Data Warehouses
Data Mining

František Ščuglík

Datové sklady a Technologie OLAP pro dolování dat

Obsah

- Co to je Datový sklad?
- Multi-dimensionální datový model
- Architektura Datového skladu
- Implementace Datového skladu
- Dolování dat z Datového skladu

Co je to Datový sklad?

- Je popsán několika různými definicemi, ale žádná není přesná:
  - Databáze zpracovávaná separátně od podnikové relační databáze, určená pro podporu v rozhodování
  - Umožňuje zpracovávání informací díky velké platformě pevných historických dat pro analýzu
- „Datový sklad je subjektově orientovaná, integrovaná, časově proměnná a stálá kolekce dat pro podporu rozhodování managementu“ – W. H. Inmon

Subjektově orientovaná

- Organizovaná kolem majoritních subjektů, jako např. zákazník, produkt, prodej…
- Zaměřuje se na modelování a analýzu dat, na na denní zpracování operací a transakcí
- Zajišťuje jednoduchý a stručný pohled na různé části subjektů přičemž vynechává data která nejsou užitečná v rozhodovacím procesu.

Integrovaná

- Sestavená integrací několika heterogenních zdrojů dat
  - Relační databáze, soubory, transakce
- Používají se techniky na čištění a integraci dat
  - Zaručuje konzistenci v pojmenovávání, atributech apod. při zpracování různých zdrojů dat
    - Např., Cena Hotelu: měna, taxa, atd.
  - Když se data přesouvají do datového skladu, jsou konvertována.

Časově proměnná

- Časový horizont Datového skladu je výrazně delší než u operační databáze
  - Operační databáze: aktuální data.
  - Datový sklad: informace z historické perspektivy (např. posledních 5 až 10 let)
- Každá klíčová struktura v Datovém skladu obsahuje časový element (expicitně nebo implicitně)
  - Ale klíč operační databáze může a nemusí obsahovat časový element

Stálá

- Fyzicky oddělený sklad dat transformovaný z operačního prostředí
- Operační update dat se v prostředí datových skladů nevyskytuje
  - Nevyžaduje zpracování transakcí, zotavení a mechanismus řízení konkurentního přístupu
  - Vyžaduje pouze dvě operace v přístupu k datům:
    - *Prvotní nahrání dat* a *přístup k datům*

Datové sklady vs. Heterogenní DB

- Tradiční integrace heterogenních DB:
  - vytvoření wrapperů/mediatorů nad heterogenními DB
  - Query driven přístup
    - **Pokud je na klientské straně vložen dotaz, použije se slovník metadat metadat který transformuje daný dotaz do podoby přijatelné pro danou heterogenní DB a výsledek je pak integrován do globální množiny odpovědí**
    - **Neefektivní a potenciálně nákladné pro časté dotazy**
- Datový sklad: update-driven, vysoký výkon
  - Informace z heterogenních zdrojů jsou integrovány předem a uloženy v datovém skladu pro přímé

Datový sklad vs. Operační DB

- OLTP (on-line transaction processing)
  - Hlavní úkol relačních DB je zpracovávat on-line
  - Day-to-day operace: nákupy, zásoby, bankovnictví, výroba, výplaty, registrace, účtování, atd.
- OLAP (on-line analytical processing)
  - Hlavní úkol Datového skladu je analyzovat uložená data
- Rozdílné vlastnosti (OLTP vs. OLAP):
  - Orientace systému: uživatel vs. marketing
  - Obsah dat: aktuální, detailní vs. Historické, sloučené
  - Design databáze: ER vs. star, snowflake
  - Pohled: aktuální, lokální vs. Vývoj firmy, integrovaný
  - Modely přístupu: update vs. read-only, ale complexní dotazy

OLTP vs. OLAP

	OLTP	**OLAP**
uživatelé	Uředník, IT professional	Knowledge pracovník
funkce	Operarace day-to-day	Podpora v rozhodování
DB design	application-oriented	subject-oriented
data	aktuální, up-to-date detailní	historické, sumarizované, multidimensionálně integrované
přístup	read/write index/hash na prim. key	Množství scanů
jednotka práce	krátké, jednoduché transakce	complexní dotazy
počet záznamů	Desítky	Miliony
počet uživatelů	Tisíce	Stovky
velikost DB	100MB-GB	100GB-TB
metrika	Propustnost transakcí	Propustnost dotazů, doba odezvy

Proč separovaný Datový sklad?

- Vysoký výkon pro oba systémy
  - DBMS— laděný pro OLTP: přístupové metody, indexování, konkurentní přístup, zotavení
  - Dat. sklad—laděný pro OLAP: complexní OLAP dotazy, multidimensionální pohledy, slučování.
- Rozdílné funkce a rozdílná data:
  - Chybějící data: proces rozhodování potřebuje historická data, která operační DB obvykle neudržují
  - Slučování dat: proces rozhodování potřebuje slučování (agregaci, sumaci) dat z heterogenních zdrojů
  - Kvalita dat: rozdílné zdroje dat typicky používají nekonzistentní reprezentaci dat

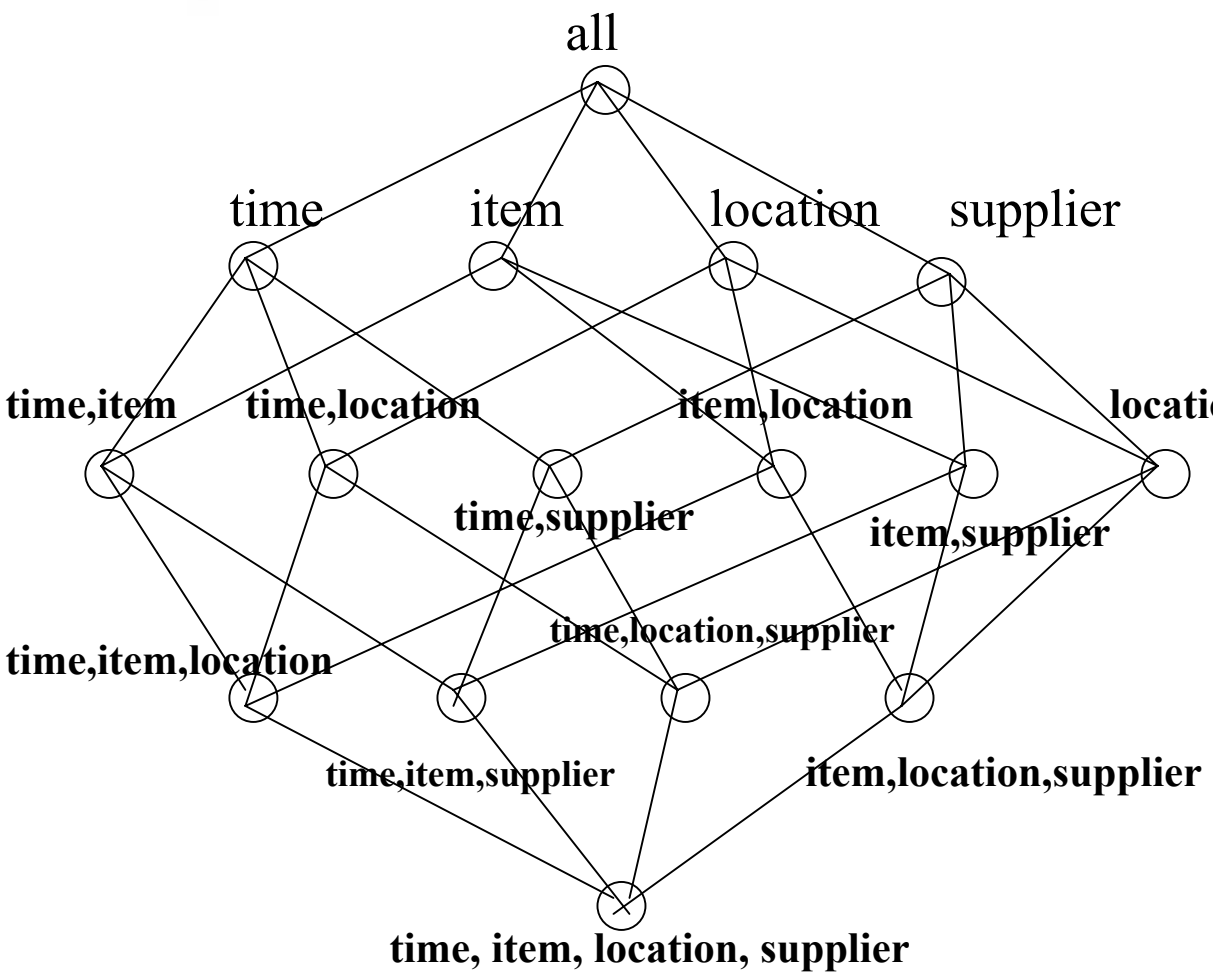
Obsah

- Co to je Datový sklad?
- Multi-dimensionální datový model
- Architektura Datového skladu
- Implementace Datového skladu
- Dolování dat z Datového skladu

Datové kostky

- Datový sklad je založen na multidimensionálním datovém modelu, který ukazuje data ve formě datové kostky
- Datová kostka, jako např. sales, umožňuje datům, aby byla modelována a prohlížena v několika dimenzích
  - **Tabulky dimenzí, jako item (item_name, brand, type), nebo time(day, week, month, quarter, year)**
  - **Tabulka faktů obsahuje metriky (jako dollars_sold) a klíče ke každé ze souvisejících tabulek dimenzí**
- n-D základní kostka se nazývá základní cuboid. Nejvyšší 0-D cuboid, který má hodnotu největšího stupně agregace, se nazývá apex cuboid. Mřížka cuboidů tvoří datovou kostku.

