

**COURSE OBJECTIVES:**

To make the students

1. To understand the concept of Data Warehouse and its significance.
2. To gain the knowledge of hardware and operational design of data warehouses
3. To obtain the knowledge of planning the requirements for data warehousing.
4. To understand the types of the data mining techniques and its application
5. To comprehend on the concept of knowledge discovery process and its application

**COURSE OUTCOMES:**

Learners should be able to

1. Understand the basic principles, concepts and applications of data warehousing and data mining,
2. Comprehend the importance of Processing raw data to make it suitable for various data mining algorithms.
3. Visualize the techniques of clustering, classification, association finding, feature selection and its importance in analysing the real-world data.
4. Understand the Conceptual, Logical, and Physical design of Data Warehouses OLAP applications and OLAP deployment
5. Exhibit the communication skills to convey the thoughts and ideas of case analysis to the individuals and group.

**UNIT I Data warehousing**

Meaning and Significance – Data Warehouse Architecture: System Process – Process architecture – Design – Database scheme – Partitioning strategy – Aggregations – Data mart – Meta data – Systems and data Warehouse process managers.

**UNIT II Hardware and Operational design of data warehouses**

Hardware and Operational design of data warehouses – Hardware architecture – Physical layout – Security – Backup and Recovery – Service level agreement – Operating the data warehouse.

### **UNIT III Data warehouse Planning**

Tuning and Testing – Capacity planning – Testing the data warehouses – Data warehouse features.

### **UNIT IV Data mining**

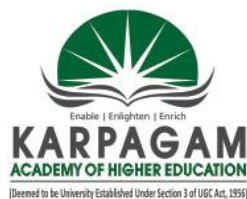
Introduction – Information and production factor – Data mining Vs Query tools – Data mining in marketing – Self learning computer systems – concept learning.

### **UNIT V Knowledge discovery process**

Data selection – Cleaning – Enrichment – Coding – Preliminary analysis of the data set using traditional query tools – Visualization techniques – OLAP tools – Decision trees – Association rules – Neural networks – Genetic Algorithms KDD (Knowledge discover in Database) environment.

### **SUGGESTED READINGS :**

1. Alex Berson, Stephen Smith (2017), Data Warehousing, Data Mining, & OLAP, McGraw Hill Education, New Delhi
2. Daniel T. Larose, Chantal D. Larose (2016), Data Mining and Predictive Analytics, 2<sup>nd</sup> edition, Wiley, New Delhi.
3. Daniel T. Larose, Chantal D. Larose (2015), Discovering Knowledge in Data: An Introduction to Data Mining, 2<sup>nd</sup> edition, Wiley, New Delhi.
4. Mehmed Kantardzic (2017), Data Mining: Concepts, Models, Methods and Algorithms, 2<sup>nd</sup> edition, Wiley, New Delhi.
5. Gordon S. Linoff, Michael J.A. Berry (2012), Data Mining Techniques: For Marketing, Sales and Customer Relationship Management, 3rd edition, Wiley, New Delhi.



# KARPAGAM ACADEMY OF HIGHER EDUCATION

(Deemed to be University)

(Established under section 3 of UGC Act 1956)

Coimbatore-641021

**Department of Commerce**

Name: **M RAM KUMAR**

Department: **Commerce**

Subject Code: **18CCP202**

Semester: **II**

Year: **2018 - 19 Batch**

Subject: **Data Mining and Warehousing**

<b>UNIT 1</b>			
<b>S.No</b>	<b>Lecture Hours</b>	<b>Contents</b>	<b>References</b>
1	1	Data warehouse architecture- Systemprocess-process flow with in data ware house	T2 : PG NO: 15 & 33
2	1	Process architecture-load manager, Warehouse manager, query manager	T2 : PG NO: 43
3	1	Design – database schema , starflake schema, identifying facts, designing fact tables	T2 : PG NO: 71
4	1	Partitioning strategy	T2 : PG NO: 101
5	1	Aggregations, Data Marting	T2 : PG NO: 115 & 130
6	1	Meaning of meta data and uses,	T2 : PG NO: 139
7	1	System and data warehouse process managers- System managers	T2 : PG NO: 149
8	1	Recapitulation and discussion of Important questions	
<b>Total Number of hours planned for Unit 1</b>			<b>8</b>
<b>UNIT 2</b>			
1	1	Hardware and operational design of data warehouse - Hardware architecture-process, server hardware	T2 : PG NO: 169
2	1	physical layout – paralleltechnology, disk technology	T2 : PG NO: 179
3	1	Database layout, file systems	T2 : PG NO: 193
4	1	security- requirements	T2 : PG NO: 202
5	1	Performance impact of security, security impact of Design	T2 : PG NO: 213
6	1	Backup strategies, testing the strategies,	T2 : PG NO: 219
7	1	Disaster Recovery	T2 : PG NO: 233

8	1	service level agreement- types of system, data warehouse is operational system or not, operating the data warehouse- day to day operation of the data warehouse	T2 : PG NO: 236 TO 249
9	1	Recapitulation and discussion of Important questions	
<b>Total Number of hours planned for Unit 2</b>			<b>9</b>
<b>UNIT 3</b>			
1		Tuning the data warehouse – assessing performance	T2 : PG NO: 255 TO 267
2		Tuning the data load	T2 : PG NO: 269
3		Explain the concept of Tuning queries	T2 : PG NO: 272
4		Testing the data warehouses-developing the test plan	T2 : PG NO: 278
5		Testing backup recovery and operational Environment	T2 : PG NO: 281 & 282
6		Testing the data base, Testing the applications, Logistics of the test	T2 : PG NO: 284 & 286
7		Data warehouse futures	T2 : PG NO: 291
8	1	Recapitulation and discussion of Important questions	
<b>Total Number of hours planned for Unit 3</b>			<b>8</b>
<b>UNIT 4</b>			
1	1	Introduction to Data mining	T2 : PG NO: 13
2	1	Data mining and Data warehouse	T2 : PG NO: 37
3	1	Information and production factor	T2 : PG NO: 14
4	1	Data mining Vs query tools	T2 : PG NO: 18
5	1	Data mining and Marketing ,	T2 : PG NO: 19
6	1	Self learning computer system	T2 : PG NO: 24
7	1	Concept Learning	T2 : PG NO: 29
8	1	Recapitulation and Discussion of Important questions	
<b>Total Number of hours planned for Unit 3</b>			<b>8</b>
<b>UNIT 5</b>			
1	1	Data selection , Cleaning, Enrichment, Coding	T2 : PG NO:51
2	1	Preliminary analysis of the data set using traditional query tools	T2 : PG NO: 59

3	1	Visualization techniques	T2 : PG NO: 64
4	1	OLAP Tools, Decision trees, Association rules, Neural networks	T2 : PG NO: 68
5	1	Genetic Algorithms	T2 : PG NO: 84
6	1	KDD (Knowledge Discovery in Database)	T2 : PG NO: 91
7	1	Knowledge Discovery Environment	T2 : PG NO: 93
8	1	Recapitulation and discussion of Important questions	
9	1	Discussion of previous year ESE Question papers	
10	1	Discussion of previous year ESE Question papers	
11	1	Discussion of previous year ESE Question papers	
<b>Total Number of hours planned for Unit 5 and discussion of previous year ESE Question papers</b>			<b>11</b>
<b>Total Number of hours allotted for all five units</b>			<b>44</b>

### SUGGESTED READINGS:

#### TEXT BOOKS

1. Sam Anahory, Dennis Murray "Data Warehousing in real world" (2009), AddisonWesley
2. Pieter Adriaans, Dolf, Zantinge (2007), "Data mining", Addison Wesley”
3. Mark Hall, Ian Witten and Eibe Frank (2011),”Data Mining: Practical MachineLearning Tools and Techniques”, Third edition, Morgan Kaufmann Publisher.
4. PaulrajPonniah (2012), “Data Warehousing: Fundamentals for IT Professionals”,Second Edition, Wiley India Pvt Ltd.

# Data Warehousing - Quick Guide

## Data Warehousing - Overview

The term "Data Warehouse" was first coined by Bill Inmon in 1990. According to Inmon, a data warehouse is a subject oriented, integrated, time-variant, and non-volatile collection of data. This data helps analysts to take informed decisions in an organization.

An operational database undergoes frequent changes on a daily basis on account of the transactions that take place. Suppose a business executive wants to analyze previous feedback on any data such as a product, a supplier, or any consumer data, then the executive will have no data available to analyze because the previous data has been updated due to transactions.

A data warehouses provides us generalized and consolidated data in multidimensional view. Along with generalized and consolidated view of data, a data warehouses also provides us Online Analytical Processing (OLAP) tools. These tools help us in interactive and effective analysis of data in a multidimensional space. This analysis results in data generalization and data mining.

Data mining functions such as association, clustering, classification, prediction can be integrated with OLAP operations to enhance the interactive mining of knowledge at multiple level of abstraction. That's why data warehouse has now become an important platform for data analysis and online analytical processing.

## Understanding a Data Warehouse

A data warehouse is a database, which is kept separate from the organization's operational database.

There is no frequent updating done in a data warehouse.

It possesses consolidated historical data, which helps the organization to analyze its business.

A data warehouse helps executives to organize, understand, and use their data to take strategic decisions.

Data warehouse systems help in the integration of diversity of application systems.

A data warehouse system helps in consolidated historical data analysis.

## Why a Data Warehouse is Separated from Operational Databases

A data warehouses is kept separate from operational databases due to the following reasons –

An operational database is constructed for well-known tasks and workloads such as searching particular records, indexing, etc. In contrast, data warehouse queries are often complex and they present a general form of data.

Operational databases support concurrent processing of multiple transactions. Concurrency control and recovery mechanisms are required for operational databases to ensure robustness and consistency of the database.

An operational database query allows to read and modify operations, while an OLAP query needs only **read only** access of stored data.

An operational database maintains current data. On the other hand, a data warehouse maintains historical data.

## Data Warehouse Features

The key features of a data warehouse are discussed below –

**Subject Oriented** – A data warehouse is subject oriented because it provides information around a subject rather than the organization's ongoing operations. These subjects can be product, customers, suppliers, sales, revenue, etc. A data warehouse does not focus on the ongoing operations, rather it focuses on modelling and analysis of data for decision making.

**Integrated** – A data warehouse is constructed by integrating data from heterogeneous sources such as relational databases, flat files, etc. This integration enhances the effective analysis of data.

**Time Variant** – The data collected in a data warehouse is identified with a

particular time period. The data in a data warehouse provides information from the historical point of view.

**Non-volatile** – Non-volatile means the previous data is not erased when new data is added to it. A data warehouse is kept separate from the operational database and therefore frequent changes in operational database is not reflected in the data warehouse.

**Note** – A data warehouse does not require transaction processing, recovery, and concurrency controls, because it is physically stored and separate from the operational database.

## Data Warehouse Applications

As discussed before, a data warehouse helps business executives to organize, analyze, and use their data for decision making. A data warehouse serves as a sole part of a plan-execute-assess "closed-loop" feedback system for the enterprise management. Data warehouses are widely used in the following fields –

- Financial services
- Banking services
- Consumer goods
- Retail sectors
- Controlled manufacturing

## Types of Data Warehouse

Information processing, analytical processing, and data mining are the three types of data warehouse applications that are discussed below –

**Information Processing** – A data warehouse allows to process the data stored in it. The data can be processed by means of querying, basic statistical analysis, reporting using crosstabs, tables, charts, or graphs.

**Analytical Processing** – A data warehouse supports analytical processing of the information stored in it. The data can be analyzed by means of basic OLAP operations, including slice-and-dice, drill down, drill up, and pivoting.

**Data Mining** – Data mining supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction. These mining results can be presented using the visualization tools.



Sr.No.	Data Warehouse (OLAP)	Operational Database(OLTP)
1	It involves historical processing of information.	It involves day-to-day processing.
2	OLAP systems are used by knowledge workers such as executives, managers, and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
3	It is used to analyze the business.	It is used to run the business.
4	It focuses on Information out.	It focuses on Data in.
5	It is based on Star Schema, Snowflake Schema, and Fact Constellation Schema.	It is based on Entity Relationship Model.
6	It focuses on Information out.	It is application oriented.
7	It contains historical data.	It contains current data.
8	It provides summarized and consolidated data.	It provides primitive and highly detailed data.
9	It provides summarized and multidimensional view of data.	It provides detailed and flat relational view of data.
10	The number of users is in hundreds.	The number of users is in thousands.
11	The number of records accessed is in millions.	The number of records accessed is in tens.
12	The database size is from 100GB to 100 TB.	The database size is from 100 MB to 100 GB.
13	These are highly flexible.	It provides high performance.

## Data Warehousing - Concepts

### What is Data Warehousing?

Data warehousing is the process of constructing and using a data warehouse. A data warehouse is constructed by integrating data from multiple heterogeneous sources that support analytical reporting, structured and/or ad hoc queries, and decision making. Data warehousing involves data cleaning, data integration, and data consolidations.

### Using Data Warehouse Information

There are decision support technologies that help utilize the data available in a data warehouse. These technologies help executives to use the warehouse quickly and effectively. They can gather data, analyze it, and take decisions based on the information present in the warehouse. The information gathered in a warehouse can be used in any of the following domains –

**Tuning Production Strategies** – The product strategies can be well tuned by repositioning the products and managing the product portfolios by comparing the sales quarterly or yearly.

**Customer Analysis** – Customer analysis is done by analyzing the customer's buying preferences, buying time, budget cycles, etc.

**Operations Analysis** – Data warehousing also helps in customer relationship management, and making environmental corrections. The information also allows us to analyze business operations.

## Integrating Heterogeneous Databases

To integrate heterogeneous databases, we have two approaches –

Query-driven Approach

Update-driven Approach

### Query-Driven Approach

This is the traditional approach to integrate heterogeneous databases. This approach was used to build wrappers and integrators on top of multiple heterogeneous databases. These integrators are also known as mediators.

#### Process of Query-Driven Approach

When a query is issued to a client side, a metadata dictionary translates the query into an appropriate form for individual heterogeneous sites involved.

Now these queries are mapped and sent to the local query processor.

The results from heterogeneous sites are integrated into a global answer set.

#### Disadvantages

Query-driven approach needs complex integration and filtering processes.

This approach is very inefficient.

It is very expensive for frequent queries.

This approach is also very expensive for queries that require aggregations.

## Update-Driven Approach

This is an alternative to the traditional approach. Today's data warehouse systems follow update-driven approach rather than the traditional approach discussed earlier. In update-driven approach, the information from multiple heterogeneous sources are integrated in advance and are stored in a warehouse. This information is available for direct querying and analysis.

### Advantages

This approach has the following advantages –

This approach provide high performance.

The data is copied, processed, integrated, annotated, summarized and restructured in semantic data store in advance.

Query processing does not require an interface to process data at local sources.

## Functions of Data Warehouse Tools and Utilities

The following are the functions of data warehouse tools and utilities –

**Data Extraction** – Involves gathering data from multiple heterogeneous sources.

**Data Cleaning** – Involves finding and correcting the errors in data.

**Data Transformation** – Involves converting the data from legacy format to warehouse format.

**Data Loading** – Involves sorting, summarizing, consolidating, checking integrity, and building indices and partitions.

**Refreshing** – Involves updating from data sources to warehouse.

**Note** – Data cleaning and data transformation are important steps in improving the quality of data and data mining results.

## Data Warehousing - Terminologies

In this chapter, we will discuss some of the most commonly used terms in data warehousing.

### Metadata

Metadata is simply defined as data about data. The data that are used to represent other data is known as metadata. For example, the index of a book serves as a metadata for the contents in the book. In other words, we can say that metadata is the summarized data that leads us to the detailed data.

In terms of data warehouse, we can define metadata as following –

Metadata is a road-map to data warehouse.

Metadata in data warehouse defines the warehouse objects.

Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.

## Metadata Repository

Metadata repository is an integral part of a data warehouse system. It contains the following metadata –

**Business metadata** – It contains the data ownership information, business definition, and changing policies.

**Operational metadata** – It includes currency of data and data lineage. Currency of data refers to the data being active, archived, or purged. Lineage of data means history of data migrated and transformation applied on it.

**Data for mapping from operational environment to data warehouse** – It metadata includes source databases and their contents, data extraction, data partition, cleaning, transformation rules, data refresh and purging rules.

**The algorithms for summarization** – It includes dimension algorithms, data on granularity, aggregation, summarizing, etc.

## Data Cube

A data cube helps us represent data in multiple dimensions. It is defined by dimensions and facts. The dimensions are the entities with respect to which an enterprise preserves the records.

### Illustration of Data Cube

Suppose a company wants to keep track of sales records with the help of sales data warehouse with respect to time, item, branch, and location. These dimensions allow to keep track of monthly sales and at which branch the items were sold. There is a table associated with each dimension. This table is known as dimension table. For example, "item" dimension table may have attributes such as item\_name, item\_type, and

item\_brand.

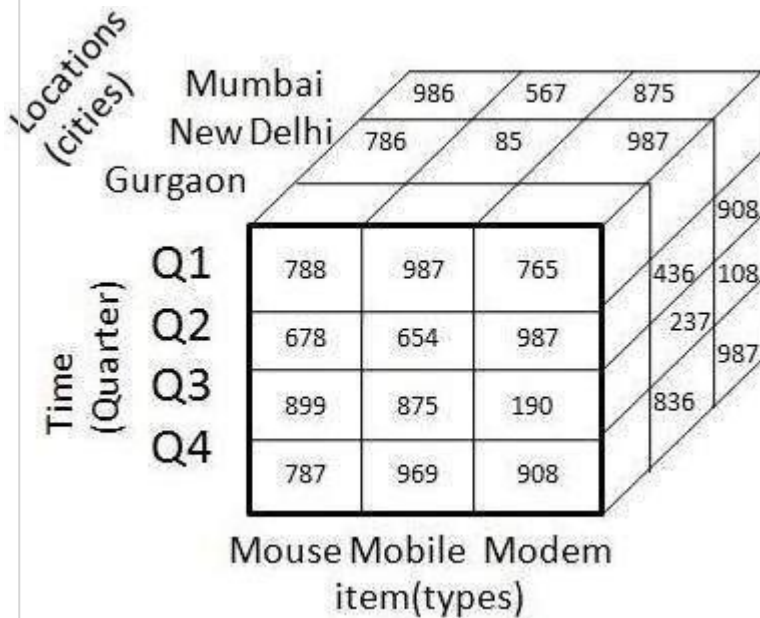
The following table represents the 2-D view of Sales Data for a company with respect to time, item, and location dimensions.

Location="New Delhi"				
Time(quarter)	Item(type)			
	Entertainment	Keyboard	Mobile	Locks
Q1	500	700	10	300
Q2	769	765	30	476
Q3	987	489	18	659
Q4	666	976	40	539

But here in this 2-D table, we have records with respect to time and item only. The sales for New Delhi are shown with respect to time, and item dimensions according to type of items sold. If we want to view the sales data with one more dimension, say, the location dimension, then the 3-D view would be useful. The 3-D view of the sales data with respect to time, item, and location is shown in the table below –

Time	Location="Gurgaon"			Location="New Delhi"			Location="Mumbai"		
	Item			Item			Item		
	Mouse	Mobile	Modem	Mouse	Mobile	Modem	Mouse	Mobile	Modem
Q1	788	987	765	786	85	987	986	567	875
Q2	678	654	987	659	786	436	980	876	908
Q3	899	875	190	983	909	237	987	100	1089
Q4	787	969	908	537	567	836	837	926	987

The above 3-D table can be represented as 3-D data cube as shown in the following figure –



## Data Mart

Data marts contain a subset of organization-wide data that is valuable to specific groups of people in an organization. In other words, a data mart contains only those data that is specific to a particular group. For example, the marketing data mart may contain only data related to items, customers, and sales. Data marts are confined to subjects.

### Points to Remember About Data Marts

Windows-based or Unix/Linux-based servers are used to implement data marts. They are implemented on low-cost servers.

The implementation cycle of a data mart is measured in short periods of time, i.e., in weeks rather than months or years.

The life cycle of data marts may be complex in the long run, if their planning and design are not organization-wide.

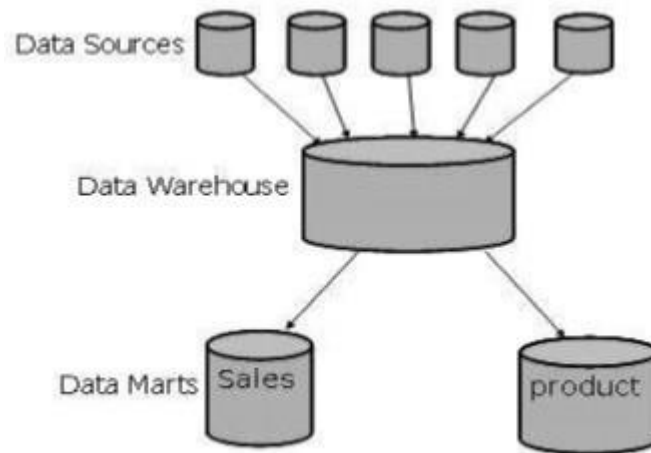
Data marts are small in size.

Data marts are customized by department.

The source of a data mart is departmentally structured data warehouse.

Data marts are flexible.

The following figure shows a graphical representation of data marts.



## Virtual Warehouse

The view over an operational data warehouse is known as virtual warehouse. It is easy to build a virtual warehouse. Building a virtual warehouse requires excess capacity on operational database servers.

## Data Warehousing - Delivery Process

A data warehouse is never static; it evolves as the business expands. As the business evolves, its requirements keep changing and therefore a data warehouse must be designed to ride with these changes. Hence a data warehouse system needs to be flexible.

Ideally there should be a delivery process to deliver a data warehouse. However data warehouse projects normally suffer from various issues that make it difficult to complete tasks and deliverables in the strict and ordered fashion demanded by the waterfall method. Most of the times, the requirements are not understood completely. The architectures, designs, and build components can be completed only after gathering and studying all the requirements.

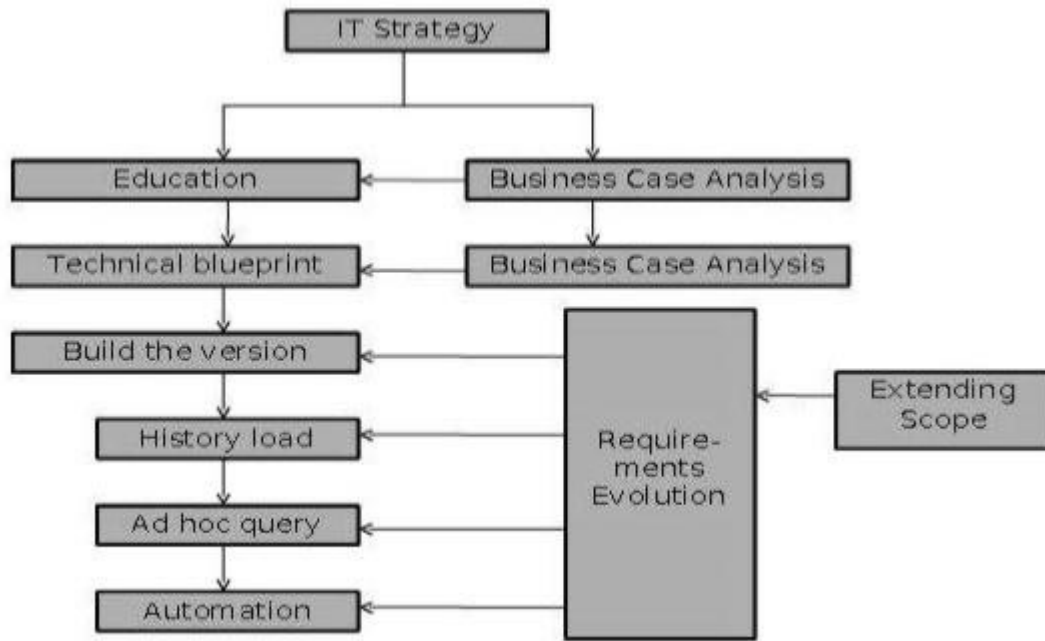
## Delivery Method

The delivery method is a variant of the joint application development approach adopted for the delivery of a data warehouse. We have staged the data warehouse delivery process to minimize risks. The approach that we will discuss here does not reduce the overall delivery time-scales but ensures the business benefits are delivered incrementally through the development process.

**Note** – The delivery process is broken into phases to reduce the project and delivery risk.

The following diagram explains the stages in the delivery process –





## IT Strategy

Data warehouse are strategic investments that require a business process to generate benefits. IT Strategy is required to procure and retain funding for the project.

## Business Case

The objective of business case is to estimate business benefits that should be derived from using a data warehouse. These benefits may not be quantifiable but the projected benefits need to be clearly stated. If a data warehouse does not have a clear business case, then the business tends to suffer from credibility problems at some stage during the delivery process. Therefore in data warehouse projects, we need to understand the business case for investment.

## Education and Prototyping

Organizations experiment with the concept of data analysis and educate themselves on the value of having a data warehouse before settling for a solution. This is addressed by prototyping. It helps in understanding the feasibility and benefits of a data warehouse. The prototyping activity on a small scale can promote educational process as long as –

- The prototype addresses a defined technical objective.

- The prototype can be thrown away after the feasibility concept has been shown.

- The activity addresses a small subset of eventual data content of the data warehouse.



The activity timescale is non-critical.

The following points are to be kept in mind to produce an early release and deliver business benefits.

Identify the architecture that is capable of evolving.

Focus on business requirements and technical blueprint phases.

Limit the scope of the first build phase to the minimum that delivers business benefits.

Understand the short-term and medium-term requirements of the data warehouse.

## Business Requirements

To provide quality deliverables, we should make sure the overall requirements are understood. If we understand the business requirements for both short-term and medium-term, then we can design a solution to fulfil short-term requirements. The short-term solution can then be grown to a full solution.

The following aspects are determined in this stage –

The business rule to be applied on data.

The logical model for information within the data warehouse.

The query profiles for the immediate requirement.

The source systems that provide this data.

## Technical Blueprint

This phase need to deliver an overall architecture satisfying the long term requirements. This phase also deliver the components that must be implemented in a short term to derive any business benefit. The blueprint need to identify the followings.

The overall system architecture.

The data retention policy.

The backup and recovery strategy.

The server and data mart architecture.

The capacity plan for hardware and infrastructure.

The components of database design.

## Building the Version

In this stage, the first production deliverable is produced. This production deliverable is the smallest component of a data warehouse. This smallest component adds business benefit.

### History Load

This is the phase where the remainder of the required history is loaded into the data warehouse. In this phase, we do not add new entities, but additional physical tables would probably be created to store increased data volumes.

Let us take an example. Suppose the build version phase has delivered a retail sales analysis data warehouse with 2 months' worth of history. This information will allow the user to analyze only the recent trends and address the short-term issues. The user in this case cannot identify annual and seasonal trends. To help him do so, last 2 years' sales history could be loaded from the archive. Now the 40GB data is extended to 400GB.

**Note** – The backup and recovery procedures may become complex, therefore it is recommended to perform this activity within a separate phase.

### Ad hoc Query

In this phase, we configure an ad hoc query tool that is used to operate a data warehouse. These tools can generate the database query.

**Note** – It is recommended not to use these access tools when the database is being substantially modified.

### Automation

In this phase, operational management processes are fully automated. These would include –

- Transforming the data into a form suitable for analysis.

- Monitoring query profiles and determining appropriate aggregations to maintain system performance.

- Extracting and loading data from different source systems.

- Generating aggregations from predefined definitions within the data warehouse.

- Backing up, restoring, and archiving the data.

### Extending Scope

In this phase, the data warehouse is extended to address a new set of business requirements. The scope can be extended in two ways –

By loading additional data into the data warehouse.

By introducing new data marts using the existing information.

**Note** – This phase should be performed separately, since it involves substantial efforts and complexity.

## Requirements Evolution

From the perspective of delivery process, the requirements are always changeable. They are not static. The delivery process must support this and allow these changes to be reflected within the system.

This issue is addressed by designing the data warehouse around the use of data within business processes, as opposed to the data requirements of existing queries.

The architecture is designed to change and grow to match the business needs, the process operates as a pseudo-application development process, where the new requirements are continually fed into the development activities and the partial deliverables are produced. These partial deliverables are fed back to the users and then reworked ensuring that the overall system is continually updated to meet the business needs.

## Data Warehousing - System Processes

We have a fixed number of operations to be applied on the operational databases and we have well-defined techniques such as **use normalized data, keep table small**, etc. These techniques are suitable for delivering a solution. But in case of decision-support systems, we do not know what query and operation needs to be executed in future. Therefore techniques applied on operational databases are not suitable for data warehouses.

In this chapter, we will discuss how to build data warehousing solutions on top open-system technologies like Unix and relational databases.

## Process Flow in Data Warehouse

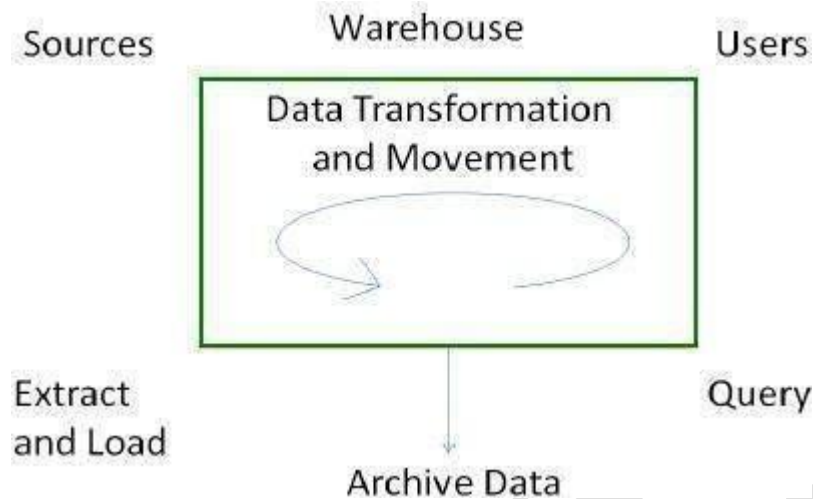
There are four major processes that contribute to a data warehouse –

Extract and load the data.

Cleaning and transforming the data.

Backup and archive the data.

Managing queries and directing them to the appropriate data sources.



## Extract and Load Process

Data extraction takes data from the source systems. Data load takes the extracted data and loads it into the data warehouse.

**Note** – Before loading the data into the data warehouse, the information extracted from the external sources must be reconstructed.

## Controlling the Process

Controlling the process involves determining when to start data extraction and the consistency check on data. Controlling process ensures that the tools, the logic modules, and the programs are executed in correct sequence and at correct time.

## When to Initiate Extract

Data needs to be in a consistent state when it is extracted, i.e., the data warehouse should represent a single, consistent version of the information to the user.

For example, in a customer profiling data warehouse in telecommunication sector, it is illogical to merge the list of customers at 8 pm on Wednesday from a customer database with the customer subscription events up to 8 pm on Tuesday. This would mean that we are finding the customers for whom there are no associated subscriptions.

## Loading the Data

After extracting the data, it is loaded into a temporary data store where it is cleaned up and made consistent.

**Note** – Consistency checks are executed only when all the data sources have been loaded into the temporary data store.

## Clean and Transform Process

Once the data is extracted and loaded into the temporary data store, it is time to perform Cleaning and Transforming. Here is the list of steps involved in Cleaning and Transforming –

Clean and transform the loaded data into a structure

Partition the data

Aggregation

## Clean and Transform the Loaded Data into a Structure

Cleaning and transforming the loaded data helps speed up the queries. It can be done by making the data consistent –

within itself.

with other data within the same data source.

with the data in other source systems.

with the existing data present in the warehouse.

Transforming involves converting the source data into a structure. Structuring the data increases the query performance and decreases the operational cost. The data contained in a data warehouse must be transformed to support performance requirements and control the ongoing operational costs.

## Partition the Data

It will optimize the hardware performance and simplify the management of data warehouse. Here we partition each fact table into multiple separate partitions.

## Aggregation

Aggregation is required to speed up common queries. Aggregation relies on the fact that most common queries will analyze a subset or an aggregation of the detailed data.

## Backup and Archive the Data

In order to recover the data in the event of data loss, software failure, or hardware failure, it is necessary to keep regular back ups. Archiving involves removing the old data from the system in a format that allow it to be quickly restored whenever required.

For example, in a retail sales analysis data warehouse, it may be required to keep data for 3 years with the latest 6 months data being kept online. In such as scenario, there is often a requirement to be able to do month-on-month comparisons for this year and last year. In this case, we require some data to be restored from the archive.

## Query Management Process

KAHE

This process performs the following functions –

manages the queries.

helps speed up the execution time of queries.

directs the queries to their most effective data sources.

ensures that all the system sources are used in the most effective way.

monitors actual query profiles.

The information generated in this process is used by the warehouse management process to determine which aggregations to generate. This process does not generally operate during the regular load of information into data warehouse.

## Data Warehousing - Architecture

In this chapter, we will discuss the business analysis framework for the data warehouse design and architecture of a data warehouse.

### Business Analysis Framework

The business analyst gets the information from the data warehouses to measure the performance and make critical adjustments in order to win over other business holders in the market. Having a data warehouse offers the following advantages –

Since a data warehouse can gather information quickly and efficiently, it can enhance business productivity.

A data warehouse provides us a consistent view of customers and items, hence, it helps us manage customer relationship.

A data warehouse also helps in bringing down the costs by tracking trends, patterns over a long period in a consistent and reliable manner.

To design an effective and efficient data warehouse, we need to understand and analyze the business needs and construct a **business analysis framework**. Each person has different views regarding the design of a data warehouse. These views are as follows –

**The top-down view** – This view allows the selection of relevant information needed for a data warehouse.

**The data source view** – This view presents the information being captured, stored, and managed by the operational system.

**The data warehouse view** – This view includes the fact tables and dimension

tables. It represents the information stored inside the data warehouse.

**The business query view** – It is the view of the data from the viewpoint of the end-user.

## Three-Tier Data Warehouse Architecture

Generally a data warehouses adopts a three-tier architecture. Following are the three tiers of the data warehouse architecture.

**Bottom Tier** – The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back end tools and utilities to feed data into the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.

**Middle Tier** – In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.

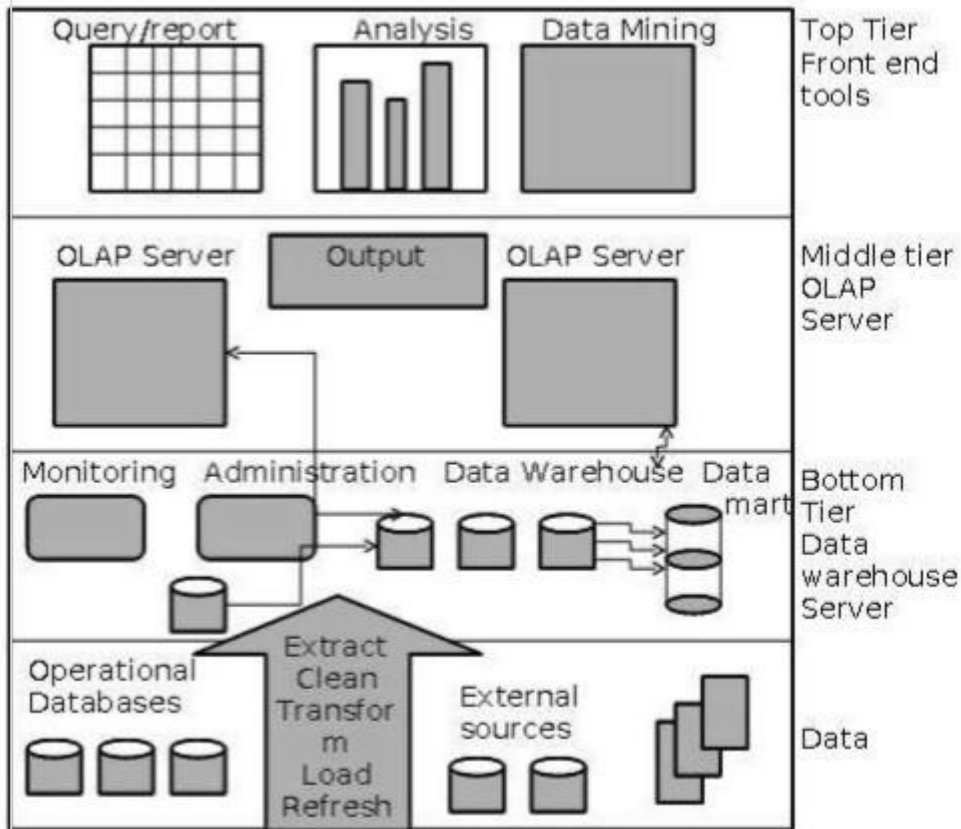
By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.

By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.

**Top-Tier** – This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

The following diagram depicts the three-tier architecture of data warehouse –





## Data Warehouse Models

From the perspective of data warehouse architecture, we have the following data warehouse models –

- Virtual Warehouse
- Data mart
- Enterprise Warehouse

### Virtual Warehouse

The view over an operational data warehouse is known as a virtual warehouse. It is easy to build a virtual warehouse. Building a virtual warehouse requires excess capacity on operational database servers.

### Data Mart

Data mart contains a subset of organization-wide data. This subset of data is valuable to specific groups of an organization.

In other words, we can claim that data marts contain data specific to a particular group. For example, the marketing data mart may contain data related to items, customers, and sales. Data marts are confined to subjects.

Points to remember about data marts –

Window-based or Unix/Linux-based servers are used to implement data marts. They are implemented on low-cost servers.

The implementation data mart cycles is measured in short periods of time, i.e., in weeks rather than months or years.

The life cycle of a data mart may be complex in long run, if its planning and design are not organization-wide.

Data marts are small in size.

Data marts are customized by department.

The source of a data mart is departmentally structured data warehouse.

Data mart are flexible.

## Enterprise Warehouse

An enterprise warehouse collects all the information and the subjects spanning an entire organization

It provides us enterprise-wide data integration.

The data is integrated from operational systems and external information providers.

This information can vary from a few gigabytes to hundreds of gigabytes, terabytes or beyond.

## Load Manager

This component performs the operations required to extract and load process.

The size and complexity of the load manager varies between specific solutions from one data warehouse to other.

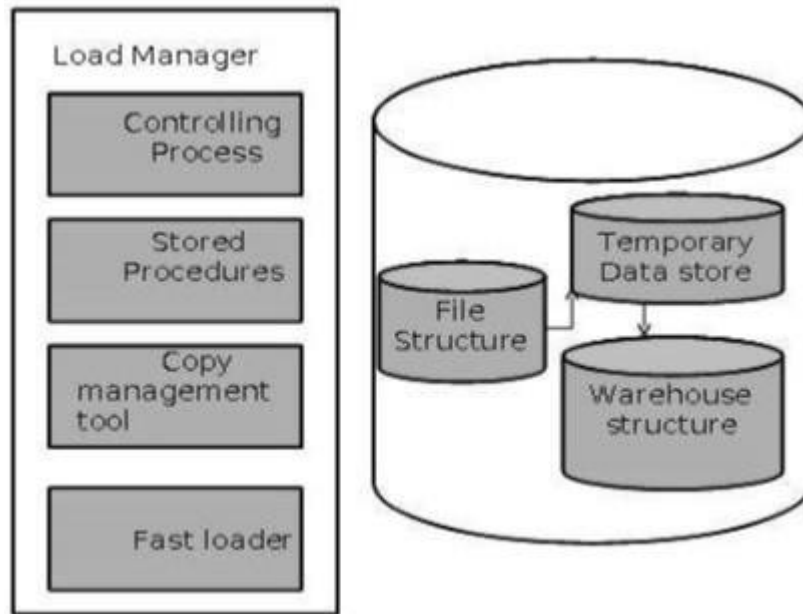
## Load Manager Architecture

The load manager performs the following functions –

Extract the data from source system.

Fast Load the extracted data into temporary data store.

Perform simple transformations into structure similar to the one in the data warehouse.



### Extract Data from Source

The data is extracted from the operational databases or the external information providers. Gateways is the application programs that are used to extract data. It is supported by underlying DBMS and allows client program to generate SQL to be executed at a server. Open Database Connection(ODBC), Java Database Connection (JDBC), are examples of gateway.

### Fast Load

In order to minimize the total load window the data need to be loaded into the warehouse in the fastest possible time.

The transformations affects the speed of data processing.

It is more effective to load the data into relational database prior to applying transformations and checks.

Gateway technology proves to be not suitable, since they tend not be performant when large data volumes are involved.

### Simple Transformations

While loading it may be required to perform simple transformations. After this has been completed we are in position to do the complex checks. Suppose we are loading the EPOS sales transaction we need to perform the following checks:

Strip out all the columns that are not required within the warehouse.

Convert all the values to required data types.

# Warehouse Manager

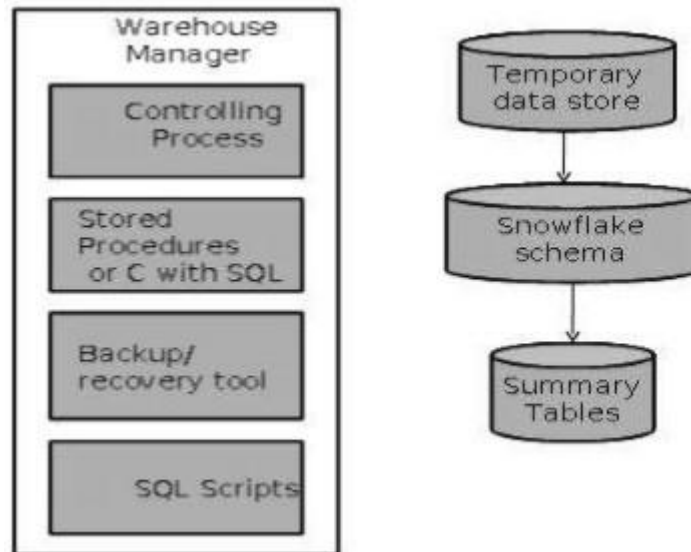
A warehouse manager is responsible for the warehouse management process. It consists of third-party system software, C programs, and shell scripts.

The size and complexity of warehouse managers varies between specific solutions.

## Warehouse Manager Architecture

A warehouse manager includes the following –

- The controlling process
- Stored procedures or C with SQL
- Backup/Recovery tool
- SQL Scripts



## Operations Performed by Warehouse Manager

A warehouse manager analyzes the data to perform consistency and referential integrity checks.

Creates indexes, business views, partition views against the base data.

Generates new aggregations and updates existing aggregations. Generates normalizations.

Transforms and merges the source data into the published data warehouse.

Backup the data in the data warehouse.

Archives the data that has reached the end of its captured life.

**Note** – A warehouse Manager also analyzes query profiles to determine index and

aggregations are appropriate.

## Query Manager

Query manager is responsible for directing the queries to the suitable tables.

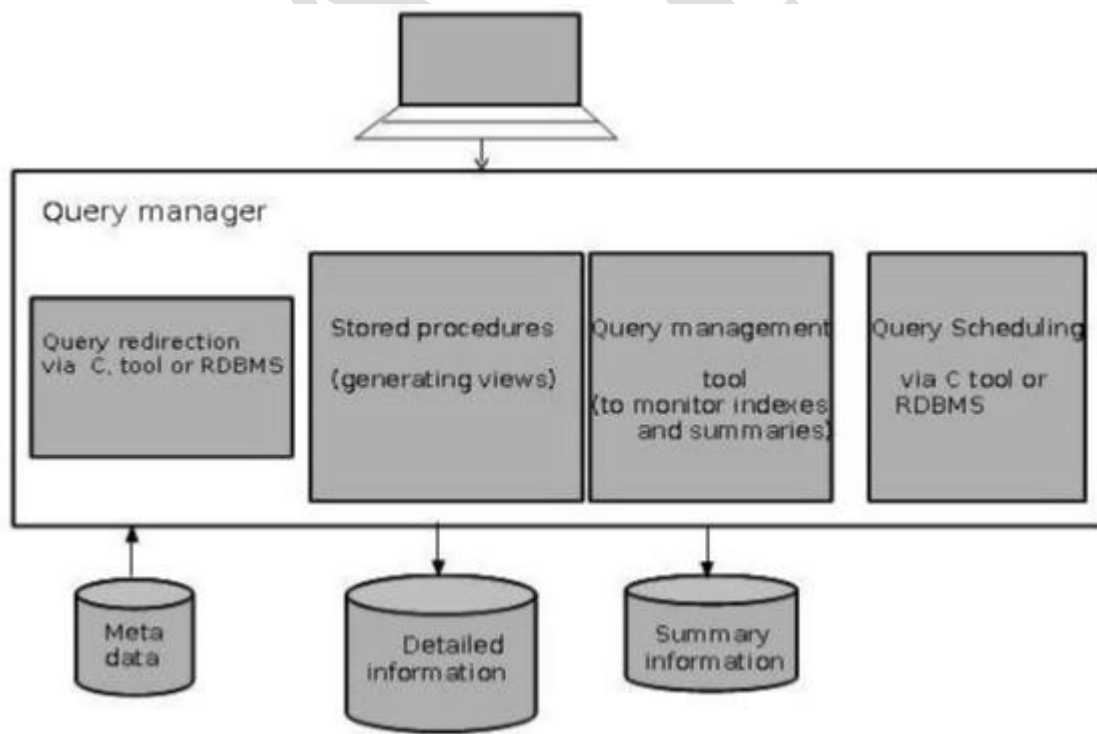
By directing the queries to appropriate tables, the speed of querying and response generation can be increased.

Query manager is responsible for scheduling the execution of the queries posed by the user.

### Query Manager Architecture

The following screenshot shows the architecture of a query manager. It includes the following:

- Query redirection via C tool or RDBMS
- Stored procedures
- Query management tool
- Query scheduling via C tool or RDBMS
- Query scheduling via third-party software

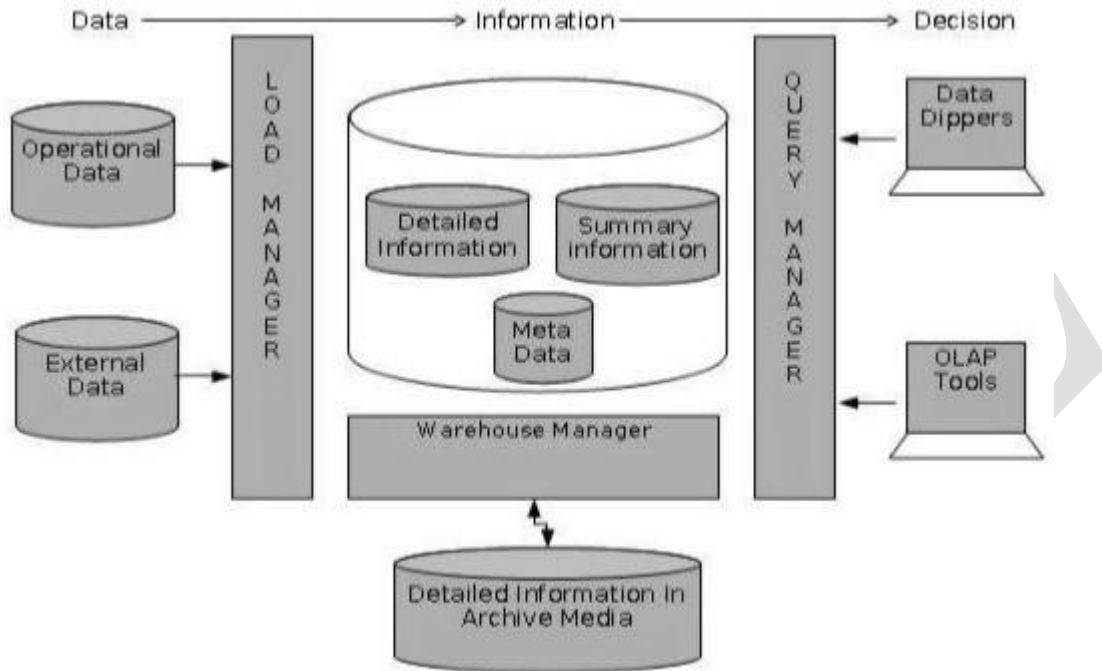


## Detailed Information

Detailed information is not kept online, rather it is aggregated to the next level of detail

and then archived to tape. The detailed information part of data warehouse keeps the detailed information in the starflake schema. Detailed information is loaded into the data warehouse to supplement the aggregated data.

The following diagram shows a pictorial impression of where detailed information is stored and how it is used.



**Note** – If detailed information is held offline to minimize disk storage, we should make sure that the data has been extracted, cleaned up, and transformed into starflake schema before it is archived.

## Summary Information

Summary Information is a part of data warehouse that stores predefined aggregations. These aggregations are generated by the warehouse manager. Summary Information must be treated as transient. It changes on-the-go in order to respond to the changing query profiles.

The points to note about summary information are as follows –

- Summary information speeds up the performance of common queries.

- It increases the operational cost.

- It needs to be updated whenever new data is loaded into the data warehouse.

- It may not have been backed up, since it can be generated fresh from the detailed information.

# Data Warehousing - OLAP

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. This chapter cover the types of OLAP, operations on OLAP, difference between OLAP, and statistical databases and OLTP.

## Types of OLAP Servers

We have four types of OLAP servers –

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)
- Specialized SQL Servers

## Relational OLAP

ROLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.

ROLAP includes the following –

- Implementation of aggregation navigation logic.
- Optimization for each DBMS back end.
- Additional tools and services.

## Multidimensional OLAP

MOLAP uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP server use two levels of data storage representation to handle dense and sparse data sets.

## Hybrid OLAP

Hybrid OLAP is a combination of both ROLAP and MOLAP. It offers higher scalability of ROLAP and faster computation of MOLAP. HOLAP servers allows to store the large data volumes of detailed information. The aggregations are stored separately in MOLAP store.

## Specialized SQL Servers

Specialized SQL servers provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

## OLAP Operations

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

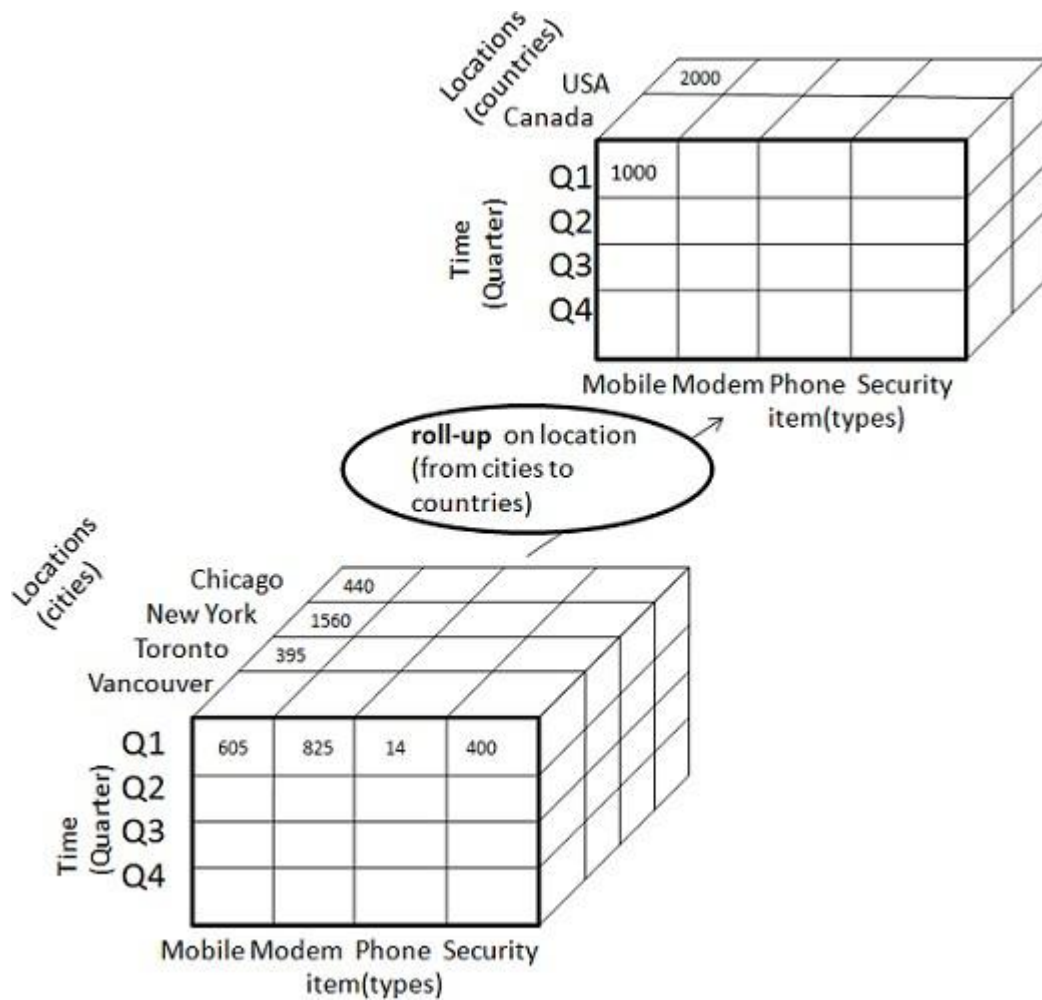
Here is the list of OLAP operations –

- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

### Roll-up

Roll-up performs aggregation on a data cube in any of the following ways –





By climbing up a concept hierarchy for a dimension

By dimension reduction The following diagram illustrates how roll-up works.

Roll-up is performed by climbing up a concept hierarchy for the dimension location.

Initially the concept hierarchy was "street < city < province < country".

On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.

The data is grouped into cities rather than countries.

When roll-up is performed, one or more dimensions from the data cube are removed.

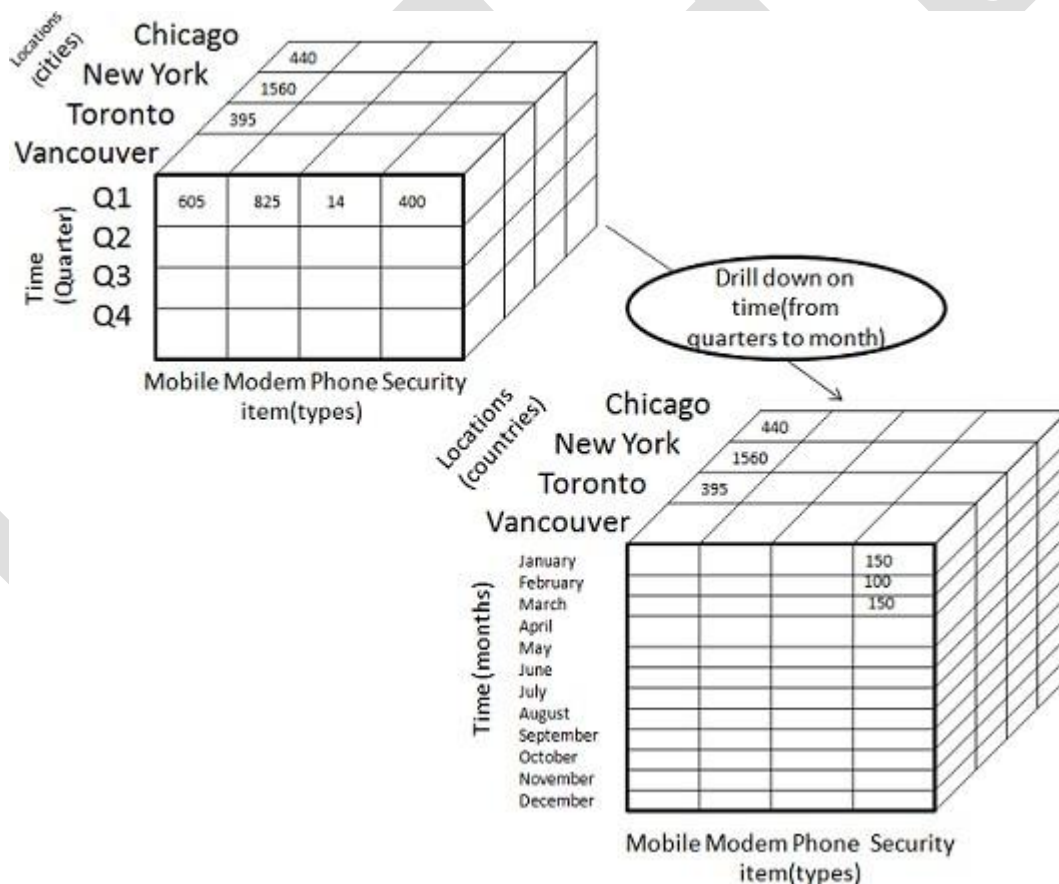
## Drill-down

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways –

By stepping down a concept hierarchy for a dimension

By introducing a new dimension.

The following diagram illustrates how drill-down works –



Drill-down is performed by stepping down a concept hierarchy for the dimension time.

Initially the concept hierarchy was "day < month < quarter < year."

On drilling down, the time dimension is descended from the level of quarter to the

level of month.

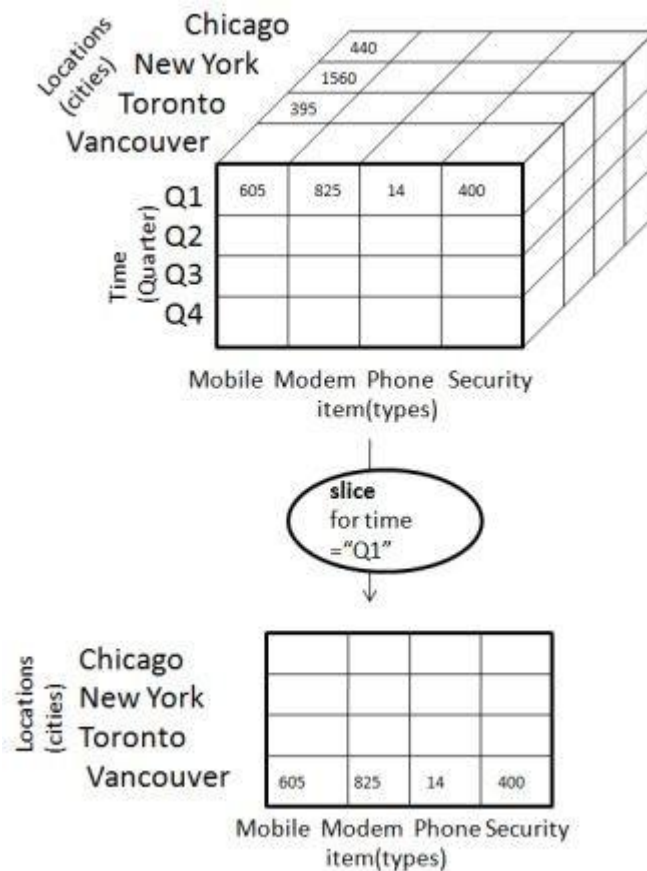
KAHE

When drill-down is performed, one or more dimensions from the data cube are added.

It navigates the data from less detailed data to highly detailed data.

## Slice

The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.

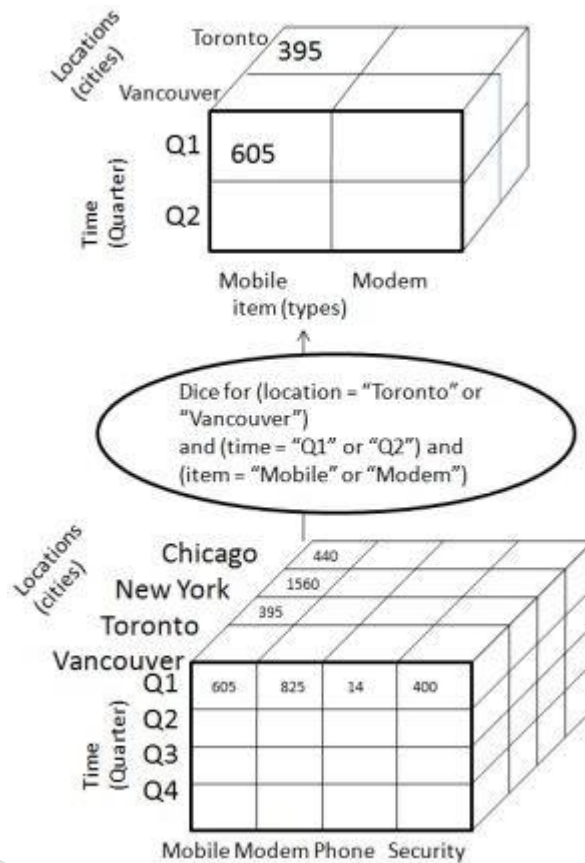


Here Slice is performed for the dimension "time" using the criterion time = "Q1".

It will form a new sub-cube by selecting one or more dimensions.

## Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.



The dice operation on the cube based on the following selection criteria involves three dimensions.

