# Data warehouse i OLAP tehnologije

# Glava 2. Sadržaj

- Šta je data warehouse?
- Više-dimenzioni model podataka
- Arhitektura data warehouse sistema
- Implementacija data warehouse sistema

# Šta je data warehouse?

- W. H. Inmon je prvi definisao data warehouse sisteme. Definicija glasi:
  - Data warehouse je subjektno-orijentisana, integrisana, vremenski zavisna i postojana kolekcija podataka za podršku u procesu odlučivanja
- Danas postoji veći broj ne strogih definicija
  - Baza podataka koja se koristi u procesu odlučivanja i koja je odvojena od operativnih podataka

# Data warehouse – subjektno-orijentisana

- Postoji centralni subjekat sistema (npr. klijent, proizvod, transakcija)
- Data warehouse sadrži sumarne podatke vezane za centralni subjekat, ne podatke vezane za dnevno poslovanje
- Osnovni cilj je podrška u procesu donošenja odluka (decision support), tj. modelovanje i analiza podataka

# Data warehouse - integrisana

- Data warehouse je kreirana integracijom podataka iz više heterogenih izvora
  - Relacione baze, obične datoteke, transakcione baze itd.
- Primjenjuju se tehnike za čišćenje i integraciju podataka
  - Obezbjeđuje konzistentnost u konvencijama imenovanja, načinima kodiranja i predstavljanja podataka, sistemima mjera itd.
  - Prilikom upisivanja u data warehouse podaci se konvertuju

# Data warehouse – vremenski zavisna

- U data warehouse sistemima je potrebno obezbijediti prikaz iz istorijske perspektive (5 do 10 godina unazad)
  - Operativne baze podataka sadrže samo trenutne vrijednosti
- Svaki ključ u data warehouse sistemima sadrži vremensku odrednicu (eksplicitno ili implicitno)

# Data warehouse - postojana

- Data warehouse sistemi su uvijek odvojeni od operativnih podataka
- Ne postoje transakcije, mijenjanje podataka (update), šeme oporavka, mehanizam kontrole konkurentnosti itd.
- U opštem slučaju su podržane operacije
  - Inicijalno učitavanje podataka
  - Pristup podacima

# Data warehouse vs. heterogeni SUBP

- Tradicionalni način integracije podataka je izgradnja tzv. medijatora ili wrapper-a
  - Query-driven approach: upit klijenta se transformiše u upite koji odgovaraju pojedninačnim bazama podataka, rezultati se spajaju i prevode i formu kakvu klijent očekuje
  - Neefikasni ako je broj upita veliki u jedinici vremena
- Data warehouse implementira update-driven approach
  - Podaci iz heterogenih izvora se prethodno integrišu i postaju dostupni za direktno izvršavanje upita i analizu

# Data warehouse vs. operativne BP

- OLTP (on-line transaction processing)
  - On-line izvršavanje upita i transakcija
  - Pokrivaju dnevne operacije
- OLAP (on-line analytical processing)
  - Analiza podatka i donošenje odluka
  - Podrška data warehouse sistemima
- Poređenje: OLTP vs. OLAP
  - Korisnik: obični vs. napredni korisnici
  - Podaci: tekući i detaljni vs. istorijski i sumarni prikaz
  - Dizajn BP: ER model vs. višedimenzionalni model
  - Operacije: update vs. read-only ali sa složenim upitima

# OLTP vs. OLAP

| | **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<br>detailed, flat relational<br>isolated | historical,<br>summarized, multidimensional<br>integrated, consolidated |
| **usage** | repetitive | ad-hoc |
| **access** | read/write<br>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 |

# Zašto data warehouse?

- Visoke performanse za oba sistema
  - SUBP za OLTP: metode pristupa, indeksiranje, oporavak, konkurentnost
  - Data warehouse za OLAP: složeni upiti, generalizacija podataka
- Različite funkcionalnosti i različiti podaci
  - Istorijski podaci
  - Generalizacija podataka
  - Kvalitet podataka

# Glava 2. Sadržaj

- Šta je data warehouse?
- Više-dimenzioni model podataka
- Arhitektura data warehouse sistema
- Implementacija data warehouse sistema

# Više-dimenzioni model podataka

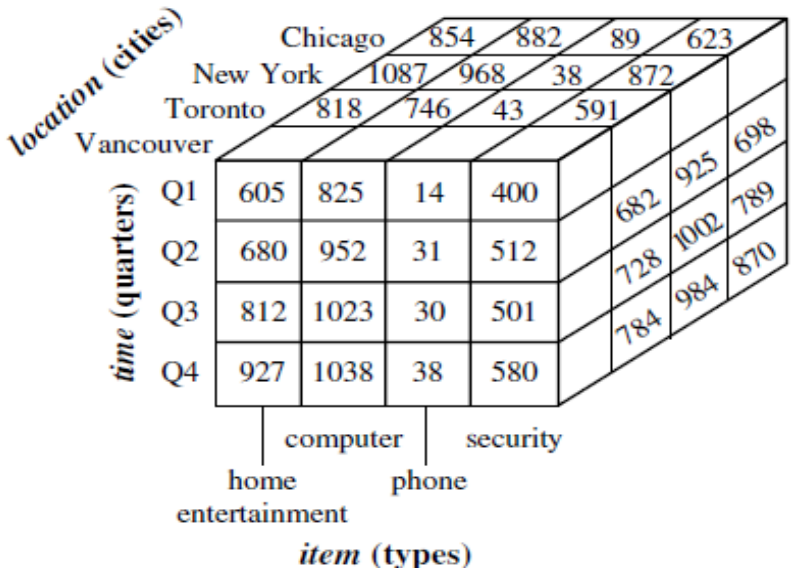
- Data warehouse je zasnovan na više-dimenzionom modelu koji podatke predstavlja u formi tzv. n-dimenzionih kocaka podataka
- Kocke podataka definišu se sa
  - Tabela činjenica: sadrži uglavnom numeričke podatke i spoljašnje ključeve na tabele dimenzije
  - Tabele dimenzije: omogućavaju prikaz podataka na više načina

