

Data Mining & Warehousing

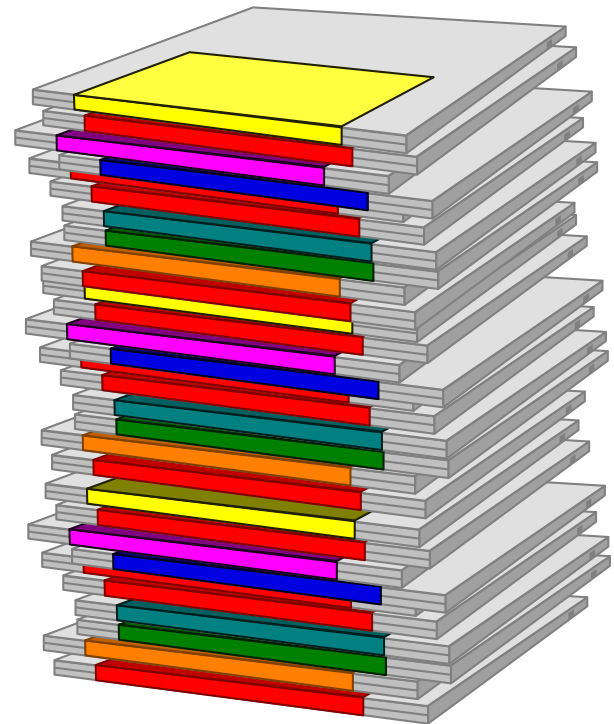
Data, Data everywhere yet ...



- I can't find the data I need
 - data is scattered over the network
 - many versions, subtle differences
- I can't get the data I need
 - need an expert to get the data
- I can't understand the data I found
 - available data poorly documented
- I can't use the data I found
 - results are unexpected
 - data needs to be transformed from one form to other

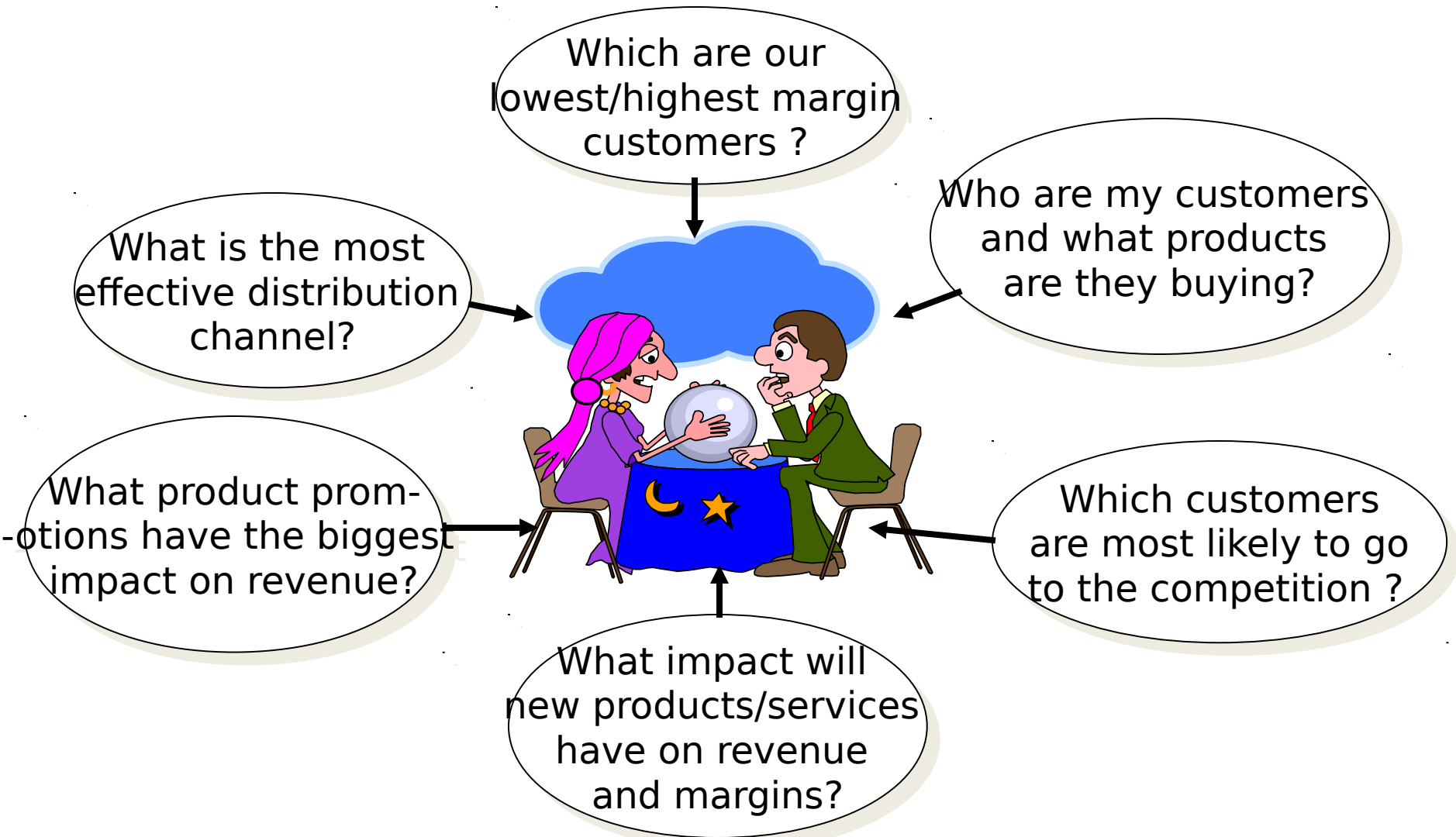
What is a Data Warehouse?

A single, complete and consistent store of data obtained from a variety of different sources made available to end users in a way that they can understand and use in a business context.



[Barry Devlin]

Why Data Warehousing?



What is Data Warehouse?

- Defined in many different ways, but not rigorously.
 - A decision support database that is maintained **separately** from the organization's operational database
 - Support **information processing** by providing a solid platform of consolidated, historical data for analysis.
- “A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.”—W. H. Inmon
- Data warehousing:
 - The process of constructing and using data warehouses

Data Warehouse—Subject-Oriented

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

Data Warehouse— Integrated

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

Data Warehouse—Time Variant

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



Data Warehouse— Nonvolatile

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

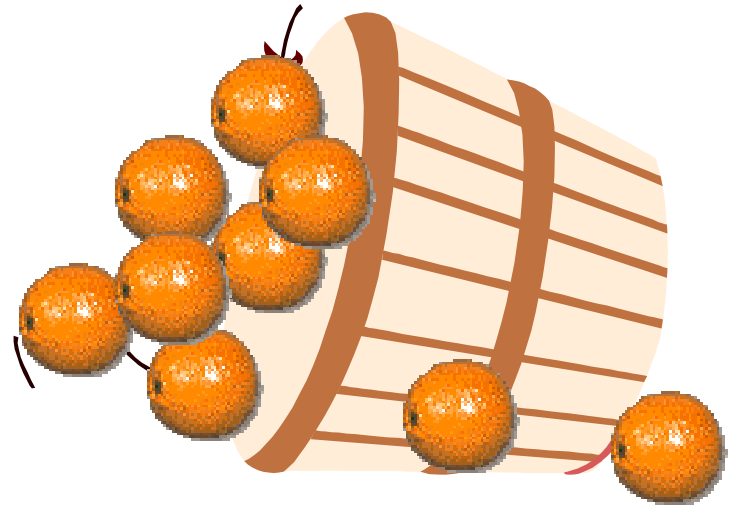
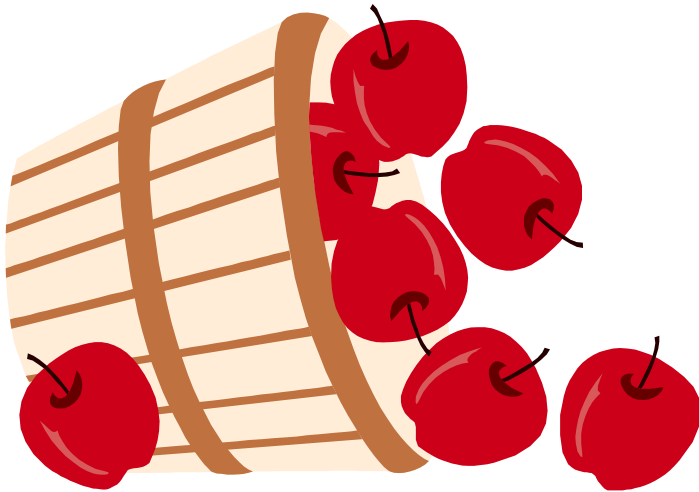
Data Warehouse vs. Heterogeneous DBMS

- Traditional **heterogeneous DB integration**: A **query driven** approach
 - 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
 - Complex information filtering, compete for resources
- **Data warehouse**: **update-driven**, high performance
 - Information from heterogeneous sources is integrated in advance and stored in warehouses for direct query and analysis

Data Warehouse vs. Operational DBMS

- OLTP (on-line transaction processing)
 - Major task of traditional relational DBMS
 - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
 - Major task of data warehouse system
 - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
 - User and system orientation: customer vs. market
 - Data contents: current, detailed vs. historical, consolidated
 - Database design: ER + application vs. star + subject
 - View: current, local vs. evolutionary, integrated
 - Access patterns: update vs. read-only but complex queries

So, what's different?