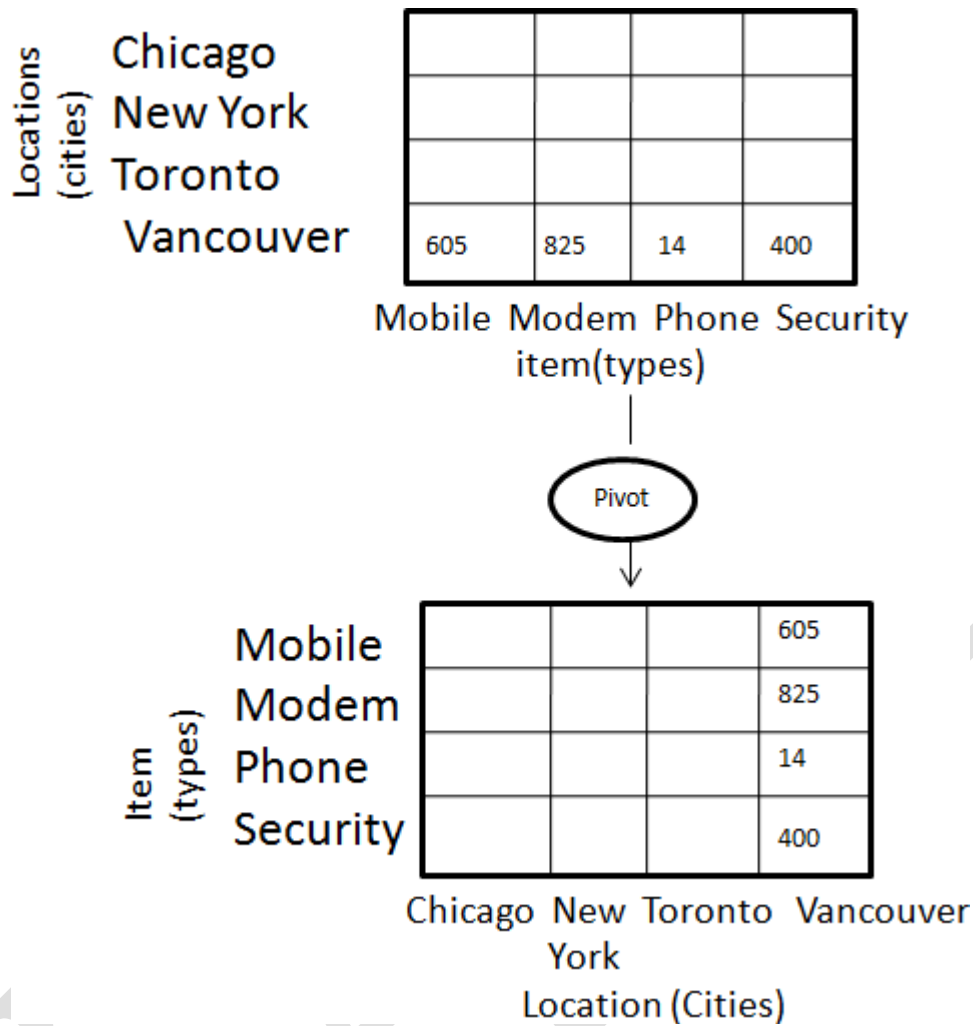
(location = "Toronto" or "Vancouver")

(time = "Q1" or "Q2")

(item = "Mobile" or "Modem")

## Pivot

The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.



## OLAP vs OLTP

Sr.No.	Data Warehouse (OLAP)	Operational Database (OLTP)
1	Involves historical processing of information.	Involves day-to-day processing.
2	OLAP systems are used by knowledge workers such as executives, managers and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
3	Useful in analyzing the business.	Useful in running the business.
4	It focuses on Information out.	It focuses on Data in.

5	Based on Star Schema, Snowflake, Schema and Fact Constellation Schema.	Based on Entity Relationship Model.
6	Contains historical data.	Contains current data.
7	Provides summarized and consolidated data.	Provides primitive and highly detailed data.
8	Provides summarized and multidimensional view of data.	Provides detailed and flat relational view of data.
9	Number of users is in hundreds.	Number of users is in thousands.
10	Number of records accessed is in millions.	Number of records accessed is in tens.
11	Database size is from 100 GB to 1 TB	Database size is from 100 MB to 1 GB.
12	Highly flexible.	Provides high performance.

## Data Warehousing - Relational OLAP

Relational OLAP servers are placed between relational back-end server and client front-end tools. To store and manage the warehouse data, the relational OLAP uses relational or extended-relational DBMS.

ROLAP includes the following –

- Implementation of aggregation navigation logic
- Optimization for each DBMS back-end
- Additional tools and services

## Points to Remember

ROLAP servers are highly scalable.

ROLAP tools analyze large volumes of data across multiple dimensions.

ROLAP tools store and analyze highly volatile and changeable data.

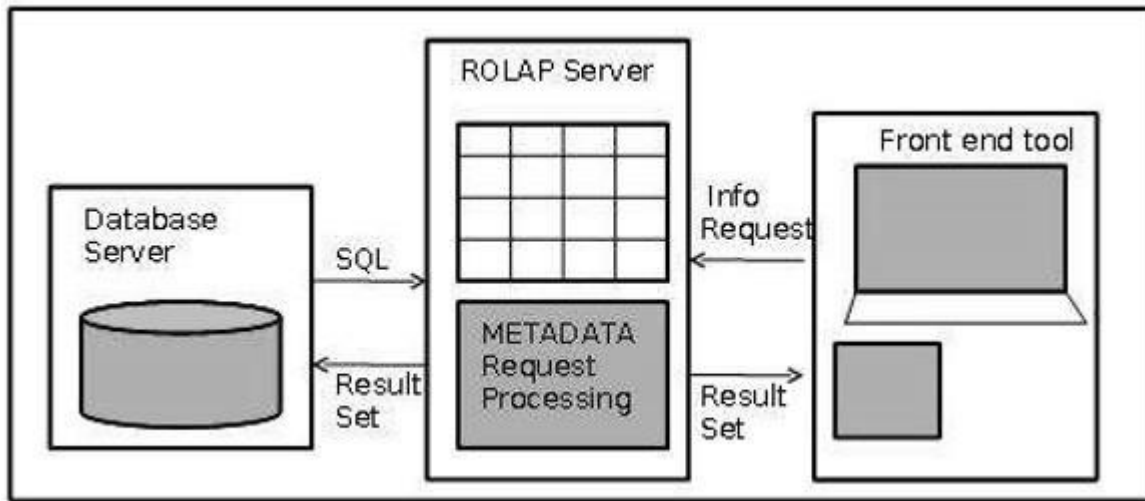
## Relational OLAP Architecture

ROLAP includes the following components –

- Database server

ROLAP server

Front-end tool.



## Advantages

ROLAP servers can be easily used with existing RDBMS.

Data can be stored efficiently, since no zero facts can be stored.

ROLAP tools do not use pre-calculated data cubes.

DSS server of micro-strategy adopts the ROLAP approach.

## Disadvantages

Poor query performance.

Some limitations of scalability depending on the technology architecture that is utilized.

## Data Warehousing - Multidimensional OLAP

Multidimensional OLAP (MOLAP) uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the dataset is sparse. Therefore, many MOLAP servers use two levels of data storage representation to handle dense and sparse datasets.

## Points to Remember –

MOLAP tools process information with consistent response time regardless of level of summarizing or calculations selected.

MOLAP tools need to avoid many of the complexities of creating a relational



database to store data for analysis.

MOLAP tools need fastest possible performance.

MOLAP server adopts two level of storage representation to handle dense and sparse data sets.

Denser sub-cubes are identified and stored as array structure.

Sparse sub-cubes employ compression technology.

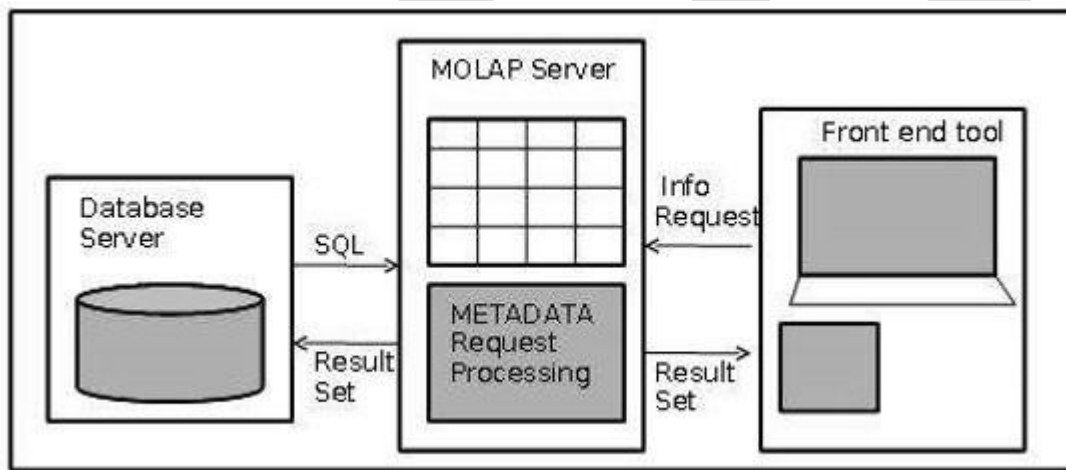
## MOLAP Architecture

MOLAP includes the following components –

Database server.

MOLAP server.

Front-end tool.



## Advantages

MOLAP allows fastest indexing to the pre-computed summarized data.

Helps the users connected to a network who need to analyze larger, less-defined data.

Easier to use, therefore MOLAP is suitable for inexperienced users.

## Disadvantages

MOLAP are not capable of containing detailed data.

The storage utilization may be low if the data set is sparse.

## MOLAP vs ROLAP

Sr.No.	MOLAP	ROLAP
1	Information retrieval is fast.	Information retrieval is comparatively slow.
2	Uses sparse array to store data-sets.	Uses relational table.
3	MOLAP is best suited for inexperienced users, since it is very easy to use.	ROLAP is best suited for experienced users.
4	Maintains a separate database for data cubes.	It may not require space other than available in the Data warehouse.
5	DBMS facility is weak.	DBMS facility is strong.

## Data Warehousing - Schemas

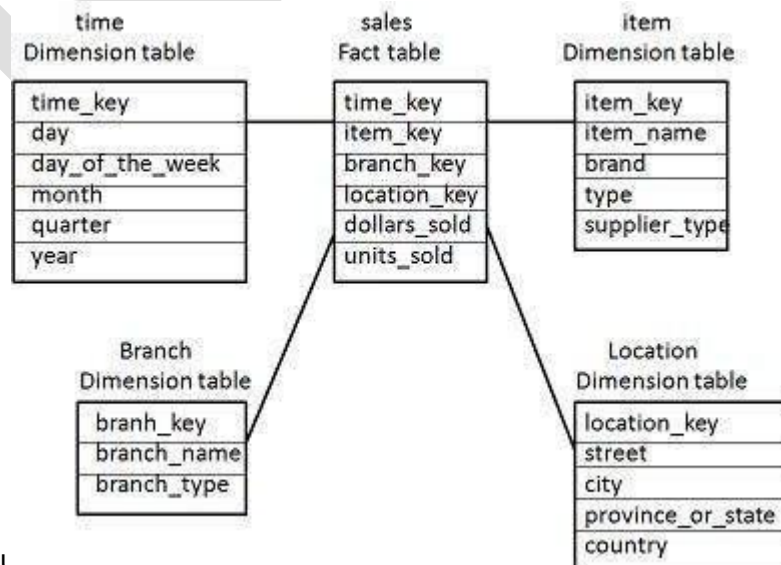
Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse.

### Star Schema

Each dimension in a star schema is represented with only one-dimension table.

This dimension table contains the set of attributes.

The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.



There is a fact table at the center. It contains the keys to each of four dimensions.

The fact table also contains the attributes, namely dollars sold and units sold.

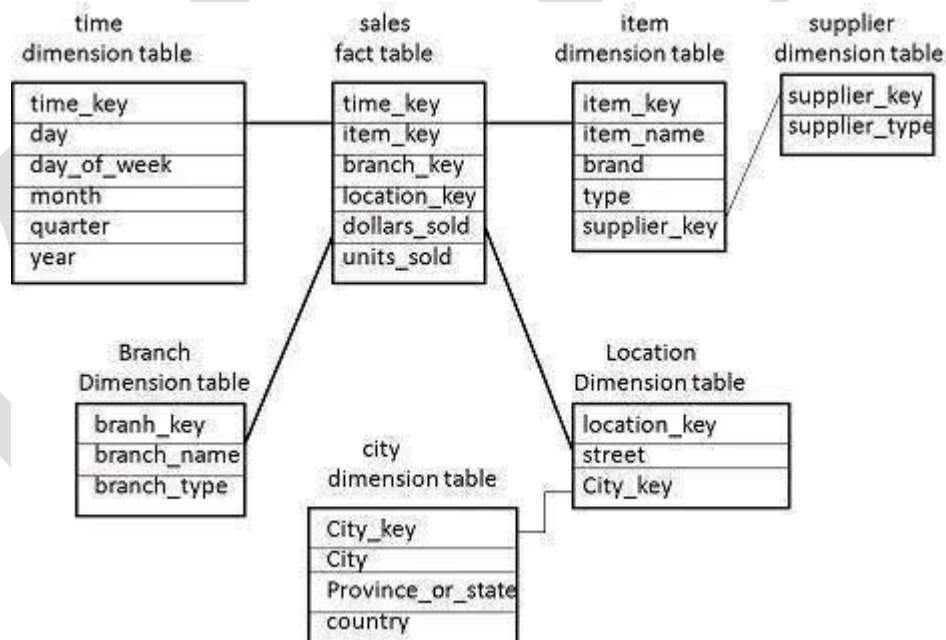
**Note** – Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location\_key, street, city, province\_or\_state, country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province\_or\_state and country.

## Snowflake Schema

Some dimension tables in the Snowflake schema are normalized.

The normalization splits up the data into additional tables.

Unlike Star schema, the dimensions table in a snowflake schema are normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.



Now the item dimension table contains the attributes item\_key, item\_name, type, brand, and supplier-key.

The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier\_key and supplier\_type.

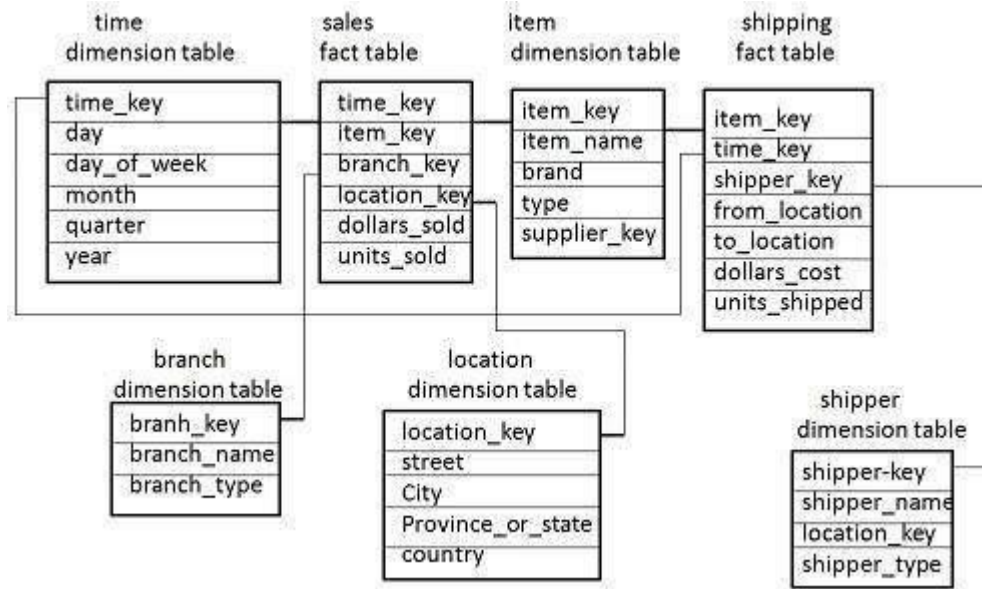
**Note** – Due to normalization in the Snowflake schema, the redundancy is reduced and

therefore, it becomes easy to maintain and the save storage space.

## Fact Constellation Schema

A fact constellation has multiple fact tables. It is also known as galaxy schema.

The following diagram shows two fact tables, namely sales and shipping.



The sales fact table is same as that in the star schema.

The shipping fact table has the five dimensions, namely item\_key, time\_key, shipper\_key, from\_location, to\_location.

The shipping fact table also contains two measures, namely dollars sold and units sold.

It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

## Schema Definition

Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

### Syntax for Cube Definition

```
define cube < cube_name > [ < dimension-list > ]: < measure_list >
```

## Syntax for Dimension Definition

```
define dimension < dimension_name > as ( < attribute_or_dimension_list > )
```

## Star Schema Definition

The star schema that we have discussed can be defined using Data Mining Query Language (DMQL) as follows –

```
define cube sales star [time, item, branch, location]:

dollars sold = sum(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)
```

## Snowflake Schema Definition

Snowflake schema can be defined using DMQL as follows –

```
define cube sales snowflake [time, item, branch, location]:

dollars sold = sum(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, city, province or state, country))
```

## Fact Constellation Schema Definition

Fact constellation schema can be defined using DMQL as follows –

```
define cube sales [time, item, branch, location]:

dollars sold = sum(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]:

dollars cost = sum(cost in dollars), units 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
```

## Data Warehousing - Partitioning Strategy

Partitioning is done to enhance performance and facilitate easy management of data. Partitioning also helps in balancing the various requirements of the system. It optimizes the hardware performance and simplifies the management of data warehouse by partitioning each fact table into multiple separate partitions. In this chapter, we will discuss different partitioning strategies.

### Why is it Necessary to Partition?

Partitioning is important for the following reasons –

- For easy management,
- To assist backup/recovery,
- To enhance performance.

#### For Easy Management

The fact table in a data warehouse can grow up to hundreds of gigabytes in size. This huge size of fact table is very hard to manage as a single entity. Therefore it needs partitioning.

#### To Assist Backup/Recovery

If we do not partition the fact table, then we have to load the complete fact table with all the data. Partitioning allows us to load only as much data as is required on a regular basis. It reduces the time to load and also enhances the performance of the system.

**Note** – To cut down on the backup size, all partitions other than the current partition can be marked as read-only. We can then put these partitions into a state where they cannot be modified. Then they can be backed up. It means only the current partition is to be backed up.

#### To Enhance Performance

By partitioning the fact table into sets of data, the query procedures can be enhanced. Query performance is enhanced because now the query scans only those partitions that are relevant. It does not have to scan the whole data.

## Horizontal Partitioning

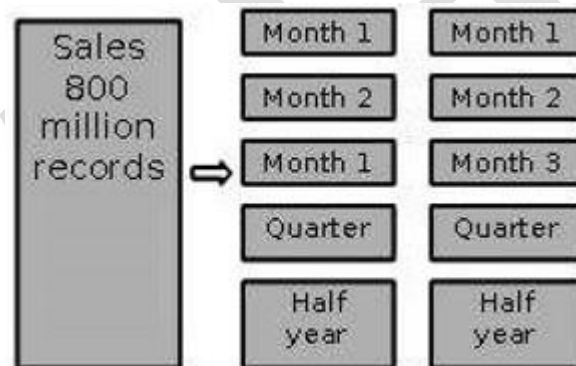
There are various ways in which a fact table can be partitioned. In horizontal partitioning, we have to keep in mind the requirements for manageability of the data warehouse.

### Partitioning by Time into Equal Segments

In this partitioning strategy, the fact table is partitioned on the basis of time period. Here each time period represents a significant retention period within the business. For example, if the user queries for **month to date data** then it is appropriate to partition the data into monthly segments. We can reuse the partitioned tables by removing the data in them.

### Partition by Time into Different-sized Segments

This kind of partition is done where the aged data is accessed infrequently. It is implemented as a set of small partitions for relatively current data, larger partition for inactive data.



### Points to Note

The detailed information remains available online.

The number of physical tables is kept relatively small, which reduces the operating cost.

This technique is suitable where a mix of data dipping recent history and data mining through entire history is required.

This technique is not useful where the partitioning profile changes on a regular basis, because repartitioning will increase the operation cost of data warehouse.

### Partition on a Different Dimension

The fact table can also be partitioned on the basis of dimensions other than time such as product group, region, supplier, or any other dimension. Let's have an example.

Suppose a market function has been structured into distinct regional departments like on a

**state by state** basis. If each region wants to query on information captured within its region, it would prove to be more effective to partition the fact table into regional partitions. This will cause the queries to speed up because it does not require to scan information that is not relevant.

### Points to Note

The query does not have to scan irrelevant data which speeds up the query process.

This technique is not appropriate where the dimensions are unlikely to change in future. So, it is worth determining that the dimension does not change in future.

If the dimension changes, then the entire fact table would have to be repartitioned.

**Note** – We recommend to perform the partition only on the basis of time dimension, unless you are certain that the suggested dimension grouping will not change within the life of the data warehouse.

### Partition by Size of Table

When there are no clear basis for partitioning the fact table on any dimension, then we should **partition the fact table on the basis of their size**. We can set the predetermined size as a critical point. When the table exceeds the predetermined size, a new table partition is created.

### Points to Note

This partitioning is complex to manage.

It requires metadata to identify what data is stored in each partition.

### Partitioning Dimensions

If a dimension contains large number of entries, then it is required to partition the dimensions. Here we have to check the size of a dimension.

Consider a large design that changes over time. If we need to store all the variations in order to apply comparisons, that dimension may be very large. This would definitely affect the response time.

### Round Robin Partitions

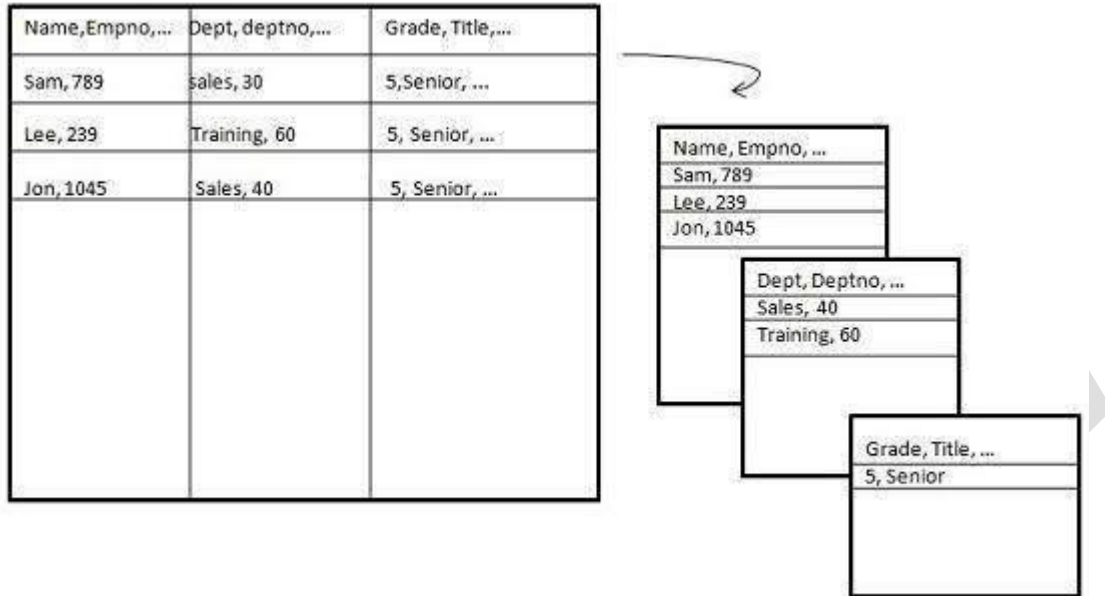
In the round robin technique, when a new partition is needed, the old one is archived. It uses metadata to allow user access tool to refer to the correct table partition.

This technique makes it easy to automate table management facilities within the data warehouse.



## Vertical Partition

Vertical partitioning, splits the data vertically. The following images depicts how vertical partitioning is done.



Vertical partitioning can be performed in the following two ways –

Normalization

Row Splitting

### Normalization

Normalization is the standard relational method of database organization. In this method, the rows are collapsed into a single row, hence it reduce space. Take a look at the following tables that show how normalization is performed.

Table before Normalization

Product_id	Qty	Value	sales_date	Store_id	Store_name	Location	Region
30	5	3.67	3-Aug-13	16	sunny	Bangalore	S
35	4	5.33	3-Sep-13	16	sunny	Bangalore	S
40	5	2.50	3-Sep-13	64	san	Mumbai	W
45	7	5.66	3-Sep-13	16	sunny	Bangalore	S

Table after Normalization

Store_id	Store_name	Location	Region
----------	------------	----------	--------

16	sunny	Bangalore	W
64	san	Mumbai	S

Product_id	Quantity	Value	sales_date	Store_id
30	5	3.67	3-Aug-13	16
35	4	5.33	3-Sep-13	16
40	5	2.50	3-Sep-13	64
45	7	5.66	3-Sep-13	16

### Row Splitting

Row splitting tends to leave a one-to-one map between partitions. The motive of row splitting is to speed up the access to large table by reducing its size.

**Note** – While using vertical partitioning, make sure that there is no requirement to perform a major join operation between two partitions.

### Identify Key to Partition

It is very crucial to choose the right partition key. Choosing a wrong partition key will lead to reorganizing the fact table. Let's have an example. Suppose we want to partition the following table.

```
Account_Txn_Table
transaction_id
account_id
transaction_type
value
transaction_date
region
branch_name
```

We can choose to partition on any key. The two possible keys could be

region

transaction\_date

Suppose the business is organized in 30 geographical regions and each region has different number of branches. That will give us 30 partitions, which is reasonable. This partitioning

is good enough because our requirements capture has shown that a vast majority of queries are restricted to the user's own business region.

If we partition by transaction\_date instead of region, then the latest transaction from every region will be in one partition. Now the user who wants to look at data within his own region has to query across multiple partitions.

Hence it is worth determining the right partitioning key.

## Data Warehousing - Metadata Concepts

### What is Metadata?

Metadata is simply defined as data about data. The data that is used to represent other data is known as metadata. For example, the index of a book serves as a metadata for the contents in the book. In other words, we can say that metadata is the summarized data that leads us to detailed data. In terms of data warehouse, we can define metadata as follows.

Metadata is the road-map to a data warehouse.

Metadata in a data warehouse defines the warehouse objects.

Metadata acts as a directory. This directory helps the decision support system to locate the contents of a data warehouse.

**Note** – In a data warehouse, we create metadata for the data names and definitions of a given data warehouse. Along with this metadata, additional metadata is also created for time-stamping any extracted data, the source of extracted data.

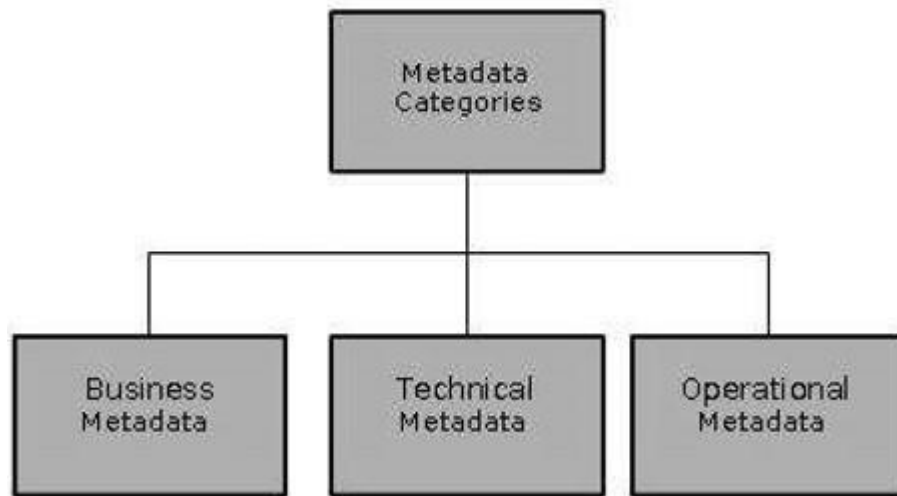
### Categories of Metadata

Metadata can be broadly categorized into three categories –

**Business Metadata** – It has the data ownership information, business definition, and changing policies.

**Technical Metadata** – It includes database system names, table and column names and sizes, data types and allowed values. Technical metadata also includes structural information such as primary and foreign key attributes and indices.

**Operational Metadata** – It includes currency of data and data lineage. Currency of data means whether the data is active, archived, or purged. Lineage of data means the history of data migrated and transformation applied on it.



## Role of Metadata

Metadata has a very important role in a data warehouse. The role of metadata in a warehouse is different from the warehouse data, yet it plays an important role. The various roles of metadata are explained below.

Metadata acts as a directory.

This directory helps the decision support system to locate the contents of the data warehouse.

Metadata helps in decision support system for mapping of data when data is transformed from operational environment to data warehouse environment.

Metadata helps in summarization between current detailed data and highly summarized data.

Metadata also helps in summarization between lightly detailed data and highly summarized data.

Metadata is used for query tools.

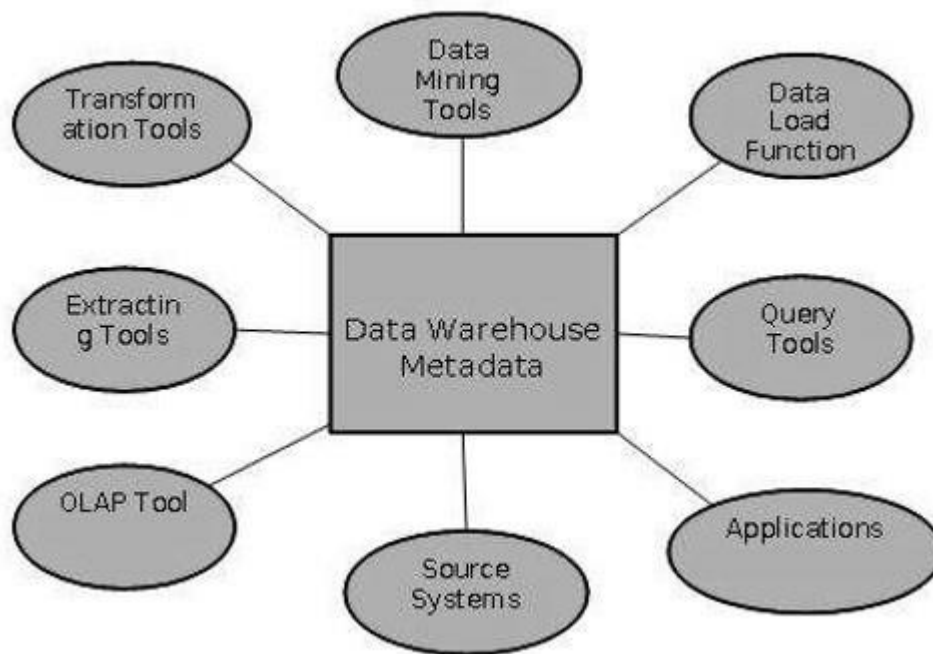
Metadata is used in extraction and cleansing tools.

Metadata is used in reporting tools.

Metadata is used in transformation tools.

Metadata plays an important role in loading functions.

The following diagram shows the roles of metadata.



## Metadata Repository

Metadata repository is an integral part of a data warehouse system. It has the following metadata –

**Definition of data warehouse** – It includes the description of structure of data warehouse. The description is defined by schema, view, hierarchies, derived data definitions, and data mart locations and contents.

**Business metadata** – It contains has the data ownership information, business definition, and changing policies.

**Operational Metadata** – It includes currency of data and data lineage. Currency of data means whether the data is active, archived, or purged. Lineage of data means the history of data migrated and transformation applied on it.

**Data for mapping from operational environment to data warehouse** – It includes the source databases and their contents, data extraction, data partition cleaning, transformation rules, data refresh and purging rules.

**Algorithms for summarization** – It includes dimension algorithms, data on granularity, aggregation, summarizing, etc.

## Challenges for Metadata Management

The importance of metadata can not be overstated. Metadata helps in driving the accuracy of reports, validates data transformation, and ensures the accuracy of calculations.

Metadata also enforces the definition of business terms to business end-users. With all

these uses of metadata, it also has its challenges. Some of the challenges are discussed below.

Metadata in a big organization is scattered across the organization. This metadata is spread in spreadsheets, databases, and applications.

Metadata could be present in text files or multimedia files. To use this data for information management solutions, it has to be correctly defined.

There are no industry-wide accepted standards. Data management solution vendors have narrow focus.

There are no easy and accepted methods of passing metadata.

## Data Warehousing - Data Marting

### Why Do We Need a Data Mart?

Listed below are the reasons to create a data mart –

To partition data in order to impose **access control strategies**.

To speed up the queries by reducing the volume of data to be scanned.

To segment data into different hardware platforms.

To structure data in a form suitable for a user access tool.

**Note** – Do not data mart for any other reason since the operation cost of data marting could be very high. Before data marting, make sure that data marting strategy is appropriate for your particular solution.

### Cost-effective Data Marting

Follow the steps given below to make data marting cost-effective –

Identify the Functional Splits

Identify User Access Tool Requirements

Identify Access Control Issues

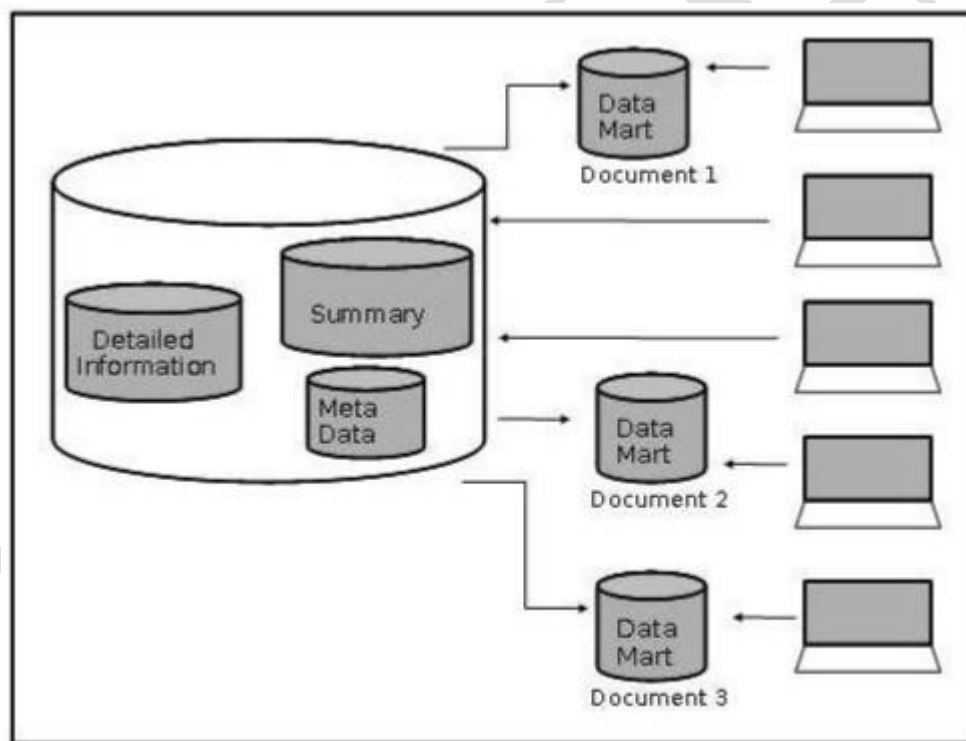
#### Identify the Functional Splits

In this step, we determine if the organization has natural functional splits. We look for departmental splits, and we determine whether the way in which departments use information tend to be in isolation from the rest of the organization. Let's have an example.

Consider a retail organization, where each merchant is accountable for maximizing the sales of a group of products. For this, the following are the valuable information –

- sales transaction on a daily basis
- sales forecast on a weekly basis
- stock position on a daily basis
- stock movements on a daily basis

As the merchant is not interested in the products they are not dealing with, the data marting is a subset of the data dealing which the product group of interest. The following diagram shows data marting for different users.



Given below are the issues to be taken into account while determining the functional split –

- The structure of the department may change.
- The products might switch from one department to other.
- The merchant could query the sales trend of other products to analyze what is happening to the sales.

**Note** – We need to determine the business benefits and technical feasibility of using a data mart.

### Identify User Access Tool Requirements

We need data marts to support **user access tools** that require internal data structures.

The data in such structures are outside the control of data warehouse but need to be populated and updated on a regular basis.

There are some tools that populate directly from the source system but some cannot. Therefore additional requirements outside the scope of the tool are needed to be identified for future.

**Note** – In order to ensure consistency of data across all access tools, the data should not be directly populated from the data warehouse, rather each tool must have its own data mart.

### Identify Access Control Issues

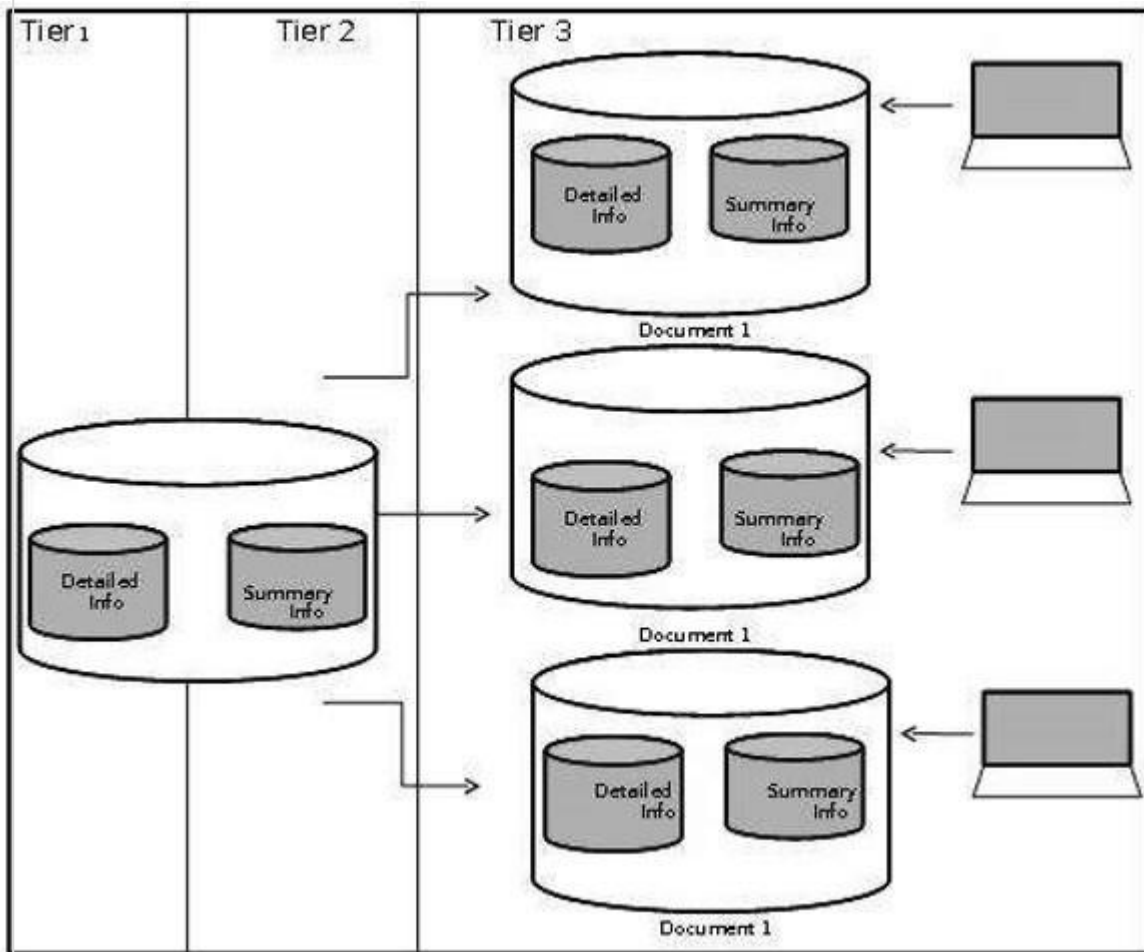
There should to be privacy rules to ensure the data is accessed by authorized users only. For example a data warehouse for retail banking institution ensures that all the accounts belong to the same legal entity. Privacy laws can force you to totally prevent access to information that is not owned by the specific bank.

Data marts allow us to build a complete wall by physically separating data segments within the data warehouse. To avoid possible privacy problems, the detailed data can be removed from the data warehouse. We can create data mart for each legal entity and load it via data warehouse, with detailed account data.

### Designing Data Marts

Data marts should be designed as a smaller version of starflake schema within the data warehouse and should match with the database design of the data warehouse. It helps in maintaining control over database instances.





The summaries are data marted in the same way as they would have been designed within the data warehouse. Summary tables help to utilize all dimension data in the starflake schema.

## Cost of Data Marting

The cost measures for data marting are as follows –

- Hardware and Software Cost

- Network Access

- Time Window Constraints

### Hardware and Software Cost

Although data marts are created on the same hardware, they require some additional hardware and software. To handle user queries, it requires additional processing power and disk storage. If detailed data and the data mart exist within the data warehouse, then we would face additional cost to store and manage replicated data.

**Note** – Data marting is more expensive than aggregations, therefore it should be used as an additional strategy and not as an alternative strategy.

## Network Access

A data mart could be on a different location from the data warehouse, so we should ensure that the LAN or WAN has the capacity to handle the data volumes being transferred within the **data mart load process**.

## Time Window Constraints

The extent to which a data mart loading process will eat into the available time window depends on the complexity of the transformations and the data volumes being shipped. The determination of how many data marts are possible depends on –

- Network capacity.
- Time window available
- Volume of data being transferred
- Mechanisms being used to insert data into a data mart

## Data Warehousing - System Managers

System management is mandatory for the successful implementation of a data warehouse. The most important system managers are –

- System configuration manager
- System scheduling manager
- System event manager
- System database manager
- System backup recovery manager

## System Configuration Manager

The system configuration manager is responsible for the management of the setup and configuration of data warehouse.

The structure of configuration manager varies from one operating system to another.

In Unix structure of configuration, the manager varies from vendor to vendor.

Configuration managers have single user interface.

The interface of configuration manager allows us to control all aspects of the system.

**Note** – The most important configuration tool is the I/O manager.

## System Scheduling Manager

System Scheduling Manager is responsible for the successful implementation of the data warehouse. Its purpose is to schedule ad hoc queries. Every operating system has its own scheduler with some form of batch control mechanism. The list of features a system scheduling manager must have is as follows –

- Work across cluster or MPP boundaries
- Deal with international time differences
- Handle job failure
- Handle multiple queries
- Support job priorities
- Restart or re-queue the failed jobs
- Notify the user or a process when job is completed
- Maintain the job schedules across system outages
- Re-queue jobs to other queues
- Support the stopping and starting of queues
- Log Queued jobs
- Deal with inter-queue processing

**Note** – The above list can be used as evaluation parameters for the evaluation of a good scheduler.

Some important jobs that a scheduler must be able to handle are as follows –

- Daily and ad hoc query scheduling
- Execution of regular report requirements
- Data load
- Data processing
- Index creation
- Backup
- Aggregation creation
- Data transformation

**Note** – If the data warehouse is running on a cluster or MPP architecture, then the system scheduling manager must be capable of running across the architecture.

## System Event Manager

The event manager is a kind of a software. The event manager manages the events that are defined on the data warehouse system. We cannot manage the data warehouse manually because the structure of data warehouse is very complex. Therefore we need a tool that automatically handles all the events without any intervention of the user.

**Note** – The Event manager monitors the events occurrences and deals with them. The event manager also tracks the myriad of things that can go wrong on this complex data warehouse system.

### Events

Events are the actions that are generated by the user or the system itself. It may be noted that the event is a measurable, observable, occurrence of a defined action.

Given below is a list of common events that are required to be tracked.

- Hardware failure
- Running out of space on certain key disks
- A process dying
- A process returning an error
- CPU usage exceeding an 80% threshold
- Internal contention on database serialization points
- Buffer cache hit ratios exceeding or failure below threshold
- A table reaching to maximum of its size
- Excessive memory swapping
- A table failing to extend due to lack of space
- Disk exhibiting I/O bottlenecks
- Usage of temporary or sort area reaching a certain thresholds
- Any other database shared memory usage

The most important thing about events is that they should be capable of executing on their own. Event packages define the procedures for the predefined events. The code associated with each event is known as event handler. This code is executed whenever an event occurs.

## System and Database Manager

System and database manager may be two separate pieces of software, but they do the

same job. The objective of these tools is to automate certain processes and to simplify the execution of others. The criteria for choosing a system and the database manager are as follows –

- increase user's quota.
- assign and de-assign roles to the users
- assign and de-assign the profiles to the users
- perform database space management
- monitor and report on space usage
- tidy up fragmented and unused space
- add and expand the space
- add and remove users
- manage user password
- manage summary or temporary tables
- assign or deassign temporary space to and from the user
- reclaim the space form old or out-of-date temporary tables
- manage error and trace logs
- to browse log and trace files
- redirect error or trace information
- switch on and off error and trace logging
- perform system space management
- monitor and report on space usage
- clean up old and unused file directories
- add or expand space.

## System Backup Recovery Manager

The backup and recovery tool makes it easy for operations and management staff to back-up the data. Note that the system backup manager must be integrated with the schedule manager software being used. The important features that are required for the management of backups are as follows –

- Scheduling
- Backup data tracking
- Database awareness

Backups are taken only to protect against data loss. Following are the important points to remember –

The backup software will keep some form of database of where and when the piece of data was backed up.

The backup recovery manager must have a good front-end to that database.

The backup recovery software should be database aware.

Being aware of the database, the software then can be addressed in database terms, and will not perform backups that would not be viable.

## Data Warehousing - Process Managers

Process managers are responsible for maintaining the flow of data both into and out of the data warehouse. There are three different types of process managers –

Load manager

Warehouse manager

Query manager

## Data Warehouse Load Manager

Load manager performs the operations required to extract and load the data into the database. The size and complexity of a load manager varies between specific solutions from one data warehouse to another.

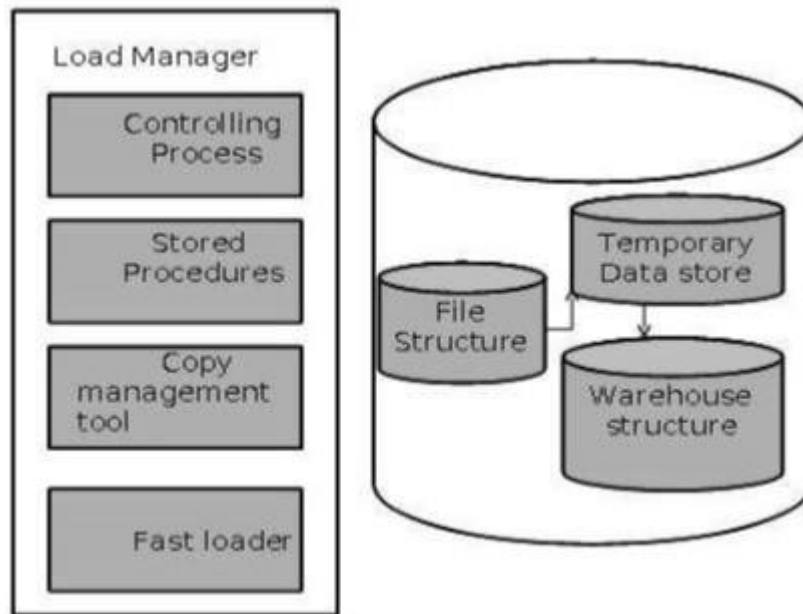
### Load Manager Architecture

The load manager does performs the following functions –

Extract data from the source system.

Fast load the extracted data into temporary data store.

Perform simple transformations into structure similar to the one in the data warehouse.



### Extract Data from Source

The data is extracted from the operational databases or the external information providers. Gateways are the application programs that are used to extract data. It is supported by underlying DBMS and allows the client program to generate SQL to be executed at a server. Open Database Connection (ODBC) and Java Database Connection (JDBC) are examples of gateway.

### Fast Load

In order to minimize the total load window, the data needs to be loaded into the warehouse in the fastest possible time.

Transformations affect the speed of data processing.

It is more effective to load the data into a relational database prior to applying transformations and checks.

Gateway technology is not suitable, since they are inefficient when large data volumes are involved.

### Simple Transformations

While loading, it may be required to perform simple transformations. After completing simple transformations, we can do complex checks. Suppose we are loading the EPOS sales transaction, we need to perform the following checks –

Strip out all the columns that are not required within the warehouse.

Convert all the values to required data types.

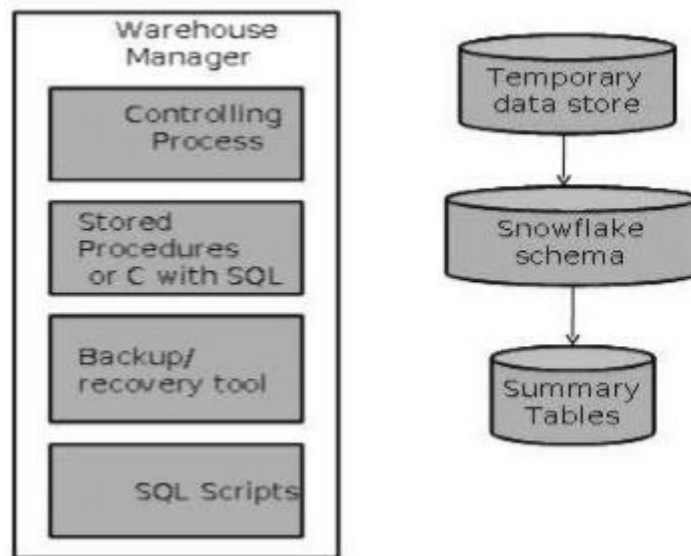
# Warehouse Manager

The warehouse manager is responsible for the warehouse management process. It consists of a third-party system software, C programs, and shell scripts. The size and complexity of a warehouse manager varies between specific solutions.

## Warehouse Manager Architecture

A warehouse manager includes the following –

- The controlling process
- Stored procedures or C with SQL
- Backup/Recovery tool
- SQL scripts



## Functions of Warehouse Manager

A warehouse manager performs the following functions –

- Analyzes the data to perform consistency and referential integrity checks.
- Creates indexes, business views, partition views against the base data.
- Generates new aggregations and updates the existing aggregations.
- Generates normalizations.
- Transforms and merges the source data of the temporary store into the published data warehouse.
- Backs up the data in the data warehouse.
- Archives the data that has reached the end of its captured life.



aggregations are appropriate.

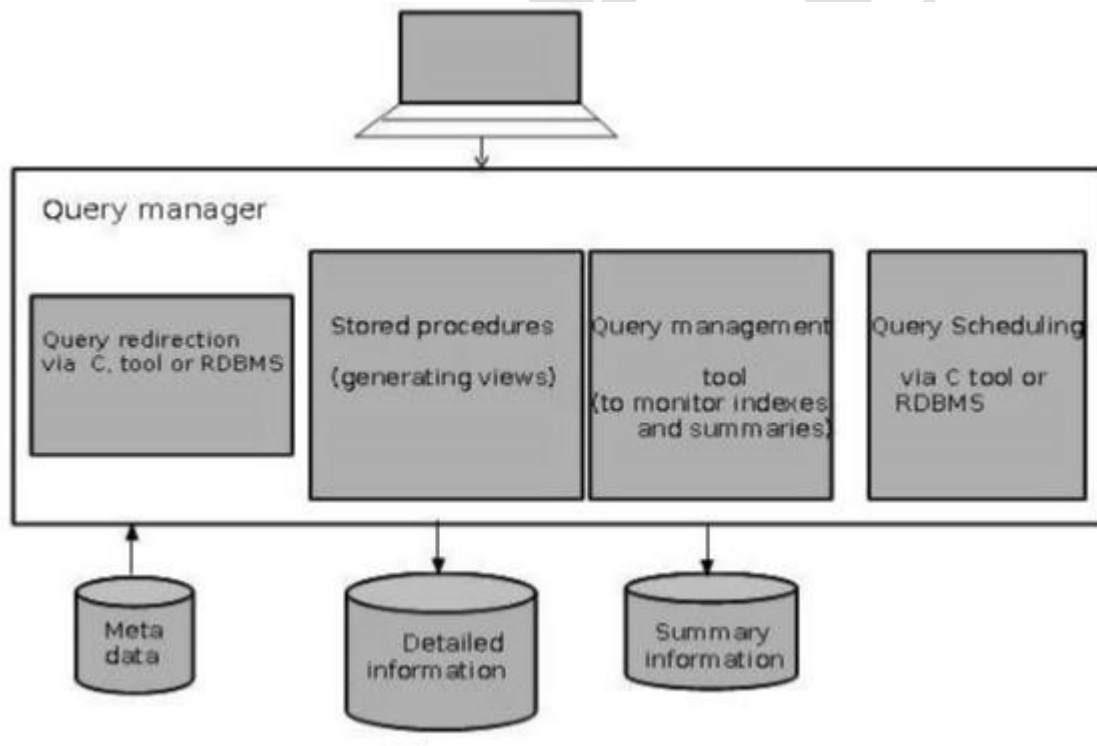
## Query Manager

The query manager is responsible for directing the queries to suitable tables. By directing the queries to appropriate tables, it speeds up the query request and response process. In addition, the query manager is responsible for scheduling the execution of the queries posted by the user.

### Query Manager Architecture

A query manager includes the following components –

- Query redirection via C tool or RDBMS
- Stored procedures
- Query management tool
- Query scheduling via C tool or RDBMS
- Query scheduling via third-party software



### Functions of Query Manager

- It presents the data to the user in a form they understand.
- It schedules the execution of the queries posted by the end-user.
- It stores query profiles to allow the warehouse manager to determine which

indexes and aggregations are appropriate.

## Data Warehousing - Security

The objective of a data warehouse is to make large amounts of data easily accessible to the users, hence allowing the users to extract information about the business as a whole. But we know that there could be some security restrictions applied on the data that can be an obstacle for accessing the information. If the analyst has a restricted view of data, then it is impossible to capture a complete picture of the trends within the business.

The data from each analyst can be summarized and passed on to management where the different summaries can be aggregated. As the aggregations of summaries cannot be the same as that of the aggregation as a whole, it is possible to miss some information trends in the data unless someone is analyzing the data as a whole.

### Security Requirements

Adding security features affect the performance of the data warehouse, therefore it is important to determine the security requirements as early as possible. It is difficult to add security features after the data warehouse has gone live.

During the design phase of the data warehouse, we should keep in mind what data sources may be added later and what would be the impact of adding those data sources. We should consider the following possibilities during the design phase.

Whether the new data sources will require new security and/or audit restrictions to be implemented?

Whether the new users added who have restricted access to data that is already generally available?

This situation arises when the future users and the data sources are not well known. In such a situation, we need to use the knowledge of business and the objective of data warehouse to know likely requirements.

The following activities get affected by security measures –

User access

Data load

Data movement

Query generation

### User Access

We need to first classify the data and then classify the users on the basis of the data they can access. In other words, the users are classified according to the data they can access.

### **Data Classification**

The following two approaches can be used to classify the data –

Data can be classified according to its sensitivity. Highly-sensitive data is classified as highly restricted and less-sensitive data is classified as less restrictive.

Data can also be classified according to the job function. This restriction allows only specific users to view particular data. Here we restrict the users to view only that part of the data in which they are interested and are responsible for.

There are some issues in the second approach. To understand, let's have an example. Suppose you are building the data warehouse for a bank. Consider that the data being stored in the data warehouse is the transaction data for all the accounts. The question here is, who is allowed to see the transaction data. The solution lies in classifying the data according to the function.

### **User classification**

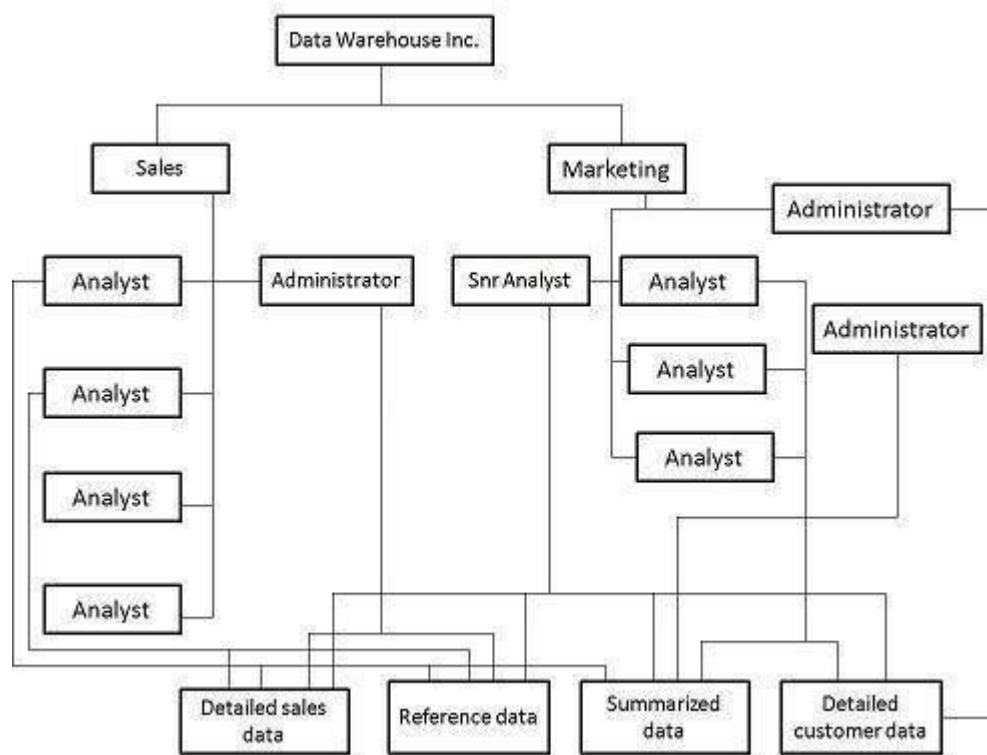
The following approaches can be used to classify the users –

Users can be classified as per the hierarchy of users in an organization, i.e., users can be classified by departments, sections, groups, and so on.

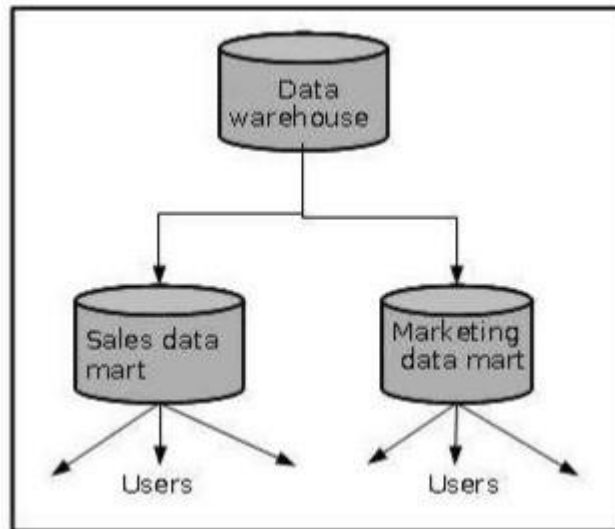
Users can also be classified according to their role, with people grouped across departments based on their role.

### **Classification on basis of Department**

Let's have an example of a data warehouse where the users are from sales and marketing department. We can have security by top-to-down company view, with access centered on the different departments. But there could be some restrictions on users at different levels. This structure is shown in the following diagram.

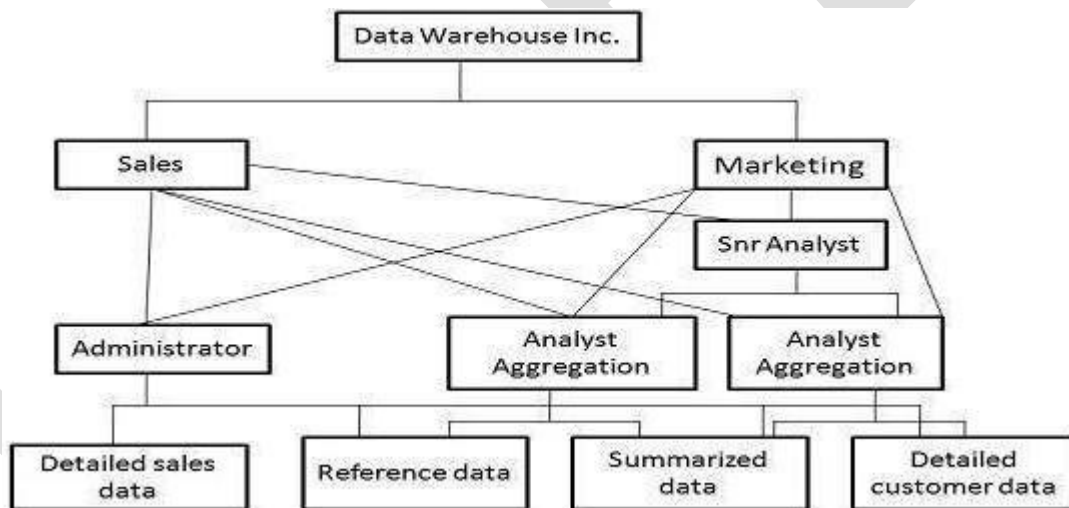


But if each department accesses different data, then we should design the security access for each department separately. This can be achieved by departmental data marts. Since these data marts are separated from the data warehouse, we can enforce separate security restrictions on each data mart. This approach is shown in the following figure.



### Classification Based on Role

If the data is generally available to all the departments, then it is useful to follow the role access hierarchy. In other words, if the data is generally accessed by all the departments, then apply security restrictions as per the role of the user. The role access hierarchy is shown in the following figure.



### Audit Requirements

Auditing is a subset of security, a costly activity. Auditing can cause heavy overheads on the system. To complete an audit in time, we require more hardware and therefore, it is recommended that wherever possible, auditing should be switched off. Audit requirements can be categorized as follows –

- Connections
- Disconnections
- Data access
- Data change

**Note** – For each of the above-mentioned categories, it is necessary to audit success, failure, or both. From the perspective of security reasons, the auditing of failures are very important. Auditing of failure is important because they can highlight unauthorized or fraudulent access.

## Network Requirements

Network security is as important as other securities. We cannot ignore the network security requirement. We need to consider the following issues –

Is it necessary to encrypt data before transferring it to the data warehouse?

Are there restrictions on which network routes the data can take?

These restrictions need to be considered carefully. Following are the points to remember –

The process of encryption and decryption will increase overheads. It would require more processing power and processing time.

The cost of encryption can be high if the system is already a loaded system because the encryption is borne by the source system.

## Data Movement

There exist potential security implications while moving the data. Suppose we need to transfer some restricted data as a flat file to be loaded. When the data is loaded into the data warehouse, the following questions are raised –

Where is the flat file stored?

Who has access to that disk space?

If we talk about the backup of these flat files, the following questions are raised –

Do you backup encrypted or decrypted versions?

Do these backups need to be made to special tapes that are stored separately?

Who has access to these tapes?

Some other forms of data movement like query result sets also need to be considered. The questions raised while creating the temporary table are as follows –

Where is that temporary table to be held?

How do you make such table visible?

We should avoid the accidental flouting of security restrictions. If a user with access to the restricted data can generate accessible temporary tables, data can be visible to non-authorized users. We can overcome this problem by having a separate temporary area for

users with access to restricted data.

