a \leq \theta b \text{ в } B$$

В конечном чум A с 0 ($\forall x \in A : x \geq 0$), высотой или размерностью $h[x]$ элемента x называется точная верхняя грань длин цепей $0 = x_0 < x_1 < \dots < x_l = x$ между 0 и x . $h[x] = 1$ тогда и только, когда x покрывает 0 , - такие элементы называются *атомами* или *точками* множества A .

Градуированным чум называется чум A с заданной на нем функцией $g : A \rightarrow Z$, принимающей значения в цепи всех целых чисел (с их естественной упорядоченностью) и такой, что

- если $x > y$, то $g[x] > g[y]$ (строгая изотонность)
- если x покрывает y , то $g[x] = g[y] + 1$

Для решетки (A, \geq) оператор $C : A \rightarrow A$ называется *оператором замыкания* если:

- $x \leq C(x), x \in A$ (экстенсивность)
- $x \leq y \Rightarrow C(x) \leq C(y), x, y \in A$ (монотонность)
- $C(C(x)) = C(x), x \in A$ (идемпотентность)

4 Математическая модель OLAP-кубов

4.1 Общие определения. Меры, измерения, операторы в многомерном пространстве

В этом разделе используется ряд определений из работ [Cas04], [NCCL07] и других работ этого коллектива авторов.

Пусть r отношение базы данных над схемой R . Атрибуты R разделяются на 2 группы:

1. **D** - множество измерений. $(d_1, d_2, \dots, d_{|D|}), d_i \in D[i]$ - точка в многомерном пространстве.
2. **M** - множество мер. Для каждой $(d_1, d_2, \dots, d_{|D|})$ - точки, существует $(m_1, m_2, \dots, m_{|M|})$ - множество значений мер в этой точке.

$\forall A \in D, r[A]$ означает проекцию r по A . $\forall a \in r[A] : a \in ALL$

Определение 5. Многомерное пространство $Space(r) = \{\times_{A \in D} (r[A] \cup ALL) \cup \{0, 0, \dots, 0\}\}$, где \times - декартово произведение, $\{0, 0, \dots, 0\}$ - нулевой кортеж.

$\forall s \in Space(r), s$ - многомерный кортеж. Для куба из примера (см. таблицу 1) получаем следующее многомерное пространство:

$\{(R1, книги, весна), (R1, еда, осень), (R2, книги, осень), (R1, еда, весна), \dots, (ALL, еда, осень), (R1, ALL, весна), \dots, (ALL, ALL, ALL)\}$ - всего 28 кортежей.

Введем отношение обобщения/специализации на $Space(r)$ обозначаемое \geq_g .

Определение 6. Отношение порядка $\geq_g u, v \in Space(r)$

$$u \geq_g v = \begin{cases} \forall A \in D, v[A] \subseteq u[A] \\ \text{в противном случае } v = \{0, 0, \dots, 0\} \end{cases}$$

Если $u, v \in Space(r)$, $u \geq_g v$ тогда u обобщает v в $Space(r)$.

Для куба из примера (см. таблицу 1): $(ALL, еда, осень) \geq_g (R1, еда, осень)$, $(ALL, ALL, ALL) \geq_g (ALL, еда, осень)$.

4.2 Операторы в $Space(r)$

Определение 7. Операторы минимума, максимума в $Space(r)$.

$T \subseteq Space(r)$ - подмножество $Space(r)$.

$$\min_{\geq_g}(T) = \{t \in T \mid \neg \exists u \in T : t \geq_g u\}$$

$$\max_{\geq_g}(T) = \{t \in T \mid \neg \exists u \in T : u \geq_g t\}$$

Операторы, создающие новые кортежи: сумма (+) и произведение (\bullet).

Определение 8. Сумма двух кортежей – наименьший кортеж, обобщающий оба операнда.

$u, v \in Space(r)$

$$t = u + v \Leftrightarrow \forall A \in D, t[A] = \begin{cases} u[A], \text{ if } u[A] = v[A] \\ ALL, \text{ в противном случае} \end{cases}$$

Для примера из таблицы 1, $(ALL, еда, осень) + (R1, ALL, осень) = (R1, еда, осень)$

Определение 9. Произведение двух кортежей – наибольший кортеж, уточняющий оба операнда.

