Ambo University||WC

School of Technology & Informatics

Information Systems

CHAPTER TWO

DATA WAREHOUSING

OUTLINES

- What is Data Warehouse?
- Data Warehouse vs. Operational DBMS
- OLTP vs. OLAP
- Design of a Data Warehouse: A Business Analysis Framework
- From Tables and Spreadsheets to Data Cubes
- Cube: A Lattice of Cuboids
- Conceptual Modeling of Data Warehouses
- A Data Mining Query Language, DMQL: Language Primitives
- Measures

What is Data Warehouse?

- “Data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process.”

- A decision support ***database*** that is maintained separately from the organization’s operational database and supports information processing by providing a solid platform of consolidated, historical data for analysis.

- It is a repository of multiple heterogeneous data sources organized under a unified schema at a single site in order to facilitate management decision making

What is Data Warehouse?

- Data warehouse allows “knowledge workers” (such as managers, analysts, and executives) to use the warehouse to quickly and conveniently obtain an overview of the data and to make sound decision based on information in the warehouse

- **Data warehousing** is 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.

Data Warehouse:
Time Variant

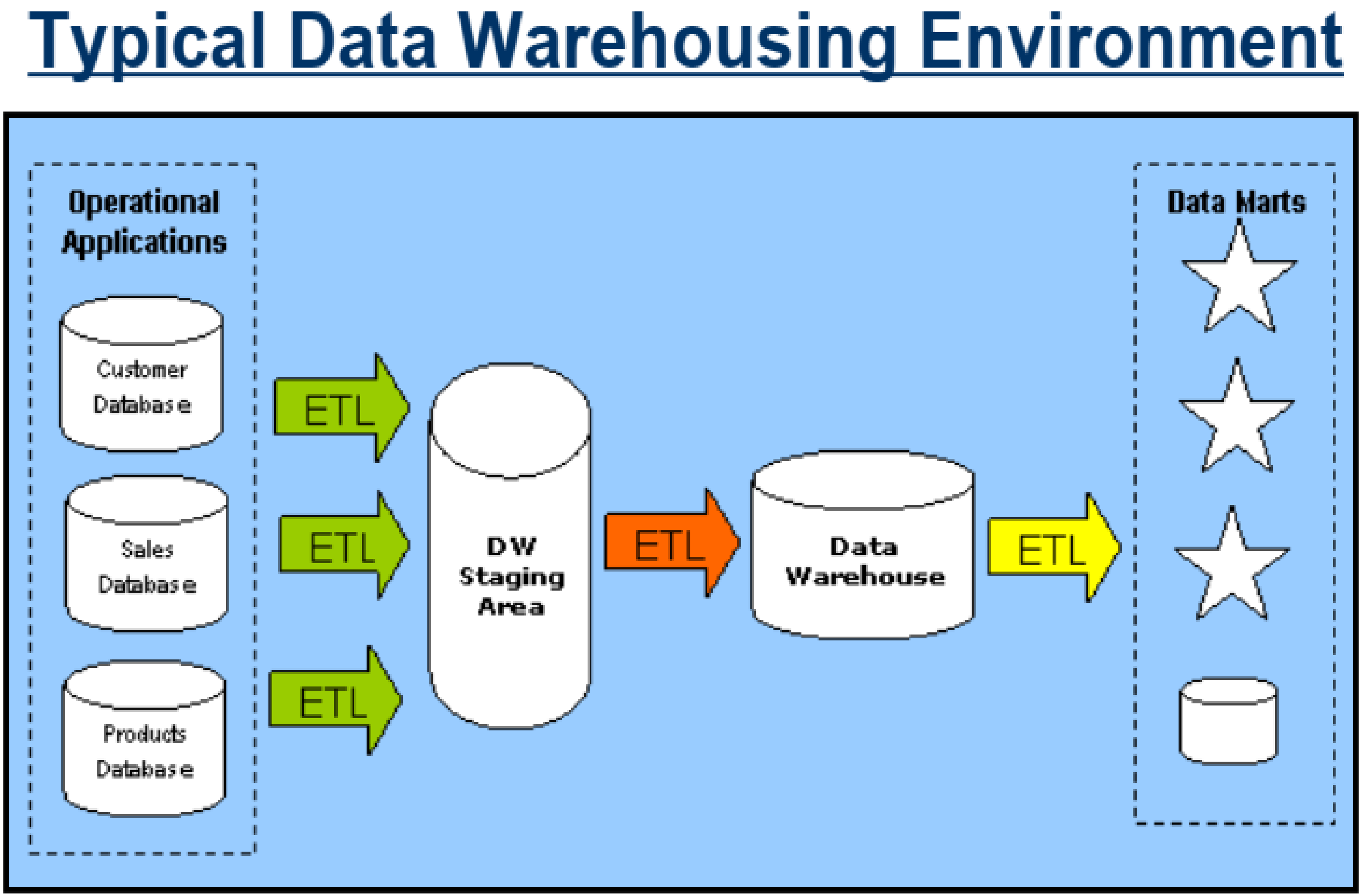
- The time horizon for the data warehouse is significantly longer than that of operational systems.
  - Operational database: current value data.
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)

- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly
  - But the key 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*.

Architecture of DW



Operational Systems

- An operational data store (ODS) is basically a database that is used for being an temporary storage area for a datawarehouse.
- Its primary purpose is for handling data which are progressively in use.
- Operational data store contains data which are constantly updated through the course of the business operations.

ETL Process

- ETL (Extract, Transform, Load) is used to copy data from:-
- ODS to data warehouse staging area.
- Data warehouse staging area to data warehouse .
- Data warehouse to data mart .
- ETL extracts data, transforms values of inconsistent data, cleanses "bad" data, filters data and loads data into a target database.

