

## МНОГОДИМЕНСИОННИ МОДЕЛИ НА ДАННИТЕ В БИЗНЕС-ИНФОРМАЦИОННИТЕ СИСТЕМИ

Мила Нейкова, Румяна Цанкова  
Технически университет – София

Изграждането на информационни системи, подпомагащи вземането на решения, изисква бърз и гъвкав анализ на данните. Бизнес задачите обикновено имат многодименсионна структура и изискват представянето на данните в различни потребителски перспективи. Технологията OLAP (On Line Analytical Processing), базираща се на многодименсионни бази данни, е подходяща за целите на йерархичния анализ и агрегиране на данните, особено в големите бази данни. Високата степен на агрегиране на данните и малкото време за отговор на заявки за анализ на данните по различни критерии са основните предимства на многодименсионните бази данни. Отчитайки че основният недостатък на многодименсионните бази данни е големият обем използвана памет, ние предлагаме методи за подобряване на многодименсионните модели на данните по отношение на използваната памет. Представена е технология за трансформация на релационния модел на данните в многодименсионни модели. Разглежданите многодименсионни модели са оценени по показателите използвана памет и функционалност. Многодименсионните модели се генерират програмно чрез SQL заявки.

**Keywords:** Decision Support Systems, multidimensional data models, OLAP, memory consumption

**Ключови думи:** Информационни системи, подпомагащи вземането на решения, многодименсионни модели на данните, OLAP, обем памет.

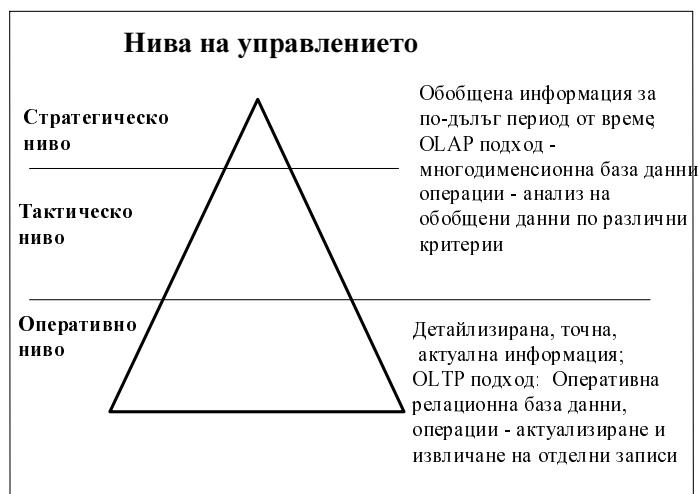
### 1. Въведение

Според особеностите и функционалните характеристики на потребителското ниво на представяне на данните и операциите с тях, информационните системи (ИС) се групират в няколко вида [1]:

- управленски ИС (MIS - management information systems);
- ИС за управление на обекти и технологични процеси (вкл. и околната среда);
- ИС за автоматизация на проектантския и инженерния труд;
- учебни и самообразователни ИС.

Системите, подпомагащи вземането на решения (DSS - Decision Support Systems), са важен клас от MIS. Бизнес-информационни системи са тези

системи, които подпомагат вземането на решения в управлението на стопански дейности. Най-общо управлението на една организация може да се раздели на три нива - оперативно, тактическо и стратегическо [2] (фиг. 1). Решенията, които се вземат на най-ниското (оперативно) ниво на управление, обикновено изискват вътрешна за организацията, детайлизирана, точна и актуална информация. В системите, подпомагащи вземането на решения на по-високите нива на управление - тактическо и стратегическо, е необходима както вътрешна, така и външна за организацията информация, обобщена за по-дълъг период от време. Вземането на по-добри управленски решения се подпомага чрез обобщаване и анализ на данните по различни критерии.



Фиг. 1 Нива на управлението

Представянето на данните в информационните системи може да стане чрез традиционната технология OLTP (On Line Transaction Processing), основаваща се на релационни бази данни (БД) или чрез сравнително новата технология OLAP (On Line Analytical Processing), базираща се на използването на многодименсионни БД. Технологията OLTP е ориентирана към обработката на отделни записи - основните операции върху данните са свързани със създаване, актуализиране и извличане на отделни записи. OLAP е насочена към обработката на агрегирани данни. В OLAP приложенията се осъществява обединяване на данните и разполагане на агрегираните данни по различни координатни оси (дименсии).