Пусть $\forall A \in D, z[A] = u[A] \wedge v[A]$. Тогда $u, v \in Space(r)$

$$t = u \bullet v \Leftrightarrow \begin{cases} t = z \text{ если } \neg \exists A \in D \mid z[A] = \{0\} \\ \{0, 0, \dots, 0\}, \text{ в противном случае} \end{cases}$$

Например, $(R2, еда, осень) \bullet (R1, ALL, осень) = (ALL, ALL, осень)$

Введя определения $Space(r)$, \geq_g и операторы + и \bullet мы можем дать определение решетки куба.

Определение 10. Решетка куба

Пусть r – отношение базы данных над $D \cup M$. Чум $CL(r) = \{Space(r), \geq_g\}$ – решетка, в которой пересечение и объединение вводятся следующим образом:

$$\forall T \subseteq CL(r), \wedge T = +_{t \subseteq T} t$$

$$\forall T \subseteq CL(r), \vee T = \bullet_{t \subseteq T} t$$

$\{CL(r), \geq_g, \wedge, \vee\}$ – решетка куба.

Определим мощность решетки куба. Пусть $L(r)$ множество подмножеств атрибутов, например $P(\bigvee_{A \in D} A.a, \forall a \in r[A])$, где $P(X)$ – множество подмножеств X .

Существует отображение $\Phi: CL(r) \rightarrow L(r)$

$$t \leftarrow \begin{cases} \bigvee_{A \in D} r[A], t = \{0, 0, \dots, 0\} \\ \{t[A] \mid \forall A \in D, t[A] \neq ALL\} \text{ иначе} \end{cases}$$

Ранг кортежа t ($rank(t)$) – длина минимального пути в решетке, соединяющего $\{ALL, ALL, \dots, ALL\}$ и t .

$$\text{Тогда } rank(t) = \begin{cases} |\Phi(t)| \text{ если } t \neq \{0, 0, \dots, 0\} \\ |D| + 1 \text{ в противном случае} \end{cases}$$

Например, $rank((R1, ALL, ALL))=1$, $rank((R1, еда, весна))=3$.

Таким образом, решетка куба – градуированная решетка. Число уровней в решетке $|D| + 1$.

Число элементов решетки на уровне i , ($i \in 1 \dots |D|$):

$$\sum_{X \subseteq D, |X|=i} \left(\prod_{A \in X} |r[A]| \right) \leq \binom{|D|}{i} \max_{A \in D} (|r[A]|)^i$$

Мощность решетки куба: $\left(\prod_{A \in D} (|r[A]| + 1) + 1 \right)$, где для каждого измерения добавляется значение

ALL (или количество уровней иерархии) и добавляется $\{0, \dots, 0\}$.

Поскольку большая часть ячеек многомерного пространства не меняет значения меры (избыточность) или не существует (разреженность), представляется разумным хранить только необходимые ячейки. Разобьем многомерное пространство на классы эквивалентности, чтобы для каждого из классов хранить только нижнюю и верхние грани, сократив таким образом объем хранимой информации.

Простейшим разбиением, является разбиение по значению меры ячеек, т.е. разбиение по отношению

$$R : R(x, y) \leftrightarrow M(x) = M(y), x, y \in A$$

Однако разбиение только по равенству значения меры не сохраняет отношения roll-up/drill-down. Покажем это.

Например, рассмотрим таблицу 1. Получившиеся разбиение изображено на рисунке 2. Ячейки одного класса эквивалентности выделены одним цветом. В качестве агрегирующей функции используется $Average()$.

У нас получается следующая схема классов эквивалентности (см. рисунок 3).

Выбор агрегирующей функции в данном случае не важен, т.к. у нас есть возможность двигаться в обе стороны по roll-up/drill-down отношениям. Например, мы идем вверх от ячейки (ALL, ALL, ALL) в C2 в ячейку (ALL, книги, ALL) в C5, а потом в (R2, книги, ALL) в C2. Нарушается семантика roll-up/drill-down отношений.

Таким образом, разбиение только по значению агрегирующей функции не порождает связанной решетки классов эквивалентности.

Необходимо другое отношение для порождения классов эквивалентности. Переформулируем отношение покрытия (см. определение 3) для решетки куба.

Определение 11. Отношение покрытия для решетки куба

Кортеж $c \in CL(r)$ покрывает базовый (фактический) кортеж $t \in r$, если $c \geq_g t$.

В нашем примере, (R1, ALL, ALL) покрывает $\{(R1, книги, весна), (R1, еда, осень)\}$.