Kostka: mřížka cuboidů



0-D(apex) cuboid

1-D cuboids

2-D cuboids

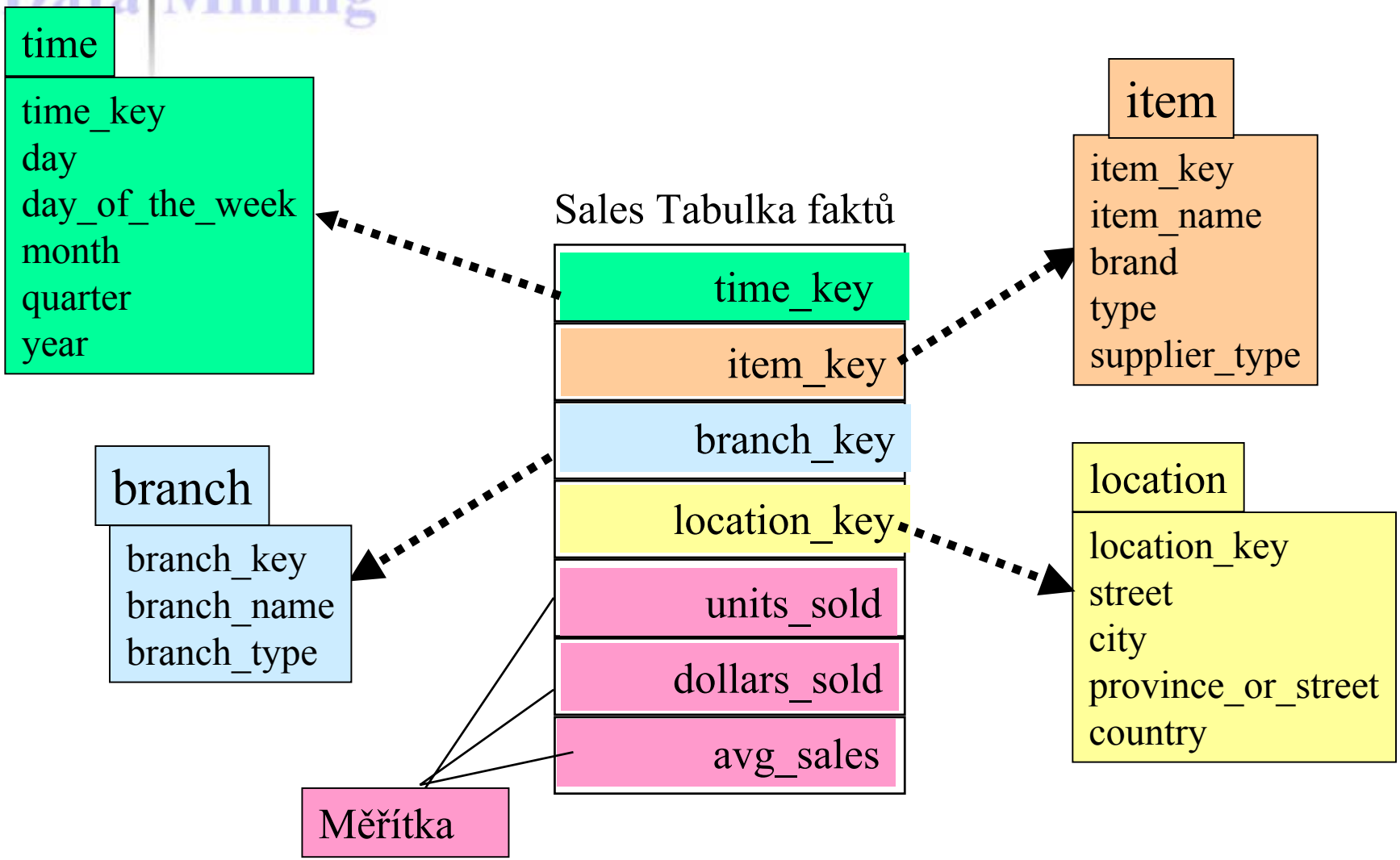
3-D cuboids

4-D(base) cuboid

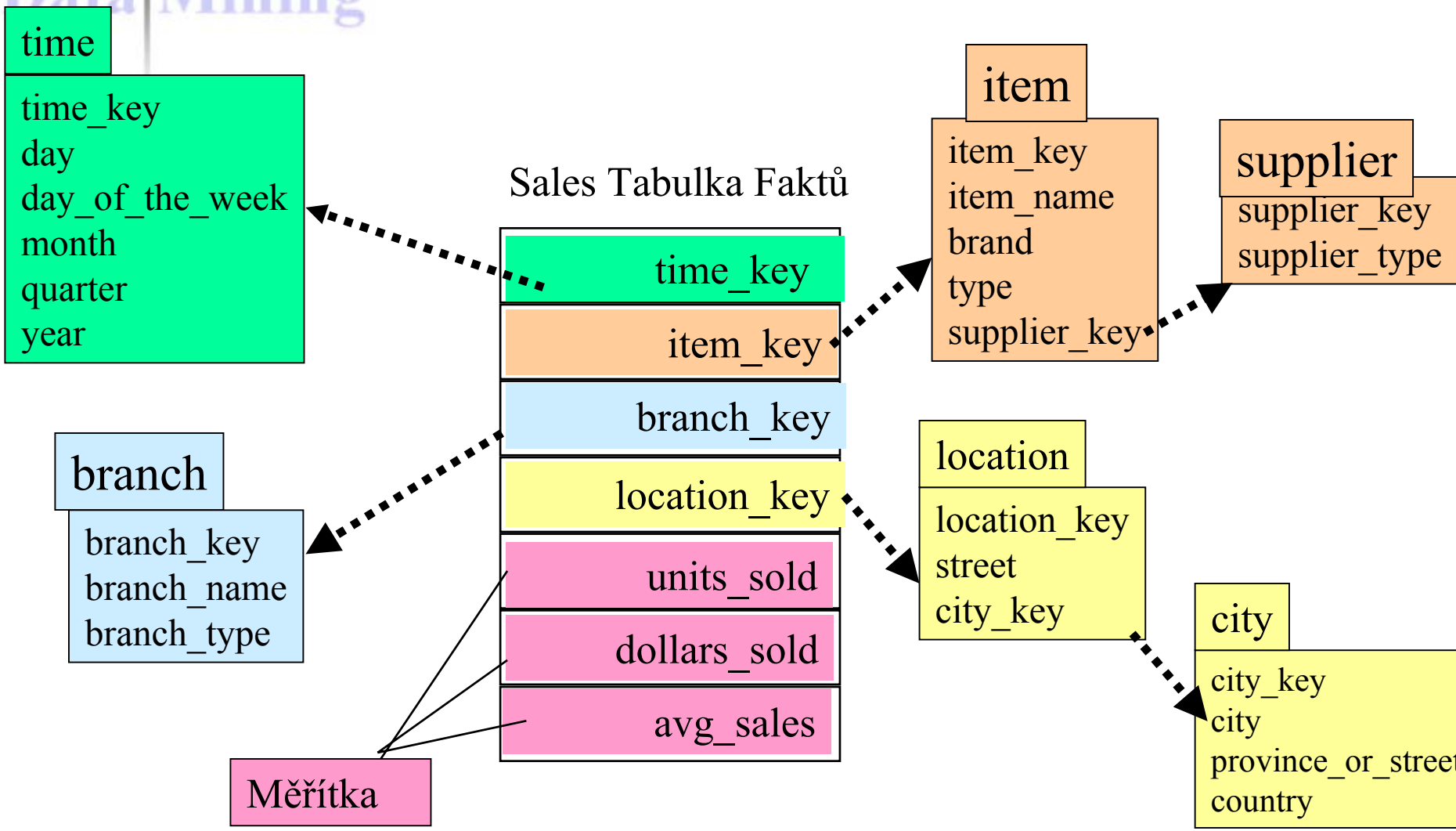
Modelování Dat. skladu

- Modelování dat. skladu: dimenze & měřítka
  - Star schéma: Tabulka faktů uprostřed spojená s množinou tabulek dimenzí
  - Snowflake schéma: zlepšení star schematu, kde některé hirearchie dimenzí jsou normalizovány do množiny menších tabulek dimenzí
  - Konstelace Faktů: Několik tabulek faktů sdílí tabulky dimenzí

