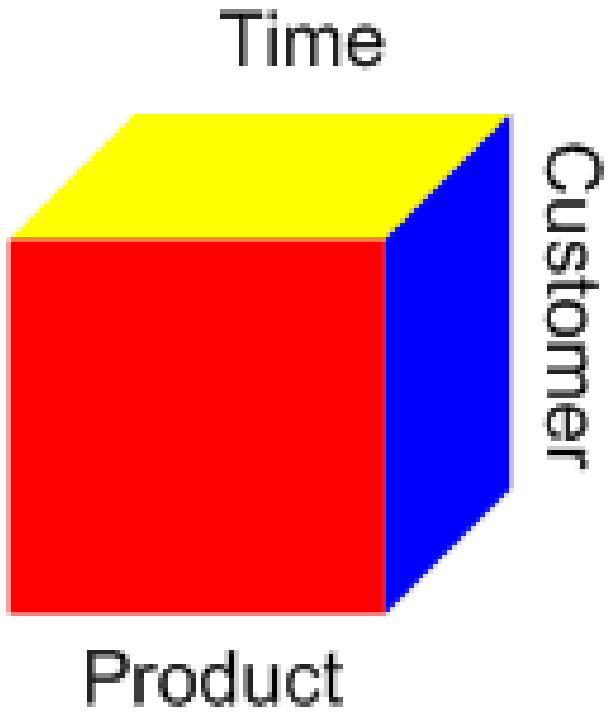
Анализ на куб

Кубове с данни и OLAP

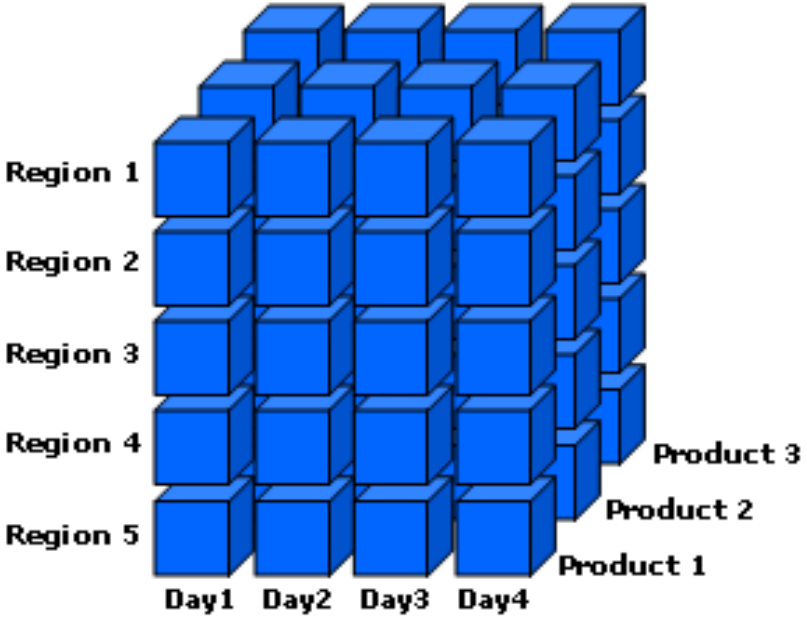
Куб

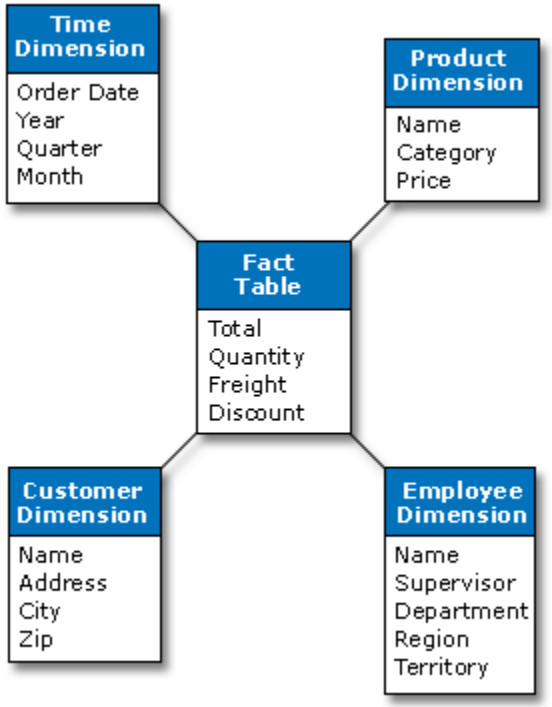


Кубове и OLAP

- Куб – множество от факти и добре дефинирани димензии
  - фактите са измерени или изчислени величини – мерки от по-ниско концептуално ниво на по-високо ниво на абстракция
  - димензиите съдържат йерархии
    - напр. обект Client може да съдържа Address, който да съдържа Country, City, Street и т.н.
- Предварителен анализ на многомерни данни
- OLAP - бизнес - интелигентна технология за анализ на данни, съхранявани в кубове