В системите, подпомагащи вземането на управленски решения на ниско ниво, е подходящо използването на OLTP и съхранение на данните в оперативна релационна БД. На по-високите нива на управление, където основните операции върху данните са свързани с обобщаване и анализ, по-подходящо е приложението на OLAP подхода и съхраняване на данните в многодименсионни БД. Изискванията на различните потребители на информация (анализатори, мениджъри) налага представянето на информацията в различни потребителски

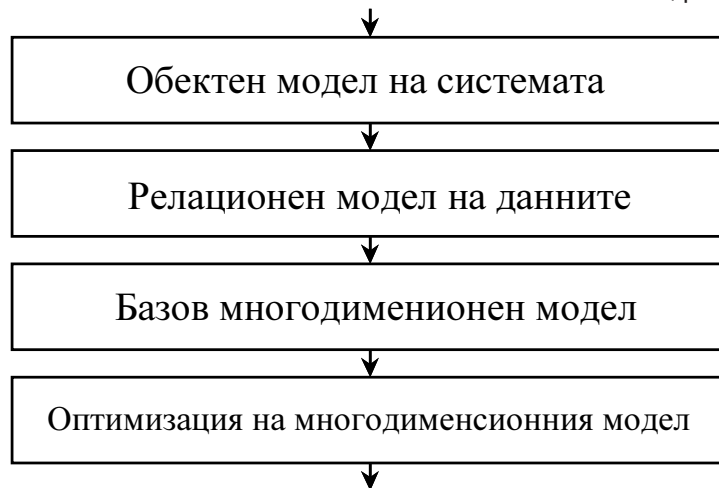
ки перспективи. Бизнес-задачите обикновено имат многодименсионна структура [3] (напр. мениджърите по продажбите биха попитали не само "Колко коли са продадени? - въпрос с нулева дименсия, но и "Какви са продажбите по отделни градове, магазини, модели и т.н.? "). Съхраняването на данните в многодименсионна БД позволява отговор на подобни многодименсионни въпроси.

Спецификата на многодименсионните модели на данните обуславя техните предимства и недостатъци в сравнение с релационните модели. Основно предимство на многодименсионните БД е високото бързодействие при анализ на данните, а съществен недостатък е големият обем необходимата памет [4].

Целта на настоящето изследване е генериране на различни многодименсионни модели на базата на релационна БД, оценка на многодименсионните модели по отношение на тяхната функционалност и използвана памет и оптимизация на многодименсионните модели по отношение на използваната памет.

## 2. Многодименсионни модели на данните и оценка на паметта

Технологията на изследването ни следва няколко основни стъпки (фиг.2).

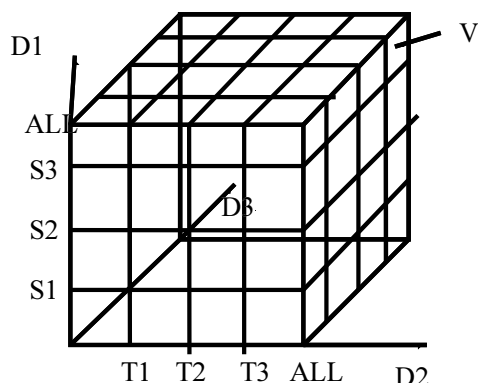


Фиг. 2 Основни стъпки на технологията на изследването

Първоначално разглежданата приложна област се представя с обектен модел, включващ основните обекти и връзките между тях. На базата на този модел се изгражда релационен модел на данните, основаващ се на отношения (релации) между различни свойства (атрибути) на отделните обекти. На всеки атрибут на обект съответства област на определеност (домен) - множество от допустими стойности [5]. Генерирането на многодименсионен модел на данните, известен още и като многодименсионен куб (multidimensional data cube) на базата на релационна база данни предполага

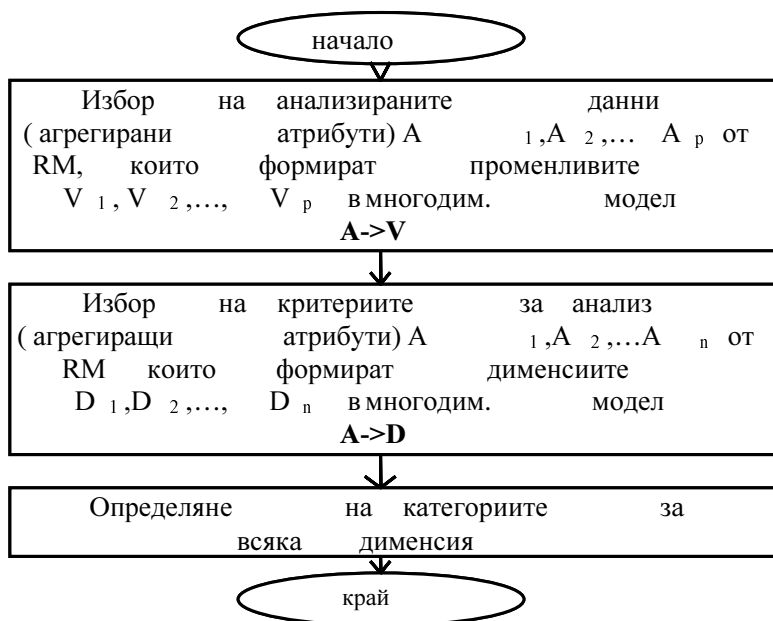
изобразяването на някои полета (агрегиращи атрибути) от релационния модел в дименсии в многодименсионния модел. Различните стойности на съответните полета стават категории в отделните дименсии. Анализиранияте атрибути се агрегират и изобразяват в т.нар. променливи (variables, summaries) в многодименсионния модел (фиг.3).