## Documentation

The audit and security requirements need to be properly documented. This will be treated as a part of justification. This document can contain all the information gathered from –

- Data classification
- User classification
- Network requirements
- Data movement and storage requirements
- All auditable actions

## Impact of Security on Design

Security affects the application code and the development timescales. Security affects the following area –

- Application development
- Database design
- Testing

## Application Development

Security affects the overall application development and it also affects the design of the important components of the data warehouse such as load manager, warehouse manager, and query manager. The load manager may require checking code to filter record and place them in different locations. More transformation rules may also be required to hide certain data. Also there may be requirements of extra metadata to handle any extra objects.

To create and maintain extra views, the warehouse manager may require extra codes to enforce security. Extra checks may have to be coded into the data warehouse to prevent it from being fooled into moving data into a location where it should not be available. The query manager requires the changes to handle any access restrictions. The query manager will need to be aware of all extra views and aggregations.

## Database design

The database layout is also affected because when security measures are implemented, there is an increase in the number of views and tables. Adding security increases the size of the database and hence increases the complexity of the database design and management. It will also add complexity to the backup management and recovery plan.

## Testing

Testing the data warehouse is a complex and lengthy process. Adding security to the data warehouse also affects the testing time complexity. It affects the testing in the following two ways –

It will increase the time required for integration and system testing.

There is added functionality to be tested which will increase the size of the testing suite.

## Data Warehousing - Backup

A data warehouse is a complex system and it contains a huge volume of data. Therefore it is important to back up all the data so that it becomes available for recovery in future as per requirement. In this chapter, we will discuss the issues in designing the backup strategy.

## Backup Terminologies

Before proceeding further, you should know some of the backup terminologies discussed below.

**Complete backup** – It backs up the entire database at the same time. This backup includes all the database files, control files, and journal files.

**Partial backup** – As the name suggests, it does not create a complete backup of the database. Partial backup is very useful in large databases because they allow a strategy whereby various parts of the database are backed up in a round-robin fashion on a day-to-day basis, so that the whole database is backed up effectively once a week.

**Cold backup** – Cold backup is taken while the database is completely shut down. In multi-instance environment, all the instances should be shut down.

**Hot backup** – Hot backup is taken when the database engine is up and running. The requirements of hot backup varies from RDBMS to RDBMS.

**Online backup** – It is quite similar to hot backup.

## Hardware Backup

It is important to decide which hardware to use for the backup. The speed of processing the backup and restore depends on the hardware being used, how the hardware is connected, bandwidth of the network, backup software, and the speed of server's I/O



system. Here we will discuss some of the hardware choices that are available and their pros and cons. These choices are as follows –

Tape Technology

Disk Backups

## Tape Technology

The tape choice can be categorized as follows –

Tape media

Standalone tape drives

Tape stackers

Tape silos

## Tape Media

There exists several varieties of tape media. Some tape media standards are listed in the table below –

Tape Media	Capacity	I/O rates
DLT	40 GB	3 MB/s
3490e	1.6 GB	3 MB/s
8 mm	14 GB	1 MB/s

Other factors that need to be considered are as follows –

Reliability of the tape medium

Cost of tape medium per unit

Scalability

Cost of upgrades to tape system

Cost of tape medium per unit

Shelf life of tape medium

## Standalone Tape Drives

The tape drives can be connected in the following ways –

Direct to the server

As network available devices

Remotely to other machine

There could be issues in connecting the tape drives to a data warehouse.

Consider the server is a 48node MPP machine. We do not know the node to connect the tape drive and we do not know how to spread them over the server nodes to get the optimal performance with least disruption of the server and least internal I/O latency.

Connecting the tape drive as a network available device requires the network to be up to the job of the huge data transfer rates. Make sure that sufficient bandwidth is available during the time you require it.

Connecting the tape drives remotely also require high bandwidth.

## Tape Stackers

The method of loading multiple tapes into a single tape drive is known as tape stackers. The stacker dismounts the current tape when it has finished with it and loads the next tape, hence only one tape is available at a time to be accessed. The price and the capabilities may vary, but the common ability is that they can perform unattended backups.

## Tape Silos

Tape silos provide large store capacities. Tape silos can store and manage thousands of tapes. They can integrate multiple tape drives. They have the software and hardware to label and store the tapes they store. It is very common for the silo to be connected remotely over a network or a dedicated link. We should ensure that the bandwidth of the connection is up to the job.

## Disk Backups

Methods of disk backups are –

- Disk-to-disk backups

- Mirror breaking

These methods are used in the OLTP system. These methods minimize the database downtime and maximize the availability.

### Disk-to-Disk Backups

Here backup is taken on the disk rather on the tape. Disk-to-disk backups are done for the following reasons –

- Speed of initial backups

- Speed of restore

Backing up the data from disk to disk is much faster than to the tape. However it is the

intermediate step of backup. Later the data is backed up on the tape. The other advantage of disk-to-disk backups is that it gives you an online copy of the latest backup.

### Mirror Breaking

The idea is to have disks mirrored for resilience during the working day. When backup is required, one of the mirror sets can be broken out. This technique is a variant of disk-to-disk backups.

**Note** – The database may need to be shutdown to guarantee consistency of the backup.

### Optical Jukeboxes

Optical jukeboxes allow the data to be stored near line. This technique allows a large number of optical disks to be managed in the same way as a tape stacker or a tape silo. The drawback of this technique is that it has slow write speed than disks. But the optical media provides long-life and reliability that makes them a good choice of medium for archiving.

### Software Backups

There are software tools available that help in the backup process. These software tools come as a package. These tools not only take backup, they can effectively manage and control the backup strategies. There are many software packages available in the market. Some of them are listed in the following table –

Package Name	Vendor
Networker	Legato
ADSM	IBM
Epoch	Epoch Systems
Omniback II	HP
Alexandria	Sequent

### Criteria for Choosing Software Packages

The criteria for choosing the best software package are listed below –

How scalable is the product as tape drives are added?

Does the package have client-server option, or must it run on the database server itself?

Will it work in cluster and MPP environments?

What degree of parallelism is required?

What platforms are supported by the package?

Does the package support easy access to information about tape contents?

Is the package database aware?

What tape drive and tape media are supported by the package?

## Data Warehousing - Tuning

A data warehouse keeps evolving and it is unpredictable what query the user is going to post in the future. Therefore it becomes more difficult to tune a data warehouse system. In this chapter, we will discuss how to tune the different aspects of a data warehouse such as performance, data load, queries, etc.

### Difficulties in Data Warehouse Tuning

Tuning a data warehouse is a difficult procedure due to following reasons –

Data warehouse is dynamic; it never remains constant.

It is very difficult to predict what query the user is going to post in the future.

Business requirements change with time.

Users and their profiles keep changing.

The user can switch from one group to another.

The data load on the warehouse also changes with time.

**Note** – It is very important to have a complete knowledge of data warehouse.

### Performance Assessment

Here is a list of objective measures of performance –

Average query response time

Scan rates

Time used per day query

Memory usage per process

I/O throughput rates

Following are the points to remember.

It is necessary to specify the measures in service level agreement (SLA).

It is of no use trying to tune response time, if they are already better than those

required.

It is essential to have realistic expectations while making performance assessment.

It is also essential that the users have feasible expectations.

To hide the complexity of the system from the user, aggregations and views should be used.

It is also possible that the user can write a query you had not tuned for.

## Data Load Tuning

Data load is a critical part of overnight processing. Nothing else can run until data load is complete. This is the entry point into the system.

**Note** – If there is a delay in transferring the data, or in arrival of data then the entire system is affected badly. Therefore it is very important to tune the data load first.

There are various approaches of tuning data load that are discussed below –

The very common approach is to insert data using the **SQL Layer**. In this approach, normal checks and constraints need to be performed. When the data is inserted into the table, the code will run to check for enough space to insert the data. If sufficient space is not available, then more space may have to be allocated to these tables. These checks take time to perform and are costly to CPU.

The second approach is to bypass all these checks and constraints and place the data directly into the preformatted blocks. These blocks are later written to the database. It is faster than the first approach, but it can work only with whole blocks of data. This can lead to some space wastage.

The third approach is that while loading the data into the table that already contains the table, we can maintain indexes.

The fourth approach says that to load the data in tables that already contain data, **drop the indexes & recreate them** when the data load is complete. The choice between the third and the fourth approach depends on how much data is already loaded and how many indexes need to be rebuilt.

## Integrity Checks

Integrity checking highly affects the performance of the load. Following are the points to remember –

Integrity checks need to be limited because they require heavy processing power.

Integrity checks should be applied on the source system to avoid performance degrade of data load.

## Tuning Queries

We have two kinds of queries in data warehouse –

- Fixed queries

- Ad hoc queries

### Fixed Queries

Fixed queries are well defined. Following are the examples of fixed queries –

- regular reports

- Canned queries

- Common aggregations

Tuning the fixed queries in a data warehouse is same as in a relational database system. The only difference is that the amount of data to be queried may be different. It is good to store the most successful execution plan while testing fixed queries. Storing these executing plan will allow us to spot changing data size and data skew, as it will cause the execution plan to change.

**Note** – We cannot do more on fact table but while dealing with dimension tables or the aggregations, the usual collection of SQL tweaking, storage mechanism, and access methods can be used to tune these queries.

### Ad hoc Queries

To understand ad hoc queries, it is important to know the ad hoc users of the data warehouse. For each user or group of users, you need to know the following –

- The number of users in the group

- Whether they use ad hoc queries at regular intervals of time

- Whether they use ad hoc queries frequently

- Whether they use ad hoc queries occasionally at unknown intervals.

- The maximum size of query they tend to run

- The average size of query they tend to run

- Whether they require drill-down access to the base data

- The elapsed login time per day

- The peak time of daily usage

The number of queries they run per peak hour

### Points to Note

It is important to track the user's profiles and identify the queries that are run on a regular basis.

It is also important that the tuning performed does not affect the performance.

Identify similar and ad hoc queries that are frequently run.

If these queries are identified, then the database will change and new indexes can be added for those queries.

If these queries are identified, then new aggregations can be created specifically for those queries that would result in their efficient execution.

## Data Warehousing - Testing

Testing is very important for data warehouse systems to make them work correctly and efficiently. There are three basic levels of testing performed on a data warehouse –

Unit testing

Integration testing

System testing

### Unit Testing

In unit testing, each component is separately tested.

Each module, i.e., procedure, program, SQL Script, Unix shell is tested.

This test is performed by the developer.

### Integration Testing

In integration testing, the various modules of the application are brought together and then tested against the number of inputs.

It is performed to test whether the various components do well after integration.

### System Testing

In system testing, the whole data warehouse application is tested together.

The purpose of system testing is to check whether the entire system works correctly together or not.

System testing is performed by the testing team.

Since the size of the whole data warehouse is very large, it is usually possible to perform minimal system testing before the test plan can be enacted.

## Test Schedule

First of all, the test schedule is created in the process of developing the test plan. In this schedule, we predict the estimated time required for the testing of the entire data warehouse system.

There are different methodologies available to create a test schedule, but none of them are perfect because the data warehouse is very complex and large. Also the data warehouse system is evolving in nature. One may face the following issues while creating a test schedule –

A simple problem may have a large size of query that can take a day or more to complete, i.e., the query does not complete in a desired time scale.

There may be hardware failures such as losing a disk or human errors such as accidentally deleting a table or overwriting a large table.

**Note** – Due to the above-mentioned difficulties, it is recommended to always double the amount of time you would normally allow for testing.

## Testing Backup Recovery

Testing the backup recovery strategy is extremely important. Here is the list of scenarios for which this testing is needed –

- Media failure
- Loss or damage of table space or data file
- Loss or damage of redo log file
- Loss or damage of control file
- Instance failure
- Loss or damage of archive file
- Loss or damage of table
- Failure during data failure

## Testing Operational Environment

There are a number of aspects that need to be tested. These aspects are listed below.



**Security** – A separate security document is required for security testing. This document contains a list of disallowed operations and devising tests for each.

**Scheduler** – Scheduling software is required to control the daily operations of a data warehouse. It needs to be tested during system testing. The scheduling software requires an interface with the data warehouse, which will need the scheduler to control overnight processing and the management of aggregations.

**Disk Configuration.** – Disk configuration also needs to be tested to identify I/O bottlenecks. The test should be performed with multiple times with different settings.

**Management Tools.** – It is required to test all the management tools during system testing. Here is the list of tools that need to be tested.

- Event manager
- System manager
- Database manager
- Configuration manager
- Backup recovery manager

## Testing the Database

The database is tested in the following three ways –

**Testing the database manager and monitoring tools** – To test the database manager and the monitoring tools, they should be used in the creation, running, and management of test database.

**Testing database features** – Here is the list of features that we have to test –

- Querying in parallel
- Create index in parallel
- Data load in parallel

**Testing database performance** – Query execution plays a very important role in data warehouse performance measures. There are sets of fixed queries that need to be run regularly and they should be tested. To test ad hoc queries, one should go through the user requirement document and understand the business completely. Take time to test the most awkward queries that the business is likely to ask against different index and aggregation strategies.

## Testing the Application

All the managers should be integrated correctly and work in order to ensure that the end-to-end load, index, aggregate and queries work as per the expectations.

Each function of each manager should work correctly

It is also necessary to test the application over a period of time.

Week end and month-end tasks should also be tested.

## Logistic of the Test

The aim of system test is to test all of the following areas –

Scheduling software

Day-to-day operational procedures

Backup recovery strategy

Management and scheduling tools

Overnight processing

Query performance

**Note** – The most important point is to test the scalability. Failure to do so will leave us a system design that does not work when the system grows.

## Data Warehousing - Future Aspects

Following are the future aspects of data warehousing.

As we have seen that the size of the open database has grown approximately double its magnitude in the last few years, it shows the significant value that it contains.

As the size of the databases grow, the estimates of what constitutes a very large database continues to grow.

The hardware and software that are available today do not allow to keep a large amount of data online. For example, a Telco call record requires 10TB of data to be kept online, which is just a size of one month's record. If it requires to keep records of sales, marketing customer, employees, etc., then the size will be more than 100 TB.

The record contains textual information and some multimedia data. Multimedia data cannot be easily manipulated as text data. Searching the multimedia data is

not an easy task, whereas textual information can be retrieved by the relational software available today.

Apart from size planning, it is complex to build and run data warehouse systems that are ever increasing in size. As the number of users increases, the size of the data warehouse also increases. These users will also require to access the system.

With the growth of the Internet, there is a requirement of users to access data online.

Hence the future shape of data warehouse will be very different from what is being created today.

**CLASS**  
**COURSE CODE**

**KARPAGAM ACADEMY OF HIGHER EDUCATION**  
**: I M.COM CA**  
**: 18CCP202**

**COURSE NAME: COST ACCOUNTING**  
**BATCH : 2018-2020**

KAHE

1.	In its simplest perception, _____ is no more than a collection of the key prices of information used to manage and direct the business for the most profitable outcome.	data warehouse	data mining	big data	business intelligence	data warehouse
2.	_____ should be subject oriented, be consistent across sources, and so on.	data mining	data warehouse	big data	business intelligence	data warehouse
3.	_____ is more than just data, it is also the process involved in getting that data from sources to table, and in getting the data from table to analysts.	data mining	big data	data warehouse	business intelligence	data warehouse
4.	_____ is the data and the process managers that make information available, enabling propel to make informed decisions.	data mining	big data	business intelligence	data warehouse	data warehouse
5.	Marketing database, customer loyalty scheme, customer profiling and segmentation will come under	business solution	system solution	marketing solutions	stock solution	marketing solutions
6.	Sales analysis, shrinkage analysis, promotions analysis, space planning will come under	sale	retail	trade	marketing	retail
7.	_____ are built to support large data volumes cost-effectively.	data warehouse	data mining	big data	business intelligence	data warehouse
8.	The architecture of a _____ is defined within the technical blueprint stage of the process	data warehouse	data mining	big data	business intelligence	data warehouse
9.	_____ Stage should have identified the initial used requirements and have developed an understanding of the longer term business requirements.	technical blue print	business requirements'	marketing requirements	sale requirements	business requirements'
10	The processes required to populate the warehouse focus on extracting the data, _____ it up and making it available for analysis.	loading	extracting	cleaning	transformation	cleaning

11	A common misconception is that_____ are read-only system	data mining	big data	business intelligence	data warehouse	data warehouse
12	_____ takes data from source systems and makes it available to the data warehouse.	data loading	extracting	data extraction	transformation	data extraction
13	_____ is the system component that performs all the operations necessary to support the extract and load process.	warehouse manager	load manager	query manager	system manager	load manager
14	_____ should be loaded into the warehouse in the fastest possible time, in order to minimize the total load window.	data	information	record	field	data
15	_____ is the system component that performs all the operations necessary to support the warehouse management process.	warehouse manager	data manager	load manager	system manager	warehouse manager
16	_____ is the system component that performs all the operations necessary to support the query management process.	warehouse manager	query manager	load manager	system manager	query manager
17	SKU stands for_____	store keeping unit	material keeping unit	Stock-Keeping Unit	stock keeping under	Stock-Keeping Unit
18	SQL Stands for_____	Structure query language	Star Query Language	Safe Query Language	Structured Query Language	Structured Query Language
19	All design decisions should be based on the system architecture defined in the _____	business requirement	partitioning	Technical blue print	aggregation	Technical blue print
20	_____ Schemas are physical database structures that store the factual data in the "center".	snow	Star	snowflake	star flake	Star
21	EPOS Stands for_____	Electronic Point of Sale	point of sale	mobile banking	online ale	Electronic Point of Sale
22	_____ is the central table in the star schema of a data warehouse.	Fact table	partition table	primary table	aggregation table	Fact table

23	_____ contains elements of both concepts, plus a number of additional ideas.	node	star flake schema	schema	snowflake schema	star flake schema
24	Integral to the _____ are the techniques that direct a query to the most appropriate source.	snow schema	star schema	star flake schema	snowflake schema	star flake schema
25	_____ is the database process where very large tables are divided into multiple smaller parts	Aggregation	data mart	Meta data	Partitioning	Partitioning
26	Data is split vertical called as _____ Partitioning	horizontal	normal	vertical	up to down	vertical
27	SMP Stands for _____	system manager Processing	Systematic Multi Processing	system manager process	system more processor	Systematic Multi Processing
28	MPP Stands for _____	Massively Parallel Processing	Mass Parallel Processing	Massive Parallel Processor	Mass Parallel Processing	Massively Parallel Processing
29	SLA Stands for _____	Service Level Agreement	terms	condition	agreement	Service Level Agreement
30	_____ technique spreads the processing load by horizontally portioning the fact table into small segments.	vertical partitioning	Horizontal hardware partitioning	partitioning	hardware partitioning	Horizontal hardware partitioning
31	_____ is performed in order to speed up common queries.	data aggregation	portioning	Aggregation	data partitioning	Aggregation
32	_____ is an essential component of decision support data warehouse.	Aggregation	partitioning	data partitioning	Data aggregation	Data aggregation
33	_____ Strategies rely on the fact that most common queries will analyze either a subset.	partitioning	subset	data mart	Aggregation	Aggregation
34	The primary purpose of using _____ tables is to cut down the time it takes to execute queries.	Fact table	partitioning table	Summary table	primary table	Summary table
35	_____ are probably inappropriate within a summary table.	data subset	Data offsets	data mart	data disk	Data offsets

36	_____ are designed using a similar process to that followed to design fact tables.	Summary table	subset	Meta data	fact table	Summary table
37	The _____ is a subset of the data warehouse and is usually oriented to a specific business line or team.	Data mart	subset	Meta data	data partitioning	Data mart
38	_____ That are not created to satisfy the requirements of user access tools should be designed to match the database design of the data warehouse.	meta data	Data mart	subset	data partitioning	Data mart
39	_____ allow us to build complete Chinese walls by physically separating data segments within the data warehouse.	meta data	Data mart	subset	data partitioning	Data mart
40	_____ is a business term describing an information barrier within an organization that was erected to prevent exchanges or communication that could lead to conflicts of interest.	Falling model	ups and down	Chinese wall	data offset	Chinese wall
41	LAN Stands for _____	wide area network	small area network	main are network	Local Area Network	Local Area Network
42	WAN Stands for _____	local area network	small area network	Wide Area Network	main area network	Wide Area Network
43	_____ is data that describes other data.	data mart	Meta data	sub set	data about data	Meta data
44	_____ is used for data transformation and load, data management and query generation.	data mart	Meta data	sub set	base data	Meta data
45	_____ may be used during data transformation and load to describe the source data and any changes the need to be made.	data mart	Meta data	sub set	base data	Meta data
46	_____ is required to avoid any confusion occurring between two fields of the same name from different sources.	identifier	primary identifier	unique identifier	aggregate identifier	unique identifier



47.	The term _____ is used here to distinguish between the original data load destination and any copies.	Meta data	data mart	data drive	base data	base data
48.	_____ is required to describe the data as it resides in the data warehouse.	Data mart	aggregation	partitioning	Meta data	Meta data
49.	_____ is also required by the query manager to enable it to generate queries.	data mart	aggregation	Meta data	Partitioning	Meta data
50.	_____ key is the column or columns on which the table is partitioned and reference identifier is the unique identifier from the column metadata.	Fact table	Partition key	identifying key	primary key	Partition key
51.	_____ is responsible for the setup and configuration of the hardware.	Configuration manager	process manager	system manager	query manager	Configuration manager
52.	Behind ever successful data warehouse there is a good solid_____	Scheduler	identifier	partitioner	aggregator	Scheduler
53.	_____ is a piece of software that deals with defined events on the data warehouse system.	query manager	Event manager	process manager	system manager	Event manager
54.	A _____ is a measurable, observable occurrence of a defined action.	node	disk	Event	parallel	Event
55.	_____ will probably be two separate prices of software, but we have lumped them together here because they do essentially the same job.	system manager	query manager	load manager	the system and database manager	the system and database manager
56.	The _____ are pieces of software responsible for flow, maintenance and upkeep of the data.	system manager	query manager	load manager	data warehouse process managers	data warehouse process managers
57.	The _____ is responsible for any data transformation required and for the loading of data into the database.	warehouse manager	query manager	load manager	the system and database manager	load manager
58.	The _____ is manager is responsible for maintaining the data while it is in the data warehouse.	load manager	warehouse manager	query manager	the system and database manager	warehouse manager
59.	_____ is the process of moving data that is no longer actively	Data archiving	data aggregation	data partitioning	identifying	Data archiving

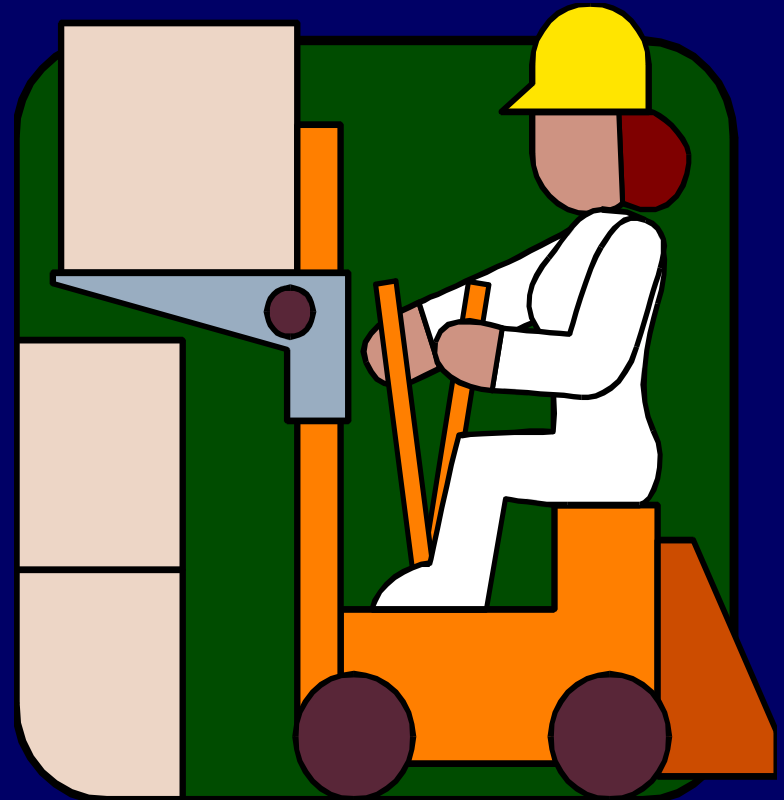
	used to a separate storage device for long-term retention					
60	_____ is the software interface between the users and the data.	query manager	warehouse manager	load manager	the system and database manager	query manager

# DATA WAREHOUSING AND DATA MINING



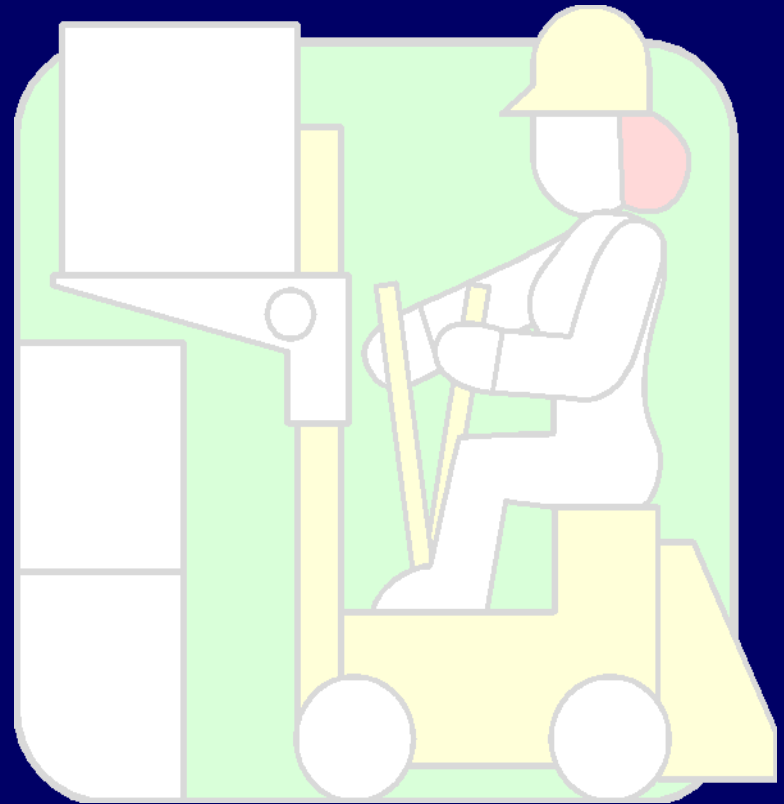
# Course Overview

- ⌘ The course:  
what and how
- ⌘ 0. Introduction
- ⌘ I. Data Warehousing
- ⌘ II. Decision Support  
and OLAP
- ⌘ III. Data Mining
- ⌘ IV. Looking Ahead
- ⌘ Demos and Labs



# 0. Introduction

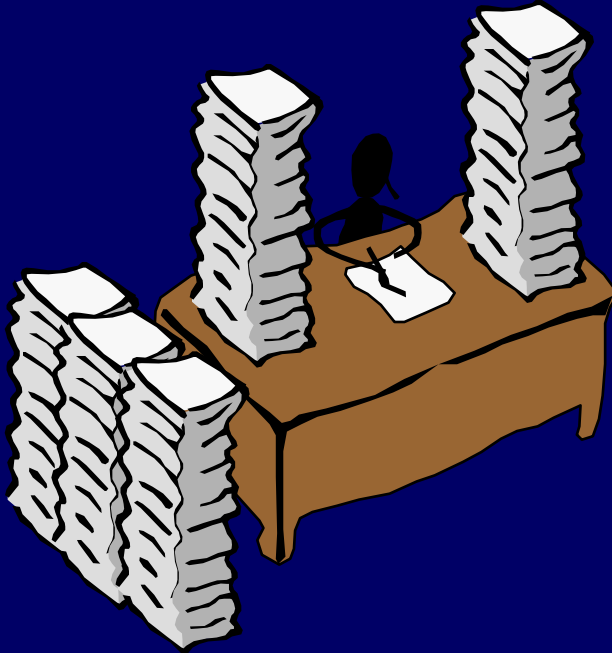
- ⌘ Data Warehousing, OLAP and data mining: what and why (now)?
- ⌘ Relation to OLTP
- ⌘ A case study
- ⌘ demos, labs



# A producer wants to know....



# Data, Data everywhere yet ...

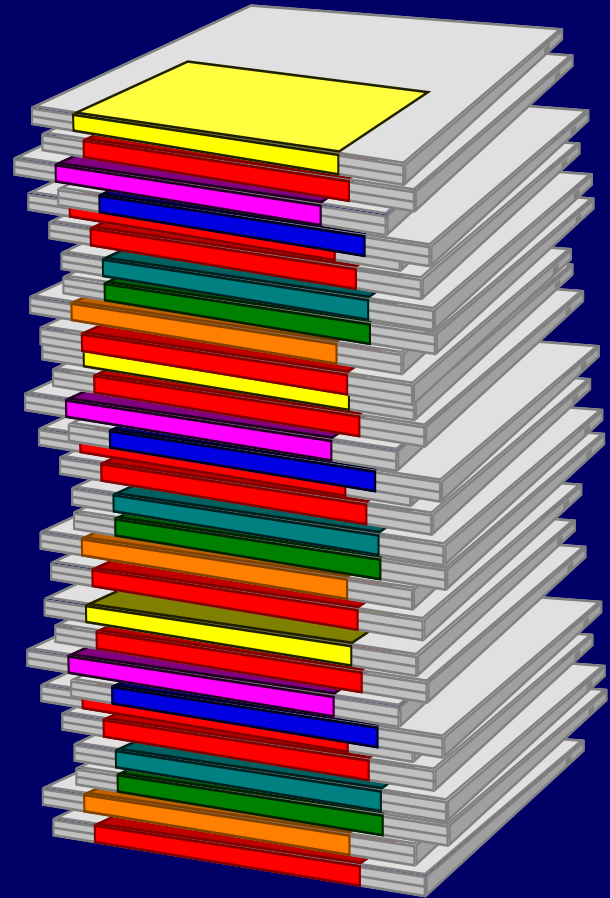


- ⌘ I can't find the data I need
  - ⌘ data is scattered over the network
  - ⌘ many versions, subtle differences
- ⌘ I can't get the data I need
  - ⌘ need an expert to get the data
- ⌘ I can't understand the data I found
  - ⌘ available data poorly documented
- ⌘ I can't use the data I found
  - ⌘ 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.

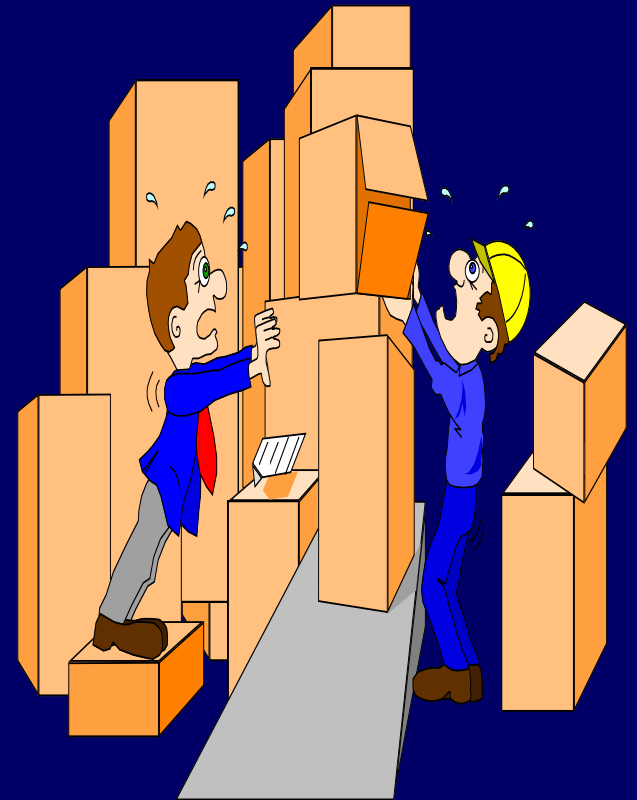
[Barry Devlin]



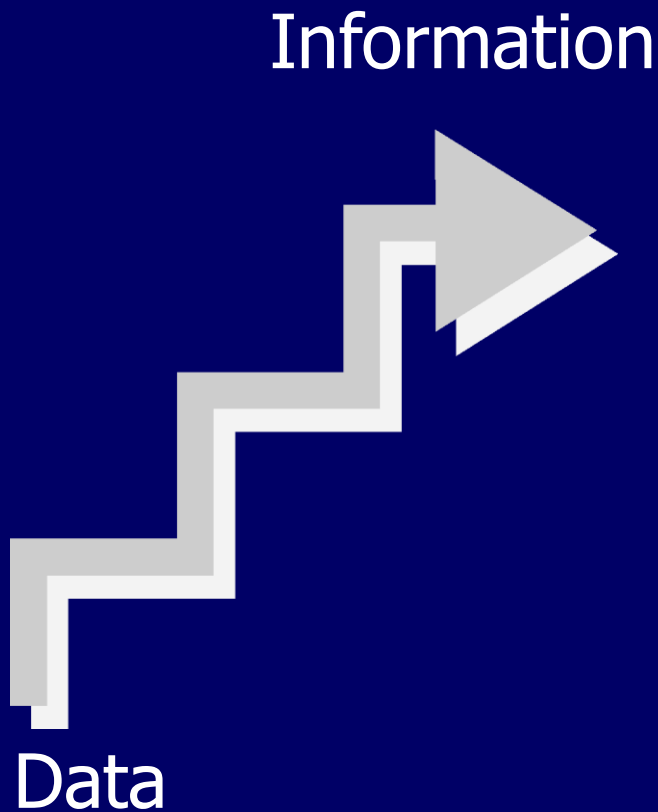


# What are the users saying...

- ⌘ Data should be integrated across the enterprise
- ⌘ Summary data has a real value to the organization
- ⌘ Historical data holds the key to understanding data over time
- ⌘ What-if capabilities are required



# What is Data Warehousing?



A **process** of transforming **data** into **information** and making it available to users in a timely enough manner to make a difference

[Forrester Research, April 1996]

# Evolution

## ⌘ 60's: Batch reports

- ⌘ hard to find and analyze information
- ⌘ inflexible and expensive, reprogram every new request

## ⌘ 70's: Terminal-based DSS and EIS (executive information systems)

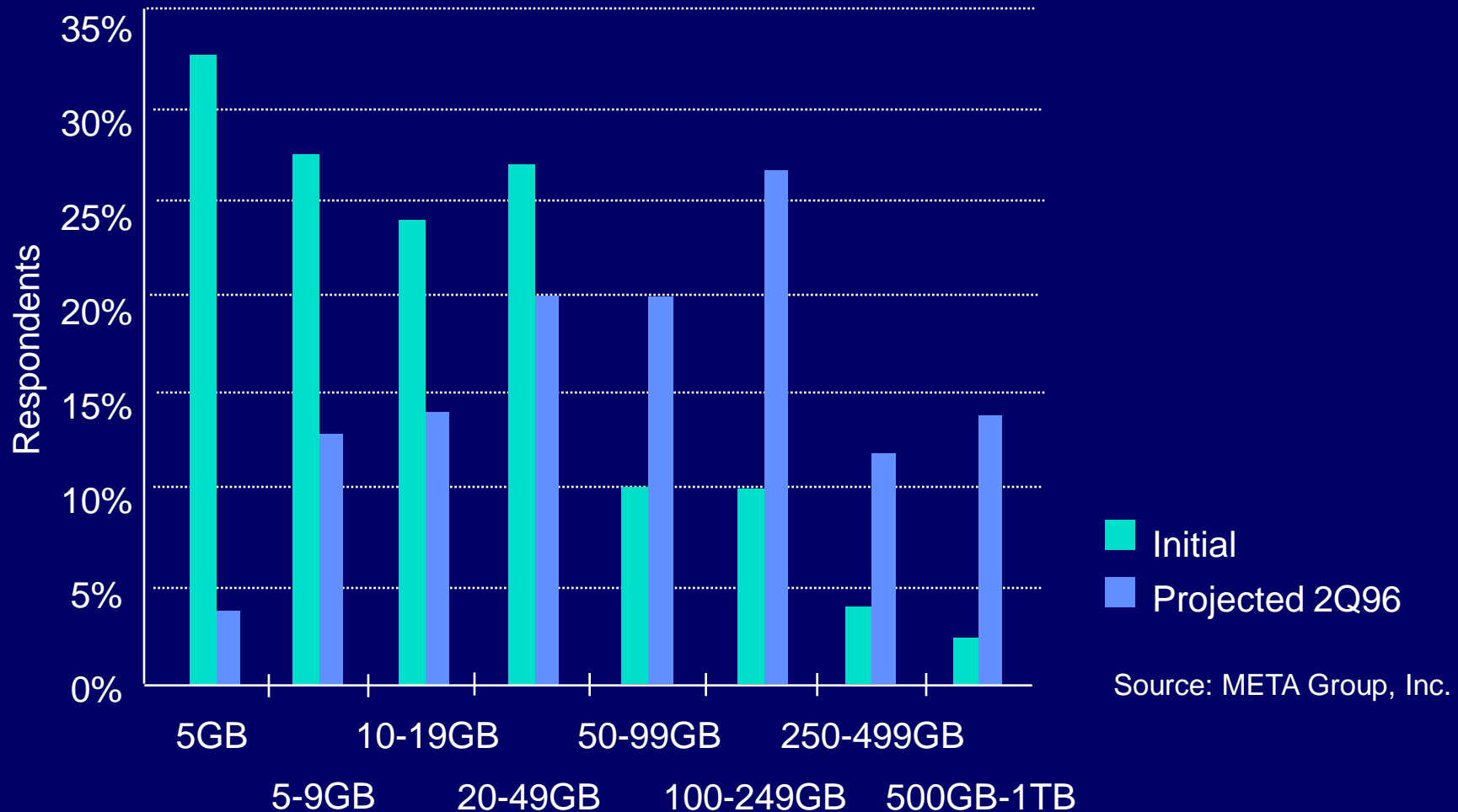
- ⌘ still inflexible, not integrated with desktop tools

## ⌘ 80's: Desktop data access and analysis tools

- ⌘ query tools, spreadsheets, GUIs
- ⌘ easier to use, but only access operational databases

## ⌘ 90's: Data warehousing with integrated OLAP engines and tools

# Warehouses are Very Large Databases



# Very Large Data Bases

⌘ Terabytes --  $10^{12}$  bytes: Walmart -- 24 Terabytes

⌘ Petabytes --  $10^{15}$  bytes: Geographic Information Systems

⌘ Exabytes --  $10^{18}$  bytes: National Medical Records

⌘ Zettabytes --  $10^{21}$  bytes:

Weather images

⌘ Zottabytes --  $10^{24}$  bytes:

Intelligence Agency Videos

# Data Warehousing -- It is a process



- ⌘ Technique for assembling and managing data from various sources for the purpose of answering business questions. Thus making decisions that were not previous possible
- ⌘ A decision support database maintained separately from the organization's operational database

# Data Warehouse

⌘ A data warehouse is a

- ☑ subject-oriented

- ☑ integrated

- ☑ time-varying

- ☑ non-volatile

collection of data that is used primarily in organizational decision making.

-- Bill Inmon, Building the Data Warehouse 1996

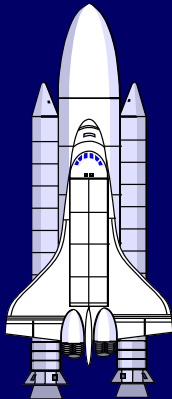
# Explorers, Farmers and Tourists



Tourists: Browse information harvested by farmers



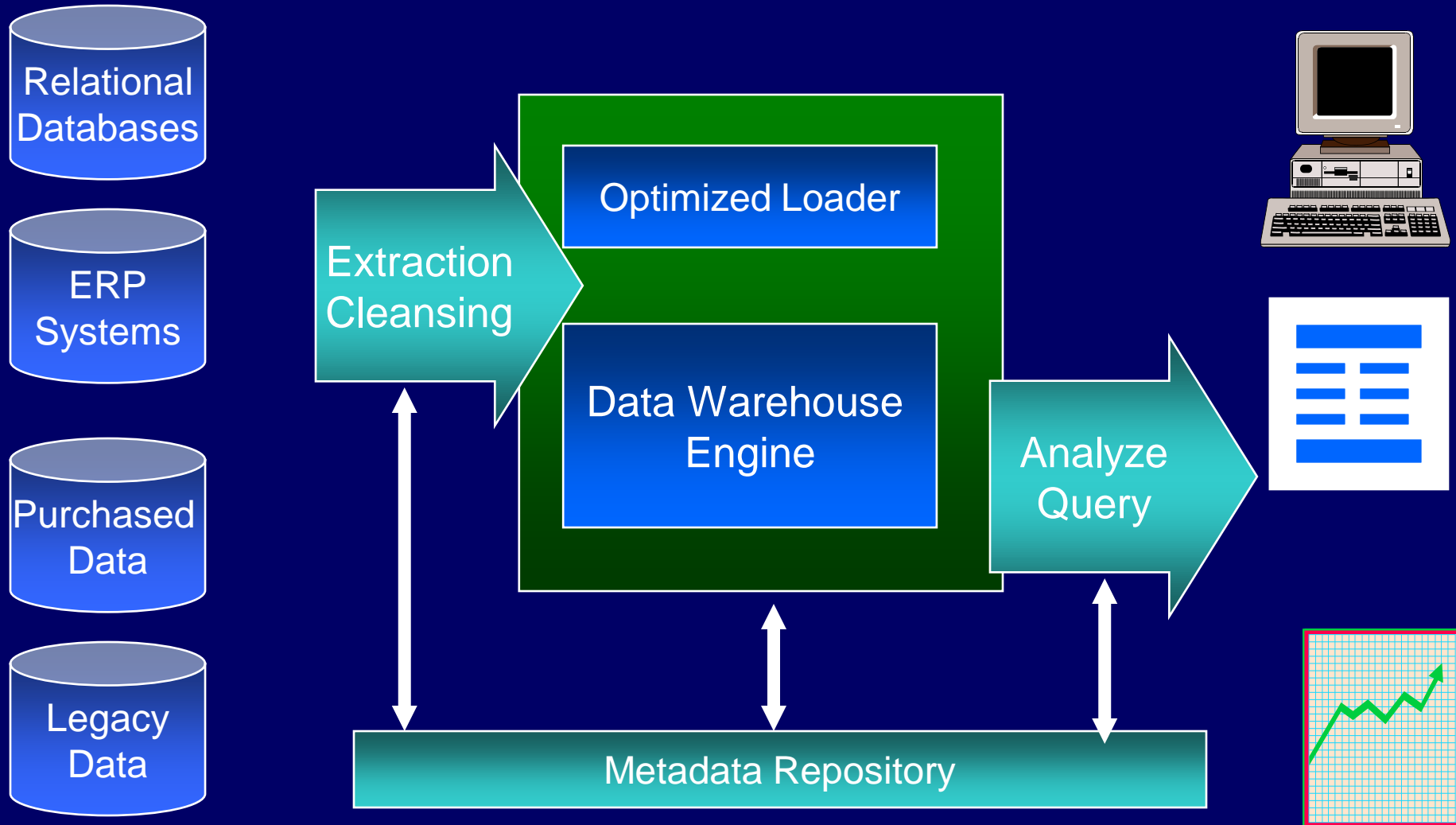
Farmers: Harvest information from known access paths



Explorers: Seek out the unknown and previously unsuspected rewards hiding in the detailed data



# Data Warehouse Architecture



# Data Warehouse for Decision Support & OLAP

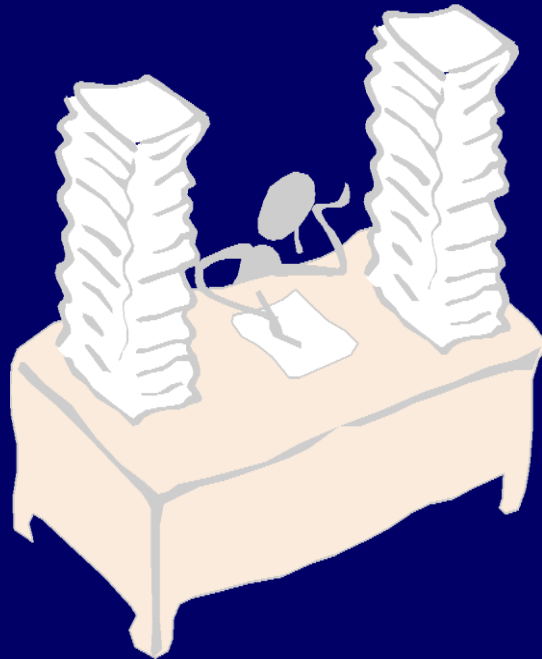
⌘ Putting Information technology to help the knowledge worker make faster and better decisions

- ☒ Which of my customers are most likely to go to the competition?
- ☒ What product promotions have the biggest impact on revenue?
- ☒ How did the share price of software companies correlate with profits over last 10 years?

# Decision Support

- ⌘ Used to manage and control business
- ⌘ Data is historical or point-in-time
- ⌘ Optimized for inquiry rather than update
- ⌘ Use of the system is loosely defined and can be ad-hoc
- ⌘ Used by managers and end-users to understand the business and make judgements

# Data Mining works with Warehouse Data



⌘ Data Warehousing provides the Enterprise with a memory

⌘ Data Mining provides the Enterprise with intelligence



# We want to know ...

- ⌘ Given a database of 100,000 names, which persons are the least likely to default on their credit cards?
- ⌘ Which types of transactions are likely to be fraudulent given the demographics and transactional history of a particular customer?
- ⌘ If I raise the price of my product by Rs. 2, what is the effect on my ROI?
- ⌘ If I offer only 2,500 airline miles as an incentive to purchase rather than 5,000, how many lost responses will result?
- ⌘ If I emphasize ease-of-use of the product as opposed to its technical capabilities, what will be the net effect on my revenues?
- ⌘ Which of my customers are likely to be the most loyal?

**Data Mining helps extract such information**

# 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
- ⌘ Warranty Claims Routing
- ⌘ Holding on to Good Customers
- ⌘ Weeding out Bad Customers

# What makes data mining possible?

⌘ Advances in the following areas are making data mining deployable:

- ☑ data warehousing
- ☑ better and more data (i.e., operational, behavioral, and demographic)
- ☑ the emergence of easily deployed data mining tools and
- ☑ the advent of new data mining techniques.

- -- Gartner Group



# Why Separate Data Warehouse?

## ⌘ Performance

- ⌘ Op dbs designed & tuned for known txs & workloads.
- ⌘ Complex OLAP queries would degrade perf. for op txs.
- ⌘ Special data organization, access & implementation methods needed for multidimensional views & queries.

## ⌘ Function

- ⌘ Missing data: Decision support requires historical data, which op dbs do not typically maintain.
- ⌘ Data consolidation: Decision support requires consolidation (aggregation, summarization) of data from many heterogeneous sources: op dbs, external sources.
- ⌘ Data quality: Different sources typically use inconsistent data representations, codes, and formats which have to be reconciled.

# What are Operational Systems?

- ⌘ They are OLTP systems
- ⌘ Run mission critical applications
- ⌘ Need to work with stringent performance requirements for routine tasks
- ⌘ Used to run a business!



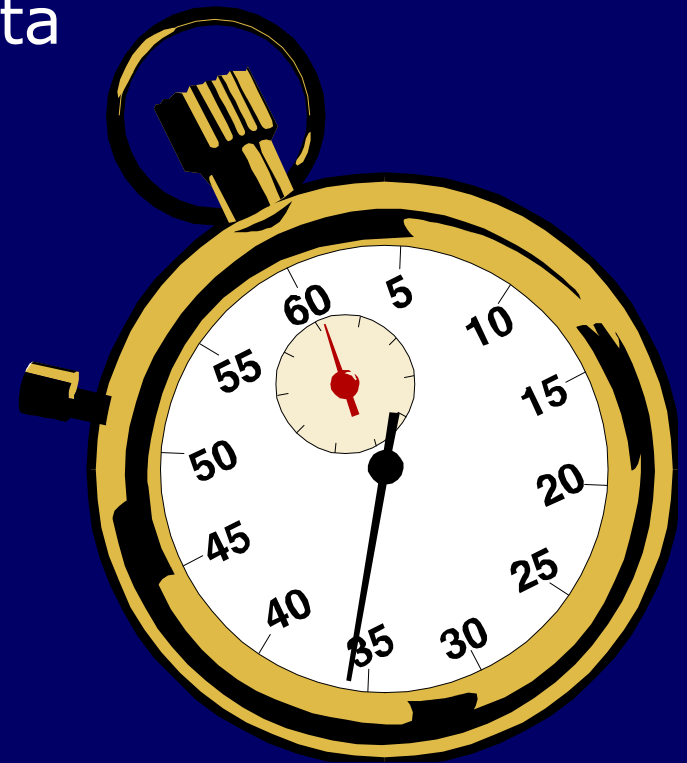
# RDBMS used for OLTP

⌘ Database Systems have been used traditionally for OLTP

- ☑ clerical data processing tasks
- ☑ detailed, up to date data
- ☑ structured repetitive tasks
- ☑ read/update a few records
- ☑ isolation, recovery and integrity are critical

# Operational Systems

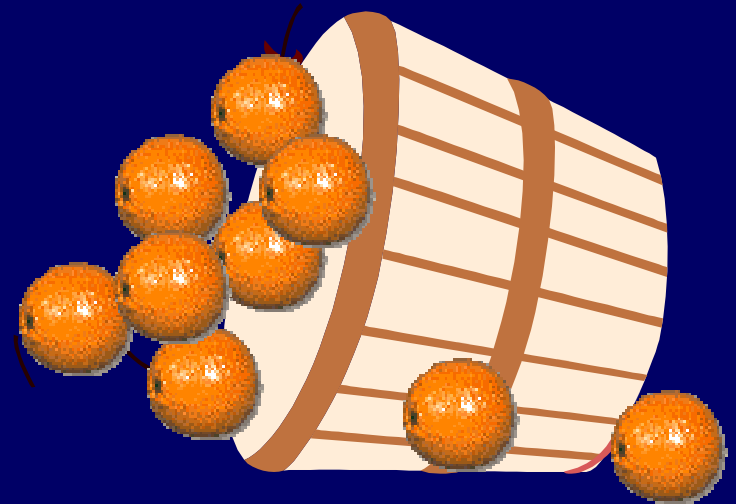
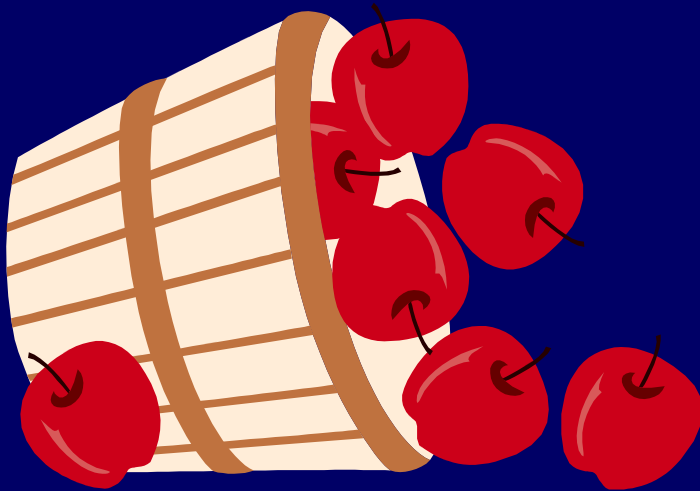
- ⌘ Run the business in real time
- ⌘ Based on up-to-the-second data
- ⌘ Optimized to handle large numbers of simple read/write transactions
- ⌘ Optimized for fast response to predefined transactions
- ⌘ Used by people who deal with customers, products -- clerks, salespeople etc.
- ⌘ They are increasingly used by customers



# Examples of Operational Data

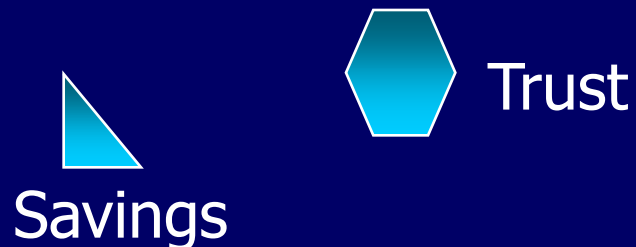
Data	Industry	Usage	Technology	Volumes
Customer File	All	Track Customer Details	Legacy application, flat files, main frames	Small-medium
Account Balance	Finance	Control account activities	Legacy applications, hierarchical databases, mainframe	Large
Point-of-Sale data	Retail	Generate bills, manage stock	ERP, Client/Server, relational databases	Very Large
Call Record	Telecommunications	Billing	Legacy application, hierarchical database, mainframe	Very Large
Production Record	Manufacturing	Control Production	ERP, relational databases, AS/400	Medium

So, what's different?



# Application-Orientation vs. Subject-Orientation

Application-Orientation



Subject-Orientation



# OLTP vs. Data Warehouse

- ⌘ OLTP systems are tuned for known transactions and workloads while workload is not known a priori in a data warehouse
- ⌘ Special data organization, access methods and implementation methods are needed to support data warehouse queries (typically multidimensional queries)
  - ☑ e.g., *average amount spent on phone calls between 9AM-5PM in Pune during the month of December*



# OLTP vs Data Warehouse

## ⌘ OLTP

- ☑ Application Oriented
- ☑ Used to run business
- ☑ Detailed data
- ☑ Current up to date
- ☑ Isolated Data
- ☑ Repetitive access
- ☑ Clerical User

## ⌘ Warehouse (DSS)

- ☑ Subject Oriented
- ☑ Used to analyze business
- ☑ Summarized and refined
- ☑ Snapshot data
- ☑ Integrated Data
- ☑ Ad-hoc access
- ☑ Knowledge User (Manager)

# OLTP vs Data Warehouse

## ⌘ OLTP

- ☑ Performance Sensitive
- ☑ Few Records accessed at a time (tens)
- ☑ Read/Update Access
- ☑ No data redundancy
- ☑ Database Size      100MB  
                             -100 GB

## ⌘ Data Warehouse

- ☑ Performance relaxed
- ☑ Large volumes accessed at a time (millions)
- ☑ Mostly Read (Batch Update)
- ☑ Redundancy present
- ☑ Database Size  
                             100 GB - few terabytes

# OLTP vs Data Warehouse

## ⌘ OLTP

- ☒ Transaction throughput is the performance metric
- ☒ Thousands of users
- ☒ Managed in entirety

## ⌘ Data Warehouse

- ☒ Query throughput is the performance metric
- ☒ Hundreds of users
- ☒ Managed by subsets

# To summarize ...

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



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

# Why Now?

- ⌘ Data is being produced
- ⌘ ERP provides clean data
- ⌘ The computing power is available
- ⌘ The computing power is affordable
- ⌘ The competitive pressures are strong
- ⌘ Commercial products are available

# Myths surrounding OLAP Servers and Data Marts

- ⌘ Data marts and OLAP servers are departmental solutions supporting a handful of users
- ⌘ Million dollar massively parallel hardware is needed to deliver fast time for complex queries
- ⌘ OLAP servers require massive and unwieldy indices
- ⌘ Complex OLAP queries clog the network with data
- ⌘ Data warehouses must be at least 100 GB to be effective

– Source -- Arbor Software Home Page

# Wal\*Mart Case Study

- ⌘ Founded by Sam Walton
- ⌘ One the largest Super Market Chains in the US
- ⌘ Wal\*Mart: 2000+ Retail Stores
- ⌘ SAM's Clubs 100+Wholesalers Stores

⌘ This case study is from Felipe Carino's (NCR Teradata) presentation made at Stanford Database Seminar

# Old Retail Paradigm

## ⌘ Wal\*Mart

- ☑ Inventory Management
- ☑ Merchandise Accounts Payable
- ☑ Purchasing
- ☑ Supplier Promotions: National, Region, Store Level

## ⌘ Suppliers

- ☑ Accept Orders
- ☑ Promote Products
- ☑ Provide special Incentives
- ☑ Monitor and Track The Incentives
- ☑ Bill and Collect Receivables
- ☑ Estimate Retailer Demands



# New (Just-In-Time) Retail Paradigm

- ⌘ No more deals

- ⌘ Shelf-Pass Through (POS Application)

  - ☒ One Unit Price

    - ☒ Suppliers paid once a week on ACTUAL items sold

  - ☒ Wal\*Mart Manager

    - ☒ Daily Inventory Restock

    - ☒ Suppliers (sometimes SameDay) ship to Wal\*Mart

- ⌘ Warehouse-Pass Through

  - ☒ Stock some Large Items

    - ☒ Delivery may come from supplier

  - ☒ Distribution Center

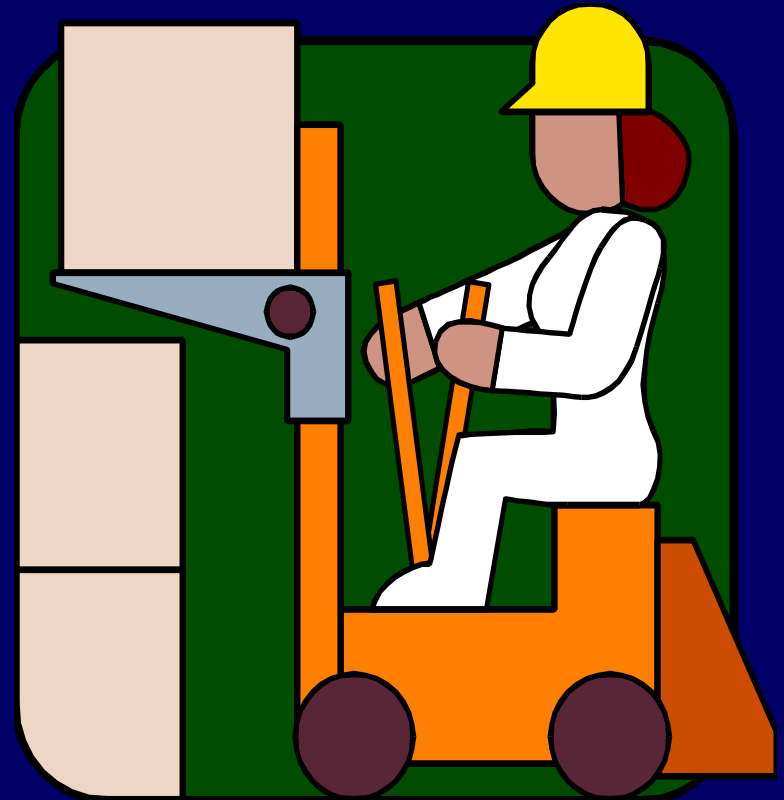
    - ☒ Supplier's merchandise unloaded directly onto Wal\*Mart Trucks

# Wal\*Mart System

- ⌘ NCR 5100M 96 Nodes; *24 TB Raw Disk; 700 - 1000 Pentium CPUs*
- ⌘ Number of Rows: *> 5 Billions*
- ⌘ Historical Data: *65 weeks (5 Quarters)*
- ⌘ New Daily Volume: *Current Apps: 75 Million  
New Apps: 100 Million +*
- ⌘ Number of Users: *Thousands*
- ⌘ Number of Queries: *60,000 per week*

# Course Overview

- ⌘ 0. Introduction
- ⌘ I. **Data Warehousing**
- ⌘ II. Decision Support and OLAP
- ⌘ III. Data Mining
- ⌘ IV. Looking Ahead
- ⌘ Demos and Labs

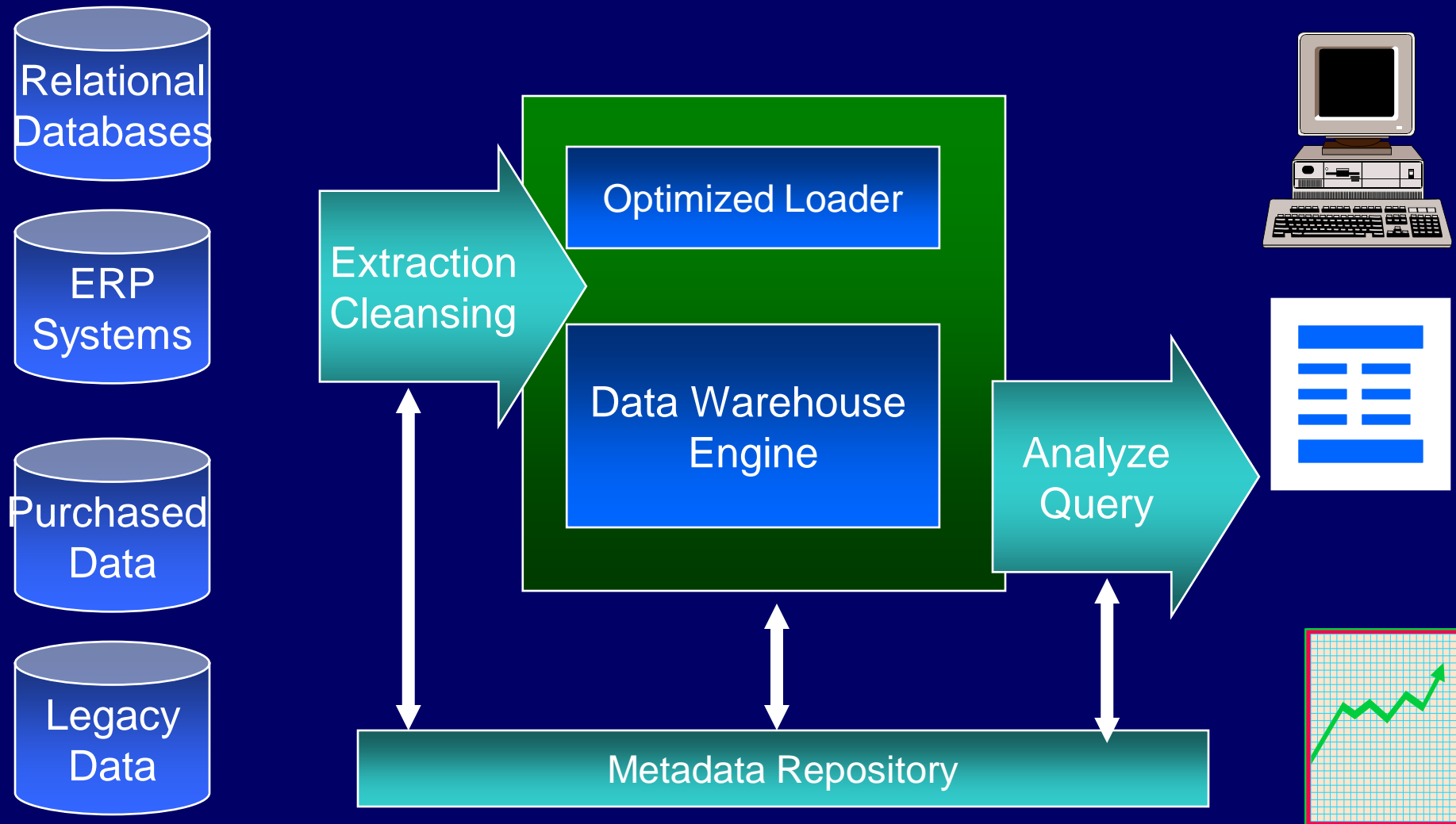


# I. Data Warehouses: Architecture, Design & Construction

- ⌘ DW Architecture
- ⌘ Loading, refreshing
- ⌘ Structuring/Modeling
- ⌘ DWs and Data Marts
- ⌘ Query Processing
  
- ⌘ demos, labs



# Data Warehouse Architecture



# Components of the Warehouse

- ⌘ Data Extraction and Loading
- ⌘ The Warehouse
- ⌘ Analyze and Query -- OLAP Tools
- ⌘ Metadata
- ⌘ Data Mining tools

# Loading the Warehouse



Cleaning the data  
before it is loaded

# Source Data

Operational/  
Source Data

Sequential

Legacy

Relational

External

⌘ Typically host based, legacy applications

⌘ Customized applications, COBOL, 3GL, 4GL

⌘ Point of Contact Devices

⌘ POS, ATM, Call switches

⌘ External Sources

⌘ Nielsen's, Acxiom, CMIE, Vendors, Partners



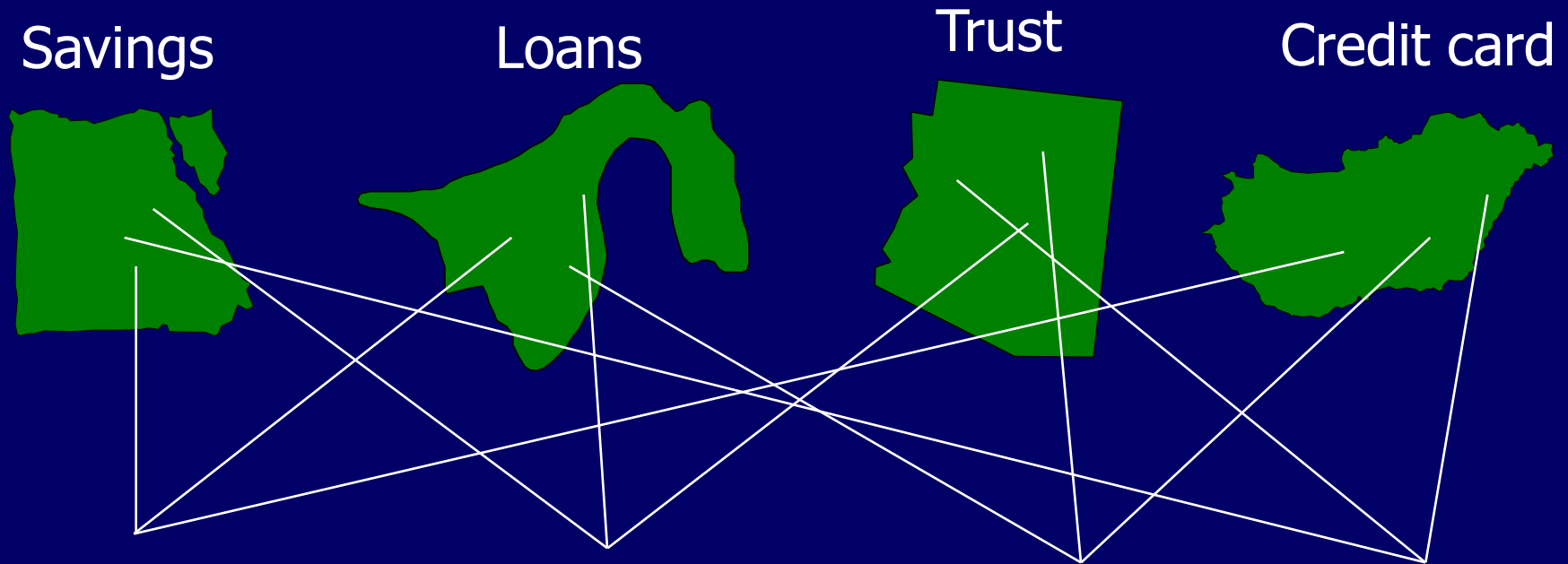
# Data Quality - The Reality

- ⌘ Tempting to think creating a data warehouse is simply extracting operational data and entering into a data warehouse
- ⌘ Nothing could be farther from the truth
- ⌘ Warehouse data comes from disparate questionable sources

# Data Quality - The Reality

- ⌘ Legacy systems no longer documented
- ⌘ Outside sources with questionable quality procedures
- ⌘ Production systems with no built in integrity checks and no integration
  - ☑ Operational systems are usually designed to solve a specific business problem and are rarely developed to a corporate plan
    - ☒ "And get it done quickly, we do not have time to worry about corporate standards..."

