Агрегирующие запросы на основе шаблонов к БД НИКА для OLAP-анализа

В.А. Тищенко[I,II]

[I] Федеральный исследовательский центр «Информатика и управление» Российской академии наук, г. Москва, Россия
[II] Образовательное частное учреждение высшего образования «Православный Свято-Тихоновский гуманитарный университет», г. Москва, Россия

Аннотация. Агрегирующие запросы на основе шаблонов (ATQ) – тип аналитического запроса в OLAP-системе. Предлагается реализация ATQ запросов к БД НИКА с использованием массовых запросов. Шаблон запроса формулируется на языке описания данных OOML. Способ организации ATQ запросов позволяет выполнять параллельно несколько таких запросов, что бывает необходимо для последующего интеллектуального анализа данных. Агрегированные данные, полученные в результате ATQ запроса, сериализуются в файл на диске для последующей обработки и отображения аналитических данных в виде таблиц и графиков посредством гипертекстовой системы СУБД НИКА.

Ключевые слова: *ООСУБД НИКА, агрегирующие запросы на основе шаблонов, оперативная аналитическая обработка иерархических данных, объектно-ориентированный язык разметки, массовые запросы, интеллектуальный анализ данных.*

DOI: 10.14357/20790279250406 **EDN:** JGISYQ

Введение

Статистическая обработка собранной информации в базе данных является известной технологией, реализованной в различных СУБД в виде оперативной аналитической обработки данных OLAP. Современные средства OLAP унифицированы и имеют стандартный набор функциональных блоков. В гипертекстовой системе для СУБД НИКА целью реализации данной технологии является возможность построения статистических исследований на основе OLAP. С этой точки зрения множество агрегированных значений, полученных в результате OLAP-обработки, следует обогатить некоторыми статистиками, характеризующими полученные срезы данных. Например, при рассмотрении срезов, описывающих статистику арестов и смертей по годам для разных категорий репрессированных в БД «За Христа пострадавшие», является важным сравнение графиков по разным категориям. Для этого кроме обычных агрегированных значений рассчитывается статистика R^2 *множественный коэффициент детерминации* для различных категорий репрессированных по отношению к общей статистике репрессированных по смертям и арестам. Такой подход дает возможность наделить технологию OLAP элементами интеллектуального анализа данных.

Предварительными понятиями являются массовые запросы [1] и объектно-ориентированный язык разметки OOML [2].

Запрос состоит из объекта запроса и набора ограничений, накладываемых на значения атрибутов запроса, связанных логическими операциями. Если значения одного или нескольких атрибутов определить как параметры запроса, то параметризованный запрос будет задавать класс однотипных запросов при разных значениях параметров.

Под *массовым запросом* к БД НИКА будем понимать серию однотипных запросов с одним или несколькими параметрами, которые последовательно принимают значения из заданных подмножеств значений соответствующих атрибутов. Границы изменения параметров указываются в запросе для соответствующих атрибутов. По умолчанию берется все множество значений, которые данный атрибут принимает в базе данных.

Объектно-ориентированный язык разметки OOML является языком описания данных, предназначенным для организации документного интер-

фейса БД НИКА. OOML имеет текстовый формат, содержащий некоторые символы разметки, и разработан раньше языка SGML по смыслу аналогичен ему. Запросы к БД НИКА формулируются на языке разметки OOML.

Формально ATQ запрос – это массовый запрос в виде *параметризованного* запроса на агрегированные значения. Параметризованный запрос задает множество R однотипных запросов r_a вида:

$$r_a(P_1,\ldots,P_m)=<O,f(A_l,\ldots,A_{lm}\,A_{lm+1},\ldots,A_n;\{P_1\},\ldots,\{P_m\},S_{m+1},\ldots,S_n)>.$$

