Datawarehouse

Vu Tuyet Trinh
trinhvt@soict.hust.edu.vn

Department of Information Systems
SoICT-HUST

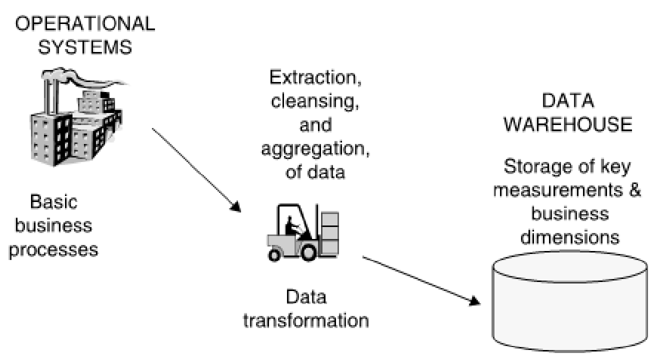
Introduction to Data Warehouse

- Traditional database supports Online Transaction Processing (OLTP)
- Data warehouse support Online Analysis Processing (OLAP)
- Data warehouse contains large amounts of data from multiple sources such as databases, file

Introduction to Data Warehouse

- Operational systems: Run the business on a current basis
  - Example transactions
    - Take an order
    - Generate invoice
    - Reserve a book
    - Process circulation
- Informational systems: Support managerial decision making
  - Example transactions
    - Show me the book which is request most
    - Show me the patron with bad record on late return
    - Show me the problem sale regions of the organization
    - Tell me why we have that problem (drill down to districts and sales offices)

Data Warehouse



Definitions

- Data Warehouse:
  - A Data Warehouse is a subject oriented, integrated, time variant and non volatile collection of data in support of management's decision making process *(W.H.Immon)*

Data Warehouse—Subject-Oriented

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

Data Warehouse—Integrated

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

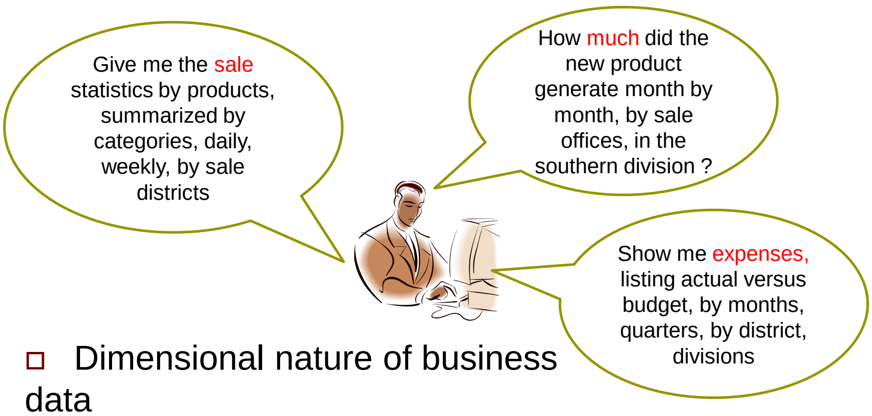
Data Warehouse—Time Variant

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

Data Warehouse—Non-Volatile

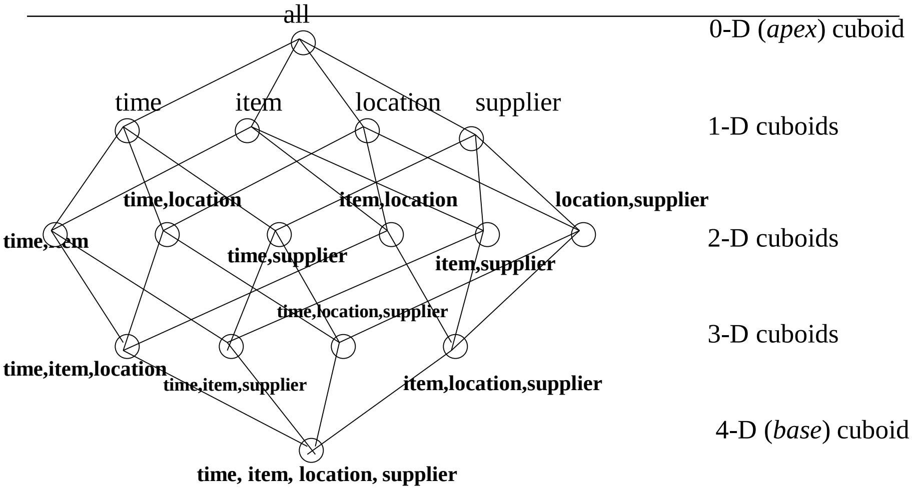
- A physically separate store of data transformed from the operational environment.
- Operational update of data does not occur in the data warehouse environment.
  - Does not require transaction processing, recovery, and concurrency control mechanisms
  - Requires only two operations in data accessing: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

