Data Warehousing

Past Business Scenerio

- No Web for Business
  - Customers appear ”physically” in the store
  - Customers do not change to other stores more easily

Today’s Business

- The Web made Business Intelligence (BI) more necessary
  - Customers do not appear ”physically” in the store
  - Customers can change to other stores more easily
- Thus,
  - Can know customers using data and BI
  - Web logs makes is possible to analyze customer behavior in a more detailed than before (what was not bought?)
  - Combine web data with traditional customer data
- Wireless Internet adds further to this
  - Customers are always ”online”
  - Customer’s position is known
  - Combine position and customer knowledge => very valuable

Data Warehousing

- On October 11, 2000, find the 5 top-selling products for each product subcategory that contributes more than 20% of the sales within its product category?
- Regular database models and systems are not suitable for this type of queries

Data Warehousing

- Learn how to design, build, and use a data warehouse
- Multidimensional modeling
  - Dimensions, facts, measures

Why?

- Data is a valuable asset for ERP, etc.
- Data is available from every step of a sales pipeline, from
  - Internal Data Sources
    - ERP systems, Sales/Financials, Support/CRM, Marketing
  - External Data Sources
    - Social networks, Clickstreams, Websites, Supply chain/Logistics (Through customer support)

Why?

- Data is available
  - As increase in software as a service (SaaS) products contributes to the data
  - In 2016, the use of cloud services for business processes have accelerated past current forecasts by 30%
- But, more data does not necessarily translate into better analytics
- But, it difficult to build meaningful analytics with various data sources

Why?

- Key Problems of Business

1) **Complex and unusable models**
    Many DB models are difficult to understand
    DB models do not focus on a single clear business purpose

2) **Same data found in many different systems**
    Example: customer data in 14 systems
    The same concept is defined differently

3) **Data is suited for operational systems**
    Just for Accounting, billing, etc., Do not support analysis across business functions

4) **Data quality is bad**
    Missing data, imprecise data, different use of systems

5) **Data are ”volatile”**
    Data deleted in operational systems (6 months), but Data change over time – no historical information

Why?

- To give full picture, analytics must require more than one data source
- Centralizing data provides the structure necessary to enable cross-platform analytics
- To identify an anomaly before it had a long-term negative impact

Enterprise Organizations Challenge

- Collect diverse kinds of data and maintain large databases from
  - Multiple, Heterogeneous and Distributed information sources
- Challenge
  - To integrate such data to provide easy and efficient access to it

To answer…

- Analyzing operations, to
  - Increase the customer focus
    - By the buying patterns of preference, time
  - Look for source of profit or to manage product portfolio
    - By comparing the performance of sales by quarter, year, geographic regions

Solution

- **New analysis environment**
- where data is
  - Integrated (logically and physically)
  - Subject oriented (versus function oriented)
  - Supporting management decisions (different organization)
  - Stable (data not deleted, several versions)
  - Time variant (data can always be related to time)

Earlier Approach

- Traditional heterogeneous DB integration
- Query based approach: data is integrated based on query

Data Warehouse vs. Heterogeneous DBMS

- Traditional heterogeneous DB integration
  - A query driven approach
    - Data is integrated based on query
    - Build wrappers/mediators on top of heterogeneous databases
    - 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
    - Advantages
      - Access to most up-to-date data Different interfaces

Data Warehouse vs. Heterogeneous DBMS

- Traditional heterogeneous DB integration
  - A query driven approach
  - Problems
    - Complex information filtering and integration process
    - Competes for resources at local sources
    - Inefficient and Expensive for frequent queries especially for queries requiring aggregations
    - Delay in query processing
      - Slow or unavailable information sources
      - Complex filtering and integration
    - Data loss at the sources (e.g. historical data cannot be recovered

Data Warehouse vs. Heterogeneous DBMS

- Data warehouse
  - Alternative to the traditional approach of heterogeneous database integration
  - Update-driven
    - Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis
  - High performance
    - As data are copied, preprocessed, integrated, summarized and restructured into one semantic data store
  - Support multidimensional queries from historical information

Operational DBMS vs. Data Warehouse

- OLTP (On-Line Transaction Processing)
  - Major task of traditional relational DBMS
    - Day-to-day operations
      - Purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- Data Warehouse Systems
  - Known as OLAP (On-line Analytical Processing)

Operational DBMS vs. Data Warehouse

- User and system orientation
  - OLTP – Customer oriented
    - Transaction and query processed by clerks and clients
  - OLAP – Market oriented
    - Data analysis by knowledge workers – managers, executives and analysts
- Data contents for easy decision making
  - OLTP – Current data (Too detailed)
  - OLAP – Large amount of historical data
    - To provide summarization and aggregation
    - Stores and manages information at different level of granularity
- Database design
  - OLTP – ER Model and application oriented database design
  - OLAP – Star or Snowflake model and subject oriented database design

Operational DBMS vs. Data Warehouse

- View
  - OLTP – Current data within an enterprise or department
  - OLAP – Spans the multiple versions of a database schema due to evolutionary process of an organization and integrated from many data stores
    - Stored on multiple storage media
- Access patterns
  - OLTP – Consist of short, atomic transactions - Update
  - OLAP – Consist of MOSTLY read-only 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

Goals

- How data centralization enables better analytics
- How to redefine data for proper usage
- How the right BI tool eliminates the data analyst bottleneck
- How to define single sources of truth for your organization
- How to build a data-driven (not just data-rich) organization

Data Warehousing

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

Data Warehousing

- The standard model to aggregate data and provide business-directed analytics, for nearly 30 years
  - Data is extracted from various sources, transformed to a predefined model, and loaded into the data warehouse
  - This extract, transform, and load (ETL) process results in query able analytics contextualized by key dimensions (e.g., customer, product, location)
- But this process is slow and leads to latencies in the data warehouse
- Stale data (even just a week or two old) can be useless data for many purposes

Data Warehousing

- The Argument for Data Centralization
  - For disparate data sources to be compared, they must contain common fields that can be mapped or linked
  - Evaluate each data source for existing common fields and, if you can, resolve minor variances (for example, region vs. state vs. zip code)
  - Standardize data references, though some tools will allow you to specify relationships without needing to unify labels (e.g., product_id vs. product_number)

What is Data Warehouse?

- Defined in many different ways
  - 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 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 - Data Integration

- Why this Problem?
  - Different interfaces
  - Different data representations
  - Duplicated information
  - Inconsistent information
- Solution
  - Collect and combine information
  - Provide an integrated view
  - Provide a uniform user interface
  - Support sharing of data

Data Integration-Approach 1

- Query based approach: data is integrated based on query
- Advantages
  - Access to most up-to-date data Different interfaces
- Disadvantages
  - Delay in query processing
    - Slow or unavailable information sources
    - Complex filtering and integration
  - Inefficient and expensive for frequent queries
  - Competes with local processing at sources
  - Data loss at the sources (e.g. historical data cannot be recovered

Data Integration-Approach 2

- Data Warehouse approach: data is integrated in advance
- Data is stored in DW for querying and analysis
- Advantages
  - High query performance
  - Does not interfere with local processing at sources
    - Assumes that data warehouse update is possible during downtime of local processing
    - Complex queries are run at the data warehouse
    - OLTP queries are run at the source systems
- Disadvantages
  - The most current source data is not available

Data Warehouse—Integrated

- Constructed by integrating multiple, heterogeneous data sources, like
  - Relational databases, flat files, on-line transaction records
- Data cleaning and data integration techniques applied, to ensure consistency in among different data sources
  - Naming conventions
  - Encoding structures
  - Attribute measures, etc.
    - 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

Data Warehouse—Nonvolatile

- A physically separate store of data transformed from the operational environment
- A data warehouse
  - 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

- Different functions and different data:
  - <u>Missing data</u>: Decision support (DS) requires historical data which operational DBs do not typically maintain
  - <u>Data consolidation</u>: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
  - <u>Data quality</u>: Different sources typically use inconsistent data representations, codes and formats which have to be reconciled
- Nowadays, There are more and more systems which perform OLAP analysis directly on relational databases

Data Warehousing

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- 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

- Data cube is defined by dimensions and facts
  - Dimensions
    - A data cube allows data to be modeled and viewed in multiple dimensions
    - Dimensions are the entities with respect to which an organization wants to keep records
    - e.g. , For the data cube sales, the dimensions are time, item, branch, location that allows to keep track of things like monthly sales of items and the branches and locations at which the items were sold

From Tables and Spreadsheets to Data Cubes

- Dimensions
  - Each dimension may have a table associated with it, called a Dimension table
  - Which further describes the dimension, e.g.
    - A dimension table item contains attributes (item_name, brand, type)
    - A dimension table time contains attributes (day, week, month, quarter, year)
  - Dimension tables can be specified by the users or experts or automatically generated and adjusted based on data distribution

From Tables and Spreadsheets to Data Cubes

- Facts
  - Represented with Fact table that contains the name of the facts, or measures, as well as keys to each of the related dimension tables
  - Numerical measures to analyze the relationship between dimensions, such as
    - Dollars_sold (Sales amount in dollars)
    - Units_sold (Number of units sold)

Data Cubes

- Cross Tabulations
  - Useful for the historical data
  - Sale of Electronics Item during June

Month/Product	TV	PC	VCR	Total
June	5	10	2	17
July	6	10	1	17
August	5	9	1	15
Total	16	29	4	39

Data Cubes

- Cross Tabulations
  - Useful for the historical data
  - Sale of Electronics Item during June

Dimensions

Month/Product	TV	PC	VCR	Total
June	5	10	2	17
July	6	10	1	17
August	5	9	1	15
Total	16	29	4	39

Aggregated w. r. t. X axis

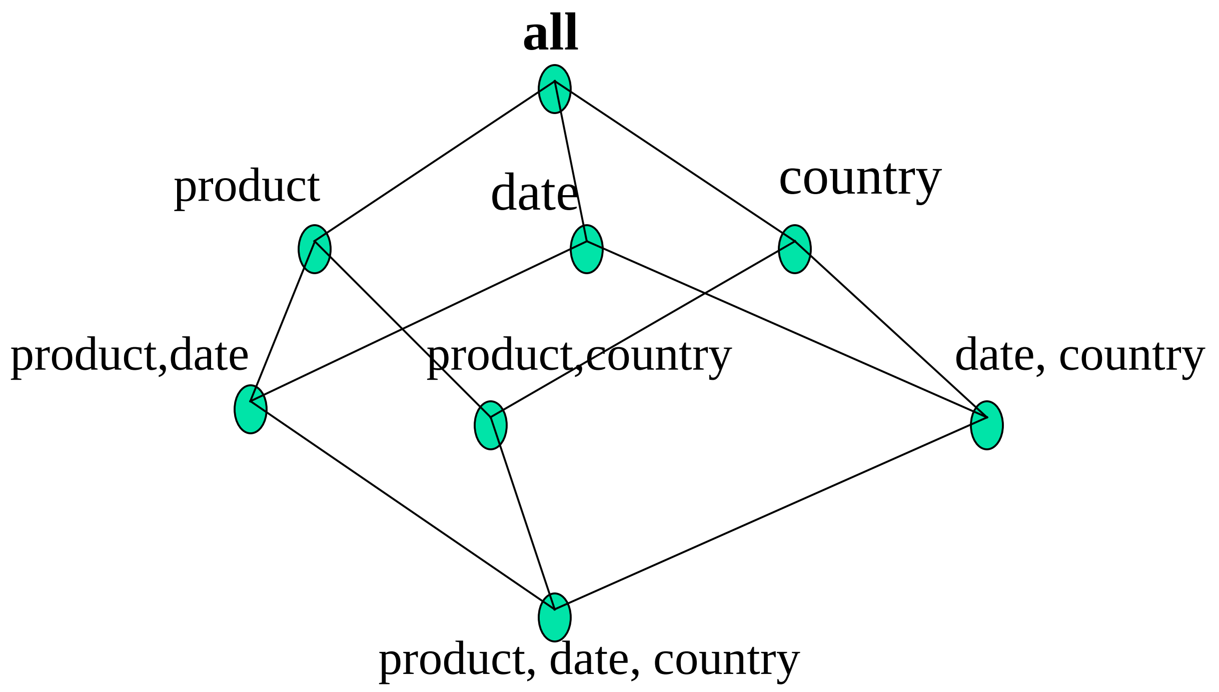
Aggregated w. r. t. Y axis

Aggregated w. r. t. X & Y axis

From Tables and Spreadsheets to Data Cubes

- Data cube
  - 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
    - Denoted by ALL
  - The lattice of cuboids forms a data cube
    - Each showing the data at a different level of summarization or group by

Cuboids Corresponding to the Cube



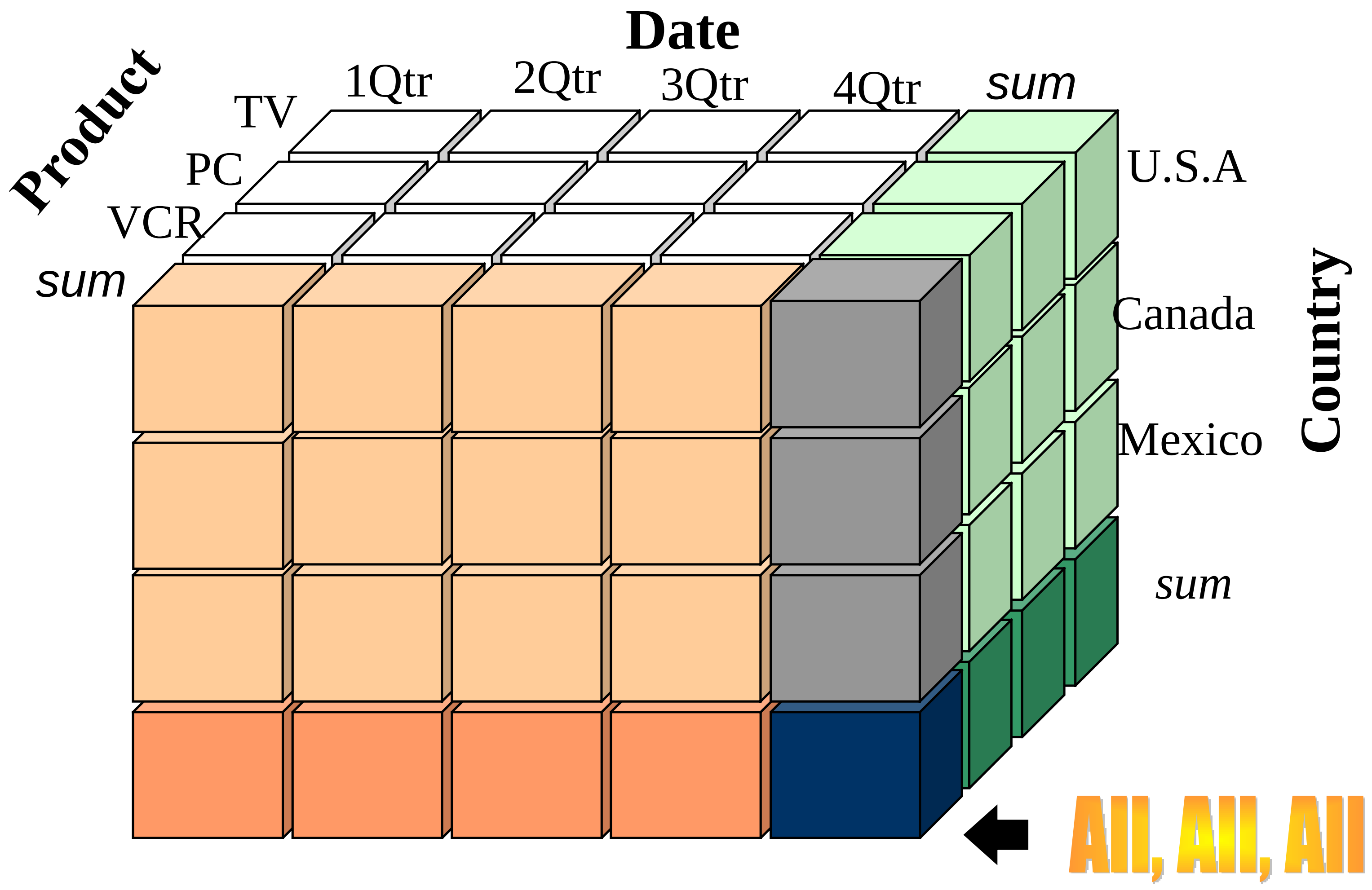
0-D(apex) cuboid

1-D cuboids

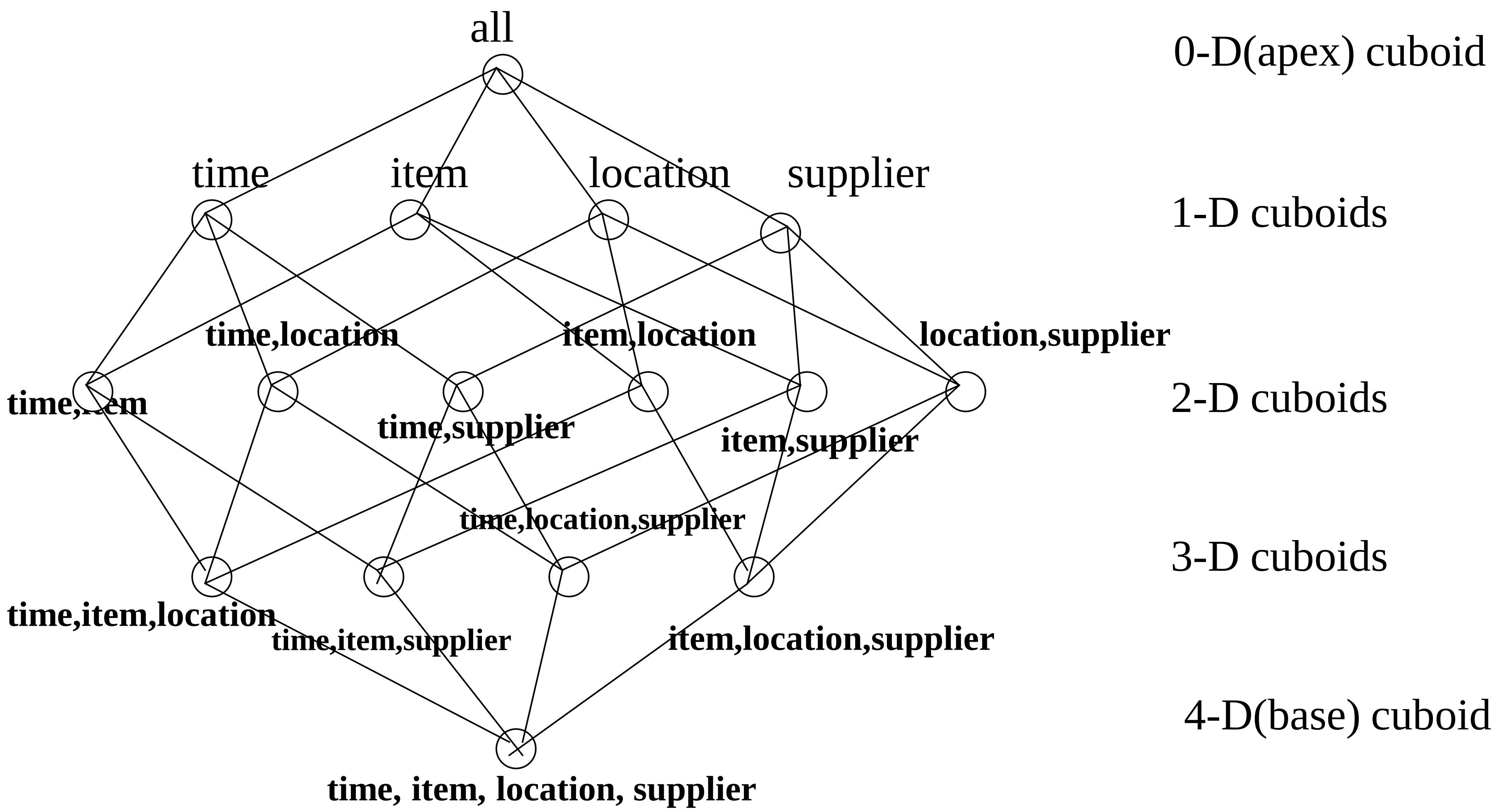
2-D cuboids

3-D(base) cuboid

Cube


Date
1Qtr
2Qtr
3Qtr
4Qtr
sum
Product
TV
PC
VCR
sum
U.S.A
Canada
Mexico
sum
Country
All, All, All

