Knowledge Discovery in Databases

Data Warehousing and OLAP Technology

School of Software, Nanjing University

Data Warehousing and OLAP Technology

- 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

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—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.

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.

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".

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

- **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

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

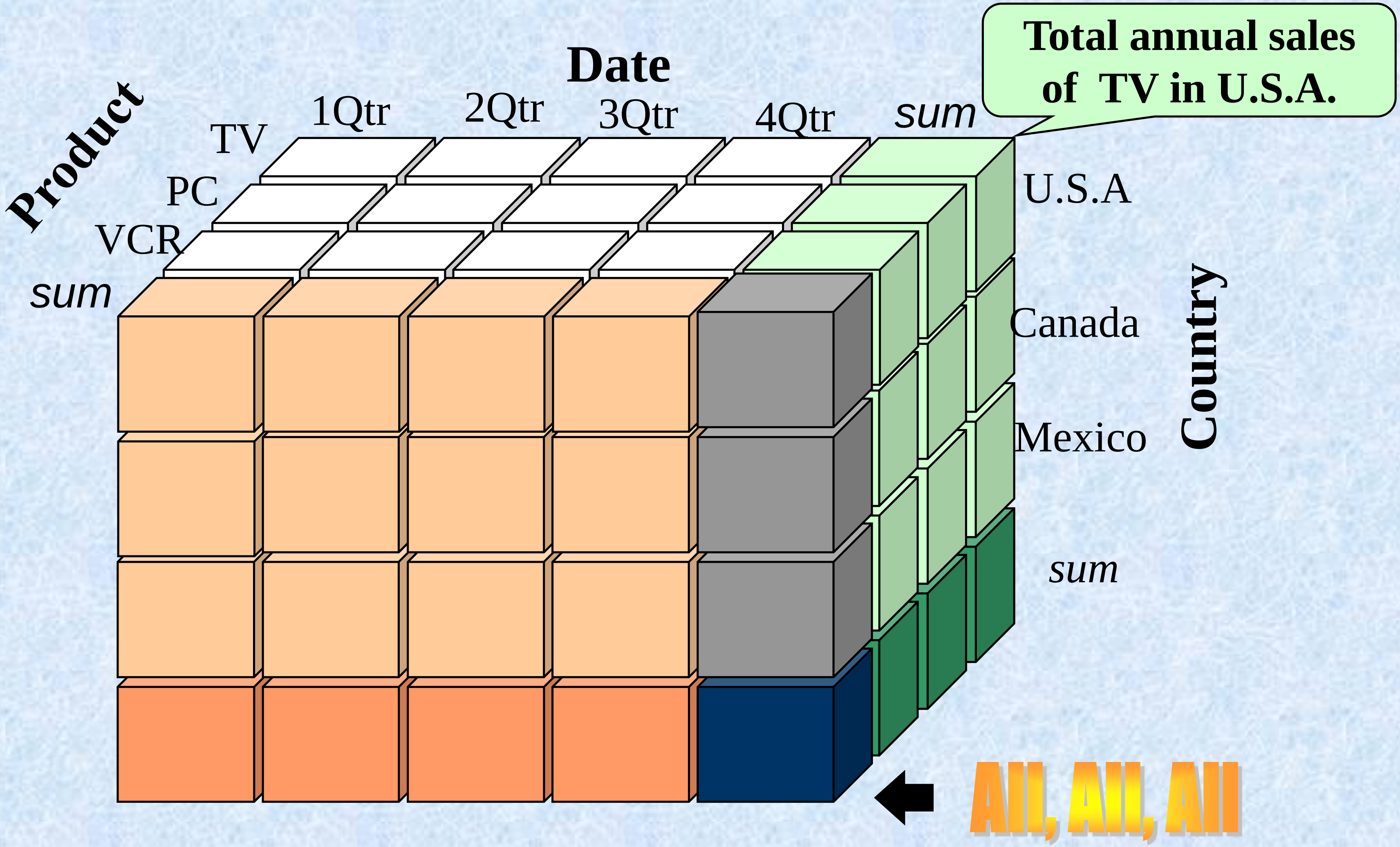
Data Warehousing and OLAP Technology

- 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

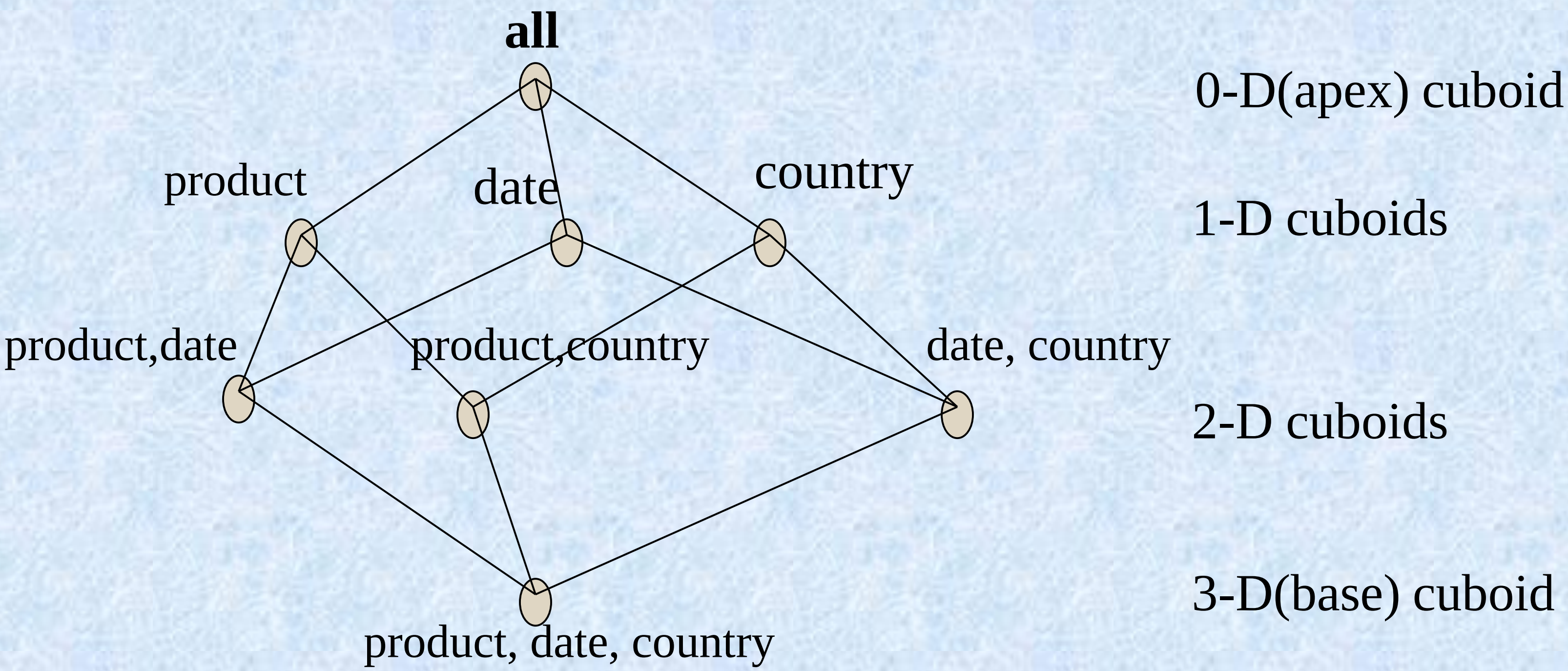
From Tables and Spreadsheets to Data Cubes

- **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

Multi-Dimensional Data Model(Cube)

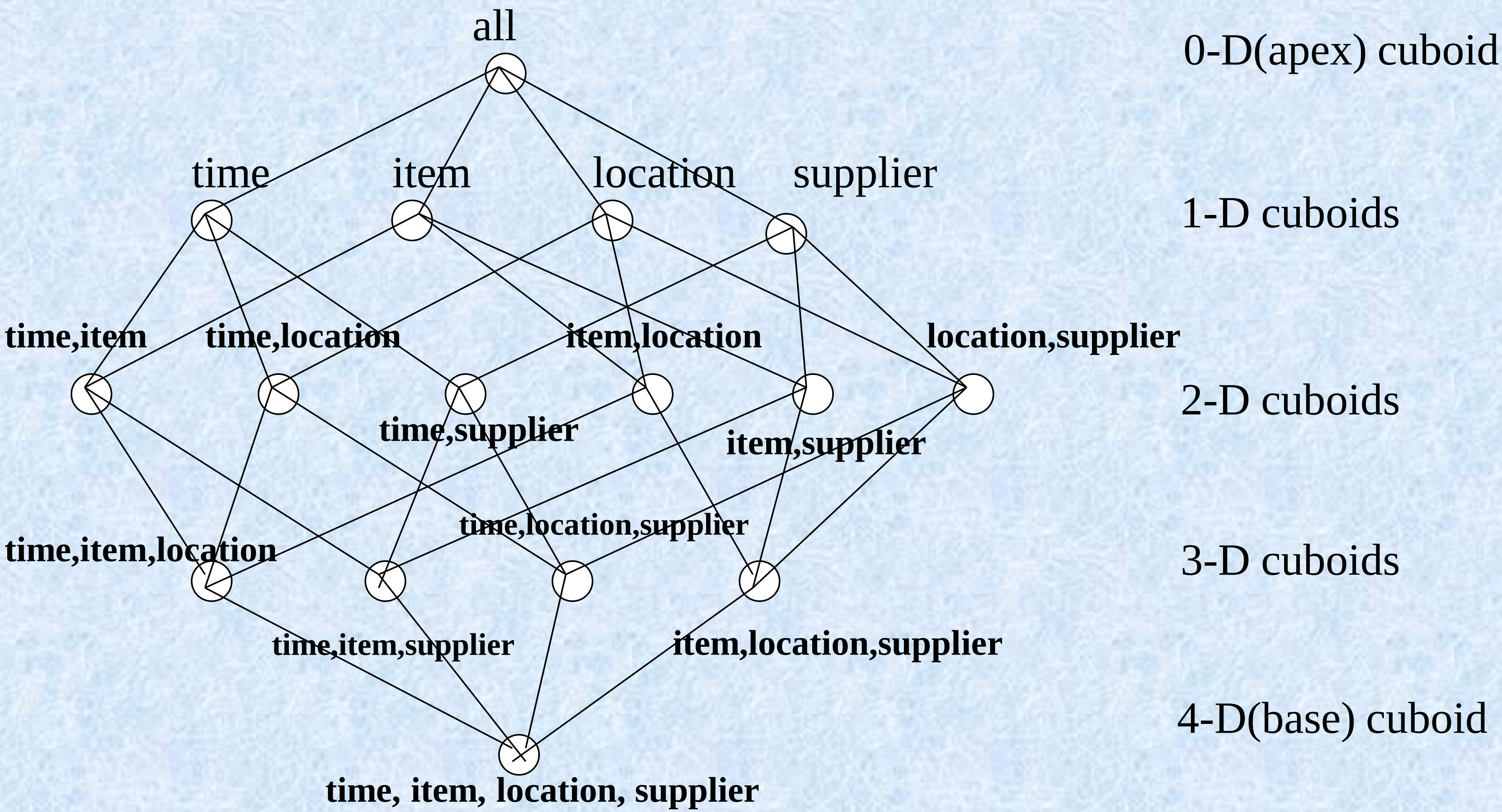


Cube: A Lattice of Cuboids(I)



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.

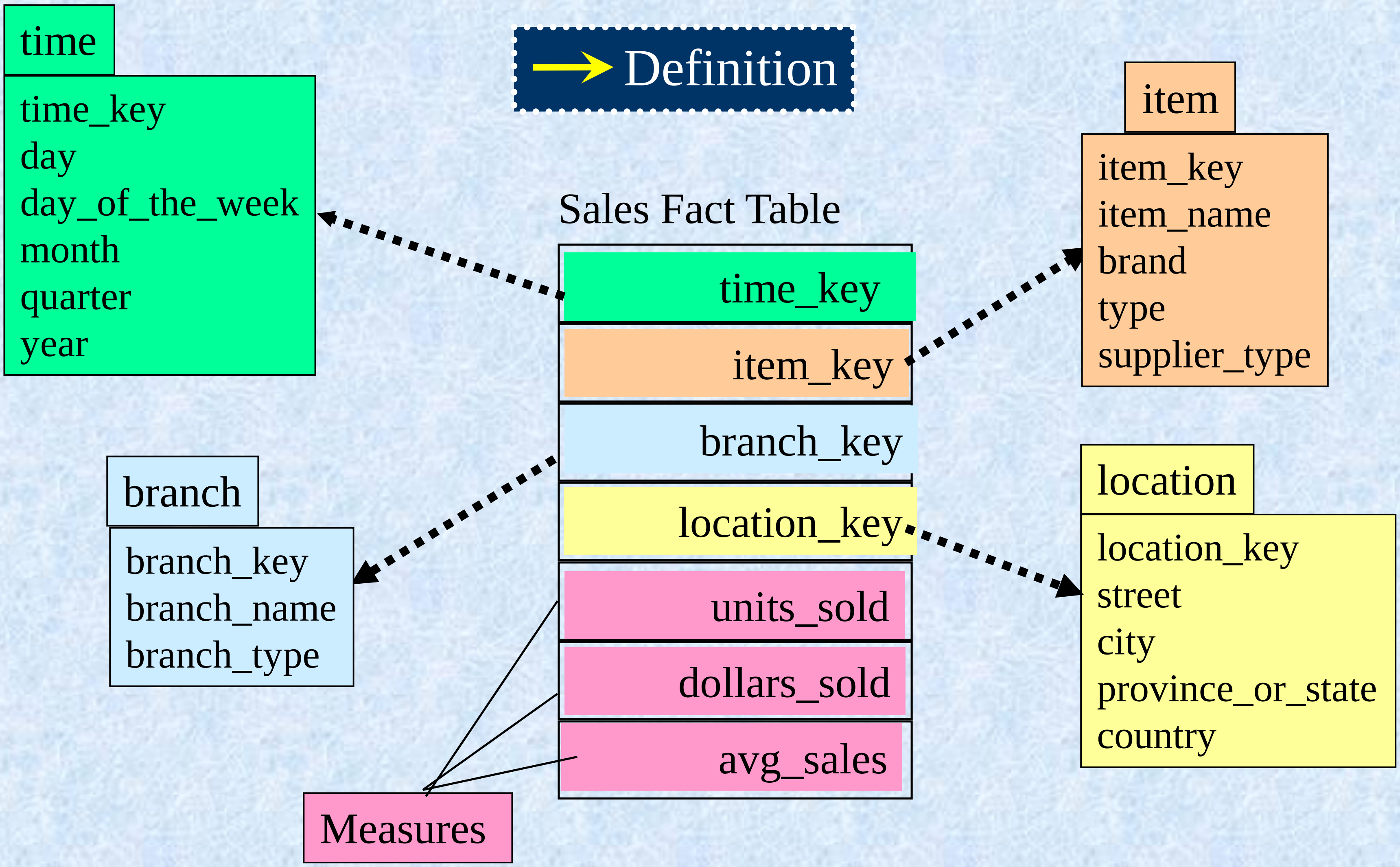
Cube: A Lattice of Cuboids(II)