Агрегираните стойности на променливите за всяка комбинация от категории в дименсиите формират клетките (cells) в многодименсионния куб. Алгоритъмът за изобразяване на релационен в многодименсионен модел на данните е показан на



D1,D2,D3 are the dimensions; S1,S2,S3 - categories in D1  
 T1,T2,T3 - categories in D2  
 V - a variable (summary)  
 ALL - a fictive value for all categories in a dimension

Фиг. 3



Фиг. 4 Алгоритъмът за изобразяване на релационен в многодименсионен модел на данните

фиг.4.

Има два основни подхода за съхраняване на получените многодименсионни БД - в таблици (т.нар. Relational OLAP (ROLAP) подход) и в многодименсионни масиви (Multidimensional OLAP (MOLAP) подход) [6].

За илюстрация на OLTP и OLAP подхода за представяне на данните ще разгледаме примерна информационна система, подпомагаща управлението на фирма за продажба на автомобили. Регистрирането на всяка сделка е удобно да се извършва в оперативна релационна база данни. Обобщаването и анализа на данните за продажбите за по-голям период от време може да стане с

помощта на технологията OLAP и многодименсионна база данни (фиг.5).

Вижда се че в многодименсионната база данни се появяват записи за обобщените стойности за различните агрегиращи атрибути (all), както и запис за общата сума на продажбите за цялата база данни (стойности all за всички агрегиращи атрибути). В примера като агрегиращи атрибути са избрани "Магазин", "Модел", "Цвят" на колите, а анализираната величина е "Сума (общ брой) на продажбите".

Размерът на паметта, заемана от многодименсионната БД, зависи от броя на дименсиите, броя на категориите във всяка дименсия, броя и раз-

The screenshot shows a software window titled "CARS SALES SYSTEM" with two data tables side-by-side. The left table is labeled "OLTP APPROACH : RELATIONAL DATABASE" and the right table is labeled "OLAP APPROACH: MULTIDIMENSIONAL DATABASE, STORED AS A TABLE".

STORE	MODEL	COLOR	SALES
M1	Ford	White	10
M2	Ford	Red	20
M3	Reno	Red	5
M1	Reno	Red	5
M2	Ford	White	10
M3	Fiat	Green	15
M3	Fiat	Green	5
M1	Reno	Green	2
M3	Reno	Red	4
M2	Ford	White	1
M1	Ford	White	7
M1	Reno	Red	6
M1	Fiat	Red	8
M2	Fiat	Green	10
M3	Fiat	Green	3
M3	Ford	Red	3
M2	Ford	Red	2
M1	Ford	White	1
M2	Reno	White	5
M1	Reno	Red	5

Store	Model	Color	SUM OF SALES
M2	all	Red	22
M2	all	White	16
M2	all	all	48
M3	Fiat	Green	23
M3	Fiat	all	23
M3	Ford	Red	3
M3	Ford	all	3
M3	Reno	Red	9
M3	Reno	all	9
M3	all	Green	23
M3	all	Red	12
M3	all	all	35
all	Fiat	Green	33
all	Fiat	Red	8
all	Fiat	all	41
all	Ford	Red	25
all	Ford	White	29
all	Ford	all	54
all	Reno	Green	2
all	Reno	Red	25
all	Reno	White	5
all	Reno	all	32
all	all	Green	35
all	all	Red	58
all	all	White	34
all	all	all	127

Фиг. 5

мера на променливите. Добавянето на нова променлива увеличава използваната памет линейно (напр. многодименсионен куб с 2 променливи използва 2 пъти повече памет от същия куб с 1 променлива). Между размера на всяка променлива и размера на куба също съществува линейна зависимост. Добавянето на нова дименсия увеличава използваната памет по-бързо (напр. добавянето на нова дименсия с 10 категории увеличава използваната памет 10 пъти) [7]. Отчитайки, че броят на дименсиите е най-важният фактор, определящ размера на многодименсионния куб, ние насочихме вниманието си към методи за моделиране на многодименсионната БД, водещи до намаляване на броя на дименсиите.

Важен фактор за оценка на многодименсионните модели е и тяхната функционалност. Тя се определя от възможността за изчисляване на обобщени стойности на променливите за всички комбинации от агрегиращи атрибути (intersections) и частични и пълна сума на променливите в многодименсионния куб (за примерната система за продажба на коли е необходимо да се изчислят сумите от продажбите по цвят, модел и магазин; по модел и цвят; по цвят и магазин; по магазин и модел;

по цвят; по модел; по магазин и общата сума на продажбите).