Здесь O – объект запроса; f – булева функция, представляющая собой логическое выражение, связывающее между собой накладываемые на атрибуты $A_l,\ldots,A_{lm},A_{lm+1},\ldots,A_n$ ограничения, принимающие значения на соответствующих подмножествах $\{P_1\},\ldots,\{P_m\},S_{m+1},\ldots,S_n$, где $S_l\subseteq D(A_l)$, l=m+1,…,n. Элементы одноэлементных подмножеств $P_1,\ldots,P_m$ являются параметрами запроса, соответствующие значениям измерений, по которым считаются агрегированные значения. Таким образом

$$R=\{r_a(P_1,\ldots,P_m)\mid P_l\in S_l\subseteq D(A_l),\ l=1,\ldots,m\ \},$$

где S_l, l=1,…,m обозначает рассматриваемое подмножество значений на множестве определения $D(A_l)$ для атрибута A_l. Мощность множества R равно произведению мощностей подмножеств S_l:

$$|R|=\prod_{l=1}^{m}|S_l|.$$

1. Выполнение массовых запросов к СУБД НИКА

В ATQ запросах к БД НИКА используется типичная процедура параллельной обработки данных (рис.1). Процедура состоит из: разбиения большей задачи на меньшие, независимые подзадачи; их одновременное выполнение; сбор и объединение промежуточных результатов для получения итогового результата.

Естественным разбиением ATQ запроса являются отдельные запросы, из которых он состоит. Процедура включает в себя на 1-ом этапе параллельный вызовов запросной функции с очередным набором параметров. При каждом вызове на вход запросной функции передается i-ый набор параметров ($P_1[i_j],\ldots,P_m[i_j]$), j=1,…,k, которым инициализируется OOML-шаблон для формирования очередного запроса к БД НИКА. На 2-ом этапе происходит одновременное выполнения k запросов в цикле. Условием выхода из цикла является значение переменной цикла i, достигшее

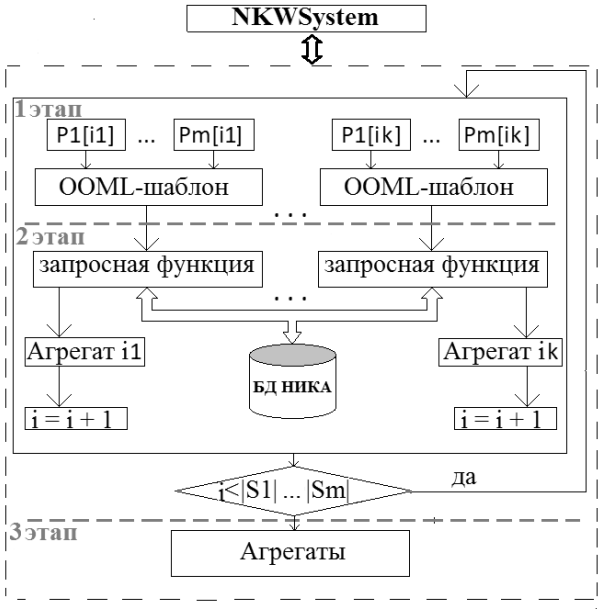


Рис. 1. Процедура выполнения массового запроса

полного числа сочетаний параметров на заданных подмножествах. На 3-ем этапе после выхода из цикла посчитанные в результате запросов агрегированные значения сериализуются в файл для дальнейшей обработки и отображения в гипертекстовой системе БД НИКА NKWSystem. В качестве значения k по умолчанию предлагается брать величину, равную числу логических потоков процессора. В общем случае эта величина является параметром, задаваемым в конфигурации программы запроса. На рис.1 каждое из значений $i_1,\ldots,i_k$ принимает текущее значение i. $S_l\subseteq D(A_l)$, l=1,…,m обозначают подмножества значений, которые принимают соответствующие параметры при обработке агрегирующего запроса. Здесь $D(A_l)$ обозначает соответствующую область определения параметра. Число таких сочетаний равно произведению мощностей подмножеств значений параметров: $|S_1|\ldots|S_m|$. Под параметрами понимаются значения измерений, которым соответствуют определенные поля в БД, выбранных в качестве параметров.

2. Шаблоны запросов на языке OOML

2.1. Порядок выполнения запросов на языке OOML

Опыт реализации запроса на языке XML изложен в работе [3]. На языке OOML [2] запрос представляет собой объект запроса и ограничения на атрибуты объекта: $r=<O,f(A_1,\ldots,A_n,S_l,\ldots,S_n))>$. Здесь O – объект запроса, f – булева функция с ключами $A_1,\ldots,A_n$ и подмножествами значений

атрибутов $S_l \subseteq D(A_l)$, l=1,…,*n* в качестве аргументов, которая определяется логическим выражением над ограничениями.