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

Application-Orientation vs. Subject-Orientation


Application-Orientation

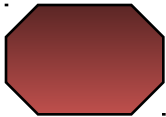
Subject-Orientation


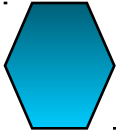
 **Operational Database**


 **Data Warehouse**

 Loans  Credit Card

Customer


Vendor


 Savings  Trust

Product


Activity


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
- Note: There are more and more systems which perform OLAP analysis directly on relational databases

To summarize ...

- ❖ OLTP Systems are used to *“run”* a business



- ❖ The Data Warehouse helps to *“optimize”* the business

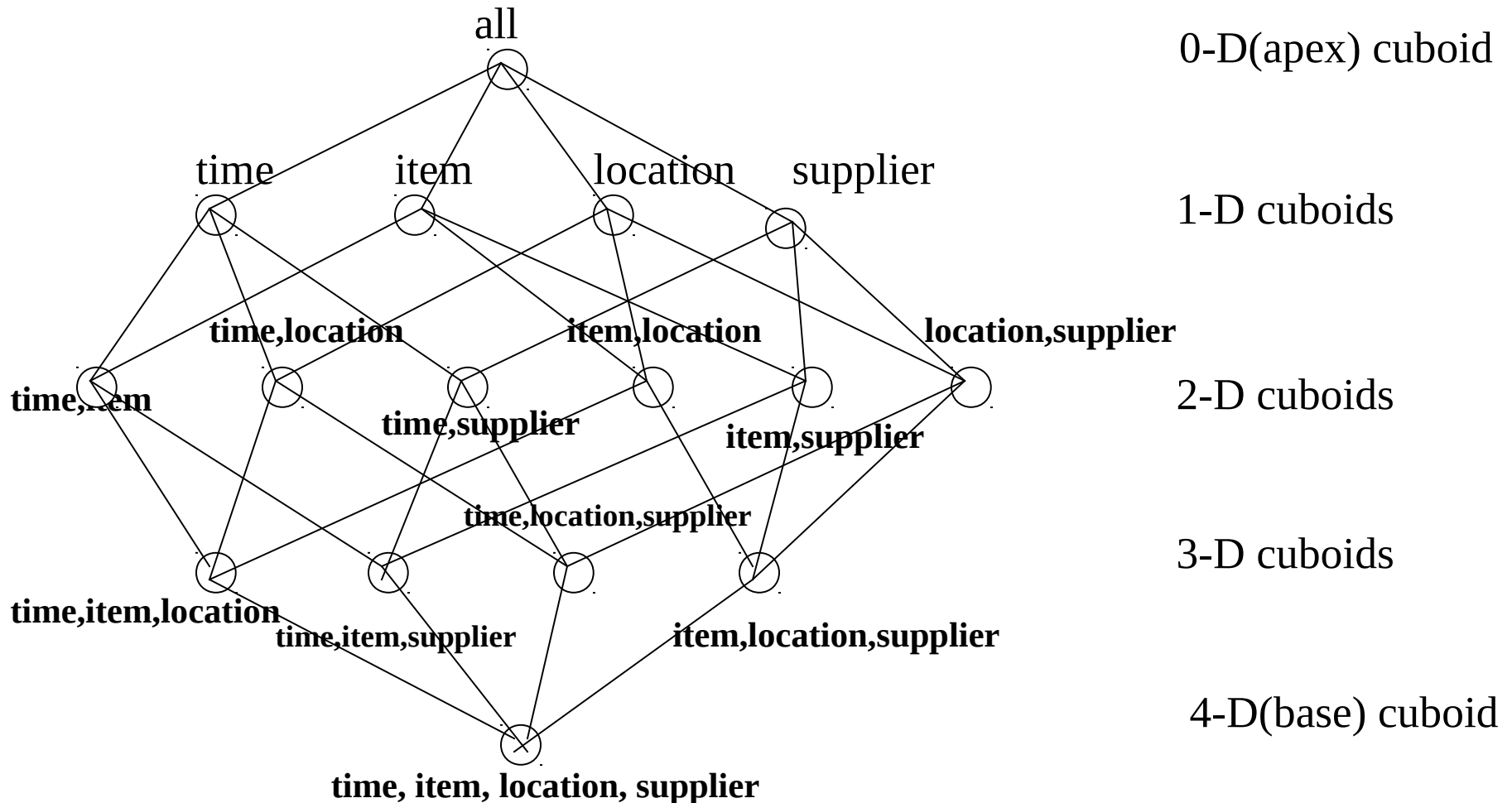
Data Warehousing and OLAP Technology: An Overview

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

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
- In data warehousing literature, an n-D base cube is called a **base cuboid**. The top most 0-D cuboid, which holds the highest-level of summarization, is called the **apex cuboid**. The lattice(network) of cuboids forms a **data cube**.

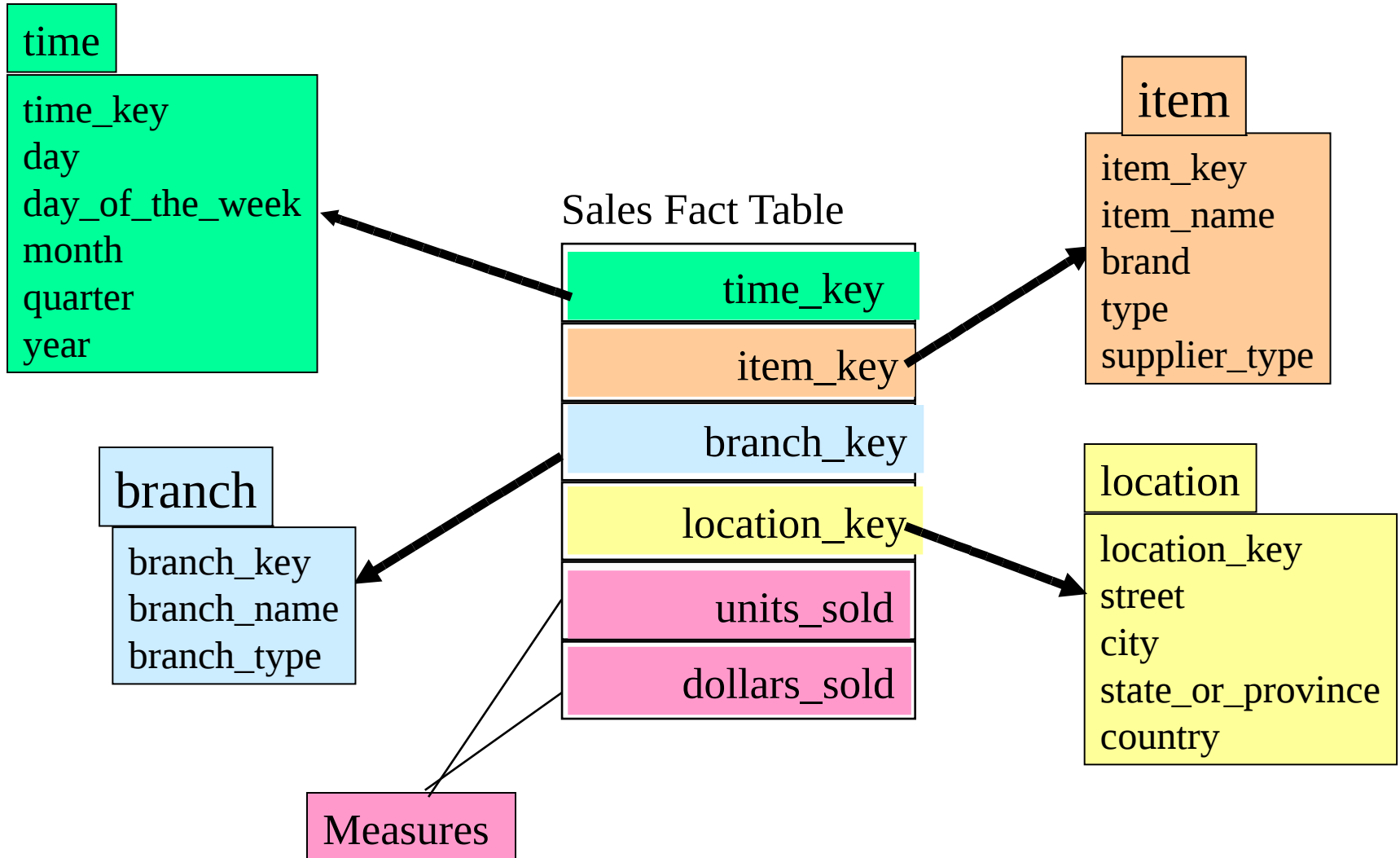
Cube: A Lattice of Cuboids



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: A schema which contains multiple fact tables shares dimensions. It is a collection of star schemas which shares their dimension. So it is also called as a galaxy schema.