DW Staging Area

- The Data Warehouse Staging Area is temporary location where data from source systems is copied.
- It increases the speed of data warehouse architecture.
- It is very essential since data is increasing day by day.

Data Warehouse

- The purpose of the Data Warehouse is to integrate corporate data.
- The amount of data in the Data Warehouse is massive. Data is stored at a very deep level of detail.
- This allows data to be grouped in unimaginable ways.
- Data Warehouses does not contain all the data in the organization ,It's purpose is to provide base that are needed by the organization for strategic and tactical decision making.

Data Marts

- ETL extract data from the Data Warehouse and send to one or more Data Marts for use of users.
- Data marts are represented as shortcut to a data warehouse ,to save time.
- It is just an partition of data present in data warehouse.
- Each Data Mart can contain different combinations of tables, columns and rows from the Enterprise Data Warehouse.

REASONS FOR CREATING AN DATA MART

- Easy access to frequently needed data.
- Creates collective view by a group of users.
- Improves user response time.
- Ease of creation.
- Lower cost than implementing a full Data warehouse

Data Warehouse vs. Operational DBMS

- DBMS— tuned for OLTP:
  - access methods, indexing, concurrency control, recovery mechanism are desirable

- Warehouse—tuned for OLAP:
  - complex OLAP queries, multidimensional view, consolidation are desirable.
  - Indexing, concurrency control, recovery mechanism are not desirable in warehouse

OLTP vs. OLAP

- 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

OLTP vs. OLAP

- OLTP and OLAP differs in
  - ***User and system orientation***
  - ***Data contents they operate***
  - ***Database design used***
  - ***View***
  - ***Data Access patterns***

OLTP vs. OLAP

- ***User and system orientation:***
  - OLTP is customer oriented system used for transaction and query processing by clerks, clients and information technology professionals
  - OLAP is market oriented system used for data analysis by knowledge workers including managers, executives, and analysts

- ***Data contents:***
  - OLTP contains current, detailed data where as
  - OLAP systems contains large, historical, consolidated data and provides facilities for summarization, aggregation

OLTP vs. OLAP

- ***Database design:***
  - OLTP adopt ER for data modeling and application oriented DB design
  - OLAP uses star type model and subject oriented DB design.

- ***View:***
  - OLTP focus on current and local data view where as
  - OLAP has multiple version of DB schema due to evolutionary process of the enterprise

OLTP vs. OLAP

- ***Access patterns:***
  - OLTP access pattern is usually update where as
  - OLAP access pattern is 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(querying when the need arises)
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

To summarize ...

- OLTP Systems are used to *"run"* a business

- The Data Warehouse helps to *"optimize"* the business

Design of a Data Warehouse: A Business Analysis Framework

- The basic steps involved in the design process of data warehouse mainly involves business analysis framework which give clear understanding of what can a business analysts gain from having a data warehouse?
  - Some of the gains may include:
    - Provide a competitive advantage by presenting relevant information
    - Enhance business productivity as it enable to quickly and efficiently gather information that accurately describe the organization
    - Facilitate customer relationship management by providing consistent view of customers and items across all lines of business, all departments and all markets

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 allows data to be modeled and viewed ***in multiple dimensions***

- A data cube is modeled around a central team like sales which is maintained by a table called **fact table**.

- Dimensions are the perspective of entities with respect to which an organization wants to keep records

From Tables and Spreadsheets to Data Cubes

- **For example:**
  - Records of **store sales** can be maintained with respect to the dimension time(day, week, month, quarter, year), item(item_name, brand, type), branch, and location
  - **Fact table** contains measures (such as dollars_sold, unit sold, amount_budgeted) and keys to each of the related dimension tables where
    - Dollar sold refers to the amount of money sold
    - Unit sold refers to the number of items sold
    - Amount budgeted refers to the amount of money planned

From Tables and Spreadsheets to Data Cubes

- Consider the amount of money collected in Birr at "Amen Mini Market" at different branches
- Branch = Woliso Campus

		Time						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Item	Chocolate	20	19	21	34	30	35	28
	Alcoholic Drink	80	74	45	87	90	99	91
	canned foods	67	68	63	55	64	52	55
	Soft drink	44	60	63	54	64	45	54
	Baby diaper	45	54	55	65	65	54	67

From Tables and Spreadsheets to Data Cubes

- Branch = Bus Station

		Time						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Item	Chocolate	43	45	34	78	54	34	19
	Alcoholic Drink	45	43	26	33	54	71	31
	canned foods	22	76	34	34	91	42	21
	Soft drink	41	53	94	54	29	61	42
	Baby diaper	76	34	89	67	18	27	53

