Хранилища за данни

Data Warehouse

Съдържание

- **Хранилище за данни (*DW Data Warehouse*)**
  - Основни сведения, архитектури, приложения
  - Подмножества от данни - *Data Mart*
  - Многомерни модели на данните - *Data Cube*
- Технологии за извличане на информация от многомерни данни -
  - **OLAP** *(On-Line Analysis Processing)*
  - **OLDM** *(On-Line Data Mining)*

Определения

- Склад за генерализирани и консолидирани данни в многомерно пространство
- База от данни, предназначена за подпомагане на вземането на решения в бизнеса
- Технология за подготовка на данните за откриване на информация и знания
- Платформа за OLAP за интерактивен анализ на многомерни данни

Свойства

- Информационна дейност, съпътстваща основната - отделена от основните данни за операции или транзакции
- Предметно-ориентирана – организирана около някой от основните обекти на дейността, напр. Student, Plan, …
- Интегрираща множество хетерогенни ресурси
  - данните в нея са изчистени, трансформирани и интегрирани
- Дългосрочна - съдържа атрибут време и данни за история

Необходимост от отделяне

- Различна производителност - по-висока производителност на основната база от данни и хранилището за данни
- Различни характеристики на данните
  - данни за историята
  - консолидирани данни
  - по-високо качество на данните
- Различни структури на данните

Различия на DW от DB

- OLAP, не OLTP
- Ориентирана към бизнеса и печалбите, не към крайните потребители
- Съдържа трансформирани и интегрирани данни, с по-високо ниво на грануларност
- Изградена върху различни модели на данните, не само релационни ER
- Съдържа голямо количество разнородни данни от различни източници и периоди от време
- Поддържа главно read-only операции и упростени правила за достъп до данните

Различия на OLAP от OLTP

	OLTP	**OLAP**
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

Реализации на DW

- Enterprise Warehouse
  - съдържа всички данни за всички обекти от бизнеса
- Data Mart
  - подмножество от данните, за специфична операция или обекти
- Virtual Warehouse
  - множество от изгледи

Модели на данните

Модели на данните

- Многомерен модел на данните (multidimensional data model), представени като куб (data cube)
  - Моделът позволява данните да се разглеждат в различни проекции, напр. време, място, стока, отдел и др.
- Двумерен модел – матрица
  - пример: сезон – локация – брой наблюдения
  - две димензии (две координати): сезон и локация

	София	Пловдив	Варна
пролет	114	109	58
лято	543	432	654
есен	34	45	56
зима	54	43	32

Димензионни таблици

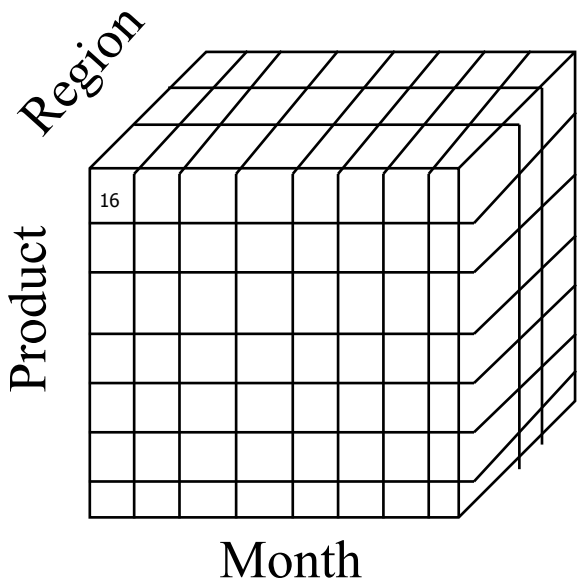
- Всяка димензия може да има асоциирана таблица с множество допустими стойности (**dimension table**)
  - примери:
    - city (name, country, number of citizens, geographical location)
    - time(day, week, month, quarter, year)
- димензионните таблици се конфигурират от експерти или автоматично, според данните

Таблици с факти

- Всеки многомерен модел е създаден във връзка с някаква страна на бизнеса, напр. продажби
- Измерването на бизнеса генерира регистрирани стойности, количества – факти
- Примери:
  - брой продадени стоки, количество на приход и др.
- Таблица с факти **(fact table) -** таблици с данни от измерванията и ключове към свързаните с данните дименсионални таблици

Многомерен модел

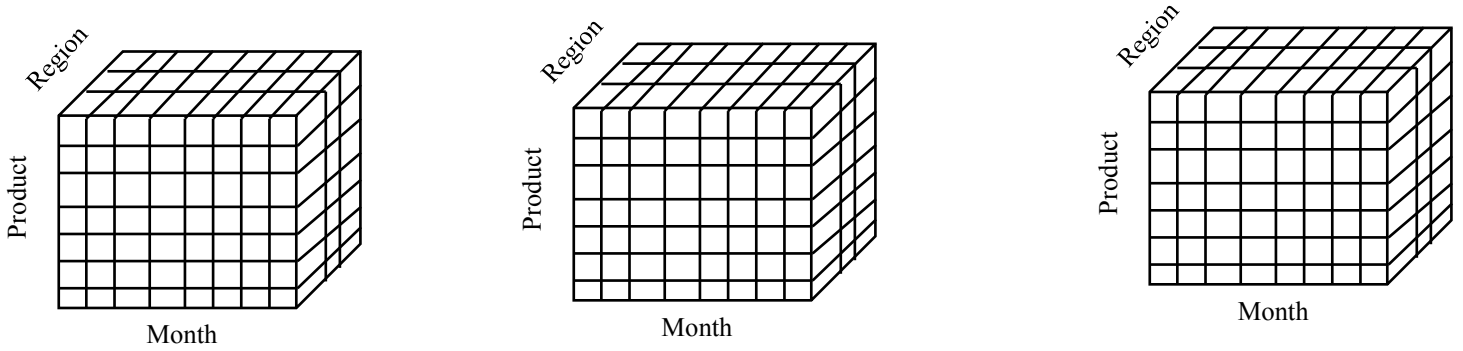
- Тримерен модел - куб
- Димензии: *Product, Location, Time*
  - може да съществуват димензионни таблици за дефинирането им
- Факти: количества записани във всяка клетка поотделно



Многомерен модел

- Четиримерен модел
  - напр. добавяне на данни и за различни производители
  - може да се разглежда като множество от тримерни кубове