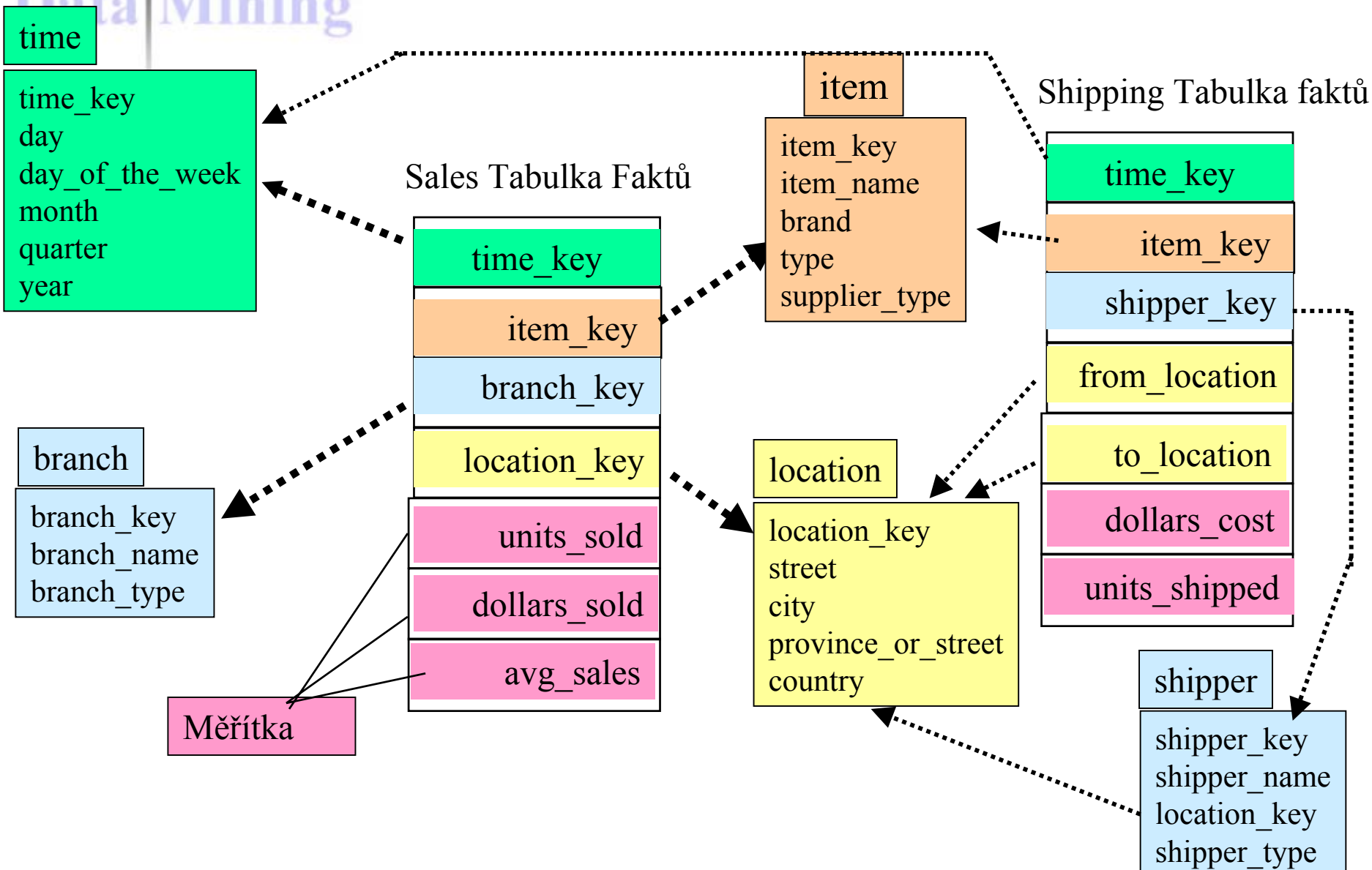
Příklad na Star schéma



Příklad na Snowflake schéma



Příklad Konstelace faktů



DMQL: Data Mining Query Language

- Definice kostky (Tabulky faktů)

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

- Definice dimenzí (Tabulek dimenzí)

  define dimension <dimension_name> as
  (<attribute_or_subdimension_list>)

- Speciální případ (Sdílené tabulky dimenzí)
  - Nejdříve definujeme první kostku
  - define dimension <dimension_name> as <dimension_name_first_time> in cube <cube_name_first_time>

Star schéma v 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)

Snowflake schéma v 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))

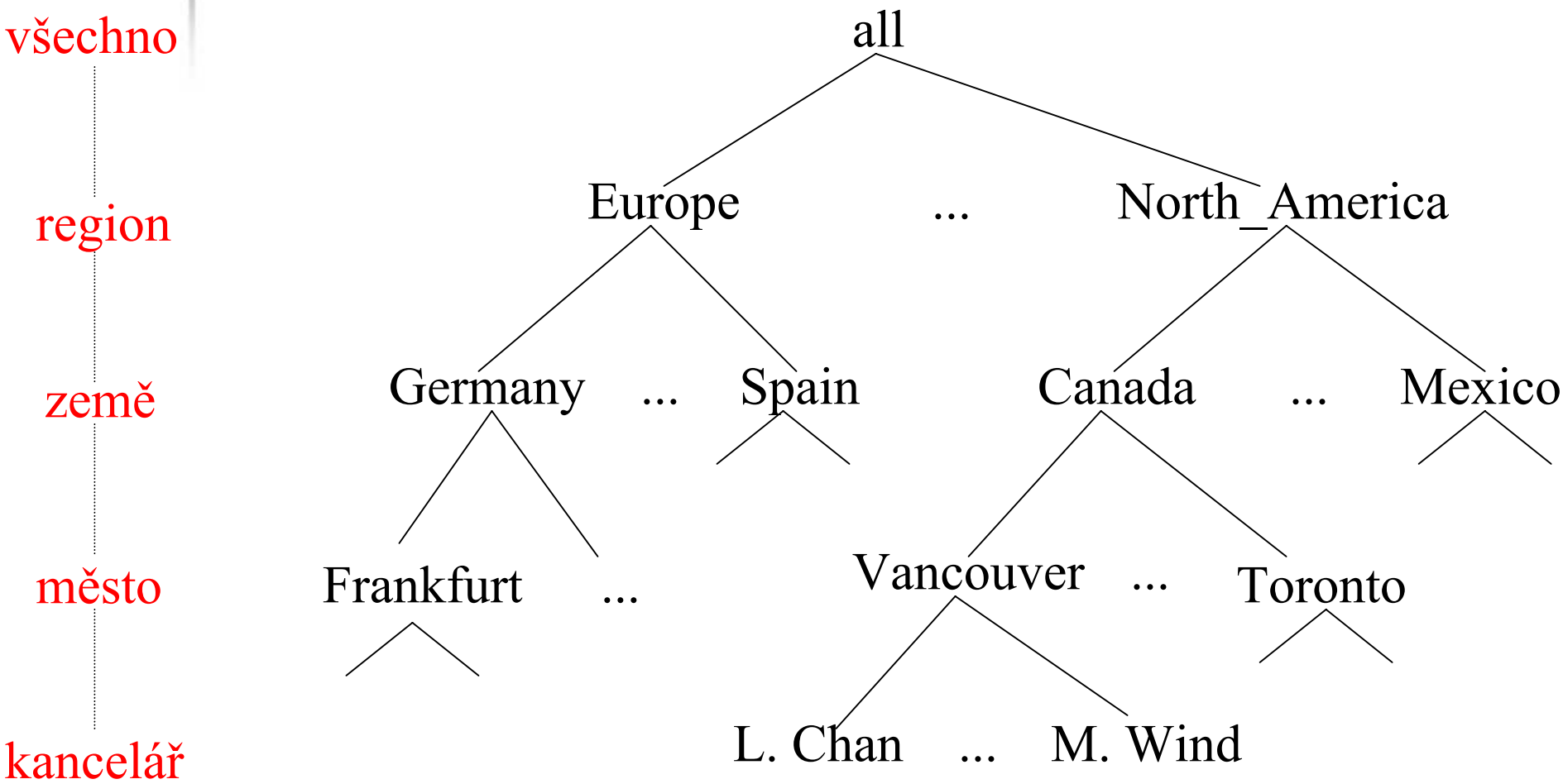
Konstelace faktů v DMQL

```
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
```