Определение 12. Отношение эквивалентности по покрытию

Определим отношение эквивалентности \equiv_{cov} как транзитивное и рефлексивное замыкание R , где:

$$R : a, b \in CL(r)$$

$$R(a, b) \iff \exists T \in CL(r), \forall t \in T \begin{cases} t \in r \\ a \geq_g t \text{ и } b \geq_g t \\ \neg \exists t' \notin T : t' \in r \\ a \geq_g t' \text{ или } b \geq_g t' \end{cases}$$

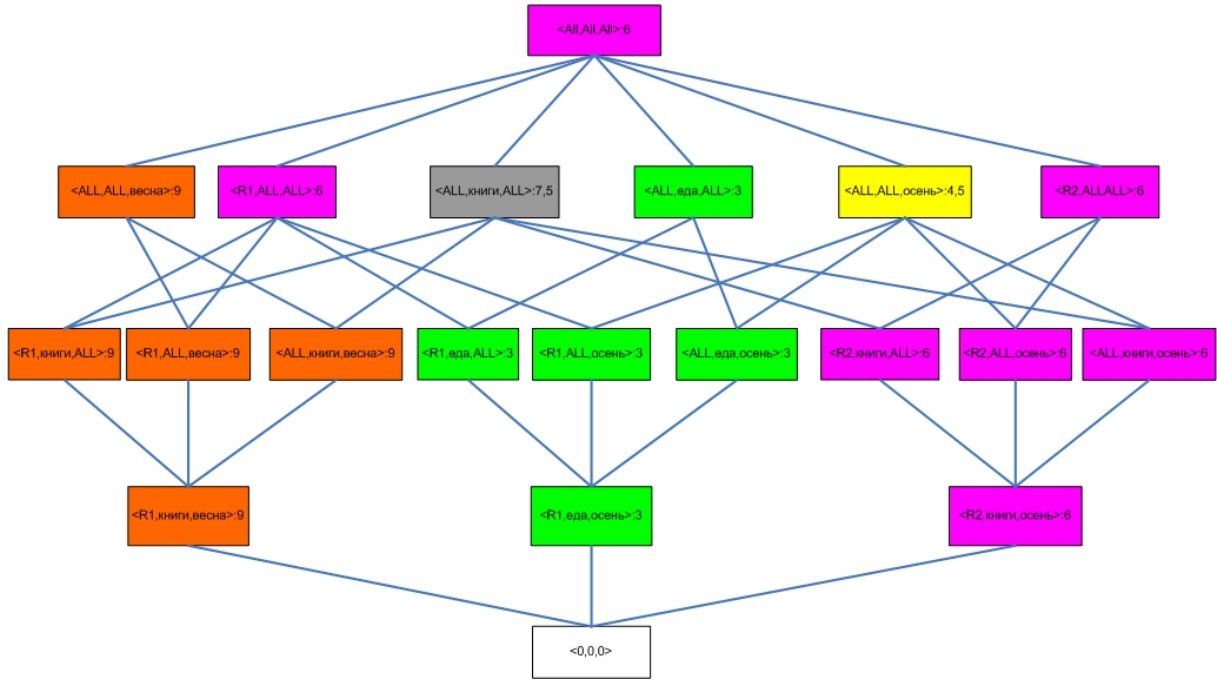


Рис. 2: Разбиение только по значению агрегирующей функции

Таким образом, решетка куба для алгоритма Quotient Cube ([Zha03]) определяется следующим образом:

Определение 13. Quotient решетка куба

Пусть $CL(r)$ – решетка куба (см. определение 10), и \equiv_{cov} – отношение эквивалентности (см. определение 12 на $Space(r)$). Quotient решетка куба $QCLR(r) = (CL(r) / \equiv_{cov}, \preceq)$. Для двух классов эквивалентности $A, B \in QCLR(r)$, $A \succeq B$ если $\exists a \in A, \exists b \in B | a \geq_g b$.

QCLR решетка для примера: рисунок 4.

И решетка классов эквивалентности: рисунок 5.

4.3 Замыкания и замкнутые решетки кубов

Используя ранее введенные определения, докажем что quotient решетка куба из алгоритма [Zha03] является минимальным (по количеству классов эквивалентности) представлением куба.