Data Modeling for Data Warehouse

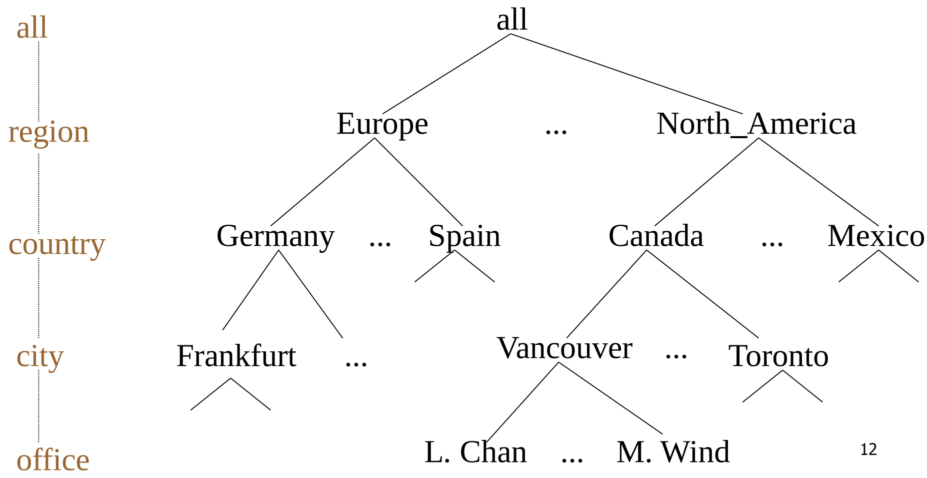


- Dimensional nature of business data
  - Users think in terms of dimensions for decision making
  - Data modelers think of dimensions for modeling process

Cube: A Lattice of Cuboids



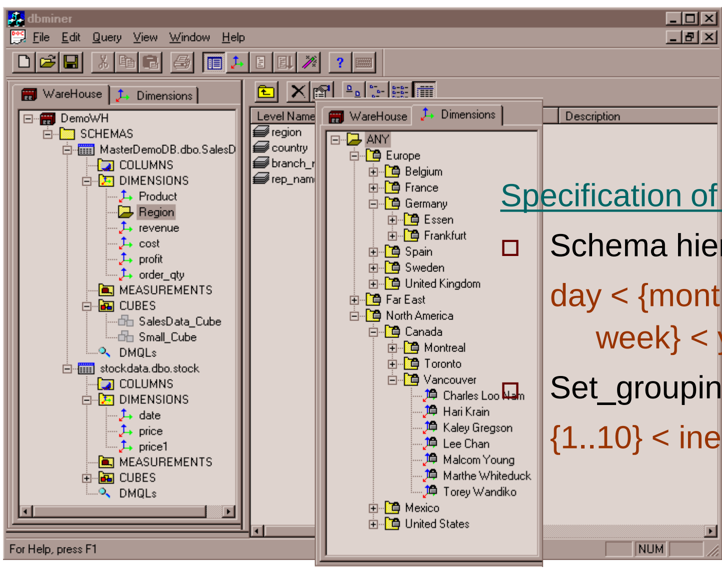
A Concept Hierarchy: **Dimension** (location)



