Data Mining:
Data Warehousing

Data Warehousing and OLAP Technology for Data Mining

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

What is Data Warehouse?

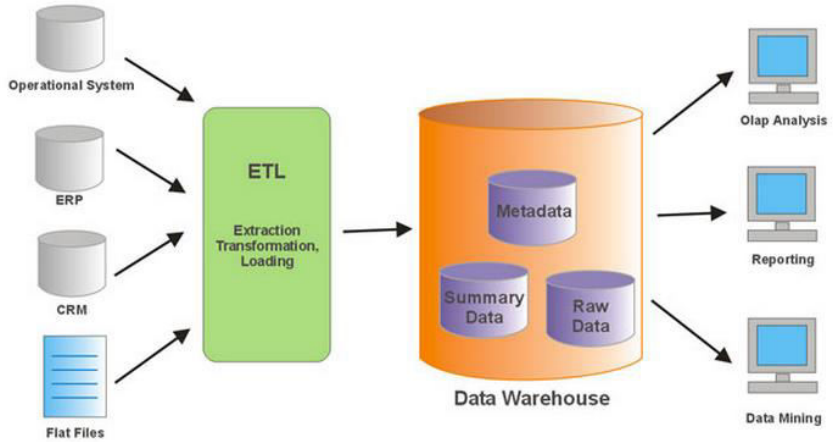
- Defined in many different ways, but not rigorously.
- A decision support database that is maintained separately from the organization's operational database
- Support information processing by providing a solid platform of consolidated, historical data for analysis.

"A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process."—W. H. Inmon

Data 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

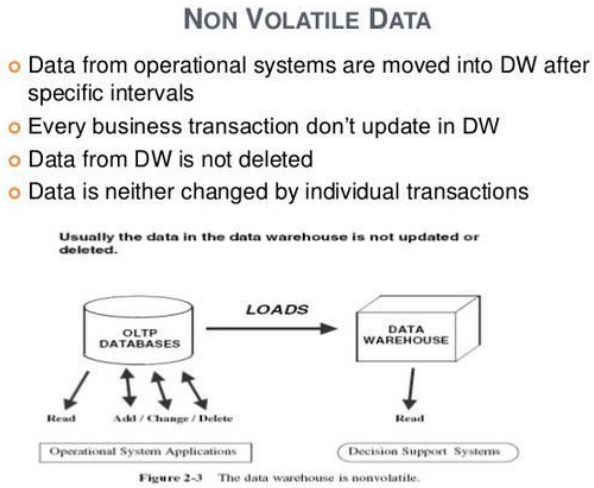


Figure 2-3 The data warehouse is nonvolatile.

- Requires only two operations in data accessing:
- *initial loading of data* and *access of data*.

Data Warehouse vs. Operational DBMS

OLTP (on-line transaction processing)

- Major task of traditional relational DBMS
- Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.

OLAP (on-line analytical processing)

- Major task of data warehouse system
- Data analysis and decision making

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, multi dimensional 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

OLTP vs. OLAP

Why Separate Data Warehouse?

High performance for both systems

- DBMS— tuned for OLTP: access methods, indexing, concurrency control, recovery
- Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation.

Different functions and different data:

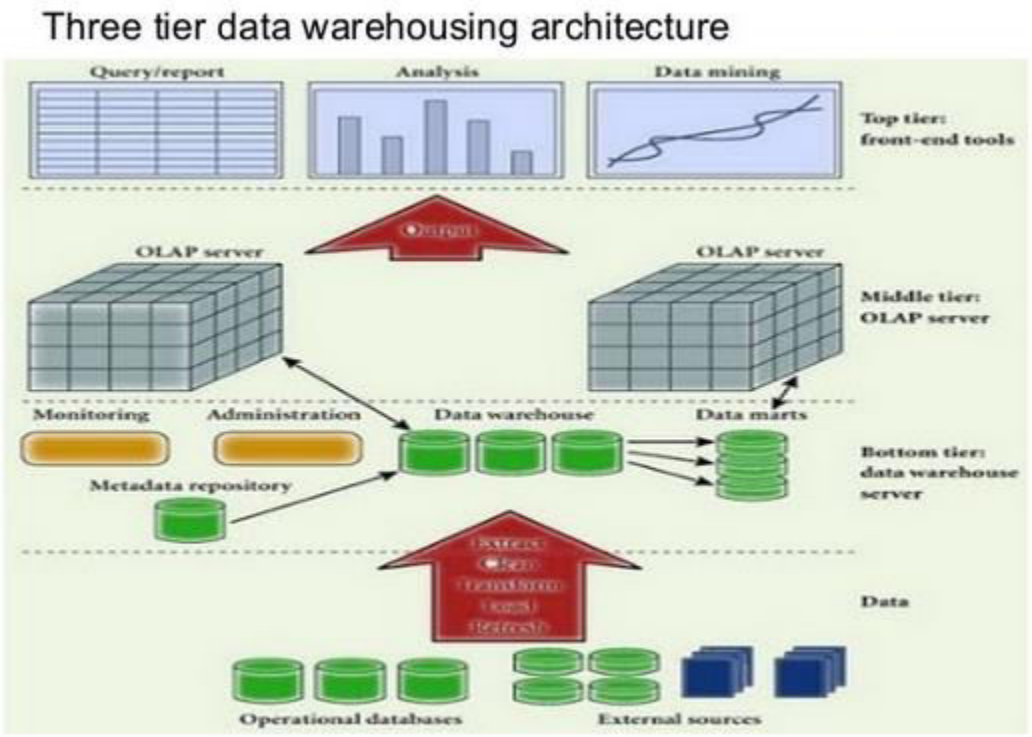
- missing data: Decision support requires historical data which operational DBs do not typically maintain
- data consolidation: DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
- data quality: different sources typically use inconsistent data representations, codes and formats which have to be reconciled