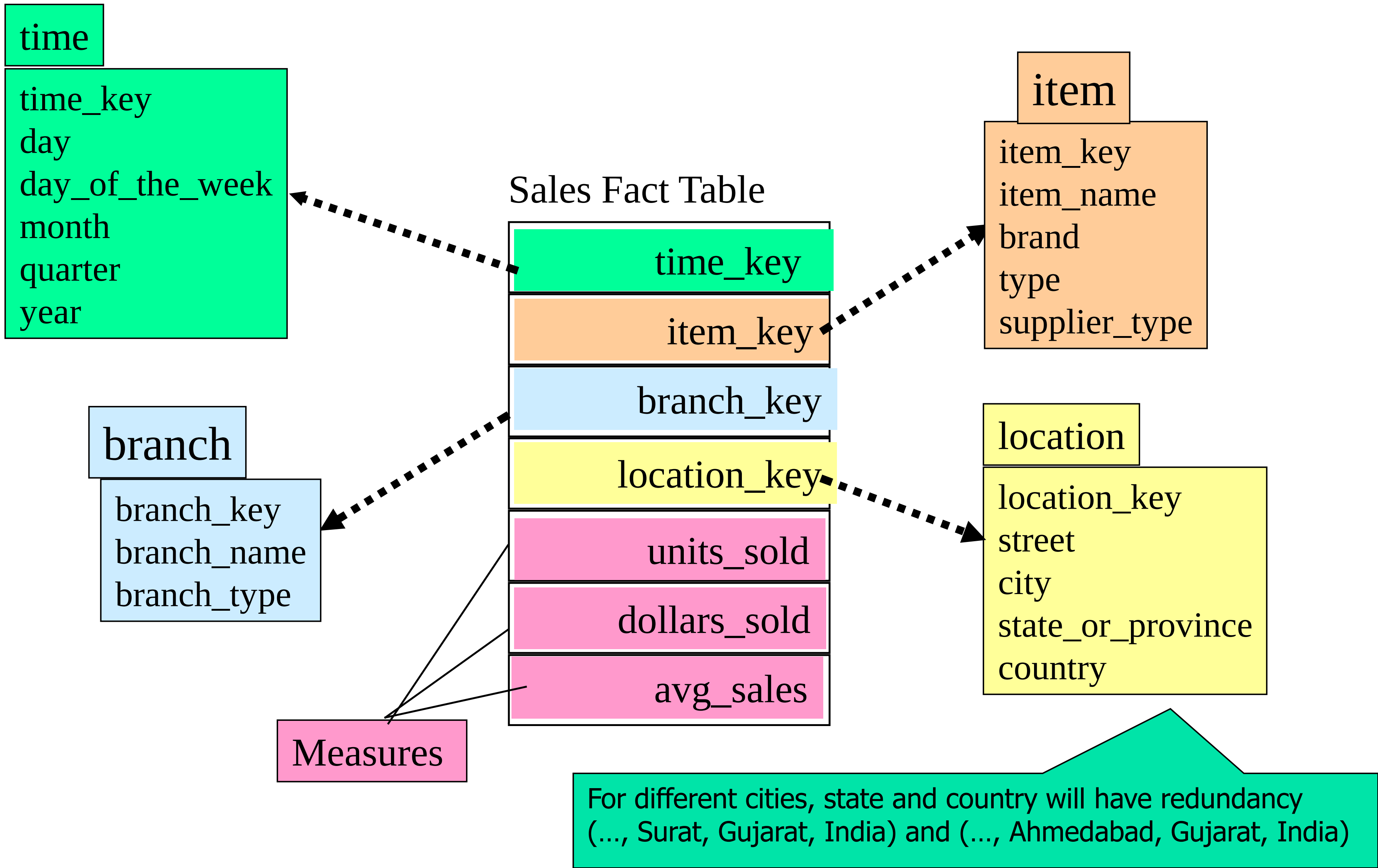
Another Cube: A Lattice of Cuboids



Conceptual Modeling of Data Warehouses

- Star schema
  - Most common schema
  - A fact table is connected to a set of dimension tables
    - A large central table (fact table)
      - Containing the bulk of the data, with no redundancy
    - Dimension tables
      - Set of smaller attendant tables, one for each dimension
- Snowflake schema
- Fact constellations

Example of Star Schema



For different cities, state and country will have redundancy (…, Surat, Gujarat, India) and (…, Ahmedabad, Gujarat, India)

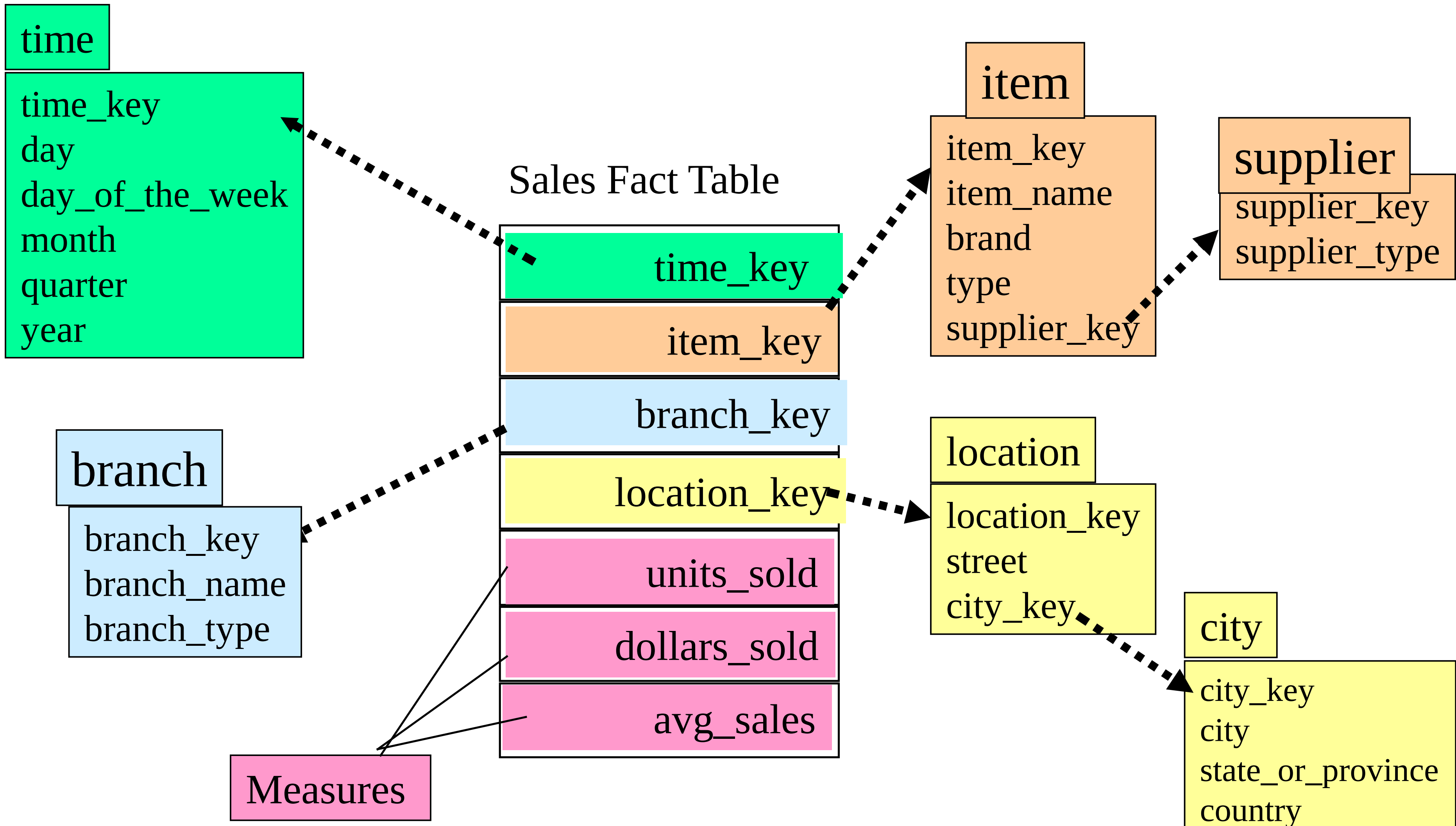
Characteristics of Star Schema

- Every dimension is represented with the only one-dimension table
- Dimension table should contain the set of attributes
- Dimension table is joined to the fact table using a foreign key
- Dimension table are not joined to each other
- Fact table would contain key and measure
- Easy to understand and provides optimal disk usage
- Dimension tables are **not normalized**

Conceptual Modeling of Data Warehouses

- Star schema
- 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
    - To reduce redundancies
      - Such a table is easy to maintain and saves storage space
- Fact constellations

Example of Snowflake Schema



Conceptual Modeling of Data Warehouses

- Star schema
- 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
    - To reduce redundancies
      - Such a table is easy to maintain and saves storage space
    - **But, Reduce the effectiveness of browsing, since more joins will be needed to execute a query**
    - **So, system performance will be affected**
    - **Therefore, not popular**
- Fact constellations

Snowflake Schema

Characteristics

- Uses smaller disk space
- Easier to implement a dimension is added to the Schema
- Due to multiple tables query performance is reduced

Problem

- More maintenance efforts because of the more lookup tables

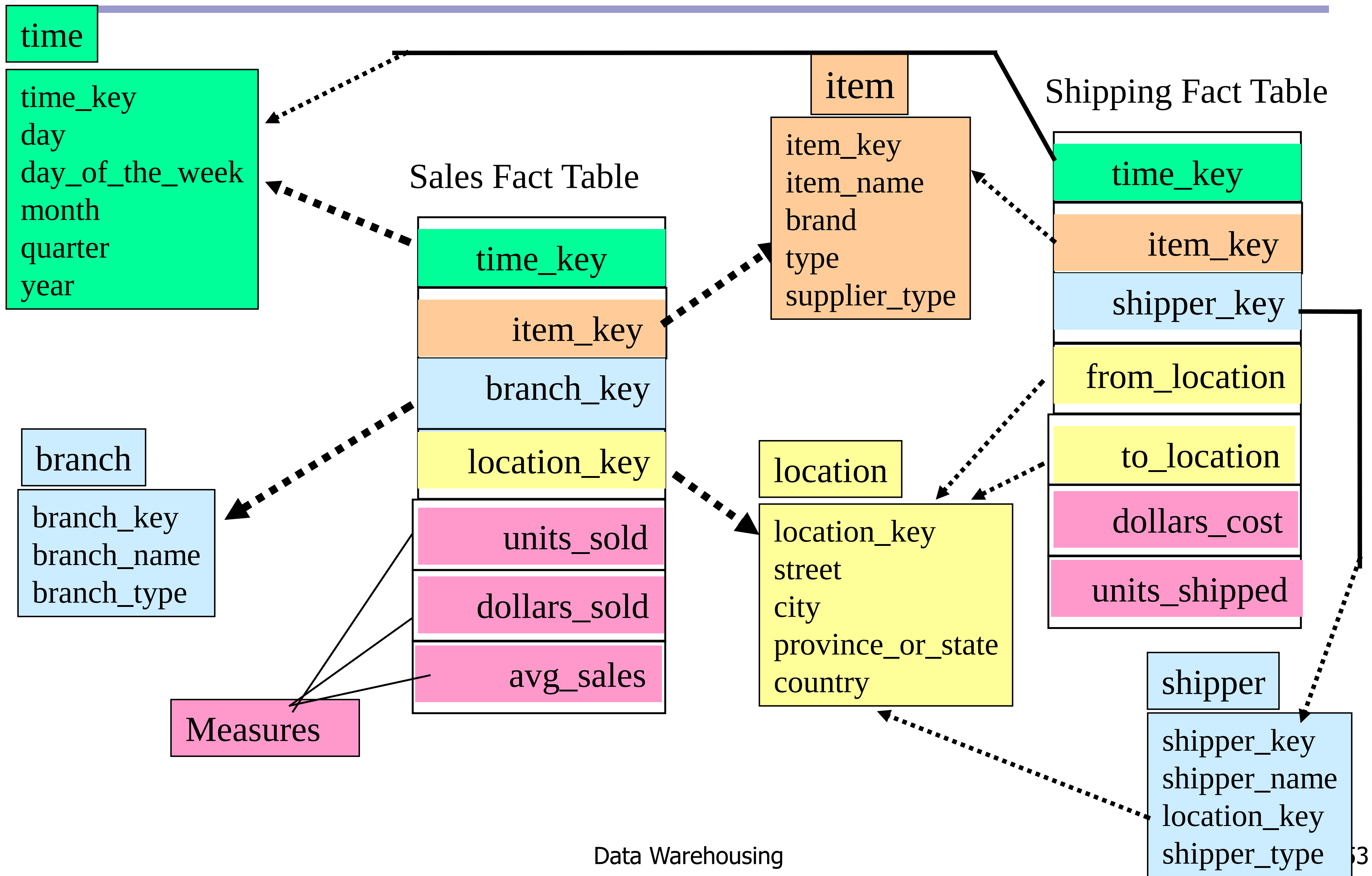
Star Schema vs Snowflake Schema

Star Schema	**Snow Flake Schema**
Hierarchies for the dimensions are stored in the dimensional table.	Hierarchies are divided into separate tables.
Contains a fact table surrounded by dimension tables.	One fact table surrounded by dimension table which are in turn surrounded by dimension table
Only single join creates the relationship between the fact table and any dimension tables.	Requires many joins to fetch the data.
Simple DB Design.	Very Complex DB Design.
Denormalized Data structure and query also run faster.	Normalized Data Structure.
High level of Data redundancy	Very low-level data redundancy
Cube processing is faster.	Cube processing might be slow because of the complex join.
Offers higher performing queries using Star Join Query Optimization.	The Snow Flake Schema is represented by centralized fact table which unlikely connected with multiple dimensions. Compare to Star poor performance.

Conceptual Modeling of Data Warehouses

- Star schema
- Snowflake schema
- Fact constellations
  - Sophisticated applications may require Multiple fact tables to share dimension tables
  - Viewed as a collection of stars, therefore called galaxy schema or fact constellation

Example of Fact Constellation



Schema Selection

- Data Warehouse spans the entire organization, such as customers, items, sales, assets and personnel
  - Scope is enterprise-wide
  - Fact-constellation schema is commonly used
    - As it can model multiple, interrelated subjects

- Data Mart is a department subset of the data warehouse that focuses on the selected subjects
  - Scope is department-wide
  - Star or Snowflake schema are commonly used
    - As both modeling single subjects

Cube Definition Syntax (BNF) in 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 Star Schema in 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)
```

Defining Snowflake Schema in 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)**)

Defining 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
```

Measures of Data Cube

- How to compute measures?
  - A numerical function that can be evaluated at each point in the data cube space
  - By aggregating the data corresponding to the respective dimension-value pairs
  - Three categories

Measures of Data Cube: Three Categories

- Distributive: If the result derived by applying the function to *n partitions* 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 applying an algebraic function to one or more distributive measures
  - E.g., avg() → sum()/count() which are distributive aggregate functions
    - Here, sum() and count() are typically saved in precomputation, so the derivation of avg() of data cube is straightforward
  - Weighted arithmetic mean

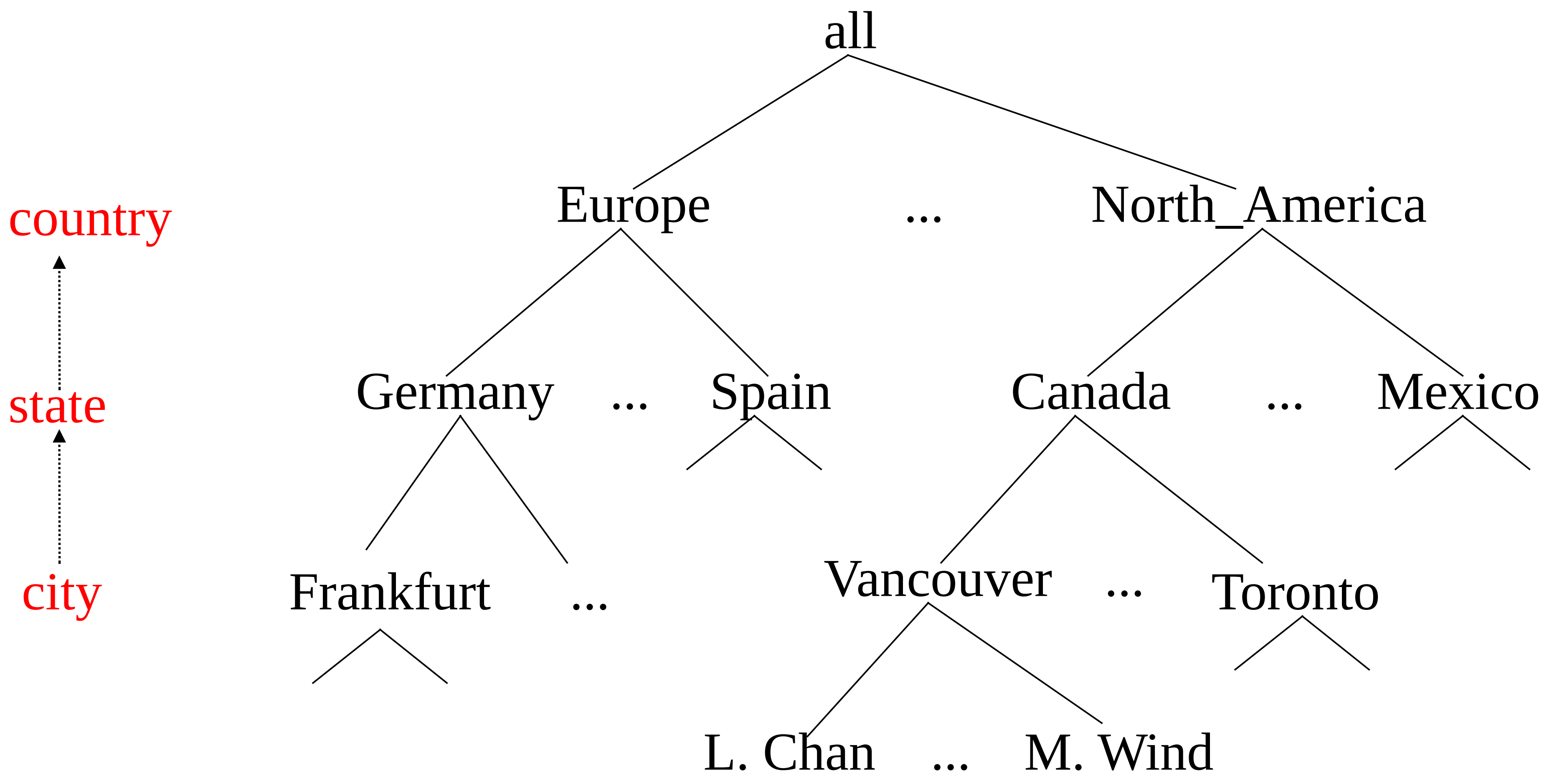
$$\bar{x} = \frac{\sum_{i=1}^{N} w_i x_i}{\sum_{i=1}^{N} w_i} = \frac{w_1 x_1 + w_2 x_2 + \cdots + w_N x_N}{w_1 + w_2 + \cdots + w_N}$$