Data Cube Measures: Three Categories

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

View of Warehouses and Hierarchies



Specification of hierarchies

- Schema hierarchy

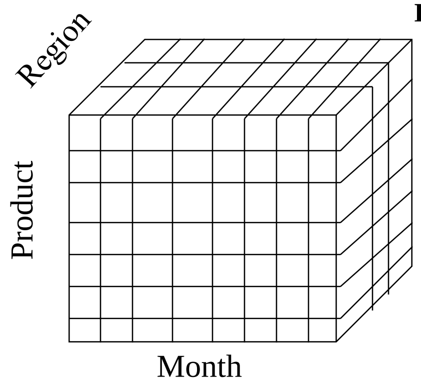
  day < {month < quarter; week} < year

- Set_grouping hierarchy

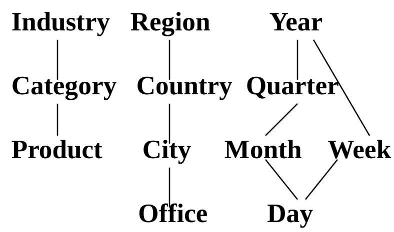
  {1..10} < inexpensive

Multidimensional Data

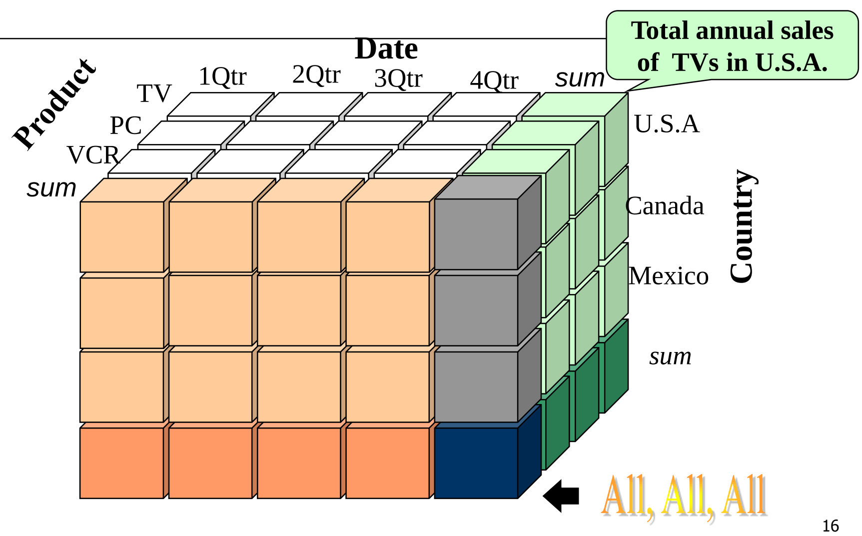
- Sales volume as a function of product, month, and region