Кубове и OLAP

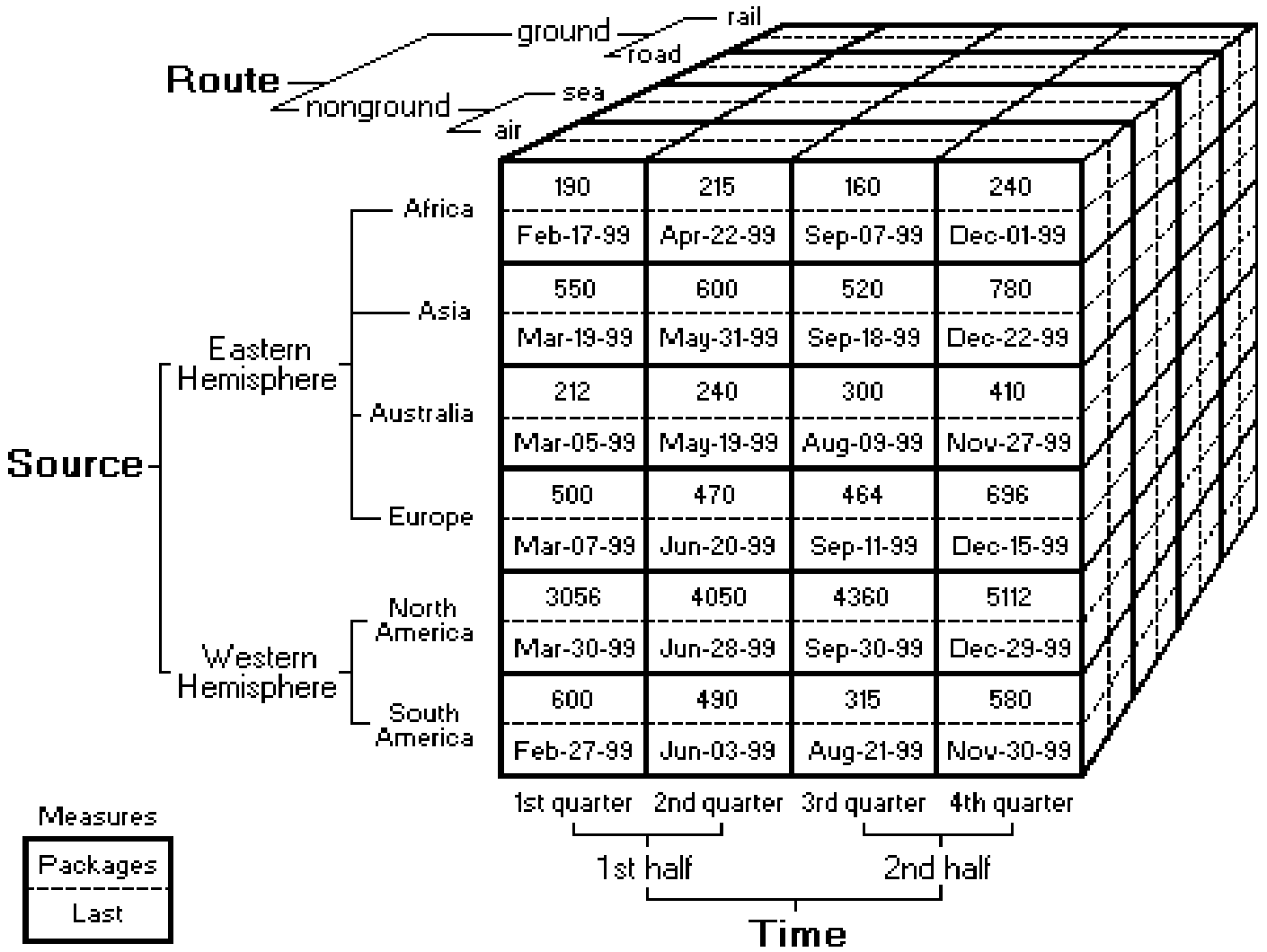




OLAP

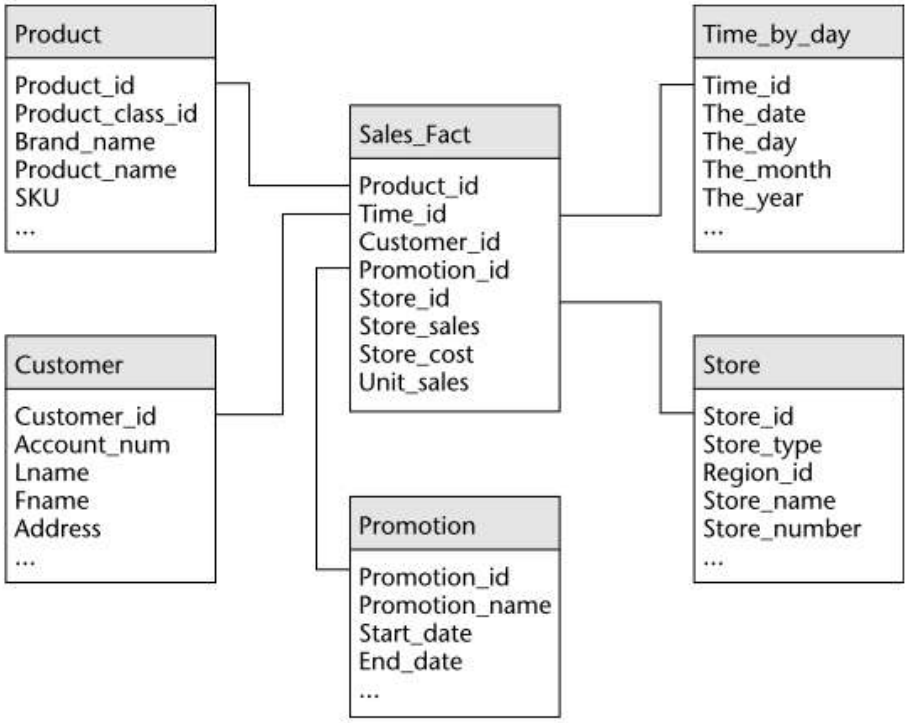
- OLAP агрегира фактите според йерархиите и ги съхранява в кубове
- OLAP съдържа кубове в база данни, както RDB съдържа таблици
- Използва се за подпомагане на вземането на решения в реално време
- Старото наименование на технологията е Decision Support System