From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube
- A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions
- Dimension tables, such as item (item_name, brand, type), or time(day, week, month, quarter, year)

Fact table contains measures (such as dollars_sold) and keys to each of the related dimension tables

data warehouse a multi-tiered architecture



Tier 1:
Tier 2:
Tier 3:

Multidimensional Data

Sales volume as a function of product, month, and region
A Sample Data Cube: Cuboids Corresponding to the 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)*

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

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 indicies and partitions

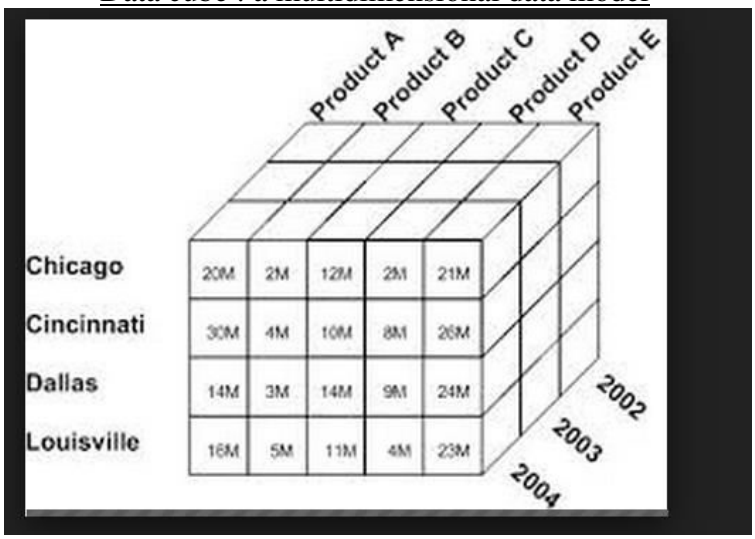
Refresh

propagate the updates from the data sources to the warehouse

Efficient Data Cube Computation

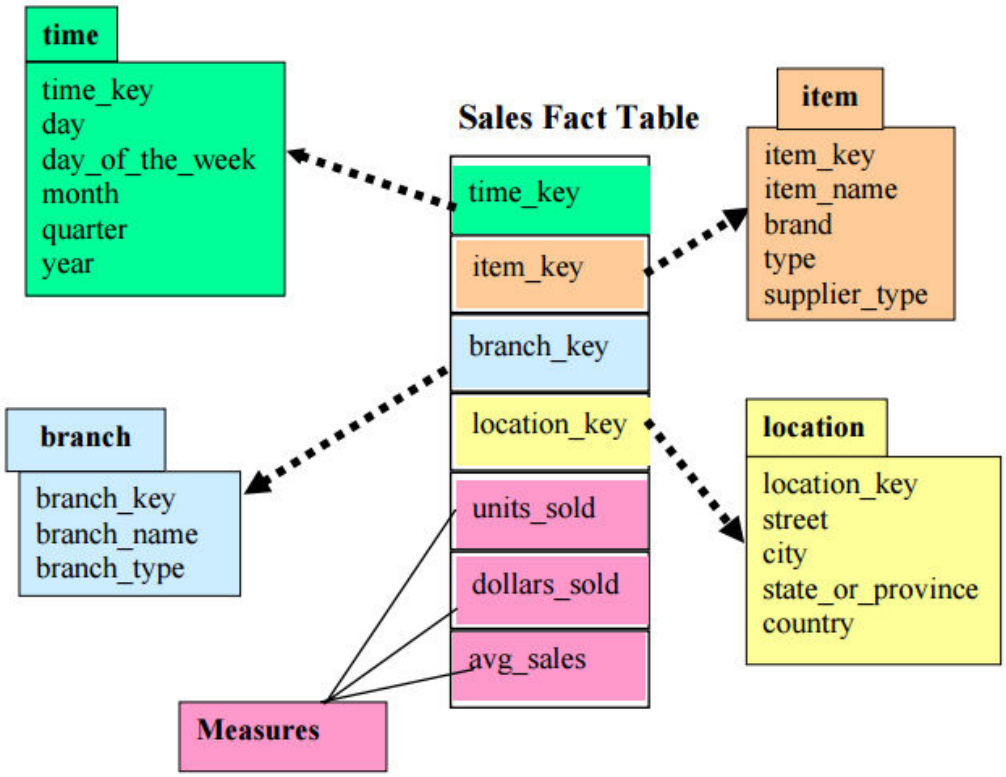
- Data cube can be viewed as a lattice of cuboids
- The bottom-most cuboid is the base cuboid
- The top-most cuboid (apex) contains only one cell
- How many cuboids in an n-dimensional cube with L levels?