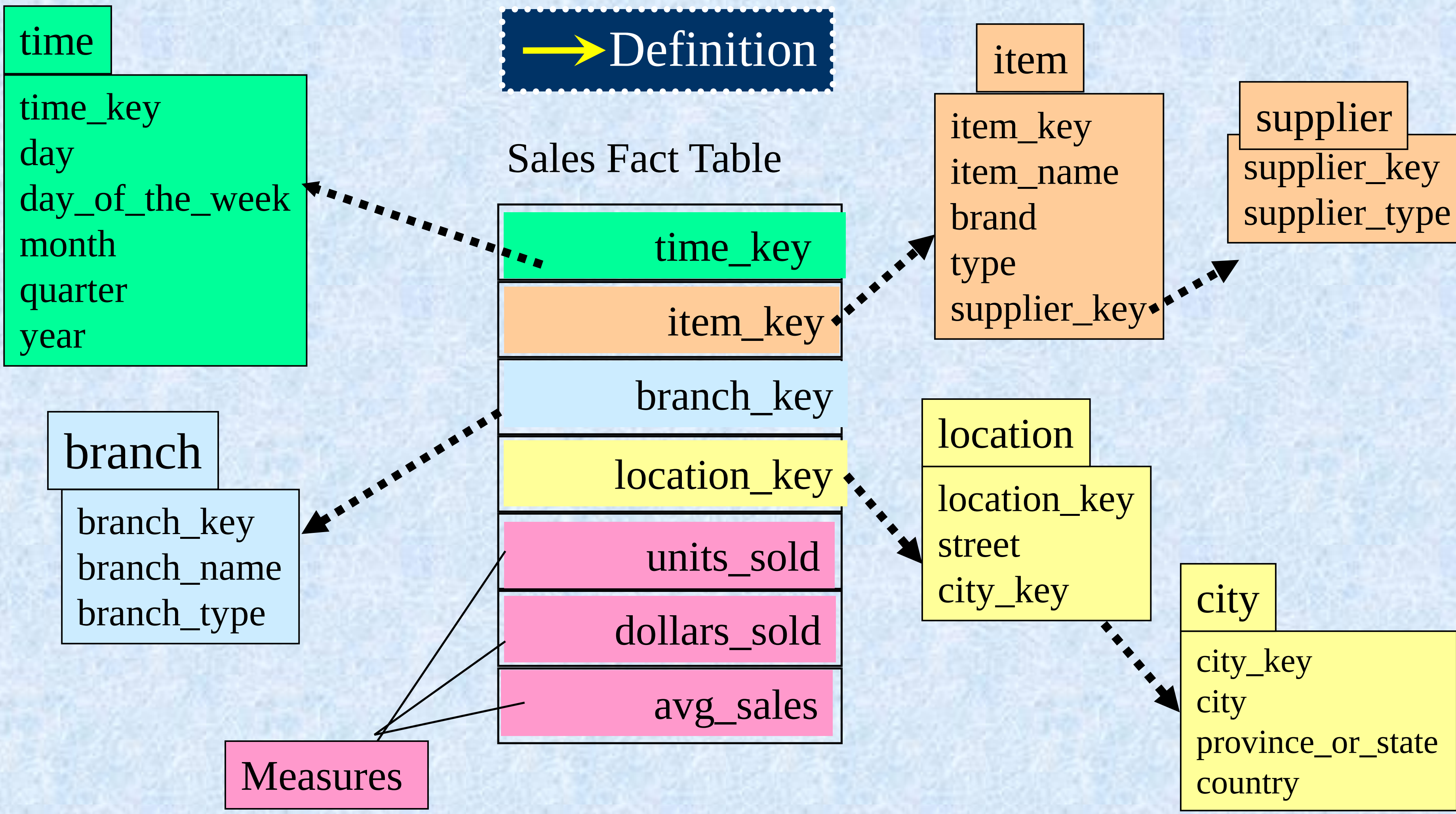
Conceptual Modeling of Data Warehouses

- **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

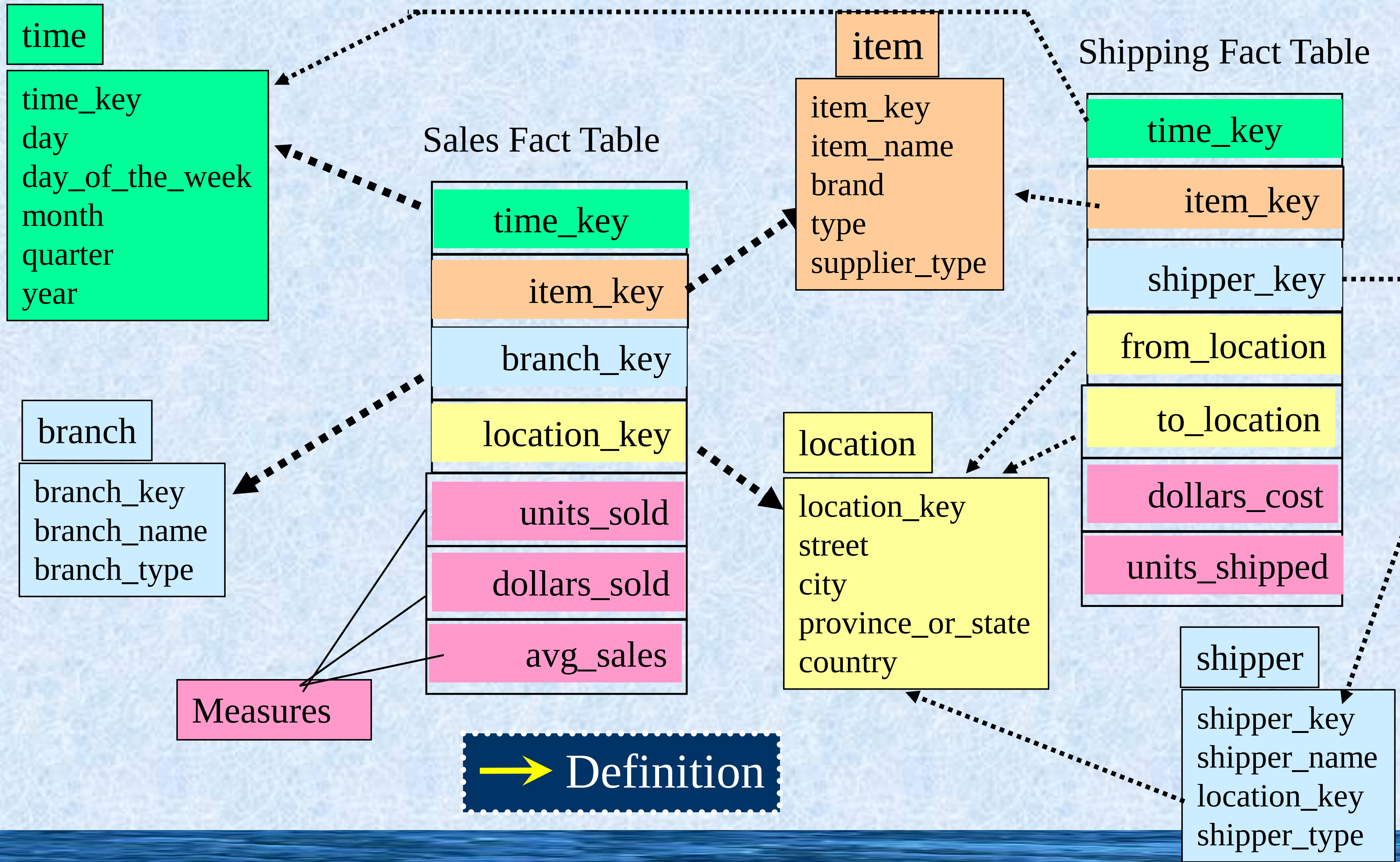
Example of Star Schema



Example of Snowflake Schema



Example of Fact Constellation



→ Definition

A Data Mining Query Language, DMQL

- **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>

Defining a Star Schema in DMQL

define cube sales_star [time, item, branch, location]:

→ Figure

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)

Defining a Snowflake Schema in DMQL

→ Figure

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

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

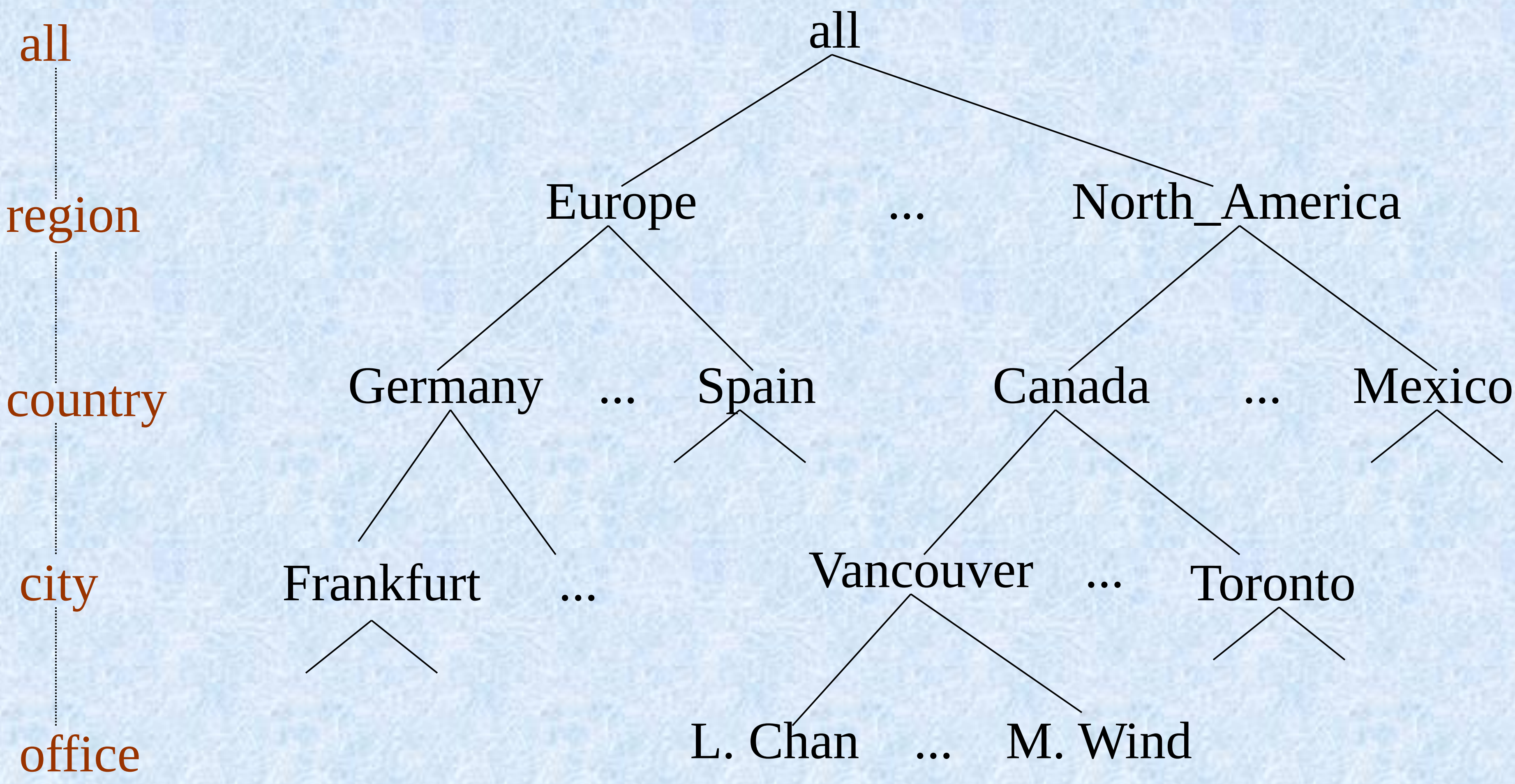
define dimension to_location **as** location **in cube** sales

→ Figure

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(), standard_deviation().
- **holistic: if there is no constant bound on the storage size needed to describe a subaggregate.**
  - ✓ E.g., median(), mode(), rank().

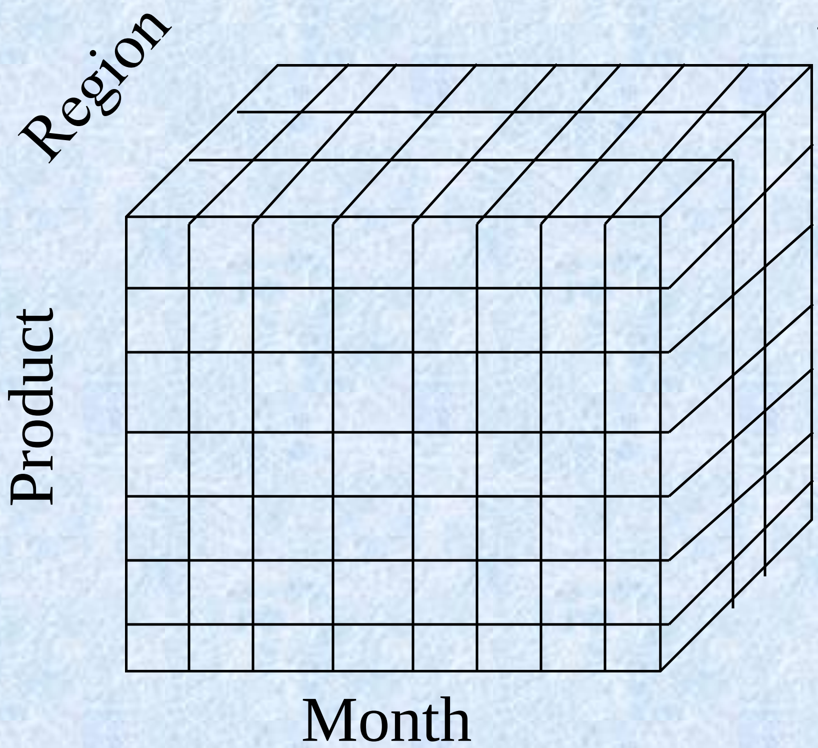
A Concept Hierarchy: Dimension (location)

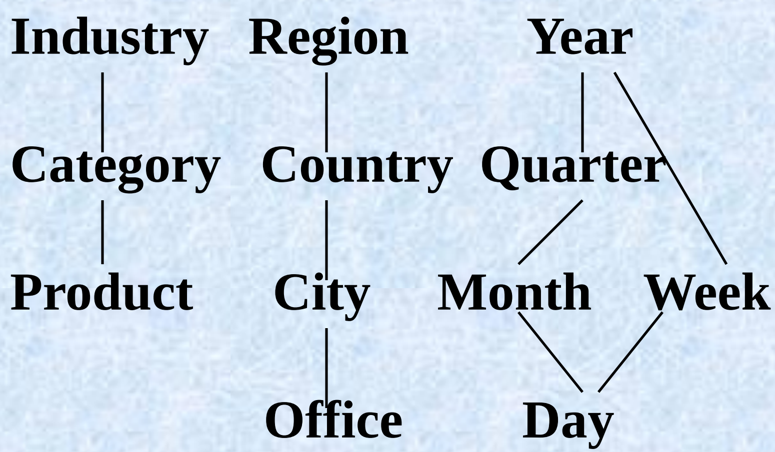


Multidimensional Data

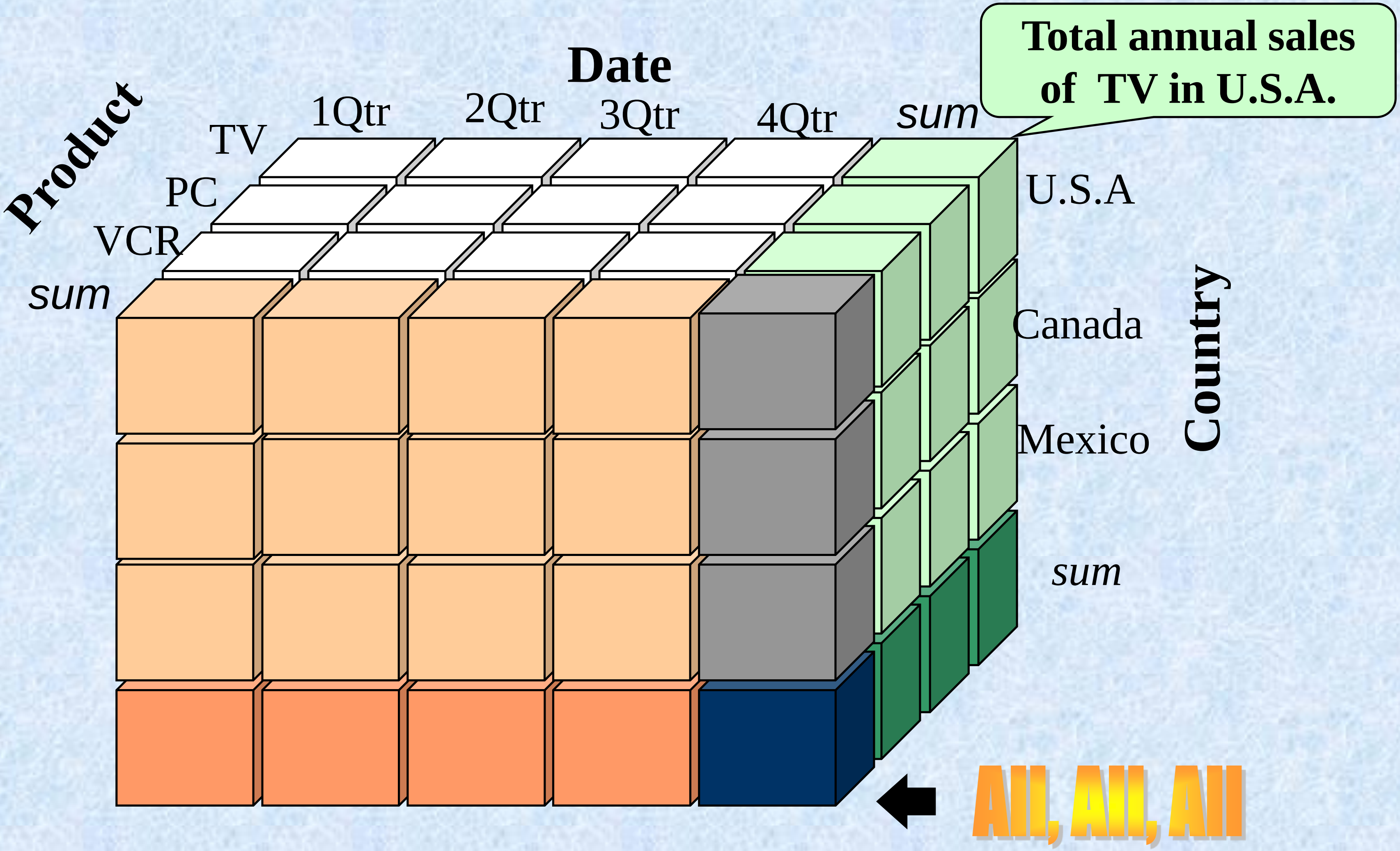
- **Sales volume as a function of product, month, and region**

Dimensions: Product, Location, Time
Hierarchical summarization paths

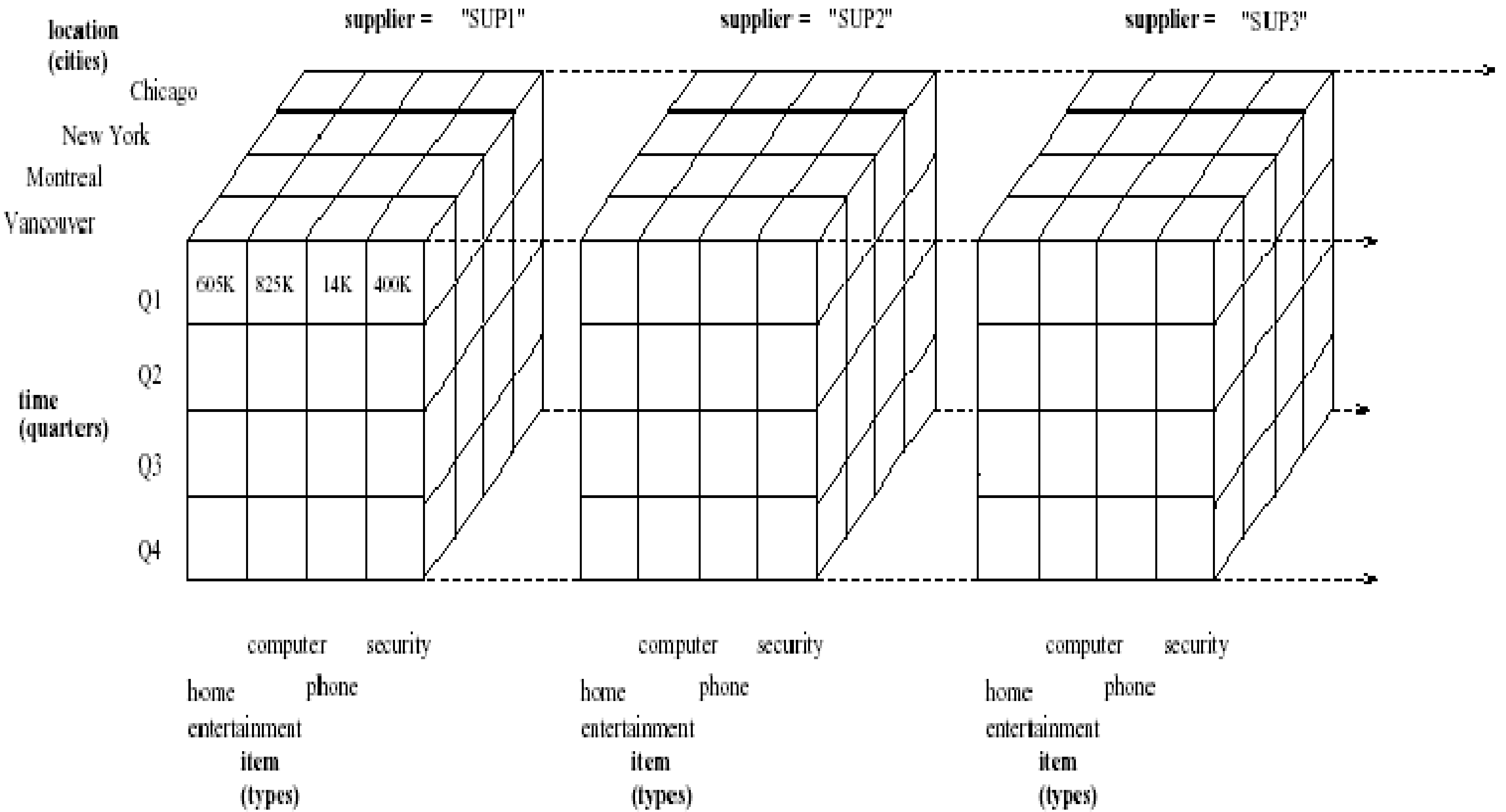




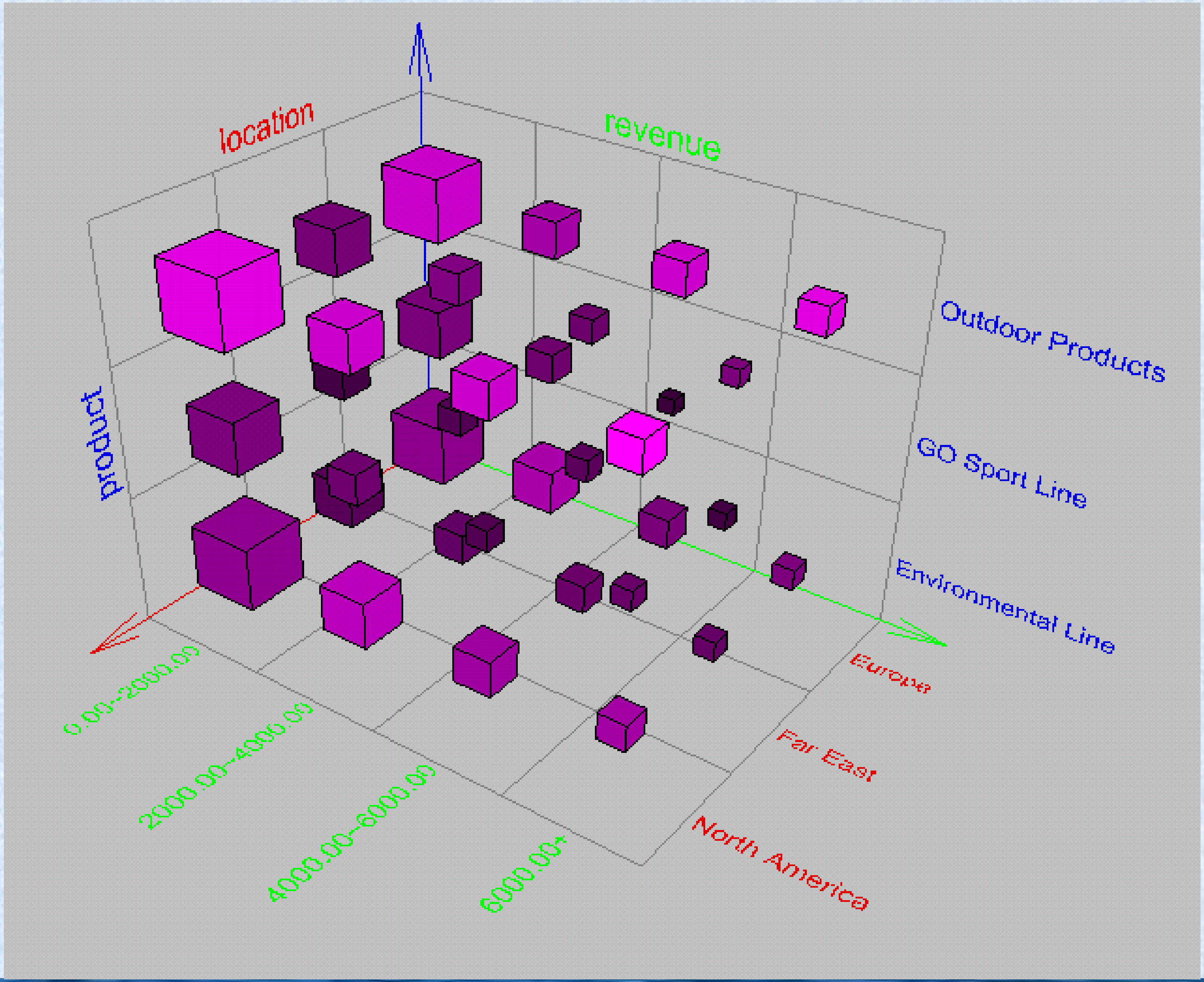
A Sample Data Cube(I)



