

DATA MINING: CONCEPTS AND TECHNIQUES

Dr Amit Swamy

DATA LIFE CYCLE

The Data Analytics Lifecycle consist of

- Discovery
- Data Preparation
- Model Planning
- Model Building
- Communicate Results
- Operationalize

3 TOPIC...

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

4

WHAT IS DATA WAREHOUSE?

- Defined in many different ways, but not rigorously.
 - A decision support database that is maintained **separately** from the organization's operational database
 - Support **information processing** by providing a solid platform of consolidated, historical data for analysis.
- “A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.”—W. H. Inmon
- Data warehousing:
 - The process of constructing and using data warehouses

DATA WAREHOUSE

- **Content:**
- Data Warehouse: Central repository for integrated data.
- Supports historical analysis and business intelligence.
- ETL Process: Extract, Transform, Load.
- **Benefits:**
- Consistent and reliable data
- Improved decision making

6 DATA WAREHOUSE—SUBJECT-ORIENTED

- Organized around major subjects, such as **customer, product, sales**.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- Provide **a simple and concise** view around particular subject issues by **excluding data that are not useful in the decision support process**.

7 DATA WAREHOUSE—INTEGRATED

- Constructed by integrating multiple, heterogeneous data sources
 - relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques are applied.
 - Ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources
 - E.g., Hotel price: currency, tax, breakfast covered, etc.
 - When data is moved to the warehouse, it is converted.

8 DATA WAREHOUSE—TIME VARIANT

- The time horizon for the data warehouse is significantly longer than that of operational systems.
 - Operational database: current value data.
 - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
 - Contains an element of time, explicitly or implicitly
 - But the key of operational data may or may not contain “time element”.

9 DATA WAREHOUSE—NON-VOLATILE

- A **physically separate store** of data transformed from the operational environment.
- Operational **update of data does not occur** in the data warehouse environment.
 - Does not require transaction processing, recovery, and concurrency control mechanisms
 - Requires only two operations in data accessing:
 - *initial loading of data* and *access of data*.

DATA WAREHOUSE VS. HETEROGENEOUS DBMS

10

- Traditional heterogeneous DB integration:
 - Build **wrappers/mediators** on top of heterogeneous databases
 - **Query driven** approach
 - When a query is posed to a client site, a meta-dictionary is used to translate the query into queries appropriate for individual heterogeneous sites involved, and the results are integrated into a global answer set
 - Complex information filtering, compete for resources
- Data warehouse: **update-driven**, high performance
 - Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis

DATA WAREHOUSE VS. OPERATIONAL DBMS



- OLTP (on-line transaction processing)
 - Major task of traditional relational DBMS
 - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
 - Major task of data warehouse system
 - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
 - User and system orientation: customer vs. market
 - Data contents: current, detailed vs. historical, consolidated
 - Database design: ER + application vs. star + subject
 - View: current, local vs. evolutionary, integrated
 - Access patterns: update vs. read-only but complex queries

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 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

I3 DATA ARCHITECTURE..

- **Bottom Tier:** Data Source layer (Operational DBs, external data)
- **Middle Tier:** OLAP server for data processing
- **Top Tier:** Front-end tools for analysis and visualization

14 WHY SEPARATE DATA WAREHOUSE?

- High performance for both systems
 - DBMS—tuned for OLTP: access methods, indexing, concurrency control, recovery
 - Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation.
- Different functions and different data:
 - missing data: Decision support requires historical data which operational DBs do not typically maintain
 - data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
 - data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled

15

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

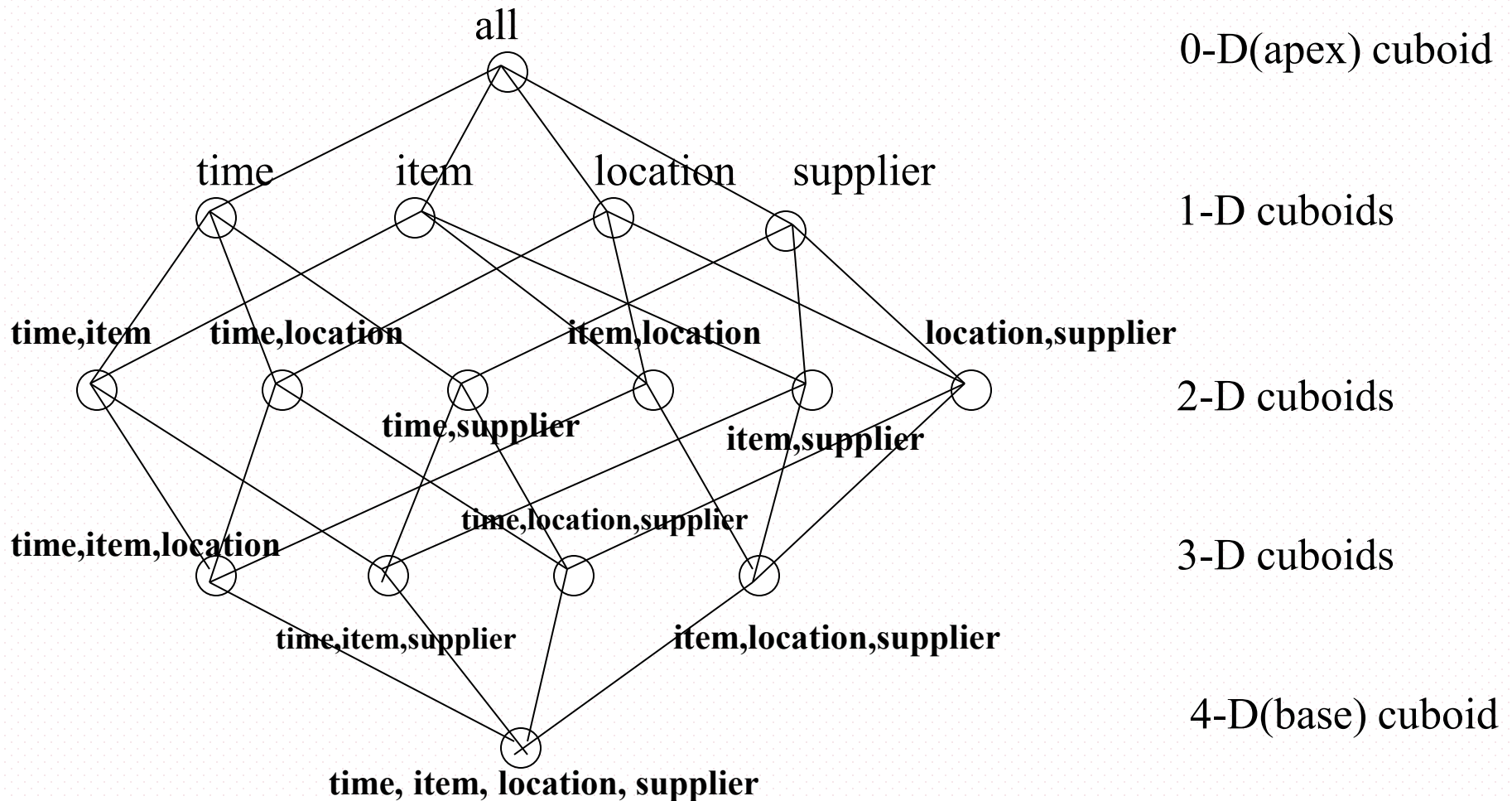
FROM TABLES AND SPREADSHEETS TO DATA CUBES

16

- A data warehouse is based on a **multidimensional data model** which views data in the form of a data cube
- A data cube, such as **sales**, allows data to be modeled and viewed in multiple dimensions
 - Dimension tables, such as **item** (**item_name**, **brand**, **type**), or **time**(**day**, **week**, **month**, **quarter**, **year**)
 - Fact table contains measures (such as **dollars_sold**) and keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a **base cuboid**. The top most 0-D cuboid, which holds the highest-level of summarization, is called the **apex cuboid**. The lattice of cuboids forms a **data cube**.