В нашето изследване многодименсионното моделиране на данните започва с базовия многодименсионен модел (Basic Multidimensional Model - BMM), в който всеки агрегиращ атрибут (поле) от релационната БД се изобразява в отделна дименсия в многодименсионния куб. Дори йерархично свързани атрибути се изобразяват в отделни дименсиии. Ако съществува проста йерархия на атрибути - "Град" - "Магазин", за обобщаване на данните за продажбите на коли по град, магазин, модел и цвят е необходимо генериране на четиридименсионна БД с дименсиии град, магазин, модел, цвят и променлива сума на продажбите (фиг. 6). Някои OLAP сървери позволяват представяне на няколко йерархични нива в една дименсия, но нашата разработка е насочена към OLAP сървери без такива възможности.

Предимства на BMM е простотата (както в структурата, така и в генерирането му) и възможността за представяне на обобщените стойности на променливата за всички комбинации от дименсиии. Основният недостатък на този модел е големият обем използвана памет. Броят на клетките в

CARS SALES SYSTEM : BASIC MULTIDIMENSIONAL DATA MODEL

		M1				M2				M3			all			
Town	Model	Green	Red	White	all	Green	Red	White	all	Green	Red	all	Green	Red	White	all
SOFIA	Fiat		8		8	10			10				10	8		18
	Ford			18	18		22	11	33				22	29		51
	Reno	2	16		18			5	5				2	16	5	23
	all	2	24	18	44	10	22	16	48				12	46	34	92
VARNA	Fiat									23		23	23			23
	Ford										3	3		3		3
	Reno										9	9		9		9
	all									23	12	35	23	12		35
all	Fiat		8		8	10			10	23		23	33	8		41
	Ford			18	18		22	11	33		3	3		25	29	54
	Reno	2	16		18			5	5		9	9	2	25	5	32
	all	2	24	18	44	10	22	16	48	23	12	35	35	58	34	127

Фиг. 6

многодименсионния куб може да се изчисли по формулата:

$$NC = C_1 * C_2 * C_3 * \dots * C_n \quad (1)$$

където NC е броят на клетките;  $C_i$  е броят на категориите за i-тата дименсия; n е броят на дименсиите. Колкото повече са агрегиращите атрибути, толкова по-голям е броят на дименсиите и обемът на необходимата памет.

За да подобрим показателите на многодименсионния модел по отношение на използваната памет, ние предлагаме намаляване на броя на дименсиите чрез обединяване на атрибути в общи дименсии. Разглеждаме няколко различни подхода за обединяване на атрибути в общи дименсии като оценяваме използваната памет и тяхната функционалност.

Първият метод за подобряване на многодименсионния модел на данните е да се обединят йерархично свързани атрибути в обща дименсия. Нека да разгледаме йерархията от атрибути  $A_1, A_2, \dots, A_m$ . Изобразяването на тези атрибути в една дименсия води до съществено намаляване на размера на многодименсионния куб. Броят на клетките може да се изчисли по формулата:

$$NC = (C_1 + C_2 + \dots + C_m) * C_{m+1} * \dots * C_n \quad (2)$$

от която се вижда, че n дименсии се редуцират до n-m + 1 дименсии. Броят на категориите в общата дименсия е равен на  $C_1 + C_2 + C_3 + \dots + C_m$ .

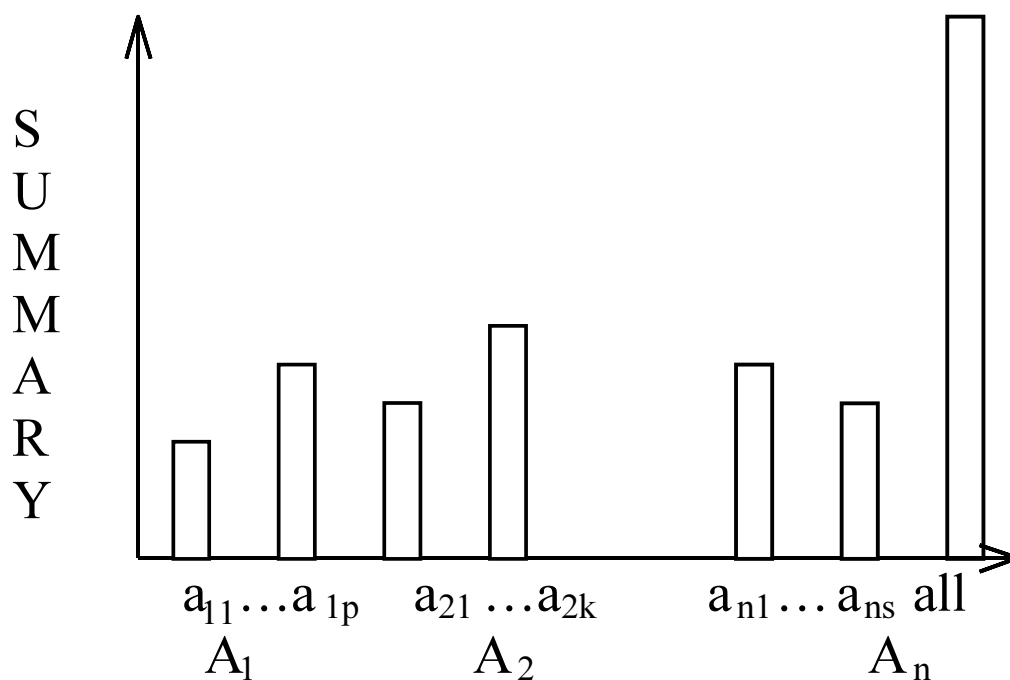
В получения подобрен многодименсионен модел (Improved Multidimensional Model - IMM) няма възможност за изчисляване на обобщените стойности на променливата за комбинации от стойностите на атрибутите, обединени в обща дименсия. Ние считаме, че това не влошава функционалността на многодименсионния модел, тъй като не е нужно изчисляването на променливата за йерархично свързани атрибути (напр. ако в обща дименсия са обединени йерархично свързаните атрибути "Град" и "Магазин", няма смисъл изчисляването на сумата от продажбите за която и да е комбинация от "Град" и "Магазин"; за нас е достатъчно да знаем сумата на продажбите или за даден град или за даден магазин).