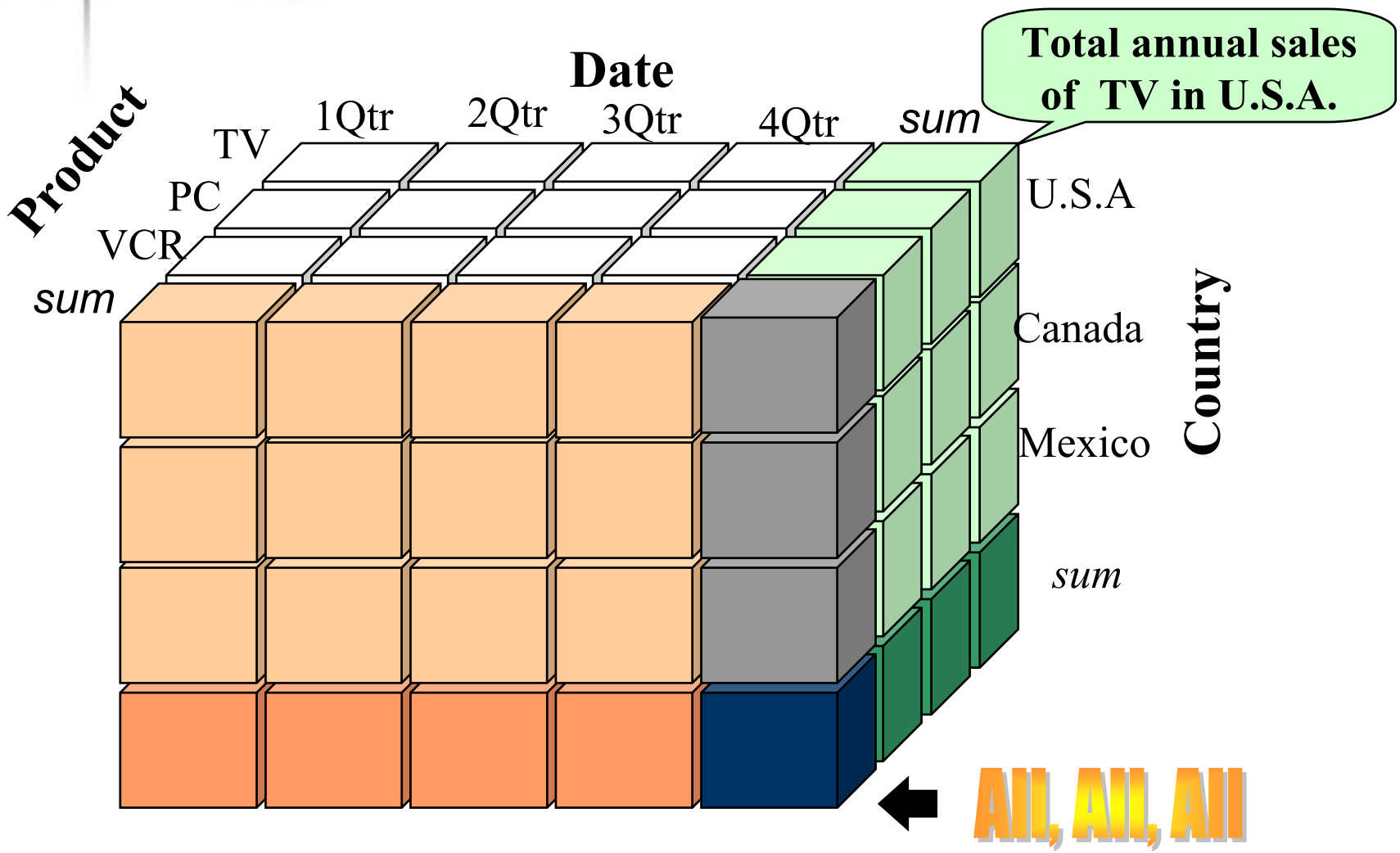
Měřítka: tři kategorie

- distributivní: pokud aplikujeme funkci na *n* podskupin, poté aplikujeme tuto funkci na výsledky podskupin a pokud aplikujeme stejnou funkci na všechny prvky a výsledek je stejný.
  - Např., count(), sum(), min(), max().
- algebraická: pokud mohou být vypočtena algebraickou funkcí s *M* argumenty, každý z nich získaný distributivní funkcí.
  - Např, avg().
- holistická: neexistuje algebraická funkce pro výpočet
  - Např., median(), rank().