CUBE: A LATTICE OF CUBOIDS

17

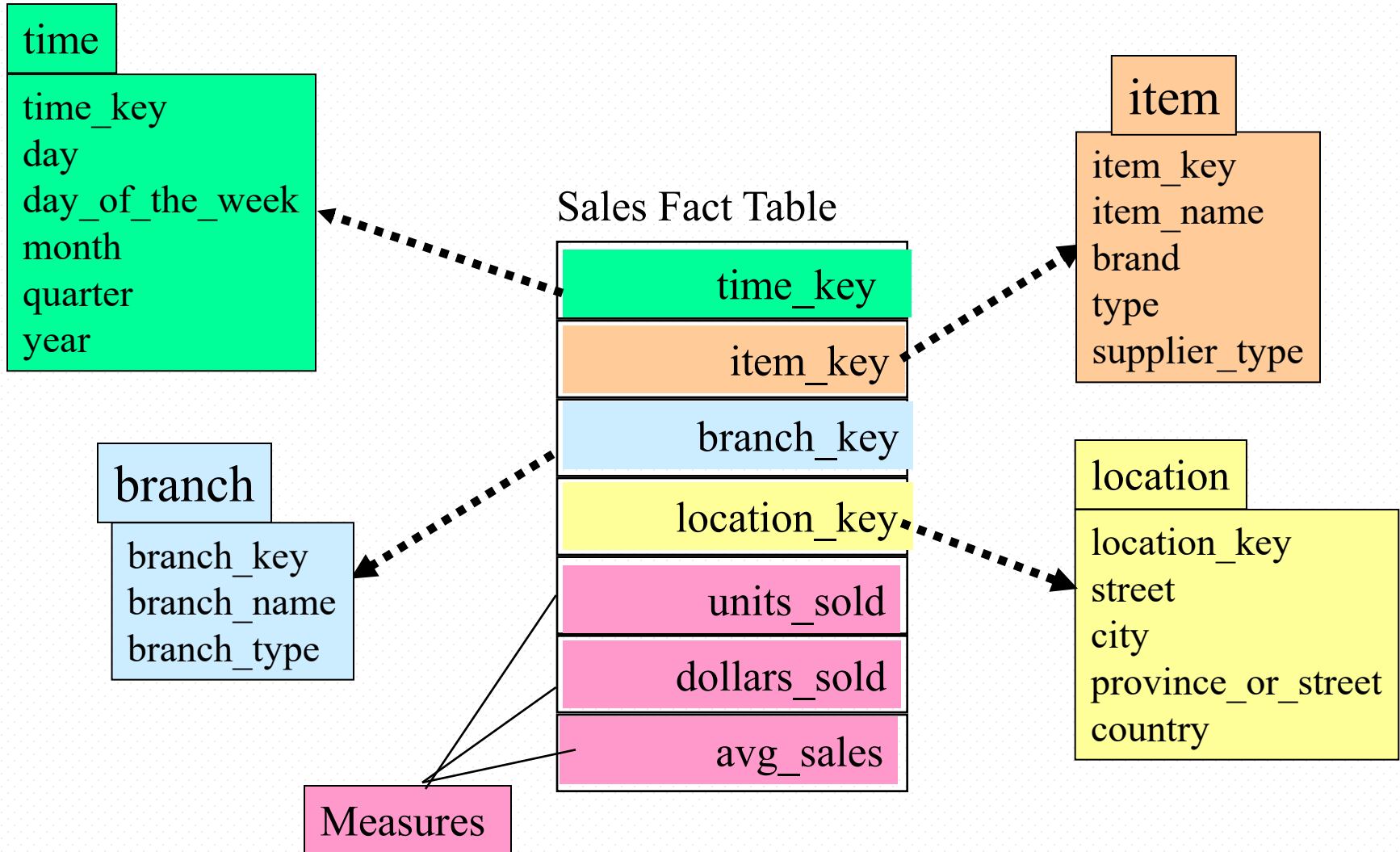


18

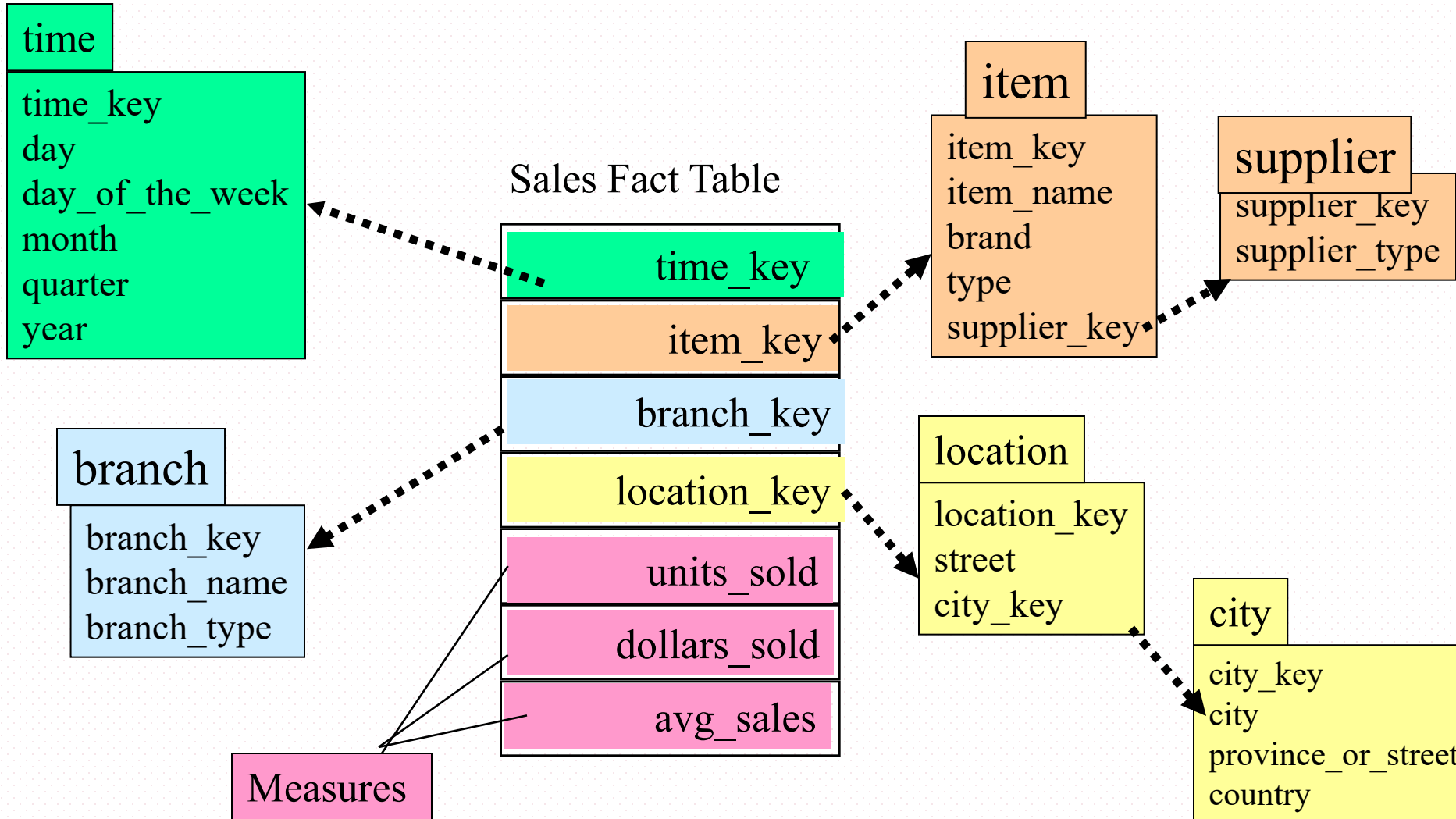
CONCEPTUAL MODELING OF DATA WAREHOUSES

- Modeling data warehouses: dimensions & measures
 - Star schema: A fact table in the middle connected to a set of dimension tables
 - Snowflake schema: A refinement of star schema where some dimensional hierarchy is **normalized** into a set of smaller dimension tables, forming a shape similar to snowflake
 - Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called **galaxy schema** or fact constellation

19 EXAMPLE OF STAR SCHEMA

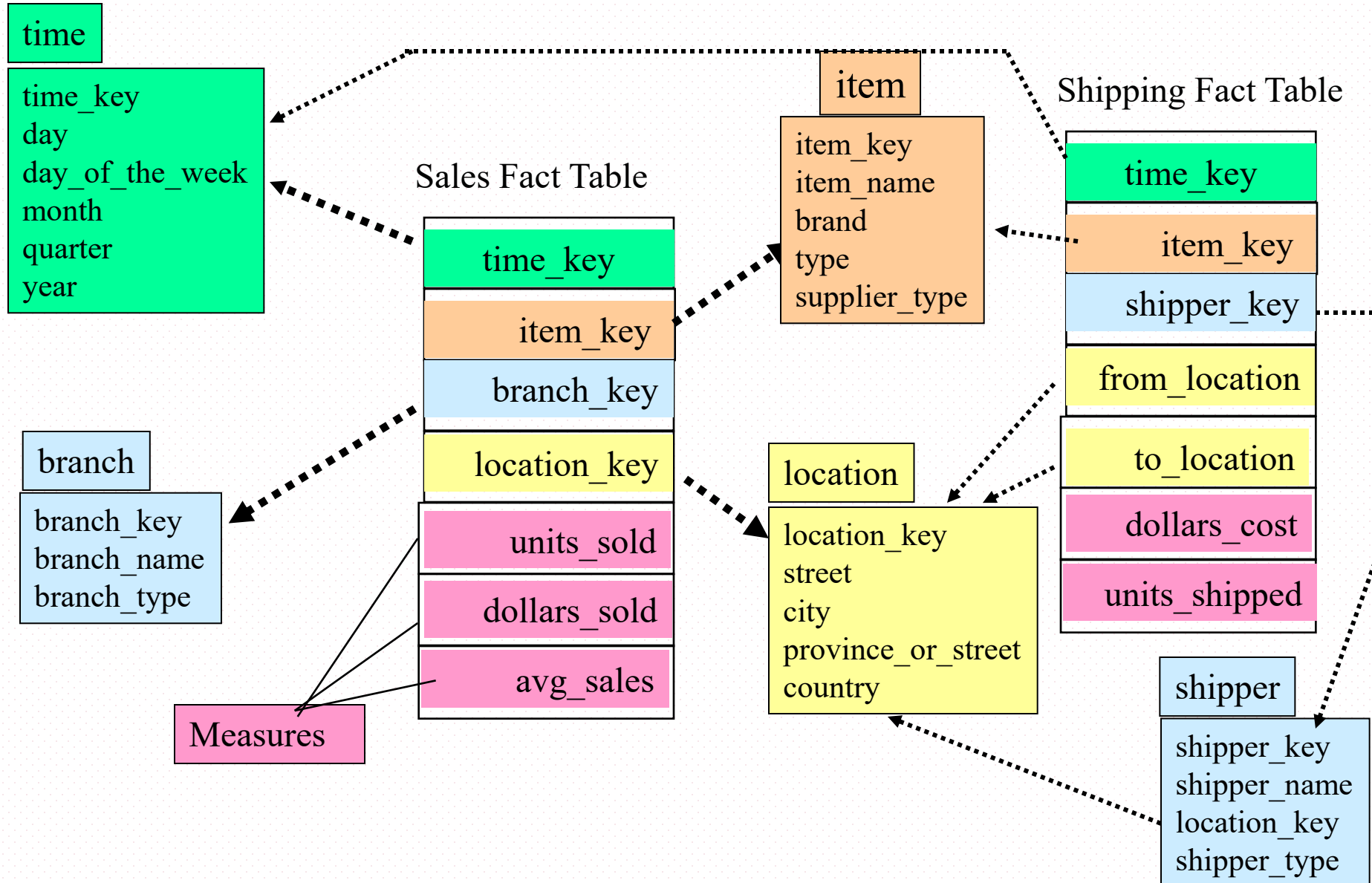


20 EXAMPLE OF SNOWFLAKE SCHEMA



EXAMPLE OF FACT CONSTELLATION

21



22

A DATA MINING QUERY LANGUAGE, DMQL: LANGUAGE PRIMITIVES

- Cube Definition (Fact Table)

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

- Dimension Definition (Dimension Table)

```
define dimension <dimension_name> as  
    (<attribute_or_subdimension_list>)
```

- Special Case (Shared Dimension Tables)

- First time as “cube definition”
- ```
define dimension <dimension_name> as
 <dimension_name_first_time> in cube
 <cube_name_first_time>
```