Идеята за обединяване на атрибути в общи дименсии може да се разшири и за атрибути между които няма йерархична връзка. Изчисленията на необходимата памет ще са аналогични, но функционалността на модела ще се влоши, тъй като няма

да има възможност за пресмятане на обобщени стойности на променливата за някои комбинации от независими атрибути (напр. ако два независими атрибута "Град" и "Модел" са проектирани в една дименсия, няма да има възможност за изчисляване на сумата от продажбите за определен модел в даден град). Обобщението на идеята за обединяване на атрибути в общи дименсии води до генериране на еднодимензионна база данни, в която всички агрегиращи атрибути са изобразени в една дименсия. Изчисляването на размера на еднодимензионната база данни става по формулата:

$$NC = (C_1 + C_2 + \dots + C_n) \quad (3)$$

Обемът на използваната памет достига своя минимум, но функционалността на модела е силно влошена, тъй като няма възможност за изчисляване на обобщени стойности на променливата за никоя комбинация от атрибути. На практика еднодимензионната БД съдържа обобщените стойности на променливата за всяка категория  $a_{i1}, a_{i2}, \dots, a_{ip}$  на всеки атрибут  $A_i$  ( $i = 1, 2, \dots, n$ ) и сумарната стойност на променливата за цялата база данни (означена като 'all' в дименсията) (фиг. 7).



Фиг. 7

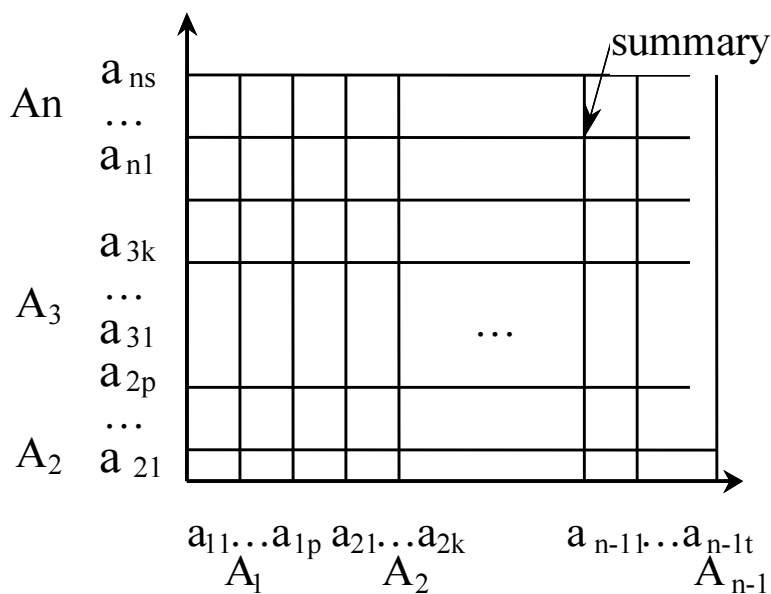
С цел намаляване на използваната памет без влошаване на функционалността на модела, ние предлагаме нов декомпозиционен модел на данните [1]. Обект с  $n$  атрибути ( $A_1, A_2, \dots, A_n$ ) се моделира чрез двумензионна база данни, където първата дименсия обединява атрибутите  $A_1, A_2, \dots, A_n$ , а втората дименсия - атрибутите  $A_2, A_3, \dots, A_n$  (фиг.8).