Друг пример


Route
ground
rail
road
nonground
sea
air
Source
Eastern Hemisphere
Western Hemisphere
Africa
Asia
Australia
Europe
North America
South America
190
Feb-17-99
215
Apr-22-99
160
Sep-07-99
240
Dec-01-99
550
Mar-19-99
600
May-31-99
520
Sep-18-99
780
Dec-22-99
212
Mar-05-99
240
May-19-99
300
Aug-09-99
410
Nov-27-99
500
Mar-07-99
470
Jun-20-99
464
Sep-11-99
696
Dec-15-99
3056
Mar-30-99
4050
Jun-28-99
4360
Sep-30-99
5112
Dec-29-99
600
Feb-27-99
490
Jun-03-99
315
Aug-21-99
580
Nov-30-99
1st quarter
2nd quarter
3rd quarter
4th quarter
1st half
2nd half
Time
Measures
Packages
Last

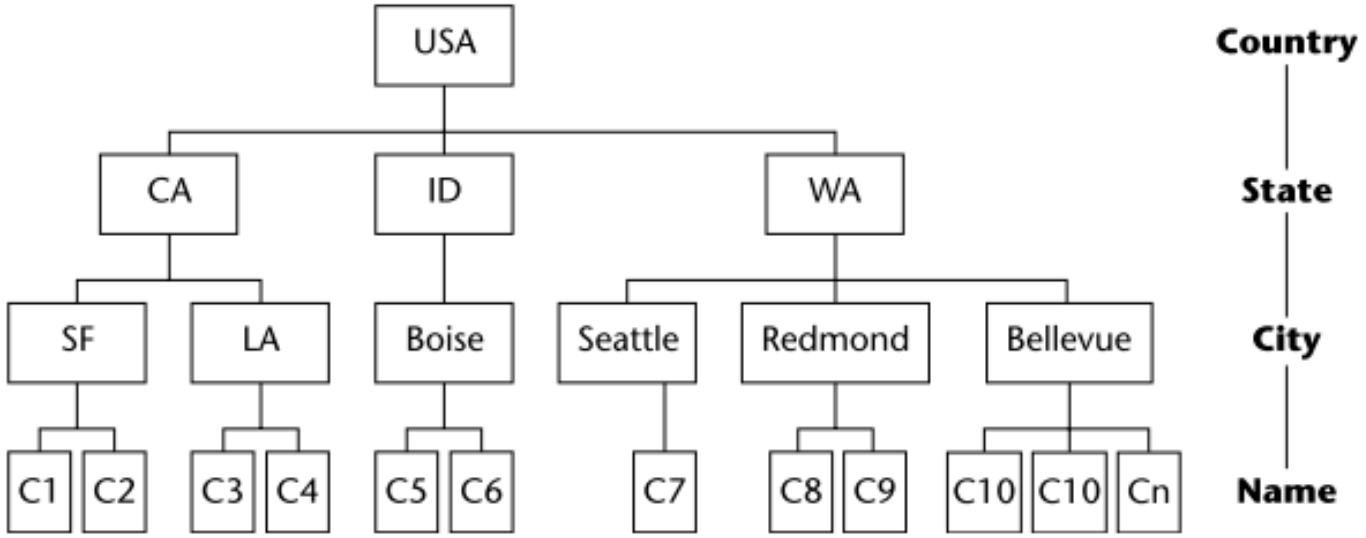
Кубове и схеми

- Факти и димензии
  - look-up таблици – за нормализиране на димензиите (напр. образование: 1-основно; 2-средно и т.н.)



Йерархия на димензиите

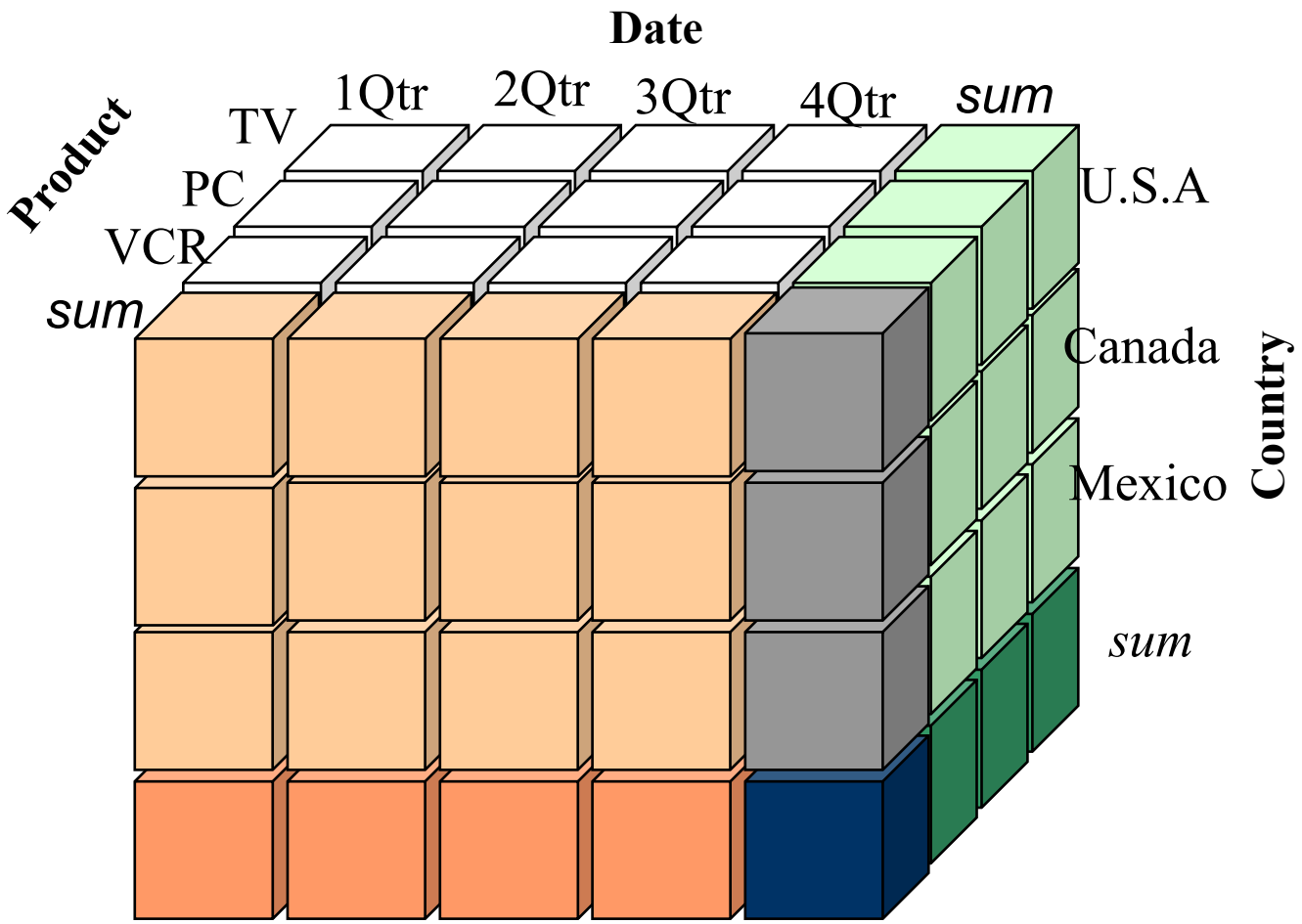
- Естествена – държава, град, улица, и т.н.
- Изкуствена – пол, образование, заплата
- Всяка йерархия има име
- Една димензия може да има много йерархии