OOML файл запроса

```
<Doc>
Object
д~
<Atr>
OP1~Attribute1~Value1~Min1~Max1~
<Atr>
OP2~Attribute2~Value2~Min2~Max2~
                • • •
<Atr>
OPk~Attributen~Valuen~Minn~Maxn~
```

Рис. 2. Формат запроса на языке OOML

На рис.2 Object обозначает объект запроса, OP_i – логическую операцию И. ИЛИ, НЕ; $Attribute_i$ – *i*-ый ключ атрибута; $Value_i$ – *i*-ое значение атрибута; Min_i – нижнее граничное значение *i*-ого атрибута; Max_i —верхнее граничное значение *i*-ого атрибута; *i*=1,2,…,*n*. $Value_i$ задает ограничение в форме равенства $v_i=Value_i$. Min_i и Max_i задают ограничения в форме неравенств: $v_i \geq Min_i$ и $v_i \leq Max_i$. Запрос в OOML-формате задает простой запрос без скобочных выражений в логическом условии. Для определения составного запроса используется цепочка простых запросов. Каждому скобочному выражению соответствует отдельный запрос, результат которого может входить как атрибут в последующие запросы цепочки. Особенность реализации OOML-запроса в том, что вычисление логического выражения происходит последовательно в порядке расположения ограничений в OOML-документе. Все логические операции имеют одинаковый приоритет. Выражение $v_1 \leq Max_1 \leq v_2=Value_2 \wedge v_3 \geq Min_3$ и будет вычисляться в порядке следования ограничений. Логическая операция $\wedge$ (И) имеет такой же приоритет как логическая операция $\wedge$ (ИЛИ). Для изменения порядка выполнения операций определяется запрос, состоящий из двух простых запросов: $v_2=Value_2 \wedge v_3 \geq Min_3$ и $v_1 \leq Max_1 \vee r(A_2,V_2,A_3,V_3)$, где $r(A_2,V_2,A_3,V_3)$ – объект, полученный в результате выполнения первого запроса.

2.2. Параметризация OOML запроса

Агрегирующие запросы на языке SQL рассматриваются в [4]. Для выполнения агрегирующих запросов на основе шаблонов ATQ предлагается ввести управляющие метасимволы, позволяющие задавать ключи и значения определенных атрибутов в качестве параметров к OOML-запросу. При каждой итерации массового запроса (рис.1) происходит вызов запросной функции с одним(-и) и тем(-и) же OOML-шаблоном(-ами), но с другим набором параметров. Метасимволы указываются в OOML-шаблоне запроса в виде строки %s в тех позициях, где в OOML-файле задаются ключи и/или значения атрибутов. Верхнюю и нижнюю границы, в которых меняются параметризованные атрибуты, задаются также как и для обычных атрибутов. Пример OOML определения для параметризованного атрибута:

или~**%s**~**%s**~1929~1933~

Здесь первый управляющий метасимвол соответствует ключу атрибута ГодАреста, а второй – его значению из диапазона [1929;1939]. На рис.3 дается пример с управляющими метасимволами.

```
<Doc>
Дела.ПЕРИОДЫ ЖИЗНИ.Аресты
д~
<Atr>
или~%s~%s~1929~1933~
<Atr>
и~%s~%s~~~
```

Рис. 3. Пример OOML-шаблона ATQ запроса

При подстановке в шаблон в качестве параметризованных могут использоваться разные атрибуты:

ATQ(ГодАреста, 1930, МесяцАреста, 5) или
ATQ(ГодОсуждения, 1930, Месяцосуждения, 5).

Для составного ATQ запроса используется сразу несколько OOML-шаблонов.

3. Сериализация агрегированных данных

ATQ запросы могут обрабатываться параллельно, поскольку результаты записываются в отдельные tre-файлы по основной схеме описания данных. Полученные в результате массовых запросов агрегированные значения счетчиков сериализуются в виде массива в olap-файл. Кроме суммарных значений числа отобранных объектов в olap-файл записываются соответствующие ключи, по которым происходит агрегация объектов. Например, для указанного в OOML-шаблоне ATQ запроса агрегирующего атрибута ГодАреста отписываются все возможные значения этого атрибута и соответствующие агрегированные значения. Возможен не один агрегирующий атрибут: (ГодАреста$\in$[1929;1933],МесяцАреста). В этом случае вместе с агрегированными значениями сериализуются значения всех агрегирующих атрибутов: {(1929,1),…,(1929,12),…,(1933,1),…,(1933,12)}. Сериализуемые данные – это OLAP-структура и массив.