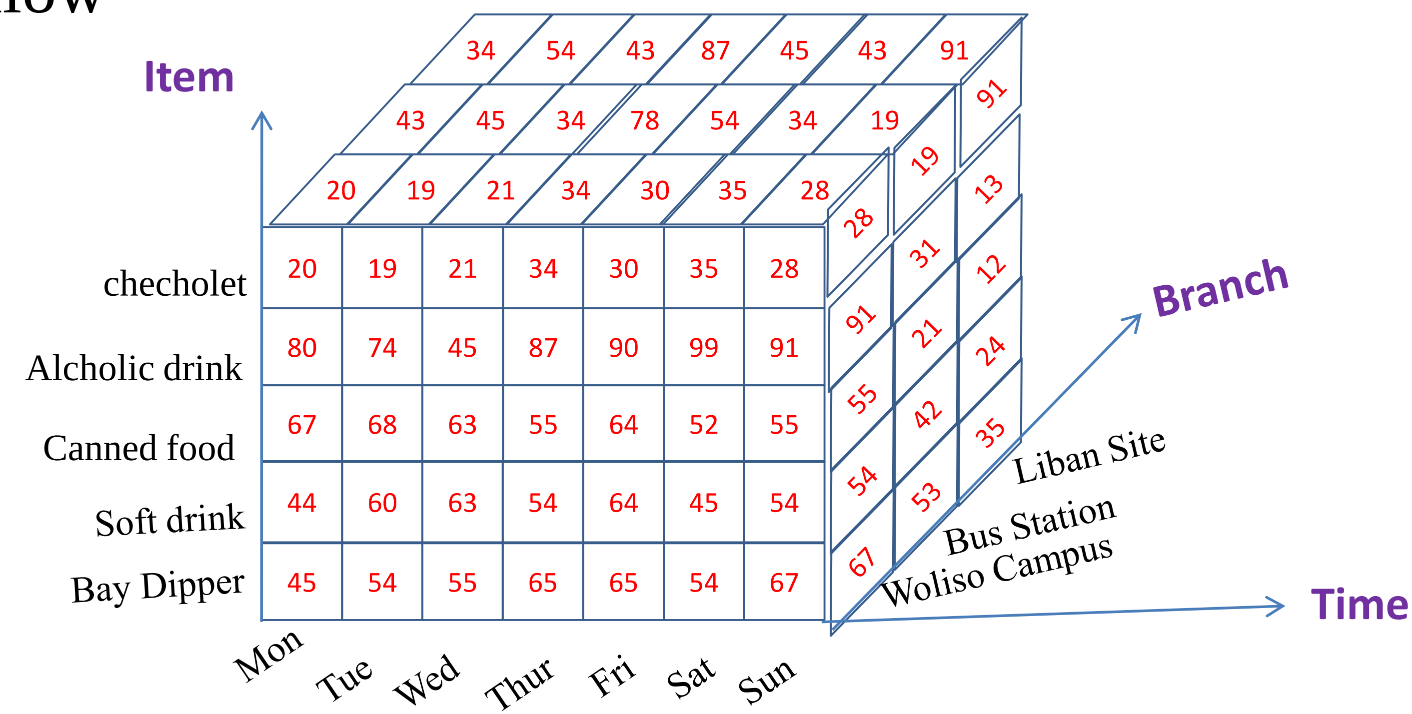
From Tables and Spreadsheets to Data Cubes

- Branch=Liban

		Time						
		Mon	Tue	Wed	Thu	Fri	Sat	Sun
Item	Chocolate	34	54	43	87	45	43	91
	Alcoholic Drink	54	34	62	33	45	27	13
	canned foods	22	67	43	43	19	24	12
	Soft drink	14	35	49	45	92	16	24
	Baby diper	67	43	98	76	81	72	35

From Tables and Spreadsheets to Data Cubes

- This data can be seen at various granularity such as amount of money per day, per week, for coca cola, sprite, biscuits, etc.
- The above three tables can be seen as sub-cuboids of the cube shown bellow



From Tables and Spreadsheets to Data Cubes

- As data warehouse can be seen from various views
- In data warehousing literature, an **n** dimensional (n-D) cube is called a base cuboid.
- Base cuboid shows some information about every attribute at most refined granularity
- The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid.
- This shows the most summarized information which is free from any attribute

From Tables and Spreadsheets to Data Cubes

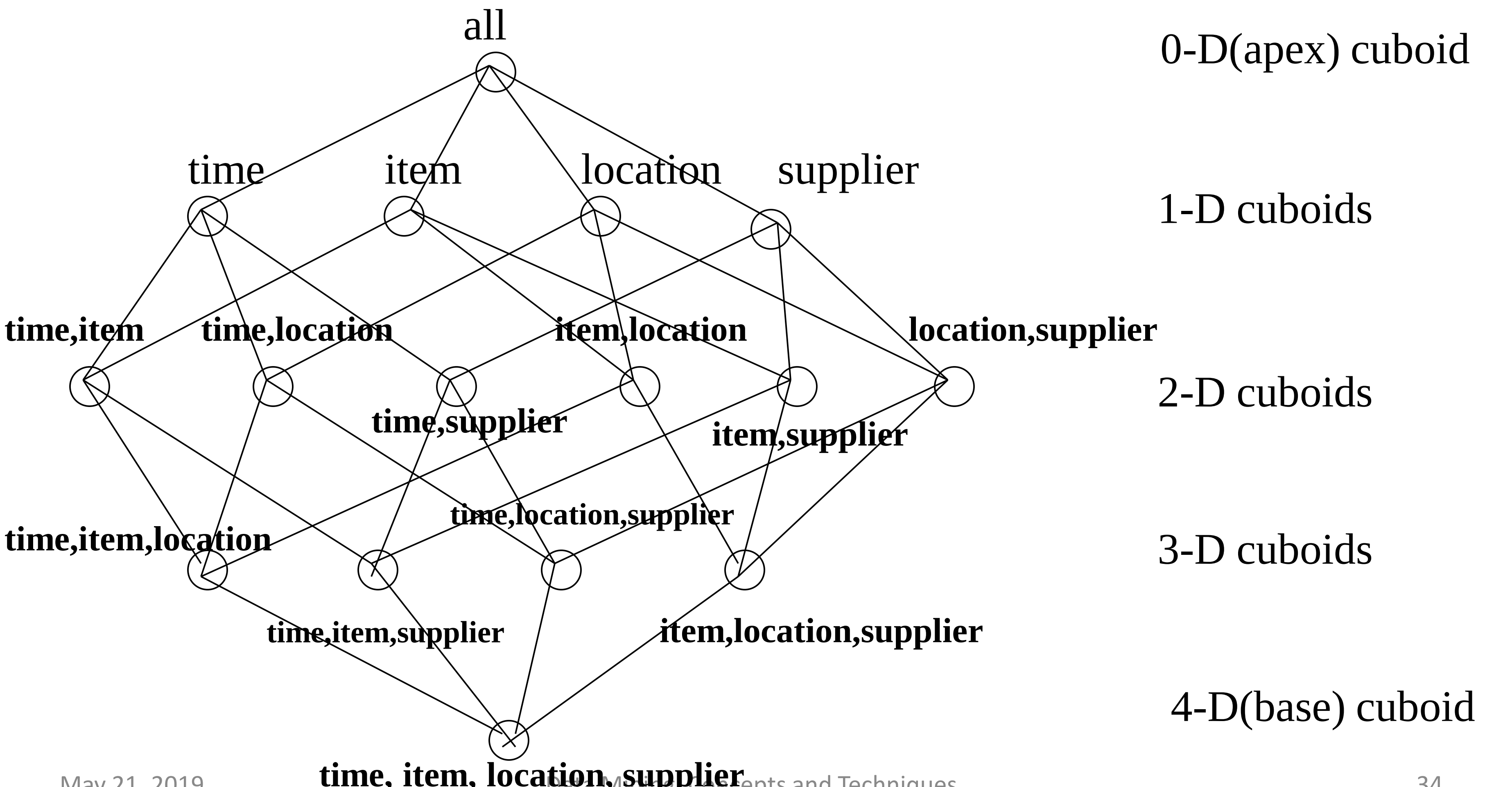
- Lattice is formed by systematically arranging the possible cuboid and their relationship