- Holistic: A measure that must be computed on the entire data set as a whole
  - Cannot be computed by partitioning the given data into subsets and merging the values obtained for the measure in each subset
  - E.g., median(), mode(), rank()

Concept Hierarchy

- Defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts
- For example, concept hierarchy for the dimension location, mapping a set of low-level concepts (i.e. cities) to higher-level, more general concepts (i.e. countries)
- Allows the data to be handled at varying levels of abstraction

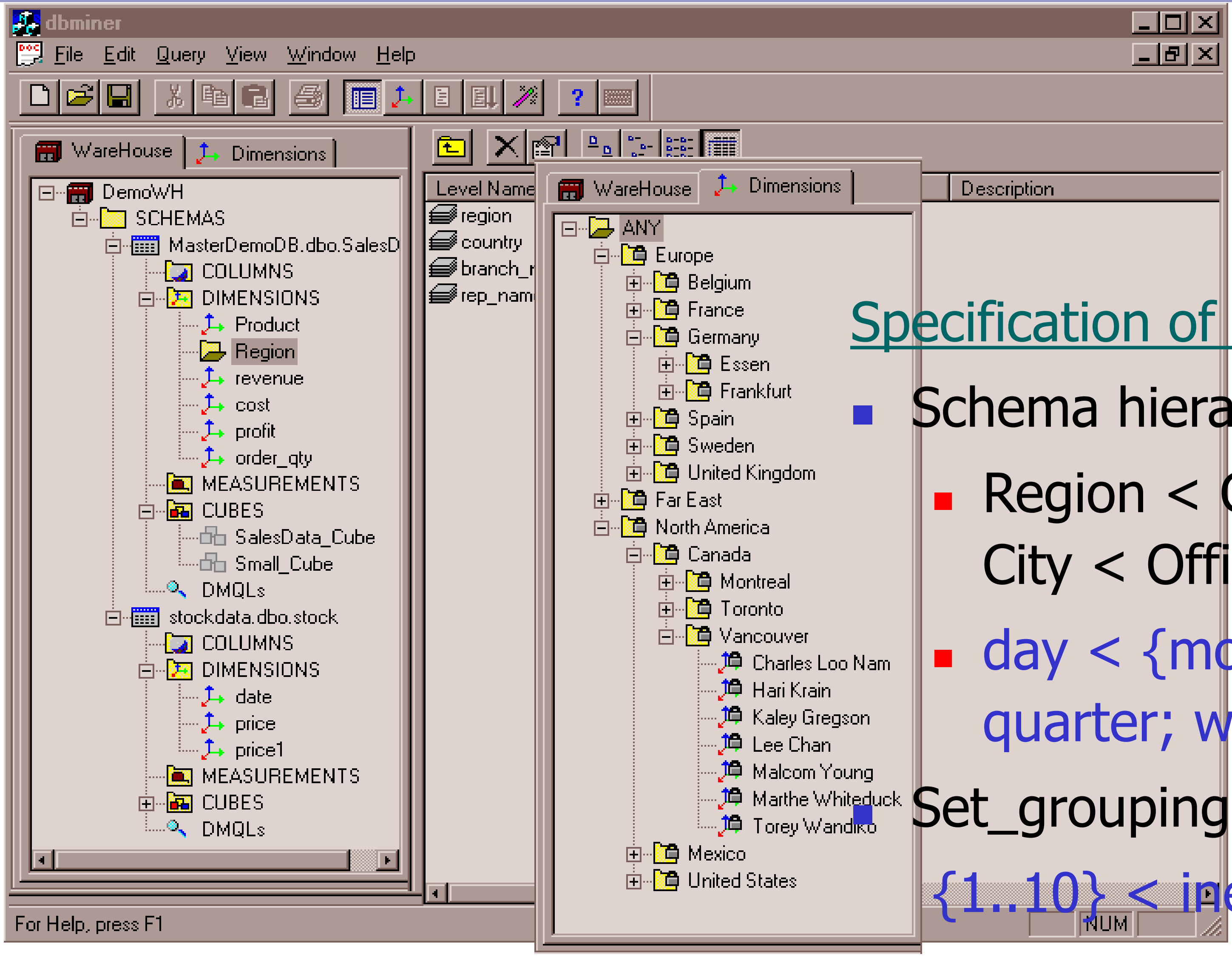
A Concept Hierarchy: Dimension (location)



Concept Hierarchy

- Also defined by discretizing or grouping values for a given dimension or attribute, resulting in a set-grouping hierarchy
  - A total or partial order can be defined among groups of values
  - For example, set-grouping hierarchy for the dimension price → an interval ($X…$Y] denotes the exclusive $X to inclusive $Y
- There may be more than one concept hierarchy for a given attribute or dimension based on different user viewpoints
  - For example, organize price by defining ranges for inexpensive, moderately_priced and expensive
- Concept hierarchy provided manually by system users, domain experts or knowledge engineers or may be automatically generated based on statistical analysis of the data distribution

View of Warehouses and Hierarchies

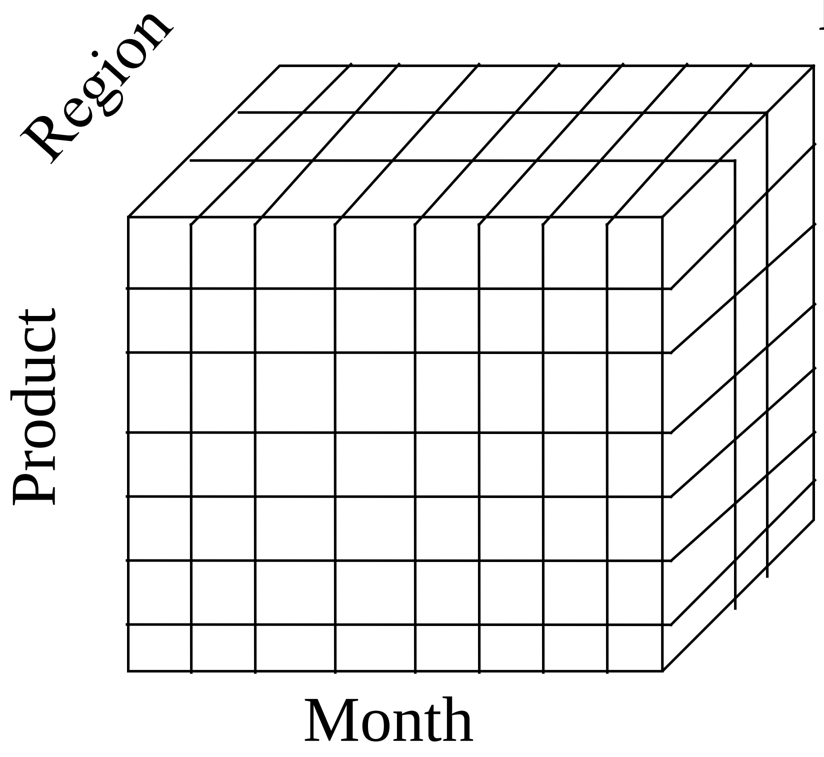


Specification of hierarchies

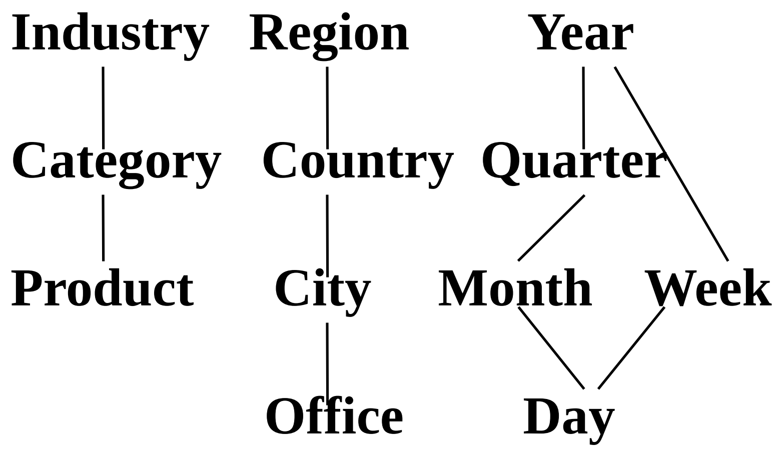
- Schema hierarchy
  - Region < Country < City < Office
  - day < {month < quarter; week} < year
- Set_grouping hierarchy
  - {1..10} < inexpensive

Sample Data Cube

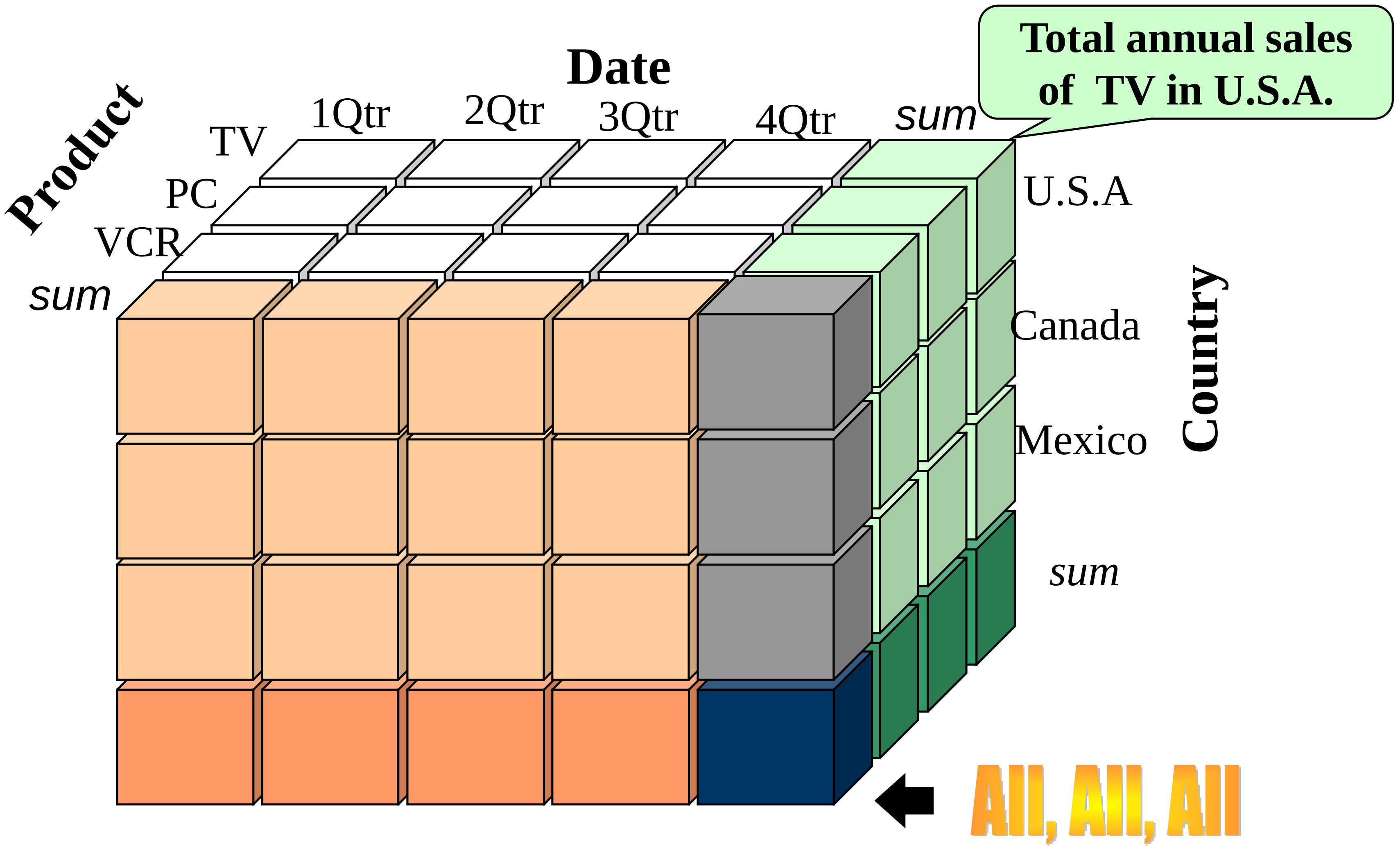
- Sales volume as a function of product, month, and region



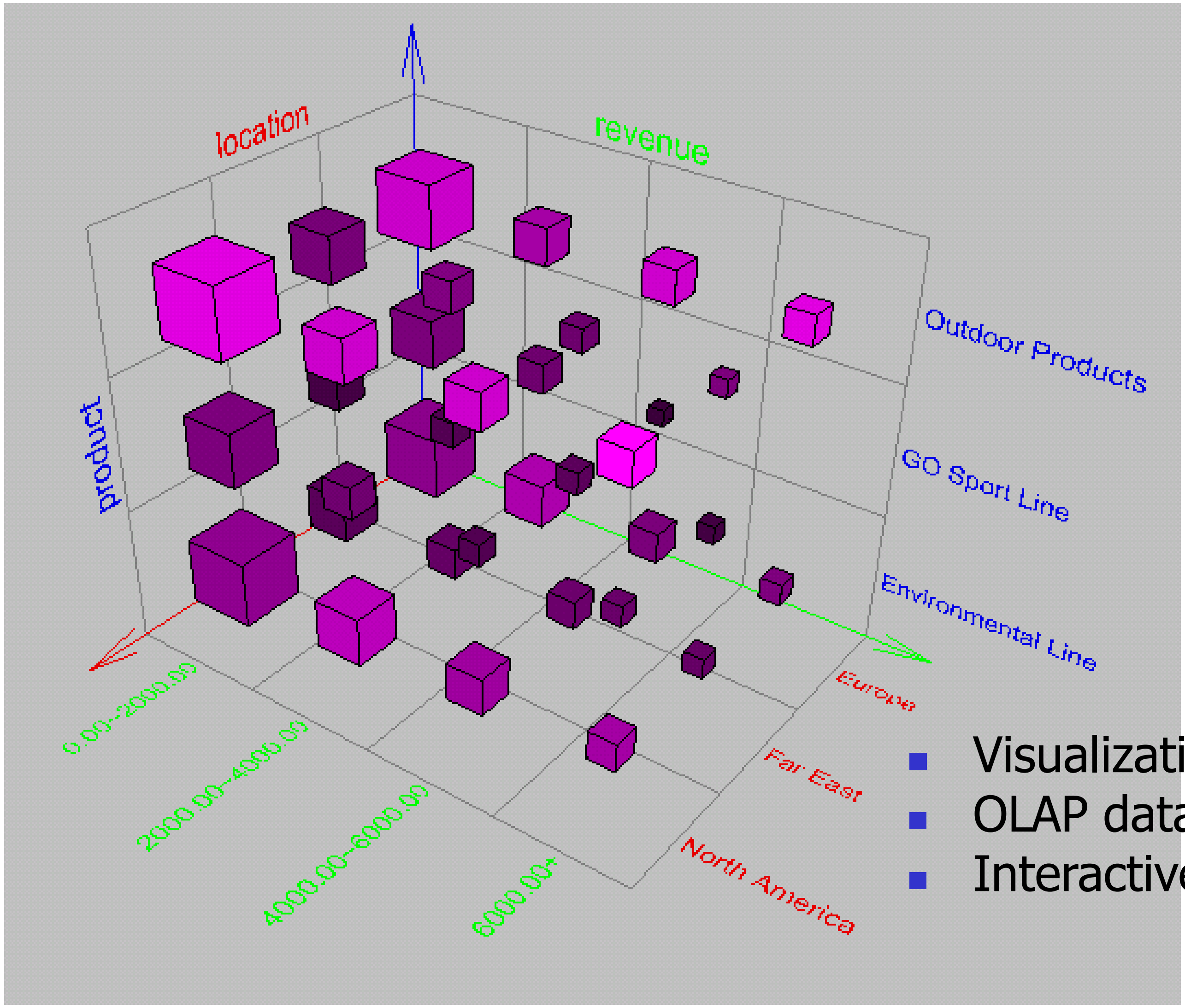
Dimensions: Product, Location, Time
Hierarchical summarization paths



A Sample Data Cube



Browsing a Data Cube



- Visualization
- OLAP data cube operations
- Interactive querying and analysis

Typical OLAP Operations

- Roll up (drill-up)
- Drill down (roll down)
- Slice and dice
- Pivot (rotate)
- Other operations
  - *Drill across*
  - *Drill through*