Агрегиране на фактите

- Фактите се съхраняват в колони на факт-таблицата
- Фактите се агрегират по различните нива на йерархиите
- Функцията на агрегиране определя как мерките на фактите формират агрегирана стойност
- Агрегираната стойност е стойността, която се предава на родителя в йерархията
- Типични функции на агрегиране са Sum, Min, Max, Average, DistinctCount

Агрегиране на фактите



Операции с куб

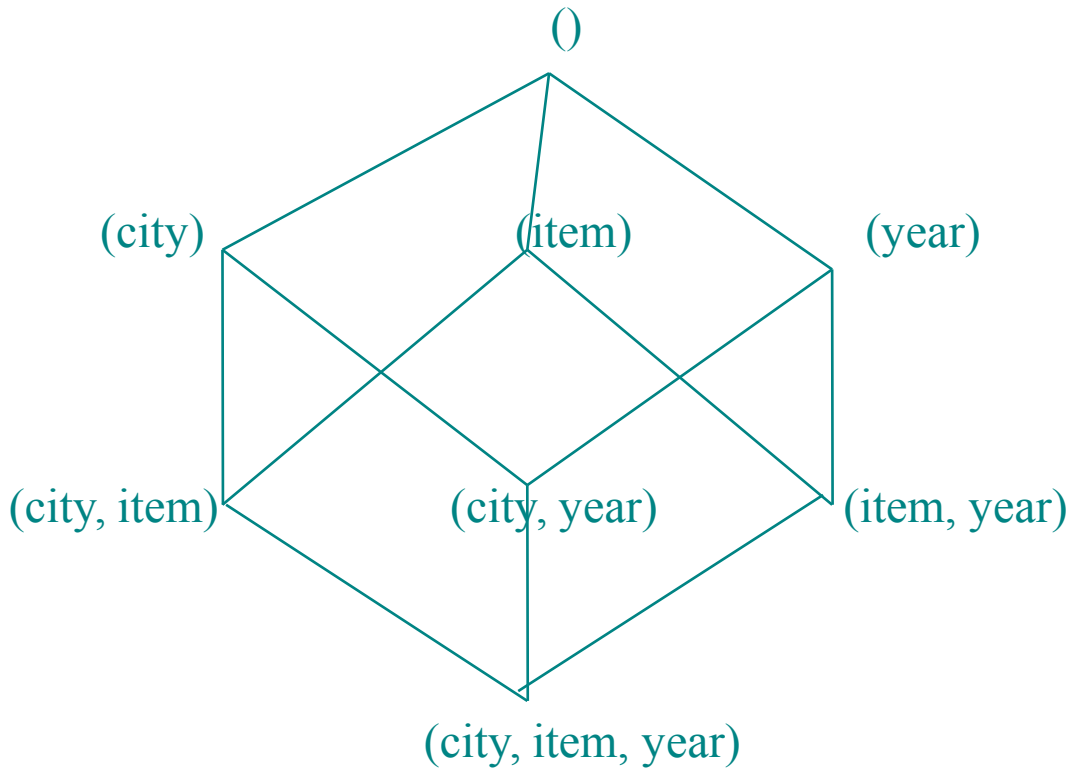
- Операции с димeнсиите
  - четене на димeнсионните таблици
  - създаване на структурата на димeнсиите
  - създаване на йерархии
  - прилагане на данни към подходящите нива на йерархиите
- Операции с целия куб
  - преизчисляване на агрегациите според йерархиите
    - общият брой на йерархиите е експоненциална функция на димeнсиите и йерархиите => необходима е селекция на агрегации за изчисляване
    - останалите агрегации могат да бъдат получени от избраните
    - пример: сумата на продажби по месеци може да произведе сумата по години