OLAP-структура содержит общую информацию о числе элементов в массиве, о числе уровней счетчиков, о массиве индексов для упорядочения по величинам агрегированных значений, а не по ключам и другую информацию. Вместе с OLAP-структурой располагается массив ключей агрегирующих атрибутов и соответствующих им агрегированных значений, т.е. суммарных счетчиков объектов.

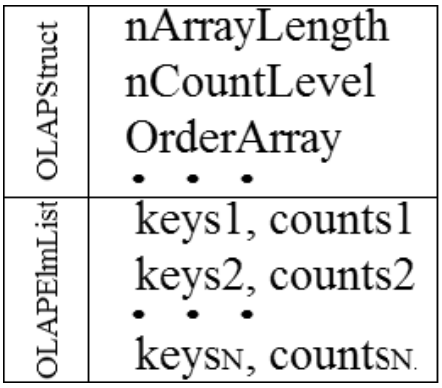


Рис. 4. Структура формата сериализуемых данных

На рис.4 *N=nArrayLength*, *OLAPElemList* – список-массив. В контексте ATQ запроса число агрегирующих атрибутов может обозначать разную степень детализации по одному иерархическому измерению [5]. В вышеприведенном примере – это временное измерение, которое может отсчитываться по годам, месяцам, дням (ГодАреста, МесяцАреста, ДеньАреста). Другим примером иерархического измерения может служить поле МестоАреста. Значения по этому полю содержат информацию о регионе, подрегионе и населенном пункте ареста. Согласно уровням административно-территориального деления пространственное измерение можно строить с разной степенью детализации. В СУБД НИКА также реализована возможность задавать пороговое значение на число объектов, в данном случае, на число арестов в регионе, подрегионе, населенном пункте. Все значения измерения, меньшие порогового, фильтруются, что позволяет уменьшить размерность задачи для дальнейшей статистической обработки.

4. Пример агрегирующего запроса

Спецификация «график», присвоенная вершине типа структура или массив [6], может задавать несколько графиков, отображающих результаты агрегирующих запросов на основе шаблонов. Агрегирующие запросы по одной спецификации «график» – это массовые запросы, запускаемые параллельно. Длительность обсчета множества массовых запросов будет определяться обсчетом наиболее продолжительного массового запроса. При числе записей порядка 36,5 тыс. имен эта величина составляет менее 1сек, что является приемлемым при работе с БД в интерактивном режиме в Интернет.

Примером агрегирующих запросов может служить спецификация «график», применяемая к вершинам ГодАреста и МесяцАреста, для подсчета числа арестов по заданным регионам по месяцам за период времени с 1929 по 1933гг. Посредством групповой спецификации получается многократное применение спецификации «график» к заданным вершинам, но с разными агрегирующими запросами: общее число арестов; число арестов по 35 регионам с наибольшим числом арестов; число арестов по Москве; число арестов по Московской области. Все запросы выполняются по месяцам рассматриваемого периода (рис.5).

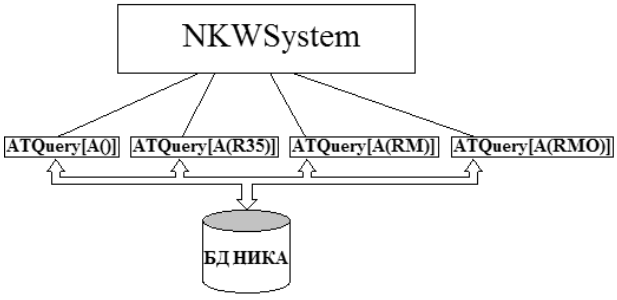


Рис. 5. Выполнение агрегирующих запросов: аресты A(0), аресты по 35 регионам A(R35), аресты по Москве A(RM), аресты по Московской области A(RMO)