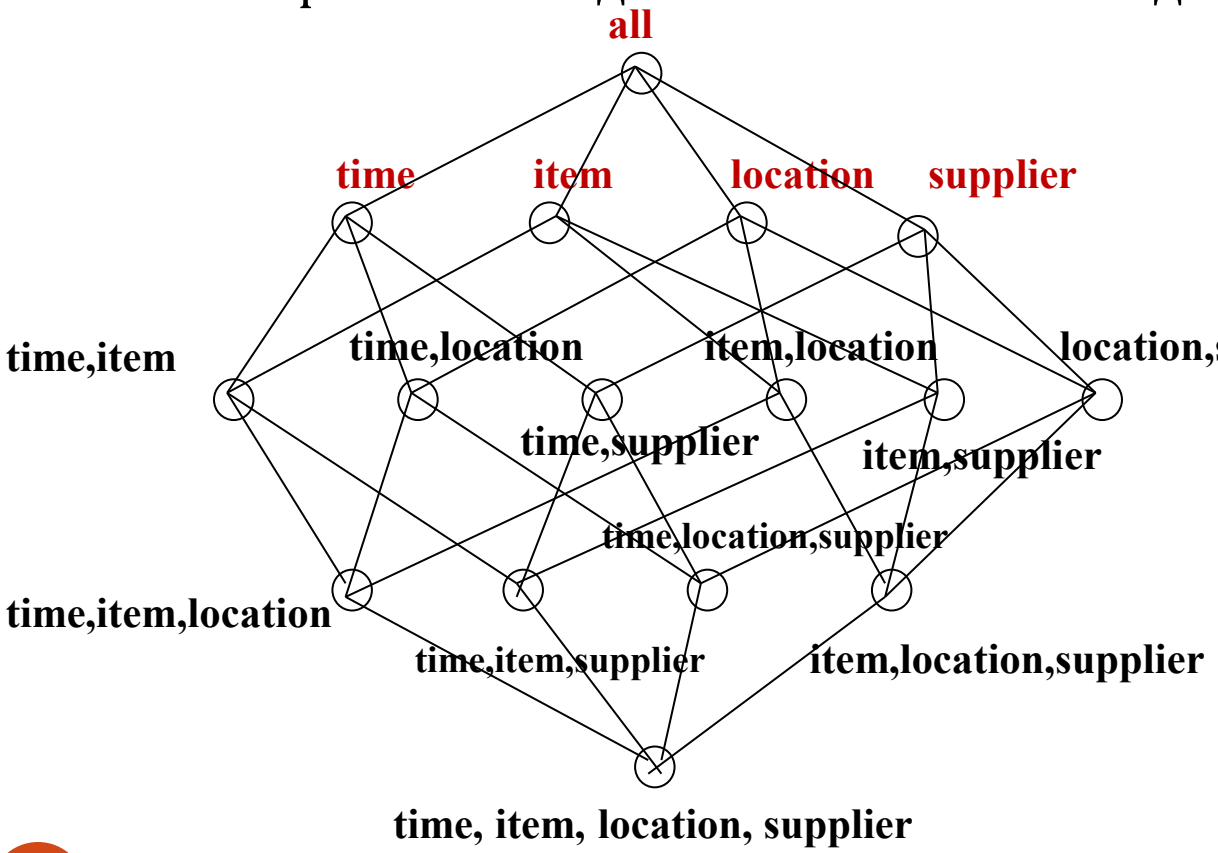
P1 P2 P3



=> n-D може да се разглежда като множество от (n-1) – D кубове

Кубоид

- Куб с различна степен на агрегиране, сумиране
  - различно подмножество от всички димензии (group by)



0-D (*apex*) cuboid
общ брой продажби

1-D cuboids
суми по категория

2-D cuboids
подмножества

3-D cuboids

4-D (*base*) cuboid

Концептуални модели

- Операционни бази от данни – релационен модел
- Многомерни DW – други модели
  - Схема звезда
  - Схема снежинка
  - Схема галактика

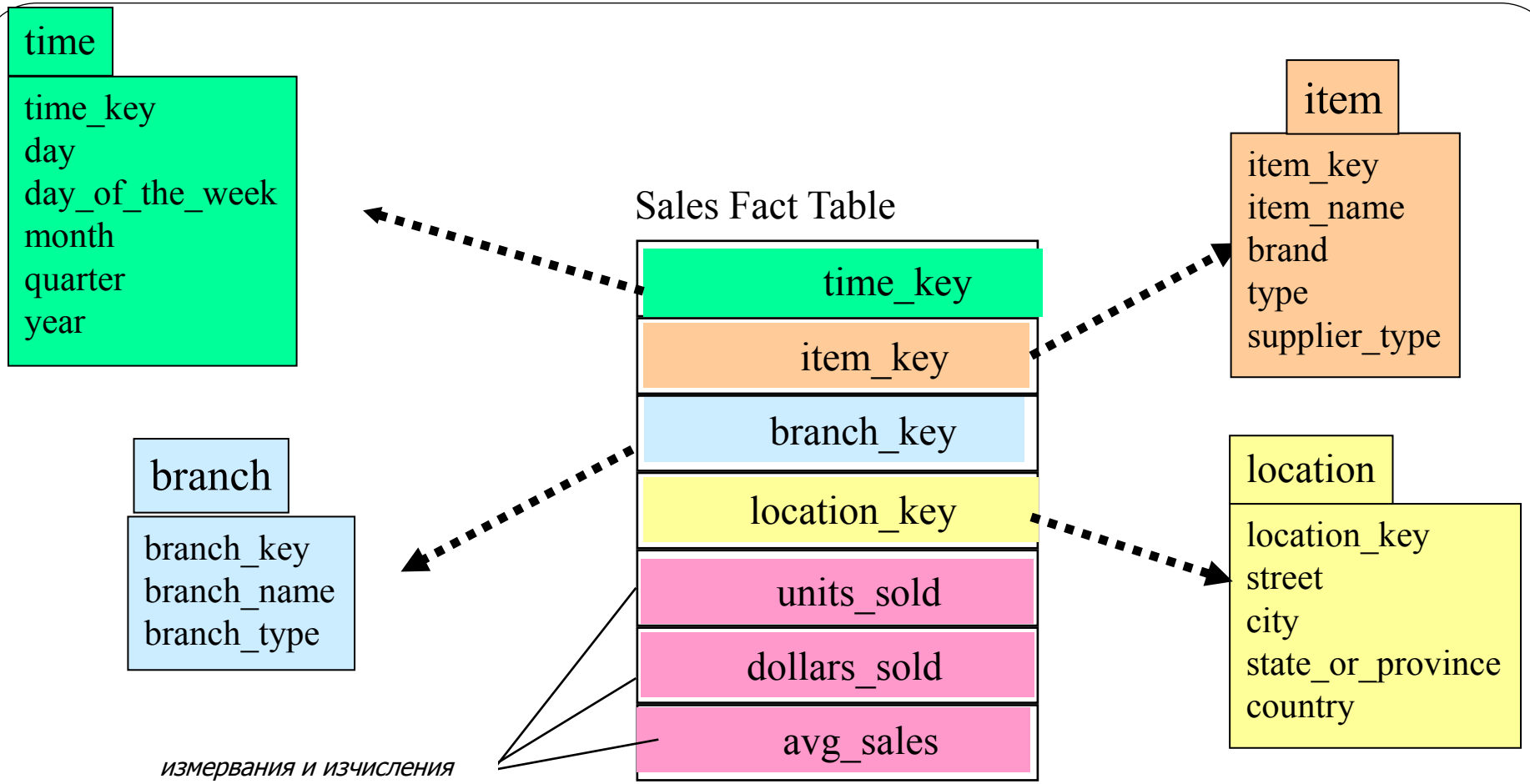


Схема звезда

- една централна таблица с данни (fact table)
- множество таблици за димeнсиите (dimension tables)
- недостатък: предпоставки за повторение на описанието на димeнсии