Data cube : a multidimensional data model

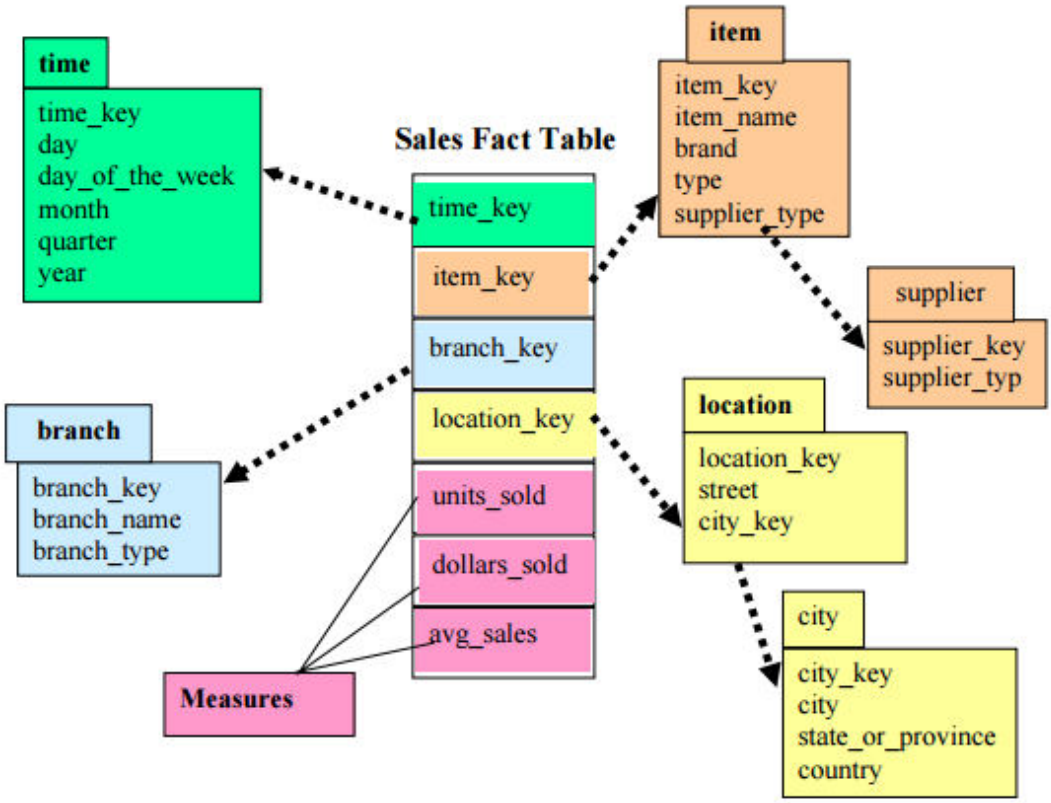


Schemas

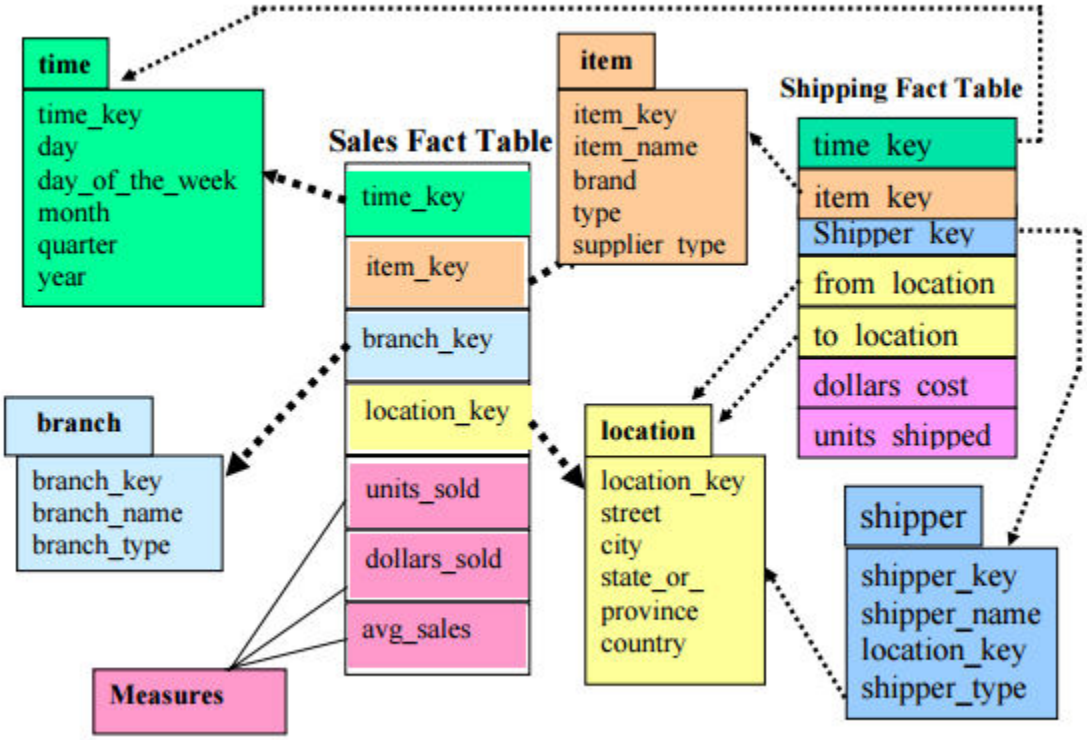
4.1. Star Schema



4.2. Snowflake Schema



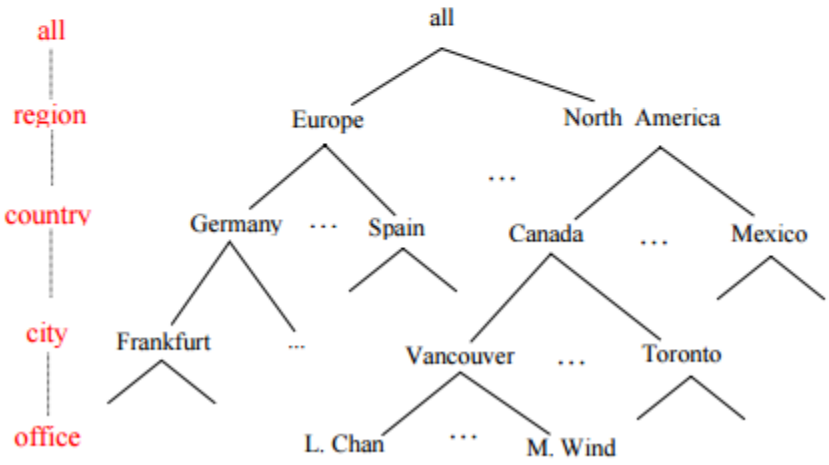
4.3. Fact Constellation



Dimensions: The role of Concept hierarchies

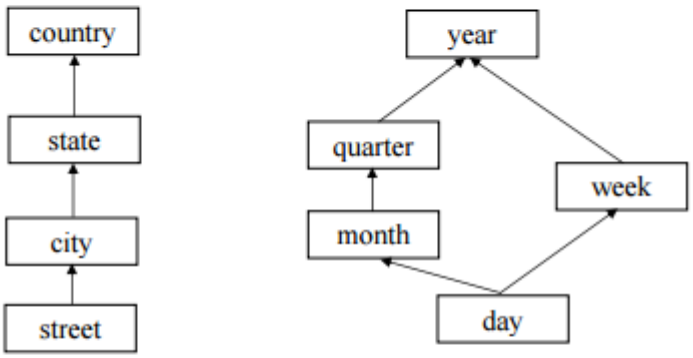
6. A Concept Hierarchy

- A concept hierarchy is an order relation between a set of attributes of a concept or dimension.
- It can be manually (users or experts) or automatically generated (statistical analysis).
- Multidimensional data is usually organized into dimension and each dimension is further defined into a lower level of abstractions defined by concept hierarchies.
- Example: Dimension (location)



- The order can be either partial or total:

Location dimension: Street <city<state<country
Time dimension: Day < {month<quarter ; week} < year



Total order hierarchy Partial order hierarchy

- Set-grouping hierarchy:
  - It is a concept hierarchy among groups of values.
  - Example: {1..10} < inexpensive