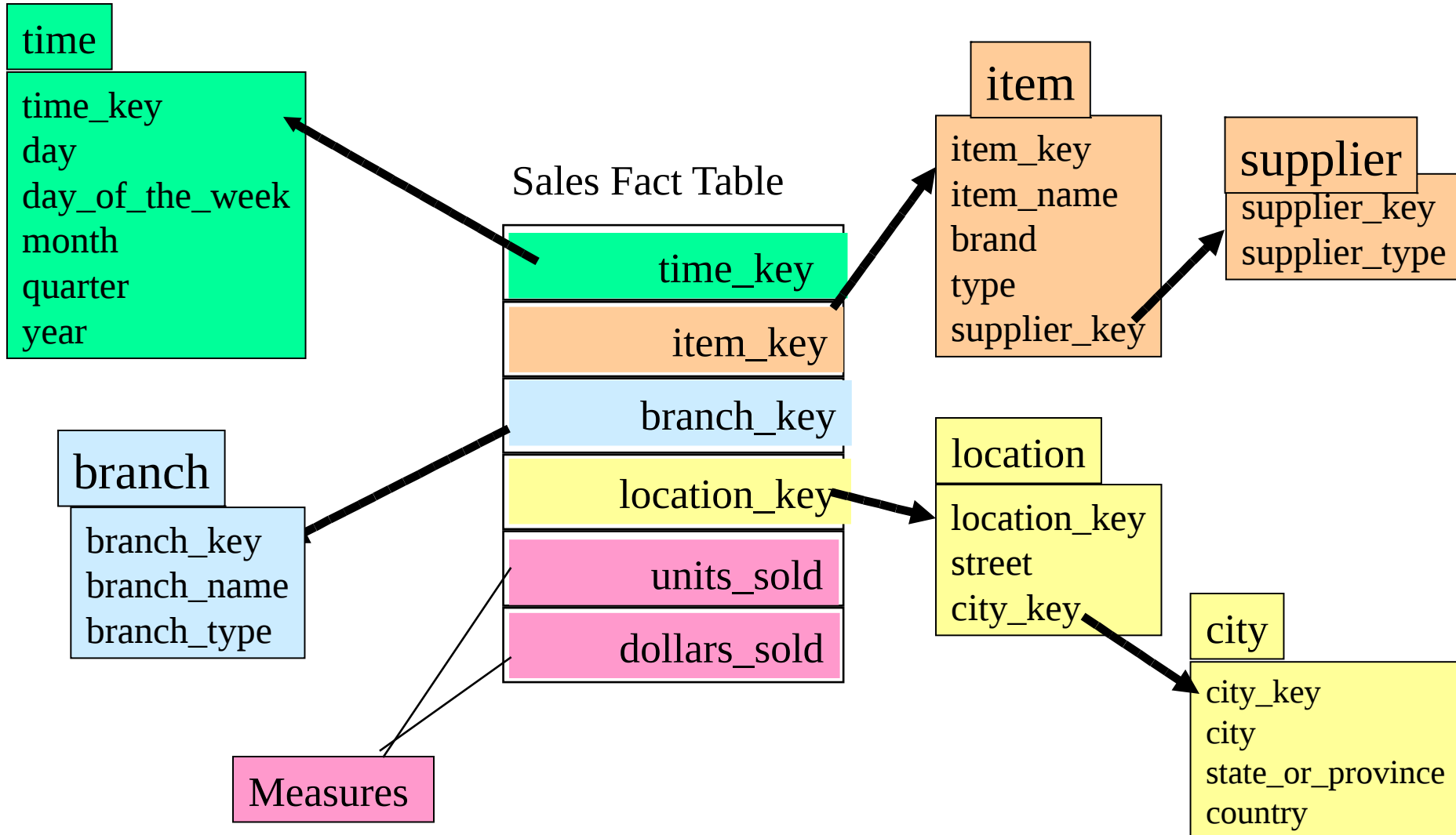
Example of Star Schema



Star Schema

- The most common modeling paradigm is the star schema, in which the data warehouse contains
 - (1) a large central table (fact table) containing the bulk of the data, with no redundancy
 - (2) a set of smaller attendant tables (dimension tables), one for each dimension.
- **Example** Sales are considered along four dimensions, namely, *time*, *item*, *branch*, and *location*. The schema contains a central fact table for *sales* that contains keys to each of the four dimensions, along with two measures: *dollars sold* and *units sold*. To minimize the size of the fact table, dimension identifiers (such as *time key* and *item key*) are system-generated identifiers.
- Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes.