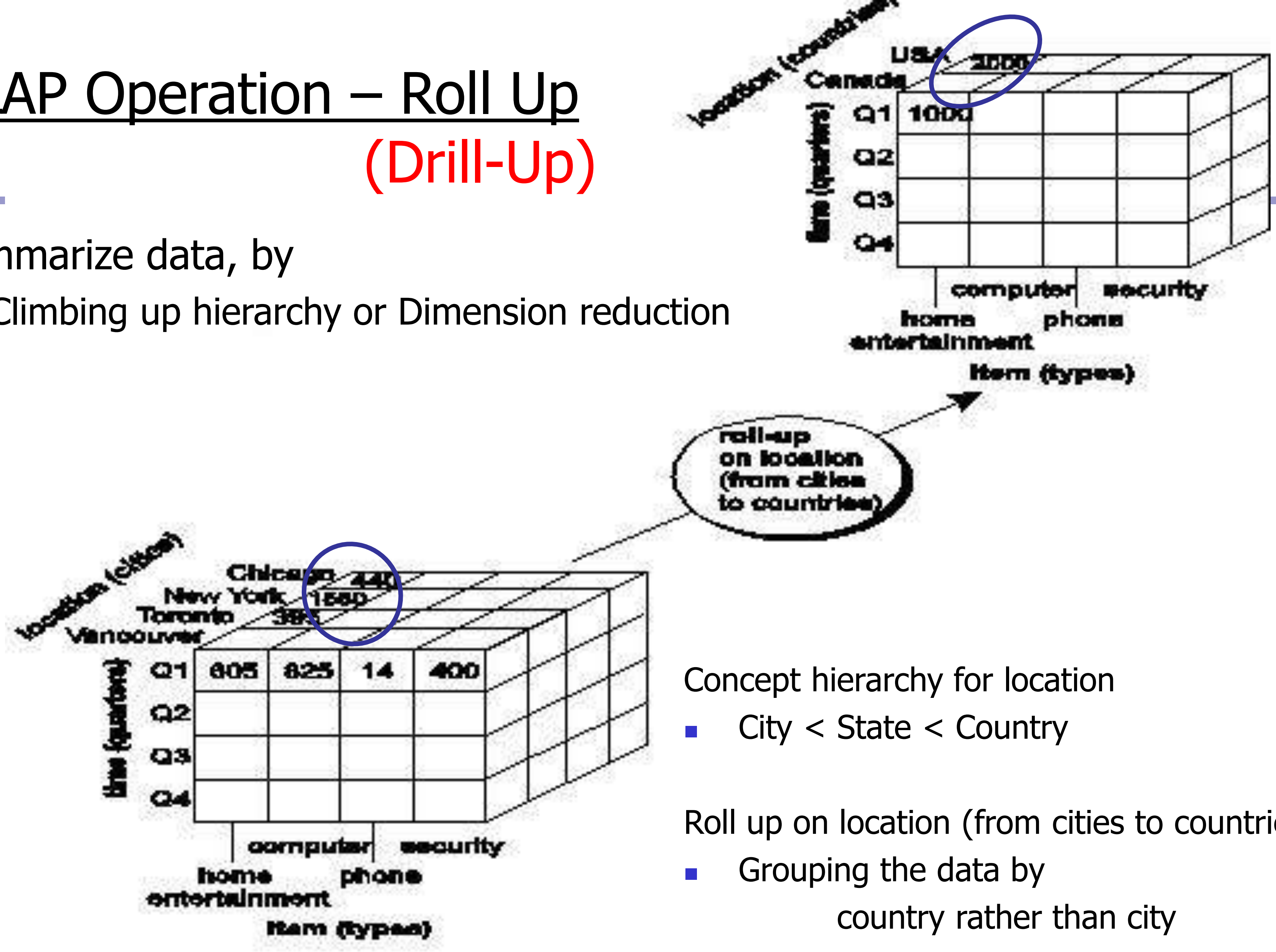
Typical OLAP Operations

- Roll up (Drill-up)
  - Summarize data
  - By climbing up hierarchy or by dimension reduction
  - If,
    - Concept hierarchy for location is City < State < Country
    - Roll up on location (from cities to countries)
      - Grouping the data by country rather than city

OLAP Operation – Roll Up (Drill-Up)

Summarize data, by

- Climbing up hierarchy or Dimension reduction



Concept hierarchy for location

- City < State < Country

Roll up on location (from cities to countries)

- Grouping the data by country rather than city

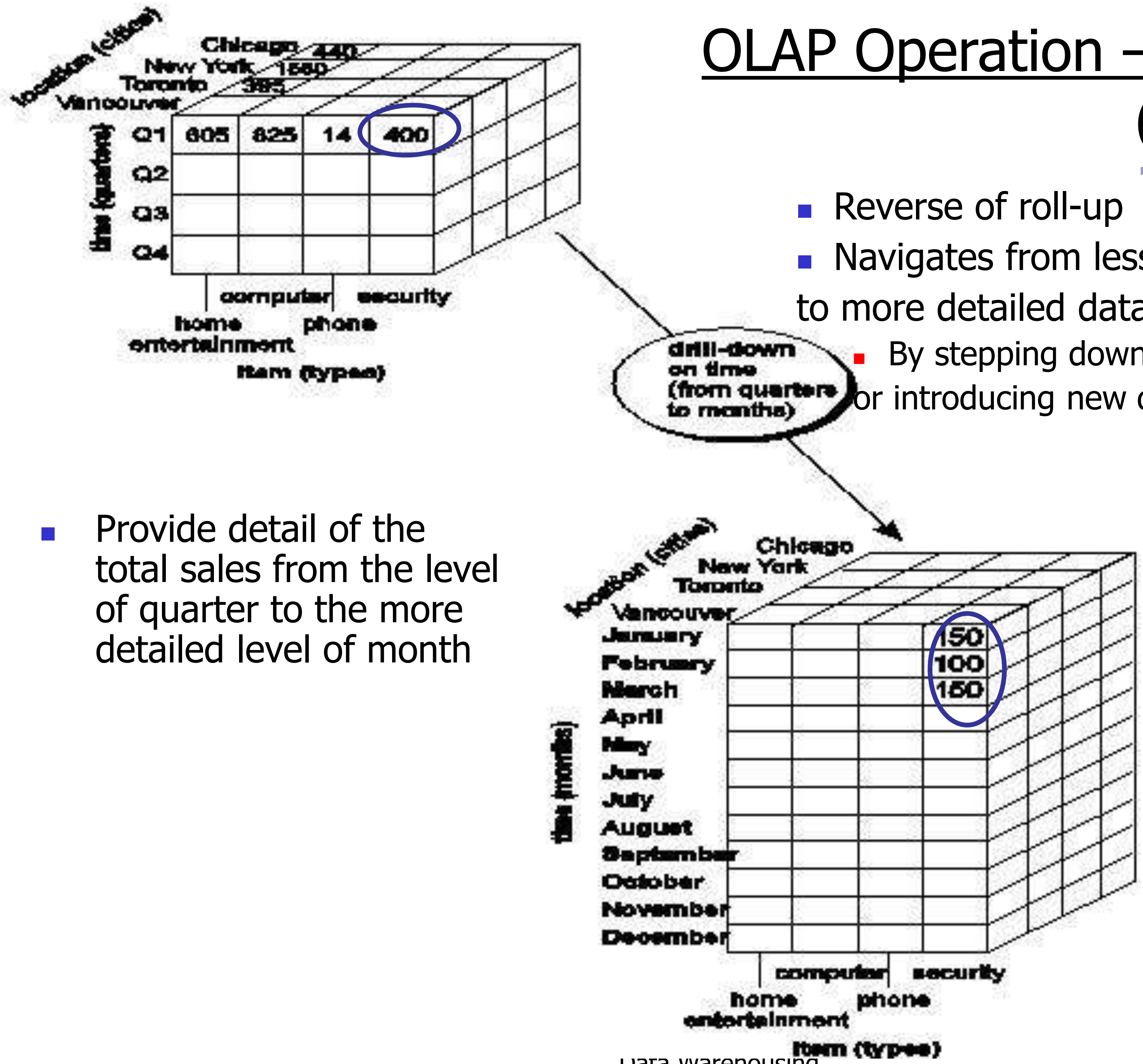
Typical OLAP Operations

- Drill Down
  - Reverse of roll-up
  - Navigates from less detailed data to more detailed data
  - By stepping down a concept hierarchy or introducing new dimensions

  - Provide detail of the total sales from the level of quarter to the more detailed level of month

OLAP Operation – Drill Down (Roll Down)

- Reverse of roll-up
- Navigates from less detailed data to more detailed data
  - By stepping down a concept hierarchy or introducing new dimensions



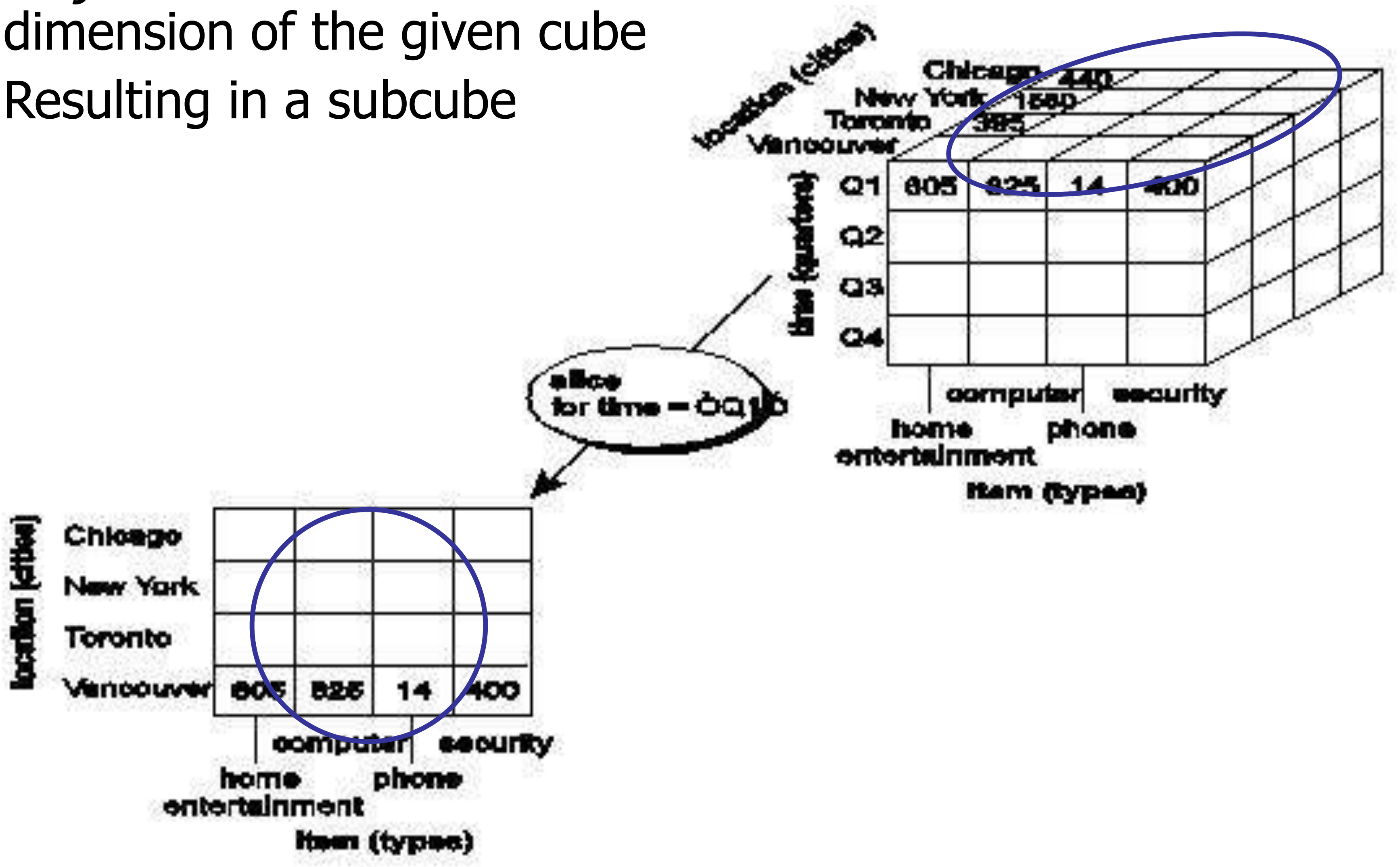
- Provide detail of the total sales from the level of quarter to the more detailed level of month

Typical OLAP Operations

- Slice
  - *Project and select* on one dimension of the given cube
  - Resulting in a subcube

OLAP Operation – Slice

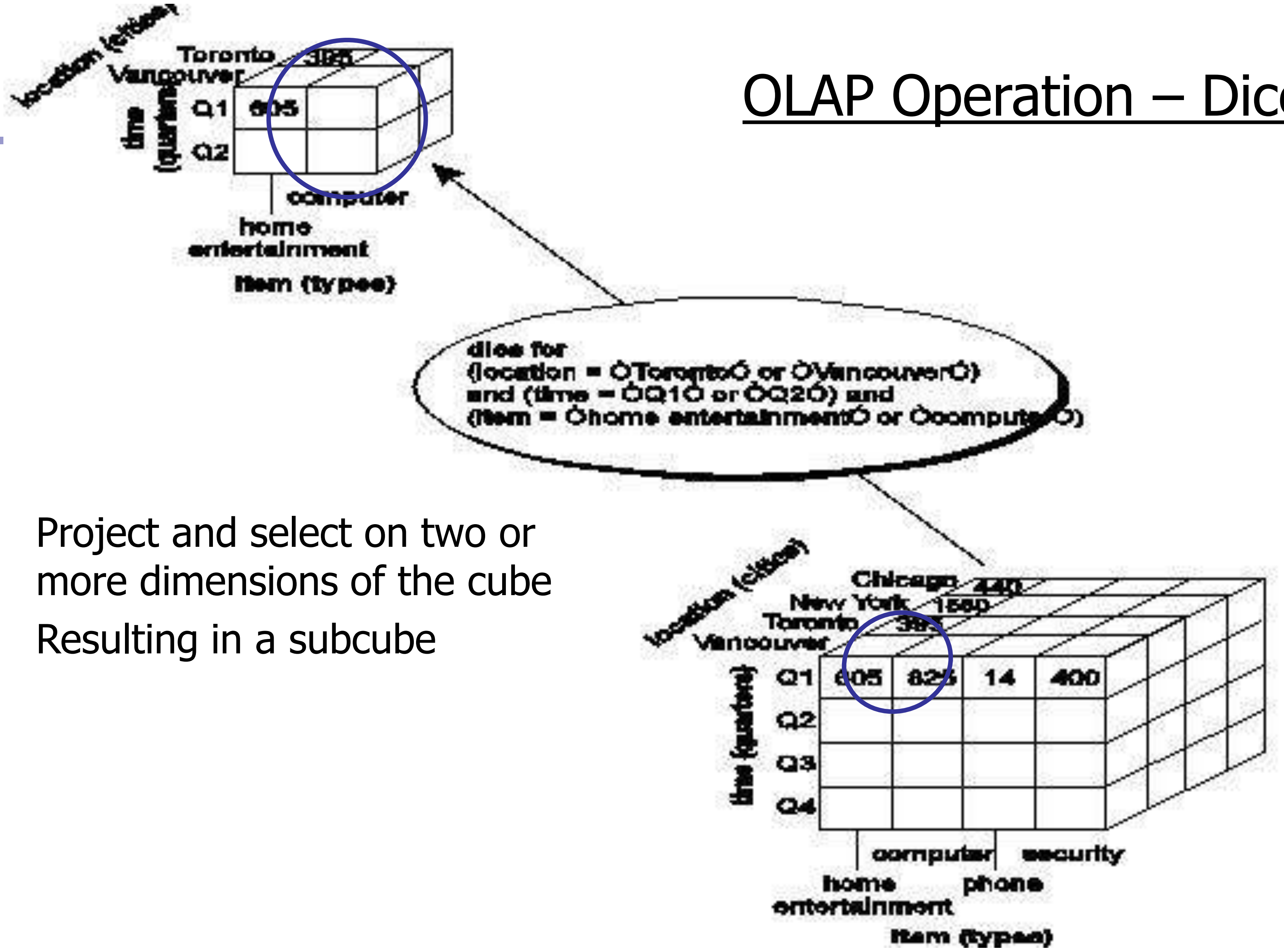
- *Project and select* on one dimension of the given cube
- Resulting in a subcube



Typical OLAP Operations

- Dice
  - Project and select on two or more dimensions of the cube
  - Resulting in a subcube

OLAP Operation – Dice

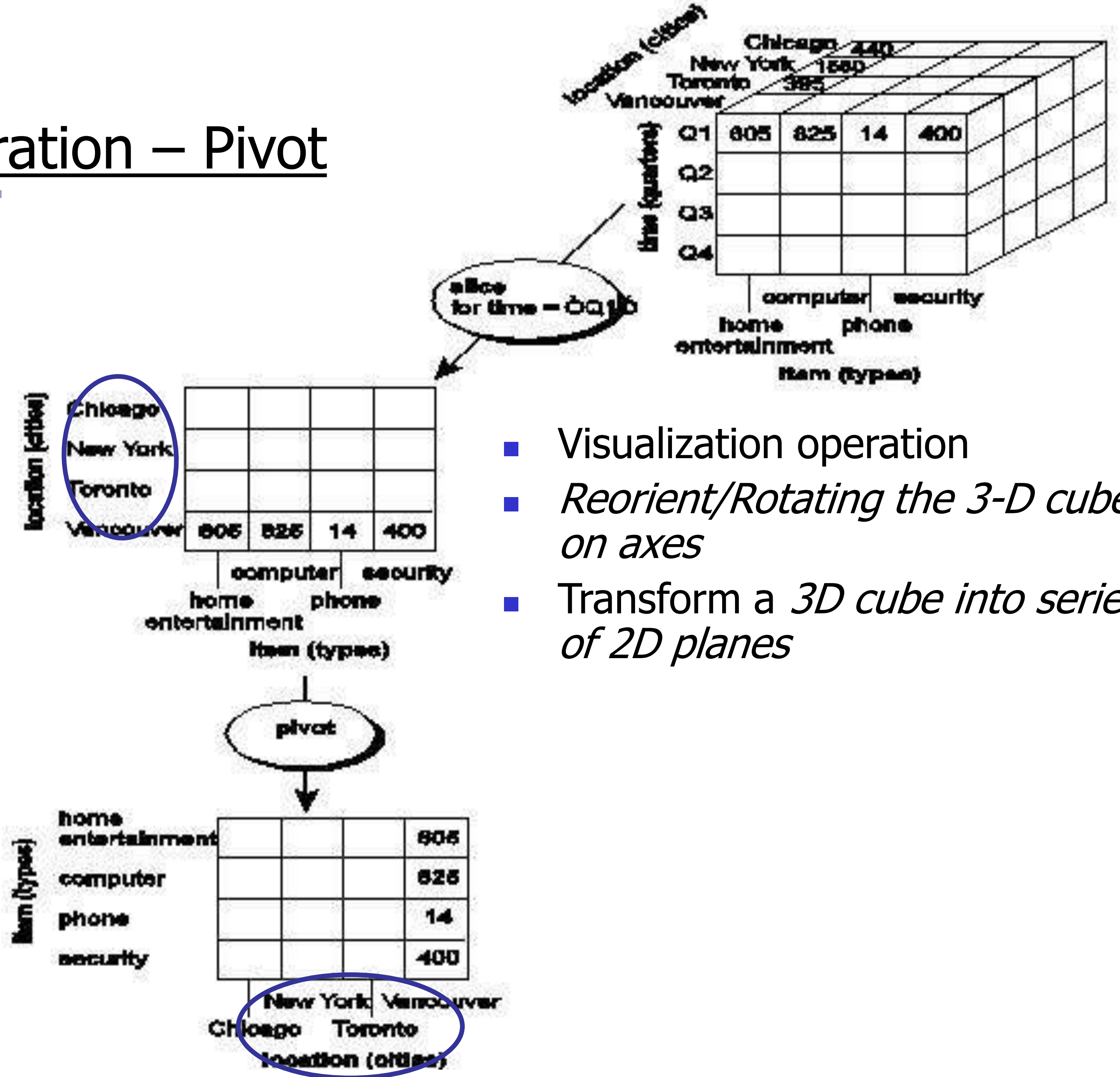


- Project and select on two or more dimensions of the cube
- Resulting in a subcube