## 23

**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)

## 24

**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))

# DEFINING A FACT CONSTELLATION IN DMQL

Data Mining: Concepts and Techniques

July 15, 2005

25

```
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
```

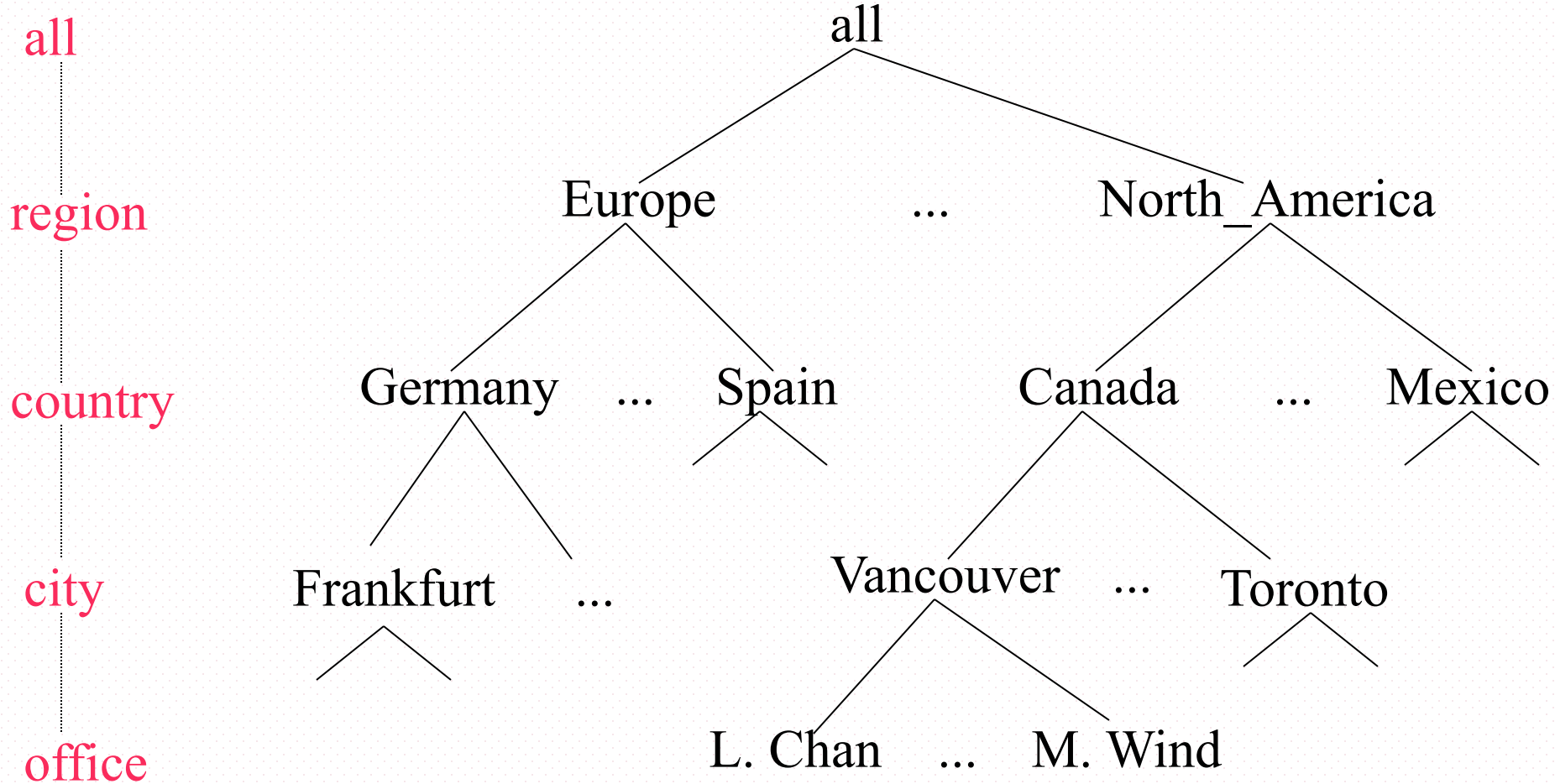
## 26

## MEASURES: THREE CATEGORIES

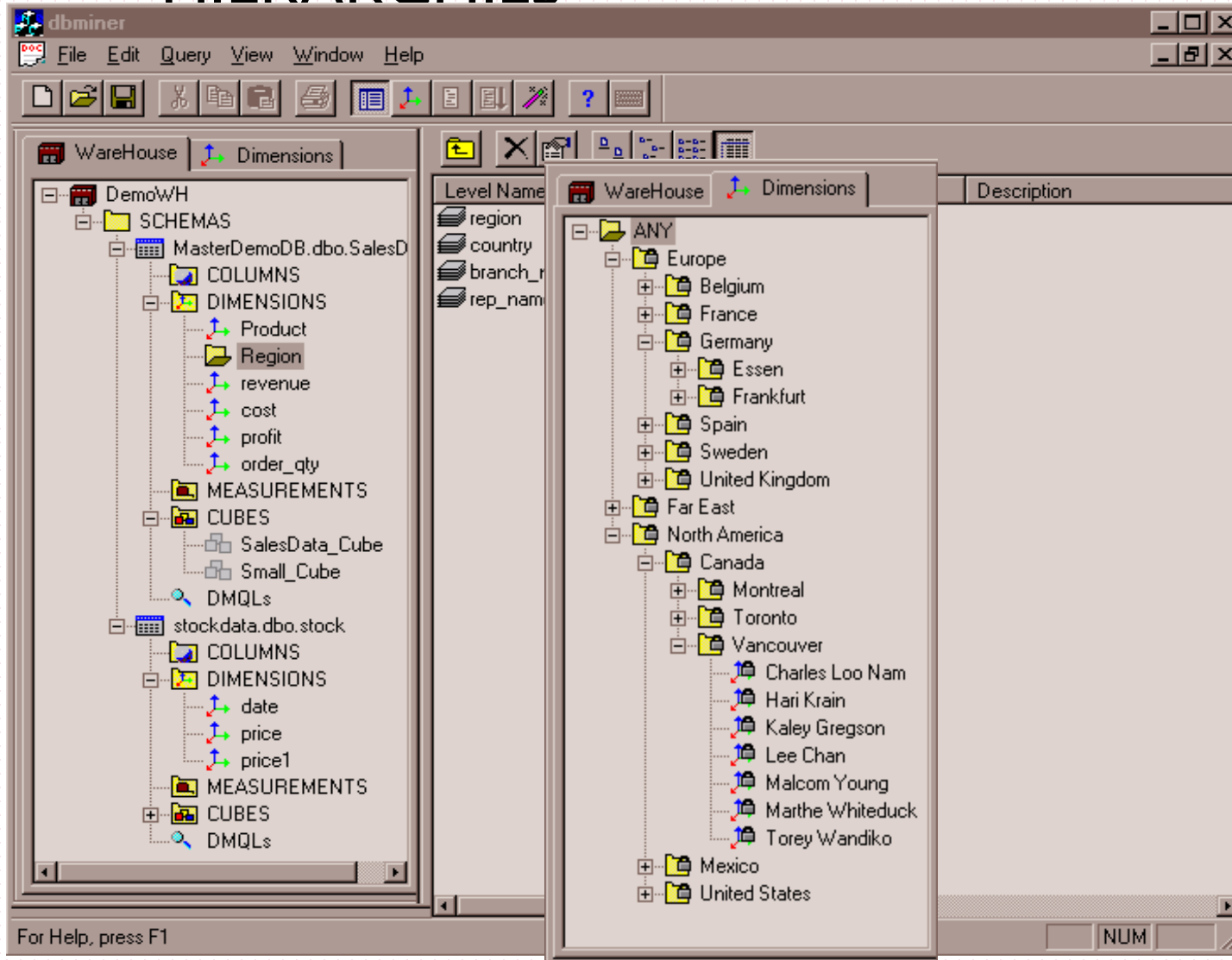
- distributive: if the result derived by applying the function to  $n$  aggregate values is the same as that derived by applying the function on all the data without partitioning.
  - E.g., `count()`, `sum()`, `min()`, `max()`.
- algebraic: if it can be computed by an algebraic function with  $M$  arguments (where  $M$  is a bounded integer), each of which is obtained by applying a distributive aggregate function.
  - E.g., `avg()`, `min_N()`, `standard_deviation()`.
- holistic: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., `median()`, `mode()`, `rank()`.

# A CONCEPT HIERARCHY: DIMENSION

## 27 (LOCATION)



# VIEW OF WAREHOUSES AND HIERARCHIES



hierarchies

hy

< quarter;