Операции с куб

- Два вида операции с димeнсиите и кубовете:
  - пълни
  - инкрементални – постъпкови
- Кубът може да се раздели на дялове, които се обработват поотделно и последователно, във времето на възникването им => така се оптимизира цялата обработка
- Индекси за улесняване на достъпа до клетките

Изчисляване на куб

item, *city*, *year*, and *sales_in_Euro*



0-D (*apex*) cuboid
една стойност за всички факти

1-D cuboids

2-D cuboids

3-D (*base*) cuboid
всички факти с по три димeнсии

Изчисляване на куб

- Кубът се представя като структура от 2^n кубоиди
  - всеки кубоид: group by
  - 0-D – една стойност, агрегираща всички данни, мярка на целия куб – base cell
  - drill down to1-D - aggregate cell
- Броят се увеличава, ако има йерархии на димeнсиите
  - брой кубоиди в n-D куб с L нива на йерархия

$$T = \prod_{i=1}^{n} (L_i + 1)$$

Материализиране на куб

- Изчисляване
  - пълна материализация - всеки кубоид
  - частична материализация – под-множество от кубоидите
    - избор на кубоиди за материализиране

Брой кубоиди - пример

- Ако всяка димензия има по 2 нива на йерархия:
  - Item(part, color) → i_1, i_2
  - City(downtown, suburb) → c_1, c_2
  - Year(good_year, bad_year) → y_1, y_2
- За 3-D куб
  - *Li* – брой на нивата на димензия I ($L_{1,2,3}=2$)
  - Общ брой на кубоидите

$$T = \prod_{i=1}^{3} (2+1) = 3*3*3 = 27$$

Множество на кубоидите

{

(),

(c1),(c2),(i1),(i2),(y1),(y2),

(c1,i1),(c1,i2),(c2,i1),(c2,i2),

(c1,y1),(c1,y2),(c2,y1),(c2,y2),

(i1,y1),(i1,y2),(i2,y1),(i2,y2),

(c1,i1,y1),(c1,i1,y2),(c1,i2,y1),(c1,i2,y2),

(c2,i1,y1),(c2,i1,y2),(c2,i2,y1),(c2,i2,y2)

}

Изчисляване на куб

- DMQL Data Mining Query Language for Relational Databases [Han et al, Simon Fraser University]
- За създаване на модели за анализ с SQL-базиран интерфейс ("Command-driven" data mining)
- Дефиниране на фактите и димeнсиите

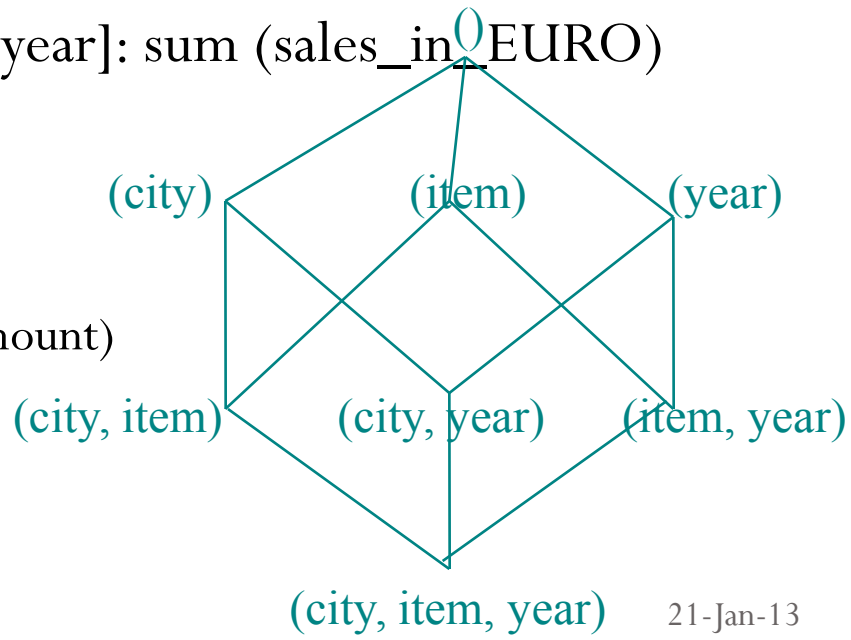
define cube sales [item, city, year]: sum (sales_in_EURO)

compute cube sales

SELECT item, city, year, SUM (amount)