# 2-D kocka podataka

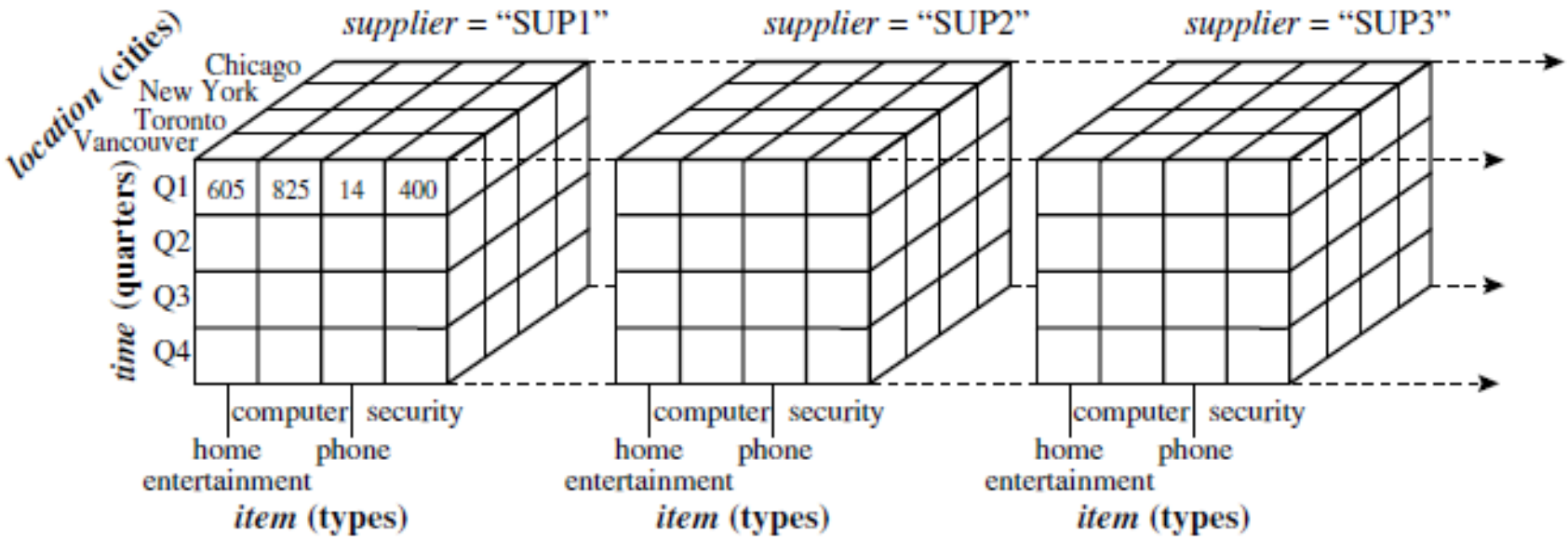
| | ***location*** = *"Vancouver"* | | | |
|---|---|---|---|---|
| | ***item*** *(type)* | | | |
| ***time*** *(quarter)* | *home entertainment* | *computer* | *phone* | *security* |
| Q1 | 605 | 825 | 14 | 400 |
| Q2 | 680 | 952 | 31 | 512 |
| Q3 | 812 | 1023 | 30 | 501 |
| Q4 | 927 | 1038 | 38 | 580 |

# 3-D kocka podataka

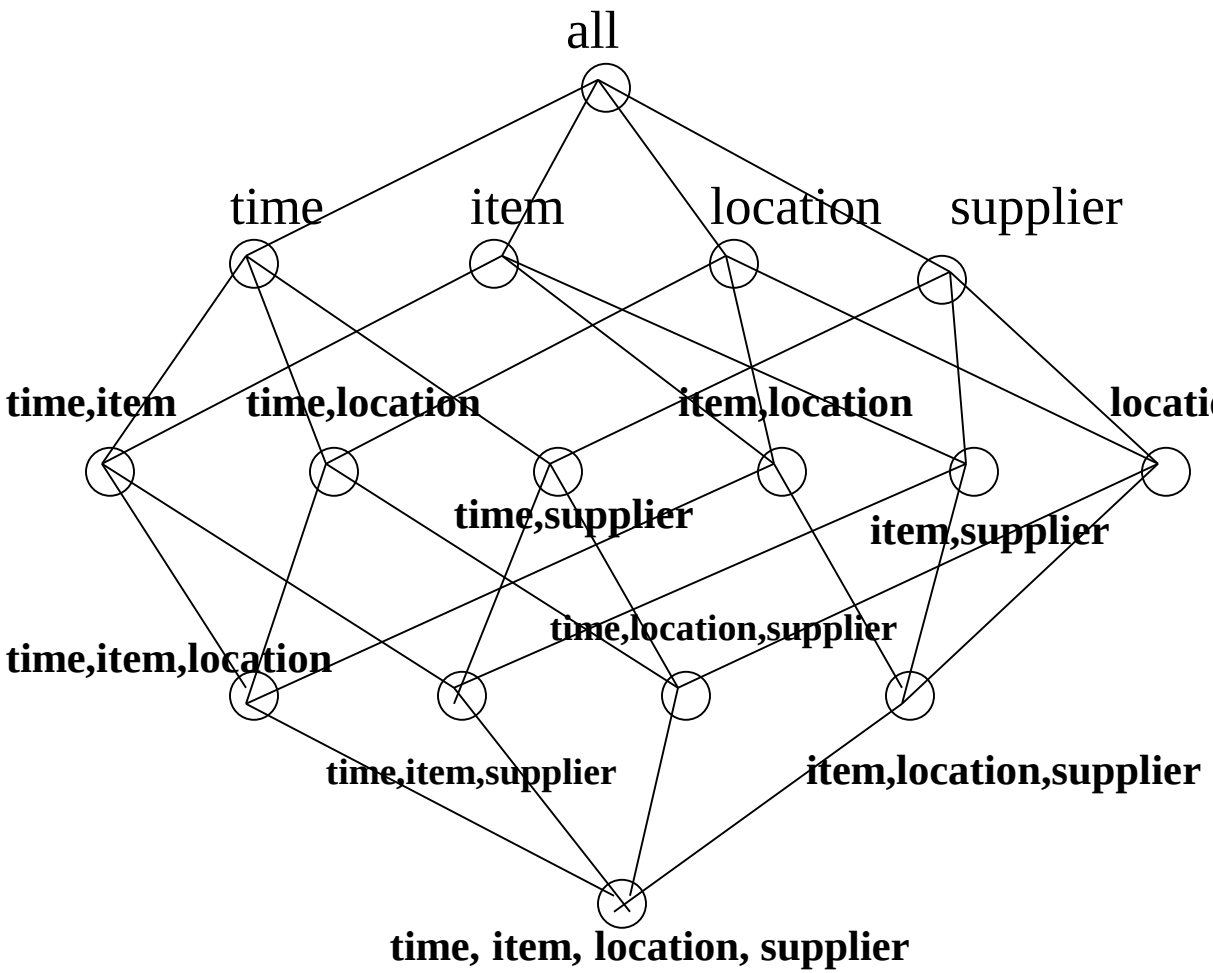
| | location = "Chicago" | | | | location = "New York" | | | | location = "Toronto" | | | | location = "Vancouver" | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | item | | | | item | | | | item | | | | item | | | |
| | home | | | | home | | | | home | | | | home | | | |
| time | ent. | comp. | phone | sec. | ent. | comp. | phone | sec. | ent. | comp. | phone | sec. | ent. | comp. | phone | sec. |
| Q1 | 854 | 882 | 89 | 623 | 1087 | 968 | 38 | 872 | 818 | 746 | 43 | 591 | 605 | 825 | 14 | 400 |
| Q2 | 943 | 890 | 64 | 698 | 1130 | 1024 | 41 | 925 | 894 | 769 | 52 | 682 | 680 | 952 | 31 | 512 |
| Q3 | 1032 | 924 | 59 | 789 | 1034 | 1048 | 45 | 1002 | 940 | 795 | 58 | 728 | 812 | 1023 | 30 | 501 |
| Q4 | 1129 | 992 | 63 | 870 | 1142 | 1091 | 54 | 984 | 978 | 864 | 59 | 784 | 927 | 1038 | 38 | 580 |