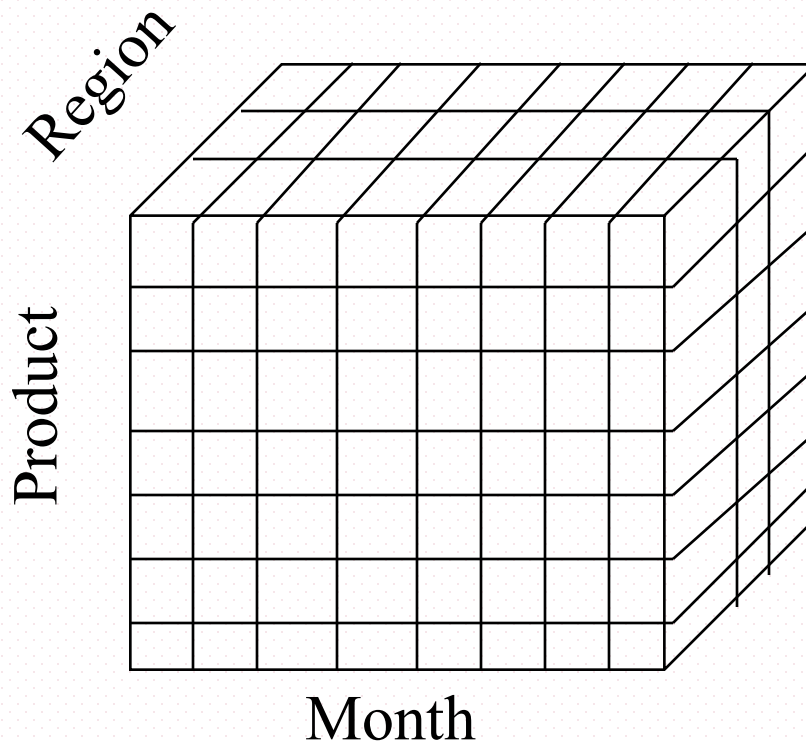
ar

ierarchy

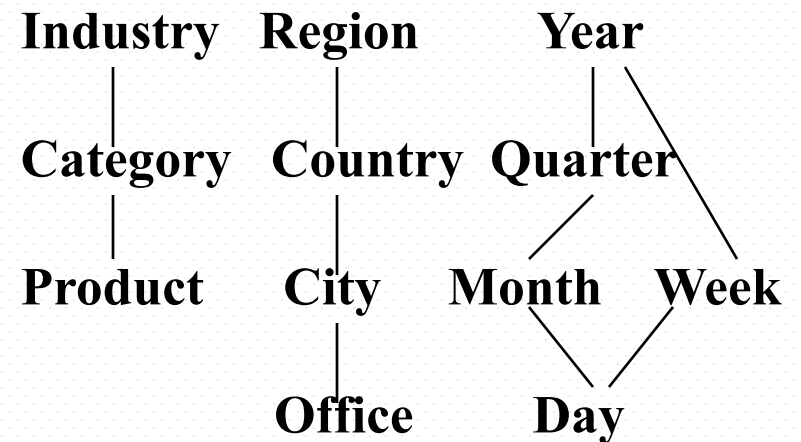
pensive

## 29 MULTIDIMENSIONAL DATA

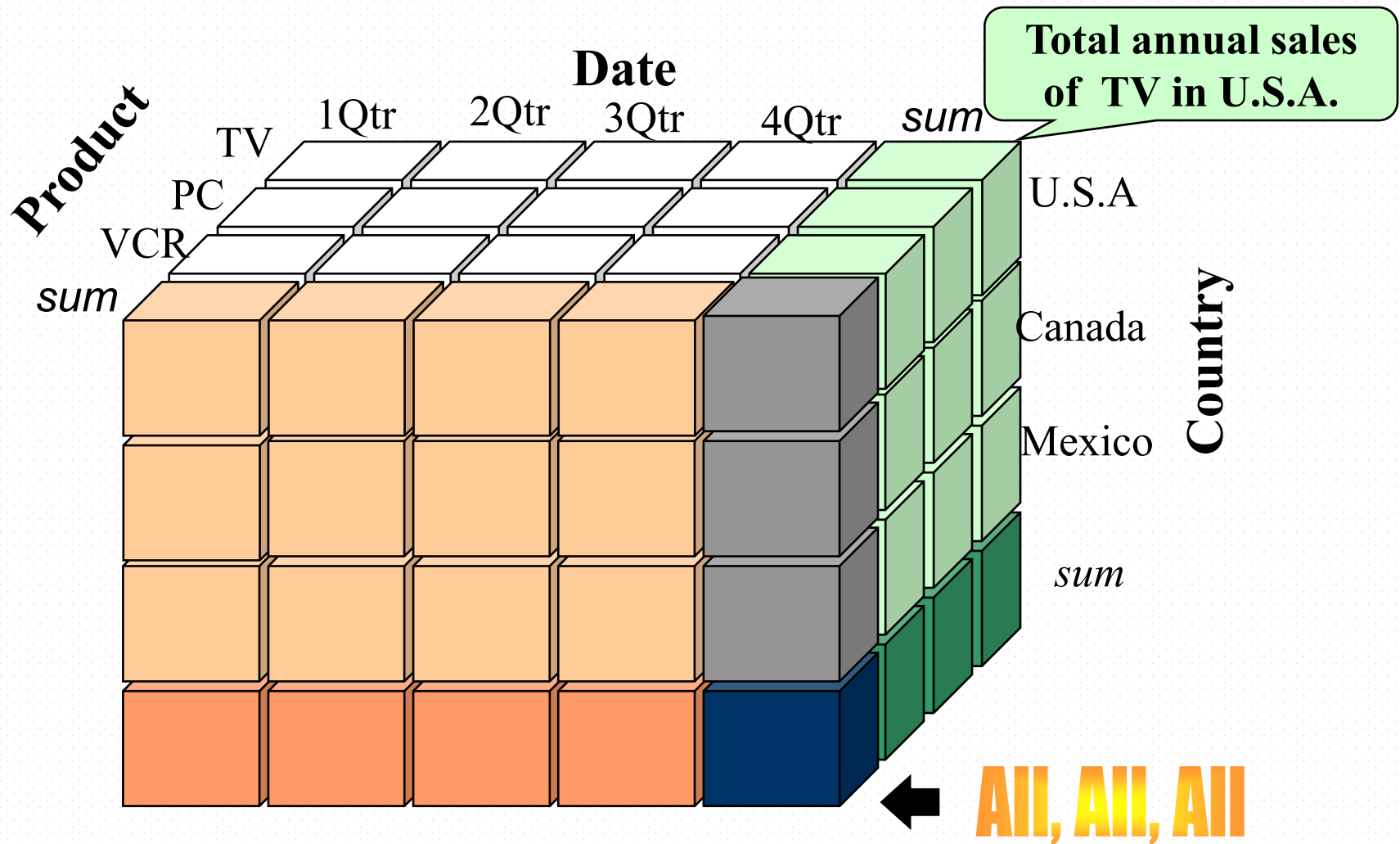
- Sales volume as a function of product, month, and region



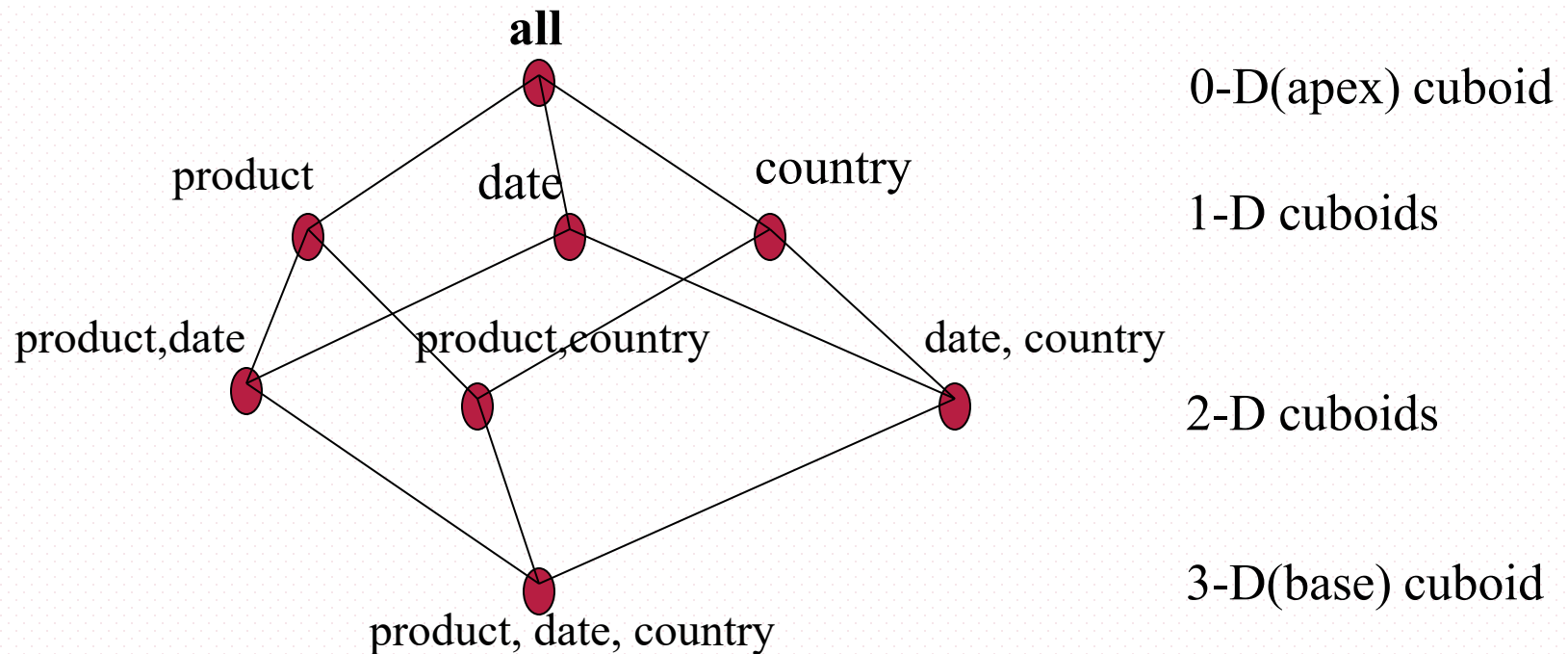
**Dimensions: Product, Location, Time**  
**Hierarchical summarization paths**