# Data Integration Across Sources



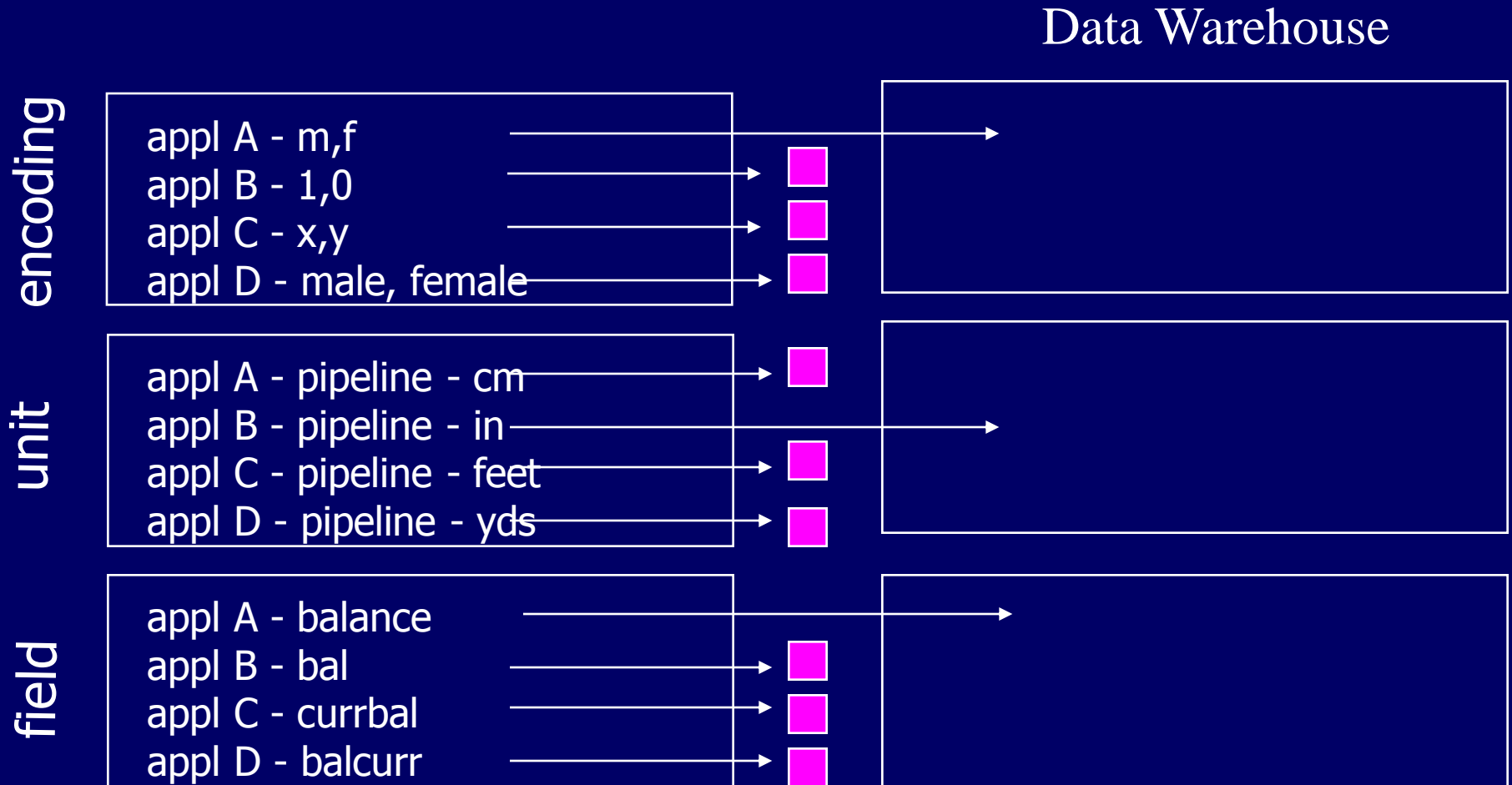
Same data  
different name

Different data  
Same name

Data found here  
nowhere else

Different keys  
same data

# Data Transformation Example



# Data Integrity Problems

- ⌘ Same person, different spellings
  - ☒ Agarwal, Agrawal, Aggarwal etc...
- ⌘ Multiple ways to denote company name
  - ☒ Persistent Systems, PSPL, Persistent Pvt. LTD.
- ⌘ Use of different names
  - ☒ mumbai, bombay
- ⌘ Different account numbers generated by different applications for the same customer
- ⌘ Required fields left blank
- ⌘ Invalid product codes collected at point of sale
  - ☒ manual entry leads to mistakes
  - ☒ "in case of a problem use 9999999"

# Data Transformation Terms

⌘ Extracting

⌘ Conditioning

⌘ Scrubbing

⌘ Merging

⌘ Householding

⌘ Enrichment

⌘ Scoring

⌘ Loading

⌘ Validating

⌘ Delta Updating

# Data Transformation Terms

## ⌘ Extracting

- ☑ Capture of data from operational source in "as is" status
- ☑ Sources for data generally in legacy mainframes in VSAM, IMS, IDMS, DB2; more data today in relational databases on Unix

## ⌘ Conditioning

- ☑ The conversion of data types from the source to the target data store (warehouse) -- always a relational database

# Data Transformation Terms

## ⌘Householding

- ☑ Identifying all members of a household (living at the same address)
- ☑ Ensures only one mail is sent to a household
- ☑ Can result in substantial savings: 1 lakh catalogues at Rs. 50 each costs Rs. 50 lakhs. A 2% savings would save Rs. 1 lakh.



# Data Transformation Terms

## ⌘ Enrichment

- ☑ Bring data from external sources to augment/enrich operational data. Data sources include Dunn and Bradstreet, A. C. Nielsen, CMIE, IMRA etc...

## ⌘ Scoring

- ☑ computation of a probability of an event. e.g..., chance that a customer will defect to AT&T from MCI, chance that a customer is likely to buy a new product

# Loads

⌘ After extracting, scrubbing, cleaning, validating etc. need to load the data into the warehouse

## ⌘ Issues

- ☒ huge volumes of data to be loaded
- ☒ small time window available when warehouse can be taken off line (usually nights)
- ☒ when to build index and summary tables
- ☒ allow system administrators to monitor, cancel, resume, change load rates
- ☒ Recover gracefully -- restart after failure from where you were and without loss of data integrity

# Load Techniques

⌘ Use SQL to append or insert new data

- ⌘ record at a time interface

- ⌘ will lead to random disk I/O's

⌘ Use batch load utility

# Load Taxonomy

⌘ Incremental versus Full loads

⌘ Online versus Offline loads

# Refresh

⌘ Propagate updates on source data to the warehouse

⌘ Issues:

- ☑ when to refresh

- ☑ how to refresh -- refresh techniques

# When to Refresh?

- ⌘ periodically (e.g., every night, every week) or after significant events
- ⌘ on every update: not warranted unless warehouse data require current data (up to the minute stock quotes)
- ⌘ refresh policy set by administrator based on user needs and traffic
- ⌘ possibly different policies for different sources

# Refresh Techniques

## ⌘ Full Extract from base tables

- ☑ read entire source table: too expensive
- ☑ maybe the only choice for legacy systems

# How To Detect Changes

- ⌘ Create a snapshot log table to record ids of updated rows of source data and timestamp
- ⌘ Detect changes by:
  - ☑ Defining after row triggers to update snapshot log when source table changes
  - ☑ Using regular transaction log to detect changes to source data



# Data Extraction and Cleansing

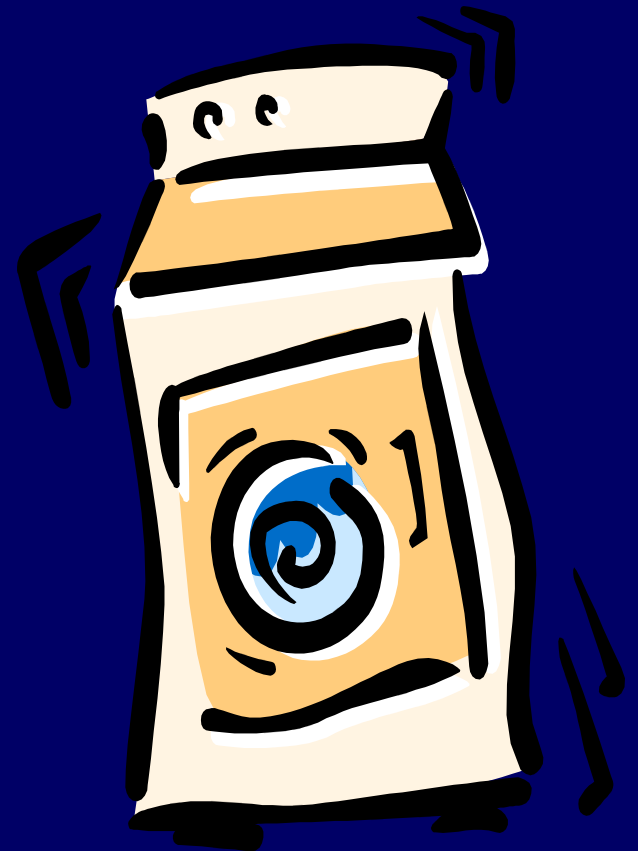
⌘ Extract data from existing operational and legacy data

⌘ Issues:

- ☒ Sources of data for the warehouse
- ☒ Data quality at the sources
- ☒ Merging different data sources
- ☒ Data Transformation
- ☒ How to propagate updates (on the sources) to the warehouse
- ☒ Terabytes of data to be loaded

# Scrubbing Data

- ⌘ Sophisticated transformation tools.
- ⌘ Used for cleaning the quality of data
- ⌘ Clean data is vital for the success of the warehouse
- ⌘ Example
  - ☑ Seshadri, Sheshadri, Sesadri, Seshadri S., Srinivasan Seshadri, etc. are the same person



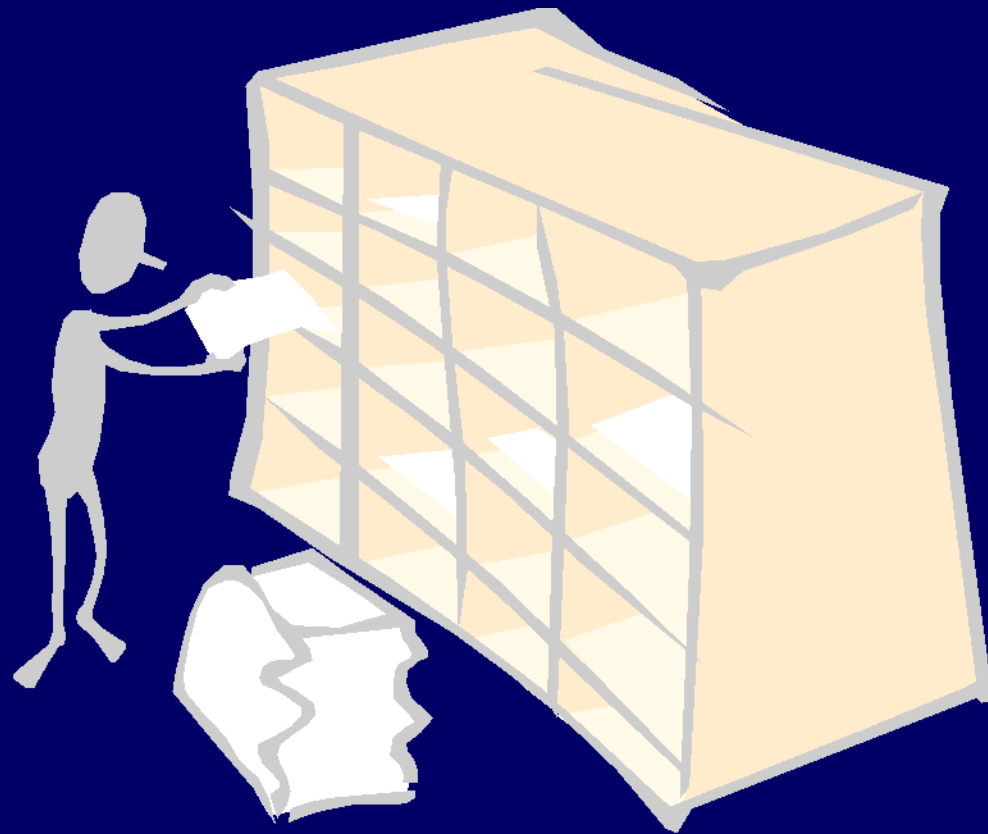
# Scrubbing Tools

⌘ Apertus -- Enterprise/Integrator

⌘ Vality -- IPE

⌘ Postal Soft

# Structuring/Modeling Issues



# Data -- Heart of the Data Warehouse

- ⌘ Heart of the data warehouse is the data itself!
- ⌘ Single version of the truth
- ⌘ Corporate memory
- ⌘ Data is organized in a way that represents business -- subject orientation

# Data Warehouse Structure

⌘ Subject Orientation -- customer, product, policy, account etc... A subject may be implemented as a set of related tables. E.g., customer may be five tables

# Data Warehouse Structure

⌞ base customer (1985-87)

⊗ custid, from date, to date, name, phone, dob

⌞ base customer (1988-90)

⊗ custid, from date, to date, name, credit rating,  
employer

⌞ customer activity (1986-89) -- monthly  
summary

⌞ customer activity detail (1987-89)

⊗ custid, activity date, amount, clerk id, order no

⌞ customer activity detail (1990-91)

⊗ custid, activity date, amount, line item no, order no

Time is  
part of  
key of  
each table

# Data Granularity in Warehouse

## ⌘ Summarized data stored

- ☑ reduce storage costs
- ☑ reduce cpu usage
- ☑ increases performance since smaller number of records to be processed
- ☑ design around traditional high level reporting needs
- ☑ tradeoff with volume of data to be stored and detailed usage of data



# Granularity in Warehouse

⌘ Can not answer some questions with summarized data

☑ Did Anand call Seshadri last month?

Not possible to answer if total duration of calls by Anand over a month is only maintained and individual call details are not.

⌘ Detailed data too voluminous

# Granularity in Warehouse

⌘ Tradeoff is to have dual level of granularity

- ☑ Store summary data on disks

  - ☒ 95% of DSS processing done against this data

- ☑ Store detail on tapes

  - ☒ 5% of DSS processing against this data

# Vertical Partitioning

Acct. No	Name	Balance	Date Opened	Interest Rate	Address
-------------	------	---------	-------------	------------------	---------

Frequently  
accessed

Acct. No	Balance
-------------	---------

Rarely  
accessed

Acct. No	Name	Date Opened	Interest Rate	Address
-------------	------	-------------	------------------	---------

Smaller table  
and so less I/O

# Derived Data

- ⌘ Introduction of derived (calculated data) may often help
- ⌘ Have seen this in the context of dual levels of granularity
- ⌘ Can keep auxiliary views and indexes to speed up query processing

# Schema Design

## ⌘ Database organization

- ☑ must look like business
- ☑ must be recognizable by business user
- ☑ approachable by business user
- ☑ Must be simple

## ⌘ Schema Types

- ☑ Star Schema
- ☑ Fact Constellation Schema
- ☑ Snowflake schema

# Dimension Tables

## ⌘ Dimension tables

- ☑ Define business in terms already familiar to users
- ☑ Wide rows with lots of descriptive text
- ☑ Small tables (about a million rows)
- ☑ Joined to fact table by a foreign key
- ☑ heavily indexed
- ☑ typical dimensions
  - ☒ time periods, geographic region (markets, cities), products, customers, salesperson, etc.

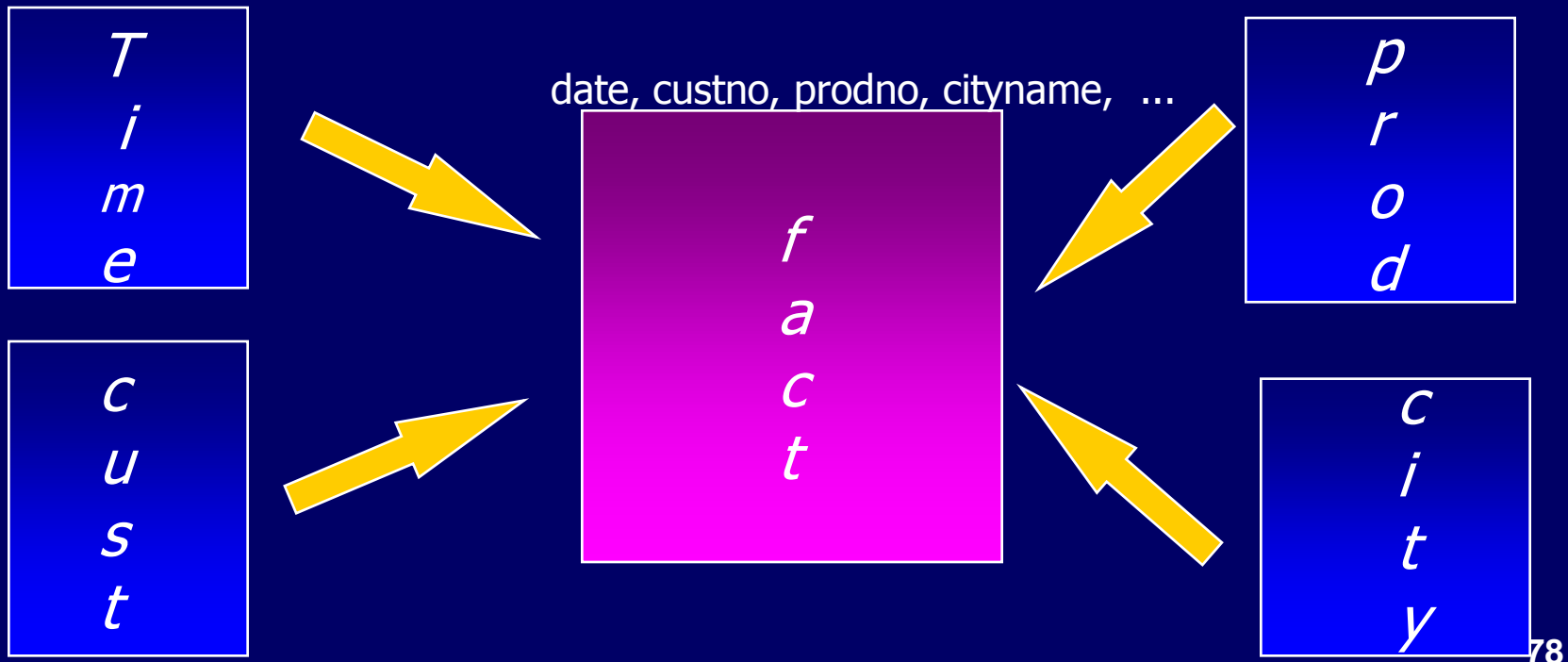
# Fact Table

## ⌘ Central table

- ☑ mostly raw numeric items
- ☑ narrow rows, a few columns at most
- ☑ large number of rows (millions to a billion)
- ☑ Access via dimensions

# Star Schema

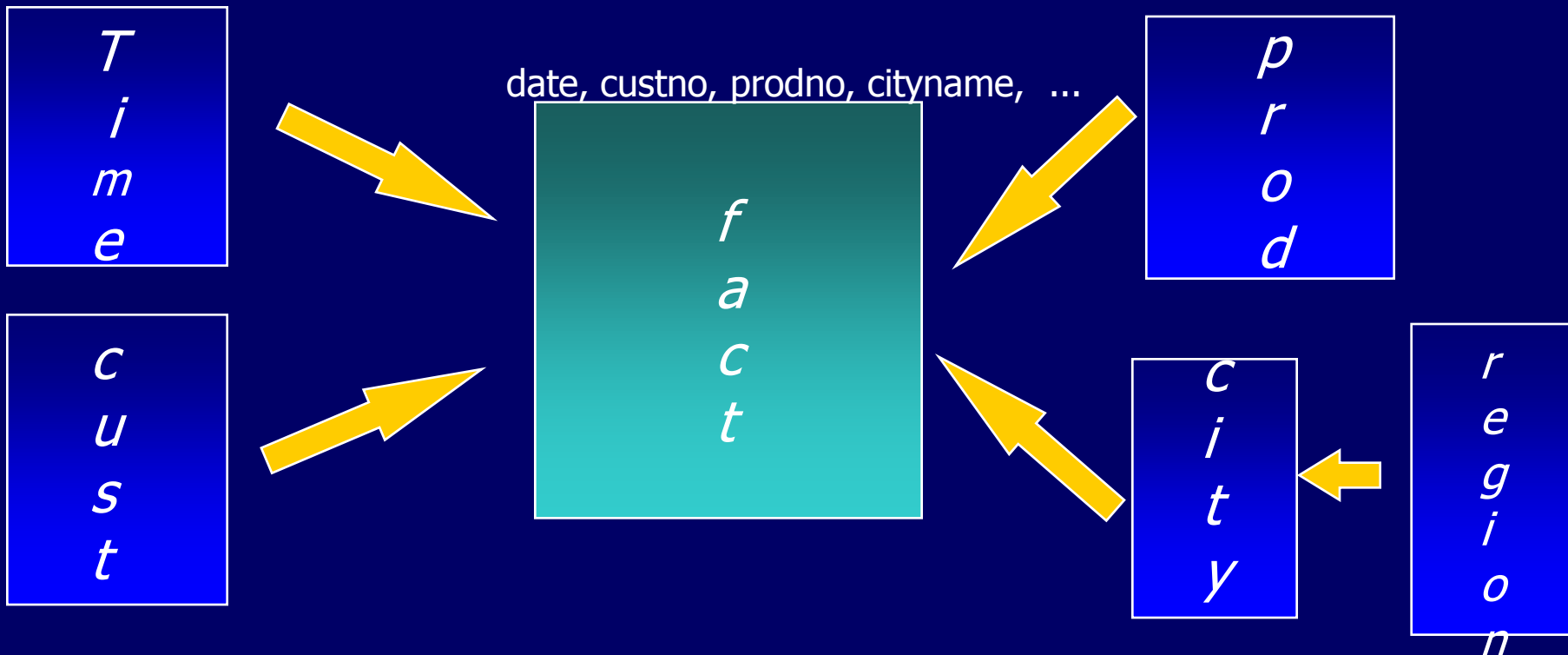
- ⌘ A single fact table and for each dimension one dimension table
- ⌘ Does not capture hierarchies directly





# Snowflake schema

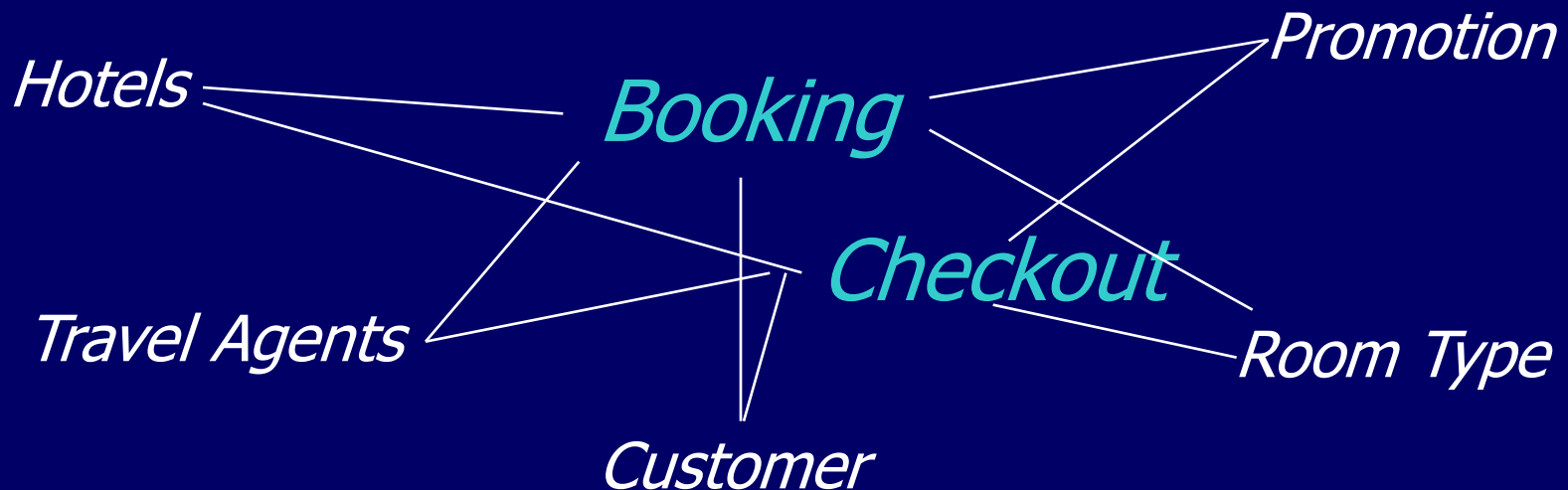
- ⌘ Represent dimensional hierarchy directly by normalizing tables.
- ⌘ Easy to maintain and saves storage



# Fact Constellation

## ⌘ Fact Constellation

- ☑ Multiple fact tables that share many dimension tables
- ☑ Booking and Checkout may share many dimension tables in the hotel industry



# De-normalization

- ⌘ Normalization in a data warehouse may lead to lots of small tables
- ⌘ Can lead to excessive I/O's since many tables have to be accessed
- ⌘ De-normalization is the answer especially since updates are rare

# Creating Arrays

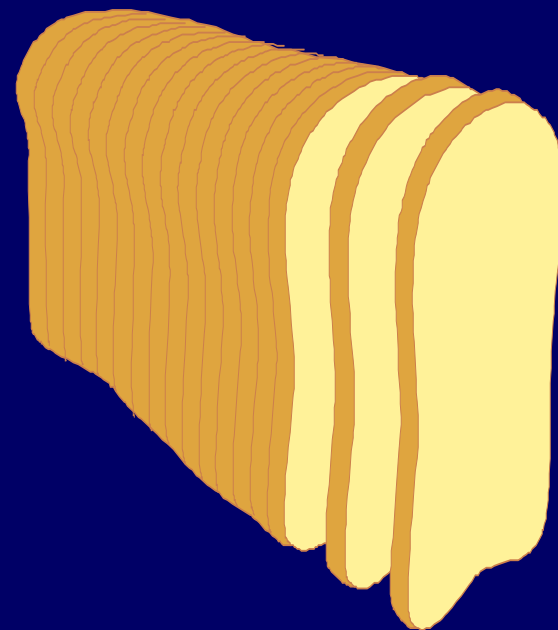
- ⌘ Many times each occurrence of a sequence of data is in a different physical location
- ⌘ Beneficial to collect all occurrences together and store as an array in a single row
- ⌘ Makes sense only if there are a stable number of occurrences which are accessed together
- ⌘ In a data warehouse, such situations arise naturally due to time based orientation
  - ⌘ can create an array by month

# Selective Redundancy

- ⌘ Description of an item can be stored redundantly with order table -- most often item description is also accessed with order table
- ⌘ Updates have to be careful

# Partitioning

- ⌘ Breaking data into several physical units that can be handled separately
- ⌘ Not a question of *whether* to do it in data warehouses but *how* to do it
- ⌘ Granularity and partitioning are key to effective implementation of a warehouse



# Why Partition?

⌘ Flexibility in managing data

⌘ Smaller physical units allow

- ☑ easy restructuring

- ☑ free indexing

- ☑ sequential scans if needed

- ☑ easy reorganization

- ☑ easy recovery

- ☑ easy monitoring

# Criterion for Partitioning

⌘ Typically partitioned by

- ☑ date

- ☑ line of business

- ☑ geography

- ☑ organizational unit

- ☑ any combination of above



# Where to Partition?

- ⌘ Application level or DBMS level

- ⌘ Makes sense to partition at application level

  - ☑ Allows different definition for each year

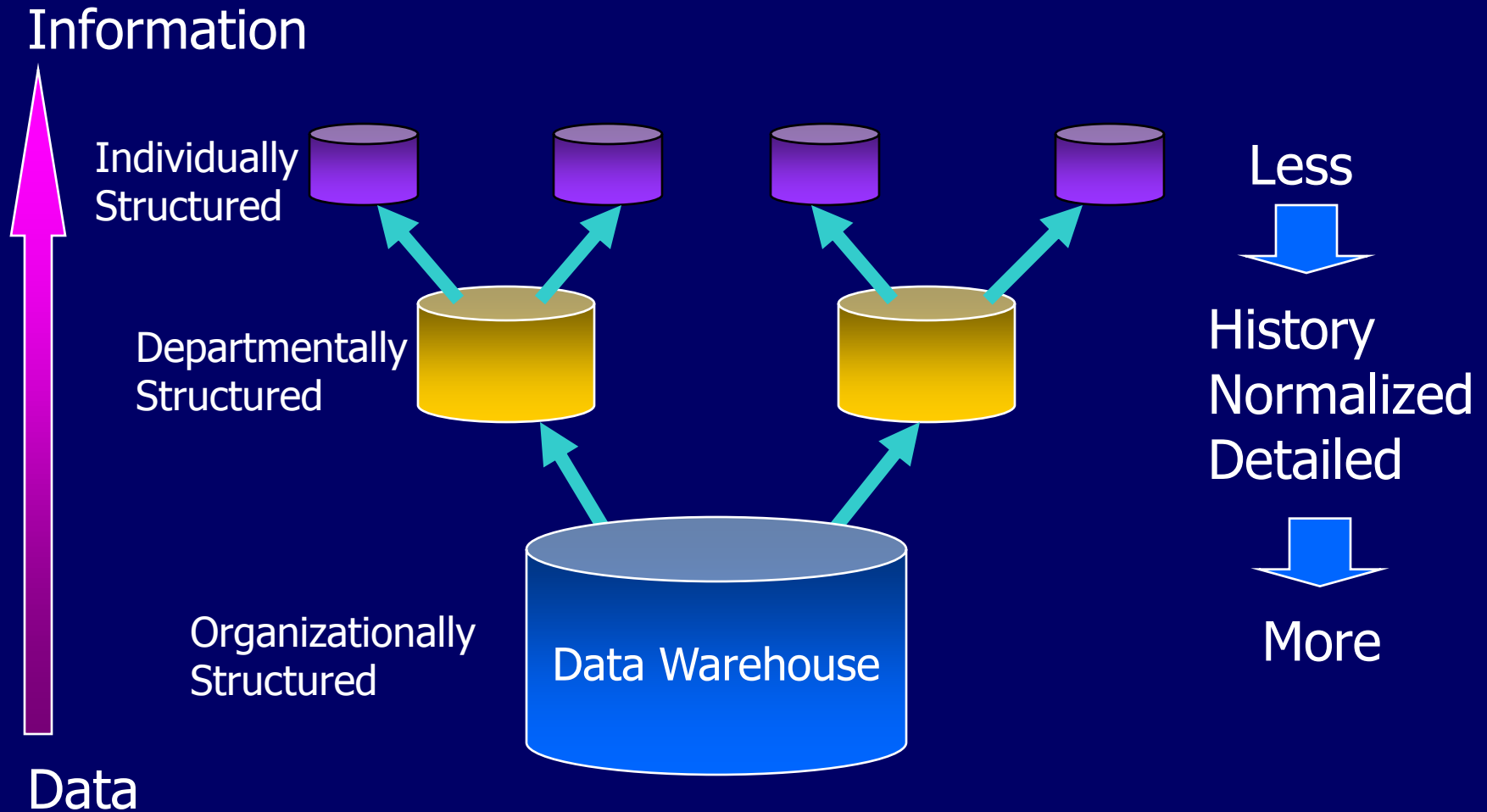
    - ☒ Important since warehouse spans many years and as business evolves definition changes

  - ☑ Allows data to be moved between processing complexes easily

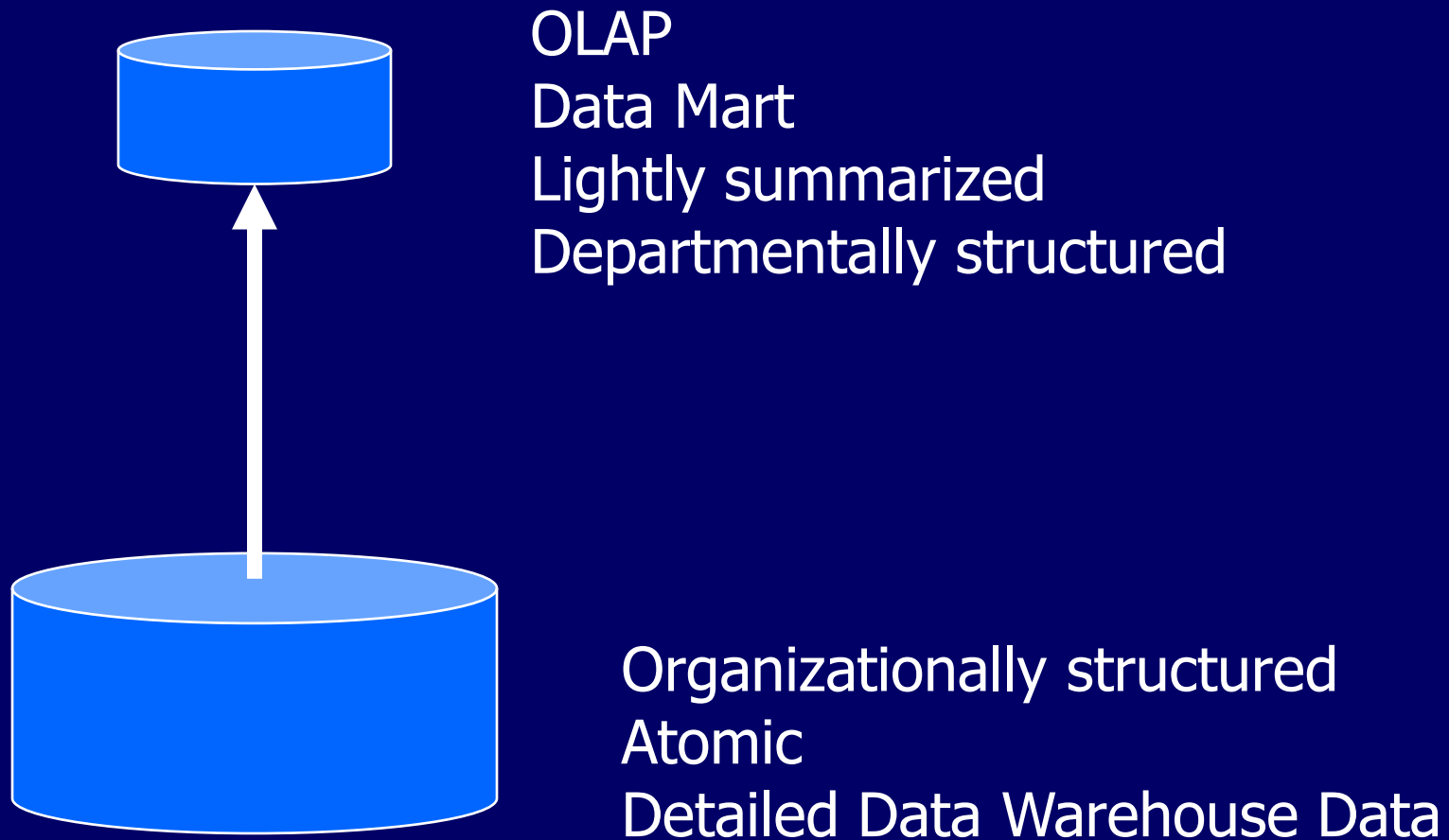
# Data Warehouse vs. Data Marts

What comes first

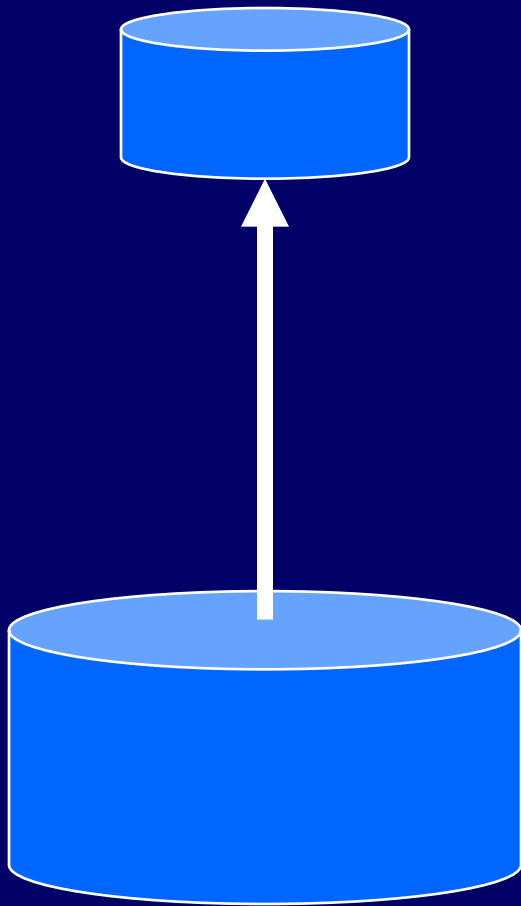
# From the Data Warehouse to Data Marts



# Data Warehouse and Data Marts

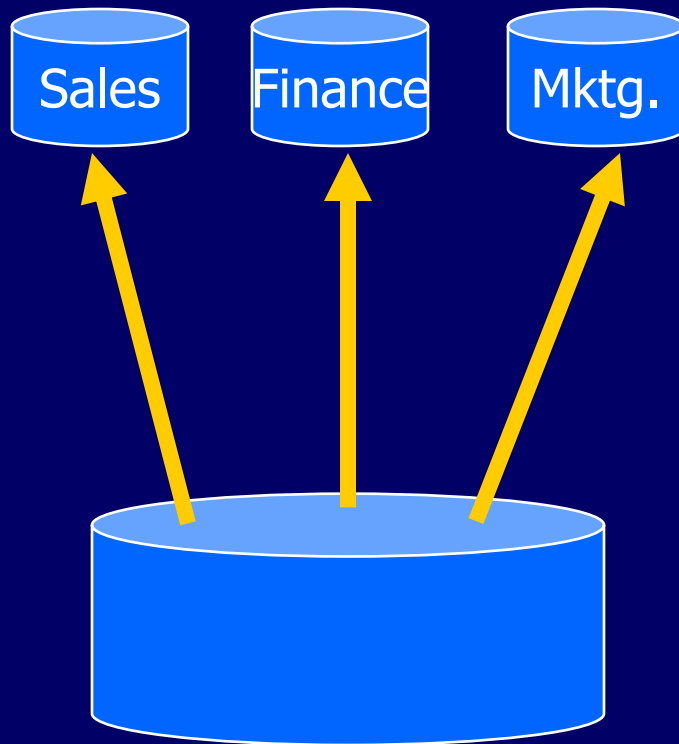


# Characteristics of the Departmental Data Mart



- ⌘ OLAP
- ⌘ Small
- ⌘ Flexible
- ⌘ Customized by Department
- ⌘ Source is departmentally structured data warehouse

# Techniques for Creating Departmental Data Mart



⌘ OLAP

⌘ Subset

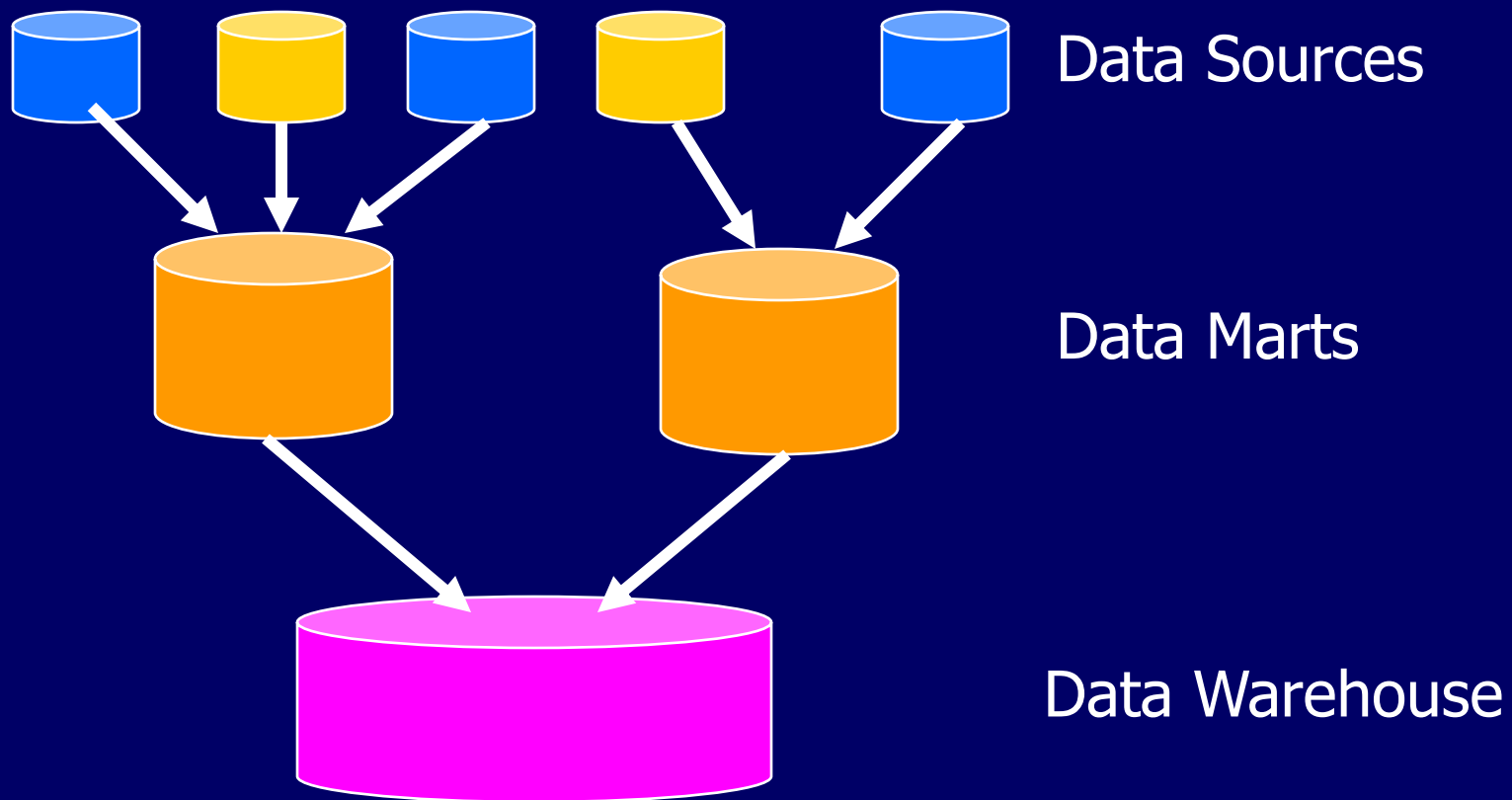
⌘ Summarized

⌘ Superset

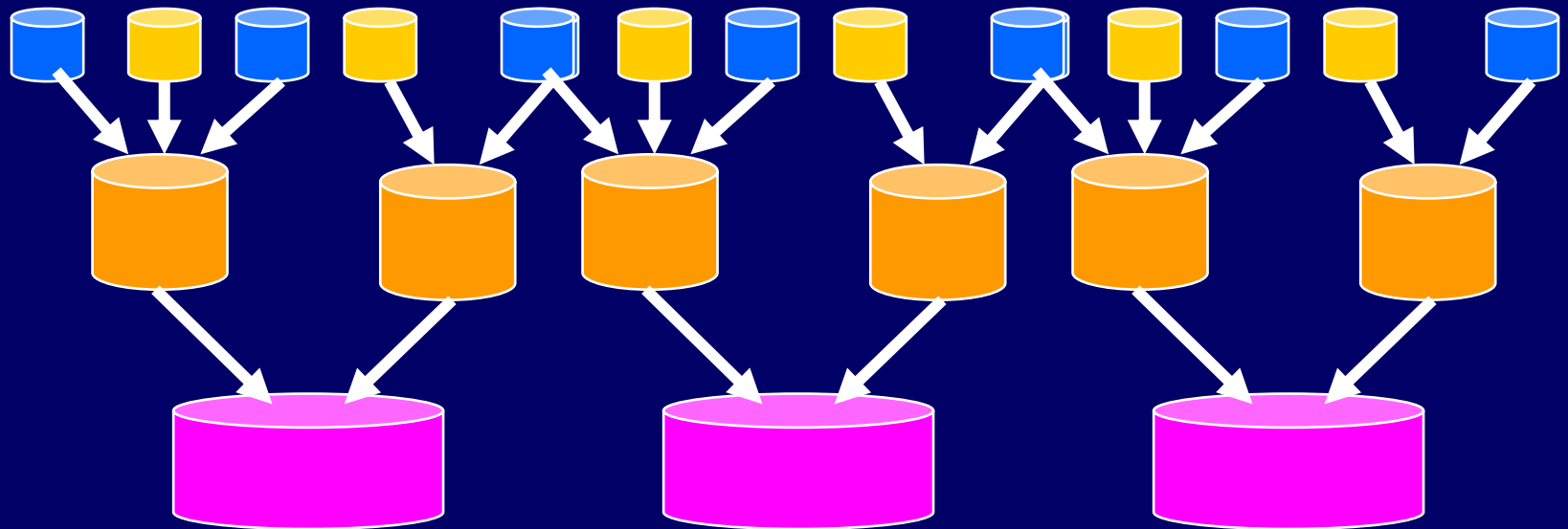
⌘ Indexed

⌘ Arrayed

# Data Mart Centric



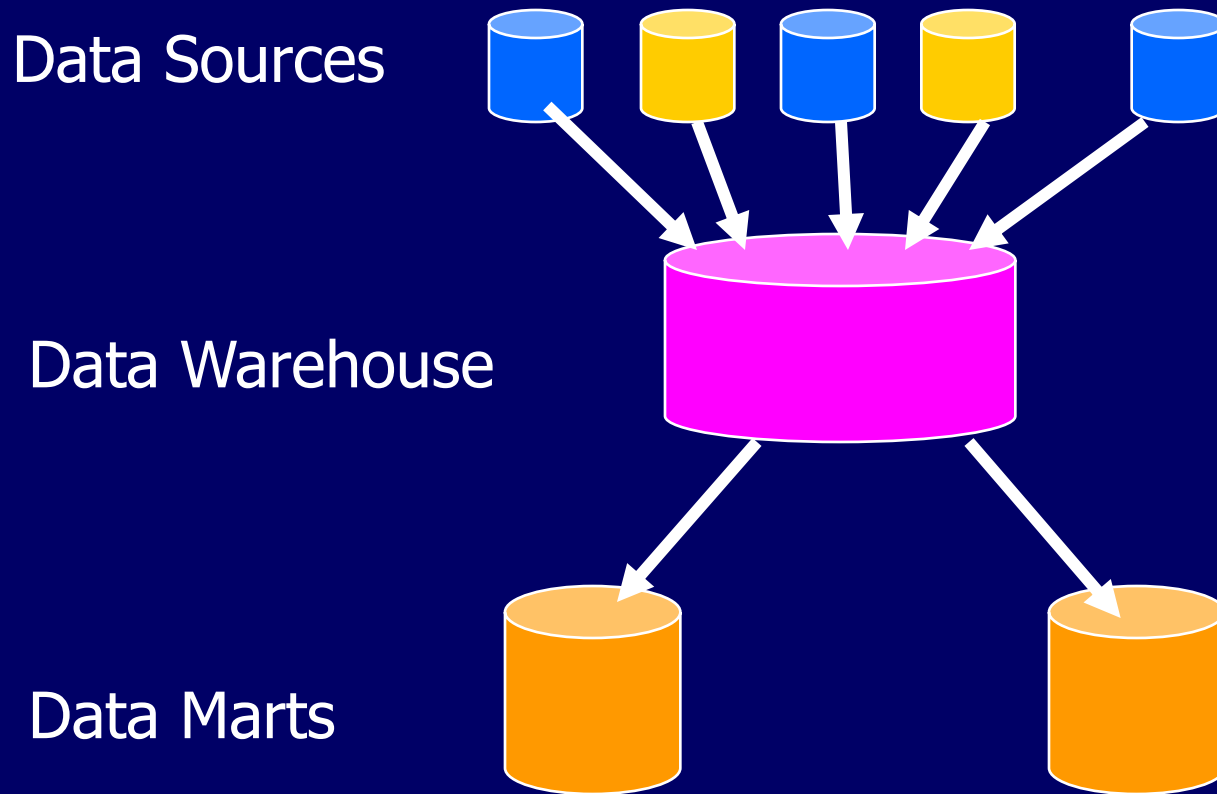
# Problems with Data Mart Centric Solution



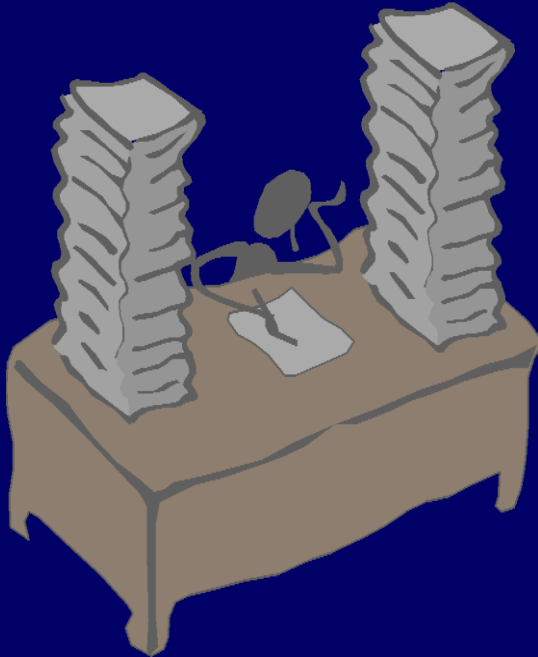
If you end up creating multiple warehouses, integrating them is a problem



# True Warehouse



# Query Processing



⌘ Indexing

⌘ Pre computed  
views/aggregates

⌘ SQL extensions

# Indexing Techniques

- ⌘ Exploiting indexes to reduce scanning of data is of crucial importance

- ⌘ Bitmap Indexes

- ⌘ Join Indexes

- ⌘ Other Issues

  - ☑ Text indexing

  - ☑ Parallelizing and sequencing of index builds and incremental updates

# Indexing Techniques

## ⌘ Bitmap index:

- ☑ A collection of bitmaps -- one for each distinct value of the column
- ☑ Each bitmap has  $N$  bits where  $N$  is the number of rows in the table
- ☑ A bit corresponding to a value  $v$  for a row  $r$  is set if and only if  $r$  has the value for the indexed attribute

# BitMap Indexes

- ⌘ An alternative representation of RID-list
- ⌘ Specially advantageous for low-cardinality domains
- ⌘ Represent each row of a table by a bit and the table as a bit vector
- ⌘ There is a distinct bit vector  $B_v$  for each value  $v$  for the domain
- ⌘ Example: the attribute sex has values M and F. A table of 100 million people needs 2 lists of 100 million bits

# Bitmap Index

<i>gender</i>	<i>vote</i>	
M	Y	
F	Y	
F	N	
M	N	
F	Y	
F	N	

**Customer**

*gender (f)*

0
1
1
0
1
1

*vote (y)*

1
1
0
0
1
0



*result*

0
1
0
0
1
0

**Query : select \* from customer where  
gender = 'F' and vote = 'Y'**

# Bit Map Index

**Base Table**

Cust	Region	Rating
C1	N	H
C2	S	M
C3	W	L
C4	W	H
C5	S	L
C6	W	L
C7	N	H

**Region Index**

Row ID	N	S	E	W
1	1	0	0	0
2	0	1	0	0
3	0	0	0	1
4	0	0	0	1
5	0	1	0	0
6	0	0	0	1
7	1	0	0	0

**Rating Index**

Row ID	H	M	L
1	1	0	0
2	0	1	0
3	0	0	0
4	0	0	0
5	0	1	0
6	0	0	0
7	1	0	0

***Customers where***

**Region = W**

***And***

**Rating = M**

# BitMap Indexes

- ⌘ Comparison, join and aggregation operations are reduced to bit arithmetic with dramatic improvement in processing time
- ⌘ Significant reduction in space and I/O (30:1)
- ⌘ Adapted for higher cardinality domains as well.
- ⌘ Compression (e.g., run-length encoding) exploited
- ⌘ Products that support bitmaps: Model 204, TargetIndex (Redbrick), IQ (Sybase), Oracle 7.3



# Join Indexes

- ⌘ Pre-computed joins
- ⌘ A join index between a fact table and a dimension table correlates a dimension tuple with the fact tuples that have the same value on the common dimensional attribute
  - ☑ e.g., a join index on *city* dimension of *calls* fact table
  - ☑ correlates for each city the calls (in the *calls* table) from that city

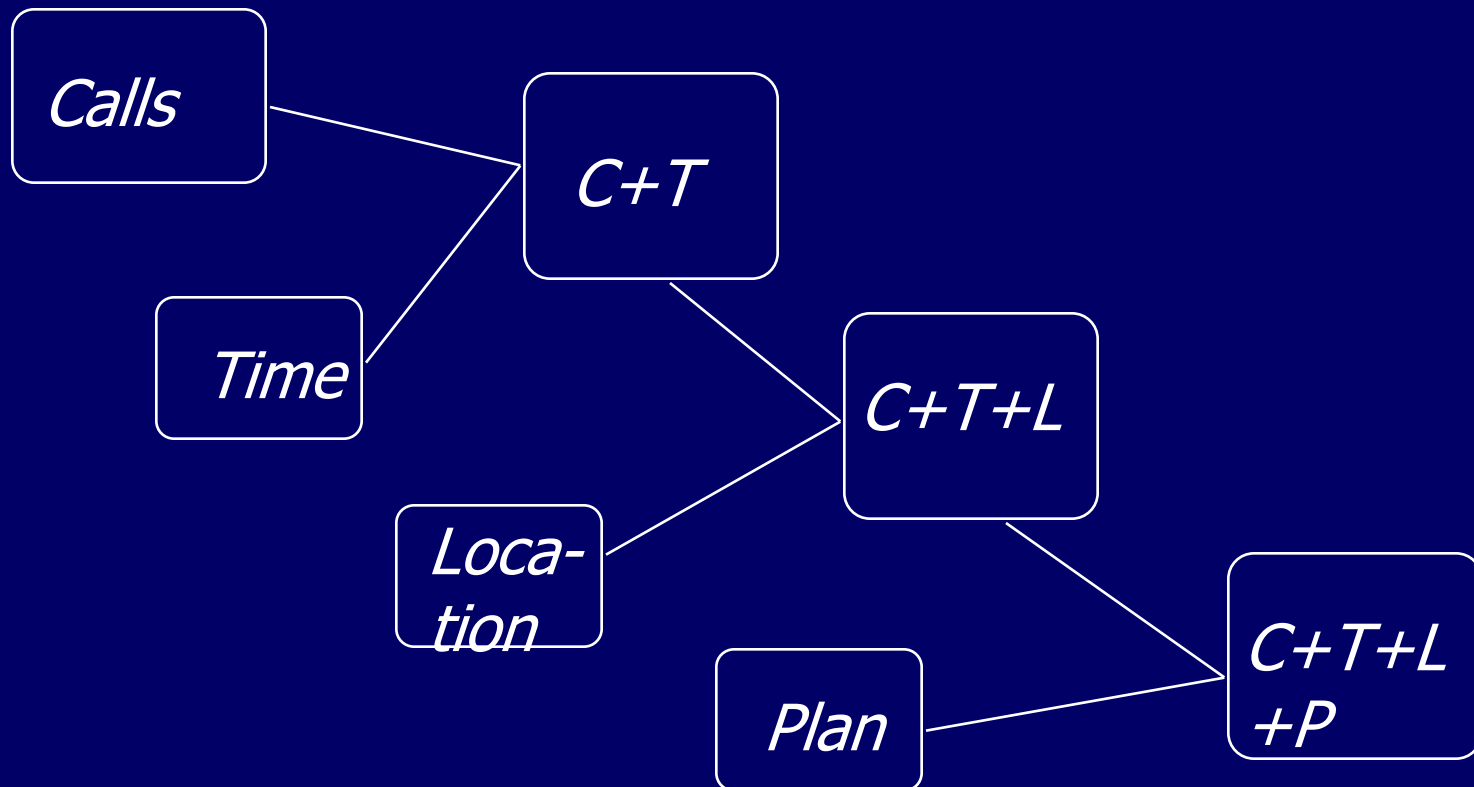
# Join Indexes

⌘ Join indexes can also span multiple dimension tables

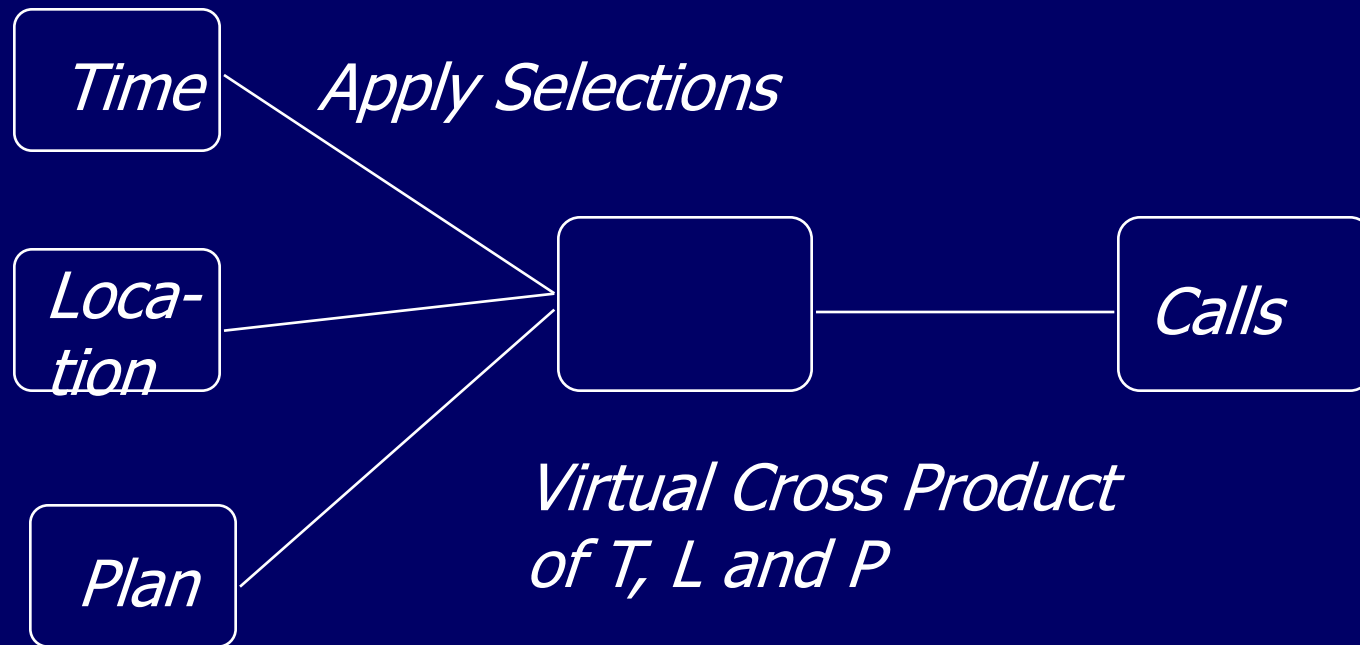
☑ e.g., a join index on *city* and *time* dimension of *calls* fact table

# Star Join Processing

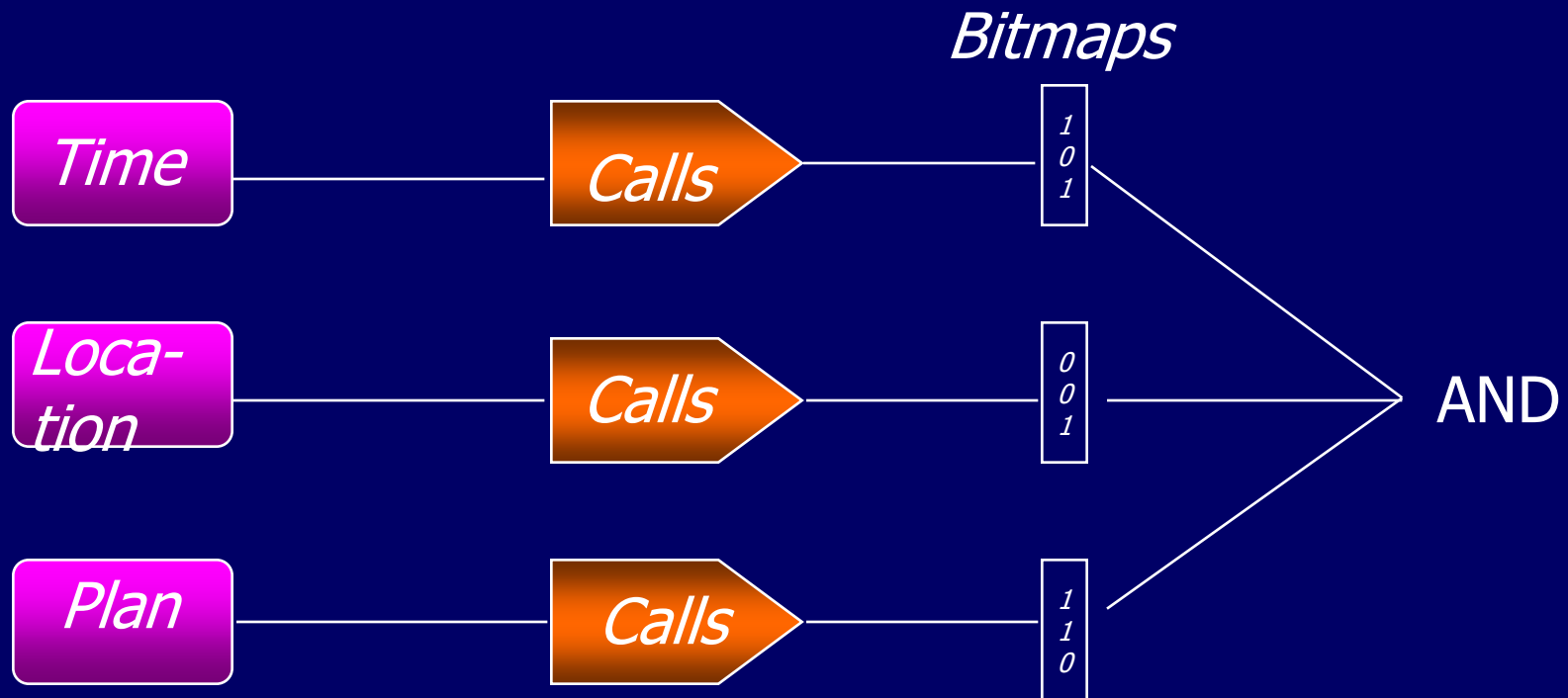
⌘ Use join indexes to join dimension and fact table



# Optimized Star Join Processing



# Bitmapped Join Processing



# Intelligent Scan

- ⌘ Piggyback multiple scans of a relation (Redbrick)

- ⏏ piggybacking also done if second scan starts a little while after the first scan

# Parallel Query Processing

## ⌘ Three forms of parallelism

- ☑ Independent

- ☑ Pipelined

- ☑ Partitioned and “partition and replicate”

## ⌘ Deterrents to parallelism

- ☑ startup

- ☑ communication

# Parallel Query Processing

## ⌘ Partitioned Data

- ⌘ Parallel scans

- ⌘ Yields I/O parallelism

## ⌘ Parallel algorithms for relational operators

- ⌘ Joins, Aggregates, Sort

## ⌘ Parallel Utilities

- ⌘ Load, Archive, Update, Parse, Checkpoint, Recovery

## ⌘ Parallel Query Optimization



# Pre-computed Aggregates

⌘ Keep aggregated data for efficiency (pre-computed queries)

⌘ Questions

- ☑ Which aggregates to compute?