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



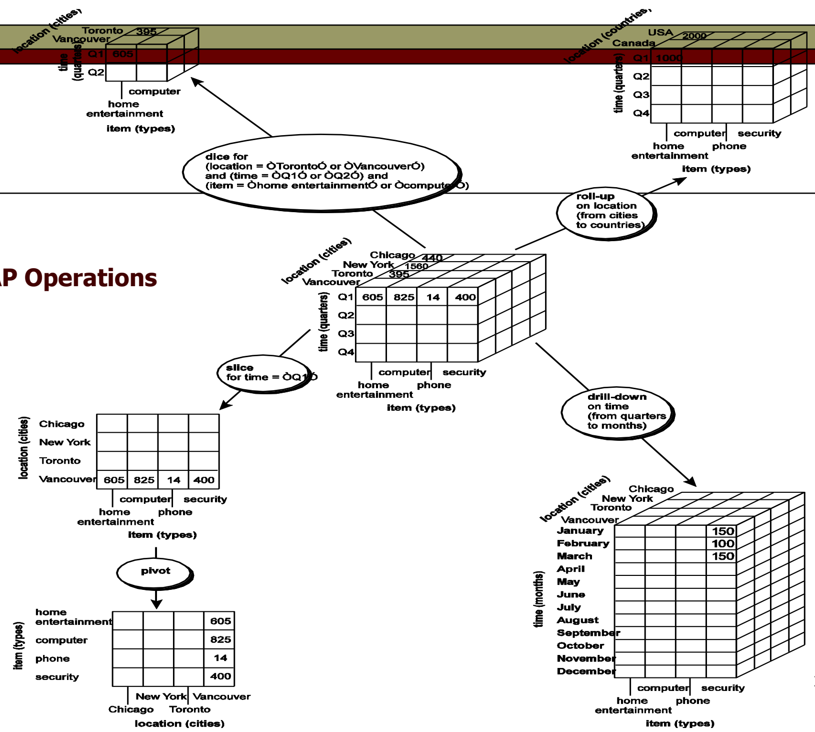
A Sample Data Cube



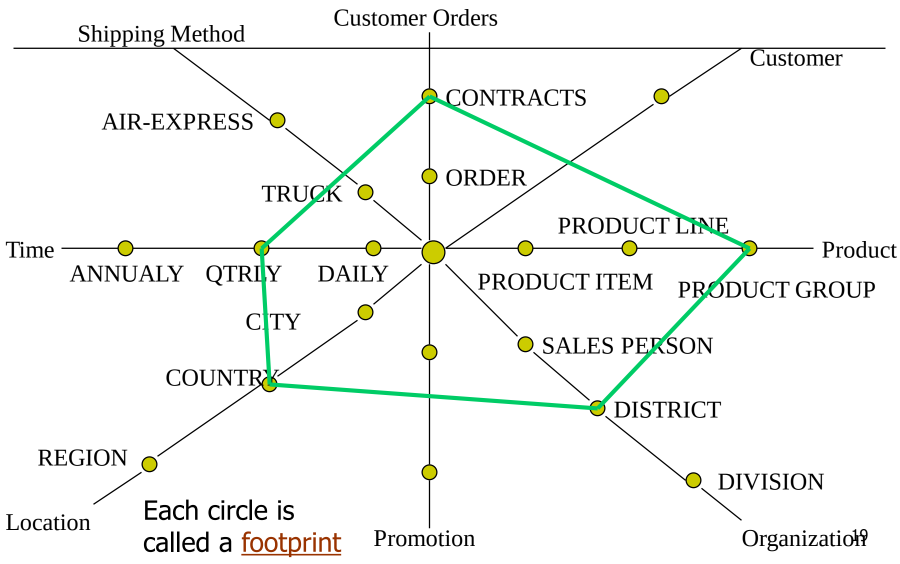
Typical OLAP Operations

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

OLAP Operations



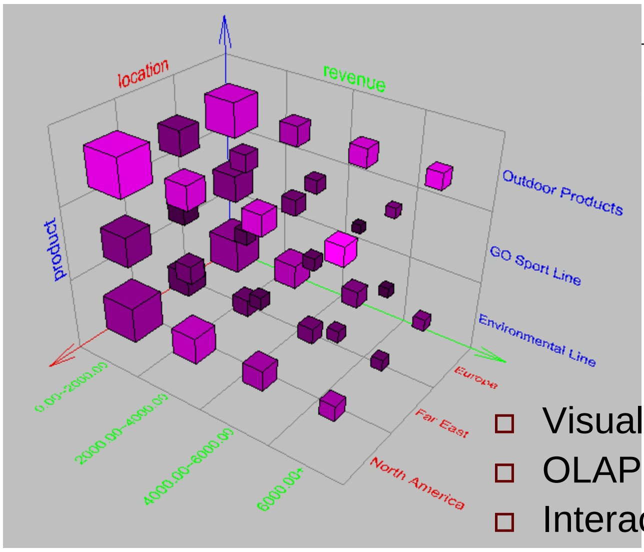
A Star-Net Query Model



Each circle is called a footprint

19

Browsing a Data Cube



- Visualization
- OLAP capabilities
- Interactive manipulation

20

Dimensional Modeling

- Technique for conceptualizing and visualizing data models as a set of measures that are described by common aspects of the business
- ER models describe “entities” and “relationships”
- Dimensional models describe “measures” and “dimensions”
- Basic concepts
  - Facts
  - Dimensions
  - Measures