Определение 14. Оператор замыкания $C : CL(r) \rightarrow CL(r)$ – оператор замыкания.

$$t \rightarrow \begin{cases} +t \in_r t' & |t \geq_g t' \text{ если } \exists t' \in r \\ \{0, 0, \dots, 0\} & \text{иначе} \end{cases}$$

В нашем примере:

$$C((All, All, весна)) = (R1, книги, весна)$$

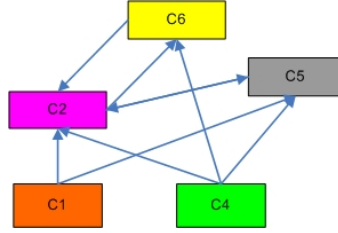


Рис. 3: Классы при разбиении только по значению агрегирующей функции

$$C((\text{All}, \text{книги}, \text{All})) = (\text{R2}, \text{книги}, \text{осень}) + (\text{R1}, \text{книги}, \text{весна}) = (\text{All}, \text{книги}, \text{All})$$

Минимальное замыкание куба уничтожает избыточность, удаляя кортежи, не меняющие значения меры. Вычислив замыкание каждого кортежа, определим *систему замыканий куба*.

Определение 15. Система замыканий куба $C(r) = \{t \in CL(r) | C(t, r) = t\}$ - *система замыканий с оператором C*.

Система замыканий в нашем примере: $\{(\text{R1}, \text{книги}, \text{весна}), (\text{R1}, \text{еда}, \text{осень}), (\text{R2}, \text{книги}, \text{осень}), (\text{R1}, \text{All}, \text{All}), (\text{All}, \text{книги}, \text{All}), (\text{All}, \text{All}, \text{осень}), (\text{All}, \text{All}, \text{All}), (0, 0, 0)\}$

Минимальные (с точки зрения \geq_g) кортежи, порождающие одинаковые замыкания называются *ключами куба*.

Определение 16. Ключи куба $Key(t) = \min_{\geq_g}(\{t' \in CL(r) | t' \geq_g t \text{ и } C(t', r) = t\})$ - *множество минимальных кортежей, создающий t. Каждый $t \in Key(t)$ - ключ куба.*

Замкнутая решетка куба - минимальное представление куба, не сохраняющее roll-up/drill-down отношение.

Определение 17. Замкнутая решетка куба

Чум $CCCL(r) = \{C(r), \geq_g\}$ - *замкнутая решетка куба.*

- $\forall T \subseteq CCCL(r), \vee T = +_{t \in T} t$
- $\forall T \subseteq CCCL(r), \wedge T = C(\bullet_{t \in T} t, r)$

Замкнутая решетка для куба из примера изображена на рисунке 6.

Для доказательства оптимальности (минимального количества классов эквивалентности) QCL мы построим решетку классов эквивалентности для замкнутой решетки куба и покажем изоморфизм получившейся решетки и QCL.

Для построения классов эквивалентности используем следующее отношение ϕ .

$$t\phi t' \longrightarrow C(t, r) = C(t', r)$$

Класс эквивалентности в таком случае: $[t] = \{t' \in CL(r) | t\phi t'\}$. Таким образом, $\max_{\geq_g}([t]) = C(t, r)$ и $\min_{\geq_g}([t]) = Key(t, r)$. множество классов эквивалентности вкупе с отношением порядка \geq_g называется решеткой классов эквивалентности для замкнутой решетки куба. Как и в QCL, каждый класс такой решетки имеет одну нижнюю грань и несколько верхних.