Example of Snowflake Schema

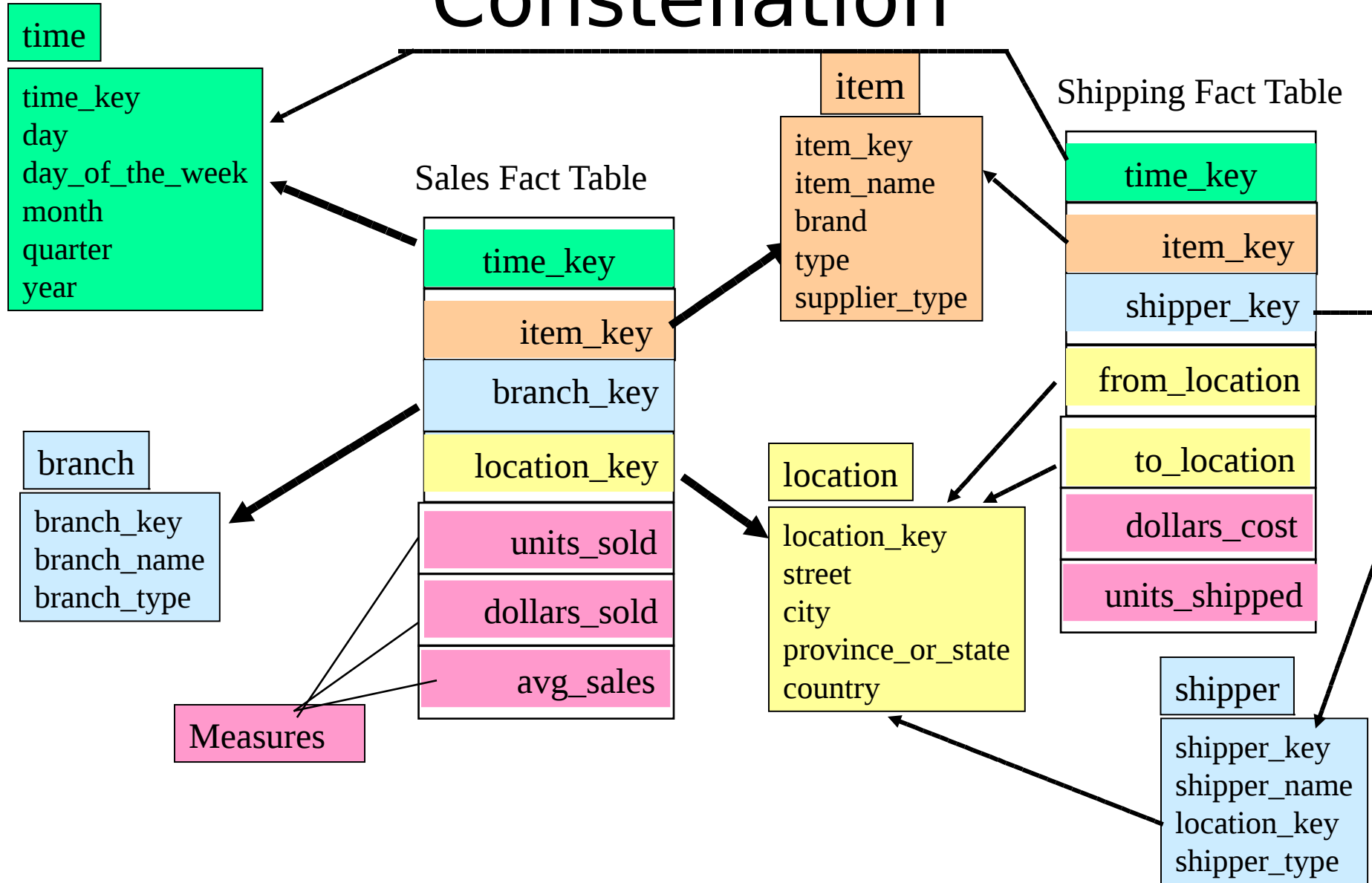


Snowflake Schema

Here,

- some dimension tables are *normalized*, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.
- The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies.
- Such a table is easy to maintain and saves storage space. However, this saving of space is negligible in comparison to the typical magnitude of the fact table.
- Snowflake structure can reduce the effectiveness of browsing, since more joins will be needed to execute a query.
- The system performance may be adversely impacted. Hence, although the snowflake schema reduces redundancy, it is not as popular as the star schema in data warehouse design.

Example of Fact Constellation



Fact Constellation

- Sophisticated applications may require multiple fact tables to *share* dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.
- This schema specifies two fact tables, *sales* and *shipping*. The *sales* table definition is identical to that of the star schema .
- The *shipping* table has five dimensions, or keys: *item key*, *time key*, *shipper key*, *from location*, and *to location*, and two measures: *dollars cost* and *units shipped*.
- A fact constellation schema allows dimension tables to be shared between fact tables.
- For example, the dimensions tables for *time*, *item*, and *location* are shared between both the *sales* and *shipping* fact tables. The fact constellation schema is commonly used, since it can model multiple, interrelated subjects.

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

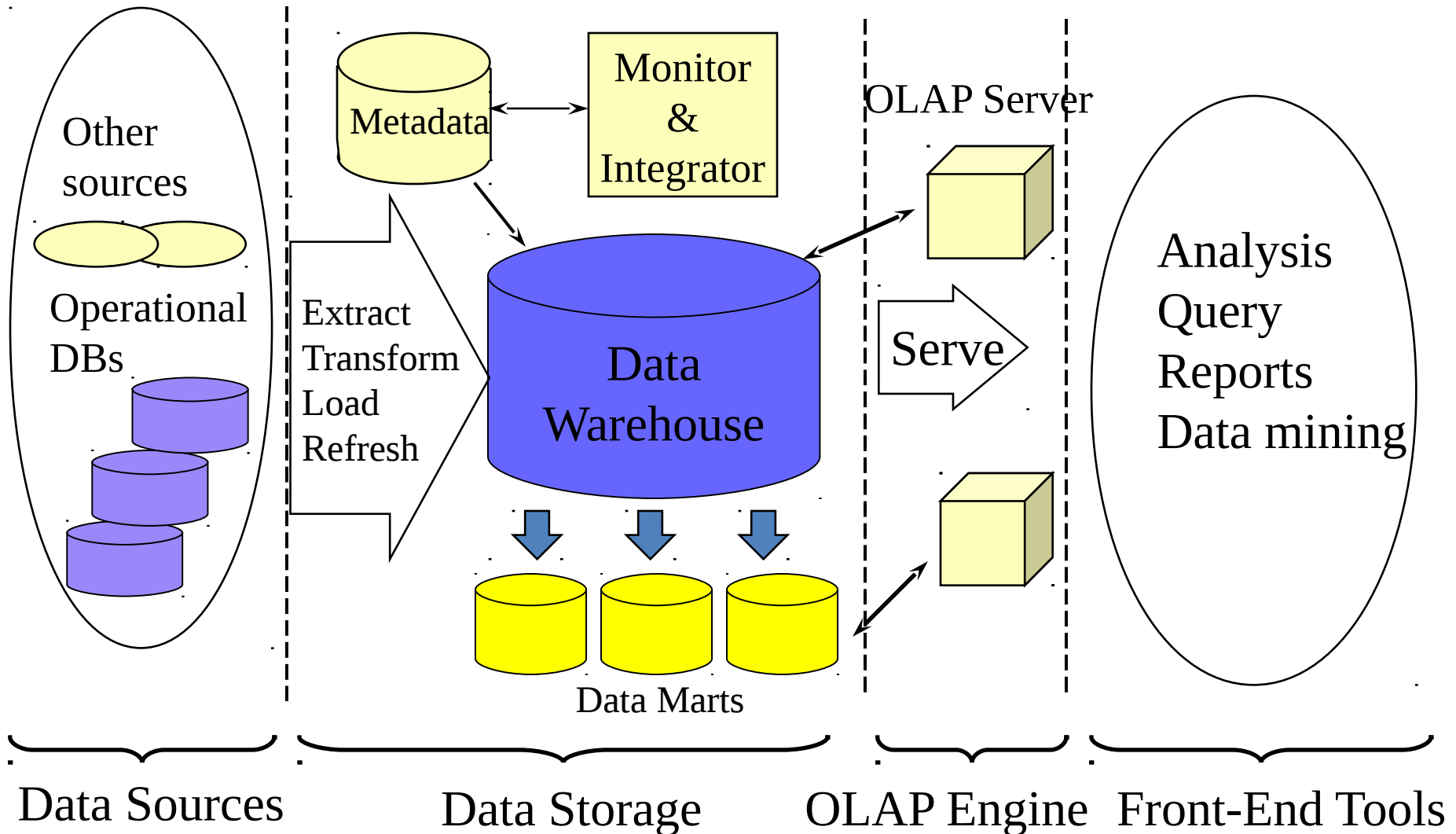
Design of Data Warehouse: A Business Analysis Framework

- 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

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

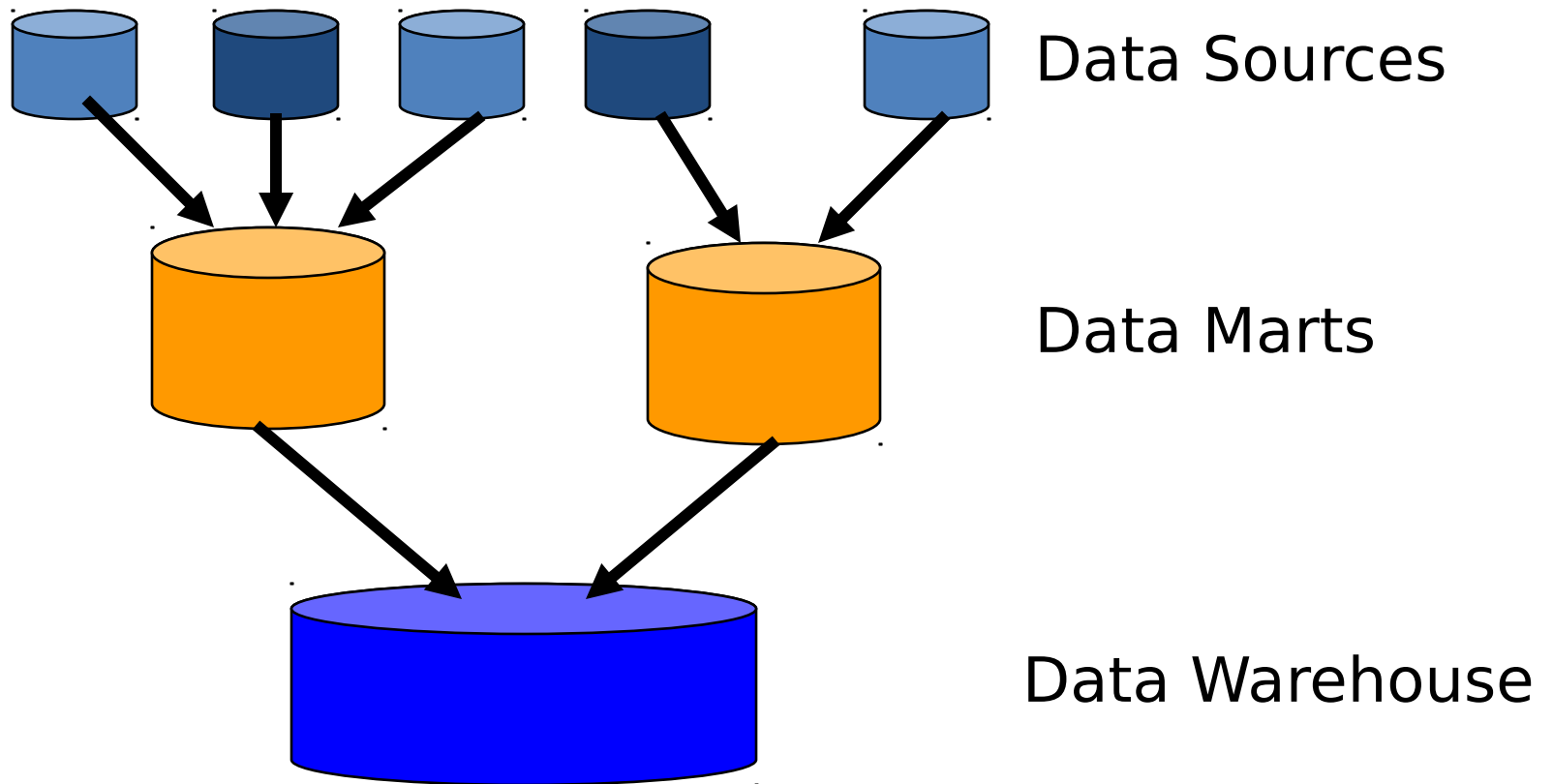
Data Warehouse: A Multi-Tiered Architecture