# 30 A SAMPLE DATA CUBE



# 3 | CUBOIDS CORRESPONDING TO THE CUBE

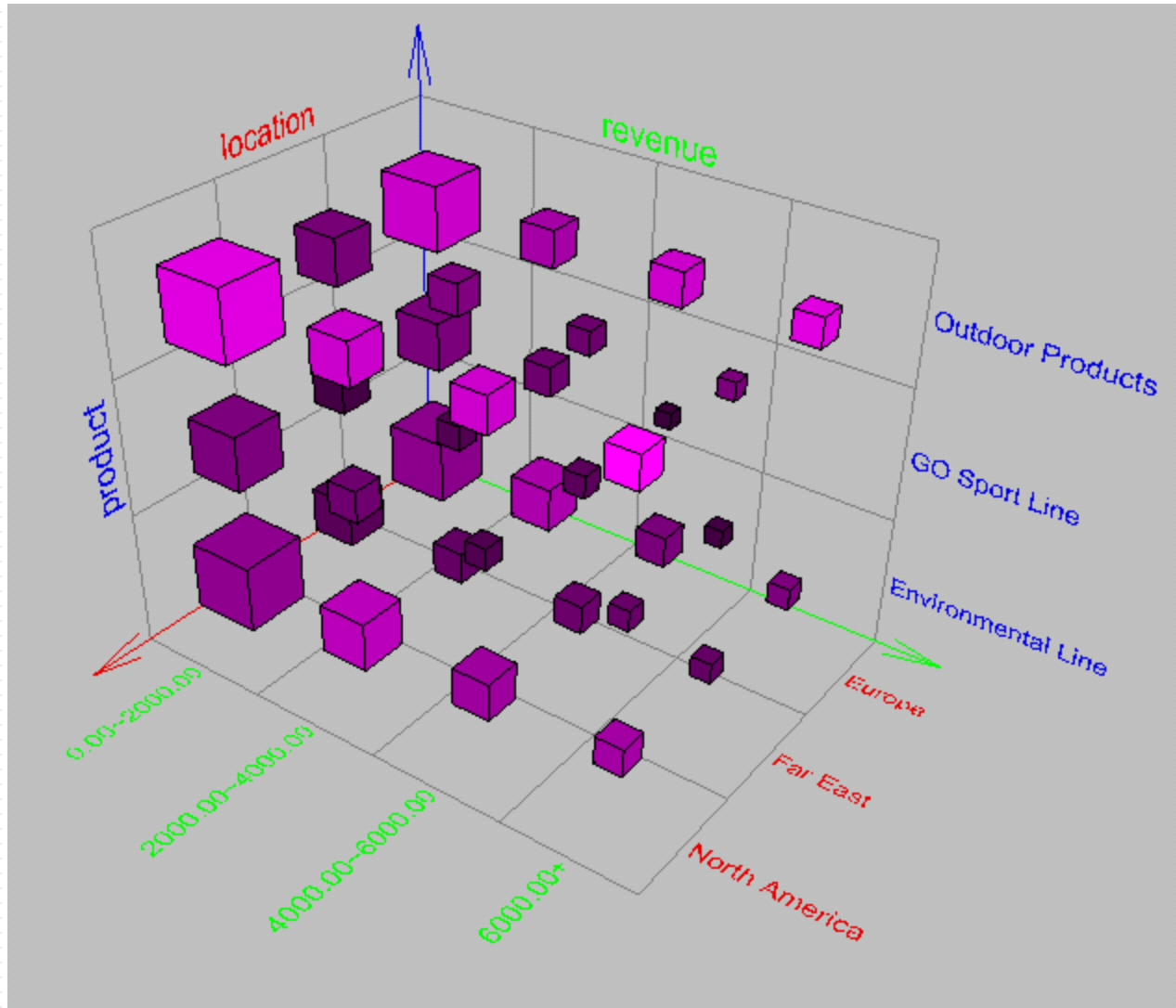


# BROWSING A DATA CUBE

Data Mining: Concepts and Techniques

July 15, 2025

32



n  
bilities  
manipulation

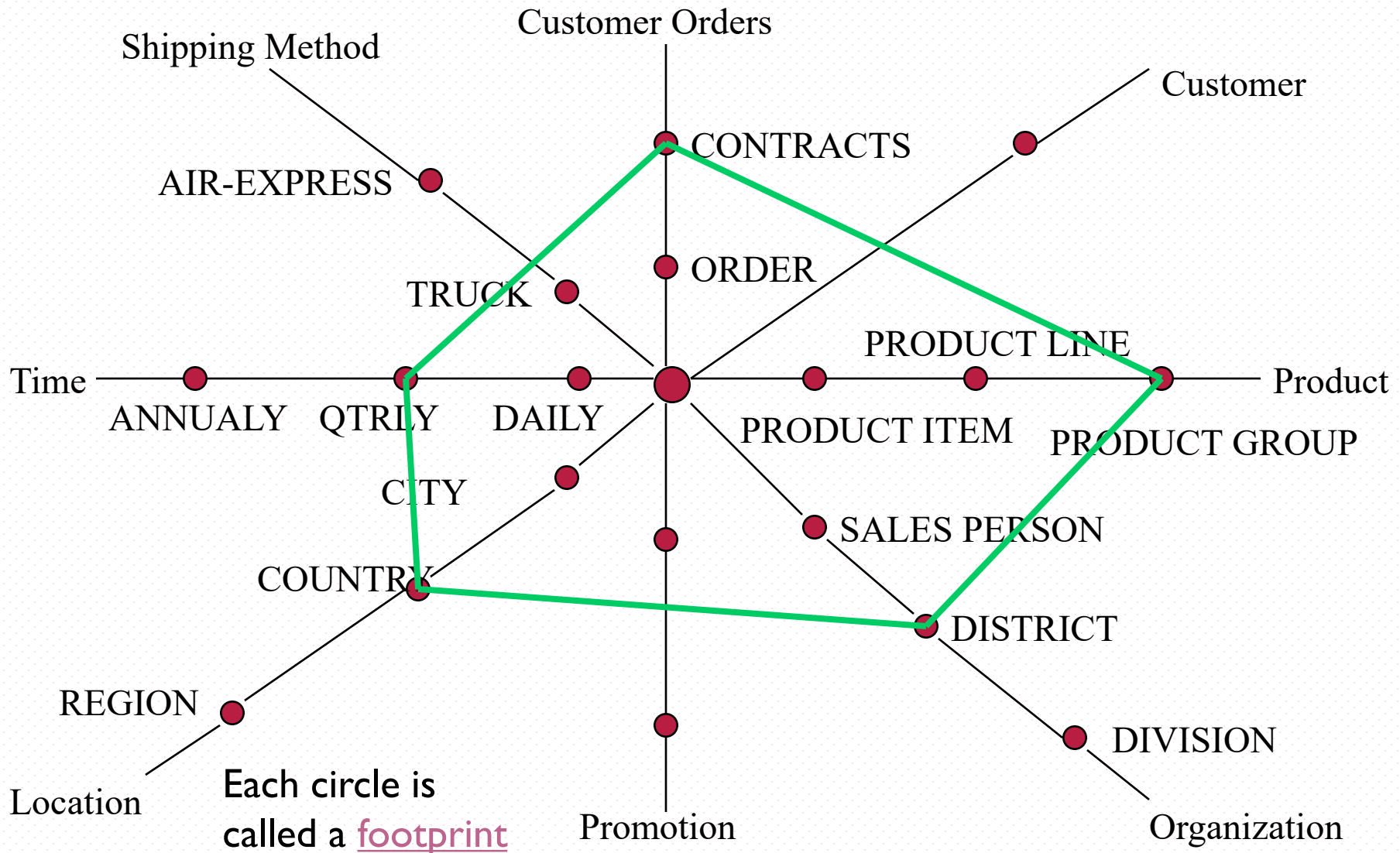
# TYPICAL OLAP OPERATIONS

## 33

- **Roll up (drill-up):** summarize data
  - *by climbing up hierarchy or by dimension reduction*
- **Drill down (roll down):** reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- **Slice and dice:**
  - *project and select*
- **Pivot (rotate):**
  - *reorient the cube, visualization, 3D to series of 2D planes.*
- Other operations
  - **drill across:** *involving (across) more than one fact table*
  - **drill through:** *through the bottom level of the cube to its back-end relational tables (using SQL)*

# A STAR-NET QUERY MODEL

34



# CHAPTER 2: DATA WAREHOUSING AND OLAP TECHNOLOGY FOR DATA MINING

## 35

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

## 36 DESIGN OF A DATA WAREHOUSE: A BUSINESS ANALYSIS FRAMEWORK

- Four views regarding the design of a data warehouse
  - **Top-down view**
    - allows selection of the relevant information necessary for the data warehouse
  - **Data source view**
    - exposes the information being captured, stored, and managed by operational systems
  - **Data warehouse view**
    - consists of fact tables and dimension tables
  - **Business query view**
    - sees the perspectives of data in the warehouse from the view of end-user

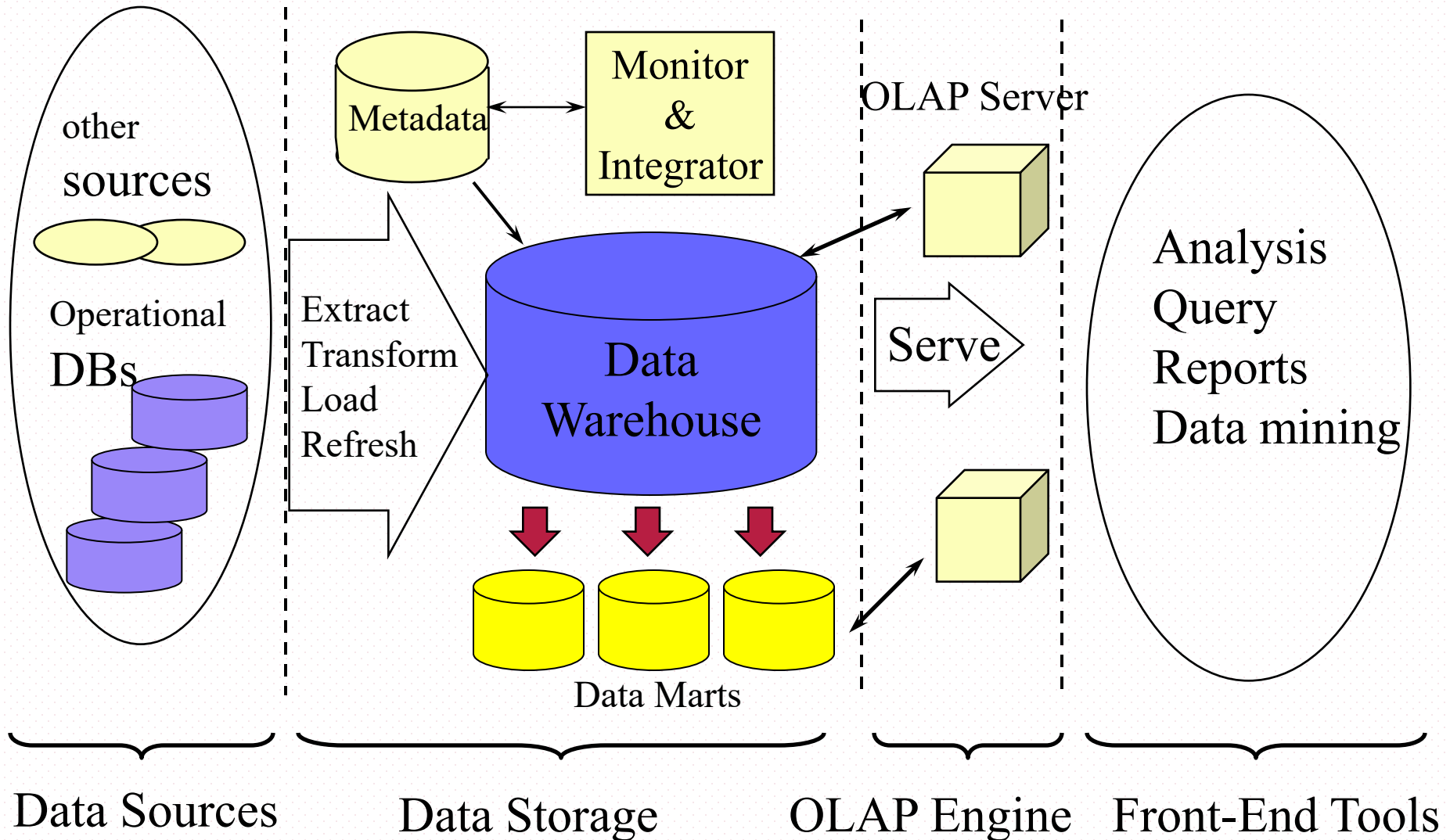
# DATA WAREHOUSE DESIGN PROCESS

## 37

- Top-down, bottom-up approaches or a combination of both
  - Top-down: Starts with overall design and planning (mature)
  - Bottom-up: Starts with experiments and prototypes (rapid)
- From software engineering point of view
  - Waterfall: structured and systematic analysis at each step before proceeding to the next
  - Spiral: rapid generation of increasingly functional systems, short turn around time, quick turn around
- Typical data warehouse design process
  - Choose a **business process** to model, e.g., orders, invoices, etc.
  - Choose the **grain** (*atomic level of data*) of the business process
  - Choose the **dimensions** that will apply to each fact table record
  - Choose the **measure** that will populate each fact table record