Полное описание применения агрегирующих запросов и последующая статистическая обработка их результатов приведены в статье [7]. Исходным массивом данных являются агрегированные значения по арестам отдельно для каждого из 35 регионов с наибольшим количеством арестов по 60 месяцам 5-летнего периода 1929-1933гг., полученные в результате запросов к БД «За Христа пострадавшие» [8]. В результате применения метода главных компонент были выделены 5 *характерных* регионов ареста. Первые 2 из них – Москва и Московская область – рассматриваются в приведенном примере агрегирующих запросов. Полученные агрегированные значения удобно сохранять в виде среза многомерного куба данных для дальнейшего их использования и обработки отдельно от основной БД подобно материализованным представлениям в реляционных базах данных. Сохраненные срезы соответствуют случаю агрегированных значений в виде статических счетчиков для основных индек-

сов, которые обсчитываются заранее и сохраняются в БД. Чтение этих счетчиков требует менее 1 с.

5. Отображение агрегированных данных

Для отображения статистики в виде таблицы и графиков используется спецификация «график» (GRA), которая присваивается вершине типа массив. Например, спецификация GRA может быть присвоена любой вершине в индексе. Для анализа и сравнения статистических зависимостей спецификация GRA присваивается вершине INDEX (структура) и в ней указывается список индексных вершин, участвующих в отображении.

Табл. 1

Атрибуты спецификации GRA

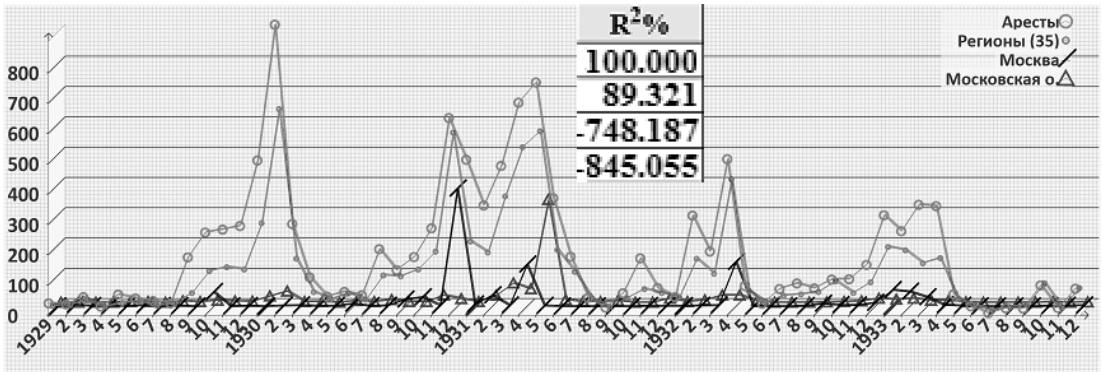
Тип вершины	Атрибут	Описание
Массив	274	Полное имя файла OOML-запроса
Структура	404	Иерархический список имен индексных вершин

Пример графиков (рис.6) распределения по месяцам периода 1929-1933гг. числа арестов общий (кружки), по 35 регионам с наибольшим количеством арестов (точки), по Москве (наклонные черточки) и по Московской облапсти (треугольники) иллюстрирует применение спецификации GRA к вершинам INDEX и ГодАреста [7]. Первая таблица на рис.6 содержит исходные статистические данные – агрегированные значения по арестам, вторая – некоторые статистики, посчитанных на основании данных из табл. 1.

Для регионов Москва и Московская область статистика R^2 принимает значения меньшие нуля. Это означает, что графики для этих регионов некорректно описывают или приближают общий график арестов. В противоположность этому, для 35 регионов R^2>80%, что говорит об очень хорошем приближении общих данных по арестам посредством данных по 35 регионам. Основной вывод состоит в том, что распределения арестов для отдельных регионов не характеризует общего их распределения. Пять характерных регионов ареста (Москва, Московская область и т.д.) названы так в смысле отдельных пиков (волн) ареста, для которых число арестов в этих регионах было максимальным. Для Москвы это пик арестов, соответствующий декабрю 1930 г., а для Московской области – маю 1931 г.