- ☑ How to update aggregates?

- ☑ How to use pre-computed aggregates in queries?

# Pre-computed Aggregates

⌘ Aggregated table can be maintained by the

- ☑ warehouse server

- ☑ middle tier

- ☑ client applications

⌘ Pre-computed aggregates -- special case of materialized views -- same questions and issues remain

# SQL Extensions

⌘ Extended family of aggregate functions

- ☑ rank (top 10 customers)

- ☑ percentile (top 30% of customers)

- ☑ median, mode

- ☑ Object Relational Systems allow addition of new aggregate functions

# SQL Extensions

## ⌘ Reporting features

- ⌘ running total, cumulative totals

## ⌘ Cube operator

- ⌘ group by on all subsets of a set of attributes (month,city)
- ⌘ redundant scan and sorting of data can be avoided

# Red Brick has Extended set of Aggregates

```
⌘ Select month, dollars, cume(dollars) as  
  run_dollars, weight, cume(weight) as  
  run_weights  
from sales, market, product, period t  
where year = 1993  
and product like 'Columbian%'  
and city like 'San Fr%'  
order by t.perkey
```

# RISQL (Red Brick Systems) Extensions

## ⌘ Aggregates

- ⌘ CUME

- ⌘ MOVINGAVG

- ⌘ MOVINGSUM

- ⌘ RANK

- ⌘ TERTILE

- ⌘ RATIO TO REPORT

## ⌘ Calculating Row Subtotals

- ⌘ BREAK BY

## ⌘ Sophisticated Date Time Support

- ⌘ DATEDIFF

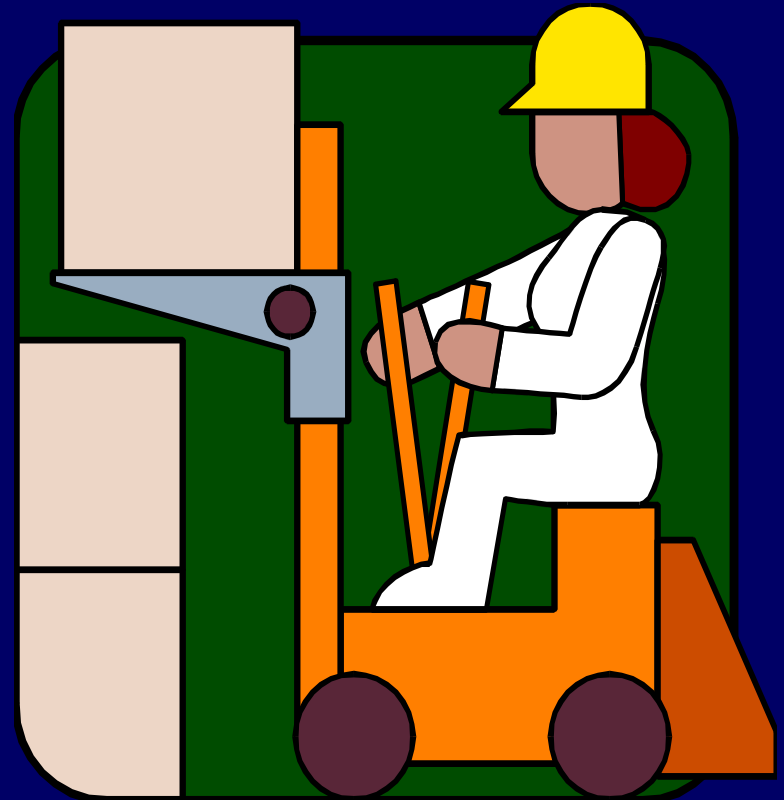
## ⌘ Using SubQueries in calculations

# Using SubQueries in Calculations

```
select product, dollars as jun97_sales,  
    (select sum(s1.dollars)  
     from market mi, product pi, period, ti, sales si  
     where pi.product = product.product  
     and   ti.year    = period.year  
     and   mi.city    = market.city) as total97_sales,  
    100 * dollars/  
    (select sum(s1.dollars)  
     from market mi, product pi, period, ti, sales si  
     where pi.product = product.product  
     and   ti.year    = period.year  
     and   mi.city    = market.city) as percent_of_yr  
from market, product, period, sales  
where year = 1997  
and   month = 'June' and city like 'Ahmed%'  
order by product;
```

# Course Overview

- ⌘ The course:  
what and how
- ⌘ 0. Introduction
- ⌘ I. Data Warehousing
- ⌘ II. Decision Support  
and OLAP
- ⌘ III. Data Mining
- ⌘ IV. Looking Ahead
- ⌘ Demos and Labs





## II. On-Line Analytical Processing (OLAP)



Making Decision  
Support Possible

# Limitations of SQL



“A Freshman in  
Business needs  
a Ph.D. in SQL”

-- Ralph Kimball

# Typical OLAP Queries

- ⌘ Write a multi-table join to compare sales for each product line YTD this year vs. last year.
- ⌘ Repeat the above process to find the top 5 product contributors to margin.
- ⌘ Repeat the above process to find the sales of a product line to new vs. existing customers.
- ⌘ Repeat the above process to find the customers that have had negative sales growth.

# What Is OLAP?

- ⌘ Online Analytical Processing - coined by EF Codd in 1994 paper contracted by Arbor Software\*
- ⌘ Generally synonymous with earlier terms such as Decisions Support, Business Intelligence, Executive Information System
- ⌘ OLAP = Multidimensional Database
- ⌘ MOLAP: Multidimensional OLAP (Arbor Essbase, Oracle Express)
- ⌘ ROLAP: Relational OLAP (Informix MetaCube, Microstrategy DSS Agent)

\* Reference: [http://www.arborsoft.com/essbase/wht\\_ppr/coddTOC.html](http://www.arborsoft.com/essbase/wht_ppr/coddTOC.html)

# The OLAP Market

## ⌘ Rapid growth in the enterprise market

- ☒ 1995: \$700 Million

- ☒ 1997: \$2.1 Billion

## ⌘ Significant consolidation activity among major DBMS vendors

- ☒ 10/94: Sybase acquires ExpressWay

- ☒ 7/95: Oracle acquires Express

- ☒ 11/95: Informix acquires Metacube

- ☒ 1/97: Arbor partners up with IBM

- ☒ 10/96: Microsoft acquires Panorama

## ⌘ Result: OLAP shifted from small vertical niche to mainstream DBMS category

# Strengths of OLAP

- ⌘ It is a powerful visualization paradigm
- ⌘ It provides fast, interactive response times
- ⌘ It is good for analyzing time series
- ⌘ It can be useful to find some clusters and outliers
- ⌘ Many vendors offer OLAP tools

# OLAP Is FASMI

⌘ Fast

⌘ Analysis

⌘ Shared

⌘ Multidimensional

⌘ Information

**Nigel Pendse, Richard Creath - The OLAP Report**

# Multi-dimensional Data

⌘ “Hey...I sold \$100M worth of goods”



Dimensions: Product, Region, Time  
Hierarchical summarization paths

Product  
Industry

Category

Product

Region  
Country

Region

City

Office

Time  
Year

Quarter

Month

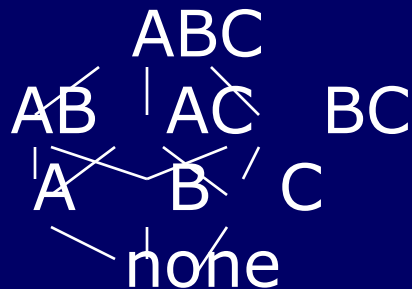
Week

Day<sup>126</sup>



# Data Cube Lattice

## ⌘ Cube lattice



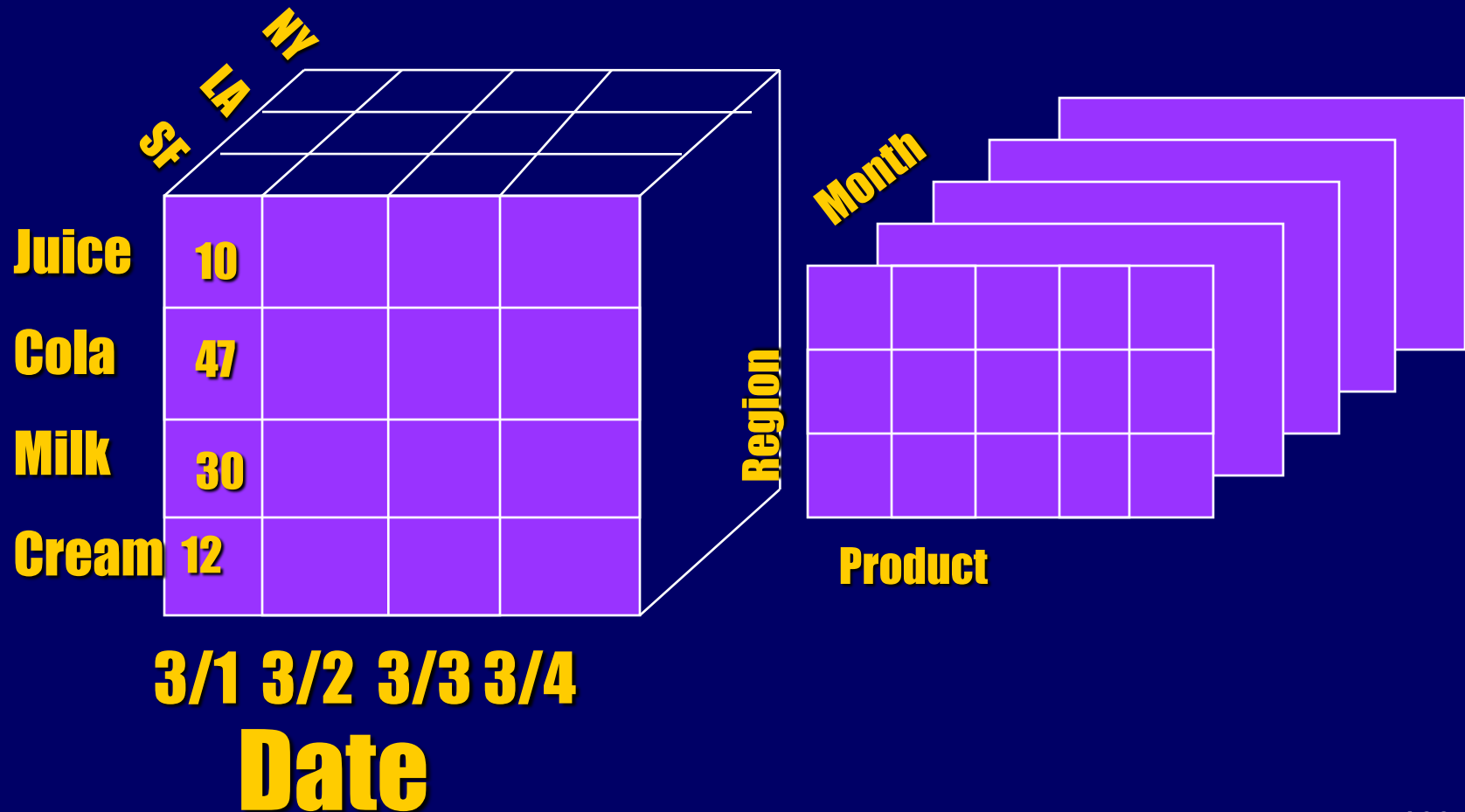
- ⌘ Can materialize some groupbys, compute others on demand
- ⌘ Question: which groupbys to materialize?
- ⌘ Question: what indices to create
- ⌘ Question: how to organize data (chunks, etc)

# Visualizing Neighbors is simpler

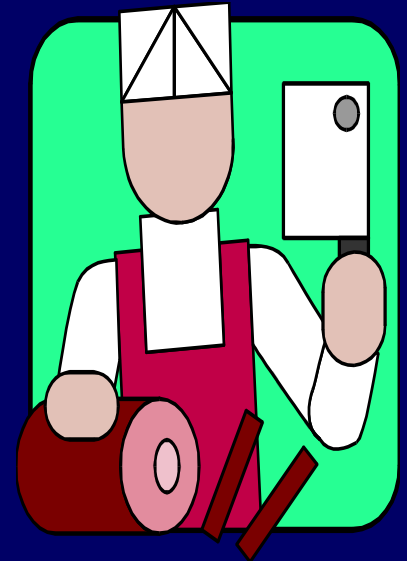
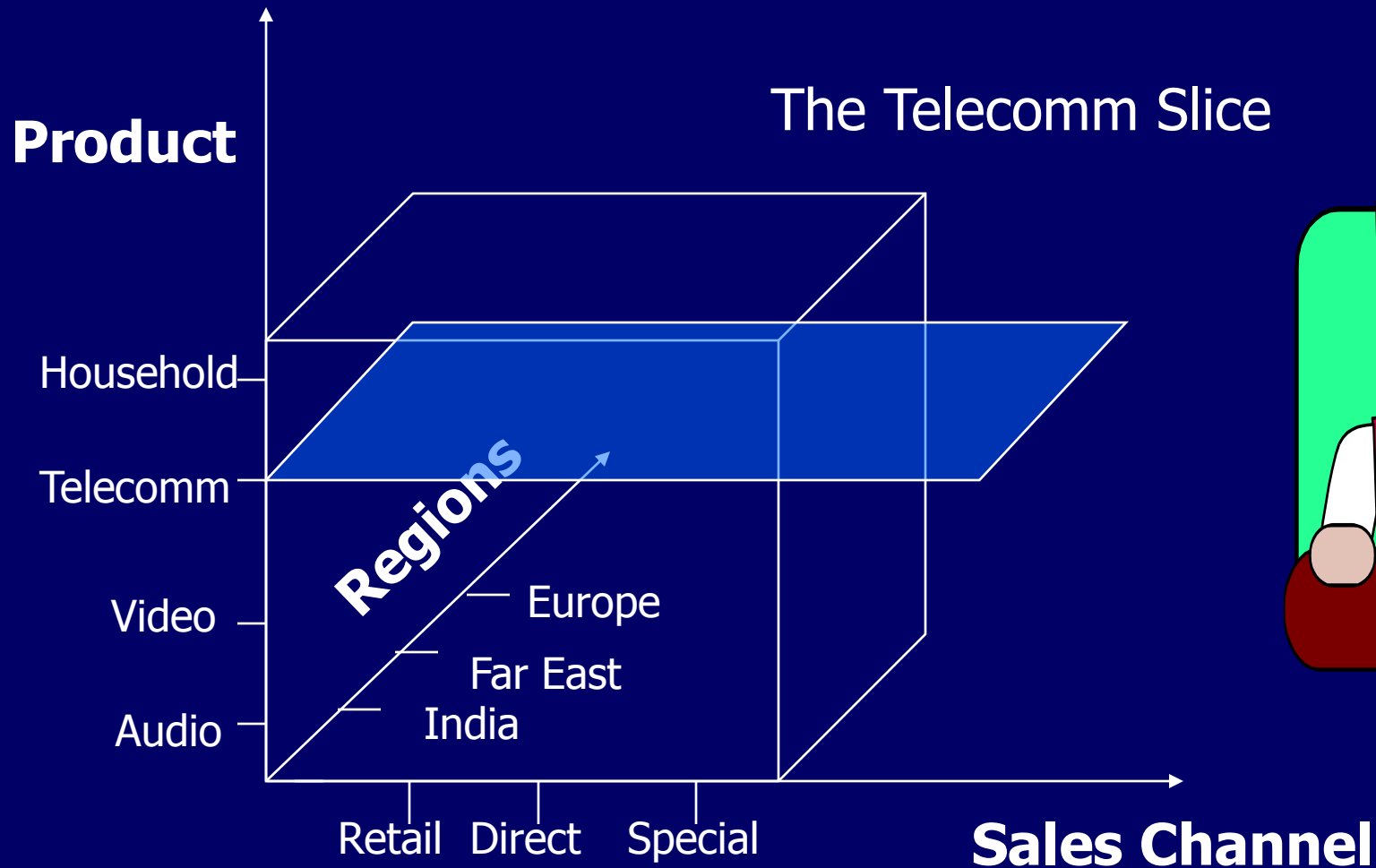
	1	2	3	4	5	6	7	8
Apr								
May								
Jun								
Jul								
Aug								
Sep								
Oct								
Nov								
Dec								
Jan								
Feb								
Mar								

Month	Store	Sales
Apr	1	
Apr	2	
Apr	3	
Apr	4	
Apr	5	
Apr	6	
Apr	7	
Apr	8	
May	1	
May	2	
May	3	
May	4	
May	5	
May	6	
May	7	
May	8	
Jun	1	
Jun	2	

# A Visual Operation: Pivot (Rotate)



# "Slicing and Dicing"



# Roll-up and Drill Down

Higher Level of  
Aggregation

Roll Up



- ⌘ Sales Channel
- ⌘ Region
- ⌘ Country
- ⌘ State
- ⌘ Location Address
- ⌘ Sales Representative

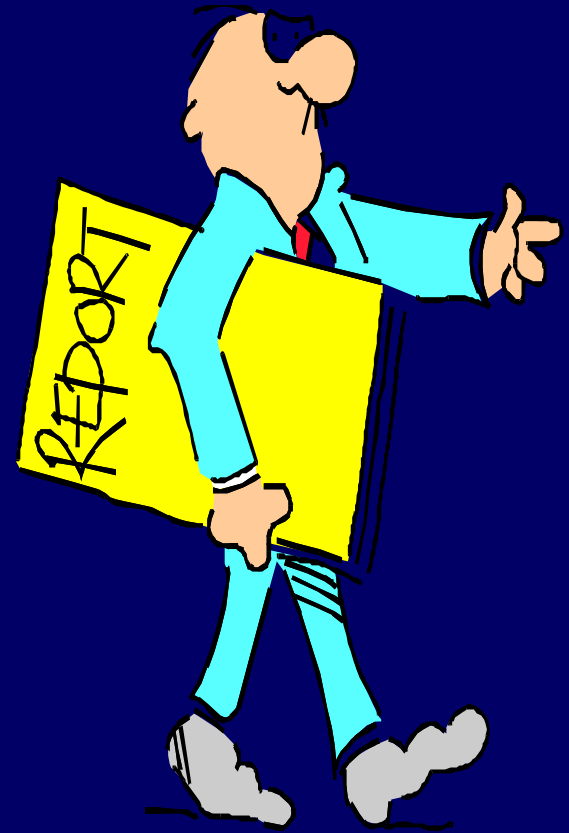
Drill-Down



Low-level  
Details

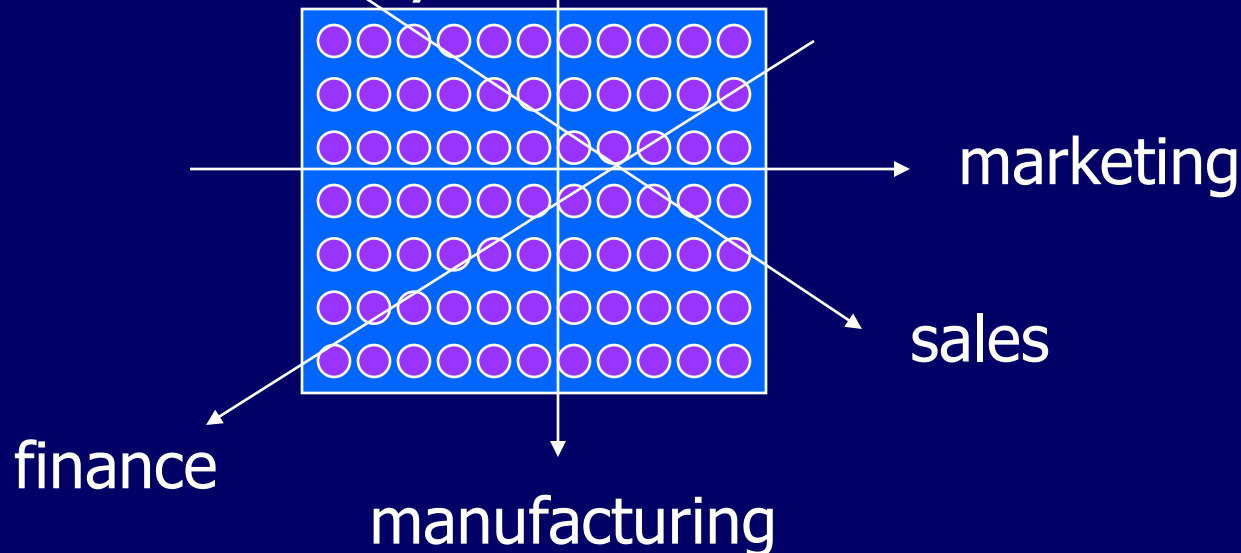
# Nature of OLAP Analysis

- ⌘ Aggregation -- (total sales, percent-to-total)
- ⌘ Comparison -- Budget vs. Expenses
- ⌘ Ranking -- Top 10, quartile analysis
- ⌘ Access to detailed and aggregate data
- ⌘ Complex criteria specification
- ⌘ Visualization

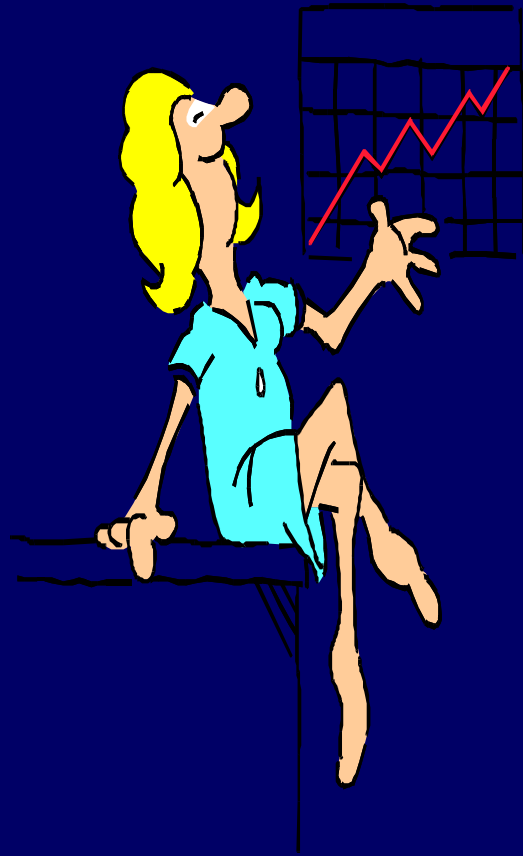


# Organizationally Structured Data

⌘ Different Departments look at the same detailed data in different ways. Without the detailed, organizationally structured data as a foundation, there is no reconcilability of data



# Multidimensional Spreadsheets



- ⌘ Analysts need spreadsheets that support
  - ☑ pivot tables (cross-tabs)
  - ☑ drill-down and roll-up
  - ☑ slice and dice
  - ☑ sort
  - ☑ selections
  - ☑ derived attributes
- ⌘ Popular in retail domain



## OLAP - Data Cube

⌘ Idea: analysts need to group data in many different ways

⌘ eg. Sales(region, product, prodtype, prodstyle, date, saleamount)

⌘ saleamount is a measure attribute, rest are dimension attributes

⌘ groupby every subset of the other attributes

⌘ materialize (precompute and store)  
groupbys to give online response

⌘ Also: hierarchies on attributes: date -> weekday,  
date -> month -> quarter -> year

# SQL Extensions

## ⌘ Front-end tools require

### ☑ Extended Family of Aggregate Functions

- ☑ rank, median, mode

### ☑ Reporting Features

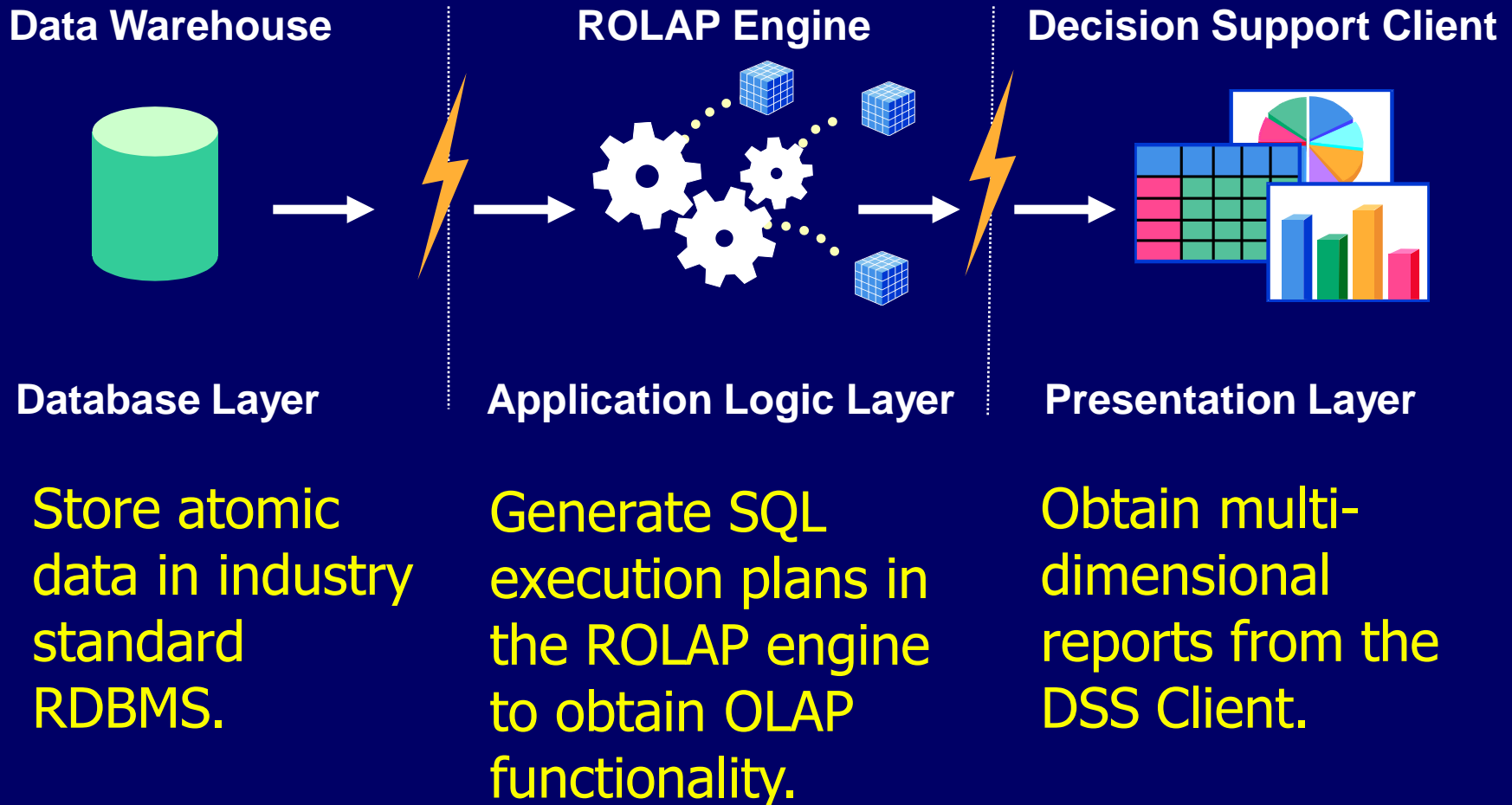
- ☑ running totals, cumulative totals

### ☑ Results of multiple group by

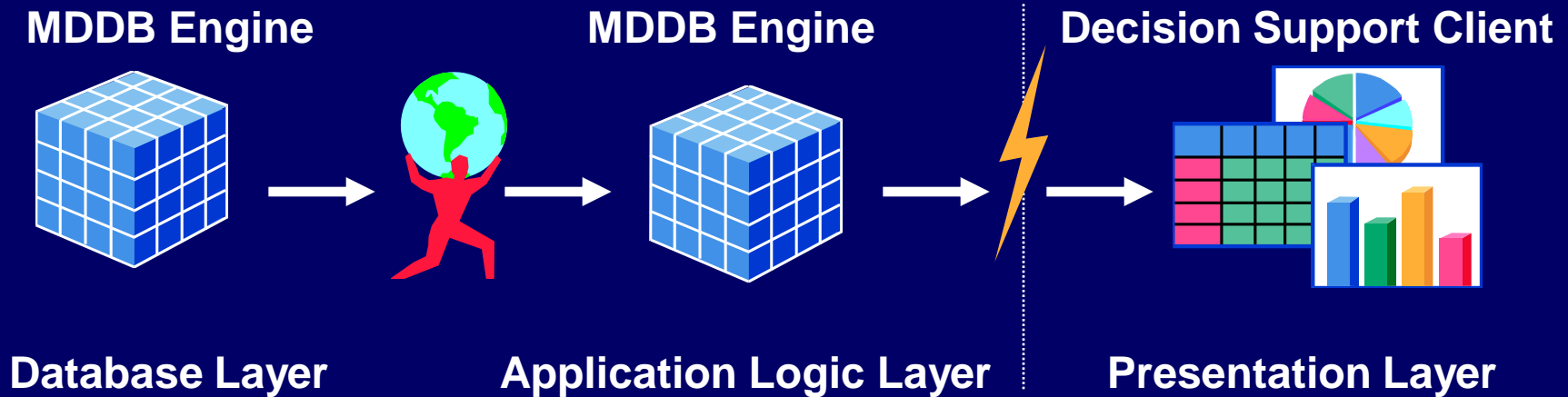
- ☑ total sales by month and total sales by product

### ☑ Data Cube

# Relational OLAP: 3 Tier DSS



# MD-OLAP: 2 Tier DSS

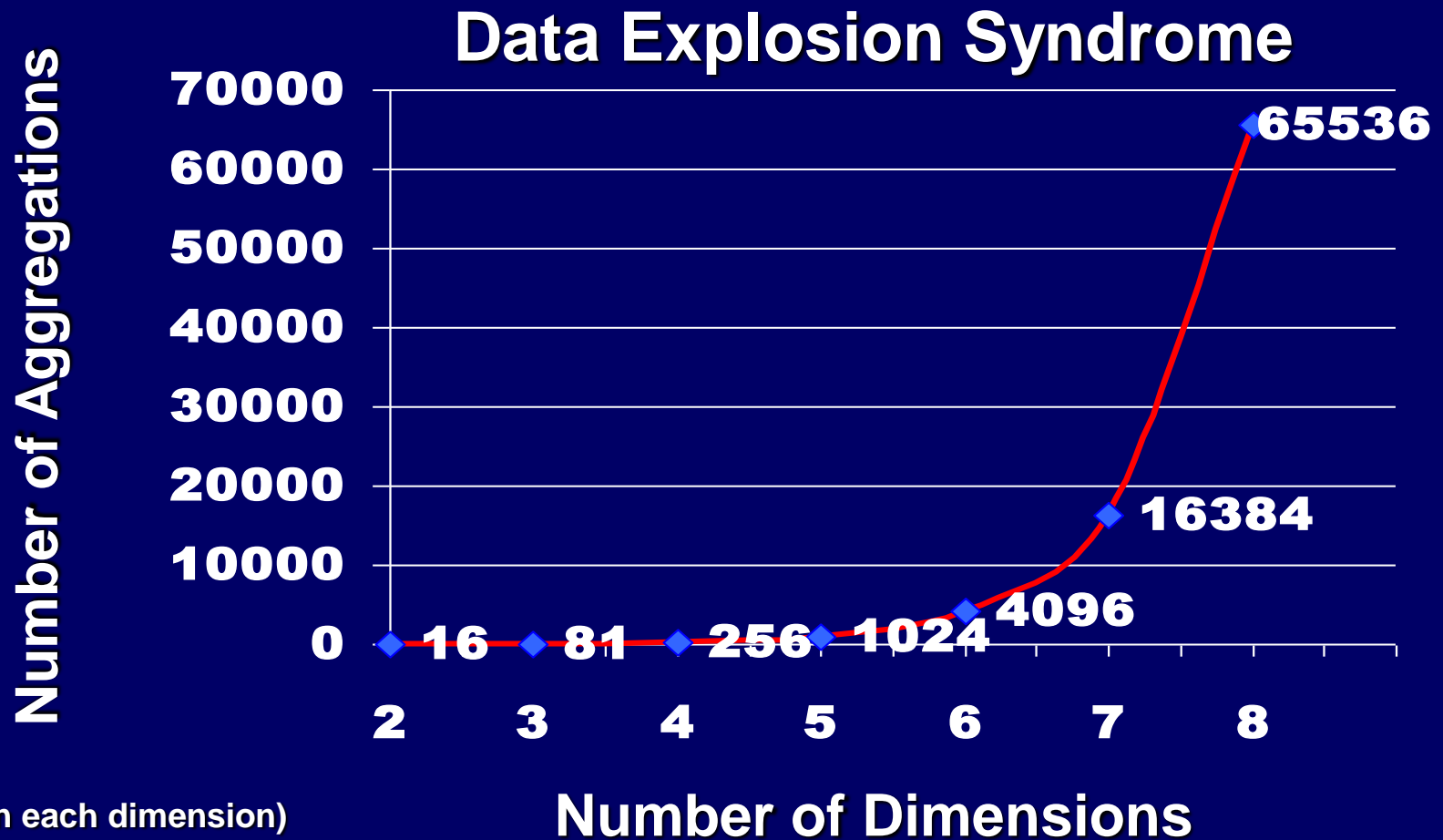


Store atomic data in a proprietary data structure (MDDDB), pre-calculate as many outcomes as possible, obtain OLAP functionality via proprietary algorithms running against this data.

Obtain multi-dimensional reports from the DSS Client.

# Typical OLAP Problems

## Data Explosion



# Metadata Repository

## ⌘ Administrative metadata

- ☒ source databases and their contents
- ☒ gateway descriptions
- ☒ warehouse schema, view & derived data definitions
- ☒ dimensions, hierarchies
- ☒ pre-defined queries and reports
- ☒ data mart locations and contents
- ☒ data partitions
- ☒ data extraction, cleansing, transformation rules, defaults
- ☒ data refresh and purging rules
- ☒ user profiles, user groups
- ☒ security: user authorization, access control

# Metadata Repository .. 2

## ⌘ Business data

- ☑ business terms and definitions
- ☑ ownership of data
- ☑ charging policies

## ⌘ operational metadata

- ☑ data lineage: history of migrated data and sequence of transformations applied
- ☑ currency of data: active, archived, purged
- ☑ monitoring information: warehouse usage statistics, error reports, audit trails.

# Recipe for a Successful Warehouse





# For a Successful Warehouse

From Larry Greenfield, <http://pwp.starnetinc.com/larryg/index.html>

- ⌘ From day one establish that warehousing is a joint user/builder project
- ⌘ Establish that maintaining data quality will be an *ONGOING* joint user/builder responsibility
- ⌘ Train the users one step at a time
- ⌘ Consider doing a high level corporate data model in no more than three weeks

# For a Successful Warehouse

- ⌘ Look closely at the data extracting, cleaning, and loading tools
- ⌘ Implement a user accessible automated directory to information stored in the warehouse
- ⌘ Determine a plan to test the integrity of the data in the warehouse
- ⌘ From the start get warehouse users in the habit of 'testing' complex queries

# For a Successful Warehouse

- ⌘ Coordinate system roll-out with network administration personnel
- ⌘ When in a bind, ask others who have done the same thing for advice
- ⌘ Be on the lookout for small, but strategic, projects
- ⌘ Market and sell your data warehousing systems

# Data Warehouse Pitfalls

- ⌘ You are going to spend much time extracting, cleaning, and loading data
- ⌘ Despite best efforts at project management, data warehousing project scope will increase
- ⌘ You are going to find problems with systems feeding the data warehouse
- ⌘ You will find the need to store data not being captured by any existing system
- ⌘ You will need to validate data not being validated by transaction processing systems

# Data Warehouse Pitfalls

- ⌘ Some transaction processing systems feeding the warehousing system will not contain detail
- ⌘ Many warehouse end users will be trained and never or seldom apply their training
- ⌘ After end users receive query and report tools, requests for IS written reports may increase
- ⌘ Your warehouse users will develop conflicting business rules
- ⌘ Large scale data warehousing can become an exercise in data homogenizing

# Data Warehouse Pitfalls

- ⌘ 'Overhead' can eat up great amounts of disk space
- ⌘ The time it takes to load the warehouse will expand to the amount of the time in the available window... and then some
- ⌘ Assigning security cannot be done with a transaction processing system mindset
- ⌘ You are building a HIGH maintenance system
- ⌘ You will fail if you concentrate on resource optimization to the neglect of project, data, and customer management issues and an understanding of what adds value to the customer

# DW and OLAP Research Issues

## ⌘ Data cleaning

- ☒ focus on data inconsistencies, not schema differences
- ☒ data mining techniques

## ⌘ Physical Design

- ☒ design of summary tables, partitions, indexes
- ☒ tradeoffs in use of different indexes

## ⌘ Query processing

- ☒ selecting appropriate summary tables
- ☒ dynamic optimization with feedback
- ☒ acid test for query optimization: cost estimation, use of transformations, search strategies
- ☒ partitioning query processing between OLAP server and backend server.

# DW and OLAP Research Issues .. 2

## ⌘ Warehouse Management

- ☒ detecting runaway queries
- ☒ resource management
- ☒ incremental refresh techniques
- ☒ computing summary tables during load
- ☒ failure recovery during load and refresh
- ☒ process management: scheduling queries, load and refresh
- ☒ Query processing, caching
- ☒ use of workflow technology for process management



# Products, References, Useful Links



# Reporting Tools

- ⌘ Andyne Computing -- GQL
- ⌘ Brio -- BrioQuery
- ⌘ Business Objects -- Business Objects
- ⌘ Cognos -- Impromptu
- ⌘ Information Builders Inc. -- Focus for Windows
- ⌘ Oracle -- Discoverer2000
- ⌘ Platinum Technology -- SQL\*Assist, ProReports
- ⌘ PowerSoft -- InfoMaker
- ⌘ SAS Institute -- SAS/Assist
- ⌘ Software AG -- Esperant
- ⌘ Sterling Software -- VISION:Data

# OLAP and Executive Information Systems

- ⌘ Andyne Computing -- Pablo
- ⌘ Arbor Software -- Essbase
- ⌘ Cognos -- PowerPlay
- ⌘ Comshare -- Commander OLAP
- ⌘ Holistic Systems -- Holos
- ⌘ Information Advantage -- AXSYS, WebOLAP
- ⌘ Informix -- Metacube
- ⌘ Microstrategies -- DSS/Agent
- ⌘ Microsoft -- Plato
- ⌘ Oracle -- Express
- ⌘ Pilot -- LightShip
- ⌘ Planning Sciences -- Gentium
- ⌘ Platinum Technology -- ProdeaBeacon, Forest & Trees
- ⌘ SAS Institute -- SAS/EIS, OLAP++
- ⌘ Speedware -- Media

# Other Warehouse Related Products

⌘ Data extract, clean, transform, refresh

- ☒ CA-Ingres replicator

- ☒ Carleton Passport

- ☒ Prism Warehouse Manager

- ☒ SAS Access

- ☒ Sybase Replication Server

- ☒ Platinum Inforefiner, Infopump

# Extraction and Transformation Tools

- ⌘ Carleton Corporation -- Passport
- ⌘ Evolutionary Technologies Inc. -- Extract
- ⌘ Informatica -- OpenBridge
- ⌘ Information Builders Inc. -- EDA Copy Manager
- ⌘ Platinum Technology -- InfoRefiner
- ⌘ Prism Solutions -- Prism Warehouse Manager
- ⌘ Red Brick Systems -- DecisionScape Formation

# Scrubbing Tools

⌘ Apertus -- Enterprise/Integrator

⌘ Vality -- IPE

⌘ Postal Soft

# Warehouse Products

- ⌘ Computer Associates -- CA-Ingres
- ⌘ Hewlett-Packard -- Allbase/SQL
- ⌘ Informix -- Informix, Informix XPS
- ⌘ Microsoft -- SQL Server
- ⌘ Oracle -- Oracle7, Oracle Parallel Server
- ⌘ Red Brick -- Red Brick Warehouse
- ⌘ SAS Institute -- SAS
- ⌘ Software AG -- ADABAS
- ⌘ Sybase -- SQL Server, IQ, MPP

# Warehouse Server Products

⌘ Oracle 8

⌘ Informix

- ☑ Online Dynamic Server

- ☑ XPS --Extended Parallel Server

- ☑ Universal Server for object relational applications

⌘ Sybase

- ☑ Adaptive Server 11.5

- ☑ Sybase MPP

- ☑ Sybase IQ



# Warehouse Server Products

- ⌘ Red Brick Warehouse

- ⌘ Tandem Nonstop

- ⌘ IBM

  - ☒ DB2 MVS

  - ☒ Universal Server

  - ☒ DB2 400

- ⌘ Teradata

# Other Warehouse Related Products

## ⌘ Connectivity to Sources

- ☑ Apertus
- ☑ Information Builders EDA/SQL
- ☑ Platinum Infohub
- ☑ SAS Connect
- ☑ IBM Data Joiner
- ☑ Oracle Open Connect
- ☑ Informix Express Gateway

# Other Warehouse Related Products

## ⌘ Query/Reporting Environments

- ☑ Brio/Query
- ☑ Cognos Impromptu
- ☑ Informix Viewpoint
- ☑ CA Visual Express
- ☑ Business Objects
- ☑ Platinum Forest and Trees

# 4GL's, GUI Builders, and PC Databases

- ⌘ Information Builders -- Focus
- ⌘ Lotus -- Approach
- ⌘ Microsoft -- Access, Visual Basic
- ⌘ MITI -- SQR/Workbench
- ⌘ PowerSoft -- PowerBuilder
- ⌘ SAS Institute -- SAS/AF

# Data Mining Products

⌘ DataMind -- neurOagent

⌘ Information Discovery -- IDIS

⌘ SAS Institute -- SAS/Neuronets

# Data Warehouse

- ⌘ W.H. Inmon, Building the Data Warehouse, Second Edition, John Wiley and Sons, 1996
- ⌘ W.H. Inmon, J. D. Welch, Katherine L. Glassey, Managing the Data Warehouse, John Wiley and Sons, 1997
- ⌘ Barry Devlin, Data Warehouse from Architecture to Implementation, Addison Wesley Longman, Inc 1997

# Data Warehouse

- ⌘ W.H. Inmon, John A. Zachman, Jonathan G. Geiger, Data Stores Data Warehousing and the Zachman Framework, McGraw Hill Series on Data Warehousing and Data Management, 1997
- ⌘ Ralph Kimball, The Data Warehouse Toolkit, John Wiley and Sons, 1996

# OLAP and DSS

- ⌘ Erik Thomsen, OLAP Solutions, John Wiley and Sons 1997
- ⌘ Microsoft TechEd Transparencies from Microsoft TechEd 98
- ⌘ Essbase Product Literature
- ⌘ Oracle Express Product Literature
- ⌘ Microsoft Plato Web Site
- ⌘ Microstrategy Web Site



# Data Mining

- ⌘ Michael J.A. Berry and Gordon Linoff, Data Mining Techniques, John Wiley and Sons 1997
- ⌘ Peter Adriaans and Dolf Zantinge, Data Mining, Addison Wesley Longman Ltd. 1996
- ⌘ KDD Conferences

# Other Tutorials

- ⌘ Donovan Schneider, Data Warehousing Tutorial, Tutorial at International Conference for Management of Data (SIGMOD 1996) and International Conference on Very Large Data Bases 97
- ⌘ Umeshwar Dayal and Surajit Chaudhuri, Data Warehousing Tutorial at International Conference on Very Large Data Bases 1996
- ⌘ Anand Deshpande and S. Seshadri, Tutorial on Datawarehousing and Data Mining, CSI-97

# Useful URLs

⌘ Ralph Kimball's home page

⌘ <http://www.rkimball.com>

⌘ Larry Greenfield's Data Warehouse Information Center

⌘ <http://pwp.starnetinc.com/larryg/>

⌘ Data Warehousing Institute

⌘ <http://www.dw-institute.com/>

⌘ OLAP Council

⌘ <http://www.olapcouncil.com/>

1.	The hardware architecture of a data warehouse is defined within the _____ stage of process	Technical blue print	business requirements	Technical blue print	system requirements	Technical blue print
2.	The _____ stage should have identified the initial user requirements and given an indication of the capacity planning requirements.	business requirements	business requirements	Technical blue print	system requirements	business requirements
3.	The backup and security strategies are also determined during the _____ phase.	business requirements	software requirements	Technical blue print	system requirements	Technical blue print
4.	A _____ is a set of loosely coupled SMP machines connected by a high speed interconnect.	Stratified	Record	Parallel index build.	Cluster	Cluster
5.	Each machine has its own CPUs and memory, but they share access to disk is known as _____	Shared -disk system	set data system	shared data system	Cluster	Shared -disk system
6.	Each machine in the cluster is called a _____	Event	Node	Record data management system	Record	Node
7.	A layer of software called the _____	Disk Lock Manager	Device Lock Manager	distributed lock manager	Data Lock Manager	distributed lock manager
8.	NUMA stands for _____	Not uniform memory architecture	None Uniform memory architecture	None unified memory architecture	Non uniform memory architecture	Non uniform memory architecture
9.	A _____ machine is basically a tightly coupled cluster of SMP nodes, with an extremely high speed interconnect.	NUMA	MUMA	NAMA	NUMR	NUMA
10	Network management is a black _____	lake	science	Art	math	Art
11	The heart of any computer is _____	CPU	CUP	PCU	PPU	CPU
12	I/O stands for _____	Output and input	Input and Output	Data record	internal and outer	Input and Output

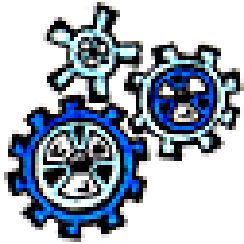
13	RDBMS stands for_____	Relational Database Management System	Rational data management system	Record data management system	Relation data management system	Relational Database Management System
14	_____is where a process requests for the data to be shipped to the location where the process is running.	Pipelining	Intra Statement	Function shipping	Data shipping	Data shipping
15	_____is where the function to be performed is moved to the locale of the data.	Pipelining	Intra Statement	Function shipping	Data shipping	Function shipping
16	_____ Operations are separate operations that occur within the confines of the SQL statement itself	Pipelining	Intra Statement	Function shipping	Data shipping	Intra Statement
17	_____is where operations are carried out sequentially on a given piece of data.	Pipelining	Intra Statement	Function shipping	Data shipping	Pipelining
18	_____is the number of parallelism is the number of processes that will be used to simultaneously perform any single operation.	Degree of parallelism	degree of parallel	pipe lining	parallel language	Degree of parallelism
19	DML Stands for_____	Data manipulator language	Data Manipulation Language	Parallel index build.	Data shipping	Data Manipulation Language
20	_____ is an extension of the parallel query functionality.	Degree of parallelism	Intra Statement	Parallel index build.	Data shipping	Parallel index build.
21	GUI stands for _____	Graphical User Interface	Graph user interface	Graphical unit interface	Graph under interfere	Graphical User Interface
22	_____by nature is an open, accessible system.	Data mining	A Data warehouse	Data mart	Data disk	A Data warehouse
23	_____ can also be classified by job function.	Mart	subset	Data	Auditing	Data
24	The_____ can be separated from the data warehouse.	subset	data event	data load	Data marts	Data marts

25	_____ is a specific subset of security that is often mandated by organizations.	Testing	Node	Auditing	event	Auditing
26	TCSEC_____	Trust computing system evaluation criteria	Trusted Computing System Evaluation Criteria	Treat Computer system evaluation criteria	Three computing system evaluation criteria	Trusted Computing System Evaluation Criteria
27	_____ is any method whereby restricted information is inadvertently given away by implication rather than by design.	Covert channel	pivot channel	fact channel	partitioning channel	Covert channel
28	BI stands for_____	data warehouse	business Intelligence	data mining	operation data	business Intelligence
29	_____ is important to get all the security and audit requirements clearly documented as this will be needed as part of any cost justification.	printing	data purging	Documentation	data record	Documentation
30	The_____ of the design of a data warehouse is a complex and lengthy process.	disk data	Node	event	Testing	Testing
31	_____ is one of the most important regular operations carried out on any system.	online backup	Cold backup	Complete backup	Backup	Backup
32	In _____, the entire database is backed up at the same time.	partial backup	Cold backup	Complete backup	hot backup	Complete backup
33	_____ is any backup that is not complete.	cold backup	Partial backup	Complete backup	hot backup	Partial backup
34	A _____ is a backup that is taken while the database is completely shut down.	Hot backup	Cold backup	Complete backup	online backup	Cold backup
35	_____ is any backup that is not cold is considered to be hot.	Hot backup	Cold backup	Complete backup	online backup	Hot backup
36	_____ is a synonym for hot backup.	Hot backup	Online backup	Complete backup	partial backup	Online backup

37	_____ are a method of loading multiple tapes into a single tape drive.	Tape silos	Record	Tape stackers	Stackers	Tape stackers
38	_____ are large tape storage facilities, which can store and mange thousands of tapes.	Tape	Record	processing	Tape silos	Tape silos
39	In _____ the backup is performed to disk rather than to tape.	Disk to disk backup	backup to backup	system to system backup	data backup	Disk to disk backup
40	_____ Breaking is really a variant of disk to disk backup.	mirror	Mirror breaking	breaking	value store	Mirror breaking
41	_____ are devices that allow data to be stored near-line.	juke box	optical	tape silo	Optical jukeboxes	Optical jukeboxes
42	_____ WORM Stands for _____	Write one read many	wrote one read many	tape silo	value store	Write one read many
43	_____ Technology allows large numbers of optical disks to be managed in much the same way as a tape stacker or silo.	Tape	Jukebox	silos	Tape Silos	Jukebox
44	_____ VSD Stands for _____	visual share disk	visual share data	Virtual shared Disk	visual store disk	Virtual shared Disk
45	_____ is a single statement that requests a certain subset of data ordered or presented in a particular way.	Query	load	system	software	Query
46	_____ RAID Stands for _____	array disk	disk load	redundancy	Redundant array of inexpensive disk	Redundant array of inexpensive disk
47	Data can also be classified by _____ function	Job	tape	tape silos	tape structure	Job
48	_____ are expensive backup solutions, and it is often difficult to justify such expense for a single application.	Job	Silos	tape silos	tape structure	Silos
49	In _____, the backup will normally be backed up to tape later.	disk backup	data backup	Disk to disk backup	backup to backup	Disk to disk backup

50	OmnibackII Package Produced by_____	HP	IBM	Sequent	Epoch System	HP
51	ADSM Package produced by _____	HP	IBM	Sequent	Epoch System	IBM
52	Alexandria Package Produced by_____	legato	IBM	Sequent	Epoch System	Sequent
53	Epoch Package Produced by_____	legato	IBM	Sequent	Epoch System	Epoch System
54	Networker Software package Produced by_____	IBM	sequent	Legato	Epoch System	Legato
55	_____ is a system that has responsibilities to the operations of the business.	operational System	Mission Critical System	7 x 24 x 52 System	7 x 24 System	operational System
56	_____ is a system that the business absolutely depends on to function.	operational System	Mission Critical System	7 x 24 x 52 System	7 x 24 System	Mission Critical System
57	_____ is a system that needs to be available all day every day, except for small periods of planned downtime.	operational System	Mission Critical System	7 x 24 x 52 System	7 x 24 System	7 x 24 System
58	_____ is a true 7 x 24 System, which is required to be running all the time	operational System	Mission Critical System	7 x 24 x 52 System	7 x 24 System	7 x 24 x 52 System
59	_____ are the elements that directly affect the users, such as hours of access and response times.	business requirements	User requirements	system requirements	software requirements	User requirements
60	_____ are the needs imposed on the system by the business, such as system availability.	System requirements	User requirements	business requirements	software requirements	System requirements





# Data Mining: Concepts & Techniques

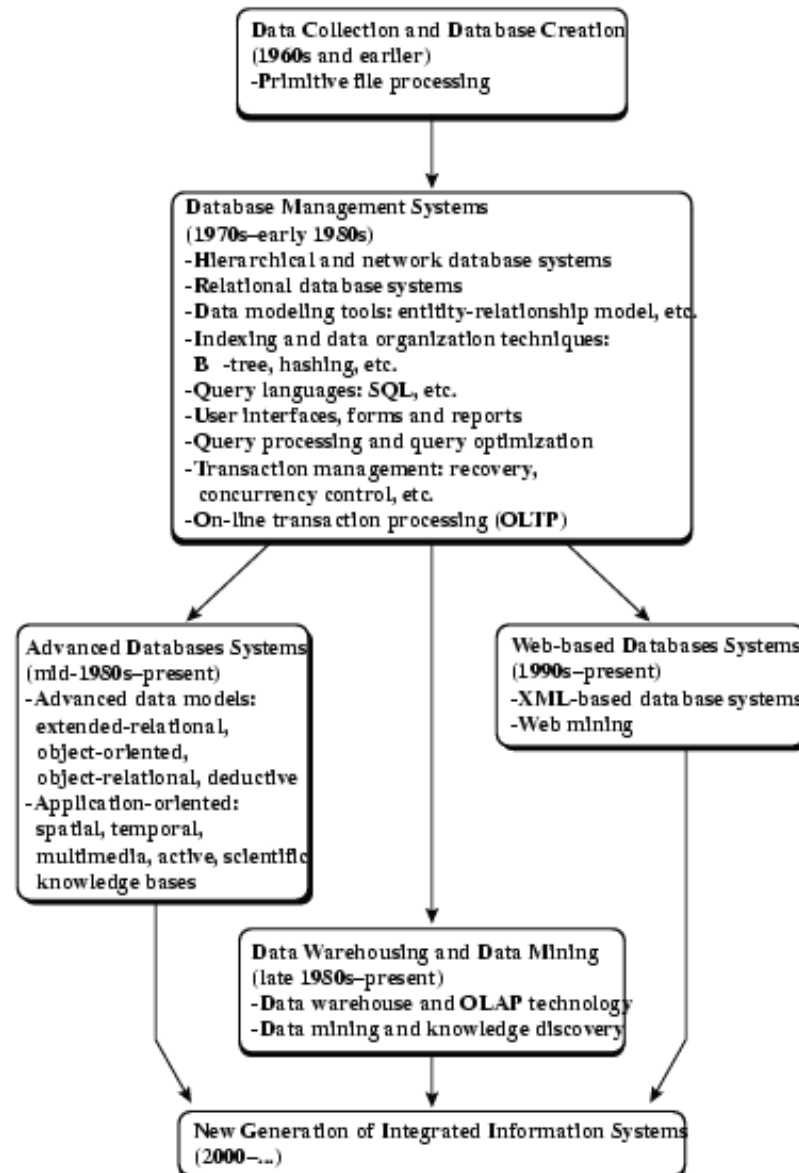


# Motivation:

## Necessity is the Mother of Invention

- Data explosion problem
  - Automated data collection tools and mature database technology lead to tremendous amounts of data stored in databases, data warehouses and other information repositories
- We are drowning in data, but starving for knowledge!
- Solution: Data warehousing and data mining
  - Data warehousing and on-line analytical processing
  - Extraction of interesting knowledge (rules, regularities, patterns, constraints) from data in large databases

# Evolution of Database Technology



**How can I analyze this data?**





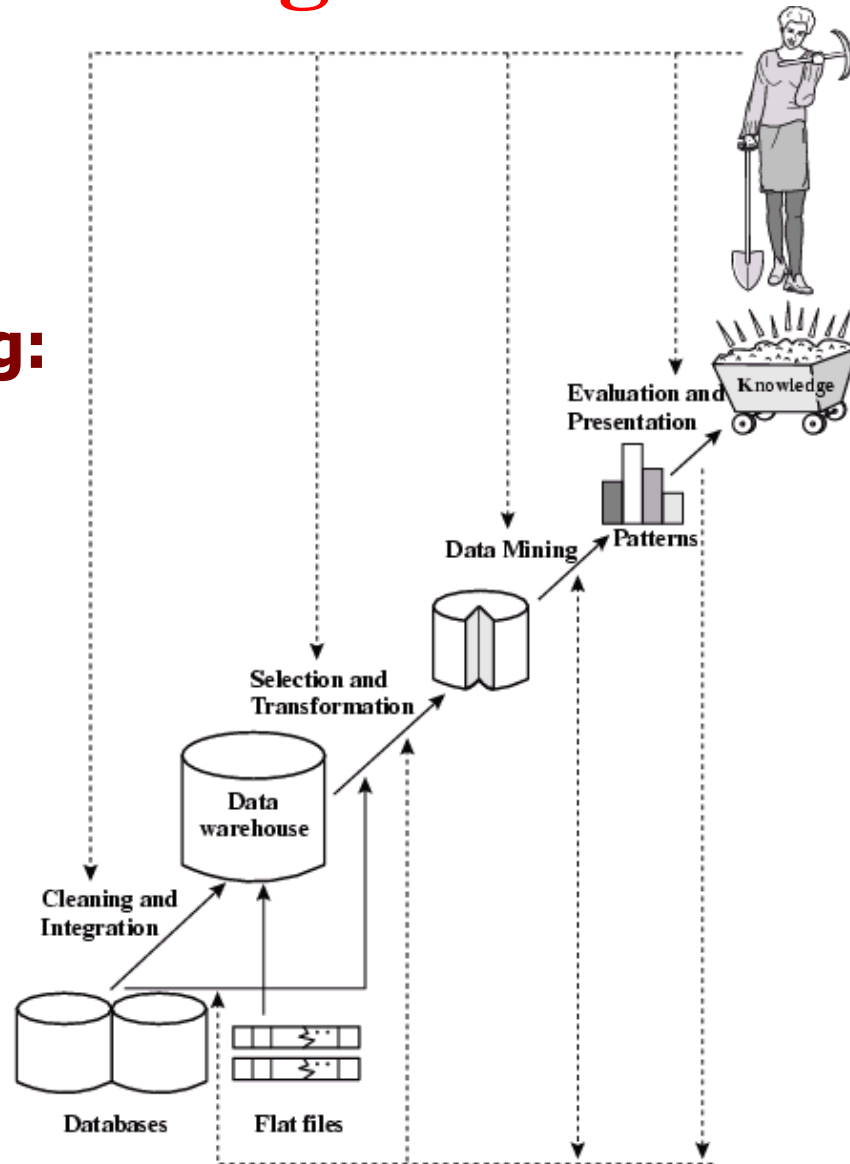
# What Is Data Mining?