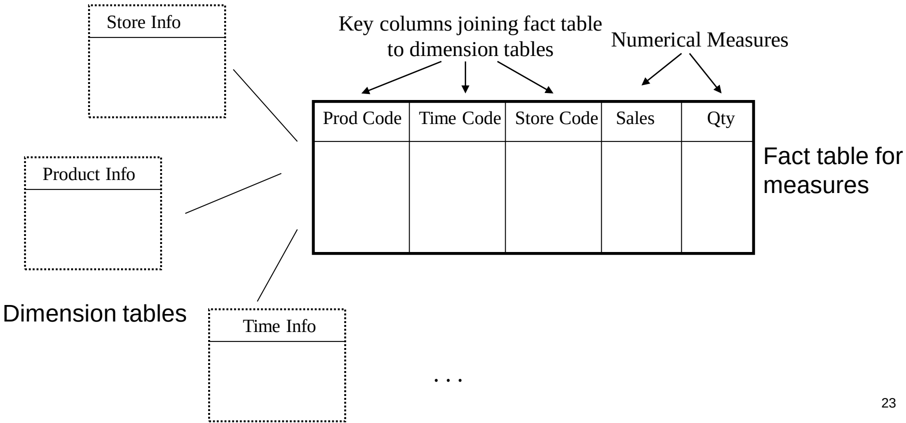
Facts , Dimension

- A fact is a collection of related data items, consisting of measures and context data
- Each fact represents a business item or transaction
- Dimensions are reference information that give context to the fact
- Example: Sales
  - Facts: number of products purchased (unit-sale), price paid for the products (full price),
  - Dimension: order date, product id, sale person , customer age, customer gender

The Multi-Dimensional Model

~~*“Sales by product line over the past six months”*~~

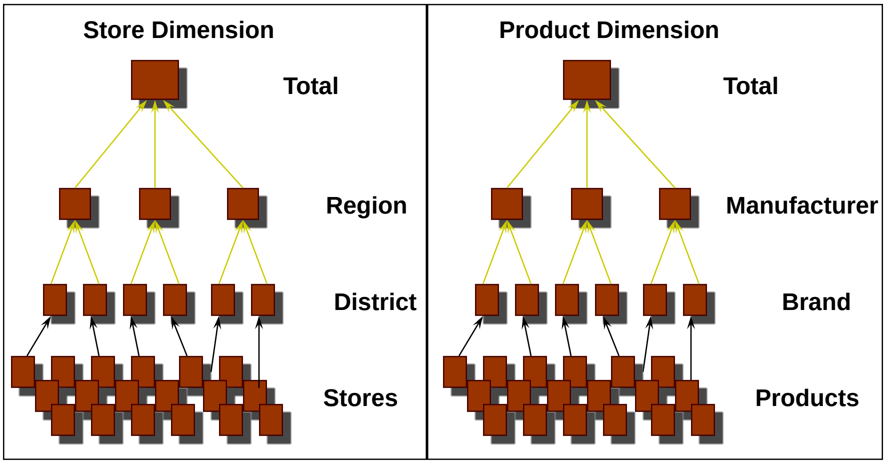
“Sales by store between 1990 and 1995”



Dimensional Modeling

- Dimensions are organized into hierarchies
  - E.g., Time dimension: days → weeks → quarters
  - E.g., Product dimension: product → product line → brand
- Dimensions have attributes

Dimension Hierarchies



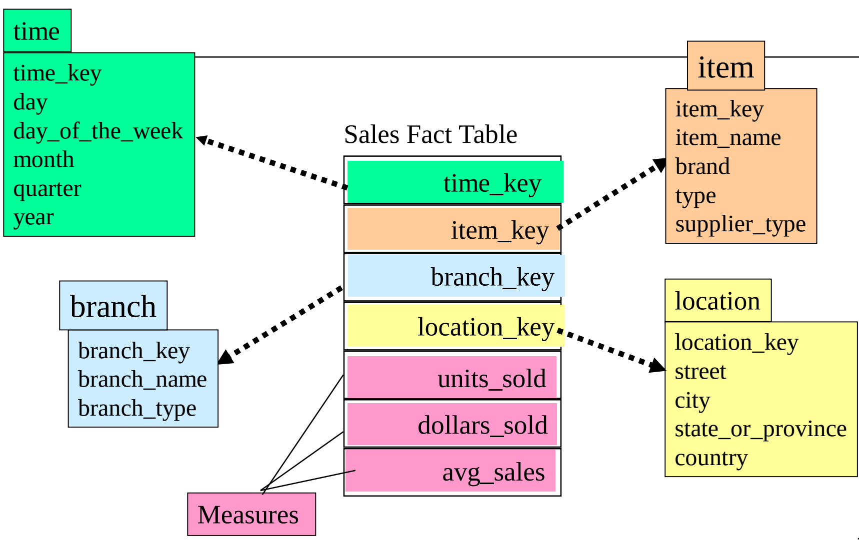
ROLAP: Dimensional Modeling Using Relational DBMS