# 4-D kocka podataka



# Kocke podataka, mreža kuboida




0-D(apex) cuboid

1-D cuboids

2-D cuboids

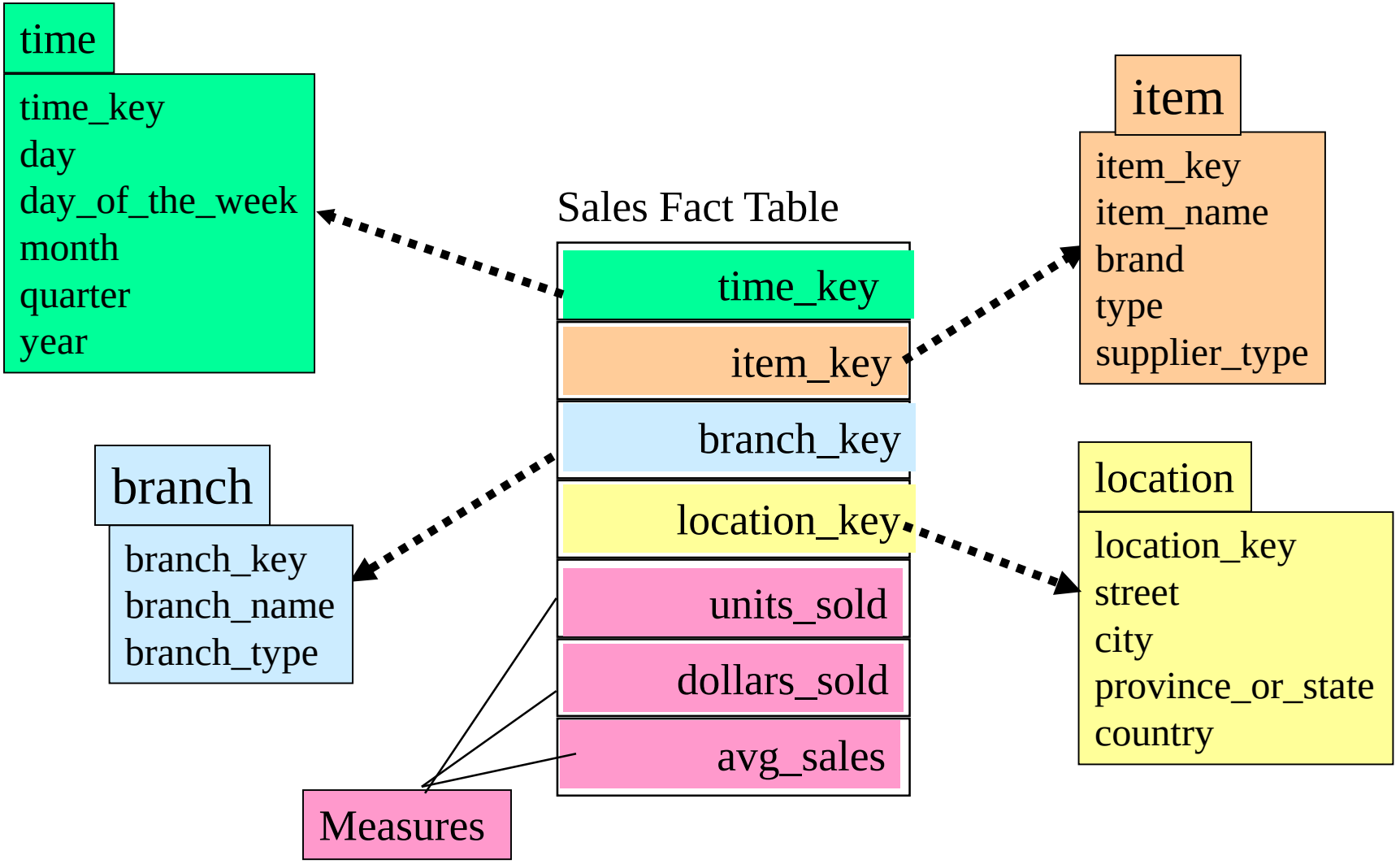
3-D cuboids

4-D(base) cuboid

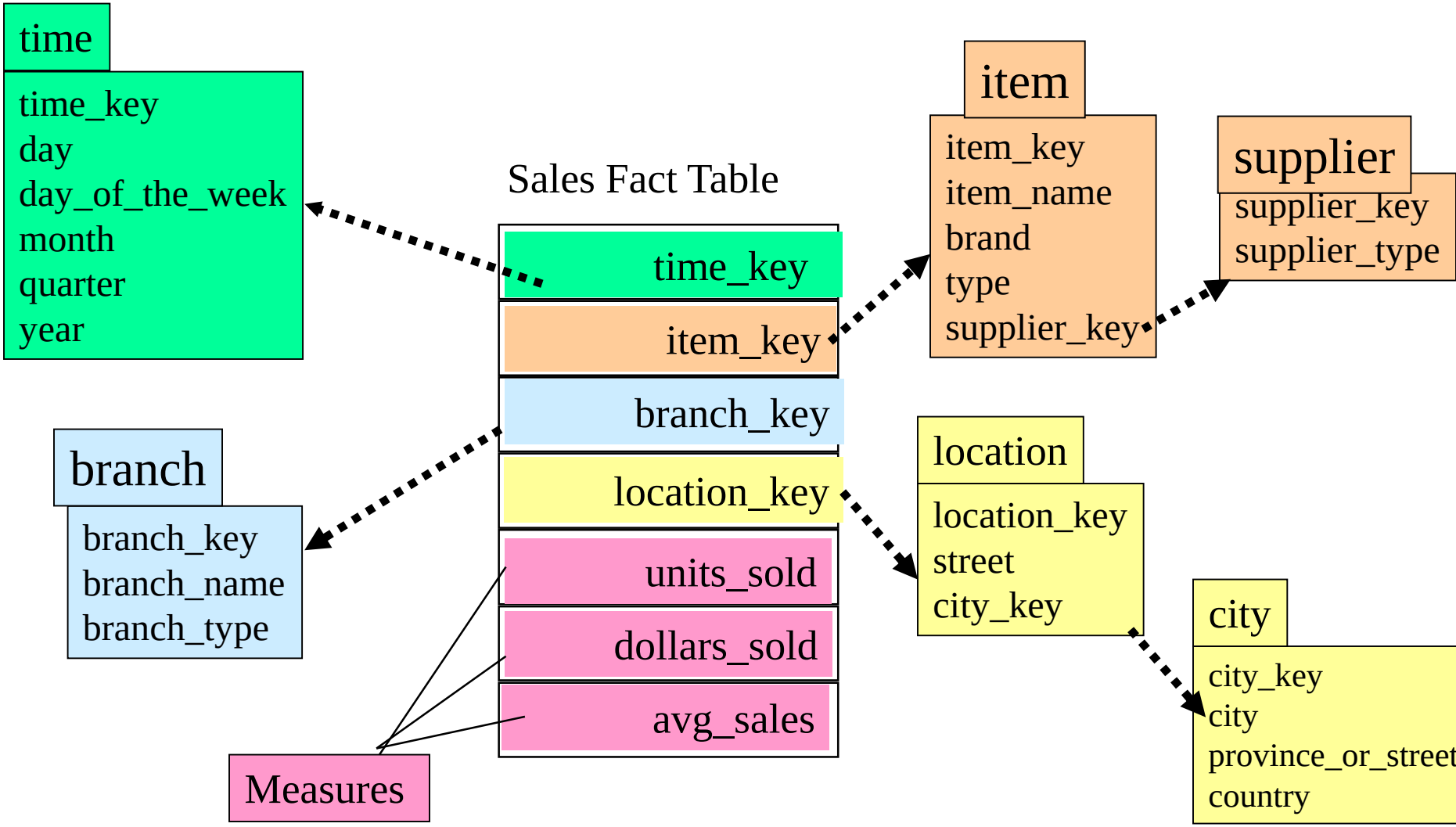
# Konceptualni model data warehouse sistema

- Šema zvijezda
  - Tabela činjenica je u “sredini” i povezana je sa tabelama dimenzijama
- Šema pahulja
  - Modifikacija šeme zvijezda na način da su neke tabele dimenzije normalizovane
- Šema galaksija
  - Tabele činjenica dijele tabele dimenzija, kolekcija zvjezdastih šema