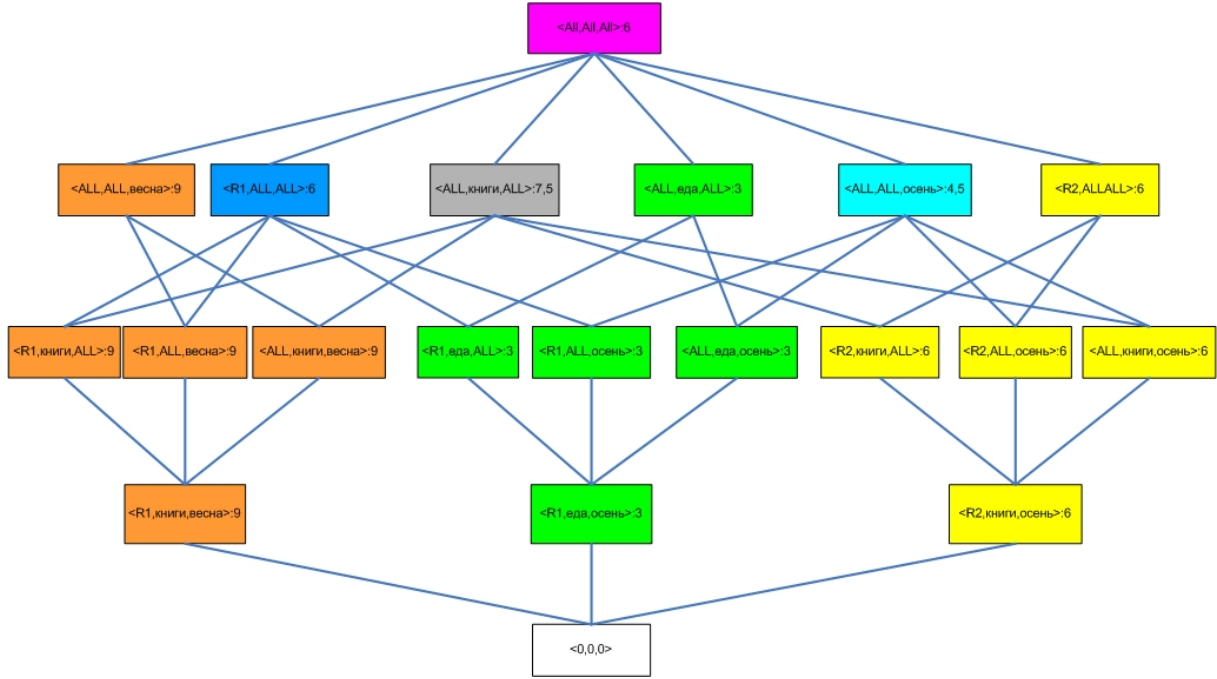


Рис. 4: Разбиение по покрытию

Определение 18. Решетка классов эквивалентности для замкнутой решетки куба $ECCL(r) = (C(r)_{\equiv_{\phi}}, \preceq)$, где для двух классов эквивалентности $A, B \in QCLR(r)$, $A \succeq B$ если $\exists a \in A, \exists b \in B | a \geq_g b$.

Решетка классов эквивалентности для примера изображена на рисунке 7.

Исходя из введенного определения замыкания решетки куба (см. определение 14) легко сформировать требуемый изоморфизм $\eta : QCL(r) \iff ECCL(r)$.

$$\eta : A \in QCL(r), \eta(A) \iff C(t)_{\phi} | t \in A$$

$$\eta^{-1} : B \in ECCL(r), \eta^{-1}(A) \iff cov(t)_{\equiv_{cov}} | t \in B$$

Таким образом, минимальным представлением решетки куба является замкнутая решетка куба, или эквивалентная ей quotient решетка куба.

5 Выводы и направление дальнейших исследований

В данной статье представлена математическая модель OLAP-данных, проведена связь между представленной моделью и теорией решеток, доказана оптимальность представления OLAP-кубов замкнутыми решетками и quotient решетками.

Дальнейшими направлениями исследований являются:

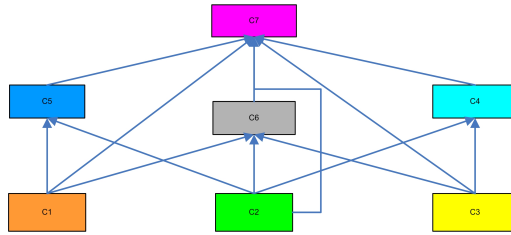


Рис. 5: Классы разбиения по покрытию

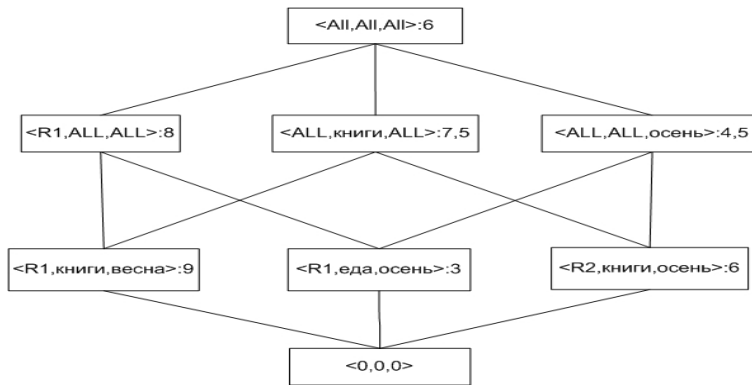


Рис. 6: Замкнутая решетка куба

- Развитие математической модели. Модель должна удовлетворять требованиям, перечисленным в работах [PJD01] и [Raf03]. Основными задачами в данном направлении являются:
 - Поддержка различных типов иерархий. Возможность задавать несбалансированные, неонто, нестрогие иерархии.
 - Вероятностные меры. Возможность вводить данные с некоторым уровнем точности, часто точное число неизвестно, и на основе этих данных получать корректные результаты запросов
 - Объединение данных различных уровней гранулярности. Данные могут быть представлены разных уровнях гранулированности, (например, продажи на уровне регионе, а не в конкретной кассе) В таком случае, данные должны корректно отображаться и позволять проводить анализ.
- Реализация параллельного алгоритма создания замкнутых решеток кубов. Представленная модель будет реализована в рамках открытого проекта MROLL (Map/Reduce OLAP Lattices) на базе кластера Apache/Hadoop, с учетом существующих работ по распараллеливанию обработки OLAP-кубов (см. [YJA03] и [GC97]).