- Special schema design: *star, snowflake*
- Special indexes: bitmap, multi-table join
- Special tuning: maximize query throughput
- Proven technology (relational model, DBMS), tend to outperform specialized MDDB especially on large data sets
- Products
  - IBM DB2, Oracle, Sybase IQ, RedBrick, Informix

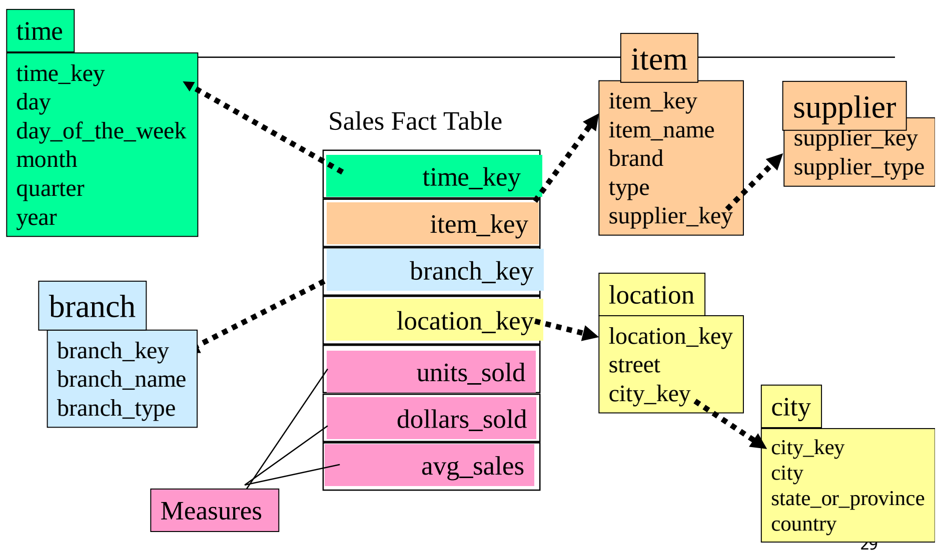
Conceptual Modeling of Data Warehouses

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

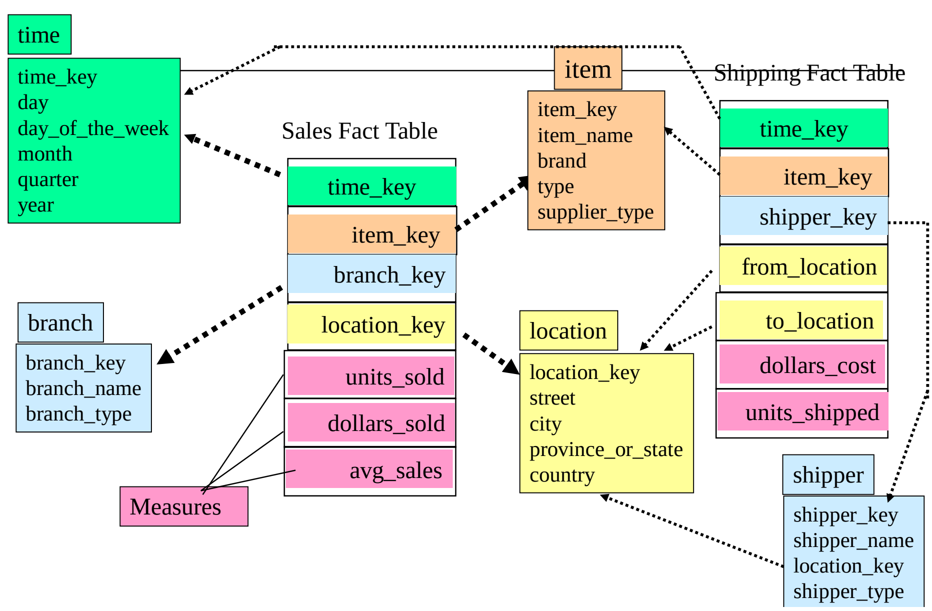
Example of Star Schema



Example of Snowflake Schema



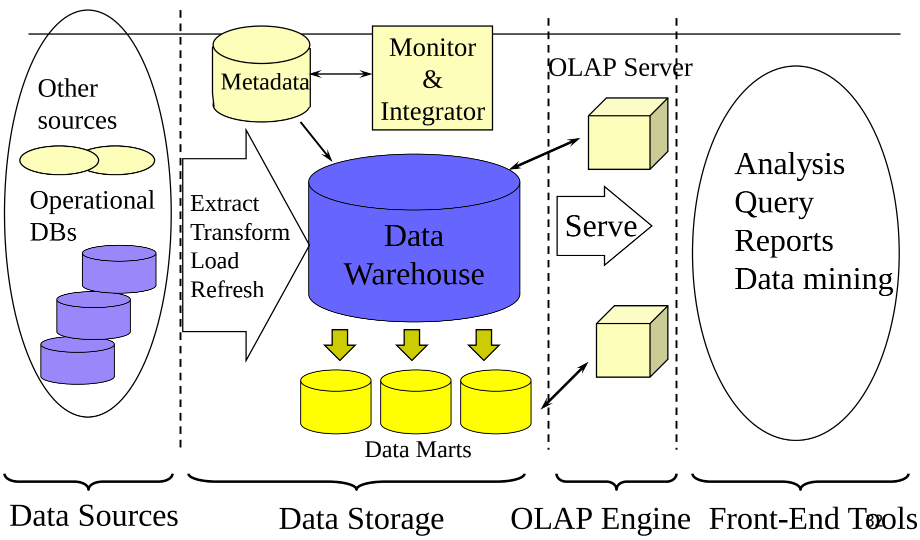
Example of Fact Constellation



MOLAP: Dimensional Modeling Using the Multi Dimensional Model