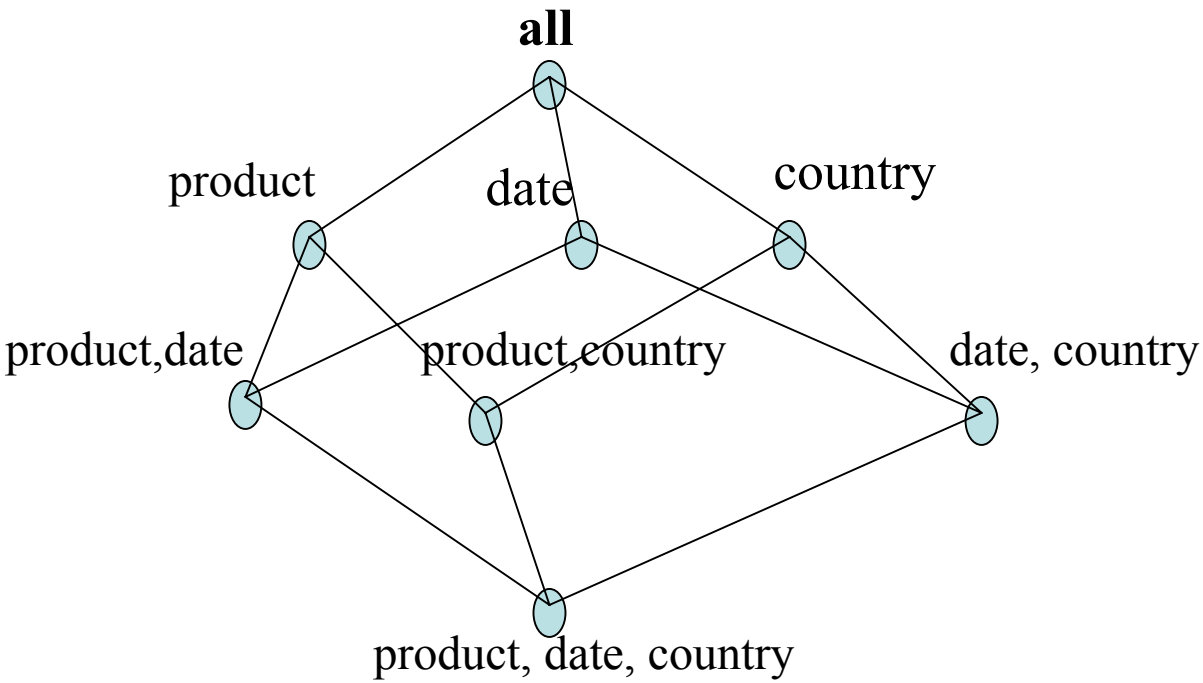
Hierarchie dimenzí



Příklad datové kostky



Cuboidy pro kostku



0-D(apex) cuboid

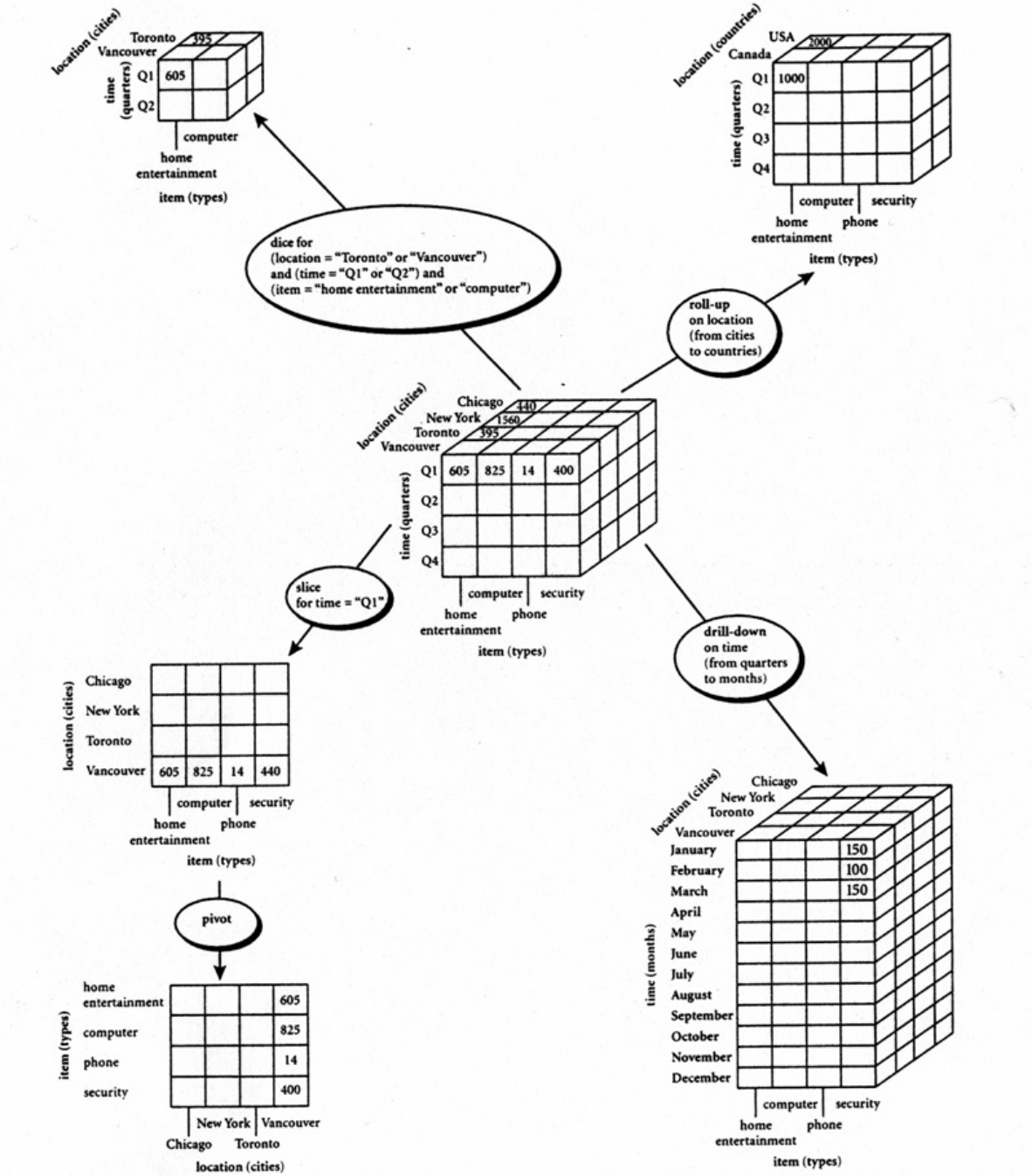
1-D cuboids

2-D cuboids

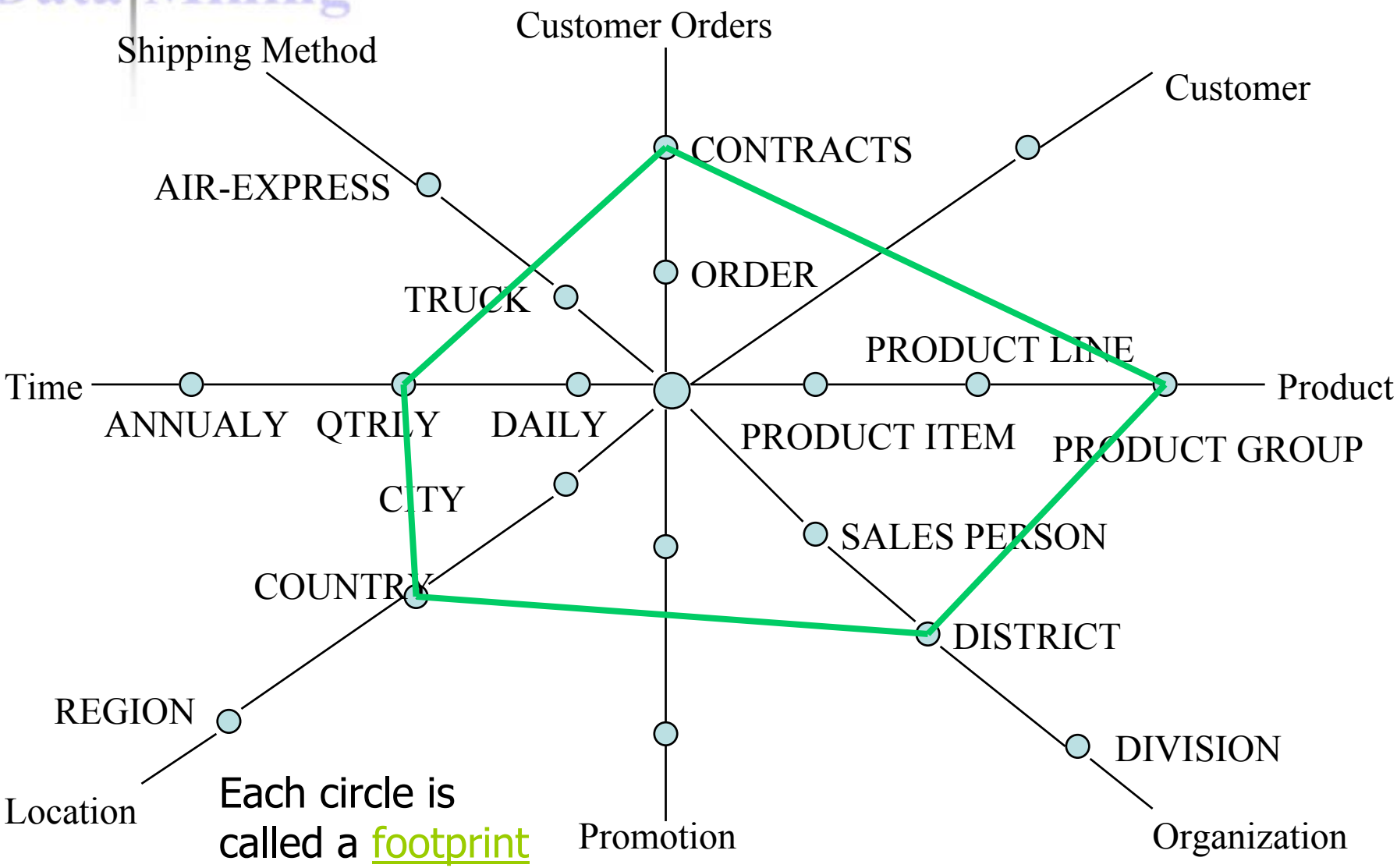
3-D(base) cuboid

Typické OLAP operace

- Roll up (drill-up): sumarizace dat
  - ***Vystoupáním v hierarchii nebo redukcí dimenzí***
- Drill down (roll down): opak roll-up
  - ***Z vyššího stupně sumarizace k nižšímu nebo detainějším datům, nebo přidání další dimenzes***
- Slice and dice:
  - ***Projekce a selekce***
- Pivot (rotate):
  - ***Reorientace kostky, vizualizace, 3D na sérii 2D ploch.***
- Other operations
  - *drill across: týká se více než jedné tabulky faktů*
  - *drill through: skrz nejnižší stupeň kostky k relačním tabulkám*


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

Star-Net Query Model



Each circle is called a footprint

Obsah

- Co to je Datový sklad?
- Multi-dimensionální datový model
- Architektura Datového skladu
- Implementace Datového skladu
- Dolování dat z Datového skladu