A Sample Data Cube(II)



Browsing a Data Cube

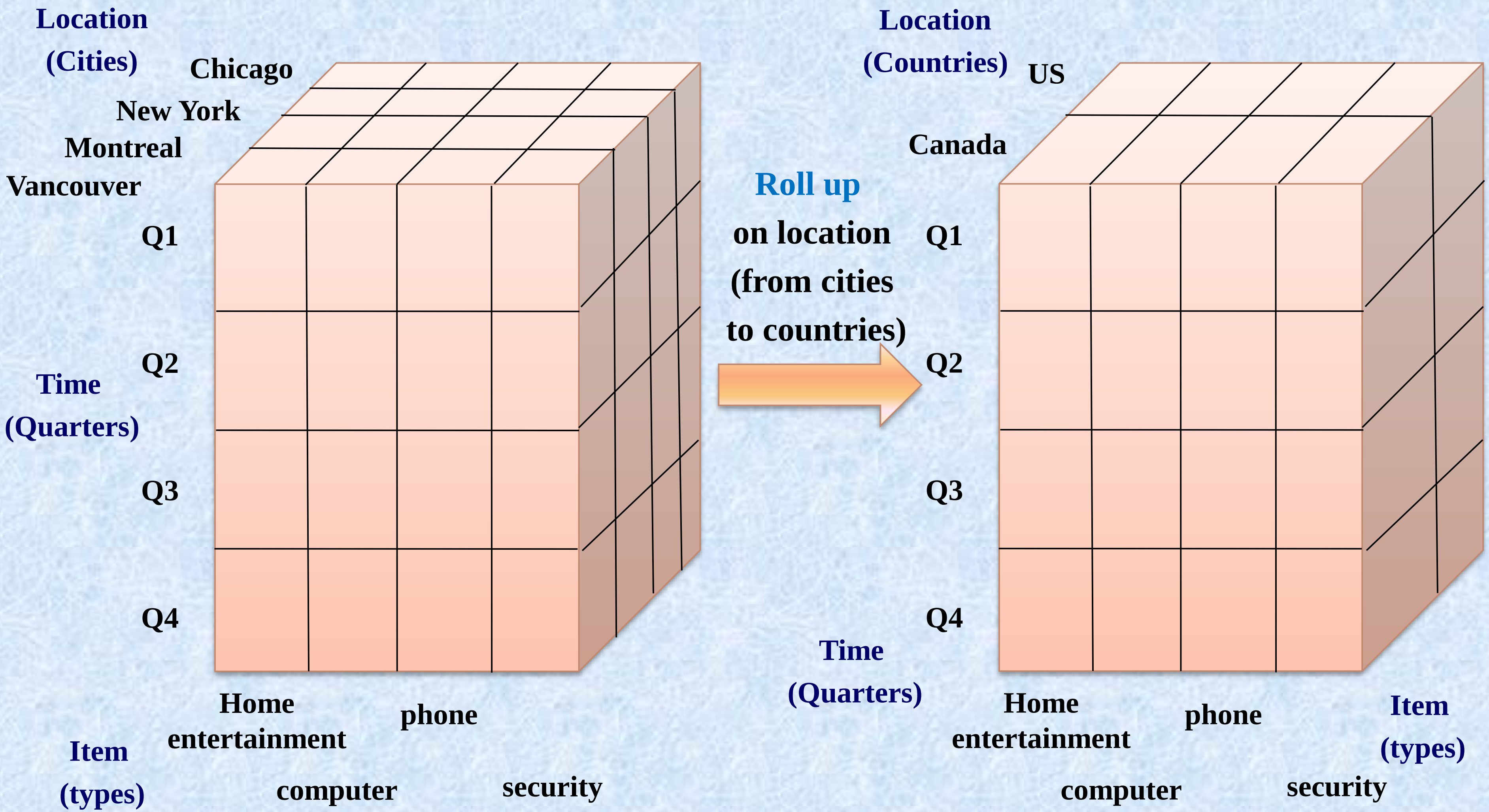


- Visualization
- OLAP capabilities
- Interactive manipulation

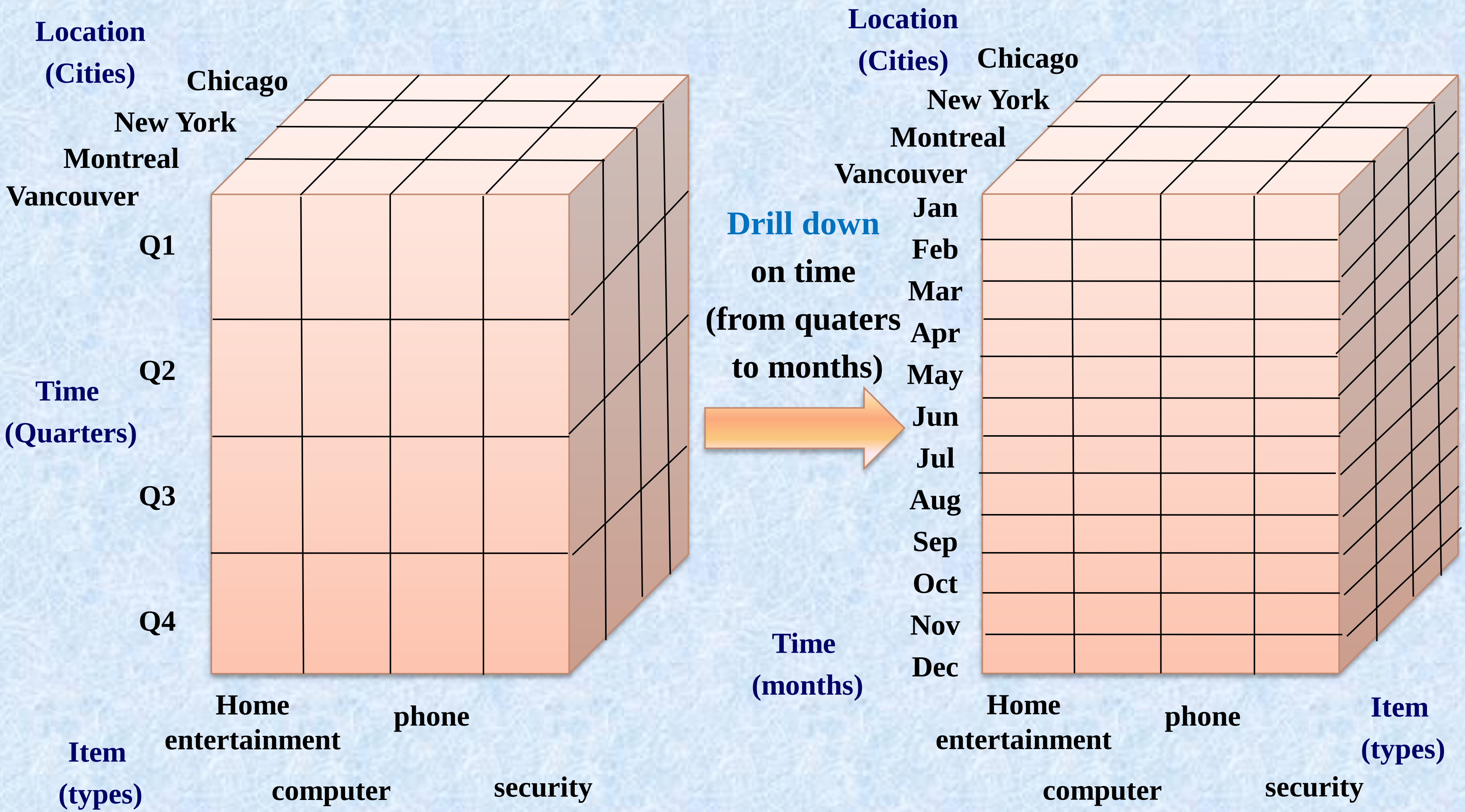
Typical OLAP Operations

- **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)*

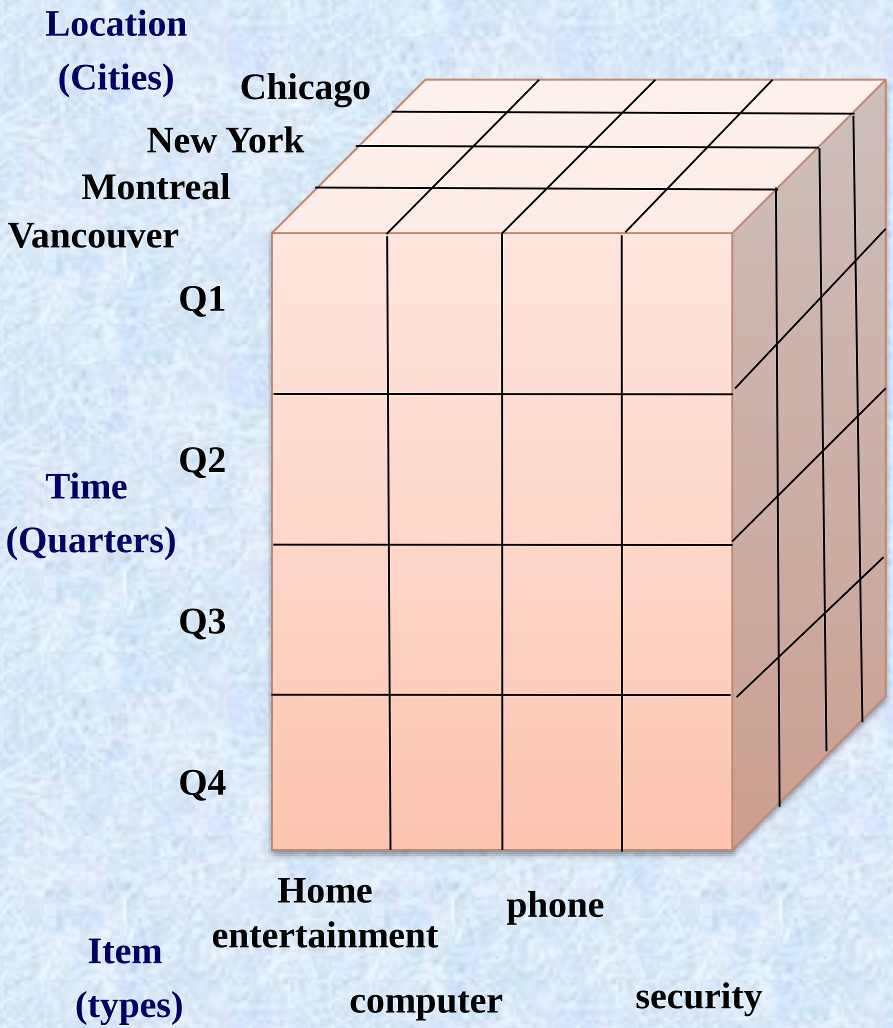
Typical OLAP Operations—— Roll-up



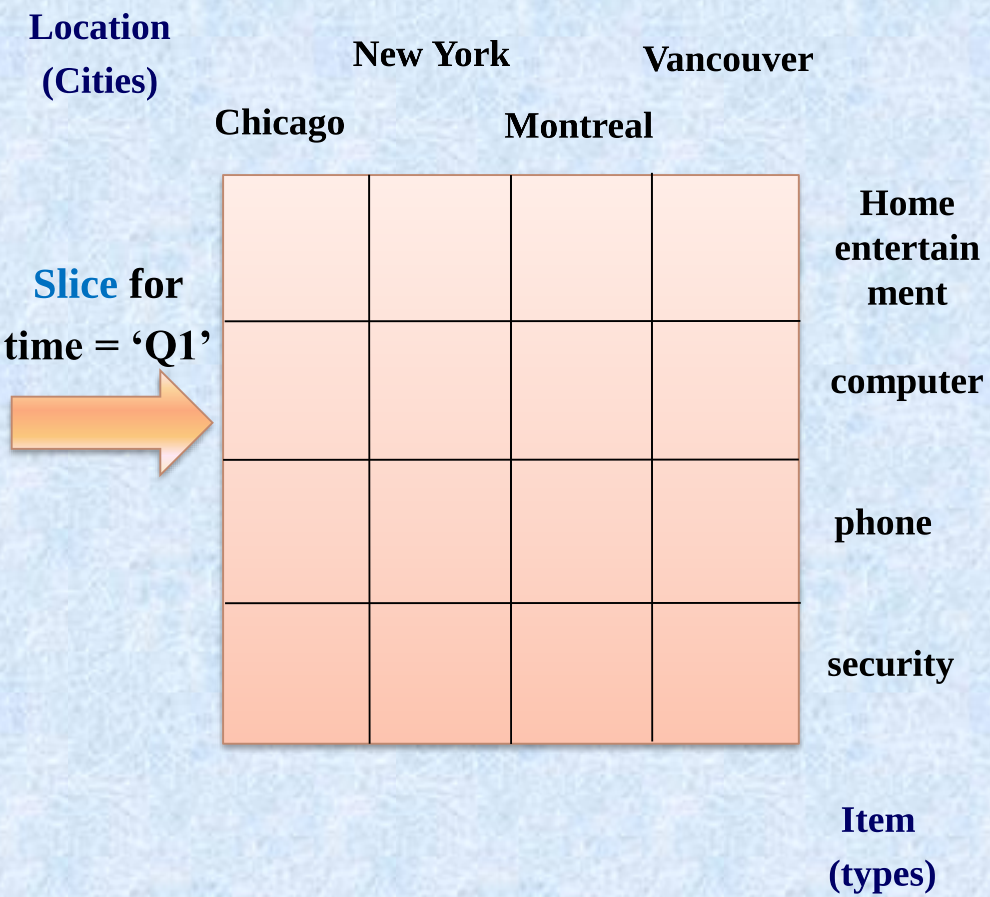
Typical OLAP Operations—— Drill-down



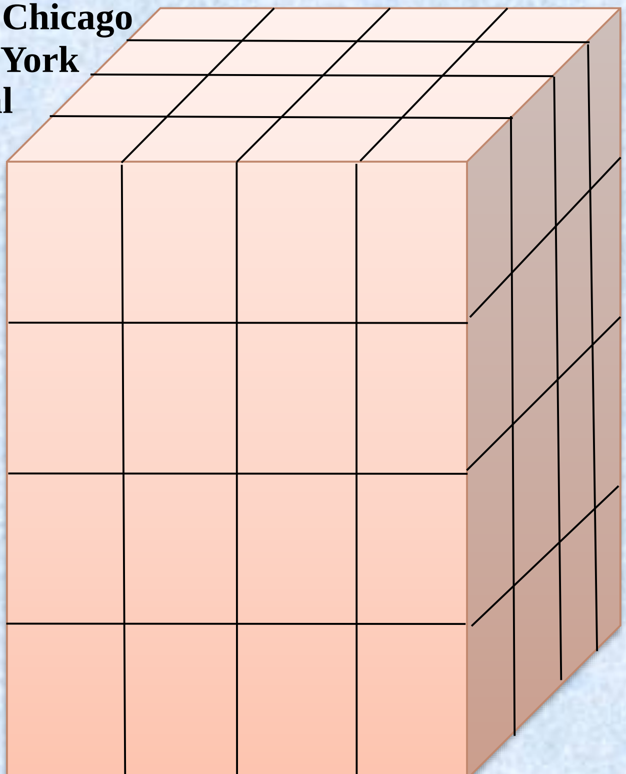
Typical OLAP Operations—— Slice



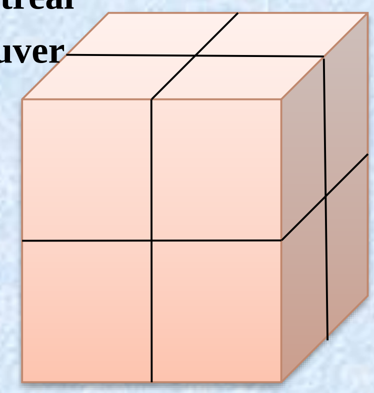
Slice for time = 'Q1'