- The lattice of cuboids forms a data cube.

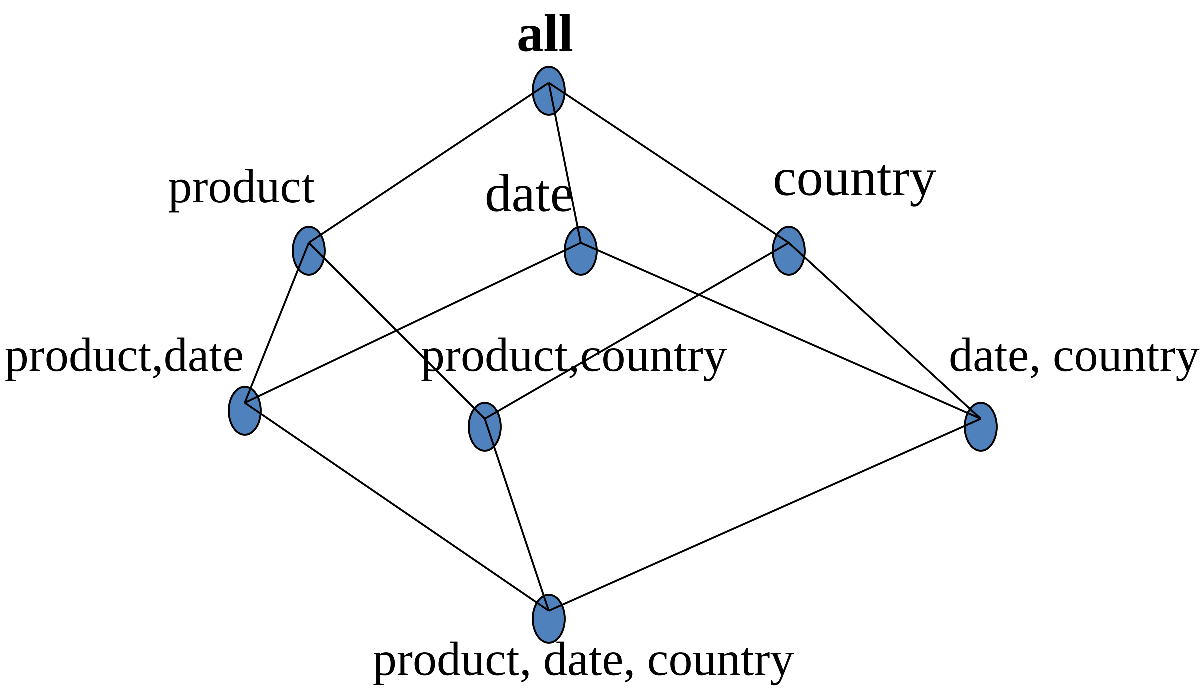
- Example of a lattice with four dimension *(time, item, location, supplier)*