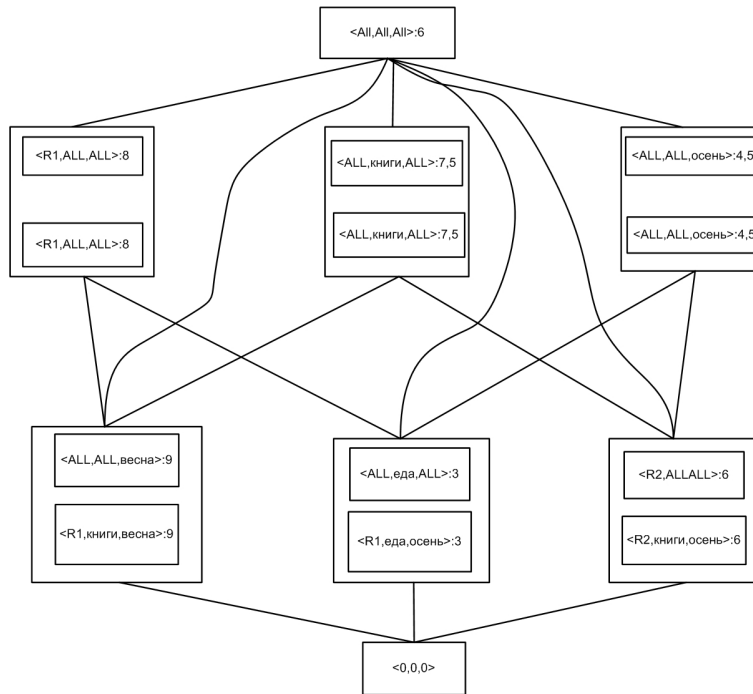


Рис. 7: Решетка классов эквивалентности для замкнутой решетки куба

Список литературы

- [Г.81] Грегцер Г. *Общая теория решеток*. Мир, 1981.
- [Cas04] Alain Casali. Mining borders of the difference of two datacubes. In *DaWaK*, 2004.
- [Cod93] E.F. Codd. Providing OLAP for end-user analysis: An IT mandate. *ComputerWorld*, 1993.
- [GBLP95] Jim Gray, Adam Bosworth, Andrew Layman, and Hamid Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Microsoft Lab*, 1995.
- [GC97] Sanjay Goil and Alok Choudhary. High performance olap and data mining on parallel computers. *Center of Parallel and Distributed Computing Technical Report TR-97-05*, 1997.
- [NCCL07] Sébastien Nedjar, Alain Casali, Rosine Cicchetti, and Lotfi Lakhal. Emerging cubes for trends analysis in olapdatabases. In Il Yeal Song, Johann Eder, and Tho Manh Nguyen, editors, *DaWaK*, volume 4654 of *Lecture Notes in Computer Science*, pages 135–144. Springer, 2007.
- [Pen05] Nigel Pendse. *Olapreport: What is olap?*, 2005.
- [PJD01] Torben Bach Pedersen, Christian S. Jensen, and Curtis E. Dyreson. A foundation for capturing and querying complex multidimensional data. *Inf. Syst.*, 26(5):383–423, 2001.

- [Raf03] Maurizio Rafanelli, editor. *Multidimensional Databases: Problems and Solutions*. Idea Group Publishing, 2003.
- [Tho02] Erik Thomsen. *OLAP Solutions: Building Multidimensional Information Systems Second Edition*. Wiley Computer Publishing John Wiley & Sons, Inc., 2002.
- [YJA03] Ge Yang, Ruoming Jin, and Gagan Agrawal. Implementing data cube construction using a cluster middleware: algorithms, implementation experience, and performance evaluation. *Future Gener. Comput. Syst.*, 19(4):533–550, 2003.
- [Zha03] Yan Zhao. *Quotient Cube and QC-Tree: Efficient Summarizations for Semantic OLAP*. 2003.