7. OLAP Operations in a Multidimensional Data

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

- **Dimensions hierarchical concepts: Product, Location, Time**

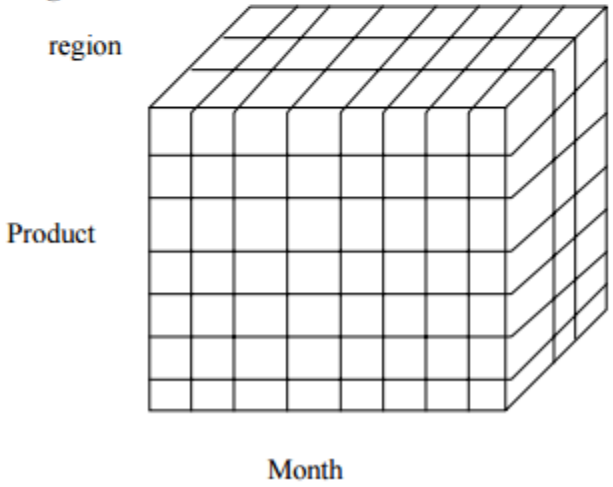
Industry → Category → Product

Region → Country → City → Office

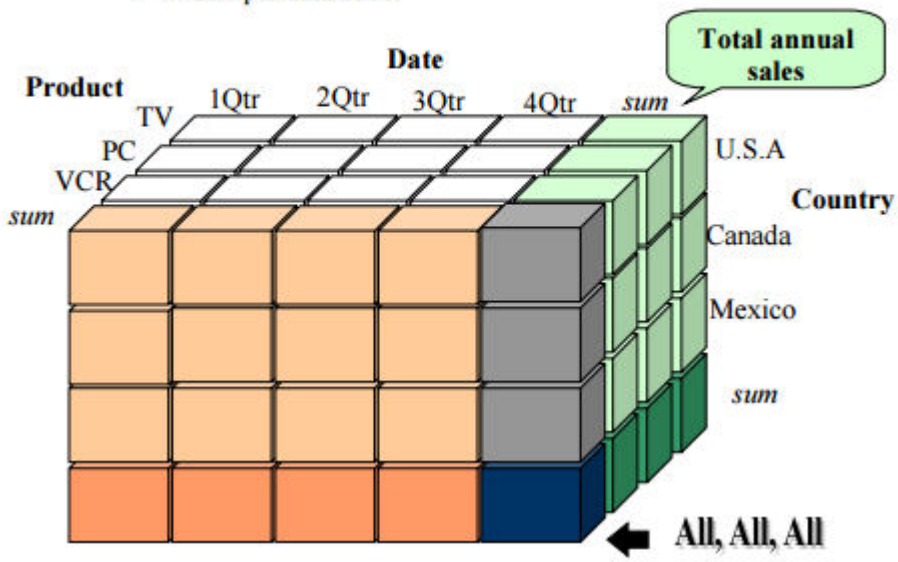
Year → Quarter → Month → Day

Week

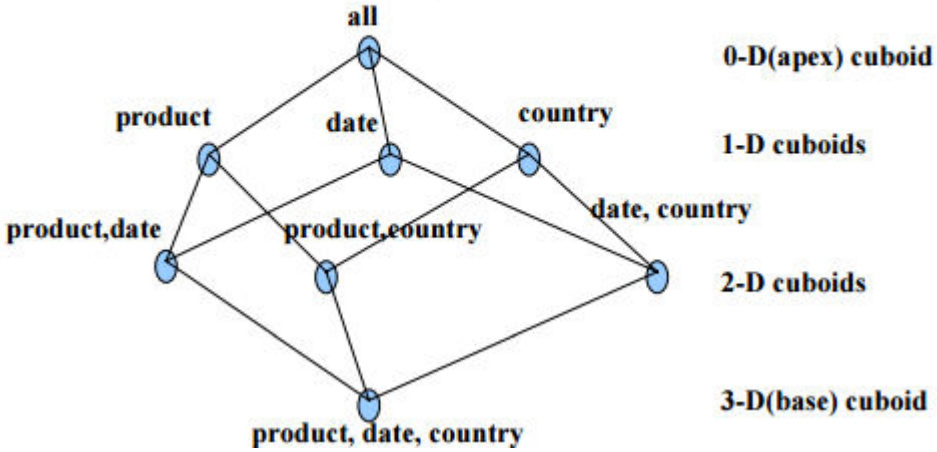
- Sales volume as a function of **product**, **month**, and **region.**



- A Sample data cube:



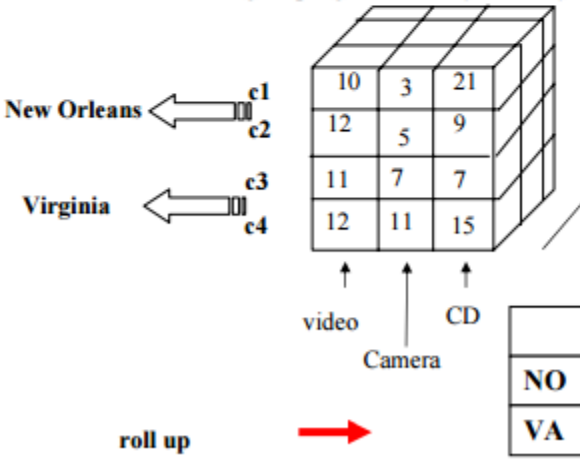
- Cuboids of the sample cube:



8. OLAP Operations

- Objectives:
  - OLAP is a powerful analysis tool:
    - Forecasting
    - Statistical computations,
    - aggregations,
    - etc.