- The fact and dimension table model will be discussed soon

Cube: A Lattice of Cuboids



Cuboids Corresponding to the Cube



0-D(apex) cuboid

1-D cuboids

2-D cuboids

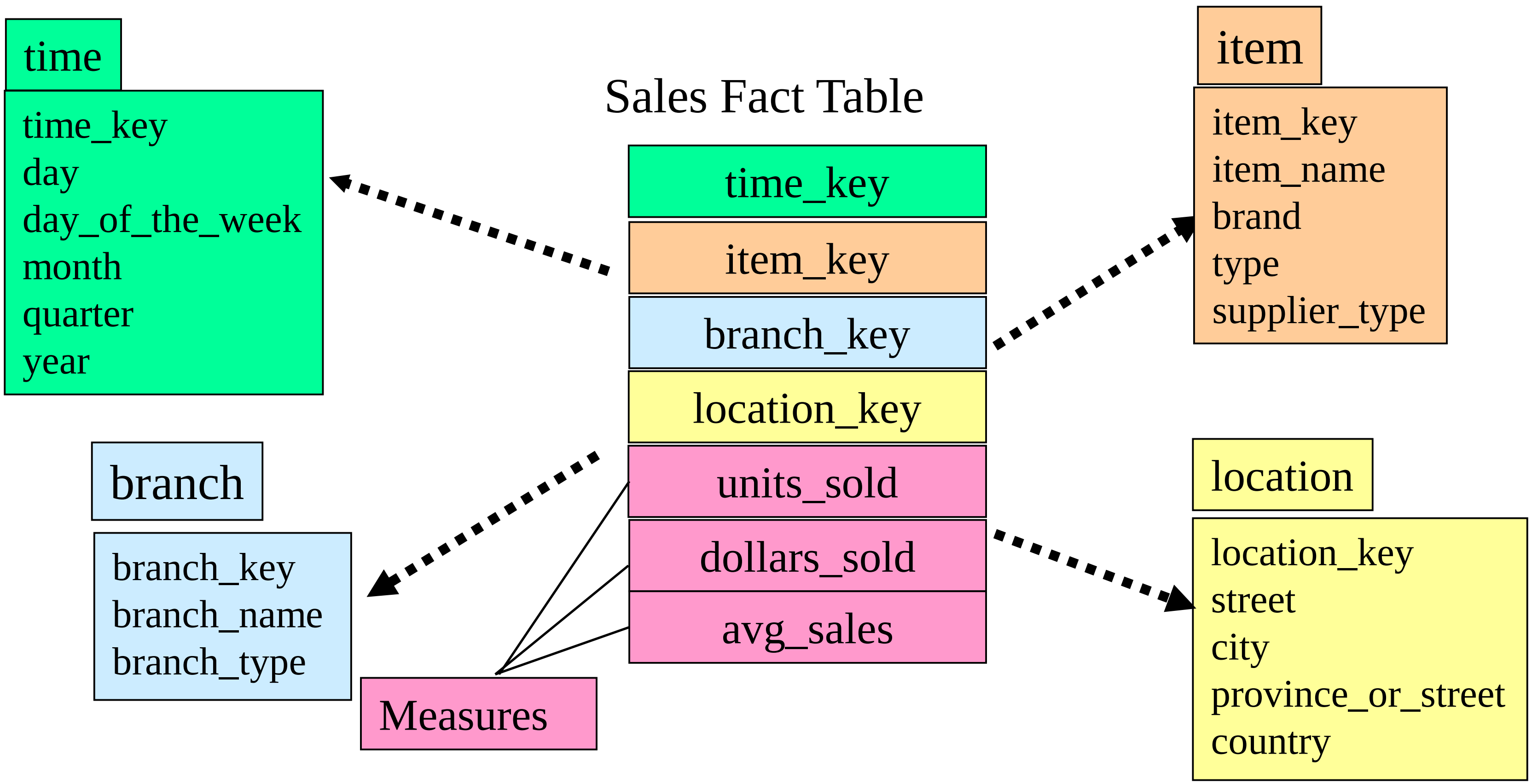
3-D(base) cuboid

Conceptual Modeling of Data Warehouses

- Modeling data warehouses: dimensions & measures
  - Three types of Modeling
    - Star Schema
    - Snowflake Schema
    - Fact constellations