38

# Multi-Tiered Architecture



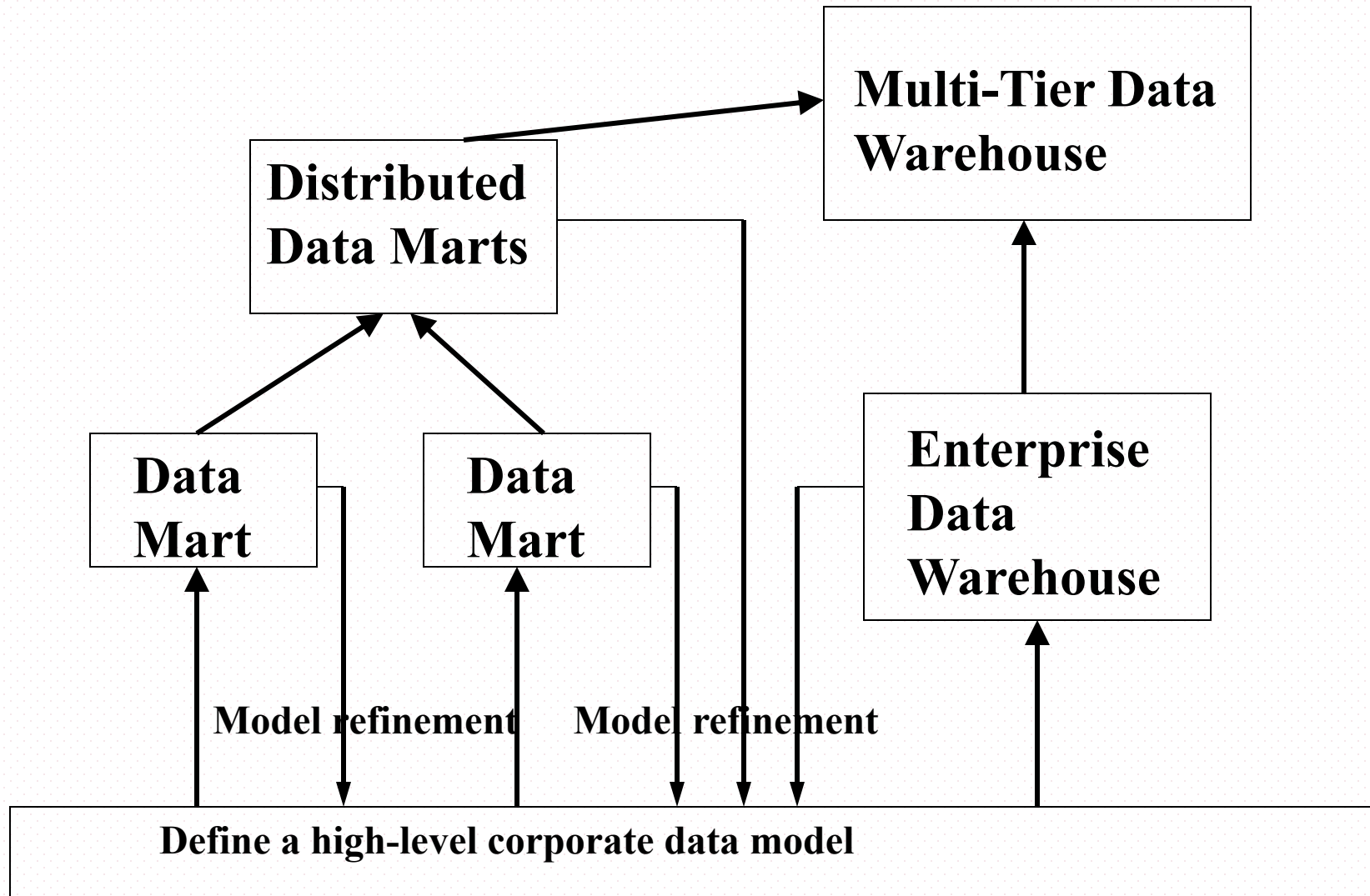
# THREE DATA WAREHOUSE MODELS

39

- **Enterprise warehouse**
  - collects all of the information about subjects spanning the entire organization
- **Data Mart**
  - a subset of corporate-wide data that is of value to a specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
- **Virtual warehouse**
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized

40

# DATA WAREHOUSE DEVELOPMENT: A RECOMMENDED APPROACH



## 4 | OLAP SERVER ARCHITECTURES

- Relational OLAP (ROLAP)
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware to support missing pieces
  - Include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
  - greater scalability
- Multidimensional OLAP (MOLAP)
  - Array-based multidimensional storage engine (sparse matrix techniques)
  - fast indexing to pre-computed summarized data
- Hybrid OLAP (HOLAP)
  - User flexibility, e.g., low level: relational, high-level: array
- Specialized SQL servers
  - specialized support for SQL queries over star/snowflake schemas

# CHAPTER 2: DATA WAREHOUSING AND OLAP TECHNOLOGY FOR DATA MINING

42

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

## 43 EFFICIENT DATA CUBE COMPUTATION

- Data cube can be viewed as a lattice of cuboids
  - The bottom-most cuboid is the base cuboid
  - The top-most cuboid (apex) contains only one cell
  - How many cuboids in an n-dimensional cube with L levels?

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

- Materialization of data cube
  - Materialize every (cuboid) (full materialization), none (no materialization), or some (partial materialization)
  - Selection of which cuboids to materialize
    - Based on size, sharing, access frequency, etc.

# 44 CUBE OPERATION

- Cube definition and computation in DMQL

**define cube** sales[item, city, year]: sum(sales\_in\_dollars)

**compute cube** sales

- Transform it into a SQL-like language (with a new operator **cube by**, introduced by Gray et al.'96)

SELECT item, city, year, SUM (amount)

FROM SALES

**CUBE BY** item, city, year

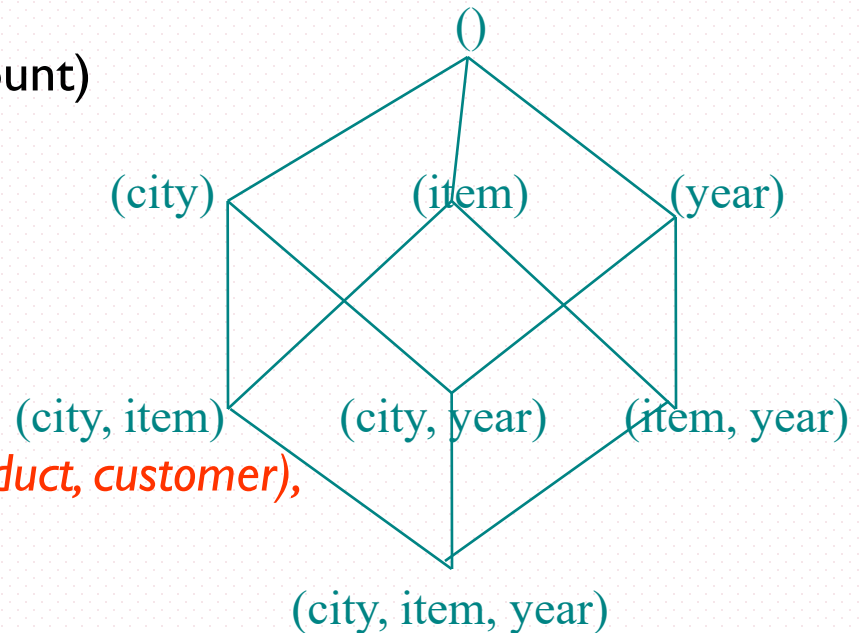
- Need compute the following Group-Bys

(date, product, customer),

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

(date), (product), (customer)

()



## 45 CUBE COMPUTATION: ROLAP-BASED METHOD

- Efficient cube computation methods
  - ROLAP-based cubing algorithms (Agarwal et al'96)
  - Array-based cubing algorithm (Zhao et al'97)
  - Bottom-up computation method (Bayer & Ramakrishnan'99)
- ROLAP-based cubing algorithms
  - Sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples
  - Grouping is performed on some subaggregates as a “partial grouping step”
  - Aggregates may be computed from previously computed aggregates, rather than from the base fact table

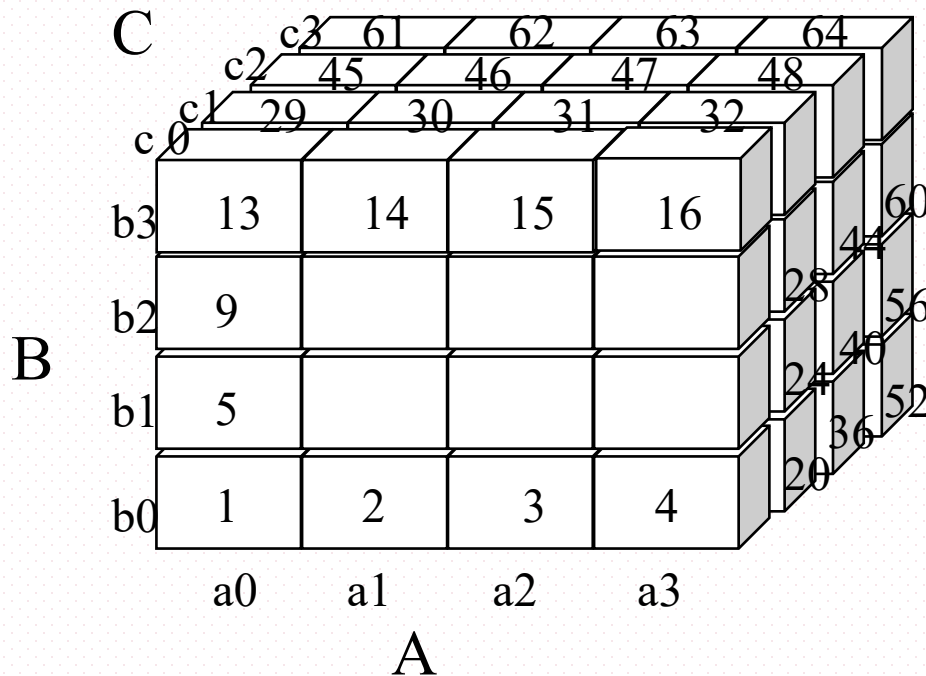
# MULTI-WAY ARRAY AGGREGATION FOR CUBE COMPUTATION

Data Mining: Concepts and Techniques

July 18, 2025

47

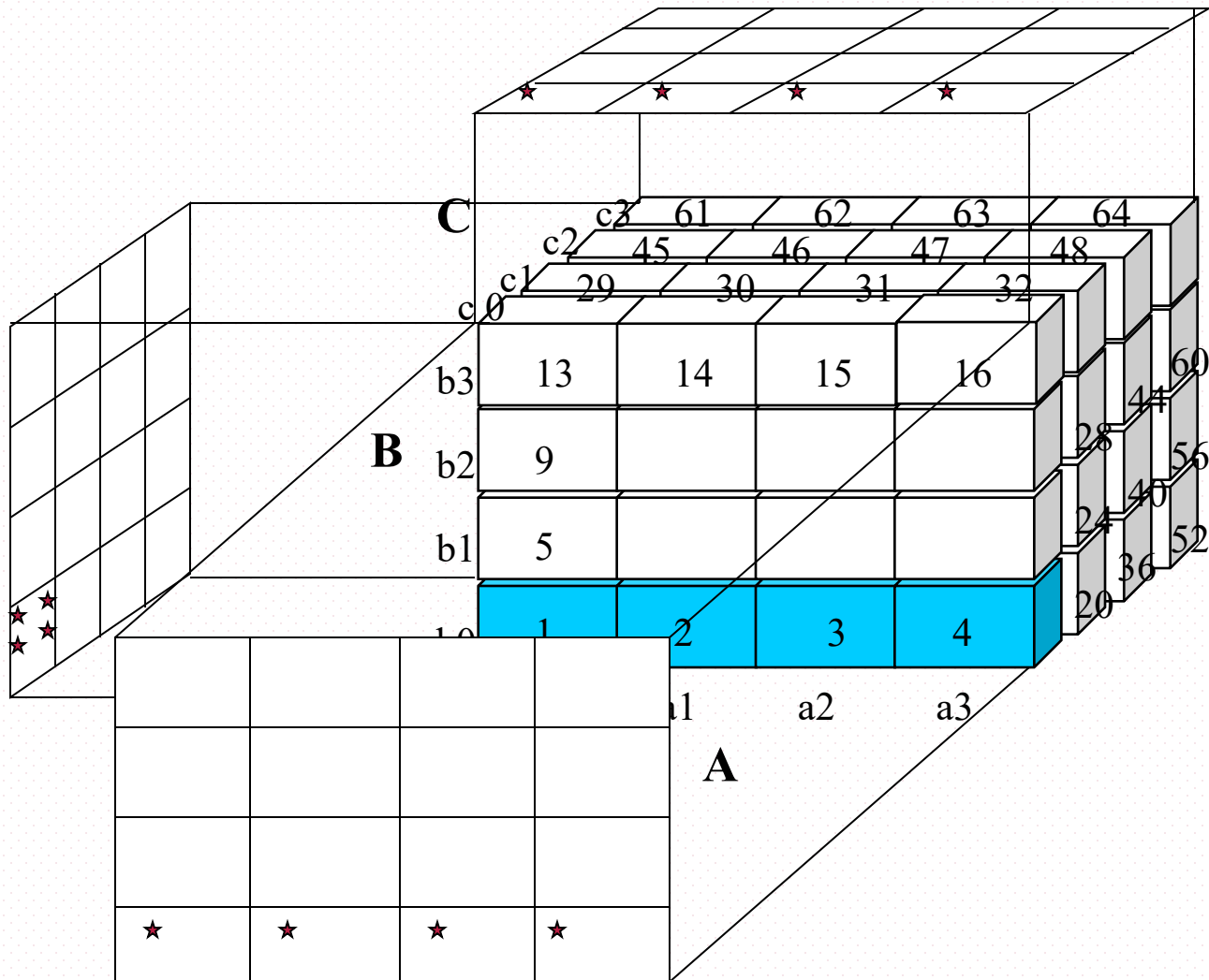
- Partition arrays into chunks (a small subcube which fits in memory).
- Compressed sparse array addressing: (chunk\_id, offset)
- Compute aggregates in “multiway” by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost.