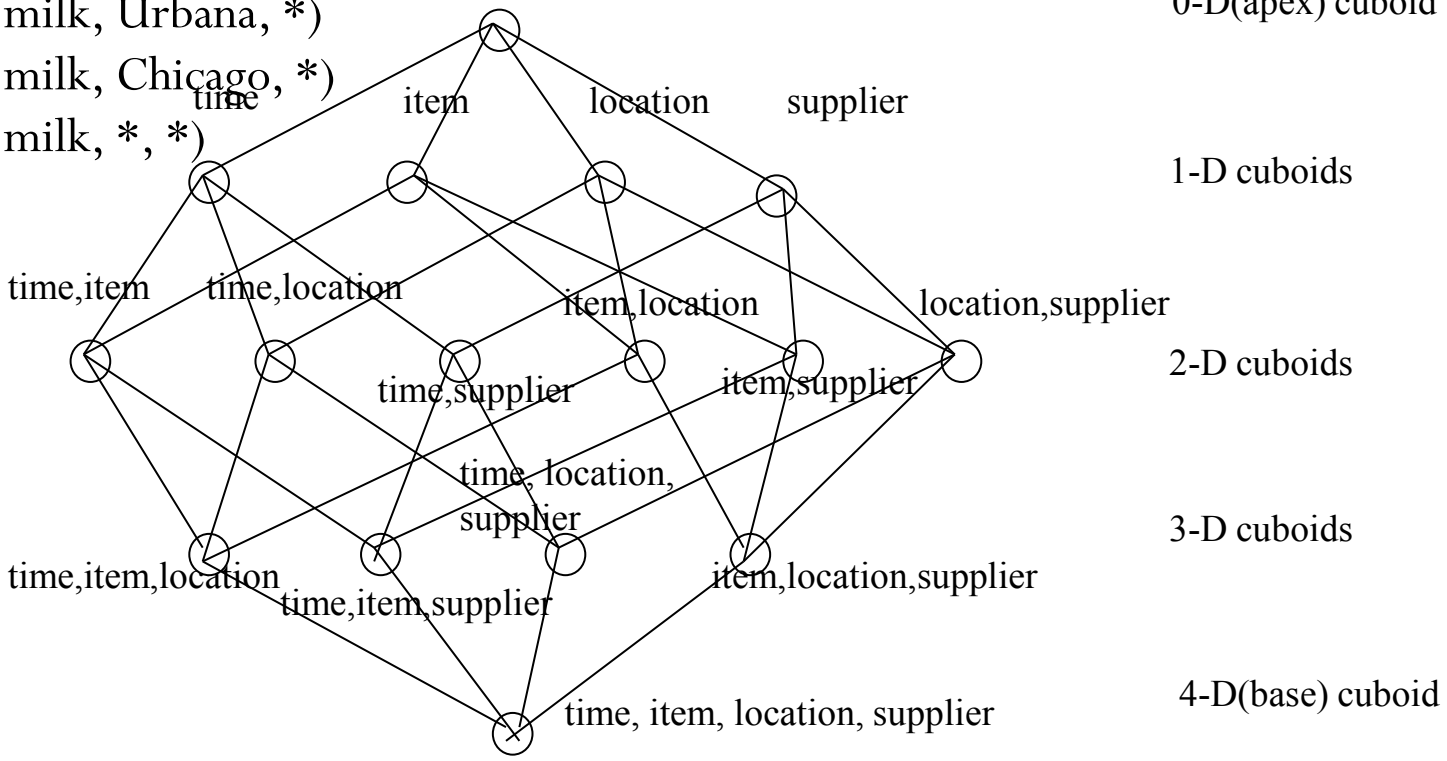
FROM SALES

CUBE BY item, city, year



Нотации

- Base vs. aggregate cells; ancestor vs. descendant cells; parent vs. child cells
  1. (9/15, milk, Urbana, Dairy_land)
  2. (9/15, milk, Urbana, *)
  3. (*, milk, Urbana, *)
  4. (*, milk, Urbana, *)
  5. (*, milk, Chicago, *)
  6. (*, milk, *, *)



Iceberg

- Full cube vs. iceberg cube

```
compute cube sales iceberg as
select month, city, customer group, count(*)
from salesInfo
cube by month, city, customer group
having count(*) >= min support
```

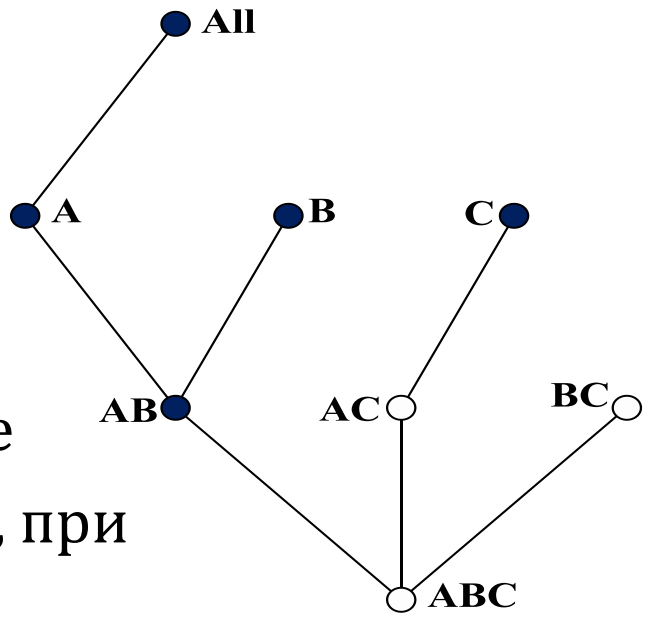
- Изчисляват се само тези клетки, които отговарят на условието
- Само малък брой клетки остават на повърхността
- Избягва се участието на неинформативни данни
- Пример
  - 2 base cells: (a1, a2, …., a100), (b1, b2, …, b100)
  - колко агрегации ще има,
    - ако “having count >= 1”?
    - ако “having count >= 2”?

Методи за изчисление на кубове

- Multi-Way Array Aggregation
- BUC Bottom Up Calculation
- Star-Cubing
- High-Dimensional OLAP
- и др.