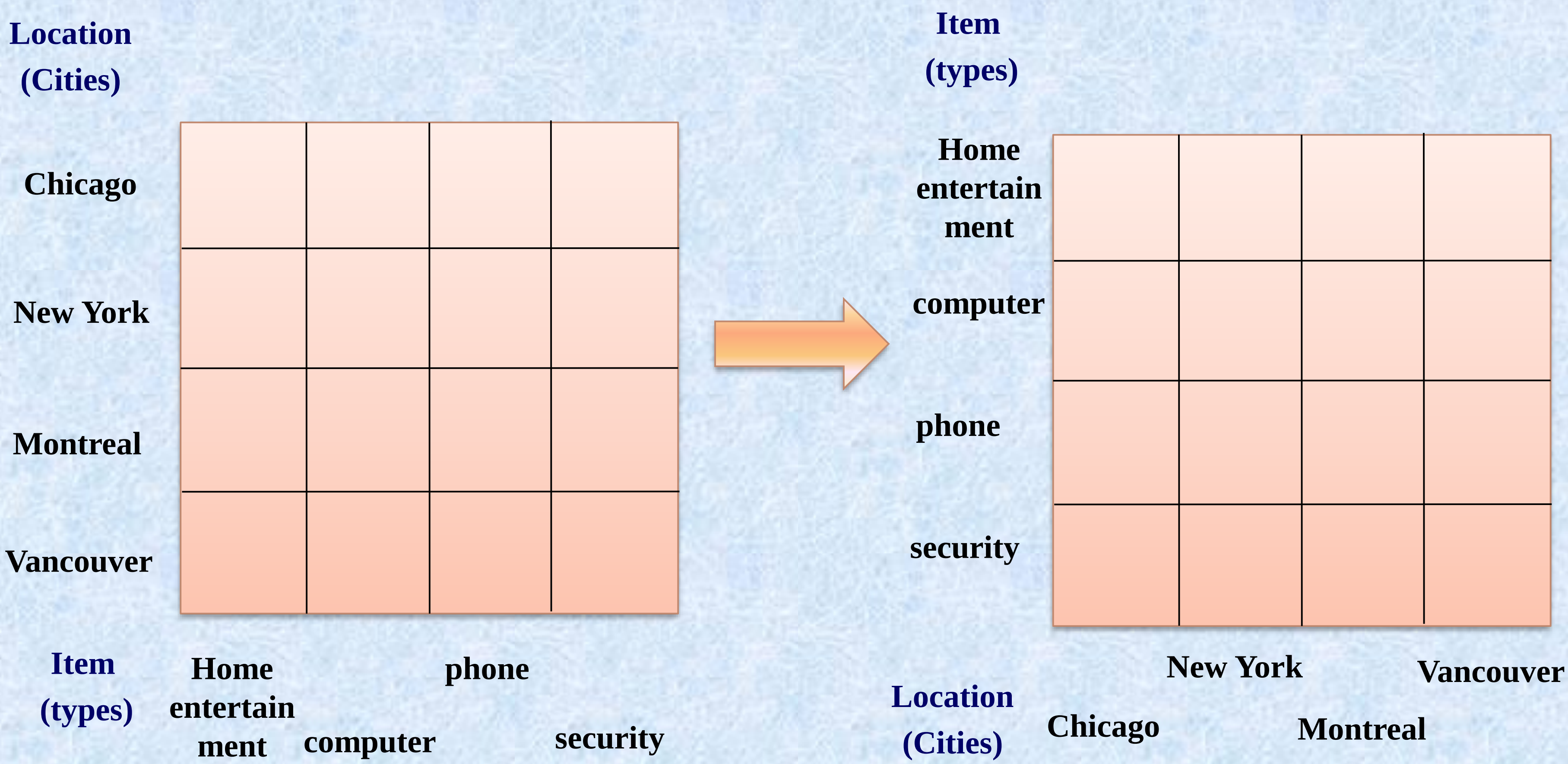
Typical OLAP Operations —— Dice



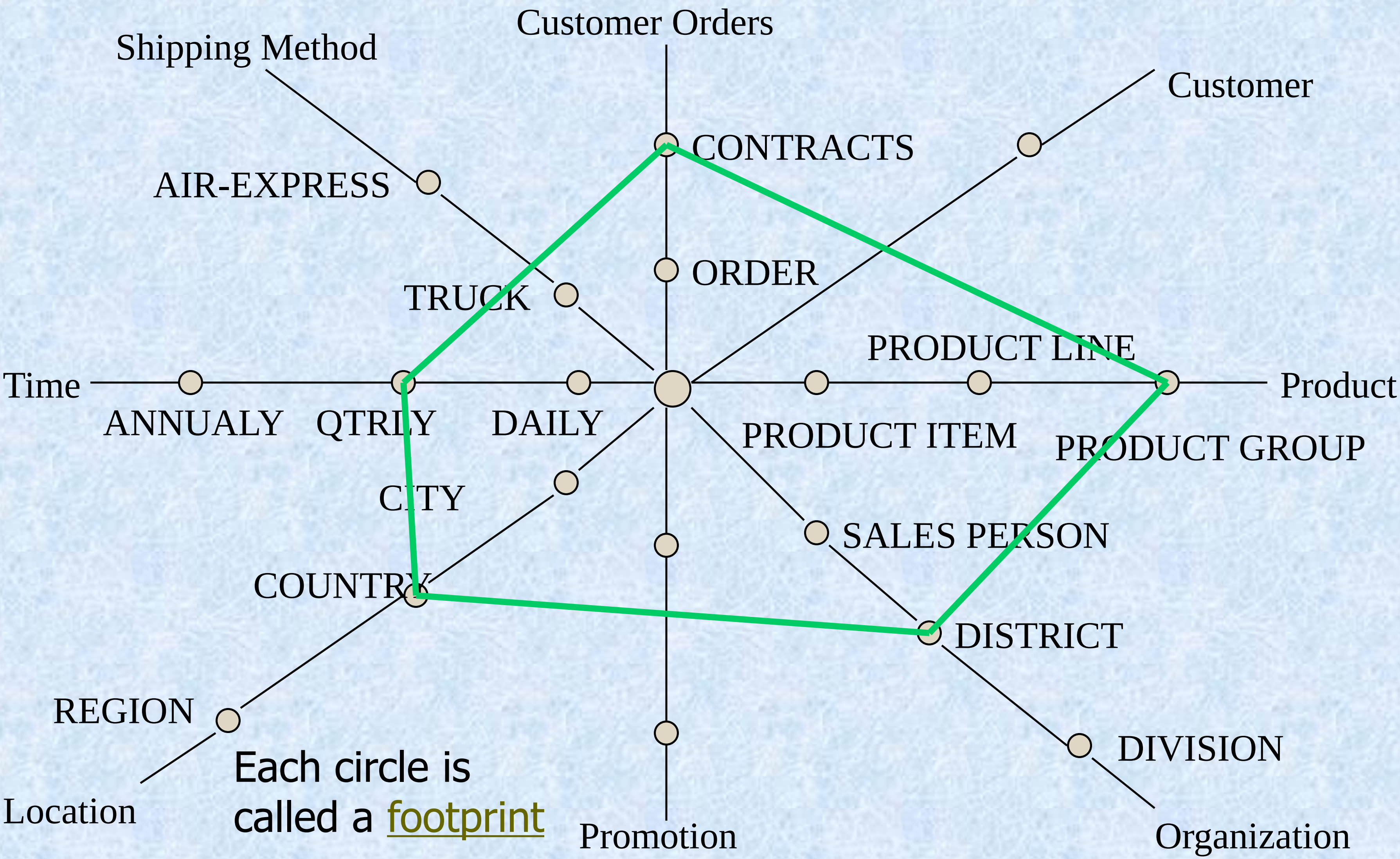
Dice for
(location = ‘Montreal’ or ‘Vancouver’)
and
(time=‘Q1’ or ‘Q2’)
and
(item=‘home entertainment’ or ‘computer’)



Typical OLAP Operations —— Pivot(Rotate)



A Star-Net Query Model

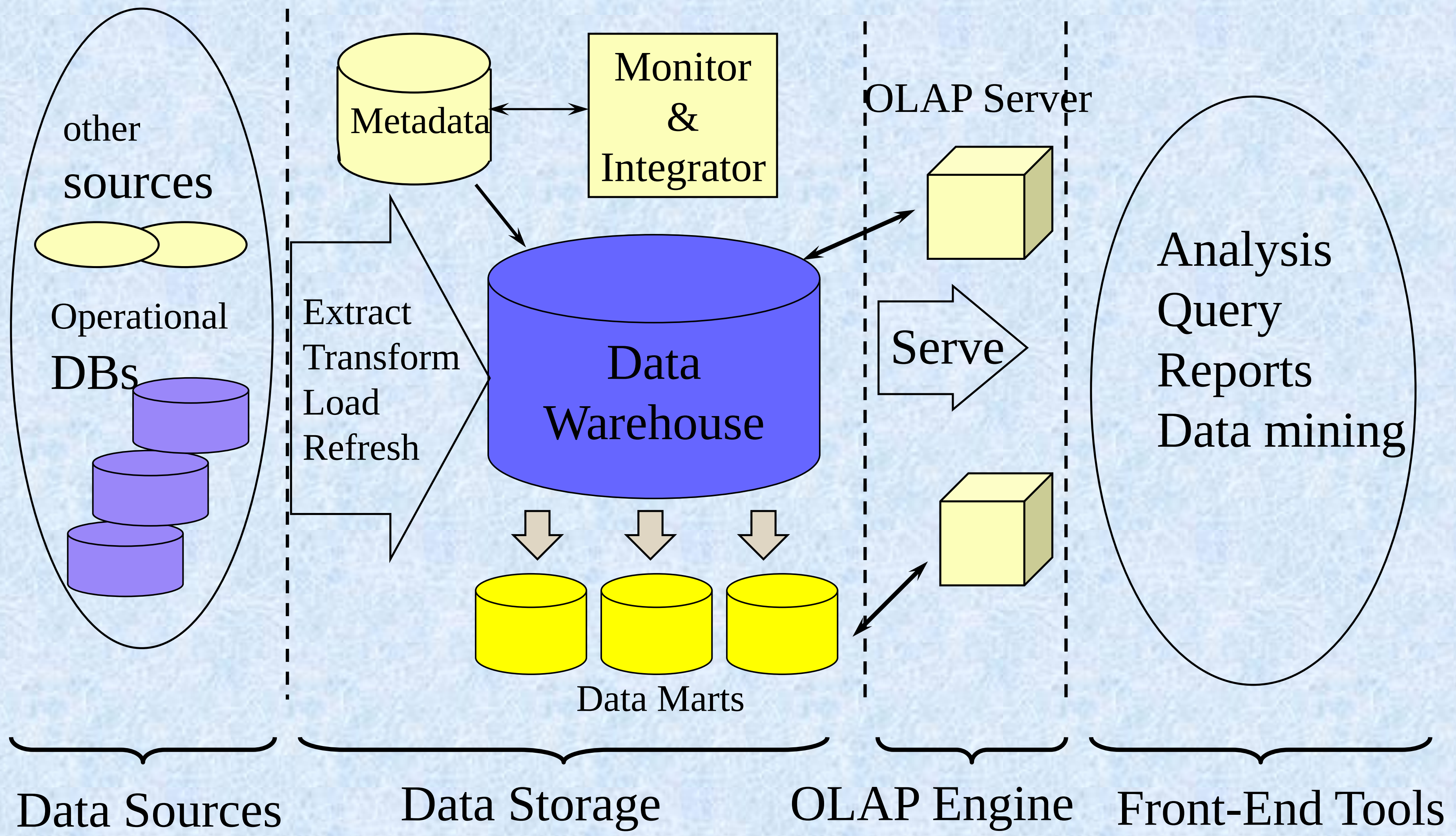


Each circle is called a footprint

Data Warehousing and OLAP Technology

- 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

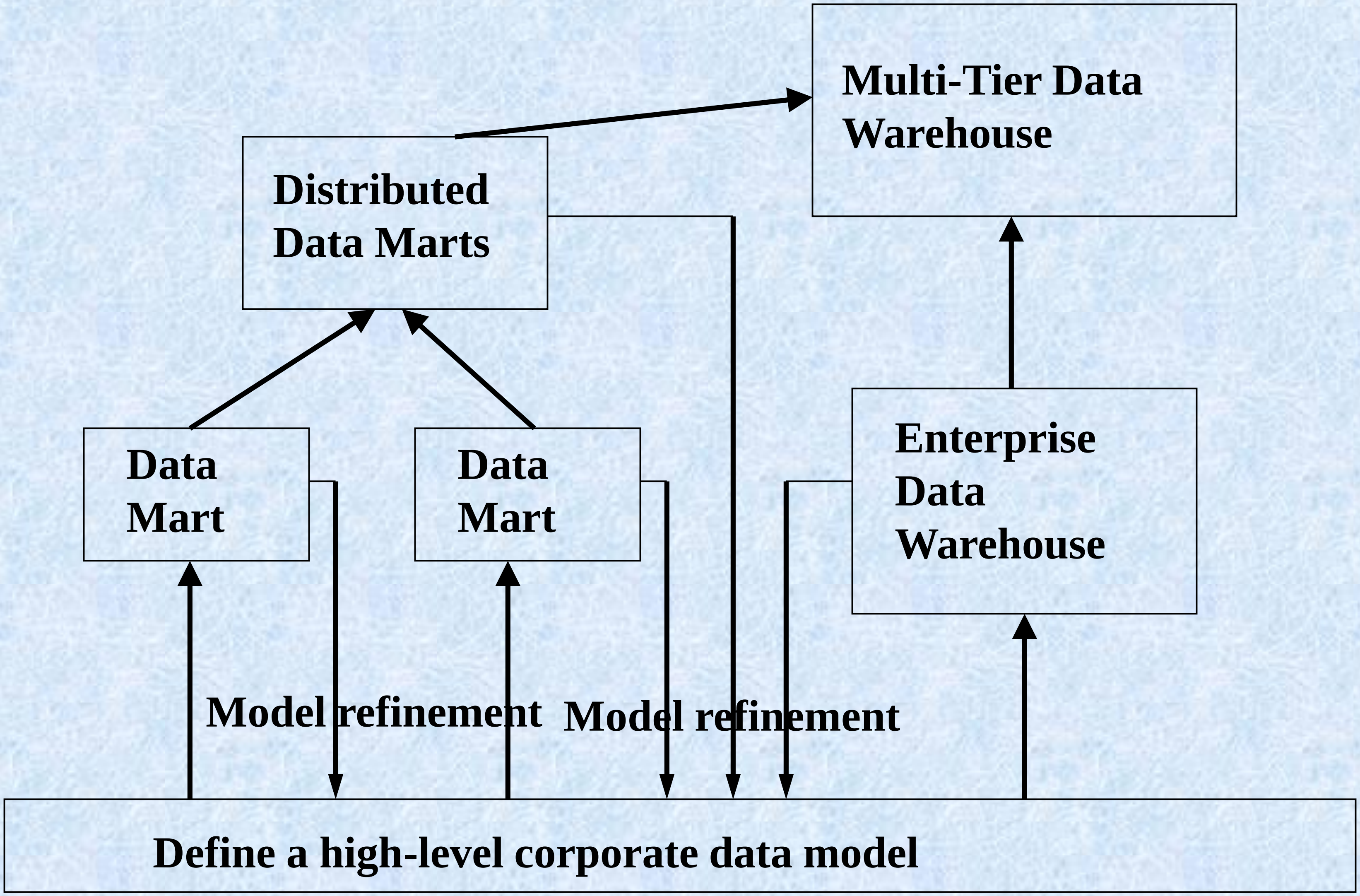
Multi-Tiered Architecture



Three Data Warehouse Models

- **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

Data Warehouse Development: A Recommended Approach



OLAP Server Architectures

- **Relational OLAP (ROLAP)**
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middleware 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

Data Warehousing and OLAP Technology

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
  - Data Warehouse Design
  - OLAP Modeling Methods
  - Optimization of Logical Model/Physical Model
  - Efficient Methods for Data Cube Computation
  - Plan and Implementation of Data Warehouse
- Further development of data cube technology
- From data warehousing to data mining

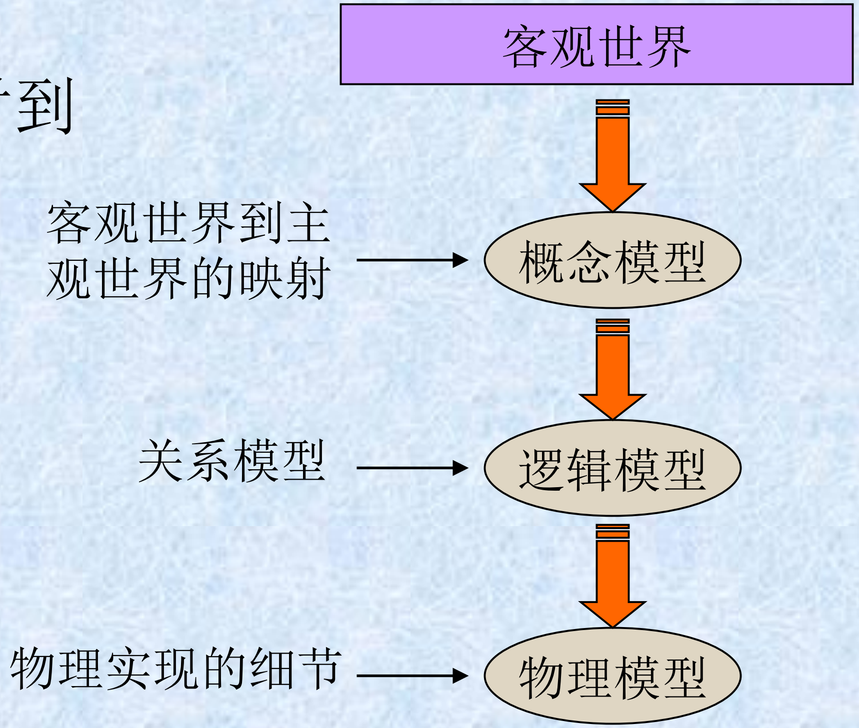
数据仓库设计

- 数据仓库设计和数据库设计的差异
  - 面向需求不同：数据库面向具体应用，需求一开始就很明确，而数据仓库是一个渐进的过程
  - 设计目标不同：OLTP vs. OLAP
  - 处理类型不同：面向操作型应用 vs. 面向分析型应用
  - 数据来源不同：业务员输入 vs. 已存在的业务系统数据
  - 系统设计的方法不同：
    - ✓数据仓库可以采用“数据驱动”的设计方法
    - ✓数据仓库设计可以分为数据仓库模型设计和数据装载接口设计两部分

数据仓库的设计步骤

将企业模型映射到数据仓库系统的过程

- 分析建立企业模型并映射到数据仓库概念模型
- 逻辑模型设计
- 物理模型设计



Data Warehousing and OLAP Technology

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
  - Data warehouse Design
  - OLAP Modeling Methods
  - Optimization of Logical Model/Physical Model
  - Efficient Methods for Data Cube Computation
  - Plan and Implementation of Data Warehouse
- Further development of data cube technology
- From data warehousing to data mining

OLAP建模方法

- 维表设计
  - 维的变化
  - 维表的共享
  - 层次信息和分类信息的位置
- 事实表设计
  - 事实表的特性
  - 通用数据和专用数据事实表

维表的变化（一）

- 维表通过记录因素的属性描述事件中包含的诸多因素
- 维表的本质是多维分析空间在某个角度上的投影
- 由于维表描述的是事物的属性，因此随着事物本身的变化，其属性也会产生改变
  - 如果该属性与决策没有太大关系，例如电话号码属性对于分析顾客购买行为没有什么作用，则此属性的变化可以忽略不计
  - 如果该属性与决策有关，例如某位顾客搬家后离超市更远了，我们试图分析其购买行为与家里距离变远有何关系，则不能将之忽略