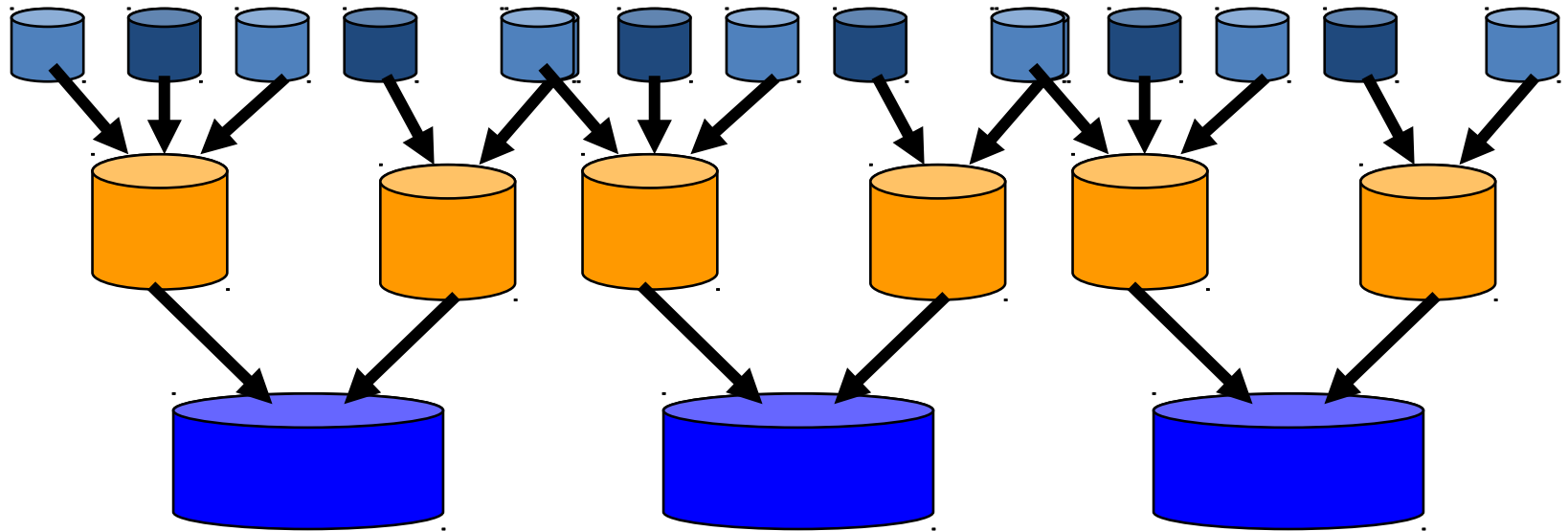
Three Data Warehouse Models

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

Data Mart Centric

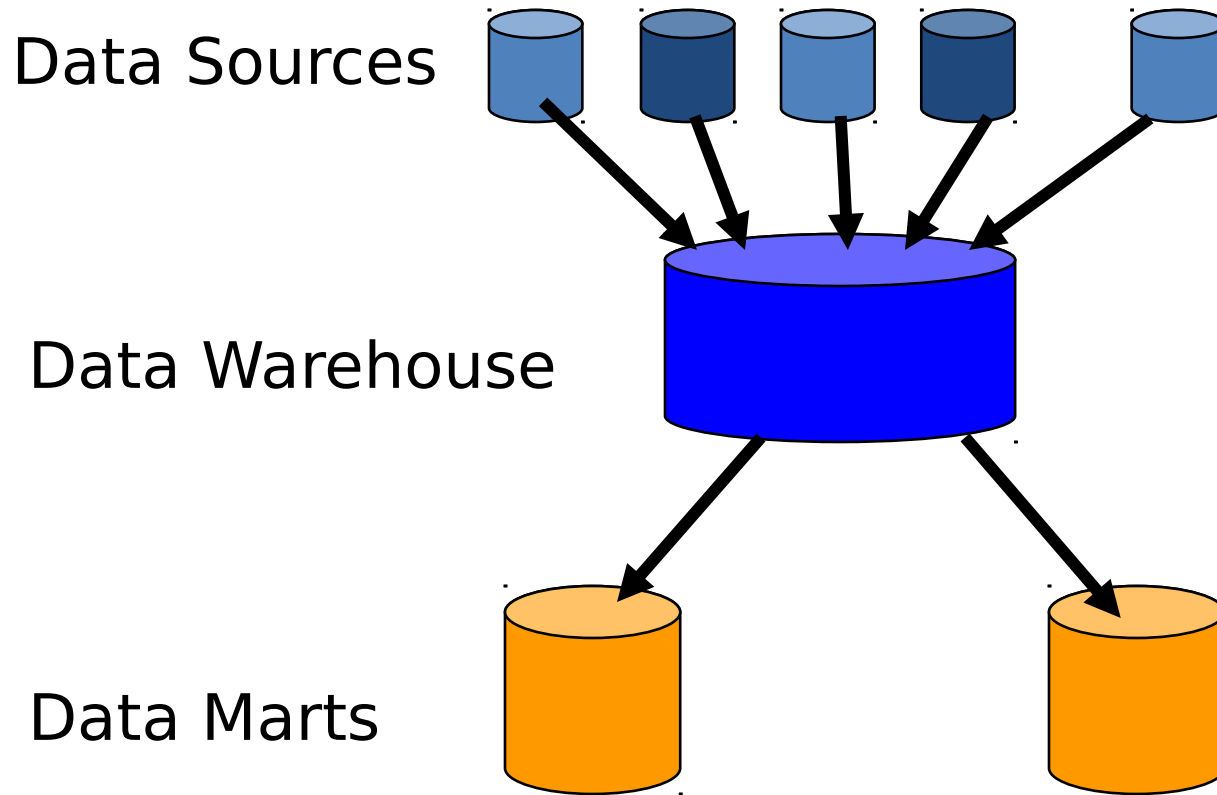


Problems with Data Mart Centric Solution



If you end up creating multiple warehouses, integrating them is a problem

True Warehouse



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 is the data defining warehouse objects. It stores:
- Description of the structure of the data 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