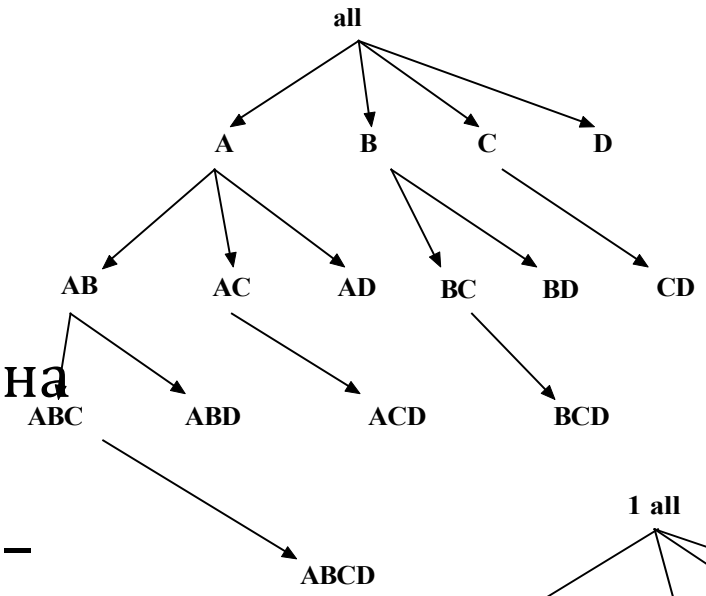
Multi-Way Array Aggregation

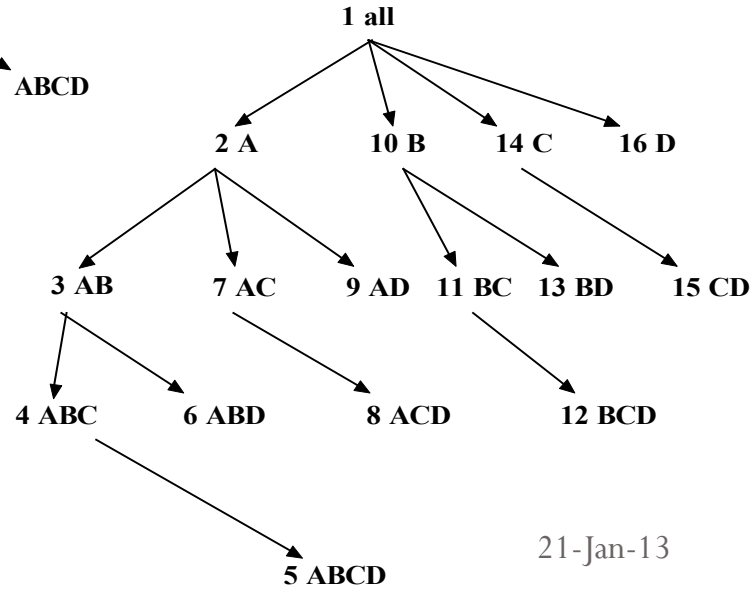
- Кубът се разделя на части
  - multi-dimensional chunks
- Отдолу-нагоре
- Едновременно агрегиране по множество димензии
- вече агрегираните стойности се използват на по-високото ниво, при родителите
- не може да се прави оптимизация на айсберг



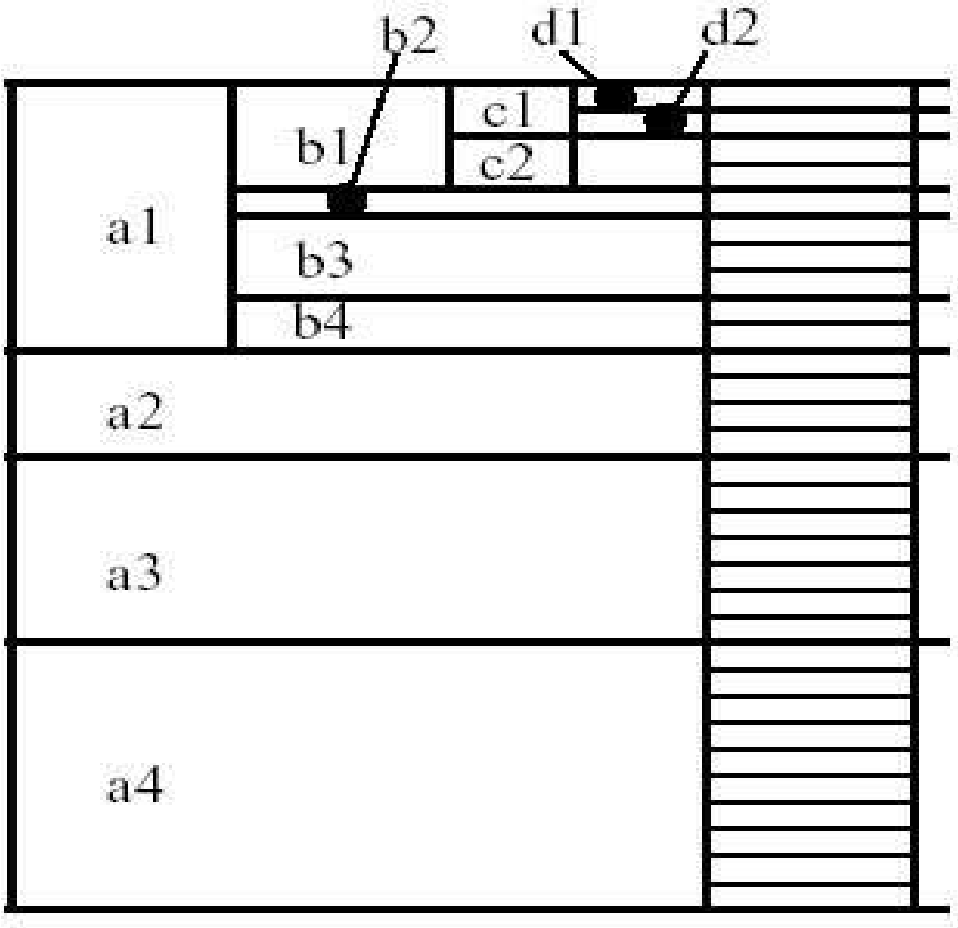
Bottom-Up Computation (BUC)