Typical OLAP Operations

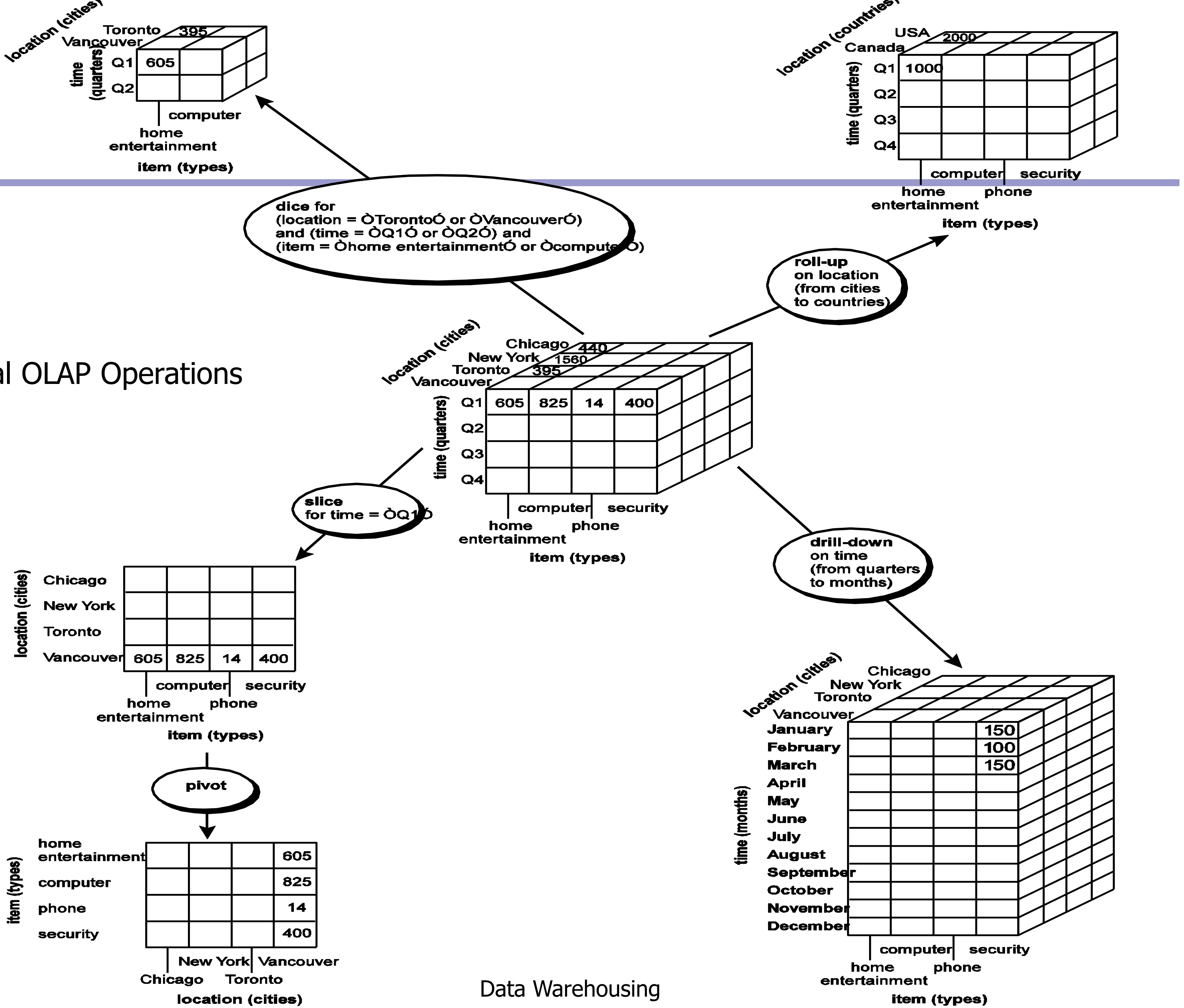
- Pivot (rotate):
  - Visualization operation
  - *Reorient/Rotating the 3-D cube on axes*
  - Transform a *3D cube into series of 2D planes*

OLAP Operation – Pivot



- Visualization operation
- *Reorient/Rotating the 3-D cube on axes*
- Transform a *3D cube into series of 2D planes*

Typical OLAP Operations



Typical OLAP Operations

- 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)*
  - *Ranking of top N or bottom N items*
  - *Computing averages*
  - *Growth rates*
  - *Interests*
  - *Depreciation*
  - *Currency conversions*
  - *Statistical functions…*

Aggregates

- Roll up on date

- Add up amounts for date=1

  SELECT sum(amt) FROM SALE WHERE date = 1

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

$\Rightarrow$ 81

Aggregates

- Add up amounts by day

SELECT date, sum(amt) FROM SALE GROUP BY date

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

$\Rightarrow$

ans	date	sum
	1	81
	2	48

Another Example

- Add up amounts by day, product

SELECT date, sum(amt) FROM SALE
GROUP BY date, prodId

sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

$\Rightarrow$

sale	prodId	date	amt
	p1	1	62
	p2	1	19
	p1	2	48

$\longrightarrow$ rollup $\longrightarrow$

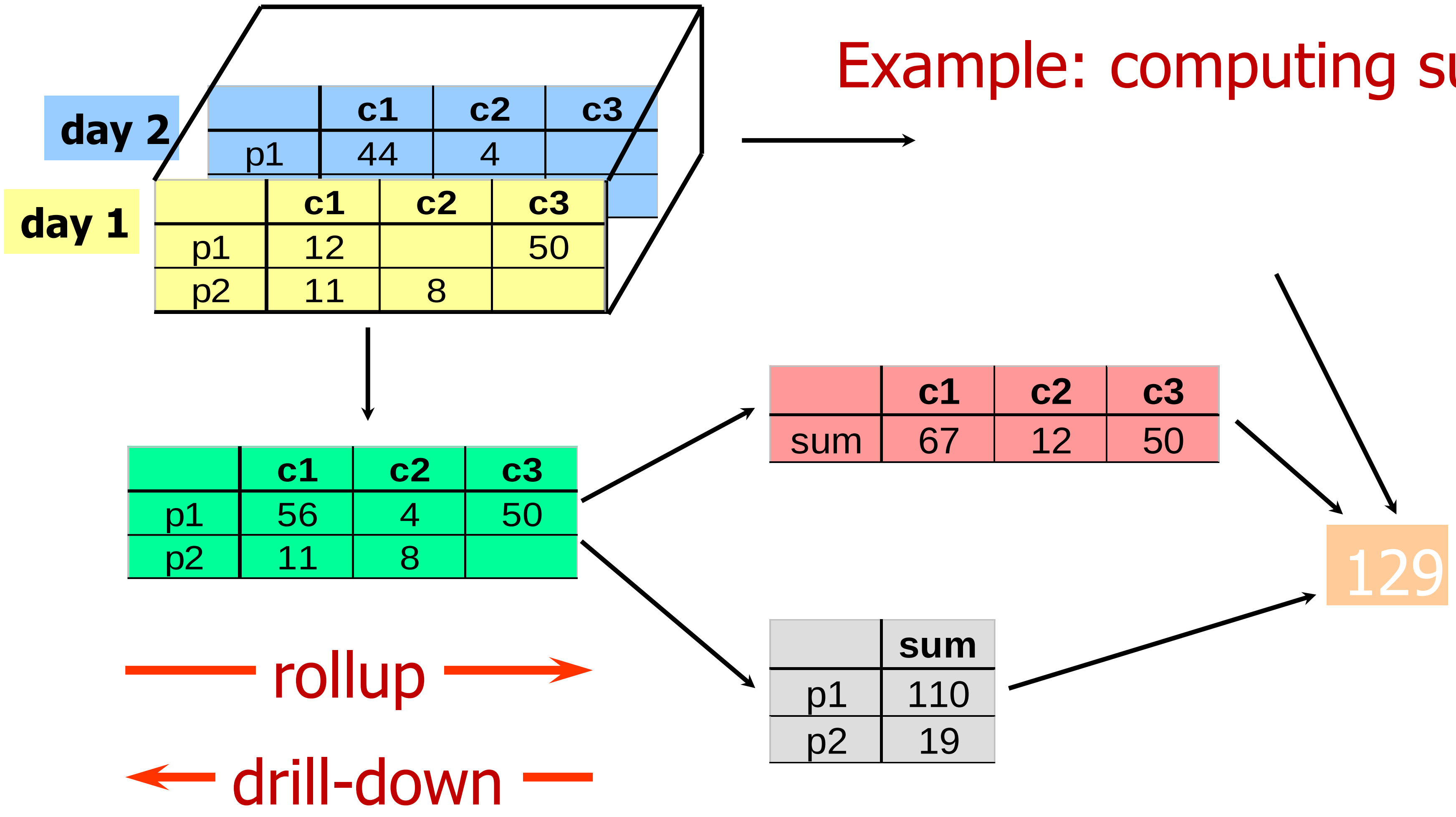
$\longleftarrow$ drill-down $\longleftarrow$

Aggregates

- Operators: sum, count, max, min, median, ave
- "Having" clause
- Using dimension hierarchy
  - average by region (within store)
  - maximum by month (within date)

Cube Aggregation

Example: computing sums



day 2

	c1	c2	c3
p1	44	4	

day 1

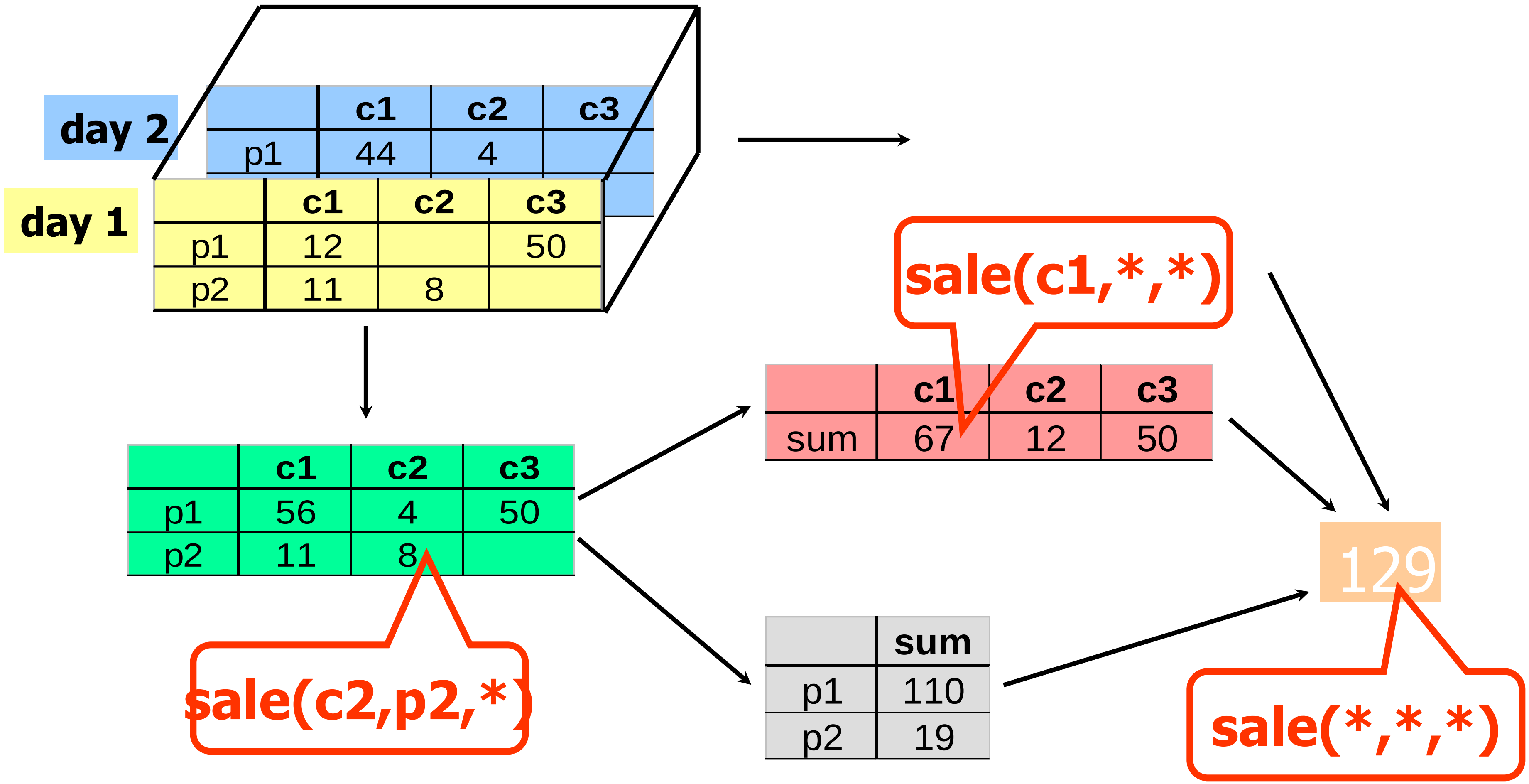
	c1	c2	c3
p1	12		50
p2	11	8	

	c1	c2	c3
p1	56	4	50
p2	11	8	

	c1	c2	c3
sum	67	12	50

	sum
p1	110
p2	19

Cube Operators



day 2

	c1	c2	c3
p1	44	4	

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

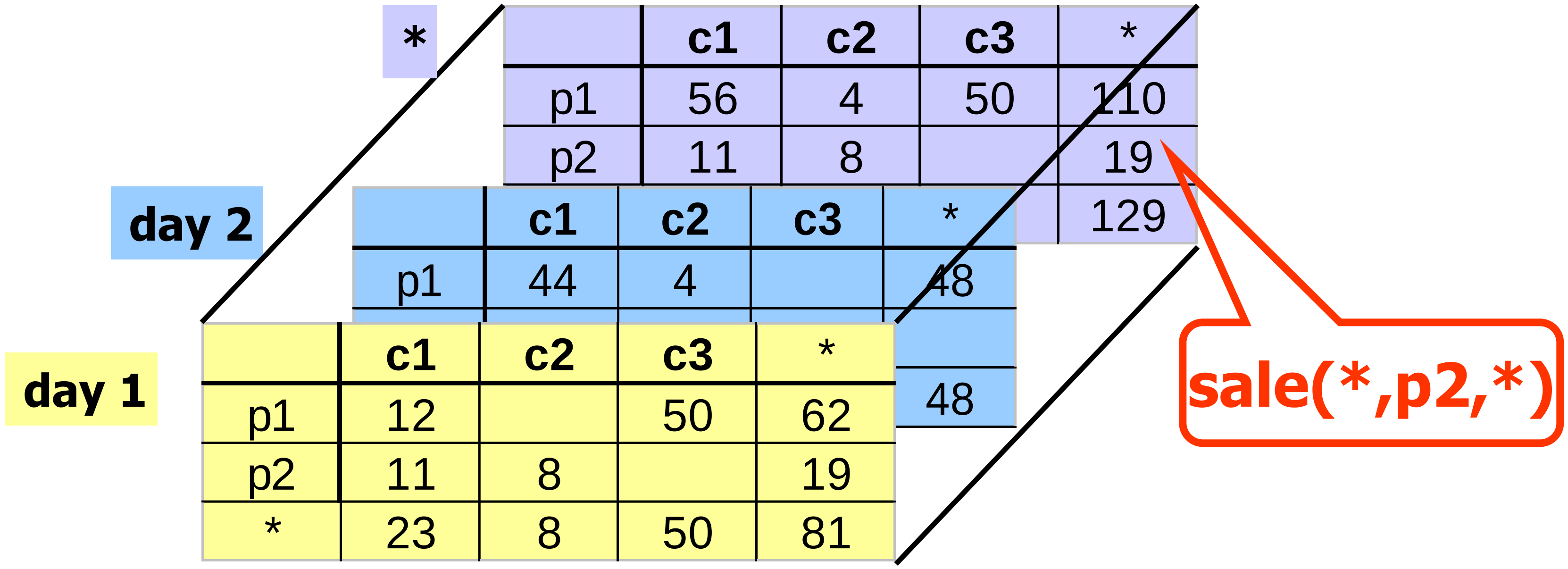
	c1	c2	c3
p1	56	4	50
p2	11	8	

	c1	c2	c3
sum	67	12	50

	sum
p1	110
p2	19

129

Extended Cube

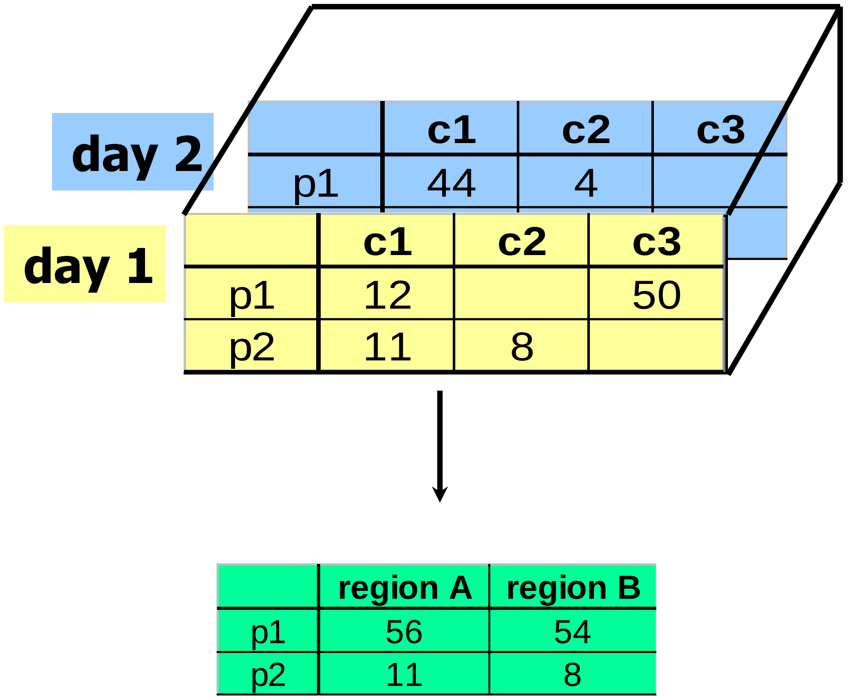


	c1	c2	c3	*
p1	56	4	50	110
p2	11	8		19
				129

	c1	c2	c3	*
p1	44	4		48
				48

	c1	c2	c3	*
p1	12		50	62
p2	11	8		19
*	23	8	50	81

Aggregation Using Hierarchies



day 2

	c1	c2	c3
p1	44	4	

day 1

	c1	c2	c3
p1	12		50
p2	11	8	

	region A	region B
p1	56	54
p2	11	8

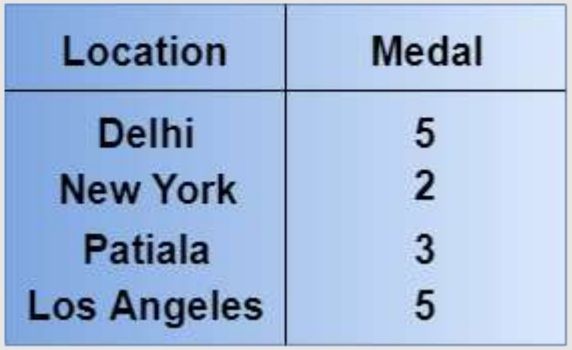
$$\begin{tikzcd}
\text{customer} \arrow[d, dash] \\
\text{region} \arrow[d, dash] \\
\text{country}
\end{tikzcd}$$

(customer c1 in Region A;
customers c2, c3 in Region B)

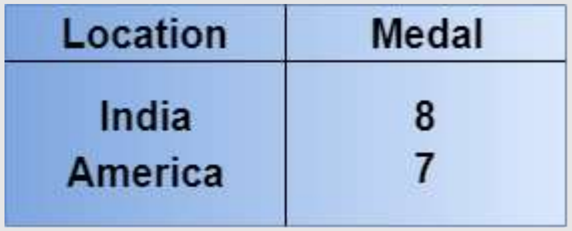
Example

Roll-up (drill-up):-
The roll-up operation performs aggregation on a data cube either by climbing up the hierarchy or by dimension reduction.
Consider an example:-

Delhi, New York, Patiala and Los Angeles wins 5, 2, 3 and 5 medals respectively. So in this example, roll upon Location from cities to countries.