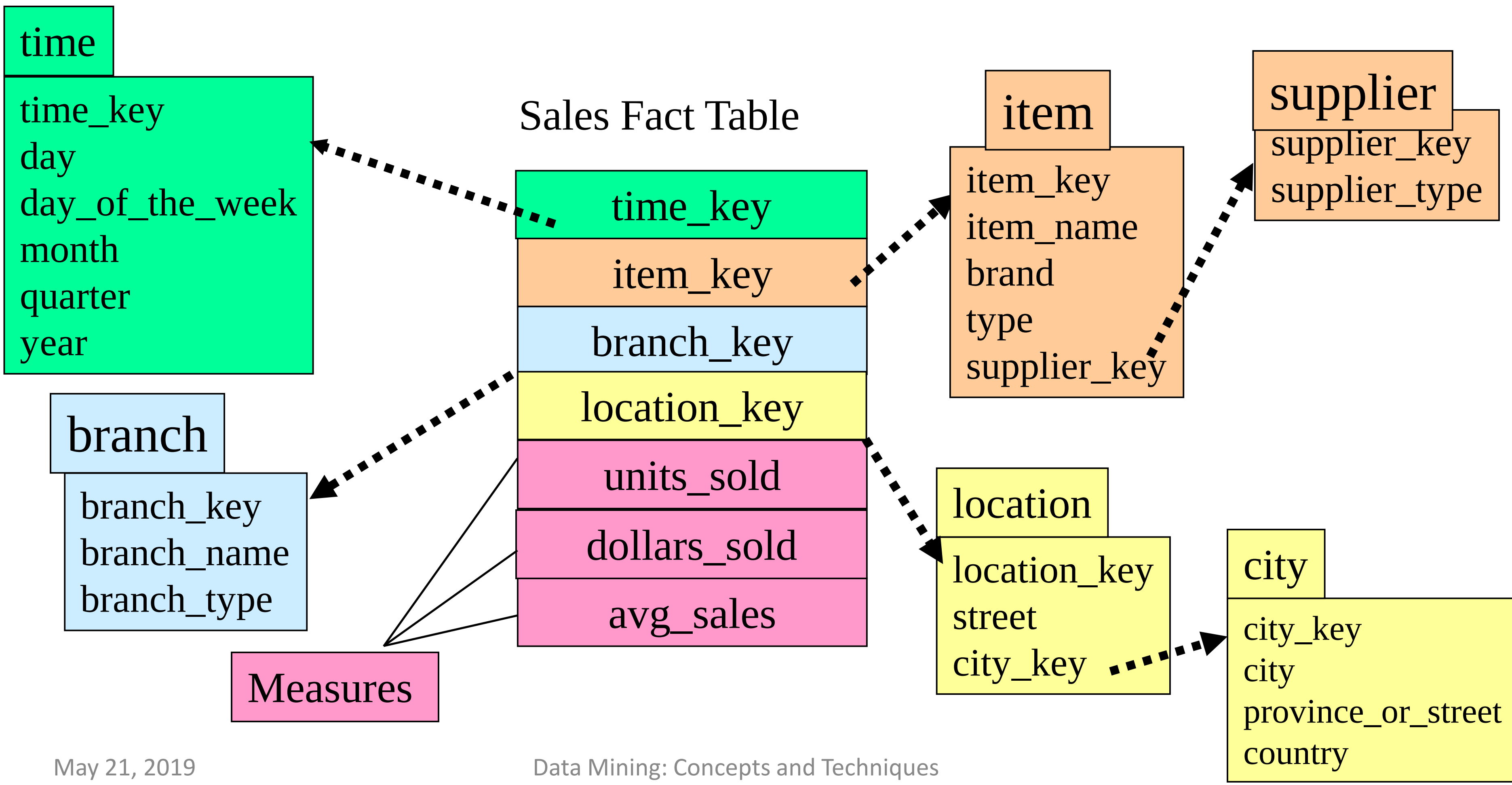
Example of Star Schema

Star schema: A fact table in the middle connected to a set of dimension tables



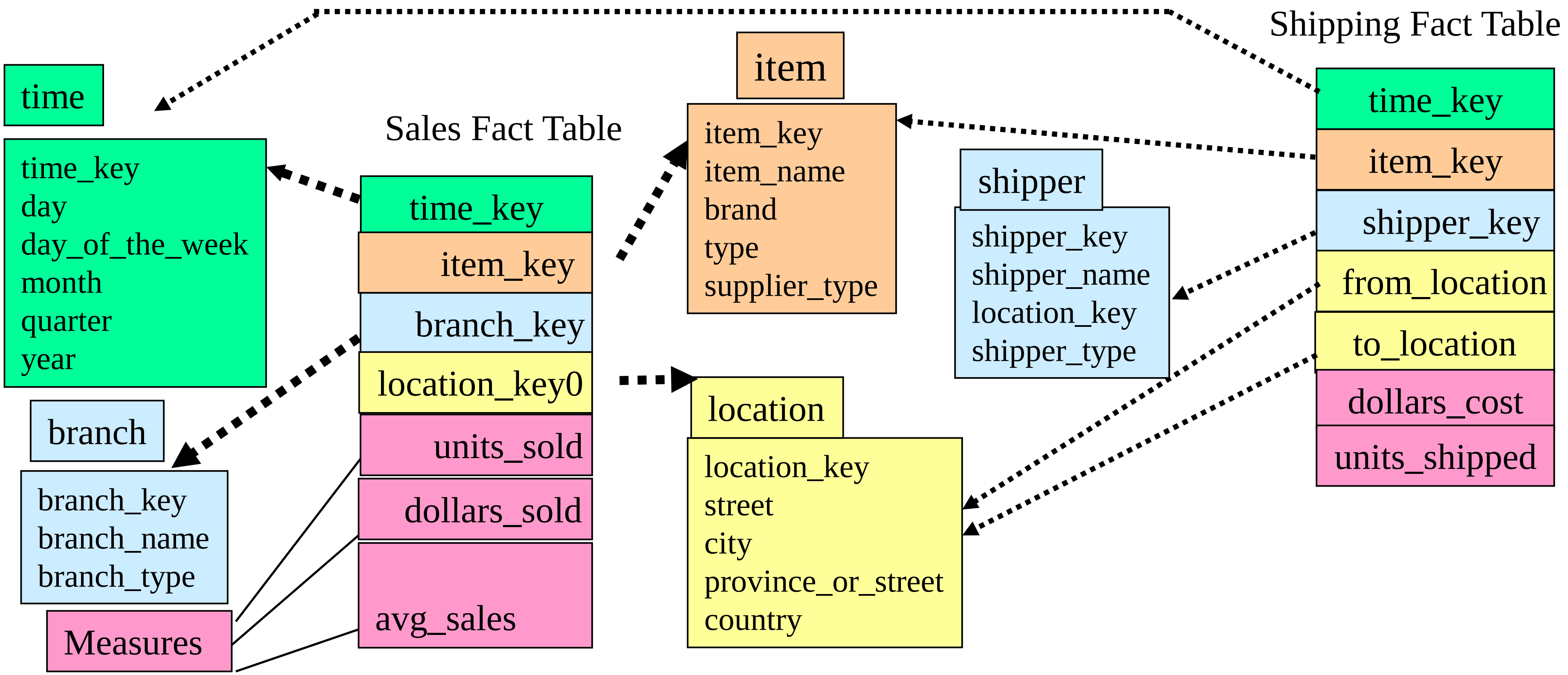
Example of Snowflake Schema

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



Example of Fact Constellation

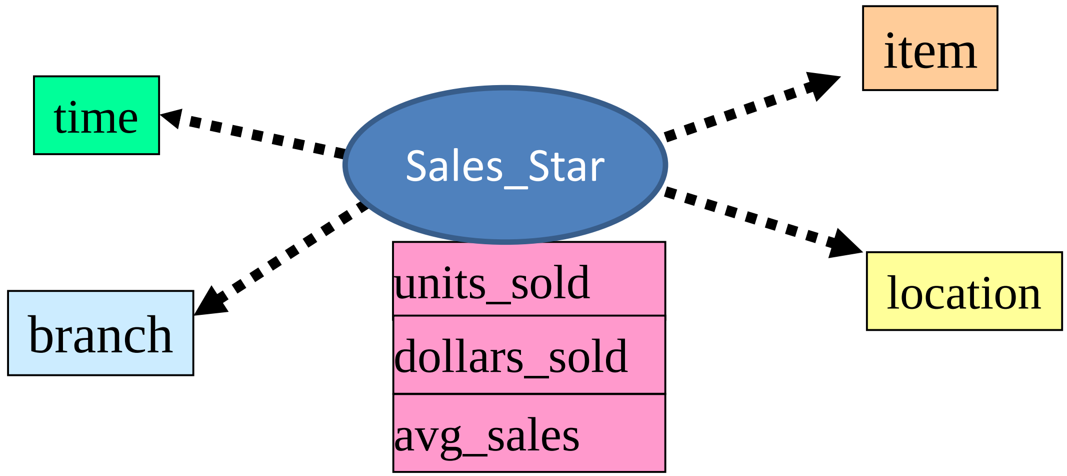
Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation



A Data Mining Query Language, DMQL: Language Primitives

- **Cube Definition (Fact Table)**

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



Example

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

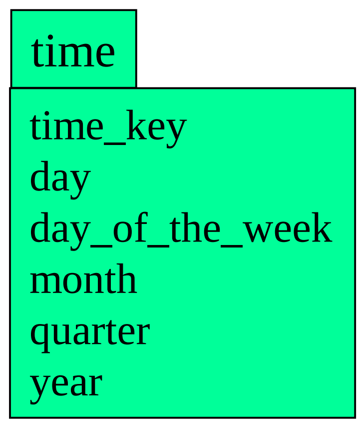
dollars_sold = sum(sales_in_dollars), avg_sales = avg(sales_in_dollars), units_sold = count(*)

Defining a Star Schema in DMQL

- **Dimension Definition (Dimension Table)**

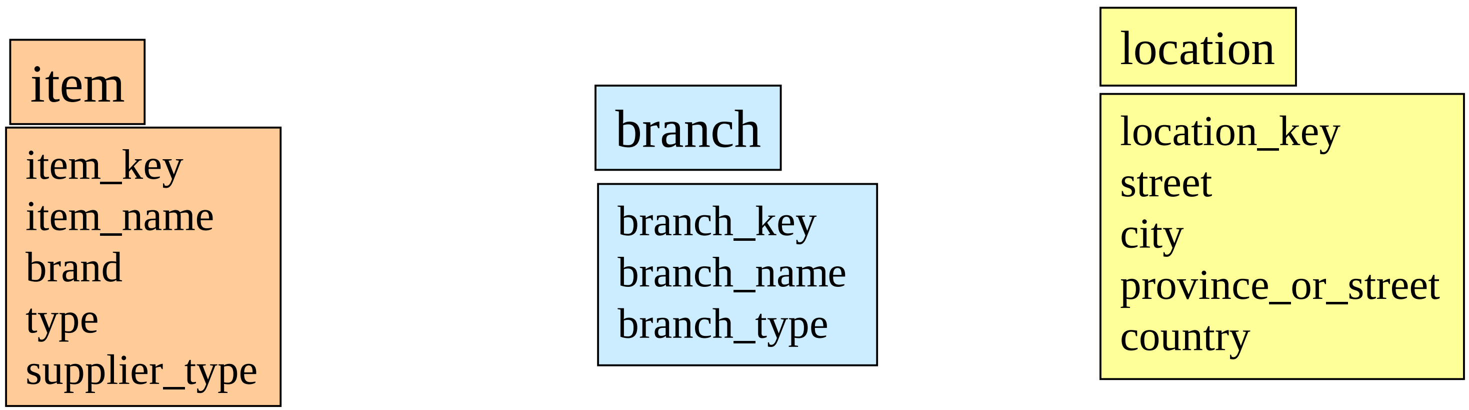
  define dimension <dimension_name> as
  (<attribute_or_subdimension_list>)

Example: The following defines all the dimensions in the sales_star cube



define dimension time as (time_key, day, day_of_week, month, quarter, year)

Defining a Star Schema in DMQL



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)

Measures

- The data cube space is a set of points where each point is dimension-value pair
- Dimension refers to the technique that uniquely define the point where as the value refer to a numerical value (s) at that point
- Example of *dimension*

  ***item** = "coca cola", **location**="Woliso Campus", **supplier**="East Africa Bottling"*

- Example of *value*

  ***unit-sold**= 56, **dollar-sold**=$100000, **average-sold**=$350*

- An aggregate function is applied on a set of point values to analyze the data cube to make various decisions.
- Note the above value may be aggregate of all the time (say for a year)
- The result of the aggregate function is said to be a **measure**

Measures: Three Categories

- An aggregate value is a value produced from sub set of the entire data

- Various measure functions can be applied onto the data cube and these measures can be categorized into three as:
  - **Distributive**
  - **Algebraic**
  - **Holistic**

Measures: Three Categories

- Distributive: A function is said to be distributive if the result derived by applying the function on all the data is the same as the value derived by applying the same/another function on to the aggregate values derived from the subset of data formed from n disjoint partition of the entire data points
- F is distributive iff $F(D) = F(F(D_1), F(D_2), \ldots, F(D_n))$ or

  $F(D) = G(F(D_1), F(D_2), \ldots, F(D_n))$ and

  $\varnothing = D_1 \cap D_2 \cap D_3 \ . \ . \ . \cap D_n$ and

  $D = D_1 \cup D_2 \cup D_3 \ . \ . \ . \cup D_n$
- Sample distributive functions
  - count(), sum(), min(), max().