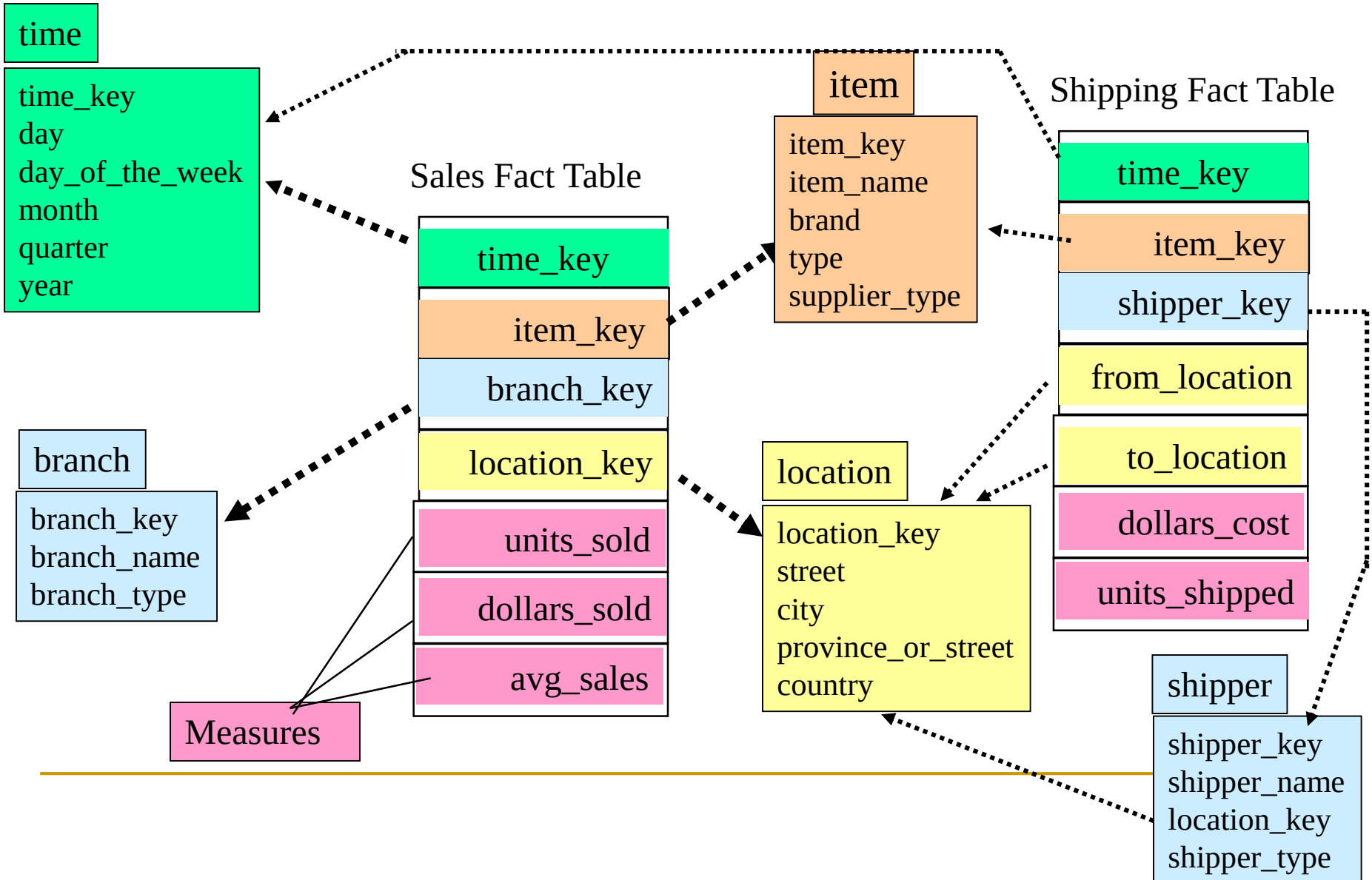
# Primjer šeme zvijezda



# Primjer šeme pahulja



# Primjer šema galaksija



# DMQL – Data Mining Query Language

- Tabela činjenica

  define cube <cube_name>[<dimension_list>]: <measure_list>

- Tabela dimenzija

  define dimension <dimension_name> as (<attribute_or_subdimension_list>)

- Dijeljene tabele dimenzije

  define dimension <dimension_name> as <dimension_name_first_time> in cube <cube_name_first_time>

# Definicija šeme zvijezda u DMQL

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)

# Definicija šeme pahulja u DMQL

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))

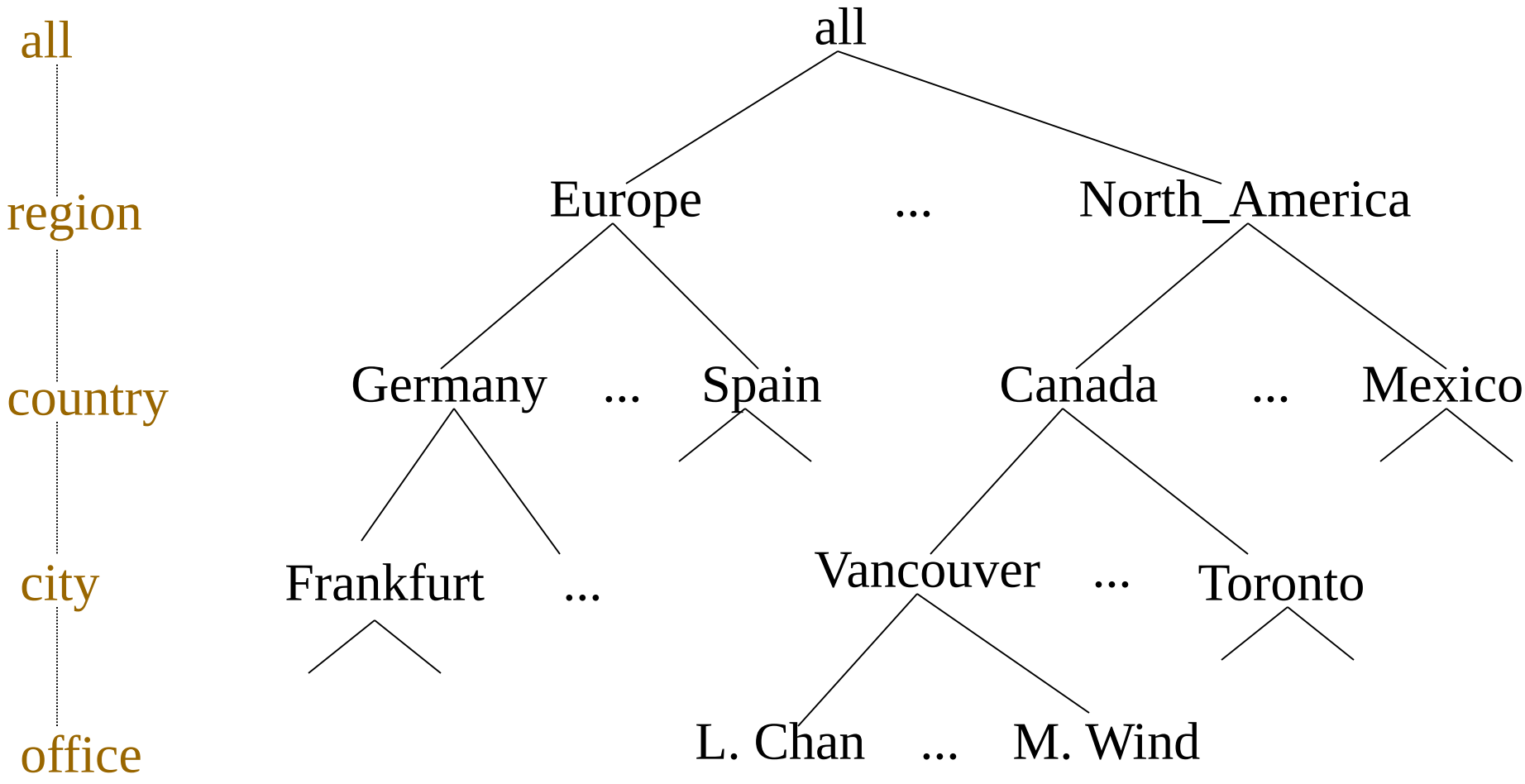
define dimension branch as (branch_key, branch_name, branch_type)

define dimension location as (location_key, street, city(city_key, province_or_state, country))