Location	Medal
Delhi	5
New York	2
Patiala	3
Los Angeles	5



Location	Medal
India	8
America	7

Roll-up
More detailed data to less detailed data. WRITE SQL QUERY.

Example

Drill-down:-

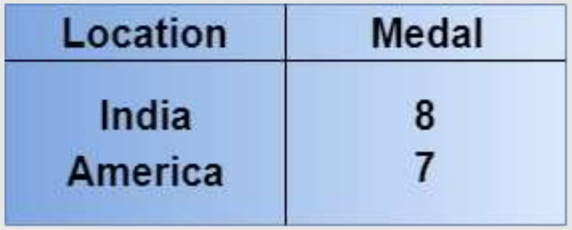
Drill-down is the reverse of roll-up.That means lower level summary to higher level summary.

Drill-down can be performed either by:-

1 Stepping down a concept hierarchy for a dimension

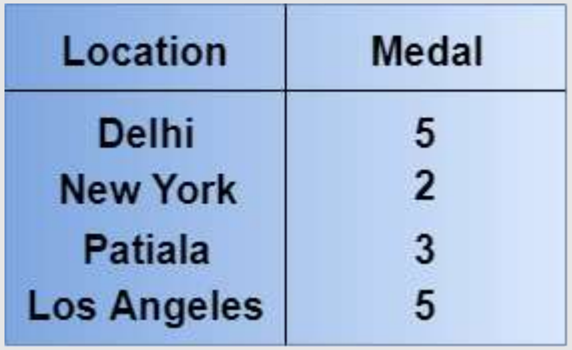
2 By introducing a new dimension.

Consider an example:-



Location	Medal
India	8
America	7

Drill-down on Location form countries to cities.



Location	Medal
Delhi	5
New York	2
Patiala	3
Los Angeles	5

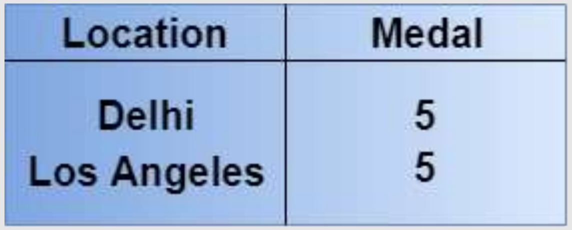
Drill-down

Less detailed data to More detailed data. WRITE SQL QUERY.

Example

Slice and dice
The **slice operation** performs a selection on one dimension of the given cube, resulting in a subcube.Reduces the dimensionality of the cubes.
For example, if we want to make a select where Medal = 5

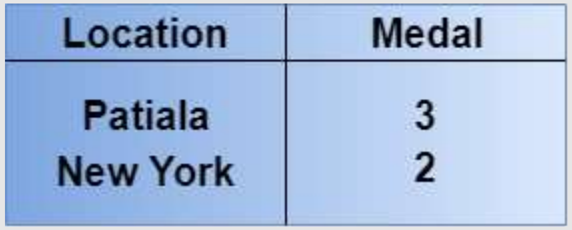
Slice Operation



Location	Medal
Delhi	5
Los Angeles	5

The **dice operation** defines a sub-cube by performing a selection on two or more dimensions.
For example, if we want to make a select where Medal = 3 or Location = New York

Dice Operation



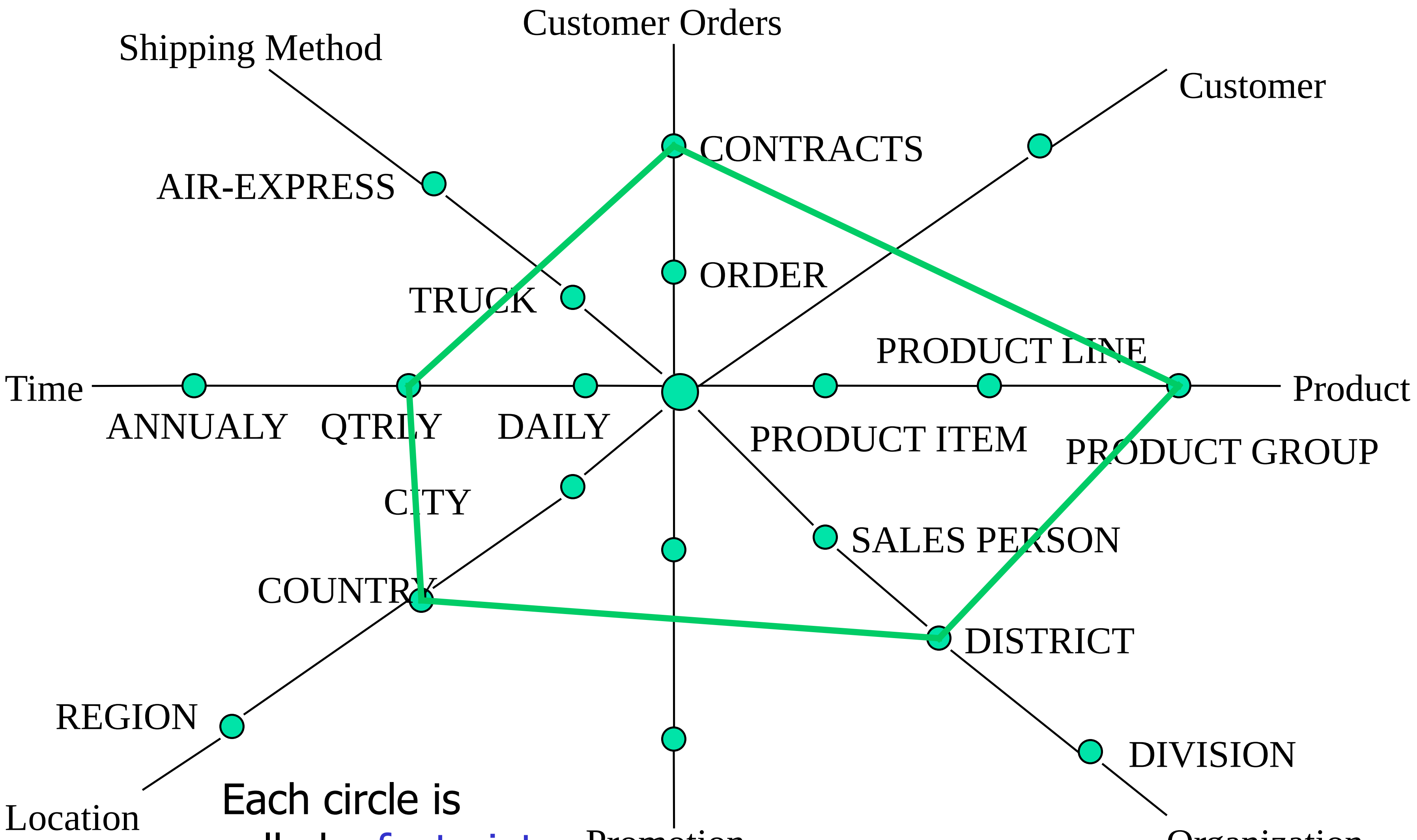
Location	Medal
Patiala	3
New York	2

WRITE SQL QUERY.

Querying the Multidimensional Model

- Based on a starnet model
- Consists of radial lines emanating from a central point, where each line represents a concept hierarchy for a dimension
- Each abstraction level in the hierarchy is called a footprint

A Star-Net Query Model



Each circle is called a footprint

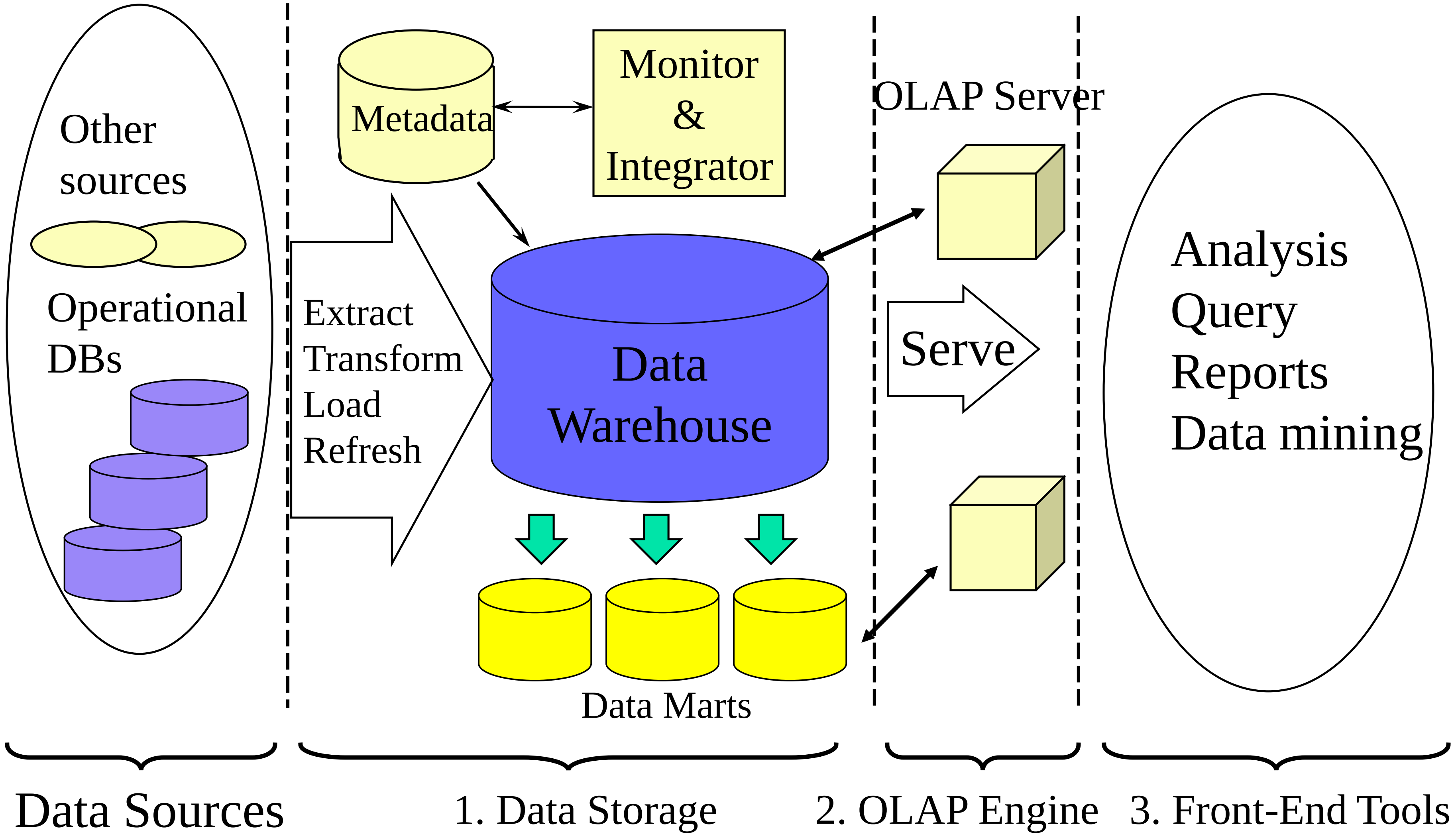
Data Warehousing

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

Data Warehouse Design Process

- Typical data warehouse design process
  1. Choose a business process to model
     - e.g., orders, invoices, etc.
     - If the process is organizational, consider DW or if it is departmental, then consider DMart
  2. Choose the *grain* (*atomic level of data*) of the business process e.g. individual transactions or individual daily snapshots
  3. Choose the dimensions that will apply to each fact table record e.g. item, time, customer, supplier, etc
  4. Choose the measure that will populate each fact table record e.g. dollars_sold, units_sold, avg_sales, etc.

Data Warehouse: A Multi-Tiered Architecture



Data Warehouse: A Multi-Tiered Architecture

- Bottom Tier
  - DataWareHouse Server
    - Almost a relational database system
    - Back end tools and utilities are used to feed data from operational databases or other external sources
      - By performing Extraction, Cleaning and Transformation, as well as Load and refresh functions to update DW
- Middle Tier
  - OLAP Server
    - Implemented with
      - ROLAP – extended RDBMS that maps operations on multidimensional data to standard relational operations
      - MOLAP – a special purpose server that directly implements multidimensional data and operations
- Top Tier
  - Front End Tools
    - Contains tools for query and reporting, analysis, and/or data mining

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

Metadata Repository

- Meta data -Data about data
  - For DW, it is the data that define warehouse objects
  - Play an important role
    - Used as a directory to help the decision support system analyst to locate the contents of the data warehouse
    - As a guide to the mapping of data when the data are transformed from the operational environment to the DW environment, etc.
- Stores:
  - Description of the structure of the data warehouse
    - Schema
    - View
    - Dimensions and Hierarchies
    - Derived data definition
    - Data mart locations and contents

Metadata Repository

- Stores:
  - Operational meta-data
    - Data lineage (history of migrated data and transformation applied to it)
    - Currency of data (active, archived or purged)
    - Monitoring information (warehouse usage statistics, error reports, audit trails)
  - The algorithms used for summarization
    - Related to measure and dimension
  - The mapping from operational environment to the data warehouse
    - Source databases and their contents
    - Data partitions
    - Data extraction
    - Cleaning, transformation rules, data refresh and purging rules and security

Metadata Repository

- Stores:
  - Data related to system performance
    - Indices and profiles that improve data access and retrieval performance
    - Rules for the timing and scheduling of refresh, update and replication cycles
  - Business metadata
    - Business terms and definitions
    - Ownership of data, charging policies
- Stored and managed persistently (on Disk) at Bottom tier

Three Data Warehouse Models

- Enterprise warehouse
  - Collects all of the information about subjects spanning the entire organization
  - Data ranges from GB to TB or beyond
  - Implemented on mainframes, super servers or parallel architecture platforms
  - Time to design and build in years
- Data Mart
- Virtual warehouse

Three Data Warehouse Models

- Enterprise warehouse
  - Collects all of the information about subjects spanning the entire organization
  - Data ranges from GB to TB or beyond
  - Implemented on mainframes, super servers or parallel architecture platforms
  - Time to design and build in years
- Data Mart
  - A subset of corporate-wide data that is of value to a specific groups of users
  - Scope is confined to specific, selected groups, such as marketing data mart or sales data mart
  - Implemented on low cost departmental servers
  - Implementation time in weeks
  - Categorized as
    - Independent where data captures from operational systems or external information provider
    - Dependent where data captures directly from enterprise data warehouse
- Virtual warehouse

Three Data Warehouse Models

- Enterprise warehouse
  - Collects all of the information about subjects spanning the entire organization
  - Data ranges from GB to TB or beyond
  - Implemented on mainframes, super servers or parallel architecture platforms
  - Time to design and build in years
- Data Mart
  - A subset of corporate-wide data that is of value to a specific groups of users
  - Scope is confined to specific, selected groups, such as marketing data mart or sales data mart
  - Implemented on low cost departmental servers
  - Implementation time in weeks
  - Categorized as
    - Independent where data captures from operational systems or external information provider
    - Dependent where data captures directly from enterprise data warehouse
- Virtual warehouse
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized
  - Easy to build but requires excess capacity on operational database servers

Differences between Data Warehouse and Data Mart

Parameter	**Data Warehouse**	**Data Mart**
Definition	A Data Warehouse is **a large repository** of data collected from different organizations or departments within a corporation.	A data mart is an only **subtype** of a Data Warehouse. It is designed to meet the need of a certain user group.
Usage	It helps to take a **strategic decision**.	It helps to take **tactical decisions** for the business.
Objective and Subject-area	The main objective of Data Warehouse is to provide an **integrated environment** and coherent picture of the business at a point in time for entire company.	A data mart mostly used in a business division at the **department level**. One subject area- for example, Sales figure.
Designing	The designing process of Data Warehouse is **quite difficult**.	The designing process of Data Mart is **easy**.
	Fact constellation schema is used.	It is built focused on a dimensional model using a **start schema**.
Data Handling	Data warehousing includes large area of the corporation which is why it takes a **long time to process** it.	Data marts are easy to use, design and implement as it can only handle small amounts of data.
Data type	The data stored inside the Data Warehouse are always **detailed when compared with data mart.**	Data Marts are built for particular user groups. Therefore, **data short and limited compare to DW**.
Data storing	Designed to store **enterprise-wide decision data**, not just marketing data.	Dimensional modeling and **star schema design** employed for optimizing the performance of access layer.
Data type	**Time variance and non-volatile design are strictly** enforced.	Mostly includes **consolidation data structures to meet subject area's query and reporting needs**.

Differences between Data Warehouse and Data Mart

Parameter	Data Warehouse	Data Mart
Data value	**Read-Only** from the end-users standpoint.	**Transaction data** regardless of grain fed directly from the Data Warehouse.
Source	In Data Warehouse Data comes from **many sources**.	In Data Mart data comes from **very few sources**.
Size	The size of the Data Warehouse may range from **100 GB to 1 TB+.**	The Size of Data Mart is less than **100 GB.**
Implementation time	The implementation process of Data Warehouse can be extended from **months to years**.	The implementation process of Data Mart is restricted to **few months**.