Measures: Three Categories

- If **F** is **count** the **G** is **sum** as shown bellow

  count(v1, v2, v3,v4,v5,v6)

  = sum(count(v1,v2,v3), count(v4,v5,v6))

  =sum(3,3)=6

- If **F** is **Sum** the **G** is **sum** as shown bellow

  sum(v1, v2, v3,v4,v5,v6)

  = sum(sum(v1,v2,v3), sum(v4,v5,v6))

  =sum(v7,v8)

- If **F** is **Max** the **G** is **Max** as shown bellow

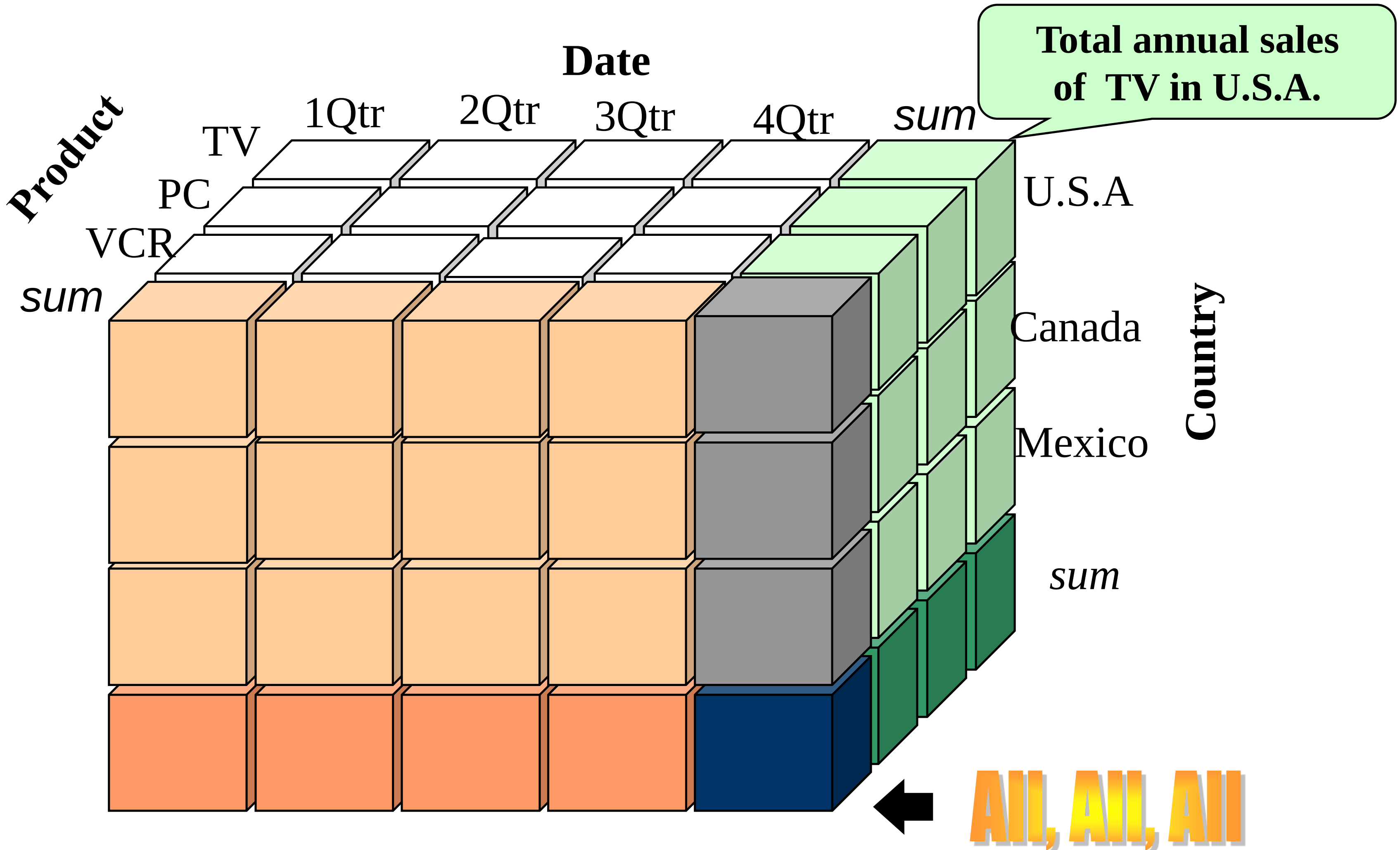
  Max(v1, v2, v3,v4,v5,v6)

  = max(max(v1,v2,v3), max(v4,v5,v6))

  =max(v7,v8)

Measures: Three Categories

- A Sample Data Cube with the distributive aggregate function (SUM)



Measures: Three Categories

- Algebraic: A function is said to be Algebraic if it can be computed as a an algebraic function of M arguments in which each of the arguments are obtained by applying distributive aggregate function.
- F is said to be algebraic iff

$$F(D) = H(G(D_1), G(D_2), \ldots, G(D_n)) \text{ where}$$

G is distributive function,

H is Algebraic function,

$\emptyset = D_1 \cap D_2 \cap D_3 \; . \; . \; . \cap D_n$ and

$D = D_1 \cup D_2 \cup D_3 \; . \; . \; . \cup D_n$

Measures: Three Categories

- Avg(.)= H(sum(.),count(.)) where H(x, y) = x/y
- Note: sum and count are distributive aggregate function and H is algebraic function that apply division on its two arguments
- Some more algebraic functions includes,
  - weighed Average $= \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$
  - Standard Devation= $\frac{1}{n}\sum_{i=1}^{n} x_i - \left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2$
  - variance=$\sqrt{standard\ devation}$
  - Min_N which finds the first N minimum
  - Max_N which finds the first N maximum

(Max_N from 1000 element is Max_N of Max_N of all the sub data group

Measures: Three Categories

- Holistic: A function F is said to be holistic if there is no constant bound on the storage size needed to describe a sub-aggregate.
- That means, there doesn't exist an algebraic function with M argument that characterizes the computation
  - E.g., median(), mode(), rank().
- There are efficient computation techniques for distributive and algebraic function but not for holistic type aggregate functions