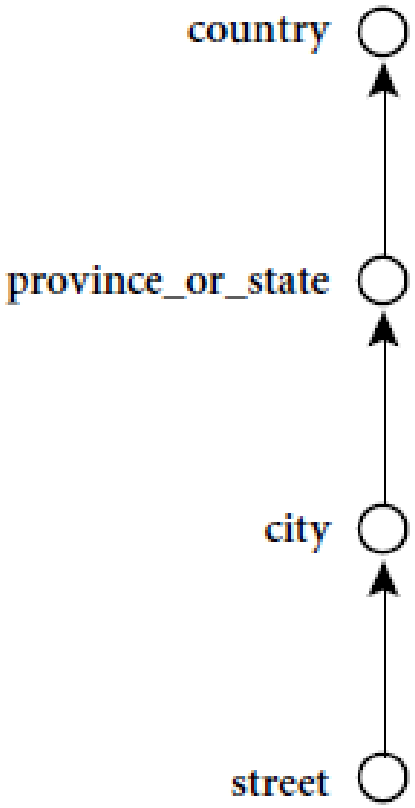
# Definicija šeme galaksija u DMQL

```
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
```

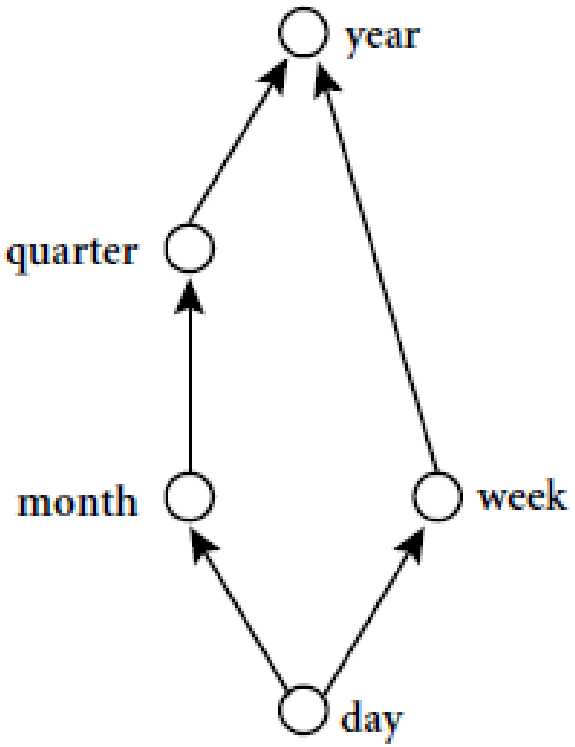
# Hijerarhije koncepata



# Hijerarhije koncepata (2)

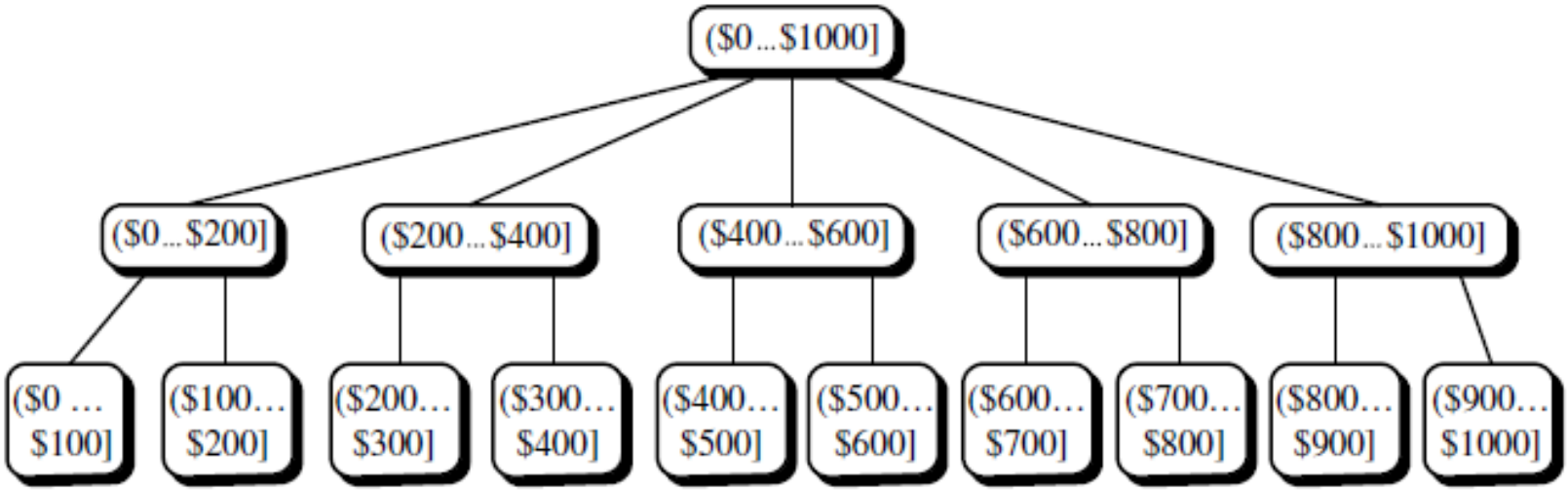



Potpuno uređenje




Mreža

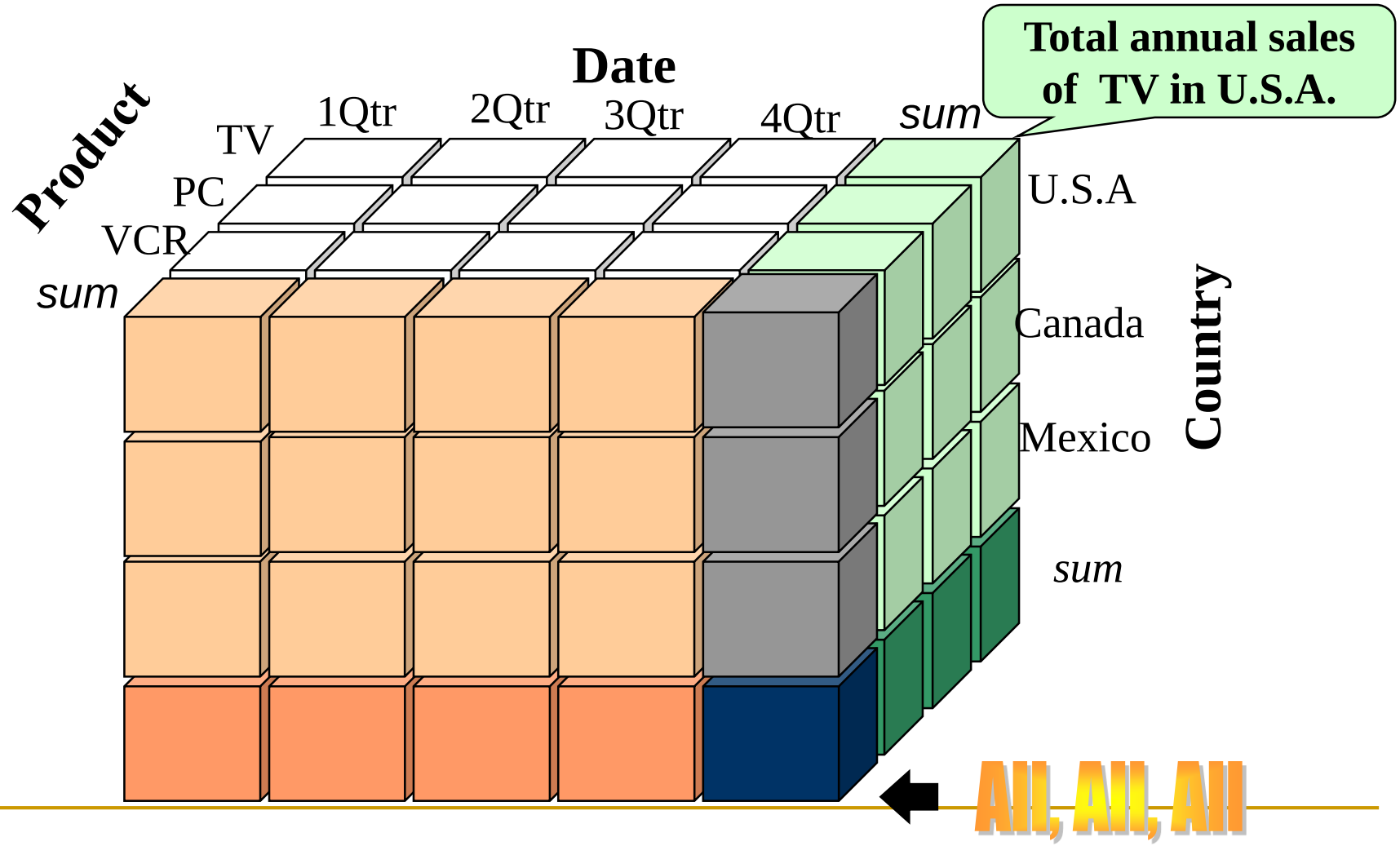
# Hijerarhije koncepata (3)