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

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

Efficient Data Cube Computation

- Data cube can be viewed as a lattice(network) of cuboids
 - The bottom-most cuboid is the base cuboid
 - The top-most cuboid (apex) contains only one cell
- Materialization(appearance) of data cube
 - Materialize every (cuboids) (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
```

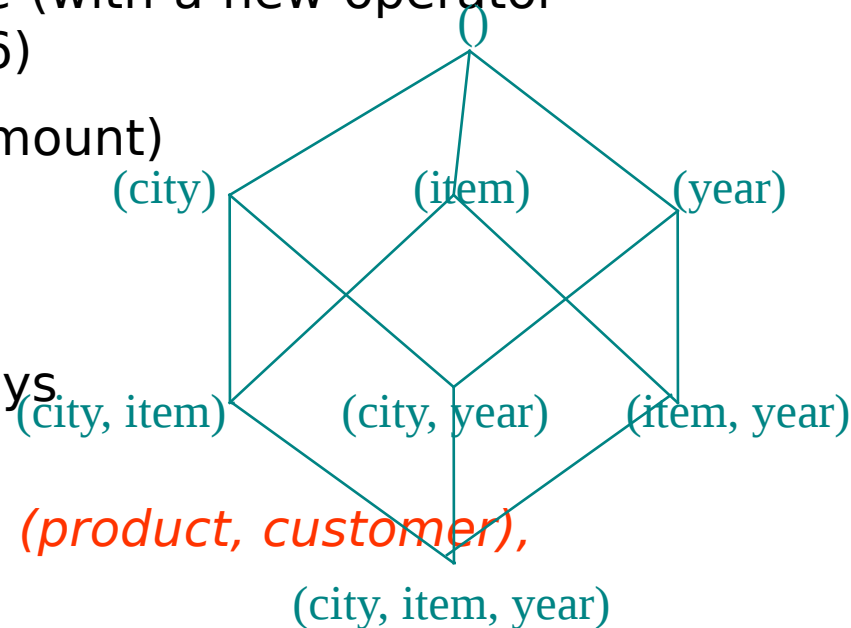
- Need compute the following Group-Bys.

(date, product, customer),

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

(date), (product), (customer)

()



Indexing OLAP Data: Bitmap Index

- Index on a particular column
- Each value in the column has a bit vector: bit-op is fast
- The length of the bit vector: # of records in the base table
- The i -th bit is set if the i -th row of the base table has the value for the indexed column
- not suitable for high cardinality domains

Base table

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

Index on Region

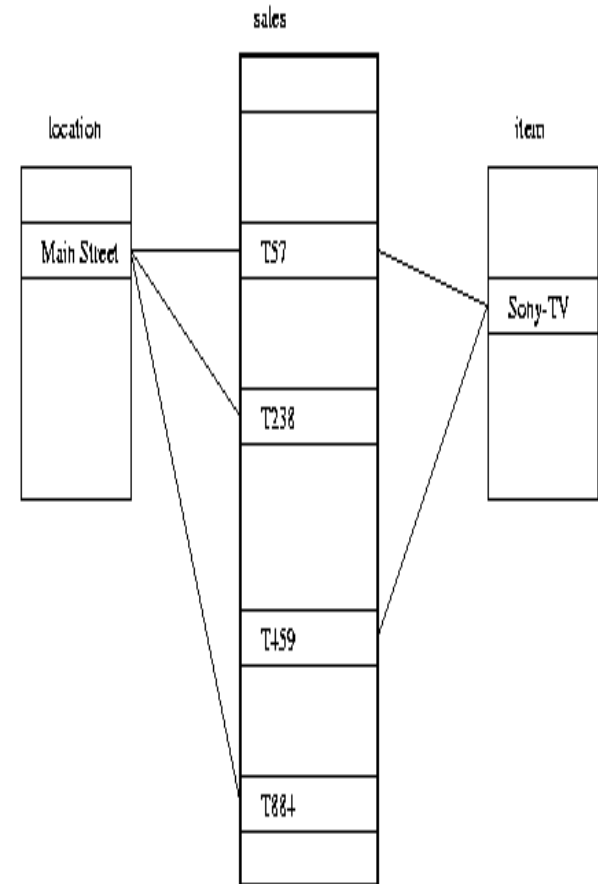
RecID	Asia	Europe	America
1	1	0	0
2	0	1	0
3	1	0	0
4	0	0	1
5	0	1	0

Index on Type

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

Indexing OLAP Data: Join Indices

- Join index: $Jl(R\text{-id}, S\text{-id})$ where $R(R\text{-id}, \dots) \bowtie S(S\text{-id}, \dots)$
- 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 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 dimension



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 = 2004Which should be selected to process the query?
- Explore indexing structures and compressed vs. dense array structs in MOLAP

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

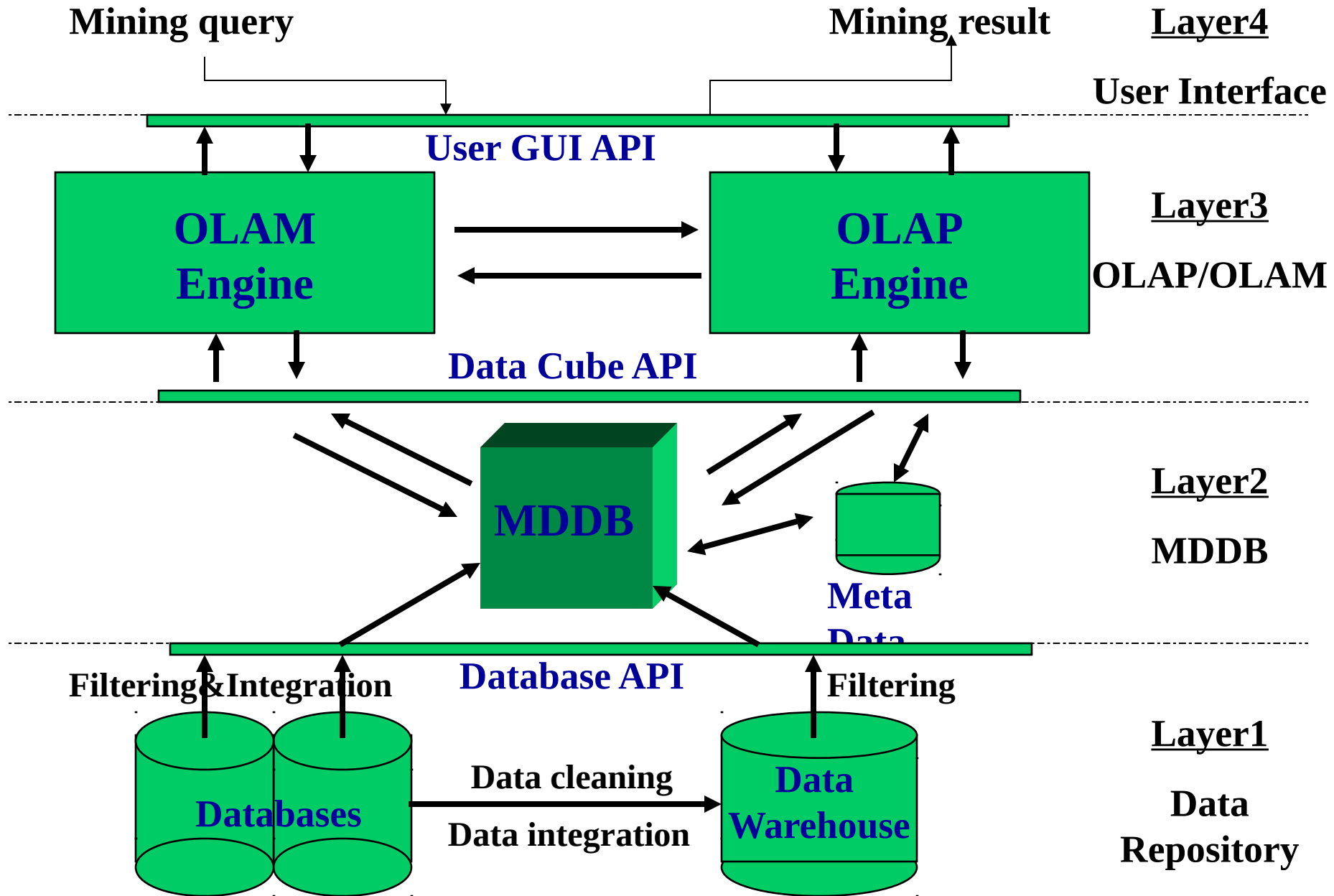
Data Warehouse Usage

- 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

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

An OLAM System Architecture



Data Mining

Why Data Mining ? (Applications)

- Banking: loan/credit card approval
 - predict good customers based on old customers
- Customer relationship management:
 - identify those who are likely to leave for a competitor.
- Targeted marketing:
 - identify likely responders to promotions
- Fraud detection: telecommunications, financial transactions
 - from an online stream of event identify fraudulent events
- Manufacturing and production:
 - automatically adjust knobs when process parameter changes

Applications (continued)

- Medicine: disease outcome, effectiveness of treatments
 - analyze patient disease history: find relationship between diseases
- Molecular/Pharmaceutical: identify new drugs
- Scientific data analysis:
 - identify new galaxies by searching for sub clusters
- Web site/store design and promotion:
 - find affinity of visitor to pages and modify layout

Definition- Data mining

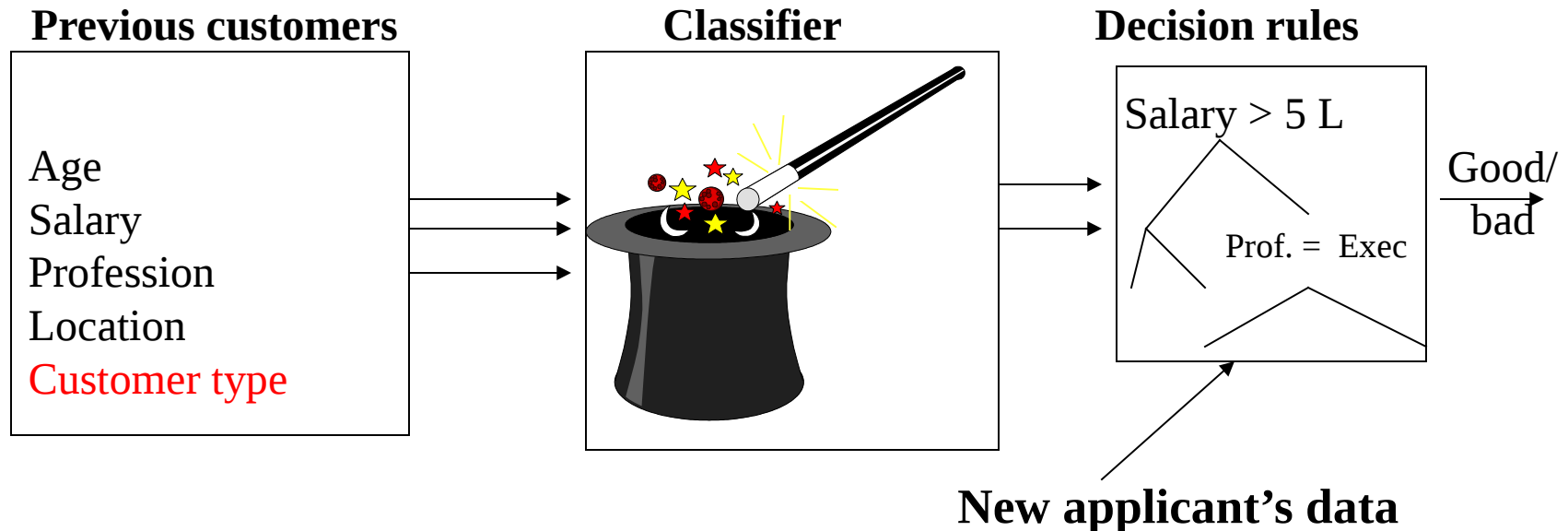
- Process of semi-automatically analyzing large databases to find interesting and useful patterns
- Overlaps with machine learning, statistics, artificial intelligence and databases but
 - more scalable in number of features and instances
 - more automated to handle heterogeneous data
- Also known as Knowledge Discovery in Databases (KDD)

Some basic operations