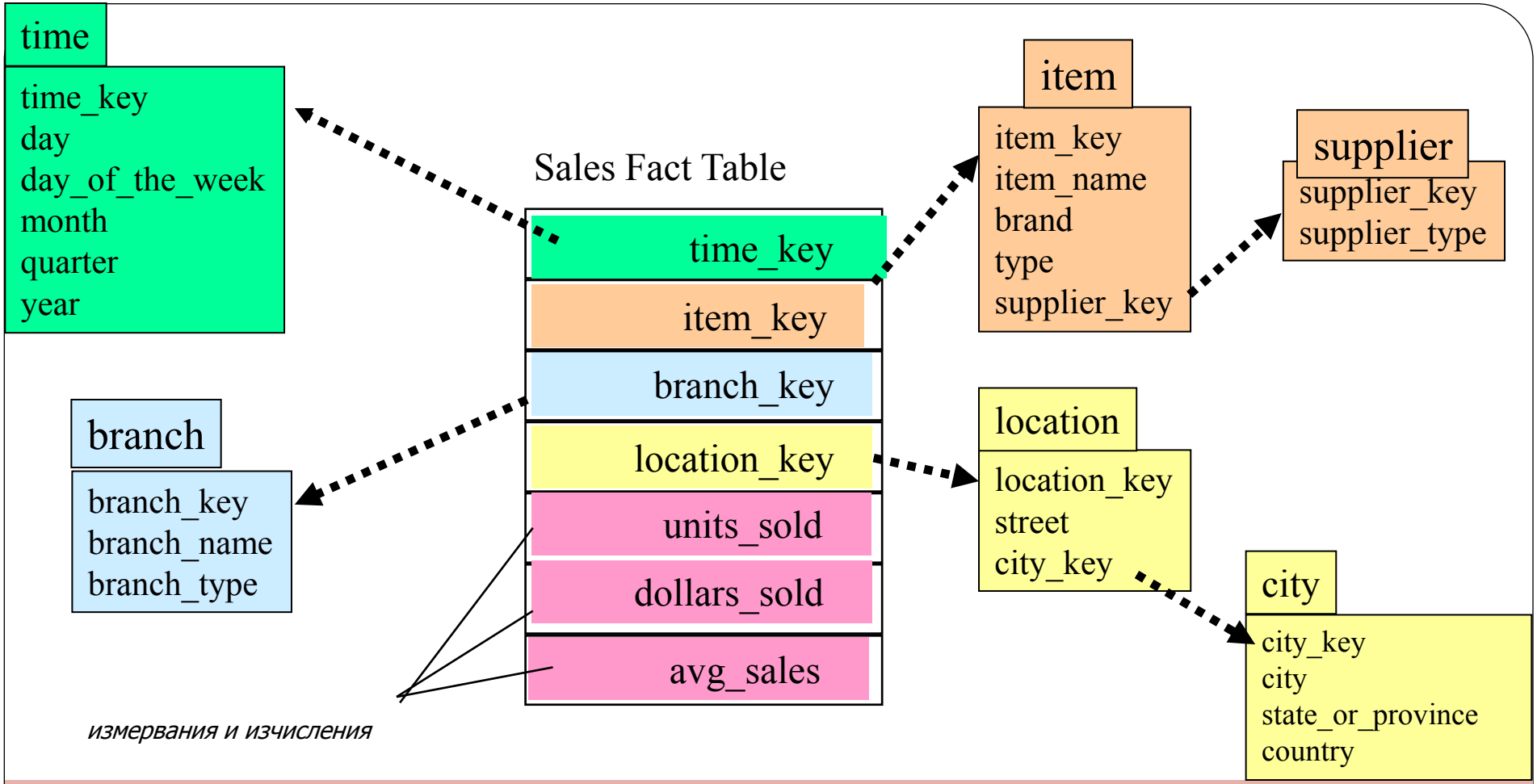


Схема снежинка

- схема звезда, в която някоя от димeнсиите е нормализирана в множество по-малки димeнсии
- предимство: избягва се дублирането в описание на димeнсии

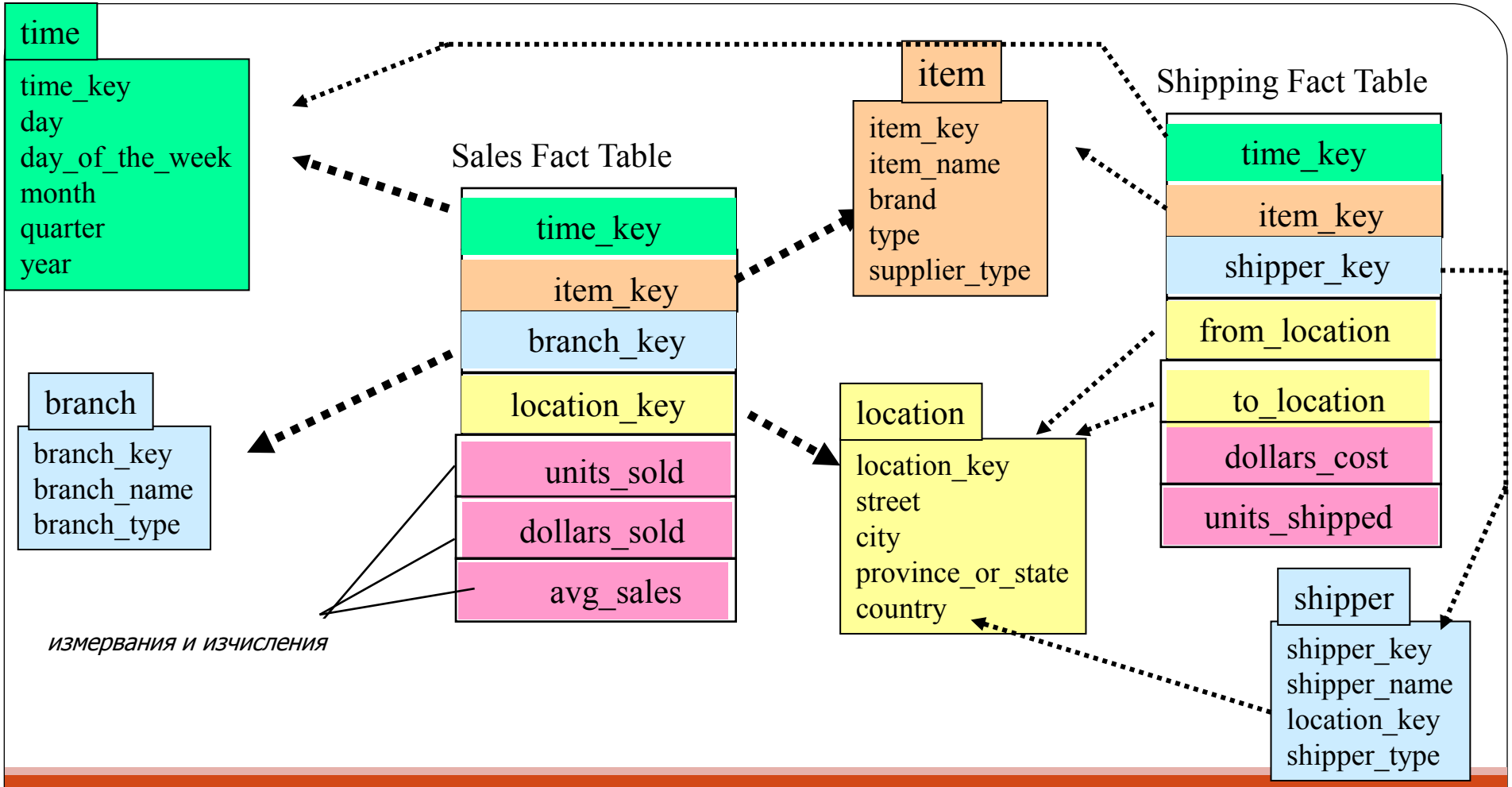


Схема галактика

- множество таблици с данни, споделящи едни и същи димензионни таблици
- колекция от звезди
- подходящ за голяма DW

Data Mart

Data Mart

- Подмножество на Data Warehouse
- Данни на по-ниско ниво от структурата на бизнеса, напр. отдел
- Подходящи модели
  - звезда
  - снежинка

Дефиниране на схеми

- Език за дефиниране
  - Data Mining Query Language DMQL
  - SQL based
- Две нива на дефиниране
  - дефиниране на куб
  - дефиниране на димeнсии
- Синтаксис

  define cube <cube name> [<dimensions list>]: <measure list>

  define dimension <dimension name> as (<attribute or dimension list>)
- Примери

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

  define dimension time as (timekey, day, month, quarter, year)

Дефиниране на схема Звезда

```
define cube sales_star [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales =
          avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week,
   month, quarter, year)
define dimension item as (item_key, item_name, brand,
   type, supplier_type)
define dimension branch as (branch_key, branch_name,
   branch_type)
define dimension location as (location_key, street, city,
   province_or_state, country)
```

Дефиниране на схема Снежинка

```
define cube sales_snowflake [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales =
          avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month,
    quarter, year)
define dimension item as (item_key, item_name, brand, type,
    supplier(supplier_key, supplier_type))
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city(city_key,
    province_or_state, country))
```

Дефиниране на схема Галактика

```
define cube sales [time, item, branch, location]:
        dollars_sold = sum(sales_in_dollars), avg_sales =
          avg(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state,
    country)
define cube shipping [time, item, shipper, from_location, to_location]:
        dollar_cost = sum(cost_in_dollars), unit_shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as location
    in cube sales, shipper_type)
define dimension from_location as location in cube sales
define dimension to_location as location in cube sales
```