Systém Bussines analýzy

- Čtyři pohledy na týkající se designu datového skladu
  - Top-down pohled
    - Umožňuje selekci relevantních informací důležitých pro datový sklad
  - Pohled zdroje dat
    - Ukazuje informace které jsou snímány, ukládány a obhospodařovány operačními systémy
  - Pohled datového skladu
    - Skládá se z tabulek faktů a tabulek dimenzí
  - Business query pohled
    - Nahlíží na data v datovém skladu z pohledu koncového uživatele

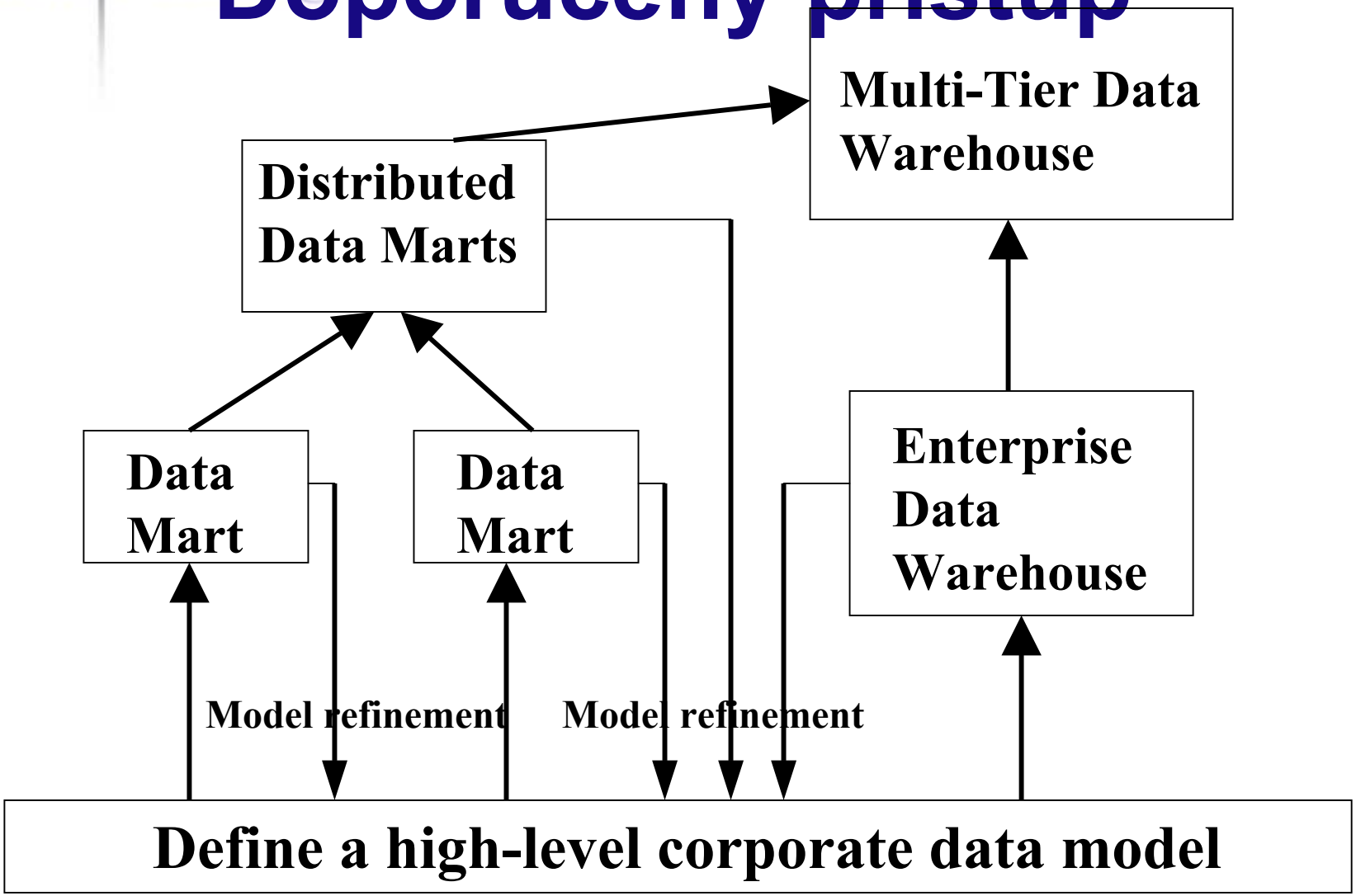
Proces designu skladu

- Top-down, bottom-up přístupy, jejich kombinace
  - **Top-down: Začíná s globálním návrhem a plánováním, problémy pro řešení jsou známé, jasné a plně pochopené**
  - **Bottom-up: Začíná s experimenty a prototypy, nižší náklady, rychlé**
- Z pohledu softwarového inženýrství
  - **Vodopád: strukturovaná a systematiská analýza v každém kroku před posunem k dalšímu kroku**
  - **Spirála: rychlé generování systému s funkčními přírustky, krátký čas oběhu**
- Typický proces návrhu datového skladu
  - Zvolit si bussinesový proces k modelování, např., objednávky, faktury, atd.
  - Zvolit si zrnitost procesu (data pro tabulku faktů)
  - Zvolit si dimenze pro každý záznam v tabulce faktů
  - Zvolit si měřítka které budou reprezentovat tabulku faktů

Modely Datových skladů

- Podnikový datový sklad (Enterprise)
  - Shromažďuje všechny informace zasahující do celé organizace
- Datový trh (Data Mart)
  - Podmnožina celopodnikových dat které mají hodnotu pro určitou skupinu uživatelů.
    - Nezávislé vs. Závislé (přímo z dat. skladu) datové trhy
- Virtuální datový sklad
  - Množina pohledů nad operačními DB
  - Pouze některé z pohledů na data jsou realizovatelné

Vývoj Datového skladu: Doporučený přístup



OLAP Servery

- Relační OLAP (ROLAP)
  - Používá relační nebo rozšířené relační DB k ukládání a správě datových skladů, stojí mezi Rel. DB a klientskou aplikací
  - Obsahují optimalizaci DB, implementaci agregace a další utility a služby
  - Větší rozšiřitelnost
- Multidimensionální OLAP (MOLAP)
  - Multidimenzionální ukládání založené na polích
  - Rychlé indexování pro předpočítaná data
- Hybridní OLAP (HOLAP)
  - Kombinace ROLAPu a MOLAPu
- Specializovaná SQL servery
  - Specializovaná podpora pro SQL dotazy nad star/snowflake schematy

Obsah

- Co to je Datový sklad?
- Multi-dimensionální datový model
- Architektura Datového skladu
- Implementace Datového skladu
- Dolování dat z Datového skladu

Efektivní výpočet kostky

- Na kostku se můžeme dívat jako na mřížku cuboidů
  - Nejnižší cuboid se nazývá base cuboid
  - Nejvyšší cuboid (apex) obsahuje pouze jednu buňku
  - Kolik cuboidů má n-dimensionální kostka s L úrovněmi?

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

- Realizace datové kostky
  - realizace každého (cuboidu) (plná realizace), žádného (bez realizace), nebo některého (částečná realizace)
  - Výběr které cuboidy realizovat
    - Založeno na velikosti, sdílení, frekvenci přístupů, atd.