- Predictive:
 - Regression
 - Classification
- Descriptive:
 - Clustering / similarity matching
 - Association rules and variants
 - Deviation detection

Classification

- Given old data about customers and payments, predict new applicant's loan eligibility.



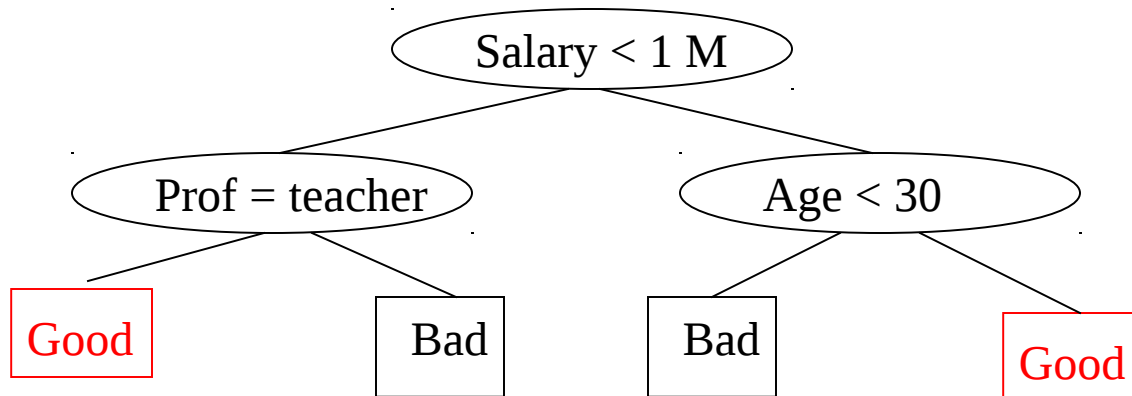
Classification methods

Goal: Predict class $C_i = f(x_1, x_2, \dots, x_n)$

- Regression: (linear or any other polynomial)
 - $a \cdot x_1 + b \cdot x_2 + c = C_i$.
- Nearest neighbour
- Decision tree classifier: divide decision space into piecewise constant regions.
- Probabilistic/generative models
- Neural networks: partition by non-linear boundaries

Decision trees

- Tree where internal nodes are simple decision rules on one or more attributes and leaf nodes are predicted class labels.



Pros and Cons of decision trees

- Pros

- + Reasonable training time
- + Fast application
- + Easy to interpret
- + Easy to implement
- + Can handle large number of features

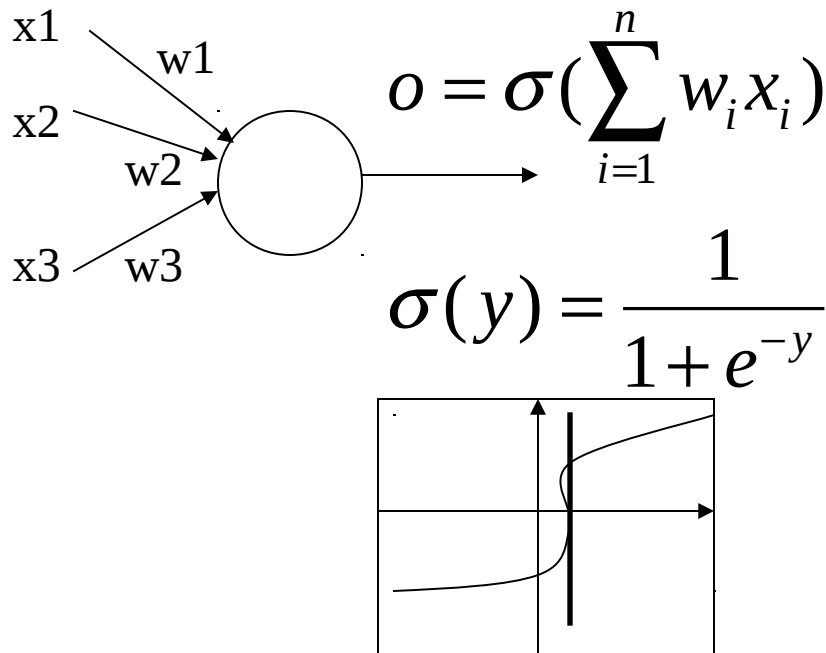
- Cons

- Cannot handle complicated relationship between features
- simple decision boundaries
- problems with lots of missing data

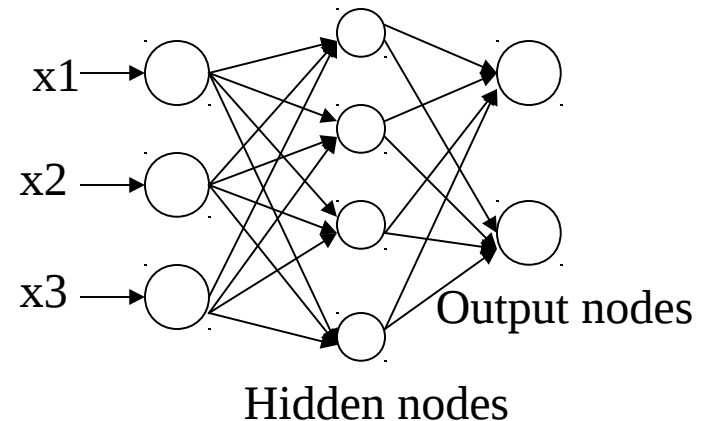
Neural network

- Set of nodes connected by directed weighted edges

Basic NN unit



A more typical NN



Pros and Cons of Neural Network

- Pros

- + Can learn more complicated class boundaries
- + Fast application
- + Can handle large number of features

- Cons

- Slow training time
- Hard to interpret
- Hard to implement: trial and error for choosing number of nodes

Bayesian learning

- Assume a probability model on generation of data.
- Apply bayes theorem to find most likely class as:
predicted class : $c = \max_{c_j} p(c_j | d) = \max_{c_j} \frac{p(d | c_j) p(c_j)}{p(d)}$
- Naïve bayes: Assume attributes conditionally independent given class value
$$c = \max_{c_j} \frac{p(c_j)}{p(d)} \prod_{i=1}^n p(a_i | c_j)$$
- Easy to learn probabilities by counting,
- Useful in some domains e.g. text

Clustering

- Unsupervised learning when old data with class labels not available e.g. when introducing a new product.
- Group/cluster existing customers based on time series of payment history such that similar customers in same cluster.
- Key requirement: Need a good measure of similarity between instances.
- Identify micro-markets and develop policies for each

Association rules

- Given set T of groups of items
- Example: set of item sets purchased
- Goal: find all rules on itemsets of the form $a \rightarrow b$ such that
 - **support** of a and b $>$ user threshold s
 - conditional probability (**confidence**) of b given a $>$ user threshold c
- Example: Milk \rightarrow bread
- Purchase of product A \rightarrow service B

T

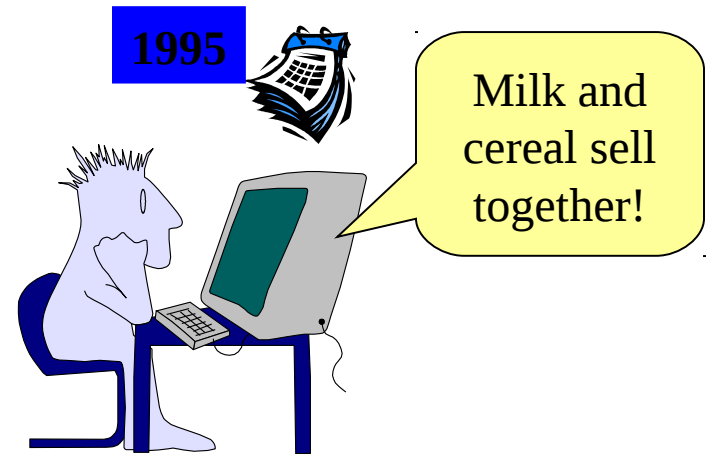
Milk, cereal
Tea, milk
Tea, rice, bread
cereal

Variants

- High confidence may not imply high correlation
- Use correlations. Find expected support and large departures from that interesting..
 - see statistical literature on contingency tables.
- Still too many rules, need to prune...

Prevalent \neq Interesting

- Analysts already know about prevalent rules
- Interesting rules are those that *deviate* from prior expectation
- Mining's payoff is in finding *surprising* phenomena