维表的变化（二）

- 对于需要记录其改变的维，有若干方法可以进行处理
  - 当属性进行变化时，创建一个新记录
    - ✓例如：

345	顾客A	东城区

→

368	顾客A	西城区

    - ✓缺点：
      - 由于ID产生变化，被认为是两条记录
  - 创建一个新的字段，将新地址填入
    - ✓例如：

345	顾客A	东城区	西城区

    - ✓缺点：
      - 可扩展性不佳
  - 增加一个修订号码字段和当前标记字段
    - ✓例如：

345	0	过往	东城区

345	1	当前	西城区

    - ✓缺点：
      - 维表和事实表连接时需要采用“主关键字”＋“修订号码”，增加了事实表的复杂性

维表的变化（三）

- 最为理想的解决方案
  - 新建一个关键字客户ID，通过主关键字与之相连，使用时间字段标志当前的值
    - ✓缺点：相对较为复杂

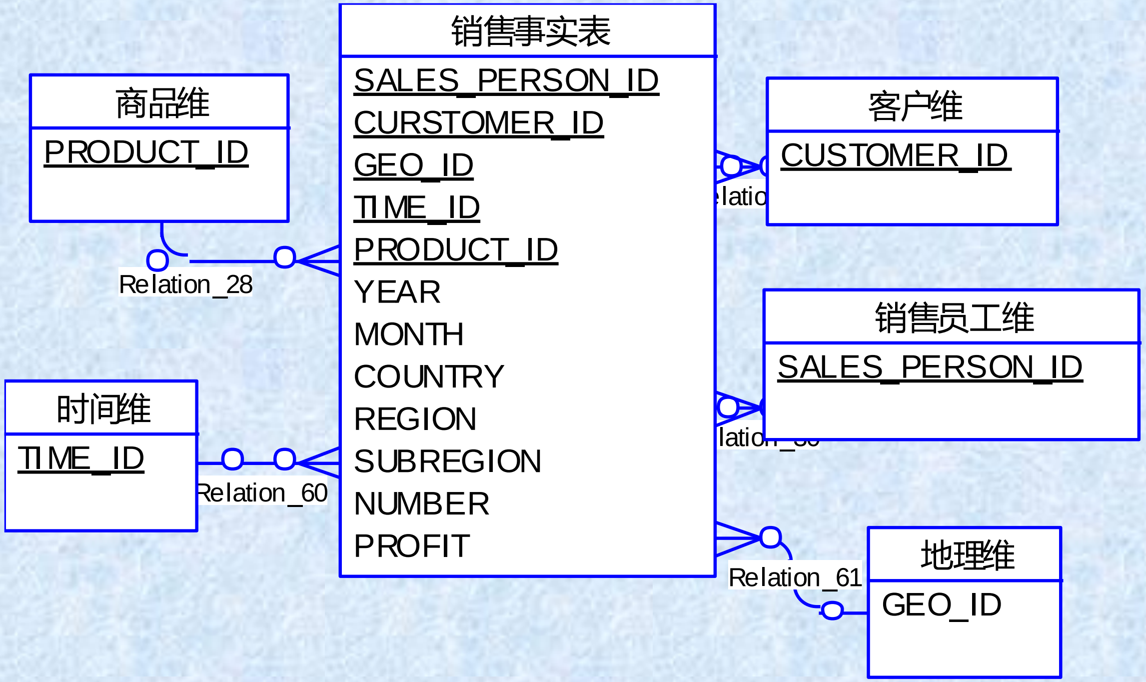
345	**2003／1**	当前	西城区
345	**1999／3**	过往	东城区
345	**1998／7**	过往	海淀区

维表的共享

- 多个维表中可能包含相同的属性：
  - 供货商维中包含地址维，而销售商维中可能也包含地址维，因而可能共享维表
  - 由于数据仓库中时间维的重要性，各个维中都有可能包含时间维，因而可能共享时间维表
- 可以采用扩展的星座结构来描述共享维表
- 具体内容请参见上节

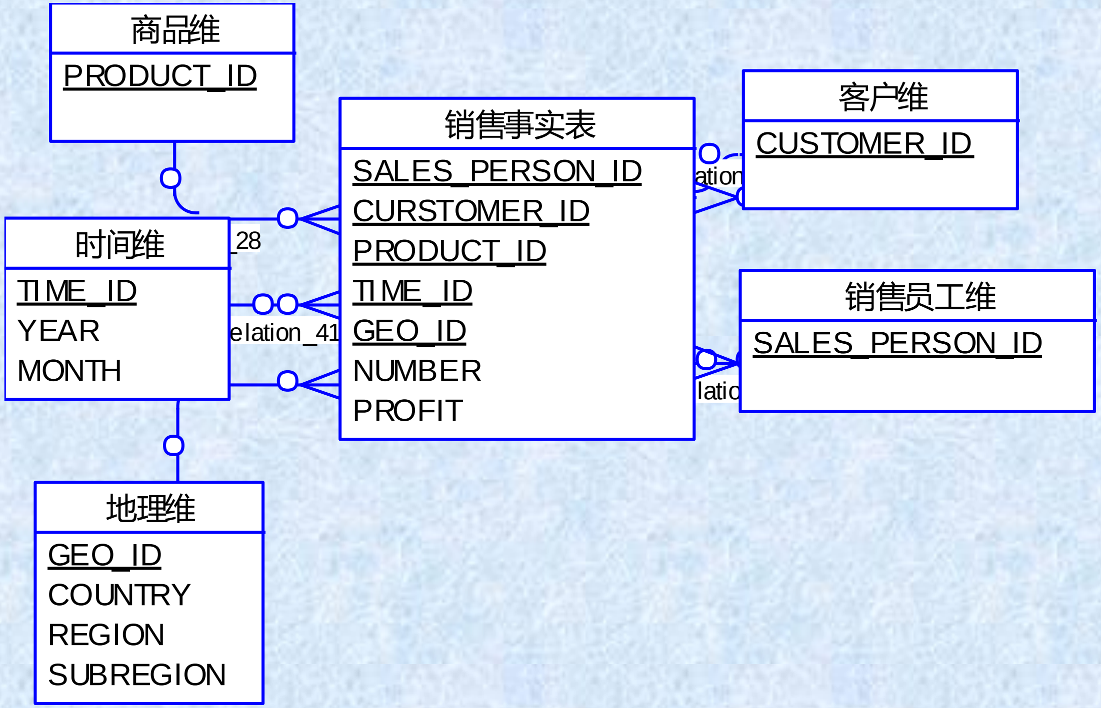
层次信息的位置（一）

- 将维层次信息放入事实表
  - 优点： 计算极为方便
  - 缺点： 事实表会因此变得极为庞大



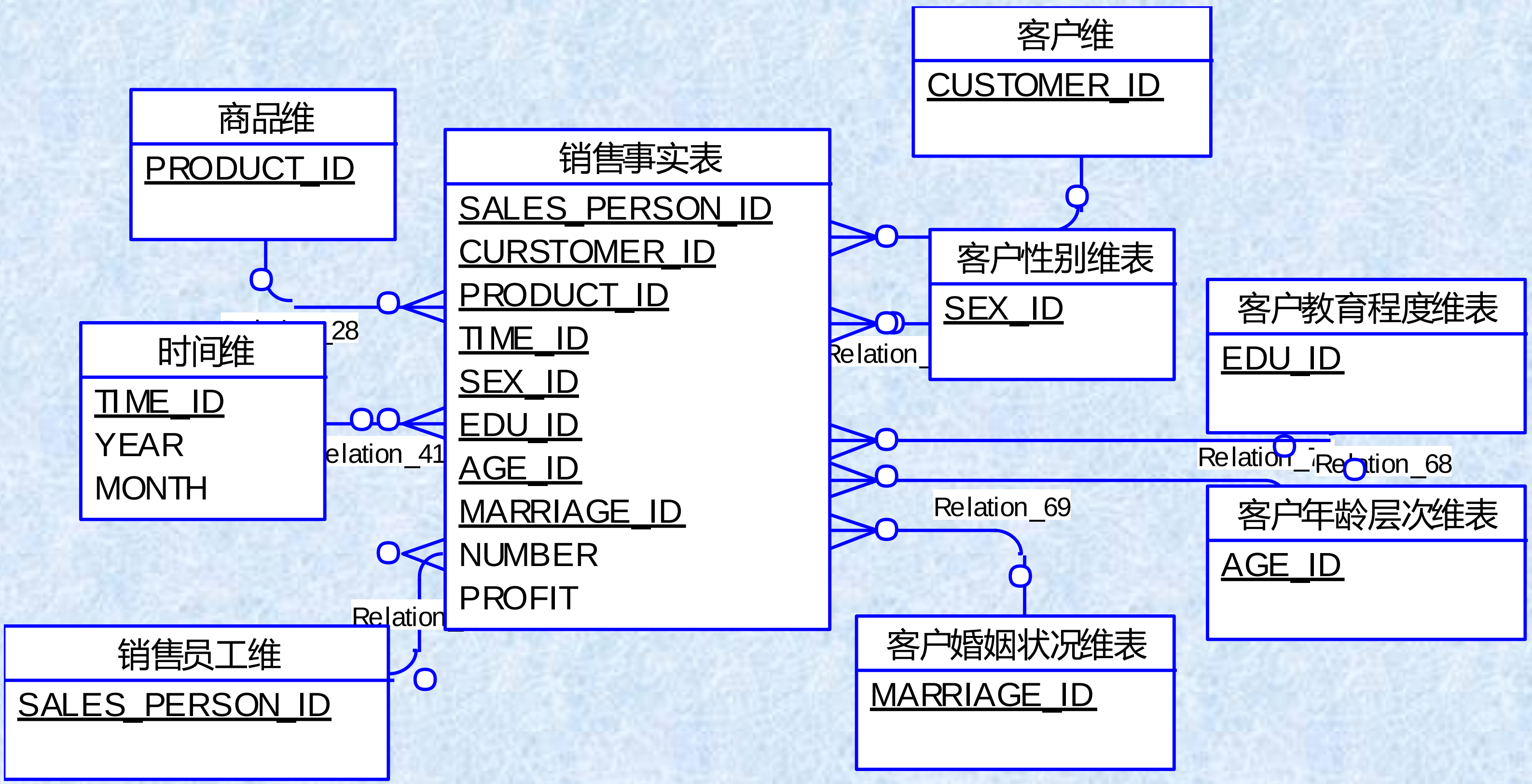
层次信息的位置（二）

- 将维层次放在各自的维表中，通过主关键字与事实表相连
  - 优点：减少了事实表的大小
  - 缺点：OLAP性能比上一种方式差



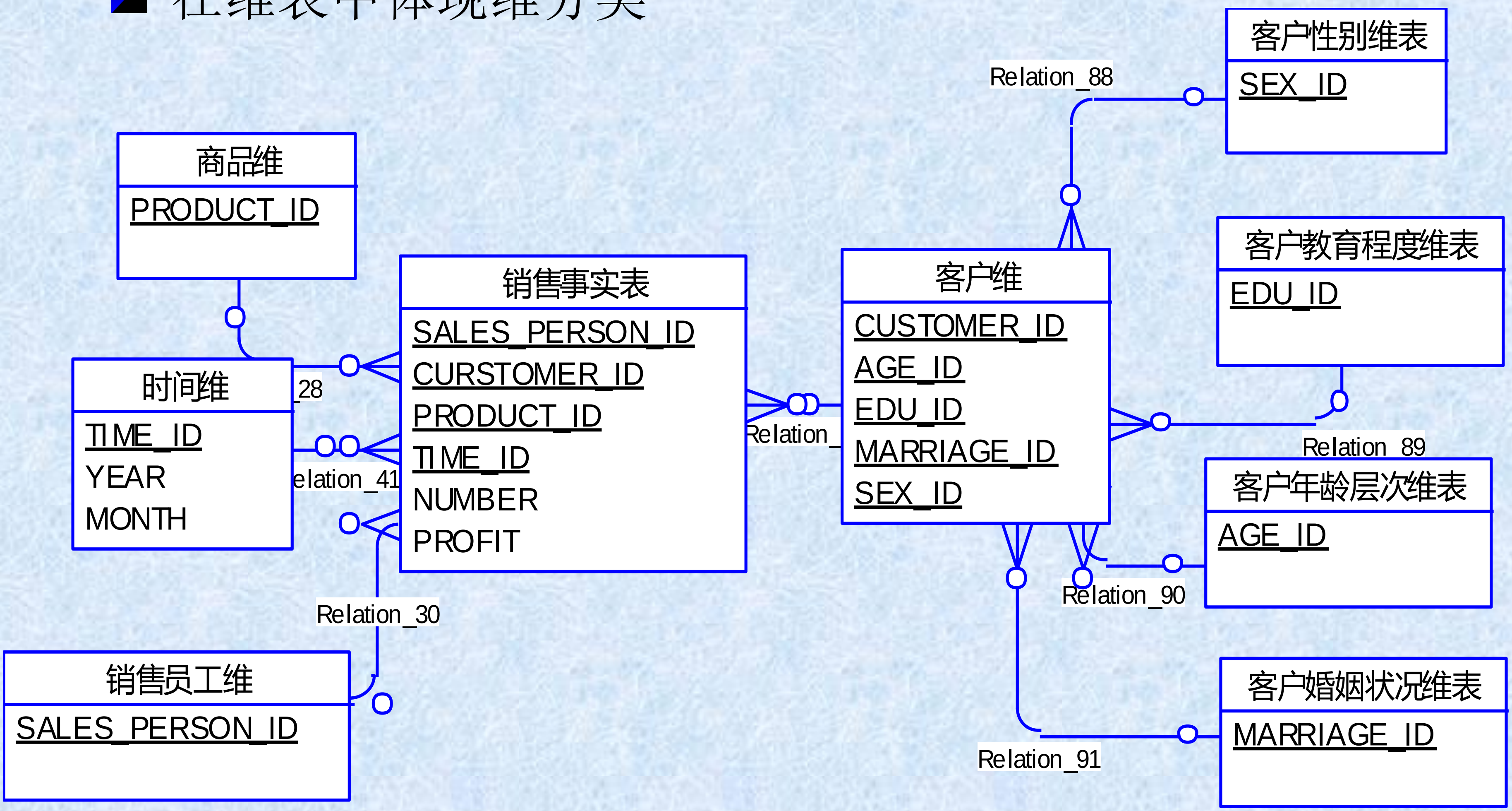
分类信息的位置（一）

- 在事实表中体现维分类
  - 优点： OLAP性能好
  - 缺点： 事实表的字段数和记录数增加



分类信息的位置（二）

- 在维表中体现维分类



事实表的特征

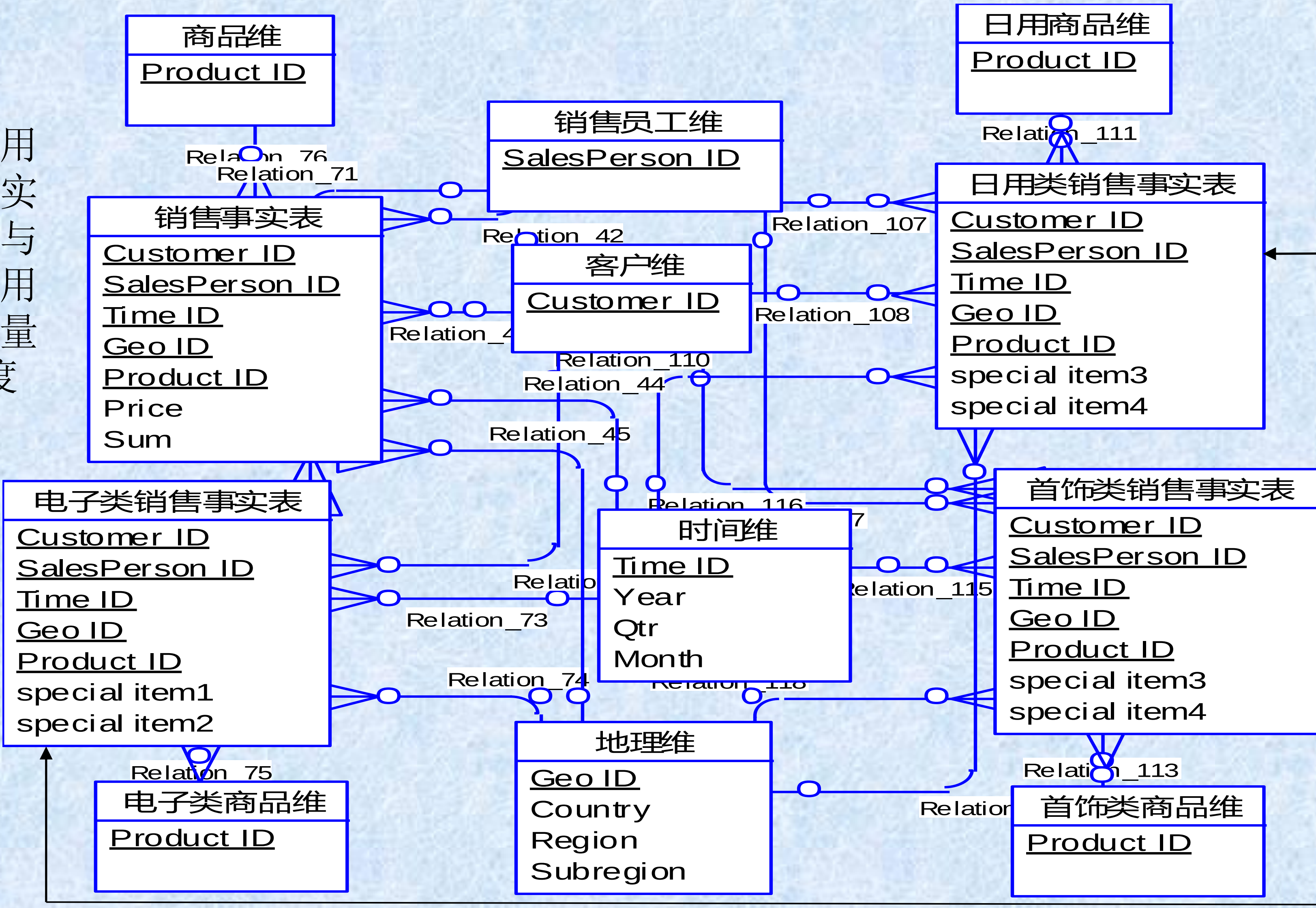
- 与维表相比，事实表具有以下特征：
  - 记录数量非常多
  - 除了度量外，其他字段都是维表或者中间维表（雪花模型）的关键字
  - 如果事实相关的维度很多，则事实表的字段数也会比较多
- 因此应当尽量减小一条记录的长度，才能避免事实表过大而难于管理
- 数据的粒度是影响事实表大小的关键因素，因而必须认真设计

通用数据和专用数据事实表

- 对应一个问题通常采用一个事实表，但在特殊情况下，也允许采用多个事实表
- 例如：超市里出售多种商品，由于商品本身分类不同，因此所采用的量度可能也不相同
  - 如果将这些量度全部置于一个事实表中，由于某种类型的商品的量度其他商品可能不具备，因此则不可避免将在事实表中造成大量的数据空缺
  - 解决办法：采用多个事实表，分为通用数据事实表与专用数据事实表加以管理

通用数据和专用数据事实表

通用事实表与通用的量度



专用事实表与专有的量度

Data Warehousing and OLAP Technology

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
  - Data warehouse Design
  - OLAP Modeling Methods
  - Optimization of Logical Model/Physical Model
  - Efficient Methods for Data Cube Computation
  - Plan and Implementation of Data Warehouse
- Further development of data cube technology
- From data warehousing to data mining

数据仓库的逻辑模型设计

- 系统数据量估算
- 数据粒度的选择
- 数据的分割
- 表的合理划分
- 去除纯操作数据
- 增加导出字段
- 定义关系模式
- 定义元数据存储
- 定义记录系统