**What is the best  
traversing order  
to do multi-way  
aggregation?**

# MULTI-WAY ARRAY AGGREGATION FOR CUBE COMPUTATION

48

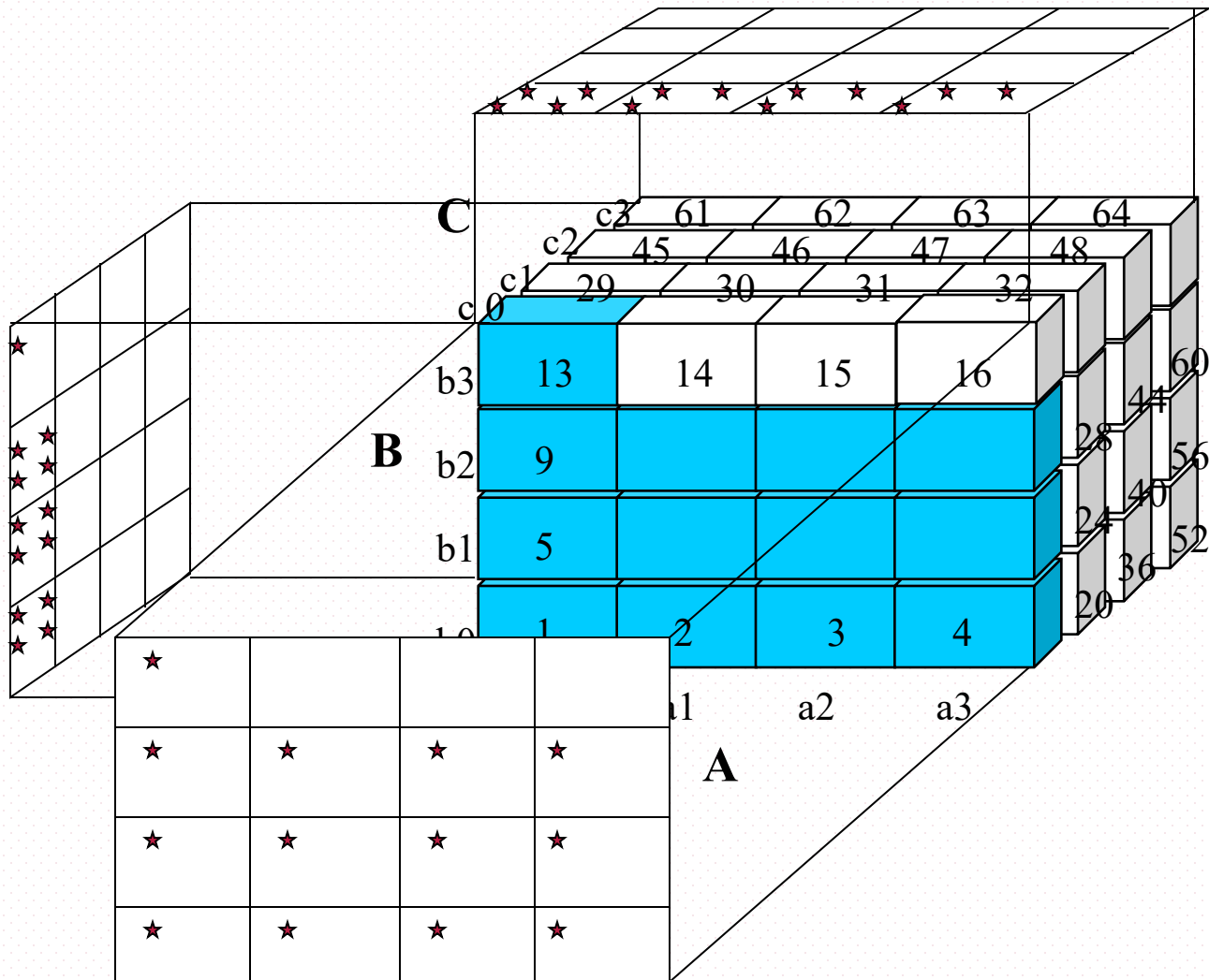


# MULTI-WAY ARRAY AGGREGATION FOR CUBE COMPUTATION

Data Mining: Concepts and Techniques

July 15, 2025

49



## 50

# MULTI-WAY ARRAY AGGREGATION FOR CUBE COMPUTATION (CONT.)

- Method: the planes should be sorted and computed according to their size in ascending order.
  - See the details of Example 2.12 (pp. 75-78)
  - Idea: keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane
- Limitation of the method: computing well only for a small number of dimensions
  - If there are a large number of dimensions, “bottom-up computation” and iceberg cube computation methods can be explored

## 51 INDEXING OLAP DATA: BITMAP INDEX

- Index on a particular column
- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # of records in the base table
- The  $i$ -th bit is set if the  $i$ -th row of the base table has the value for the indexed column
- not suitable for high cardinality domains

**Base table**

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

**Index on Region**

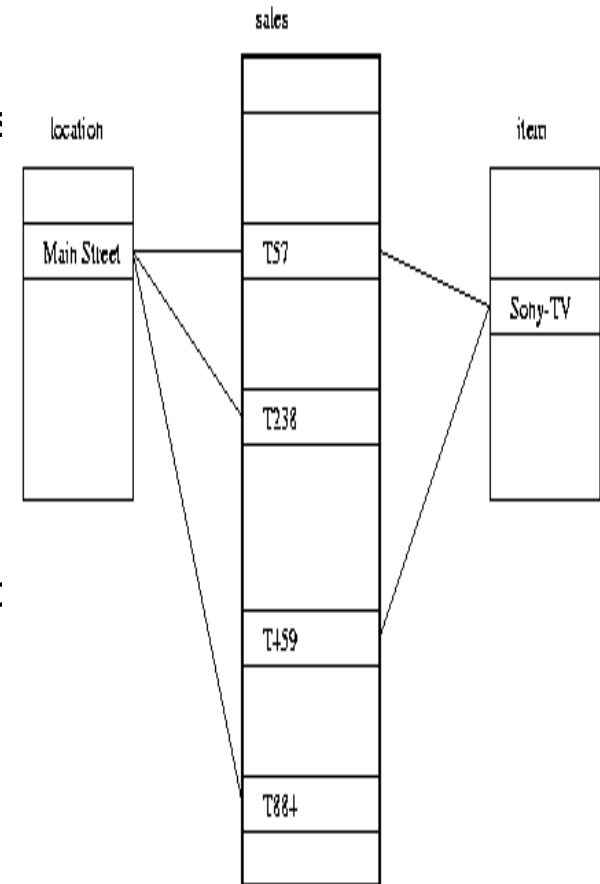
| RecID | Asia | Europe | America |
|-------|------|--------|---------|
| 1     | 1    | 0      | 0       |
| 2     | 0    | 1      | 0       |
| 3     | 1    | 0      | 0       |
| 4     | 0    | 0      | 1       |
| 5     | 0    | 1      | 0       |

**Index on Type**

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

# 52 INDEXING OLAP DATA: JOIN INDICES

- Join index:  $Jl(R\text{-id}, S\text{-id})$  where  $R(R\text{-id}, \dots) \triangleright \triangleleft S(S\text{-id}, \dots)$
- Traditional indices map the values to a list of record ids
  - It materializes relational join in JI file and speeds up relational join — a rather costly operation
- In data warehouses, join index relates the values of the dimensions of a star schema to rows in the fact table.
  - E.g. fact table: *Sales* and two dimensions *city* and *product*
    - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
  - Join indices can span multiple dimensions



## 53 EFFICIENT PROCESSING OLAP QUERIES

- Determine which operations should be performed on the available cuboids:
  - transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g, dice = selection + projection
- Determine to which materialized cuboid(s) the relevant operations should be applied.
- Exploring indexing structures and compressed vs. dense array structures in MOLAP

## 54 METADATA REPOSITORY

- Meta data is the data defining warehouse objects. It has the following kinds
  - Description of the structure of the warehouse
    - schema, view, dimensions, hierarchies, derived data defn, data mart locations and contents
  - Operational meta-data
    - data lineage (history of migrated data and transformation path), currency of data (active, archived, or purged), monitoring information (warehouse usage statistics, error reports, audit trails)
  - The algorithms used for summarization
  - The mapping from operational environment to the data warehouse
  - Data related to system performance
    - warehouse schema, view and derived data definitions
  - Business data
    - business terms and definitions, ownership of data, charging policies

## 55 DATA WAREHOUSE BACK-END TOOLS AND UTILITIES

- Data extraction:
  - get data from multiple, heterogeneous, and external sources
- Data cleaning:
  - detect errors in the data and rectify them when possible
- Data transformation:
  - convert data from legacy or host format to warehouse format
- Load:
  - sort, summarize, consolidate, compute views, check integrity, and build indices and partitions
- Refresh
  - propagate the updates from the data sources to the warehouse

# CHAPTER 2: DATA WAREHOUSING AND OLAP TECHNOLOGY FOR DATA MINING

## 56

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

## 57

# DISCOVERY-DRIVEN EXPLORATION OF DATA CUBES

- Hypothesis-driven: exploration by user, huge search space
- Discovery-driven (Sarawagi et al.'98)
  - pre-compute measures indicating exceptions, guide user in the data analysis, at all levels of aggregation
  - Exception: significantly different from the value anticipated, based on a statistical model
  - Visual cues such as background color are used to reflect the degree of exception of each cell
  - Computation of exception indicator (modeling fitting and computing SelfExp, InExp, and PathExp values) can be overlapped with cube construction