Мерки на куба

- Числова стойност на точка от многомерното пространство
  - множество двойки: димeнсия = стойност
  - напр. time=Q1, location="Sofia"
- Мярка на куба
  - числова функция, която може да се приложи на всяка точка
  - агрегираща функция
- Категории стойности
  - според вида на агрегиращата функция

Категории стойности

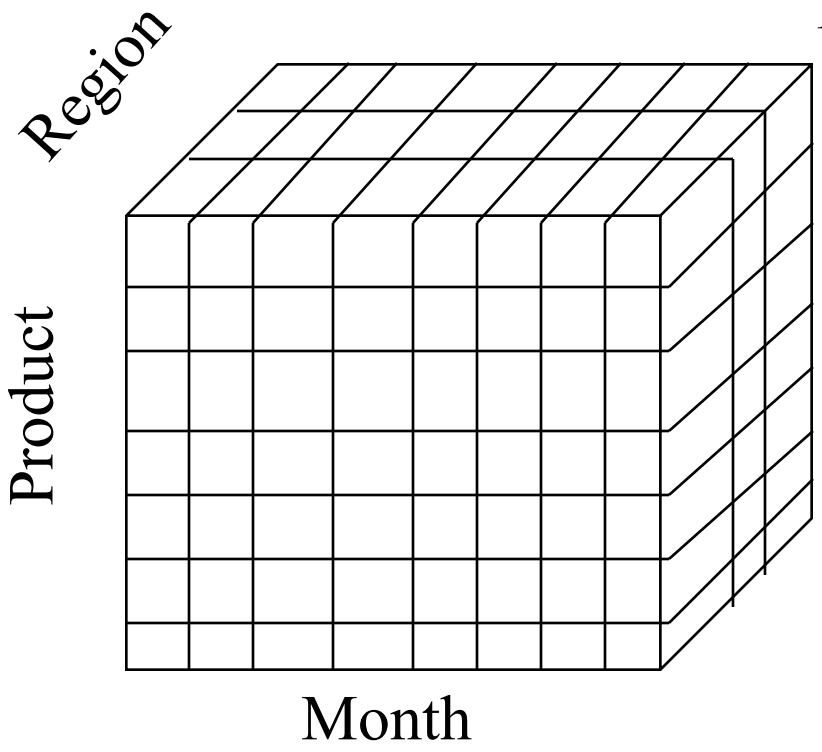
- Distributive
  - стойности, които могат да бъдат получени от агрегиране на други стойности, напр. count(), sum(), min(), max()
- Algebraic
  - които могат да се изчислят като алгебрична функция с *M* аргумента, всеки от които е получен чрез горните функции, напр. avg(), minN(), standarddeviation()
- Holistic
  - subaggregate – стойност, получена без участието на размера на множеството, напр. median(), mode(), rank()

Йерархия на димензиите

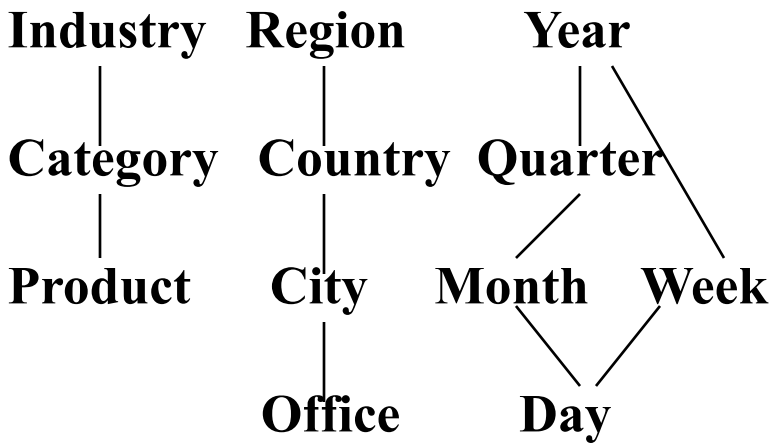
- Разлагане / агрегиране на димензиите в различни нива на грануларност
- Може да се дефинира предварително и прилага при изследванията на куба, на различни нива на абстракция
- Осигуряват се различни перспективи за изследване, чрез групиране на данните на различните нива на йерархиите

Многомерни данни

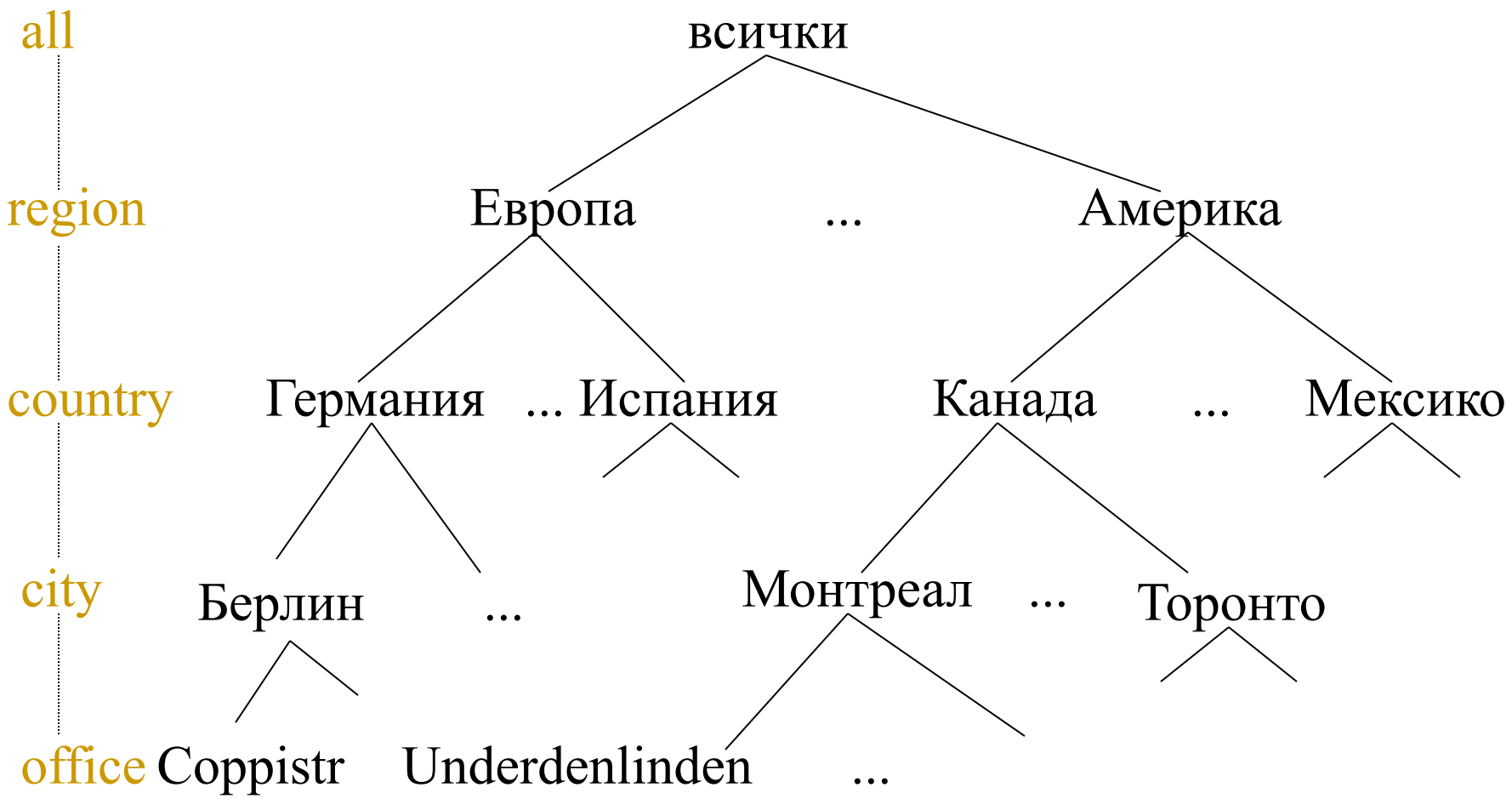
- Напр. продажбите, като функция на продукта, месеца и региона.