数据仓库的逻辑模型设计

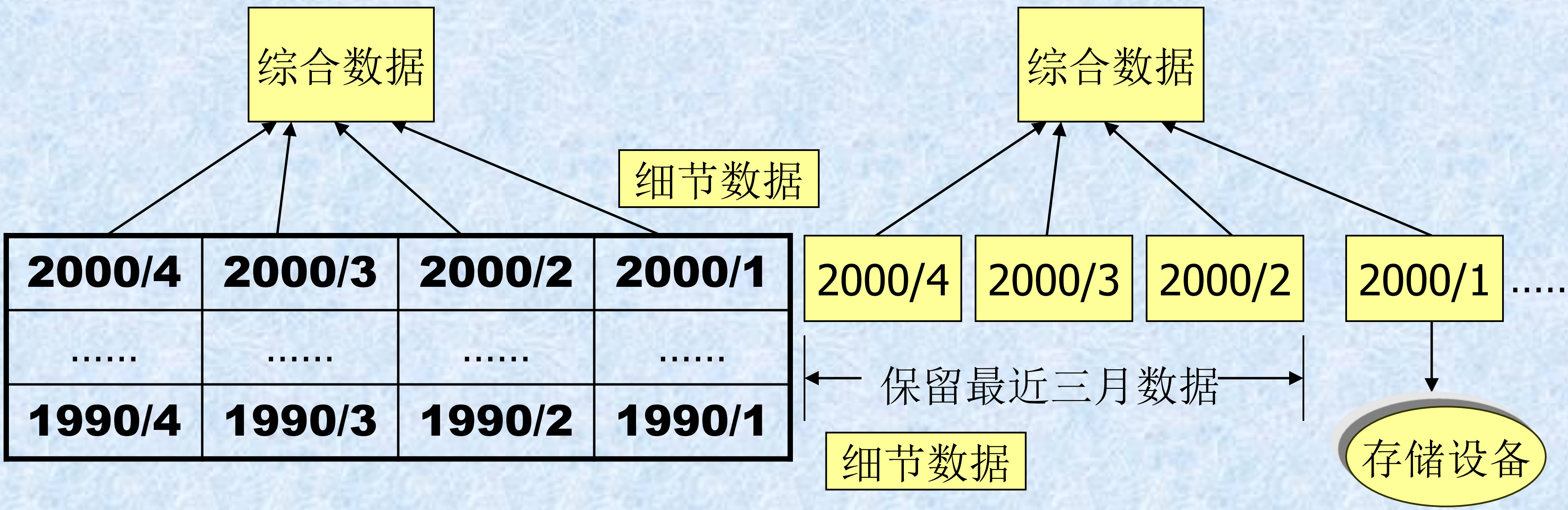
- 系统数据量估算
  - 设在概念模型中出现的表的个数为N（这些表中应不包含不会放进数据仓库的表），对于每个表i($0 \le i \le N$), 计算表的大小S_i和表的主关键字大小K_i，然后估计每张表i 在单位时间内最大记录数L_{max}和最少记录数L_{min},则数据仓库的粗略数据量在以下范围:

$$\alpha \times \left(\sum_{i=1}^{N} (S_i + K_i) \times L_{\max}^{i} \times T \right) \sim \alpha \times \left(\sum_{i=1}^{N} (S_i + K_i) \times L_{\min}^{i} \times T \right)$$

  - 其中,T是数据在数据仓库中存在的周期。α是考虑由于数据冗余和数据索引而使数据量增大的冗余因子,通常取1.2~2
  - 本公式的含义是：
    - ✓ 数据仓库数据量＝{累加[（每张表记录大小＋每张表主关键字大小）*每张表单位时间内记录的数量]*存储时间}*冗余因子
  - 公式估算出的结果仅能作为参考

数据粒度的选择

- 数据量较小的情况下使用单一的数据粒度，即直接存储细节数据并定期在细节数据基础上进行数据综合
- 对于大数据量需要采用双重粒度，对于细节数据只保存近期的数据在数据仓库之中，当保留周期到达时，将距离当前较远的数据导出到磁带或存储设备上



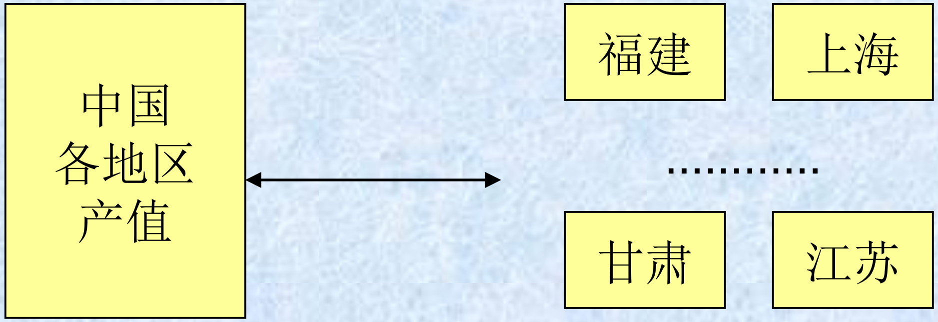
单一数据粒度　　多重数据粒度

数据粒度策略

一年内数据量（行）	五年内数据量（行）	数据粒度策略
10,000	**100,000**	简单的单一粒度策略
100,000	**1,000,000**	如果选择单一粒度，则需要认真设计
1,000,000	**10,000,000**	最好使用双重粒度
10,000,000	**20,000,000**	必须采用双重粒度且认真设计

数据的分割

- 为何要进行数据分割?
  - 数据仓库中数据量过大时，检索速度很慢
- 数据分割是指将数据分散到各自的物理单元里以便能够独立处理，以提高数据处理的效率
- 数据分割没有固定的标准，分割的方法和粒度应当根据实际情况确定
  - 通常选择时间、地点、业务等划分
  - 一般按照时间分割数据分布比较均匀，因此按照时间分割最为常见



合理的表划分

- 直接存储字段数目很大的表，会造成以下问题：
  - 各个字段的更新频率不一，放在一张表里造成数据追加工作的浪费
  - 各个字段的访问频率不一，放在一张表里影响访问效率
- 因此需要对表中的内容进行合理的划分
  - 按数据的稳定性划分
  - 按业务规则进行表划分(略)

按数据稳定性进行表划分

原始表	几乎不变的数据	有时改变的数据	经常改变的数据
商品ID 数据加载时间 商品名称 商品类型 商品规格 商品质量 主要原料供应商 最后订购时间 最后订购数量 生产成本	商品ID 商品名称 商品类型 商品规格 商品质量	商品ID 主要原料供应商 生产成本	商品ID 数据加载时间 最后订购时间 最后订购数量

避免了整张表记录快速增长的现象，节约了存储空间

删除纯操作数据及增加导出字段

- 在将业务系统中的数据抽取到数据仓库系统中的过程里，如果发现某些数据对于决策没有作用，属于纯操作型数据,则可以将之删除
  - 例：“收款人”字段
- 导出数据本身是冗余的，但是增加导出字段有利于数据以后的使用
  - 例：在按月综合表中，可以加入“平均价格”，“供货总价”，“供货总数量”等导出字段

Meta Data

- **What is meta data?**
  Meta data are data describing data

- **In data warehouse, there are several data levels:**
  - meta data
  - current detailed data
  - older detailed data
  - lightly summarized data
  - highly summarized data

Meta Data Storage

- **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 definition, 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
  - Business data
    - ✓ business terms and definitions, ownership of data, charging policies

记录系统的定义

- 记录系统是操作型元数据的一部分
- 记录系统指明数据仓库中关系表的各个字段来源于业务数据库何处
  - 例：

主题	表名	属性字段	源数据库	源表名	源字段
商品	供应表	交易**ID**	采购	采购表	交易号
商品	供应表	商品**ID**	采购	采购表	商品号
商品	供应表	供应商**ID**	采购	采购表	供应商号
商品	供应表	供货数量	采购	采购表	供货数量
商品	供应表	供货价格	采购	采购表	供货价
………	………	………	………	………	………

数据仓库物理模型设计

- 确定数据的存储结构
- 索引策略
- 数据存储策略与性能优化
  - 多路聚集优化
  - 表的归并
  - 分割表的存放
  - 按列存储
  - 存储分配优化
- 数据装载接口设计
- 并行优化设计

Define the Storage Structure

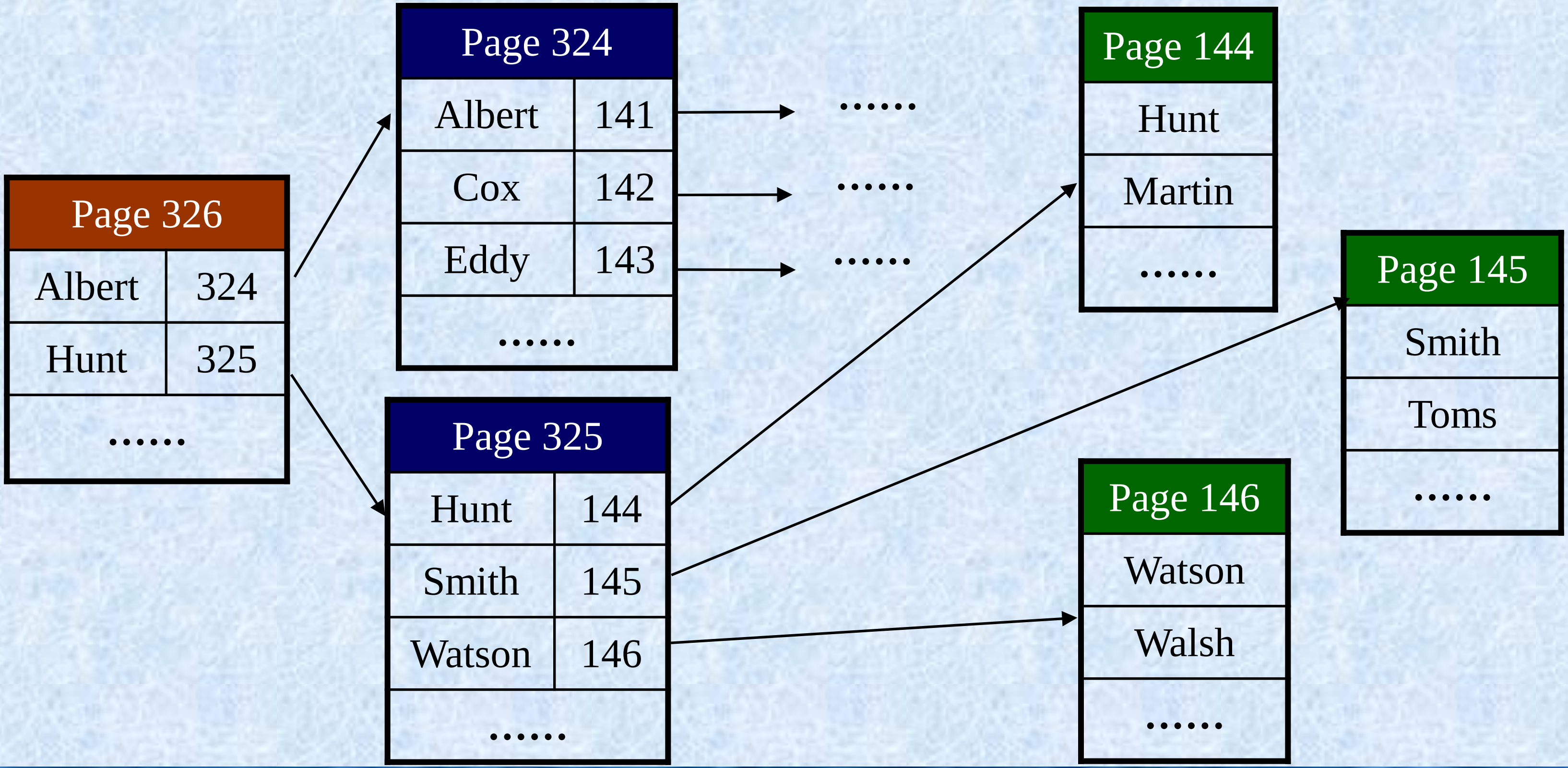
- 一般的数据库数据量相对较小，除非业务要求必须保证数据的安全性和可恢复性，否则可以不采用并行存储结构
- 数据仓库由于数据的巨量存储，必须采用并行存储结构，例如**RAID**

Indexing Technology: why not B-tree?

- **B-tree is a widely-used technology in database**
  - B-tree indexing has high performance when used to find some records in the database
- **While B-tree is not a good technology for data warehouse, Why?**

B-tree Indexing Technology

- **Because:**
  - B-tree demands that the attribute must have many different values, such as itemID, customer ID, etc.
  - B-tree demands that the query should have simpler conditions and less results
  - The space complexity and time complexity of creating B-tree are huge



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

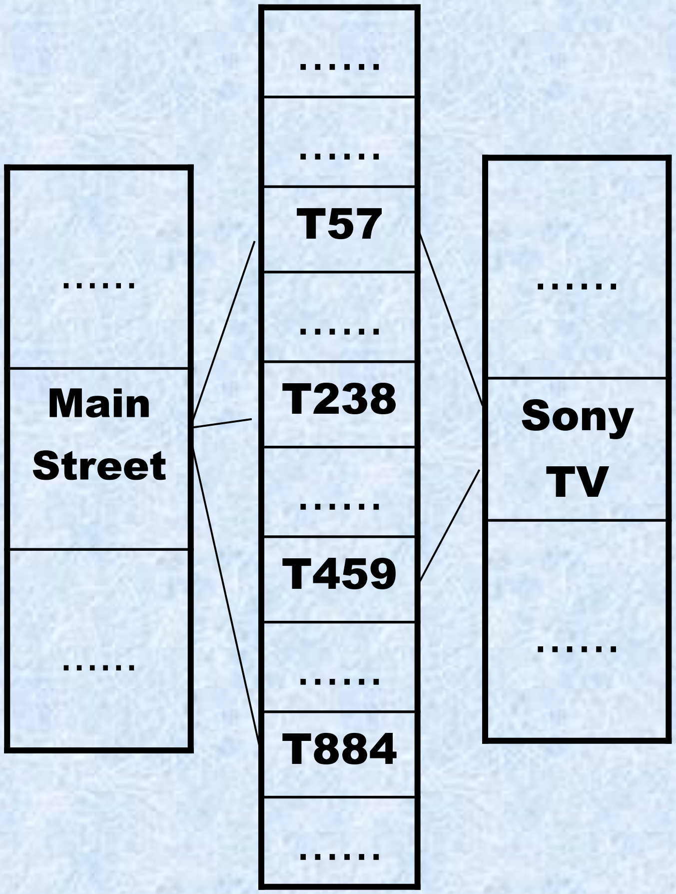
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on Type

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

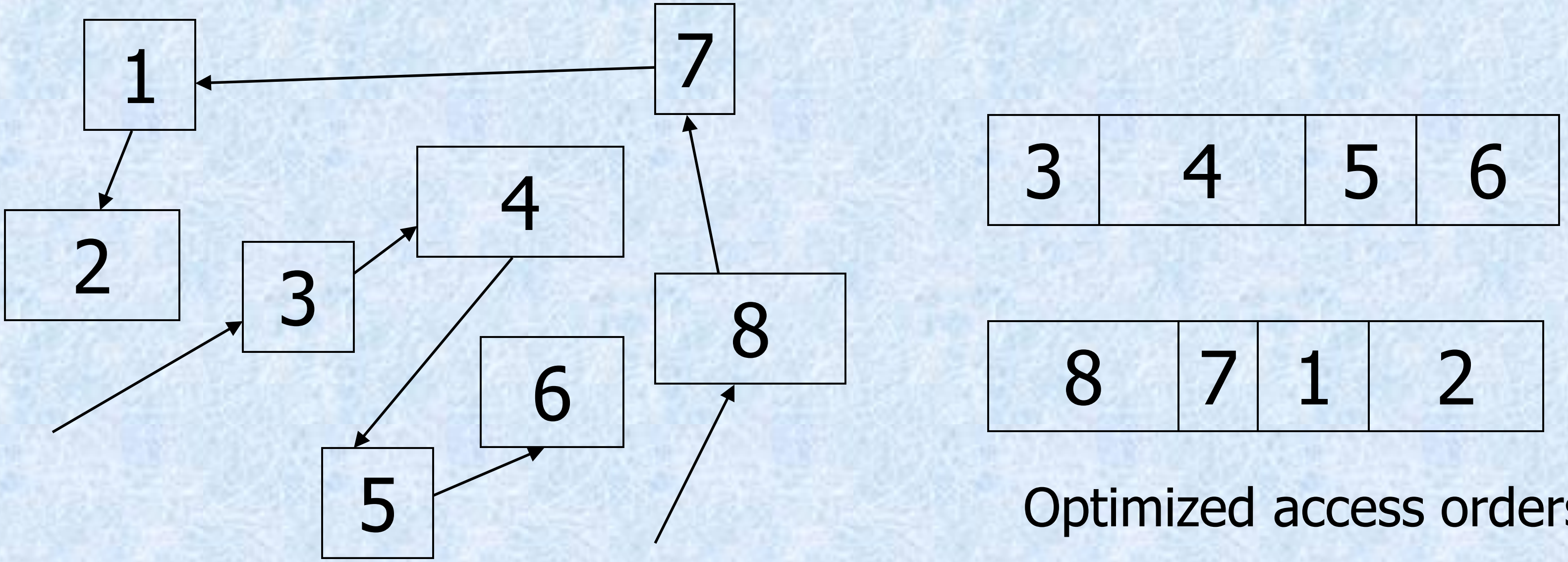
Indexing OLAP Data: Join Index

- **Join index: JI(R-id, S-id) where R (R-id, …) $\bowtie$ S (S-id, …)**
- **Traditional indices map the values to a list of record ids**
- **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



Data Storage Strategy

- **Table Mergence(I)**



Common access orders

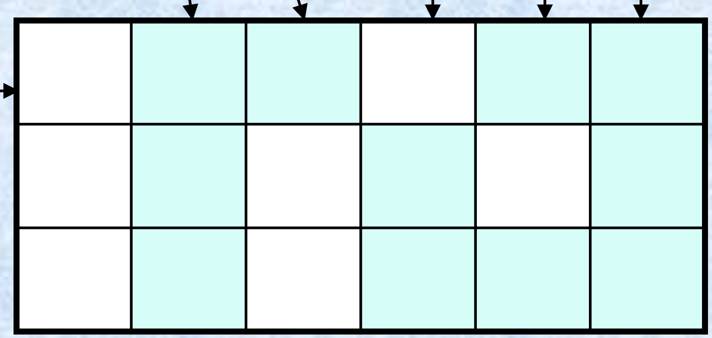
Optimized access orders

Data Storage Strategy

- Table Mergence(II)

ID	NAME	TYPE
1	Dell PC	Computer
2	Cisco 2600	Router
3	IBM PC	Computer
4	HP Laserjet 1000	Printer

ID	Storage ID	Amount
1	5	300
1	3	35
2	3	67
2	2	348
3	6	20
4	1	908



Data Storage Strategy

- **Add Redundancy**

ID Name Type Model Weight	ID Storage ID Amount

Original tables

ID Name Type Model Weight	ID Storage ID Amount Name

Add redundancy

Data Storage Strategy

Dividing Table

- In logical design, big tables can be divided into small tables. When a big table is visited, the table can be substituted with small ones.
- In physical design, distributed storage methods can be used.
- Tables can be stored into a disk array

Data Warehousing and OLAP Technology

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
  - Data warehouse Design
  - OLAP Modeling Methods
  - Optimization of Logical Model/Physical Model
  - Efficient Methods for Data Cube Computation
  - Plan and Implementation of Data Warehouse
- Further development of data cube technology
- From data warehousing to data mining

Efficient Methods for Data Cube Computation

Data Cube: High Efficiency Computing

- MutiWay Array Aggregation
- BUC

Data Cube: High Efficiency Computing

- **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.