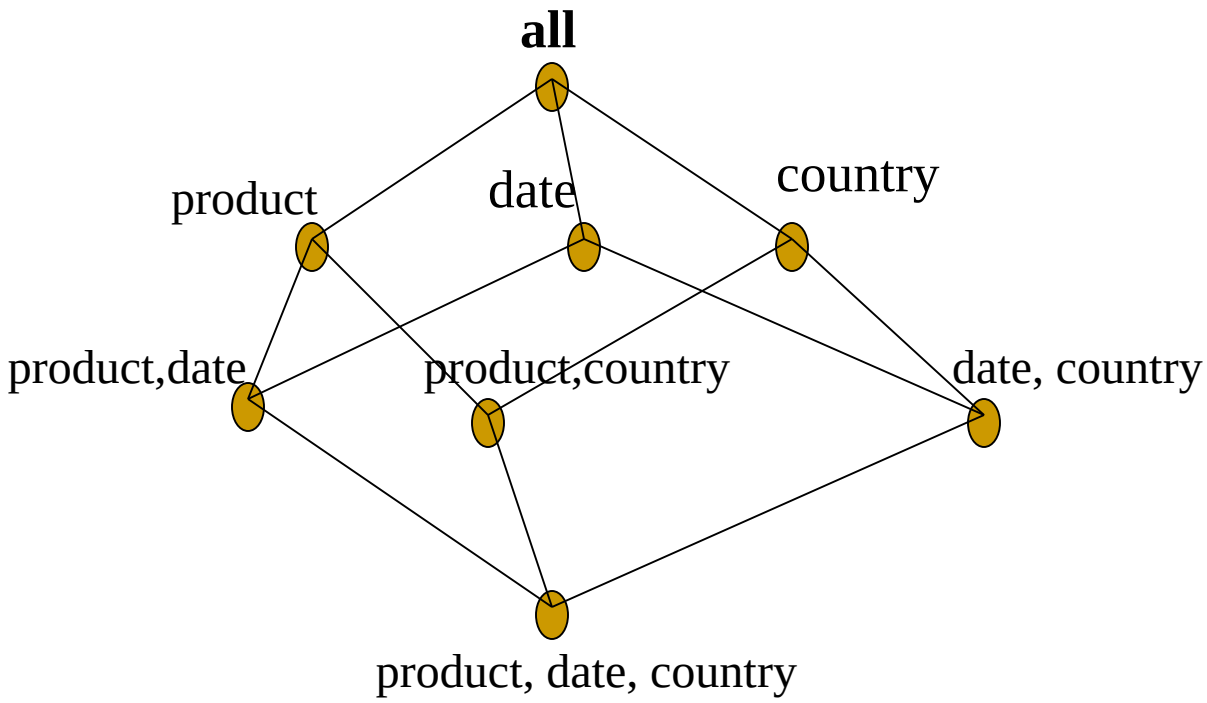
# Višedimenzioni podaci, primjer

- Dimenzije: vrijeme, lokacija i proizvod
- Činjenica: vrijednost prodate robe u hiljadama eura