Dimensions: *Product, Location, Time*
Hierarchical summarization paths



Пример: йерархия на димензия location



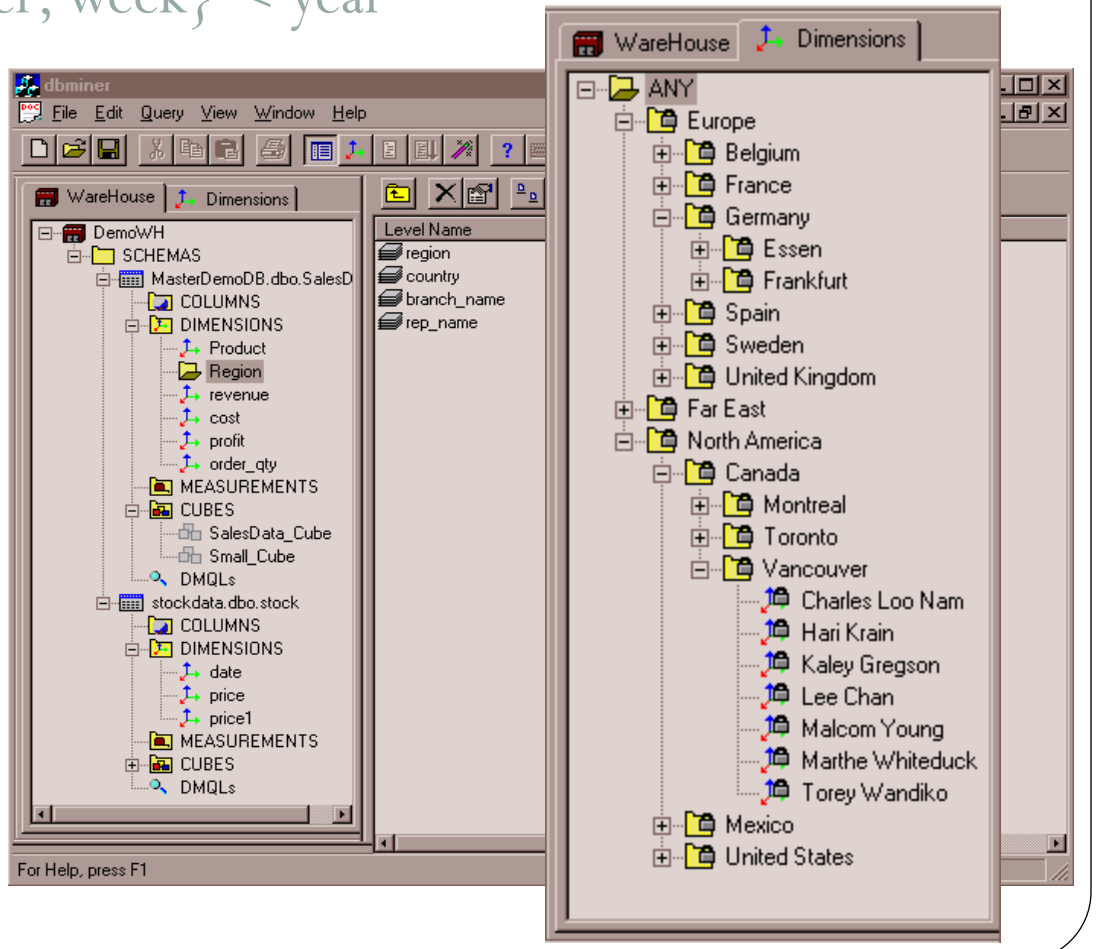
Други примери за йерархии

- Schema hierarchy

  day < {month < quarter; week} < year

- Set_grouping hierarchy

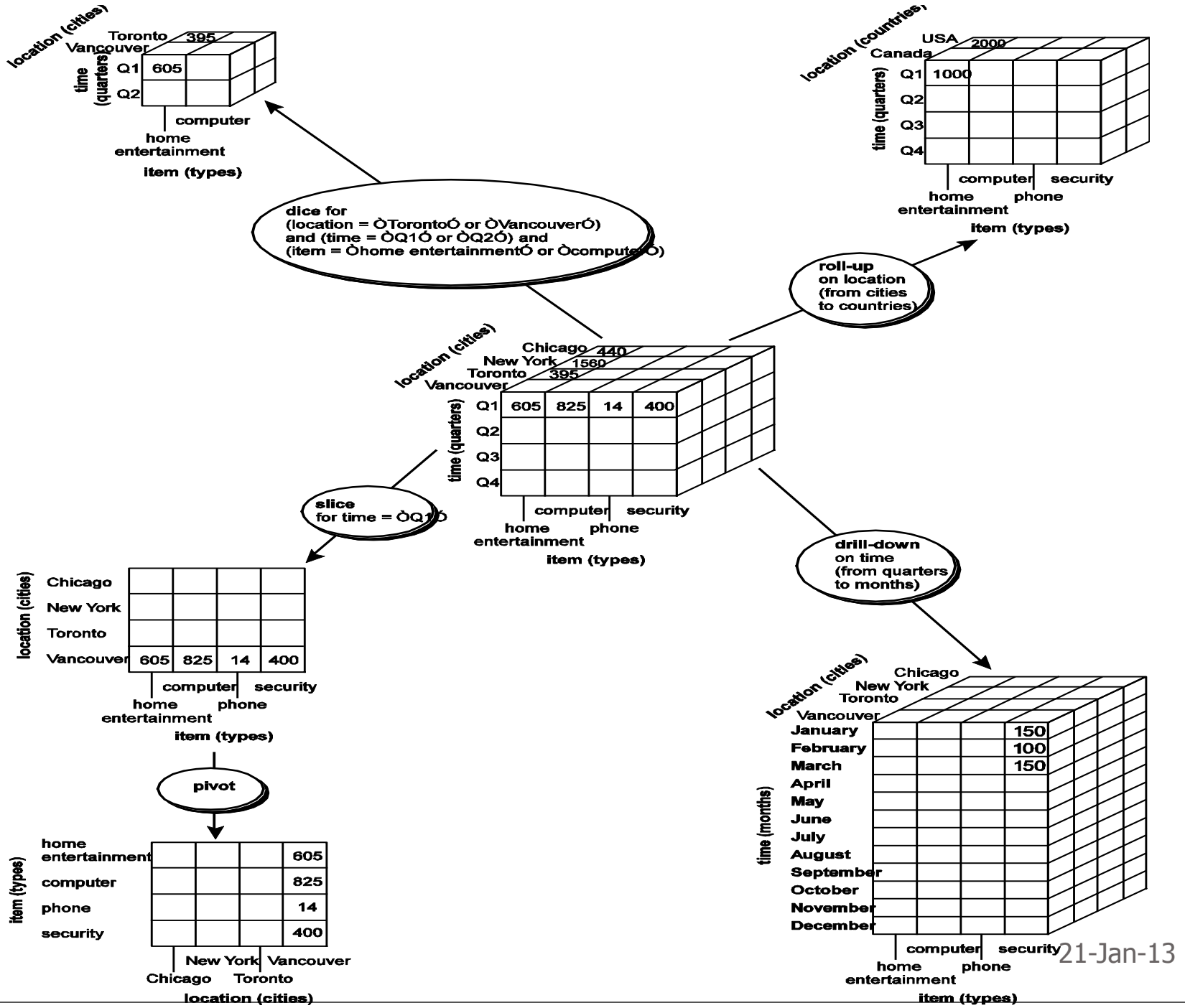
  {1..10} < inexpensive



Операции с куб

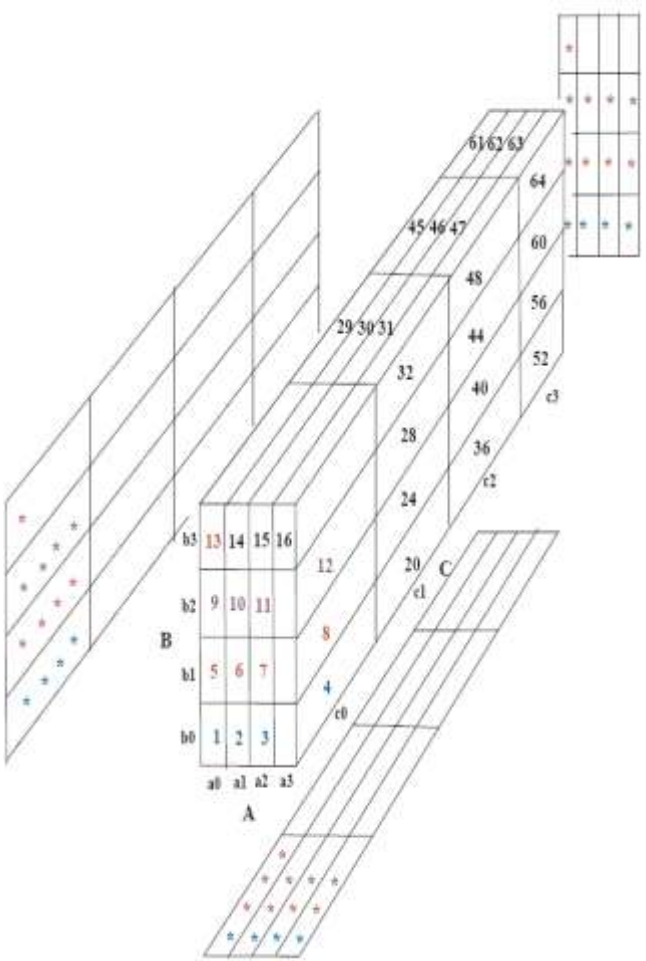
Операции с куб

- Roll up (drill-up): обобщаване на данни (сумиране)
  - чрез намаляване на броя на димeнсиите
- Drill down (roll down): детайлизиране
  - чрез увеличаване на броя на димeнсиите
  - чрез добавяне на нови димeнсии
- Slice: проекция
- Dice: селеция
- Pivot (rotate):
  - преориентиране на куба
  - визуализация
  - 3D представяне на 2D повърхнини
- Други операции
  - drill across: обобщаване на данни от повече от една таблици
  - drill through: включване на данни от релационни таблици

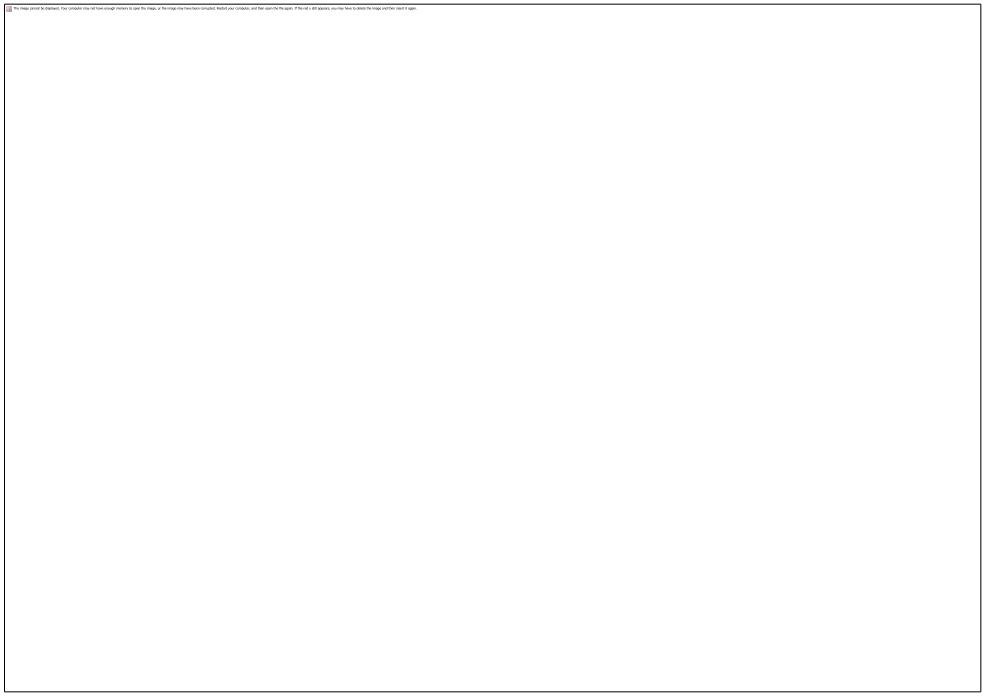

location (cities)
Toronto 395
Vancouver
time (quarters)
Q1 605
Q2
computer
home entertainment
item (types)
location (countries)
USA 2000
Canada
Q1 1000
Q2
Q3
Q4
computer security
home entertainment phone
item (types)
dice for
(location = ÒTorontoÓ or ÒVancouverÓ)
and (time = ÒQ1Ó or ÒQ2Ó) and
(item = Òhome entertainmentÓ or Òcomputer Ó)
roll-up
on location
(from cities
to countries)
location (cities)
Chicago 440
New York 1560
Toronto 395
Vancouver
time (quarters)
Q1 605 825 14 400
Q2
Q3
Q4
computer security
home entertainment phone
item (types)
slice
for time = ÒQ1Ó
drill-down
on time
(from quarters
to months)
location (cities)
Chicago
New York
Toronto
Vancouver 605 825 14 400
computer security
home entertainment phone
item (types)
pivot
item (types)
home entertainment 605
computer 825
phone 14
security 400
New York Vancouver
Chicago Toronto
location (cities)
location (cities)
Chicago
New York
Toronto
Vancouver
January 150
February 100
March 150
April
May
June
July
August
September
October
November
December
time (months)
computer security
home entertainment phone
item (types)