Този модел позволява представянето на обобщени стойности на променливата за всяка двойка агрегиращи атрибути при значително намаляване на размера на използваната памет благодарение на значителното намаляване на броя на дименсиите. Броят на клетките в двумензионния модел може да се изчисли по формулата:

$$NC = (C_1 + C_2 + \dots + C_{n-1}) * (C_2 + C_3 + \dots + C_n) \quad (4)$$

На базата на формулите (1), (2), (3), (4) мо-

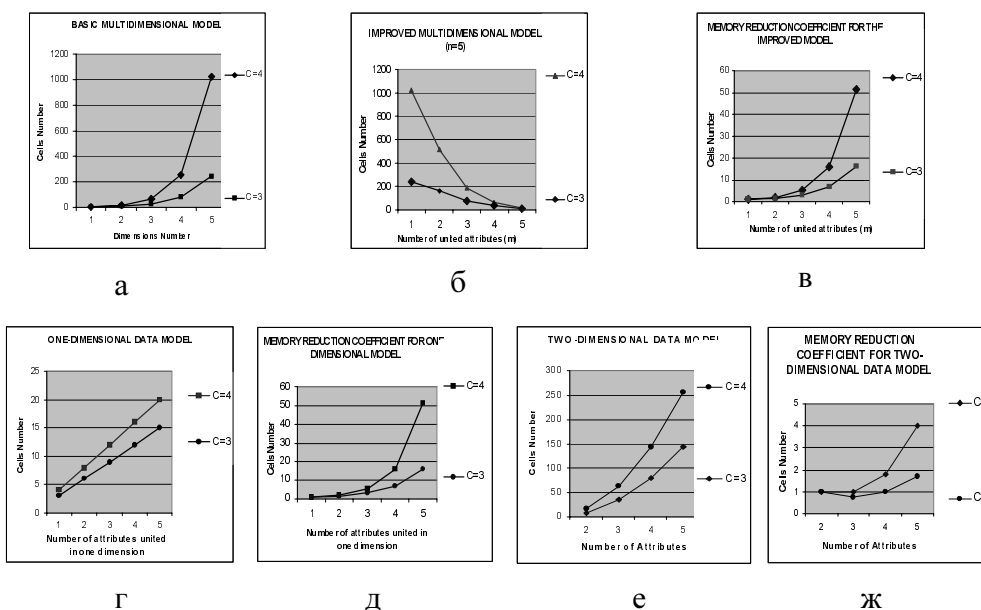
жем да направим сравнителен анализ на разглежданите модели по отношение на използвания обем памет в зависимост от броя на дименсиите и броя на категориите във всяка дименсия. На фиг.9а е представена графически зависимостта на размера на BMM (в брой клетки) от броя на агрегиращите атрибути -  $n$  (който е равен на броя на дименсиите) и броя на категориите във всяка дименсия -  $C$  (за простота е прието, че броят на категориите е еднакъв за всяка дименсия; разгледани са два примерни случая за брой на категориите -  $C = 3$  и  $C = 4$ ). Графиката показва експоненциалната зависимост на размера на многодимензионния куб от броя на дименсиите. На фиг. 9б е демонстрирано намаляването на използваната памет в IMM, при който няколко атрибута се обединяват в обща дименсия. Броят на обединените в обща дименсия атрибути е



Фиг. 8

означен с  $m$  (общият брой на агрегиращите атрибути  $n = 5$ ; разгледани са два случая за брой категории -  $C = 3$  и  $C = 4$ ). За  $m = 1$  (всеки атрибут е в отделна дименсия) моделът се свежда до ВММ, а при  $m = 5$  (всички атрибути са обединени в обща дименсия) моделът преминава в еднодименсионен. На фиг. 9в е показано намалението на обема използвана памет от IMM спрямо ВММ чрез коефициент на редукция на паметта. Вижда се, че този коефициент нараства бързо с увеличаване на броя на обединените в обща дименсия атрибути и е по-висок при по-голям брой категории в дименсия. При еднодименсионния модел изискванията към памет-

та са най-малки (фиг. 9г). Зависимостта на паметта от броя на агрегиращите атрибути ( $n$ ) е линейна. Коефициентът на редукция на паметта за еднодименсионния модел спрямо ВММ е голям, като нараства с нарастването на броя на атрибутите и на категориите (фиг. 9д). За двуменсионния модел обемът на използваната памет заема междинно място между ВММ и еднодименсионния модел (фиг. 9е). Зависимостта на паметта от броя на агрегиращите атрибути ( $n$ ) е квадратична. Коефициентът на редукция на паметта за двуменсионния модел спрямо ВММ нараства с увеличаването на броя на атрибутите и на категориите (фиг. 9ж).



Фиг. 9

### 3. Реализиране на многодименсионните модели чрез SQL запитвания

Използването на многодименсионни модели на данните в приложенията за анализ на данните, включва две стъпки:

- създаване на многодименсионната база данни;
- визуализация на многодименсионната база данни.

Генерирането на многодименсионна БД въз основа на съществуваща реляционна БД може да се осъществи с помощта на OLAP сървер или програмно чрез SQL запитвания. Първият метод е по-лесен за изпълнение, но вторият позволява следващи оптимизации на модела. Средата за програмиране Borland Delphi 4+ притежава набор от компоненти, позволяващи създаването и визуализирането на многодименсионни БД. Ние използваме само средствата за визуализация на многодименсионни БД (TDecisionGrid, TDecisionGraph) на Delphi, а генерирането на многодименсионните БД осъществяваме програмно чрез SQL запитвания, което позволява оптимизирането на модела на данните по отношение на паметта. За съхраняването на многодименсионната БД използваме ROLAP подход, т.е. съхраняваме я във вид на таблица.