# Više-dimenzioni podaci, primjer (2)



# Mreža kuboida za prethodni primjer




0-D(apex) cuboid

1-D cuboids

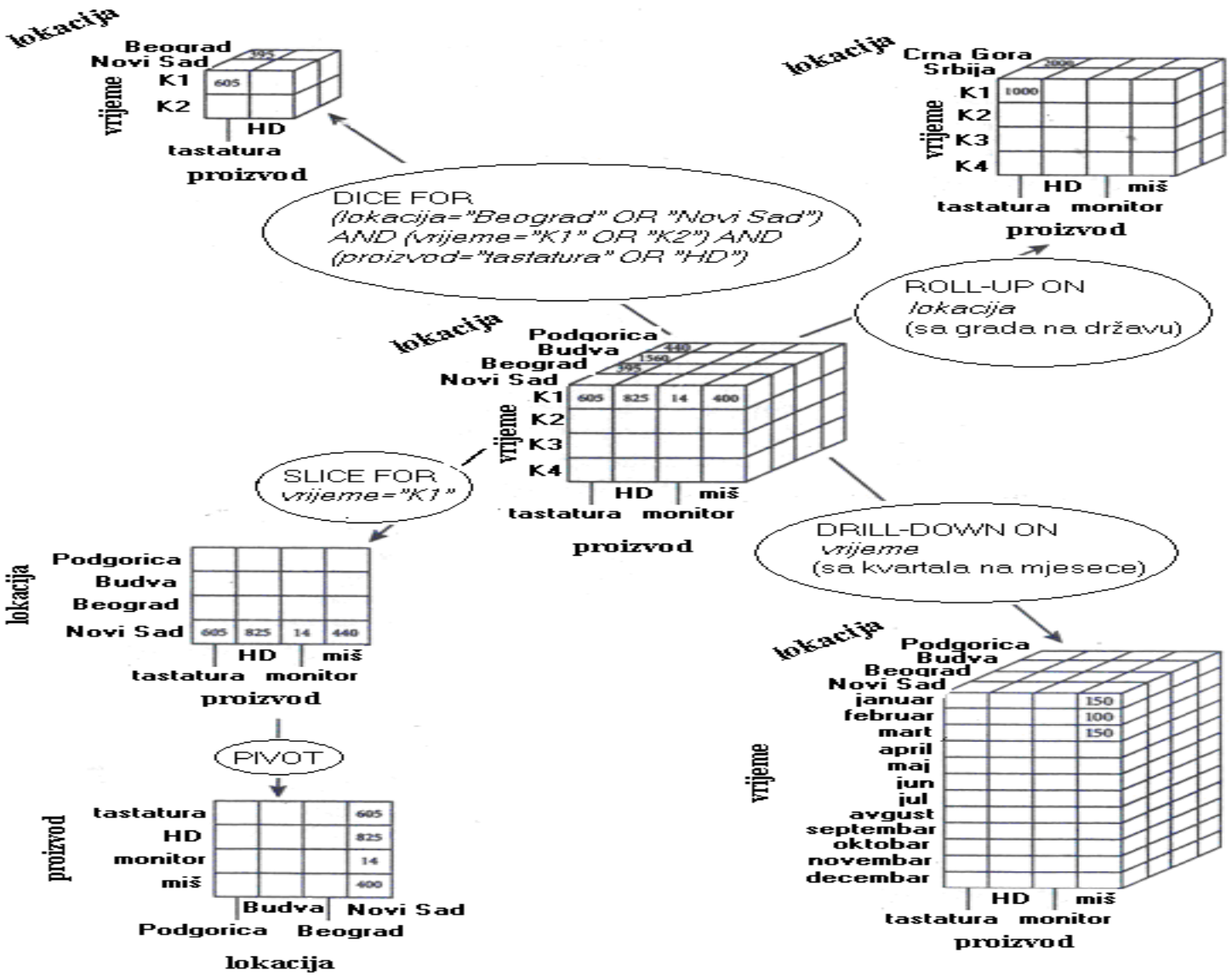
2-D cuboids

3-D(base) cuboid

# OLAP operacije

- Roll up (drill up): agregacija ukidanjem dimenzija ili prelaskom sa nižeg na viši nivo hijerarhije
- Drill down (roll down): suprotno od roll up
- Slice: selekcija po jednoj dimenziji
- Dice: selekcija po dvije ili više dimenzije
- Pivot: rotacije kubova podataka

# OLAP operacije, primjer



# Glava 2. Sadržaj

- Šta je data warehouse?
- Više-dimenzioni model podataka
- Arhitektura data warehouse sistema
- Implementacija data warehouse sistema

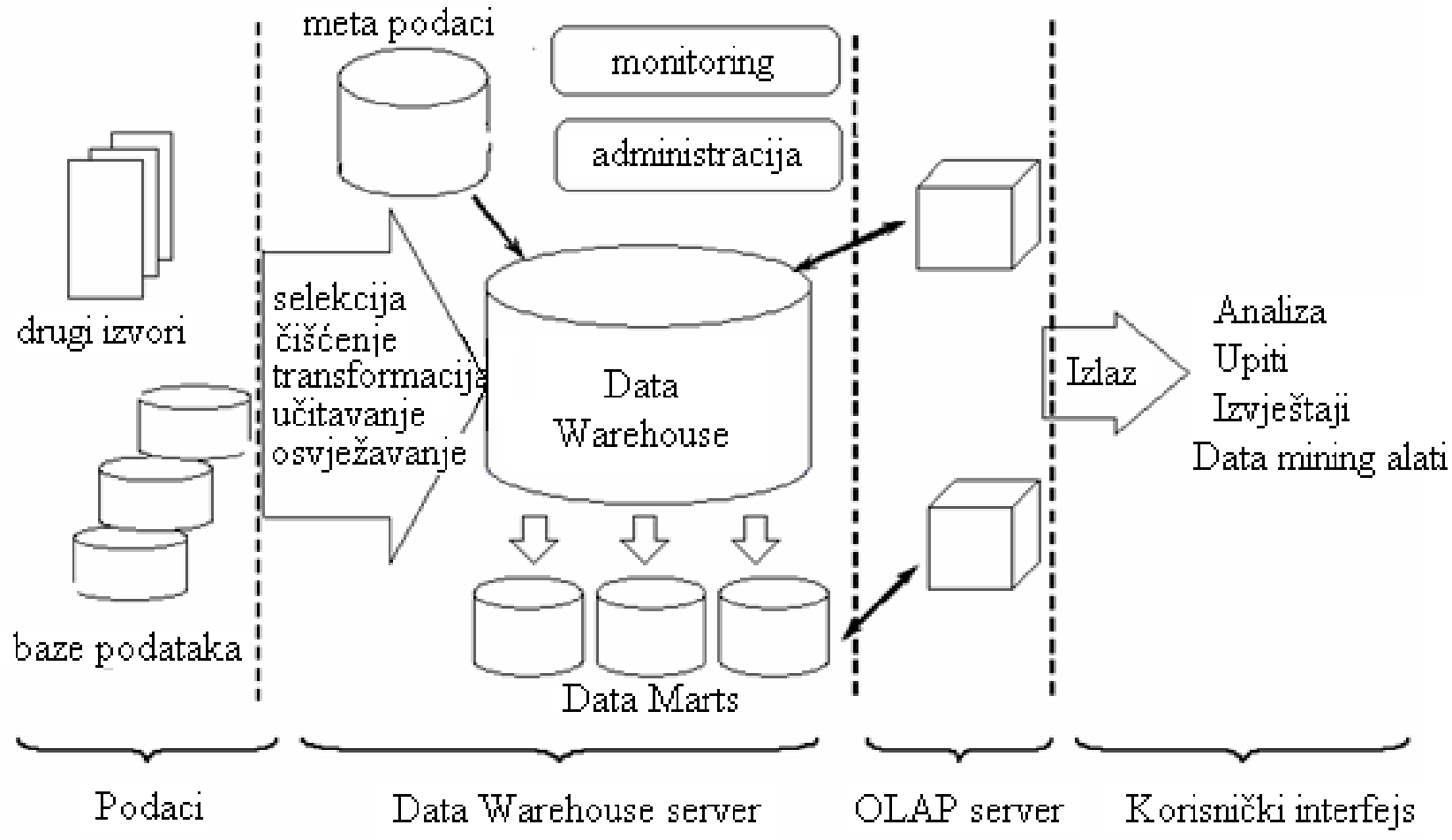
# Dizajn data warehouse sistema

- Četiri pogleda na dizajn data warehouse sistema:
  - Top-down view: selekcija neophodnih podataka
  - Data source view: izvori podataka (tabele, relacione baze podataka, itd.)
  - Data warehouse view: tabele činjenica i dimenzija
  - Business query view: način na koji krajnji korisnici vide podatke u data warehouse sistemu

# Izgradnja data warehouse sistema

- Top-down, bottom-up ili kombinacija
- Iz ugla SE
  - Metoda vodopada
  - Inkrementalna metoda
- Izgradnja data warehouse sistema se u opštem slučaju sastoji od:
  - Definisanje poslovnog procesa koji se modeluje
  - Definisanje atomičnog nivoa podataka
  - Definisanje dimenzija za tabele činjenice
  - Definisanje numeričkih podataka za tabele činjenice