Заключение

Актуальность применения ATQ запросов состоит в том, чтобы добавить возможность оперативной обработки данных для ООСУБД НИКА, в частности для фактографической БД «За Христа пострадавшие». Это позволяет отображать полученные срезы данных в виде таблиц с агрегированными значениями по различным измерениям и с основанными на них статистиками, а также представлять их в виде графиков в гипертекстовой системе СУБД НИКА [9]. Кроме того, полученная статистика служит исходным материалом для применения статистических методов с целью полу-



Месяц	1930(13)	2(14)	3(15)	4(16)	5(17)	6(18)	7(19)	8(20)	9(21)	10(22)	11(23)	12(24)	1931(25)
Аресты	474	**890**	279	116	57	71	61	202	138	178	266	604	476
Регионы (35)	269	**619**	160	58	36	54	35	109	106	127	182	547	213
Москва	0	2	3	1	1	8	1	2	21	28	3	**361**	6
Московская о.	19	34	2	2	0	5	2	8	2	2	22	11	2

Статистика	AVG	EXP	STD	CST	ABL	I	T	K	I*	d	S	td	tS	SAV	ABS	LSM	R^2%
Аресты	192.833328	383.373901	239.977936	3	0.344137	912	856	-56	0.344408	4	12	1.474446	2.107877	-2.616111	0.000000	0.000000	**100.000**
Регионы (35)	122.766670	285.641357	201.292099	7	0.456205	872	888	16	-0.165826	4	10	1.474446	1.199360	-2.143679	0.201406	0.001759	**89.321**
Москва	16.133333	181.064056	144.886078	2	1.037398	727	793	66	-2.015425	3	3	1.105834	-1.980449	-0.461196	1.208238	0.139684	**-748.187**
Московская о.	11.866667	152.955048	145.001373	0	0.869798	802	748	-54	-1.058736	7	7	2.580280	-0.163415	-1.982939	0.939067	0.155636	**-845.055**

Рис.6. Примеры графиков и таблиц со статистическими данными по арестам [7]. В таблицах: 1 строка – заголовок, 2 – аресты, 3 – аресты по 35 регионам, 4 – аресты по Москве, 5 – аресты по Московской области Последняя колонка R^2 во второй таблице со статистиками вынесена на график

чения результатов в прикладной области, а также выявлению наиболее значимых факторов для объяснения полученных зависимостей.

Литература

1. *Соловьев А. В., Тищенко В. А.* Расширение запросной системы макетного генератора информационных систем возможностью организации массовых запросов к БД НИКА // Изд-во ООО «Аспект». Самара. 2022. Т. 2. № 6. С. 226-232.
2. *Bogacheva A.N.* Object Oriented Markup Language and Restructuring Hierarchical Database Objects / A.N. Bogacheva, N.E. Emeljanov, A.P. Romanov // Proceeding ADBIS '95 Proceedings of the Second International Workshop on Advances in Databases and Information Systems. June 27 - 30. 1995. P. 137-142.
3. Реализация запросной системы на основе XPath для ООСУБД НИКА. Богданов А.С., Емельянов Н.Е., Ерохин В.И., Романов Б.Л. // Труды ИСА РАН. М.: Едиториал УРСС. 2003. С. 130-146.
4. *Гарсиа-Молина Г., Ульман Д., Уидом Д.* Системы баз данных. Полный курс.: Пер. с англ. М.: Издательский дом. «Вильямс». 2003. С. 1003-1048.
5. *Хрусталев Е.М.* Агрегация данных в OLAP-кубах // Открытые системы. 2003. №5. С. 33-38.
6. *Emelyanov N.E., Tishchenko V.A.* Principles of building a web server based on an object-oriented database // Information technology and computing systems. 1997;4:90-99.
7. *Soloviev A., Bogacheva A., Tishchenko V.* Construction of a Multidimensional Data Cube for a Factual Database: Feature Extraction Using the Principal Component Analysis Based on the Example of Repression Statistics by Region // Artificial Intelligence for System Oriented Design. CoMeSySo 2024. Lecture Notes in Networks and Systems. Springer, Cham. 2025;1489:43–68 https://doi.org/10.1007/978-3-031-96798-6_5
8. БД «За Христа пострадавшие» https://martyrs.pstbi.ru
9. *Емельянов Н.Е., Тищенко В.А.* Представление гипертекста в СУБД НИКА // Технология программирования и хранения данных / Труды ИСА РАН. 2009. Т.45. С. 17-36.