По-долу е представено SQL запитване, генериращо BMM въз основа на съществуваща реляционна БД. Приемаме, че оперативните данни за разглежданата система за продажба на коли се съхраняват в реляционна БД. За подобряване на вземането на решения, касаещи контрола, управлението и планирането на продажбите е необходимо системата да позволява анализ и обобщаване на данните по различни критерии (напр. сума на продажбите по градове, магазини, модел и цвят на колите). Изграждането на многодименсионна БД, позволяваща бърз и гъвкав анализ на данните по тези критерии, започва с определянето на анализиранията величина (променливата) в многодименсионната БД - сумата от продажбите (този атрибут е означен с A5). Агрегиращите атрибути са - "Град" (A1), "Магазин" (A2), "Модел" (A3) и "Цвят" (A4). Градовете и магазините образуват йерархия (всеки магазин се намира в даден град). За изчисляването на общата и частичните суми е необходимо въвеждането на фиктивна обобщаваща категория ('all') във всяка дименсия [8]. SQL запитването се състои от обединения (UNION) на  $2^n$  обобщени (GROUP BY) по различните агрегиращи атрибути данни (т.нар. кубоиди), където  $n$  е броят на дименсиите. За  $n = 4$  броят на елементите на SQL запитването е 16 (1 - за данните, обобщени по 4 дименсии (базов кубоид), 4 - за обобщаване по 3 дименсии, 6 - за обобщаване по 2 дименсии, 4 - за

обобщаване по 1 дименсия и 1 - за общата сума (празен кубоид)). Чрез SQL запитване от вида:

```
SELECT DISTINCT A1,A2,A3,A4, SUM(A5)
FROM REL_DB GROUP BY A1,A2,A3,A4
```

```
UNION /*DATA GROUPED
BY 4 DIMENSIONS*/
```

```
SELECT CAST( 'all' AS CHAR(30) ),A2,A3,A4,
SUM(A5) FROM REL_DB
```

```
GROUP BY A2,A3,A4 /*DATA GROUPED
BY 3 DIMENSIONS*/
```

```
UNION
```

```
SELECT CAST( 'all' AS CHAR(30) ), CAST( 'all'
AS CHAR(30) ), A3,A4, SUM(A5) FROM REL_DB
GROUP BY A3,A4 /*DATA GROUPED BY 2
DIMENSIONS*/
```

```
....
```

```
UNION
```

```
SELECT CAST( 'all' AS CHAR(30) ), CAST( 'all'
AS CHAR(30) ), CAST( 'all' AS CHAR(30) ),A4,
SUM(A5) FROM REL_DB
```

```
GROUP BY A4 /*DATA GROUPED
BY 1 DIMENSION*/
```

```
UNION
```

```
/*FOR TOTAL SUM*/
SELECT CAST( 'all' AS CHAR(30) ), CAST( 'all'
AS CHAR(30) ), CAST( 'all' AS CHAR(30) ),
CAST( 'all' AS CHAR(30) ), SUM(A5) FROM REL_DB
```

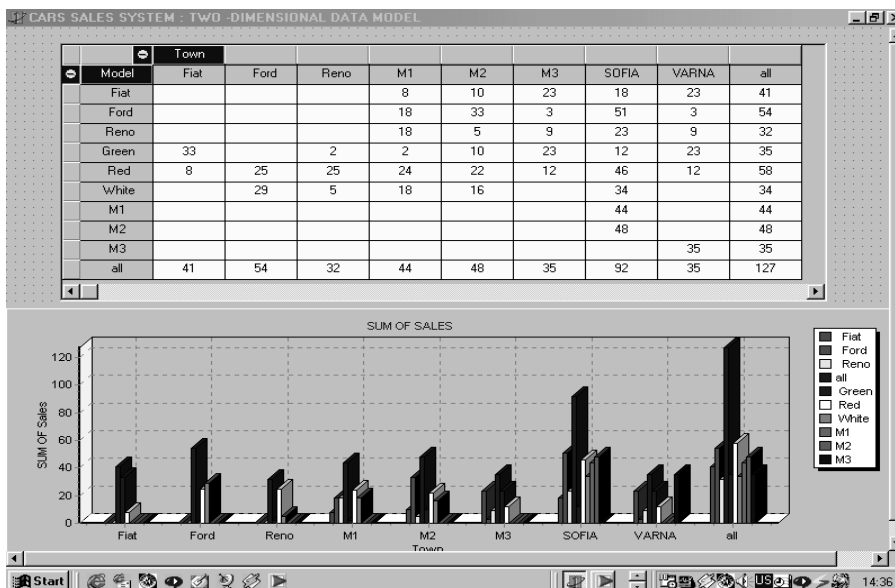
реляционният модел на данните се трансформира в BMM. Поради сложния реляционен модел SQL запитването обикновено включва свързване (join) на много таблици, което не е показано тук (с REL\_DB е означена резултатната таблица, получена при свързването на таблиците). Поради изискването за съвместимост на типовете във всяка дименсия използваме операцията CAST('all' AS CHAR(30)), предполагайки, че стойностите на атрибутите са символни низове с дължина 30. В получения многодименсионен куб има множество празни клетки. (напр. клетките, които са пресечни точки между даден магазин с градове, към които той не принадлежи) (фиг. 6).