Проекции

3D-2D



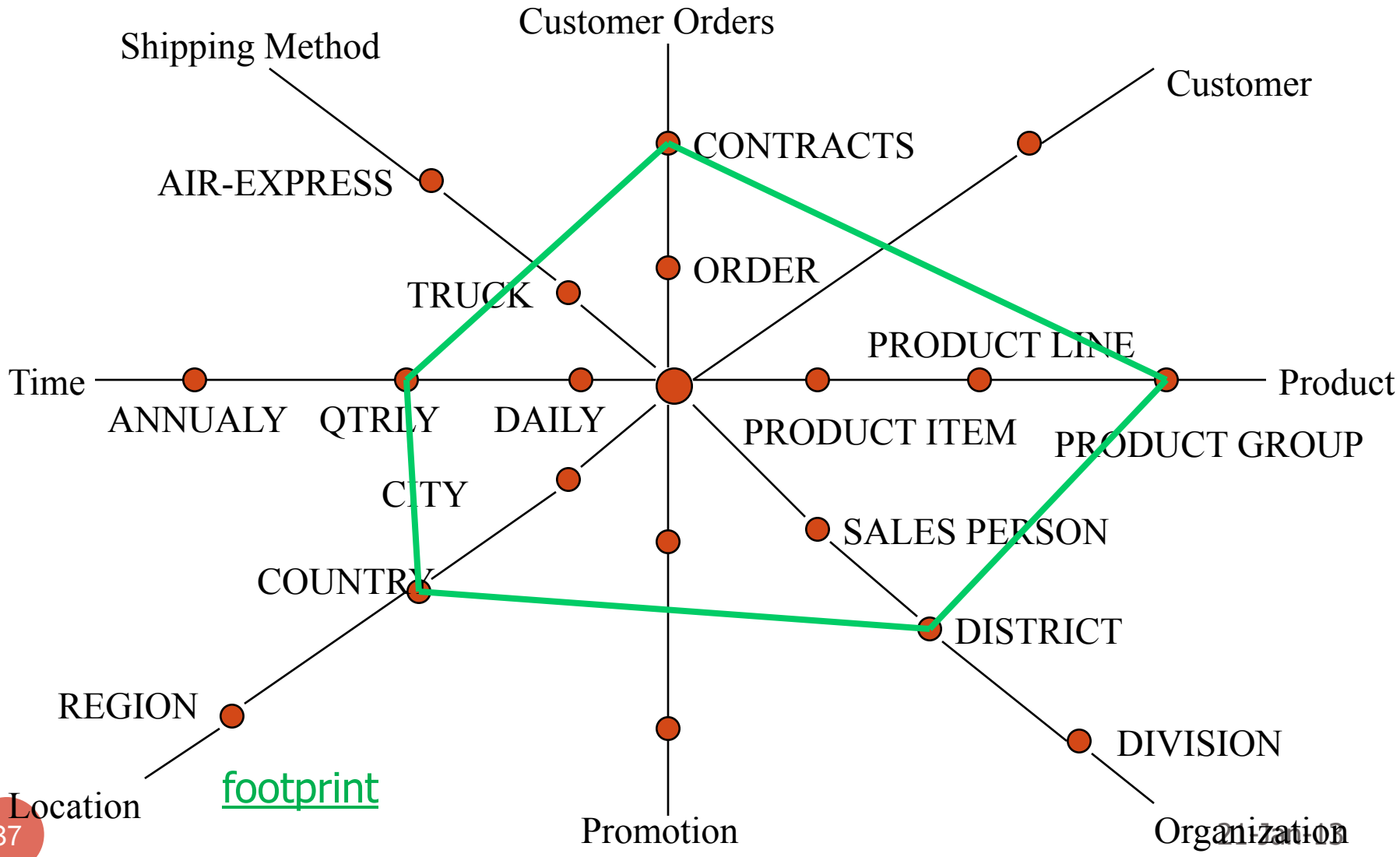
2D-1D



Модел Star-Net Query

- Радиални линии, пресичащи се в един център
- Всяка представя една димeнсия и стойностите от нейната йерархия
- Footprint – всяко абстрактно ниво на димeнсията
- Всички заедно представят множеството от нива на грануларност за изпълняваните операции върху куба