- BUC [Beyer & Ramakrishnan, SIGMOD'99]
- Bottom-up cube computation (top-down при нас)
- Разделяне на димензиите на части partitions
- Всяка част се оптимизира – айсберг
- Не позволява едновременно изчисляване на частите



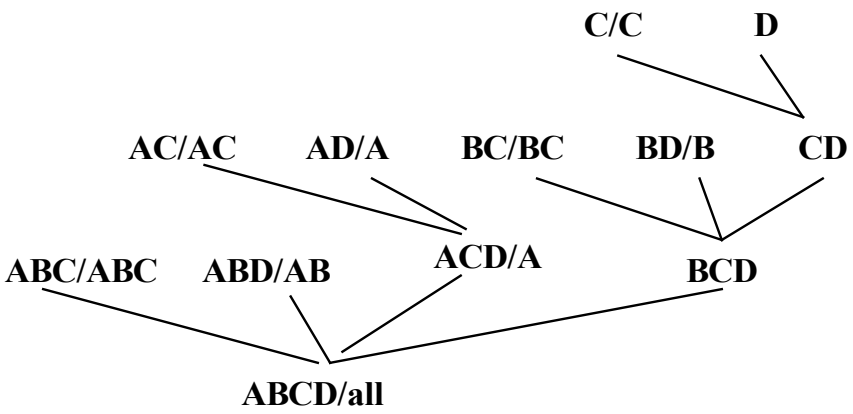


BUC: Partitioning



Star-Cubing

- [D. Xin, J. Han, X. Li, B. W. Wah, Star-Cubing: Computing Iceberg Cubes by Top-Down and Bottom-Up Integration, VLDB'03]
- Споделени димензии
  - напр. димензия A е обща за ACD и AD
  - ABD/AB означава ABD има споделена димензия AB
- Позволява споделени изчисления
  - напр. AB се изчислява едновременно с ABD



Редукция на неинформативните данни

A	B	C	D	Count
a1	b1	*	*	1
a1	b1	*	*	1
a1	*	*	*	1
a2	*	c3	d4	1
a2	*	c3	d4	1

- Ако minsup = 2
- Всички стойности count < 2 се заместват със * и се събират заедно

A	B	C	D	Count
a1	b1	*	*	2
a1	*	*	*	1
a2	*	c3	d4	2

Извличане на информация от куб

- MDX Multi Dimensional Expressions
- Език за формулиране на заявки за извличане на информация от куб
- Подобен на SQL
  - SELECT … FROM … WHERE …
  - работи с многомерни данни
  - SELECT селектира участващите димeнсии,
  - FROM показва куба, а
  - WHERE ограничава операцията до определена димeнсия или факти

Разлики

- MDX оператор
  - вход - кубоид
  - резултат – кубоид
- SQL
  - вход – таблици
  - изход – таблица
- Заявките в MDX имат дименсии (axes)
  - първите три са **rows**, **columns** и **pages**

Пример

```
Select
{Measures.[Unit Sales], Measures.[Store Sales]} on columns,
{Store.[Store Name].members} on rows
From Sales
Where Product.[All Products].Drink.Beverages
```

сумира продажбите и приходите от разхладителни напитки по магазини

Друг пример

```
SELECT
{[Time].[2010],[Time].[2011]} ON COLUMNS,
{[Measures].[Warehouse Sales],[Measures].[Warehouse Cost]}  ON
   ROWS
FROM Warehouse
WHERE  ([Store].[All Stores].[Bulgaria])
```

- Сумира продажбите и разходите за всички стоки, за определени години поотделно за всички магазини в България
- Продажбите и разходите са в редовете, а годините – в колони

Операции - изчисления

- Кубът съдържа факти и агрегатни данни, от които могат да се изчисляват **производни стойности**, напр.
  - **profit** = [sales] – [costs]
  - Агрегатните данни се изчисляват по време на формиране на куба, а изчислените – по време на изпълнение на заявката за извличане на информация
- Изчисляват се и **членове на йерархии** на дименсии, напр.

```
WITH
   MEMBER [Time].[1st Half Sales] AS 'Sum({[Time].[Q1], [Time].[Q2]})'
   MEMBER [Time].[2nd Half Sales] AS 'Sum({[Time].[Q3], [Time].[Q4]})'
SELECT
   {[Time].[1st Half Sales], [Time].[2nd Half Sales]} ON COLUMNS
FROM Sales
WHERE [Measures].[Store Sales]
```

- Могат да се правят изчисления и на ниво **клетки от куба**, напр. при обновяване на съдържанието

Схеми за съхраняване на агрегирани данни

- Relational OLAP (ROLAP)
  - данните от дяловете се записват в релационни таблици
  - предварително изчислените агрегации – също
  - търсенето в базата се осъществява в реално време
- Multidimensional OLAP (MOLAP)
  - данните и агрегациите се записват в специален формат за ефективно извличане
  - по-добро представяне на данните, но се изисква предварителна обработка на куба
- Hybrid OLAP (HOLAP)
  - данните се записват в релационни таблици
  - предварително изчислените агрегации – в специален формат

UDM Unified Data Modeling

- Метод за описание на куб с факти и димeнсии, призхождащи от разнородни източници и формати
- Позволява използване на унифициран модел на данни във всички системи на BI - склад за данни, куб, OLAP, отчети и др.
- Позволява дефиниране на йерархии като набор от атрибути
- Може да съдържа един или повече кубове, предварително агрегирани за по-бърза обработка в реално време
- Позволява допълнителна обработка на данните чрез MDX за генериране на нови информативни данни, като напр. средна стойност за тримесечие и др.
- Позволява визуализация на резултатите и интерактивна обработка

Архитектура на UDM в MS AS