- MDDB: a special-purpose data model which views data in the form of a data cube
- Facts stored in multi-dimensional arrays
- Dimensions used to index array
- Sometimes on top of relational DB
- Products
  - Pilot, Arbor Essbase, Gentia

Data Warehouse: A Multi-Tiered Architecture



Design of Data Warehouse: A Business Analysis Framework

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

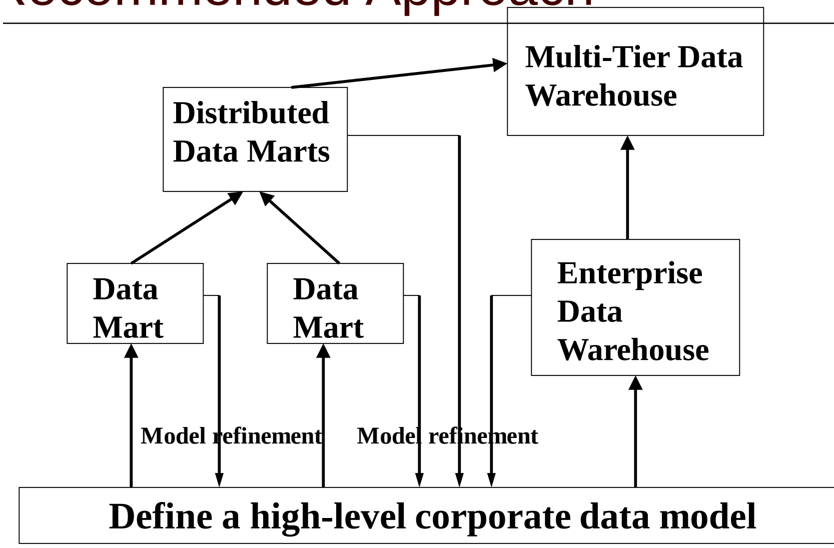
33

Data Warehouse Design Process

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

34

Data Warehouse Development: A Recommended Approach



35

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

36

From On-Line Analytical Processing (OLAP) to On Line Analytical Mining (OLAM)