Data 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

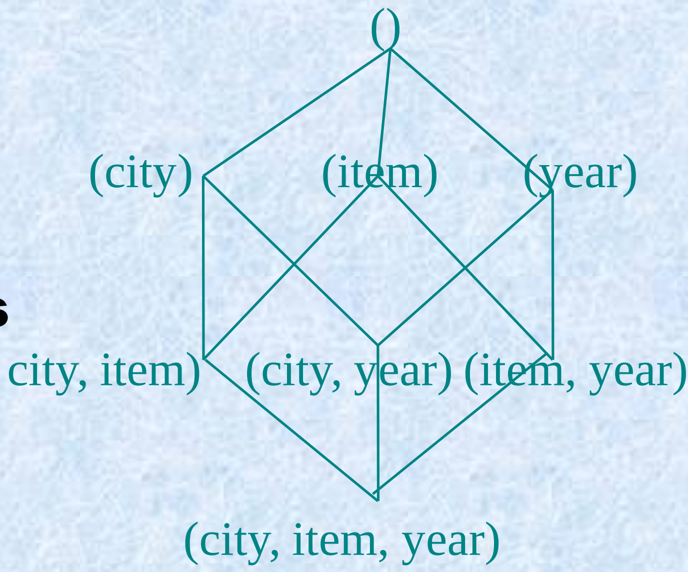
- **Need compute the following Group-Bys**

    (date, product, customer),

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

    (date), (product), (customer)

    ()



Iceberg Cube

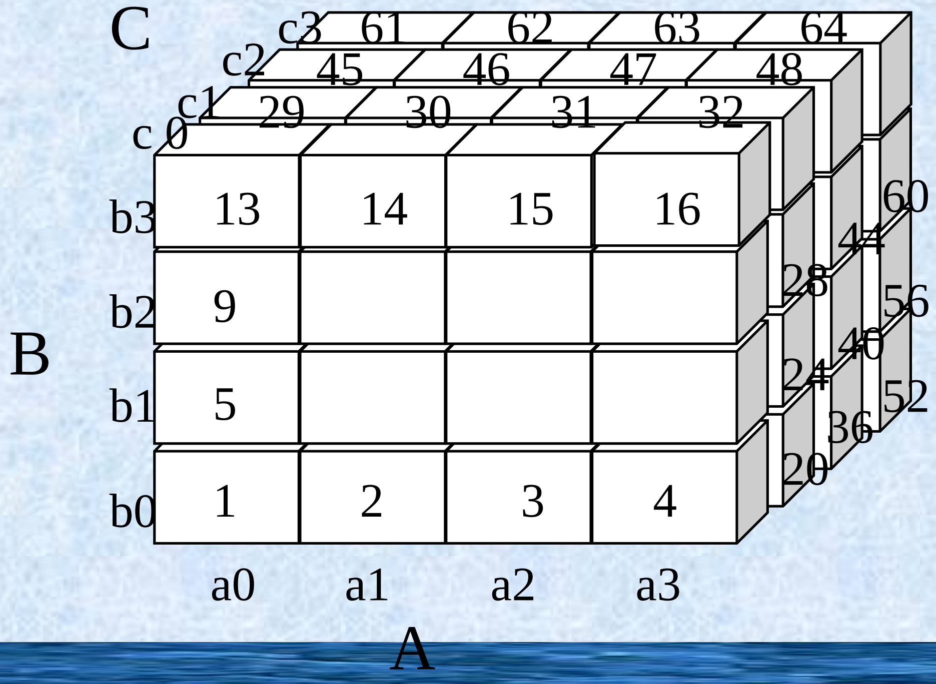
- **Computing only the cuboid cells whose count or other aggregates satisfying the condition like**

  HAVING COUNT(*) >= *minsup*

- **Motivation**
  - Only a small portion of cube cells may be "above the water" in a sparse cube
  - Only calculate "interesting" cells—data above certain threshold
  - Avoid explosive growth of the cube

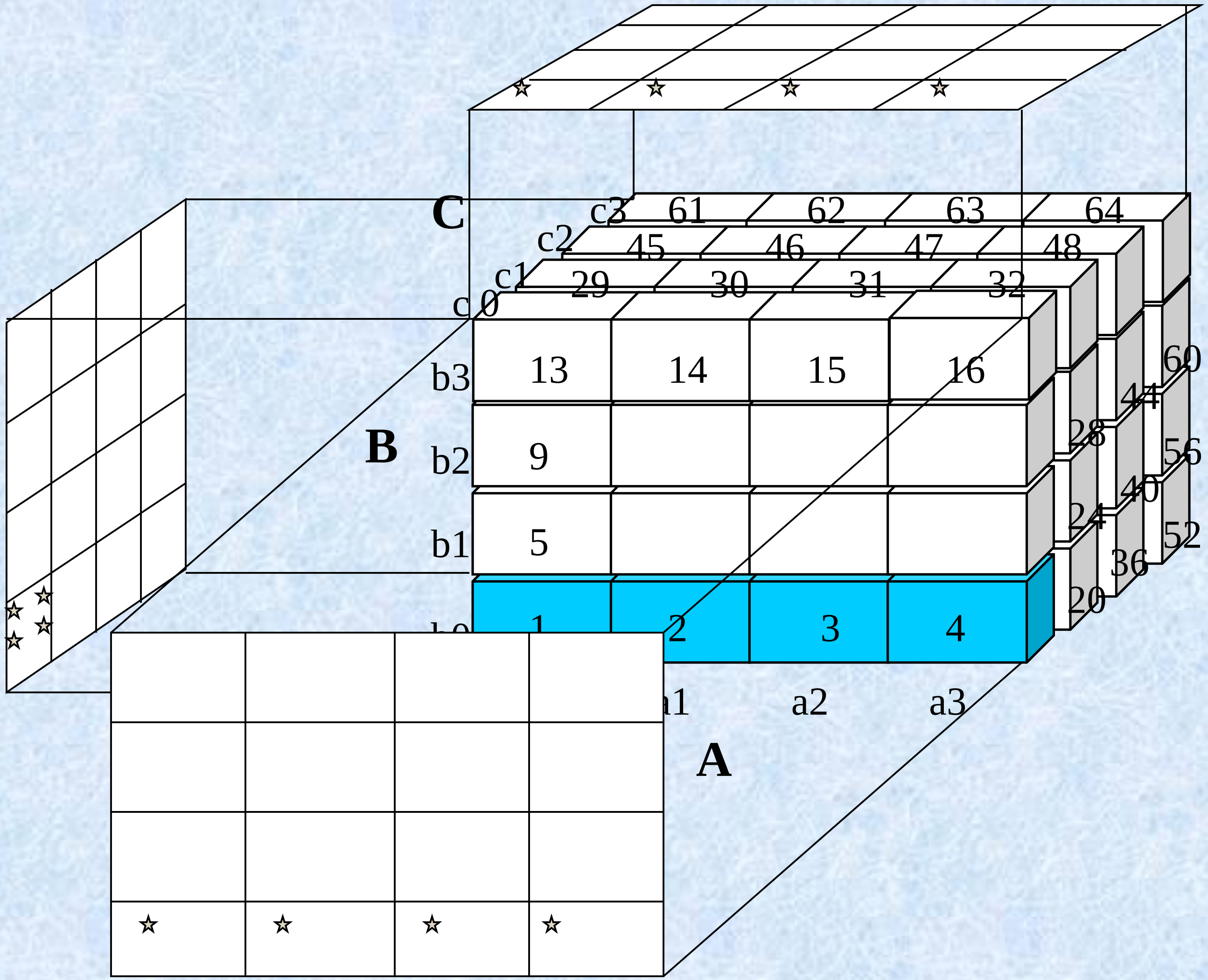
MutiWay Array Aggregation

- **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.**

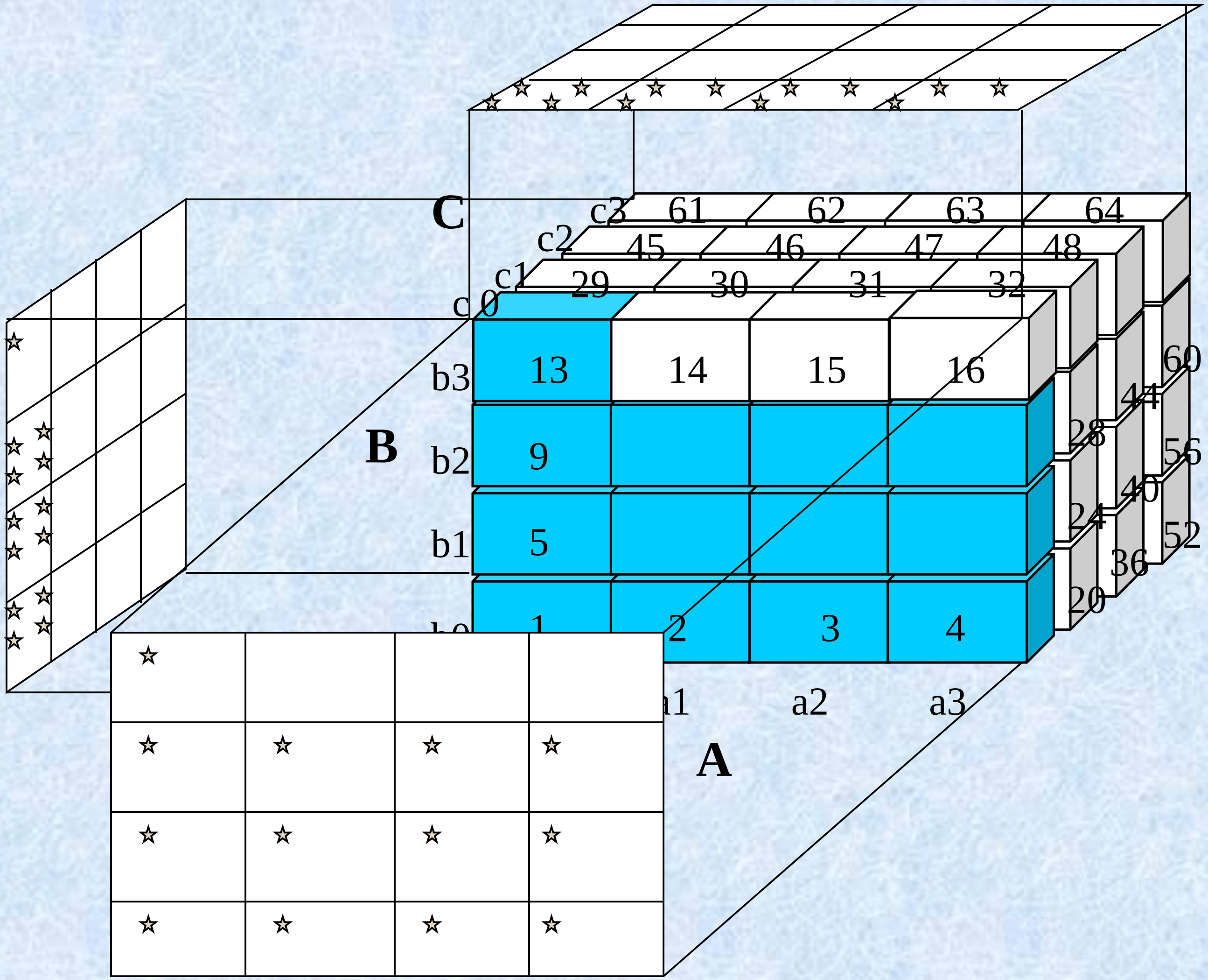


What is the best traversing order to do multi-way aggregation?

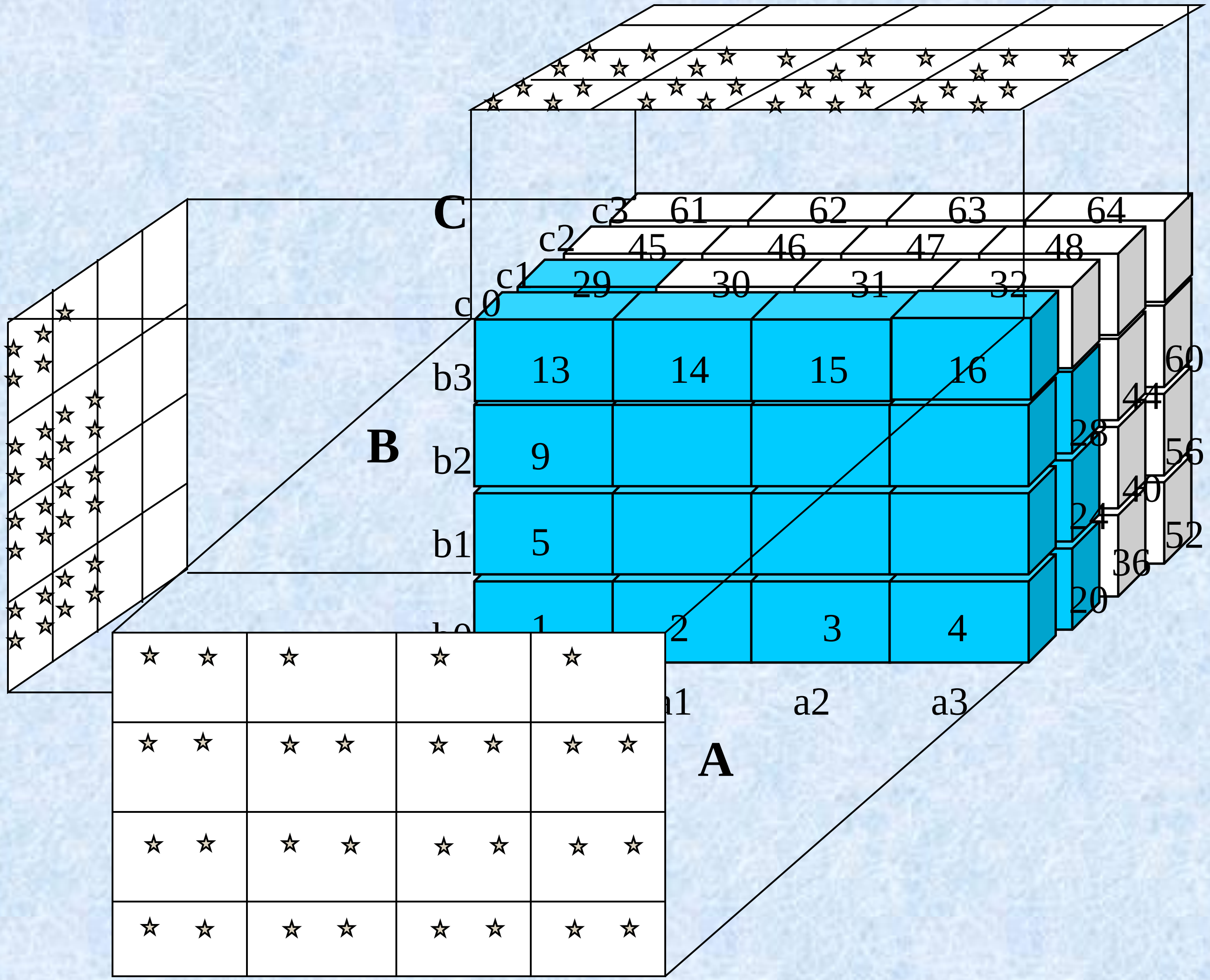
MutiWay Array Aggregation



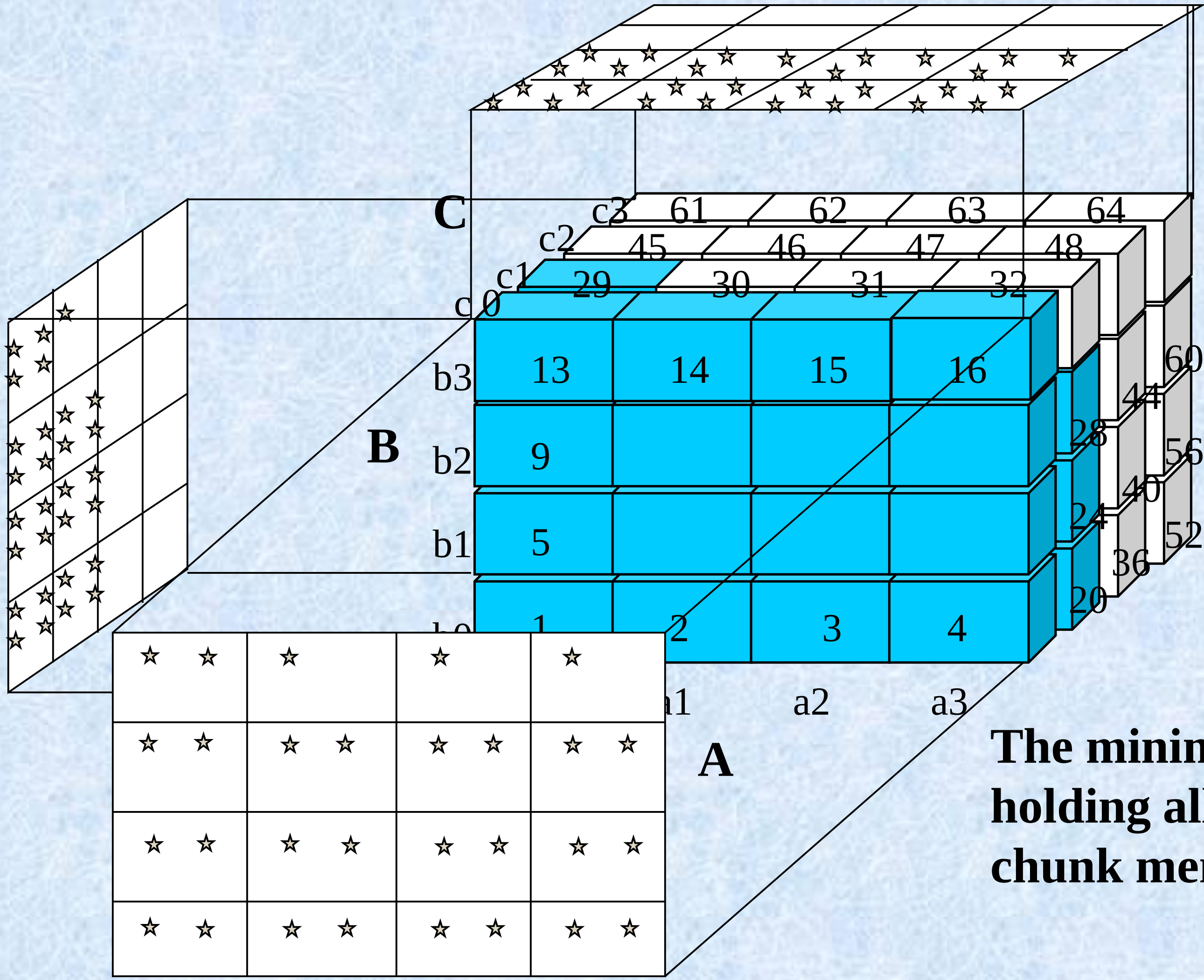
MutiWay Array Aggregation



MutiWay Array Aggregation



MutiWay Array Aggregation



If the dimensions are sorted as:

A: 10, B:100, C: 1000

For BC plane:

100*1000=100,000

For AC plane:

40*1000=40,000

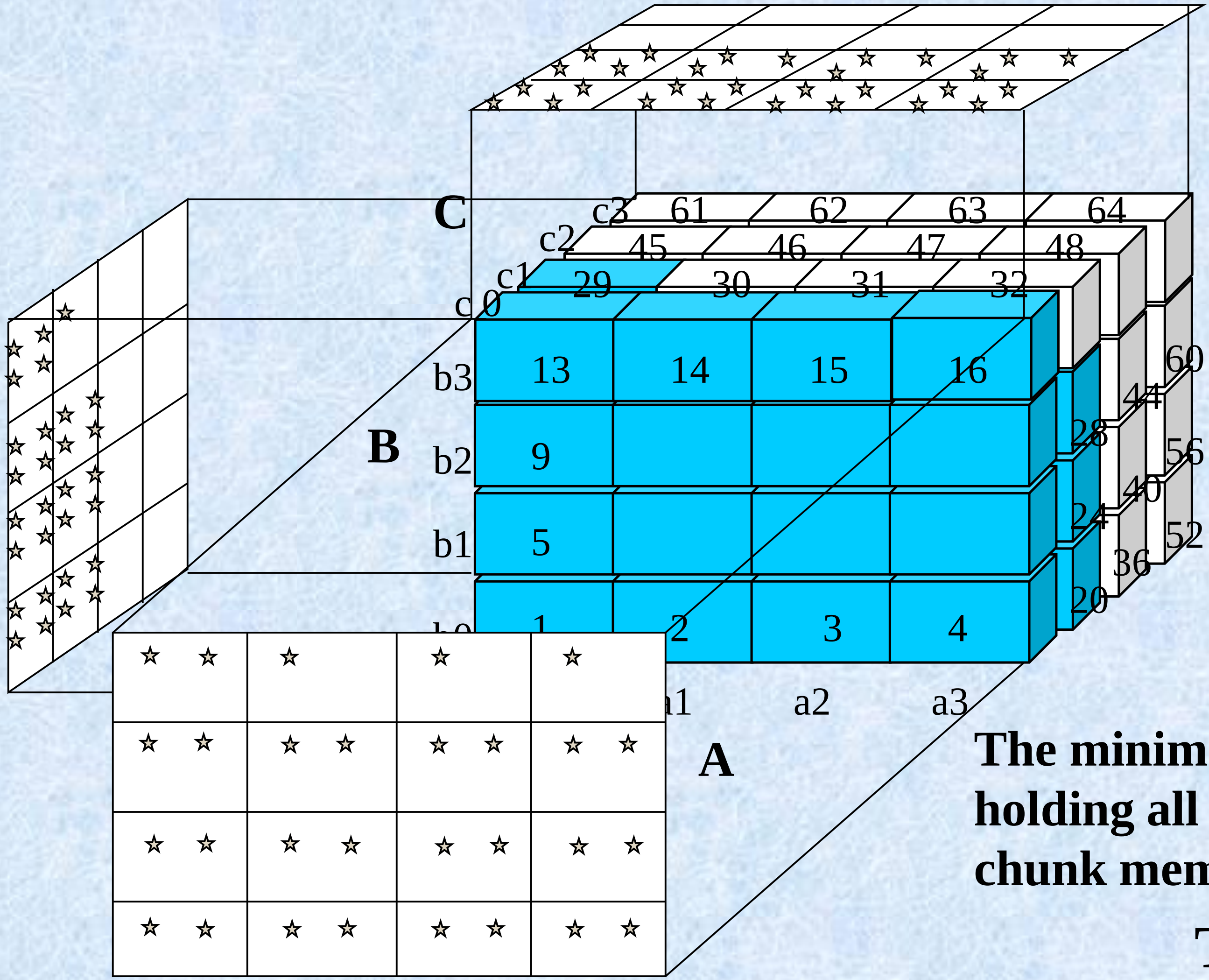
For AB plane:

40*400=16,000

The minimal memory required for holding all relevant 2-D planes in chunk memory:

Total: *156,000*

MutiWay Array Aggregation



If the dimensions are sorted as:

A: 1000, B:100, C: 10

For BC plane:

10*100=1,000

For AC plane:

10*4000=40,000

For AB plane:

400*4000=1,600,000

The minimal memory required for holding all relevant 2-D planes in chunk memory:

Total: *1,641,000*

MutiWay Array Aggregation

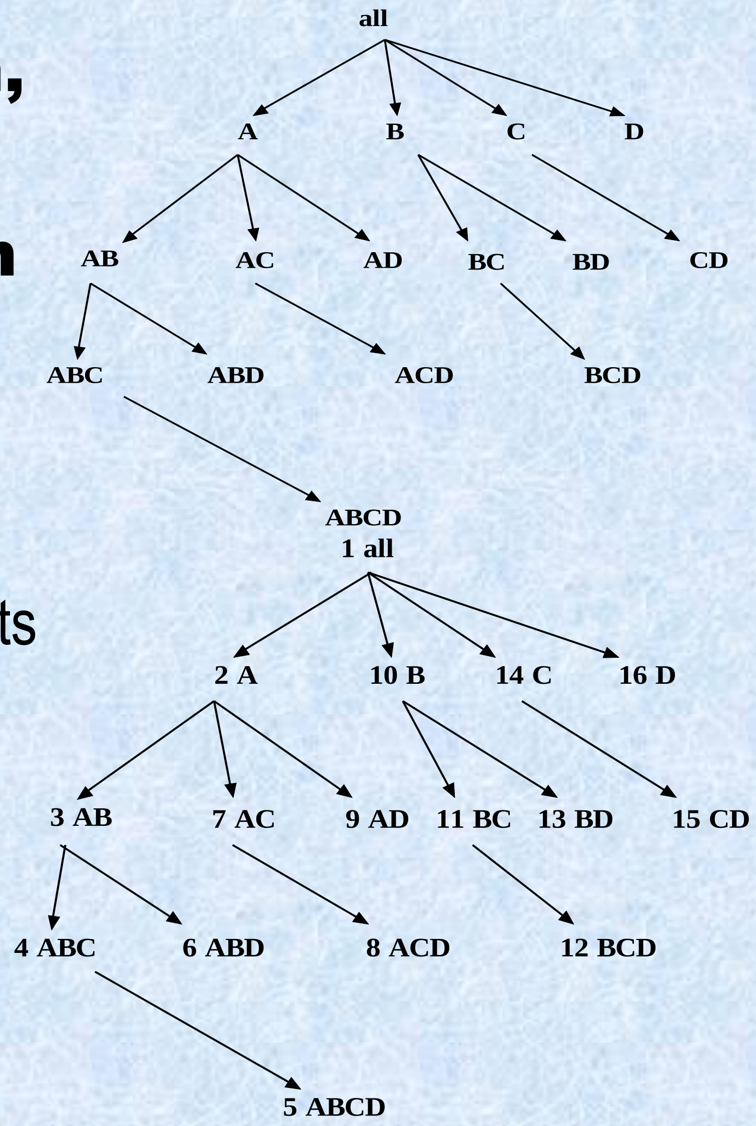
- **Method: the planes should be sorted and computed according to their size in ascending order.**
  - Idea: keep the smallest plane in the main memory, fetch and compute only one chunk at a time for the largest plane
- **Limitation of the method: computing well only for a small number of dimensions**
  - If there are a large number of dimensions, iceberg cube computation methods can be explored

Processing OLAP Query Effectively

- **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**

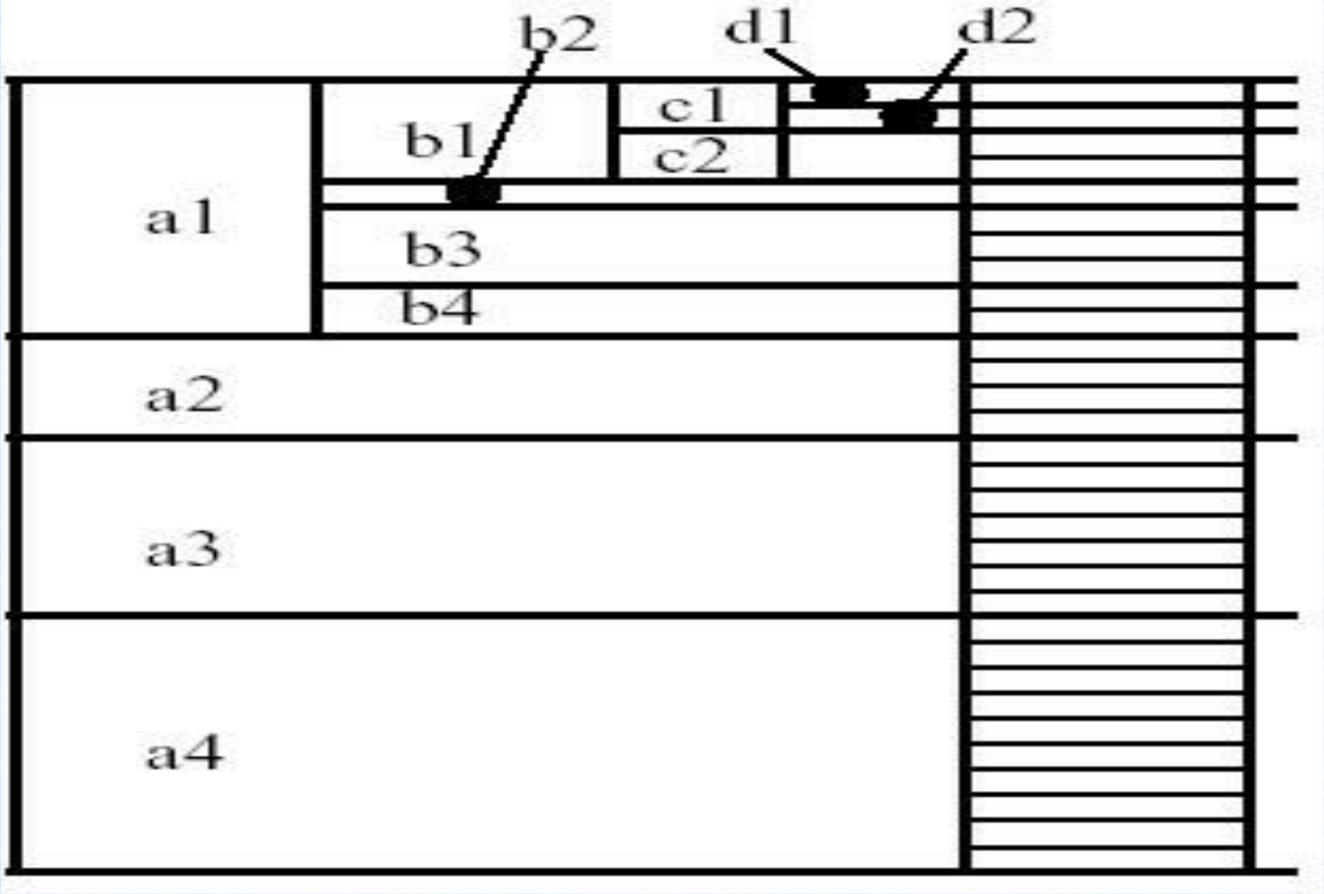
Bottom-Up Computation (BUC)

- **BUC (Beyer & Ramakrishnan, SIGMOD'99)**
- **Bottom-up cube computation**
- **Divides dimensions into partitions and facilitates iceberg pruning**
  - If a partition does not satisfy *min_sup*, its descendants can be pruned
  - If *minsup* = 1 $\Rightarrow$ compute full CUBE!



BUC: Partitioning

- Usually, entire data set can't fit in main memory
- Sort *distinct* values, partition into blocks that fit

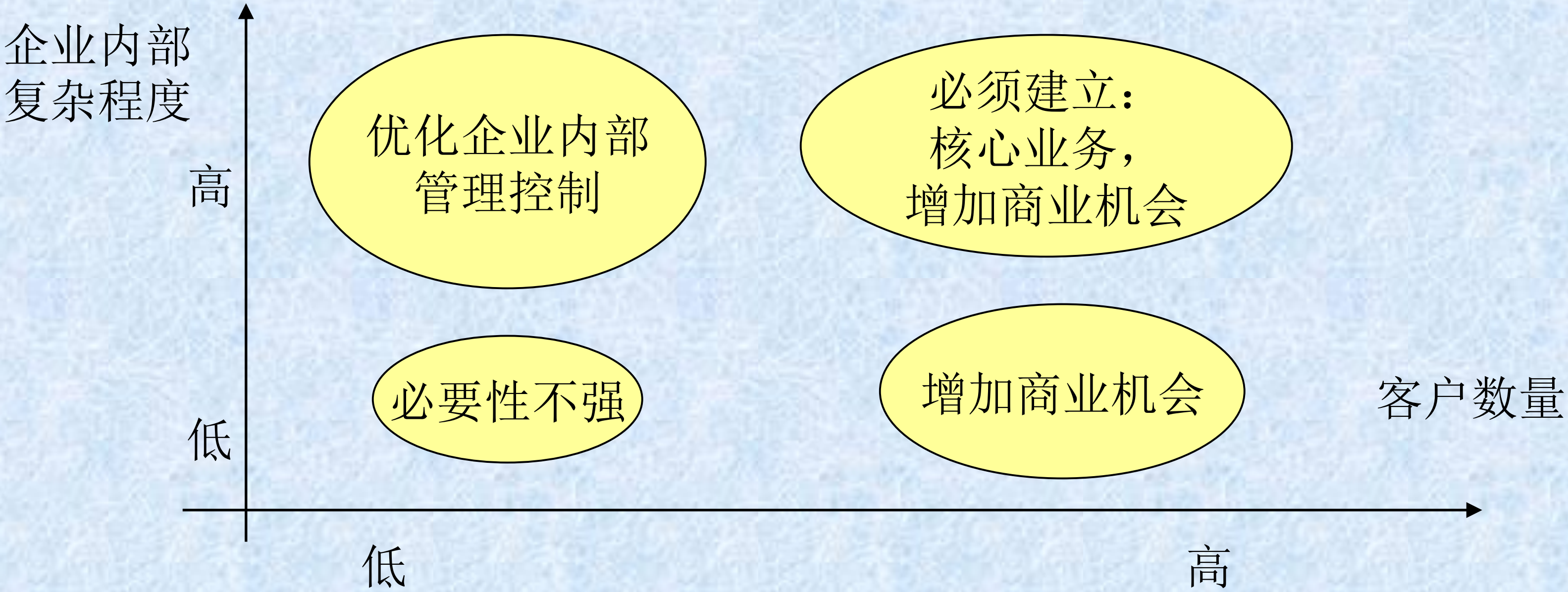


Data Warehousing and OLAP Technology

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
  - Data warehouse Design
  - OLAP Modeling Methods
  - Optimization of Logical Model/Physical Model
  - Plan and Implementation of Data Warehouse
- Further development of data cube technology
- From data warehousing to data mining

数据仓库的投资分析

- 数据仓库的应用目标
  - 企业的核心业务
  - 优化企业内部管理控制
  - 为企业增加商业机会
- 建设数据仓库的必要性



可通过计算ROI（Return of Investment）来衡量投资回报的价值

数据仓库主题的选择和阶段规划

- 数据仓库的实施是一个极为复杂的长期过程，因此，应选择当前最急需、能在短期内产生效益、业务模型清晰的任务首期实现
- 选择首期实现主题的参考原则：
  - 优先实现管理者目前需求最迫切和最关心的主题
  - 优先选择能在短期内产生效益的主题
  - 推后选择业务逻辑准备不充分的主题
  - 推后实施技术难度较大、可实现性较低、投资风险大的主题
- 维护阶段
  - 数据仓库的维护极为重要，一般数据仓库在建立完成之后，都需要一至两年的维护
  - 数据仓库的维护过程就是DSS逐步产生效益的过程

数据仓库后端工具

- **数据抽取(Data extraction):**
  - get data from multiple, heterogeneous, and external sources
- **数据清洗(Data cleaning):**
  - detect errors in the data and rectify them when possible
- **数据转换(Data transformation):**
  - convert data from legacy or host format to warehouse format
- **数据装载(Load):**
  - sort, summarize, consolidate, compute views, check integrity, and build indicies and partitions
- **刷新(Refresh):**
  - propagate the updates from the data sources to the warehouse

Data Warehousing and OLAP Technology

- **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**

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 Cubes

item	all
region	all

Selfexp: Background color; Inexp: Margin;
Pathexp:Which path will lead to an exception?

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%

Data Warehousing and OLAP Technology

- 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

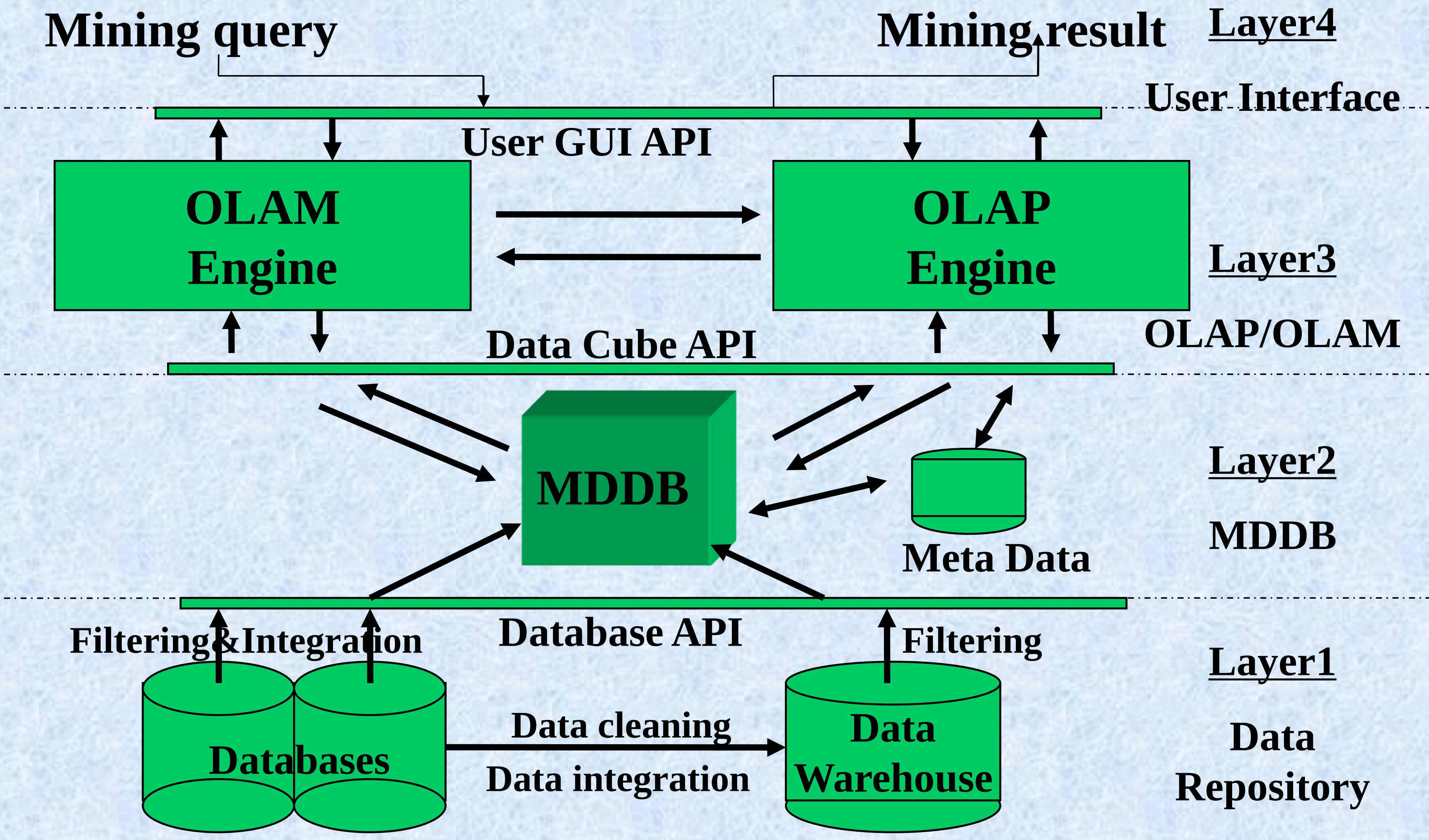
Data Warehouse Usage

- **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**

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

An OLAM Architecture



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**
- **Data warehouse implementation**
- **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)

References (I)

References

- Jiawei Han and Micheline Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.(Including Course Materials)
- 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

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.
- 林宇等编著,《数据仓库：原理与实践》,人民邮电出版社

Thank you !!!