Обединяването на йерархично свързаните атрибути "Град" (A1) и "Магазин" (A2) в обща дименсия намалява броя на празните клетки и на паметта, използвана от многодименсионната БД (фиг.10). 4-дименсионната БД се свежда до 3-дименсионна. Чрез SQL запитвания са реализирани и еднодименсионният и 2-дименсионният модел на данните. Двудименсионната БД е показана таблично и графично на фиг. 11.

CARS SALES SYSTEM : IMPROVED MULTIDIMENSIONAL DATA MODEL

		Town						
Model	Color	M1	M2	M3	SOFIA	VARNA	all	
Fiat	Green		10	23	10	23	33	
	Red	8			8		8	
	all	8	10	23	18	23	41	
Ford	Red		22	3	22	3	25	
	White	18	11		29		29	
	all	18	33	3	51	3	54	
Reno	Green	2			2		2	
	Red	16		9	16	9	25	
	White		5		5		5	
	all	18	5	9	23	9	32	
all	Green	2	10	23	12	23	35	
	Red	24	22	12	46	12	58	
	White	18	16		34		34	
	all	44	48	35	92	35	127	

Фиг. 10



Фиг. 11

**4. Заключение**

Технологията OLAP и многодименсионните бази данни позволяват бърз и гъвкав анализ на данните. OLAP технологията е особено ефективна в системите, подпомагащи вземането на решения, работещи с големи БД. Алгоритмите, трансформиращи реляционни в многодименсионни модели на данните могат да намерят приложение при генериране на многодименсионни БД въз основа на съществуващи реляционни БД. Някои информационни системи изискват съвместно използване на реляционна и многодименсионна БД, които да под-

помагат вземането на решения на различните нива на управлението - операционно, тактическо и стратегическо. Операционното равнище на управлението, изисква силно детайлизирана, точна информация за настоящия момент от време, която се осигурява от операционна реляционна БД. Двете по-горни нива на управление - тактическо и стратегическо - са свързани с вземане на решения, изискващи обобщена по определени критерии информация за даден период от време, за която е подходящо представянето като многодименсионна БД.

Поради големия обем памет, използвана от

многодимензионните бази данни, търсенето на методи за намаляване на паметта е важна задача за разработчиците на бизнес информационни системи.

Математическите изчисления и практическите резултати показват, че обединяването на агрегиращи атрибути е ефективен метод за намаляване на броя на димензиите и оттам - размера на многодимензионните бази от данни.

#### Литература

1. Цанкова Р., (Дисертация), "Инф. моделиране в системи за обработка на данни и управление", София, 2000 г.
2. Къртис Г., Бизнес-информационни системи, СИК "Иван Вазов", 1995
3. Oracle OLAP Products: Adding Value to the Data Warehouse, September 1995
4. An Introduction to OLAP, Multidimensional Terminology and Technology, White Paper, Pilot Software inc., 1993
5. Романски Р., "Компютърни технологии", ТУ - София, 1999
6. Deshpande P. et al., Cubing Algorithms, Storage Estimation and Storage and Processing Alternatives for OLAP", Bulletin of the IEEE Comp. Society Tech. Committee on Data Eng.", 1997
7. Borland Delphi 5.0.Help, Inprise Corporation, 1999
8. Gray J., A. Bosworth, A. Layman, H. Pirahesh, Data Cube: A Relational Aggregation Operator Generalizing Group\_By, Cross\_Tab, and Sub\_Totals, MS Research, Tech.Report, 1995

## MULTIDIMENSIONAL DATA MODELS IN BUSINESS INFORMATION SYSTEM

**Mila Nejкова, Rumiana Cankova**  
**Technical University – Sofia, Bulgaria**

### SUMMARY

Building information systems as decision support systems (DSS) requires fast and flexible on line data analysis. Business queries usually have multidimensional structure and require data presentation from different users perspectives. OLAP (On Line Analytical Processing) technology, based on multidimensional databases, is suitable for the needs of hierarchical data analysis and aggregation, especially in large databases. High levels of data aggregation and low response time are the main advantages of OLAP technology. Taking into account that the main disadvantage of multidimensional databases is the high memory consumption, we propose ways to improve multidimensional data models regarding their storage size. A technology for transformation of the relational data model to multidimensional data models is presented. The discussed multidimensional data models are evaluated by their memory consumption and their functionality. Multidimensional models are generated programatically by SQL queries.