- Roll up (drill-up): summarize data
  - It is performed by climbing up hierarchy of a dimension or by dimension reduction (reduce the cube by one or more dimensions).
  - The roll up operation in the example is based location (roll up on location) is equivalent to grouping the data by country.



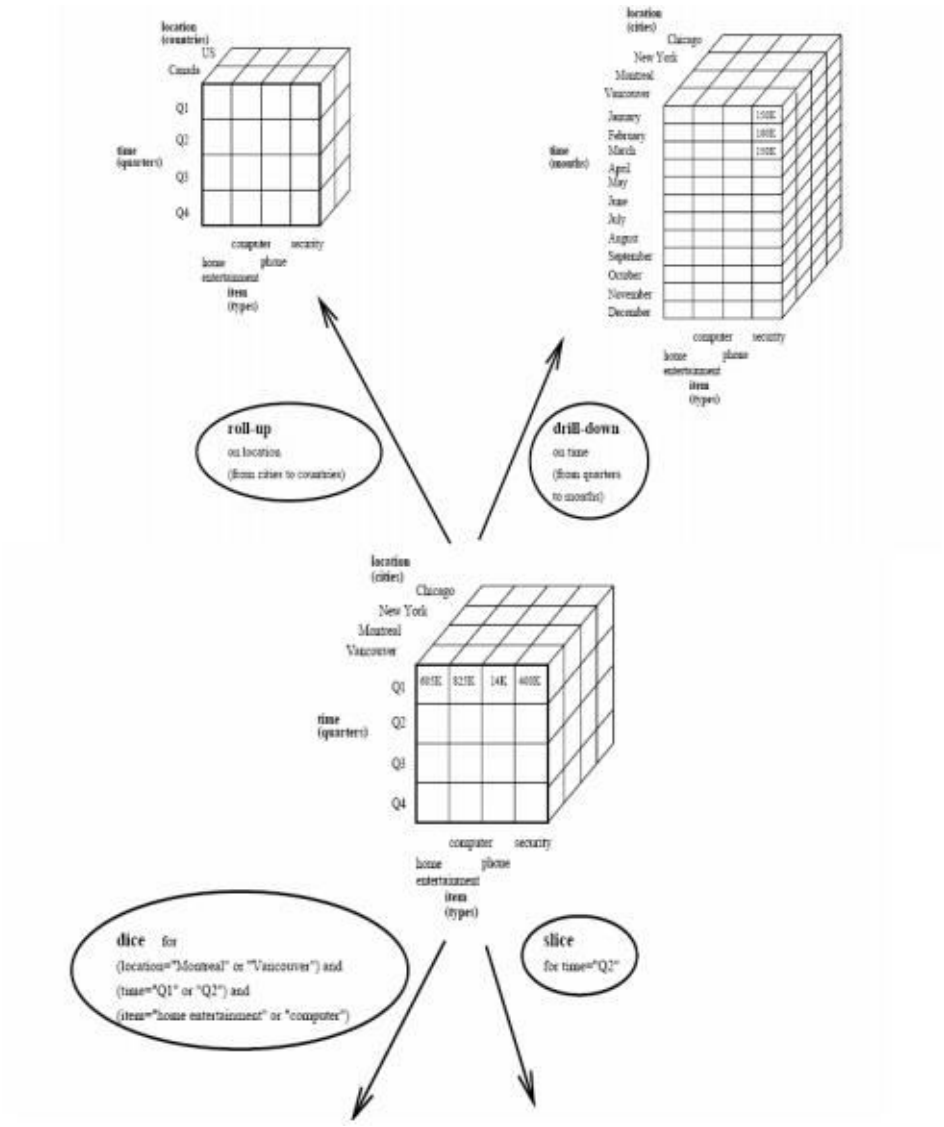
	Video	Camera	CD
NO	22	8	30
VA	23	18	22

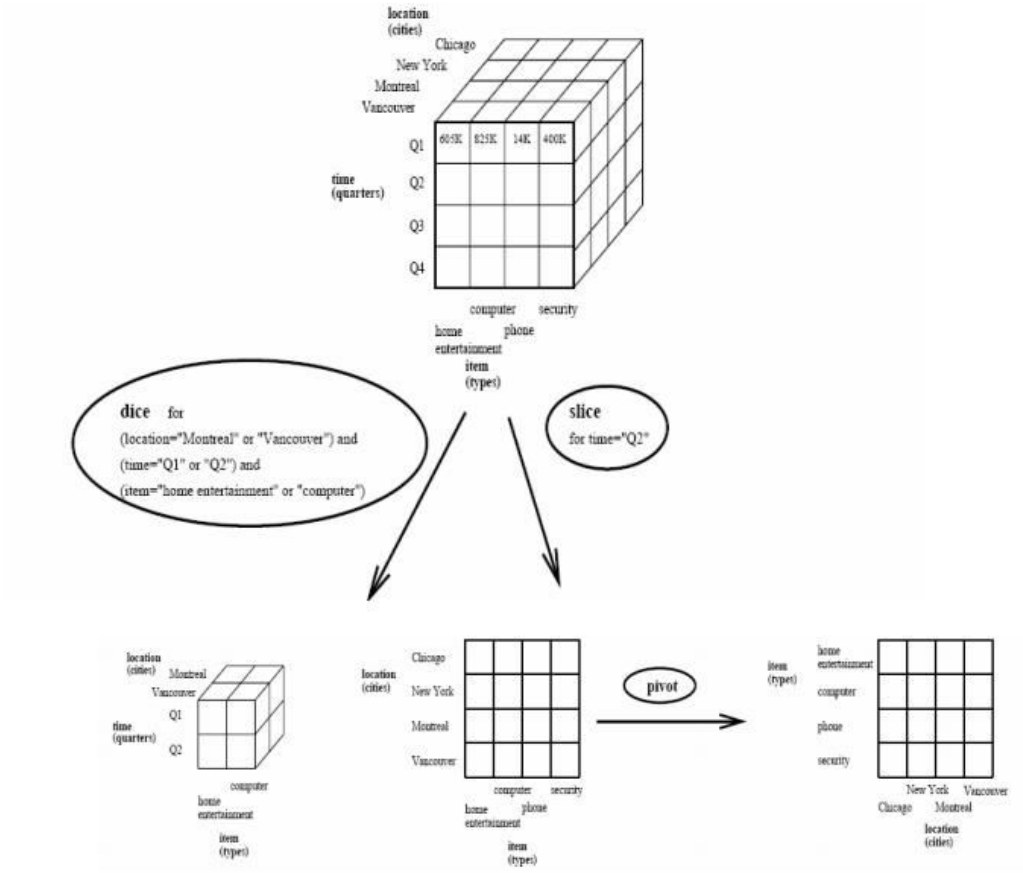
- Drill down (roll down):
    - It is the reverse of roll-up
    - It is performed by stepping down a concept hierarchy for a dimension or introducing new dimensions.

- Slice and Dice:
    - Project and Select operations
    - Check the example.

- Pivot (rotate):
    - Re-orient the cube for an alternative presentation of the data
    - Transform 3D view 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)


location (countries)
US
Canada
Q1
Q2
Q3
Q4
time (quarters)
computer
security
home entertainment
phone
item (types)
location (cities)
Chicago
New York
Montreal
Vancouver
January
February
March
April
May
June
July
August
September
October
November
December
time (months)
roll-up
on location
(from cities to countries)
drill-down
on time
(from quarters to months)
location (cities)
Chicago
New York
Montreal
Vancouver
605K
825K
14K
400K
time (quarters)
dice for
(location="Montreal" or "Vancouver") and
(time="Q1" or "Q2") and
(item="home entertainment" or "computer")
slice
for time="Q2"



10.2. Three Data Warehouse models

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

11. Data Warehouse Implementation

- Objectives:
  - **Monitoring:** Sending data from sources
  - **Integrating:** Loading, cleansing,...
  - **Processing:** Efficient cube computation, and query processing in general, indexing, ...