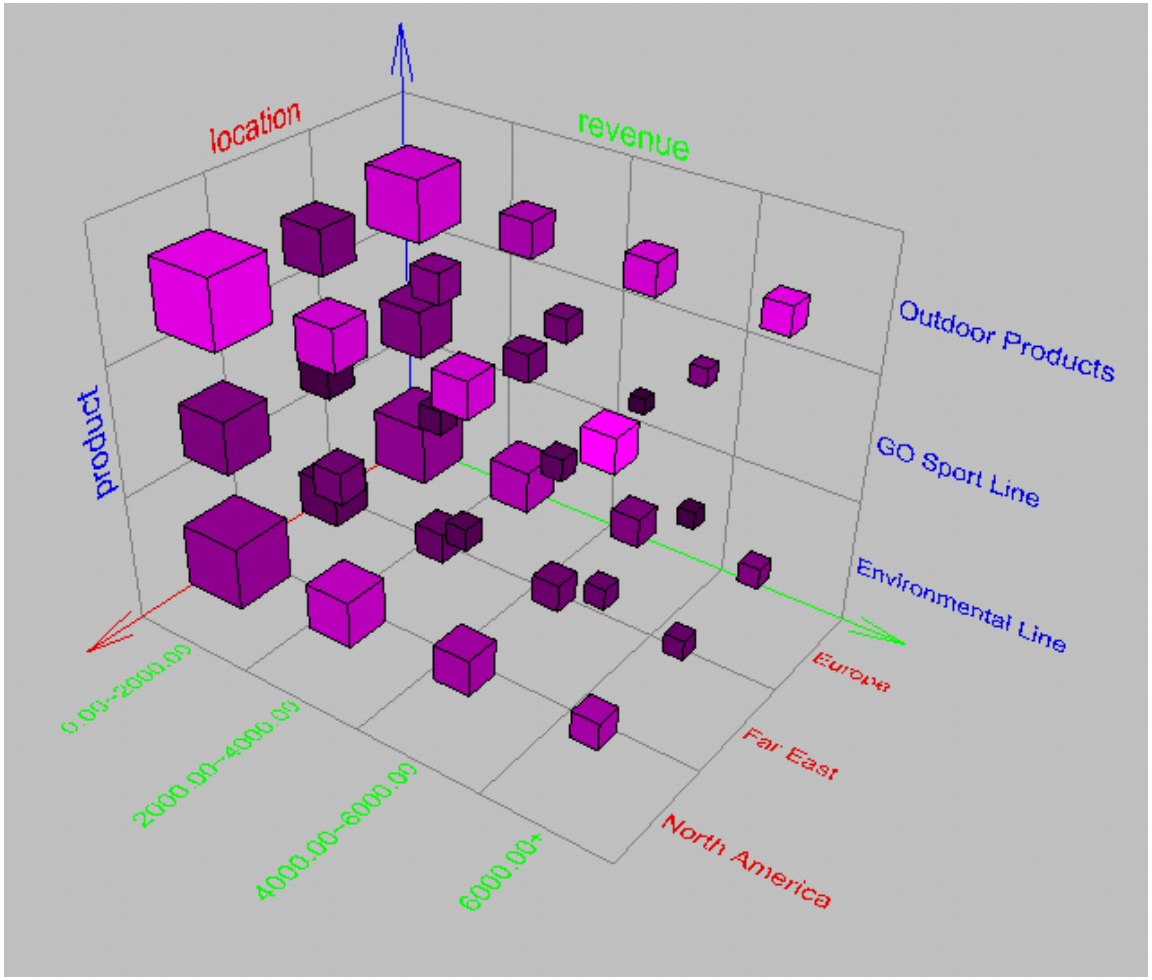
Модел Star-Net Query



footprint

Разглеждане и изследване на куб

- Визуализация
- OLAP
- Интерактивно изследване на куба



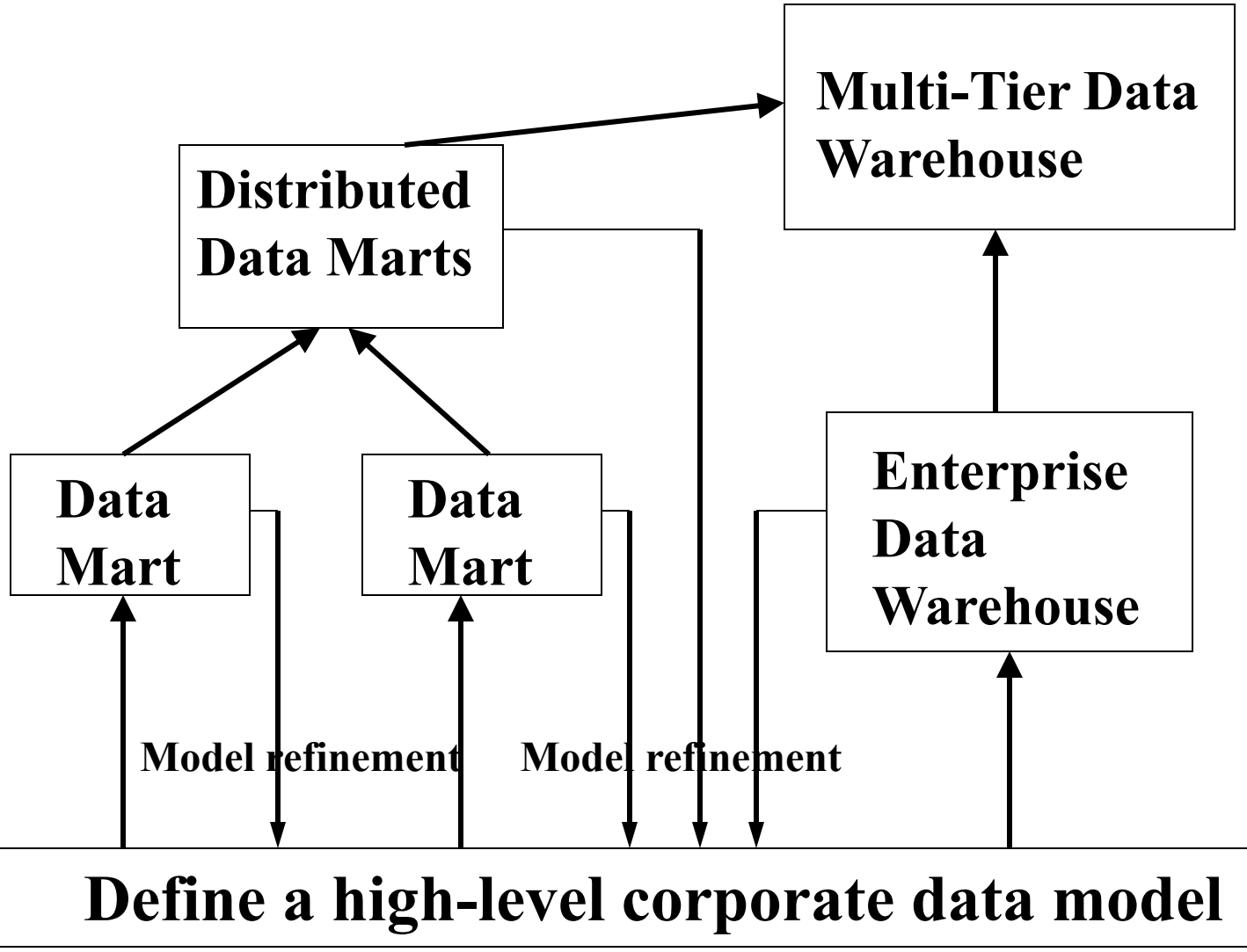
Проектиране на склад за данни

- Бизнес анализ на дейността
  - Top-down view
    - избор на подходяща информация от бизнеса за включване в склада
  - Data source view
    - анализ и включване на данни, събрани от операционните бази от данни
  - Data warehouse view
    - определяне на таблици със стойности и таблици с димeнсии
  - Business query view
    - разглеждане на склада за данни от гледна точка на крайния потребител

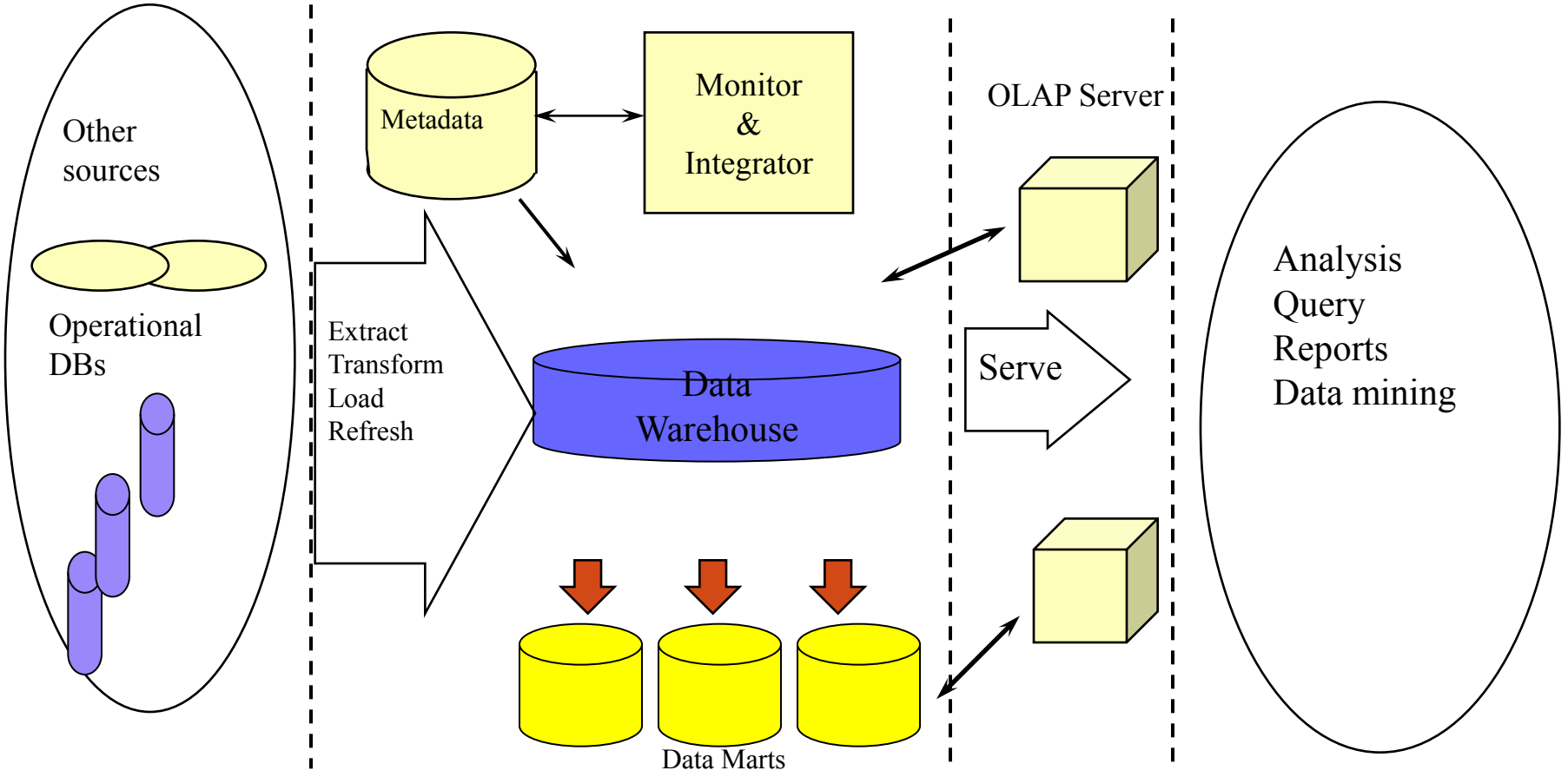
Проектиране на склад за данни

- **Процес на проектиране**
  - Избор на бизнес-процес за моделиране, напр. поръюки, продажби,...
  - Избор на атомарно ниво на регистриране на данни
  - Избор на димeнсии
  - Избор на метрики за факт-таблиците
- **Подходи**
  - Top-down: Цялостен анализ и проектиране
  - Bottom-up: Експериментиране и създаване на прототипи
- **Методи за проектиране**
  - Waterfall: структурен и систематичен анализ на всеки етап от проектирането
  - Spiral: бързо и циклично генериране на система с нарастваща сложност
  - други

Препоръчителен подход за създаване



Трислойна архитектура



Предварителна подготовка (ETL)

- **Data extraction**