# EXAMPLES: DISCOVERY-DRIVEN DATA

## 58 CUBES

|        |     |
|--------|-----|
| item   | all |
| region | all |

| Sum of sales | month |     |     |     |     |     |     |     |     |     |     |     |
|--------------|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|              | Jan   | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| Total        |       | 1%  | -1% | 0%  | 1%  | 3%  | -1  | -9% | -1% | 2%  | -4% | 3%  |

| Avg sales               | month |     |     |     |     |     |      |      |     |      |      |      |
|-------------------------|-------|-----|-----|-----|-----|-----|------|------|-----|------|------|------|
| item                    | Jan   | Feb | Mar | Apr | May | Jun | Jul  | Aug  | Sep | Oct  | Nov  | Dec  |
| Sony b/w printer        |       | 9%  | -8% | 2%  | -5% | 14% | -4%  | 0%   | 41% | -13% | -15% | -11% |
| Sony color printer      |       | 0%  | 0%  | 3%  | 2%  | 4%  | -10% | -13% | 0%  | 4%   | -6%  | 4%   |
| HP b/w printer          |       | -2% | 1%  | 2%  | 3%  | 8%  | 0%   | -12% | -9% | 3%   | -3%  | 6%   |
| HP color printer        |       | 0%  | 0%  | -2% | 1%  | 0%  | -1%  | -7%  | -2% | 1%   | -5%  | 1%   |
| IBM home computer       |       | 1%  | -2% | -1% | -1% | 3%  | 3%   | -10% | 4%  | 1%   | -4%  | -1%  |
| IBM laptop computer     |       | 0%  | 0%  | -1% | 3%  | 4%  | 2%   | -10% | -2% | 0%   | -9%  | 3%   |
| Toshiba home computer   |       | -2% | -5% | 1%  | 1%  | -1% | 1%   | 5%   | -3% | -5%  | -1%  | -1%  |
| Toshiba laptop computer |       | 1%  | 0%  | 3%  | 0%  | -2% | -2%  | -5%  | 3%  | 2%   | -1%  | 0%   |
| Logitech mouse          |       | 3%  | -2% | -1% | 0%  | 4%  | 6%   | -11% | 2%  | 1%   | -4%  | 0%   |
| Ergo-way mouse          |       | 0%  | 0%  | 2%  | 3%  | 1%  | -2%  | -2%  | -5% | 0%   | -5%  | 8%   |

|      |                   |
|------|-------------------|
| item | IBM home computer |
|------|-------------------|

| Avg sales | month |     |     |     |     |     |      |      |      |     |     |     |
|-----------|-------|-----|-----|-----|-----|-----|------|------|------|-----|-----|-----|
| region    | Jan   | Feb | Mar | Apr | May | Jun | Jul  | Aug  | Sep  | Oct | Nov | Dec |
| North     |       | -1% | -3% | -1% | 0%  | 3%  | 4%   | -7%  | 1%   | 0%  | -3% | -3% |
| South     |       | -1% | 1%  | -9% | 6%  | -1% | -39% | 9%   | -34% | 4%  | 1%  | 7%  |
| East      |       | -1% | -2% | 2%  | -3% | 1%  | 18%  | -2%  | 11%  | -3% | -2% | -1% |
| West      |       | 4%  | 0%  | -1% | -3% | 5%  | 1%   | -18% | 8%   | 5%  | -8% | 1%  |

# COMPLEX AGGREGATION AT MULTIPLE GRANULARITIES: MULTI-FEATURE CUBES

Data Mining: Concepts and Techniques

July 15, 2025

59

- Multi-feature cubes (Ross, et al. 1998): Compute complex queries involving multiple dependent aggregates at multiple granularities
- Ex. Grouping by all subsets of {item, region, month}, find the maximum price in 1997 for each group, and the total sales among all maximum price tuples

```
select item, region, month, max(price), sum(R.sales)
```

```
from purchases
```

```
where year = 1997
```

```
cube by item, region, month: R
```

```
such that R.price = max(price)
```

- Continuing the last example, among the max price tuples, find the min and max shelf life, and find the fraction of the total sales due to tuple that have min shelf life within the set of all max price tuples

# CHAPTER 2: DATA WAREHOUSING AND OLAP TECHNOLOGY FOR DATA MINING

60

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- Further development of data cube technology
- From data warehousing to data mining

# 61 DATA WAREHOUSE USAGE

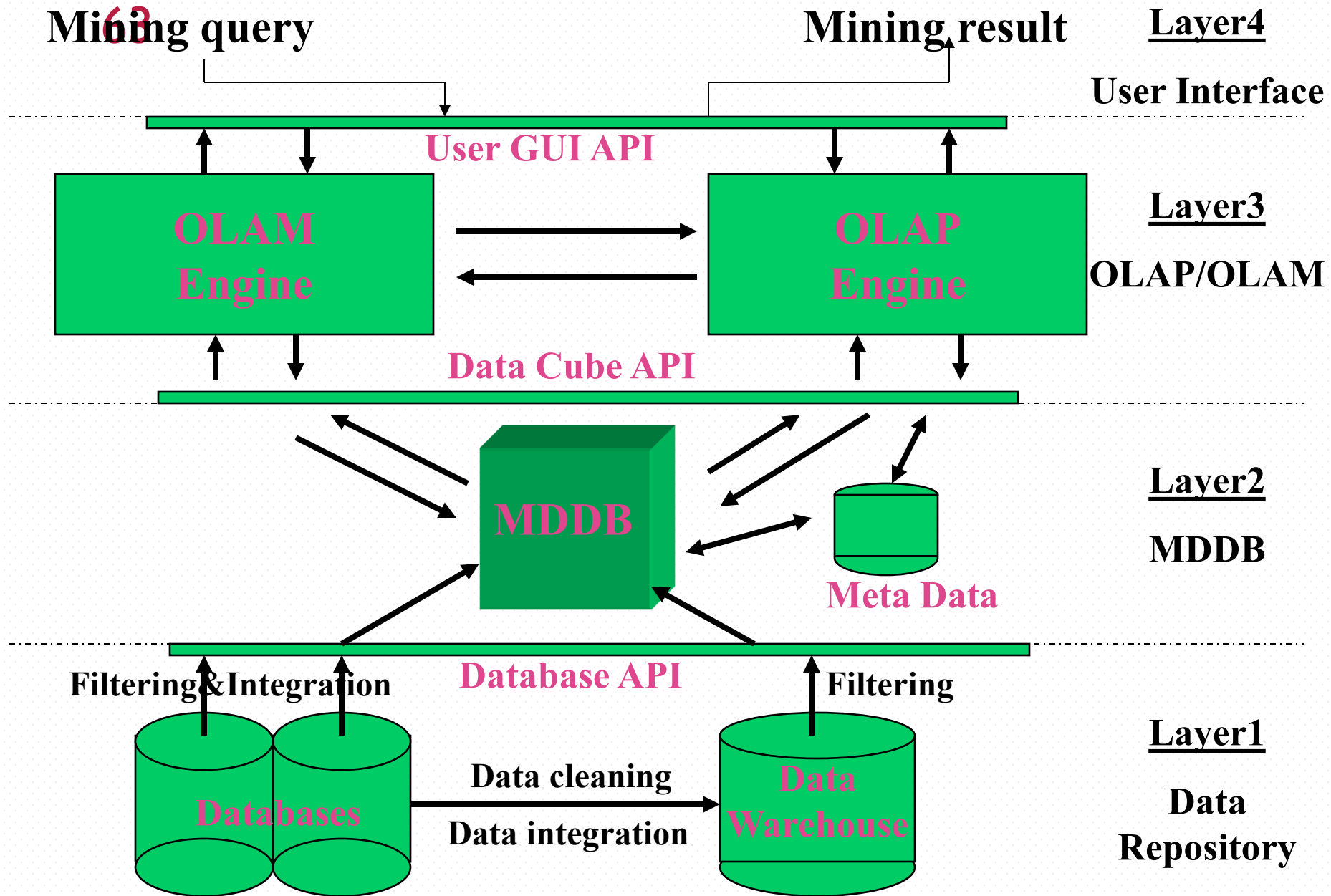
- Three kinds of data warehouse applications
  - **Information processing**
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - **Analytical processing**
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - **Data mining**
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.
- Differences among the three tasks

## 62

# FROM ON-LINE ANALYTICAL PROCESSING TO ON LINE ANALYTICAL MINING (OLAM)

- Why online analytical mining?
  - High quality of data in data warehouses
    - DW contains integrated, consistent, cleaned data
  - Available information processing structure surrounding data warehouses
    - ODBC, OLEDB, Web accessing, service facilities, reporting and OLAP tools
  - OLAP-based exploratory data analysis
    - mining with drilling, dicing, pivoting, etc.
  - On-line selection of data mining functions
    - integration and swapping of multiple mining functions, algorithms, and tasks.
- Architecture of OLAM

# AN OLAM ARCHITECTURE



## 64 SUMMARY

- **Data warehouse**
  - A subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process
- A **multi-dimensional model** of a data warehouse
  - Star schema, snowflake schema, fact constellations
  - A data cube consists of dimensions & measures
- **OLAP** operations: drilling, rolling, slicing, dicing and pivoting
- OLAP servers: ROLAP, MOLAP, HOLAP
- Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Multiway array aggregation
  - Bitmap index and join index implementations
- Further development of data cube technology
  - Discovery-drive and multi-feature cubes
  - From OLAP to OLAM (on-line analytical mining)

## 65 REFERENCES (I)

- S. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In Proc. 1996 Int. Conf. Very Large Data Bases, 506-521, Bombay, India, Sept. 1996.
- D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek. Efficient view maintenance in data warehouses. In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data, 417-427, Tucson, Arizona, May 1997.
- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data, 94-105, Seattle, Washington, June 1998.
- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In Proc. 1997 Int. Conf. Data Engineering, 232-243, Birmingham, England, April 1997.
- K. Beyer and R. Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg CUBEs. In Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99), 359-370, Philadelphia, PA, June 1999.
- S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997.
- OLAP council. MDAPI specification version 2.0. In <http://www.olapcouncil.org/research/apily.htm>, 1998.
- J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Mining and Knowledge Discovery, 1:29-54, 1997.

## 66 REFERENCES (II)

- V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. In Proc. 1996 ACM-SIGMOD Int. Conf. Management of Data, pages 205-216, Montreal, Canada, June 1996.
- Microsoft. OLEDB for OLAP programmer's reference version 1.0. In <http://www.microsoft.com/data/oledb/olap>, 1998.
- K. Ross and D. Srivastava. Fast computation of sparse datacubes. In Proc. 1997 Int. Conf. Very Large Data Bases, 116-125, Athens, Greece, Aug. 1997.
- K. A. Ross, D. Srivastava, and D. Chatziantoniou. Complex aggregation at multiple granularities. In Proc. Int. Conf. of Extending Database Technology (EDBT'98), 263-277, Valencia, Spain, March 1998.
- S. Sarawagi, R. Agrawal, and N. Megiddo. Discovery-driven exploration of OLAP data cubes. In Proc. Int. Conf. of Extending Database Technology (EDBT'98), pages 168-182, Valencia, Spain, March 1998.
- E. Thomsen. OLAP Solutions: Building Multidimensional Information Systems. John Wiley & Sons, 1997.
- Y. Zhao, P. M. Deshpande, and J. F. Naughton. An array-based algorithm for simultaneous multidimensional aggregates. In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data, 159-170, Tucson, Arizona, May 1997.

[HTTP://WWW.CS.SFU.CA/~HAN](http://www.cs.sfu.ca/~han)

67



Thank you !!!