- Data mining (knowledge discovery in databases):
  - Extraction of interesting (**non-trivial, implicit, previously unknown and potentially useful**) information or patterns from data in **large databases**
- Alternative names and their “inside stories”:
  - Data mining: a misnomer?
  - Knowledge discovery(mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.
- What is not data mining?
  - (Deductive) query processing.
  - Expert systems or small ML/statistical programs

# Data Mining: A KDD Process

**Data mining:  
the core of  
knowledge  
discovery  
process**



# Steps of a KDD Process

- Learning the application domain:
  - relevant prior knowledge and goals of application
- Creating a target data set: data selection
- Data cleaning and preprocessing: (may take 60% of effort!)
- Data reduction and transformation:
  - Find useful features, dimensionality/variable reduction, invariant representation.
- Choosing functions of data mining
  - summarization, classification, regression, association, clustering.
- Choosing the mining algorithm(s)
- Data mining: search for patterns of interest
- Pattern evaluation and knowledge presentation
  - visualization, transformation, removing redundant patterns, etc.
- Use of discovered knowledge



# Knowledge Discovery Process

- The whole process of extraction of implicit, previously unknown and potentially useful knowledge from a large database
  - It includes **data selection, cleaning, enrichment, coding, data mining, and reporting**
  - Data Mining is the key stage of Knowledge Discovery Process
    - The process of finding the desired information from large database

# Knowledge Discovery Process

- **Example: the database of a magazine publisher which sells five types of magazines – on cars, houses, sports, music and comics**
  - Data mining:
    - Find interesting categorical properties
  - Questions:
    - What is the profile of a reader of a car magazine?
    - Is there any correlation between an interest in cars and an interest in comics?
- The knowledge discovery process consists of six stages

# Data Selection

- Select the information about people who have subscribed to a magazine

Client number	Name	Address	Date purchase made	Magazine purchased
23003	Johnson	1 Downing Street	04-15-94	car
23003	Johnson	1 Downing Street	06-21-93	music
23003	Johnson	1 Downing Street	05-30-92	comic
23009	Clinton	2 Boulevard	01-01-01	comic
23013	King	3 High Road	02-30-95	sports
23019	<b>Jonson</b>	1 Downing Street	01-01-01	house

# Cleaning

- Pollutions: **Type errors**, moving from one place to another without notifying change of address, people give incorrect information about themselves
  - Pattern Recognition Algorithms

Client number	Name	Address	Date purchase made	Magazine purchased
23003	Johnson	1 Downing Street	04-15-94	car
23003	Johnson	1 Downing Street	06-21-93	music
23003	Johnson	1 Downing Street	05-30-92	comic
23009	Clinton	2 Boulevard	01-01-01	comic
23013	King	3 High Road	02-30-95	sports
23003	Johnson	1 Downing Street	01-01-01	house

# Cleaning

- Lack of **domain consistency**

Client number	Name	Address	Date purchase made	Magazine purchased
23003	Johnson	1 Downing Street	04-15-94	car
23003	Johnson	1 Downing Street	06-21-93	music
23003	Johnson	1 Downing Street	05-30-92	comic
23009	Clinton	2 Boulevard	NULL	comic
23013	King	3 High Road	02-30-95	sports
23003	Johnson	1 Downing Street	12-20-94	house

# Enrichment

- Need extra information about the clients consisting of date of birth, income, amount of credit, and whether or not an individual owns a car or a house

Client name	Date of birth	Income	Credit	Car owner	House owner
Johnson	04-13-76	\$18,500	\$17,800	no	no
Clinton	10-20-71	\$36,000	\$26,600	yes	no

# Enrichment

- The new information need to be easily joined to the existing client records
  - Extract more knowledge

Client number	Name	Date of birth	Income	Credit	Car owner	House owner	Address	Date purchase made	Magazine purchased
23003	Johnson	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	04-15-94	car
23003	Johnson	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	06-21-93	music
23003	Johnson	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	05-30-92	comic
23009	Clinton	10-20-71	\$36,000	\$26,600	yes	no	2 Boulevard	NULL	comic
23013	King	NULL	NULL	NULL	NULL	NULL	3 High Road	02-30-95	sports
23003	Johnson	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	12-20-94	house

# Coding

- We select only those records that have enough information to be of value (row)
- Project the fields in which we are interested (column)

Client number	Date of birth	Income	Credit	Car owner	House owner	Address	Date purchase made	Magazine purchased
23003	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	04-15-94	car
23003	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	06-21-93	music
23003	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	05-30-92	comic
23009	10-20-71	\$36,000	\$26,600	yes	no	2 Boulevard	NULL	comic
23003	04-13-76	\$18,500	\$17,800	no	no	1 Downing Street	12-20-94	house



# Coding

- Code the information which is too detailed
  - Address to region
  - Birth date to age
  - Divide income by 1000
  - Divide credit by 1000
  - Convert cars yes-no to 1-0
  - Convert purchase date to month numbers starting from 1990
    - The way in which we code the information will determine the type of patterns we find
    - Coding has to be performed repeatedly in order to get the best results

# Coding

- The way in which we code the information will determine the type of patterns we find

Client number	Age	Income	Credit	Car owner	House owner	Region	Month of purchase	Magazine purchased
23003	20	18.5	17.8	0	0	1	52	car
23003	20	18.5	17.8	0	0	1	42	music
23003	20	18.5	17.8	0	0	1	29	comic
23009	25	36.0	26.6	1	0	1	NULL	comic
23003	20	18.5	17.8	0	0	1	48	house

# Coding

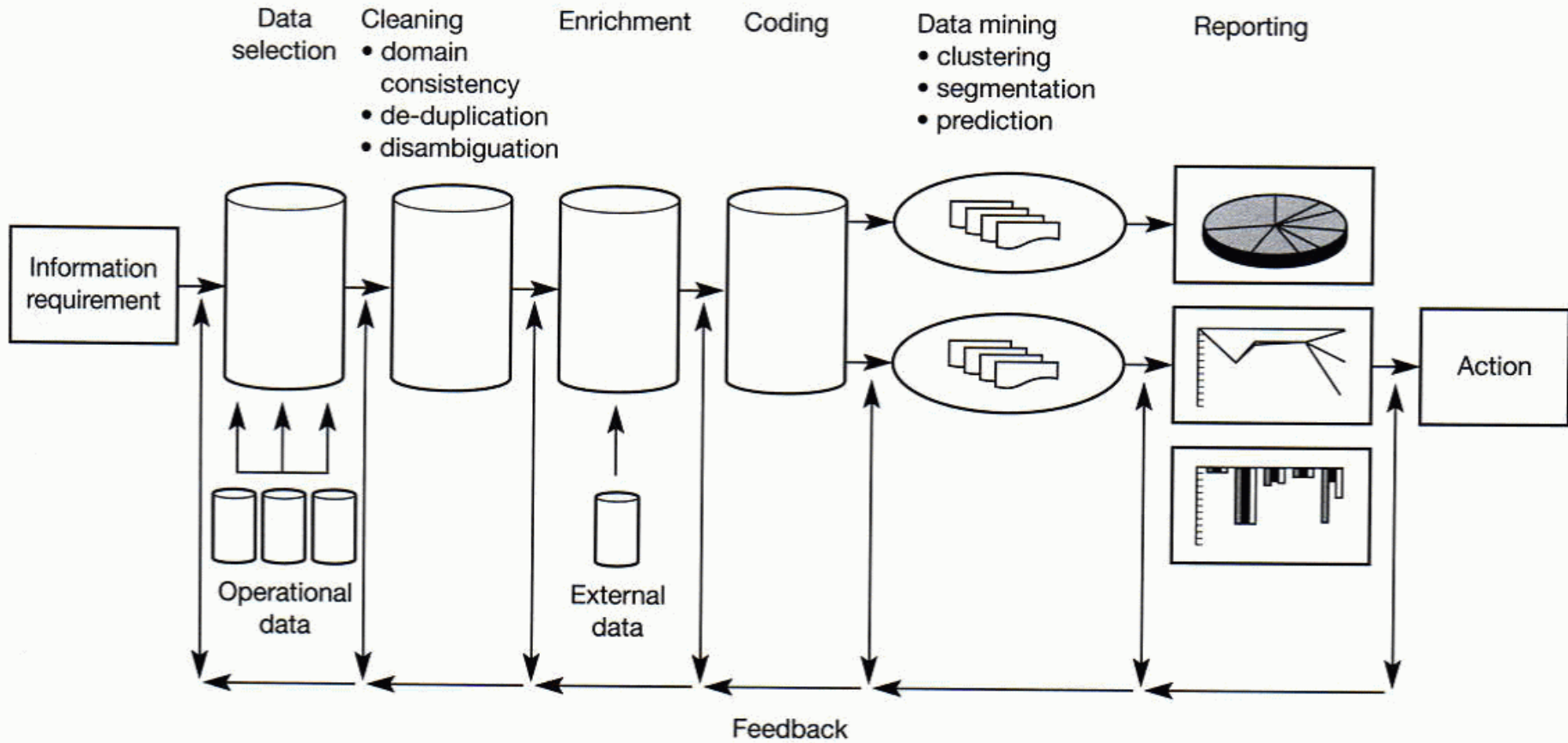
- We are interested in the relationships between readers of different magazines
  - Perform **flattening** operation

Client number	Age	Income	Credit	Car owner	House owner	Region	Magazines purchased				
							car magazine	house magazine	sports magazine	music magazine	comic magazine
23003	20	18.5	17.8	0	0	1	1	1	0	1	1
23009	25	36.0	26.6	1	0	1	0	0	0	0	1

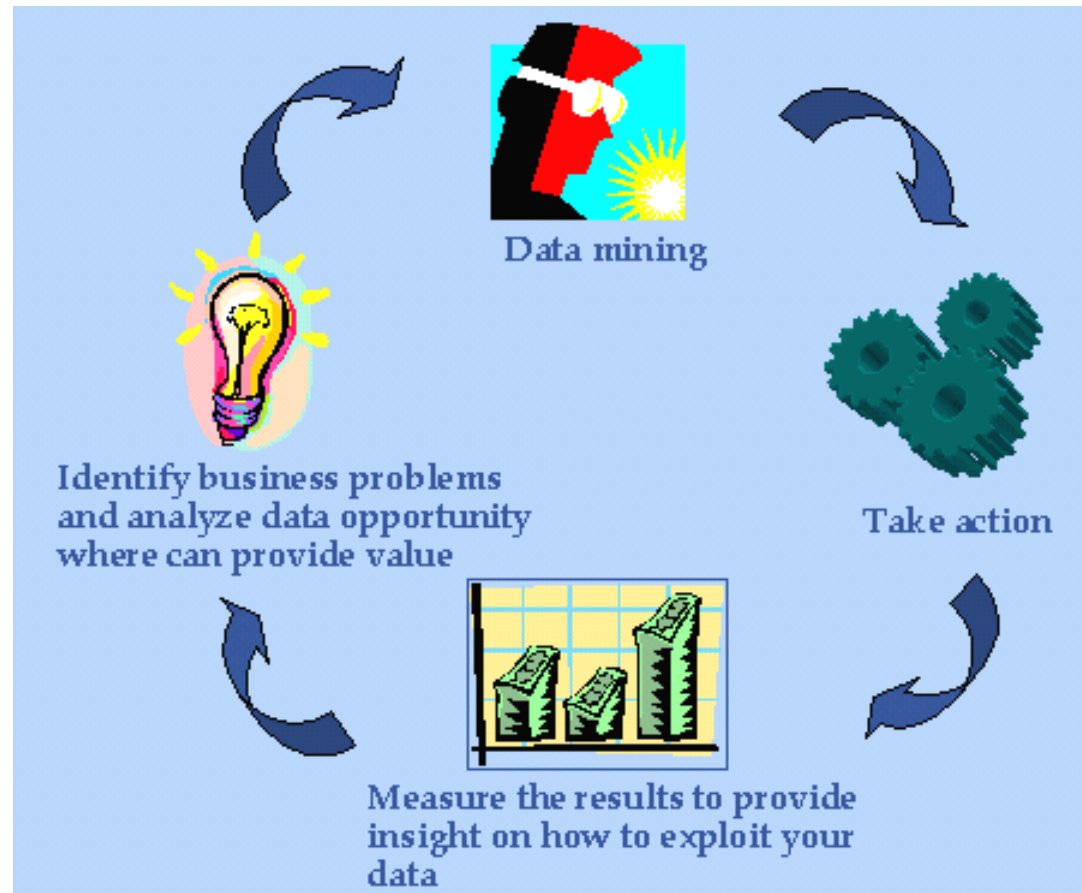
# Data mining

- We may find the following rules
  - A customer with credit  $> 13000$  and aged between 22 and 31 who has subscribed to a comics at time T will very likely subscribe to a car magazine five years later
  - The number of house magazines sold to customers with credit between 12000 and 31000 living in region 4 is increasing
  - A customer with credit between 5000 and 10000 who reads a comics magazine will very likely become a customer with credit between 12000 and 31000 who reads a sports and a house magazine after 12 years

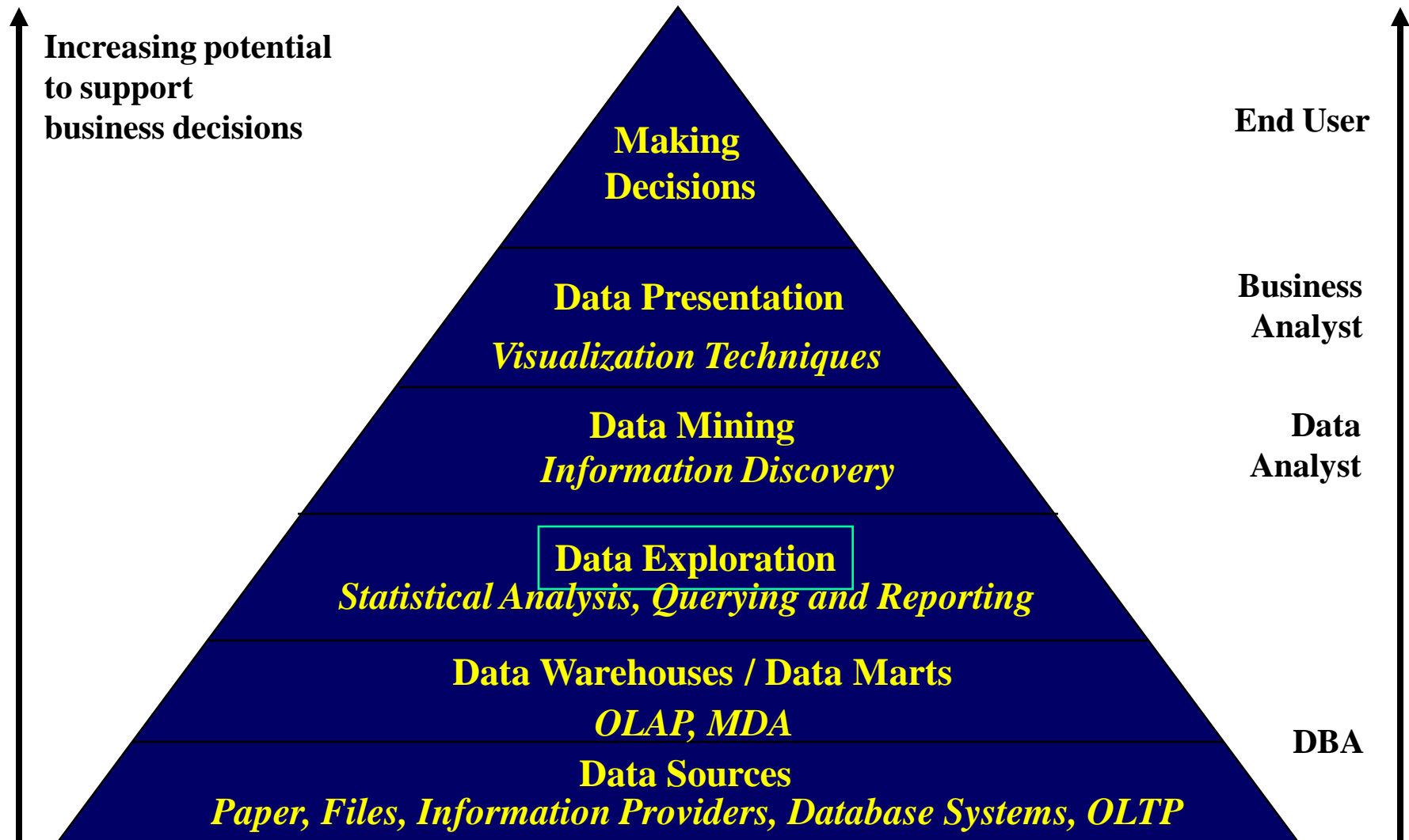
# Knowledge Discovery Process



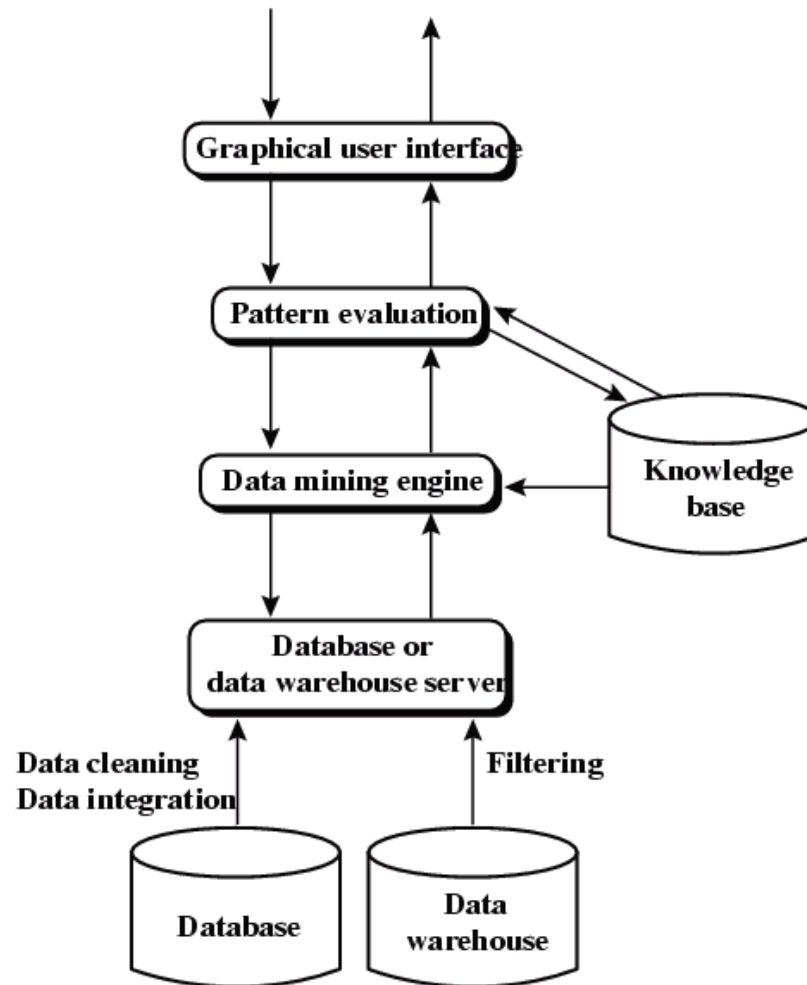
# Business-Question-Driven Process



# Data Mining and Business Intelligence



# Architecture of a Typical Data Mining System





# Data Mining: On What Kind of Data?

- Relational databases
- Data warehouses
- Transactional databases
- Advanced DB and information repositories
  - Object-oriented and object-relational databases
  - Spatial databases
  - Time-series data and temporal data
  - Text databases and multimedia databases
  - Heterogeneous databases
  - WWW

*customer*

cust_ID	name	address	age	income	credit_info	...
C1	Smith, Sandy	5463 E Hastings, Burnaby,	21	\$27000	1	...
...	...	BC V5A 4S9, Canada	...	...	...	...
...	...	...	...	...	...	...

*item*

item_ID	name	brand	category	type	price	place_made	supplier	cost
I3	high-res-TV	Toshiba	high resolution	TV	\$988.00	Japan	NikoX	\$600.00
I8	multidisc-CDplay	Sanyo	multidisc	CD player	\$369.00	Japan	MusicFront	\$120.00
...	...	...	...	...	...	...	...	...

*employee*

empl_ID	name	category	group	salary	commission
E55	Jones, Jane	home entertainment	manager	\$18,000	2%
...	...	...	...	...	...

*branch*

branch_ID	name	address
B1	City Square	369 Cambie St., Vancouver, BC V5L 3A2, Canada
...	...	...

*purchases*

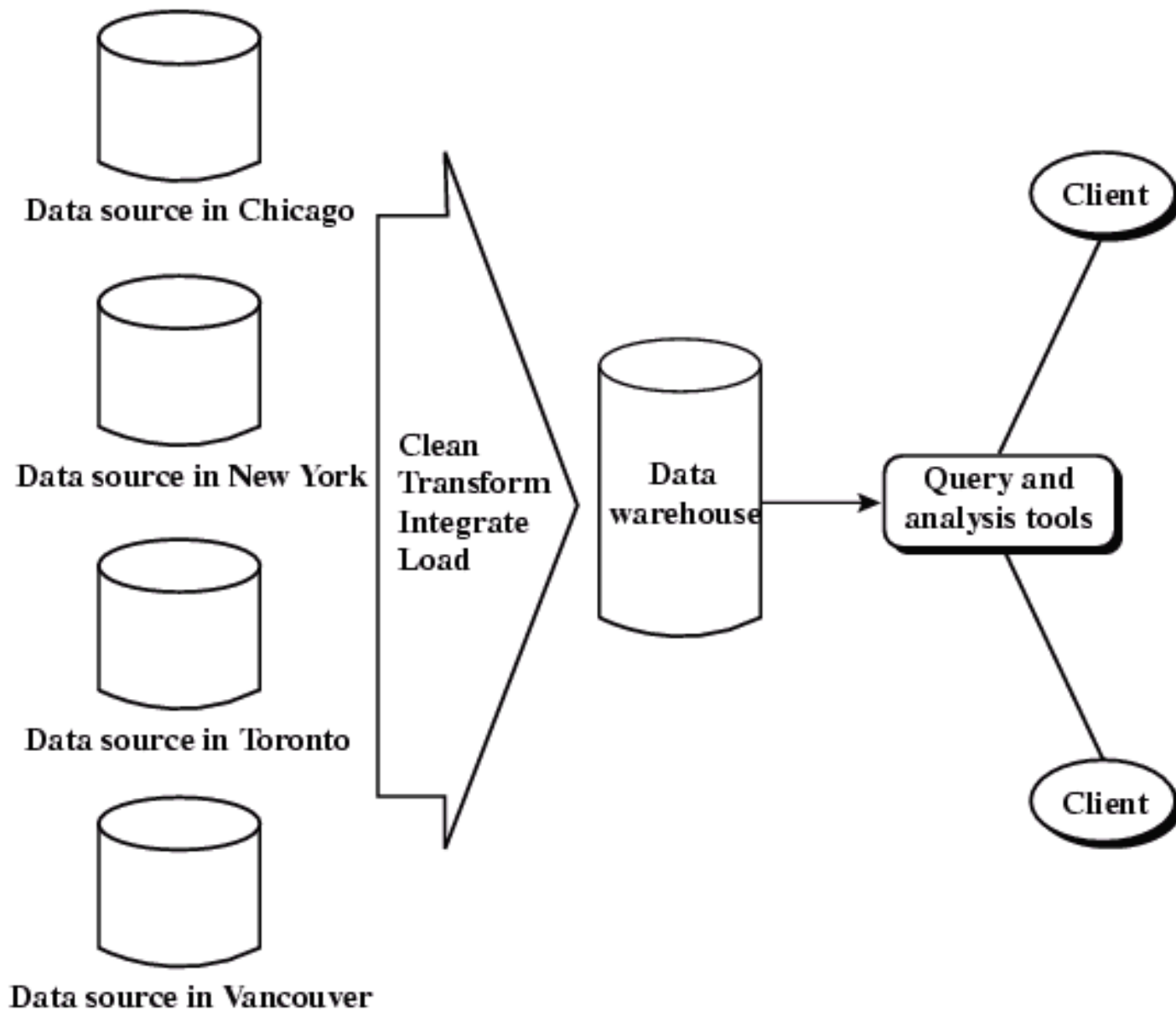
trans_ID	cust_ID	empl_ID	date	time	method_paid	amount
T100	C1	E55	09/21/98	15:45	Visa	\$1357.00
...	...	...	...	...	...	...

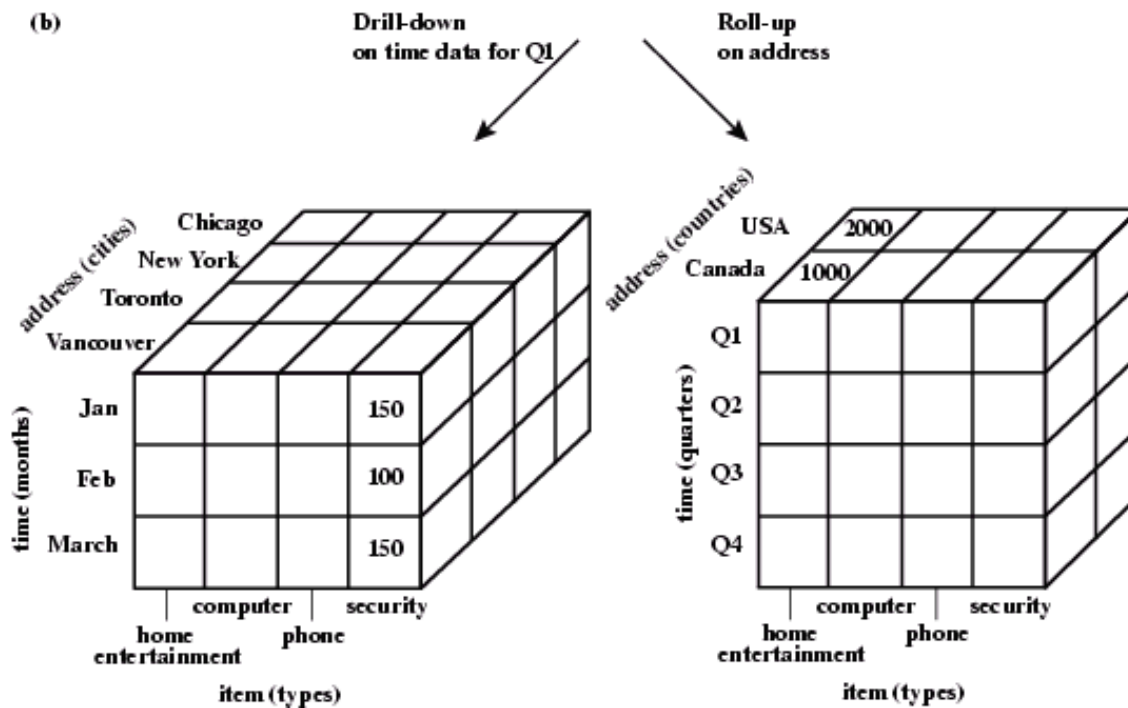
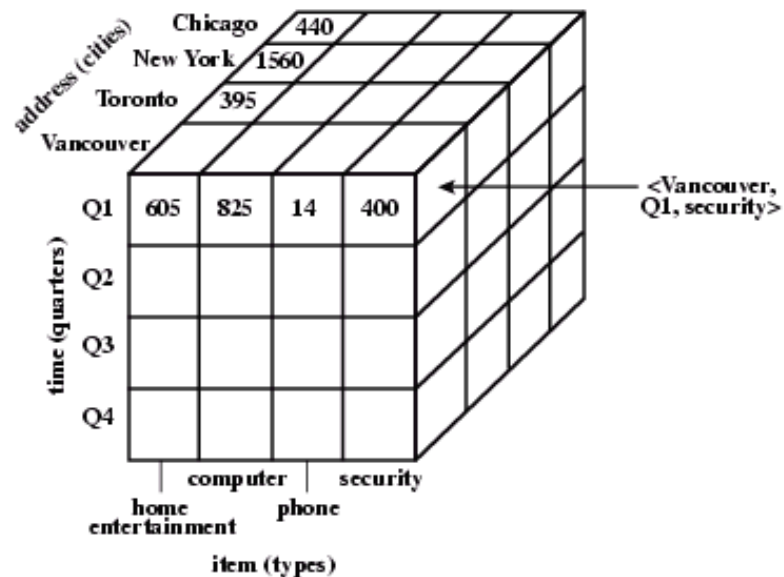
*items\_sold*

trans_ID	item_ID	qty
T100	I3	1
T100	I8	2
...	...	...

*works\_at*

empl_ID	branch_ID
E55	B1
...	...





*sales*

<b>trans_ID</b>	<b>list of item_IDs</b>
<b>T100</b>	<b>I1, I3, I8, I16</b>
<b>...</b>	<b>...</b>

1.	ELDB Stands for_____	Extreme Large Database	Extreme Large Data	Extremely Large Database	Extremely Large Data	Extremely Large Database
2.	_____ = $(2n + 1)P$	T	D	A	S	T
3.	How many managers are there in testing the data warehouse	3	4	5	6	5
4.	VLDB Stands for	Vast Large Data Base	Very Large Data Base	Very Low Database	Vertical Load Data Base	Very Large Data Base
5.	The _____ is centered on the user queries.	Daily Monitoring	Daily Processing	Daily transferring	Overnight Processing	Daily Processing
6.	_____ is a commodity that you can never enough of.	Unit	Disk	Memory	Hardware	Memory
7.	In _____, each development unit is tested on its own.	System Testing	Integration testing	Unit Testing	User Acceptance Testing	Unit Testing
8.	In _____, the separate development units that make up a component of the data warehouse application are tested to ensure that they work together.	System Testing	Integration testing	Unit Testing	User Acceptance Testing	Integration testing
9.	In _____, the whole data warehouse application is tested together.	Unit testing	Integration Testing	System Testing	User Acceptance Testing	System Testing
10.	One of the first tasks in developing a test plan is to come up with a _____	Test Schedule	Test Module	Test Data	Test Information	Test Schedule
11.	_____ is difficult to test unless you have clearly documented what is not allowed.	Disk Configuration	Test Schedule	Security	System	Security
12.	The testing of the database can be broken down into _____ separate sets of tests	1	2	3	4	3
13.	A _____ tool works by taking snapshots of statistics at various stages.	Monitoring	Active Monitoring	Passive Monitoring	Loading	Passive Monitoring
14.	A _____ tool gathers data continuously.	Monitoring	Active Monitoring	Passive Monitoring	Loading	Passive Monitoring

15.	UPS Stands for	Uninterruptible Power Supply	Uninterrupted Power System	Uninterrupted Power Scheduling	Unique Power System	Uninterruptible Power Supply
16.	Aggregation Can be classified into _____ types	1	2	3	4	2
17.	_____ is one of the most difficult things to measure in a data warehouse.	Data Performance	Database Performance	Query Performance	Performance choking	Query Performance
18.	_____ Aggregation are those that you already have stated requirements.	Unidentified Aggregation	Unique Aggregation	Identified Aggregation	Ad hoc Aggregation	Ad hoc Aggregation
19.	_____ Aggregation are those that are created to solve particular performance.	Unidentified Aggregation	Unique Aggregation	Identified Aggregation	Ad hoc Aggregation	Ad hoc Aggregation
20.	1 Petabyte = _____	1024 MB	1024 GB	1024 TB	1024 KB	1024 TB
21.	_____ Tools are extremely important during the system testing.	Monitoring	purging	modeling	loading	Monitoring
22.	Which of the following _____ is not included in the system test	overnight processing	backup recovery strategy	query performance	memory checking	memory checking
23.	_____ is difficult to test unless you have clearly documented what is not allowed.	Security	field	disk	system	Security
24.	During the system testing the _____ should be tested thoroughly to identify any potential I/O bottlenecks.	disk space	Disk configuration	disk load	disk memory unit	Disk configuration
25.	To control the daily operations of the data warehouse requires some sort of _____	unscheduled software	load software	scheduling software	processing software	scheduling software
26.	The capacity plan for a _____ is defined within the technical blueprint stage of the process.	Data mining	system	record	Data warehouse	Data warehouse
27.	The _____ stage should have identified the approximate sizes for data, users and any other issues that constrain system performance.	technical blue print	tuning	Business requirements	testing	Business requirements
28.	_____ are carried out prior to the	technical blue	tuning and	data load	data	tuning and testing

	completion of each build phase.	print	testing		processing	
29.	_____ is another task that can use massive parallelism to speed up us operation.	Data load	tuning and testing	Business requirements	data processing	Data load
30.	_____ as the baseline for the CPU required rather than the aggregation processing.	data processing	Post processing	Business requirements	pre processing	Post processing
31.	How many hours are allowed to complete the overnight processing?	11	12	10	15	10
32.	_____ is a different couple's query and or update plus an intensive write operation.	partitioning	tuning	testing	Aggregation	Aggregation
33.	_____ is a commodity that you can never enough of?	data	field	Memory	record	Memory
34.	The _____ will need memory to cache data blocks as they are used.	data	Database	information	Data warehouse	Database
35.	The largest calculation you will need to perform is the amount of _____ required.	disk load	Disk configuration	Disk space	disk check	Disk space
36.	How many types of data requirements are there?	5	4	3	2	2
37.	The _____ will occupy the bulk of the disk space that you use, and the task of getting the sizing right far from easy.	data	database	information	Data warehouse	database
38.	From the following which one is not included in calculating the actual size of fact or dimension data?	Average size of data in each field	the percentage occupancy of each field	the position of each data field in the table	Memory space of database	Memory space of database
39.	The final determinant of the ultimate size of the _____ is amount of data that you intend to keep online.	data load	fact data	Fact table	data disk	fact data
40.	When calculating the amount of non disk space _____ required?	configuration	Disk configuration	disk space	disk load	disk space
41.	8 Bit is equal to _____	1PB	1KB	1MB	1 byte	1 byte
42.	1024 byte is equal to _____	1MB	1 KB	1TB	1PB	1 KB
43.	1024 KB is equal to _____	1 MB	1PB	1KB	1TB	1 MB



44.	1024 MB is equal to_____	1MB	1 GB	1TB	1KB	1 GB
45.	1024 GB is equal to_____	1MB	1KB	1 TB	1PB	1 TB
46.	1024 TB is equal to _____	1MB	1TB	1KB	1 PB	1 PB
47.	_____ is the entry into the system.	Data Load	Data mining	Data disk	Data warehouse	Data Load
48.	_____ provides the opportunity to improve the system	Data mining	Data load	Data warehouse	Data disk	Data load
49.	_____ is the crucial part of overnight processing.	Data mining	Data load	Data warehouse	Data disk	Data load
50.	How many types of queries are there in data warehouse?	1	2	3	4	2
51.	_____ Queries that are clearly defined and well understood, such as regular reports, canned queries and common aggregation.	Ad hoc queries	integrated queries	consecutive queries	fixed	fixed
52.	_____ are unpredictable, both in quantify and frequency.	Ad hoc queries	integrated queries	consecutive queries	fixed	Ad hoc queries
53.	Restart able queries are a function of the _____ manager.	Query	warehouse	load	data	Query
54.	_____ Manager will be required to capture information about each query submitted.	data	Query	warehouse	load	Query
55.	_____ is that comprehensive testing of a data warehouse takes a long time.	down shot	adhoc	Upshot	Load	Upshot
56.	How many levels of testing are there?	3	4	5	6	3
57.	_____ is just traditional testing of developed code	Load manager	warehouse	query manager	all	all
58.	_____ testing will catch any dependency errors on moving from daily processing to overnight processing to daily processing again.	repetitive testing	system testing	normal testing	warehouse testing	repetitive testing
59.	Query manager gather _____ statistics	Query	warehouse	load	disk space	Query
60.	monitoring tools are extremely important during the _____	load testing	system testing	warehouse testing	System testing	System testing



# Data Mining

# Introduction

# Definition

Data mining is the exploration and analysis of large quantities of data in order to discover valid, novel, potentially useful, and ultimately understandable patterns in data.

**Valid:** The patterns hold in general.

**Novel:** We did not know the pattern beforehand.

**Useful:** We can devise **actions** from the patterns.

**Understandable:** We can interpret and comprehend the patterns.

# Case Study: Bank



- **Business goal:** Sell more home equity loans
- **Current models:**
  - Customers with college-age children use home equity loans to pay for tuition
  - Customers with variable income use home equity loans to even out stream of income
- **Data:**
  - Large data warehouse
  - Consolidates data from 42 operational data sources



# Case Study: Bank (Contd.)

1. Select subset of customer records who have received home equity loan offer
  - Customers who declined
  - Customers who signed up

Income	Number of Children	Average Checking Account Balance	...	Reponse
\$40,000	2	\$1500		Yes
\$75,000	0	\$5000		No
\$50,000	1	\$3000		No
...	...	...	...	...



## Case Study: Bank (Contd.)

2. Find rules to predict whether a customer would respond to home equity loan offer

IF (Salary < 40k) and  
(numChildren > 0) and  
(ageChild1 > 18 and ageChild1 < 22)

THEN YES

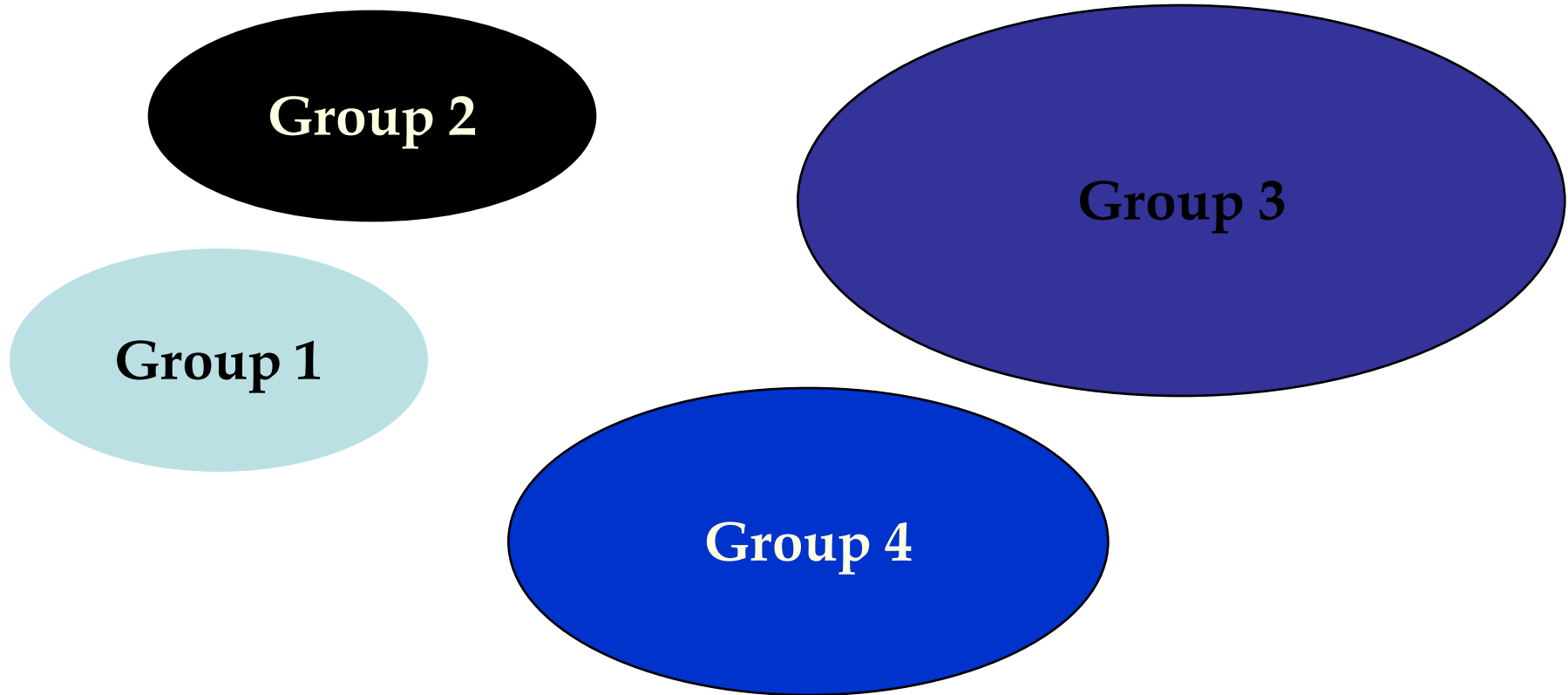
...



# Case Study: Bank (Contd.)



3. Group customers into clusters and investigate clusters



# Case Study: Bank (Contd.)



## 4. Evaluate results:

- Many “uninteresting” clusters
- **One interesting cluster!** Customers with both business and personal accounts; unusually high percentage of likely respondents

# Example: Bank (Contd.)



## Action:

- New marketing campaign

## Result:

- Acceptance rate for home equity offers more than doubled

# Example Application: Fraud Detection

- **Industries:** Health care, retail, credit card services, telecom, B2B relationships
- **Approach:**
  - Use historical data to build models of fraudulent behavior
  - Deploy models to identify fraudulent instances

# Fraud Detection (Contd.)

- Examples:
  - Auto insurance: Detect groups of people who stage accidents to collect insurance
  - Medical insurance: Fraudulent claims
  - Money laundering: Detect suspicious money transactions (US Treasury's Financial Crimes Enforcement Network)
  - Telecom industry: Find calling patterns that deviate from a norm (origin and destination of the call, duration, time of day, day of week).

# Other Example Applications

- CPG: Promotion analysis
- Retail: Category management
- Telecom: Call usage analysis, churn
- Healthcare: Claims analysis, fraud detection
- Transportation/Distribution: Logistics management
- Financial Services: Credit analysis, fraud detection
- Data service providers: Value-added data analysis

# What is a Data Mining Model?

A **data mining model** is a description of a certain aspect of a dataset. It produces output values for an assigned set of inputs.

## Examples:

- Clustering
- Linear regression model
- Classification model
- Frequent itemsets and association rules
- Support Vector Machines

# Data Mining Methods



# Overview

- Several well-studied tasks
  - Classification
  - Clustering
  - Frequent Patterns
- Many methods proposed for each
- Focus in database and data mining community:
  - Scalability
  - Managing the process
  - Exploratory analysis

# Classification

## Goal:

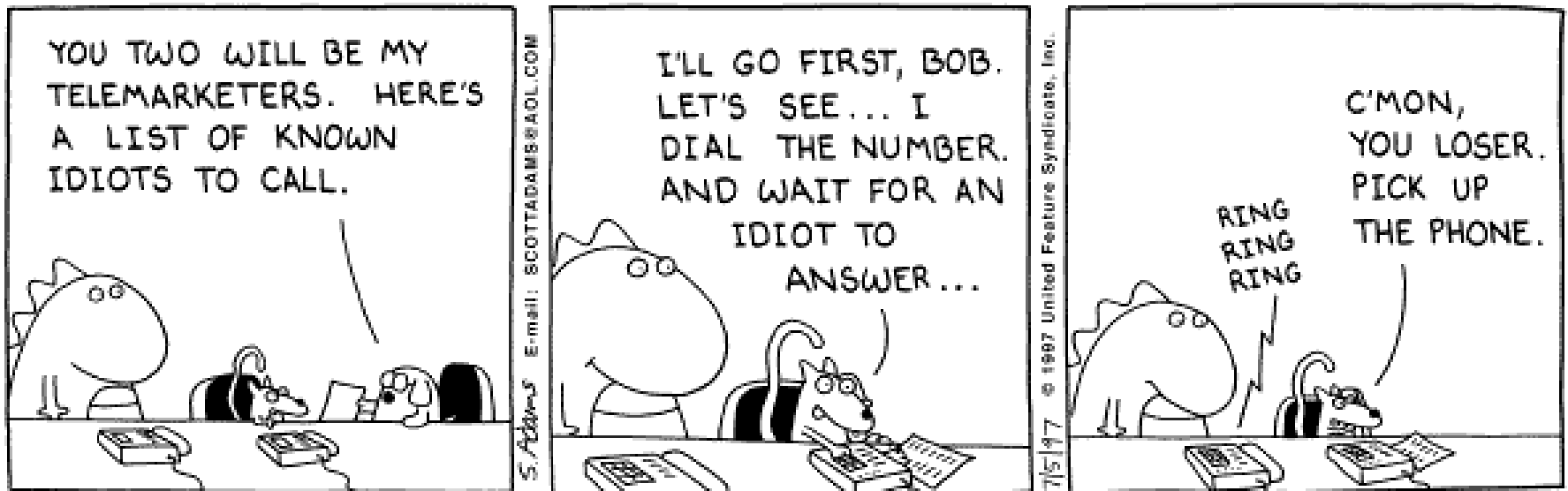
Learn a function that assigns a record to one of several predefined classes.

## Requirements on the model:

- High accuracy
- Understandable by humans, interpretable
- Fast construction for very large training databases

# Classification

## Example application: telemarketing



Copyright © 1997 United Feature Syndicate, Inc.  
Redistribution in whole or in part prohibited

# Classification (Contd.)

- Decision trees are one approach to classification.
- Other approaches include:
  - Linear Discriminant Analysis
  - *k*-nearest neighbor methods
  - Logistic regression
  - Neural networks
  - Support Vector Machines

# Classification Example

- Training database:
  - Two predictor attributes:  
Age and Car-type (**S**port, **M**inivan and **T**ruck)
  - Age is ordered, Car-type is categorical attribute
  - Class label indicates whether person bought product
  - Dependent attribute is *categorical*

Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

# Classification Problem

- If  $Y$  is categorical, the problem is a *classification problem*, and we use  $C$  instead of  $Y$ .  $|\text{dom}(C)| = J$ , the number of classes.
- $C$  is the *class label*,  $d$  is called a *classifier*.
- Let  $r$  be a record randomly drawn from  $P$ .  
Define the *misclassification rate* of  $d$ :  
$$RT(d, P) = P(d(r.X_1, \dots, r.X_k) \neq r.C)$$
- Problem definition: Given dataset  $D$  that is a random sample from probability distribution  $P$ , find classifier  $d$  such that  $RT(d, P)$  is minimized.

# Regression Problem

- If  $Y$  is numerical, the problem is a *regression problem*.
- $Y$  is called the dependent variable,  $d$  is called a *regression function*.
- Let  $r$  be a record randomly drawn from  $P$ .  
Define mean squared error rate of  $d$ :  
$$RT(d,P) = E(r.Y - d(r.X_1, \dots, r.X_k))^2$$
- Problem definition: Given dataset  $D$  that is a random sample from probability distribution  $P$ , find regression function  $d$  such that  $RT(d,P)$  is minimized.

# Regression Example

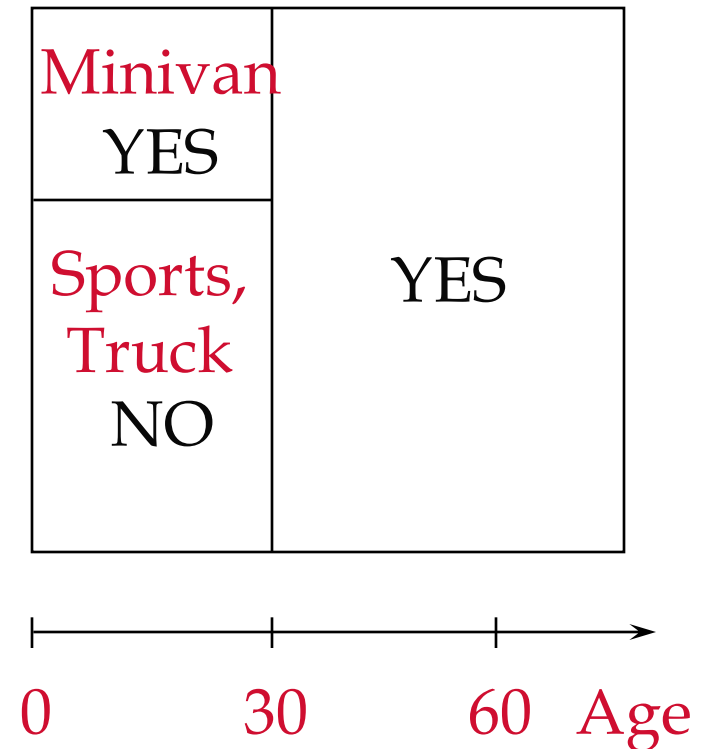
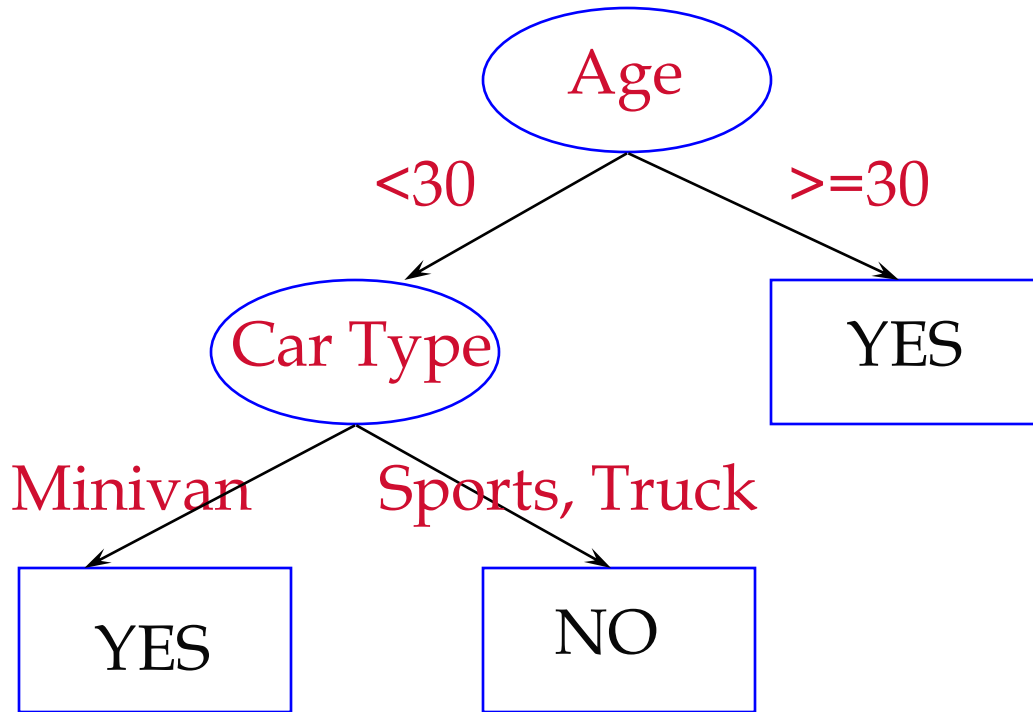
- Example training database
  - Two predictor attributes:  
Age and Car-type (**S**port, **M**inivan and **T**ruck)
  - Spent indicates how much person spent during a recent visit to the web site
  - Dependent attribute is *numerical*

Age	Car	Spent
20	M	\$200
30	M	\$150
25	T	\$300
30	S	\$220
40	S	\$400
20	T	\$80
30	M	\$100
25	M	\$125
40	M	\$500
20	S	\$420



# Decision Trees

# What are Decision Trees?



# Decision Trees

- A *decision tree*  $T$  encodes  $d$  (a classifier or regression function) in form of a tree.
- A node  $t$  in  $T$  without children is called a *leaf node*. Otherwise  $t$  is called an *internal node*.

# Internal Nodes

- Each internal node has an associated **splitting predicate**. Most common are binary predicates.

Example predicates:

- Age  $\leq 20$
- Profession in {student, teacher}
- $5000 \cdot \text{Age} + 3 \cdot \text{Salary} - 10000 > 0$

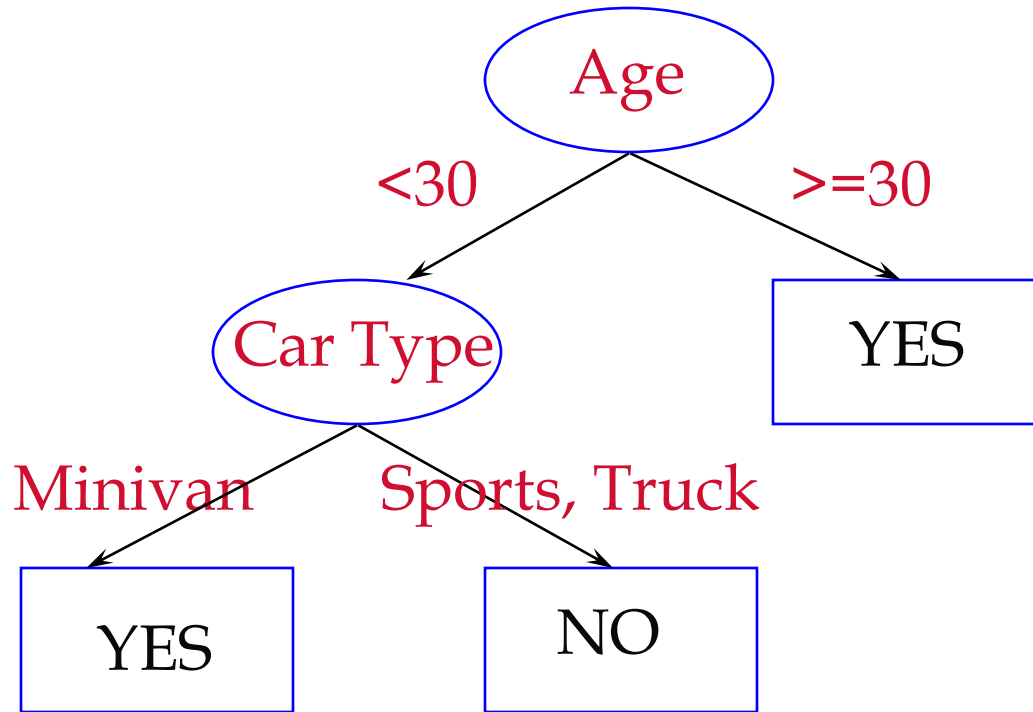
# Leaf Nodes

## Consider leaf node $t$ :

- Classification problem: Node  $t$  is labeled with one class label  $c$  in  $\text{dom}(C)$
- Regression problem: Two choices
  - Piecewise constant model:  
 $t$  is labeled with a constant  $y$  in  $\text{dom}(Y)$ .
  - Piecewise linear model:  
 $t$  is labeled with a linear model

$$Y = y_t + \sum a_i X_i$$

# Example



## Encoded classifier:

If (age<30 and  
carType=Minivan)  
Then YES

If (age <30 and  
(carType=Sports or  
carType=Truck))  
Then NO

If (age >= 30)  
Then YES

# Issues in Tree Construction

- Three algorithmic components:
  - Split Selection Method
  - Pruning Method
  - Data Access Method

# Top-Down Tree Construction

**BuildTree**(Node  $n$ , Training database  $D$ ,  
Split Selection Method  $\mathcal{S}$ )

[ (1) Apply  $\mathcal{S}$  to  $D$  to find splitting criterion ]

(1a) **for** each predictor attribute  $X$

(1b)     Call  $\mathcal{S}.\text{findSplit}(\text{AVC-set of } X)$

(1c) **endfor**

(1d)  $\mathcal{S}.\text{chooseBest}()$ ;

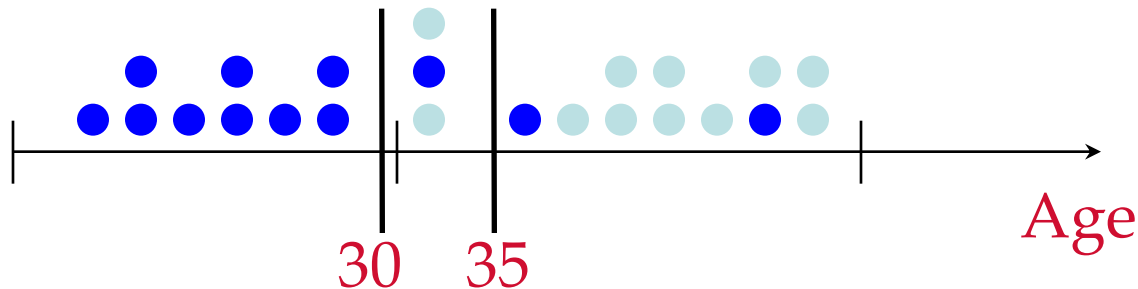
(2) **if** ( $n$  is not a leaf node) ...

$\mathcal{S}$ : C4.5, CART, CHAID, FACT, ID3, GID3, QUEST, etc.



# Split Selection Method

- Numerical Attribute: Find a split point that separates the (two) classes



(Yes: ● No: ● )

# Split Selection Method (Contd.)

- Categorical Attributes: How to group?