What makes a rule surprising?

- Does not match prior expectation
 - Correlation between milk and cereal remains roughly constant over time
- Cannot be trivially derived from simpler rules
 - Milk 10%, cereal 10%
 - Milk and cereal 10% ... surprising
 - Eggs 10%
 - Milk, cereal and eggs 0.1% ... surprising!
 - Expected 1%

Application Areas

Industry

Finance

Insurance

Telecommunication

Transport

Consumer goods

Data Service providers

Utilities

Application

Credit Card Analysis

Claims, Fraud Analysis

Call record analysis

Logistics management

promotion analysis

Value added data

Power usage analysis

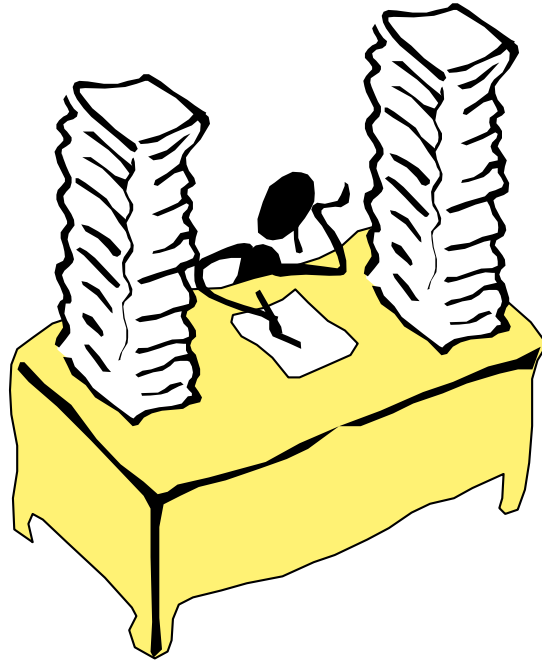
Data Mining in Use

- The US Government uses Data Mining to track fraud
- A Supermarket becomes an information broker
- Basketball teams use it to track game strategy
- Cross Selling
- Target Marketing
- Holding on to Good Customers
- Weeding out Bad Customers

Why Now?

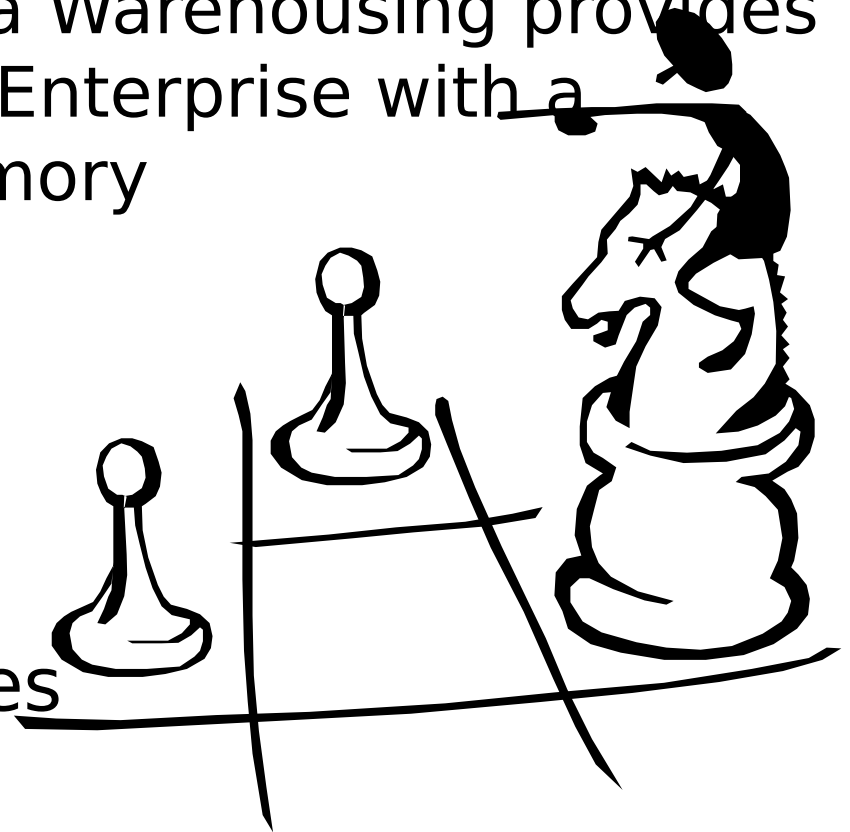
- Data is being produced
- Data is being warehoused
- The computing power is available
- The computing power is affordable
- The competitive pressures are strong
- Commercial products are available

Data Mining works with Warehouse Data



- Data Warehousing provides the Enterprise with a memory

- Data Mining provides the Enterprise with intelligence



Mining market

- Around 20 to 30 mining tool vendors
- Major players:
 - Clementine,
 - IBM's Intelligent Miner,
 - SGI's MineSet,
 - SAS's Enterprise Miner.
- All pretty much the same set of tools
- Many embedded products: fraud detection, electronic commerce applications

OLAP Mining integration

- OLAP (On Line Analytical Processing)
 - Fast interactive exploration of multidim. aggregates.
 - Heavy reliance on manual operations for analysis:
 - Tedious and error-prone on large multidimensional data
- Ideal platform for vertical integration of mining but needs to be interactive instead of batch.

State of art in mining OLAP integration

- Decision trees [**Information discovery**, Cognos]
 - find factors influencing high profits
- Clustering [Pilot software]
 - segment customers to define hierarchy on that dimension
- Time series analysis: [Seagate's Holos]
 - Query for various shapes along time: eg. spikes, outliers etc
- Multi-level Associations [Han et al.]
 - find association between members of dimensions

Vertical integration: Mining on the web

- Web log analysis for site design:
 - what are popular pages,
 - what links are hard to find.
- Electronic stores sales enhancements:
 - recommendations, advertisement:
 - **Collaborative filtering**: Net perception, Wisewire
 - Inventory control: what was a shopper looking for and could not find..

The KDD process

- Problem fomulation
- Data collection
 - subset data: sampling might hurt if highly skewed data
 - feature selection: principal component analysis, heuristic search
- Pre-processing: cleaning
 - name/address cleaning, different meanings (annual, yearly), duplicate removal, supplying missing values
- Transformation:
 - map complex objects e.g. time series data to features e.g. frequency
- Choosing mining task and mining method:
- Result evaluation and Visualization:

Knowledge discovery is an iterative process

Thank You