Тищенко Владимир Александрович. Федеральное государственное учреждение «Федеральный исследовательский центр «Информатика и управление» Российской академии наук», г. Москва, Россия. Научный сотрудник. Кандидат технических наук. Образовательное частное учреждение высшего образования «Православный Свято-Тихоновский гуманитарный университет», г. Москва, Россия. Инженер-программист. Область научных интересов: средства создания и поддержки электронных изданий и информационных систем. E-mail: vatischenko@frccsc.ru

Aggregate template-based queries to the NIKA database for OLAP analysis

V.A. Tishchenko[I,II]

[I] Federal Research Center "Computer Science and Control" of Russian Academy of Sciences, Moscow, Russia

[II] St. Tikhons's Orthodox University, Moscow, Russia

Abstract. Aggregate template-based queries (ATQ) are a type of analytical query in the OLAP system. The implementation of ATQ queries to the NIKA DB using bulk queries is proposed. The query template is formulated in the OOML data description language. The method of organizing ATQ queries allows several such queries to be executed in parallel, which is sometimes necessary for subsequent data mining. Aggregated data obtained as a result of the ATQ query are serialized into a file on the disk for subsequent processing and display via the NIKA DBMS hypertext system.

Keywords: *NIKA DBMS, template-based aggregate queries, OLAP of hierarchical data, object- oriented markup language, bulk queries, data mining*

DOI: 10.14357/20790279250406 **EDN:** JGISYQ

References

1. *Soloviev A.V., Tishchenko V.A.* Expanding the inquiry system of the information systems layout editor with the possibility of organizing bulk requests to the NIKA DB // Aspekt, Samara. 2022;2(6):226-232.
2. *Bogacheva A.N.* Object Oriented Markup Language and Restructuring Hierarchical Database Objects / A.N. Bogacheva, N.E. Emeljanov, A.P. Romanov // Proceeding ADBIS '95 Proceedings of the Second International Workshop on Advances in Databases and Information Systems. June 27–30. 1995. P. 137-142.
3. *Bogdanov A.S., Emelianov N.E., Erokhin V.I., Romanov B.L.* Implementation of a query system based on XPath for the OODBMS NIKA // Organizational management and artificial intelligence. Proceedings of the ISA RAS. Ed. by Doctor of Technical Sciences prof. Arlazarov V.L. and d.t.s. prof. Emelyanov N.E. M.: URSS. 2003. P. 130-146.
4. *Garcia-Molina H., Ullman J., Widom J.* Database Systems – The Complete Book. Pearson. 2008. 1248 p.
5. *Khrustalev E.M.* Data aggregation in OLAP cubes // Open Systems. 2003;5:33–38.
6. *Emelyanov N.E., Tishchenko V.A.* Principles of building a web server based on an object-oriented database // Information technology and computing systems. 1997;4:90–99.
7. *Soloviev A., Bogacheva A., Tishchenko V.* Construction of a Multidimensional Data Cube for a Factual Database: Feature Extraction Using the Principal Component Analysis Based on the Example of Repression Statistics by Region // Artificial Intelligence for System Oriented Design. CoMeSySo 2024. Lecture Notes in Networks and Systems. Springer, Cham. 2025;1489:43–68. https://doi.org/10.1007/978-3-031-96798-6_5
8. Database 'for Christ Suffered' https://martyrs.pstbi.ru
9. *Emelyanov N.E., Tishchenko V.A.* Representation of hypertext in the NIKA DBMS // Technology of programming and data storage / Sat. Proceedings of the ISA RAS. V.45. Ed. Corresponding Member RAS Arlazarov V.L. and Doctor of Technical Sciences prof. Emelyanov N.E. M. 2009. P. 17-36.

Vladimir A. Tishchenko. Researcher, Candidate of Technical Sciences. Federal Research Center "Computer Science and Control" of Russian Academy of Sciences. Software engineer. St. Tikhon Orthodox Humanitarian University. Number of printed works: 40. Research interests: tools of creation and support of electronic publications and information systems. E-mail: vatischenko@frccsc.ru