Operace na kostce

- Definice a výpočet kostky v DMQL

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

  compute cube sales

- Transformace do jazyka typu SQL (s novým operátorem cube by)

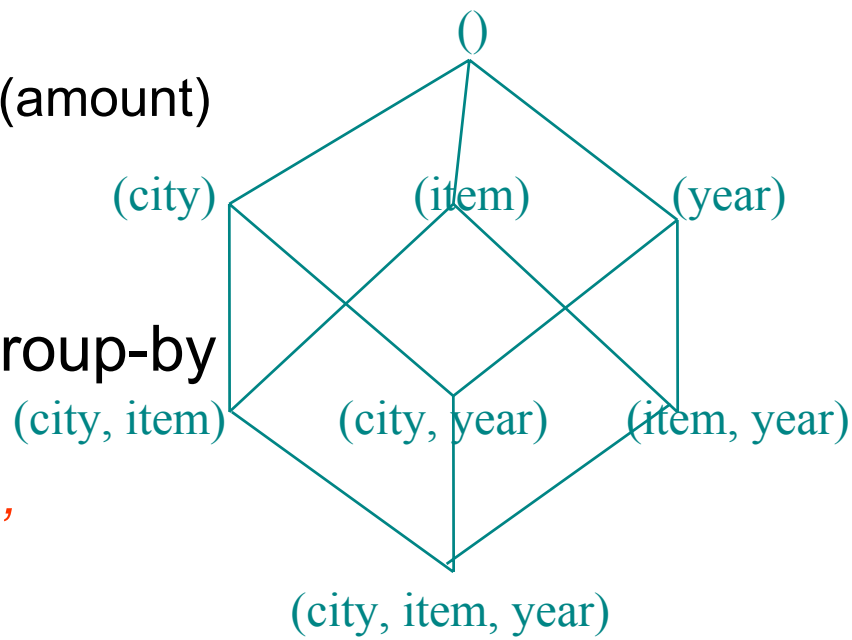
  SELECT item, city, year, SUM (amount)

  FROM SALES

  CUBE BY item, city, year

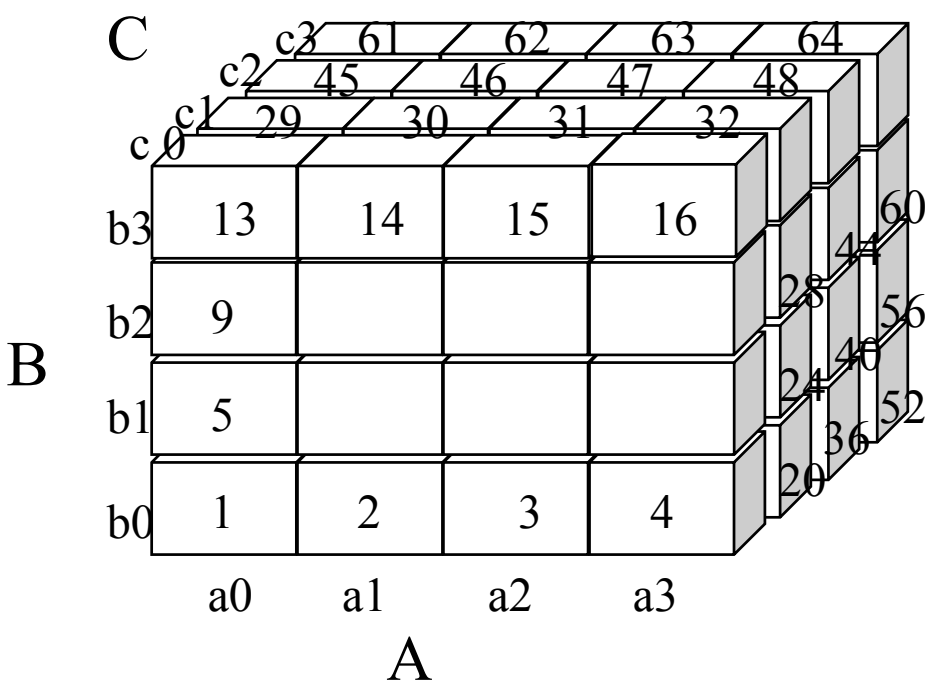
- Je třeba spočítat následující group-by

  (date, product, customer),
  (date,product),(date, customer),
  (product, customer),
  (date), (product), (customer), ()



Multicestná agregace polí

- Rozdělení polí na části (malé podkostky které se vejdou do paměti).
- Komprimované adresování řídkého pole: (id_části, offset)
- Multicestný výpočet celků v takovém pořadí, aby každá buňka byla načítána co nejméněkrát a redukoval se přístup do paměti a náklady na uložení.



Jaké je nejlepší pořadí procházení kostky?

Příklad:

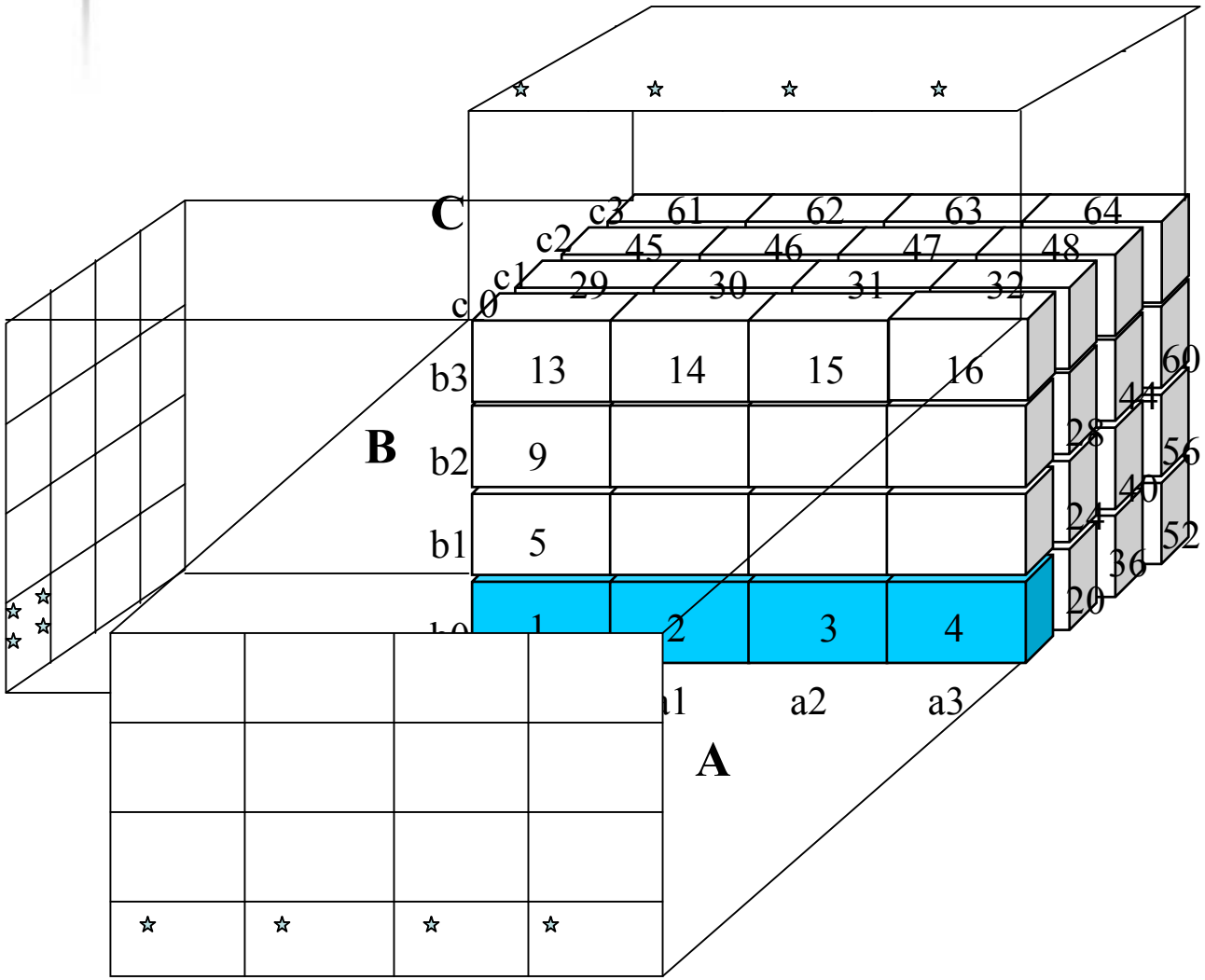
Velikost dimenzí A, B, C:

40, 400, 4000

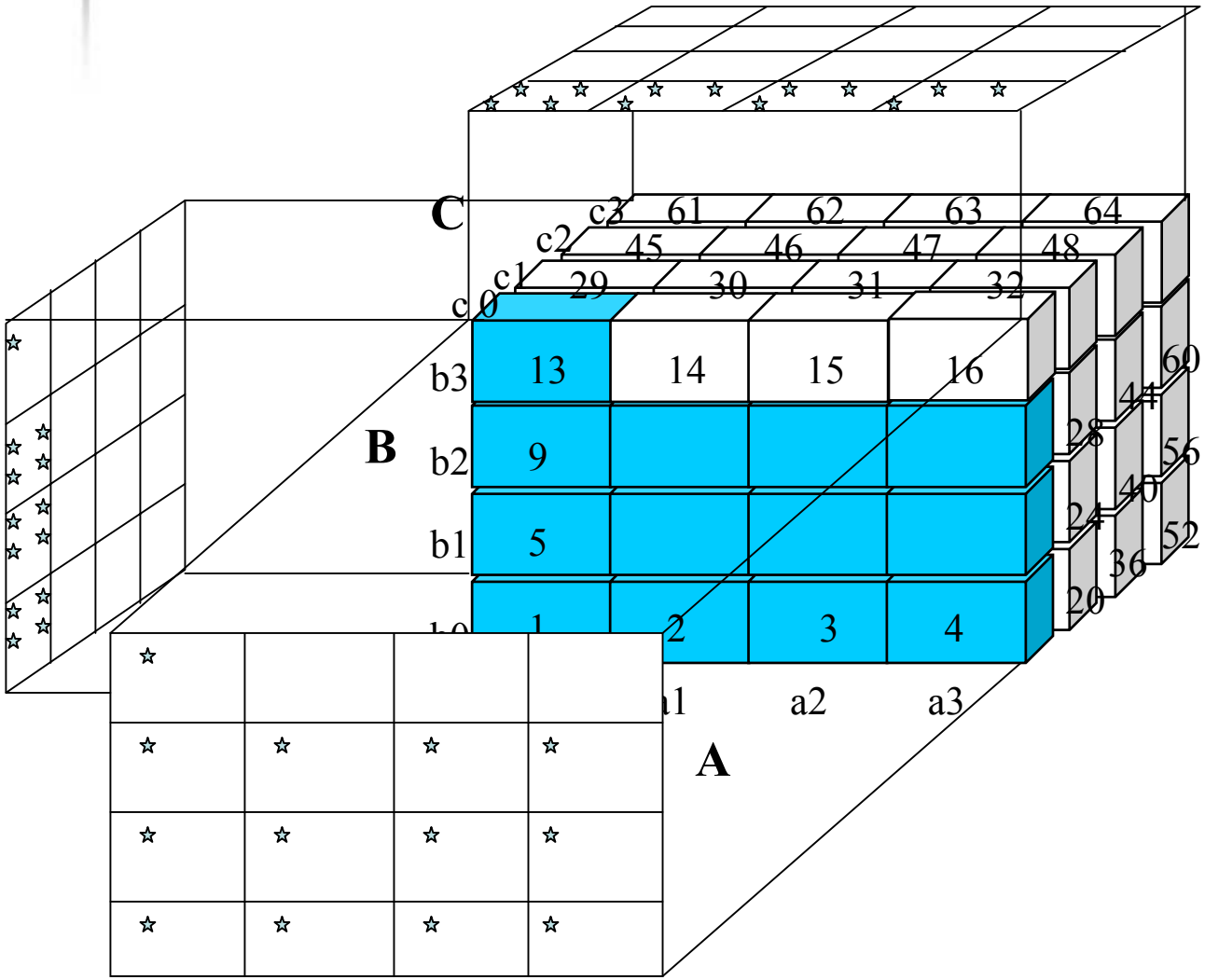
Velikost každé části A, B, C:

10, 100, 1000

Multicestná agregace polí



Multicestná agregace polí



Příklad využití paměti

- Pořadí průchodu 1, 2, 3, 4, 5, atd.
  - je třeba projít 13 částí
  - 40x400 (celá plocha AB) + 10x4000 (jeden řádek plochy AC) + 100x1000 (jedna buňka plochy BC) = 156 000 jednotek

- Pořadí průchodu 1, 17, 33, 49, 5, 21, 37, 53, atd.
  - 400x4000 (celá plocha BC), + 10x4000 (jeden řádek plochy AC) + 10x100 (jedna buňka plochy AB) = 1 641 000 jednotek

Multicestná agregace polí

- Metoda: plochy by měly být seřazeny dle velikosti od nejmenší.
  - Idea: udržovat v paměti nejmenší plochu, počítat v každém časovém bloku pouze jednu část největší plochy
- Omezení metody: výpočet vhodný pouze pro malý počet dimenzí

Indexování OLAP dat: Bitmap Index

- Index pro konkrétní řádek
- Každý řádek má svůj bitový vektor: bitové operace jsou rychlé
- Velikost bitového vektoru: počet záznamů v základní tabulce
- *i*-tý bit je nastaven pokud *i*-tý řádek základní tabulky obsahuje hodnotu pro indexovaný sloupec
- Není vhodné pro domény s vysokou kardinalitou

Base table

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

Index on Region

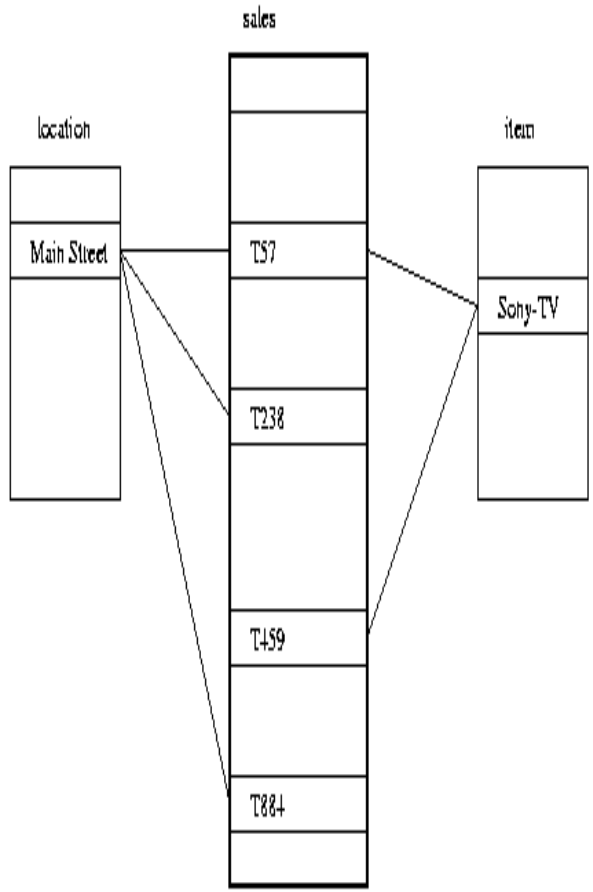
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on Type

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

Indexování OLAP dat: Join Index

- Tradiční indexování mapuje hodnoty na seznam identifikátorů záznamů
  - Urychluje relační JOIN
- V datových skladech se Join index vztahuje k hodnotám dimenzí a k řádkům v tabulce faktů
  - Např. tabulka faktů *Sales* a dvě dimenze *city* a *product*
    - join index na *city* udržuje pro každé rozdílné město seznam R-ID označujících prodej v daném městě.
  - Join indexy mohou zahrnovat několik dimenzí



Skladiště Metadat

- Metadata jsou data definující objekty datových skladů. Má následující formy:
  - Popis struktury datového skladu
    - schéma, pohledy, dimenze, hierarchie, definice odvozených dat, lokace a obsah datových tržišť
  - Operační metadata
    - Časová linie dat (historie přesouvaných dat a transformační cesty), hodnota dat (aktivní, archivovaná, nebo vyloučená), monitorovací informace (statistiky použití skladu, error reporty, audity)
  - Algoritmus používaný pro sumarizaci dat
  - Mapování z operačního prostředí na datový sklad
  - Data související s výkonem systému
    - Schéma skladu, pohled a definice odvozených dat
  - Business data
    - businesové pojmy a definice, vlastnictví dat

Nástroje a utility

- Extrakce dat:
  - Načtení dat z několika heterogenních a externích zdrojů
- Čištění dat:
  - Detekce chyb v datech a jejich oprava, pokud je to možné
- Transformace dat:
  - Konvertování dat z formátu relační DB do formátu datového skladu
- Nahrání dat:
  - Seřazení, sumarizace, výpočty pohledů, kontrola integrity a vytvoření indexů
- Obnova dat
  - Rozšíření updatů dat do datového skladu

Obsah

- Co to je Datový sklad?
- Multi-dimensionální datový model
- Architektura Datového skladu
- Implementace Datového skladu
- Dolování dat z Datového skladu

OnLine Analytical Mining: OLAM

- Proč OnLine analytické dolování?
  - Vysoká kvalita dat v datovém skladu
    - Dat. sklad obsahuje integrovaná, konzistentní data
  - Dostupné struktury pro zpracování informací z dat. skladů
    - ODBC, OLEDB, Web přístup, reportování a OLAP nástroje
  - Výzkumná analýza dat založená na OLAP
    - Dolování s drillingem, dicingem, pivotingem, atd.
  - On-line výběr funkcí pro dolování dat
    - Integrace a swapování několika dolovacích funkcí, algoritmů a úkolů
- Architectura OLAMu

Architektura OLAMu