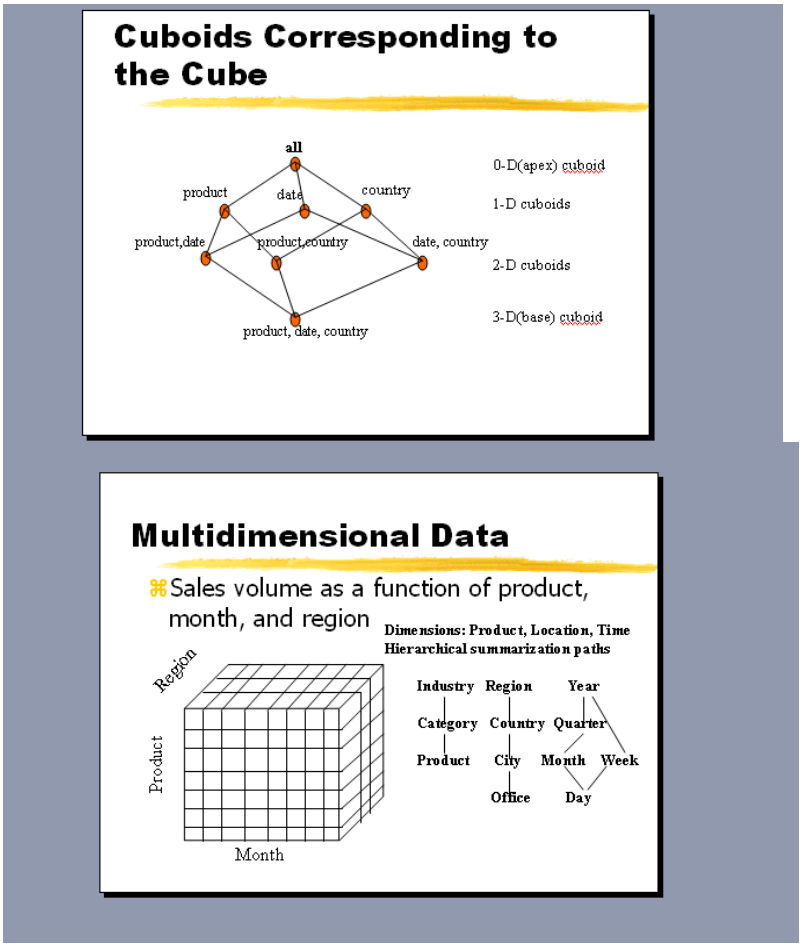
- Cube Computation
  - One approach extends SQL using compute cube operator
  - A cube operator is the n-dimensional generalization of the group-by SQL clause.
  - OLAP needs to compute the cuboid corresponding each input query.
  - Pre-computation: for fast response time, it seems a good idea to pre-compute data for all cuboids or at least a subset of cuboids since the number of cuboids is:

$$\text{number of cuboids} = \begin{cases} 2^n & \text{If no hierarchy} \\ \prod_{i=1}^{n}(L_i + 1) & \text{if hierarchy and } L_i \text{ is number of levels associated with } d \text{ dimension } i \end{cases}$$

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.

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
- 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
- Indexing OLAP Data: Join Indices
- Traditional indices map the values to a list of record ids
- It materializes relational join in JI file and speeds up relational join — a rather costly operation
- In data warehouses, join index relates the values of the dimensions of a 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 to which materialized cuboid(s) the relevant operations should be applied.

Metadata Repository

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

Discovery-Driven Exploration of Data Cubes

- Hypothesis-driven: exploration by user, huge search space

Discovery-driven

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

Examples: Discovery-Driven Data Cubes

Complex Aggregation at Multiple Granularities: Multi-Feature Cubes

Ex. Grouping by all subsets of {item, region, month}, find the maximum price in 1997 for each group, and the total sales among all maximum price tuples

```
select item, region, month, max(price), sum(R.sales)
from purchases
where year = 1997
cube by item, region, month: R
such that R.price = max(price)
```

Data Warehouse Usage

Three kinds of data warehouse applications

Information processing

- supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs

Analytical processing

- multidimensional analysis of data warehouse data
- supports basic OLAP operations, slice-dice, drilling, pivoting

Data mining

- knowledge discovery from hidden patterns
- supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

Differences among the three tasks

Summary

Data warehouse

A subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process

A multi-dimensional model of a data warehouse

- Star schema, snowflake schema, fact constellations
- A data cube consists of dimensions & measures

OLAP operations: drilling, rolling, slicing, dicing and pivoting

- 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)

()

Efficient computation of data cubes

- Partial vs. full vs. no materialization
- Multiway array aggregation
- Bitmap index and join index implementations

Further development of data cube technology

- Discovery-drive and multi-feature cubes

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

11.4. Indexing OLAP Data: Bitmap Index

- Approach:
  - 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
- Example:

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

11.5. Indexing OLAP Data: Join Indices

- Join index:

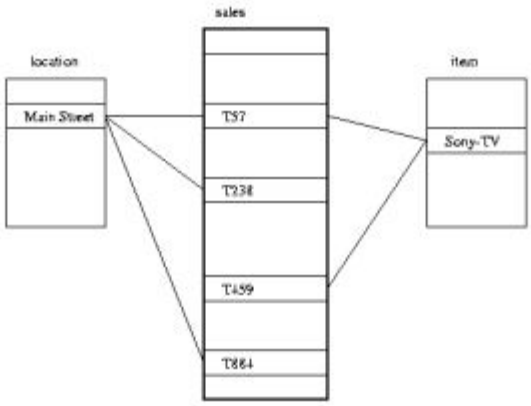
  JI(R-id, S-id)

  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 — a rather costly operation

- In data warehouses, join index relates the values of the dimensions of a star schema to rows in the fact table.
  - E.g. fact table: *Sales* and two dimensions *city* and *product*
    - A join index on *city* maintains for each distinct city a list of R-IDs of the tuples recording the Sales in the city
    - Join indices can span multiple dimensions



Efficient Processing OLAP Queries

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