Sport: 

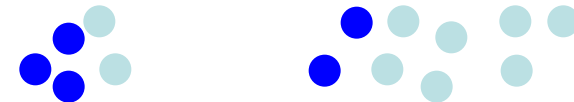
Truck: 

Minivan: 

(Sport, Truck) -- (Minivan)



(Sport) --- (Truck, Minivan)



(Sport, Minivan) --- (Truck)



# Impurity-based Split Selection Methods

- Split selection method has two parts:
  - Search space of possible splitting criteria.  
Example: All splits of the form “age  $\leq$  c”.
  - Quality assessment of a splitting criterion
- Need to quantify the quality of a split: **Impurity function**
- Example impurity functions: Entropy, gini-index, chi-square index

# Data Access Method

- Goal: Scalable decision tree construction, using the complete training database

# AVC-Sets

Training Database

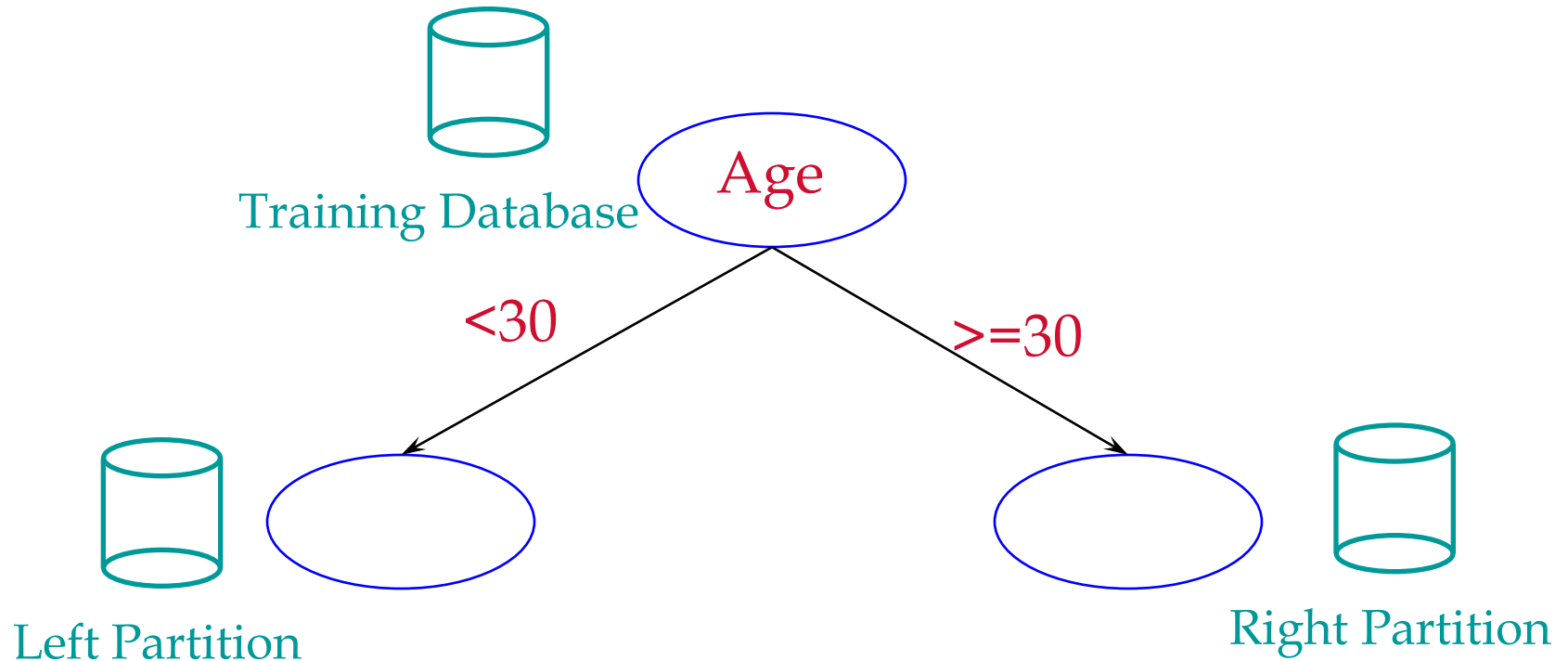
Age	Car	Class
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

AVC-Sets

Age	Yes	No
20	1	2
25	1	1
30	3	0
40	2	0

Car	Yes	No
Sport	2	1
Truck	0	2
Minivan	5	0

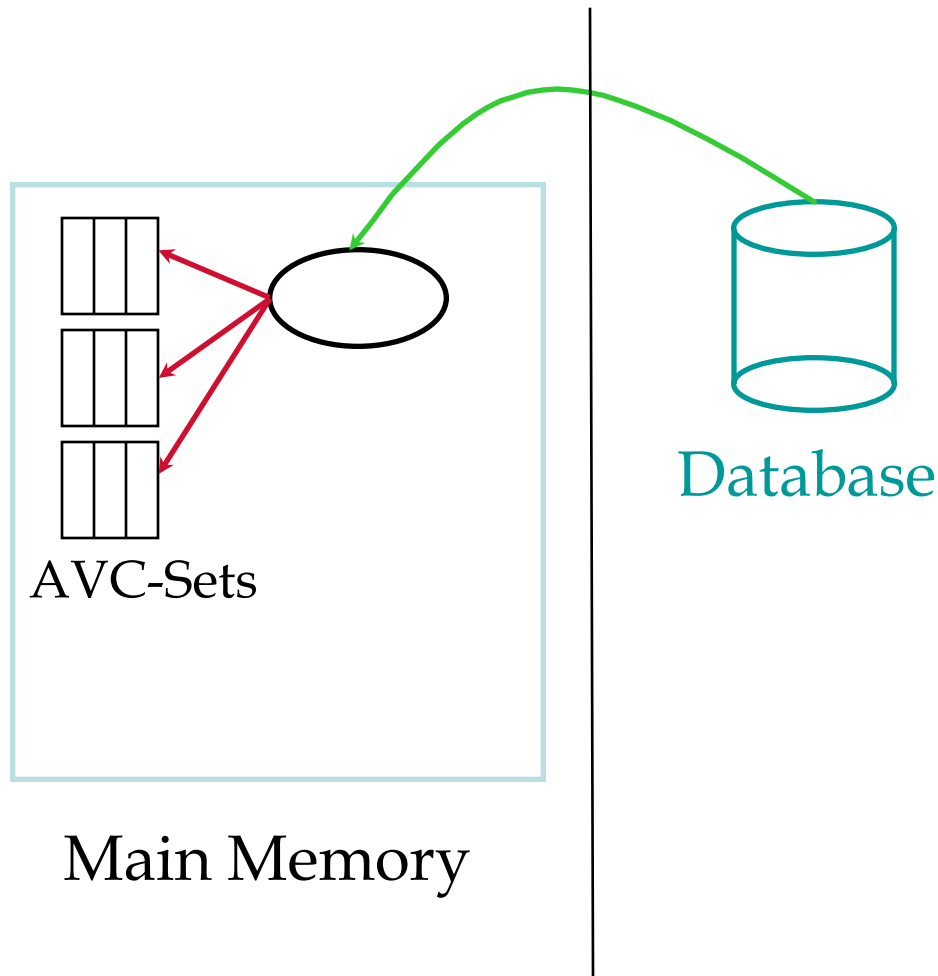
# Motivation for Data Access Methods



In principle, one pass over training database for each node.  
Can we improve?

# RainForest Algorithms: RF-Hybrid

First scan:

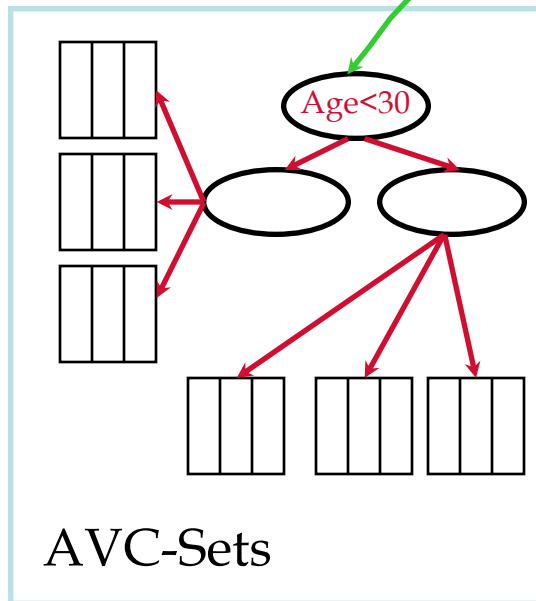


Build AVC-sets for root

# RainForest Algorithms: RF-Hybrid

Second Scan:

Build AVC sets for children of the root



Main Memory

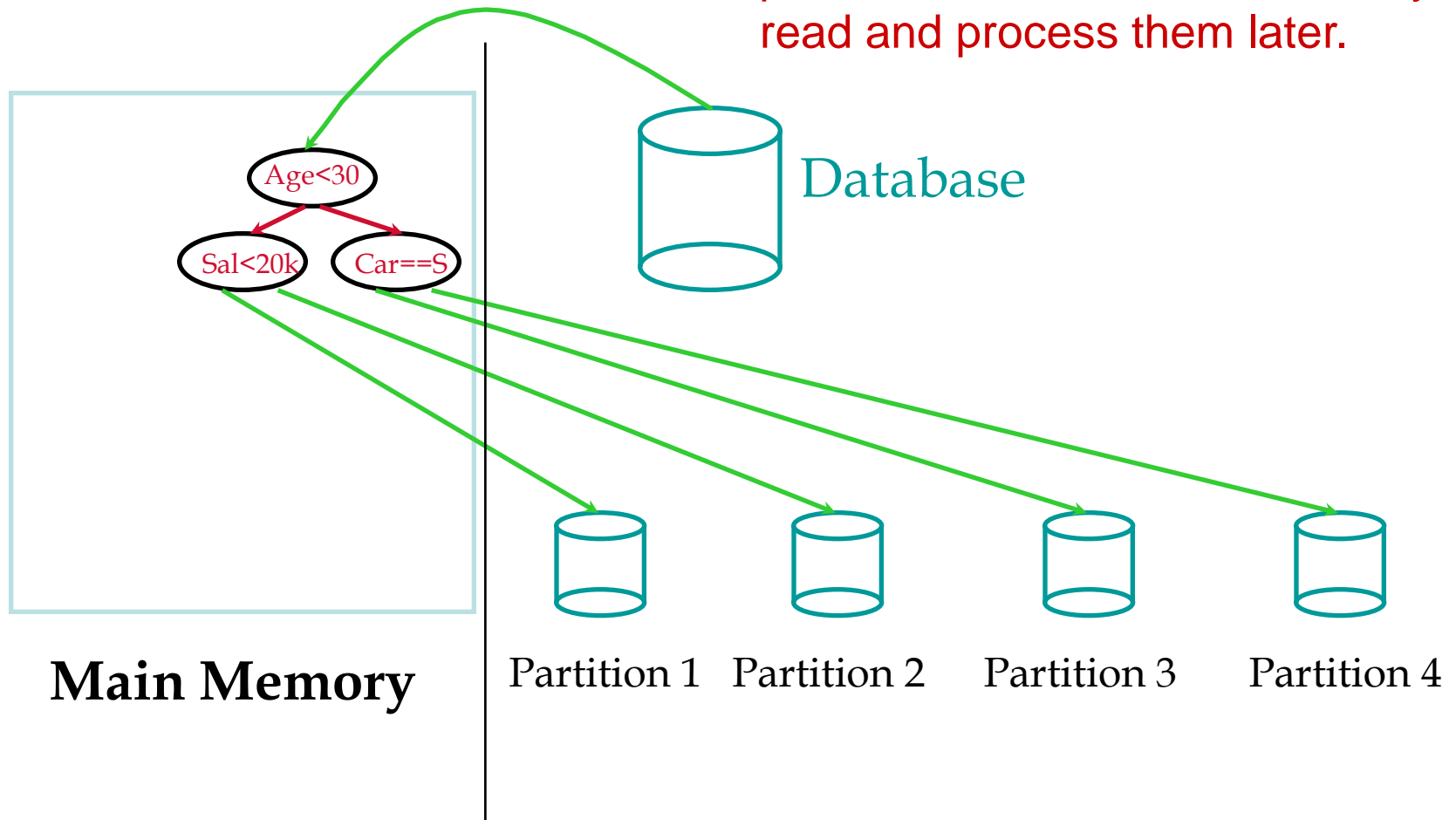




# RainForest Algorithms: RF-Hybrid

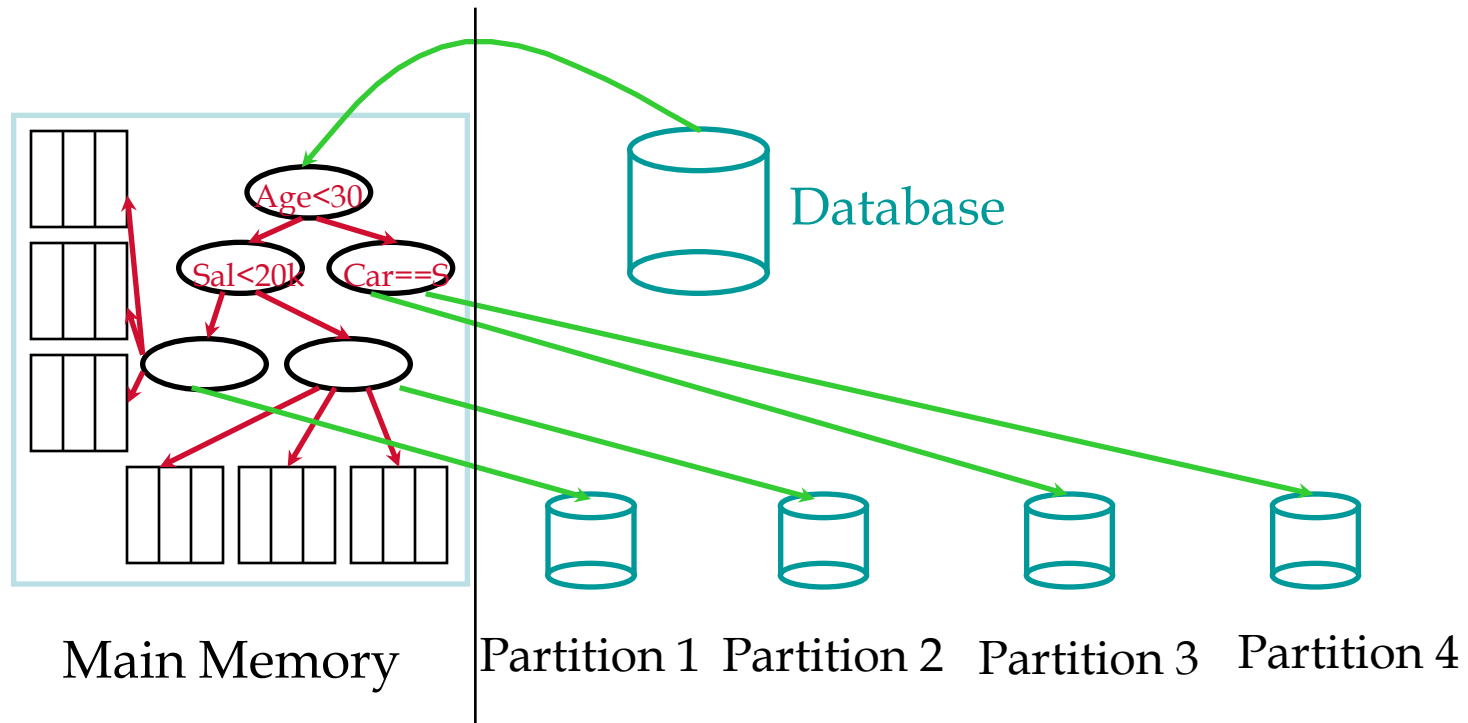
Third Scan:

As we expand the tree, we run out Of memory, and have to “spill” partitions to disk, and recursively read and process them later.



# RainForest Algorithms: RF-Hybrid

Further optimization: While writing partitions, concurrently build AVC-groups of as many nodes as possible in-memory. This should remind you of Hybrid Hash-Join!



# CLUSTERING

# Problem

- Given points in a multidimensional space, group them into a small number of **clusters**, using some measure of “nearness”
  - E.g., Cluster documents by topic
  - E.g., Cluster users by similar interests

# Clustering

- **Output:** (k) **groups** of records called **clusters**, such that the records within a group are more similar to records in other groups
  - Representative points for each cluster
  - Labeling of each record with each cluster number
  - Other description of each cluster
- *This is unsupervised learning:* No record labels are given to learn from
- Usage:
  - Exploratory data mining
  - Preprocessing step (e.g., outlier detection)

# Clustering (Contd.)

- **Requirements:** Need to define “similarity” between records
- **Important:** Use the “right” similarity (distance) function
  - Scale or normalize all attributes. Example: seconds, hours, days
  - Assign different weights to reflect importance of the attribute
  - Choose appropriate measure (e.g., L1, L2)

# Approaches

- **Centroid-based:** Assume we have  $k$  clusters, guess at the centers, assign points to nearest center, e.g., K-means; over time, centroids shift
- **Hierarchical:** Assume there is one cluster per point, and repeatedly merge nearby clusters using some distance threshold

Scalability: Do this with fewest number of passes over data, ideally, sequentially

# Scalable Clustering Algorithms for Numeric Attributes

CLARANS

DBSCAN

BIRCH

CLIQUE

CURE

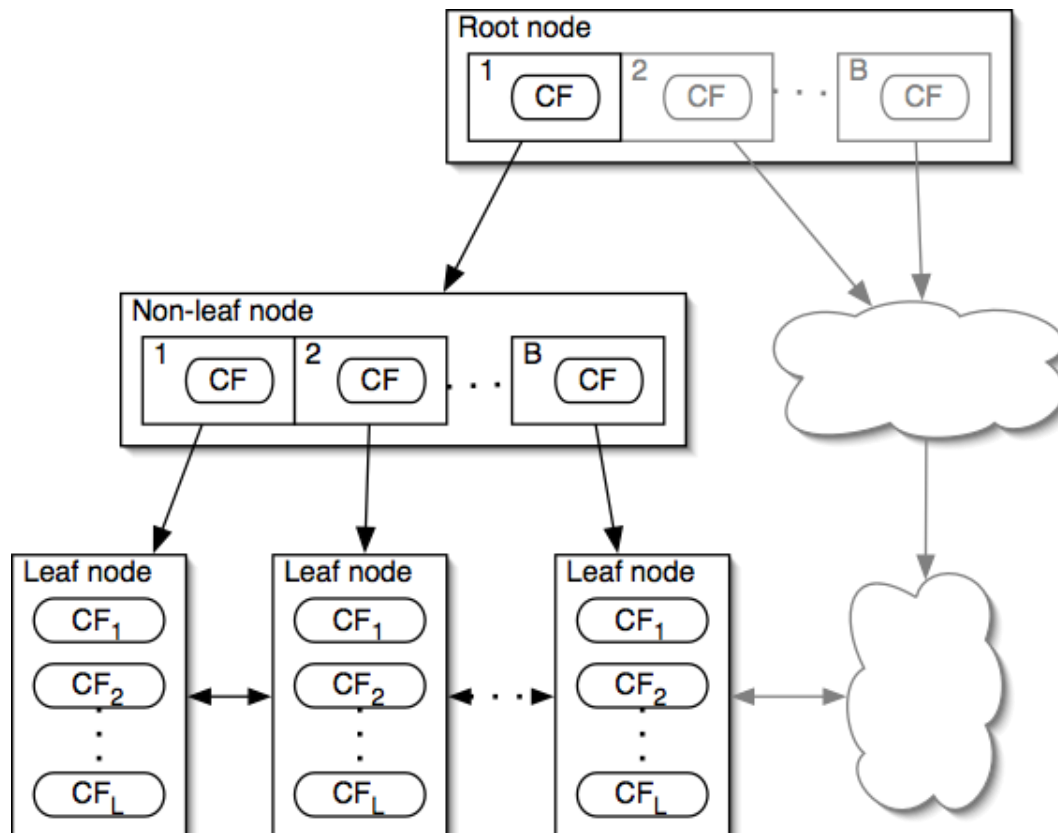
.....

- Above algorithms can be used to cluster documents after reducing their dimensionality using SVD



# Birch [ZRL96]

Pre-cluster data points using “CF-tree” data structure



# Clustering Feature (CF)

Given a cluster  $\{\vec{X}_1, \vec{X}_2, \dots, \vec{X}_N\}$      $\mathbf{CF} = (N, \vec{LS}, SS)$

$N$  is the number of data points

$$\vec{LS} = \sum_{i=1}^N \vec{X}_i$$

$$SS = \sum_{i=1}^N \vec{X}_i^2$$

$$\mathbf{CF}_1 + \mathbf{CF}_2 = (N_1 + N_2, \vec{LS}_1 + \vec{LS}_2, SS_1 + SS_2)$$

*Allows incremental merging of clusters!*

# Points to Note

- Basic algorithm works in a single pass to condense metric data using spherical summaries
  - Can be incremental
- Additional passes cluster CFs to detect non-spherical clusters
- Approximates density function
- Extensions to non-metric data

# Market Basket Analysis: Frequent Itemsets

# Market Basket Analysis

- Consider shopping cart filled with several items
- Market basket analysis tries to answer the following questions:
  - Who makes purchases
  - What do customers buy

# Market Basket Analysis

- **Given:**
  - A database of customer transactions
  - Each transaction is a set of items
- **Goal:**
  - Extract rules

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

# Market Basket Analysis (Contd.)

- **Co-occurrences**
  - 80% of all customers purchase items X, Y and Z together.
- **Association rules**
  - 60% of all customers who purchase X and Y also buy Z.
- **Sequential patterns**
  - 60% of customers who first buy X also purchase Y within three weeks.

# Confidence and Support

We prune the set of all possible association rules using two interestingness measures:

- **Confidence** of a rule:
  - $X \Rightarrow Y$  has confidence  $c$  if  $P(Y|X) = c$
- **Support** of a rule:
  - $X \Rightarrow Y$  has support  $s$  if  $P(XY) = s$

We can also define

- **Support of a co-occurrence**  $XY$ :
  - $XY$  has support  $s$  if  $P(XY) = s$



# Example

- Example rule:  
 $\{\text{Pen}\} \Rightarrow \{\text{Milk}\}$   
Support: 75%  
Confidence: 75%
- Another example:  
 $\{\text{Ink}\} \Rightarrow \{\text{Pen}\}$   
Support: 100%  
Confidence: 100%

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

# Exercise

- Can you find all itemsets with support  $\geq 75\%$ ?

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

# Exercise

- Can you find all association rules with support  $\geq 50\%$ ?

TID	CID	Date	Item	Qty
111	201	5/1/99	Pen	2
111	201	5/1/99	Ink	1
111	201	5/1/99	Milk	3
111	201	5/1/99	Juice	6
112	105	6/3/99	Pen	1
112	105	6/3/99	Ink	1
112	105	6/3/99	Milk	1
113	106	6/5/99	Pen	1
113	106	6/5/99	Milk	1
114	201	7/1/99	Pen	2
114	201	7/1/99	Ink	2
114	201	7/1/99	Juice	4

# Extensions

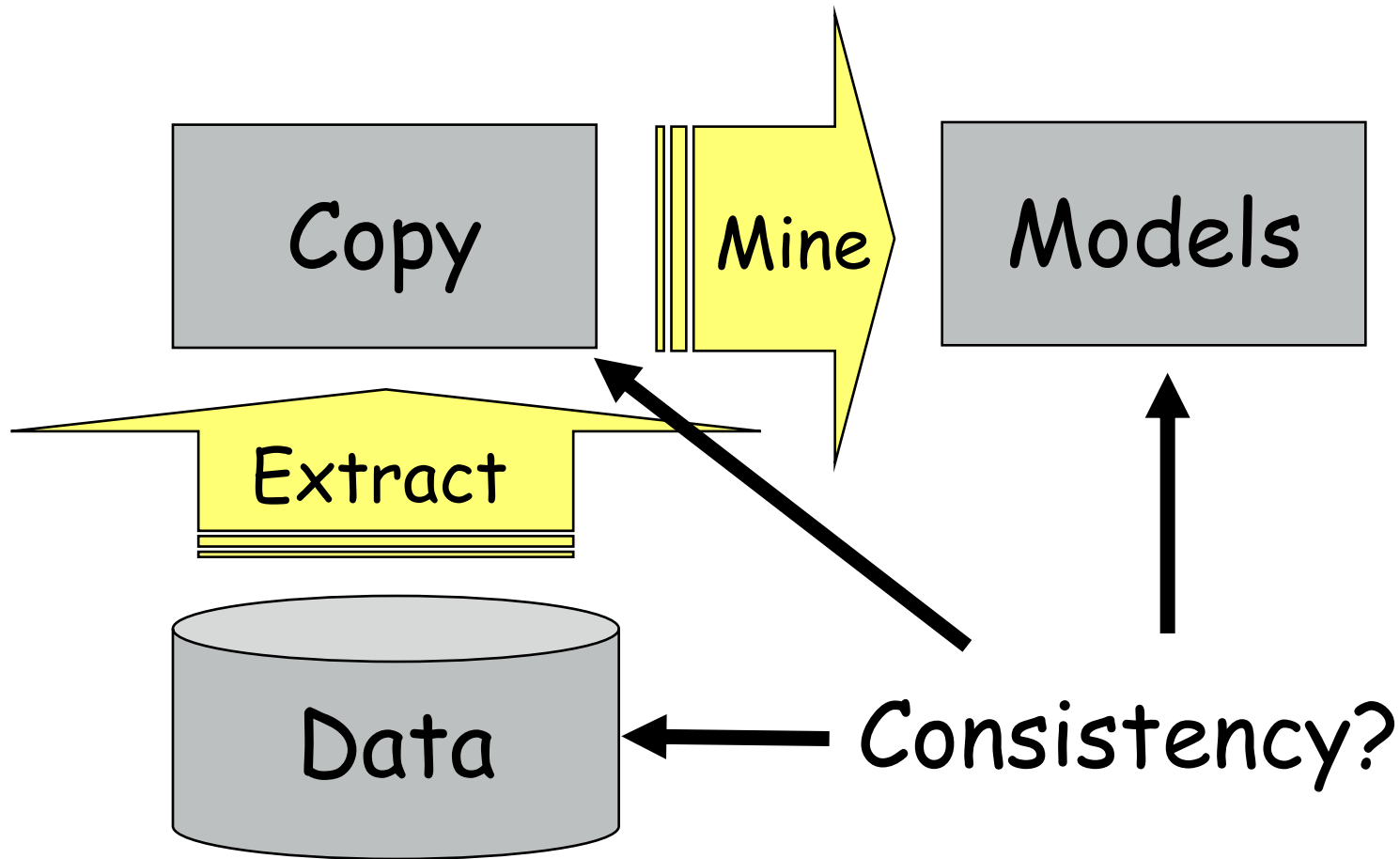
- Imposing constraints
  - Only find rules involving the dairy department
  - Only find rules involving expensive products
  - Only find rules with “whiskey” on the right hand side
  - Only find rules with “milk” on the left hand side
  - Hierarchies on the items
  - Calendars (every Sunday, every 1<sup>st</sup> of the month)

# Market Basket Analysis: Applications

- Sample Applications
  - Direct marketing
  - Fraud detection for medical insurance
  - Floor/shelf planning
  - Web site layout
  - Cross-selling

# DBMS Support for DM

# Why Integrate DM into a DBMS?



# Integration Objectives

- Avoid isolation of querying from mining
  - Difficult to do “ad-hoc” mining
- Provide simple programming approach to creating and using DM models
- Make it possible to add new models
- Make it possible to add new, scalable algorithms

Analysts (users)

DM Vendors



# SQL/MM: Data Mining

- A collection of classes that provide a standard interface for invoking DM algorithms from SQL systems.
- Four data models are supported:
  - Frequent itemsets, association rules
  - Clusters
  - Regression trees
  - Classification trees

# DATA MINING SUPPORT IN MICROSOFT SQL SERVER \*

\* Thanks to Surajit Chaudhuri for permission to use/adapt his slides

# Key Design Decisions

- Adopt relational data representation
  - A Data Mining Model (DMM) as a “tabular” object (externally; can be represented differently internally)
- Language-based interface
  - Extension of SQL
  - Standard syntax

# DM Concepts to Support

- Representation of input (*cases*)
- Representation of *models*
- Specification of *training step*
- Specification of *prediction step*

Should be independent of specific algorithms

# What are “Cases”?

- DM algorithms analyze “cases”
- The “case” is the entity being categorized and classified
- Examples
  - Customer credit risk analysis: Case = Customer
  - Product profitability analysis: Case = Product
  - Promotion success analysis: Case = Promotion
- Each case encapsulates all we know about the entity

# Cases as Records: Examples

<b>Cust ID</b>	<b>Age</b>	<b>Marital Status</b>	<b>Wealth</b>
1	35	M	380,000
2	20	S	50,000
3	57	M	470,000

<b>Age</b>	<b>Car</b>	<b>Class</b>
20	M	Yes
30	M	Yes
25	T	No
30	S	Yes
40	S	Yes
20	T	No
30	M	Yes
25	M	Yes
40	M	Yes
20	S	No

# Types of Columns

<u>Cust ID</u>	Age	Marital Status	Wealth	Product Purchases		
				Product	Quantity	Type
1	35	M	380,000	TV	1	Appliance
				Coke	6	Drink
				Ham	3	Food

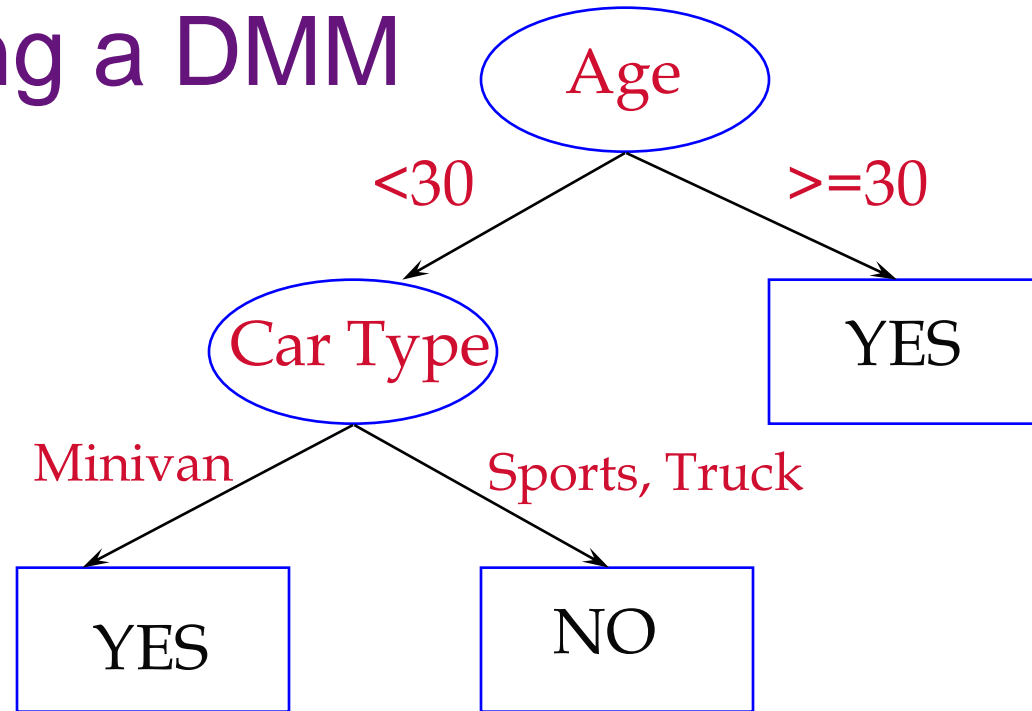
- Keys: Columns that uniquely identify a case
- Attributes: Columns that describe a case
  - Value: A state associated with the attribute in a specific case
  - Attribute Property: Columns that describe an attribute
    - Unique for a specific attribute value (TV is always an appliance)
  - Attribute Modifier: Columns that represent additional “meta” information for an attribute
    - Weight of a case, Certainty of prediction

# More on Columns

- Properties describe attributes
  - Can represent generalization hierarchy
- Distribution information associated with attributes
  - Discrete/Continuous
  - Nature of Continuous distributions
    - Normal, Log\_Normal
  - Other Properties (e.g., ordered, not null)



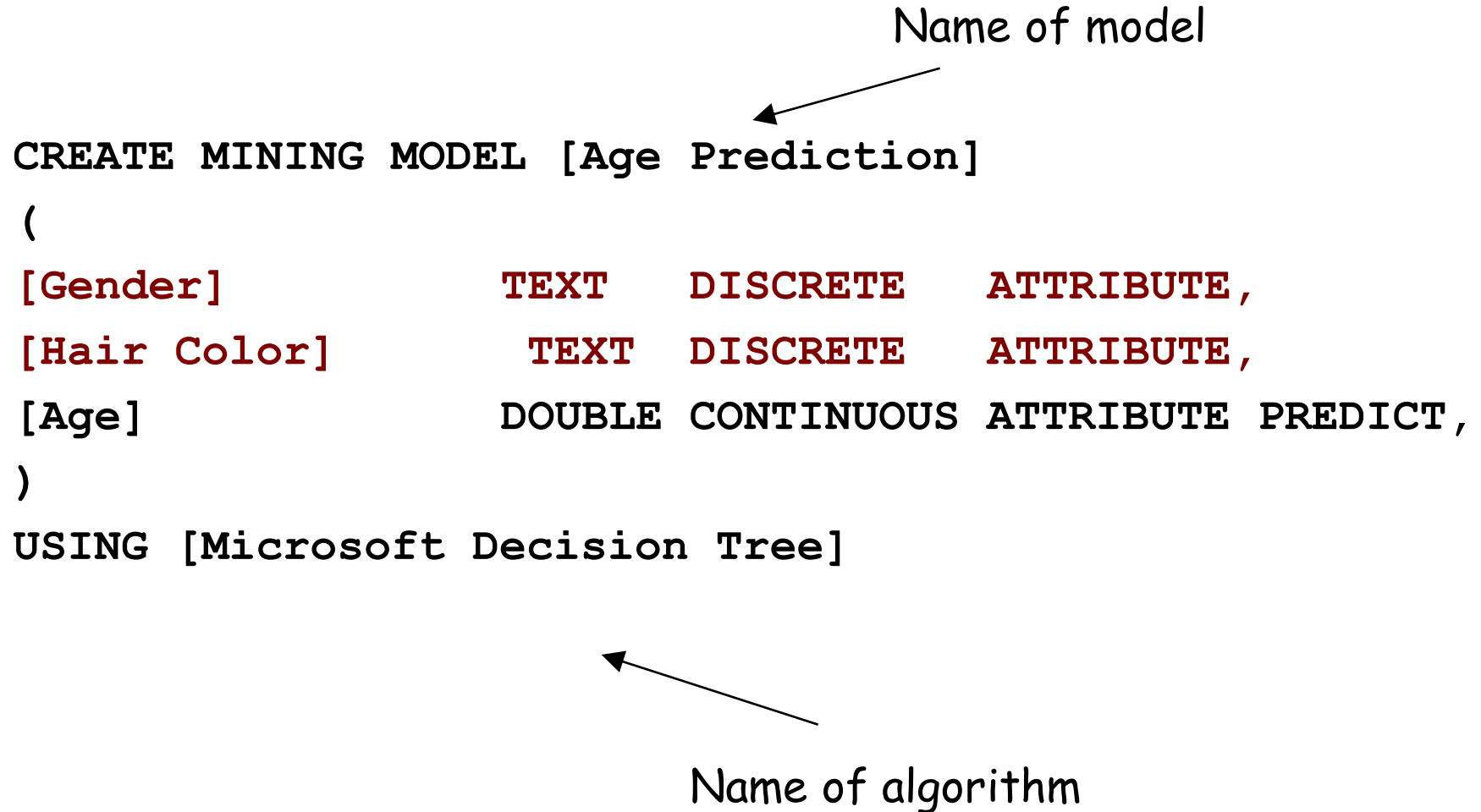
# Representing a DMM



- Specifying a Model
  - Columns to predict
  - Algorithm to use
  - Special parameters
- Model is represented as a (nested) table
  - Specification = Create table
  - Training = Inserting data into the table
  - Predicting = Querying the table

# CREATE MINING MODEL

Name of model



```
CREATE MINING MODEL [Age Prediction]
(
  [Gender]           TEXT    DISCRETE    ATTRIBUTE,
  [Hair Color]       TEXT    DISCRETE    ATTRIBUTE,
  [Age]              DOUBLE  CONTINUOUS  ATTRIBUTE PREDICT,
)
USING [Microsoft Decision Tree]
```

Name of algorithm

# CREATE MINING MODEL

```
CREATE MINING MODEL [Age Prediction]
(
[Customer ID]    LONG        KEY,
[Gender]          TEXT        DISCRETE    ATTRIBUTE,
[Age]             DOUBLE      CONTINUOUS  ATTRIBUTE PREDICT,
[ProductPurchases] TABLE (
[ProductName]     TEXT        KEY,
[Quantity]        DOUBLE      NORMAL CONTINUOUS,
[ProductType]     TEXT DISCRETE RELATED TO [ProductName]
)
)
USING [Microsoft Decision Tree]
```

Note that the ProductPurchases column is a nested table. SQL Server computes this field when data is "inserted".

# Training a DMM

- Training a DMM requires passing it “known” cases
  - Use an INSERT INTO in order to “insert” the data to the DMM
    - The DMM will usually not retain the inserted data
    - Instead it will analyze the given cases and build the DMM content (decision tree, segmentation model)
- 
- `INSERT [INTO] <mining model name>`  
    `[(columns list)]`  
    `<source data query>`

# INSERT INTO

```
INSERT INTO [Age Prediction]
(
  [Gender], [Hair Color], [Age]
)
OPENQUERY ([Provider=MSOLESQL...,
'SELECT
    [Gender], [Hair Color], [Age]
  FROM [Customers]'
)
```

# Executing Insert Into

- The DMM is trained
  - The model can be retrained or incrementally refined
- Content (rules, trees, formulas) can be explored
- Prediction queries can be executed

# What are Predictions?

- Predictions apply the trained model to estimate missing attributes in a data set
- Predictions = Queries
- Specification:
  - Input data set
  - A trained DMM (think of it as a truth table, with one row per combination of predictor-attribute values; this is only conceptual)
  - Binding (mapping) information between the input data and the DMM

# Prediction Join

```
SELECT [Customers].[ID],  
       MyDMM.[Age],  
       PredictProbability(MyDMM.[Age])  
FROM  
  MyDMM PREDICTION JOIN [Customers]  
  ON MyDMM.[Gender] = [Customers].[Gender] AND  
  MyDMM.[Hair Color] =  
                                [Customers].[Hair Color]
```



# Exploratory Mining: Combining OLAP and DM

# Databases and Data Mining

- What can database systems offer in the grand challenge of understanding and learning from the flood of data we've unleashed?
  - The plumbing
  - Scalability

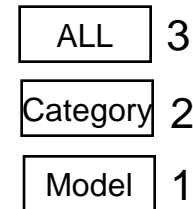
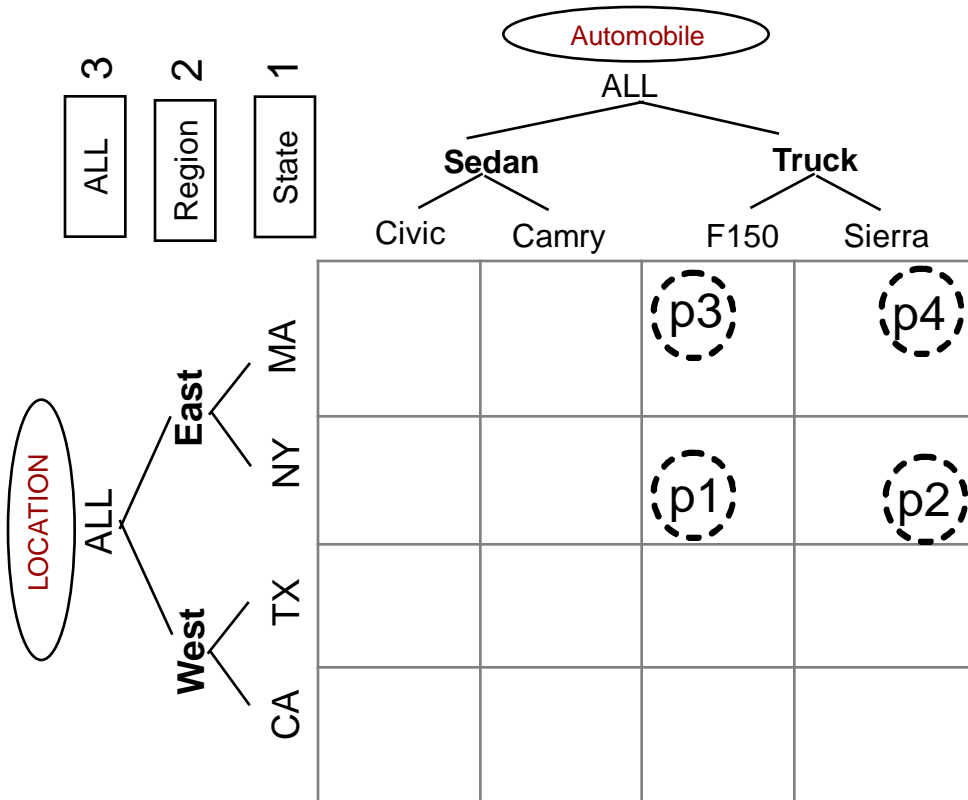
# Databases and Data Mining

- What can database systems offer in the grand challenge of understanding and learning from the flood of data we've unleashed?
  - The plumbing
  - Scalability
  - Ideas!
    - Declarativeness
    - Compositionality
    - **Ways to conceptualize your data**

# Multidimensional Data Model

- One fact table  $\Delta=(\mathbf{X},\mathbf{M})$ 
  - $\mathbf{X}=X_1, X_2, \dots$  Dimension attributes
  - $\mathbf{M}=M_1, M_2, \dots$  Measure attributes
- Domain hierarchy for each dimension attribute:
  - Collection of domains  $\text{Hier}(X_i) = (D_i^{(1)}, \dots, D_i^{(k)})$
  - The extended domain:  $EX_i = \cup_{1 \leq k \leq t} DX_i^{(k)}$
- Value mapping function:  $\gamma_{D_1 \rightarrow D_2}(x)$ 
  - e.g.,  $\gamma_{\text{month} \rightarrow \text{year}}(12/2005) = 2005$
  - Form the value hierarchy graph
  - Stored as dimension table attribute (e.g., week for a time value) or conversion functions (e.g., month, quarter)

# Multidimensional Data



**DIMENSION  
ATTRIBUTES**

<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200

# Cube Space

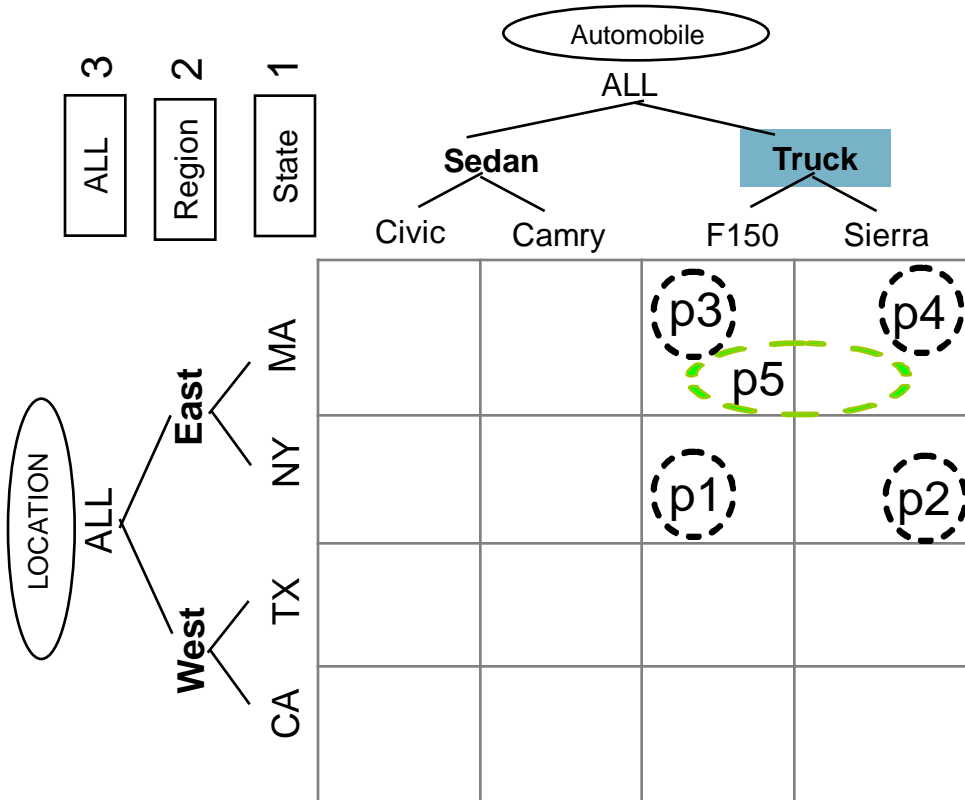
- Cube space:  $C = EX_1 \times EX_2 \times \dots \times EX_d$
- Region: Hyper rectangle in cube space
  - $c = (v_1, v_2, \dots, v_d)$  ,  $v_i \in EX_i$
- Region granularity:
  - $\text{gran}(c) = (d_1, d_2, \dots, d_d)$ ,  $d_i = \text{Domain}(c.v_i)$
- Region coverage:
  - $\text{coverage}(c) = \text{all facts in } c$
- Region set: All regions with same granularity

# OLAP Over Imprecise Data

with Doug Burdick, Prasad Deshpande, T.S. Jayram, and  
Shiv Vaithyanathan

In VLDB 05, 06 joint work with IBM Almaden

# Imprecise Data



ALL	3
Category	2
Model	1

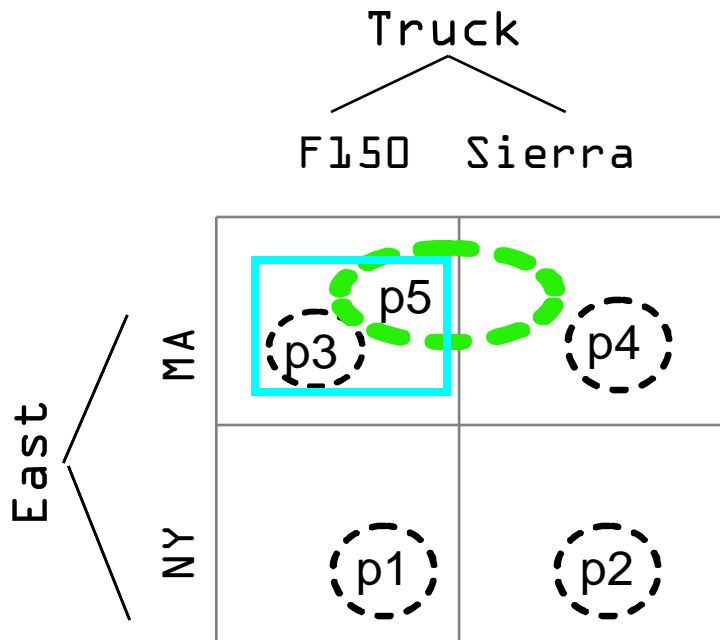
<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100



# Querying Imprecise Facts

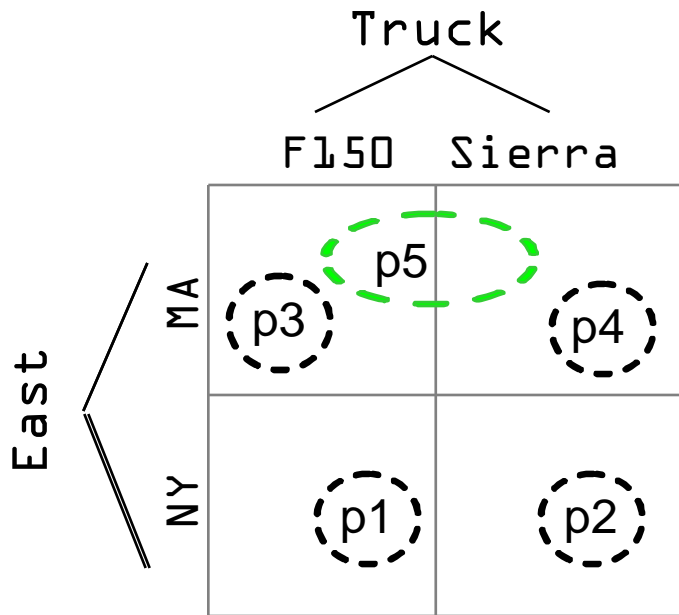
Auto = F150  
Loc = MA  
SUM(Repair) = ???

*How do we treat p5?*



<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

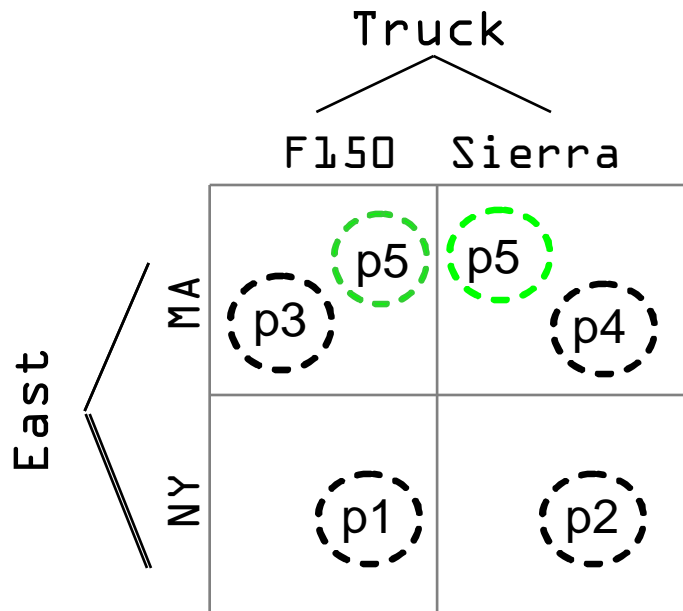
# Allocation (1)



<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>
p1	F150	NY	100
p2	Sierra	NY	500
p3	F150	MA	100
p4	Sierra	MA	200
p5	Truck	MA	100

# Allocation (2)

(Huh? Why 0.5 / 0.5?  
- Hold on to that thought)

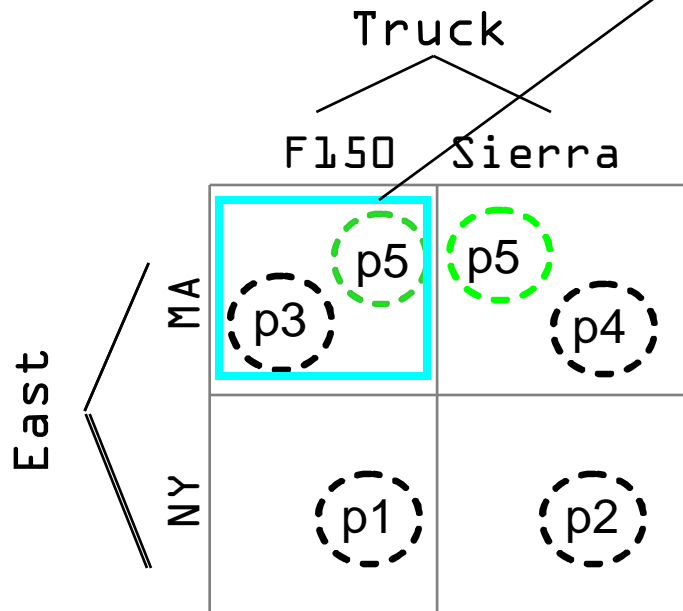


<i>ID</i>	<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>	<i>Weight</i>
1	p1	F150	NY	100	1.0
2	p2	Sierra	NY	500	1.0
3	p3	F150	MA	100	1.0
4	p4	Sierra	MA	200	1.0
5	p5	F150	MA	100	0.5
6	p5	Sierra	MA	100	0.5

# Allocation (3)

Auto = F150  
Loc = MA  
SUM(Repair) = 150

Query the Extended Data Model!

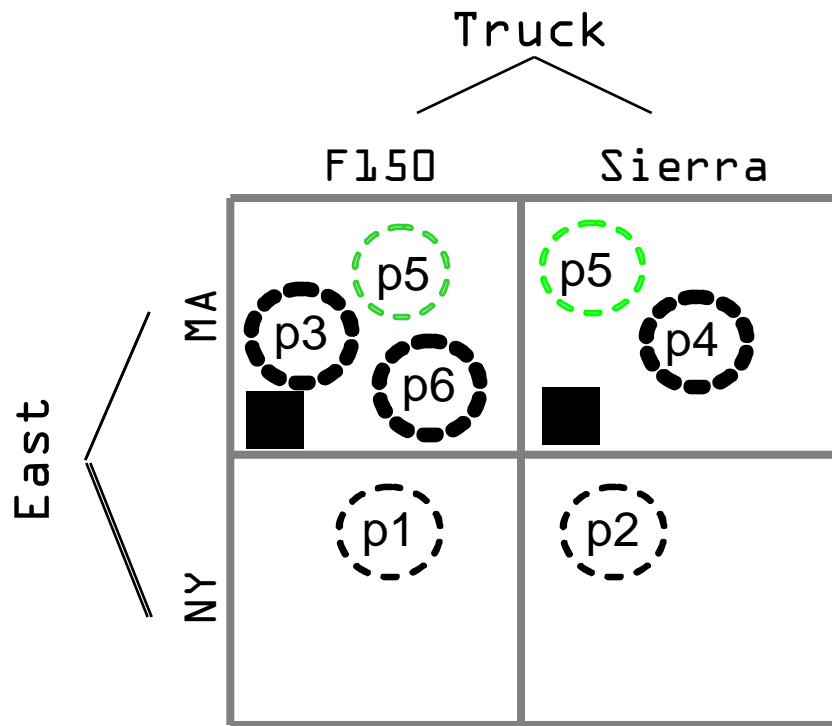


<i>ID</i>	<i>FactID</i>	<i>Auto</i>	<i>Loc</i>	<i>Repair</i>	<i>Weight</i>
1	p1	F150	NY	100	1.0
2	p2	Sierra	NY	500	1.0
3	p3	F150	MA	100	1.0
4	p4	Sierra	MA	200	1.0
5	p5	F150	MA	100	0.5
6	p5	Sierra	MA	100	0.5

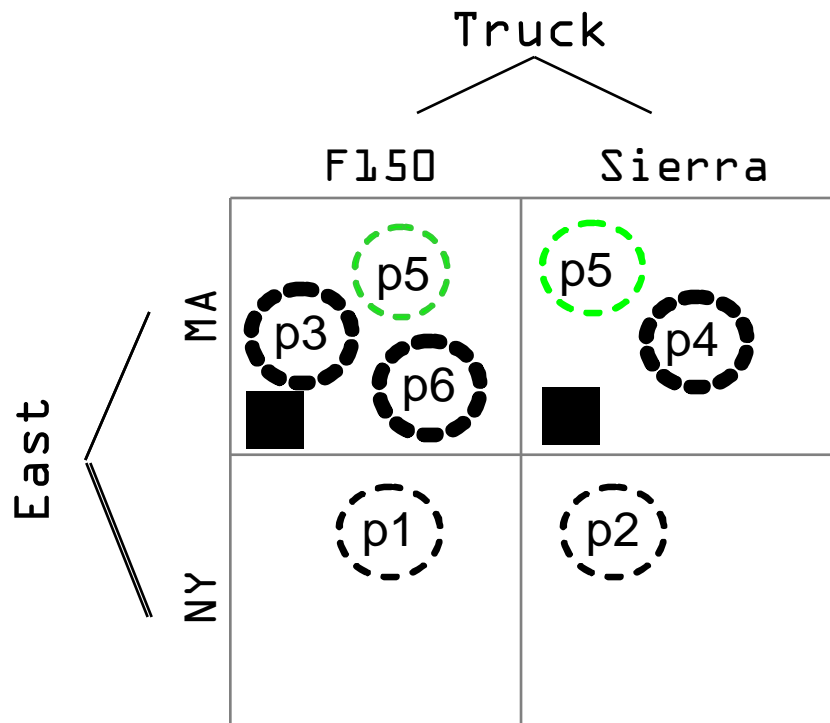
# Allocation Policies

- The procedure for assigning allocation weights is referred to as an allocation policy:
  - Each allocation policy uses different information to assign allocation weights
  - Reflects assumption about the correlation structure in the data
    - Leads to EM-style iterative algorithms for allocating imprecise facts, maximizing likelihood of observed data

# Allocation Policy: *Count*

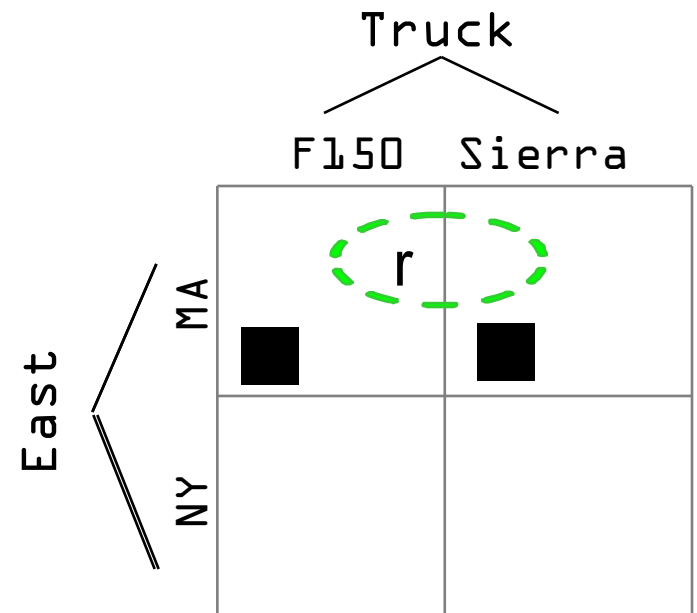


# Allocation Policy: *Measure*



<i>ID</i>	<i>Sales</i>
p1	100
p2	150
p3	300
p4	200
p5	250
p6	400

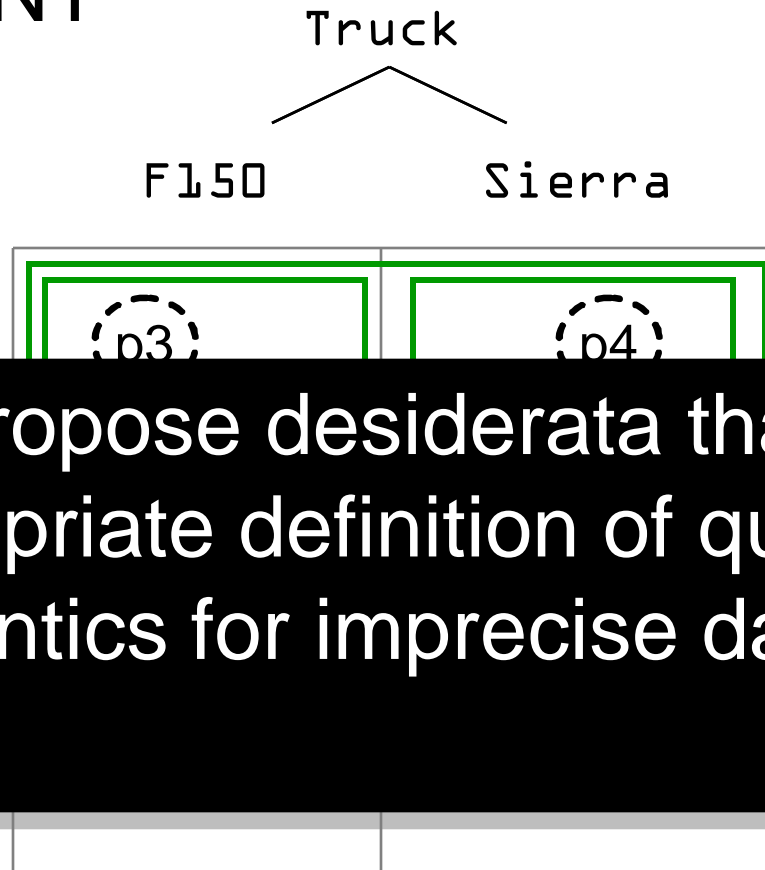
# Allocation Policy Template





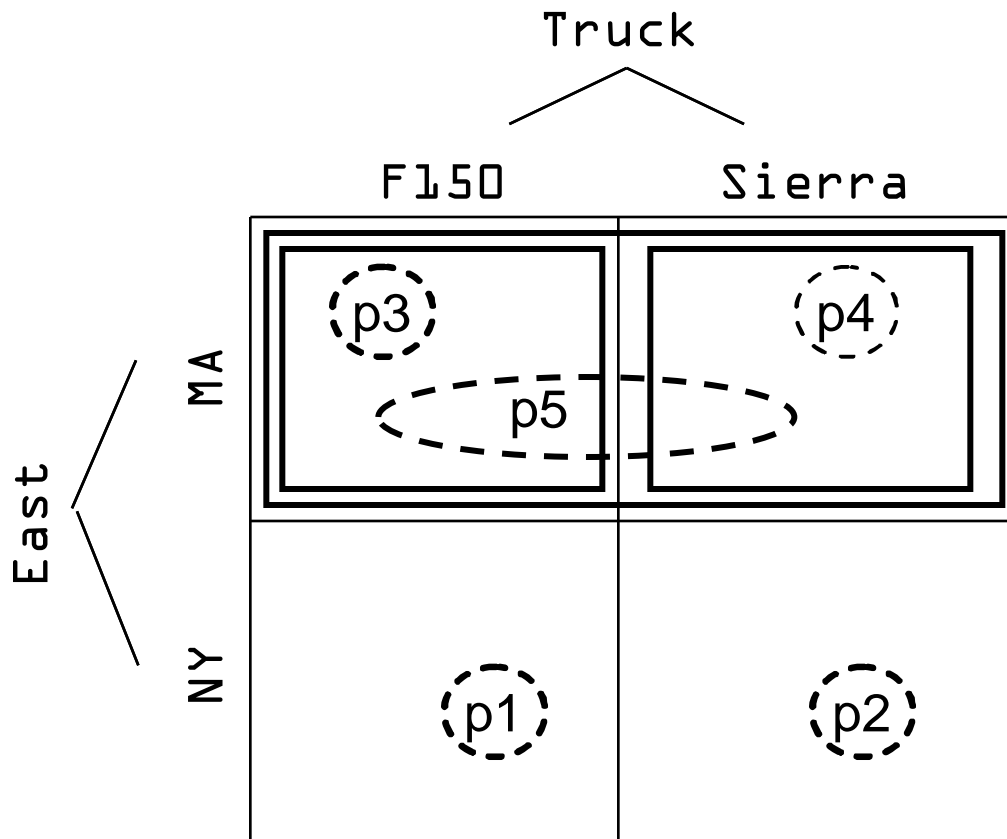
# What is a Good Allocation Policy?