  - събиране на данни от разнородни ресурси
- **Data cleaning**
  - контрол на пропуските и грешките
- **Data transformation**
  - преобразуване на данните в подходящ формат
- **Load**
  - проверка, обобщаване и зареждане на данните
- **Refresh**
  - обновяване на данните в съответствие с ресурсите

Метаданни

- Дефиниране на обектите на DW
  - описание на структурата на включените обети
  - операционни метаданни – статистики, грешки и др.
  - описание на обобщаващия алгоритъм
  - описание на връзката на средата с DW – ETL rules, gateways, etc
  - дефиниране на достъпа до данните в системата
  - бизнес данни

Използване на склад за данни

- Видове приложения
  - Information processing
    - заявки за извличане на информация
    - статистически анализ
    - генериране на отчети
    - визуализация на отчети
  - Analytical processing
    - анализ на многомерни данни
    - OLAP: slice-dice, drill, pivot
  - Data mining
    - откриване на знания за скрити отношения
    - асоциации, класификации, прогнози
    - визуализация на процеси

Примерни приложения

- Изследване на поведението на клиентите, чрез откриване на зависимости
- Анализ на реализацията на продуктите на бизнеса по отношение на време и пространство
- Анализ на средата и откриване на източници за печалба
- Анализ на управлението на бизнеса
- Оптимално използване на данни от различни източници
- …