Data Warehouse Design Process

Top-down, bottom-up approaches or a combination of both

- Top-down
  - Starts with overall design and planning
    - If technology is mature and well known and business problems that must be solved are clear and well understood
    - DW as a single repository feeds data into data marts
    - Minimizes the integration problems
    - Expensive, take long time to develop and common data model for the entire organization
- Bottom-up
  - Starts with experiments and prototype
    - Useful in the early stage of business modeling and technology development, allows to move forward at less expense by independent data mart
    - But, leads to problems when integrating various disparate data marts into a consistent enterprise DW
- Combined
  - Planned and strategic nature of the top-down approach and rapid implementation of the bottom-up approach

Data Warehouse Design Process

Top-down, bottom-up approaches or a combination of both

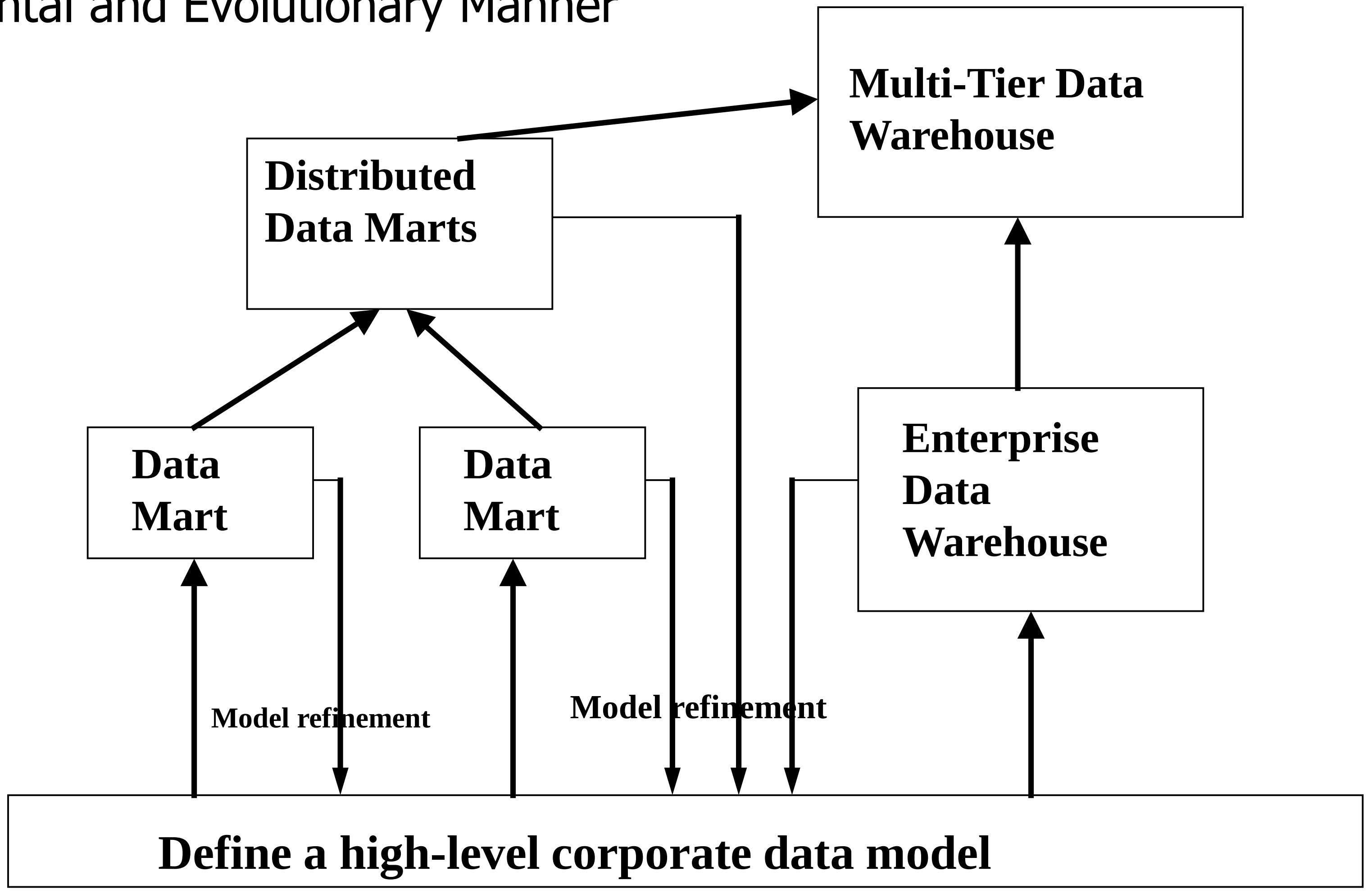
- Top-down
  - Starts with overall design and planning
    - If technology is mature and well known and business problems that must be solved are clear and well understood
    - Minimizes the integration problems
    - Expensive, take long time to develop and common data model for the entire organization
- Bottom-up
  - Starts with experiments and prototype
    - Useful in the early stage of business modeling and technology development, allows to move forward at less expense by independent data mart
    - Data flows from source into data marts, then into the data warehouse
    - But, leads to problems when integrating various disparate data marts into a consistent enterprise DW
- Combined
  - Planned and strategic nature of the top-down approach and rapid implementation of the bottom-up approach

Data Warehouse Design Process

- From software engineering point of view
  1. Planning
  2. Requirement Study
  3. Problem Analysis
  4. WareHouse Design
  5. Data Integration
  6. Testing
  7. Deployment of DataWareHouse
- 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 modification

Data Warehouse Development: A Recommended Approach

Incremental and Evolutionary Manner


Multi-Tier Data Warehouse
Distributed Data Marts
Data Mart
Data Mart
Enterprise Data Warehouse
Model refinement
Model refinement
Define a high-level corporate data model

OLAP Server Architectures

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)
- Specialized SQL servers
- Other OLAPs
  - DOLAP
  - WOLAP
  - MOLAP
  - SOLAP

OLAP Server Architectures

- <u>Relational OLAP (ROLAP)</u>
  - Uses relational tables or extended-relational DBMS to store and manage warehouse data and OLAP middle ware
  - Includes optimization of DBMS backend, implementation of aggregation navigation logic and additional tools and services
  - Greater scalability than MOLAP
  - Advantages
    - Scalable (billions of facts)
    - Flexible, design easier to change
    - New techniques adapted from MOLAP like Indexes, materialized views, special handling of star schemas
  - Disadvantages
    - Storage use (often 3-4 times MOLAP)
    - Response times

OLAP Server Architectures

- Multidimensional OLAP (MOLAP)
  - Sparse array-based multidimensional storage engine
  - Fast indexing to pre-computed summarized data
  - Advantages
    - Less storage use ("foreign keys" not stored)
    - Faster query response times
  - Disadvantages
    - Less flexible, e.g., cube must be re-computed when design changes

OLAP Server Architectures

- Hybrid OLAP (HOLAP)
  - Flexibility, e.g., low level: relational, high-level: array
  - e.g., Microsoft SQLServer
  - Advantages
    - Scalable and Fast
  - Disadvantages
    - Complexity
- Specialized SQL servers
  - Specialized support for SQL queries over star/snowflake schemas
  - e.g., Redbricks

Other OLAPs

Desktop OLAP (DOLAP)	In Desktop OLAP, a user downloads a part of the data from the database locally, or on their desktop and analyze it. DOLAP is relatively cheaper to deploy as it offers very few functionalities compares to other OLAP systems.
Web OLAP (WOLAP)	Web OLAP which is OLAP system accessible via the web browser. WOLAP is a three-tiered architecture. It consists of three components: client, middleware, and a database server.
Mobile OLAP	Mobile OLAP helps users to access and analyze OLAP data using their mobile devices
Spatial OLAP	SOLAP is created to facilitate management of both spatial and non-spatial data in a Geographic Information system (GIS)

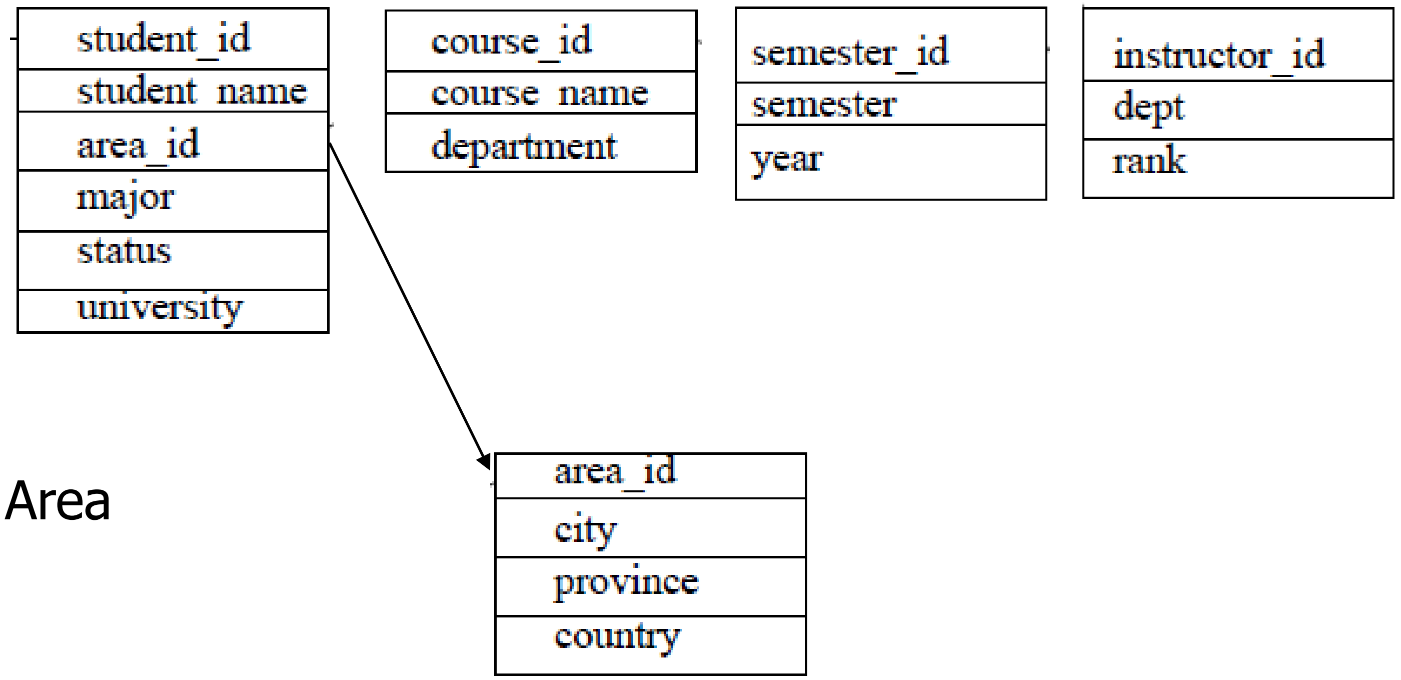
ROLAP Example

- A data warehouse for *Big University* consists of four dimensions: *student, course, semester* and *instructor*, and two measures *count* and *avg grade*.
- At the lowest conceptual level (e.g., for a given student, course, semester and instructor combination), the *avg grade* measure stores the actual course grade of the student. And at higher conceptual levels, *avg grade* stores the average grade for the given combination.
- (a) Draw a *snowflake schema* diagram for the data warehouse.

ROLAP Example

(a) Snowflake Schema

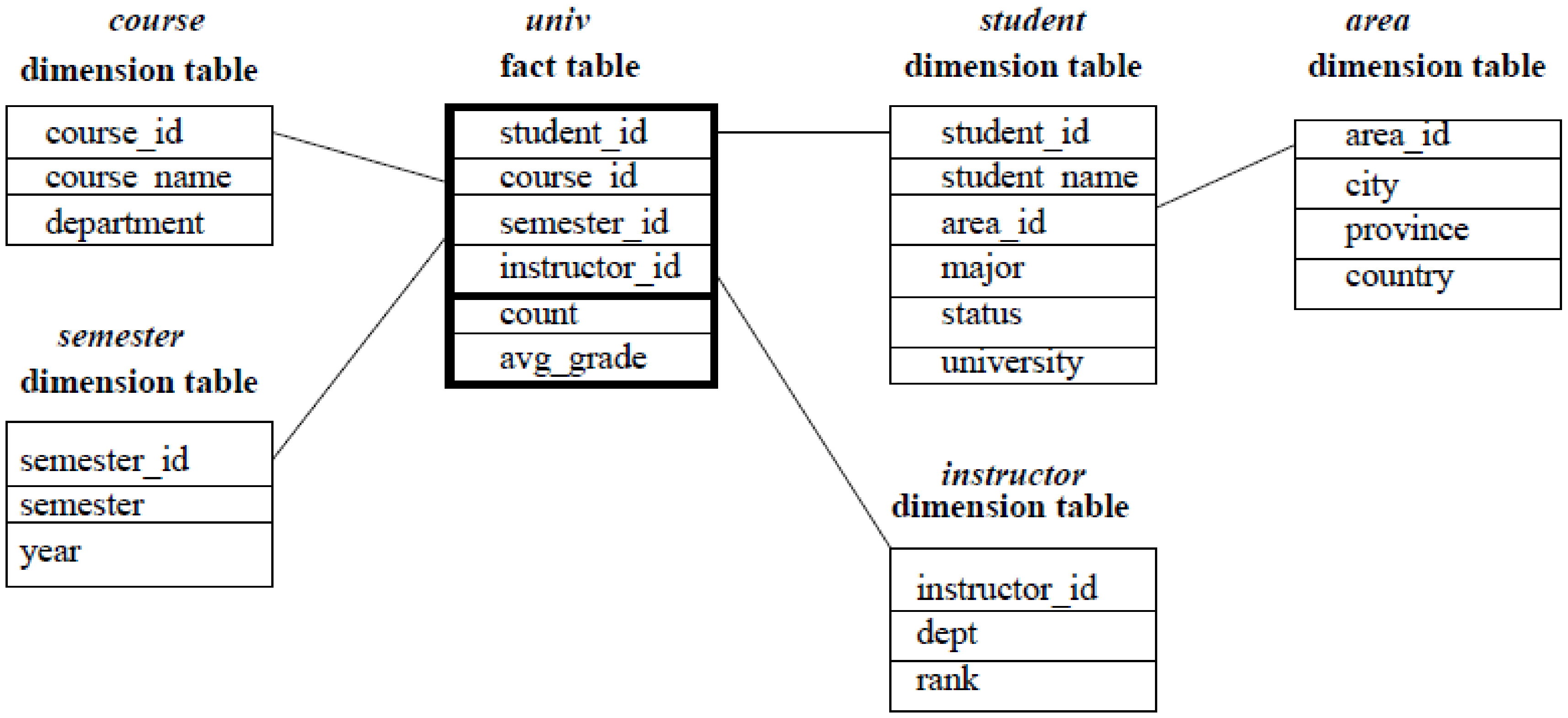
- Dimension Tables
- Student Course Semester Instructor



- Area

ROLAP Example

(a) Snowflake Schema



ROLAP Example

- (b) Starting with the base cuboid [*student*, *course*, *semester*, *instructor*], what specific *OLAP operations* should one perform in order to list the average grade of "*CS"* departments for each *Big University* student.

ROLAP Example

(b) OLAP operation to perform:

- To list the average grade of "*CS*" courses for each *Big University* student
  - Roll-up on course from course id to department
  - Roll-up on student from student id to university
  - Dice on course, student with department="CS" and university = "Big-University"
  - Drill-down on student from university to student name

Data Warehouse

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining

Data warehouse implementation

- DW system handles huge volumes of data
- OLAP servers demand that decision support query should be answered in the order of seconds
- DW needs
  - Highly efficient aggregation computation techniques
  - Access methods
  - Query processing techniques

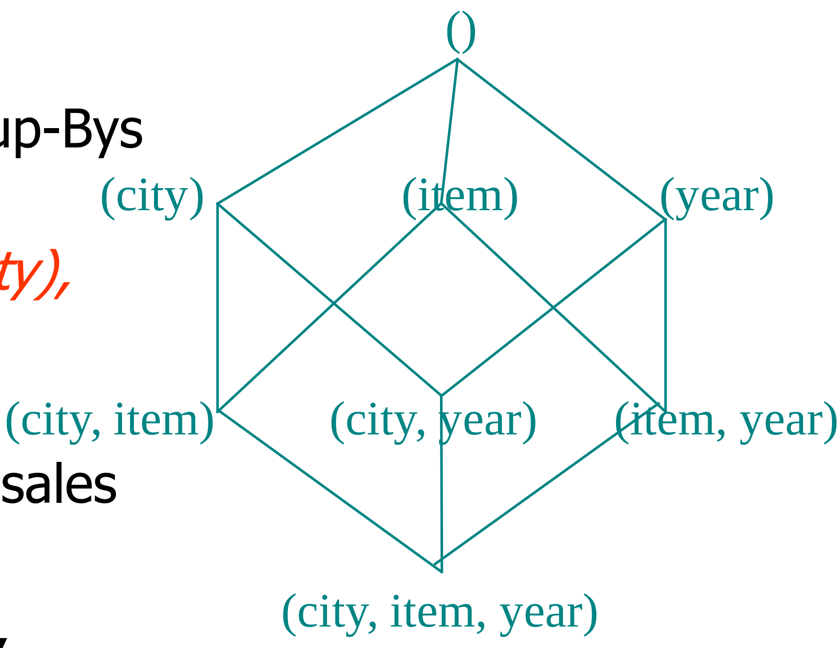
Efficient Data Cube Computation

- Highly efficient aggregation/CUBE computation techniques
  - Aggregation → group-by's
  - Each group-by → cuboid
  - Set of group-by's → A lattice of cuboids → Cube
- 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

Cube Operation

- For the cube with three attributes / dimensions
  - City, Item, Year
  - Need to compute the following Group-Bys

    (date, item, city),
    (date, item), (date, city), (item, city),
    (date), (item), (city)
    () → group-by is empty
    contains the total sum of all sales



  - Total no. of cuboids/group-by's

$= 2^3 = 8$

No. of Cuboids of Cube

- Total no. of cuboids
  - If no hierarchies associated with each dimension
    - $T=2^n$
  - If hierarchies
    - $T=\prod_{i=1}^{n}(L_i+1)$ where L_i=No. of levels associated with dimension i, 1 is added to L_i for virtual level all

- Example
  - If 10 dimension, each having 5 levels (including all)
    - $T= 5^{10} = 9.8 \times 10^6$ (Approx)
    - Very large

No. of Cuboids of Cube

- Size of each cuboid also depends on the cardinality (i.e. no. of distinct values) of each dimension
- For example, if in Sales cube, each city sold every item
  - |city| x |item| tuples in the city-item group-by alone

- Storage space of cube (i.e. Precomputation) depends on
  - No. of dimensions
  - No. of concept hierarchies
  - Cardinality

Selected Computation of cuboids

- Materialization of data cube
    - No materialization
        - No precomputation any of the Nonbase cuboids
        - Computing expensive multidimensional aggregates on the fly
        - Slow
    - Full materialization
    - Partial materialization

Selected Computation of cuboids

- Materialization of data cube
  - No materialization
    - No precomputation any of the Nonbase cuboids
    - Computing expensive multidimensional aggregates on the fly
    - Slow
  - Full materialization
    - Precompute/Materialize <u>every</u> cuboid
    - Resultant lattice is Full cube
    - Need of huge amount of memory space
  - <u>Partial materialization</u>

Selected Computation of cuboids

- Materialization of data cube
  - No materialization
    - No precomputation any of the Nonbase cuboids
    - Computing expensive multidimensional aggregates on the fly
    - Slow
  - Full materialization
    - Precompute/Materialize <u>every</u> cuboid
    - Resultant lattice is Full cube
    - Need of huge amount of memory space
  - <u>Partial materialization</u>
    - Selectively compute a proper subset of the whole set of possible cuboids
      - Based on size, sharing, access frequency, etc.

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
    - Suppose 100 dimensions, only 1 base cell. How many aggregate cells if count >= 1? What about count >= 2?

Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- 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

Chapter 3: Data Warehousing and OLAP Technology: An Overview

- What is a data warehouse?
- A multi-dimensional data model
- Data warehouse architecture
- Data warehouse implementation
- From data warehousing to data mining
- Summary

Design of Data Warehouse

Need to understand and analyze business needs, Known as Business Analysis Framework consisting of Four views regarding the design of a DW

- Top-down view
  - Allows selection of the relevant information necessary for the DW like current and future business needs
- Data source view
  - Exposes the information being captured, stored, and managed by operational systems
  - Modeled by ER model
- Data warehouse view
  - Consists of fact tables and dimension tables
  - Represents the actual information that is stored inside the DW, including precalculated totals and counts, as well as information regarding the source, date, time, etc. to provide historical context
- Business query view
  - Sees the perspectives of data in the DW from the view of end-user

After DW design

- Initial deployment of the warehouse
  - Initial installation
  - Roll-out planning
  - Training
  - Orientation
  - Platform upgradation
  - Maintenance
- DW Administration
  - Data refreshment
  - Data source synchronization
  - Planning for disaster recovery
  - Managing access control and security
  - Managing data growth
  - Managing database performance
  - DW enhancement and extension

- Scope Management
  - Controlling the number and range of queries, dimensions and reports
  - Limiting the size of the DW or schedule

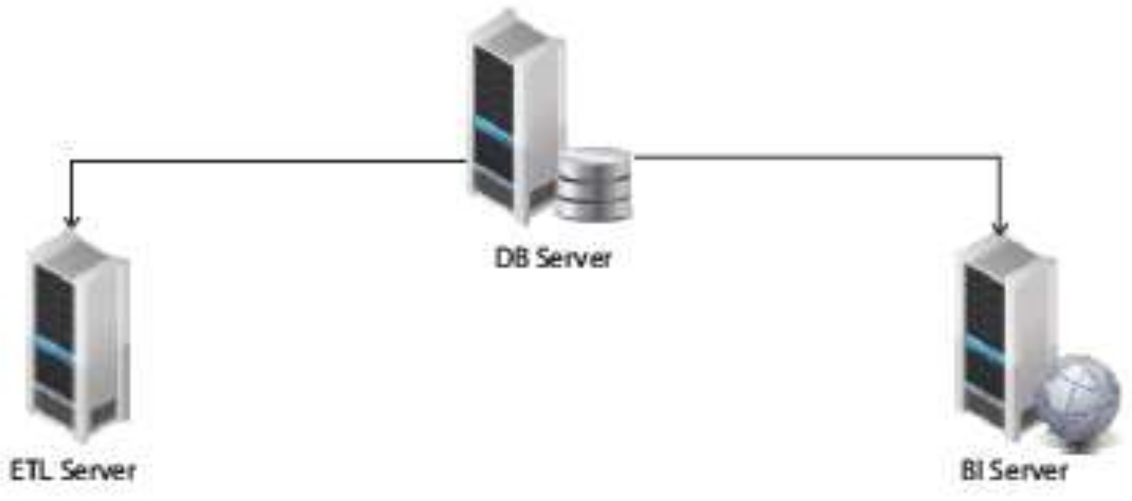
Big Data Warehousing

Traditional Data Warehouse

- A central repository of information, created by
  - Integrating data from multiple disparate sources
  - Storing current as well as historical data
- Utilized to create trending reports for senior management, such as annual and quarterly comparisons
- The data uploaded from operational systems such as sales, customer relationship management, and marketing systems

Traditional DW Architecture

- Monolithic Servers



- Majority of DWs run on monolithic-style server systems
- Data warehouse solution has three key monolithic servers
  - The database (DB) server,
  - The extraction, transformation, and loading (ETL) server and
  - The Business Intelligence (BI) server

Traditional DW Architecture

- Monolithic Servers
  - Powerful machines
    - Accommodate multiple users accessing the systems and achieve low-latency response times to queries
  - Works well for queries
    - On the data for which the data structures are known
  - Runs trend analyses and business reporting
    - On a certain subset of data or lines of business
  - Additionally, produces a lot of network traffic
    - As the ETL server must pull data from the DB server in order to process the request and load the results back into the DB server

Traditional DW Architecture

- Monolithic Servers
  - The database (DB) server
    - The database of the data warehouse, the data marts, and the data staging area
    - Purpose is to be **used as the data repository** that **the other two servers will read from and/or feed data into**
  - The extraction, transformation, and loading (ETL) server
    - Purpose is to house the extract, transform, and load systems that will be responsible for **collecting data and changing it into a more usable form** to produce reports or decision making systems
  - The Business Intelligence (BI) server
    - Houses the **reporting and dashboard components** the business uses to gain insight into the data found on the DB server

Traditional DW Advantages

- Known structures of data can easily be related to one another
  - Data that is known to most of a company’s personnel and that is tightly coupled to other aspects of the company makes it an invaluable source of information for critical decisions
- Answers business questions quickly, making them the most efficient place to find information on the status of the enterprise
  - By adding custom-tailored business intelligence dashboards, data in a DW can be interpreted at a glance
- Well known technologies most professionals understand, like RDBMS and DWs, have been around for several decades and have small learning curves
  - Almost anyone with a basic knowledge of these systems can perform analysis with relative ease. Even recent college graduates have a general understanding of these technologies and don't require extensive training to be able to contribute to projects or solve data problems

Traditional DW Limitations

- Inflexibility in integrating new data models
  - When new sources of information are identified, it can take **tremendous effort to integrate these new data models** into a highly-structured and tightly-coupled system like a DW
- Takes a great deal of effort to implement changes to a DW in order to answer new questions
  - When an enterprise's priorities change or markets change, the business will **want a system that can change just as quickly to answer new questions**
  - But, having to modify or build new data structures, dashboards, and reports is a **large undertaking** because the new designs ideally should be compatible with the old in order to provide a holistic view of enterprise data

Traditional DW Limitations

- DWs are ill-suited for data discovery
  - Businesses increasingly rely on data to inform decisions; however, the problem with DWs is that much of the data they contain may end up **being left behind in the upload portion of the ETL process due to size constraints** or the business not being interested in certain data at the time of transfer
  - An incomplete data set mined for a specific purpose will be ill suited to answer any other question, meaning that businesses trying to use a DW for data discovery will likely **draw incorrect conclusions about patterns and trends**

Traditional DW Limitations

- The downtime for a complete hardware upgrade can be massive
  - Due to the monolithic architecture of DW, if ever an upgrade of an entire server is needed, an enterprise runs the risk of downtime while the upgrade is in progress
  - The time and effort spent migrating a current system architecture to a new server **makes upgrading even more costly**

Data Warehouse vs Big Data

- Data warehouse
  - Common words for last 10-20 years
- Big data
  - Hot trend for last 5-10 years

- Both of them hold a lot of data, used for reporting, managed by an electronic storage device

Data Warehouse vs Big Data

BASIS FOR COMPARISON	**DATA WAREHOUSE**	**BIG DATA**
Meaning	An architecture, not a technology **Extracting data from varieties SQL based data source** (mainly relational database) and help for generating analytic reports A data repository, which using for any analytic reports, has been generated from ETL process	A technology, which stands on **volume, velocity, and variety of the data** Volumes define the amount of data coming from different sources, velocity refers to the speed of data processing, and varieties refer to the number of types of data (mainly support all type of data format)
Preferences	If an organization wants to know some informed decision (like what is going on in their corporation, **next year planning based on current year performance data** etc), they prefer to choose data warehousing, as for this kind of report they need reliable or believable data from the sources	If organization need to compare with a lot of big data, which contain valuable information and help them **to take a better decision** (like how to lead more revenue, more profitability, more customers etc), they obviously preferred Big Data approach
Accepted Data Source	**One or more homogeneous** (all sites use the same DBMS product) or **heterogeneous** (sites may run different DBMS product) **data sources**	**Any kind of sources**, including business transactions, social media, and information from sensor or machine specific dat
Accepted type of formats	Handles mainly **structural data** (specifically relational data)	Accepted **all types of formats**: Structure data, relational data, and unstructured data including text documents, email, video, audio, stock ticker data and financial transaction

Data Warehouse vs Big Data

BASIS FOR COMPARISON	DATA WAREHOUSE	BIG DATA
Subject-Oriented	**Subject oriented** because it actually provides information on the specific subject (like a product, customers, suppliers, sales, revenue etc) not on organization ongoing operation. It does not focus on ongoing operation, it mainly focuses on analysis or displaying data which help on decision making.	**Also subject-oriented**, the **main difference** is a source of data, as big data can **accept and process data from all the sources** including social media, sensor or machine specific data. It also **provide exact analysis on data** specifically on subject oriented.
Time-Variant	The data collected is actually identified by a **particular time period**. As it mainly holds historical data for an analytical report.	Big Data have a lot of approach to identified already loaded data, a time period is one of the approaches on it. As Big data mainly processing flat files, so archive with date and time will be the best approach to identify loaded data. But it **also works with streaming data**, so it **not always holding historical data**.
Non-volatile	**Previous data never erase** when new data added to it. This is one of the major features of a data warehouse. As it is **totally different from an operational database**, so any changes on an operational database **will not directly impact** to a data warehouse.	For Big data, again **previous data never erase** when new data added to it. It stored as a file which represents a table. But, here sometime in **case of streaming directly use technology as operation environment**.
Distributed File System	Processing of huge data in Data Warehousing is really **time-consuming** and sometimes it took an entire day for complete the process.	This is one of the big utility of Big Data. HDFS defined to load huge data in distributed systems by using map reduce program.

Data Warehouse vs Big Data

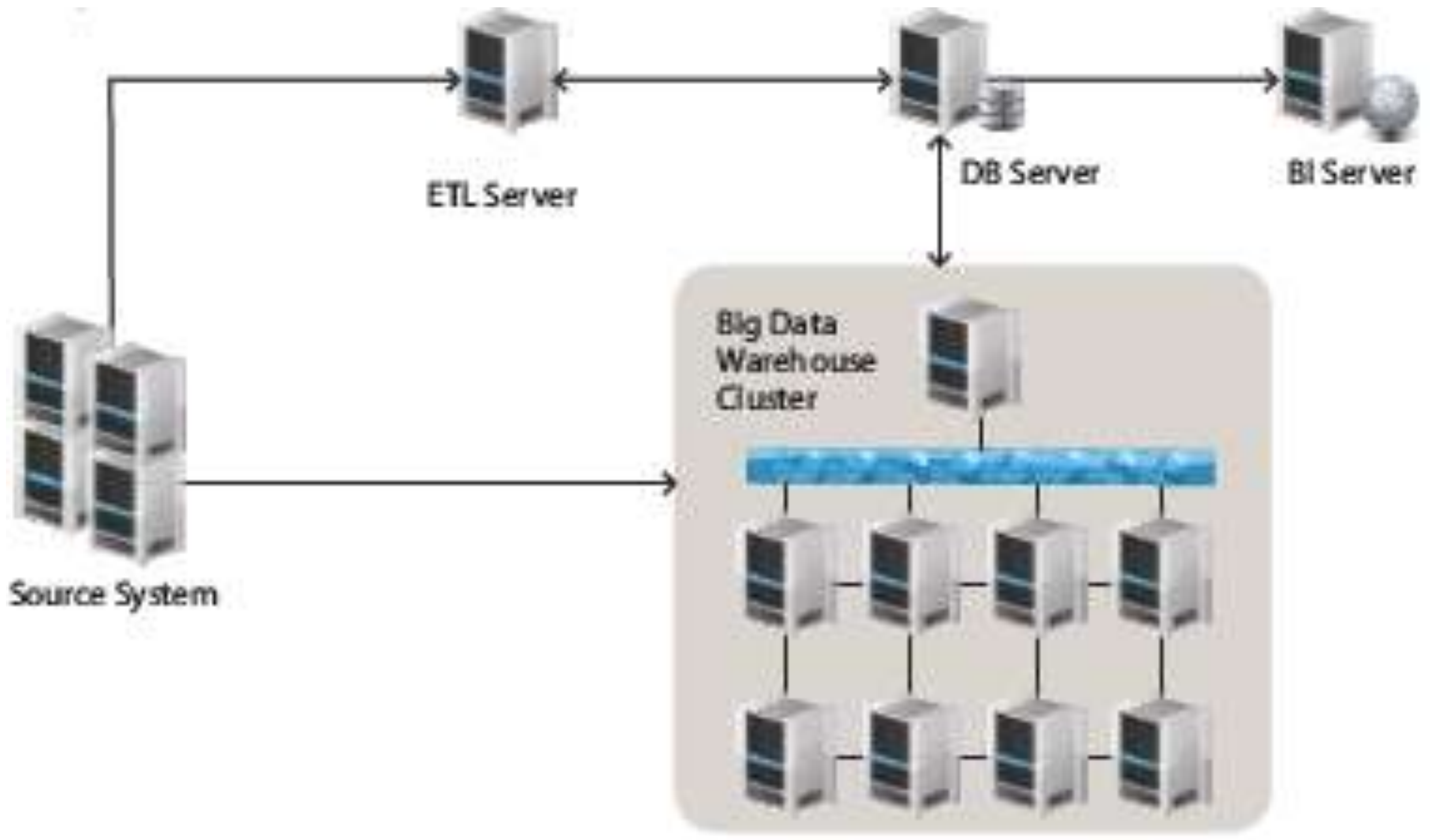
- Both are not same, so they are not interchangeable
- An organization can follow Data Warehouse or Big Data or the combination of both big data as well as data warehouse solution based on their need

- **Big Data Warehouse Architecture**

Big Data Warehouse Architecture

- **Distributed Architecture**
  - Implementing a DW using big data technologies takes an enterprise away from monolithic architecture and into distributed architectures, in which several servers or components of hardware work together to accomplish a task or process
  - Big data technologies are the equivalent of a file system as opposed to a database.
  - This gives big data technologies some compelling advantages like unlimited computing power and storage space, automatic data redundancy, automatic load balancing, or the ability to store an unlimited amount of data in any format
  - Depending on how a BDW is implemented, an enterprise can save money on hardware by outsourcing the data center and by leveraging cloud computing services to host and process data. Since most big data technologies are based on open source projects, enterprises can save money on licenses and other fees

Use case - Big Data Repository to Traditional DW



Use case - Big Data Repository to Traditional DW

- In this solution, an enterprise can run its current ETL processes on a traditional DW that houses the data attributes important to the business
- Additionally, an enterprise will create ELT processes that collect all data from source transactional systems
- Not having to retroactively load data or run heavy data mining queries will give an enterprise the flexibility to answer new questions as they arise and find new trends and patterns without slowing the source system. Once an enterprise establishes the new data model that will be required in the traditional DW, it can collect the new data quickly and easily, in a way that will be transparent to other departments
- With this solution, an enterprise could also use the big data repository as a backup site for the traditional DW in case of a catastrophic system failure
- Additionally, if an enterprise has processes that transfer data from a big data repository to the DW table structures, no retroactive loads will be required upon recovery

Use case-Augmented BD Processing to DW or DM

- By augmenting data processing from a monolithic structure to a distributed system, an enterprise can use the distributed system as a data aggregation/integration point for data that might not be available in the traditional DW
- This solution is typically used when an enterprise has a very processor intensive function that takes too long to complete or because an enterprise needs to run calculations on raw data that is impractical to store in a DW system
- By leveraging big data technologies that offer greater computing power and storage space, an enterprise can process data quickly and efficiently

Big Data-ELT Workflow

- In traditional DW environments, ETL is the term usually associated with the process of data acquisition and processing
- With big data technologies, the order of execution is usually different compared to classic DW processes, wherein an ELT process has the load step occurring before the transform step
  - Due to the fact that big data technologies are distributed file systems that can ingest data of any kind and format
  - Thus, an enterprise can use big data technologies to extract raw data from source systems, load it to a big data implementation program, and later transform it into any required format or structure
  - Useful when mining new trends and behavior patterns from data; because one is managing raw, unstructured data, the same piece of information can be used to gain insight into multiple scenarios or can be used to answer other questions that the business may ask
  - When dealing with structured data, the ELT flow can still apply, only data profiling and staging are done within the same distributed system

Big DW Advantages

- Limitless storage space and computing power to process terabytes of data
  - As easy as adding new servers to a cluster
    - This is something monolithic servers and databases cannot accomplish due to their physical limitations
  - Big data technologies can be implemented on cheap commodity servers, the investment to create a cluster is minimal as compared to building a system on a larger, higher-end server
  - Technologies are available for free through open source projects, licensing costs can be kept low compared to the cost of database licenses
    - Example, one could create a cluster of 10 servers that have 100 TB of storage and 40 processing cores that would cost as much as a single server that has less storage capacity and fewer processing cores

Big DW Advantages

- Effortless integration of any and all data types and models
  - Because big data is implemented on a file system and not a DBMS, an enterprise is able to ingest any and all formats and structures of data without incurring upfront overhead costs of integration
- Flexibility to run purpose-built DW business decision systems, trend analysis and predictive analytics processes
  - Since a BDW can handle structured, semi-structured, and unstructured data, it can output very loosely coupled data that can be changed, manipulated and combined in any way to discover new insights on data
- No downtime for forklift upgrades
  - Unlike systems using a more monolithic architecture, big data is able to accommodate entire sets of servers being taken offline for upgrade without incurring downtime or data loss
  - Since big data technologies replicate their data across difference nodes in the cluster, the risk of losing data due to hardware changes is minimal

Big DW Limitations

- Personnel will need to be trained on how to use new technologies
  - Because these technologies are relatively new and introduce new programming models (map, reduce), few professionals know how to properly leverage these technologies
  - This in turn introduces a few challenges such as:
    - Rapid technological change
    - Personnel must know a wide range of technologies and how they can be integrated

Big DW Limitations

- This in turn introduces a few challenges such as:
  - Rapid technological change
    - Because of their open source nature, big data technologies are undergo faster iteration cycles than traditional proprietary technologies, and new solutions are introduced into the market quickly
    - An example of this is that Hadoop will soon go to phase 2.0, addressing the single point of failure Namenode and implementing the concept of a Namenode Federation to address the single point of failure issue
  - Personnel must know a wide range of technologies and how they can be integrated
    - An example of this is how the Hadoop project has spawned an entire ecosystem
    - Just knowing and training for Hadoop is not enough; personnel must also know how to implement and integrate frameworks and systems like Avro, Cassandra, Chukwa, Hbase, Hive, Mahout, Pig, Impala, Sqoop, Oozie and ZooKeeper

Summary: Data Warehouse and OLAP Technology

- Why data warehousing?
- 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
- Data warehouse architecture
- OLAP servers: ROLAP, MOLAP, HOLAP
- Efficient computation of data cubes
  - Partial vs. full vs. no materialization
  - Indexing OALP data: Bitmap index and join index
  - OLAP query processing
- From OLAP to OLAM (on-line analytical mining)

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. VLDB'96
- D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek. Efficient view maintenance in data warehouses. SIGMOD'97
- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. ICDE'97
- S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997
- E. F. Codd, S. B. Codd, and C. T. Salley. Beyond decision support. Computer World, 27, July 1993.
- J. Gray, et al. Data cube: A relational aggregation operator generalizing group-by, cross-tab and sub-totals. Data Mining and Knowledge Discovery, 1:29-54, 1997.
- A. Gupta and I. S. Mumick. Materialized Views: Techniques, Implementations, and Applications. MIT Press, 1999.
- J. Han. Towards on-line analytical mining in large databases. *ACM SIGMOD Record*, 27:97-107, 1998.
- V. Harinarayan, A. Rajaraman, and J. D. Ullman. Implementing data cubes efficiently. SIGMOD'96

References (II)

- C. Imhoff, N. Galemmo, and J. G. Geiger. Mastering Data Warehouse Design: Relational and Dimensional Techniques. John Wiley, 2003
- W. H. Inmon. Building the Data Warehouse. John Wiley, 1996
- R. Kimball and M. Ross. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2ed. John Wiley, 2002
- P. O'Neil and D. Quass. Improved query performance with variant indexes. SIGMOD'97
- Microsoft. OLEDB for OLAP programmer's reference version 1.0. In http://www.microsoft.com/data/oledb/olap, 1998
- A. Shoshani. OLAP and statistical databases: Similarities and differences. PODS'00.
- S. Sarawagi and M. Stonebraker. Efficient organization of large multidimensional arrays. ICDE'94
- OLAP council. MDAPI specification version 2.0. In http://www.olapcouncil.org/research/apily.htm, 1998
- E. Thomsen. OLAP Solutions: Building Multidimensional Information Systems. John Wiley, 1997
- P. Valduriez. Join indices. ACM Trans. Database Systems, 12:218-246, 1987.
- J. Widom. Research problems in data warehousing. CIKM'95.

References (III)

- *Courtney Webster,* "Integrated Analytics *Platforms and Principles for Centralizing Your Data", O'reilly*