Query: COUNT



- We propose desiderata that enable appropriate definition of query semantics for imprecise data

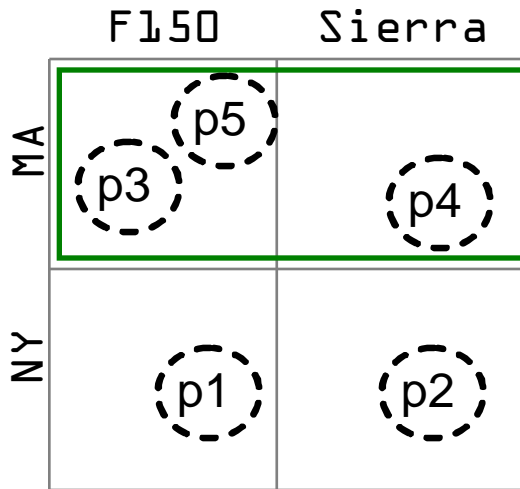
# Desideratum I: Consistency



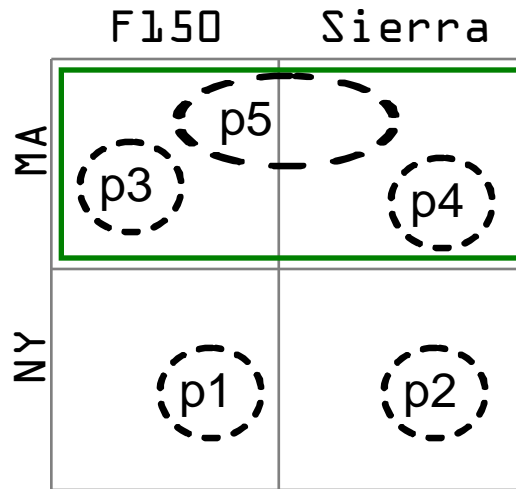
- Consistency specifies the relationship between answers to **related queries** on a **fixed data set**

# Desideratum II: Faithfulness

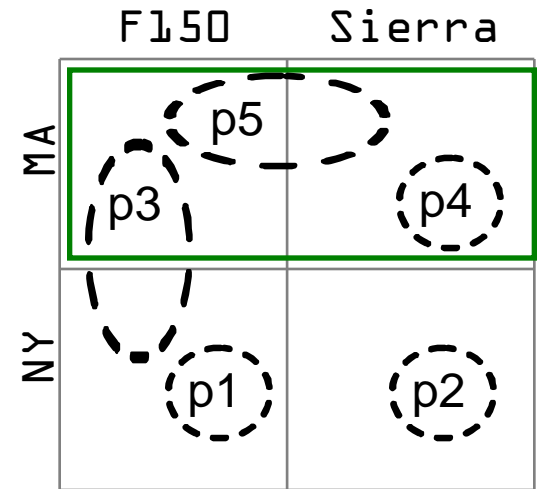
Data Set 1



Data Set 2



Data Set 3

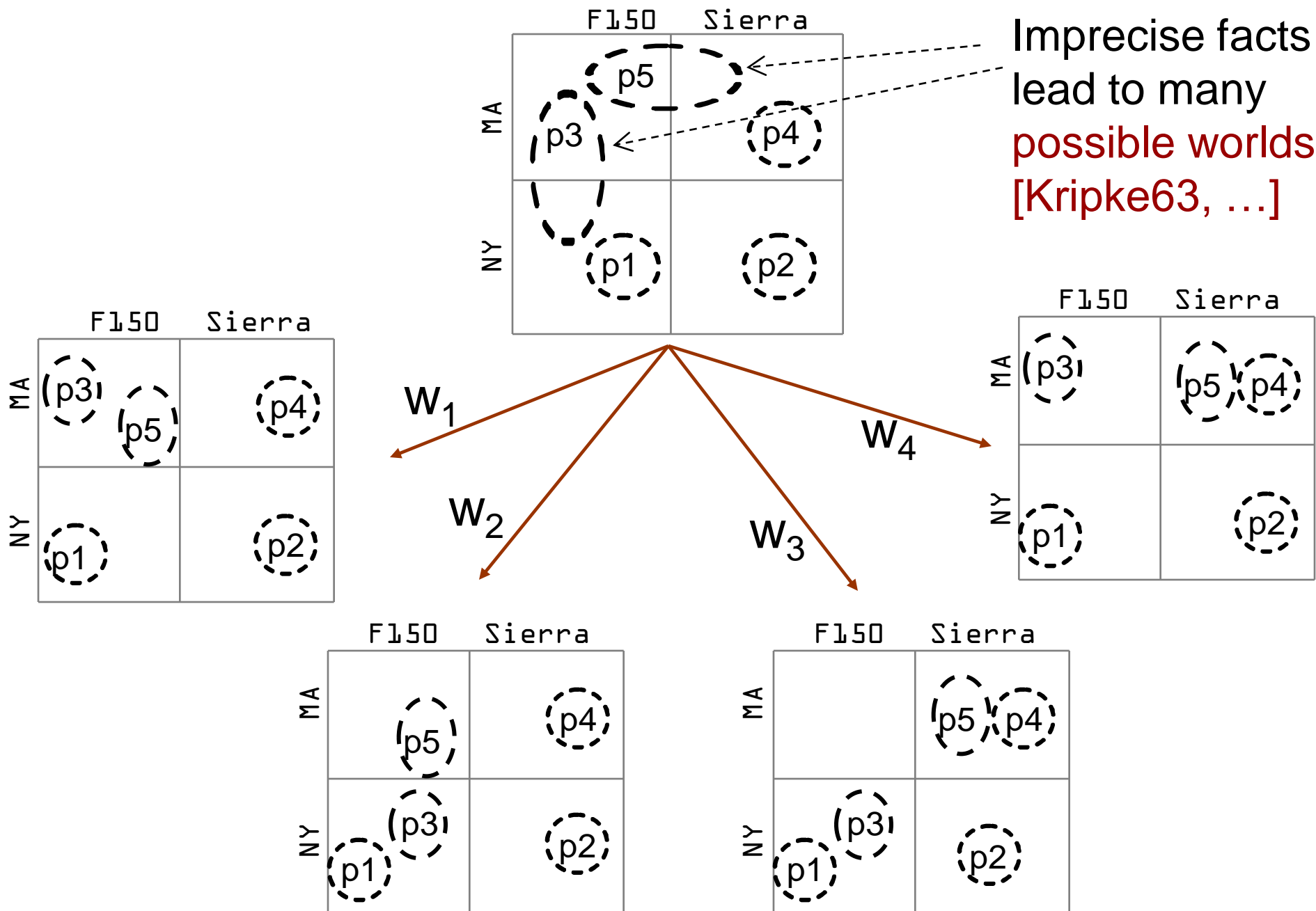


- Faithfulness specifies the relationship between answers to a **fixed query** on **related data sets**

# Results on Query Semantics

- Evaluating queries over extended data model yields expected value of the aggregation operator over all possible worlds
- Efficient query evaluation algorithms available for SUM, COUNT; more expensive dynamic programming algorithm for AVERAGE
  - Consistency and faithfulness for SUM, COUNT are satisfied under appropriate conditions
  - (Bound-)Consistency does not hold for AVERAGE, but holds for  $E(\text{SUM})/E(\text{COUNT})$ 
    - Weak form of faithfulness holds
  - Opinion pooling with LinOP: Similar to AVERAGE

Imprecise facts  
lead to many  
possible worlds  
[Kripke63, ...]



# Query Semantics

- Given all possible worlds together with their probabilities, queries are easily answered using expected values
  - But number of possible worlds is exponential!
- Allocation gives facts weighted assignments to possible completions, leading to an extended version of the data
  - Size increase is linear in number of (completions of) imprecise facts
  - Queries operate over this extended version

# Exploratory Mining: Prediction Cubes

with Beechun Chen, Lei Chen, and Yi Lin  
In VLDB 05; EDAM Project

# The Idea

- Build OLAP data cubes in which cell values represent **decision/prediction behavior**
  - In effect, build a tree for each cell/region in the cube—observe that this is **not** the same as a collection of trees used in an ensemble method!
  - The idea is simple, but it leads to promising data mining tools
  - **Ultimate objective:** Exploratory analysis of the entire space of “data mining choices”
    - Choice of algorithms, data conditioning parameters ...

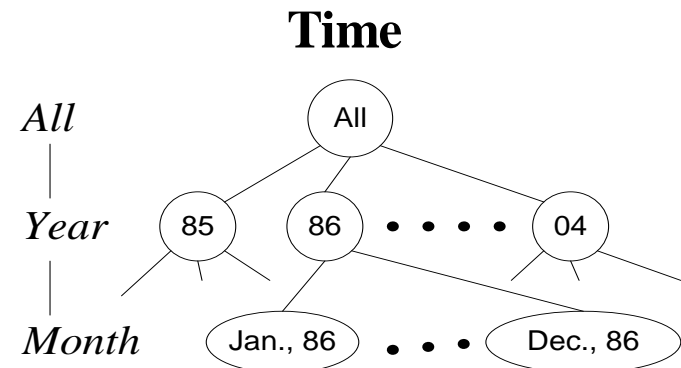
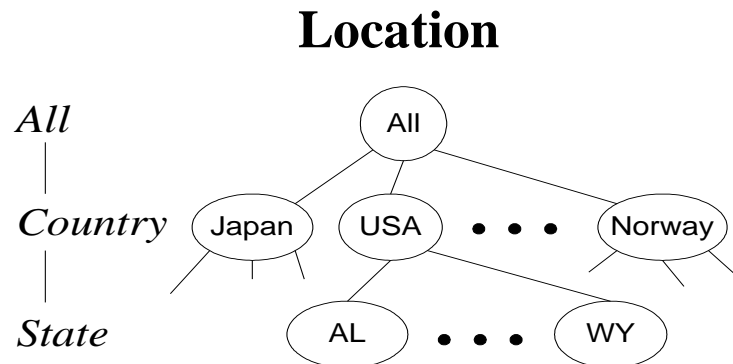


# Example (1/7): Regular OLAP

**Goal:** Look for patterns of unusually high numbers of applications:

**Z:** Dimensions    **Y:** Measure

Location	Time	# of App.
...	...	...
AL, <b>USA</b>	<b>Dec, 04</b>	2
...	...	...
WY, <b>USA</b>	<b>Dec, 04</b>	3



# Example (2/7): Regular OLAP

**Goal:** Look for patterns of unusually high numbers of applications:

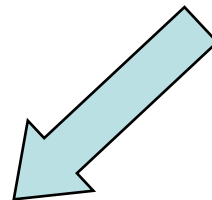
**Z:** Dimensions    **Y:** Measure

Location	Time	# of App.
...	...	...
AL, <b>USA</b>	<b>Dec, 04</b>	2
...	...	...
WY, <b>USA</b>	<b>Dec, 04</b>	3

Coarser regions		04	03	...
	CA	100	90	...
	USA	80	90	...
	...	...	...	...



Roll up



Drill down



	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	30	20	50	25	30	...	...
USA	70	2	8	10	...	...	...
...	...	...	...	...	...	...	...

Cell value: Number of loan applications

		2004			...
		Jan	...	Dec	...
CA	AB	20	15	15	...
	...	5	2	20	...
	YT	5	3	15	...
USA	AL	55	...	...	...
	...	5	...	...	
	WY	10	...	...	...
...	...	...	...	...	...

Finer regions

# Example (3/7): Decision Analysis

**Goal:** Analyze a bank's loan **decision process**  
w.r.t. two dimensions: *Location* and *Time*

## Fact table **D**

**Z:** Dimensions **X:** Predictors **Y:** Class

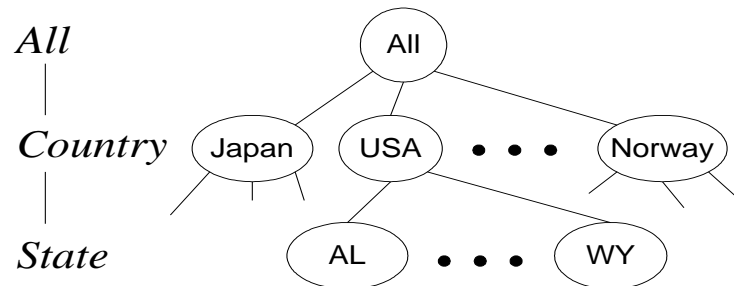
Location	Time	Race	Sex	...	Approval
AL, <b>USA</b>	<b>Dec, 04</b>	White	M	...	Yes
...	...	...	...	...	...
WY, <b>USA</b>	<b>Dec, 04</b>	Black	F	...	No



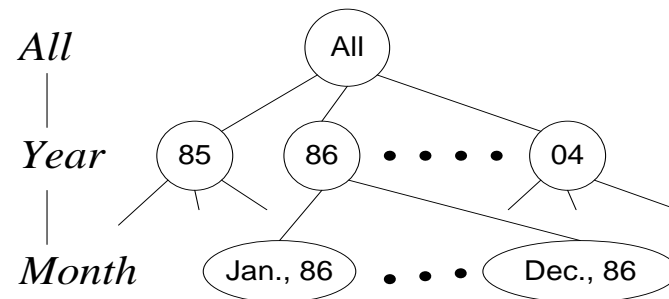
Cube subset

Model  $h(\mathbf{X}, \sigma_{\mathbf{Z}}(\mathbf{D}))$   
E.g., decision tree

*Location*



*Time*



# Example (3/7): Decision Analysis

- Are there branches (and time windows) where approvals were closely tied to sensitive attributes (e.g., race)?
  - Suppose you partitioned the training data by location and time, chose the partition for a given branch and time window, and built a classifier. You could then ask, “Are the predictions of this classifier closely correlated with race?”
- Are there branches and times with decision making reminiscent of 1950s Alabama?
  - Requires comparison of classifiers trained using different subsets of data.

# Example (4/7): Prediction Cubes

	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	0.4	0.8	0.9	0.6	0.8	...	...
USA	0.2	0.3	0.5	...	...	...	...
...	...	...	...	...	...	...	...

Data  $\sigma_{[\text{USA}, \text{Dec 04}]}(\mathbf{D})$

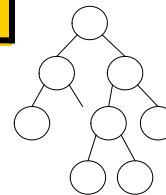
Location	Time	Race	Sex	...	Approval
AL, USA	Dec, 04	White	M	...	Y
...	...	...	...	...	...
WY, USA	Dec, 04	Black	F	...	N

1. Build a model using data from USA in Dec., 1985
2. Evaluate that model

Measure in a cell:

- **Accuracy** of the model
- **Predictiveness** of *Race* measured based on that model
- **Similarity** between that model and a given model

Model  $h(\mathbf{X}, \sigma_{[\text{USA}, \text{Dec 04}]}(\mathbf{D}))$   
E.g., decision tree



# Example (5/7): Model-Similarity

## Given:

- Data table **D**
- Target model  $h_0(\mathbf{X})$
- Test set  $\Delta$  w/o labels

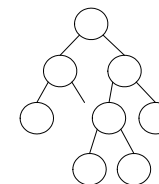
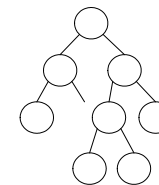
Data table **D**

Location	Time	Race	Sex	...	Approval
AL, <b>USA</b>	<b>Dec, 04</b>	White	M	...	Yes
...	...	...	...	...	...
WY, <b>USA</b>	<b>Dec, 04</b>	Black	F	...	No

	2004			2003			...
	Jan	...	<b>Dec</b>	Jan	...	Dec	...
<b>CA</b>	0.4	0.2	0.3	0.6	0.5	...	...
<b>USA</b>	0.2	0.3	<b>0.9</b>	...	...	...	...
...	...	...	...	...	...	...	...

Level: [Country, Month]

Similarity



$h_0(\mathbf{X})$

Build a model

Race	Sex		
White	F	Yes	Yes
...	...	...	...
Black	M	No	Yes

Test set  $\Delta$

The loan decision process in **USA during Dec 04** was **similar to** a discriminatory decision model

# Example (6/7): Predictiveness

## Given:

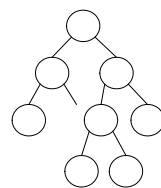
- Data table **D**
- Attributes **V**
- Test set  $\Delta$  w/o labels

	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	...
CA	0.4	0.2	0.3	0.6	0.5	...	...
USA	0.2	0.3	0.9		...	...	...
...	...	...	...	...	...	...	...

Level: [Country, Month]

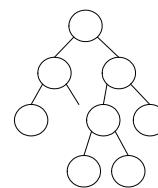
Data table **D**

Location	Time	Race	Sex	...	Approval
AL, USA	Dec, 04	White	M	...	Yes
...	...	...	...	...	...
WY, USA	Dec, 04	Black	F	...	No
...	...	...	...	...	...



$h(X)$

Yes  
No  
.  
Yes



$h(X-V)$

Yes  
No  
.  
No

Predictiveness of **V**

Build models

Race	Sex	...
White	F	...
...	...	...
Black	M	...

Test set  $\Delta$

Race was an important predictor of loan approval decision in **USA during Dec 04**

# Model Accuracy

- A probabilistic view of classifiers: A dataset is a random sample from an underlying pdf  $p^*(\mathbf{X}, Y)$ , and a classifier

$$h(\mathbf{X}; \mathbf{D}) = \operatorname{argmax}_y p^*(Y=y \mid \mathbf{X}=\mathbf{x}, \mathbf{D})$$

- i.e., A classifier approximates the pdf by predicting the “most likely”  $y$  value
- Model Accuracy:
  - $E_{\mathbf{x}, y} [ I( h(\mathbf{x}; \mathbf{D}) = y ) ]$ , where  $(\mathbf{x}, y)$  is drawn from  $p^*(\mathbf{X}, Y \mid \mathbf{D})$ , and  $I(\Psi) = 1$  if the statement  $\Psi$  is true;  $I(\Psi) = 0$ , otherwise
  - In practice, since  $p^*$  is an unknown distribution, we use a set-aside test set or cross-validation to estimate model accuracy.



# Model Similarity

- The prediction similarity between two models,  $h_1(\mathbf{X})$  and  $h_2(\mathbf{X})$ , on test set  $\Delta$  is

$$\frac{1}{|\Delta|} \sum_{\mathbf{x} \in \Delta} I(h_1(\mathbf{x}) = h_2(\mathbf{x}))$$

- The KL-distance between two models,  $h_1(\mathbf{X})$  and  $h_2(\mathbf{X})$ , on test set  $\Delta$  is

$$\frac{1}{|\Delta|} \sum_{\mathbf{x} \in \Delta} \sum_y p_{h_1}(y | x) \log \frac{p_{h_1}(y | x)}{p_{h_2}(y | x)}$$

# Attribute Predictiveness

- Intuition:  $V \subseteq X$  is not predictive if and only if  $V$  is independent of  $Y$  given the other attributes  $X - V$ ; i.e.,

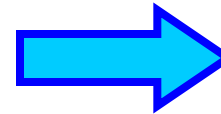
$$p^*(Y | X - V, D) = p^*(Y | X, D)$$

- In practice, we can use the distance between  $h(X; D)$  and  $h(X - V; D)$
- Alternative approach: Test if  $h(X; D)$  is more accurate than  $h(X - V; D)$  (e.g., by using cross-validation to estimate the two model accuracies involved)

# Example (7/7): Prediction Cube

	2004			2003			...
	Jan	...	Dec	Jan	...	Dec	
CA	0.4	0.1	0.3	0.6	0.8	...	...
USA	0.7	0.4	0.3	0.3	...	...	...
...	...	...	...	...	...	...	...

Roll up



	04	03	...
CA	0.3	0.2	...
USA	0.2	0.3	...
...	...	...	...

Cell value: Predictiveness of *Race*



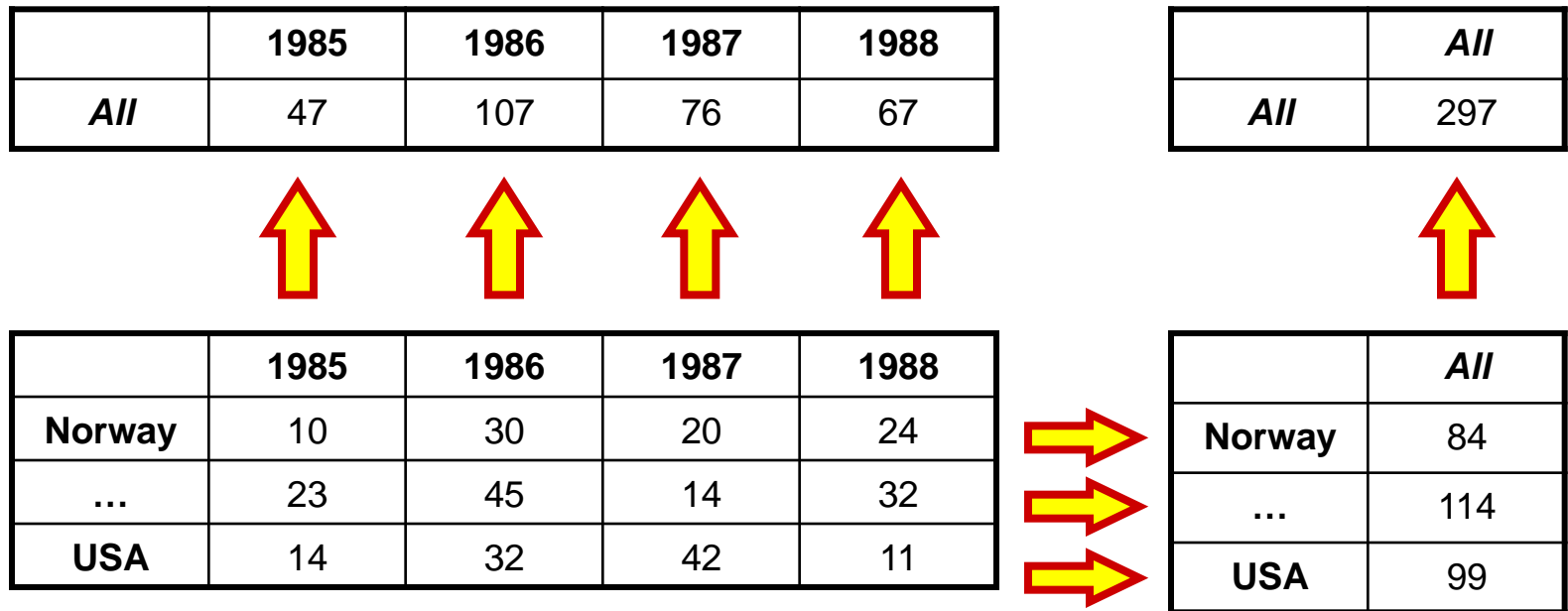
Drill down

		2004			2003			...
		Jan	...	Dec	Jan	...	Dec	
CA	AB	0.4	0.2	0.1	0.1	0.2	...	...
	...	0.1	0.1	0.3	0.3	...	...	...
	YT	0.3	0.2	0.1	0.2	...	...	...
USA	AL	0.2	0.1	0.2	...	...	...	...
	...	0.3	0.1	0.1	...	...	...	...
	WY	0.9	0.7	0.8	...	...	...	...
...	...	...	...	...	...	...	...	...

# Efficient Computation

- Reduce prediction cube computation to data cube computation
  - Represent a data-mining model as a distributive or algebraic (bottom-up computable) aggregate function, so that data-cube techniques can be directly applied

# Bottom-Up Data Cube Computation



Cell Values: Numbers of loan applications

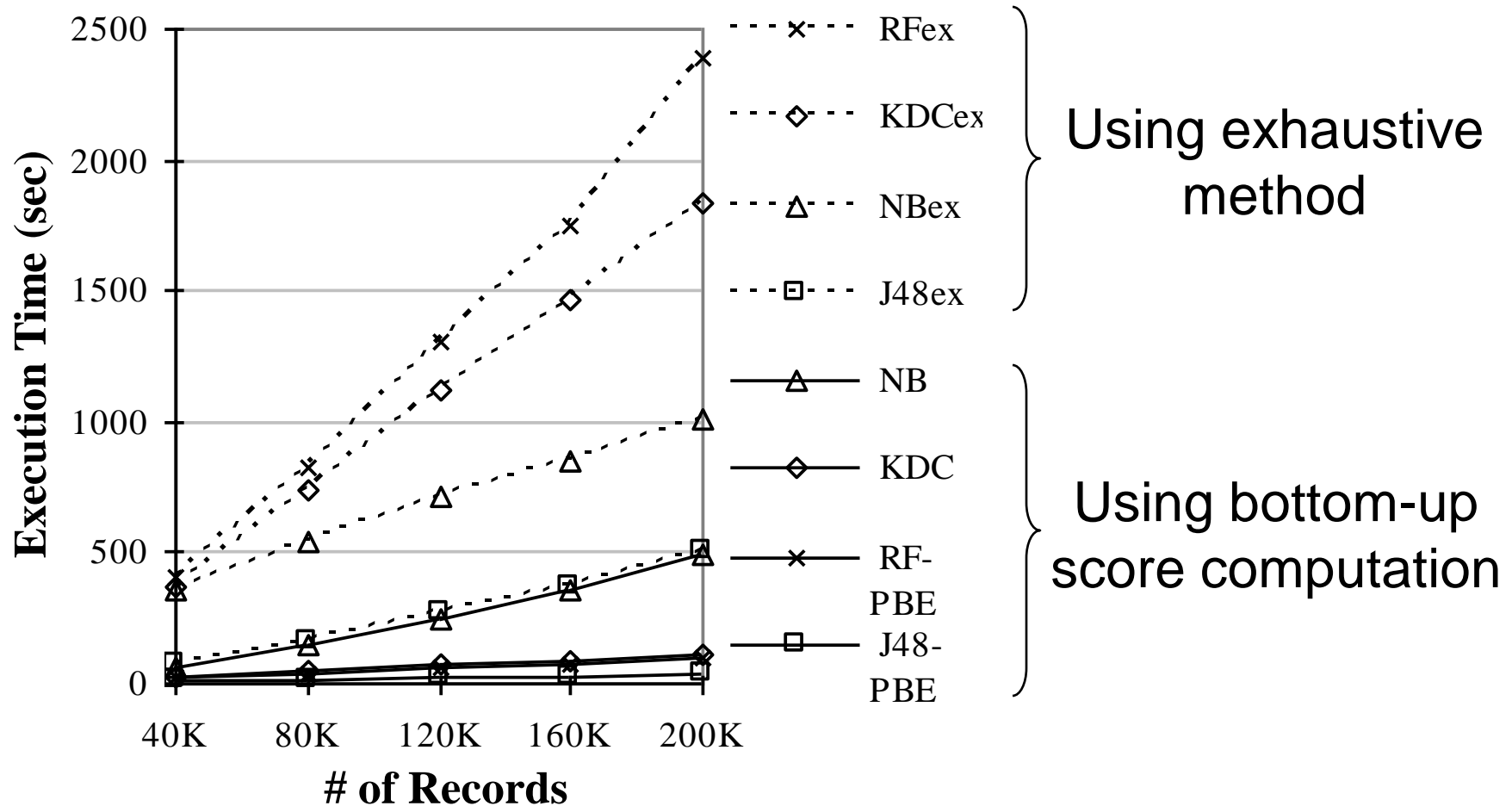
# Scoring Function

- Represent a model as a function of sets
- Conceptually, a machine-learning model  $h(\mathbf{X}; \sigma_{\mathbf{Z}}(\mathbf{D}))$  is a scoring function  $\text{Score}(y, \mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D}))$  that gives each class  $y$  a score on test example  $\mathbf{x}$ 
  - $h(\mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D})) = \operatorname{argmax}_y \text{Score}(y, \mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D}))$
  - $\text{Score}(y, \mathbf{x}; \sigma_{\mathbf{Z}}(\mathbf{D})) \approx p(y \mid \mathbf{x}, \sigma_{\mathbf{Z}}(\mathbf{D}))$
  - $\sigma_{\mathbf{Z}}(\mathbf{D})$ : The set of training examples (a cube subset of  $\mathbf{D}$ )

# Machine-Learning Models

- Naïve Bayes:
  - Scoring function: algebraic
- Kernel-density-based classifier:
  - Scoring function: distributive
- Decision tree, random forest:
  - Neither distributive, nor algebraic
- PBE: Probability-based ensemble (new)
  - To make any machine-learning model distributive
  - Approximation

# Efficiency Comparison





# Bellwether Analysis: Global Aggregates from Local Regions

with Beechun Chen, Jude Shavlik, and Pradeep Tamma  
In VLDB 06

# Motivating Example

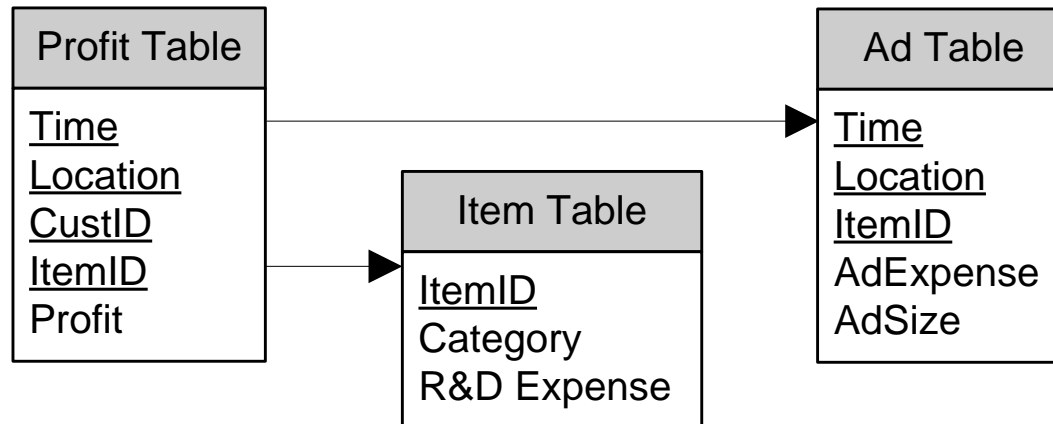
- A company wants to predict the first year worldwide profit of a new item (e.g., a new movie)
  - By looking at **features and profits of previous (similar) movies**, we predict **expected total profit** (1-year US sales) **for new movie**
    - Wait a year and write a query! If you can't wait, stay awake ...
  - The most predictive “features” may be based on sales data gathered by releasing the new movie in many “regions” (different locations over different time periods).
    - Example **“region-based” features**: 1<sup>st</sup> week sales in Peoria, week-to-week sales growth in Wisconsin, etc.
    - Gathering this data has a **cost** (e.g., marketing expenses, waiting time)
- **Problem statement:** Find the most predictive region features that can be obtained within a given “cost budget”

# Key Ideas

- Large datasets are rarely labeled with the targets that we wish to learn to predict
  - But for the tasks we address, we can readily use OLAP queries to generate features (e.g., 1<sup>st</sup> week sales in Peoria) and even **targets** (e.g., profit) for mining
- We use data-mining models as building blocks in the mining process, rather than thinking of them as the end result
  - The central problem is to find data subsets (**“bellwether regions”**) that lead to predictive features which can be gathered at low cost for a new case

# Motivating Example

- A company wants to predict the first year's worldwide profit for a new item, by using its historical database
- Database Schema:

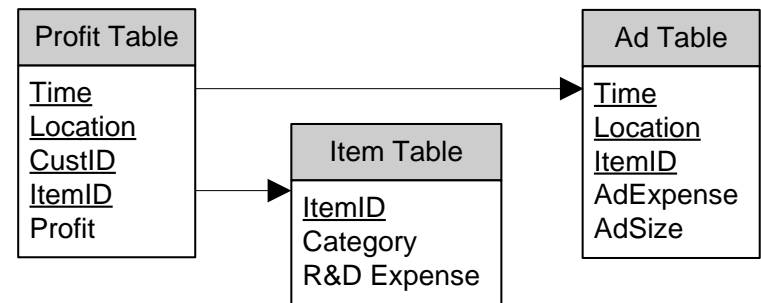


- The combination of the underlined attributes forms a key

# A Straightforward Approach

- Build a regression model to predict item profit

By joining and aggregating tables in the **historical database** we can create a **training set**:



Item-table features			Target
ItemID	Category	R&D Expense	Profit
1	Laptop	500K	12,000K
2	Desktop	100K	8,000K
...	...	...	...

An Example regression model:  
$$Profit = \beta_0 + \beta_1 Laptop + \beta_2 Desktop + \beta_3 RdExpense$$

- There is much room for accuracy improvement!

# Using Regional Features

- Example region: [1<sup>st</sup> week, HK]
- **Regional features:**
  - **Regional Profit:** The 1<sup>st</sup> week profit in HK
  - **Regional Ad Expense:** The 1<sup>st</sup> week ad expense in HK
- A possibly more accurate model:

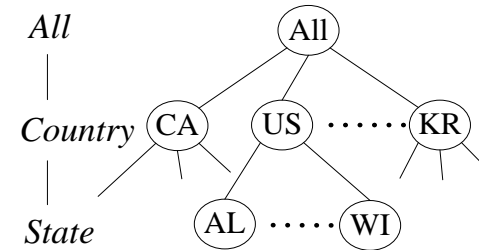
$$Profit_{[1yr, All]} = \beta_0 + \beta_1 Laptop + \beta_2 Desktop + \beta_3 RdExpense + \beta_4 \textbf{Profit}_{[1wk, KR]} + \beta_5 \textbf{AdExpense}_{[1wk, KR]}$$

- **Problem:** Which region should we use?
  - The smallest region that improves the accuracy the most
  - We give each candidate region a cost
  - The most “cost-effective” region is the **bellwether region**

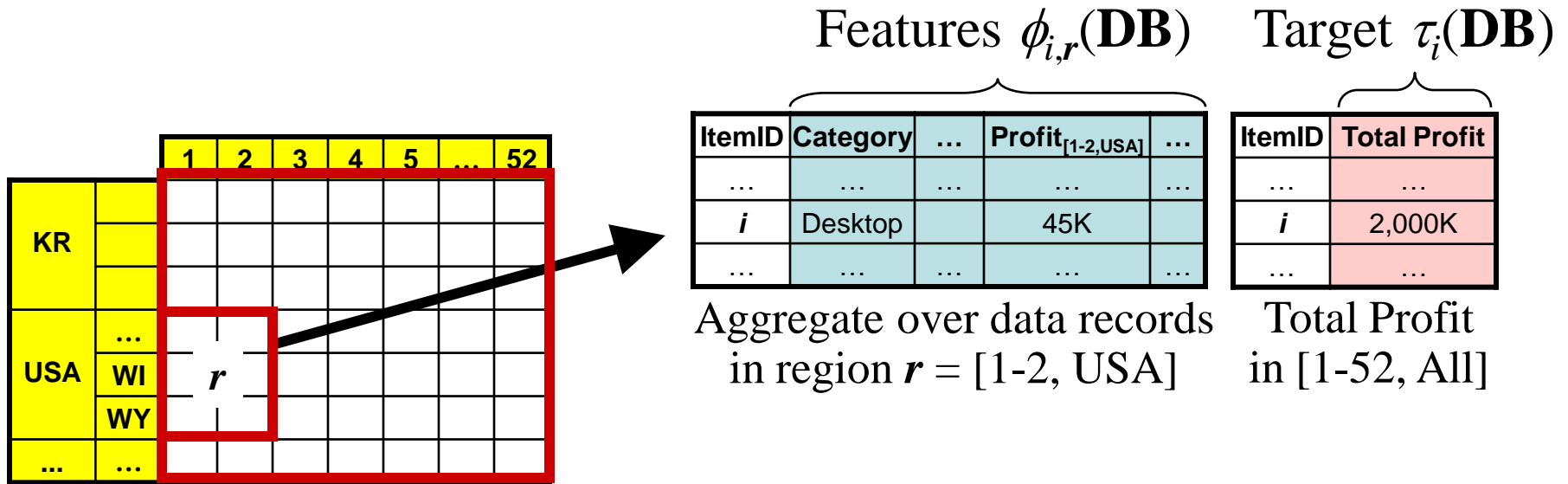
# Basic Bellwether Problem

- Historical database: **DB**
- Training item set: **I**
- Candidate region set: **R**
  - E.g.,  $\{ [1-n \text{ week}, \text{Location}] \}$
- Target generation query:  $\tau_i(\mathbf{DB})$  returns the target value of item  $i \in \mathbf{I}$ 
  - E.g.,  $\alpha_{\text{sum(Profit)}} \sigma_{i, [1-52, \text{All}]} \text{ProfitTable}$
- Feature generation query:  $\phi_{i,r}(\mathbf{DB})$ ,  $i \in \mathbf{I}_r$  and  $r \in \mathbf{R}$ 
  - $\mathbf{I}_r$ : The set of items in region  $r$
  - E.g.,  $[ \text{Category}_i, \text{RdExpense}_i, \text{Profit}_{i, [1-n, \text{Loc}]}, \text{AdExpense}_{i, [1-n, \text{Loc}]} ]$
- Cost query:  $\kappa_r(\mathbf{DB})$ ,  $r \in \mathbf{R}$ , the cost of collecting data from  $r$
- Predictive model:  $h_r(\mathbf{x})$ ,  $r \in \mathbf{R}$ , trained on  $\{(\phi_{i,r}(\mathbf{DB}), \tau_i(\mathbf{DB})) : i \in \mathbf{I}_r\}$ 
  - E.g., linear regression model

Location domain hierarchy



# Basic Bellwether Problem



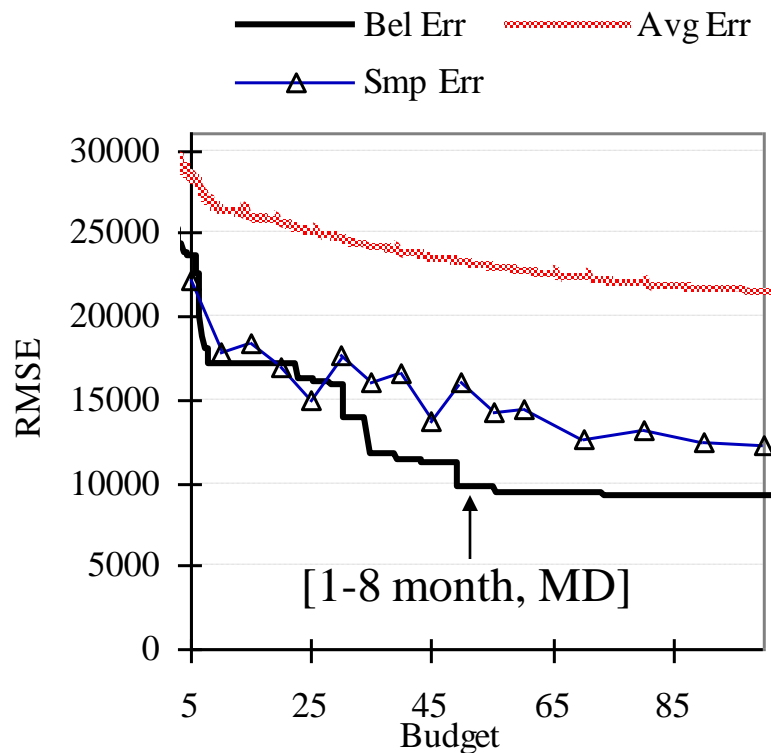
For each region  $r$ , build a predictive model  $h_r(\mathbf{x})$ ; and then choose **bellwether region**:

- $\text{Coverage}(r) \equiv$  fraction of all items in region  $\geq$  minimum coverage support
- $\text{Cost}(r, \mathbf{DB}) \leq \text{cost threshold}$
- $\text{Error}(h_r)$  is minimized



# Experiment on a Mail Order Dataset

## Error-vs-Budget Plot

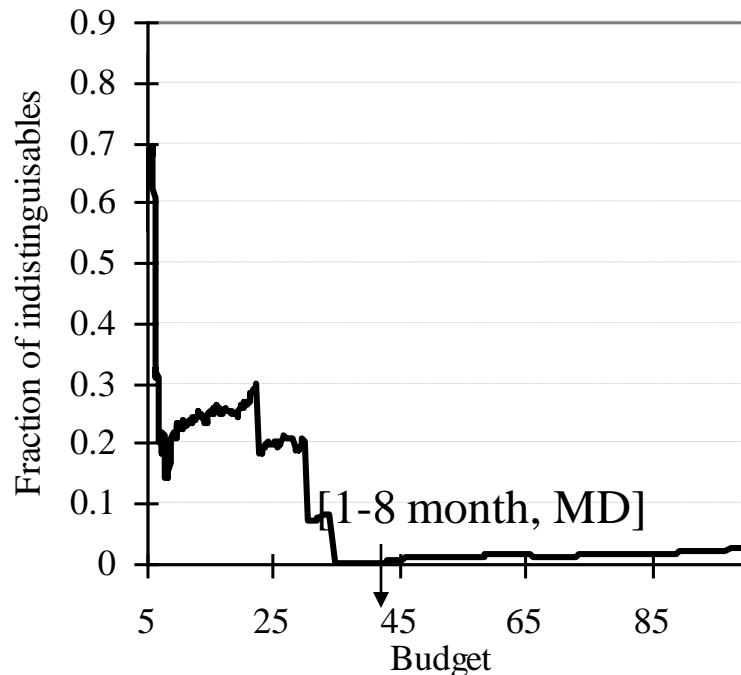


- **Bel Err:** The error of the bellwether region found using a given budget
- **Avg Err:** The average error of all the cube regions with costs under a given budget
- **Smp Err:** The error of a set of randomly sampled (non-cube) regions with costs under a given budget

(RMSE: Root Mean Square Error)

# Experiment on a Mail Order Dataset

## Uniqueness Plot

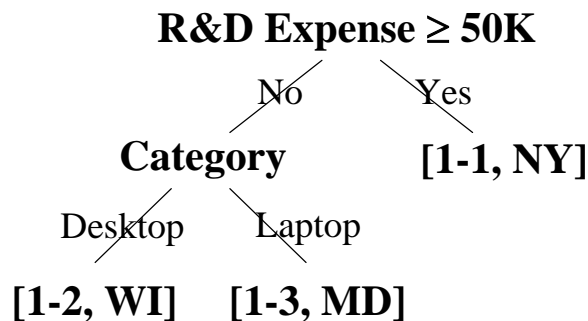


- **Y-axis:** Fraction of regions that are as good as the bellwether region
  - The fraction of regions that satisfy the constraints and have errors within the 99% confidence interval of the error of the bellwether region
- We have 99% confidence that that [1-8 month, MD] is a quite unusual bellwether region

# Subset-Based Bellwether Prediction

- **Motivation:** Different subsets of items may have different bellwether regions
  - E.g., The bellwether region for laptops may be different from the bellwether region for clothes
- Two approaches:

## Bellwether Tree



## Bellwether Cube

		R&D Expenses			
		Low	Medium	High	
Category	Software	OS	[1-3,CA]	[1-1,NY]	[1-2,CA]
		...	...	...	...
	Hardware	Laptop	[1-4,MD]	[1-1, NY]	[1-3,WI]
		...	...	...	...
	...	...	...	...	...

# Conclusions

# Related Work: Building models on OLAP Results

- Multi-dimensional regression [Chen, VLDB 02]
  - Goal: Detect changes of trends
  - Build linear regression models for cube cells
- Step-by-step regression in stream cubes [Liu, PAKDD 03]
- Loglinear-based quasi cubes [Barbara, J. IIS 01]
  - Use loglinear model to approximately compress dense regions of a data cube
- NetCube [Margaritis, VLDB 01]
  - Build Bayes Net on the entire dataset of approximate answer count queries

# Related Work (Contd.)

- Cubegrades [Imielinski, J. DMKD 02]
  - Extend cubes with ideas from association rules
  - How does the measure change when we rollup or drill down?
- Constrained gradients [Dong, VLDB 01]
  - Find pairs of similar cell characteristics associated with big changes in measure
- User-cognizant multidimensional analysis [Sarawagi, VLDBJ 01]
  - Help users find the most informative unvisited regions in a data cube using max entropy principle
- Multi-Structural DBs [Fagin et al., PODS 05, VLDB 05]

# Take-Home Messages

- Promising exploratory data analysis paradigm:
  - Can use **models** to identify interesting subsets
  - Concentrate only on subsets in **cube space**
    - Those are meaningful subsets, tractable
  - **Precompute** results and provide the users with an **interactive** tool
- A simple way to plug “something” into cube-style analysis:
  - Try to describe/approximate “something” by a distributive or algebraic function

# Big Picture

- **Why stop with decision behavior?** Can apply to other kinds of analyses too
- **Why stop at browsing?** Can mine prediction cubes in their own right
- **Exploratory analysis of mining space:**
  - Dimension attributes can be parameters related to algorithm, data conditioning, etc.
  - Tractable evaluation is a challenge:
    - Large number of “dimensions”, real-valued dimension attributes, difficulties in compositional evaluation
    - Active learning for experiment design, extending compositional methods



1.	How Many Phases involved in Data mining and Knowledge discovery process found by Usama Fayyad and Evangelos simoudis?	4	5	6	7	5
2.	_____ is the automation of a learning process and learning is tantamount to the construction of the rules based on observation of environmental states and transition.	Self Learning	Automatic Learning	Machine Learning	Artificial Learning	Machine Learning
3.	_____ is the inference of information from data and inductive learning is the model building process where the environment	Induction	Self	Machine	Supervised	Induction
4.	_____ is the learning from examples.	Induction Learning	Machine Learning	Un-supervised Learning	Supervised Learning	Supervised Learning
5.	KDD Stands for_____	Knowledge Discovery in Database	Knowledge Discovery in Data	Knowledge Discovered in Database	Knowledge Discovered in Data	Knowledge Discovery in Database
6.	A _____ Computed can generate program itself, enabling it to carry out new tasks.	Self	Self Studying	Self knowing	Self Learning	Self Learning
7.	How many stages involved in empirical cycle of scientific research.	1	2	3	4	4
8.	_____ Content is closely related to statistical significance and transparency.	Data	Information	Record	Field	Information
9.	_____ is an essential process where intelligent methods are applied to extract data patterns.	Data warehousing	Data mining	Text mining	Data Selection	Data mining
10.	Which of the following is not a data mining functionality?	Characterization and Discrimination	Classification and regression	Selection and interpretation	Clustering and Analysis	Selection and interpretation

11.	_____ is a summarization of the general characteristics or features of a target class of data.	Data characterization	Data Classification	Data Discrimination	Data Selection	Data characterization
12.	_____ is a comparison of the general features of the target class data object against the general features of objects from one or multiple contrasting classes.	Data characterization	Data Classification	Data Discrimination	Data Selection	Data Discrimination
13.	_____ predicts future trends and behavior, allowing business managers to make proactive knowledge driven decision.	Data warehouse	Data mining	Data marts	Meta data	Data mining
14.	Query tool is meant for _____	Data Acquisition	Information delivery	Information exchange	Communication	Data Acquisition
15.	Classification rules are extracted from _____	Root node	Decision Tree	Siblings	Branches	Decision Tree
16.	The first international conference on KDD was held in the year _____	1996	1997	1995	1994	1995
17.	The Power of Self Learning System lies in _____	Cost	Speed	Accuracy	Simplicity	Accuracy
18.	Which of the following is not a data mining metric?	Space complexity	Usefulness	Space/Time	Accuracy	Space complexity
19.	Data that are not of interest to the data mining task is called as _____	Missing data	Changing data	Irrelevant data	Noisy data	Irrelevant data
20.	Research on mining multi – types of data is termed as _____ data	Graphics	Multimedia	Meta	digital	Multimedia
21.	_____ is a process of extracting hidden trends within a data warehouse.	Data Purging	Data warehousing	Data mining	Data cube	Data mining
22.	_____ is merely extracting data from different sources, cleaning the data and storing it in the warehouse.	Data Purging	Data warehousing	Data mining	Data cube	Data warehousing

23.	The process of cleaning junk data is termed as _____	data warehousing	Data Purging	Data mining	Data cube	Data Purging
24.	A _____ stores data in a summarized version which helps in a faster analysis of data	Data Cube	Data Purging	Data mining	Data cube	Data Cube
25.	_____ – categorized by short online transactions	OLTP	BI	OT	OLTP	OLTP
26.	_____ – Low volumes of transactions are categorized by OLAP	OLAP	BI	OLAP	OT	OLAP
27.	_____ helps analysts in making faster business decisions which increases revenue with lower costs.	Data warehousing	Data mining	Data purging	Data cube	Data mining
28.	_____ This stage involves preparation and collection of data	Transformation	Exploration	loading	transfer	Exploration
29.	_____ This stage involves choosing the best model based on their predictive performance.	Validation	Model building	Model building and validation	none	Model building and validation
30.	_____ can be considered as defined or finite data.	data	information	Discreet data	discreet information	Discreet data
31.	_____ can be considered as data which changes continuously and in an ordered fashion.	interval data	dynamic data	Continuous data	ratio data	Continuous data
32.	_____ in Data mining help the different algorithms in decision making or pattern matching.	interval	scale	mining	Models	Models
33.	Using _____, one can forecast the business needs.	data warehousing	Data mining	Data purging	data model	Data mining
34.	How many stages involved in data mining?	1	2	3	5	3
35.	A _____ is a tree in which every node is either a leaf node or a decision node.	fact table	disk load	mining	Decision tree	Decision tree

36.	Naive Bays Algorithm is used to generate _____ models.	extraction	mining	loading	transformation	mining
37.	_____ is used to group sets of data with similar characteristics also called as clusters.	algorithm	Clustering Algorithm	stratified algorithm	time series algorithm	Clustering Algorithm
38.	_____ can be used to predict continuous values of data.	Time series algorithm	Clustering Algorithm	stratified algorithm	association algorithm	Time series algorithm
39.	_____ is used for recommendation engine that is based on a market based analysis.	Time series algorithm	Association Algorithm	stratified algorithm	clustering algorithm	Association Algorithm
40.	_____ algorithm collects similar or related paths, sequences of data containing events.	Stratified	sequence	modeling	Sequence clustering algorithm	Sequence clustering algorithm
41.	_____ is used to examine or explore the data using queries	data warehousing	Data mining	Data purging	data loading	Data mining
42.	_____ extension is based on the syntax of SQL.	data warehousing	Data Purging	Data mining	data loading	Data mining
43.	A _____ extension can be used to slice the data the source cube in the order as discovered by data mining.	data warehousing	Data Purging	Data mining	data pruning	Data mining
44.	A tree is pruned by halting its construction early is called _____	data warehousing	Data Purging	Data mining	Pre Pruning	Pre Pruning
45.	_____ variables are continuous measurements of linear scale	ratio scale	Interval scale	ordinal scale	scale	Interval scale
46.	Statistical Information Grid is called as _____	string	integer	mining	Sting	Sting
47.	Density Based Spatial Clustering of Application Noise is called as _____	DBSCSN.	DBSSCAN.	DBSCAN.	DSSCAN.	DBSCAN.
48.	Chameleon is another hierarchical clustering method that uses _____ modeling.	dynamic	continuous	stratified	cluster modeling	dynamic

49.	ODS Stands for _____	operation disk loading	operation data store	operation disk set	Operational Data Store	Operational Data Store
50.	_____ is the application of data mining methods to spatial data.	Spatial data mining	spatial data mining	spatial disk mining	spatial disk modeling	Spatial data mining
51.	_____ is an approach that is used to remove the nonsystematic behaviors found in time series.	Stratified	clustering	Smoothing	modeling	Smoothing
52.	Indexes of _____ Server are similar to the indexes in books.	PQL	TQL	RQL	SQL	SQL
53.	_____ is a type of organizing the tables such that we can retrieve the result from the database easily and fastly in the warehouse environment.	Star Schema	snow flake schema	database schema	star flake schema	Star Schema
54.	A _____ is the one which is used when updating a warehouse.	fact table	dynamic table	lookup table	continuous table	lookup table
55.	A _____ is a set of attribute values over a period of time.	interval series	dynamic series	continuous series	time series	time series
56.	_____, each dimension has a primary dimension table, to which one or more additional dimensions can join	Snow flake schema	star schema	database schema	star flake schema	Snow flake schema
57.	ETL stands for _____	Executing, transferring and load	Extraction, transformation and Loading	Extract, Transfer and Load	Executing, Transfer and Labeling	Extraction, transformation and Loading
58.	_____ Helps analysts in making faster business decisions which increases revenue with lower costs.	Data warehousing	Data Purging	Data mining	data loading	Data mining
59.	_____ helps to understand, explore and identify patterns of data.	Data mining	Data Purging	data warehousing	data loading	Data mining
60.	_____ automates process of finding predictive information in large databases.	Data mining	Data Purging	data warehousing	Data mining	Data mining



1.	_____ is the learning from observation and discovery.	Induction Learning	Machine Learning	Un-supervised Learning	Supervised Learning	Un-supervised Learning
2.	_____ is the data cleansing stage where certain information is removed.	Selection	Prepossessing	Transformation	Data Mining	Prepossessing
3.	Knowledge Discovery Process was formalized in _____	1973	1987	1989	1995	1989
4.	Information can be converted into _____ about historical patterns and future trends.	Knowledge	Data	Record	Field	Knowledge
5.	Once you have collected the data, the next stage is _____	Loading	Extracting	Cleaning	Transferring	Cleaning
6.	How many golden rules are there in KDD Environment?	10	11	12	13	10
7.	How many stages involved in Reporting stage?	1	2	3	4	2
8.	Learning tasks can be divided in to _____ areas	1	2	3	4	3

9.	Strategic value of data mining is_____	Case – sensitive	Work – Sensitive	Time – Sensitive	Technical – Sensitive	Time – Sensitive
10.	_____ is the process of finding a model that describes and distinguishes data classes or concepts.	Data characterization	Data Classification	Data Discrimination	Data Selection	Data Classification
11.	The Full form of KDD is_____	Knowledge Database Development	Knowledge Discovery Database	Knowledge Data Development	Knowledge Data Definition	Knowledge Discovery Database
12.	The output of KDD is_____	Data	Information	Query	Useful information	Useful information
13.	Which of the following is not other name of Data Mining?	Exploratory Data Analysis	Data driven Discovery	Deductive Learning	Data Analytical Model	Data Analytical Model
14.	_____ is a input to KDD	Data	information	Query	Process	Data
15.	The KDD Process consist of _____ Steps	3	4	5	6	5
16.	Various visualization techniques are used in _____ step of KDD	Selection	Transformation	Data Mining	Interpretation	Interpretation



17.	_____ Mining is concerned with discovering the model underlying the link structure of the web.	Data Structure	Web Structure	Text Structure	Image Structure	Web Structure
18.	In Web mining, _____ is used to find natural grouping of users, pages etc	Clustering	Associations	Sequential Analysis	Classification	Clustering
19.	In Web mining, _____ is used to know the order in which URLs tend to be accessed.	Clustering	Associations	Sequential Analysis	Classification	Sequential Analysis
20.	In web mining, _____ is used to know which URL tend to be requested together.	Clustering	Associations	Sequential Analysis	Classification	Associations
21.	Data mining in the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data	Data purging	Data warehousing	Data mining	data lose	Data mining

22.	_____ the extraction of the hidden predictive information from large databases is a powerful new technology with great potential to analyze important information in the data warehouse.	Data mining	Data warehousing	Data mining	data lose	Data mining
23.	Knowledge Discovery in Database (KDD) was formalized in _____	data formalized	data structured	data unstructured	data unformulated	data formalized
24.	How many stages involved in KDD?	6	7	8	9	7
25.	The problem of finding hidden structure in unlabeled data is called	Supervised learning	Unsupervised learning	Reinforcement learning	machine learning	Unsupervised learning
26.	Task of inferring a model from labeled training data is called _____	Unsupervised learning	Supervised learning	Reinforcement learning	machine learning	Supervised learning

27.	Some telecommunication company wants to segment their customers into distinct groups in order to send appropriate subscription offers, this is an example of ____	Supervised learning	Data extraction	Serration	Unsupervised learning	Unsupervised learning
28.	Self-organizing maps are an example of ____	Unsupervised learning	Supervised learning	Reinforcement learning	Missing data imputation	Unsupervised learning
29.	You are given data about seismic activity in Japan, and you want to predict a magnitude of the next earthquake, this is in an example of ____	Supervised learning	Unsupervised learning	Serration	Dimensionality reduction	Supervised learning
30.	Assume you want to perform supervised learning and to predict number of newborns according to size of storks' population , it is an example of ____	Classification	Regression	blustering	Structural equation modeling	Regression
31.	Discriminating between spam and ham e-mails is a	True	False	Partially true	Partially false	True

	classification task, true or false?					
32.	In the example of predicting number of babies based on storks' population size, number of babies is ____	outcome	feature	attribute	observation	outcome
33.	It may be better to avoid the metric of ROC curve as it can suffer from accuracy paradox.	True	False	Partially true	Partially false	False
34.	Which of the following is not involve in data mining?	Knowledge extraction	Data archaeology	Data exploration	Data transformation	Data transformation
35.	Which is the right approach of Data Mining?	Infrastructure, exploration, analysis, interpretation, exploitation	Infrastructure, exploration, analysis, exploitation, interpretation	Infrastructure, analysis, exploration, interpretation, exploitation	Infrastructure, analysis, exploration, exploitation, interpretation	Infrastructure, exploration, analysis, interpretation, exploitation
36.	Which of the following issue is considered before investing in Data Mining?	Functionality	Vendor consideration	compatibility	All	All
37.	Adaptive system management is _____	It uses machine-learning techniques. Here	Computational procedure that takes some value as input	Siene of making machines performs tasks that would require	None	It uses machine-learning

		program can learn from past experience and adapt themselves to new situations	and produces some value as output.	intelligence when performed by humans		techniques. Here program can learn from past experience and adapt themselves to new situations
38.	Bayesian classifiers is ____	A class of learning algorithm that tries to find an optimum classification of a set of examples using the probabilistic theory.	Any mechanism employed by a learning system to constrain the search space of a hypothesis	An approach to the design of learning algorithms that is inspired by the fact that when people encounter new situations, they often explain them by reference to familiar experiences, adapting the explanations to fit the new situation	None	A class of learning algorithm that tries to find an optimum classification of a set of examples using the probabilistic theory.
39.	Algorithm is ____	It uses machine-learning techniques. Here	Computational procedure that takes some value as input	Seine of making machines performs tasks that would require	All	Computational procedure that

		program can learn from past experience and adapt themselves to new situations	and produces some value as output	intelligence when performed by humans		takes some value as input and produces some value as output
40.	Bias is ____	A class of learning algorithm that tries to find an optimum classification of a set of examples using the probabilistic theory	Any mechanism employed by a learning system to constrain the search space of a hypothesis	An approach to the design of learning algorithms that is inspired by the fact that when people encounter new situations, they often explain them by reference to familiar experiences, adapting the explanations to fit the new situation	Noe	Any mechanism employed by a learning system to constrain the search space of a hypothesis
41.	Background knowledge referred to _____	Additional acquaintance used by a learning algorithm to facilitate the learning process	A neural network that makes use of a hidden layer	It is a form of automate learning	both A neural network that makes use of a hidden layer and It is a form of	Additional acquaintance used by a learning algorithm to facilitate the

					automate learning	learning process
42.	Case-based learning is ____	A. A class of learning algorithm that tries to find an optimum classification of a set of examples using the probabilistic theory.	B Any mechanism employed by a learning system to constrain the search space of a hypothesis	An approach to the design of learning algorithms that is inspired by the fact that when people encounter new situations, they often explain them by reference to familiar experiences, adapting the explanations to fit the new situation.	All	An approach to the design of learning algorithms that is inspired by the fact that when people encounter new situations, they often explain them by reference to familiar experiences, adapting the explanations to fit the new situation.
43.	Classification is ____	A subdivision of a set of examples	A measure of the accuracy, of the	The task of assigning a classification to a set of	All	A subdivision of a set of

		into a number of classes	classification of a concept that is given by a certain theory	examples		examples into a number of classes
44.	Binary attribute are ____	This takes only two values. In general, these values will be 0 and 1 and .they can be coded as one bit	The natural environment of a certain species	Systems that can be used without knowledge of internal operations	All	This takes only two values. In general, these values will be 0 and 1 and .they can be coded as one bit
45.	Classification accuracy is ____	A subdivision of a set of examples into a number of classes	Measure of the accuracy, of the classification of a concept that is given by a certain theory	The task of assigning a classification to a set of examples	All	Measure of the accuracy, of the classification of a concept that is given by a certain theory



46.	Biotope are ____	This takes only two values. In general, these values will be 0 and 1 and they can be coded as one bit.	The natural environment of a retain species	Systems that can be used without knowledge of internal operations	both The natural environment of a retain species and ystems that can be used without knowledge of internal operations	The natural environment of a certain species
47.	Cluster is ____	Group of similar objects that differ significantly from other objects	Operations on a database to transform or simplify data in order to prepare it for a machine-learning algorithm	Symbolic representation of fats or ideas from which information and potentially be extracted	both Operations on a database to transform or simplify data in order to prepare it for a machine-learning algorithm and Symbolic representation	Group of similar objects that differ significantly from other objects

					of facts or ideas from which information and potentially be extracted	
48.	Black boxes are ____	This takes only two values. In general, these values will be 0 and 1 and they can be coded as one bit.	The natural environment of a certain species	Systems that can be used without knowledge of internal operations	both The natural environment of a certain species and Systems that can be used without knowledge of internal operations	Systems that can be used without knowledge of internal operations
49.	A definition of a concept is if it recognizes all the instances of that concept ____	Complete	Consistent	instant	Partial complete partial consisten	Complete
50.	Data mining is	The actual discovery phase of	The stage of selecting the right data for a	A subject-oriented integrated time variant	All are true	The actual discovery

		a knowledge discovery process	KDD process	non-volatile collection of data in support of management		phase of a knowledge discovery process
51.	A definition or a concept is if it classifies any examples as coming within the concept _____	Complete	Consistent	constant	consecutive	Consistent
52.	Data independence means _____	Data is defined separately and not included in programs	Programs are not dependent on the physical attributes of data.	Programs are not dependent on the logical attributes of data.	Both Programs are not dependent on the physical attributes of data and Programs are not dependent on the logical attributes of data.	Both Programs are not dependent on the physical attributes of data and Programs are not dependent on the logical attributes of data.
53.	E-R model uses this symbol to represent weak entity set?	Dotted rectangle	Diamond	Doubly outlined rectangle	Chinese wall	Doubly outlined

						rectangle
54.	SET concept is used in _____	Network Model	Hierarchical Model	Relational Model	None	None
55.	Relational Algebra is _____	Data Definition Language	Meta Language	Procedural query Language	Query Language	Procedural query Language
56.	Key to represent relationship between tables is called _____	Primary key	Secondary Key	Foreign Key	Product Key	Foreign Key
57.	_____ produces the relation that has attributes R1 and R2	Cartesian product	Difference	Intersection	Product	Cartesian product
58.	Which of the following are the properties of entities?	Groups	Table	Attributes	Switchboards	Attributes
59.	In a relation _____ i) Ordering of rows is immaterial      ii) No two rows are identical	i is true	ii true	i and ii are true	All are False	i and ii are true
60.	_____ are any facts, number or text can be processes by a computer.	Data	Information	Record	Field	Data