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

37

Efficient Data Cube Computation

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

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

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

38

Cube Operation

- Cube definition and computation in DMQL

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

    compute cube sales

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

    SELECT item, city, year, SUM (amount)

    FROM SALES

    CUBE BY item, city, year

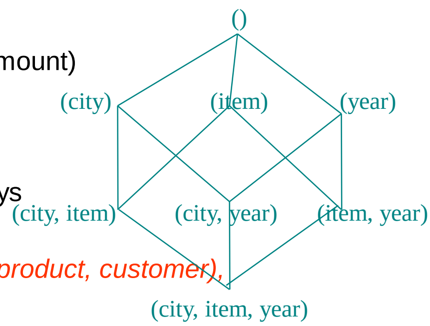
- Need compute the following Group-Bys

    (date, product, customer),

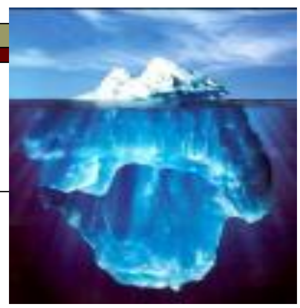
    (date,product),(date, customer), (product, customer),

    (date), (product), (customer)

    ()



Iceberg Cube



- Computing *only* the cuboid cells whose count or other aggregates satisfy a condition with minimum support, e.g.

    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?

Indexing OLAP Data: **Bitmap Index**

- Index on a particular column
- ~~Each value in the column has a bit vector: bit-op is fast~~
- The length of the bit vector: # of records in the base table
- The *i*-th bit is set if the *i*-th row of the base table has the value for the indexed column
- not suitable for high cardinality domains
- A recent bit compression technique, Word-Aligned Hybrid (WAH), makes it work for high cardinality domain as well [Wu, et al. TODS'06]

Base table

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

Index on Region

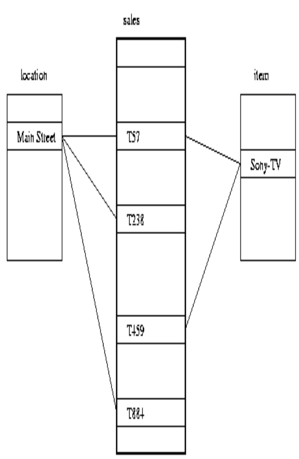
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on Type

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

Indexing OLAP Data: **Join Indices**

- Join index: JI(R-id, S-id) where R (R-id, …) ▷◁ S (S-id, …)
- Traditional indices map the values to a list of record ids
  - It materializes relational join in JI file and speeds up relational join
- In data warehouses, join index relates the values of the dimensions of a start schema to rows in the fact table.
  - E.g. fact table: *Sales* and two dimensions *city* and *product*
    - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
  - Join indices can span multiple dimensions



Efficient Processing OLAP Queries

- **Determine which operations** should be performed on the available cuboids
  - Transform drill, roll, etc. into corresponding SQL and/or OLAP operations, e.g., dice = selection + projection
- **Determine which materialized cuboid(s)** should be selected for OLAP op.
  - Let the query to be processed be on {*brand, province_or_state*} with the condition "*year = 2004*", and there are 4 materialized cuboids available:

    1) {*year, item_name, city*}

    2) {*year, brand, country*}

    3) {*year, brand, province_or_state*}

    4) {*item_name, province_or_state*} where *year = 2004*

    Which should be selected to process the query?
- Explore indexing structures and compressed vs. dense array structs in MOLAP

OLAP Server Architectures

- Relational OLAP (ROLAP)
  - Use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware
  - Include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services
  - Greater scalability
- Multidimensional OLAP (MOLAP)
  - Sparse array-based multidimensional storage engine
  - Fast indexing to pre-computed summarized data
- Hybrid OLAP (HOLAP) (e.g., Microsoft SQLServer)
  - Flexibility, e.g., low level: relational, high-level: array
- Specialized SQL servers (e.g., Redbricks)
  - Specialized support for SQL queries over star/snowflake schemas

45