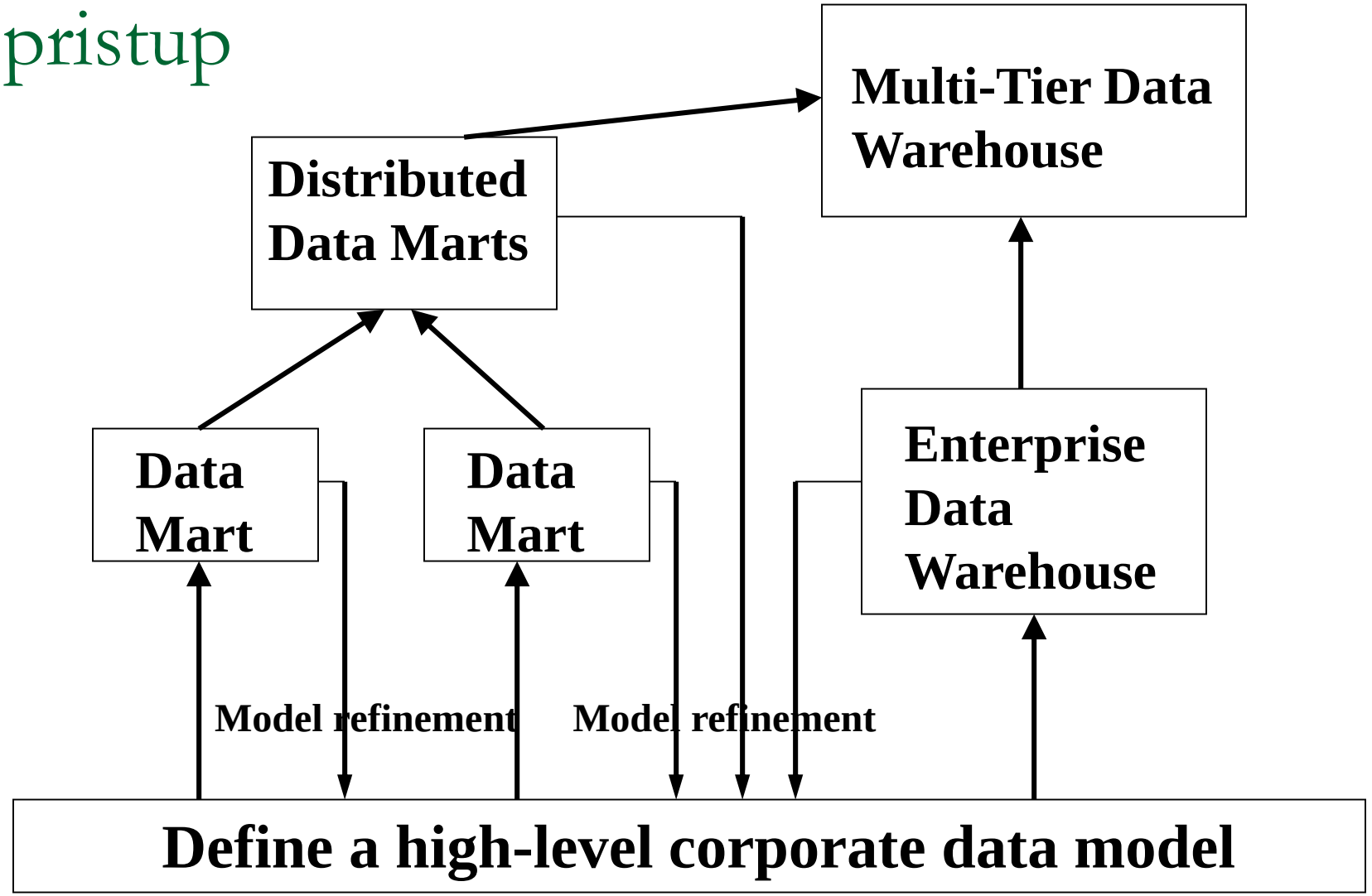
# Arhitektura data warehouse sistema



# Modeli data warehouse sistema

- Iz ugla arhitekture, postoje tri modela:
  - Enterprise warehouse: obuhvata podatke cijele organizacije
  - Data mart: obuhvata podatke koji se odnose samo na dio organizacije, namijenjena specifičnoj grupi korisnika
    - Zavisni vs. Nezavisni
  - Virtualni warehouse: sastoji se od pogleda (eventualno materijalizovanih zbog boljih performansi)

# Izgradnja data warehouse, preporučeni pristup



# Arhitektura OLAP servera

- Relacioni OLAP (ROLAP)
  - Koriste relacione SUBP za smještanje i upravljanje podacima
- Multidimensional OLAP (MOLAP)
  - Direktno podržava višedimenzione podatke i smješta ih u višedimenzione nizove; određene upite unaprijed izračunava
- Hibridni OLAP (HOLAP)
  - Niži nivo: relacije, viši nivo: nizovi

# Glava 2. Sadržaj

- Šta je data warehouse?
- Više-dimenzioni model podataka
- Arhitektura data warehouse sistema
- Implementacija data warehouse sistema

# Efikasno izračunanje kocaka podataka

- Kocka podataka može da se predstavi kuboidima
  - Kuboid na najnižem nivou je osnovni
  - Kuboid na najvišem nivou je apex kuboid
  - Koliko ima kuboida u kubu sa n dimenzija?
- Materijalizacija kocaka podataka
  - Kompletna, djelimična ili bez materijalizacije
  - Koji kuboidi se materijalizuju zavisi od veličine, broja pristupa u jedinici vremena itd.

# Operacije sa kubovima podataka

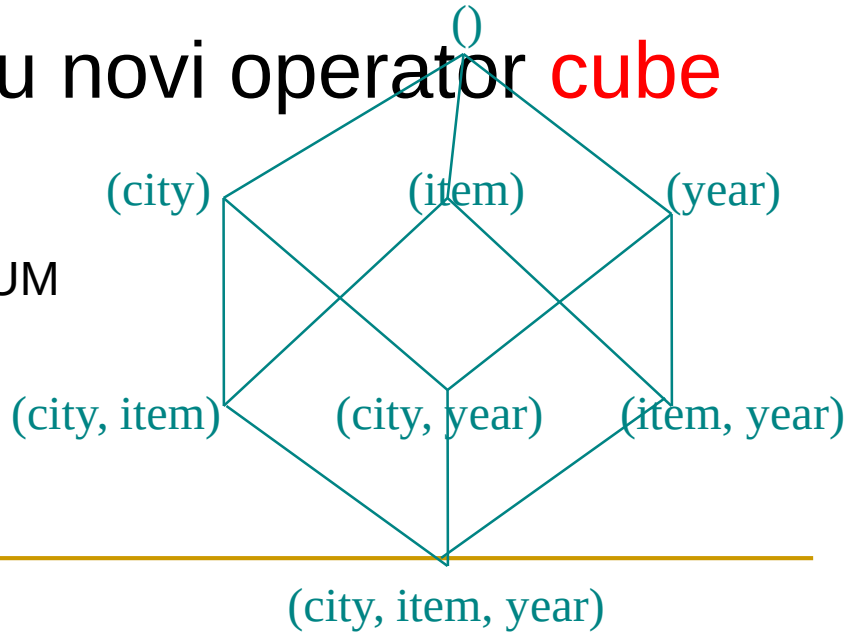
- Definisanje i izračunavanje kuba podataka u DMQL-u

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

  compute cube sales

- Gray et al. 1996 uveli su novi operator cube by

```
SELECT item, city, year, SUM
(amount)
FROM SALES
CUBE BY item, city, year
```



# Izračunavanje kubova - algoritmi

- Algoritmi
  - ROLAP-based cubing algorithms (Agarwal et al'96)
  - Array-based cubing algorithm (Zhao et al'97)
  - Bottom-up computation method (Bayer & Ramarkrishnan'99)
  - Multi-way Array Aggregation

# Indeksiranje podataka u OLAP sistemima, bitmap indeksi

- Za svaki atribut kreira se indeks
- U okviru ovoga indeksa, postoje posebni bit vektori Bv, za svaku vrijednost v iz domena atributa

**Base table**

| Cust | Region | Type |
|---|---|---|
| C1 | Asia | Retail |
| C2 | Europe | Dealer |
| C3 | Asia | Dealer |
| C4 | America | Retail |
| C5 | Europe | Dealer |

**Index on Region**

| RecID | Asia | Europe | America |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 |

**Index on Type**

| RecID | Retail | Dealer |
|---|---|---|
| 1 | 1 | 0 |
| 2 | 0 | 1 |
| 3 | 0 | 1 |
| 4 | 1 | 0 |
| 5 | 0 | 1 |

# Indeksiranje podataka u OLAP sistemima, join indeksi

- Join indeks povezuje torke koje se spajaju: JI(RID, SID)
- U DW sistemima spajaju torke iz tabele činjenica sa odgovarajućim iz tabele dimenzija